Summative Assessment 2

Using Monte Carlo Sampling Techniques on Salary vs Years of Experience Dataset

Submitted by:

Gonzales, Dwight
Khafaji, Mostafa A.
Quintero, John Carlos

In partial fulfillment of the requirements for:
APM1210 - Statistical Computing

Submitted to:
Dr. May Anne Caspe Tirado

Data Selection

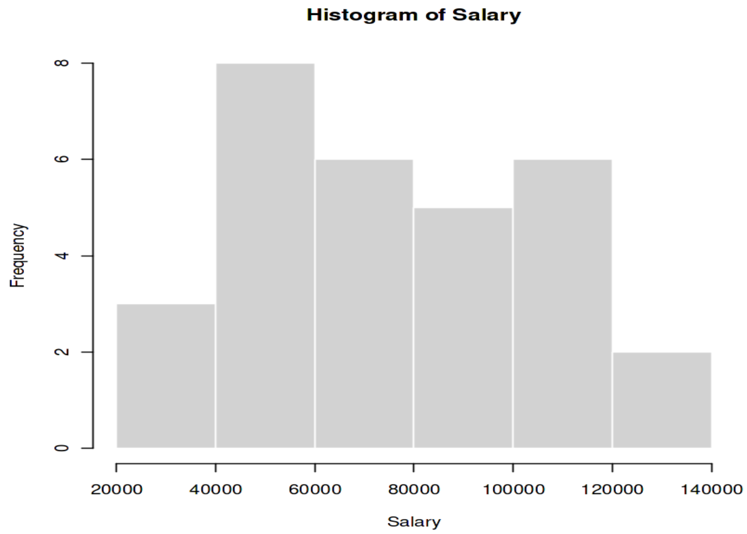
Our team aims to investigate the relationship between years of experience and salary of engineers. Using the real-world dataset from Kaggle with a total of 30 observations, we explore the data using Monte Carlo techniques.

Using Monte Carlo techniques, this project can assess trends and uncertainty levels for companies or individuals in setting salary expectations based on years of experience. We are able to make reasonable assumptions regarding the population mean and variance, the bias of the sample. We are also able, using Monte Carlo techniques, create both frequentist and bayesian linear regression models in order to make estimates on the salary when given the years of experience. We have also studied the distribution of the sample, drawing conclusions on possible parent distributions using Monte Carlo.

The dataset that was used for this project can be accessed with : https://www.kaggle.com/datasets/rohankayan/years-of-experience-and-salary-dataset/data.

Bootstrap and Jackknife

Theoretical Mean Salary Observation



```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
37731   56721   65237   76003  100545  122391
```

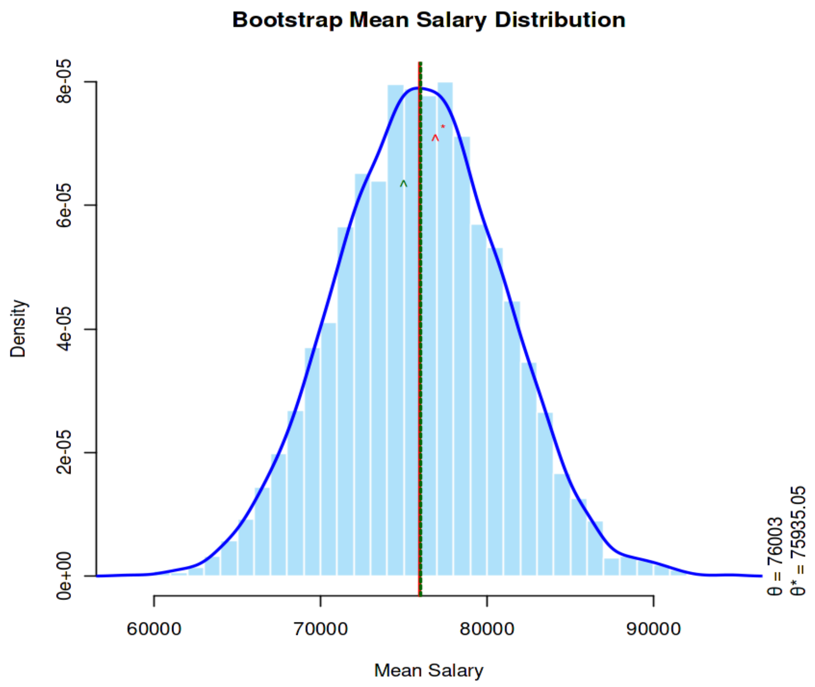
Figure 2.1

It was found out that the mean salary of engineers based on the data is $76,003 (Figure 1.1). To test the reliability and accuracy of this estimate, we calculate the confidence interval (CI), standard error (SE), and variance to understand how much the sample mean may vary from the true population mean.

Based on the data , the standard error is $5,005.17. Standard Error (SE) is a measurement of the variability from our mean to the samples. We also calculate the 95% confidence interval (CI) with [$38,899.70, $122,014.73]. This means we are 95% confident that the true average salary of all engineers falls within this range. It was also found out that the salaries differ slightly among engineers, with a variance of 25051699 and standard deviation of $5,005.

We want to evaluate further on how reliable the computed salaries are. We can do this by applying resampling methods to assess the Bias and Variance and compare it to our theoretical computed values..

Bootstrap Resampling Method



```
Min. 1st Qu.  Median   Mean 3rd Qu.    Max.
58143   72541   75933  75935   79253   94718
```

Figure 2.2

.

We resampled our data for 5000 times with replacement using Bootstrap Method to assess the mean salary distribution from the new resampled data.

The Bootstrap resampled data computed the bias for mean salary of -67.95098, which means the mean of this sampled data is 76,003-67.95098 = $75,935.04. This is slightly lower compared to our theoretical mean of $76,003. We also compute the 95% CI [$66,288.51,$85,667.32] to check the confidence level of the mean range. As expected, the standard error is much lower, $4969.622.

The Bootstrap resampled data values gave us a little difference from what our theoretical values are. This indicates that there is a little uncertainty from our original mean but it does not affect it too much.

Jackknife Resampling Method

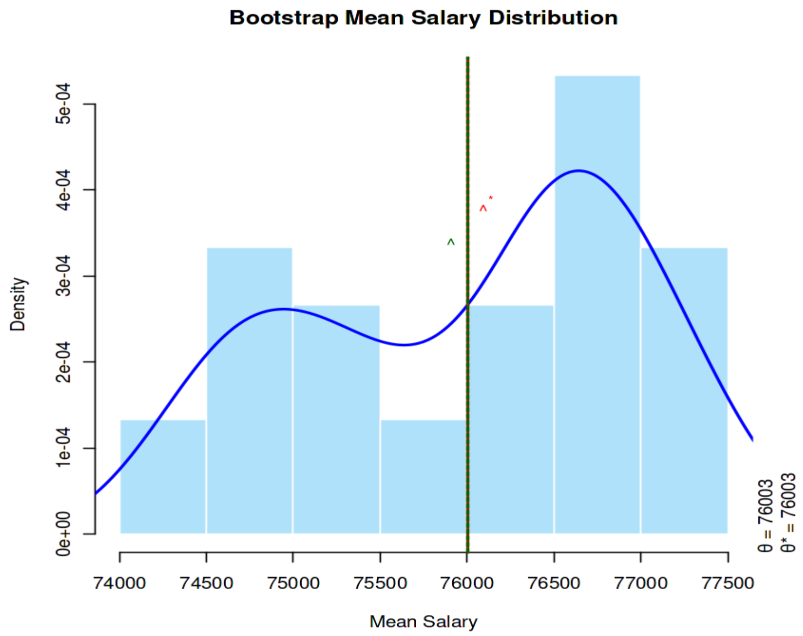


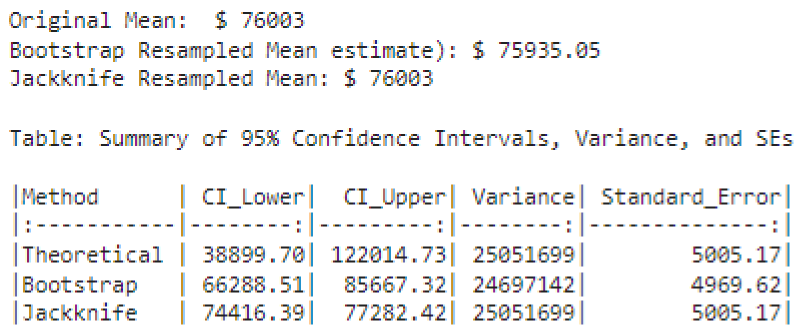
Figure 2.3

Further on, we used Jackknife Resampling method to re-assess new computed statistical values from our data. This method systematically leaves out one observation at a time from the sample and recalculates the mean each time.

The Jackknife resampled data computed the bias for mean salary of 0, which means the mean of this sampled data is 76,003. The computed mean of salary from the resampled data is the same from our theoretical value. We also compute the 95% CI [$74416.39,$77282.42] to check the confidence level of the mean range. As expected, the standard error is the same from our theoretical value, $5005.17

The Jackknife resampled computed data is almost the same as from what our theoretical values are. It just gives us a narrow range of our mean compared to our theoretical CI. This indicates that there is a little uncertainty as well from our original mean but it does not affect it too much.

Theoretical vs Computed Estimates



```
Original Mean:  $ 76003
Bootstrap Resampled Mean estimate): $ 75935.05
Jackknife Resampled Mean: $ 76003

Table: Summary of 95% Confidence Intervals, Variance, and SEs

|Method      | CI_Lower|  CI_Upper| Variance| Standard_Error|
|:-----------|--------:|---------:|--------:|--------------:|
|Theoretical | 38899.70| 122014.73| 25051699|        5005.17|
|Bootstrap   | 66288.51|  85667.32| 24697142|        4969.62|
|Jackknife   | 74416.39|  77282.42| 25051699|        5005.17|
```

Figure 2.4

As we observed, the Jackknife method and the theoretical value provided us the same values. This can be due .This result can be due to Jackknife method of resampling data (only one value removed per iteration) with a small dataset. As a result, the confidence interval is narrow, and it may underestimate the true uncertainty.

Meanwhile, Bootstrap Resampling method have a lower variance, mean and standard error. This can be due to the technique of Bootstrap by resampling with replacement from the data to create many resampled datasets. This allows it to explore a wider range of possible outcomes, including more variability in the estimate.

All three values agree closely, showing that the mean salary is a stable and reliable estimate. However, the bootstrap method provides the best balance of realism to estimate uncertainty.

Resampling for Model Validation

We now proceed on fitting a regression model using Bootstrap Resampling method to predict the salary using years of experience as a predictor.

