- Python Data Visualization & Storytelling Project -

Jaejoon JJ Lee

Tech Industry Compensation: A Data Visualization & Storytelling Analysis (Python)

Introduction:

This project explores how experience, education, and demographics relate to total compensation in the U.S. tech industry using Python-based data visualization and statistical modeling with the given dataset techSalaries2017 (2).csv.
I focus on transforming raw salary data into visual narratives that explain real-world compensation patterns through regression analysis and distribution visualizations.

Tools used: pandas, matplotlib, seaborn, scikit-learn

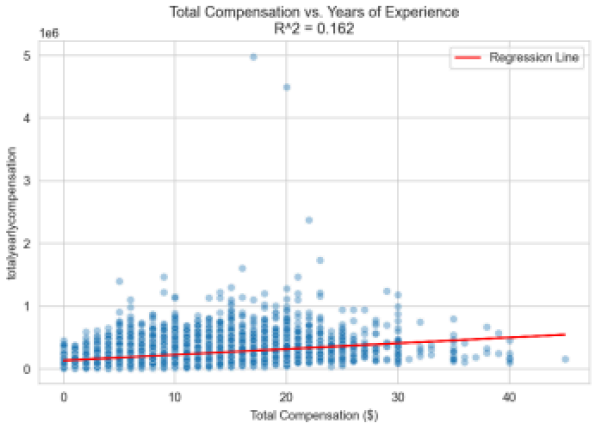
Please load and use the “techSalaries2017.csv” data file. This dataset contains self-reported information on over 62,000 workers in the US tech industry in 2017.
The first row represents the column header. Each row after that represents the information of one person.
Columns represent (in order):
1) Company where they work
2) Job title
3) Office location
4) Total annual compensation (in $)
5) Base salary (in $)
6) Value of stock grants (in $)
7) Bonus payments (in $)
8) Years of relevant experience (in years)
9) Time with this company (in years)
10) Gender (self-reported)
11) Terminal Degree is Masters (1 = yes)
12) Terminal Degree is Bachelors (1 = yes)
13) Terminal Degree is Doctorate (1 = yes)
14) Terminal Degree is High School (1 = yes)
15) Terminal Degree is some college (1 = yes)
16) Self-identifies as Asian (1 = yes)
17) Self-identifies as White (1 = yes)
18) Self-identifies as Multi-Racial (1 = yes)
19) Self-identifies as Black (1 = yes)
20) Self-identifies as Hispanic (1 = yes)
21) Race as a qualitative variable
22) Education as a qualitative variable
23) Age (in years)
24) Height (in inches)
25) Zodiac sign (Tropical calendar, 1 = Aries, 12 = Pisces, with everything else in between)
26) SAT score
27) GPA
We/you will want to use most of these variables in prediction models.
This data is self-reported, so sometimes it will be missing if the person (for whatever reason) did not provide this information. For instance, the information on education (variables 11-15) is only meaningfully interpretable for any given row, if the corresponding value in variable 22 is not “NA”. NA indicates missing data. The same is true for variables 16 to 21.
Mission command approach: As per §4.5 of the Sittyba, we will tell you what to do (“answer these questions”), not how to do it. That is up to you. However, we want you to:
a) Do the homework yourself. Do not copy answers from someone else.
b) Restrict your methods (for now) to what was covered in the lecture/lab (in other words, linear regression, regularized regression and logistic regression)
c) Include the following elements in your answer (so we can grade consistently):

