Data mining

Project part 1 and 2

Beata Paszkowska Karolina Ostrowska

24 January 2022

Problem description

For this project, we analyze the Churn data set. The term “churn” is used to describe customer dropping services of one company in favor of the other.

We want to inspect what factors matters in terms of churning. By finding patterns and figuring out which variables are significant, we hope to create a prediction model that will help minimize the amount of customers churning.

Data characteristics

The set contains details about 3333 customers, for each of them there are gathered 20 predictors as well as information if they churned.

Looking at our data below, we can see they are mostly continuous. There are two binary columns - International Plan and Voice Mail Plan.

```r
head(data)
```

```
##   State Account.Length Area.Code    Phone Int.l.Plan VMail.Plan VMail.Message
## 1    KS            128       415 382-4657         no        yes            25
## 2    OH            107       415 371-7191         no        yes            26
## 3    NJ            137       415 358-1921         no         no             0
## 4    OH             84       408 375-9999        yes         no             0
## 5    OK             75       415 330-6626        yes         no             0
## 6    AL            118       510 391-8027        yes         no             0
##     Day.Mins Day.Calls Day.Charge   Eve.Mins Eve.Calls Eve.Charge Night.Mins
## 1 265.100000       110  45.070000 197.400000        99  16.780000 244.700000
## 2 161.600000       123  27.470000 195.500000       103  16.620000 254.400000
## 3 243.400000       114  41.380000 121.200000       110  10.300000 162.600000
## 4 299.400000        71  50.900000  61.900000        88   5.260000 196.900000
## 5 166.700000       113  28.340000 148.300000       122  12.610000 186.900000
## 6 223.400000        98  37.980000 220.600000       101  18.750000 203.900000
##   Night.Calls Night.Charge Intl.Mins Intl.Calls Intl.Charge CustServ.Calls
## 1          91    11.010000 10.000000          3    2.700000              1
## 2         103    11.450000 13.700000          3    3.700000              1
## 3         104     7.320000 12.200000          5    3.290000              0
## 4          89     8.860000  6.600000          7    1.780000              2
## 5         121     8.410000 10.100000          3    2.730000              3
## 6         118     9.180000  6.300000          6    1.700000              0
##   Churn.
## 1 False.
## 2 False.
## 3 False.
## 4 False.
## 5 False.
## 6 False.
```

```r
sapply(data, class)
```

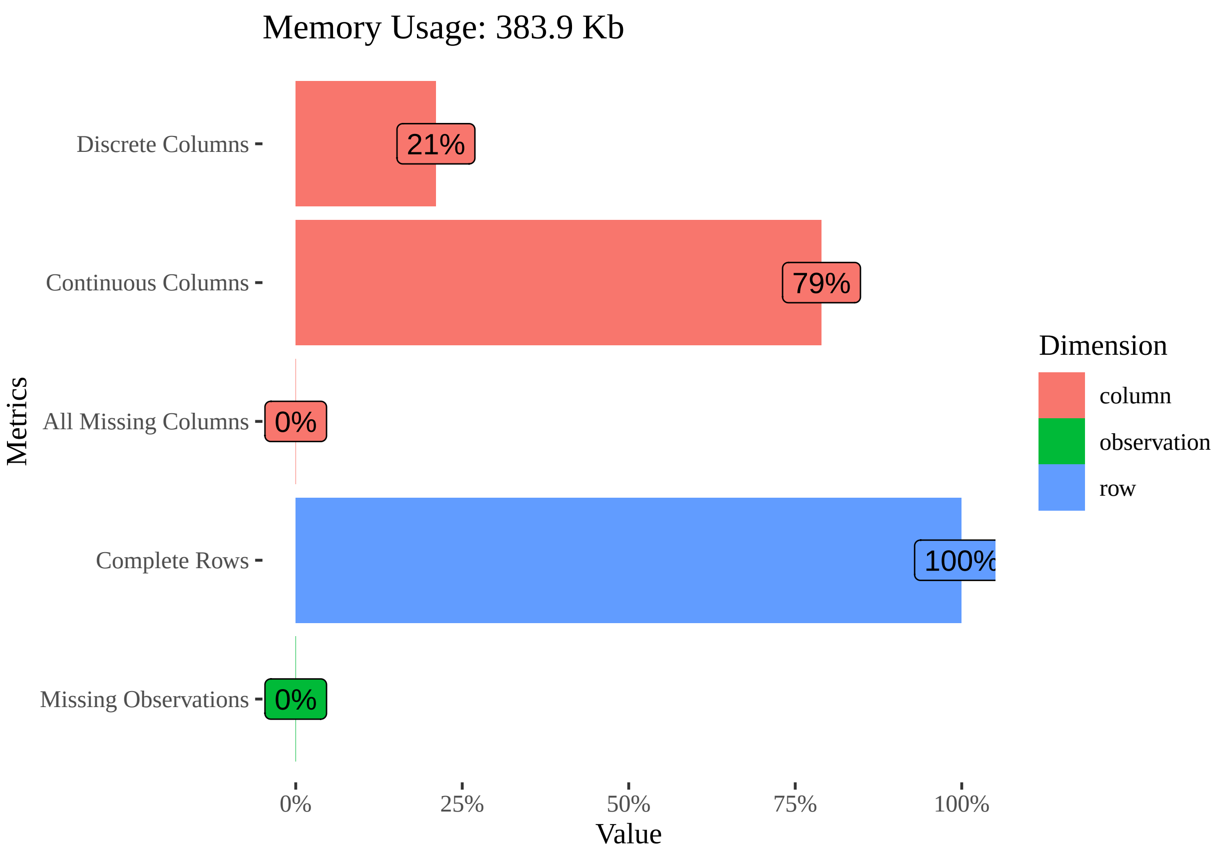
```
##          State Account.Length      Area.Code          Phone     Int.l.Plan
##       "factor"      "integer"      "integer"       "factor"       "factor"
##     VMail.Plan  VMail.Message       Day.Mins      Day.Calls     Day.Charge
##       "factor"      "integer"       "factor"      "integer"       "factor"
##       Eve.Mins      Eve.Calls     Eve.Charge     Night.Mins    Night.Calls
##       "factor"      "integer"       "factor"       "factor"      "integer"
##   Night.Charge      Intl.Mins     Intl.Calls    Intl.Charge CustServ.Calls
##       "factor"       "factor"      "integer"       "factor"      "integer"
##         Churn.
##       "factor"
```

Firstly, we check if the types of data are properly recognized and changed them accordingly if needed. We also drop columns "State" and "Phone" as it most probably won't give much insight into our analysis.

After such cleaning, data's metrics presents as fallows:

```
sapply(data, class)
```

```
## Account.Length      Area.Code     Int.l.Plan     VMail.Plan  VMail.Message
##      "integer"    "character"       "factor"       "factor"      "integer"
##       Day.Mins      Day.Calls     Day.Charge       Eve.Mins      Eve.Calls
##      "numeric"      "integer"      "numeric"      "numeric"      "integer"
##     Eve.Charge     Night.Mins    Night.Calls   Night.Charge      Intl.Mins
##      "numeric"      "numeric"      "integer"      "numeric"      "numeric"
##     Intl.Calls    Intl.Charge CustServ.Calls         Churn.
##      "numeric"      "numeric"      "integer"       "factor"
```



There is no missing observation and all the rows are complete. As we noticed before, over three fourth of data are continuous, with only 21% of them being discrete.

Here we can see basics statistics for each column. We can mark that almost in every case the median is very close to mean. The average amount of calls doesn't really depend on the time of day, but they tend to be a bit shorter during day than evening and night.

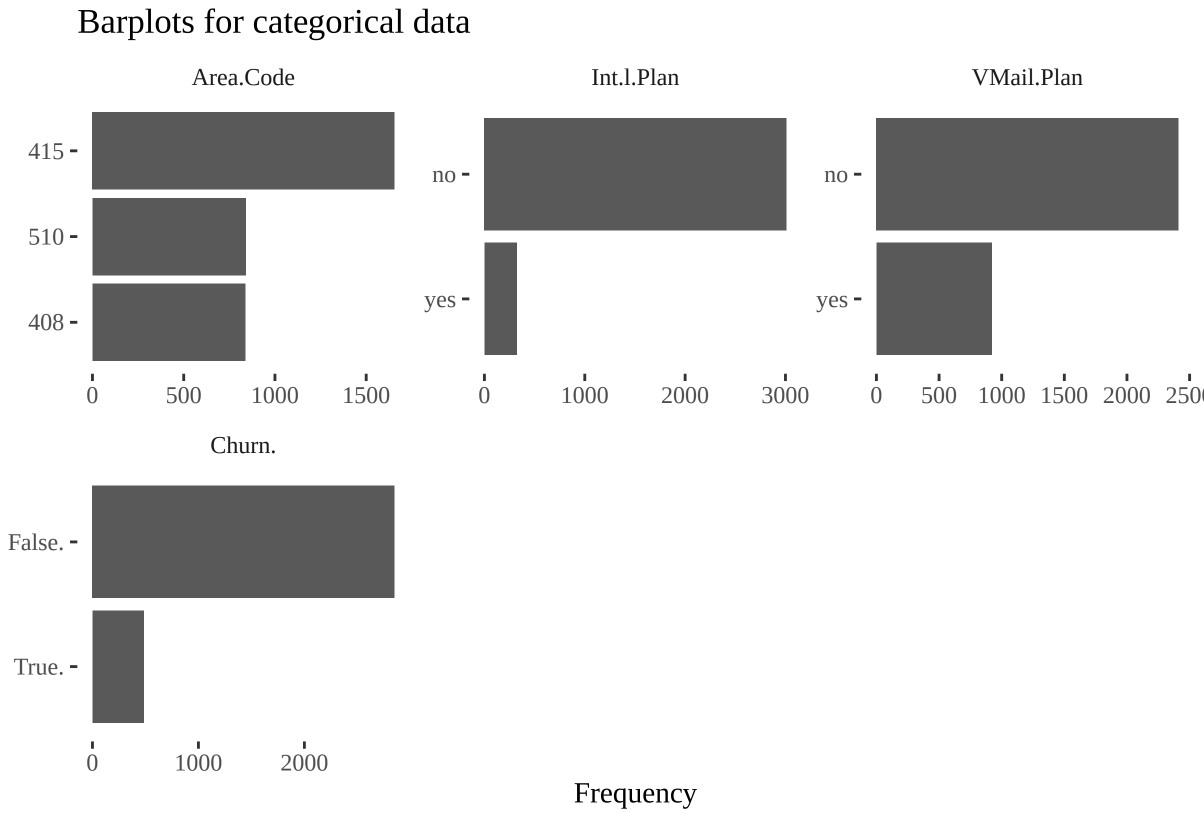
```r
summary(data)
```

```
##  Account.Length   Area.Code         Int.l.Plan VMail.Plan VMail.Message
##  Min.   :  1.0   Length:3333        no :3010   no :2411   Min.   : 0.000
##  1st Qu.: 74.0   Class :character   yes: 323   yes: 922   1st Qu.: 0.000
##  Median :101.0   Mode  :character                         Median : 0.000
##  Mean   :101.1                                            Mean   : 8.099
##  3rd Qu.:127.0                                            3rd Qu.:20.000
##  Max.   :243.0                                            Max.   :51.000
##     Day.Mins       Day.Calls       Day.Charge       Eve.Mins
##  Min.   :  0.0   Min.   :  0.0   Min.   : 0.00   Min.   :  0.0
##  1st Qu.:143.7   1st Qu.: 87.0   1st Qu.:24.43   1st Qu.:166.6
##  Median :179.4   Median :101.0   Median :30.50   Median :201.4
##  Mean   :179.8   Mean   :100.4   Mean   :30.56   Mean   :201.0
##  3rd Qu.:216.4   3rd Qu.:114.0   3rd Qu.:36.79   3rd Qu.:235.3
##  Max.   :350.8   Max.   :165.0   Max.   :59.64   Max.   :363.7
##    Eve.Calls       Eve.Charge      Night.Mins     Night.Calls
##  Min.   :  0.0   Min.   : 0.00   Min.   : 23.2   Min.   : 33.0
##  1st Qu.: 87.0   1st Qu.:14.16   1st Qu.:167.0   1st Qu.: 87.0
##  Median :100.0   Median :17.12   Median :201.2   Median :100.0
##  Mean   :100.1   Mean   :17.08   Mean   :200.9   Mean   :100.1
##  3rd Qu.:114.0   3rd Qu.:20.00   3rd Qu.:235.3   3rd Qu.:113.0
##  Max.   :170.0   Max.   :30.91   Max.   :395.0   Max.   :175.0
##   Night.Charge      Intl.Mins       Intl.Calls      Intl.Charge
##  Min.   : 1.040   Min.   : 0.00   Min.   : 0.000   Min.   :0.000
##  1st Qu.: 7.520   1st Qu.: 8.50   1st Qu.: 3.000   1st Qu.:2.300
##  Median : 9.050   Median :10.30   Median : 4.000   Median :2.780
##  Mean   : 9.039   Mean   :10.24   Mean   : 4.479   Mean   :2.765
##  3rd Qu.:10.590   3rd Qu.:12.10   3rd Qu.: 6.000   3rd Qu.:3.270
##  Max.   :17.770   Max.   :20.00   Max.   :20.000   Max.   :5.400
##  CustServ.Calls     Churn.
##  Min.   :0.000   False.:2850
##  1st Qu.:1.000   True. : 483
##  Median :1.000
##  Mean   :1.563
##  3rd Qu.:2.000
##  Max.   :9.000
```

