```
import pandas as pd
from matplotlib import pyplot as plt
pd.set_option("display.max.rows",None)
pd.set_option("max_colwidth",None)
#Q1
print ("Q1")
df = pd.read_csv (r"world_population.csv")
print (df) ##print out csv file
print("#####################################################################
#####################################")
#Q2
print ("Q2")
a = df.iloc[:,[2,6,12]] ##set choose which data
a['Increase%'] = a.apply(lambda x: round(((x['2020 Population'] - x['1970
Population']) / x['1970 Population']) * 100, 2) , axis=1)##claculate the
difference and calcualte the %
lowestcountry1 = a.sort_values(by = ['Increase%']).head(5) ##fina top 5
topcountry1 = a.sort_values(by = ['Increase%']).tail(5) ##find lowest5
print("HIghest 5")
print(topcountry1)
print()
print("Lowest 5")
print(lowestcountry1)
print("#####################################################################
#####################################")
#Q3.1
print ("Q3.1")
a = df.iloc[:,[2,6,8]]
a['Increase%'] = a.apply(lambda x: round(((x['2020 Population'] - x['2010
Population']) / x['2010 Population']) * 100, 2) , axis=1)
lowestcountry2 = a.sort_values(by = ['Increase%']).head(5)
topcountry2 = a.sort_values(by = ['Increase%']).tail(5)
print("Highest 5")
print(topcountry2)
print()
print("Lowest 5")
print(lowestcountry2)
print("Not totally the same. Just Turks and Caicos Islands,Qatar and Vatican
City appear twice")
print("#####################################################################
#####################################")
#Q4
print ("Q4")
```

```
maxcountry = df.sort_values("2020 Population", ascending=False).head(5)
##sort the value
mincountry = df.sort_values("2020 Population", ascending=False).tail(5)
print(maxcountry)
print(mincountry)
print("#####################################################################
#####################################")
#5
print ("Q5")
maxcountry1 = df.sort_values("1970 Population", ascending=False).head(5)
mincountry1= df.sort_values("1970 Population", ascending=False).tail(5)
maxcountry2 = df.sort_values("2022 Population", ascending=False).head(5)
mincountry2= df.sort_values("2022 Population", ascending=False).tail(5)
print("1970")
print('Highest')
print(maxcountry1)
print('Lowest')
print(mincountry1)
print('2022')
print('Highest')
print(maxcountry2)
print('Lowest')
print(mincountry2)
print('Highest 5 remain:CHN,IND,USA,IDN')
print('Lowest 5 remain:FLK,VAT,NIU,TKL')
print("#####################################################################
#####################################")
#Q6
print("Q6")
a = df['2020 Population'].mean()#calculat the mean
b = df['2020 Population'].quantile([0.25,0.5,0.75])#calculate the Q1 Q2 Q3
print("The mean of 2020 population is ",a)
print("The Q1,Q2,Q3 of 2020 population are ")
print(b)
print("#####################################################################
#####################################")
#Q7
print("Q7")
print('Three countries around the mean are:Yemen,Malaysia and Ghana')
print('Three countries around the Q1 are:Belize,Bahamas,Buadeloupe')
print('Three countries around the M are:Norway,Lebanon,finland')
print('Three countries caround the Q3 are:Sri Lanka,Syria and Mali')
print("#####################################################################
#####################################")
```

```
#Q8
print("Q8")
a = df['2020 Population'].quantile([0.25,0.75])
b = df['2010 Population'].quantile([0.25,0.75])
c = df['2000 Population'].quantile([0.25,0.75])
d = df['1990 Population'].quantile([0.25,0.75])
e = df['1980 Population'].quantile([0.25,0.75])
print("The Q1,Q3 of 2020 population are ")
print(a)
print()
print("The Q1,Q3 of 2010 population are ")
print(b)
print()
print("The Q1,Q3 of 2000 population are ")
print(c)
print()
print("The Q1,Q3 of 1990 population are ")
