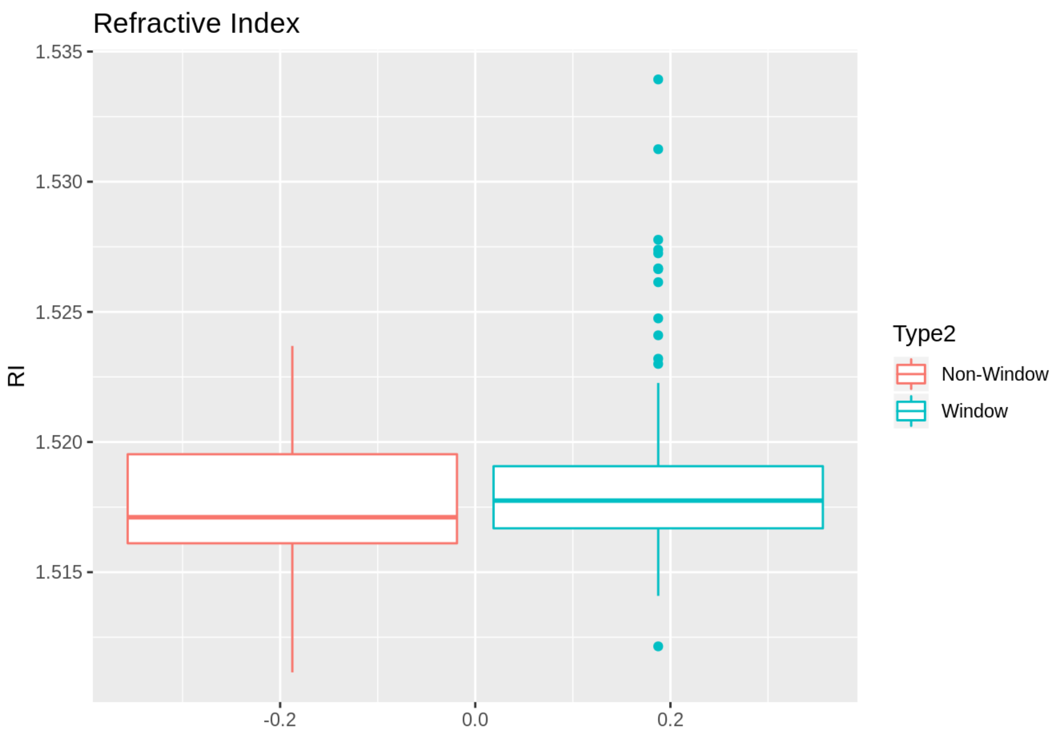
Classification

```r
library(mlbench)
data(Glass)
library(ggplot2)
library(class)
library(caret)
library(e1071)
library(dplyr)
```

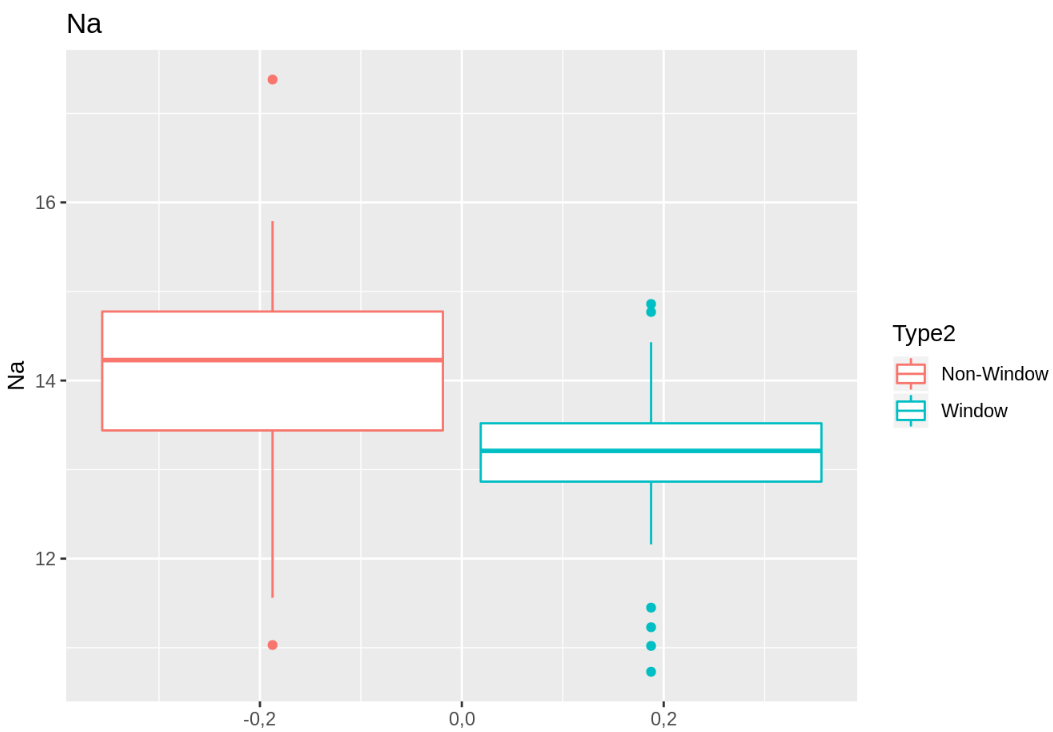
```r
Glass$Type2 <- as.factor(c(rep('Window', 163), rep('Non-Window', 51)))
```

1. Boxplots

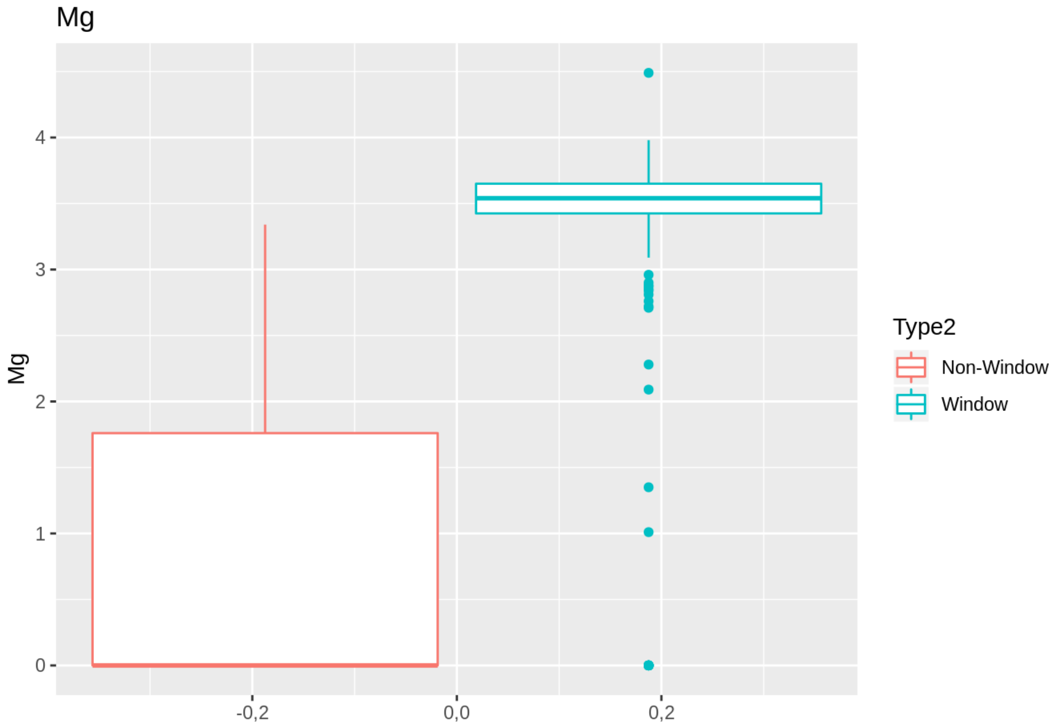
```r
qplot(data = Glass, y = RI, color = Type2, geom = "boxplot", main = "Refractive Index")
```



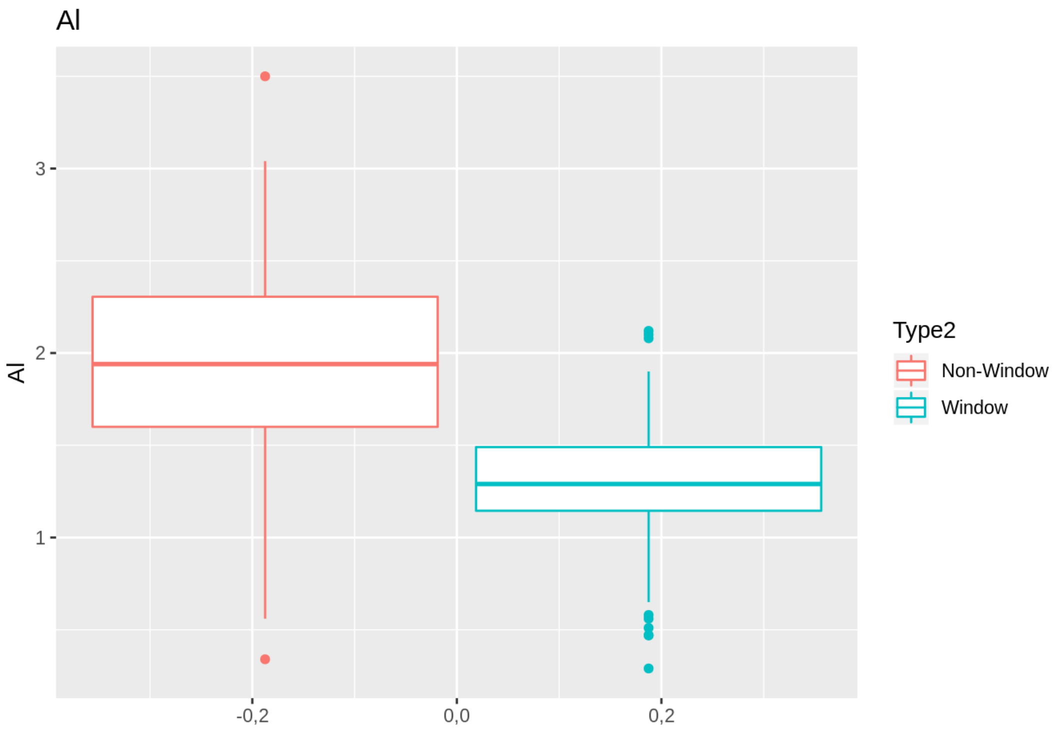

```r
qplot(data = Glass, y = Na, color = Type2, geom = "boxplot", main = "Na")
```