Data Visualization for Key Assumption

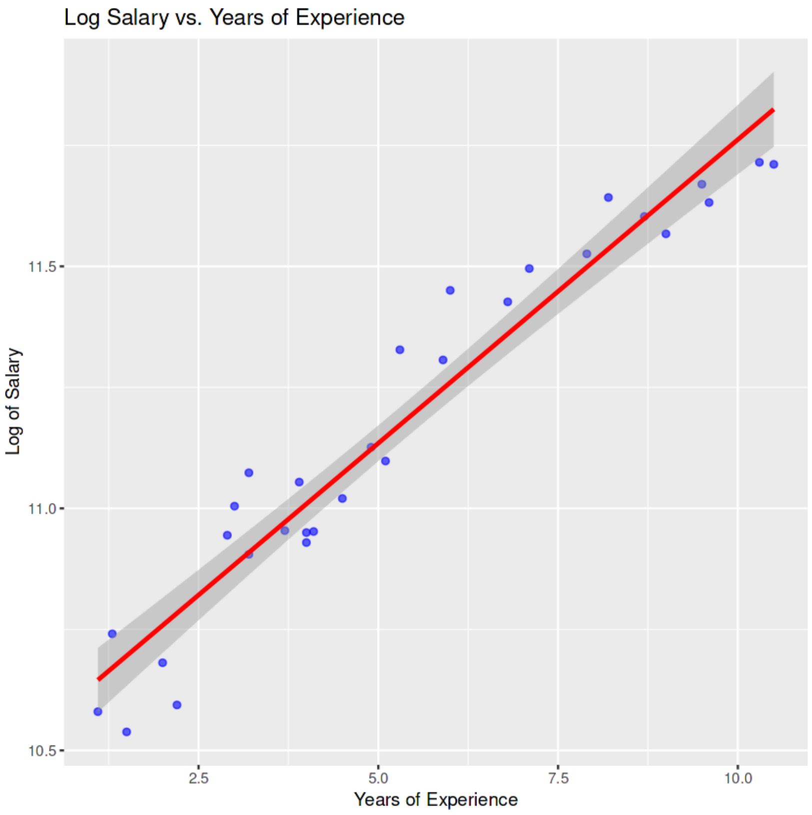


Figure 3.1

Correlation of Predictor and Target Variable is: 0.9653844

In the plot we observed a clear upward trend as salaries tend to increase as years of experience increase. The correlation test also shows that the predictor (years of experience) and target variable (salary) is highly correlated. This indicates a linear relationship between our variables which is an important assumption when we are fitting our data to the model.

We also normalized the data by using log transformation to ensure that we can meet the assumptions of homoscedasticity (equal variance) and normality of residuals in regression analysis.

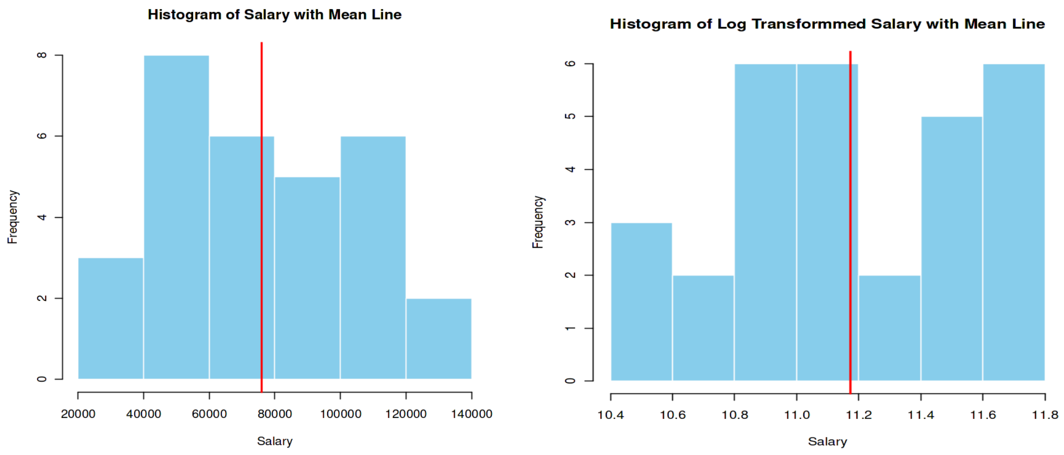


Figure 3.2

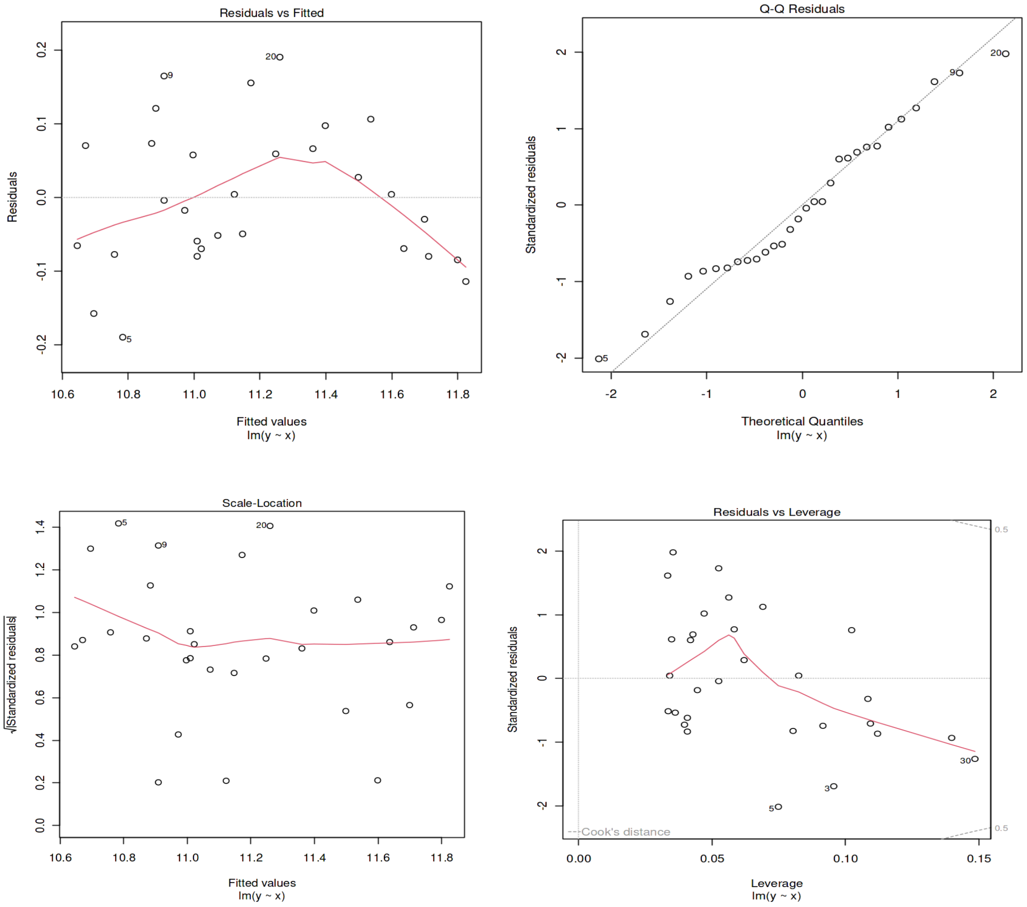
As shown on the left histogram, the original distribution of salary is right skewed. Showing that there is a larger observation for lower values of salary. Applying log transformation, the distribution becomes more symmetric (right histogram). The transformation of data helps us to stabilize variance and improve the normality assumption for our regression model.

We can now observe assumptions such as homoscedasticity, normality test and outliers using plots.

Let us now fit our model:

```
y<-df$Salary
x<-df$YearsExperience
lr_model<-lm(y~x) # OUR MODEL
```

```
plot(lr_model)
```



Residuals vs Fitted

Q-Q Residuals

Scale-Location

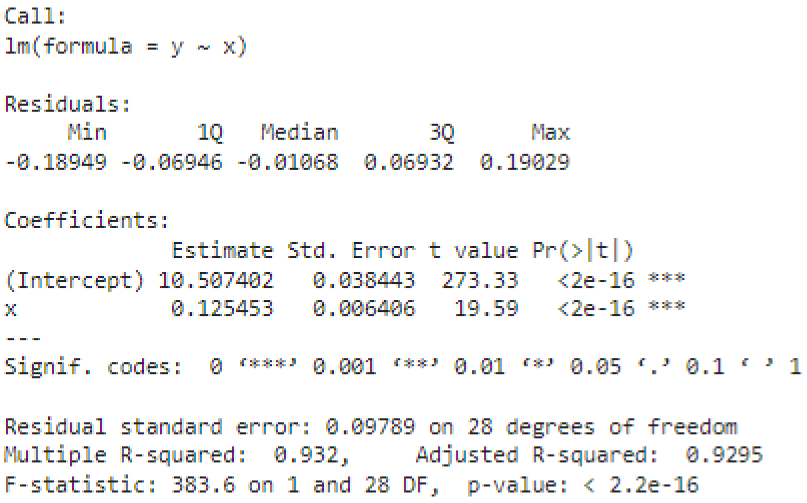
Residuals vs Leverage

Figure 3.3

The residual vs fitted plot shows that the points are randomly scattered around the horizontal line near zero; it suggests that the assumptions are met for homoscedasticity. QQ-Residuals shows that our data points follow the plotted line, this indicates normality and no significant outliers. We can also check whether there are no significant outliers on Residuals vs Leverage plot, as we observed that there are no data points falling off from the Cook's Distance.

Since we already check the assumptions, we can now proceed on fitting our regression model.

Linear Regression Model (Before Bootstrap)



```
Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.18949 -0.06946 -0.01068  0.06932  0.19029 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 10.507402   0.038443  273.33   <2e-16 ***
x            0.125453   0.006406   19.59   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.09789 on 28 degrees of freedom
Multiple R-squared:  0.932,	Adjusted R-squared:  0.9295 
F-statistic: 383.6 on 1 and 28 DF,  p-value: < 2.2e-16
```

Figure 3.4

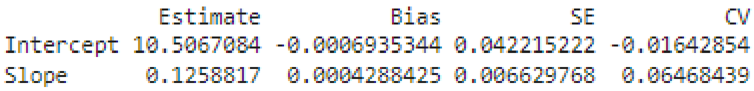
Based on the summary, our model fits well. We can see that the residuals are not too far away based on their min and max value. Which is good because the model prediction is consistently close to the actual salary values, with no extreme underestimations or overestimations.

Model Interpretation

$$\hat{y} = 10.5074 + 0.1255x$$

When the person has no experience = 0. The salary is expected to be log .507402 (Intercept) or exp(Intercept) =$36,585.31. As for every one unit increase in experience, the salary increases by 0.125 or exp(0.125) = 1.13 - 1 x 100 = 13%.

Linear Regression (Applying Bootstrap with 5000 resampled data)



```
             Estimate          Bias          SE          CV
Intercept 10.5067084 -0.0006935344 0.042215222 -0.01642854
Slope      0.1258817  0.0004288425 0.006629768  0.06468439
```

Figure 3.5

It was observed that after resampling the data, it gives a slightly lowered intercept and slightly higher slope.

Model Interpretation (Bootstrap)

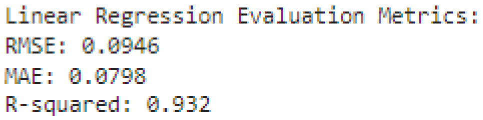
$$\hat{y} = 10.5066276 + 0.1259256*x$$

When the person has no experience = 0 the salary is expected to be log 10.5066276 or exp(10.5066276) = $36556.99. As for every one unit increase in experience, the salary increases by 0.1259256 or exp (0.1259256) = 1.13 - 1 x 100 = 13%

As we can see, there is almost no difference between the original estimates from our linear model and those obtained after resampling using the Bootstrap method. The bias of -0.0006935344 for Intercept and 0.0004288425 for slope followed by the standard error of 0.04 for intercept and 0 for slope indicates that our model estimates are stable and not overly sensitive to the specific sample used.

