UNIVERSITY
of Karachi

DATA ANALYSIS REPORT

“F.R.I.E.N.D.S“

Prepared By

- *Munaem Qasim*
- *Syeda Muskan Batool*

DATED: January 2022

DEGREE PROGRAM:	*BS Financial Mathematics*
	Second Year, Fourth Semester
NAME OF PROJECT:	*Data Analysis of the Show Friends*
	Using PANDAS on CSV file
SUBMITTED TO:	*Sir Syed Umaid Ahmed*

TABLE OF CONTENT

Introduction:

In this project we use Python (Jupyter Note Book) And Import Pandas , Matplotlib , and seaborn libraries to analyze the data of the show "Friends". The data contains thousands of episodes.

Let's discuss some important terms and libraries first...

Data Science:

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence (AI) systems to perform tasks that ordinarily require human intelligence. In turn, these systems generate insights which analysts and business users can translate into tangible business value.

Data Analysis:

Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively.

Pandas:

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

Matplotlib:

Matplotlib is a plotting library available for the Python programming language as a component of NumPy, a big data numerical handling resource. Matplotlib uses an object oriented API to embed plots in Python applications.

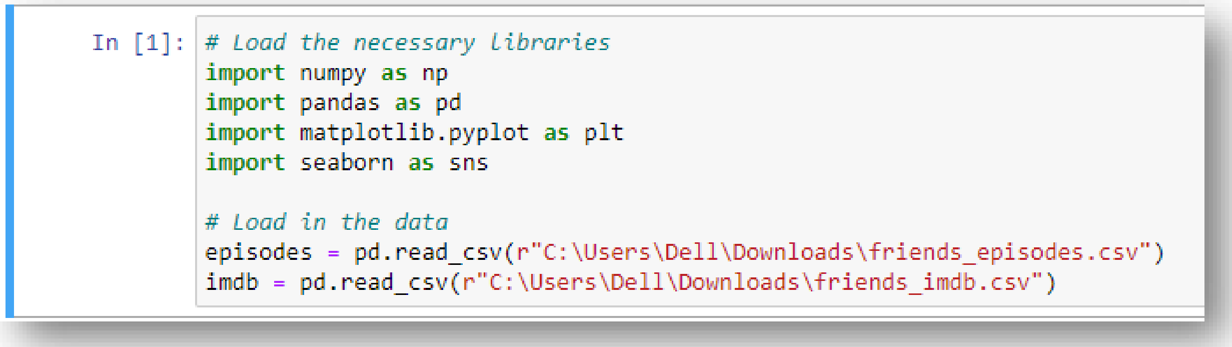
Seaborn:

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.

…Now moving towards the coding section…

STEP NO: 1

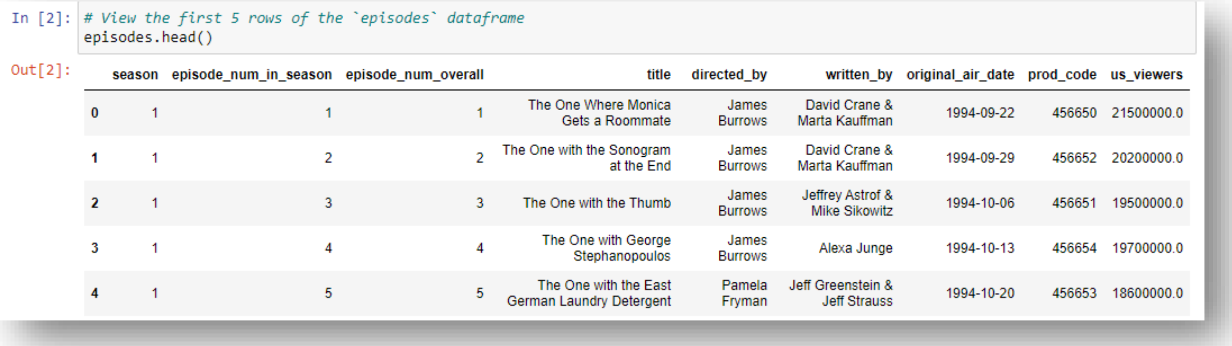
Import All the Necessary Libraries



```
In [1]: # Load the necessary libraries
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns

        # Load in the data
        episodes = pd.read_csv(r"C:\Users\Dell\Downloads\friends_episodes.csv")
        imdb = pd.read_csv(r"C:\Users\Dell\Downloads\friends_imdb.csv")
```

STEP NO: 2

View the First 5 rows of 'episodes' dataframe



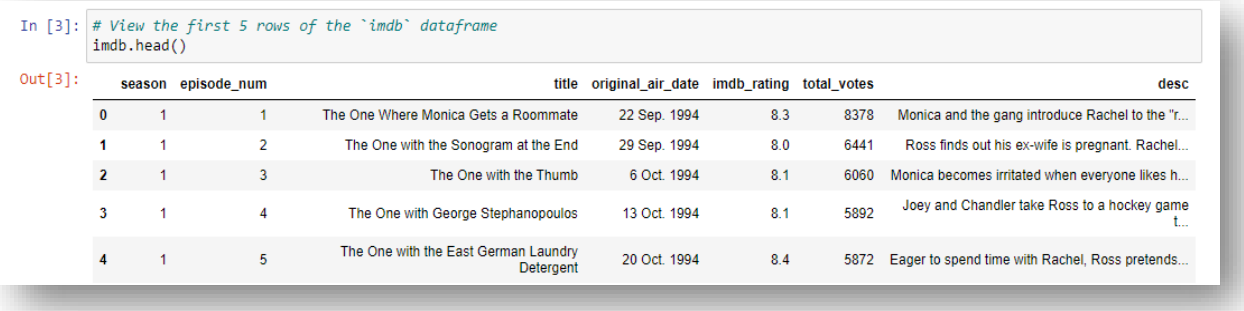
In [2]:
```
# View the first 5 rows of the `episodes` dataframe
episodes.head()
```

Out[2]:

	season	episode_num_in_season	episode_num_overall	title	directed_by	written_by	original_air_date	prod_code	us_viewers
0	1	1	1	The One Where Monica Gets a Rommate	James Burrows	David Crane & Marta Kauffman	1994-09-22	456650	21500000.0
1	1	2	2	The One with the Sonogram at the End	James Burrows	David Crane & Marta Kauffman	1994-09-29	456652	20200000.0
2	1	3	3	The One with the Thumb	James Burrows	Jeffrey Astrof & Mike Sikowitz	1994-10-06	456651	19500000.0
3	1	4	4	The One with George Stephanopoulos	James Burrows	Alexa Junge	1994-10-13	456654	19700000.0
4	1	5	5	The One with the East German Laundry Detergent	Pamela Fryman	Jeff Greenstein & Jeff Strauss	1994-10-20	456653	18600000.0

STEP NO: 3

View the First 5 rows of 'imbd' dataframe



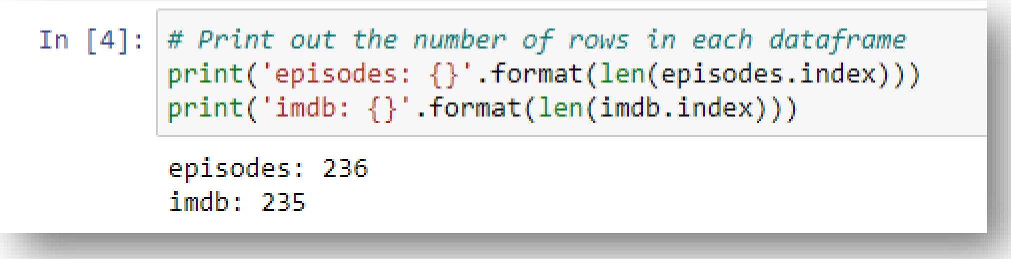
In [3]:
```
# View the first 5 rows of the `imdb` dataframe
imdb.head()
```

Out[3]:

	season	episode_num	title	original_air_date	imdb_rating	total_votes	desc
0	1	1	The One Where Monica Gets a Roommate	22 Sep. 1994	8.3	8378	Monica and the gang introduce Rachel to the "r...
1	1	2	The One with the Sonogram at the End	29 Sep. 1994	8.0	6441	Ross finds out his ex-wife is pregnant. Rachel...
2	1	3	The One with the Thumb	6 Oct. 1994	8.1	6060	Monica becomes irritated when everyone likes h...
3	1	4	The One with George Stephanopoulos	13 Oct. 1994	8.1	5892	Joey and Chandler take Ross to a hockey game t...
4	1	5	The One with the East German Laundry Detergent	20 Oct. 1994	8.4	5872	Eager to spend time with Rachel, Ross pretends...

STEP NO: 4

Print out the numbers of rows in each dataframe

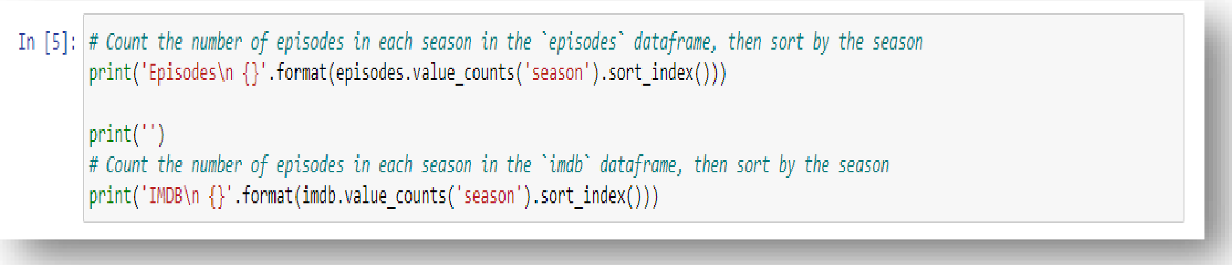


In [4]:
```
# Print out the number of rows in each dataframe
print('episodes: {}'.format(len(episodes.index)))
print('imdb: {}'.format(len(imdb.index)))
```

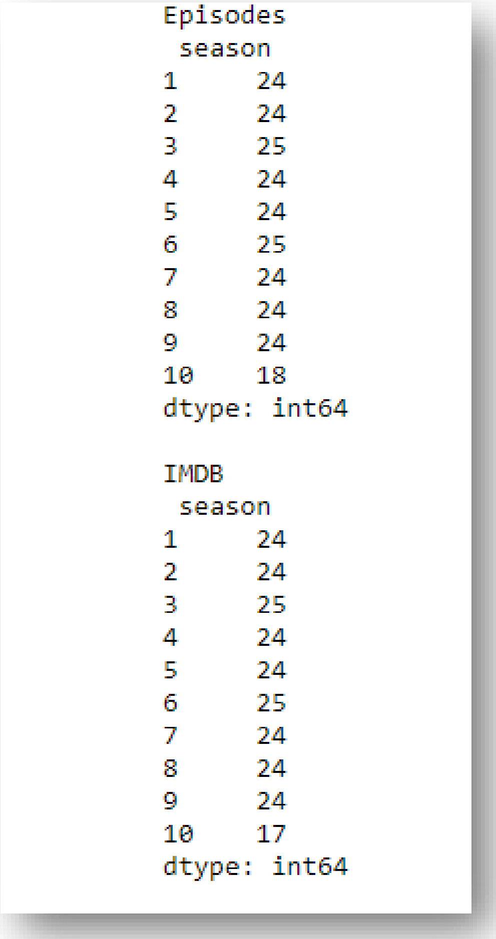
```
episodes: 236
imdb: 235
```

STEP NO: 5

Counting the Number of episodes in each season



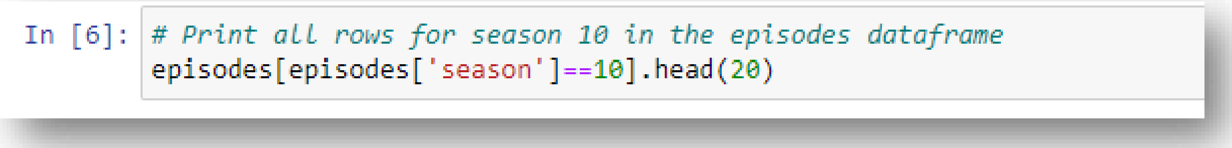
```
In [5]: # Count the number of episodes in each season in the `episodes` dataframe, then sort by the season
        print('Episodes\n {}'.format(episodes.value_counts('season').sort_index()))

        print('')
        # Count the number of episodes in each season in the `imdb` dataframe, then sort by the season
        print('IMDB\n {}'.format(imdb.value_counts('season').sort_index()))
```



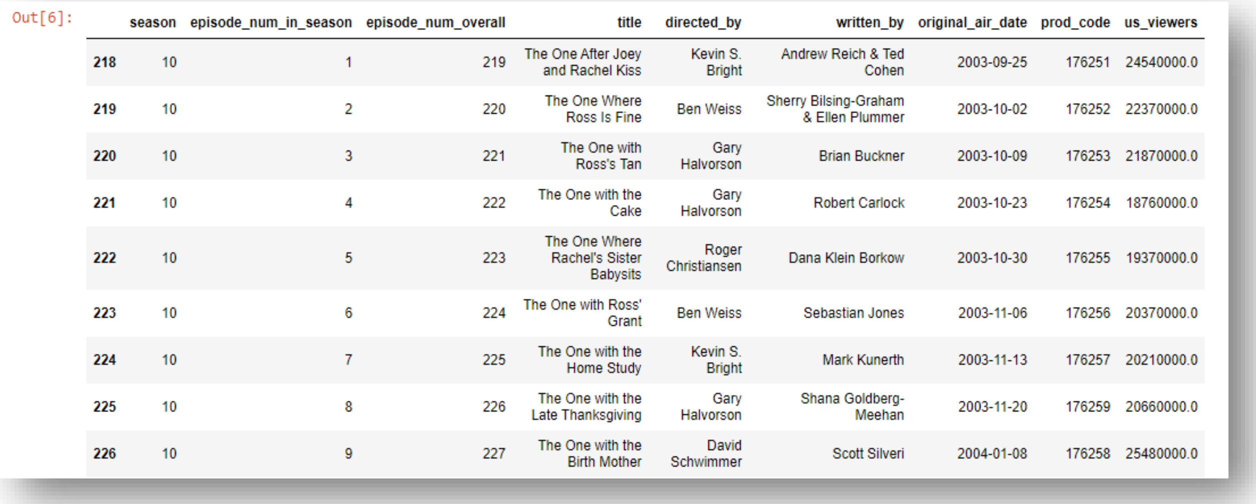
```
Episodes
 season
1      24
2      24
3      25
4      24
5      24
6      25
7      24
8      24
9      24
10     18
dtype: int64

IMDB
 season
1      24
2      24
3      25
4      24
5      24
6      25
7      24
8      24
9      24
10     17
dtype: int64
```

