Week_1_Simple_Regression_GradedAssignment

August 12, 2017

1 Table of Contents

2 Notes

The notebook and other code can be found on my github repo.

3 Week 1 Graded Assignment

3.0.1 Questions

This exercise considers an example of data that do not satisfy all the standard assumptions of simple regression. In the considered case, assumption A6 that the coefficients and are the same for all observations is violated. The dataset contains survey outcomes of a travel agency that

wishes to improve recommendation strategies for its clients. The dataset contains 26 observations on age and average daily expenditures during holidays.

1. Use all data to estimate the coefficients a and b in a simple regression model, where **expenditures** is the dependent variable and **age** is the explanatory factor. Also compute the standard error and the t-value of b.

2. Make the scatter diagram of expenditures against age and add the regression line $y = a + bx$ of part (1) in this diagram. What conclusion do you draw from this diagram?
3. It seems there are two sets of observations in the scatter diagram, one for clients **aged 40 or higher** and another for clients **aged below 40**. Divide the sample into these two clusters, and for each cluster estimate the coefficients a and b and determine the standard error and t-value of b.
4. Discuss and explain the main differences between the outcomes in parts (1) and (3). Describe in words what you have learned from these results.

TestExer1-holiday expenditure
Simulated data set on holiday expenditures of 26 clients. 1. Age: age in years 2. Expenditures: average daily expenditures during holidays

3.0.2 Solutions

Libraries / Modules

```
In [1]: # getting the necessary libraries to read the data
        import pandas as pd

        # import the libraries for any potential mathematical operation
        import math

        # getting libraries for the scatter diagram
        import matplotlib.pyplot as plt
        import seaborn as sns

        # necessary function to display the chart here once it is generated

        %matplotlib inline
```

Read Data

```
In [2]: TestExer1 = pd.read_csv("./data/TestExer1_holiday_expenditure.txt", sep="\t") # the fi
        TestExer1
```

Out[2]:

	Observ.	Age	Expenditures
0	1	49	95
1	2	15	104
2	3	43	91
3	4	45	98
4	5	40	94

5	6	35	107
6	7	42	96
7	8	38	108
8	9	46	98
9	10	30	108
10	11	52	101
11	12	55	89
12	13	42	96
13	14	25	105
14	15	35	107
15	16	35	106
16	17	35	105
17	18	27	105
18	19	48	97
19	20	37	109
20	21	45	94
21	22	19	103
22	23	57	103
23	24	55	94
24	25	34	108
25	26	39	108

Answer 1 - Code and Calculations

```
In [3]: ## calculating the value of b which is needed to derive a

        Y = TestExer1.Expenditures              # the dependent variable
        X = TestExer1.Age                       # the independent variable
```

coefficient b

```
In [4]: b = ((X*Y).mean() - X.mean()*Y.mean()) / ((X**2).mean() - (X.mean())**2)
        print("Value of b, the coefficient, rounded to 3 digits is: ",round(b,3))
```

```
Value of b, the coefficient, rounded to 3 digits is:  -0.334
```

```
In [5]: X_bar = TestExer1.Age.mean()                # sample mean of age
        Y_bar = TestExer1.Expenditures.mean()   # sample mean of expenditures

        print("Mean Age rounded to 3 digits is: ", round(X_bar, 3))
        print("Mean Expenditure rounded to 3 digits is: ", round(Y_bar, 3))
```

```
Mean Age rounded to 3 digits is:  39.346
Mean Expenditure rounded to 3 digits is:  101.115
```

intercept a

```
In [6]: a = Y_bar - b*X_bar
        print("Value of a, the intercept, rounded to 3 digits is: ", round(a, 3))
```

```
Value of a, the intercept, rounded to 3 digits is:  114.241
```

error

```
In [7]: ## calculate error now that we know a and b

        TestExer1["error"] = TestExer1.Expenditures - a - b*TestExer1.Age
        TestExer1.head() # looking at the first 5 rows of the data set
```

Out[7]:

	Observ.	Age	Expenditures	error
0	1	49	95	-2.894899
1	2	15	104	-5.237167
2	3	43	91	-8.896476
3	4	45	98	-1.229284
4	5	40	94	-6.897264

```
In [8]: sum_sq_error = (TestExer1.error ** 2).sum() # calculating the sum of squares
```

standard error

```
In [9]: n = TestExer1.shape[0]    # number of entries

        s = math.sqrt(1/(n-2) * sum_sq_error) # standard deviation
        print("The standard error rounded to 3 digits is: ", round(s, 3))
```

```
The standard error rounded to 3 digits is:  5.073
```

t-value From the lecture 1.4 and the correspoding slides

$$t_b = \frac{b - \beta}{s_b}$$

where

$$s_b{}^2 = \frac{s^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

and

$$\beta = b - \sum_{i=1}^{n} c_i e_i$$

where

$$c_i = \frac{x_i - \bar{x}}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

```
In [10]: ## calclating ci in the dataset

         TestExer1["c"] = (TestExer1.Age - X_bar) / ((TestExer1.Age - X_bar)**2).sum()
         TestExer1.head(6) # showing the first few rows of the enhanced dataset
```

