

AIBP - Invloed van COVID19 op het vliegverkeer

Project gemaakt door:

- Rens Pierloot
- Arthur De Groot
- Laurens Giesen

0. Loading packages en dataset

```
In [51]: import matplotlib inline
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('seaborn-darkgrid')
```

```
In [52]: df = pd.read_csv('covid_impact_on_airport_traffic.csv')
```

1. De data bekijken

```
In [53]: df.shape
```

```
Out[53]: (7247, 11)
```

de dataset bevat 7247 observations en 11 features.

```
In [54]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7247 entries, 0 to 7246
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ---
 0   AggregationMethod    7247 non-null   object
 1   Date                 7247 non-null   object
 2   Version              7247 non-null   float64
 3   AirportName          7247 non-null   object
 4   PercentOfBaseline    7247 non-null   int64
 5   Centroid             7247 non-null   object
 6   City                 7247 non-null   object
 7   State                7247 non-null   object
 8   ISO_3166_2           7247 non-null   object
 9   Country              7247 non-null   object
10  Geography            7247 non-null   object
dtypes: float64(1), int64(1), object(9)
memory usage: 622.9+ KB
```

We zien dat de dataset 9 features het datatype 'object' bevatten, 1 feature met datatype 'float' en 1 feature met het datatype 'integer'. We controleren ook even of er geen missing data is.

```
In [55]: df.isnull().sum()
Out[55]: Series(0, dtype: int64)
```

we veranderen het datatype naar:

- datetime van 'Date'
- string van 'AggregationMethod', 'AirportName', 'City', 'State', 'ISO_3166_2' en 'Country'

```
In [56]: df['Date'] = pd.to_datetime(df['Date'])
df['AggregationMethod'] = df['AggregationMethod'].astype('string')
df['AirportName'] = df['AirportName'].astype('string')
df['City'] = df['City'].astype('string')
df['State'] = df['State'].astype('string')
df['ISO_3166_2'] = df['ISO_3166_2'].astype('string')
df['Country'] = df['Country'].astype('string')
```

```
In [57]: df.head()
```

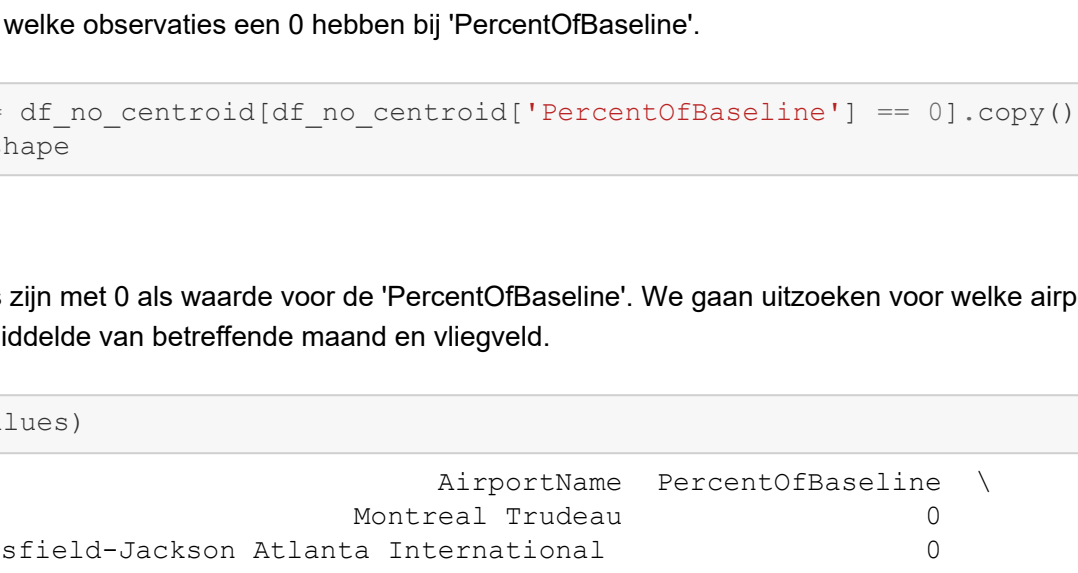
- We zien een minimum 'PercentOfBaseline' van 0. Dit wil zeggen dat er ofwel data missing is of dat er die dag geen vliegtuigen gevlogen hebben (wat vrij onwaarschijnlijk is). We zoeken dit verder uit.
- De 'the' na 'United States of America' is overbodig en zal ons later in de problemen brengen bij het verwerken van de data in Tableau. We gaan er 'the' dus moeten uithalen.
- Centroid heeft een 'longitude' en 'latitude', maar is een object. We kunnen deze waarden voorlopig moeilijk gebruiken en gaan deze feature dus splitten. Latitude en longitude zijn een paar nummers (coördinaten) gebruikt om een positie op het vlak van een geografie coördinatenstelsel te beschrijven. De getallen zijn in decimale waarden die variëren van -90 tot 90 voor latitude en -180 tot 180 voor longitude. Het eerste nummer is dus longitude en het tweede nummer is latitude.

```
In [58]: df.describe(include='all')
```

AggregationMethod	Date	Version	AirportName	PercentOfBaseline	Centroid	City	State	ISO_3166_2	Co
count	7247	7247	7247.0	7247	7247.000000	7247	7247	7247	7247
unique	1	262	NaN	28	NaN	28	27	23	23
top	Daily	2020-03-27 00:00:00	NaN	John F. Kennedy International	NaN	PONT(151.180067713813 33.9459774986125)	New York	California	US-CA
freq	7247	28	NaN	282	NaN	262	523	523	523
first	NaN	2020-03-16 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
last	NaN	2020-10-02 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	1.0	NaN	66.651442	NaN	NaN	NaN	NaN
std	NaN	NaN	0.0	NaN	22.134433	NaN	NaN	NaN	NaN
min	NaN	NaN	1.0	NaN	0.000000	NaN	NaN	NaN	NaN
25%	NaN	NaN	1.0	NaN	53.000000	NaN	NaN	NaN	NaN
50%	NaN	NaN	1.0	NaN	67.000000	NaN	NaN	NaN	NaN
75%	NaN	NaN	1.0	NaN	84.000000	NaN	NaN	NaN	NaN
max	NaN	NaN	1.0	NaN	100.000000	NaN	NaN	NaN	NaN

Commentaar:

- We zien dat alle observaties van 'AggregationMethod' hetzelfde zijn. We interpretern deze feature dus als overbodig en zullen deze verwijderen
- 'Version' heeft een standaardafwijking van 0, wat wil zeggen dat al deze waarden ook hetzelfde zijn. Deze feature gaan we dus ook verwijderen
- de feature 'Geography' toont de oppervlakte van het vliegveld. Bij ons onderzoek Victoria was voldoende uit het centraal punt van het vliegveld. Deze feature biedt geen meerwaarde en we zullen deze dus verwijderen.



