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.

Load necessary packages

```
library(tidyverse)    # for general data science
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2

## Warning: package 'tibble' was built under R version 4.1.2

## Warning: package 'tidyr' was built under R version 4.1.2
```

```
library(tidymodels)   # for machine learning
```

```
## Warning: package 'tidymodels' was built under R version 4.1.2
## Warning: package 'dials' was built under R version 4.1.2

## Warning: package 'infer' was built under R version 4.1.2

## Warning: package 'modeldata' was built under R version 4.1.2

## Warning: package 'parsnip' was built under R version 4.1.2

## Warning: package 'recipes' was built under R version 4.1.2

## Warning: package 'rsample' was built under R version 4.1.2

## Warning: package 'tune' was built under R version 4.1.2

## Warning: package 'workflows' was built under R version 4.1.2

## Warning: package 'workflowsets' was built under R version 4.1.2

## Warning: package 'yardstick' was built under R version 4.1.2
```

```
library(corrplot)     # for visualising correlation matrices
```

```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
library(vip)          # for variable importance plots
```

```
## Warning: package 'vip' was built under R version 4.1.3
```

```
library(randomForest) # for the random forest model engine
```

```
## Warning: package 'randomForest' was built under R version 4.1.2
```

Load chosen dataset

```
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,~
```

```
knee_data_col<-knee_data%>%
  select(`Pre-Op Q EQ5D Index`,`Age Band`,`Gender`,`Pre-Op Q EQ VAS`,`Knee Replacement Pre-Op Q Pain`:`
  drop_na() %>%     # remove rows with missing values
  unique()%>% # keep unique row
  filter(`Age Band`!="*")%>% # filter rows for Age Band is NOT "*"
filter (`Gender`!="*") # filter rows for Gender is NOT "*"
```

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

```
## Rows: 3,209
## Columns: 18
## $ ‘Pre-Op Q EQ5D Index‘                   <dbl> 0.189, 0.088, 0.088, 0.587, 0.6~
## $ ‘Age Band‘                              <chr> "60 to 69", "60 to 69", "60 to ~
## $ Gender                                  <chr> "1", "1", "1", "1", "1", "1", "~
## $ ‘Pre-Op Q EQ VAS‘                       <dbl> 70, 65, 80, 55, 80, 80, 75, 55,~
## $ ‘Knee Replacement Pre-Op Q Pain‘        <dbl> 1, 0, 0, 1, 3, 1, 1, 0, 0, 0, 0~
## $ ‘Knee Replacement Pre-Op Q Night Pain‘  <dbl> 0, 0, 0, 2, 4, 2, 0, 0, 2, 0, 0~
## $ ‘Knee Replacement Pre-Op Q Washing‘     <dbl> 4, 2, 4, 3, 3, 4, 4, 2, 3, 2, 1~
## $ ‘Knee Replacement Pre-Op Q Transport‘   <dbl> 2, 1, 2, 2, 2, 3, 3, 2, 3, 2, 2~
## $ ‘Knee Replacement Pre-Op Q Walking‘     <dbl> 3, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3~
## $ ‘Knee Replacement Pre-Op Q Standing‘    <dbl> 2, 1, 3, 2, 4, 2, 2, 0, 2, 1, 1~
## $ ‘Knee Replacement Pre-Op Q Limping‘     <dbl> 0, 0, 2, 0, 1, 3, 1, 4, 3, 1, 0~
## $ ‘Knee Replacement Pre-Op Q Kneeling‘    <dbl> 0, 1, 2, 1, 2, 1, 1, 1, 2, 1, 0~
## $ ‘Knee Replacement Pre-Op Q Work‘        <dbl> 0, 1, 3, 2, 2, 2, 2, 0, 2, 1, 1~
## $ ‘Knee Replacement Pre-Op Q Confidence‘  <dbl> 4, 0, 2, 4, 3, 3, 3, 0, 3, 2, 1~
## $ ‘Knee Replacement Pre-Op Q Shopping‘    <dbl> 1, 2, 2, 3, 3, 3, 4, 1, 3, 2, 0~
## $ ‘Knee Replacement Pre-Op Q Stairs‘      <dbl> 0, 1, 3, 2, 3, 1, 3, 2, 1, 2, 0~
## $ ‘Knee Replacement Pre-Op Q Score‘       <dbl> 17, 11, 25, 24, 33, 28, 27, 15,~
## $ ‘Post-Op Q EQ5D Index‘                  <dbl> 0.725, 1.000, 0.725, 0.814, 1.0~
```

Note: I was going to use age and gender as predictor variables, but randomForest wouldn't work for character variable, so I changed to factor but it still didn't work so I removed the variables age and gender from the recipe. This step could have been completed within the recipe. I have left the code to change the variables from character to factor.