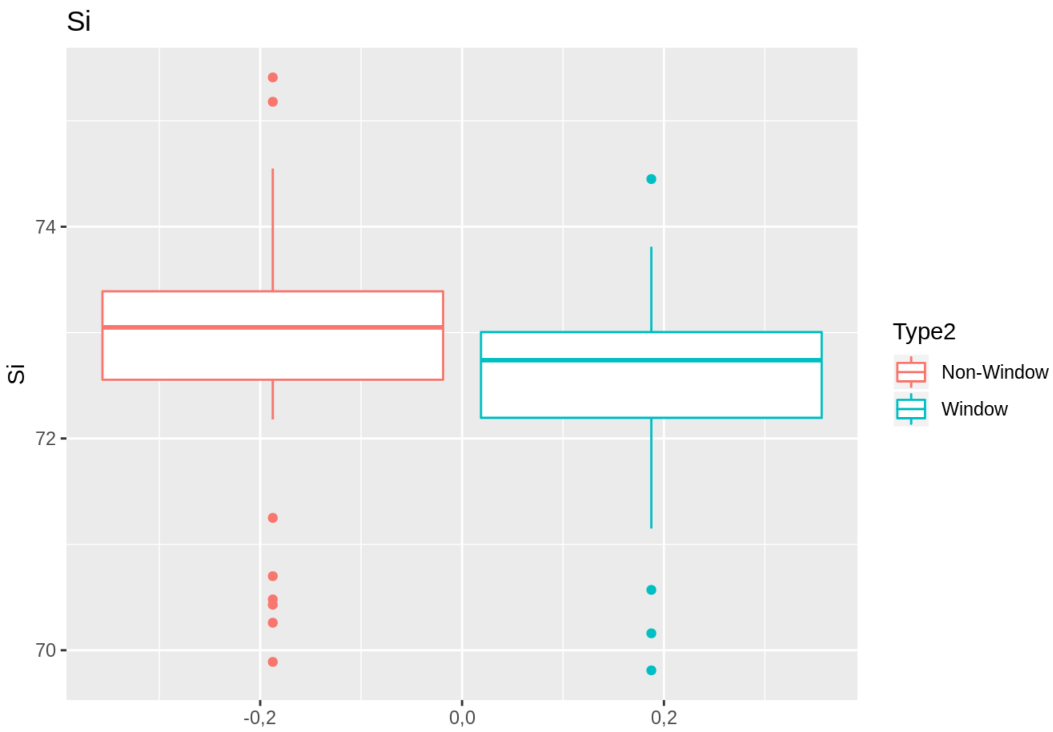

```r
qplot(data = Glass, y = Mg, color = Type2, geom = "boxplot", main = "Mg")
```



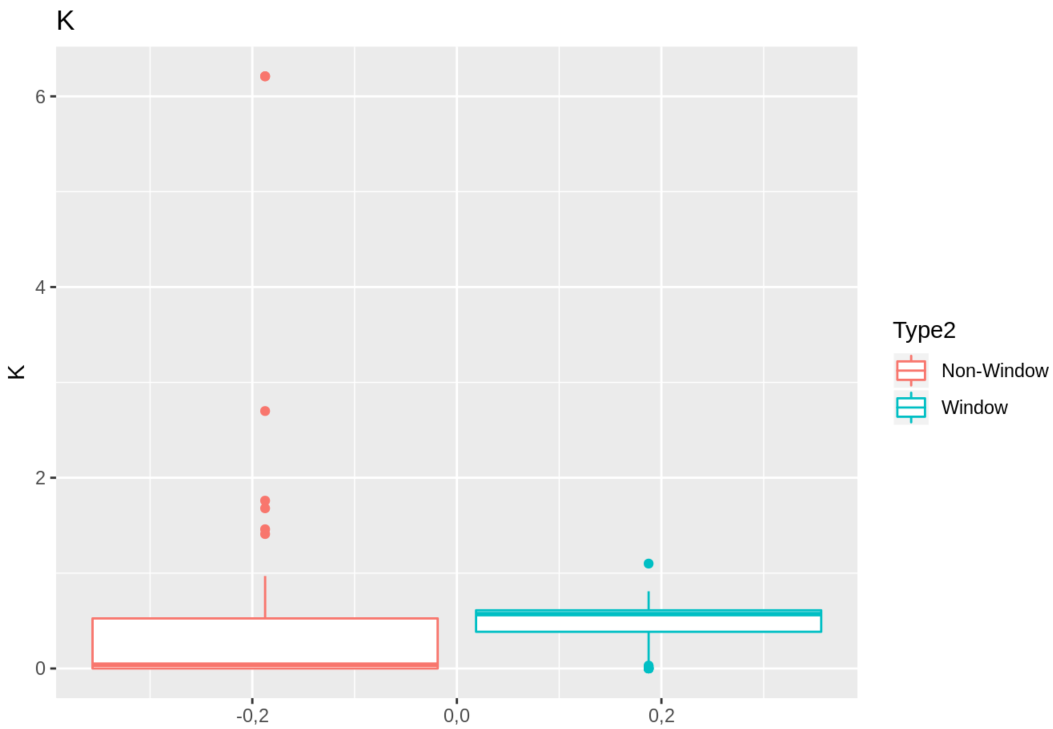

```r
qplot(data = Glass, y = Al, color = Type2, geom = "boxplot", main = "Al")
```




```r
qplot(data = Glass, y = Si, color = Type2, geom = "boxplot", main = "Si")
```

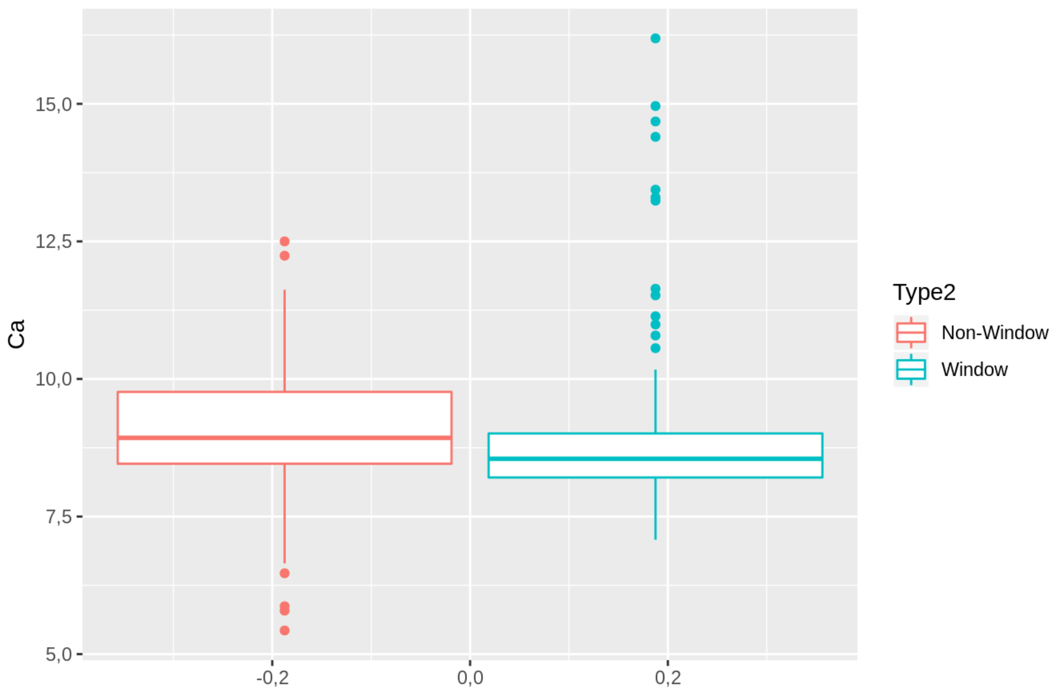



```
qplot(data = Glass, y = K, color = Type2, geom = "boxplot", main = "K")
```



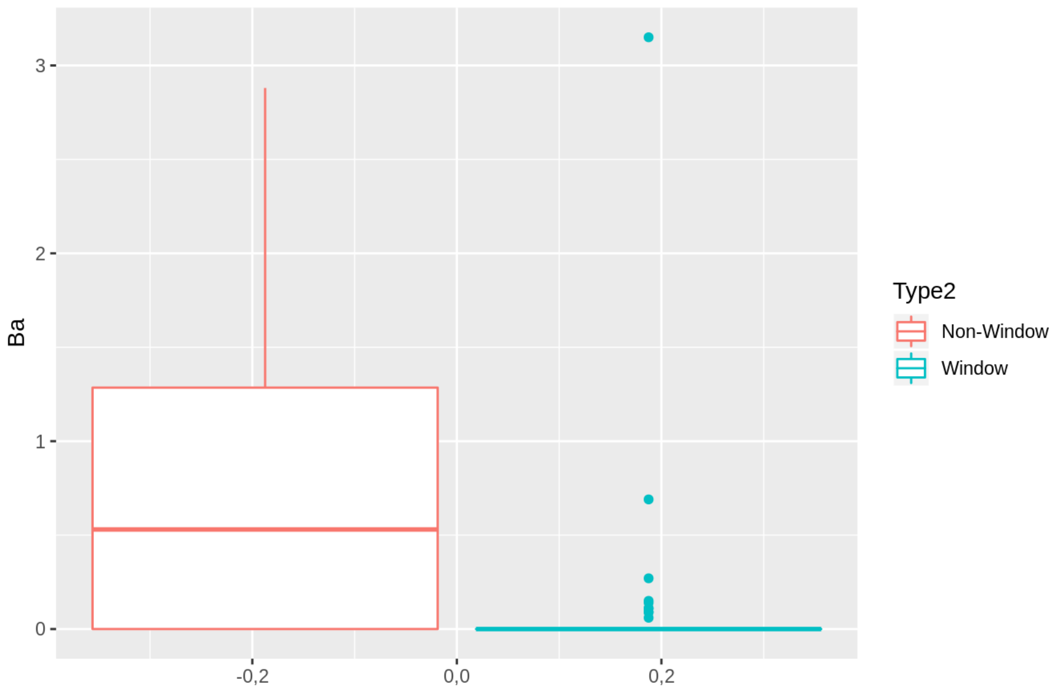

```
qplot(data = Glass, y = Ca, color = Type2, geom = "boxplot", main = "Ca")
```

Ca

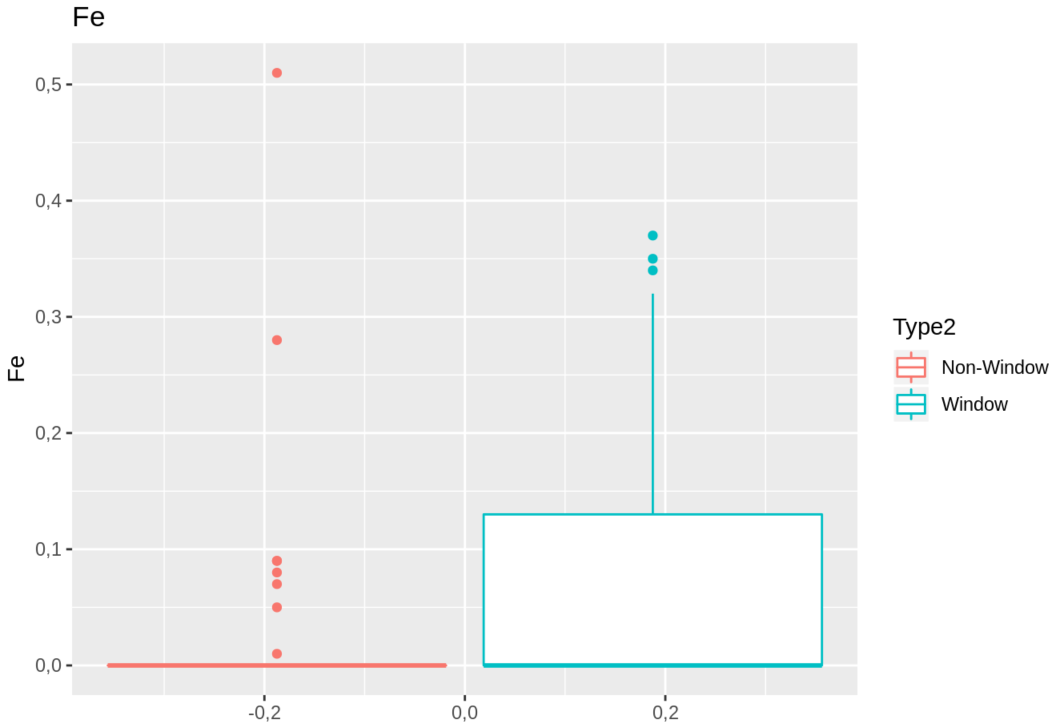



```
qplot(data = Glass, y = Ba, color = Type2, geom = "boxplot", main = "Ba")
```

