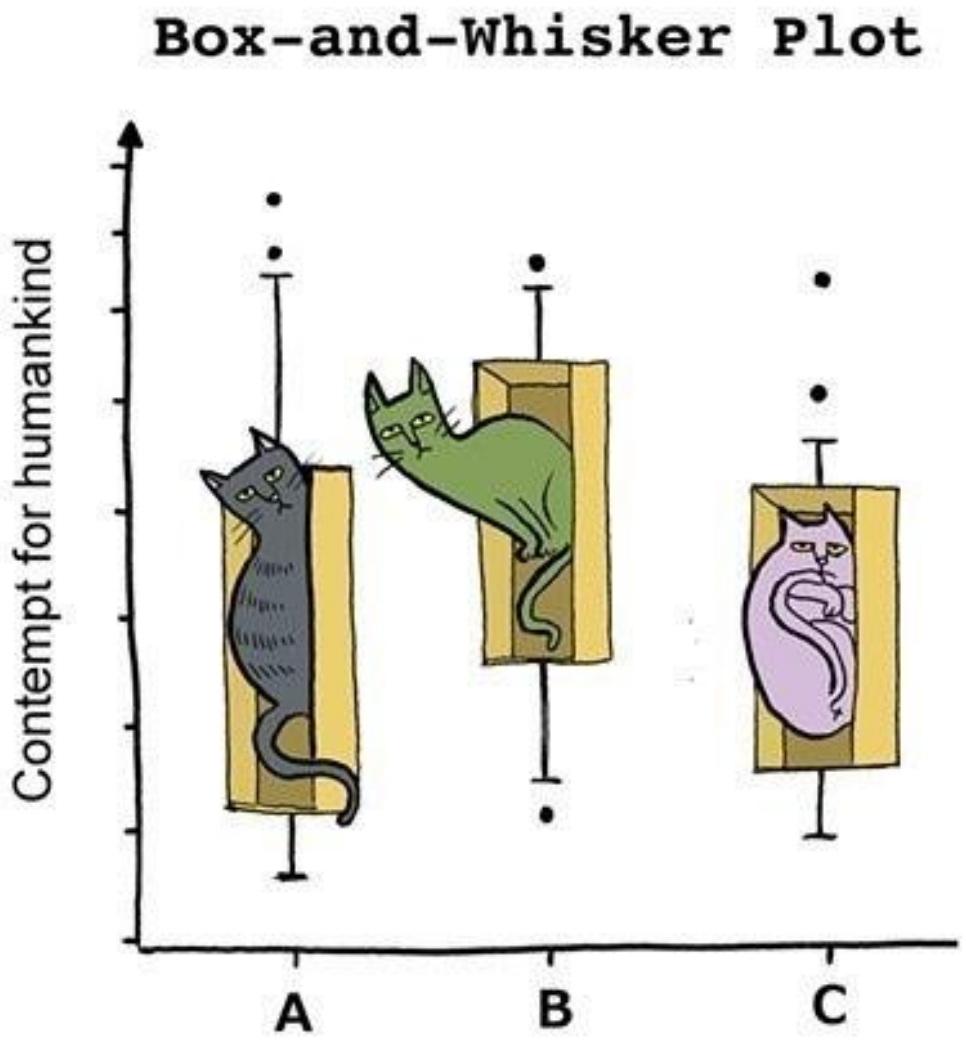
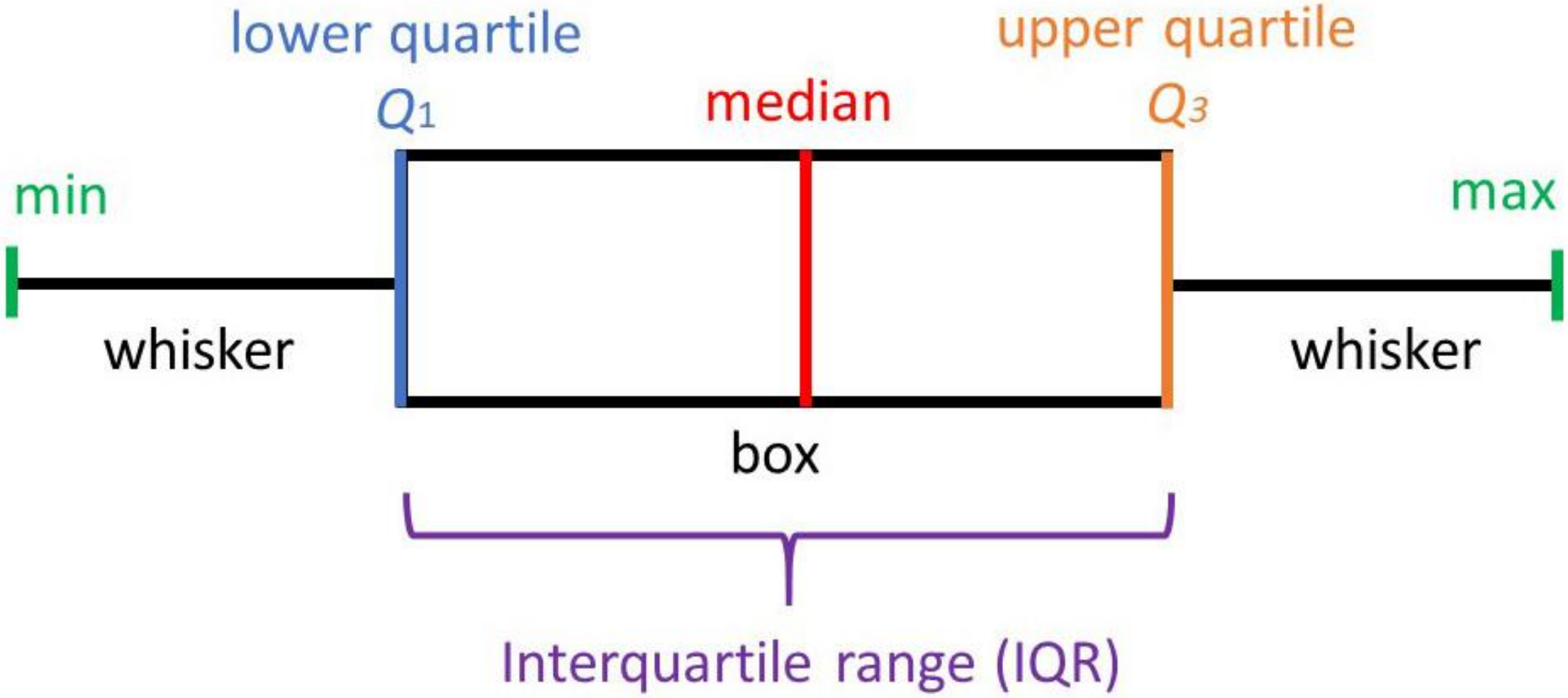
BOX PLOT

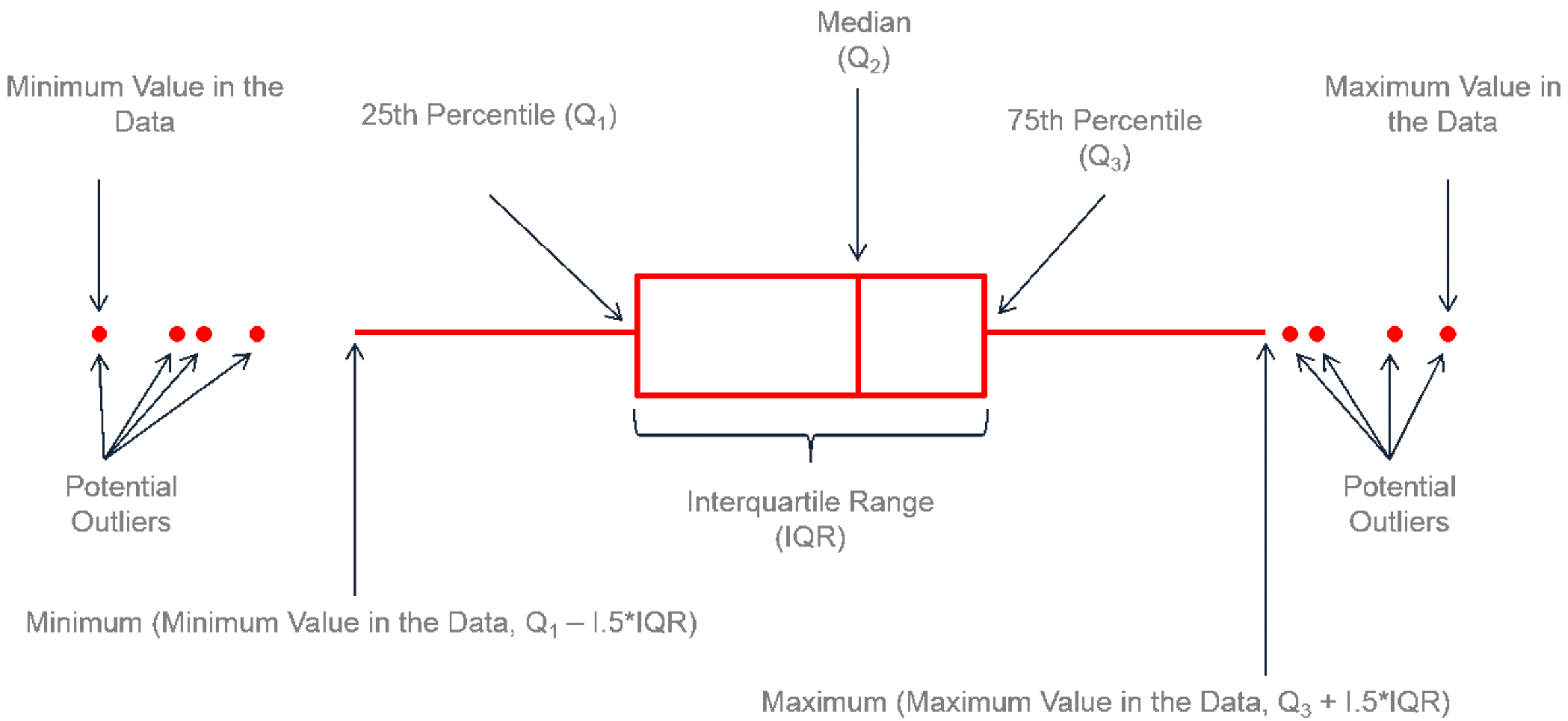
Python:

