

k04ldqjcr

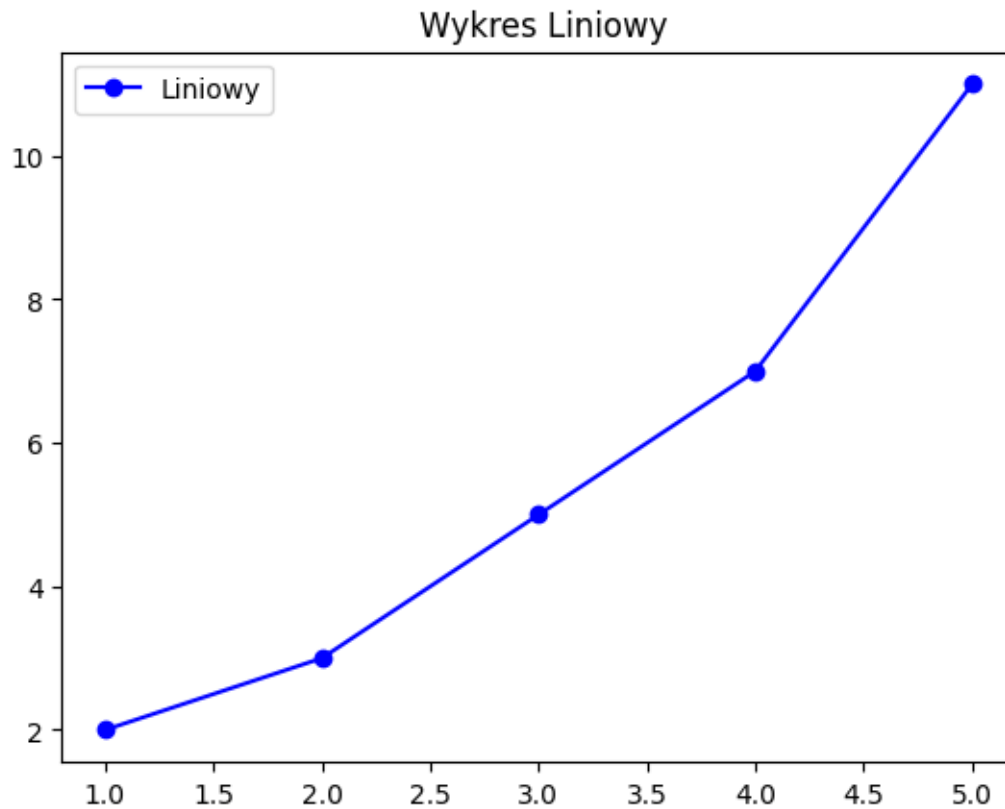
January 2, 2025

```
[82]: import matplotlib . pyplot as plt
```

```
[83]: #Wykres liniowy

x = [1 , 2 , 3 , 4 , 5]
y = [2 , 3 , 5 , 7 , 11]

plt.plot(x , y , marker = 'o' , linestyle = '-' , color = 'b' , label = 'Liniowy' )
plt.title(" Wykres Liniowy")
plt.legend()
plt.show()
```

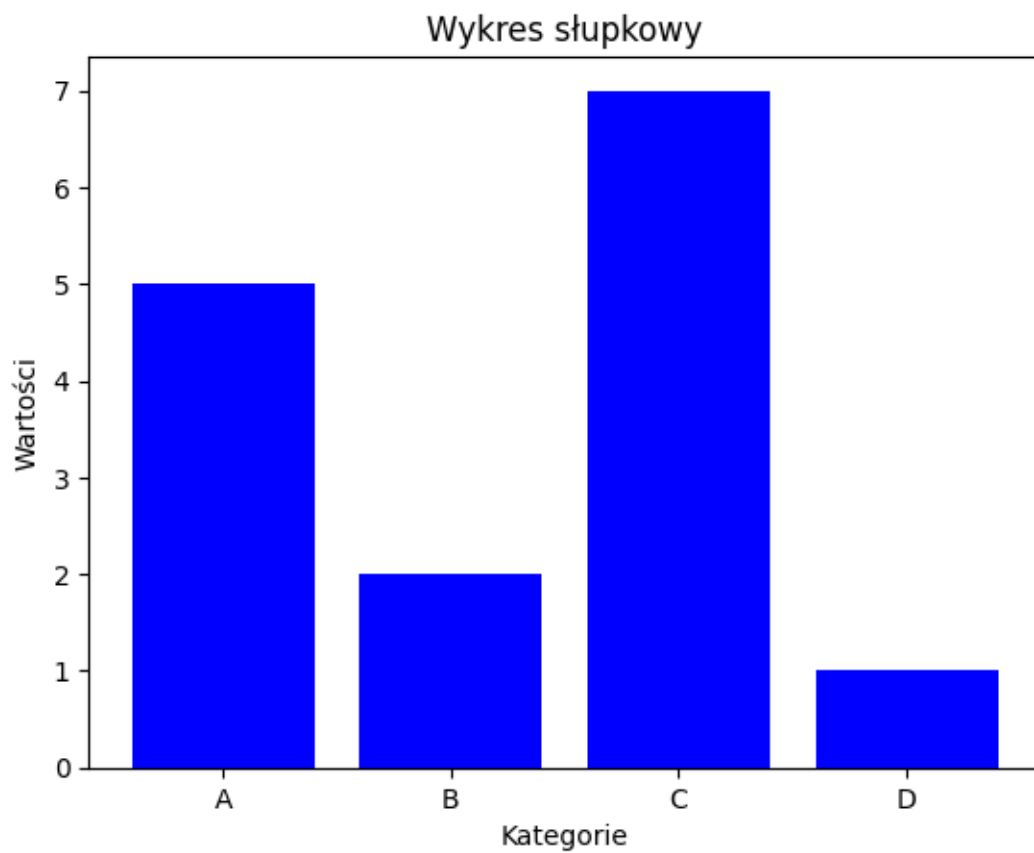


```
[84]: #Wykres słupkowy

kategorie = ['A', 'B', 'C', 'D']
wartosci = [5,2,7,1]

#Tworzenie wykresu słupkowego

plt.bar(kategorie, wartosci, color="blue")
plt.xlabel("Kategorie")
plt.ylabel("Wartości")
plt.title("Wykres słupkowy")
plt.show()
```



```
[85]: #histogram
import numpy as np

#Dane
```

