**Predicting Workers' Compensation Claims: An Analysis Using Linear Regression and Neural Networks**

Christopher King

STA4102 Methods for Applied Statistics

Dr. Houston Sanders

June 23, 2023

1. Abstract

In this study, I attempted to predict workers' compensation claims using multiple linear regression, log-transformed multiple linear regression, and neural networks based on several explanatory variables. The data consisted of 54,000 workers' compensation claims. After evaluating and comparing models, the Log-transformed linear regression model is found to be the best fit of the three models. This model considered age, hourly pay, gender, marital status, and whether the individual has dependents, along with an interaction between the term between gender and hourly pay, all of which had a statistically significant impact on the claim costs. These findings reveal questions that should be explored in further research, such as why these variables affect claim costs, and what additional factors might affect claim costs and be used to further increase the model accuracy.

1. Introduction

In the United States, there are nearly 5 million workers' compensation claims filed each year. Workers’ compensation is a critical component of employment, to ensure that both employees have financial security if they are hurt on the job and businesses do not get sued in the case of injury. Most often, workers comp insurance is provided to businesses by insurance companies, playing a critical role in keeping the economy stable. However, reliably predicting how much will be paid out in a Compensation Claim is a very difficult task. Without being able to predict the amount of workers’ Compensation paid, insurance companies are forced to raise their premiums and pass the cost down to the business and employees to hedge this unknown. Without taking such steps, Insurance companies would be assuming too much risk to stay in business for long. Therefore, the task of predicting Compensation claims is critical to keeping insurance companies in business, keeping prices at a competitive level, and preventing unreasonably high premiums from being passed on to businesses. For these reasons, the research questions this analysis seeks to answer is to understand the relationship between several explanatory variables and the Claim amount and attempt to predict the compensation claim based on these explanatory variables, including the employee’s age, gender, hourly pay, marital status, whether the patient has any dependents, the number of hours worked per week and day and the average number of hours worked per day. This assumes that a claim is filed and does not attempt to predict the likelihood of a claim being filed. Furthermore, this seeks to predict the claim amount before any injury has occurred. This may be useful in a variety of ways, such as helping insurance companies to more accurately forecast and mitigate risk, and helping insurance companies and businesses accurately understand the risk and potential cost of hiring a new employee.

1. Data Description
   1. Data Description

In this analysis, the dataset I used was obtained from Kaggle. This is “highly realistic synthetic data”, showing samples of 54,000 individuals, each with 14 explanatory variables after removing some useless variables. Some variables I created are the average hours worked per day, hourly pay, and the total number of dependents. The dataset consisted of both continuous and binary variables, with a continuous outcome variable. Some potential limitations of this are that the model may not be considering factors that could have a significant impact that an insurance company would typically have access to, such as the level of education, work history, injury history, etc. Variables also may not be affecting the outcome because of the change in that variable itself, but because of other trends that coincide with it. For example, hourly pay may correlate positively with Claim cost not because paying an employee makes them get a more serious injury, but because the more dangerous jobs are higher paying.

* 1. Data Cleaning

Data cleaning began with removing rows in which:

* The average working day is longer than 24 hours
* The number of hours worked is zero
* The gender and marital status are unknown.

Every column in the dataframe is also converted to numeric, to more easily work with the data.

Data is sorted into continuous and binary variables, with a total of 10 Continuous, 4 binaries Before variable selection and 2 continuous, 3 binary variables After variable selection.

Binary variables are encoded to 0 and 1

* Gender: Female = 0, Married = 1
* Marital status: Single = 0, Married = 1
* Has dependents: none = 0, one or more = 1

After data cleaning, 48,598 individuals are remaining in the study.

* 1. Data Exploration and Transformations

Descriptive statistics such as minimum, 1st quartile, median, mean, 3rd quartile, and maximum along with the Skew and Kurtosis of the Claim Cost and continuous variables. This also includes the “trimmed” claim cost, which removes the top 1% and bottom 1% of the data as well as the Log transformed claim cost. The positive skewness values indicate a longer right tail, and a high Kurtosis value indicates a higher peak in the data distribution. Here we can observe the Claim cost ranges from $122 to over 4 million, with a median of $3,104. From the high skew and kurtosis values we also see the data is likely not normally distributed and contains extreme outliers. For this reason, the claim cost is log-transformed to attempt to normalize the data and make it easier to work with. The Hourly pay also ranges widely, from 2 cents per hour up to $250 per hour as well.