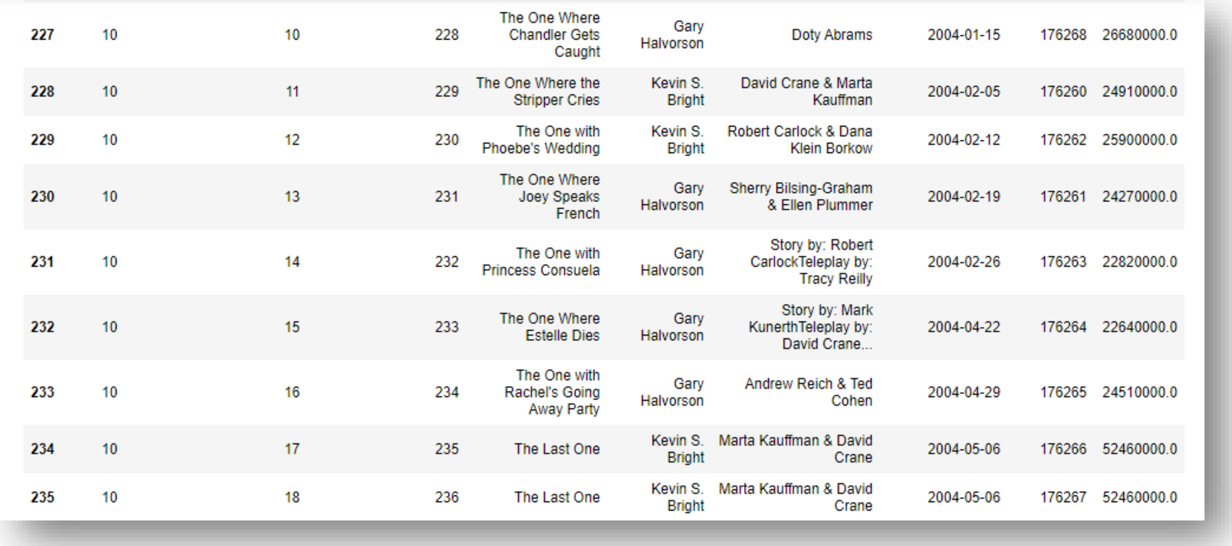
STEP NO: 6

Print all rows for all season in the episodes dataframe



```
In [6]: # Print all rows for season 10 in the episodes dataframe
        episodes[episodes['season']==10].head(20)
```



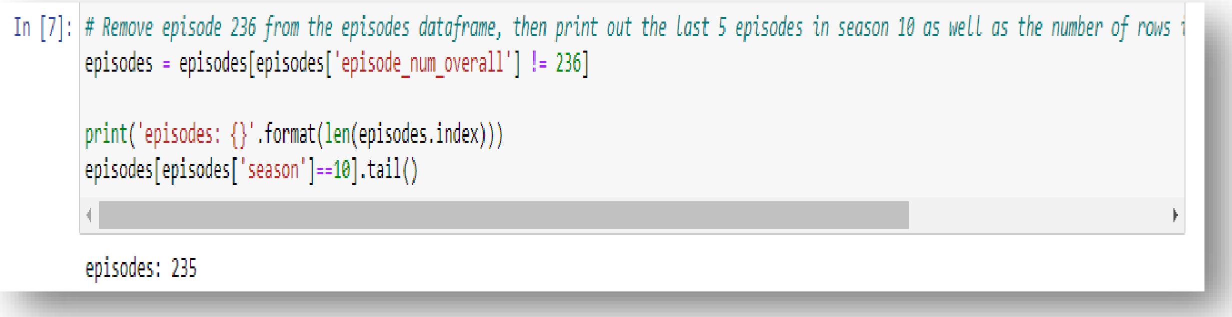


Out[6]:

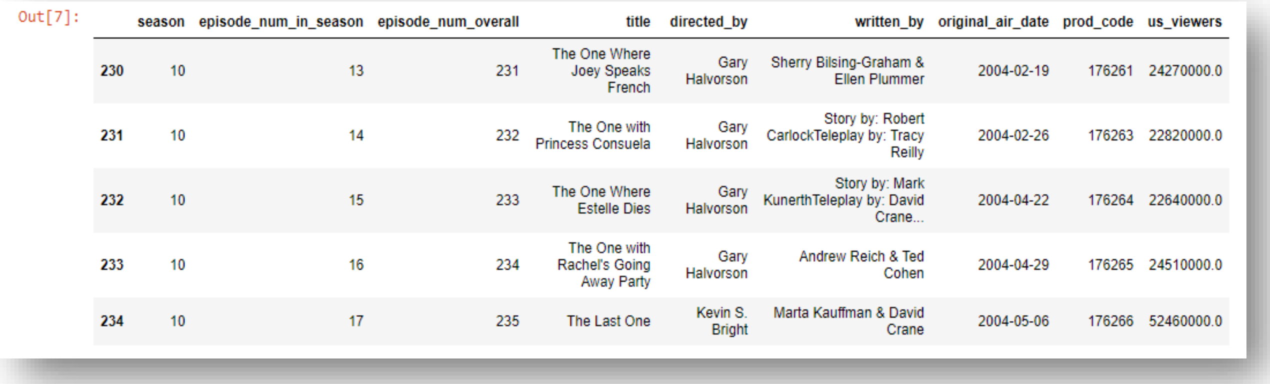
	season	episode_num_in_season	episode_num_overall	title	directed_by	written_by	original_air_date	prod_code	us_viewers
218	10	1	219	The One After Joey and Rachel Kiss	Kevin S. Bright	Andrew Reich & Ted Cohen	2003-09-25	176251	24540000.0
219	10	2	220	The One Where Ross Is Fine	Ben Weiss	Sherry Bilsing-Graham & Ellen Plummer	2003-10-02	176252	22370000.0
220	10	3	221	The One with Ross's Tan	Gary Halvorson	Brian Buckner	2003-10-09	176253	21870000.0
221	10	4	222	The One with the Cake	Gary Halvorson	Robert Carlock	2003-10-23	176254	18760000.0
222	10	5	223	The One Where Rachel's Sister Babysits	Roger Christiansen	Dana Klein Borkow	2003-10-30	176255	19370000.0
223	10	6	224	The One with Ross' Grant	Ben Weiss	Sebastian Jones	2003-11-06	176256	20370000.0
224	10	7	225	The One with the Home Study	Kevin S. Bright	Mark Kunerth	2003-11-13	176257	20210000.0
225	10	8	226	The One with the Late Thanksgiving	Gary Halvorson	Shana Goldberg-Meehan	2003-11-20	176259	20660000.0
226	10	9	227	The One with the Birth Mother	David Schwimmer	Scott Silveri	2004-01-08	176258	25480000.0
227	10	10	228	The One Where Chandler Gets Caught	Gary Halvorson	Doty Abrams	2004-01-15	176268	26680000.0
228	10	11	229	The One Where the Stripper Cries	Kevin S. Bright	David Crane & Marta Kauffman	2004-02-05	176260	24910000.0
229	10	12	230	The One with Phoebe's Wedding	Kevin S. Bright	Robert Carlock & Dana Klein Borkow	2004-02-12	176262	25900000.0
230	10	13	231	The One Where Joey Speaks French	Gary Halvorson	Sherry Bilsing-Graham & Ellen Plummer	2004-02-19	176261	24270000.0
231	10	14	232	The One with Princess Consuela	Gary Halvorson	Story by: Robert CarlockTeleplay by: Tracy Reilly	2004-02-26	176263	22820000.0
232	10	15	233	The One Where Estelle Dies	Gary Halvorson	Story by: Mark KunerthTeleplay by: David Crane...	2004-04-22	176264	22640000.0
233	10	16	234	The One with Rachel's Going Away Party	Gary Halvorson	Andrew Reich & Ted Cohen	2004-04-29	176265	24510000.0
234	10	17	235	The Last One	Kevin S. Bright	Marta Kauffman & David Crane	2004-05-06	176266	52460000.0
235	10	18	236	The Last One	Kevin S. Bright	Marta Kauffman & David Crane	2004-05-06	176267	52460000.0

STEP NO: 7

Removing any random episode from the dataframe in order to verify that it works or not



```
In [7]: # Remove episode 236 from the episodes dataframe, then print out the last 5 episodes in season 10 as well as the number of rows i
        episodes = episodes[episodes['episode_num_overall'] != 236]

        print('episodes: {}'.format(len(episodes.index)))
        episodes[episodes['season']==10].tail()

        episodes: 235
```

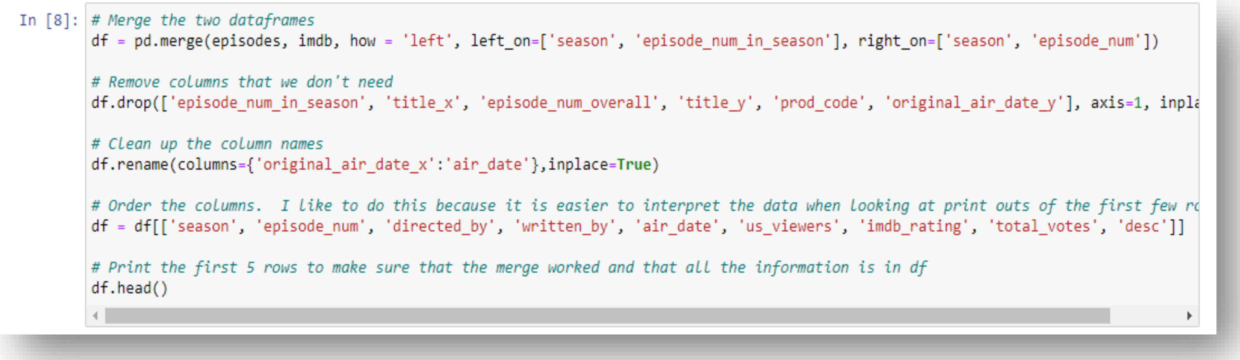


Out[7]:

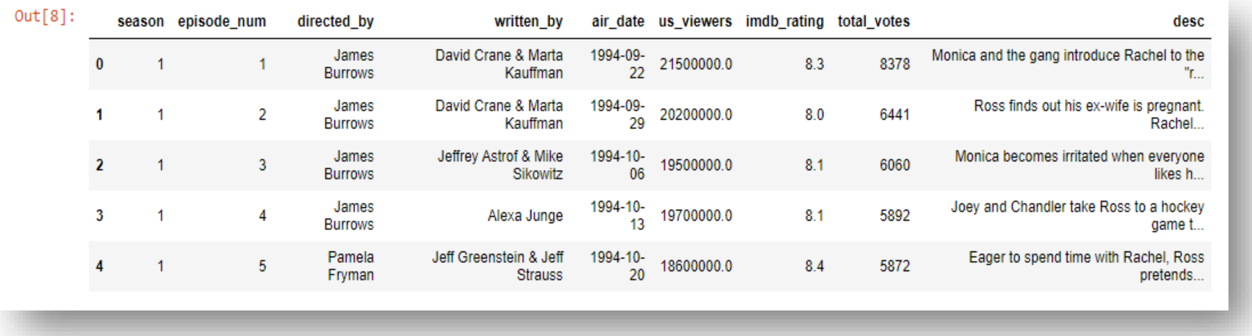
	season	episode_num_in_season	episode_num_overall	title	directed_by	written_by	original_air_date	prod_code	us_viewers
230	10	13	231	The One Where Joey Speaks French	Gary Halvorson	Sherry Bilsing-Graham & Ellen Plummer	2004-02-19	176261	24270000.0
231	10	14	232	The One with Princess Consuela	Gary Halvorson	Story by: Robert CarlockTeleplay by: Tracy Reilly	2004-02-26	176263	22820000.0
232	10	15	233	The One Where Estelle Dies	Gary Halvorson	Story by: Mark KunerthTeleplay by: David Crane...	2004-04-22	176264	22640000.0
233	10	16	234	The One with Rachel's Going Away Party	Gary Halvorson	Andrew Reich & Ted Cohen	2004-04-29	176265	24510000.0
234	10	17	235	The Last One	Kevin S. Bright	Marta Kauffman & David Crane	2004-05-06	176266	52460000.0

STEP NO: 8

Sorting and simplifying our data as much as possible



```
In [8]: # Merge the two dataframes
        df = pd.merge(episodes, imdb, how = 'left', left_on=['season', 'episode_num_in_season'], right_on=['season', 'episode_num'])

        # Remove columns that we don't need
        df.drop(['episode_num_in_season', 'title_x', 'episode_num_overall', 'title_y', 'prod_code', 'original_air_date_y'], axis=1, inpla

        # Clean up the column names
        df.rename(columns={'original_air_date_x':'air_date'},inplace=True)

        # Order the columns.  I like to do this because it is easier to interpret the data when looking at print outs of the first few ro
        df = df[['season', 'episode_num', 'directed_by', 'written_by', 'air_date', 'us_viewers', 'imdb_rating', 'total_votes', 'desc']]

        # Print the first 5 rows to make sure that the merge worked and that all the information is in df
        df.head()
```

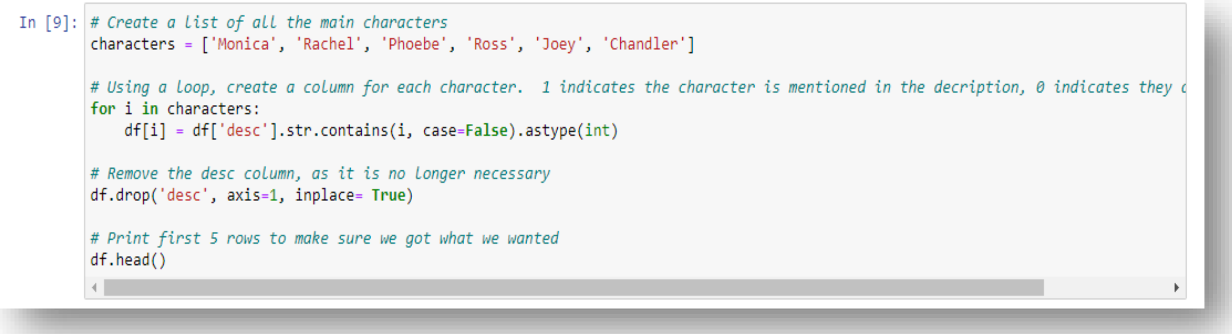


Out[8]:

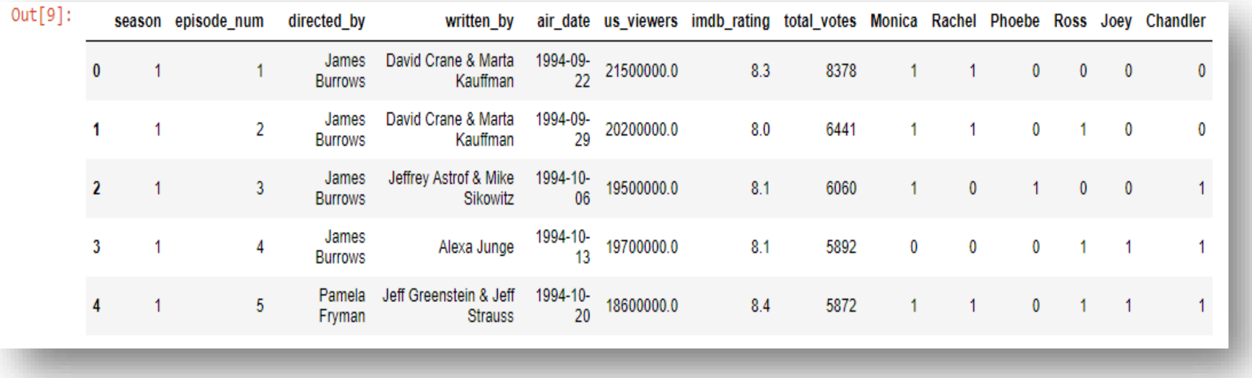
	season	episode_num	directed_by	written_by	air_date	us_viewers	imdb_rating	total_votes	desc
0	1	1	James Burrows	David Crane & Marta Kauffman	1994-09-22	21500000.0	8.3	8378	Monica and the gang introduce Rachel to the "r...
1	1	2	James Burrows	David Crane & Marta Kauffman	1994-09-29	20200000.0	8.0	6441	Ross finds out his ex-wife is pregnant. Rachel...
2	1	3	James Burrows	Jeffrey Astrof & Mike Sikowitz	1994-10-06	19500000.0	8.1	6060	Monica becomes irritated when everyone likes h...
3	1	4	James Burrows	Alexa Junge	1994-10-13	19700000.0	8.1	5892	Joey and Chandler take Ross to a hockey game t...
4	1	5	Pamela Fryman	Jeff Greenstein & Jeff Strauss	1994-10-20	18600000.0	8.4	5872	Eager to spend time with Rachel, Ross pretends...