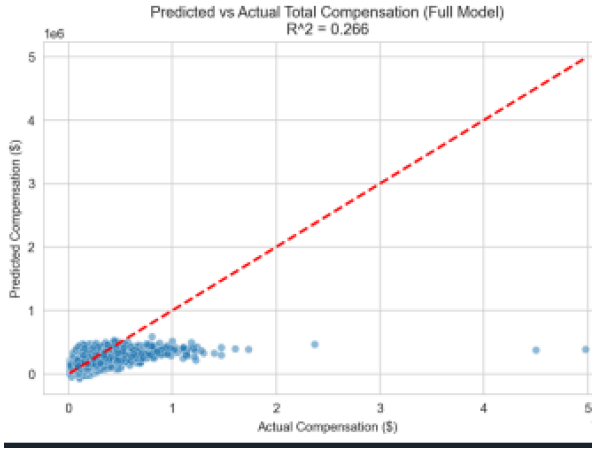
Each answer should contain these elements:
1) A brief statement (~paragraph) of what was done to answer the question (narratively explaining what you did in code to answer the question, at a high level).
2) A brief statement (~paragraph) as to why this was done (why the question was answered in this way, not by doing something else. Some kind of rationale as to why you did x and not y or z to answer the question – why is what you did a suitable approach?).
3) A brief statement (~paragraph) as to what was found. This should be as objective and specific as possible – just the results/facts. Do make sure to include numbers and a figure (=a graph or plot) in your statement, to substantiate and illustrate it, respectively.
4) A brief statement (~paragraph) as to what you think the findings mean. This is your interpretation of your findings and should answer the original question.
Note: Brief actually means "brief". There is no need to write a dissertation. There is value to being concise. A couple of pages should be sufficient for the entire report. Do – however – write a report. A data and code-dump is not very useful or valuable in practice. People who pay you so they can ask you questions usually want them answered.

Please answer the following questions in your report:

1. Using multiple linear regression: What is the best predictor of total annual compensation, how much variance is explained by this predictor vs. the full multiple regression model?

I chose "years of experience" as the only predictor of total yearly compensation for the simple linear regression model and variables such as education, race, GPA, SAT, age, etc were chosen for a multiple linear regression model. These two models were performed to compare how much variance each model explains in total yearly compensation. Linear regression is a direct method that quantifies how predictors explain variance in a continuous target variable. "Years of experience" was chosen for a single predictor model since it is a strong variable that serves as the baseline model, heavily affecting total yearly compensation. Then, I have expanded the model with multiple predictors to test if additional variables significantly affect total yearly compensation. Throughout the model, the single-predictor model displayed an r^2 value of 0.162, which explains about 16% of the salary variance. On the other hand, the full multiple regression model had an r^2 value of 0.266, explaining 27% of the variance. Gathering r^2 values and graphs for both models, years of experience is shown as the best single predictor, but explains only the modest portion of salary variation. However, the full multiple regression model conveyed how compensation is affected by multiple variables when other variables were added as r^2 value increased. Lastly, there exist many unobserved variables such as company and position that are likely to play bigger roles as r^2 value is still low.





2. Using ridge regression to do the same as in 1): How does the model change or improve compared to OLS? What is the optimal lambda?

A ridge regression model with 5 fold cross-validation was built with predictors such as years of experience, years at company, GPA, SAT, and age. Standard Scaler was used to apply feature scaling, and chose lambda (optimal regularization strength) from the logarithmic range between 10^-2 and 10^2. (Indicated the lambda value as 'alpha' in code). In order to address potential issues of multicollinearity and overfitting from correlated predictors, ridge regression was used since it shrinks coefficient values to zero without removing features, stabilizing predictions when predictors are interrelated. The output of the model displayed an lambda value of 56.8987, r^2 value of -110,736.46 and RMSE of 43,450,621.97. Negative R^2 indicates how the model performed worse than simply predicting the mean salary as well as extreme underperformance due to r^2 and RMSE values. These outputs of the ridge regression model and its over-penalization (extremely high lambda value) shrunk coefficient toward zero, disabling predictive power. It implies how regularization underperformed was too strong. Additionally, ridge regression didn't improve results and performed worse than OLS from this given data.

```
In [28]: %runfile '/Users/jaejoonlee/
Ridge Regression
Best alpha: 56.8987
Test R^2 Score: -110736.456921
Test RMSE: 43450621.97
```

3. Using Lasso regression to do the same as in 1): How does the model change now? How many of the predictor betas are shrunk to exactly 0? What is the optimal lambda now?

