descriptive_statistics

May 28, 2022

1 Investigation of flight delays in San Francisco with focus on American Airlines.

[ ]:
```
# import packages
import os # file handling
import sqlite3 # database handling
import pandas as pd # data handling
import matplotlib.pyplot as plt # plotting
import seaborn as sns # plotting

import db_params # holds information about the database path

# set the background color of figures white
plt.rcParams.update({'figure.facecolor':'white'})
```

As the focus is on San Francisco, we can first have brief look on the airport situation there.

[ ]:
```
# we are interested in the flights from San Francisco
city_of_interest = 'San Francisco, CA'

# connect to the database
conn = sqlite3.connect(os.path.join(db_params.DB_PATH, db_params.DB_NAME))
crsr = conn.cursor()

# get airport ID and code
crsr.execute("SELECT AirportID FROM airports WHERE CityName=?",␣
 ↪(city_of_interest,))
sf_airport_IDs = crsr.fetchone()
crsr.execute("SELECT Airport FROM airports WHERE CityName=?",␣
 ↪(city_of_interest,))
sf_airport = crsr.fetchone()

# print amount of found airports, airport ID and code
if len(sf_airport_IDs) == 1:
    print(f"Found {len(sf_airport_IDs)} airport in {city_of_interest}.")
    print(f"It's code is '{sf_airport[0]}'.")
else:
```

```
    print(f"Found {len(sf_airport_IDs)} airports in {city_of_interest}.")
    print(f"Their codes are {sf_airport}.")


sf_airport_id = sf_airport_IDs[0]
sf_airport = sf_airport[0]
print(f"San Francisco Airport ID: {sf_airport_id}")
```

There is only one Airport and we can use the ID to identify it later on.

[ ]:
```
# define columns from the flight table of the database that could be interesting
cols =[
    "FlightDate",
    "Reporting_Airline",
    "DestAirportID",
    "ArrDelayMinutes",
    "Cancelled",
    "CancellationCode",
    "Diverted",
    "CarrierDelay",
    "WeatherDelay",
    "NASDelay",
    "SecurityDelay",
    "LateAircraftDelay",
]

# make string and separate the column names with commas
query_cols = ""
for col in cols:
    query_cols = query_cols + f"{col}, "
query_cols = query_cols[:-2]

# query tha database and get data that might be helpful to understand flight␣
 ↪delays
query = f"""
    SELECT {query_cols}, Month, DayOfWeek, DayOfWeek_str, DestAirport
    FROM (
        SELECT *
        FROM flights
        LEFT JOIN (
            SELECT FlightDate AS f_d, Month, DayOfWeek, Description AS␣
 ↪DayOfWeek_str
            FROM time_period
            LEFT JOIN L_WEEKDAYS
            ON time_period.DayOfWeek = L_WEEKDAYS.Code
        ) time_table
        ON flights.FlightDate = time_table.f_d
```

```
        WHERE OriginAirportID = {sf_airport_id}
    ) data
    LEFT JOIN (
        SELECT Code, Description AS Cancellation_Cause
        FROM L_CANCELLATION
    ) cancellation
    ON data.CancellationCode = cancellation.Code
    LEFT JOIN (
        SELECT AirportID, Airport AS DestAirport
        FROM airports
    ) airports_enc
    ON data.DestAirportID = airports_enc.AirportID
"""

# read the data into a dataframe for later use
df_sf_orig = pd.read_sql(
    query,
    conn
)
```

[ ]:
```
# show random rows to control the dataframe
df_sf_orig.sample(5)
```

2 Airlines wit departure at SFO

As we are interested in flight delays in departure, we can check how many and which airlines offer flights from SFO.

[ ]:
```
# get all airlines with departure in SF
query = f"""
      SELECT DISTINCT Reporting_Airline
      FROM flights
      WHERE OriginAirportID = {sf_airport_id}

"""
crsr.execute(query)
airlines = [airline[0] for airline in crsr.fetchall()]
print(f"There are {len(airlines)} departuring from {sf_airport}.")
print(f"Airlines: {airlines}")
```

3 Cause of delay

In this dataset, there are several causes of delay listed. Those are Carrier Delay, Weather Delay, National Air System (NAS) Delay, Security Delay and Late Aircraft Delay. To get an idea, how and if airlines can imporve their service and avoiding delays, we can have a look at those delay causes and their impact on total delays.

