Lab 1:
Introduction to R

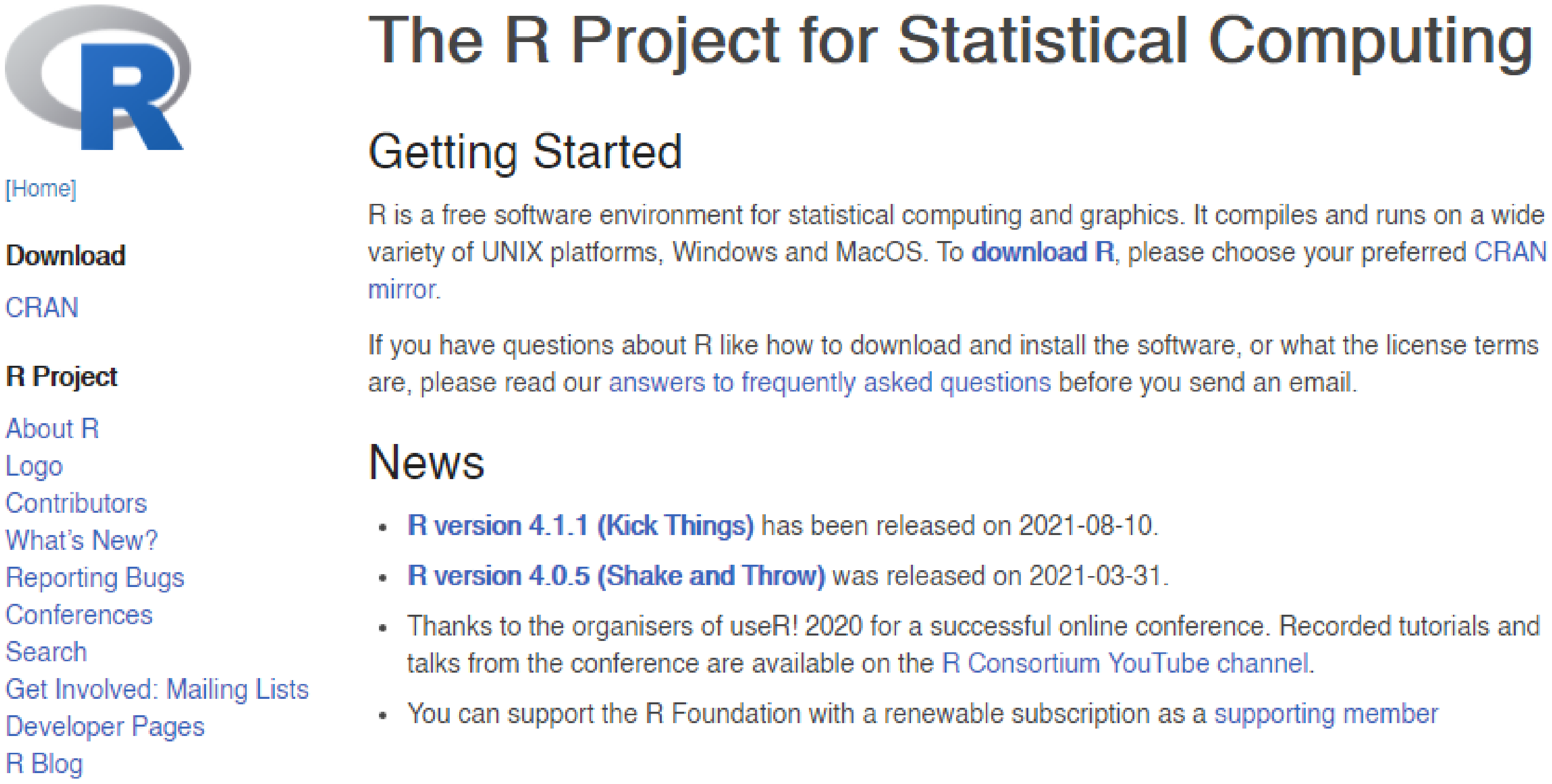
Outline

- Install R and data packages
- Basic commands
- Example of preliminary analysis of a dataset