Ba




```
qplot(data = Glass, y = Fe, color = Type2, geom = "boxplot", main = "Fe")
```



2. KNN

a. Mg

```
set.seed(42)
# Stratified sampling
test <- createDataPartition(Glass$Type2, p=0.25, list = FALSE)

pred_knn <- knn(train = Glass[-test, "Mg", drop = FALSE],
            test = Glass[test, "Mg", drop = FALSE],

            cl = Glass[-test, "Type2"],
            k = 1)

table(pred_knn, Real = Glass[test, "Type2"])
```

```
##            Real
## pred_knn     Non-Window Window
##   Non-Window         10      4
##   Window              3     37
```

```
mean(pred_knn == Glass[test, "Type2"])
```

```
## [1] 0,8703704
```

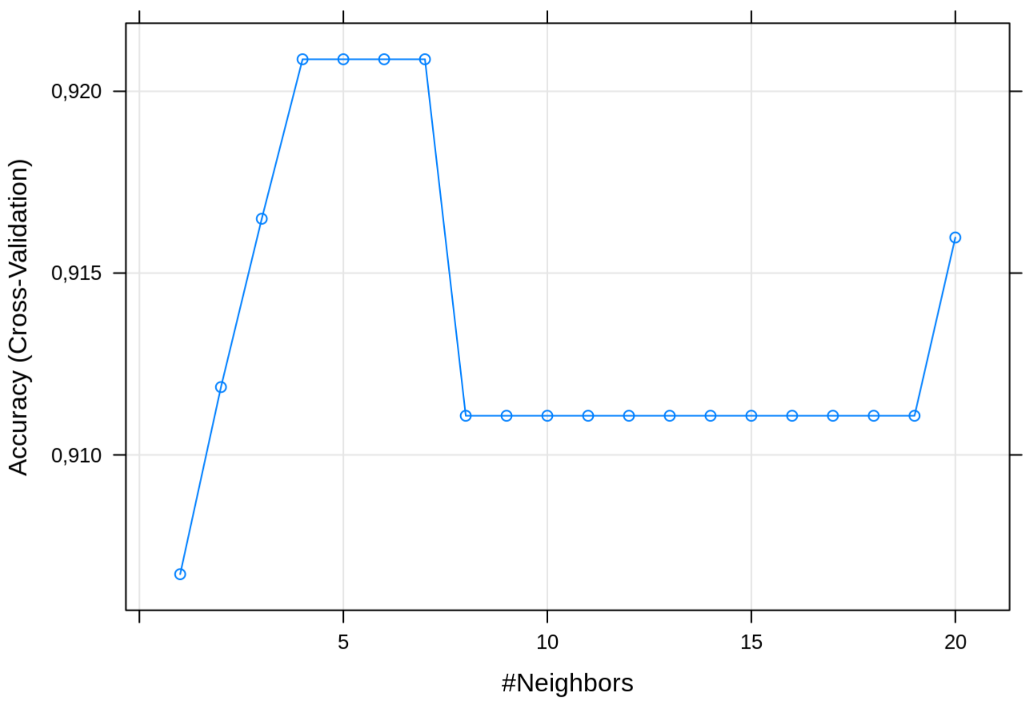
k-fold cross val

```
set.seed(42)

trControl <- trainControl(method  = "cv",
                          number  = 12,
                          savePredictions = TRUE)
fit <- train(Type2 ~ Mg,
             method     = "knn",
             tuneGrid   = expand.grid(k = 1:20),
             trControl  = trControl,
             metric     = "Accuracy",
             data       = Glass)
```

```
fit
```

```
## k-Nearest Neighbors
##
## 214 samples
##   1 predictor
##   2 classes: 'Non-Window', 'Window'
##
## No pre-processing
## Resampling: Cross-Validated (12 fold)
## Summary of sample sizes: 197, 196, 197, 197, 195, 196, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0,9067194  0,7660998
##    2  0,9118650  0,7773690
##    3  0,9164947  0,7928742
##    4  0,9208806  0,8004159
##    5  0,9208806  0,8004159
##    6  0,9208806  0,8004159
##    7  0,9208806  0,8004159
##    8  0,9110767  0,7701743
##    9  0,9110767  0,7701743
##   10  0,9110767  0,7701743
##   11  0,9110767  0,7701743
##   12  0,9110767  0,7701743
##   13  0,9110767  0,7701743
##   14  0,9110767  0,7701743
##   15  0,9110767  0,7701743
##   16  0,9110767  0,7701743
##   17  0,9110767  0,7701743
##   18  0,9110767  0,7701743
##   19  0,9110767  0,7701743
##   20  0,9159787  0,7783715
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 7.
```

```
plot(fit)
```

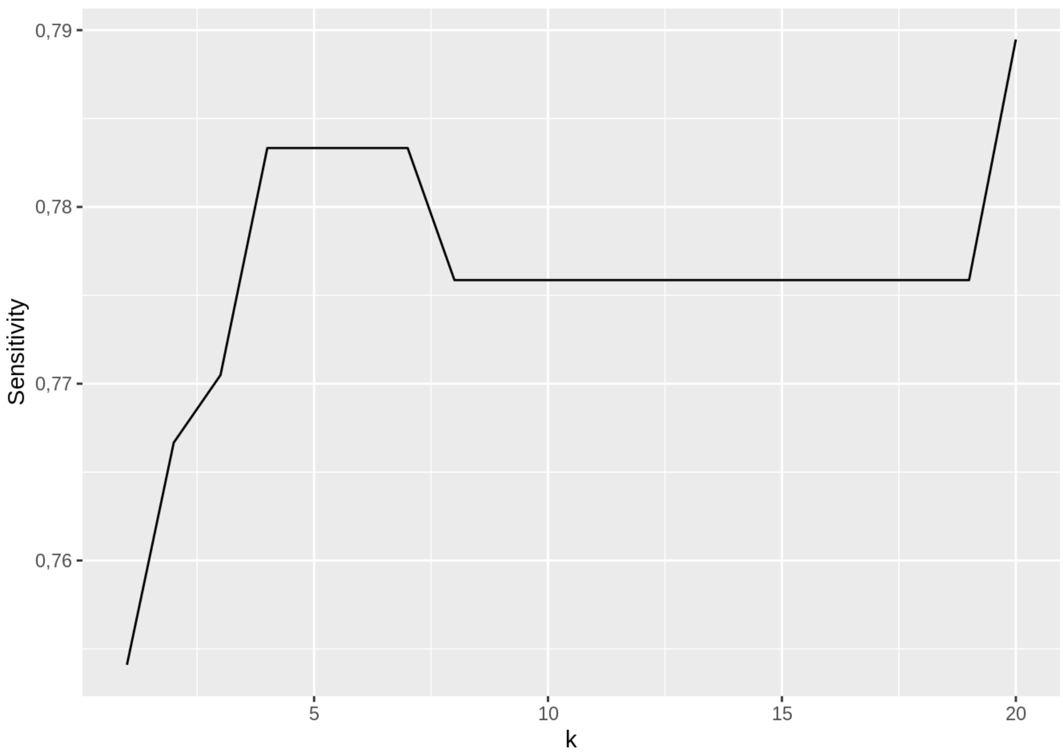



```
Sensitivity = NULL
Specifity = NULL
Precision = NULL
for(i in 1:20){
  one <- fit$pred %>% filter(k == i)
  Sensitivity <- c(Sensitivity, confusionMatrix(one$obs, one$pred)$byClass[1])
  Specifity <- c(Specifity, confusionMatrix(one$obs, one$pred)$byClass[2])
  Precision <- c(Precision, confusionMatrix(one$obs, one$pred)$byClass[5])
}
confusion <- data.frame(Sensitivity, Specifity, Precision, k = (1:20))
```

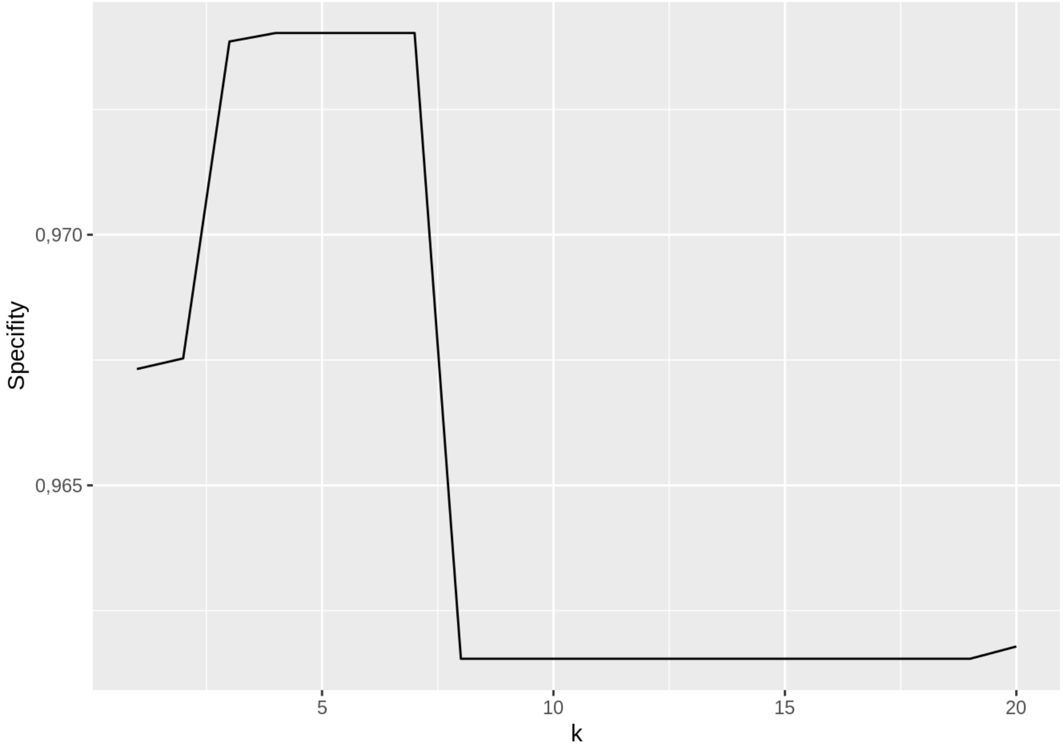
```
confusion
```

```
##    Sensitivity Specifity Precision  k
## 1    0,7540984 0,9673203 0,9019608  1
## 2    0,7666667 0,9675325 0,9019608  2
## 3    0,7704918 0,9738562 0,9215686  3
## 4    0,7833333 0,9740260 0,9215686  4
## 5    0,7833333 0,9740260 0,9215686  5
## 6    0,7833333 0,9740260 0,9215686  6
## 7    0,7833333 0,9740260 0,9215686  7
## 8    0,7758621 0,9615385 0,8823529  8
## 9    0,7758621 0,9615385 0,8823529  9
## 10   0,7758621 0,9615385 0,8823529 10
## 11   0,7758621 0,9615385 0,8823529 11
## 12   0,7758621 0,9615385 0,8823529 12
## 13   0,7758621 0,9615385 0,8823529 13
## 14   0,7758621 0,9615385 0,8823529 14
## 15   0,7758621 0,9615385 0,8823529 15
## 16   0,7758621 0,9615385 0,8823529 16
## 17   0,7758621 0,9615385 0,8823529 17
## 18   0,7758621 0,9615385 0,8823529 18
## 19   0,7758621 0,9615385 0,8823529 19
## 20   0,7894737 0,9617834 0,8823529 20
```

