R

Yi-Ju Tseng

2017-04-16

Contents

Chapter 1

R 101

Chapter 2

R

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

Chapter 10

10.1

Data mining

- /
- /
- /

- Supervised learning
  - Regression ' '
    - Linear Regression
    - Logistic Regression
  - Classification P/N, Yes/No, M/F, Sick/Not sick / (A/B/C/D)
    - Support Vector Machines
    - Decision Trees
    - Neural Networks
    - K-Nearest Neighbor
- Unsupervised learning
  - Clustering
    - Hierarchical clustering
    - K-means clustering
  - Association Rules

10.2

10.2.1 Regression

Regression Analysis

- Linear Regression
- Logistic Regression

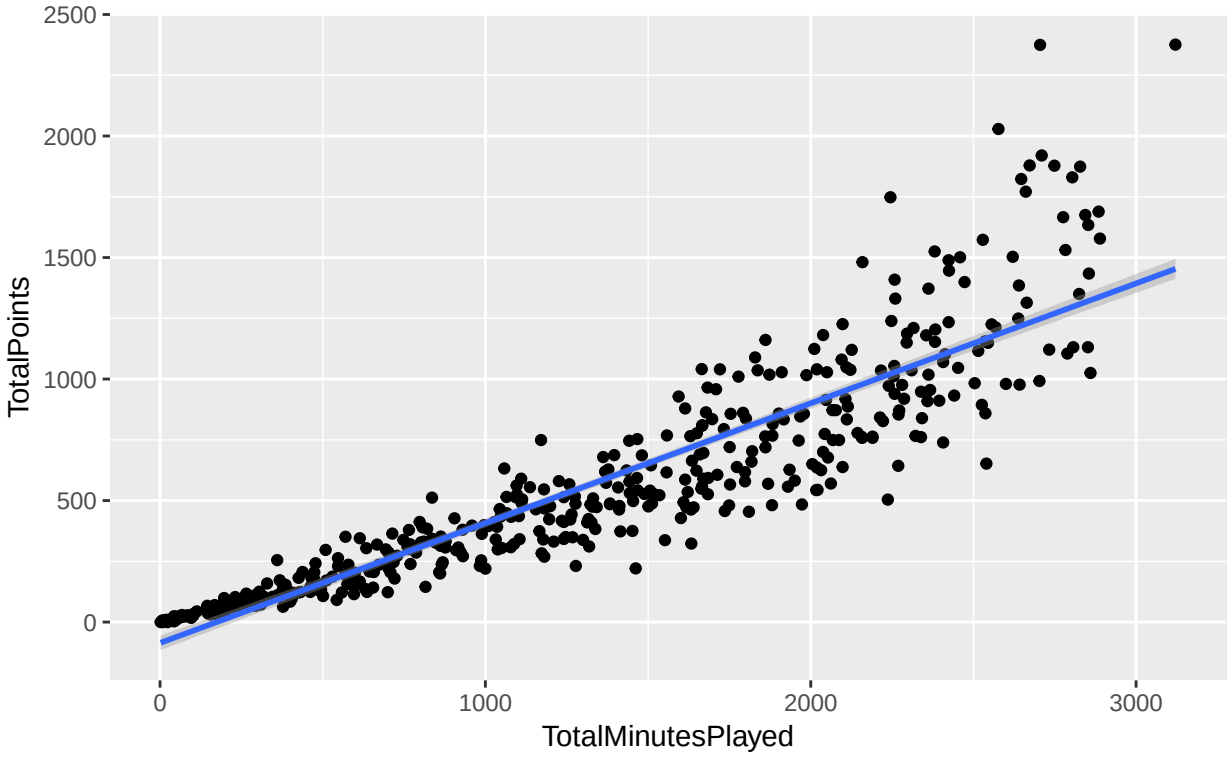
10.2.1.1 Linear Regression

NBA

```
# SportsAnalytics package
library(SportsAnalytics)
# 2015-2016
NBA1516<-fetch_NBAPlayerStatistics("15-16")
```

NBA

```
library(ggplot2)
ggplot(NBA1516,aes(x=TotalMinutesPlayed,y=TotalPoints))+
    geom_point()+geom_smooth(method = "glm")
```



- glm()

```
# formula: Y~X1+X2+...+Xn  Y:   X:
# data:
glm(TotalPoints~TotalMinutesPlayed,data =NBA1516)
```

```
##
## Call:  glm(formula = TotalPoints ~ TotalMinutesPlayed, data = NBA1516)
```

```
##
## Coefficients:
##        (Intercept)  TotalMinutesPlayed
##           -85.9071              0.4931
##
## Degrees of Freedom: 475 Total (i.e. Null);  474 Residual
## Null Deviance:       99360000
## Residual Deviance: 16720000  AIC: 6339
```

TotalPoints = 0.4931 * TotalMinutesPlayed -85.9071

glm() - generalized linear models (glm) - - lm() - - family="gaussian" - family="binomial" - family="poisson"

Gaussian distribution

Binomial distribution n /

Poisson distribution

-
-
-
-
- DNA .....

-

```
# e+01: 10^1 / e-04: 10^(-4)
glm(TotalPoints~TotalMinutesPlayed+FieldGoalsAttempted,
    data =NBA1516)
```

```
##
## Call:  glm(formula = TotalPoints ~ TotalMinutesPlayed + FieldGoalsAttempted,
##     data = NBA1516)
##
## Coefficients:
##         (Intercept)   TotalMinutesPlayed  FieldGoalsAttempted
##          -1.799e+01           -2.347e-04            1.256e+00
##
## Degrees of Freedom: 475 Total (i.e. Null);  473 Residual
## Null Deviance:       99360000
## Residual Deviance: 2160000   AIC: 5367
```

TotalPoints = -0.0002347 * TotalMinutesPlayed + 1.255794 *FieldGoalsAttempted -17.99

-

```
glm(TotalPoints~TotalMinutesPlayed+FieldGoalsAttempted+Position,
    data =NBA1516)
```

```
##
## Call:  glm(formula = TotalPoints ~ TotalMinutesPlayed + FieldGoalsAttempted +
##     Position, data = NBA1516)
##
## Coefficients:
##         (Intercept)   TotalMinutesPlayed  FieldGoalsAttempted
##           22.852223            -0.006537             1.275721
##          PositionPF           PositionPG           PositionSF
##          -39.416327           -65.034646           -38.522299
##          PositionSG
```

```
##          -52.175144
##
## Degrees of Freedom: 474 Total (i.e. Null);  468 Residual
##   (1 observation deleted due to missingness)
## Null Deviance:       99080000
## Residual Deviance: 1975000   AIC: 5322
```

```
# e+01: 10^1 / e-04: 10^(-4)
```

TotalPoints = -0.0065 * TotalMinutesPlayed + 1.28 *FieldGoalsAttempted* +22.85 + 22.85 PositionPF + -65.03 * PositionPG + -38.52 * PositionSF + -52.18 * PositionSG

Dummy Variable

- PositionPF? PositionPG? PositionSF? PositionSG?
- PG …
  - PositionPF=0
  - PositionPG=1
  - PositionSF=0
  - PositionSG=0
- —
  - PositionPF=0
  - PositionPG=0
  - PositionSF=0
  - PositionSG=0

-
- X
- R factor R
-

```
class(NBA1516$Position)
```

```
## [1] "factor"
```

```
levels(NBA1516$Position)
```

```
## [1] "C"  "PF" "PG" "SF" "SG"
```

-
-
  - Akaike's Information Criterion (AIC)
  - Bayesian Information Criterion (BIC)
-

```
OneVar<-glm(TotalPoints~TotalMinutesPlayed,data =NBA1516)
TwoVar<-glm(TotalPoints~TotalMinutesPlayed+FieldGoalsAttempted,
            data =NBA1516)
ThreeVar<-glm(TotalPoints~TotalMinutesPlayed+FieldGoalsAttempted+Position,
              data =NBA1516)
c(OneVar$aic,TwoVar$aic,ThreeVar$aic)
```

```
## [1] 6338.913 5366.763 5321.972
```

```
sum2<-summary(TwoVar)
sum2$coefficients
```

```
##                           Estimate  Std. Error     t value      Pr(>|t|)
## (Intercept)         -1.798855e+01 5.659758251 -3.17832538  1.578333e-03
## TotalMinutesPlayed  -2.347183e-04 0.009474631 -0.02477334  9.802462e-01
## FieldGoalsAttempted  1.255794e+00 0.022239494 56.46682752 2.474028e-212
```

```
sum3<-summary(ThreeVar)
sum3$coefficients
```

```
##                           Estimate   Std. Error    t value      Pr(>|t|)
## (Intercept)          22.852222668  9.014714391  2.5349913  1.156964e-02
## TotalMinutesPlayed   -0.006536874  0.009199968 -0.7105322  4.777281e-01
## FieldGoalsAttempted   1.275721212  0.021647176 58.9324535 1.144607e-218
## PositionPF          -39.416326742  9.936541704 -3.9668053  8.425605e-05
## PositionPG          -65.034646215 10.269250388 -6.3329497  5.648565e-10
## PositionSF          -38.522298887 10.488170409 -3.6729284  2.674727e-04
## PositionSG          -52.175143670  9.985331185 -5.2251791  2.625062e-07
```

10.2.1.2 Logistic Regression

Logistic Regression

- 0 1
- /
- /
- family="binomial"

/

```
mydata <- read.csv("http://www.ats.ucla.edu/stat/data/binary.csv")
# GRE:   , GPA:    , rank:
head(mydata)
```

```
##   admit gre  gpa rank
## 1     0 380 3.61    3
## 2     1 660 3.67    3
## 3     1 800 4.00    1
## 4     1 640 3.19    4
## 5     0 520 2.93    4
## 6     1 760 3.00    2
```

Hmm….

```
mydata$rank <- factor(mydata$rank)
mylogit <- glm(admit ~ gre + gpa + rank,
               data = mydata, family = "binomial")
sum<-summary(mylogit)
sum$coefficients
```

```
##                 Estimate  Std. Error   z value     Pr(>|z|)
## (Intercept) -3.989979073 1.139950936 -3.500132 0.0004650273
## gre          0.002264426 0.001093998  2.069864 0.0384651284
## gpa          0.804037549 0.331819298  2.423119 0.0153878974
## rank2       -0.675442928 0.316489661 -2.134171 0.0328288188
```

```
## rank3       -1.340203916 0.345306418 -3.881202 0.0001039415
## rank4       -1.551463677 0.417831633 -3.713131 0.0002047107
```

10.2.2 Classification

10.2.2.1 Support Vector Machines

10.2.2.2 Decision Trees

(Node)

- Classificaiton - Regression - Classification And Regression Tree (CART)

/ / /

```
install.packages("rpart")
```

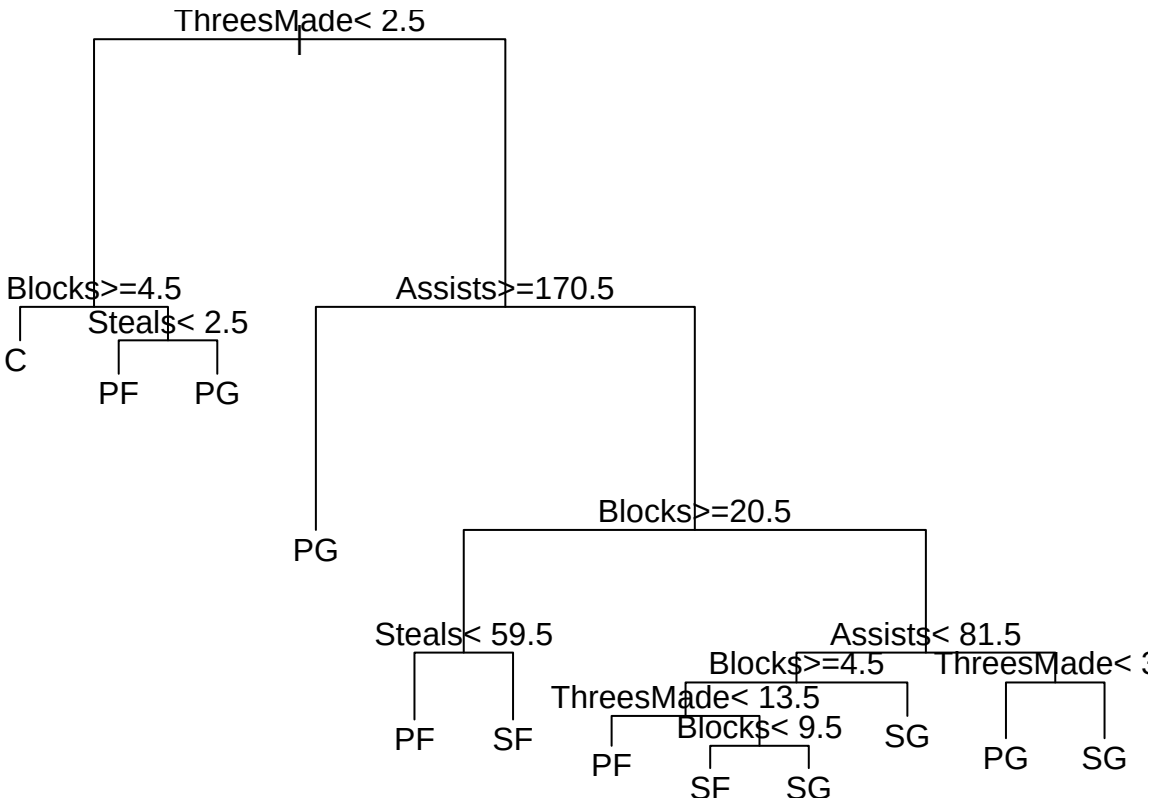
```
library(rpart)
DT<-rpart(Position~Blocks+ThreesMade+Assists+Steals,data=NBA1516)
DT
```

```
## n=475 (1 observation deleted due to missingness)
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 475 364 PF (0.15 0.23 0.21 0.18 0.23)
##     2) ThreesMade< 2.5 132  74 C (0.44 0.35 0.098 0.053 0.061)
##       4) Blocks>=4.5 89  37 C (0.58 0.38 0.011 0.011 0.011) *
##       5) Blocks< 4.5 43  31 PF (0.14 0.28 0.28 0.14 0.16)
##        10) Steals< 2.5 29  19 PF (0.17 0.34 0.14 0.21 0.14) *
##        11) Steals>=2.5 14   6 PG (0.071 0.14 0.57 0 0.21) *
##     3) ThreesMade>=2.5 343 242 SG (0.035 0.19 0.25 0.23 0.29)
##       6) Assists>=170.5 96  39 PG (0.031 0.052 0.59 0.15 0.18) *
##       7) Assists< 170.5 247 163 SG (0.036 0.24 0.12 0.26 0.34)
##        14) Blocks>=20.5 80  42 PF (0.062 0.48 0 0.26 0.2)
##          28) Steals< 59.5 58  21 PF (0.069 0.64 0 0.14 0.16) *
##          29) Steals>=59.5 22   9 SF (0.045 0.045 0 0.59 0.32) *
##        15) Blocks< 20.5 167  99 SG (0.024 0.13 0.17 0.26 0.41)
##          30) Assists< 81.5 110  68 SG (0.027 0.18 0.091 0.32 0.38)
##            60) Blocks>=4.5 63  39 SF (0.032 0.29 0.016 0.38 0.29)
##             120) ThreesMade< 13.5 19   9 PF (0.11 0.53 0 0.26 0.11) *
##             121) ThreesMade>=13.5 44  25 SF (0 0.18 0.023 0.43 0.36)
##               242) Blocks< 9.5 17   7 SF (0 0.18 0.059 0.59 0.18) *
##               243) Blocks>=9.5 27  14 SG (0 0.19 0 0.33 0.48) *
##            61) Blocks< 4.5 47  23 SG (0.021 0.043 0.19 0.23 0.51) *
##          31) Assists>=81.5 57  31 SG (0.018 0.035 0.33 0.16 0.46)
##            62) ThreesMade< 37 17   5 PG (0 0.12 0.71 0.059 0.12) *
##            63) ThreesMade>=37 40  16 SG (0.025 0 0.17 0.2 0.6) *
```

```
#   PG    SG    SF    PF    C
```