2. Data cleaning

2.1. Verwijderen van overbodige features

```
In [59]: df = df.drop(['AggregationMethod', 'Version', 'Geography'],1)
```

2.2. Opsplitsen van Centroid in Longitude en Latitude

Het formaat die we gaan toepassen is als volgt:

- r: Raw string notation
- ~: Het getal voor we naar op zoek gaan is mogelijk vooraf gegaan met een min teken
- \d+: Zoek alle decimale waarden tot de onderbroken wordt door iets anders
- \.: Zoek een punt

```
In [60]: import re
df_no_centroid = df.copy()
p = re.compile(r'~?(\d+\.?\d+)?')
df_no_centroid['Longitude']=0.00
df_no_centroid['Latitude']=0.00

for i in df.index:
    coordinates = p.findall(df_no_centroid['Centroid'][i])
    df_no_centroid['Longitude'][i] = coordinates[0]
    df_no_centroid['Latitude'][i] = coordinates[1]

df_no_centroid = df_no_centroid.drop(['Centroid', 1])
```

<ipython-input-60-604574603e1a>:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df_no_centroid['Longitude'][i] = coordinates[0]

<ipython-input-60-604574603e1a>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df_no_centroid['Latitude'][i] = coordinates[1]

2.3. 'the' Weg halen uit de observations van de United States

```
In [61]: df_no_centroid.loc[df_no_centroid['Country'] == "United States of America (the)", ['Country']] = "United States of America"
```

2.4. Behandelen van missing data

We beginnen met te zoeken welke observaties een 0 hebben bij 'PercentOfBaseline'.

```
In [62]: df_only_zero_values = df_no_centroid[df_no_centroid['PercentOfBaseline'] == 0].copy()
df_only_zero_values.shape
```

```
Out[62]: (4, 9)
```

We zien dat er 4 observaties zijn met 0 als waarde voor de 'PercentOfBaseline'. We gaan uitzoeken welke airports dit is en deze opzien met de maandaggen in welke periode van betreffende maand en vliegveld.

```
In [63]: print(df_only_zero_values)

      Date                AirportName  PercentOfBaseline  \
2670 2020-08-17      Montreal Trudeau                      0
3904 2020-08-17  Hartsfield-Jackson Atlanta International    0
6307 2020-08-17      John F. Kennedy International          0
6469 2020-08-17  Dallas/Fort Worth International            0

      City  State ISO_3166_2  Country  Longitude  \
2670  Dorval  Quebec  CA-QC  United States of America  -73.742412
3904  College Park  Georgia  US-GA  United States of America  -84.427919
6307  New York  New York  US-NY  United States of America  -73.778447
6469  Grapevine  Texas  US-TX  United States of America  -97.039498

      Latitude
2670  45.467844
3904  33.641076
6307  40.646027
6469  32.894059
```

We zien dat alle nulwaarden van dezelfde dag zijn, we zoeken dit verder uit.

```
In [64]: date = df_only_zero_values['Date'][2670]
print(df_no_centroid[df_no_centroid['Date'] == date])

      Date                AirportName  PercentOfBaseline  \
2670 2020-08-17      Montreal Trudeau                      0
2845 2020-08-17      Los Angeles International              1
3904 2020-08-17  Hartsfield-Jackson Atlanta International    0
6307 2020-08-17      John F. Kennedy International          0
6469 2020-08-17  Dallas/Fort Worth International            0

      City  State ISO_3166_2  Country  Longitude  \
2670  Dorval  Quebec  CA-QC  United States of America  -73.742412
2845  Los Angeles  California  US-CA  United States of America  -118.404993
3904  College Park  Georgia  US-GA  United States of America  -84.427919
6307  New York  New York  US-NY  United States of America  -73.778447
6469  Grapevine  Texas  US-TX  United States of America  -97.039498

      Latitude
2670  45.467844
2845  -118.404993
3904  33.641076
6307  40.646027
6469  32.894059
```

```
In [65]: df_17_aug = df_no_centroid[df_no_centroid['Date'] == date]

for index, row in df_17_aug.iterrows():
    month_avg_airport = df_no_centroid[df_no_centroid['Date'].dt.month == row['Date'].month] \
        & (df_no_centroid['AirportName'] == row['AirportName'])['PercentOfBaseline'].mean()

    df_no_centroid.loc[index, ['PercentOfBaseline']] = month_avg_airport
```

Nu gaan we nog een laatste controle doen of alles in de dataset klopt.

```
In [66]: df_no_centroid.describe(include='all')

Out[66]:
```

	Date	AirportName	PercentOfBaseline	City	State	ISO_3166_2	Country	Longitude	Latitude
count	7247	7247	7247.000000	7247	7247	7247	7247	7247.000000	7247.000000
unique	262	28	NaN	27	23	23	4	NaN	NaN
top	2020-03-27 00:00:00	John F. Kennedy International	NaN	New York	California	US-CA	United States of America	NaN	NaN
freq	28	262	NaN	523	523	523	4441	NaN	NaN
first	2020-03-16 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
last	2020-10-02 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	NaN	66.699235	NaN	NaN	NaN	-84.521801	35.553030
std	NaN	NaN	NaN	22.066109	NaN	NaN	NaN	50.348150	20.010337
min	NaN	NaN	NaN	4.000000	NaN	NaN	NaN	-157.918285	-33.945977
25%	NaN	NaN	NaN	67.000000	NaN	NaN	NaN	-114.013123	35.213689
50%	NaN	NaN	NaN	67.000000	NaN	NaN	NaN	-83.353731	40.691503
75%	NaN	NaN	NaN	84.000000	NaN	NaN	NaN	-74.048380	44.883017
max	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	151.180088	53.368783

```
In [67]: df_no_centroid.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7247 entries, 0 to 7246
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype
---  ---
 0   Date                 7247 non-null   datetime64[ns]
 1   AirportName          7247 non-null   string
 2   PercentOfBaseline    7247 non-null   float64
 3   City                 7247 non-null   string
 4   State                7247 non-null   string
 5   ISO_3166_2           7247 non-null   string
 6   Country              7247 non-null   string
 7   Longitude             7247 non-null   float64
 8   Latitude             7247 non-null   float64
dtypes: datetime64[ns](1), float64(3), string(5)
memory usage: 509.7 KB
```

