STAT 224 Lecture 0
A Brief Introduction to R and RStudio

Yibi Huang
Department of Statistics
University of Chicago

Preface

This lecture is for students have never or haven't used R and RStudio for a long time.

If you have been using R and RStudio, you can quickly skim through the slides and can skip the video.

What Are R & RStudio?

- R is a statistical programming language
- RStudio is a convenient interface for R (an integrated development environment, IDE)
- An analogy
    - R is like a car's engine
    - RStudio is like a car's dashboard

R: Engine

RStudio: Dashboard

Installation of R and RStudio

Installation of R

Go to https://cran.r-project.org/. Select Download R for **Windows**, **MacOS**, or **Linux** depending on your operation system to the dowloading page for the installation file.

- Windows: Click “Download R x.y.z for Windows”. Then run downloaded .exe file to install. Agree to all of the installation defaults (unless you already know how to customize R).
- MacOS: Click on one of the “R-x.y.z.pkg” file depending on **the version of your Mac OS** to begin the installation. You can leave the default options as is just like for Windows. Please note that you might have to (re)install Xquartz if you use Mac OS X.
- Linux: I assume Linux users know how to install software on it yourself . . .

Installation of RStudio

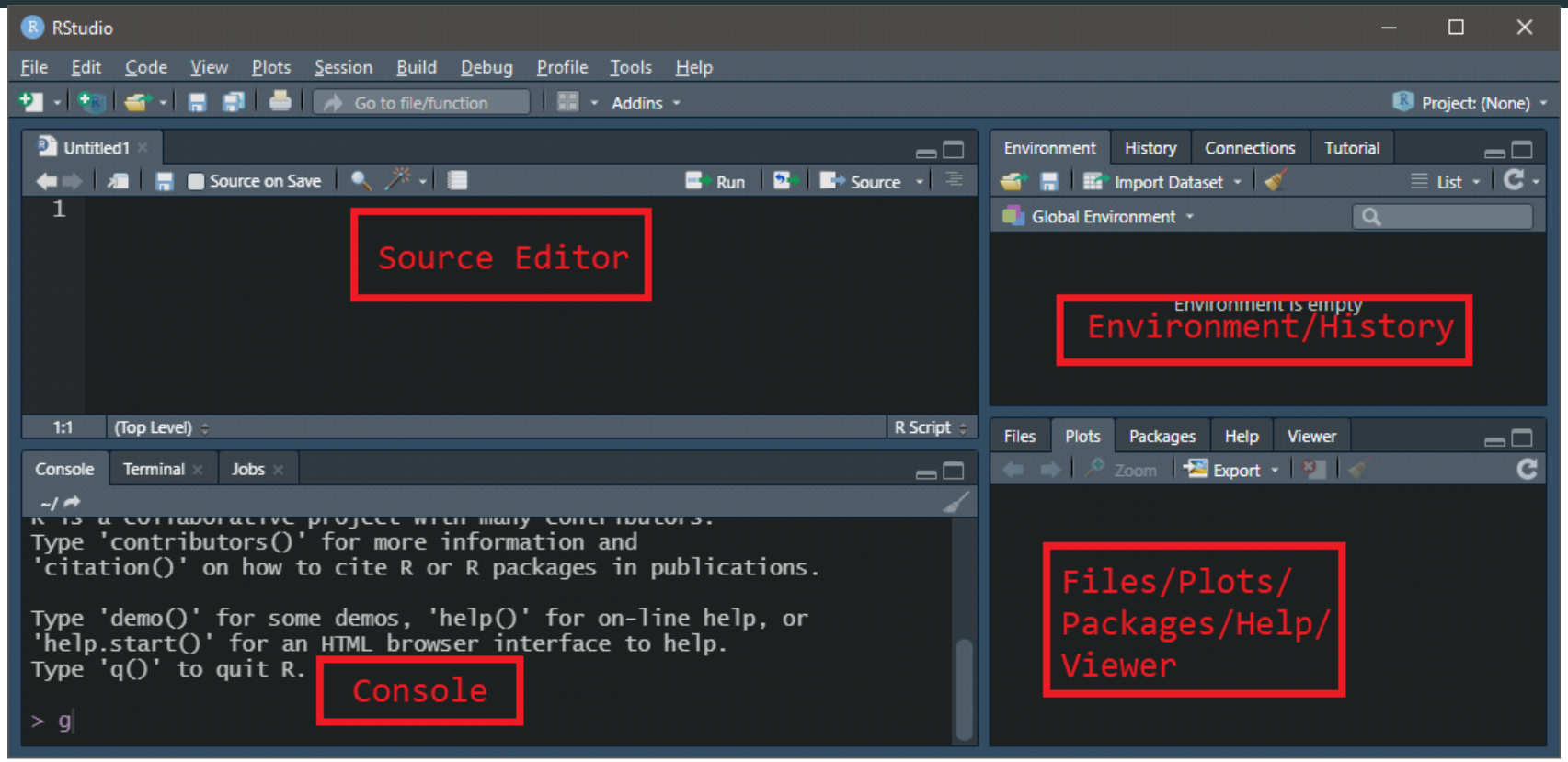
Go to https://www.rstudio.com/products/rstudio/download/ and select [RStudio Desktop]. Select an installer based on your OS and then install.