For the data set description please go to line 69.

```
class("Age Band") # This checks what type of variable of Age Band. It can also be seen in the summary a
```

```
## [1] "character"
```

```
knee_data_col$`Age Band` <-as.factor(knee_data_col$`Age Band`) # this selects the column Age Band withi
```

```
knee_data_col$Gender<-as.factor(knee_data_col$Gender) # this selects the column Gender within the named
```

```
glimpse(knee_data_col)
```

```
## Rows: 3,209
## Columns: 18
## $ `Pre-Op Q EQ5D Index`                  <dbl> 0.189, 0.088, 0.088, 0.587, 0.6~
## $ `Age Band`                             <fct> 60 to 69, 60 to 69, 60 to 69, 6~
## $ Gender                                 <fct> 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2~
## $ `Pre-Op Q EQ VAS`                      <dbl> 70, 65, 80, 55, 80, 80, 75, 55,~
## $ `Knee Replacement Pre-Op Q Pain`       <dbl> 1, 0, 0, 1, 3, 1, 1, 0, 0, 0, 0~
## $ `Knee Replacement Pre-Op Q Night Pain` <dbl> 0, 0, 0, 2, 4, 2, 0, 0, 2, 0, 0~
## $ `Knee Replacement Pre-Op Q Washing`    <dbl> 4, 2, 4, 3, 3, 4, 4, 2, 3, 2, 1~
## $ `Knee Replacement Pre-Op Q Transport`  <dbl> 2, 1, 2, 2, 2, 3, 3, 2, 3, 2, 2~
## $ `Knee Replacement Pre-Op Q Walking`    <dbl> 3, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3~
## $ `Knee Replacement Pre-Op Q Standing`   <dbl> 2, 1, 3, 2, 4, 2, 2, 0, 2, 1, 1~
## $ `Knee Replacement Pre-Op Q Limping`    <dbl> 0, 0, 2, 0, 1, 3, 1, 4, 3, 1, 0~
## $ `Knee Replacement Pre-Op Q Kneeling`   <dbl> 0, 1, 2, 1, 2, 1, 1, 1, 2, 1, 0~
## $ `Knee Replacement Pre-Op Q Work`       <dbl> 0, 1, 3, 2, 2, 2, 2, 0, 2, 1, 1~
## $ `Knee Replacement Pre-Op Q Confidence` <dbl> 4, 0, 2, 4, 3, 3, 3, 0, 3, 2, 1~
## $ `Knee Replacement Pre-Op Q Shopping`   <dbl> 1, 2, 2, 3, 3, 3, 4, 1, 3, 2, 0~
## $ `Knee Replacement Pre-Op Q Stairs`     <dbl> 0, 1, 3, 2, 3, 1, 3, 2, 1, 2, 0~
## $ `Knee Replacement Pre-Op Q Score`      <dbl> 17, 11, 25, 24, 33, 28, 27, 15,~
## $ `Post-Op Q EQ5D Index`                 <dbl> 0.725, 1.000, 0.725, 0.814, 1.0~
```

Dataset description

Describle the variable and the name of PROMS

This assessment will use the Knee Replacement CCG 2021 data set. Note there are character (qualitative) and double (quantitative) variables within the data set. 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 18 from the 81 variables.

The data set containing the 18 variables was then tidied to remove any rows with missing values. Only unique rows were kept. Finally, only rows in the Age Band and Gender without * were kept.

I then changed the Age Band and Gender variables from characters to factors, but these variables weren't used to train the model.

Chosen potential predictor variables:

Reference to PROMS and briefly explain the variables

- Pre-Op EQ5D Index
- Age Band
- Gender
- Pre-Op Q EQ VAS
- Knee Replacement Pre-Op Q Pain…consecutive variables to… Knee Replacement Pre-Op Q Score

Outcome Variable

- Post-Op Q EQ5D Index

Suitable machine learning algorithm for three questions:

When choosing a model there are many factors to consider; types of data, sample size, accuracy of the model, time available to train and develop and the interpretability of the model. Model type, mode and engine.

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, rounded to 3 decimal places. 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. ref 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. ref

If there is a linear relationship between the outcome variables and the predictors then you can use linear regression. See flowchart which ML. Linear regression, Random Forest - state can be used for both. On analysis the outcome variable 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. ref It would be recommended to transform the outcome variable, see Appendix 1. However, random forest doesn't require transformation of the outcome variable. ref 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. ref

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 categorisation.

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 overfitting??

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.

```
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. Str

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 and extracting raw data from a variable i.e. a day of the week from a date variable.