Out[10]:

	Observ.	Age	Expenditures	error	c
0	1	49	95	-2.894899	0.003411
1	2	15	104	-5.237167	-0.008603
2	3	43	91	-8.896476	0.001291
3	4	45	98	-1.229284	0.001998
4	5	40	94	-6.897264	0.000231
5	6	35	107	4.434755	-0.001536

```
In [11]: beta = b - (TestExer1.c * TestExer1.error).sum()
         print("The value of beta rounded to 3 digits is: ", round(beta, 3))
```

```
The value of beta rounded to 3 digits is:  -0.334
```

```
In [12]: ## calculating sb^2 before arriving at t
         s_b_sq = s**2 / ((X - X_bar)**2).sum()
         s_b_sq
```

```
Out[12]: 0.009095281025772856
```

```
In [13]: t_b = (b - beta)/s_b_sq
         print("The t value of b is: ", t_b) # it is too low to round
```

```
The t value of b is:  2.31925076400655e-13
```

Answer 1 Summary

```
In [14]: print("Quick Summary of Answer 1 results\n")
         print("Value of a, the intercept, rounded to 3 digits is: ", round(a, 3))
         print("Value of b, the coefficient, rounded to 3 digits is: ",round(b,3))
         print("The standard error rounded to 3 digits is: ", round(s, 3))
         print("The value of beta rounded to 3 digits is: ", round(beta, 3))
         print("The t value of b is: ", t_b)
```

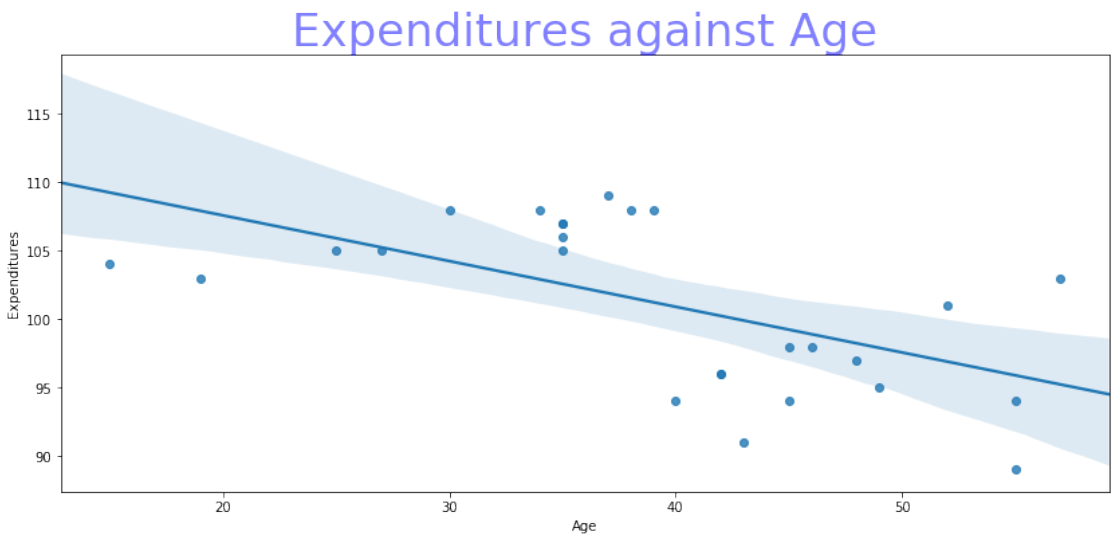
```
Quick Summary of Answer 1 results

Value of a, the intercept, rounded to 3 digits is:  114.241
Value of b, the coefficient, rounded to 3 digits is:  -0.334
The standard error rounded to 3 digits is:  5.073
The value of beta rounded to 3 digits is:  -0.334
The t value of b is:  2.31925076400655e-13
```

Answer 2

```
In [15]: plot = sns.regplot(data=TestExer1, x= "Age", y= "Expenditures")
         plot.figure.set_size_inches(14,6)
         plot.axes.set_title('Expenditures against Age', fontsize=34,color="b",alpha=0.5)

Out[15]: <matplotlib.text.Text at 0x1a981e48fd0>
```



We can clearly observe that there is a decreasing trend between the expeditures and age. As Age inceases, the expense goes down.

