S&DS 101
Intro Statistics: Life Sciences

Overview

R continued

- R Markdown, objects, vectors, and rbinom()

Log in to R Studio Cloud

Join the class group: bit.ly/SDS101

Sign into workspace 2:
https://rstudio.cloud/spaces/37222/project/684026

Or download and install R Studio

RMarkdown

RMarkdown (.Rmd files) allow you to embed written descriptions, R code and the output of that code into a nice looking document

Creates a way to do reproducible research!

Boot up R Studio to follow along:

- Either on your own computer or on R Studio Cloud: https://rstudio.cloud/spaces/25704/project/481362

RMarkdown

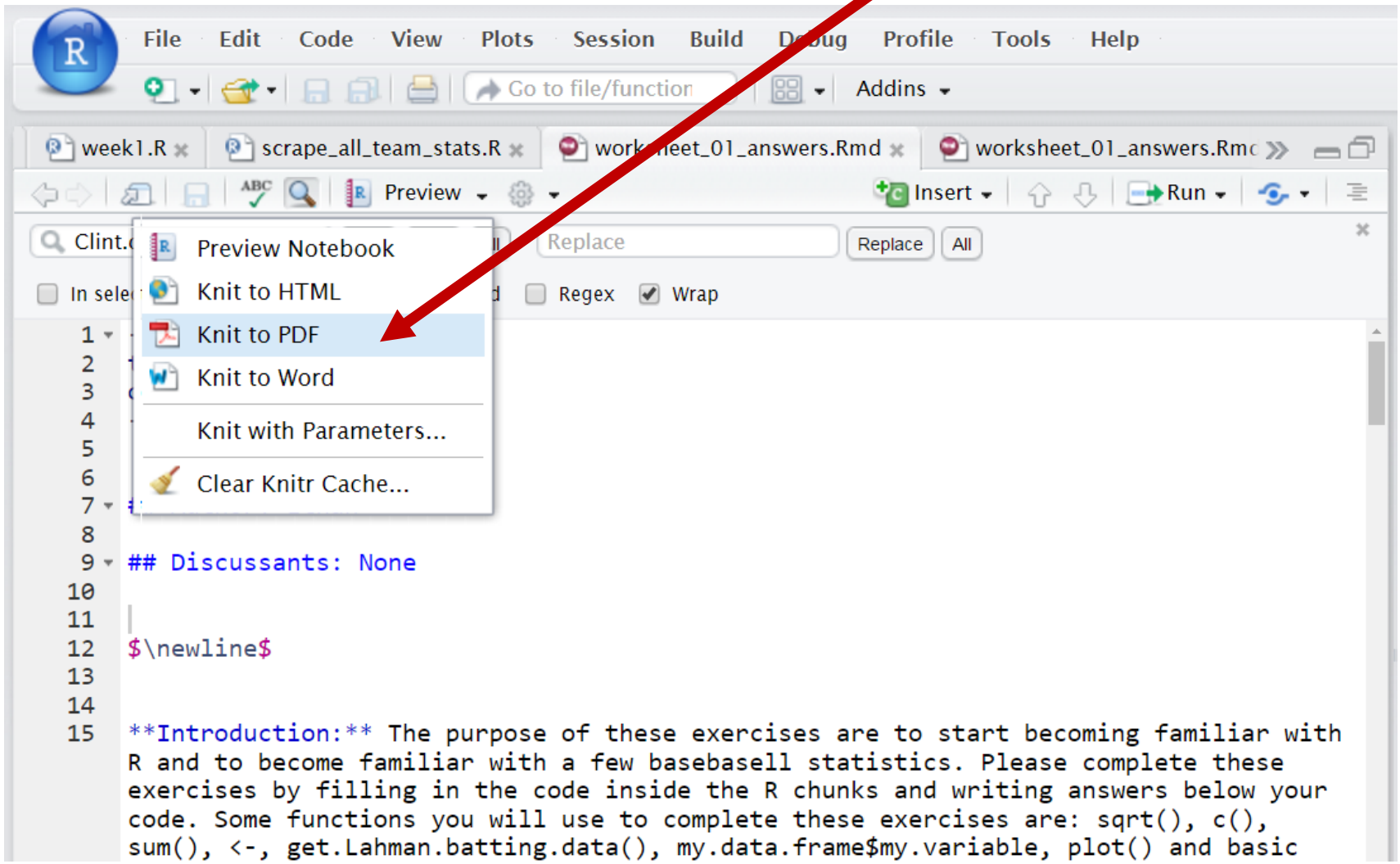
Everything in R chunks is executed as code:

```{r}
    # this is a comment
    # the following code will be executed
    2 + 3
```

