SAINT PETERSBURG STATE UNIVERSITY

Faculty of Applied Mathematics and Control Processes

Mathematical Game Theory and Statistical Decisions Department

Applied Statistics in R

Exam

Professor: Parilina Elena M.

Student: Orlov Ivan M., 19.M09-пу

Saint Petersburg

2020

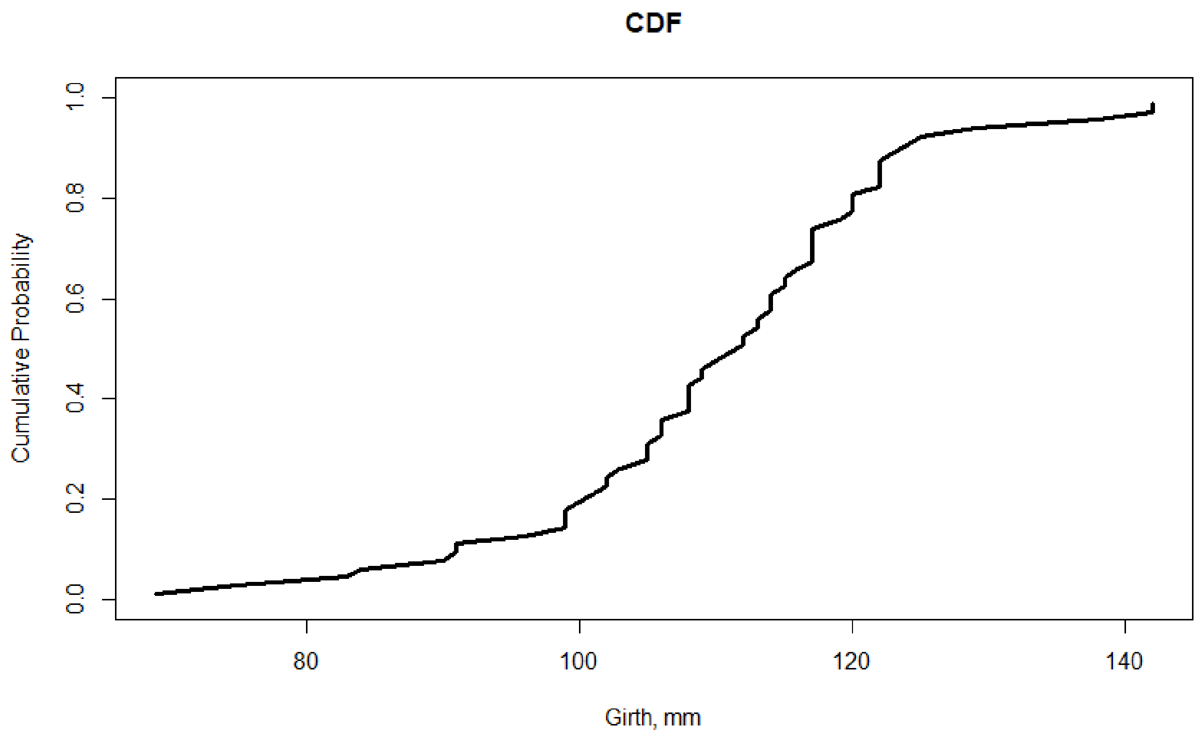
Contents

1. Descriptive statistics.

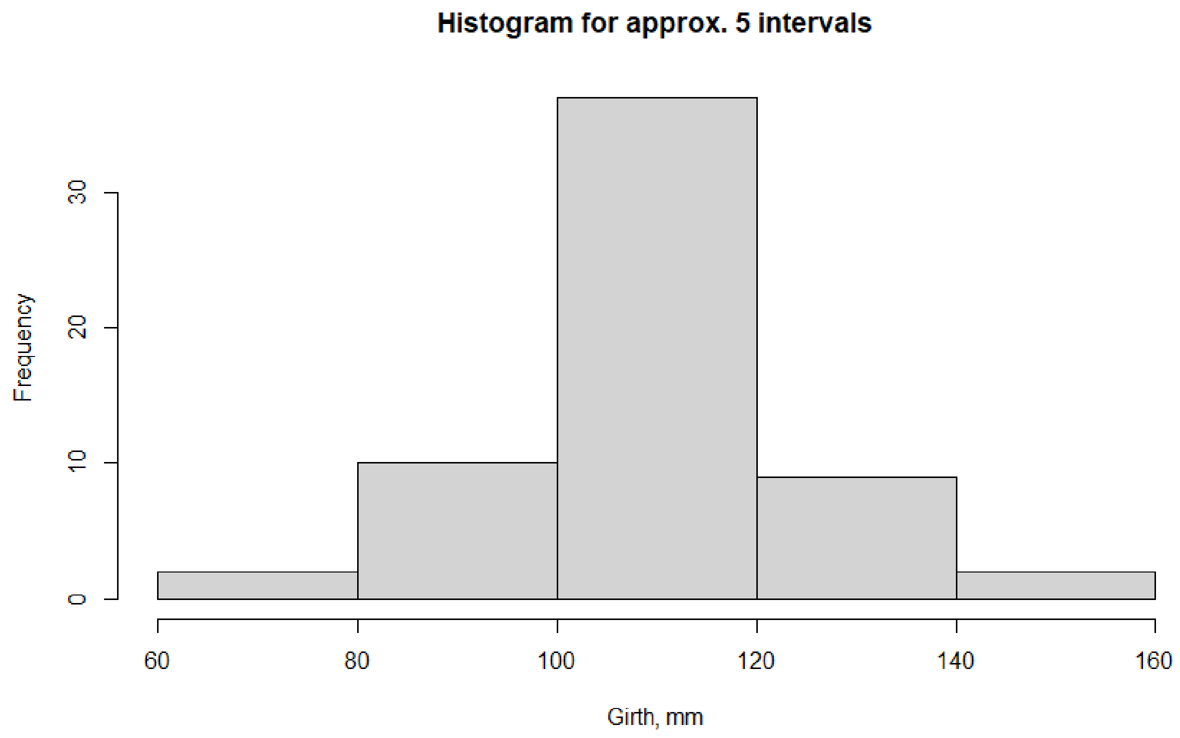
The data represent trunk girth (mm) of a random sample of 60 four-year-old apple trees at East Malling Research Station (England). Reference: S.C. Pearce, University of Kent at Canterbury

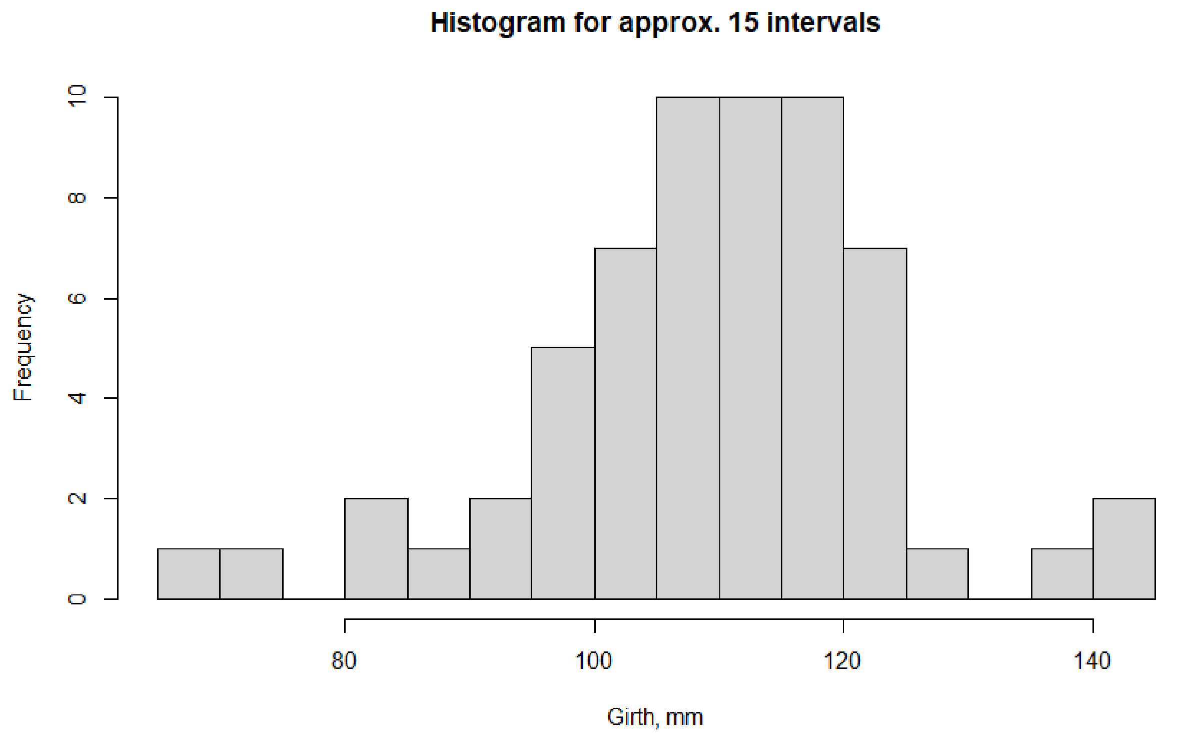
Statistics	**Value**
n	60
mean	110.0167
variance	197.237
standard deviation	14.04411
median	111.5
0.25-quantile	102.75
0.5-quantile	111.5
0.75-quantile	117.5
mode	117
minimal value	69
maximal value	142
skewness	-0.39736
kurtosis	0.832999

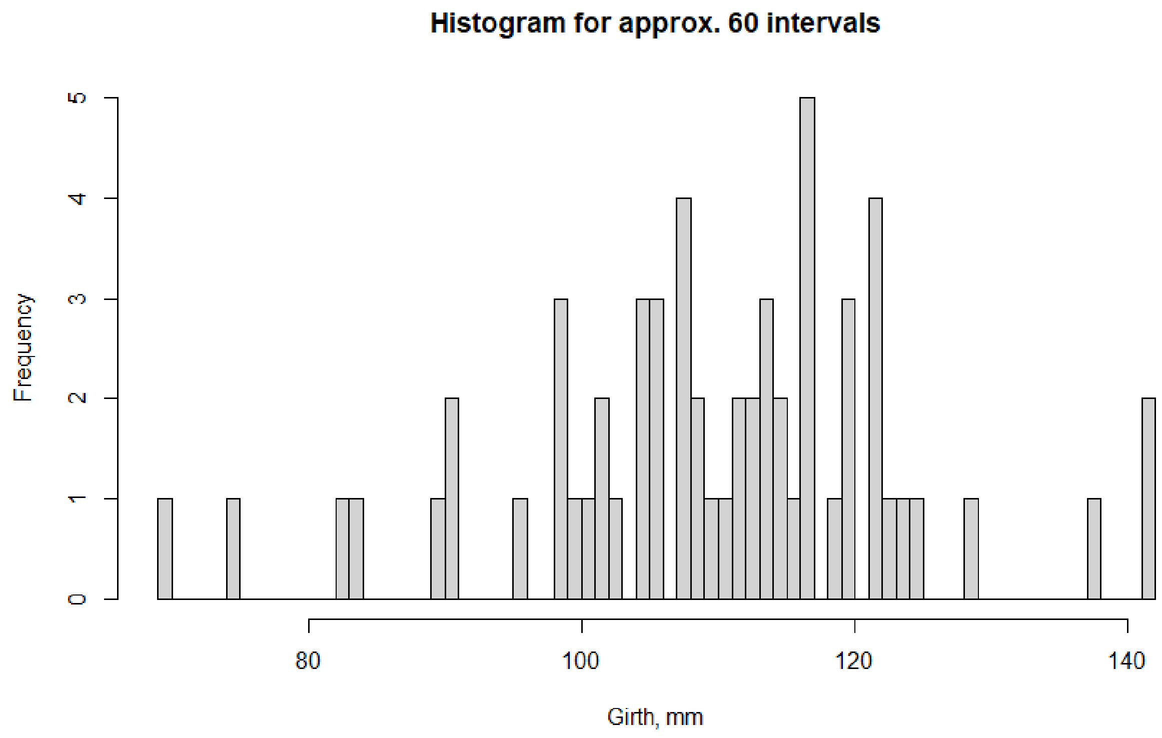
The task asked for “skewness, asymmetry”, but skewness is a measure of asymmetry and is often paired with kurtosis (plus that’s what ds.statistics() function returns), so these are the measures in the report.



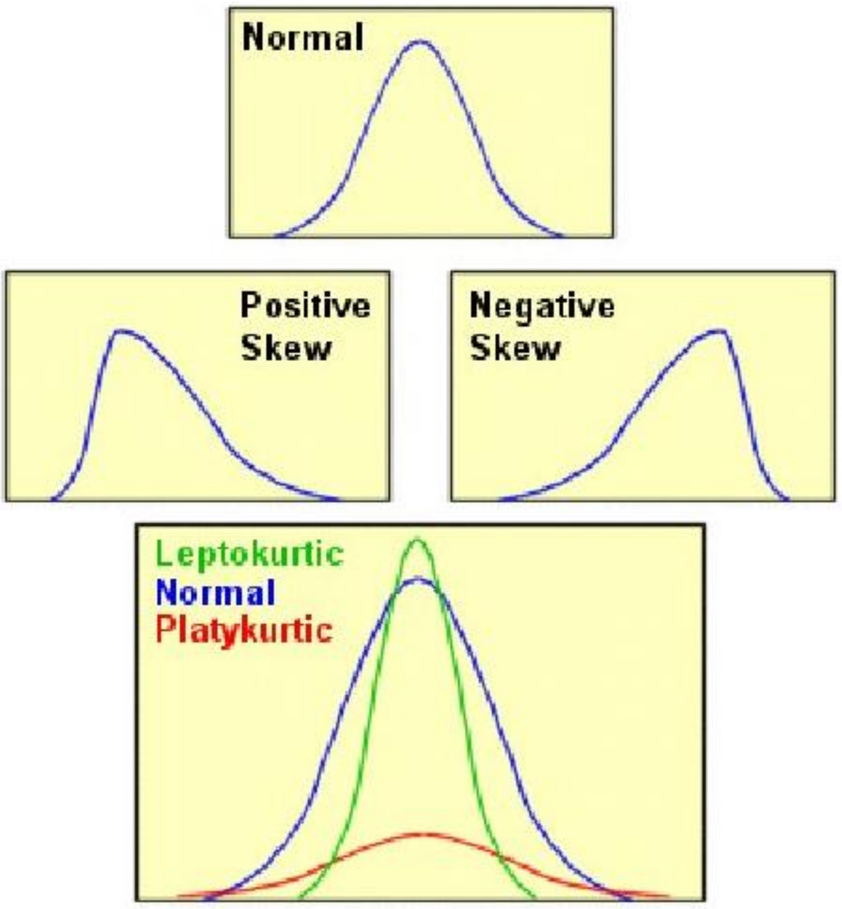
Cumulative distribution function graph shows that the biggest grow is in the interval between 100 and 120 mm: almost 60% of girths are in there.

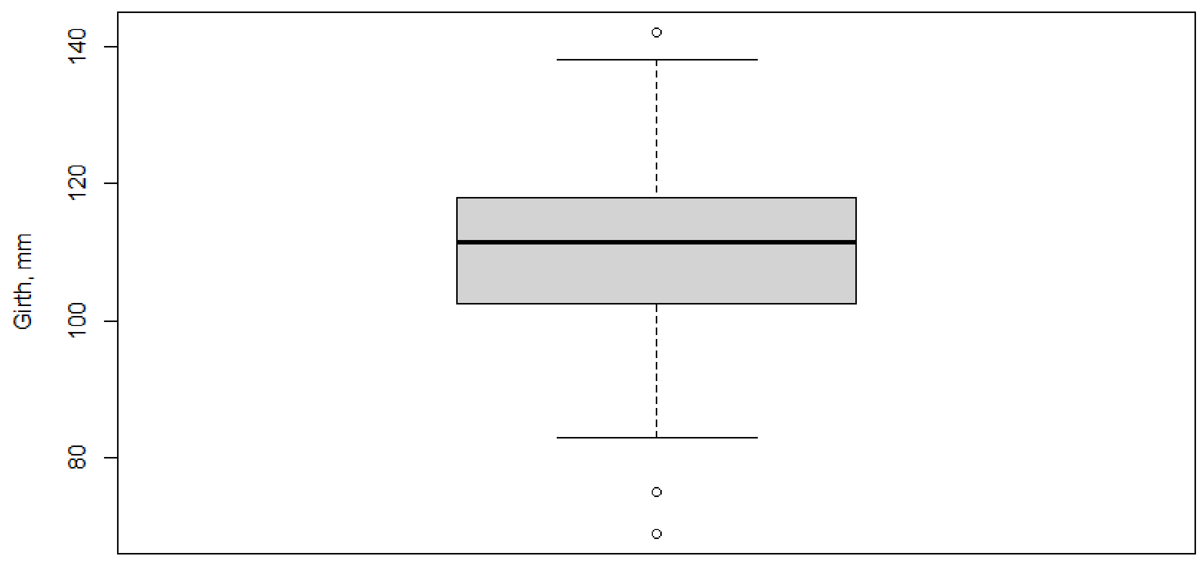

Histogram for approx. 5 intervals
Frequency
0
10
20
30
60
80
100
120
140
160
Girth, mm


Histogram for approx. 15 intervals
Frequency
0
2
4
6
8
10
80
100
120
140
Girth, mm



The histogram confirms that data has negative skewness and kurtosis that is less than that of normal distribution (i.e. platykurtic).
R hist() function computes the number of bins as close as possible to the specified breaks, but makes sure the graph remains "pretty".

