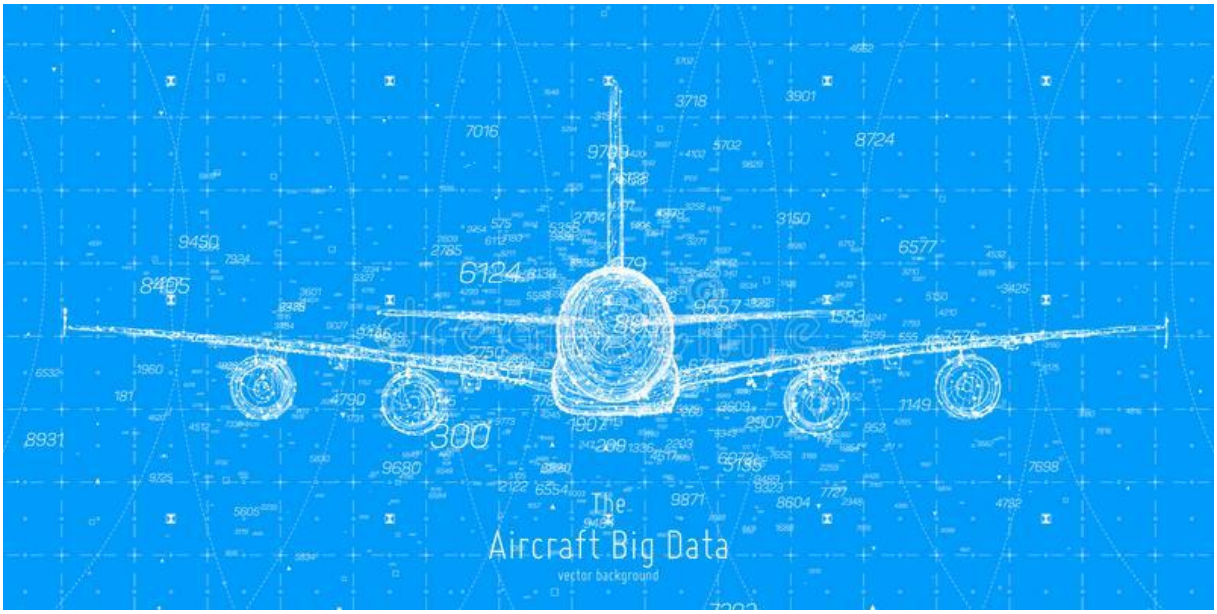
Project Big Data, Business Intelligence and NoSQL



Teamleden:

Thomas Billiet

Ishan Ameel

Sven Depickere

Jonas Anseel

1. Inhoud

Voor meer info over onze voortgang of code zie Github:

https://github.com/DepickereSven/Datatonic-challenge

2. Preprocessing van de dataset

We hebben onze code in verschillende stappen opgesplitst namelijk:

- Data laden
- Opkuis functies
- Converteer de militaire tijd
- Voeg tijdzones toe
- Zet alle tijden op UTC
- Bereken de vliegtijd
- Bereken de afstand
- Bereken de snelheid
- Verwijder te korte en te lange vluchten
- Uitvoeren van de opkuis-functies

2.1 Data laden

```
IMPORT_PATH = '../Initial_Data/Unzipped'
IMPORT_ALL_PATH = IMPORT_PATH + '/*'

import pandas as pd
import numpy as np
import glob as glob

def readAllFiles():
    files = glob.glob(IMPORT_ALL_PATH)
    frames = []

    for file in files:
        df = pd.read_csv(file, index_col = 0)
        frames.append(df)

    return pd.concat(frames)

def readOneFile(url):
    return pd.read_csv(url, index_col = 0)

df = readAllFiles()
```

Hier lezen we de data van de originele csv files, die zich in de map */Initial_Data/Unzipped* bevinden, in als 1 panda dataframe.

Om te beginnen hebben we **11.401.196** records.

2.2 Opkuis functies

```
import pandas as pd
import numpy as np
import glob as glob
from math import *

usaStates = [
    "AL",
    "AK",
    "AZ",
    "AR",
    "CA",
    "CO",
    "CT",
    "DE",
    "FL",
    "GA",
    "HI",
    "ID",
    "IL",
    "IN",
    "IA",
    "KS",
    "KY",
    "LA",
    "ME",
    "MD",
    "MA",
    "MI",
    "MN",
    "MS",
    "MO",
    "MT",
    "NE",
    "NV",
    "NH",
    "NJ",
    "NM",
    "NY",
    "NC",
    "ND",
    "OH",
    "OK",
    "OR",
    "PA",
    "RI",
    "SC",
    "SD",
    "TN",
    "TX",
    "UT",
    "VT",
    "VA",
    "WA",
    "WV",
    "WI",
    "WY",
]

def deleteWrongStates(df):
    print("Aantal records:", len(df))

    for el in df.departure_state.unique():
        if(el not in usaStates):
```

```
            df = df.drop(df[df['departure_state'] == el].index)
    print("Aantal records na verwijderen foute vertrek staat:", len(df))

    for el in df.arrival_state.unique():
        if(el not in usaStates):
            df = df.drop(df[df['arrival_state'] == el].index)
    print("Aantal records na verwijderen foute aankomst staat:", len(df))

    return df

def convertColumnTypes(df):
    df.departure_schedule = df.departure_schedule.astype(int)
    df.departure_delay = df.departure_delay.astype(float)
    df.arrival_schedule = df.arrival_schedule.astype(int)
    df.arrival_delay = df.arrival_delay.astype(float)
    df.arrival_actual = df.arrival_actual.astype(int)
    df.departure_actual = df.departure_actual.astype(int)
    return df

def dropMoreAdvancedDuplicates(df):
    df = df.groupby(['date', 'airline', 'airline_code', 'departure_airport', 'departure_state', 'departure_lat', 'departure_lon', 'departure_schedule', 'arrival_airport', 'arrival_state', 'arrival_lat', 'arrival_lon', 'arrival_schedule']).mean().reset_index()
    df = convertColumnTypes(df)
    if 'index' in df.columns:
        df = df.drop(['index'], axis=1) #remove old index
    return df
```

De Verenigde Staten van Amerika telt 50 staten. In de data dat we gekregen hebben bevonden zich 3 foutieve, niet officiële staten. Deze hebben we verwijderd aangezien ze niet in deze dataset thuishoren.

De foutieve staten zijn:

- TT (Trust Territory of the Pacific Islands):
  - bestaat niet meer sinds 1986
  - bestaat nu uit:
    - Republic of the Marshall Islands
    - Federated States of Micronesia
    - Republic of Palau
    - Commenwealth of the Northern Mariana Islands
- PR (Puerto Rico):
  - is een territorium
- VI (U.S Virgin Islands) :
  - is een territorium

Er zijn **56 413** vluchten die vertrekken vanuit staten die geen officiële staten zijn.

Er zijn **53 184** vluchten die landen in staten die geen officiële staten zijn.

We veranderen ook de kolommen naar het gepaste type omdat de verdere verwerking makkelijker zou gaan.

Daarna controleren we op verborgen dubbels. Een verborgen dubbel komt voor indien date, airline, airline_code, departure_airport, departure_state, departure_lat, datarture_lon, departure_schedule, arrival_airport, arrival_state, arrival_lat, arrival_lon en arrival_schedule hetzelfde zijn van meerdere

records, maar bijvoorbeeld enkel de vertraging verschilt. In dat geval beschouwen we de records als dezelfde vlucht en nemen we het gemiddelde van de vertraging.

We controleren ook nog of er records zijn die een vertrektijd hebben die vroeger is dat de aankomsttijd. Omdat de aankomstdatum niet bijgehouden wordt, en we enkel de tijd ter beschikking hebben kan je niet zeker zijn of het om een foutief record gaat of dat de vlucht de volgende dag land. Als extra controle kijken we of de geplande vliegtijd meer dan x aantal minuten te snel is. We hebben hier geen extreme waardes gevonden en daarom geen extra records verwijderd.

2.3 Converteer de militaire tijd

```
1.  import math
2.  from datetime import datetime
3.
4.  def getTotalMinutes(time):
5.      time_str = str(time).rjust(4, "0")
6.      hours = time_str[:-2]
7.      hours = int(hours)
8.      if hours == 24:
9.          hours = 0
10.     minutes = int(time_str[-2:])
11.     return datetime(2010, 1, 1, hours, minutes, 0)
12.
13. def convertMilitaryTime(df):
14.     df["departure_schedule"] = df["departure_schedule"].map(getTotalMinutes)
15.     df["arrival_schedule"] = df["arrival_schedule"].map(getTotalMinutes)
16.     return df
```

De dataset bevat vertrek en aankomsttijden in militair formaat (bv. 615 = 06:15). Om met deze tijden te kunnen werken in Python moeten we ze omzetten naar een “datetime” object.

2.4 Voeg tijdzones toe

```
# UTC offset by state
timezones = {
    "AL" :-6,
    "AK" :-9,
    "AZ" :-7,
    "AR" :-6,
    "CA" :-8,
    "CO" :-7,
    "CT" :-5,
    "DE" :-5,
    "FL" :-5,
    "GA" :-5,
    "HI" :-10,
    "ID" :-7,
    "IL" :-6,
    "IN" :-5,
    "IA" : -6,
    "KS" : -6,
    "KY" : -6,
    "LA" : -6,
    "ME" : -5,
    "MD" : -5,
    "MA" : -5,
    "MI" : -5,
    "MN" : -6,
    "MS" : -6,
    "MO" : -6,
    "MT" : -7,
    "NE" : -6,
    "NV" : -8,
    "NH" : -5,
    "NJ" : -5,
    "NM" : -7,
    "NY" : -5,
    "NC" : -5,
    "ND" : -6,
    "OH" : -5,
    "OK" : -6,
    "OR" : -8,
    "PA" : -5,
    "RI" : -5,
    "SC" : -5,
    "SD" : -6,
    "TN" : -6,
    "UT": -7,
    "TX": -6,
    "VT": -5,
    "VA": -5,
    "WA": -8,
    "DC": -5,
    "WV": -5,
    "WI": -6,
    "WY": -5
}

def addTimezones(df):
    df['arrival_tz'] = df['arrival_state'].map(lambda state: timezones[state])
    df['departure_tz'] = df['departure_state'].map(lambda state: timezones[state])
    return df
```

Alle tijden in de dataset zijn lokaal. Dus om ze bruikbaar te maken moeten we ze omzetten naar UTC-tijden. Hier voegen we de afwijking van UTC (in uren) toe aan elke vlucht.

2.5 Zet alle tijden op UTC

```
from datetime import timedelta

def timeToUTC(schedule, tz):
    tdelta = timedelta(hours=abs(tz))
    if tz > 0:
        schedule -= tdelta
    else:
        schedule += tdelta
    return schedule

def convertLocalToUTC(df):
    arrival_schedules = []
    departure_schedules = []
    for i, row in df.iterrows():
        arrival_schedule = row["arrival_schedule"]
        arrival_tz = row["arrival_tz"]
        departure_schedule = row["departure_schedule"]
        departure_tz = row["departure_tz"]

        arrival_schedules.append(timeToUTC(arrival_schedule, arrival_tz))
        departure_schedules.append(timeToUTC(departure_schedule, departure_tz))

    df["arrival_schedule"] = arrival_schedules
    df["departure_schedule"] = departure_schedules
    return df
```

Nu we de tijdzones hebben kunnen we de tijden omzetten naar UTC.

2.6 Bereken de vliegtijd

```
from datetime import timedelta

def timeToUTC(schedule, tz):
    tdelta = timedelta(hours=abs(tz))
    if tz > 0:
        schedule -= tdelta
    else:
        schedule += tdelta
    return schedule

def convertLocalToUTC(df):
    arrival_schedules = []
    departure_schedules = []
    for i, row in df.iterrows():
        arrival_schedule = row["arrival_schedule"]
        arrival_tz = row["arrival_tz"]
        departure_schedule = row["departure_schedule"]
        departure_tz = row["departure_tz"]

        arrival_schedules.append(timeToUTC(arrival_schedule, arrival_tz))
        departure_schedules.append(timeToUTC(departure_schedule, departure_tz))

    df["arrival_schedule"] = arrival_schedules
    df["departure_schedule"] = departure_schedules
    return df
```

Met de vertrek en aankomsttijden in dezelfde tijdzone kunnen we nu de duur afleiden van de vluchten.

2.7 Bereken de afstand

```
def addDistances(df):
    distances = []
    for index, row in df.iterrows():
        distances.append(calcTheDistance(row))
    df["distance"] = distances
    return df

def calcTheDistance(el):
    slat = radians(float(el["arrival_lat"]))
    slon = radians(float(el["arrival_lon"]))
    elat = radians(float(el["departure_lat"]))
    elon = radians(float(el["departure_lon"]))
    return 6371.01 * acos(sin(slat)*sin(elat) + cos(slat)*cos(elat)*cos(slon - elon))
```

We kunnen de afstand berekenen aan de hand van coördinaten van de luchthavens.

We zijn er bewust van dat die afstand niet 100% perfect is vanwege de gebruikte methode de *Great-circle distance* bereking. Die is niet 100% perfect want de aarde geen perfecte bol. En vliegtuigen vliegen niet altijd dezelfde routes vanwege de straalstroom, weeromstandigheden… Maar het is een goede benadering.

