

✓ Continuous Variables - Boxplot

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
from google.colab import drive
import os
```

```
drive.mount('/content/drive')
os.chdir('/content/drive/MyDrive/')
for item in os.listdir():
    print(item)
print("-----")
os.chdir('/content/drive/MyDrive/cloud/GitHub/AdvDataViz/Notebooks/')
for item in os.listdir():
    print(item)
print("-----")
notebooks = "/content/drive/MyDrive/cloud/GitHub/AdvDataViz/Notebooks"
print(os.listdir(notebooks))
print("-----")
```

```
file = "heart-disease.csv"
file_path = os.path.join(notebooks, file)
with open(file_path, "r") as f:
    contents = f.read()
```

```
📁 Mounted at /content/drive
learningStore
healthyCar
startup
cloud
Artificial Intelligence
```

```
-----
03 Matplotlib - Exercise.ipynb
02 Matplotlib.ipynb
01 Python_Pandas.ipynb
04 Continuous Variables - Histogram .ipynb
05 Continuous Variables - Histogram - Exercise .ipynb
07 Continuous Variables - Boxplot - Exercise .ipynb
03 Matplotlib - Exercise Solutions.ipynb
05 Continuous Variables - Histogram - Exercise Solutions.ipynb
06 Continuous Variables - Boxplot.ipynb
08 Continuous Variables - Scatterplot.ipynb
07 Continuous Variables - Boxplot - Exercise Solutions.ipynb
09 Continuous Variables - Scatterplot - Exercise Solutions.ipynb
09 Continuous Variables - Scatterplot - Exercise .ipynb
10 Categorical Variables - Bar_Pie.ipynb
12 Seaborn.ipynb
11 Pandas Data Visualization.ipynb
13 Seaborn - Exercise .ipynb
Top 50 US Tech Companies.csv
13 Seaborn - Exercise Solution.ipynb
15 Custom Modules.ipynb
14 Functions.ipynb
churn.csv
student_performance.csv
myplotlib.py
```

employee_attrition_.csv
heart-disease.csv

['03 Matplotlib - Exercise.ipynb', '02 Matplotlib.ipynb', '01 Python_Pandas.ipynb', '04 Cont

Dataset: Heart Disease

```
#df = pd.read_csv("heart-disease.csv")
df = pd.read_csv(file_path)
```

```
df.head()
```

	age	sex	chest_pain	rest_bp	chol	max_hr	st_depr	heart_disease	
0	63	female	3	145	233	150	2.3	1	
1	37	female	2	130	250	187	3.5	1	
2	41	male	1	130	204	172	1.4	1	
3	56	female	1	120	236	178	0.8	1	
4	57	male	0	120	354	163	0.6	1	

Next steps:

[Generate code with df](#)
[View recommended plots](#)
[New interactive sheet](#)

Boxplot

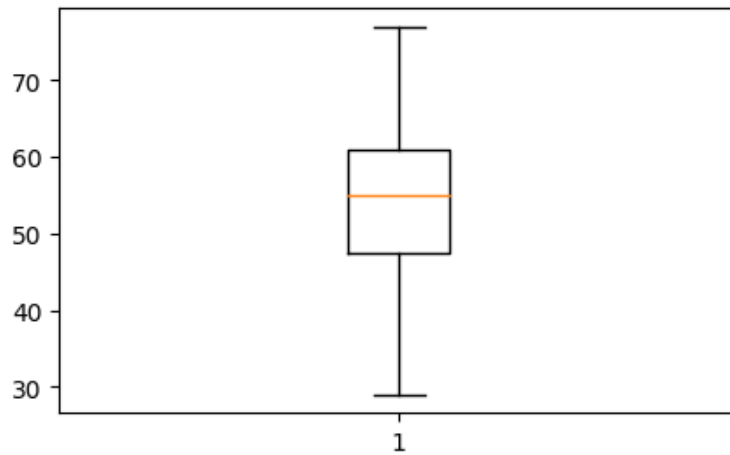
```
fig, ax = plt.subplots(figsize = (5, 3))
```