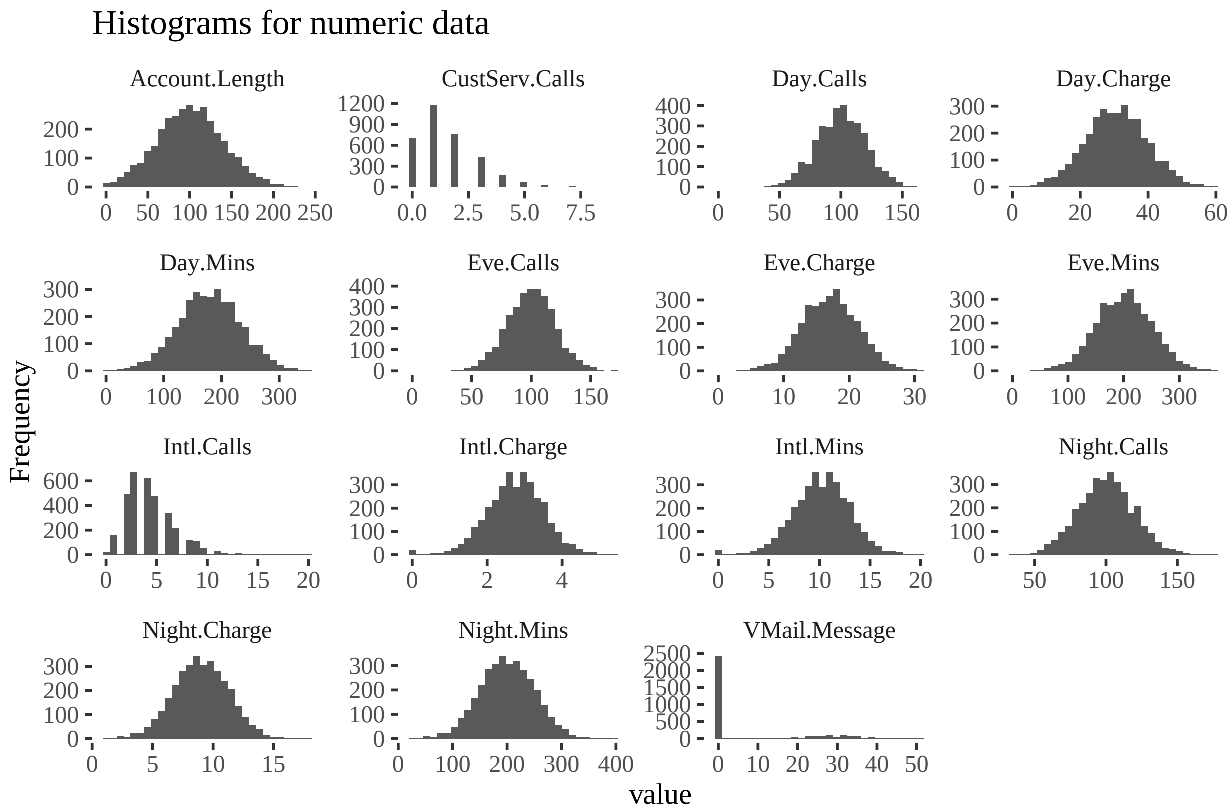
Plots

The data mostly consists of the cases in which customer didn't churn. We also see that both international and voice mail plan isn't a common thing for clients.

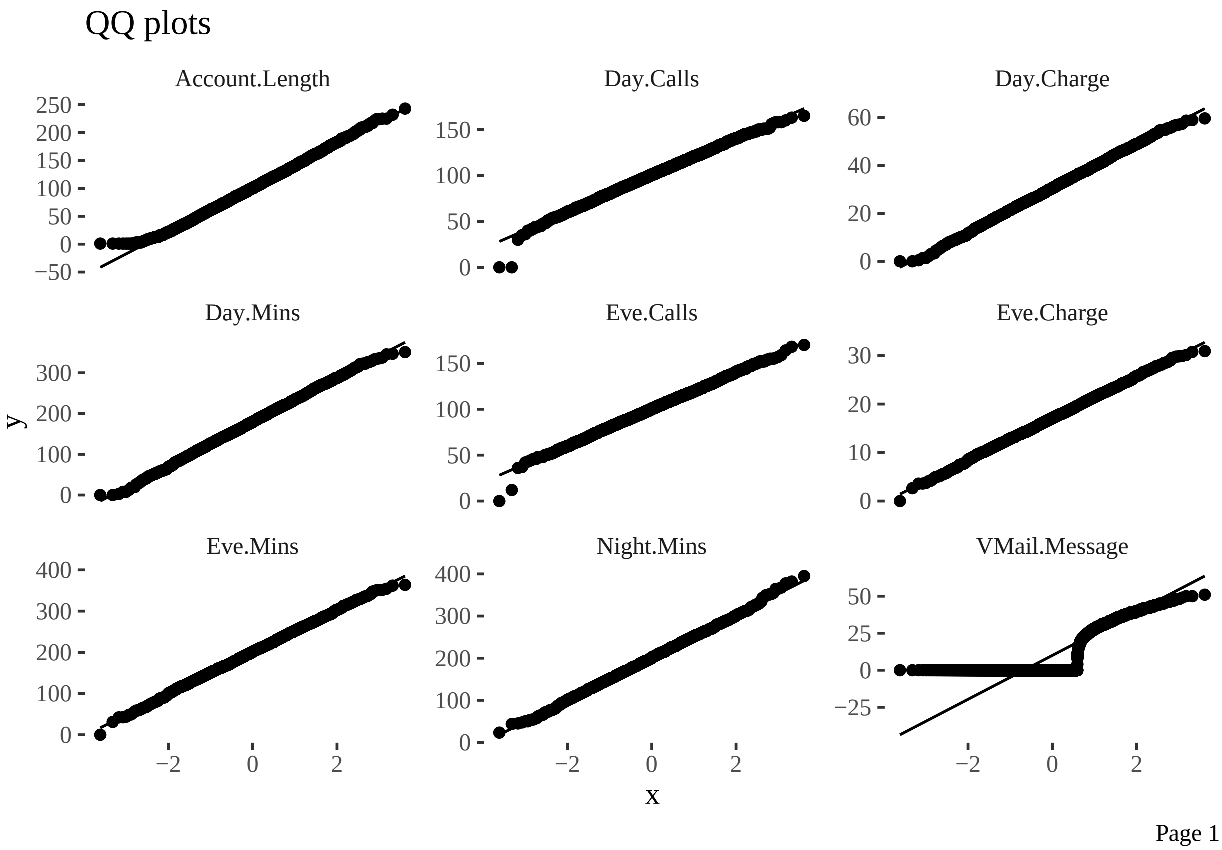
Something we noted is that there are only 3 area codes, even though there were a lot of different states.

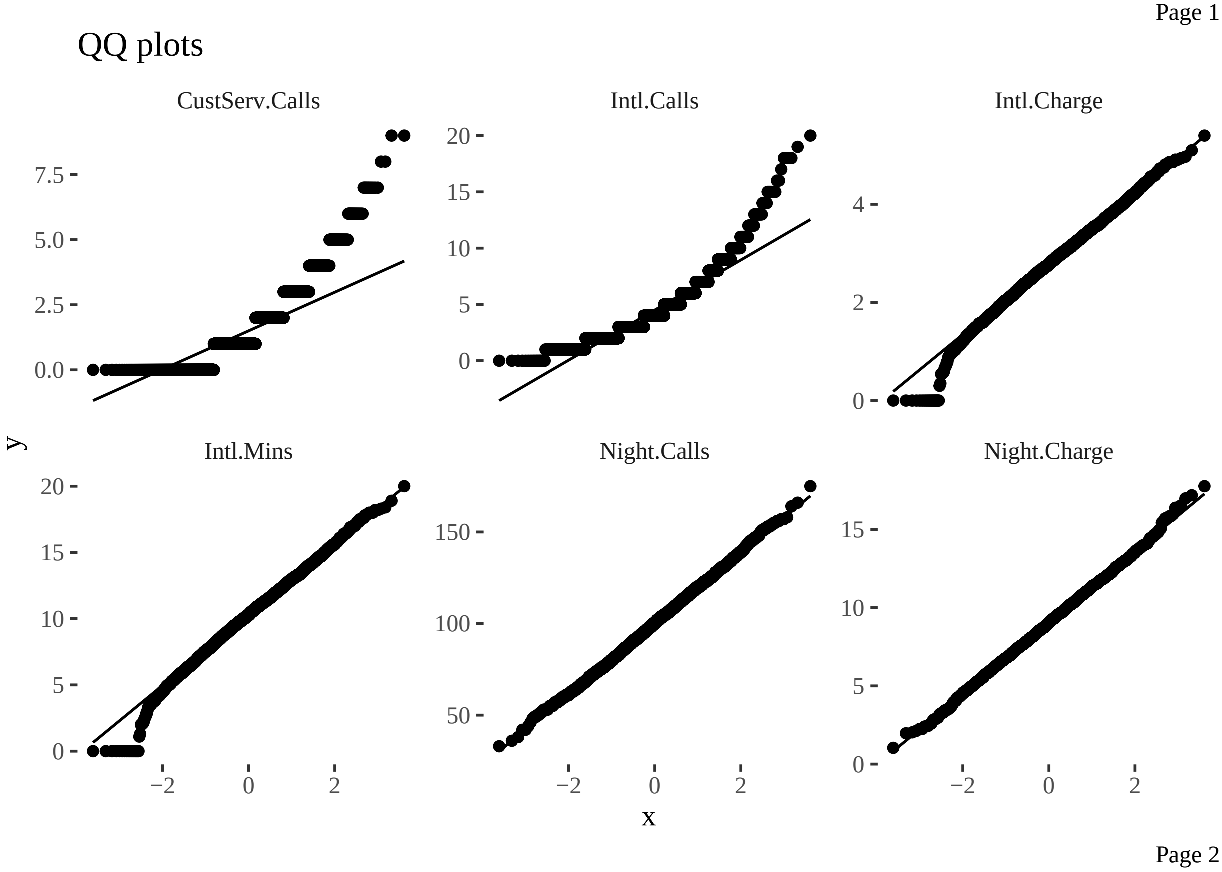


For continuous data, we plotted histograms and QQ plots:



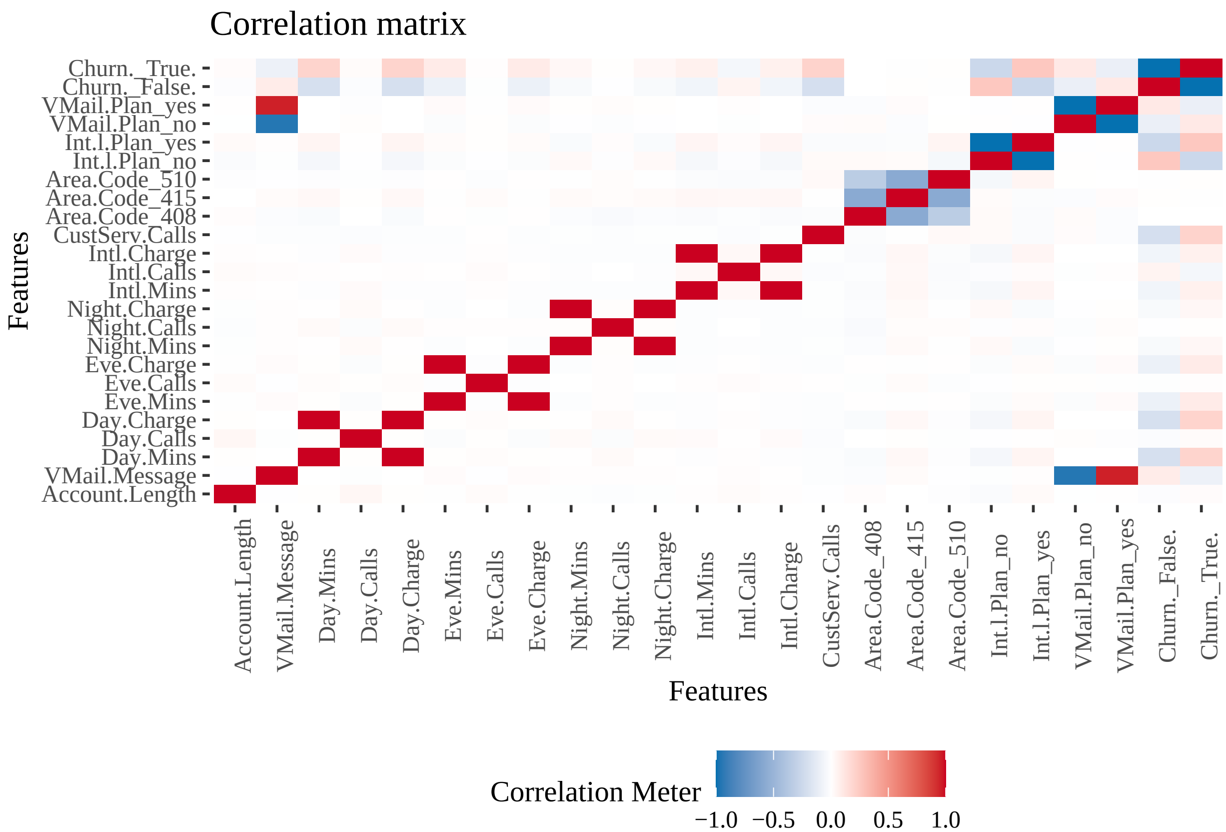
Histograms for numeric data
Account.Length
CustServ.Calls
Day.Calls
Day.Charge
Day.Mins
Eve.Calls
Eve.Charge
Eve.Mins
Intl.Calls
Intl.Charge
Intl.Mins
Night.Calls
Night.Charge
Night.Mins
VMail.Message
Frequency
value