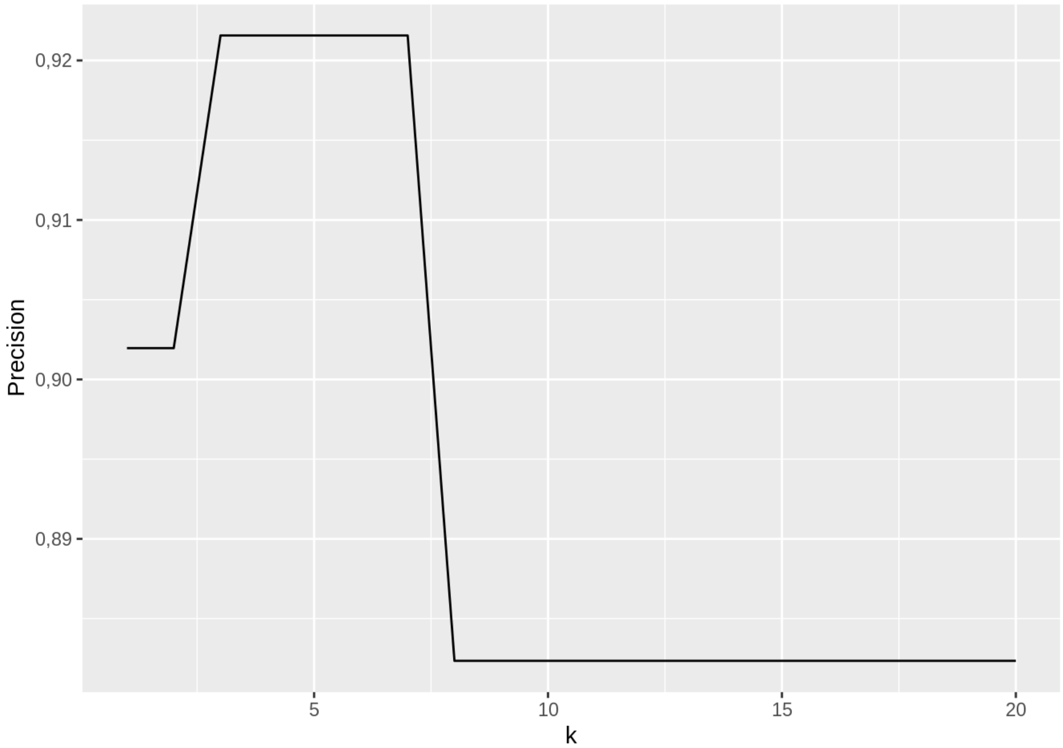
```
ggplot(confusion, aes(x = k, y = Sensitivity)) +
  geom_line()
```




```
ggplot(confusion, aes(x = k, y =  Specifity)) +
  geom_line()
```




```
ggplot(confusion, aes(x = k, y = Precision)) +
  geom_line()
```



b. Mg, Al, Ba

```
set.seed(42)
pred_knn <- knn(train = Glass[-test, c("Mg","Al","Ba")],
                test = Glass[test, c("Mg","Al","Ba")],

                cl = Glass[-test, "Type2"],
                k = 1)

table(pred_knn, Real = Glass[test, "Type2"])
```

```
##             Real
## pred_knn     Non-Window Window
##   Non-Window         11      2
##   Window              2     39
```

```
mean(pred_knn == Glass[test, "Type2"])
```

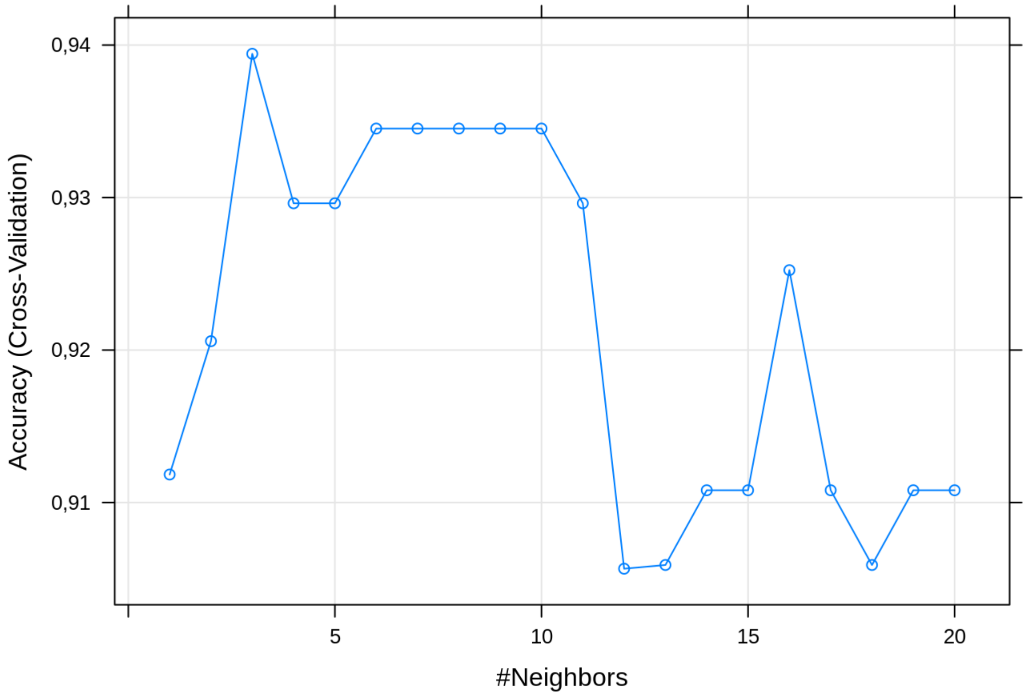
```
## [1] 0,9259259
```

```
set.seed(42)

trControl <- trainControl(method  = "cv",
                          number  = 12,
                          savePredictions = TRUE)

fit <- train(Type2 ~ Mg + Al + Ba,
             method     = "knn",
             tuneGrid   = expand.grid(k = 1:20),
             trControl  = trControl,
             metric     = "Accuracy",
             data       = Glass)
```

```
fit
```

```
## k-Nearest Neighbors
##
## 214 samples
##   3 predictor
##   2 classes: 'Non-Window', 'Window'
##
## No pre-processing
## Resampling: Cross-Validated (12 fold)
## Summary of sample sizes: 197, 196, 197, 197, 195, 196, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0,9118364  0,7518397
##    2  0,9205796  0,7707813
##    3  0,9394278  0,8341526
##    4  0,9296239  0,8043281
##    5  0,9296239  0,7991661
##    6  0,9345259  0,8078360
##    7  0,9345259  0,8078360
##    8  0,9345259  0,8078360
##    9  0,9345259  0,8078360
##   10  0,9345259  0,8109572
##   11  0,9296239  0,8012069
##   12  0,9056588  0,7456295
##   13  0,9059024  0,7517192
##   14  0,9108044  0,7599164
##   15  0,9108044  0,7599164
##   16  0,9252379  0,8002008
##   17  0,9108044  0,7630376
##   18  0,9059024  0,7455443
##   19  0,9108044  0,7604566
##   20  0,9108044  0,7570205
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 3.
```

```
plot(fit)
```

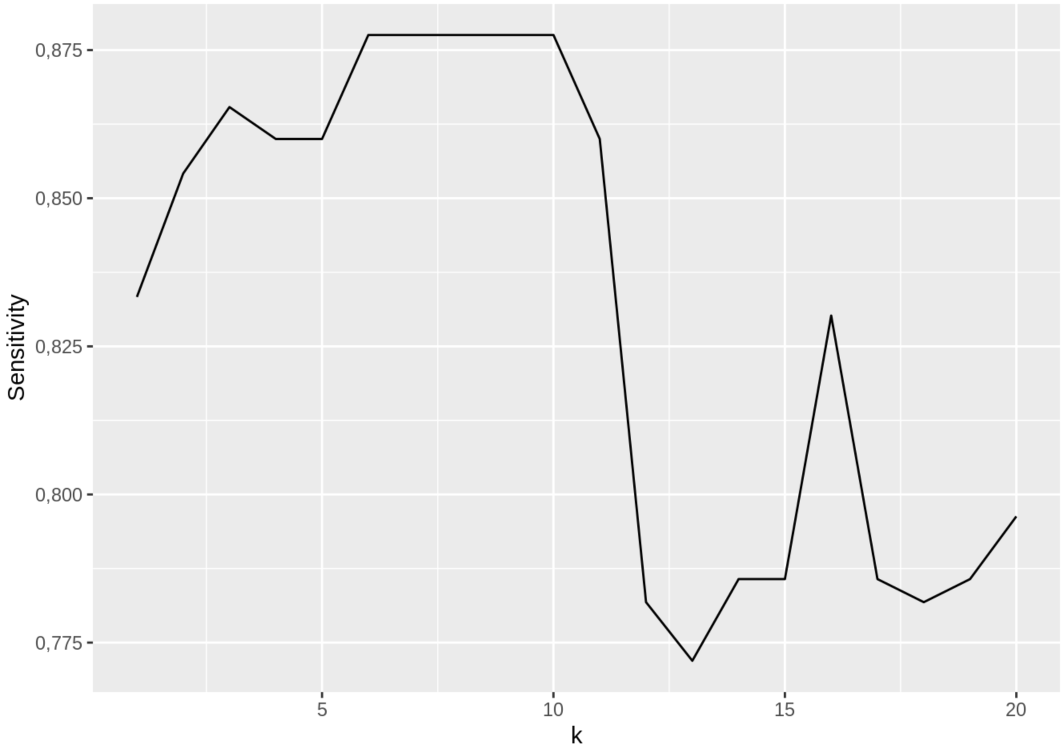



```
Sensitivity = NULL
Specifity = NULL
Precision = NULL
for(i in 1:20){
  one <- fit$pred %>% filter(k == i)
  Sensitivity <- c(Sensitivity, confusionMatrix(one$obs, one$pred)$byClass[1])
  Specifity <- c(Specifity, confusionMatrix(one$obs, one$pred)$byClass[2])
  Precision <- c(Precision, confusionMatrix(one$obs, one$pred)$byClass[5])
}
confusion <- data.frame(Sensitivity, Specifity, Precision, k = (1:20))
```