/ / /

```
par(mfrow=c(1,1), mar = rep(1,4)) #, , ,
plot(DT)
text(DT, use.n=F, all=F, cex=1)
```

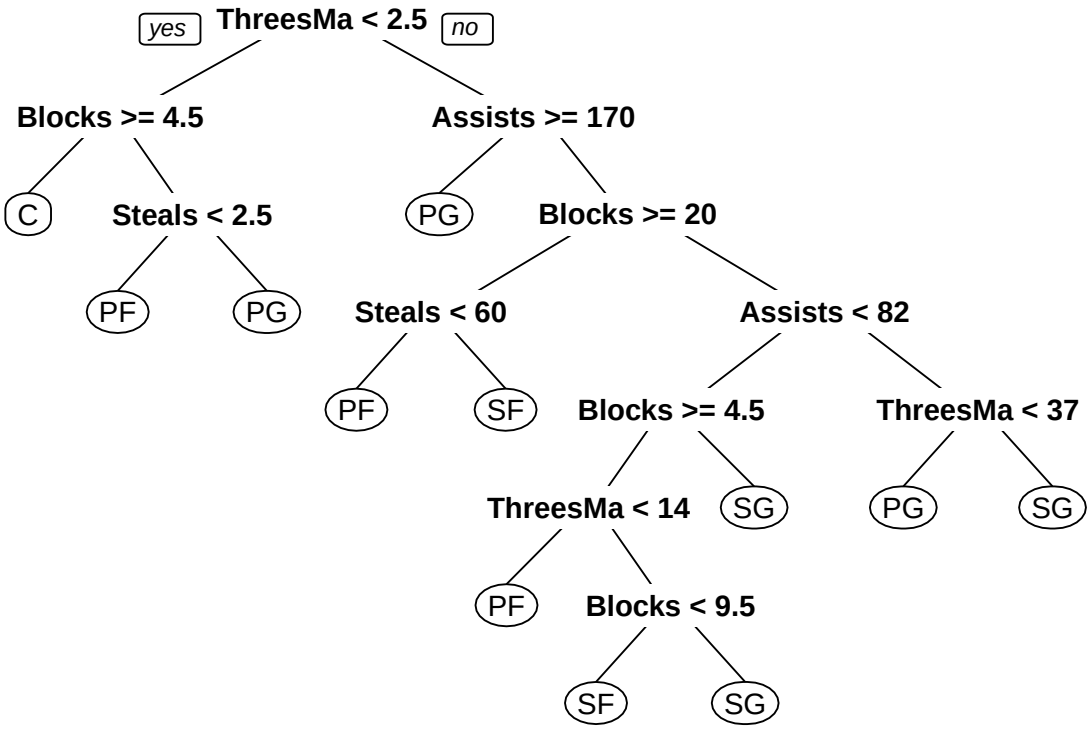



```
#   PG    SG    SF    PF    C
```

/ / / plot() rpart.plot package prp()

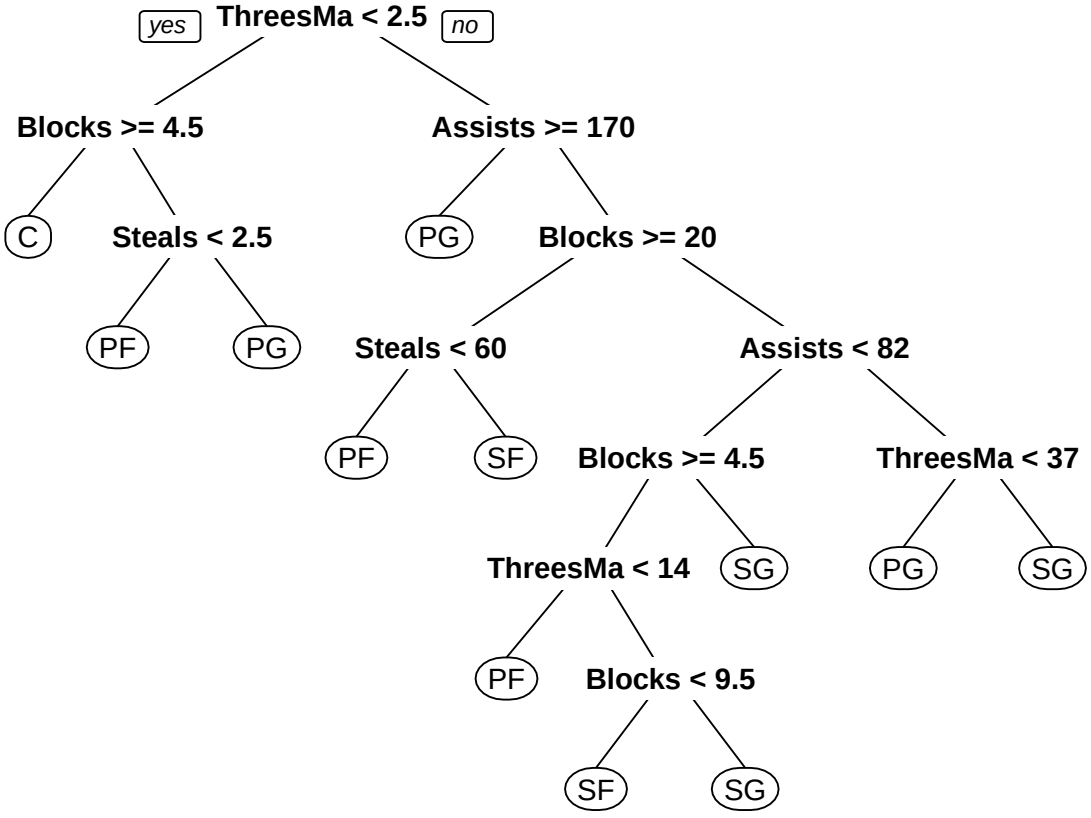
```
install.packages("rpart.plot")
```

```
library(rpart.plot)
prp(DT)                  # Will plot the tree
```



/ / /

```
prp(DT)
```



- Gini impurity - Information gain - Variance reduction

... .

10.2.2.3 Neural Networks

10.2.2.4 K-Nearest Neighbor

10.3

10.3.1 Clustering

/

Clustering organizes things that are `close` into groups

- How do we define close?
- How do we group things?
- How do we visualize the grouping?
- How do we interpret the grouping?

10.3.1.1 Hierarchical clustering

- An agglomerative approach
  - Find closest two things
  - Put them together
  - Find next closest
- Requires
  - A defined distance
  - A merging approach
- Produces
  - A tree showing how close things are to each other

Hierarchical clustering

- An agglomerative approach
  - Find closest two things
  - Put them together
  - Find next closest
- Requires
  - `A defined distance`
  - A merging approach
- Produces
  - A tree showing how close things are to each other

How do we define close? `distance`

- Most important step
  - Garbage in -> garbage out
- Distance or similarity
  - Continuous - euclidean distance
  - Continuous - correlation similarity
  - Binary - manhattan distance
- Pick a distance/similarity that makes sense for your problem

Example distances - Euclidean

$$\sqrt{(A_1 - A_2)^2 + (B_1 - B_2)^2 + \ldots + (Z_1 - Z_2)^2}$$

Example distances - Manhattan

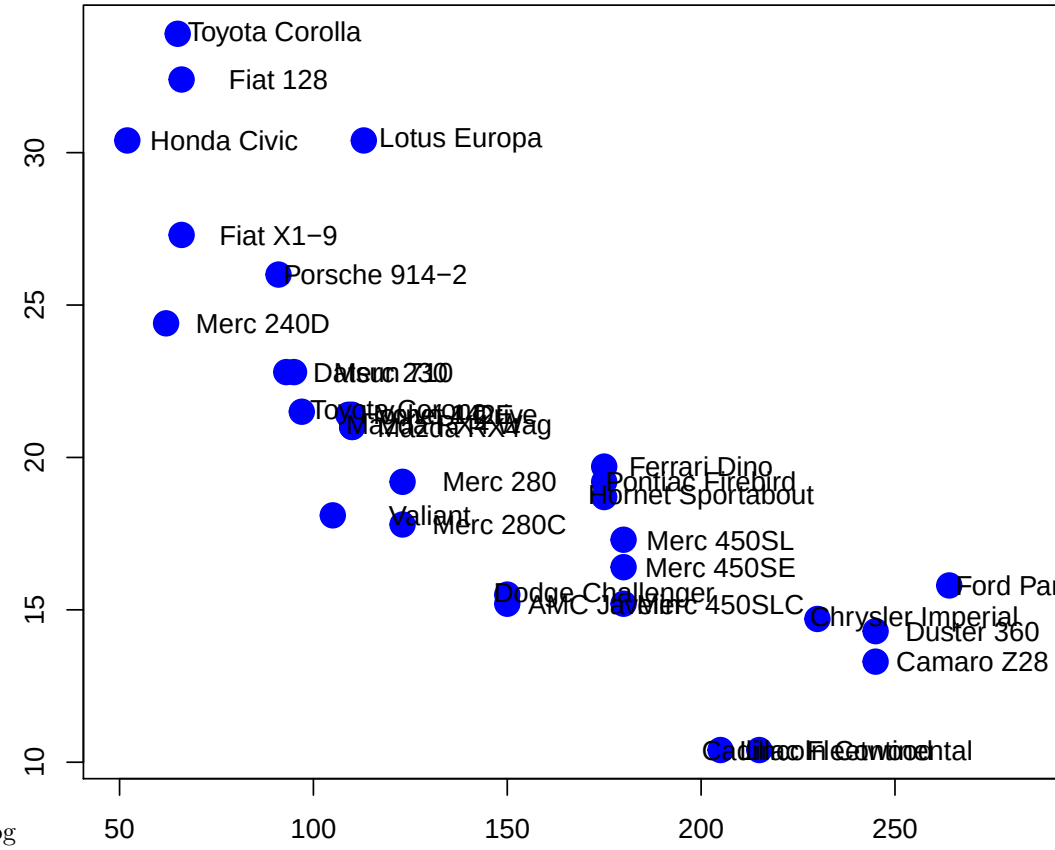
$$|A_1 - A_2| + |B_1 - B_2| + \ldots + |Z_1 - Z_2|$$

Green line: Euclidean, Blue line: Manhattan

Hierarchical clustering

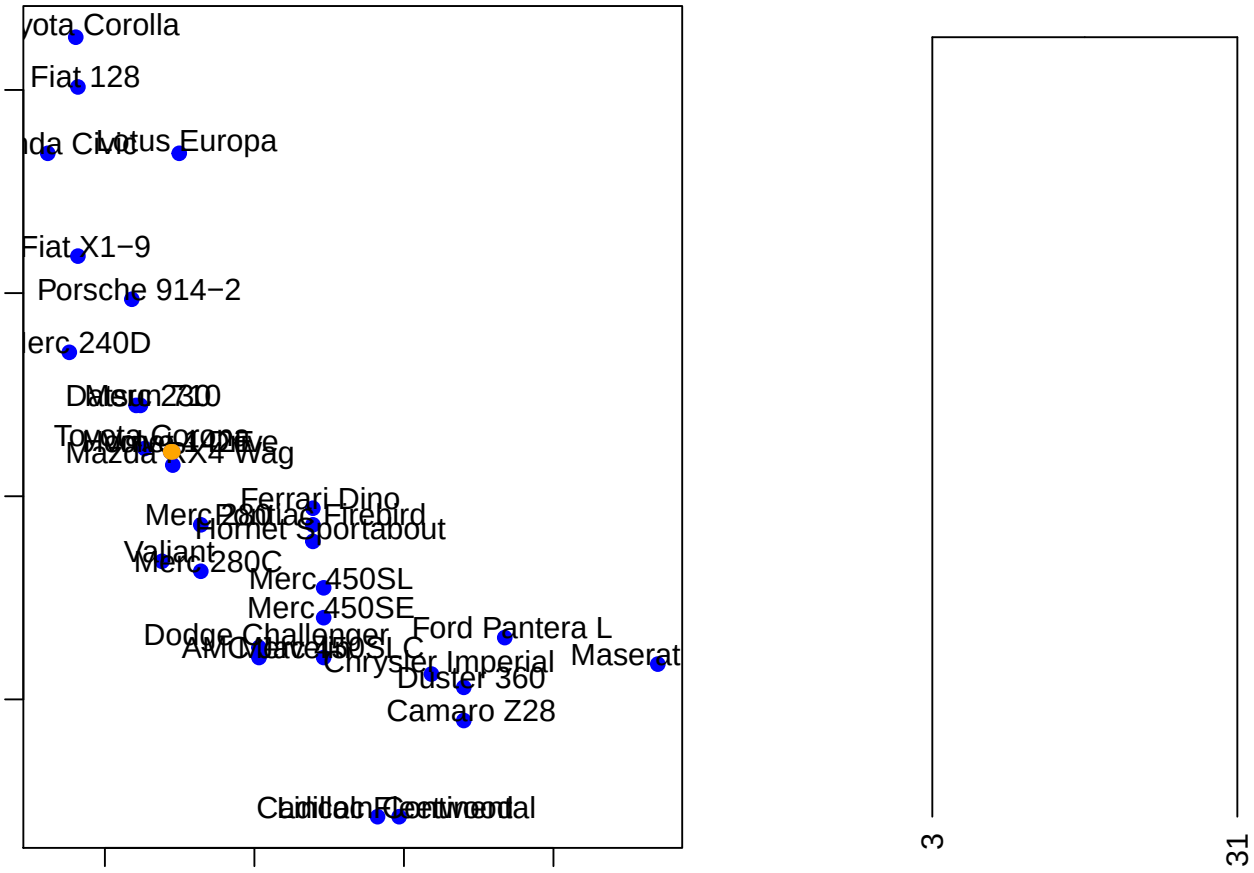
- An agglomerative approach
  - Find closest two things
  - Put them together
  - Find next closest
- Requires
  - A defined distance
  - `A merging approach`
- Produces
  - A tree showing how close things are to each other