2.8 Bereken de snelheid

```
def addSpeed(df):
    speeds = []
    for i, row in df.iterrows():
        duration = row["duration"]
        speed = row["distance"] / ((duration.total_seconds() / 60) / 60)
        speeds.append(speed)

    df["speed"] = speeds
    return df
```

We berekenen de gemiddelde snelheid van het vliegtuig tijdens de geplande vlucht. Dit houd geen rekening met vertragingen.

2.9 Verwijder te korte en te lange vluchten

```
def deleteImpossibleFlights(df):
    shortest_distance_in_us = 100 #KM
    longest_distance_in_us = 9000 #KM
    shortest_duration_in_us = timedelta(minutes=16)
    longest_duration_in_us = timedelta(hours=11, minutes=40)

    df = df[df.distance.between(shortest_distance_in_us, longest_distance_in_us)]
    df = df[df.duration.between(shortest_duration_in_us,longest_duration_in_us)]
    return df
```

De kortste commerciële passagiersvlucht in de VS is 100km tussen San Francisco en Santa Rosa. De langste is 8019km tussen New York en Honolulu. Alle vluchten die hierbuiten vallen horen dus niet in onze dataset.

2.10 Uitvoeren van de opkuis-functies

```
print("Aantal records om te beginnen:", len(df))
cleaning = df

cleaning = cleaning.drop_duplicates()
print("Aantal records na verwijderen van dubbels:", len(cleaning))

cleaning = cleaning.reset_index()
print("Aantal records na nieuwe index:", len(cleaning))

cleaning = cleaning.dropna()
print("Aantal records na verwijderen lege waarden:", len(cleaning))

cleaning = deleteWrongStates(cleaning)
cleaning = convertColumnTypes(cleaning)

cleaning = dropMoreAdvancedDuplicates(cleaning)
print("Aantal records na het verwijderen van de verborgen dubbels:", len(cleaning))

cleaning = convertMilitaryTime(cleaning)
cleaning = addTimezones(cleaning)
cleaning = convertLocalToUTC(cleaning)
cleaning = addDurations(cleaning)
cleaning = addDistances(cleaning)
cleaning = addSpeed(cleaning)

cleaning = deleteImpossibleFlights(cleaning)
print("Aantal records na het verwijderen van de onmogelijk lange of korte vluchten:",
    len(cleaning))

cleanData = cleaning
```

Als we de geschreven python functies voor het opkuisen van de data uitvoeren, print python de volgende uitvoer:

```
Aantal records om te beginnen: 11401196
Aantal records na verwijderen van dubbels: 10751921
Aantal records na nieuwe index: 10751921
Aantal records na verwijderen lege waarden: 10751919
Aantal records: 10751919
Aantal records na verwijderen foute vertrek staat: 10695506
Aantal records na verwijderen foute aankomst staat: 10642322
Aantal records na het verwijderen van de verborgen dubbels: 10642032
Aantal records na het verwijderen van de onmogelijk lange of korte
vluchten: 10552133
```

We waren begonnen met **11.401.196** records, na het opkuisen van de data hebben we er nog **10.552.133** over. Dat wil zeggen dat er **849.063** verkeerde records waren, dat is ongeveer **7.447%** van de dataset.

3. Tableau

We hebben met de data de volgende grafieken gemaakt:

- How many flights are being served by which carrier from 2011 to 2012
    - => p. 14
- How many flights did each carrier carried out from 2011 to 2012
    - => p. 16
- What are the hubs for each carrier?
    - => p. 19
- The total amount of scheduled flight time compared to the actual flight time per carrier from 2011 to 2012
    - => p. 21
- How many turns around the world could you travel from the total flow km per carrier from 2011 to 2012
    - => p. 23
- The flight distance compared to the scheduled flight time per carrier
    - => p. 25
- Per carrier what are the most popular departure states and arrival states
    - => p. 27
- What are the top 10 busiest airports and their average time delays
    - => p. 29
- Average scheduled flight time compared to the average delay per carrier from 2011 to 2012
    - => p. 31
- The average departure delay and the average arrival delay compared to the number of flights per month
    - => p. 33
- The average flight time and average delay per flight category compared with each carrier
    - => p. 35

3.1 Algemene informatie over de vliegtuig maatschappijen

3.1.1 American Airlines (AA)

American Airlines is opgericht in 1930.
Het hoofdkwartier is gelegen in Fort Worth in de staat Texas.
De main-hub is DFW en deze is ligt in de staat Texas.

3.1.2 Alaska Airlines (AS)

Alaska Airlines is opgericht in 1932
Het hoofdkwartier is gelegen in Seattle in de staat Washington.
De main-hub is SEA en deze is ligt in de staat Washington.

3.1.3 Jetblue Airways (B6)

Jetblue Airways is opgericht in 1998.
Het hoofdkwartier is gelegen in New York City in de staat New York.
De main-hub is JFK en deze is ligt in de staat New York.

3.1.4 Continental Airlines (CO)

Continental Airlines is opgericht in 1934.
Het hoofdkwartier is gelegen in Houston in de staat Texas.
De main-hub is IAH en deze is ligt in de staat Texas.
Continental Airlines is echter in 2012 overgenomen door United Airlines.

3.1.5 Delta Air Lines (DL)

Delta Air Lines is opgericht in 1924.
Het hoofdkwartier is gelegen in Atlanta in de staat Georgia.
De main-hub is ATL en deze is ligt in de staat Georgia.

3.1.6 ExpressJet Airlines (EV)

ExpressJet Airlines is opgericht in 1986
Het hoofdkwartier is gelegen in Atlanta in de staat Georgia.
De main-hub is IAH en deze is ligt in de staat Texas.
ExpressJet Airlines is echter een regionale maatschappij wilt dus zeggen dat ze in de naam van een grote speler vluchten uitvoeren.

3.1.7 Frontier Airlines (F9)

Frontier Airlines is opgericht in 1994
Het hoofdkwartier is gelegen in Denver in de staat Colorado.
De main-hub is DEN en deze is ligt in de staat Colorado.

3.1.8 AirTran Airways (FL)

AirTran Airways is opgericht in 1992.
Het hoofdkwartier is gelegen in Orlando in de staat Florida.
De main-hub is ATL en deze is ligt in de staat Florida.
AirTran Airways is echter in 2011 overgenomen door Southwest Airlines.

3.1.9 Hawaiian Airlines (HA)

Hawaiian Airlines is opgericht in 1929.
Het hoofdkwartier is gelegen in Honolulu in de staat Hawaii.
De main-hub is HNL en deze is ligt in de staat Hawaii.

3.1.10 Envoy Air (MQ)

Envoy Air is opgericht in 1984.
Het hoofdkwartier is gelegen in Irving in de staat Texas.
De main-hub is DFW en deze is ligt in de staat Texas.
Envoy Air is echter een regionale maatschappij wilt dus zeggen dat ze in de naam van een grote speler vluchten uitvoeren.

3.1.11 SkyWest Airlines (OO)

SkyWest Airlines is opgericht in 1972.
Het hoofdkwartier is gelegen in St. George in de staat Utah.
De main-hub is LAX en deze is ligt in de staat California.
SkyWest Airlines is echter een regionale maatschappij wilt dus zeggen dat ze in de naam van een grote speler vluchten uitvoeren.

3.1.12 United Airlines (UA)

United Airlines is opgericht in 1926.
Het hoofdkwartier is gelegen in Chicago in de staat Illinois.
De main-hub is ORD en deze is ligt in de staat Illinois.

3.1.13 US Airways (US)

US Airways is opgericht in 1937.
Het hoofdkwartier is gelegen in Temp in de staat Arizona.
De main-hub is PHL en deze is ligt in de staat Pennsylvania.
US Airways is echter in 2013 gefuseerde met American Airlines.

3.1.14 Virgin America (VX)

Virgin America is opgericht in 2004.
Het hoofdkwartier is gelegen in Burlingame in de staat California.
De main-hub is SFO en deze is ligt in de staat California.
Virgin America is echter in 2018 overgenomen door Alaska Airlines.

3.1.15 Southwest Airlines (WN)

Southwest Airlines is opgericht in 1967.
Het hoofdkwartier is gelegen in Dallas in de staat Texas.
De main-hub is MDW en deze is ligt in de staat Illinois.

3.1.16 JetSuite (XE)

JetSuite is opgericht in 2006.
Het hoofdkwartier is gelegen in Irvine in de staat California.
De main-hub is IAH en deze is ligt in de staat Texas.
JetSuite is echter een private jet maatschappij.

3.1.17 Mesa Airlines (YV)

Mesa Airlines is opgericht in 1980.
Het hoofdkwartier is gelegen in Phoenix in de staat Arizona.
De main-hub is PHX en deze is ligt in de staat Arizona.
Mesa Airlines is echter een regionale maatschappij wilt dus zeggen dat ze in de naam van een grote speler vluchten uitvoeren.

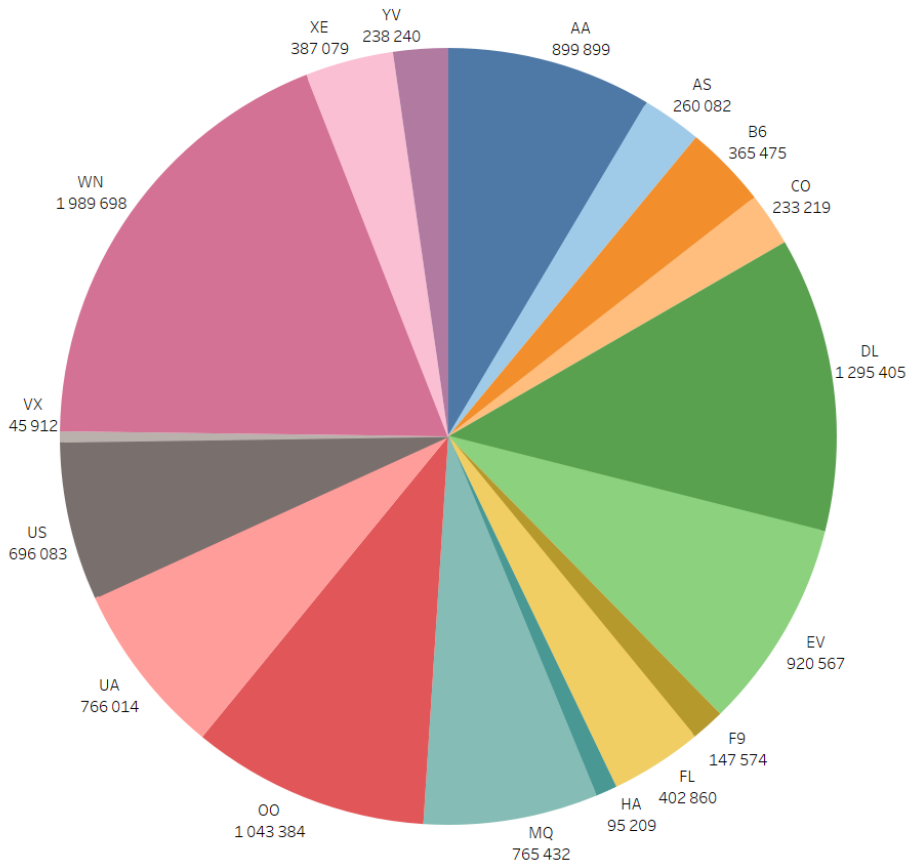
3.2 Extra info over SkyWest en ExpressJet

SkyWest Airlines en ExpressJet Airlines zijn 2 regionale maatschappijen die samen gaan onder 1 groot bedrijf. Dit bedrijf/holding noemt SkyWest Incorporated. Er zijn echter geen rapport beschikbaar van 2011 en 2012 van enkel SkyWest of ExpressJet. Er is maar 1 annual rapport en dat is dat van SkyWest Incorporated, waarin de gegevens van beide maatschappijen staan. Dit zullen we dus nemen om te vergelijken.

3.3 Grafieken

3.3.1 Grafiek 1: How many flights are being served by which carrier from 2011 to 2012

How many flights are being served by which carrier from 2011 to 2012



Airline
AA
AS
B6
CO
DL
EV
F9
FL
HA
MQ
OO
UA
US
VX
WN
XE
YV

AA : (American Airlines)
AS : (Alaska Airlines)
B6 : (Jetblue Airways)
CO : (Continental Airlines)
DL : (Delta Air Lines)
EV : (ExpressJet Airlines)
F9 : (Frontier Airlines)
FL : (AirTran Airways)
HA : (Hawaiian Airlines)
MQ : (Envoy Air)
OO : (SkyWest Airlines)
UA : (United Airlines)
US : (US Airways)
VX : (Virgin America)
WN : (Southwest Airlines)
XE : (JetSuite)
YV : (Mesa Airlines)

Conclusie grafiek 1: How many flights are being served by which carrier from 2011 to 2012

De top 3 maatschappijen met de meeste vluchten:

- *Southwest Airlines* met 1.98 miljoen vluchten in 2 jaar.
- *Delta Airlines* met 1.29 miljoen vluchten in 2 jaar.
- *SkyWest Airlines* met 1.04 miljoen vluchten in 2 jaar.

