In [ ]:

Laboratorium 3 - Wizualizacja danych przy użyciu MatPlotLib i innych bibliotek

Bazowo zacznijmy od przygotowania danych

In [7]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("netflix_titles.csv")  # <-- plik musi być w tym samym folderze
print(df.columns)
df.head()
```

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

Out[7]:

	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021

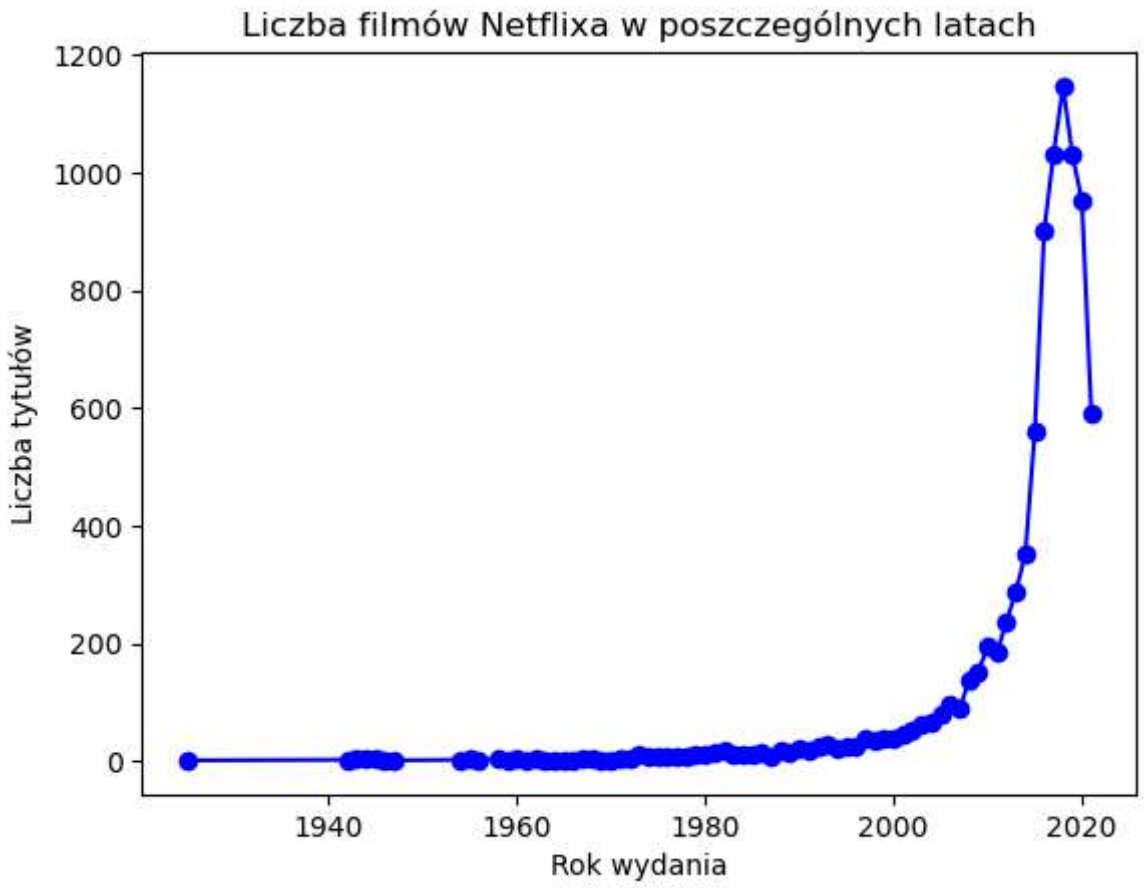
In [8]:
```
# import matplotlib.pyplot as plt
# import numpy as np
```

In [9]:
```