I trained the Lasso regression model with the same standardized predictors as the Ridge regression model. The model searched for an optimal lambda value (Indicated lambda value as 'alpha' in code) that minimizes prediction error where some coefficients will turn to zero by employing cross–validation. Lasso regression serves two functions: regularization and feature selection, which allows some predictors with weak relationships to outcome suitable. Lasso regression was also used since it simplifies the model by automatically erasing less important features. The output of the model came out as an lambda value of 100 and achieved r^2 value of 0.2947 and RMSE value of 109,658.14. Also, the model set 1 out of 5 coefficients to zero, implying how one variable was effectively removed from the model since it was insignificant. Lasso regression provided a model with similar accuracy to OLS with better interpretability by suggesting that only few predictors carry most of the predictive weight, likely years and age relative to SAT and GPA, which were effectively removed by Lasso regression.

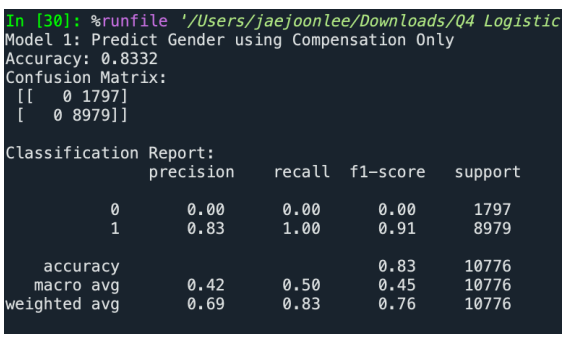
```
In [29]: %runfile '/Users/jaejoonlee/Downloads/
 Data Split Summary:
 -- Training set: 46981 samples
 -- Test set: 15661 samples

Lasso Regression
Best alpha: 100.0000
Test R^2 Score: 0.2947
Test RMSE: 109658.14
Zeroed Coefficients: 1 / 5
```

4. There is controversy as to the existence of a male/female gender pay gap in tech job compensation. Build a logistic regression model (with gender as the outcome variable) to see if there is an appreciable beta associated with total annual compensation with and without controlling for other factors.

A logistic regression model was built to test if total yearly compensation can predict an employee's gender by examining potential gender-based pay patterns in the technology field. Two models were used where the first model used only total yearly compensation to predict gender and the second model was run with additional control variables (years of experience, years at company, GPA, and age) to observe potential effects. Data was preprocessed as I removed missing gender values and set gender objects manually where 1=male, and 0=female. The dataset was split into training and test sets (75/25), and the model was evaluated via accuracy and classification metrics. Logistic regression was used as our target variable gender is binary (either 1 or 0), allowing to observe estimation of the changes in predictors such as compensation to the probability of being male or female. Employing two different models where one was solely dependent on salary and other was

dependent on multiple variables, we can detect whether salary alone is associated with gender differences. Both models demonstrated an equal accuracy value of 0.8332, but the confusion matrix shows how the model predicted every observation as male(1) and none as female(0). The precision and recall scores support this display where the female class had 0 on both precision and recall, whereas male class had precision value of 0.83 and recall value of 1, and F1 score of 0.91. This indicates an imbalanced dataset and implies how adding other predictors would not improve accuracy or classification balance since models resulted in identical outputs. The result of the model suggests how it cannot distinguish gender based compensation mainly due to severe class imbalance in the dataset. Logistic regression classified everyone as male that achieves highest overall accuracy, not reflecting meaningful prediction. Even though the model contains numeric value, it isn't informative as it failed to identify gender-pay gap.