Southwest Airlines is een low-cost maatschappij die alleen maar in de VS vliegt, in het jaar 2010 vlogen ze naar 72 verschillende steden in 37 staten[1].

Bij *Delta Airlines* zijn er dagelijks +15.000 vluchten, maar *Delta Airlines* vliegt naar 306 bestemming over 52 landen[2]. Dus het aantal vluchten per dag binnen de USA zal minder zijn.

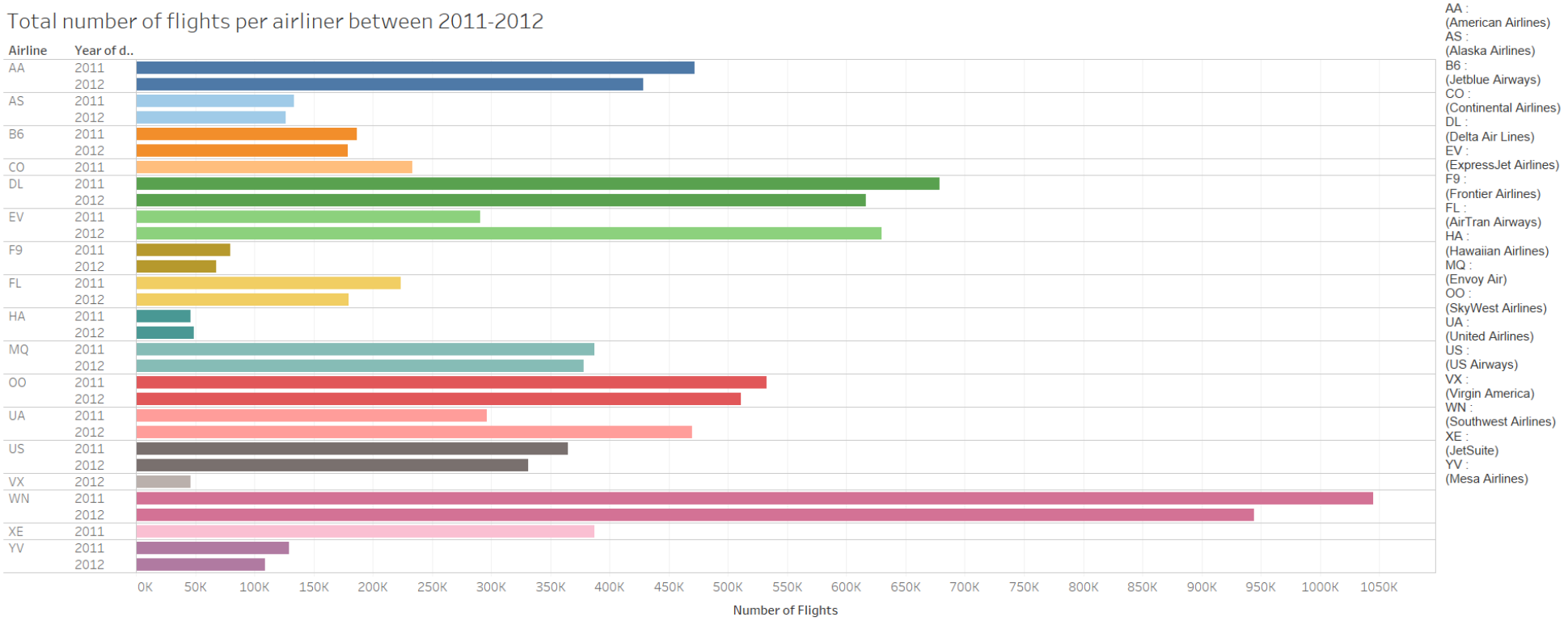
Skywest Airlines is een regionale maatschappij die vluchten uitvoert in naam van *Delta Airlines, United Airlines, Continental Airlines* …

	CRJ 200	ERJ 145	CRJ700	CRJ 900	EMB 120	Total
Delta	160	—	67	31	10	268
United	96	36	70	—	35	237
Continental	—	206	—	—	—	206
Alaska	—	—	5	—	—	5
US Airways	2	—	—	—	—	2
Maintenance Spare	8	—	—	—	—	8
Subleased to an un-affiliated entity	2	—	—	—	—	2
Subleased to an affiliated entity	—	—	—	4	—	4
Total	268	242	142	35	45	732

Figuur 1 verdeling vloot van Skywest Inc.[3] voor het jaar 2011

[1] Bron: http://investors.southwest.com/~/media/Files/S/Southwest-IR/documents/company-reports-ar/ar-2011.PDF pagina 7
[2] https://news.delta.com/corporate-stats-and-facts
[3] http://inc.skywest.com/assets/Uploads/AnnualReports/10k-2011.pdf Pagina 45

3.3.2 Grafiek 2: How many flights did each carrier carried out from 2011 to 2012



Conclusie grafiek 2: How many flights did each carrier carried out from 2011 to 2012

Wat er hier in opvalt is dat iedere maatschappij een daling ziet van het aantal uitgevoerde vluchten. Behalve 2 maatschappijen:

- ExpressJet Airlines
- United Airlines

Virgin America heeft enkel maar flight records in 2012 en geen van 2011, er is daar echter geen logische uitleg daarvoor want de maatschappij is opgericht in 2004 en de eerste vluchten waren in 2007[4].

Zoals je kan zien zijn er van *Continental Airlines* enkel maar records van 2011, dat komt omdat *Continental Airlines* en *United Airlines* gefuseerd zijn begin 2012[5]. Dat verklaart de plotse stijging van het aantal vluchten van *United Airlines* van +300K naar +460K .

Bij *ExpressJet Airlines* is er echter een verdubbeling van de vluchten van 300K naar 640K in 2012. Als we gaan kijken naar het jaarrapport van 2012 zien we in het geel staan hoeveel vluchten *ExpressJet* en *SkyWest Airlines* hebben uitgevoerd. In 2011 waren dit 1.390 miljoen vluchten, in 2012 was dit echter gestegen tot 1.435 miljoen vluchten. Tellen we echter de data op dat wij hebben komt dit voor 2011 neer op 0.823 miljoen vluchten en voor 2012 is dat dan 1.140 miljoen. Dit wilt zeggen dat er in die 2 jaar 0.9 miljoen records ontbreken in de gekregen dataset.

	Year Ended December 31,		
	2012	2011	% Change
Revenue passenger miles (000)	30,088,278	29,109,039	3.4%
Available seat miles ("ASMs") (000)	37,278,554	36,698,859	1.6%
Block hours	2,297,014	2,250,280	2.1%
Departures	1,435,512	1,390,523	3.2%
Passengers carried	58,803,690	55,836,271	5.3%
Passenger load factor	80.7%	79.3%	1.4pts
Revenue per available seat mile	9.5¢	10.0¢	(5.0)%
Cost per available seat mile	9.2¢	10.1¢	(8.9)%
Fuel cost per available seat mile	1.1¢	1.6¢	(31.3)%
Average passenger trip length (miles)	512	521	(1.7)%

Figuur 2 Aantal vluchten voor SkyWest Inc.[6]

[4] https://en.wikipedia.org/wiki/Virgin_America
[5] https://uk.reuters.com/article/us-unitedcontinental/united-gets-faa-single-operating-certificate-idUKTRE7AT1JP20111130
[6] http://inc.skywest.com/assets/Uploads/AnnualReports/10k-2012.pdf pagina 54

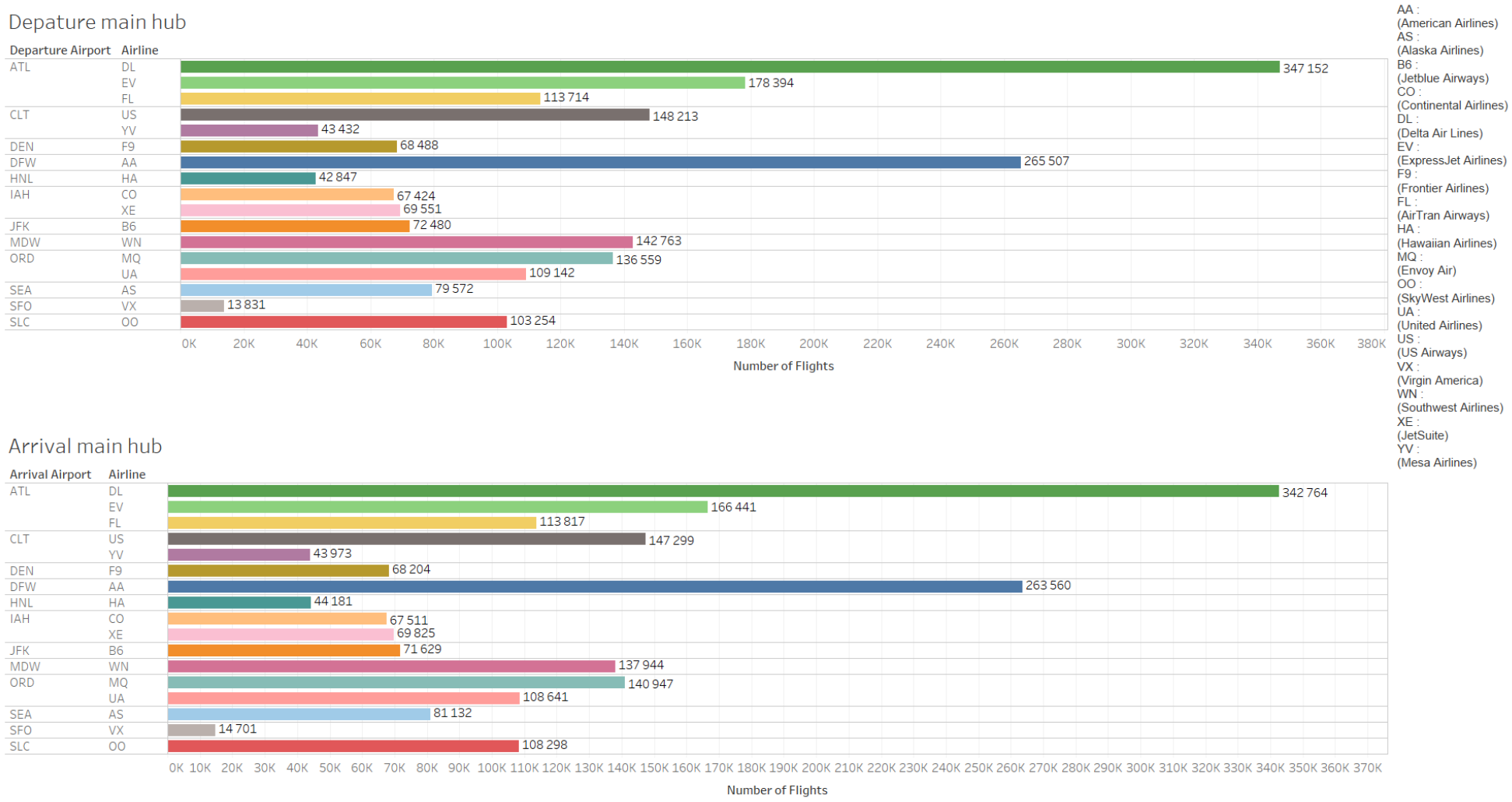
Opmerkelijk is het aantal vluchten van *SouthWest,* die dalen met ongeveer 10% van 1.05 miljoen naar 0.94 miljoen. Zoeken we echter het jaarrapport 2012 – 2011 van *SouthWest* op, dan vinden we iets opmerkelijk terug (zie cijfers in het geel). We zien een lichte stijging van het aantal vluchten van 1.317 miljoen in 2011 naar 1.361 miljoen in 2012. Kijken we echter naar de cijfers dat wij hebben, zien we een daling van 1.050 miljoen in 2011 naar 0.940 miljoen in 2012. Dit wilt zeggen dat er + 0.7 miljoen vluchten van *SouthWest* ontbreken in de gekregen dataset.