Install R and data packages

Install R

- https://www.r-project.org/



[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Reporting Bugs
Conferences
Search
Get Involved: Mailing Lists
Developer Pages
R Blog

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

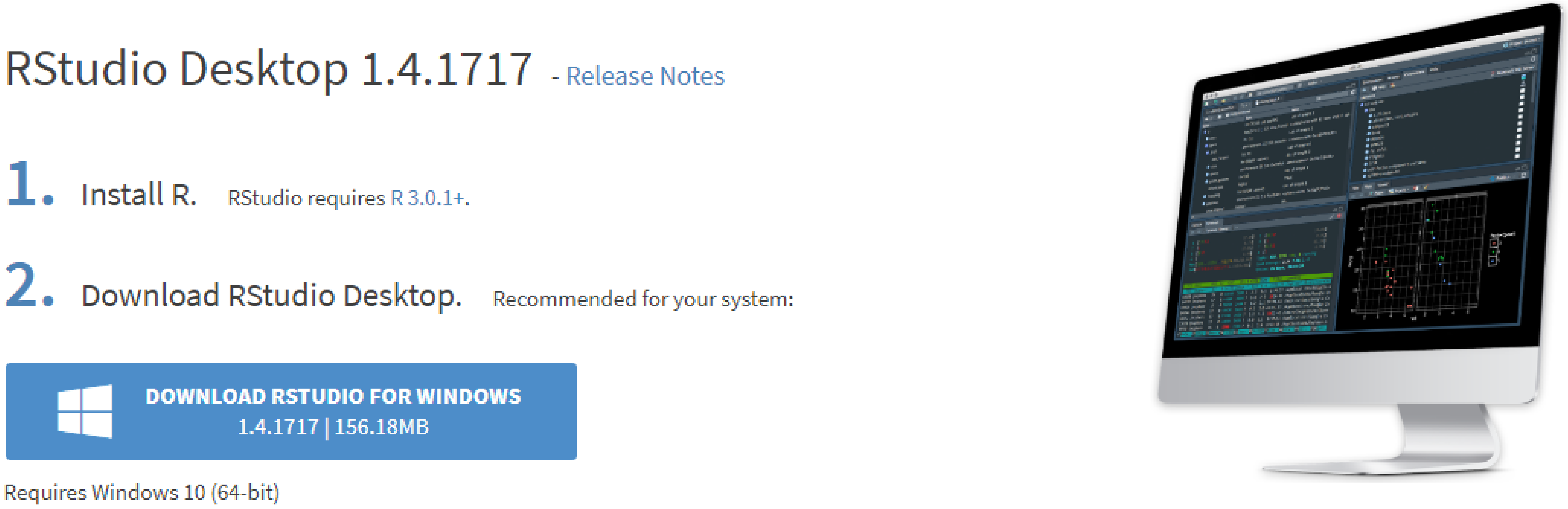
If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- **R version 4.1.1 (Kick Things)** has been released on 2021-08-10.
- **R version 4.0.5 (Shake and Throw)** was released on 2021-03-31.
- Thanks to the organisers of useR! 2020 for a successful online conference. Recorded tutorials and talks from the conference are available on the R Consortium YouTube channel.
- You can support the R Foundation with a renewable subscription as a supporting member