 # x = [1, 2, 3, 4, 5]
 # y = [2, 3, 5, 7, 11]
```

3.1 Rodzaje wykresów podstawowych

3.1.1 Wykres Liniowy

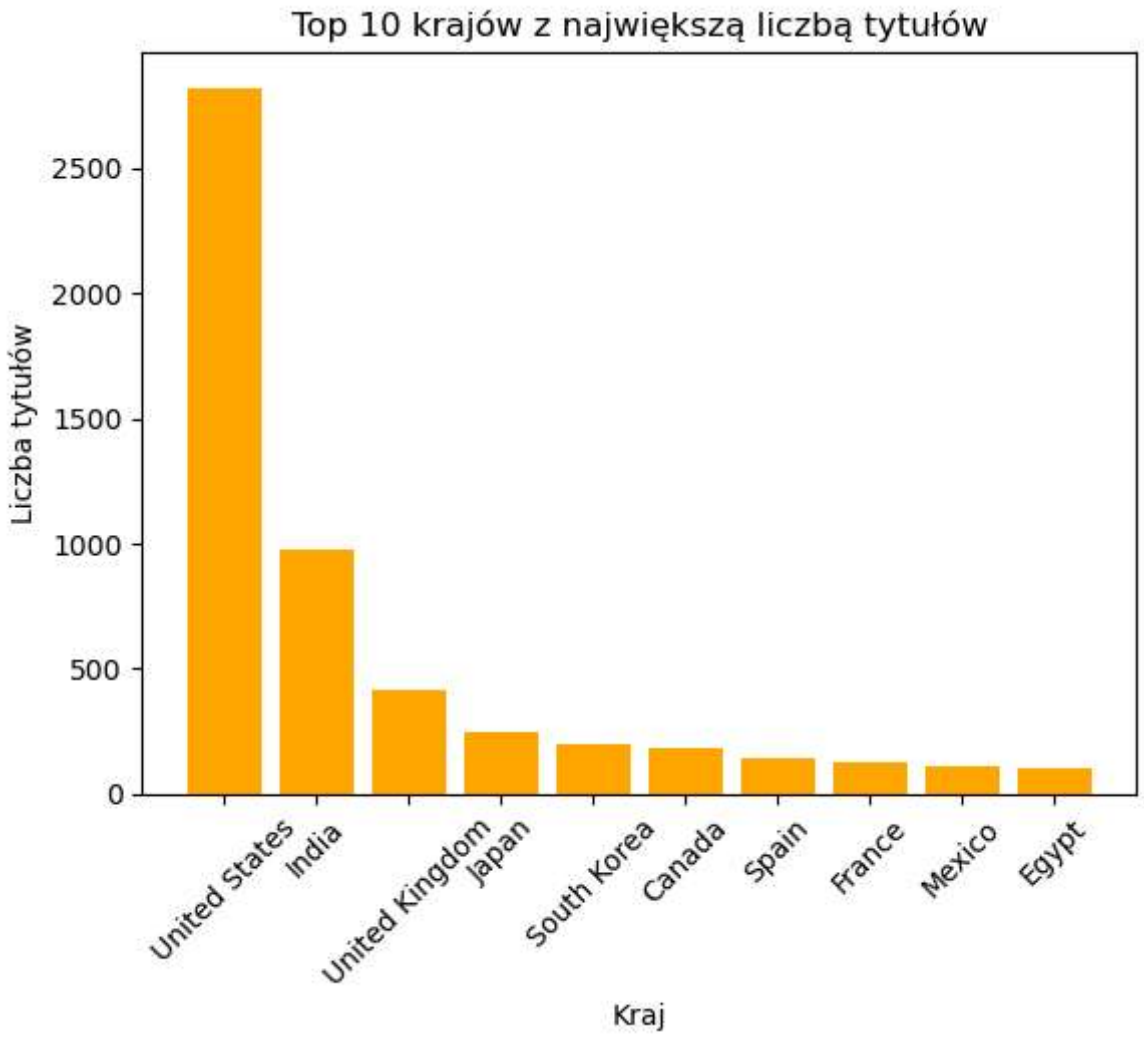
In [15]:
```
dane = df['release_year'].value_counts().sort_index()
plt.plot(dane.index, dane.values, marker='o', color='blue')
plt.title("Liczba filmów Netflixa w poszczególnych latach")
plt.xlabel("Rok wydania")
plt.ylabel("Liczba tytułów")
plt.show()
```



3.1.2 Wykres słupkowy

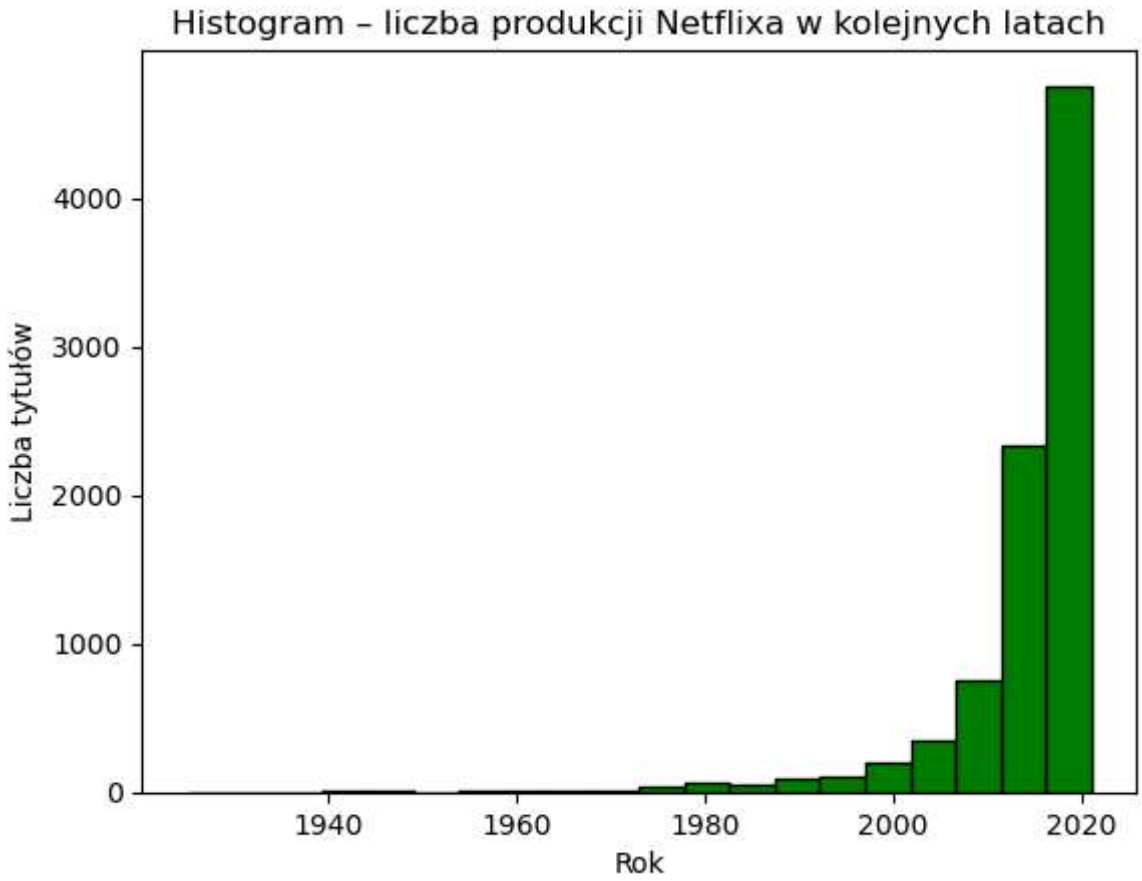
In [16]:
```
kraje = df['country'].value_counts().head(10)

