DSO 545: Statistical Computing and Data Visualization

Abbass Al Sharif

Fall 2019

Lab 5: Data Visualization Using Matplotlib (part 2)

1. Read the dataset `movie_scores.csv` into Python, and clean the data if necessary.

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

movies = pd.read_csv("movie_scores.csv")
movies = movies.drop(movies.columns[0], axis = 1)
movies
```

```
##                            MovieTitle  Tomatometer  AudienceScore
## 0                 The Shape of Water           91             73
## 1                      Black Panther           97             79
## 2                            Dunkirk           92             81
## 3                         The Martian           91             91
## 4  The Hobbit: An Unexpected Journey           64             83
```

2. Create the following bar plot to compare the movie scores from Tomatometer and Audience Score.

```
plt.figure(figsize = (12, 6))

#creat a bar plot
pos = np.arange(len(movies.MovieTitle))
width = 0.3

plt.bar(pos -width/2,
        movies.Tomatometer,
        width = width,
        label = "Tomatometer")
```

```
## <BarContainer object of 5 artists>
```

```
plt.bar(pos + width/2,
        movies.AudienceScore,
        width = width,
        label = "Audience Score")

#specify ticks
```

```
## <BarContainer object of 5 artists>
```

```
plt.xticks(pos, rotation = 10)
```

```
## ([<matplotlib.axis.XTick object at 0x11688b090>, <matplotlib.axis.XTick object at 0x116872790>, <mat
```

```
plt.yticks(np.arange(0, 101, 20))

#set tick labels
```

```
## ([<matplotlib.axis.YTick object at 0x116893210>, <matplotlib.axis.YTick object at 0x11688b890>, <mat
ax = plt.gca() # get current axes for setting tick labels
ax.set_xticklabels(movies.MovieTitle)
```

```
## [Text(0, 0, 'The Shape of Water'), Text(0, 0, 'Black Panther'), Text(0, 0, 'Dunkirk'), Text(0, 0, 'T
```

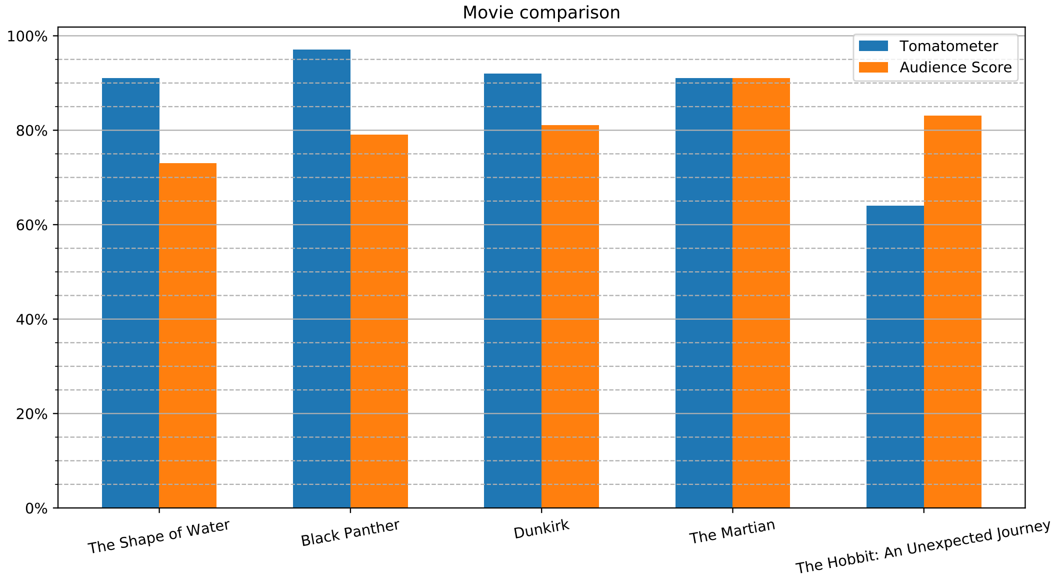
```
ax.set_yticklabels([str(i)+"%" for i in np.arange(0, 101, 20)])

# add minor ticks for y-axis in the interval of 5
```

```
## [Text(0, 0, '0%'), Text(0, 0, '20%'), Text(0, 0, '40%'), Text(0, 0, '60%'), Text(0, 0, '80%'), Text(
```