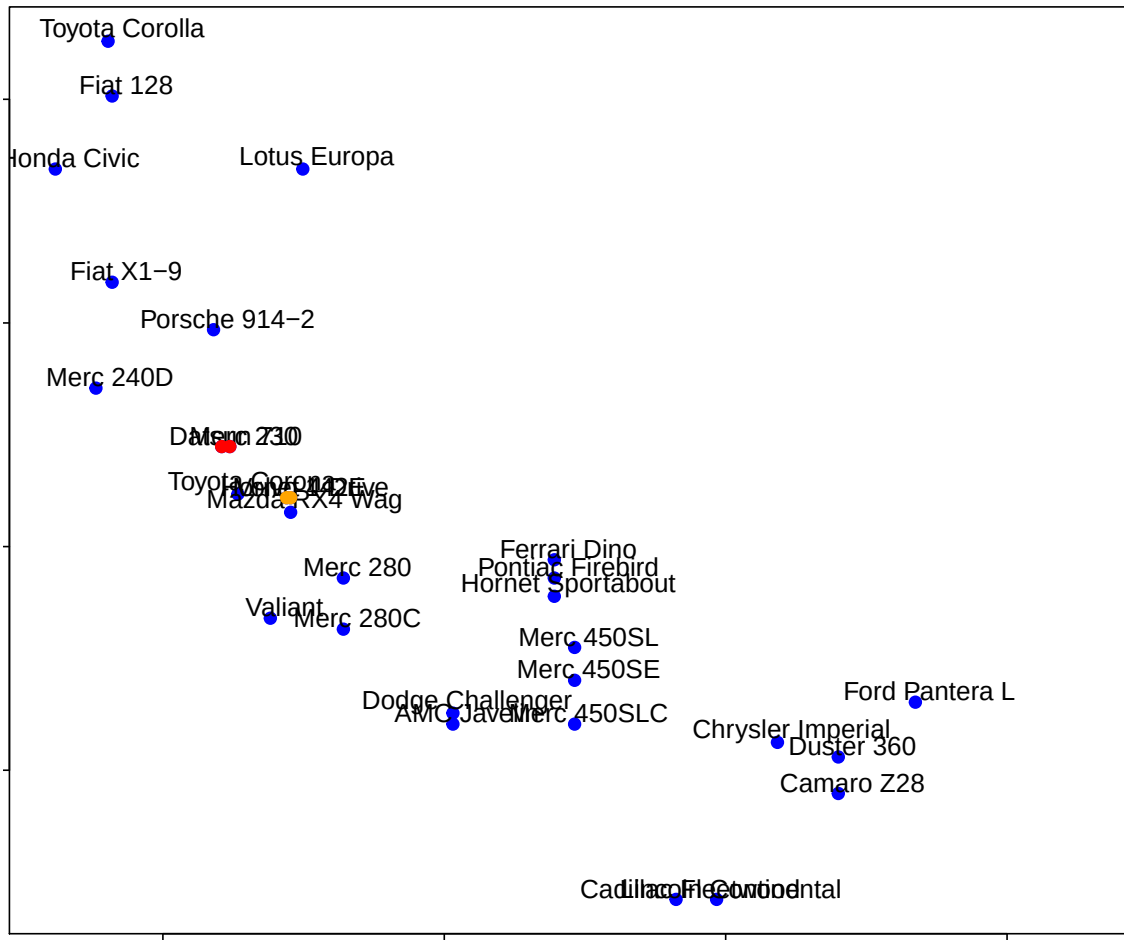
Merging apporach - Agglomerative - Single-linkage - Complete-linkage - Average-linkage



Hierarchical clustering - hp vs. mpg

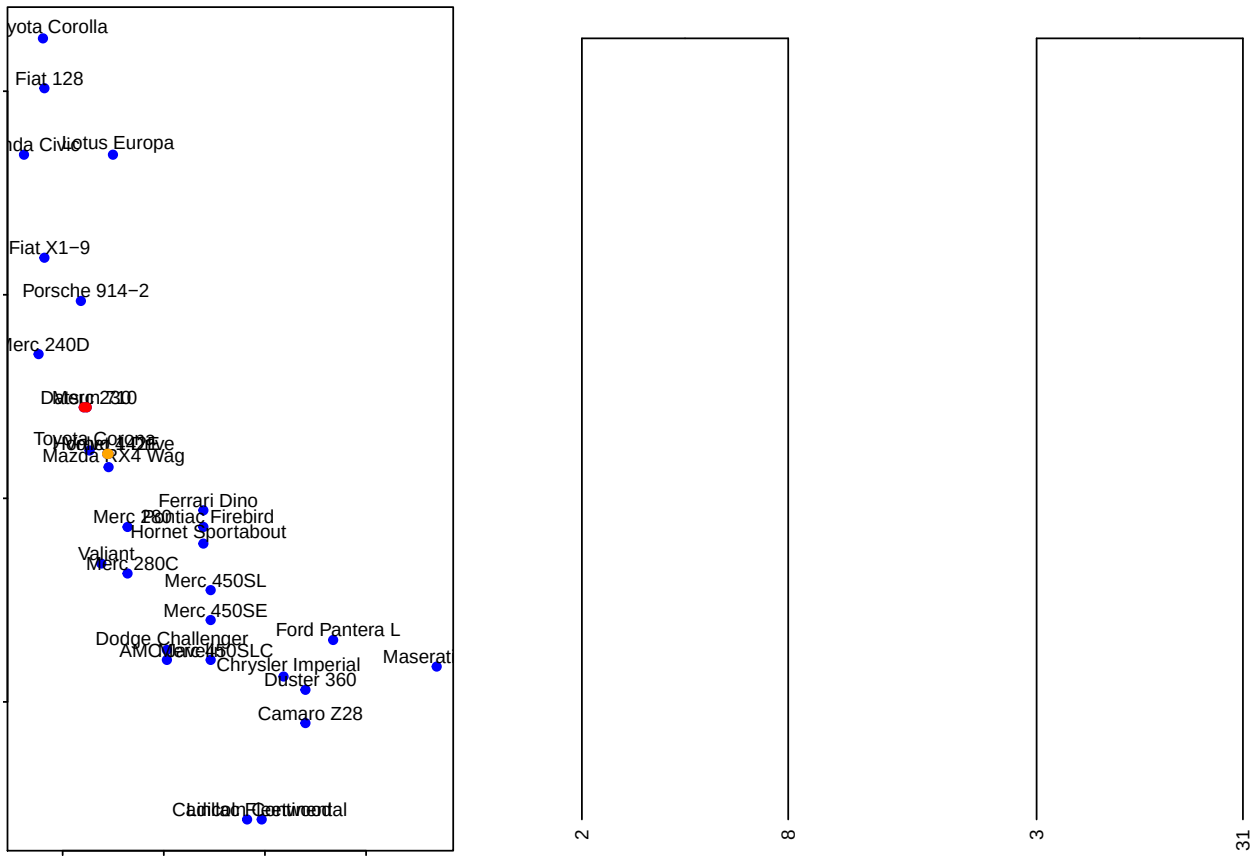
Hierarchical clustering - #1


yota Corolla
Fiat 128
Lotus Europa
Fiat X1−9
Porsche 914−2
erc 240D
Mazda RX4 Wag
Ferrari Dino
Hornet Sportabout
Merc 280C
Merc 450SL
Merc 450SE
Ford Pantera L
Chrysler Imperial
Duster 360
Camaro Z28
Maserat
3
31



Hierarchical clustering - #2

Hierarchical clustering - #3



Hierarchical clustering

- An agglomerative approach
  - Find closest two things
  - Put them together
  - Find next closest
- `Requires`
  - A defined distance
  - A merging approach
- Produces
  - A tree showing how close things are to each other