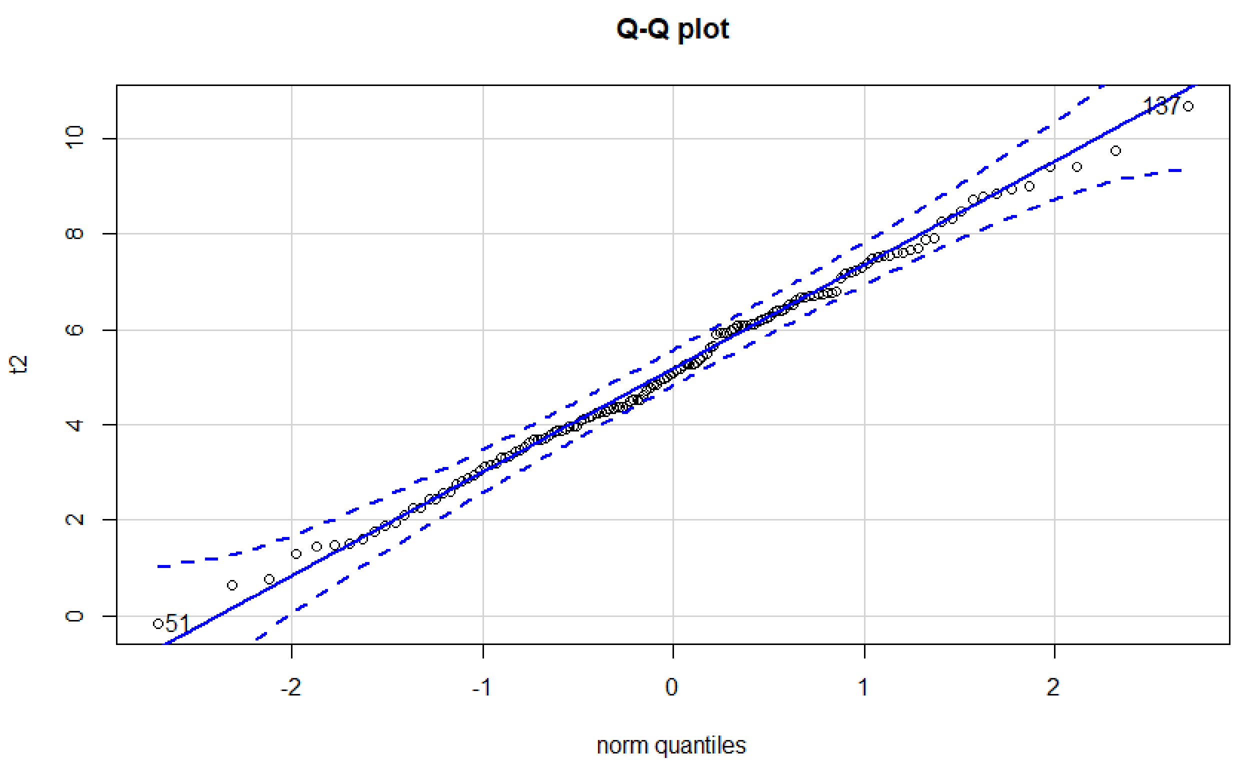




The boxplot of the graph shows us that 60-3=57 of 60 values are inside the interval *[q_{25}-1.5S; q_{75}+1.5S]*, which is bounded by horizontal lines ending the dashed vertical lines.
It can be seen that median q_{50} is closer to q_{75} than to q_{25}. The same result can be obtained from the summary table (*117.5-111.5=6<8.75=111.5-102.75*).

2. Statistical goodness-of-fit tests.

The simplest (and probably the most unreliable) way to check for normality is Q-Q plot:



The dashed lines outline the envelope for normality with 0.9 level of confidence and visual examination suggests that, since all points of data are inside the envelope, the data itself is normally distributed. However, this is not enough and we need real tests.
uniNorm function no longer exists in R 4.0.1, MVN::mvn() is used instead.

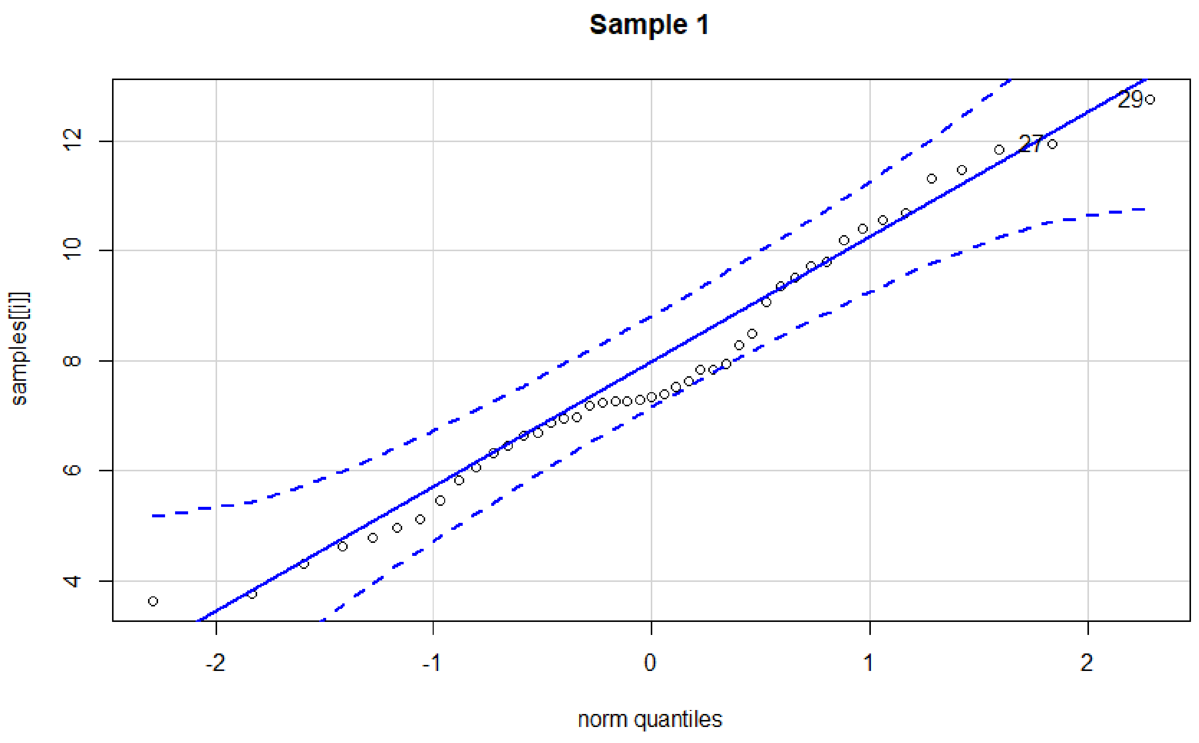
Statistics	Value
n	145
Mean	5.148819
Std.Dev	2.104814
Median	5.083483
Min	-0.1551615
Max	10.67074
25th	3.726937
75th	6.656663
Skew	0.04403965
Kurtosis	-0.388785

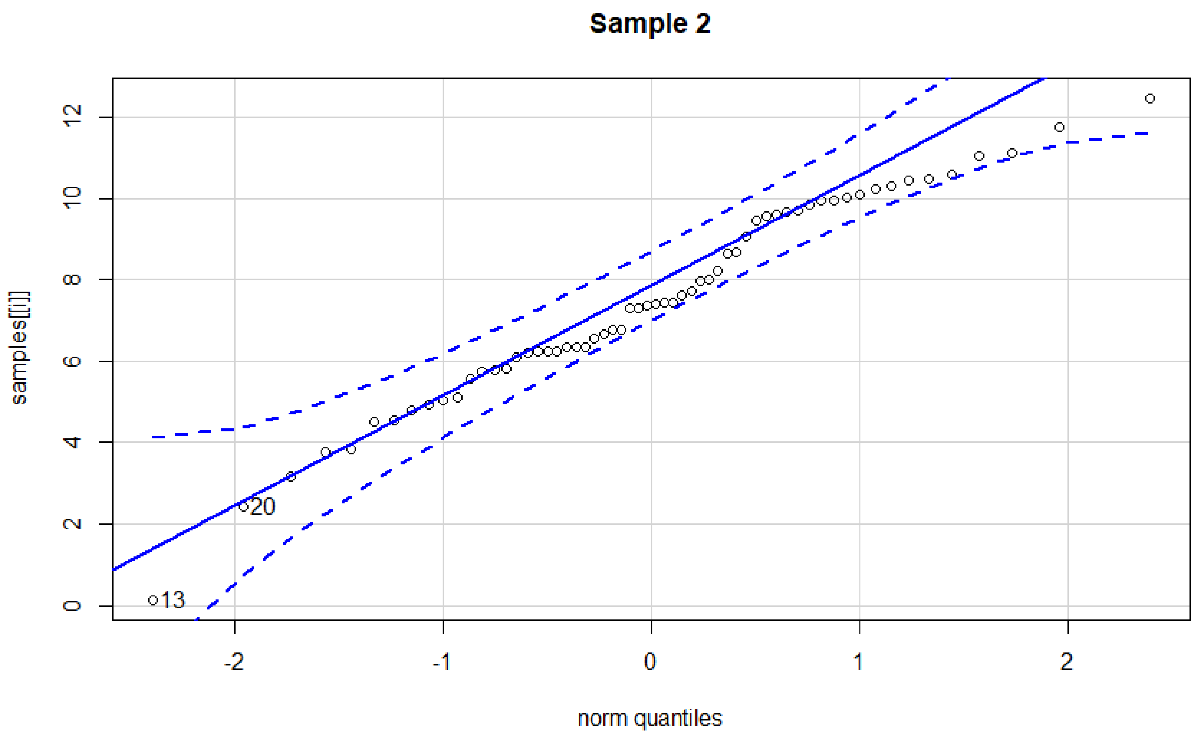
The skewness and kurtosis are close to zero (as in normal distribution) which also might be a good sign.
pearson.test was run twice, with adjust = TRUE (default) and with adjust =FALSE with the real p-value somewhere between the results.

Test	**Statistic**	**p-value**	**Normality**
Lilliefors (Kolmogorov-Smirnov)	0.0511	0.466	YES
Pearson chi-square (adjusted)	9.655172	0.6461832	YES
Pearson chi-square	9.655172	0.7869349	YES
Cramer-von Mises	0.0408	0.6648	YES
Anderson-Darling	0.2117	0.8542	YES

No test found a proof of non-normality on 0.9 level of confidence.

3. Comparing two or more samples.





Again, visual testing suggest that both samples are normal.
Shapiro-Wilk normality test gives p-values of 0.322654 and 0.3356359 that both declare absence of proofs of non-normality on 0.95 level of confidence.

Two Sample Student’s t-test gives the following results

```
t = 0.65262, df = 103, p-value = 0.5155
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.6324875  1.2529050
sample estimates:
mean of x mean of y
 7.787951  7.477742
```

F-test to compare two variances gives the following results:

```
F = 0.8464, num df = 44, denom df = 59, p-value = 0.5669
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.4897799 1.4966227
sample estimates:
ratio of variances
        0.8463991
```

Two-sample Kolmogorov-Smirnov test gives the following results:

```
D = 0.13889, p-value = 0.6606
alternative hypothesis: two-sided
```

Wilcoxon rank sum test gives the following results:

```
W = 1424, p-value = 0.6341
alternative hypothesis: true location shift is not equal to 0
```

The results are:

1) distributions' means are equal;
2) distributions' variances are equal;
3) distributions of two random variables are equal (proven by both K-S and W tests).

(At least the evidence that this is not true is not found.)

4. Regression analysis.

The data (Y, X1, X2) are for each patient.
Y = systolic blood pressure
X1 = age in years
X2 = weight in pounds

Linear:

Y ~ X1+X2	Estimate	Std.Error	t-value	Pr(>|t|)	sig.code
(Intercept)	30.9941	11.9438	2.595	0.03186	*
X1	0.8614	0.2482	3.47	0.00844	**
X2	0.3349	0.1307	2.563	0.03351	*

Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 2.318 on 8 degrees of freedom
Multiple R-squared: 0.9768, Adjusted R-squared: 0.9711
F-statistic: 168.8 on 2 and 8 DF, p-value: 2.874e-07

Test	P-value
Breusch-Pagan	0.3120339
Durbin-Watson	0.4101264

The coefficients might be interpreted in the following way: for every 1 year increase in age the systolic blood pressure increases by 0.8614; for every 1 pound increase in weight the systolic blood pressure increases by 0.3349. The explanation of intercept as systolic blood pressure of a person who is 0 years old and weighs 0 pounds is meaningless; intercept does not have an actual meaning.

The model is good: the p-values of coefficients are more than 0.001 and some are more than 0.01, p-value of the equation is less than 0.001; there are no proofs of heteroscedasticity or autocorrelation, and R-squared (which is the proportion of variance of Y predicted from Xs) and adjusted R-squared are close to 1. The problem is that correlation between Xs is too high:

	Y	X1	X2
Y	1	0.978693	0.970564
X1	0.978693	1	0.946052
X2	0.970564	0.946052	1

	X1	X2
VIF	9.525022	9.525022

Removing Xs from the regression helps but not much: R-squared gets less, but in X1 regression significance got better for coefficients and the model.

Y ~ X1	Estimate	Std.Error	t-value	Pr(>|t|)	sig.code
(Intercept)	58.7055	6.4524	9.098	7.81E-06	***
X1	1.4632	0.1023	14.3	1.71E-07	***

Residual standard error: 2.949 on 9 degrees of freedom
Multiple R-squared: 0.9578, Adjusted R-squared: 0.9532
F-statistic: 204.5 on 1 and 9 DF, p-value: 1.708e-07

Test	P-value
Breusch-Pagan	0.497905
Durbin-Watson	0.0133535

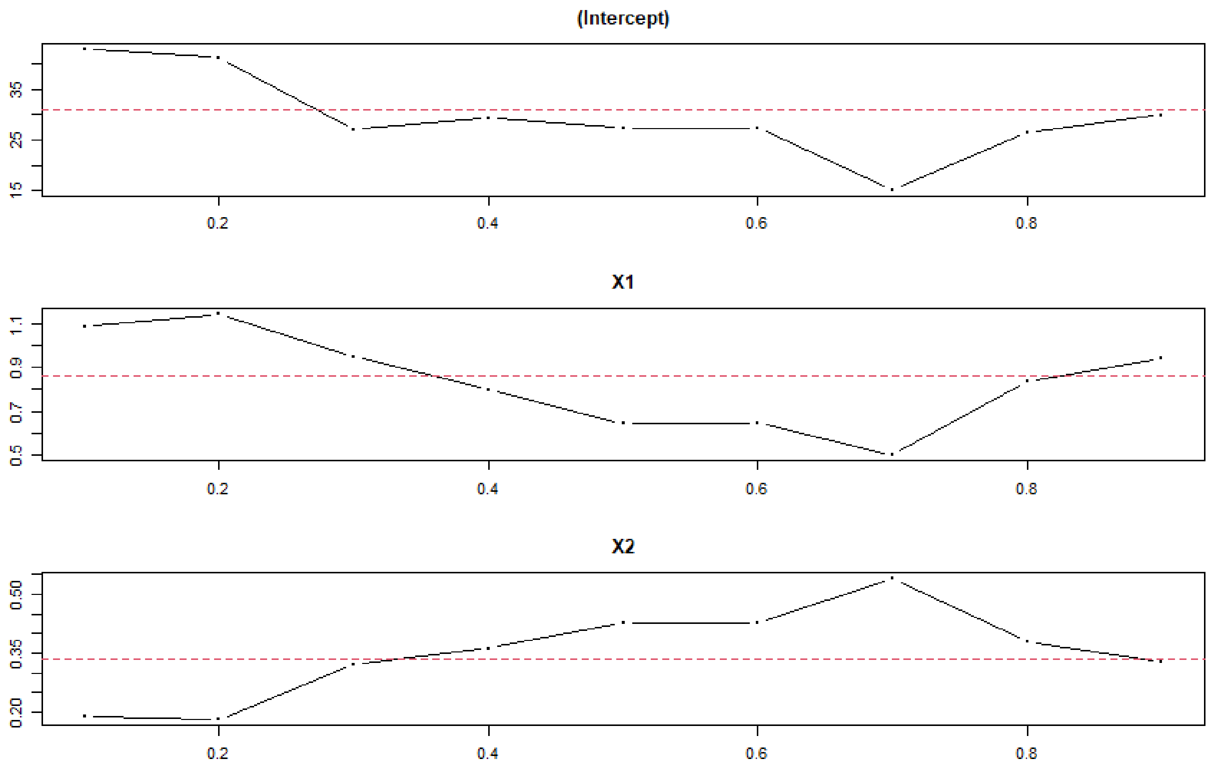
In case of X2 it gets worse:

Y ~ X2	Estimate	Std.Error	t-value	Pr(>|t|)	sig.code
(Intercept)	1.14161	12.36447	0.092	0.928	
X2	0.76384	0.06318	12.09	7.23E-07	***

Residual standard error: 3.459 on 9 degrees of freedom
Multiple R-squared: 0.942, Adjusted R-squared: 0.9356
F-statistic: 146.2 on 1 and 9 DF, p-value: 7.227e-07

Test	P-value
Breusch-Pagan	0.5431526
Durbin-Watson	0.8227221

Quantile:

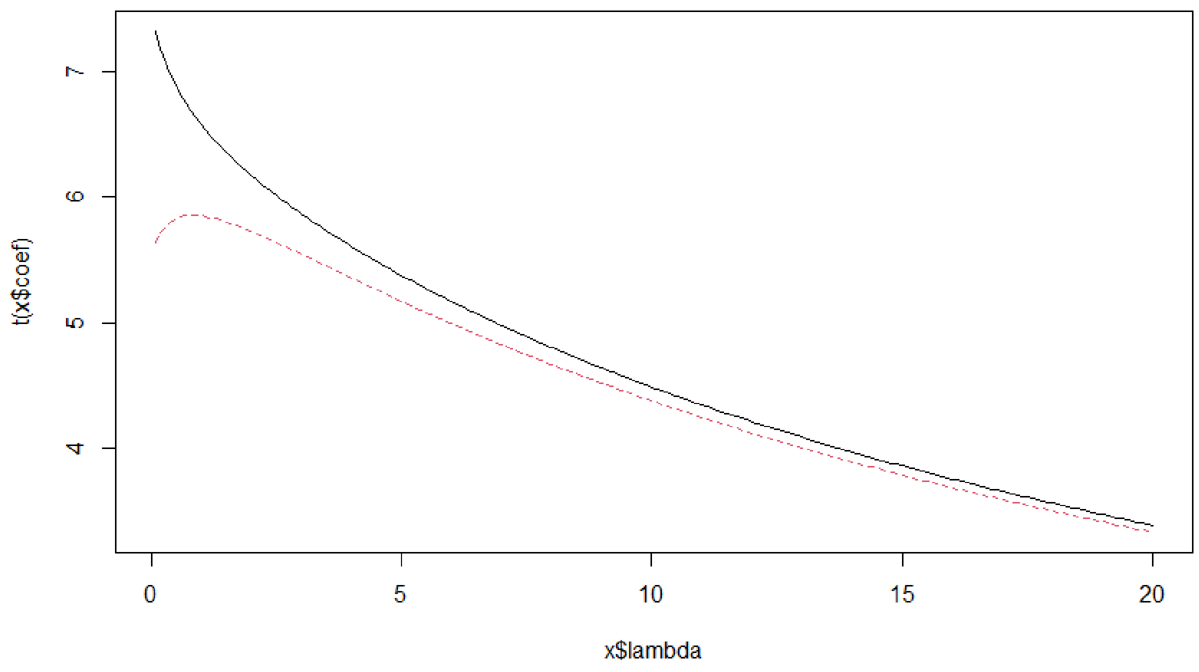


The graph shows how coefficients change with tau changing from 0.1 to 0.9. Dashed line is the OLS regression coefficient.

tau=0.1	**Value**	**Std.Error**	**t-value**	**Pr(>|t|)**
(Intercept)	42.85774	19.50102	2.19772	0.0592
X1	1.08787	0.30036	3.62189	0.00676
X2	0.18828	0.187	1.00689	0.34347
tau=0.2	**Value**	**Std.Error**	**t-value**	**Pr(>|t|)**
(Intercept)	41.13115	20.83687	1.97396	0.08383
X1	1.14754	0.35556	3.22742	0.0121
X2	0.18033	0.20596	0.87554	0.4068
tau=0.3	**Value**	**Std.Error**	**t-value**	**Pr(>|t|)**
(Intercept)	27.03704	17.6027	1.53596	0.1631
X1	0.95062	0.41336	2.29975	0.05049
X2	0.32099	0.20397	1.57372	0.1542
tau=0.4	**Value**	**Std.Error**	**t-value**	**Pr(>|t|)**
(Intercept)	29.30435	19.32343	1.51652	0.16786
X1	0.7971	0.47614	1.67408	0.13265
X2	0.36232	0.23908	1.51549	0.16811
tau=0.5	**Value**	**Std.Error**	**t-value**	**Pr(>|t|)**
(Intercept)	27.36782	35.07284	0.78031	0.45767
X1	0.64368	0.78965	0.81514	0.43858
X2	0.42529	0.40509	1.04985	0.32446
tau=0.6	**Value**	**Std.Error**	**t-value**	**Pr(>|t|)**
(Intercept)	27.36782	16.36911	1.67192	0.13308
X1	0.64368	0.4241	1.51774	0.16756
X2	0.42529	0.20921	2.03282	0.07652
tau=0.7	**Value**	**Std.Error**	**t-value**	**Pr(>|t|)**
(Intercept)	15.18654	16.67271	0.91086	0.38899
X1	0.49847	0.38598	1.29145	0.23261
X2	0.53823	0.19467	2.76475	0.02449
tau=0.8	**Value**	**Std.Error**	**t-value**	**Pr(>|t|)**
(Intercept)	26.3787	21.30348	1.23823	0.25073
X1	0.83432	0.52131	1.60043	0.14817
X2	0.3787	0.25802	1.46769	0.18037
tau=0.9	**Value**	**Std.Error**	**t-value**	**Pr(>|t|)**
(Intercept)	29.91045	18.3925	1.62623	0.14255
X1	0.9403	0.46425	2.02541	0.07741
X2	0.32836	0.22815	1.43925	0.18803

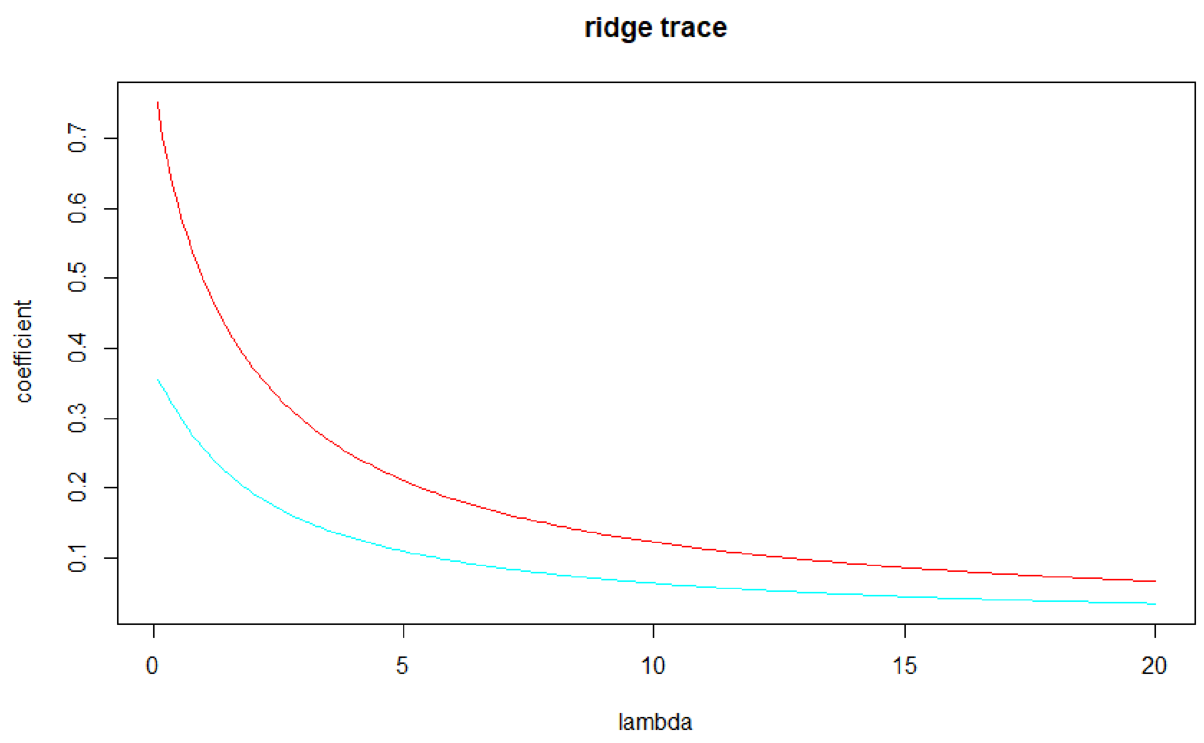
These tau cannot provide a better significane than a linear model.

Ridge:



lambda	intercept	X1	X2
0.1	30.91604	0.841689	0.341577
0.2	30.99722	0.826091	0.346157
4.9	49.98273	0.621472	0.314331
5.0	50.35494	0.618947	0.31323
5.1	50.72458	0.616447	0.312135
9.9	65.81617	0.518362	0.266158
10.0	66.0829	0.516671	0.265331
10.1	66.34797	0.514992	0.264509
14.9	77.37479	0.445796	0.230124
15.0	77.57389	0.444556	0.2295
15.1	77.77191	0.443323	0.228879
19.9	86.15548	0.391304	0.202547
20.0	86.30957	0.390351	0.202062

lm.ridge() function does not calculate significance levels, but we have another function for ridge regression.



Significance of different coefficients calculated with ridge::linearRidge() is very low, all of them are between 0.0001-0.0004, even lower than in the linear (yet we don't have information about the significance of intercept coefficient):

Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

lambda	intercept	X1	X2	sig(X1)	code(X1)	sig(X2)	code(X2)
0.1	34.0954	0.7517	0.3541	2.22E-16	***	1.93E-13	***
0.2	38.9496	0.7029	0.3448	<2e-16	***	<2e-16	***
4.9	114.9989	0.2141	0.1114	1.47E-05	***	1.84E-05	***
5.0	115.5039	0.211	0.1098	1.61E-05	***	2.01E-05	***
5.1	115.9945	0.208	0.1082	1.77E-05	***	2.20E-05	***
9.9	129.80666	0.12362	0.06443	0.00015	***	0.000174	***
10.0	129.97642	0.12258	0.06389	0.000153	***	0.000178	***
10.1	130.14336	0.12156	0.06336	0.000157	***	0.000182	***
14.9	135.82621	0.0869	0.04532	0.000307	***	0.00035	***
15.0	135.91038	0.08639	0.04505	0.00031	***	0.000353	***
15.1	135.99356	0.08588	0.04479	0.000313	***	0.000356	***
19.9	139.09068	0.06701	0.03495	0.000434	***	0.00049	***
20.0	139.1408	0.0667	0.03479	0.000437	***	0.000492	***

The results are different probably due to different methods of calculation (for example, there is another ridge function glmnet() which standardizes the y variable and uses the mean squared errors instead of sum of squared errors, so its results are different from lm.ridge()).

The data is insufficient for building a very good model, there are only 11 values. Age and weight should be good predictors of blood pressure but they are highly correlated so only one should be chosen and paired with other predictor(s).

X = Age
Y = "1" if person has kids and "0" if person has no kids

Logit:

Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Coefficient	Estimate	Std.Error	z-value	Pr(>|z|)	sig.code
(Intercept)	-5.30945	1.13365	-4.683	2.82E-06	***
X	0.11092	0.02406	4.61	4.02E-06	***

Wald test:
X2 = 22.0, df = 2, P(> X2) = 1.7e-05

The significance of both the model and coefficients is less than 0.001.

Confusion matrix for default cutoff

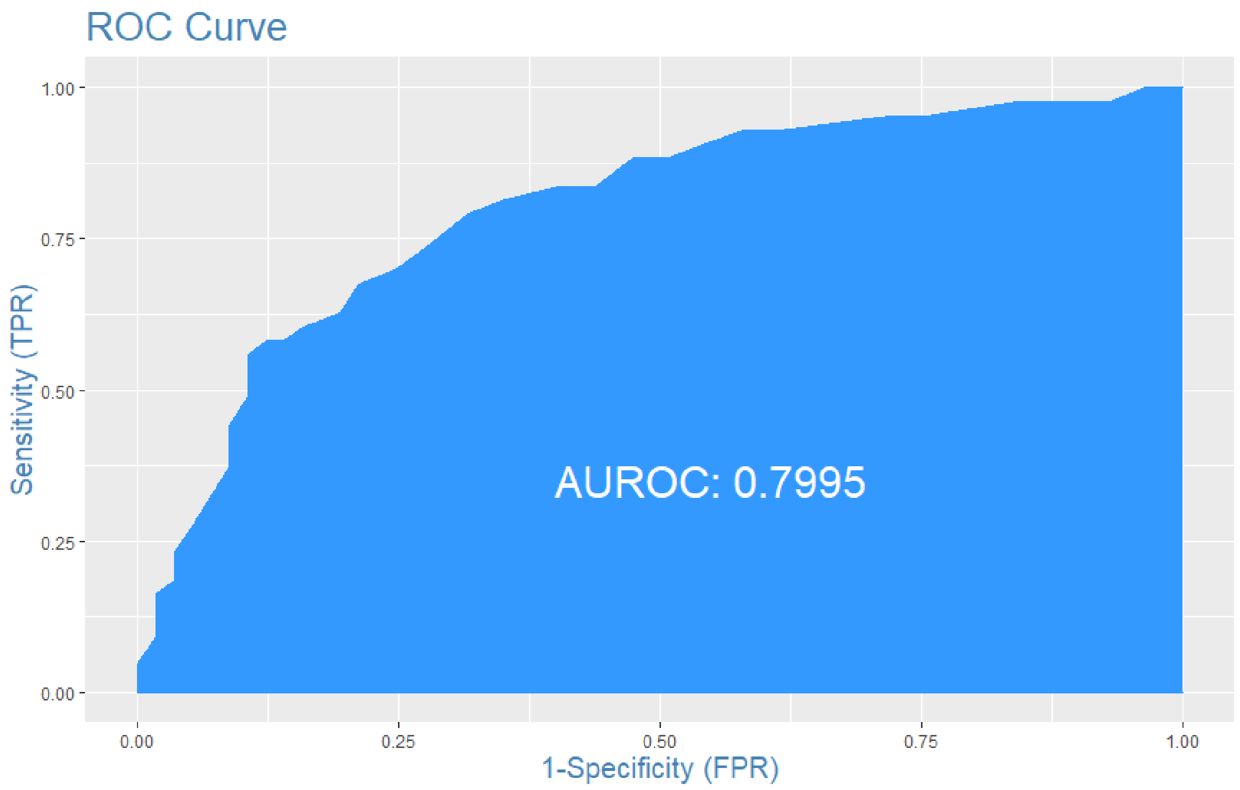
	0	1
0	45	14
1	12	29

Optimal_cutoff = 0.5924646

Confusion matrix for optimal cutoff

	0	1
0	50	18
1	7	25

Change in cutoff reduces the number of false positives by 5, but increases the number of false negatives by 4.



Area Under the Receiver Operating Characteristics tells how model is capable of distinguishing between classes, the closer AUROC to 1, the better.

Probit:

Coefficient	Estimate	Std.Error	z-value	Pr(>|z|)	sig.code
(Intercept)	-3.14573	0.6246	-5.036	4.74E-07	***
X	0.0658	0.01335	4.93	8.20E-07	***

Wald test:

X2 = 25.4, df = 2, P(> X2) = 3.1e-06

The significance of both the model and coefficients is less than 0.001.

Confusion matrix for default cutoff

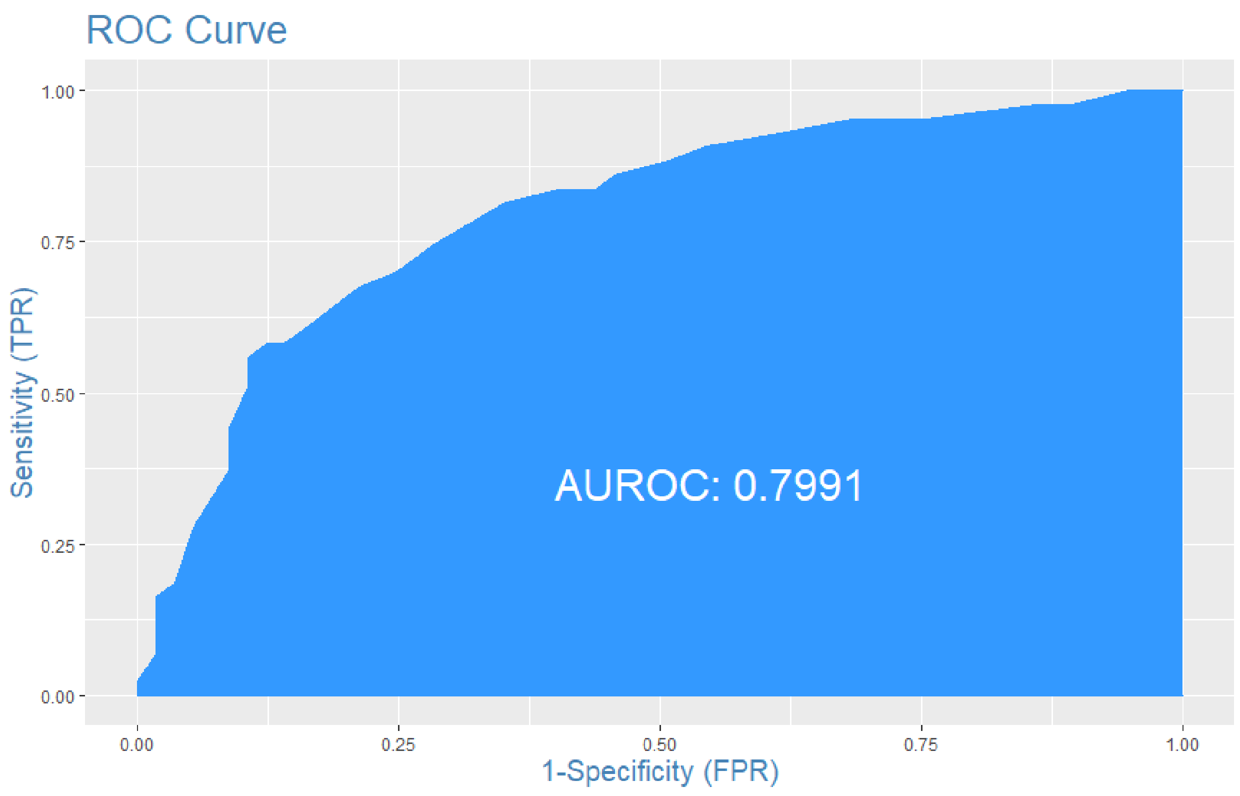
	0	1
0	45	14
1	12	29

Optimal_cutoff = 0.5613517

Confusion matrix for optimal cutoff

	0	1
0	50	18
1	7	25

Change in cutoff yields exactly the same resutls as in logit.



AUROC is slightly worse than in logit (by 0.0004).