Mathplotlib

Seaborn




lower quartile
Q_1
median
upper quartile
Q_3
min
max
whisker
whisker
box
Interquartile range (IQR)


Median
(Q_2)
Minimum Value in the Data
25th Percentile (Q_1)
75th Percentile
(Q_3)
Maximum Value in the Data
Potential Outliers
Interquartile Range
(IQR)
Potential Outliers
Minimum (Minimum Value in the Data, Q_1 – 1.5*IQR)
Maximum (Maximum Value in the Data, Q_3 + 1.5*IQR)

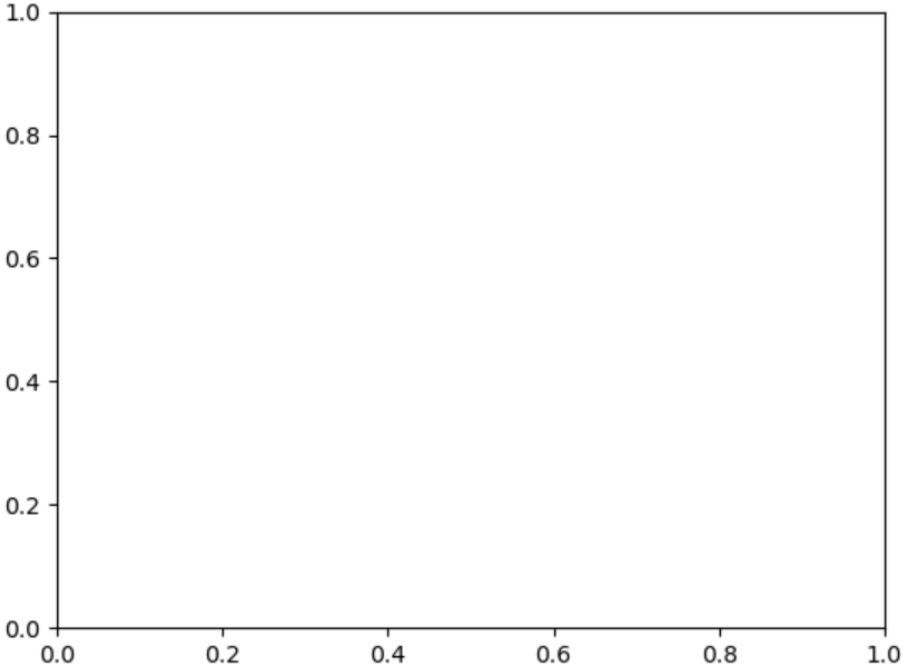
Генерим четыре выборки из нормального распределения

```
import numpy as np
np.random.seed(10)
collectn_1 = np.random.normal(100, 10, 200)
collectn_2 = np.random.normal(80, 30, 200)
collectn_3 = np.random.normal(90, 20, 200)
collectn_4 = np.random.normal(70, 25, 200)
data_to_plot = [collectn_1, collectn_2, collectn_3, collectn_4]
```

matplotlib

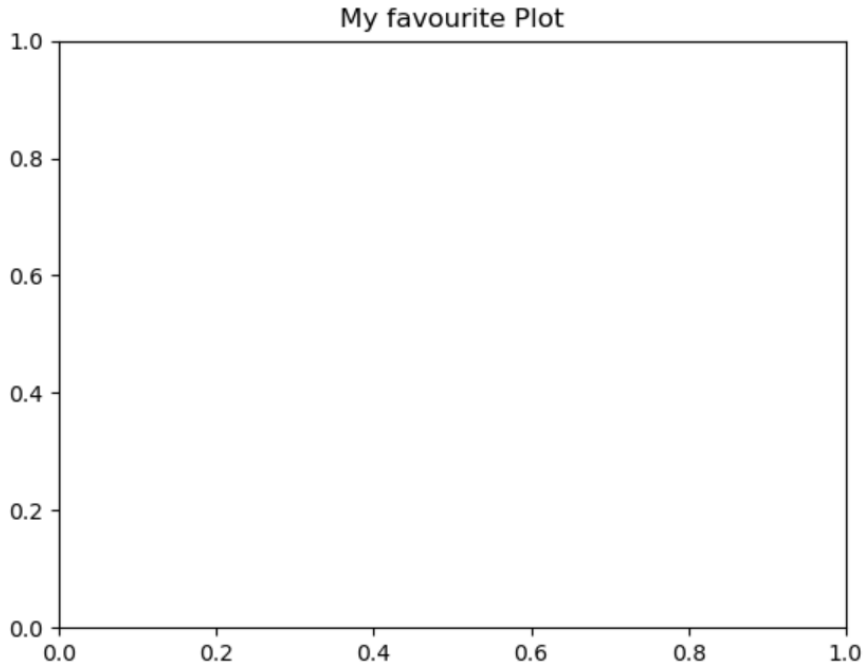
Оси

```
import matplotlib.pyplot as plt

fig1, ax1 = plt.subplots()
```



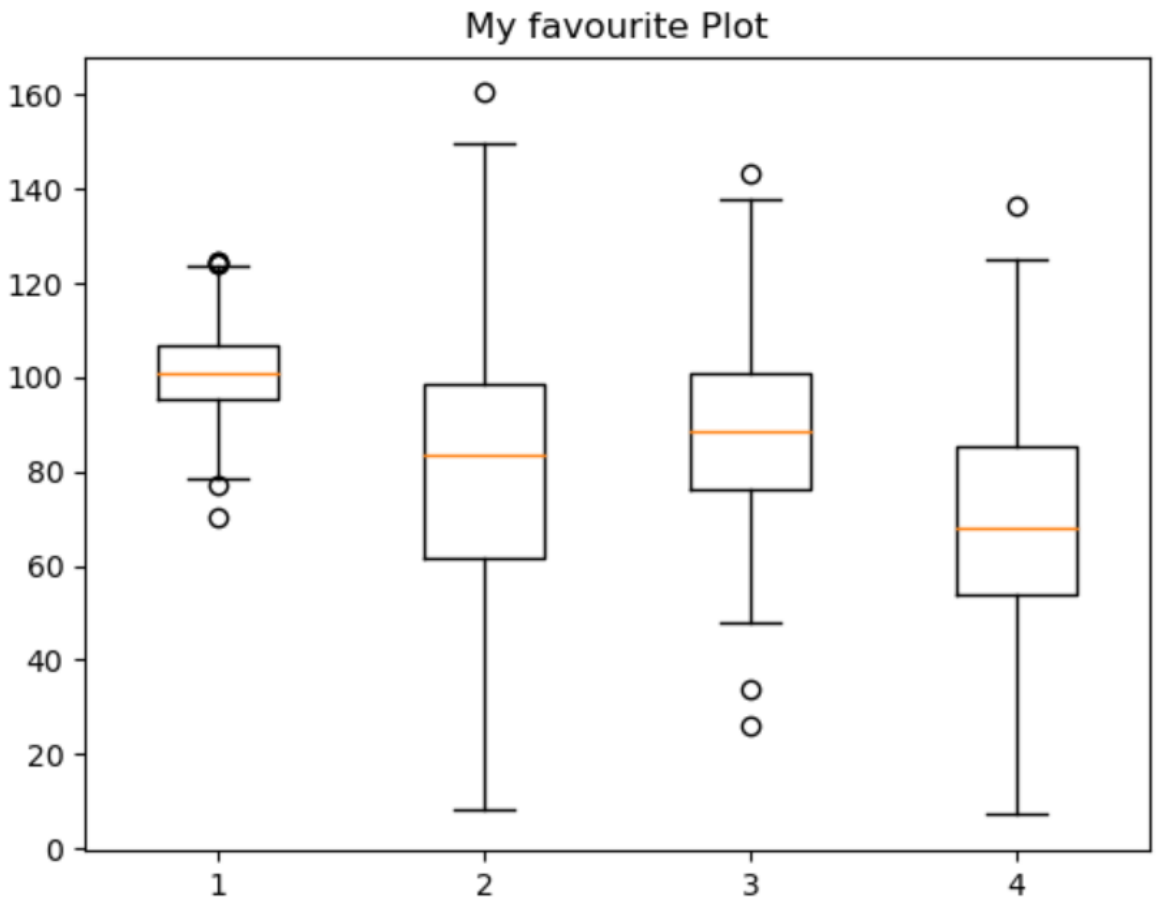
Заголовок

```
ax1.set_title('My favourite Plot')
```