If You have previously installed R or R Studio . . .

If you have previously installed R or RStudio on your machine, I recommend **uninstalling the old versions** and **installing the latest version of R and RStudio** since some libraries that we'll install might not be compatible with older versions of R.

RStudio Interface

RStudio Interface



- Type R codes in the [Source Editor], and hit [Ctrl+Enter] (Windows) or [Cmd+Enter] (Mac) to execute. Output in the [Console].
- Select several lines of R codes and hit [Ctrl+Enter] (or [Cmd+Enter] on Mac) to execute the selected lines.
- You can save your R codes in a .R file (R script) for reuse.

R as a Calculator

R as a Calculator

```
3+2
[1] 5
3-2
[1] 1
3*2
[1] 6
3/2
[1] 1.5
3^2 # power
[1] 9
sqrt(3)
[1] 1.732051
exp(3)
[1] 20.08554
log(100)   # natural log
[1] 4.60517
log10(100) # base 10 log
[1] 2
```

$$\log\left(\frac{10}{1+\sqrt{3}}\right)$$

```
log(10/(1+sqrt(3)))
[1] 1.297533
```

$$\sqrt{\frac{3^2}{e^{1.5}+1}}$$

```
sqrt(3^2/(exp(1.5)+1))
[1] 1.281339
```

You can save them as an “object” and then reuse them later.

```
x = log(10/(1+sqrt(3)))
y = sqrt(3^2/(exp(1.5)+1))
x*y
[1] 1.662579
```

Vectors

Vectors

A `vector` is a sequence of values, which can be created using the `c()` function, where `c()` stands for "combine" or "concatenate."

```
z = c(5, 3, 1, 6, 7, 2) # This line saves 5,3,1,6,7,2 as a vector z
z     # This line prints the entire vector
[1] 5 3 1 6 7 2
```

Ways to create some commonly used vectors:

```
1:8
[1] 1 2 3 4 5 6 7 8
10:15
[1] 10 11 12 13 14 15
seq(from = 3, to = 21, by = 2)
 [1]  3  5  7  9 11 13 15 17 19 21
rep(5, 4)        # rep means "repeat"
[1] 5 5 5 5
rep(c(1,2), 4)
[1] 1 2 1 2 1 2 1 2
```

Indexes of Vectors

You can use square brackets `[]` to retrieve elements in a vector.

```
z = c(5, 3, 1, 6, 7, 2)
z[3]        # the 3rd element
[1] 1
z[c(1,4)]   # the 1st and 4th elements
[1] 5 6
z[1:4]      # the first 4 elements
[1] 5 3 1 6
```

Negative indexes means excluding those elements

```
z[-c(1,4)]  # excluding the 1st and 4th elements
[1] 3 1 7 2
z[-(1:3)]   # excluding the first 3 elements
[1] 6 7 2
```

Computation with Vectors

Comutations (`+`,`-`,`/`,`*`,`log()`, `...`) on a vector are applied to every element of the vector.

```
z = c(5, 3, 1, 6, 7, 2)
z+2 # add every element by 2
[1] 7 5 3 8 9 4
z*2 # multiply every element by 2
[1] 10  6  2 12 14  4
z^2 # square every element
[1] 25  9  1 36 49  4
```