Hierarchical Clustering -dist() `dist()` method=“”

```
mtcars.mxs<-as.matrix(mtcars)
d<-dist(mtcars.mxs) # euclidean
d
```

```
##                    Mazda RX4 Mazda RX4 Wag  Datsun 710 Hornet 4 Drive
## Mazda RX4 Wag       0.6153251
## Datsun 710         54.9086059    54.8915169
## Hornet 4 Drive     98.1125212    98.0958939 150.9935191
## Hornet Sportabout 210.3374396   210.3358546 265.0831615    121.0297564
## Valiant            65.4717710    65.4392224 117.7547018     33.5508692
## Duster 360        241.4076490   241.4088680 294.4790230    169.4299647
## Merc 240D          50.1532711    50.1146059  49.6584796    121.2739722
## Merc 230           25.4683117    25.3284509  33.1803843    118.2433145
## Merc 280           15.3641921    15.2956865  66.9363534     91.4224033
## Merc 280C          15.6724727    15.5837744  67.0261397     91.4612914
```

```
## Merc 450SE          135.4307018   135.4254826 189.1954941     72.4964325
## Merc 450SL          135.4014424   135.3960351 189.1631745     72.4313532
## Merc 450SLC         135.4794674   135.4723157 189.2345426     72.5718466
## Cadillac Fleetwood  326.3395903   326.3355070 381.0926242    234.4403876
## Lincoln Continental 318.0469808   318.0429333 372.8012090    227.9726091
## Chrysler Imperial   304.7203408   304.7169175 359.3014906    218.1548299
## Fiat 128             93.2679950    93.2530993  40.9933763    184.9689734
## Honda Civic         102.8307567   102.8238713  52.7704607    191.5518700
## Toyota Corolla      100.6040368   100.5887588  47.6535017    192.6714187
## Toyota Corona        42.3075233    42.2659224  12.9654743    138.5304725
## Dodge Challenger    163.1150750   163.1134210 217.7795805     72.4403915
## AMC Javelin         149.6047203   149.6014522 204.3188913     61.3601899
## Camaro Z28          233.2228758   233.2248748 286.0049209    163.6632641
## Pontiac Firebird    248.6780270   248.6762035 303.3583889    156.2240346
## Fiat X1-9            92.5048389    92.4940020  39.8815148    184.4471198
## Porsche 914-2        44.4033659    44.4073589  13.1357109    139.1579524
## Lotus Europa         65.7328377    65.7362635  25.0948550    163.2367437
## Ford Pantera L      245.4247064   245.4293785 297.2940489    180.1140339
## Ferrari Dino         66.7661029    66.7764167  90.2415509    130.5523007
## Maserati Bora       265.6454248   265.6491465 309.7718171    229.3419352
## Volvo 142E           39.1894029    39.1626037  20.6939436    137.0363299
##                    Hornet Sportabout     Valiant  Duster 360   Merc 240D
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant                  152.1241352
## Duster 360                70.1767262 194.6094525
## Merc 240D                241.5069657  89.5911056 281.2962502
## Merc 230                 233.4924012  85.0079649 265.8823313  33.6873047
## Merc 280                 199.3344960  60.2909811 227.8998521  64.7754228
## Merc 280C                199.3406564  60.2655656 227.8813169  64.8898713
## Merc 450SE                84.3888482  90.6970264 106.4084264 175.1620073
## Merc 450SL                84.3683999  90.6769728 106.4320572 175.1189767
## Merc 450SLC               84.4332423  90.7092989 106.4010305 175.2118218
## Cadillac Fleetwood       116.2804201 266.6280942 119.0239068 355.6627498
## Lincoln Continental      108.0624299 259.6304391 104.5112999 348.9901277
## Chrysler Imperial         97.2049146 248.7713290  81.4297699 338.1959373
## Fiat 128                 302.0377212 152.1153263 333.9792070  68.6105903
## Honda Civic              310.0324645 158.9615769 344.0518316  72.0014488
## Toyota Corolla           309.5581776 159.8302995 341.0218232  76.2806458
## Toyota Corona            252.3331988 105.2876428 282.0508820  44.0850975
## Dodge Challenger          48.9838851 103.4310693 103.9023864 192.8617917
## AMC Javelin               61.4274240  91.0444349 110.3084921 180.5479760
## Camaro Z28                70.9665308 187.8463771  10.0761203 273.8367985
## Pontiac Firebird          40.0052475 188.5272116  80.8057339 277.4606884
## Fiat X1-9                301.5669483 151.4379425 333.4843231  67.9163981
## Porsche 914-2            254.1452553 106.0585767 285.1986201  39.4469276
## Lotus Europa             272.3582423 130.8248192 296.4572287  72.8971106
## Ford Pantera L            89.5934049 203.0177926  21.2655990 287.5238795
## Ferrari Dino             215.0673853 106.5694802 226.2036333 113.3023005
## Maserati Bora            170.7094473 242.4393015 107.7224977 313.8633093
## Volvo 142E               248.0063378 104.1863681 275.1353516  53.6823481
##                       Merc 230    Merc 280   Merc 280C  Merc 450SE
```

```
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280             39.2994160
## Merc 280C            39.3868519   1.5231546
## Merc 450SE          159.8179555 122.3642489 122.3461050
## Merc 450SL          159.7760899 122.3443771 122.3355492   0.9826495
## Merc 450SLC         159.8495837 122.3934970 122.3586862   1.3726252
## Cadillac Fleetwood  349.2832611 315.3904859 315.3557081 197.8842803
## Lincoln Continental 341.3154316 306.6760719 306.6406187 187.5997191
## Chrysler Imperial   328.4335161 292.7146896 292.6989332 171.6600758
## Fiat 128             69.3127910 106.5053149 106.6829794 228.3247948
## Honda Civic          78.5387212 116.7280991 116.8711475 238.0141824
## Toyota Corolla       76.7731674 113.6290721 113.8118009 235.5183809
## Toyota Corona        21.0962017  54.3641713  54.4258314 176.6020527
## Dodge Challenger    185.8331870 152.8929263 152.8722437  51.8008639
## AMC Javelin         172.5312555 139.1457974 139.1181977  41.2080044
## Camaro Z28          257.7469734 219.5520854 219.5276434  98.7203049
## Pontiac Firebird    271.3871978 238.1726099 238.1806292 124.3368538
## Fiat X1-9            68.5564864 105.7412910 105.8560373 227.7627676
## Porsche 914-2        22.1180967  57.6458160  57.8473863 179.5034108
## Lotus Europa         50.1094030  74.1443580  74.3824296 193.3074449
## Ford Pantera L      269.9772035 231.4081306 231.4024263 112.8181834
## Ferrari Dino         80.6550953  56.8365103  56.8987601 131.0272205
## Maserati Bora       288.8755628 250.5874125 250.5774357 157.1633256
## Volvo 142E           24.6913548  48.8053450  48.8884618 170.4500681
##                      Merc 450SL Merc 450SLC Cadillac Fleetwood
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280
## Merc 280C
## Merc 450SE
## Merc 450SL
## Merc 450SLC           2.1383405
## Cadillac Fleetwood  197.9154476 197.8526242
## Lincoln Continental 187.6330806 187.5671081         15.6224446
## Chrysler Imperial   171.6743028 171.6557637         40.8399636
## Fiat 128            228.2592340 228.4051825        417.7687579
## Honda Civic         237.9588183 238.0828999        425.3271621
## Toyota Corolla      235.4481971 235.6024098        425.3446517
## Toyota Corona       176.5727477 176.6305359        368.3195488
## Dodge Challenger     51.8242520  51.8012606        163.6314881
## AMC Javelin          41.2411618  41.1929050        176.8610896
```

```
## Camaro Z28           98.7566899  98.7035830        128.4587210
## Pontiac Firebird    124.3204160 124.3726128         78.5385347
## Fiat X1-9           227.7173075 227.8176554        417.2490481
## Porsche 914-2       179.4550855 179.5720446        370.0956775
## Lotus Europa        193.2407697 193.3969216        388.5350012
## Ford Pantera L      112.8296774 112.8332602        134.8119464
## Ferrari Dino        131.0077635 131.0704490        328.5441628
## Maserati Bora       157.1768956 157.1683970        214.9366858
## Volvo 142E          170.4225164 170.4843735        364.1000930
##                     Lincoln Continental Chrysler Imperial    Fiat 128
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280
## Merc 280C
## Merc 450SE
## Merc 450SL
## Merc 450SLC
## Cadillac Fleetwood
## Lincoln Continental
## Chrysler Imperial            25.3714237
## Fiat 128                    410.0206984       397.2276375
## Honda Civic                 417.9679574       405.8152201  14.5590942
## Toyota Corolla              417.5429986       404.6335386   7.8324789
## Toyota Corona               360.0267515       346.5724649  52.8798281
## Dodge Challenger            156.2805020       145.9194779 254.2367888
## AMC Javelin                 169.0925457       157.8097554 241.1203621
## Camaro Z28                  114.0932078        91.2880886 325.6636235
## Pontiac Firebird             72.6947903        68.2030747 339.5857659
## Fiat X1-9                   409.4998363       396.7597522   5.1473415
## Porsche 914-2               362.0145494       348.8466861  49.0644372
## Lotus Europa                379.4716659       364.5994326  49.9112509
## Ford Pantera L              119.7236456        95.3805385 337.1639236
## Ferrari Dino                317.7063117       300.1640703 128.3950054
## Maserati Bora               199.3420611       174.2936864 349.5338830
## Volvo 142E                  355.4009443       341.2896659  61.3301247
##                     Honda Civic Toyota Corolla Toyota Corona
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280
## Merc 280C
## Merc 450SE
## Merc 450SL
```

```
## Merc 450SLC
## Cadillac Fleetwood
## Lincoln Continental
## Chrysler Imperial
## Fiat 128
## Honda Civic
## Toyota Corolla       14.3480626
## Toyota Corona        63.8985563     59.8451285
## Dodge Challenger    261.8498815    261.8345312   205.0347927
## AMC Javelin         248.9636504    248.6917065   191.5580526
## Camaro Z28          335.8883188    332.6589699   273.6316895
## Pontiac Firebird    347.0655360    347.1667643   290.6240706
## Fiat X1-9            14.7807070     10.3922856    51.8411748
## Porsche 914-2        59.4588768     56.3243031     8.6535903
## Lotus Europa         64.0495153     53.8846563    31.2536926
## Ford Pantera L      347.8337714    343.9920962   285.1287911
## Ferrari Dino        141.7044478    133.4707617    82.2355734
## Maserati Bora       362.1620777    355.2601619   299.1865216
## Volvo 142E           73.3766041     67.7189421    12.2505275
##                    Dodge Challenger AMC Javelin  Camaro Z28
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280
## Merc 280C
## Merc 450SE
## Merc 450SL
## Merc 450SLC
## Cadillac Fleetwood
## Lincoln Continental
## Chrysler Imperial
## Fiat 128
## Honda Civic
## Toyota Corolla
## Toyota Corona
## Dodge Challenger
## AMC Javelin           14.0154995
## Camaro Z28           100.3046106 105.6062618
## Pontiac Firebird      85.8075196  99.2836114  86.2665759
## Fiat X1-9            253.6624046 240.5266823 325.1490914
## Porsche 914-2        206.6452569 193.3080584 276.8924414
## Lotus Europa         226.5004836 212.7568765 287.6179004
## Ford Pantera L       118.7516779 123.3832044  19.3589023
## Ferrari Dino         174.9280395 161.1060307 216.7489910
## Maserati Bora        185.9059273 185.1553411 102.5946154
## Volvo 142E           201.3682522 187.6978440 266.5277736
##                    Pontiac Firebird   Fiat X1-9 Porsche 914-2
## Mazda RX4 Wag
## Datsun 710
```

```
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280
## Merc 280C
## Merc 450SE
## Merc 450SL
## Merc 450SLC
## Cadillac Fleetwood
## Lincoln Continental
## Chrysler Imperial
## Fiat 128
## Honda Civic
## Toyota Corolla
## Toyota Corona
## Dodge Challenger
## AMC Javelin
## Camaro Z28
## Pontiac Firebird
## Fiat X1-9               339.1396182
## Porsche 914-2           292.1646488  48.3775209
## Lotus Europa            311.3862342  49.8406880    33.7678653
## Ford Pantera L          101.7389686 336.7018783   288.5852993
## Ferrari Dino            255.0570519 127.8210813    87.9105966
## Maserati Bora           188.3240020 349.1199576   303.9222549
## Volvo 142E              286.7497823  60.4120429    18.7555858
##                     Lotus Europa Ford Pantera L Ferrari Dino Maserati Bora
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280
## Merc 280C
## Merc 450SE
## Merc 450SL
## Merc 450SLC
## Cadillac Fleetwood
## Lincoln Continental
## Chrysler Imperial
## Fiat 128
## Honda Civic
## Toyota Corolla
## Toyota Corona
## Dodge Challenger
## AMC Javelin
## Camaro Z28
## Pontiac Firebird
```

```
## Fiat X1-9
## Porsche 914-2
## Lotus Europa
## Ford Pantera L       297.5376920
## Ferrari Dino          80.4553451    224.4587490
## Maserati Bora        303.2796468     86.9383253  223.5342175
## Volvo 142E            27.8104457    277.4803312   70.4751034   289.1157363
```

Hierarchical Clustering -dist() `dist()` method="" "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski"

```
d<-dist(mtcars.mxs, method="manhattan") # manhattan
d
```

```
##                     Mazda RX4 Mazda RX4 Wag Datsun 710 Hornet 4 Drive
## Mazda RX4 Wag           0.815
## Datsun 710             79.300        78.995
## Hornet 4 Drive        108.795       107.980    174.895
## Hornet Sportabout     275.430       274.615    349.510        176.415
## Valiant                84.640        83.825    141.540         42.645
## Duster 360            347.960       348.265    427.160        254.185
## Merc 240D              75.020        74.205     75.720        167.495
## Merc 230               48.990        48.175     41.990        141.965
## Merc 280               27.080        26.265    100.700        111.805
## Merc 280C              29.080        28.265    102.080        112.605
## Merc 450SE            198.620       197.805    273.940        100.705
## Merc 450SL            197.580       196.765    272.500         99.265
## Merc 450SLC           200.130       199.315    274.250        101.015
## Cadillac Fleetwood    426.720       425.905    502.880        329.645
## Lincoln Continental   424.664       423.849    501.144        327.909
## Chrysler Imperial     414.655       413.840    491.935        319.000
## Fiat 128              146.310       146.005     67.110        240.345
## Honda Civic           160.795       160.490     83.775        258.670
## Toyota Corolla        157.345       157.040     78.145        251.380
## Toyota Corona          65.305        65.000     21.095        154.940
## Dodge Challenger      211.950       211.435    286.330        113.095
## AMC Javelin           198.205       197.390    271.725         98.630
## Camaro Z28            339.140       339.445    418.340        246.405
## Pontiac Firebird      315.435       314.620    389.455        216.220
## Fiat X1-9             140.605       140.300     61.405        235.720
## Porsche 914-2          69.950        70.285     23.170        173.465
## Lotus Europa           84.977        84.912     45.097        185.832
## Ford Pantera L        356.030       356.335    435.330        267.725
## Ferrari Dino           85.690        86.205    134.890        193.625
## Maserati Bora         382.170       382.475    461.370        293.055
## Volvo 142E             47.910        47.285     32.130        145.305
##                     Hornet Sportabout Valiant Duster 360 Merc 240D
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant                       213.210
## Duster 360                     77.770 289.740
## Merc 240D                     341.770 133.020    419.420
## Merc 230                      316.240 107.050    393.890    43.670
```

```
## Merc 280                  252.950  83.600    326.600    93.280
## Merc 280C                 253.950  82.200    325.800    94.080
## Merc 450SE                 93.590 136.240    154.500   266.200
## Merc 450SL                 92.550 134.800    155.260   264.760
## Merc 450SLC                95.100 136.550    153.610   266.510
## Cadillac Fleetwood        155.290 364.900    160.000   495.140
## Lincoln Continental       153.234 363.304    137.944   493.404
## Chrysler Imperial         143.385 354.555     98.775   484.195
## Fiat 128                  416.620 206.930    494.270    83.910
## Honda Civic               431.105 225.315    508.755    92.295
## Toyota Corolla            427.655 217.105    505.305    92.085
## Toyota Corona             331.215 120.445    408.865    67.245
## Dodge Challenger           70.820 148.010    141.730   278.590
## AMC Javelin                84.785 134.235    155.555   263.985
## Camaro Z28                 89.990 281.960     12.220   410.680
## Pontiac Firebird           41.005 253.975    118.515   381.715
## Fiat X1-9                 410.915 202.365    488.565    79.345
## Porsche 914-2             340.900 140.110    417.910    65.090
## Lotus Europa              349.267 162.477    426.677   115.457
## Ford Pantera L            109.760 303.770     35.250   429.950
## Ferrari Dino              227.660 166.870    298.950   133.390
## Maserati Bora             234.640 328.610    158.270   455.630
## Volvo 142E                317.900 119.950    395.550    78.930
##                     Merc 230 Merc 280 Merc 280C Merc 450SE Merc 450SL
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280             67.290
## Merc 280C            68.090    2.000
## Merc 450SE          240.670  175.380   174.580
## Merc 450SL          239.230  173.940   173.140     1.440
## Merc 450SLC         240.980  175.690   174.890     2.090     2.550
## Cadillac Fleetwood  469.610  402.320   401.520   230.100   231.140
## Lincoln Continental 467.874  400.584   399.784   228.044   229.084
## Chrysler Imperial   458.665  391.375   390.575   218.355   219.755
## Fiat 128            107.240  167.670   168.470   341.050   339.610
## Honda Civic         123.625  182.155   183.715   355.535   354.095
## Toyota Corolla      117.415  178.705   179.505   352.085   350.645
## Toyota Corona        29.795   84.705    85.505   255.645   254.205
## Dodge Challenger    253.060  189.770   188.970    75.490    76.250
## AMC Javelin         238.455  175.175   174.375    61.215    61.975
## Camaro Z28          385.070  317.780   316.980   146.180   147.160
## Pontiac Firebird    356.185  292.895   294.895   133.585   132.775
## Fiat X1-9           102.675  161.965   162.765   335.345   333.905
## Porsche 914-2        38.420   96.510    98.510   266.090   265.050
## Lotus Europa         81.087  103.177   105.177   274.457   273.417
## Ford Pantera L      403.920  337.170   336.370   168.750   169.510
## Ferrari Dino        104.380   83.870    85.870   150.850   149.810
## Maserati Bora       430.100  362.810   362.010   193.370   194.130
```

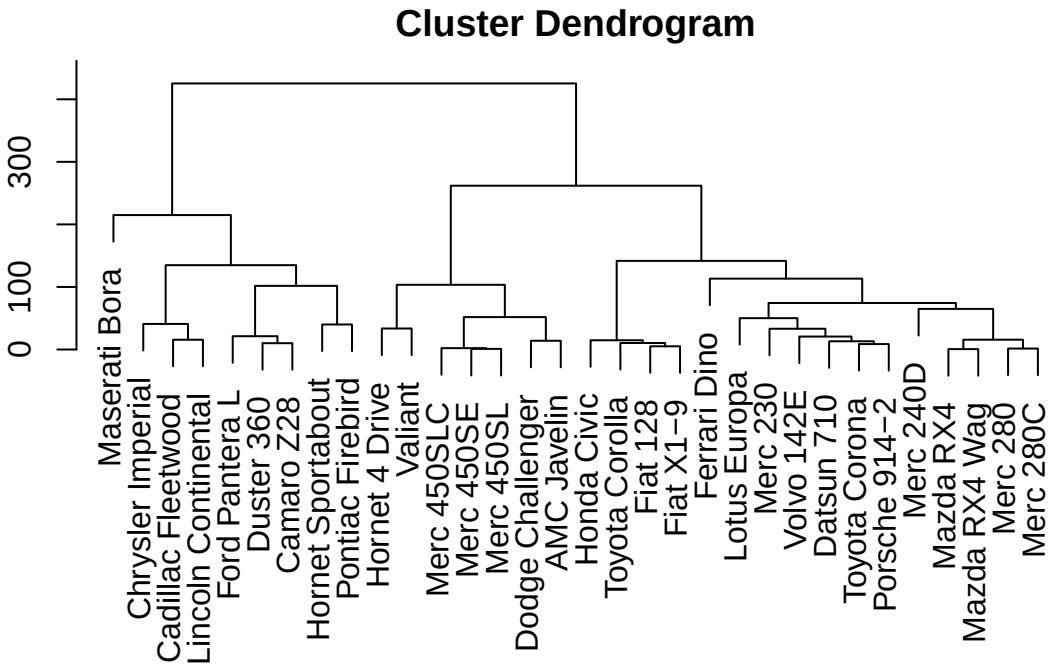
```
## Volvo 142E           41.060   68.950    70.350    242.330    240.890
##                    Merc 450SLC Cadillac Fleetwood Lincoln Continental
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280
## Merc 280C
## Merc 450SE
## Merc 450SL
## Merc 450SLC
## Cadillac Fleetwood      228.630
## Lincoln Continental     226.894             22.404
## Chrysler Imperial       218.005             62.255              40.009
## Fiat 128                341.360            569.990             568.254
## Honda Civic             355.845            584.475             582.739
## Toyota Corolla          352.395            581.025             579.289
## Toyota Corona           255.955            484.585             482.849
## Dodge Challenger         75.200            219.110             217.194
## AMC Javelin              60.325            232.515             230.459
## Camaro Z28              145.410            169.680             147.624
## Pontiac Firebird        135.225            115.285             113.229
## Fiat X1-9               335.655            564.285             562.549
## Porsche 914-2           267.600            496.190             494.134
## Lotus Europa            275.967            504.557             502.501
## Ford Pantera L          169.060            195.250             173.194
## Ferrari Dino            152.360            378.950             376.894
## Maserati Bora           192.480            318.270             296.214
## Volvo 142E              242.640            471.270             469.534
##                    Chrysler Imperial Fiat 128 Honda Civic Toyota Corolla
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280
## Merc 280C
## Merc 450SE
## Merc 450SL
## Merc 450SLC
## Cadillac Fleetwood
## Lincoln Continental
## Chrysler Imperial
## Fiat 128                     559.045
## Honda Civic                  573.530   22.385
## Toyota Corolla               570.080   11.035      24.410
## Toyota Corona                473.640   86.485     104.870         96.660
```

```
## Dodge Challenger             207.645  353.440     367.925        364.475
## AMC Javelin                  220.610  338.835     353.320        349.870
## Camaro Z28                   110.415  485.450     499.935        496.485
## Pontiac Firebird             103.520  456.565     471.050        467.600
## Fiat X1-9                    553.340    6.235      22.950         16.740
## Porsche 914-2                484.125   79.180      92.845         89.815
## Lotus Europa                 492.492   70.967      84.282         81.272
## Ford Pantera L               133.185  501.980     516.185        512.735
## Ferrari Dino                 366.885  202.000     216.485        213.035
## Maserati Bora                256.205  528.480     542.965        539.515
## Volvo 142E                   460.325   98.780     113.365        109.755
##                     Toyota Corona Dodge Challenger AMC Javelin Camaro Z28
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280
## Merc 280C
## Merc 450SE
## Merc 450SL
## Merc 450SLC
## Cadillac Fleetwood
## Lincoln Continental
## Chrysler Imperial
## Fiat 128
## Honda Civic
## Toyota Corolla
## Toyota Corona
## Dodge Challenger          268.035
## AMC Javelin               253.430           15.205
## Camaro Z28                400.105          133.950     147.775
## Pontiac Firebird          371.160          111.525     125.730    130.195
## Fiat X1-9                  81.920          347.735     333.130    479.745
## Porsche 914-2              20.065          277.420     263.675    409.090
## Lotus Europa               58.032          285.847     272.042    417.857
## Ford Pantera L            421.335          156.480     170.735     27.570
## Ferrari Dino              120.595          214.180     200.435    289.670
## Maserati Bora             447.075          214.600     200.425    148.970
## Volvo 142E                 18.135          254.720     240.115    386.730
##                     Pontiac Firebird Fiat X1-9 Porsche 914-2 Lotus Europa
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280
## Merc 280C
```

```
## Merc 450SE
## Merc 450SL
## Merc 450SLC
## Cadillac Fleetwood
## Lincoln Continental
## Chrysler Imperial
## Fiat 128
## Honda Civic
## Toyota Corolla
## Toyota Corona
## Dodge Challenger
## AMC Javelin
## Camaro Z28
## Pontiac Firebird
## Fiat X1-9                   450.860
## Porsche 914-2               380.905    73.355
## Lotus Europa                389.272    70.932        54.087
## Ford Pantera L              150.765   496.275       423.340      433.007
## Ferrari Dino                267.665   196.295       123.640      132.407
## Maserati Bora               275.385   522.775       450.120      458.887
## Volvo 142E                  357.845    93.075        28.160       43.207
##                     Ford Pantera L Ferrari Dino Maserati Bora
## Mazda RX4 Wag
## Datsun 710
## Hornet 4 Drive
## Hornet Sportabout
## Valiant
## Duster 360
## Merc 240D
## Merc 230
## Merc 280
## Merc 280C
## Merc 450SE
## Merc 450SL
## Merc 450SLC
## Cadillac Fleetwood
## Lincoln Continental
## Chrysler Imperial
## Fiat 128
## Honda Civic
## Toyota Corolla
## Toyota Corona
## Dodge Challenger
## AMC Javelin
## Camaro Z28
## Pontiac Firebird
## Fiat X1-9
## Porsche 914-2
## Lotus Europa
## Ford Pantera L
## Ferrari Dino               304.900
## Maserati Bora              126.980      326.480
## Volvo 142E                 403.200      103.300       429.760
```

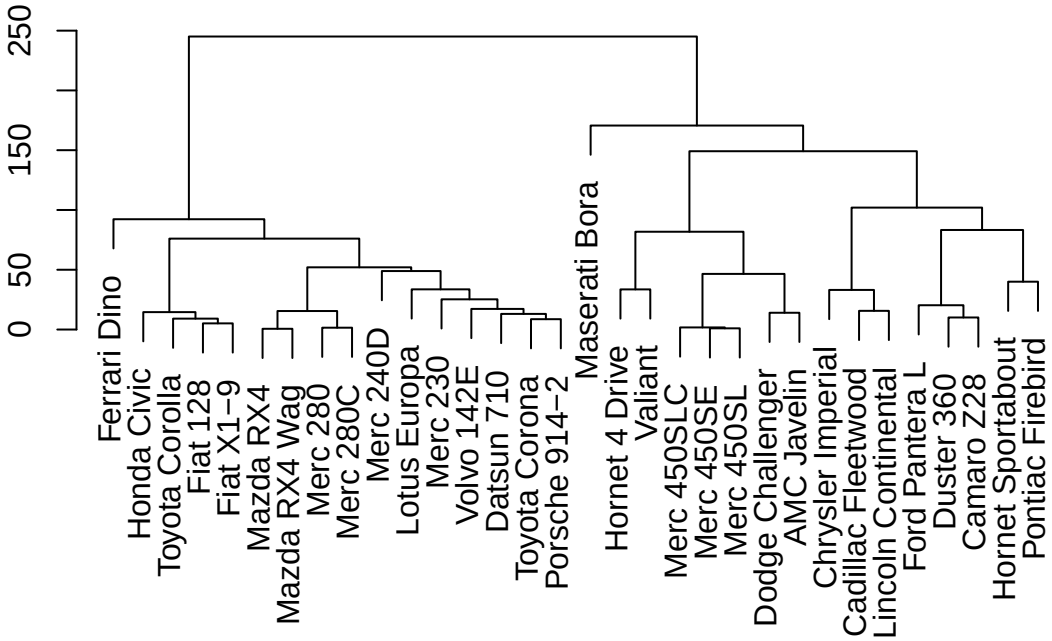
Hierarchical Clustering -hclust() `hclust` `dist()`

```
par(mar=rep(2,4),mfrow=c(1,1))
hc<-hclust(dist(mtcars.mxs)) # method       complete
plot(hc)
```



Hierarchical Clustering -hclust() `hclust` `dist()`

```
par(mar=rep(2,4),mfrow=c(1,1))
hc<-hclust(dist(mtcars.mxs),method="average") #
plot(hc)
```



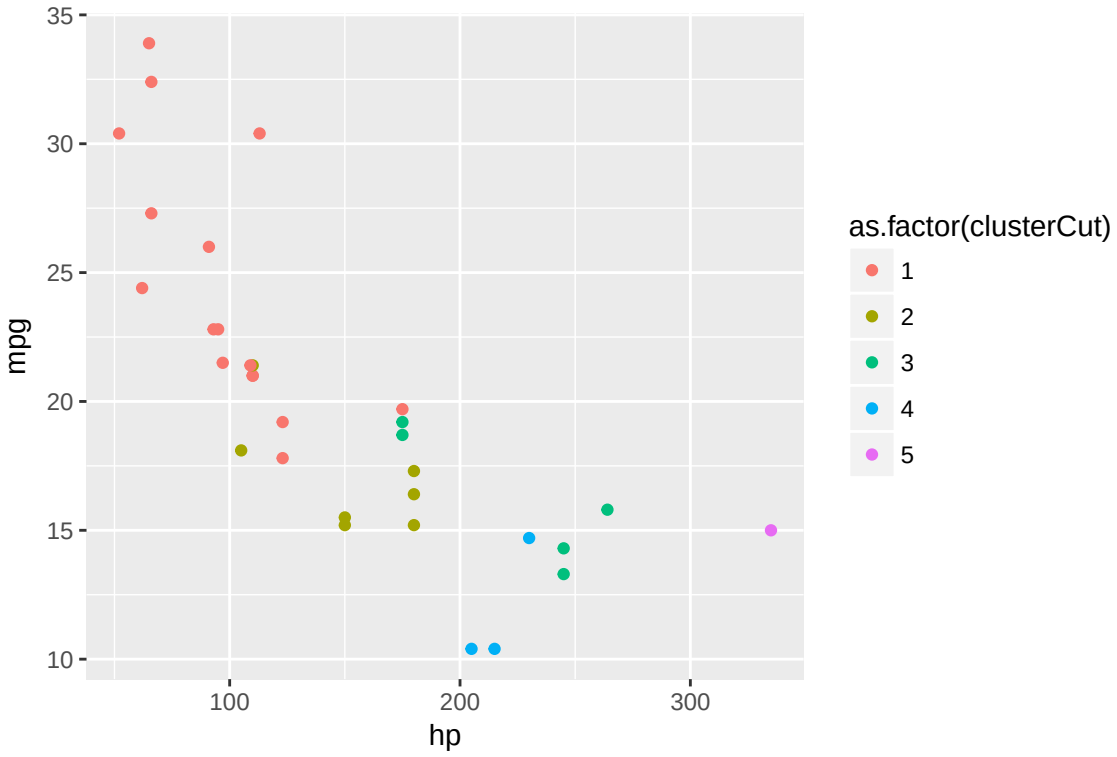
Hierarchical Clustering -cutree()

```
clusterCut <- cutree(hc, k=5) # 5
sort(clusterCut)
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710
##                   1                   1                   1
##           Merc 240D            Merc 230            Merc 280
##                   1                   1                   1
##           Merc 280C           Fiat 128         Honda Civic
##                   1                   1                   1
##      Toyota Corolla      Toyota Corona          Fiat X1-9
##                   1                   1                   1
##       Porsche 914-2       Lotus Europa        Ferrari Dino
##                   1                   1                   1
##          Volvo 142E     Hornet 4 Drive             Valiant
##                   1                   2                   2
##          Merc 450SE          Merc 450SL         Merc 450SLC
##                   2                   2                   2
##    Dodge Challenger         AMC Javelin   Hornet Sportabout
##                   2                   2                   3
##          Duster 360          Camaro Z28    Pontiac Firebird
##                   3                   3                   3
##      Ford Pantera L  Cadillac Fleetwood Lincoln Continental
##                   3                   4                   4
##   Chrysler Imperial       Maserati Bora
##                   4                   5
```

HC- clusters & variables

```
ggplot()+geom_point(data=mtcars,
                    aes(x=hp,y=mpg,color=as.factor(clusterCut)))
```