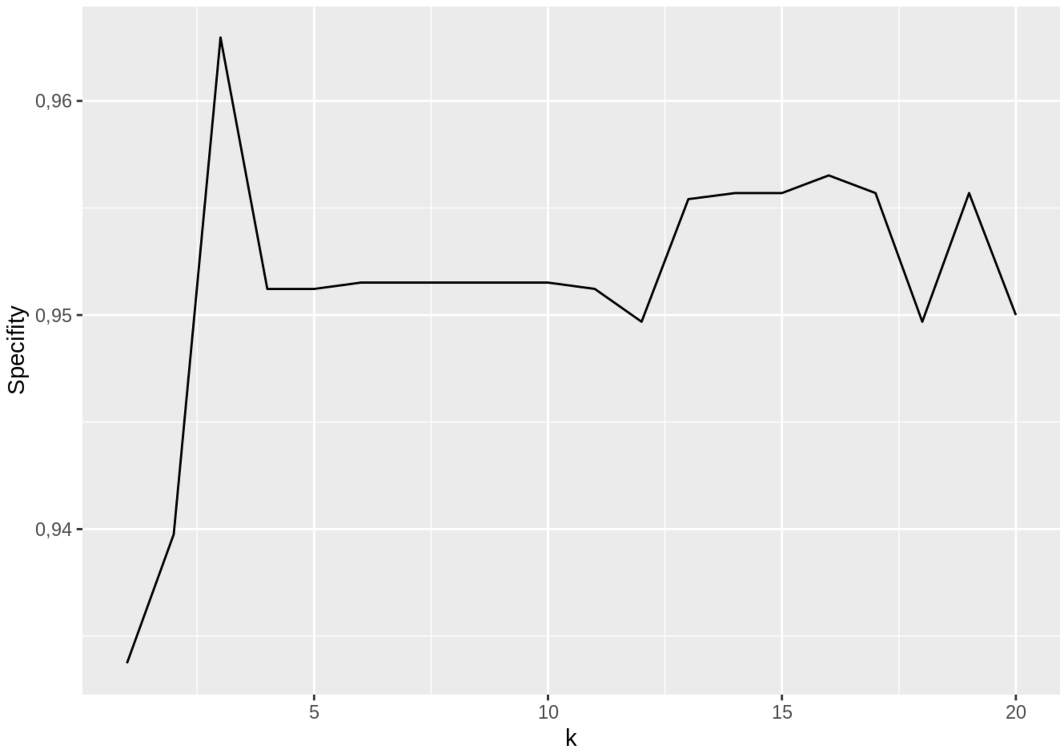
```
confusion
```

```
##    Sensitivity Specifity Precision  k
## 1    0,8333333 0,9337349 0,7843137  1
## 2    0,8541667 0,9397590 0,8039216  2
## 3    0,8653846 0,9629630 0,8823529  3
## 4    0,8600000 0,9512195 0,8431373  4
## 5    0,8600000 0,9512195 0,8431373  5
## 6    0,8775510 0,9515152 0,8431373  6
## 7    0,8775510 0,9515152 0,8431373  7
## 8    0,8775510 0,9515152 0,8431373  8
## 9    0,8775510 0,9515152 0,8431373  9
## 10   0,8775510 0,9515152 0,8431373 10
## 11   0,8600000 0,9512195 0,8431373 11
## 12   0,7818182 0,9496855 0,8431373 12
## 13   0,7719298 0,9554140 0,8627451 13
## 14   0,7857143 0,9556962 0,8627451 14
## 15   0,7857143 0,9556962 0,8627451 15
## 16   0,8301887 0,9565217 0,8627451 16
## 17   0,7857143 0,9556962 0,8627451 17
## 18   0,7818182 0,9496855 0,8431373 18
## 19   0,7857143 0,9556962 0,8627451 19
## 20   0,7962963 0,9500000 0,8431373 20
```

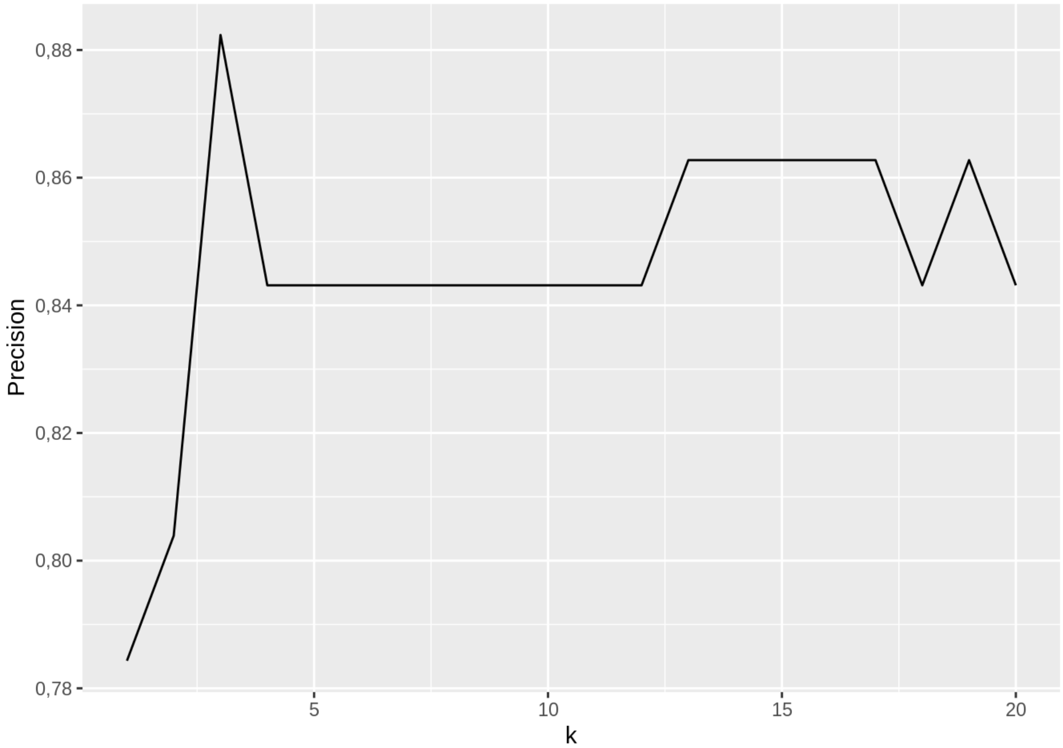
```
ggplot(confusion, aes(x = k, y = Sensitivity)) +
  geom_line()
```




```
ggplot(confusion, aes(x = k, y =  Specifity)) +
  geom_line()
```




```
ggplot(confusion, aes(x = k, y = Precision)) +
  geom_line()
```



c. All possible predictors

```
set.seed(42)
pred_knn <- knn(train = Glass[-test, c("RI","Na","Mg" ,"Al","Si", "K", "Ca","Ba","Fe")],
             test = Glass[test, c("RI","Na","Mg" ,"Al","Si", "K", "Ca","Ba","Fe")],

             cl = Glass[-test, "Type2"],
             k = 1)

table(pred_knn, Real = Glass[test, "Type2"])
```

```
##           Real
## pred_knn     Non-Window Window
##   Non-Window        12      2
##   Window             1     39
```

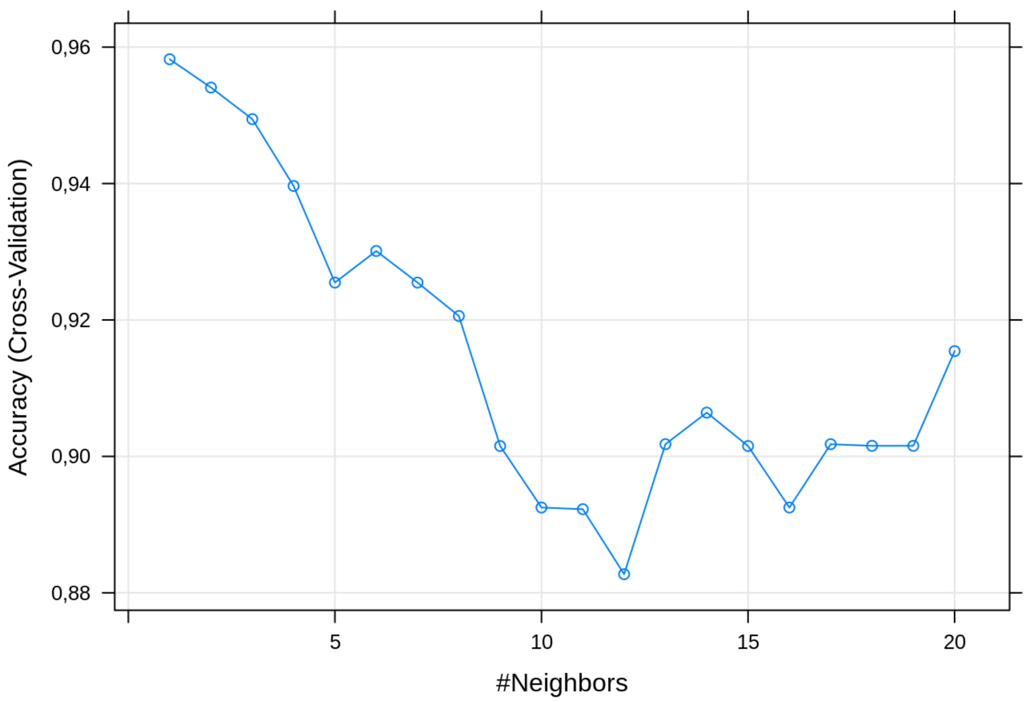
```
mean(pred_knn == Glass[test, "Type2"])
```

```
## [1] 0,9444444
```

```
set.seed(42)

trControl <- trainControl(method  = "cv",
                          number  = 12,
                          savePredictions = TRUE)

fit <- train(Type2 ~ RI + Na + Mg + Al + Si + K + Ca + Ba + Fe,
             method     = "knn",
             tuneGrid   = expand.grid(k = 1:20),
             trControl  = trControl,
             metric     = "Accuracy",
             data       = Glass)
fit
```

```
## k-Nearest Neighbors
##
## 214 samples
##   9 predictor
##   2 classes: 'Non-Window', 'Window'
##
## No pre-processing
## Resampling: Cross-Validated (12 fold)
## Summary of sample sizes: 197, 196, 197, 197, 195, 196, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0,9582187  0,8874281
##    2  0,9540764  0,8753209
##    3  0,9494467  0,8660122
##    4  0,9396428  0,8290554
##    5  0,9254816  0,7763593
##    6  0,9301112  0,7966729
##    7  0,9254816  0,7819670
##    8  0,9205796  0,7722167
##    9  0,9015165  0,7082696
##   10  0,8925009  0,6786465
##   11  0,8922572  0,6774745
##   12  0,8827256  0,6472843
##   13  0,9017888  0,7059401
##   14  0,9064184  0,7180200
##   15  0,9015165  0,7073233
##   16  0,8925009  0,6859833
##   17  0,9017888  0,7059401
##   18  0,9015451  0,7060405
##   19  0,9015451  0,7060405
##   20  0,9154340  0,7432584
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

```
plot(fit)
```

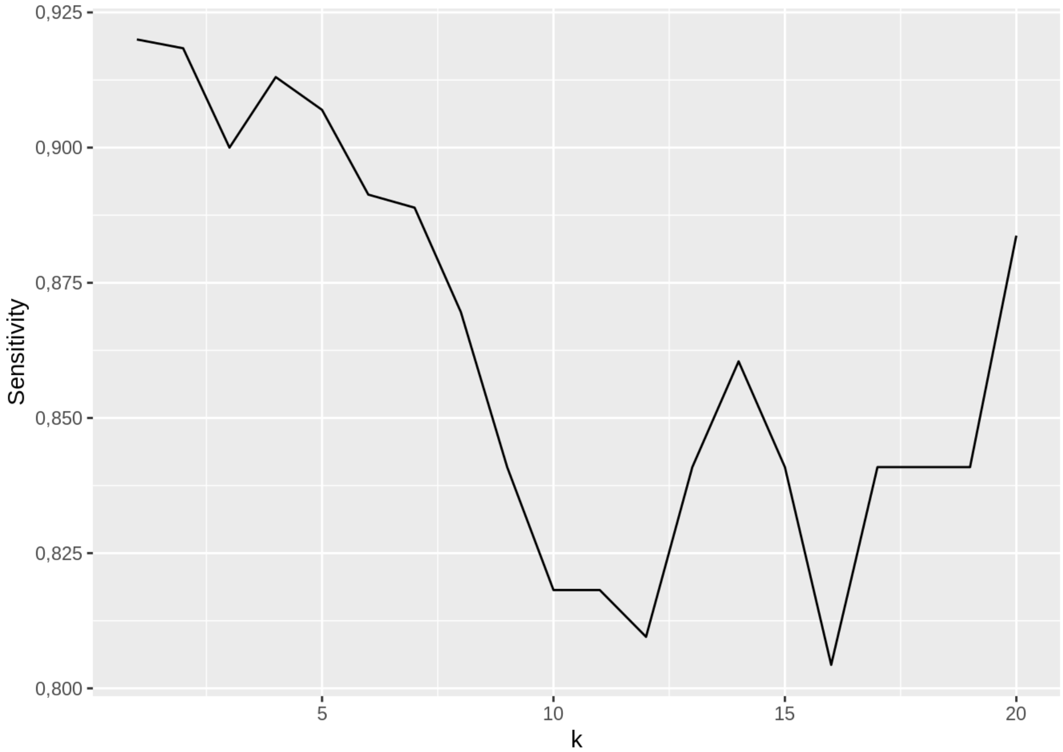



```
Sensitivity = NULL
Specifity = NULL
Precision = NULL
for(i in 1:20){
  one <- fit$pred %>% filter(k == i)
  Sensitivity <- c(Sensitivity, confusionMatrix(one$obs, one$pred)$byClass[1])
  Specifity <- c(Specifity, confusionMatrix(one$obs, one$pred)$byClass[2])
  Precision <- c(Precision, confusionMatrix(one$obs, one$pred)$byClass[5])
}
confusion <- data.frame(Sensitivity, Specifity, Precision, k = (1:20))
```

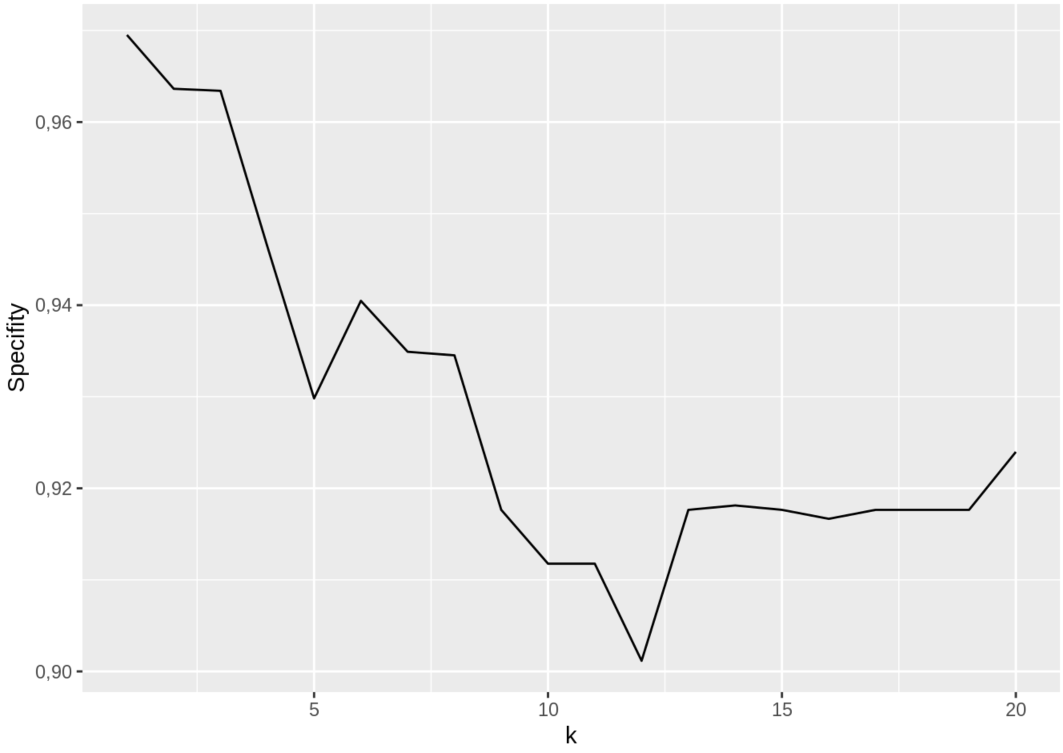
```
confusion
```

```
##    Sensitivity Specifity Precision  k
## 1    0,9200000 0,9695122 0,9019608  1
## 2    0,9183673 0,9636364 0,8823529  2
## 3    0,9000000 0,9634146 0,8823529  3
## 4    0,9130435 0,9464286 0,8235294  4
## 5    0,9069767 0,9298246 0,7647059  5
## 6    0,8913043 0,9404762 0,8039216  6
## 7    0,8888889 0,9349112 0,7843137  7
## 8    0,8695652 0,9345238 0,7843137  8
## 9    0,8409091 0,9176471 0,7254902  9
## 10   0,8181818 0,9117647 0,7058824 10
## 11   0,8181818 0,9117647 0,7058824 11
## 12   0,8095238 0,9011628 0,6666667 12
## 13   0,8409091 0,9176471 0,7254902 13
## 14   0,8604651 0,9181287 0,7254902 14
## 15   0,8409091 0,9176471 0,7254902 15
## 16   0,8043478 0,9166667 0,7254902 16
## 17   0,8409091 0,9176471 0,7254902 17
## 18   0,8409091 0,9176471 0,7254902 18
## 19   0,8409091 0,9176471 0,7254902 19
## 20   0,8837209 0,9239766 0,7450980 20
```

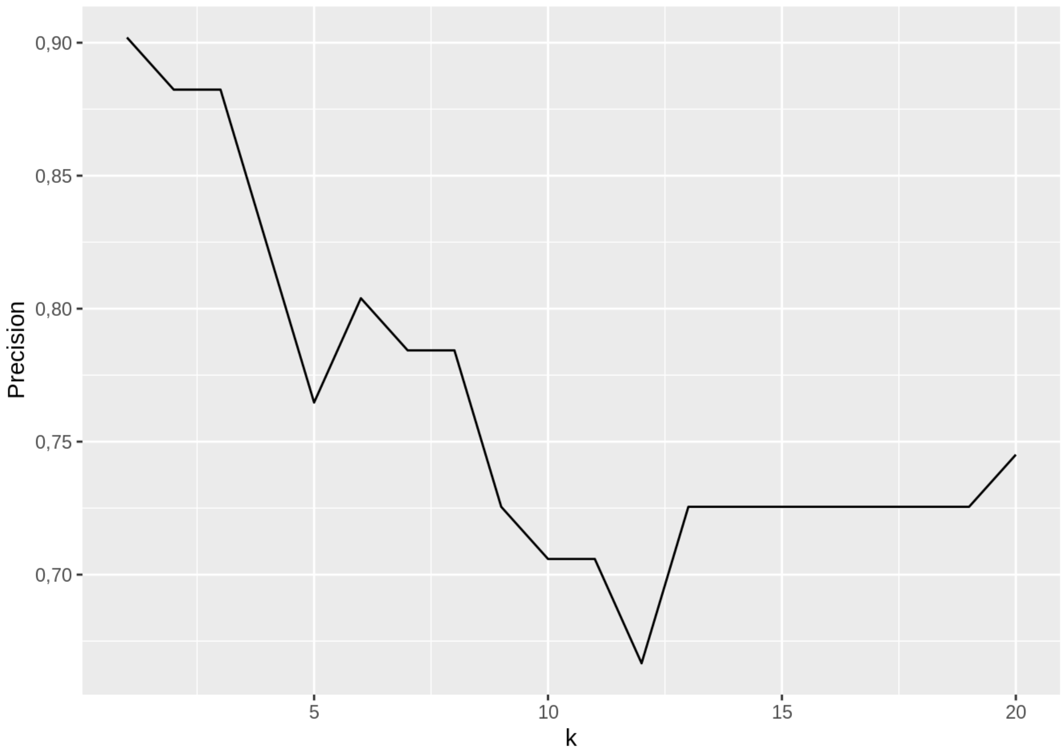
```
ggplot(confusion, aes(x = k, y = Sensitivity)) +
  geom_line()
```




