HW05_Lastname_Firstname

January 7, 2022

1 Homework 5: Hypothesis Testing and Simple Linear Regression

---

Name: Cal Brynestad

This assignment is due on Canvas by **6:00PM on Friday November 19**. Your solutions to theoretical questions should be done in Markdown directly below the associated question. Your solutions to computational questions should include any specified Python code and results as well as written commentary on your conclusions. Remember that you are encouraged to discuss the problems with your classmates, but **you must write all code and solutions on your own**.

NOTES:

- Any relevant data sets should be available in the Homework 05 assignment write-up on Canvas. To make life easier on the grader if they need to run your code, do not change the relative path names here. Instead, move the files around on your computer.
- If you're not familiar with typesetting math directly into Markdown then by all means, do your work on paper first and then typeset it later. **All** of your written commentary, justifications and mathematical work should be in Markdown.
- Because you can technically evaluate notebook cells is a non-linear order, it's a good idea to do Kernel → Restart & Run All as a check before submitting your solutions. That way if we need to run your code you will know that it will work as expected.
- It is **bad form** to make your reader interpret numerical output from your code. If a question asks you to compute some value from the data you should show your code output **AND** write a summary of the results in Markdown directly below your code.
- This probably goes without saying, but… For any question that asks you to calculate something, you **must show all work and justify your answers to receive credit**. Sparse or nonexistent work will receive sparse or nonexistent credit.

---

[1]:

```
import numpy as np
import math
import scipy.stats as stats
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline
```

1.0.1 [50 points] Problem 1 - Naps vs Coffee for Memory?

It is estimated that about 75% of adults in the United States drink coffee. Often, coffee is used to replace the need for sleep. It works alright, or so we think. Let's find out, in this exciting homework problem!

One recent study investigated the effects of drinking coffee, taking a nap, and having a "coffee-nap" - the practice of drinking some coffee *and then* having a short nap. The study broke participants up into three groups of 10 participants each, where the groups would have a nap, or have a coffee, or have a coffee-nap, then perform a task where their reaction time was measured. In previous experiments the mean reaction time measurement was found to be normally distributed. The reaction time means (milliseconds, ms) and standard deviations for the three groups of participants are given in the table below.

Group	Sample Size	Mean	Standard Deviation
Coffee+Nap	10	351.6	39.9
Coffee	10	391.2	37.6
Nap	10	382.8	40.5

Part A: Compute a 90% t-confidence interval for the mean reaction time measurement for participants in each of these three groups. (You should find three separate confidence intervals.) For your computations in Python make sure that you do not use canned functions. Use the CI formulas given in lecture and show your steps. Print each CI to the screen.

1. Can you make any conclusions regarding whether coffee, naps or both (coffee-naps) are better for faster reaction times?
2. Why did we use a t-distribution?

Solution: Our confidence intervals are constructed as: $\bar{x} \pm t_{\alpha/2,n-1} \cdot \frac{s}{\sqrt{n}}$ $\bar{x}$, s and $\sqrt{n}$ is given for each of the three groups.

[2]:

```python
# Code here
t = stats.t.ppf(1-(.1/2), 9)
n = 10

s1 = 39.9
x1 = 351.6
CIH1 = x1 + (t*(s1/np.sqrt(n)))
CIL1 = x1 - (t*(s1/np.sqrt(n)))

s2 = 37.6
x2 = 391.2
CIH2 = x2 + (t*(s2/np.sqrt(n)))
CIL2 = x2 - (t*(s2/np.sqrt(n)))

s3 = 40.5
x3 = 382.8
```

```
CIH3 = x3 + (t*(s3/np.sqrt(n)))
CIL3 = x3 - (t*(s3/np.sqrt(n)))


print("t-CI for the mean reaction time for participants in the Coffee + Nap is:
 ↪", CIL1, CIH1)
print("t-CI for the mean reaction time for participants in the Coffee group is:
 ↪", CIL2, CIH2)
print("t-CI for the mean reaction time for participants in the Nap group is:",␣
 ↪CIL3, CIH3)
```

```
t-CI for the mean reaction time for participants in the Coffee + Nap is:
328.4707198187697 374.72928018123037
t-CI for the mean reaction time for participants in the Coffee group is:
369.4039865961338 412.9960134038662
t-CI for the mean reaction time for participants in the Nap group is:
359.3229110942399 406.27708890576014
```

Each of the three confidence intervals contain common values so we cannot make any conclusions regarding whether coffee, naps or both (coffee-naps) are better for faster reaction times. We used a t-test because our sample sizes were less than 30 and the population variance was unknown.