	Year ended December 31,				
	2012	2011	2010	2009	2008
Financial Data (in millions, except per share amounts):					
Operating revenues	$ 17,088	$ 15,658	$ 12,104	$ 10,350	$ 11,023
Operating expenses	16,465	14,965	11,116	10,088	10,574
Operating income	623	693	988	262	449
Other expenses (income) net	(62)	370	243	98	171
Income before taxes	685	323	745	164	278
Provision for income taxes	264	145	286	65	100
Net income	$ 421	$ 178	$ 459	$ 99	$ 178
Net income per share, basic	$ .56	$ .23	$ .62	$ .13	$ .24
Net income per share, diluted	$ .56	$ .23	$ .61	$ .13	$ .24
Cash dividends per common share	$ .0345	$ .0180	$ .0180	$ .0180	$ .0180
Total assets at period-end	$ 18,596	$ 18,068	$ 15,463	$ 14,269	$ 14,068
Long-term obligations at period-end	$ 2,883	$ 3,107	$ 2,875	$ 3,325	$ 3,498
Stockholders' equity at period-end	$ 6,992	$ 6,877	$ 6,237	$ 5,454	$ 4,953
Operating Data:					
Revenue passengers carried	109,346,509	103,973,759	88,191,322	86,310,229	88,529,234
Enplaned passengers	133,978,100	127,551,012	106,227,521	101,338,228	101,920,598
Revenue passenger miles (RPMs) (000s) (1)	102,874,979	97,582,530	78,046,967	74,456,710	73,491,687
Available seat miles (ASMs) (000s) (2)	128,137,110	120,578,736	98,437,092	98,001,550	103,271,343
Load factor (3)	80.3%	80.9%	79.3%	76.0%	71.2%
Average length of passenger haul (miles)	941	939	885	863	830
Average aircraft stage length (miles)	693	679	648	639	636
Trips flown	1,361,558	1,317,977	1,114,451	1,125,111	1,191,151
Average passenger fare	$ 147.17	$ 141.90	$ 130.27	$ 114.61	$ 119.16
Passenger revenue yield per RPM (cents) (4)	15.64	15.12	14.72	13.29	14.35
Operating revenue per ASM (cents) (5)	13.34	12.99	12.30	10.56	10.67
Passenger revenue per ASM (cents) (6)	12.56	12.24	11.67	10.09	10.21
Operating expenses per ASM (cents) (7)	12.85	12.41	11.29	10.29	10.24
Operating expenses per ASM, excluding fuel (cents)	8.07	7.73	7.61	7.18	6.64
Operating expenses per ASM, excluding fuel and profitsharing (cents)	7.98	7.65	7.45	7.15	6.56
Fuel costs per gallon, including fuel tax	$ 3.30	$ 3.19	$ 2.51	$ 2.12	$ 2.44
Fuel costs per gallon, including fuel tax, economic	3.28	3.19	2.39	1.97	2.32
Fuel consumed, in gallons (millions)	1,847	1,764	1,437	1,428	1,511
Active fulltime equivalent Employees	45,861	45,392	34,901	34,726	35,499
Aircraft in service at period-end (8)	694	698	548	537	537

Figuur 3 Aantal vluchten van SouthWest[7]

[7] http://investors.southwest.com/~/media/Files/S/Southwest-IR/documents/company-reports-ar/ar-2012.PDF pagina 51

3.3.3 Grafiek 3: What are the hubs for each carrier?



Conclusie Grafiek 3: What are the hubs for each carrier?

We hebben gekeken welke luchthavens het vaakst gebruikt worden om op te vertrekken en op te landen door een vliegtuigmaatschappij.

Als je kijkt naar de **Arrival Aiport** en de **Departure Airport** zijn de luchthavens gelijk aan elkaar. Dit wil dus zeggen dat hun hubs gelijk zijn.

Hieronder maken we een tabel waarin we de hubs waar ze het meest op landen/vertrekken vergelijken met de main-hub (zie 2.1 Algemene Informatie over de vliegtuig maatschappijen)

Vliegtuig Maatschappij	**Hubs van de grafiek**	**Main-hub**
American Airlines (AA)	DFW	DFW
Alaska Airlines (AS)	SEA	SEA
Jetblue Airways (B6)	JFK	JFK
Continental Airlines (CO)	IAH	IAH
Delta Airlines (DL)	ALT	ALT
ExpressJet Airlines (EV)	ALT	IAH
Frontier Airlines (F9)	DEN	DEN
AirTran Airways (FL)	ALT	ATL
Hawaiian Airlines (HA)	HNL	HNL
Envoy Air (MQ)	ORD	DFW
SkyWest Airlines (OO)	SLC	LAX
United Airlines (UA)	ORD	ORD
US Airways (US)	CLT	CLT
Virgin America (VX)	SFO	SFO
Southwest Airlines (WN)	MDW	MDW
JetSuite (XE)	IAH	IAH
Mesa Airlines (YV)	CLT	PHX

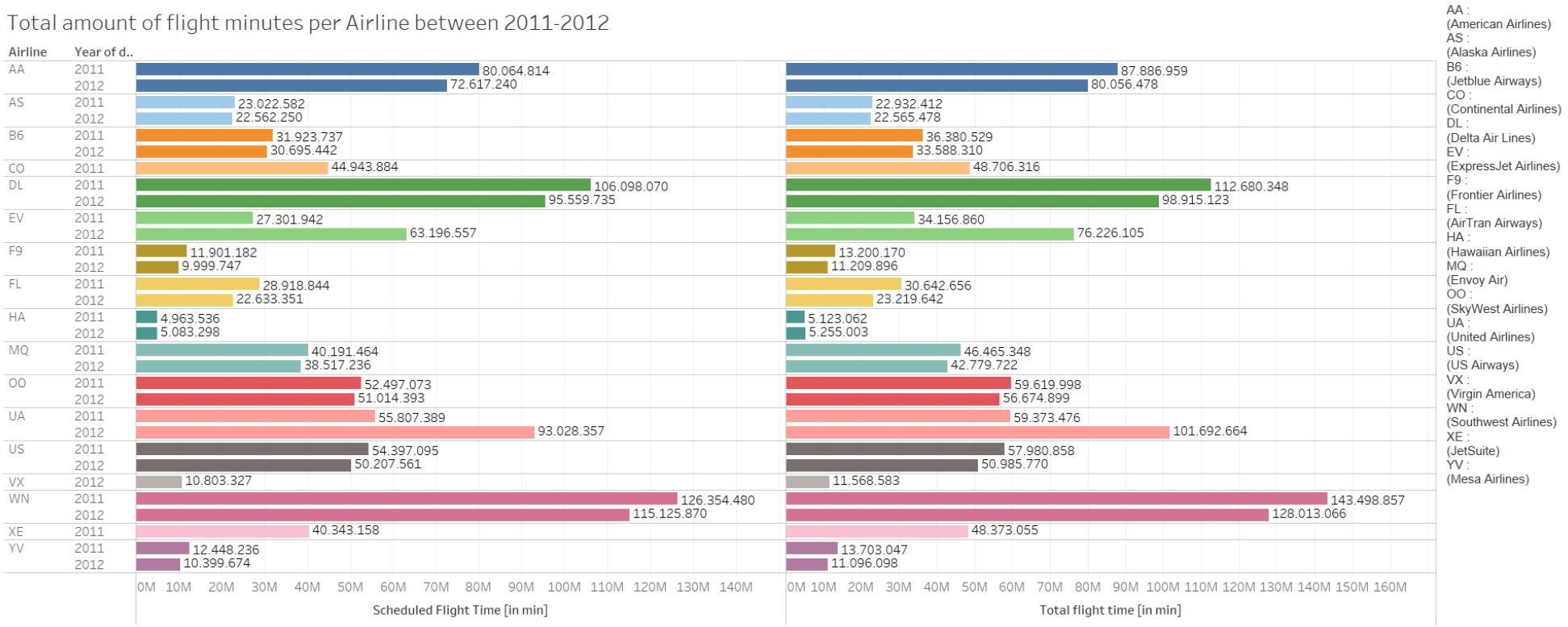
De hubs in het rood zijn die hubs die niet gelijk zijn aan hun main-hub. Hoe komt dit?

Het gaat om de volgende maatschappijen :

- *ExpressJet Airlines*
- *Envoy Air*
- *SkyWest Airlines*
- *Mesa Airlines*

Wat er speciaal is aan die maatschappijen is dat alle 4 maatschappijen zijn die regionale vluchten uitvoeren voor grote spelers. Dus de luchthaven dat ze het meest gebruiken is niet de luchthaven dat zij zien als hun main-hub. Zij kiezen namelijk zelf geen routes maar voeren routes uit van de vliegtuigmaatschappijen die hun een contract aan bieden om in naam van hun die route te vliegen.

3.3.4 Grafiek 4: The total amount of scheduled flight time compared to the actual flight time per carrier from 2011 to 2012



Conclusie grafiek 4: The total amount of scheduled flight time compared to the actual flight time per carrier from 2011 to 2012

Wat we hier hebben gedaan is gekeken per jaar per vliegtuigmaatschappij naar de vliegtijd dat ze voorzien/gepland hadden en de tijd die het vliegtuig uiteindelijk nodig had om op zijn bestemming te komen.

Als we kijken naar de vliegtuigmaatschappij die het meeste aantal vluchten heeft uitgevoerd[8]. Dat waren namelijk:

- *Southwest Airlines*
- *Delta Airlines*
- *SkyWest Airlines*

Maar als je kijkt naar de vliegtuigmaatschappijen die het langst gevlogen hebben komt dit op een andere top 3 neer namelijk:

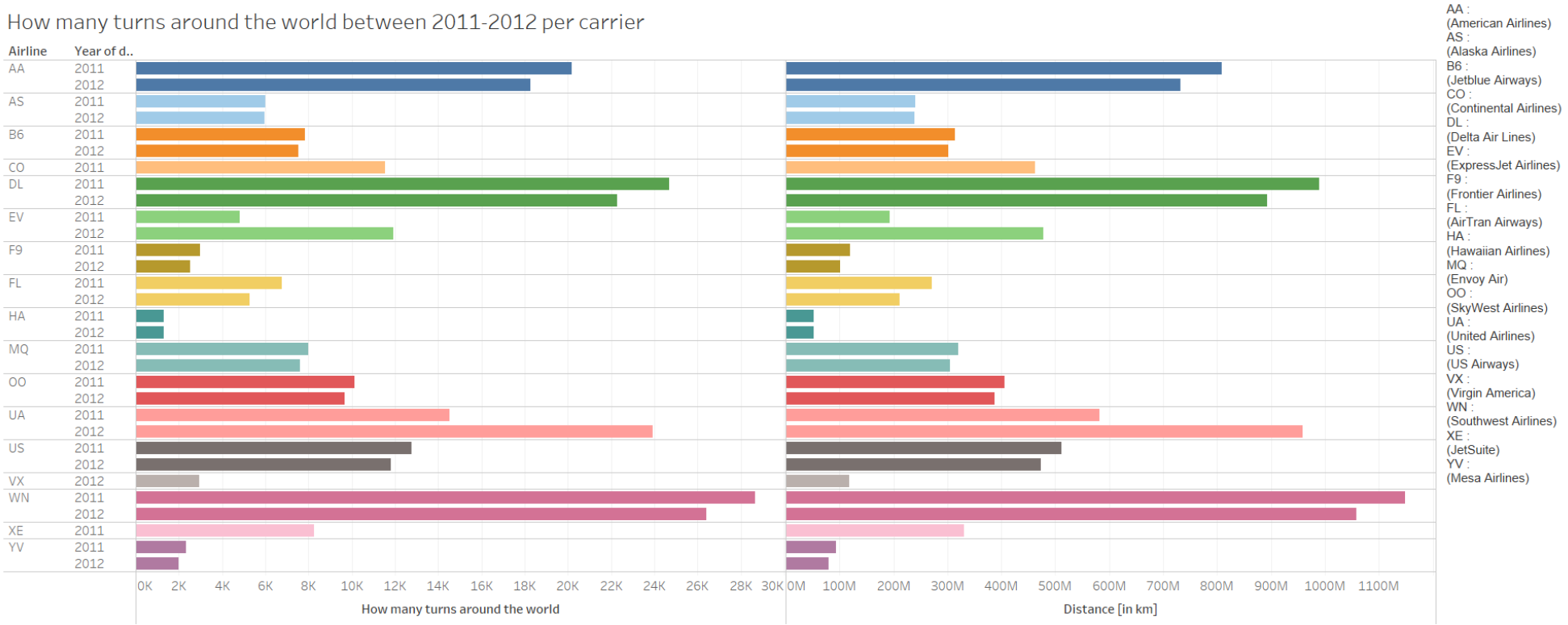
- *Southwest Airlines*
- *Delta Airlines*
- *American Airlines*

De nieuwkomer in deze lijst is American Airlines. Als je dan gaat kijken waar SkyWest zou staan zou dat maar op 6de plaats zijn .

Wat er ook op valt is dat Alaska Airlines in 2011 sneller vliegt dan dat er ingepland was. De ingeplande vliegtijd was 23.022.582 minuten en ze hebben uiteindelijk 22.932.412 minuten gevlogen. Er is hier echt geen uitleg voor. Waarschijnlijk is de gekregen data niet correct.

[8] Zie grafiek 1 pagina 15

3.3.5 Grafiek 5: How many turns around the world could you travel from the total flown km per carrier from 2011 to 2012



Conclusie grafiek 5: How many turns around the world could you travel from the total flown km per carrier from 2011 to 2012

We proberen hier duidelijk te maken hoeveel kilometers er per vliegtuigmaatschappij gevlogen word.

Omdat een aantal miljoenen kilometers niet voor iedereen zoveel zegt, hebben we er ook voor gekozen om de afstand uit te drukken in het aantal keer dat je rond de aarde kan vliegen.