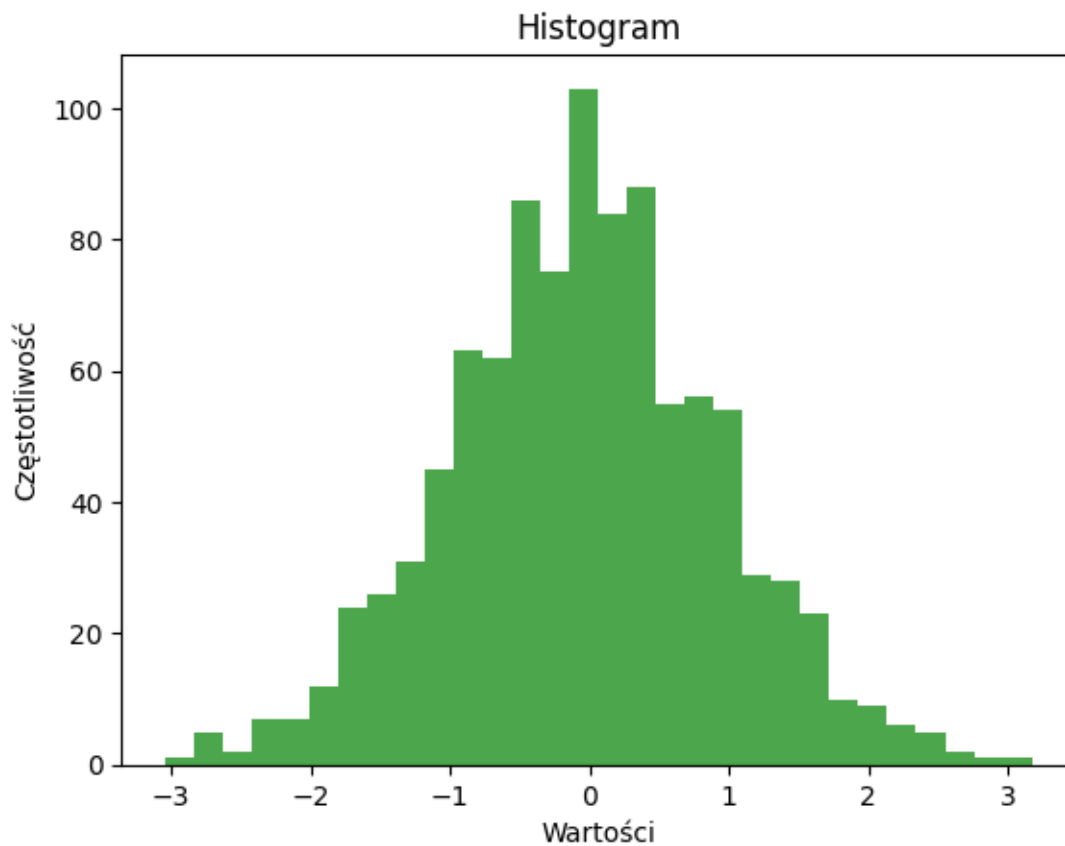
```

dane = np.random.normal(0, 1, 1000)

#tworzenie histogramu

plt.hist(dane, bins=30, color="green", alpha=0.7)
plt.xlabel("Wartości")
plt.ylabel("Częstotliwość")
plt.title("Histogram")
plt.show()

```



```

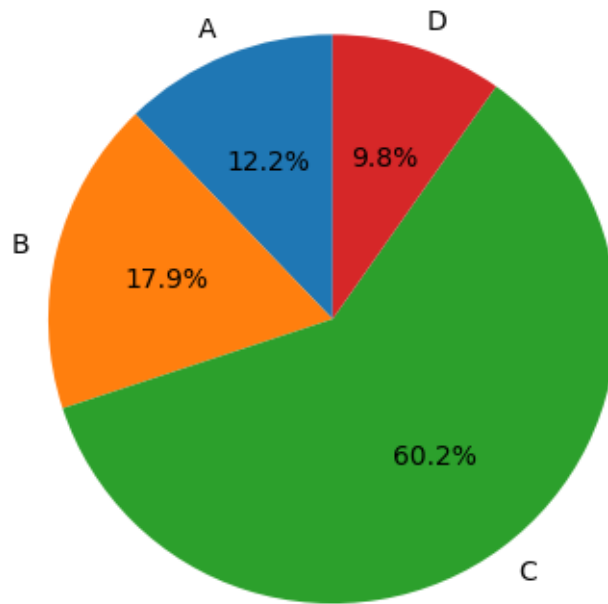
[86]: #wykres kołowy
kategorie = ['A', 'B', 'C', 'D']
wartosci = [15,22,74,12]

#tworzenie wykresu kołowego

plt.pie(wartosci, labels=kategorie, autopct="%1.1f%%", startangle=90)
plt.title("Wykres kołowy")
plt.show()

```

Wykres kołowy



[87]: *#Dostosowywanie wykresów*

#kolory, linie i style

```
x = [1,2,3,4,5]
```

```
y=[10, 20, 25, 30, 35]
```

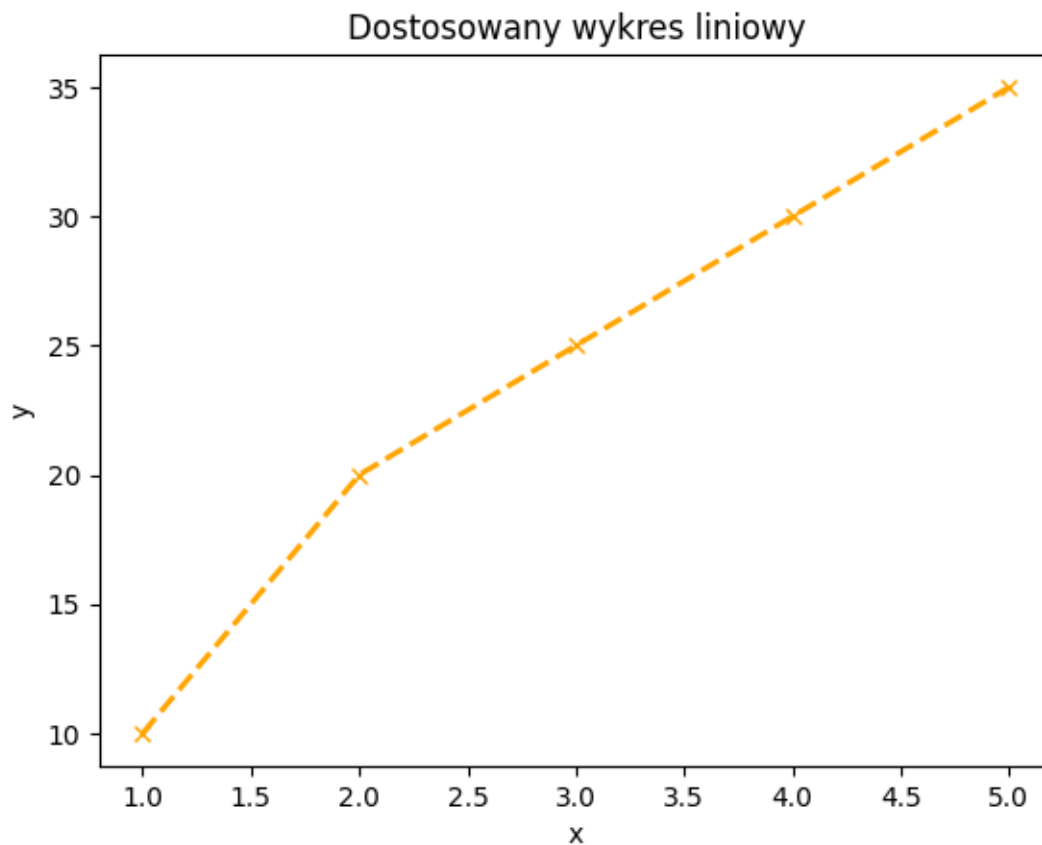
```
plt.plot(x,y, color="orange", marker="x", linestyle="--", linewidth=2)
```

```
plt.xlabel("x")
```

```
plt.ylabel("y")
```

```
plt.title("Dostosowany wykres liniowy")
```