3.1 Amount of Delays per Cause

[ ]:
```
causes_of_delay_counts = {}

# define column names with causes of delay
causes_of_delay = ["CarrierDelay", "WeatherDelay", "NASDelay", "SecurityDelay",␣
 ↪"LateAircraftDelay"]

# query for each cause and find the amount of delays
for cause in causes_of_delay:
    # make query
    query = f"""
        SELECT COUNT({cause})
        FROM flights
        WHERE OriginAirportID = {sf_airport_id} AND NOT {cause}=0
    """
    # execute query
    crsr.execute(query)

    # get results of the query
    causes_of_delay_counts[cause], = crsr.fetchone()

# sort results by value
causes_of_delay_counts = dict(sorted(causes_of_delay_counts.items(), key=lambda␣
 ↪x:x[1], reverse=True))
causes_of_delay_counts
```

[ ]:
```
# define figure size
plt.figure(figsize=(10, 5))
# make a barplot from the dictionary causes_of_delay_counts using its keys and␣
 ↪values
g=sns.barplot(x=list(causes_of_delay_counts.keys()),␣
 ↪y=list(causes_of_delay_counts.values()), color=sns.color_palette()[0])

# set title and y-label
g.set_title("Amount of Delays per Delay Cause")
g.set_ylabel("Counts");
```

The bar chart shows that Security and Weather Delays don't happen often (<1000), whereas there were between 15000 and 20000 of each Late Aircraft Delay, NAS Delay and Carrier Delay.

3.2 Average delay time per cause

[ ]:
```
causes_of_delay_avg = {}

# query for each cause and find the avergage time delay
for cause in causes_of_delay:
```

```
    query = f"""
        SELECT SUM({cause}) / COUNT({cause})
        FROM flights
        WHERE OriginAirportID = {sf_airport_id} AND NOT {cause}=0
    """
    # execute and get results
    crsr.execute(query)
    causes_of_delay_avg[cause], = crsr.fetchone()

# sort results by value
causes_of_delay_avg = dict(sorted(causes_of_delay_avg.items(), key=lambda x:
 ↪x[1], reverse=True))
causes_of_delay_avg
```

[ ]:
```
# define figure size
plt.figure(figsize=(10, 5))
# make a barplot from the dictionary causes_of_delay_avg using its keys and␣
 ↪values
g=sns.barplot(x=list(causes_of_delay_avg.keys()), y=list(causes_of_delay_avg.
 ↪values()), color=sns.color_palette()[0])
# set title and y-label
g.set_title("Average Delay Time per Delay Cause")
g.set_ylabel("Time in minutes");
```

If there is a delay, then Weather Delays are the longest (on average about 100 minutes).

3.3 Average Delay of Carriers

[ ]:
```
# get the average delay per airline
query = f"""
    SELECT Reporting_Airline, AVG(ArrDelayMinutes)
    FROM flights
    WHERE OriginAirportID = {sf_airport_id}
    GROUP BY Reporting_Airline
    ORDER BY AVG(ArrDelayMinutes) DESC
"""
crsr.execute(query)
delay_all_avg_airlines_per_flight = crsr.fetchall()

delay_all_avg_airlines_per_flight
```

[ ]:
```
# define figure size
plt.figure(figsize=(10, 5))
# make a barplot from the results of the query
airl = [al[0] for al in delay_all_avg_airlines_per_flight]
g=sns.barplot(
    x=[airl[1] for airl in delay_all_avg_airlines_per_flight],
```

```
    y=airl,
    palette=[sns.color_palette()[0] if al!='AA' else sns.color_palette()[3] for␣
 ↪al in airl]
)
# set title, x- and y-label
g.set_title("Average Delay per Airline per Flight")
g.set_xlabel("Time in minutes")
g.set_ylabel("Airlines");
```

Comparing the average delay of carriers per flight shows, that there are recognizable differences. Focusing on American shows that there are three airlines with higher average delay and five with lower. The best airlines has 12 minutes delay on average, the worst 26 minutes. American Airline has 17 minutes average delay.

