Introduction to R Programming

Jordan Fish

May 22, 2013

Table of contents

What is R?

R is a statistical computing environment

- Many many common statistical tools baked in
- Easy data manipulation
- Robust add-on community
  `http://cran.r-project.org/web/packages/`
- Extensive plotting capabilities
- Free and open source! `http://www.r-project.org/`

R as a calculator

- Support for all the basic operators

R as a calculator

- Support for all the basic operators

```
> 2 + 2
[1] 4
> 4 * 8
[1] 32
> 20/3
[1] 6.7
```

R as a calculator

- Support for all the basic operators

```
> 2 + 2
[1] 4
> 4 * 8
[1] 32
> 20/3
[1] 6.7
```

- And more

R as a calculator

- Support for all the basic operators

```
> 2 + 2
[1] 4
> 4 * 8
[1] 32
> 20/3
[1] 6.7
```

- And more

```
>   5**2
[1] 25
>   sqrt(625)
[1] 25
>   log(10)
[1] 2.3
```

Variables in R

- We can store the results of any expression in a variable

Variables in R

- We can store the results of any expression in a variable

```
> a = 2 + 2
```

Variables in R

- We can store the results of any expression in a variable

```
> a = 2 + 2
```

- And we can display the contents of a variable

```
> a
[1] 4
```

Variables in R

- We can store the results of any expression in a variable

```
> a = 2 + 2
```

- And we can display the contents of a variable

```
> a
[1] 4
```

- Not as common anymore, but <- can be used instead of =

```
>   a <- 2 + 2
```

(Almost) everything is a vector!

- R thinks about data differently. In R, almost everything is a vector.

(Almost) everything is a vector!

- R thinks about data differently. In R, almost everything is a vector.
- Lets look at 'a' again

(Almost) everything is a vector!

- R thinks about data differently. In R, almost everything is a vector.
- Lets look at 'a' again

```
> a
```

```
[1] 4
```

(Almost) everything is a vector!

- R thinks about data differently. In R, almost everything is a vector.
- Lets look at 'a' again

```
> a
```

```
[1] 4
```

- Notice the [1], R treats scalars as one dimensional vectors

R vectors and you

- We can create vectors using the c(...) function

R vectors and you

- We can create vectors using the c(...) function

```
> a = c(1, 2, 3)
> b = c(4, 5, 6)
> a
[1] 1 2 3
> b
[1] 4 5 6
```

R vectors and you

- We can create vectors using the c(...) function

```
> a = c(1, 2, 3)
> b = c(4, 5, 6)
> a
```

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

```
> b
```

```
[1] 4 5 6
```

- And since R thinks of everything as vectors we can do some very intuitive things with them

R vectors and you

- We can create vectors using the c(...) function

```
> a = c(1, 2, 3)
> b = c(4, 5, 6)
> a
[1] 1 2 3
> b
[1] 4 5 6
```

- And since R thinks of everything as vectors we can do some very intuitive things with them

```
> a + b
[1] 5 7 9
> a * b
[1]  4 10 18
```

R vectors and you

- We can access the elements of a vector

R vectors and you

- We can access the elements of a vector

```
> a[2]
[1] 2
```

R vectors and you

- We can access the elements of a vector

```
> a[2]
[1] 2
> a[1] + a[2] + a[3]
[1] 6
```

R vectors and you

- We can access the elements of a vector

```
> a[2]
[1] 2
> a[1] + a[2] + a[3]
[1] 6
```

- Note R vector indices start at **1** not 0

R vectors and you

- We can access the elements of a vector

```
> a[2]
[1] 2
> a[1] + a[2] + a[3]
[1] 6
```

- Note R vector indices start at **1** not 0
- We can also create ranges of values

R vectors and you

- We can access the elements of a vector

```
> a[2]
[1] 2
> a[1] + a[2] + a[3]
[1] 6
```

- Note R vector indices start at **1** not 0
- We can also create ranges of values

```
> 1:10
 [1]  1  2  3  4  5  6  7  8  9 10
```