Computation with Vectors

Computations on two or more vectors are applied elementwise if the vectors are of the same length.

```
v1 = c(1,2,3,1,2)
v2 = c(0,1,2,3,0)
v1+v2
[1] 1 3 5 4 2
v1*v2
[1] 0 2 6 3 0
```

sort(), min(), max(), sum(), mean(), length()

```
z = c(5, 3, 1, 6, 7, 2)
sort(z)  # sort elements in z from minimum to maximum
[1] 1 2 3 5 6 7
min(z)   # minimum of z
[1] 1
max(z)   # maximum of z
[1] 7
sum(z)   # the sum of all elements of z
[1] 24
mean(z)  # the mean of all elements of z
[1] 4
length(z) # the length of vector z
[1] 6
```

Loading Data and Changing Working Directory

The FEV Data

The data file `fevdata.txt` contains data from a sample of 654 youths, aged 3 to 19, in the area of East Boston during middle to late 1970's.

```
  age   fev   ht sex smoke
    9 1.708 57.0   0     0
    8 1.724 67.5   0     0
    7 1.720 54.5   0     0
    9 1.558 53.0   1     0
    9 1.895 57.0   1     0
    8 2.336 61.0   0     0
...(omitted)...
   16 2.795 63.0   0     1
   15 3.211 66.5   0     0
```

Variables of the FEV Data

- **age**: Subject's age in years
- **fev**: Lung capacity of subject, measured by **forced expiratory volume** (abbreviated as **FEV**), the amount of air an individual can exhale in the first second of forceful breath in liters
- **ht**: Subject's height in inches
- **sex**: Gender of the subject, coded as: 0 = Female, 1 = Male
- **smoke**: Subject's smoking status, coded as:
  0 = Nonsmoker, 1 = Smoker

How to Import Data from a File to R?

- First, download the data file `fevdata.txt` from the link `http://www.stat.uchicago.edu/~yibi/s224/data/fevdata.txt` and save it on your computer
- The R command to load data from a text file is `read.table()`

```
fevdata = read.table("fevdata.txt", header=TRUE)
# cannot open file 'fevdata.txt': No such file or directory Error
# in file(file, "rt") : cannot open the connection
```

Oops! We get an error message!!

How to Import Data from a File to R?

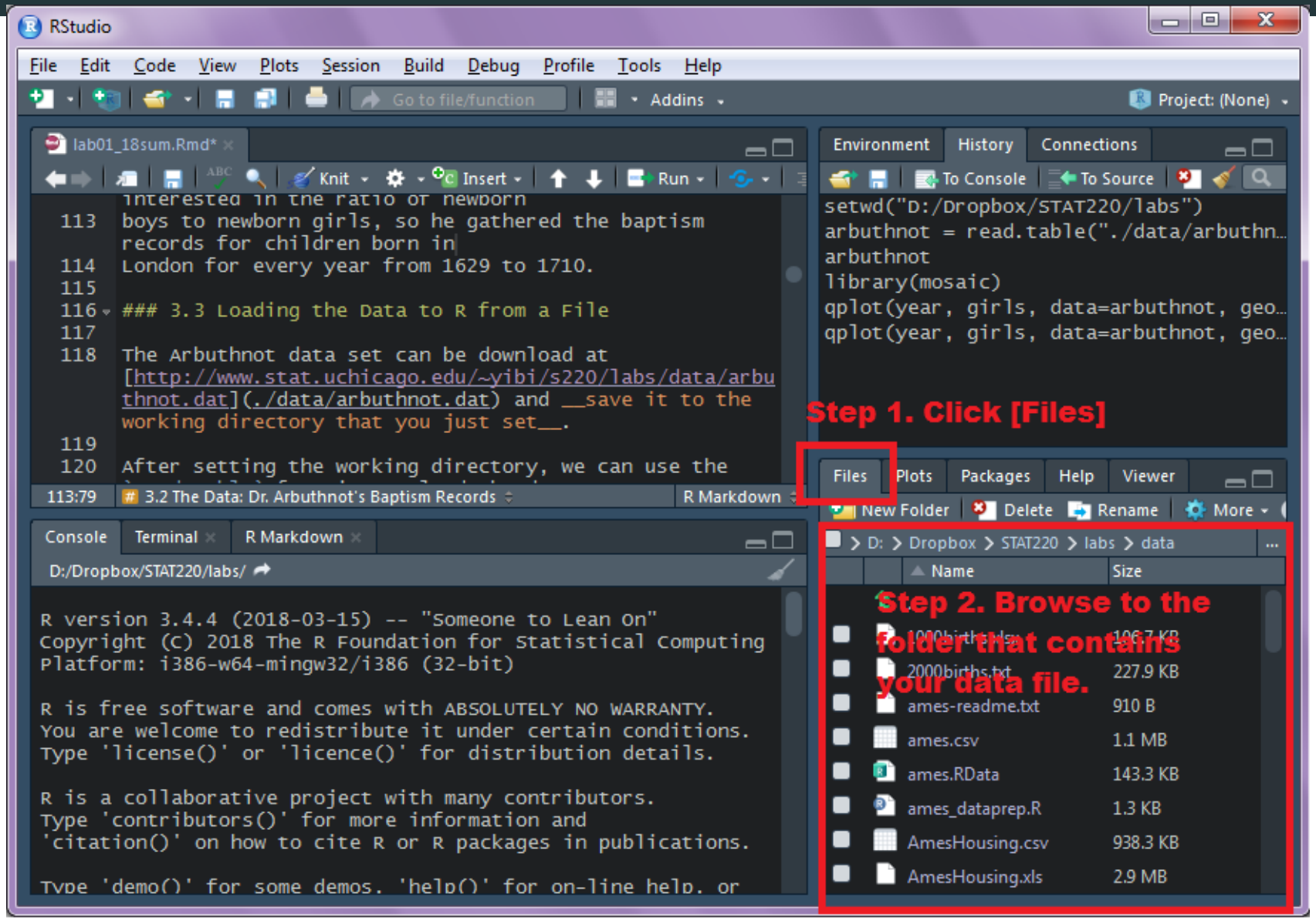
- First, download the data file `fevdata.txt` from the link `http://www.stat.uchicago.edu/~yibi/s224/data/fevdata.txt` and save it on your computer
- The R command to load data from a text file is `read.table()`

```
fevdata = read.table("fevdata.txt", header=TRUE)
# cannot open file 'fevdata.txt': No such file or directory Error
# in file(file, "rt") : cannot open the connection
```