```
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
  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. Nodes and trees....see CN supervised linear regression

```
# 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.

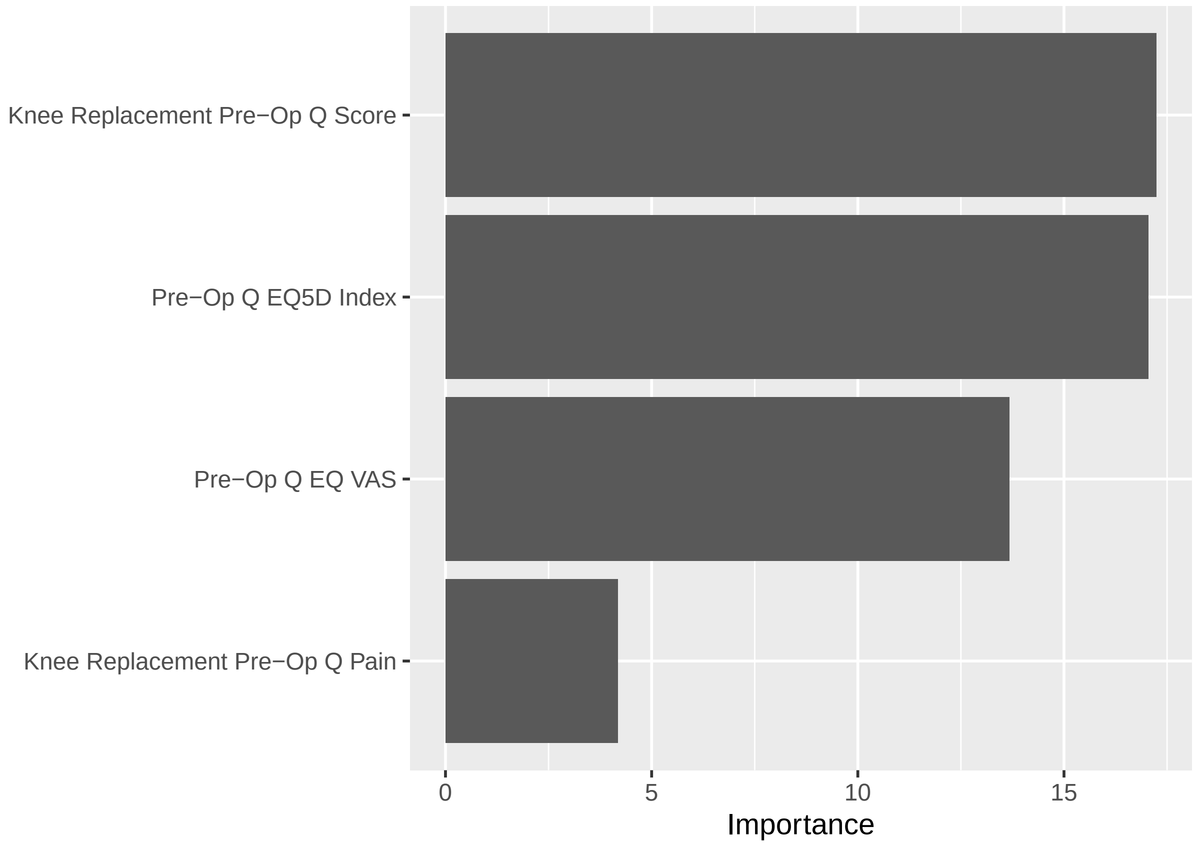
```
knee_wflow_rf <-workflow() %>%
           add_recipe(simple_rec) %>% # adds the recipe created earlier to the workflow
           add_model(rf_spec) # adds the model specification
```

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

```r
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.

```r
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.

---

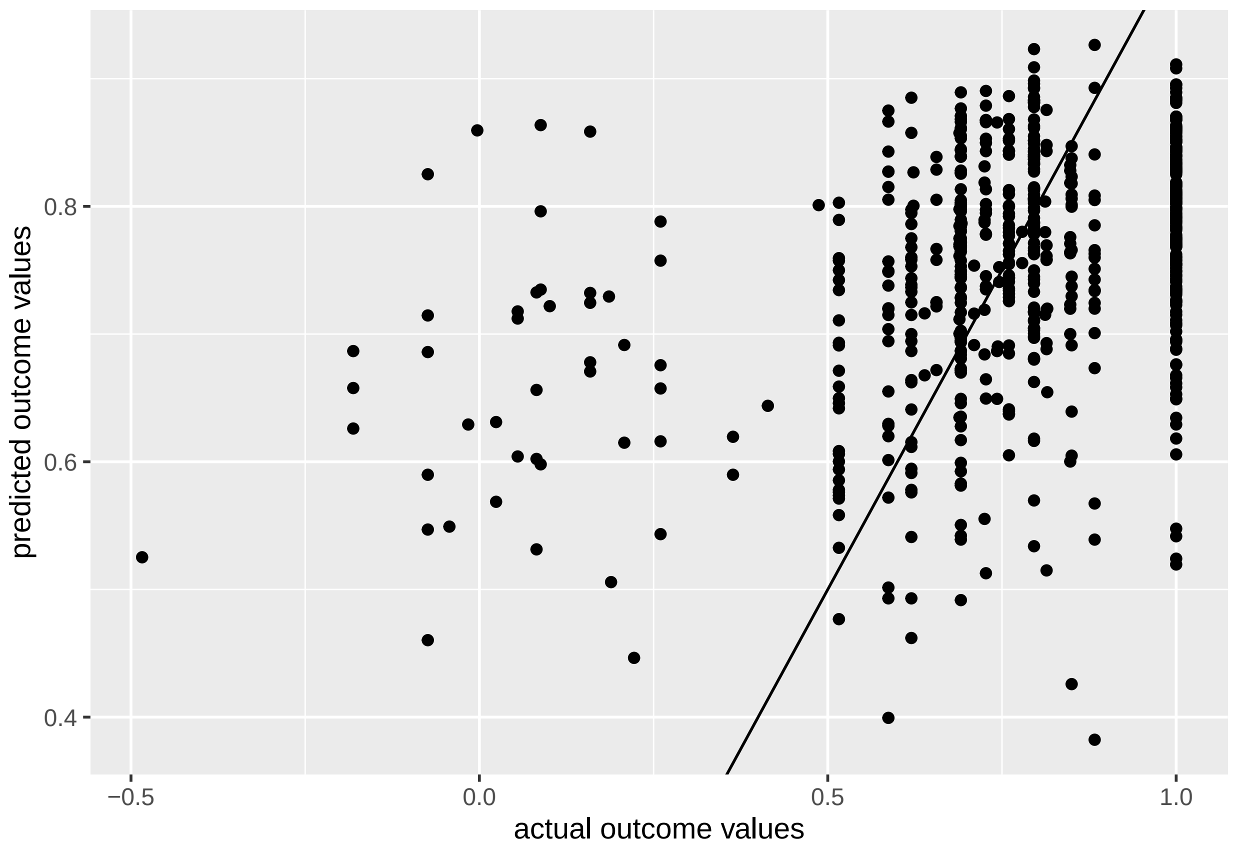
4. Model evaluation