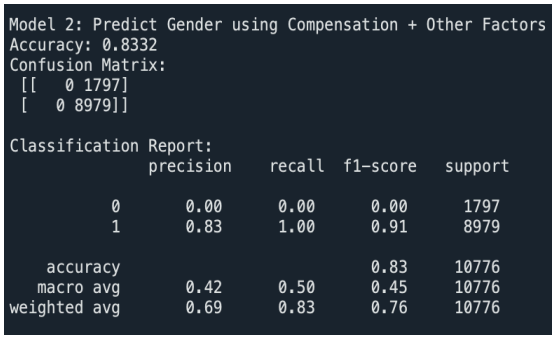


```
In [30]: %runfile '/Users/jaejoonlee/Downloads/Q4 Logistic
Model 1: Predict Gender using Compensation Only
Accuracy: 0.8332
Confusion Matrix:
 [[   0 1797]
 [   0 8979]]

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      1797
           1       0.83      1.00      0.91      8979

    accuracy                           0.83     10776
   macro avg       0.42      0.50      0.45     10776
weighted avg       0.69      0.83      0.76     10776
```



```
Model 2: Predict Gender using Compensation + Other Factors
Accuracy: 0.8332
Confusion Matrix:
 [[   0 1797]
 [   0 8979]]

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      1797
           1       0.83      1.00      0.91      8979

    accuracy                           0.83     10776
   macro avg       0.42      0.50      0.45     10776
weighted avg       0.69      0.83      0.76     10776
```

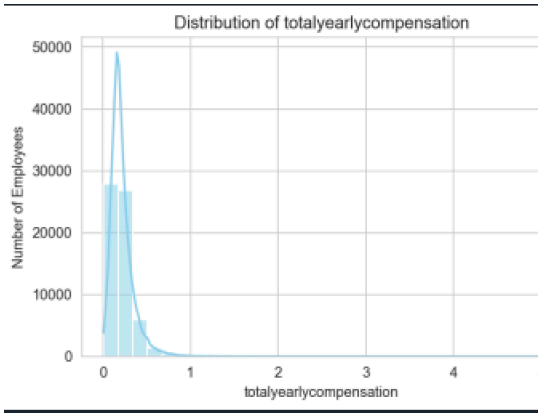
Key Visualization Insights

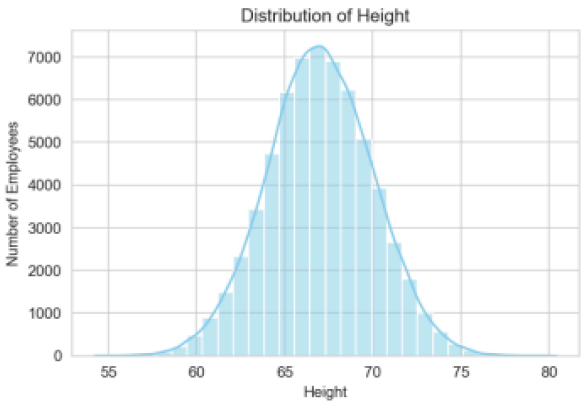
- Compensation increases nonlinearly with years of experience with high variance at senior levels
- Salary distribution is heavily right-skewed, underscoring extreme outliers in executive compensation
- Height and age follows near-normal distributions, which contrasts with income inequality patterns

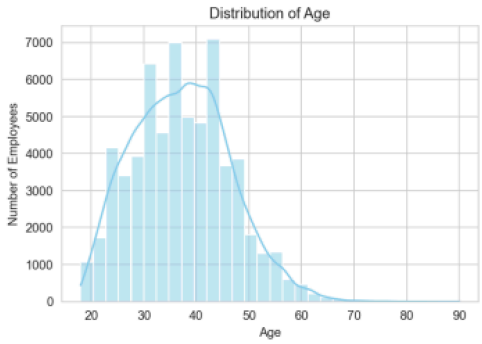
Extra Credit:

a) Is salary, height or age normally distributed? Does this surprise you? Why or why not?

In order to test if salary, height or age are normally distributed, I plotted histograms for total yearly compensation, height, and age for visualization purposes. Histogram is a simple way to determine if data has a bell-shaped (normal) curve, identifying skewness or outliers without running statistical tests. Employing the histogram, salary was right-skewed where most employees earn average salaries, with few earning higher pay. Height was roughly symmetric and age was slightly right-skewed as there were more young workers than older workers in the dataset. Together, none of the variables showed normal distribution, but only height came close to it. This result can be interpreted as salary distributions are almost always skewed since a small group of people earn extremely high salaries. Also, the reason why height came close to normal distribution from the histogram is due to biological variation.







b) Tell us something interesting about this dataset that is not already covered by the questions above and that is not obvious.

