**Unit 3 Assignment**

Laurence T. Burden

Purdue Global University

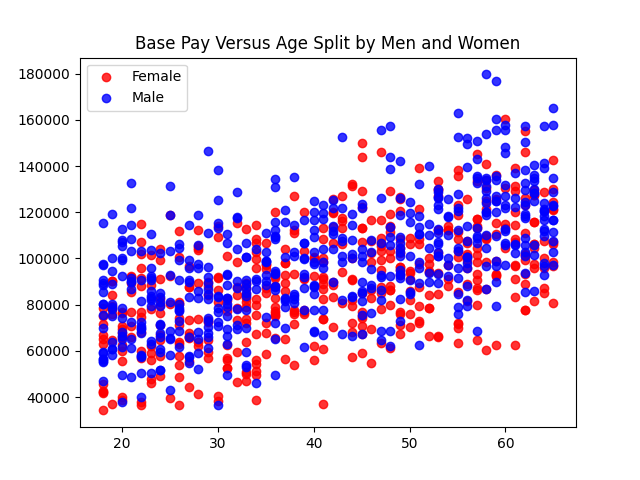
IN402: Modeling and Predictive Analysis

Steward Huang

November 26, 2023

The Python code examines data from 1,000 employees with 9 features (Glass Door, n.d.). The data is first read in from a CSV file. The data types, data frame shape, the first five, last five, and general information are all printed to the console next. Duplicates are then removed from the data frame using the duplicated() method.

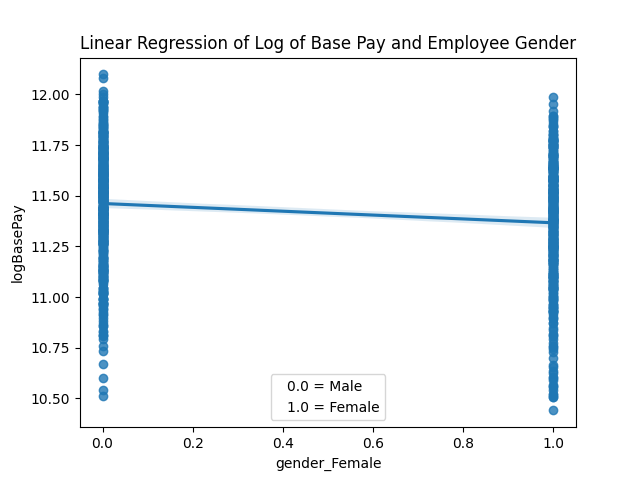
The null hypothesis is printed next. This is stated as “gender has no effect on base pay.” The data is visualized next to see the relationships. Male and female base pay are plotted using a scatter plot as shown below in figure 1.

**Figure 1**

Dummy variables are created for both the gender and ‘edu’ columns. This splits the categorical values of these columns into new columns with a 0 and 1 value to represent true and false. This allows for using the education and gender features within the regression methods because they are now numerical, rather than text/categorical. The natural log of the base pay is also calculated to help with the regression calculations by making the skewed data more normally distributed.

The initial linear regression of the data is completed using the gender as a true/false for being a female and the log of the base pay observations. The coefficient found for employees who are female is -0.0953. This means that the linear equation is y = 11.4618 + -0.0953x and a female employee can expect to make slightly less than a male employee based on this single variable regression model. This can be viewed by the slight decline from left to right in the regression line of Figure 2.

**Figure 2**

****

The next model is another linear regression but uses the least square method. This function strives to best fit the data by reducing the sum of squares for the residuals. The log base pay values are used and give the following coefficients: female = 3.4811, male = 3.5944, age = 0.0115, and seniority = 0.1090. This shows that age and seniority have a slight positive impact on base pay. As age and seniority go up, so does the base pay. There is not much difference in the male and female coefficients, but the female employees do show that they get paid slightly less with this model, too.

The R2 value of the least square method regression is 0.623 and the adjusted R2 value is 0.621. R2 measures the variation within the regression model but generally increases any time more predictor variables are added. The adjusted R2 value introduces a penalty for introducing more predictor variables and is therefore better for multivariable regression models (Muralidhar, 2023).