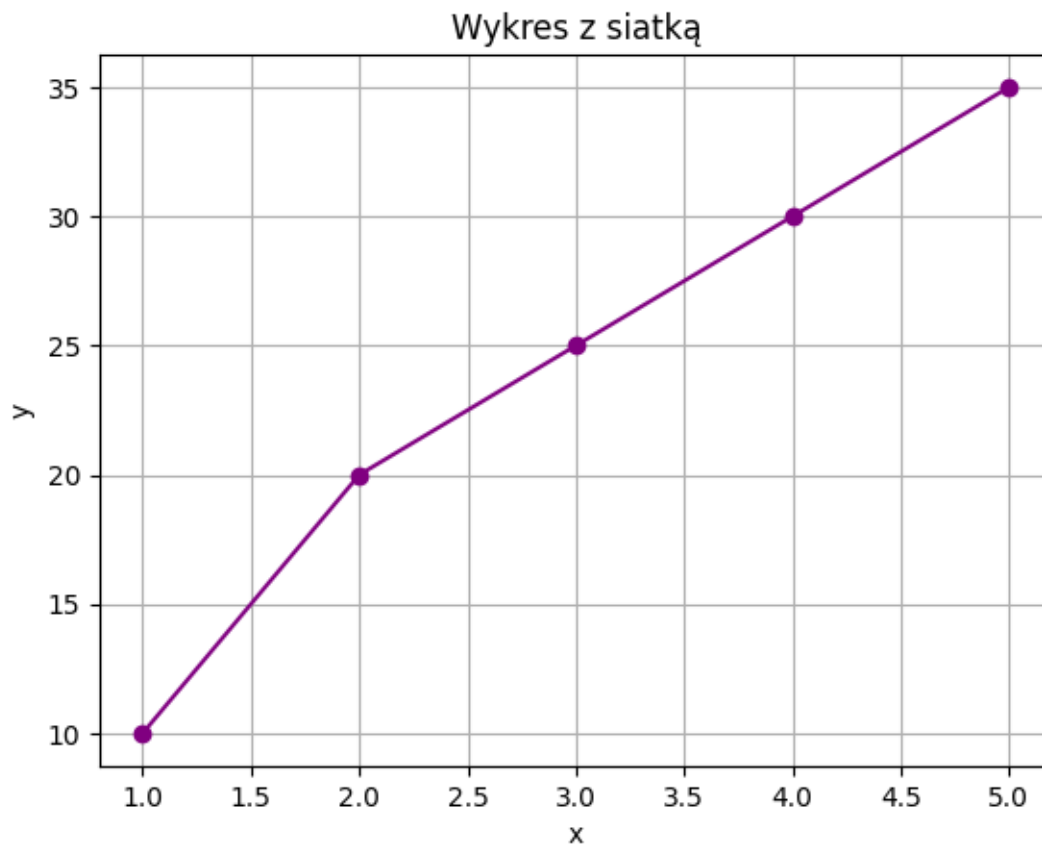
```
plt.show()
```



```
[88]: #Dodawanie siatki

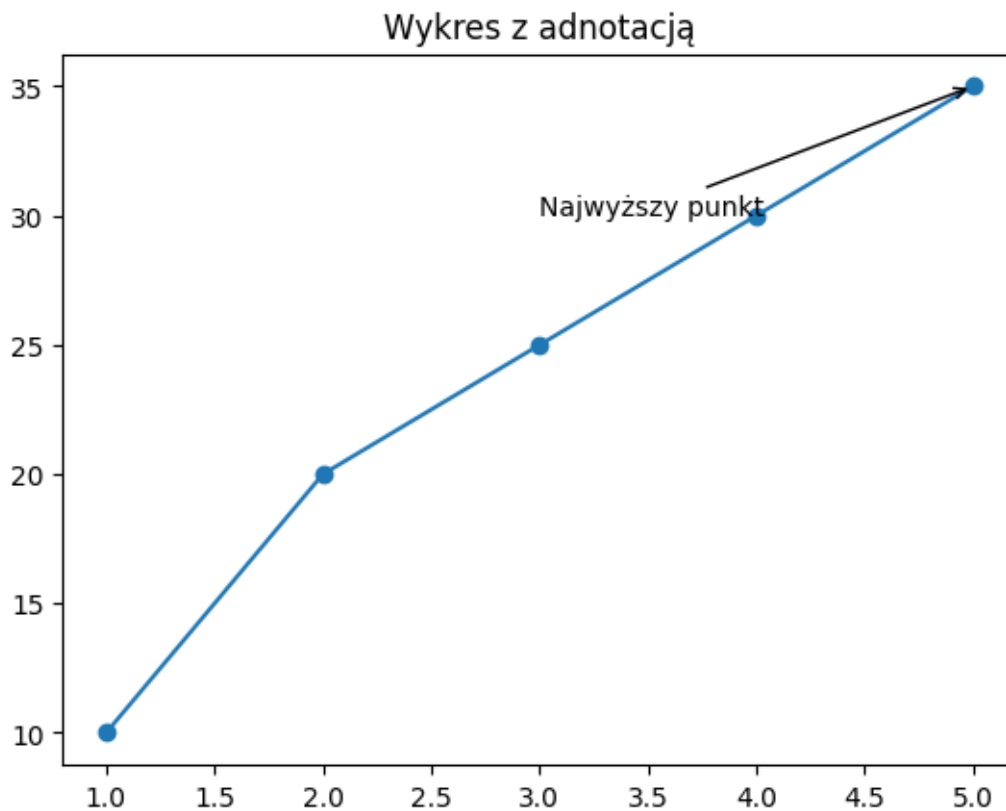
plt.plot(x,y,color="purple", marker="o")
plt.grid(True)

plt.xlabel("x")
plt.ylabel("y")
plt.title("Wykres z siatką")
plt.show()
```



```
[89]: #dodawanie adnotacji

plt.plot(x,y,marker="o")
plt.annotate("Najwyższy punkt", xy=(5,35), xytext=(3, 30),
            ↪arrowprops=dict(facecolor="pink", arrowstyle="->"))
plt.title("Wykres z adnotacją")
plt.show()
```



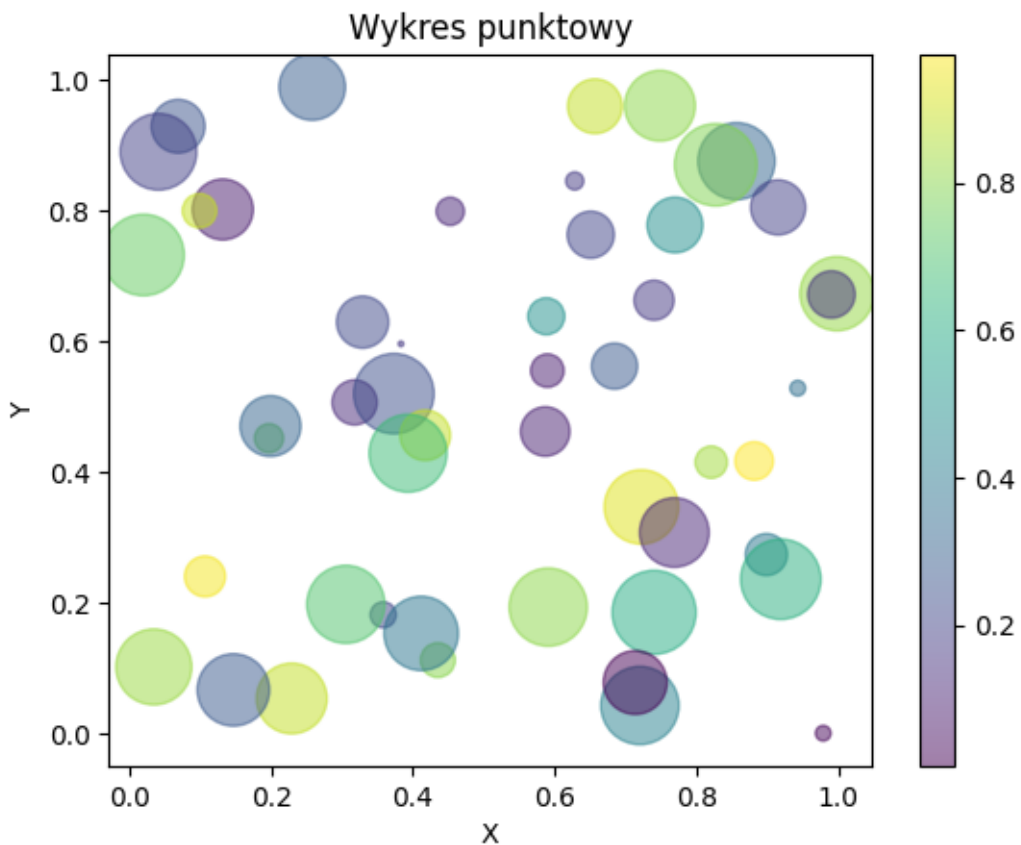
```
[90]: #Wykres punktowy

x = np.random.rand(50)
y = np.random.rand(50)

sizes = 1000 * 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.ylabel("Y")
plt.xlabel("X")
plt.title("Wykres punktowy")
plt.show()
```



```
[91]: #wykres 3d

from mpl_toolkits.mplot3d import Axes3D

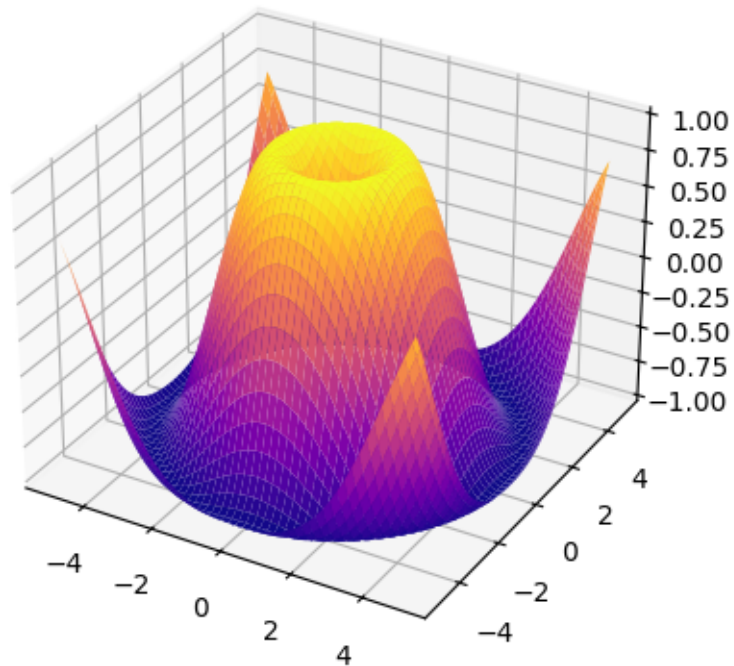
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)

x, y = np.meshgrid(x, y)

z = np.sin(np.sqrt(x**2 + y ** 2))

ax.plot_surface(x, y, z, cmap='plasma')
plt.title("Wykres 3D")
plt.show()
```


