Statistics Homework 04

B08705034 資管二 施芊羽

Chapter 4

4.61

In [2]:
```
from matplotlib import pyplot as plt
%matplotlib inline
# 設定圖形大小; DPI越大圖越大
plt.rcParams["figure.dpi"] = 150
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
import statistics
```

In [4]:
```
df_c04_61 = pd.read_excel("Xr04-61.xlsx")
df_c04_61 = df_c04_61.sort_values(by = 'X')

def percentile(data, p, n): #n is the number of data to calculate
    if type(data) == np.ndarray:
        alldata = data.copy()
        #data1 = data.copy()
    else:
        alldata = data.values.copy()
        #data1 = data.values.copy()
    alldata.sort(kind = 'quicksort')
    l = (n + 1) * p / 100 - 1
    f_l = math.floor(l)
    c_l = math.ceil(l)
    percentile_v = alldata[f_l] + (alldata[c_l] - alldata[f_l]) * (l - f_l)
    return percentile_v

n = df_c04_61.size -  df_c04_61.isnull().values.sum()

p25 = percentile(df_c04_61, 25, n)
p50 = percentile(df_c04_61, 50, n)
p75 = percentile(df_c04_61, 75, n)
print("25th percentile = ", p25)
print("50th percentile = ", p50)
print("75th percentile = ", p75)
```

```
25th percentile =  [13.05]
50th percentile =  [14.7]
75th percentile =  [15.6]
```

From the given data, I use the function `percentile(data, p)` and make a little modification to `percentile(data, p, n)` to solve some other kind of problems in other exercises.

After the program, we've got that the $25-$th percentile is 13.05, the $50-$th percentile is 14.7 and the 75th percentile is 15.6. *All round to the two decimal places*

4.39

In [6]:
```
df_c04_39 = pd.read_excel("Xr04-39.xlsx")

mean = np.mean(df_c04_39["Prozac"])
var = statistics.variance(df_c04_39["Prozac"])
std = statistics.stdev(df_c04_39["Prozac"])

print("Range = " ,df_c04_39["Prozac"].max() - df_c04_39["Prozac"].min())
print("Mean = ", np.round(mean, 2))
print("Variation = ", np.round(var, 2))
print("Standard Deviation = ", np.round(std,2))

df_c04_39.describe()
```

```
Range =  25.85000000000001
Mean =  106.49
Variation =  29.46
Standard Deviation =  5.43
```

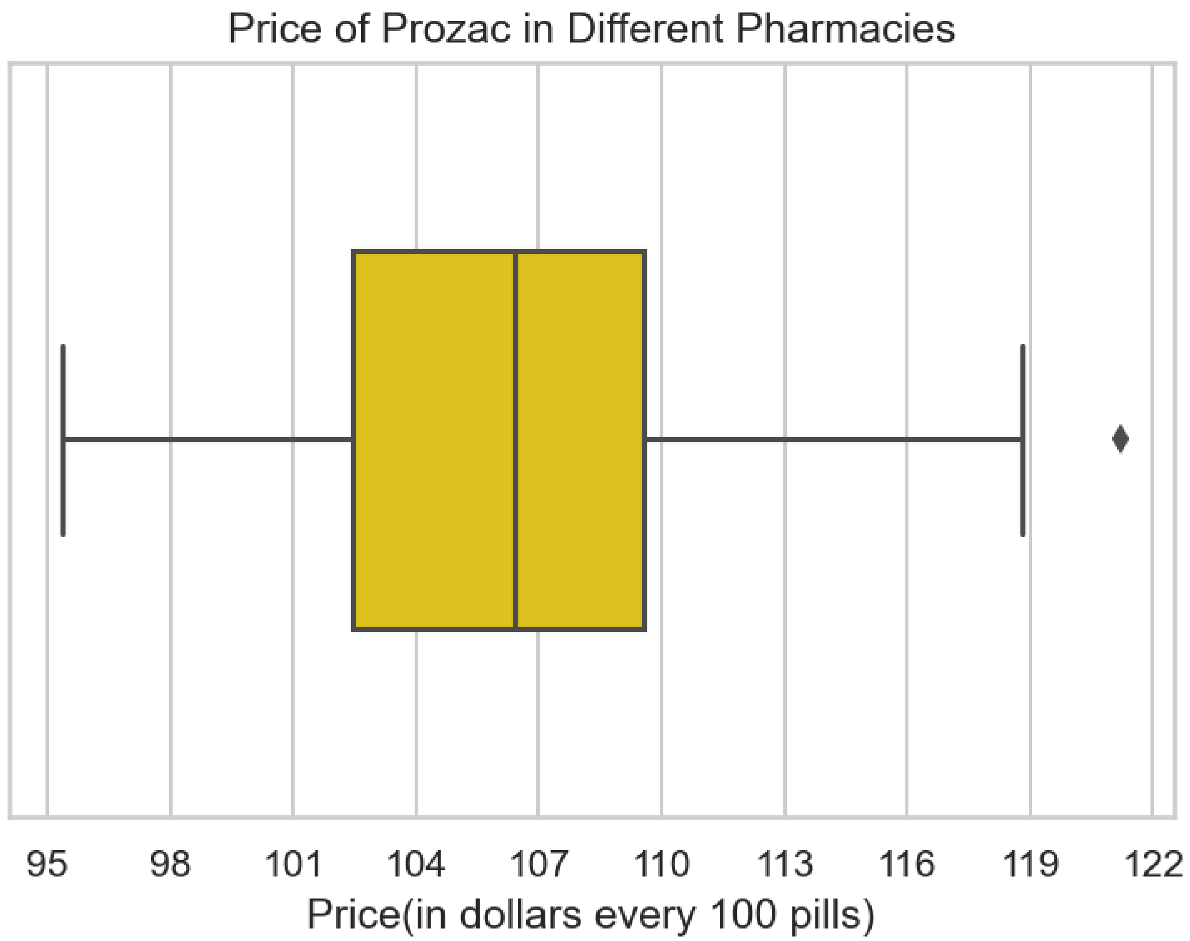
Out[6]:

