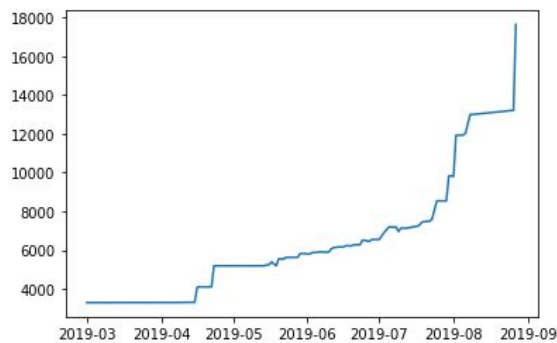


Wizualizacja danych - odpowiedzi

1.

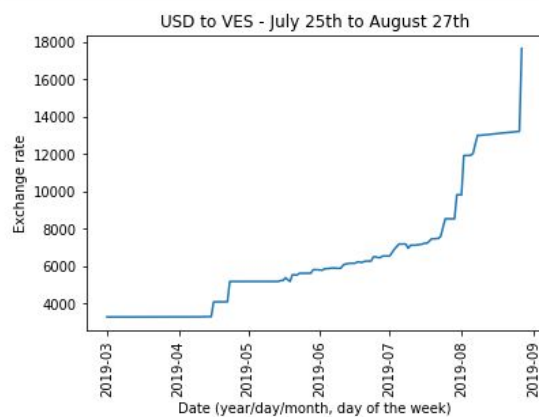
```
ves_usd = pd.read_csv('ml-throwdown-datasets/ves-usd.csv', thousands=',', parse_dates=True, index_col='Date')

fig, ax = plt.subplots()
ax.plot(ves_usd.index, ves_usd['Rate'])
plt.show()
```



2.

```
fig, ax = plt.subplots()
ax.plot(ves_usd.index, ves_usd['Rate'])
ax.set_xlabel('Date (year/day/month, day of the week)')
# etykieta osi Y
ax.set_ylabel('Exchange rate')
# tytuł wykresu
ax.set_title('USD to VES - July 25th to August 27th')
# obrót etykiet na osi odciętych
ax.xaxis.set_tick_params(rotation=90)
plt.show()
```



3.

```
medals = pd.read_csv('ml-throwdown-datasets/olympic.csv', index_col='Year')

def analyze_medals(medal_df):
    countries = groupby_column(medal_df, 'NOC')
    ret = {}
    for country, df in countries.items():
        counts = df["Medal"].value_counts()
        gold = counts['Gold'] if 'Gold' in counts else 0
        silver = counts['Silver'] if 'Silver' in counts else 0
        bronze = counts['Bronze'] if 'Bronze' in counts else 0
        ret[country] = (gold, silver, bronze)
    return ret

analyze_medals(medals)
```

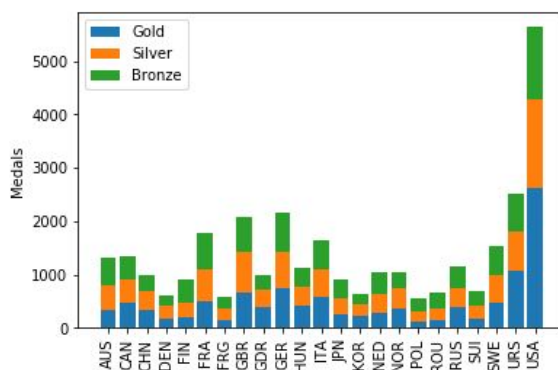
```
c_df = pd.DataFrame.from_dict(analyze_medals(medals),
                              orient='index',
                              columns=['Gold', 'Silver', 'Bronze'])

new_medals = c_df[c_df.sum(axis=1) > 500]
new_medals
```

```
# stacked barchart
fig, ax = plt.subplots()
ax.bar(new_medals.index, new_medals["Gold"], label="Gold")
ax.bar(new_medals.index, new_medals["Silver"],
       bottom=new_medals["Gold"], label="Silver")
ax.bar(new_medals.index, new_medals["Bronze"],
       bottom=new_medals["Gold"]+new_medals["Silver"], label="Bronze")
ax.xaxis.set_tick_params(rotation=90)
ax.set_ylabel('Medals')

ax.legend()

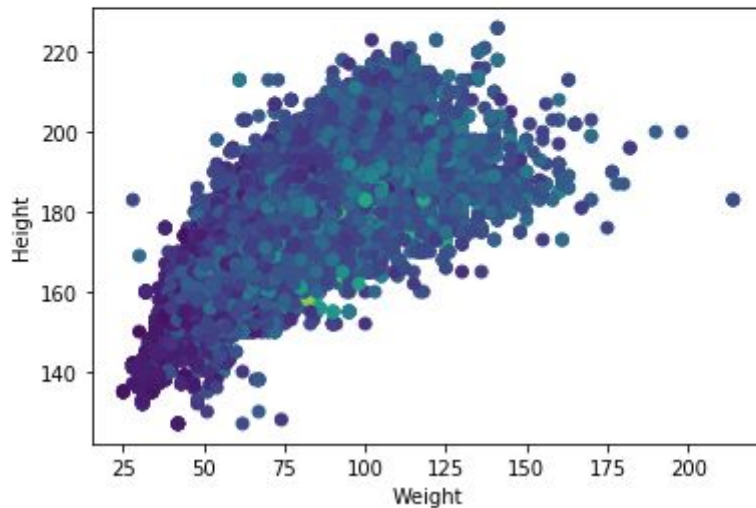
plt.show()
```



