Decision Trees, Knn, kMeans Clustering

Student : A00209408

1. Decision Trees

1.1. Business Understanding

The problem at hand is how to correctly identify and predict which customers are at risk of leaving the current telecommunications based on data and characteristics about them and their contracts.

This is often referred to as “churn” or “turnover”.

Goals and success criteria:

- Identify customers at risk of leaving
- Classify which customers will not leave
- Develop the associated cost matrix of getting someone to stay vs a customer leaving

1.2. Data Understanding & Preparation

Describe Data

Data is a mix of nominal and ordinal data with a few specific values such as "Yearly /Monthly" contract, and some numerical ones such as tenure with provider (in months) or total billing amounts.

customerI	gender	SeniorCitiz	Partner	Dependen	tenure	PhoneServ	MultipleLi	InternetSe	OnlineSecu	OnlineBacl
7590-VHVI	Female	0	Yes	No	1	No	No phone	DSL	No	Yes
5575-GNV	Male	0	No	No	34	Yes	No	DSL	Yes	No
3668-QPYI	Male	0	No	No	2	Yes	No	DSL	Yes	Yes
7795-CFO(	Male	0	No	No	45	No	No phone	DSL	Yes	No

DevicePro	TechSuppc	Streaming	Streaming	Contract	PaperlessE	PaymentM	MonthlyCh	TotalCharg	Churn
No	No	No	No	Month-to-	Yes	Electronic	29.85	29.85	No
Yes	No	No	No	One year	No	Mailed che	56.95	1889.5	No
No	No	No	No	Month-to-	Yes	Mailed che	53.85	108.15	Yes
Yes	Yes	No	No	One year	No	Bank trans	42.3	1840.75	No

The yes / no for churn can be easily used as boolean values on churn prediction

Explore Data

```
> summary(customer_data)
  customerID            gender          SeniorCitizen       Partner           Dependents           tenure     
 Length:7043        Length:7043        Min.   :0.0000   Length:7043        Length:7043        Min.   : 0.00  
 Class :character   Class :character   1st Qu.:0.0000   Class :character   Class :character   1st Qu.: 9.00  
 Mode  :character   Mode  :character   Median :0.0000   Mode  :character   Mode  :character   Median :29.00  
                                       Mean   :0.1621                                         Mean   :32.37  
                                       3rd Qu.:0.0000                                         3rd Qu.:55.00  
                                       Max.   :1.0000                                         Max.   :72.00  

 PhoneService       MultipleLines      InternetService    OnlineSecurity     OnlineBackup       DeviceProtection
 Length:7043        Length:7043        Length:7043        Length:7043        Length:7043        Length:7043     
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character


 TechSupport        StreamingTV        StreamingMovies      Contract         PaperlessBilling   PaymentMethod   
 Length:7043        Length:7043        Length:7043        Length:7043        Length:7043        Length:7043     
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character


 MonthlyCharges    TotalCharges        Churn          
 Min.   : 18.25   Min.   :  18.8   Length:7043       
 1st Qu.: 35.50   1st Qu.: 401.4   Class :character  
 Median : 70.35   Median :1397.5   Mode  :character  
 Mean   : 64.76   Mean   :2283.3                     
 3rd Qu.: 89.85   3rd Qu.:3794.7                     
 Max.   :118.75   Max.   :8684.8                     
                  NA's   :11                         
> |
```

Verify Data quality

TotalCharges
NA
NA
NA
NA

In some cases I found NA values in the total charges column but it seemed to correlate with "0" tenure time.
Since these were all **not** at risk of leaving I took them out since they wouldn't contribute much to our model and the predictions.

I also dropped the "id" column as it was not useful.

Preparation

Split the data into training and testing data

```
#Split into test and training data
customer_data_train <- customer_data [1:7000,]
customer_data_test <- customer_data[7000:7032,]
```

The proportions of the spread of the sample in both training and test data are roughly equal

```
> prop.table(table(customer_data_train$Churn))

   No   Yes
0.734 0.266
> prop.table(table(customer_data_test$Churn))

       No       Yes
0.7575758 0.2424242
>
```

The Churn column had to be converted to a factor to be usable within the modelling

```
#Convert outcome to a factor
customer_data$Churn <- as.factor(customer_data$Churn)
```

Turnover data split :

```
> table(customer_data$Churn)

  No  Yes
5163 1869
```

1.3. Modelling

- **Basic c50 model:**
    - Very basic model
    - Clear bias towards contract type, which makes sense as someone on “month to month” contract is more likely to leave the company than someone on a fixed contract.
    - Length of tenure seems to play a big part too
    - InternetService is most likely some customers having smaller bandwidth connections (fibre vs copper cables)
    - 19.2% error rate is not amazing but not too bad either with 1 in 5 being wrong.

```
Evaluation on training data (7000 cases):

	    Decision Tree
	  ----------------
	  Size      Errors

	    16 1342(19.2%)   <<


	   (a)   (b)    <-classified as
	  ----  ----
	  4629   509    (a): class No
	   833  1029    (b): class Yes


	Attribute usage:

	100.00% Contract
	 55.07% tenure
	 55.07% InternetService
	 21.14% OnlineSecurity
	 14.61% TechSupport
	 12.43% PaperlessBilling
	 10.57% SeniorCitizen
	  7.87% PaymentMethod
	  4.66% MultipleLines
	  1.36% StreamingTV
```

- **C50 model with “boosting” of 10:**
    - This seems to improve the error rate marginally by 0.2% at a boost of 10 so I do not see much of a performance improvement to continue with this technique

```
Trial       Decision Tree
-----     ----------------
          Size      Errors

   0        16 1342(19.2%)
   1         6 1641(23.4%)
   2        10 1551(22.2%)
   3        12 1712(24.5%)
   4        11 1949(27.8%)
   5         8 1838(26.3%)
   6        12 1902(27.2%)
   7         5 1578(22.5%)
   8         9 1515(21.6%)
   9         9 1424(20.3%)
boost          1327(19.0%)   <<


           (a)   (b)    <-classified as
          ----  ----
          4726   412    (a): class No
           915   947    (b): class Yes


        Attribute usage:

        100.00% Contract
         76.01% InternetService
         70.83% tenure
         70.80% OnlineSecurity
         56.37% StreamingMovies
         55.07% PaymentMethod
         55.07% TotalCharges
         53.59% PaperlessBilling
         42.47% StreamingTV
         39.40% MultipleLines
         32.89% SeniorCitizen
         32.09% PhoneService
         27.44% TechSupport
         11.87% OnlineBackup
          7.70% gender
```