Install Rstudio Desktop （Optional）

- https://www.rstudio.com/products/rstudio/download/



RStudio Desktop 1.4.1717 - Release Notes

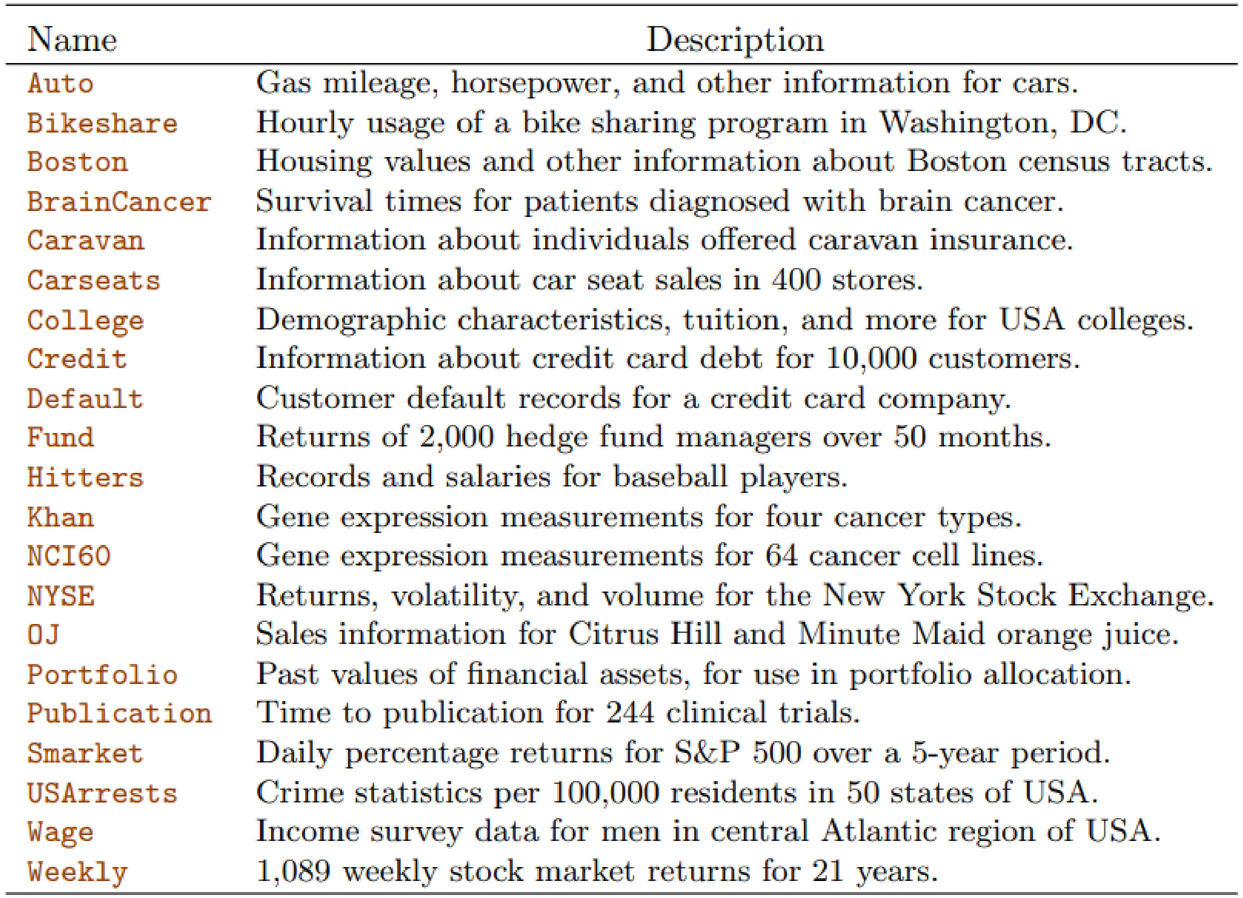
1. Install R. RStudio requires R 3.0.1+.

2. Download RStudio Desktop. Recommended for your system:

Requires Windows 10 (64-bit)

Data Sets Used in Labs and Exercises

- ISLR2:
  - Data for an Introduction to Statistical Learning with Applications in R
  - https://cran.r-project.org/package=ISLR2
- The MASS library
  - Functions and datasets to support Venables and Ripley's MASS
  - https://cran.r-project.org/package=MASS
- Base R



Name	Description
`Auto`	Gas mileage, horsepower, and other information for cars.
`Bikeshare`	Hourly usage of a bike sharing program in Washington, DC.
`Boston`	Housing values and other information about Boston census tracts.
`BrainCancer`	Survival times for patients diagnosed with brain cancer.
`Caravan`	Information about individuals offered caravan insurance.
`Carseats`	Information about car seat sales in 400 stores.
`College`	Demographic characteristics, tuition, and more for USA colleges.
`Credit`	Information about credit card debt for 10,000 customers.
`Default`	Customer default records for a credit card company.
`Fund`	Returns of 2,000 hedge fund managers over 50 months.
`Hitters`	Records and salaries for baseball players.
`Khan`	Gene expression measurements for four cancer types.
`NCI60`	Gene expression measurements for 64 cancer cell lines.
`NYSE`	Returns, volatility, and volume for the New York Stock Exchange.
`OJ`	Sales information for Citrus Hill and Minute Maid orange juice.
`Portfolio`	Past values of financial assets, for use in portfolio allocation.
`Publication`	Time to publication for 244 clinical trials.
`Smarket`	Daily percentage returns for S&P 500 over a 5-year period.
`USArrests`	Crime statistics per 100,000 residents in 50 states of USA.
`Wage`	Income survey data for men in central Atlantic region of USA.
`Weekly`	1,089 weekly stock market returns for 21 years.

Install ISLR2 Package

- Manual download with R
  - Click Packages & Data → Package Installer → input the package name → select a mirror site
- Manual download with Rstudio
  - Click Tools → Install package → install from CRAN/local archive file
- By R command line
  - install.packages("ISLR2")

Basic commands

Vector

- Insert vector using function c()
- Check length of vector using length()

```
> x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
> x
[1] 1 2 3 4 5 6 7 8 9
> length(x)
[1] 9
```

Matrix

- Declare a matrix using function matrix()
- Use byrow =TRUE/FALSE to specify order
- Use dim() to find dimension of a matrix

Matrix

```
> x <- matrix (data = c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3)
> x
    [,1] [,2] [,3]
[1,]   1    3    5
[2,]   2    4    6
> x <- matrix (data = c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3, byrow = TRUE)
> x
    [,1] [,2] [,3]
[1,]   1    2    3
[2,]   4    5    6
> x <- matrix (data = c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3, byrow = FALSE)
> x
    [,1] [,2] [,3]
[1,]   1    3    5
[2,]   2    4    6
> dim(x)
[1] 2 3
```

Select Elements in A Matrix

```
> A <- matrix(1:16, 4, 4)
> A
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
> A[2, 3]
[1] 10
> A[c(1, 3), c(2, 4)]
     [,1] [,2]
[1,]    5   13
[2,]    7   15
> A[1:2,]
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
```

```
> A[, 1]
[1] 1 2 3 4
> A[, 1:2]
     [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
> A[-1,]
     [,1] [,2] [,3] [,4]
[1,]    2    6   10   14
[2,]    3    7   11   15
[3,]    4    8   12   16
> A[-c(1, 2),]
     [,1] [,2] [,3] [,4]
[1,]    3    7   11   15
[2,]    4    8   12   16
```

Generate Random Numbers

- Generate random numbers from a standard normaldistribution using rnorm (n)

```
> y <- rnorm(20)
> y
 [1] -1.09460899  0.22386861 -0.20583813 -1.11530919  0.58994271 -2.06441523  1.39271334
 [8] -0.23312401 -0.10541311 -0.63185659  0.25970922  0.43194340 -0.04194421  0.09849715
[15] -0.50593705  1.15531491 -1.14990503 -0.26525399  0.87302274  0.12407061
```

- Calculate mean(), var(), sd() of random numbers

```
> mean(y)
[1] -0.1132261
> var(y)
[1] 0.7009881
> sd(y)
[1] 0.8372503
```

Set the Seed of Random Number Generator

- Set the seed of random number generator using set.seed
- To reproduce the exact same set of random numbers, use the same seed

```
> set.seed(1)
> rnorm(5)
[1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078
> rnorm(5)
[1] -0.8204684  0.4874291  0.7383247  0.5757814 -0.3053884
> set.seed(1)
> rnorm(5)
[1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078
```

Example of preliminary analysis of a dataset

Load Dataset

- To load a data set in the ISLR2 package or other packages/libraries, you only need to load the package

```
> library (ISLR2)
```

- To load an external data set, first specify the directory, Misc → Change Working Directory
  - If the data are saved as a text file

  ```
  > Auto <- read.table("Auto.data", header=T, na.strings="?", stringsAsFactors=T)
  ```

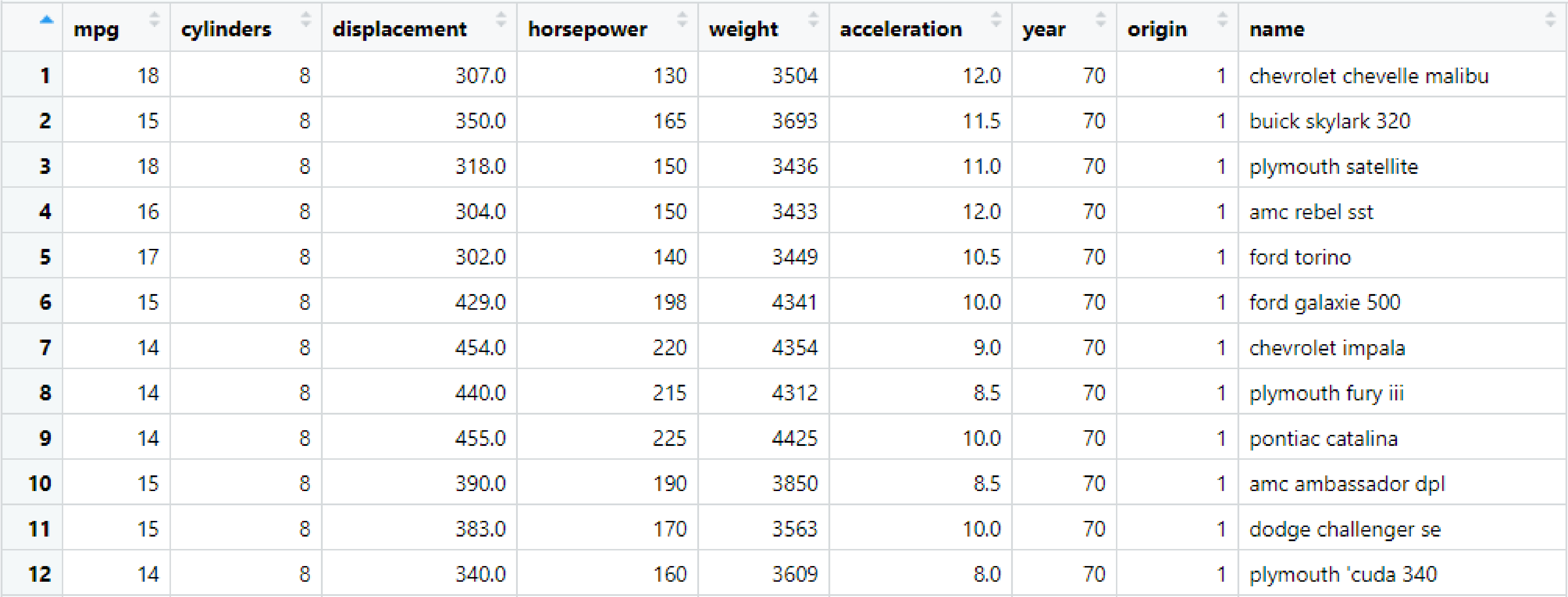
  - If the data are saved as a csv file (Excel)

  ```
  > College <- read.csv("College.csv", na.strings="?", stringsAsFactors=T)
  ```

- Try loading external data files using datasets available on the textbook website https://www.statlearning.com/resources-second-edition

View Data

Then, check a dataset by typing its name in console or view()



	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
1	18	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
2	15	8	350.0	165	3693	11.5	70	1	buick skylark 320
3	18	8	318.0	150	3436	11.0	70	1	plymouth satellite
4	16	8	304.0	150	3433	12.0	70	1	amc rebel sst
5	17	8	302.0	140	3449	10.5	70	1	ford torino
6	15	8	429.0	198	4341	10.0	70	1	ford galaxie 500
7	14	8	454.0	220	4354	9.0	70	1	chevrolet impala
8	14	8	440.0	215	4312	8.5	70	1	plymouth fury iii
9	14	8	455.0	225	4425	10.0	70	1	pontiac catalina
10	15	8	390.0	190	3850	8.5	70	1	amc ambassador dpl
11	15	8	383.0	170	3563	10.0	70	1	dodge challenger se
12	14	8	340.0	160	3609	8.0	70	1	plymouth 'cuda 340

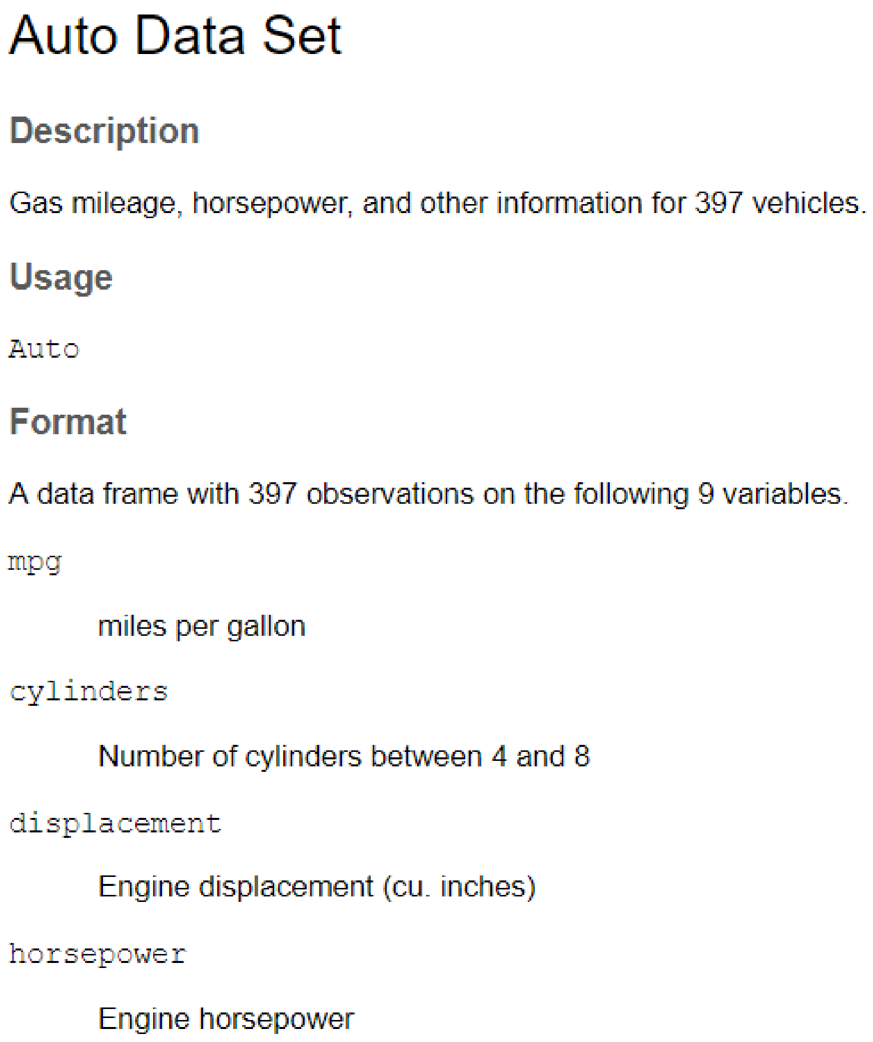
Dimension & Variables

- Type names(datasetname), e.g., names(Auto), to list all attributes (column names) of the table

```
> names(Auto)
[1] "mpg"          "cylinders"    "displacement" "horsepower"  
[5] "weight"       "acceleration" "year"         "origin"      
[9] "name"        
> dim(Auto)
[1] 397   9
> Auto <- na.omit(Auto)
> dim(Auto)
[1] 392   9
```

Background

- To gather more information about the data set, type ?datasetname (e.g., ?Auto, or help(Auto))



Auto Data Set

Description

Gas mileage, horsepower, and other information for 397 vehicles.

Usage

```
Auto
```

Format

A data frame with 397 observations on the following 9 variables.