The correlation matrix is used to view correlations between continuous explanatory variables as well as the Claim Cost. The results show there is a weak positive correlation between all variables, as well as the possibility of some collinearity between Age and Hourly Pay.

A picture containing text, screenshot, plot, line

Description automatically generated

3.3.1 Claim Cost

The bar plots below are used to visualize the distribution of the Costs. Here we see Costs are extremely concentrated below $10,000, as the outlier of ~$4M greatly distorts the graph.

A picture containing line, black and white

Description automatically generated

3.3.2 Log-transformed Claim Cost

Because the Claim Costs are not normally distributed and have extremely large outliers, the data is Log-transformed to stabilize the Cost to be more linear and reign in outliers. The log-transformed data can also be used in modeling; however, the results will need to be converted out of the log form to be useful. As we can see from the plots below, this data is much closer to a normal distribution and without any extreme outliers

A picture containing diagram, text, plot, screenshot

Description automatically generated

A diagram of a box plot

Description automatically generated with low confidence

3.3.3 Age vs Claim Cost

The zoomed-in scatterplot better represents the data because of extreme outliers, at the cost of excluding roughly 20% of the dataset. Here it is hard to draw solid conclusions as there is a large amount of data (over 48,000 individuals), including extreme outliers. However, one can roughly see that Age is favored in the 20-40 range, and there is a small positive correlation between the 2 variables.

A picture containing line, plot, screenshot

Description automatically generated

3.3.4 Hourly Pay vs Claim Cost

The zoomed-in scatterplot better represents the data because of extreme outliers, at the cost of excluding roughly 20% of the dataset. Here it is hard to draw solid conclusions as there is a large amount of data (over 48,000 individuals), including extreme outliers. However, one can roughly see that Hourly Pay is favored around $10-$20 per hour, and there is again a small positive correlation between the 2 variables.

A picture containing line, plot, text, diagram

Description automatically generated

3.3.5 Marital Status

The correlation between Marital Status is explored using a combination of box and whisker plots, along with Pearson Product Moment Correlation and Wilcox Rank sum tests. To do this, the data is split up into a married group and a single group. Based on the box and whisker plot shown below, we see the married group has a higher tail, upper quartile, and median.

A picture containing line, diagram, plot, rectangle

Description automatically generated

3.3.6 Gender vs Claim Cost

The correlation between Marital Status is explored using a combination of box and whisker plots, along with Pearson Product Moment Correlation and Wilcox Rank sum tests. To do this, the data is split up into a male group and a female group. Based on the box and whisker plot shown below, we see the groups have a similar tail, while the female group has a higher lower quartile, upper quartile, and median. This indicates that gender is playing a role in the claim cost and should be considered in the modeling.

A picture containing line, diagram, plot, rectangle

Description automatically generated

3.3.7 Has Dependents vs Claim Cost

The correlation between whether the individual has dependents is explored using a combination of box and whisker plots, along with Pearson Product Moment Correlation and Wilcox Rank sum tests. The Box and Whisker plot is used to compare Final Costs for Individuals with dependents vs Without dependents. Here we see there is some variation, and it appears that those with dependents have a greater median and upper quartile, with a shorter tail and less extreme outliers.

A picture containing diagram, line, screenshot, rectangle

Description automatically generated

1. Methodology
   1. Statistical techniques

A Pearson’s product-moment correlation is used to measure the if there is a linear relationship and the strength of that relationship with a coefficient ranging from -1 to 1. This is used to compare the claim costs to marital status, gender, and whether the individual has dependents. Because the variables are encoded so that single = 0 and married = 1, female = 0 and male = 1, and no dependents = 0 while one or more dependents = 1, a positive coefficient indicates that the variable encoded as 1 will have a higher claim cost than the variable encoded as 0. This is used to better understand the relationship between the binary variables and claim cost.

The Anderson-Darling Normality test is used to understand whether the data is normally distributed. This establishes a null hypothesis that the data is normal distribution, with an alternative hypothesis that is it not. This is applied to the outcome variable, claim costs to understand whether it is more appropriate to use a parametric test such as t-tests or non-parametric tests such as the Wilcoxon rank sum test.

A Wilcox rank sum test is a non-parametric test comparing 2 distributions. We use this, rather than a T-test because the claim cost is not normally distributed, which is shown through a combination of bar graphs and the Anderson Darling Normality test. The Wilcox rank sum test establishes a null hypothesis that there is no difference in the correlation of the 2 groups and an alternate hypothesis that there is some difference between the 2 groups. This is again used to compare how the binary variables (marital status, gender, and dependents) affect the claim cost.

