**Problem Description:** Health insurance companies need to make more money from the premiums they collect than they spend on medical care for their customers. To do this, they work hard to predict how much they'll need to pay for medical treatments. This is tricky because serious health conditions are rare and unpredictable. However, some conditions are more common in certain groups of people. For example, smokers are more likely to get lung cancer, and obese individuals may be more prone to heart disease.

The aim here is to analyze patient information to figure out the average medical expenses for different groups of people. This helps insurance companies decide how much to charge for premiums, based on the expected costs of treating different health conditions.

**Introduction of OLS Algorithm:** In Ordinary Least Squares (OLS) regression, we find the best values for α and β by minimizing the sum of the squared differences between our predicted y values and the actual ones. These differences are called residuals and they measure how far off our predictions are from reality. OLS regression aims to make the error between the actual y value and the predicted y value as small as possible.

It does this by adding up the squared differences between the actual and predicted y values for all the data points. The solution for a depends on the value of b which can be obtained from this formula:

If we simplify this equation and look at its different parts, we can make it simpler. The bottom part of the fraction for 'b' is quite similar to something we already know, which is the variance of 'x', usually written as Var(x).

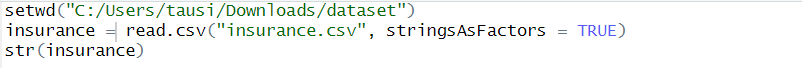
The top part of the equation includes adding up how much each data point differs from the average 'x' value, and then multiplying that difference by how much the same data point differs from the average 'y' value. This is kind of like the covariance calculation for 'x' and 'y'.

If we divide the variance function with the covariance we can get b.

OLS is widely used because it is easy to understand, computationally efficient, and often provides unbiased estimates of the coefficients when certain assumptions about the data are met. However, it has limitations, especially when the underlying relationship between the variables is not linear or when there is multicollinearity among the independent variables.

**Collecting Data:** For this study, we're using simulated data that represents medical costs for people in the United States. This information is based on numbers from the US Census Bureau but is not real. The insurance.csv file has information about 1,338 people who are currently part of an insurance plan. It includes details like age, gender, body mass index (BMI), number of children covered, smoking habits, and region of residence in the US. We'll analyze how these factors relate to the medical expenses charged to the insurance plan. For example, we might expect older people and smokers to have higher medical expenses. Unlike some other methods, in regression analysis, we need to decide how these factors relate to each other instead of the system figuring it out automatically.

**Preparing The Data:** We'll use the read.csv() function to bring in the data for analysis. It's okay to set stringsAsFactors = TRUE because we want to convert the three categorical variables into factors. We'll then use the str() function to double-check that the data is organized the way we want it to be.



A close-up of numbers

Description automatically generated

Before creating a regression model, it's useful to see if our dependent variable (expenses) follows a normal distribution. Even though linear regression doesn't absolutely need this, the model usually works better if it's the case.



A close up of numbers

Description automatically generated

Because the mean value is bigger than the median, this suggests that the distribution of insurance expenses is right-skewed. We can confirm this visually using a histogram.

A black and white text

Description automatically generated

A graph of a number of expenses

Description automatically generated with medium confidence

The graph shows that most people in our data have yearly medical expenses below $15,000, with a few having much higher costs. The graph also shows a right-skewed distribution. Although it's not great for a linear regression model, knowing this upfront can help us make a better model later on. But before we deal with that, there's another problem. Regression models need all features to be numbers, but we have three features that are categories, like sex (male or female) and smoker (yes or no). Also, the region feature has four different areas. We need to check how evenly the data is spread across these regions.

A black and blue text

Description automatically generated

A black and blue text

Description automatically generated

We can see that it's split pretty evenly among the four regions.

**Relationships Among Features:** Before we start building a regression model with our data, it's helpful to understand how the different factors we're looking at relate to each other and to the overall outcome. One way to do this is by using something called a correlation matrix. This matrix gives us a snapshot of how each factor is connected to the others and to our main focus. To make this matrix for the four numerical variables in our insurance data, we'll use a command called cor().

A number and numbers on a white background

Description automatically generated

In the correlation matrix, each row and column pair shows how the variables relate to each other. The diagonal always shows a perfect correlation of 1.0000000 because a variable is always perfectly correlated with itself. The correlations are symmetrical, meaning that the correlation between x and y is the same as the correlation between y and x. None of the correlations in the matrix are very strong, but there are some noticeable connections. For example, age and BMI seem to have a weak positive correlation, suggesting that as people get older, they tend to have higher body mass. Additionally, there's a moderate positive correlation between age and expenses, BMI and expenses, and the number of children and expenses. This suggests that as age, body mass, and the number of children increase, the expected cost of insurance also goes up.

**Visualizing Relationships Among Features:** To understand how different numerical features relate to each other, we can use scatterplots. Making a scatterplot for every possible combination of features could be overwhelming, especially with a lot of features. Instead, we can make something called a scatterplot matrix (SPLOM). It's like a grid of scatterplots, where each plot shows the relationship between two features. It helps us see patterns among three or more variables, even though each plot only looks at two features at a time. We'll use R's built-in graphical tools to make a scatterplot matrix for the four numeric features: age, BMI, children, and expenses. We'll use a function called pairs(). We'll use it with our insurance data frame, focusing on those four numeric variables.

A screenshot of a graph

Description automatically generated

In the scatterplot matrix, each row and column shows a scatterplot of the variables indicated by that pair. The plots above and below the diagonal are just mirror images because the x and y axes are swapped. While some plots look like random scattered points, a few show noticeable patterns. For example, the relationship between age and expenses shows several fairly straight lines, and the plot of BMI versus expenses has two clear groups of points. It's hard to spot trends in the other plots. To make the scatterplot matrix even more useful, we can add more information. We can do this with a function called pairs.panels() from the psych package.

A screenshot of a graph

Description automatically generated

Above the diagonal in the scatterplot matrix, we can see a correlation matrix instead of scatterplots. On the diagonal, there are histograms showing the distribution of values for each feature. Below the diagonal, the scatterplots have extra visual aids. In each scatterplot, we can notice an oval-shaped object called a correlation ellipse. It shows the strength of correlation between the two variables. The dot in the middle of the ellipse represents the average values for both variables. The more stretched out the ellipse, the stronger the correlation between the variables. If the ellipse is nearly round, like with BMI and children, it means there's a very weak correlation. We can also see a curve drawn on the scatterplots called a loess curve. This curve shows the general relationship between the two variables. For example, the curve for age and children looks like an upside-down U, peaking around middle age. This means that people around middle age tend to have more children on the insurance plan compared to younger or older people. This type of trend isn't apparent from just looking at the correlations. Another example is the curve for age and BMI, which is a gradually sloping line, showing that body mass tends to increase with age, confirming what we already saw in the correlation matrix.

**Training Model:** To fit a linear regression model to data with R, we can use the lm() function. This is included in the stats package.

A black text on a white background

Description automatically generated

A black and white text

Description automatically generated

Understanding regression coefficients is quite simple. The intercept is the expected expense when all independent variables are zero. But in real life, this often doesn't make sense because some variables can't be zero, like age or BMI. The beta coefficients show how much expenses are expected to increase for a one-unit increase in each feature, assuming everything else stays the same. For example, each year of age is associated with an average increase in medical expenses of $256.80, and each additional child means about $475.70 more in expenses each year. Similarly, for each unit increase in BMI, expenses go up by around $339.30. Even though we only included six features in our model, we see eight coefficients because of a method called dummy coding. This method turns categorical variables into numeric ones by creating binary variables, called dummy variables. For example, the "sex" variable becomes "sexmale" and "sexfemale". One category is left out as a reference. In our case, that's female non-smokers in the northeast region. So, the coefficients are compared to this reference group.

The results make sense: older age, smoking, and higher BMI are associated with higher medical expenses, while having more family members might mean more healthcare expenses. But we don't know yet how well the model fits the data.

**Model Performance:** The estimates we got from typing ins\_model show how the independent variables relate to the dependent variable, but they don't tell us how accurately our model fits the data. To check that, we can use the summary() command on the stored model.

A black and white text

Description automatically generated

A screenshot of a computer

Description automatically generated

The summary() output might seem complicated, but it's easy to understand once we know what to look for.

Residuals: These are the differences between the actual values and our model's predictions. Some of these differences are quite big. For example, the maximum error suggests our model underestimated expenses by almost $30,000 for at least one observation. However, most errors fall within a certain range.

P-values: These tell us the likelihood that each variable in our model is unrelated to the outcome. Smaller p-values mean the variable is likely important. Some p-values have stars (\*\*\*) to show they're statistically significant, meaning they're probably not just due to chance.