Part B: Use an appropriate hypothesis test to determine if there sufficient evidence, at the $\alpha = 0.1$ significance level, to conclude that taking a nap promotes faster reaction time than drinking coffee. Be sure to clearly explain the test that you're doing and state all hypotheses. Do all computations in Python, and report results.

Solution: μ_1 is the mean reaction time after coffee μ_2 is the mean reaction time after a nap $H_0 : \mu_1 - \mu_2 = 0$ $H_1 : \mu_1 - \mu_2 < 0$ We are testing at the $\alpha = 0.1$ significance level. We will calculate a t-value that will be used to calculate a p-value which will be compared to our significance level to see if there is a statistically significant difference between the two means. We will compute our t-values using this equation: $t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{{s_1}^2}{n_1} + \frac{{s_2}^2}{n_2}}}$

[3]:

```
# Code here
x1 = 391.2
x2 = 382.8
s1 = 37.6
s2 = 40.5
n = 10

xDiff = x1 - x2

se1 = (s1**2)/n
se2 = (s2**2)/n

se = np.sqrt(se1 + se2)

t = xDiff/se
```

```
p = 1 - stats.t.cdf(t, 18)

print("P-val:", p)
```

```
P-val: 0.31827183338468035
```

So our t-score produced a p-value = .318 and we see that our p-value > α, or .318 > .1. Thus, we cannot reject the null hypothesis and conclude that on average the difference in mean reaction times between those who drink coffee and those who take naps is 0.

Part C: Perform two hypothesis tests both at the $\alpha = 0.1$ significance level, to first determine if there is sufficient evidence to conclude that taking a coffee-nap promotes faster reaction time than only drinking coffee. The second hypothesis test should determine if there is sufficient evidence to conclude that taking a coffee-nap promotes faster reaction time than only having a nap. Be sure to clearly explain the test that you're doing and state all hypotheses. Do all computations in Python, and report results.

Solution: μ_1 is the mean reaction time after coffee μ_2 is the mean reaction time after a nap μ_3 is the mean reaction time after a coffee nap Hypotheses to determine if there is sufficient evidence to conclude that taking a coffee-nap promotes faster reaction time than only drinking coffee: $H_0 : \mu_3 - \mu_1 = 0$ $H_1 : \mu_3 - \mu_1 < 0$ Hypotheses to determine if there is sufficient evidence to conclude that taking a coffee-nap promotes faster reaction time than only having a nap: $H_0 : \mu_3 - \mu_2 = 0$ $H_1 : \mu_3 - \mu_2 < 0$ For each test, we are testing at the $\alpha = 0.1$ significance level. We will calculate a t-value that will be used to calculate a p-value which will be compared to our significance level to see if there is a statistically significant difference between the two means, for each test. We will compute our t-values using this equation: $t = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{\frac{{s_1}^2}{n_1} + \frac{{s_2}^2}{n_2}}}$

[4]:

```
# Code here
xCN = 351.6
sCN = 39.9
xC = 391.2
sC = 37.6
xN = 382.8
sN = 40.5
n = 10

xDiff = xC - xCN

se1 = (sCN**2)/n
se2 = (sC**2)/n

se = np.sqrt(se1 + se2)

t = xDiff/se

p = 1 - stats.t.cdf(t, 18)
```

```
print("P-val for difference between Coffee and Coffee + Nap:", p)

############################

xDiff = xN - xCN

se1 = (sCN**2)/n
se2 = (sN**2)/n

se = np.sqrt(se1 + se2)

t = xDiff/se

p = 1 - stats.t.cdf(t, 18)

print("P-val for difference between Nap and Coffee + Nap:", p)
```

```
P-val for difference between Coffee and Coffee + Nap: 0.017362883319716915
P-val for difference between Nap and Coffee + Nap: 0.049878269947941645
```

Our t-scores produced p-values of .017 and .049, for the difference of means in reaction time between those who drink coffee and those who take a coffee nap, and between those who take a nap and those who take a coffee nap, respectively. We see that for both tests our p-value < α, or .017 < .1 and .049 < .1. Thus, we reject the null hypotheses for both tests in favor of the alternate hypotheses and conclude that on average the mean reaction time of those who take coffee naps is lower than those who drink coffee and the mean reaction time of those who take coffee naps is lower than those who take naps. In other words, those who take coffee naps have faster reaction times than those who drink coffee and those who take naps.