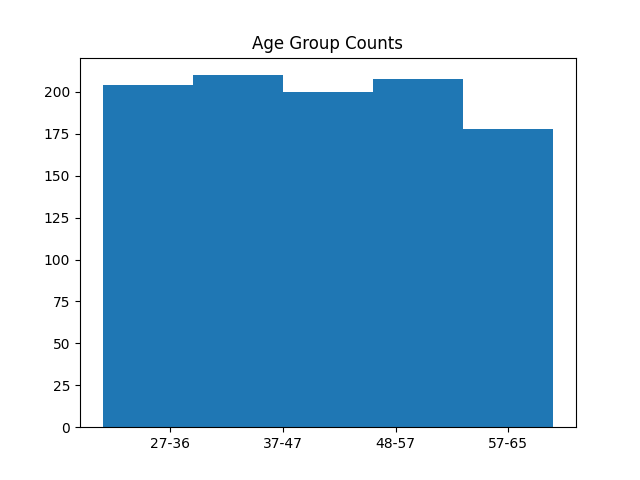
The data analyzed here clearly shows a small pay gap between male and female employees, and the null hypothesis should be rejected. Two models are created, a standard linear regression and a least square method regression model. These both point to the small pay gap. Other charts, the code, and the code output can be found in appendices A–C.

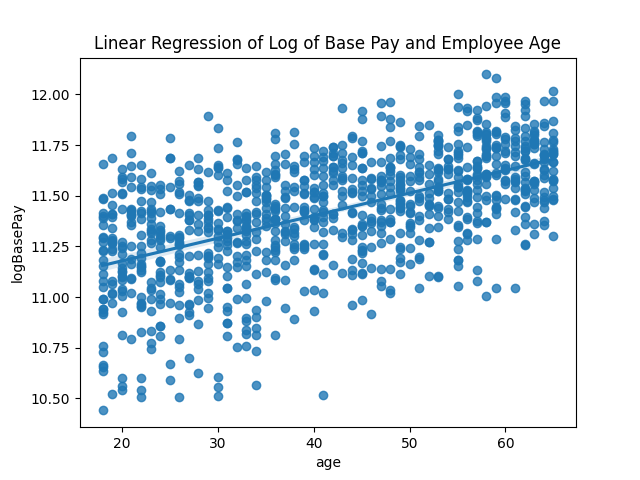
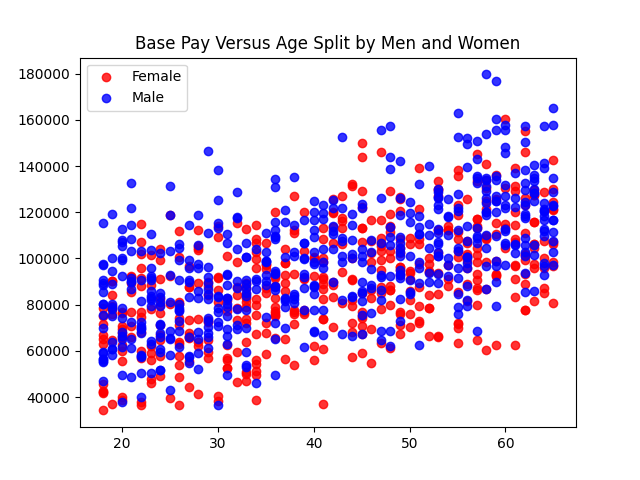
**References**

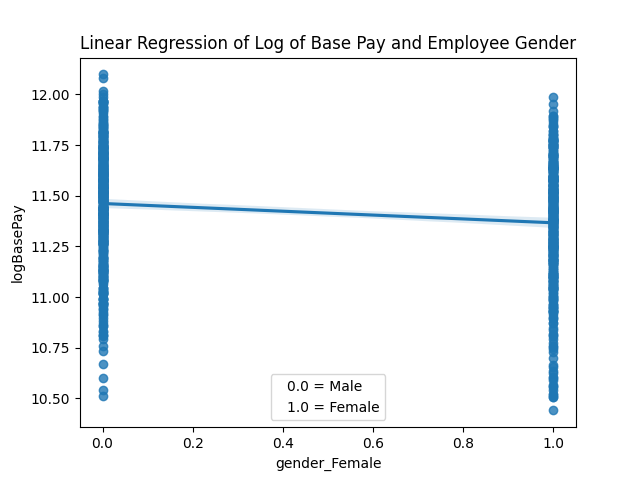
*Gender Pay Data*. Glass Door. (n.d.). https://glassdoor.app.box.com/v/gender-pay-data

