Introduction to R

Keegan Korthauer

Department of Statistics

UW Madison

What is R?

- An free and open source language and environment for statistical computing and graphics
- Similar to the commercial language and environment ‘S’
- Many common statistical functions are built-in but there are also thousands of user-written packages that can be downloaded
- Widely used in academia for research and teaching
- Also used in the commercial sector: Facebook, Google, National Weather Service, Orbitz, etc.

There’s an R Package for that

https://www.youtube.com/watch?v=yhTerzNFLbo

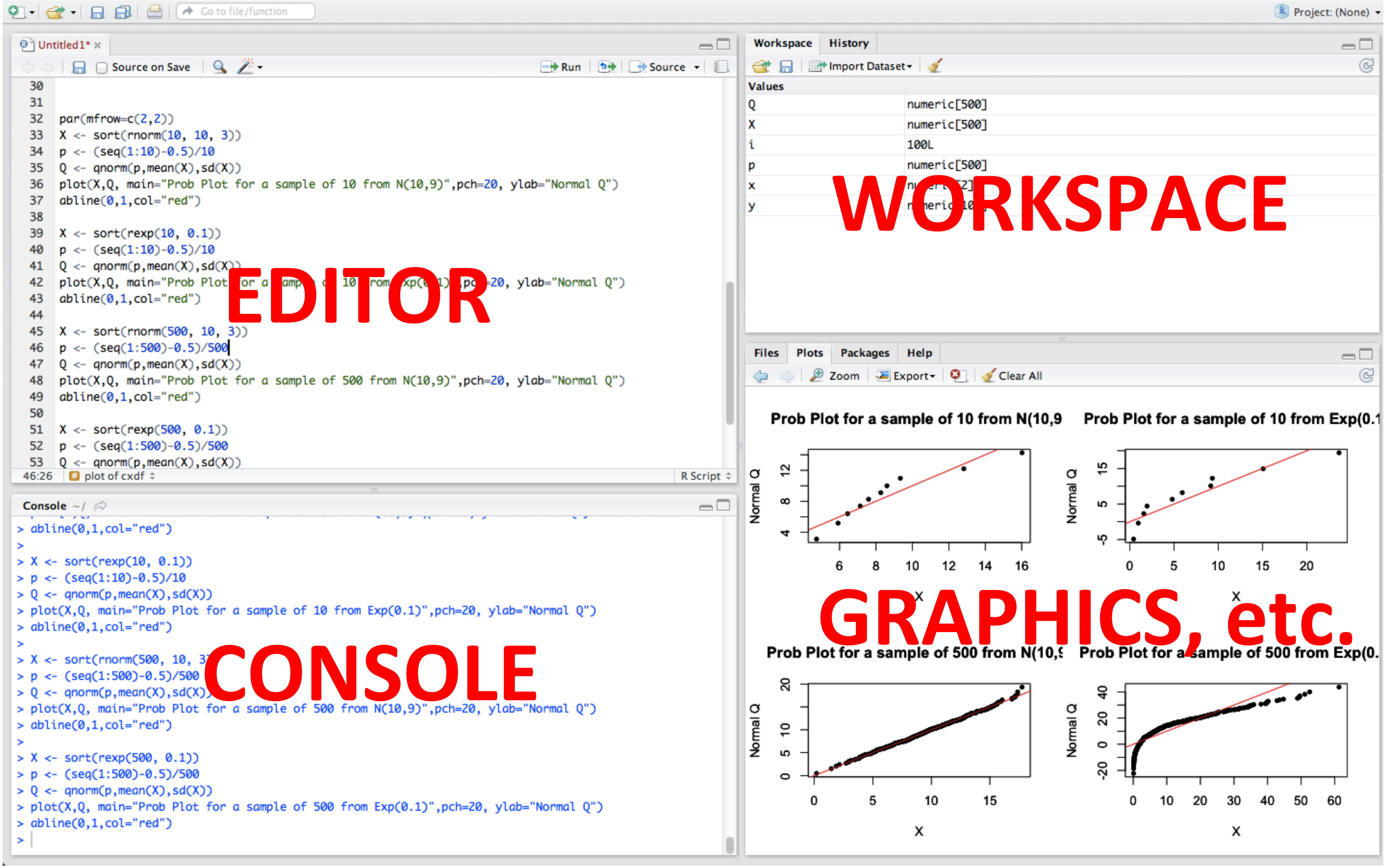
Using R

- R is an interpreted language
  - typically used at the command line, where commands are executed one-by-one
  - similar to MATLAB

- We'll need to download/install two things to get started:
  1. R itself: http://cran.us.r-project.org/
     (choose 'base' for Windows, 'R-3.0.2.pkg' for Mac)
  2. RStudio: http://www.rstudio.com/ide/download/desktop
     (an alternative to running R from a command line; provides a nice, clean graphical user interface)

- Select the appropriate versions of both according to your operating system and follow the instructions for installation

RStudio Interface



```
par(mfrow=c(2,2))
X <- sort(rnorm(10, 10, 3))
p <- (seq(1:10)-0.5)/10
Q <- qnorm(p,mean(X),sd(X))
plot(X,Q, main="Prob Plot for a sample of 10 from N(10,9)",pch=20, ylab="Normal Q")
abline(0,1,col="red")

X <- sort(rexp(10, 0.1))
p <- (seq(1:10)-0.5)/10
Q <- qnorm(p,mean(X),sd(X))
plot(X,Q, main="Prob Plot for a sample of 10 from Exp(0.1)",pch=20, ylab="Normal Q")
abline(0,1,col="red")

X <- sort(rnorm(500, 10, 3))
p <- (seq(1:500)-0.5)/500
Q <- qnorm(p,mean(X),sd(X))
plot(X,Q, main="Prob Plot for a sample of 500 from N(10,9)",pch=20, ylab="Normal Q")
abline(0,1,col="red")

X <- sort(rexp(500, 0.1))
p <- (seq(1:500)-0.5)/500
Q <- qnorm(p,mean(X),sd(X))
```

Values

```
Q    numeric[500]
X    numeric[500]
i    100L
p    numeric[500]
x    numeric[2]
y    numeric[10]
```

```
> abline(0,1,col="red")
>
> X <- sort(rexp(10, 0.1))
> p <- (seq(1:10)-0.5)/10
> Q <- qnorm(p,mean(X),sd(X))
> plot(X,Q, main="Prob Plot for a sample of 10 from Exp(0.1)",pch=20, ylab="Normal Q")
> abline(0,1,col="red")
>
> X <- sort(rnorm(500, 10, 3))
> p <- (seq(1:500)-0.5)/500
> Q <- qnorm(p,mean(X),sd(X))
> plot(X,Q, main="Prob Plot for a sample of 500 from N(10,9)",pch=20, ylab="Normal Q")
> abline(0,1,col="red")
>
> X <- sort(rexp(500, 0.1))
> p <- (seq(1:500)-0.5)/500
> Q <- qnorm(p,mean(X),sd(X))
> plot(X,Q, main="Prob Plot for a sample of 500 from Exp(0.1)",pch=20, ylab="Normal Q")
> abline(0,1,col="red")
>
```

RStudio Layout

- **Console**: where R is actually running; where you put commands
- **Editor**: collections of commands you plan to send to the console; can save them as text files (.txt) or R files (.R)
- **Workspace**: displays data that is currently loaded into memory; click on the 'History' tab for a list of commands you have entered
- **Graphics, etc**: displays any plots you have made; the other tabs allow you to open other files, install add-on packages, and read the help files

Simple R Commands

- R as a calculator

  `+, -, *, /, ^` operate as you would expect

- Various mathematical functions

  `log()` : natural logarithm

  `exp()` : exponential function

  `sqrt()` : square root

  `abs()` : absolute value

  `choose(n,k)` : # of ways to choose k items from n

Try it out

```
> # this is a comment
>
> # try out R as a calculator
> 8-5
[1] 3
> sqrt(144) + 3^2
[1] 21
> 5*89 - log(306)
[1] 439.2764
> choose(10,4)
[1] 210
```

```
> # create some variables
> a = 10
> b = 18
> x = c(a,b,9)
>
> # print the variables
> a
[1] 10
> b
[1] 18
> x
[1] 10 18  9
```

Script with these commands posted on Learn@UW – with Lecture notes

Basic Functions

- To find the mean and variance of 5 numbers, we could do this:

```
> (5+9+3+4+2)/5
[1] 4.6
> ((5^2+9^2+3^2+4^2+2^2) – 5*4.6^2)/4
[1] 7.3
```

- With a very long vector it is more convenient to do this:

```
> x = c(5,9,3,4,2)
> mean(x)
[1] 4.6
> var(x)
[1] 7.3
```

Built-in functions

Help Files

- Help files contain information about built-in functions
  - input arguments & their defaults
  - output values
  - description of what it does
  - examples
  - who wrote it, etc…
- To see the help file for a function, use the **`help()`** command
- For example, try

```
> help(mean)
> help(sd)
```