De top 3:

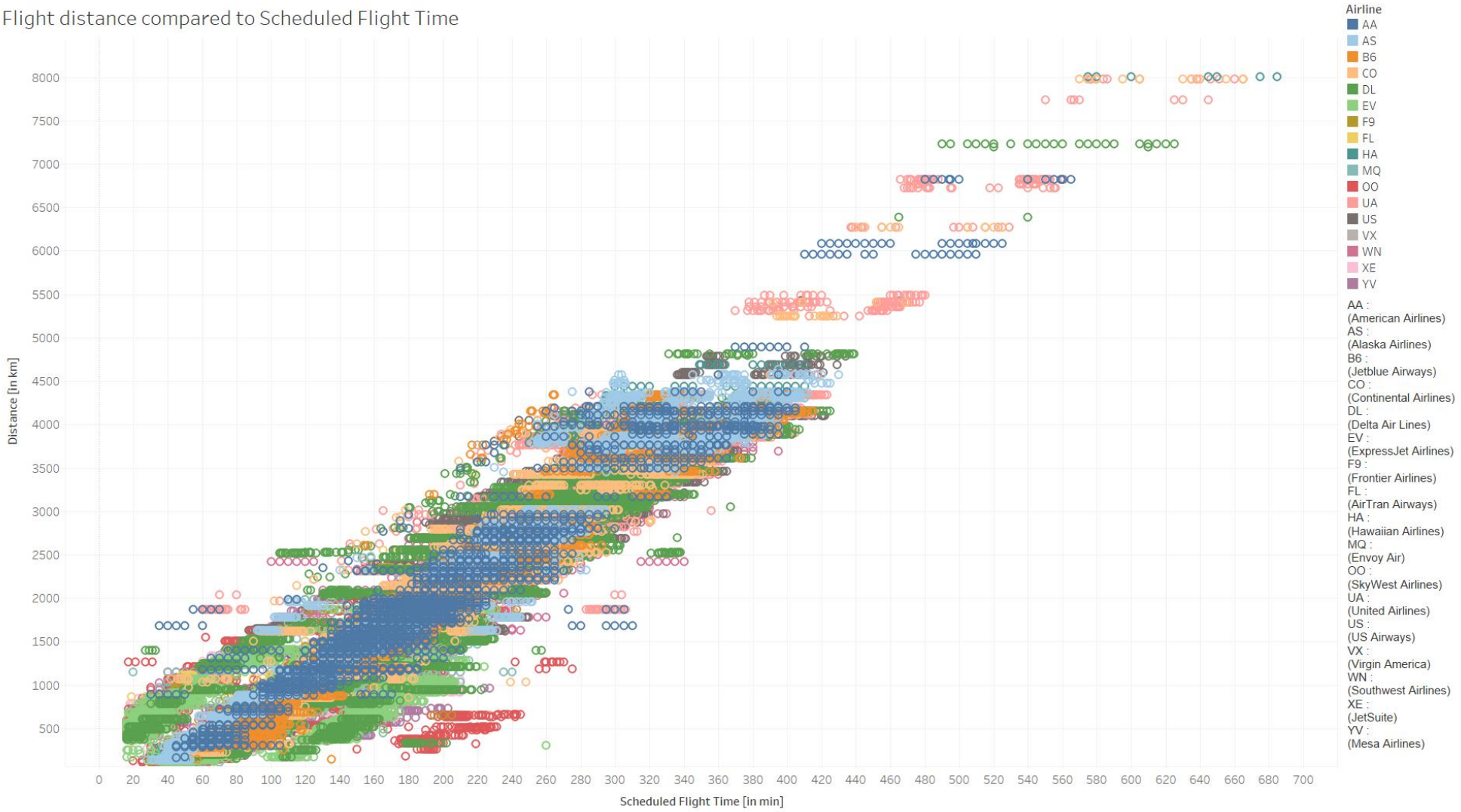
- *Southwest Airlines*
  - Vliegen 2.150 miljoen km => 53.000 keer rond de aarde
- *Delta Airlines*
  - Vliegen 1.900 miljoen km => 48.000 keer rond de aarde
- *American Airlines*
  - Vliegen 1.500 miljoen km => 38.000 keer rond de aarde

Als je kijkt naar *Hawaiian Airlines* zie je dat deze maatschappij de minste aantal keren rond de wereld kan vliegen namelijk 2.602. Dit is 20 keer minder dan Southwest Airlines. Dit komt echter omdat *Hawaiian Airlines* niet enorm veel vluchten heeft naar The United States Of America, ze vliegen ook naar New Zeeland, Japan, Korea, Australië, Filipijnen en paar eilanden[9]. In juni 2012[10] begonnen ze maar non-stop naar JFK te vliegen.

[9] Pagina 2:
https://newsroom.hawaiianairlines.com/internal_redirect/cms.ipressroom.com.s3.amazonaws.com/249/files/20156/Hawaiian%20Holdings%2010-K%20Annual%20Report.pdf

[10] Pagina 2:
https://newsroom.hawaiianairlines.com/internal_redirect/cms.ipressroom.com.s3.amazonaws.com/249/files/20156/231925-3-web%20ready.pdf

3.3.6 Grafiek 6: The fight distance compared to the scheduled flight time per carrier



Conclusie grafiek 6: The fight distance compared to the scheduled flight time per carrier

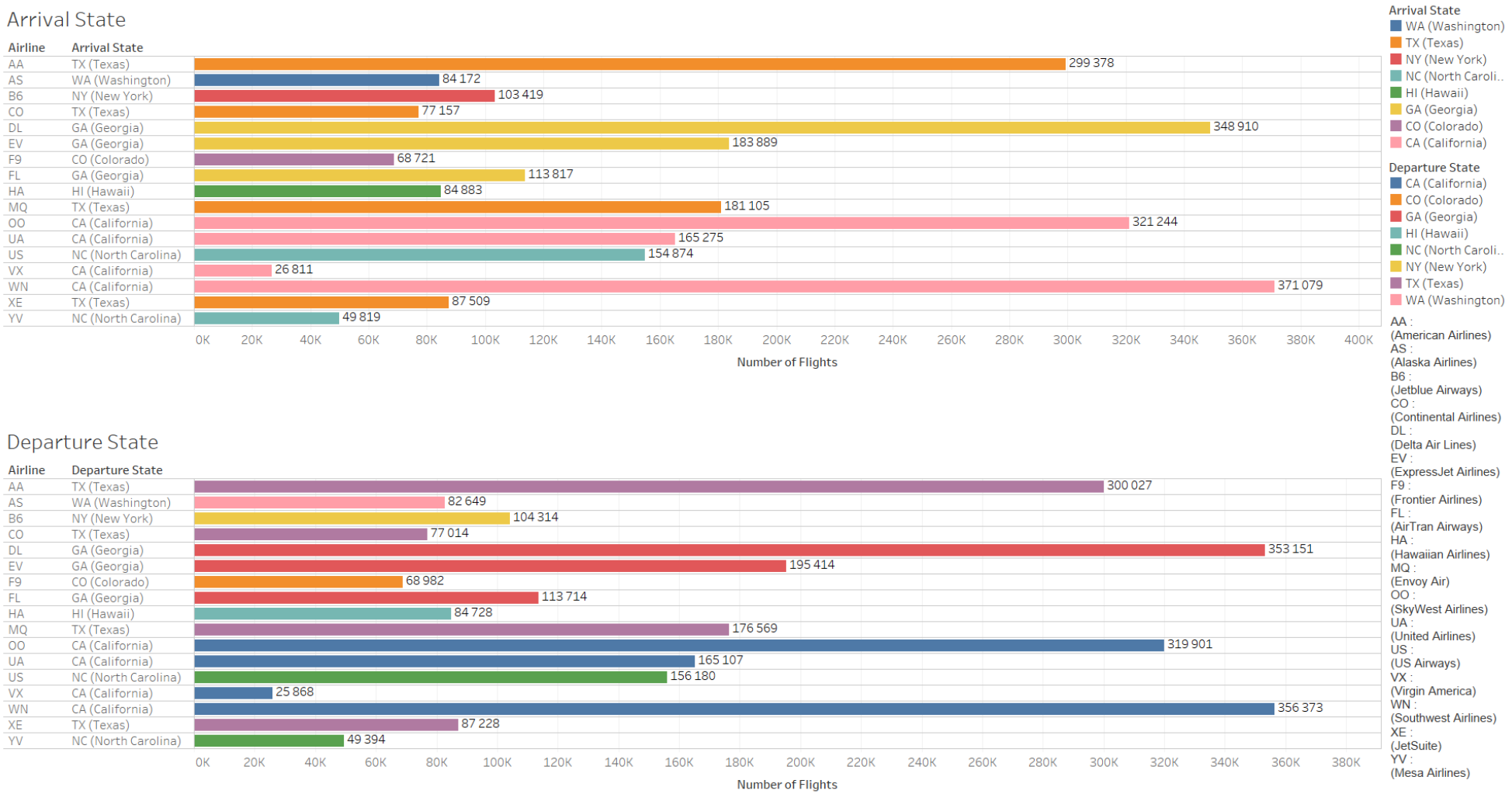
Hierin wordt er voor iedere geplande vlucht met een andere afstand een bolletje geplaatst. Zoals je kan zien is er veel groen (*Delta Airlines*) en donker blauw (*American Airlines*). *Delta Airlines* en *American Airlines* zijn overal aanwezig zoals je kan zien. Zij vliegen zowel korte vluchten in de VS als langere vluchten, zoals directe vluchten van de Westkust van de VS naar de Oostkust van de VS.

Het paarse (*Southwest Airlines*) kan je niet zien omdat het onder het groene ligt. *Southwest Airlines* doet voornamelijk korte vluchten.

Het lichtblauwe (*Alaska Airlines*) kan je 2 geconcentreerde gebieden vinden namelijk rond de 3500 km tot 4000k m en ook rond de 500 km. De zone van rond de 500 km is echter binnenvluchten binnen de staat van Alaska. De zone rond de van 3500 km tot 4000 km zijn vluchten van Alaska naar het andere deel van de VS.

Helemaal van boven rechts vind je 2 bolletjes van *Hawaii Airlines,* dit zijn namelijk vluchten van Honolulu naar New York City. Dit zijn de langste non-stop vluchten die mogelijk zijn binnen de 50 staten.

3.3.7 Grafiek 7: Per carrier what are the most popular departure states and arrival states



Conclusie grafiek 7: Per carrier what are the most popular departure states and arrival states

We hebben gekeken per vliegtuig maatschappij welke staat dat ze het meest van vertrekken/landen.

Als je kijkt per maatschappij zie je dat de staat waar ze meest van vertrekken gelijk is aan de staat waar ze in landen. Maar als je beter kijkt naar de cijfers zie je dat het niet exact gelijk is aan elkaar.

Vliegtuig Maatschappij	**Vertrokken vluchten**	**Gelande vluchten**
American Airlines (AA)	300 027	299 378
Alaska Airlines (AS)	82 649	84 172
Jetblue Airways (B6)	104 314	103 419
Continental Airlines (CO)	77 014	77 157
Delta Airlines (DL)	353 151	348 910
ExpressJet Airlines (EV)	195 414	183 889
Frontier Airlines (F9)	68 982	68 721
AirTran Airways (FL)	113 714	113 817
Hawaiian Airlines (HA)	84 728	84 883
Envoy Air (MQ)	176 569	181 105
SkyWest Airlines (OO)	319 901	321 244
United Airlines (UA)	165 107	165 275
US Airways (US)	156 180	154 874
Virgin America (VX)	25 868	26 811
Southwest Airlines (WN)	356 373	371 079
JetSuite (XE)	87 288	87 509
Mesa Airlines (YV)	49 394	49 819

De rode kleuren is de kolom waar er het meest vluchten zijn. Meestal er is maar maximaal een verschil van een paar duizend vluchten. Bij de volgende maatschappijen is er een groter verschil:

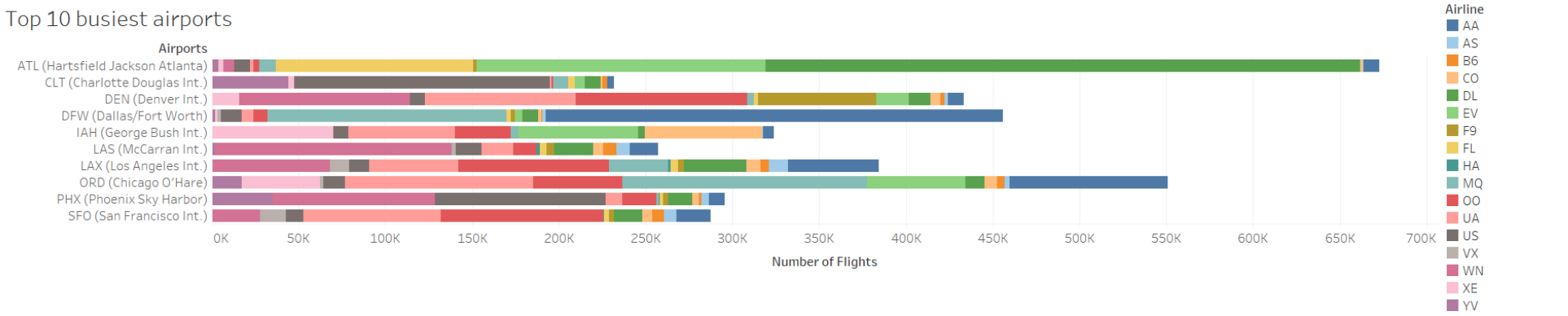
- *Delta Airlines*
- *ExpressJet Airlines*
- *Envoy Air*
- *Southwest Airlines*

De reden daarvoor is waarschijnlijk geannuleerde vluchten door de stormen, één ervan is de *"North American blizzard*[11]*"* die zorgde ervoor dat er over heel The United States Of America 14 000 geannuleerd vluchten waren.