	Prozac
count	100.000000
mean	106.490100
std	5.427267
min	95.350000
25%	102.467500
50%	106.430000
75%	109.560000
max	121.200000

In [7]:
```
def outlier(data_k, n):
    Q1 = percentile(data_k, 25, n)
    Q2 = percentile(data_k, 50, n)
    Q3 = percentile(data_k, 75, n)
    IQR = Q3 - Q1      #IQR is interquartile range.
    print("Q1 = ", Q1)
    print("Q2 = ", Q2)
    print("Q3 = ", Q3)
    print("IQR = ", IQR)
    filter = (data_k < Q1 - 1.5 * IQR) | (data_k > Q3 + 1.5 * IQR)
    print("Outliers are listed as follows \n")
    print(data_k.loc[filter])

plt.title("Price of Prozac in Different Pharmacies")
sns.set(style="whitegrid")
```

```
ax = sns.boxplot(x = df_c04_39["Prozac"], color = "gold", width = 0.5)
plt.xlabel("Price(in dollars every 100 pills)")
plt.xticks(np.arange(95, 125, 3), np.arange(95, 125, 3))
print(" \n")
print("Outliers of Bills \n")
outlier(df_c04_39["Prozac"], df_c04_39.size)
```

```
Outliers of Bills

Q1 =  102.46249999999999
Q2 =  106.43
Q3 =  109.56
IQR =  7.097500000000011
Outliers are listed as follows

93    121.2
Name: Prozac, dtype: float64
```



The statistics that the description required to answer:

		Unit
Range	25.85	Dollars(per 100 pills)
Mean	106.49	Dollars(per 100 pills)
Variation	29.46	$Dollars^2$
Standard Deviation	5.43	Dollars(per 100 pills)

From these statistics, we can know that the average price of Prozac around U.S. is 106.49 with a range 25.85 which means there's still a 25 dollars difference of price in different pharmacies. The variation and standard deviation are hard to compare in this case since there's only one data in the comparison. However, we can know that the range is smaller than 6 standard deviation, we can guess that the data doesn't distribute very seperately. In addition, I've drawn a box and whisker plot to graphically understand the statistics.

-1

by chebyshev ?

