Statistical analysis using R

R as data analysis software

- "R is a free **programming language** and **software environment** for statistical computing and graphics".
- So far we've focused on the "language" aspect (writing functions and packages).
- Many times you want to perform data analysis, and R provides powerful functionalities for that.
- Compared to "easier" data analysis software (e.g., MS Excel):
  - More powerful and flexible.
  - Needs some coding.

Data analysis functionalities

- Descriptive analysis: mean, median, var, sd, etc.
- Simple visualization: plot, hist, boxplot, barplot, etc.
- Statistical tests: t.test, var.test, prop.test, chisq.test,fisher.test, mantelhaen.test, ks.test, etc.
- Linear models: lm, glm, etc.
- Smoothing: smooth, lowess/loess, spline, etc.
- Clustering: kmeans, hclust.
- Discriminative analysis: lda, svm, randomForest, etc.

Statistical test functions

- R has functions for all major statistical test procedures.
- For continuous data:
  - t.test: one or two sample t-tests.
  - var.test: F test to compare the variances of two samples.
  - prop.test: test that proportions in several groups are the same.
  - binom.test: test the probability of success in a Bernoulli trial.
- For categorical data:
  - chi-squared contingency table tests and goodness-of-fit tests.
  - Fisher's exact test for testing independence of rows and columns in a contingency table with fixed marginals.
  - mantelhaen.test: Cochran-Mantel-Haenszel chi-squared test.
- Many others. If you need one, google.

Statistical data analysis

- As statisticians, we need to analyze a lot of data.
- There are different types of statistical data analyses:
  - With very specific aims, for example, test the efficacy of a drug. This is relatively easier. You first need to identify proper method(s), then apply to the data.
  - Without specific aim, for example, identify the differences between cancer and normal cells. This falls into the “data mining” category in machine learning field. There are infinite number of things to look at because there’s no boundary.

Typical data analysis steps

1. Exploratory analysis to get a “feeling” of the data: dimension, variable, distribution, outliers, pairwise relationship, etc.
2. Select proper method(s) to analyze the data. There could be many methods serving the same purpose. As long as your method is justifiable, you’ll be fine.
3. Check whether data satisfy the assumptions of the method. Transform the data if needed.
4. Analyze the data and generate a report.
5. Discuss with investigators. Probably need to change method and repeat steps 2-4.

Statistical data analysis – a case study

- We will use the “mtcars” data to demonstrate the data analysis capabilities of R.
- The “mtcars” data frame is distributed with R. It contains 11 specs of 32 different cars.

```
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1

> dim(mtcars)
[1] 32 11
```

Possible questions to ask

- Do cars with automatic and manual transmission have different fuel consumption?
- Do cars with different number of cylinders have different fuel consumption?
- What are the important factors for fuel consumption?
- …