R vectors and you

- We can access the elements of a vector

```
> a[2]
[1] 2
> a[1] + a[2] + a[3]
[1] 6
```

- Note R vector indices start at **1** not 0
- We can also create ranges of values

```
> 1:10
 [1]  1  2  3  4  5  6  7  8  9 10
```

- And more complex sequences

R vectors and you

- We can access the elements of a vector

```
> a[2]
[1] 2
> a[1] + a[2] + a[3]
[1] 6
```

- Note R vector indices start at **1** not 0
- We can also create ranges of values

```
> 1:10
 [1]  1  2  3  4  5  6  7  8  9 10
```

- And more complex sequences

```
> seq(from=1, to=20, by=2)
 [1]  1  3  5  7  9 11 13 15 17 19
```

Functions with vectors

- Since R is vector based we can pass vectors to functions

Functions with vectors

- Since R is vector based we can pass vectors to functions

```
> a
[1] 1 2 3
> b
[1] 4 5 6
```

Functions with vectors

- Since R is vector based we can pass vectors to functions

```
> a
[1] 1 2 3
> b
[1] 4 5 6
> length(a)
[1] 3
```

Functions with vectors

- Since R is vector based we can pass vectors to functions

```
> a
[1] 1 2 3
> b
[1] 4 5 6
> length(a)
[1] 3
> sqrt(b)
[1] 2.0 2.2 2.4
```

Functions with vectors

- Since R is vector based we can pass vectors to functions

```
> a
[1] 1 2 3
> b
[1] 4 5 6
> length(a)
[1] 3
> sqrt(b)
[1] 2.0 2.2 2.4
> sum(a)
[1] 6
```

Functions with vectors

- Since R is vector based we can pass vectors to functions

```
> a
[1] 1 2 3
> b
[1] 4 5 6
> length(a)
[1] 3
> sqrt(b)
[1] 2.0 2.2 2.4
> sum(a)
[1] 6
> mean(a + b)
[1] 7
```

Exercises

- Create a vector containing 4 numbers
- Compute the mean by directly adding the numbers
- Compute the mean with the mean(...) function

Exercises

- Create a vector containing 4 numbers

Exercises

- Create a vector containing 4 numbers

```
> my.numbers = c(2/3, 1.1, 5, 22)
```

Exercises

- Create a vector containing 4 numbers

  ```
  > my.numbers = c(2/3, 1.1, 5, 22)
  ```

- Compute the mean by directly adding the numbers

Exercises

- Create a vector containing 4 numbers

```
> my.numbers = c(2/3, 1.1, 5, 22)
```

- Compute the mean by directly adding the numbers

```
> (my.numbers[1] + my.numbers[2] +
+    my.numbers[3] + my.numbers[4]) / 4
[1] 7.2
```

Exercises

- Create a vector containing 4 numbers

```
> my.numbers = c(2/3, 1.1, 5, 22)
```

- Compute the mean by directly adding the numbers

```
> (my.numbers[1] + my.numbers[2] +
+    my.numbers[3] + my.numbers[4]) / 4
[1] 7.2
```

- Compute the mean with the mean(...) function

Exercises

- Create a vector containing 4 numbers

```
> my.numbers = c(2/3, 1.1, 5, 22)
```

- Compute the mean by directly adding the numbers

```
> (my.numbers[1] + my.numbers[2] +
+    my.numbers[3] + my.numbers[4]) / 4
```

```
[1] 7.2
```

- Compute the mean with the mean(...) function

```
> mean(my.numbers)
```

```
[1] 7.2
```

Getting help

R can do a whole lot!

- Always remember: tab completion

Getting help

R can do a whole lot!

- Always remember: tab completion
  - Works with variable names
  - Works with column names (more on that in a minute)
  - Works with function names AND arguments

Getting help

R can do a whole lot!

- Always remember: tab completion
  - Works with variable names
  - Works with column names (more on that in a minute)
  - Works with function names AND arguments
- The ? operator will tell you what functions do

Getting help

R can do a whole lot!

