

Analiza danych dotyczących transportu kolejowego we Francji

Autor: Mateusz Sliwka

Wstęp

Zbiór danych pochodzi z serwisu Kaggle.com(<https://www.kaggle.com/gatandubuch/public-transport-traffic-data-in-france>). Liczba kolumn: 32 Liczba wierszy: 7806

```
In [1]: # użyte biblioteki

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: data_set = pd.read_csv('Regularities_by_Liaisons_Trains_France.csv')
data_set.head()
```

Out[2]:

	Year	Month	Departure station	Arrival station	Average travel time (min)	Number of expected circulations	Number of cancelled circulations	Number of late trains at departure	Average delay of late departing trains (min)	Average delay of all departing trains (min)	...	Average train delay > 15min	Number of late trains > 15min	Number of late trains > 30min	Per
0	2019	7.0	ANGOULEME	PARIS MONTPARNAASSE	131.914980	247.0	0.0	181.0	3.576353	2.678273	...	32.965873	7.0	2.0	20
1	2019	7.0	PARIS MONTPARNAASSE	LA ROCHELLE VILLE	175.611570	242.0	0.0	178.0	9.780805	7.033609	...	32.057143	14.0	2.0	20
2	2019	7.0	LE MANS	PARIS MONTPARNAASSE	62.395349	435.0	5.0	391.0	3.896974	3.529341	...	42.367241	13.0	4.0	20
3	2019	7.0	ST MALO	PARIS MONTPARNAASSE	172.421053	114.0	0.0	101.0	1.950990	1.685673	...	27.620833	2.0	0.0	20
4	2019	7.0	PARIS MONTPARNAASSE	ST PIERRE DES CORPS	67.310000	404.0	4.0	284.0	8.379108	5.803125	...	37.658333	12.0	3.0	20

5 rows × 16 columns

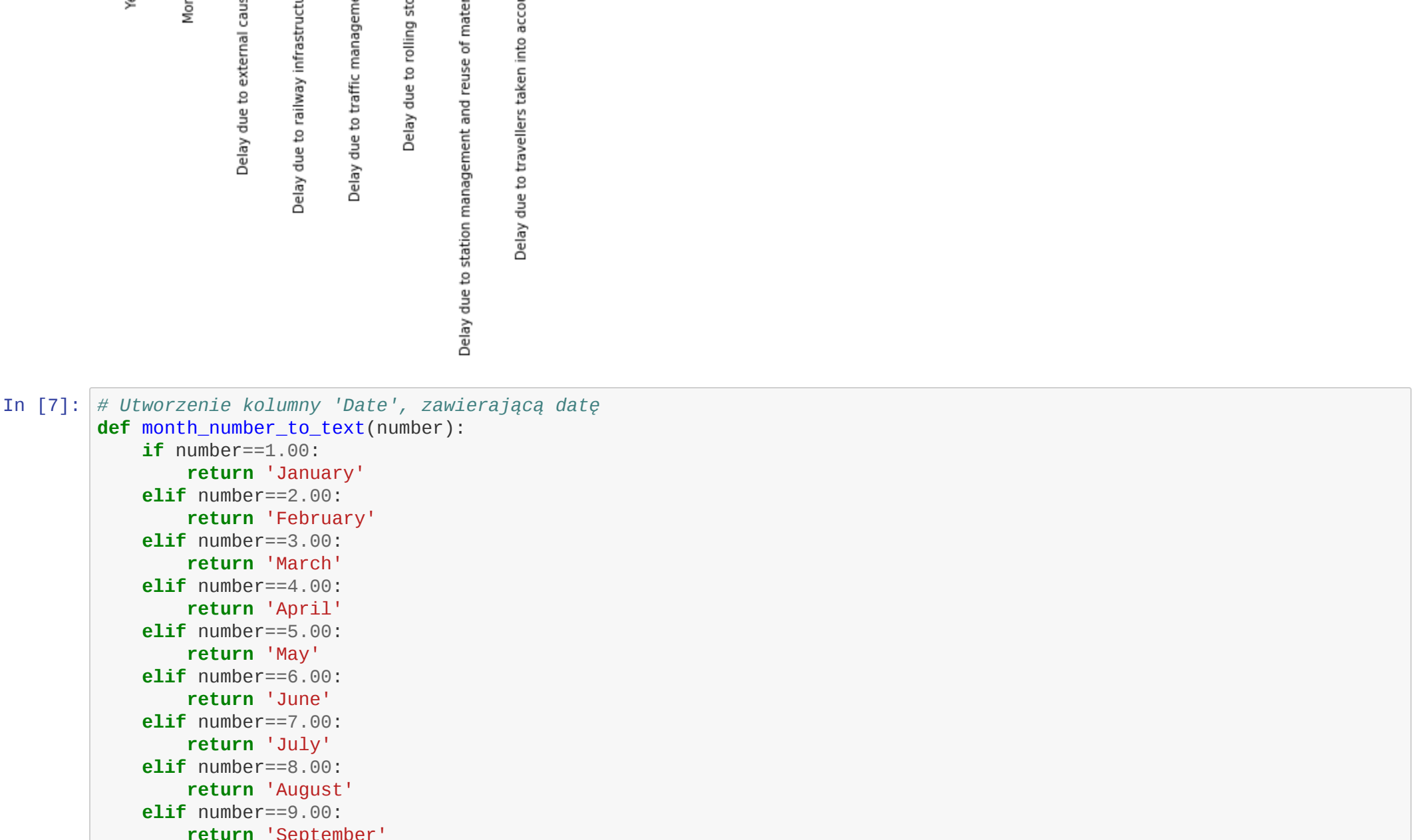
Do ostatecznej analizy zostały użyte kolumny dotyczą roku oraz miesiąca, w którym wystąpiły opóźnienia oraz liczba opóźnień w podziale na rodzaj przyczyny, która spowodowała opóźnienie.