* 1. Modeling

The multiple linear regression model assumes a linear relationship between the outcome and explanatory variables. In the form of Y = a + b1\*X1 + b2\*X2 + ... + bn\*Xn, where the X’s represent explanatory variables, the b’s represent a change in the outcome variable for each unit of change in its corresponding explanatory variable, and a represents the intercept, which is hypothetically what the outcome variable would be if every explanatory variable was zero. This makes several assumptions, including the normality of the outcome variable (that the outcome variable is normally distributed), homoscedasticity (that the variances of residuals are constant), and independence (that the residuals are independent).

The log-transformed multiple linear regression is the same as a typical multiple linear regression model, however, the outcome variable is subject to a log-transformation. This is done when the linear model is not a good fit, because the data is not normally distributed or has extreme outliers. A log transformation normalizes the data, bringing it much closer to a linear fit.

Neural Networks, inspired by biological neurons are a subset of machine learning models used to recognize patterns in data that is fed into the model. Neural Networks are made up of neurons/nodes, which take input values and apply a mathematical function to them to produce an output. These are organized into 1 or more layers, which is where the first input layer receives the data, which is passed to “hidden layers” where computations are applied, producing a final prediction. Connections between neurons are also made and given a Weighted Value, which is adjusted while training to improve predictions. Neurons can also shift their activation function left or right to better fit data. These activation functions are used to determine whether the neuron is used in the calculation by calculating how relevant the neuron's input would be to the final prediction. This is used when modeling the claims cost because of the large dataset available, and neural networks may handle outliers and unusual data distributions more effectively than a linear regression model would.

* 1. Model Validation

Residuals vs Fitted values plots are a scatterplot, showing the predicted values of a model and the residuals (difference between predicted and actual values). These are used for checking model fit, linearity, homoscedasticity, and outliers. Typically, the closer the residuals are to zero, the better the model fit is. Linearity can be checked by seeing whether the residuals are randomly dispersed along the horizontal axis. Homoscedasticity is measured by seeing whether the residuals increase or decrease as the predicted values increase. If this happens, the data is homoscedastic, which violates the assumptions of linear regression.

R-squared checks the proportion of variance in the outcome variable that can be attributed to the explanatory variables. This ranges from 0-1, where 0 means the explanatory variables play no role in the outcome, and 1 means the model can perfectly predict the outcome variable. This is not sufficient on its own however, as a high R-squared can also be attributed to overfitting.

RMSE measures the difference between predicted and actual values observed. The errors for each corresponding value are averaged to find the mean, which is the Mean Squared Error (MSE). The RMSE takes this and applies a square root, resulting in the RMSE. A lower RMSE indicates higher accuracy, and the true accuracy can be understood by comparing the MSE value to the typical outcome values.

Akaike Information Criterion (AIC) again measures the difference in model accuracy using the Kullback-Leibler, however, this also penalizes the number of variables used, which discourages overfitting. This can be represented as 2K – 2(log-likelihood) where K is the number of parameters and log-likelihood is the maximum log-likelihood (MLE) of the model.

Bayesian Information Criterion (BIC) is very similar to AIC; however, this has a stronger penalty on the number of parameters used. This can be represented as K(log( n)) – 2(ln(Log-Likelihood)) where n is the number of observations in the dataset, K is the number of parameters and log-likelihood is the maximum log-likelihood (MLE) of the model.

Olden is used to calculate the importance of each input variable with the output of the model. These can be positive or negative and are useful for understanding how the model is operating. For each input, the paths (connections) from that node to the output are found. The weights of each path’s connections are multiplied together, and the sum of these results in the Olden value.

1. Results:
   1. Statistical Analysis
      1. Pearson Product Moment Correlation

When comparing Single vs Married Individuals, The Pearson Product Moment Correlation indicates a very weak positive correlation based on the coefficient of 0.0437, implying marital status may play a minor role in modeling the claim cost. Because the data is encoded so that 0=single and 1=female, a positive correlation indicates that the married group would have higher claim costs than the single group.

A screenshot of a computer program

Description automatically generated with low confidence

When comparing female vs males, the coefficient of -0.0333 indicates a very weak negative correlation between Male and Female groups, implying Gender may play a minor role in modeling the Claim Cost, and should be included in the modeling portion. Because the genders are encoded as 0=female and 1=male, a negative correlation coefficient indicates females would have a slightly higher claim cost.