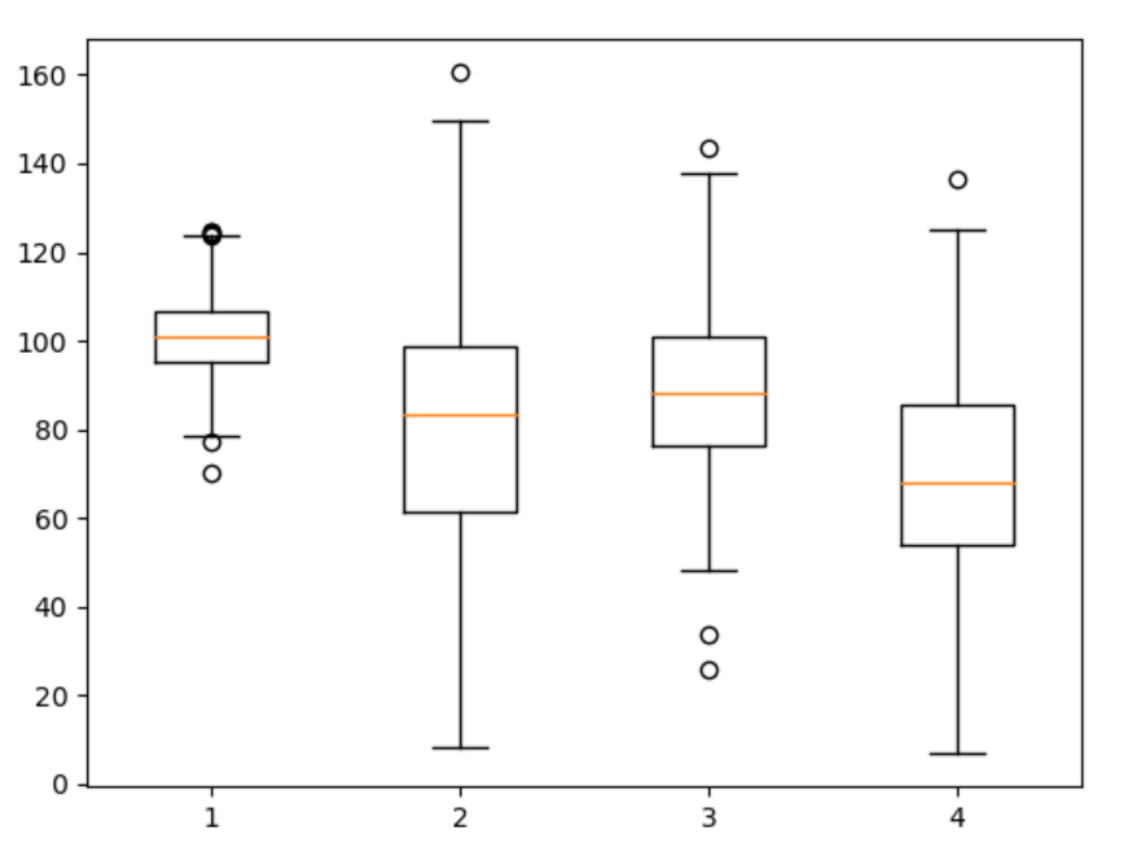
Ящик с усами

```
ax1.boxplot(data_to_plot)
```



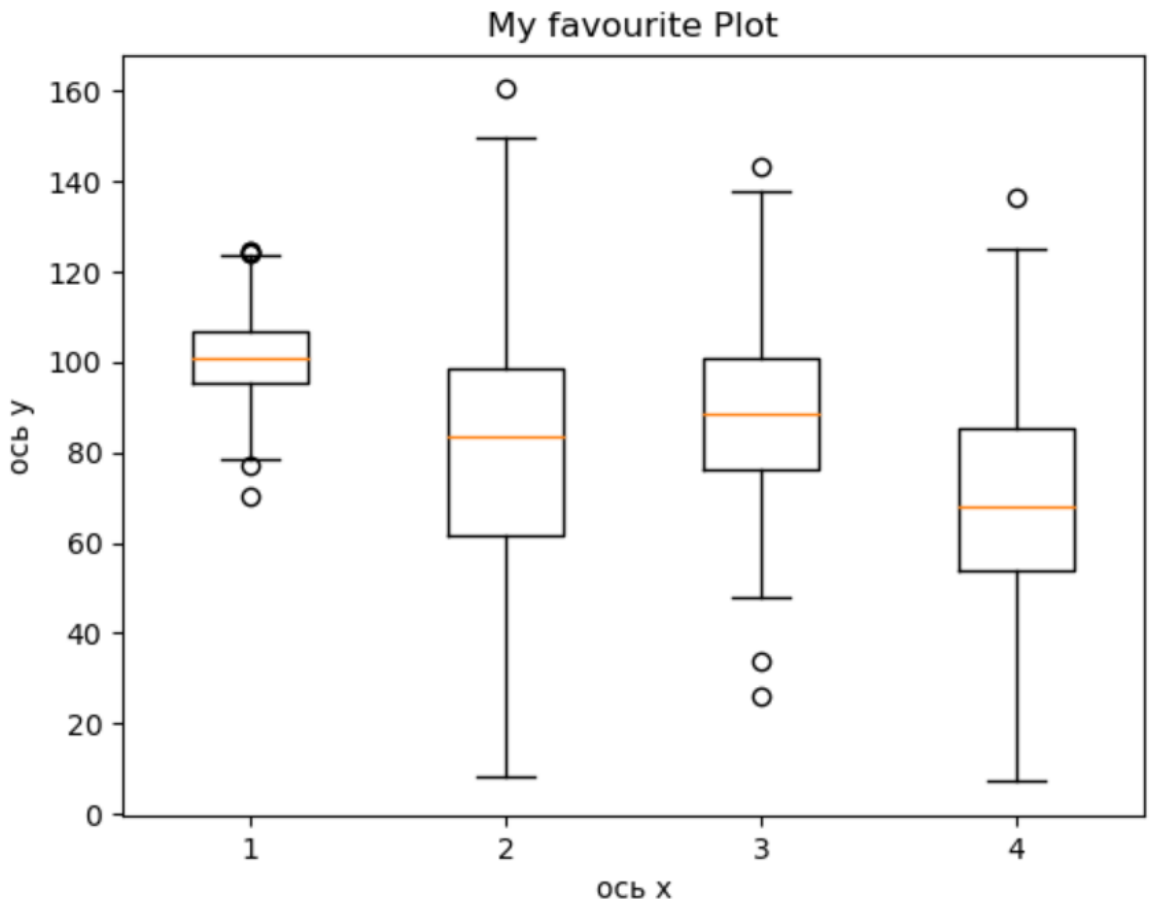
Другой вариант

```
fig = plt.figure()
# Create an axes instance
ax = fig.add_axes([0.1, 0.1, 0.8, 0.8])
# Create the boxplot
ax.boxplot(data_to_plot)
```



