PU5558-Machine Learning in Healthcare

Student ID - 52102609

Contents

Introduction

The answers to the assessment questions are within the template sections, any additional work completed to help work out the answers can be found in the appendix. I have created a machine learning model for my answer to question 1.

Load necessary packages

```
library(tidyverse)    # for general data science
library(tidymodels)   # for machine learning
library(corrplot)     # for visualising correlation matrices
library(vip)          # for variable importance plots
library(randomForest) # for the random forest model engine
```

Load chosen dataset

The data chosen for the machine learning task was the Knee Replacement data set. Information about the variables contained within the data set can be found in the document Patient Reported Outcome Measures in England Data Dictionary version 3.4.

```
knee_data <- read_csv("Knee Replacement CCG 2021 (2).csv")

glimpse(knee_data) #provides a general overview of the data
```

```
## Rows: 5,422
## Columns: 81
## $ ‘Provider Code‘                                    <chr> "00C", "00C", "00C"~
## $ Procedure                                          <chr> "Knee Replacement",~
## $ ‘Revision Flag‘                                    <dbl> 0, 0, 0, 0, 0, 0, 0~
## $ Year                                               <chr> "2020/21", "2020/21~
## $ ‘Age Band‘                                         <chr> "*", "*", "*", "*",~
## $ Gender                                             <chr> "*", "*", "*", "*",~
## $ ‘Pre-Op Q Assisted‘                                <dbl> 2, 2, 2, 2, 2, 2, 2~
## $ ‘Pre-Op Q Assisted By‘                             <dbl> 0, 0, 0, 0, 0, 0, 0~
## $ ‘Pre-Op Q Symptom Period‘                          <dbl> 2, 4, 4, 2, 4, 2, 2~
## $ ‘Pre-Op Q Previous Surgery‘                        <dbl> 2, 1, 2, 2, 2, 2, 2~
## $ ‘Pre-Op Q Living Arrangements‘                     <dbl> 2, 2, 1, 1, 1, 1, 1~
## $ ‘Pre-Op Q Disability‘                              <dbl> 1, 2, 2, 2, 1, 2, 2~
## $ ‘Heart Disease‘                                    <dbl> 9, 9, 9, 9, 9, 1, 9~
## $ ‘High Bp‘                                          <dbl> 1, 9, 9, 1, 1, 1, 9~
## $ Stroke                                             <dbl> 9, 9, 9, 9, 9, 9, 9~
## $ Circulation                                        <dbl> 9, 9, 9, 9, 9, 9, 9~
## $ ‘Lung Disease‘                                     <dbl> 9, 9, 9, 1, 9, 9, 9~
## $ Diabetes                                           <dbl> 9, 9, 9, 1, 9, 9, 9~
## $ ‘Kidney Disease‘                                   <dbl> 9, 9, 9, 9, 9, 9, 9~
## $ ‘Nervous System‘                                   <dbl> 9, 9, 9, 9, 9, 9, 9~
## $ ‘Liver Disease‘                                    <dbl> 9, 9, 9, 9, 9, 9, 9~
## $ Cancer                                             <dbl> 9, 9, 9, 9, 9, 9, 9~
## $ Depression                                         <dbl> 9, 9, 9, 9, 9, 9, 9~
## $ Arthritis                                          <dbl> 9, 1, 1, 1, 1, 1, 1~
## $ ‘Pre-Op Q Mobility‘                                <dbl> 2, 2, 2, 2, 2, 2, 2~
## $ ‘Pre-Op Q Self-Care‘                               <dbl> 1, 1, 1, 1, 1, 1, 1~
## $ ‘Pre-Op Q Activity‘                                <dbl> 3, 2, 3, 2, 1, 2, 2~
## $ ‘Pre-Op Q Discomfort‘                              <dbl> 2, 2, 3, 2, 1, 2, 2~
## $ ‘Pre-Op Q Anxiety‘                                 <dbl> 1, 1, 2, 2, 1, 2, 1~
## $ ‘Pre-Op Q EQ5D Index Profile‘                      <dbl> 21321, 21221, 21332~
## $ ‘Pre-Op Q EQ5D Index‘                              <dbl> 0.364, 0.691, 0.030~
## $ ‘Post-Op Q Assisted‘                               <dbl> 2, 2, 2, 2, 2, 2, 2~
## $ ‘Post-Op Q Assisted By‘                            <dbl> 9, 9, 9, 9, 9, 9, 9~
## $ ‘Post-Op Q Living Arrangements‘                    <dbl> 2, 2, 1, 1, 1, 1, 1~
## $ ‘Post-Op Q Disability‘                             <dbl> 1, 2, 2, 2, 2, 1, 2~
## $ ‘Post-Op Q Mobility‘                               <dbl> 1, 1, 1, 1, 1, 2, 1~
## $ ‘Post-Op Q Self-Care‘                              <dbl> 2, 1, 1, 1, 1, 2, 1~
## $ ‘Post-Op Q Activity‘                               <dbl> 1, 1, 1, 1, 1, 2, 1~
## $ ‘Post-Op Q Discomfort‘                             <dbl> 1, 1, 1, 1, 1, 2, 2~
## $ ‘Post-Op Q Anxiety‘                                <dbl> 2, 1, 1, 1, 1, 1, 1~
## $ ‘Post-Op Q Satisfaction‘                           <dbl> 2, 3, 1, 2, 1, 4, 3~
## $ ‘Post-Op Q Sucess‘                                 <dbl> 1, 1, 1, 1, 1, 4, 1~
## $ ‘Post-Op Q Allergy‘                                <dbl> 2, 2, 2, 2, 2, 2, 2~
## $ ‘Post-Op Q Bleeding‘                               <dbl> 2, 2, 2, 2, 1, 2, 2~
## $ ‘Post-Op Q Wound‘                                  <dbl> 1, 2, 2, 2, 1, 2, 2~
## $ ‘Post-Op Q Urine‘                                  <dbl> 2, 2, 2, 2, 2, 2, 2~
## $ ‘Post-Op Q Further Surgery‘                        <dbl> 2, 2, 2, 2, 2, 2, 2~
## $ ‘Post-Op Q Readmitted‘                             <dbl> 2, 2, 2, 2, 2, 2, 2~
## $ ‘Post-Op Q EQ5D Index Profile‘                     <dbl> 12112, 11111, 11111~
## $ ‘Post-Op Q EQ5D Index‘                             <dbl> 0.744, 1.000, 1.000~
## $ ‘Knee Replacement EQ 5D Index Post-Op Q Predicted‘ <dbl> 0.6621424, 0.740953~
## $ ‘Pre-Op Q EQ VAS‘                                  <dbl> 85, 50, 46, 60, 60,~
## $ ‘Post-Op Q EQ VAS‘                                 <dbl> 60, 100, 90, 95, 90~
## $ ‘Knee Replacement EQ VAS_Post-Op Q Predicted‘      <dbl> 75.63073, 66.38606,~
## $ ‘Knee Replacement Pre-Op Q Pain‘                   <dbl> 1, 1, 0, 0, 1, 1, 1~
## $ ‘Knee Replacement Pre-Op Q Night Pain‘             <dbl> 1, 2, 2, 2, 2, 3, 0~
## $ ‘Knee Replacement Pre-Op Q Washing‘                <dbl> 3, 4, 3, 4, 3, 3, 3~
## $ ‘Knee Replacement Pre-Op Q Transport‘              <dbl> 1, 4, 2, 4, 2, 2, 2~
## $ ‘Knee Replacement Pre-Op Q Walking‘                <dbl> 3, 3, 2, 3, 3, 1, 2~
## $ ‘Knee Replacement Pre-Op Q Standing‘               <dbl> 3, 2, 2, 3, 3, 2, 2~
## $ ‘Knee Replacement Pre-Op Q Limping‘                <dbl> 0, 3, 0, 0, 3, 1, 1~
## $ ‘Knee Replacement Pre-Op Q Kneeling‘               <dbl> 0, 0, 0, 2, 1, 1, 2~
## $ ‘Knee Replacement Pre-Op Q Work‘                   <dbl> 2, 2, 1, 3, 2, 1, 2~
## $ ‘Knee Replacement Pre-Op Q Confidence‘             <dbl> 3, 3, 3, 2, 1, 1, 2~
## $ ‘Knee Replacement Pre-Op Q Shopping‘               <dbl> 2, 4, 0, 2, 3, 0, 2~
## $ ‘Knee Replacement Pre-Op Q Stairs‘                 <dbl> 1, 4, 2, 2, 3, 2, 2~
## $ ‘Knee Replacement Pre-Op Q Score‘                  <dbl> 20, 32, 17, 27, 27,~
## $ ‘Knee Replacement Post-Op Q Pain‘                  <dbl> 4, 3, 3, 4, 3, 1, 2~
## $ ‘Knee Replacement Post-Op Q Night Pain‘            <dbl> 4, 4, 3, 4, 4, 3, 2~
## $ ‘Knee Replacement Post-Op Q Washing‘               <dbl> 3, 4, 4, 4, 4, 1, 4~
## $ ‘Knee Replacement Post-Op Q Transport‘             <dbl> 2, 4, 4, 4, 4, 3, 4~
## $ ‘Knee Replacement Post-Op Q Walking‘               <dbl> 4, 4, 4, 4, 4, 1, 4~
## $ ‘Knee Replacement Post-Op Q Standing‘              <dbl> 3, 4, 4, 4, 4, 1, 3~
## $ ‘Knee Replacement Post-Op Q Limping‘               <dbl> 3, 4, 4, 4, 4, 1, 4~
## $ ‘Knee Replacement Post-Op Q Kneeling‘              <dbl> 0, 4, 0, 3, 2, 0, 2~
## $ ‘Knee Replacement Post-Op Q Work‘                  <dbl> 3, 3, 4, 4, 4, 1, 4~
## $ ‘Knee Replacement Post-Op Q Confidence‘            <dbl> 4, 4, 4, 4, 4, 1, 4~
## $ ‘Knee Replacement Post-Op Q Shopping‘              <dbl> 4, 4, 4, 4, 4, 0, 4~
## $ ‘Knee Replacement Post-Op Q Stairs‘                <dbl> 3, 4, 4, 4, 3, 1, 4~
## $ ‘Knee Replacement Post-Op Q Score‘                 <dbl> 37, 46, 42, 47, 44,~
## $ ‘Knee Replacement OKS Post-Op Q Predicted‘         <dbl> 34.02619, 36.88715,~
```

```r
knee_data_col<-knee_data%>%
  select(`Pre-Op Q EQ5D Index`,`Pre-Op Q EQ VAS`,`Knee Replacement Pre-Op Q Pain`:`Knee Replacement Pre-
  #chosen from the available data source.Note, all the variables from Knee
  #Replacement Pre-Op Q Pain to Knee Replacement Pre-Op Q Score are included.

  drop_na() %>%     # remove rows with missing values
  unique() # keep unique row
```

```r
glimpse(knee_data_col) #provides a general overview of the processed data
```

```
## Rows: 4,974
## Columns: 16
## $ ‘Pre-Op Q EQ5D Index‘                  <dbl> 0.364, 0.691, 0.030, 0.620, 0.8~
## $ ‘Pre-Op Q EQ VAS‘                      <dbl> 85, 50, 46, 60, 60, 60, 90, 76,~
## $ ‘Knee Replacement Pre-Op Q Pain‘       <dbl> 1, 1, 0, 0, 1, 1, 1, 2, 0, 0, 1~
## $ ‘Knee Replacement Pre-Op Q Night Pain‘ <dbl> 1, 2, 2, 2, 2, 3, 0, 2, 0, 0, 2~
## $ ‘Knee Replacement Pre-Op Q Washing‘    <dbl> 3, 4, 3, 4, 3, 3, 3, 4, 2, 2, 2~
## $ ‘Knee Replacement Pre-Op Q Transport‘  <dbl> 1, 4, 2, 4, 2, 2, 2, 2, 2, 2, 2~
## $ ‘Knee Replacement Pre-Op Q Walking‘    <dbl> 3, 3, 2, 3, 3, 1, 2, 4, 2, 1, 1~
## $ ‘Knee Replacement Pre-Op Q Standing‘   <dbl> 3, 2, 2, 3, 3, 2, 2, 2, 0, 1, 0~
## $ ‘Knee Replacement Pre-Op Q Limping‘    <dbl> 0, 3, 0, 0, 3, 1, 1, 1, 0, 0, 0~
## $ ‘Knee Replacement Pre-Op Q Kneeling‘   <dbl> 0, 0, 0, 2, 1, 1, 2, 1, 1, 0, 1~
## $ ‘Knee Replacement Pre-Op Q Work‘       <dbl> 2, 2, 1, 3, 2, 1, 2, 2, 0, 1, 0~
## $ ‘Knee Replacement Pre-Op Q Confidence‘ <dbl> 3, 3, 3, 2, 1, 1, 2, 3, 0, 1, 0~
## $ ‘Knee Replacement Pre-Op Q Shopping‘   <dbl> 2, 4, 0, 2, 3, 0, 2, 4, 1, 1, 0~
## $ ‘Knee Replacement Pre-Op Q Stairs‘     <dbl> 1, 4, 2, 2, 3, 2, 2, 4, 1, 0, 1~
## $ ‘Knee Replacement Pre-Op Q Score‘      <dbl> 20, 32, 17, 27, 27, 18, 21, 31,~
## $ ‘Post-Op Q EQ5D Index‘                <dbl> 0.744, 1.000, 1.000, 1.000, 1.0~
```