Wykres 3D



```
[92]: #wykres liniowy plotly
import plotly.express as px
#dane
x = [1,2,3,4,5]
y=[10,15,24,13,23]
#tworzenie wykresu
fig = px.line(x=x, y=y, title="wykres liniowy", labels={'x':"0 x", 'y':"0 y"})
fig.show()
```

```
[93]: #wykres słupkowy
kategorie = ['A', 'B', 'C', 'D']
wartosci = [4,5,7,8]

#Tworzenie wykresu słupkowego

fig = px.bar(x=kategorie, y=wartosci, title="Wykres słupkowy", labels={'x': 'Kategorie', 'y': "Wartości"})
fig.show()
```

```
[94]: #Wykres kołowy

fig = px.pie(names=kategorie, values=wartosci, title="Wykres kołowy")
```

```
fig.show()
```

```
[95]: #Histogram

import numpy as np

dane = np.random.normal(0, 1, 1000)

#tworzenie histogramu
fig = px.histogram(x=dane, nbins=30, title="Histogram", labels={'x': 'Wartości', 'y': "Częstotliwość"})
fig.show()
```

```
[96]: #dostosowywanie wykresów

fig = px.line(x=x, y=y, title="Wykres liniowy z dostosowanymi kolorami")
fig.update_traces(line=dict(color="purple", width=4))
fig.show()
```

```
[97]: #dodawanie adnotacji

fig = px.line(x=x, y=y, title="Wykres z adnotacjami")
fig.add_annotation(x=3, y=13, text="Wyróżniony punkt", showarrow=True,
    ↪arrowhead=1)
fig.show()
```

```
[98]: #Tworzenie zaawansowanych wykresów

#Wykres punktowy (Scatter Plot)

x = np.random.rand(50)
y = np.random.rand(50)
sizes = 1000 * np.random.rand(50)
colors = np.random.rand(50)

fig = px.scatter(x=x, y=y, size=sizes, color=colors, title= "Wykres punktowy")
fig.show()
```

```
[99]: import plotly . graph_objects as go
# Dane
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))
#Tworzenie wykresu 3D
fig = go.Figure(data = [go.Surface(z = Z , x = X , y = Y)])
fig.update_layout(title = "Wykres 3D" , scene = dict(
```

```

axis_title = 'X',
axis_title = 'Y',
axis_title = 'Z'))
fig.show()

```

```

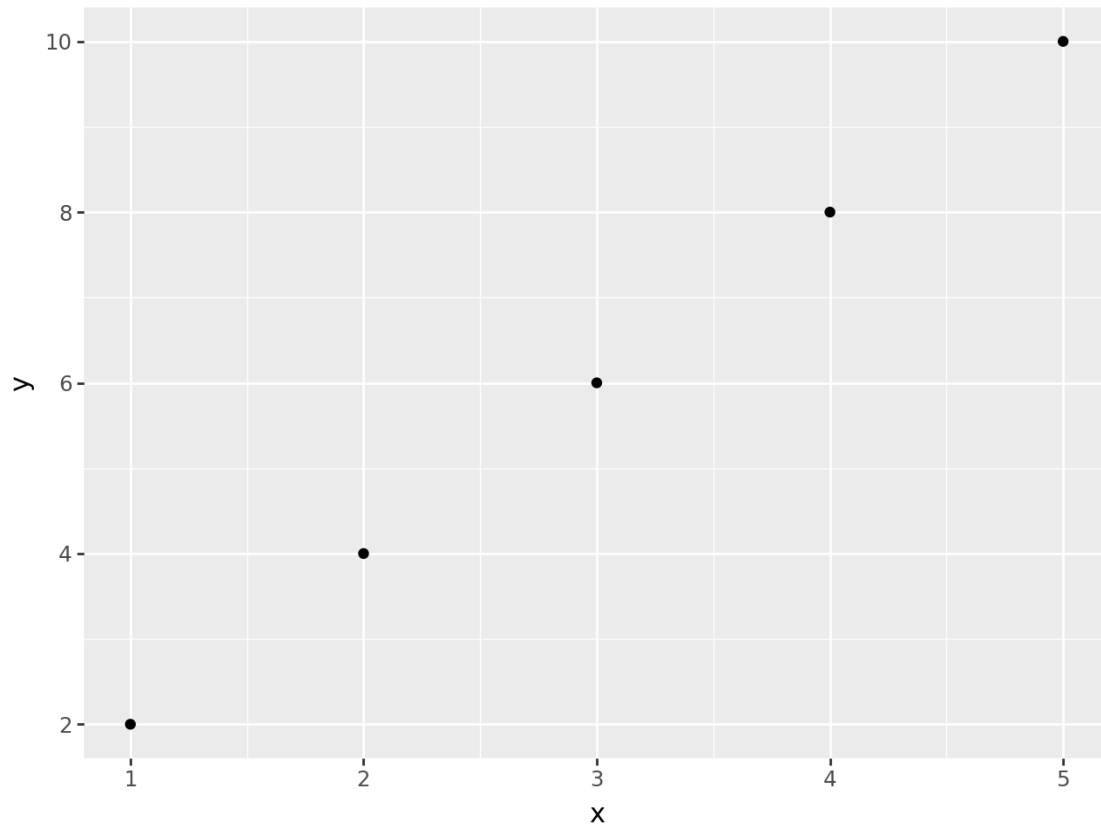
[100]: from plotly.subplots import make_subplots
        #Dane
        x = [1, 2, 3, 4, 5]
        y1 = [10, 15, 13, 17, 10]
        y2 = [5, 6, 2, 3, 8]
        #Tworzenie subplotów
        fig = make_subplots(rows = 1, cols = 2, subplot_titles = ( "Wykres 1", "Wykres
        ↪2"))
        fig.add_trace(go.Scatter(x = x, y = y1, mode = 'lines + markers', name =
        ↪"Liniowy 1"), row =1, col = 1)
        fig.add_trace(go.Scatter(x = x, y = y2, mode = 'lines + markers', name =
        ↪"Liniowy 2"), row =1, col = 2)
        fig.update_layout(title = "Subploty w Plotly")
        fig.show()