Подписи осей

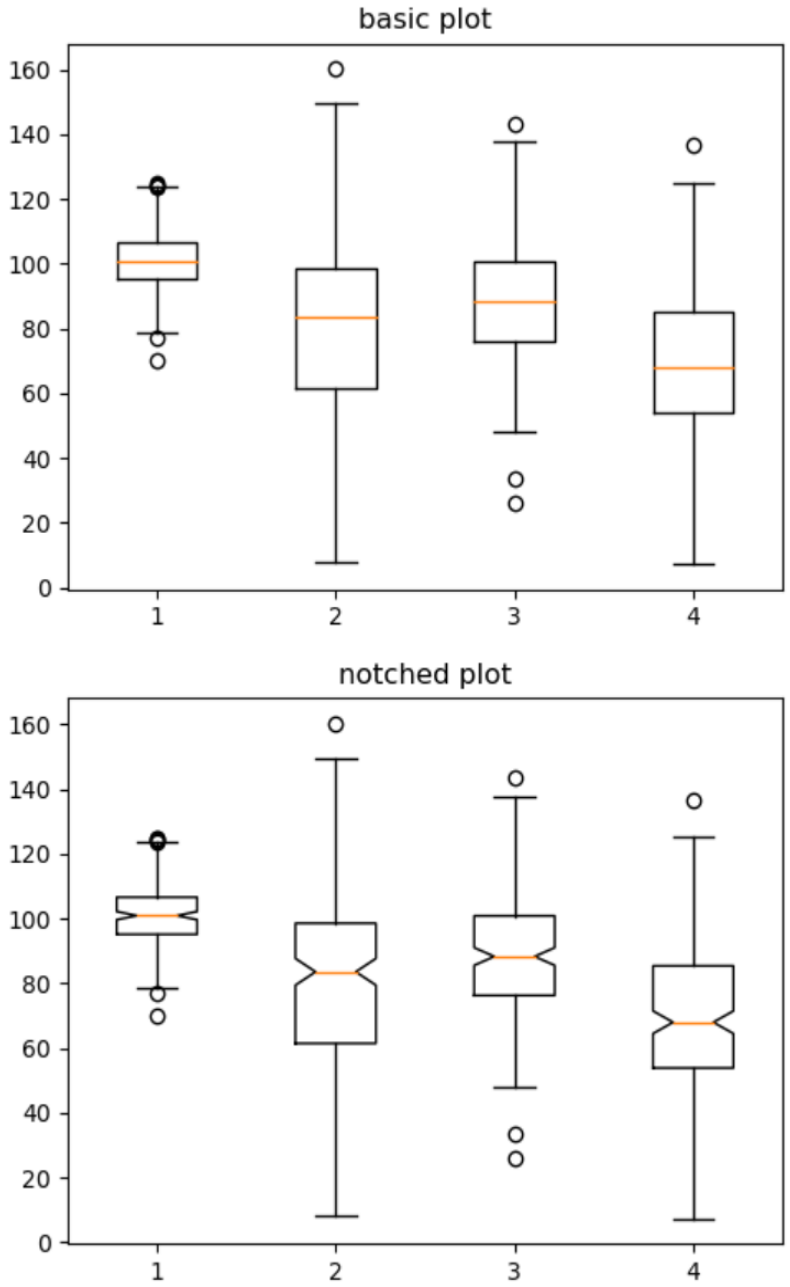
```
ax1.set_xlabel('ось x')
ax1.set_ylabel('ось y')
```



Зубчатый

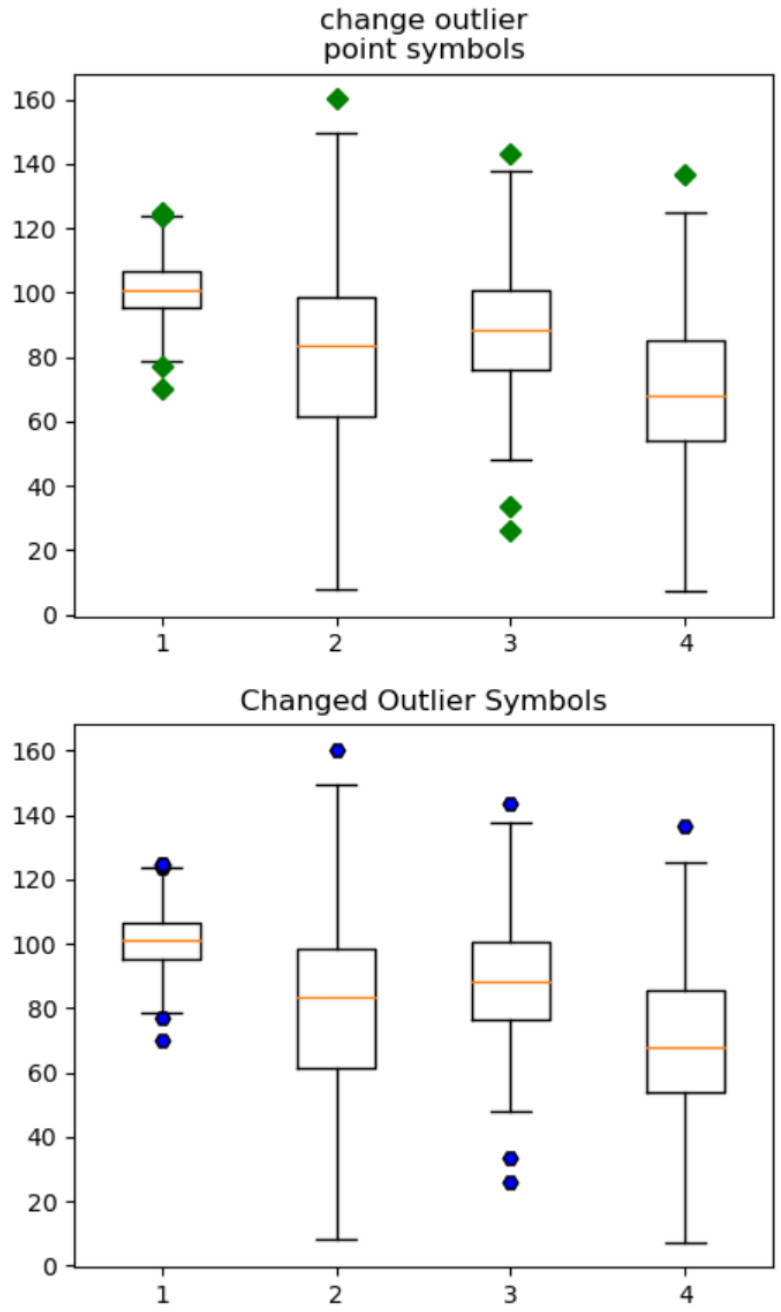
```
fig, axs = plt.subplots(2, 1)
# basic plot
axs[0].boxplot(data)
axs[0].set_title('basic plot')

# notched plot
axs[1].boxplot(data, 1)
# or
axs[1].boxplot(data, notch = True)
axs[1].set_title('notched plot')
```

confidence interval around the median



Форма и цвет выбросов

```
fig, axs = plt.subplots(2, 1)

# change outlier point symbols

axs[0].boxplot(data, 0, 'gD')
axs[0].set_title('change outlier\npoint symbols’)

blue_hexagon = dict(markerfacecolor='b', marker='H')
axs[1].set_title('Changed Outlier Symbols')
axs[1].boxplot(data, flierprops=blue_hexagon )
```