Dataset description

This assessment will use the Knee Replacement CCG 2021 data set. Only double (quantitative) variables within the data set were kept and character variables were omitted. I chose variables that would be available before the operation. Only using the Post-Op EQ5D Index to train the model. There are 81 variables in the original data set, I selected 16 from the 81 variables. The data must also be clear of missing or incorrect values. The more accurate the data set the better functioning of the model.

Outcome Variable

- Post-Op Q EQ5D Index This is the variable we are trying to predict using information available from before the operation. This is a value calculated after the operation. The EQ5D Index is a number ranging from -0.594 to 1. The lower the score the worse the patient reports. The number is generated from a combination of answers from 5 topics; mobility, self care, usual activities, pain and discomfort and anxiety and depression.

Chosen potential predictor variables

- Pre-Op EQ5D Index This variable is calculated the same way as in the Post-Op EQ5D Index but the questions are asked before the knee replacement.
- Pre-Op Q EQ VAS The EQ VAS score is a number stated by the participant and is between 0 and 100. The lower the number the worse health of the patient.
- Knee Replacement Pre-Op Q Pain…consecutive variables to… Knee Replacement Pre-Op Q Score All of the variables from Knee Replacement Pre-Op Q Pain to Knee Replacement Pre-Op Q Score range from 0-4, 0 is the worst scenario and 4 is the best scenario. Note the missing values are “9”, the data should have been checked and these observations removed. Knee Replacement Pre-Op Q Score ranges from 0 to 48. The higher the score the better the outcome.