```
ggplot(confusion, aes(x = k, y =  Specifity)) +
  geom_line()
```




```
ggplot(confusion, aes(x = k, y = Precision)) +
  geom_line()
```



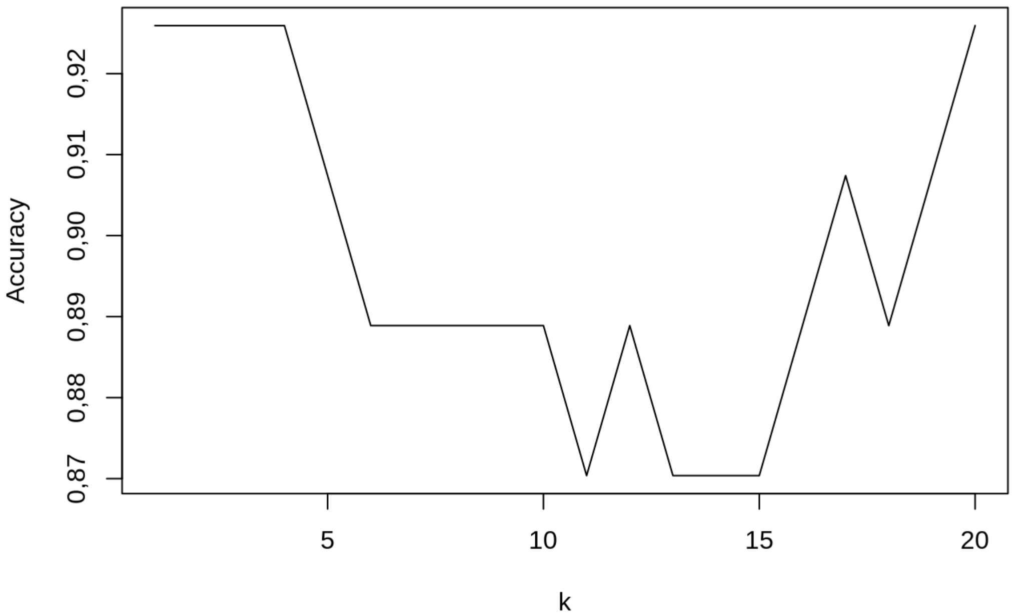
d. with PCA

```
pca.result <- prcomp(Glass[-test,-c(10:11)], scale=T)
train.pca <- pca.result$x
test.pca <- predict(pca.result, Glass[test,-c(10:11)])

k <- NULL
accuracy <- NULL
set.seed(42)
for (i in 1:20){
  pred_knn <- knn(train.pca, test.pca, Glass[-test, "Type2"], k=i)
  accuracy <- c(accuracy, mean(pred_knn == Glass[test, "Type2"]))
  k <- c(k, i)
}
result <- data.frame('k' = k, 'Accuracy' = accuracy)
```

```
result
```

```
##     k  Accuracy
## 1   1 0,9259259
## 2   2 0,9259259
## 3   3 0,9259259
## 4   4 0,9259259
## 5   5 0,9074074
## 6   6 0,8888889
## 7   7 0,8888889
## 8   8 0,8888889
## 9   9 0,8888889
## 10 10 0,8888889
## 11 11 0,8703704
## 12 12 0,8888889
## 13 13 0,8703704
## 14 14 0,8703704
## 15 15 0,8703704
## 16 16 0,8888889
## 17 17 0,9074074
## 18 18 0,8888889
## 19 19 0,9074074
## 20 20 0,9259259
```

```
plot(result, type = 'l')
```

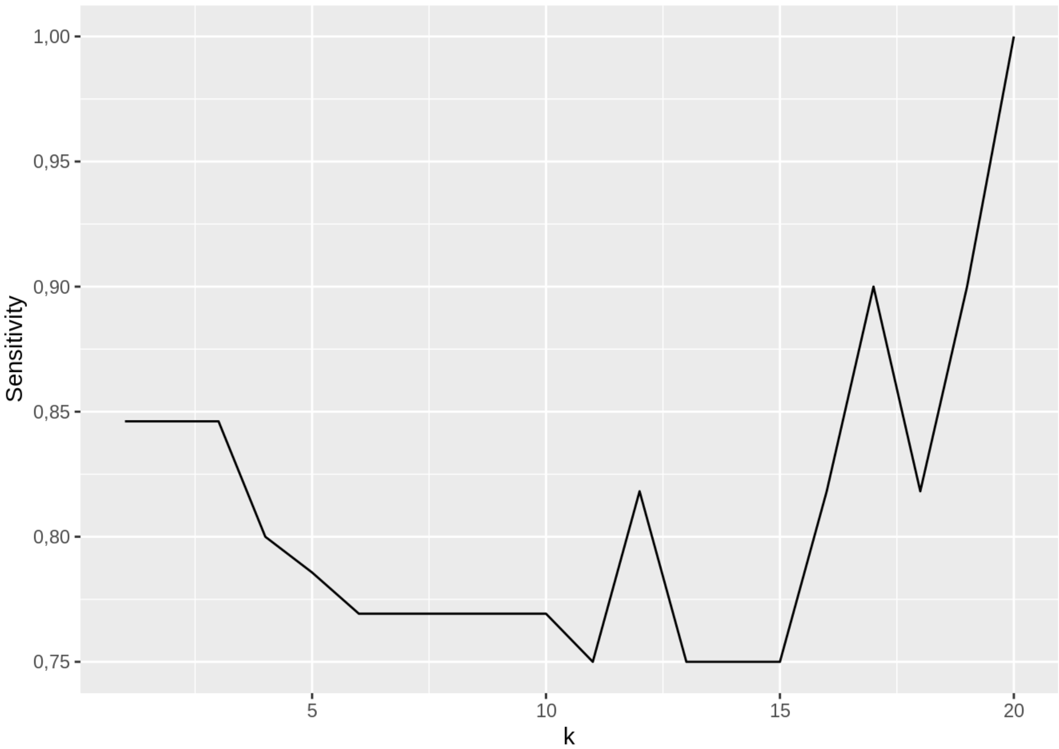



```
Sensitivity = NULL
Specifity = NULL
Precision = NULL
set.seed(42)
for (i in 1:20){
  pred_knn <- knn(train.pca, test.pca, Glass[-test, "Type2"], k=i)
  actual <- Glass[test, "Type2"]
  Sensitivity <- c(Sensitivity, confusionMatrix(actual, pred_knn)$byClass[1])
  Specifity <- c(Specifity, confusionMatrix(actual, pred_knn)$byClass[2])
  Precision <- c(Precision, confusionMatrix(actual, pred_knn)$byClass[5])
}
confusion <- data.frame(Sensitivity, Specifity, Precision, k = (1:20))
```

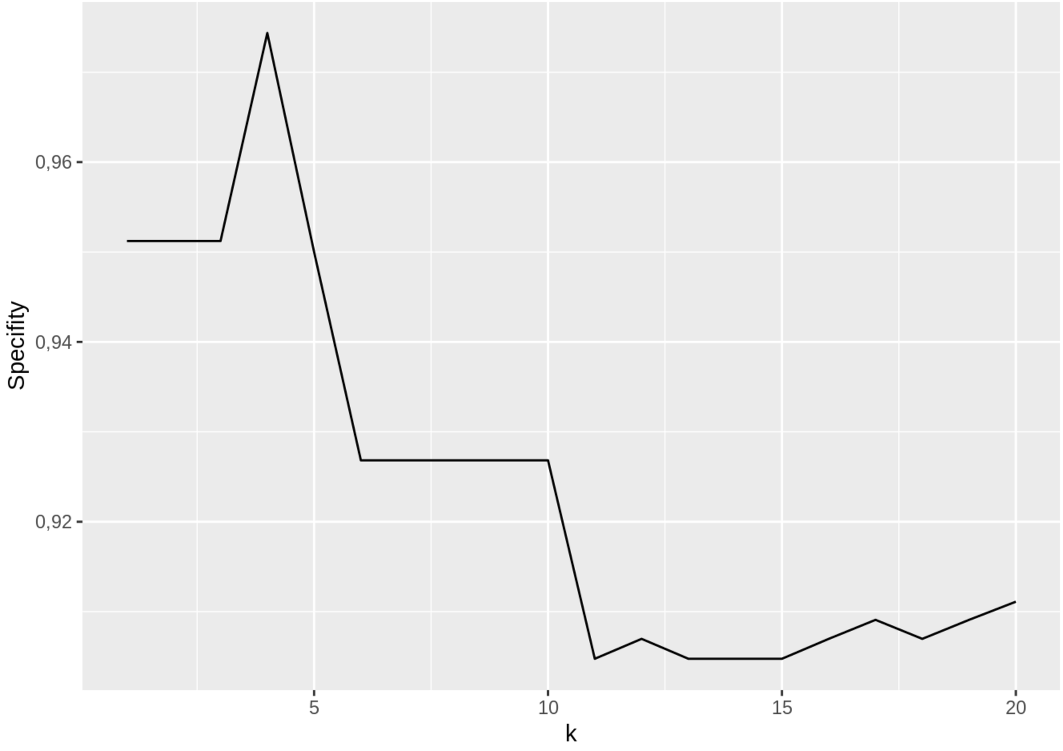
```
confusion
```

```
##    Sensitivity Specifity Precision  k
## 1    0,8461538 0,9512195 0,8461538  1
## 2    0,8461538 0,9512195 0,8461538  2
## 3    0,8461538 0,9512195 0,8461538  3
## 4    0,8000000 0,9743590 0,9230769  4
## 5    0,7857143 0,9500000 0,8461538  5
## 6    0,7692308 0,9268293 0,7692308  6
## 7    0,7692308 0,9268293 0,7692308  7
## 8    0,7692308 0,9268293 0,7692308  8
## 9    0,7692308 0,9268293 0,7692308  9
## 10   0,7692308 0,9268293 0,7692308 10
## 11   0,7500000 0,9047619 0,6923077 11
## 12   0,8181818 0,9069767 0,6923077 12
## 13   0,7500000 0,9047619 0,6923077 13
## 14   0,7500000 0,9047619 0,6923077 14
## 15   0,7500000 0,9047619 0,6923077 15
## 16   0,8181818 0,9069767 0,6923077 16
## 17   0,9000000 0,9090909 0,6923077 17
## 18   0,8181818 0,9069767 0,6923077 18
## 19   0,9000000 0,9090909 0,6923077 19
## 20   1,0000000 0,9111111 0,6923077 20
```

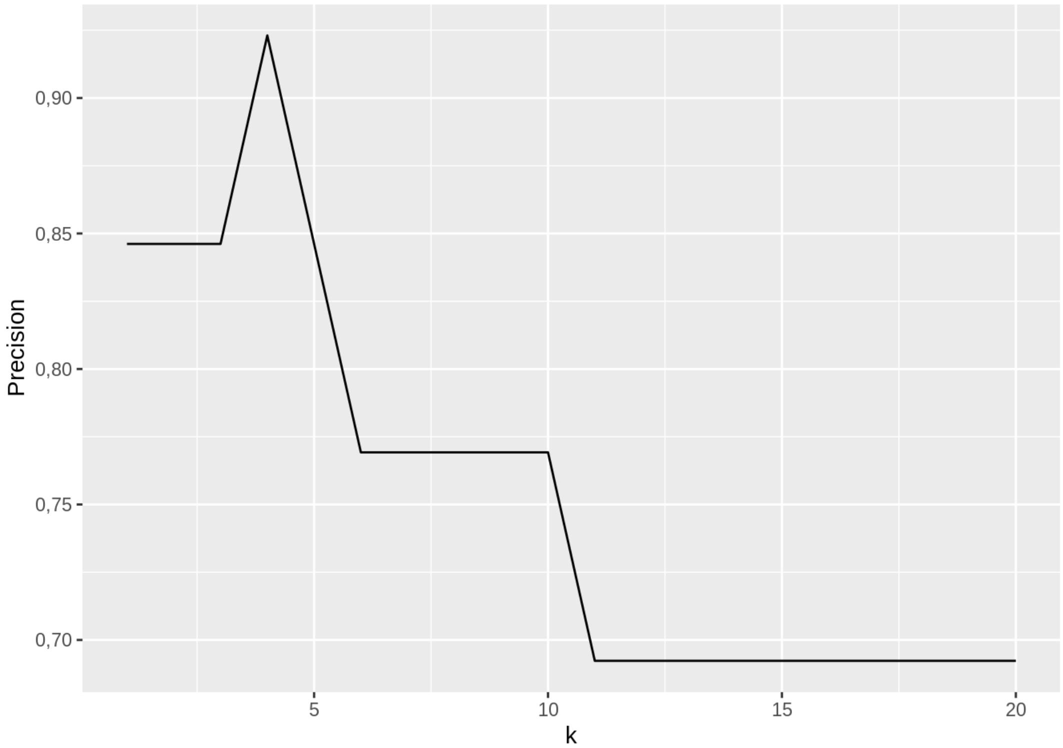
```
ggplot(confusion, aes(x = k, y = Sensitivity)) +
  geom_line()
```