plt.bar(kraje.index, kraje.values, color='orange')
plt.title("Top 10 krajów z największą liczbą tytułów")
plt.xlabel("Kraj")
plt.ylabel("Liczba tytułów")
plt.xticks(rotation=45)
plt.show()
```



3.1.3 Histogram

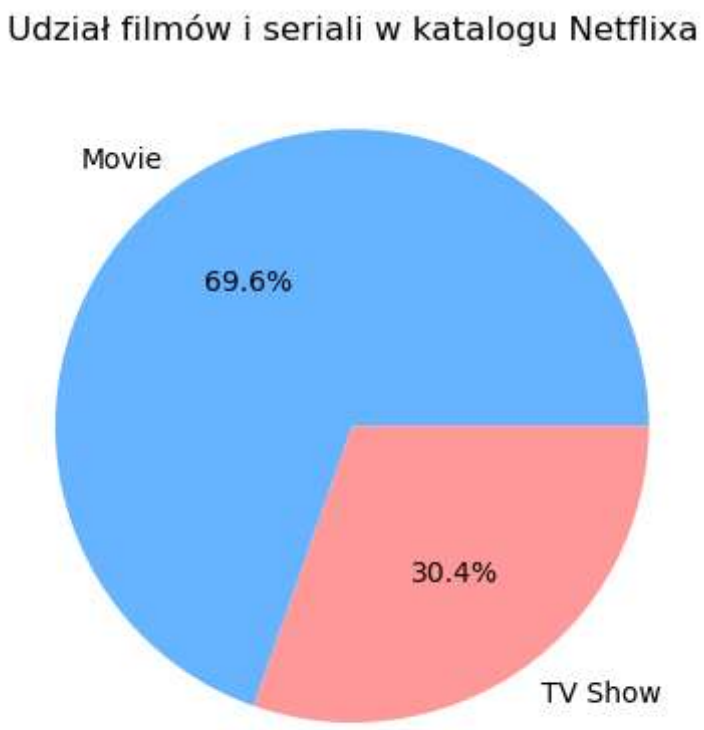
In [18]:
```
plt.hist(df['release_year'], bins=20, color='green', edgecolor='black')
plt.title("Histogram – liczba produkcji Netflixa w kolejnych latach")
plt.xlabel("Rok")
plt.ylabel("Liczba tytułów")
plt.show()
```



3.1.4 Wykres kołowy

In [17]:
```
typy = df['type'].value_counts()

plt.pie(typy.values, labels=typy.index, autopct='%1.1f%%', colors=['#66b3ff','#f
plt.title("Udział filmów i seriali w katalogu Netflixa")
plt.show()
```

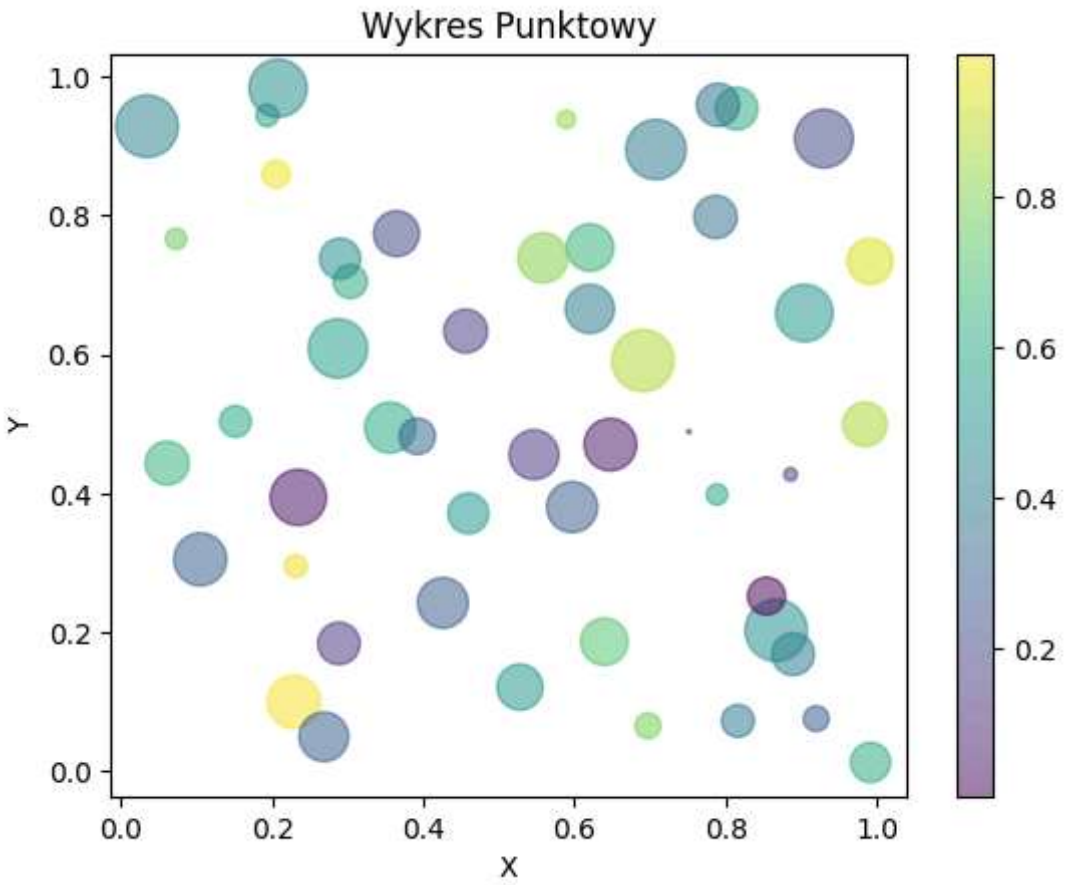


3.2 Zaawansowane typy wykresów

3.2.1 Wykres punktowy (scatter plot)

In [ ]:
```
x = np.random.rand(50)
y = np.random.rand(50)
sizes = 500 * np.random.rand(50)
colors = np.random.rand(50)