Linear Model Performance Results

We predict the log salaries of our data using our model. We used the function predict(lr_model) to assess the result and performance.



```
Linear Regression Evaluation Metrics:
RMSE: 0.0946
MAE: 0.0798
R-squared: 0.932
```

Figure 3.6

Based on the Root Mean Squared Error (RMSE), the model predictions deviate from the actual values by about 0.0946 units. The Mean Absolute Error (MAE) is a measure of errors between paired observations expressing the same phenomenon which is 0.0798 units. The model explained our data 93% which is high and a good indication that our model can really fit into our data.

We can now compare it to a different model to check what model is much more efficient to use. For our second model we use Regression Trees.

Regression Trees

Now, as for another model, we decided to use regression trees, as this is trained to model the relationship between years of experience and salary using the given dataset.

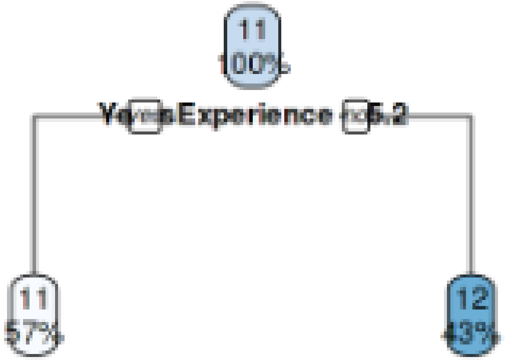


Figure 3.7. Regression Tree Structure

In terms of training the decision tree, the decision tree was generated by the rpart() function. This indicates the splitting of the years of experiences that can be observed in nodes 11 and 12. The root node, which is located at the upper part, shows that the years of experience in the given data set are split by 5.2 years. The left branch, which is node 11, indicates that for workers ≤ 5.2, the predicted log-salary is ~10.97, and the right branch is for people/workers with YearsExperience > 5.2; the predicted log-salary is ~11.51..

With the tree, we can also show the fit of our model before bootstrapping, the scatterplot overlays the tree model prediction on top of the actual data.

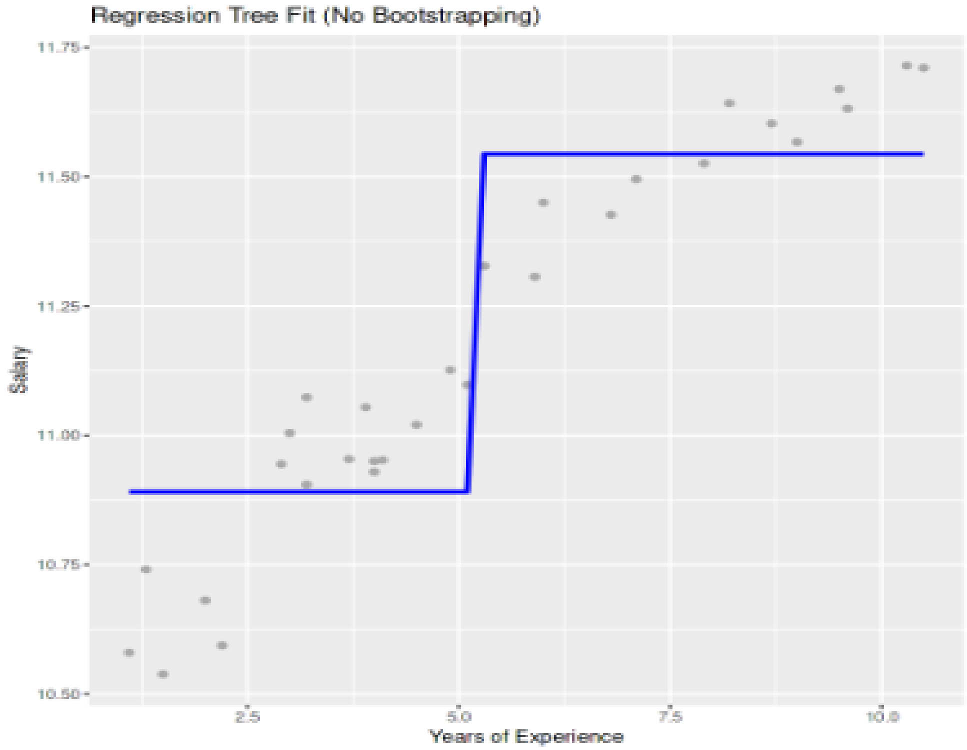
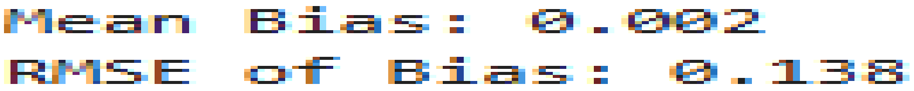


Figure 3.8. Regression Tree Fit (Without Bootstrapping)

In the scatterplot, the dots represent the log-salary observations, and the blue line represents the regression tree's predicted values. As we can observe, the tree makes piecewise constant predictions, and also the jump at 5.2 years of experience corresponds to the tree split. This scatterplot is without the bootstrap sampling, meaning it's trained on the full dataset at once without random resampling.

Now, we will compute the bias,



```
Mean Bias: 0.002
RMSE of Bias: 0.138
```

Figure 3.9. Bias of the Model (Regression Trees)

There was very little systematic overestimation or underestimate in the bootstrapped regression tree model, as seen by its extremely low mean bias (0.002). There was a slight overall difference between the sample's observed salaries and projections, as indicated by the bias's RMSE of 0.138.

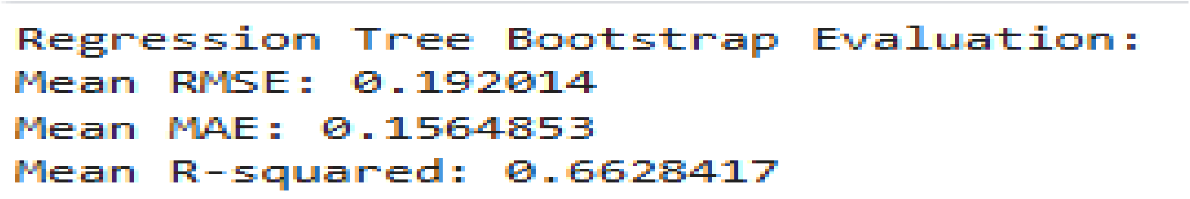
Now, we will use the bootstrap resampling,

Bootstrap Evaluation - With OOB

In obtaining a more reliable performance, we used a bootstrap resampling with 5000 iterations. In each iteration:

- A sample with replacement used for training
- The out-of-bag (OOB) data is used for testing

For the result of this code, this is the regression tree bootstrap evaluation.



```
Regression Tree Bootstrap Evaluation:
Mean RMSE: 0.192014
Mean MAE: 0.1564853
Mean R-squared: 0.6628417
```

Figure 3.9. Regression Tree Results

On average, the predictions of the salary deviate from the actual salary by the value of RMSE, which is 0.192, which is on the same scale as the salary. And the average absolute error between the predicted and actual is 0.156. This indicates that the tree is off by about 0.192 on average, which indicates a moderate prediction accuracy. And lastly, the regression tree explains about 66.2 percent of the variables in the given data.

Comparison of Linear Regression and Regression Tree

After doing the linear regression and regression tree model, now we will compare the model we used in this task to know what model is better for this dataset.

Linear Regression	**Regression Trees**
RMSE: 0.0946	RMSE: 0.192
MAE: 0.0798	MAE: 0.156
R-squared: 0.932	R-squared: 0.662

Figure 4.0. Table of Results in Linear Regression and Regression Tree

After applying bootstrap resampling, the regression tree model shows and gives us a value of RMSE 0.192, MAE of 0.156, and R-squared of 0.662. This means that regression trees models are showing moderate prediction accuracy with 66.2% explained in the salary data. On the other hand, using linear regression, this model gives us a RMSE of 0.0946, MAE of 0.0798, and R-squared of 0.932. As we observed, using linear regression gives us more precise and accurate results than using the regression tree model as the value we got in RMSE and MAE in linear regression are lower than the value in the regression tree model. The R-squared also in linear regression is an indicator that almost all of the variables in the given data were explained with having 93.2%. In conclusion, linear regression reflects a strong predictive model and a better fit to the salary dataset. This comparison also shows that the linear regression outclassed the regression trees for this given dataset, likely because of the reason that the relationship between the predictor and salary is linear and not complex. Therefore, in the salary dataset, we can clearly say that linear regression is the better fit for this dataset than regression trees.

Permutation Tests

Fitting Distributions

Given the small sample size, we are not confident on how reliable the central limit theorem is when describing the distribution of the years of experience of the engineers in the data set.

For reference, the density histogram of the sample showed the following:

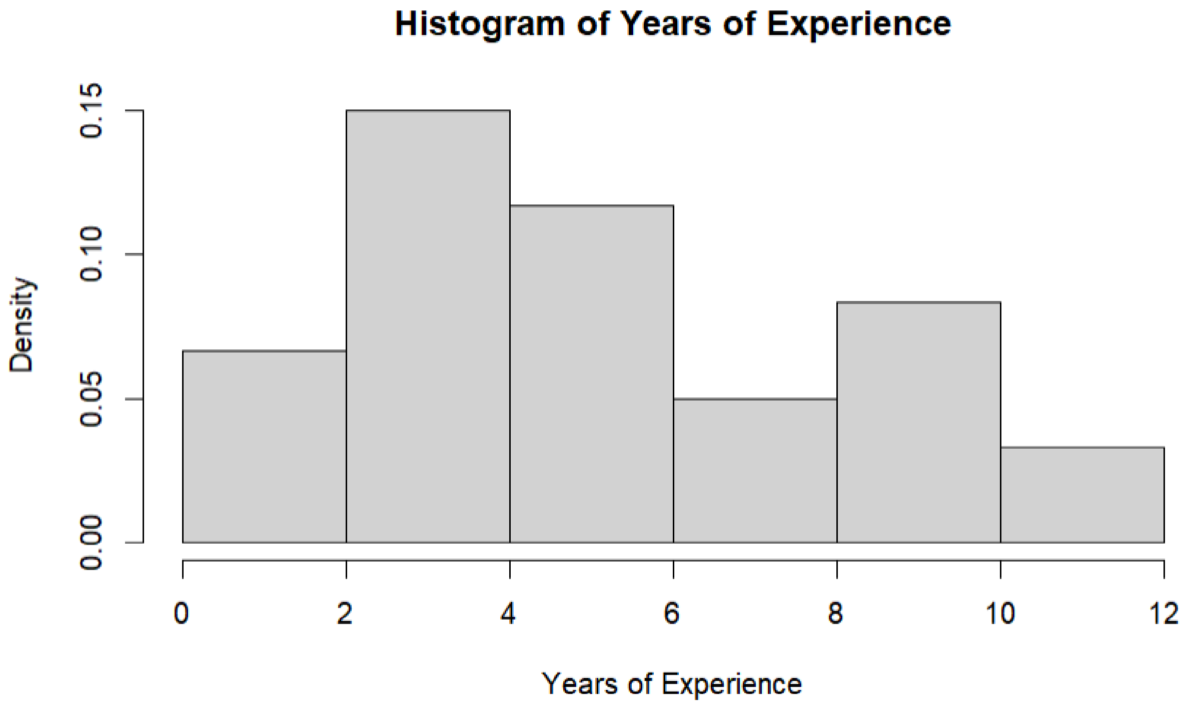


Figure 4.1. Histogram of Years of Experience

The sample distribution above does not give confidence that the sample follows a normal distribution. We can see that there is some multimodality in the data, with the peak at 2-4 years of experience and another, albeit smaller, peak, at 8-10 years.