```

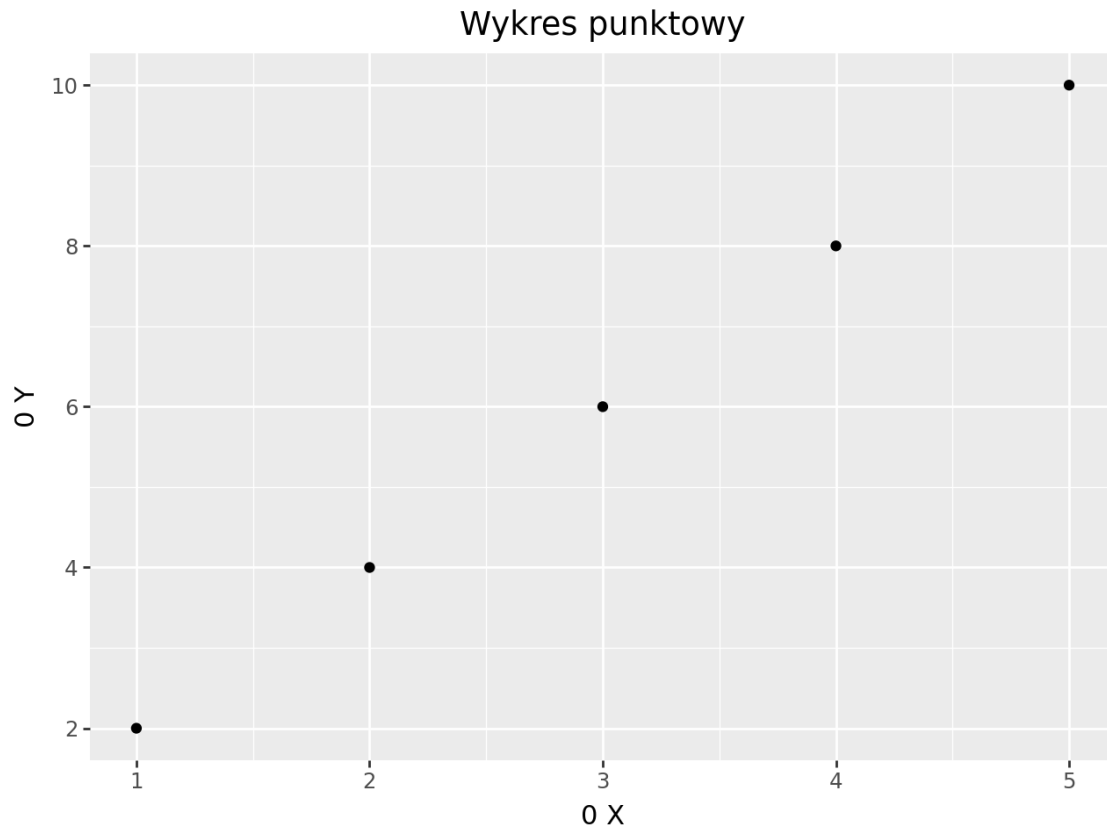
```

[101]: from plotnine import *
        import pandas as pd
        #przykładowe dane
        data = pd.DataFrame({
            'x': [1,2,3,4,5],
            'y': [2,4,6,8,10]
        })
        #tworzenie podstawowego wykresu
        (ggplot(data) + aes(x = 'x', y = 'y') + geom_point())

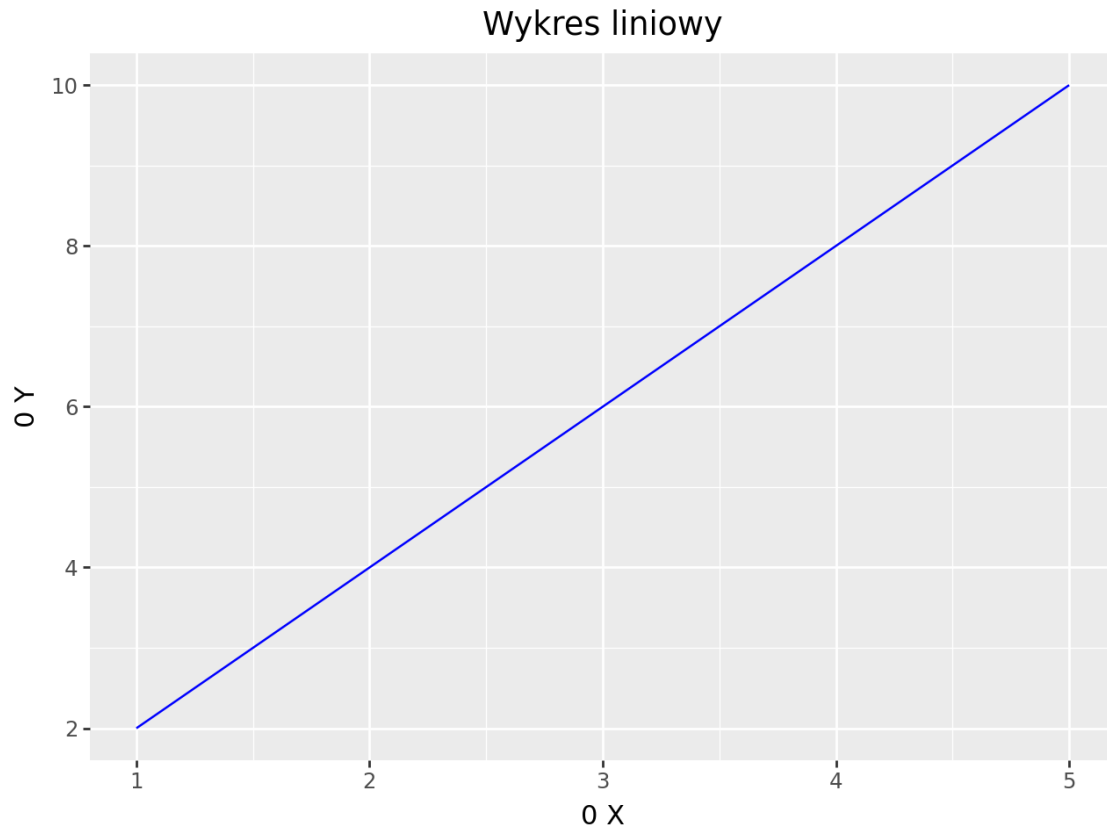
```



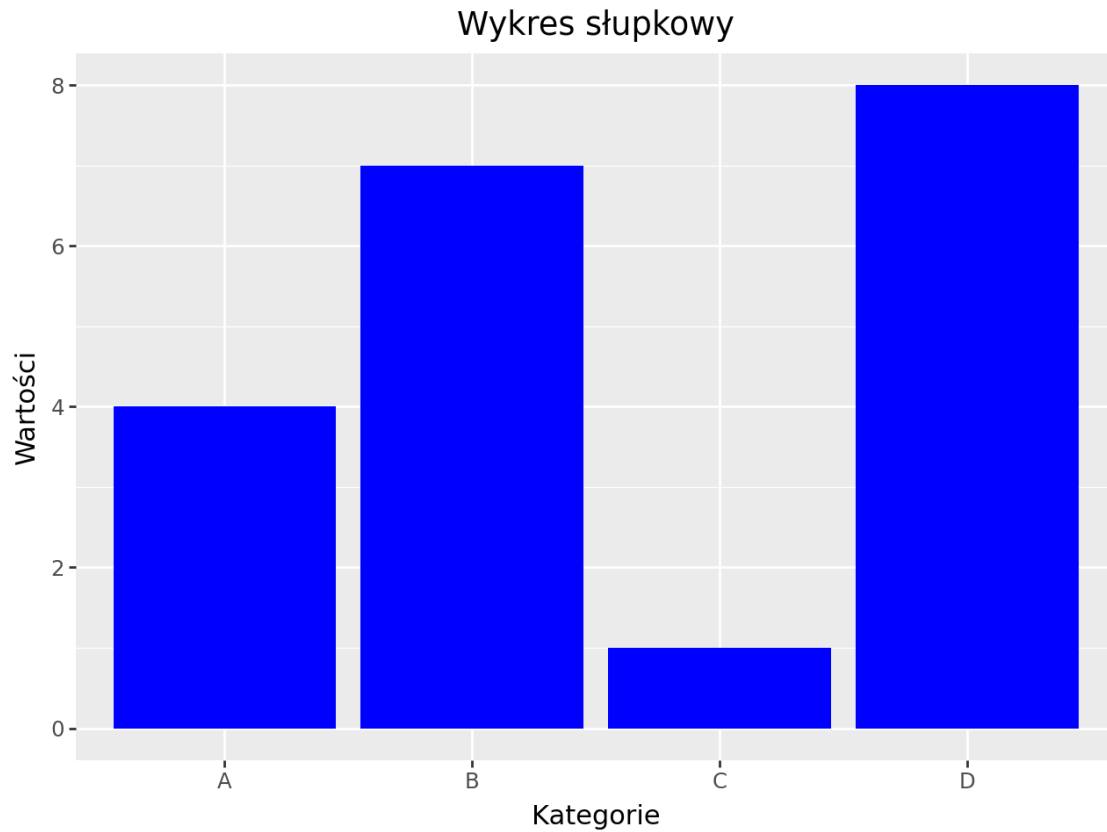
```
[102]: #wykres punktowy
(ggplot(data) + aes(x = 'x', y = 'y') + geom_point() + ggtitle("Wykres
↪punktowy") + xlab("X") + ylab("Y"))
```