3. Heatmaps

3.0. Imports en authenticatie

```
In [68]: API_KEY = "AizAsyD0rW0G6vYzHvMnMjG6I47Jz81GLyFE8E"
import gmaps
gmaps.configure(api_key=API_KEY)
```

3.1. globaal overzicht

3.1.1. Visualisatie van de positionering van de airports

We hebben een extra dataframe gemaakt met enkel de vliegvelden, anders zit er qua positionering te veel repetitie in en zal het laden van de map te lang duren. Daarna genereren we een map met een overzicht van alle vliegvelden met wat informatie erbij.

```
In [69]: airports = df_no_centroid[['AirportName', 'City', 'Latitude', 'Longitude', 'Country']]
airports_no_dup = airports.drop_duplicates()
airports_locations = airports_no_dup[['Latitude', 'Longitude']]

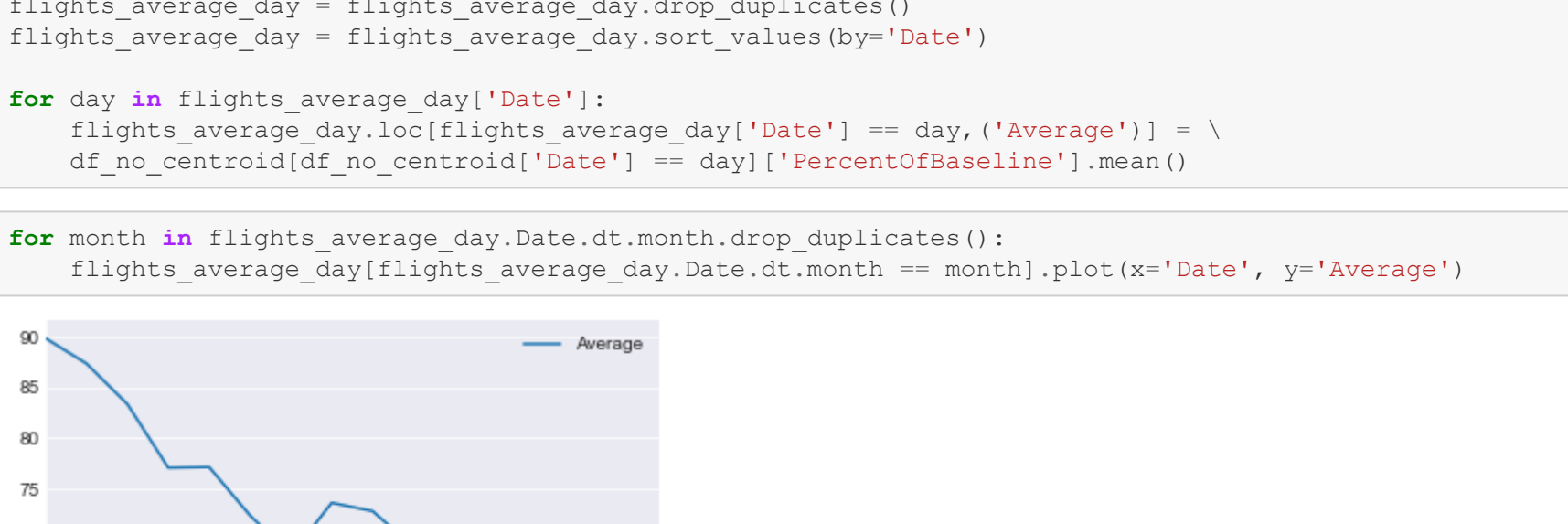
mapping = {}

for index, row in airports_no_dup.iterrows():
    mapping.append({'AirportName': row['AirportName'], 'City': row['City'], 'Country': row['Country']})

info_box_template = """
<div>
<div><b>Airport</b></div><div><b>AirportName</b></div>
<div><b>Country</b></div><div><b>City</b></div>
</div>
"""

airport_info = [info_box_template.format(**airport) for airport in mapping]

fig = gmaps.figure()
fig.add_layer(gmaps.marker_layer(airport_locations, info_box_content=airport_info))
fig
```



3.1.2 Heatmap van de de gemiddelde waarden per vliegveld gedurende de gemeten periode

We beginnen met een feature die het gemiddelde over de hele periode per vliegveld berekend.

```
In [70]: locations = df_no_centroid[['Latitude', 'Longitude']]
locations_no_dup = locations.drop_duplicates()

for airportname in airports_no_dup['AirportName']:
    airports_no_dup.loc[airports_no_dup['AirportName'] == airportname, 'Average'] = \
        df_no_centroid[df_no_centroid['AirportName'] == airportname]['PercentOfBaseline'].mean()

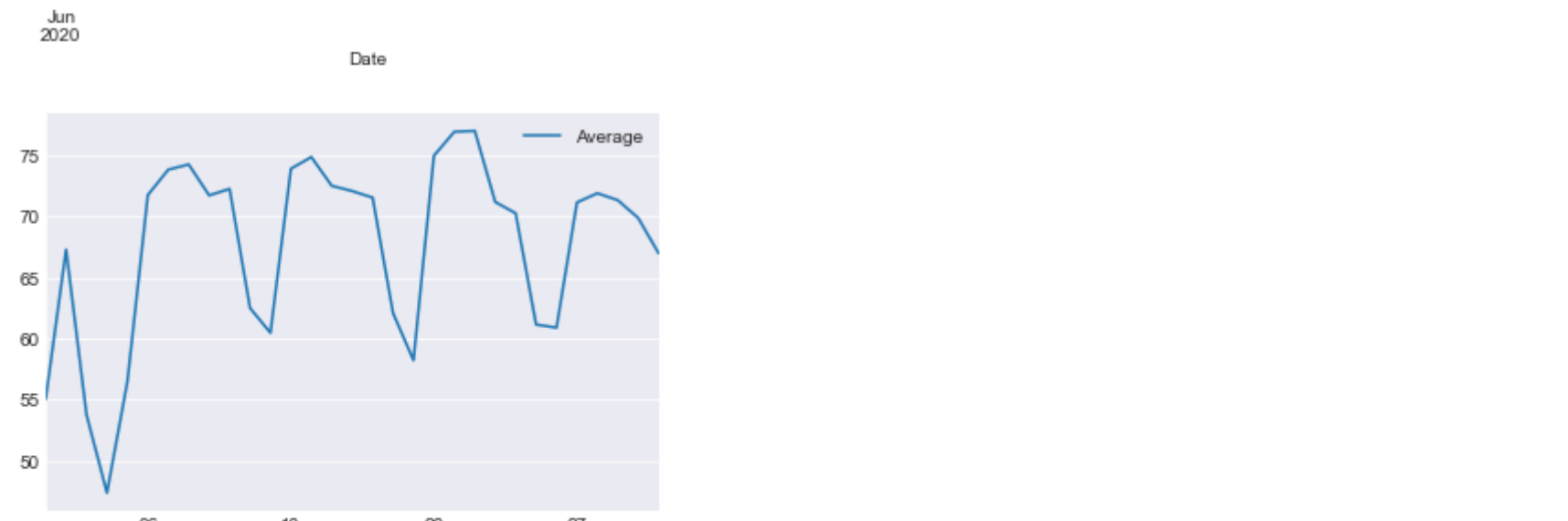
try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
self.obj[key] = infer_fill_value(value)
self.obj[locations_no_dup['Latitude']] = self.obj[locations_no_dup['Longitude']]
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
self.obj[iterms] = s
```