print(d)
print()
print("The Q1,Q3 of 1980 population are ")
print(e)
print()
print("Picture in pdf file")
print("##########################################################################
#####################################")
#Q8.1
df_1980 = df[['CCA3','Country','1980 Population']]
Q1_1980 = df_1980['1980 Population'].quantile(0.25)
Q3_1980 = df_1980['1980 Population'].quantile(0.75)
df_1980Q1Q3 = df_1980[(df_1980['1980 Population']>=Q1_1980)
&(df_1980['1980 Population']<=Q3_1980)] ##find the value lower than Q3 and
higher than Q1
plt.bar(df_1980Q1Q3['Country'],df_1980Q1Q3['1980 Population'],color =
['red'])
plt.tick_params(axis='x',labelsize= 2) ##set the form
plt.xticks(rotation=-90)
plt.savefig('1980 population')
plt.figure()

df_1990 = df[['CCA3','Country','1990 Population']]
Q1_1990 = df_1990['1990 Population'].quantile(0.25)
Q3_1990 = df_1990['1990 Population'].quantile(0.75)
df_1990Q1Q3 = df_1990[(df_1990['1990 Population']>=Q1_1990)
&(df_1990['1990 Population']<=Q3_1990)]
```

```
plt.bar(df_1990Q1Q3['Country'],df_1990Q1Q3['1990 Population'],color =
['black'])
plt.tick_params(axis='x',labelsize= 2)
plt.xticks(rotation=-90)
plt.savefig('1990 population')
plt.figure()
```

```
df_2000 = df[['CCA3','Country','2000 Population']]
Q1_2000 = df_2000['2000 Population'].quantile(0.25)
Q3_2000 = df_2000['2000 Population'].quantile(0.75)
df_2000Q1Q3 = df_2000[(df_2000['2000 Population']>=Q1_2000)
&(df_2000['2000 Population']<=Q3_2000)]
plt.bar(df_2000Q1Q3['Country'],df_2000Q1Q3['2000 Population'],color =
['cyan'])
plt.tick_params(axis='x',labelsize= 2)
plt.xticks(rotation=-90)
plt.savefig('2000 population')
plt.figure()
```

```
df_2010 = df[['CCA3','Country','2010 Population']]
Q1_2010 = df_2010['2010 Population'].quantile(0.25)
Q3_2010 = df_2010['2010 Population'].quantile(0.75)
df_2010Q1Q3 = df_2010[(df_2010['2010 Population']>=Q1_2010)
&(df_2010['2010 Population']<=Q3_2010)]
plt.bar(df_2010Q1Q3['Country'],df_2010Q1Q3['2010 Population'],color =
['blue'])
plt.tick_params(axis='x',labelsize= 2)
plt.xticks(rotation=-90)
plt.savefig('2010 population')
plt.figure()
```

```
df_2020 = df[['CCA3','Country','2020 Population']]
Q1_2020 = df_2020['2020 Population'].quantile(0.25)
Q3_2020 = df_2020['2020 Population'].quantile(0.75)
df_2020Q1Q3 = df_2020[(df_2020['2020 Population']>=Q1_2020)
&(df_2020['2020 Population']<=Q3_2020)]
plt.bar(df_2020Q1Q3['Country'],df_2020Q1Q3['2020 Population'],color =
['green'])
plt.tick_params(axis='x',labelsize= 2)
plt.xticks(rotation=-90)
plt.savefig('2020 population')
plt.figure()
print("##################################################################
#####################################")
```

```
#Q9
print("Q9")
print('They look very similar')
print("##########################################################################
#####################################")
#Q10
print("Q10")
density = df[['Country','Density (per km²)']]
desntiy = density.copy()
density['Rank']=desntiy['Density (per km²)'].rank()
density.sort_values('Rank',inplace = True, ascending = True)
print('The rank of density is')
print(density)
print("##########################################################################
#####################################")
#Q11
print('Q11')
population1970and2020 = df[['Country','1970 Population','2020
Population']]
population1970and2020 = population1970and2020.copy()
population1970and2020['1970 Rank'] = population1970and2020['1970
Population'].rank()
print('The rank of 1970')
print(population1970and2020)
population1970and2020['2020 Rank'] = population1970and2020['2020
Population'].rank()
population1970and2020['Change'] = population1970and2020.apply(lambda x:
x['2020 Rank']-x['1970 Rank'],axis = 1)
print('Top 5 largest positive change')
print(population1970and2020.nlargest(5,'Change'))