Get to know your data

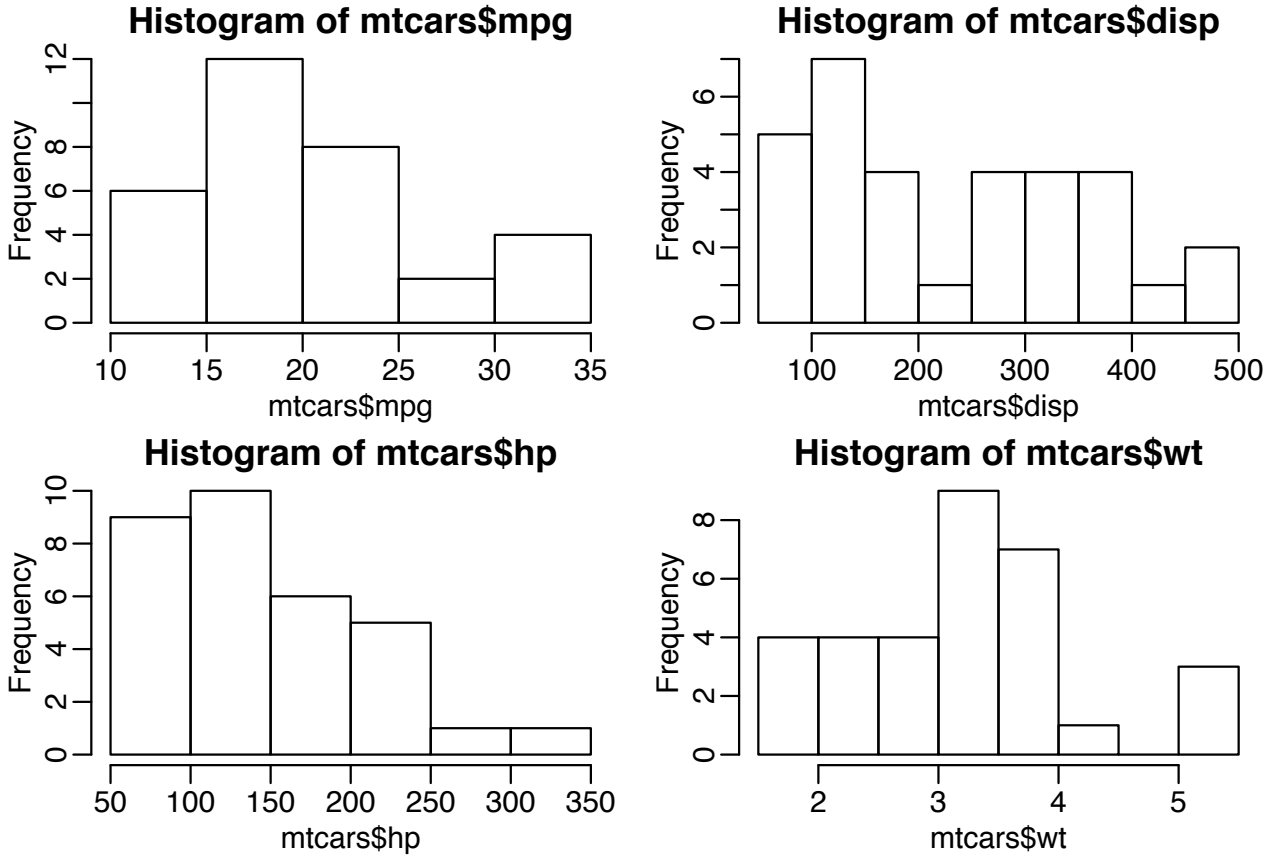
- We first want to look at the distribution of each variable.
  - For categorical data, use “table”.
  - For continuous data, use “hist” to look at histogram.

```
> table(mtcars$cyl)
 4  6  8
11 7 14

> table(mtcars$am)
 0  1
19 13

> table(mtcars$gear)
 3  4  5
15 12  5
```

```
> hist(mtcars$mpg)
> hist(mtcars$disp)
> hist(mtcars$hp)
> hist(mtcars$wt)
```



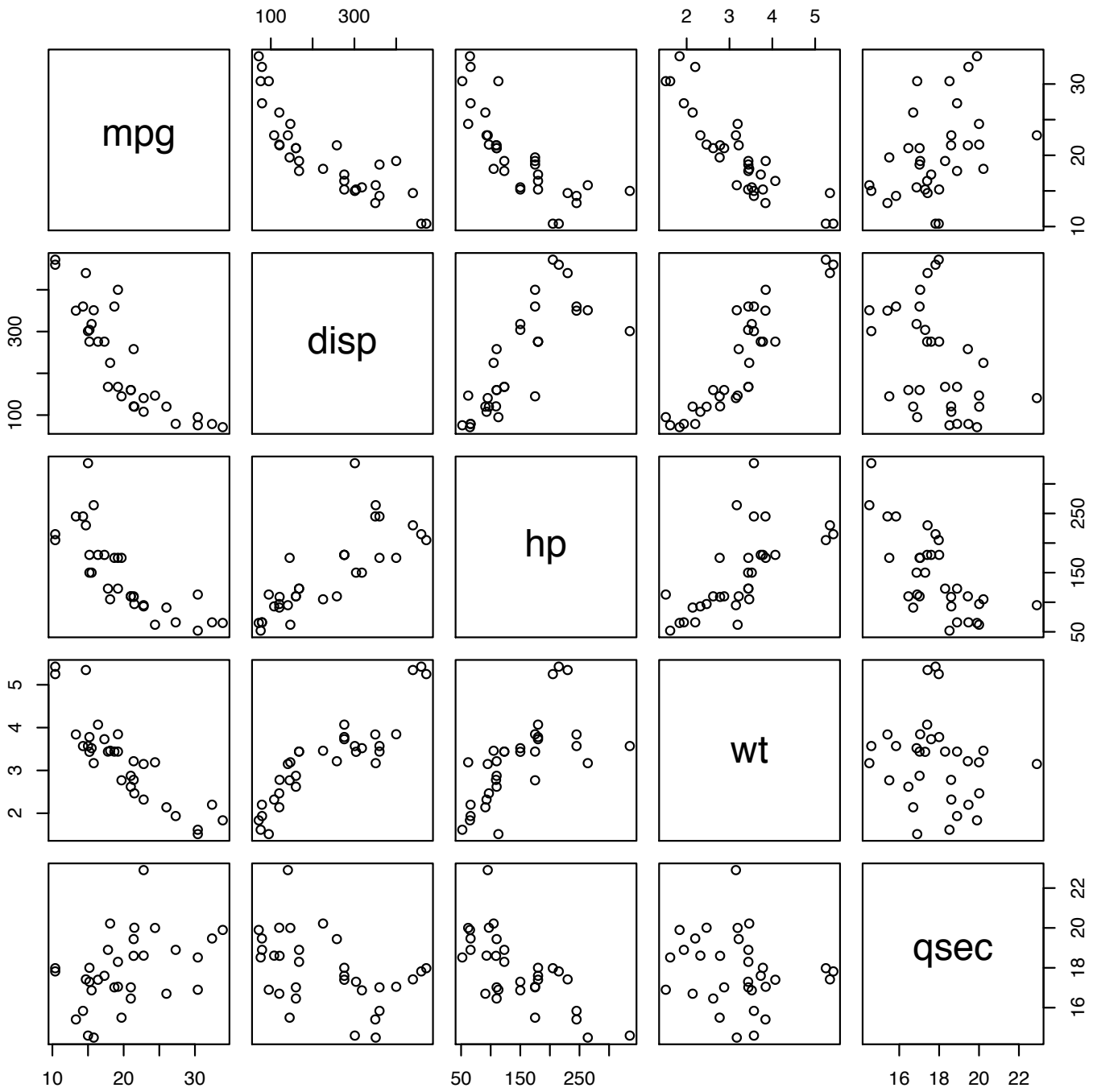
Get to know your data (cont.)

- Next, we want to look at the correlations of different variable.
  - For categorical data, use “table” function again.

```
> table(mtcars$am, mtcars$cyl)
     4  6  8
  0  3  4 12
  1  8  3  2
> table(mtcars$am, mtcars$gear)
     3  4  5
  0 15  4  0
  1  0  8  5
```

– For continuous data, use “cor” to compute correlation, and use “pairs” function to do pairwise scatter plot.

```
> round(cor(mtcars[,c("mpg", "disp", "hp", "drat", "wt", "qsec")]), 4)
         mpg    disp      hp    drat      wt    qsec
mpg   1.0000 -0.8476 -0.7762  0.6812 -0.8677  0.4187
disp -0.8476  1.0000  0.7909 -0.7102  0.8880 -0.4337
hp   -0.7762  0.7909  1.0000 -0.4488  0.6587 -0.7082
drat  0.6812 -0.7102 -0.4488  1.0000 -0.7124  0.0912
wt   -0.8677  0.8880  0.6587 -0.7124  1.0000 -0.1747
qsec  0.4187 -0.4337 -0.7082  0.0912 -0.1747  1.0000
```


mpg
disp
hp
wt
qsec
100
300
2
3
4
5
10
20
30
100
300
50
150
250
2
3
4
5
16
18
20
22
10
20
30
50
150
250
16
18
20
22

Do cars with automatic and manual transmission have different fuel consumption?

- There are two types of transmissions, so a two sample t-test using “t.test” function is adequate.

```
> t.test(mtcars$mpg ~ mtcars$am)

    Welch Two Sample t-test

data:  mtcars$mpg by mtcars$am
t = -3.7671, df = 18.332, p-value = 0.001374
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -11.280194  -3.209684
sample estimates:
mean in group 0 mean in group 1
       17.14737        24.39231
```

Results from `t.test`

- Result of t.test is a list containing a lot of information including test statistics, p-values, confident interval, etc.

```
> res = t.test(mtcars$mpg ~ mtcars$am)

> names(res)
[1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
[6] "null.value"  "alternative" "method"      "data.name"

> res$statistic
        t
-3.767123

> res$p.value
[1] 0.001373638
```

Do cars with different number of cylinders have different fuel consumption?

- There are three levels, so a two group t-test doesn't work. We need a ANOVA F-test here, using "aov" function.

```
> res = aov(mtcars$mpg ~ mtcars$cyl)
> res
Call:
   aov(formula = mtcars$mpg ~ mtcars$cyl)
Terms:
                mtcars$cyl Residuals
Sum of Squares    817.7130  308.3342
Deg. of Freedom          1        30

Residual standard error: 3.205902
Estimated effects may be unbalanced

> summary(res)
             Df Sum Sq Mean Sq F value   Pr(>F)
mtcars$cyl    1  817.7   817.7   79.56 6.11e-10 ***
Residuals    30  308.3    10.3
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What are the important factors for fuel consumption

- To answer this question, we can use
  - mpg as response.
  - others variables as predictors, individually.
- Assume a linear relationship between output and predictor, we perform simple linear regression analysis, using “lm” function:
  - Obtain a vector of p-values, each for a predictor.

lm example: mpg vs. wt

```
> fit = lm(mtcars$mpg ~ mtcars$wt)
> summary(fit)

Call:
lm(formula = mtcars$mpg ~ mtcars$wt)

Residuals:
    Min      1Q  Median      3Q     Max
-4.5432 -2.3647 -0.1252  1.4096  6.8727

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
mtcars$wt    -5.3445     0.5591  -9.559 1.29e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared: 0.7528, Adjusted R-squared: 0.7446
F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

Result p-values

```
> pvals
         cyl          disp            hp          drat            wt          qsec
6.112687e-10 9.380327e-10 1.787835e-07 1.776240e-05 1.293959e-10 1.708199e-02
          vs            am          gear          carb
3.415937e-05 2.850207e-04 5.400948e-03 1.084446e-03
```

- It shows that most of the variables are statistically significant (with $p<0.05$).

Build a joint model for fuel consumption

- The goal is to build a joint model (with multiple predictors) for mpg.
- This is a model building problem, an easy way is
  - Do simple linear regression and obtain p-values for each predictor.
  - Add predictors to the joint model one by one, start from the most significant one.
  - Come up with a final multivariate model, where all predictors are significant.
  - Use cross validation to assess the predictive power.

- For example, using wt, cyl, and disp as predictors, we found that disp is **not** significant (although disp is significant in univariate analysis). So we need to exclude it in the final model.

```
> fit = lm(mtcars$mpg ~ mtcars$wt + mtcars$cyl + mtcars$disp)
> summary(fit)

Call:
lm(formula = mtcars$mpg ~ mtcars$wt + mtcars$cyl + mtcars$disp)

Residuals:
    Min      1Q  Median      3Q     Max
-4.4035 -1.4028 -0.4955  1.3387  6.0722

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 41.107678   2.842426  14.462 1.62e-14 ***
mtcars$wt   -3.635677   1.040138  -3.495  0.00160 **
mtcars$cyl  -1.784944   0.607110  -2.940  0.00651 **
mtcars$disp  0.007473   0.011845   0.631  0.53322
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.595 on 28 degrees of freedom
Multiple R-squared: 0.8326,   Adjusted R-squared: 0.8147
F-statistic: 46.42 on 3 and 28 DF,  p-value: 5.399e-11
```

Final model

- After the process, we found that only wt and cyl are significant predictors for mpg.

```
> fit = lm(mtcars$mpg ~ mtcars$wt + mtcars$cyl)
> summary(fit)

Call:
lm(formula = mtcars$mpg ~ mtcars$wt + mtcars$cyl)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2893 -1.5512 -0.4684  1.5743  6.1004

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  39.6863     1.7150  23.141  < 2e-16 ***
mtcars$wt    -3.1910     0.7569  -4.216 0.000222 ***
mtcars$cyl   -1.5078     0.4147  -3.636 0.001064 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.568 on 29 degrees of freedom
Multiple R-squared: 0.8302,   Adjusted R-squared: 0.8185
F-statistic: 70.91 on 2 and 29 DF,  p-value: 6.809e-12
```

A little more about the "lm" function

- Class "lm", containing information like estimated coefficients, residuals, fitted values, etc.

```
> class(fit)
[1] "lm"
> names(fit)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
```

- Fields can be accessed by some generic functions:
  - coef: extract the estimated coefficients.
  - resid: extract the model fitted residuals.
  - fitted: extract the fitted values.

The estimates

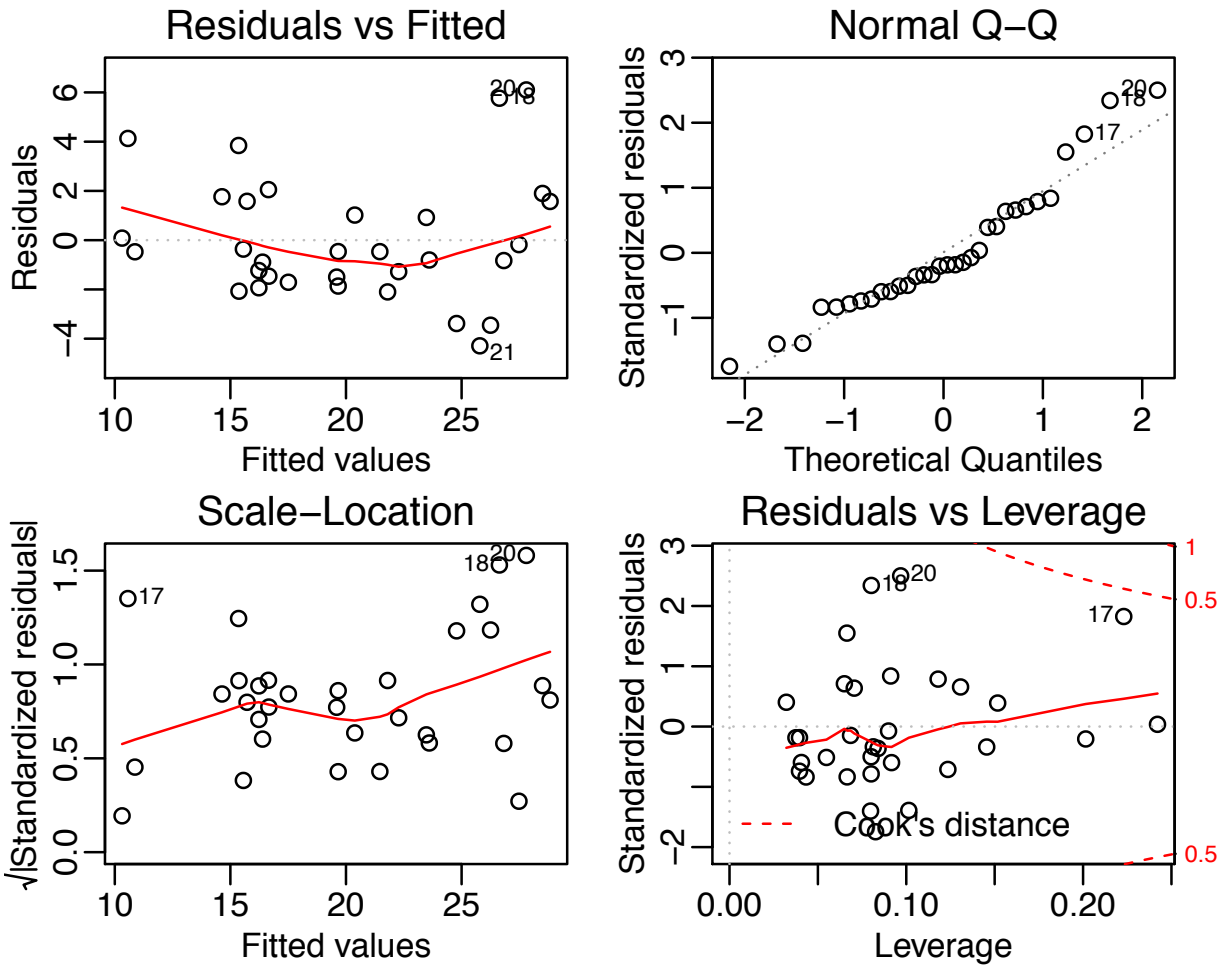
- The estimated parameters and their confidence intervals can be obtained by `coef` and `confint` functions.

```
> coef(fit)
(Intercept)   mtcars$wt  mtcars$cyl
  39.686261   -3.190972   -1.507795

> confint(fit)
                2.5 %     97.5 %
(Intercept) 36.178725 43.1937976
mtcars$wt   -4.739020 -1.6429245
mtcars$cyl  -2.355928 -0.6596622
```

Some diagnostic plot for the fitting

```
plot(fit)
```

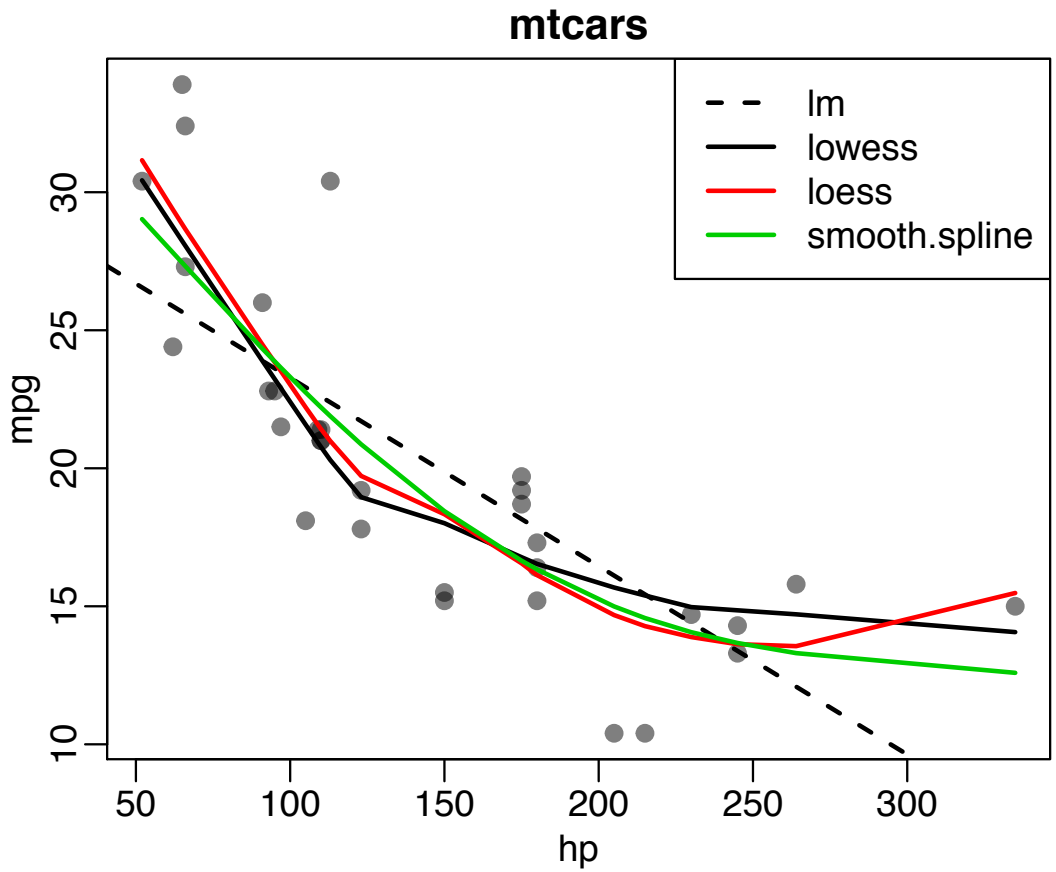


Building non-linear model

- Based on the scatter plot, mpg and hp seems to have non-linear relationship. In this case building a non-linear model is useful.
- R has a series of powerful functions for that, including:
  - lowess: locally-weighted polynomial regression.
  - loess: Local Polynomial Regression Fitting.
  - smooth.spline: cubic smoothing spline.
  - smoothScatter: smoothed color density representation of the 2D scatterplot.
- The statistical methods and algorithms of these functions are beyond the scope of this class. I will only provide a few examples.

Non-linear model

```
fit.lowess=lowess(mtcars$hp, mtcars$mpg)
fit.loess=loess(mpg~hp, data=mtcars)
fit.spline=smooth.spline(mtcars$hp, mtcars$mpg,df=3)
```



Test of dependence of categorical variables

- Another question to ask is “does number of cylinders correlate with transmission type?”
