Exploration of the Evolution of Airport Ground Delay Programs

by

Kexin Ren

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

TRANSPORTATION ENGINEERING

Department of Civil and Environmental Engineering
University of Alberta

Abstract

A Ground Delay Program (GDP) delays flights at their departure airports on ground to absorb their potential airborne delays, when an imbalance between flight demand and airport capacity is expected at the arrival airport. The existing literature have looked into the issues existing before (e.g. weather forecasts), during (e.g. GDP planning) and after (e.g. GDP evaluation) the initiation of GDPs. However, no one has explored how GDPs evolve over the course of their lifetimes (typically, a day).

The thesis introduces a novel method of merging disparate but complementary datasets and applying data mining techniques to gain more insights into GDPs – particularly with respect to their evolving characteristics. More specifically, it aims to characterize GDPs with respect to realized weather conditions, and individual flight information. This research then identified several scenarios of GDP evolution based on the merged master dataset, by first reducing the dimensionality of the master GDP dataset, then applying cluster analysis on the resulting lower-dimensional data. We found that GDPs at EWR can be categorized into 10 types based on (changing) weather forecasts, GDP scope, program rate, and duration. This research then further explored the characteristics of these 10 GDP evolution scenario clusters by examining the relationships between GDP scenarios and their performance (described using metrics previous developed by other researchers) using statistical analysis. It was found that GDPs under stable, low-severity weather and with large scope may score higher on the efficiency metric than we would expect. When GDPs called in similar weather conditions have high program rates, medium durations, and narrow scopes, it was found that capacity utilization is higher than expected – less flights lead to fewer cancellations and more arrivals (albeit delayed), and therefore, higher capacity utilization. Results also suggest that program rates are set more conservatively than needed for

some poor weather conditions that end earlier than expected, with GDP being canceled early as well. GDPs with fewer revisions were associated with a higher predictability score but lower efficiency score.

These findings can provide greater insights and knowledge about GDPs for future planning purposes. For future work, it is recommended that additional data be utilized to provide a more comprehensive operational picture of GDPs, and that a wider range of performance metrics be considered in the analysis. In addition, it is also recommended that the patterns of how GDPs evolve over their lifetimes be further explored using other novel machine learning techniques that may provide new and useful insights.

Keywords: Ground Delay Programs, GDP evolution characterization, unsupervised learning, clustering analysis, multivariate statistical analysis.

Dedicated to my mother

Acknowledgements

Firstly, I would like to express my gratitude to my advisor, Dr. Amy Kim. Thank her for leading me into such an interesting research field. With her great guidance and patience, I have learned how to be professional in research work as a research assistant. I am also grateful that Dr. Kenneth Kuhn has provided me with lots of help and encouragements. Without the support from Dr. Kim and Dr. Kuhn, it would have been impossible for me to complete the thesis. I would like to thank Dr. Kenneth Kuhn and Dr. Lijun Deng for being my thesis committee members. My sincere thanks also go out to Dr. Karim El-Basyouny and Dr. Tony Qiu for their help during my master study.

Secondly, I feel very lucky to know my great colleagues. I would like to thank Yang (Naomi) Li, Qianqian Du, Kathy Hui, Gloria Duran Castillo, Yunzhuang Zheng, Kasturi Mahajan, Matthew Woo, Can Zhang, Jiangchen Li, Chenhao Wang, Yuwei Bie, Ai Teng, Chen Qiu, Lian Gu, and Maged Gouda for their support and help in the past two years.

Finally, special thanks to my mother and Danyang Sun for their understanding, encouragement and company. It is their love that makes me brave in the face of difficulties.

.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ARTCC	Air Route Traffic Control Center
ATCSCC	Air Traffic Control System Command Center
EWR	Newark Liberty International Airport
FAA	Federal Aviation Administration
GDP	Ground Delay Program
METAR	Meteorological Terminal Aviation Routine Weather Report
NAS	National Airspace System
TAF	Terminal Aerodrome Forecast
TMI	Traffic Management Initiative
CFA	Configural Frequency Analysis

Abbreviations specific to this thesis

CL	Ceiling
CW	Crosswinds
DR	GDP Planned Advisory Duration
DS	GDP Departure Scope
IF	Individual Flight
PC	Precipitation
PR	GDP Program Rate
TS	Thunderstorm
VC	Visibility

Chapter 1. Introduction

1.1 Background

Airport arrival capacity is vulnerable to severe weather, in addition to other disturbances such as less desirable runway usage configurations and even runway closures (Liu & Hansen, 2014). Capacity reductions usually result in demand-capacity imbalances at the airport, in turn resulting in flight delays that are costly to passengers and airlines. Besides increasing airport arrival capacity by airport infrastructure expansion which requires huge investments (of public money, in the U.S.), the Federal Aviation Administration (FAA) has developed Traffic Management Initiatives (TMIs) to resolve the imbalance between flight demand and capacity during times of inclement and less-than-ideal conditions, by shifting the demand to alternative resources, such as different routes, or later (delayed) times (Manley & Sherry, 2008; Xiong, 2010).

When there is a shortfall in capacity at an airport, arriving flights may suffer airborne delay as they are forced to queue in the air. When a demand-capacity imbalance is expected at an airport, a Ground Delay Program (GDP) is implemented. The GDP is a TMI that transfers potential airborne delays to cheaper and safer ground delays by holding some aircraft at their departure airports for a period of time (Ball & Lulli, 2004; Barnhart, Bertsimas, Caramanis, & Fearing, 2012). A GDP is applied to an airport with a pre-specified start time, stop time, and Program Rate (PR – the allowed flight arrival rate). GDPs are planned under a framework called Collaborative Decision Making (CDM), in which the Air Traffic Control System Command Center (ATCSCC), FAA, and airlines communicate and collaborate to improve traffic flow management. The ATCSCC implements GDPs after communicating with regional FAA centers and airline

operations centers, and then airlines are able to swap and cancel flights based on the GDP details (Willemain, 2002; Hoffman, Ball, & Mukherjee, 2007; Ball, Hoffman, & Knorr, 2000).

The GDP has become one of the most commonly used TMIs since implementation in 1981 (Donohue, Shaver, & Edwards, 2008). In 2005, more than 1,350 GDPs were issued in the U.S. and they assigned delays to over 530,000 flights with a total of 16.8 million minutes (Ball, Hoffman, & Mukherjee, 2010). In 2011, there were 1,065 GDPs implemented in the U.S., and they assigned delays totaling 26.8 million minutes distributed over 519,940 flights (Liu & Hansen, 2014).

Considering the extensive use of GDPs and their deep and wide-ranging impacts within the National Airspace System (NAS), much effort has been made to study and improve them.

1.2 Motivation and objective

The majority of the existing literature focuses on GDP design, such as improving GDP planning by accounting for weather forecasts, with one (delay minimization) or more performance objectives. Efforts have been made to generate airport capacity profiles for GDP planning from weather forecasts. Some research has also been conducted to evaluate GDP performance retrospectively. Overall, the existing studies have provided some insights into the issues existing before, during and after the implementation of GDPs. However, there has been no research exploring how GDPs evolve over the course of their lifetimes (typically, a day).

To fill this gap in the literature, this research aims to characterize GDPs with respect to changing weather forecasts, GDP plan parameters, and operational performance. The purpose of this analysis is to gain some insights into the temporal patterns of GDPs with respect to these several key dimensions, by describing GDP performance in response to key (changing) variables. This research first generated a master dataset by merging several datasets on GDPs, weather forecasts, and individual flight information. This research then identified several scenarios of GDP

evolution based on the merged master dataset, by first reducing the dimensionality of the master GDP dataset, then applying cluster analysis on the resulting lower-dimensional data.

1.3 Research Scope

This research effort focused on data from Newark Liberty International Airport (EWR) from 2010 through 2014. Based on TMI advisory, weather forecast, and flight data, this research applied data mining techniques to better observe the characteristics of GDPs as they evolved over their lifetimes (i.e. the course of a day). As shown in Figure 1.1, this research first developed a master dataset through the merging of several datasets of weather forecasts, realized weather, TMI advisories, and individual flights information. Second, this research provided some basic information regarding the characteristics of weather and GDPs at EWR. Third, we used the autoencoder technique to visualize the data and reduce 585 dimensions of GDP evolution into two, in order to support the cluster analysis. Fourth, this research identified GDP evolution scenarios through cluster analysis based on the compressed 2-dimensional data. Finally, this research assessed correlations between the identified GDP clusters and performance, using Configural Frequency Analysis.

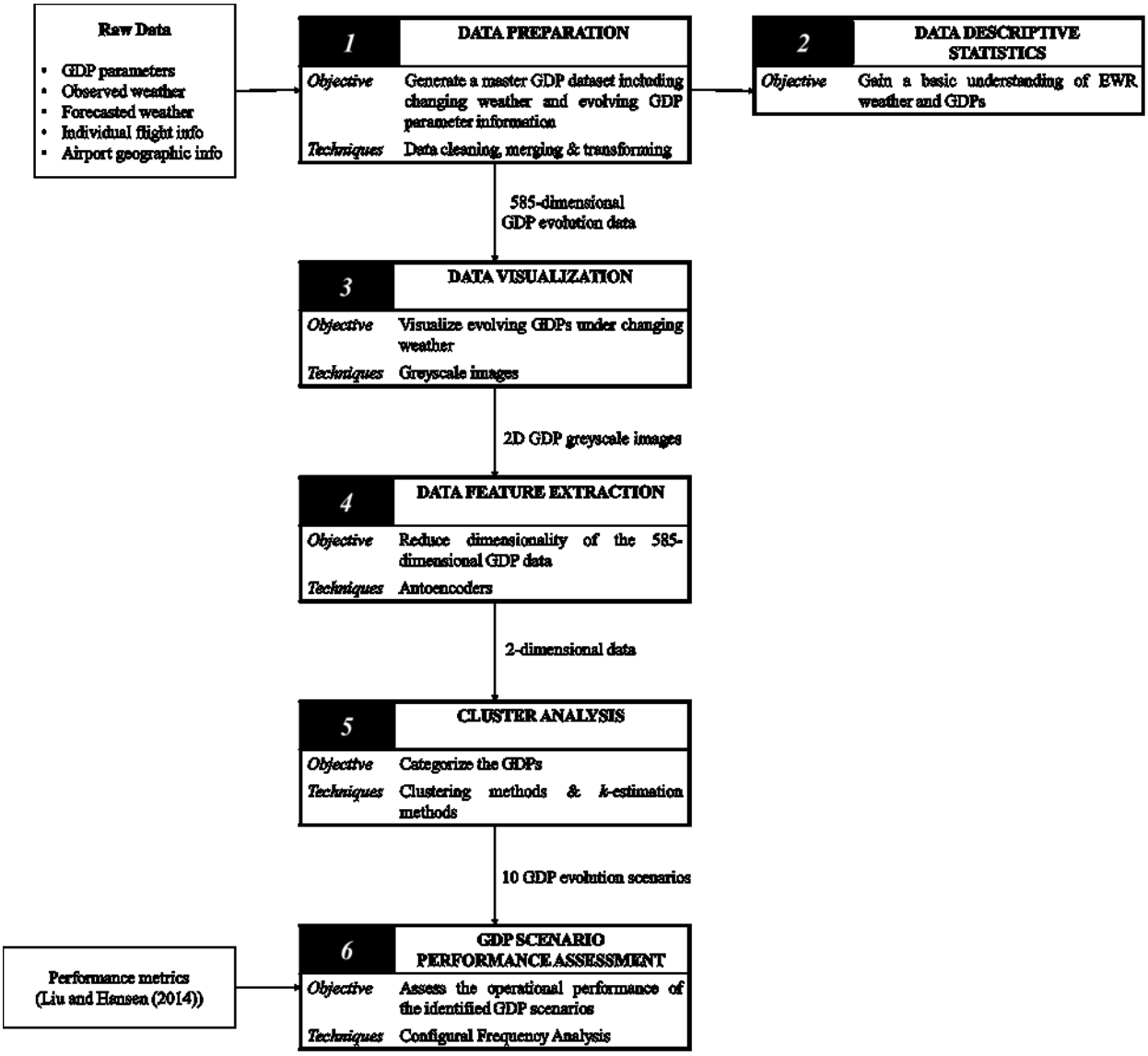


Figure 1.1 Research steps taken

In the third step, this research reduced data dimensionality because the GDP evolution data involves various types of data (i.e. GDP parameters, forecasted weathers, time indexes) and many dimensions (585 in total), which does not allow for regression or cluster analysis. Autoencoder, a machine learning technique, was applied because it has been successfully applied to solve similar problems, such as greyscale handwritten digits classification (Hinton & Salakhutdinov, 2006), and

I believed it could be appropriate applied in our research context. Then this research categorized the GDPs based on the compressed data through cluster analysis; because GDPs are the results of human-driven decision-making (i.e. by air traffic controllers), it is likely that repeated patterns of decision-making would emerge. This research determined 10 scenarios of GDP evolution through comparing the results of different clustering methods and *k*-estimation methods.

1.4 Thesis structure

This thesis consists of five chapters. Chapter 2 presents a literature review GDPs and identifies gaps in the literature. Chapter 3 introduces the data and data preparation procedures, and also provides basic descriptive statistics. Chapter 4 introduces the techniques and approaches used to illuminate the evolution characteristics of GDPs at EWR. Chapter 5 presents conclusions, highlights the contributions, and recommends future extensions of this research.

Chapter 2. Literature review

This chapter provides a brief review of the existing (academic and applied) literature on Ground Delay Programs (GDPs), and identifies gaps in this literature.

2.1 Introduction to Ground Delay Programs

A GDP was first implemented by the Federal Aviation Administration (FAA) in 1981. It is a traffic management initiative that delays flights at their departure airport in order to control inbound flight volumes at the airport where an imbalance between flight capacity and demand is expected (FAA, 2009a). The purpose of a GDP is to absorb flight delays (due to arrival airport demand-capacity imbalance) on the ground before take-off rather than in the air, to reduce the probability and duration of airborne delay, as delays incurred on the ground are significantly cheaper (in terms of fuel savings and personnel hours) and safer (Ball & Lulli, 2004). Airport capacity shortfall is the most common reason for implementing a GDP at an airport; capacity reduction is normally a result of adverse weather conditions (FAA, 2017a) such as low ceiling, low visibility, thunderstorms or strong winds.

The planned airport capacity and GDP duration are determined by the FAA Air Traffic Control System Command Center (ATCSCC) based on the predicted conditions of the airports. The ATCSCC is a facility responsible for balancing air traffic demand to National Airspace System (NAS) system capacity (FAA, 2014a). Flights' controlled arrival times (CTAs) are calculated and converted to controlled departure times (CTDs) and result in (ground) delays, using a software called Flight Schedule Monitor. Furthermore, since 1998, GDPs have been implemented under the Collaborative Decision Making (CDM) framework, in which the FAA,

airlines, and other airspace users cooperate with one other to improve the safety and efficiency of traffic flow management (Ball, Hoffman, Chen, & Vossen, 2000; Reynolds, Clark, Wilson, & Cook, 2012). Under CDM, the ration-by-schedule (RBS) algorithm has been applied by the FAA to allocate arrival slots to carriers. Under RBS, flights are prioritized according to their original scheduled arrival times at the GDP airport (FAA, 2017b; Jonkeren, Rietveld, & Ommeren, 2007). Airlines have the flexibility to swap, substitute, or cancel their own flights based on the RBS allocation.

However, RBS has some limitations for equity and efficiency; for example, flights originating from more distant airports may suffer excessive delays when the capacity is increased and GDP is cancelled (Manley & Sherry, 2008; Ball & Lulli, 2004). Thus, in reality, a distance-based RBS algorithm (DB-RBS) is applied. In DB-RBS, flights departing from airports beyond a certain radius of the arrival airport are exempted from the GDP, while flights originating at airports within the radius will be assigned ground delays. In addition, airborne (regardless of their origins) and international flights are also exempted from GDPs (Vossen & Ball, 2005; Hoffman, Ball, & Mukherjee, 2007; Barnhart, Bertsimas, Caramanis, & Fearing, 2012).

2.2 Previous studies on GDPs

The existing literature has spanned a variety of topics regarding different stages in the life time of a GDP. This section summarizes the literature, according to GDP planning and lifecycle stages, and discusses the limitations of the literature.

2.2.1 Airport capacity modeling for planning GDPs

Adverse weather is the most common cause of airport capacity reductions (FAA, 2017a). The accuracy of weathers forecasts impact airport arrival capacity predictions, and flights

experience unnecessary delays when predicted capacity does not well reflect realized capacities (Buxi & Hansen, 2011). Thus, there has been research focusing on modelling arrival capacity considering the probabilistic nature of weather forecasts. Richetta and Odoni (1994) first modeled airport capacity by assuming a limited number of scenarios considering the uncertainty in weather forecasts. Liu et al. (2008) demonstrated that capacity scenarios at several major U.S. airports can be inferred from historical data. Inniss and Ball (2004) and Buxi and Hansen (2011) employed weather forecasts to generate probabilistic capacity profiles.

2.2.2 GDP planning

A large number of studies focus on designing GDPs by accounting for airport capacity uncertainty caused by adverse weather conditions. In static models, decisions are only made once and will not be revised, while in dynamic GDP models, GDP decisions can be revised according to the updated capacity forecasts (Buxi & Hansen, 2011). Richetta and Odoni (1993) first developed a static stochastic model while Mukherjee and Hansen (2007) developed a dynamic stochastic model, which were the first of these models and have subsequently been built upon in subsequent research. Some studies minimize delay as a sole objective, or trade-off multiple performance objectives for GDP design. For example, Inniss and Ball (2004) optimized GDPs by determining the appropriate amount of ground delays to be assigned to flights considering probabilistic capacity forecasts; Liu et al. (2017) determined GDP file time, end time, and distance under uncertain airport capacity, taking into account GDP operational efficiency, airline and flight equity, and Air Traffic Control (ATC) risks.

GDP decision support tools based on weather forecasts can be used to notify traffic controllers to take actions for dealing with the expected capacity reduction appropriately (Mukherjee, Grabbe, & Sridhar, 2014). Most related studies developed GDP decision support tools

to predict GDP occurrence, the Airport Arrival Rate (AAR) or airport delays using machine learning techniques based on historical data. Mukherjee et al. (2014) and Bloem and Bambos (2015) focused on the prediction of the occurrence of a GDP during a given hour based on forecasted weather conditions; Kulkarni et al. (2013) and Wang (2011) focused on airport capacity prediction using different machine learning methods; Smith et al. (2008) predicted Aircraft Arrival Rates (AAR) and airport delays using Support Vector Machine (SVM) based on weather forecasts, to determine GDP program rate, duration and estimate passenger delays. Kuhn (RAND Corporation) (2016a) defined a methodology in which past GDPs issued under conditions similar to a present situation are ranked in order to make recommendations to traffic controllers regarding GDPs to be implemented at present.

There has also been some research on designing GDPs according to principles and objectives that differ from the currently adopted distance-based Ration-by-Schedule algorithm (DB-RBS). Hoffman, Ball and Mukherjee (2007) designed a Ration-by-Distance (RBD) algorithm in which flights are prioritized by their travel distance when allocating arrival slots. Manley and Sherry (2008) examined the performances of Ration-by-Passengers (RBPax) and Ration-by-Aircraft Size (RBAcSize), and demonstrated that the current rationing rule (DB-RBS) can be improved by taking into account the trade-off between airline equity and passenger flow efficiency. Delgado et al. (2013) designed a new strategy where aircraft can recover more delays without extra fuel consumption by departing earlier and flying slowly, instead of absorbing all delays on the ground, in early cancelled GDP events. Zhang and Hansen (2009) indicated that through evaluating the capacity of a hub airport together with that of other airports in the same region, traffic controllers can issue a regional GDP to improve the overall regional system efficiency.

2.2.3 Performance evaluation

Only a few studies have introduced GDP performance metrics and then evaluated performance retrospectively, due to the difficulty of accessing high quality, operational-level data (such as that used in this research), and the complexity and onerousness of the analysis procedures. Hoffman and Ball (2000) first defined a single metric called the rate control index, which measures actual against planned traffic flow, to evaluate the effectiveness of a GDP. According to a list of 11 globally endorsed key air transport performance criteria proposed by ICAO (International Civil Aviation Organization, 2005), Liu and Hansen (2014) proposed five GDP performance criteria and corresponding metrics – capacity utilization, predictability, efficiency, equity, and flexibility. Kuhn (RAND Corporation) (2016b) identified six criteria and metrics – efficiency, safety, equity, predictability, flexibility, and adherence.

2.3 Summary

The previous literature on GDPs is extensive in several aspects, although there has not been as much work on post-GDP performance evaluation asides from Liu and Hansen (2014). Moreover, there has been little to no work exploring how GDPs evolve over time (typically, over the course of a day).

To address this gap, this research attempts to characterize GDP evolution over their lifetimes (i.e. a day). In order to achieve the goal, the following work will be performed: 1) categorize the GDPs that were planned under particular weather forecasts, and how they evolved as weather forecasts evolved, using data mining tools and based on historical weather forecast data, GDP plan data, and flight schedule data; and 2) identify correlations between different types of GDP evolution and GDP performance through statistical multivariate analysis (Configural Frequency Analysis, or CFA).

Chapter 3. Dataset introduction and preparation

This chapter introduces the datasets used in the research, which include those of weather forecasts, GDP information, and individual flight data for arrivals into EWR from 2010 through 2014. Three datasets were merged to prepare a single master dataset for the GDP evolution characterization work documented later in Chapter 4. Data descriptive statistics are presented here in order to provide a preliminary overview of weather and GDP characteristics.

3.1 Background on Newark Liberty International Airport (EWR)

Newark Liberty International Airport (EWR) is one of the busiest airport in the U.S. and experiences significant flight delays (McCartney, 2011). The TMI data used in this study (introduced in 3.2.1) indicates that EWR GDPs accounted for 15% of the GDPs implemented in the U.S. from 2010 to 2014, a significant number.

EWR is located about 14 miles from New York City. EWR, LaGuardia Airport (LGA), and John F. Kennedy International Airport (JFK) are the three major airports in the New York metropolitan area, all operated by the Port Authority of New York and New Jersey (The Port Authority of New York and New Jersey, 2017). Flights into and out of EWR are managed by the New York Air Route Traffic Control Center (ZNY). An Air Route Traffic Control Center (ARTCC) is a facility responsible for en route control of aircraft operating under Instrument Flight Rules within controlled airspace (FAA, 2009b).

EWR's location is shown in Figure 3.1. The airport serves more than 30 airlines for both domestic and international flights, and is the main hub for United Airlines. United Airlines dominates international service to major cities in Europe, Asia, the Middle East, the Pacific region,

and the Americas. As one of the primary airports serving New Jersey and the New York metropolitan area, EWR served over 35 million passengers in 2015 (The Port Authority of New York and New Jersey, 2015).

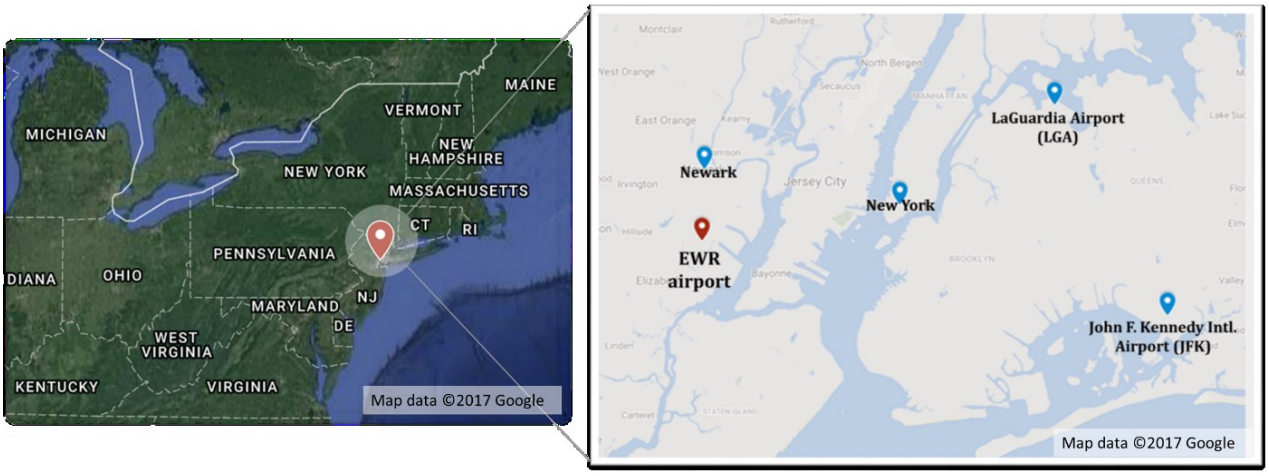


Figure 3.1 Newark Liberty International Airport (EWR) geographic location (Source: Google Maps, 2017)

Figure 3.2 shows the airside plan diagram for EWR (FAA, 2008), EWR has two parallel runways (Runway 4R/22L and Runway 4L/22R) which cross Runway 11/29. Typically, departing aircraft take off on the inner runway, Runway 4L/22R, and arriving aircraft use Runway 4R/22L, while Runway 11/29 is used less frequently (FAA, 2014d). The most frequently used runway configuration in VMC[1] is 22L|22R, while in IMC[2] it is 4R|4L (Kim, Rokib, & Liu, 2015).

[1] Visual meteorological conditions (VMC): conditions which provide sufficient visibility for pilots to operate aircraft with visual references. At EWR, VMC requires at least 3000 ft ceilings and 4 miles visibility.
[2] Instrument meteorological conditions (IMC): conditions which necessitate instrument indications for pilots to operate aircraft in poor visibility conditions. For EWR, it occurs with less than 1000 ft ceilings or 3 miles visibility (FAA, 2014d).

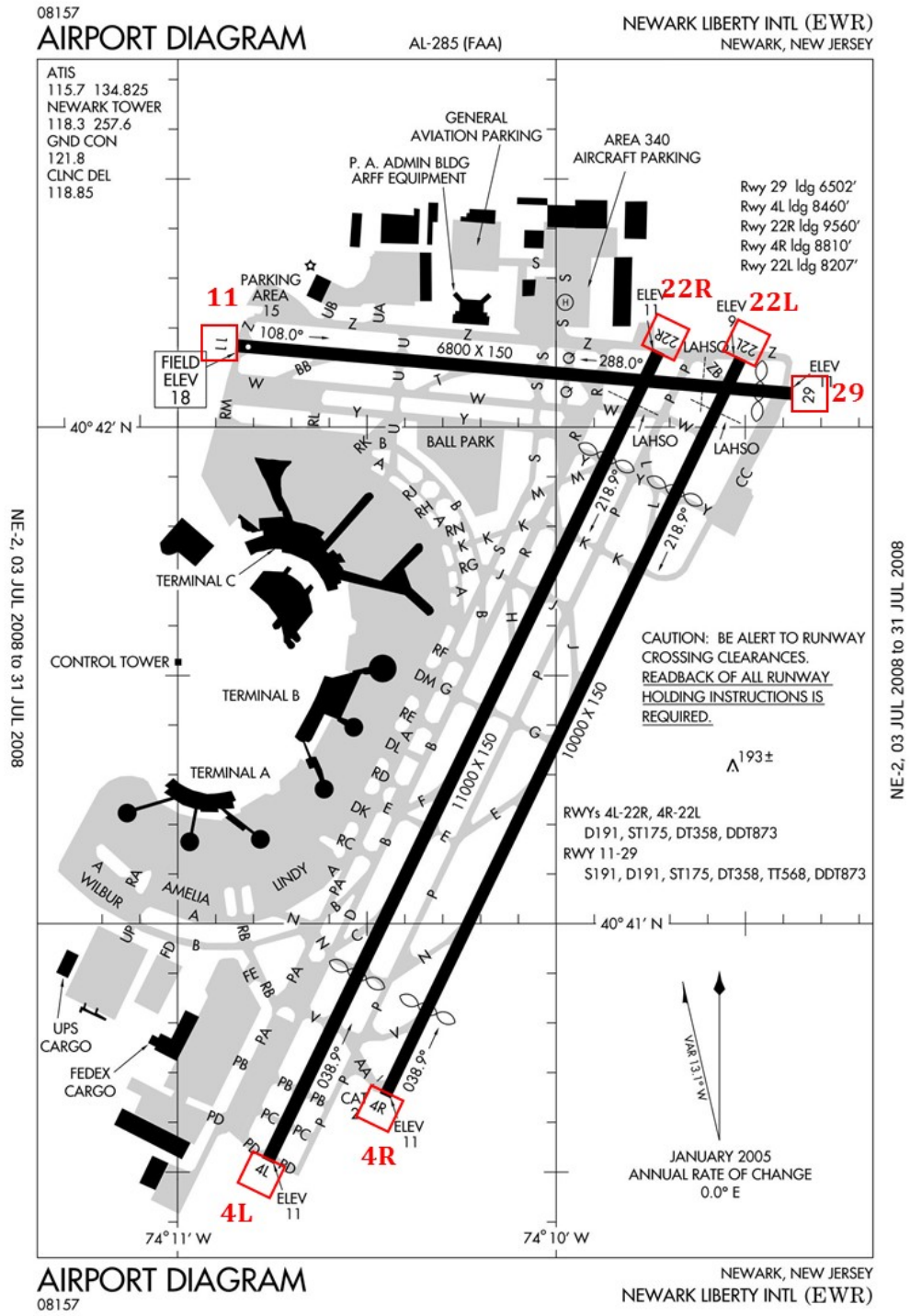


Figure 3.2 EWR airport layout (FAA, 2008)

3.2 Data sources

We use three datasets, plus other supplemental data, to generate a comprehensive merged dataset for this research. GDP plan data was obtained in the FAA Air Traffic Flow Management Initiative (TMI) advisories database. Weather forecast data was collected from the Terminal Aerodrome Forecast (TAF) database. Flight departure and arrival information was extracted from the Individual Flight (IF) dataset from the FAA's Aviation System Performance Metrics (ASPM) database. In addition, airport longitudes, latitudes, countries and Air Route Traffic Control Centers were gathered to supplement the three datasets identified.