3.4 Delay Causes of American Airlines

[ ]:
```
causes_of_delay_avg_AA = {}

# query for each cause and find the avergage time delay
for cause in causes_of_delay:
    query = f"""
        SELECT SUM({cause}) / COUNT({cause}), SUM({cause}), COUNT({cause})
        FROM flights
        WHERE OriginAirportID = {sf_airport_id} AND NOT {cause}=0 AND␣
 ↪Reporting_Airline = 'AA'
    """
    # execute and get results
    crsr.execute(query)
    causes_of_delay_avg_AA[cause] = crsr.fetchone()

# sort results by value
causes_of_delay_avg_AA = dict(sorted(causes_of_delay_avg_AA.items(), key=lambda␣
 ↪x:x[1], reverse=True))
causes_of_delay_avg_AA
```

[ ]:
```
# define figure size
fig, ax = plt.subplots(1, 3, figsize=(10, 5))
for i in range(3):
# make a barplot from the dictionary causes_of_delay_avg using its keys and␣
 ↪values
    sns.barplot(
        x=list(causes_of_delay_avg_AA.keys()),
        y=[val[i] for val in list(causes_of_delay_avg_AA.values())],
        color=sns.color_palette()[0],
        ax=ax[i]
    )
    ax[i].set_xticklabels(
```

```
        ax[i].get_xticklabels(),
        rotation=90,
    )
# set title and y-label
plt.suptitle("Delay Causes of American Airlines Flights", fontsize=20)
ax[0].set_ylabel("Average Delay in minutes")
ax[1].set_ylabel("Sum of Delays in minutes")
ax[2].set_ylabel("Amount of Delays")
plt.tight_layout();
```

For American Airlines the occurence of delays is very close to what we observed before for all airlines: * Security and Weather Delays don't happen often. * Aircraft Delays are the occuring the most. * And Carrier and NAS Delays are a little bit less than Late Aircraft Delays.

The Delay durations however differe from the averages shown above: * Weather Delays were quite short. Since the airlines can't influence that, it's not that interesting. * The other Delays causes behave quite similar to the average. * Carrier Delays were longer than average. AA: 54 minutes, average all airlines: 46 minutes. Since this is the only cause that can be tackeled in San Francisco, we focus on **Carrier Delay**.

4 Carriers: carrier delay only

4.1 Average delay per carrier

Next, we will have a closer look at Carrier Delays and compare airlines.

[ ]:
```
# query for the average delay per carrier considering carrier delay only
query = f"""
    SELECT Reporting_Airline, SUM(CarrierDelay) / COUNT(*)
    FROM flights
    WHERE OriginAirportID = {sf_airport_id}
    GROUP BY Reporting_Airline
    ORDER BY SUM(CarrierDelay) / COUNT(*) DESC
"""
crsr.execute(query)
delay_avg_airlines_per_flight = crsr.fetchall()
```

[ ]:
```
# define figure size
plt.figure(figsize=(10, 5))
# make a barplot from the results of the query
airl = [al[0] for al in delay_all_avg_airlines_per_flight]
g=sns.barplot(
    x=[airl[1] for airl in delay_avg_airlines_per_flight],
    y=airl,
    palette=[sns.color_palette()[0] if al!='AA' else sns.color_palette()[3] for␣
 ↪al in airl]
)
# set title, x- and y-label
```

```
g.set_title("Average Delay per Airline per Flight (carrier delay only)")
g.set_xlabel("Time in minutes")
g.set_ylabel("Airlines");
```

The ranking is the same as when all delay causes were considered. * F9: worst with > 11 minutes carrier delay per flight * WN: best with ~ 2 minutes carrier delay per flight * AA: in the midfieled with 5.5 minutes average carrier delay.

4.2 Median delay per carrier

Next median and mode are calculated to get a better understanding of the distributions.