STEP NO: 9

Creating the list of all main characters



```
In [9]: # Create a list of all the main characters
        characters = ['Monica', 'Rachel', 'Phoebe', 'Ross', 'Joey', 'Chandler']

        # Using a loop, create a column for each character.  1 indicates the character is mentioned in the decription, 0 indicates they
        for i in characters:
            df[i] = df['desc'].str.contains(i, case=False).astype(int)

        # Remove the desc column, as it is no longer necessary
        df.drop('desc', axis=1, inplace= True)

        # Print first 5 rows to make sure we got what we wanted
        df.head()
```

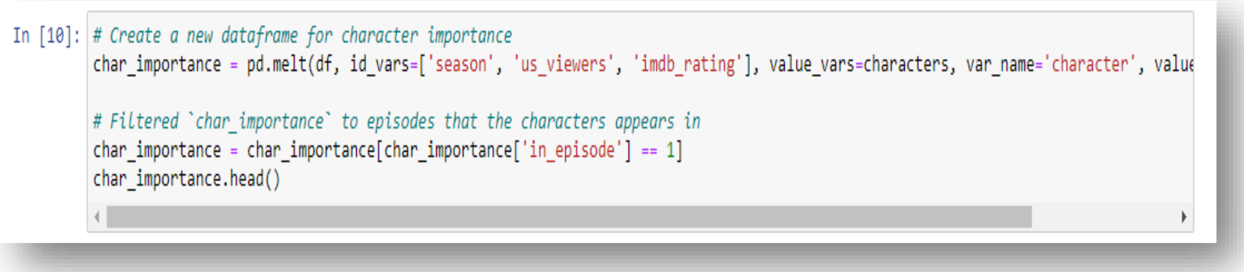


Out[9]:

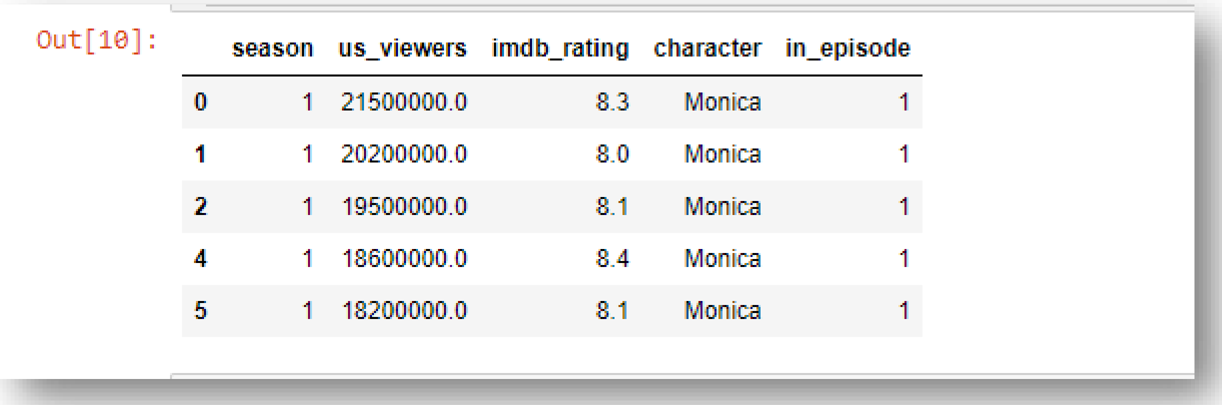
	season	episode_num	directed_by	written_by	air_date	us_viewers	imdb_rating	total_votes	Monica	Rachel	Phoebe	Ross	Joey	Chandler
0	1	1	James Burrows	David Crane & Marta Kauffman	1994-09-22	21500000.0	8.3	8378	1	1	0	0	0	0
1	1	2	James Burrows	David Crane & Marta Kauffman	1994-09-29	20200000.0	8.0	6441	1	1	0	1	0	0
2	1	3	James Burrows	Jeffrey Astrof & Mike Sikowitz	1994-10-06	19500000.0	8.1	6060	1	0	1	0	0	1
3	1	4	James Burrows	Alexa Junge	1994-10-13	19700000.0	8.1	5892	0	0	0	1	1	1
4	1	5	Pamela Fryman	Jeff Greenstein & Jeff Strauss	1994-10-20	18600000.0	8.4	5872	1	1	0	1	1	1

STEP NO: 10

Creating a new data frame for character importance



```
In [10]: # Create a new dataframe for character importance
         char_importance = pd.melt(df, id_vars=['season', 'us_viewers', 'imdb_rating'], value_vars=characters, var_name='character', value

         # Filtered `char_importance` to episodes that the characters appears in
         char_importance = char_importance[char_importance['in_episode'] == 1]
         char_importance.head()
```

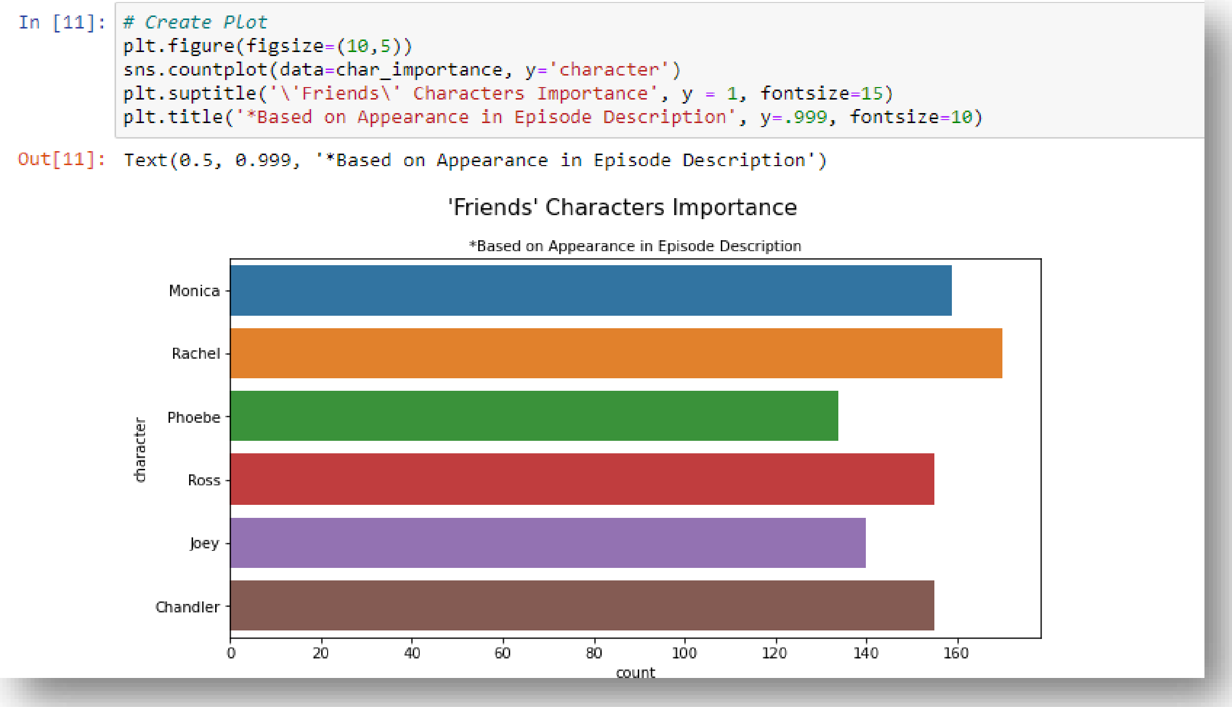


Out[10]:

	season	us_viewers	imdb_rating	character	in_episode
0	1	21500000.0	8.3	Monica	1
1	1	20200000.0	8.0	Monica	1
2	1	19500000.0	8.1	Monica	1
4	1	18600000.0	8.4	Monica	1
5	1	18200000.0	8.1	Monica	1

STEP NO: 11

Creating Plot : Character Importance

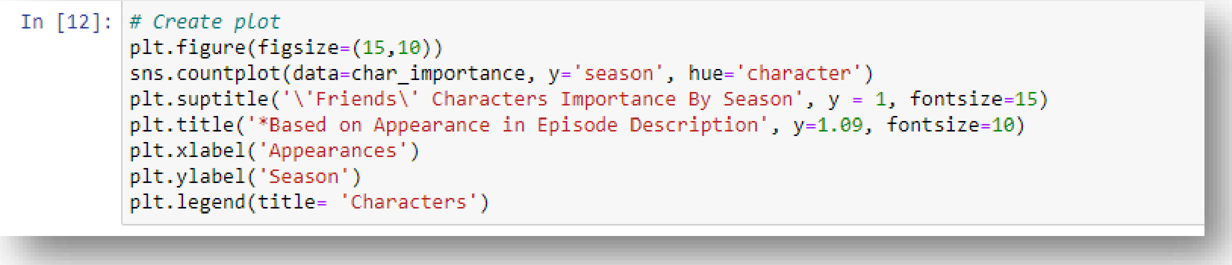


```
In [11]: # Create Plot
         plt.figure(figsize=(10,5))
         sns.countplot(data=char_importance, y='character')
         plt.suptitle('\'Friends\' Characters Importance', y = 1, fontsize=15)
         plt.title('*Based on Appearance in Episode Description', y=.999, fontsize=10)

Out[11]: Text(0.5, 0.999, '*Based on Appearance in Episode Description')
```

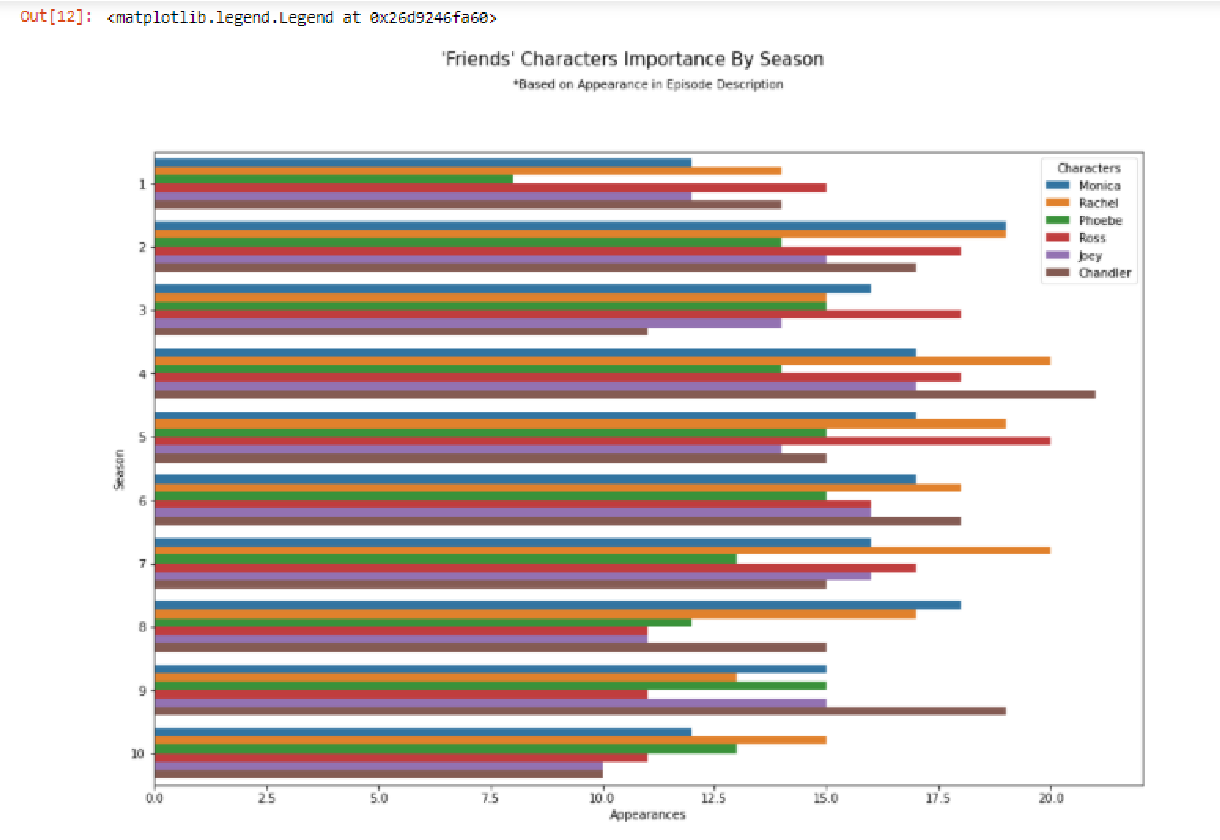
'Friends' Characters Importance
*Based on Appearance in Episode Description
character
Monica
Rachel
Phoebe
Ross
Joey
Chandler
0
20
40
60
80
100
120
140
160
count