4.45

-2

In [8]:
```
df_c04_45 = pd.read_excel("Xr04-45.xlsx")

mean = np.mean(df_c04_45["Flight delay (minutes)"])
std = statistics.stdev(df_c04_45["Flight delay (minutes)"])

print("Mean = ", np.round(mean,2))
print("Standard Deviation = ", np.round(std,2))

df_c04_45.describe()
```

```
Mean =  26.02
Standard Deviation =  11.81
```

Out[8]:

	Flight delay (minutes)
count	125.000000
mean	26.024000
std	11.807231
min	-10.000000
25%	20.000000
50%	28.000000
75%	33.000000
max	49.000000

		Unit
Mean	26.02	minutes
Standard Deviation	11.81	minutes

68, 95, all ...

-2

Round to the two decimal spaces

Since we assume that the distribution is approximately bell shaped, we can know that the modal class of the data will include the mean 26.02 minutes. The standard deviation can tell us that the distribution of the data is still big since the mean is just abuout twice the standard deviation, I can guess that the kurtosis of the graph might be low.

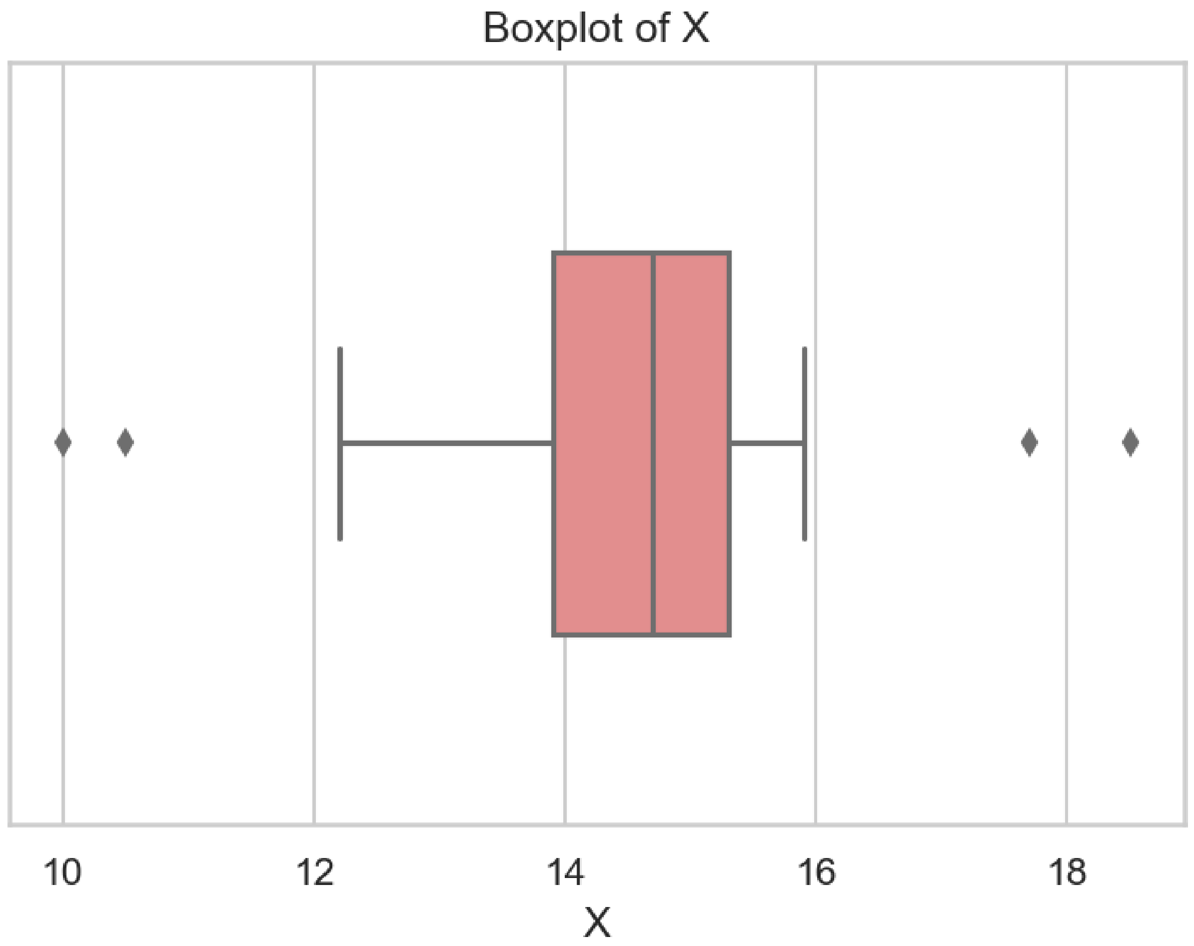
4.63

-0

In [9]:
```
plt.title("Boxplot of X")
```

```
ax = sns.boxplot(x = df_c04_61["X"], color = "lightcoral", width = 0.5)

print(" \n")
print("Outliers of Bills \n")
outlier(df_c04_61["X"], df_c04_61.size)
```

```
Outliers of Bills

Q1 =  13.05
Q2 =  14.7
Q3 =  15.600000000000001
IQR =  2.5500000000000007
Outliers are listed as follows

Series([], Name: X, dtype: float64)
```



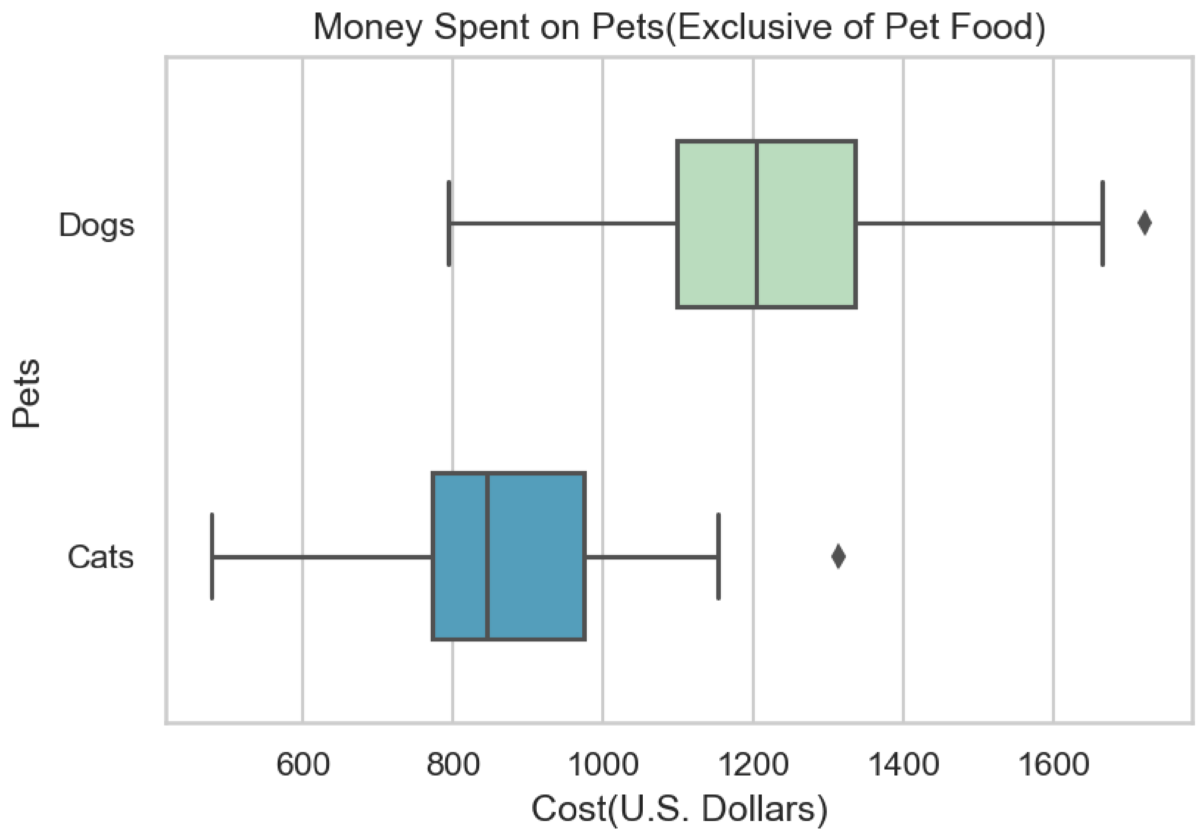
The inter quartile of the given data is 2.55 after rounded to the two decimal places.