- Always remember: tab completion
  - Works with variable names
  - Works with column names (more on that in a minute)
  - Works with function names AND arguments
- The ? operator will tell you what functions do

```
> ?sqrt
```

```
> ?"+"
```

Getting help

R can do a whole lot!

- Always remember: tab completion
  - Works with variable names
  - Works with column names (more on that in a minute)
  - Works with function names AND arguments
- The ? operator will tell you what functions do

  ```
  > ?sqrt
  ```

  ```
  > ?"+"
  ```

- The ?? operator will search the help for a keyword

  ```
  > ??binomial
  ```

Getting help

R can do a whole lot!

- Always remember: tab completion
  - Works with variable names
  - Works with column names (more on that in a minute)
  - Works with function names AND arguments
- The ? operator will tell you what functions do

  ```
  > ?sqrt
  ```

  ```
  > ?"+"
  ```
- The ?? operator will search the help for a keyword

  ```
  > ??binomial
  ```
- More on the web!
  - `http://cran.r-project.org/manuals.html`
  - `http://cran.r-project.org/doc/FAQ/R-FAQ.html`

Loading Data

- Loading data in R uses the read.* family of functions
  - read.csv(...) for comma separated value files
  - read.table(...) for tab delimited files
- The read functions can read from a local file and the web
- When working with local files always be aware of your current directory

```
> getwd()

[1] "/export/home/fishjord/documents/ged_lab/SROP/R_intro"
```

- `http://fishjord.github.io/resources/weather_year.csv`

```
> data = read.csv("http://fishjord.github.io/resources/weather_year.csv", header=T)
```

Data details

- head(...) will show us the first few rows

```
>   head(data)
```

	temp	humidity	pressure	wind.speed	precipitation
1	40	50	30	6	0e+00
2	49	53	30	7	1e-07
3	62	76	30	14	3e-02
4	63	66	30	5	0e+00
5	62	68	30	6	0e+00
6	69	69	30	10	4e-02

- where tail(...) does the opposite

```
>   tail(data)
```

	temp	humidity	pressure	wind.speed	precipitation
361	35	80	30	10	2.6e-01
362	31	85	30	9	4.0e-02
363	32	71	30	2	0.0e+00
364	35	65	30	3	0.0e+00
365	45	57	30	10	1.0e-07
366	52	48	30	8	0.0e+00

Data Structure

- length(...) will tell us the length of something

```
> length(data)
[1] 5
```

Data Structure

- length(...) will tell us the length of something

```
> length(data)
[1] 5
```

- dim(...) will tell us the dimensions of the object

```
> dim(data)
[1] 366   5
```

Data Structure

- class(...) will tell us what kind of object we have

```
> class(data)
[1] "data.frame"
```

Data Structure

- class(...) will tell us what kind of object we have

```
> class(data)
[1] "data.frame"
```

- DataFrames are the primary way to interact with data in R
- Can think of them as matrices

Data details

str(...) will tell us about the structure of the object

```
> str(data)

'data.frame': 366 obs. of  5 variables:
 $ temp          : int  40 49 62 63 62 69 64 62 70 72 ...
 $ humidity      : int  50 53 76 66 68 69 75 78 69 66 ...
 $ pressure      : num  30.4 30.3 30 30.1 30.1 ...
 $ wind.speed    : int  6 7 14 5 6 10 5 5 8 6 ...
 $ precipitation: num  0e+00 1e-07 3e-02 0e+00 0e+00 4e-02 1e-07
```

Data details

summary(...) will give a summary of the object

```
>   summary(data)
```

```
      temp          humidity        pressure
 Min.   :11.0   Min.   :37.0   Min.   :29.4
 1st Qu.:41.0   1st Qu.:61.2   1st Qu.:29.9
 Median :59.0   Median :68.0   Median :30.0
 Mean   :55.7   Mean   :67.9   Mean   :30.0
 3rd Qu.:70.8   3rd Qu.:74.0   3rd Qu.:30.1
 Max.   :89.0   Max.   :95.0   Max.   :30.5
   wind.speed    precipitation
 Min.   : 0.00   Min.   :0.0000
 1st Qu.: 4.00   1st Qu.:0.0000
 Median : 6.00   Median :0.0000
 Mean   : 6.06   Mean   :0.0969
 3rd Qu.: 8.00   3rd Qu.:0.0275
 Max.   :19.00   Max.   :2.0000
```