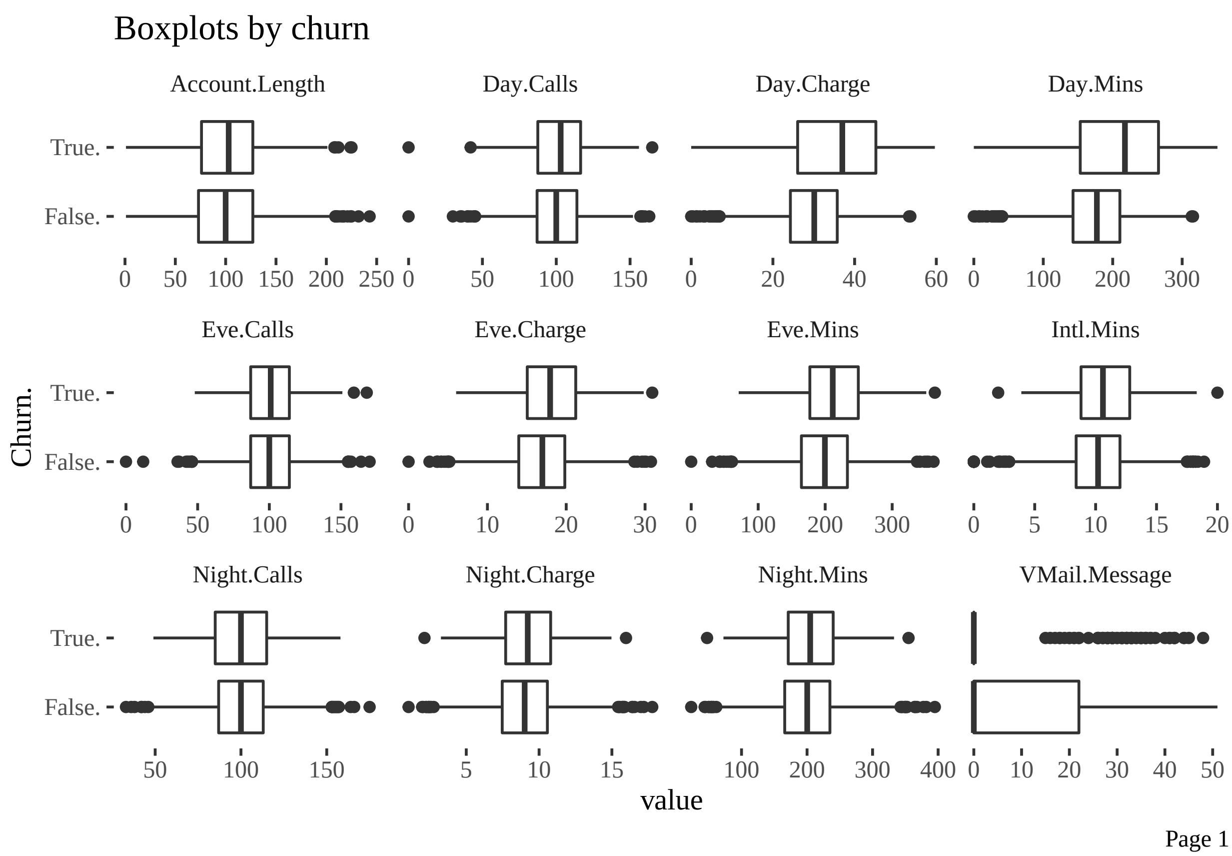


The majority of them is normally distributed with exception to Customer Service Calls, International Calls and Voice Mail Message.

Next we check correlations in order to finally start distinguishing which variables may be useful in creating prediction models.



Of course, we immediately see that strong correlation between charge and minutes of a respective time of day exist. What is of most interest for us is the connection of data to churn, and it is present with both international and voice mail plan as well as with amount of customer service calls, voice mail messages and minutes.

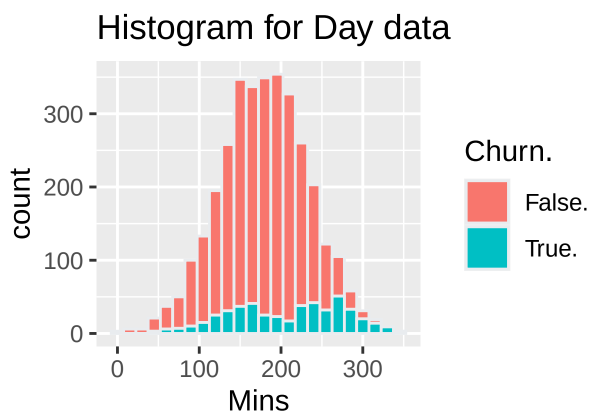


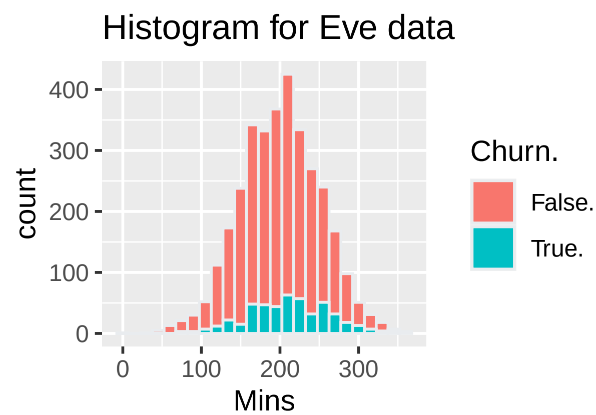
On boxplots we can see where are the biggest differences in data split by churn. For example, boxplot for

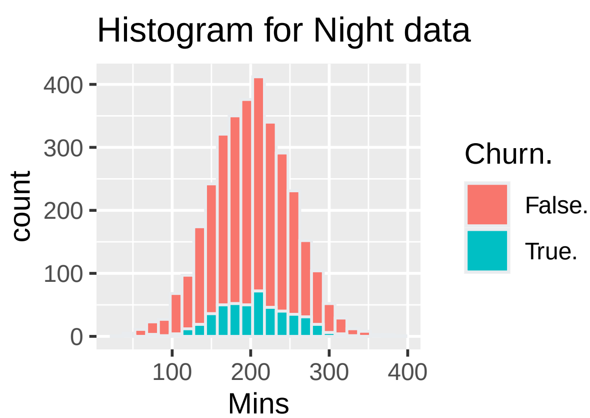
Customer Service Calls with Churn = 'True' has bigger box and greater median, what can lead us to the conclusion that this characteristic has an impact on churn variable. We can see similar dependences as in correlation matrix.

We will now focus more on the variables that have some correlation with the churn, starting with minutes of calls for day, evening and night.