- **Cost matrix model:**
    - A weight of 3 was given for customers leaving and 1 for giving out discounts for customers who were going to stay
    - This actually seemed to decrease performance atleast during the modelling phase

```
Evaluation on training data (7000 cases):

	    Decision Tree
	  ----------------------
	  Size      Errors   Cost

	    54 1735(24.8%)   0.31   <<


	   (a)   (b)    <-classified as
	  ----  ----
	  3629  1509    (a): class No
	   226  1636    (b): class Yes


	Attribute usage:

	100.00% Contract
	 92.54% InternetService
	 68.86% tenure
	 56.01% DeviceProtection
	 31.10% PaymentMethod
```

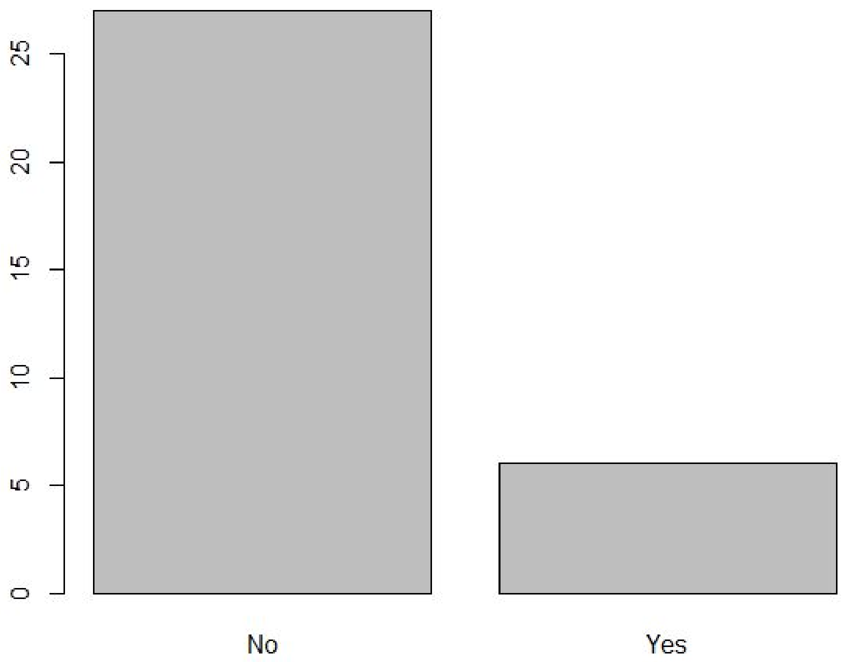
- **CART model:**
    - Seems to give good probabilities for each of the possible options (yes / no)

```
Node number 14: 1083 observations
  predicted class=No   expected loss=0.4099723  P(node) =0.1547143
    class counts:   639   444
   probabilities: 0.590 0.410

Node number 15: 1033 observations
  predicted class=Yes  expected loss=0.3088093  P(node) =0.1475714
    class counts:   319   714
   probabilities: 0.309 0.691

> |
```

1.4. Evaluation

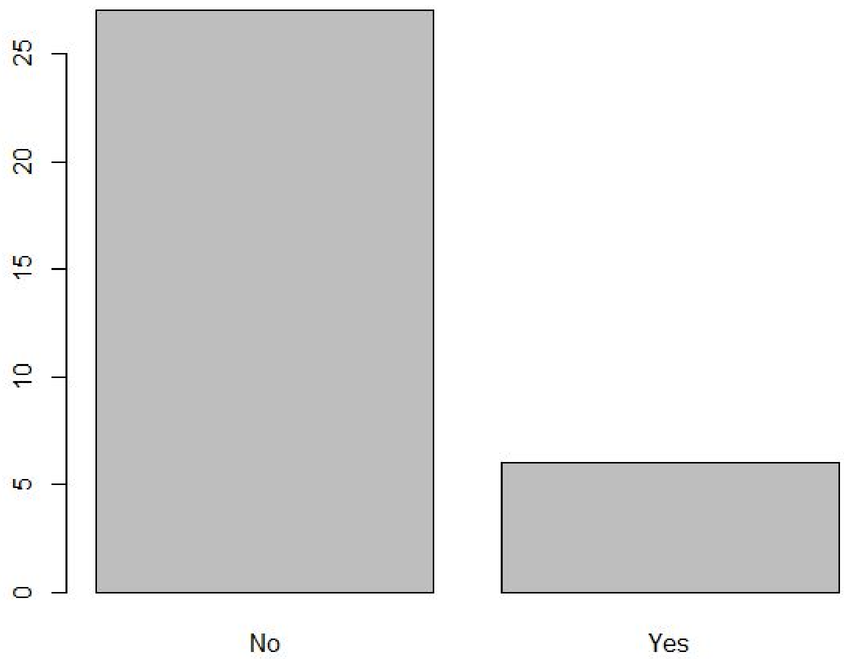
- **Basic c50 model:**
  - By far the best performing model as we can see by the confusion matrix below
  - .697+.121 = 0.818 which is a very good value for the prediction
  - Much smaller values for “yes”

```
                  | actual default 
predicted default |        No |       Yes | Row Total | 
------------------|-----------|-----------|-----------|
               No |        23 |         4 |        27 | 
                  |     0.697 |     0.121 |           | 
------------------|-----------|-----------|-----------|
              Yes |         2 |         4 |         6 | 
                  |     0.061 |     0.121 |           | 
------------------|-----------|-----------|-----------|
     Column Total |        25 |         8 |        33 | 
------------------|-----------|-----------|-----------|
```



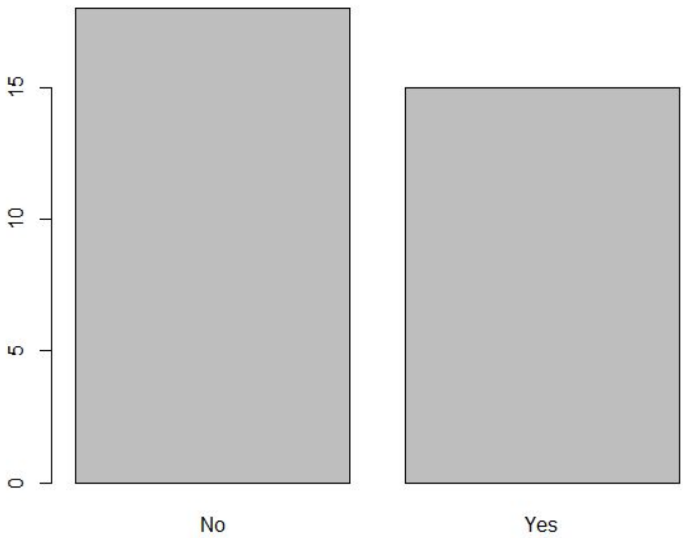
- **Basic c50 model with boosting of 10:**
  - No visible improvements on the base model so boosting does not help

predicted default	actual default No	Yes	Row Total
No	23 0.697	4 0.121	27
Yes	2 0.061	4 0.121	6
Column Total	25	8	33