Hence, we can test, using Kolmogorov-Smirnov tests in conjunction with permutation testing, whether a mixture distribution or a single distribution fits the sample data best.

First, let's find one distribution that could fit the sample data. Using R's fitdistrplus package, we can find likely distribution candidates that can fit the sample. The function fitdistrplus::descdist yields the following Cullen and Frey graph:

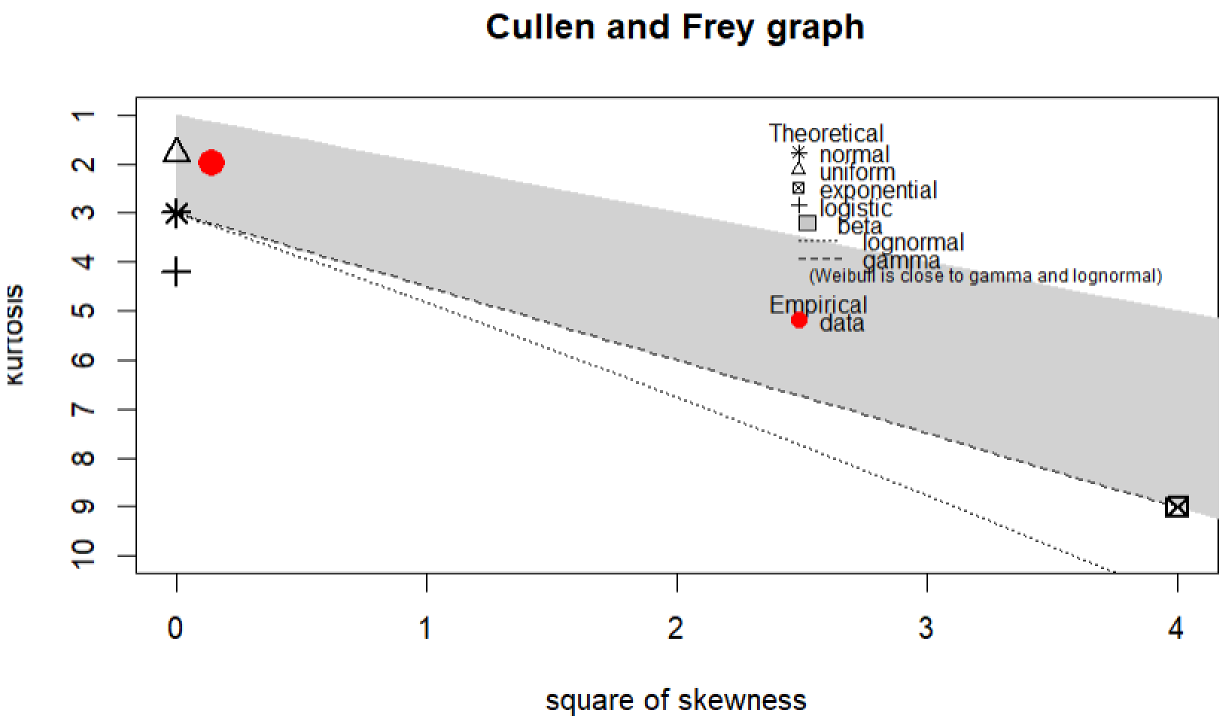


Figure 4.2. Years of Experience candidate distributions

The Cullen and Frey graph gives us two likely candidates: the Normal distribution and the Weibull distribution.

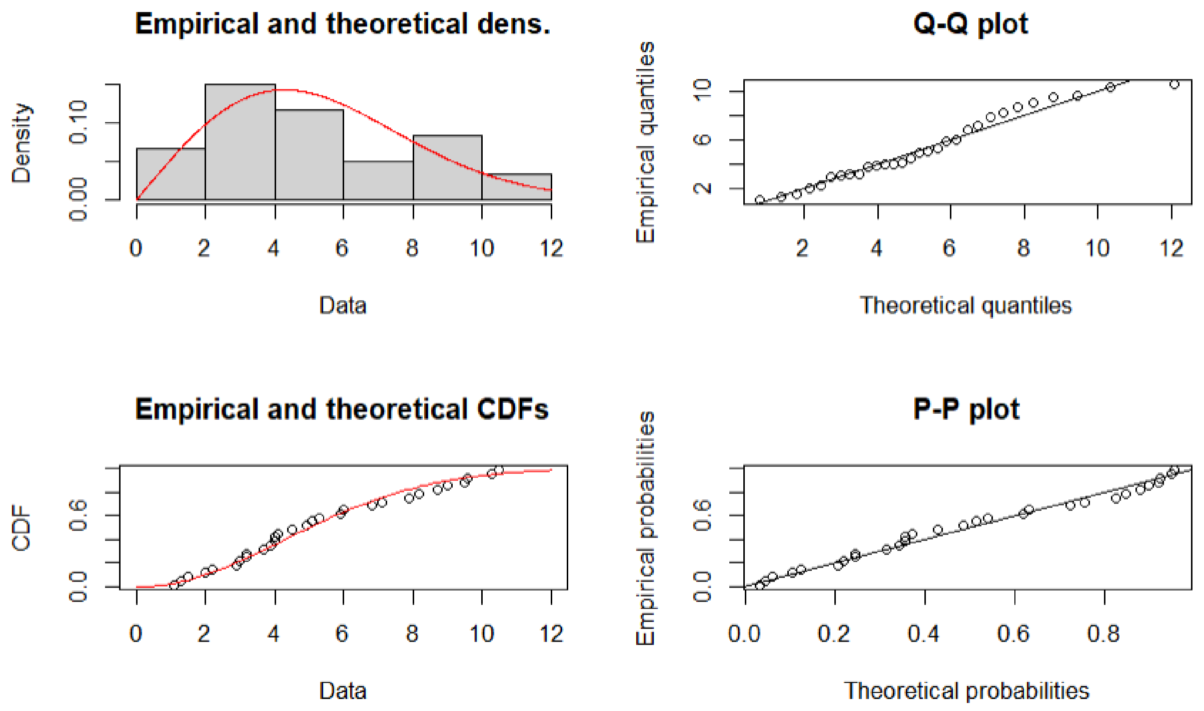


Figure 4.3. Weibull Distribution fit

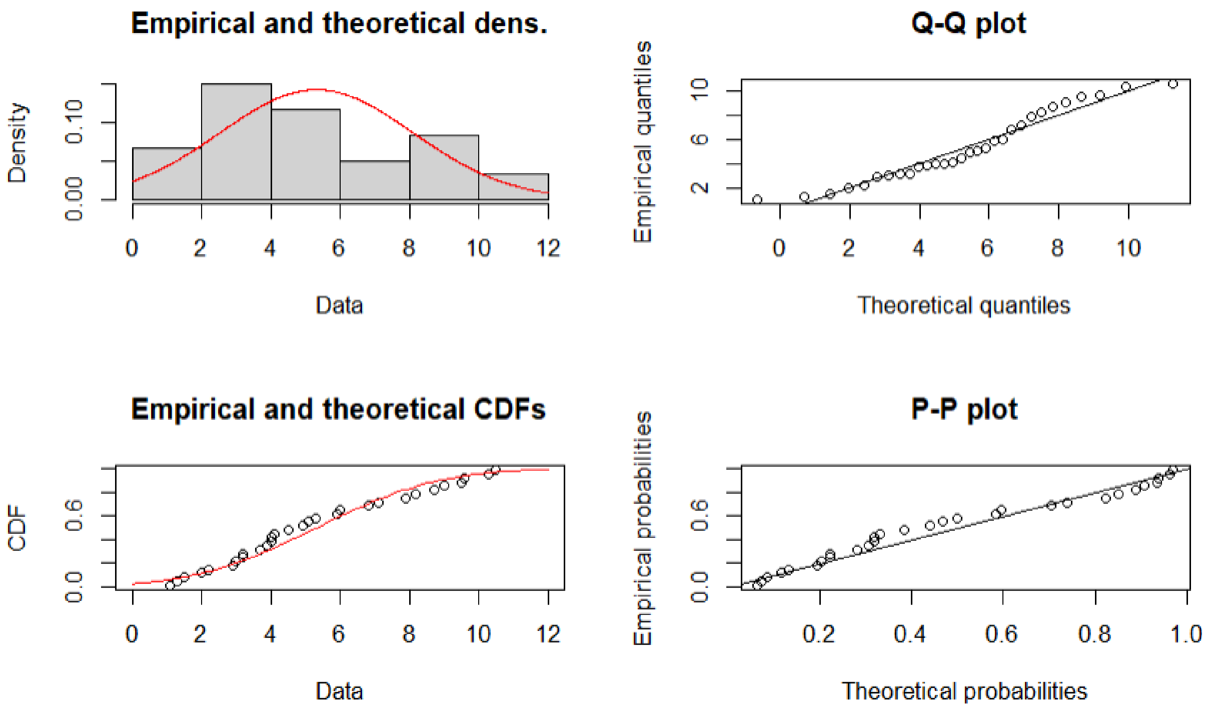


Figure 4.4. Normal Distribution Fit

From figures 4.3 and 4.4, we can see, graphically, from the Q-Q plot, that the fitted Weibull distribution fits our sample better. The Akaike Information Criterion of each model also proves this, with the Weibull's being around 147.15, and the Normal's AIC is around 150.70. Therefore, when using one distribution, we can use the fitted Weibull's.

The estimated parameters of the fitted Weibull Distribution are $k = 2.016509, \lambda = 6.012105$.

Next, we'll fit a mixture of Normal Distributions using the R package mixtools. Using the function mixtools::normalmixEM, we are able to fit a mixture of normal distributions. In our case, we fit a mixture of two normal distributions.

