In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


from datetime import datetime  # to access datetime
import scipy.stats as stats

import plotly.express as px # for interactive plotting
import plotly.graph_objects as go # for interactive plotting

# set the graphics style initially to defaul
plt.style.use('default')
```

```
KeyboardInterrupt
```

The imported dataset 'EthnicDist" shows the distribution of population by gender and ethnic groups for counties in the United States

In [ ]:
```
df2=pd.read_csv('EthnicDist.csv')
df2.head()
```

Q1. Create two new variables in the dataframe that meaures males and females as a percentage of total population

In [ ]:
```
df2['PCTMALE']=df2['TOT_MALE']/df2['TOT_POP']
df2['PCTFEMALE']=df2['TOT_FEMALE']/df2['TOT_POP']
df2.head()
```

Q2. Create a new dataframe, df2MaFe, that calculates the average percentage distribution of males and females by States. Reset the index.

In [ ]:
```
df2MaFe=df2.groupby(['STNAME'])[['PCTMALE','PCTFEMALE']].mean().reset_index(
df2MaFe.head()
```

Q3. Using Plotly graph-object, create a scatterplot of distribution of Males and Females in each State as shown below. Include the Title, axes-labels and legends. Make the graph background white.

In [ ]:
```
fig = go.Figure()

# add scatter dots for percentage of voting age population
fig.add_trace(go.Scatter(
  x=df2MaFe['PCTMALE'],
  y=df2MaFe['STNAME'], name='Percent of Male population',
  ))

fig.add_trace(go.Scatter(
     x=df2MaFe['PCTFEMALE'],
    y=df2MaFe['STNAME'],
     name='Percent of Female Population',
    ))

fig.update_traces(mode='markers',
                  marker=dict(line_width=1,symbol='circle',size=16))
```

```
fig.update_layout(
        plot_bgcolor='white',
)

fig.update_layout(title="Percentage Distribution of Males vs Females in U.S.
                  xaxis_title="Percentage Distribution",
                  yaxis_title="States")

fig.show()
```

The datasets apples3 and Google 3 show the Date and adjusted closing prices for Apple and Google stocks.

In [ ]:
```
Apple=pd.read_excel('apple3.xlsx',parse_dates=['Date'])
Apple.head()
```

In [ ]:
```
Google=pd.read_excel('Google3.xlsx',parse_dates=['Date'])
Google.head()
```

Q4. Merge the apple and google datasets

In [ ]:
```
Stocks=Apple.merge(Google,on='Date',suffixes=("_aapl","_goog"))
Stocks.head()
```

Q5.Generate the summary statistics for both Apple and Google closing prices

In [ ]:
```
Stocks[['Adj Close_goog','Adj Close_aapl']].describe()
```

Q6. Slice out Apple stocks with adjusted closing prices less than $150

In [ ]:
```
Stocks.loc[(Stocks['Adj Close_aapl'] < 150)].head()
```

Q7.Using Plotly, generate line graphs for Apple and Google adjusted closing prices. Show the rangeslider, but don't show the gridlines.Label the graph as shown below

In [ ]:
```
fig = go.Figure()
fig.add_trace(go.Scatter(x=Stocks['Date'], y=Stocks['Adj Close_aapl'],mode=

fig.add_trace(go.Scatter(x=Stocks['Date'], y=Stocks['Adj Close_goog'],
                    mode='lines',
                    name='Google'))

fig.update_xaxes(rangeslider_visible = True)

fig.update_layout(xaxis=dict(showline=True,showgrid=False),
  yaxis=dict(
        showgrid=False,
        showline=False,
        showticklabels=False),
  legend=dict(title='Stocks'),)


fig.update_layout(title= 'Apple vs Google Closing Prices',
                  xaxis_title='Day',
                  yaxis_title='Price')


fig.show()
```

The dataset AIrlines3 shows the delay time for departures for United (UA) and American Airlines (AA) for select days in June 2023

In [ ]:
```
FL=pd.read_excel('Airline3.xlsx')
FL.head()
```

Q8.Calculate the summary statistics for the departure delay times for United and American Airline

In [ ]:
```
FLSUMM=FL.groupby('AIRLINE_CODE')['DEP_DELAY'].describe()
FLSUMM
```

Q9. Using Plotly, generate a histogram for the departure delays for American and United airlines and include the 'rug' plot. Set the opacity to 0.35 and the x-axis range from -30 to 150

In [ ]:
```
#Histogram with rug plot
fig = px.histogram(FL, x="DEP_DELAY", color="AIRLINE_CODE",marginal='rug') #
fig.update_layout(barmode='overlay')
fig.update_traces(opacity=0.35)
fig.update_xaxes(range=[-30, 150])

fig.show()
```

Q10. Using Plotly generate box-plots to show the departure delay times for American and United Airlines. Include all data-points and use 'Blue' and 'Green'to distinguish the two airlines. Set the y-axis range to -100 to 2500.

In [ ]:
```
import plotly.express as px
fig = px.box(FL, x="AIRLINE_CODE", y="DEP_DELAY", points='all', color="AIRLI
fig.update_yaxes(range=[-100, 2500])

fig.show()
```

Q.11 Is there a statistical difference between average departure dela time for American and United Airlines? Run a two-sample t-test.

In [ ]:
```
FL1=FL[FL['AIRLINE_CODE']=='UA']['DEP_DELAY']
FL2=FL[FL['AIRLINE_CODE']=='AA']['DEP_DELAY']
```

In [ ]:
```
stats.ttest_ind(a=FL1, b=FL2,equal_var=True)
```

Q12. If Federal Aviation was considering imposing a penalty on the airlines for any departure delays more than 50 minutes, what proportion of American and United fights will be penalized? Create a new variable, 'PenDel' that would classify a fight as being 'Penalized' or non penalized ('Non-Pen').

In [ ]:
```
FL['PenDel']=['Penalized' if i >=50  else 'Non-Pen' for i in FL['DEP_DELAY']
FL.head()
```

In [ ]:
```
FL['PenDel'].value_counts(normalize=True)
```

The dataset WHR2 is drawn from the World Happiness Report.

In [ ]:
```
whr=pd.read_excel('WHR2.xlsx')
whr.head()
```

In [ ]:
```
whr.columns
```

Q13.Extract a subset of variables from the dataframe to include 'Life Ladder', 'Log GDP per capita','Healthy life expectancy at birth','Generosity','Democratic Quality'and store them in a new dataframe whr_core

In [ ]:
```
whr_core=whr[['Life Ladder', 'Log GDP per capita','Healthy life expectancy a
whr_core.head()
```

Q14. In the World Happiness Report, the Cantril 'life ladder' represents a measure of 'happiness' where top of the ladder represents the best possible life for a country's citizen and the bottom of the ladder represents the worst possible life. What are the factors that determine "happiness? Run a multiple regression model to test if Life Ladder (the dependent variable) is affected by 'Log GDP per capita','Healthy life expectancy at birth','Generosity','Democratic Quality.

In [ ]:
```
import statsmodels.api as sm

explanatory_var=['Log GDP per capita','Healthy life expectancy at birth', 'G
x=whr[explanatory_var]
y=whr['Life Ladder']
x=sm.add_constant(x)
model=sm.OLS(y,x,missing='drop')
results=model.fit()
results.params
print(results.summary())
```

Q15.Identify which variables are significant at an alpha of 0.05.

Q16. Based on your model, what is the effect on the Life Ladder if Generosity increased by 1 unit?

In [ ]:
```

```