3.2.1 Air Traffic Flow Management Initiative (TMI) Advisories

The FAA Air Traffic Flow Management Initiative (TMI) Advisory Database contains the Air Traffic Control System Command Center (ATCSCC) advisories and Canadian Advisories, reporting planned TMIs, modifications of planned TMIs, and cancellations of TMIs. The TMIs mainly include Ground Delay Program (GDP), Ground Stop (GS), Reroute, and Airspace Flow Program (AFP).

We extracted 21 variables (columns) from the original TMI advisories dataset, including advisory type, dates, times, causes, impacted scopes and program rates. The 18 variables are described in Table 3.1.

With the goal of analyzing GDPs at EWR, this research extracted advisories with Advisory Category "GDP" (Column 5 in Table 3.1), Control Element "EWR/ZNY" (Column 7 in Table 3.1) and implemented in 2010 through 2014 (Column 7 in Table 3.1). There were finally 2,410 advisories (rows) in total meeting these conditions, including 765 root advisories.

Table 3.1 Variables in TMI data set

Column No.	Column name	Description
1	Year	Advisory send year
2	AdvisoryDate UTC	Advisory send date
3	AdvisoryNumber	Label of the advisory
4	SendDate.Time.UTC	Advisory send date and time (time zone = GMT)
5	AdvisoryCategory	TMI category, including GDP, GS, AFP, Reroute, etc. Here, it should be "GDP".
6	AdvisoryType	Advisory type, "GDP" or "GDPX" (GDP cancellation)
7	ControlElement	The Air Route Traffic Control Center (ARTCC) which issued the advisory. Here, it should be "EWR/ZNY".
8	RootAdvisoryDate.UTC	Send date of this advisory's root advisory
9	RootAdvisoryNumber	Advisory Number of this advisory's root advisory
10	Derived.BgnDate.Time.UTC	The begin time of the GDP or GDPX advisory, equal to column 15 or 17 (time zone = GMT)
11	Derived.EndDate.Time.UTC	The end time of the GDP or GDPX advisory, equal to column 16 or 18 (time zone = GMT)
12	Is.RootAdvisory	Whether this advisory is a root advisory ("Yes" or "No")
13	Canadian.Dep.Arpts.Included	Impacted Canadian departure airports included in the advisory
14	Dep.Scope	Impacted departure scope, a radius or a set of Air Route Traffic Control Centers (ARTCC). Airports outside this scope are exempt from the advisory.
15	GDP.Bgn.Date.Time.UTC	GDP begin time (time zone = GMT)
16	GDP.End.Date.Time.UTC	GDP end time (time zone = GMT)
17	GDPX.Bgn.Date.Time.UTC	GDP cancel begin time (time zone = GMT)
18	GDPX.End.Date.Time.UTC	GDP cancel end time (time zone = GMT)
19	Impacting.Condition	Causes of the advisory
20	Program.Rate	The number of aircraft that the GDP software is to provide to the airport, for each hour.
21	Exempt.Dep.Facilities	Airports exempt by the advisory

3.2.2 Terminal Aerodrome Forecast (TAF) data

A Terminal Aerodrome Forecast (TAF) is a concise statement issued by the U.S. National Weather Service (NWS) for all major U.S. airports, reporting forecasted meteorological conditions at each airport. It contains forecasts about visibility, ceiling, winds, and other meteorological features of interest (National Weather Service, 2016). Four to eight TAFs are issued every six

hours and generally cover a 24- to 30-hour period following the forecast (Federal Aviation Administration; National Weather Service, 2010).

We retained 28 variables (columns) in the TAF data including TAF issue and forecast coverage times, forecast visibility, ceiling, winds, thunderstorms and precipitations. The variables are presented in Table 3.2. TAFs issued at EWR from January 1, 2010 to August 31, 2014 were available to us. After removing duplicate TAFs and TAFs with illogical duration (negative or too long), the final dataset contained 96,829 TAF records (rows), averaging 58 records per day.

Table 3.2 Variables in TAF data set

Column No.	Column Name	Description
1	Issued Year	Year of the TAF issue date time
2	Issued Month	Month of the TAF issue date time
3	Issued Day	Day of the TAF issue date time
4	Issued Hour	Hour of the TAF issue date time
5	Issued Minute	Minute of the TAF issue date time
6	From Year	Year of the forecast start date time
7	From Month	Month of the forecast start date time
8	From Day	Day of the forecast start date time
9	From Hour	Hour of the forecast start date time
10	From Minute	Minute of the forecast start date time
11	To Year	Year of the forecast end date time
12	To Month	Month of the forecast end date time
13	To Day	Day of the forecast end date time
14	To Hour	Hour of the forecast end date time
15	To Minute	Minute of the forecast end date time
16	Wind Angle	Forecasted wind angle (degrees)
17	Wind Speed	Forecasted wind angle (knots)
18	Visibility	Forecasted visibility (miles)
19	Ceiling	Forecasted ceiling (100 feet)
20	RA	Forecasted occurrence of rain (1 = yes, 0 = no)
21	DZ	Forecasted occurrence of drizzle (1 = yes, 0 = no)
22	SN	Forecasted occurrence of snow (1 = yes, 0 = no)
23	SG	Forecasted occurrence of snow grains (1 = yes, 0 = no)
24	GR	Forecasted occurrence of hail (1 = yes, 0 = no)
25	GS	Forecasted occurrence of snow pellets (1 = yes, 0 = no)
26	IC	Forecasted occurrence of ice crystals (1 = yes, 0 = no)
27	UP	Forecasted occurrence of unknown precipitation (1 = yes, 0 = no)
28	TS	Forecasted occurrence of thunderstorm (1 = yes, 0 = no)

3.2.3 Aviation Routine Weather Report (METAR) data

METAR is the meteorological code for an *aviation routine weather report* and it is a format for reporting observational surface weather data by the U.S. National Weather Service (NWS). METARs are generated and published once an hour (UQAM Atmosphere Sciences Group, 2017).

In this study, METAR data for EWR in the period of January 1, 2010 to December 31, 2014 were used. It contains 55,459 records (rows) and 15 variables (columns). The variables of the dataset are presented in Table 3.3.

Table 3.3 Variables in METAR data set

Column No.	Column name	Description
1	start.time	Start date and time of the METAR observation
2	end.time	End date and time of the METAR observation
3	Wind.Angle	Observed wind angle (degrees)
4	Wind.Speed	Observed wind angle (knots)
5	Visibility	Observed visibility (miles)
6	Ceiling	Observed ceiling (100 feet)
7	RA	Observed occurrence of rain (1 = yes, 0 = no)
8	DZ	Observed occurrence of drizzle (1 = yes, 0 = no)
9	SN	Observed occurrence of snow (1 = yes, 0 = no)
10	SG	Observed occurrence of snow grains (1 = yes, 0 = no)
11	GR	Observed occurrence of hail (1 = yes, 0 = no)
12	GS	Observed occurrence of snow pellets (1 = yes, 0 = no)
13	IC	Observed occurrence of ice crystals (1 = yes, 0 = no)
14	UP	Observed occurrence of unknown precipitation (1 = yes, 0 = no)
15	TS	Observed occurrence of thunderstorm (1 = yes, 0 = no)

3.2.4 Individual Flight (IF) data

The FAA Individual Flight (IF) Database of Aviation System Performance Metrics (ASPM) provides detailed information about flights such as departure and arrival times and flight delays. The time information includes actual, flight plan, ETMS plan and scheduled times of Gate

Out, Wheels Off, Wheels On and Gate In[3], from TFMS (formerly known as ETMS), OOOI and ASQP records. TFMS refers to Traffic Flow Management System, a data exchange system providing flow information (FAA, 2014b); OOOI data refers to actual aircraft movement times of Gate Out, Wheels Off, Wheels On, and Gate In (FAA, 2015); ASQP refers to the Airline Service Quality Performance System (ASQP) which provides flight delay information (FAA, 2014c).

We selected 37 variables (columns) from the original IF data. The variables are introduced in Table 3.4. By restricting the arrival airport as EWR, and arrival date between January 1 2010 through December 31 2014, this research was left with 879,507 flights (rows) in the dataset. The departure airports consist of both U.S. and international airports. 18% of the flights were departed from foreign airports.

Table 3.4 Variables in IF data set (FAA, 2003)

Column No.	Column name	Description
1	DEP_YYYYMM	Scheduled Departure Year and Month (Local Date)
2	DEP_DAY	Scheduled Departure Day (Local Day)
3	DEP_HOUR	Scheduled Departure Hour (Local Hour)
4	DEP_QTR	Scheduled Departure Quarter Hour (Local Qtr)
5	ARR_YYYYMM	Scheduled Arrival Year and Month (Local Date)
6	ARR_DAY	Scheduled Arrival Day (Local Day)
7	ARR_HOUR	Scheduled Arrival Hour (Local Hour)
8	ARR_QTR	Scheduled Arrival Quarter Hour (Local Qtr)
9	OFF_YYYYMM	Actual Wheels Off Year and Month (ASQP/OOOI Off Local Date)
10	OFF_DAY	Actual Wheels Off Day (ASQP/OOOI Off Local Day)
11	OFF_HOUR	Actual Wheels Off Hour (ASQP/OOOI Off Local Hour)
12	OFF_QTR	Actual Wheels Off Quarter Hour (ASQP/OOOI Off Local Qtr)
13	ON_YYYYMM	Actual Wheels on Year and Month (ASQP/OOOI On Local Date)
14	ON_DAY	Actual Wheels on Day (ASQP/OOOI On Local Day)
15	ON_HOUR	Actual Wheels on Hour (ASQP/OOOI On Local Hour)
16	ON_QTR	Actual Wheels on Quarter Hour (ASQP/OOOI On Local Qtr)
17	FAACARRIER	Flight Carrier Code - ICAO

[3] Gate Out: Aircraft leaves gate or parking position.
Wheels Off: Aircraft takes off.
Wheels On: Aircraft touches down.
Gate In: Aircraft arrives at gate or parking position.

Column No.	Column name	Description
18	FLTNO	Flight Number
19	Dep_LOCID	Departure Location Identifier: Domestic = space + 3-character identification code, foreign = ICAO 4-character identification code
20	Arr_LOCID	Arrival Location Identifier: Domestic = space + 3-character identification code, foreign = ICAO 4-character identification code.
21	SchOutTm	Scheduled Gate Departure Time (Local) HH:MM
22	FPDepTm	Flight Plan Gate Departure Time HH:MM
23	ActOutTm	Actual Gate Out Time HH:MM
24	SchOffTm	Scheduled Wheels Off Time HH:MM
25	FPOffTm	Flight Plan Wheels Off Time HH:MM
26	ActOffTm	Actual Wheels Off Time HH:MM
27	DlaSchOff	Airport Departure Delay Minutes (Based on Schedule)
28	DlaFPOff	Airport Departure Delay Minutes (Based on Flight Plan)
29	DELAY_AIR	Airborne Delay Minutes
30	EDCTOnTm	Wheels on Time HH:MM (Filed on EDCT)
31	ActOnTm	Actual Wheels on Time HH:MM
32	EDCTArrDif	Difference between EDCT Expected and Actual Wheels-On (EDCT Arrival)
33	SchInTm	Scheduled Gate-In HH:MM
34	FPInTm	Flight Plan Gate-In HH:MM
35	ActInTm	Actual Gate In Time HH:MM
36	DlaSchArr	Arrival Delay in Minutes (Compared to Scheduled)
37	DlaFPArr	Arrival Delay in Minutes (Compared to Flight Plan)

3.2.5 Airport Information (AI) Dataset

We combined airport geographic information gathered from OpenFlights database (OpenFlights, 2017) and the calculated distances between EWR and other airports to create a dataset entitled Airport Information (AI). The OpenFlights database contains airport longitudes and latitudes, countries, and Air Route Traffic Control Centers (within the NAS) for airports with flights departing to EWR. The GDP departure scope in the TMI dataset (Column 14 in Table 3.1) identifies the airports included in the GDP scope by specifying a radius (in miles) or a set of ARTCC. Flights destined for EWR from departure airports within the geographic scope can be assigned ground delays as specified by the GDP parameters. For GDPs with a radius-defined scope, this research inferred the impacted departure airports by calculating the distances between the

departure airports and EWR airport based on their longitudes and latitudes; similarly, for GDPs with an ARTCC-defined scope, this research matched the impacted ARTCCs to airports within these ARTCCs using the airport-ARTCC membership information contained in the AI dataset. This is how this research determined whether a flight was involved in a particular GDP – by matching the flight departure airport (Column 19 in Table 3.4) with the GDP departure scope (Column 14 in Table 3.1). The variables in AI data set are listed in Table 3.5.

Table 3.5 Variables in AI data set

Column No.	Column name	Description
1	AirportID	Airport identifier: domestic = space + 3-character identification code, foreign = ICAO 4-character identification code
2	Country	The country in which the airport is located
3	City	The city in which the airport is located
4	Latitude	Latitude of the airport
5	Longitude	Longitude of the airport
6	ARTCC	The ARTCC which the airport belongs to (for U.S. airports and some Canadian airports only)
7	Distance	The distance between the airport and EWR airport (in miles)

In addition, the great-circle distances (in miles) between the departure airports and EWR were calculated using the following formula (Veness, 2016):

$$d = \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \times 2 \times 3959 \quad (1)$$

Where:

$a = sin^2 \frac{\Delta\varphi}{2} + cos\,\varphi_1 \cdot cos\,\varphi_2 \cdot sin^2 \frac{\Delta\lambda}{2}$;

φ_1, φ_2 are the latitudes of EWR and the departure airport, respectively, and

$\Delta\lambda$ is the difference between the longitudes of EWR and the departure airport.

3.3 Data preparation

The data was first preprocessed through filtering, cleaning and time zone unifying. Next, some new variables were calculated and attached to the primary data. Then, the processed TMI, TAF, METAR and IF data were merged to create a comprehensive GDP dataset. Lastly, each hour a GDP was in place was represented by a row in the integrated dataset, to represent the temporal evolution of weather forecasts and GDP plans. The R code for preparing the GDP evolution data is provided in Appendix A.

3.3.1 Data preprocessing

The data preprocessing work includes filtering, cleaning, and unifying the time zone. It was filtered such that the retained records were only those for TMI data with Advisory Category "GDP" (Column 5 in Table 3.1), Control Element "EWR/ZNY" (Column 7 in Table 3.1) and Advisory Date from January 1, 2010 to August 31, 2014 (Column 2 in Table 3.1). TAF data issued at EWR from January 1, 2010 to August 31, 2014 were retained (Columns 1, 2 and 3 in Table 3.2); and IF data with arrival airport EWR (Column 20 in Table 3.4), arrival date from January 1, 2010 to August 31, 2014 (Columns 5 and 6 in Table 3.4) were retained. The time horizon of January 1, 2010 to August 31, 2014 was chosen because the available TAF data was limited to this period.

After filtering, illogical data, such as TAFs or TMIs with abnormal (too long or negative) durations, and duplicate data (mainly an issue in the TAF dataset) were removed from the original datasets.

Finally, time zones were unified as the GDP and weather data is in Universal Time Coordinated (UTC) while the IF data is in Eastern Time (UTC -5:00 / -4:00) or New York City local time. All datasets were unified into New York City local time.

3.3.2 Generating additional variables

We produced additional variables calculated from other data in the raw datasets, for the purpose of matching the datasets or describing GDP features. In the TMI dataset, only the "planned" parameters of the GDP advisories are available. For example, the advisory end times (Column 16 in Table 3.1) do not typically match the "actual" end times of the advisory. The actual advisory end time information would be the advisory begin times (Column 15) of the subsequent revision advisory belonging to the same GDP (if there should be one).

All new variables calculated are presented in Column 2 of Table 3.6, along with descriptions.

Table 3.6 Summary of the new variables

Code	Name	Sources	Description	Computational process	Involved variables (column no.)
#1	Actual advisory end time	TMI	The actual end time of the advisories	If the advisory is not the last one of the initiative and the start time of the next advisory (column 15 in Table 3.1) is earlier than the planned end time of this advisory (16), the actual advisory end time should be the start time of the new advisory; otherwise, it equals to the planned end time of this advisory.	GDP.Bgn.Date.Time (Column 15 in Table 3.1); GDP.End.Date.Time (16)
#2	Actual initiative end time	TMI	The actual end time of the initiatives	If the GDP was cancelled (6), the actual initiative end time is the cancellation time (17); otherwise, it is the end time (16) of the last advisory of this GDP.	AdvisoryType (Column 6); GDPX.Bgn.Date.Time (17); GDP.End.Date.Time (16)
#3	Planned advisory duration	TMI	The planned duration of the advisories	The difference between the end time (16) and start time (15) of a GDP advisory.	GDP.Bgn.Date.Time (15); GDP.End.Date.Time (16)
#4	Actual advisory duration	TMI	The actual duration of the advisories	The difference between the actual end time (#1) and start time (15) of the GDP advisory	Actual advisory end time (#1); GDP.Bgn.Date.Time (15)
#5	Planned initiative duration	TMI	The planned duration of a GDP initiative after each advisory of it was issued	The difference between the end time (16) of the GDP advisory and the initial start time (15) of the GDP initiative	GDP.Bgn.Date.Time (15); GDP.End.Date.Time (16)
#6	Actual initiative duration	TMI	The final actual duration of the GDP initiatives	The difference between the actual initiative end time (#2) and the initial start time of the GDP initiative.	Actual initiative end time (#2); GDP.End.Date.Time (16)
#7	Number of modifications	TMI	Number of modifications/advisories of a GDP	The number of the GDP advisories belonging to the GDP (whether an advisory belongs to an initiative depends on the right variables)	RootAdvisoryNumber (9); Is.RootAdvisory (12); AdvisoryType (6)
#8	Affected airports (in TMI)	TMI, AI	The airports which would be affected by the advisory	Translate the “Departure scope” (14) of the advisory into airports using AI data; then add the Canadian airports affected by the advisory (13); delete the airports exempted in the advisory (21)	Canadian.Dep.Arpts.Included (13); Dep.Scope (14); Exempt.Dep.Facilities (21); AirportID (1 in Table 3.5); ARTCC (6 in Table 3.5)

Code	Name	Sou-rces	Description	Computational process	Involved variables (column no.)
#9	Early cancel time	TMI	For the cancelled GDPs, it describes how advance a GDP was ended by a cancellation than it was planned to be ended	The difference between the end time (18) of the last GDP advisory of the initiative and its cancellation time (17).	GDPX.Bgn.Date.Time (17); GDP.End.Date.Time (18)
#10	CW0422	TAF	The forecasted crosswind strength to Runways 4/22 of EWR airport	Wind Speed * |sin((Wind Angle – 40)*(pi/180))|	Wind.Angle (16 in Table 3.2); Wind.Speed (17)
#11	CW1129	TAF	The forecasted crosswind strength to Runway 11/29 of EWR airport	Wind Speed * |sin((Wind Angle – 110)*(pi/180))|	Wind.Angle (16); Wind.Speed (17)
#12	PC	TAF	Whether there would be any precipitation	For the following variables in TAF - RA(rain), DZ(drizzle), SN(snow), SG(snow grains), GR(hail), GS (snow pellets), IC (ice crystals), UP (unknown precipitation (auto)), if there is any of the variables equals to 1, PC = 1. (National Weather Service, 2017)	RA (20); DZ (21); SN (22); SG (23); GR (24); GS (25); IC (A26); UP (27)
#13	ARTCC (in IF)	AI, IF	The ARTCC of the airports arriving at EWR airport	Match the airport ID in IF and in AI and attach the ARTCC from AI to IF for the airport	AirportID (1 in Table 3.5); ARTCC (6 in Table 3.5); Dep_LOCID (19 in Table 3.4)
#14	Country (in IF)	AI, IF	The country category where the airport is located: U.S, Canada or International	Match the airport ID in IF and in AI and attach the Country from AI to IF for the airport	AirportID (1 in Table 3.5); Country (2 in Table 3.5); Dep_LOCID (19 in Table 3.4)
#15	Distance (in IF)	AI, IF	The distance (in miles) between the departure airport and EWR airport	Match the airport ID in IF and in AI and attach the Distance from AI to IF for the airport	AirportID (1 in Table 3.5); Distance (7 in Table 3.5); Dep_LOCID (19 in Table 3.4)

To capture crosswinds from the TAF and METARs data, this research calculated a crosswind variable based on wind speed and angle. Crosswinds pose serious safety risks to aircraft on runways by exerting a lateral force (Khurana, 2009). Additionally, Kuhn (2016) indicated that crosswind strength was one of the most important features for modelling the initiation of GDPs at EWR. Crosswinds can be calculated based on wind speed (Column 16 in Table 3.2 for TAFs; Column 3 in Table 3.3 for METARs), wind angle (Column 17 in Table 3.2 for TAFs; Column 4 in Table 3.3 for METARs) and runway direction using the following formula (Neufville & Odoni, 2003):

$$crosswind\ speed = wind\ speed \times sin(\alpha) \quad (2)$$

Where α is the angle of the wind from the aircraft travel direction on the runway (in radians).

The direction of runways 4L/22R and 4R/22L at EWR are 40 or 220 degrees, and runway 11/29 is 110 or 290 degrees. Thus, α will be the difference between the wind angle and 40 (or 220) degrees for Runway 4L/22R and 4R/22L, and the difference between the wind angle and 110 (or 290) degrees for Runway 11/29.

The weather variables RA (rain), DZ (drizzle), SN (snow), SG (snow grains), GR (hail), GS (snow pellets), IC (ice crystals) and UP (unknown precipitation (auto)) (Columns 20 – 27 in Table 3.2 for TAFs; Columns 7 -14 in Table 3.3 for METARs) were combined into an integrated variable – precipitation (National Weather Service, 2017).

For IF data, the information of country, ARTCC and distance from EWR of the departure airport from AI data was supplemented for each flight.

All new variables and how they were generated are described in Table 3.6.

3.3.3 Generating GDP evolution data

GDP plans are often revised as the weather forecasts change over time. Thus, GDP evolution is dependent on weather forecasts (from TAF dataset) and GDP parameters (from TMI dataset and IF dataset) over time.

3.3.3.1 Matching TAFs to GDPs (TMI dataset)

By merging the TMI and TAF datasets, the hourly weather forecast (i.e. visibility, ceiling, crosswind to Runways 4/22, crosswind to Runway 11/29, thunderstorm and precipitation) was attached to each GDP advisory for describing the forecast weather conditions of the advisories. The variables attached to the TMI data from TAFs are described in Table 3.7.

Regarding the forecast coverage time of the TAFs, overlapped forecast time periods existed among different TAF records, meaning that a GDP hour may have several corresponding TAF records. Only one TAF record was chosen for each GDP hour by the following steps:

a) For a GDP advisory, pick out the TAFs with: 1) an issued time earlier than its corresponding GDP's send time, 2) start time earlier than GDP end time, or 3) end time later than GDP start time.
b) For each hour of the GDP advisory, pick out the TAFs with start time earlier than the last minute of this hour, and TAF end time later than the first minute of this hour.
c) If for an hour, there are several TAF records, then the TAF with the latest issue time is chosen for matching to a GDP, as forecasts issued later have a higher likelihood of accuracy to real events.
d) Finally, attach the weather variables obtained above to each hour of each GDP.

Table 3.7 Variables attached to TMIs from TAF data

Code	Name	Sources	Description
#16	Visibility	TMI, TAF	The forecasted visibility condition (in miles) for each hour at EWR
#17	Ceiling	TMI, TAF	The forecasted ceiling condition (in 100 feet) for each hour at EWR
#18	Crosswind to Runways 4/22	TMI, TAF	The forecasted crosswind condition (in knots) for each hour for Runways 4/22 at EWR
#19	Crosswind to Runway 11/29	TMI, TAF	The forecasted crosswind condition (in knots) for each hour for Runway 11/29 at EWR
#20	Thunderstorm	TMI, TAF	The forecasted thunderstorm status (0 for none, 1 otherwise) for each hour at EWR
#21	Precipitation	TMI, TAF	The forecasted precipitation status (0 for none, 1 otherwise) for each hour at EWR

3.3.3.2 Matching individual flights (IF dataset) to GDPs (TMI dataset)

We then matched IF data to GDP data to assign GDPs the flights they impacted. To merge the TMI and IF datasets, this research matched by geography and time. The steps are detailed below:

a) Match GDPs and flights by time. Compare the flight base estimated time of arrival (scheduled gate-in time) (Column 33 in Table 3.4) with GDP times and pick out the flights whose flight arrival times fall within the GDPs times (Columns 15 and 16 in Table 3.1).

b) Match GDPs and flights by geography. Check flight originating Country (#14). For flights originating in the U.S. or Canada, do step c.

c) For U.S. and Canadian flights, check whether they were affected or exempted by the GDPs, through matching the ARTCC (#13) or Distance (#15) in IF dataset with the GDP departure scope (Column 14 in Table 3.1) or GDP affected Canadian departure

airports (Column 13 in Table 3.1) and GDP exempted departure airports (Column 21 in Table 3.1).

The detailed computational process for the variables from IF dataset and attached to TMI dataset were described in Table 3.8.

Table 3.8 Variables attached to TMIs from IF dataset

Code	Name	Sources	Description	Computational process	Involved variables
#22	Scheduled Arrivals	IF, TMI	Number of flights which were scheduled to arrive at EWR during a GDP plan before it was issued	Number of flights whose base estimated time/gate-in time (AG in IF) of arrival before it was affected by the GDP was within the GDP time horizon (O and P in TMI)	SCHINTM (AG in IF); GDP.Bgn.Date.Time (O in TMI); GDP.End.Date.Time (P in TMI)
#23	Impacted Arrivals	IF, TMI	Number of flights affected by the GDP (involved in GDP departure scope)	For the flights whose base estimated time of arrival (AG in IF) was within the GDP time horizon (O and P in TMI), if the flight was included in "AffectedAirports" (#8), then it would be affected by the GDP.	Variables in #22; Affected Airports (#8); DEP_LOCID (S in IF)
#24	Ground Delay	IF, TMI	Sum of the ground delay of all the flights affected by a GDP	Sum of the "Schedule-based Departure Delay" (AA in IF) of the flights affected by the GDP	DLASCHOFF (AA in IF); Variables in #24
#25	Actual Total Arrival Delay	IF, TMI	Sum of the actual total delay (ground delay and airborne delay) of all the flights affected by a GDP	Sum of the "Schedule-based Arrival Delay" (AJ in IF) of the flights affected by the GDP	DLASCHARR (AJ in IF)
#26	Planned Arrival Delay	IF, TMI	Sum of the planned arrival delay of all the flights affected by the GDP	Sum of the difference between Planned Arrival Time and Schedule Arrival Time of all the flights affected by the GDP. For computational convenience, it equals to the difference between Flight Plan based Arrival Delay (CC in IF) and Schedule-based Arrival Delay (AK in IF)	DLAFPARR (AK in IF); DLASCHARR (AJ in IF)

3.3.3.3 Descriptions of the merged dataset

Through the data processing work detailed above, a final GDP advisory dataset was established which matches GDP advisories to their corresponding weather forecasts and planned advisory parameters. The final dataset consists of 2,282 rows and 37 columns. Each row consists of a GDP advisory. The advisories were reordered according to their Root Advisory Number and Modification Number, unlike the original TMI data where advisories were ordered by their issue time. The dataset variables (i.e. columns) are shown in Table 3.9. The descriptions are not provided here since all variables in this dataset were introduced in previous sections (Tables 3.1-3.8).

Table 3.9 Variables in GDP Advisory Dataset

Column No.	Column name	Column No.	Column name
1	AdvisoryNumber	20	Planned advisory duration
2	SendDate.Time.UTC	21	Actual advisory duration
3	AdvisoryType	22	Planned initiative duration
4	RootAdvisoryNumber	23	Actual initiative duration
5	Derived.BgnDate.Time.UTC	24	Number of modifications
6	Derived.EndDate.Time.UTC	25	Affected airports
7	Is.RootAdvisory	26	Early cancel time
8	Canadian.Dep.Arpts.Included	27	Visibility
9	Dep.Scope	28	Ceiling
10	Eff.Bgn.Date.Time.UTC	29	Crosswind to Runways 4/22
11	Eff.End.Date.Time.UTC	30	Crosswind to Runway 11/29
12	GDP.Bgn.Date.Time.UTC	31	Thunderstorm
13	GDP.End.Date.Time.UTC	32	Precipitation
14	GDPX.Bgn.Date.Time.UTC	33	Scheduled Arrivals
15	GDPX.End.Date.Time.UTC	34	Impacted Arrivals
16	Impacting.Condition	35	Ground Delay
17	ProgramRate	36	Actual Total Arrival Delay
18	Actual advisory end time	37	Planned Arrival Delay
19	Actual initiative end time		

3.3.3.4 Generating Hourly GDP Evolution Dataset

Based on the master GDP advisory dataset I created, an Hourly GDP Dataset was built by creating a new file with each row representing an hour, and then organizing all the GDP information into this new time-based format.