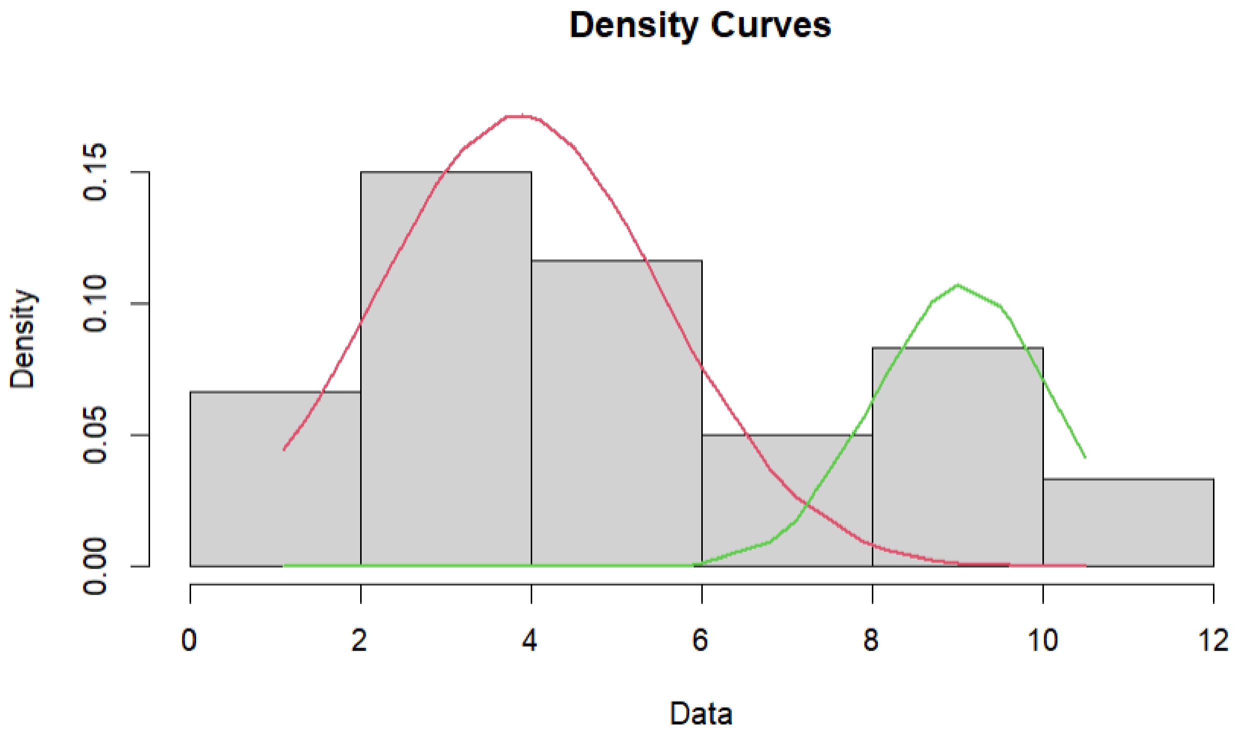


Figure 4.5. Density Curves of Fitted Mixture of Two Normal Distributions.

The fitted mixture has a log-likelihood of -69.4553, which is a promising statistic on the fit of the model.

The first normal distribution has an estimated proportion of 0.7210976 , with an estimated mean of 3.859548, and an estimated standard deviation of 1.674672.

The second normal distribution has an estimated proportion of 0.2789024, an estimated mean of 9.07207, and an estimated standard deviation of 1.034407.

Kolmogorov-Smirnov Test Using Monte Carlo Permutation Test of Fitted Weibull Distribution

In order to test whether or not our sample can be said to come from our fitted Weibull distribution, we conducted Kolmogorov-Smirnov test in conjunction with Monte Carlo Permutation Testing.

First we created a sample using our fitted distribution. A sample size of 30 should do, equal to our original sample. We then conducted the initial Kolmogorov-Smirnov test using R's ks.test function, with our original sample and the weibull samples as the inputs.

In our initial test, our null hypothesis, H_0, is that both samples come from the same distributions, while the alternative hypothesis, H_1, is that the compared samples come from different distributions. We used an alpha of 0.05 in making the statistical decision. Since the initial test yielded a test statistic of 0.1667, and a p-value of 0.80, we reject the alternative hypothesis.

However, given the small sample size, we can not be confident about how well our initial test reflects the real situation. We then carry on our Monte Carlo Permutation Testing to tackle this. Only slightly differing from our initial testing, our null hypothesis, H_0, is now that the distributions of the original sample (X) and our Fitted Weibull distribution (Y) are equal ($F_X = F_y$), while our alternative hypothesis, H_1, is the contrary.

In permutation testing, we "pool" our samples together, before creating two sample sets from the pooled group at random, with the sample sizes the same as the original sets. Let's call the new sample sets X* and Y*. If our null hypothesis is correct, i.e. $F_X = F_y$, then for all X* and Y*, $F_{X*} = F_{y*}$ should also be true. Using this, along with the Kolmogorov-Smirnov test, we can calculate the estimated achieved significance level for the comparison of the population distributions, using the formula: $\hat{p} = \frac{[1+\sum_{b=1}^{B} I(\hat{\theta}^{(b)} \geq \hat{\theta})]}{B+1}$, where B is the number of iterations conducted, and $\hat{\theta}$ is the test statistic of the initial test.

In code, we can do this by iteration. For each iteration, we chose the sample data for X* at random, and giving the remaining sample data points to Y*, before conducting a Kolmogorov-Smirnov test. The average of the indicator function, i.e. 1 if the iteration's test statistic is greater than or equal to the initial test, and 0 otherwise, to calculate the achieved significance level.

Using 10,000 iterations, the achieved significance level of the Monte Carlo Permutation Tests is 0.808, with most iterations yielding a test statistic close to or greater than the initial test statistic. We can then say that our samples of years of experience of engineers follows $weibull(k\ = 2.016509,\ \lambda =\ 6.012105\ )$.

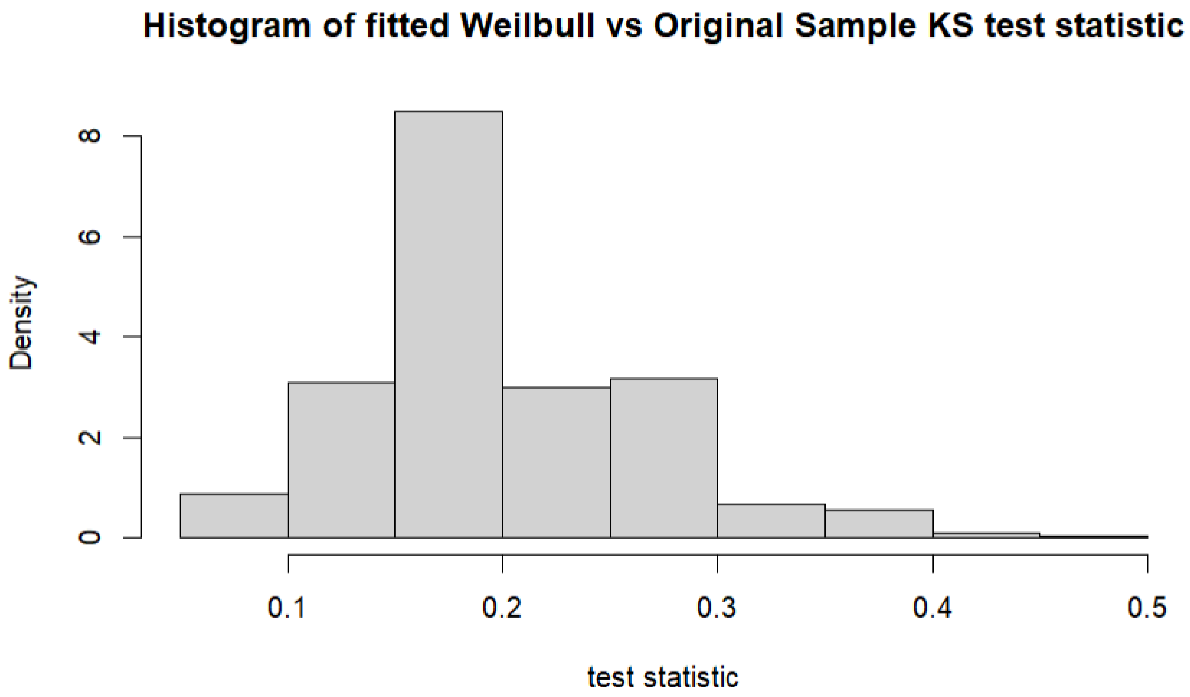


Figure 4.6 Histogram of test statistics of fitted Weibull vs original sample

Kolmogorov-Smirnov Test Using Monte Carlo Permutation Test of Fitted Mixture of Normal Distributions

Our process would follow the previous monte carlo permutation testing that we have done with the fitted Weibull distribution, in which we'll use the Kolmogorov-Smirnov test.

First, when creating our sample, we have created a sample of indices using R's sample function. The choices made in the sampling function have the probability given by the lambda attribute of the mixtools::normalmixEM function. The rnorm function, used for creating the samples of the mixed distribution, then gets the parameter using the mu and sigma attributes of the mixtools::normalmixEM, using the indices provided by the sample function.

We then conducted the initial Kolmogorov-Smirnov test, comparing the original sample and the sample from the mixed normal distribution, by using R's ks.test. An alpha of 0.05 was used, leading us to accept the null hypothesis H_0 = the original sample distribution and the mixed normal sample have equal distribution, due to a test-statistic of 0.1333333 and a p-value of 0.9560382.

We now proceed to the monte carlo permutation testing. Following the same algorithm, and again doing 10,000 iterations. The conducted permutation testing yielded an achieved significance level of 0.956, with most of the test statistics of the iterations being equal to or greater than the initial test statistic.

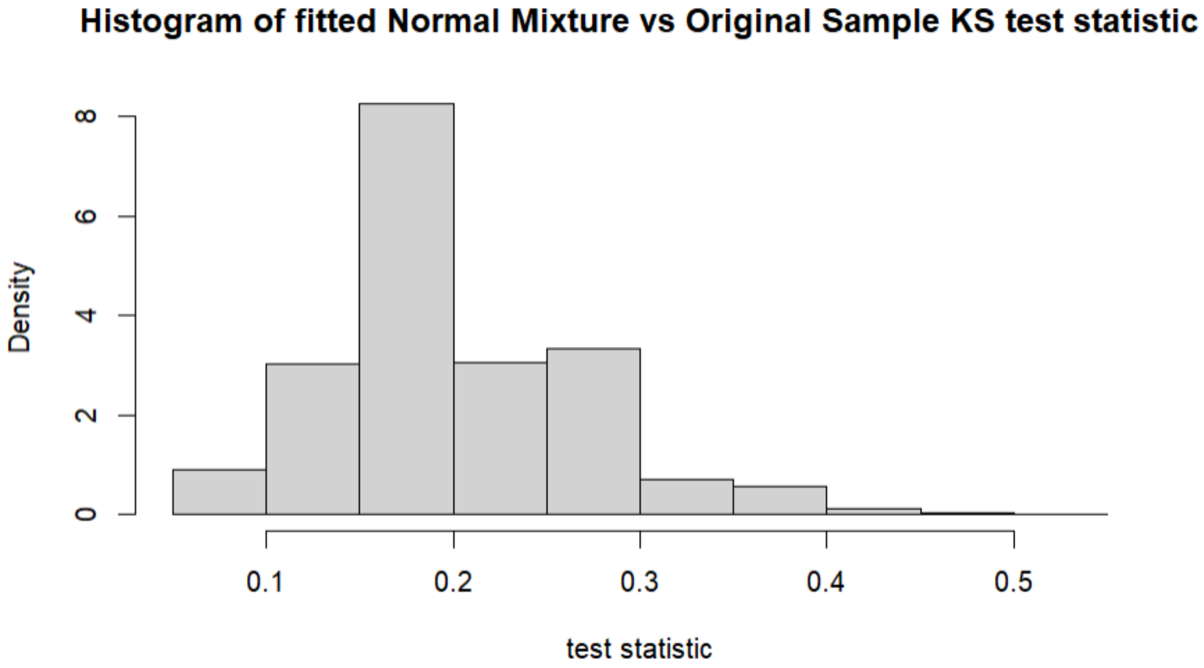


Figure 4.7 Histogram of test statistics of fitted Weibull vs original sample

Subsection Conclusion

Using Monte Carlo Permutation Testing, along with the Kolmogorov-Smirnov test, we were able to compare the sample distribution with two theoretical distributions, namely a weibull distribution and a mixture of two normal distributions.