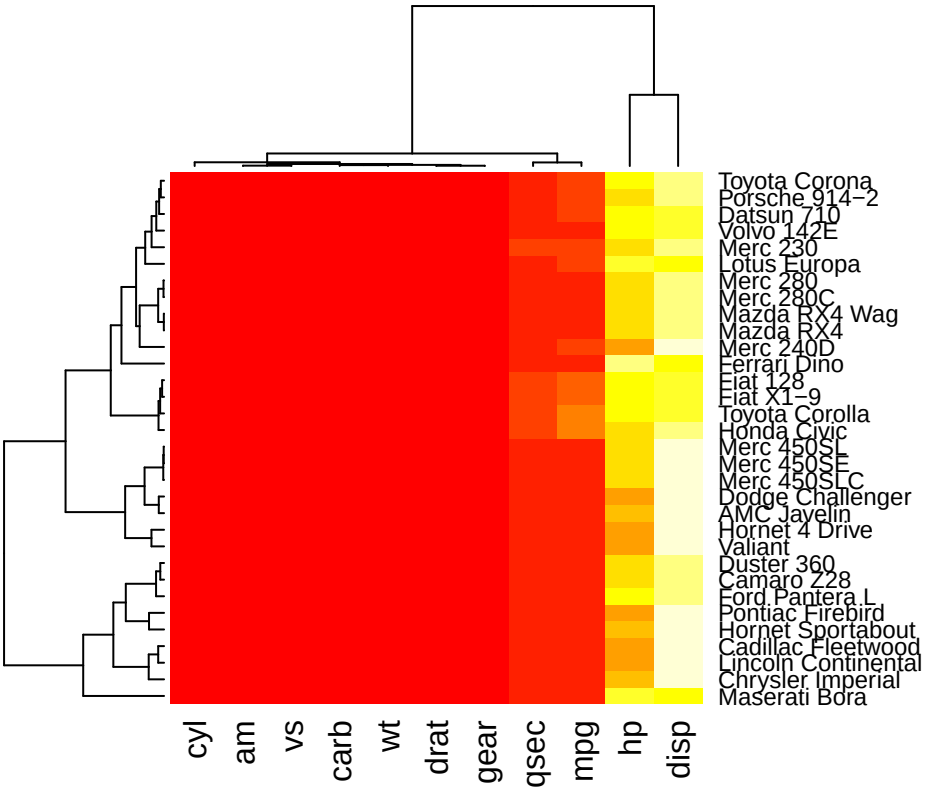
Hierarchical Clustering -cutree(),2

```
clusterCut <- cutree(hc,h =4) #  =4    =4
sort(clusterCut)
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710
##                   1                   1                   2
##      Hornet 4 Drive   Hornet Sportabout             Valiant
##                   3                   4                   5
##          Duster 360           Merc 240D            Merc 230
##                   6                   7                   8
##            Merc 280           Merc 280C          Merc 450SE
##                   9                   9                  10
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
##                  10                  10                  11
## Lincoln Continental   Chrysler Imperial            Fiat 128
##                  12                  13                  14
##         Honda Civic      Toyota Corolla       Toyota Corona
##                  15                  16                  17
##    Dodge Challenger         AMC Javelin          Camaro Z28
##                  18                  19                  20
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2
##                  21                  22                  23
##        Lotus Europa      Ford Pantera L        Ferrari Dino
##                  24                  25                  26
##       Maserati Bora          Volvo 142E
##                  27                  28
```

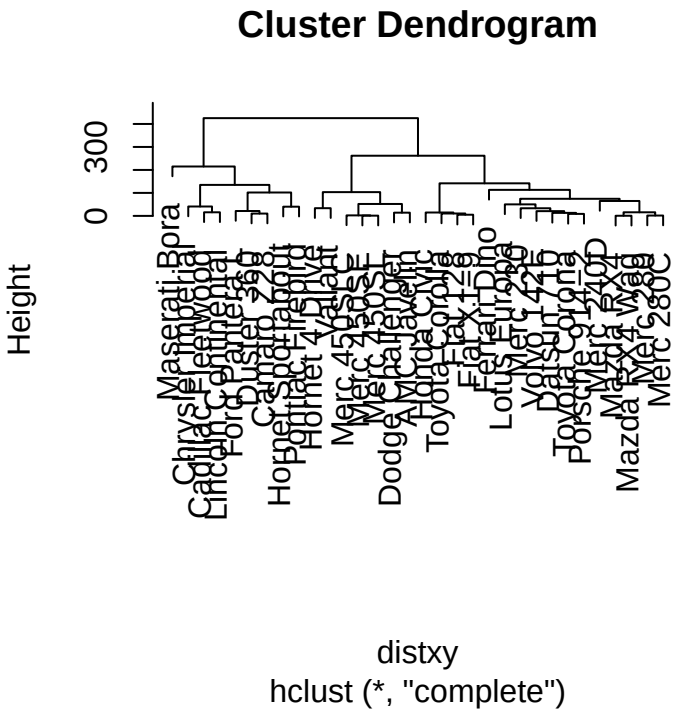
Cluster the data -heatmap(),2

```
par(mar=rep(0.2,4),mfrow=c(1,1))
heatmap(mtcars.mxs)
```



Hierarchical clustering - hclust

```
distxy <- dist(mtcars.mxs)
hClustering <- hclust(distxy)
plot(hClustering)
```



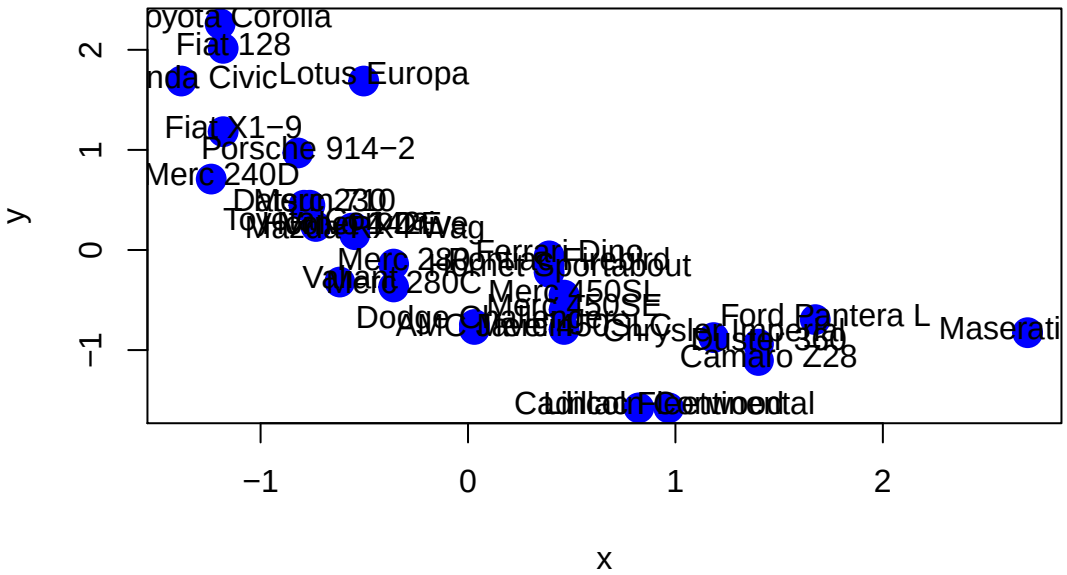
Hierarchical clustering: summary -

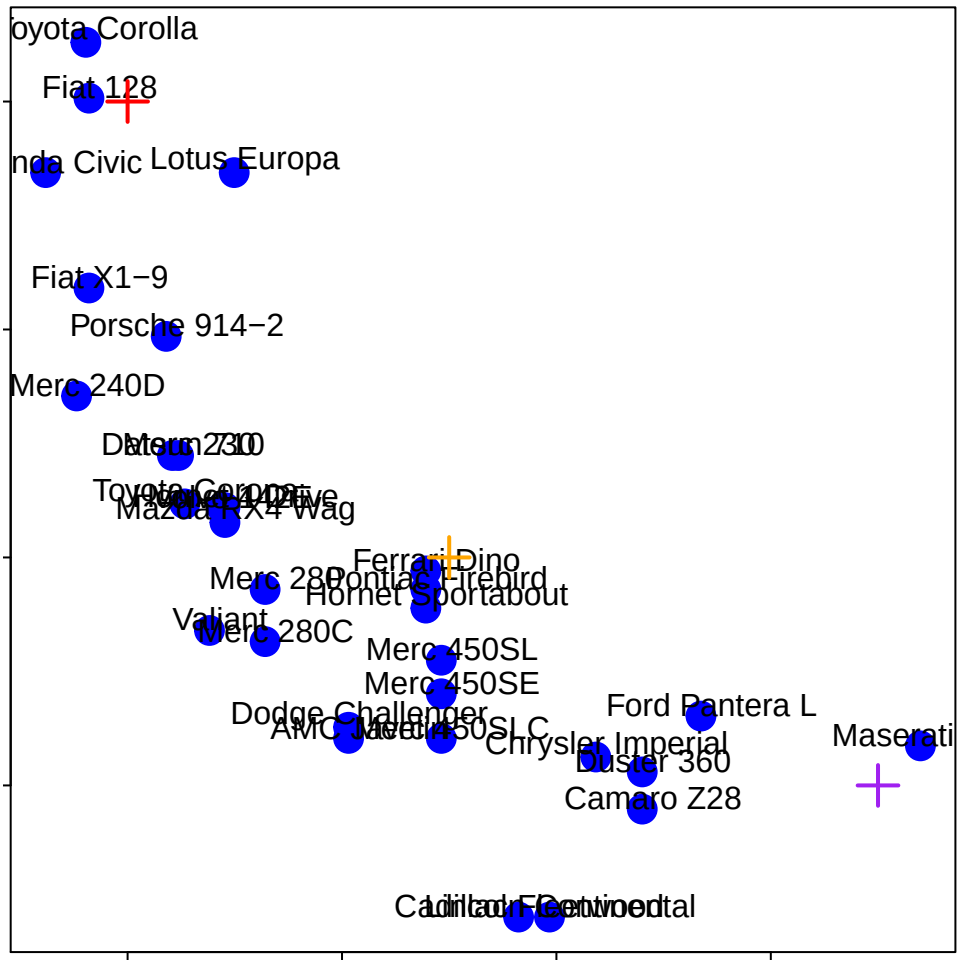
- –
  –
  –
- –

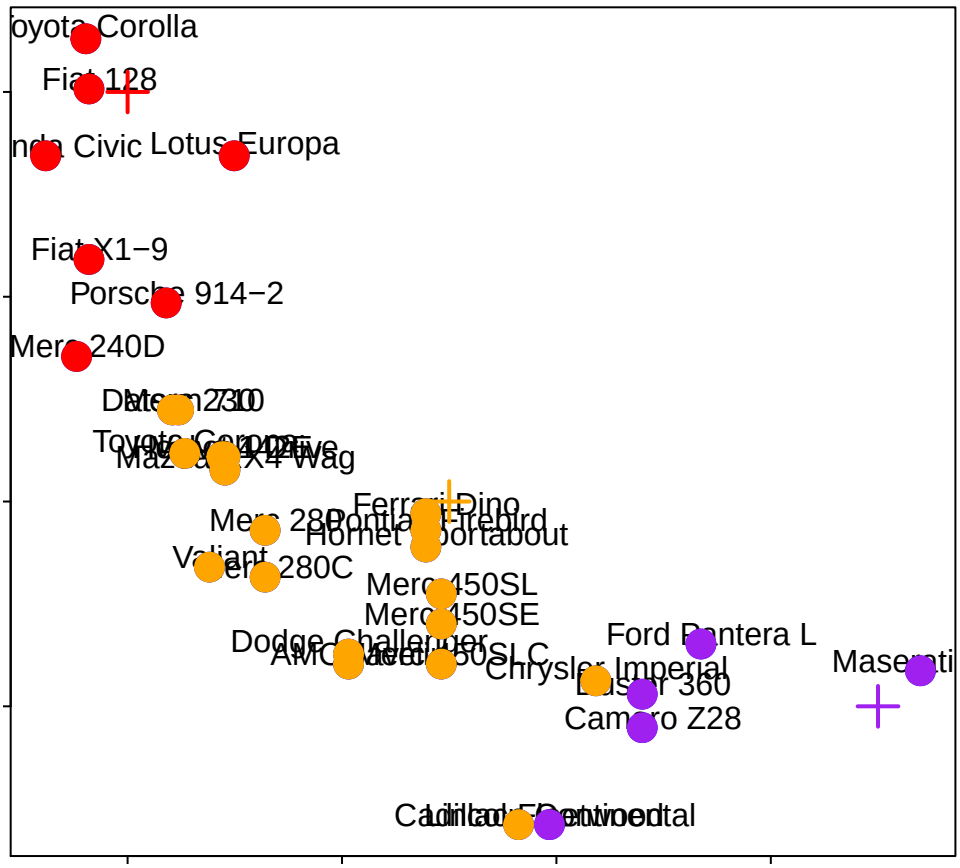
10.3.1.2 K-means clustering

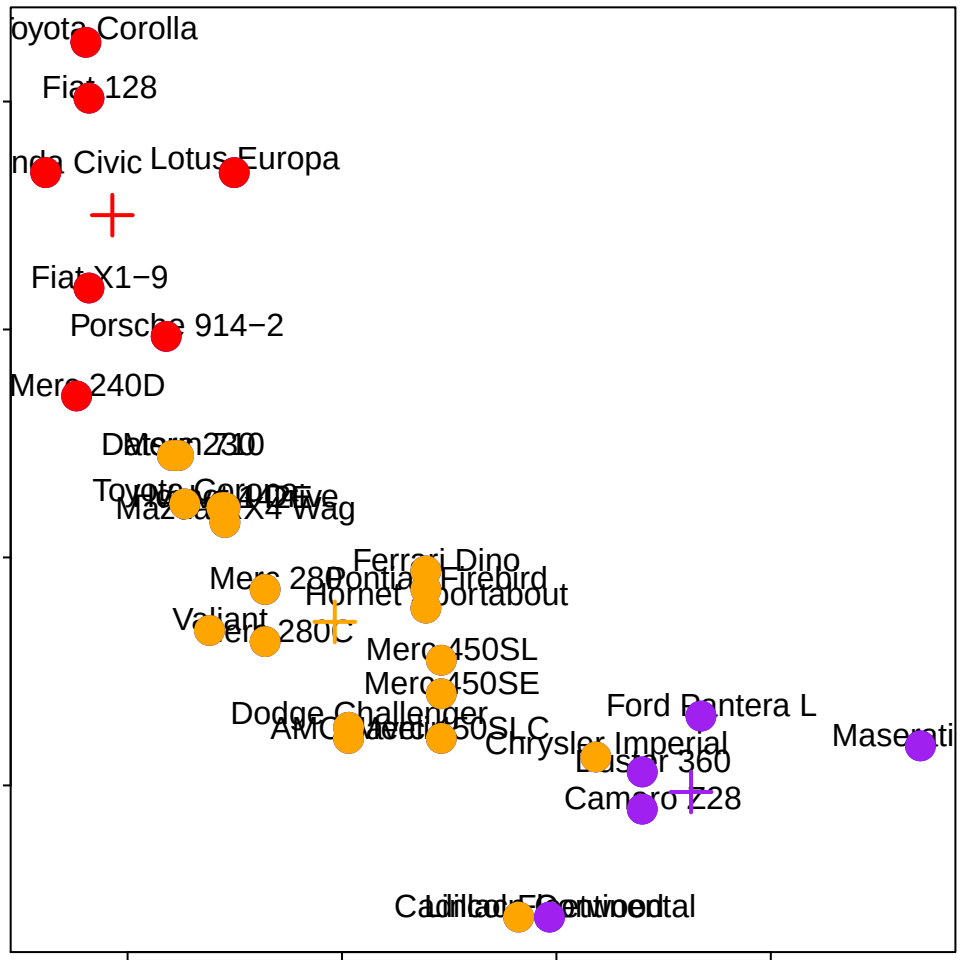
- –
  –
  – /
  –
- –
  – # of clusters
  –
- ’ ‘
  – ’ ‘