[ ]:
```
# calcualte the mean for those airlines
# derived from the answer of user 'CL.' at https://stackoverflow.com/questions/
 ↪15763965/how-can-i-calculate-the-median-of-values-in-sqlite
delay_median_airlines = {}

for airline in airlines:
      query = f"""
      SELECT AVG(CarrierDelay)
      FROM (
            SELECT CarrierDelay
            FROM flights WHERE OriginAirportID = {sf_airport_id} AND␣
 ↪Reporting_Airline='{airline}'
            ORDER BY CarrierDelay
            LIMIT 2 - (
                  SELECT COUNT(*)
                  FROM flights
                  WHERE OriginAirportID = {sf_airport_id} AND␣
 ↪Reporting_Airline='{airline}'
            ) % 2    -- get 1 value, if odd amount, else 2
            OFFSET (
                  SELECT (COUNT(*) - 1) / 2
                  FROM flights
                  WHERE OriginAirportID = {sf_airport_id} AND␣
 ↪Reporting_Airline='{airline}'
            ) -- start in the middle of the data
      )
      """
      crsr.execute(query)
      delay_median_airlines[airline], = crsr.fetchone()
```

The median Carrier Delay for all airlines is None, respectively 0. That means that at least half of the flights of the airlines had no Carrier Delay.

4.3 Mode of delay per carrier

```
[ ]: delay_mode_airlines = {}

# calculate the mode of delay per airline
for airline in airlines:
    query = f"""
        SELECT CarrierDelay
        FROM flights
        WHERE OriginAirportID = {sf_airport_id} AND␣
 ↪Reporting_Airline='{airline}'
        GROUP BY CarrierDelay

    """
    crsr.execute(query)
    delay_mode_airlines[airline], = crsr.fetchone()
```

The mode of Carrier Delay for all airlines is None as well, respectively 0. That means that of all Carrier Delay occurences no delay was the most common. Combining this knowledge, one can say that the distributions of the Carrier Delays of all airlines look similar. The have a sharp peek at 0 minutes, are right skewed as mean > mode.

4.4 Carriers: cancelled flights

In addition to delays, cancelled flights are also intersting because both being on time and arriving at all are important to customers.

```
[ ]: # get the percentage of cancelled flights per airline
query = f"""
    SELECT Reporting_Airline, SUM(Cancelled) / COUNT(*) * 100 AS percentage
    FROM flights
    WHERE OriginAirportID = {sf_airport_id}
    GROUP BY Reporting_Airline
    ORDER BY percentage DESC
"""
crsr.execute(query)
cancelled_airlines = crsr.fetchall()
```

```
[ ]: # define figure size
plt.figure(figsize=(10, 5))
# make a barplot from the results of the query
airl = [airl[0] for airl in cancelled_airlines]
g=sns.barplot(
    x=[airl[1] for airl in cancelled_airlines],
    y=airl,
    palette=[sns.color_palette()[0] if al!='AA' else sns.color_palette()[3] for␣
 ↪al in airl]
)
```

```
# set title, x- and y-label
g.set_title("Percentage of cancelled flights per airline")
g.set_xlabel("Cancelled flights in %")
g.set_ylabel("Airlines");
```

Here again, two airlines perform worse and five better than American Airlines. * Three airlines managed to have less than 1 % cancelled flights. * 2.4 % of American Airlines flights were cancelled.

That shows that here as well preformance can be improved.

4.5 Carriers: Cancellation causes

Since there are cancellation causes airlines can't influence, a closer look can be helpful to evaluate the performance.

[ ]:
```
# cancellation_causes = {}
cancellation_causes = pd.DataFrame()
# get the percantage of cancelations per cause

for airline in airlines:
    query = f"""
        SELECT Reporting_Airline, Description, SUM(Cancelled) / (
            SELECT COUNT(*)
            FROM flights
            WHERE OriginAirportID = {sf_airport_id} AND Reporting_Airline =␣
 ↪'{airline}'
        ) * 100 AS percentage
        FROM flights
        LEFT JOIN L_CANCELLATION
        ON flights.CancellationCode = L_CANCELLATION.Code
        WHERE OriginAirportID = {sf_airport_id} AND Reporting_Airline =␣
 ↪'{airline}'
        GROUP BY CancellationCode
        ORDER BY percentage DESC
    """

    cancellation_causes= pd.concat([cancellation_causes, pd.read_sql(query,␣
 ↪conn)])
```

