Continuous Variables - Histogram - Exercise

```
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 = "churn.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
Artificial Intelligence
cloud
--------------
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'
--------------
```

Dataset: Customer Churn

```
#df = pd.read_csv("churn.csv")
df = pd.read_csv(file_path)
df.head()
```

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMembe
0	619	1	42	2	0.00	1	1	
1	608	1	41	1	83807.86	1	0	
2	502	1	42	8	159660.80	3	1	
3	699	1	39	1	0.00	2	0	
4	850	1	43	2	125510.82	1	1	

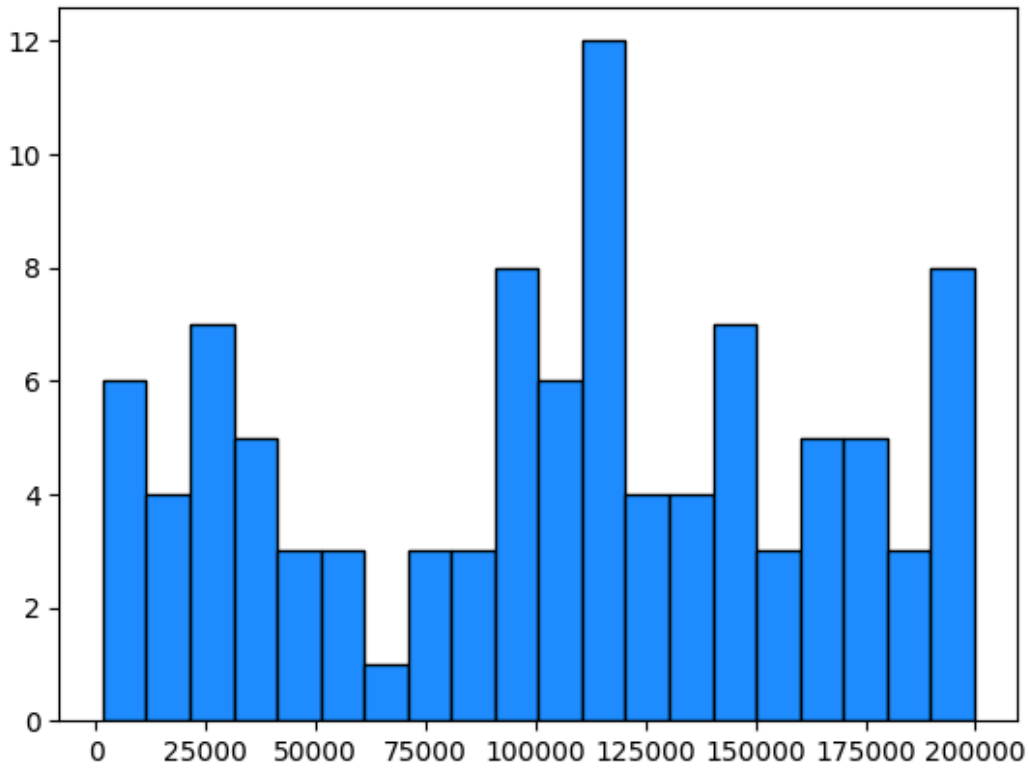
Next steps: Generate code with df | View recommended plots | New interactive sheet

Use salaries for exercises

```
females_salary = df.loc[df["Gender"] == 0, "EstimatedSalary"][:100]
males_salary = df.loc[df["Gender"] == 1, "EstimatedSalary"][:100]
```

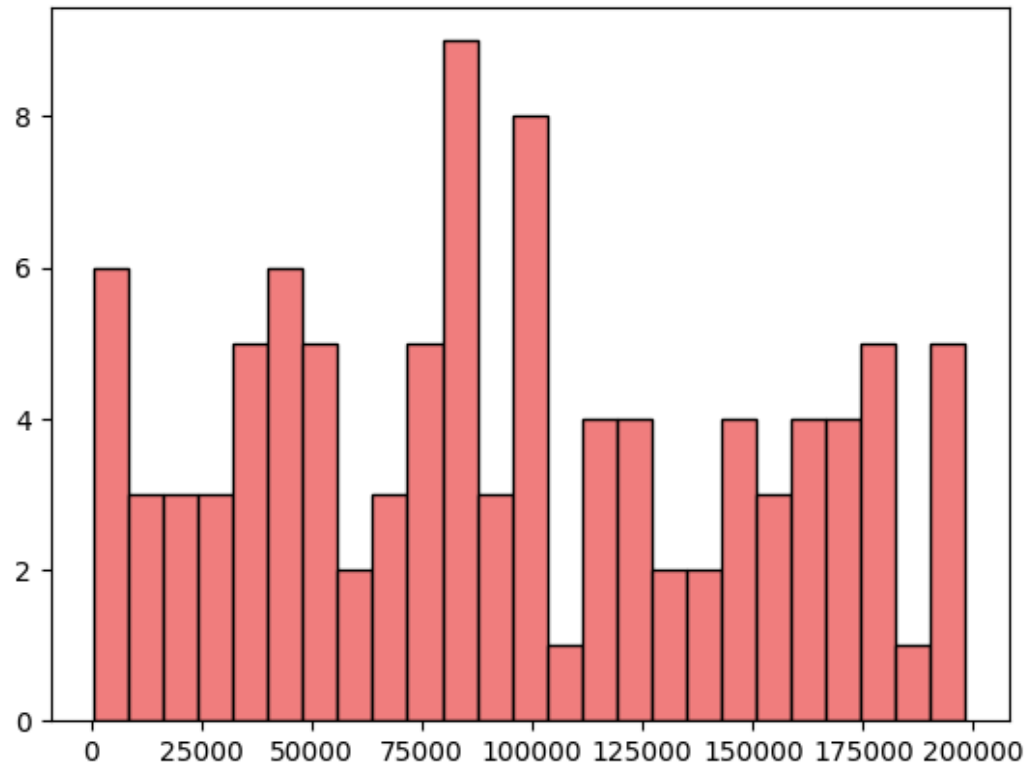
1.) Make a histogram of "males_salary". Set the color to "dodgerblue" and the edgecolor to "black". Set the number of bins to 20.

```
fig, ax = plt.subplots()

ax.hist(males_salary, color='dodgerblue', edgecolor='black', bins=20);
```