STEP NO: 12

Creating Plot : Character Importance By Seasons



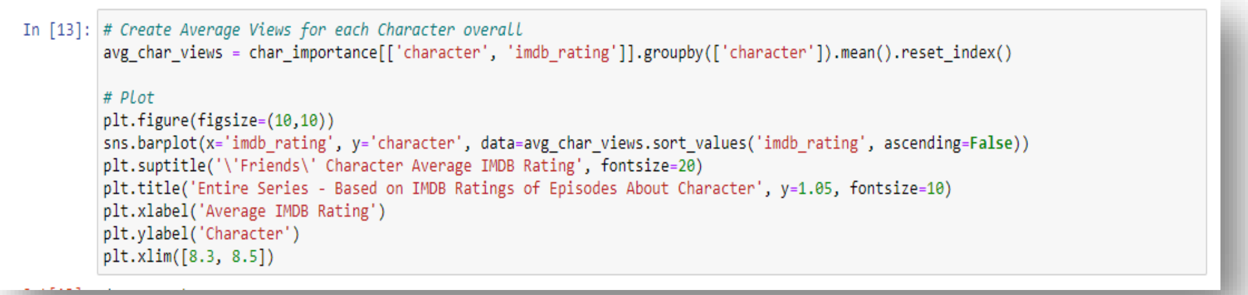
```
In [12]: # Create plot
         plt.figure(figsize=(15,10))
         sns.countplot(data=char_importance, y='season', hue='character')
         plt.suptitle('\'Friends\' Characters Importance By Season', y = 1, fontsize=15)
         plt.title('*Based on Appearance in Episode Description', y=1.09, fontsize=10)
         plt.xlabel('Appearances')
         plt.ylabel('Season')
         plt.legend(title= 'Characters')
```



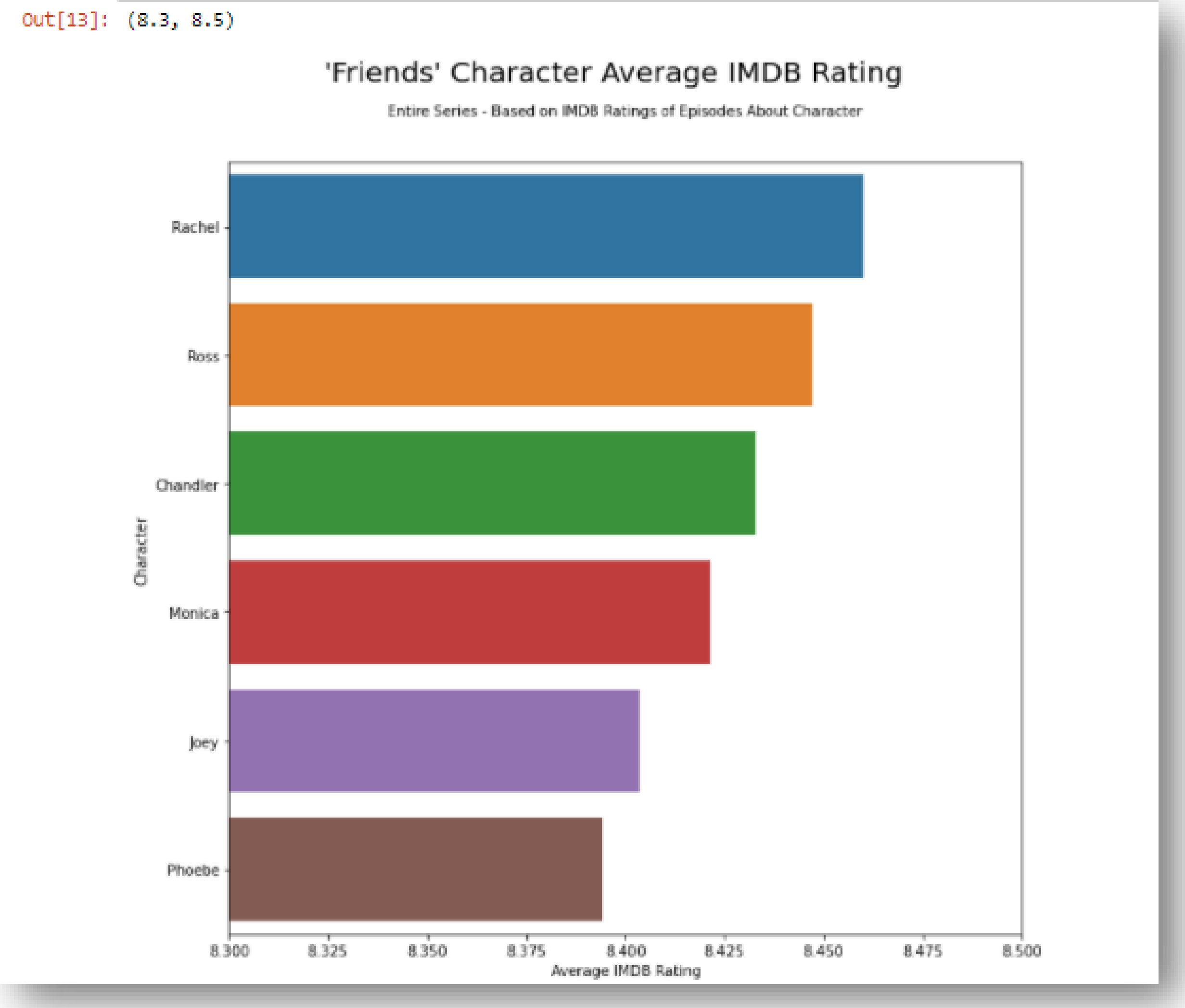
```
Out[12]: <matplotlib.legend.Legend at 0x26d9246fa60>
```

STEP NO: 13

Characters average IMDB rating



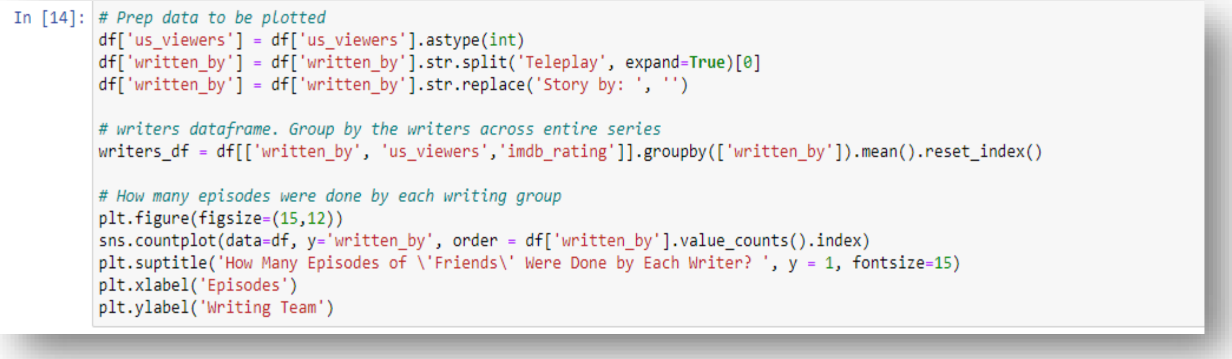
```
In [13]: # Create Average Views for each Character overall
         avg_char_views = char_importance[['character', 'imdb_rating']].groupby(['character']).mean().reset_index()

         # Plot
         plt.figure(figsize=(10,10))
         sns.barplot(x='imdb_rating', y='character', data=avg_char_views.sort_values('imdb_rating', ascending=False))
         plt.suptitle('\'Friends\' Character Average IMDB Rating', fontsize=20)
         plt.title('Entire Series - Based on IMDB Ratings of Episodes About Character', y=1.05, fontsize=10)
         plt.xlabel('Average IMDB Rating')
         plt.ylabel('Character')
         plt.xlim([8.3, 8.5])
```



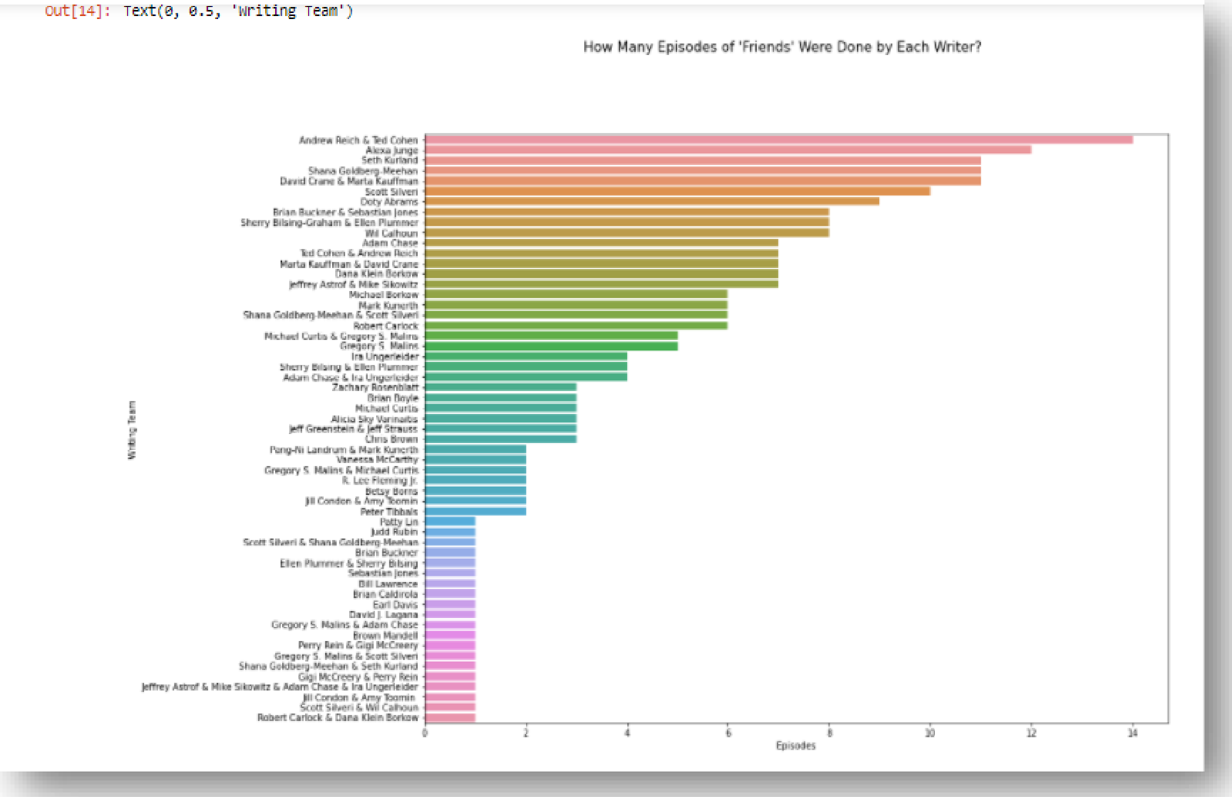
```
Out[13]: (8.3, 8.5)
```

STEP NO: 14

How many episodes were written by each writer?



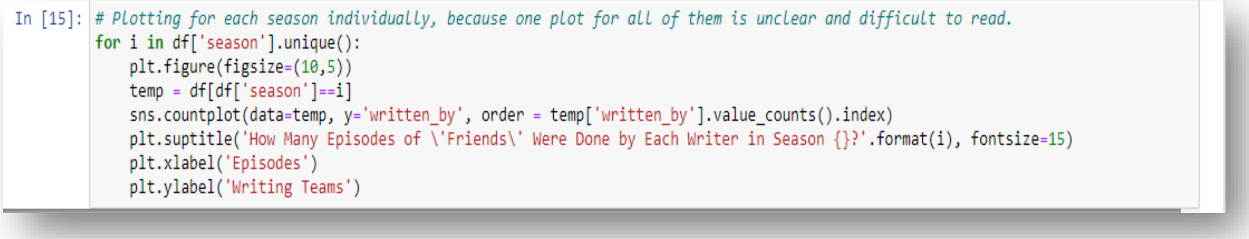
```
In [14]: # Prep data to be plotted
         df['us_viewers'] = df['us_viewers'].astype(int)
         df['written_by'] = df['written_by'].str.split('Teleplay', expand=True)[0]
         df['written_by'] = df['written_by'].str.replace('Story by: ', '')

         # writers dataframe. Group by the writers across entire series
         writers_df = df[['written_by', 'us_viewers','imdb_rating']].groupby(['written_by']).mean().reset_index()

         # How many episodes were done by each writing group
         plt.figure(figsize=(15,12))
         sns.countplot(data=df, y='written_by', order = df['written_by'].value_counts().index)
         plt.suptitle('How Many Episodes of \'Friends\' Were Done by Each Writer? ', y = 1, fontsize=15)
         plt.xlabel('Episodes')
         plt.ylabel('Writing Team')
```



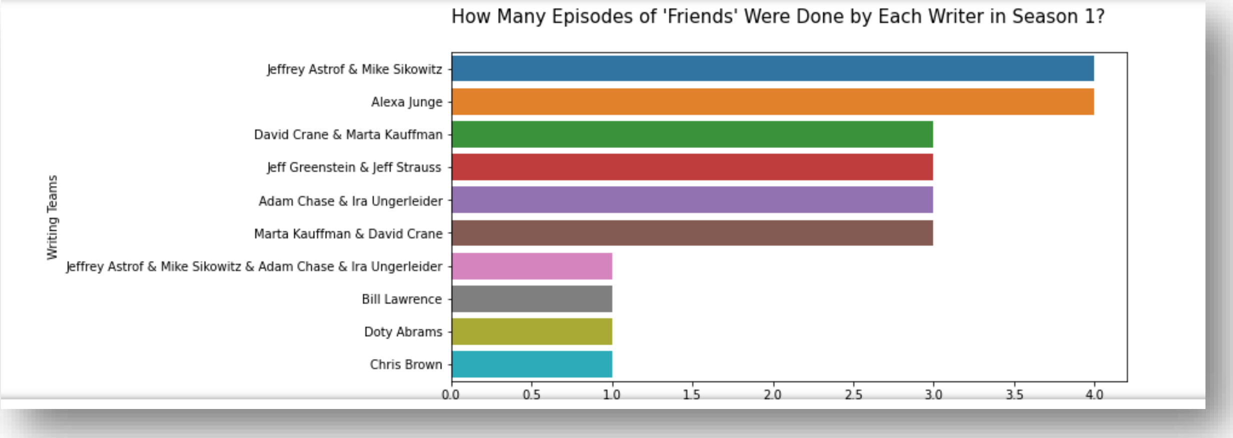
```
Out[14]: Text(0, 0.5, 'Writing Team')
```