Basic Graphics

- Built-in functions exist for many types of graphical summaries
- For a data vectors `x` and `y`
  - histogram: `hist(x)`
  - box plot: `boxplot(x)`
  - scatterplot: `plot(x,y)`
- All of these commands will use the default settings; to add a title, change axes labels, add colors, etc. refer to help files to change the optional input arguments
- To save a plot, click on ‘Export’ in the RStudio Graphics window pane

Try it out

```
> # Generate some plots
>
> # first let's get a vector of data (random sample of 20 from standard normal)
> x <- rnorm(20)
> y <- rnorm(20)
>
> # plot a histogram, boxplot, and scatterplot using all defaults
> hist(x)
> boxplot(x)
> plot(x,y)
```

```
> # create a density histogram (instead of frequency) with 4 bars (instead of default)
> hist(x, freq=FALSE, breaks=4)
>
> # create a scatter plot with blue points (instead of black circles)
> plot(x,y, col="blue", pch=20)
```

*Note that the output here is sent to the graphics console

The `pnorm` Function

- Evaluates the left-tail areas of the normal probability density function without the standard normal table:

```
pnorm(q, mean=0, sd=1, lower.tail=TRUE)
```

- Where `q` is the quantile (or z-score) you wish to integrate up to
- Leave all other arguments default if using standard normal, or else specify the mean and standard deviation
- To get the right-tail instead, input `lower.tail=FALSE`

The `pbinom` Function

- Evaluates the left-tail areas of the binomial probability mass function:

```
pbinom(q, size, prob, lower.tail=TRUE)
```

- Where `q` is the quantile you wish to sum up to, `size` is the parameter n and `prob` is the parameter p
- Gives probability less than *or equal to* (so the interval is **inclusive** of `q`)
- To get the right-tail instead, input `lower.tail=FALSE`

The `ppois` Function

- Evaluates the left-tail areas of the binomial probability mass function:

```
ppois(q, lambda, lower.tail=TRUE)
```

- Where `q` is the quantile you wish to sum up to and `lambda` is the rate parameter λ
- Gives probability less than *or equal to* (so the interval is **inclusive** of `q`)
- To get the right-tail instead, input `lower.tail=FALSE`

The `qnorm` Function

- Like the ‘reverse table lookup’ – gives the quantile of the normal distribution for a given left-tail area

```
qnorm(p, mean=0, sd=1, lower.tail=TRUE)
```

- Where `p` is area to the left of the quantile you wish to solve for
- Leave all other arguments default if using standard normal, or else specify the mean and standard deviation
- When `p` corresponds to the right-tail instead, input `lower.tail=FALSE`

Try it out – pnorm & qnorm

```
# pnorm - CDF of Normal Distribution
# find area to the left of zero for standard normal
pnorm(0)

# find area to the right of 3 for mean 2, sd 2
pnorm(3, mean=2, sd=2, lower.tail=FALSE)
# or
1 - pnorm(3, mean=2, sd=2)

# qnorm – Inverse CDF of Normal Distribution
# find the quantile of the standard normal where the left-tail
# area is 0.025
qnorm(0.025)
```

*Note that the output of the commands is not shown here – this is just the script

Try it out – pbinom

```
# pbinom - CDF of Binomial Distribution
# find P(X>8) for X~Bin(50,0.15)
1-pbinom(8, size=50, prob=0.15)
# or
pbinom(8, size=50, prob=0.15, lower.tail=FALSE)

# if we wanted P(X>=8) (so 8 is included in the interval) for
# X~Bin(50,0.15)
1-pbinom(7, size=50, prob=0.15)

# find P(X=3) for X~Bin(10,0.5)
pbinom(3, size=10, prob=0.5) - pbinom(2, size=10, prob=0.5)
# without using pbinom
choose(10,3)*(0.5)^5*(0.5)^5
```

*Note that the output of the commands is not shown here – this is just the script

Try it out – ppois

```
# ppois - CDF of Poisson Distribution
# find P(X>4) for X~Poisson(2)
1-ppois(4, lambda=2)
# or
ppois(4, lambda=2, lower.tail=FALSE)

# find P(X=6) for X ~ Poisson(4)
exp(-4)*4^(6)/factorial(6)
# without using ppois
ppois(6,lambda=4)-ppois(5,lambda=4))
```

*Note that the output of the commands is not shown here – this is just the script

Resources

- On the web
  - A (Very Short) Introduction to R by by Paul Torfs & Claudia Brauer
  - An Introduction to R by W. N. Venables, D. M. Smith and the R Core Team
  - Google/RWeb
- Through the UW Library System
  - R for Dummies by Andrie de Vries, Joris Meys (ebook)
  - Data Analysis and Graphics Using R by John Maindonald, W. John Braun (e-book)

Next

- Review for Exam 1