1.0.2 [50 points] Problem 2 - Simple Linear Regression on Children's Lung Function

Part A: Load up the data in `LungFunction.csv` into a Pandas DataFrame. The variables to study in this problem are given by the data in the first two columns. The response variable is y = forced exhalation volume (FEV), a measure of how much air somebody can forcibly exhale from their lungs, and the feature variable is x = age in years.

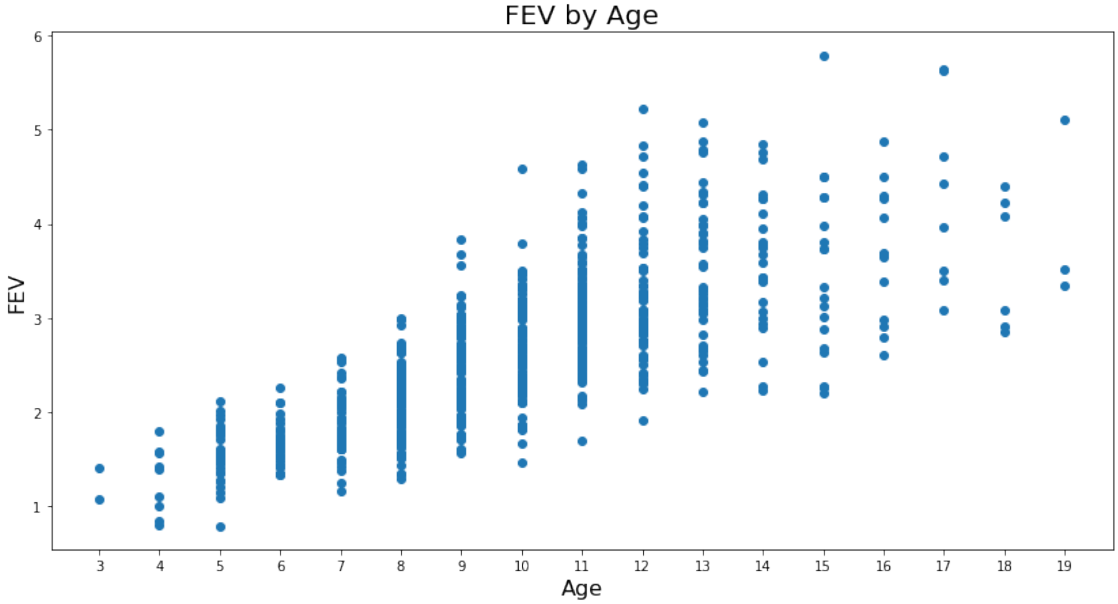
Make a **scatter plot** of all of the data points, showing how lung function (measured by FEV) in children, changes with age. Create x-axis tick marks by age in yearly increments displaying ticks for each age from 3 to 19. Make sure to label your x-axis, y-axis, and title of your plot.

[5]:

```
# Code here
file_path = 'LungFunction.csv'

df = pd.read_csv(file_path)

fig, ax = plt.subplots(figsize=(14,7))
```

```
scatter = ax.scatter(x=df['age'],y=df['FEV'])
labels = np.linspace(3,19,17)
ax.set_xticks(labels)

ax.set_title("FEV by Age", fontsize=20)
ax.set_xlabel("Age", fontsize=16)
ax.set_ylabel("FEV", fontsize=16)
```

```
[5]: Text(0, 0.5, 'FEV')
```



Part B: Use the stats.linregress package to fit a linear model to your data from **Part A**.