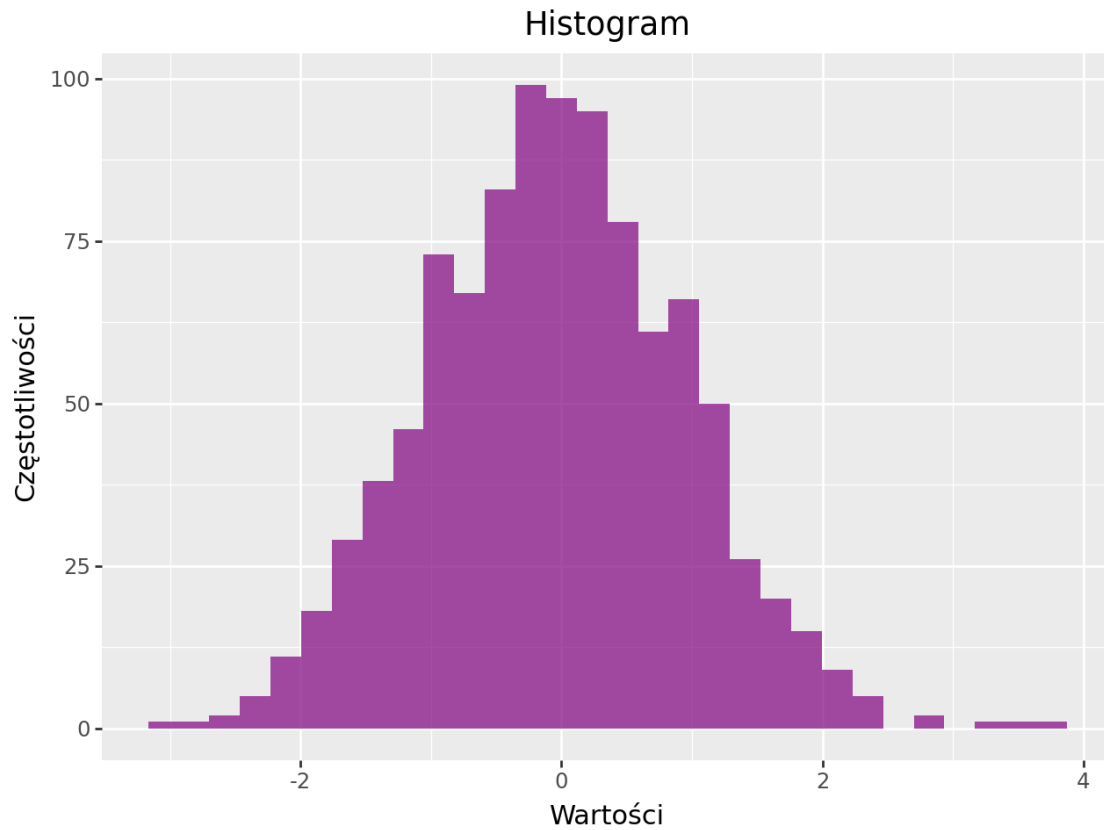
```
[103]: #wykres liniowy
(ggplot(data) + aes(x = 'x', y = 'y') + geom_line(color='blue') +
  ↪ggtitle("Wykres liniowy") + xlab("O X") + ylab("O Y"))
```



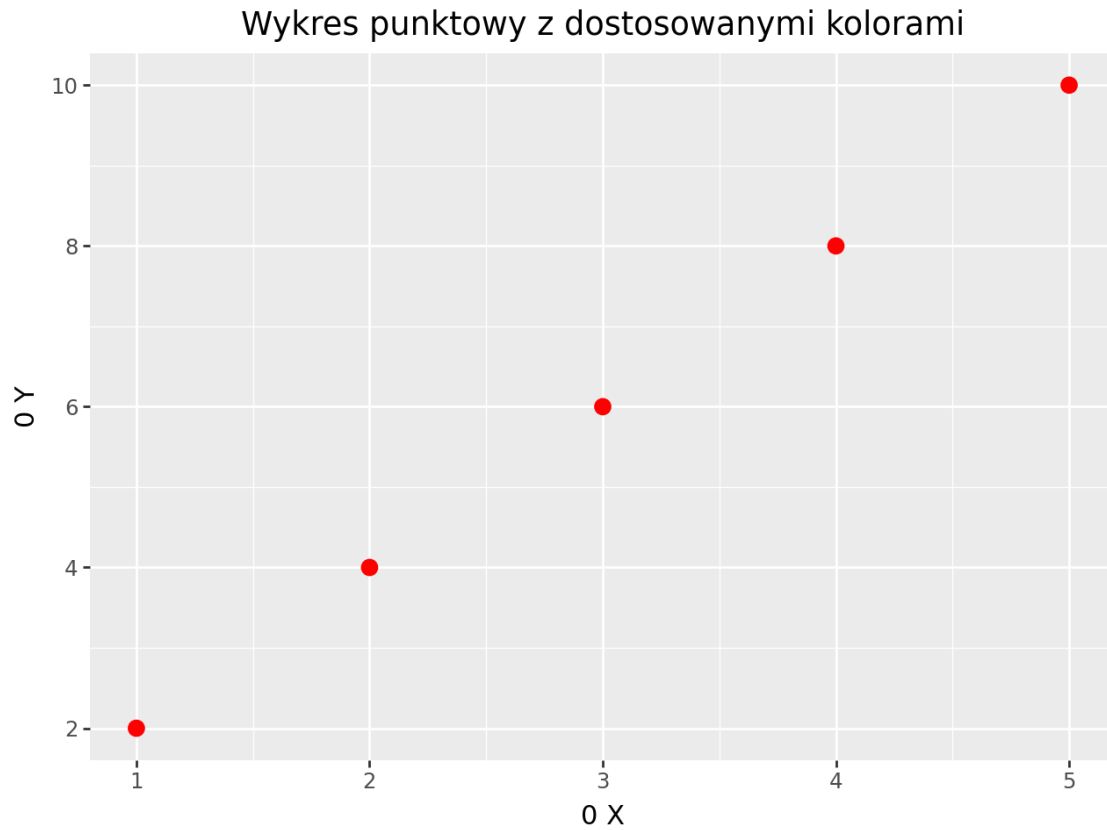
```
[104]: # Dane do wykresu słupkowego
data_bar = pd.DataFrame ({
    'kategorie': ['A', 'B', 'C', 'D'],
    'wartosci': [4 , 7 , 1 , 8]
})
# Wykres słupkowy
(ggplot(data_bar) + aes(x= 'kategorie', y ='wartosci') + geom_bar(stat =_
↪ 'identity', fill = 'blue') + ggtitle("Wykres słupkowy") + xlab("Kategorie")_
↪ ylab("Wartości"))
```



```
[105]: #Histogram
import numpy as np
#dane
data_hist = pd.DataFrame({'wartości': np.random.normal(0, 1, 1000)})
#histogram
(ggplot(data_hist) + aes(x= 'wartości') + geom_histogram(bins = 30, fill = 'purple', alpha = 0.7) + ggtitle("Histogram") + xlab("Wartości") + ylab("Częstotliwość"))
```