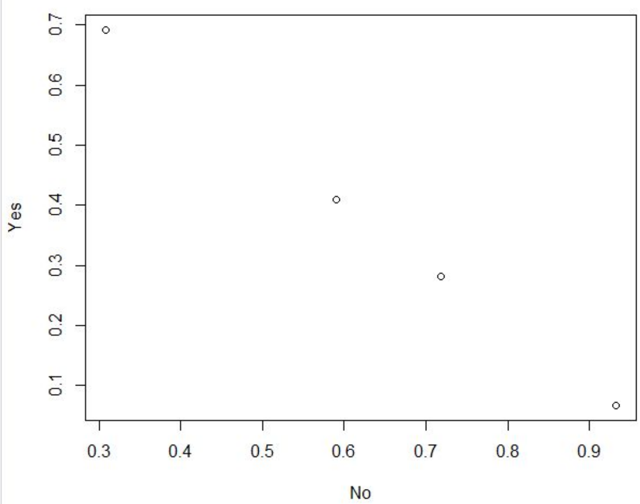


- **Basic model with cost matrix:**
    - The total diagonal value is : .485 + .182 = .667 which is much weaker than the original algorithm
    - The value predictions also seem to be spread out much more equally as seen below

```
                  | actual default
predicted default |        No |       Yes | Row Total |
------------------|-----------|-----------|-----------|
               No |        16 |         2 |        18 |
                  |     0.485 |     0.061 |           |
------------------|-----------|-----------|-----------|
              Yes |         9 |         6 |        15 |
                  |     0.273 |     0.182 |           |
------------------|-----------|-----------|-----------|
     Column Total |        25 |         8 |        33 |
------------------|-----------|-----------|-----------|
```



- **Cartesian:**



2. kNN

2.1. Business Understanding

The problem at hand is how to correctly identify and predict which customers are at risk of leaving the current telecommunications based on data and characteristics about them and their contracts.

This is often referred to as “churn” or “turnover”.

Goals and success criteria:

- Identify customers at risk of leaving
- Classify which customers will not leave
- Develop the associated cost matrix of getting someone to stay vs a customer leaving

2.2. Data Understanding & Preparation

Describe Data

Data is a mix of nominal and ordinal data with a few specific values such as “Yearly /Monthly” contract, and some numerical ones such as tenure with provider (in months) or total billing amounts.

customerI	gender	SeniorCitiz	Partner	Dependen	tenure	PhoneServ	MultipleLi	InternetSe	OnlineSecu	OnlineBac
7590-VHVI	Female	0	Yes	No	1	No	No phone	DSL	No	Yes
5575-GNV	Male	0	No	No	34	Yes	No	DSL	Yes	No
3668-QPYI	Male	0	No	No	2	Yes	No	DSL	Yes	Yes
7795-CFO(	Male	0	No	No	45	No	No phone	DSL	Yes	No

DevicePro	TechSuppo	Streaming	Streaming	Contract	PaperlessE	PaymentM	MonthlyCh	TotalCharg	Churn
No	No	No	No	Month-to-	Yes	Electronic	29.85	29.85	No
Yes	No	No	No	One year	No	Mailed che	56.95	1889.5	No
No	No	No	No	Month-to-	Yes	Mailed che	53.85	108.15	Yes
Yes	Yes	No	No	One year	No	Bank trans	42.3	1840.75	No

The yes / no for churn can be easily used as boolean values on churn prediction

Explore Data

```
> summary(customer_data)
  customerID           gender          SeniorCitizen       Partner           Dependents            tenure     
 Length:7043        Length:7043        Min.   :0.0000   Length:7043        Length:7043        Min.   : 0.00  
 Class :character   Class :character   1st Qu.:0.0000   Class :character   Class :character   1st Qu.: 9.00  
 Mode  :character   Mode  :character   Median :0.0000   Mode  :character   Mode  :character   Median :29.00  
                                       Mean   :0.1621                                         Mean   :32.37  
                                       3rd Qu.:0.0000                                         3rd Qu.:55.00  
                                       Max.   :1.0000                                         Max.   :72.00  

 PhoneService       MultipleLines      InternetService    OnlineSecurity     OnlineBackup       DeviceProtection
 Length:7043        Length:7043        Length:7043        Length:7043        Length:7043        Length:7043     
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character



 TechSupport        StreamingTV        StreamingMovies      Contract         PaperlessBilling   PaymentMethod   
 Length:7043        Length:7043        Length:7043        Length:7043        Length:7043        Length:7043     
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character



 MonthlyCharges    TotalCharges       Churn          
 Min.   : 18.25   Min.   :  18.8   Length:7043       
 1st Qu.: 35.50   1st Qu.: 401.4   Class :character  
 Median : 70.35   Median :1397.5   Mode  :character  
 Mean   : 64.76   Mean   :2283.3                     
 3rd Qu.: 89.85   3rd Qu.:3794.7                     
 Max.   :118.75   Max.   :8684.8                     
                  NA's   :11                         
> |
```

Verify Data quality

TotalCharges
NA
NA
NA
NA

In some cases I found NA values in the total charges column but it seemed to correlate with “0” tenure time.
Since these were all **not** at risk of leaving I took them out since they wouldn’t contribute much to our model and the predictions.

I also dropped the “id” column as it was not useful.

Preparation

Split the data into training and testing data

```
#Split into test and training data
customer_data_train <- customer_data [1:7000,]
customer_data_test <- customer_data[7000:7032,]
```

The proportions of the spread of the sample in both training and test data are roughly equal

```
> prop.table(table(customer_data_train$Churn))

   No   Yes
0.734 0.266
> prop.table(table(customer_data_test$Churn))

       No       Yes
0.7575758 0.2424242
> |
```

The Churn column had to be converted to a factor to be usable within the modelling

```
#Convert outcome to a factor
customer_data$Churn <- as.factor(customer_data$Churn)
```

Turnover data split :

```
> table(customer_data$Churn)

  No  Yes
5163 1869
```

In this case I also normalized the “monthly charges” column so that it would not dominate the predictions.