A screenshot of a computer code

Description automatically generated with low confidence

When comparing those with vs without dependents, we find a coefficient of 0.0428. This indicates a weak positive correlation, suggesting dependents may play a minor role in modeling claim cost. Because the genders are encoded as 0=female and 1=male, a negative correlation coefficient indicates females would have a slightly higher claim cost.

A screenshot of a computer code

Description automatically generated with low confidence

5.1.2 Anderson-Darling normality test

When running this test, we see the Claim cost is not normally distributed based on the extremely low p-value of 2.2e-16. Because of this, we will use the Wilcox rank sum test rather than a t-test when comparing the Claim Cost to binary variables

A black text on a white background

Description automatically generated with low confidence

5.1.3 Wilcox

When comparing the median value of Single vs Married Individuals, the Wilcox rank sum test produced a p-value of 2.2e-16, based on these results we can be very confident that the medians of married and single groups are different, and the variable is worth considering in modeling.

A picture containing text, font, receipt, white

Description automatically generated

When comparing the median value of those with vs without dependents, The Wilcox Rank Sum test is again used to compare the medians of the 2 groups. Because our P value is extremely small, the medians between each group must be different, and the HasDependents variable is worth considering in modeling.

A picture containing text, font, white, receipt

Description automatically generated

* 1. Modeling
     1. Multiple linear regression

From the exploratory and statistical analysis, we see that Age, Hourly Pay, Gender, Marital Status, and Has Dependents all played a statistically significant role in predicting Claims Costs. This is further supported by the linear model depicted below, showing that each term, including an interaction term between hourly pay and gender was significant, all with p-values below 0.001, while marital status and the interaction term have a p-value below 0.05. However, this model still is relatively weak, as it only accounts for about 2.15% of the variance in Claim costs with an R-squared value of 0.0215. The median residual value of -5,299 and residual standard error of 34,770 also indicate a high degree of inaccuracy. For these reasons, the next modeling attempt is a linear regression with a log transform, which will better handle extreme values.

A screenshot of a computer

Description automatically generated with medium confidence

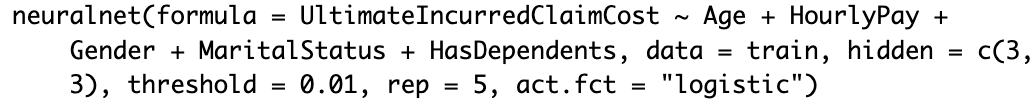
* + 1. Log-transformed linear regression

Based on the linear model depicted below, we see that applying a log transformation has significantly improved the fit. Now every explanatory variable is significant, with p-values well below 0.001. The residuals also appear to be normally distributed, and the residual standard error of 1.426 is significantly decreased. The model now accounts for roughly 12.19% of the Claim costs variance based on the R-squared value, which is a significant improvement, however, most of the variance remains unexplained, suggesting further research may need to be done to evaluate the relationship between different variables and Claim CostsA screenshot of a computer

Description automatically generated with medium confidence

* + 1. Neural network

This was created with the “neuralnet” package in R. The network includes a total of 6 neurons/nodes split up into 2 hidden layers, with a logistic activation function. The activation function takes the weighted sum of a neuron's inputs and calculates whether the neuron should be activated. The explanatory variables are Age, Hourly Pay, Gender, Marital Status, and Dependents. 5 training repetitions are performed, and the number of training repetitions is a sort of balancing act. With each repetition, the network becomes more accurate but also becomes more prone to overfitting. The threshold of 0.01 indicates that when the sum of squared errors reaches 0.01 the model will stop training, to prevent overfitting. The neural networks are a sort of “black box,” meaning they are harder to understand and interpret. However, in model evaluation, we can still find the RMSE, median residual, and residual standard deviation to understand the accuracy of the network.



A picture containing line, diagram, screenshot, circle

Description automatically generated

* 1. Model validation
     1. Residual vs fitted

Linear regression

The Residual vs Fitted values plot is useful for checking linearity, accuracy, outliers, and homoscedasticity. When checking the linear model, we see the variance is not constant and the data is homoscedastic because the residuals exhibit a sort of “funnel” shape. There are also many outliers, and the residuals overall indicate a generally inaccurate model.

A picture containing text, screenshot, line, diagram

Description automatically generated

Log-transformed linear regression

This plot can again be used to check the Log-transformed linear model for linearity, accuracy, outliers, and homoscedasticity. Here, we can see the spread does not exhibit any clear pattern or funnel shape, so the plot does not violate assumptions of linearity or homoscedasticity. However, there are some outliers that the model inaccurately predicted.

