CS 249 | HW #2 | Jason Chapman

① a) One-tailed T-test

Ho: Mean Salary of 1^{st} choice ≤ Current Salary

⇒ Test in Python

$p = 0.00444$, $\alpha = 0.05$

- $p < \alpha$, so, we reject the null hypothesis & can say that the friend can expect an increase in her salary on average if she chooses #1.

b) One-way ANOVA Test

Ho: Mean Salary of 1^{st} choice = Mean Salary of 2^{nd} choice = Mean Salary of 3^{rd} choice

⇒ Test in python

$p = 0.0403$, $\alpha = 0.05$

- $p < \alpha$, so we reject the null hypothesis & can say that there is a difference between the average salary of the three choices.

c) Two-sided T-test

Ho: Mean Salary of 1^{st} choice = Mean salary of 3^{rd} choice

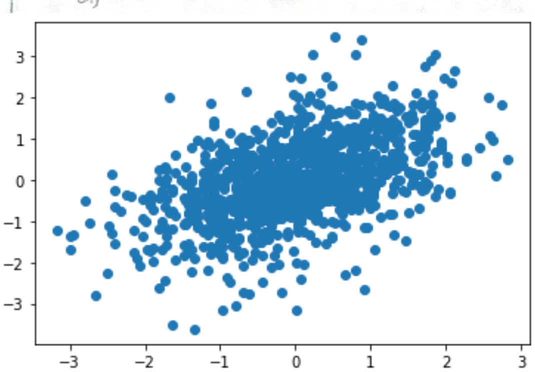
⇒ Test in Python

$p = 0.0104$, $\alpha = 0.05$

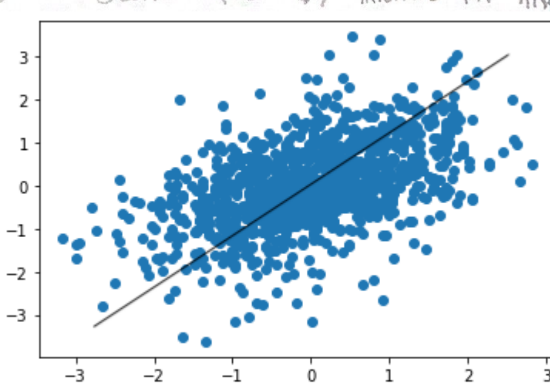
- $p < \alpha$, so we reject the null hypothesis & say there is a difference between the average salary of #1 & #3

- We should at least acknowledge the fact that the multiple hypothesis tests ran for comparing multiple groups can lead to a high probability of falsley rejecting the null Hypothesis.

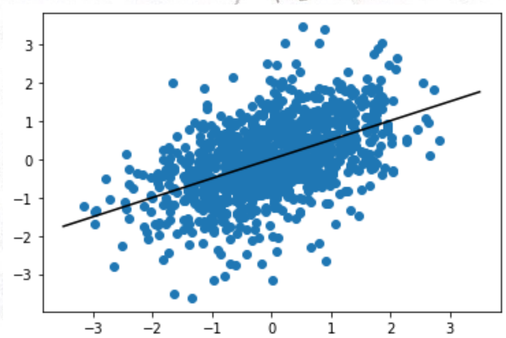
② a) Scatter Plot



b) Scatter Plot w/ intuitive fit line



c.) Scatter Plot w/ Regression Line



- My estimate was fairly close to the true model fit. With a greater slope

d.) $\hat{\beta}_0 = -3.2606e^{-15}$

$\hat{\beta}_1 = 0.50116268$

$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$

$\hat{Y} = (-3.2606e^{-15}) + (0.50116268)(12) = 6.1395216$

$\hat{Y} = (-3.2606e^{-15}) + (0.50116268)(-12) = -6.1395216$

- If a father is 12" taller than the average, the expected height of the son is 6.14" above the average.
  If a father is 12" shorter than the average, the expected height of the son is 6.14" below the average.

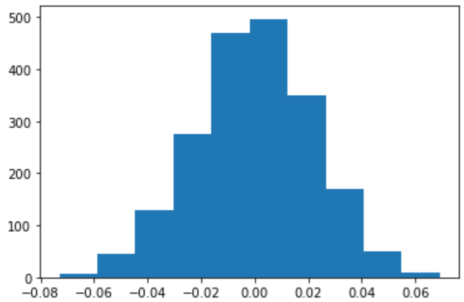
e.) When x & y are standardized, as they are in this problem, it can lead to an interpretation that the data "regresses to the mean" because a 1 unit increase in X will always lead to a <1 unit increase in y. And over time this will be a "regression to the mean". However, this view is mistaken because it ignores the error in the regression predicting y from x. For any data point X_i, the point prediction for y_i will be regressed toward the mean, but the actual y_i that is observed will not be exactly where it's predicted. Some points end up falling closer to the mean & some fall further.