I also had to to dummy code most of the data since they were categorical values.

2.3. Modelling

There was not a lot involved in the modelling phase, I have simply created three models with different K values as follows:

- K = 2
- K = 10
- K = 30

Scaling and normalization did not improve the results so I have left them out after testing.

The main bulk of the work within the modelling was the conversion of all categorical variables into numerical ones to help with the prediction which worked pretty well in the end.

kNN seems better for numerical data sets compared to the decision tree technique.

2.4. Evaluation

- K = 2
  - This model gave a pretty average prediction score of a total of : .515 + .152 = .662

```
                         | customer_data_test_labels
customer_data_predictions |        no |       yes | Row Total |
-------------------------|-----------|-----------|-----------|
                      no |        17 |         3 |        20 |
                         |     0.515 |     0.091 |           |
-------------------------|-----------|-----------|-----------|
                     yes |         8 |         5 |        13 |
                         |     0.242 |     0.152 |           |
-------------------------|-----------|-----------|-----------|
            Column Total |        25 |         8 |        33 |
-------------------------|-----------|-----------|-----------|
```

- K = 10
  - The total prediction score was : .727 + .121 = .848
  - This model had the highest score and even beat the score of the “Decison Tree” model by a tiny margin

```
                              | customer_data_test_labels
customer_data_predictions_k_10 |        no |       yes | Row Total |
------------------------------|-----------|-----------|-----------|
                           no |        24 |         4 |        28 |
                              |     0.727 |     0.121 |           |
------------------------------|-----------|-----------|-----------|
                          yes |         1 |         4 |         5 |
                              |     0.030 |     0.121 |           |
------------------------------|-----------|-----------|-----------|
                 Column Total |        25 |         8 |        33 |
------------------------------|-----------|-----------|-----------|
```

- K = 30
  - It would seem that increasing the “k” value above 10 made no difference to the prediction scores

```
                              | customer_data_test_labels
customer_data_predictions_k_30 |        no |       yes | Row Total |
------------------------------|-----------|-----------|-----------|
                           no |        24 |         4 |        28 |
                              |     0.727 |     0.121 |           |
------------------------------|-----------|-----------|-----------|
                          yes |         1 |         4 |         5 |
                              |     0.030 |     0.121 |           |
------------------------------|-----------|-----------|-----------|
                 Column Total |        25 |         8 |        33 |
------------------------------|-----------|-----------|-----------|
```

3. kMeans Clustering

3.1. Business Understanding

I have chosen a new dataset for this assignment as the last one was not ideal for classification.

This data contains a few simple columns and identifies customers and their spending scores within a shopping centre.

From this data we want to identify clusters as possible targeting points for advertisements based on people's age, gender, income and spending score

Goals and success criteria:

- Identify customer clusters based on income and age
- Find potential links to spending habits based on the other columns
- Prepare a advertisement targeting campaign based on a users cluster membership

3.2. Data Understanding & Preparation

Describe Data

The data consists of a few simple numerical columns and one Gender categorical column which can be easily converted to a binary value.

The Spending.Score is a important value that is assigned based on behavior and spending nature (the higher the score the more likely to spend)

	CustomerID	Gender	Age	Annual.Income	Spending.Score
1	1	Male	19	15	39
2	2	Male	21	15	81
3	3	Female	20	16	6
4	4	Female	23	16	77

Explore Data

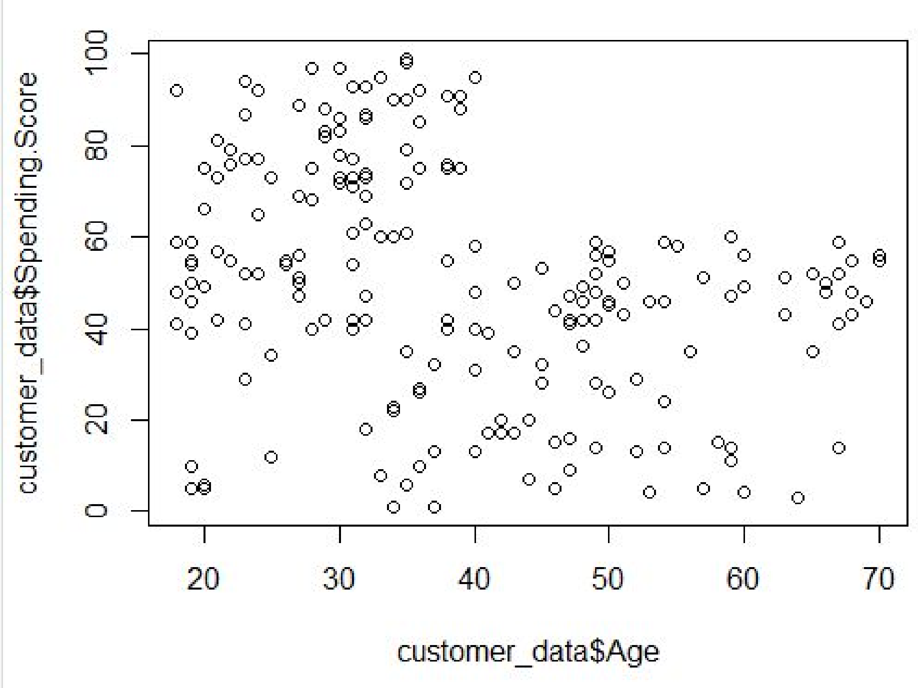
There appears to be no odd data within the set.

```
  CustomerID         Gender               Age        Annual.Income    Spending.Score
 Min.   :  1.00   Length:200         Min.   :18.00   Min.   : 15.00   Min.   : 1.00
 1st Qu.: 50.75   Class :character   1st Qu.:28.75   1st Qu.: 41.50   1st Qu.:34.75
 Median :100.50   Mode  :character   Median :36.00   Median : 61.50   Median :50.00
 Mean   :100.50                      Mean   :38.85   Mean   : 60.56   Mean   :50.20
 3rd Qu.:150.25                      3rd Qu.:49.00   3rd Qu.: 78.00   3rd Qu.:73.00
 Max.   :200.00                      Max.   :70.00   Max.   :137.00   Max.   :99.00
> 
```

There is a slight skew towards female customers in the set, spending scores are all over the place with a few clusters around the midpoint

```
 Max.   :200.00                      Max.   :70.00   Max.   :137.00   Max.   :99.00
> table(customer_data$Spending.Score)

 1  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 20 22 23 24 26 27 28 29 31 32 34 35 36 39 40 41 42 43 44 45 46 47 48 49
 2  1  2  4  2  1  1  1  2  1  1  3  4  3  2  3  1  2  1  1  1  2  1  2  2  1  2  1  5  2  2  4  4  8  3  1  1  6  4  5  3
50 51 52 53 54 55 56 57 58 59 60 61 63 65 66 68 69 71 72 73 74 75 76 77 78 79 81 82 83 85 86 87 88 89 90 91 92 93 94 95 97
 5  3  5  1  3  7  4  2  2  5  3  2  1  1  1  1  2  1  2  6  2  5  2  3  1  2  2  1  2  1  2  2  3  1  2  2  3  2  1  2  2
98 99
 1  1
> table(customer_data$Gender)

Female   Male
   112     88
> 
```

There is also a clear distinction between younger / middle aged people to spend much more than the older ones.