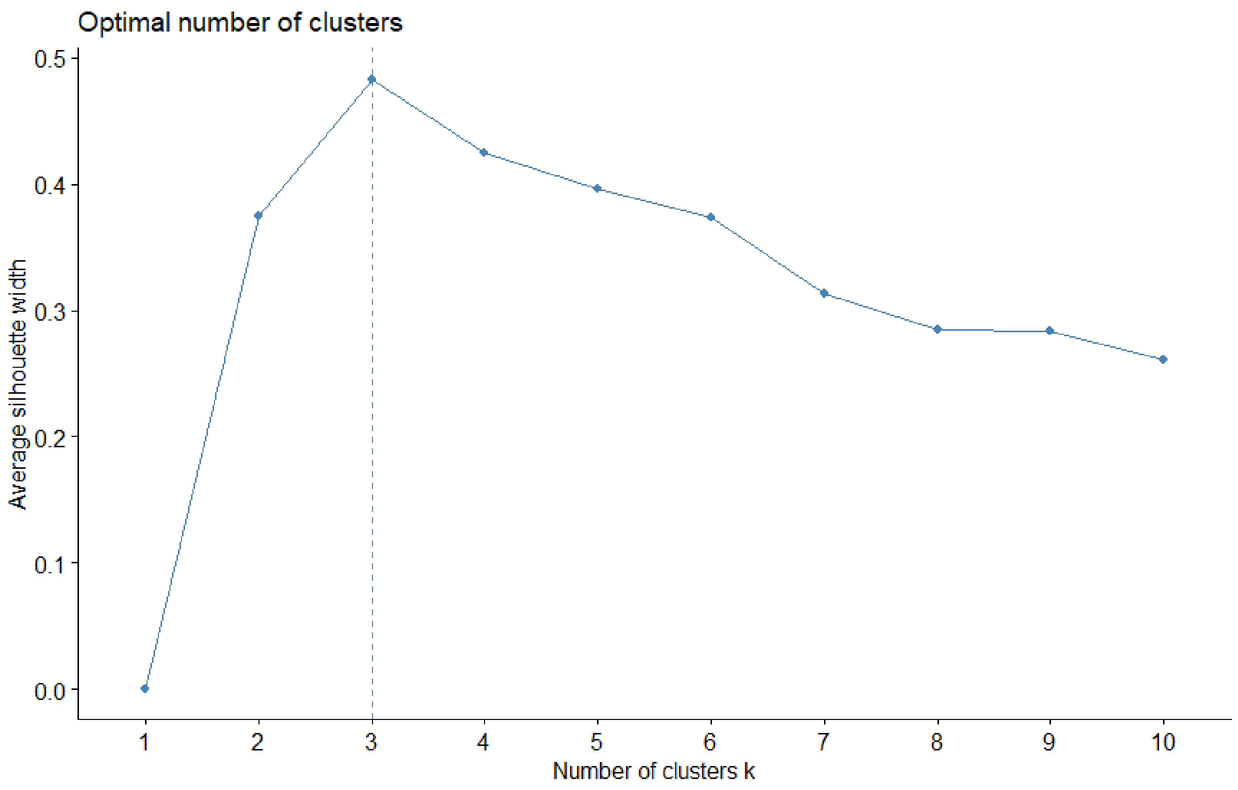
5. Cluster analysis.

Table 10.12 Attributes and Prices of Beer Brands for Cluster Analysis

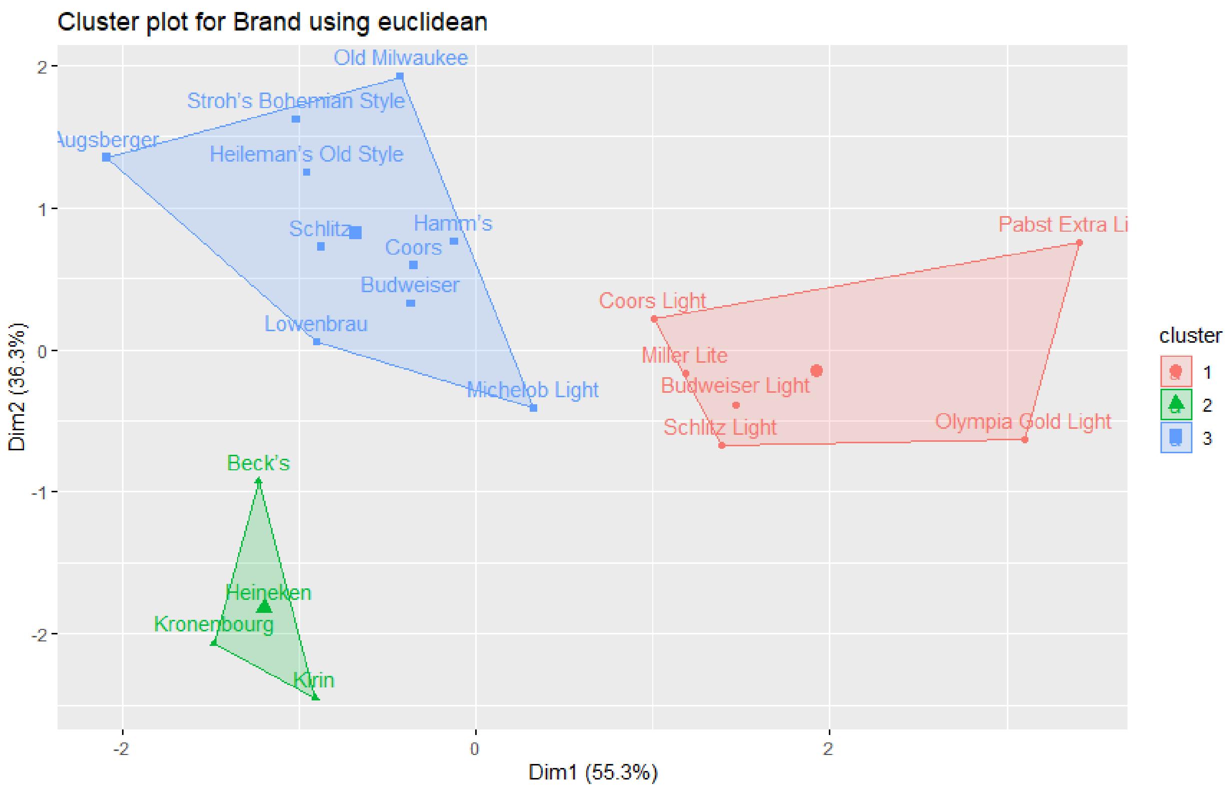
What parameters to use in cluster analysis?

Brand	**Calories (12 oz.)**	**Sodium (mg/12 oz)**	**Alcohol (%)**	**Price (Wholesale)**

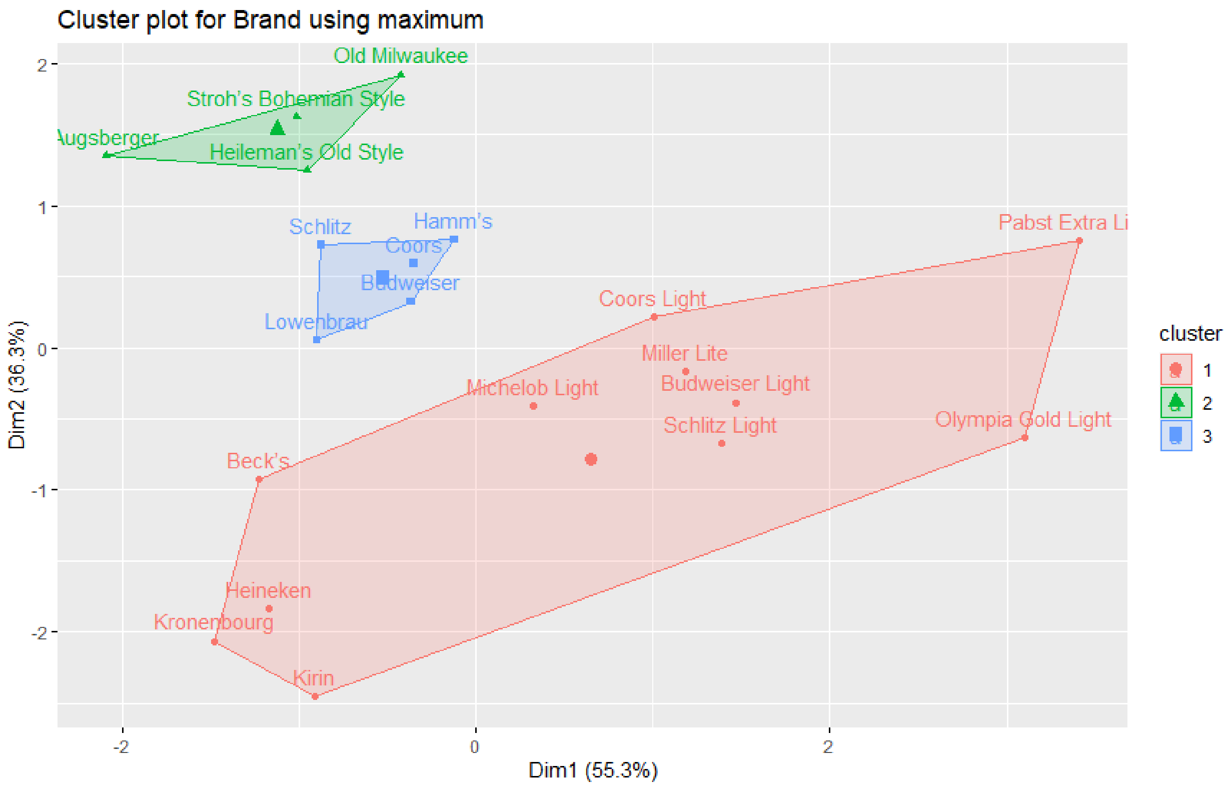
If we are just people who are interested in our health, we (should probably not drink in the first place but) need to analyze sodium and alcohol, calories if we are on a diet.
If we are poor but craving beer, we might only consider price.
If we are analysts from the big company, we need to analyze the whole market – all parameters. For what? For example, we might want to create a product, which will be placed in the smallest cluster in order to dominate the least competitive environment.



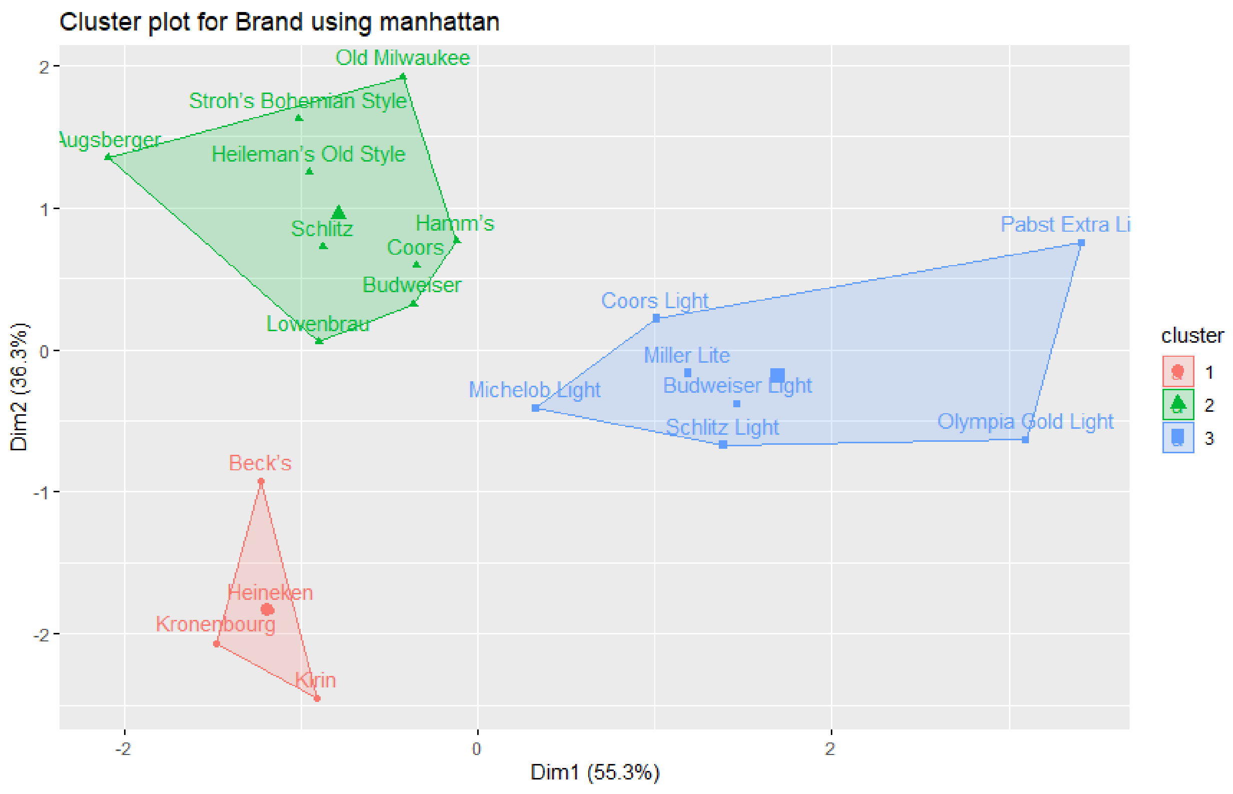
The "silhouette" method is used in kmeans fbiz_nbclust() function because in it the optimal number is the one with the biggest y.



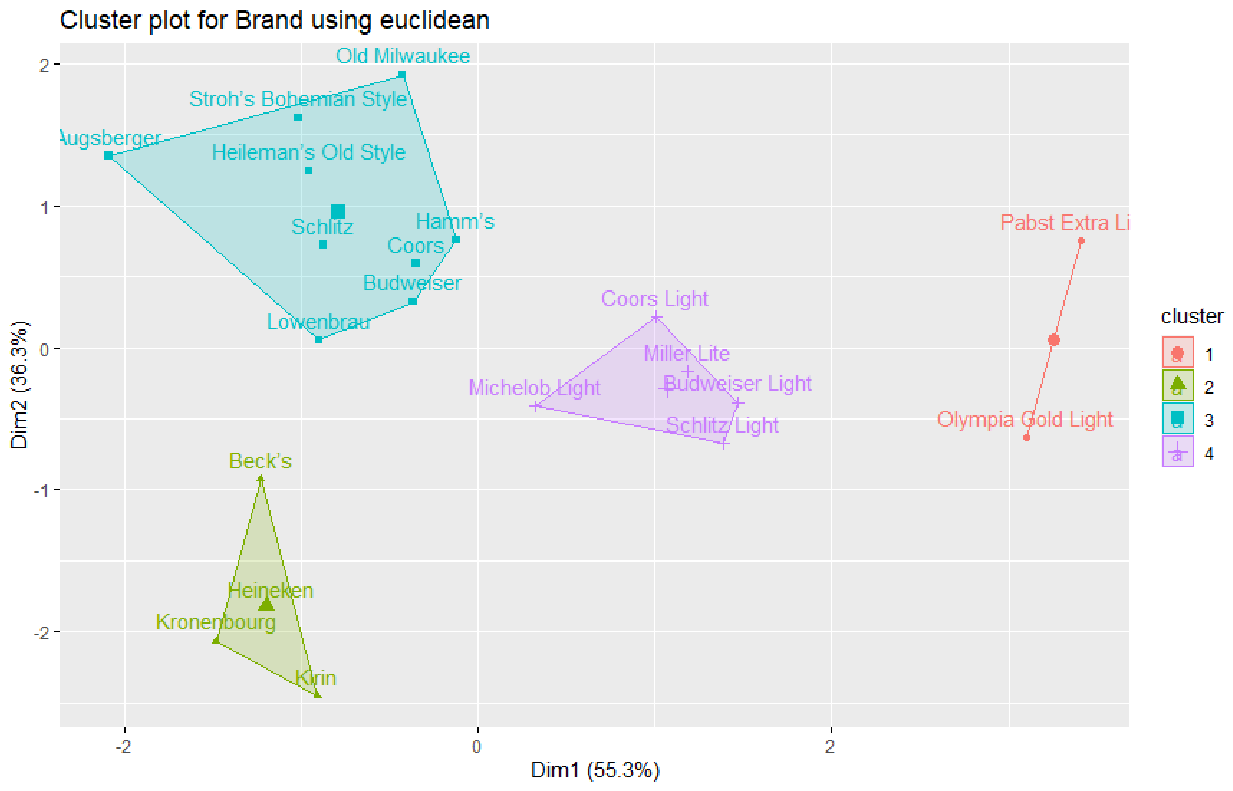
Cluster plot for Brand using euclidean
Old Milwaukee
Stroh's Bohemian Style
Augsberger
Heileman's Old Style
Schlitz
Hamm's
Coors
Budweiser
Lowenbrau
Michelob Light
Coors Light
Miller Lite
Budweiser Light
Schlitz Light
Pabst Extra Li
Olympia Gold Light
Beck's
Heineken
Kronenbourg
Kirin
Dim1 (55.3%)
Dim2 (36.3%)
cluster
1
2
3