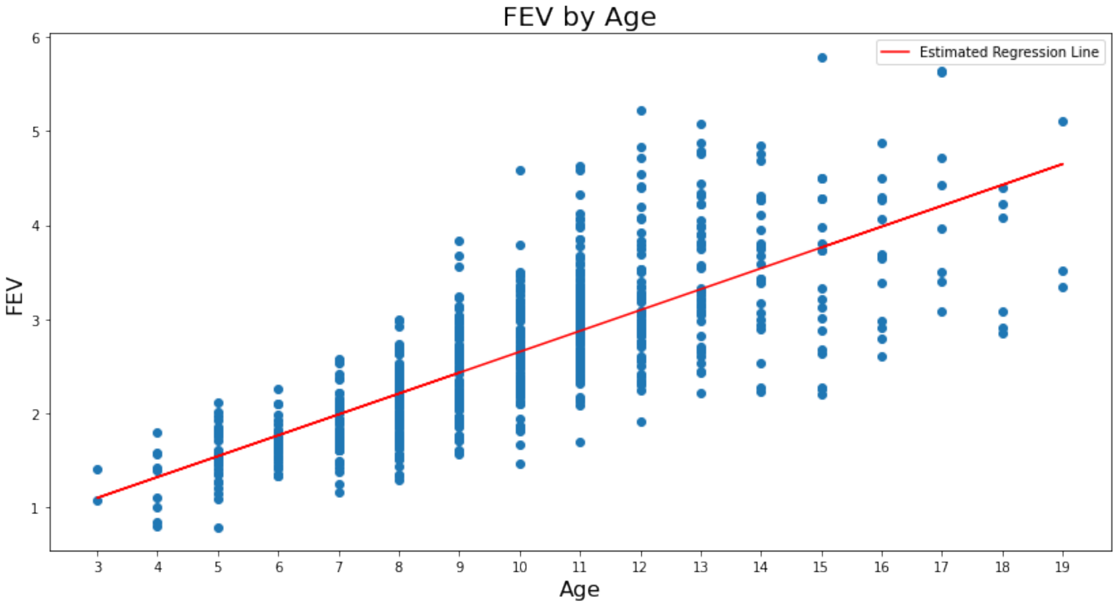
- Print the parameters of the regression line in the forrm $Y = \hat{\alpha} + \hat{\beta}X$.
- Then make a scatter plot of the FEV values as a function of age, and overlay the estimated regression line. Label your axes and provide a legend.

```
[6]: # Code here
bhat, ahat, rval, pval, stderr = stats.linregress(df["age"], df["FEV"])
print("Regression Line: Y = {:.5f} + {:.5f}x".format(ahat, bhat))

fig, ax = plt.subplots(figsize=(14,7))

x = df['age'].tolist()


scatter = ax.scatter(x=df['age'],y=df['FEV'])
```

```
line = ax.plot(df['age'],ahat + bhat*df['age'], label='Estimated Regression␣
 ↪Line', color='red')

labels = np.linspace(3,19,17)
ax.set_xticks(labels)

ax.set_title("FEV by Age", fontsize=20)
ax.set_xlabel("Age", fontsize=16)
ax.set_ylabel("FEV", fontsize=16)
ax.legend()
```

```
Regression Line: Y = 0.43165 + 0.22204x
```

```
[6]: <matplotlib.legend.Legend at 0x7f82504790d0>
```



Part C: Give a physical interpretation of the coefficient $\hat{\beta}$, estimated from your model. Include addressing whether the relationship between age and lung function is positive or negative. Fully justify your responses.

Solution: $\hat{\beta} = 0.22204$ This $\hat{\beta}$, a positive slope for the linear regression line, suggests a positive relationship between age and FEV, meaning the older one is, the more air one can forcibly exhale from their lungs. This claim passes the sanity check and can be seen in the scatter plot above, where as age increases, it appears FEV measurements do as well.

Part D: Compute the R^2 value for your regression line in **Part B**. Use the formula given in lecture involving SSE, SSR, and SST. Once you have computed R^2 using the formula from lecture. Verify that stats.linregress provides the same value. Note: stats.linregress returns "rval" and $R^2 = (rval)^2$

$$\hat{y} = \hat{a} + \hat{b} \cdot x$$

$$\text{SSE} = \sum (y - \hat{y})^2$$

$$\text{SSR} = \sum (\bar{y} - \hat{y})^2$$

$$\text{SST} = \sum (\bar{y} - y)^2$$

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}}$$

[7]:

```
# Code here
yhat = []
for i in x:
    j = bhat*i
    k = j + ahat
    yhat.append(k)

SSE  = np.sum((df['FEV']-yhat)**2)
SST  = np.sum((np.mean(df['FEV'])-df['FEV'])**2)
R2 = 1 - SSE/SST
print("R squared computed from formula is", R2)






Rsq = rval**2
print("R squared provided by stats.linregress is", Rsq) #rval computed in cell␣
 ↪above
```

```
R squared computed from formula is 0.5722302035360854
R squared provided by stats.linregress is 0.5722302035360854
```

Part E: Give an interpretation of the R^2 value that you found. Mention what it indicates about the fit of your model.