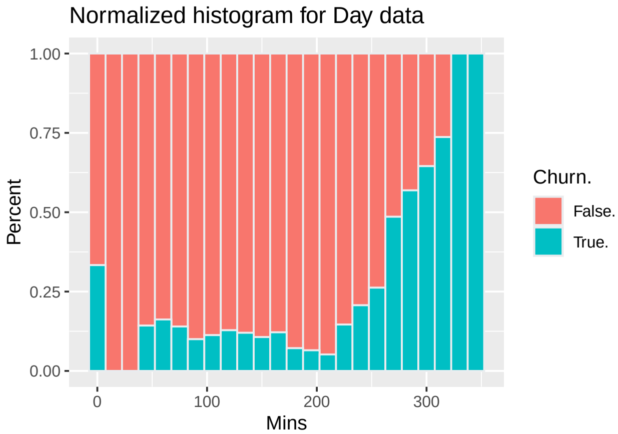
```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

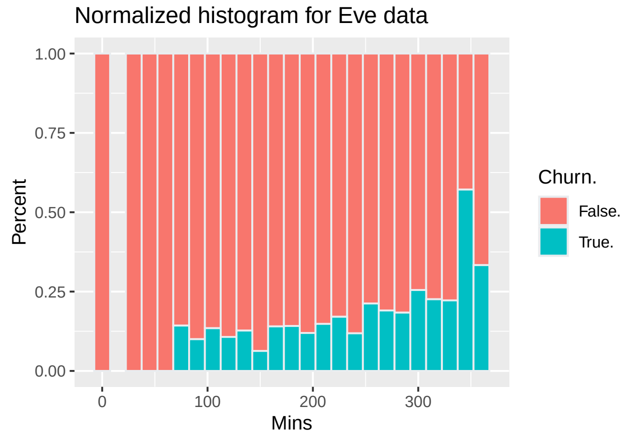


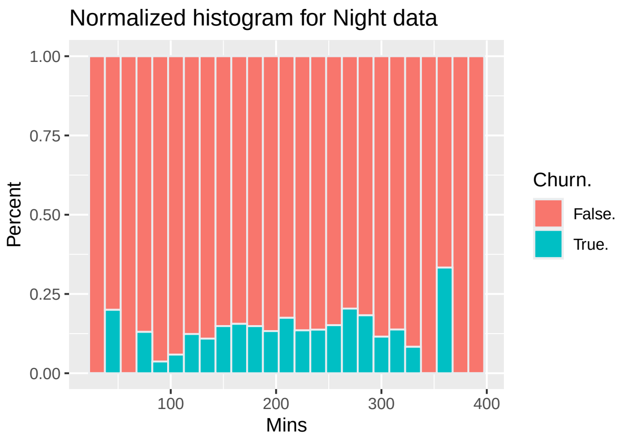




From looking at the above histograms, it appears that clients with high day minutes tends to churn more often. To see more clearly, we will "normalize" the histograms so all the bars are of the same length, so we can see the proportion much better.

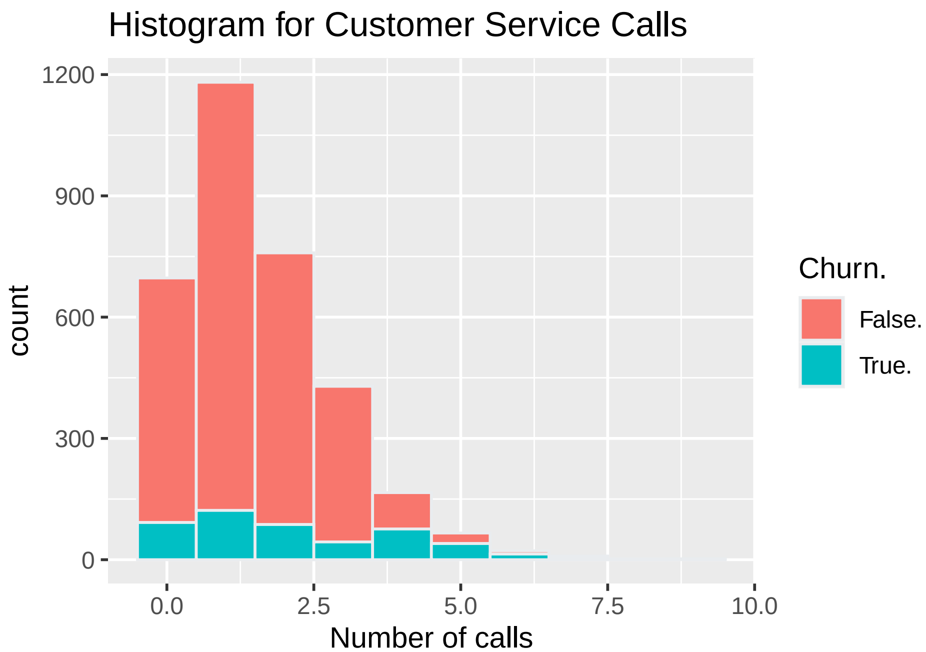


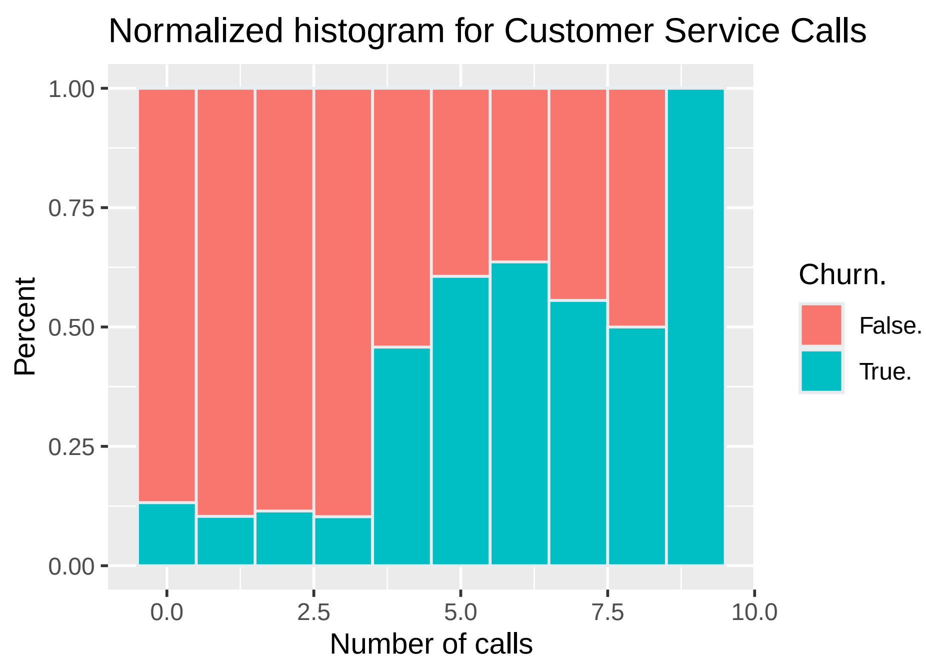




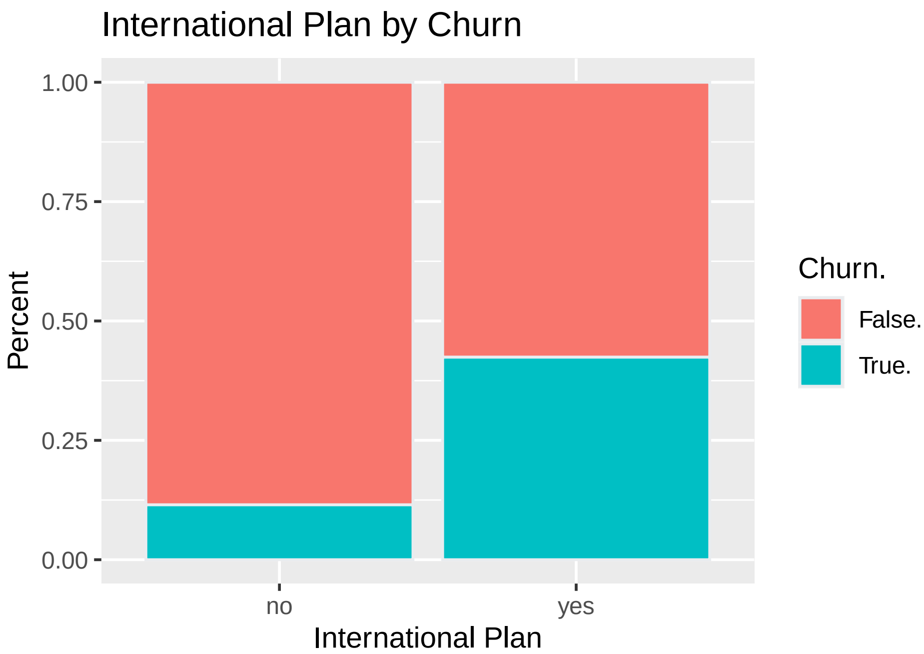
Now it is very clear that customers with high amount of day minutes have higher rate of churn and should definitively be used in our model as a predictor. From the company perspective, it could be useful to monitor those users and pay them special care to potentially prevent churn. The situation isn't that extreme when it comes to evening minutes, but still there is an increase in churn percentage after reaching 300 minutes, so we can still include it in the prediction model. For night minutes, there is no visible trend.

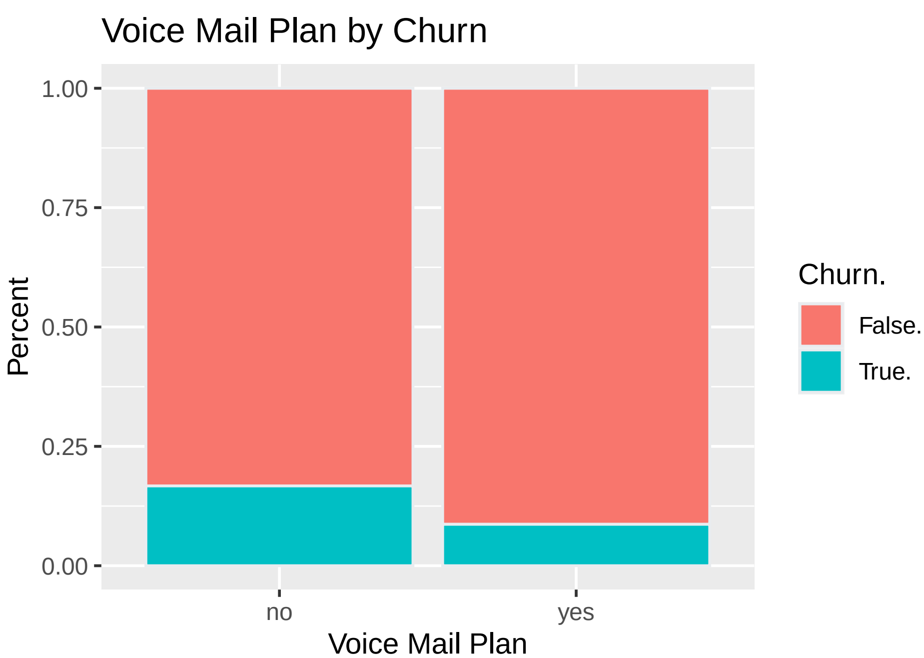
We examine the amount of customer service calls in a similar manner. The higher number of calls corresponds to a much higher rate of churning. It is to be expected because as the more a client have problems that are not resolve quickly with one to three calls the more dissatisfy they are with the service and want to change them.





At last, we focus on different plans customer can obtain - international and voice mail. Those having an international plan are more than three times more likely to churn than those who don't. We can speculate that maybe the plan isn't attractive in comparison to the ones at some other company.





The case with voice mail plan is quite the opposite. Clients who don't have it are approximately two times more likely to churn.

It would be wise for the company to pay close attention to those offers and try to identify problems with them to decrease the churning rate. We will use both of those variables as predictor in our models.

Classification

In our classification models, we used examined above variables that stand out with high correlation with churn and showed they play part in clients' decision to churn. We ended up with 7 predictors, and they are as follows:

```
head(data2)
```

```
##   Int.l.Plan VMail.Plan VMail.Message Day.Mins Eve.Mins Intl.Mins
## 1          1          2            25    265.1    197.4      10.0
## 2          1          2            26    161.6    195.5      13.7
## 3          1          1             0    243.4    121.2      12.2
## 4          2          1             0    299.4     61.9       6.6
## 5          2          1             0    166.7    148.3      10.1
## 6          2          1             0    223.4    220.6       6.3
##   CustServ.Calls
## 1              1
## 2              1
## 3              0
## 4              2
## 5              3
## 6              0
```

We used 4 different methods for classification: linear regression, k-nearest neighbors, linear and quadratic discriminant analysis.

Linear Regression

Firstly, we split our data into learning set and test set. We used one third of the data as a learning set and the rest to test the models.

To get the coefficients for regression, we solve the equation using learning data. We check how well the model is doing on training data and then on the test set.

```
model <- solve(t(X)%*%X) %*% t(X) %*% Y

Y.hat <- X%*%model
Y.hat.test <- X_test%*%model
```

```
##            predicted.labels
## real.labels False. True.
##      False.    966     7
##      True.     129     9
```

```
## [1] 0.8775878
```

```
##                 predicted.labels.test
## real.labels.test False. True.
##           False.   1868     9
##           True.     327    18
```

```
## [1] 0.8487849
```

From the confusion matrices, we see that the model handles the cases of churning really well, while it struggles to correctly distinguish true cases.

Nevertheless, it reaches accuracy grater than 80% for both training set and test set, which is a quite good result.

k-NN

```
#first kNN model, k = 5
model.knn.1 <- ipredknn(Churn. ~ Int.l.Plan+VMail.Plan+VMail.Message+Day.Mins
+Eve.Mins+Night.Mins+Intl.Mins+CustServ.Calls, data=learning.set, k=5)

predicted.labels.knn.train <- predict(model.knn.1,learning.set, type="class")
predicted.labels.knn <- predict(model.knn.1,test.set, type="class")

#second kNN model, k = 10
model.knn.2 <- ipredknn(Churn. ~ Int.l.Plan+VMail.Plan+VMail.Message+Day.Mins
+Eve.Mins+Night.Mins+Intl.Mins+CustServ.Calls, data=learning.set, k=10)

predicted.labels.knn.train2 <- predict(model.knn.2,learning.set, type="class")
predicted.labels.knn2 <- predict(model.knn.2,test.set, type="class")
```

Results for k = 5 (training set, test set):

```
## [1] 0.9068407
```

```
## [1] 0.8865887
```

Results for k = 10 (training set, test set):

```
## [1] 0.9023402
```

```
## [1] 0.8856886
```

With k-NN method, we tested 2 models, for k=5 and k=10. The results were better than in case of regression. For both models, the accuracy was almost equal to 90% for both learning and test set.

```
##
## Call:
## errorest.data.frame(formula = Churn. ~ Int.l.Plan + VMail.Plan +
##     VMail.Message + Day.Mins + Eve.Mins + Night.Mins + Intl.Mins +
##     CustServ.Calls, data = data, model = my.ipredknn, predict = my.predict,
##     estimator = "cv", est.para = control.errorest(k = 10), n.of.neighbors = 5)
##
##   10-fold cross-validation estimator of misclassification error
##
## Misclassification error:  0.1104
```

```
##
## Call:
## errorest.data.frame(formula = Churn. ~ Int.l.Plan + VMail.Plan +
##     VMail.Message + Day.Mins + Eve.Mins + Night.Mins + Intl.Mins +
##     CustServ.Calls, data = data, model = my.ipredknn, predict = my.predict,
##     estimator = "boot", est.para = control.errorest(nboot = 50),
##     n.of.neighbors = 5)
##
##   Bootstrap estimator of misclassification error
##   with 50 bootstrap replications
##
## Misclassification error:  0.1393
## Standard deviation: 0.0012
```

We also did cross-validation and bootstrap-based procedure for model with k=5. They both yield misclassification error not greater than 0.15.

LDA

```r
data.lda  <- lda(Churn. ~ Int.l.Plan+VMail.Plan+VMail.Message+Day.Mins
                 +Eve.Mins+Night.Mins+Intl.Mins
                +CustServ.Calls, data=data, subset=learning.indx)
```

Error values for LDA:

```
## [1] 0.1539154
```

```
##
## Call:
## errorest.data.frame(formula = Churn. ~ Int.l.Plan + VMail.Plan +
##     VMail.Message + Day.Mins + Eve.Mins + Night.Mins + Intl.Mins +
##     CustServ.Calls, data = data, model = my.ipred.lda, predict = my.predict,
##     estimator = "cv", est.para = control.errorest(k = 10))
##
##   10-fold cross-validation estimator of misclassification error
##
## Misclassification error:  0.1503
```

```
##
## Call:
## errorest.data.frame(formula = Churn. ~ Int.l.Plan + VMail.Plan +
##     VMail.Message + Day.Mins + Eve.Mins + Night.Mins + Intl.Mins +
##     CustServ.Calls, data = data, model = my.ipred.lda, predict = my.predict,
##     estimator = "boot", est.para = control.errorest(nboot = 50))
##
##   Bootstrap estimator of misclassification error
##   with 50 bootstrap replications
##
## Misclassification error:  0.1482
## Standard deviation: 4e-04
```

The LDA model got us similar results as the regression model. The misclassification error calculated from confusion matrix equaled around 15%, and the ones from cross-validation and bootstrap were also close to 15%.

QDA

```r
data.qda  <- qda(Churn. ~ Int.l.Plan+VMail.Plan+VMail.Message+Day.Mins
                +Eve.Mins+Night.Mins+Intl.Mins
                +CustServ.Calls, data=data, subset=learning.indx)
```

Error values for QDA:

```
## [1] 0.2412241
```

```
##
## Call:
## errorest.data.frame(formula = Churn. ~ Int.l.Plan + VMail.Plan +
##     VMail.Message + Day.Mins + Eve.Mins + Night.Mins + Intl.Mins +
##     CustServ.Calls, data = data, model = my.ipred.qda, predict = my.predict,
##     estimator = "cv", est.para = control.errorest(k = 10))
##
```

```
##   10-fold cross-validation estimator of misclassification error
##
## Misclassification error:  0.1326
```