One interesting observation I found was that ‘years at company’ and ‘years of experience’ are weakly correlated where many individuals report that they have a considerable amount of experience but only a short tenure at their current company. I was able to infer that there exists high job mobility in the technology field where employees often switch companies for better opportunities and higher pay. However, experience still heavily affects salary compensation regardless of short tenures, which indicates how cumulative expertise and skills are more important than loyalty in a company in the technology industry.

CODE:

Q1 DAT2 Q1 LinearRegression.pdf

—----------

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import warnings
warnings.filterwarnings('ignore')

sns.set_style("whitegrid")
plt.rcParams['figure.figsize'] = (10,5)


df = pd.read_csv('techSalaries2017.csv')
df = df.dropna(subset=['Education','Race'])

target = 'totalyearlycompensation'

predictors = ['yearsofexperience', 'yearsatcompany', 'Masters_Degree', 'Bachelors_Degree',
      'Doctorate_Degree', 'Highschool','Some_College','Race_Asian', 'Race_White','Race_Two_Or_More',
      'Race_Black','Race_Hispanic','Age','Height','SAT','GPA']

X = df[predictors]
y = df[target]

X_single = df[['yearsofexperience']]
y_single = df[target]

model_single = LinearRegression()
model_single.fit(X_single, y_single)

y_pred_single = model_single.predict(X_single)

r2_single = r2_score(y_single, y_pred_single)
rmse_single = np.sqrt(mean_squared_error(y_single, y_pred_single))

plt.figure(figsize=(8,5))
sns.scatterplot(x='yearsofexperience', y = target, data=df, alpha = 0.4)
```

```
plt.plot(df['yearsofexperience'],y_pred_single, color= 'red', label='Regression Line')
plt.title(f"Total Compensation vs. Years of Experience\nR^2 = {r2_single:.3f}")
plt.xlabel("Years of Experience")
plt.xlabel("Total Compensation ($)")
plt.legend()
plt.show()


model_full = LinearRegression()
model_full.fit(X,y)

y_pred_full = model_full.predict(X)

r2_full = r2_score(y, y_pred_full)
rmse_full = np.sqrt(mean_squared_error(y, y_pred_full))

coef_df = pd.DataFrame({
   'Predictor': predictors,
   'Coefficient': model_full.coef_
    }).sort_values('Coefficient', key=abs, ascending = False)

plt.figure(figsize=(8,5))
sns.scatterplot(x=y, y= y_pred_full, alpha=0.5)
plt.plot([y.min(), y.max()],[y.min(), y.max()], 'r--', lw=2)
plt.title(f"Predicted vs Actual Total Compensation (Full Model)\nR^2 = {r2_full:.3f}")
plt.xlabel("Actual Compensation ($)")
plt.ylabel("Predicted Compensation ($)")
plt.show()
```

Q2 DAT2 Q2 RidgeRegression.pdf

—----------

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import RidgeCV
from sklearn.metrics import mean_squared_error, r2_score

df = pd.read_csv('techSalaries2017.csv')

features = ['yearsofexperience', 'yearsatcompany','GPA', 'SAT','Age']
target = 'totalyearlycompensation'
```

```
X = df[features]
y = df[target]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 7)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

lambda_values = np.logspace(-2,2,50)
ridge_cv = RidgeCV(alphas=lambda_values, cv=5)
ridge_cv.fit(X_train_scaled, y_train)

y_pred_ridge = ridge_cv.predict(X_test)

r2_ridge = r2_score(y_test, y_pred_ridge)
rmse_ridge = np.sqrt(mean_squared_error(y_test, y_pred_ridge))

print("Ridge Regression")
print(f"Best alpha: {ridge_cv.alpha_:.4f}")
print(f"Test R^2 Score: {r2_ridge:4f}")
print(f"Test RMSE: {rmse_ridge:.2f}")
```

