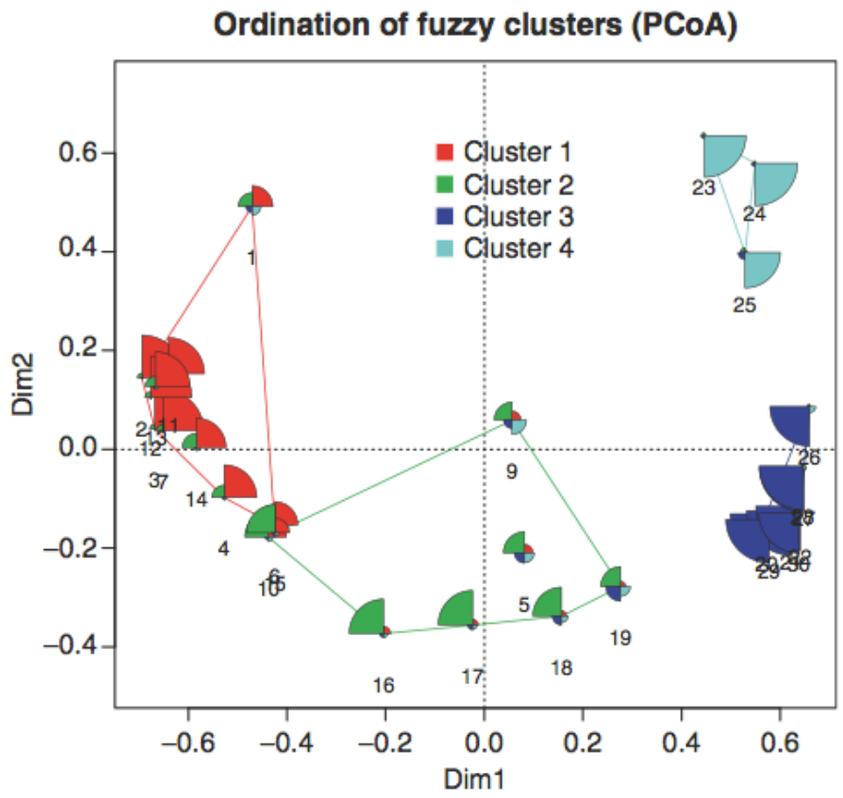
An Introduction to R

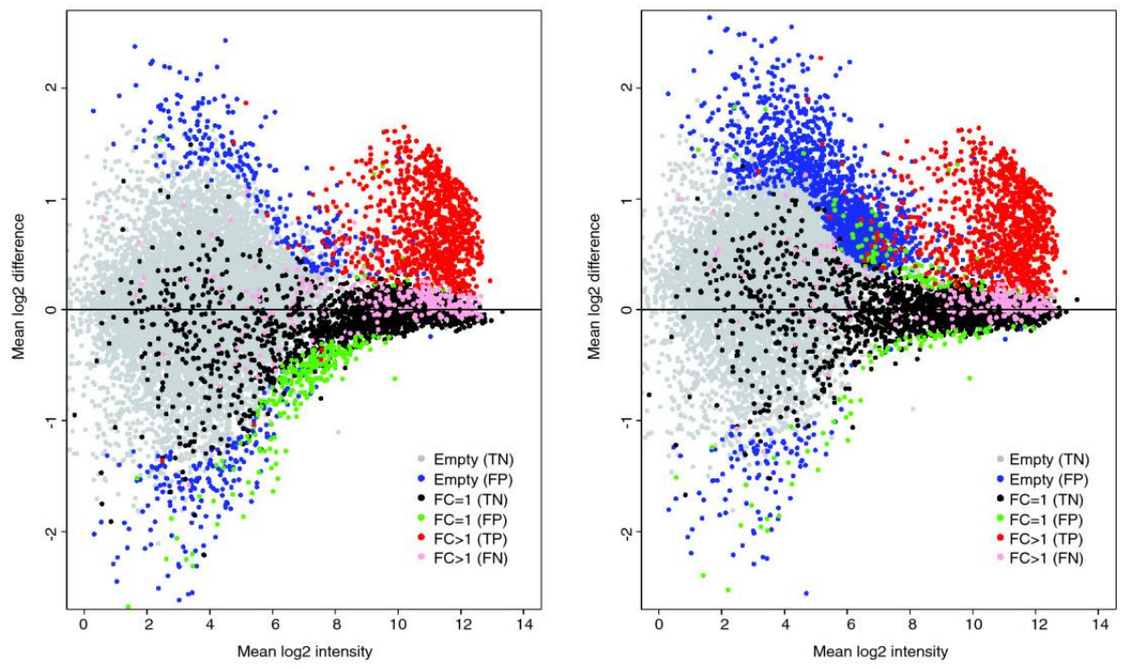
Chris Miller, PhD
Washington University St Louis

Many slides used or adapted with permission from Scott A. Handley, PhD

What is R?

A free software environment for statistical computing and graphics





Why is R useful?

- Data management and manipulation
- Well established system of packages and documentation, especially in bioinformatics
- Support for rich statistical simulation and modeling
- Active development and dedicated community
- Cutting-edge graphical data visualization
- Free!