Two steps have been performed to establish this dataset: first, group the GDP advisories with the same root advisory number (Column 9 in Table 3.1) into one GDP initiative; second, divide the GDP initiative active time into hours, and use the GDP parameters of the advisory whose active time matches with this hour, as the initiative parameters of this hour. Here, advisory/initiative active time is determined by the difference between actual advisory/initiative end time GDP and advisory/initiative begin time. At this point, an hourly GDP initiative dataset reflecting the evolving parameters of GDP initiatives has been created.

Thus, in the Hourly GDP Dataset, the unit (or each row) is one hour of a GDP initiative. Finally, 11,177 rows and 38 columns are included in the dataset. Asides from a new column containing the hour number, the remaining columns/variables of this dataset remain the same as the GDP advisory dataset of Section 3.3.3.3.

3.4 Descriptive statistics

This section presents descriptive statistics for the weather, GDP and flight data prepared as described previously in this chapter, in order to gain a basic knowledge of the weather features and GDP characteristics at EWR from 2010 through 2014. The data distribution, components (proportions), and trend over the time were summarized.

3.4.1 Weather characteristics at EWR

The data indicates that from 2010 through 2014, 89% of the EWR GDPs were initiated due to adverse weather, confirming that weather was the heavily dominating cause of GDPs over the five years in question.

Because crosswinds may pose safety risks by exerting a lateral force to aircraft on the runways (Khurana, 2009), a runway experiencing crosswinds greater than 15 knots are not used, according to FAA Airplane Flying Handbook (FAA, 2016).

The EWR capacity profile for EWR (FAA, 2014d) indicates that pilots are required to use instruments to fly aircraft (in a situation called Instrument Meteorological Conditions, IMC) when the ceiling or visibility is below 1000 feet or 3 miles, respectively. Thus, I defined "low ceilings", "low visibility" and created "IMC" variable according to this IMC criteria.

Figure 3.3 shows the total number of observations (hours) for the weather variables for each month of the year. The variables include strong crosswind to Runways 4/22 (StrCW0422), strong crosswind to Runways 11/29 (StrCW11/29), IMC, precipitation, and thunderstorms. Precipitation, StrCW0422, and IMC were found to be the most frequent weather phenomenon at EWR in 2010 to 2014, while StrCW1129 were observed with relatively lower frequency. Thunderstorms (TS) occurred rarely except in summer. Precipitation was observed more from December - March and May, and less in June through November; StrCW0422 occurred most frequently from December to April, and least frequently from May to October; IMC appeared most often from December - February and May, and least often in July and August. In general, the frequencies of all three severe weather variables were low in summer and fall, and high in spring and winter. However, thunderstorms were more prevalent at EWR in the summer months, which

is in keeping with general knowledge about thunderstorms across the eastern states (Kim & Hansen, 2013).

Overall, precipitation appears to be the most commonly occurring adverse weather condition at EWR from 2010-2014, followed by crosswinds to runways 4/22, and low ceiling and visibility causing IMC. Weather conditions in December to May were generally worse than in other months.

Although the total number of hours of thunderstorm was very low, thunderstorms will typically lead to highly restrictive GDPs and significant arrival delays at EWR (Allan, Beesley, Evans, & Gaddy, 2001).

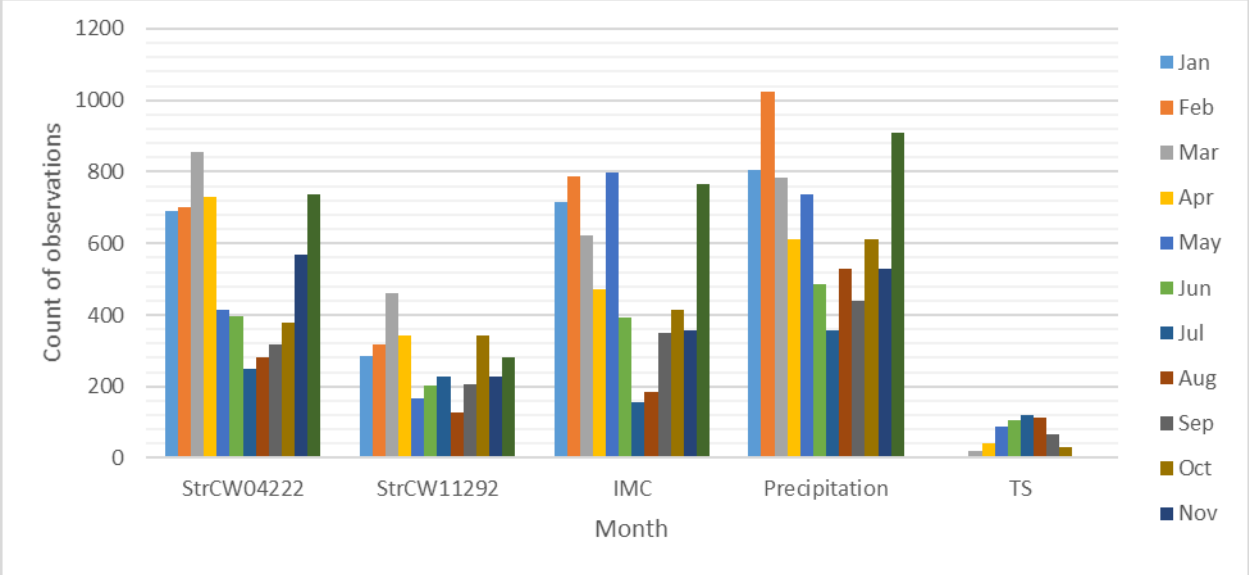


Figure 3.3 Count of observations of different weather variables in different months

Figure 3.4 shows the total hours where the different adverse weather conditions were in place over the five years from 2010-2014. Weather in 2010 and 2011 was comparatively worse than subsequent years – 2010 experienced more crosswinds to Runways 4/22 than other years; precipitation and IMC was most prevalent in 2011. Weather conditions from 2012 to 2014 appear

to be more consistent compared with the two previous years. These observations are consistent with reports from NOAA (U.S. Global Change Research Program, 2014).

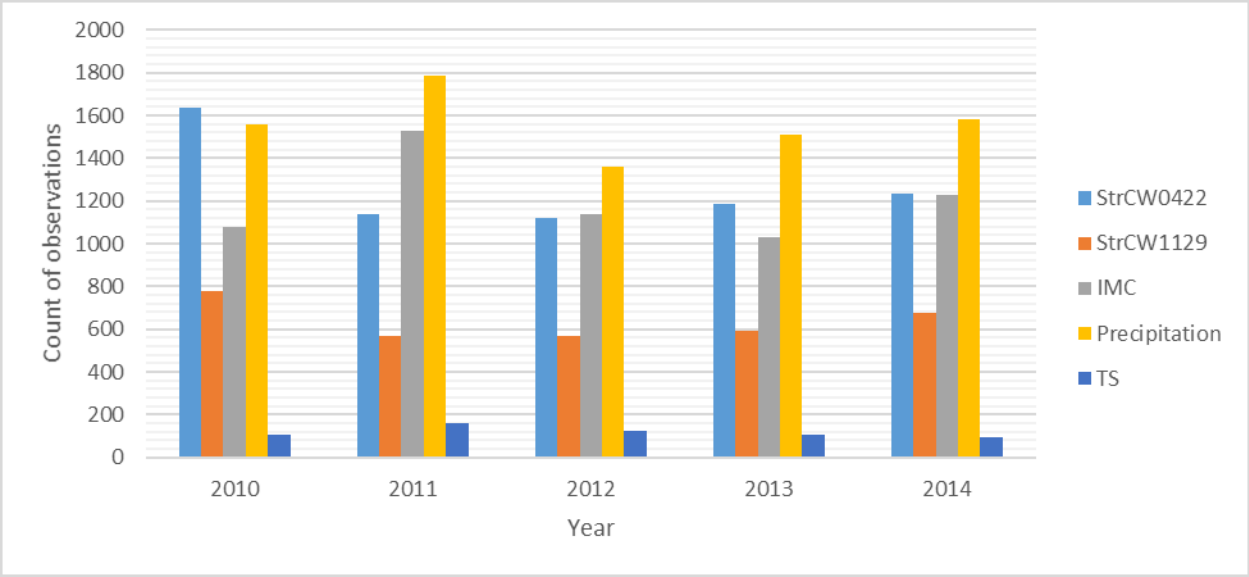


Figure 3.4 Weather constitutions of each year

Overall, among the five years, 2010 and 2011 appear to have been the worst years for weather while 2012 to 2014 were relatively better. The main adverse weather condition in 2010 were strong crosswinds to Runways 0422 and precipitation, while Instrument Meteorology Condition and Precipitation were the main conditions in 2011.

3.4.2 GDP characteristics at EWR

3.4.2.1 GDP causes

While adverse weather is cited as the most common reason for GDP issuance, it is usually a combination of weather and heavy flight demands (Manley & Sherry, 2008). This section discusses the major causes of EWR GDPs based on the five years of GDP, weather, and flight data.

As introduced in 3.2.1, the TMI dataset contains the causes of GDP advisories. GDP causes from January 1, 2010 through December 31, 2014 are shown in Figure 3.5. The figure demonstrates that weather was the major factor (89%) initiating GDPs, while runway-taxi problems (such as construction and maintenance), other conditions (such as security and emergency events) and flight demand prompted 8%, 2% and 1% GDPs respectively. However, the extremely low percentage of demand seems unreasonable because EWR is a busy airport with heavy flight demand as introduced in 3.1. The reason may be that 89% GDPs were usually initiated due to a combination of heavy demand and severe weather, and rarely just because of demand alone.

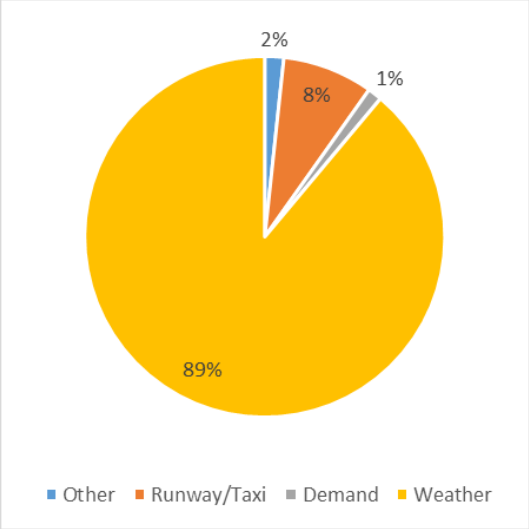


Figure 3.5 Causes of EWR GDPs

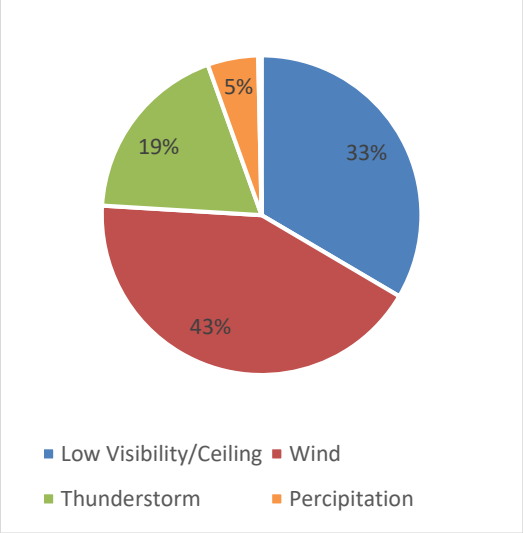


Figure 3.6 Weather causes of EWR GDPs

For the GDPs prompted by weather, strong winds and low ceiling or visibility were the most common factors for those GDPs. Thunderstorms and precipitation accounted only for 19% and 5% respectively of those GDPs, as shown in Figure 3.6. Comparing with the EWR weather characteristics discussed in 3.4.1, it can be observed that although precipitation was the most frequently occurring adverse weather phenomena at EWR, it was not as impactful for GDP initiation. Strong crosswind to Runways 4/22 and IMC (low ceiling/visibility) most frequently

caused GDP initiation, in line with the findings of Wang and Kulkarni (2011) and Grabbe, Sridhar and Mukherjee (2013).

Figure 3.7 shows the components of the GDP causes in different months. Though Figure 3.3 and Figure 3.6 illustrated that thunderstorms were the least frequent weather condition and relatively insignificant GDP contributors, they had a large impact on the summer GDPs issued (June to August) as shown in Figure 3.7. GDPs prompted by precipitations mainly existed in winter. October to March, GDPs were mainly issued due to winds. GDPs in April, May and September were issued due to multiple factors including winds, thunderstorms, low ceilings and runway/taxi problems with roughly even percentages. There were more winter (December to March) GDPs prompted by precipitation by comparing with other months. Besides the cause component information, this figure also reveals a difference in number of GDPs in different months. A figure clearly showing the number of GDP initiatives in each month is presented as Figure 3.8.

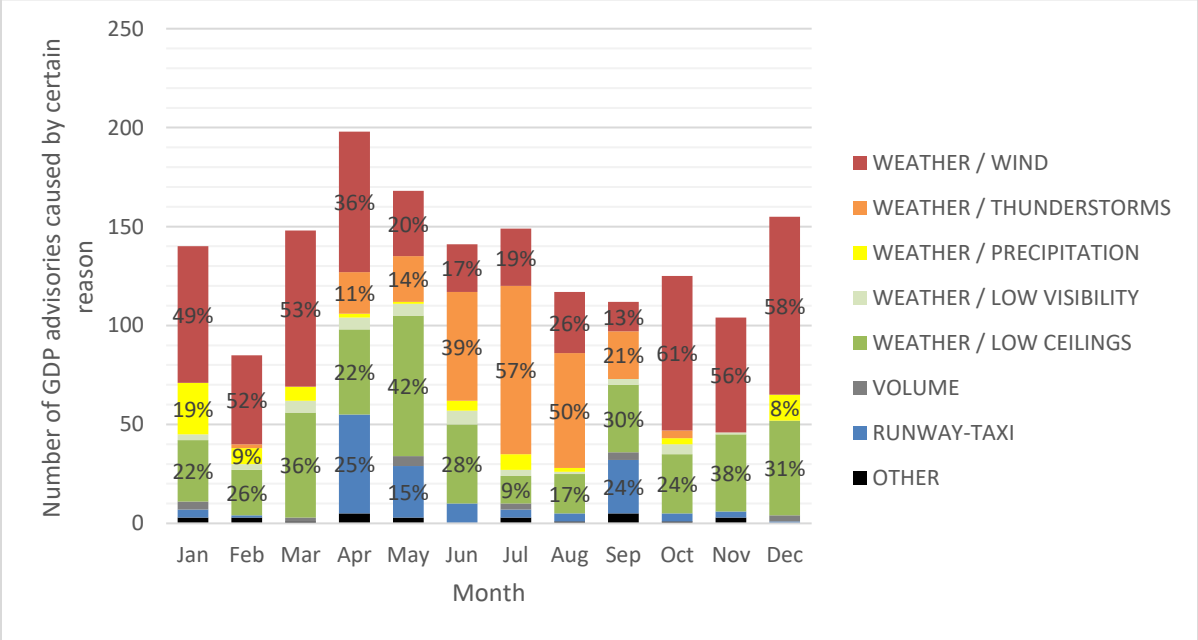


Figure 3.7 EWR GDP causes in different months

As shown in Figure 3.8, GDPs were most frequently observed in April and May, followed by December, July, June and March. GDPs were least frequently observed in February, followed by September-November. However, in Figure 3.3 it was observed that crosswinds, precipitation and IMC occurred very frequently in February which suggests that GDPs at EWR were often prompted by factors other than weather alone.

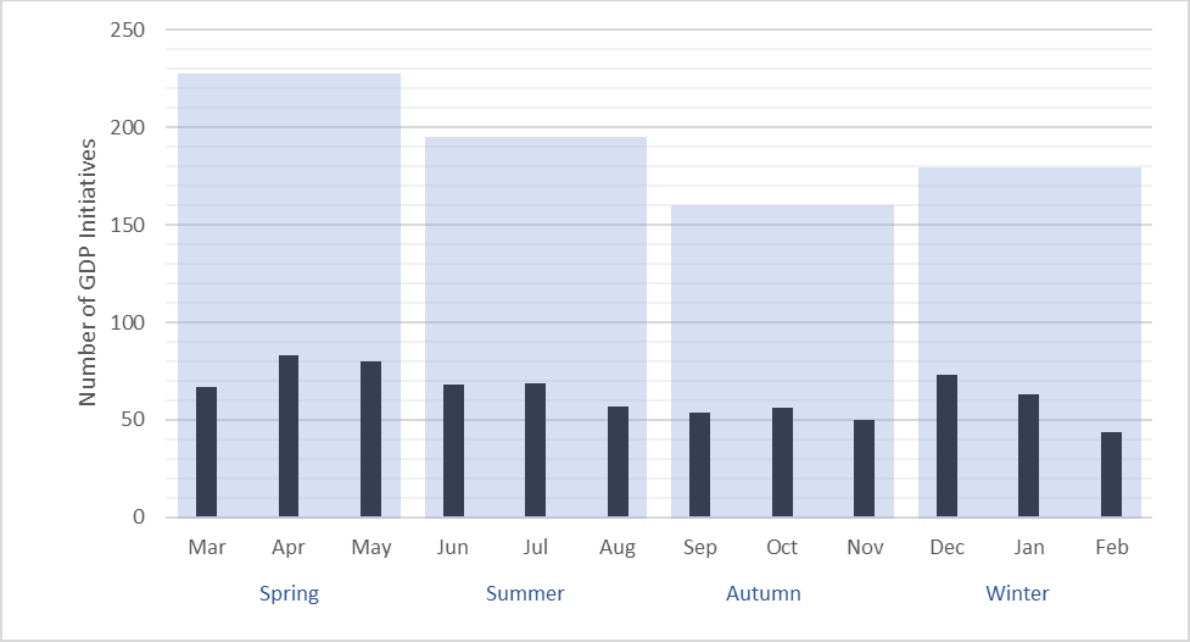


Figure 3.8 Number of GDP initiatives per month

The individual flights (IF) dataset contains scheduled flight information. It must, however, be noted that the ASPM data (including IF data) does not include cancelled flights, meaning that not all flights originally scheduled to arrive or depart at EWR were included in the dataset. Although less than ideal given that flight cancellations due to GDP can be extensive, for the purpose of this study, this research used the number of scheduled flights in IF as an approximation of the "true" runway demand. Figure 3.9 shows the total flight demand as per the IF dataset at EWR over the months of the year. It was observed that flight demand was lower in February,

which provides an explanation of why GDPs were issued with less frequency that month despite the greater frequency of adverse weather.

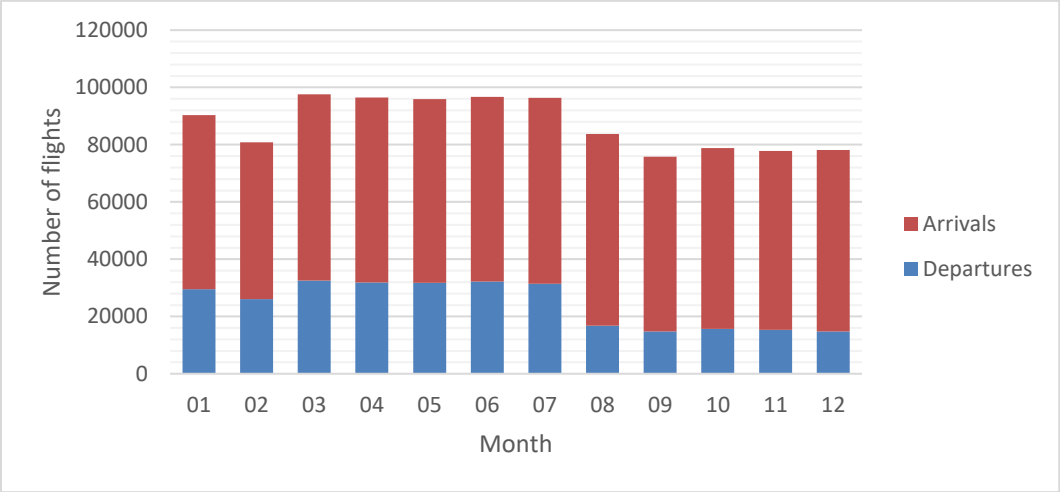


Figure 3.9 Flight demand at EWR in each month

To further examine the impact of flight demand on GDPs at EWR, Figure 3.10 which shows the flight demand per day of the week at EWR from 2010-2014. As expected, there are more scheduled flights during the weekdays than weekends (with Saturday having the lowest flight demand). By comparing with Figure 3.11 which presents the number of GDP initiatives implemented in the days of the week, it was observed that same trend appears in Figure 3.11– fewer GDPs were implemented on weekends, particularly Saturday. The impact of flight demand on the EWR GDPs was verified again through the comparison.

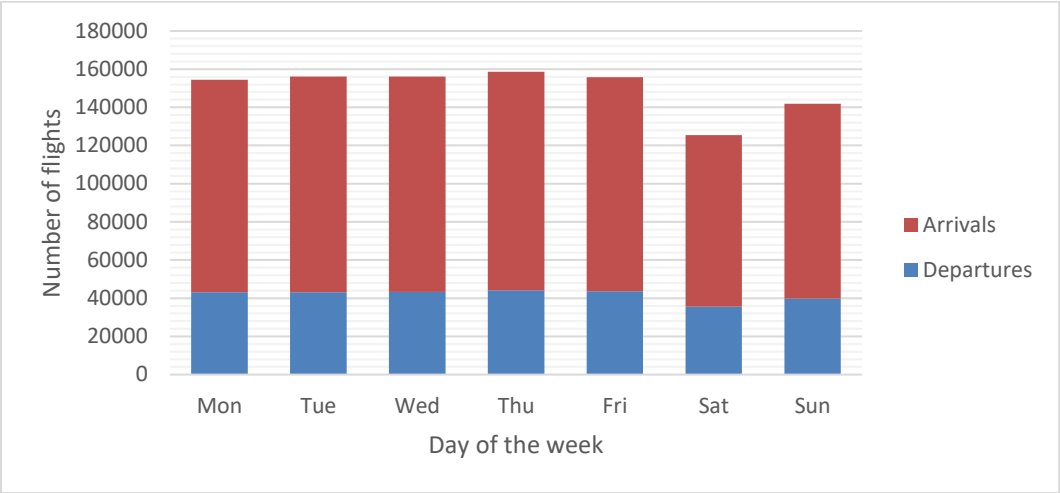


Figure 3.10 Number of scheduled flights at EWR per day of week over the 5 years

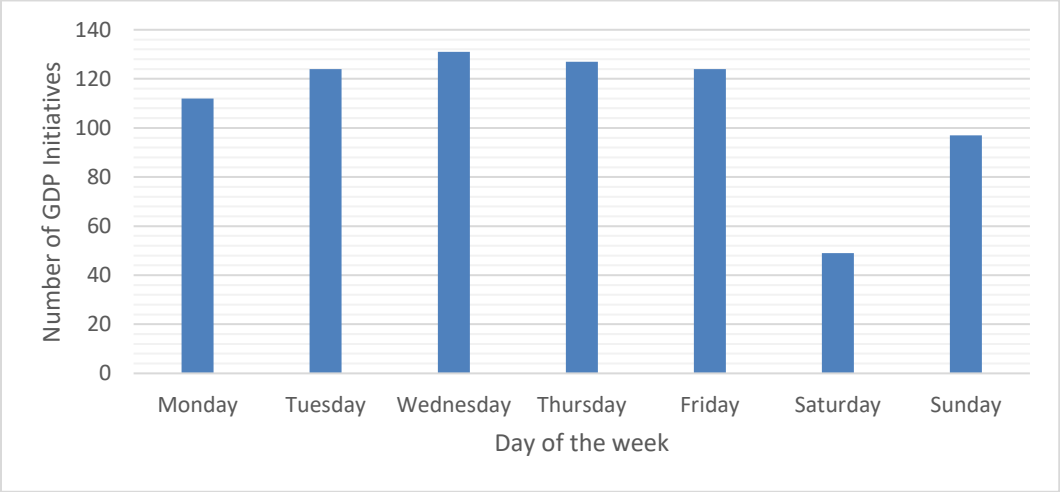


Figure 3.11 EWR GDP in per day of week over the 5 years

As introduced in 2.1, the purpose of a GDP is to mitigate the imbalance between flight demand and capacity at an airport. The capacity-demand imbalance can be caused by capacity reduction due to inclement weather or excessive demand due to heavy flight schedules. Although the TMI data demonstrated that weather was the major cause of GDPs at EWR, we know that GDPs would not be as prevalent at less busy airports with lower flight demands than EWR.

3.4.2.2 GDP revisions and cancellations

For the EWR GDPs implemented from 2010 through 2014, each GDP plan had on average 1.16 revisions (usually an extension). 95% of GDPs were cancelled (usually early cancellation) with an average early cancel time of 1.9 hours, meaning that GDPs at EWR ended almost 2 hours earlier than planned. This seems to suggest that air traffic controllers were either conservative in their GDP planning, TAF forecasts are conservative, or both.

3.4.2.3 GDP times and durations

There are three critical time variables for GDP implementation – GDP send time, start time and end time. This section provides a statistical description regarding the times of the EWR GDPs issued from 2010 through 2014. All the GDP times stated in this section are in New York City local time.

Figure 3.12 presents box plots of EWR GDP send times, begin times, revise times and end times by hour of day. Send, begin, and end times each occur once for a GDP initiative, while revisions can occur multiple times. It can be observed that half the send times (in the center, 25th to 75th percentile) were between 9 am and 12 pm, began times were between 11 am and 1 pm, revised between 2 and 6 pm; and ended between 7 and 10 pm.

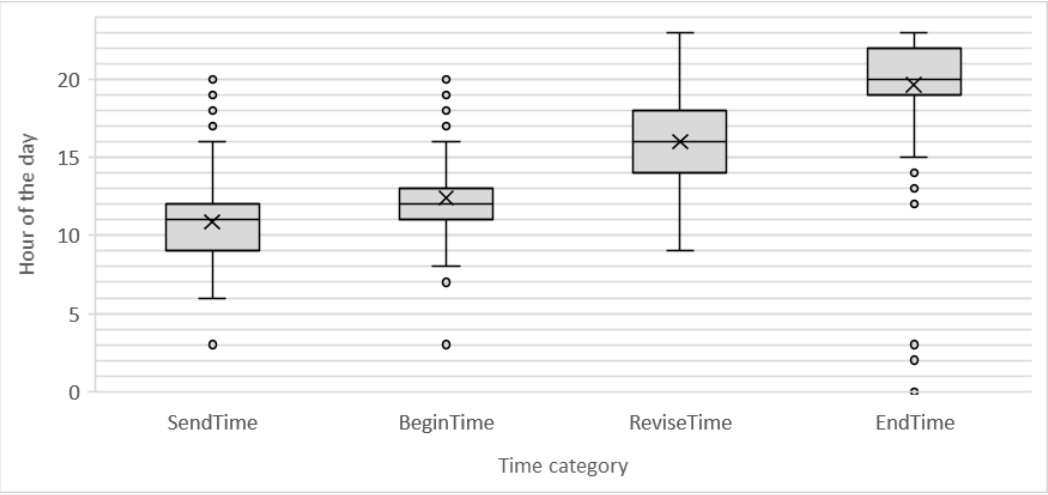


Figure 3.12 Send times, begin times, revise times and end times of the GDPs over a day

The planned duration of a GDP is the difference between its start time and planned end time. The actual duration of a GDP is the difference between its start time and actual end time (sometimes it is "cancel time"). Figure 3.13 shows descriptive statistics on GDP initial planned duration, planned duration after revisions and actual duration. By comparing the third boxplot with other boxplots in Figure 3.13, it can be found that the actual GDP duration was generally shorter than their initial and revised planned length. Specifically, most GDPs were initially planned to run for about 8 to 12 hours. Then after being revised, the GDPs were generally extended, reaching a planned duration of 10 to 13 hours. Finally, the GDPs were generally shorter than their planned length, lasting for 6 to 11 hours.

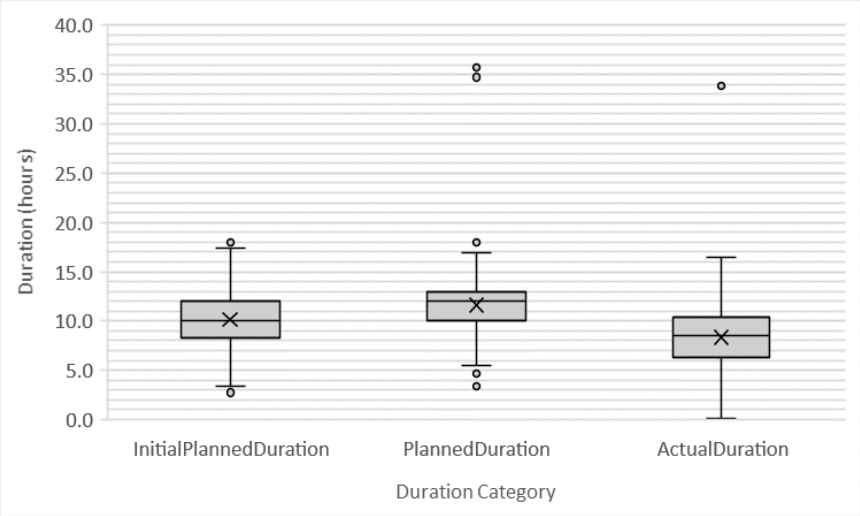


Figure 3.13 EWR GDP durations

In summary, EWR GDPs issued in the five years were sent in the late morning, activated around noon, modified in the afternoon, and ended in late evening or early night. The GDPs were often revised to extend, but ended earlier than even the planned duration.