plt.scatter(x, y, s=sizes, c=colors, alpha=0.5, cmap="viridis")
plt.colorbar()
plt.xlabel("X")
plt.ylabel("Y")
plt.title("Wykres Punktowy")
plt.show()
```



3.2.2 Wykres 3D

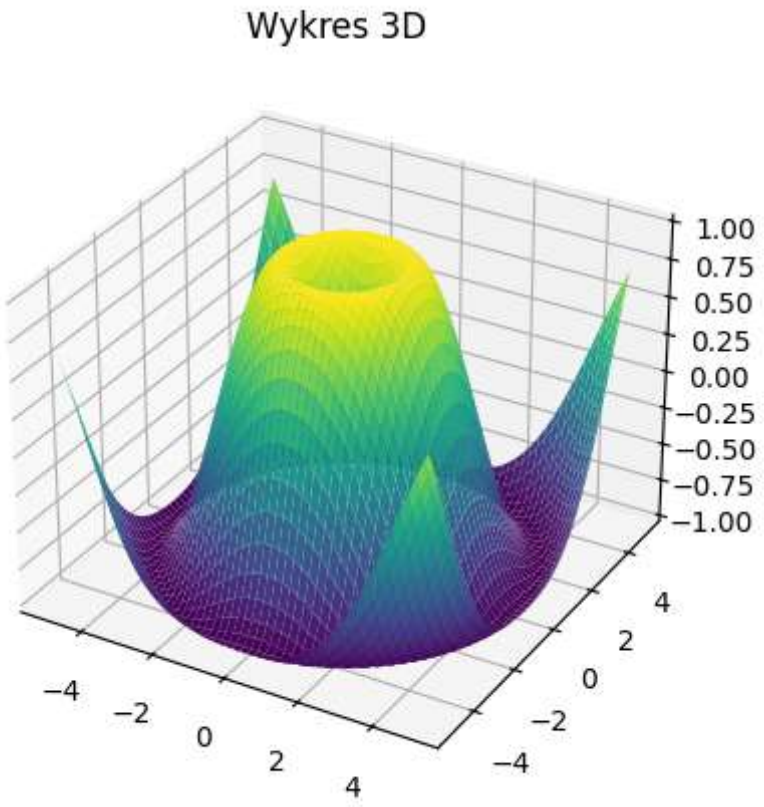
In [ ]:
```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()

ax = fig.add_subplot(111, projection="3d")

x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
X, Y = np.meshgrid(x, y)
Z = np.sin(np.sqrt(X**2 + Y**2))