```
ax.boxplot(df["age"])
```

```

{
  'whiskers': [<matplotlib.lines.Line2D at 0x7c1d69adeaa0>,
               <matplotlib.lines.Line2D at 0x7c1d69aded40>],
  'caps': [<matplotlib.lines.Line2D at 0x7c1d69adefe0>,
           <matplotlib.lines.Line2D at 0x7c1d69adf280>],
  'boxes': [<matplotlib.lines.Line2D at 0x7c1d69ade800>],
  'medians': [<matplotlib.lines.Line2D at 0x7c1d69adf520>],
  'fliers': [<matplotlib.lines.Line2D at 0x7c1d69adf7c0>],
  'means': []}

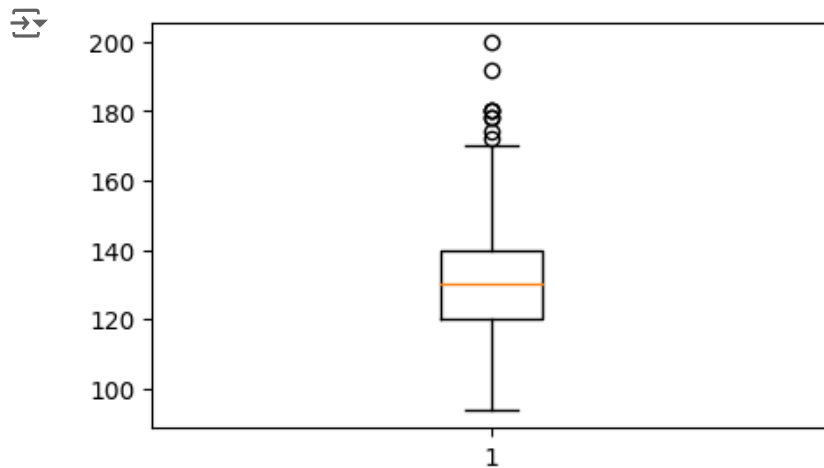
```



✓ Outliers

```
fig, ax = plt.subplots(figsize = (5, 3))
```

```
ax.boxplot(df["rest_bp"]);
```



✓ Using Ages for Data

```

female_ages = df.loc[df["sex"] == "female", "age"]
male_ages = df.loc[df["sex"] == "male", "age"]