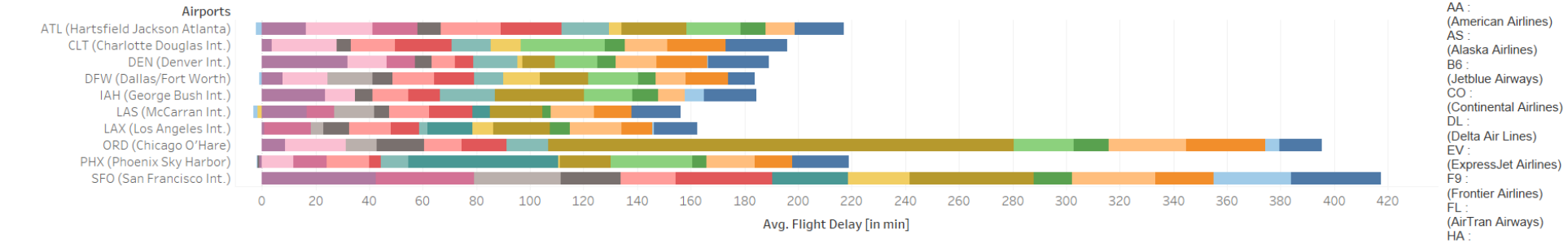
[11] https://en.wikipedia.org/wiki/January_31_%E2%80%93_February_2,_2011_North_American_blizzard

3.3.8 Grafiek 8: What are top 10 busiest airports and their average time delays

Top 10 busiest airports

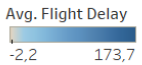


Avg. time delays on the busiest airports ranked by airlines



Avg. time delays on the busiest airports ranked by airlines (2)

Airports	Airline AA	AS	B6	CO	DL	EV	F9	FL	HA	MQ	OO	UA	US	VX	WN	XE	YV
ATL (Hartsfield Jackson Atlanta)	18,4	-2,2		10,7	9,5	19,9	24,5	4,4		17,8	22,6	22,6	8,6		17,0	24,7	16,4
CLT (Charlotte Douglas Int.)	23,0		21,7	16,0	7,3	31,6		11,1		14,6	21,0	16,7	5,0			24,5	3,5
DEN (Denver Int.)	22,7	0,2	19,1	15,2	6,7	15,8	12,1	1,8		16,5	7,0	8,7	6,1		10,6	14,6	31,9
DFW (Dallas/Fort Worth)	9,8	-0,9	16,0	11,2	6,4	18,8	17,8	13,7		11,0	14,9	15,5	7,5	16,7		16,9	7,6
IAH (George Bush Int.)	19,5	7,2		9,7	9,7	18,1	33,0			20,5	11,8	13,5	6,4			11,4	23,5
LAS (McCarran Int.)	18,2	-1,6	14,2	16,0	3,3		19,5	-1,6	6,5		16,2	14,9	5,6	15,0	10,3		16,6
LAX (Los Angeles Int.)	16,2	0,5	11,4	19,2	7,8		20,8	8,0	16,7	3,0	10,6	15,6	9,5	4,6	17,6		0,8
ORD (Chicago O'Hare)	15,7	5,3	29,5	29,1	12,9	22,5	173,7			15,5	16,8	13,9	17,7	11,7		22,6	8,6
PHX (Phoenix Sky Harbor)	21,0	-0,4	14,0	18,0	5,4	30,6	18,8	0,8	55,7	10,4	4,1	16,0	-0,5		12,4	11,7	-1,1
SFO (San Francisco Int.)	33,6	29,0	21,8	31,0	14,3		46,2	23,0	28,4		35,8	20,7	22,3	32,2	36,7		42,5



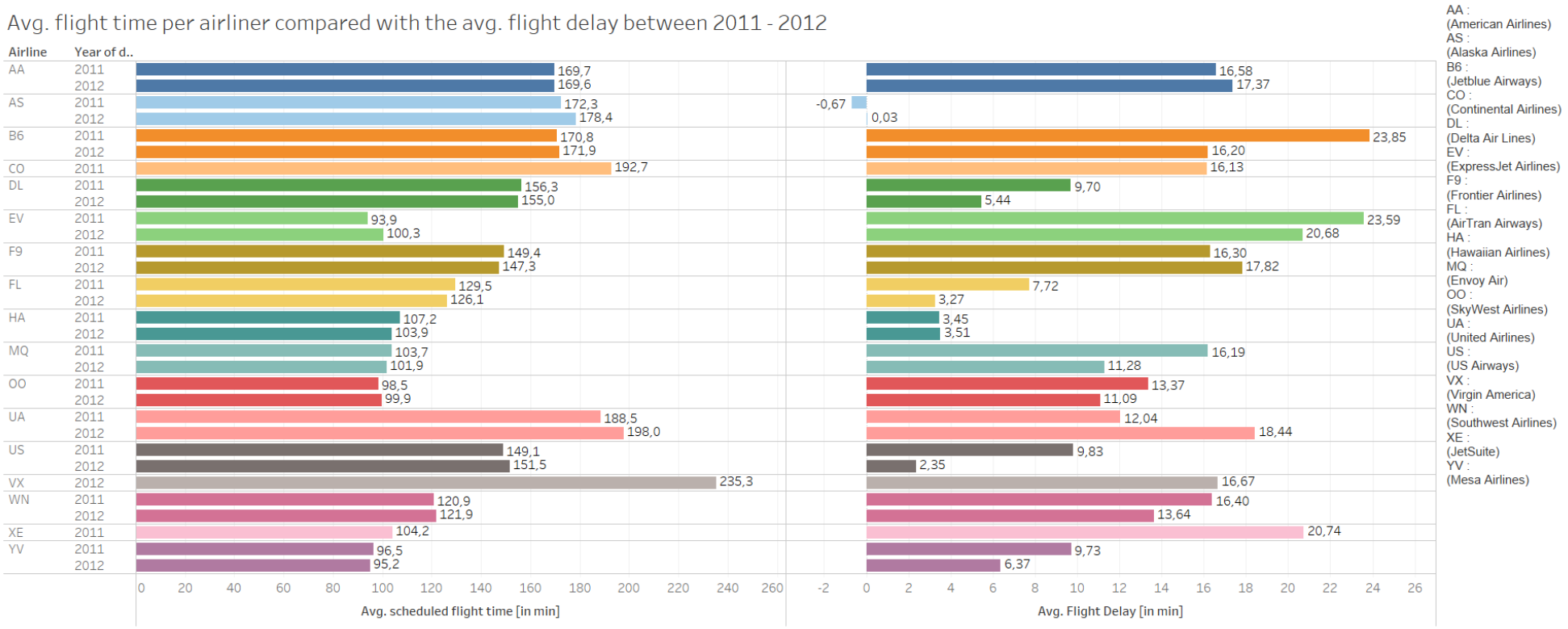
Conclusie grafiek 8: What are top 10 busiest airports and their average time delays

De grafiek met de naam *Top 10 busiest airports* toont de top 10 drukste luchthavens van The United States Of America, met daarin ook hoeveel vluchten er daar vertrekken van welke maatschappij.

In de grafiek met de naam *Avg. time delays on the busiest airports ranked by airlines* zie je de top 10 drukste luchthavens van The United States Of America met daarin de totale gemiddelde vertraging van de maatschappij.

In de grafiek met de naam *Avg.* Time *delays on the busiest airports ranked by airlines (2)* zie je dezelfde grafiek als de grafiek met titel *Avg. time delays on the busiest airports ranked by airlines* maar dan in tabel vorm zodat je kan zien welke specifiek gemiddelde vertraging er is bij de vliegtuigmaatschappij.

3.3.9 Grafiek 9: Average scheduled flight time compared the average delay per carrier from 2011 to 2012



Conclusie grafiek 9: Average scheduled flight time compared the average delay per carrier from 2011 to 2012

Hierin wordt er per vliegtuigmaatschappij per jaar gekeken naar de gemiddelde geplande vluchttijd en dat wordt vergelijken met de gemiddelde vertraging.

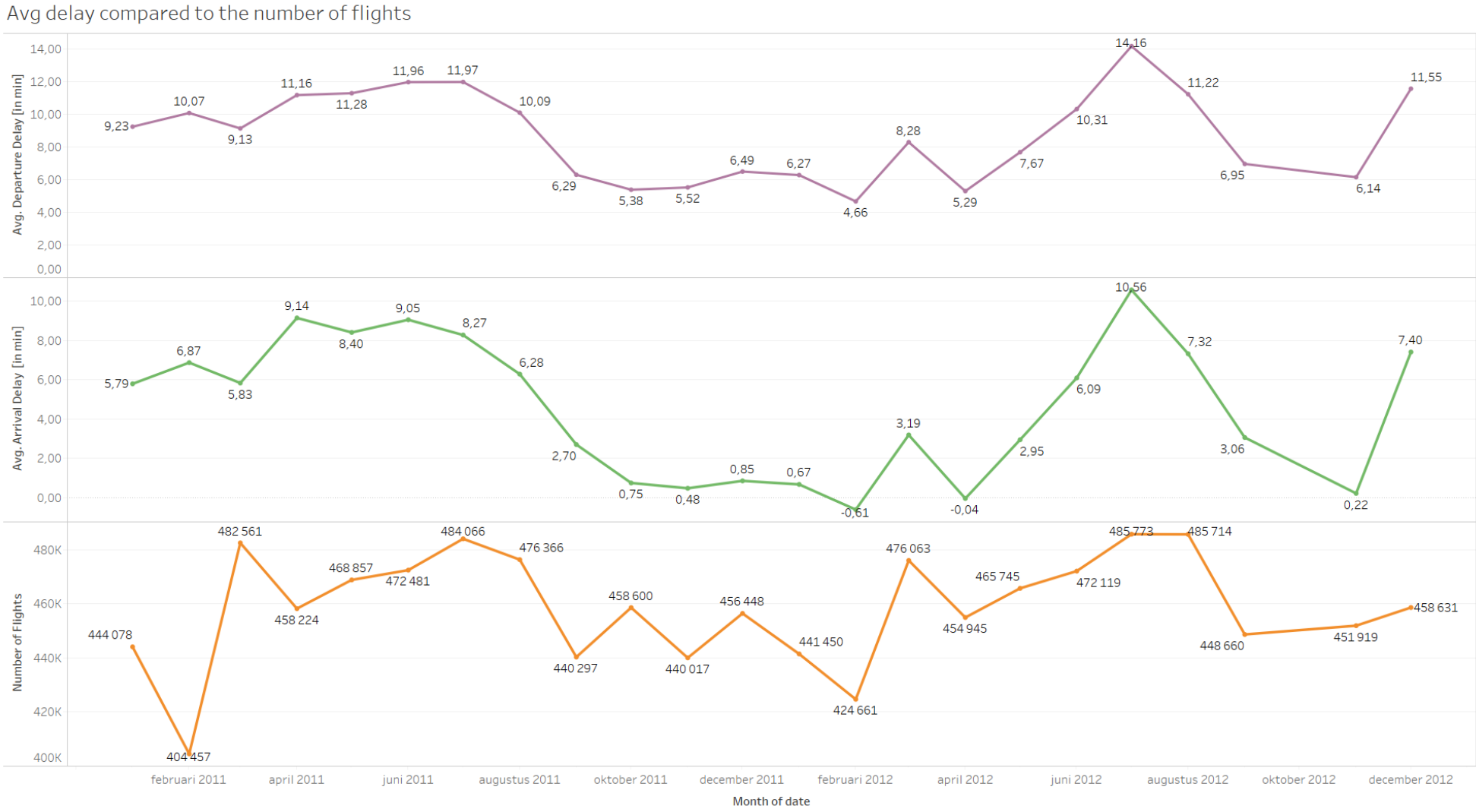
Als je naar de data kijkt zie je dat *Alaska Airlines* de vliegtuigmaatschappij is met de minste vertraging. De vraag is echter klopt deze data wel?

United Airlines is de maatschappij die het slecht scoort ze zijn van gemiddeld 12,04 min vertraging in 2011 gegaan naar 18,04 min in 2012 dat is een stijging van 49,83 %

US Airways is de betere leerling van de klas hun gemiddelde vertraging is van 9,83 min in 2011 naar 2,35 min gegaan in 2012, dat is een daling van 418,30 %.

De vliegtuigmaatschappij waar je gemiddeld het langst wacht voor je op de vlucht kan is *ExpressJet Airlines* met 23,59 min vertraging in 2011 en 20,68 min vertraging in 2012.