ax.plot_surface(X, Y, Z, cmap="viridis")
plt.title("Wykres 3D")
plt.show()
```

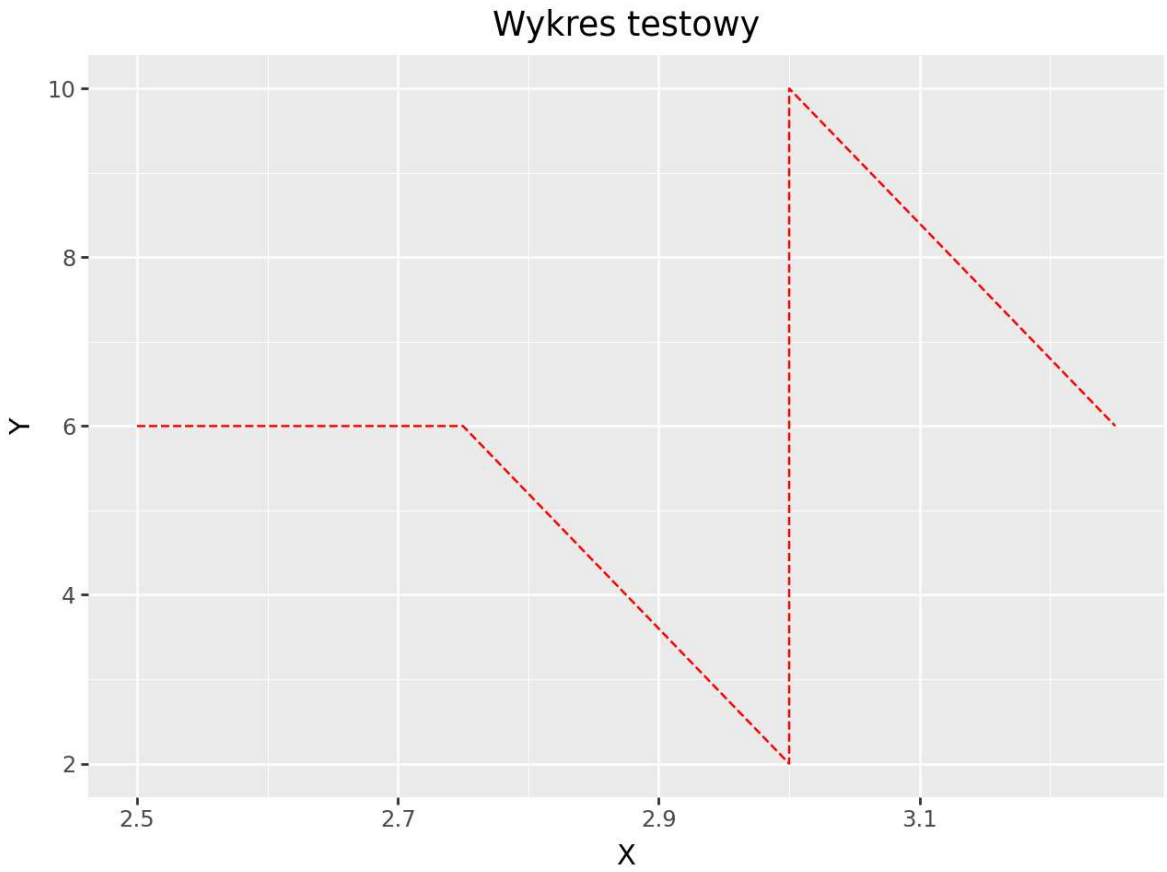


3.3 Bibilioteka Plotnine

In [ ]:
```
from plotnine import *
import pandas as pd
import numpy as np
```

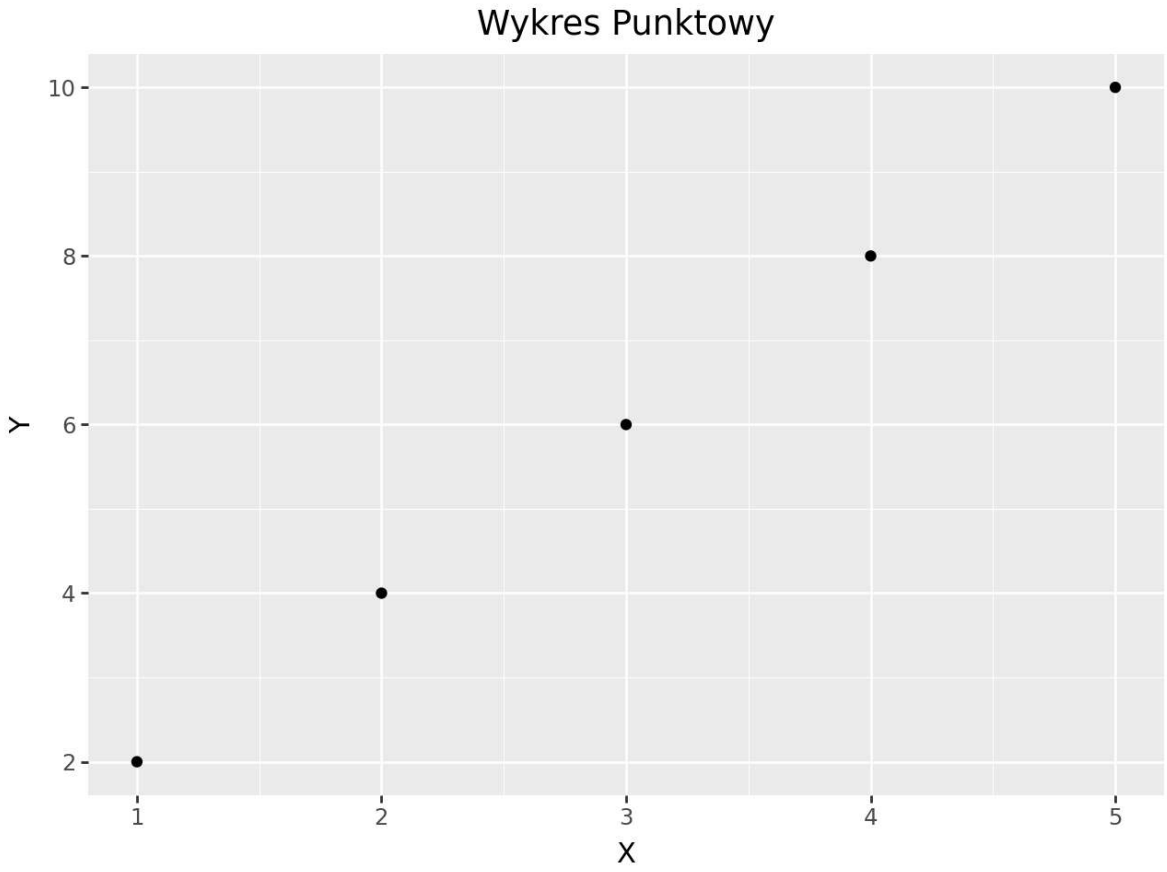
In [ ]:
```
data = pd.DataFrame({
 "x": [3, 3.25, 2.75, 3, 2.5],
 "y": [2, 6, 6, 10, 6]
 })
```

In [ ]:
```
ggplot(data) + aes(x="x", y="y") + geom_line(color="red", linetype="dashed") + g
```



3.3.1 Wykres punktowy

