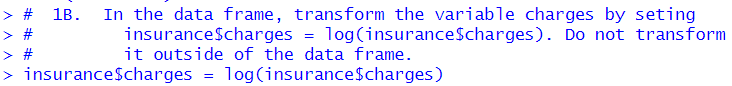
1. Data Preparation
   1. Load the dataset [insurance.csv](https://maryville.instructure.com/courses/71038/files/16328241/download?wrap=1)[Download insurance.csv](https://maryville.instructure.com/courses/71038/files/16328241/download?download_frd=1)into memory.

A screenshot of a computer program

Description automatically generated with low confidence

* 1. In the data frame, transform the variable **charges** by setting **insurance$charges = log(insurance$charges)**. Do not transform it outside of the data frame.



* 1. Using the data set from 1.b, use the **model.matrix()** function to create another data set that uses dummy variables in place of categorical variables. Verify that the first column only has ones (1) as values, and then discard the column only after verifying it has only ones as values.

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, screenshot

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, screenshot

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

* 1. Use the **sample()** function with set.seed equal to 1 to generate row indexes for your training and tests sets, with 2/3 of the row indexes for your training set and 1/3 for your test set. **Do not use any method other than the sample() function for splitting your data.**

A picture containing text, screenshot, font, electric blue

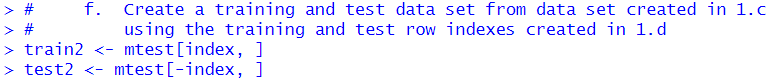
Description automatically generated

* 1. Create a training and test data set from the data set created in 1.b using the training and test row indexes created in 1.d. **Unless otherwise stated, only use the training and test data sets created in this step.**

A picture containing text, screenshot, font, line

Description automatically generated

* 1. Create a training and test data set from data set created in 1.c using the training and test row indexes created in 1.d

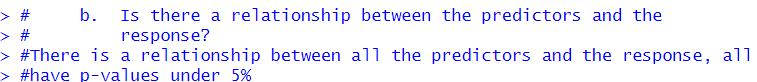


1. Build a multiple linear regression model.
   1. Perform multiple linear regression with **charges** as the response and the predictors are **age**, **sex**, **bmi**, **children**, **smoker**, and **region**. Print out the results using the summary() function. **Use the training data set created in step 1.e to train your model.**

A screenshot of a computer

Description automatically generated with medium confidence

* 1. Is there a relationship between the predictors and the response?



* 1. Does **sex** have a statistically significant relationship to the response?



Yes sex has a statistically significant relationship to the response.

* 1. Perform best subset selection using the stepAIC() function from the MASS library, choose best model based on AIC. For the "direction" parameter in the stepAIC() method, set direciton="backward"

A screenshot of a computer code

Description automatically generated with medium confidence

* 1. Compute the test error of the best model in #3d based on AIC using LOOCV using trainControl() and train() from the caret library. Report the MSE by squaring the reported RMSE.

A picture containing text, screenshot, font

Description automatically generated

* 1. Calculate the test error of the best model in #3d based on AIC using 10-fold Cross-Validation. Use train and trainControl from the caret library. Refer to model selected in #3d based on AIC. Report the MSE.

A picture containing text, screenshot, font

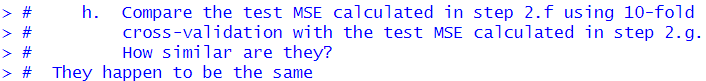
Description automatically generated

* 1. Calculate and report the test MSE using the best model from 2.d and test data set created in step 1.e.

A screenshot of a computer

Description automatically generated with medium confidence

* 1. Compare the test MSE calculated in step 2.f using 10-fold cross-validation with the test MSE calculated in step 2.g. How similar are they?



They happen to be the same.

1. Build a regression tree model.
   1. Build a regression tree model using function tree(), where **charges** is the response and the predictors are **age**, **sex**, **bmi**, **children**, **smoker**, and **region**.

A picture containing text, screenshot, font, electric blue

Description automatically generated

* 1. Find the optimal tree by using cross-validation and display the results in a graphic. Report the best size.

A picture containing text, font, screenshot, line

Description automatically generated

A picture containing text, diagram, rectangle, line

Description automatically generated

The best size is 4.

* 1. Justify  the number you picked for the optimal tree with regard to the principle of variance-bias trade-off.

I chose 4 because its about where the graph begins to level off, we want to balance between variance and bias. We could get less bias if we went with 5 or higher but we would end up getting too much variance at that point.

* 1. Prune the tree using the optinal size found in 3.b



* 1. Plot the best tree model and give labels.

A picture containing text, font, line, screenshot

Description automatically generated

A picture containing diagram, text, line, rectangle

Description automatically generated

* 1. Calculate the test MSE for the best model.

A close-up of a computer code

Description automatically generated with low confidence

1. Build a random forest model.
   1. Build a random forest model using function randomForest(), where **charges** is the response and the predictors are **age**, **sex**, **bmi**, **children**, **smoker**, and **region**.

A picture containing text, font, screenshot, line

Description automatically generated

* 1. Compute the test error using the test data set.

A picture containing text, font, screenshot, electric blue

Description automatically generated

* 1. Extract variable importance measure using the importance() function.

A screenshot of a computer

Description automatically generated with low confidence

* 1. Plot the variable importance using the function, varImpPlot(). Which are the top 3 important predictors in this model?

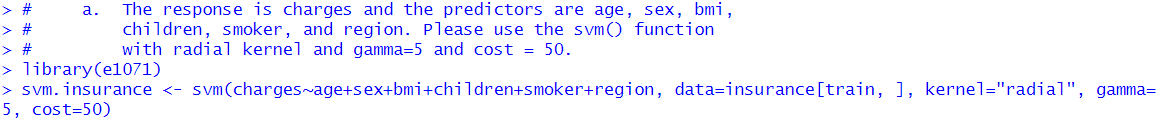


A picture containing text, diagram, number, parallel

Description automatically generated

The top three important predictors in this model are smoker, age, and children.

1. Build a support vector machine model
   1. The response is **charges** and the predictors are **age**, **sex**, **bmi**, **children**, **smoker**, and **region**. Please use the svm() function with radial kernel and gamma=5 and cost = 50.



* 1. Perform a grid search to find the best model with potential cost: 1, 10, 50, 100 and potential gamma: 1,3 and 5 and potential kernel: "linear","radial" and "sigmoid". And use the training set created in step 1.e.

**The first screenshot is what I tried to run, it wouldn’t work, even after running all night. Tried to add max iterations, still nothings so I ended up running it with just radial as an option for kernel.**

A blue text on a white background

Description automatically generated with low confidence



* 1. Print out the model results. What are the best model parameters?

A screenshot of a computer program

Description automatically generated with low confidence

* 1. Forecast **charges** using the test dataset and the best model found in c).



* 1. Compute the MSE (Mean Squared Error) on the test data.

A picture containing text, font, screenshot, line

Description automatically generated

1. Perform the k-means cluster analysis.
   1. Remove the **sex**, **smoker**, and **region,** since they are not numerical values.

A picture containing text, font, screenshot, electric blue

Description automatically generated

* 1. Determine the optimal number of clusters. Justify your answer. It may take longer running time since it uses a large dataset.

A picture containing text, font, screenshot, electric blue

Description automatically generated

* 1. Perform k-means clustering using the **3 clusters.**

A picture containing text, font, screenshot, electric blue

Description automatically generated

A picture containing text, diagram, line, screenshot

Description automatically generated

* 1. Visualize the clusters in different colors.

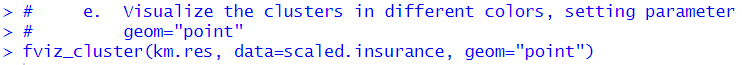
A picture containing text, font, electric blue, screenshot

Description automatically generated

A picture containing text, diagram, screenshot, plot

Description automatically generated

E.



1. Build a neural networks model.
   1. Remove the **sex**, **smoker**, and **region,** since they are not numerical values.

A picture containing text, screenshot, font, electric blue

Description automatically generated

* 1. Standardize the inputs using the scale() function.



A picture containing text, screenshot, diagram, line

Description automatically generated

* 1. Convert the standardized inputs to a data frame using the as.data.frame() function.



* 1. Split the dataset into a training set containing 80% of the original data and the test set containing the remaining 20%.

A close-up of blue text

Description automatically generated with low confidence

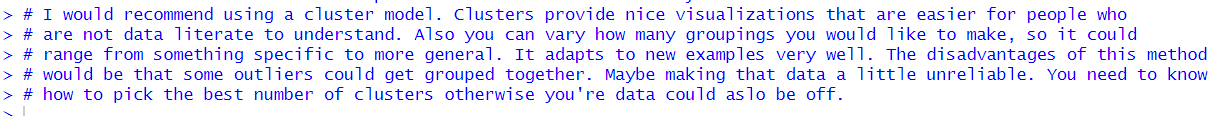
1. Putting it all together.
   1. For predicting insurance charges, your supervisor asks you to choose the best model among the multiple regression, regression tree, random forest, support vector machine, and neural network models. Compare the test MSEs of the models generated in steps 2.g, 3.f, 4.b, 5.e, and 7.d. Display the names for these types of these models, using these labels: **Multiple Linear Regression**, **Regression Tree**, **Random Forest**, **Support Vector Machine**, and **Neural Network** and their corresponding test MSEs in a data.frame. Label the column in your data frame with the labels as **Model.Type**, and label the column with the test MSEs as **Test.MSE** and round the data in this column to 4 decimal places. Present the formatted data to your supervisor and recommend which model is best and why.

A picture containing text, screenshot, font, line

Description automatically generated

Based on the table of MSEs for the variable models the choice of best model is the Random Forest. We want our result(MSE) to be as close to zero as possible and this one is clearly the closest.

* 1. Another supervisor from the sales department has requested your help to create a predictive model that his sales representatives can use to explain to clients what the potential costs could be for different kinds of customers, and they need an easy and visual way of explaining it. What model would you recommend, and what are the benefits and disadvantages of your recommended model compared to other models?



* 1. The supervisor from the sales department likes your regression tree model. But she says that the salespeople say the numbers in it are way too low and suggests that maybe the numbers on the leaf nodes predicting charges are log transformations of the actual charges. You realize that in step 1.b of this project that you had indeed transformed charges using the log function. And now you realize that you need to reverse the transformation in your final output. The solution you have is to reverse the log transformation of the variables in the regression tree model you created and redisplay the result.  
       
     ***Follow these steps:***
     1. Copy your pruned tree model to a new variable.



* + 1. In your new variable, find the data.frame named "frame" and reverse the log transformation on the data.frame column **yval** using the **exp()** function. (If the copy of your pruned tree model is named **copy\_of\_my\_pruned\_tree**, then the data frame is accessed as **copy\_of\_my\_pruned\_tree$frame**, and it works just like a normal data frame.).

A screenshot of a computer code

Description automatically generated with medium confidence

* + 1. After you reverse the log transform on the **yval** column, then replot the tree with labels.

A picture containing text, font, screenshot, line

Description automatically generated