Inspecting Dataframes

- Can access a single row

Inspecting Dataframes

- Can access a single row

```
>   data[1,]

  temp humidity pressure wind.speed precipitation
1   40       50       30          6             0
```

Inspecting Dataframes

- Can access a single row

```
>   data[1,]

  temp humidity pressure wind.speed precipitation
1   40       50       30          6             0
```

- Or a range of rows

Inspecting Dataframes

- Can access a single row

```
>   data[1,]
```

```
  temp humidity pressure wind.speed precipitation
1   40       50       30          6             0
```

- Or a range of rows

```
>   data[1:5,]
```

```
  temp humidity pressure wind.speed precipitation
1   40       50       30          6         0e+00
2   49       53       30          7         1e-07
3   62       76       30         14         3e-02
4   63       66       30          5         0e+00
5   62       68       30          6         0e+00
```

Inspecting Dataframes

- Can access a single row

```
>   data[1,]
```

```
  temp humidity pressure wind.speed precipitation
1   40       50       30          6             0
```

- Or a range of rows

```
>   data[1:5,]
```

```
  temp humidity pressure wind.speed precipitation
1   40       50       30          6         0e+00
2   49       53       30          7         1e-07
3   62       76       30         14         3e-02
4   63       66       30          5         0e+00
5   62       68       30          6         0e+00
```

- Or a single column

Inspecting Dataframes

- Can access a single row

```
>   data[1,]
```

```
  temp humidity pressure wind.speed precipitation
1   40       50       30          6             0
```

- Or a range of rows

```
>   data[1:5,]
```

```
  temp humidity pressure wind.speed precipitation
1   40       50       30          6         0e+00
2   49       53       30          7         1e-07
3   62       76       30         14         3e-02
4   63       66       30          5         0e+00
5   62       68       30          6         0e+00
```

- Or a single column

```
>   head(data[,"temp"])
```

```
[1] 40 49 62 63 62 69
```

Inspecting Dataframes

- Can access a single row

```
>   data[1,]
```

```
  temp humidity pressure wind.speed precipitation
1   40       50       30          6             0
```

- Or a range of rows

```
>   data[1:5,]
```

```
  temp humidity pressure wind.speed precipitation
1   40       50       30          6         0e+00
2   49       53       30          7         1e-07
3   62       76       30         14         3e-02
4   63       66       30          5         0e+00
5   62       68       30          6         0e+00
```

- Or a single column

```
>   head(data[,"temp"])
```

```
[1] 40 49 62 63 62 69
```

- Or a set of columns

Inspecting Dataframes

- Can access a single row

```
>   data[1,]
```

```
  temp humidity pressure wind.speed precipitation
1   40       50       30          6             0
```

- Or a range of rows

```
>   data[1:5,]
```

```
  temp humidity pressure wind.speed precipitation
1   40       50       30          6         0e+00
2   49       53       30          7         1e-07
3   62       76       30         14         3e-02
4   63       66       30          5         0e+00
5   62       68       30          6         0e+00
```

- Or a single column

```
>   head(data[,"temp"])
```

```
[1] 40 49 62 63 62 69
```

- Or a set of columns

```
>   head(data[,c("temp", "wind.speed")])
```

```
  temp wind.speed
1   40          6
2   49          7
3   62         14
4   63          5
5   62          6
6   69         10
```

Exercises

- What was the weather like on the 53rd day in the dataset?
- How would you retrieve the pressure on the 5th day?
- The conditions last 3 days? (hint: ?tail)

Exercises

- What was the weather like on the 53rd day in the dataset?

Exercises

- What was the weather like on the 53rd day in the dataset?

  ```
  > data[53,]
  ```

  ```
     temp humidity pressure wind.speed precipitation
  53   69       90       30          6          0.86
  ```

- How would you retrieve the pressure on the 5th day?