A picture containing text, screenshot, diagram, line

Description automatically generated

Neural network

When examining the Neural Networks, the Residuals vs Predicted values plot indicates that the network likely did not suffer from overfitting. The residuals are mostly concentrated along the x-axis, with some outliers once again. The plot does not indicate any heteroscedasticity, and because this is not a linear network, the assumption of linearity is not applicable.

A picture containing screenshot, sketch, text

Description automatically generated

* + 1. R^2, AIC, BIC, RMSE, Median Residual, Residual Standard Deviation



The above metrics are commonly used to measure the accuracy of a model.

The R-squared values for the linear model are extremely low, while the log-transformed model has a significantly improved, albeit still low R-squared value of 12.9%. Altering the linear model also greatly improved the AIC and BIC values, while the RMSE vastly improved to 1.6. Interestingly, the RMSE of the Neural Network is only marginally better than the Linear regression models. This indicates that both models were inaccurate in predicting claim costs, as the median claim cost is only around $3,104. The residual Median and Standard Deviation also indicate that the log-linear regression model is far more accurate than the linear model or neural network.

5.3.6 Olden

When analyzing the importance of each input in the neural network, the Olden algorithm can be used. From this, we see that the neural network attributed Age and Gender as the variables with the greatest effect on the claim cost. This varies from the linear model, which believed that Gender and Dependents were the most important variables.

A picture containing screenshot, diagram, plot, rectangle

Description automatically generated

1. Discussion/Interpretation of Results:

Because the linear regression model exhibited extremely high AIC, BIC, and RMSE values, along with a low R-squared value we know that this model did not accurately predict the data. This is because the Claim Costs were extremely skewed and did not have a normal distribution. For these reasons, the Log-transformed linear regression performed much better with low RMSE, AIC, and BIC values, and an improved R-squared value. The neural network surprisingly performed only marginally better than the linear regression model. This may be due to a small number of neurons and hidden layers, and only 5 training repetitions. Because of this, it is clear that the log-transformed linear regression model is the most reliable and accurate Model available. Some future work that could be done are further studies examining the effects of different variables on the claim cost because the log-transformed linear model only explains about 12% of the variance in claim costs, such as prior health and injury history, the specific job title, and amount of training given to employees. Further studies can also be done to examine why the variables such as age and hourly pay influence the claim costs, and whether these factors affect the cost directly or are proxy indicators of other factors such as the risk involved with the job. Businesses may also use these findings to identify what jobs have the highest claims, this likely correlates with the severity rate

1. Conclusion

This study attempted to predict workers' compensation claims, using a variety of statistical tests and modeling techniques. This proved difficult, due in part to Claim Costs which were highly skewed and contained extreme outliers. Of the models developed, the best-performing model is a log-transformed linear regression, including the following variables: age, hourly pay, gender, marital status, and has dependents (whether the individual has dependents) as well as an interaction term gender\*hourly pay. Applying a log transform on the claim costs proved to normalize the data and vastly improve the model. The neural network was also only marginally more accurate than the linear regression model, but this might be attributed to only 6 neurons and only 2 hidden layers, or only 5 training repetitions. Both linear models and neural networks found Gender to have the greatest impact on claim costs, however, the neural network believed dependents to be the least significant variable whereas the linear models considered this to be the second most important variable. Because this study is based only on a single dataset which is synthetic, more variables and real data are needed to make the results more generalizable. The low R-squared values also indicated that further research should be done to understand how different variables such as the amount of training employees get, injury history, and the specific job title affect the claim cost.

**References:**

Massaron, Luca. “Easy Peasy: It’s Lemon Squeezy.” *Kaggle*, 10 Nov. 2021, www.kaggle.com/datasets/lucamassaron/easy-peasy-its-lemon-squeezy.

“Home.” *RDocumentation*, www.rdocumentation.org/. Accessed 22 June 2023.

Parajuli, Paridhi. “Interpretation of Linear Regression.” *Medium*, 23 Sept. 2020, towardsdatascience.com/interpretation-of-linear-regression-dba45306e525.

Awan, Abid Ali. “Building Neural Network (NN) Models in R.” *DataCamp*, 6 Feb. 2023, www.datacamp.com/tutorial/neural-network-models-r.

*Posit*, 7 June 2023, posit.co/.