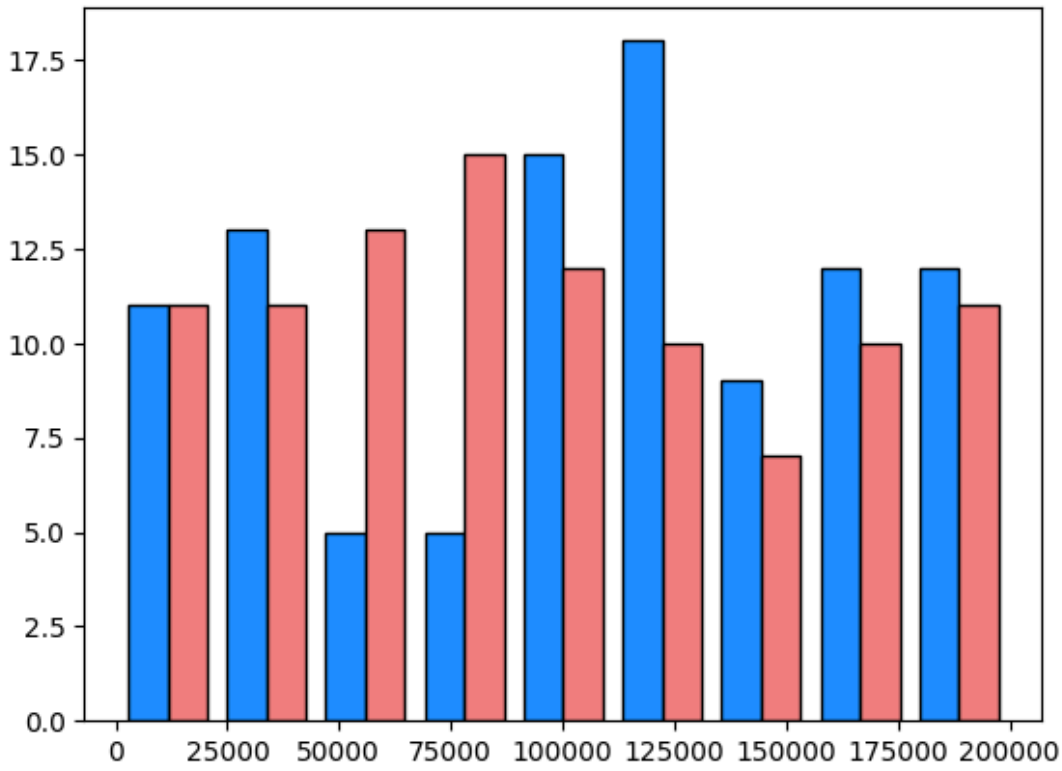
2.) Make a histogram of "females_salary". Set the color to "lightcoral" and the edgecolor to "black". Set the number of bins to 25.

```
fig, ax = plt.subplots()

ax.hist(females_salary, color='lightcoral', edgecolor='black', bins=25);
```



3.) Make a paired histogram of "males_salary" and "females_salary" so that their respective bins are displayed next to each other. Set the male's color to "dodgerblue" and the female's color to "lightcoral". Set the edgecolor to "black". Set the number of bins to "auto".

```
fig, ax = plt.subplots()
ax.hist([males_salary, females_salary], color=["dodgerblue", 'lightcoral'], edgecolor='
```



4.) Create a stacked histogram of "males_salary" and "females_salary" as follows:

- Display males on the bottom.
- Include a legend to indicate "Male" and "Female".
- Set custom colors for each histogram, and set the edgecolor to "black".
- Set the number of bins to 9.
- Set the title to "Male and Female Stacked Salaries"
- Label the x-axis "Salary"

```
fig, ax = plt.subplots()

ax.hist([males_salary, females_salary], color = ['#338DFF', "#FF33A2"],
        edgecolor="black", bins=9, stacked=True)

ax.set(title="Male and Female Stacked Salaries", xlabel="Salary")

plt.legend(["Male", "Female"]);
```