4.69

-2

In [152]:
```
df_c04_69 = pd.read_excel("Xr04-69.xlsx")


df_c04_69_new = df_c04_69.rename(columns = {'Dogs': 'Money_Dogs', "Cats"
: "Money_Cats"})
#需要一個ID欄位
df_c04_69_new["id"] = df_c04_69_new.index
```

```
#呼叫wide_to_long(); 文件請見: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.wide_to_long.html
df_c04_69_new = pd.wide_to_long(df_c04_69_new, ["Money"],  sep = "_", suffix = '\w+', i="id", j="Pets").reset_index()

plt.title("Money Spent on Pets(Exclusive of Pet Food)")
ax1 = sns.boxplot(y = "Pets", x = "Money", data = df_c04_69_new, orient = "h", palette = "GnBu", width = 0.5)
plt.xlabel("Cost(U.S. Dollars)")
plt.show()

n1 = df_c04_69["Dogs"].size -  df_c04_69["Dogs"].isnull().values.sum()
n2 = df_c04_69["Cats"].size -  df_c04_69["Cats"].isnull().values.sum()

print(" \n")
print("Outliers of Cost on Dogs \n")
outlier(df_c04_69["Dogs"], n1)
print("Outliers of Cost on Cats \n")
outlier(df_c04_69["Cats"], n2)
```




```
Outliers of Cost on Dogs

Q1 =  1097.5
Q2 =  1204.0
Q3 =  1337.0
IQR =  239.5
Outliers are listed as follows

23    1723
Name: Dogs, dtype: int64
Outliers of Cost on Cats

Q1 =  773.0
```

```
Q2 =  846.0
Q3 =  988.0
IQR =  215.0
Outliers are listed as follows

9    1315.0
Name: Cats, dtype: float64
```

The quartiles of the different kinds of pets are different.

	Dogs	Cats
Q1	1097.5	773
Q2	1204	846
Q3	1337	998

-2

skewness?