Final Project

Christopher King

2023-06-23

knitr::opts\_chunk$set(echo = TRUE)

raw\_data <- read.csv("ClaimData.csv")

original\_dataframe <- data.frame(raw\_data) #put the raw data in a dataframe

original\_dataframe$AvgWorkPerDay <- (original\_dataframe$HoursWorkedPerWeek / original\_dataframe$DaysWorkedPerWeek) #create a new column showing the average work per day  
original\_dataframe$TimeToReport <- difftime(original\_dataframe$DateReported, original\_dataframe$DateTimeOfAccident, units = "hours") #create a new column showing the time it took for incident to be reported  
original\_dataframe$HourlyPay <- (original\_dataframe$WeeklyWages / original\_dataframe$HoursWorkedPerWeek) #create a new column estimating hourly pay  
original\_dataframe$TotalDependents <- (original\_dataframe$DependentChildren + original\_dataframe$DependentsOther)  
original\_dataframe$HasDependents <- ifelse(test = original\_dataframe$TotalDependents > 0, yes = 1, no = 0)

original\_dataframe <- na.omit(original\_dataframe) #take away any rows containing NA's  
original\_dataframe <- subset(original\_dataframe, MaritalStatus != 'U') #take away all the rows where marital status is unknown  
original\_dataframe <- subset(original\_dataframe, MaritalStatus != '') #take away all rows where marital status is left blank  
original\_dataframe <- subset(original\_dataframe, HoursWorkedPerWeek < 167) #only keep rows where hours worked per week is less than 167  
original\_dataframe <- subset(original\_dataframe, HoursWorkedPerWeek > 0) #only keep rows where hours worked per week is greater than 0  
original\_dataframe <- subset(original\_dataframe, AvgWorkPerDay < 25) #only keep rows where hours per day is less than 25  
original\_dataframe <- subset(original\_dataframe, Gender != 'U') #remove rows where gender is unknown

df <- subset(original\_dataframe, select = c(UltimateIncurredClaimCost, Age, HourlyPay, Gender, MaritalStatus, HasDependents))

df$MaritalStatus[df$MaritalStatus == 'M'] <- 1  
df$MaritalStatus[df$MaritalStatus == 'S'] <- 0  
df$Gender[df$Gender == 'M'] <- 1  
df$Gender[df$Gender == 'F'] <- 0

df$Age <- as.numeric(df$Age)  
df$Gender <- as.numeric(df$Gender)  
df$MaritalStatus <- as.numeric(df$MaritalStatus)  
df$UltimateIncurredClaimCost <- as.numeric(df$UltimateIncurredClaimCost)  
df$HourlyPay <- as.numeric(df$HourlyPay)  
df$HasDependents <- as.numeric(df$HasDependents)

continuous\_variables <- subset(df, select = c(Age, UltimateIncurredClaimCost, HourlyPay))  
binary\_variables <- subset(df, select = c(Gender, MaritalStatus, HasDependents))

install.packages("fitdistrplus")  
install.packages("ggplot2")  
install.packages("PerformanceAnalytics")  
install.packages("corrplot")  
install.packages("nortest")  
install.packages("Metrics", dep = TRUE)  
install.packages("lmtest")  
install.packages('neuralnet', dependencies = T)  
install.packages('NeuralNetTools')  
library(fitdistrplus)  
library(ggplot2)  
library(PerformanceAnalytics)  
library(corrplot)  
library(nortest)  
library(Metrics)  
library(lmtest)  
library(neuralnet)  
library(NeuralNetTools)

###Correlation matrix, this can be plugged into chart.correlation  
correlation\_matrix <- cor(continuous\_variables, use = "everything");correlation\_matrix  
correlation\_chart <- chart.Correlation(continuous\_variables)

correlation\_Gender\_FinalCost <- cor.test(df$Gender, df$UltimateIncurredClaimCost, method = "pearson"); correlation\_Gender\_FinalCost

correlation\_MaritalStatus\_FinalCost <- cor.test(df$MaritalStatus, df$UltimateIncurredClaimCost, method = "pearson"); correlation\_MaritalStatus\_FinalCost

correlation\_Dependents\_FinalCost <- cor.test(df$HasDependents, df$UltimateIncurredClaimCost, method = "pearson"); correlation\_Dependents\_FinalCost

summary(df$UltimateIncurredClaimCost)

skewness(df$UltimateIncurredClaimCost)

kurtosis(df$UltimateIncurredClaimCost)