Both of the fitted theoretical distributions had positive results when compared with the sample data. However, it is the mixture distribution that showed the most promising results, and is more reliably closer to the population distribution.

Markov Chain Monte Carlo of Bayesian Inference

Defining the Bayesian Model

In this chapter, we would be trying to predict the engineer's salary using their years of experience. In most regression models, this would be a simple linear model. However, even in the real world, engineers with the same years of experience would not have the same salary. Which is why we will be using bayesian linear regression.

Our model would be like so:

$$salary \sim N(\beta_0 + \beta_1 \cdot Years\ of\ Experience,\ \sigma^2)$$

Where β_0 is the estimated value of the intercept, i.e. the estimated starting salary for an engineer in the real world, and β_1 is the estimated value of slope, which is the assumed increase for each year of experience.

In crafting the MCMC Bayesian model, several weakly informative priors were assumed. This was decided upon assumptions on real life data regarding engineer salary in the United States, as well as available information within the sample data set. For example: $\beta_0 \sim N(\mu = sample\ salary\ mean,\ \sigma = \$15,000),\ \beta_1 \sim N(\mu = 0,\ \sigma = \$1,500)$ was used for the intercept and the slope, as indicated above.

The Metropolis-Hastings sampler was used to estimate the Bayesian linear regression model, due to the relative simplicity of the variables involved, not needing the complexity offered by the Gibbs sampler which are more effective for multivariate data.

For each iteration of the chain, the log ratio, calculated by subtracting the sum of the log likelihoods of the previous iteration from the log likelihood of the current iteration, was then compared from the logarithm of a random value $u \sim Unif(0, 1)$. For iterations where log(u) is less than the log ratio, the proposed values are accepted. Using this algorithm, 10,000 iterations were made, with 2,000 burn-in iterations.

In this manner, we achieved a 26.71% acceptance rate, close to the ideal of around 23%, showing that our chain was efficient. This is also shown by our trace plots below. Although we could somewhat see that they are spiking out, the middle remains dark, highlighting that the samples are centered around it.

Results of Bayesian Linear Regression by MCMC

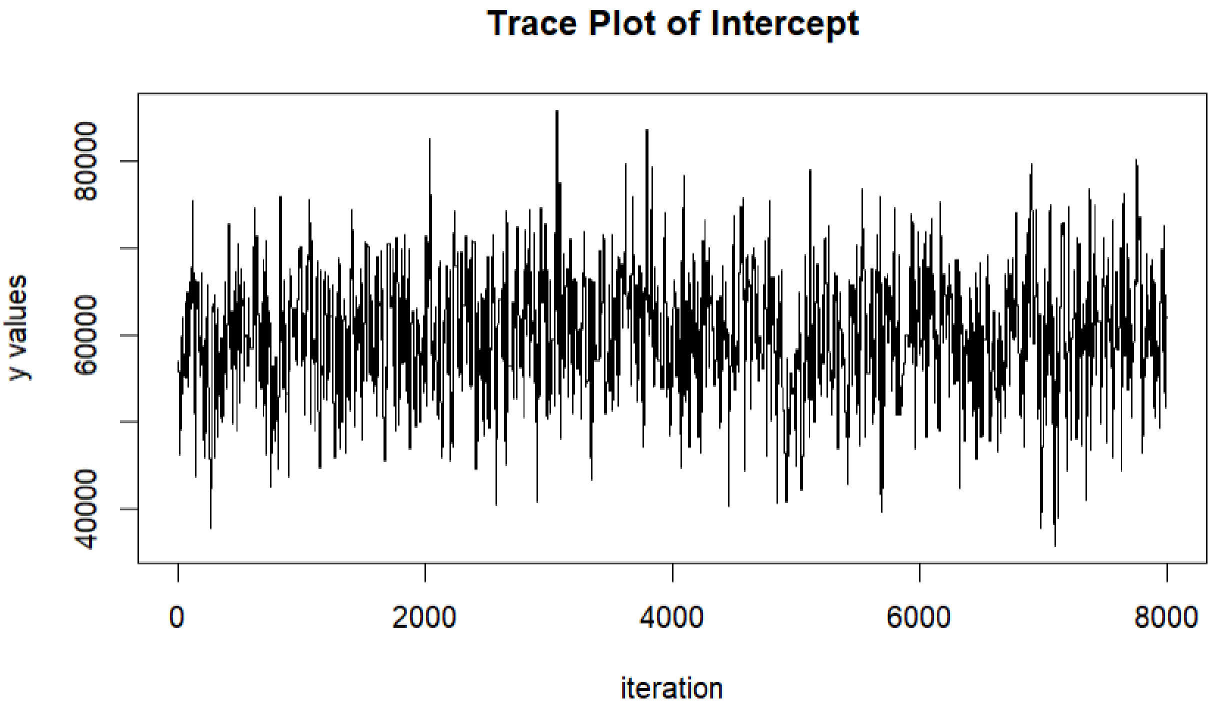


Figure 5.1. Trace Plot of Intercept

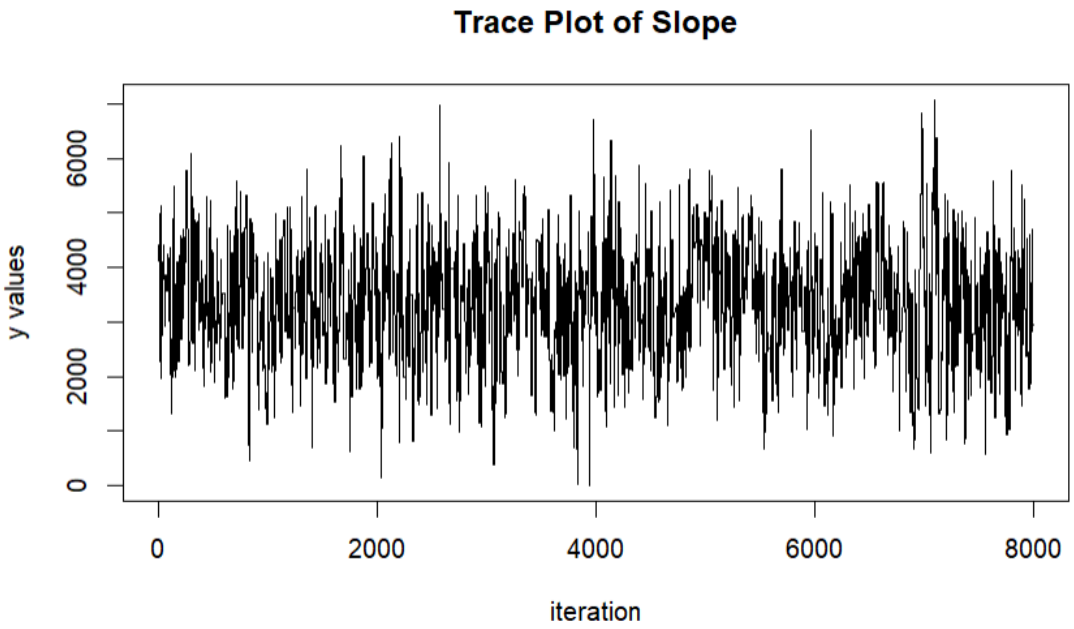


Figure 5.2 Trace Plot of Slope

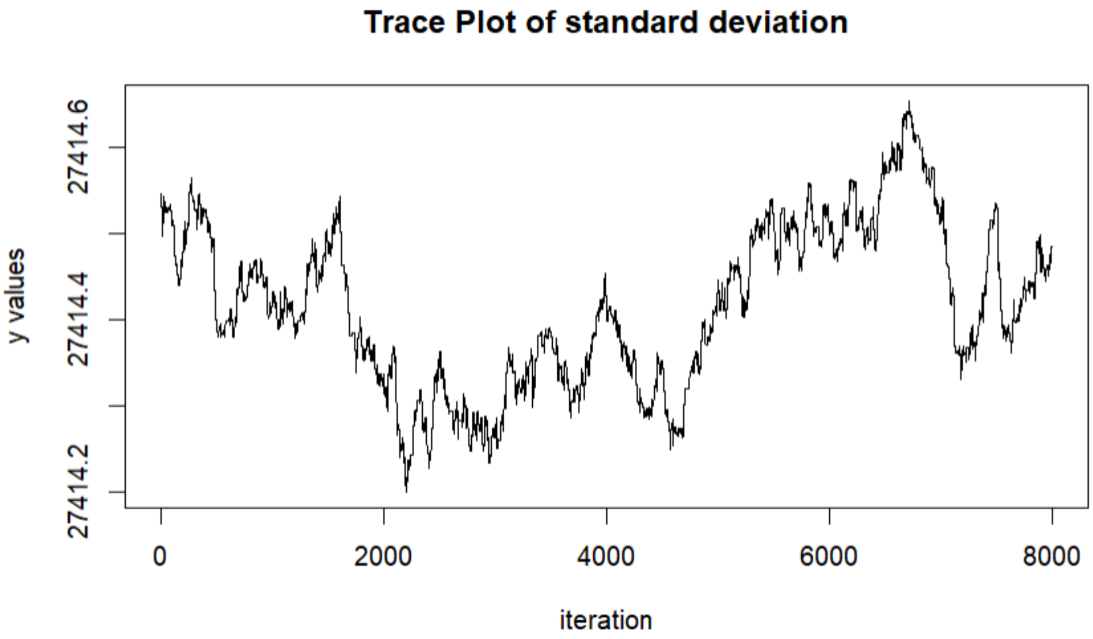


Figure 5.3. Trace Plot of standard deviation

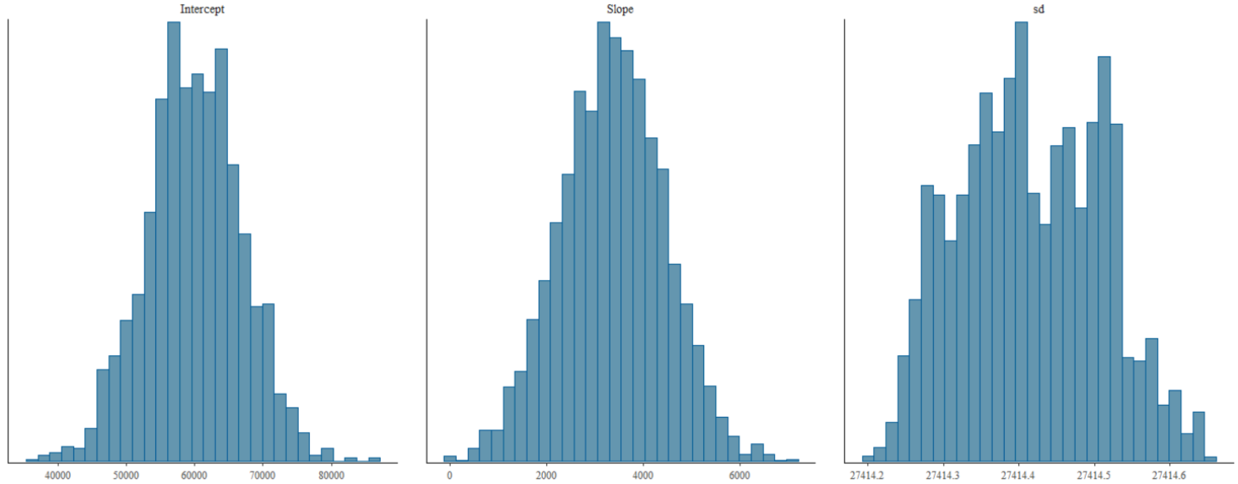


Figure 5.4. Univariate Marginal Posterior Distributions

With regards to the estimates themselves, the mean of the intercept was $59,968, with a naive standard error of 77.81. The mean of the slope was $3,352, with a naive standard error of 11.99. The standard deviation, which can be seen as error, had a mean of $27,414, and a naive standard error of $0.001.