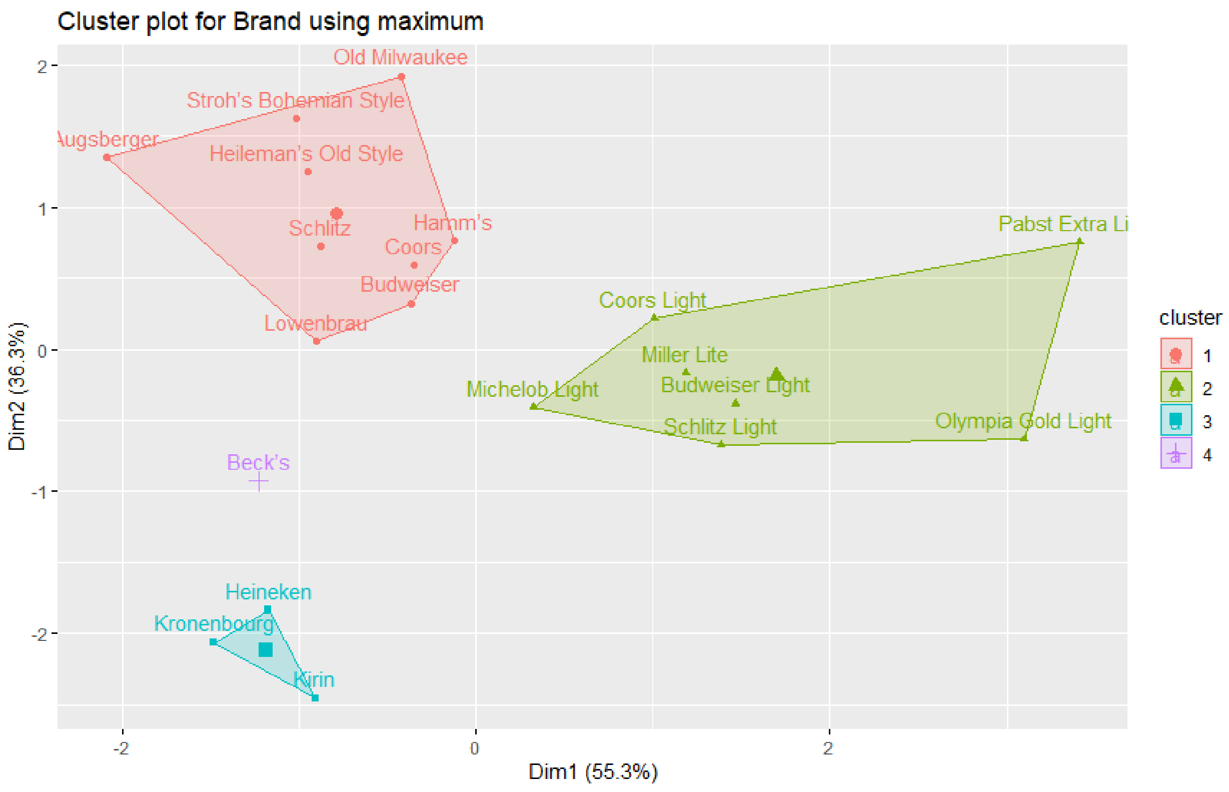


Cluster plot for Brand using maximum
Old Milwaukee
Stroh's Bohemian Style
Augsberger
Heileman's Old Style
Schlitz
Hamm's
Coors
Budweiser
Lowenbrau
Coors Light
Miller Lite
Michelob Light
Budweiser Light
Schlitz Light
Pabst Extra Li
Olympia Gold Light
Beck's
Heineken
Kronenbourg
Kirin
Dim1 (55.3%)
Dim2 (36.3%)
cluster
1
2
3

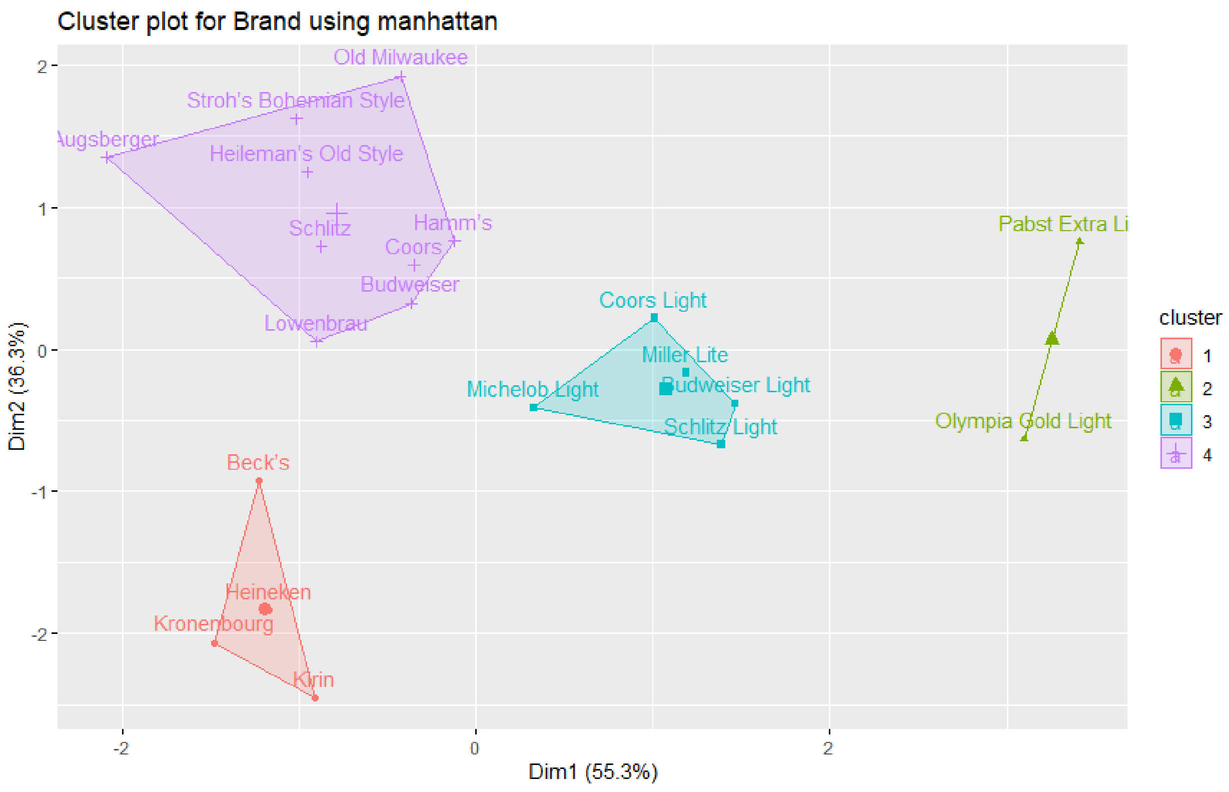


Distance measure strongly affects the forming of clusters in that case: no measure yields the same results.
Number of clusters should be anywhere between 1 and 21 (in both of these situations cluster analysis is meaningless since either everything or nothing is the same), so it is better to use numbers between 2 and 20. Subjectively, on manhattan plot the cluster on the right seems too big and the top one can be divided in two, so let us try 4 and 5.
With *n=4:*



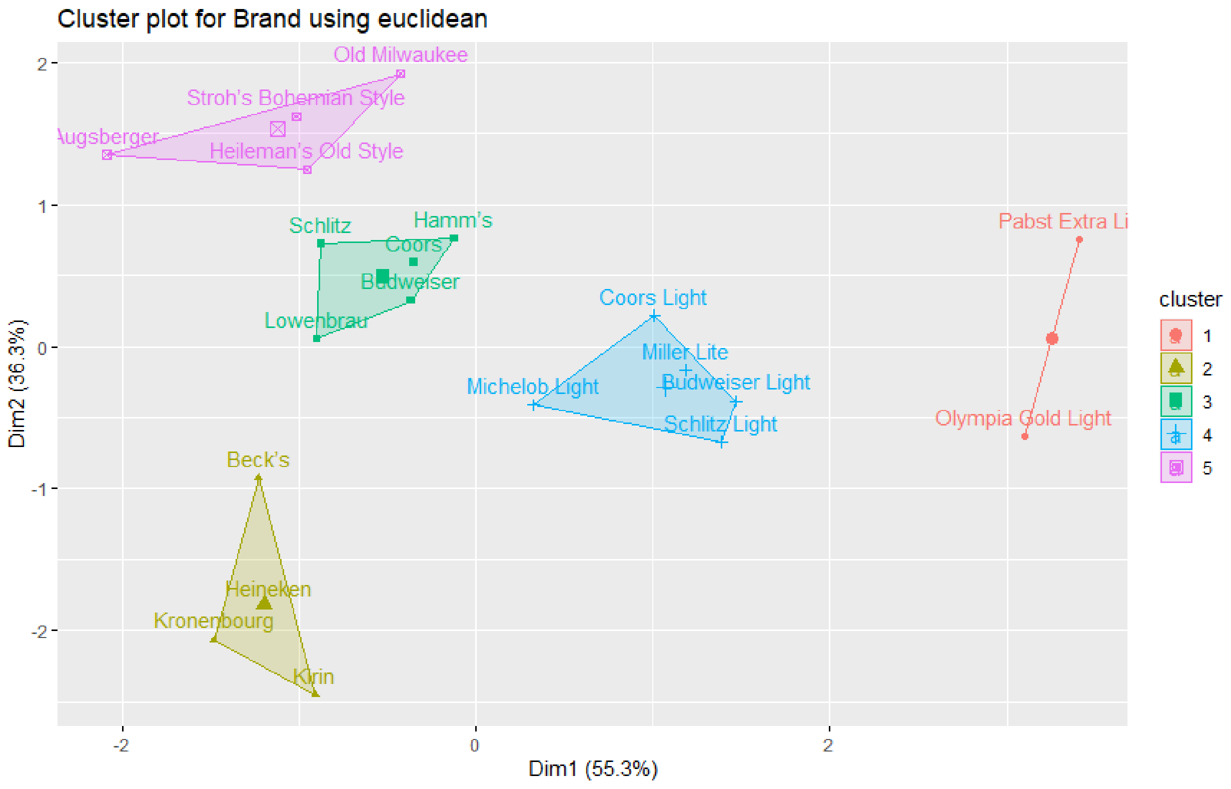


Cluster plot for Brand using maximum
Old Milwaukee
Stroh's Bohemian Style
Augsberger
Heileman's Old Style
Schlitz
Hamm's
Coors
Budweiser
Lowenbrau
Coors Light
Pabst Extra Li
Miller Lite
Michelob Light
Budweiser Light
Schlitz Light
Olympia Gold Light
Beck's
Heineken
Kronenbourg
Kirin
Dim2 (36.3%)
Dim1 (55.3%)
cluster
1
2
3
4

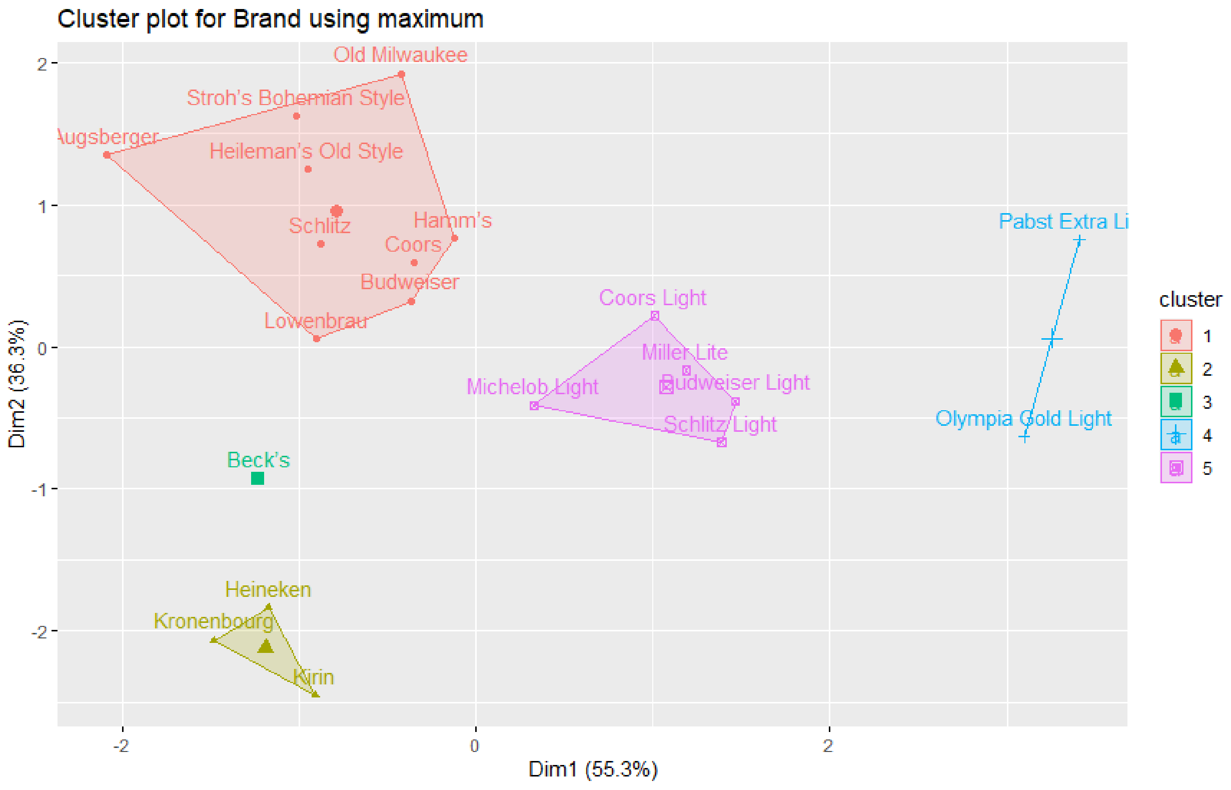


Cluster plot for Brand using manhattan
Old Milwaukee
Stroh's Bohemian Style
Augsberger
Heileman's Old Style
Schlitz
Hamm's
Coors
Budweiser
Lowenbrau
Coors Light
Pabst Extra Li
Miller Lite
Michelob Light
Budweiser Light
Schlitz Light
Olympia Gold Light
Beck's
Heineken
Kronenbourg
Kirin
Dim2 (36.3%)
Dim1 (55.3%)
cluster
1
2
3
4

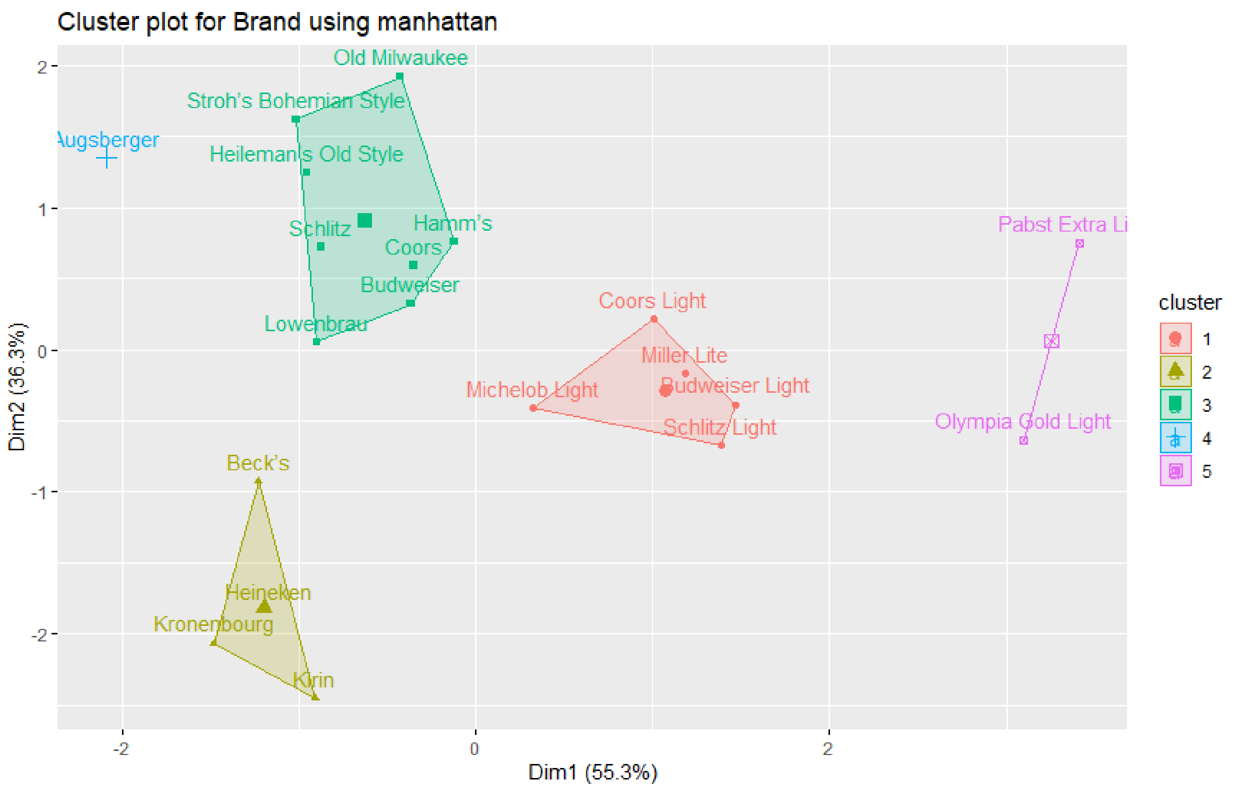
With *n=5:*



Cluster plot for Brand using euclidean
Old Milwaukee
Stroh's Bohemian Style
Augsberger
Heileman's Old Style
Schlitz
Hamm's
Coors
Budweiser
Lowenbrau
Coors Light
Miller Lite
Budweiser Light
Michelob Light
Schlitz Light
Pabst Extra Li
Olympia Gold Light
Beck's
Heineken
Kronenbourg
Kirin
Dim1 (55.3%)
Dim2 (36.3%)
cluster
1
2
3
4
5



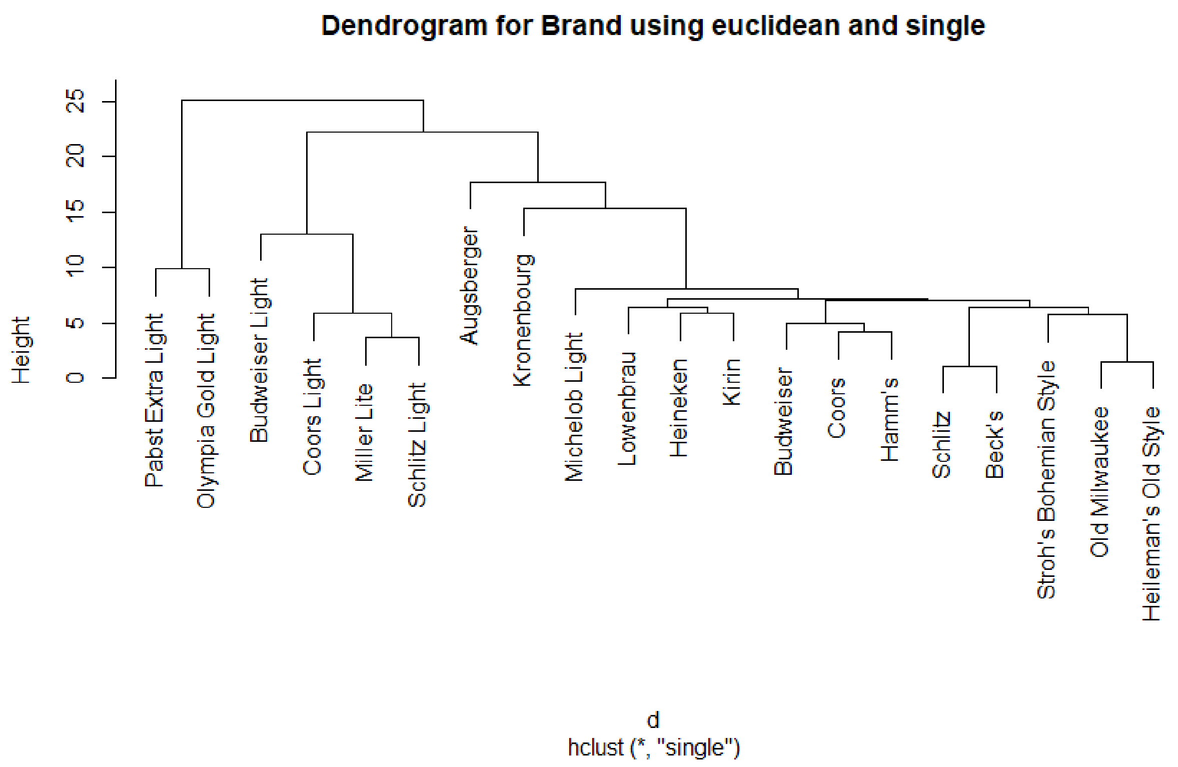
Cluster plot for Brand using maximum
Old Milwaukee
Stroh's Bohemian Style
Augsberger
Heileman's Old Style
Schlitz
Hamm's
Coors
Budweiser
Lowenbrau
Coors Light
Miller Lite
Budweiser Light
Michelob Light
Schlitz Light
Pabst Extra Li
Olympia Gold Light
Beck's
Heineken
Kronenbourg
Kirin
Dim1 (55.3%)
Dim2 (36.3%)
cluster
1
2
3
4
5



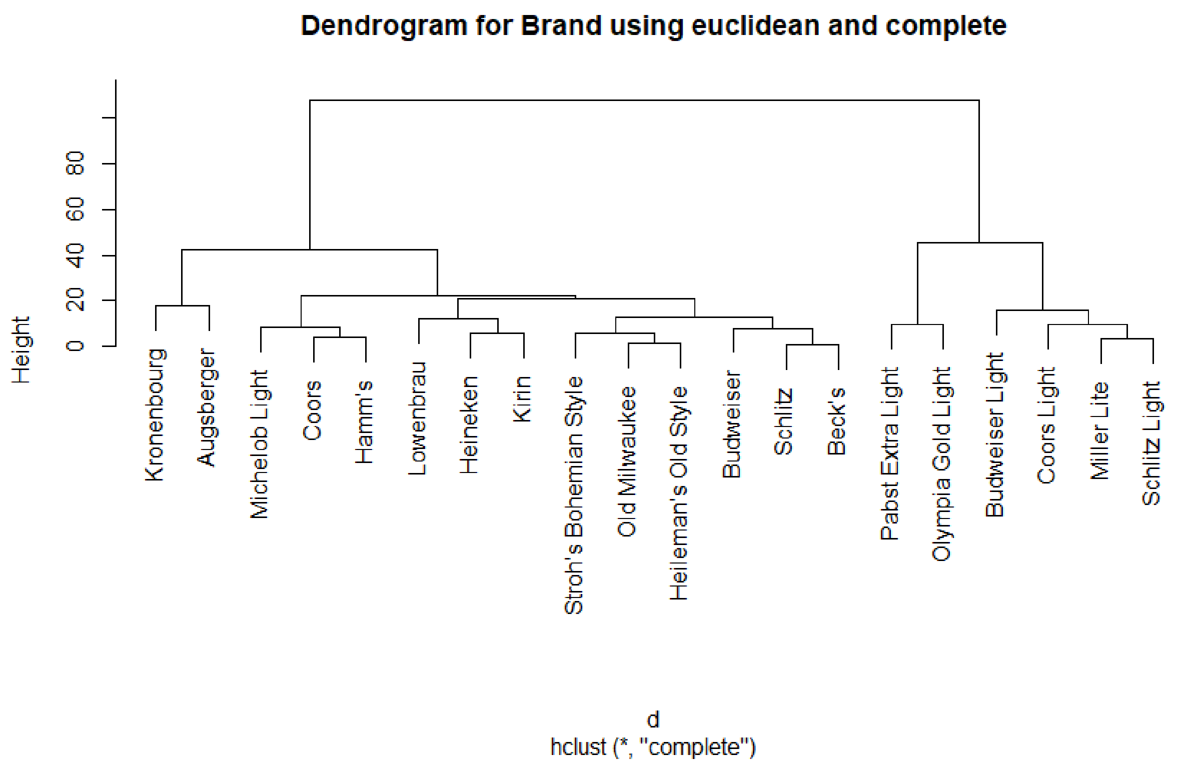
The euclidean with *n=5* looks the most right (no singletons, not too big clusters), although Olympia Gold Light and Pabst Extra Light might need to be singletons since they are just too unique.

If we remember our analysis idea, the cluster to infiltrate is the “Olympia Gold Light and Pabst Extra Light” one. By creating the product with similar parameters but superior taste and less price we can dominate this part of beer market.

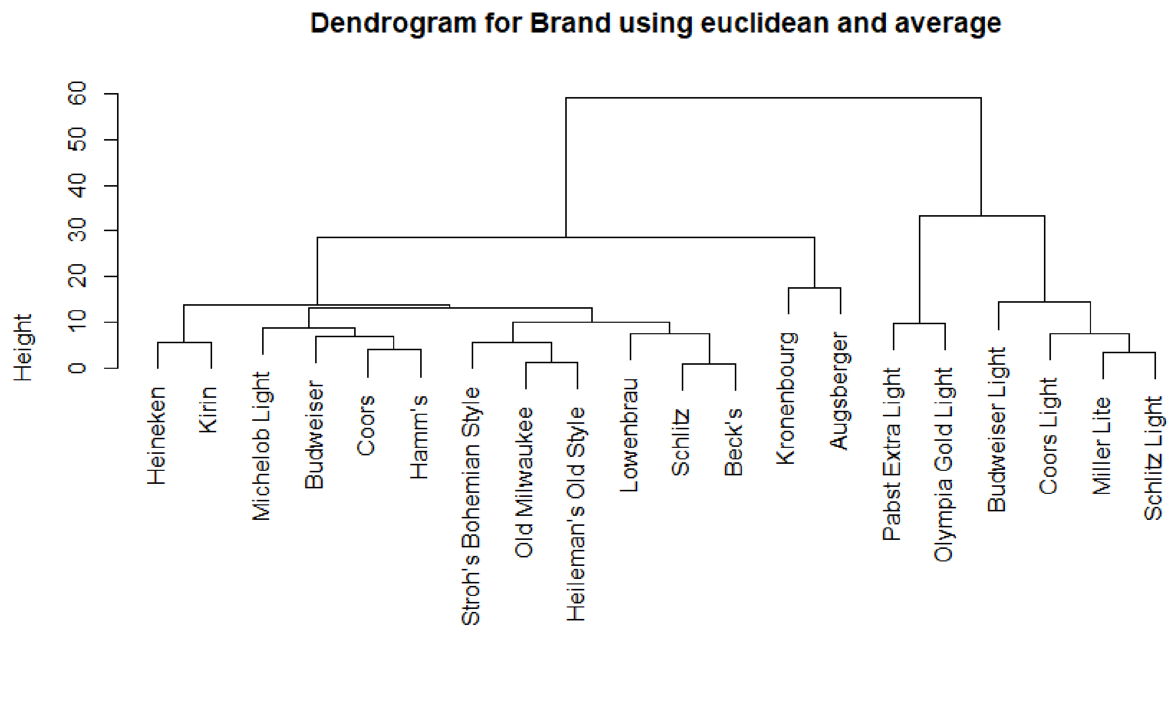
In hierarchical clusters **euclidean/maximum/manhattan** distance methods and **single/complete/average/centroid/ward.D2** clusterization methods are used.
n is the number of clusters on the levels 1-2 (with 0 being the highest one).



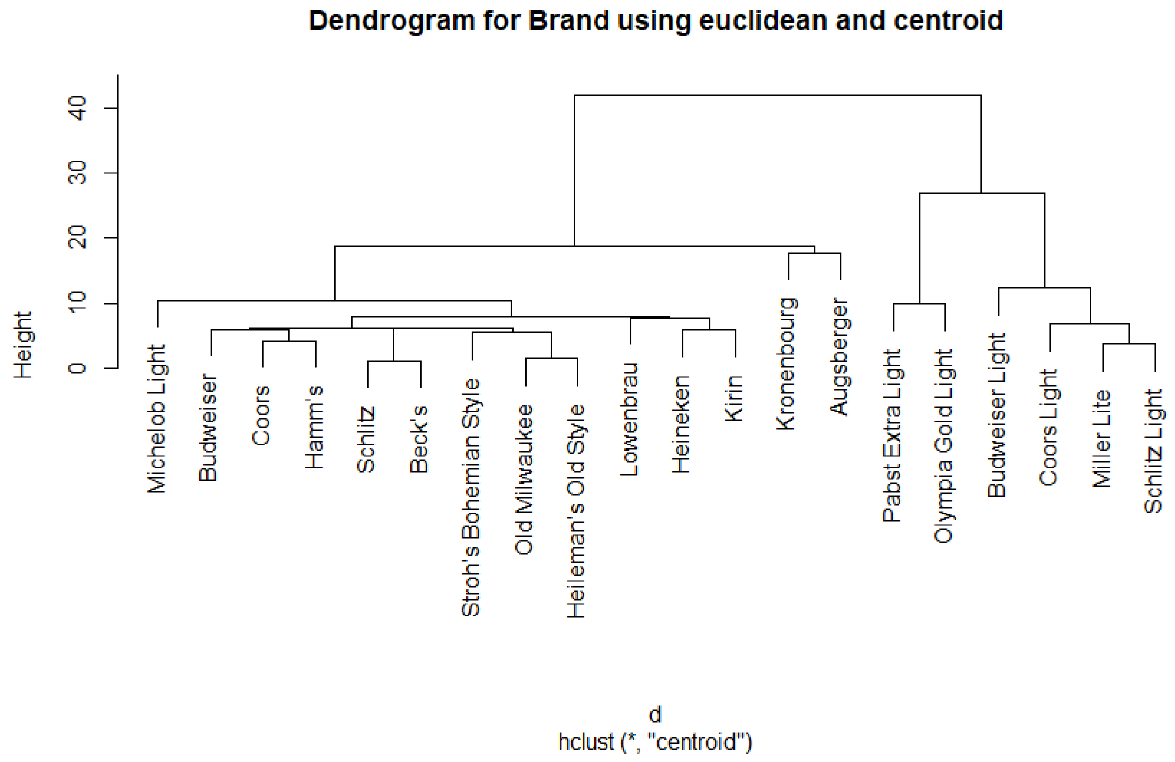
n=3



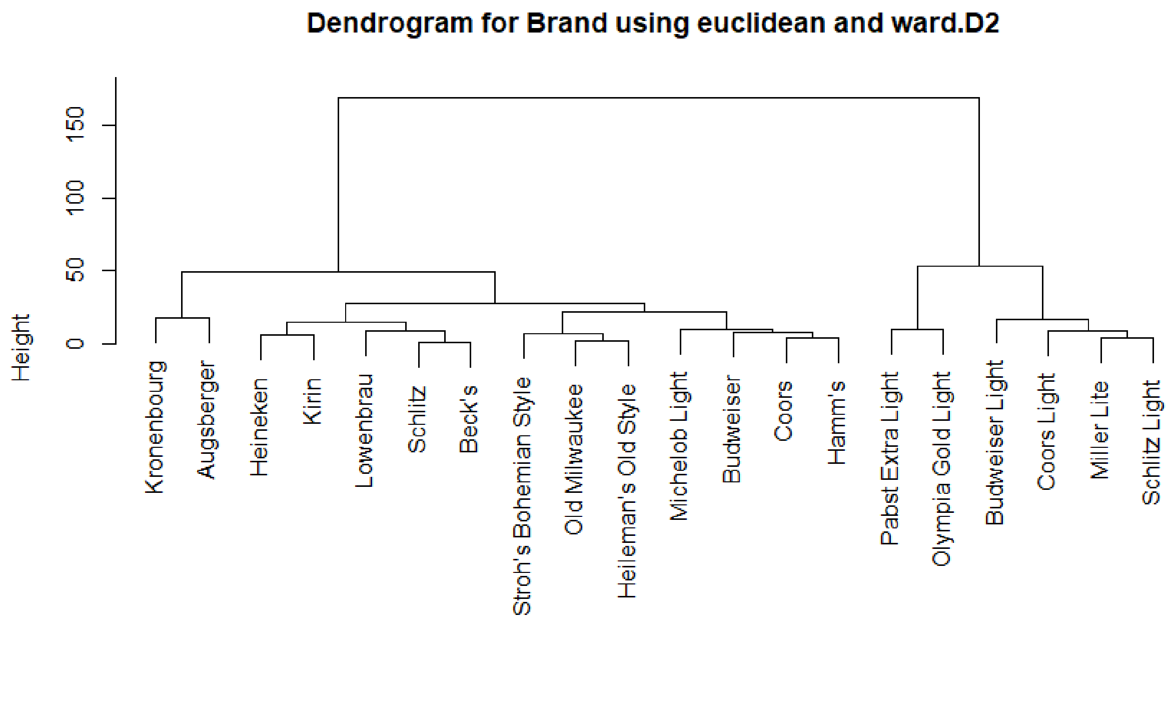
n=4


Dendrogram for Brand using euclidean and average
Height
0
10
20
30
40
50
60
Heineken
Kirin
Michelob Light
Budweiser
Coors
Hamm's
Stroh's Bohemian Style
Old Milwaukee
Heileman's Old Style
Lowenbrau
Schlitz
Beck's
Kronenbourg
Augsberger
Pabst Extra Light
Olympia Gold Light
Budweiser Light
Coors Light
Miller Lite
Schlitz Light
d
hclust (*, "average")

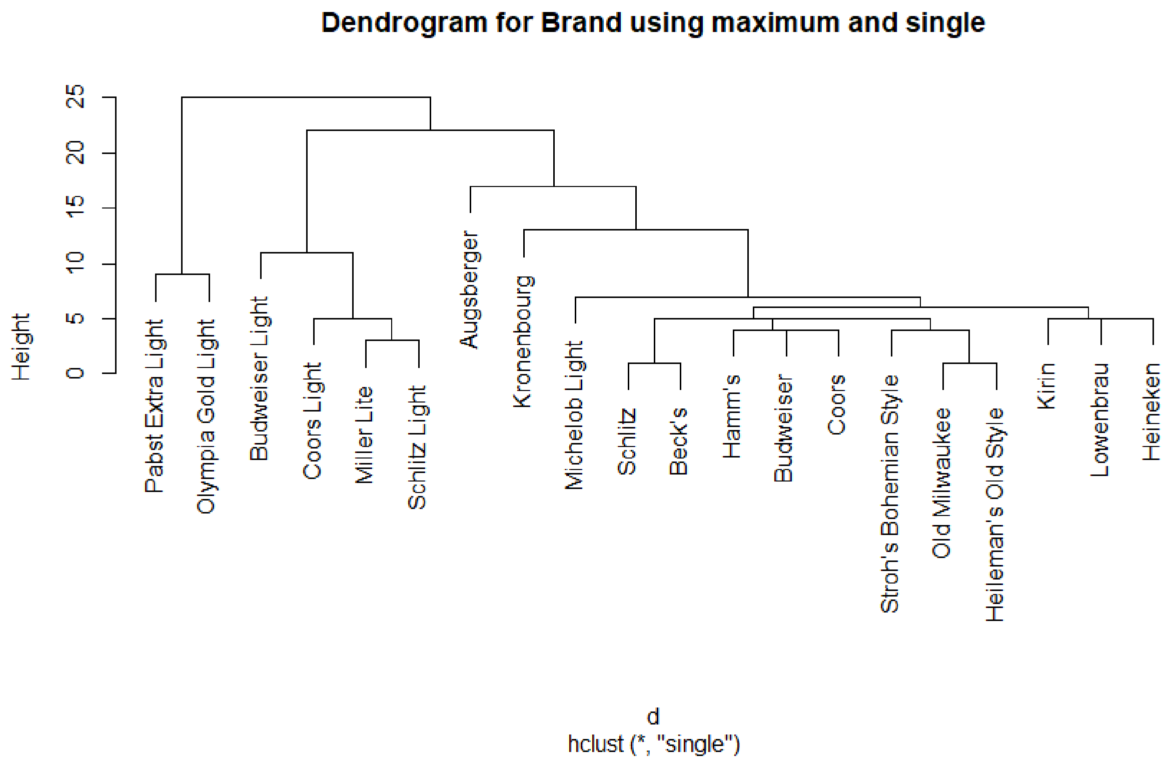
n=4


Dendrogram for Brand using euclidean and centroid
Height
0
10
20
30
40
Michelob Light
Budweiser
Coors
Hamm's
Schlitz
Beck's
Stroh's Bohemian Style
Old Milwaukee
Heileman's Old Style
Lowenbrau
Heineken
Kirin
Kronenbourg
Augsberger
Pabst Extra Light
Olympia Gold Light
Budweiser Light
Coors Light
Miller Lite
Schlitz Light
d
hclust (*, "centroid")