Suitable machine learning algorithm for three questions:

When choosing a model there are many factors to consider; types and availability of data, sample size, accuracy of the model, time available to train and develop and the interpretability of the model. There are 4974 rows of data in our dataset so that doesn't limit from the choices of models taught in the unit. The time to train and the model accuracy don't have to be considered in this project. Interpretability will be discussed in the limitations section.

1. Before the operation, can we estimate the post-operative EQ5D index for a patient?

The type of machine learning will be supervised learning. This is because we have data for the outcome variable of interest to train the algorithm, Post-Op Q EQ5D. The output variable is a numerical continuous variable. Whether the outcome variable is numerical or categorical determines the type of supervised learning algorithm. Regression is used when the outcome variable is numeric. There are various types of regression algorithms that can be selected for this problem. It is usually best to select the simplest algorithm that can solve the problem rather than an overly complicated algorithm which only provides a small increase in accuracy.

If there is a linear relationship between the outcome variable and the predictors then you can use linear regression. However analysis of the data shown in appendix 1 shows there is no linearity between the outcome variable and selected predictors. Therefore linear regression was not chosen and random forest in regression mode was selected. Also shown in appendix 1 is the outcome variable data is skewed to the right. If skewed data was used in the linear regression it could result in an error in the estimated confidence intervals of the feature weights values. It would be recommended to transform the outcome variable, see appendix 1. However, random forest doesn't require transformation of the outcome variable. Random forest is the algorithm chosen to answer this question.

2. Before the operation, can we predict how much pain a patient will have after the operation?

This question can also be solved by using supervised learning. We have data for the outcome variable, Post-Op Q Pain. This variable has discrete answers 0-4 and each of the values represents a category. Therefore the type of algorithm will be classification. There are different classification algorithms that could be chosen for this problem.

Logistic regression would not be suitable for this problem because the algorithm requires binary output. In this case there would be 5 categories. Support Vector Machine (SVM) is an algorithm suitable for classification supervised learning. Specifically it would be radial basis function support vector machines. This uses a non-linear hyperplane of the data for multiple categories.

3. Before the operation, can we calculate how many patients have had previous surgery?

I believe that this question doesn't require a machine learning algorithm to answer the question. The variable "Pre-Op Q Previous Surgery" could be used to calculate this answer by filtering for unique values and then counting the number of participants who have data value "1", which represents YES.

Model building to answer chosen question

1. Data splitting

The data is split into a test set and a training set. The training set is used to train the machine algorithm. The test set is only used at the very end of the machine learning process and is used only once.

The proportion of the split really depends on the context of the question. If the training data is too small it reduces the probability of finding accurate parameter estimates (coefficients). The coefficients are measured as the change in outcome variable for 1 unit change in a predictor, with all other predictors remaining the same. If the test data is too small it will provide poor estimates of the performance. For this machine learning problem I have proportioned the training set to contain 80% of the filtered data set and the test data will contain 20% of the filtered data set.

The data set is kept separate from the training data set, this is to ensure there is no information leakage. The test set is used to to check that there is no overfitting of the model. This is when the model learns the data from the data set but then is not generalisable to new data from the outside world.

```
knee_data_split <- knee_data_col %>%
    initial_split(prop = 0.8,
                  strata = `Post-Op Q EQ5D Index`) # 0.8 states the proportion
#of the training set. Strata ensures that the distribution of the variables in
#Post-Op Q EQ5D Index variable matches both the training and data set. The
#distribution is skewed to the right as shown in appendix 1.

knee_train <- training(knee_data_split) #this is the training data
knee_test <- testing(knee_data_split) #this is the test data
```

2. Selection and preprocessing of predictors

A recipe consists of a formula and the data to be used in the machine learning model. The formula is used to specify the outcome variable and predictors. The data undergoes any preprocessing. The recipe includes a combination of any necessary preprocessing steps on the training data before training the model begins. There are different types of preprocessing steps, some examples include transformation of the outcome variable to a more normal distribution, changing qualitative variables to dummy variables (not required for random forest models) and extracting raw data from a variable i.e. a day of the week from a date variable. The recipe can be used during several steps during the training and testing of the model.

```
simple_rec <- knee_train %>% #we are gong to use the training data
  recipe(`Post-Op Q EQ5D Index`~`Pre-Op Q EQ5D Index`+`Pre-Op Q EQ VAS`+
           `Knee Replacement Pre-Op Q Pain`+`Knee Replacement Pre-Op Q Score`) %>%
  # The outcome variable is, Post-Op Q EQ5D and all of the variables to the
  #right of the tilda are the chosen predictors.

  step_zv(all_predictors()) %>% # this step removes any single values
  step_corr(all_predictors()) # this step removes large absolute corrections
```

3. Model specification and training

The random forest model can either be in regression or classification mode. The mode will be set to regression because we have a continuous variable as an output. A random forest model is made up of multiple decision trees. The number of predictors for each to create each tree is chosen at random. The variables to be split at each node (branch) is also chosen at random. The selected engine for the model is randomForest. This engine consists of the parameters mtry, trees, and min_n.

*mtry - This is the number of predictors randomly sampled for each of the decision trees, default - number of columns in the dataset divided by 3

*trees - the number of trees in the model, default - 500L

*min_n - the minimum number of data points in a node (branch) to be able to split further, default - 5

```
# Random forest model specification: rf_spec
rf_spec <- rand_forest() %>%
    set_mode("regression") %>% # select regression as mode
    set_engine("randomForest") # select randomForest for the engine
```

The workflow combines the model specification with the recipe. Different recipes are often needed for different models. It is useful to combine models and recipes because it is easier to train and test workflows.

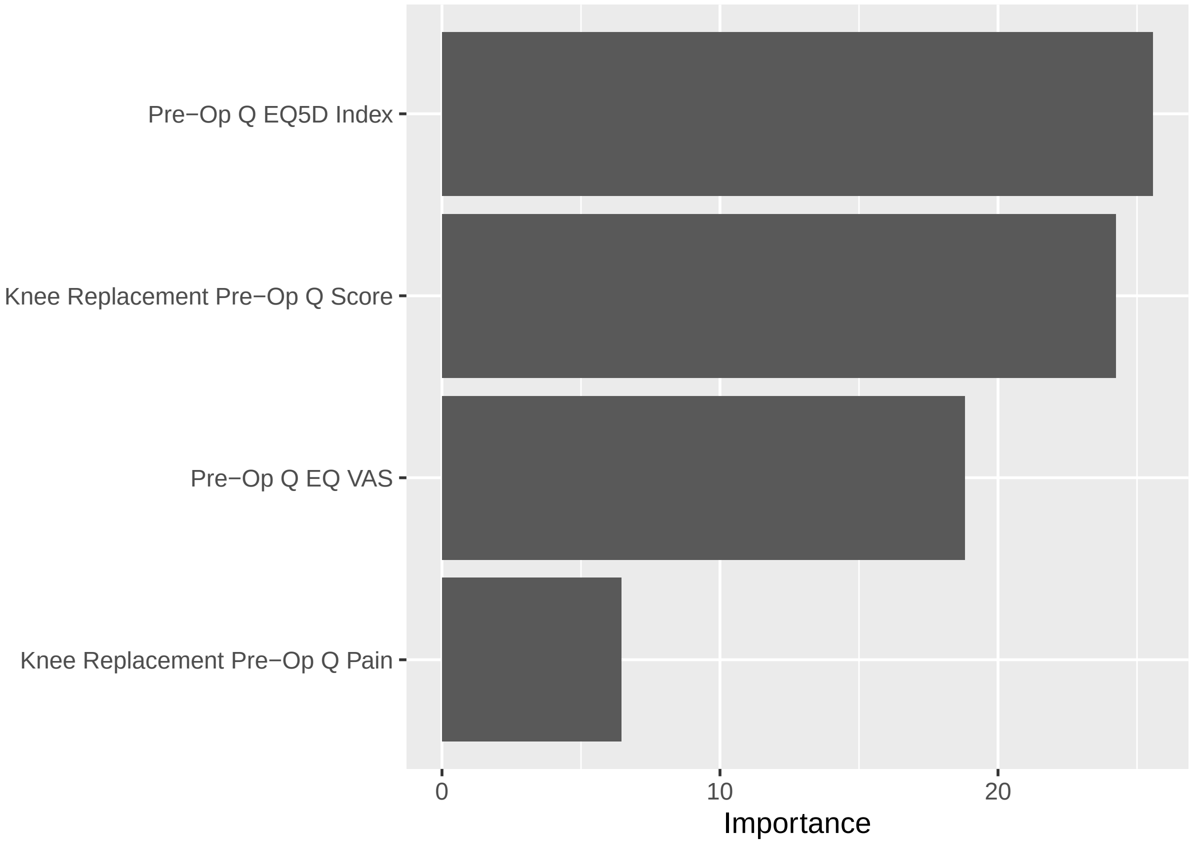
```
knee_wflow_rf <-workflow() %>%
           add_recipe(simple_rec) %>% # adds the recipe created earlier to the
  #workflow

           add_model(rf_spec) # adds the model specification
```