Muralidhar, K. (2023, February 27). *Demystifying R-squared and adjusted R-squared*. Built In. https://builtin.com/data-science/adjusted-r-squared

**Appendix A**

****





**Appendix B**

**Code Output**

Unit 3 Assignment / Module 3 Part 1 Competency Assessment Output

11/26/2023 19:51:23

### Data Types ###

jobTitle object

gender object

age int64

perfEval int64

edu object

dept object

seniority int64

basePay int64

bonus int64

dtype: object

### Data Frame Shape ###

(1000, 9)

### Data Frame Head ###

jobTitle gender age perfEval edu dept \

0 Graphic Designer Female 18 5 College Operations

1 Software Engineer Male 21 5 College Management

2 Warehouse Associate Female 19 4 PhD Administration

3 Software Engineer Male 20 5 Masters Sales

4 Graphic Designer Male 26 5 Masters Engineering

seniority basePay bonus

0 2 42363 9938

1 5 108476 11128

2 5 90208 9268

3 4 108080 10154

4 5 99464 9319

### Data Frame Tail ###

jobTitle gender age perfEval edu dept \

995 Marketing Associate Female 61 1 High School Administration

996 Data Scientist Male 57 1 Masters Sales

997 Financial Analyst Male 48 1 High School Operations

998 Financial Analyst Male 65 2 High School Administration

999 Financial Analyst Male 60 1 PhD Sales

seniority basePay bonus

995 1 62644 3270

996 2 108977 3567

997 1 92347 2724

998 1 97376 2225

999 2 123108 2244

### Data Frame Info ###

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1000 entries, 0 to 999

Data columns (total 9 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 jobTitle 1000 non-null object

1 gender 1000 non-null object

2 age 1000 non-null int64

3 perfEval 1000 non-null int64

4 edu 1000 non-null object

5 dept 1000 non-null object

6 seniority 1000 non-null int64

7 basePay 1000 non-null int64

8 bonus 1000 non-null int64

dtypes: int64(5), object(4)

memory usage: 70.4+ KB

None

### Data Frame Counts ###

jobTitle 1000

gender 1000

age 1000

perfEval 1000

edu 1000

dept 1000

seniority 1000

basePay 1000

bonus 1000

dtype: int64

### Duplicates ###

Empty DataFrame

Columns: [jobTitle, gender, age, perfEval, edu, dept, seniority, basePay, bonus]

Index: []

age perfEval seniority basePay bonus

count 1000.000000 1000.000000 1000.000000 1000.000000 1000.000000

mean 41.393000 3.037000 2.971000 94472.653000 6467.161000

std 14.294856 1.423959 1.395029 25337.493272 2004.377365

min 18.000000 1.000000 1.000000 34208.000000 1703.000000

25% 29.000000 2.000000 2.000000 76850.250000 4849.500000

50% 41.000000 3.000000 3.000000 93327.500000 6507.000000

75% 54.250000 4.000000 4.000000 111558.000000 8026.000000

max 65.000000 5.000000 5.000000 179726.000000 11293.000000

Null Hypothesis: Gender has no effect on base pay

jobTitle age perfEval dept seniority basePay \

0 Graphic Designer 18 5 Operations 2 42363

1 Software Engineer 21 5 Management 5 108476

2 Warehouse Associate 19 4 Administration 5 90208

3 Software Engineer 20 5 Sales 4 108080

4 Graphic Designer 26 5 Engineering 5 99464

bonus gender\_Female gender\_Male edu\_College edu\_High School \

0 9938 True False True False

1 11128 False True True False

2 9268 True False False False

3 10154 False True False False

4 9319 False True False False

edu\_Masters edu\_PhD

0 False False

1 False False

2 False True

3 True False

4 True False

### Age Bins ###

AgeGroup

27-36 210

48-57 208

18-26 204

37-47 200

57-65 178

Name: count, dtype: int64

### Base Pay Log ###

0 10.654031

1 11.594284

2 11.409873

3 11.590627

4 11.507551

Name: logBasePay, dtype: float64

### Linear Regression of Gender and Log of Base Pay ###

Intercept: [11.46181801]

