ASSIGNMENT CS V A 2023 - SUBMISSION DAY 10 November 2023

You have to submit a handwritten assignment for the same . No need to write questions.

WRITE QUESTION NUMBERS AND NUMBERING OF SUB PARTS SAME AS MENTIONED IN THIS SHEET.

Data Exercise

Answer the following questions using the `bestsellers.csv` dataset

Question 1

1)import necessary libraries books = read the csv file - bestsellers.csv dataset

In [1]:
```
import pandas as pd
books = pd.read_csv("bestsellers.csv")
```

2)Inspect the first 5 rows of dataset 3)Describe the data

In [2]:
```
print(books.head())
print(books.describe())
```

```
                                                Name  \
0                        10-Day Green Smoothie Cleanse
1                                    11/22/63: A Novel
2              12 Rules for Life: An Antidote to Chaos
3                               1984 (Signet Classics)
4  5,000 Awesome Facts (About Everything!) (Natio...

                     Author  User Rating  Reviews  Price  Year        Genre
0                  JJ Smith          4.7    17350      8  2016  Non Fiction
1              Stephen King          4.6     2052     22  2011      Fiction
2         Jordan B. Peterson          4.7    18979     15  2018  Non Fiction
3              George Orwell          4.7    21424      6  2017      Fiction
4  National Geographic Kids          4.8     7665     12  2019  Non Fiction
       User Rating       Reviews        Price         Year
count   550.000000    550.000000  550.000000   550.000000
mean      4.618364  11953.281818   13.100000  2014.000000
std       0.226980  11731.132017   10.842262     3.165156
min       3.300000     37.000000    0.000000  2009.000000
25%       4.500000   4058.000000    7.000000  2011.000000
50%       4.700000   8580.000000   11.000000  2014.000000
75%       4.800000  17253.250000   16.000000  2017.000000
max       4.900000  87841.000000  105.000000  2019.000000
```

4)Find the books written by Pete Souza

In [3]:
```
books[books["Author"] == "Pete Souza"]
```

Out[3]:

	Name	Author	User Rating	Reviews	Price	Year	Genre
244	Obama: An Intimate Portrait	Pete Souza	4.9	3192	22	2017	Non Fiction

5)Find the books that have a price between 50 and 60 dollars

In [4]:
```
books[books["Price"].between(50,60)]
```

Out[4]:

	Name	Author	User Rating	Reviews	Price	Year	Genre
151	Hamilton: The Revolution	Lin-Manuel Miranda	4.9	5867	54	2016	Non Fiction
159	Harry Potter Paperback Box Set (Books 1-7)	J. K. Rowling	4.8	13471	52	2016	Fiction
346	The Book of Basketball: The NBA According to T...	Bill Simmons	4.7	858	53	2009	Non Fiction

6)Find all the books written by Kristin Hannah, Andy Weir, or Delia Owens

In [5]:
```
books[books["Author"].isin(["Kristin Hannah", "Andy Weir", "Delia Owens"])]
```

Out[5]:

	Name	Author	User Rating	Reviews	Price	Year	Genre
433	The Martian	Andy Weir	4.7	39459	9	2015	Fiction
437	The Nightingale: A Novel	Kristin Hannah	4.8	49288	11	2015	Fiction
438	The Nightingale: A Novel	Kristin Hannah	4.8	49288	11	2016	Fiction
534	Where the Crawdads Sing	Delia Owens	4.8	87841	15	2019	Fiction

7)Find 2012's top 5 Fiction books with the most Reviews

In [6]:
```
df = books[(books["Year"] == 2012) & (books["Genre"] == "Fiction")]
df.sort_values("Reviews", ascending=False).head()
```

Out[6]:

	Name	Author	User Rating	Reviews	Price	Year	Genre
135	Gone Girl	Gillian Flynn	4.0	57271	10	2012	Fiction
365	The Fault in Our Stars	John Green	4.7	50482	13	2012	Fiction
106	Fifty Shades of Grey: Book One of the Fifty Sh...	E L James	3.8	47265	14	2012	Fiction
409	The Hunger Games (Book 1)	Suzanne Collins	4.7	32122	8	2012	Fiction
238	Mockingjay (The Hunger Games)	Suzanne Collins	4.5	26741	8	2012	Fiction