```
ax.set_yticks(np.arange(0,100, 5), minor = True)

# add major horizontal grid with solid lines
```

```
## [<matplotlib.axis.YTick object at 0x1168a7550>, <matplotlib.axis.YTick object at 0x1168d6fd0>, <matpl
```

```
ax.yaxis.grid(which = "major")

# add minor horizontal grid with dashed lines
ax.yaxis.grid(which = "minor", linestyle = '--')

# add a title
plt.title("Movie comparison")
plt.legend()
plt.show()
```



3. Read the dataset `smartphone_sales.csv` into Python, and clean the data if necessary.

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

sales = pd.read_csv('smartphone_sales.csv')
sales = sales.drop(sales.columns[0], axis = 1)
sales
```

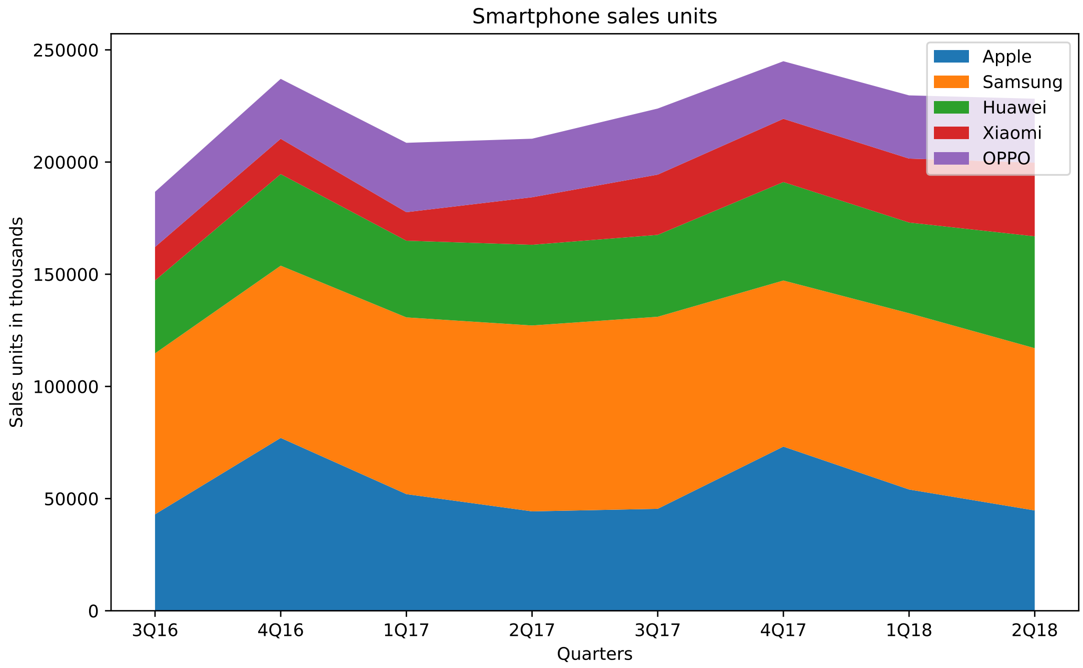
```
##    Quarter  Apple  Samsung  Huawei  Xiaomi   OPPO
## 0     3Q16  43001    71734   32490   14926  24591
## 1     4Q16  77039    76783   40804   15751  26705
## 2     1Q17  51993    78776   34181   12707  30922
## 3     2Q17  44315    82855   35964   21179  26093
## 4     3Q17  45442    85605   36502   26853  29449
## 5     4Q17  73175    74027   43887   28188  25660
## 6     1Q18  54059    78565   40426   28498  28173
## 7     2Q18  44715    72336   49847   32826  28511
```

4. Create the following stacked area plot to compare the sales units of different smart phone manufacturer.

```
plt.figure(figsize = (10,6))

#create a stacked area graph
labels =  sales.columns[1:]

plt.stackplot('Quarter', 'Apple', 'Samsung', 'Huawei', 'Xiaomi', 'OPPO', data=sales, labels=labels)
# Add legend
```

```
## [<matplotlib.collections.PolyCollection object at 0x1196fed10>, <matplotlib.collections.PolyCollecti
```

```
plt.legend()
# Add labels and title
plt.xlabel('Quarters')
plt.ylabel('Sales units in thousands')
plt.title('Smartphone sales units')
# Show plot
plt.show()
```



5. Read the dataset `anage.csv` into Python, and clean the data if necessary.

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

data = pd.read_csv('anage_data.csv')
data = data.drop(data.columns[0], axis = 1)
```

6. The dataset `anage.csv` is not complete. Filter the data so that you end up with samples containing a body mass and a maximum longevity.