3.5 Summary

This chapter provided an overview of the data used in this study. Datasets on GDPs, weather forecasts, and individual flight information at EWR from 2010 through 2014 were combined to generate a master dataset for the GDP evolution analysis in next chapter. This master dataset contains the weather forecasts and GDP parameters for each hour when a GDP was in place over the five analysis years.

Descriptive statistics were generated to gain a basic understanding of the EWR weather and GDP data over the five analysis years. The main characteristics of the actual weather at EWR include that: 1) precipitation, crosswinds to runways 4/22, and low ceiling or visibility were the most frequently observed adverse phenomenon; 2) December to May generally experienced more inclement weather than in other months, and 3) weather from 2010 through 2011 was relatively worse than in other years. GDPs generally were in place for 8-10 hours and on average ended two hours earlier than planned. GDPs were found to be more frequently initiated in spring and on weekdays (due to heavier flight demand), while crosswinds were found to be the most common cause of the GDPs. To further explore the characteristics of the EWR GDPs, data mining techniques will be applied to GDP evolution data in next chapter.

Chapter 4. GDP evolution characteristics exploration

This chapter introduces the techniques, analysis procedures and results of GDP characteristics exploration using data mining. Section 4.1 provides a brief introduction to the dimensionality reduction techniques, clustering methods, and statistical methods used in the study. Section 4.2 describes the process of GDP evolution scenario establishment and scenario-performance correlation assessment. Through the analysis, 10 GDP scenarios were finally identified for describing how GDPs evolve under changing weather forecasts. Significant correlations between scenario parameters and GDP performances were observed.

4.1 Techniques

This section introduces the techniques used in our exploration of GDP evolution. Autoencoder, an unsupervised learning method, was used to extract some of the basic patterns in the high-dimensional GDP data (i.e. with many features changing over time) with the goal of dimensionality reduction. Three clustering methods - k-means, PAM, and hierarchical clustering - along with three k-estimation methods - average silhouette and gap statistic - were employed to develop GDP scenarios. A statistical method called Configural Frequency Analysis was utilized to examine correlations between different GDP parameters.

4.1.1 Autoencoders

As introduced in Section 3.3.3.3, the GDP evolution dataset developed for this study contains weather forecasts and GDP parameters by hour, and thus involves multiple dimensions. Autoencoder, which can capture important features of high-dimensional data (such as the prepared

GDP dataset) automatically, was applied with the purpose of dimensionality reduction. Dimensionality reduction is the process of reducing the number of variables in the data set by selecting a subset of the original data (feature selection) or transforming the data to a lower-dimensional space (feature extraction). The transformation could be linear (such as in Principle Components Analysis) or nonlinear. As linear methods could be restrictive, autoencoder, a more automatic approach without the linearity assumption, was used. There have been some studies using autoencoders for the characterization of high-dimensional and time-varying data (such as that of the GDP dataset). For example, Shin et al. (2011) applied autoencoders to automatically classify tissue types according to the change in brightness of resonance images.

An autoencoder is an artificial neural network which learns the features of inputs by backpropagation algorithm - reconstructing the input data in output layer and minimizing reconstruction errors (Hinton & Salakhutdinov, 2006). An autoencoder includes an input layer, one or more hidden layers, and an output layer whose number of nodes is the same as the one of input layer. The structure of an autoencoder can be divided into two parts – encoder and decoder, as shown in Figure 4.1. In the encoder part (from input layer to middle hidden layer), the autoencoder learns representation for a data set (encoding), while it is trained to optimize a loss function which measures how well the data is reconstructed based on the encoder representation in the decoder part (from the middle hidden layer to output layer). Thus, an autoencoder is a special type of neural net where the input is also used as the target; in other words, the output is an optimized reconstruction of the input.

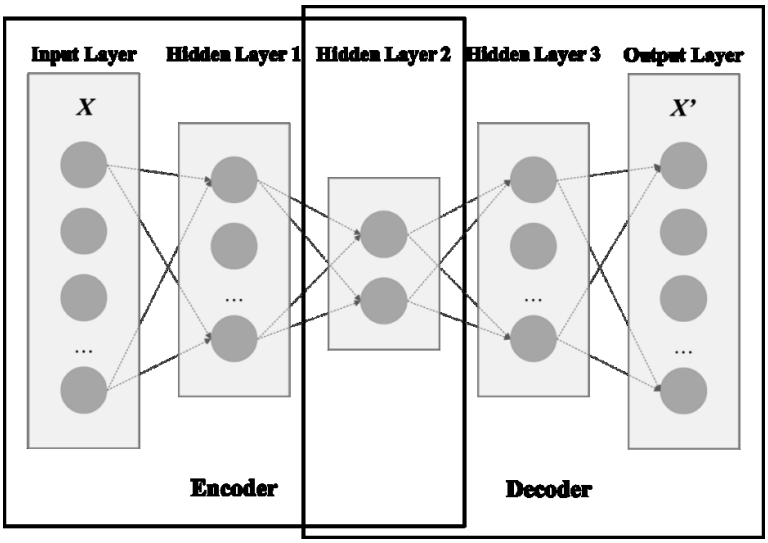


Figure 4.1 An autoencoder with 3 hidden layers

In the encoder, the structure can be represented using the following:

$$\boldsymbol{h} = f(\boldsymbol{x}) = \sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{3}$$

Where:

$\boldsymbol{x}$ is the input;

$\boldsymbol{h}$ is the representations of input in lower dimensions;

σ is an element-wise activation function;

$\boldsymbol{W}$ is a weight matrix;

and $\boldsymbol{b}$ is a bias vector.

For example, in Figure 4.1, Hidden Layer 1 (with fewer neurons) is the lower-dimensional representation of the data in Input Layer (with more neurons). Thus, the nodes in Hidden Layer 1 is $\boldsymbol{h}$ and data in the Input Layer is $\boldsymbol{x}$. Each node in the Hidden Layer 1 is determined through a weighted sum ($\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$) and a nonlinear transformation (σ, a non-linear activation function) of the nodes in Input Layer 1. The other encoder layers can be derived in the same manner. In this study, the Tanh activation function was used to relax the linear transformation assumption made

in other methods like PCA. The function has been widely used (Sarle, 2002). The Tanh function is shown as below:

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (4)$$

In Decoder, the structure can be represented as follows:

$$\boldsymbol{x}' = g(\boldsymbol{h}) = \sigma(\boldsymbol{W}'\boldsymbol{h} + \boldsymbol{b}') \quad (5)$$

Where:

$\boldsymbol{x}' \approx \boldsymbol{x}$,

$\sigma, \boldsymbol{W}'$ and $\boldsymbol{b}'$ are the activation function (Tanh function), weight matrix and bias vector for decoder.

The constraint of the neural network is to minimize the loss function - Mean Square Error (MSE):

$$\text{Min } L(\boldsymbol{x}, \boldsymbol{x}') = \left|\left|\boldsymbol{x}' - \boldsymbol{x}\right|\right|^2 \quad (6)$$

Where:

$L(\boldsymbol{x}, \boldsymbol{x}')$ is the cost function reflecting the discrepancy between the input and output, namely reconstruction errors.

This research built a 5-layer autoencoder net to compress the high-dimensional and time-varying data was compressed into two dimensions. The detailed process is introduced in Section 4.2.2.

4.1.2 Cluster methods

Clustering is a common technique for exploratory data mining, which groups data objects with the goal that objects within the same group are similar to each other and different from objects in other groups (Tan, Steinbach, & Kumar, 2006). Since the structure of the GDP evolution data

was unknown, I applied three different clustering methods in order to split the data into groups of similar objects (GDPs), and two approaches to determine the optimal number of clusters. Then, the results of these methods were compared.

4.1.2.1 Clustering methods

Clustering can be divided into partitioning methods (such as k-means and PAM) and hierarchical clustering. This section introduces the k-means, PAM and hierarchical clustering used in this study. Clustering was performed using R (with all codes provided in Appendix B).

(1) k-means clustering

k-means is one of the most popular clustering methods for unsupervised data learning. The aim of this method is to classify a given set of data points into a certain number of clusters, in which each data point belongs to the cluster with the closest mean. Ultimately, k centroids are defined, one for each cluster. The detail steps of the k-means method are shown as below (MacQueen, 1967).

- Place k points into the space of the dataset to be clustered and define these points as the initial group centroids.
- Assign each data points to the cluster group that has the closest centroid.
- When all data points have been assigned, recalculate the positions of the k centroids as the center of the clusters generated from the previous step.
- Repeat Steps 2 and 3 until the centroids no longer move. This produces a partition of the data points into clusters, in which the centroid is the closest to binding data points.

(2) PAM clustering

Partition Around Medoids (PAM) is another frequently used clustering method for data mining, also known as the k-medoids clustering method. Similar to k-means, PAM also aims to

minimize the distance between data points grouped to belong to a cluster and a point defined as the center of that cluster. In contrast to k-means, PAM chooses data points (observations) as centers (defined as medoids in this method) and uses Manhattan Norm instead of Euclidean norm to calculate the distance or similarity between data points (Kaufman & Rousseeuw, Clustering by means of medoids, 1987). The equation of the Manhattan Norm is:

$$\left|\left|\mathrm{x}\right|\right| = \sum_1^n x_i \tag{7}$$

(3) Hierarchical clustering

Hierarchical clustering builds a hierarchy of clusters. The number of clusters to be generated does not need to be specified a-priori, and uses pairwise distance matrix between observations as clustering criteria. The result is a tree-based representation of the observations. There are two types of hierarchical clustering - agglomerative and divisive (Kaufman & Rousseeuw, Finding groups in data: an introduction to cluster analysis, 2009). Agglomerative clustering is a bottom-up algorithm, in which each object is initially considered as a single-element cluster (leaf). Then the two clusters that are the most similar are combined into a new bigger cluster (nodes), until all points are included in one single big cluster (root). Divisive clustering is a top-down algorithm, in which all objects are included in a single cluster. Then the most heterogeneous cluster is divided into two until all objects are in their own cluster.

4.1.2.2 Determining the optimal number of clusters

This section briefly describes three methods for determining the optimal number of clusters, including a direct method (average silhouette) and a statistical testing method (gap statistic).

The direct methods, which optimize a criterion about how well the clusters summarize the data, used in this study is average silhouette method. The criterion to be optimized is the average silhouette width. To choose the optimal number of clusters (k), we usually look for k with the

maximum silhouette width (Kaufman & Rousseeuw, Finding groups in data: an introduction to cluster analysis, 2009)

The statistical testing methods used in this study is the gap statistic, which can be applied to any clustering method. The optimal k is the one with maximum gap statistic. The gap statistic is the deviation of the total within intra-cluster variation of the real data from the expected value of the reference data with a uniform distribution (Tibshirani, Walther, & Hastie, 2001).

4.1.3 Configural Frequency Analysis

Configural Frequency Analysis (CFA) is a widely used multivariate data analysis method which is parameter-free and can be can be applied to any data set regardless of its statistical distribution. It identifies the configurations which occur statistically more or less than expected by chance. For example, we classify GDPs into several types, and count how frequently they occur for each day of the week. The combination of a particular GDP type and day of the week is a configuration, and CFA would detect whether the observed frequency of this configuration is greater or less than expected by random occurrence. The formula for calculating the expected frequency is shown in $Ei,j= N * p_{i.} * p_{.j}$ (8.

A configuration occurring significantly more than expected will be identified as a "type," while a configuration occurring significantly less than expected will be identified as an "antitype." (Eye, Spiel, & Wood, 1996).

We firstly calculate the expected frequencies E for each configuration by assuming that no relationships exist between the two categories (e.g. GDP types and days of the week). The formulas are shown as below:

$$E_{i,j} = N * p_{i.} * p_{.j} \quad (8)$$

Where:

E_{ij} is the expected frequency of a configuration;

N is the total numbers of observations of all configurations;

$p_{i.} = O_i/N\,;\;\; p_{.j} = O_j/N$

O_{ij} is the observed frequencies of the configuration;

Then the observed frequencies and expected frequencies are compared. As long as the expected frequency of most configurations is $E_{i,j} \geq 10$, the Z-test can be applied to test the significance of the difference between observed and expected frequencies (Eye, Spiel, & Wood, 1996). The binomial statistic Z is calculated as below:

$$Z_{ij} = \frac{O_{ij} - E_{ij}}{\sqrt{E_{ij}(1 - p_{ij})}} \tag{9}$$

Where:

$$p_{ij} = E_{ij}/N \tag{10}$$

Finally, we identify the types and antitypes based on the p-value calculated from the Z statistic. In this study, type/antitypes were detected based on a confidence level of 90% and were then used to interpret the relationship between GDP category and other categories.

4.2 GDP characteristics exploration

This section presents an exploratory analysis for GDP evolution characterization, including the process and results of the data visualization, dimensionality reduction, cluster analysis and statistical analysis.

4.2.1 Data visualization

As described in 0, the merged dataset contains the weather forecast data and GDP plan data as different columns, with each row representing an hour number for each GDP initiative. Here,

this research further broke down the hourly data in order to describe the GDP evolution more precisely, as GDP parameters can change within a single hour. Visualization of GDP data can help explain the abstract data transforming process, as well as help evaluate the clustering results which are introduced in 4.2.3.

We designed a form of images to visualize the GDP evolution process under changing weather over three key elements – times, GDP/weather parameter categories, and parameter value magnitudes. As shown in Figure 4.2, the GDP images are two-dimensional and in greyscale, in which the x-axis represents time, y-axis represents parameters, and the grey scale represents the parameters' values. As such, the images can be used to describe the varying GDP parameters planned under varying forecast weather conditions.

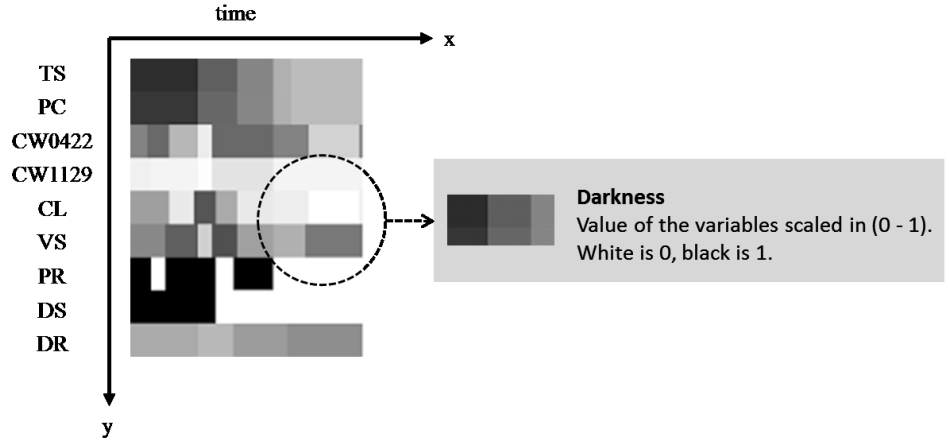


Figure 4.2 A greyscale image for GDP visualization

Figure 4.2 shows nine variables on the y-axis, including six weather variables and three GDP parameters. The six weather variables include the forecast weather variables from the TAF: 1. TS (thunderstorm), 2. PC (precipitation), 3. CW0422 (crosswind strength to Runways 4/22), 4. CW1129 (crosswind strength to Runway 11/29), 5. CL (ceiling), and 6. VS (visibility). The three GDP variables are the basic GDP planning parameters and include: 1. PR (program rate), 2. DS (departure scope, represented by the number of affected flights of a GDP advisory), and 3. DR

(planned initiative duration). The three GDP parameters describe the impact of the GDPs. Specifically, PR is the maximum number of aircraft the GDP will allow to arrive at the airport per hour (i.e. GDP arrival capacity); DS is the number of delayed flights during the GDP; and DR is the effective duration of the GDP advisories.

The maximum time shown on the x-axis represents the actual end time of a GDP, and there are total of 65 intervals on the axis. The design reasons and process are explained as follows. Different GDPs have different length. If I make the maximum on x-axis all the same for all GDPs, besides the distribution of the greyscale blocks, there will be another feature appearing on the image – the size of the colored space (or the blank space). It was found that, by using this type of design, GDPs were overwhelmingly characterized by their durations, rather than other evolution patterns of interest. Although the actual GDP length is an important feature of GDPs, this research is more interested in describing under what situations GDPs were planned and revised. Thus, I removed the "size" (i.e. real time) component from the GDP images by normalizing all GDP durations. The next step is to set up the intervals on time axis for GDPs. A GDP parameter may change due to its hourly setting, for example, "program rate" of a GDP advisory is set up in an hourly format. Besides, a GDP parameter may also change due to a revision. In this way, the time difference between a GDP parameter issuance and modification can be less than one hour. In our dataset, the parameter modification times varied between one and sixty minutes. If we describe GDP evolution by minute the dataset size will be too large; if we describe GDP evolution by hour, we lose details. Balancing dataset size and detail level, this research divided each GDP into 65 intervals (66 moments) such that all GDPs are described by intervals no longer than 15 minutes. All variables are scaled to [0,1] and their values are represented by the grey scale. The darker the block is, the higher the variable value is. Thus, a greyscale GDP image has 9 * 65 = 585 pixels.

The images describe GDP evolution by presenting at what moment, under what condition, what kind advisory was planned.

4.2.2 Dimensionality reduction

As introduced above, a GDP is represented by a 9 x 65 matrix (585 cells), and this was transformed into vectors 585 cells long. Then this research applied the autoencoder technique on the processed data using the H2O platform from R. H2O is a fast and scalable open-source machine learning platform, with interfaces to Java, Python, R and Scala.

The autoencoder technique requires the design of a neural network by setting up the number of neurons in each layer and the number of hidden layers for the autoencoder. According to the definition of autoencoders explained in 4.1.1, the number of neurons in the input and output layers is the same. Each GDP has 585 cells; thus, there are 585 neurons in both the input and output layers. There are no rules for choosing the number of hidden layers and the number of neurons in hidden layers. For most problems, one hidden layer is sufficient (Panchal, Ganatra, Kosta, & Panchal, 2011). this research compared the effects of the autoencoders with one to three hidden layers by viewing the clustering results. It was found that using an autoencoder with three hidden layers, the GDP were more well-classified ("well-classified" refers to that GDP images within the same group are very similar to each other and distinctive to the images in other groups) than the results of autoencoders with one or two hidden layers. I set the number of neurons in the middle hidden layer to two, so that the original data can be represented in a two-dimensional space. Finally, I designed an autoencoder with one input layer, three hidden layers and one output layer, with 585, 300, 2, 200, and 585 neurons respectively. The structure of the autoencoder this research used is shown as Figure 4.3.

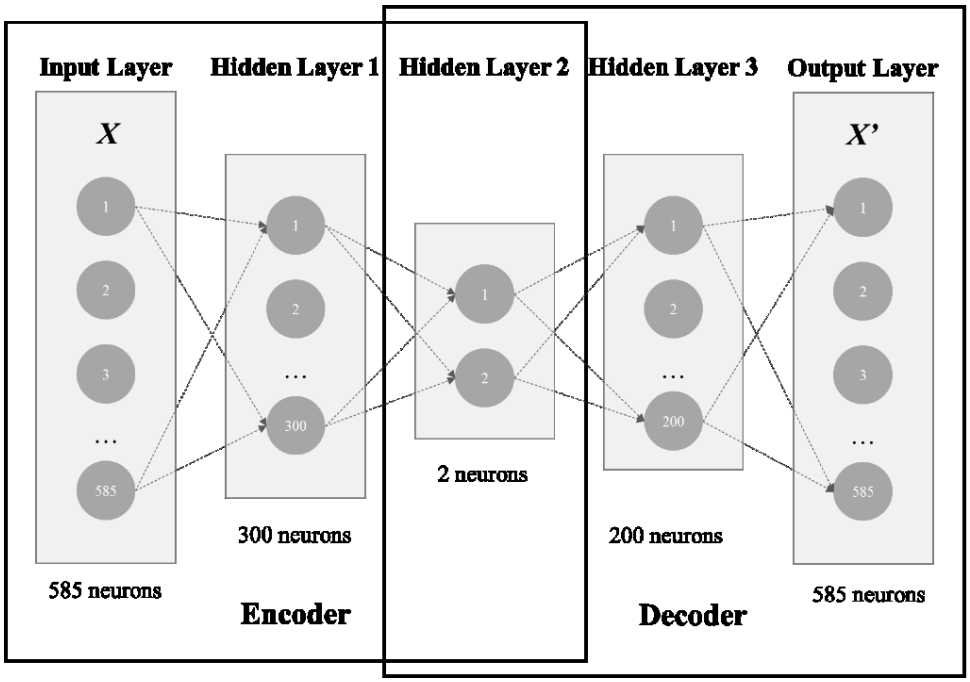


Figure 4.3 Autoencoder structure in this study

The original 585-dimensional data was compressed to two dimensions in the second hidden layer. There are some missing time periods in the TAF data, which results in discontinuous weather forecasts for some GDP records from the TMI database. Considering that incomplete data will lead to untrue GDP images, I removed those GDPs with missing TAF data; in the end, 495 GDP initiatives are retained. The GDPs were then plotted as shown in Figure 4.4.

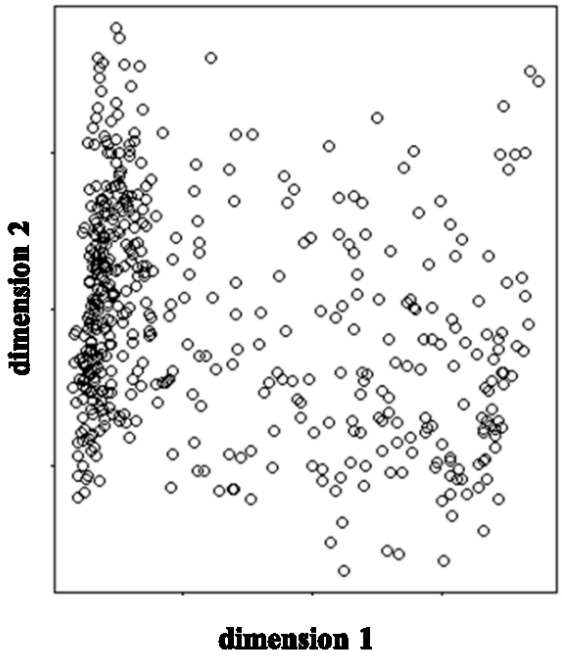


Figure 4.4 GDPs in 2-dimensional space

In Figure 4.4, the dots in the plot represents the 495 GDPs. The GDPs are represented by two dimensions, which were determined by the autoencoder as nonlinear combinations of the 585 original variables of that GDP.

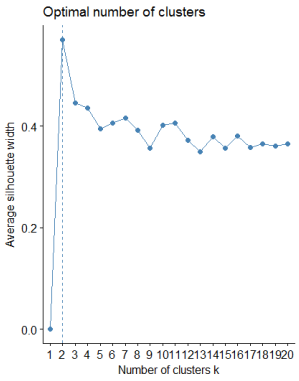
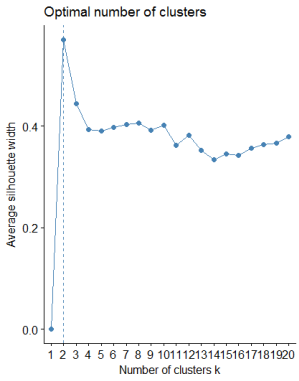
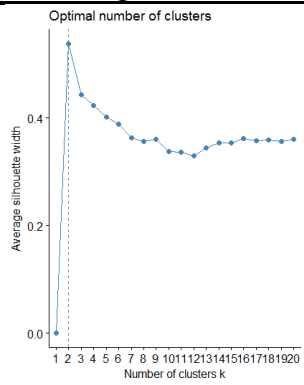
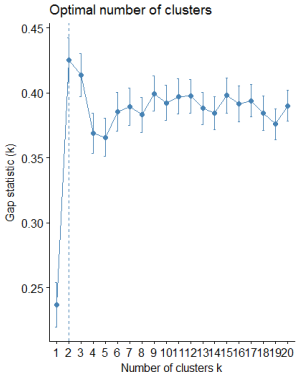
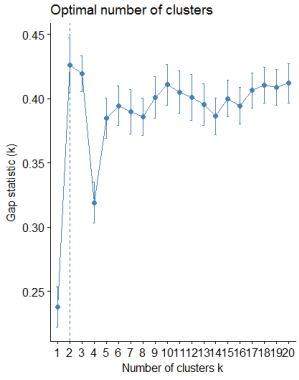
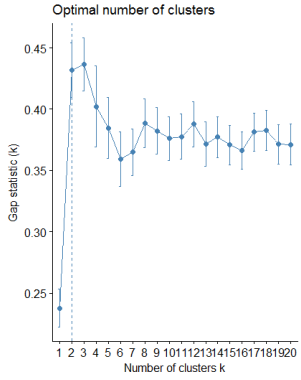
4.2.3 Cluster analysis

Three clustering methods – k-means, PAM and hierarchical clustering, were applied to the 2-dimensional GDP data obtained after dimensionality reduction, and the k-estimation methods – average silhouette and gap statistic methods – were used to determine the optimal number of clusters.

The clustering results are summarized in Table 4.1. The columns represent the clustering methods and the rows are the k-estimation methods. Each cell contains a k-estimation plot for the corresponding clustering method. The plots are labeled from (a) to (i). All possible values of k between 2 and 20 were considered. To choose the optimal number of clusters, as per 4.1.2, the average silhouette method and gap statistic returns k values that maximize the average silhouette

width or gap statistic. Here, that value is two in plots (d) to (h) and three in plot (i). But it is unlikely that there are only two or three types of days with respect to air traffic flow management in the New York area (Kuhn, Shah, Skeels, & Murra, 2016). The classified GDP images when $k \leq 5$ indicate that GDPs within the same group may include images with notably different patterns. Thus, as introduced in 4.1.2, this research decided upon an optimal k for each clustering method by choosing the k (larger than 5) with highest average silhouette width and gap statistic. They are listed below the clustering figures as shown in Table 4.1. Three figures (a, b, e) suggest that 10 may be a good candidate for k, two figures suggest that 8 (b, f) or 9 (c, d) is optimal.

Table 4.1 Cluster analysis results

	k-means	PAM	Hierarchical clustering
Average silhouette	 (a) $k = 7, 10, 11$	 (b) $k = 8, 10$	 (c) $k = 6, 9$
Gap statistic	 (d) $k = 9$	 (e) $k = 10$	 (f) $k = 8, 12$

Then, this research conducted a graphical exploration to determine whether k lies somewhere between $8 - 12$. this research examined the clustering results using GDP greyscale images, and comparing the similarity of images within the same group as well as the difference of images in different groups. It was found that, for $k = 8$ or 9, some cluster appeared to hold very different images and thus were candidates for further division into more groups. With $k = 11$ or 12, some different clusters were similar and can be merged into one group. Finally, with the PAM clustering method and $k = 10$, the greyscale images were such that GDPs within a group were quite similar while those in different groups are more distinguished. The GDPs clustered into 10 groups by PAM are shown in Figure 4.5.

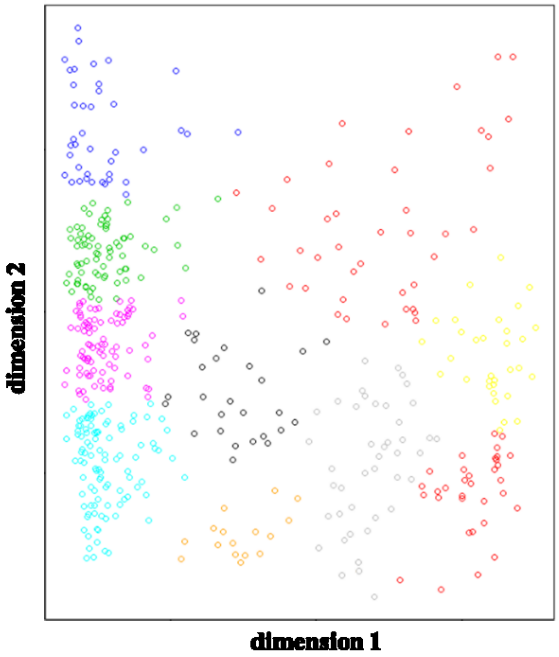


Figure 4.5 GDPs clustered by PAM with $k = 10$

Sample greyscale GDP images of each cluster are provided in Figure 4.6. It can be seen that GDPs were clustered well with $k = 10$, reflected by their different color distribution patterns both horizontally and vertically. The "color distribution patterns" refers to two points. First, how dark are the colors of the cells? Second, how consistent are the colors of the cells? For example,

GDP images in Cluster 3 look “bright on top, black in middle, grey in bottom”, while GDP images in Cluster 8 look “partially black on top, light grey in middle, dark in bottom”; images in Cluster 3 look very “neat” while images in Cluster 8 look very “messy”.