The 95% percentile interval of the intercept is from $46,113 to $73,502; of the slope is from $1,253 to $5,378; and of the standard deviation is from $27,414 to $27,415.

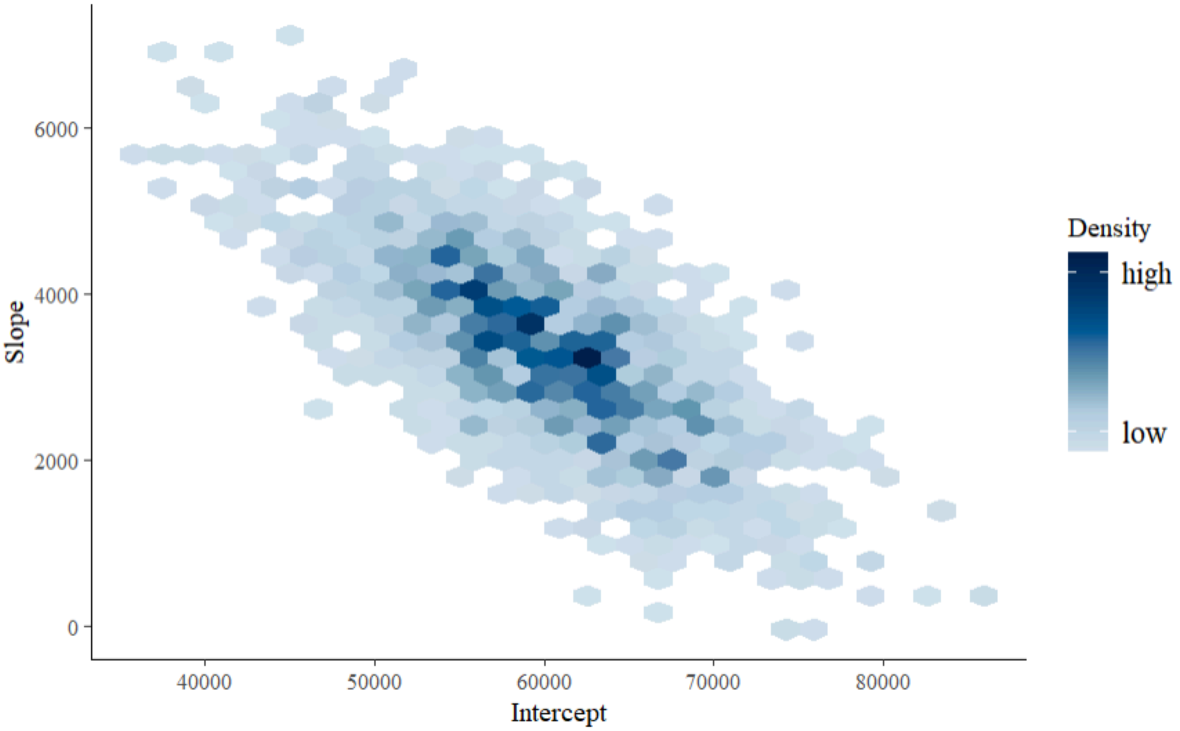


Figure 5.4. Bivariate Marginal Posterior Hexbin Plot

Figure 5.4 shows that the majority of the slope and intercept pairs are concentrated around $\beta_0 \in (\$50000\ ,\ \$70000)\ and\ \beta_1 \in (\$5000,\ \$1000)$. This shows that, although most of the intercept is concentrated along the value of 60,000, the slope is, at the same time, fluctuating over a wider spread.

Probability Density Estimation

As an initial exploration within the salary dataset, a histogram-based density plot of the salary variable is presented.

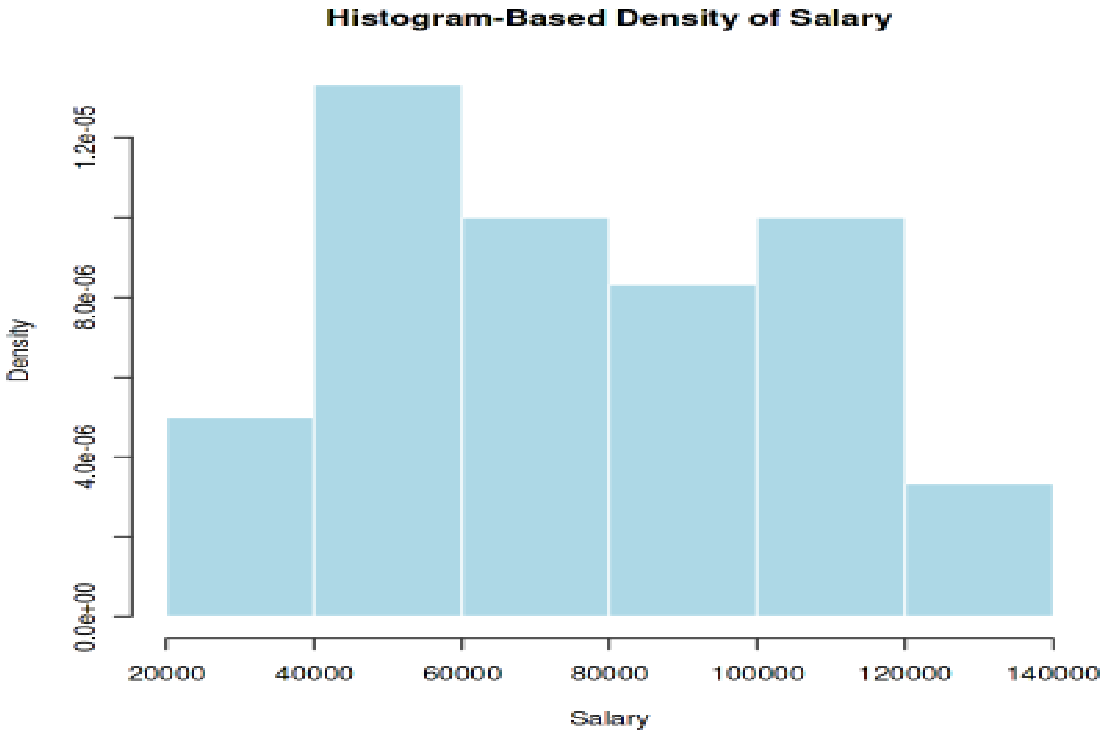


Figure 6.1. Histogram-Based Density of Salary

This histogram shows the distribution of the salary range (x-axis) into intervals and shows the frequency (density) of the given dataset within each bin. The visualization of the histogram shows the basic distribution of the salary while highlighting where the data is concentrated (peak).

In this dataset, the group wants to know what fits better in the given dataset. So the group uses the empirical (KDE) and theoretical distribution.

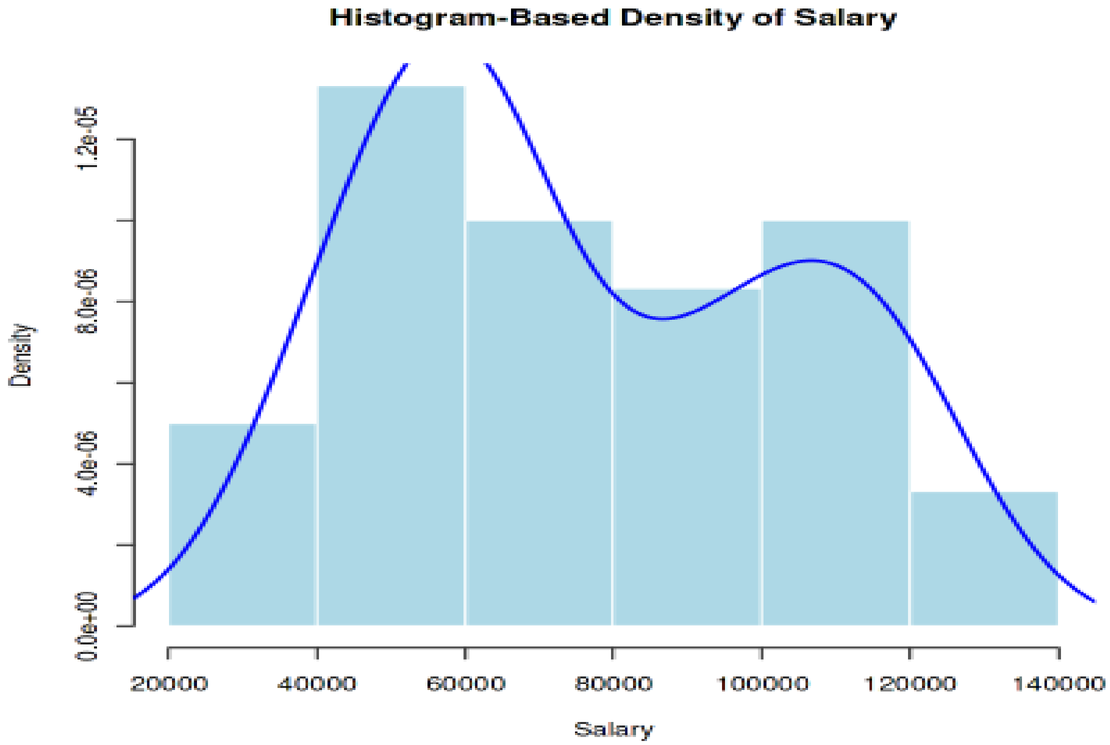


Figure 6.2. Empirical Distribution with Histogram

As the group wants to gain deeper insights about the salary distribution, given the initial histogram, a KDE (kernel density estimate) is applied to the histogram. As we can observe, the smooth blue line (KDE line) offers a non-parametric and smooth representation within the given dataset. The KDE line records variations and reveals a bimodal distribution because the line has two bumps: the first one is at around 60,000, and the second one is around 110,000, meaning there are two common salary ranges in the actual dataset. The visualization with applying KDE reflects the actual data without assuming any specific form of distribution.