Multiple R-squared: This tells us how well our model explains the variation in the outcome. Our model explains about 75 percent of the variation in medical expenses, which is pretty good. Adjusted R-squared is also provided to help compare models with different numbers of variables.

Overall, our model seems to be doing well. While some errors are big, which is normal for medical expense data, our model explains a good portion of the variation. We might be able to make it even better with a few changes.

**Improving Model Performance:** One big difference between regression modeling and other machine learning methods is that in regression, the user has to decide which features to include and how to set up the model. This means if we know something about how a feature is connected to the result we're looking at, we can use that knowledge to make the model better. To account for a non-linear relationship, we can add a higher order term to the regression model, treating the model as a polynomial.

In linear regression, we assume that the relationship between a variable we're looking at and the result we're interested in is a straight line. But sometimes, this isn't true. For example, as people get older, medical expenses might not increase at the same rate. They might start to go up more quickly for very old people. To deal with this, we can make the model more complex by adding a term that involves squaring the variable we're interested in. This helps capture non-linear relationships. So instead of just looking at age, we also consider age squared.

Adding this squared term means we're estimating another coefficient. This coefficient helps us understand how the effect of age changes as people get older. To add this non-linear relationship to the model, we just need to include a new variable that represents age squared.

A black and blue text

Description automatically generated

So, to make our model better, we'll include both age and age squared in the lm() formula like this: expenses ~ age + age2. This way, the model can understand both the straight-line relationship and the curved relationship between age and medical expenses.

Let's say we suspect that BMI only affects medical expenses after a certain point, like when someone is obese with a BMI of 30 or higher. To model this, we'll create a new variable called "obesity indicator." If someone's BMI is 30 or more, the indicator will be 1; otherwise, it'll be 0. We can use the ifelse() function to do this. For each person's BMI, if it's 30 or higher, we'll assign a 1; otherwise, a 0.

Then, we'll include this new variable, bmi30, in our improved model. We can either replace the original BMI variable or include it alongside, depending on whether we think obesity has an effect in addition to just BMI. If we don't have a good reason to do otherwise, we'll include both in our final model.

A black and blue text

Description automatically generated

Sometimes, certain features might have a stronger impact on the outcome when they're combined than when they're looked at separately. For example, smoking and obesity might each individually affect medical expenses, but their combined effect could be worse than just adding up their individual effects.

When two features have this combined effect, it's called an interaction. To check if two variables interact, we can add their interaction to the model. We do this in R by using a formula like expenses ~ bmi30\*smoker.

This formula tells R to look at the impact of obesity (bmi30), smoking (smoker), and their interaction on medical expenses. The \* symbol is a shortcut that tells R to include both the individual variables and their interaction. So, bmi30\*smoker is the same as saying expenses ~ bmi30 + smoker + bmi30:smoker.

We'll use the lm() function again to build the model, but this time we'll include the new variables we created and also the interaction term.

A close up of text

Description automatically generated

Results:

A screenshot of a computer

Description automatically generated

The statistics from our model help us see if our changes made the regression model better. Compared to our first model, the R-squared value has gone up from 0.75 to around 0.87. This means our model now explains 87 percent of the variation in medical costs, which is a big improvement. The adjusted R-squared value, which considers the model's complexity, also improved from 0.75 to 0.87. Our theories about how the model works seem to be confirmed. The higher-order age term, age2, and the obesity indicator, bmi30, are both statistically significant, meaning they have a real impact on medical costs.

The interaction between obesity and smoking also shows a big effect. Not only do smokers have increased costs of over $13,404 compared to non-smokers, but obese smokers spend an extra $19,810 per year. This suggests that smoking makes health problems related to obesity even worse.

**Linear Regression Using Box-Cox Transformation:** Box-Cox Transformation is used to stabilize variance and make the data more normally distributed, which can improve the performance of linear regression models. After predicting the transformed dependent variable, it's necessary to reverse the transformation to obtain predictions for the original dependent variable, so the predictions are in the original scale of the data.

A white background with text

Description automatically generated with medium confidence

boxcox() function from the MASS package is used to find the optimal lambda value for the Box-Cox transformation. The formula ‘‘expenses ~ 1’’ specifies that we are transforming the expenses variable. bc\_trans$x contains a sequence of lambda values tested, and bc\_trans$y contains the corresponding likelihood values. lambda is set to the lambda value that maximizes the likelihood, which indicates the optimal transformation. If lambda is 0, a log transformation is applied; otherwise, the Box-Cox transformation formula (x^lambda - 1) / lambda is used to transform the expenses variable. Linear regression is performed using the transformed dependent variable (expenses\_transformed) and other independent variables (age, children, bmi, sex, smoker, region). The predict() function is used to predict the transformed dependent variable based on the linear regression model (ins\_model\_bc). If lambda is 0, indicating a log transformation, the exponential function exp() is used to reverse the transformation. Otherwise, the reverse of the Box-Cox transformation formula (lambda \* x + 1)^(1/lambda) is applied to obtain predictions for the original dependent variable.