In [7]:
```
#Use pivot table to show the sum of 'Price' for genres and authors. Example as shown below but for every author .
```

In [8]:
```
pivot_total_Sum = pd.pivot_table(books, values='Price', index='Genre', columns='Author', aggfunc='sum')
print(pivot_total_Sum )
```

```
Author       Abraham Verghese  Adam Gasiewski  Adam Mansbach  Adir Levy  \
Genre
Fiction                  22.0             NaN            9.0       13.0
Non Fiction               NaN             6.0            NaN        NaN

Author       Admiral William H. McRaven  Adult Coloring Book Designs  \
Genre
Fiction                             NaN                          NaN
Non Fiction                        11.0                          4.0

Author       Alan Moore  Alex Michaelides  Alice Schertle  Allie Brosh  ...  \
Genre                                                                   ...
Fiction            42.0              14.0             0.0          NaN  ...
Non Fiction         NaN               NaN             NaN         17.0  ...

Author       Todd Burpo  Tony Hsieh  Tucker Carlson  Veronica Roth  \
Genre
Fiction             NaN         NaN             NaN           49.0
Non Fiction        20.0        15.0            16.0            NaN

Author       W. Cleon Skousen  Walter Isaacson  William Davis  \
Genre
Fiction                   NaN              NaN            NaN
Non Fiction              12.0             61.0           12.0

Author       William P. Young  Wizards RPG Team  Zhi Gang Sha
Genre
Fiction                  16.0              81.0           NaN
Non Fiction               NaN               NaN          23.0

[2 rows x 248 columns]
```

In [9]:
```
# Create Pivot table for average price and rating based on publication year
pivot_price_rating_year = pd.pivot_table(books, values=['Price', 'User Rating'], index='Year', aggfunc='mean')
print(pivot_price_rating_year)
```

```
      Price  User Rating
Year
2009  15.40        4.584
2010  13.48        4.558
2011  15.10        4.558
2012  15.30        4.532
2013  14.60        4.554
2014  14.64        4.622
2015  10.42        4.648
2016  13.18        4.678
2017  11.38        4.660
2018  10.52        4.668
2019  10.08        4.740
```

In [10]:
```
# Cut data into price ranges of 5 bins and analyze number of Reviews
books['price_range'] = pd.cut(books['Price'], bins=5)
copies_sold_by_price_range = books.groupby('price_range')['Reviews'].sum()
print(copies_sold_by_price_range)
```

```
price_range
(-0.105, 21.0]    6113486
(21.0, 42.0]       337664
(42.0, 63.0]       105996
(63.0, 84.0]         3801
(84.0, 105.0]       13358
Name: Reviews, dtype: int64
```

In [11]:
```
# Calculate the average Price for each genre in each publication year, as shown below.
```