Solution: R^2, the coefficient of determination, is a goodness of fit measure. It's a statistical measure that represents the proportion of the variance for a dependent variable that is explained by an independent variable in a regression model. An $R^2 = .572$ indicates that 57% of the variance for FEV is explained by age in the regression model. This is a fairly high R^2 value as $0 \leq R^2 \leq 1$, and indicates a fairly high correlation between age and FEV.

Part F: Use the data to compute a 95% confidence interval for the slope. Note that there are built in packages that compute the CI, however you should be performing all calculations yourself in Python. You may use the stderr value that was returned by stats.linregress above. Show all computation involved in finding the CI and print your CI for the slope to the screen.

[13]:

```
# Code here
k = len(df['age'])-2
```

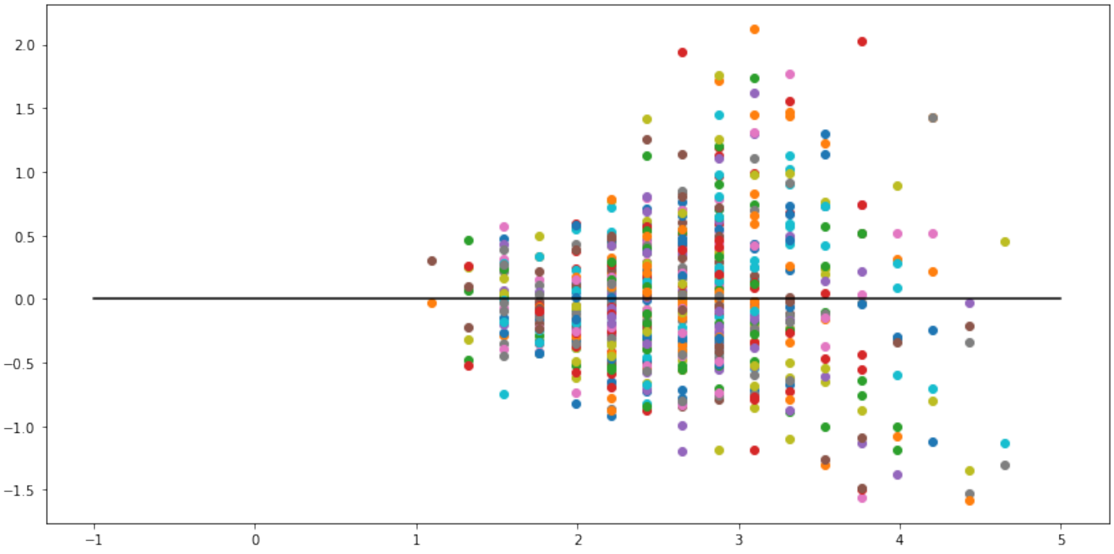
```
L_CI = (bhat - stats.t.ppf(0.975, k) * stderr)
U_CI = (bhat + stats.t.ppf(0.975, k) * stderr)

print("95% CI = [{:.3f}, {:.3f}], CI width = {:.3f}".format(L_CI, U_CI, U_CI -␣
 ↪L_CI))
```

```
95% CI = [0.207, 0.237], CI width = 0.030
```

Part G: Another way to study how well a simple linear regression model fits data is to plot the residuals. Make a plot of the residuals and provide a brief explanation of what you find.

[9]:
```
# Code here
fig, ax = plt.subplots(figsize=(14,7))



y = df['FEV'].tolist()
x = df['age'].tolist()

j = 0
for i in y:
    res = i - (ahat + bhat*x[j])
    pre = ahat + bhat*x[j]
    ax.scatter(x=pre, y=res)
    j+=1

x = np.linspace(-1,5,6)
y = np.linspace(0,0,6)
ax.plot(x,y,color="black")
```

[9]:
```
[<matplotlib.lines.Line2D at 0x7f8250bbacd0>]
```



Solution: After plotting the residuals, we see that the residuals are unbiased, meaning they have an average value of 0 in any thin vertical strip found in the plot, and we see that the residuals are fairly homoscedastic, meaning the spread of the residuals is about the same in any thin vertical strip. We can notice that this spread is less variable at the lowest and highest ages in our plot. Given the residuals are fairly unbiased and homoscedastic, the linear regression model fits the data fairly well.

[ ]: