Medical Data Analysis with KNN

Introduction:

The purpose of this project is to see if it is possible to develop a model that can determine if a patient is a risk for readdmission.

Part One: Research Question

Research Question: Which factors, if any, contribute most to a patient being readmitted to the hospital.

Step One: Install the nessecary packages.

In [2]:
```
install.packages("tidyverse")
```

```
also installing the dependencies 'tinytex', 'DBI', 'blob', 'fs', 'rmarkdown', 'whisker',
'selectr', 'dbplyr', 'modelr', 'reprex', 'rvest', 'xml2'

package 'tinytex' successfully unpacked and MD5 sums checked
package 'DBI' successfully unpacked and MD5 sums checked
package 'blob' successfully unpacked and MD5 sums checked
package 'fs' successfully unpacked and MD5 sums checked
package 'rmarkdown' successfully unpacked and MD5 sums checked
package 'whisker' successfully unpacked and MD5 sums checked
package 'selectr' successfully unpacked and MD5 sums checked
package 'dbplyr' successfully unpacked and MD5 sums checked
package 'modelr' successfully unpacked and MD5 sums checked
package 'reprex' successfully unpacked and MD5 sums checked
package 'rvest' successfully unpacked and MD5 sums checked
package 'xml2' successfully unpacked and MD5 sums checked
package 'tidyverse' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\ContactTracer\AppData\Local\Temp\Rtmp2ZO5m0\downloaded_packages
```

In [60]:
```
install.packages('e1071', dependencies=TRUE)
install.packages("ROCR")
install.packages("repr")
```

```
package 'e1071' successfully unpacked and MD5 sums checked
Warning message:
"cannot remove prior installation of package 'e1071'"Warning message in file.copy(savedc
opy, lib, recursive = TRUE):
"problem copying C:\Users\ContactTracer\anaconda3\Lib\R\library\00LOCK\e1071\libs\x64\e1
071.dll to C:\Users\ContactTracer\anaconda3\Lib\R\library\e1071\libs\x64\e1071.dll: Perm
ission denied"Warning message:
"restored 'e1071'"
The downloaded binary packages are in
        C:\Users\ContactTracer\AppData\Local\Temp\Rtmp2ZO5m0\downloaded_packages
Warning message:
"package 'ROCR' is in use and will not be installed"Warning message:
"package 'repr' is in use and will not be installed"
```

In [59]:
```
library(readxl)
library(class)
```

```
library(tidyverse)
library(caret)
library(pROC)
library(ROCR)
library(repr)
```

```
package 'e1071' successfully unpacked and MD5 sums checked
Warning message:
"cannot remove prior installation of package 'e1071'"Warning message in file.copy(savedc
opy, lib, recursive = TRUE):
"problem copying C:\Users\ContactTracer\anaconda3\Lib\R\library\00LOCK\e1071\libs\x64\e1
071.dll to C:\Users\ContactTracer\anaconda3\Lib\R\library\e1071\libs\x64\e1071.dll: Perm
ission denied"Warning message:
"restored 'e1071'"
The downloaded binary packages are in
        C:\Users\ContactTracer\AppData\Local\Temp\Rtmp2ZO5m0\downloaded_packages
Warning message:
"package 'ROCR' is in use and will not be installed"Warning message:
"package 'repr' was built under R version 3.6.3"
```

Step Two: Import and clean the data.

In [4]:
```
medical_clean <- read.csv("C:/Users/ContactTracer/Desktop/D207 Data files/medical_clean
```

In [6]:
```
medical_data <- medical_clean%>% select(15,16,17,19,20,21,22)
```

In [7]:
```
medical_data$Gender <- as.numeric(medical_data$Gender)
```

In [11]:
```
str(medical_data)
```

```
'data.frame':   10000 obs. of  7 variables:
 $ Children   : int  1 3 3 0 1 3 0 7 0 2 ...
 $ Age        : int  53 51 53 78 22 76 50 40 48 78 ...
 $ Income     : num  86576 46806 14370 39741 1210 ...
 $ Gender     : num  2 1 1 2 1 2 2 1 2 1 ...
 $ ReAdmis    : num  1 1 1 1 1 1 1 1 1 1 ...
 $ VitD_levels: num  19.1 18.9 18.1 16.6 17.4 ...
 $ Doc_visits : int  6 4 4 4 5 6 6 7 6 7 ...
```

In [9]:
```
medical_data$ReAdmis <- as.factor(medical_data$ReAdmis)
```

In [10]:
```
medical_data$ReAdmis <- as.numeric(medical_data$ReAdmis)
```

In [13]:
```
summary(medical_data)
```

```
    Children           Age            Income            Gender     
 Min.   : 0.000   Min.   :18.00   Min.   :   154.1   Min.   :1.00  
 1st Qu.: 0.000   1st Qu.:36.00   1st Qu.: 19598.8   1st Qu.:1.00  
 Median : 1.000   Median :53.00   Median : 33768.4   Median :1.00  
 Mean   : 2.097   Mean   :53.51   Mean   : 40490.5   Mean   :1.52  
 3rd Qu.: 3.000   3rd Qu.:71.00   3rd Qu.: 54296.4   3rd Qu.:2.00  
 Max.   :10.000   Max.   :89.00   Max.   :207249.1   Max.   :3.00  
    ReAdmis       VitD_levels      Doc_visits    
 Min.   :1.000   Min.   : 9.806   Min.   :1.000  
 1st Qu.:1.000   1st Qu.:16.626   1st Qu.:4.000  
 Median :1.000   Median :17.951   Median :5.000  
 Mean   :1.367   Mean   :17.964   Mean   :5.012  
 3rd Qu.:2.000   3rd Qu.:19.348   3rd Qu.:6.000  
 Max.   :2.000   Max.   :26.394   Max.   :9.000  
```

Next, create a function to normalize our data so that we can use in

the KNN model. (Sharkie, 2020)

In [14]:
```
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
```

In [15]:
```
Medical_data_norm<- normalize(medical_data)
```

In [16]:
```
summary(Medical_data_norm)
```

```
    Children             Age                Income         
Min.   :0.000e+00   Min.   :8.685e-05   Min.   :0.0007435  
1st Qu.:0.000e+00   1st Qu.:1.737e-04   1st Qu.:0.0945663  
Median :4.825e-06   Median :2.557e-04   Median :0.1629364  
Mean   :1.012e-05   Mean   :2.582e-04   Mean   :0.1953712  
3rd Qu.:1.448e-05   3rd Qu.:3.426e-04   3rd Qu.:0.2619862  
Max.   :4.825e-05   Max.   :4.294e-04   Max.   :1.0000000  
     Gender             ReAdmis           VitD_levels       
Min.   :4.825e-06   Min.   :4.825e-06   Min.   :4.732e-05  
1st Qu.:4.825e-06   1st Qu.:4.825e-06   1st Qu.:8.022e-05  
Median :4.825e-06   Median :4.825e-06   Median :8.662e-05  
Mean   :7.332e-06   Mean   :6.595e-06   Mean   :8.668e-05  
3rd Qu.:9.650e-06   3rd Qu.:9.650e-06   3rd Qu.:9.336e-05  
Max.   :1.448e-05   Max.   :9.650e-06   Max.   :1.274e-04  
  Doc_visits        
Min.   :4.825e-06   
1st Qu.:1.930e-05   
Median :2.413e-05   
Mean   :2.418e-05   
3rd Qu.:2.895e-05   
Max.   :4.343e-05   
```

Part II: Method Justification

KNN is a method for classifying data into groups based on the group membership of the K nearest neighbors to the point in question. There is one main assumption of the KNN method: "The primary assumption that a KNN model makes is that data points/instances which exist in close proximity to each other are highly similar, while if a data point is far away from another group it's dissimilar to those data points." Nelson (2020)

The expected outcome from applying this model to the dataset in question is that the model should have an accuracy of greater than 50% in predicting if a patient will be readmitted to the hospital.

Step Three: Model Creation

To determine which features to use with the KNN model, first I will fit the data to a linear regression model.

In [19]:
```
fit <- glm(ReAdmis~Children+Age+Income+Gender+VitD_levels+Doc_visits,data=Medical_data_
```

In [20]:
```
summary(fit)
```

```
Call:
glm(formula = ReAdmis ~ Children + Age + Income + Gender + VitD_levels + 
    Doc_visits, family = binomial(), data = Medical_data_norm)

Deviance Residuals:
       Min          1Q      Median          3Q         Max  
```

```
-0.0008393  -0.0007288  -0.0007008   0.0010976   0.0012182

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.197e+01  4.269e+01  -0.280    0.779
Children     7.885e+02  3.708e+05   0.002    0.998
Age          5.512e+01  3.911e+04   0.001    0.999
Income      -2.950e-02  2.838e+01  -0.001    0.999
Gender       1.741e+03  1.491e+06   0.001    0.999
VitD_levels  1.318e+02  4.001e+05   0.000    1.000
Doc_visits   2.685e+01  7.720e+05   0.000    1.000

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 0.0078566  on 9999  degrees of freedom
Residual deviance: 0.0078475  on 9993  degrees of freedom
AIC: 14.132

Number of Fisher Scoring iterations: 15
```

From this model the features that will be kept are: Children, Income, and Gender. These features will be used to predict ReAdmis.

In [21]:
```
medical_final <-Medical_data_norm%>% select(1, 3, 4, 5)
```

In [22]:
```
med_samp <- sample(2, nrow(medical_final), replace=TRUE, prob=c(0.67, 0.33))
```

The data will next be split into a training and test set. The split will be 67% to 33% (K-Nearest-Neighbors in R Example – Learn by Marketing, 2020)

In [23]:
```
med_training <- medical_final[med_samp==1, 1:3]
```

In [24]:
```
med_test <- medical_final[med_samp==2, 1:3]
```

In [25]:
```
med_training_group <- medical_final[med_samp==1, 4]
```

In [26]:
```
med_testing_group <- medical_final[med_samp==2, 4]
```

Next, the KNN model will be passed the training and test data. K = 3, K = 5, and K = 7 will all be considered.

In [27]:
```
med_pred3 <- knn(train = med_training, test = med_test, cl = med_training_group, k=3)
```

In [31]:
```
confusionMatrix(table(med_pred3 ,med_testing_group))
```

```
Confusion Matrix and Statistics

                     med_testing_group
med_pred3             4.82511142388556e-06 9.65022284777111e-06
  4.82511142388556e-06                 1467                  841
  9.65022284777111e-06                  640                  359

               Accuracy : 0.5522
                 95% CI : (0.535, 0.5692)
    No Information Rate : 0.6371
    P-Value [Acc > NIR] : 1

                  Kappa : -0.0048

 Mcnemar's Test P-Value : 2.025e-07
```

```
            Sensitivity : 0.6963
            Specificity : 0.2992
         Pos Pred Value : 0.6356
         Neg Pred Value : 0.3594
             Prevalence : 0.6371
         Detection Rate : 0.4436
   Detection Prevalence : 0.6979
      Balanced Accuracy : 0.4977

       'Positive' Class : 4.82511142388556e-06
```

For K = 3 the model is shown to be 55% accurate in predicting if a patient will be readmitted.

In [32]:
```
med_pred5 <- knn(train = med_training, test = med_test, cl = med_training_group, k=5)
```

In [33]:
```
confusionMatrix(table(med_pred5 ,med_testing_group))
```

```
Confusion Matrix and Statistics

                     med_testing_group
med_pred5             4.82511142388556e-06 9.65022284777111e-06
  4.82511142388556e-06                 1565                  867
  9.65022284777111e-06                  542                  333

               Accuracy : 0.5739
                 95% CI : (0.5569, 0.5909)
    No Information Rate : 0.6371
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0215

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7428
            Specificity : 0.2775
         Pos Pred Value : 0.6435
         Neg Pred Value : 0.3806
             Prevalence : 0.6371
         Detection Rate : 0.4732
   Detection Prevalence : 0.7354
      Balanced Accuracy : 0.5101

       'Positive' Class : 4.82511142388556e-06
```

For K = 5 the model is shown to be 57% accurate in predicting if a patient will be readmitted.

In [34]:
```
med_pred7 <- knn(train = med_training, test = med_test, cl = med_training_group, k=7)
```

In [35]:
```
confusionMatrix(table(med_pred7 ,med_testing_group))
```

```
Confusion Matrix and Statistics

                     med_testing_group
med_pred7             4.82511142388556e-06 9.65022284777111e-06
  4.82511142388556e-06                 1628                  911
  9.65022284777111e-06                  479                  289

               Accuracy : 0.5797
```

```
                 95% CI : (0.5626, 0.5966)
    No Information Rate : 0.6371
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0146

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7727
            Specificity : 0.2408
         Pos Pred Value : 0.6412
         Neg Pred Value : 0.3763
             Prevalence : 0.6371
         Detection Rate : 0.4923
   Detection Prevalence : 0.7678
      Balanced Accuracy : 0.5067

       'Positive' Class : 4.82511142388556e-06
```

For K = 7 the model is shown to be 58% accurate in predicting if a patient will be readmitted.

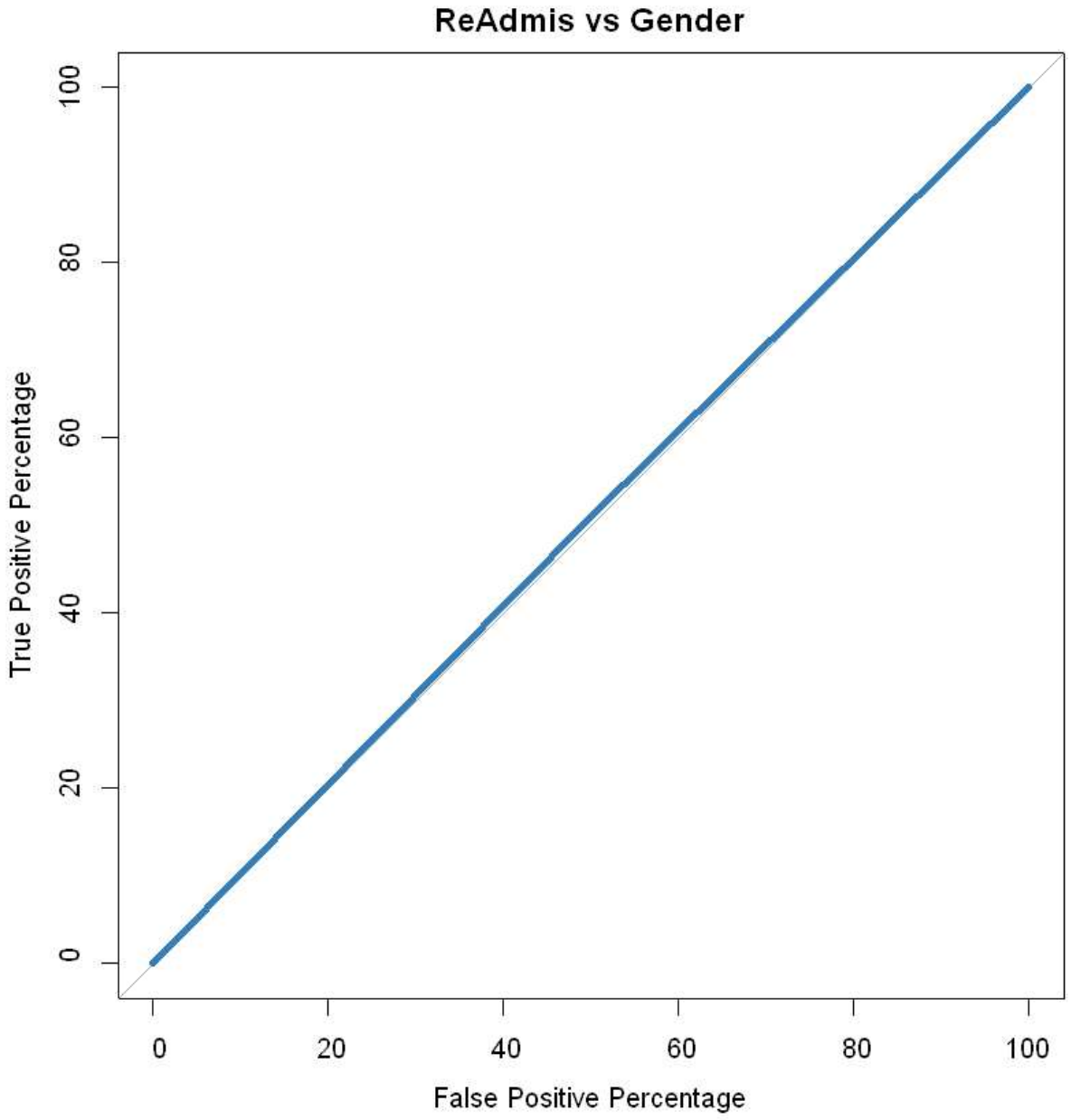
Step Four: Model Testing

To test the validity of the model, a ROC graph will be created and the AUC will be analyzed. (Classification: ROC Curve and AUC | Machine Learning Crash Course, 2020)

In [36]:
```
mod<-class::knn(train = med_training, test = med_test, cl = med_training_group, k=7)
```

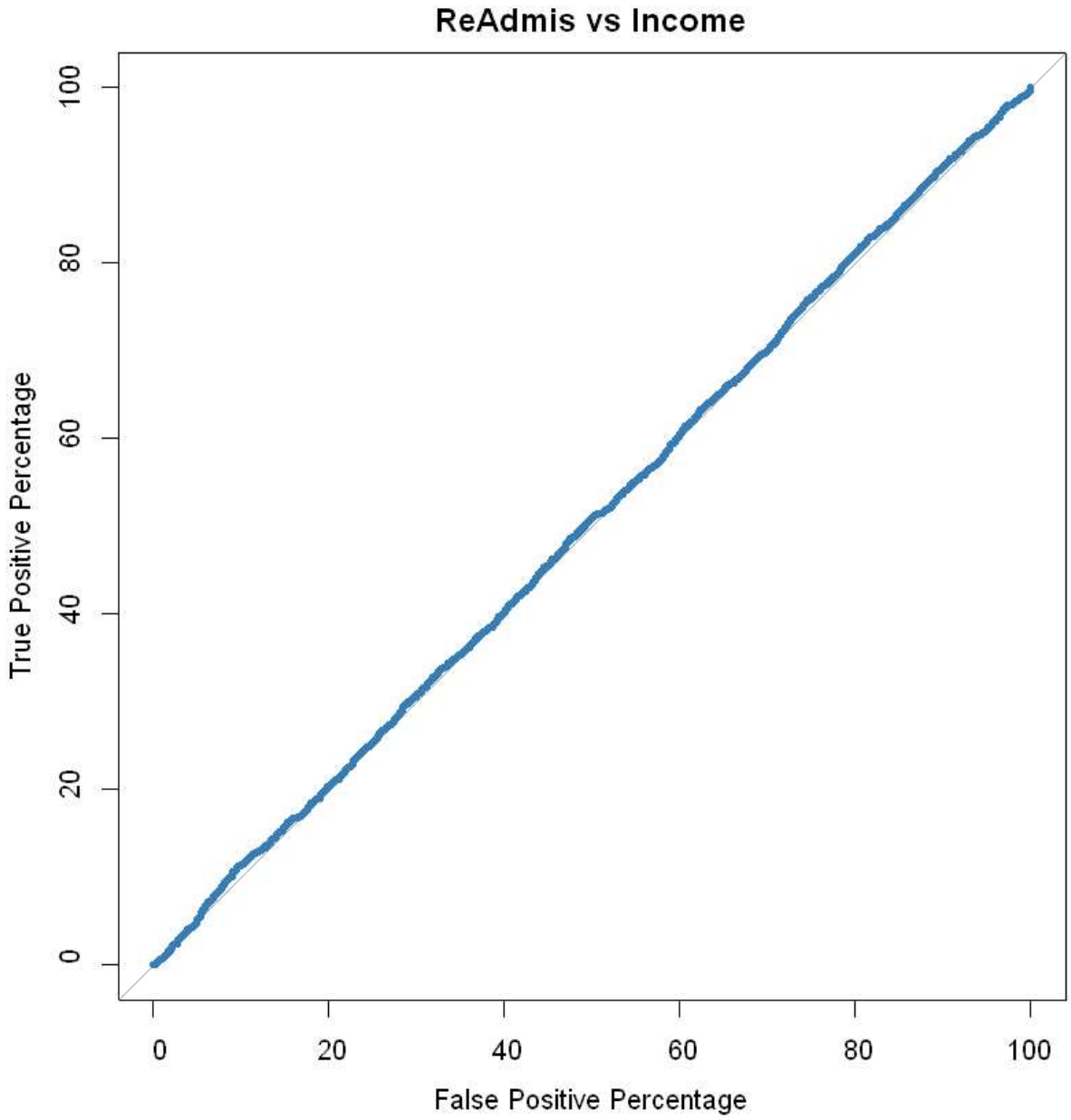
In [56]:
```
roc(medical_final$ReAdmis, medical_final$Gender, plot=TRUE, legacy.axes=TRUE, percent=T
```

```
Setting levels: control = 4.82511142388556e-06, case = 9.65022284777111e-06
Setting direction: controls < cases
Call:
roc.default(response = medical_final$ReAdmis, predictor = medical_final$Gender,     perc
ent = TRUE, plot = TRUE, legacy.axes = TRUE, xlab = "False Positive Percentage",     yla
b = "True Positive Percentage", lwd = 4, col = "#377eb8",     main = "ReAdmis vs Gende
r")

Data: medical_final$Gender in 6331 controls (medical_final$ReAdmis 4.82511142388556e-06)
< 3669 cases (medical_final$ReAdmis 9.65022284777111e-06).
Area under the curve: 50.64%
```