Things R is less good at

- BIG data

- There is a learning curve from many other languanges

- A common paradigm is to use other tools to massage your data into a bite size chunk, then import that into R for exploration/vizualization

    - e.g. Generate coverage in 10,000 bp bins from a bam file using mosdepth, run stats and make pretty plots of them with R

Where to learn more about R

- The R Project Homepage: http://www.r-project.org
- Quick R Homepage: http://www.statmethods.net
- Bioconductor: http://www.bioconductor.org
- An Introduction to R (long!): http://cran.r-project.org/doc/manuals/R-intro.html
- Google - there are tons of tutorials, guides, demos, packages and more

R for Biologists

- Bioconductor (http://bioconductor.org)
  - 2,140 packages (21-August, 2022):
    - Variant detection: coding changes, PolyPhen database
    - Annotation: pathway analysis, access GO, KEGG, NCBI and many others
    - High-throughput assays: flow cytometry, mass spec
    - Transcription factor binding detection
- Ecology (see: http://cran.r-project.org/web/views/Environmetrics.html)
    - Ordination
    - Cluster Analysis
    - Ecological Theory
    - Population Dynamics
    - Spatial Data Analysis
- Phylogenetics and Evolution (see: http://cran.r-project.org/web/views/Phylogenetics.html)
  - Ancestral State Reconstruction
  - Phylogenetic Inference
  - Trait Evolution

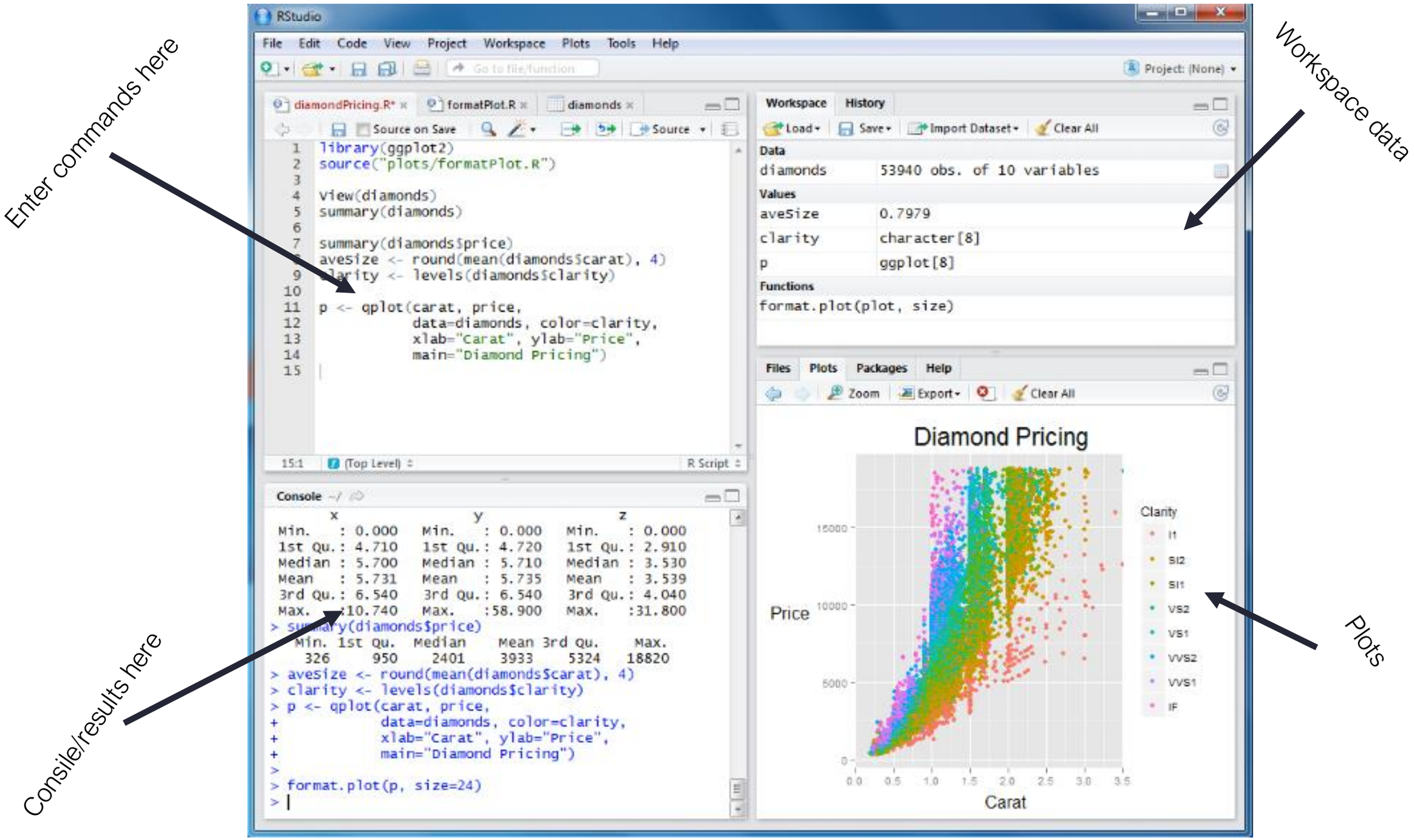
Obtaining R

- Windows, Mac or Linux OS: https://www.r-project.org

Running R

- Install a R Integrated Development Environment (IDE)
  - RStudio: http://www.rstudio.com
  - Makes working with R much easier, particularly for a new R user
  - Run on Windows, Mac or Linux OS
- Or from the command line, type R

R Studio


Enter commands here
Consile/results here
Workspace data
Plots
RStudio
File Edit Code View Project Workspace Plots Tools Help
diamondPricing.R* formatPlot.R diamonds
Source on Save Source
1 library(ggplot2)
2 source("plots/formatPlot.R")
4 View(diamonds)
5 summary(diamonds)
7 summary(diamonds$price)
8 avesize <- round(mean(diamonds$carat), 4)
9 clarity <- levels(diamonds$clarity)
11 p <- qplot(carat, price,
12 data=diamonds, color=clarity,
13 xlab="Carat", ylab="Price",
14 main="Diamond Pricing")
15:1 (Top Level) R Script
Console ~/
x y z
Min. : 0.000 Min. : 0.000 Min. : 0.000
1st Qu.: 4.710 1st Qu.: 4.720 1st Qu.: 2.910
Median : 5.700 Median : 5.710 Median : 3.530
Mean : 5.731 Mean : 5.735 Mean : 3.539
3rd Qu.: 6.540 3rd Qu.: 6.540 3rd Qu.: 4.040
Max. :10.740 Max. :58.900 Max. :31.800
> summary(diamonds$price)
Min. 1st Qu. Median Mean 3rd Qu. Max.
326 950 2401 3933 5324 18820
> avesize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+ data=diamonds, color=clarity,
+ xlab="Carat", ylab="Price",
+ main="Diamond Pricing")
> format.plot(p, size=24)
Workspace History
Load Save Import Dataset Clear All
Data
diamonds 53940 obs. of 10 variables
Values
aveSize 0.7979
clarity character[8]
p ggplot[8]
Functions
format.plot(plot, size)
Files Plots Packages Help
Zoom Export Clear All
Diamond Pricing
Price
Carat
Clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF

Basic R functionality

Calculator

- +, -, /, *, ^, log(), exp(), sqrt(), abs(), cos(), sin(), tan(), ...

```
(4+5^2)/3.14
[1] 9.235669
```

Set Variables / Vectors

```
y=13.4
>y
[1] 13.4
```

```
y=c(1,2,3,4,5)
>y
[1] 1 2 3 4 5
```

Sequences

```
y=rep(2,10)
[1] 2 2 2 2 2 2 2 2 2 2
```

```
y=2:8
[1]  2 3 4 5 6 7 8
```

Statistics

```
t.test(7:34, 5:29)
```

```
t = 1.6348, df = 50.999, p-value = 0.1082
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.797982  7.797982
sample estimates:
mean of x mean of y
     20.5      17.0
```

Manipulation I

n=c(3, 7, 12, 50, 103)

n[4]	[1] 50
n[-2]	[1] 3 12 50 103
n[1:3]	[1] 3 7 12
n[c(1,3,5)]	[1] 3 12 103
n[n<50]	[1] 3 7 12
n[n>8 & n!=50]	[1] 12 103

Manipulation II

```
n=c(3, 7, 12, 50, 103)
```

```
n+1
```

```
[1] 4 8 13 51 104
```

```
sum(n)
```

```
[1] 175
```

```
mean(n)
```

```
[1] 35
```

```
var(n)
```

```
[1] 1796.5
```

```
min(n)
```

```
[1] 3
```

```
max(n)
```

```
[1] 103
```