Q3 DAT2 Q3 LassoRegression.pdf

—----------

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LassoCV
from sklearn.metrics import mean_squared_error, r2_score

df = pd.read_csv('techSalaries2017.csv')

features = ['yearsofexperience', 'yearsatcompany','GPA', 'SAT','Age']
target = 'totalyearlycompensation'

X = df[features]
y = df[target]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 7)
```

```
print(f" Data Split Summary:")
print(f" -- Training set: {X_train.shape[0]} samples")
print(f" -- Test set: {X_test.shape[0]} samples\n")

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

lambda_grid = np.logspace(-2,2,50)
lasso_model = LassoCV(alphas=lambda_grid, cv=5, random_state = 7, max_iter = 15000)
lasso_model.fit(X_train_scaled, y_train)

y_pred_lasso = lasso_model.predict(X_test_scaled)

lasso_r2 = r2_score(y_test, y_pred_lasso)
lasso_rmse = np.sqrt(mean_squared_error(y_test, y_pred_lasso))

zero_count = np.sum(lasso_model.coef_ == 0)
total_count= len(lasso_model.coef_)

print("Lasso Regression")
print(f"Best alpha: {lasso_model.alpha_:.4f}")
print(f"Test R^2 Score: {lasso_r2:.4f}")
print(f"Test RMSE: {lasso_rmse:.2f}")
print(f"Zeroed Coefficients: {zero_count} / {total_count}")
```

Q4 DAT2 Q4 Logistic Regression.pdf

—----------

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# ================================
# Load and prepare dataset
# ================================
df = pd.read_csv('techSalaries2017.csv')
```

```
# Drop rows with missing gender or compensation
df = df.dropna(subset=['gender', 'totalyearlycompensation'])

# Encode gender manually (append style)
gender_encoded = []
for g in df['gender']:
    if g == 'Male':
        gender_encoded.append(1)
    else:
        gender_encoded.append(0)

df['gender_encoded'] = gender_encoded

# ================================
# Model 1: Compensation only
# ================================
X1 = df[['totalyearlycompensation']]
y = df['gender_encoded']

X_train1, X_test1, y_train1, y_test1 = train_test_split(X1, y, test_size=0.25, random_state=7)

model1 = LogisticRegression(max_iter=1000)
model1.fit(X_train1, y_train1)
y_pred1 = model1.predict(X_test1)

acc1 = accuracy_score(y_test1, y_pred1)

print("Model 1: Predict Gender using Compensation Only")
print(f"Accuracy: {acc1:.4f}")
print("Confusion Matrix:\n", confusion_matrix(y_test1, y_pred1))
print("\nClassification Report:\n", classification_report(y_test1, y_pred1))

# ================================
# Model 2: Compensation + controls
# ================================
X2 = df[['totalyearlycompensation', 'yearsofexperience', 'yearsatcompany', 'GPA', 'Age']]
y = df['gender_encoded']

X_train2, X_test2, y_train2, y_test2 = train_test_split(X2, y, test_size=0.25, random_state=7)

scaler = StandardScaler()
X_train2_scaled = scaler.fit_transform(X_train2)
X_test2_scaled = scaler.transform(X_test2)
```

```
model2 = LogisticRegression(max_iter=1000)
model2.fit(X_train2_scaled, y_train2)
y_pred2 = model2.predict(X_test2_scaled)

acc2 = accuracy_score(y_test2, y_pred2)

print("\nModel 2: Predict Gender using Compensation + Other Factors")
print(f"Accuracy: {acc2:.4f}")
print("Confusion Matrix:\n", confusion_matrix(y_test2, y_pred2))
print("\nClassification Report:\n", classification_report(y_test2, y_pred2))
```

Extra Credit A DAT2 EC(a).pdf

—----------

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


df = pd.read_csv('techSalaries2017.csv')

cols = ['totalyearlycompensation', 'Height', 'Age']

for col in cols:
    plt.figure(figsize=(6,4))
    sns.histplot(df[col].dropna(), bins=30, kde=True, color='skyblue')
    plt.title(f"Distribution of {col}")
    plt.xlabel(col)
    plt.ylabel("Number of Employees")
    plt.show()
```