FA_LJC_DataVisuPyth

July 11, 2020

Final Assignment for Data Visualisation using Python course By Laura-June Clarke>

BAR PLOT

Question 1: Use the pandas read_csv method to read the csv file into a pandas dataframe and upload a screenshot of your dataframe with the actual numbers.

[1]:
```
# import the libraries needed
import numpy as np
import pandas as pd

# read the dataset into a pandas dataframe
df_survey = pd.read_csv('https://cocl.us/datascience_survey_data')

# confirm it has been completed
print('Dataset on survey downloaded and read into a pandas dataframe!')
```

```
Dataset on survey downloaded and read into a pandas dataframe!
```

[2]:
```
# Print the 5 first rows to visualise the dataset
df_survey.head()
```

[2]:

	Unnamed: 0	Very interested	Somewhat interested	Not interested
0	Big Data (Spark / Hadoop)	1332	729	127
1	Data Analysis / Statistics	1688	444	60
2	Data Journalism	429	1081	610
3	Data Visualization	1340	734	102
4	Deep Learning	1263	770	136

[3]:
```
df_survey.rename(columns={'Unnamed: 0':'Data Science Area'},inplace=True) #␣
↪Let's rename the first column
```

```
df_survey.set_index('Data Science Area', inplace=True) # Set the index to the␣
↪first column about Data Science Areas
df_survey.sort_values(by='Very interested', ascending=False, inplace=True) #␣
↪Sort in descending order of Very interested.
df_survey.head() # print the dataframe
```

[3]:

Data Science Area	Very interested	Somewhat interested	Not interested
Data Analysis / Statistics	1688	444	60
Machine Learning	1629	477	74
Data Visualization	1340	734	102
Big Data (Spark / Hadoop)	1332	729	127
Deep Learning	1263	770	136

Question 2: Use the artist layer of Matplotlib to replicate the bar chart seen in the instructions to visualize the percentage of the respondents' interest in the different data science topics surveyed.

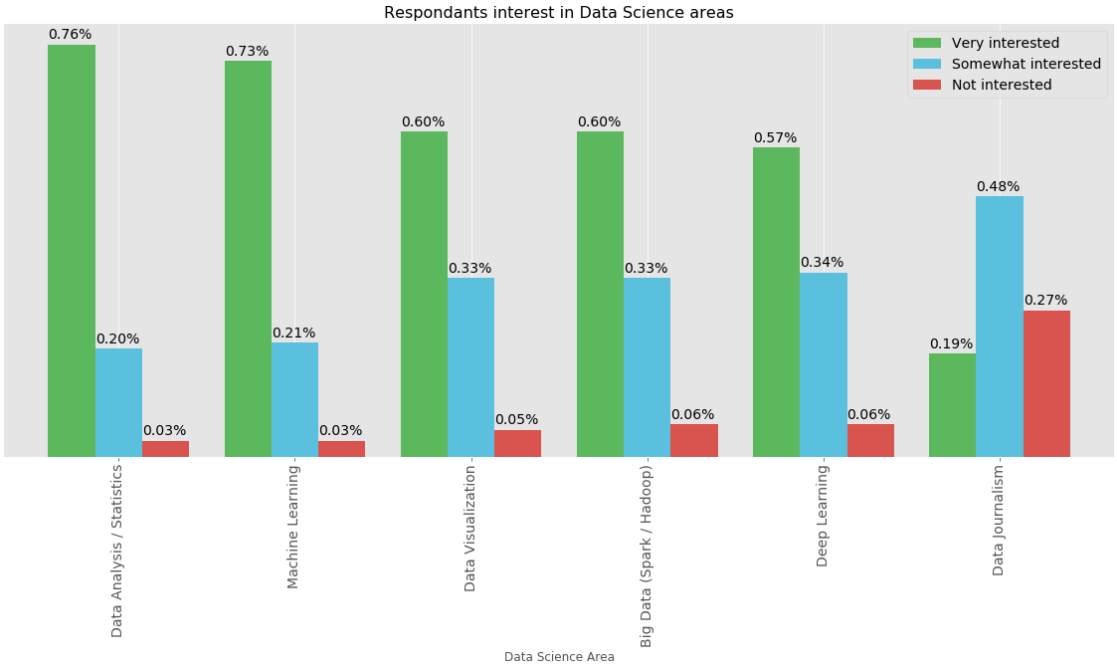
[4]:
```
df_percent = df_survey.apply(lambda row: round(row/2233.0, 2), axis=1) #Let's␣
↪convert numbers into percentage
df_percent.head()
```

[4]:

Data Science Area	Very interested	Somewhat interested	Not interested
Data Analysis / Statistics	0.76	0.20	0.03
Machine Learning	0.73	0.21	0.03
Data Visualization	0.60	0.33	0.05
Big Data (Spark / Hadoop)	0.60	0.33	0.06
Deep Learning	0.57	0.34	0.06