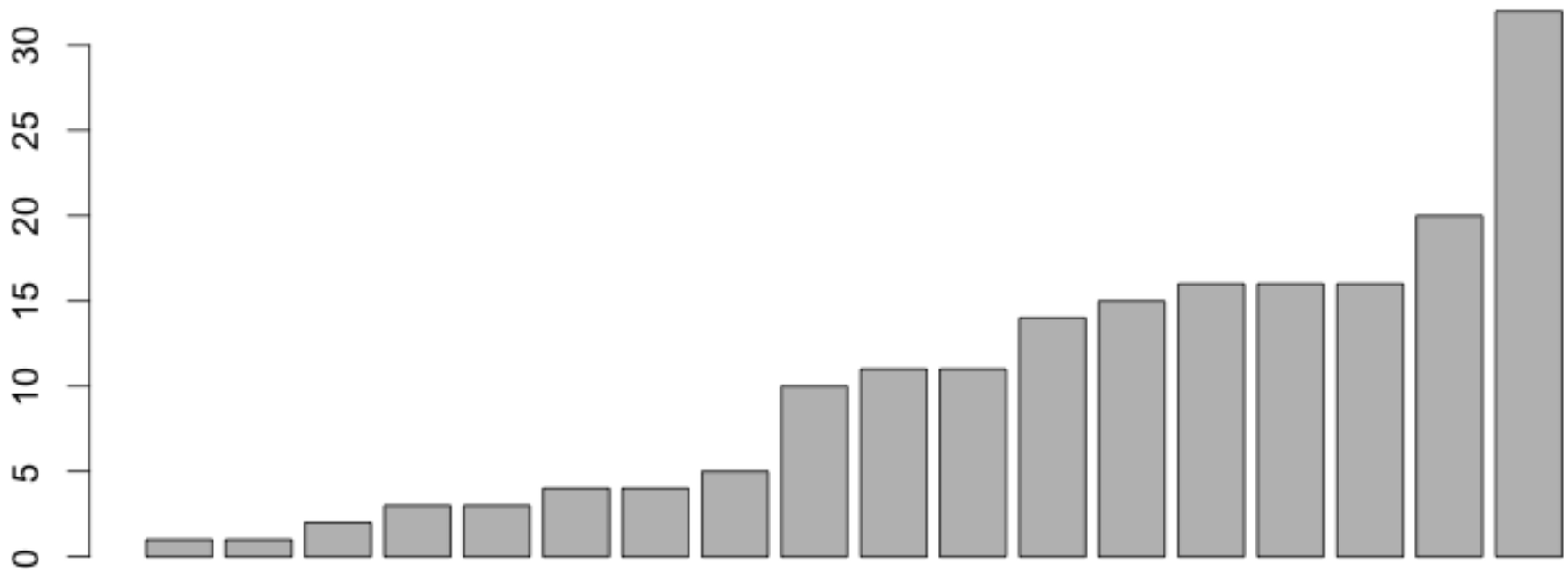
Basic Visualization I

y=c(1,1,2,3,3,4,4,5,10,11,11,14,15,16,16,16,20,32)

Basic Visualization I

y=c(1,1,2,3,3,4,4,5,10,11,11,14,15,16,16,16,20,32)

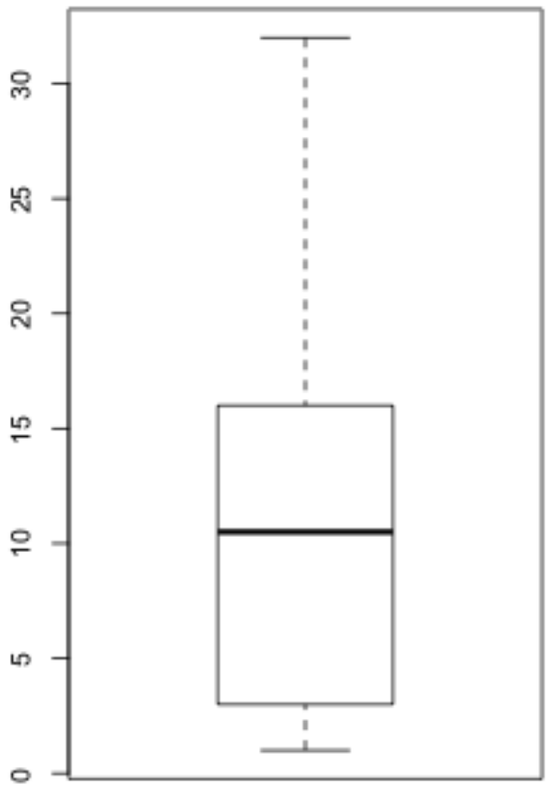
barplot(y)



Basic Visualization II

```
y=c(1,1,2,3,3,4,4,5,10,11,11,14,15,16,16,16,20,32)
```