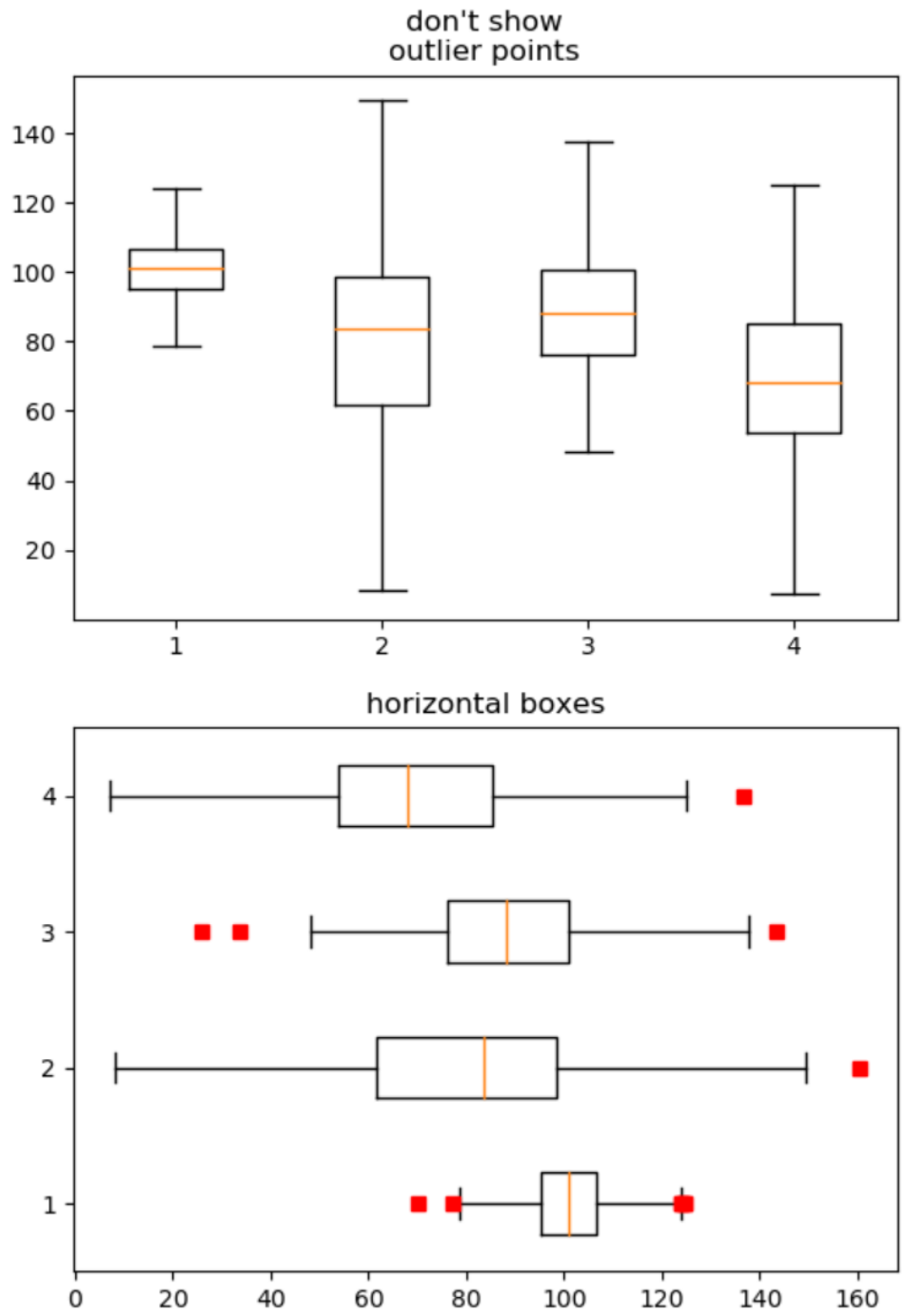
Маркеры

matplotlib.markers:
https://matplotlib.org/3.1.1/api/markers_api.html

marker	symbol	description
"."	●	point
","	·	pixel
"o"	●	circle
"v"	▼	triangle_down
"^"	▲	triangle_up
"<"	◀	triangle_left
">"	▶	triangle_right
"1"	Y	tri_down
"2"	⅄	tri_up
"3"	≺	tri_left
"4"	≻	tri_right
"8"	●	octagon
"s"	■	square
"p"	⬟	pentagon
"P"	✚	plus (filled)
"*"	★	star
"h"	⬢	hexagon1
"H"	⬢	hexagon2

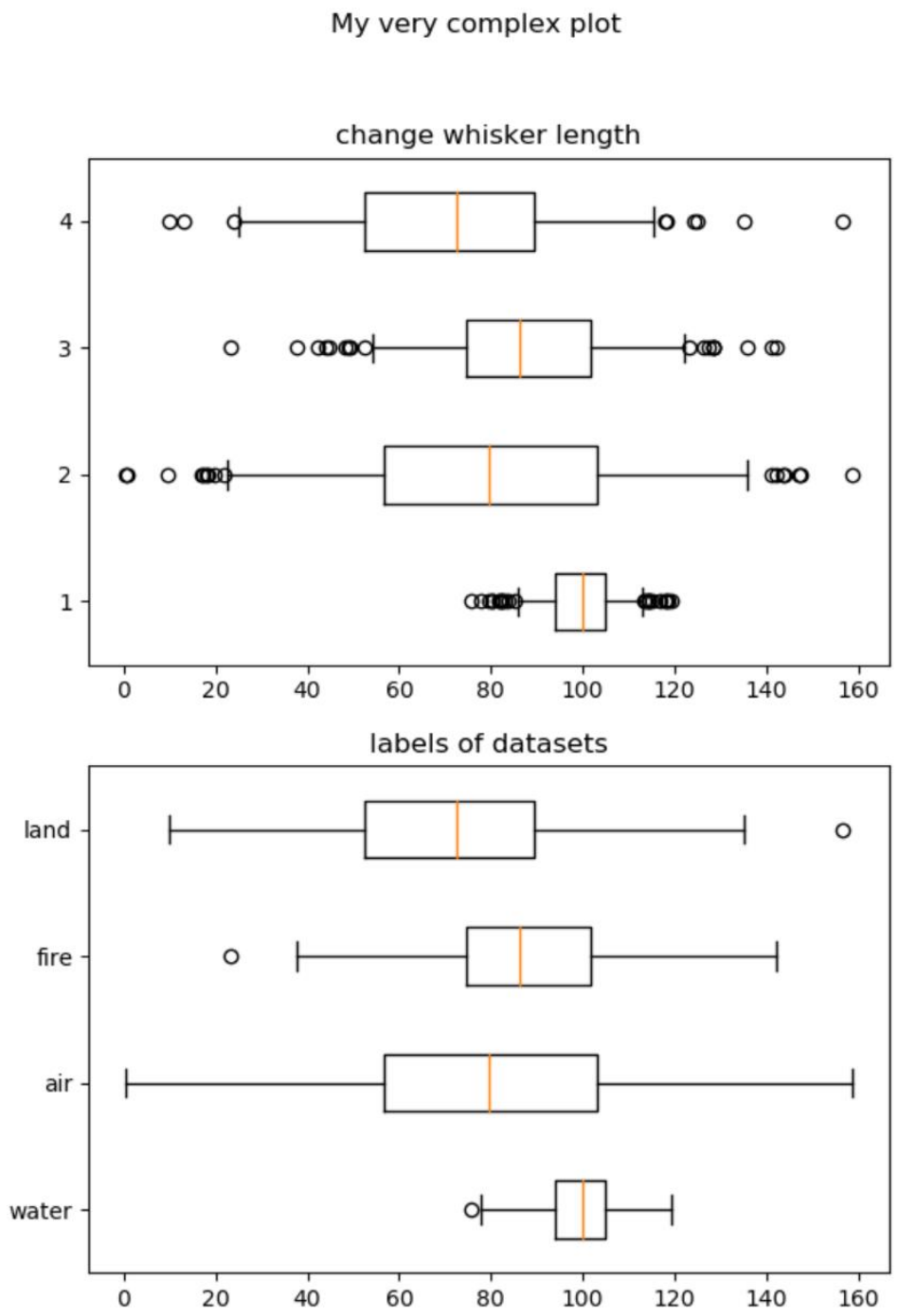
Без выбросов и перевернутый

```
fig, axs = plt.subplots(2, 1)

# don't show outlier points
axs[0].boxplot(data, 0, '')
axs[0].set_title("don't show\noutlier points")

# horizontal boxes
axs[1].boxplot(data, 0, 'rs', 0)
axs[1].set_title('horizontal boxes')
```