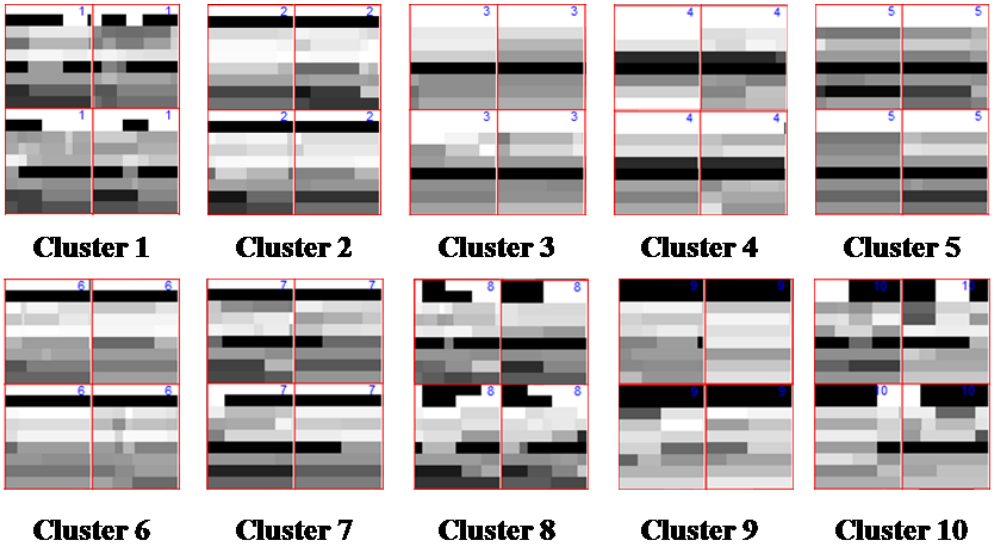


Figure 4.6 Greyscale images of GDPs in each cluster

Figure 4.6 shows that the clusters may have severe/relatively good and even/unstable forecasted weather conditions, and GDPs with high/low values. For example, in Cluster 3, forecasted weather (first 6 rows) and GDP plans (last 3 rows) appear very stable across time (x-axis), and there was little thunderstorm action (first row), precipitation (second row) and crosswinds (third and four rows); ceiling (fifth row) was medium low (grey colored), and visibility was very high (dark). Also, GDP program rate, departure scope and planned duration (last three rows) were in medium level (grey colored). this research further examined the characteristics shown in the clustered GDP images - firstly, this research calculated the average of each variable for all the clusters to examine the level of adverse weather severity and the level of GDP impact for each cluster. Secondly, this research examined the variability (or stability) of the forecasted weather conditions for all clusters, by calculating the variance of each weather variable in a GDP and then averaging the variance of each weather variable for the GDPs in same cluster.

Table 4.2 shows the average of each variable of these clusters which demonstrates the level of adverse weather severity and the level of GDP impact. The cells are colored in gradient red according to their relative magnitudes within the column, indicating how severe the forecasted weather condition was (based on columns TS, PC, CW0422, CW1129, CL, VS) and how impactful the GDP plan was (based on columns PR, DS, DR) - dark red means that the cell represents high weather variable impact (impactful GDP or severe weather) and light red corresponds to low impact by the weather variable in question. The averages of the variables in each cluster were found to be consistent with their features (darkness) shown in Figure 4.6. Thus, this research finally described the weather severity and GDP impacts according to the visual features and averages.

Table 4.2 Average of weather and GDP variables of each cluster[4]

Cluster No.	TS	PC	CW 0422	CW 1129	CL	VS	PR	DS	DR
1	0.00	0.56	17.01	9.43	17.01	8.01	33.06	175.51	12.33
2	0.06	0.98	7.31	9.43	7.31	4.24	33.47	179.75	12.51
3	0.00	0.00	12.41	8.02	12.41	9.81	37.56	116.92	7.89
4	0.00	0.00	9.41	6.87	9.41	9.83	37.26	71.03	5.31
5	0.00	0.00	20.62	9.75	20.62	9.95	35.78	156.93	10.80
6	0.14	0.94	5.94	10.51	5.94	4.07	31.99	121.53	10.34
7	0.09	0.82	9.36	8.33	9.36	7.53	34.74	153.07	11.31
8	0.12	0.31	9.08	9.45	9.08	8.47	34.42	149.41	10.99
9	0.48	0.88	5.47	8.47	5.47	4.37	32.20	64.48	5.57
10	0.35	0.60	7.48	7.61	7.48	7.76	34.01	113.11	8.97

[4] Explanation of the variables' abbreviated names:
- TS: thunderstorm in TAF. 1 represents existence of thunderstorm, and 0 represents the absence of thunderstorm.
- PC: precipitation in TAF. 1 represents existence of precipitation, and 0 represents the absence of precipitation.
- CW0422: crosswind strength to Runways 4/22 in TAF (in knots).
- CW1129: crosswind strength to Runway 11/29 in TAF (in knots).
- CL: ceiling in TAF (in 100 feet).
- VS: visibility in TAF (in miles).
- PR: planned program rate of GDP advisories (in number of aircraft).
- DS: planned departure scope of GDP advisories, represented by the number of affected flights.
- DR: planned initiative duration (in hours).

Table 4.3 shows the variance of the forecasted weather conditions for the clusters. Cells with dark red represents that weather forecasts varied significantly over time. The variances were found to be consistent with the variation features shown in Figure 4.6. Information about the clusters is summarized in Table 4.4, along with the average analysis results.

Table 4.3 Variance of weather variables of each cluster

Cluster No.	TS	PC	CW 0422	CW 1129	CL	VS
1	0.00	0.17	0.47	0.05	0.00	0.51
2	0.03	0.00	0.01	0.00	0.00	0.03
3	0.00	0.00	0.01	0.00	0.01	0.00
4	0.00	0.00	0.01	0.00	0.00	0.00
5	0.00	0.00	0.01	0.00	0.01	0.00
6	0.05	0.02	0.20	0.00	0.00	0.12
7	0.03	0.08	0.18	0.00	0.04	0.60
8	0.07	0.18	0.01	0.02	0.09	0.43
9	0.02	0.02	0.19	0.00	0.00	0.09
10	0.13	0.19	0.23	0.03	0.12	0.77

Table 4.4 Cluster descriptions

Group No.	Main weather types	Weather severity	Weather stability	GDP Type	Number of GDPs
1	PC, CW	Severe	Unstable	Low, Wide, Long	23
2	PC, LVC	Severe	Medium	Low, Wide, Long	36
3	LVC, CW	Less severe	Stable	High, Medium, Short	151
4	LVC, CW	Less severe	Completely Stable	High, Narrow, Short	39
5	CW	Less severe	Stable	High, Wide, Medium	110
6	PC, LVC	Severe	Medium	Low, Medium, Medium	37
7	PC, LVC	Severe	Unstable	Medium, Wide, Long	34
8	LVC, PC	Severe	Unstable	Medium, Wide, Long	46
9	TS, PC, LVC	Very severe	Medium	Low, Narrow, Short	10
10	TS, PC, LVC	Very severe	Unstable	Medium, Medium, Short	26

By considering both the absolute magnitude of each of the six weather variables for each cluster, as well as their relative magnitudes compared with other clusters, this research described the weather by indicating the main weather types (e.g. TC, PC, etc.), weather level (less severe, severe or very severe), and weather stability (unstable, medium, stable, or completely stable). For example, Cluster 3 was defined as "less severe" weather because the images and averages shows that only some crosswinds (12 knots on average) were forecasted for the GDPs in the cluster and the cluster has only 3 red-colored weather cells, ranking as the best one. Cluster 10 was defined as "unstable" weather because its images and variance shows that most of the weather variables' variances are high and the cluster has the most red-colored cells (relative magnitude). Similarly, for the GDP variables, since there are no references or standards for identifying a GDP parameter's level to be low or high, this research simply described a GDPs by indicating program rate level (relatively low, medium, or high), departure scope (relatively wide, medium or narrow) and advisory duration (relatively long, medium or short) according to their relative magnitudes as well as considering their absolute magnitudes.

The quantitative bounds for the classifications used to describe weather and GDPs are introduced as follows:

(1) Weather severity

- Less severe: only strong crosswinds (bigger than 15 knots) or low ceiling (less than 1000 feet) forecasted.
- Severe: precipitation plus strong crosswinds, low ceiling or low visibility (less than 4 miles) forecasted.
- Very severe: thunderstorms forecasted.

(2) Weather stability

- Stable: no weather variables expected to change significantly over time. (The variance obtained in this research is not applicable to general conditions, thus this may require subjective judgement.)
- Medium: only one weather variable expected to change significantly over time.
- Unstable: two or more weather variables expected to change significantly over time.

(3) GDP program rate

- Low: hourly program rate less than 34
- Medium: hourly program rate between 34 – 35
- High: hourly program rate bigger than 35

(4) GDP departure scope

- Narrow: the number of impacted scheduled flights (excludes canceled flights) less than 100
- Medium: the number of impacted scheduled flights (excludes canceled flights) bigger than 100 and less than 130
- Wide: the number of impacted scheduled flights (excludes canceled flights) more than 130

(5) GDP planned duration

- Short: planned duration less than 9 hours
- Medium: planned duration between 9 – 11 hours
- Long: planned duration more than 11 hours

The cluster descriptions are listed as follows according to Table 4.4:

- Group 1 represents low-wide-short GDPs under severe and unstable precipitation and windy weather.

- Group 2 represents low-wide-short GDPs under severe and medium-stable low visibility/ceiling weather.
- Group 3 represents high-medium-short GDPs under less severe and stable windy low visibility/ceiling weather.
- Group 4 represents high-narrow-short GDPs under less severe and completely stable windy low visibility/ceiling weather.
- Group 5 represents high-wide-medium GDPs under less severe and stable windy weather.
- Group 6 represents low-medium-medium GDPs under severe and medium-stable precipitation, thunderstorm and low visibility/ceiling weather.
- Group 7 represents medium-wide-long GDPs under unstable and severe precipitation, thunderstorm and low visibility/ceiling weather.
- Group 8 represents medium-wide-long GDPs under unstable and severe low visibility/ceiling, precipitation and thunderstorm weather.
- Group 9 represents low-narrow-short GDPs under very severe and medium-stable thunderstorm, precipitation and low visibility/ceiling weather.
- Group 10 represents medium-medium-short GDPs under very severe and unstable thunderstorm, precipitation and low visibility/ceiling weather.

We found three major categories for the clusters' forecasted weather conditions – 1) less severe and stable weather, where severe low visibility/ceiling ("LVC" in second column in Table 4.4) and strong crosswinds (CW) were the main weather types; 2) Severe and unstable weather, where precipitation (PC) was the main weather type, and CW or LVC occurred together; 3) Very

severe and unstable weather, where thunderstorms (TS) was the main weather conditions, PC and LVC also occurred together.

The first category, including Clusters 3, 4, and 5, had the highest number of observations. GDPs under this forecasted weather combination were all planned with high program rate ("high" in the fifth column in Table 4.4)), medium to short duration ("medium" / "short"). Most had medium to wide departure scopes (Cluster 3 and 5) while a few had narrow departure scope (Cluster 4).

The second category, including Clusters 1, 2, 6, 7 and 8, was the second most frequently occurring group. All the GDPs in this category had medium to low program rates, medium to wide departure scopes, and medium to long planned durations.

The third category, which includes Clusters 9 and 10, occurred with the least frequency. GDPs in this category had medium to low program rates, medium to narrow departure scopes, and short planned durations.

4.2.4 Statistical analysis

This section explores the relationships between the GDP scenarios (clusters) and their performance. The performance of the GDPs in each cluster were evaluated using the efficiency, capacity utilization and predictability metrics proposed by Liu and Hansen (2013). Other features, such as early cancelation time, number of revisions, days of the week, and time of the day were also included.

A series of Configural Frequency Analysis (CFA) tests were conducted to assess the correlations between GDP clusters and each performance metric. I set the clusters as rows and the levels of GDP performance (e.g. high/medium/low capacity utilization) as columns. The number of observations of each configuration was the cell value. For example, the capacity utilization of

the GDPs ranges from 0.10 to 1.09. Since there is no definite reference for identifying capacity utilization level to be low or high, to simplify, I divided this range into 3 equal sections – [0.10, 0.43), [0.43, 0.76) and [0.76, 1.09], and described them as relatively "low", "medium" and "high" capacity utilization. In Table 4.5, the 2nd-4th columns show the count of observations in the low, medium, and high groups, respectively; for instance, among the 23 GDPs in Cluster 1, 13 of them have "low capacity utilization." According to the CFA principles introduced in 4.1.3, the null hypothesis is that there is no correlation between the GDP cluster and the capacity utilization levels. Under this null hypothesis, the expected cell frequencies of each configuration can be calculated and compared with the observed frequencies. Finally, CFA detected that Cluster 4 was significantly related to high capacity utilization while Cluster 6 was significantly related to low capacity utilization.

Table 4.5 Configural Frequency Analysis Example

Cluster	Low capacity utilization	Medium capacity utilization	High capacity utilization	Total
1	13	1	9	23
2	14	2	20	36
3	51	6	94	151
4	9	1	**29 (type)**	39
5	53	0	57	110
6	**24 (type)**	0	13	37
7	17	1	16	34
8	18	0	11	29
9	4	1	5	10
10	15	0	11	26
Total	218	12	265	495

The CFA tests were conducted separately for different GDP performance metrics. I reordered the clusters according to their weather patterns, to allow for easier visual identification. Table 4.6 shows the CFA results. Table 4.7 shows the averages of the performance metrics for

each cluster, supplementing Table 4.6. The two tables contain many results that could be discussed; however, in the interest of space, I will discuss two sets of results of particular interest.

The first set of observations are from clusters 1-3 (highlighted in light grey and bordered in dark grey). The weather forecast in those clusters is less severe and stable, such that initiation of GDPs in this group may be attributed more to high demands or other factors rather than severe weather. For GDPs with high program rates, medium durations, and medium-wide scopes (cluster 1), their efficiency metric is higher than expected according to the CFA results. This suggests that GDPs with large scope (i.e. larger geographic scope and therefore, more impacted flights) may lead to higher-than-expected efficiency (ratio of GDP-induced departure over arrival delay). This could be interpreted by the fact that, despite a wide scope, stable weather conditions lead to more stable GDPs. When these GDPs have high program rates, medium durations, and narrow scopes, they would have higher-than-expected capacity utilization (as per CFA results). This result could be attributed to that these high program rate GDPs with narrower scopes involving less flights, leading to fewer cancellations and more arrivals (albeit delayed), and therefore, higher capacity utilization.

The second set of observations pertains to the results for clusters 6-8 (highlighted in darker grey). The weather of clusters 6-8 was forecasted to be severe and unstable (i.e. rapidly changing). When those GDPs have low program rate, wide departure scope and long duration occurs, their efficiency metric values were found lower than expected. This may be attributed to unstable weather conditions and a wide scope leading to a more volatile and rapidly changing GDP, which will lead to further delays in the air, and therefore, a lower efficiency score. When those GDPs have low program rate, low-medium departure scope and low-medium duration occurs, their capacity utilization is lower than expected. With higher duration, the capacity utilization is as

expected. This suggests that program rates are set more conservatively than needed for some poor weather conditions that end earlier than expected, with GDP being canceled early as well.

However, it is notable that although 90% of the GDPs were, according to the TMI dataset, caused by adverse weather, GDPs attributed to other causes (e.g. runway construction) were also included in the data. These GDPs (less than 10% of all GDPs) are likely to have been included in cluster 1-3, which have “less severe” and “stable” forecasted weather. Thus, the results found in the first set could attributed to factors other than those discussed above, and warrant further targeted analysis.

Table 4.6 CFA results

Original Label	Cluster No.	Weather forecast	GDP parameters	Efficiency	Capacity Utilization	Predictability	Early cancel time (hrs)	Number of revisions
5	1	Less severe, stable	High, wide, med	High	-*	-	≥2	-
4	2	Less severe, stable	High, narrow, short	-	High	-	-	0
3	3	Less severe, stable	High, med, short	High	-	-	-	0
1	4	Severe, unstable	Low, wide, long	-	-	-	-	≥2
8	5	Severe, unstable	Low, wide, long	-	-	-	-	≥2
7	6	Severe, unstable	Low, wide, long	Low	-	High	-	-
2	7	Severe, medium	Low, wide, long	Low	-	-	-	≥2
6	8	Severe, medium	Low, med, med	-	Low	-	-	≥2
10	9	Very severe, unstable	Low, med, short	-	-	-	-	-
9	10	Very severe, unstable	Low, narrow, short	-	-	-	-	0~1

Table 4.7 Simple statistics analysis results

Original Label	Cluster No.	Weather severity	Weather stability	GDP Type	Efficiency	Capacity Utilization	Predictability	Early cancel time (hrs)	Number of revisions
5	1	Less severe, stable	Stable	High, wide, med	1.03	0.55	0.50	2.20	1.15
4	2	Less severe, stable	Completely stable	High, narrow, short	1.02	0.74	0.34	1.88	0.31
3	3	Less severe, stable	Stable	High, med, short	1.05	0.64	0.44	1.94	0.64
1	4	Severe, unstable	Unstable	Low, wide, long	1.00	0.46	0.54	1.48	1.70
8	5	Severe, unstable	Unstable	Low, wide, long	0.99	0.43	0.48	1.79	1.66
7	6	Severe, unstable	Unstable	Low, wide, long	0.97	0.51	0.54	1.35	1.09
2	7	Severe, medium	Medium	Low, wide, long	0.95	0.58	0.52	1.92	1.50
6	8	Severe, medium	Medium	Low, med, med	0.93	0.41	0.45	1.65	1.57
10	9	Very severe, unstable	Unstable	Low, med, short	0.99	0.46	0.53	1.60	1.38
9	10	Very severe, unstable	Medium	Low, narrow, short	0.91	0.62	0.43	2.08	0.30

It was found that different revision decisions may involve a trade-off between predictability and efficiency. Clusters 5-8 have similar forecasted weather (severe and unstable with precipitation and low visibility/ceiling); by comparing these clusters, a trade-off was found to exist between high (2 or more) and low number of modifications – fewer revisions were associated with higher predictability but lower efficiency.

These results suggest the joint impact of GDP plans and weather forecasts on GDP efficiency. When weather is predicted to be less severe, for GDPs with wide departure scope, their efficiency would be higher than expected, while when weather is predicted to be severe and unstable over time, it would lead to lower-than-expected efficiency. It may be interpreted that, under less severe and stable forecasted weather conditions, GDPs with wider departure scope would lead to higher efficiency because they can absorb the airborne delays almost completely on ground by delaying numerous flights at their departure airport instead of en route; under long-term severe and unstable weather, less of flights' airborne delays may be transferred to the ground, due to the uncertainties induced by the long-term unstable conditions.

4.3 Summary

In this chapter, I explored how GDPs evolved over their lifetimes, taking into account weather forecasts, GDP plans, and GDP performance.

First, I visualized the GDPs using greyscale images in aid of examining the clustering results. The 585-dimensional GDP data was compressed into a 2-dimensional space using the autoencoder. Then clustering analysis was used to characterize evolving GDP plans under changing weather forecasts. I compared the results of k-means, PAM and hierarchical clustering using average silhouette width and gap statistic k-estimating methods, to describe the changing

weather forecasts and changing GDPs at EWR. Finally, Configural Frequency Analysis was performed to examine the correlations between specific GDP scenarios and GDP performances.

GDPs were clustered into 10 distinct scenarios according to weather type, severity, and stability over time, in addition to GDP duration, scope, and program rate. The most common GDPs issued in less severe and stable weather were those with high program rates, medium-to-wide departure scopes and short-to-medium durations. The most common GDPs issued in severe and unstable weather were those with medium-to-low program rates, medium-to-wide scopes, and medium-to-long durations. The results of the Configural Frequency Analysis (CFA) suggest that GDPs under stable, low-severity weather and with large scope (i.e. more impacted flights) may score higher on the efficiency metric than we would expect. This could be attributed to the fact that, despite a wide scope, stable weather conditions lead to more stable GDPs. When these GDPs have high program rates, medium durations, and narrow scopes, it was found that capacity utilization is higher than expected – less flights lead to fewer cancellations and more arrivals (albeit delayed), and therefore, higher capacity utilization. The results also suggest that program rates are set more conservatively than ultimately needed for some poor weather conditions that end earlier than expected, with GDP being canceled early as well. GDPs with fewer revisions were associated with a higher predictability score but lower efficiency score.

Chapter 5. Conclusions

This chapter provides a summary of the key findings and major conclusions of this research, presents the contributions, limitations of the research, and recommended future work.

5.1 Overview

This research explored the characteristics of evolving GDP plans under changing weather conditions. Based on the GDP data, weather data and flight data at EWR from 2010 through 2014, this research applied data mining tools to summarize the characteristics of the EWR GDPs as they evolved over their lifetimes. The work included:

1) Development of a dataset including GDP parameters and weather variables through the merging of forecast weather data (TAF), GDP data (TMI) and flight data (IF).

2) Visualization of high-dimensional and time-varying GDP evolution data, to support clustering results.

3) Feature extraction for the GDP evolution data by applying a deep neural network technique named autoencoder, compressing 585 dimensions into two.

4) Identification of the GDP evolution scenarios through cluster analysis based on compressed 2-dimensional data, with the purpose of characterizing GDP evolution under changing weather forecasts. Clustering methods (k-means, PAM and hierarchical clustering) and k-estimation methods (average silhouette and gap statistic) were employed. The final clusters were determined through comparing the results of these clustering methods.

5) Assessment of the correlations between the GDP evolution scenarios and GDP performances through multivariate statistical analysis, for further exploring GDP characteristics.

The Configural Frequency Analysis was applied to interpret whether certain GDP scenarios and certain GDP performances were significantly correlated.

5.2 Findings

The data confirmed that EWR GDPs from 2010 to 2014 1) were typically issued in late morning, began around noon, experienced modifications in afternoon, and ended in late evening or at early night; 2) involved one revision and one cancellation, on average; 3) typically lasted between 8 and 10 hours; 4) on average ended two hours earlier than planned; 5) were most frequently initiated due to various inclement weather, including crosswinds to runways 4/22, and 6) were more typically initiated in the spring months and on weekdays (the latter due to heavier flight schedules).

Based on the compressed 2-dimensinal data, the GDPs were clustered into 10 distinct scenarios as per weather type, severity, and stability over time, in addition to GDP duration, scope, and program rate. The most common GDPs issued in less severe and stable weather were those with high program rates, medium-to-wide departure scopes and short-to-medium durations. The most common GDPs issued in severe and unstable weather were those with medium-to-low program rates, medium-to-wide scopes, and medium-to-long durations.

The results of the Configural Frequency Analysis (CFA) suggest that under stable, less severe forecasted weather, GDPs with large scope (i.e. more impacted flights) scored higher on the efficiency metric than would be expected. This could be attributed to the fact that, stable weather conditions lead to more stable GDPs despite a wide scope. Under same weather forecast, for those GDPs with high program rates, medium durations, and narrow scopes, their capacity utilization is higher than expected. This could be attributed to less flights lead to fewer cancellations and more arrivals (albeit delayed), and therefore, higher capacity utilization. The

results also suggest that program rates are set more conservatively than needed for some adverse weather conditions that end earlier than expected, with GDP being canceled early as well. GDPs with fewer revisions were associated with a higher predictability score but lower efficiency score.

5.3 Contributions

This paper has introduced a novel method of merging disparate but complementary datasets and applying machine learning techniques to gain more insights into GDPs – particularly with respect to their changing characteristics and parameters. Four major research contributions have been identified for this research:

- We have generated a unique and comprehensive dataset describing GDP evolution, including GDP plans, flight schedules and times based on Traffic Management Initiative (TMI) data, TAF data, METAR data and Individual Flight (IF) data.
- It has identified the characteristics of how GDPs at EWR evolved over their lifetimes, by identifying 10 scenarios for the evolving GDPs under changing forecasted weather, and detecting the correlations between the scenarios and GDP performances. The results may be helpful to organizations like the FAA in better understanding GDPs.
- It proposed an exploration process to characterize GDP evolution by integrating unsupervised learning method, clustering methods and multivariate data analysis method. It can provide additional insights for researchers about the approaches to studying GDP evolution in multi-dimensional space.
- A visualization method was developed to present GDP evolution based on high-dimensional and time-varying GDP data. Through this method, big data involving time and multiple variables can be presented in 2-dimensional greyscale grid.

5.4 Research limitations and future work

There are several limitations in this research that may be addressed in future studies.

First, it is recommended that additional data be utilized to provide a more comprehensive operational picture of GDPs, and that a wider range of performance metrics be considered in the CFA analysis. In particular, the On-Time Performance dataset from the Bureau of Transportation Statistics (Bureau of Transportation Statistics, 2017) may be utilized to extract schedule data for individual flights.

Second, this study only takes into account impact of forecast weather conditions on GDP plans when building scenarios for GDPs. In future studies, more causes of GDPs, such as flight demand and runway conditions, can be considered into GDP characterization.

In addition, it is also recommended that the patterns of how GDPs evolve over their lifetimes, with respect to several key variables identified using statistical analysis and dimensionality reductions, be further explored using other novel machine learning techniques that may provide new and useful insights. For example, learning the patterns of the colored space instead of the size of the colored space. In this way, the feature of "real time" of the day can be integrated into the GDP evolution data.

References

Allan, S. S., Beesley, J. A., Evans, J. E., & Gaddy, S. G. (2001). Analysis of delay causality at Newark International Airport. *4th USA/Europe Air Traffic Management R&D Seminar* (pp. 1-11). Santa Fe: ATM.

Ball, M. O., & Lulli, G. (2004). Ground Delay Programs: Optimizing over the included flight set based on distance. *Air Traffic Control Quarterly*, 1-25.

Ball, M. O., Hoffman, R., & Knorr, D. (2000). Assessing the benefits of Collaborative Decision Making in air traffic management. *USA/Europe Air Traffic Management R & D Seminar* (pp. 1-11). Naples: ATM.

Ball, M. O., Hoffman, R., & Mukherjee, A. (2010, February). Ground Delay Program planning under uncertainty based on the ration-by-distance principle. *Transportation Science*, 1-14.

Ball, M. O., Hoffman, R., Chen, C., & Vossen, T. (2000). *Collaborative Decision Making in air traffic management: Current and future research directions.* National Center of Excellence in Aviation Operations Research.

Barnhart, C., Bertsimas, D., Caramanis, C., & Fearing, D. (2012, May). Equitable and efficient coordination in traffic flow management. *Transportation Science*, 262-280.

Bloem, M., & Bambos, N. (2015). Ground Delay Program analytics with behavioral cloning and inverse reinforcement learning. *Journal of Aerospace Information Systems*.

Bureau of Transportation Statistics. (2017). *Bureau of Transportation Statistics*. Retrieved from United States Department of Transportation: https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time

Buxi, G., & Hansen, M. (2011). Generating probabilistic capacity profiles from weather forecast: A design-of-experiment approach. *USA/Europe Air Traffic Management Research & Development Seminar* (pp. 1-10). Berlin: ATM.

Delgado, L.; Prats, X.; Sridhar, B. (2013). Cruise speed reduction for ground delay programs: A case study for San Francisco International Airport arrivals. *Transportation Research Part C: Emerging Technologies*, 83-96.

Donohue, G. L., Shaver, R. D., & Edwards, E. (2008). *Terminal chaos: Why U.S. air travel is broken and how can we fix it.* Reston: AIAA.

Eye, A., Spiel, C., & Wood, P. K. (1996). Configural frequency analysis in applied psychological research. *Applied Psychology*, 301-327.

FAA. (2003). *ASPM Individual Flight Data Dictionary.* Washington, D.C.: FAA.

FAA. (2008). *EWR Layout*. Retrieved from FAA: http://www.fly.faa.gov/Information/east/zny/ewr/EWR_layout.pdf

FAA. (2009a). *Traffic Flow Management in the National Airspace System.* Washington D.C.: FAA.

FAA. (2009b). *Air Traffic Management Glossary of Terms*. Retrieved from Federal Aviation Administration: http://www.fly.faa.gov/Products/Glossary_of_Terms/glossary_of_terms.html

FAA. (2014a). *Air Traffic Control System Command Center (ATCSCC)*. Retrieved from Federal Aviation Administration: https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/systemops/nas_ops/atcscc/

FAA. (2014b). *Traffic Flow Management System (TFMS)*. Retrieved from FAA: http://aspmhelp.faa.gov/index.php/Traffic_Flow_Management_System_(TFMS)

FAA. (2014c). *Airline Service Quality Performance (ASQP)*. Retrieved from FAA: http://aspmhelp.faa.gov/index.php/ASQP

FAA. (2014d). *Newark Liberty International Airport Capacity Profile*. Retrieved from FAA: https://www.faa.gov/airports/planning_capacity/profiles/media/EWR-Airport-Capacity-Profile-2014.pdf

FAA. (2015). *OOOI Data*. Retrieved from FAA: http://aspmhelp.faa.gov/index.php/OOOI_Data

FAA. (2016). Approaches and Landings. In FAA, *Airplane Flying Handbook* (pp. 8-17). Washington, D.C.: U.S. Department of Transportation.

FAA. (2017a). *Ground Delay Program*. Retrieved from Federal Aviation Administration: http://www.fly.faa.gov/Products/AIS_ORIGINAL/shortmessage.html

FAA. (2017b). *Federal Aviation Administration*. Retrieved from CDM Home: http://cdm.fly.faa.gov/

Federal Aviation Administration; National Weather Service. (2010). *Aviation Weather Services. Advisory circular, AC 00-45G, Change 1.* Oklahoma City: National Weather Service, Federal Aviation Administration.

Grabbe, S., Sridhar, B., & Mukherjee, A. (2013). Similar days in the NAS: an airport perspective. *AIAA Aviation Technology, Integration, and Operations* (pp. 1-14). Los Angeles: AIAA.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 504-507.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 504-507.

Hoffman, R., & Ball, M. O. (2000). Measuring Ground Delay Program effectiveness using the rate control index. *The Journal of Air Traffic Control, 42*(2), 19-23.