```r
# Use the model for prediction on the same data it was trained on (bad idea!)
predicted_EQ5D_test <- knee_wflow_fit %>%
  predict(new_data = knee_test)

# Add the predicted column to the dataset so we have everything in one dataframe
results_test <- knee_test %>%
  bind_cols(predicted_EQ5D_test) %>%
  rename(pred_EQ5D = .pred)    # rename the predicted column
```

```r
# Evaluate the performance
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       0.221
## 2 rsq     standard       0.149
## 3 mae     standard       0.160
```

```r
results_test %>%
  ggplot(aes(x = `Post-Op Q EQ5D Index`, y = pred_EQ5D)) +
  geom_abline(intercept=0, slope=1) +  # we want data to fall close to this line
  geom_point() +
  xlab("actual outcome values") +
  ylab("predicted outcome values")
```



***

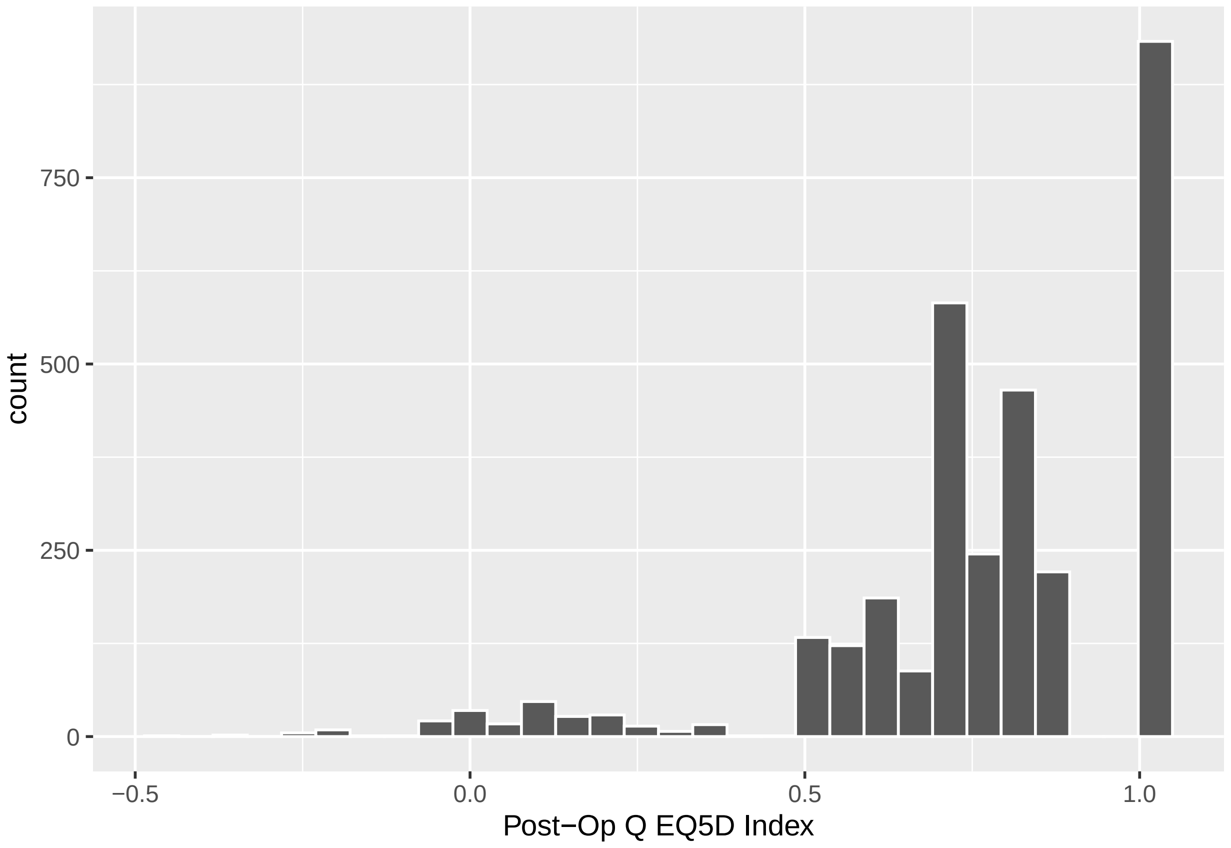
Limitations of machine learning model

Size of sample cost parameter?? i think this is for SVM this is random forest…see CN supervised random forest choose difference variables Understanding of casual limits resampling PCA ?? - see CN section Blackbox Hyperparameters for random forest I iltered the original data source to remove missing values for age and gender. I didn't use age or gender for the machine learning so the extra data removed reduced the training set for the chosen predictors.

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")
```