```
ggplot(confusion, aes(x = k, y =  Specifity)) +
  geom_line()
```




```
ggplot(confusion, aes(x = k, y = Precision)) +
  geom_line()
```




```
##            Model  Accuracy Sensitivity Specifity Precision
## 1             Mg 0,9208806   0,7833333 0,9740260 0,9215686
## 2   Mg + Al + Ba 0,9394278   0,8653846 0,9629630 0,8823529
## 3 All predictors 0,9582187   0,9200000 0,9695122 0,9019608
## 4            PCA 0,9259259   0,8000000 0,9743590 0,9230769
```

3. Logistic regression

"RI"

```
glm_fit_ri <- glm(Type2 ~ RI, data = Glass,
                  family = "binomial")
summary(glm_fit_ri)
```

```
##
## Call:
## glm(formula = Type2 ~ RI, family = "binomial", data = Glass)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2,0040   0,4244   0,7257   0,7738   1,0046
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -188,62      97,78  -1,929   0,0537 .
## RI            125,02      64,42   1,941   0,0523 .
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,03  on 213  degrees of freedom
## Residual deviance: 230,70  on 212  degrees of freedom
## AIC: 234,7
##
## Number of Fisher Scoring iterations: 4
```

“Na”

```
glm_fit_na <- glm(Type2 ~ Na, data = Glass,
                  family = "binomial")
summary(glm_fit_na)
```

```
##
## Call:
## glm(formula = Type2 ~ Na, family = "binomial", data = Glass)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3,3057   0,1144   0,4661   0,6407   1,5963
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  23,9071     3,9084   6,117 9,54e-10 ***
## Na           -1,6725     0,2847  -5,874 4,25e-09 ***
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,03  on 213  degrees of freedom
## Residual deviance: 186,33  on 212  degrees of freedom
## AIC: 190,33
##
## Number of Fisher Scoring iterations: 5
```

“Mg”

```
glm_fit_mg <- glm(Type2 ~ Mg, data = Glass,
                  family = "binomial")
summary(glm_fit_mg)
```

```
##
## Call:
## glm(formula = Type2 ~ Mg, family = "binomial", data = Glass)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2,3583   0,2363   0,3019   0,3260   1,9983
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1,8507     0,4113  -4,500  6,8e-06 ***
## Mg            1,3675     0,1644   8,321  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,03  on 213  degrees of freedom
## Residual deviance: 116,52  on 212  degrees of freedom
## AIC: 120,52
##
## Number of Fisher Scoring iterations: 5
```

“Al”

```
glm_fit_al <- glm(Type2 ~ Al, data = Glass,
                  family = "binomial")
summary(glm_fit_al)
```

```
##
## Call:
## glm(formula = Type2 ~ Al, family = "binomial", data = Glass)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3,5480   0,0815   0,3686   0,5699   1,6877
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   7,7136     1,0776   7,158 8,17e-13 ***
## Al           -4,1804     0,6595  -6,338 2,32e-10 ***
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,03  on 213  degrees of freedom
## Residual deviance: 151,67  on 212  degrees of freedom
## AIC: 155,67
##
## Number of Fisher Scoring iterations: 6
```

“Si”

```
glm_fit_si <- glm(Type2 ~ Si, data = Glass,
                  family = "binomial")
summary(glm_fit_si)
```

```
##
## Call:
## glm(formula = Type2 ~ Si, family = "binomial", data = Glass)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2,3017   0,4670   0,6968   0,7760   1,0521
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  37,4232    17,0421   2,196   0,0281 *
## Si           -0,4986     0,2341  -2,130   0,0332 *
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,03  on 213  degrees of freedom
## Residual deviance: 230,04  on 212  degrees of freedom
## AIC: 234,04
##
## Number of Fisher Scoring iterations: 4
```

“K”

```
glm_fit_k <- glm(Type2 ~ K, data = Glass,
                 family = "binomial")
summary(glm_fit_k)
```

```
##
## Call:
## glm(formula = Type2 ~ K, family = "binomial", data = Glass)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1,7323   0,7105   0,7385   0,7426   0,7701
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1,2480     0,1978   6,310  2,8e-10 ***
## K            -0,1677     0,2190  -0,766    0,444
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,03  on 213  degrees of freedom
## Residual deviance: 234,47  on 212  degrees of freedom
## AIC: 238,47
##
## Number of Fisher Scoring iterations: 4
```

“Ca”

```
glm_fit_ca <- glm(Type2 ~ Ca, data = Glass,
                  family = "binomial")
summary(glm_fit_ca)
```

```
##
## Call:
## glm(formula = Type2 ~ Ca, family = "binomial", data = Glass)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1,7962   0,7133   0,7250   0,7337   0,8984
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1,7400     0,9908   1,756   0,0791 .
## Ca           -0,0643     0,1084  -0,593   0,5529
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,03  on 213  degrees of freedom
## Residual deviance: 234,69  on 212  degrees of freedom
## AIC: 238,69
##
## Number of Fisher Scoring iterations: 4
```

“Ba”

```
glm_fit_ba <- glm(Type2 ~ Ba, data = Glass,
               family = "binomial")
summary(glm_fit_ba)
```

```
##
## Call:
## glm(formula = Type2 ~ Ba, family = "binomial", data = Glass)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1,9591   0,5634   0,5634   0,5634   4,2473
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1,7604     0,2060   8,544  < 2e-16 ***
## Ba           -3,4223     0,7271  -4,707 2,52e-06 ***
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,03  on 213  degrees of freedom
## Residual deviance: 177,38  on 212  degrees of freedom
## AIC: 181,38
##
## Number of Fisher Scoring iterations: 6
```

“Fe”

```
glm_fit_fe <- glm(Type2 ~ Fe, data = Glass,
               family = "binomial")
summary(glm_fit_fe)
```

```
##
## Call:
## glm(formula = Type2 ~ Fe, family = "binomial", data = Glass)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2,9563   0,2878   0,6363   0,8327   0,8327
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0,8809     0,1782   4,942 7,73e-07 ***
## Fe            6,8160     2,5385   2,685  0,00725 **
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,03  on 213  degrees of freedom
## Residual deviance: 224,93  on 212  degrees of freedom
## AIC: 228,93
##
## Number of Fisher Scoring iterations: 5
```

“All predictors”

```
glm_fit_all <- glm(Type2 ~ RI + Na + Mg + Al + Si + K + Ca + Ba + Fe , data = Glass,
                family = "binomial")
summary(glm_fit_all)
```

```
##
## Call:
## glm(formula = Type2 ~ RI + Na + Mg + Al + Si + K + Ca + Ba +
##     Fe, family = "binomial", data = Glass)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2,30170   0,00101   0,03612   0,12396   1,63400
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3292,972   1102,056   2,988  0,00281 **
## RI          -1232,505    444,483  -2,773  0,00556 **
## Na            -12,635      6,120  -2,065  0,03897 *
## Mg             -8,334      5,802  -1,437  0,15086
## Al            -19,869      7,060  -2,814  0,00489 **
## Si            -15,269      6,688  -2,283  0,02242 *
## K             -10,987      6,306  -1,742  0,08146 .
## Ca             -9,157      5,867  -1,561  0,11859
## Ba            -11,206      6,106  -1,835  0,06646 .
## Fe              6,654      5,819   1,143  0,25285
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,029  on 213  degrees of freedom
## Residual deviance:  44,079  on 204  degrees of freedom
## AIC: 64,079
##
## Number of Fisher Scoring iterations: 9
```

“Best”

```
glm_fit_best <- glm(Type2 ~ Mg + Al + Na , data = Glass,
                family = "binomial")
summary(glm_fit_best)
```

```
##
## Call:
## glm(formula = Type2 ~ Mg + Al + Na, family = "binomial", data = Glass)
##
## Deviance Residuals:
##      Min        1Q    Median       3Q       Max
## -2,72309   0,06959   0,16101   0,26506   1,93821
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  18,4457     4,8212   3,826 0,000130 ***
## Mg            1,3963     0,2345   5,953 2,63e-09 ***
## Al           -3,3514     0,8428  -3,976 7,00e-05 ***
## Na           -1,1387     0,3341  -3,408 0,000654 ***
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,029  on 213  degrees of freedom
## Residual deviance:  72,425  on 210  degrees of freedom
## AIC: 80,425
##
## Number of Fisher Scoring iterations: 7
```

Transfromation

```
one <- Glass$Mg^2
two <-  Glass$Al^2
three <- Glass$Na^2

glm_fit_transf <- glm(Type2 ~ one + two + three , data = Glass,
                  family = "binomial")
summary(glm_fit_transf)
```

```
##
## Call:
## glm(formula = Type2 ~ one + two + three, family = "binomial",
##     data = Glass)
##
## Deviance Residuals:
##      Min        1Q    Median       3Q       Max
## -2,81717   0,06331   0,11715   0,20020   1,85976
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  7,75316    2,43188   3,188 0,001432 **
## one          0,40140    0,06895   5,822 5,81e-09 ***
## two         -1,03728    0,29863  -3,473 0,000514 ***
## three       -0,03716    0,01276  -2,911 0,003600 **
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235,029  on 213  degrees of freedom
## Residual deviance:  64,793  on 210  degrees of freedom
## AIC: 72,793
##
## Number of Fisher Scoring iterations: 7
```

```r
set.seed(42)
Glass$Mg <- Glass$Mg^2
Glass$Al <-  Glass$Al^2
Glass$Na <- Glass$Na^2

glm_fit_transf_2 <- glm(Type2 ~ Mg + Al + Na , data = Glass[-test,],
                  family = "binomial")
pred_glm <- predict(glm_fit_transf_2, type = "response", newdata = Glass[test, ]) > .5
table(pred_glm, Real = Glass[test, 11])
```

```
##        Real
## pred_glm Non-Window Window
##    FALSE         12      3
##    TRUE           1     38
```

```r
50/54
```

```
## [1] 0,9259259
```