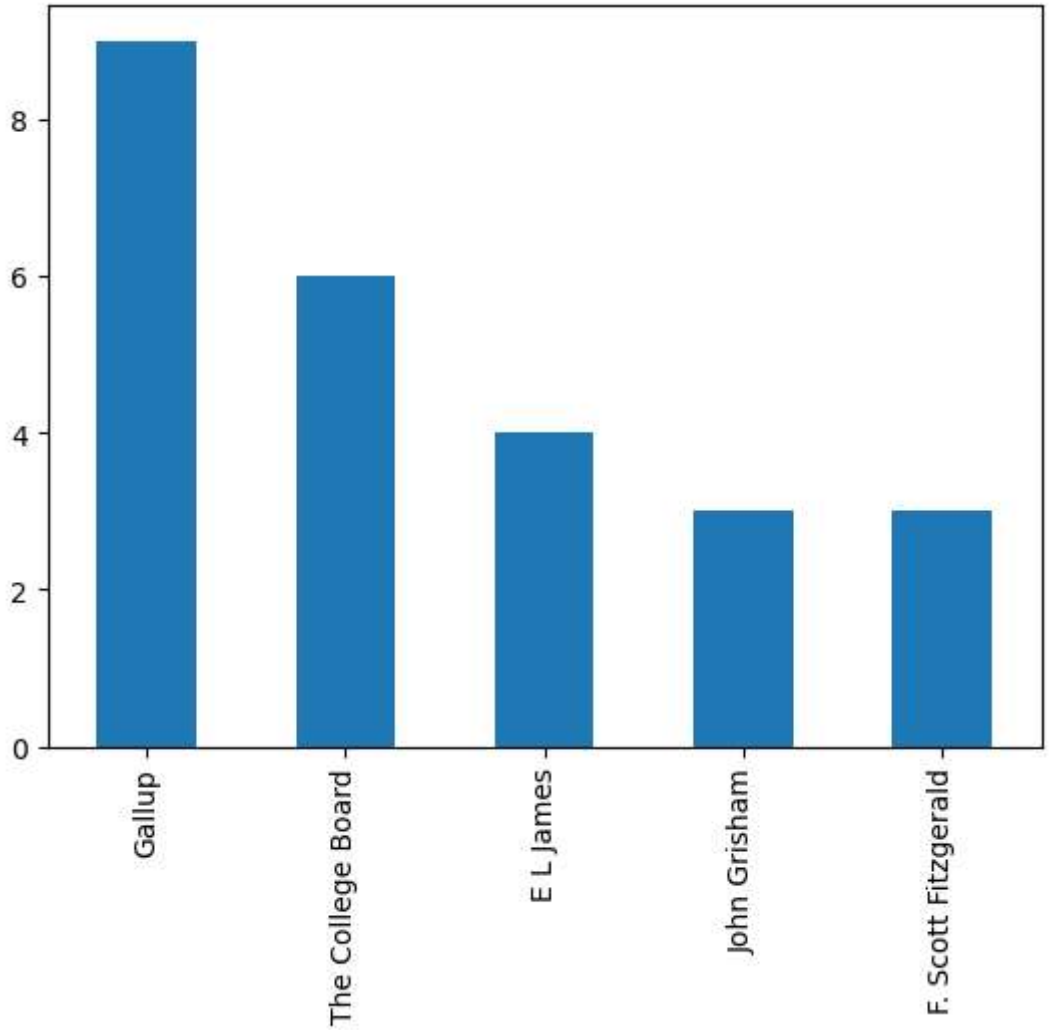
In [12]:
```
multi_index_data = books.set_index(['Genre', 'Year'])

# Calculate the average Price for each genre in each publication year
avg_copies_sold_per_genre_per_year = multi_index_data.groupby(level=[0, 1])['Price'].mean()
print(avg_copies_sold_per_genre_per_year)
```

```
Genre        Year
Fiction      2009    15.583333
             2010     9.700000
             2011    11.619048
             2012    12.285714
             2013    10.708333
             2014    10.172414
             2015     9.352941
             2016    12.631579
             2017     8.833333
             2018     8.761905
             2019     9.350000
Non Fiction  2009    15.230769
             2010    16.000000
             2011    17.620690
             2012    17.482759
             2013    18.192308
             2014    20.809524
             2015    10.969697
             2016    13.516129
             2017    13.730769
             2018    11.793103
             2019    10.566667
Name: Price, dtype: float64
```

8)Create a bar plot showing the 5 authors who have the most books with a rating under 4.5

In [13]:
```
low_rated = books[books["User Rating"] < 4.5]
low_rated["Author"].value_counts().head().plot(kind="bar")
```

Out[13]: <Axes: >



Group By Exercise

This exercise uses the `sports.csv` dataset

Question 2

9)import necessary libraries stats = read the csv file - sports.csv

In [14]:
```
import pandas as pd
import matplotlib.pyplot as plt
stats = pd.read_csv("sports.csv")
```

In [15]:
```
#Find the 5 teams that had the most "Red Cards"
stats.groupby("Team")["Red Cards"].sum().nlargest(5)
```

Out[15]:
```
Team
Rayo Vallecano    8
Levante UD        7
Getafe CF         6
RC Celta          6
Real Madrid       6
Name: Red Cards, dtype: int64
```

11)Find the average number of "Long passes" made by each Position (Goalkeeper, Forward, etc.)