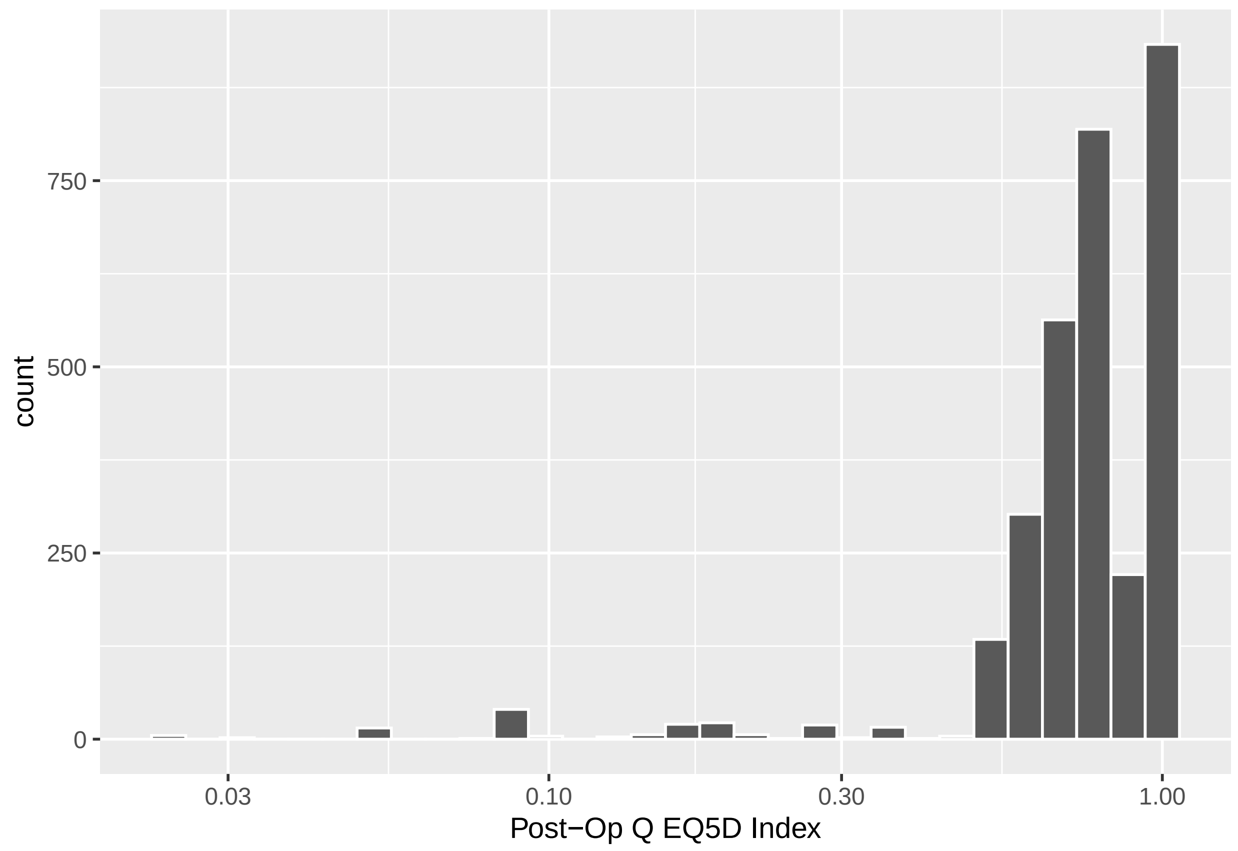


The variable we are trying to predict is not a normal distribution so we would have to transform the data. I used the log10() scale added to the histogram, but it didn't seem to transform the graph to a normal distribution. There are other types of transforming that could be used.

```
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
```

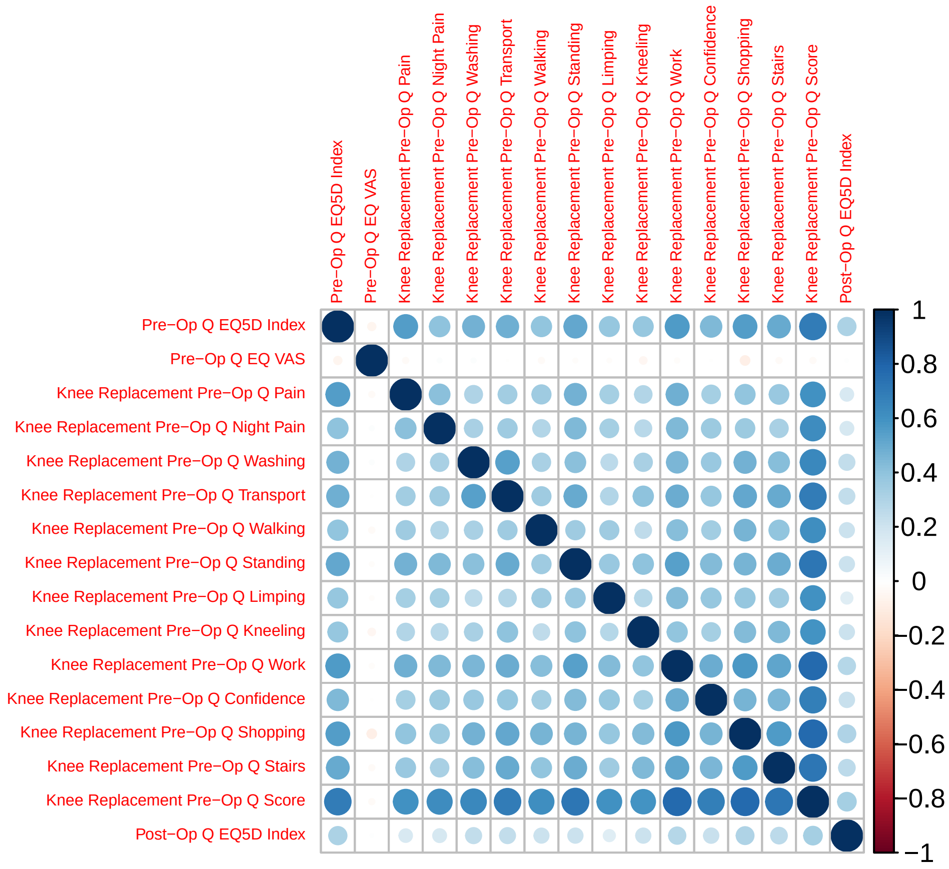
```
## Warning: Removed 70 rows containing non-finite values (stat_bin).
```



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.