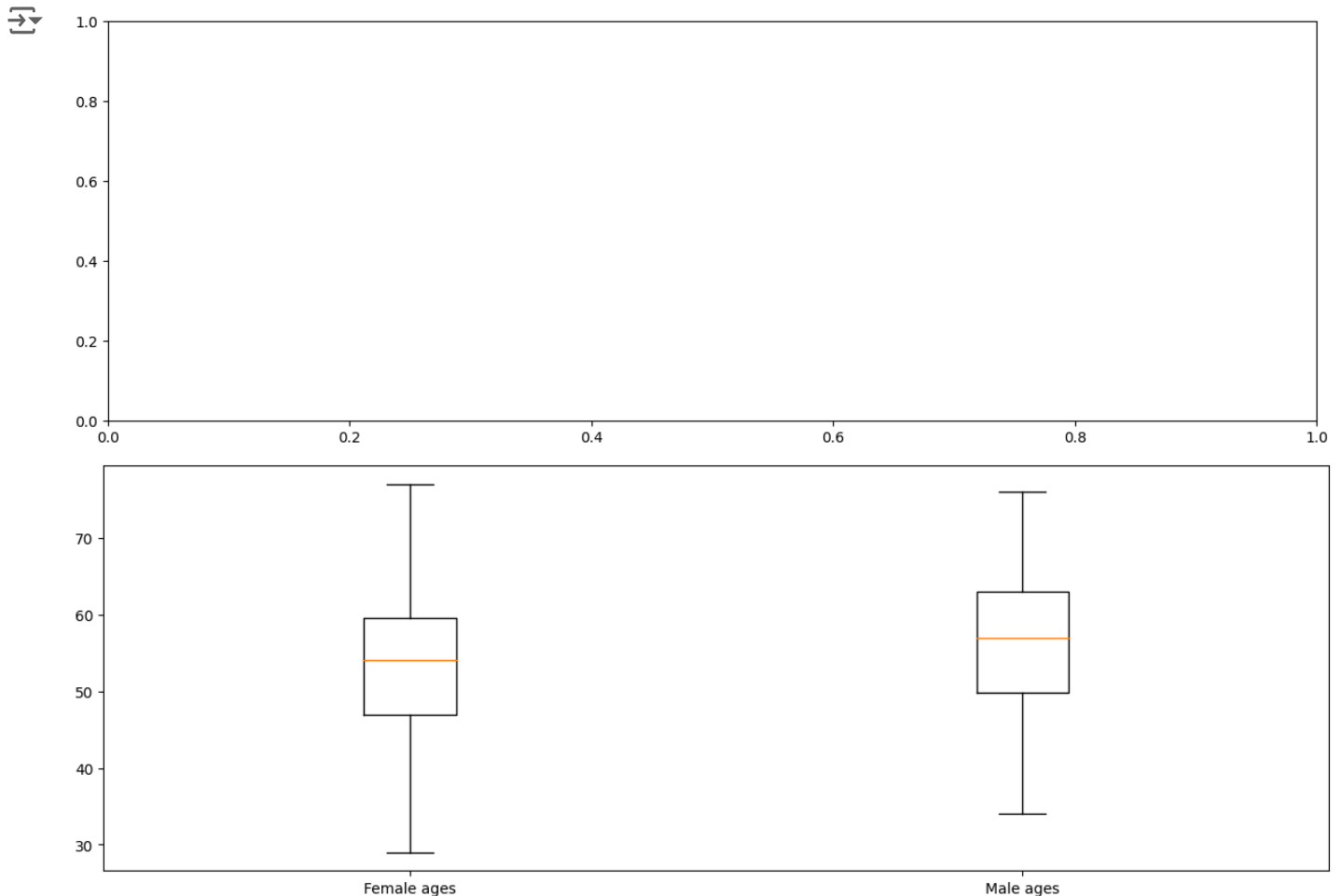
```

✓ Joint: continuous x categorical

Create paired boxplots

```
fig, ax = plt.subplots(figsize = (15, 5))
#                                     # used to label x-ticks for each boxplot
#ax.boxplot([female_ages, male_ages], tick_labels = ["Female ages", "Male ages"]);