4.

```
fig, ax = plt.subplots()
ax.scatter(medals["Weight"], medals["Height"],
           c=medals["Age"])
ax.set_xlabel('Weight')
ax.set_ylabel('Height')

plt.show()
```

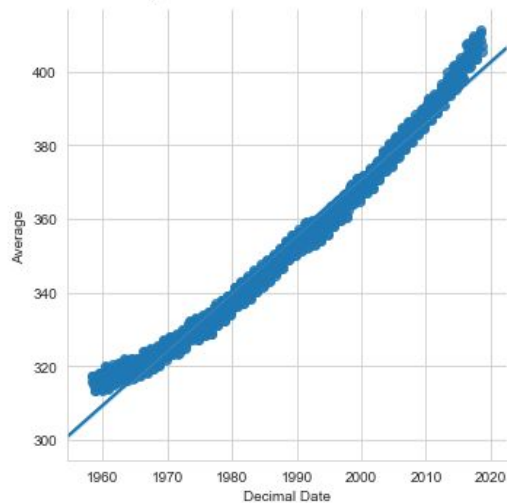


5.

```
co2 = pd.read_csv('ml-throwdown-datasets/co2.csv', na_values=-99.99,
                  parse_dates=True, index_col='Date')

sns.lmplot(x='Decimal Date', y='Average', data=co2)
```

<seaborn.axisgrid.FacetGrid at 0x1a25860fd0>



6.

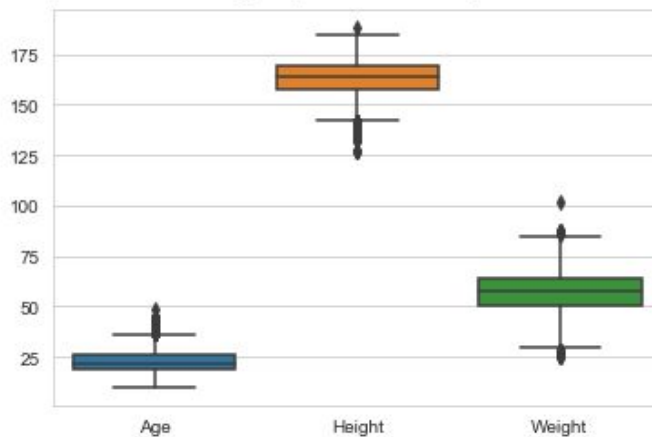
```
medals = pd.read_csv('ml-throwdown-datasets/olympic.csv', index_col=['Year', 'ID'])

def groupby_column(dataframe, column):
    groupby = dataframe.groupby(column)
    return {x: groupby.get_group(x) for x in groupby.groups}

all_sports = groupby_column(medals, 'Sport')
```

```
sns.boxplot(data=all_sports['Gymnastics'])
```

<matplotlib.axes._subplots.AxesSubplot at 0x1a294dfba8>



```
sns.violinplot(data=all_sports['Basketball'])
```

<matplotlib.axes._subplots.AxesSubplot at 0x1a2ffc6320>