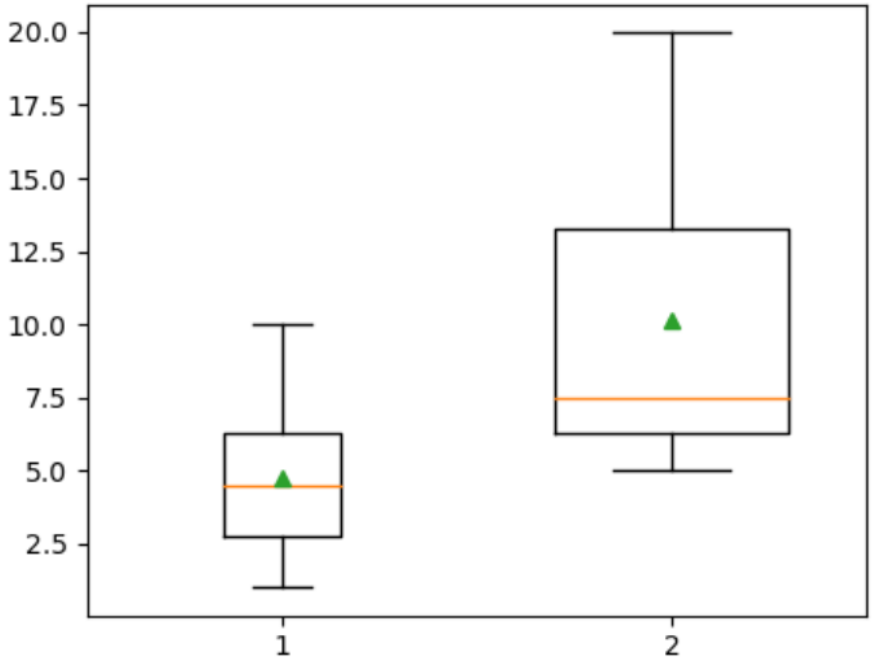
Заголовок
Длина усов
Названия боксов

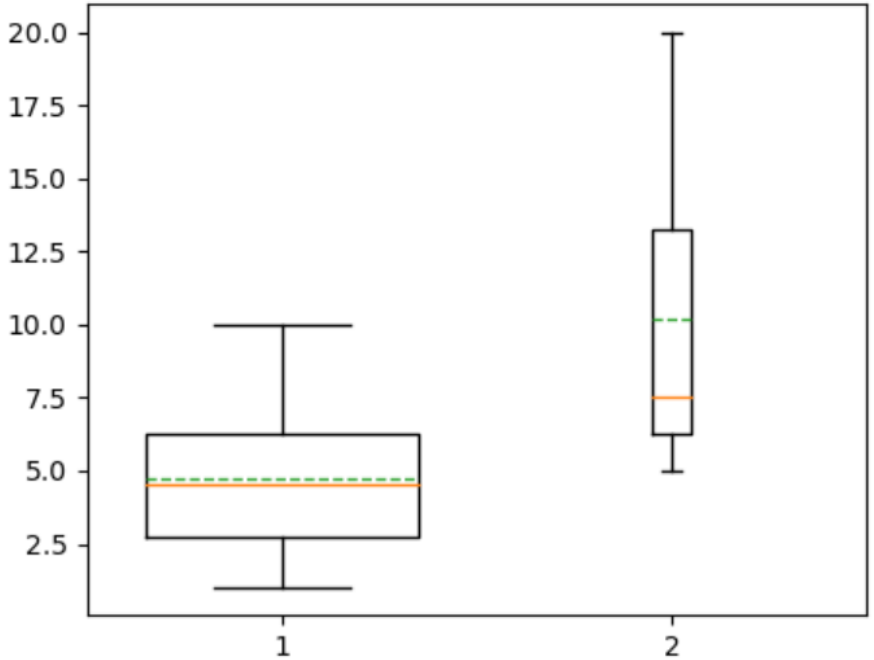
```
fig, axs = plt.subplots(2)

# Main title
fig.suptitle('My very complex plot')
# change whisker length
axs[0].boxplot(data, vert=False, whis=0.75)
axs[0].set_title('change whisker length')

# labels
axs[1].boxplot(data, vert=False, labels = (
      'water', 'air', 'fire', 'land'))
axs[1].set_title('labels of datasets')
```



Два способа задать среднюю
Ширина боксов

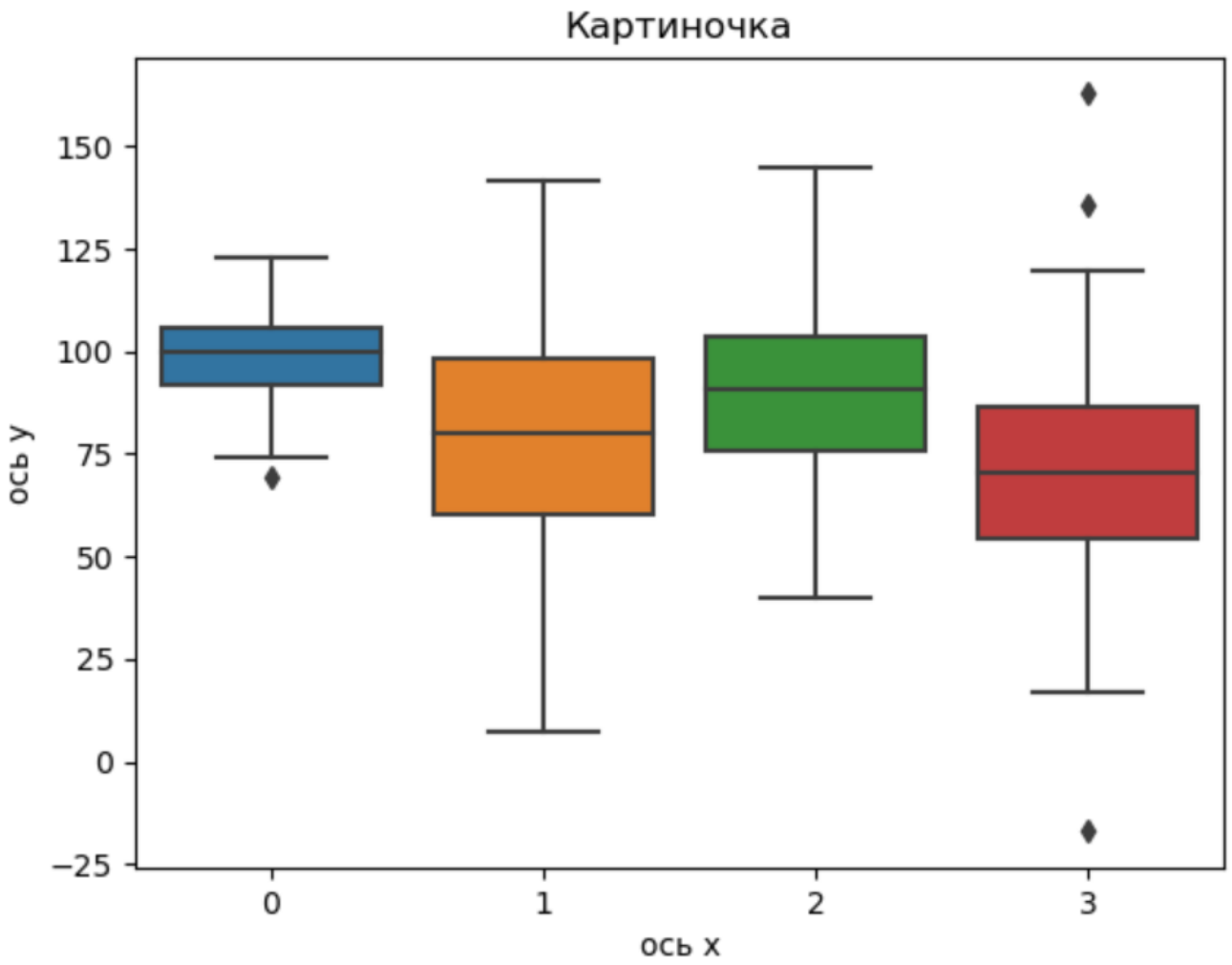
```
data1 = [1, 2, 3, 4, 5, 6, 7, 10]
data2 = [5, 6, 7, 8, 15, 20]
data = [data1, data2]

fig, axs = plt.subplots(2)

axs[0].boxplot(data, showmeans=True,
               widths=[0.3, 0.6])
axs[1].boxplot(data, meanline = True,
               showmeans=True, widths=[0.7,0.1])
```





Seaborn

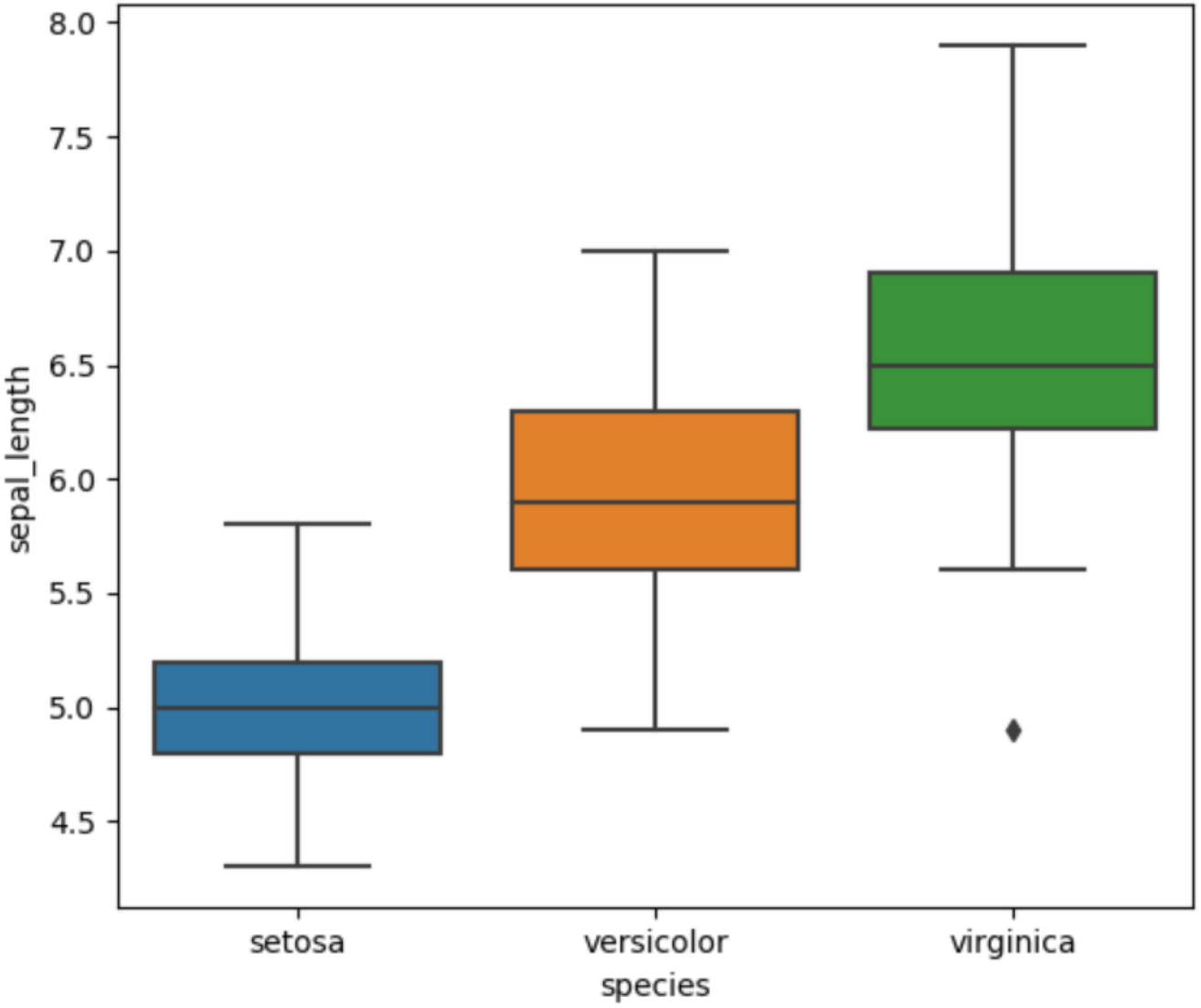
```python
import seaborn as sns

sns.boxplot(data=data_to_plot)

graf.set(xlabel='ось x’,
      ylabel='ось y’,
      title="Картиночка")
```



