



Importing all the required dependencies

In []:

```
pip install -U textblob
```

In [71]:

```
import matplotlib
import numpy as np
import pandas as pd
import nltk
import re
from nltk.book import *
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
import unicodedata
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from textblob import TextBlob, Word
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB # Naive Bayes
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
```

Lead Actor Analysis

In [72]:

```
df = pd.read_csv('transcripts_friends/friends-dataset.csv', index_col=0)
df.head()
```

Out[72]:

	Location	Scene	Season	Speaker	Text
0	Chandler and Joey's	Joey is repacking the furniture into boxes to...	4	Joey	What?
1	Monica and Rachel's	the next morning, Monica is getting the door,	1	Monica	Oh, you got the whole night, huh?
2	Mike's apartment	His phone rings and he picks up	9	Chandler	Phoebe is going to say "yes"? That's, that's g...
3	Monica and Rachel's	Ross and Monica are sitting on the couch play...	4	Ross	Yeah.
4	Phoebe's birthday	she's telling everyone what she found out at ...	7	Phoebe	Oh!



Can you guess who is the most talkative person in F.R.I.E.N.D.S?

Well here it is! Let's run this

In [73]:

```
"""Count the number of dialogues by per person in F.R.I.E.N.D.S """
dialogue_count = df['Speaker'].value_counts()
```

In [74]:

```
print(dialogue_count)
```

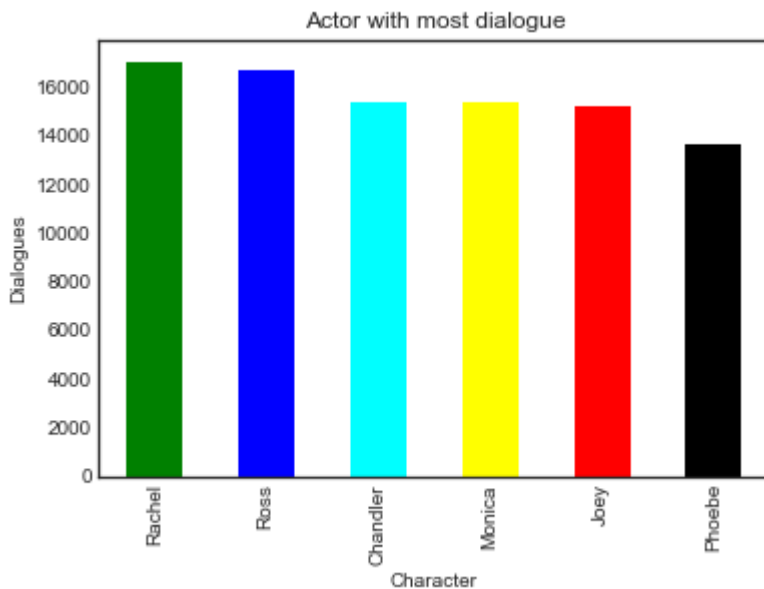
```
Rachel      17058
Ross        16672
Chandler    15389
Monica      15353
Joey        15235
Phoebe      13628
Name: Speaker, dtype: int64
```

In [75]:

```

"""Plot the bar graph with count"""
import matplotlib.pyplot as plt
dialogue_count.plot.bar(color=['green', 'blue', 'cyan', 'yellow', 'red', 'black'
])
plt.title("Actor with most dialogue")
plt.ylabel('Dialogues')
plt.xlabel('Character')
plt.show()

```



Top Location in the F.R.I.E.N.D.S

In [76]:

```

"""Find top most 6 locations in F.R.I.E.N.D.S """
n=6
count_location = df['Location'].value_counts()[:n]
print(count_location)

```

```

Central Perk          16805
Monica and Rachel's   15722
Monica and Chandler's  5314
Chandler and Joey's   3837
Joey and Rachel's     3024
Monica                2558
Name: Location, dtype: int64

```

OK! Lets see actual number of locations

In [77]:

```

"""All locations in shooting"""
count_location1 = df['Location'].value_counts()
print(len(count_location1))

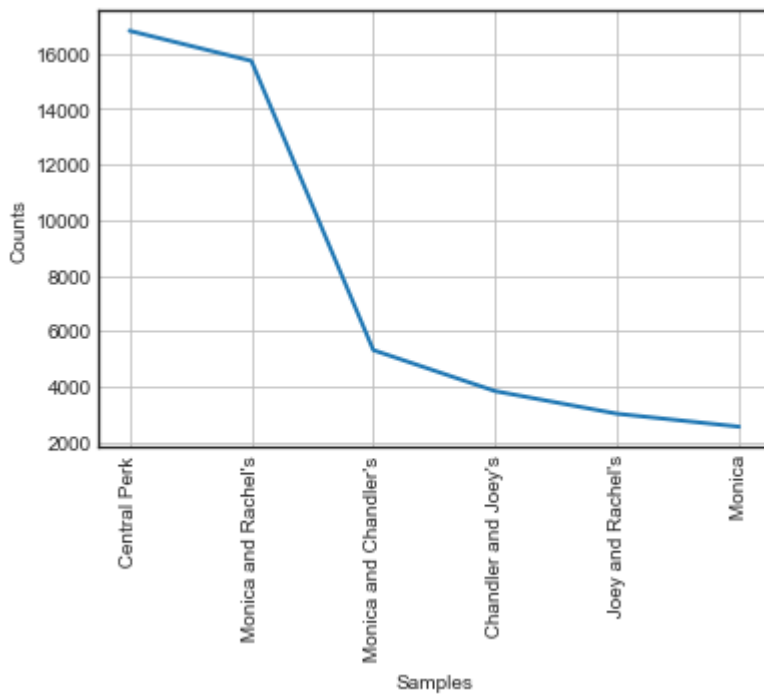
```

743

In [79]:

```
locations = []  
for row in df['Location']:  
    locations.append(row)  
freqdist = FreqDist(locations)  
print(freqdist)  
  
freqdist.plot(n)
```

<FreqDist with 744 samples and 93335 outcomes>



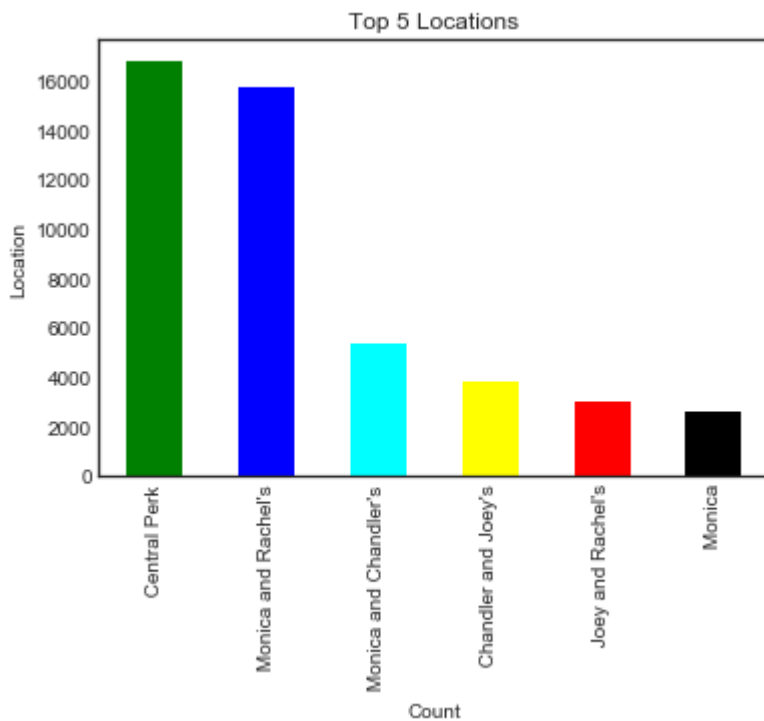
Out[79]:

<matplotlib.axes._subplots.AxesSubplot at 0x1a29241ef0>



In [57]:

```
import matplotlib.pyplot as plt
count_location.plot.bar(color=['green', 'blue', 'cyan', 'yellow', 'red', 'black']
)
plt.title("Top 5 Locations")
plt.ylabel('Location')
plt.xlabel('Count')
plt.show()
```



Ok! We know now who is talkative and where the series are shot! Let's move on to relation between two actors!

Link Analysis

In [43]:

```
pip install --user git+https://github.com/mwaskom/seaborn.git #install seaborn to create environment in case the
```

```
Collecting git+https://github.com/mwaskom/seaborn.git
  Cloning https://github.com/mwaskom/seaborn.git to /private/var/folders/bx/3nfnzncgd6mx9l40jzl85pwzw0000gn/T/pip-req-build-swjn0_yk
  Running command git clone -q https://github.com/mwaskom/seaborn.git /private/var/folders/bx/3nfnzncgd6mx9l40jzl85pwzw0000gn/T/pip-req-build-swjn0_yk
Requirement already satisfied (use --upgrade to upgrade): seaborn==0.9.1.dev0 from git+https://github.com/mwaskom/seaborn.git in /Users/rani/.local/lib/python3.7/site-packages
Requirement already satisfied: numpy>=1.9.3 in /Users/rani/anaconda3/lib/python3.7/site-packages (from seaborn==0.9.1.dev0) (1.16.4)
Requirement already satisfied: scipy>=0.14.0 in /Users/rani/anaconda3/lib/python3.7/site-packages (from seaborn==0.9.1.dev0) (1.3.0)
Requirement already satisfied: pandas>=0.15.2 in /Users/rani/anaconda3/lib/python3.7/site-packages (from seaborn==0.9.1.dev0) (0.24.2)
Requirement already satisfied: matplotlib>=1.4.3 in /Users/rani/anaconda3/lib/python3.7/site-packages (from seaborn==0.9.1.dev0) (3.1.0)
Requirement already satisfied: pytz>=2011k in /Users/rani/anaconda3/lib/python3.7/site-packages (from pandas>=0.15.2->seaborn==0.9.1.dev0) (2019.1)
Requirement already satisfied: python-dateutil>=2.5.0 in /Users/rani/anaconda3/lib/python3.7/site-packages (from pandas>=0.15.2->seaborn==0.9.1.dev0) (2.8.0)
Requirement already satisfied: cycler>=0.10 in /Users/rani/anaconda3/lib/python3.7/site-packages (from matplotlib>=1.4.3->seaborn==0.9.1.dev0) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/rani/anaconda3/lib/python3.7/site-packages (from matplotlib>=1.4.3->seaborn==0.9.1.dev0) (1.1.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /Users/rani/anaconda3/lib/python3.7/site-packages (from matplotlib>=1.4.3->seaborn==0.9.1.dev0) (2.4.0)
Requirement already satisfied: six>=1.5 in /Users/rani/anaconda3/lib/python3.7/site-packages (from python-dateutil>=2.5.0->pandas>=0.15.2->seaborn==0.9.1.dev0) (1.12.0)
Requirement already satisfied: setuptools in /Users/rani/anaconda3/lib/python3.7/site-packages (from kiwisolver>=1.0.1->matplotlib>=1.4.3->seaborn==0.9.1.dev0) (41.0.1)
Building wheels for collected packages: seaborn
  Building wheel for seaborn (setup.py) ... done
  Stored in directory: /private/var/folders/bx/3nfnzncgd6mx9l40jzl85pwzw0000gn/T/pip-ephem-wheel-cache-db_5sdvf/wheels/4e/57/af/138c7bf2390a67f7010588cf080fb8b4e741bb290c2ad12018
Successfully built seaborn
Note: you may need to restart the kernel to use updated packages.
```

In [58]:

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
plt.style.use('seaborn-white')
```

In [59]:

```
num_seasons = 10
characters = ['Chandler', 'Joey', 'Monica', 'Phoebe', 'Rachel', 'Ross']
num_characters = len(characters)
mentions_file = 'transcripts_friends/friendsMentions.csv'
```

In [60]:

```
friends_mentions = pd.read_csv(mentions_file)
#friends_mentions.head()
```

Lets see which character mentions all the other characters in 10 seasons

In [61]:

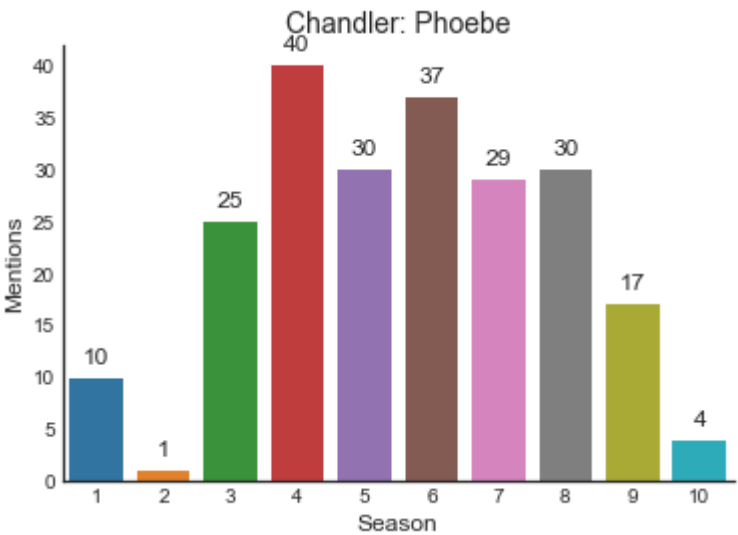
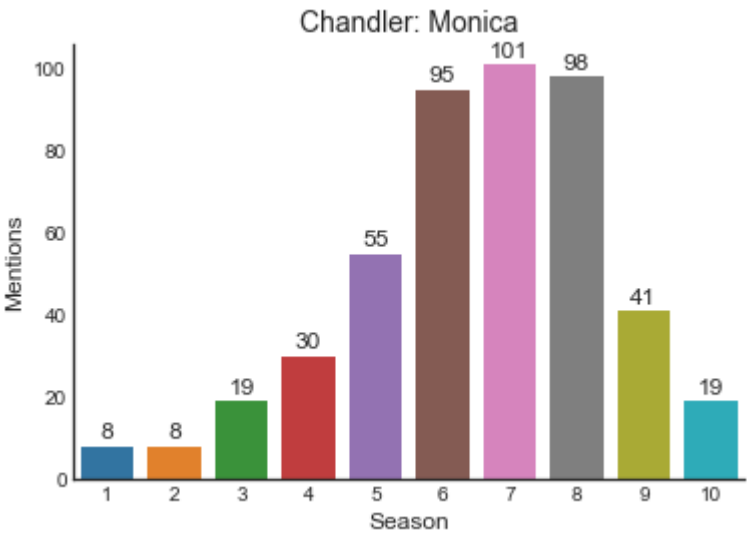
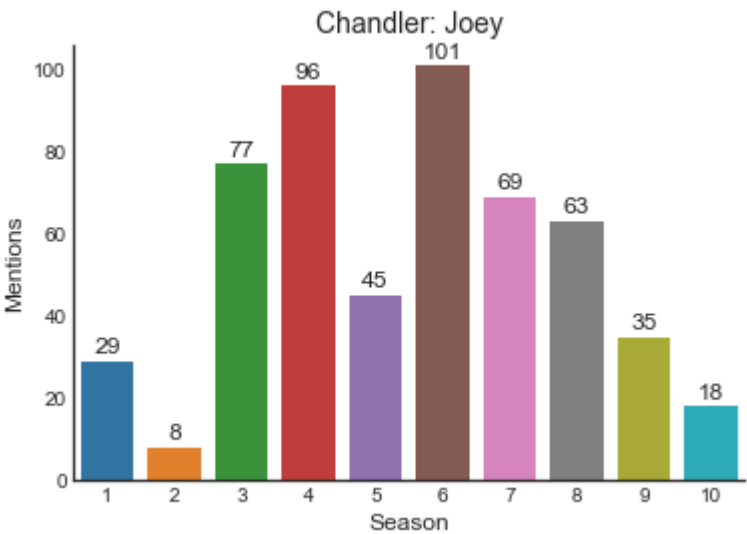
```
def get_plot(speakerName, mentionedName):
    """Return the frequency number with bar graph of character mentioning the name of other character """

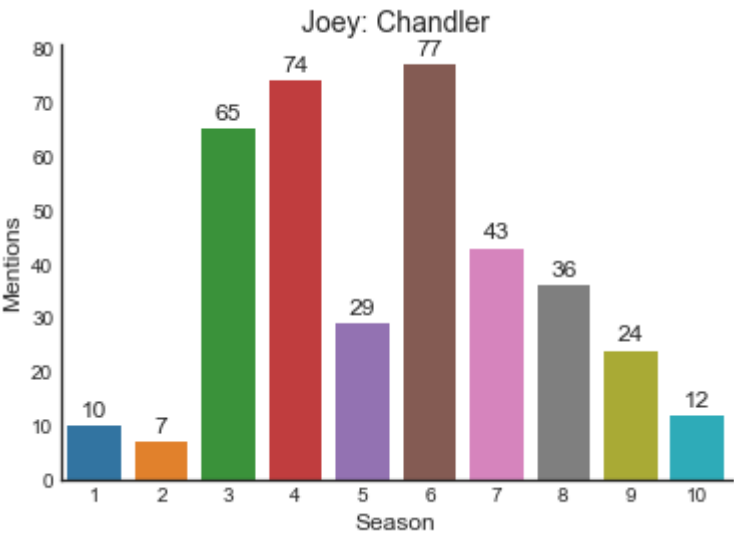
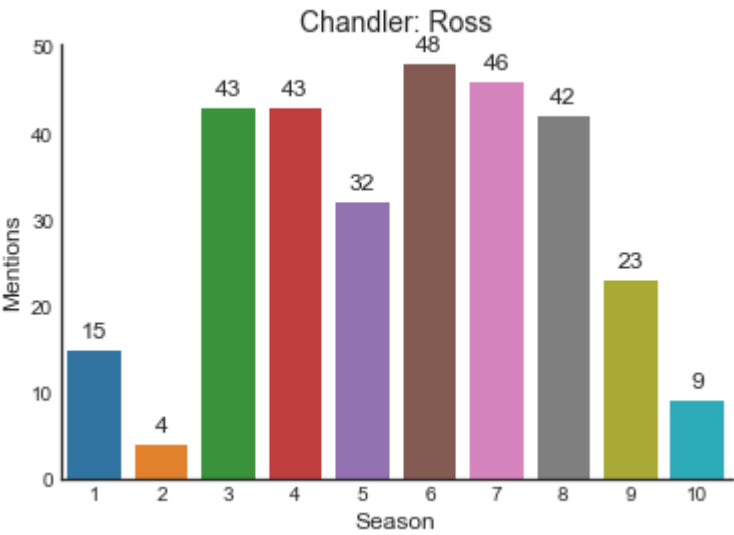
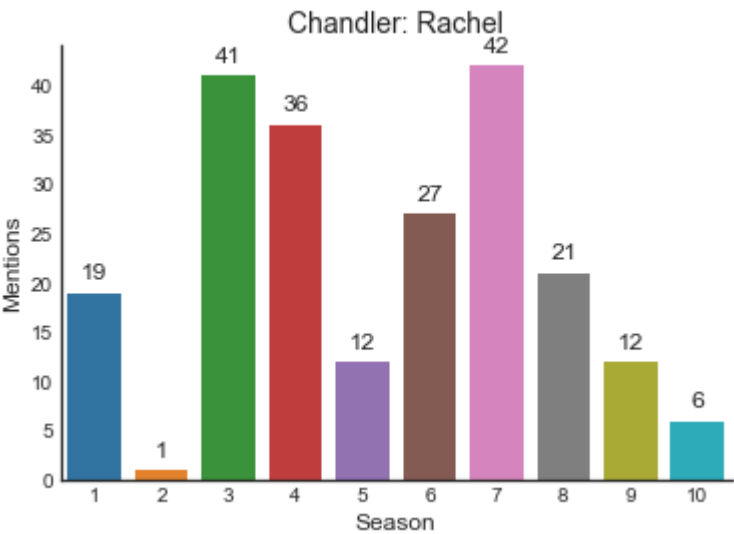
    data = friends_mentions.loc[(friends_mentions['speakerName']==speakerName)&(
friends_mentions['mentionedName']==mentionedName)]
    sns.barplot(x = 'seasonNumber', y = 'mentions', data = data)#, color = '#6000df')
    sns.despine()
    ax = plt.gca()
    ax.set_xlabel('Season', fontsize = 12)
    ax.set_ylabel('Mentions', fontsize = 12)
    ax.set_title('{}: {}'.format(speakerName, mentionedName), fontsize = 14)
    for p in ax.patches:
        ax.text(p.get_x() + p.get_width()/2., p.get_height()+1, '%d' % int(p.get_height()),
        fontsize=12, ha='center', va='bottom')
    plt.show()
```

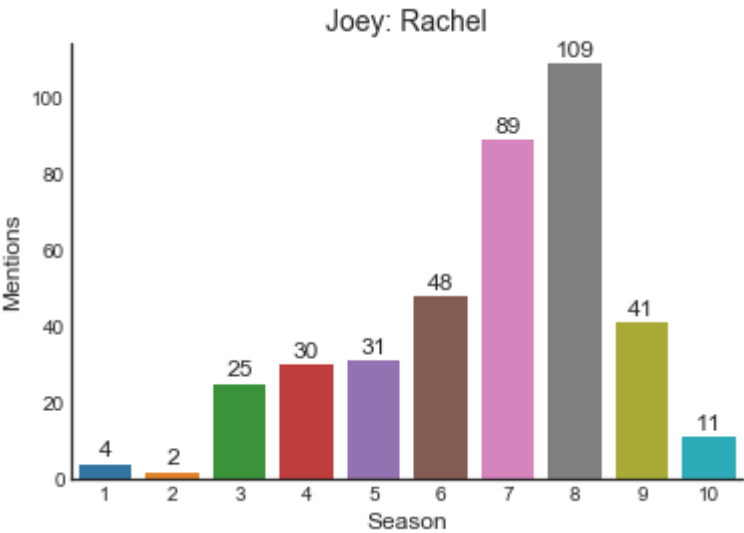
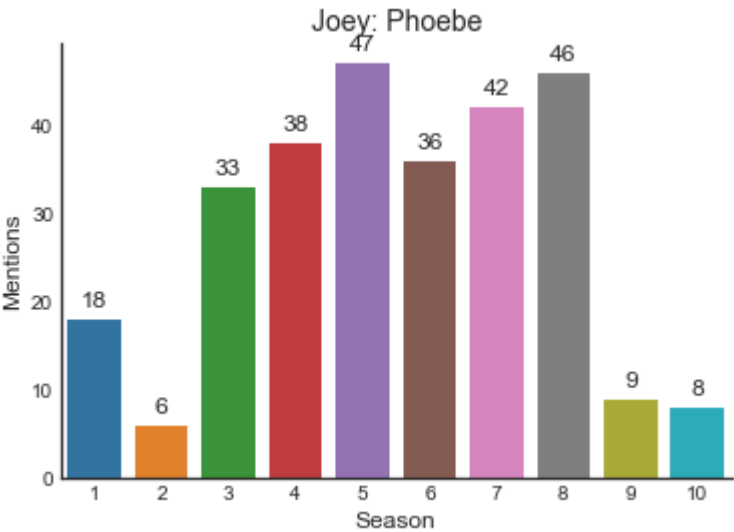
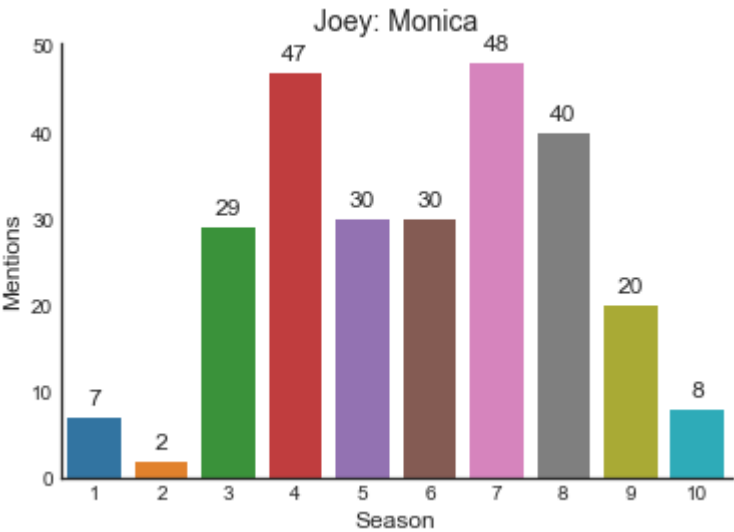


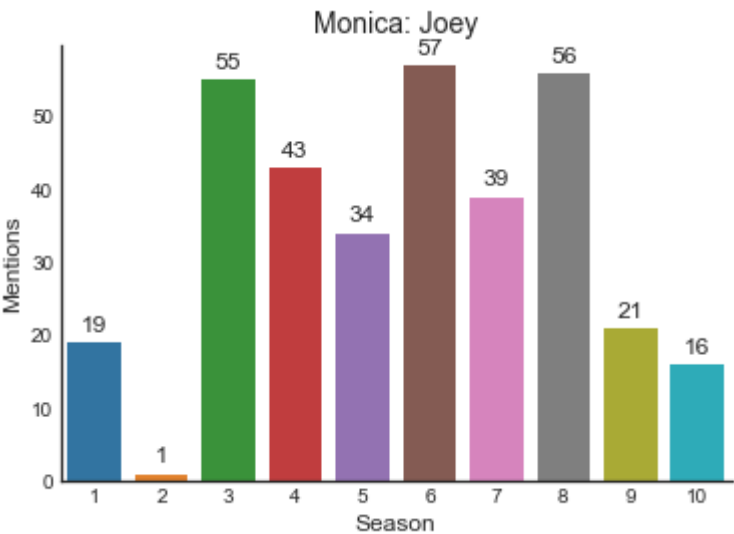
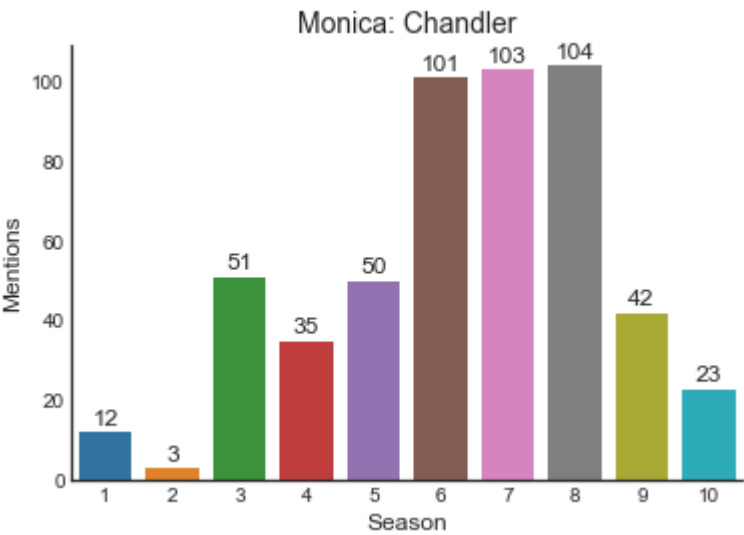
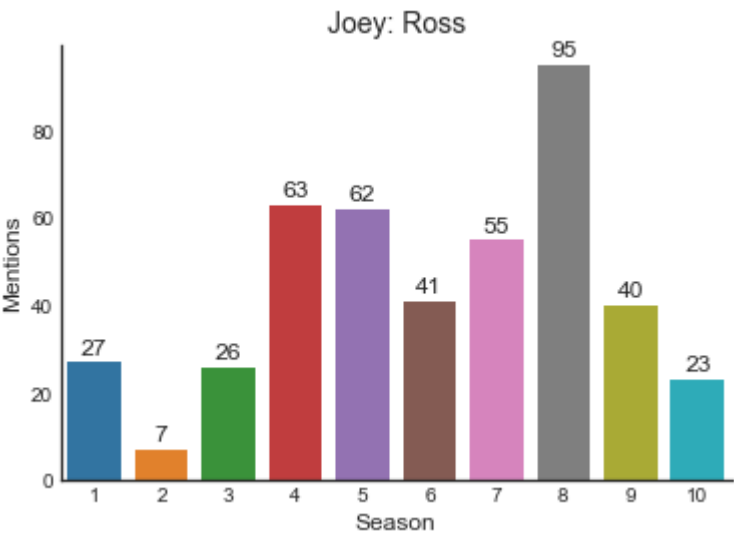
In [62]:

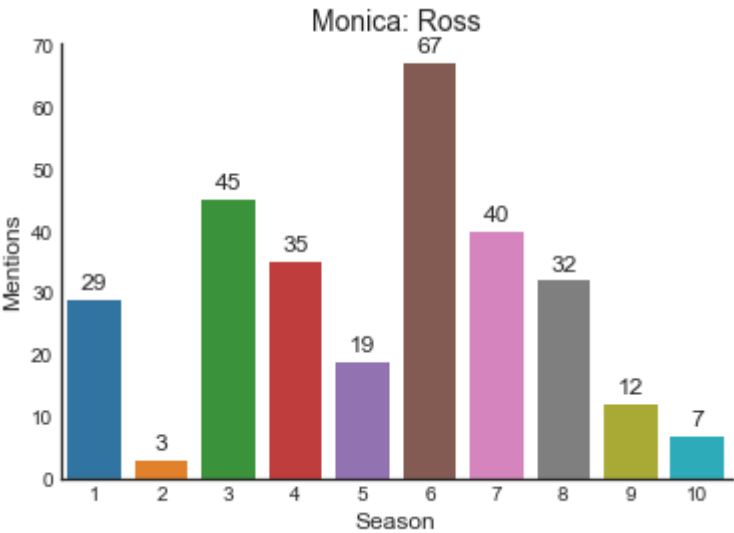
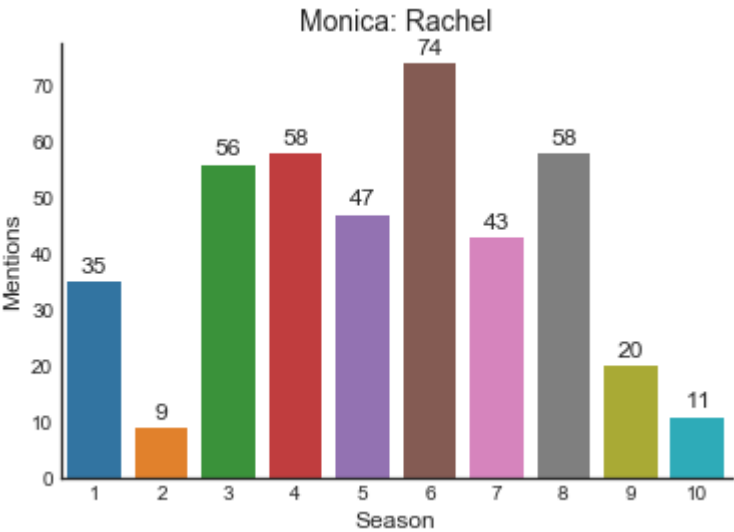
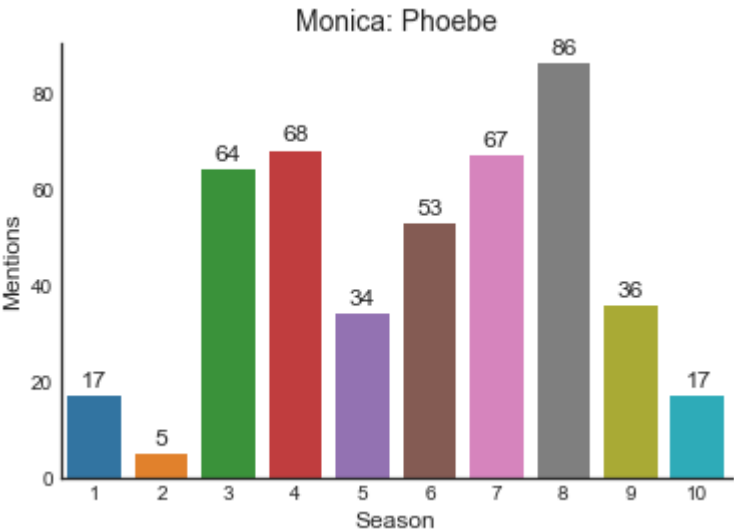
```
"""Lets call the fuction and see how many times, lets say Chandler has called Jo  
ey"""  
  
#loop through the character list  
for speakerName in characters:  
    #loop through the mentioned name  
    for mentionedName in characters:  
        #Skip the self call  
        if speakerName == mentionedName:  
            continue  
        else:  
            #get a plot of speaker and reffered name by the speaker  
            get_plot(speakerName, mentionedName)
```

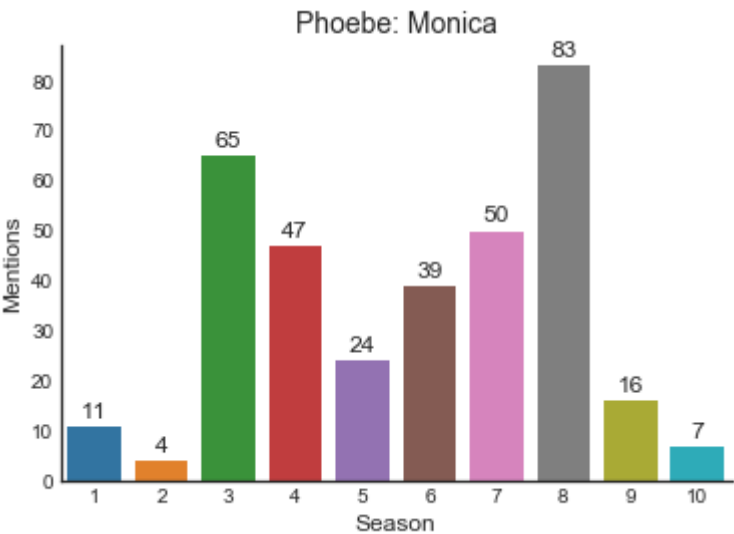
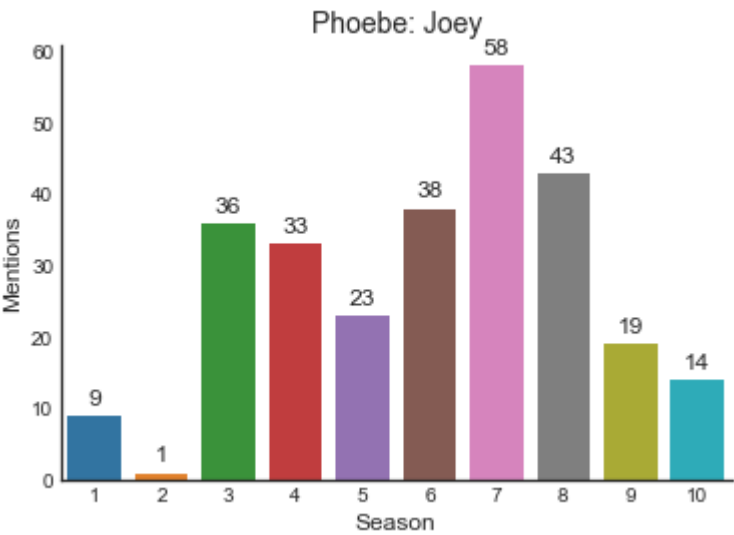
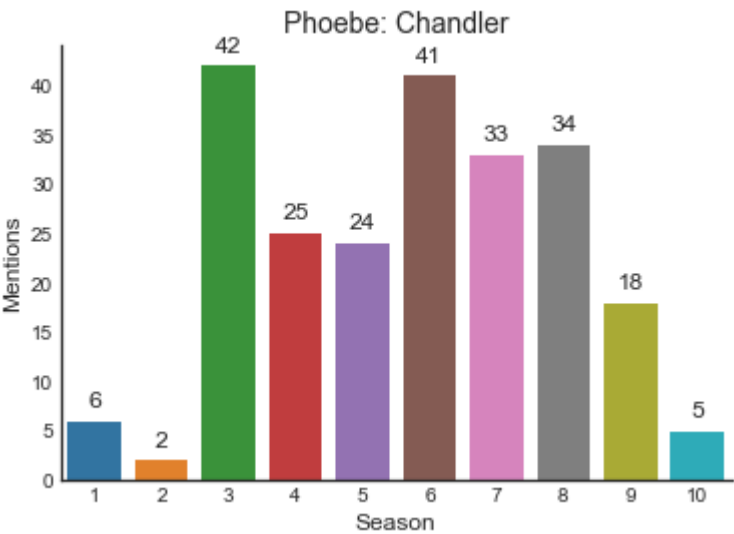



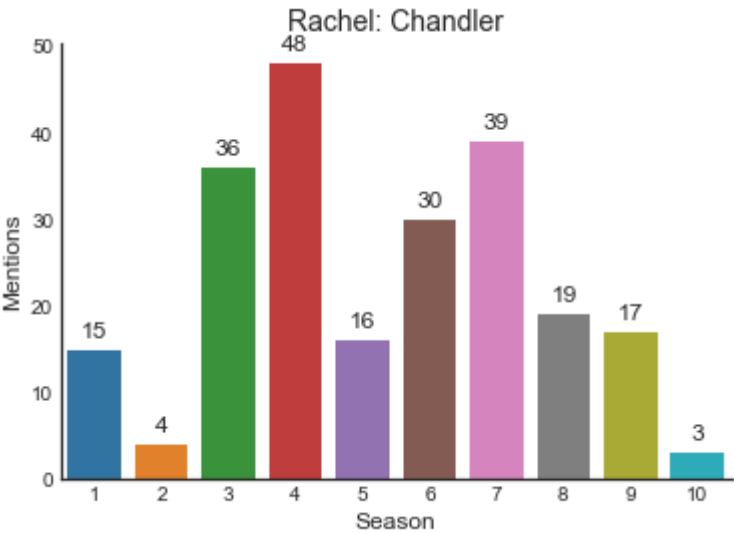
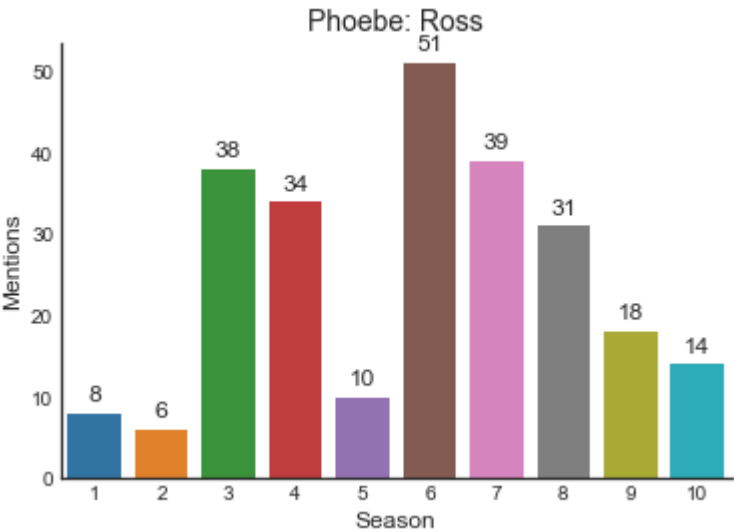
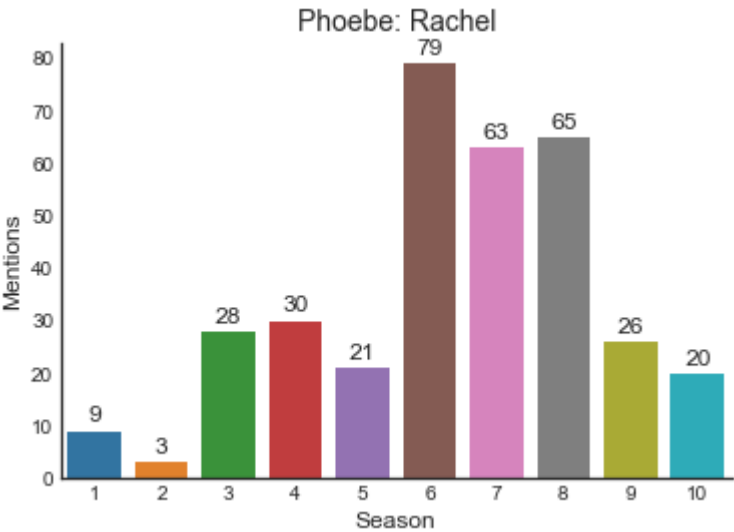


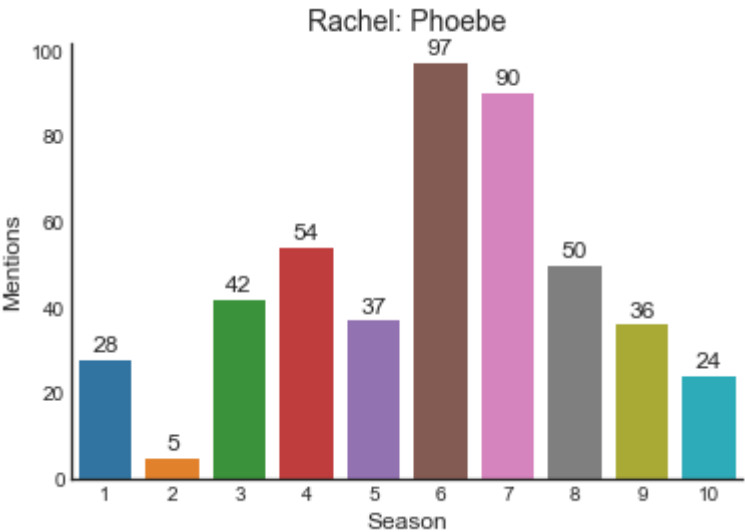
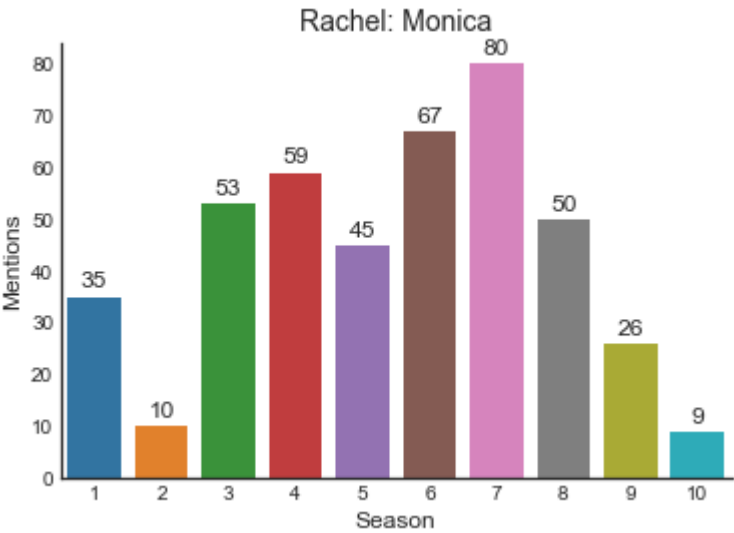
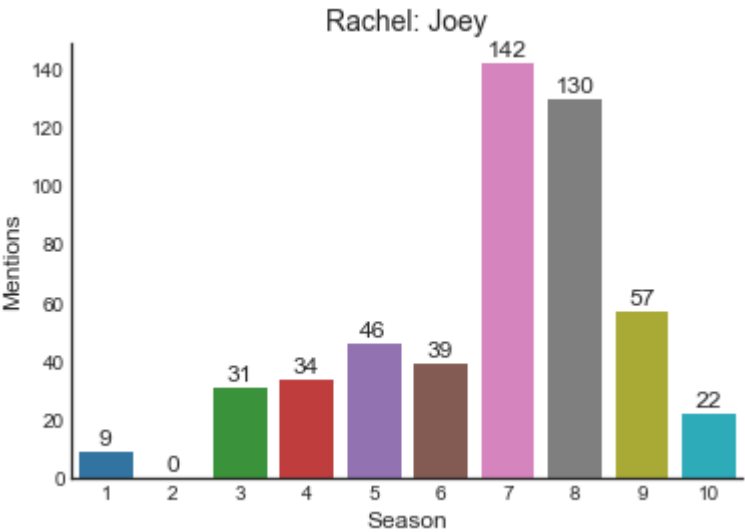


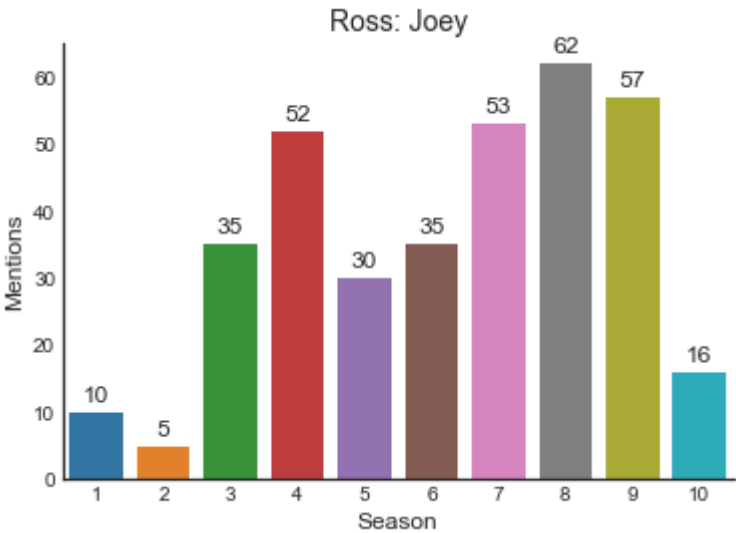
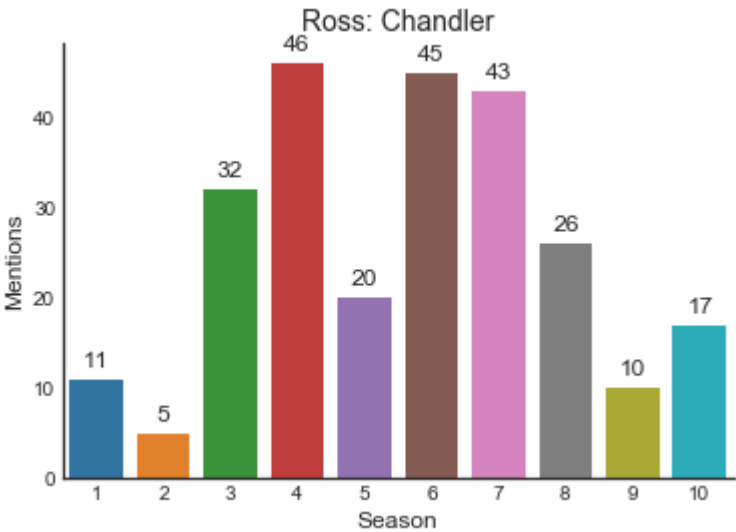
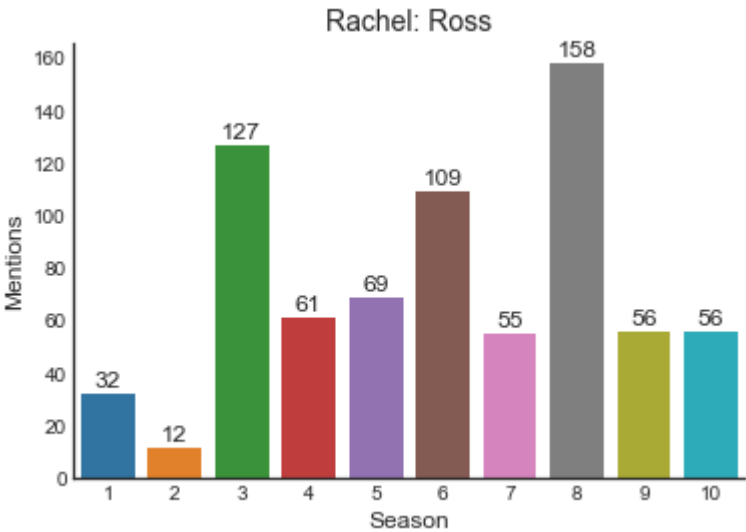


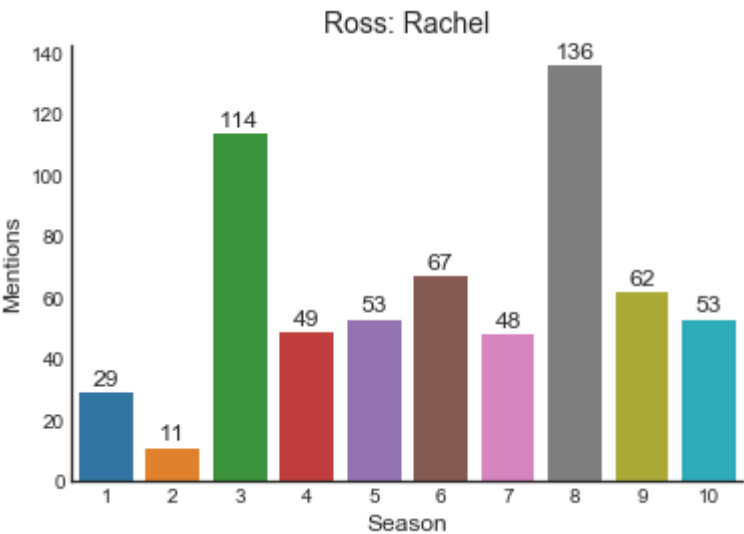
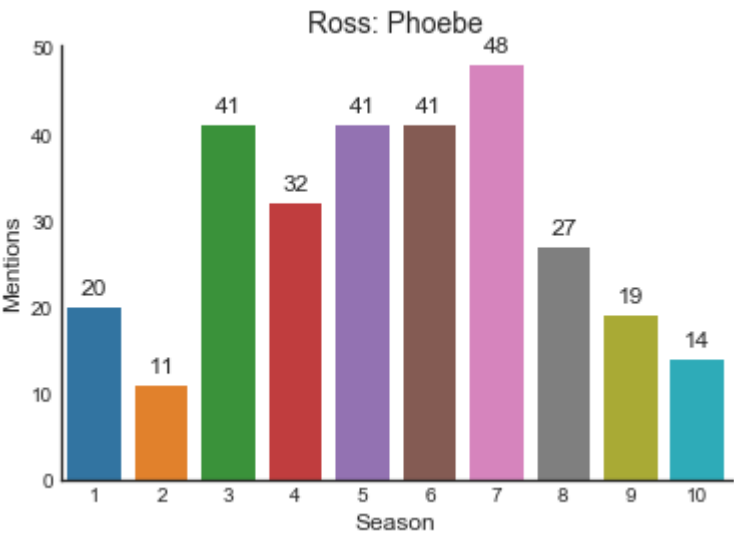
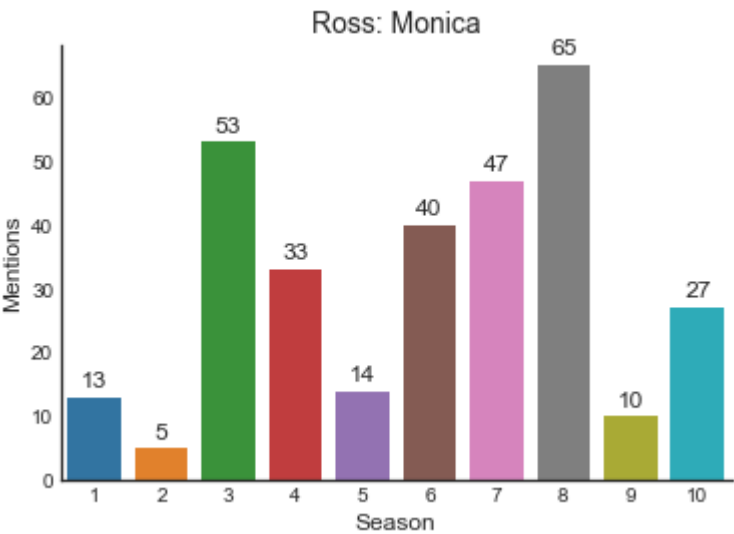












Shall we count it?

In [63]:

```
data = friends_mentions.loc[(friends_mentions['speakerName']==speakerName)&(friends_mentions['mentionedName']==mentionedName)]
```

In [64]:

```
list(data.mentions)
```

Out[64]:

```
[]
```

In [69]:

```
#plt.style.available
```

In [66]:

```
"""Lets count numeric values of the mentioned name"""
friends_mentions.groupby('mentionedName').mentions.sum()
```

Out[66]:

```
mentionedName
Chandler      1613
Joey          2021
Monica        1822
Phoebe        1710
Rachel        1984
Ross          2017
Name: mentions, dtype: int64
```

In [67]:

```
"""Lets count numeric values of the speaker name"""
friends_mentions.groupby('speakerName').mentions.sum()
```

Out[67]:

```
speakerName
Chandler      1760
Joey          1750
Monica        2012
Phoebe        1443
Rachel        2369
Ross          1833
Name: mentions, dtype: int64
```

Lets move on to next part