Met deze gemiddelden gaan we een map van maken met een 'max intensity' van 100 omdat dat onze maximum percent of baseline is.

```
In [71]: fig = gmaps.figure()
weights = airports_no_dup['Average']
heatmap_layer = gmaps.heatmap_layer(airport_locations, weights=weights, max_intensity=100)
fig.add_layer(heatmap_layer)
fig
```



Hoe roder de nodes op deze kaart, hoe minder zwaar het vliegveld getroffen is door de pandemie want hoe roder ze zijn hoe groter de percentOfBaseline. We zien dus dat Australië en Chili over het algemeen zwaarder getroffen zijn dan Amerika en Canada.

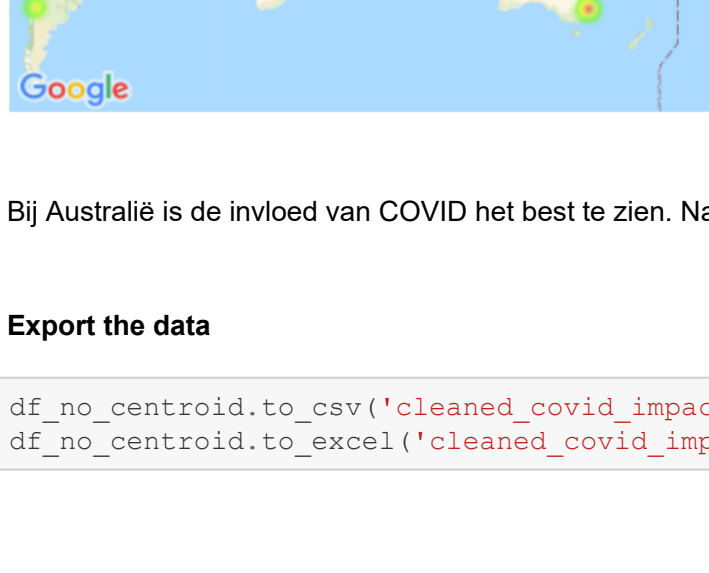
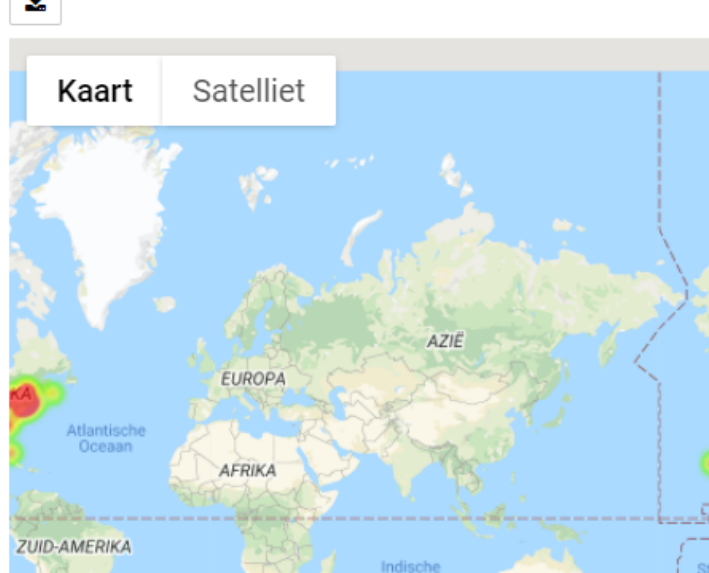
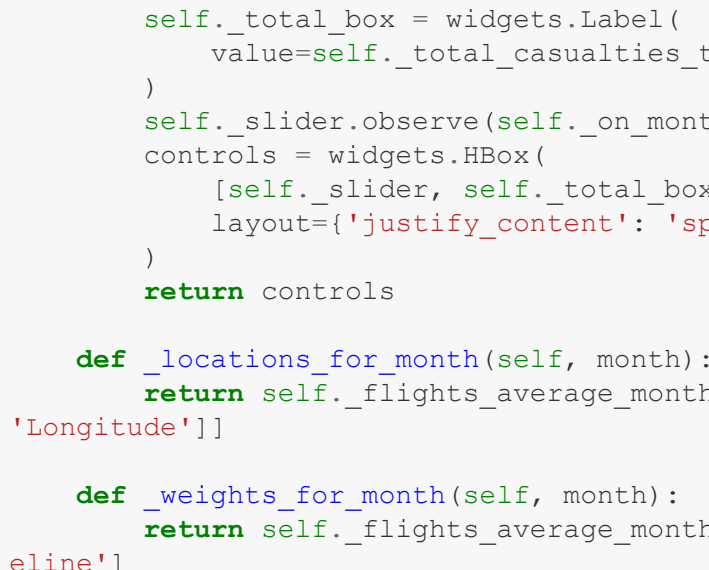
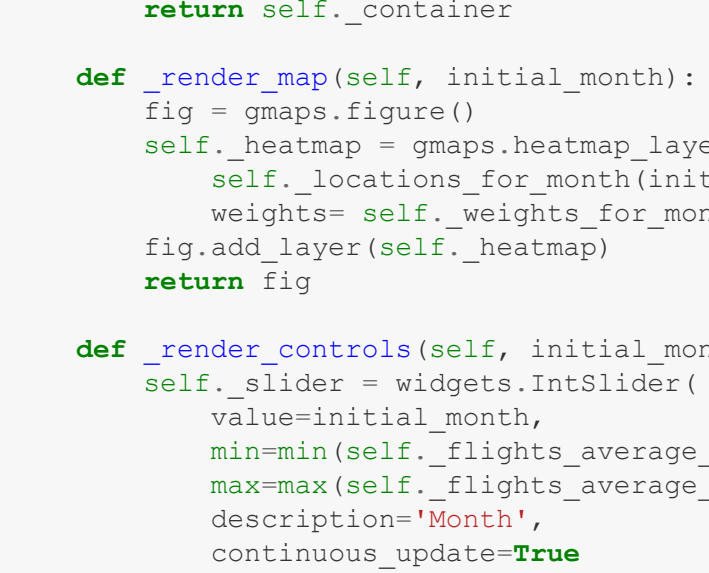
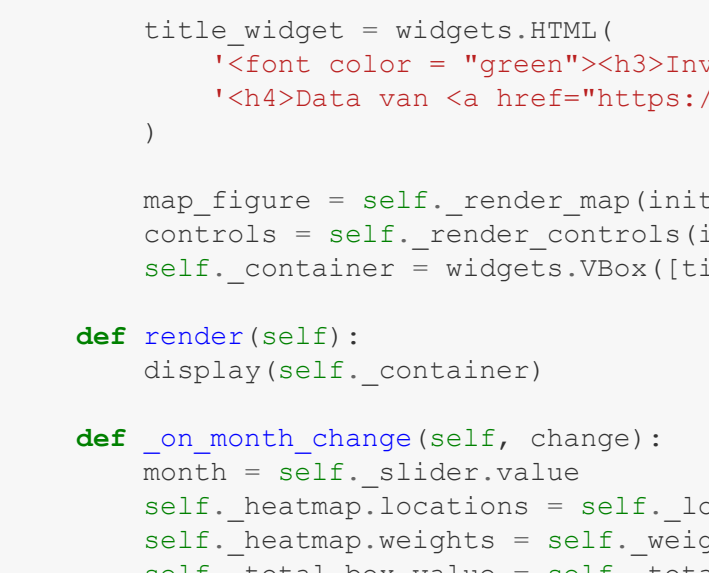
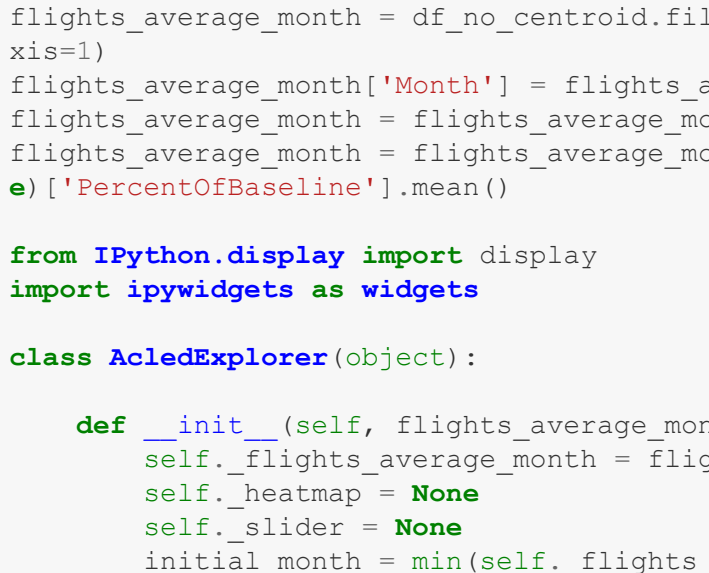
3.1.3. Linechart van de evolutie van vluchten (gemiddelde wereldwijd per dag)