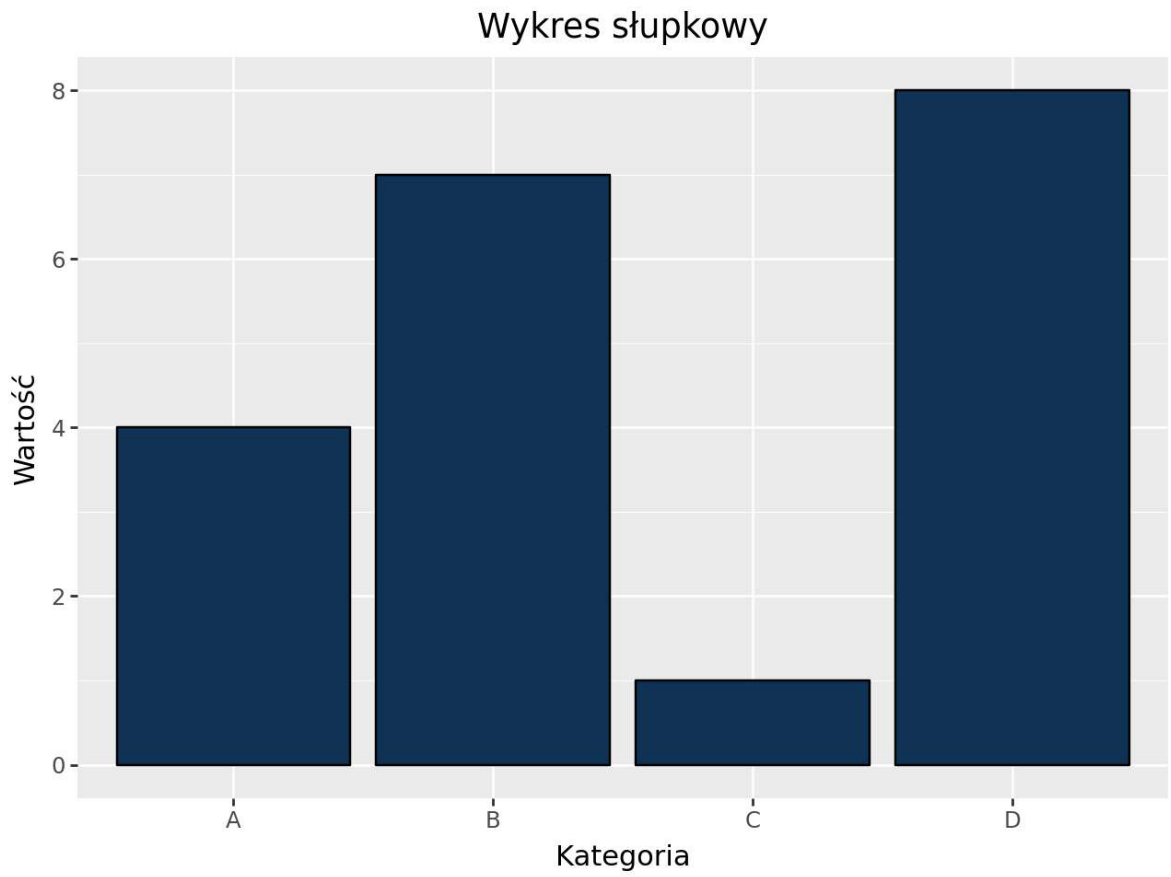
In [ ]:
```
ggplot(data) + aes(x="x", y="y") + geom_point() + ggtitle("Wykres Punktowy") + x
```



3.3.2 Wykres punktowy

In [ ]:
```
data_bar = pd.DataFrame({
 "kategorie": ["A", "B", "C", "D"],
 "wartości": [4, 7, 1, 8]
 })
```

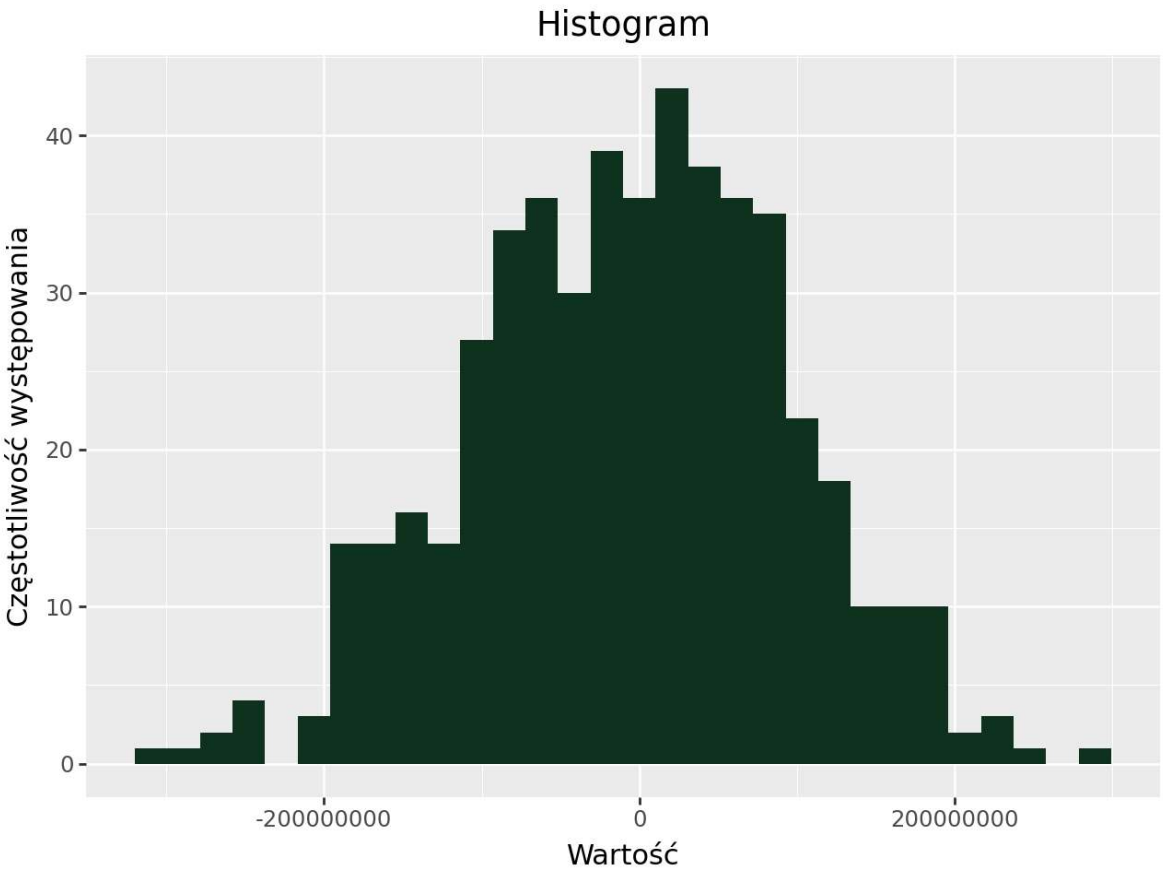
In [ ]:
```
ggplot(data_bar) + aes(x="kategorie", y="wartości") + geom_bar(stat="identity",
```



3.3.3 Histogram

In [ ]:
```
data_hist = pd.DataFrame({
    "wartosci": np.random.normal(0, 100_000_000, 500)
})