③ a)

```
                                 OLS Regression Results
=======================================================================================
Dep. Variable:                      y   R-squared (uncentered):                   0.001
Model:                            OLS   Adj. R-squared (uncentered):              0.000
Method:                 Least Squares   F-statistic:                              1.060
Date:                Wed, 19 Oct 2022   Prob (F-statistic):                       0.303
Time:                        07:39:42   Log-Likelihood:                         -2816.5
No. Observations:                2000   AIC:                                      5635.
Df Residuals:                    1999   BIC:                                      5641.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.0226      0.022      1.029      0.303      -0.020       0.066
==============================================================================
Omnibus:                        1.953   Durbin-Watson:                   1.912
Prob(Omnibus):                  0.377   Jarque-Bera (JB):                2.008
Skew:                           0.063   Prob(JB):                        0.366
Kurtosis:                       2.910   Cond. No.                         1.00
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

b)



- As you can see, the distribution follows that of a normal distribution centered around a slope of 0. The "slopes" that result from regression are essentially the "errors" because the two distributions should be the same & have a slope of 0. But, because we are simulating random point there will inherently be difference between x & y. But, over 100 simulations, the results are as expected.

④ a) $\sum_{i=1}^{n} \hat{\varepsilon}_i = 0$: $Y_i = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j X_{ij} + \hat{\varepsilon}_i$

Residual sum of squares : $S = \sum_{i=1}^{n} \left(Y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j X_{ij} \right)^2$

Differentiate S wrt $\hat{\beta}_0$ & set to 0 (least squares principle)

$$\Rightarrow \sum_{i=1}^{n} (-2)\left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j X_{ij} \right) = 0$$

$$\Rightarrow \sum_{i=1}^{n} \underbrace{\left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j X_{ij} \right)}_{\hat{\varepsilon}_i} = 0$$

$$\Rightarrow \sum_{i=1}^{n} \hat{\varepsilon}_i = 0 \quad \checkmark$$

b.) $\hat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n} y_i$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\beta}_0 + \sum_{j=1}^{p}\hat{\beta}_j X_{ij} + \hat{\varepsilon}_i\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\hat{\beta}_0 + \frac{1}{n}\sum_{j=1}^{p}\hat{\beta}_j\sum_{i=1}^{n} X_{ij} + \frac{1}{n}\sum_{i=1}^{n}\hat{\varepsilon}_i$$

// we know from ⓐ $\sum_{i=1}^{n}\hat{\varepsilon}_i = 0$

centered feature $\Rightarrow$ $\frac{1}{n}\sum_{i=1}^{n} X_{ij} = 0$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{n}\hat{\beta}_0 + 0 + 0$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{\cancel{n}}\cdot\hat{\beta}_0\cdot\cancel{n}$$

$$\Rightarrow \hat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n} y_i \quad \checkmark$$

```
# CS 249 - HW #2
# Jason Chapman
from sklearn.linear_model import LinearRegression
from scipy import stats
import statsmodels.api as sm
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# 1
salary = 80;
opt1 = [143, 102, 119, 157, 146, 61, 119, 85, 87, 102]
opt2 = [77, 143, 108, 76, 92, 87, 145, 60, 86, 27]
opt3 = [19, 83, 87, 55, 115, 41, 71, 66, 101, 99]

# 1a
t_stat, p_val1a = stats.ttest_1samp(opt1, salary, alternative='greater')

# 1b
f_stat, p_val1b = stats.f_oneway(opt1, opt2, opt3)

# 1c
t_stat, p_val1c = stats.ttest_ind(opt1, opt3, equal_var=True)

# 2
df = pd.read_csv('Pearson.txt', sep="\t", header=0)
i = 0
x_raw = []
y_raw = []
for row in df.iterrows():
    x_raw.append(df.iat[i,0])
    y_raw.append(df.iat[i,1])
    i += 1

x = (x_raw-np.mean(x_raw))/np.std(x_raw)
y = (y_raw-np.mean(y_raw))/np.std(y_raw)
x = np.reshape(x, (-1,1))
y = np.reshape(y, (-1,1))
plt.figure(1)
plt.scatter(x,y)

# 2c
model = LinearRegression()
model.fit(x, y)
beta_0_hat = model.intercept_
beta_1_hat = model.coef_[0]
x_line = np.linspace(-3.5,3.5)
y_line = beta_0_hat+beta_1_hat*x_line
plt.figure(1)
plt.plot(x_line, y_line, 'k-')
plt.show()
```

```python
# 3
# 3a
X = np.random.normal(0, 1, 2000)
Y = np.random.normal(0, 1, 2000)
model = sm.OLS(X, Y).fit()
print(model.summary())
# 3b
iter = []
slopes = []
for i in range(2000):
    X = np.random.normal(0, 1, 2000)
    Y = np.random.normal(0, 1, 2000)
    model = sm.OLS(X, Y).fit()
    iter.append(i+1)
    slopes.append(model.params)

counts, bins = np.histogram(slopes)
plt.figure(2)
plt.hist(bins[:-1], bins, weights=counts)
```