The model is now ready to be trained using the workflow and the training data set.

```
knee_wflow_fit <- knee_wflow_rf %>% #the workflow is selected
    fit(data = knee_train) # the training data is used to train the random
#forest algorithm
```

The trained workflow can now be assessed to determine which of the predictors is most important when predicting the outcome variable.

```
knee_wflow_fit %>%
  extract_fit_parsnip() %>% # this extracts the data
  vip(num_features = 4) # provides an output showing the predictor values in
```



```
#order of importance.
```

It can be seen in the diagram above that the Pre-Op Q EQ5D Index is the most important variable when calculating the outcome variable.

4. Model evaluation

The test data can now be used to assess the random forest model. The success of the prediction is assessed using metrics. The metrics calculated for this regression model are the root mean square error (rmse), R squared (rsq), and the mean absolute error (mae). The root mean square error is used for continuous numerical variables. The difference between each predicted and the actual value are squared. The mean is found for all the squared differences and then the square root is taken. If you have a small RMSE then the actual values and predicted values are close.

R squared (rsq) is also known as the Coefficient of Determination. The closer to 1 the more the change in the outcome variable can be explained by the predictor.

```
predicted_EQ5D_test <- knee_wflow_fit %>% #create a new object for the test
  #using the fitted model

  predict(new_data = knee_test) #use the test data for prediction


results_test <- knee_test %>% #create a new object called results_test
  bind_cols(predicted_EQ5D_test) %>% #add the new predicted column to the data
  #set

  rename(pred_EQ5D = .pred)   # rename the predicted column


metrics(results_test, truth = `Post-Op Q EQ5D Index`, estimate = pred_EQ5D)
```

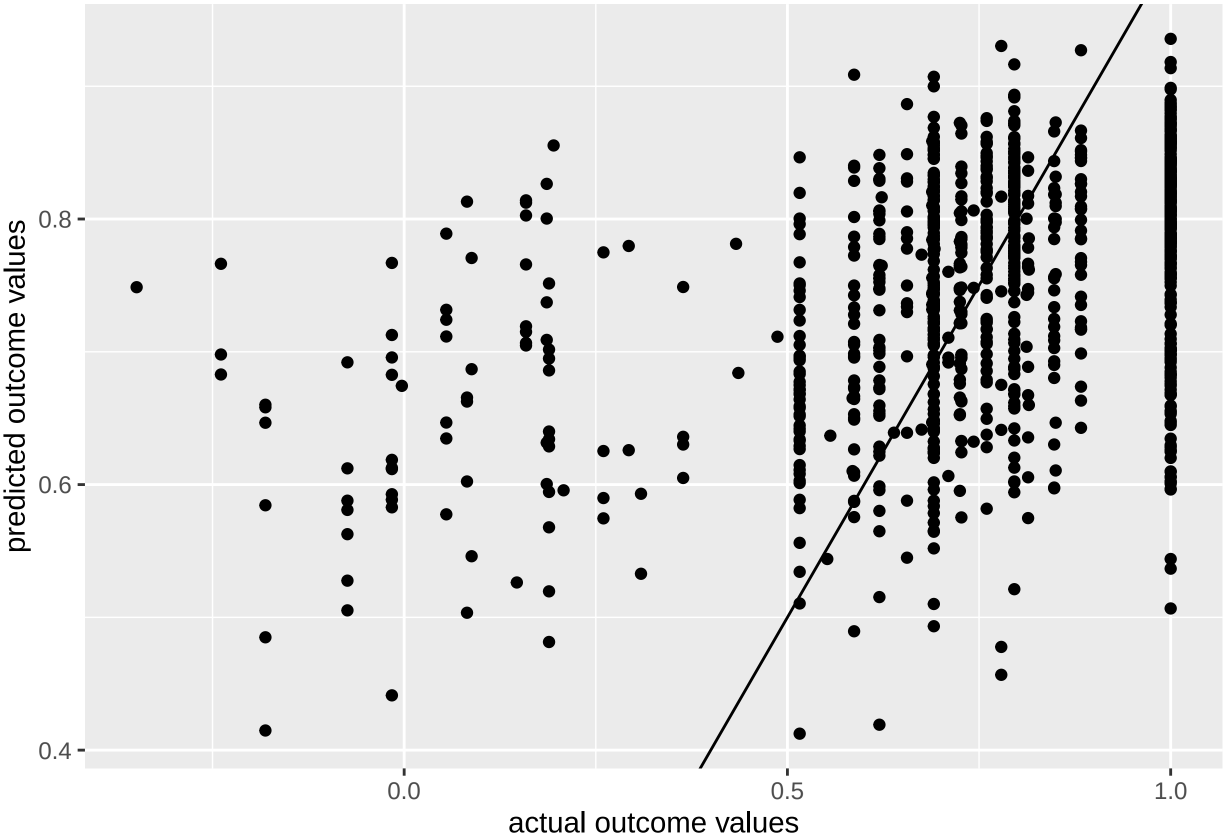
```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.229
## 2 rsq     standard       0.178
## 3 mae     standard       0.164
```

```
# calculate the success of the prediction
```

The RMSE is fairly low and suggests the model fits reasonably well. However, the rsq value suggests there is weak correlation.

The predicted values and the actual values can now be plotted on a scatter graph.

```
results_test %>%
  ggplot(aes(x = `Post-Op Q EQ5D Index`, y = pred_EQ5D)) + # the known outcome
  #variable values and the predicted values are selected
  geom_abline(intercept=0, slope=1) +  # this plots a line going through the
  #origin and with an angle of 45 degrees to the x-axis, we want out points to
  #be as close as possible to the line

  geom_point() +
  xlab("actual outcome values") +
  ylab("predicted outcome values")
```



You can see from the scatter graph above there are considerable amount of data points to the left of the diagonal line. This model would require further work to improve the success of the predictions.

Limitations of machine learning model

There were 81 different variables to chose from in the data set. I selected the variables I thought would be the most suitable. These were numerical variables and pre-operation information. Principle Component Analysis (PCA) is a form of unsupervised learning that can reduce the number of variables in a data set. This would require further investigation as to whether this would be beneficial prior to the random forest being used.