fig, ax = plt.subplots(figsize=(15, 5))
ax.boxplot([female_ages, male_ages])
ax.set_xticklabels(["Female ages", "Male ages"])
plt.show()
```

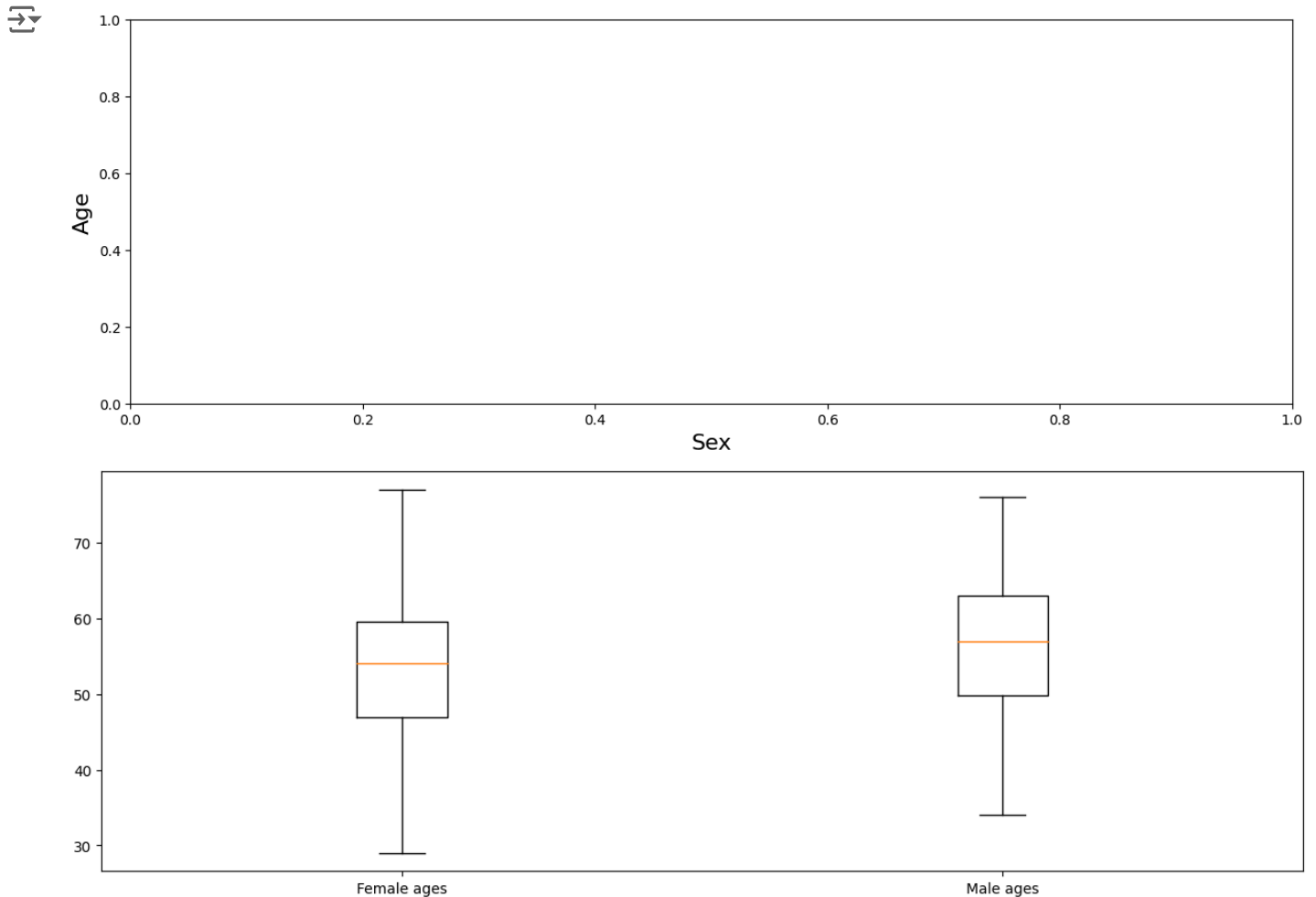


✓ Add axis labels

```
fig, ax = plt.subplots(figsize = (15, 5))
#
#                                     # labels for ticks
#ax.boxplot([female_ages, male_ages], tick_labels = ["Female Ages", "Male Ages"])

ax.set_xlabel('Sex', fontsize=16)
ax.set_ylabel('Age', fontsize=16);
```

```
fig, ax = plt.subplots(figsize=(15, 5))
ax.boxplot([female_ages, male_ages])
ax.set_xticklabels(["Female ages", "Male ages"])
plt.show()
```



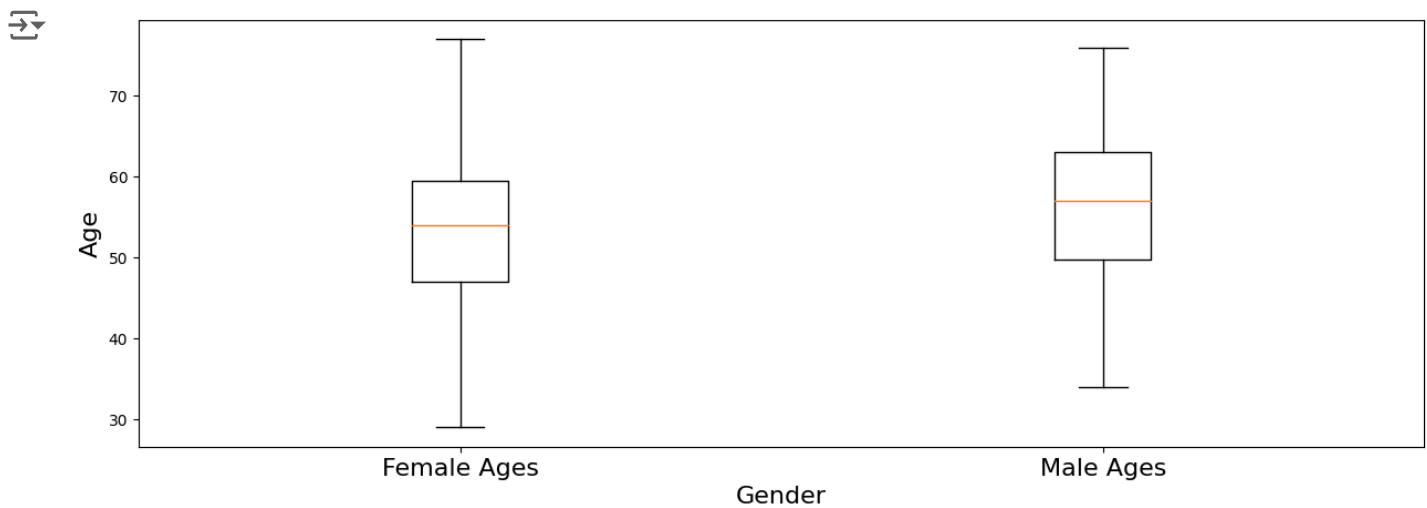
✓ Set tick labels for each of the boxplots and modify the fontsize