Everything outside R chunks appears as text

Knitting to a pdf

Turn in a pdf or html document with your solutions to Canvas



RMarkdown

Note: When you knit, RMarkdown files **do not have access to variables in the global environment**, but instead have their own environment.

Why is this a good thing???

Formatting in R Markdown

We can add formatting to text outside the code chunks

Examples:

```
## Level 2 header
**bold**
```

LaTeX in R Markdown

We can add LaTeX symbols to documents using $\symbol$ syntax

For example, try these:

θ

$\bar{x}$

$\hat{\theta}$

Knit early and knit often to avoid errors!!!

LaTeX in R Markdown

I have added a link on Canvas in the resources section to help find LaTeX symbols

How else could you get help to learn more about LaTeX symbols?


HELPDESK CAT SAYS
GOOGLE IS YOUR FRIEND!
memegenerator.net

To avoid hard to debug code!

Only change a few lines at a time and then knit your document to make sure everything is working!

If you document isn't knitting:

- **For code:** use the # symbol until you can find the line of code that is giving the error message
- **For syntax:** cut part of the document until it knits and then paste it back

Question

Q: What kind of grades the pirate get in Introduction to Statistics?

A: High Seas

Q: Worst joke of the semester?

A: Not likely

Back to R coding…

Number journey

```
> a <- 7
> b <- 52
> d <- a * b
> d
[1]  364
```

Character strings and booleans

```
> a <- 7
> s <- "s is a terrible name for an object"
> b <- TRUE

> class(a)
[1] numeric

> class(s)
[1] character
```

Functions

Functions use parenthesis: functionName(x)

```
> sqrt(49)
> tolower("DATA is AWESOME!")
```

To get help

```
> ? sqrt
```

One can add comments to your code

```
> sqrt(49)    # this takes the square root of 49
```

Vectors

Vectors are ordered sequences of numbers or letters

The c() function is used to create vectors

```
> v  <-  c(5, 232, 5, 543)
> s  <-  c("statistics", "data", "science", "fun")
```

One can access elements of a vector using square brackets []

```
> s[4]        # what will the answer be?
```

We can get multiple elements from a vector too

```
> s[c(1, 2)]
```

Vectors continued

One can assign a sequence of numbers to a vector

```
> z <- 2:10
> z[3]
```

One can test which elements are greater than a value

```
> z > 3
```

Vectors continued

One can also apply functions to vectors

```
> z <- 2:10
> sqrt(z)
> mean(z)
```

Create a vector of elements are greater than a value

```
> the_trues <- z > 3
```

TRUEs are treated as 1's and FALSEs are treated as 0

```
> sum(the_trues)
```

Question

Q: What was the movie, ‘Pirates of the Caribbean’ rated?

A: PG-13

Q: Worst joke of the semester?

A: We are just getting started!

Random numbers from a binomial distribution

We can generate random numbers from a binomial distribution using the rbinom() function, which simulates flipping a coin.

The arguments to the function are:

flip_simulations <- rbinom(num_sims, size, prob)

- **num_sims**: the number of simulations (random numbers) we want to generate
- **size**: the number of coin flips on each simulation (i.e., *n*)
- **prob**: the probability of a "head" on each coin flip (i.e., *p* or π)

Let's generate 100 random numbers k, from P(k; n = 20, π = .75)

Tables and bar plots

Let's generate 100 random numbers k, from P(k; n = 20, π = .75)

```
> flip_simulations <- rbinom(100, 20, .75)
```

We can count how many times we got k heads using:

```
> count_num_heads <- table(flip_simulations)
```

We can plot this as a barplot using

```
> barplot(count_num_heads)
```