Exercises

- What was the weather like on the 53rd day in the dataset?

```
> data[53,]
```

```
   temp humidity pressure wind.speed precipitation
53   69       90       30          6          0.86
```

- How would you retrieve the pressure on the 5th day?

```
> data[53,"pressure"]
```

```
[1] 30
```

```
> data$pressure[53]
```

```
[1] 30
```

- The conditions last 3 days? (hint: ?tail)

Exercises

- What was the weather like on the 53rd day in the dataset?

```
> data[53,]
```

```
   temp humidity pressure wind.speed precipitation
53   69       90       30          6          0.86
```

- How would you retrieve the pressure on the 5th day?

```
> data[53,"pressure"]
```

```
[1] 30
```

```
> data$pressure[53]
```

```
[1] 30
```

- The conditions last 3 days? (hint: ?tail)

```
> tail(data, n=3)
```

```
    temp humidity pressure wind.speed precipitation
364   35       65       30          3          0e+00
365   45       57       30         10          1e-07
366   52       48       30          8          0e+00
```

Sub setting Data

- Days below 15 degrees

Sub setting Data

- Days below 15 degrees

```
> data[data$temp < 15, ]
```

```
    temp humidity pressure wind.speed precipitation
299   13       71       30          1          0e+00
319   12       56       30          6          0e+00
329   11       65       30          8          1e-07
```

Sub setting Data

- Days below 15 or above 85 degrees

Sub setting Data

- Days below 15 or above 85 degrees

```
> data[data$temp < 15 | data$temp > 85, ]
```

	temp	humidity	pressure	wind.speed	precipitation
112	89	50	30	6	0.0e+00
117	87	58	30	5	0.0e+00
118	87	57	30	3	0.0e+00
119	87	62	30	2	0.0e+00
120	89	55	30	3	3.6e-01
130	87	69	30	6	1.0e-07
131	87	61	30	5	0.0e+00
137	88	55	30	7	0.0e+00
138	88	59	30	7	0.0e+00
139	86	55	30	9	1.0e-07
299	13	71	30	1	0.0e+00
319	12	56	30	6	0.0e+00
329	11	65	30	8	1.0e-07

Sub setting Data

- Days 15 degrees and greater than 60% humidity

Sub setting Data

- Days 15 degrees and greater than 60% humidity

```
> data[data$temp < 15 & data$humidity > 60, ]
```

```
    temp humidity pressure wind.speed precipitation
299   13       71       30          1          0e+00
329   11       65       30          8          1e-07
```

Exercises

- List the days with more than 1.5 inches of precipitation
- List the windy (wind speed higher than 15 mph) and rainy (more than 0 precipitation)

Exercises

- List the days with more than 1.5 inch's of precipitation

Exercises

- List the days with more than 1.5 inch's of precipitation

```
> data[data$precipitation > 1.5, ]
```

	temp	humidity	pressure	wind.speed	precipitation
160	72	68	30	8	1.9
177	74	91	30	9	2.0
210	51	79	30	6	1.5
310	45	87	30	7	1.7

Exercises

- List the days with more than 1.5 inch's of precipitation

```
> data[data$precipitation > 1.5, ]
```

	temp	humidity	pressure	wind.speed	precipitation
160	72	68	30	8	1.9
177	74	91	30	9	2.0
210	51	79	30	6	1.5
310	45	87	30	7	1.7

- List the windy (wind speed higher than 15 mph) and rainy (more than 0 precipitation)

Exercises

- List the days with more than 1.5 inch's of precipitation

```
> data[data$precipitation > 1.5, ]
```

	temp	humidity	pressure	wind.speed	precipitation
160	72	68	30	8	1.9
177	74	91	30	9	2.0
210	51	79	30	6	1.5
310	45	87	30	7	1.7