Hoffman, R., Ball, M. O., & Mukherjee, A. (2007). Ration-by-distance with equity guarantees: A new approach to ground delay program planning and control. *7th USA/Europe Air Traffic Management R&D Seminar* (pp. 1-29). Barcelona: ATM.

Inniss, T. R., & Ball, M. O. (2004). Estimating One-Parameter Airport Arrival Capacity Distributions for Air Traffic Flow Management. *Air Traffic Control Quarterly, 12*(3), 223-251.

International Civil Aviation Organization. (2005). *Global Air Traffic Management Operational Concept.* Quebec: International Civil Aviation Organization.

Jonkeren, O., Rietveld, P., & Ommeren, J. V. (2007). Climate change and inland waterway transport: welfare effects of low water levels on the river Rhine. *Journal of Transport Economics and Policy*, 387-411.

Kaufman, L., & Rousseeuw, P. (1987). *Clustering by means of medoids.* Amsterdam: North-Holland.

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis.* New York: Wiley.

Khurana, K. C. (2009). *Aviation management: global perspectives.* New Delhi: Global India Publications.

Kim, A., & Hansen, M. (2013). Deconstructing delay: A non-Parametric approach to analyzing delay changes in single server queuing systems. *Transportation Research Part B: Methodological, 58*, 119-133.

Kim, A., Rokib, A. S., & Liu, Y. (2015). Refinements to a procedure for estimating airfield capacity.

Transportation Research Record: Journal of the Transportation Research Board, 18-24.
Kuhn, K. (2016). A Methodology for identifying similar days in air traffic flow management initiative planning. *Transportation Research Part C: Emerging Technologies*, 1-15.
Kuhn, K., Shah, A., Skeels, C., & Murra, K. (2016). *Traffic flow management plan characterization.* Santa Monica: RAND Corporation.
Kulkarni, D., Wang, Y., & Sridhar, B. (2013). Data mining for understanding and improving decision-making affecting ground delay programs. *Digital Avionics Systems Conference* (pp. 5B1-1). East Syracuse: IEEE.
Liu, J., Li, K., Yin, M., Zhu, X., & Han, K. (2017). Optimizing key parameters of Ground Delay Program with uncertain airport capacity. *Journal of Advanced Transportation*, 1-9.
Liu, P., Hansen, M., & Mukherjee, A. (2008). Scenario-Based air traffic flow management: from theory to practice. *Transportation Research Part B: Methodological*, 685-702.
Liu, Y., & Hansen. (2014). Evaluation of the performance of Ground Delay Programs. *Transportation Research Board*, 54-64.
Liu, Y., & Hansen, M. (2013). Ground Delay Program decision-making using multiple criteria: A aingle airport case. *World Conference on Transport Research* (pp. 1-42). Rio de Janeiro: WCTR.
MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *15th Berkeley symposium on mathematical statistics and probability* (pp. 281-297). Berkeley: University of California.
Manley, B., & Sherry, L. (2008). The impact of Ground Delay Program (GDP) rationing rules on passenger and airline equity. *Integrated Communications, Navigation and Surveillance Conference.* Bethesda: IEEE.
McCartney, S. (2011, August 4). *The one airport to avoid Is ... - 40 of the 100 most-delayed flights in the country begin or end in Newark, N.J.* Retrieved from The Wall Street Journal: https://www.wsj.com/articles/SB10001424053111903885604576486111667671024
Mukherjee, A., & Hansen, M. (2007). Dynamic Stochastic Model for Dynamic Stochastic Model for a Single Airport Ground a Single Airport Ground Holding Problem. *Transportation Science, 41*(4), 444-456.
Mukherjee, A., Grabbe, S., & Sridhar, B. (2014). Predicting Ground Delay Program at an Airport Based on Meteorological Conditions. *14th AIAA Aviation Technology, Integration, and Operations Conference* (pp. 1-18). Atlanda: AIAA.
National Weather Service. (2016). *National Weather Service Instruction 10-813.* Silver Spring: National Weather Service. Retrieved from NOAA's National Weather Service, Aviation Weather Center: http://www.aviationweather.gov/static/help/taf-decode.php
National Weather Service. (2017). *TAF Decoder*. Retrieved from Aviation Weather Center: http://www.aviationweather.gov/static/help/taf-decode.php
Neufville, R., & Odoni, A. (2003). *Airport Systems: Planning, Design and Management.* New York: McGraw-Hill.
OpenFlights. (2017, January). *Airport database*. Retrieved from OpenFlights: https://openflights.org/data.html
Panchal, G., Ganatra, A., Kosta, Y. P., & Panchal, D. (2011). Behaviour analysis of multilayer perceptronswith multiple hidden neurons and hidden layers. *International Journal of Computer Theory and Engineering*, 3(2), 332.
RAND Corporation. (2016a). *Air traffic flow management plans associated with similar days.* Santa Monica: RAND Corporation.

RAND Corporation. (2016b). *Performance metric ranking of air traffic flow management plans.* Santa Monica: RAND Corporation.

Reynolds, D. W., Clark, D. A., Wilson, F. W., & Cook, L. (2012). *Forecast-Based Decision Support for San Francisco International Airport: A NextGen Prototype System That Improves Operations during Summer Stratus Season.* Boston: American Meteorological Society.

Richetta, O., & Odoni, A. R. (1993). Solving optimally the static ground-holding policy problem in air traffic control. *Transportation science*, 228-238.

Richetta, O., & Odoni, A. R. (1994). Dynamic solution to the ground-holding problem in air traffic control. *Transportation research part A: Policy and practice*, 167-185.

Shin, H. C., Orton, M., Collins, D. J., Doran, S., & Leach, M. O. (2011). Autoencoder in time-series analysis for unsupervised tissues characterisation in a large unlabelled medical image dataset. *Machine Learning and Applications and Workshops (ICMLA), 10th International Conference* (pp. 259-264). Honolulu: IEEE.

Smith, D. A., Sherry, L., & Donohue, G. (2008). Decision support tool for predicting aircraft arrival rates, Ground Delay Programs, and airport delays from weather forecasts. *International Conference on Research in Air Transportation* (pp. 1-12). Fairfax: IEEE.

Tan, P., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining.* Pearson Education.

The Port Authority of New York and New Jersey. (2015). *Airport Traffic Report.* New York: The Port Authority of NY&NJ.

The Port Authority of New York and New Jersey. (2017). *Facts & Information*. Retrieved from Newark Liberty International Airport: http://www.panynj.gov/airports/ewr-facts-info.html

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 411-423.

U.S. Global Change Research Program. (2014). *2014 National Climate Assessment*. Retrieved from 2014 National Climate Assessment: http://nca2014.globalchange.gov/report

UQAM Atmosphere Sciences Group. (2017). *METAR Study Guide, Aviation Routine Weather Report (METAR)*. Retrieved from Center Meteo UQAM: http://meteocentre.com/doc/metar.html

Veness, C. (2016). *Calculate distance, bearing and more between Latitude/Longitude points*. Retrieved from Movable Type Scripts: http://www.movable-type.co.uk/scripts/latlong.html

Vossen, T., & Ball, M. O. (2005, 2 1). Slot Trading Opportunities in Collaborative Ground Delay Programs. *Transportation Science*, 29-43.

Wang, Y. (2011). Prediction of weather impacted airport capacity using ensemble learning. *In Digital Avionics Systems Conference (DASC)* (pp. 2D6-1). Seattle: IEEE.

Wang, Y., & Kulkarni, D. (2011). *Modeling weather impact on ground delay programs.* Toulouse: NASA Ames Research Center.

Willemain, T. R. (2002, 10). Contingencies and Cancellations in Ground Delay Programs. *Air Traffic Control Quarterly*, pp. 43-64.

Xiong, J. (2010). *Revealed Preference of Airlines' Behavior under Air Traffic Management Initiatives.* Berkeley: UC Berkeley.

Zhang, Y., & Hansen, M. (2009). Regional GDP — Extending Ground Delay Programs to Regional Airport Systems. *Eighth USA/Europe Air Traffic Management Research and Development Seminar.* Napa.

Appendix A R Code for preparing GDP evolution data

#Part 1 import GDP data

```
TMI = read.csv("advisory.csv")
focus = which(TMI$AdvisoryType == "GDP"| TMI$AdvisoryType == "GDP CNX")
GDP = TMI[focus,]
focus = which(GDP$ControlElement == "EWR/ZNY")
EWR_GDP = GDP[focus,]
```

 # Part 2 Change timezone for GDP times (GMT->NY)

```
EWR_GDP$SendDate.Time.UTC =
as.POSIXct(strptime(as.character(EWR_GDP$SendDate.Time.UTC),"%Y-%m-%d %H:%M",tz
="GMT"))
EWR_GDP$Derived.BgnDate.Time.UTC =
as.POSIXct(strptime(as.character(EWR_GDP$Derived.BgnDate.Time.UTC),"%Y-%m-%d %H:
%M",tz="GMT"))
EWR_GDP$Derived.EndDate.Time.UTC =
as.POSIXct(strptime(as.character(EWR_GDP$Derived.EndDate.Time.UTC),"%Y-%m-%d %H:
%M",tz="GMT"))
EWR_GDP$Eff.Bgn.Date.Time.UTC =
as.POSIXct(strptime(as.character(EWR_GDP$Eff.Bgn.Date.Time.UTC),"%Y-%m-%d %H:%M
",tz="GMT"))
EWR_GDP$Eff.End.Date.Time.UTC =
as.POSIXct(strptime(as.character(EWR_GDP$Eff.End.Date.Time.UTC),"%Y-%m-%d %H:%M
",tz="GMT"))
EWR_GDP$GDP.Bgn.Date.Time.UTC=
as.POSIXct(strptime(as.character(EWR_GDP$GDP.Bgn.Date.Time.UTC),"%Y-%m-%d %H:%
M",tz="GMT"))
EWR_GDP$GDP.End.Date.Time.UTC=
as.POSIXct(strptime(as.character(EWR_GDP$GDP.End.Date.Time.UTC),"%Y-%m-%d %H:%
M",tz="GMT"))
EWR_GDP$GDPX.Bgn.Date.Time.UTC=
as.POSIXct(strptime(as.character(EWR_GDP$GDPX.Bgn.Date.Time.UTC),"%Y-%m-%d %H:
%M",tz="GMT"))
EWR_GDP$GDPX.End.Date.Time.UTC=
as.POSIXct(strptime(as.character(EWR_GDP$GDPX.End.Date.Time.UTC),"%Y-%m-%d %H:
%M",tz="GMT"))
EWR_GDP$RootAdvisoryDate.UTC=as.POSIXct(strptime(as.character(EWR_GDP$RootAdvis
oryDate.UTC),"%Y-%m-%d",tz="GMT"))

attr(EWR_GDP$SendDate.Time.UTC, "tzone") <- "America/New_York"
attr(EWR_GDP$Derived.BgnDate.Time.UTC, "tzone") <- "America/New_York"
attr(EWR_GDP$Derived.EndDate.Time.UTC, "tzone") <- "America/New_York"
attr(EWR_GDP$Eff.Bgn.Date.Time.UTC, "tzone") <- "America/New_York"
attr(EWR_GDP$Eff.End.Date.Time.UTC, "tzone") <- "America/New_York"
```

```
attr(EWR_GDP$GDP.Bgn.Date.Time.UTC, "tzone") <- "America/New_York"
attr(EWR_GDP$GDP.End.Date.Time.UTC, "tzone") <- "America/New_York"
attr(EWR_GDP$GDPX.Bgn.Date.Time.UTC, "tzone") <- "America/New_York"
attr(EWR_GDP$GDPX.End.Date.Time.UTC, "tzone") <- "America/New_York"
attr(EWR_GDP$RootAdvisoryDate.UTC, "tzone") <- "America/New_York"

# Part 3 Duration variables
N=nrow(EWR_GDP)
## duration of each single advisory
EWR_GDP$Duration_Advisory = difftime(EWR_GDP$Derived.EndDate.Time.UTC,
EWR_GDP$Derived.BgnDate.Time.UTC, units = "hours")
## duration of the GDP plan when a new advisory is issued
EWR_GDP$RootBgnTime= EWR_GDP$Derived.BgnDate.Time.UTC
for (i in 1:N){
if (EWR_GDP$Is.RootAdvisory[i]== "Yes") {
j=i
root= as.numeric(EWR_GDP$RootAdvisoryNumber[j])
roottime= EWR_GDP$RootAdvisoryDate.UTC[j]
}
EWR_GDP$RootBgnTime[as.numeric(EWR_GDP$RootAdvisoryNumber)== root  &
EWR_GDP$RootAdvisoryDate.UTC== roottime]= EWR_GDP$Derived.BgnDate.Time.UTC[j]
}
EWR_GDP$Duration_Initiative[as.character(EWR_GDP$AdvisoryType)== "GDP"]=
difftime(EWR_GDP$Derived.EndDate.Time.UTC[as.character(EWR_GDP$AdvisoryType)==
"GDP"], EWR_GDP$RootBgnTime[as.character(EWR_GDP$AdvisoryType)== "GDP"], units
= "hours")
EWR_GDP$Duration_Initiative[as.character(EWR_GDP$AdvisoryType)== "GDP CNX"]=
difftime(EWR_GDP$Derived.BgnDate.Time.UTC[as.character(EWR_GDP$AdvisoryType)==
"GDP CNX"], EWR_GDP$RootBgnTime[as.character(EWR_GDP$AdvisoryType)== "GDP
CNX"], units = "hours")
## Actual duration of the GDP plan when no more new advisory will be issued
for (i in 1:N){
if (EWR_GDP$Is.RootAdvisory[i]== "Yes") {
root= as.numeric(EWR_GDP$RootAdvisoryNumber[i])
roottime= EWR_GDP$RootAdvisoryDate.UTC[i]}
last = max(which(as.numeric(EWR_GDP$RootAdvisoryNumber)== root  &
EWR_GDP$RootAdvisoryDate.UTC== roottime))
EWR_GDP$Duration_Actual[as.numeric(EWR_GDP$RootAdvisoryNumber)== root  &
EWR_GDP$RootAdvisoryDate.UTC== roottime]=EWR_GDP$Duration_Initiative[last]
}

# Part 4 Number of Modifications of GDP plan including & excluding GDP CNX
for (i in 1:N){
if (EWR_GDP$Is.RootAdvisory[i]== "Yes") {
root= as.numeric(EWR_GDP$RootAdvisoryNumber[i])
roottime= EWR_GDP$RootAdvisoryDate.UTC[i]
```

```
}
EWR_GDP$Number_Revisions_NoCNX[as.numeric(EWR_GDP$RootAdvisoryNumber)==
root  & EWR_GDP$RootAdvisoryDate.UTC== roottime] =
nrow(EWR_GDP[which(as.character(EWR_GDP$AdvisoryType)== "GDP"&
as.numeric(EWR_GDP$RootAdvisoryNumber)== root  &
EWR_GDP$RootAdvisoryDate.UTC== roottime),])-1
EWR_GDP$Number_TotModif_incldCNX[as.numeric(EWR_GDP$RootAdvisoryNumber)==
root  & EWR_GDP$RootAdvisoryDate.UTC== roottime] =
nrow(EWR_GDP[which(as.numeric(EWR_GDP$RootAdvisoryNumber)== root  &
EWR_GDP$RootAdvisoryDate.UTC== roottime),])-1
}
```

 # Part 5 Early Cancel, Lead Time

##early cancel time

```
for (i in 1:N){
if (as.character(EWR_GDP$AdvisoryType[i])== "GDP CNX") {
beforeCNX=max(which(as.character(EWR_GDP$AdvisoryType)== "GDP" &
as.numeric(EWR_GDP$RootAdvisoryNumber)==
as.numeric(EWR_GDP$RootAdvisoryNumber[i])  & EWR_GDP$RootAdvisoryDate.UTC==
EWR_GDP$RootAdvisoryDate.UTC[i]))

EWR_GDP$EarlyCancelTime[as.numeric(EWR_GDP$RootAdvisoryNumber)==
as.numeric(EWR_GDP$RootAdvisoryNumber[i])  & EWR_GDP$RootAdvisoryDate.UTC==
EWR_GDP$RootAdvisoryDate.UTC[i]]= EWR_GDP$Duration_Initiative[beforeCNX]-
EWR_GDP$Duration_Initiative[i]
}
}
```

##lead time

```
EWR_GDP$LeadTime = difftime(EWR_GDP$Derived.BgnDate.Time.UTC,
EWR_GDP$SendDate.Time.UTC, units = "hours")
```

 # Part 6 time of the day, day of the week, month-day of the year

##send time of the day

```
EWR_GDP$SendTime= format(EWR_GDP$SendDate.Time.UTC, "%H:%M")
```

##begin time of the day

```
EWR_GDP$BgnTime= format(EWR_GDP$Derived.BgnDate.Time.UTC, "%H:%M")
```

##end time of the day

```
EWR_GDP$EndTime= format(EWR_GDP$Derived.EndDate.Time.UTC, "%H:%M")
```

##GDP month-day of the year

```
EWR_GDP$BgnDay=format(EWR_GDP$Derived.BgnDate.Time.UTC, "%m-%d")
```

##GDP day of the week

```
EWR_GDP$BgnWeekday=weekdays(EWR_GDP$Derived.BgnDate.Time.UTC, abbreviate =
FALSE)
```

#Part 7 match TAF data (hourly)

```
TAF = read.csv("EWRTAF.csv")
```

```
## combine date and time
TAF$issue.time = as.character(paste(as.character(TAF$Issued.Year),
as.character(TAF$Issued.Month), as.character(TAF$Issued.Day), as.character
(TAF$Issued.Hour), as.character (TAF$Issued.Minute)))
TAF$start.time = as.character(paste(as.character(TAF$From.Year),
as.character(TAF$From.Month), as.character(TAF$From.Day), as.character (TAF$From.Hour),
as.character (TAF$From.Minute)))
TAF$end.time = as.character(paste(as.character(TAF$To.Year), as.character(TAF$To.Month),
as.character(TAF$To.Day), as.character (TAF$To.Hour), as.character (TAF$To.Minute)))

## convert date time to datetime format
TAF$issue.time = strptime(TAF$issue.time, "%Y %m %d %H %M", tz ="GMT")
TAF$issue.time = format(TAF$issue.time, format = "%Y-%m-%d %H:%M", tz ="GMT" )
TAF$issue.time =
as.POSIXct(strptime(as.character(TAF$issue.time),"%Y-%m-%d %H:%M",tz="GMT"))

TAF$start.time = strptime(TAF$start.time, "%Y %m %d %H %M", tz ="GMT")
TAF$start.time = format(TAF$start.time, format = "%Y-%m-%d %H:%M", tz ="GMT" )
TAF$start.time =
as.POSIXct(strptime(as.character(TAF$start.time),"%Y-%m-%d %H:%M",tz="GMT"))

TAF$end.time = strptime(TAF$end.time, "%Y %m %d %H %M", tz ="GMT")
TAF$end.time = format(TAF$end.time, format = "%Y-%m-%d %H:%M", tz ="GMT" )
TAF$end.time =
as.POSIXct(strptime(as.character(TAF$end.time),"%Y-%m-%d %H:%M",tz="GMT"))

## convert GMT time to NY time
attr(TAF$issue.time, "tzone") <- "America/New_York"
attr(TAF$start.time, "tzone") <- "America/New_York"
attr(TAF$end.time, "tzone") <- "America/New_York"

## identify low visibility/ceiling, precipitation, and crosswind weather
TAF$IMC=0
TAF$IMC[TAF$Ceiling < 10 | TAF$Visibility < 3]=1

TAF$MCIC=0
TAF$MCIC[TAF$Ceiling < 30 | TAF$Visibility < 4]=1

TAF$PC=0
TAF$PC[TAF$DZ+TAF$RA+ TAF$SN+ TAF$SG+ TAF$IC+ TAF$PL+ TAF$GR+
TAF$GS+ TAF$UP>0]=1

TAF$WindAngle= as.numeric(as.character(TAF$Wind.Angle))
TAF$CW0422 = round(as.numeric(TAF$Wind.Spee
d)*abs(sin((TAF$WindAngle-40)*(pi/180))), digits = 0)
```

```
TAF$CW1129 = round(as.numeric(TAF$Wind.Speed)*abs(sin((TAF$WindAngle-
110)*(pi/180))), digits = 0)

## delete TAFs beyond time horizon (2010-1-1-2014-8-31)
boundary1 = as.character("2010-1-1 00:00")
boundary2 = as.character("2014-8-31 23:59")

bdr1 = strptime(boundary1, "%Y-%m-%d %H:%M", tz ="America/New_York")
bdr1 =  format(boundary1, format = "%Y-%m-%d %H:%M", tz ="America/New_York" )
bdr1 = as.POSIXct(strptime(as.character(boundary1),"%Y-%m-%d %H:%M", tz
="America/New_York"))
bdr2 = strptime(boundary2, "%Y-%m-%d %H:%M", tz ="America/New_York")
bdr2 =  format(boundary2, format = "%Y-%m-%d %H:%M", tz ="America/New_York" )
bdr2 = as.POSIXct(strptime(as.character(boundary2),"%Y-%m-%d %H:%M", tz
="America/New_York"))

TAF1=TAF[TAF$end.time>bdr1 & TAF$start.time < bdr2,]

##delete duplicated TAF with same issue, begin, and end time
TAF2 = TAF1[order(TAF1$IMC,TAF1$MCIC, TAF1$TS, TAF1$CW0422, TAF1$CW1129,
TAF1$PC, decreasing = TRUE ),]
TAF3= TAF2[!duplicated(TAF2[c("end.time","start.time","issue.time")]),]

##delete data with abnormal forecast time horizon
TAF3$timediff= difftime(TAF3$end.time, TAF3$start.time,units ="hours" )
TAF4 = TAF3[TAF3$timediff >0& TAF3$timediff<=24, ]
write.csv(x = TAF4, file="TAF-20140831-nonduplicated-24h.csv")

TAF4 = read.csv("TAF-20140831-nonduplicated-24h.csv",  as.is=TRUE)
TAF4$issue.time= as.POSIXct(TAF4$issue.time, tz =  "America/New_York")
TAF4$start.time= as.POSIXct(TAF4$start.time, tz =  "America/New_York")
TAF4$end.time= as.POSIXct(TAF4$end.time, tz =  "America/New_York")

EWR_GDP$IMC <- ""
EWR_GDP$MCIC <- ""
EWR_GDP$Visibility <- ""
EWR_GDP$Ceiling <- ""
EWR_GDP$TS <- ""
EWR_GDP$CW0422<- ""
EWR_GDP$CW1129<- ""
EWR_GDP$PC<- ""
EWR_GDP$RNSNIC<- ""
for ( i in 1:N) {
TAFGDP1=TAF4[TAF4$issue.time < EWR_GDP$SendDate.Time.UTC[i] & TAF4$end.time >
EWR_GDP$Derived.BgnDate.Time.UTC[i] & TAF4$start.time<
EWR_GDP$Derived.EndDate.Time.UTC[i],]
```

```
H = ceiling(as.numeric(EWR_GDP$Duration_Advisory[i]))

for (h in 1:H) {
TAFGDP2=TAFGDP1[TAFGDP1$start.time<EWR_GDP$Derived.BgnDate.Time.UTC[i]+h*6
0*60 & TAFGDP1$end.time> EWR_GDP$Derived.BgnDate.Time.UTC[i]+(h-1)*60*60,]

TAFGDP3 = TAFGDP2[order(TAFGDP2$issue.time,decreasing = TRUE)[1],]

EWR_GDP$IMC[i] = paste(EWR_GDP$IMC[i], TAFGDP3$IMC, sep = " ")
EWR_GDP$MCIC[i] = paste(EWR_GDP$MCIC[i], TAFGDP3$MCIC, sep = " ")
EWR_GDP$Visibility[i] = paste(EWR_GDP$Visibility[i], TAFGDP3$Visibility, sep = " ")
EWR_GDP$Ceiling[i] = paste(EWR_GDP$Ceiling[i], TAFGDP3$Ceiling, sep = " ")
EWR_GDP$TS[i] = paste(EWR_GDP$TS[i], TAFGDP3$TS, sep = " ")
EWR_GDP$CW0422[i] = paste(EWR_GDP$CW0422[i], TAFGDP3$CW0422, sep = " ")
EWR_GDP$CW1129[i] = paste(EWR_GDP$CW1129[i], TAFGDP3$CW1129, sep = " ")
EWR_GDP$PC[i] = paste(EWR_GDP$PC[i], TAFGDP3$PC, sep = " ")
EWR_GDP$RNSNIC[i] = paste(EWR_GDP$RNSNIC[i], TAFGDP3$RNSNIC, sep = " ")
}
}
write.csv(x = EWR_GDP, file="EWR_GDP7【TAF】.csv")
```

#Part 8 Convert program rate to hourly vector

```
EWR_GDP$ProgramRate = as.character(EWR_GDP$ProgramRate)
EWR_GDP$ProgramRateVct= EWR_GDP$ProgramRate

for (i in 1:N) {
if (EWR_GDP$AdvisoryType[i] == "GDP" & !grepl("/",EWR_GDP$ProgramRate[i])) {
DurationRoundUp = ceiling(as.numeric(EWR_GDP$Duration_Advisory[i]))

EWR_GDP$ProgramRateVct[i]=paste(replicate(DurationRoundUp,
EWR_GDP$ProgramRate[i]), collapse = "/")
}
}
```

#Part 9 Types of Dep Scope

```
EWR_GDP$Dep.Scope = as.character(EWR_GDP$Dep.Scope)
EWR_GDP$DepScopeType = "ARTCC"
EWR_GDP$DepScopeType[grepl("0", EWR_GDP$Dep.Scope) | grepl("5",
EWR_GDP$Dep.Scope)]= "Radius"
EWR_GDP$DepScopeType[grepl("-", EWR_GDP$Dep.Scope)]= "-"
```

 # Part 10 preprocess IF

```
EWRIF = read.csv("EWR_IF.csv")
EWRIF$ARR_YYYY = as.character(substr(EWRIF$ARR_YYYYMM,1,4))
EWRIF$ARR_MM = as.character(substr(EWRIF$ARR_YYYYMM,5,6))
```

```
EWRIF$SchIn = as.character(paste(EWRIF$ARR_YYYY, EWRIF$ARR_MM,
as.character(EWRIF$ARR_DAY), as.character (EWRIF$SCHINTM)))
EWRIF$SchIn = strptime(EWRIF$SchIn, "%Y %m %d %H:%M", tz ="America/New_York")

DEPARP = read.csv("DEPARP.csv")
DEPARP$ARTCC=as.character(DEPARP$ARTCC)
DEPARP$Country=as.character(DEPARP$Country)
DEPARP$Mile=as.character(DEPARP$Mile)

DEPARP$LocationID=as.character(DEPARP$LocationID)
EWRIF$DEP_LOCID=as.character(EWRIF$DEP_LOCID)

for (i in 1:nrow(DEPARP)) {
EWRIF$ARTCC[EWRIF$DEP_LOCID %in% DEPARP$LocationID[i]] = DEPARP$ARTCC
[i]
EWRIF$Country[EWRIF$DEP_LOCID %in% DEPARP$LocationID[i]] =
DEPARP$Country[i]
EWRIF$Mile[EWRIF$DEP_LOCID %in% DEPARP$LocationID[i]] = DEPARP$Mile[i]
}

write.csv(x = EWRIF, file="EWRIF-fixed.csv")
```

 # Part 11 Match IF and TMI (suggest to run this part separately with following parts as it is very slow)

```
EWR_GDP$Exempt.Dep.Facilities = as.character(EWR_GDP$Exempt.Dep.Facilities)
EWR_GDP$AdvisoryType =as.character(EWR_GDP$AdvisoryType)
EWR_GDP1=EWR_GDP[difftime(as.Date(EWR_GDP$AdvisoryDate.UTC), as.Date('2014-09-
01 '))<0,]
EWR_GDP1$Dep.Scope = as.character(EWR_GDP1$Dep.Scope)

EWR_GDP1$SchArrSchIn = 0
EWR_GDP1$ImpArrSchIn = 0

EWRIF$Mile = as.numeric(EWRIF$Mile)
EWRIF$DEP_LOCID=as.character(EWRIF$DEP_LOCID)

IFCA= EWRIF[EWRIF$Country == "CA",]
IFUS= EWRIF[EWRIF$Country == "US",]
IFINT= EWRIF[EWRIF$Country == "INT",]

for (i in 1:nrow(EWR_GDP1)) {
if (EWR_GDP1$AdvisoryType[i] == "GDP") {
#scheduled arrivals
EWR_GDP1$SchArrSchIn[i] =
length(which( EWRIF$SchIn >=EWR_GDP1$Derived.BgnDate.Time.UTC[i] & EWRIF$SchIn
<= EWR_GDP1$Derived.EndDate.Time.UTC[i]))
```

```
##CA flights
###Spatial
rows<-sapply(IFCA$DEP_LOCID, function(x) grepl(x,
EWR_GDP1$Canadian.Dep.Arpts.Included[i]))
IFGDPCA = IFCA[rows,]
###Temporal
CA = length(which(IFGDPCA$SchIn>=EWR_GDP1$Derived.BgnDate.Time.UTC[i] &
IFGDPCA$SchIn <=EWR_GDP1$Derived.EndDate.Time.UTC[i]))
##US flights
### tick out exempted flights
if (EWR_GDP1$Exempt.Dep.Facilities[i]!= "-") {
rows<-sapply(IFUS$DEP_LOCID, function(x) !grepl(x,
EWR_GDP1$Exempt.Dep.Facilities[i]))
IFUS1 = IFUS[rows,]
rows<- sapply(IFUS1$ARTCC, function(x) !grepl(x, EWR_GDP1$Exempt.Dep.Facilities[i]) )
IFGDPUS = IFUS1[rows,]
} else{
IFGDPUS= IFUS
}
### spatial scope
if (!grepl("ALL",EWR_GDP1$Dep.Scope[i]) ){
if(EWR_GDP1$DepScopeType[i] == "ARTCC")
{
IFGDPUS1=IFGDPUS[sapply(IFGDPUS$ARTCC, function(x) grepl(x,
EWR_GDP1$Dep.Scope[i] )),]
}
if(EWR_GDP1$DepScopeType[i] == "Radius")
{
IFGDPUS1 = IFGDPUS[IFGDPUS$Mile <= as.numeric(EWR_GDP1$Dep.Scope[i]),]
}
}else {
IFGDPUS1= IFGDPUS}
### temporal scope
US= length(which(IFGDPUS1$SchIn>=EWR_GDP1$Derived.BgnDate.Time.UTC[i] &
IFGDPUS1$SchIn<=EWR_GDP1$Derived.EndDate.Time.UTC[i]))

##Int Flights
#INT= length(which(IFINT$SchIn>=EWR_GDP1$Derived.BgnDate.Time.UTC[i] &
#IFINT$SchIn<=EWR_GDP1$Derived.EndDate.Time.UTC[i]))

##Flight No.
EWR_GDP1$ImpArrSchIn[i] = as.numeric(CA)+ as.numeric(US)
} }
write.csv(x = EWR_GDP1, file="EWR_GDP11.csv")
```