I was going to use age and gender as predictor variables, but random forest wouldn't work for character variables, so I changed to factor but it still didn't work so I removed the variables age and gender from the recipe. I don't think age and gender would have greatly improved the results. The age group is generally older for participants requiring knee replacement, so there would be little variation in the data. Again further investigation into using factor variables would be recommended.

Decision trees are known for high variance. Random forest models reduce the variance and but in turn would increase the bias. Variance is how much the values will fluctuate from the true answer. Bias is how much the value deviates from the actual value. To help improve this the parameters have to be set to give the best balance between variance and bias. In this project the default values were used for the random forest engine, randomForest, finding the settings that optimise the model performance would result in a more accurate model.

Resampling can be used to help chose the correct settings of the parameters. Resampling only uses the training data and not the test data. In appendix 2 cross validation, a form of resampling, is used to compare linear regression and random forest. The cross validation split the data into 10 folds (subsets of data) known as V folds. This is the default value and finds a balance between variance and bias. Each of the subsets of data are then used to evaluate the training set and the average from all the subsets calculated. In this project resampling wasn't used to optimise parameters, but it would be recommended.

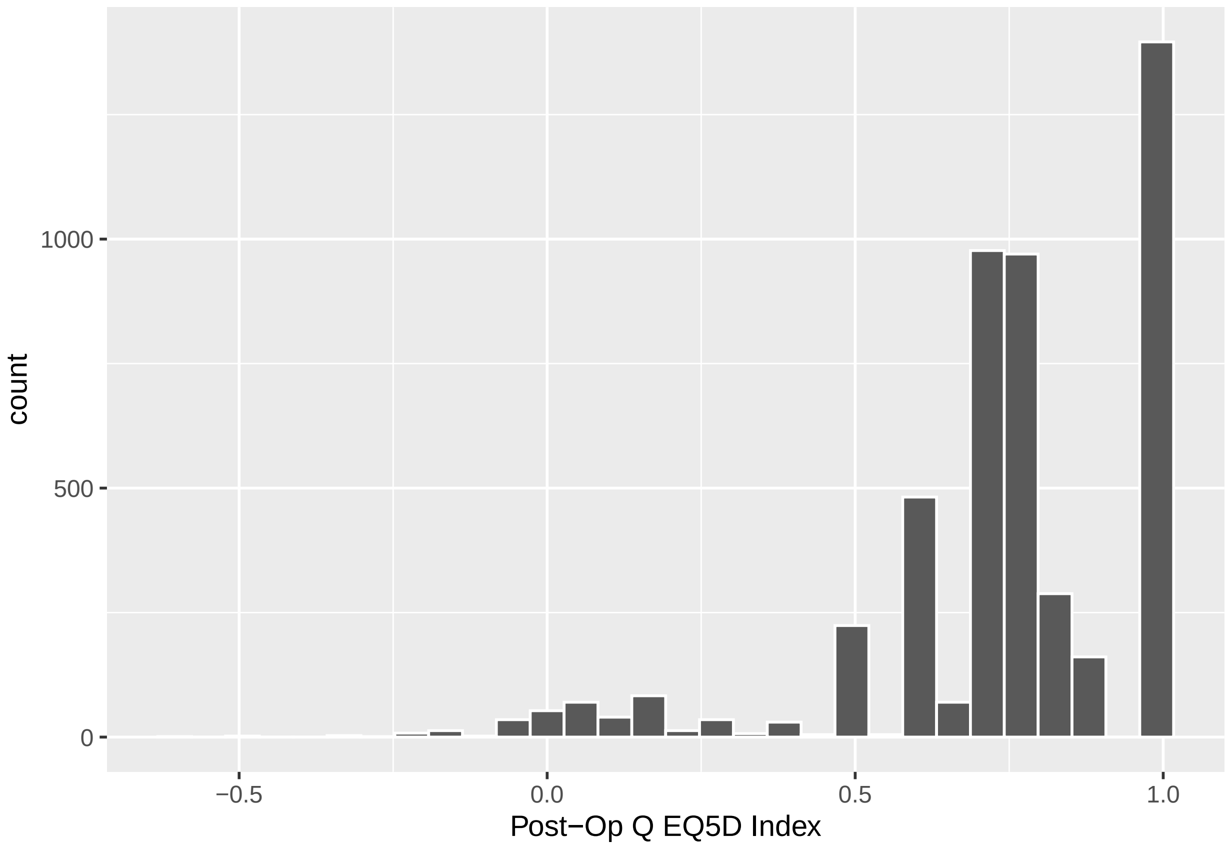
When using regression it is very important to understand the casual relationship between outcome variables and predictor variables. Does one actually cause a change in the other variable or is there a confounding variable that is actually causing the change. I don't fully understand the relationship between each of the variables and this is something that would require further research.

Finally, random forest models are much less interpretable than linear regression models. The calculations at each step of the model within the potentially hundreds of decision trees cannot be fully followed or understood. When it comes to health care it is really important to be able to explain how decisions were made. This would have to be something to consider if used for health care prediction.

Appendix 1 - Linear Regression

I initially decided to try linear regression as a model. There may be confounding variable, factors that affect the Post-Op EQ5D Index other than those chosen, but due to lack of understanding of the research area I have simply chosen variables that are likely to have a relationship.

```
knee_data_col %>%
ggplot(aes(x = `Post-Op Q EQ5D Index`)) +
  geom_histogram(bins = 30, col= "white")
```



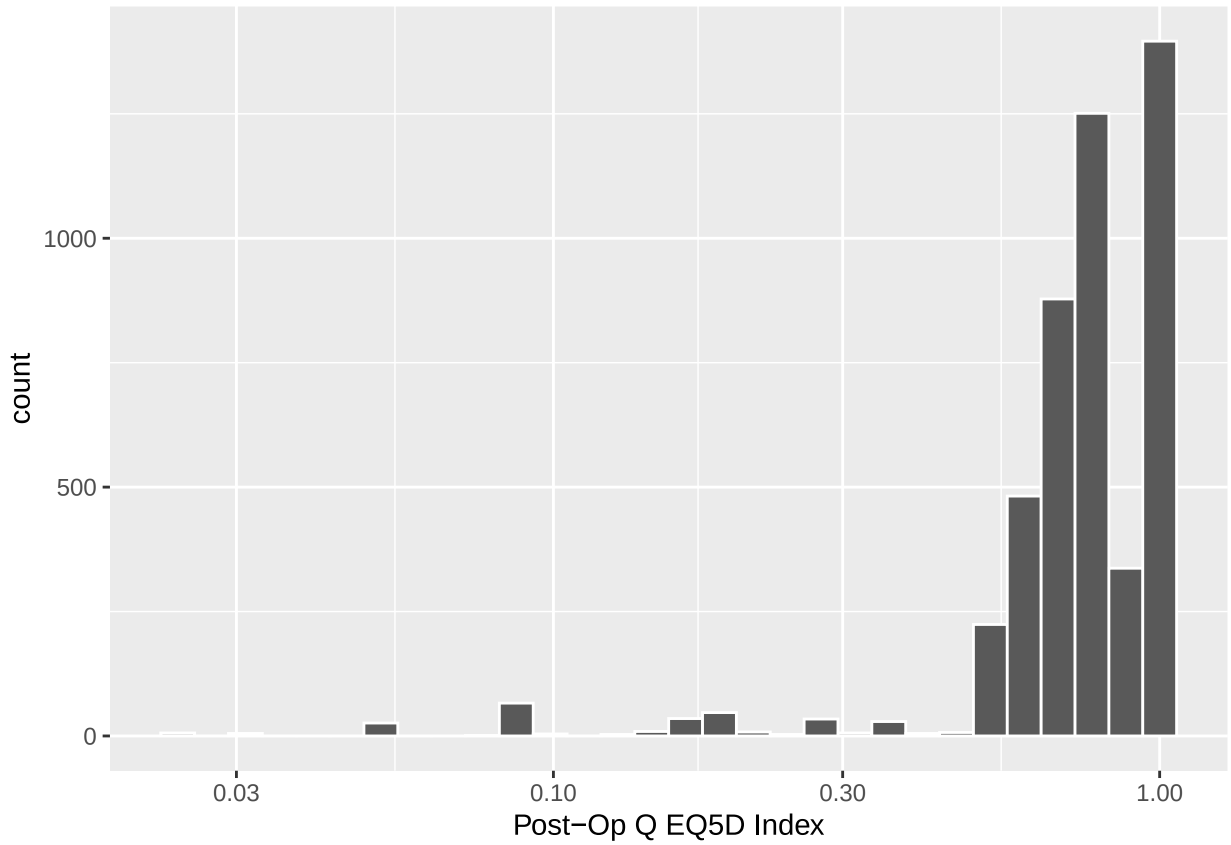
The variable we are trying to predict is not a normal distribution so we would have to transform the data. I used the log10() scale added to the histogram, but it didn't seem to transform the graph to a normal distribution. There are other types of transforming that could be used.

```
knee_data_col %>%
ggplot(aes(x = `Post-Op Q EQ5D Index`)) +
  geom_histogram(bins = 30, col= "white")+ scale_x_log10()
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

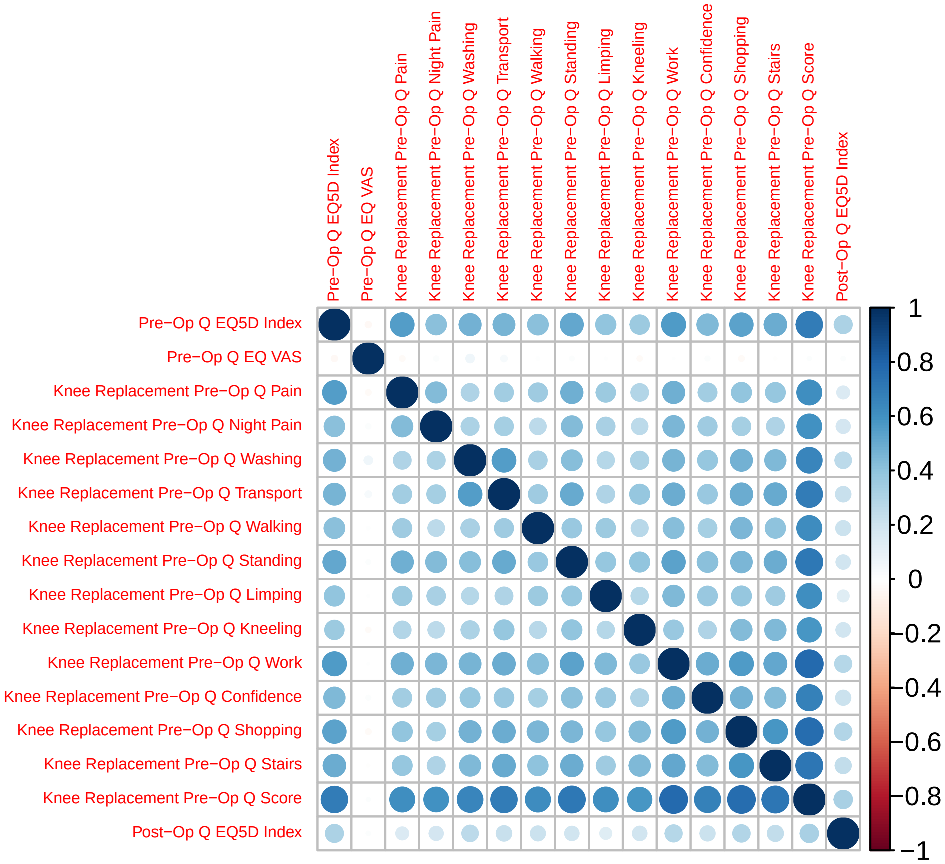
```
## Warning: Removed 112 rows containing non-finite values (stat_bin).
```



The potential predictors can be checked for correlation using the corrplot function as shown below. The reduced number of variables make the corrplot easier to read. If all 81 variables had been included in the corrplot it wouldn't have been legible. It would have been good practice to try different selection of variables and check for correlation.

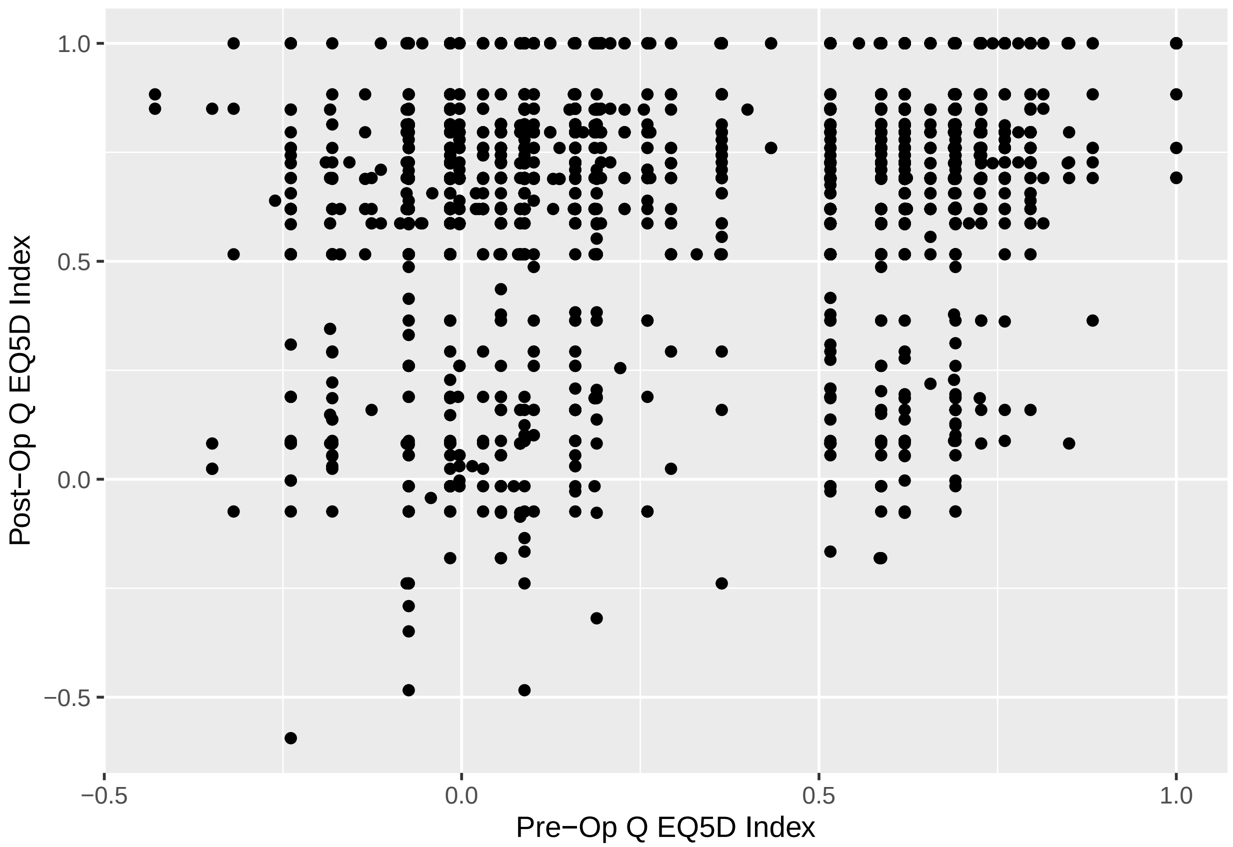
Notice Age Band and Gender are not in the corrplot. This is because these variables are still characters and not numeric. The variables will be changed to numeric in the workflow.

```
knee_data_cor <- cor(knee_train %>% select_if(is.numeric))
corrplot(knee_data_cor, tl.cex = 0.5)
```

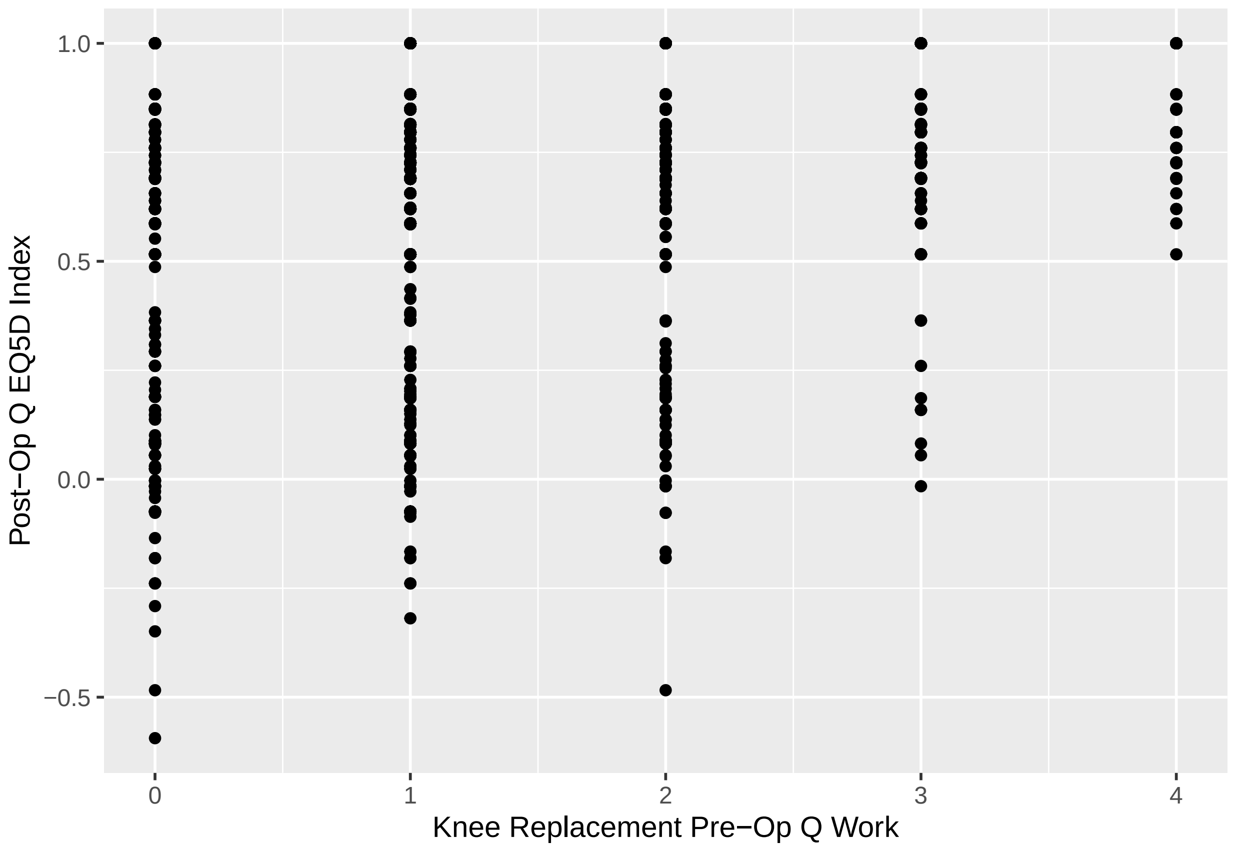


It can be seen in the corrplot that there aren't any strong linear relationships between Post-Op Q EQ5D and the other variables. I checked three of the predictor variables against the outcome variable in separate scatterplots.

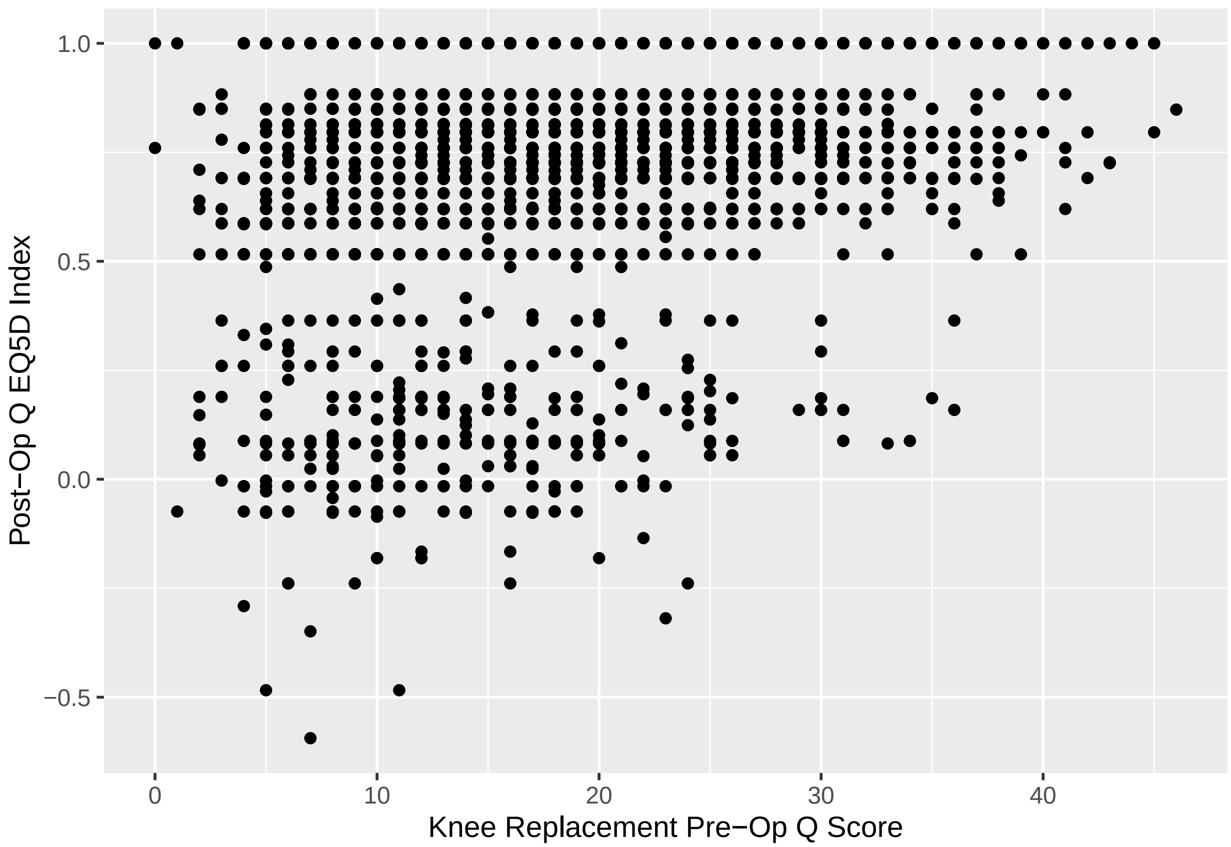
```
knee_train %>%
  ggplot(aes(x =`Pre-Op Q EQ5D Index`, y = `Post-Op Q EQ5D Index`)) +
  geom_point()
```