Verify Data quality

There appears to be no missing data but null checks and cases will be dropped just in case.

Preparation

The gender column will be converted to a binary dummy variable to allow for the model to compile.

3.3. Modelling

I have created 3 distinct models with different centroid counts and parameter columns.

It is clear that increasing the centroid count will give models with higher perfiormance

- **2 centroids**
  - Attempted all the columns

```
Within cluster sum of squares by cluster:
[1]  8934.321 13982.051 62323.158  9106.071  2916.200
 (between_SS / total_SS =  68.5 %)
```

```
> model$tot.withinss
[1] 97261.8
```

```
     Gender      Age Annual.Income Spending.Score
1 1.607143 40.17857      78.89286       17.42857
2 1.461538 32.69231      86.53846       82.12821
3 1.378947 44.89474      48.70526       42.63158
4 1.500000 24.82143      28.71429       74.25000
5 1.300000 41.00000     109.70000       22.00000
> 
```

- **5 centroids**

```
Within cluster sum of squares by cluster:
[1]  4627.739  8954.087 30157.266 13982.051 17678.472
 (between_SS / total_SS =  75.6 %)
```

```
> model$tot.withinss
[1] 75399.62
> model$centers
     Gender      Age Annual.Income Spending.Score
1 1.391304 25.52174      26.30435       78.56522
2 1.391304 45.21739      26.30435       20.91304
3 1.417722 43.08861      55.29114       49.56962
4 1.461538 32.69231      86.53846       82.12821
5 1.527778 40.66667      87.75000       17.58333
> 
```

- **10 centroids**

```
Within cluster sum of squares by cluster:
 [1]  2025.412  2464.571 14815.312  3214.300  2353.333  4156.000  3095.867  2608.100  5785.368  1314.938
 (between_SS / total_SS =  86.5 %)
```

- Available components:

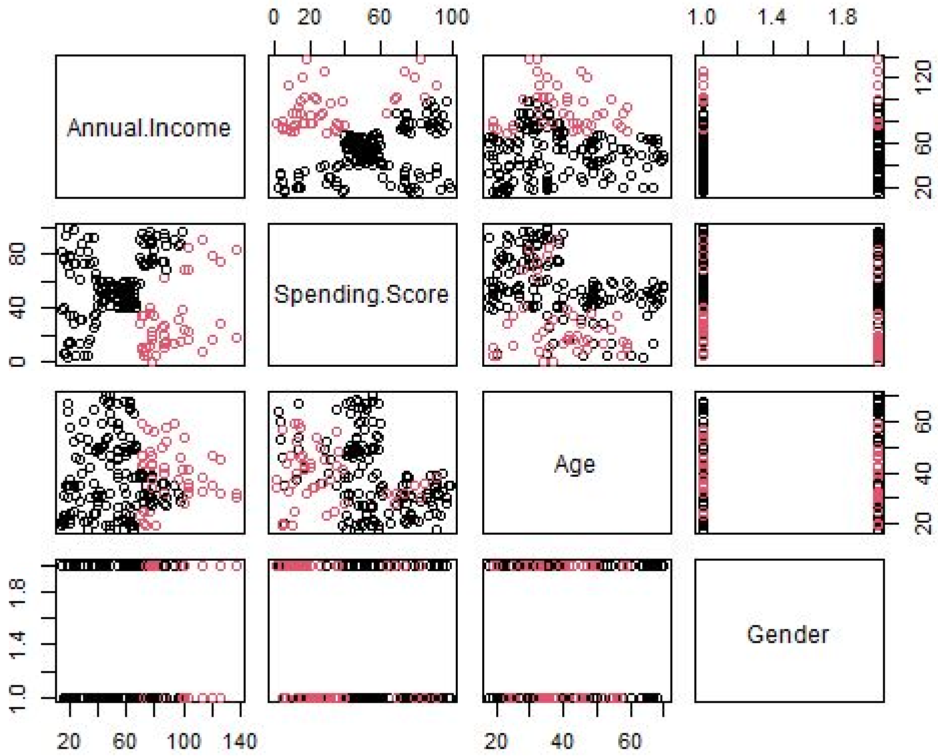
```
> model$tot.withinss
[1] 41833.2
> model$centers
      Gender      Age Annual.Income Spending.Score
1   1.352941 46.94118      65.41176       45.82353
2   1.571429 64.38095      53.33333       50.23810
3   1.562500 41.00000      89.40625       15.59375
4   1.400000 24.85000      24.95000       81.00000
5   1.333333 23.57143      62.14286       47.95238
6   1.482759 32.86207      78.55172       82.17241
7   1.400000 27.06667      38.60000       52.13333
8   1.400000 32.20000     109.70000       82.00000
9   1.368421 46.15789      26.10526       17.42105
10  1.375000 47.37500      47.81250       47.75000
> 
```