Now, we will apply theoretical distribution into the initial histogram:

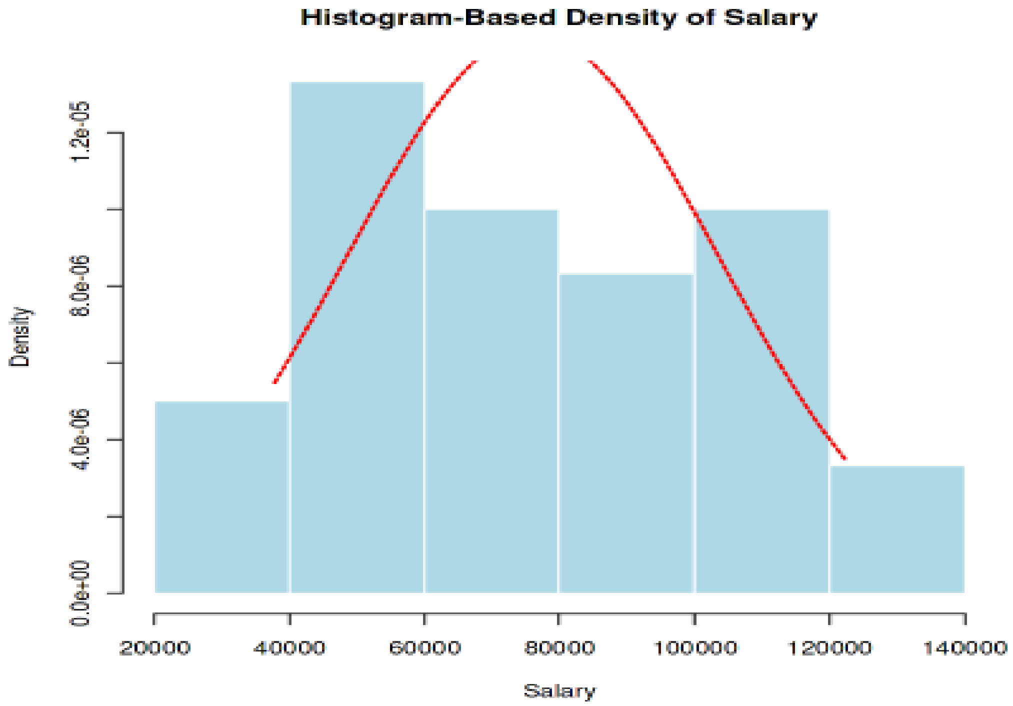


Figure 6.3. Theoretical Distribution with Histogram

Now, the group applies the theoretical distribution to the initial histogram. The curve line (theoretical line) is plotted to the initial histogram to assess if the salary dataset follows a normal distribution/pattern. The red line is computed by the mean and standard deviation of the dataset. Having the assumptions, we can see that the line is curved and symmetric indicating that the distribution is unimodal and assumes normality. However, as we can observe with the initial histogram, the assumption of theoretical distribution doesn't align with the shape of the actual salary dataset.

By applying both empirical (KDE) and theoretical distribution within the initial histogram of the dataset, we can now compare the empirical and theoretical side-by-side to evaluate the fit.

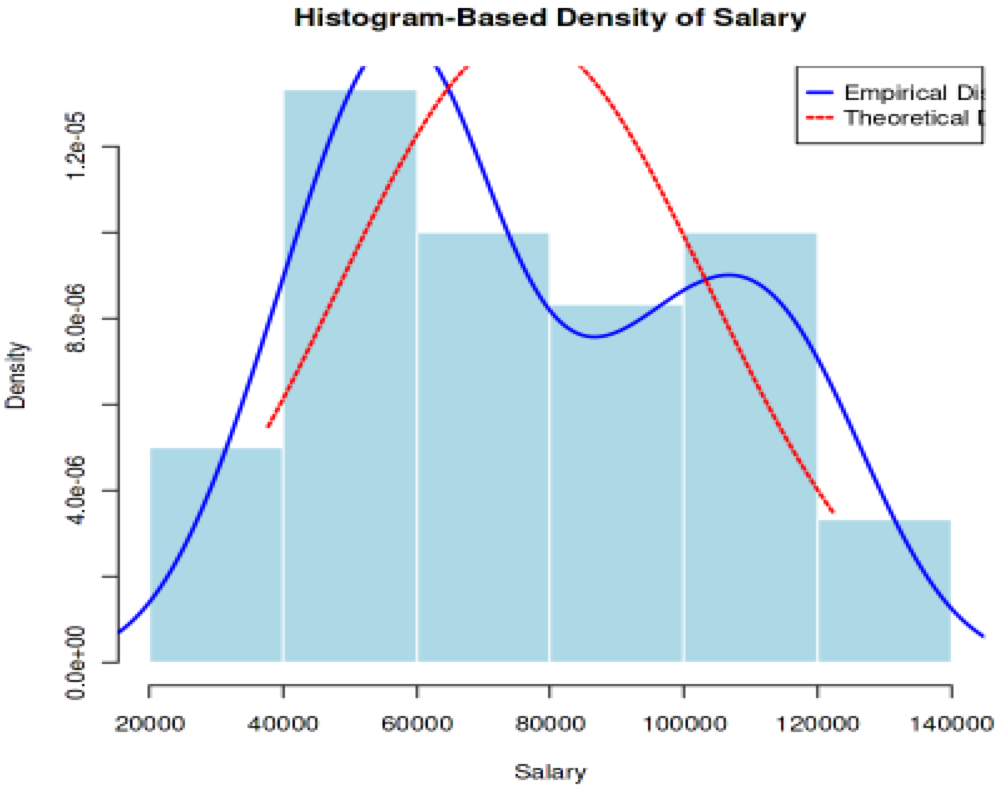


Figure 6.4. Comparison - Empirical vs. Theoretical Distribution

Finally, after applying both of the distributions to the initial histogram of our dataset. We now plot and curve both of the distributions to our histogram to show the direct comparison of the two. As we can observe, the empirical distribution (KDE) follows the actual data, capturing the irregularities and peaks of the data, while on the other hand, the theoretical distribution line was curved, symmetric, and smoother. This divergence indicates that the salary dataset does not follow a normal distribution, and it also indicates that the empirical distribution provides a more precise fit for understanding the actual structure of the given dataset.

Summary

The mean salary of engineers based on the data is $76,003 with SE of $5005.17, as we want to test whether this mean salary will hold, we used Bootstrap and Resampling method to resample the data and measure the mean salary based on resampled data. It was found out that Bootstrap resampled data computed a slightly lower mean, $75,935.04. It has a standard error of $4969.622 and the 95% CI [$66,288.51,$85,667.32]. While the Jackknife resampled data computed the same mean and standard error from the theoretical values with the 95% CI [$74416.39,$77282.42]. The result from both resampling methods shows that our theoretical mean holds as the mean value from computed resampled data are almost the same and it is in the range of CI from both methods, it just varies slightly from the Bootstrap method but still we can have our level of uncertainty that the theoretical mean will not fall off or spike up.

As we have resampled the data, we have now used two regression models to fit the Bootstrap resampling method and test what model would fit the best. The data that we have followed linearity and met the assumption for linear regressions. We then used "Salary" as our target variable and "YearsofExperience" as predictor. We then check the performance of our first model, linear regression model, lm(). It is evident that the model fits very well as explained by R- squared of 0.932 with no significant outliers based on the minimum and maximum residuals. The model computed the intercept as log 10.5066276 or exp(10.5066276) = $36556.99 with a slope of 0.1259256 or exp (0.1259256) = 1.13 - 1 x 100 = 13%. It has a bias of -0.0006935344 for Intercept and 0.0004288425 for slope with the standard error of 0.04 for intercept and 0 for slope. The result shows that the Bootstrap fitted to our model fits very well as the bias is very low, which we can say that our model did not become overwhelmed with our provided data. After doing the linear regression model, we also used one model called regression trees, wherein we trained and tested the dataset with the relationship of years of experience to the salary. Doing this will help the group in having a comparison to the first model on what's more accurate and precise in fitting the salary dataset. We first trained and split the data before bootstrapping. After doing this, we also did a bootstrap resampling with 5000 iterations, like we did in the first model. After applying the bootstrap model, we want to achieve the results in MAE, RMSE, and R-squared to compare it to the results in linear regression. After the results, we can observe that the results in regression trees are slightly higher than the linear regression; this indicates that the lower the value, the better the results. As the linear regression is much closer to 0 in RMSE and MAE, we can say that linear regression is much more accurate and precise than the regression trees. Aside from that, the linear regressions were also able to explain 93.2 percent of the data, which is higher than the 66.2 percent we got in regression trees. With this comparison that we got in RMSE, MAE, and R-squared, the group concluded that linear regression is more accurate, precise, and better for the salary dataset for fitting.

Looking at the distribution of the years of experience, we decided to add another dimension to the process: mixtures. We fit two distributions to the sample data of years of experience, and fit, first, a Weibull distribution (AIC = 147.15), and then a mixture of two normal distributions (log-likelihood = -69.4553). The fitted Weibull had the estimated parameters $weibull(k\ =\ 2.016509,\ \lambda =\ 6.012105\ )$. The fitted mixture distribution had the following parameters: The first normal distribution has an estimated proportion of 0.7210976 , with an estimated mean of 3.859548, and an estimated standard deviation of 1.674672. The second normal distribution has an estimated proportion of 0.2789024, an estimated mean of 9.07207, and an estimated standard deviation of 1.034407. After conducting Monte Carlo Permutation Testing, the fitted weibull distribution had an achieved significance level of 0.808, less than the achieved significance level of the mixture of the two normal distributions.

Bayesian modelling has a clear advantage in modelling: by attributing a small part of randomness that is anchored to parameters, the model can help describe and include in the model the randomness that occurs in real life. Predicting the engineer salary using the engineer's years of experience, the bayesian model followed the model: $salary \sim N(\beta_0\ +\ \beta_1 \cdot Years\ of\ Experience,\ \sigma^2)$. After 10,000 iterations of Metropolis-Hastings MCMC, the chain yielded an acceptance rate of 26.71%. The mean and the naive standard error of the sampled β_0 is shown to be 59,968 and 77.81, respectively. The respective figures are 3,352 and 11.99 for the slope; and 27,414 and 0.001 for the standard error. The posterior marginal distribution of these parameters $\beta_0,\ \beta_1,\ and\ \sigma$ are available at figure 5.4.

As the group wanted to gain more insights about the density and salary distribution of the salary dataset. The group explores and applies two different distributions in this dataset; the first one is the empirical distribution (KDE), and the second one is the theoretical distribution. With exploring the initial histogram of the salary dataset with these two distributions used, the group wanted to know what kind of distribution is more fit to the dataset. In applying empirical distribution, we can observe that the line in empirical distribution closely follows the shape of our actual data, since in empirical distribution (KDE) we computed the actual data; the line has two peaks, one around 60,000 and one at 110,000. On the other hand, applying a theoretical distribution doesn't compute the actual data in this dataset, as it only computes the mean and standard deviation of the dataset. So, as the theoretical distribution assumes that the dataset follows a normal distribution, the line resulted in being symmetric and smooth. However, in applying this to our dataset, we can say that empirical is more effective than the theoretical, since empirical computes the actual dataset and follows the shape and behavior of our initial histogram. And lastly, the behavior and shape of the actual salary is bimodal, which can be seen in applying empirical evidence since we have 2 peaks. Which cannot be satisfied by the theoretical since the curve in this distribution shows that the dataset is unimodal. In conclusion, empirical distribution is more fit and outperforms theoretical, which makes the empirical distribution more effective, precise, and accurate with our given dataset.