[5]:
```
# Magic function for our matplotlib graphs will be included in your notebook,␣
↪next to the code.
```

```
%matplotlib inline

# Import the different libraries needed
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.patches as patches

# Adjusts the style to emulate ggplo
mpl.style.use('ggplot')

# Create the plot and set the parameters
ax = df_percent.plot(kind='bar', figsize=(20, 8), width=0.8, color=['#5cb85c',␣
 ↪'#5bc0de', '#d9534f'])

# Get ride of borders
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)

# Adjust fontsize of labels
plt.xticks(fontsize = 14)
plt.yticks([])

# Annotate bars with percentage
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.annotate('{:.2f}%'.format(height), (x, y + height + 0.01),fontsize = 14)

# Adjust and set parameters of legend and title
ax.legend(labels=df_percent.columns, loc='upper right',fontsize = 14 )
ax.set_title('Respondants interest in Data Science areas', fontsize=16)

# Show the plot
plt.show()
```



CHLOROPLET MAP

Question 3: Convert the San Francisco dataset, which you can also find here, https://cocl.us/sanfran_crime_dataset, into a pandas dataframe,like the one shown in the instructions, that represents the total number of crimes in each neighborhood.

[6]:
```
# import the libraries needed
import numpy as np
import pandas as pd

# read the dataset into a pandas dataframe
df_SF = pd.read_csv('https://cocl.us/sanfran_crime_dataset')

# confirm it has been completed
print('Dataset on crime rate in San Francisco downloaded and read into a pandas␣
 ↪dataframe!')

df_SF.head()
```

```
Dataset on crime rate in San Francisco downloaded and read into a pandas
dataframe!
```

[6]:

	IncidntNum	Category	Descript	\
0	120058272	WEAPON LAWS	POSS OF PROHIBITED WEAPON	
1	120058272	WEAPON LAWS	FIREARM, LOADED, IN VEHICLE, POSSESSION OR USE	
2	141059263	WARRANTS	WARRANT ARREST	

```
3   160013662  NON-CRIMINAL                                  LOST PROPERTY
4   160002740  NON-CRIMINAL                                  LOST PROPERTY

  DayOfWeek                     Date   Time  PdDistrict      Resolution  \
0    Friday  01/29/2016 12:00:00 AM  11:00    SOUTHERN  ARREST, BOOKED
1    Friday  01/29/2016 12:00:00 AM  11:00    SOUTHERN  ARREST, BOOKED
2    Monday  04/25/2016 12:00:00 AM  14:59     BAYVIEW  ARREST, BOOKED
3   Tuesday  01/05/2016 12:00:00 AM  23:50  TENDERLOIN            NONE
4    Friday  01/01/2016 12:00:00 AM  00:30     MISSION            NONE

                  Address           X          Y  \
0  800 Block of BRYANT ST -122.403405  37.775421
1  800 Block of BRYANT ST -122.403405  37.775421
2   KEITH ST / SHAFTER AV -122.388856  37.729981
3  JONES ST / OFARRELL ST -122.412971  37.785788
4    16TH ST / MISSION ST -122.419672  37.765050

                              Location            PdId
0   (37.775420706711, -122.403404791479)  12005827212120
1   (37.775420706711, -122.403404791479)  12005827212168
2  (37.7299809672996, -122.388856204292)  14105926363010
3  (37.7857883766888, -122.412970537591)  16001366271000
4  (37.7650501214668, -122.419671780296)  16000274071000
```

[7]:
```
# Let's group by district
d=df_SF[["PdDistrict", "Location"]]
df_Neighbourhood = d.groupby("PdDistrict").count()
df_Neighbourhood.index.name='Neighbourhood'
df_Neighbourhood.rename(columns={'PdDistrict':'Neighbourhood','Location':
 ↪'Count'}, inplace=True)
df_Neighbourhood= df_Neighbourhood.reindex(["CENTRAL", "NORTHERN", "PARK",␣
 ↪"SOUTHERN", "MISSION", "TENDERLOIN", "RICHMOND", "TARAVAL", "INGLESIDE",␣
 ↪"BAYVIEW"])
df_Neighbourhood=df_Neighbourhood.reset_index()

df_Neighbourhood
```

[7]:
```
  Neighbourhood  Count
0       CENTRAL  17666
1      NORTHERN  20100
2          PARK   8699
3      SOUTHERN  28445
4       MISSION  19503
5    TENDERLOIN   9942
6      RICHMOND   8922
7       TARAVAL  11325
8     INGLESIDE  11594
```

```
9       BAYVIEW  14303
```

[8]:
```
!conda install -c conda-forge folium=0.5.0 --yes
import folium

print('Folium installed and imported!')

!wget --quiet https://cocl.us/sanfran_geojson
!pip install wget

print('SF_geo file downloaded!')
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

Folium installed and imported!
Requirement already satisfied: wget in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (3.2)
SF_geo file downloaded!
```

[9]:
```
SF_GEO=r'sanfran_geojson' # # geojson file

# Define San Francisco Map with folium.Map
SF_Map=folium.Map(location=[37.7749, -122.4194], zoom_start=12)

# Choropleth
SF_Map.choropleth(
    geo_data=SF_GEO,
    data=df_Neighbourhood,
    columns=['Neighbourhood', 'Count'],
    key_on='feature.properties.DISTRICT',
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Crime Rate in San Francisco'
)
SF_Map
```

[9]:
```
<folium.folium.Map at 0x7f4b9eb20198>
```

[ ]:
```

```