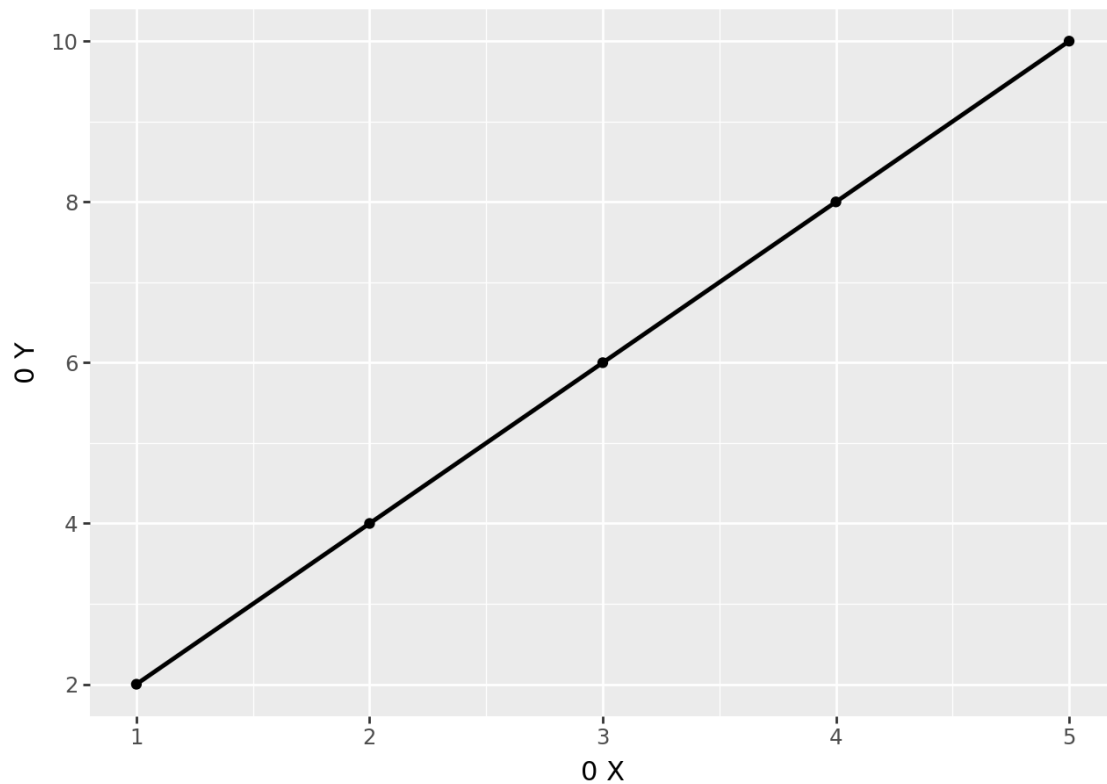


```
[106]: #wykres punktowy z dostosowanymi kolorami
(ggplot(data) + aes(x = 'x', y = 'y') + geom_point(color = 'red', size = 3) +
  ↳ ggtitle("Wykres punktowy z dostosowanymi kolorami") + xlab("X") + ylab("Y"))
```

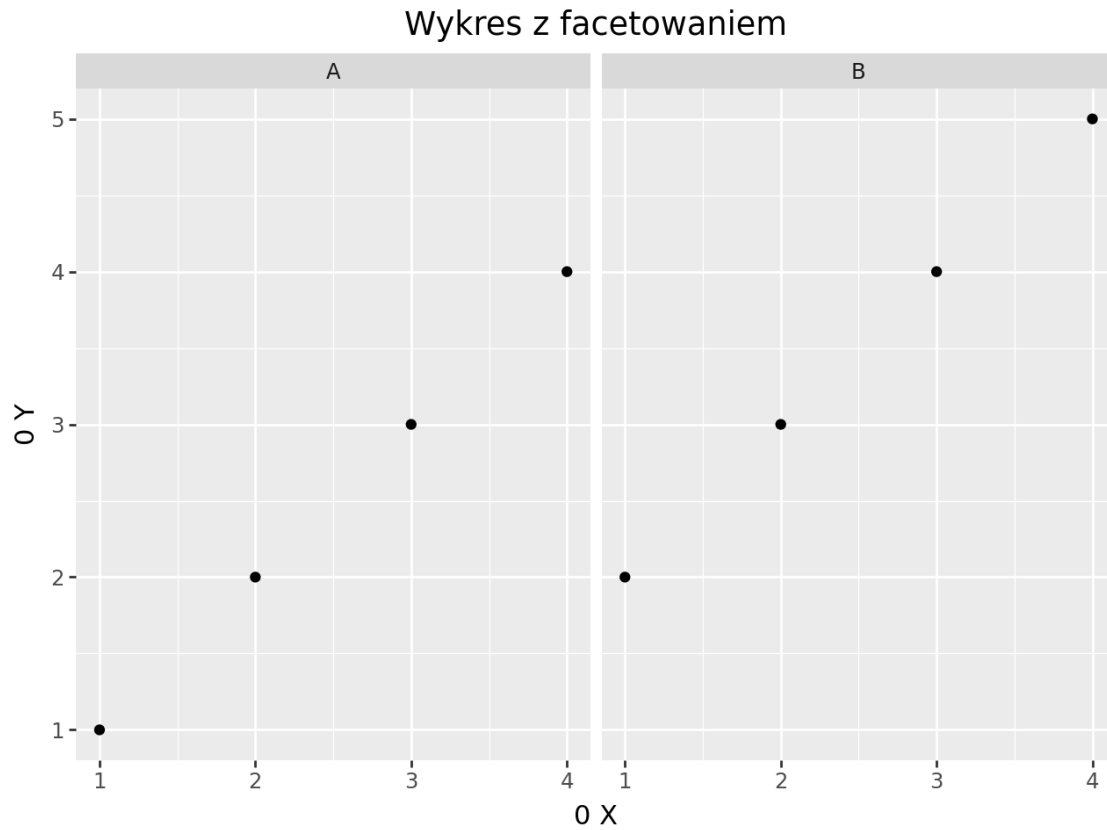



```
[107]: # wykres punktowy z linią trendu
(ggplot(data) + aes(x = 'x', y = 'y') + geom_point() + geom_smooth(method = 'lm') + ggtitle("Wykres punktowy z dostosowanymi kolorami") + xlab("O X") + ylab("O Y"))
```