3.3.10 Grafiek 10: The average departure delay and the average arrival delay compared to the number of flights per month

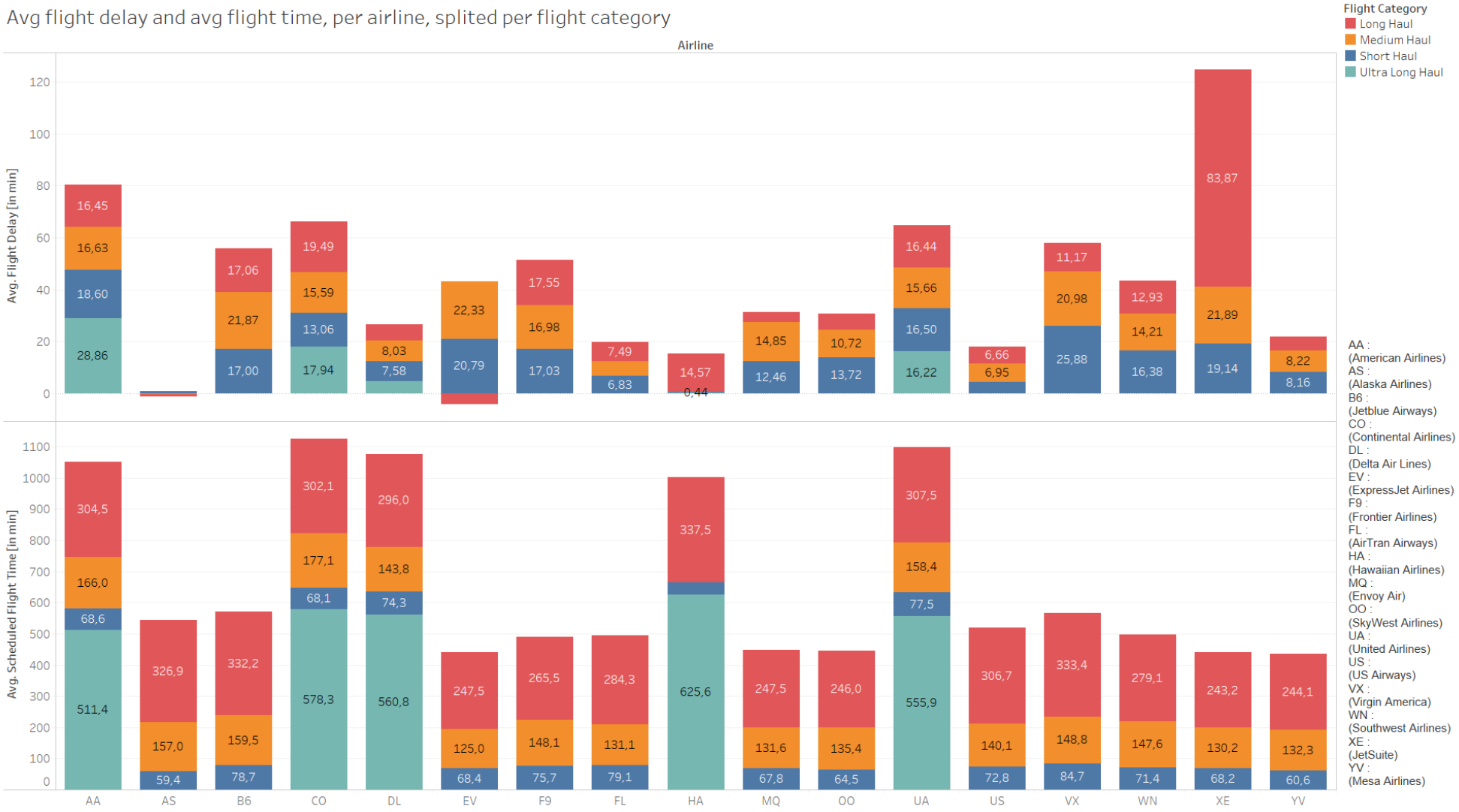


Conclusie grafiek 10: The average departure delay and the average arrival delay compared to the number of flights per month

In deze grafiek hebben we gekozen om te kijken per maand naar de gemiddelde vertraging voor het vertrekken en de vertraging dat je hebt bij het aankomen ten opzichte van het aantal vluchten.

Zoals je kan zien is de vertraging dat je hebt voor het vertrekken zeker niet gelijk aan de vertraging bij het aankomen. Wat je wel kan opmerken is dat er vanaf de maanden augustus tot en met februari een daling is in het aantal vluchten en dat er in die maanden ook minder vertraging is. Meestal in de maand juli zijn er meer vluchten vanwege de vakantie en dat zorgt dan ook voor de grootste gemiddelde vertragingen.

3.3.11 Grafiek 11: The average flight time and average delay per flight category compared with each carrier



Conclusie grafiek 11: The average flight time and average delay per flight category compared with each carrier

Wat is *Short Haul, Medium Haul* … ? Dit zijn categorieën waar we vluchten vanaf een bepaalde tijdsduur groeperen.

- *Short Haul*
  - Zijn vluchten waar de tijd korter is dan 1u30min
- *Medium Haul*
  - Zijn vluchten waar de tijd korter is dan 4u maar groter dan 1u30min
- *Short Haul*
  - Zijn vluchten waar de tijd korter is dan 8u maar groter dan 4u
- *Ultra Long Haul*
  - Zijn vluchten waar de tijd langer is dan 8u

Dus wat hebben we gedaan in deze grafiek? We hebben alle vluchten per maatschappij opgesplitst per categorie. Daarvan hebben we dan de gemiddelde tijdsduur van die categorie per maatschappij. Daar tegenover hebben we dan gekeken naar de gemiddelde vertraging dat je kan hebben als je een vlucht neemt van die categorie per maatschappij.

Zoals je ziet zijn er maar 5 maatschappijen *die Ultra Long Haul* hebben:

- *American Airlines*
- *Continental Airlines*
- *Delta Airlines*
- *Hawaiian Airlines*
- *United Airlines*

4. Machine learning met Scikit learn

4.1 Inleiding

Voor het machine learning gedeelte van de opdracht met sklearn hebben we gekozen om de vertraging van de aankomsttijd te voorspelen. We zijn begonnen met regressie, om hier een goed resultaat te bekomen is het echter nodig om de vertraging van de vertrektijd als parameter te gebruiken.

Daarna hebben we classificatie toegepast, zonder gebruik te maken van de vertraging van de vertrektijd om te voorspellen of een vlucht meer of minder dan 5 minuten vertraging zal hebben.

4.2 Regressie

4.2.1 Laden van de data

```
1.  BASIC_PATH = '../Data/'
2.  ALL_FILES = BASIC_PATH + '*.csv'
3.
4.  import pandas as pd
5.  import numpy as np
6.  import glob as glob
7.
8.  def readAllFiles():
9.      files = glob.glob(ALL_FILES)
10.     frames = []
11.
12.     for file in files:
13.         df = pd.read_csv(file, index_col = 0)
14.         frames.append(df)
15.
16.     return pd.concat(frames)
17.
18. df = readAllFiles()
```

We beginnen met het inlezen van alle data die we in de vorige stap opgekuist hebben.

4.2.2 Kolommen verwijderen die we niet gaan gebruiken voor ML

```
1.  df['month'] = pd.to_datetime(df.date).map(lambda x: x.month)
2.
3.  df = df.drop(['date',
4.                'airline_code',
5.                'departure_airport',
6.                'departure_lat',
7.                'departure_lon',
8.                'departure_schedule',
9.                'arrival_airport',
10.               'arrival_lat',
11.               'arrival_lon',
12.               'arrival_schedule',
13.               'departure_actual',
14.               'arrival_actual',
15.               'arrival_tz',
16.               'departure_tz'], axis=1)
```

Niet alle kolommen zijn even nuttig voor machine learning, bovendien kunnen we niet alles gebruiken omwille van performantie en geheugenlimieten.

4.2.3 Verdeel dataset in test en train, x en y

```
from sklearn.model_selection import train_test_split
import gc

X = df.loc[:,['airline',
              'departure_state',
              'departure_delay',
              'distance',
              'speed',
              'month']]
X = pd.get_dummies(X, columns=['airline',
                               'departure_state',
                               'month'])

y = df.loc[:,['arrival_delay']]

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

#make place in memory
del df
del X
del y
gc.collect()
```

We willen y = arrival_delay voorspellen en dat doen we op basis van de data in X. We gebruiken pandas.get_dummies voor one-hot encoding van niet lineaire data. Daarna splitsen we de dataset op in een train en een test dataset.

4.2.4 Random Forest Regressor model

Aanmaken van het model:

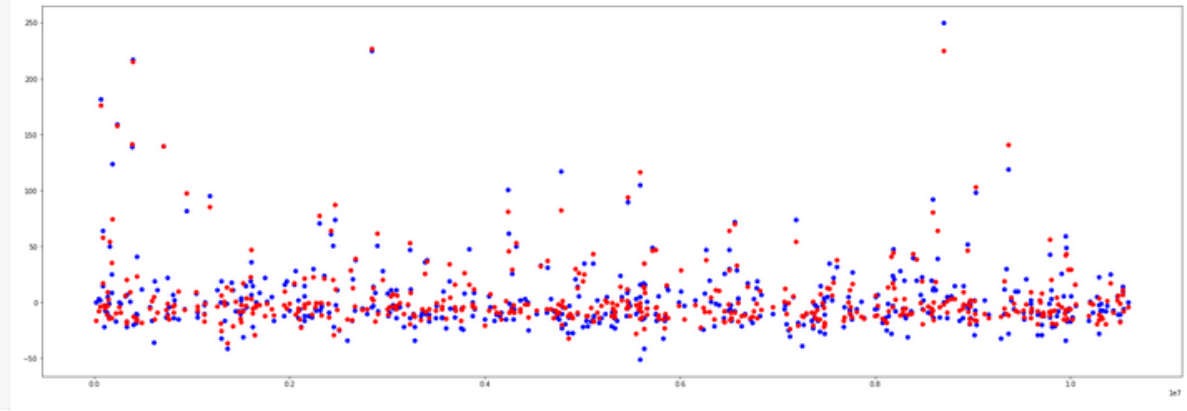
```
from sklearn.ensemble import RandomForestRegressor

rfrm = RandomForestRegressor()
rfrm.fit(X_train, y_train.values.ravel())

rfrm_y_predict = rfrm.predict(X_test)
```

Testen van het model:

```
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error

# The mean squared error
print("Mean squared error: %.2f" % mean_squared_error(y_test, rfrm_y_predict))

# The mean absolute error
print("Mean absolute error: %.2f" % mean_absolute_error(y_test, rfrm_y_predict))

# Explained variance score: 1 is perfect prediction
print('Variance score: %.5f' % r2_score(y_test, rfrm_y_predict))

#Graph
plt.scatter(y_test.iloc[:500,:].index, y_test.iloc[:500,:].arrival_delay, c='b')
predictDf = pd.DataFrame( data=rfrm_y_predict, index=y_test.index)
plt.scatter(predictDf.iloc[:500,:].index, predictDf.iloc[:500,0], c='r')
plt.rcParams['figure.figsize'] = [30, 10]
plt.show()
```

Uitvoer:

```
Mean squared error: 173.43
Mean absolute error: 9.17
Variance score: 0.86946
```



Zowel de variance score als beide error rates zijn goed van dit model.

4.2.5 Lineair Regression model

Aanmaken van het model:

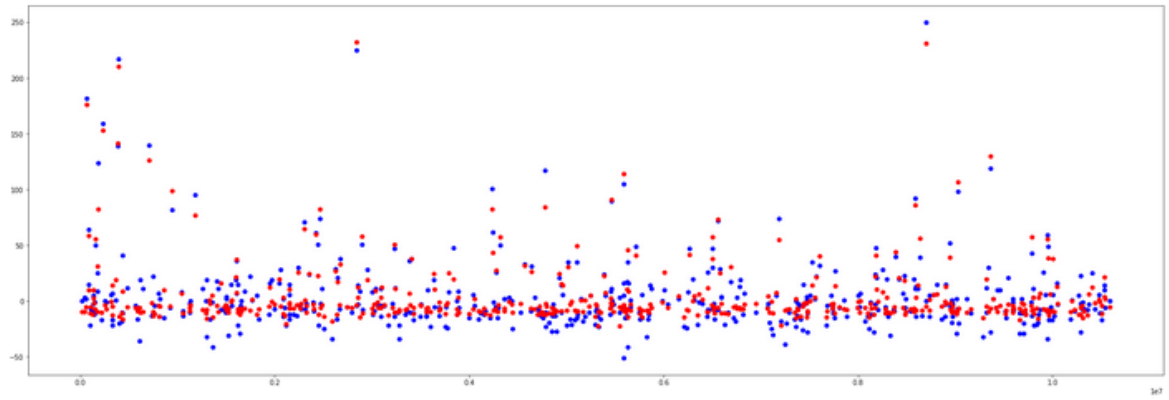
```
1.  from sklearn.linear_model import LinearRegression
2.
3.  lrm = LinearRegression()
4.  lrm.fit(X_train, y_train)
5.
6.  lrm_y_predict = lrm.predict(X_test)
```

Testen van het model:

```
1.  import matplotlib.pyplot as plt
2.  from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
3.
4.  # The mean squared error
5.  print("Mean squared error: %.2f" % mean_squared_error(y_test, lrm_y_predict))
6.
7.  # The mean absolute error
8.  print("Mean absolute error: %.2f" % mean_absolute_error(y_test, lrm_y_predict))
9.
10. # Explained variance score: 1 is perfect prediction
11. print('Variance score: %.5f' % r2_score(y_test, lrm_y_predict))
12.
13. #Graph
14. plt.scatter(y_test.iloc[:500,:].index, y_test.iloc[:500,:].arrival_delay, c='b')
15. predictDf = pd.DataFrame( data=lrm_y_predict, index=y_test.index)
16. plt.scatter(predictDf.iloc[:500,:].index, predictDf.iloc[:500,0], c='r')
17. plt.rcParams['figure.figsize'] = [30, 10]
18. plt.show()
```

Uitvoer:

```
Mean squared error: 154.39
Mean absolute error: 8.69
Variance score: 0.88379
```



Met het lineaire regressie model behalen we de beste resultaten met een variance score van 88%.