Coefficient: [[-0.09531562]]

R-Squared: 0.027466191971902365

### OLS of Original Base Pay Data ###

OLS Regression Results

==============================================================================

Dep. Variable: basePay R-squared: 0.630

Model: OLS Adj. R-squared: 0.629

Method: Least Squares F-statistic: 565.4

Date: Sun, 26 Nov 2023 Prob (F-statistic): 1.72e-214

Time: 19:51:59 Log-Likelihood: -11061.

No. Observations: 1000 AIC: 2.213e+04

Df Residuals: 996 BIC: 2.215e+04

Df Model: 3

Covariance Type: nonrobust

=========================================================================================

coef std err t P>|t| [0.025 0.975]

-----------------------------------------------------------------------------------------

Intercept 1.538e+04 1228.216 12.519 0.000 1.3e+04 1.78e+04

gender\_Female[T.True] 2631.8972 806.359 3.264 0.001 1049.539 4214.255

gender\_Male[T.True] 1.274e+04 763.694 16.688 0.000 1.12e+04 1.42e+04

age 1027.5395 34.183 30.060 0.000 960.460 1094.619

seniority 9610.1644 350.269 27.437 0.000 8922.815 1.03e+04

==============================================================================

Omnibus: 29.350 Durbin-Watson: 1.957

Prob(Omnibus): 0.000 Jarque-Bera (JB): 31.179

Skew: 0.418 Prob(JB): 1.70e-07

Kurtosis: 3.222 Cond. No. 1.07e+16

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 1.68e-26. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

### OLS of Natural Log of Base Pay Data ###

OLS Regression Results

==============================================================================

Dep. Variable: logBasePay R-squared: 0.623

Model: OLS Adj. R-squared: 0.621

Method: Least Squares F-statistic: 547.5

Date: Sun, 26 Nov 2023 Prob (F-statistic): 3.96e-210

Time: 19:51:59 Log-Likelihood: 316.53

No. Observations: 1000 AIC: -625.1

Df Residuals: 996 BIC: -605.4

Df Model: 3

Covariance Type: nonrobust

=========================================================================================

coef std err t P>|t| [0.025 0.975]

-----------------------------------------------------------------------------------------

Intercept 7.0755 0.014 503.271 0.000 7.048 7.103

gender\_Female[T.True] 3.4811 0.009 377.143 0.000 3.463 3.499

gender\_Male[T.True] 3.5944 0.009 411.177 0.000 3.577 3.612

age 0.0115 0.000 29.414 0.000 0.011 0.012

seniority 0.1090 0.004 27.196 0.000 0.101 0.117

==============================================================================

Omnibus: 14.356 Durbin-Watson: 1.917

Prob(Omnibus): 0.001 Jarque-Bera (JB): 18.310

Skew: -0.180 Prob(JB): 0.000106

Kurtosis: 3.556 Cond. No. 1.07e+16

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 1.68e-26. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

**Appendix C**

**Python Code**

##############################################################  
# Author: Laurence T. Burden  
# for Purdue Global University  
#  
# Unit 3 Assignment / Module 3 Part 1 Competency Assessment  
# Predicting Gender-Based Salary Gap  
##############################################################  
  
# Imports  
import sys  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
import statsmodels.formula.api as sm  
from sklearn.linear\_model import LinearRegression  
from datetime import datetime  
  
# Ignore warnings  
if not sys.warnoptions:  
 import warnings  
  
warnings.simplefilter("ignore")  
  
# Set Pandas to display more columns when printing  
pd.set\_option('display.max\_columns', 25)  
  
# Output Header  
print('Unit 3 Assignment / Module 3 Part 1 Competency Assessment Output\n')  
print(datetime.now().strftime("%m/%d/%Y %H:%M:%S"), '\n')  
  