- Both number of cylinders and transmission type are categorical variables, so the test need to be based on “contingency table”.
- To make the table, use the “table” function:

```
> tbl = table(mtcars$am, mtcars$cyl)
> tbl
    4  6  8
  0  3  4 12
  1  8  3  2
```

Dependence test on contingency table

- Can be done by chi-square test using “chisq.test” function:

```
> chisq.test(tbl)

    Pearson's Chi-squared test

data:  tbl
X-squared = 8.7407, df = 2, p-value = 0.01265

Warning message:
In chisq.test(tbl) : Chi-squared approximation may be incorrect
```

- As rule of thumb, if any cell of the table is less than 5, Fisher's exact test should be used, with the "fisher.test" function:

```
> fisher.test(tbl)

    Fisher's Exact Test for Count Data

data:  tbl
p-value = 0.009105
alternative hypothesis: two.sided
```

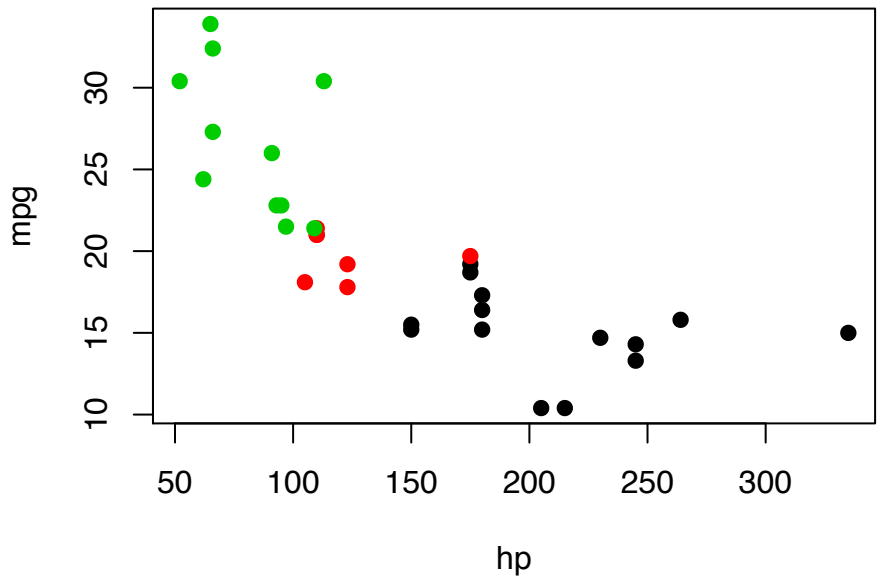
- Now we know transmission and number of cylinders are indeed correlated.