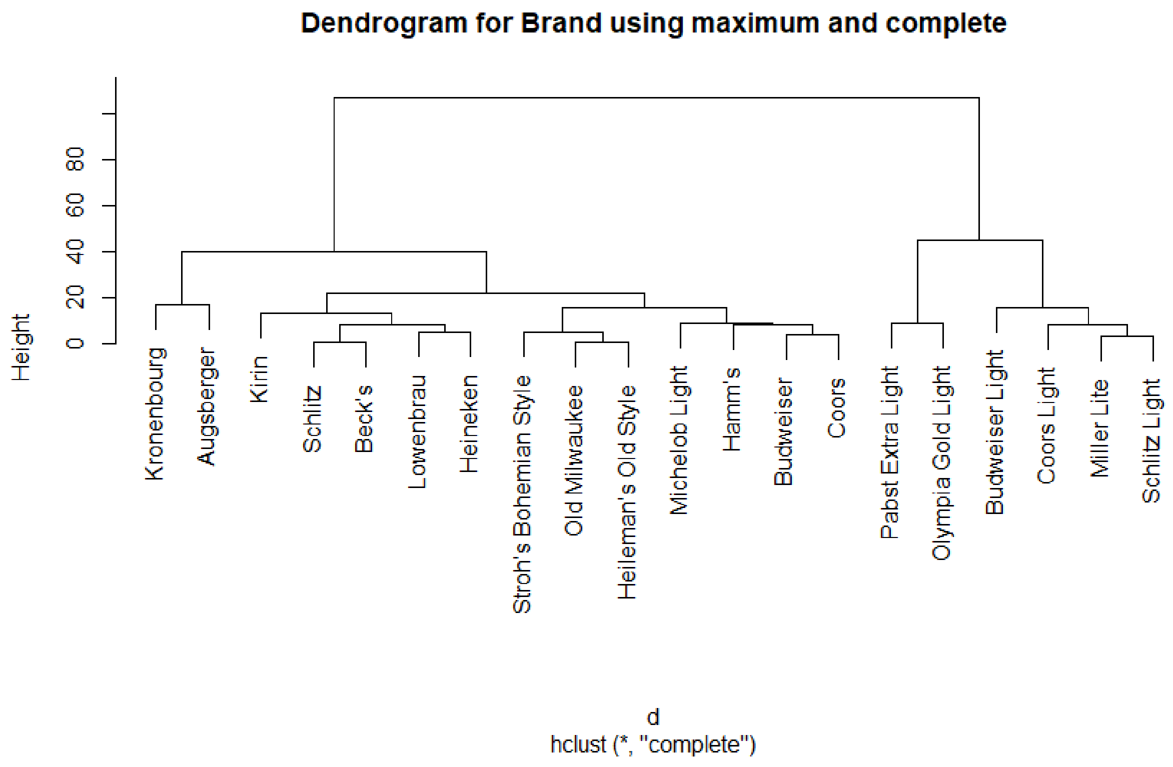
n=4


Dendrogram for Brand using euclidean and ward.D2
Height
0
50
100
150
Kronenbourg
Augsberger
Heineken
Kirin
Lowenbrau
Schlitz
Beck's
Stroh's Bohemian Style
Old Milwaukee
Heileman's Old Style
Michelob Light
Budweiser
Coors
Hamm's
Pabst Extra Light
Olympia Gold Light
Budweiser Light
Coors Light
Miller Lite
Schlitz Light
d
hclust (*, "ward.D2")

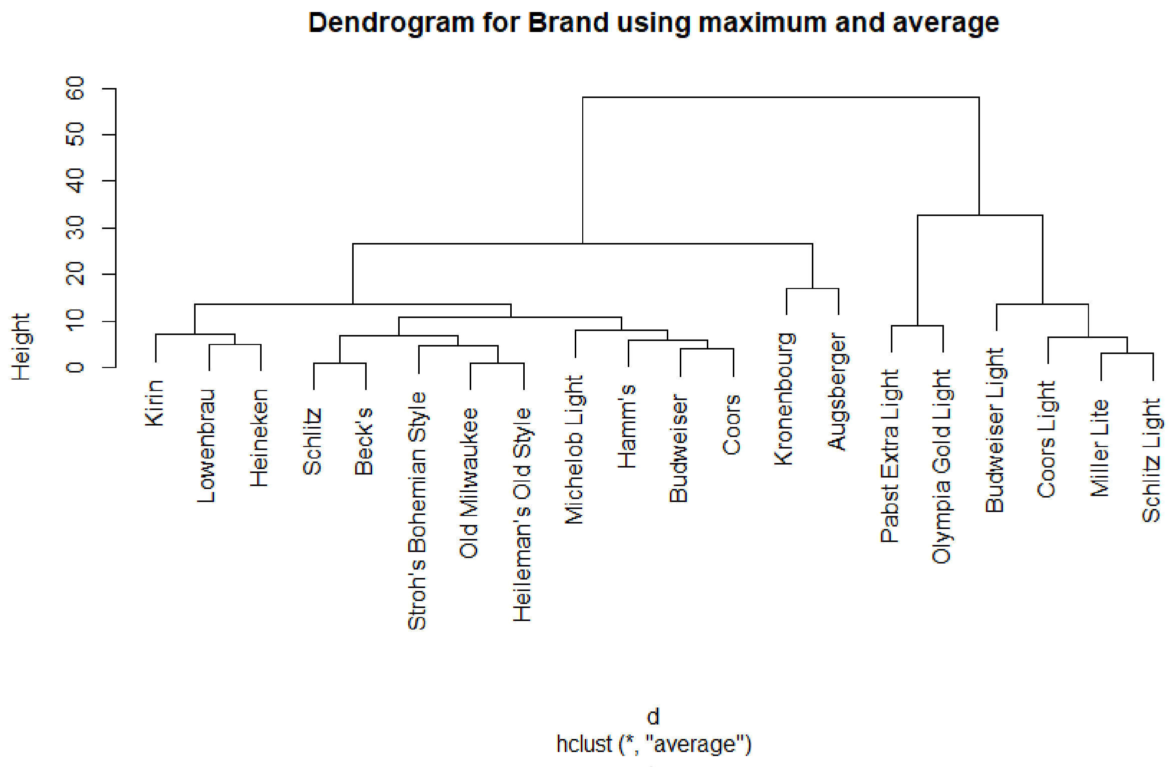
n=4


Dendrogram for Brand using maximum and single
Height
0
5
10
15
20
25
Pabst Extra Light
Olympia Gold Light
Budweiser Light
Coors Light
Miller Lite
Schlitz Light
Augsberger
Kronenbourg
Michelob Light
Schlitz
Beck's
Hamm's
Budweiser
Coors
Stroh's Bohemian Style
Old Milwaukee
Heileman's Old Style
Kirin
Lowenbrau
Heineken
d
hclust (*, "single")

n=3

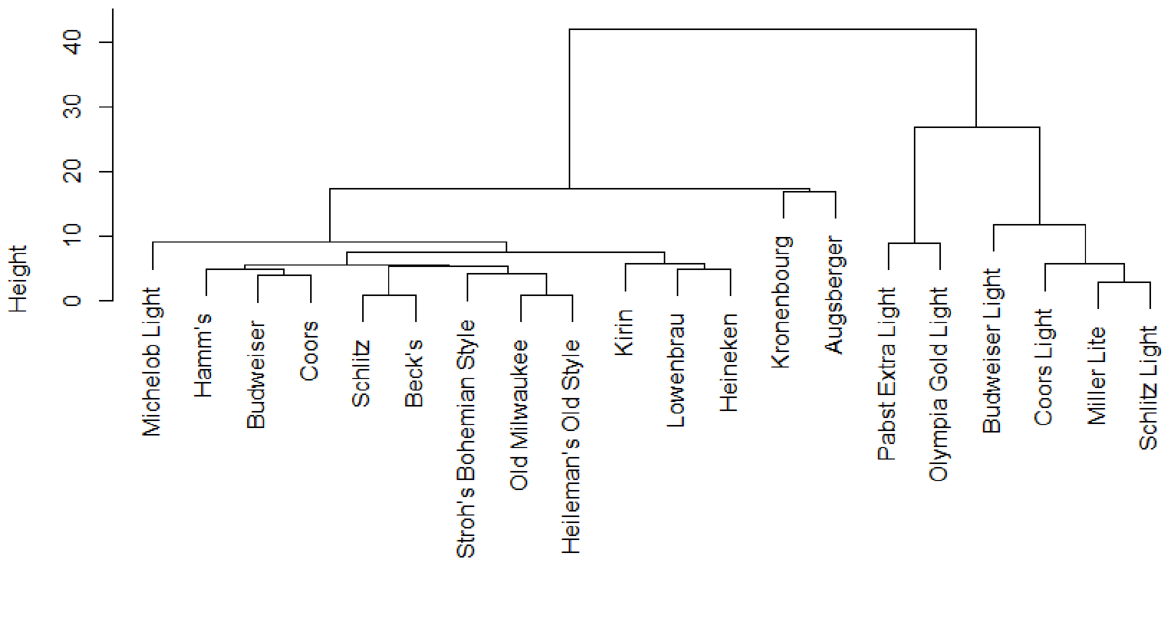

Dendrogram for Brand using maximum and complete
Height
0
20
40
60
80
Kronenbourg
Augsberger
Kirin
Schlitz
Beck's
Lowenbrau
Heineken
Stroh's Bohemian Style
Old Milwaukee
Heileman's Old Style
Michelob Light
Hamm's
Budweiser
Coors
Pabst Extra Light
Olympia Gold Light
Budweiser Light
Coors Light
Miller Lite
Schlitz Light
d
hclust (*, "complete")

n=4


Dendrogram for Brand using maximum and average
Height
0
10
20
30
40
50
60
Kirin
Lowenbrau
Heineken
Schlitz
Beck's
Stroh's Bohemian Style
Old Milwaukee
Heileman's Old Style
Michelob Light
Hamm's
Budweiser
Coors
Kronenbourg
Augsberger
Pabst Extra Light
Olympia Gold Light
Budweiser Light
Coors Light
Miller Lite
Schlitz Light
d
hclust (*, "average")