```
##
## Call:
## errorest.data.frame(formula = Churn. ~ Int.l.Plan + VMail.Plan +
##     VMail.Message + Day.Mins + Eve.Mins + Night.Mins + Intl.Mins +
##     CustServ.Calls, data = data, model = my.ipred.qda, predict = my.predict,
##     estimator = "boot", est.para = control.errorest(nboot = 50))
##
##   Bootstrap estimator of misclassification error
##   with 50 bootstrap replications
##
## Misclassification error:  0.1314
## Standard deviation: 7e-04
```

The QDA model made the worst predictions. Error from confusion matrix equaled almost 25%. It is interesting that the errors from cross-validation and bootstrap were smaller and similar to those from others models, as they were around 13%.

Summary, part 1

From the conducted analysis, we learned a couple of things. We identify factors that affect churning of customers. The amount of minutes that client spends at calling are one of those. The percentage of churn increase with high amount of minutes, especially at day, and company should monitor it and when it exceeds 200 they should pay more care to those customers, maybe propose them a better offer. The number of customer service calls is also important. After more than 3 calls, the company should contact the client and give special treatment to gain their loyalty, as the probability of churning grows rapidly. The international plan seems to be unattractive and be one of the cause of churning. It can be a good idea to revise it. In contradiction the voice mail plan makes customer stay in present company so better advertisement for this offer can be tactical.

To help predict whether client will churn or not we created 4 classification models - linear regression, k-nearest neighbors, LNA and QDA. It turns out that the best was k-NN model with almost 90% accuracy. The rest were also quite good as they reached more than 80% accuracy, except for QDA model which had accuracy of about 75%. We still can consider it as a success and conclude that our analysis of data was good, and we pinpoint important factors well.

Project - part 2

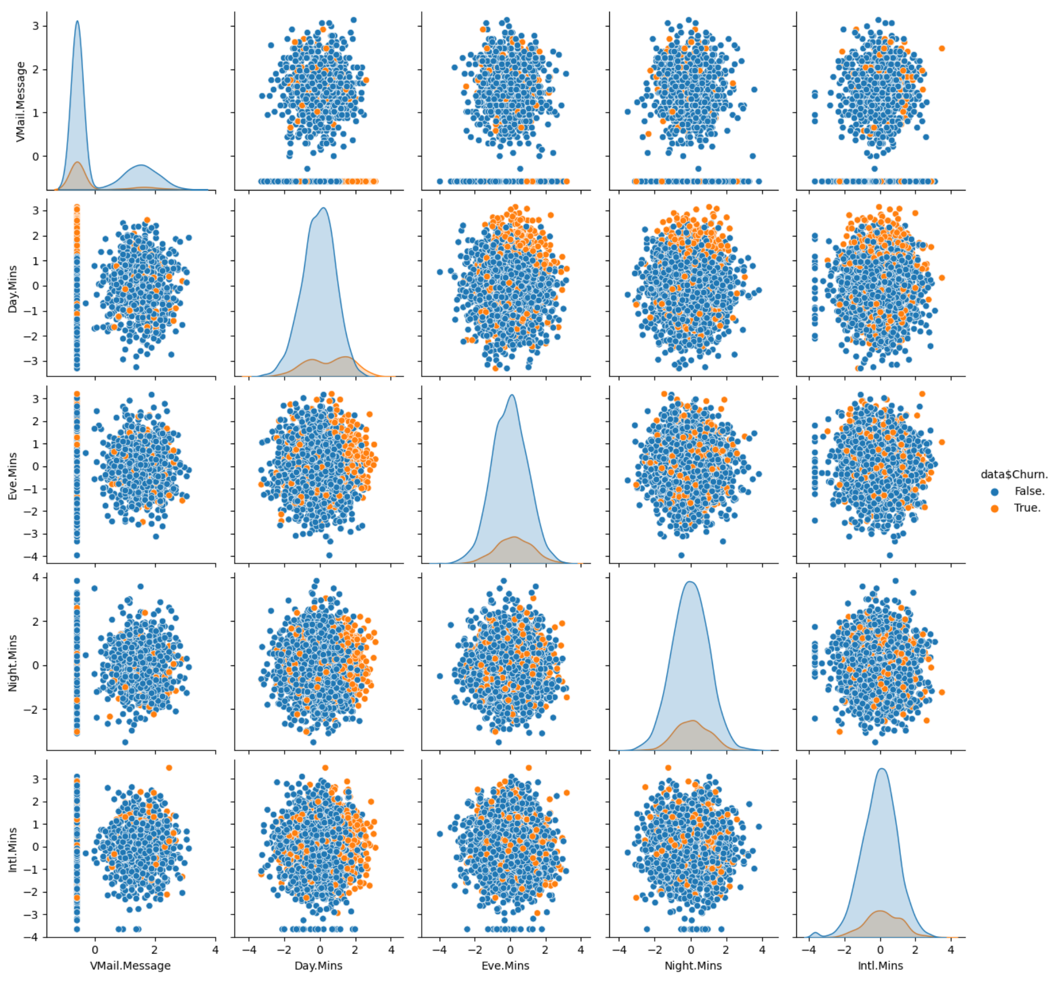
For the cluster analysis, we remove binary data from features we previously chose for classification and standardize the remaining 5 attributes.

```r
head(churn.features)
```

```
##   VMail.Message   Day.Mins    Eve.Mins  Night.Mins   Intl.Mins
## 1     1.2346975  1.5665319 -0.07059903  0.86661319 -0.08499548
## 2     1.3077522 -0.3336877 -0.10806414  1.05841193  1.24029559
## 3    -0.5916711  1.1681284 -1.57314731 -0.75675551  0.70301543
## 4    -0.5916711  2.1962665 -2.74245326 -0.07853935 -1.30283051
## 5    -0.5916711 -0.2400537 -1.03877646 -0.27627001 -0.04917680
## 6    -0.5916711  0.8009362  0.38686974  0.05987211 -1.41028654
```

```
## <seaborn.axisgrid.PairGrid>
```

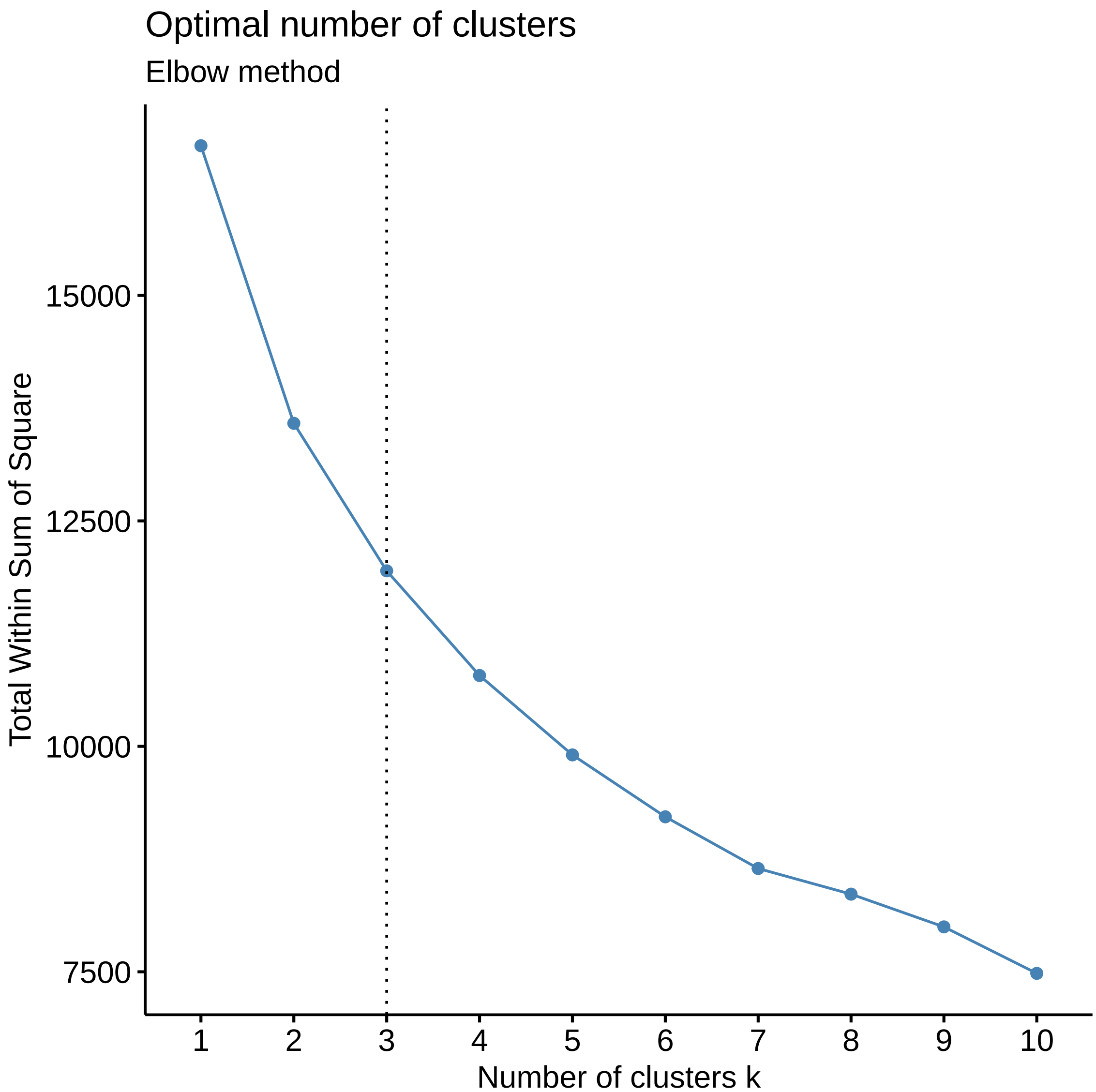
```r
knitr::include_graphics('/Users/kczk/Desktop/data mining/pairplot.png')
```

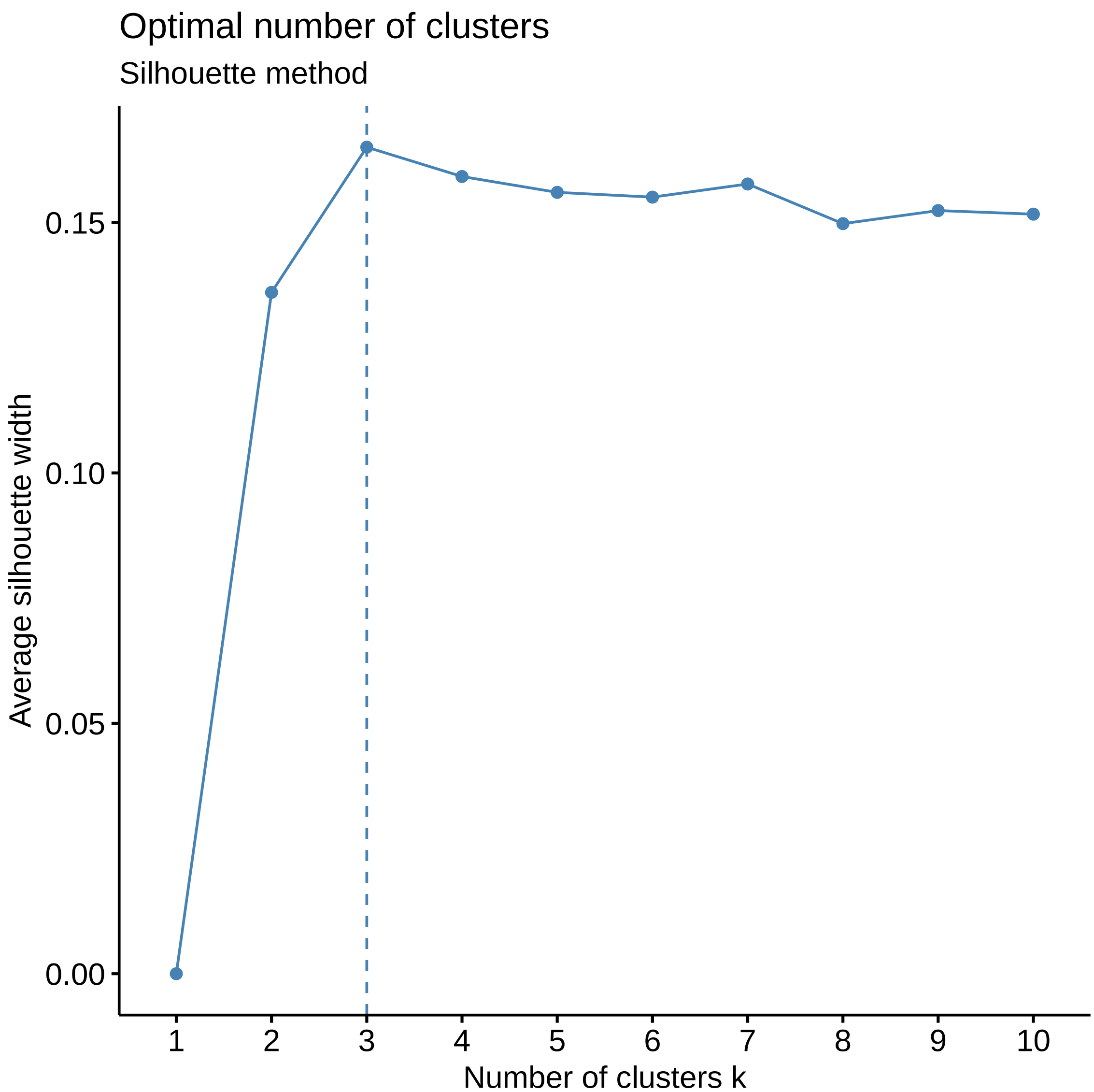


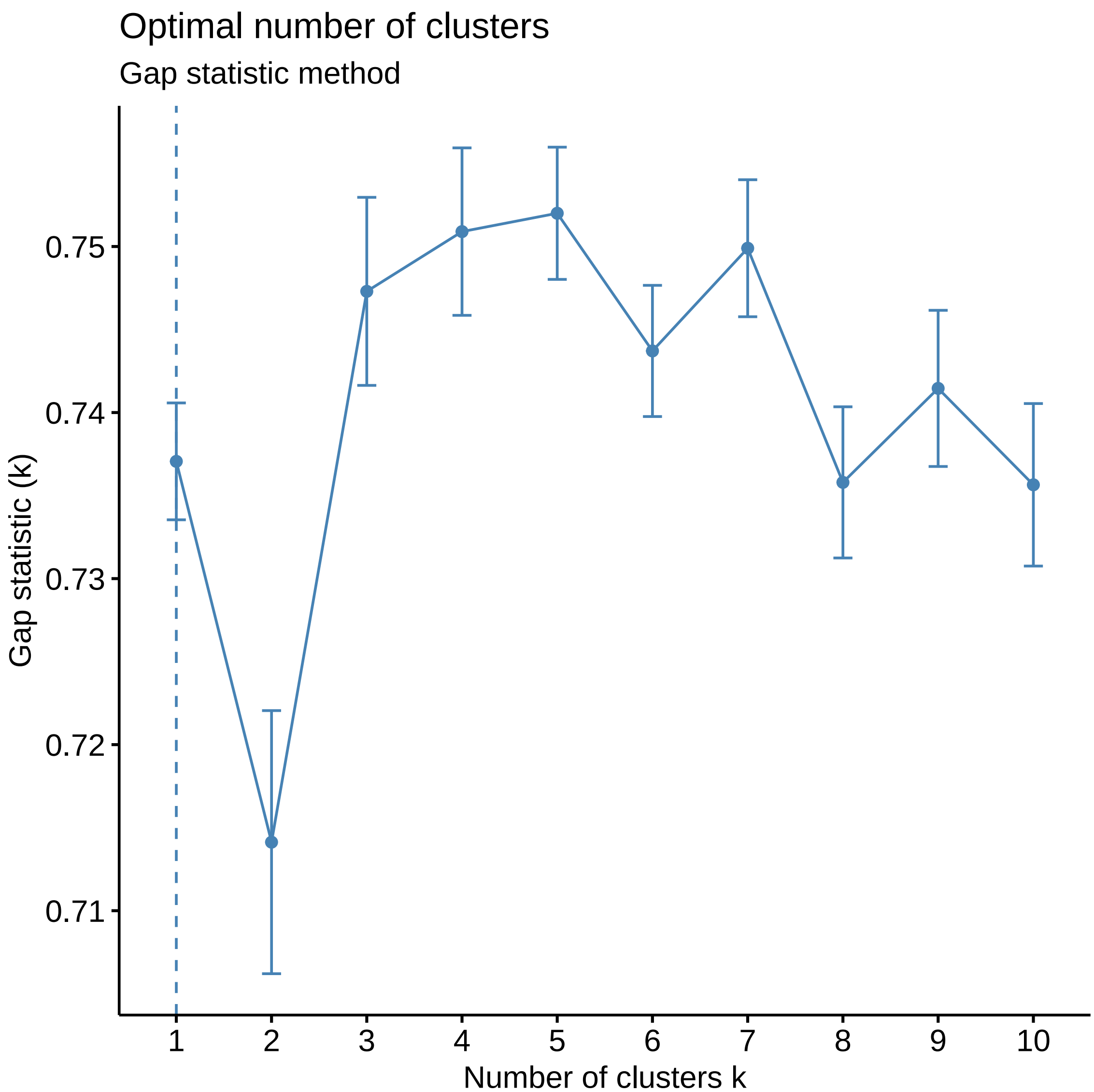
To see the nature of our data, we visualize them using a pair plot for each feature with distinction of churn. It is visible that in every case data has no amount of separation or distinct groups, so we can already suspect they are not well suited for clustering. Nevertheless, we choose to see how it will preform in case of Eve.Mins and Day.Mins as we can see some kind of distinction between churn and not churn clients.

Finding clusters number

Firstly, we check the optimal number of clusters.




Optimal number of clusters
Silhouette method
Average silhouette width
0.15
0.10
0.05
0.00
1
2
3
4
5
6
7
8
9
10
Number of clusters k



For both elbow and silhouette methods, we obtain k=3, which is really not ideal in our case of churn classification. Gap statistic method yields k=1, as there are no visible groups forming in our data, it makes sense and supports the conclusion that we may not get any satisfactory results.

Cluster analysis

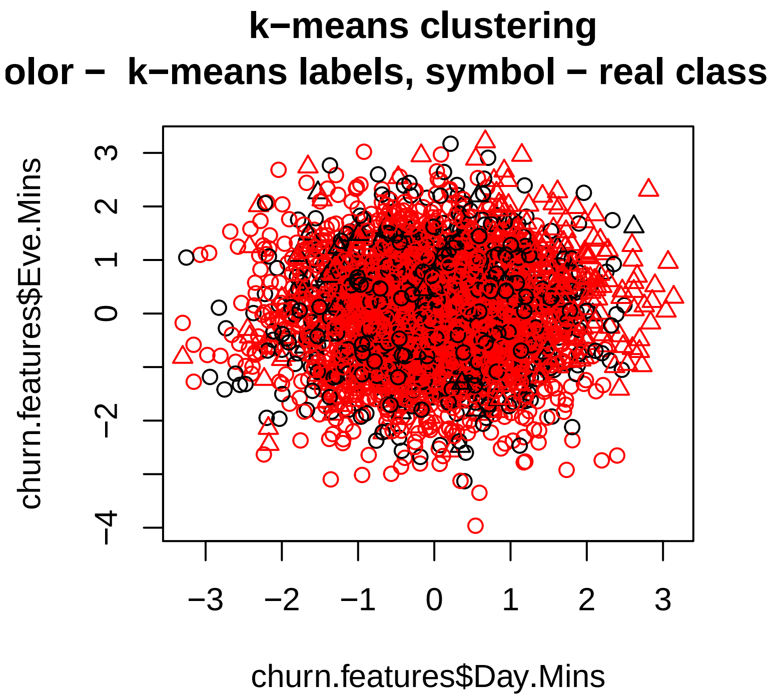
We will examine clusters for both k=3, as was suggested from our analysis, and k=2 as in reality we only have two groups we want to classify: churn and not churn.

k-means method

k=2