ggplot(data_hist) + aes(x="wartosci") + geom_histogram(bins=30, fill="#123321")
```



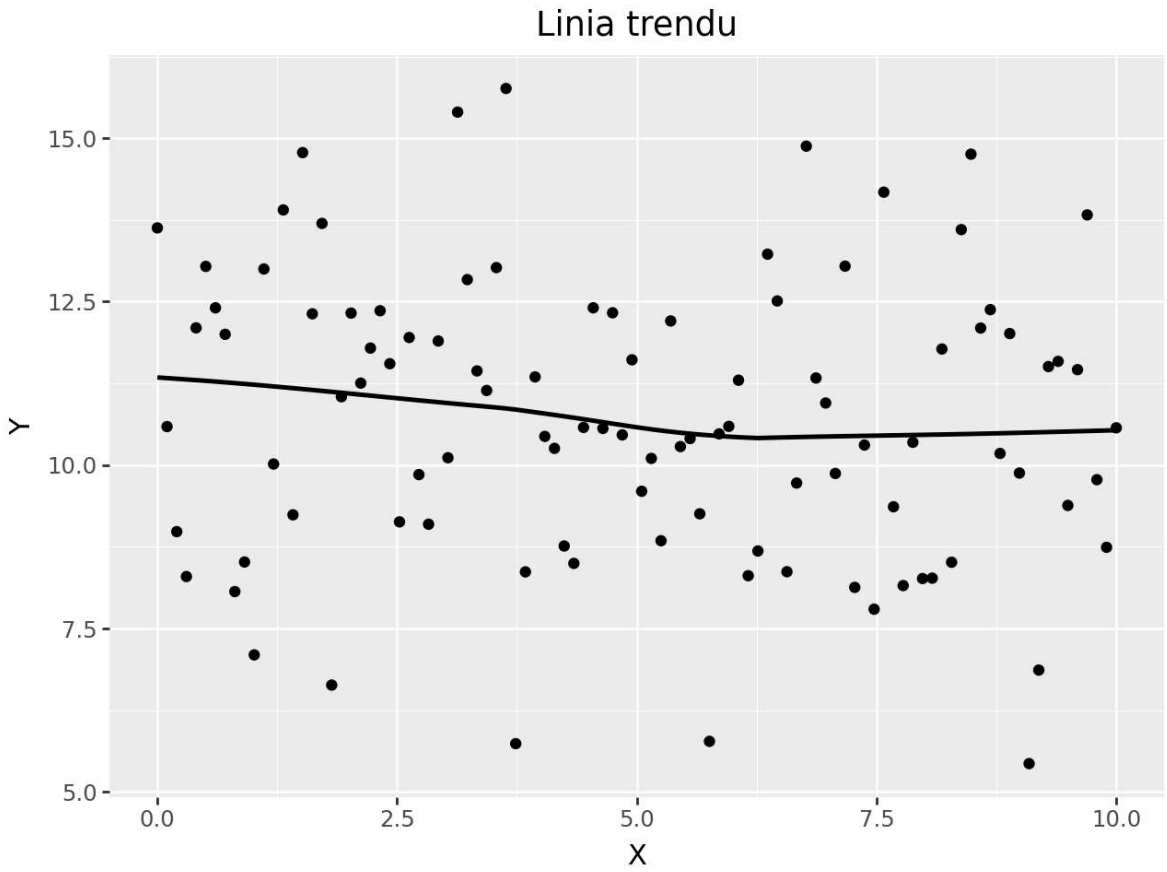
3.3.4 Linia trendu

In [ ]:
```
const = 5