Eerst berekenen we het gemiddelde per dag.

```
In [72]: flights_average_day = df_no_centroid.filter(['Date'], axis=1)
flights_average_day = flights_average_day.drop_duplicates()
flights_average_day = flights_average_day.sort_values(by='Date')

for day in flights_average_day['Date']:
    flights_average_day.loc[flights_average_day['Date'] == day, 'Average'] = \
        df_no_centroid[df_no_centroid['Date'] == day]['PercentOfBaseline'].mean()
```

```
In [73]: for month in flights_average_day.Date.dt.month.drop_duplicates():
    flights_average_day[flights_average_day.Date.dt.month == month].plot(x='Date', y='Average')
```



Commentaar:
We zien een sterke neerwaartse beweging op het begin van de pandemie en een grote outlier op 7 september.

Voor deze outlier zijn we begonnen met zoeken naar feestdagen op deze datum. Uiteindelijk hebben we ontdekt dat het in Amerika en Canada op die dag Labor Day is (Dag van de Arbeid). Dus het is vrij logisch dat er op deze dag minder gevlogen werd.

3.1.4. Interactieve heatmap met maandelijkse gemiddelden per vliegveld

```
In [74]: flights_average_month = df_no_centroid.filter(['Date', 'Longitude', 'Latitude', 'PercentOfBaseline'], axis=1)
flights_average_month['Month'] = flights_average_month['Date'].dt.month
flights_average_month = flights_average_month.drop_duplicates()
flights_average_month = flights_average_month.groupby(['Month', 'Latitude', 'Longitude'], as_index=False)['PercentOfBaseline'].mean()
```

```
from IPython.display import display
import ipynbwidgets as widgets

class AicledExplorer(object):
    def __init__(self, flights_average_month):
        self.flights_average_month = flights_average_month
        self_heatmap = None
        self_slider = None
        initial_month = min(self.flights_average_month['Month'])

        title_widget = widgets.HTML(
            '<font color = "green"><b>Invloed van covid op de vluchten</b></font>'
            '<br><b>Data van <div>kaire</div>=<div>https://www.geotab.com/</div></b></div>'
        )

        map_figure = self.render_map(initial_month)
        controls = self.render_controls(initial_month)
        self_container = widgets.VBox([title_widget, controls, map_figure])

        def render(self):
            display(self_container)

        def on_month_change(self, change):
            month = self_slider.value
            self_flights_locations = self_locations_for_month(month)
            self_heatmap = gmaps.heatmap_layer(
                self_locations_for_month(month),
                weights=self_weights_for_month(month),
                fig.add_layer(self_heatmap)
            )
            return fig

        def render_map(self, initial_month):
            self_slider = widgets.IntSlider(
                min=min(self.flights_average_month['Month']),
                max=max(self.flights_average_month['Month']),
                description='Month',
                continuous_update=True
            )
            self_title_box = widgets.Label(
                value=self_title_widget.text_for_month(initial_month)
            )
            self_slider.observe(self_on_month_change, names='value')
            controls = widgets.VBox([
                (self_slider, self_title_box),
                layout=('justify-content', 'space-between')
            ])
            return controls

        def locations_for_month(self, month):
            return self_flights_locations_for_month(self_flights_average_month['Month'] == month)[['Latitude', 'Longitude']]

        def weights_for_month(self, month):
            return self_flights_average_month[self_flights_average_month['Month'] == month]['PercentOfBaseline']

        def total_casualties_for_month(self, month):
            return int(self_flights_average_month[self_flights_average_month['Month'] == month]['Month']).count()

        def total_casualties_text_for_month(self, month):
            return "(Airport casualties).format(self.total_casualties_for_month(month))
```

```
AicledExplorer(flights_average_month).render()
```

Invloed van covid op de vluchten

Bij Australië is de invloed van COVID het best te zien. Naargelang we vooruitgaan in deze periode hoe meer invloed er aanwezig is.