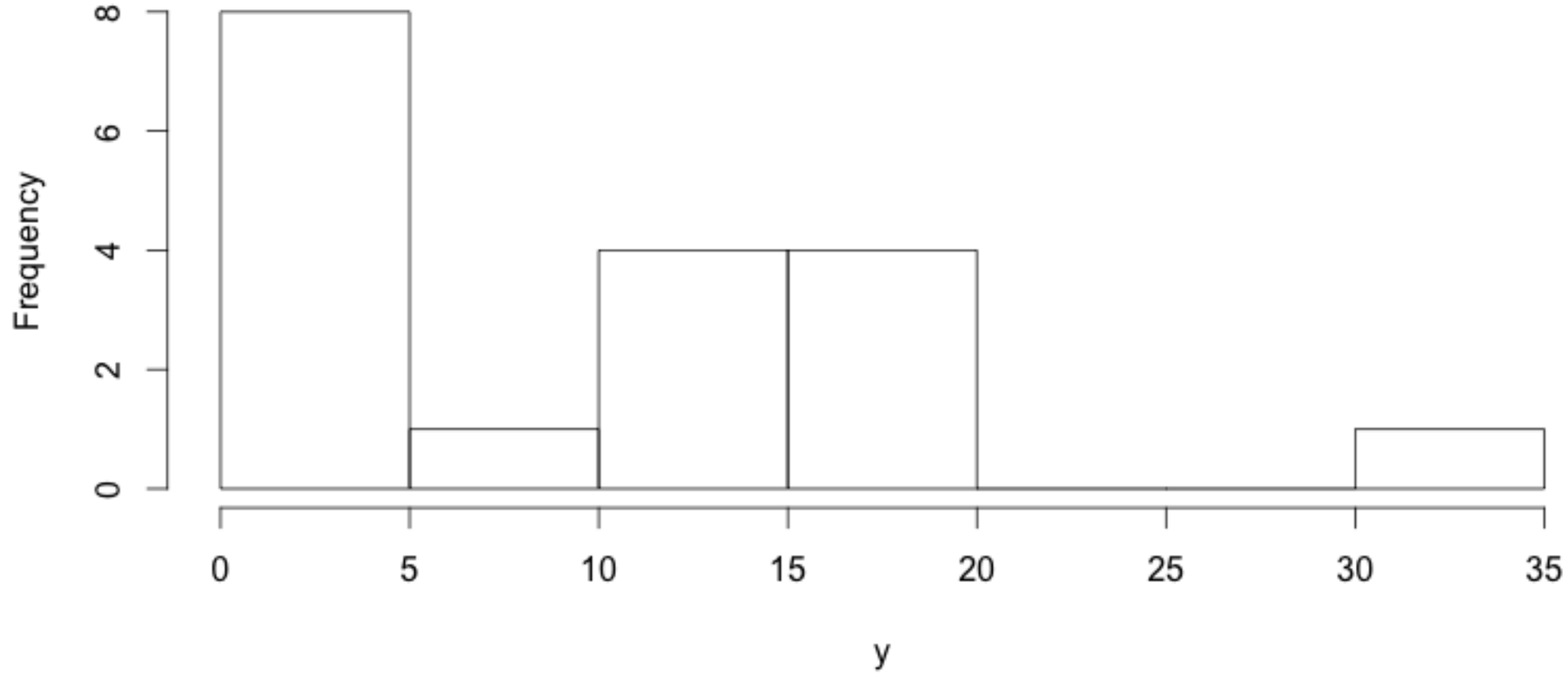
```
boxplot(y)
```



Basic Visualization III

```
y=c(1,1,2,3,3,4,4,5,10,11,11,14,15,16,16,16,20,32)
```

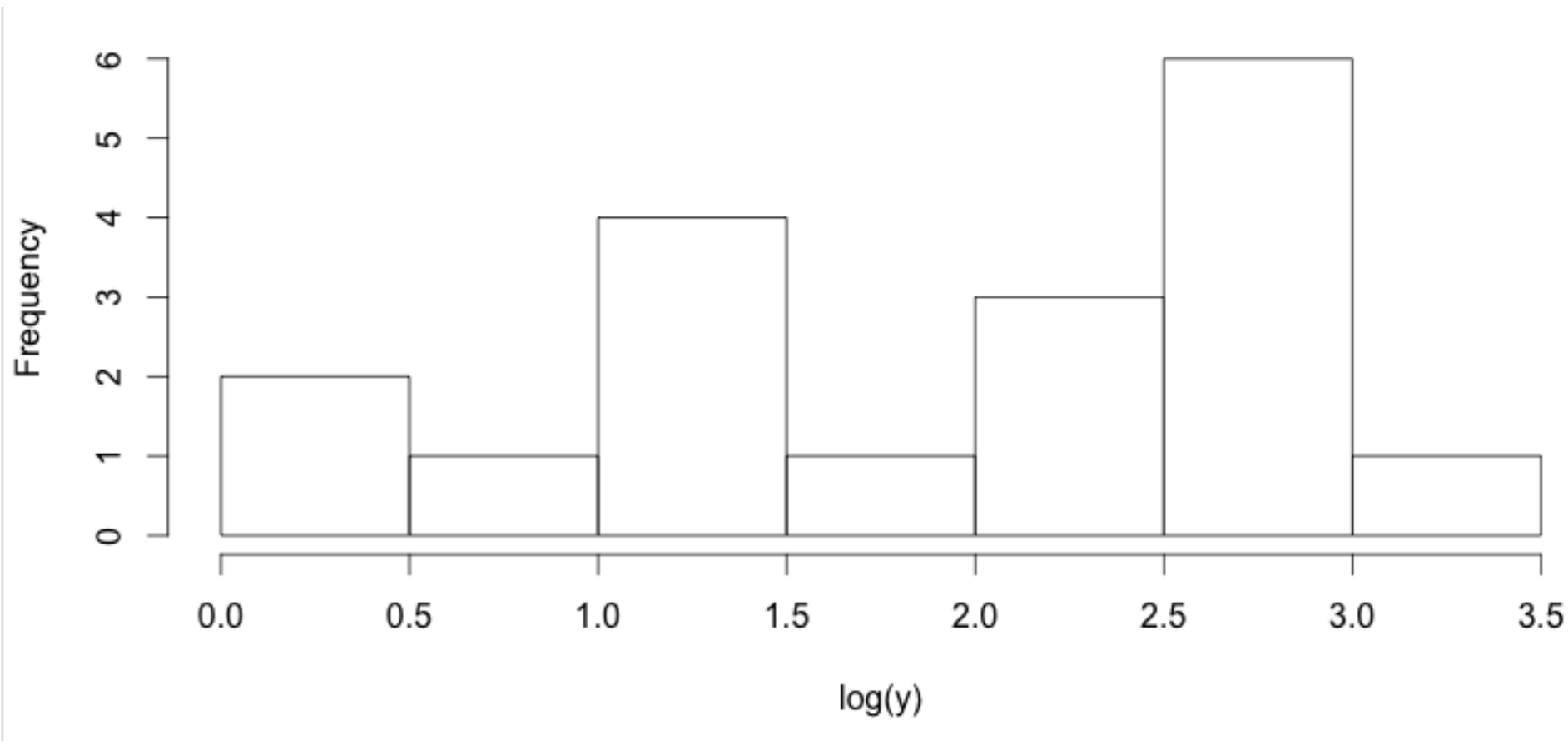
```
hist(y)
```



Basic Visualization III.i

```
y=c(1,1,2,3,3,4,4,5,10,11,11,14,15,16,16,16,20,32)
```