 # Part 12 change format of some variable, preparing for dimensionality reduction

```
EWR_GDP_IF = read.csv("EWR_GDP11.csv")
EWR_GDP_IF$SchArrSchIn =as.numeric(as.character(EWR_GDP_IF$SchArrSchIn))
EWR_GDP_IF$ImpArrSchIn= as.numeric(as.character(EWR_GDP_IF$ImpArrSchIn))

EWR_GDP1=EWR_GDP[difftime(as.Date(EWR_GDP$AdvisoryDate.UTC), as.Date('2014-09-
01 '))<0,]
EWR_GDP1$SchArrSchIn = EWR_GDP_IF$SchArrSchIn
EWR_GDP1$ImpArrSchIn = EWR_GDP_IF$ImpArrSchIn
```

##generate month

```
library(lubridate)
EWR_GDP1$Month = month(EWR_GDP1$Derived.BgnDate.Time.UTC)
EWR_GDP1$Month=as.numeric(as.character(EWR_GDP1$Month))
```

##generate time of the day

```
time1 = hm("0,0")
time2 = hm("6,0")
time3 = hm("9,0")
time4 = hm("12,0")
time5 = hm("15,0")
time6 = hm("18,0")
time7 = hm("21,0")
time8 = hm("24,0")

EWR_GDP1$SendTime=hm(EWR_GDP1$SendTime)
EWR_GDP1$BgnTime=hm(EWR_GDP1$BgnTime)
EWR_GDP1$EndTime=hm(EWR_GDP1$EndTime)

EWR_GDP1$SendTimePeriod[EWR_GDP1$SendTime<time2&
EWR_GDP1$SendTime>=time1] = 6 #"0-6"
EWR_GDP1$SendTimePeriod[EWR_GDP1$SendTime<time3&
EWR_GDP1$SendTime>=time2] =9 #"6-9"
EWR_GDP1$SendTimePeriod[EWR_GDP1$SendTime<time4&
EWR_GDP1$SendTime>=time3] =12 #"9-12"
EWR_GDP1$SendTimePeriod[EWR_GDP1$SendTime<time5&
EWR_GDP1$SendTime>=time4] =15 #"12-15"
EWR_GDP1$SendTimePeriod[EWR_GDP1$SendTime<time6&
EWR_GDP1$SendTime>=time5] =18 #"15-18"
EWR_GDP1$SendTimePeriod[EWR_GDP1$SendTime<time7&
EWR_GDP1$SendTime>=time6] =21 #"18-21"
EWR_GDP1$SendTimePeriod[EWR_GDP1$SendTime<time8&
EWR_GDP1$SendTime>=time7] =24 #"21-24"

EWR_GDP1$BgnTimePeriod[EWR_GDP1$BgnTime<time2&
EWR_GDP1$BgnTime>=time1] =6 #"0-6"
```

```
EWR_GDP1$BgnTimePeriod[EWR_GDP1$BgnTime<time3&
EWR_GDP1$BgnTime>=time2] =9 #"6-9"
EWR_GDP1$BgnTimePeriod[EWR_GDP1$BgnTime<time4&
EWR_GDP1$BgnTime>=time3] =12 #"9-12"
EWR_GDP1$BgnTimePeriod[EWR_GDP1$BgnTime<time5&
EWR_GDP1$BgnTime>=time4] =15 #"12-15"
EWR_GDP1$BgnTimePeriod[EWR_GDP1$BgnTime<time6&
EWR_GDP1$BgnTime>=time5] =18 #"15-18"
EWR_GDP1$BgnTimePeriod[EWR_GDP1$BgnTime<time7&
EWR_GDP1$BgnTime>=time6] =21 #"18-21"
EWR_GDP1$BgnTimePeriod[EWR_GDP1$BgnTime<time8&
EWR_GDP1$BgnTime>=time7] =24 #"21-24"

EWR_GDP1$EndTimePeriod[EWR_GDP1$EndTime<time2& EWR_GDP1$EndTime>=time1]
=6 #"0-6"
EWR_GDP1$EndTimePeriod[EWR_GDP1$EndTime<time3& EWR_GDP1$EndTime>=time2]
=9 #"6-9"
EWR_GDP1$EndTimePeriod[EWR_GDP1$EndTime<time4& EWR_GDP1$EndTime>=time3]
=12 #"9-12"
EWR_GDP1$EndTimePeriod[EWR_GDP1$EndTime<time5& EWR_GDP1$EndTime>=time4]
=15 #"12-15"
EWR_GDP1$EndTimePeriod[EWR_GDP1$EndTime<time6& EWR_GDP1$EndTime>=time5]
=18 #"15-18"
EWR_GDP1$EndTimePeriod[EWR_GDP1$EndTime<time7& EWR_GDP1$EndTime>=time6]
=21 #"18-21"
EWR_GDP1$EndTimePeriod[EWR_GDP1$EndTime<time8& EWR_GDP1$EndTime>=time7]
=24 #"21-24"

EWR_GDP1$SendTimePeriod= as.numeric(EWR_GDP1$SendTimePeriod)
EWR_GDP1$BgnTimePeriod= as.numeric(EWR_GDP1$BgnTimePeriod)
EWR_GDP1$EndTimePeriod= as.numeric(EWR_GDP1$EndTimePeriod)

##generate numeric weekday
EWR_GDP1$BgnWeekday=as.character(EWR_GDP1$BgnWeekday)
EWR_GDP1$WeekdayNo[EWR_GDP1$BgnWeekday =="Monday"] = 1
EWR_GDP1$WeekdayNo[EWR_GDP1$BgnWeekday =="Tuesday"] = 2
EWR_GDP1$WeekdayNo[EWR_GDP1$BgnWeekday =="Wednesday"] = 3
EWR_GDP1$WeekdayNo[EWR_GDP1$BgnWeekday =="Thursday"] = 4
EWR_GDP1$WeekdayNo[EWR_GDP1$BgnWeekday =="Friday"] = 5
EWR_GDP1$WeekdayNo[EWR_GDP1$BgnWeekday =="Saturday"] = 6
EWR_GDP1$WeekdayNo[EWR_GDP1$BgnWeekday =="Sunday"] = 7
EWR_GDP1$WeekdayNo = as.numeric(EWR_GDP1$WeekdayNo)

##change others to numeric
EWR_GDP1$Duration_Advisory= as.numeric(EWR_GDP1$Duration_Advisory)
EWR_GDP1$Duration_Actual= as.numeric(EWR_GDP1$Duration_Actual)
```

```
EWR_GDP1$Number_Revisions_NoCNX=
as.numeric(EWR_GDP1$Number_Revisions_NoCNX)
EWR_GDP1$Number_TotModif_incldCNX=
as.numeric(EWR_GDP1$Number_TotModif_incldCNX)
EWR_GDP1$LeadTime= as.numeric(EWR_GDP1$LeadTime)

#Part 13 match GDP with METAR data (hourly)
METAR = read.csv("EWRMETAR.csv")

## convert date time to datetime format
METAR$start.time = as.POSIXct(strptime(as.character(METAR$start.time),"%Y-%m-%d
%H:%M",tz="GMT"))
METAR$end.time = as.POSIXct(strptime(as.character(METAR$end.time),"%Y-%m-%d
%H:%M",tz="GMT"))

## convert GMT time to NY time
attr(METAR$start.time, "tzone") <- "America/New_York"
attr(METAR$end.time, "tzone") <- "America/New_York"

## identify non-VMC weather, precipitation, and crosswind
METAR$IMC=0
METAR$IMC[METAR$Ceiling < 10 | METAR$Visibility < 3]=1

METAR$MCIC=0
METAR$MCIC[METAR$Ceiling < 30 | METAR$Visibility < 4]=1

METAR$PC=0
METAR$PC[METAR$DZ+METAR$RA+ METAR$SN+ METAR$SG+ METAR$IC+
METAR$PL+ METAR$GR+ METAR$GS+ METAR$UP>0]=1

METAR$WindAngle= as.numeric(as.character(METAR$Wind.Angle))
METAR$CW0422 = round(as.numeric(METAR$Wind.Speed)*abs(sin((METAR$WindAngle-
40)*(pi/180))), digits = 0)
METAR$CW1129 = round(as.numeric(METAR$Wind.Speed)*abs(sin((METAR$WindAngle-
110)*(pi/180))), digits = 0)

library(lubridate)
METAR$Month = month(METAR$start.time)
METAR$Month =as.numeric(as.character(METAR$Month))

## delete METARs beyond time horizon (2010-1-1-2014-8-31)
boundary1 = as.character("2010-1-1 00:00")
boundary2 = as.character("2014-8-31 23:59")

bdr1 = strptime(boundary1, "%Y-%m-%d %H:%M", tz ="America/New_York")
bdr1 =  format(boundary1, format = "%Y-%m-%d %H:%M", tz ="America/New_York" )
```

```
bdr1 = as.POSIXct(strptime(as.character(boundary1),"%Y-%m-%d %H:%M", tz
="America/New_York"))
bdr2 = strptime(boundary2, "%Y-%m-%d %H:%M", tz ="America/New_York")
bdr2 =  format(boundary2, format = "%Y-%m-%d %H:%M", tz ="America/New_York" )
bdr2 = as.POSIXct(strptime(as.character(boundary2),"%Y-%m-%d %H:%M", tz
="America/New_York"))

METAR1=METAR[METAR$end.time>bdr1 & METAR$start.time < bdr2,]

##delete duplicated METAR with same issue, begin, and end time
METAR2 = METAR1[order(METAR1$IMC,METAR1$MCIC, METAR1$TS,
METAR1$CW0422, METAR1$CW1129, METAR1$PC, decreasing = TRUE ),]
METAR3= METAR2[!duplicated(METAR2[c("end.time","start.time")]),]
write.csv(x = METAR3, file="METAR-20140831-nonduplicated.csv")

METAR4 = read.csv("METAR-20140831-nonduplicated-.csv",  as.is=TRUE)
METAR4$start.time= as.POSIXct(METAR4$start.time, tz =  "America/New_York")
METAR4$end.time= as.POSIXct(METAR4$end.time, tz =  "America/New_York")

EWR_GDP1$MCIC_obs <- ""
EWR_GDP1$Vis_obs <- ""
EWR_GDP1$Ceiling_obs <- ""
EWR_GDP1$TS_obs <- ""
EWR_GDP1$CW0422_obs <- ""
EWR_GDP1$CW1129_obs <- ""
EWR_GDP1$PC_obs <- ""
EWR_GDP1$RNSNIC_obs <- ""

for ( i in 1:nrow(EWR_GDP1)) {

METARGDP1=METAR4[METAR4$end.time > EWR_GDP1$Derived.BgnDate.Time.UTC[i]
& METAR4$start.time< EWR_GDP1$Derived.EndDate.Time.UTC[i],]

H = ceiling(as.numeric(EWR_GDP1$Duration_Advisory[i]))

for (h in 1:H) {
METARGDP2=METARGDP1[METARGDP1$start.time<EWR_GDP1$Derived.BgnDate.Time
.UTC[i]+h*60*60 & METARGDP1$end.time>
EWR_GDP1$Derived.BgnDate.Time.UTC[i]+(h-1)*60*60,]

METARGDP3 = METARGDP2[order(METARGDP2$MCIC, METARGDP2$TS,
METARGDP2$CW0422, METARGDP2$CW1129, METARGDP2$RNSNIC,
METARGDP2$PC ,decreasing = TRUE)[1],]

EWR_GDP1$MCIC_obs[i] = paste(EWR_GDP1$MCIC_obs[i], METARGDP3$MCIC, sep = "
")
```

```
EWR_GDP1$Vis_obs[i] = paste(EWR_GDP1$Vis_obs[i], METARGDP3$Visibility, sep = " ")
EWR_GDP1$Ceiling_obs[i] = paste(EWR_GDP1$Vis_obs[i], METARGDP3$Ceiling, sep = "
")
EWR_GDP1$TS_obs[i] = paste(EWR_GDP1$TS_obs[i], METARGDP3$TS, sep = " ")
EWR_GDP1$CW0422_obs[i] = paste(EWR_GDP1$CW0422_obs[i], METARGDP3$CW0422,
sep = " ")
EWR_GDP1$CW1129_obs[i] = paste(EWR_GDP1$CW1129_obs[i], METARGDP3$CW1129,
sep = " ")
EWR_GDP1$PC_obs[i] = paste(EWR_GDP1$PC_obs[i], METARGDP3$PC, sep = " ")
EWR_GDP1$RNSNIC_obs[i] = paste(EWR_GDP1$RNSNIC_obs[i], METARGDP3$RNSNIC,
sep = " ")
}
}
```

 # Part 14 generate root advisory No. and modification No.

```
temp <-read.csv("EWR_GDP7【TAF】.csv")
temp=temp[difftime(as.Date(temp$AdvisoryDate.UTC), as.Date('2014-09-01 '))<0,]
temp$Visibility<- as.character(temp$Visibility)
temp$Ceiling<- as.character(temp$Ceiling)
EWR_GDP1 = cbind(EWR_GDP1, Visibility = temp$Visibility, Ceiling = temp$Ceiling)
EWR_GDP1$Visibility = as.character(EWR_GDP1$Visibility)
EWR_GDP1$Ceiling = as.character(EWR_GDP1$Ceiling)

##label every root GDP
EWR_GDP1$RootAdvisoryDate.UTC = as.character(EWR_GDP1$RootAdvisoryDate.UTC)
EWR_GDP1$Is.RootAdvisory = as.character(EWR_GDP1$Is.RootAdvisory)
Root = EWR_GDP1[EWR_GDP1$Is.RootAdvisory == "Yes",]
Root$RootNumber= c(1: nrow(Root))

##label root GDP no for every GDP
for (i in 1:nrow(Root)) {
rows = which(Root$RootAdvisoryDate.UTC[i] ==EWR_GDP1$RootAdvisoryDate.UTC &
Root$RootAdvisoryNumber[i]==EWR_GDP1$RootAdvisoryNumber)
for (j in 1:length(rows)) {
EWR_GDP1$RootNumber[rows[j]] <- Root$RootNumber[i]
}
}

##label mods for every GDP plan
EWR_GDP2= EWR_GDP1[order(EWR_GDP1$RootNumber, decreasing = FALSE),]
EWR_GDP2$ModNumber= 0
ModNo = 0
for (i in 2:nrow(EWR_GDP2)) {
RootNo = EWR_GDP2$RootNumber[i-1]
if (EWR_GDP2$RootNumber[i] == RootNo) {
ModNo = ModNo+1
```

```
EWR_GDP2$ModNumber[i] = ModNo
} else {
ModNo = 0}
}

##order data
EWR_GDP3= EWR_GDP2[order(EWR_GDP2$RootNumber, EWR_GDP2$ModNumber,
decreasing = FALSE),]
```

 # Part 15 convert to hourly GDPs

```
EWR_GDP3$ActAdvisoryDuration= EWR_GDP3$Duration_Advisory
EWR_GDP3$ActAdvisoryEndTime= EWR_GDP3$Derived.EndDate.Time.UTC
for (i in 1:(nrow(EWR_GDP3)-1)) {
if(EWR_GDP3$RootNumber[i]== EWR_GDP3$RootNumber[i+1] &
EWR_GDP3$Derived.BgnDate.Time.UTC[i+1] <
EWR_GDP3$Derived.EndDate.Time.UTC[i]) {
EWR_GDP3$ActAdvisoryEndTime[i] = EWR_GDP3$Derived.BgnDate.Time.UTC[i+1]
EWR_GDP3$ActAdvisoryDuration[i] =
difftime(EWR_GDP3$ActAdvisoryEndTime[i],EWR_GDP3$Derived.BgnDate.Time.UTC[i],un
its = "hours")
}
}

##delete first blank in the weather data
MCIC = substring(EWR_GDP3$MCIC,2, nchar(EWR_GDP3$MCIC))
Vis = substring(EWR_GDP3$Visibility,2, nchar(EWR_GDP3$Visibility))
Ceiling = substring(EWR_GDP3$Ceiling,2, nchar(EWR_GDP3$Ceiling))
TS = substring(EWR_GDP3$TS,2, nchar(EWR_GDP3$TS))
CW0422 = substring(EWR_GDP3$CW0422,2, nchar(EWR_GDP3$CW0422))
CW1129 = substring(EWR_GDP3$CW1129,2, nchar(EWR_GDP3$CW1129))
PC = substring(EWR_GDP3$PC,2, nchar(EWR_GDP3$PC))
RNSNIC= substring(EWR_GDP3$RNSNIC,2, nchar(EWR_GDP3$RNSNIC))

MCIC_obs = substring(EWR_GDP3$MCIC_obs,2, nchar(EWR_GDP3$MCIC_obs))
Vis_obs = substring(EWR_GDP3$Vis_obs,2, nchar(EWR_GDP3$Vis_obs))
Ceiling_obs = substring(EWR_GDP3$Ceiling_obs,2, nchar(EWR_GDP3$Ceiling_obs))
TS_obs = substring(EWR_GDP3$TS_obs,2, nchar(EWR_GDP3$TS_obs))
CW0422_obs = substring(EWR_GDP3$CW0422_obs,2, nchar(EWR_GDP3$CW0422_obs))
CW1129_obs = substring(EWR_GDP3$CW1129_obs,2, nchar(EWR_GDP3$CW1129_obs))
PC_obs = substring(EWR_GDP3$PC_obs,2, nchar(EWR_GDP3$PC_obs))
RNSNIC_obs = substring(EWR_GDP3$RNSNIC_obs,2, nchar(EWR_GDP3$RNSNIC_obs))

c = ncol(EWR_GDP3)
HrGDP <- EWR_GDP3[0,]
rowno = 0
```

```
for ( i in 1:nrow(EWR_GDP3)) {
H = ceiling(as.numeric(EWR_GDP3$ActAdvisoryDuration[i]))

MCIC1 = as.numeric(unlist(strsplit(as.character(MCIC[i]), split =" ")))
Vis1 = as.numeric(unlist(strsplit(as.character(Vis[i]), split =" ")))
Ceiling1 = as.numeric(unlist(strsplit(as.character(Ceiling[i]), split =" ")))
TS1 = as.numeric(unlist(strsplit(as.character(TS[i]), split =" ")))
CW04221 = as.numeric(unlist(strsplit(as.character(CW0422[i]), split =" ")))
CW11291 = as.numeric(unlist(strsplit(as.character(CW1129[i]), split =" ")))
PC1 = as.numeric(unlist(strsplit(as.character(PC[i]), split =" ")))
RNSNIC1 = as.numeric(unlist(strsplit(as.character(RNSNIC[i]), split =" ")))

PR1 = as.numeric(unlist(strsplit(as.character(EWR_GDP3$ProgramRateVct [i]), split ="/")))

MCIC1_obs = as.numeric(unlist(strsplit(as.character(MCIC_obs[i]), split =" ")))
Vis1_obs = as.numeric(unlist(strsplit(as.character(Vis_obs [i]), split =" ")))
Ceiling1_obs = as.numeric(unlist(strsplit(as.character(Ceiling_obs [i]), split =" ")))
TS1_obs = as.numeric(unlist(strsplit(as.character(TS_obs[i]), split =" ")))
CW04221_obs = as.numeric(unlist(strsplit(as.character(CW0422_obs[i]), split =" ")))
CW11291_obs = as.numeric(unlist(strsplit(as.character(CW1129_obs[i]), split =" ")))
PC1_obs = as.numeric(unlist(strsplit(as.character(PC_obs[i]), split =" ")))
RNSNIC1_obs = as.numeric(unlist(strsplit(as.character(RNSNIC_obs[i]), split =" ")))
if(H>0) {
for (h in 1:H) {
rowno =rowno+1
HrGDP[rowno,1:c] =EWR_GDP3[i,]

HrGDP$HourNo[rowno] = h

HrGDP$SchArrSchIn[rowno] = EWR_GDP3$SchArrSchIn[i]
HrGDP$ImpArrSchIn[rowno] = EWR_GDP3$ImpArrSchIn[i]

HrGDP$BgnTime[rowno] = as.character(EWR_GDP3$Derived.BgnDate.Time.UTC[i]+ (h-
1)*60*60)
 if(h==H) {
HrGDP$EndTime[rowno] <- as.character(EWR_GDP3$ActAdvisoryEndTime[i])
} else {
HrGDP$EndTime[rowno] =
as.character(EWR_GDP3$Derived.BgnDate.Time.UTC[i]+h*60*60)
}

HrGDP$MCIC_hr[rowno] = MCIC1[h]
HrGDP$Vis_hr[rowno] = Vis1[h]
HrGDP$Ceiling_hr[rowno] = Ceiling1[h]
HrGDP$TS_hr[rowno] = TS1[h]
```

```
HrGDP$CW0422_hr[rowno] = CW04221[h]
HrGDP$CW1129_hr[rowno] = CW11291[h]
HrGDP$PC_hr[rowno] = PC1[h]
HrGDP$RNSNIC_hr[rowno] = RNSNIC1[h]

HrGDP$PR_hr[rowno] = PR1[h]

HrGDP$MCIC_hr_obs[rowno] = MCIC1_obs[h]
HrGDP$Vis_hr_obs [rowno] = Vis1_obs [h]
HrGDP$Ceiling_hr_obs [rowno] = Ceiling1_obs [h]
HrGDP$TS_hr_obs[rowno] = TS1_obs[h]
HrGDP$CW0422_hr_obs[rowno] = CW04221_obs[h]
HrGDP$CW1129_hr_obs[rowno] = CW11291_obs[h]
HrGDP$PC_hr_obs[rowno] = PC1_obs[h]
HrGDP$RNSNIC_hr_obs[rowno] = RNSNIC1_obs[h]
}
}
}

Focus = which(HrGDP$AdvisoryType == "GDP CNX" & HrGDP$HourNo != 1)
HrGDP = HrGDP[-Focus,]

HrGDP$EarlyCancelTime[is.na(HrGDP$EarlyCancelTime)] = 0
HrGDP[HrGDP$AdvisoryType == "GDP CNX", c("Duration_Advisory", "PR_hr")] = 0

HrGDP$BgnTime = parse_date_time(HrGDP$BgnTime, guess_formats(HrGDP$BgnTime,
c("%Y-%m-%d %H:%M:%S", "%Y-%m-%d")), tz="America/New_York")
HrGDP$EndTime = parse_date_time(HrGDP$EndTime, guess_formats(HrGDP$EndTime,
c("%Y-%m-%d %H:%M:%S", "%Y-%m-%d")), tz="America/New_York")

HrGDP$StartMin= difftime(HrGDP$BgnTime, HrGDP$RootBgnTime, units = "mins")
HrGDP$EndTime[HrGDP$AdvisoryType == "GDP CNX"]=
HrGDP$BgnTime[HrGDP$AdvisoryType == "GDP CNX"]
HrGDP$EndMin= difftime(HrGDP$EndTime, HrGDP$RootBgnTime, units = "mins")
```

#Part 16 identify strong wind

```
HrGDP$CW0422_hr=as.numeric(HrGDP$CW0422_hr)
HrGDP$CW1129_hr=as.numeric(HrGDP$CW1129_hr)
HrGDP$CW0422_hr_obs =as.numeric(HrGDP$CW0422_hr_obs)
HrGDP$CW1129_hr_obs =as.numeric(HrGDP$CW1129_hr_obs)

HrGDP$CW0422_Str=0
HrGDP$CW1129_Str=0
HrGDP$CW0422_Str[HrGDP$CW0422_hr>=25]=1
HrGDP$CW1129_Str[HrGDP$CW1129_hr>=25]=1
```

```
HrGDP$CW0422_obs_Str=0
HrGDP$CW1129_obs_Str=0
HrGDP$CW0422_obs_Str[HrGDP$CW0422_hr_obs >=25]=1
HrGDP$CW1129_obs_Str[HrGDP$CW1129_hr_obs>=25]=1
```

#Part 17 GDP evolution data

```
HrGDP1= HrGDP

HrGDP1$RootNumber=as.numeric(HrGDP1$RootNumber)
HrGDP1$AdvisoryType=as.character(HrGDP1$AdvisoryType)
HrGDP1$StartMin = as.numeric(HrGDP1$StartMin)
HrGDP1$EndMin = as.numeric(HrGDP1$EndMin)

##add the root GDPs
EWR_GDP3$Is.RootAdvisory=as.character(EWR_GDP3$Is.RootAdvisory)
root = EWR_GDP3[EWR_GDP3$Is.RootAdvisory=="Yes",]

#delete GDPs with incomplete information
HrGDP1$CW0422=as.character(HrGDP1$CW0422)
HrGDP1$CW1129=as.character(HrGDP1$CW1129)
HrGDP1$MCIC=as.character(HrGDP1$MCIC)
HrGDP1$TS=as.character(HrGDP1$TS)
HrGDP1$PC=as.character(HrGDP1$PC)

focus=which(grepl("NA", HrGDP1$CW0422))
if(length(focus)>0) {
NANo=HrGDP1$RootNumber[focus]
HrGDP1 = HrGDP1[-focus,]
for(j in 1:length(NANo)) {
focus1 = which(root$RootNumber == NANo[j])
if(length(focus1)>0) {
root = root[-focus1,]
}
}
}
focus=which(grepl("NA", HrGDP1$CW1129))
if(length(focus)>0) {
NANo=HrGDP1$RootNumber[focus]
HrGDP1 = HrGDP1[-focus,]
for(j in 1:length(NANo)) {
focus1 = which(root$RootNumber == NANo[j])
if(length(focus1)>0) {
root = root[-focus1,]
}
}
```

```
}
focus=which(grepl("NA", HrGDP1$MCIC))
if(length(focus)>0) {
NANo=HrGDP1$RootNumber[focus]
HrGDP1 = HrGDP1[-focus,]
for(j in 1:length(NANo)) {
focus1 = which(root$RootNumber == NANo[j])
if(length(focus1)>0) {
root = root[-focus1,]
}
}
}
focus=which(grepl("NA", HrGDP1$PC))
if(length(focus)>0) {
NANo=HrGDP1$RootNumber[focus]
HrGDP1 = HrGDP1[-focus,]
for(j in 1:length(NANo)) {
focus1 = which(root$RootNumber == NANo[j])
if(length(focus1)>0) {
root = root[-focus1,]
}
}
}
focus=which(grepl("NA", HrGDP1$TS))
if(length(focus)>0) {
NANo=HrGDP1$RootNumber[focus]
HrGDP1 = HrGDP1[-focus,]
for(j in 1:length(NANo)) {
focus1 = which(root$RootNumber == NANo[j])
if(length(focus1)>0) {
root = root[-focus1,]
}
}
}

HrGDP2= HrGDP1
root1=root
J= nrow(HrGDP2)-1
HrGDP2$DiffSE=0
for( j in 1:J) {
if(HrGDP2$RootNumber[j]== HrGDP2$RootNumber[j+1]){
HrGDP2$DiffSE[j] = HrGDP2$EndMin[j]- HrGDP2$StartMin[j+1]
}
}
focus=which(HrGDP2$DiffSE!=0)
NANo=HrGDP2$RootNumber[focus]
```

```
for(j in 1:length(NANo)) {
focus1 = which(HrGDP2$RootNumber == NANo[j])
if(length(focus1)>0) {
HrGDP2 = HrGDP2[-focus1,]
}
focus2= which(root1$RootNumber == NANo[j])
if(length(focus2)>0) {
root1 = root1[-focus2,]
}
}

##delete GDPs with illogical times
focus=which(HrGDP2$EndMin<0 | HrGDP2$StartMin<0)
NANo=HrGDP2$RootNumber[focus]
for(j in 1:length(NANo)) {
focus1 = which(HrGDP2$RootNumber == NANo[j])
if(length(focus1)>0) {
HrGDP2 = HrGDP2[-focus1,]
}
focus2= which(root1$RootNumber == NANo[j])
if(length(focus2)>0) {
root1 = root1[-focus2,]
}
}
```

##add weather and flight information to the GDPs

```
maxtime=max(HrGDP2$EndMin)
MaxInterval=15
NumNewcol= floor(maxtime/MaxInterval)+1 #66
root2 = root1
```