Export the data

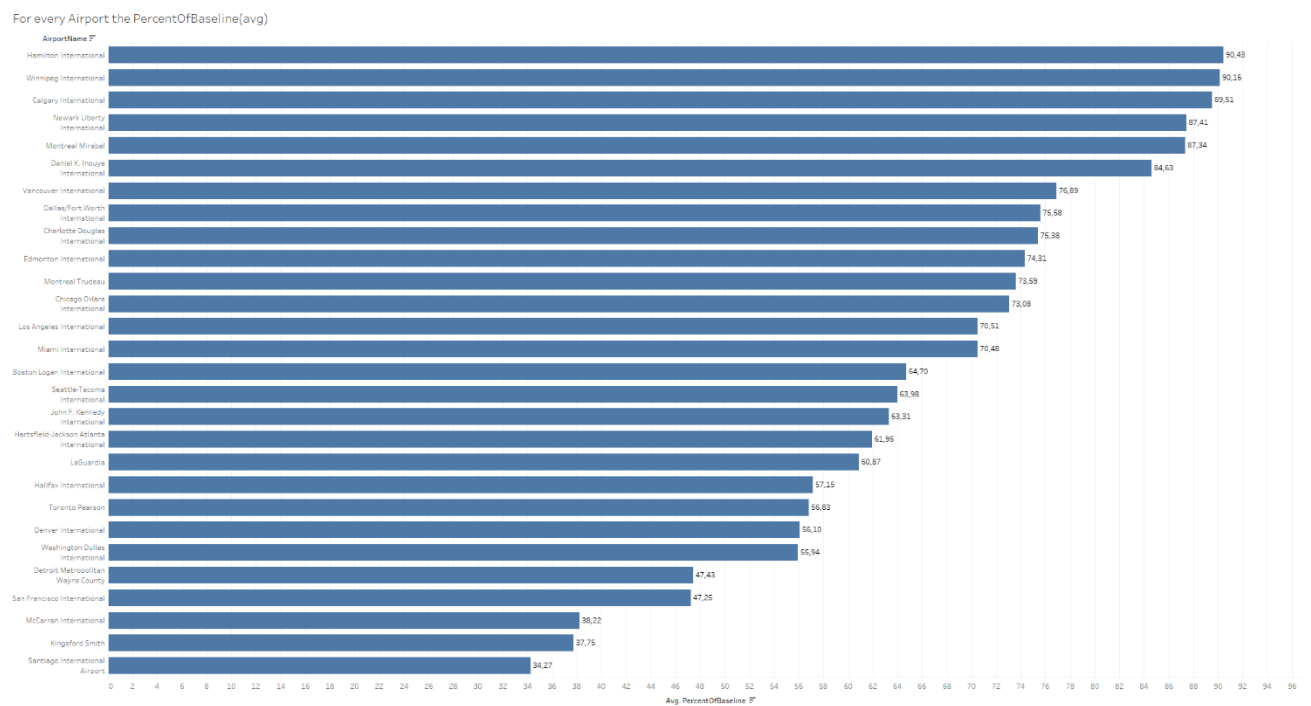
```
In [75]: df_no_centroid.to_csv('cleaned_covid_impact_on_airport_traffic.csv')
df_no_centroid.to_excel('cleaned_covid_impact_on_airport_traffic.xlsx')
```


Tableau:

Om de visualisatie nog wat uit te breiden hebben we ook via Tableau gewerkt. We hebben een 6-tal grafieken voorbereid:

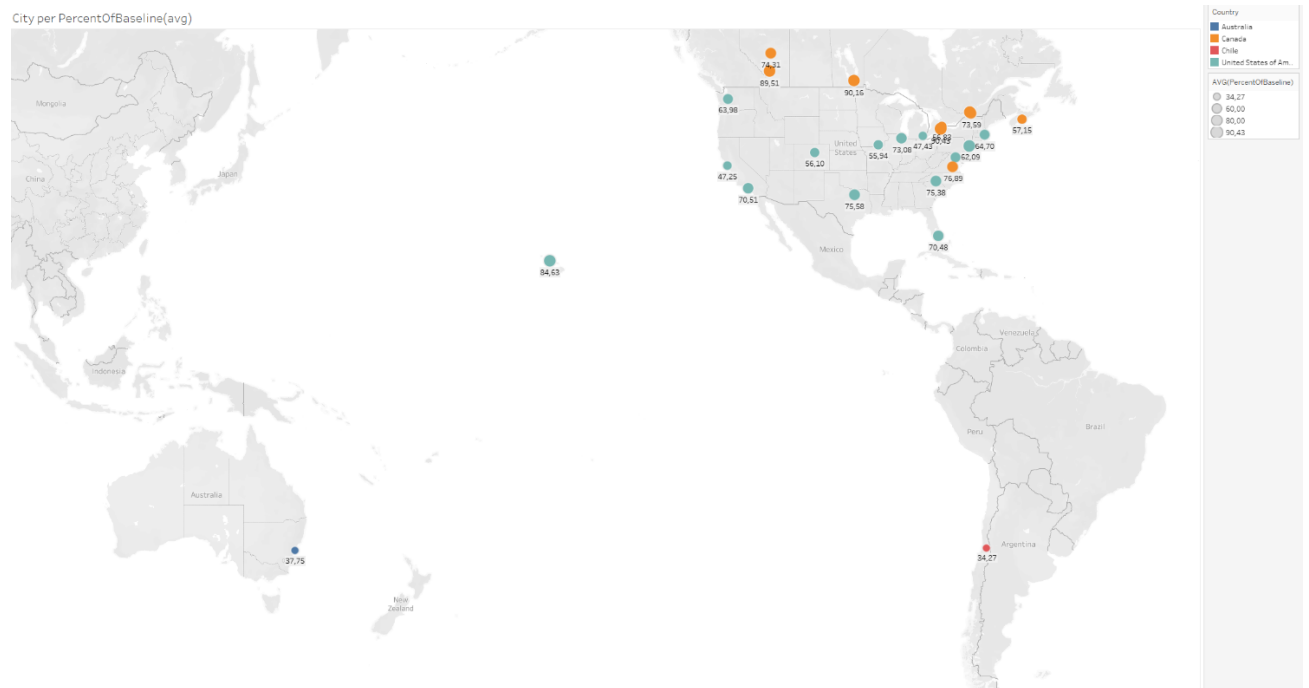
De 1^{ste} grafiek vertelt ons per luchthaven het gemiddelde percentage van vluchten tegenover voor Corona (1/02 – 15/03). Het percentage toont aan hoeveel % van de vluchten nog overgebleven zijn. Hoe hoger hoe beter dus. Als we kort kijken naar de grafiek kunnen we zien dat “Hamilton international” het minst getroffen is en “Santiago international” airport het meest.