4.2.6 Ridge Regression model

Aanmaken van het model:

```
1.  from sklearn.linear_model import Ridge
2.
3.  rrm = Ridge(alpha=0.01, normalize=True)
4.  rrm.fit(X_train, y_train.values.ravel())
5.
6.  rrm_y_predict = rrm.predict(X_test)
```

Testen van het model:

```
1.  import matplotlib.pyplot as plt
2.  from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
3.
4.  # The mean squared error
5.  print("Mean squared error: %.2f" % mean_squared_error(y_test, rrm_y_predict))
6.
7.  # The mean absolute error
8.  print("Mean absolute error: %.2f" % mean_absolute_error(y_test, rrm_y_predict))
9.
10. # Explained variance score: 1 is perfect prediction
11. print('Variance score: %.5f' % r2_score(y_test, rrm_y_predict))
```

Uitvoer:

```
Mean squared error: 154.50
Mean absolute error: 8.69
Variance score: 0.88371
```

Ridge regression gaf de beste resultaten indien we een kleine alpha waarde gebruikten omdat het dan het meest aanleunt bij lineaire regressie.

4.2.7 Lasso Regression model

Aanmaken van het model:

```
1.  from sklearn.linear_model import Lasso
2.
3.  lrm = Lasso(alpha=0.01, normalize=True)
4.  lrm.fit(X_train, y_train.values.ravel())
5.
6.  lrm_y_predict = lrm.predict(X_test)
```

Testen van het model:

```
1.  import matplotlib.pyplot as plt
2.  from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
3.
4.  # The mean squared error
5.  print("Mean squared error: %.2f" % mean_squared_error(y_test, lrm_y_predict))
6.
7.  # The mean absolute error
8.  print("Mean absolute error: %.2f" % mean_absolute_error(y_test, lrm_y_predict))
9.
10. # Explained variance score: 1 is perfect prediction
11. print('Variance score: %.5f' % r2_score(y_test, lrm_y_predict))
```

Uitvoer:

```
Mean squared error: 949.34
Mean absolute error: 17.65
Variance score: 0.28543
```

Lasso regressie geeft, zowel bij een kleine als bij een grote waarde voor alpha, slechte resultaten.

4.3 Classificatie

Regressie geeft goede resultaten wanneer de vertraging bij vertrek bekend was. Maar zonder deze kolom was de uitkomst niet bruikbaar. Om onze kansen te vergroten hebben we ook classificatie getest. Een vlucht die meer dan 5 minuten over tijd is wordt geclassificeerd als te laat.

4.3.1 Dataset voorbereiden

```
df['month'] = pd.to_datetime(df.date).map(lambda x: x.month)
df["arrival_delay"] = df["arrival_delay"].map(lambda delay: 0 if delay < 5 else 1)
df.departure_schedule = pd.to_datetime(df.departure_schedule)
df.arrival_schedule = pd.to_datetime(df.arrival_schedule)
df.departure_schedule = df.departure_schedule.map(lambda x: x.hour * 60 + x.minute)

df.arrival_schedule = df.arrival_schedule.map(lambda x: x.hour * 60 + x.minute)
df = df.drop(['date',
              'airline_code',
              'departure_delay',
              'departure_state',
              'departure_lat',
              'departure_lon',
              'arrival_state',
              'arrival_lat',
              'arrival_lon',
              'departure_actual',
              'arrival_actual',
              'arrival_tz',
              'departure_tz'], axis=1)
```

De ruwe data wordt omgezet naar formaten dat scikit verstaat. De vertraging bij aankomst wordt omgezet naar 0 of 1 dat “op tijd” en “te laat” voorstelt (data binning). De onnodige kolommen worden er ook uitgehaald.

4.3.2 Verdeel dataset in test en train, x en y

```
from sklearn.model_selection import train_test_split

X = df.loc[:,['airline',
              'departure_airport',
              'arrival_airport',
              'departure_schedule',
              'arrival_schedule',
              'distance',
              'speed',
              'month']]
X = pd.get_dummies(X, columns=['airline',
                               'departure_airport',
                               'arrival_airport',
                               'month'])

y = df.loc[:,['arrival_delay']]

X_train, X_test, y_train, y_test = train_test_split(X, y)
```

4.3.3 Random Forrest Classifier

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=100)
rfc.fit(X_train, y_train.values.ravel())
```

4.3.4 K-Nearest Neighbors Classifier

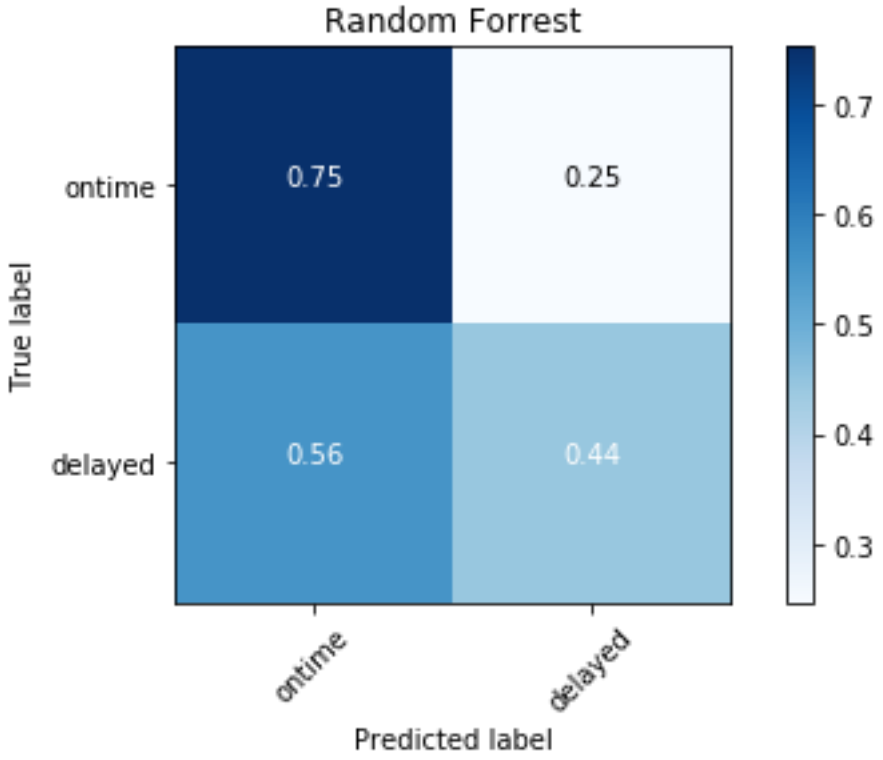
```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(X_train, y_train.values.ravel())
```

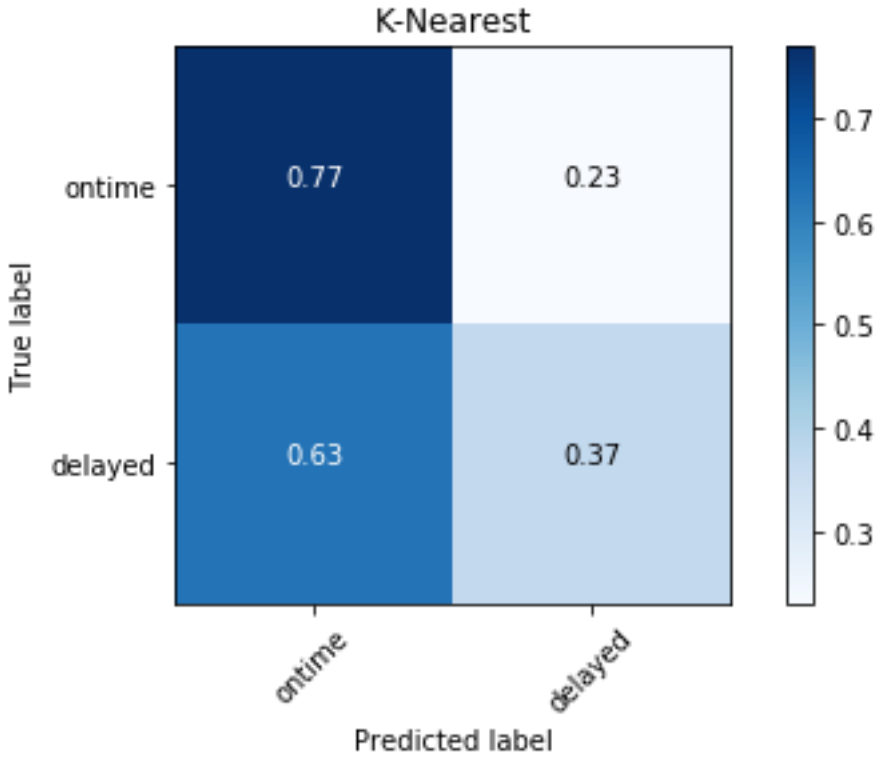
We hebben Random Forrest en K-Nearest Neighbors models gebruikt om dat deze de beste resultaten gaven.

4.3.5 Voorspelling maken met de modellen

```
import itertools
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

def plot_confusion_matrix(cm, classes,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

class_names = ["ontime", "delayed"]

models = [
    { "name": "Random Forrest", "model": rfc },
    { "name": "K-Nearest", "model": knn }
]
for model in models:
    # Compute confusion matrix
    pred = model["model"].predict(X_test)
    cnf_matrix = confusion_matrix(y_test, pred)
    np.set_printoptions(precision=2)
    plt.figure()
    plot_confusion_matrix(cnf_matrix, classes=class_names, title=model["name"])
    plt.show()
```

Om inzicht tot de resultaten te krijgen genereren we genormaliseerde confusion matrices van de voorspellingen tegenover de realiteit. Zoals je kan zien is het voorspellen van vertraagde vluchten het moeilijkste voor beide modellen.


Random Forrest
0.75
0.25
0.56
0.44
ontime
delayed
True label
ontime
delayed
Predicted label
0.7
0.6
0.5
0.4
0.3


K-Nearest
0.77
0.23
0.63
0.37
ontime
delayed
True label
ontime
delayed
Predicted label
0.7
0.6
0.5
0.4
0.3

5. NetworkX

5.1 Laad de dataset

```
1. import math
2. import json
3. import numpy as np
4. import pandas as pd
5. import networkx as nx
6. import cartopy.crs as ccrs
7. import matplotlib.pyplot as plt
8. from IPython.display import Image
9. %matplotlib inline
10.
11. import glob
12.
13. BASIC_PATH = './Data/'
14. ALL_FILES = BASIC_PATH + '*.csv'
15.
16. def readOneFile(url):
17.     return pd.read_csv(url, index_col = 0)
18.
19. data = readOneFile(BASIC_PATH + "flights_2010_9.csv")
20. flights = pd.DataFrame(data=data)
21. flights
```

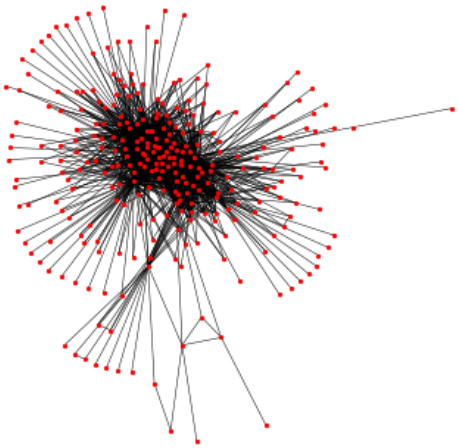
5.2 Routes tussen luchthavens

```
1. edges = flights[['departure_airport', 'arrival_airport']].values
2. edges
```

Uit de vluchten leiden we de routes tussen luchthavens af.

5.3 Grafiek van alle luchthavens en hun routes

```
g = nx.from_edgelist(edges)
fig, ax = plt.subplots(1, 1, figsize=(6, 6))
nx.draw_networkx(g, ax=ax, with_labels=False,
                 node_size=5, width=.5)
ax.set_axis_off()
```



5.4 Vind de coördinaten van elke luchthaven

```
pos = {}
airports = flights[["departure_airport", "departure_lat", "departure_lon"]].drop_dup
licates()
for i, row in airports.iterrows():
    airport = row["departure_airport"]
    pos[airport] = (row["departure_lon"], row["departure_lat"])
```

Elke vlucht heeft coördinaten bij het vertrek punt en de aankomst. We kunnen dus de locatie van elke luchthaven hieruit afleiden.

5.5 Plaats de routes op kaart

```
deg = nx.degree(g)
sizes = [5 * deg[iata] for iata in g.nodes]

labels = {iata: iata if deg[iata] >= 60 else ''
          for iata in g.nodes}

crs = ccrs.PlateCarree()
fig, ax = plt.subplots(
    1, 1, figsize=(12, 8),
    subplot_kw=dict(projection=crs))

ax.coastlines()

ax.set_extent([-163, -67, 18, 64])
nx.draw_networkx(g, ax=ax,
                 font_size=14,
                 alpha=.5,
                 width=.07,
                 node_size=sizes,
                 labels=labels,
                 pos=pos,
                 node_color=sizes,
                 cmap=plt.cm.cool)
```

De volgende kaart bevat elke route en luchthaven in de dataset. De grote en kleur van een luchthaven geeft aan hoeveel connecties het bevat.