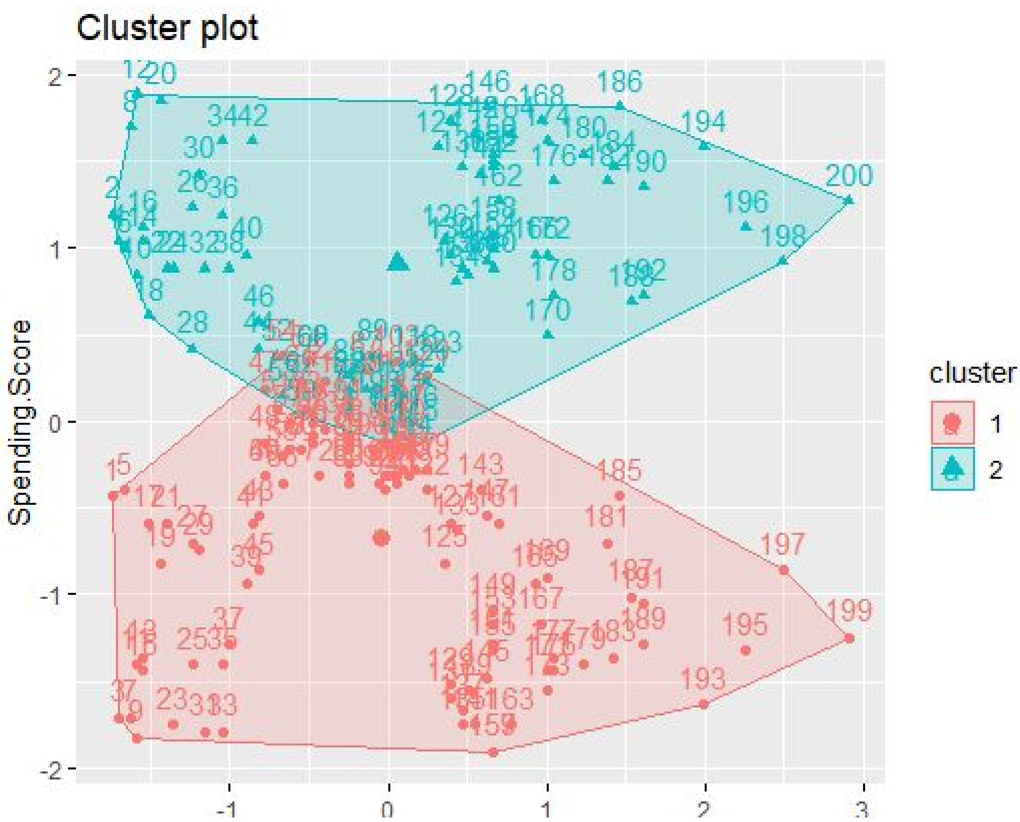
3.4. Evaluation

When considering purely the tot.withinss value the higher K value models are better but I would like to draw the attention to the "5 Centroid" model which has a higher withinss value but in the graphs makes the most sense in terms of clustering of the consumer groups
I have rerun all of these multiple times and chosen their best possible outputs

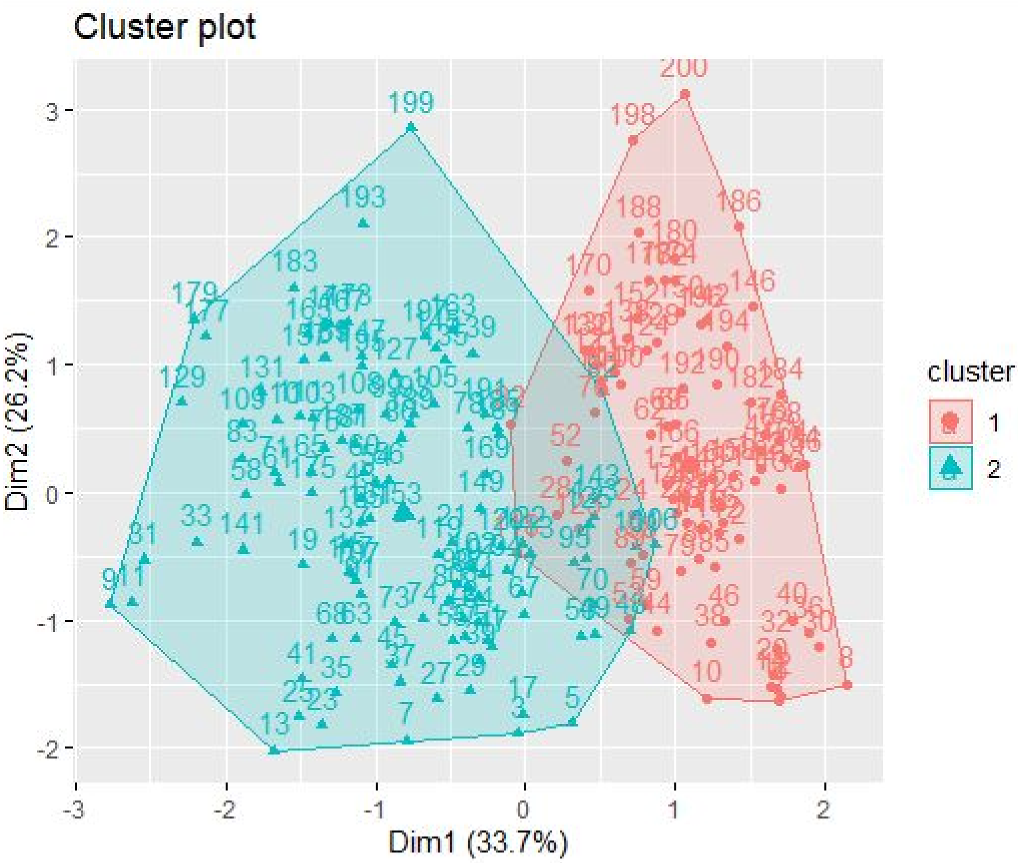
- **2 centroids**
  - This model gets the largest withinss values by far
  - The groupings do not make as much sense

```
> model$tot.withinss
[1] 225083.8
>
```