[ ]:
```
# define figure size
plt.figure(figsize=(10, 10))
# make a barplot from the results of the query
airl = [airl[0] for airl in cancelled_airlines]
g=sns.barplot(
    data=cancellation_causes,
    x='percentage',
    y='Reporting_Airline',
    hue='Description',
```

```
    # palette=[sns.color_palette()[0] if al!='AA' else sns.color_palette()[3]␣
  ↪for al in airl]
)
# set title, x- and y-label
g.set_title("Percentage of cancelled flights per airline and cause")
g.set_xlabel("Cancelled flights in %")
g.set_ylabel("Airlines")
g.legend(title="Cause of Cancellation");
```

As this figure shows the airlines WN and OO had bad luck. Weather respectively National Air System caused a large share of their cancelled flights. Focusing on American Airlines and carrier cancellations, one can say that AA should be able to reduce their amount of cancelled flights since three airlines were close to 0 % cancelled flights caused by the carrier.

5 American Airlines

Comming back to flight delays, to get an idea, how the performance of AA can be improved, patterns in delayed flights can help to find mechanisms to improve.

5.1 Weekdays

There could be repeating special events that impact daily business and increase the probability of delays.

[ ]:
```
# make a dataframe with American Airlines flights only
df_AA = df_sf_orig.loc[(df_sf_orig["Reporting_Airline"]=="AA")]
# add a counter that helpful for grouping
df_AA["counter"] = 1
# add a counter that counts carrier delays after grouping
df_AA["delay_counter"] = (df_AA["CarrierDelay"]!=0) & (~df_AA["CarrierDelay"].
  ↪isna())
```

[ ]:
```
# group data by day of week
df_AA_grouped_DayOfWeek = df_AA.groupby("DayOfWeek").sum().reset_index()
```

[ ]:
```
# get weekday encoding from the database
query = f"""
    SELECT Code, Description AS Weekday
    FROM L_WEEKDAYS
"""
crsr.execute(query)
weekdays_encoding = crsr.fetchall()
```

[ ]:
```
def make_double_delay_plot(df: pd.DataFrame, suptitle: str, x_col: str,␣
  ↪xticklabels_rot: int = 0, x_encoding: list = None):
    """Generates a two part plot with delayed flights and average delay per␣
  ↪flight.
```

```
    Args:
        df (pd.DataFrame): Dataframe that holds the data to plot.
        suptitle (str): Title of the plot.
        x_col (str): Data for the x-axis.
        xticklabels_rot (int, optional): Degree of rotation of x-tick-labels.␣
↪Defaults to 0.
        x_encoding (list, optional): Encoding of x tick values. If None default␣
↪will be used. Defaults to None.
    """
    # make a figure with tow subplots
    fig, ax = plt.subplots(1, 2, figsize=(10, 5))

    # use two barplots for those
    sns.barplot(
        x=df[x_col],
        y=df["delay_counter"] / df["counter"] * 100,
        color=sns.color_palette()[0],
        ax=ax[0]
    )
    sns.barplot(
        x=df[x_col],
        y=df["CarrierDelay"] / df["counter"],
        color=sns.color_palette()[0],
        ax=ax[1]
    )

    # set x-labels and x-tick-labels for both plots
    for i in range(2):
        ax[i].set_xlabel("")
        # encode x tick values if necessary
        if x_encoding:
            ax[i].set_xticklabels(
                [encoded[1] for encoded in x_encoding][:len(ax[i].
↪get_xticklabels())],
            )
        # rotate x tick labels if necessary
        if xticklabels_rot:
            ax[i].set_xticklabels(
                ax[i].get_xticklabels(),
                rotation=xticklabels_rot,
                ha="right",
            )

    # set y-labels individually
    ax[0].set_ylabel("Delayed Flights in %")
    ax[1].set_ylabel("Average Delay in minutes")
```

```
    # set supertitle for the plot
    plt.suptitle(suptitle, fontsize = 20)

    # show figure
    plt.show()
```