Answer 3 Part 1 - Age less than 40

Splitting the dataset for under 40 years of age

```
In [16]: msk = TestExer1.Age < 40
         young = TestExer1[msk].copy()
         young
```

```
Out[16]:     Observ.  Age  Expenditures     error         c
         1         2   15           104 -5.237167 -0.008603
         5         6   35           107  4.434755 -0.001536
         7         8   38           108  6.435544 -0.000476
         9        10   30           108  3.766775 -0.003303
         13       14   25           105 -0.901206 -0.005070
         14       15   35           107  4.434755 -0.001536
         15       16   35           106  3.434755 -0.001536
         16       17   35           105  2.434755 -0.001536
         17       18   27           105 -0.234013 -0.004363
         19       20   37           109  7.101948 -0.000829
         21       22   19           103 -4.902782 -0.007190
```

```
24       25   34           108  5.101159 -0.001889
25       26   39           108  6.769140 -0.000122
```

The code will follow a similar pattern as was done for Answer 1. I am not breaking the code down in individual blocks at this stage, however the results will be printed in a summary as was previously done with the summary of Answer 1.

Code and Summary

```
In [17]: ## calculating the value of b which is needed to derive a

         Y = young.Expenditures                # the dependent variable
         X = young.Age                         # the independent variable

         b = ((X*Y).mean() - X.mean()*Y.mean()) / ((X**2).mean() - (X.mean())**2)

         X_bar = X.mean()                 # sample mean of age
         Y_bar = Y.mean()

         a = Y_bar - b*X_bar
         print("Value of a, the intercept, rounded to 3 digits is: ", round(a, 3))
         print("Value of b, the coefficient, rounded to 3 digits is: ",round(b,3))

         ## calculate error now that we know a and b

         young["error"] = Y - a - b*X

         sum_sq_error = (young.error ** 2).sum() # calculating the sum of squares

         n = young.shape[0]    # number of entries

         s = math.sqrt(1/(n-2) * sum_sq_error) # standard deviation
         print("\nThe standard error rounded to 3 digits is: ", round(s, 3))

         ## calculating sb^2 before arriving at t
         s_b_sq = s**2 / ((X - X_bar)**2).sum()
         s_b_sq

         t_b = (b - beta)/s_b_sq
         print("The t value of b is: ", t_b) # it is too low to round

         # sample data set with errors and c
         print("\n\n Sample data for the final dataset for Age < 40 with error and c")
         young.head()
```

```
Value of a, the intercept, rounded to 3 digits is:  100.232
Value of b, the coefficient, rounded to 3 digits is:  0.198
```

```
The standard error rounded to 3 digits is:  1.153
The t value of b is:  269.84348995729385
```

```
 Sample data for the final dataset for Age < 40 with error and c
```

Out[17]:

	Observ.	Age	Expenditures	error	c
1	2	15	104	0.798154	-0.008603
5	6	35	107	-0.161272	-0.001536
7	8	38	108	0.244814	-0.000476
9	10	30	108	1.828584	-0.003303
13	14	25	105	-0.181559	-0.005070

Answer 3 Part 2 - Age equal or greater than 40

Splitting the dataset for 40 years of age or more

In [18]:
```
msk = TestExer1.Age >= 40
old = TestExer1[msk].copy()
old
```

Out[18]:

	Observ.	Age	Expenditures	error	c
0	1	49	95	-2.894899	0.003411
2	3	43	91	-8.896476	0.001291
3	4	45	98	-1.229284	0.001998
4	5	40	94	-6.897264	0.000231
6	7	42	96	-4.230072	0.000938
8	9	46	98	-0.895688	0.002351
10	11	52	101	4.105889	0.004472
11	12	55	89	-6.893323	0.005532
12	13	42	96	-4.230072	0.000938
18	19	48	97	-1.228495	0.003058
20	21	45	94	-5.229284	0.001998
22	23	57	103	7.773870	0.006238
23	24	55	94	-1.893323	0.005532

The code will follow a similar pattern as was done for Answer 1 and Answer 3, part 1. I am not breaking the code down in individual blocks at this stage, however the results will be printed in a summary as was previously done with the summary of Answer 1 and Answer 3, part 1.