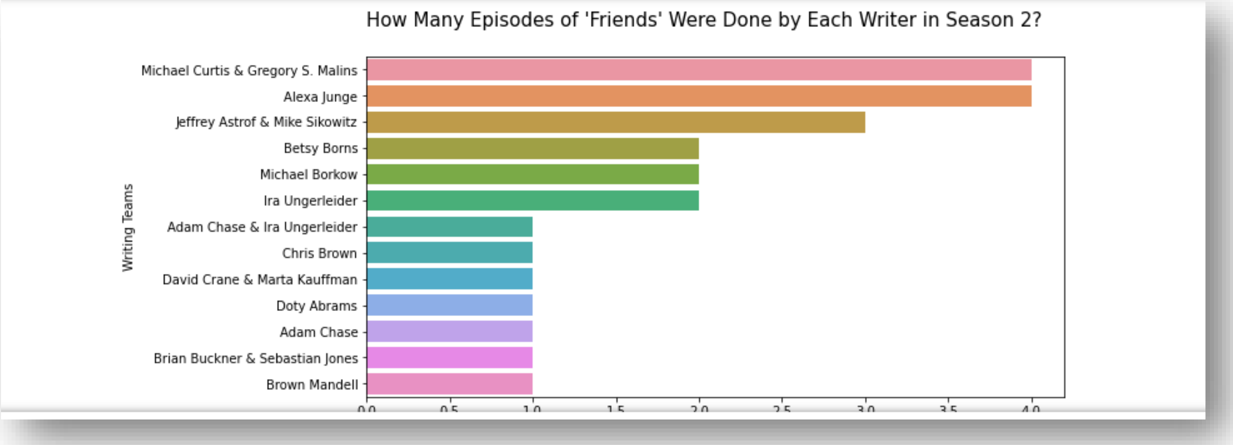
STEP NO: 15

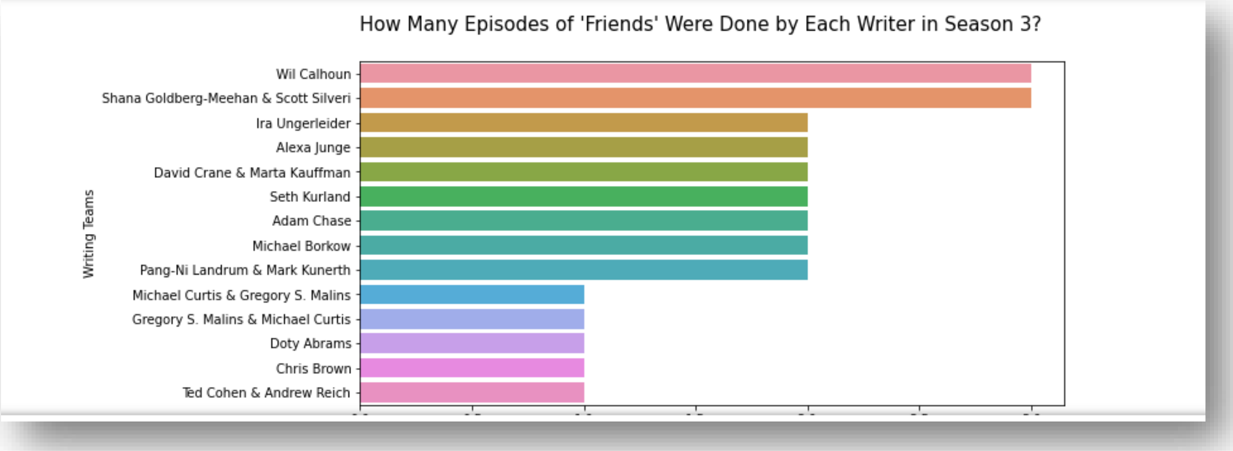
Plotting for each season individually

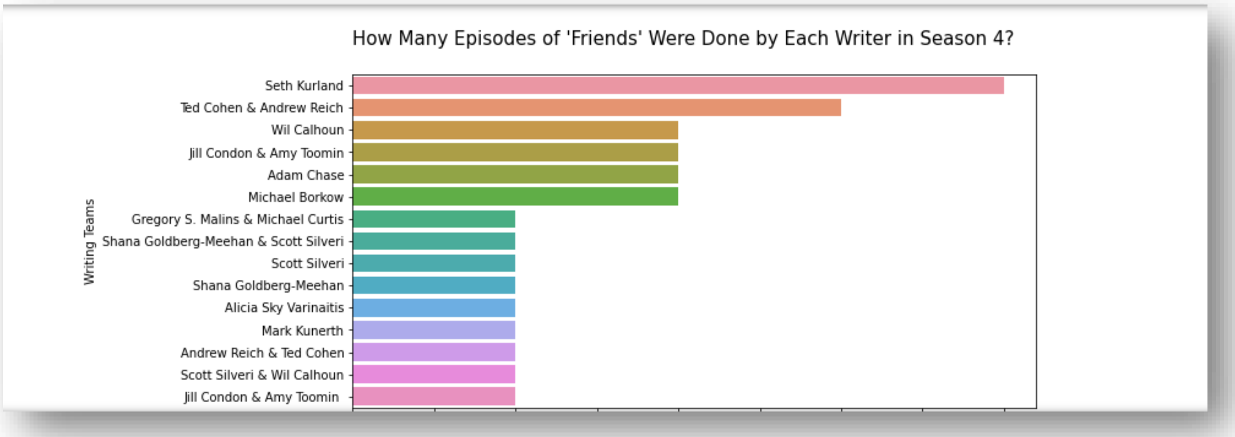


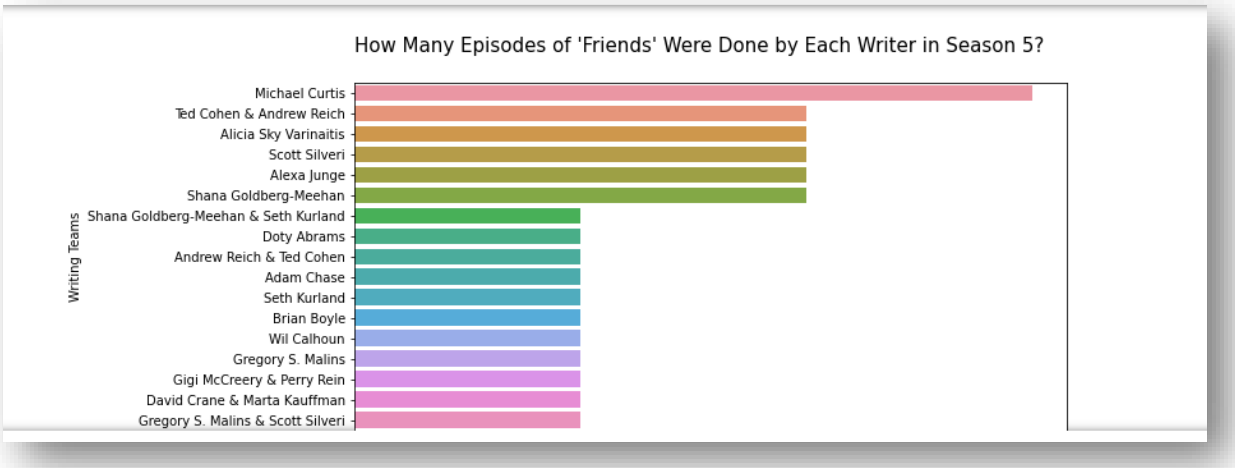
```
In [15]: # Plotting for each season individually, because one plot for all of them is unclear and difficult to read.
         for i in df['season'].unique():
             plt.figure(figsize=(10,5))
             temp = df[df['season']==i]
             sns.countplot(data=temp, y='written_by', order = temp['written_by'].value_counts().index)
             plt.suptitle('How Many Episodes of \'Friends\' Were Done by Each Writer in Season {}?'.format(i), fontsize=15)
             plt.xlabel('Episodes')
             plt.ylabel('Writing Teams')
```

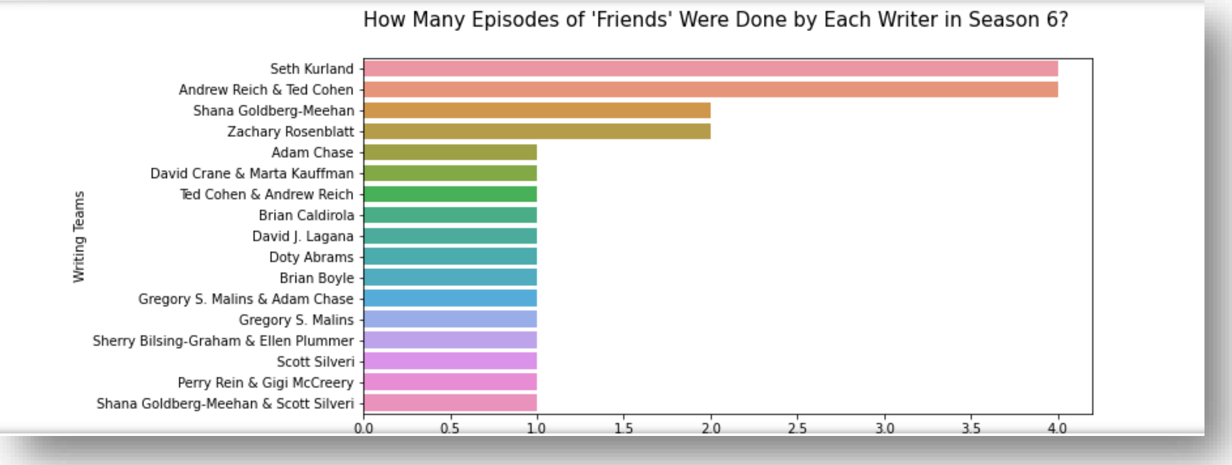

How Many Episodes of 'Friends' Were Done by Each Writer in Season 1?
Writing Teams
Jeffrey Astrof & Mike Sikowitz
Alexa Junge
David Crane & Marta Kauffman
Jeff Greenstein & Jeff Strauss
Adam Chase & Ira Ungerleider
Marta Kauffman & David Crane
Jeffrey Astrof & Mike Sikowitz & Adam Chase & Ira Ungerleider
Bill Lawrence
Doty Abrams
Chris Brown
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

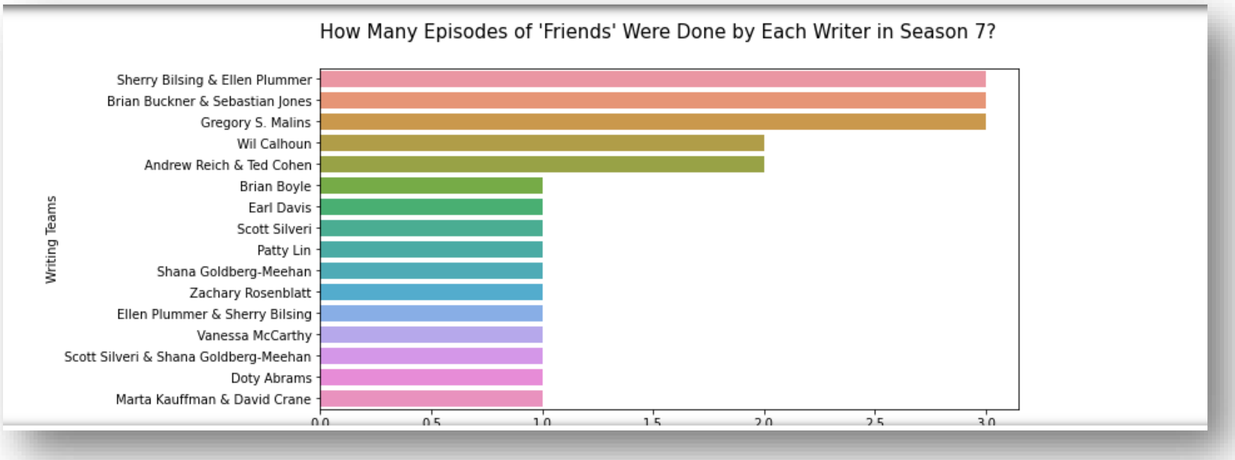

How Many Episodes of 'Friends' Were Done by Each Writer in Season 2?
Writing Teams
Michael Curtis & Gregory S. Malins
Alexa Junge
Jeffrey Astrof & Mike Sikowitz
Betsy Borns
Michael Borkow
Ira Ungerleider
Adam Chase & Ira Ungerleider
Chris Brown
David Crane & Marta Kauffman
Doty Abrams
Adam Chase
Brian Buckner & Sebastian Jones
Brown Mandell
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

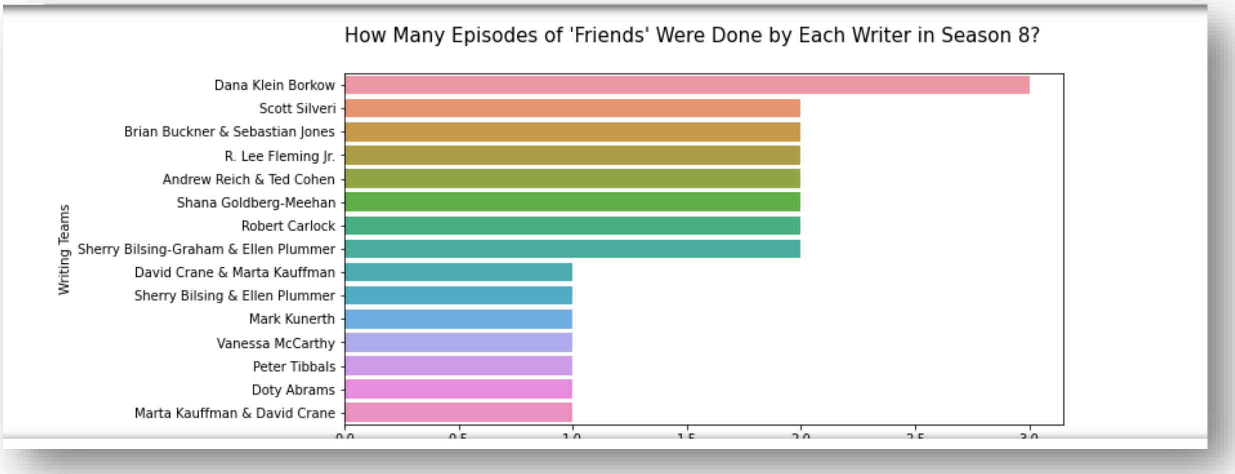

How Many Episodes of 'Friends' Were Done by Each Writer in Season 3?
Writing Teams
Wil Calhoun
Shana Goldberg-Meehan & Scott Silveri
Ira Ungerleider
Alexa Junge
David Crane & Marta Kauffman
Seth Kurland
Adam Chase
Michael Borkow
Pang-Ni Landrum & Mark Kunerth
Michael Curtis & Gregory S. Malins
Gregory S. Malins & Michael Curtis
Doty Abrams
Chris Brown
Ted Cohen & Andrew Reich

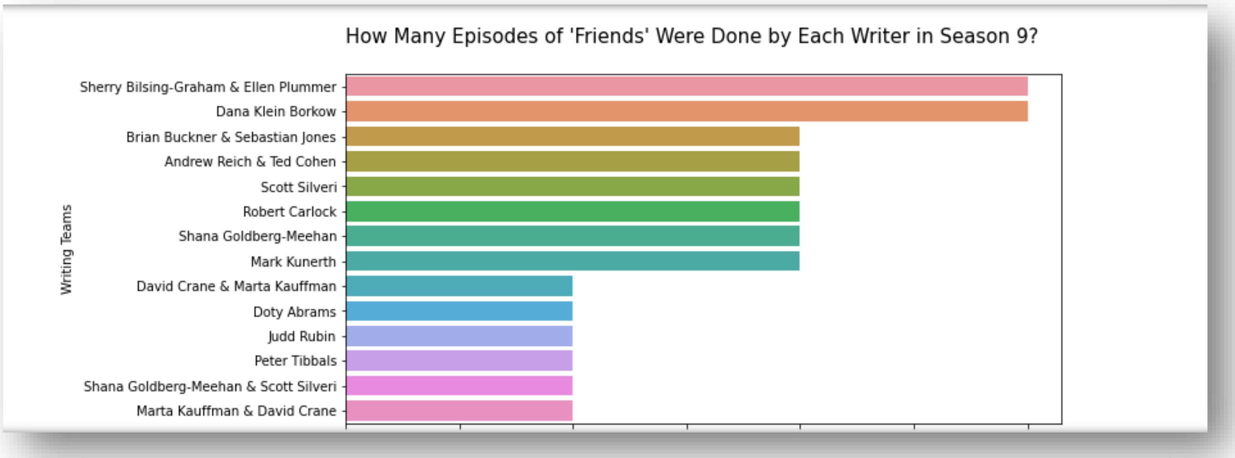

How Many Episodes of 'Friends' Were Done by Each Writer in Season 4?
Writing Teams
Seth Kurland
Ted Cohen & Andrew Reich
Wil Calhoun
Jill Condon & Amy Toomin
Adam Chase
Michael Borkow
Gregory S. Malins & Michael Curtis
Shana Goldberg-Meehan & Scott Silveri
Scott Silveri
Shana Goldberg-Meehan
Alicia Sky Varinaitis
Mark Kunerth
Andrew Reich & Ted Cohen
Scott Silveri & Wil Calhoun
Jill Condon & Amy Toomin

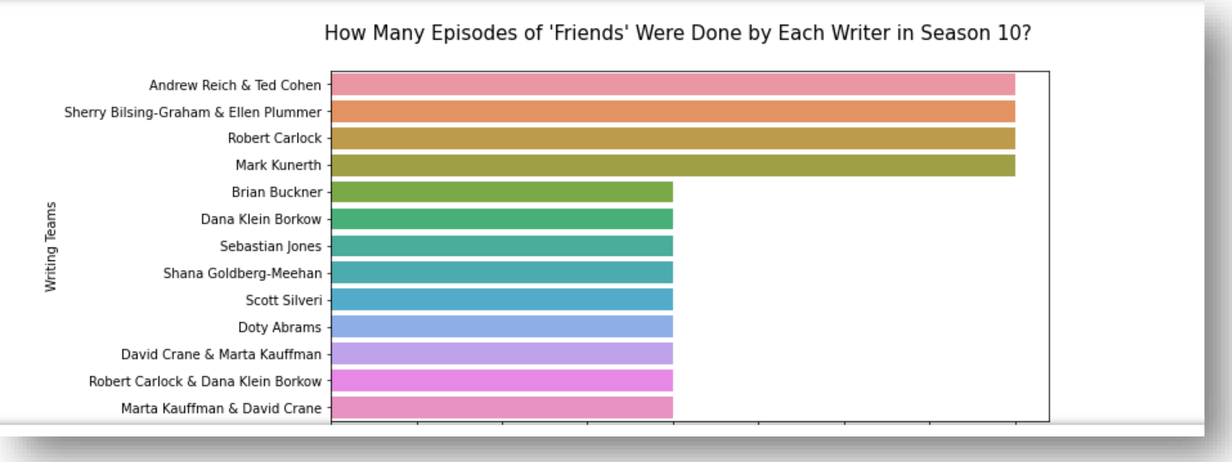

How Many Episodes of 'Friends' Were Done by Each Writer in Season 5?
Writing Teams
Michael Curtis
Ted Cohen & Andrew Reich
Alicia Sky Varinaitis
Scott Silveri
Alexa Junge
Shana Goldberg-Meehan
Shana Goldberg-Meehan & Seth Kurland
Doty Abrams
Andrew Reich & Ted Cohen
Adam Chase
Seth Kurland
Brian Boyle
Wil Calhoun
Gregory S. Malins
Gigi McCreery & Perry Rein
David Crane & Marta Kauffman
Gregory S. Malins & Scott Silveri


How Many Episodes of 'Friends' Were Done by Each Writer in Season 6?
Writing Teams
Seth Kurland
Andrew Reich & Ted Cohen
Shana Goldberg-Meehan
Zachary Rosenblatt
Adam Chase
David Crane & Marta Kauffman
Ted Cohen & Andrew Reich
Brian Caldirola
David J. Lagana
Doty Abrams
Brian Boyle
Gregory S. Malins & Adam Chase
Gregory S. Malins
Sherry Bilsing-Graham & Ellen Plummer
Scott Silveri
Perry Rein & Gigi McCreery
Shana Goldberg-Meehan & Scott Silveri
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


How Many Episodes of 'Friends' Were Done by Each Writer in Season 7?
Writing Teams
Sherry Bilsing & Ellen Plummer
Brian Buckner & Sebastian Jones
Gregory S. Malins
Wil Calhoun
Andrew Reich & Ted Cohen
Brian Boyle
Earl Davis
Scott Silveri
Patty Lin
Shana Goldberg-Meehan
Zachary Rosenblatt
Ellen Plummer & Sherry Bilsing
Vanessa McCarthy
Scott Silveri & Shana Goldberg-Meehan
Doty Abrams
Marta Kauffman & David Crane
0.0
0.5
1.0
1.5
2.0
2.5
3.0

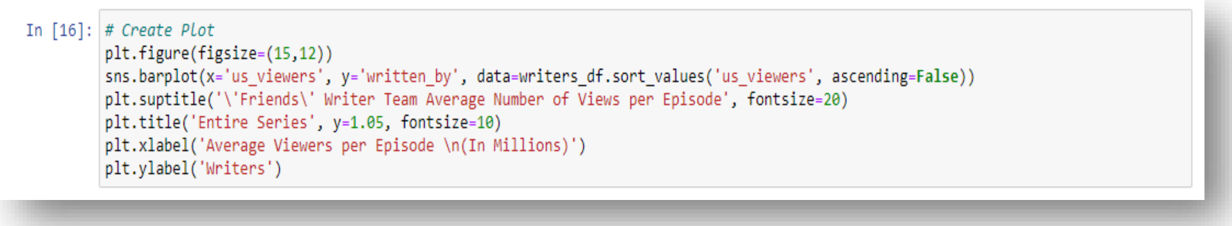

How Many Episodes of 'Friends' Were Done by Each Writer in Season 8?
Writing Teams
Dana Klein Borkow
Scott Silveri
Brian Buckner & Sebastian Jones
R. Lee Fleming Jr.
Andrew Reich & Ted Cohen
Shana Goldberg-Meehan
Robert Carlock
Sherry Bilsing-Graham & Ellen Plummer
David Crane & Marta Kauffman
Sherry Bilsing & Ellen Plummer
Mark Kunerth
Vanessa McCarthy
Peter Tibbals
Doty Abrams
Marta Kauffman & David Crane
0.0
0.5
1.0
1.5
2.0
2.5
3.0


How Many Episodes of 'Friends' Were Done by Each Writer in Season 9?
Writing Teams
Sherry Bilsing-Graham & Ellen Plummer
Dana Klein Borkow
Brian Buckner & Sebastian Jones
Andrew Reich & Ted Cohen
Scott Silveri
Robert Carlock
Shana Goldberg-Meehan
Mark Kunerth
David Crane & Marta Kauffman
Doty Abrams
Judd Rubin
Peter Tibbals
Shana Goldberg-Meehan & Scott Silveri
Marta Kauffman & David Crane

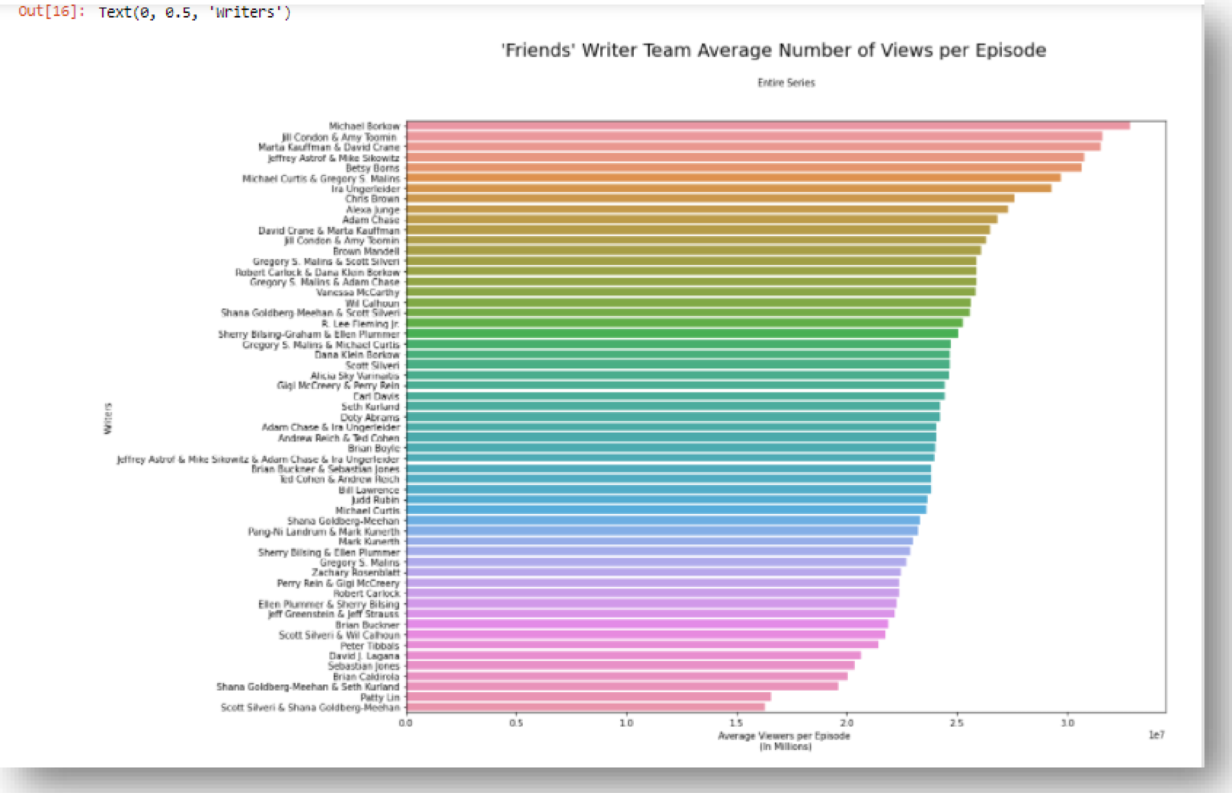


STEP NO: 16

Writer Team average number of views per episodes



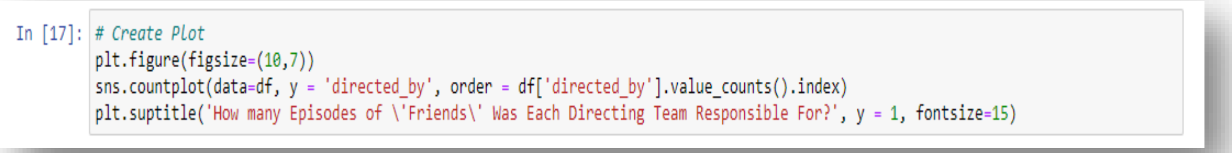
```
In [16]: # Create Plot
         plt.figure(figsize=(15,12))
         sns.barplot(x='us_viewers', y='written_by', data=writers_df.sort_values('us_viewers', ascending=False))
         plt.suptitle('\'Friends\' Writer Team Average Number of Views per Episode', fontsize=20)
         plt.title('Entire Series', y=1.05, fontsize=10)
         plt.xlabel('Average Viewers per Episode \n(In Millions)')
         plt.ylabel('Writers')
```



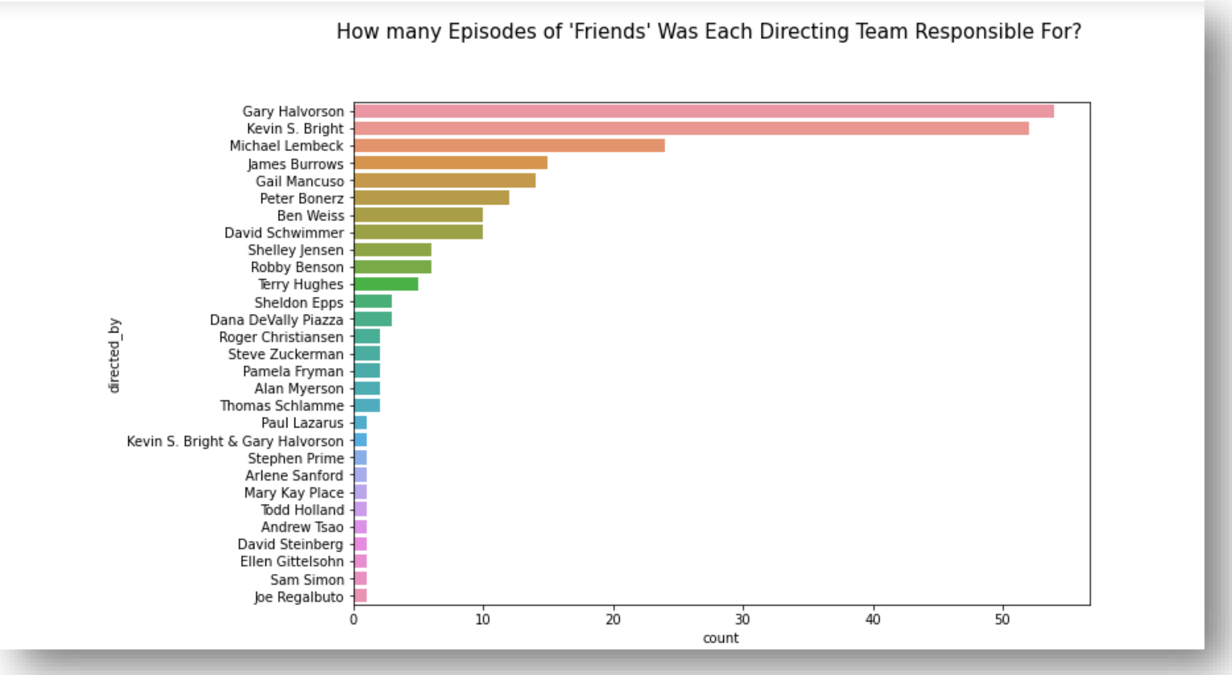
```
Out[16]: Text(0, 0.5, 'Writers')
```

STEP NO: 17

How many episodes of FRIENDS was each directing team Responsible for

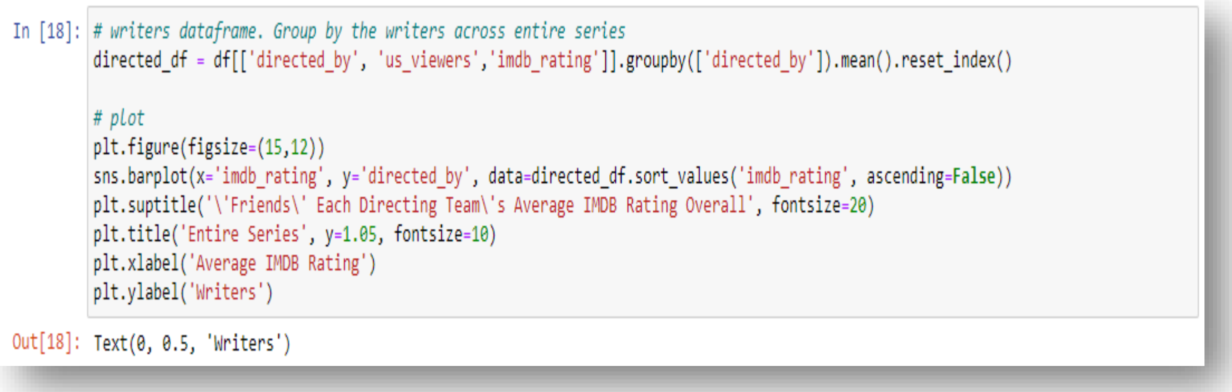


```
In [17]: # Create Plot
         plt.figure(figsize=(10,7))
         sns.countplot(data=df, y = 'directed_by', order = df['directed_by'].value_counts().index)
         plt.suptitle('How many Episodes of \'Friends\' Was Each Directing Team Responsible For?', y = 1, fontsize=15)
```

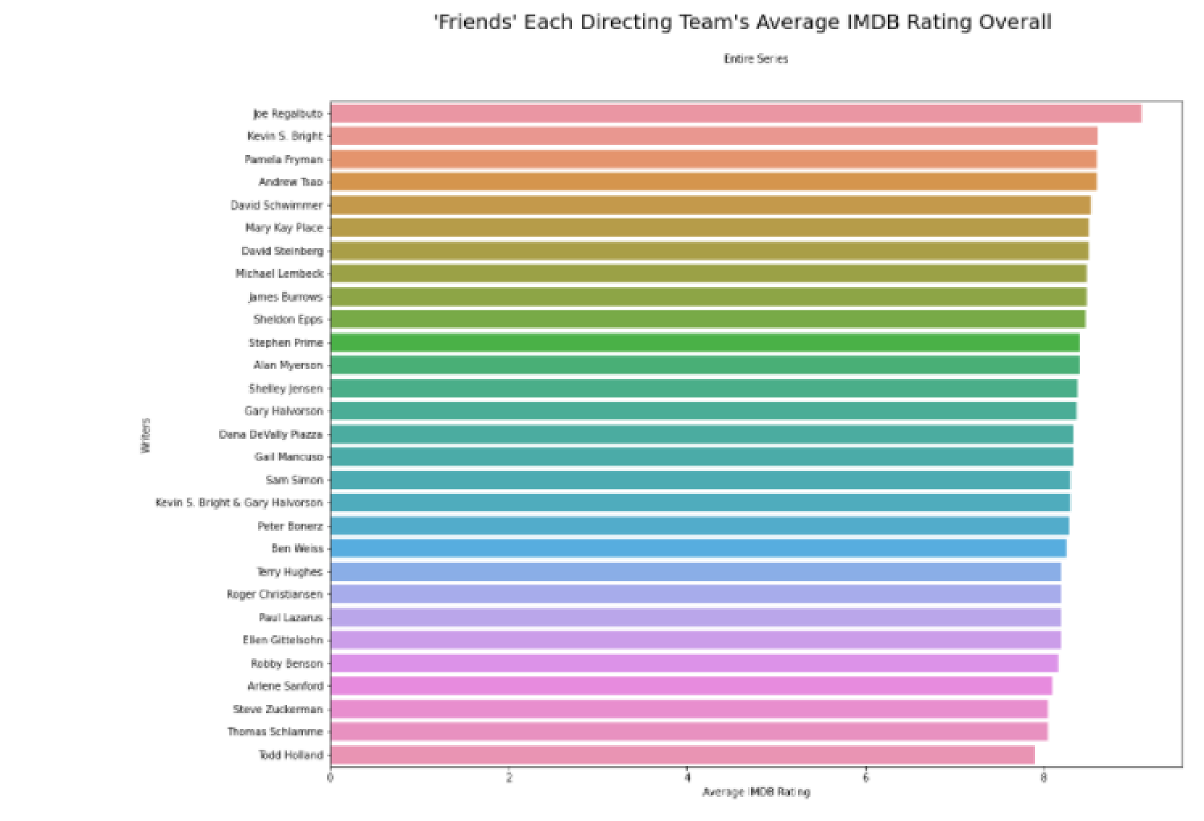


STEP NO: 18

Writer's dataframe

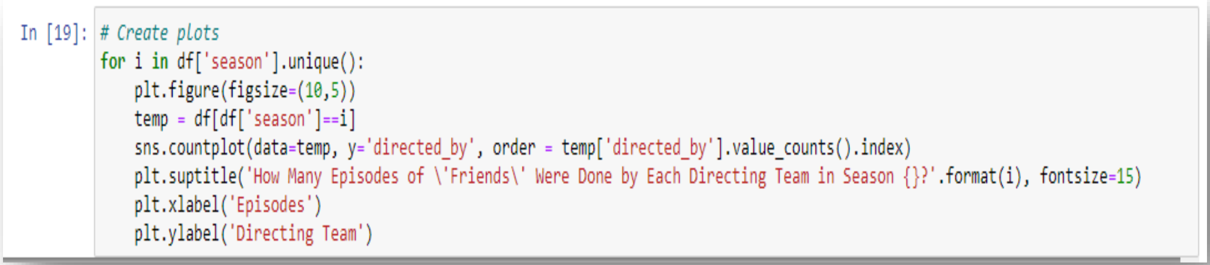


```
In [18]: # writers dataframe. Group by the writers across entire series
         directed_df = df[['directed_by', 'us_viewers','imdb_rating']].groupby(['directed_by']).mean().reset_index()

         # plot
         plt.figure(figsize=(15,12))
         sns.barplot(x='imdb_rating', y='directed_by', data=directed_df.sort_values('imdb_rating', ascending=False))
         plt.suptitle('\'Friends\' Each Directing Team\'s Average IMDB Rating Overall', fontsize=20)
         plt.title('Entire Series', y=1.05, fontsize=10)
         plt.xlabel('Average IMDB Rating')
         plt.ylabel('Writers')

Out[18]: Text(0, 0.5, 'Writers')
```

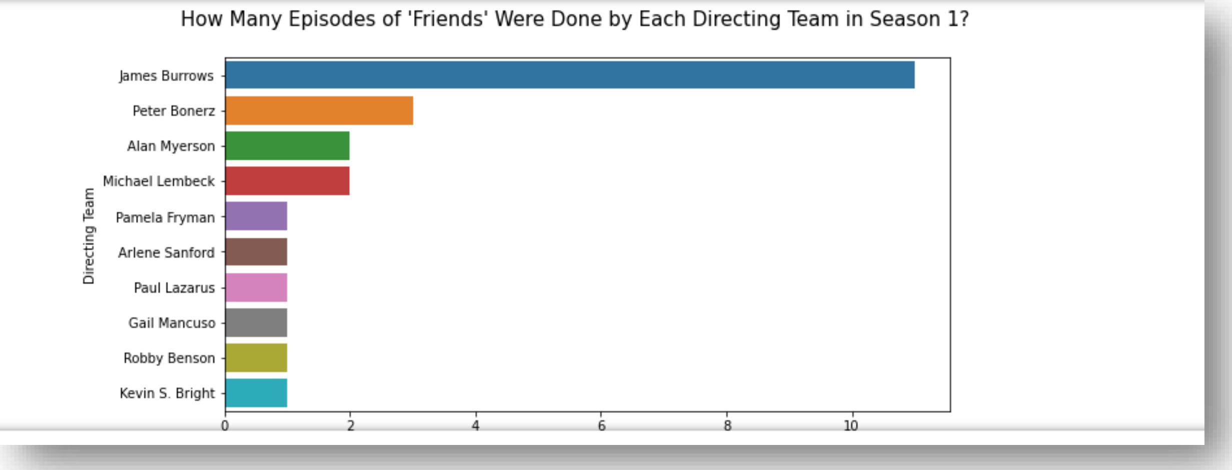


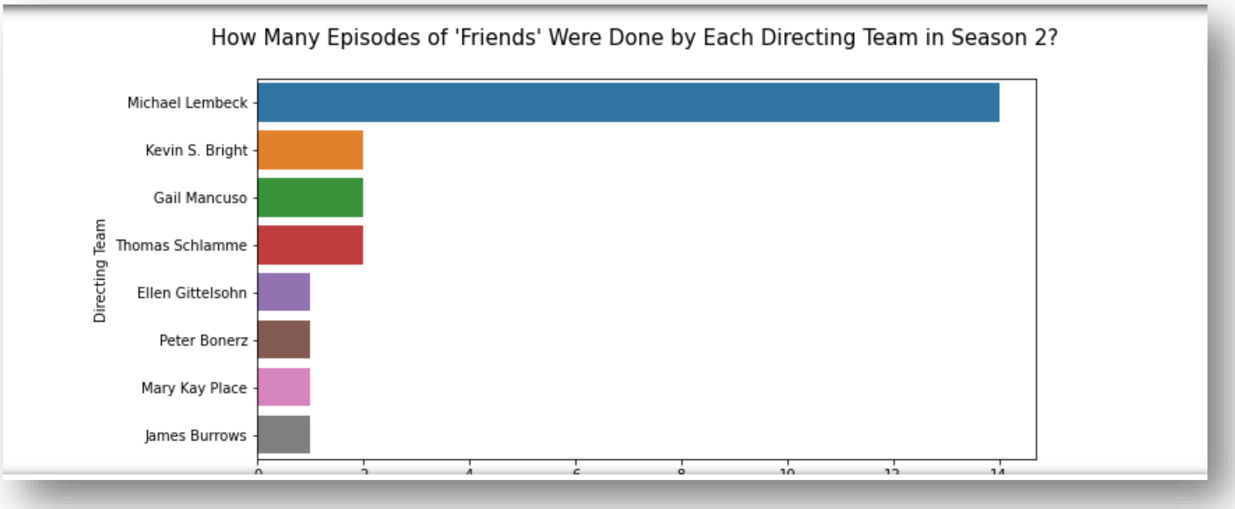
STEP NO: 19

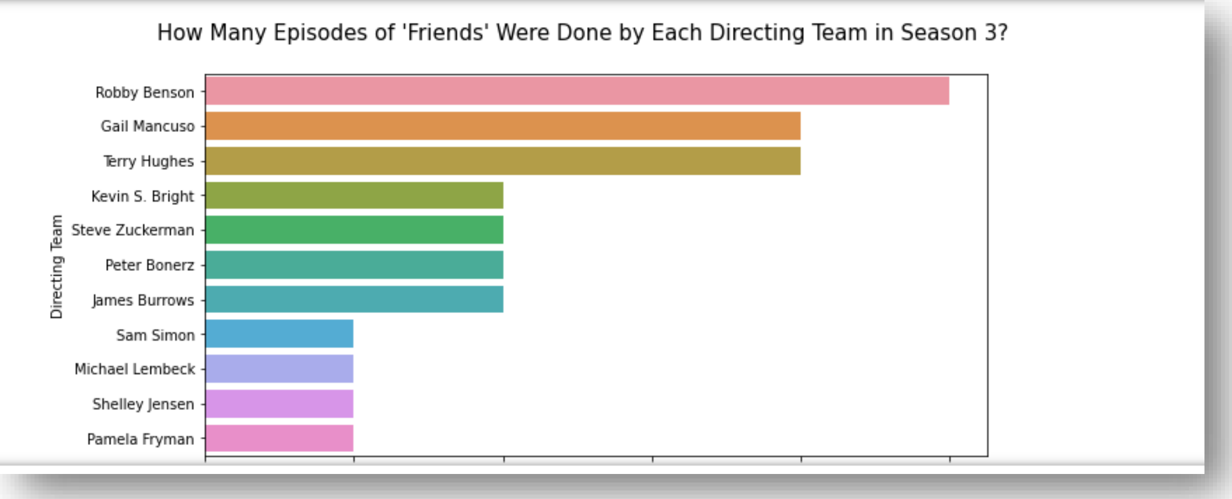
Plotting directing team of all seasons individually

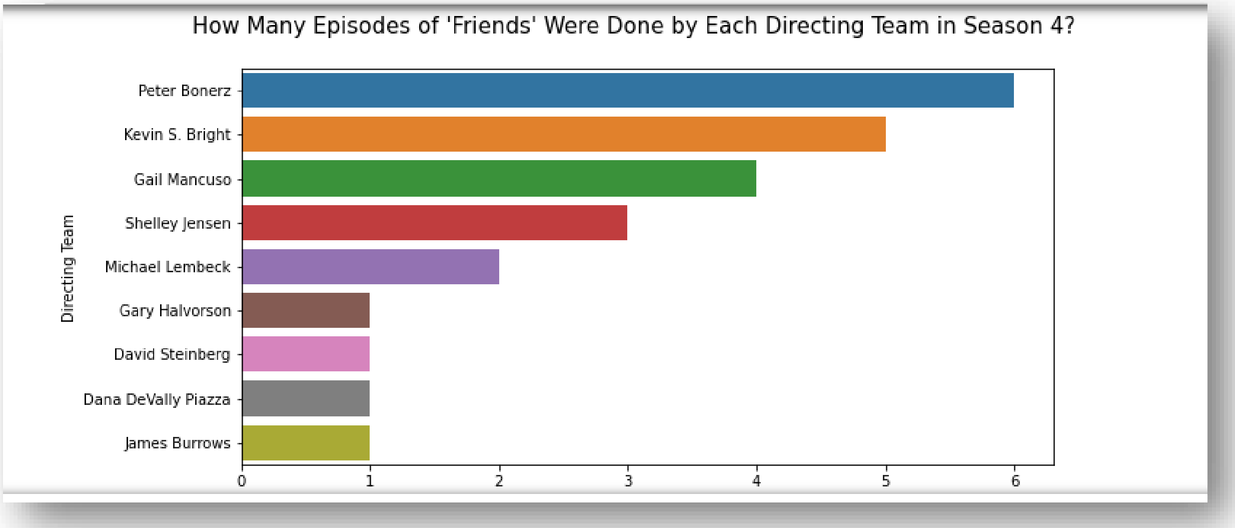


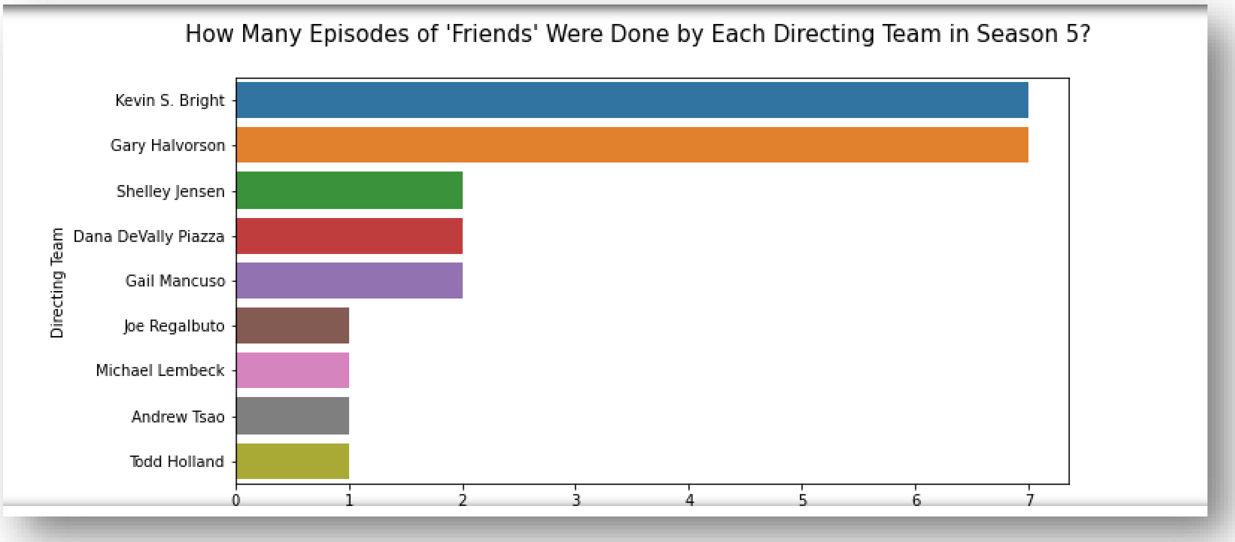
```
In [19]: # Create plots
         for i in df['season'].unique():
             plt.figure(figsize=(10,5))
             temp = df[df['season']==i]
             sns.countplot(data=temp, y='directed_by', order = temp['directed_by'].value_counts().index)
             plt.suptitle('How Many Episodes of \'Friends\' Were Done by Each Directing Team in Season {}?'.format(i), fontsize=15)
             plt.xlabel('Episodes')
             plt.ylabel('Directing Team')
```

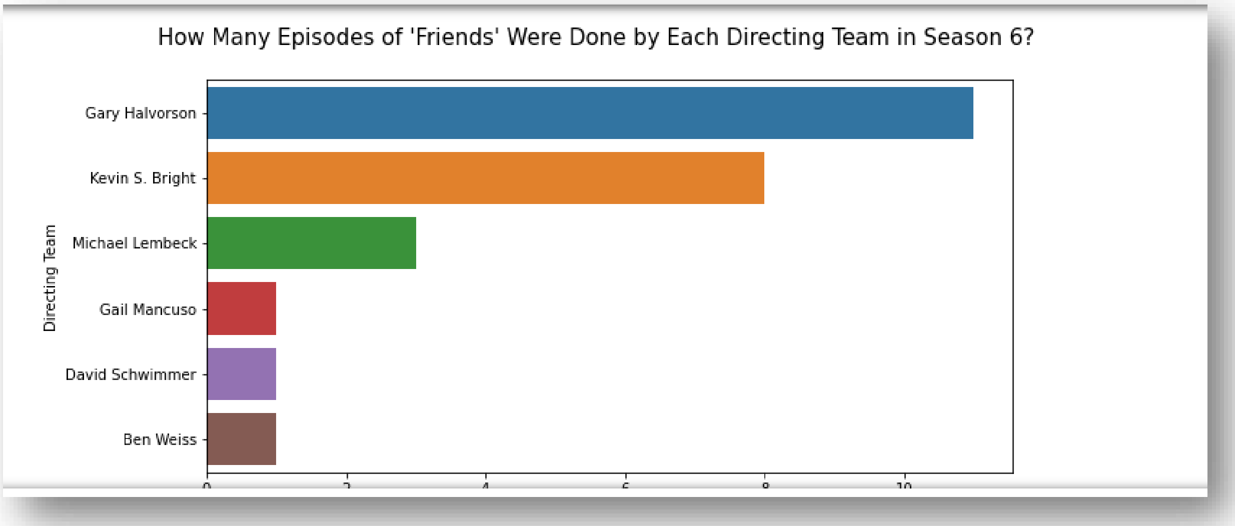

How Many Episodes of 'Friends' Were Done by Each Directing Team in Season 1?
Directing Team
James Burrows
Peter Bonerz
Alan Myerson
Michael Lembeck
Pamela Fryman
Arlene Sanford
Paul Lazarus
Gail Mancuso
Robby Benson
Kevin S. Bright
0
2
4
6
8
10


How Many Episodes of 'Friends' Were Done by Each Directing Team in Season 2?
Directing Team
Michael Lembeck
Kevin S. Bright
Gail Mancuso
Thomas Schlamme
Ellen Gittelsohn
Peter Bonerz
Mary Kay Place
James Burrows
0
2
4
6
8
10
12
14

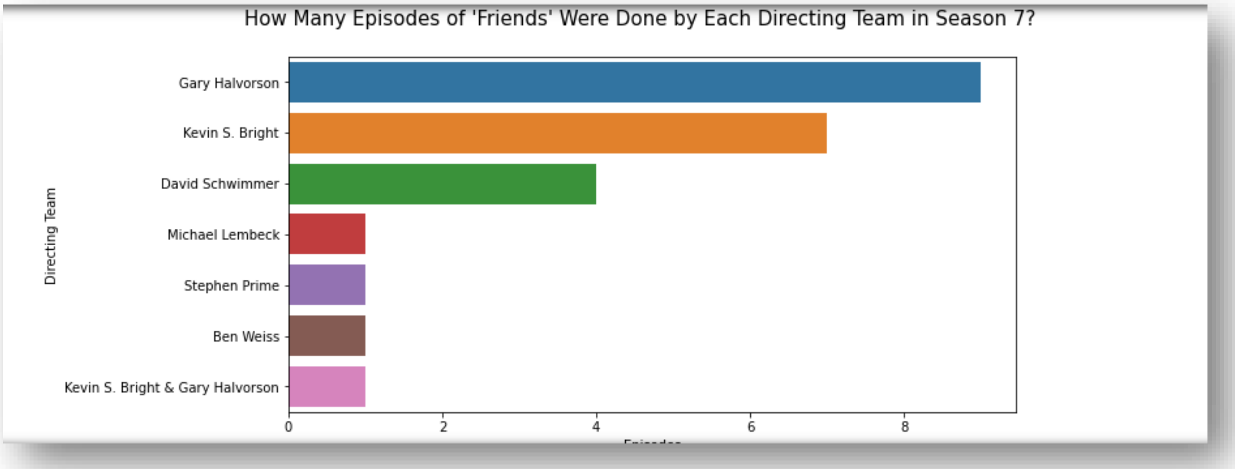

How Many Episodes of 'Friends' Were Done by Each Directing Team in Season 3?
Directing Team
Robby Benson
Gail Mancuso
Terry Hughes
Kevin S. Bright
Steve Zuckerman
Peter Bonerz
James Burrows
Sam Simon
Michael Lembeck
Shelley Jensen
Pamela Fryman


How Many Episodes of 'Friends' Were Done by Each Directing Team in Season 4?
Directing Team
Peter Bonerz
Kevin S. Bright
Gail Mancuso
Shelley Jensen
Michael Lembeck
Gary Halvorson
David Steinberg
Dana DeVally Piazza
James Burrows
0
1
2
3
4
5
6

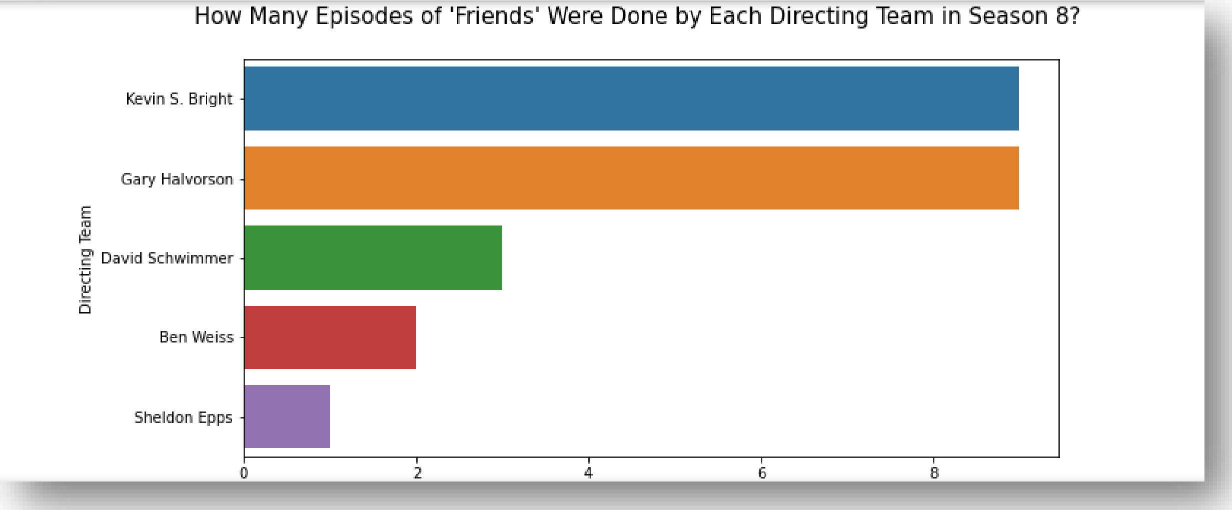

How Many Episodes of 'Friends' Were Done by Each Directing Team in Season 5?
Directing Team
Kevin S. Bright
Gary Halvorson
Shelley Jensen
Dana DeVally Piazza
Gail Mancuso
Joe Regalbuto
Michael Lembeck
Andrew Tsao
Todd Holland
0
1
2
3
4
5
6
7



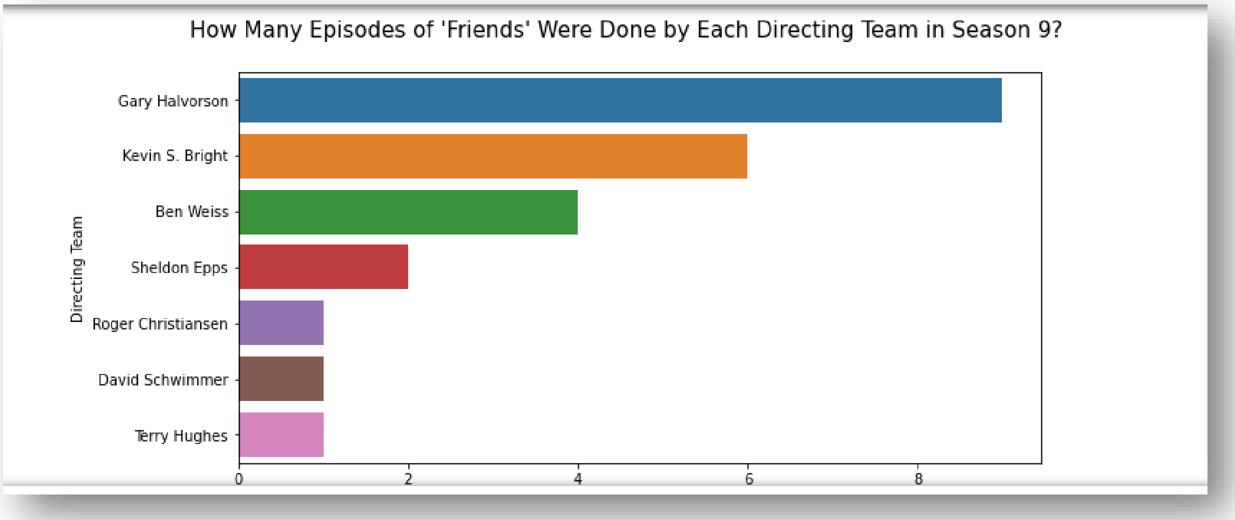
How Many Episodes of 'Friends' Were Done by Each Directing Team in Season 6?

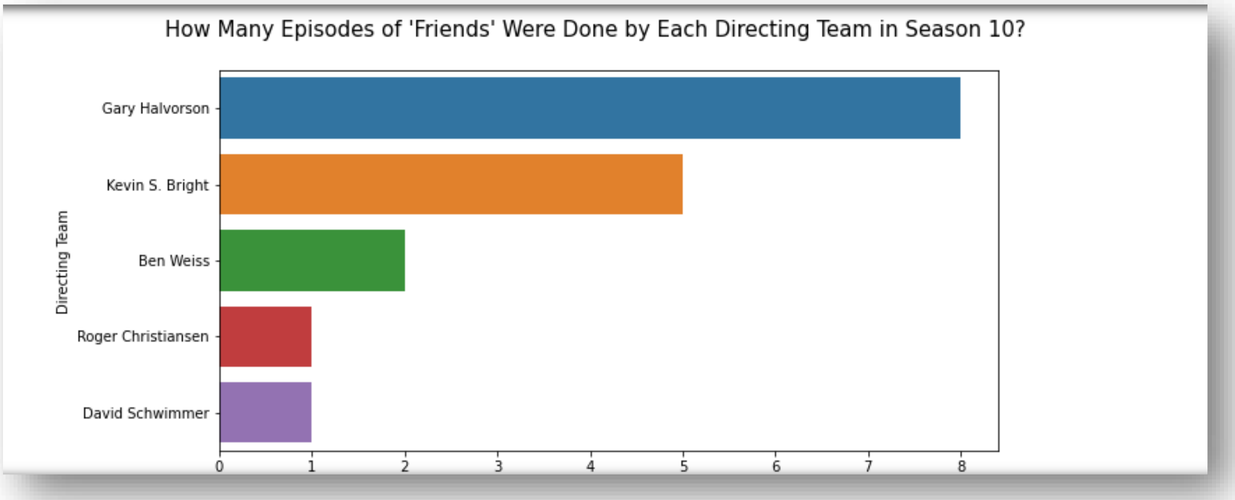


How Many Episodes of 'Friends' Were Done by Each Directing Team in Season 7?



How Many Episodes of 'Friends' Were Done by Each Directing Team in Season 8?


How Many Episodes of 'Friends' Were Done by Each Directing Team in Season 9?
Directing Team
Gary Halvorson
Kevin S. Bright
Ben Weiss
Sheldon Epps
Roger Christiansen
David Schwimmer
Terry Hughes
0
2
4
6
8


How Many Episodes of 'Friends' Were Done by Each Directing Team in Season 10?
Directing Team
Gary Halvorson
Kevin S. Bright
Ben Weiss
Roger Christiansen
David Schwimmer
0
1
2
3
4
5
6
7
8