###thunderstorms

```
OldTotCol=ncol(root2)
NewBgnCol=ncol(root2)+1
NewTotCol= OldTotCol + NumNewcol #192 = 126 +66
for(j in NewBgnCol:NewTotCol){
root2[,j]<-NA
IntNum= j- OldTotCol
colnames(root2)[j]<-paste(as.character(IntNum), "TS")
}
maxrt= max(HrGDP2$RootNumber)
for(k in 1:maxrt) {
sub1 = HrGDP2[HrGDP2$RootNumber == k,]
if(nrow(sub1)>0){
MaxEndTm = max(sub1$EndMin)
Int = MaxEndTm/(NumNewcol-1)
```

```
for(i in 1:NumNewcol){
x= (i-1)*Int
y=i*Int
sub2=sub1[which( y>=sub1$StartMin & x< sub1$EndMin),]
if(nrow(sub2)>0) {
sub2= sub2[order(sub2$TS_hr, decreasing = TRUE),]
root2[root2$RootNumber == k, OldTotCol +i]=sub2$TS_hr[1]
}else{
root2[root2$RootNumber == k, OldTotCol +i]= "/"
}}}}
```

###Precipitation

```
OldTotCol=ncol(root2)
NewBgnCol=ncol(root2)+1
NewTotCol= OldTotCol + NumNewcol #192 = 126 +66
for(j in NewBgnCol:NewTotCol){
root2[,j]<-NA
IntNum= j- OldTotCol
colnames(root2)[j]<-paste(as.character(IntNum), "PC")
}
maxrt= max(HrGDP2$RootNumber)
for(k in 1:maxrt) {
sub1 = HrGDP2[HrGDP2$RootNumber == k,]
if(nrow(sub1)>0){
MaxEndTm = max(sub1$EndMin)
Int = MaxEndTm/(NumNewcol-1)
for(i in 1:NumNewcol){
x= (i-1)*Int
y=i*Int
sub2=sub1[which( y>=sub1$StartMin & x< sub1$EndMin),] #GDP advisory
if(nrow(sub2)>0) {
sub2= sub2[order(sub2$PC_hr, decreasing = TRUE),]
root2[root2$RootNumber == k, OldTotCol +i]=sub2$PC_hr[1]
}else{
root2[root2$RootNumber == k, OldTotCol +i]= "/"
}}}}
```

###CW0422

```
OldTotCol=ncol(root2)
NewBgnCol=ncol(root2)+1
NewTotCol= OldTotCol + NumNewcol #192 = 126 +66
for(j in NewBgnCol:NewTotCol){
root2[,j]<-NA
IntNum= j- OldTotCol
colnames(root2)[j]<-paste(as.character(IntNum), "CW0422")
}
```

```
maxrt= max(HrGDP2$RootNumber)
for(k in 1:maxrt) {
sub1 = HrGDP2[HrGDP2$RootNumber == k,]
if(nrow(sub1)>0){
MaxEndTm = max(sub1$EndMin)
Int = MaxEndTm/(NumNewcol-1)
for(i in 1:NumNewcol){
x= (i-1)*Int
y=i*Int
sub2=sub1[which( y>=sub1$StartMin & x< sub1$EndMin),]
if(nrow(sub2)>0) {
sub2= sub2[order(sub2$CW0422_hr, decreasing = TRUE),]
root2[root2$RootNumber == k, OldTotCol +i]=sub2$CW0422_hr[1]
}else{
root2[root2$RootNumber == k, OldTotCol +i]= "/"
}}}}
```

###CW1129

```
OldTotCol=ncol(root2)
NewBgnCol=ncol(root2)+1
NewTotCol= OldTotCol + NumNewcol #192 = 126 +66
for(j in NewBgnCol:NewTotCol){
root2[,j]<-NA
IntNum= j- OldTotCol
colnames(root2)[j]<-paste(as.character(IntNum), "CW1129")
}
maxrt= max(HrGDP2$RootNumber)
for(k in 1:maxrt) {
sub1 = HrGDP2[HrGDP2$RootNumber == k,]
if(nrow(sub1)>0){
MaxEndTm = max(sub1$EndMin)
Int = MaxEndTm/(NumNewcol-1)
for(i in 1:NumNewcol){
x= (i-1)*Int
y=i*Int
sub2=sub1[which( y>=sub1$StartMin & x< sub1$EndMin),] #GDP advisory
if(nrow(sub2)>0) {
sub2= sub2[order(sub2$CW1129_hr, decreasing = TRUE),]
root2[root2$RootNumber == k, OldTotCol +i]=sub2$CW1129_hr[1]
}else{
root2[root2$RootNumber == k, OldTotCol +i]= "/"
}}}}
```

###Ceiling

```
OldTotCol=ncol(root2)
NewBgnCol=ncol(root2)+1
```

```
NewTotCol= OldTotCol + NumNewcol #192 = 126 +66
for(j in NewBgnCol:NewTotCol){
root2[,j]<-NA
IntNum= j- OldTotCol
colnames(root2)[j]<-paste(as.character(IntNum), "Ceiling")
}
maxrt= max(HrGDP2$RootNumber)
for(k in 1:maxrt) {
sub1 = HrGDP2[HrGDP2$RootNumber == k,]
if(nrow(sub1)>0){
MaxEndTm = max(sub1$EndMin)
Int = MaxEndTm/(NumNewcol-1)
for(i in 1:NumNewcol){
x= (i-1)*Int
y=i*Int
sub2=sub1[which( y>=sub1$StartMin & x< sub1$EndMin),]
if(nrow(sub2)>0) {
sub2= sub2[order(sub2$Ceiling_hr, decreasing = FALSE),]
root2[root2$RootNumber == k, OldTotCol +i]=sub2$Ceiling_hr[1]
}else{
root2[root2$RootNumber == k, OldTotCol +i]= "/"
}}}}
```

###Visibility

```
OldTotCol=ncol(root2)
NewBgnCol=ncol(root2)+1
NewTotCol= OldTotCol + NumNewcol #192 = 126 +66
for(j in NewBgnCol:NewTotCol){
root2[,j]<-NA
IntNum= j- OldTotCol
colnames(root2)[j]<-paste(as.character(IntNum), "Vis")
}
maxrt= max(HrGDP2$RootNumber)
for(k in 1:maxrt) {
sub1 = HrGDP2[HrGDP2$RootNumber == k,]
if(nrow(sub1)>0){
MaxEndTm = max(sub1$EndMin)
Int = MaxEndTm/(NumNewcol-1)
for(i in 1:NumNewcol){
x= (i-1)*Int
y=i*Int
sub2=sub1[which( y>=sub1$StartMin & x< sub1$EndMin),] #GDP advisory
if(nrow(sub2)>0) {
sub2= sub2[order(sub2$Vis_hr, decreasing = FALSE),]
root2[root2$RootNumber == k, OldTotCol +i]=sub2$Vis_hr[1]
}else{
```

```
root2[root2$RootNumber == k, OldTotCol +i]= "/"
}}}}
```

###Program rate

```
OldTotCol=ncol(root2)
NewBgnCol=ncol(root2)+1
NewTotCol= OldTotCol + NumNewcol #192 = 126 +66
for(j in NewBgnCol:NewTotCol){
root2[,j]<-NA
IntNum= j- OldTotCol
colnames(root2)[j]<-paste(as.character(IntNum), "PR")
}
maxrt= max(HrGDP2$RootNumber)
for(k in 1:maxrt) {
sub1 = HrGDP2[HrGDP2$RootNumber == k,]
if(nrow(sub1)>0){
MaxEndTm = max(sub1$EndMin)
Int = MaxEndTm/(NumNewcol-1)
for(i in 1:NumNewcol){
x= (i-1)*Int
y=i*Int
sub2=sub1[which( y>=sub1$StartMin & x< sub1$EndMin),] #GDP advisory
if(nrow(sub2)>0) {
sub2= sub2[order(sub2$PR_hr, decreasing = FALSE),]
root2[root2$RootNumber == k, OldTotCol +i]=sub2$PR_hr[1]
}else{
root2[root2$RootNumber == k, OldTotCol +i]= "/"
}}}}
```

##SchArrSchIn

```
OldTotCol=ncol(root2)
NewBgnCol=ncol(root2)+1
NewTotCol= OldTotCol + NumNewcol #192 = 126 +66
for(j in NewBgnCol:NewTotCol){
root2[,j]<-NA
IntNum= j- OldTotCol
colnames(root2)[j]<-paste(as.character(IntNum), "SchArrSchIn")
}
maxrt= max(HrGDP2$RootNumber)
for(k in 1:maxrt) {
sub1 = HrGDP2[HrGDP2$RootNumber == k,]
if(nrow(sub1)>0){
MaxEndTm = max(sub1$EndMin)
Int = MaxEndTm/(NumNewcol-1)
for(i in 1:NumNewcol){
x= (i-1)*Int
```

```
y=i*Int
sub2=sub1[which( y>=sub1$StartMin & x< sub1$EndMin),] #GDP advisory
if(nrow(sub2)>0) {
sub2= sub2[order(sub2$SchArrSchIn, decreasing = TRUE),]
root2[root2$RootNumber == k, OldTotCol +i]=sub2$SchArrSchIn[1]
}else{
root2[root2$RootNumber == k, OldTotCol +i]= "/"
}}}}
```

##ImpArrSchIn

```
OldTotCol=ncol(root2)
NewBgnCol=ncol(root2)+1
NewTotCol= OldTotCol + NumNewcol #192 = 126 +66
for(j in NewBgnCol:NewTotCol){
root2[,j]<-NA
IntNum= j- OldTotCol
colnames(root2)[j]<-paste(as.character(IntNum), "ImpArrSchIn")
}
maxrt= max(HrGDP2$RootNumber)
for(k in 1:maxrt) {
sub1 = HrGDP2[HrGDP2$RootNumber == k,]
if(nrow(sub1)>0){
MaxEndTm = max(sub1$EndMin)
Int = MaxEndTm/(NumNewcol-1)
for(i in 1:NumNewcol){
x= (i-1)*Int
y=i*Int
sub2=sub1[which( y>=sub1$StartMin & x< sub1$EndMin),] #GDP advisory
if(nrow(sub2)>0) {
sub2= sub2[order(sub2$ImpArrSchIn, decreasing = TRUE),]
root2[root2$RootNumber == k, OldTotCol +i]=sub2$ImpArrSchIn[1]
}else{
root2[root2$RootNumber == k, OldTotCol +i]= "/"
}}}}
```

##Duration

```
OldTotCol=ncol(root2)
NewBgnCol=ncol(root2)+1
NewTotCol= OldTotCol + NumNewcol #192 = 126 +66
for(j in NewBgnCol:NewTotCol){
root2[,j]<-NA
IntNum= j- OldTotCol
colnames(root2)[j]<-paste(as.character(IntNum), "PlanDuration")
}
maxrt= max(HrGDP2$RootNumber)
for(k in 1:maxrt) {
sub1 = HrGDP2[HrGDP2$RootNumber == k,]
```

```
if(nrow(sub1)>0){
MaxEndTm = max(sub1$EndMin)
Int = MaxEndTm/(NumNewcol-1)
for(i in 1:NumNewcol){
x= (i-1)*Int
y=i*Int
sub2=sub1[which( y>=sub1$StartMin & x< sub1$EndMin),] #GDP advisory 时间与 15 间隔时间有交集，取最大 SchArr
if(nrow(sub2)>0) {
sub2= sub2[order(sub2$Duration_Initiative, decreasing = TRUE),]
root2[root2$RootNumber == k, OldTotCol +i]=sub2$Duration_Initiative [1]
}else{
root2[root2$RootNumber == k, OldTotCol +i]= "/"
}}}}

EWR_GDP4 = EWR_GDP3[EWR_GDP3$RootNumber %in% HrGDP2$RootNumber,]

#Part 18 Evaluation
## get Ground Delays, Total Delays, Planned Delays, Actual Delays from IF
EWRIF = read.csv( "EWRIF-fixed.csv")

EWR_GDP4$Exempt.Dep.Facilities = as.character(EWR_GDP4$Exempt.Dep.Facilities)
EWR_GDP4$AdvisoryType =as.character(EWR_GDP4$AdvisoryType)
EWR_GDP4$Dep.Scope = as.character(EWR_GDP4$Dep.Scope)

EWRIF$Mile = as.numeric(EWRIF$Mile)
EWRIF$DEP_LOCID=as.character(EWRIF$DEP_LOCID)
EWRIF$Country=as.character(EWRIF$Country)

##Ground and G+Air delay
EWRIF$FpIn = as.character(paste(EWRIF$ARR_YYYY, EWRIF$ARR_MM,
as.character(EWRIF$ARR_DAY), as.character (EWRIF$FPINTM)))
EWRIF$FpIn = strptime(EWRIF$FpIn, "%Y %m %d %H:%M", tz ="America/New_York")

EWRIF$SchIn = as.character(paste(EWRIF$ARR_YYYY, EWRIF$ARR_MM,
as.character(EWRIF$ARR_DAY), as.character (EWRIF$SCHINTM)))
EWRIF$SchIn = strptime(EWRIF$SchIn, "%Y %m %d %H:%M", tz ="America/New_York")

EWRIF$DLAP = as.numeric(EWRIF$DLASCHARR) - as.numeric(EWRIF$DLAFPARR)
EWRIF$DLAP[EWRIF$DLAP<0] = 0

IFCA= EWRIF[EWRIF$Country == "CA",]
IFUS= EWRIF[EWRIF$Country == "US",]
IFINT= EWRIF[EWRIF$Country == "INT",]

for (i in 1:nrow(EWR_GDP4)) {
```

```
if (EWR_GDP4$AdvisoryType[i] == "GDP") {

##CA flights
###Spatial
rows<-sapply(IFCA$DEP_LOCID, function(x) grepl(x,
EWR_GDP4$Canadian.Dep.Arpts.Included[i]))
IFGDPCA = IFCA[rows,]
###Temporal
if(EWR_GDP4$RootNumber[i+1] == EWR_GDP4$RootNumber[i] ) {
mint = min(EWR_GDP4$Derived.BgnDate.Time.UTC[i+1],
EWR_GDP4$Derived.EndDate.Time.UTC[i])
row <-which(IFGDPCA$SchIn>=EWR_GDP4$Derived.BgnDate.Time.UTC[i] &
IFGDPCA$SchIn <=mint)} else {
row <-which(IFGDPCA$SchIn>=EWR_GDP4$Derived.BgnDate.Time.UTC[i] &
IFGDPCA$SchIn <= EWR_GDP4$Derived.EndDate.Time.UTC[i])
}
CA = IFGDPCA[row,]
CAGD = sum(CA$DLASCHOFF)
CATD = sum(CA$DLASCHARR)
CADLAA=sum(CA$DLASCHARR)

##US flights
### tick out exempted flights
if (EWR_GDP4$Exempt.Dep.Facilities[i]!= "-") {
rows<-sapply(IFUS$DEP_LOCID, function(x) !grepl(x,
EWR_GDP4$Exempt.Dep.Facilities[i]))
IFUS1 = IFUS[rows,]
rows<- sapply(IFUS1$ARTCC, function(x) !grepl(x, EWR_GDP4$Exempt.Dep.Facilities[i]) )
IFGDPUS = IFUS1[rows,]
} else{
IFGDPUS= IFUS
}
### spatial scope
if (!grepl("ALL",EWR_GDP4$Dep.Scope[i]) ){
if(EWR_GDP4$DepScopeType[i] == "ARTCC")
{
IFGDPUS1=IFGDPUS[sapply(IFGDPUS$ARTCC, function(x) grepl(x,
EWR_GDP4$Dep.Scope[i] )),]
}
if(EWR_GDP4$DepScopeType[i] == "Radius")
{
IFGDPUS1 = IFGDPUS[IFGDPUS$Mile <= as.numeric(EWR_GDP4$Dep.Scope[i]),]
}
}else {
IFGDPUS1= IFGDPUS}
### temporal scope
```

```
if(EWR_GDP4$RootNumber[i+1] == EWR_GDP4$RootNumber[i] ) {
mint = min(EWR_GDP4$Derived.BgnDate.Time.UTC[i+1],
EWR_GDP4$Derived.EndDate.Time.UTC[i])
row <-which(IFGDPUS1$SchIn>=EWR_GDP4$Derived.BgnDate.Time.UTC[i] &
IFGDPUS1$SchIn<=mint)} else {
row <-which(IFGDPUS1$SchIn>=EWR_GDP4$Derived.BgnDate.Time.UTC[i] &
IFGDPUS1$SchIn<=EWR_GDP4$Derived.EndDate.Time.UTC[i])
}

US = IFGDPUS1[row,]
USGD = sum(US$DLASCHOFF)
USTD = sum(US$DLASCHARR)
USDLAA=sum(US$DLASCHARR)

##Total Delay.
EWR_GDP4$GD[i] = as.numeric(CAGD)+ as.numeric(USGD)
EWR_GDP4$TD[i] = as.numeric(CATD)+ as.numeric(USTD)
##Total Delay.
EWR_GDP4$DLAA[i] = as.numeric(CADLAA)+ as.numeric(USDLAA)
}
}
```

##Efficiency

```
for( j in root2$RootNumber) {
x= EWR_GDP4[EWR_GDP4$RootNumber ==j & EWR_GDP4$AdvisoryType =="GDP" ,]
root2$GD[root2$RootNumber ==j] = sum(x$GD)
root2$TD[root2$RootNumber ==j] = sum(x$TD)
root2$Eff[root2$RootNumber ==j] = root2$GD[root2$RootNumber ==j]/
root2$TD[root2$RootNumber ==j]
}

for (i in 1:nrow(root2)) {
##CA flights
###Spatial
rows<-sapply(IFCA$DEP_LOCID, function(x) grepl(x, root2$Canadian.Dep.Arpts.Included[i]))
IFGDPCA = IFCA[rows,]
###Temporal
row <-which(IFGDPCA$SchIn>=root2$Derived.BgnDate.Time.UTC[i] & IFGDPCA$SchIn <=
root2$Derived.EndDate.Time.UTC[i])
CA = IFGDPCA[row,]
CADLAP= sum(CA$DLAP)

##US flights
### tick out exempted flights
if (root2$Exempt.Dep.Facilities[i]!= "-") {
rows<-sapply(IFUS$DEP_LOCID, function(x) !grepl(x, root2$Exempt.Dep.Facilities[i]))
```

```
IFUS1 = IFUS[rows,]
rows<- sapply(IFUS1$ARTCC, function(x) !grepl(x, root2$Exempt.Dep.Facilities[i]) )
IFGDPUS = IFUS1[rows,]
} else{
IFGDPUS= IFUS
}
### spatial scope
if (!grepl("ALL",root2$Dep.Scope[i]) ){
if(root2$DepScopeType[i] == "ARTCC")
{
IFGDPUS1=IFGDPUS[sapply(IFGDPUS$ARTCC, function(x) grepl(x, root2$Dep.Scope[i] )),]
}
if(root2$DepScopeType[i] == "Radius")
{
IFGDPUS1 = IFGDPUS[IFGDPUS$Mile <= as.numeric(root2$Dep.Scope[i]),]
}
}else {
IFGDPUS1= IFGDPUS}
### temporal scope
row <-which(IFGDPUS1$SchIn>=root2$Derived.BgnDate.Time.UTC[i] &
IFGDPUS1$SchIn<=root2$Derived.EndDate.Time.UTC[i])
USDLAP= sum(US$DLAP)

##Total Delay.
root2$DLAP[i] = as.numeric(CADLAP)+ as.numeric(USDLAP)
}
```

##Predictability

```
for( j in root2$RootNumber) {
x= EWR_GDP4[EWR_GDP4$RootNumber ==j & EWR_GDP4$AdvisoryType =="GDP" ,]
A = sum(x$DLAA)
root2$DLAA[root2$RootNumber ==j] = A
}

for(j in 1:nrow(root2)) {
root2$Pred[j] = min(root2$DLAA[j], root2$DLAP[j])/ max(root2$DLAA[j], root2$DLAP[j])
}
```

##Actual Arrivals

```
EWRIF$ARR_YYYY = as.character(substr(EWRIF$ARR_YYYYMM,1,4))
EWRIF$ARR_MM = as.character(substr(EWRIF$ARR_YYYYMM,5,6))

EWRIF$ActIn = as.character(paste(EWRIF$ARR_YYYY, EWRIF$ARR_MM,
as.character(EWRIF$ARR_DAY), as.character (EWRIF$ACTINTM)))
EWRIF$ActIn = strptime(EWRIF$ActIn, "%Y %m %d %H:%M", tz ="America/New_York")
for(i in 1:nrow(EWR_GDP4)) {
```

```
if(EWR_GDP4$RootNumber[i+1] == EWR_GDP4$RootNumber[i] ) {
mint = min(EWR_GDP4$Derived.BgnDate.Time.UTC[i+1],
EWR_GDP4$Derived.EndDate.Time.UTC[i])
EWR_GDP4$ActArr[i] =
length(which(EWRIF$ActIn >=EWR_GDP4$Derived.BgnDate.Time.UTC[i] & EWRIF$ActIn
<=mint)) } else {
EWR_GDP4$ActArr[i] =
length(which(EWRIF$ActIn >=EWR_GDP4$Derived.BgnDate.Time.UTC[i] & EWRIF$ActIn
<= EWR_GDP4$Derived.EndDate.Time.UTC[i]))
}
}
```

##Capacity Utilization

```
for ( j in 1:nrow(EWR_GDP4)) {
if(!grepl("-", EWR_GDP4$ProgramRate[j])) {
PR = as.numeric(unlist(strsplit(as.character(EWR_GDP4$ProgramRate[j]),split = "/")))
EWR_GDP4$AvePR[j]= mean(sum(PR))
}
}
EWR_GDP4$TotCap= EWR_GDP4$AvePR * EWR_GDP4$ActAdvisoryDuration

for( j in root2$RootNumber) {
x= EWR_GDP4[EWR_GDP4$RootNumber ==j & EWR_GDP4$AdvisoryType =="GDP" ,]
ActArr= sum(x$ActArr)
TotCap = sum(x$TotCap)
root2$CU[root2$RootNumber ==j] = ActArr/TotCap
}
```

##Paste labels to Hrs and Advisory data

```
rootevl =read.csv("EWR_GDP18-CU.csv")
rootevl$Label = allweather12[,598]
rootevl$Label[rootevl$Label==6] =3
for(j in rootevl$RootNumber) {
HrGDP2$Label[HrGDP2$RootNumber ==j] = rootevl$Label[rootevl$RootNumber == j]
}
for(j in rootevl$RootNumber) {
EWR_GDP3$Label[EWR_GDP3$RootNumber ==j] = rootevl$Label[rootevl$RootNumber ==
j]}
```

##Calculate average of each variable

```
StatAna = data.frame(matrix(vector(), 12, 9,
            dimnames=list(c(), c("TSAve","PCAve","CW0422Ave", "CW1129Ave",
"CeilingAve", "VisAve","PRAve","ScopeAve","DurationAve"))),
            stringsAsFactors=F)
###Mean
HrGDP3 = HrGDP2[!is.na(HrGDP2$Label),]
```

```
for( j in 1:12) {
x = HrGDP3[HrGDP3$Label == j,]
StatAna$TSAve[j] = mean(x$TS_hr)
StatAna$PCAve[j] = mean(x$PC_hr)
StatAna$CW0422Ave[j] = mean(x$CW0422_hr)
StatAna$CW1129Ave[j] = mean(x$CW1129_hr)
StatAna$VisAve[j] = mean(x$Vis_hr)
StatAna$CeilingAve[j] = mean(x$CW0422_hr)
StatAna$PRAve[j] = mean(x$PR_hr)
}
EWR_GDP4= EWR_GDP3[!is.na(EWR_GDP3$Label),]
for( j in 1:12) {
x = EWR_GDP4[EWR_GDP4$Label == j,]
StatAna$ScopeAve[j] = mean(x$ImpArrSchIn)
StatAna$DurationAve[j] = mean(x$Duration_Initiative)
}
```

##Variance of the each varirable

```
library(matrixStats)
STAT<- allweather12_P
N = ncol(allweather12_P)+1
for(j in 1:9) {
j1=(j-1)*66+1
j2=j1+64
stat = as.matrix(allweather12_P[,j1:j2])
STAT[,j] = rowVars(stat)
colnames(STAT)[j] =paste("var-",j)
}
for( j in 1:12) {
x = STAT[STAT$PM12 == j,]
StatAna$TSVar[j] = mean(x$var-1)
StatAna$PCVar[j] = mean(x$var-2)
StatAna$CW0422Var[j] = mean(x$var-3)
StatAna$CW1129Var[j] = mean(x$var-4)
StatAna$VisVar[j] = mean(x$var-5)
StatAna$CeilingVar[j] = mean(x$var-6)
}
colnames(STAT)[1:6] = c("TSVar","PCVar","CW0422Var", "CW1129Var", "CeilingVar",
"VisVar")
StatAna= cbind(StatAna, STAT[.1:6] )
```

Appendix B R Code for GDP clustering

```
#Part 1 import target parameters of root GDPs
Dim = root2[,128:721]
norm.data=Dim
norm.data[] = lapply(norm.data, function(x) as.numeric(as.character(x)))

#Part 2 normalize data for dimension reduction
min.max.norm <- function(x){
   (x-min1)/(max1-min1)
}

que=c(1:2,5:9)
for(n in que) {
i1 = 66*(n-1)+1
i2=i1+65
max1 = max(na.omit(unlist(norm.data[,i1:i2])))
min1 = min(na.omit(unlist(norm.data[,i1:i2])))
for(j in i1:i2) {
focus = which(!is.na(norm.data[,j]))
norm.data[focus,j] = min.max.norm(norm.data[focus,j])
}
}

n1=3
n2=4
i1 = 66*(n1-1)+1 #133
i2=66*(n2-1)+1+65 #264
max1 = max(na.omit(unlist(norm.data[,i1:i2])))
min1 = min(na.omit(unlist(norm.data[,i1:i2])))
for(j in i1:i2) {
focus = which(!is.na(norm.data[,j]))
norm.data[focus,j] = min.max.norm(norm.data[focus,j])
}

for(j in 1:ncol(norm.data)) {
focus = which(is.na(norm.data[,j]))
norm.data[focus,j] = 0
}

#Part 3 Autoencoders for dimension reduction (h2o platform)
delNo = numeric()
for(j in 1:9) {
delNo[j]=66*j}
norm.data1 = norm.data[,-delNo]
```

```
library(h2o)
localH2O <- h2o.init(ip = "localhost", port = 54321)

mfile = "C:\\Users\\Qian Fu\\Documents\\R-TMI\\new-feb\\Dim1.csv"
# mfile = "C:\\Users\\Kexin\\Documents\\Dim1.csv"
mydata = h2o.importFile(path = mfile)

set.seed(1)
NN_model <- h2o.deeplearning(
x = 2:586,
training_frame = mydata,
hidden = c(300,2,200),
epochs = 100,
activation = "Tanh",
autoencoder = TRUE,
export_weights_and_biases=T
)

Weights1<-h2o.weights(NN_model,matrix_id = 1)
Weights2<-h2o.weights(NN_model,matrix_id = 2)

train_supervised_features = h2o.deepfeatures(NN_model, mydata, layer=2)
plotdata = as.data.frame(train_supervised_features)

# Part 3 Clustering analysis
library(factoextra)
library(cluster)

##choose k
###silhouette
fviz_nbclust(plotdata, kmeans, method = "silhouette", k.max = 20)
fviz_nbclust(plotdata, pam, method = "silhouette", k.max = 20)
fviz_nbclust(plotdata, hcut, method = "silhouette", hc_method = "complete", k.max = 20)
###Gap
set.seed(123)
gap_stat <- clusGap(plotdata, FUN = kmeans, nstart = 25, K.max = 20, B = 50)
fviz_gap_stat(gap_stat)
set.seed(123)
gap_stat <- clusGap(plotdata, FUN = pam, K.max = 20, B = 50)
fviz_gap_stat(gap_stat)
set.seed(123)
gap_stat <- clusGap(plotdata, FUN = hcut, K.max = 20, B = 50)
fviz_gap_stat(gap_stat)

##Plot Clustering results
```

```
###PAM
pamc=pam(plotdata,10)
pamc1 = cbind(plotdata, clusterNum =pamc$clustering)
plot(pamc1[,1:2],col=pamc1$clusterNum, main ="PM, k=10" )
###HC
hc <- hclust(dist(plotdata), method="ward.D")
x1=cbind(plotdata, cluster = as.numeric(as.character(cutree(hc,k=10))))
plot (x1[,1:2],col=x1$cluster, main = "HC, k=10")
###k-means
cl=kmeans(plotdata,10)
dim = cbind(plotdata, clusterNum =cl$cluster)
plot(dim[,1:2],col=dim$clusterNum, main ="KM, k=10" )
```

##Visual images (using PAM)

```
allweather10=cbind(norm.data,plotdata, PM10= pamc$clustering)
allweather10_P= allweather10[order(allweather10$PM10, decreasing=FALSE),]
usr <- par( "usr" )
par( mfrow = c(10,10), mai = c(0,0,0,0))

par( mfrow = c(1,1))
plot.new()
par( mfrow = c(10,10), mai = c(0,0,0,0))
for(i in 1:nrow(allweather10_P)){
y = as.matrix(allweather10_P[i, 1:594])
dim(y) = c(66, 9)
image(y[,ncol(y):1],axes = FALSE, col = gray(255:0 / 255))
box(lty = 'solid', col = 'red')
text( usr[ 2 ], usr[ 4 ], allweather12_P[i,598],    adj = c( 2, 1 ), col = "blue" )}
```