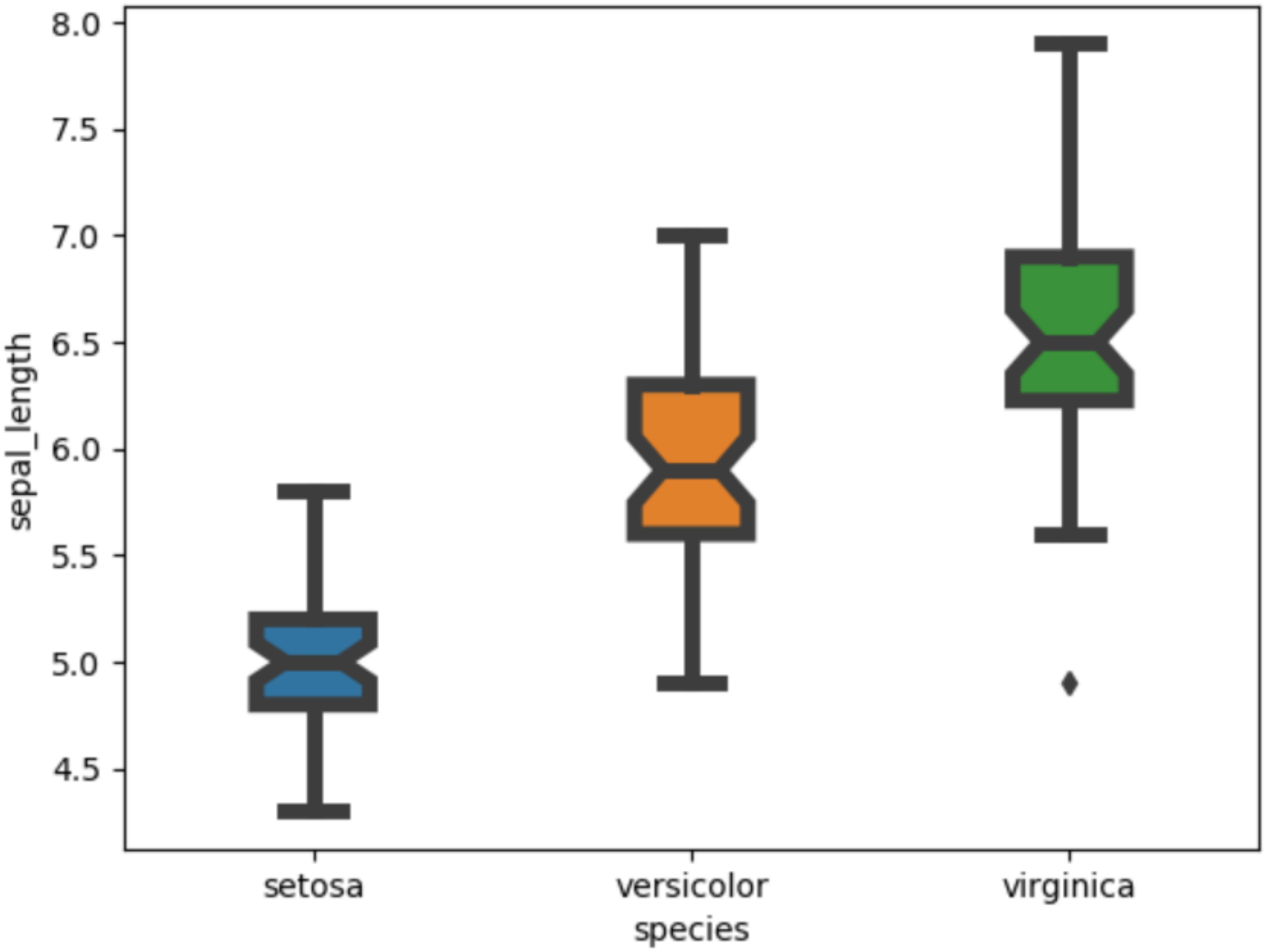
```python
df = sns.load_dataset('iris')
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
..	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

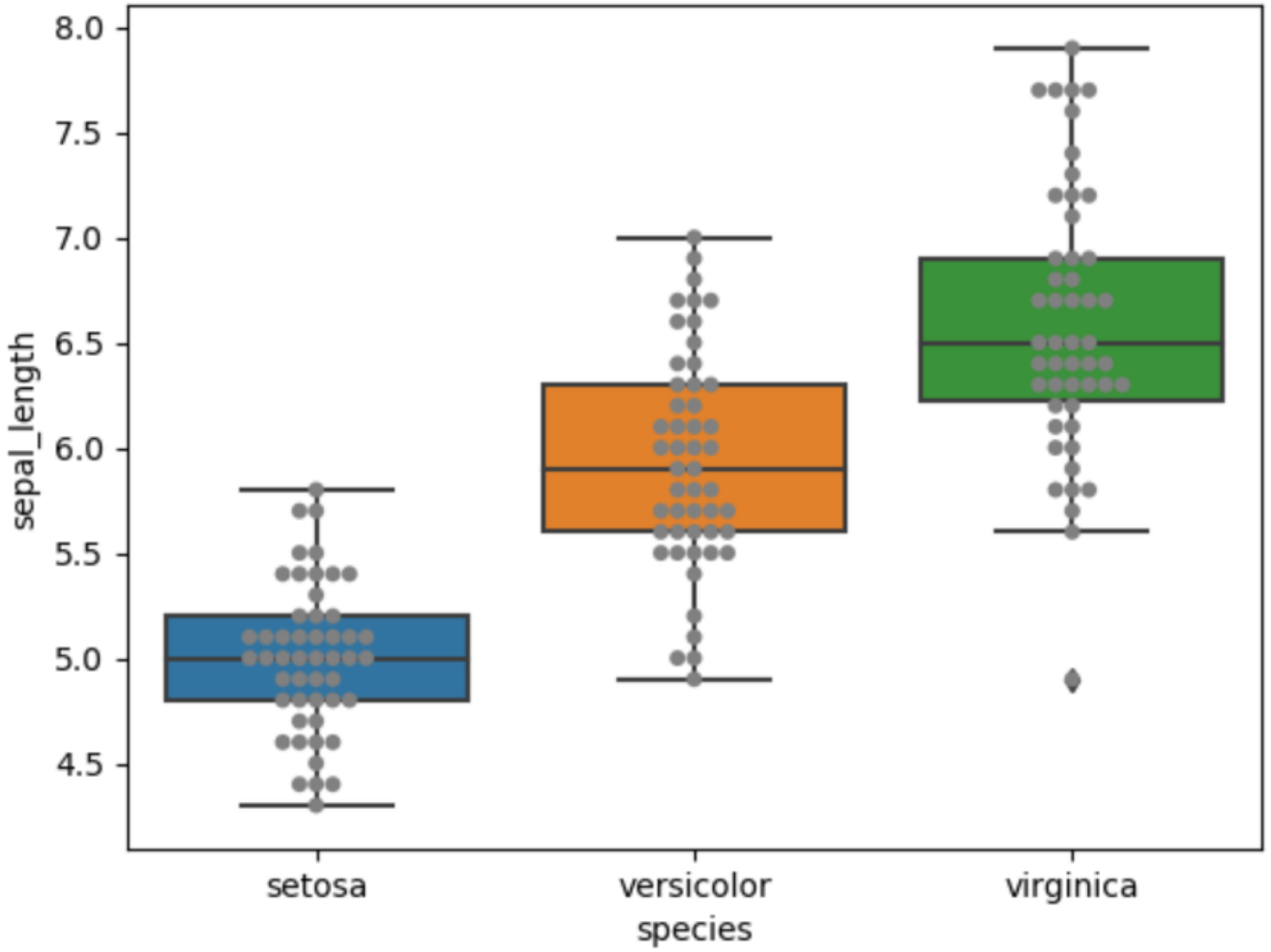
```
sns.boxplot(x=df["species"],
            y=df["sepal_length"])
```