# Import and explore the quality of the dataset  
df = pd.read\_csv('data.csv')  
  
# Print data types, shape, first few rows, and last few rows  
print('### Data Types ###')  
print(df.dtypes, '\n')  
print('### Data Frame Shape ###')  
print(df.shape, '\n')  
print('### Data Frame Head ###')  
print(df.head(), '\n')  
print('### Data Frame Tail ###')  
print(df.tail(), '\n')  
print('### Data Frame Info ###')  
print(df.info(), '\n')  
  
# Check for missing and duplicated values  
print('### Data Frame Counts ###')  
print(df.count(), '\n')  
print('### Duplicates ###')  
print(df[df.duplicated(keep=False)], '\n')  
  
# Describe the data  
print(df.describe(), '\n')  
  
# Print null hypothesis  
print('Null Hypothesis: Gender has no effect on base pay')  
  
femaleInfo = df[df['gender'] == 'Female']  
maleInfo = df[df['gender'] == 'Male']  
plt.scatter(femaleInfo['age'], femaleInfo['basePay'], color='red', label='Female', alpha=0.8)  
plt.scatter(maleInfo['age'], maleInfo['basePay'], color='blue', label='Male', alpha=0.8)  
plt.legend()  
plt.title('Base Pay Versus Age Split by Men and Women')  
plt.show()  
  
# Wrangle the data  
# Create dummy variable for gender for use in regression (1 for male and 0 for female)  
df = pd.get\_dummies(df, columns=['gender', 'edu'])  
print(df.head(), '\n')  
  
# Group ages into 5 age groups  
labels = ['18-26', '27-36', '37-47', '48-57', '57-65']  
df['AgeGroup'] = pd.qcut(df['age'], q=5, labels=labels)  
print('### Age Bins ###')  
print(df['AgeGroup'].value\_counts(), '\n')  
  
# Plot the age groups  
plt.hist(df['AgeGroup'], bins=5, align='right')  
plt.title('Age Group Counts')  
plt.show()  
  
# Natural log of base pay  
df['logBasePay'] = np.log(df['basePay'])  
print('### Base Pay Log ###')  
print(df['logBasePay'].head(), '\n')  
  
# Linear Regression of logBasePay ~ age  
sns.regplot(x='age', y='logBasePay', data=df)  
plt.title('Linear Regression of Log of Base Pay and Employee Age')  
plt.show()  
  
# Linear Regression of logBasePay ~ gender\_Female  
# reshape data to fit the linear regression method  
gender = df['gender\_Female'].to\_numpy()  
logBasePay = df['logBasePay'].to\_numpy()  
X = gender.reshape(-1, 1)  
y = logBasePay.reshape(-1, 1)  
  
# Fit and score the linear regression model  
lm = LinearRegression()  
lm.fit(X, y)  
r2 = lm.score(X, y)  
  
# Print results  
print('### Linear Regression of Gender and Log of Base Pay ###')  
print('Intercept: ', lm.intercept\_)  
print('Coefficient: ', lm.coef\_)  
print('R-Squared: ', r2, '\n')  
  
# Plot the linear regression  
sns.regplot(x='gender\_Female', y='logBasePay', data=df)  
plt.title('Linear Regression of Log of Base Pay and Employee Gender')  
plt.legend(['0.0 = Male', '1.0 = Female'], markerscale=0, handlelength=0)  
plt.show()  
  
# Run the multiple regression  
# Create the model  
model = sm.ols(data=df, formula="basePay ~ gender\_Female + gender\_Male + age + seniority")  
  
# fit the model  
result = model.fit()  
  
# View results  
print('### OLS of Original Base Pay Data ###')  
print(result.summary())  
  
# Recreate the model with the log of base pay  
model = sm.ols(data=df, formula="logBasePay ~ gender\_Female + gender\_Male + age + seniority")  
  
# fit the model  
result = model.fit()  
  
# View results  
print('### OLS of Natural Log of Base Pay Data ###')  
print(result.summary())