`mpg`

miles per gallon

`cylinders`

Number of cylinders between 4 and 8

`displacement`

Engine displacement (cu. inches)

`horsepower`

Engine horsepower

Access Variables in the Dataset

- Method 1: giving name of the variable and the dataset

```
> Auto$mpg
  [1] 18.0 15.0 18.0 16.0 17.0 15.0 14.0
  [8] 14.0 14.0 15.0 15.0 14.0 15.0 14.0
 [15] 24.0 22.0 18.0 21.0 27.0 26.0 25.0
```

- Method 2: first attach the dataset to the R search path, then all variables in the dataset can be accessed by simply giving their names

```
> attach(Auto)
> mpg
  [1] 18.0 15.0 18.0 16.0 17.0 15.0 14.0 14.0
  [9] 14.0 15.0 15.0 14.0 15.0 14.0 24.0 22.0
 [17] 18.0 21.0 27.0 26.0 25.0 24.0 25.0 26.0
```

Numerical Summaries

- Numerical summary of dataset or variable summary(dataset) or summary(dataset$colname)

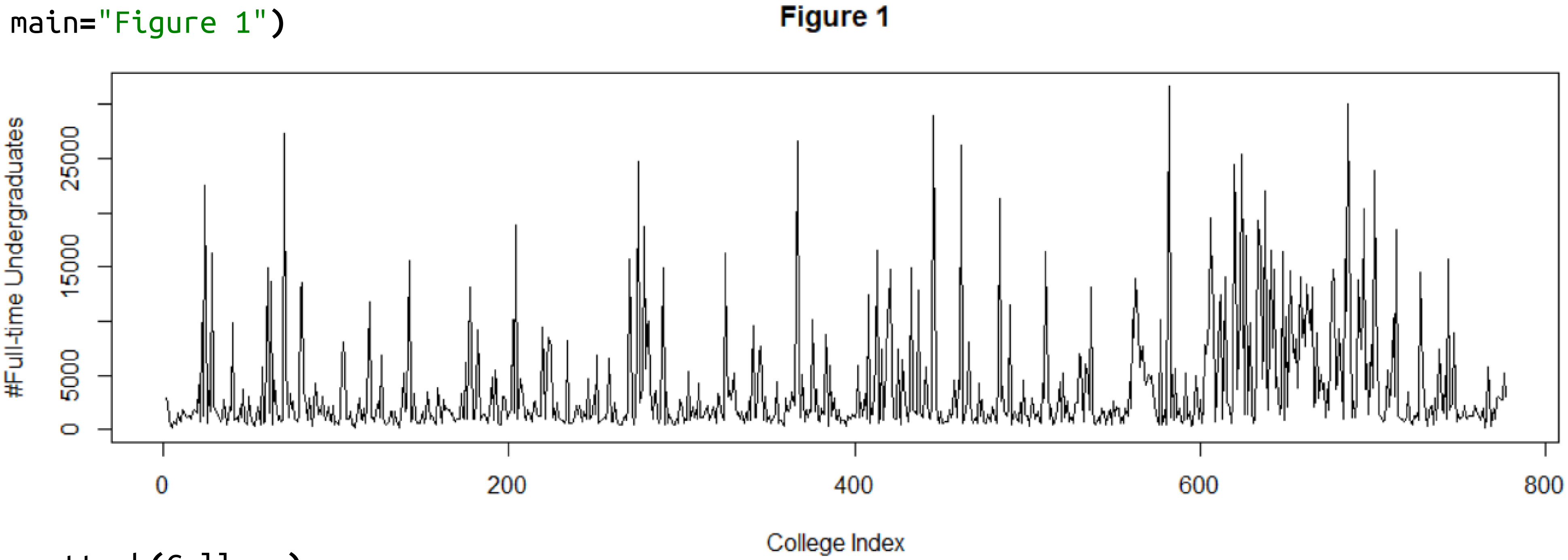
```
> summary(Auto)
      mpg          cylinders      displacement     horsepower        weight
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140

  acceleration        year           origin                      name
 Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :  5
 1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5
 Median :15.50   Median :76.00   Median :1.000   toyota corolla    :  5
 Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :  4
 3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4
 Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:  4
                                                 (Other)           :365
```