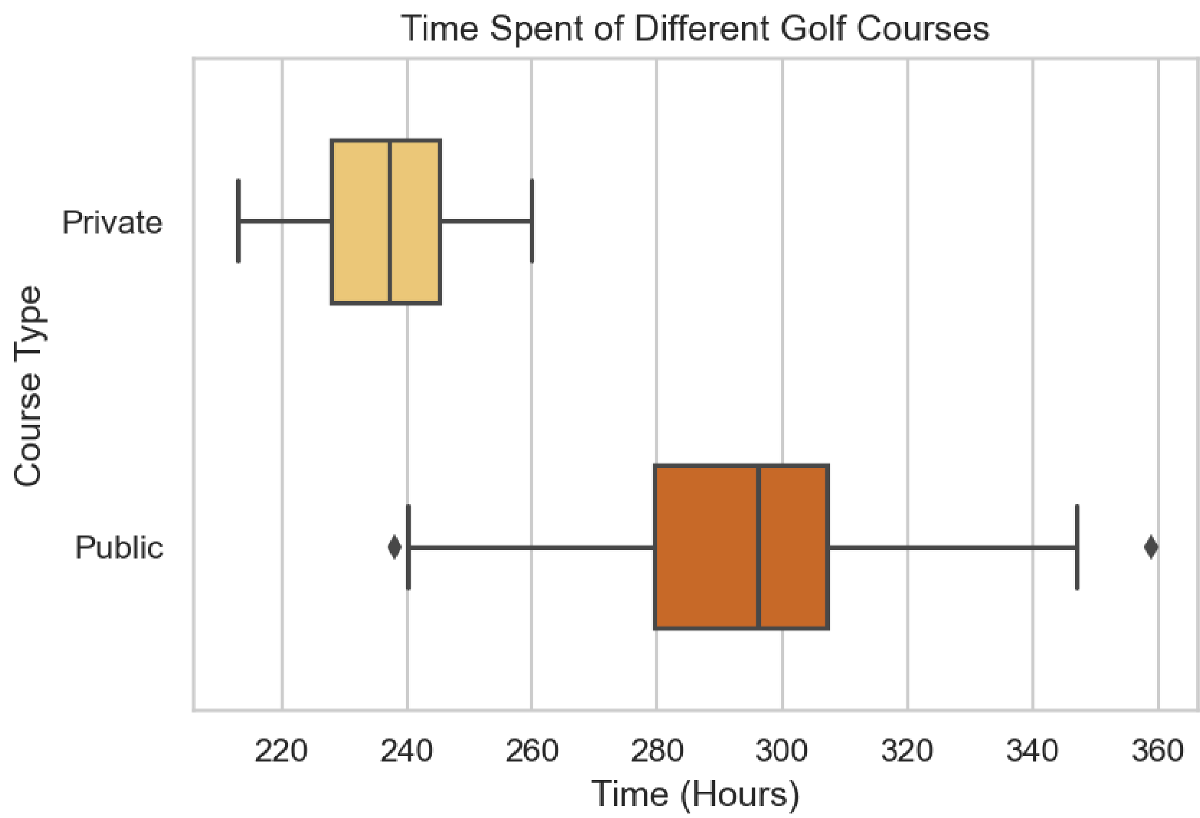
Unit: U.S. Dollars

We can see that all quartiles of the dogs are higher than those of the cats. We can guessed that it costs more to pet a dog than a cat. From the statistics, the quartiles of cost on petting dogs and on petting cats all has a difference larger than 200 U.S. dollars. The cost of dogs also has a higher IQR, we can know that the cost of dogs has a larger range in petting than cats.

4.73

-2

In [25]:
```
df_c04_73 = pd.read_excel("Xr04-73.xlsx")


df_c04_73_new = df_c04_73.rename(columns = {'Private': 'Time_Private', "Public": "Time_Public"})
df_c04_73_new["id"] = df_c04_73_new.index
df_c04_73_new = pd.wide_to_long(df_c04_73_new, ["Time"],  sep = "_", suffix = '\w+', i="id", j="Course Type").reset_index()

plt.title("Time Spent of Different Golf Courses")
ax = sns.boxplot(y = "Course Type", x = "Time", data = df_c04_73_new, orient = "h", palette = "YlOrBr", width = 0.5)
plt.xlabel("Time (Hours)")
plt.show()

n1 = df_c04_73["Private"].size -  df_c04_73["Private"].isnull().values.sum()
n2 = df_c04_73["Public"].size -  df_c04_73["Public"].isnull().values.sum()

print(" \n")
print("Outliers of Time Spent on Private Golf Courses \n")
outlier(df_c04_73["Private"], n1)
print("Outliers of Time Spent on Public Golf Courses \n")
outlier(df_c04_73["Public"], n2)
```




```
Outliers of Time Spent on Private Golf Courses

Q1 =  228.0
Q2 =  237.0
Q3 =  245.75
IQR =  17.75
Outliers are listed as follows

Series([], Name: Private, dtype: int64)
Outliers of Time Spent on Public Golf Courses

Q1 =  279.0
Q2 =  296.0
Q3 =  307.0
IQR =  28.0
Outliers are listed as follows

79    359.0
Name: Public, dtype: float64
```