In [16]:
```
stats.groupby("Position")["Long passes"].mean()
```

Out[16]:
```
Position
Defender      102.610811
Forward        23.787234
Goalkeeper    242.157895
Midfielder     60.445455
Name: Long passes, dtype: float64
```

12)Find the 10 Shirt numbers that scored the most goals (top 12)

In [17]:
```
stats.groupby("Shirt number")["Goals scored"].sum().nlargest(10)
```

Out[17]:
```
Shirt number
9.0     169
10.0    117
7.0     101
19.0     69
11.0     56
22.0     47
12.0     45
17.0     38
23.0     27
8.0      23
Name: Goals scored, dtype: int64
```

Question 3

Use `agg` to create a dataframe that contains:

- A `total` column containing the total "Shots" taken by each team
- A `on_target` column containing the total "Shots on target" taken by each team
- It should look like the following dataframe (but for all teams in the dataset):

Team	total	on_target
Real Betis	300	158
Levante UD	314	157

In [18]:
```
shots = stats.groupby("Team").agg(total=("Shots", sum),on_target=("Shots on target", sum))
shots
```

Out[18]:

Team	total	on_target
Athletic Club	332	151
Atlético de Madrid	339	159
CD Leganés	334	132
D. Alavés	299	109
FC Barcelona	445	249
Getafe CF	283	121
Girona FC	324	147
Levante UD	314	157
R. Valladolid CF	319	131
RC Celta	329	159
RCD Espanyol	333	144
Rayo Vallecano	337	151
Real Betis	300	158
Real Madrid	448	216
Real Sociedad	333	144
SD Eibar	422	153
SD Huesca	343	142
Sevilla FC	401	178
Valencia CF	381	165
Villarreal CF	354	172

Question 4

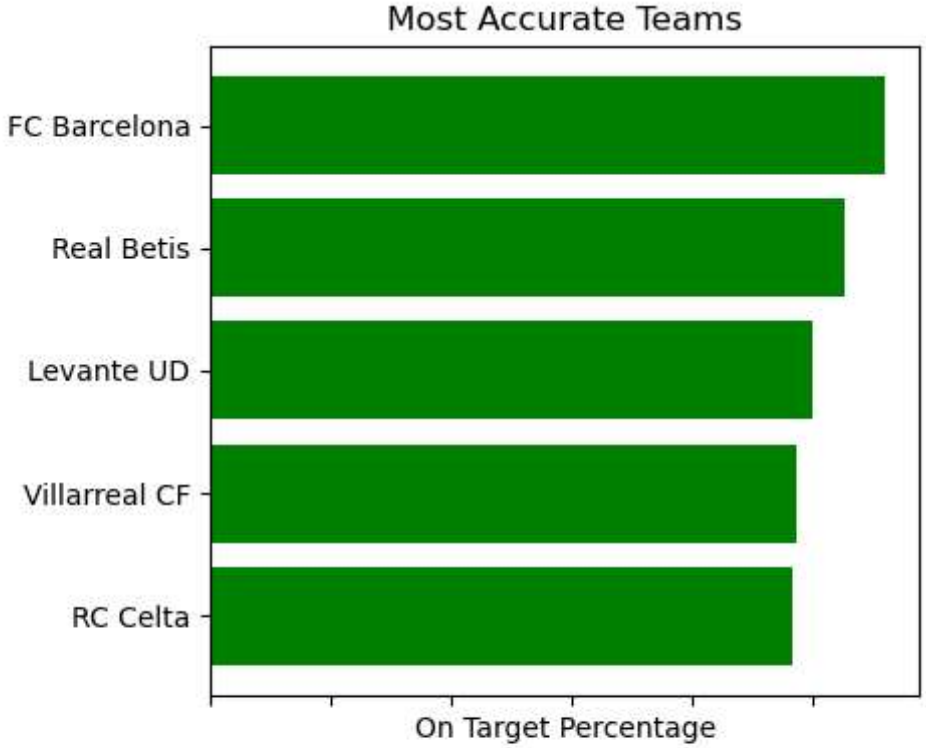
Use the dataframe from above to create the following figure:

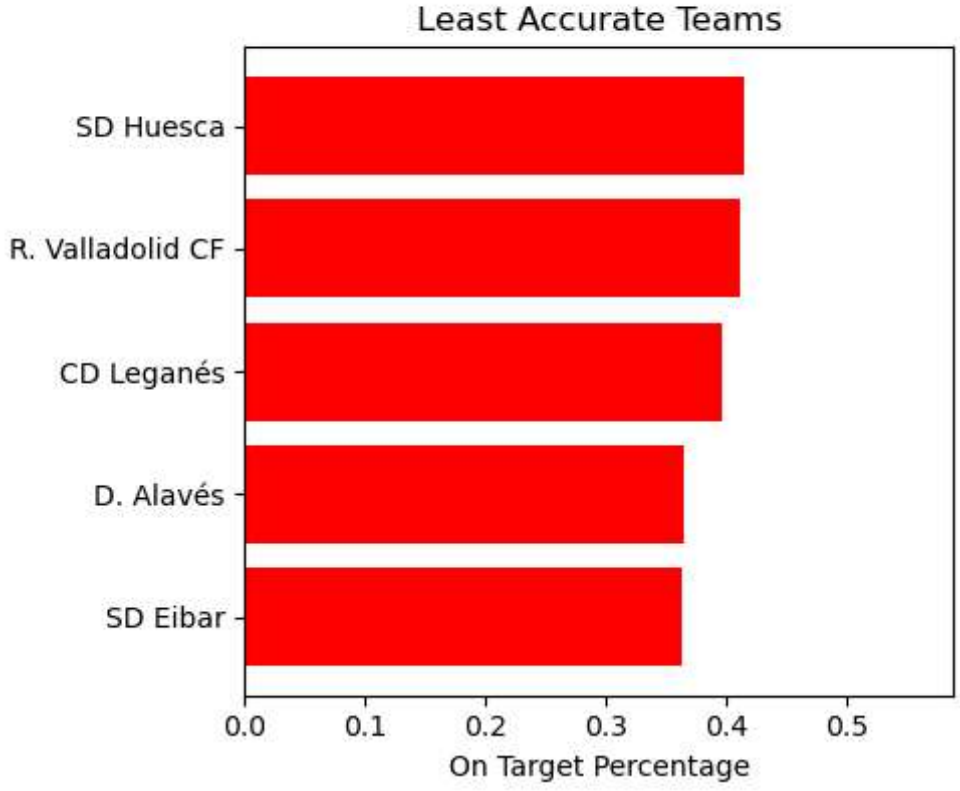
- Notice the layout (2 rows by 1 column)
- The top chart shows the top 5 most accurate teams (highest on-target shot pecentage)
- The bottom chart shows the 5 least accurate teams (lowest on-target shot pecentage)
- Both plots share the same x-axis
- Notice how the data is sorted within each plot!

In [19]:
```
shots["accuracy"] = shots["on_target"] / shots["total"]
```

In [20]:

```python
import matplotlib.pyplot as plt

fig, axs = plt.subplots(2, 1, figsize=(5, 8), sharex=True)

# Replace 'shots' and 'accuracy' with your actual data
most_accurate = shots["accuracy"].nlargest().sort_values(ascending=True)
least_accurate = shots["accuracy"].nsmallest()

# Plotting most accurate teams
axs[0].barh(most_accurate.index, most_accurate, color="green")
axs[0].set_title("Most Accurate Teams")
axs[0].set_xlabel("On Target Percentage")

# Plotting least accurate teams
axs[1].barh(least_accurate.index, least_accurate, color="red")
axs[1].set_title("Least Accurate Teams")
axs[1].set_xlabel("On Target Percentage")

plt.style.use('default')  # Using the default style (removing ggplot style)
plt.tight_layout()  # Adjusts subplot params for a neat layout
plt.show()
```


Most Accurate Teams
FC Barcelona
Real Betis
Levante UD
Villarreal CF
RC Celta
On Target Percentage


Least Accurate Teams
SD Huesca
R. Valladolid CF
CD Leganés
D. Alavés
SD Eibar
0.0
0.1
0.2
0.3
0.4
0.5
On Target Percentage

In [ ]: