Matthew Langendorfer

<https://github.com/dorf3/Midterm>

import json

import matplotlib.pyplot as plt

numlist = []

dict = {}

f = open("/Users/ml/Desktop/midterm-project/numbers.txt", "r")

text = f.read()

for num in text:

    if num.isdigit():

        numlist.append(int(num))

for num in numlist:

    if num in dict.keys():

        dict[num] += 1

    else:

        dict[num] = 1

for key, value in dict.items():

    print(f"{key} : {value}")

plt.bar(list(dict.keys()), dict.values(), color='g')

plt.xlabel("Value")

plt.ylabel("Frequency")

plt.title('Frequency of Text File')

plt.show()

y = json.dumps(dict)

print(y)

with open("out.json", "w") as outfile:

    json.dump(y, outfile)

import pandas as pd  
  
df = pd.read\_csv('/Users/ml/Desktop/ViewingActivity.csv')  
df.shape

(200, 10)

df.head(1)

Profile Name Start Time Duration Attributes \  
0 Charlie 2013-03-20 5:17:53 0:00:05 NaN   
  
 Title Supplemental Video Type \  
0 Star Trek: Deep Space Nine: Season 5: Empok No... NaN   
  
 Device Type Bookmark Latest Bookmark Country   
0 Mac 0:00:05 Not latest view US (United States)

df = df.drop(['Profile Name', 'Attributes', 'Supplemental Video Type', 'Bookmark', 'Latest Bookmark', 'Country'], axis=1)  
df.head(1)

Start Time Duration \  
0 2013-03-20 5:17:53 0:00:05   
  
 Title Device Type   
0 Star Trek: Deep Space Nine: Season 5: Empok No... Mac

df.dtypes

Start Time object  
Duration object  
Title object  
Device Type object  
dtype: object

df['Start Time'] = pd.to\_datetime(df['Start Time'], utc=True)  
df.dtypes

Start Time datetime64[ns, UTC]  
Duration object  
Title object  
Device Type object  
dtype: object

df = df.set\_index('Start Time')  
  
df.index = df.index.tz\_convert('US/Eastern')  
  
df = df.reset\_index()  
  
df.head(1)

Start Time Duration \  
0 2013-03-20 01:17:53-04:00 0:00:05   
  
 Title Device Type   
0 Star Trek: Deep Space Nine: Season 5: Empok No... Mac

df['Duration'] = pd.to\_timedelta(df['Duration'])  
df.dtypes

Start Time datetime64[ns, US/Eastern]  
Duration timedelta64[ns]  
Title object  
Device Type object  
dtype: object

startrek = df[df['Title'].str.contains('Star Trek: Deep Space Nine', regex=False)]

startrek.shape

(43, 4)

startrek = startrek[(startrek['Duration'] > '0 days 00:01:00')]  
startrek.shape

(38, 4)

startrek['Duration'].sum()

Timedelta('0 days 20:00:57')

startrek['Duration'].mean()

Timedelta('0 days 00:31:36.236842105')

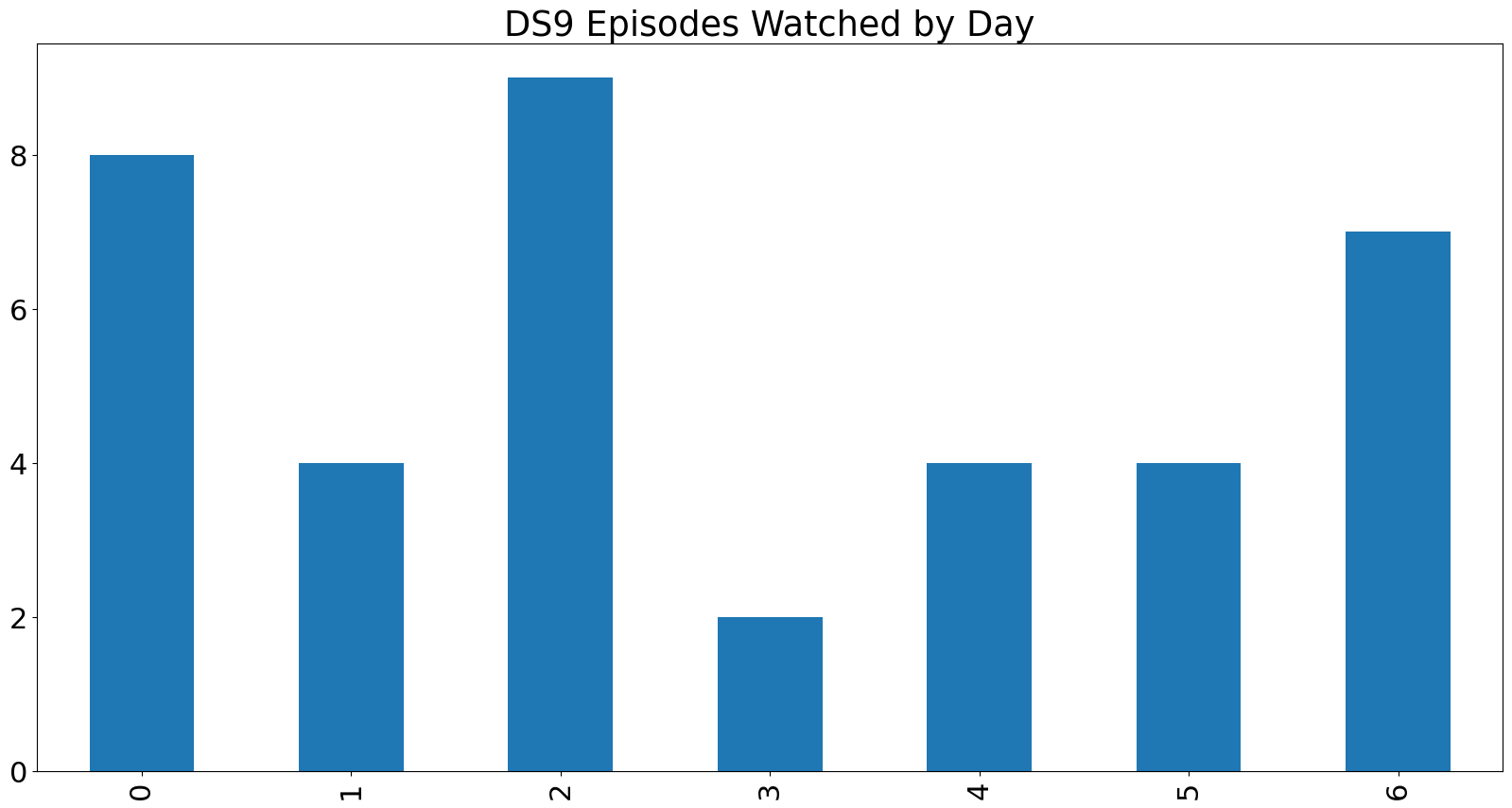
startrek['weekday'] = startrek['Start Time'].dt.weekday  
startrek['hour'] = startrek['Start Time'].dt.hour  
  
startrek.head(1)

Start Time Duration \  
1 2013-03-20 00:27:45-04:00 0 days 00:44:31   
  
 Title Device Type weekday \  
1 Star Trek: Deep Space Nine: Season 5: Blaze of... Mac 2   
  
 hour   
1 0

%matplotlib inline  
import matplotlib

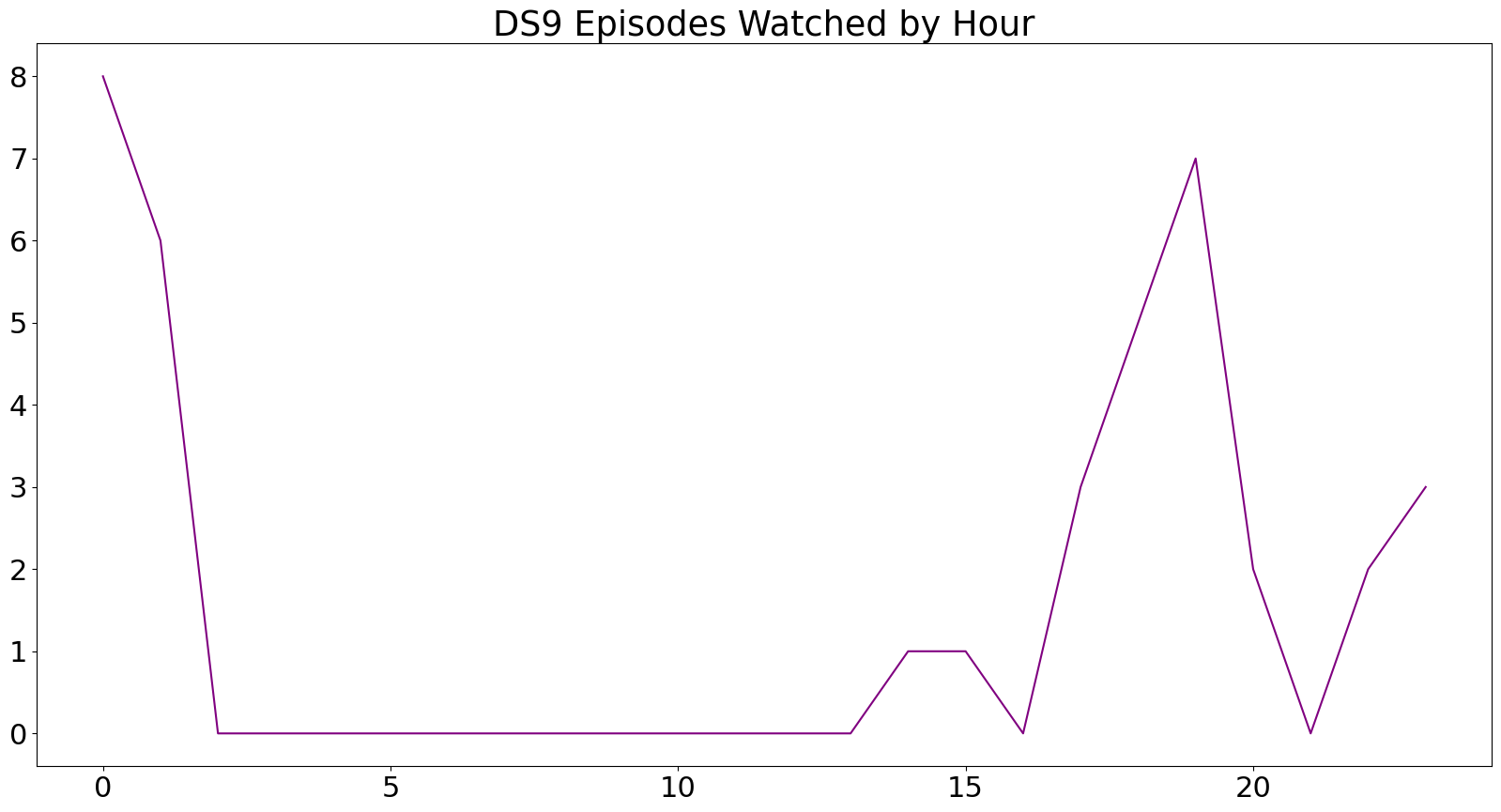
startrek['weekday'] = pd.Categorical(startrek['weekday'], categories=  
 [0,1,2,3,4,5,6],  
 ordered=True)  
  
startrek\_by\_day = startrek['weekday'].value\_counts()  
  
startrek\_by\_day = startrek\_by\_day.sort\_index()  
  
matplotlib.rcParams.update({'font.size': 22})  
  
startrek\_by\_day.plot(kind='bar', figsize=(20,10), title='DS9 Episodes Watched by Day')

<AxesSubplot:title={'center':'DS9 Episodes Watched by Day'}>



startrek['hour'] = pd.Categorical(startrek['hour'], categories=  
 [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23],  
 ordered=True)  
  
startrek\_by\_hour = startrek['hour'].value\_counts()  
  
startrek\_by\_hour = startrek\_by\_hour.sort\_index()  
  
startrek\_by\_hour.plot(kind='line', color = 'purple', figsize=(20,10), title='DS9 Episodes Watched by Hour')

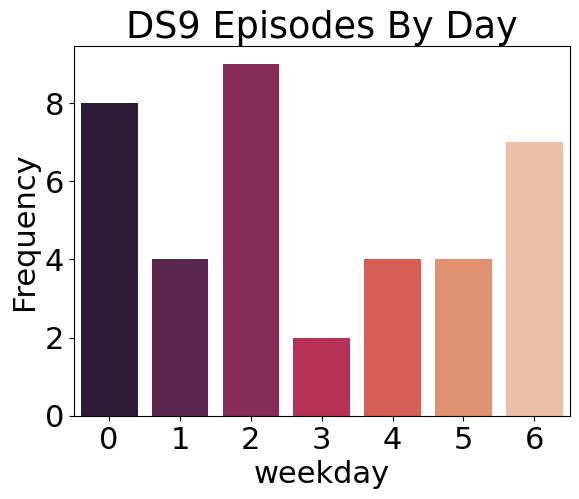
<AxesSubplot:title={'center':'DS9 Episodes Watched by Hour'}>



import seaborn as sns  
palette = sns.color\_palette("rocket", as\_cmap=True)  
ax = sns.countplot(startrek['weekday'], label='Count', palette='rocket')  
ax.set\_title('DS9 Episodes By Day')  
ax.set\_ylabel('Frequency')

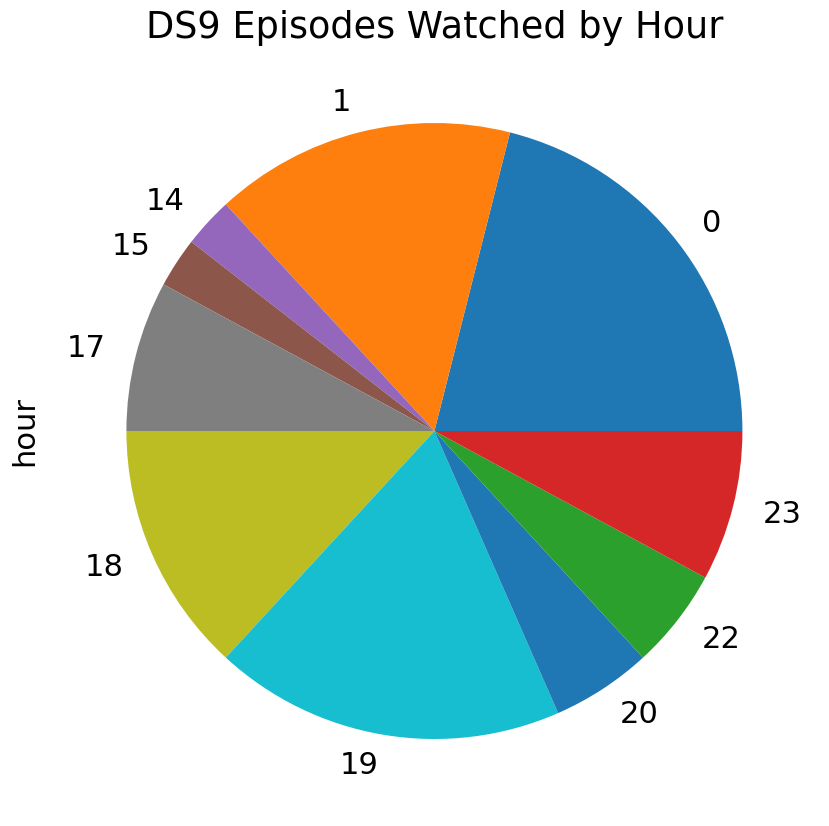
/Users/ml/opt/anaconda3/lib/python3.9/site-packages/seaborn/\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.  
 warnings.warn(

Text(0, 0.5, 'Frequency')



startrek['hour'] = pd.Categorical(startrek['hour'], categories=  
 [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23],  
 ordered=True)  
  
startrek\_by\_hour = startrek['hour'].value\_counts()  
  
startrek\_by\_hour = startrek\_by\_hour.sort\_index()  
  
startrek\_by\_hour.plot(kind='pie', figsize=(20,10), title='DS9 Episodes Watched by Hour')

<AxesSubplot:title={'center':'DS9 Episodes Watched by Hour'}, ylabel='hour'>



startrek\_by\_day = startrek['Device Type'].value\_counts()  
  
startrek\_by\_day = startrek\_by\_day.sort\_index()  
print(startrek\_by\_day)

Mac 36  
Microsoft Xbox 360 2  
Name: Device Type, dtype: int64

sns.color\_palette("cubehelix", as\_cmap=True)  
ax = sns.countplot(startrek['Device Type'], label='Count', palette='cubehelix')  
ax.set\_title('Device Type')  
ax.set\_ylabel('Viewings')

/Users/ml/opt/anaconda3/lib/python3.9/site-packages/seaborn/\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.  
 warnings.warn(

Text(0, 0.5, 'Viewings')