data_trend = pd.DataFrame({
  "x" : np.linspace(0, 10, 100),
  "y" : (2 * const + 1 + np.random.normal(0, 2, 100))
})

ggplot(data_trend) + aes(x="x", y="y") + geom_point() + geom_smooth() + ggtitle(
```

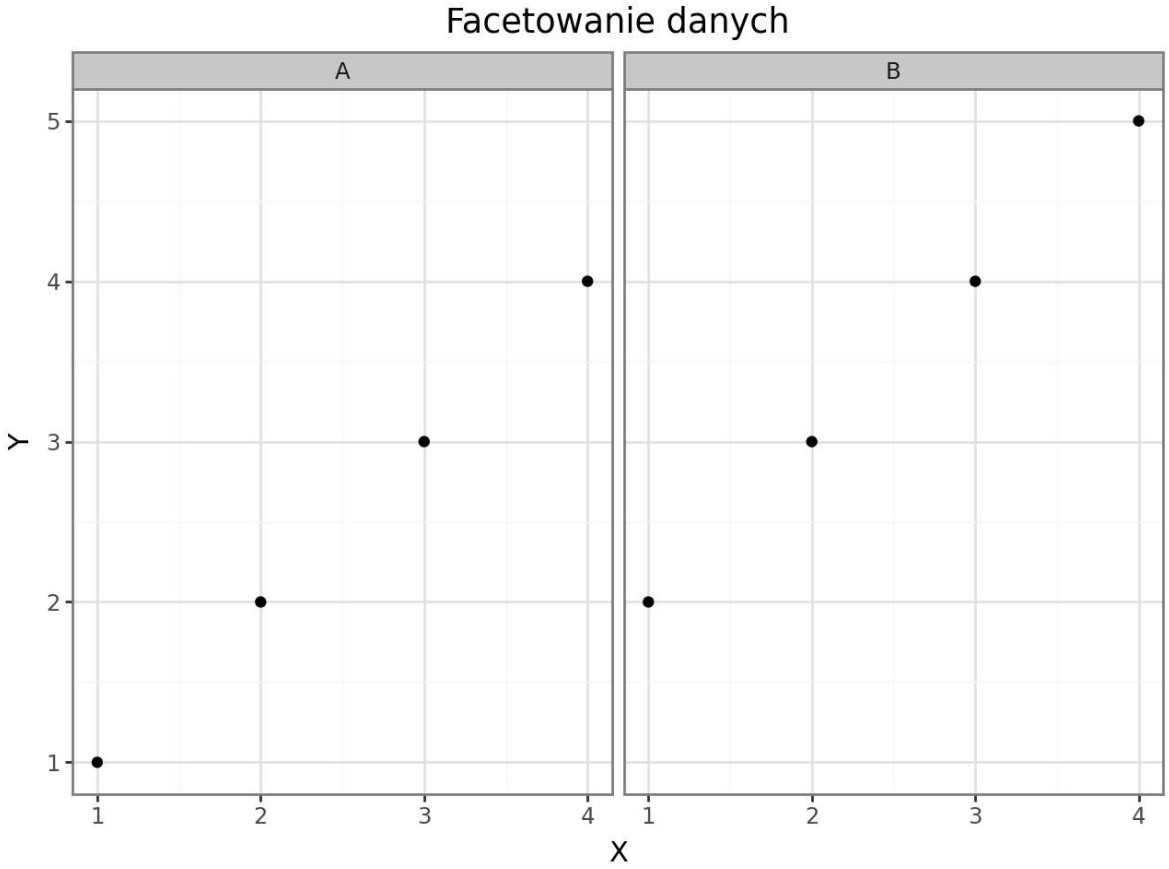
```
/usr/local/lib/python3.12/dist-packages/plotnine/stats/smoothers.py:347: PlotnineWarning: Confidence intervals are not yet implemented for lowess smoothings.
```



3.3.5 Facetowanie danych (podział na podgrupy)

In [ ]:
```
data_facet = pd.DataFrame({
 "x": np.tile([1, 2, 3, 4], 2),
 "y": [1, 2, 3, 4, 2, 3, 4, 5],
 "grupa": ["A"]*4 + ["B"]*4
 })

ggplot(data_facet) + aes(x="x", y="y") + geom_point() + facet_wrap("grupa") + gg
```



3.4 Wykresy przestrzenne

3.4.1 Mapy punktowe

In [ ]:
```
import plotly.express as px
import pandas as pd
```

In [ ]:
```
data_map = pd.DataFrame({
  "city": ["New York", "London", "Tokyo", "Sydney"],
  "latitude": [40.7128, 51.5074, 35.6895,-33.8688],
  "longitude": [-74.0060,-0.1278, 139.6917, 151.2093]
})
```

In [ ]:
```
fig = px.scatter_mapbox(data_map, lat="latitude", lon="longitude", hover_name="c
fig.update_layout(mapbox_style="open-street-map")
fig.update_layout(title="Lokalizacja Wybranych Miast na Mapie")
fig.show()
```

3.4.2 Mapy o zmiennym kolorze wartości (Choropleth)

In [ ]:
```
import plotly.express as px
```

In [ ]:
```
data = px.data.gapminder().query("year == 2007")

fig = px.choropleth(data, locations="iso_alpha", color="pop", hover_name="countr

fig.update_layout(title="Populacja Kraj w na wiecie w 2007 roku")
fig.show()
```