print('Top 5 largest negative change')
print(population1970and2020.nsmallest(5,'Change'))
print("##########################################################################
#####################################")
#Q12
print("Q12")
def getFirstDigit(num):
    while num >= 10:
        num = int(num/10)
    return num
first_digit = 1
count = 0
for i in df["2020 Population"].index:
```

```
    first_every = getFirstDigit(df["2020 Population"][i])
    if first_every == first_digit:
        count += 1
    else:
        continue
answer = round(count / len(df['2020 Population']) * 100 ,2)
print("Digit 1 has",answer ,"%")
```

```
def getFirstDigit(num):
    while num >= 10:
        num = int(num/10)
    return num
first_digit = 2
count = 0
for i in df["2020 Population"].index:
    first_every = getFirstDigit(df["2020 Population"][i])
    if first_every == first_digit:
        count += 1
    else:
        continue
answer = round(count / len(df['2020 Population']) * 100 ,2)
print("Digit 2 has",answer ,"%")
```

```
def getFirstDigit(num):
    while num >= 10:
        num = int(num/10)
    return num
first_digit = 3
count = 0
for i in df["2020 Population"].index:
    first_every = getFirstDigit(df["2020 Population"][i])
    if first_every == first_digit:
        count += 1
    else:
        continue
answer = round(count / len(df['2020 Population']) * 100 ,2)
print("Digit 3 has",answer ,"%")
```

```
def getFirstDigit(num):
    while num >= 10:
        num = int(num/10)
    return num
first_digit = 4
count = 0
```

```
for i in df["2020 Population"].index:
    first_every = getFirstDigit(df["2020 Population"][i])
    if first_every == first_digit:
        count += 1
    else:
        continue
answer = round(count / len(df['2020 Population']) * 100 ,2)
print("Digit 4 has",answer ,"%")
```

```
def getFirstDigit(num):
    while num >= 10:
        num = int(num/10)
    return num
first_digit = 5
count = 0
for i in df["2020 Population"].index:
    first_every = getFirstDigit(df["2020 Population"][i])
    if first_every == first_digit:
        count += 1
    else:
        continue
answer = round(count / len(df['2020 Population']) * 100 ,2)
print("Digit 5 has",answer ,"%")
```

```
def getFirstDigit(num):
    while num >= 10:
        num = int(num/10)
    return num
first_digit = 6
count = 0
for i in df["2020 Population"].index:
    first_every = getFirstDigit(df["2020 Population"][i])
    if first_every == first_digit:
        count += 1
    else:
        continue
answer = round(count / len(df['2020 Population']) * 100 ,2)
print("Digit 6 has",answer ,"%")
```

```
def getFirstDigit(num):
    while num >= 10:
        num = int(num/10)
    return num
first_digit = 7
```

```
count = 0
for i in df["2020 Population"].index:
    first_every = getFirstDigit(df["2020 Population"][i])
    if first_every == first_digit:
        count += 1
    else:
        continue
answer = round(count / len(df['2020 Population']) * 100 ,2)
print("Digit 7 has",answer ,"%")
```

```
def getFirstDigit(num):
    while num >= 10:
        num = int(num/10)
    return num
first_digit = 8
count = 0
for i in df["2020 Population"].index:
    first_every = getFirstDigit(df["2020 Population"][i])
    if first_every == first_digit:
        count += 1
    else:
        continue
answer = round(count / len(df['2020 Population']) * 100 ,2)
print("Digit 8 has",answer ,"%")
```

```
def getFirstDigit(num):
    while num >= 10:
        num = int(num/10)
    return num
first_digit = 9
count = 0
for i in df["2020 Population"].index:
    first_every = getFirstDigit(df["2020 Population"][i])
    if first_every == first_digit:
        count += 1
    else:
        continue
answer = round(count / len(df['2020 Population']) * 100 ,2)
print("Digit 9 has",answer ,"%")
print("All answer round 2 decimal")
```