Stat_HW4_Solution

```
In [1]: #載入相關函示庫
        from matplotlib import pyplot as plt
        %matplotlib inline
        # 設定圖形大小 dpi越大圖越大
        plt.rcParams['figure.dpi'] = 100
        import seaborn as sns
        import pandas as pd
        import numpy as np
        import scipy.stats as stats
        import statsmodels.api as sm
        import statsmodels.stats.api as sms
        import statsmodels.formula.api as smf
        import math as math
```

Exercise 4.61 (2% for each quartile)

```
In [2]: # This is a function that computes percentile from input data.
        def percentile(data1, perc):
            if type(data1) == np.ndarray:
                alldata = data1.copy()
                data1 = data1.copy()
            else:
                alldata = data1.values.copy()
                data1 = data1.values.copy()
            alldata.sort()
            n = alldata.size
            i = (n-1) * p / 100 - 1
            f_i = math.floor(i)
            c_i = math.ceil(i)
            percentile_v = alldata[f_i] + (alldata[c_i] - alldata[f_i]) * (i - f_i)
            return percentile_v

        r1 = np.array([10.5, 14.7, 15.3, 17.7, 15.9, 12.2, 10.9, 14.1, 13.9, 18.5, 13.9, 15.1, 14.7])
        p25 = percentile(r1, 25)
        p50 = percentile(r1, 50)
        p75 = percentile(r1, 75)
        print('Q1 = ', p25)
        print('Q2 = ', p50)
        print('Q3 = ', p75)

        Q1 =  13.05
        Q2 =  14.7
        Q3 =  15.600000000000001
```

Exercise 4.39

(3% for range, 3% for variance, 3% for stdandard deviation, 3% for interpretation)

```
In [11]: #讀取資料集
         df_hr0439 = pd.read_excel('Hr04-39.xlsx')
         print('Head of dataset:')
         display(df_hr0439.head())
         print('Tail of dataset:')
         display(df_hr0439.tail())

         print('Ranges: ',float(df_hr0439.max()-df_hr0439.min()),'dollars(per 100 pills)')
         print('Variance: ',float(df_hr0439.var()),'dollars^2(per 100 pills)')
         print('Standard deviation: ',float(df_hr0439.std()),'dollars(per 100 pills)')
```

Head of dataset:

	Prozac
0	102.05
1	99.85
2	113.30
3	97.15
4	111.23

Tail of dataset:

	Prozac
95	100.81
96	109.56
97	103.35
98	113.02
99	103.79

```
Ranges: 25.850000000000001 dollars(per 100 pills)
Variance:  29.455223222222216 dollars^2(per 100 pills)
Standard deviation:  5.427266643737179 dollars(per 100 pills)
```

Exercise 4.45

(3% for mean, 3% for standard deviation, 3% for interpretation)

```
In [4]: #讀取資料集
        df_hr0445 = pd.read_excel('Hr04-45.xlsx')
        print('Head of dataset:')
        display(df_hr0445.head())
        print('Tail of dataset:')
        display(df_hr0445.tail())

        print('Mean: ',float(df_hr0445.mean()),'minutes')
        print('Standard deviation: ',float(df_hr0445.std()),'minutes')
```

Head of dataset:

	Flight delay (minutes)
0	37
1	29
2	18
3	45
4	36

Tail of dataset:

	Flight delay (minutes)
120	0
121	30
122	47
123	14
124	32

```
Mean:  26.024 minutes
Standard deviation:  11.807231245193202 minutes
```

Exercise 4.63 (3%)

```
In [5]: r1 = np.array([10.5, 14.7, 15.3, 17.7, 15.9, 12.2, 10.9, 14.1, 13.9, 18.5, 13.9, 15.1, 14.7])
        p25 = percentile(r1, 25)
        p75 = percentile(r1, 75)
        print('InterQuartile ranges: ', p75-p25)

        InterQuartile ranges:  2.5500000000000007
```

Exercise 4.69 (2% for each quartile, 3% for interpretation)

```
In [6]: #讀取資料集
        df_hr0469 = pd.read_excel('Hr04-69.xlsx')
        dog=df_hr0469['Dogs']
        cat=df_hr0469['Cats'].dropna()

        dog_p25=percentile(dog,25)
        dog_p50=percentile(dog,50)
        dog_p75=percentile(dog,75)

        cat_p25=percentile(cat,25)
        cat_p50=percentile(cat,50)
        cat_p75=percentile(cat,75)

        print('Quartiles of dog\nFirst:',dog_p25,'dollars\nSecond: ',dog_p50,'dollars\nThird: ',dog_p75,'dollars')
        print('\nQuartiles of cat\nFirst:',cat_p25,'dollars\nSecond: ',cat_p50,'dollars\nThird: ',cat_p75,'dollars')
```

```
Quartiles of dog
First: 1097.5 dollars
Second:  1250.0 dollars
Third:  1337.0 dollars

Quartiles of cat
First: 773.0 dollars
Second:  846.0 dollars
Third:  988.0 dollars
```