Grafiek 1



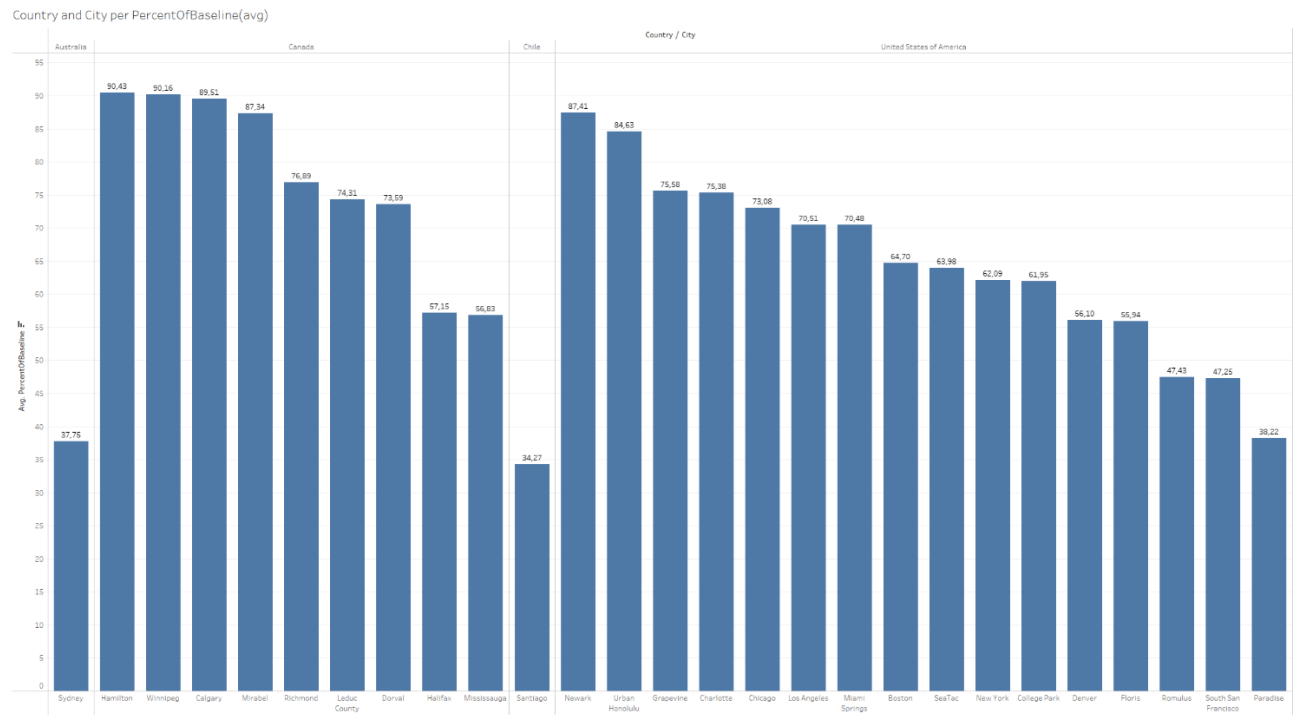
Op de 2^{de} grafiek tonen we hetzelfde als *grafiek 1* maar dan per stad, niet per luchthaven. Dit om een mooi geografisch beeld te kunnen weergeven. Zo zien we dat we ook data hebben in Australië en in Zuid-Amerika (Chili) maar die is dus niet zo heel representatief t.o.v. het land. Er zijn 2 mogelijke oorzaken van de lage percentages op die gebieden. Optie 1 is dat het reizen naar deze locaties minder mogelijk was door het beleid van de landen of bijvoorbeeld het sluiten van de grenzen voor toerisme. Optie 2 is dat op deze locaties minder data gegeven is waardoor binnenlandse vluchten minder getoond worden en hierdoor die 2 luchthavens zwaar getroffen lijken omdat die bijvoorbeeld normaal meer buitenlandse vluchten regelen.

Grafiek 2



De 3^{de} grafiek toont de hierboven staande gegevens in een staafdiagram geordend per land en dan van minste impact naar grootste.

Grafiek 3



De volgende grafiek is de meest nuttige, deze kan wel overweldigend zijn maar het toont de relevantste data. Je kan er per maand voor elk vliegveld de percentages zien aan de hand van staafdiagrammen. Zo kan je makkelijk kijken per luchthaven en bijvoorbeeld een tijdlijn interpreteren van de impact op het luchtverkeer.

Grafiek 4

For every month the AirportName with the PercentOfBaseline(avg)



De 5^{de} grafiek is dan weer hetzelfde als de 4^{de} maar met waardes in plaats van staafjes om het overweldigend gevoel wat weg te werken. Het verschil tussen grafiek 4 en 5 zit zich niet enkel in de waardes maar bevat ook de totalen, wat zeker ook handig kan zijn. Dit is eerder een samenvoeging van alle grafieken die we al hebben getoond.