```
knee_train %>%
  ggplot(aes(x = `Knee Replacement Pre-Op Q Work`, y = `Post-Op Q EQ5D Index`)) +
  geom_point()
```



```
knee_train %>%
  ggplot(aes(x = `Knee Replacement Pre-Op Q Score`, y = `Post-Op Q EQ5D Index`)) +
  geom_point()
```



There was no linearity shown in the 3 graphs. I then decided to compare the linear regression with random forest, see appendix 2.

Appendix 2 Comparison of Linear Regression and Random Forest

```
# Linear regression model specification: lm_spec
lm_spec <- linear_reg()

# Random forest model specification: rf_spec
rf_spec <- rand_forest() %>%
    set_mode("regression") %>%
    set_engine("randomForest")
```

Workflow

And two different workflows:

```
# Linear model workflow: knee_wflow_lm
knee_wflow_lm <-workflow() %>%
           add_recipe(simple_rec) %>%
           add_model(lm_spec)
```

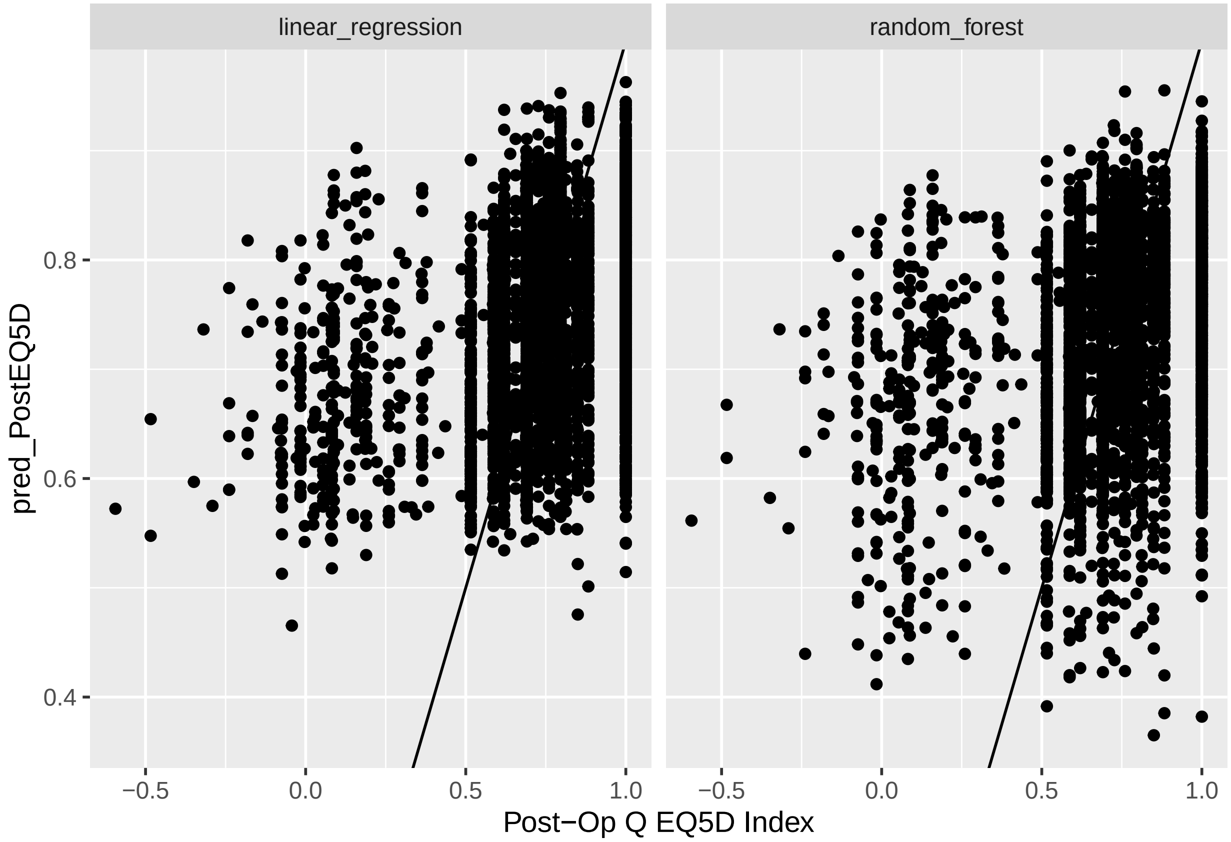
```r
# Random forest workflow: knee_wflow_rf
knee_wflow_rf <-workflow() %>%
           add_recipe(simple_rec) %>%
           add_model(rf_spec)
```

```r
knee_folds <- vfold_cv(knee_train, v = 10) # 10-fold cross validation

# We want to save the predictions
keep_pred <- control_resamples(save_pred = TRUE)

# Fit the two model:

# Linear regression
knee_wflow_lm_fit <- knee_wflow_lm %>%
    fit_resamples(resamples = knee_folds,
                  control = keep_pred)

# Random forest
knee_wflow_rf_fit <- knee_wflow_rf %>%
    fit_resamples(resamples = knee_folds,
                  control = keep_pred)
```

```r
bind_rows(collect_metrics(knee_wflow_lm_fit) %>%
                          mutate(model = "linear_regression"),
          collect_metrics(knee_wflow_rf_fit) %>%
                          mutate(model = "random_forest"))
```

```
## # A tibble: 4 x 7
##   .metric .estimator  mean     n std_err .config              model
##   <chr>   <chr>      <dbl> <int>   <dbl> <chr>                <chr>
## 1 rmse    standard   0.227    10 0.00235 Preprocessor1_Model1 linear_regression
## 2 rsq     standard   0.130    10 0.0135  Preprocessor1_Model1 linear_regression
## 3 rmse    standard   0.225    10 0.00211 Preprocessor1_Model1 random_forest
## 4 rsq     standard   0.144    10 0.0129  Preprocessor1_Model1 random_forest
```

```r
results <-  bind_rows(knee_wflow_lm_fit %>%
                          collect_predictions() %>%
                          mutate(model = "linear_regression") %>%
                          rename(pred_PostEQ5D = .pred),
                      knee_wflow_rf_fit %>%
                          collect_predictions() %>%
                          mutate(model = "random_forest") %>%
                          rename(pred_PostEQ5D = .pred))

results %>%
    ggplot(aes(x = `Post-Op Q EQ5D Index`, y = pred_PostEQ5D)) +
    geom_abline(intercept=0, slope=1) +
    geom_point() +
    facet_wrap(~ model)
```



```
# Fit the two models

knee_wflow_lm_finalfit <- knee_wflow_lm %>%
    last_fit(knee_data_split)

knee_wflow_rf_finalfit <- knee_wflow_rf %>%
    last_fit(knee_data_split)

# Print performance metrics on testing data
bind_rows(collect_metrics(knee_wflow_lm_finalfit) %>%
                          mutate(model = "linear_regression"),
          collect_metrics(knee_wflow_rf_finalfit) %>%
                          mutate(model = "random_forest"))
```

```
## # A tibble: 4 x 5
##   .metric .estimator .estimate .config              model
##   <chr>   <chr>          <dbl> <chr>                <chr>
## 1 rmse    standard       0.232 Preprocessor1_Model1 linear_regression
## 2 rsq     standard       0.158 Preprocessor1_Model1 linear_regression
## 3 rmse    standard       0.229 Preprocessor1_Model1 random_forest
## 4 rsq     standard       0.178 Preprocessor1_Model1 random_forest
```

There isn't much difference in the RSME values, but due to lack of linearity between the predictors and outcome variables I chose random forest as the model.