```
k <- 2
kmeans.k2 <- kmeans(churn.features, centers=k, iter.max=10, nstart=10)
churn.kmeans.labels <- kmeans.k2$cluster
```

With k-means method we obtain 2 clusters presented at below plot. The two groups are mixed together and there is no significant disparity. We can however look at how it managed matching classes in reference to real labels.



In data, the proportion of clients churning to not looks as follows:

```
##
## False.  True.
##   2850    483
```

Matching classes of clusters and real data, we get only 63.49% accuracy. The size of created clusters is not the worst, but as we can see from confusion matrix, most of the correct predictions come from not churning clients. With that method, we don't identify most cases of churning.

```
## < table of extent 0 >
```

```
##
##         1    2
##   1  813   79
##   2 2037  404
```

```
## Direct agreement: 1 of 2 pairs
## Iterations for permutation matching: 1
## Cases in matched pairs: 63.49 %
```

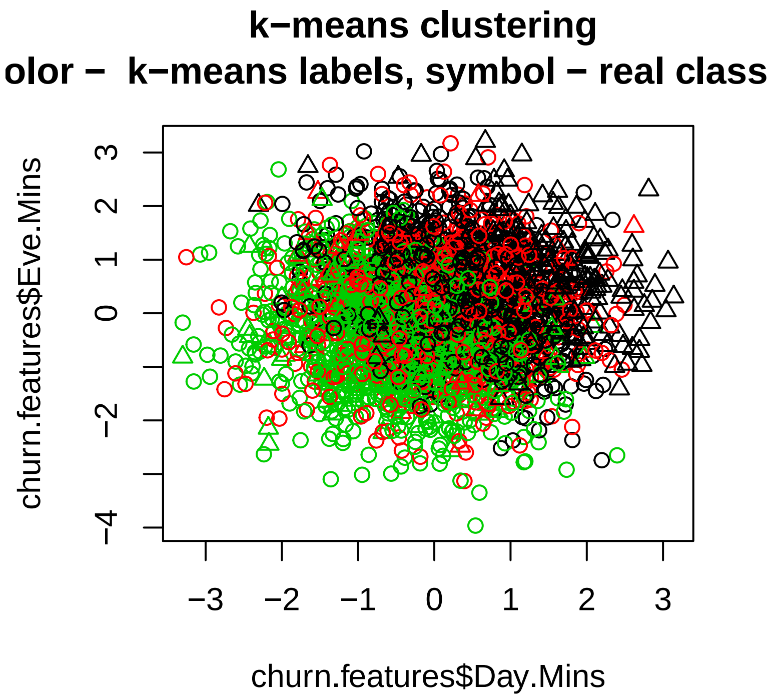
```
## 1 2
## 2 1
```

```
##            [,1]
## [1,] 0.6348635
```

k=3

```r
set.seed(123)
k <- 3
kmeans.k3 <- kmeans(churn.features, centers=k, iter.max=10, nstart=10)
```

At the below plot, we see clusters for k=3. On the contrary for previous results now there are two visible groups. As we can see in the table of labels, the bigger cluster was essentially cut in two.



In terms of classification we can't really match labels as before, but we decided to check if in any of the created groups, one identify a significant number of churn instances. Here are the results:

```
## churn.kmeans.labels1
##    1    2    3
## 1194  858 1281
```

So first we check how many churn were identified correctly in the first cluster.

```
## [1] 249
```

In the first group from 1194 cases only 249 were correctly identify as churn.

```
## [1] 79
```

In the second cluster from 858 cases only 79 were correctly identified.

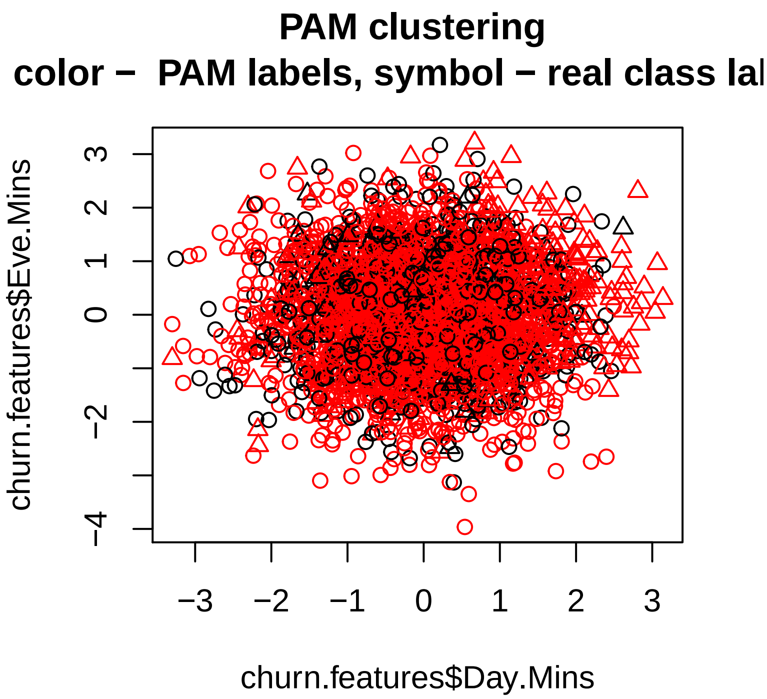
```
## [1] 155
```

And in the third group 155 cases from 1281 were identified correctly.

The most correct prediction we get in first group, but as the cluster is big we can't decide for it to be a group of churning clients as it will assign a lot of wrong labels to not churning clients. In fact, in each case the percentage isn't high enough (approximately 10-20%) to do so.

PAM - Partitioning Around Medoids

```
churn.DissimilarityMatrix <- daisy(churn.features)
churn.DissimilarityMatrix.mat <- as.matrix(churn.DissimilarityMatrix)
churn.pam3 <- pam(x=churn.DissimilarityMatrix.mat, diss=TRUE, k=2)
churn.pam.labels <- churn.pam3$clustering
```




```
##
##         1    2
##   1  810   80
##   2 2040  403
```

```
## Direct agreement: 1 of 2 pairs
## Iterations for permutation matching: 1
## Cases in matched pairs: 63.61 %
```

```
## 1 2
## 2 1
```

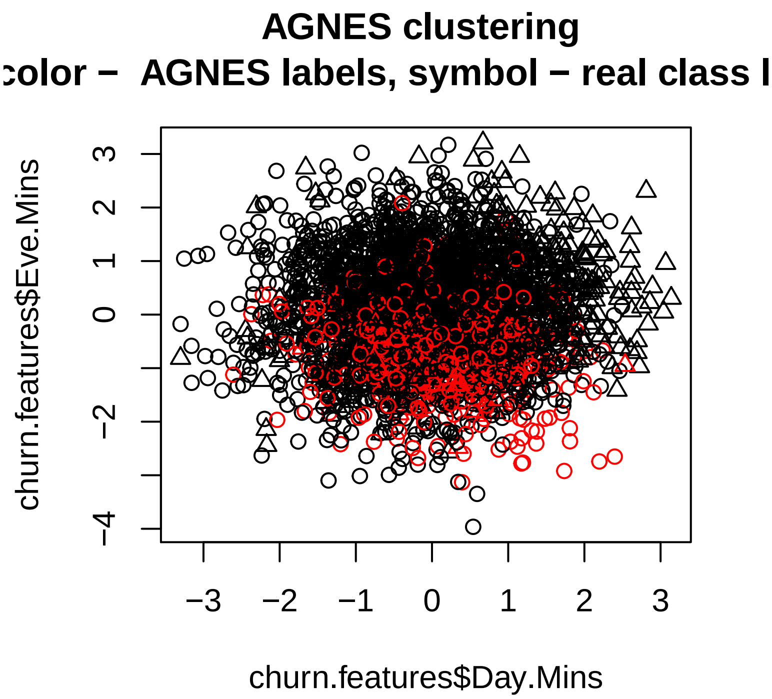
```
##           [,1]
## [1,] 0.6360636
```

Creating clusters with PAM method gives us very similar results as k-means method. With k=2 the proportion of size of cluster is almost the same and the accuracy of is also 63%. The two groups are on top of each other with no visible distinction between them.

AGNES - Agglomerative Nesting (Hierarchical Clustering)

We also preformed hierarchical clustering using agglomerative nesting. The chosen linkage method was complete, as others essentially gave only one cluster.