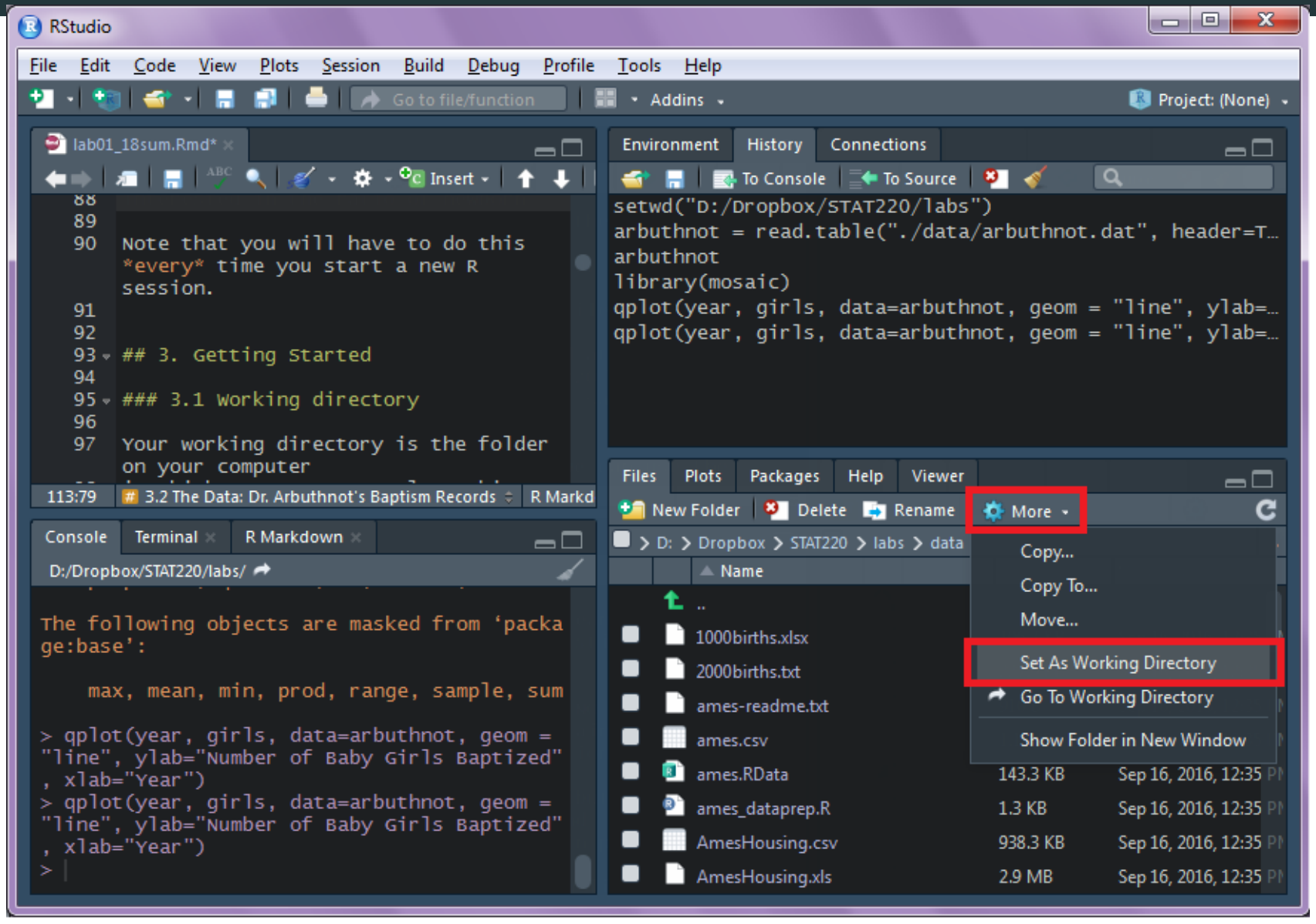
Oops! We get an error message!!