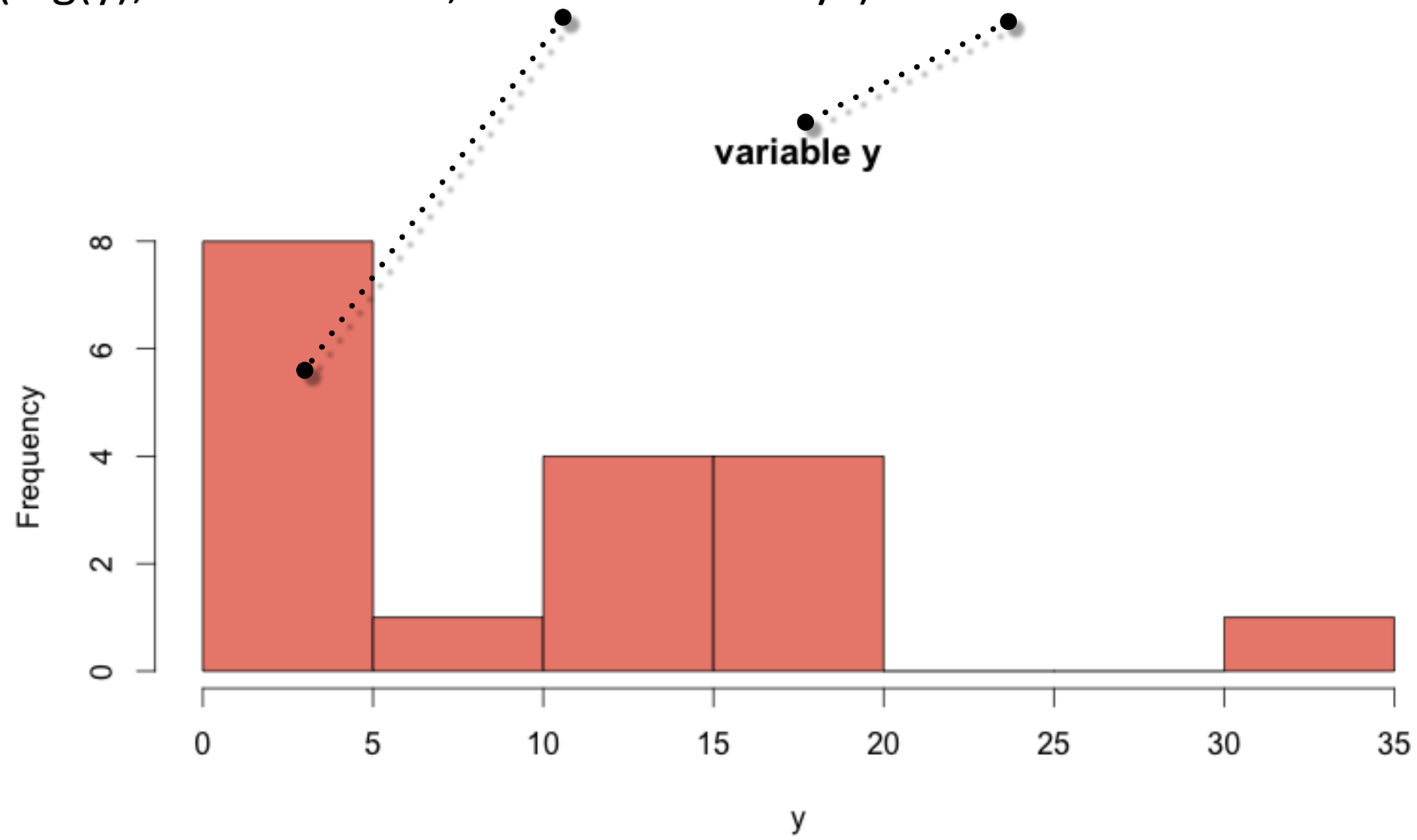
```
hist(log(y))
```



Basic Visualization III.ii

y=c(1,1,2,3,3,4,4,5,10,11,11,14,15,16,16,16,20,32)