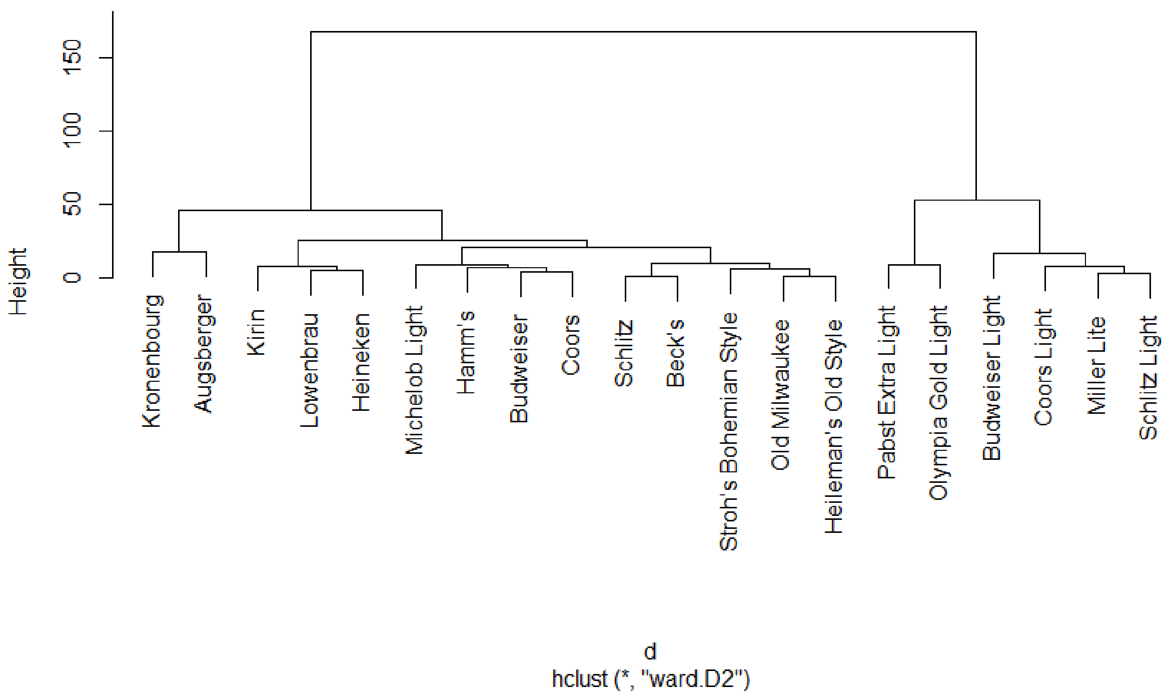
n=4

Dendrogram for Brand using maximum and centroid

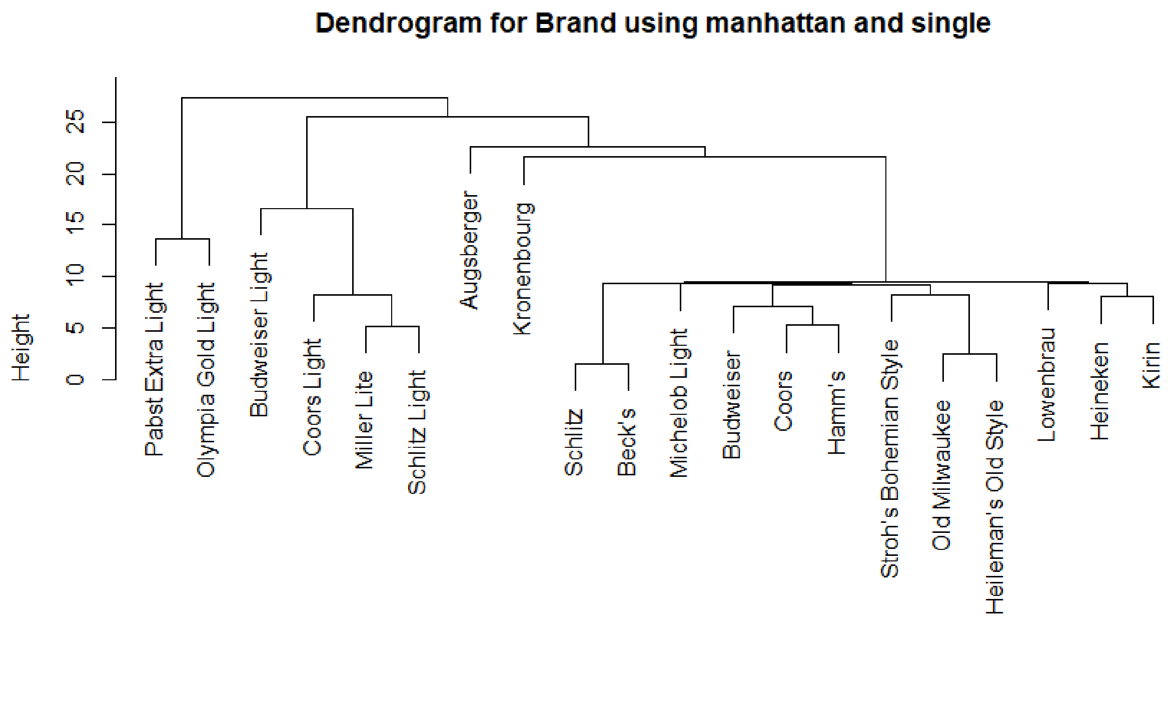


n=4

Dendrogram for Brand using maximum and ward.D2

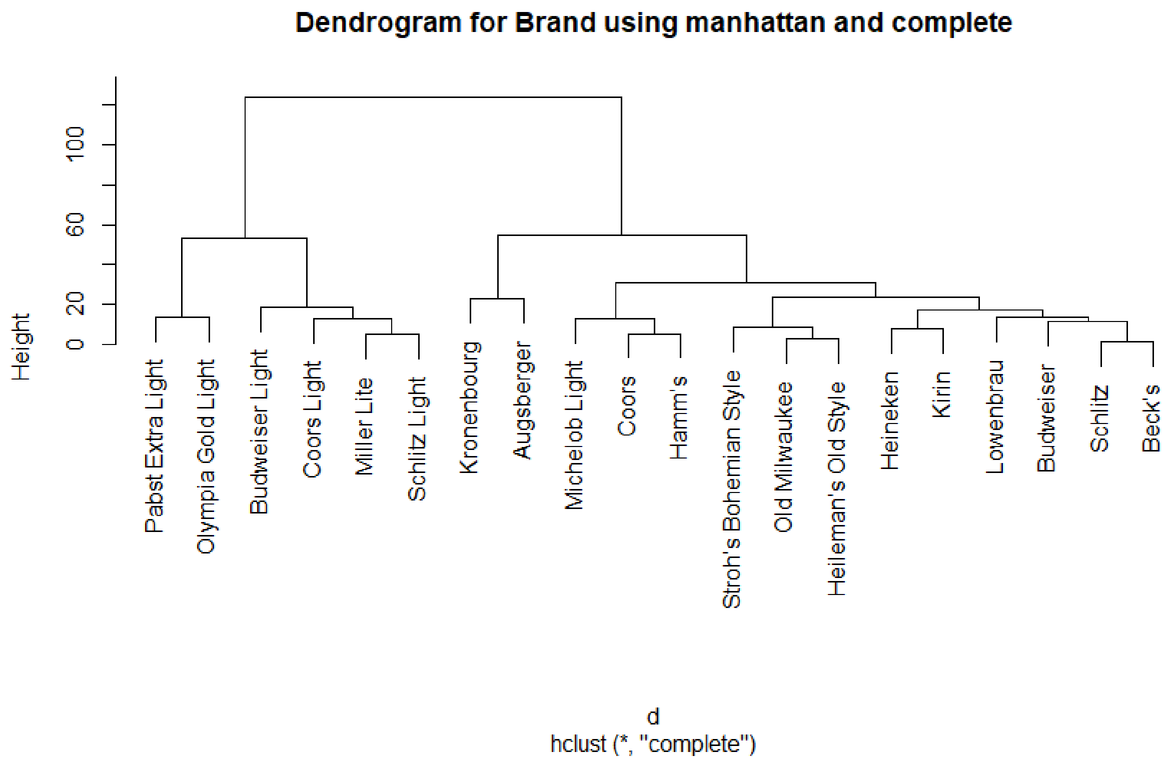


n=4



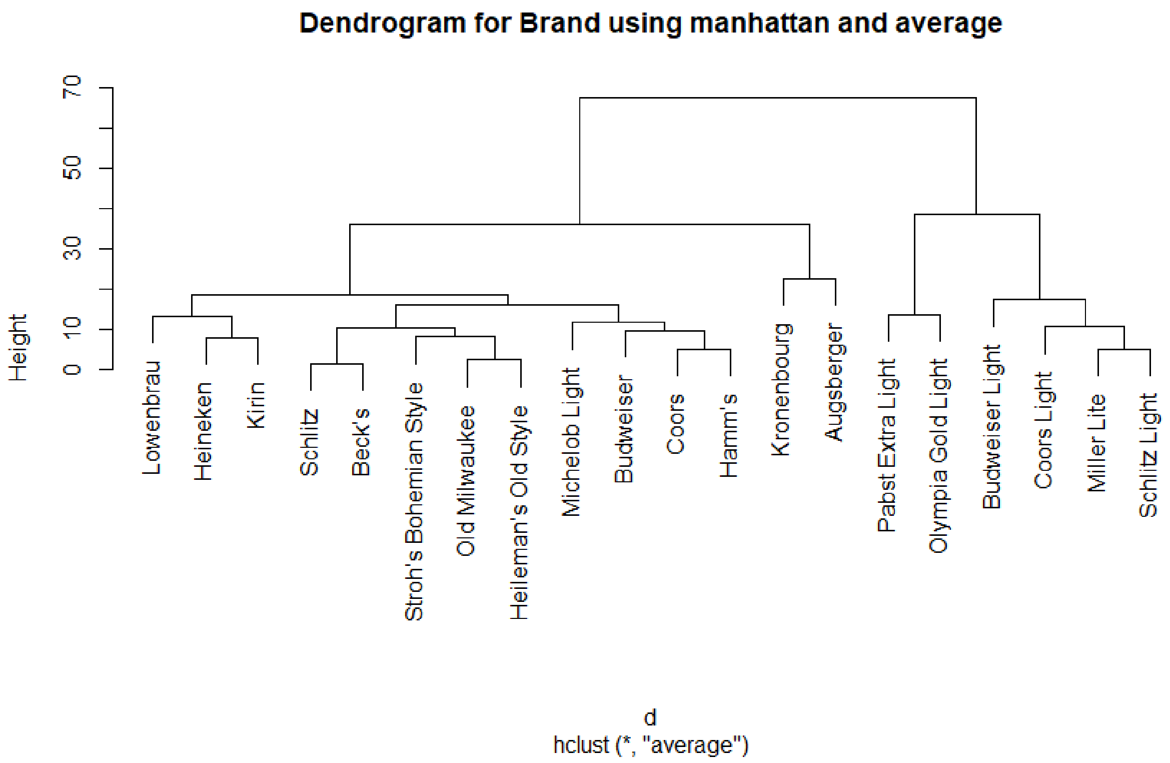
Dendrogram for Brand using manhattan and single
Height
0
5
10
15
20
25
Pabst Extra Light
Olympia Gold Light
Budweiser Light
Coors Light
Miller Lite
Schlitz Light
Augsberger
Kronenbourg
Schlitz
Beck's
Michelob Light
Budweiser
Coors
Hamm's
Stroh's Bohemian Style
Old Milwaukee
Heileman's Old Style
Lowenbrau
Heineken
Kirin
d
hclust (*, "single")

n=3

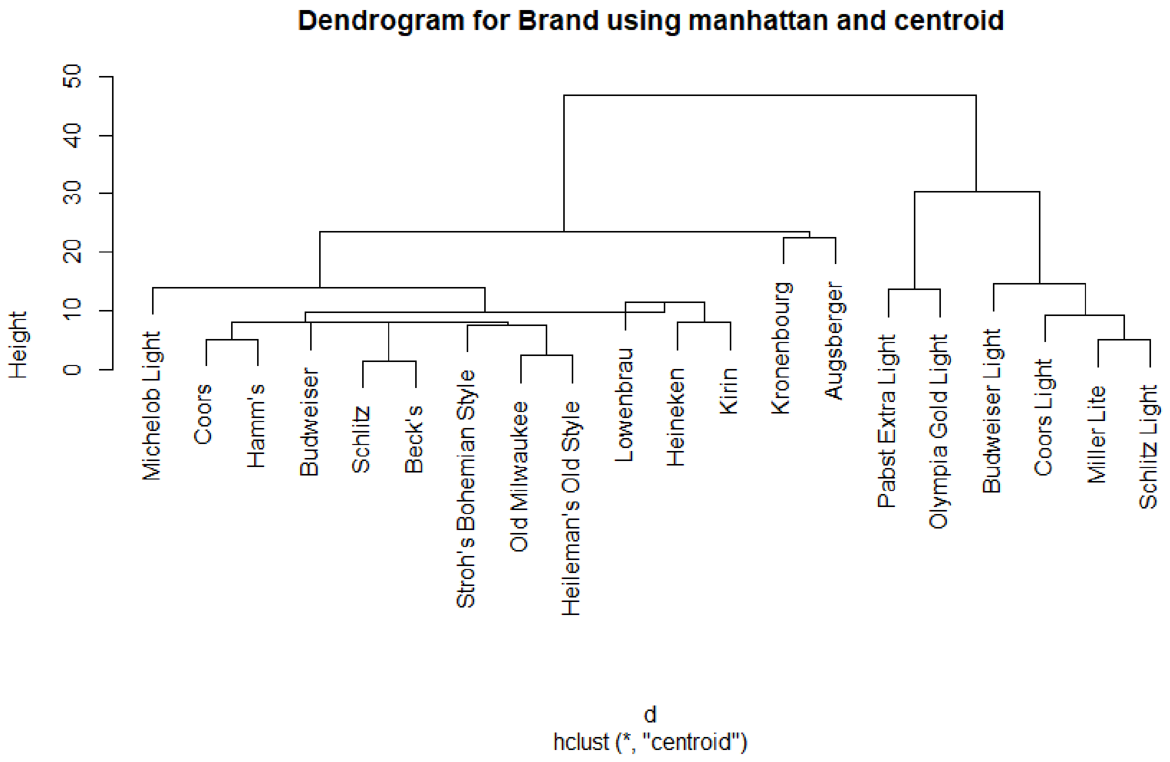


Dendrogram for Brand using manhattan and complete
Height
0
20
60
100
Pabst Extra Light
Olympia Gold Light
Budweiser Light
Coors Light
Miller Lite
Schlitz Light
Kronenbourg
Augsberger
Michelob Light
Coors
Hamm's
Stroh's Bohemian Style
Old Milwaukee
Heileman's Old Style
Heineken
Kirin
Lowenbrau
Budweiser
Schlitz
Beck's
d
hclust (*, "complete")

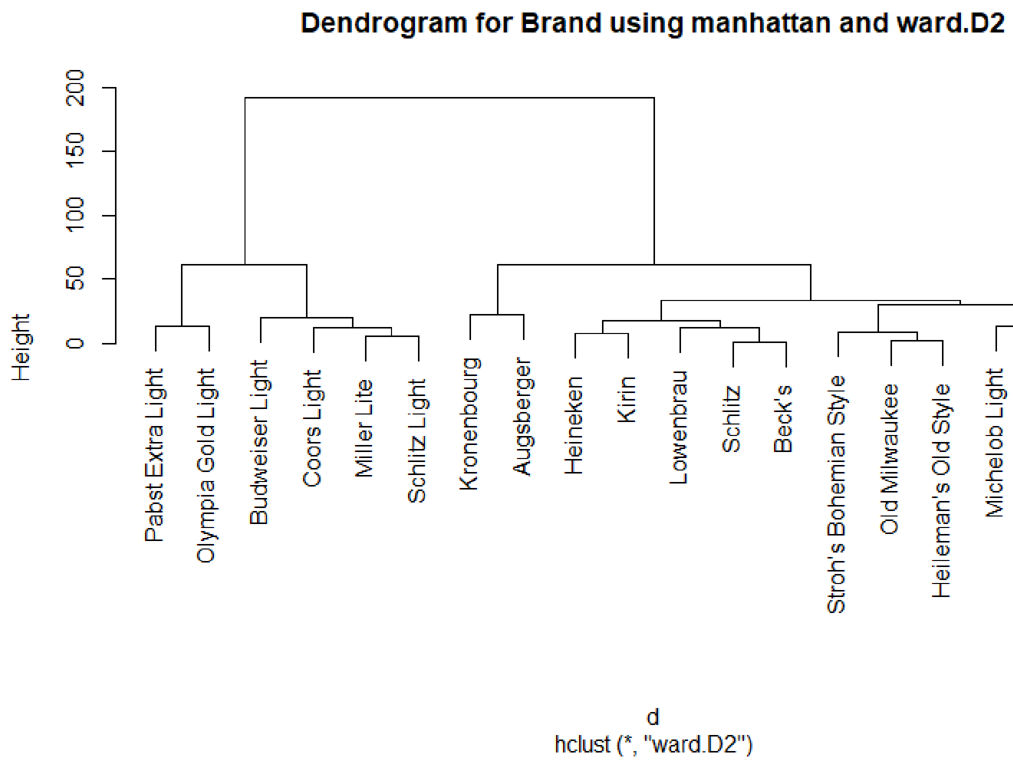
n=4


Dendrogram for Brand using manhattan and average
Height
0
10
30
50
70
Lowenbrau
Heineken
Kirin
Schlitz
Beck's
Stroh's Bohemian Style
Old Milwaukee
Heileman's Old Style
Michelob Light
Budweiser
Coors
Hamm's
Kronenbourg
Augsberger
Pabst Extra Light
Olympia Gold Light
Budweiser Light
Coors Light
Miller Lite
Schlitz Light
d
hclust (*, "average")

n=4


Dendrogram for Brand using manhattan and centroid
Height
0
10
20
30
40
50
Michelob Light
Coors
Hamm's
Budweiser
Schlitz
Beck's
Stroh's Bohemian Style
Old Milwaukee
Heileman's Old Style
Lowenbrau
Heineken
Kirin
Kronenbourg
Augsberger
Pabst Extra Light
Olympia Gold Light
Budweiser Light
Coors Light
Miller Lite
Schlitz Light
d
hclust (*, "centroid")

n=4



n=4

Dendrograms are practically the same (some differences in positions of lower hierarchies and how they stack at the top) for all methods with *n* being 3 or 4.
Hierarchical clustering gives results similar to non-hierarchical (it is possible to find similarities between pairs of clusters).

Appendix

Listing 1.

```
library(readxl)
library(EnvStats)
library(DescriptiveStats.OBeu)

getmode = function(v) {
  uniqv = unique(v)
  return(uniqv[which.max(tabulate(match(v, uniqv)))])
}


t1 = read_excel("Student 1/Task1.xls.xlsx", col_names = FALSE, col_types = c("numeric"))[[1]]
name = "Girth, mm"

#print(summary(t1))
print(ds.statistics(t1))
print(paste("Mode: ", getmode(t1)))

ecdfPlot(t1, main = "CDF", xlab = name)

for (i in list(5, 10, 15, 60)) {
  hist(t1, main = paste("Histogram for approx.", i, "intervals"), xlab = name, breaks = i)
}

boxplot(t1, ylab = name)
```

Listing 2.

```
library(readxl)
library(MVN)
library(car)
library(nortest)

t2.dat <- read_excel("Student 1/Task2.xls.xlsx", col_names = FALSE)
t2 = t2.dat[[1]]

conf.lvl = 0.9

qqPlot(t2, distribution = "norm", main = "Q-Q plot", envelope = conf.lvl)

for (adj in list(TRUE, FALSE)){
  print(pearson.test(t2, adjust = adj))
}

for (t in list("Lillie", "CVM", "AD")){
  print(mvn(t2.dat, univariateTest = t, desc = TRUE))
}
```

Listing 3.

```
library(readxl)
library(car)

t3.dat <- read_excel("Student 1/Task3.xlsx", col_names = FALSE)

names(t3.dat) = c(1,2,3,4)

conf.lvl = 0.95

samples = list()
for (i in (1:2)){
  samples[[i]] = na.omit(t3.dat[[i]])
}

for (i in (1:2)){
  qqPlot(samples[[i]], distribution = "norm", envelope = conf.lvl, main = paste("Sample", i))
  print(shapiro.test(samples[[i]]))
}

t.test(x = samples[[1]], y = samples[[2]],  mu = 0, alternative = "two.sided", var.equal = TRUE,
conf.level = conf.lvl)

var.test(x = samples[[1]], y = samples[[2]], ratio = 1, alternative = "two.sided", conf.level =
conf.lvl)

ks.test(x = samples[[1]], y = samples[[2]], alternative = "two.sided")

wilcox.test(x = samples[[1]], y = samples[[2]], mu = 0, alternative = "two.sided", conf.level =
conf.lvl)
```

Listing 4.

```
library(readxl)
library(car)
library(lmtest)
library(MASS)
library(aod)
library(InformationValue)
library(quantreg)
library(ridge)

t41 <- read_excel("Student 1/Task4-1.xlsx")

research_reg = function(data, formula, reg_name){
  print(reg_name)
  res = 0
  if (reg_name == 'linear'){
    res = lm(formula = formula, data = data)

    print(bptest(formula = res))
    if (as.character(formula[[3]])[[1]] == "+"){
      print(vif(mod = res))
    }
    print(dwtest(formula = res))
  }
  else if (reg_name == 'ridge') {
    # res = lm.ridge(formula = formula, lambda = seq(0.1, 20, by=0.1), data = data)
    res = linearRidge(formula = formula, lambda = c(0.1, 0.2, 4.9, 5.0, 5.1, 9.9, 10.0, 10.1, 
14.9, 15.0, 15.1, 19.9, 20.0), data = data)
    print(res)
  }
  else if (reg_name == 'quantile') {
    res = rq(formula = formula, tau = 1:9/10, data = data)
  }

  print(summary(res, se="ker"))
  plot(res)
}

regs = list('quantile', 'ridge')

fs = list(Y ~ X1 + X2, Y ~ X1, Y ~ X2)

for (f in fs){
  print(f)
  research_reg(data = t41, formula = f, reg_name = "linear")
}

for (r in regs){
  research_reg(data = t41, formula = fs[[1]], reg_name = r)
}

t42 <- read_excel("Student 1/Task4-2.xlsx")

research_bin = function(formula, data, actual_var, family_name){
  print(paste(family_name, "for", actual_var))
  mybin = glm(formula, data, family = binomial(family_name))
  print(summary(mybin))

  print(wald.test(b = coef(mybin), Sigma = vcov(mybin), Terms = 1:ncol(data)))

  predicted = plogis(predict(mybin, data))
  actual = data[[actual_var]]
  optCutOff = optimalCutoff(actual, predicted)[1]
  print(optCutOff)

  print(confusionMatrix(actual, predicted))
  print(confusionMatrix(actual, predicted, threshold = optCutOff))

  plotROC(actual, predicted)
}

f = Y ~ X

family_names = list('logit', 'probit')

for (f_n in family_names){
  research_bin(f, t42, "Y", f_n)
}
```

Listing 5.

```
library(readxl)
library(ggplot2)
library(factoextra)
library(amap)
library(fpc)

t5.dat = na.omit(read_excel("Student 1/Task_5.xls"))

names(t5.dat) = c("Brand", "Calories", "Sodium", "Alcohol", "Price")

t5 = data.matrix(t5.dat[c(-1)])
rownames(t5) = t5.dat[[1]]

research_clust = function(data, data_name, dists){
  data = scale(na.omit(data))
  a = fviz_nbclust(x = data, FUNcluster = kmeans, method = "silhouette")
  plot(a)
  a_data = a$data
  # opt_n_clust = as.numeric(a_data$clusters[which.max(a_data$y)])
  opt_n_clust = 5
  for (dist_method in dists){
    km.res = Kmeans(x = data, centers = opt_n_clust, method = dist_method)
    plot(fviz_cluster(km.res, data = data, frame.type = "convex",
                      main = paste("Cluster plot for", data_name, "using", dist_method)))
  }
}

research_hclust = function(data, data_name, dists, clusts){
    for (dist_method in dists){
      for (clust_method in clusts){
        d = dist(data, method = dist_method)
        res.hc = hclust(d, method = clust_method)
        plot(res.hc,  main  =  paste("Dendrogram  for",  data_name,  "using",  dist_method,  "and",
clust_method))
      }
    }
}

dist_methods = list("euclidean", "maximum", "manhattan")

research_clust(t5, "Brand", dist_methods)

clust_methods = list("single", "complete", "average", "centroid", "ward.D2")

research_hclust(t5, "Brand", dist_methods, clust_methods)
```