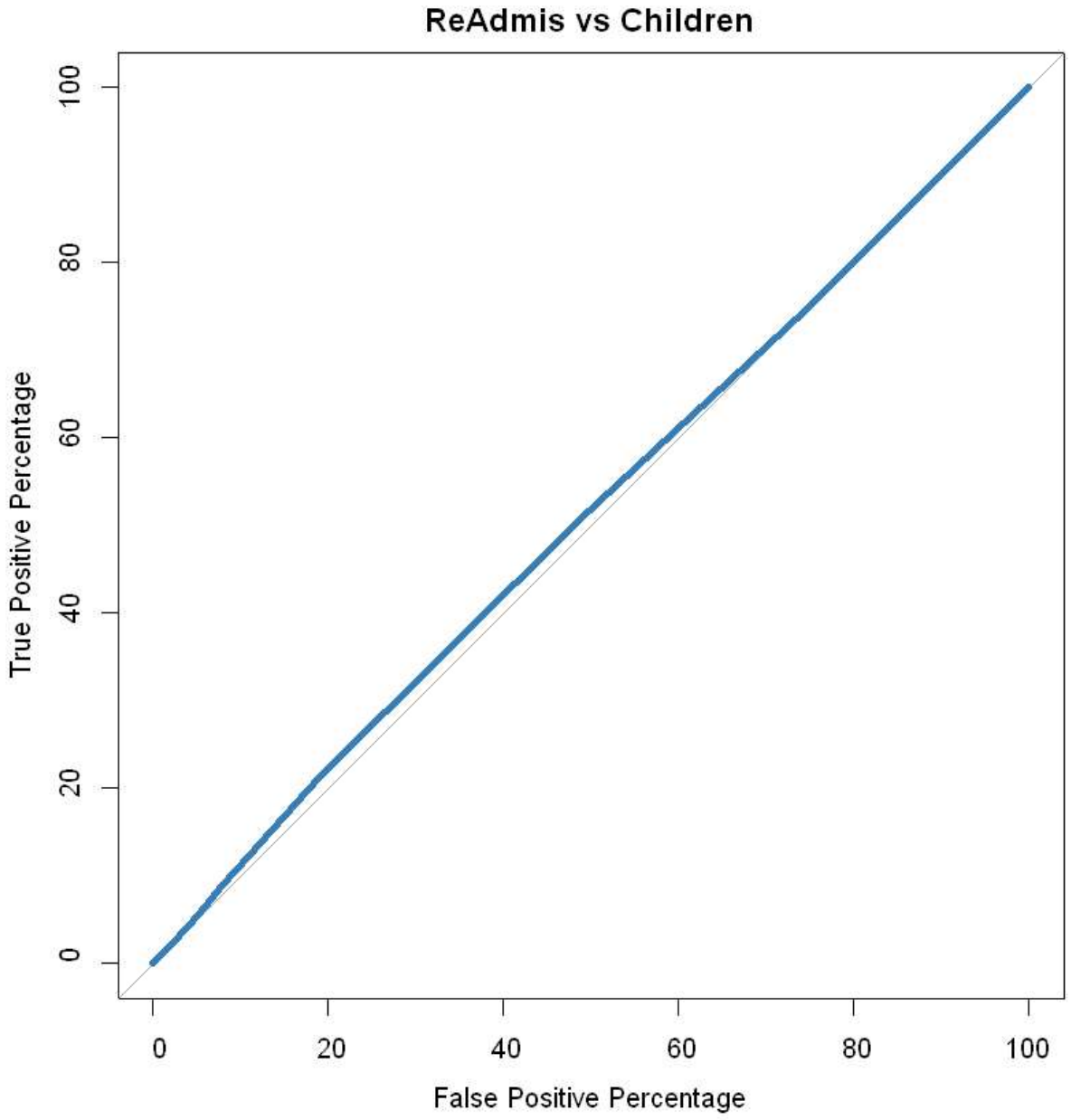
In [57]:
```
roc(medical_final$ReAdmis, medical_final$Income, plot=TRUE, legacy.axes=TRUE, percent=T
```

```
Setting levels: control = 4.82511142388556e-06, case = 9.65022284777111e-06
Setting direction: controls > cases
Call:
roc.default(response = medical_final$ReAdmis, predictor = medical_final$Income,     perc
ent = TRUE, plot = TRUE, legacy.axes = TRUE, xlab = "False Positive Percentage",     yla
b = "True Positive Percentage", lwd = 4, col = "#377eb8",     main = "ReAdmis vs Incom
e")

Data: medical_final$Income in 6331 controls (medical_final$ReAdmis 4.82511142388556e-06)
> 3669 cases (medical_final$ReAdmis 9.65022284777111e-06).
Area under the curve: 50.62%
```