- Delay due to external causes' - opóźnienia z przyczyn zewnętrznych (pogoda, przeszkody, podejrzanе pakunki, złośliwe zamiany, ruchy społeczne itp.)
- Delay due to railway infrastructure' - spóźnienia z powodu infrastruktury kolejowej (prace konserwacyjne)
- Delay due to traffic management' - opóźnienia pociągów spowodowanych zarządzaniem ruchem (ruch na linii kolejowej, interakcje w sieci)
- Delay due to rolling stock' - spóźnienia ze względu na tabor kolejowy
- Delay due to station management and reuse of material' - opóźnienia pociągów spowodowanych zarządzaniem stacją i ponownym wykorzystaniem materiałów
- Delay due to travellers taken into account' - opóźnienia spowodowane ruchem pasażerskim

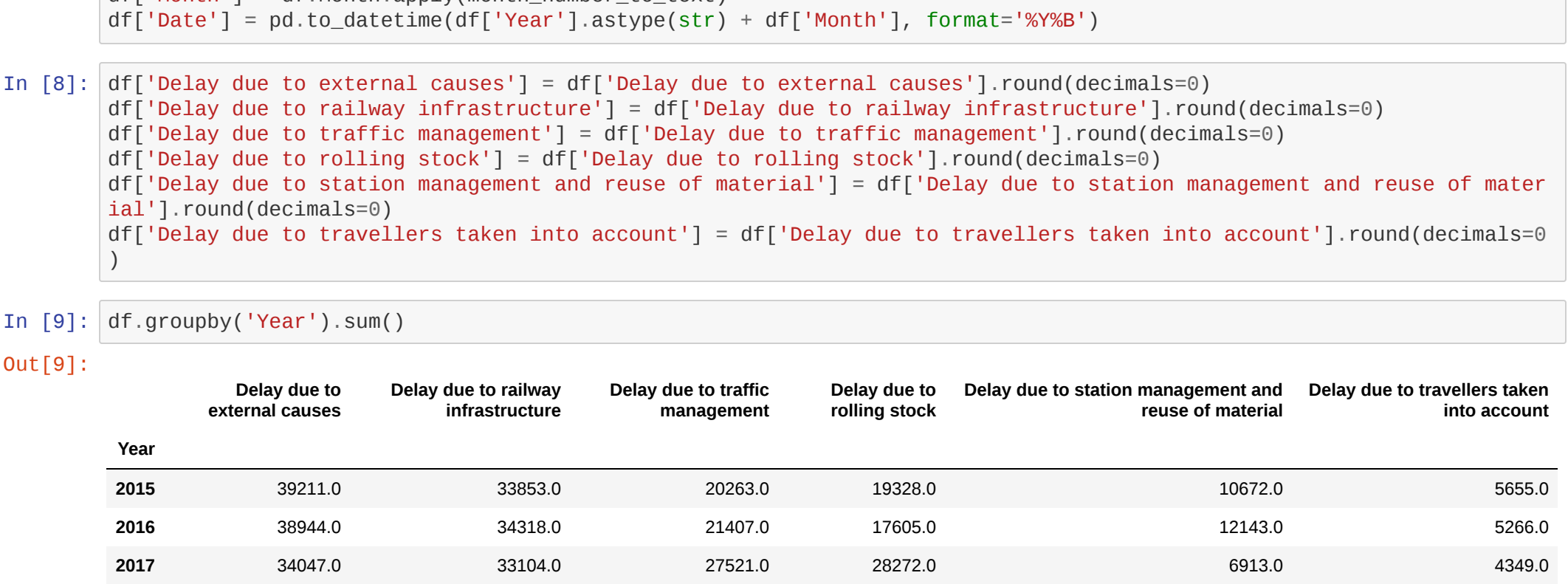
```
In [4]: df = data_set[['Year', 'Month', 'Delay due to external causes',
                    'Delay due to railway infrastructure',
                    'Delay due to traffic management',
                    'Delay due to rolling stock',
                    'Delay due to station management and reuse of material',
                    'Delay due to travellers taken into account']]
df.head()
```



```
In [5]: # wizualizacja występowania wartości pustych
sns.heatmap(df.isnull(), yticklabels = False, cbar = False, cmap = 'viridis')
```



```
In [6]: df.dropna(inplace=True)
sns.heatmap(df.isnull(), yticklabels = False, cbar = False, cmap = 'viridis')
```



```
In [7]: # Utworzenie kolumny 'Date', zawierająca datę
def month_number_to_text(number):
    if number==1.00:
        return 'January'
    elif number==2.00:
        return 'February'
    elif number==3.00:
        return 'March'
    elif number==4.00:
        return 'April'
    elif number==5.00:
        return 'May'
    elif number==6.00:
        return 'June'
    elif number==7.00:
        return 'July'
    elif number==8.00:
        return 'August'
    elif number==9.00:
        return 'September'
    elif number==10.00:
        return 'October'
    elif number==11.00:
        return 'November'
    elif number==12.00:
        return 'December'
    else:
        return 'Error'
```

```
df['Month'] = df.Month.apply(month_number_to_text)
df['Date'] = pd.to_datetime(df['Year'].astype(str) + df['Month'], format='%Y%b')
```

```
In [8]: df['Delay due to external causes'] = df['Delay due to external causes'].round(decimals=0)
df['Delay due to railway infrastructure'] = df['Delay due to railway infrastructure'].round(decimals=0)
df['Delay due to traffic management'] = df['Delay due to traffic management'].round(decimals=0)
df['Delay due to rolling stock'] = df['Delay due to rolling stock'].round(decimals=0)
df['Delay due to station management and reuse of material'] = df['Delay due to station management and reuse of material'].round(decimals=0)
df['Delay due to travellers taken into account'] = df['Delay due to travellers taken into account'].round(decimals=0)
```