Clustering

- Clustering is a powerful tool to group the data.
- There are mainly two clustering algorithms provided in R:
  - K-means clustering: artition data into several groups.
  - Hierarchical clustering: put data into a binary tree. The tree is constructed based on pairwise distances (or similarities) of the observation.

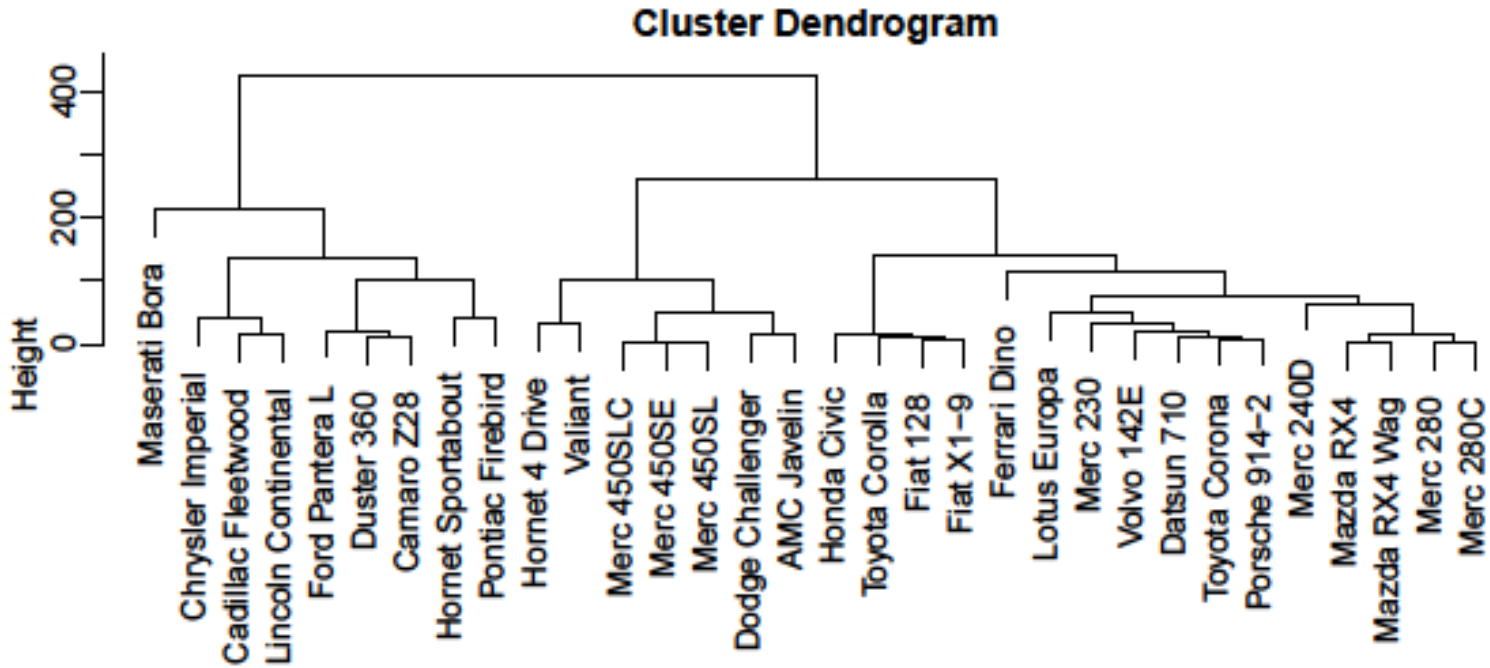
K-means clustering

```
> res.kmeans=kmeans(mtcars, 3)
> names(res.kmeans)
[1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
[6] "betweenss"    "size"
```



Hierarchical clustering

```
> dd=dist(mtcars)
> hc=hclust(dd)
```



Tips for being a good data analyst

- Be very familiar with the data: “look” at your data from different angles.
- Understand the motivations:
  - Communicate with scientists.
  - Have some common sense and think as a normal person.
- When choose a model/method, understand the fundamental assumptions. Transform the data if some are violated.
- Keep the codes tidy and well commented/documented. Make sure all results/figures can be reproduced.
- Explain the results in plain language.