[ ]:
```
# make a figure showing delayed flights and average delay per weekday
make_double_delay_plot(df_AA_grouped_DayOfWeek, "Carrier Flight Delays of␣
 ↪American Airlines per Weekday", "DayOfWeek", xticklabels_rot=60,␣
 ↪x_encoding=weekdays_encoding)
```

- There are only slight differences in delay flights per weekday. The probability of delayed flights is slightly higher on Mondays, Thursdays and Fridays.
- The average delay is around seven minutes on Mondays and Fridays, but only five minutes on the other days.
- Combining those information, one can say that on Mondays and Fridays the probability of flights being delayed is increased as well as the the delay length. Obviously those are the days before and after weekend. Therefore, there could be a relationship, e.g.
  - There could be more holiday travellers and less business travellers on those days. Business travellers know the procedures very well. Therefore, there could be less delay caused by the customers themselves on weekdays.
  - There could be more customers right before and after weekend. Common trips could be from Monday to Friday, Friday to Sunday/Monday, because that often fits in well for people who are working, either for business trips as well as for private travels.

5.2 Month

[ ]:
```
# group data by month
df_AA_grouped_month = df_AA.groupby("Month").sum().reset_index()
```

[ ]:
```
# get month encoding from the database
query = f"""
    SELECT Code, Description AS Month
    FROM L_Months
"""
crsr.execute(query)
months_encoding = crsr.fetchall()
```

[ ]:
```
# make a figure showing delayed flights and average delay per month
make_double_delay_plot(df_AA_grouped_month, "Carrier Flight Delays of American␣
 ↪Airlines per Month", "Month", 60, months_encoding)
```

The share of delayed flights seems not to show tendencies over the year. However, September and December show an increased amount of flight delays. However, the average delay increased over the year, with a noticeable peek in September. To really say, if there are any tendencies and by what they are caused, deeper insights into the processes are necessary, e.g. staff, customers, amount of travelling people, and more details about the process as a whole.

5.3 Destination Airport

Lastly, we can have a look at the destination airports and see if there any patterns.

[ ]:
```
# get airport encoding from database
query = f"""
    SELECT AirportID, Airport AS DestAirport
    FROM airports
"""

df_airport_encoding = pd.read_sql(
    query,
    conn
)
```

[ ]:
```
# group data by Destination Airport
df_AA_grouped_dest = df_AA.groupby("DestAirportID").sum().reset_index()
df_AA_grouped_dest = df_AA_grouped_dest.merge(df_airport_encoding,␣
 ↪left_on="DestAirportID", right_on="AirportID")
```

[ ]:
```
# make a figure showing delayed flights and average delay per destination␣
 ↪airport
make_double_delay_plot(df_AA_grouped_dest, "Flight Delays of American Airlines␣
 ↪per Destination", "DestAirport", xticklabels_rot=60)
```

There are only slight differences between the destination airports. However, JFK shows the least share of delayed flights , but the longetst delays.

[ ]:
```
# Close cursor and connection
crsr.close()
conn.close()
```

6 Conclusion

In conclusion, it seems that American Airlines can improve their performance regarding delays as well as cancellations. Some of their competitors have higher rates of delays and longer delays on average. However, there are also competitors with less and shorter delays at the same time as well less cancellations. A deeper understanding of the relevant processes and analysis of the competitors is necessary to find ways to improve those points. Additionally, there are first hints where to look for improvements. The rate of delays is increased on Mondays and Fridays. Therefore, an analysis is necessary to find differences of those days compared to others. Furthermore, the rate of delayed flights and average delay length is increased in September and also relatively high in December. Of the destination airports JFK airport has the lowest rate of delays, but the highest average delay length. Further inverstigations on this could also reveal ideas how to improve performance. As mentioned before, further investigations are necessary to find mechanisms to improve the delay and cancellation performance of American Airlines. Data of the customers, aircrafts, staff could help in this case. Additionally, processes of competitors could be analyzed as far as possible, especially of those with low delay rates and shotz delay durations.

7 Export Report

https://nbconvert.readthedocs.io/en/latest/install.html

[ ]:
```
!jupyter nbconvert descriptive_statistics.ipynb --to pdf
```

[ ]:
```

```