lower\_bound <- quantile(df$UltimateIncurredClaimCost, 0.01)  
upper\_bound <- quantile(df$UltimateIncurredClaimCost, 0.99)  
trimmed\_claim\_cost <- subset(df$UltimateIncurredClaimCost, df$UltimateIncurredClaimCost > lower\_bound & df$UltimateIncurredClaimCost < upper\_bound)  
summary(trimmed\_claim\_cost)

skewness(trimmed\_claim\_cost)

kurtosis(trimmed\_claim\_cost)

summary(log(df$UltimateIncurredClaimCost))

skewness(log(df$UltimateIncurredClaimCost))

kurtosis(log(df$UltimateIncurredClaimCost))

hist(log(df$UltimateIncurredClaimCost), breaks = 50, main = "Histogram of log(Claims Cost)", xlab = "log(Claims Cost)", col = "green")

barplot(df$UltimateIncurredClaimCost, col = "lightblue", xlab = "Insurance Claim", ylab = "Amount Paid", ylim = c(0,150000), main = "Barplot of Final Claim Cost")

barplot(df$UltimateIncurredClaimCost, col = "lightblue", xlab = "Insurance Claim", ylab = "Amount Paid", main = "Barplot of Final Claim Cost")

boxplot(log(df$UltimateIncurredClaimCost), main = "Boxplot of log(Claims Cost)", ylab = "log(Claims Cost)", range = 0, col = "lightblue", boxwex = 0.6)

plot(df$Age, df$UltimateIncurredClaimCost, main = "Age vs Final Claim Cost, excluding some outliers", xlab = "Age", ylab = "Final Claim Cost", ylim = c(0,15000), col = rgb(0,0,1,0.4))

plot(df$Age, df$UltimateIncurredClaimCost, main = "Age vs Final Claim Cost", xlab = "Age", ylab = "Final Claim Cost", col = rgb(0,0,1,0.4))

plot(df$HourlyPay, df$UltimateIncurredClaimCost, main = "Hourly Pay vs Final Claim Cost, excluding some outliers", xlab = "Hourly Pay", ylab = "Final Claim Cost", ylim = c(0,200000), col = rgb(0,0,1,0.4))

plot(df$HourlyPay, df$UltimateIncurredClaimCost, main = "Hourly Pay vs Final Claim Cost", xlab = "Hourly Pay", ylab = "Final Claim Cost", col = rgb(0,0,1,0.4))

boxplot(df$UltimateIncurredClaimCost ~ df$Gender,   
 xlab = "Final Claim Cost",   
 ylab = "Gender",  
 ylim = c(0,10000),  
 main = "Gender vs Final Claim Cost, 0 is Female and 1 is Male (zoomed in)",  
 range = 1,  
 boxwex = 0.6,  
 col = "lightblue")

boxplot(df$UltimateIncurredClaimCost ~ df$Gender,   
 xlab = "Final Claim Cost",   
 ylab = "Gender",  
 ylim = c(0,1000000),  
 main = "Gender vs Final Claim Cost, 0 is Female and 1 is Male (zoomed out)",  
 range = 1,  
 boxwex = 0.6,  
 col = "lightblue")

boxplot(df$UltimateIncurredClaimCost ~ df$MaritalStatus,   
 xlab = "Final Claim Cost",   
 ylab = "Marital Status",  
 ylim = c(0,10000),  
 boxwex = 0.6,  
 col = "lightblue",  
 main = "Marital Status vs Final Claim Cost, 0 is single and 1 is married (zoomed in)")

boxplot(df$UltimateIncurredClaimCost ~ df$MaritalStatus,   
 xlab = "Final Claim Cost",   
 ylab = "Marital Status",  
 ylim = c(0,1000000),  
 boxwex = 0.6,  
 col = "lightblue",  
 main = "Marital Status vs Final Claim Cost, 0 is single and 1 is married (zoomed out)")

boxplot(df$UltimateIncurredClaimCost ~ df$HasDependents,   
 xlab = "Final Claim Cost",   
 ylab = "Has Dependents",  
 ylim = c(0,20000),  
 boxwex = 0.6,  
 col = "lightblue",  
 main = "Dependents vs Final Claim Cost, 0 is no dependents and 1 is at least 1 dependent (zoomed in)",  
 range = 1)

boxplot(df$UltimateIncurredClaimCost ~ df$HasDependents,   
 xlab = "Final Claim Cost",   
 ylab = "Has Dependents",  
 ylim = c(0,1000000),  
 boxwex = 0.6,  
 col = "lightblue",  
 main = "Dependents vs Final Claim Cost, 0 is no dependents and 1 is at least 1 dependent (zoomed out)",  
 range = 1)