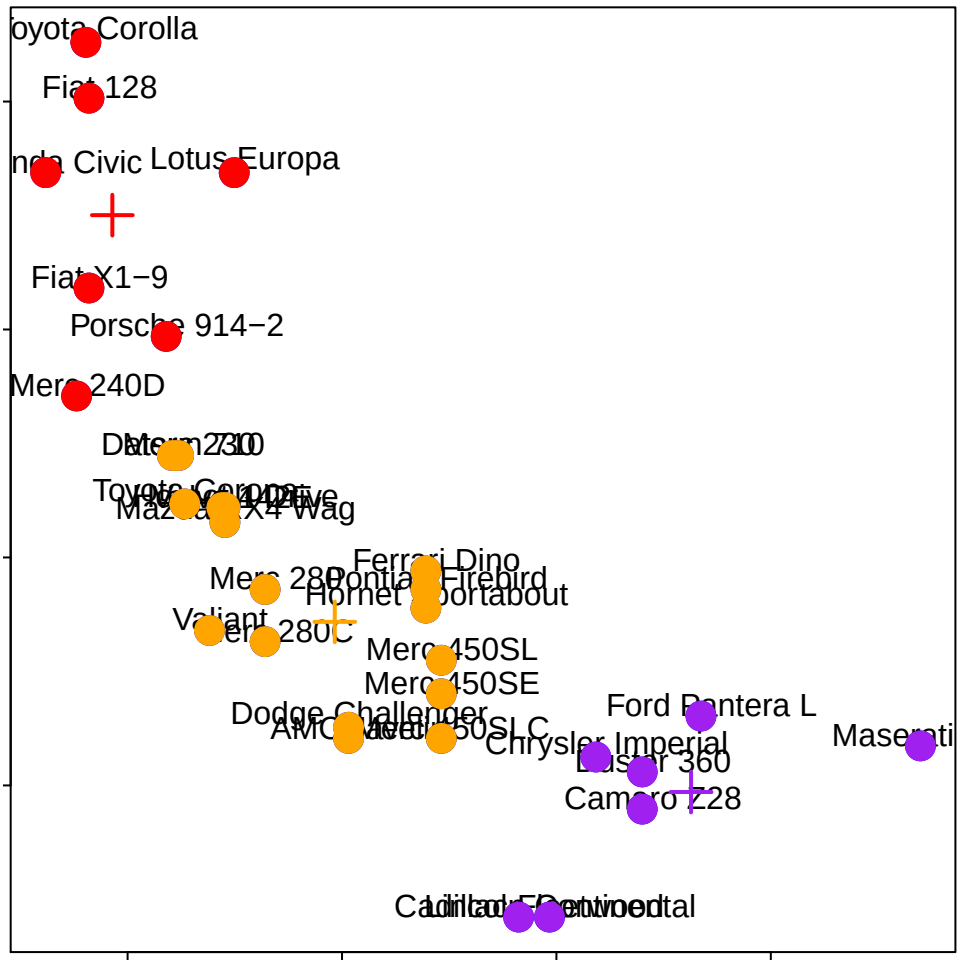
```
x<-scale(mtcars$hp[-1]);y<-scale(mtcars$mpg[-1])
plot(x,y,col="blue",pch=19,cex=2)
text(x+0.05,y+0.05,labels=labelCar)
```

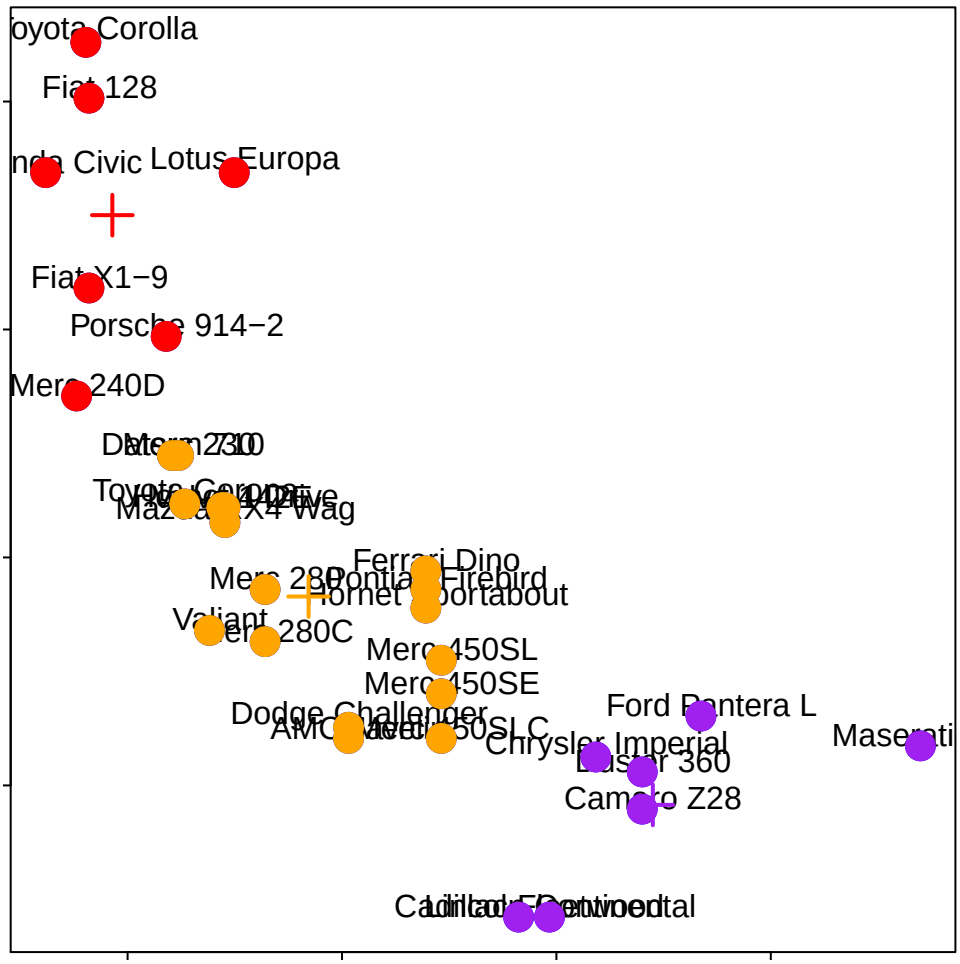



oyota Corolla
Fiat 128
nda Civic
Lotus Europa
Fiat X1−9
Porsche 914−2
Merc 240D
Mazda RX4 Wag
Ferrari Dino
Merc 280
Pontiac Firebird
Hornet Sportabout
Valiant
Merc 280C
Merc 450SL
Merc 450SE
Dodge Challenger
Ford Pantera L
Chrysler Imperial
Duster 360
Camaro Z28
Maserati


Toyota Corolla
Fiat 128
Lotus Europa
Fiat X1−9
Porsche 914−2
Merc 240D
Ferrari Dino
Ford Pantera L
Chrysler Imperial
Camaro Z28


oyota Corolla
Fiat 128
nda Civic
Lotus Europa
Fiat X1−9
Porsche 914−2
Merc 240D
Toyota Corona
Ferrari Dino
Hornet Sportabout
Valiant
Merc 450SL
Merc 450SE
Dodge Challenger
Ford Pantera L
Chrysler Imperial
Camaro Z28
Maserati


Fiat 128
Lotus Europa
Fiat X1−9
Porsche 914−2
Ferrari Dino
Firebird
Valiant
450SL
450SE
Ford Pantera L
Chrysler Imperial
Camaro Z28



`kmeans()`

- Important parameters: `x`, `centers`, `iter.max`, `nstart`

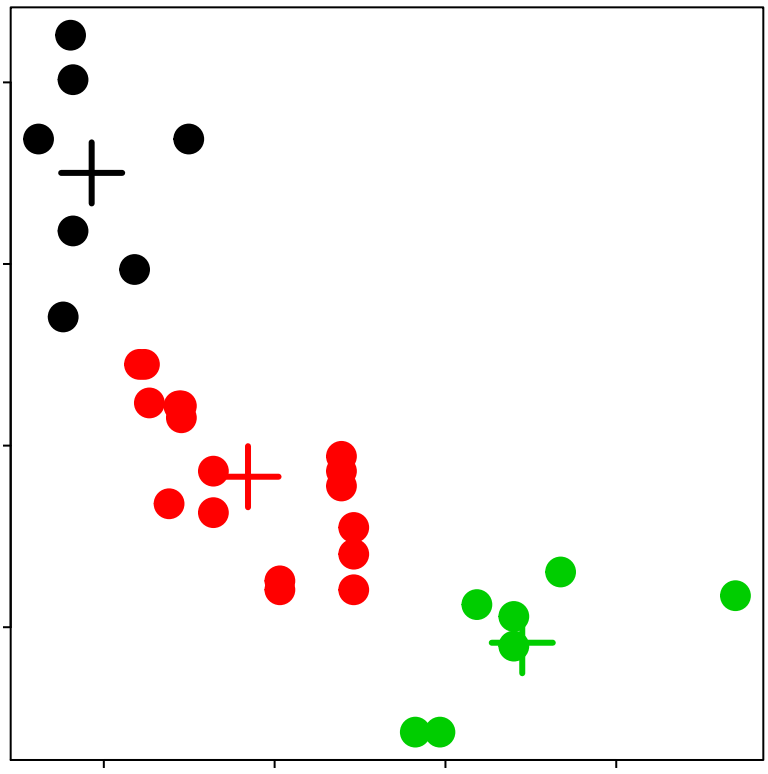
```
dataFrame <- data.frame(x,y)
kmeansObj <- kmeans(dataFrame,centers=3)
names(kmeansObj)
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
kmeansObj$cluster
```

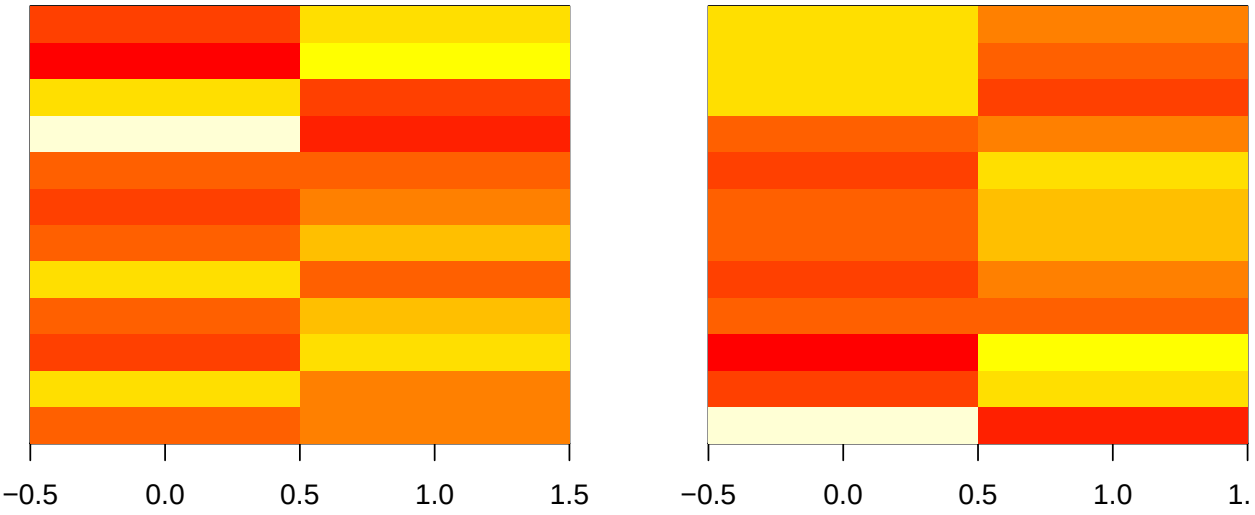
```
##  [1] 2 2 2 2 2 3 1 2 2 2 2 2 2 3 3 3 1 1 1 2 2 2 3 2 1 1 1 3 2 3 2
```

```
par(mar=rep(0.2,4))
plot(x,y,col=kmeansObj$cluster,pch=19,cex=2)
points(kmeansObj$centers,col=1:3,pch=3,cex=3,lwd=3)
```



Heatmaps

```
set.seed(1234)
dataMatrix <- as.matrix(dataFrame)[sample(1:12),]
kmeansObj <- kmeans(dataMatrix,centers=3)
par(mfrow=c(1,2), mar = c(2, 4, 0.1, 0.1))
image(t(dataMatrix)[,nrow(dataMatrix):1],yaxt="n")
image(t(dataMatrix)[,order(kmeansObj$cluster)],yaxt="n")
```