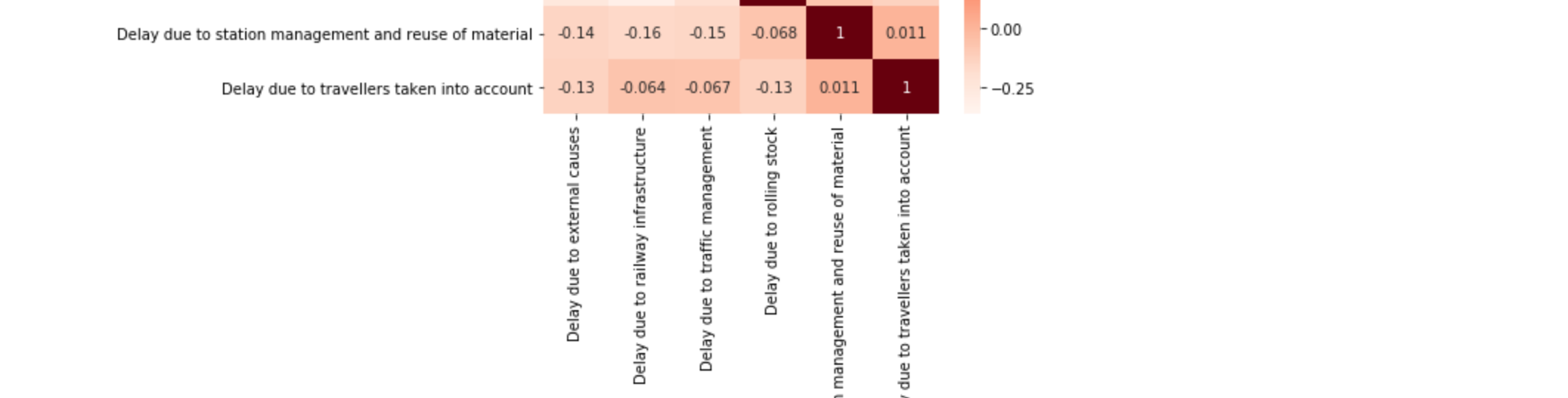
```
In [9]: df.groupby('Year').sum()
```

Out[9]:

	Delay due to external causes	Delay due to railway infrastructure	Delay due to traffic management	Delay due to rolling stock	Delay due to station management and reuse of material	Delay due to travellers taken into account
Year						
2015	39211.0	33863.0	20263.0	19328.0	10672.0	5655.0
2016	38944.0	34318.0	21407.0	17605.0	12143.0	5266.0
2017	34047.0	33104.0	27521.0	28272.0	6913.0	4349.0
2018	38330.0	34173.0	30945.0	32276.0	8048.0	4828.0
2019	37146.0	33747.0	33804.0	27885.0	9351.0	4779.0