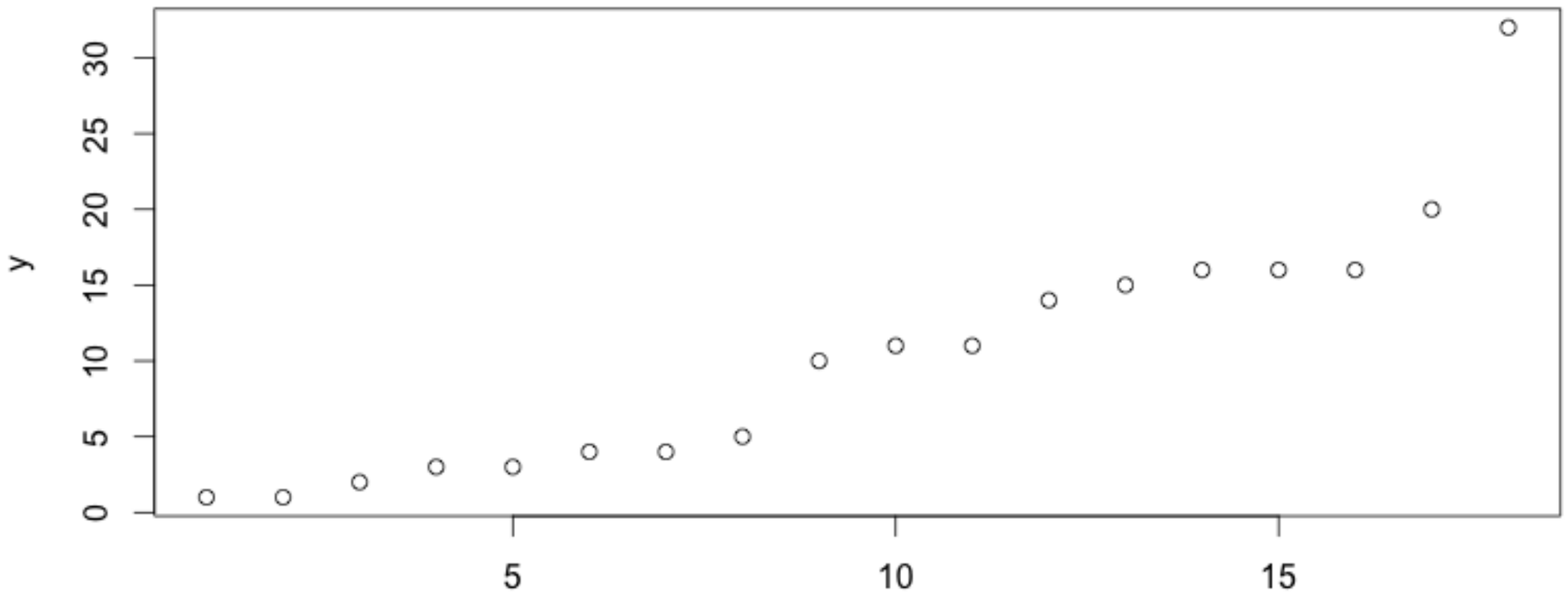
hist(log(y), col="salmon", main="variable y")



Basic Visualization IV

```
y=c(1,1,2,3,3,4,4,5,10,11,11,14,15,16,16,16,20,32)
```