Wykres punktowy z dostosowanymi kolorami

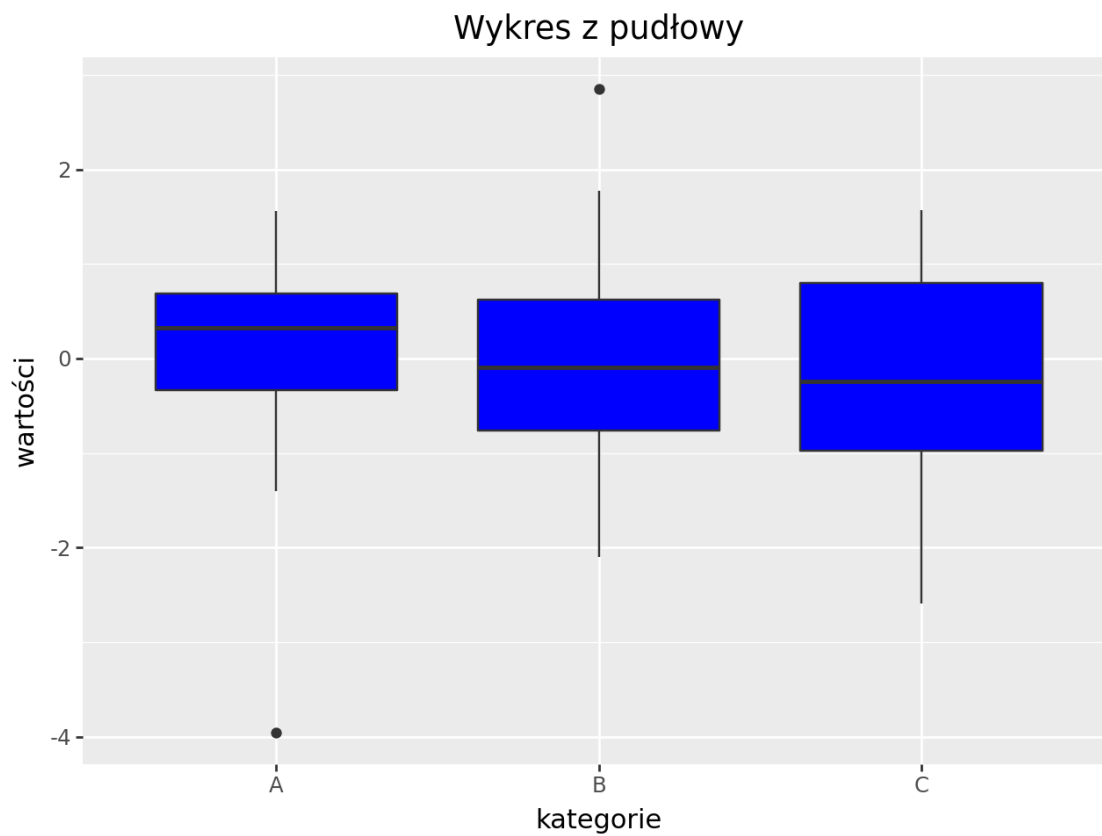


```
[139]: #przykładowe dane
data_facet = pd.DataFrame({
    'x': np.tile([1, 2, 3, 4], 2),
    'y': [1, 2, 3, 4, 2, 3, 4, 5],
    'grupa': ['A']*4 + ['B']*4
})
#wykres z facetowaniem
(ggplot(data_facet) + aes(x = 'x', y = 'y') + geom_point() +
  ↳facet_wrap('~grupa') + ggtitle("Wykres z facetowaniem") + xlab("X") +
  ↳ylab("Y"))
```

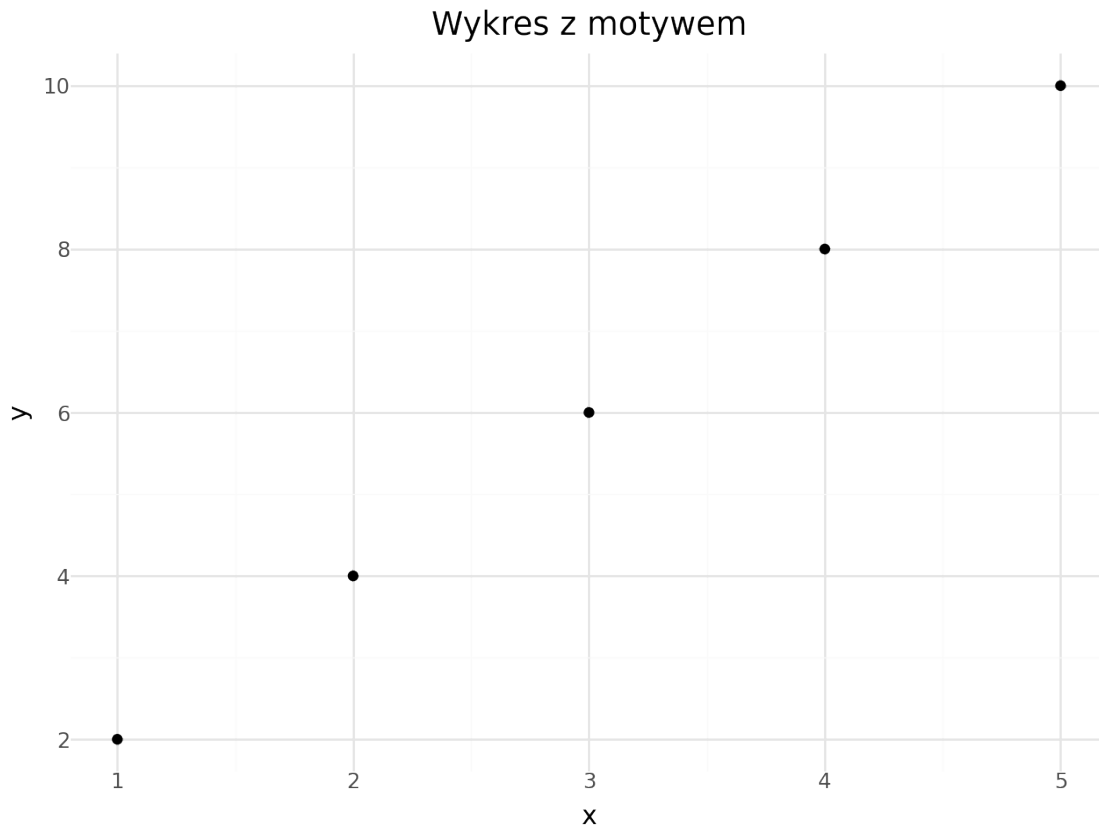


```
[109]: #przykładowe dane
data_box = pd.DataFrame({
    'kategorie': np.random.choice(['A', 'B', 'C'], 100),
    'wartości': np.random.randn(100)
})

#wykres pudłowy
(ggplot(data_box) + aes(x = 'kategorie', y = 'wartości') + geom_boxplot(fill = 'blue') + ggtitle("Wykres z pudłowy") + xlab("kategorie") + ylab("wartości"))
```



```
[110]: #dodanie motywu
(ggplot(data) + aes(x = 'x', y = 'y') + geom_point() + ggtitle("Wykres z
↳motywem") + theme_minimal())
```



```
[111]: import plotly.express as px
import pandas as pd
#przykładowe dane miast z ich współrzędnymi
data = 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]
})
#Tworzenie mapy punktowej
fig = px.scatter_mapbox(data, lat = "latitude", lon = "longitude", hover_name = "city", zoom =1, height = 500)
# Konfiguracja stylu mapy
fig.update_layout(mapbox_style = "open-street-map")
fig.update_layout( title = "Lokalizacja Wybranych Miast na Mapie")
fig.show()
```

```
[112]: import plotly.express as px
# Dane krajów z ich kodami ISO oraz populacją
data = px.data.gapminder().query("year == 2007")
# Tworzenie mapy choropleth
```

```

fig = px.choropleth(data, locations = "iso_alpha", color = "pop", hover_name = "country", color_continuous_scale = "Viridis", labels = {'pop': 'populacja'})
#Ustawienia mapy
fig.update_layout(title = "Populacja krajów na świecie w 2007 roku")
fig.show()

```

```

[138]: import plotly.express as px
#dane stanów USA z populacją
data = px.data.gapminder().query("year == 2007 & continent == 'Americas'")
data = data[data['country'] != 'Canada'] #Usunięcie Kanady, jeżeli potrzebne
#Tworzenie mapy choropleth dla stanów USA
fig = px.choropleth(data, locations="iso_alpha", color = "pop",
hover_name = "country", locationmode = "USA-states",
color_continuous_scale = "Viridis", labels = {'pop': 'Populacja'})
#Ustawienia mapy
fig.update_layout(title = "Populacja stanów USA", geo_scope = 'usa')
fig.show()

```