This is because we haven't told R where the file is located, which can be done by providing the complete path to the file or by **setting the working directory** to the folder the data file is located.

How to Set the Working Directory? (1)



How to Set the Working Directory? (2)



Let's Try Loading Data From a File Again!

```
fevdata = read.table("fevdata.txt", header=TRUE)
```

If it works, you should see `fevdata` show up in the [Environment] panel on the top right.

`fevdata` is an data matrix (in R we called it a `data frame`).

The R function `dim()` returns the dimension of a data frame.

```
dim(fevdata)
[1] 654   5
```

We can see `fevdata` has 654 rows (cases) and 5 columns (variables).

Data Frame

Indexes for a Data Frame

```
fevdata[2,]          # 2nd row of data
  age   fev   ht sex smoke
2   8 1.724 67.5   0     0
```

```
fevdata[,3]        # 3rd column of data
#  [1] 57.0 67.5 54.5 53.0 57.0 61.0 58.0
# ...(omitted)...
#[650] 67.0 68.0 60.0 63.0 66.5
```

```
fevdata[1:5,3]     # first 5 elements in the 3rd column
[1] 57.0 67.5 54.5 53.0 57.0
```

Negative indexes also mean “exclusion”

```
fevdata[1:5,-3]    # the first 5 rows but not the 3rd column
fevdata[-(1:5),]   # excluding the first 5 rows of data
```

head() and tail()

head() extract the first few rows of a data frame (6 rows by default)

```
head(fevdata)
  age   fev   ht sex smoke
1   9 1.708 57.0   0     0
2   8 1.724 67.5   0     0
3   7 1.720 54.5   0     0
4   9 1.558 53.0   1     0
5   9 1.895 57.0   1     0
6   8 2.336 61.0   0     0
```

```
head(fevdata, 3)  # the first 3 rows, output omitted
```

Similarly, tail() extracts the last few rows of a data frame.

```
tail(fevdata,3)
    age   fev   ht sex smoke
652  18 2.853 60.0   0     0
653  16 2.795 63.0   0     1
```

`str()` Shows the Structure of a Data frame

```
str(fevdata)
'data.frame':   654 obs. of  5 variables:
 $ age  : int  9 8 7 9 9 8 6 6 8 9 ...
 $ fev  : num  1.71 1.72 1.72 1.56 1.9 ...
 $ ht   : num  57 67.5 54.5 53 57 61 58 56 58.5 60 ...
 $ sex  : int  0 0 0 1 1 0 0 0 0 0 ...
 $ smoke: int  0 0 0 0 0 0 0 0 0 0 ...
```