```
agnes.res <- agnes(churn.features, method="complete")
agnes.partition <- cutree(agnes.res, k=2)
```




```
##
## agnes.partition    1    2
##               1 2298  437
##               2  552   46
```

```
## Direct agreement: 1 of 2 pairs
## Iterations for permutation matching: 1
## Cases in matched pairs: 70.33 %
```

```
## 1 2
## 1 2
```

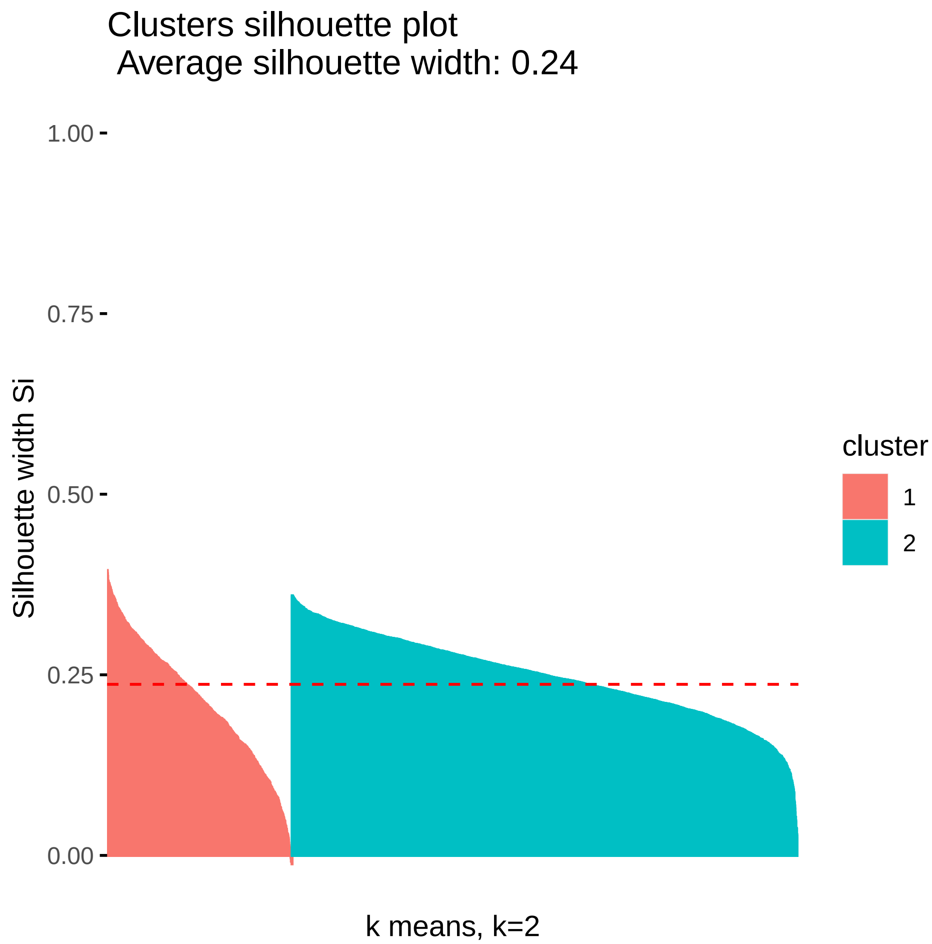
```
##           [,1]
## [1,] 0.7032703
```

In that case, the proportion of labels changed and are more similar to the real ones. The accuracy is improved and equals around 70%. However, it still not the kind of classification we hoped for. The percentage of correctly identified churn cases is equaled only around 10%, which is actually smaller than in k-means method. Checking that in reference to the size of the cluster as we did with k=3, we only get 7%, when before it was around 6%.

Silhouette

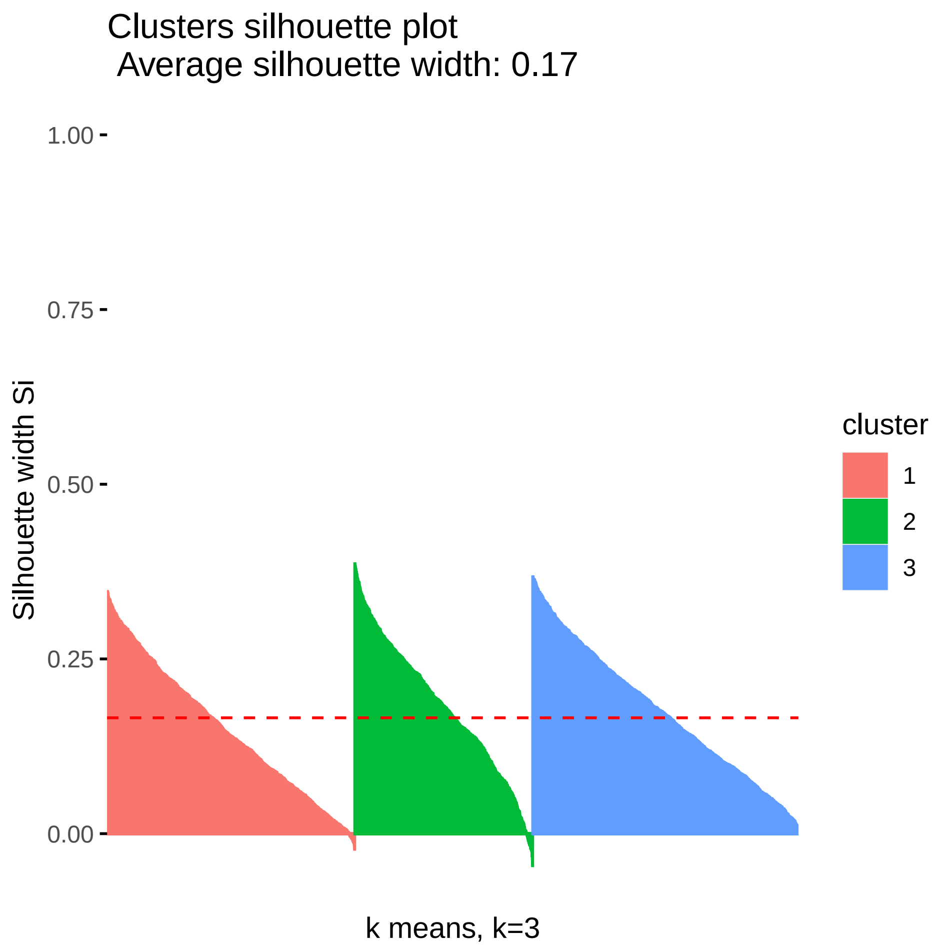
As our results were very much not satisfactory when it comes to predicting our real labels, we also preformed internal validation of used clustering algorithms. For that purpose, silhouette plots were created.

```
##   cluster size ave.sil.width
## 1       1  892          0.21
## 2       2 2441          0.25
```

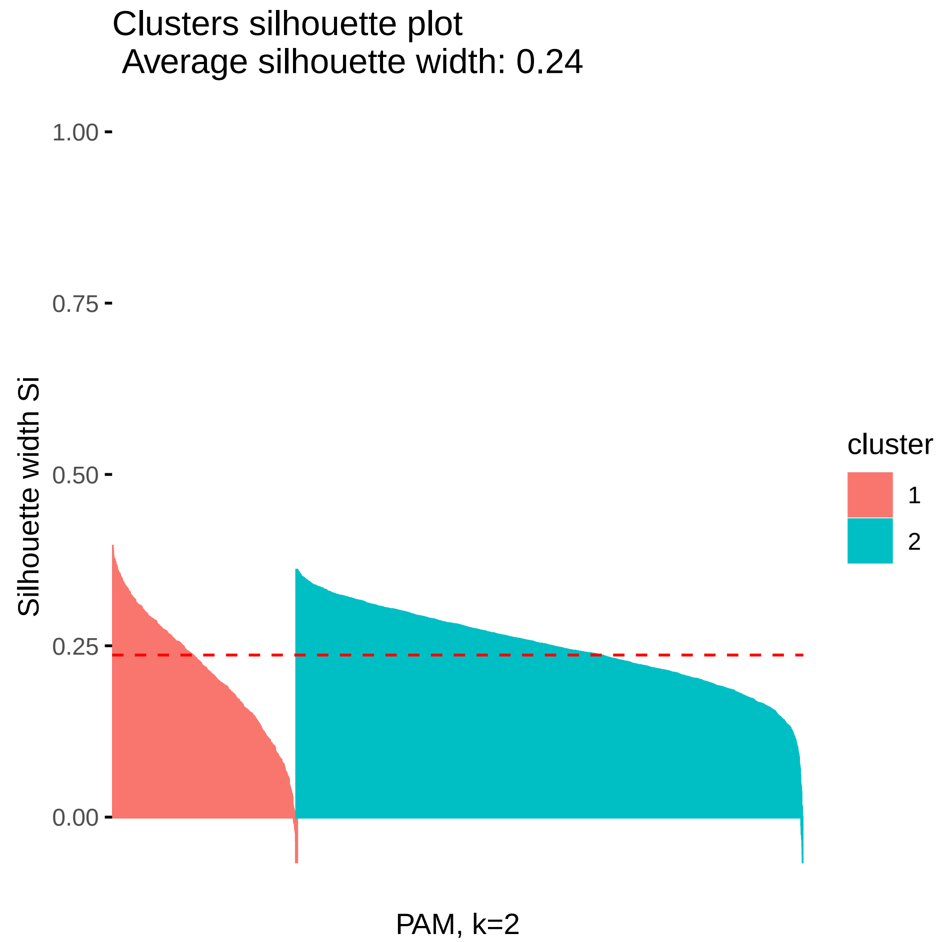


Clusters silhouette plot
Average silhouette width: 0.24
1.00
0.75
0.50
0.25
0.00
Silhouette width Si
k means, k=2
cluster
1
2

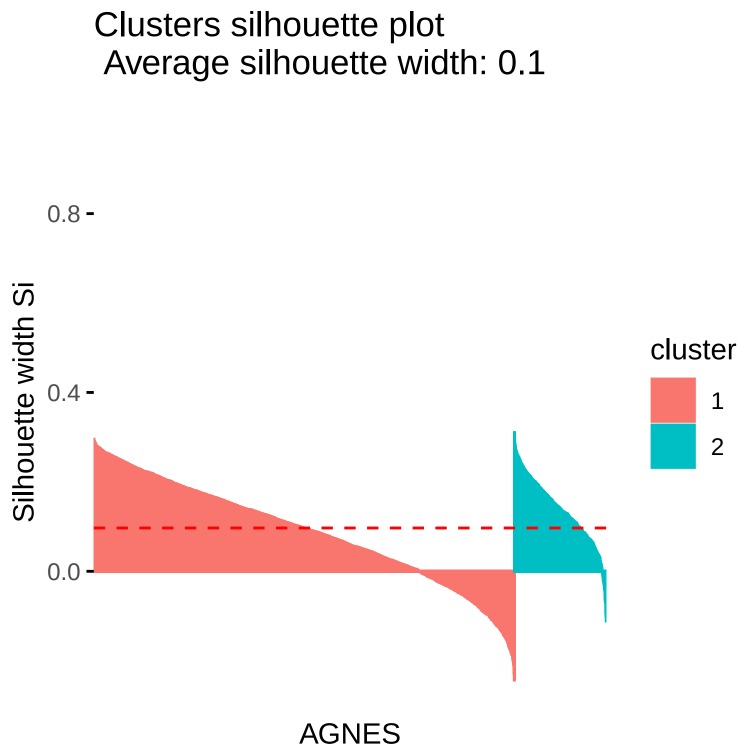
```
##   cluster size ave.sil.width
## 1       1 1194          0.15
## 2       2  858          0.18
## 3       3 1281          0.17
```