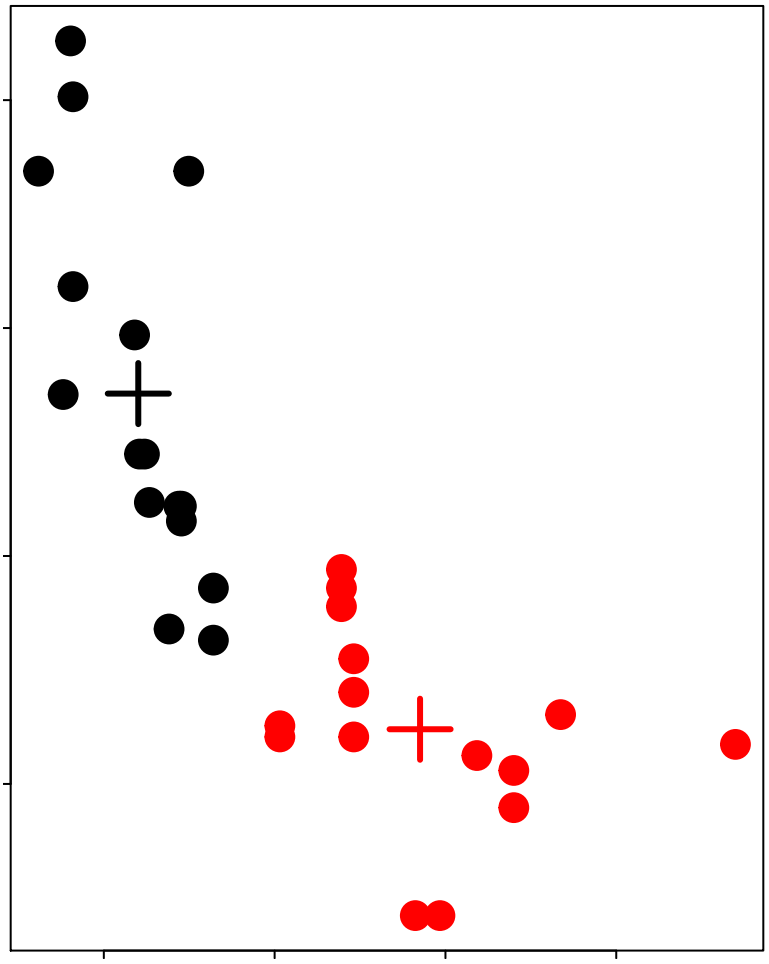


K-means

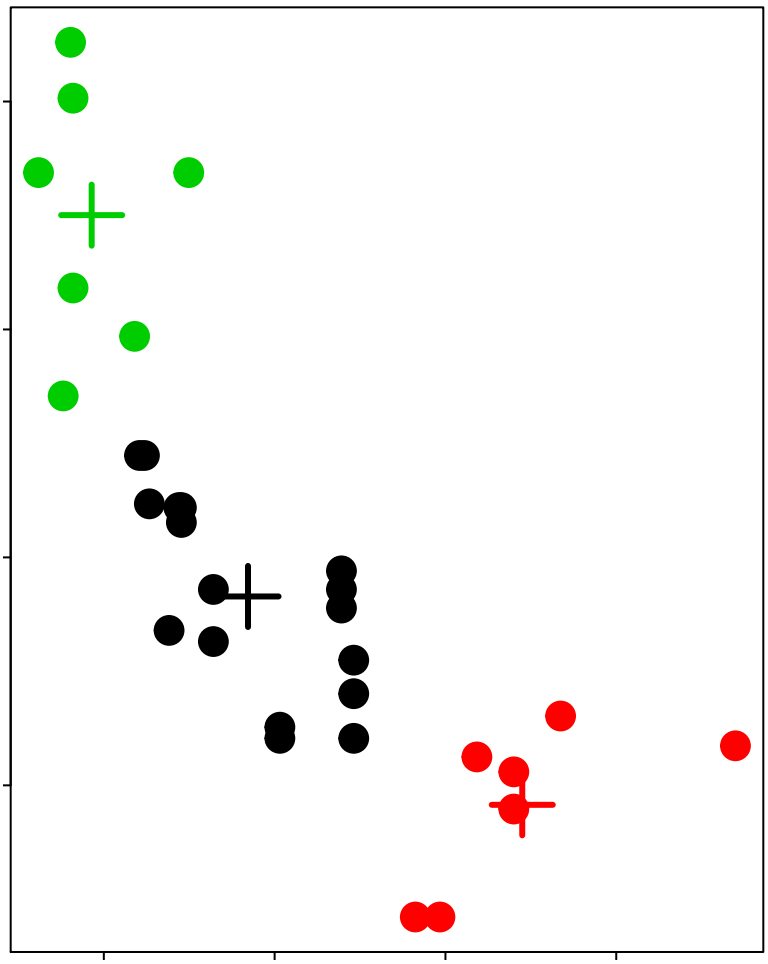
- # of clusters
  - / /
  - cross validation/information theory
  - Determining the number of clusters

- K-means
  - # of clusters
  - # of iterations

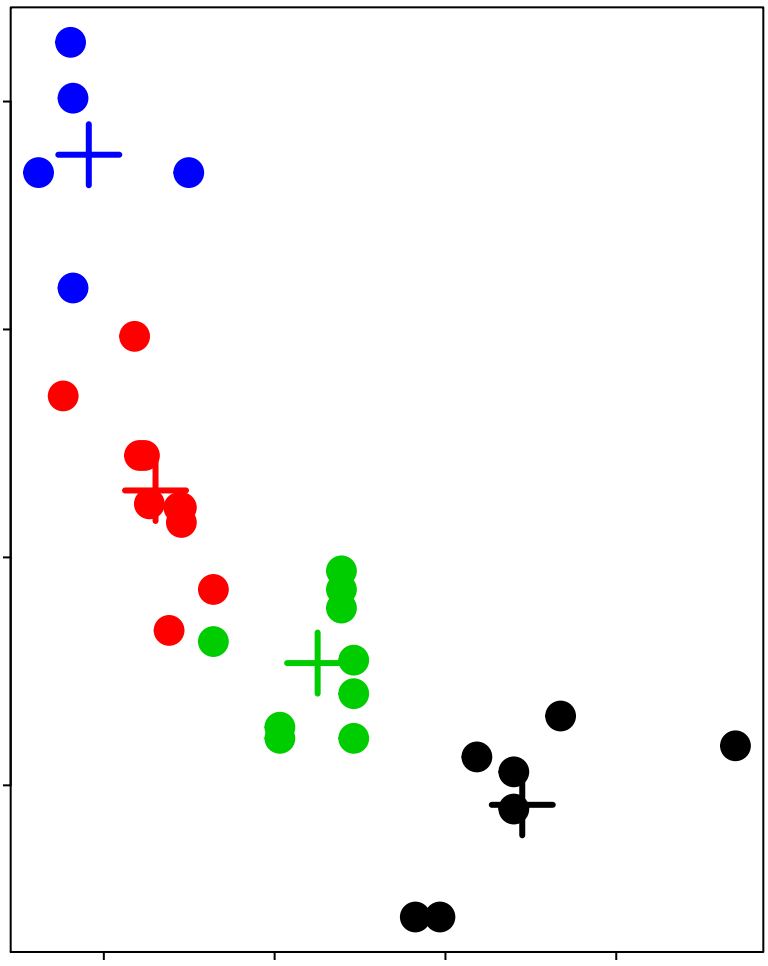
`kmeans()`, k=2



`kmeans()`, k=3



`kmeans()`, k=4

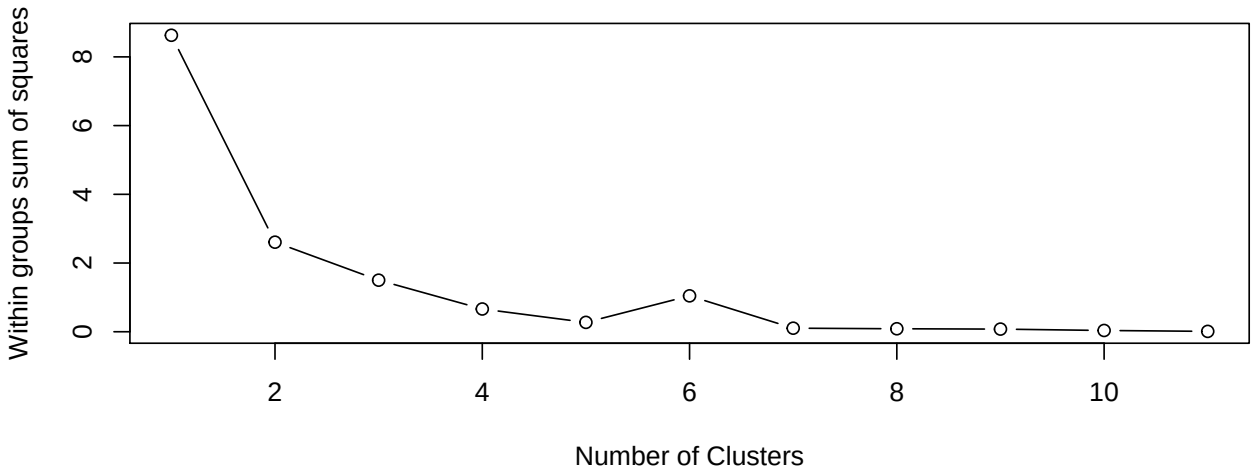


Use sum of squared error (SSE) scree plot to optimize the number of clusters

SSE: The sum of the squared distance between each member of a cluster and its cluster centroid.

SSE screen plot `withinss`

```
dataMatrix <- as.matrix(dataFrame)[sample(1:12),]
wss <- (nrow(dataMatrix)-1)*sum(apply(dataMatrix,2,var))
for (i in 2:(nrow(dataMatrix)-1)) {
    wss[i] <- sum(kmeans(dataMatrix,centers=i)$withinss)
}
par(mfrow=c(1,1), mar = c(4,4,1,1)) #, , ,
plot(1:(nrow(dataMatrix)-1), wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
```



10.3.2 Association Rules

- - - (Market Basket Analysis) - Apriori (Boolean association rules)

Apriori

```
# Load the libraries
if (!require('arules')){
    install.packages("arules");library(arules) #for Apriori
}
if (!require('datasets')){
    install.packages("datasets");library(datasets) #for Groceries data
}
data(Groceries) # Load the data set
Groceries@data@Dim #169   9835
```

```
## [1]  169 9835
```

apriori()

```
# Get the rules
rules <- apriori(Groceries, # data= Groceries
                 parameter = list(supp = 0.001, conf = 0.8), #
                 control = list(verbose=F)) #  output
options(digits=2) # Only 2 digits
inspect(rules[1:5]) # Show the top 5 rules
```

```
##     lhs                         rhs            support confidence lift
## [1] {liquor,red/blush wine} => {bottled beer} 0.0019  0.90       11.2
## [2] {curd,cereals}          => {whole milk}   0.0010  0.91        3.6
## [3] {yogurt,cereals}        => {whole milk}   0.0017  0.81        3.2
## [4] {butter,jam}            => {whole milk}   0.0010  0.83        3.3
## [5] {soups,bottled beer}    => {whole milk}   0.0011  0.92        3.6
```

=>

- Support:
- Confidence: A B

- Lift: /
  - lift=1: items on the left and right are independent.

```
rules<-sort(rules, by="confidence", decreasing=TRUE) # confidence
inspect(rules[1:5]) # Show the top 5 rules
```

```
##     lhs                     rhs          support confidence lift
## [1] {rice,
##      sugar}              => {whole milk}  0.0012          1  3.9
## [2] {canned fish,
##      hygiene articles}   => {whole milk}  0.0011          1  3.9
## [3] {root vegetables,
##      butter,
##      rice}               => {whole milk}  0.0010          1  3.9
## [4] {root vegetables,
##      whipped/sour cream,
##      flour}              => {whole milk}  0.0017          1  3.9
## [5] {butter,
##      soft cheese,
##      domestic eggs}      => {whole milk}  0.0010          1  3.9
```

```
rulesR<-apriori(data=Groceries, parameter=list(supp=0.001,conf = 0.08),
        appearance = list(default="lhs",rhs="whole milk"), #
        control = list(verbose=F)) #  output
rulesR<-sort(rulesR, decreasing=TRUE,by="confidence") # confidence
inspect(rulesR[1:5]) # Show the top 5 rules
```

```
##     lhs                     rhs          support confidence lift
## [1] {rice,
##      sugar}              => {whole milk}  0.0012          1  3.9
## [2] {canned fish,
##      hygiene articles}   => {whole milk}  0.0011          1  3.9
## [3] {root vegetables,
##      butter,
##      rice}               => {whole milk}  0.0010          1  3.9
## [4] {root vegetables,
##      whipped/sour cream,
##      flour}              => {whole milk}  0.0017          1  3.9
## [5] {butter,
##      soft cheese,
##      domestic eggs}      => {whole milk}  0.0010          1  3.9
```

```
rulesL<-apriori(data=Groceries, parameter=list(supp=0.001,conf = 0.15,minlen=2),
        appearance = list(default="rhs",lhs="whole milk"), #
        control = list(verbose=F)) #  output
rulesL<-sort(rulesL, decreasing=TRUE,by="confidence") # confidence
inspect(rulesL[1:5]) # Show the top 5 rules
```

```
##     lhs             rhs                support confidence lift
## [1] {whole milk} => {other vegetables} 0.075   0.29       1.5
## [2] {whole milk} => {rolls/buns}       0.057   0.22       1.2
## [3] {whole milk} => {yogurt}           0.056   0.22       1.6
```

```
## [4] {whole milk} => {root vegetables}  0.049   0.19       1.8
## [5] {whole milk} => {tropical fruit}   0.042   0.17       1.6
```

```
if (!require('arulesViz')){
    install.packages("arulesViz"); library(arulesViz)
}
#Mac->http://planspace.org/2013/01/17/fix-r-tcltk-dependency-problem-on-mac/
plot(rules,method="graph",interactive=TRUE,shading=NA) #
```

Chapter 11

Chapter 12

Chapter 13

Placeholder

Bibliography