```
fig, ax = plt.subplots(figsize = (15, 5))

ax.boxplot([female_ages, male_ages])

ax.set_xlabel('Gender', fontsize=16)
ax.set_ylabel('Age', fontsize=16);

# more flexibility for styling tick labels
ax.set_xticklabels(["Female Ages", "Male Ages"], fontsize=16);
```



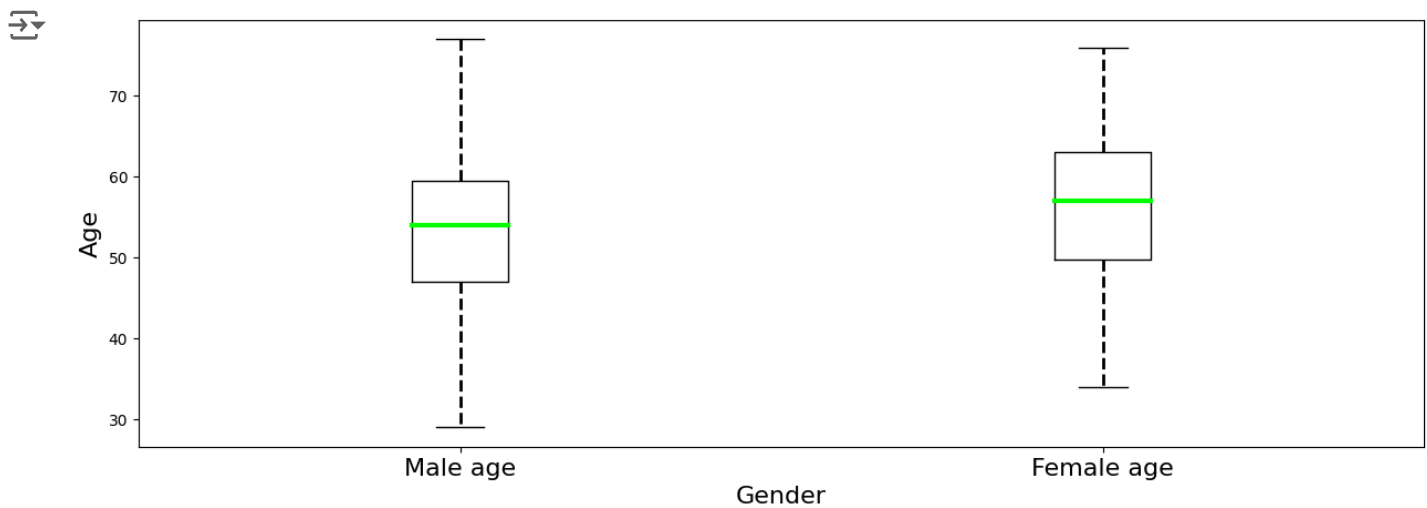
▼ Style the whiskers and median

```
fig, ax = plt.subplots(figsize = (15, 5))