2020	13537.0	22966.0	10635.0	11036.0	3930.0	2570.0

```
In [10]: # Zdefiniowanie kolejności występowania miesięcy
months = ['January', 'February', 'March', 'April', 'May', 'June',
          'July', 'August', 'September', 'October', 'November', 'December']
df['Month'] = pd.Categorical(df['Month'], categories=months, ordered=True)
```

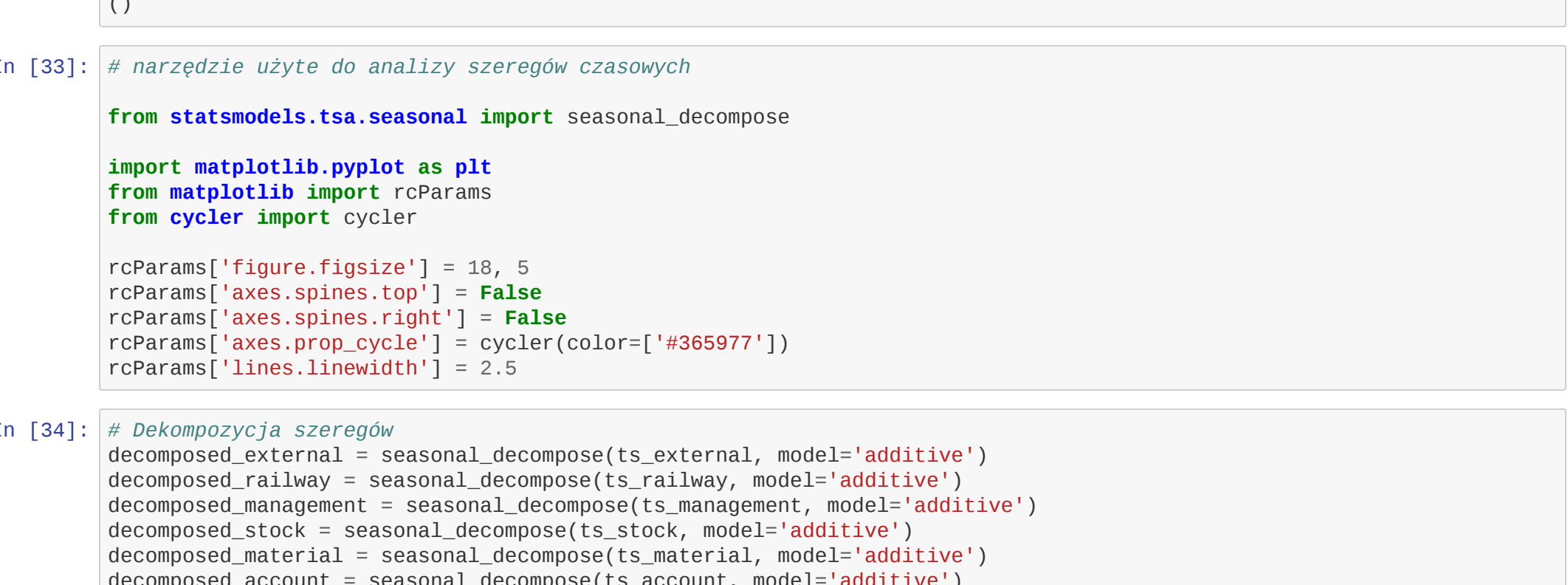
```
In [27]: df.Year.value_counts().sort_index().plot.bar()
```



```
In [26]: df.Month.value_counts().sort_index().plot()
plt.show()
```



```
In [28]: sns.heatmap(df.drop('Year',axis=1).corr(), annot=True, cmap = 'Reds')
```