Graphs

- Generate figures of single variables using plot()

```
> plot(College$F.Undergrad, type='l', xlab="College Index", ylab="#Full-time Undergraduates",
main="Figure 1")
```



```
> attach(College)
> plot(F.Undergrad, type='l', xlab="College Index", ylab="#Full-time Undergraduates", main="Figure 1")
```

Graphs

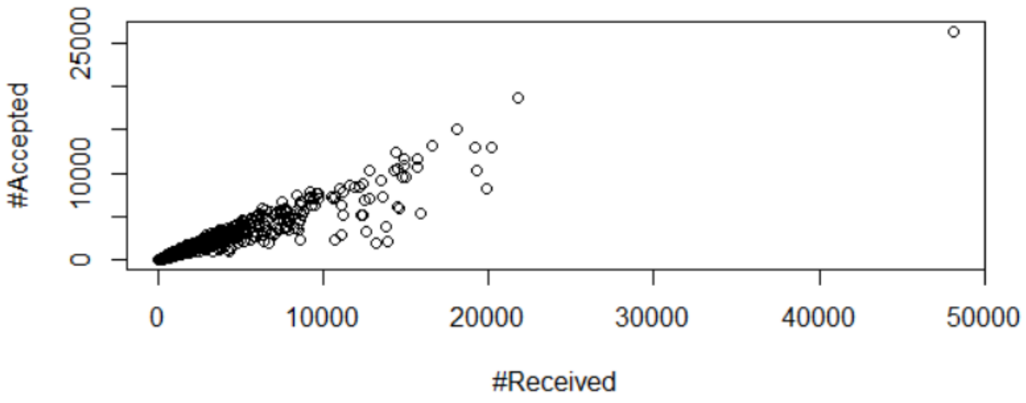
- Generate plots of two variables

```
> plot(Apps, Accept, xlab="#Received",
ylab="#Accepted")
```

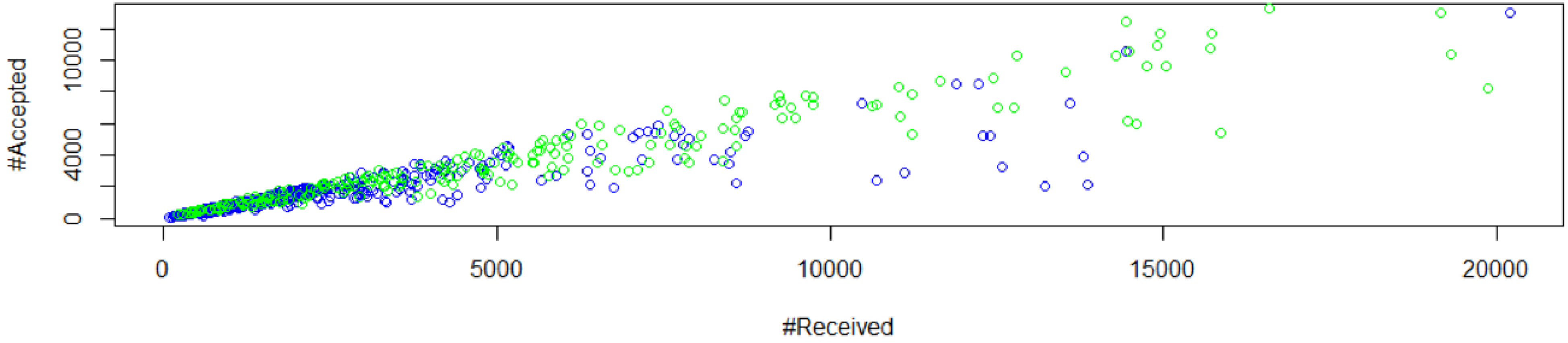
Private : Public/private indicator

Apps : Number of applications received

Accept : Number of applicants accepted



```
> plot(Apps[Private=="Yes"], Accept[Private=="Yes"], col="blue", xlab="#Received", ylab="#Accepted")
> points(Apps[Private=="No"], Accept[Private=="No"], col="green", xlab="#Received", ylab="#Accepted")
```


#Accepted
#Received