```
sns.boxplot( x=df["species"],
        y=df["sepal_length"],
        linewidth=5,
        notch=True,
        width=0.3)
```



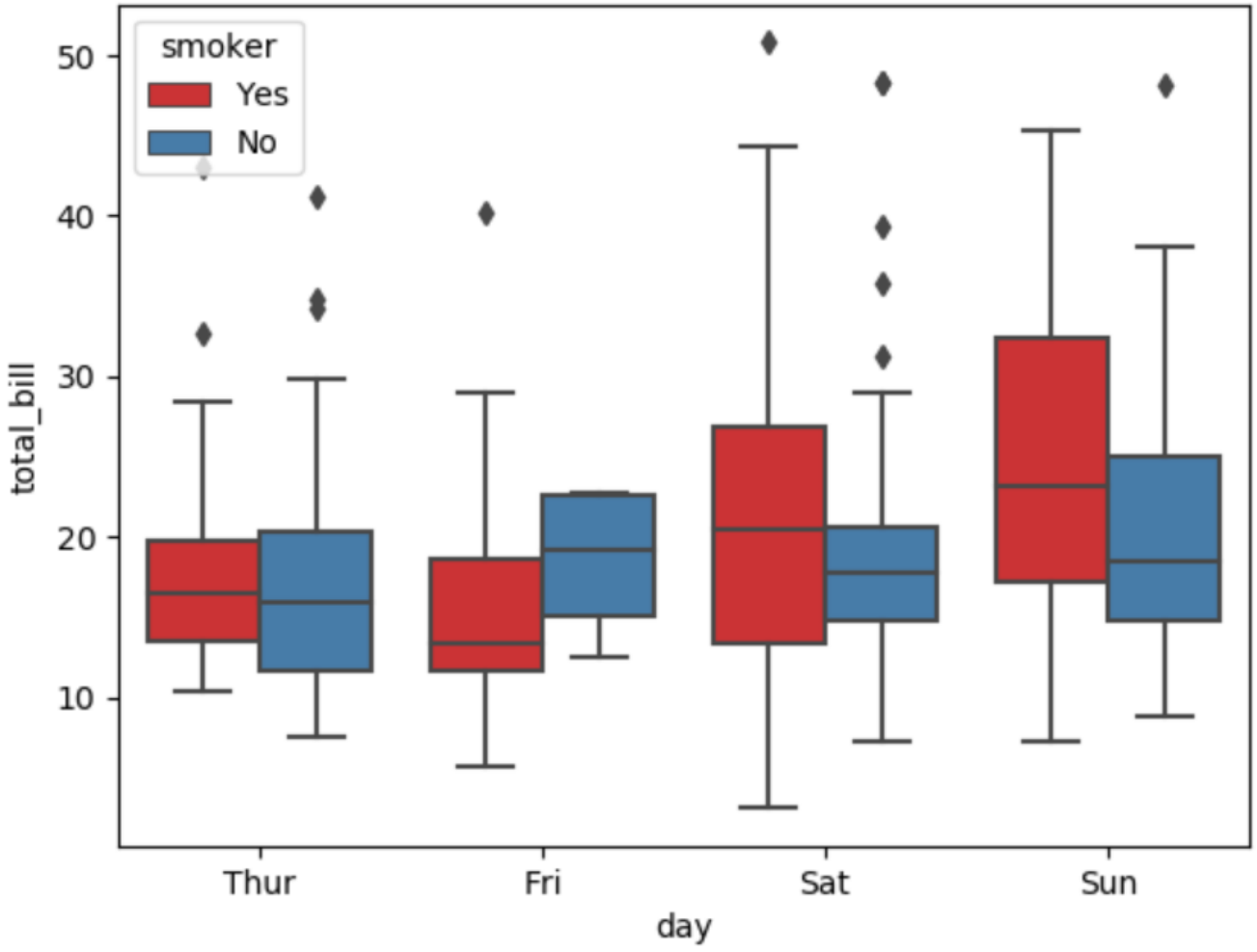
```python
ax = sns.boxplot(x='species', y='sepal_length', data=df)
# Add jitter with the swarmplot function.
ax = sns.swarmplot(x='species', y='sepal_length', data=df, color="grey")
```

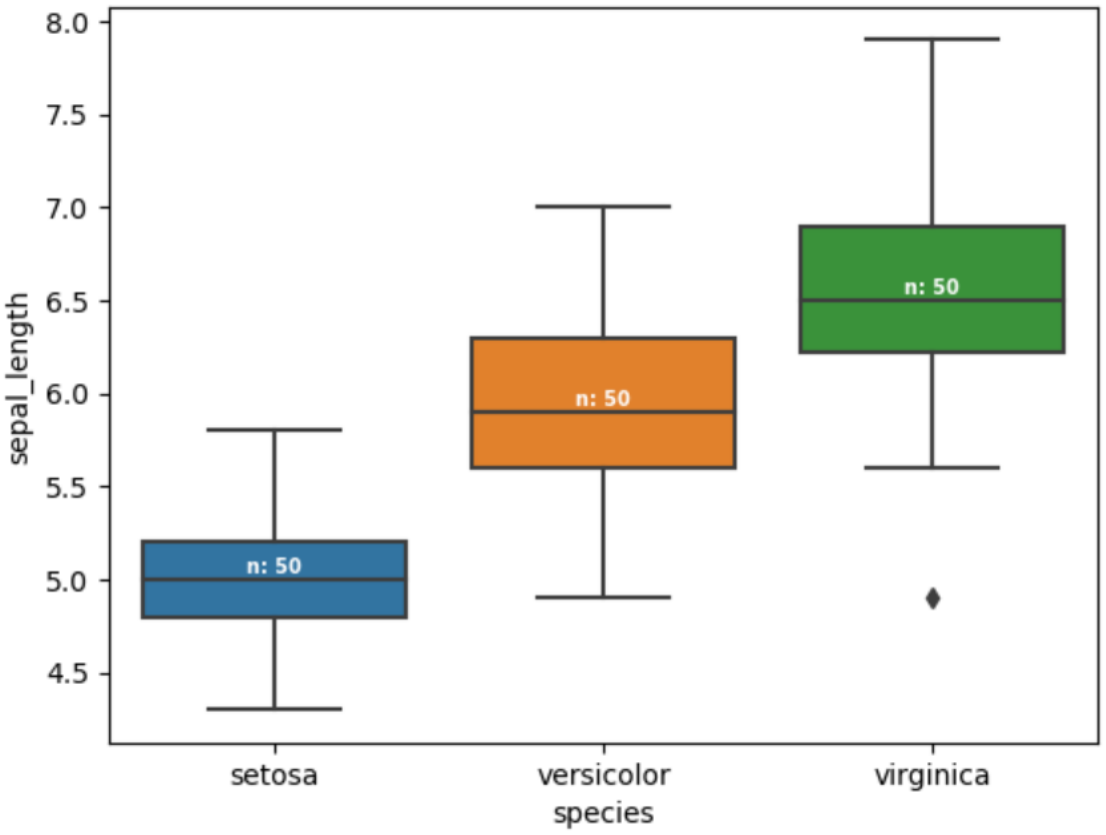


```python
df = sns.load_dataset('tips')
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
..	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

```
sns.boxplot(x="day",
      y="total_bill",
      hue="smoker",
data=df, palette="Set1")
```






```
df = sns.load_dataset("iris")

ax = sns.boxplot(x="species", y="sepal_length", data=df)

# Calculate number of obs per group & median to position labels
medians = df.groupby(['species'])['sepal_length'].median().values
nobs = df['species'].value_counts().values
nobs = [str(x) for x in nobs.tolist()]
nobs = ["n: " + i for i in nobs]

# Add it to the plot
pos = range(len(nobs))
for tick, label in zip(pos, ax.get_xticklabels()):
    ax.text(pos[tick], medians[tick] + 0.03, nobs[tick],
            horizontalalignment='center', size='x-small', color='w', weight='semibold')
```

```
sns.violinplot(x='species’,
        y='sepal_length’,
        data=df,
order=[ "versicolor", "virginica", "setosa"])
```