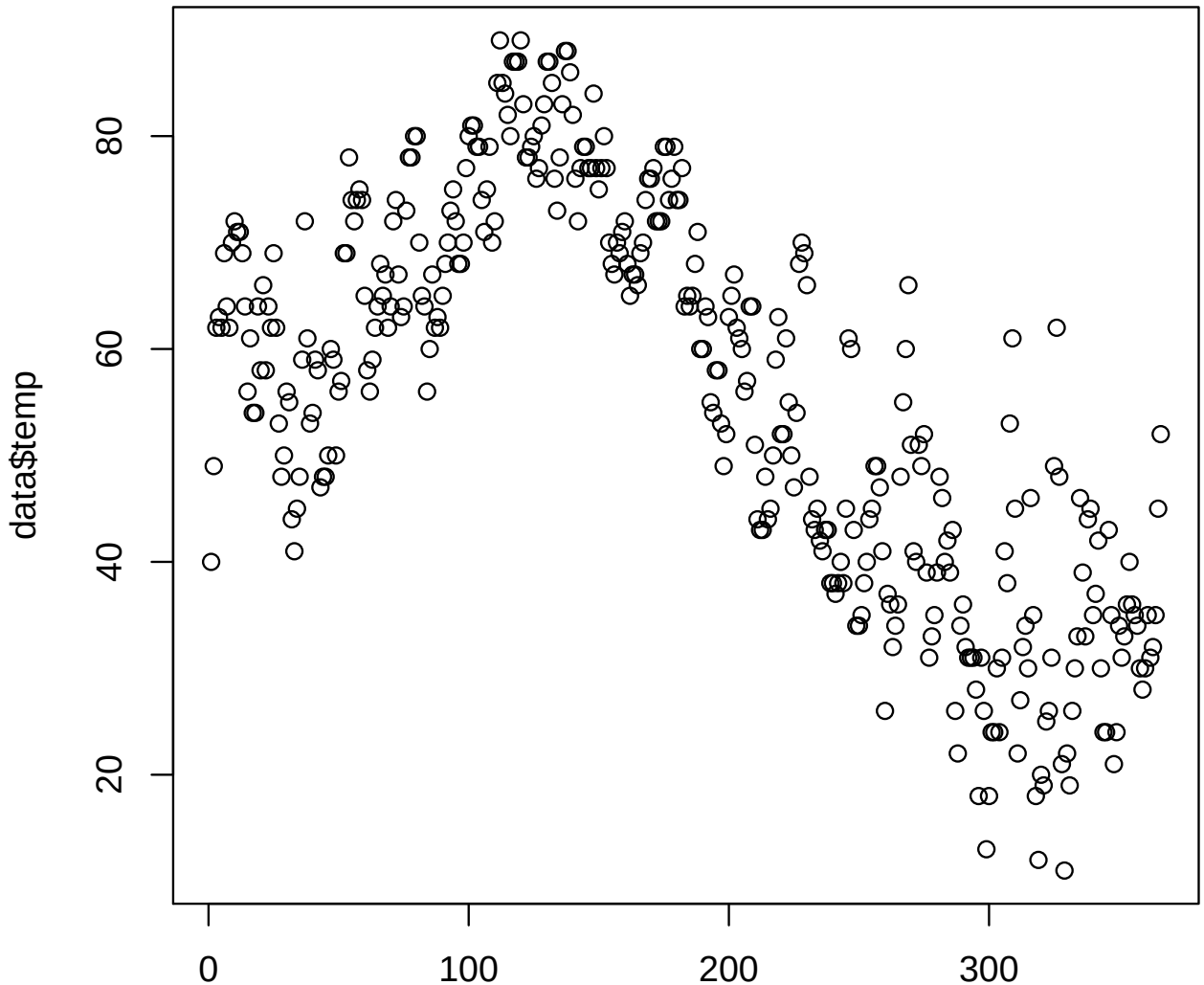
- List the windy (wind speed higher than 15 mph) and rainy (more than 0 precipitation)

```
> data[data$wind.speed > 15 & data$precipitation > 0, ]
```

	temp	humidity	pressure	wind.speed	precipitation
219	63	70	30	17	0.03
286	43	79	29	19	0.44
292	31	81	30	16	0.48
347	35	71	30	16	0.10