-2

	Private	Public
Q1	228	279
Q2	237	296
Q3	245.75	307

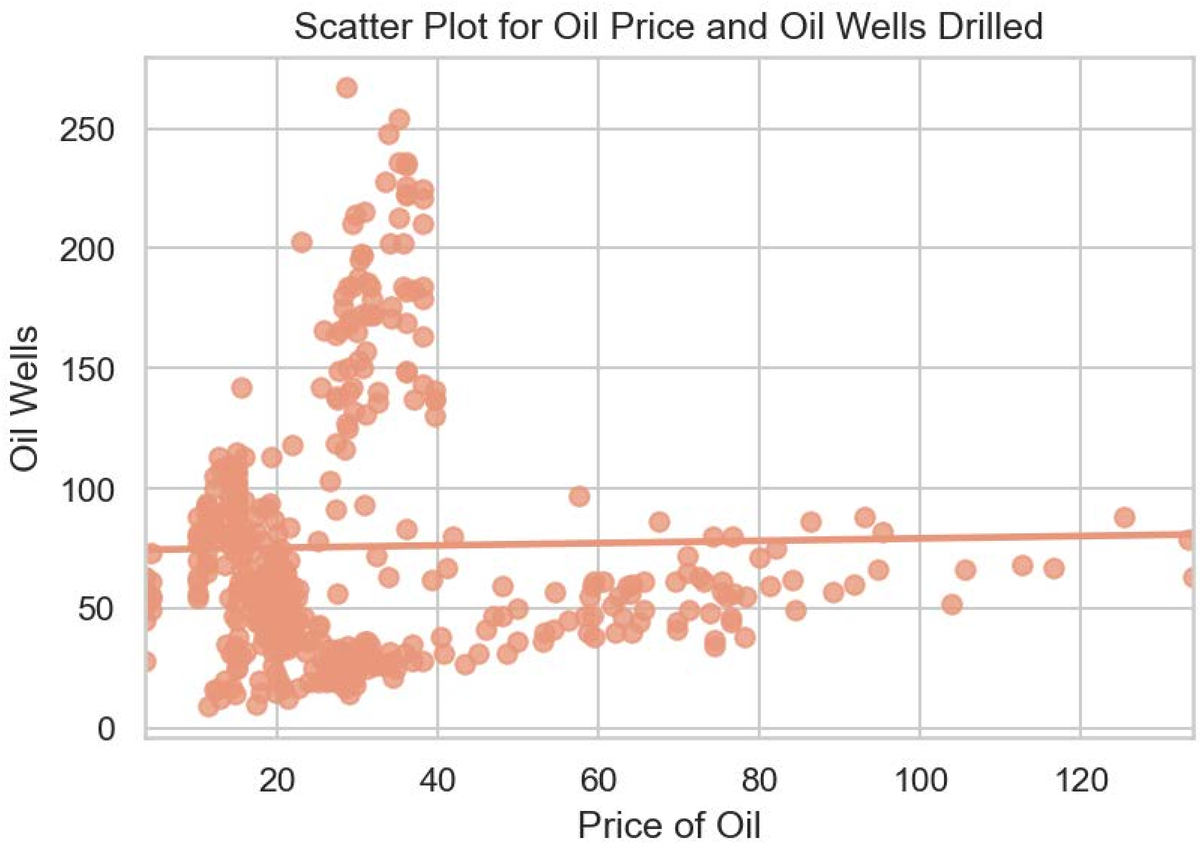
skewness?

Unit: Hours

From the statistics, we can know that the members of private courses really play faster than those of public courses from the quartiles we've got. For Q1 to Q3, we can see that the difference of two kinds of courses is all more than 50 hours. Hence, we can make a conclusion that the golfers that are members of private courses play faster than players on public courses but there are also some on public golfers can play faster than the other.

4.91

In [20]:
```
df_c04_91 = pd.read_excel('Xr04-91.xlsx')
_ = sns.regplot(x='Price of Oil', y= 'Oil Wells', data = df_c04_91, colo
r = 'darksalmon', ci = None)
plt.title('Scatter Plot for Oil Price and Oil Wells Drilled')
plt.show()
#Compute the covariance matrix
cov_mat = np.cov(df_c04_91[['Price of Oil', 'Oil Wells']].values, rowvar
 = False)
display(cov_mat)
#Compute the correlation matrix
cor_mat = np.corrcoef(df_c04_91[['Price of Oil', 'Oil Wells']].values, r
owvar = False)
display(cor_mat)
```




```
array([[ 469.62162871,   23.02331203],
       [  23.02331203, 2762.31563042]])

array([[1.        , 0.02021423],
       [0.02021423, 1.        ]])
```

The covariance matrix is $\begin{bmatrix} 69.62 & 23.0223.02 & 2762.32 \end{bmatrix}$ and the covariance between the oil price and the number of oil wells drilled is 23.02 which means the two variables are positively related. The r of the data is 0.02 which mean the linear relationship of the two variables are very weak.