In [19]:
```
## calculating the value of b which is needed to derive a

Y = old.Expenditures                # the dependent variable
X = old.Age                         # the independent variable

b = ((X*Y).mean() - X.mean()*Y.mean()) / ((X**2).mean() - (X.mean())**2)
```

```
X_bar = X.mean()                # sample mean of age
Y_bar = Y.mean()

a = Y_bar - b*X_bar
print("Value of a, the intercept, rounded to 3 digits is: ", round(a, 3))
print("Value of b, the coefficient, rounded to 3 digits is: ",round(b,3))

## calculate error now that we know a and b

old["error"] = Y - a - b*X

sum_sq_error = (old.error ** 2).sum() # calculating the sum of squares

n = old.shape[0]   # number of entries

s = math.sqrt(1/(n-2) * sum_sq_error) # standard deviation
print("\nThe standard error rounded to 3 digits is: ", round(s, 3))

## calculating sb^2 before arriving at t
s_b_sq = s**2 / ((X - X_bar)**2).sum()
s_b_sq

t_b = (b - beta)/s_b_sq
print("The t value of b is: ", t_b) # it is too low to round

# sample data set with errors and c
print("\n\n Sample data for the final dataset for Age > =  40 with error and c")
old.head()
```

```
Value of a, the intercept, rounded to 3 digits is:  88.872
Value of b, the coefficient, rounded to 3 digits is:  0.146

The standard error rounded to 3 digits is:  3.833
The t value of b is:  12.321853794664525


 Sample data for the final dataset for Age > =  40 with error and c
```

Out[19]:

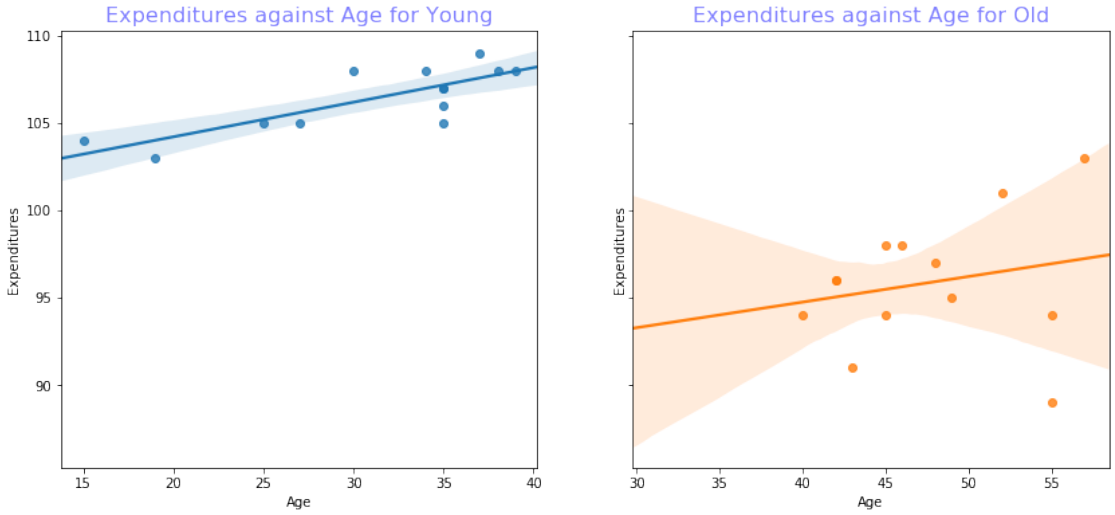
	Observ.	Age	Expenditures	error	c
0	1	49	95	-1.048960	0.003411
2	3	43	91	-4.170135	0.001291
3	4	45	98	2.536924	0.001998
4	5	40	94	-0.730722	0.000231
6	7	42	96	0.976336	0.000938

Answer 4 Let us plot the two data subsets and see if there are any visuals cues that confirm our analysis

```
In [20]: fig, (ax1, ax2) = plt.subplots(ncols=2, sharey=True)
         sns.regplot(x = young.Age, y = young.Expenditures, ax = ax1)
         ax1.figure.set_size_inches(14,6)
         ax1.axes.set_title('Expenditures against Age for Young', fontsize=16,color="b",alpha=(

         sns.regplot(x = old.Age, y = old.Expenditures, ax = ax2)
         ax2.figure.set_size_inches(14,6)
         ax2.axes.set_title('Expenditures against Age for Old', fontsize=16,color="b",alpha=0.5

Out[20]: <matplotlib.text.Text at 0x1a98246acf8>
```



Key Observations

1. Unlike the chart in Answer 1 the individual trends are quite the opposite. In Answer 1, the overal trend was decreasing. However for both the clusters, we can see an increasing trend with age. The reason for overall trend to be negative was because the expenditure for people under 40 is much more than the second cluster and hence with increasing age, the overall spend tends to come down.
2. For people under 40, the expense incrase with age is steeper than the other cluster. They spending habits are more senstive to age. The second cluster is relatively stable as can be seen with the lower slope of the regression line.