Time series analysis

```
In [30]: df['Date'].min()
```

```
Out[30]: Timestamp('2015-01-01 00:00:00')
```

```
In [31]: df['Date'].max()
```

```
Out[31]: Timestamp('2020-06-01 00:00:00')
```

```
In [32]: # Utworzenie datafraw na potrzeby stworzenia szeregów czasowych
ts_external = df[['Delay due to external causes', 'Date']].set_index('Date').groupby('Date').mean().copy()
ts_railway = df[['Delay due to railway infrastructure', 'Date']].set_index('Date').groupby('Date').mean().copy()
ts_management = df[['Delay due to traffic management', 'Date']].set_index('Date').groupby('Date').mean().copy()
ts_stock = df[['Delay due to rolling stock', 'Date']].set_index('Date').groupby('Date').mean().copy()
ts_material = df[['Delay due to station management and reuse of material', 'Date']].set_index('Date').groupby('Date').mean().copy()
ts_account = df[['Delay due to travellers taken into account', 'Date']].set_index('Date').groupby('Date').mean().copy()
```

```
In [33]: # narzędzie użyte do analizy szeregów czasowych
from statsmodels.tsa.seasonal import seasonal_decompose

import matplotlib.pyplot as plt
from matplotlib import rcParams
from cycler import cycler

rcParams['figure.figsize'] = 18, 5
rcParams['axes.spines.top'] = False
rcParams['axes.spines.right'] = False
rcParams['axes.prop_cycle'] = cycler(color=['#365977'])
rcParams['lines.linewidth'] = 2.5
```

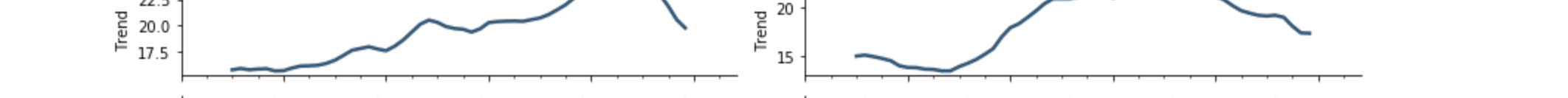
```
In [34]: # Dekompozycja szeregów
decomposed_external = seasonal_decompose(ts_external, model='additive')
decomposed_railway = seasonal_decompose(ts_railway, model='additive')
decomposed_management = seasonal_decompose(ts_management, model='additive')
decomposed_stock = seasonal_decompose(ts_stock, model='additive')
decomposed_material = seasonal_decompose(ts_material, model='additive')
decomposed_account = seasonal_decompose(ts_account, model='additive')
```

```
In [35]: # wyświetlenie wyników dekompozycji obok siebie
import matplotlib.pyplot as plt
import statsmodels.api as sm

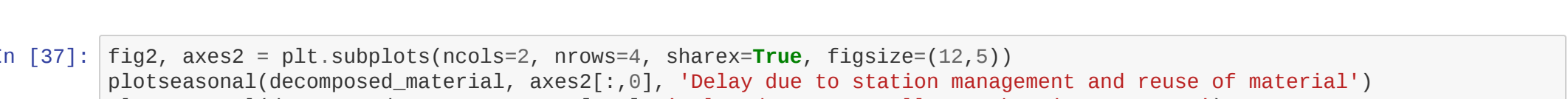
def plotseasonal(res, axes, title):
    res.observed.plot(ax=axes[0], legend=False)
    axes[0].set_title(title, fontweight='bold')
    axes[0].set_ylabel('Observed')
    res.trend.plot(ax=axes[1], legend=False)
    axes[1].set_ylabel('Trend')
    res.seasonal.plot(ax=axes[2], legend=False)
    axes[2].set_ylabel('Seasonal')
    res.resid.plot(ax=axes[3], legend=False)
    axes[3].set_ylabel('Residual')
```

```
fig, axes = plt.subplots(ncols=2, nrows=4, sharex=True, figsize=(12,5))
plotseasonal(decomposed_external, axes[:,0], 'Delay due to external causes')
plotseasonal(decomposed_railway, axes[:,1], 'Delay due to railway infrastructure')
```