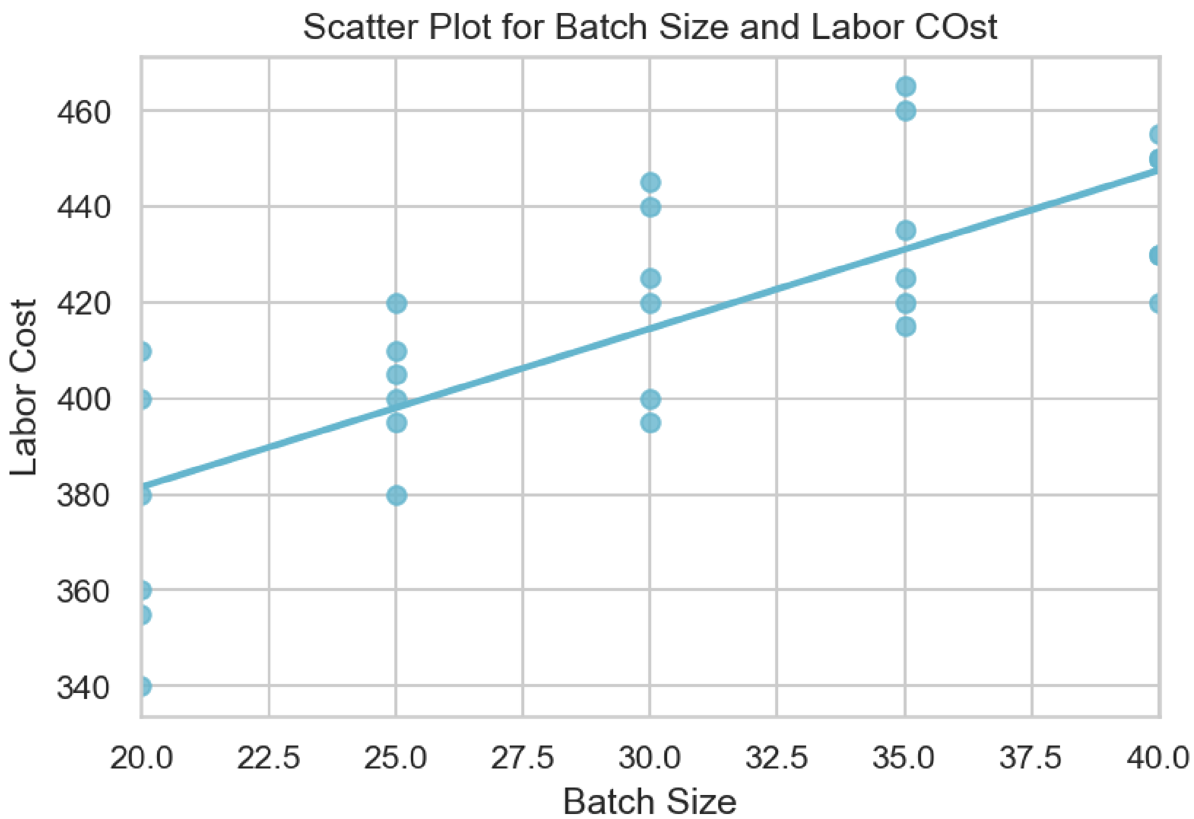
All round to the two decimal places.

4.99 -1

4.93

In [23]:
```
df_c04_93 = pd.read_excel('Xr04-93.xlsx')
_ = sns.regplot(x='Batch Size', y= 'Labor Cost', data = df_c04_93, color
 = 'c', ci = None)
plt.title('Scatter Plot for Batch Size and Labor COst')
plt.show()
#Compute the covariance matrix
cov_mat = np.cov(df_c04_93[['Batch Size', 'Labor Cost']].values, rowvar
= False)
display(cov_mat)
#Compute the correlation matrix
cor_mat = np.corrcoef(df_c04_93[['Batch Size', 'Labor Cost']].values, ro
wvar = False)
display(cor_mat)
```

-2



-3

```
array([[ 51.72413793, 170.68965517],
       [170.68965517, 950.60344828]])
```

-3

```
array([[1.        , 0.76976982],
       [0.76976982, 1.        ]])
```

-3

The covariance matrix is [1.72 170.69170.69 950.60] between the number of units per batch and label costs is 170.69 which means the two variables are positively related. The r of the data is 0.77 which mean the linear relationship of the two variables are very strong.

All round to the two decimal places.

Chapter 6

6.7

a.

1 − 0.42 = 0.58. The propability that Adams wins is 0.58.

b.

0.09 + 0.22 = 0.31. The probability that either Brown or Dalton wins is 0.31.

c.

0.42 + 0.09 + 0.27 = 0.78 = 1 − 0.22. The probability that Adams, Brown, or Collins wins is 0.78, which is equal to the sum of the probability that the three candidates win and 1 minus the probability Dalton wins.

6.11

a.

S = {Use a credit card, Use a debit card, Pay with cash}

b.

P(Use a Credit Card) = .60P(Pay with Cash) = .30P(Use a Debit Card) = 1 − 30% − 60% = .10

c.

Classical Approach