Cluster plot
Spending.Score
cluster
1
2

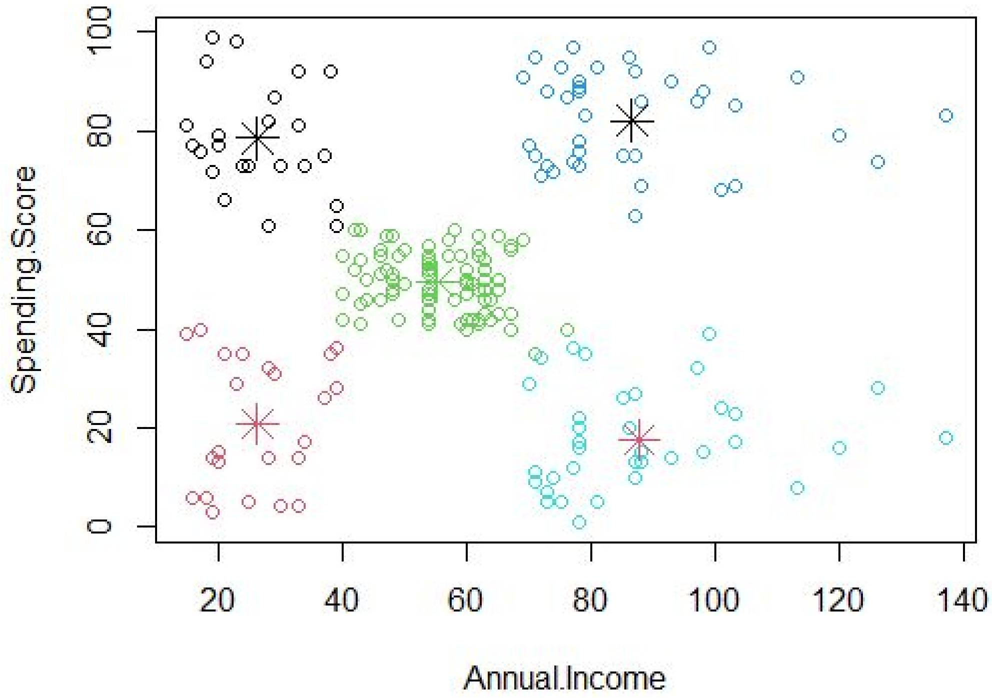


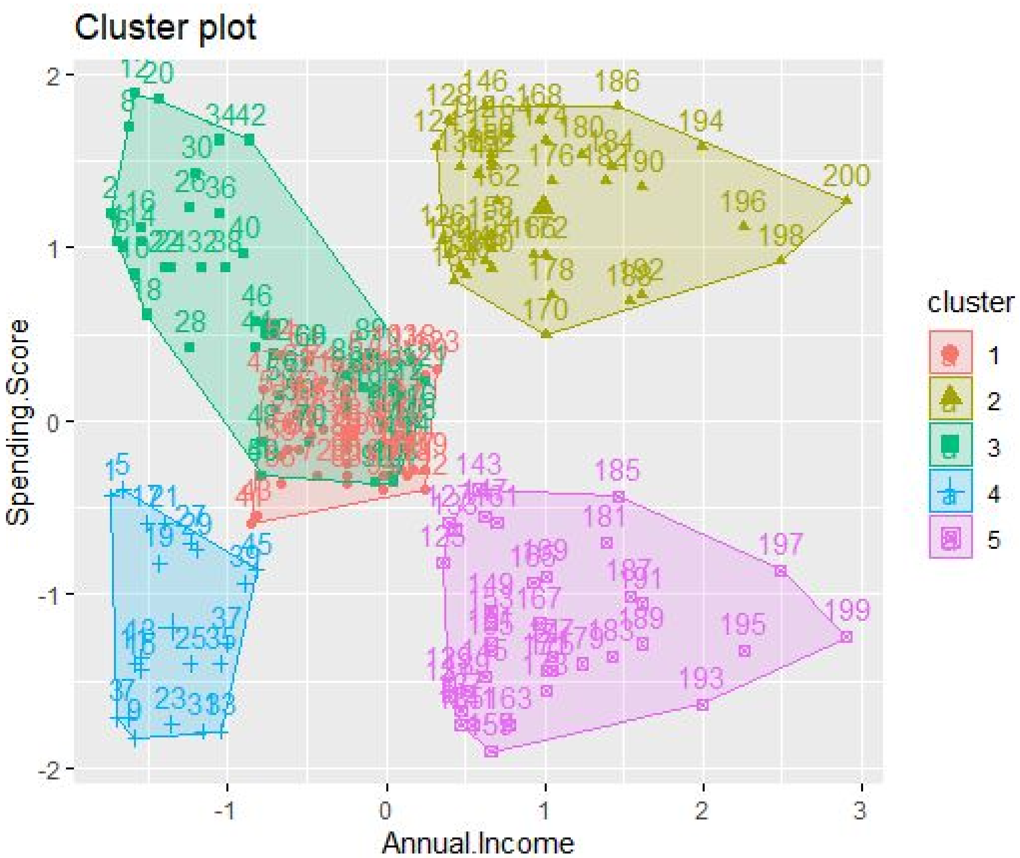
Cluster plot
Dim2 (26.2%)
Dim1 (33.7%)
cluster
1
2

- **5 centroids**

- This one receives much lower values through the generations but when mapping Spending.Score to Annual.Income provides a really great grouping and correlation as seen below

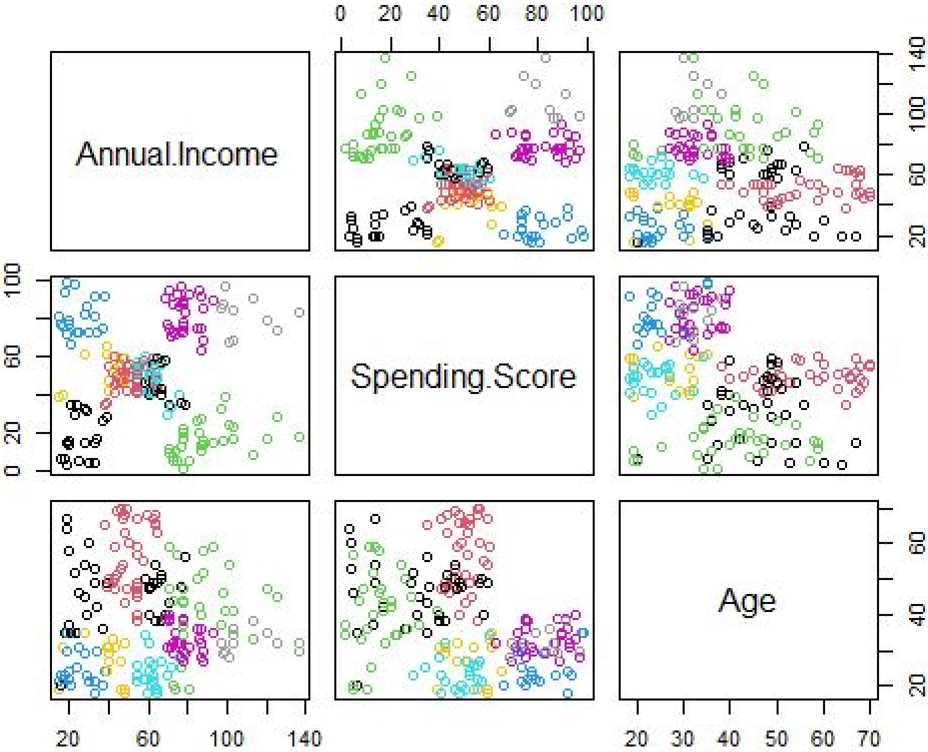
```
> model$tot.withinss
[1] 75399.62
```