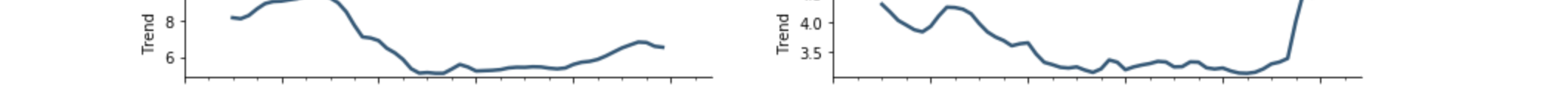
```
plt.tight_layout()
plt.show()
```



```
In [36]: fig1, axes1 = plt.subplots(ncols=2, nrows=4, sharex=True, figsize=(12,5))
plotseasonal(decomposed_management, axes1[:,0], 'Delay due to station management')
plotseasonal(decomposed_stock, axes1[:,1], 'Delay due to rolling stock')
plt.tight_layout()
plt.show()
```



```
In [37]: fig2, axes2 = plt.subplots(ncols=2, nrows=4, sharex=True, figsize=(12,5))
plotseasonal(decomposed_material, axes2[:,0], 'Delay due to station management and reuse of material')
plotseasonal(decomposed_account, axes2[:,1], 'Delay due to travellers taken into account')
plt.tight_layout()
plt.show()
```



Wnioski

- W 2020 nastąpił spadek zgłaszanych opóźnień, ale należy mieć na względzie fakt, iż dane dla roku 2020 zawierają dane tylko do połowy roku. Pozostały spadek zgłaszanych opóźnień w tym roku mógł zostać spowodowany spadkiem ogólnej liczby kursów pociągów z powodu pandemii COVID-19.
- Liczba zgłaszanych opóźnień w pierwszej połowie roku jest większa niż w drugiej połowie roku.
- Istnieje słaba korelacja pomiędzy spadkiem spowodowanych pracami konserwacyjnymi a wzrostem liczby spóźnień pociągów spowodowanych warunkami atmosferycznymi. W okresach zimowych wykonuje się mniej prac remontowych, a śnieg często powoduje opóźnienia na kolei.
- Wzrost opóźnień spowodowanych warunkami atmosferycznymi wykazuje sezonowość, aczkolwiek liczba takich opóźnień małe rok do roku.
- Liczba opóźnień spowodowanych pracami konserwacyjnymi mały w okresach zimowych.
- Liczba opóźnień spowodowanych zarządzaniem ruchem występuje sezonowo i rośnie rok do roku, co może być spowodowane wzrostem liczby kursów.
- Spóźnienia wynikające z powodu uszkodzenia taboru wykazują charakterystyczny stałość występowania.
- Opóźnienia pociągów spowodowanych zarządzaniem stacją i ponownym wykorzystaniem materiałów występują sezonowo.
- Opóźnienia spowodowane ruchem pasażerskim występują seziwno, głównie w okresie wakacyjnym i świątecznym.

```
In [ ]:
```