ad.test(df$UltimateIncurredClaimCost) #null is that its normally distributed. results: very low p value, its not normally distributed

with\_dependents <- df$UltimateIncurredClaimCost[df$HasDependents == 1]  
without\_dependents <- df$UltimateIncurredClaimCost[df$HasDependents == 0]  
wilcox.test(with\_dependents, without\_dependents)

married\_group <- df$UltimateIncurredClaimCost[df$MaritalStatus == 1]  
single\_group <- df$UltimateIncurredClaimCost[df$MaritalStatus == 0]  
wilcox.test(married\_group, single\_group)

male\_group <- df$UltimateIncurredClaimCost[df$Gender == 1]  
female\_group <-df$UltimateIncurredClaimCost[df$Gender == 0]  
wilcox.test(male\_group, female\_group)

shortened\_without\_dependents\_group <- sample(x = without\_dependents, size = length(with\_dependents), replace = FALSE)  
chisq.test(shortened\_without\_dependents\_group, with\_dependents)

#shortened\_single\_group <- sample(x = single\_group, size = length(married\_group), replace = FALSE)  
#chisq.test(shortened\_single\_group, married\_group)  
##this returns Error: vector memory exhausted (limit reached?)

shortened\_male\_group <- sample(x = male\_group, size = length(female\_group), replace = FALSE)  
chisq.test(shortened\_male\_group, female\_group)

indices <- sample(1:nrow(df), size = 0.7\*nrow(df), replace = FALSE) #this will randomly select 70% of the rows from df w/o replacement  
train <- df[indices, ] #this puts the random 70% of the df rows into a dataframe "train"  
test <- df[-indices, ] #this puts the remaning 30% of the df rows into a dataframe "test"

linear\_model <- lm(train$UltimateIncurredClaimCost ~ train$Age + train$HourlyPay + train$Gender + train$MaritalStatus + train$HasDependents + train$HourlyPay\*train$Gender, data = train)

summary(linear\_model)

AIC(linear\_model)

BIC(linear\_model)

predictions1 <- predict(linear\_model, newdata = test)  
RMSE1 <- rmse(test$UltimateIncurredClaimCost, predictions1);RMSE1

plot(predict(linear\_model12), residuals(linear\_model12),  
 xlab = "Predicted values",  
 ylab = "Residuals",  
 main = "Residuals vs Predicted values")  
abline(h=0, col='red')

residuals\_linear\_model <- residuals(linear\_model)  
if(length(residuals\_linear\_model) > 0) {  
 qqnorm(residuals\_linear\_model)  
 qqline(residuals\_linear\_model)  
} else {  
 print("Theres no residuals")  
}

log\_linear\_model <- lm(log(train$UltimateIncurredClaimCost) ~ (train$Age) + (train$HourlyPay) + (train$Gender) + (train$HasDependents) + (train$HourlyPay)\*train$Gender, data = (train))

summary(log\_linear\_model)

AIC(log\_linear\_model)

BIC(log\_linear\_model)

predictions2 <- predict(log\_linear\_model, newdata = (test))  
RMSE2 <- rmse(log(test$UltimateIncurredClaimCost), predictions2);RMSE2

residuals\_log\_linear\_model <- residuals(log\_linear\_model)  
if(length(residuals\_log\_linear\_model) > 0) {  
 qqnorm(residuals\_log\_linear\_model)  
 qqline(residuals\_log\_linear\_model)  
} else {  
 print("Theres no residuals")  
}

plot(predict(log\_linear\_model), residuals(log\_linear\_model),  
 xlab = "Predicted values",  
 ylab = "Residuals",  
 main = "Residuals vs Predicted values")  
abline(h=0, col='red')

skynet <- neuralnet(UltimateIncurredClaimCost ~ Age + HourlyPay + Gender + MaritalStatus + HasDependents, data = train, hidden = c(3,3), rep = 5, threshold = 0.01, act.fct = "logistic")

plotnet(skynet)

olden(skynet)

pred3 <- predict(skynet, test)  
RMSE3 <- rmse(test$UltimateIncurredClaimCost, pred3);RMSE3

skynet\_residuals <- test$UltimateIncurredClaimCost - pred3  
plot(skynet\_residuals)  
  
median(skynet\_residuals)  
sd(skynet\_residuals)