```
longevity = 'Maximum longevity (yrs)'
mass = 'Body mass (g)'
data = data[np.isfinite(data[longevity]) & np.isfinite(data[mass])]
```

7. Create 4 subsets of the given dataset: amphibia, aves, mammalia, and reptilia

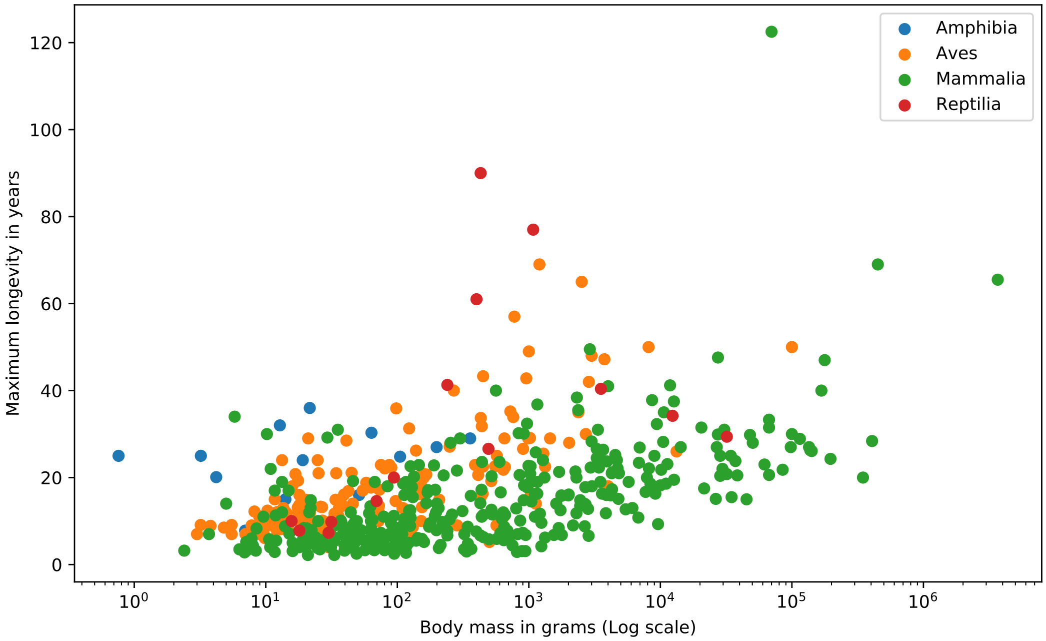
```
data.Class.unique()
```

```
## array(['Amphibia', 'Aves', 'Mammalia', 'Reptilia'], dtype=object)
```

```
amphibia = data[data['Class'] == "Amphibia"]
aves = data[data['Class'] == 'Aves']
mammalia = data[data['Class'] == 'Mammalia']
reptilia = data[data['Class'] == 'Reptilia']
```

8. Create a scatter plot that shows the corrlation between the body mass and the maximum longevity. Use log scale for the x-axis.

```
# Create figure
plt.figure(figsize=(10, 6), dpi=300)
# Create scatter plot
plt.scatter(amphibia[mass], amphibia[longevity], label='Amphibia')
plt.scatter(aves[mass], aves[longevity], label='Aves')
plt.scatter(mammalia[mass], mammalia[longevity], label='Mammalia')
plt.scatter(reptilia[mass], reptilia[longevity], label='Reptilia')
# Add legend
plt.legend()
# Log scale
ax = plt.gca()
ax.set_xscale('log')
#ax.set_yscale('log')
# Add labels
plt.xlabel('Body mass in grams (Log scale)')
plt.ylabel('Maximum longevity in years')
# Show plot
plt.show()
```


Amphibia
Aves
Mammalia
Reptilia
Maximum longevity in years
0
20
40
60
80
100
120
10^0
10^1
10^2
10^3
10^4
10^5
10^6
Body mass in grams (Log scale)