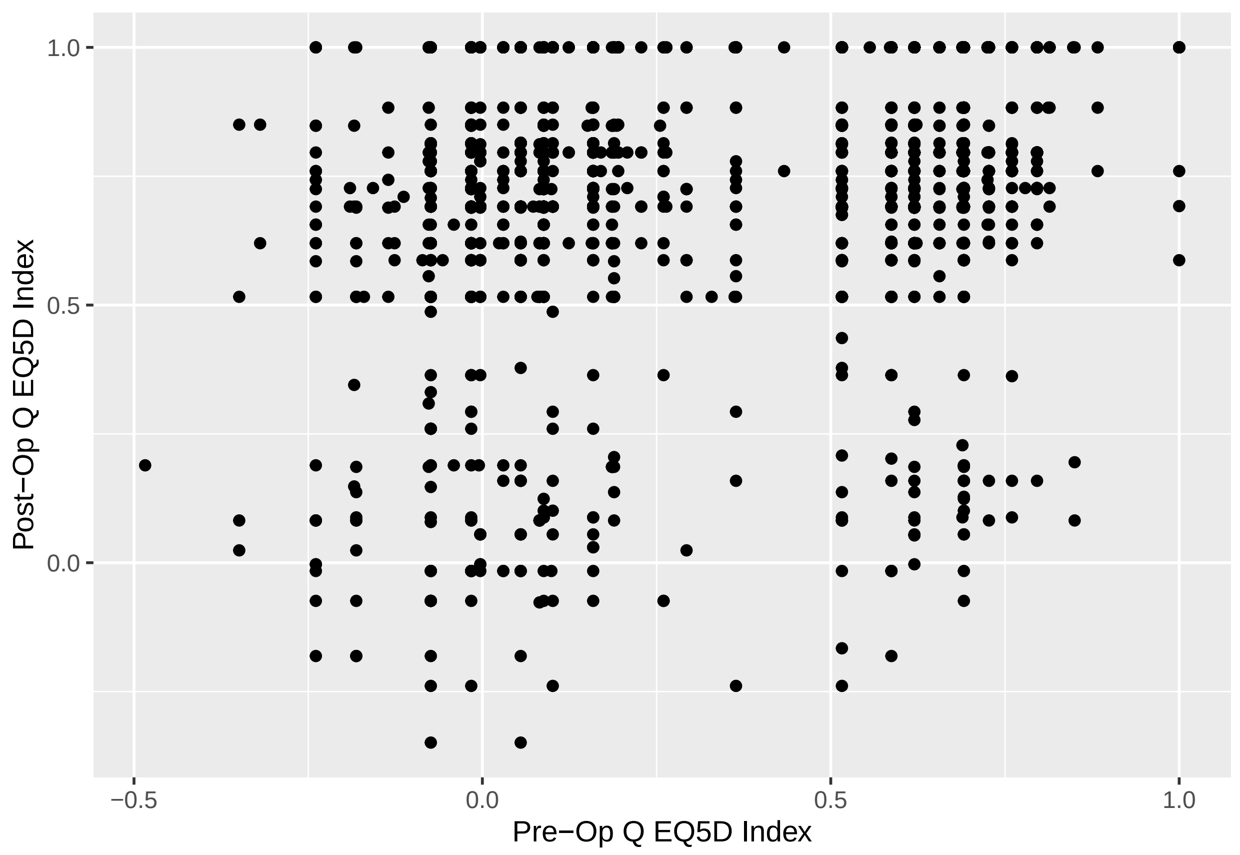
Notice Age Band and Gender are not in the corrplot. This is because these variables are still characters and not numeric. The variables will be changed to numeric in the workflow.