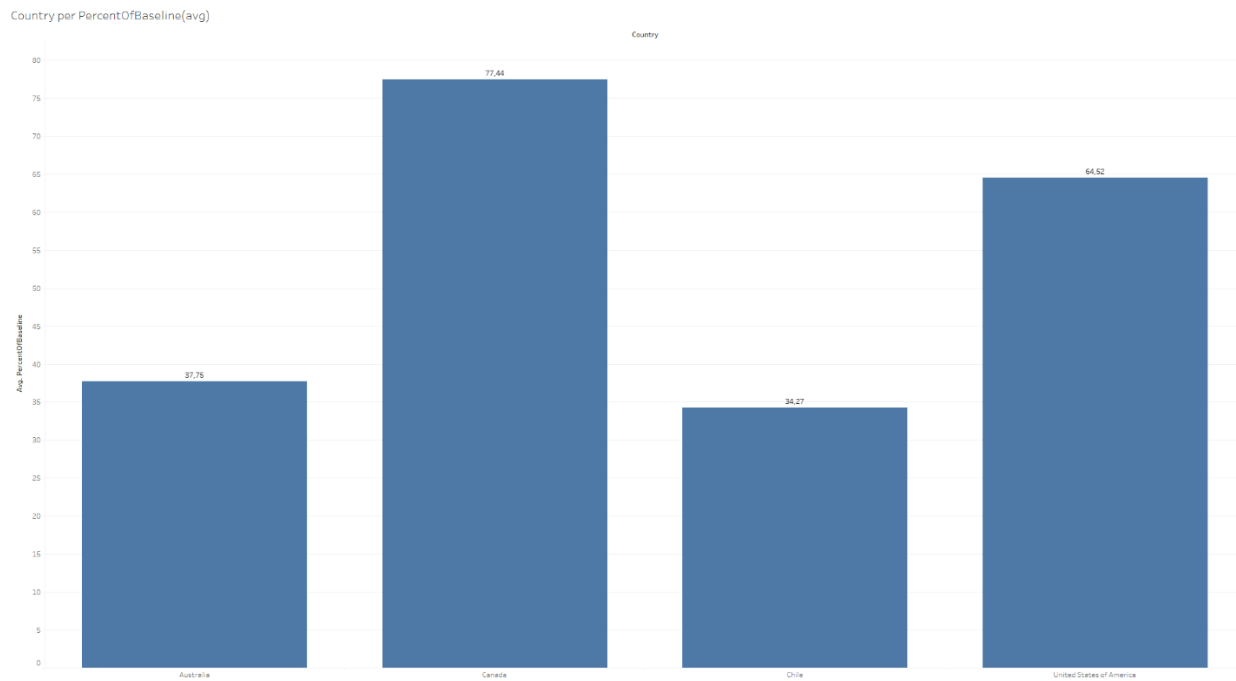
Grafiek 5

Month per AirportName filtered on the average of PercentOfBaseline

	Date											
Airport Name	March	April	May	June	July	August	September	October	November	December	Grand Total	
Hamilton International	89.80	78.90	87.26	94.57	90.39	89.10	92.70	92.81	97.40	100.00	90.43	
Winnipeg International	94.75	85.07	92.23	96.17	91.58	86.57	86.20	88.19	92.10	100.00	90.16	
Calgary International	90.63	83.67	83.74	88.60	87.26	91.07	92.33	92.10	96.30	99.00	89.51	
Newark Liberty Internatio..	79.38	63.60	66.97	83.07	96.81	100.00	97.70	98.97	95.63	100.00	87.41	
Montreal Mirabel	76.13	60.89	89.97	92.90	85.74	90.53	94.60	91.03	95.47	100.00	87.34	
Daniel K. Inouye Internati..	95.00	82.90	80.61	88.17	85.29	90.07	85.13	89.19	71.20	68.00	84.63	
Vancouver International	86.25	66.57	69.81	78.67	79.45	76.07	78.77	79.71	80.33	88.50	76.89	
Dallas/Fort Worth Interna..	70.94	57.97	57.00	71.37	75.29	78.03	85.67	90.61	89.63	100.00	75.58	
Charlotte Douglas Interna..	73.00	60.63	67.48	76.77	74.77	85.57	90.20	76.26	71.90	89.50	75.38	
Edmonton International	62.60	57.28	57.83	70.77	76.71	75.04	75.38	90.10	88.07	100.00	74.31	
Montreal Trudeau	80.69	55.37	61.19	71.30	76.26	78.03	78.73	80.81	81.77	94.50	73.59	
Chicago OHare Internatio..	77.81	65.40	69.13	75.73	71.87	74.80	71.70	76.19	76.23	90.50	73.08	
Los Angeles International	77.13	58.27	61.19	63.63	69.16	73.35	74.70	79.58	79.73	83.50	70.51	
Miami International	70.63	65.13	69.03	79.03	75.45	68.80	67.67	68.26	69.73	80.00	70.48	
Boston Logan International	80.69	59.90	56.71	61.77	65.35	65.93	66.40	67.52	65.40	69.00	64.70	
Seattle-Tacoma Internati..	69.63	53.83	58.61	66.63	67.65	66.23	64.47	66.97	63.47	78.00	63.98	
John F. Kennedy Internati..	57.56	47.73	52.71	60.27	60.94	64.00	66.73	81.74	75.07	65.50	63.31	
Hartsfield-Jackson Atlant..	78.75	57.13	55.10	58.23	62.61	65.13	62.90	66.10	59.50	60.00	61.95	
LaGuardia	68.19	54.27	51.71	58.33	61.39	67.00	63.47	65.35	60.87	73.50	60.87	
Halifax International	46.06	33.31	55.61	68.27	57.94	47.79	55.44	72.29	67.73	62.00	57.15	
Toronto Pearson	66.63	44.20	46.29	60.30	59.23	57.07	62.07	60.55	59.07	68.50	56.83	
Denver International	66.81	38.63	41.26	49.60	62.81	63.67	64.43	63.58	58.27	68.50	56.10	
Washington Dulles Intern..	65.31	48.77	48.03	52.47	51.55	58.93	60.13	63.26	58.80	67.00	55.94	
Detroit Metropolitan Way..	61.31	46.53	38.55	44.33	46.06	46.13	51.20	51.77	46.70	62.00	47.43	
San Francisco Internatio..	61.00	38.60	41.06	41.97	46.84	51.33	54.33	50.06	45.53	62.50	47.25	
McCarran International	39.94	32.67	26.39	30.53	36.03	40.93	47.80	45.03	44.37	58.00	38.22	
Kingsford Smith	83.13	53.24	49.13	52.52	46.26	17.79	16.40	19.39	23.83	24.00	37.75	
Santiago International Air..	38.40	19.57	21.44	26.85	31.11	33.27	37.21	47.81	52.54	46.50	34.27	
Grand Total	71.77	56.25	59.30	66.72	67.66	68.21	69.53	71.97	70.38	77.09	66.70	

Als laatste nog eens een staafdiagram met per land met het gemiddelde percentage t.o.v. voor Corona.

Grafiek 6



Conclusie die we kunnen trekken uit de bovenstaande grafieken:

De meeste data die we hebben bevindt zich in het Noorden van Amerika en in Canada. Hier zien we dat Corona over het algemeen niet zo'n grote impact heeft gehad ten opzichte van Chili en Australië. Maar omdat we in Chili en Australië slechts elk 1 luchthaven kunnen zien, kan het zijn dat deze data een vertekent beeld schetst. Het kan namelijk zijn dat andere luchthavens in Australië of Zuid-Amerika minder getroffen zijn en dit gewoon de uitersten zijn.