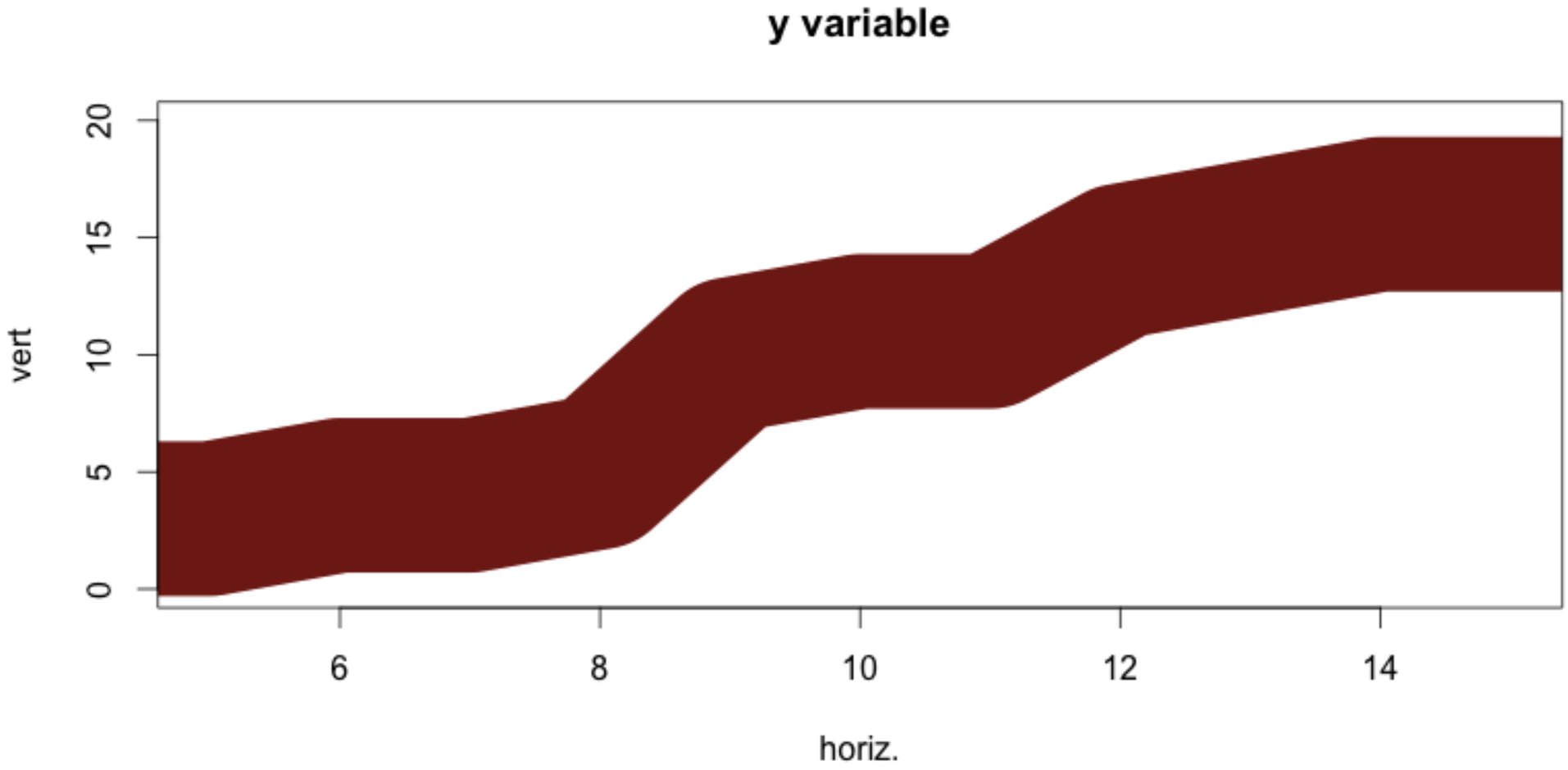
```
plot(y)
```



Basic Visualization IV.ii

y=(1,1,2,3,3,4,4,5,10,11,11,14,15,16,16,16,20,32)

```
plot(y, type="l", col="dark red", lwd=100, main="y variable", ylim=c(0,20),
xlim=c(5,15), ylab="vert", xlab="horiz.")
```



Help

- Do you need to remember all of the variables?
- ? is your friend
- ?plot

plot {graphics}

Generic X-Y Plotting

Description

Generic function for plotting of R objects. For more details about the graphical parameter arguments, see par.

For simple scatter plots, plot.default will be used. However, there are plot methods for many R objects, includi
functions, data.frames, density objects, etc. Use methods(plot) and the documentation for these.

Usage

```
plot(x, y, ...)
```

R Documentation

`type`

what type of plot should be drawn. Possible types are

- "p" for **p**oints,
- "l" for **l**ines,
- "b" for **b**oth,
- "c" for the lines part alone of "b",
- "o" for both '**o**verplotted',
- "h" for '**h**istogram' like (or 'high-density') vertical lines,
- "s" for stair **s**teps,
- "S" for other **s**teps, see 'Details' below,
- "n" for no plotting.

Data Frames

https://is.gd/bacteriacsv

read.csv

- A data.frame is essentially a table
- rows can contain mixed types
  - numeric, text strings
- cols must contain same type

	clostridia	proteobacteria	bacteroides
01_healthy	22	54	245
02_healthy	26	65	265
03_healthy	34	66	262
01_sick	32	32	116
02_sick	12	24	101
03_sick	9	18	87

---

data.frame[-1,-2]

row

column

	clostridia	bacteroides
02_healthy	26	265
03_healthy	34	262
01_sick	32	116
02_sick	12	101
03_sick	9	87

Data Frame Manipulations

https://is.gd/bacteriacsv

	clostridia	proteobacteria	bacteroides
01_healthy	22	54	245
02_healthy	26	65	265
03_healthy	34	66	262
01_sick	32	32	116
02_sick	12	24	101
03_sick	9	18	87

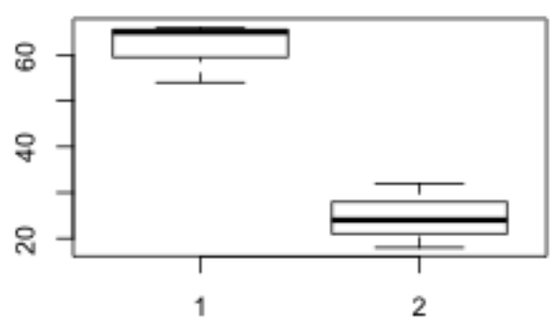
```
bac$proteobacteria
```

```
[1] 54 65 66 32 24 18
```

```
t.test(bac$proteobacteria[1:3], bac$proteobacteria[4:6])
```

p-value = 0.002725

```
boxplot(bac$proteobacteria[1:3],
        bac$proteobacteria[4:6])
```



Getting Help in R

?write.table

Description

`write.table` prints its required argument `x` (after converting it to a data frame if it is not one nor a matrix) to a file or connection.

Usage

```
write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ",
            eol = "\n", na = "NA", dec = ".", row.names = TRUE,
            col.names = TRUE, qmethod = c("escape", "double"),
            fileEncoding = "")

write.csv(...)
write.csv2(...)
```

Arguments

`x`	the object to be written, preferably a matrix or data frame. If not, it is attempted to coerce `x` to a data frame.
`file`	either a character string naming a file or a connection open for writing. `""` indicates output to the console.
`append`	logical. Only relevant if `file` is a character string. If `TRUE`, the output is appended to the file. If `FALSE`, any existing file of the name is destroyed.

Exercise: try to write out our `bac` data frame to a tab-separated file with no quoting around the data

Getting Help in R

Google is your friend!

R syntax can be weird

At first you don't even know what you don't know!

Don't use stackoverflow answers blindly, but do use them and learn from them!