In [58]:
```
roc(medical_final$ReAdmis, medical_final$Children, plot=TRUE, legacy.axes=TRUE, percent
```

```
Setting levels: control = 4.82511142388556e-06, case = 9.65022284777111e-06
Setting direction: controls < cases
Call:
roc.default(response = medical_final$ReAdmis, predictor = medical_final$Children,     pe
rcent = TRUE, plot = TRUE, legacy.axes = TRUE, xlab = "False Positive Percentage",     y
lab = "True Positive Percentage", lwd = 4, col = "#377eb8",     main = "ReAdmis vs Child
ren")

Data: medical_final$Children in 6331 controls (medical_final$ReAdmis 4.82511142388556e-0
6) < 3669 cases (medical_final$ReAdmis 9.65022284777111e-06).
Area under the curve: 51.19%
```



Comparing readdmission to gender, income, and number of children, it is clear that the number of children is the best predictor of future readmission.

References:

Nelson, D. (2020, August 24). What is a KNN (K-Nearest Neighbors)? Unite.AI. https://www.unite.ai/what-is-k-nearest-neighbors/#:%7E:text=The%20primary%20assumption%20that%20a%20KNN%20model%20makes,the%

Sharkie, D. (2020, April 6). How to Normalize Data in R : Machine Learning : Data Sharkie. https://datasharkie.com/how-to-normalize-data-in-r/

Classification: ROC Curve and AUC | Machine Learning Crash Course. (2020, February 10). Google Developers. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

K-Nearest-Neighbors in R Example – Learn by Marketing. (2020). Learn by Marketing.
http://www.learnbymarketing.com/tutorials/k-nearest-neighbors-in-r-example/