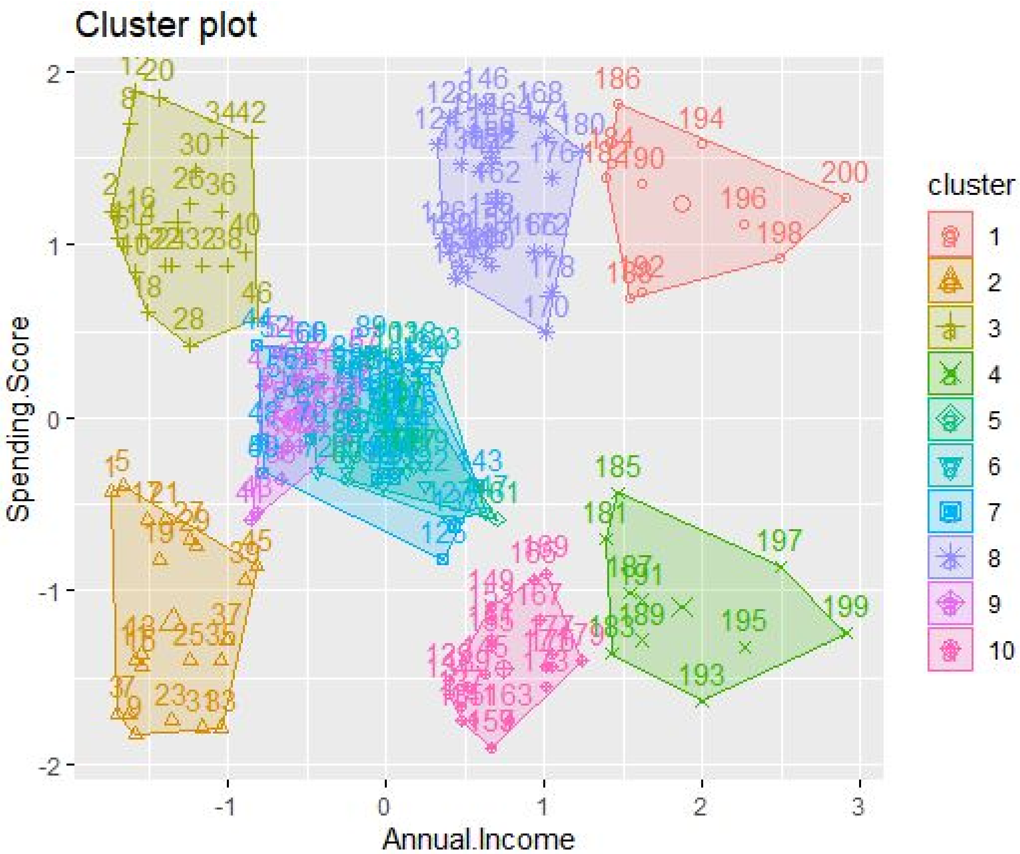


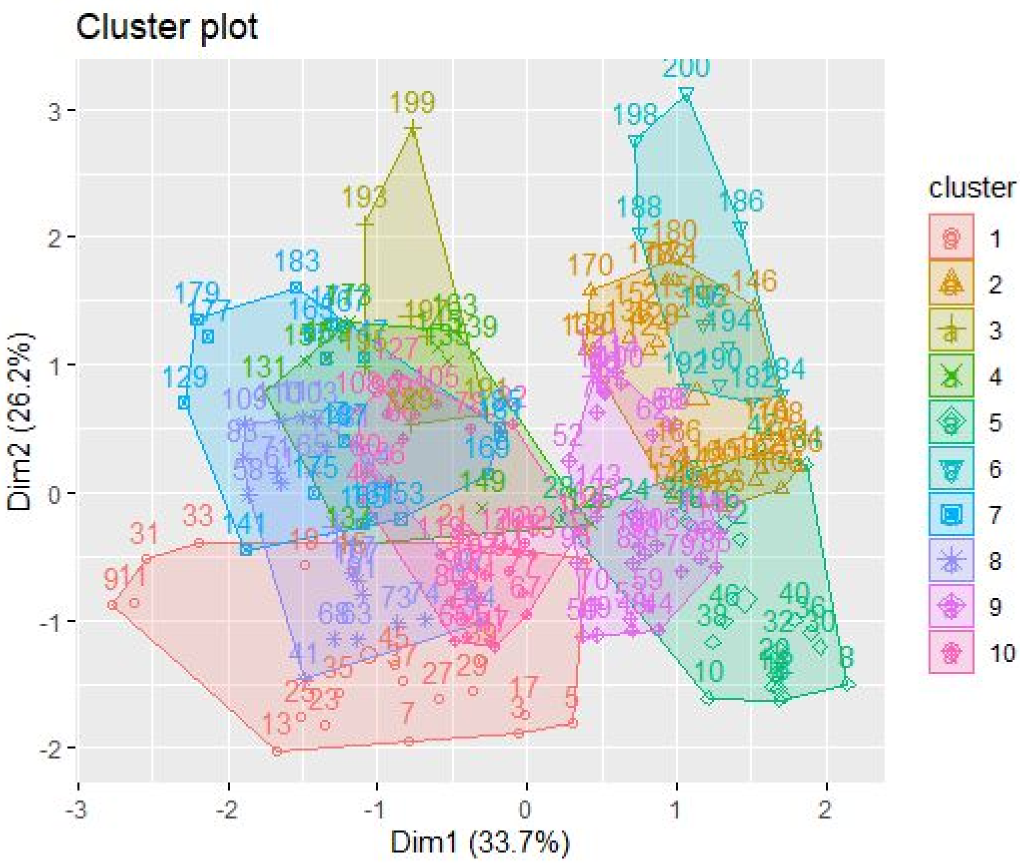


- **10 centroids**
    - There is definitely some good clustering happening here that is much more fine tuned than the earlier ones
    - The withinss values however are much lower

```
> model$tot.withinss
[1] 41833.2
> 
```




Cluster plot
Spending.Score
Annual.Income
cluster
1
2
3
4
5
6
7
8
9
10


Cluster plot
Dim2 (26.2%)
Dim1 (33.7%)
cluster
1
2
3
4
5
6
7
8
9
10

References

- https://www.r-bloggers.com/2013/01/calculating-a-gini-coefficients-for-a-number-of-locales-at-once-in-r/
- https://stackoverflow.com/questions/30058362/r-convert-from-categorical-to-numeric-for-knn
- https://quantdev.ssri.psu.edu/sites/qdev/files/kNN_tutorial.html
- https://uc-r.github.io/kmeans_clustering