**Results:**

A screenshot of a computer

Description automatically generated

A graph of a function

Description automatically generated

the model explains around 50% of the variability in the transformed expenses. Here, the F-statistic is significant, indicating that the model as a whole is significant. the model suggests that smoker is statistically significant at a 0.05 significance level, indicating that it has a significant effect on the transformed expenses. Other variables such as age, children, bmi, sex, and region do not appear to be statistically significant in this model.

**Gamma Regression:** Gamma regression is performed because the dependent variable is positively skewed and continuous, and its distribution is not normally distributed. Gamma regression is suitable for modeling non-negative continuous outcomes that have a skewed distribution, such as medical costs, insurance claims which are present in this dataset.

A screenshot of a computer program

Description automatically generated

This part of code is done just like the first regression.

A computer code with black text

Description automatically generated

glm() function is used to fit a Generalized Linear Model (GLM). “expenses ~ age + children + bmi + sex + smoker + region” specifies the formula where expenses is the dependent variable and age, children, bmi, sex, smoker, and region are the independent variables. data = insurance specifies the dataset to use for the regression analysis, which is the insurance dataset in this case. family = Gamma(link = "log") specifies the family of the GLM. Here, we're using the Gamma family of distributions with a logarithmic link function. The link = "log" argument specifies that the logarithmic link function will be used. This is a common choice for Gamma regression, as it ensures that predictions are on the scale of the original data (non-negative) and handles the positive skewness appropriately. gamma\_model = glm(expenses ~ age + children + bmi + sex + smoker + region, data = insurance, family = Gamma(link = "log")) specifies the fitting of a gamma regression model to predict expenses using age, children, bmi, sex, smoker, and region as predictors, with a logarithmic link function to appropriately model the relationship between the predictors and the dependent variable in a way that handles the positively skewed distribution of the expenses variable.

**Results:**

A screenshot of a computer

Description automatically generated

The intercept coefficient is estimated to be 7.39 (p < 0.001), indicating the expected value of medical expenses when all other predictors are zero. Age has a positive coefficient estimate of 0.029 (p < 0.001) which suggests that medical expenses tend to increase with age. Each additional child leads to an increase in medical expenses by 0.084 (p < 0.001). BMI is positively associated with medical expenses, with a coefficient estimate of 0.014 (p < 0.001). Being male is not found to be statistically significant in predicting medical expenses (p = 0.129). Smokers incur significantly higher medical expenses compared to non-smokers, with a coefficient estimate of 1.50 (p < 0.001).

The dispersion parameter for the Gamma family is 0.467, indicating the variability of medical expenses around the mean. The residual deviance is 337.75 on 1329 degrees of freedom, indicating a good fit of the model to the data. The AIC value is 26379, showing that the model adequately balances goodness of fit with model complexity.

**Comparison of Models:** Linear Regression Model: The residual standard error is 4445. The coefficients represent the expected change in expenses for a one-unit increase in the predictor variable. The model assumes linearity between the predictors and the dependent variable. Interaction terms are included in the model.

**Box-Cox Transformed Model:** The residual standard error is 1.721e-13. The model assumes linearity between the predictors and the transformed dependent variable.

**Gamma Regression Model:** The residual deviance is 337.75 with an AIC of 26379. The coefficients are interpreted as the log of the expected change in expenses for a one-unit increase in the predictor variable. The model assumes that the dependent variable (expenses) follows a gamma distribution.

The gamma regression model has the lowest AIC value, indicating better model fit compared to the other two models. However, it's important to note that the interpretation of coefficients in the gamma regression model is different due to the transformation applied to the dependent variable. The polynomial linear regression model has the highest R-squared value, suggesting that it explains the variability in expenses the best among the three models. Considering the purpose of the analysis and the assumptions of each model, the choice between models may vary. If interpretability of coefficients is crucial and the assumptions of linear regression are met, the polynomial linear regression model may be preferred. If model fit and handling of skewed data are paramount, the gamma regression model would be favored.