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

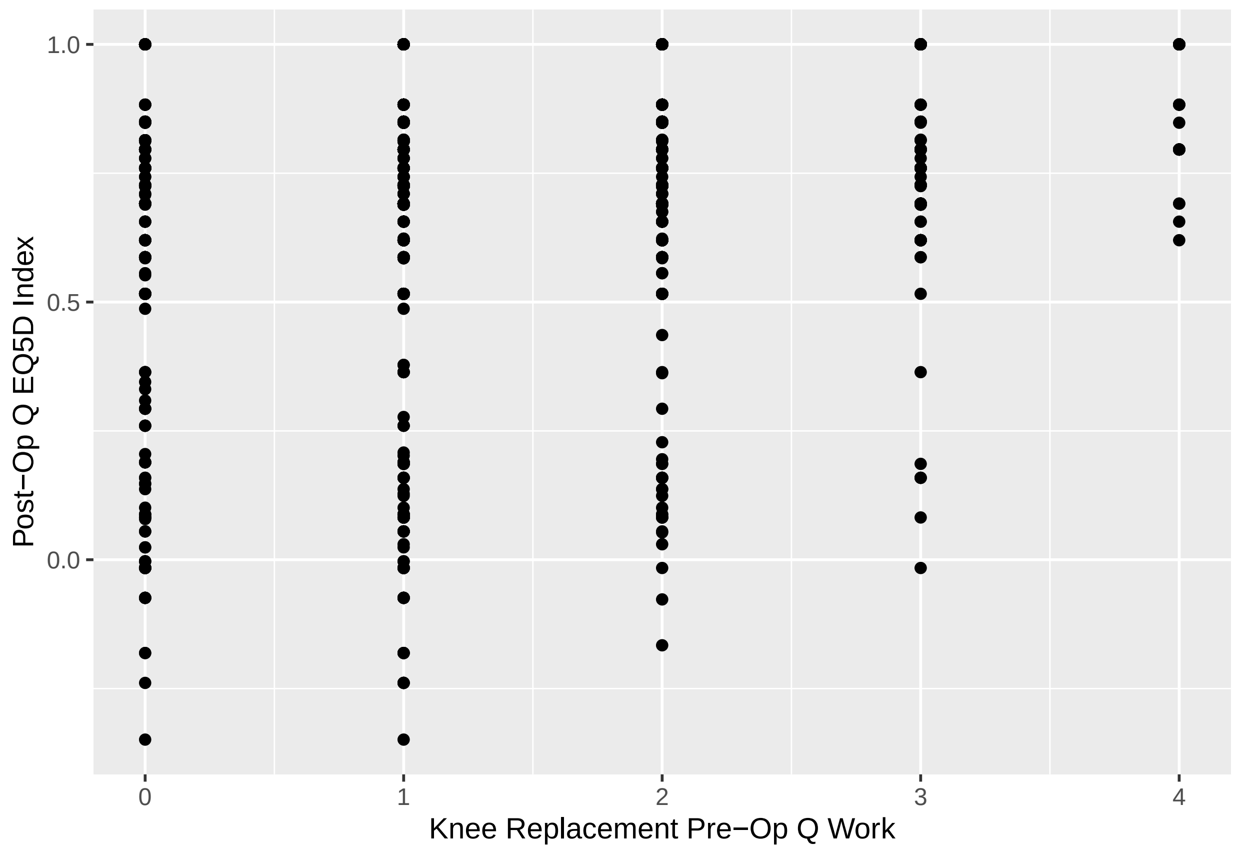


It can be seen in the corrplot that there aren't any strong linear relationships between Post-Op Q EQ5D and the other variables. I checked three of the predictor variables against the outcome variable in separate scatterplots.

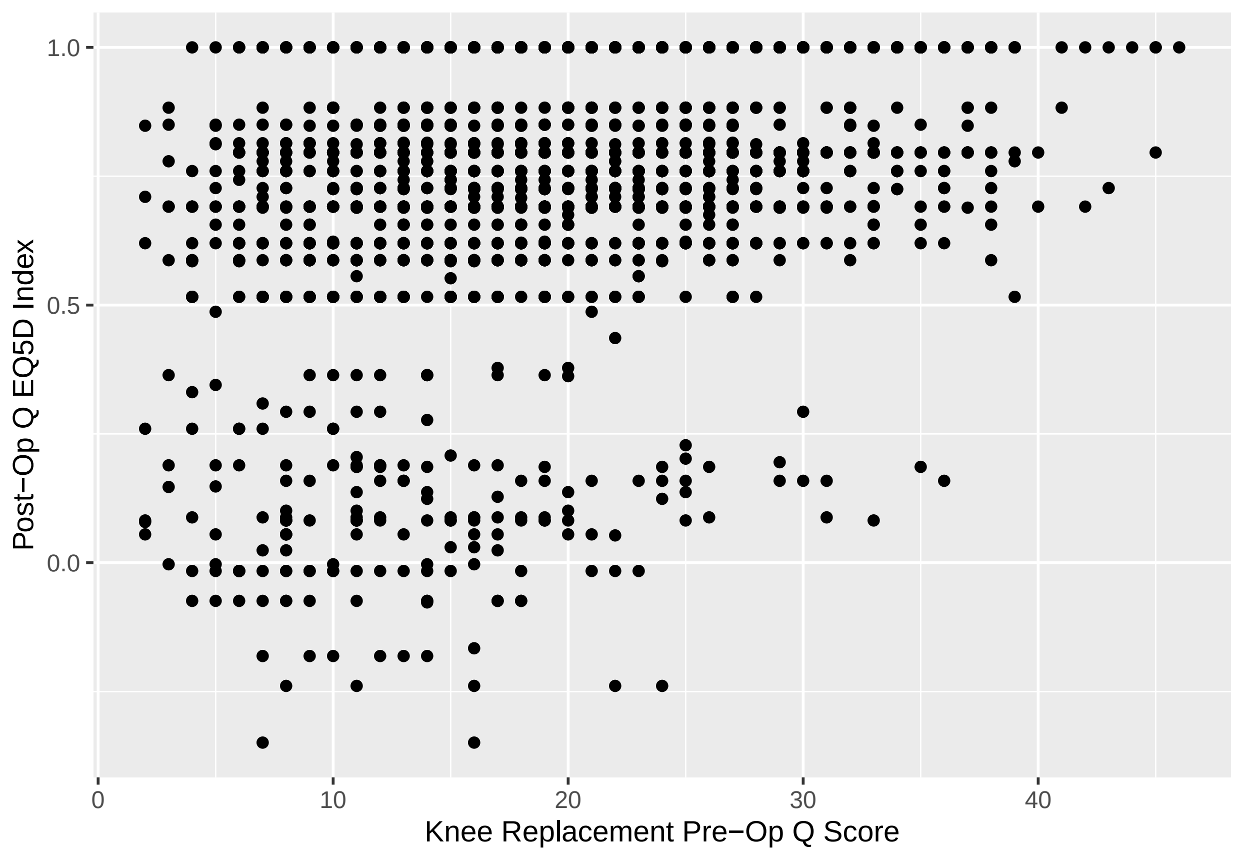
```
knee_train %>%
  ggplot(aes(x =`Pre-Op Q EQ5D Index`, y = `Post-Op Q EQ5D Index`)) +
  geom_point()
```



```r
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: DZ_wflow_lm
knee_wflow_lm <-workflow() %>%
           add_recipe(simple_rec) %>%
           add_model(lm_spec)
```