Exercise 4.73 (2% for each quartile, 3% for interpretation)

```
In [7]: #讀取資料集
        df_hr473 = pd.read_excel('Hr04-73.xlsx')
        private=df_hr473['Private']
        public=df_hr473['Public'].dropna()

        private_p25=percentile(private,25)
        private_p50=percentile(private,50)
        private_p75=percentile(private,75)

        public_p25=percentile(public,25)
        public_p50=percentile(public,50)
        public_p75=percentile(public,75)

        print('Quartiles of private\nFirst:',private_p25,'minutes\nSecond: ',private_p50,'minutes\nThird: ',private_p75,'minutes')
        print('\nQuartiles of public\nFirst:',public_p25,'minutes\nSecond: ',public_p50,'minutes\nThird: ',public_p75,'minutes')
```

```
Quartiles of private
First: 228.0 minutes
Second:  237.0 minutes
Third:  245.75 minutes

Quartiles of public
First: 279.0 minutes
Second:  296.0 minutes
Third:  307.0 minutes
```

Exercise 4.91 (Compute and interpret covariance and r)

(2% for scatter plot, 3% for covariance, 3% for correlation , 3% for interpretation)

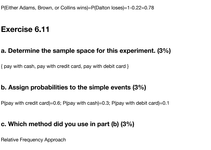
```
In [8]: #讀取資料集
        df_hr491 = pd.read_excel('Hr04-91.xlsx')

        print('Head of dataset:')
        display(df_hr491.head())

        print('Tail of dataset:')
        display(df_hr491.tail())

        _ = sns.regplot(x='Price of Oil', y= 'Oil Wells', data = df_hr491, scatter_kws={"color": 'black'}, line_kws={"color": "red"}, ci = None)
        plt.title('Scatter Plot for Price of Oil and Oil Wells Drilled')
        plt.xlabel('Price of Oil (Dollar)')
        plt.ylabel('Number of Oil Wells Drilled')
        plt.show()

        #Compute the covariance matrix
        cov_mat = np.cov(df_hr491[['Price of Oil', 'Oil Wells']].values,rowvar = False)
        print(cov_mat)
        #Compute the correlation matrix
        cor_mat = np.corrcoef(df_hr491[['Price of Oil', 'Oil Wells']].values,rowvar = False)
        print(cor_mat)
```

Head of dataset:

	Year	Month	Price of Oil	Oil Wells
0	1973	1	3.56	58
1	1973	2	3.56	52
2	1973	3	3.56	53
3	1973	4	3.56	28
4	1973	5	3.56	51

Tail of dataset:

	Year	Month	Price of Oil	Oil Wells
451	2010	8	76.82	56
452	2010	9	75.31	57
453	2010	10	81.80	75
454	2010	11	84.14	62
455	2010	12	89.04	57




```
[[ 449.62162871   23.02331203]
 [  23.02331203  2762.31563942]]
[[1.         0.02021423]
 [0.02021423 1.        ]]
```

Exercise 4.99 (Compute and interpret covariance and r)

(2% for scatter plot, 3% for covariance, 3% for correlation, 3% for interpretation)

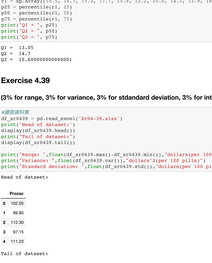
```
In [9]: #讀取資料集
        df_hr499 = pd.read_excel('Hr04-99.xlsx')

        print('Head of dataset:')
        display(df_hr499.head())

        print('Tail of dataset:')
        display(df_hr499.tail())

        _ = sns.regplot(x='Age', y= 'Carbon monoxide', data = df_hr499, scatter_kws={"color": 'black'}, line_kws={"color": 'red'}, ci = None)
        plt.title('Scatter Plot for Age of Furnace and Carbon Monoxide')
        plt.xlabel('Age')
        plt.ylabel('Carbon monoxide (ppm)')
        plt.show()

        #Compute the covariance matrix
        cov_mat = np.cov(df_hr499[['Age', 'Carbon monoxide']].values,rowvar = False)
        print(cov_mat)
        #Compute the correlation matrix
        cor_mat = np.corrcoef(df_hr499[['Age', 'Carbon monoxide']].values,rowvar = False)
        print(cor_mat)
```

Head of dataset:

	Age	Carbon monoxide
0	35	5.4
1	28	7.8
2	39	11.7
3	31	7.2
4	11	1.0

Tail of dataset:

	Age	Carbon monoxide
175	34	8.5
176	26	7.3
177	27	8.5
178	37	8.3
179	39	10.5




```
[[103.42939168  32.85101242]
 [ 32.85101242  18.43070631]]
[[1.         0.75153594]
 [0.75153594 1.        ]]
```

Exercise 6.7

a. Adam Loses (3%)

P(Adams loses) = P(Brown wins) + P(Collins wins) + P(Dalton wins) = 0.09 + 0.27 + 0.22 = 0.58

b. Either Brown or Dalton wins (3%)

P(either Brown or Dalton win)) = P(Brown wins) + P(Dalton wins) = 0.09 + 0.22 = 0.31

c. Adam, Brown, or Collins wins (3%)

P(Either Adams, Brown, or Collins wins)=P(Dalton loses)=1-0.22=0.78

Exercise 6.11

a. Determine the sample space for this experiment. (3%)

{ pay with cash, pay with credit card, pay with debit card }

b. Assign probabilities to the simple events (3%)

P(pay with credit card)=0.6; P(pay with cash)=0.3; P(pay with debit card)=0.1

c. Which method did you use in part (b) (3%)

Relative Frequency Approach