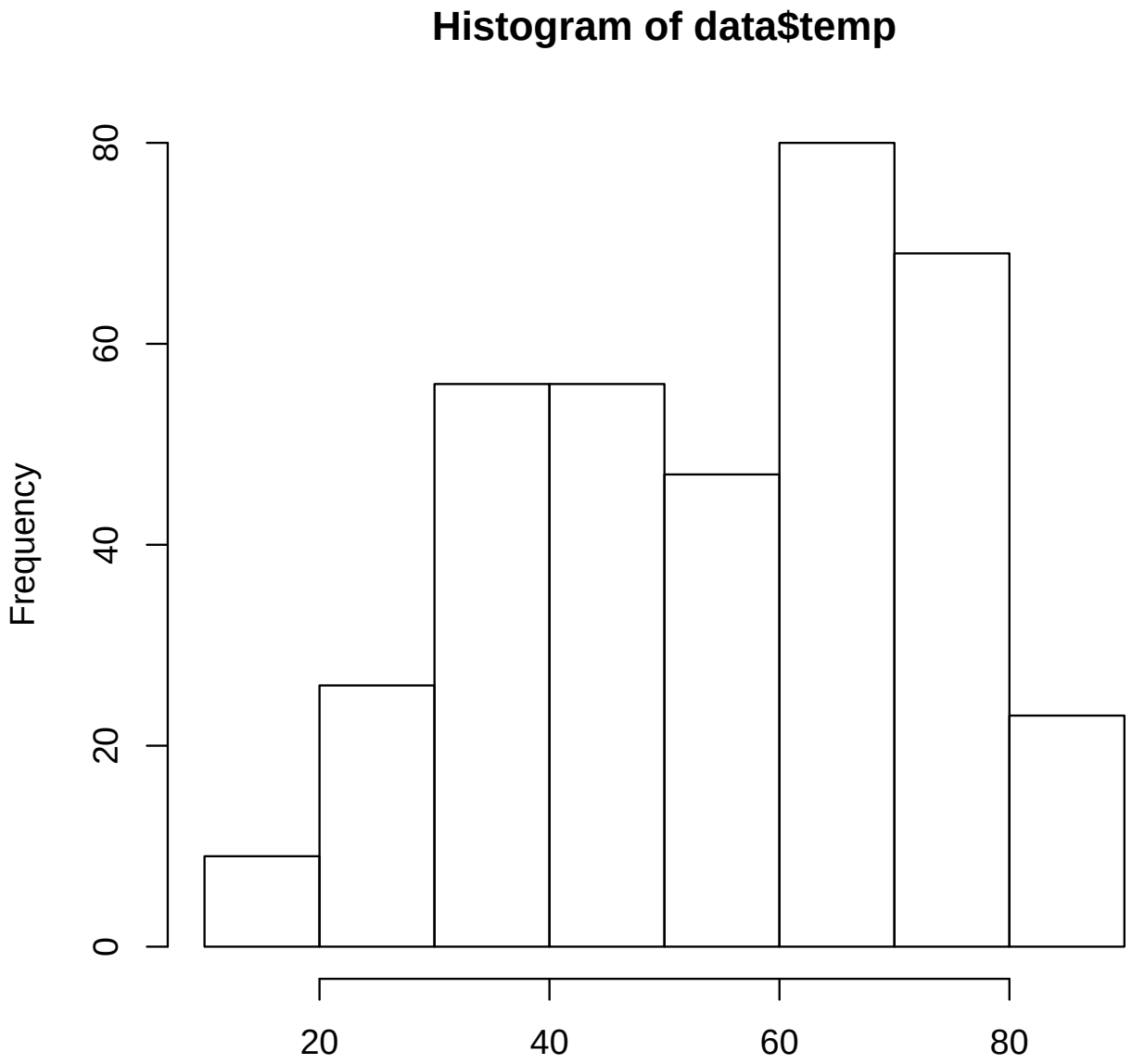
Basic plots

```
> plot(data$temp)
```



Basic plots

```
> hist(data$temp)
```



Basic plots

```
> boxplot(data$temp)
```