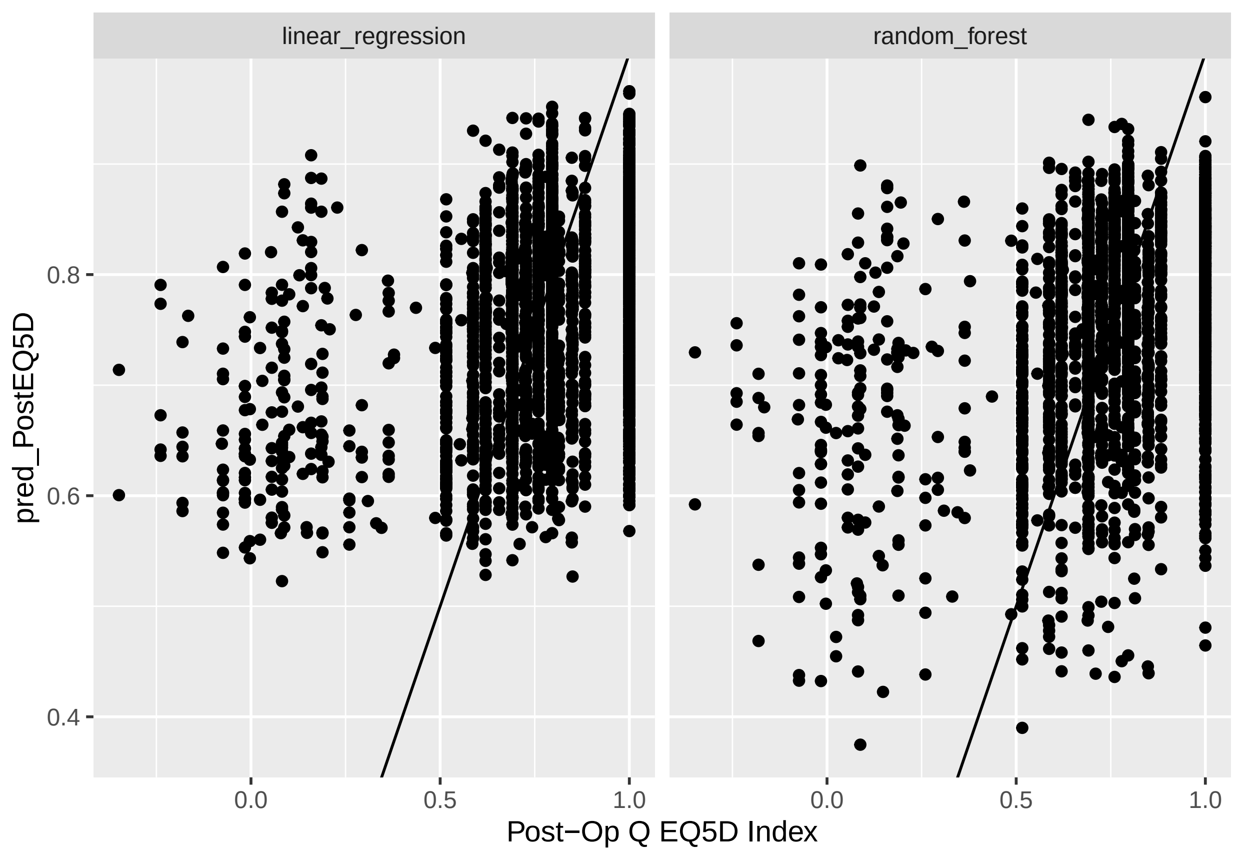
```
# Random forest workflow: DZ_wflow_rf
knee_wflow_rf <-workflow() %>%
           add_recipe(simple_rec) %>%
           add_model(rf_spec)
```

```
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 (make sure you named the workflow DZ_wflow_lm)
knee_wflow_lm_fit <- knee_wflow_lm %>%
    fit_resamples(resamples = knee_folds,
                  control = keep_pred)

# Random forest (make sure you named the workflow DZ_wflow_rf)
knee_wflow_rf_fit <- knee_wflow_rf %>%
    fit_resamples(resamples = knee_folds,
                  control = keep_pred)
```

```
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   0.224    10 0.00332 Preprocessor1_Model1 linear_regression
## 2 rsq     standard   0.134    10 0.0146  Preprocessor1_Model1 linear_regression
## 3 rmse    standard   0.222    10 0.00326 Preprocessor1_Model1 random_forest
## 4 rsq     standard   0.150    10 0.0181  Preprocessor1_Model1 random_forest
```

```
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) +  # we want data to fall close to this line
    geom_point() +
    facet_wrap(~ model)
```



```r
# Fit the two models (make sure you named the data split object DZ_split)

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       0.222 Preprocessor1_Model1 linear_regression
## 2 rsq     standard       0.141 Preprocessor1_Model1 linear_regression
## 3 rmse    standard       0.221 Preprocessor1_Model1 random_forest
## 4 rsq     standard       0.148 Preprocessor1_Model1 random_forest
```