# style the whiskers and median
ax.boxplot([female_ages, male_ages],
           whiskerprops = {"linestyle":"dashed", "linewidth":2.0},
           medianprops = {"color":"lime", "linewidth":3})

ax.set_xlabel('Gender', fontsize=16)
ax.set_ylabel('Age', fontsize=16);

ax.set_xticklabels(["Male age", "Female age"], fontsize=16);
```



▼ Fill boxes with color

```
fig, ax = plt.subplots(figsize = (15, 5))

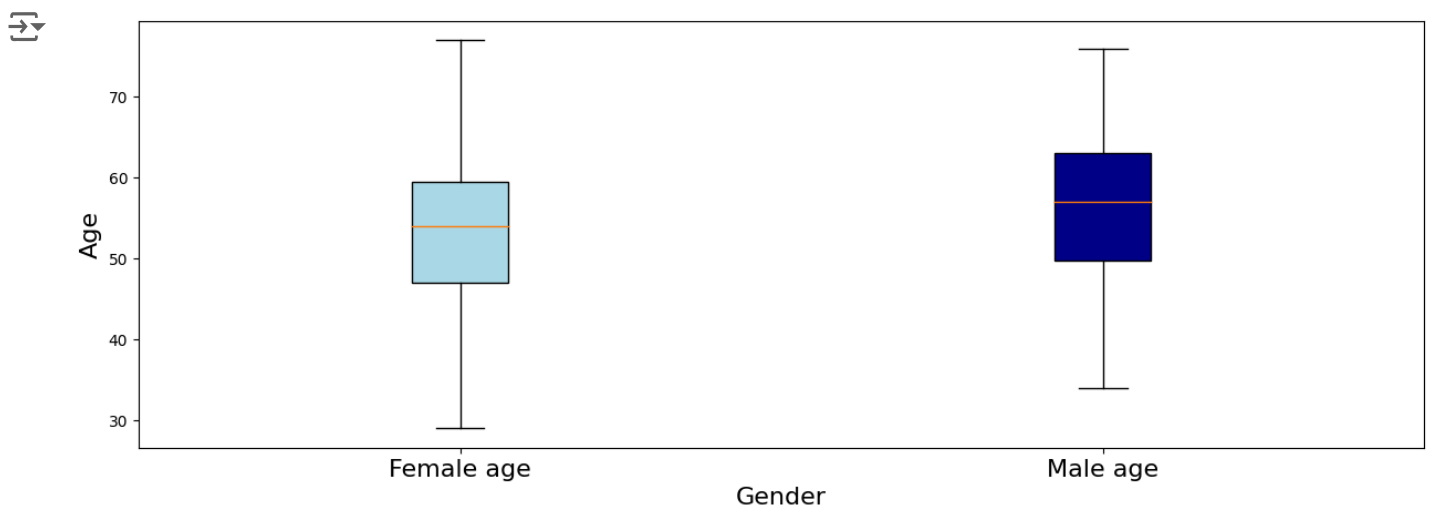
# Set patch_artist to True to color the boxes
bplot = ax.boxplot([female_ages, male_ages], patch_artist=True)

ax.set_xlabel('Gender', fontsize=16)
ax.set_ylabel('Age', fontsize=16)

ax.set_xticklabels(["Female age", "Male age"], fontsize=16)

# colors to use
colors = ['lightblue', 'darkblue']

# The patch is the box for the boxplot
# The patch/box is zipped up with its color as a tuple to be unpacked
for patch, color in zip(bplot['boxes'], colors):
    patch.set_facecolor(color);
```



✓ Add horizontal grid lines

```
fig, ax = plt.subplots(figsize = (15, 5))

bplot = ax.boxplot([male_ages, female_ages], patch_artist=True)

ax.set_xlabel('Gender', fontsize=16)
ax.set_ylabel('Age', fontsize=16)

ax.set_xticklabels(["Male age", "Female age"], fontsize=16)

colors = ['darkblue', 'lightblue']

for patch, color in zip(bplot['boxes'], colors):
    patch.set_facecolor(color);

# add horizontal grid lines
ax.yaxis.grid(True)
```