```
##   cluster size ave.sil.width
## 1       1  890          0.21
## 2       2 2443          0.25
```




```
##   cluster size ave.sil.width
## 1       1 2735          0.09
## 2       2  598          0.14
```



In each case, the average silhouette width is small - around 0.1-0.2 (with 1 being perfect). In that case, the agnes method actually preformed the worst, while having the best accuracy of predictions. The width for 3 clusters is actually worse than for 2 clusters in k-means method, even though 3 was supposed to be the optimal number.

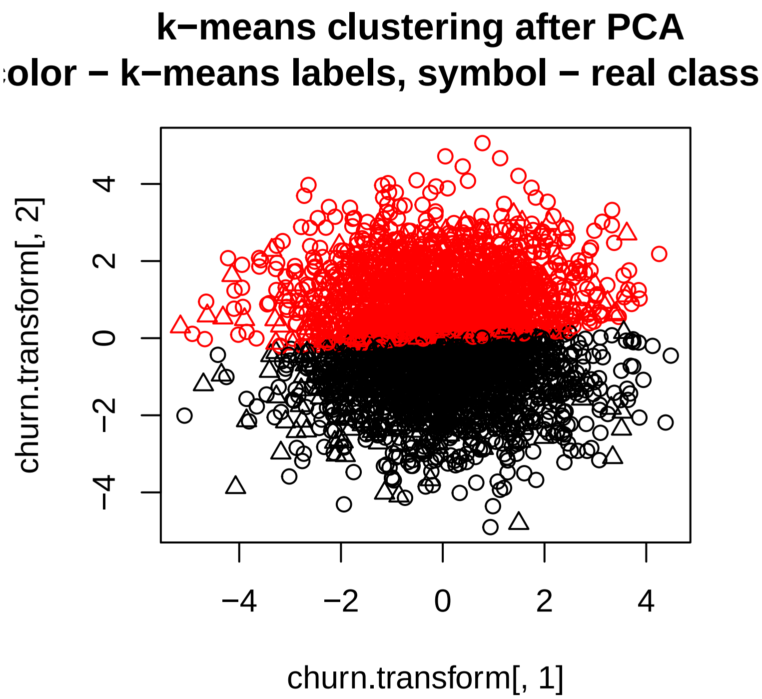
PCA - Principal Component Analysis

To hopefully improve the outcome of foregoing analysis, we preformed dimension reduction using Principal Component Analysis. For that purpose, we use all numerical features, not only those we chose for classification in part 1. Here is the importance of components created by PCA:

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.4302 1.4241 1.4096 1.3962 1.02973 1.01017 1.00307
## Proportion of Variance 0.1461 0.1449 0.1419 0.1393 0.07574 0.07289 0.07187
## Cumulative Proportion  0.1461 0.2910 0.4329 0.5721 0.64788 0.72077 0.79264
##                           PC8     PC9    PC10     PC11      PC12      PC13
## Standard deviation     0.9920 0.98411 0.97493 0.002692 0.0008851 0.0004729
## Proportion of Variance 0.0703 0.06918 0.06789 0.000000 0.0000000 0.0000000
## Cumulative Proportion  0.8629 0.93211 1.00000 1.000000 1.0000000 1.0000000
##                             PC14
## Standard deviation     0.0002185
## Proportion of Variance 0.0000000
## Cumulative Proportion  1.0000000
```

The two best components have cumulative proportion of variance is equal to 29%, which isn't a lot. Most components have a proportion of either around 14% or 7%.

k-mean with PCA



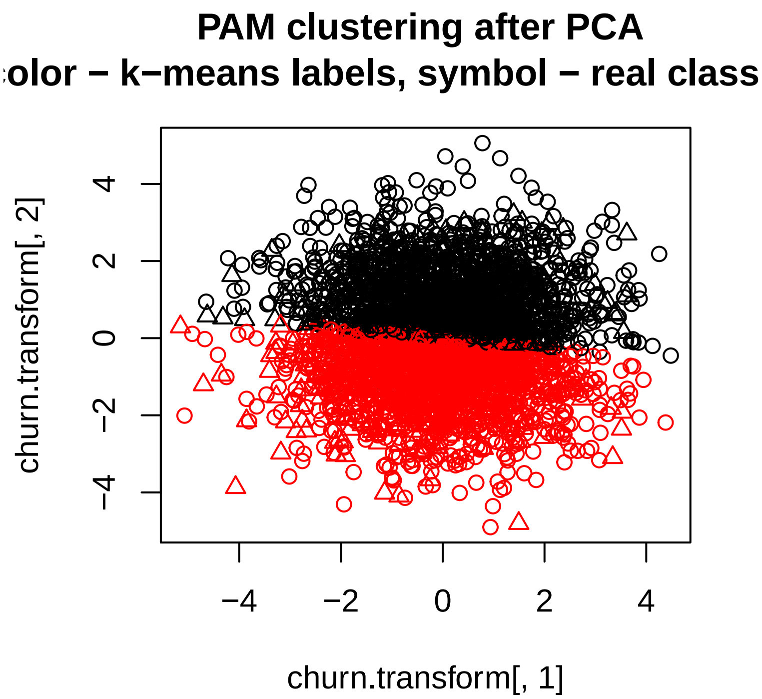

```
##
##        1    2
##   1 1362  273
##   2 1488  210
```

```
## Direct agreement: 0 of 2 pairs
## Iterations for permutation matching: 2
## Cases in matched pairs: 52.84 %
```

```
## 1 2
## 2 1
```

```
##           [,1]
## [1,] 0.5283528
```

PAM with PCA




```
##
##         1    2
##   1 1473  199
##   2 1377  284
```

```
## Direct agreement: 0 of 2 pairs
## Iterations for permutation matching: 2
## Cases in matched pairs: 52.72 %
```

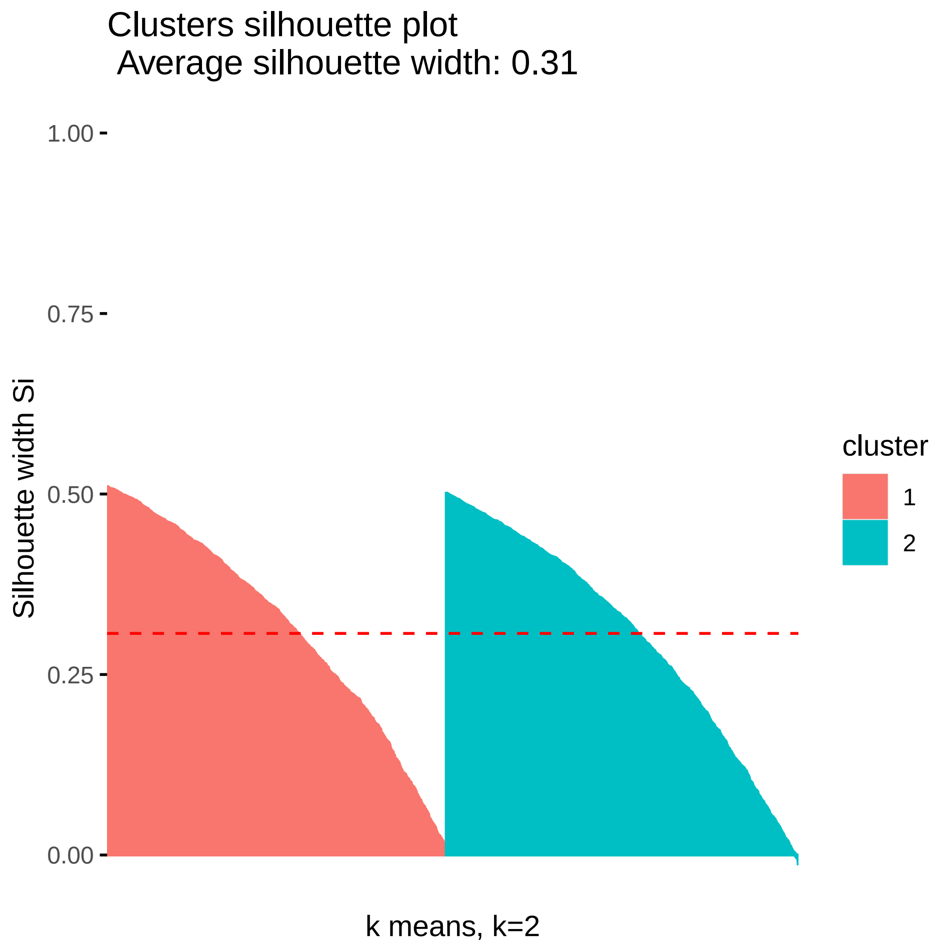
```
## 1 2
## 1 2
```

```
##           [,1]
## [1,] 0.5271527
```

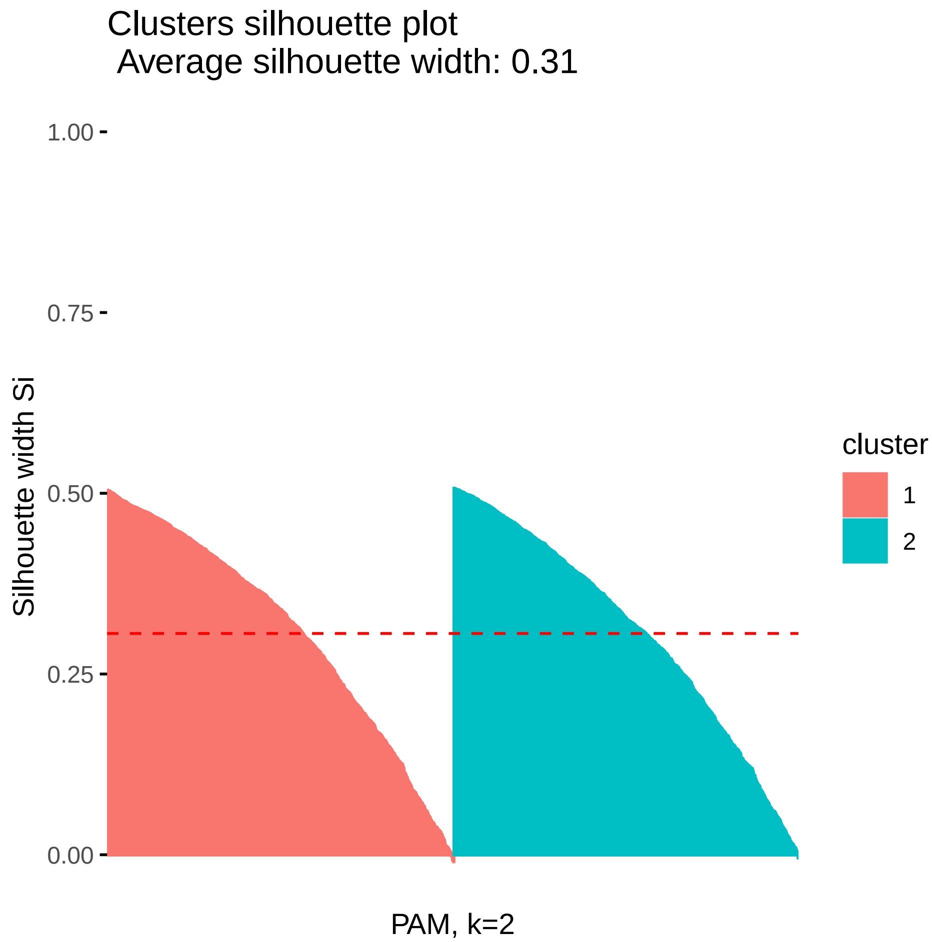
We use PC1 and PC2 for clustering. Now we got visibly split data. The proportion is actually worse than when we didn't use PCA, as now it is simply divided in approximately half. The accuracy of prediction decreased to 52% for both k-means and PAM. The two methods give once again essentially the same outcome.

Silhouette for PCA data

```
##   cluster size ave.sil.width
## 1       1 1635          0.31
## 2       2 1698          0.30
```


Clusters silhouette plot
Average silhouette width: 0.31
1.00
0.75
0.50
0.25
0.00
Silhouette width Si
cluster
1
2
k means, k=2

```
##   cluster size ave.sil.width
## 1       1 1672          0.31
## 2       2 1661          0.31
```



For PCA the internal validation looks a little better, but the results are still bad. For both methods average silhouette width equals 0.31.

Summary

The conducted analysis pretty clearly shows that the churn data aren't well suited for clustering. From the first look at data that was our suspicion, and it was confirmed. In both classification and simple clustering, preformed methods failed to achieved good or even slightly satisfactory results. The groups created by clustering didn't divide clients into churn and not churn, which were our main goal. The internal validation also showed, clusters in general for this data aren't insightful, so trying to analyse created groups at a different angle will also probably be futile and a waste of time.