R Markdown

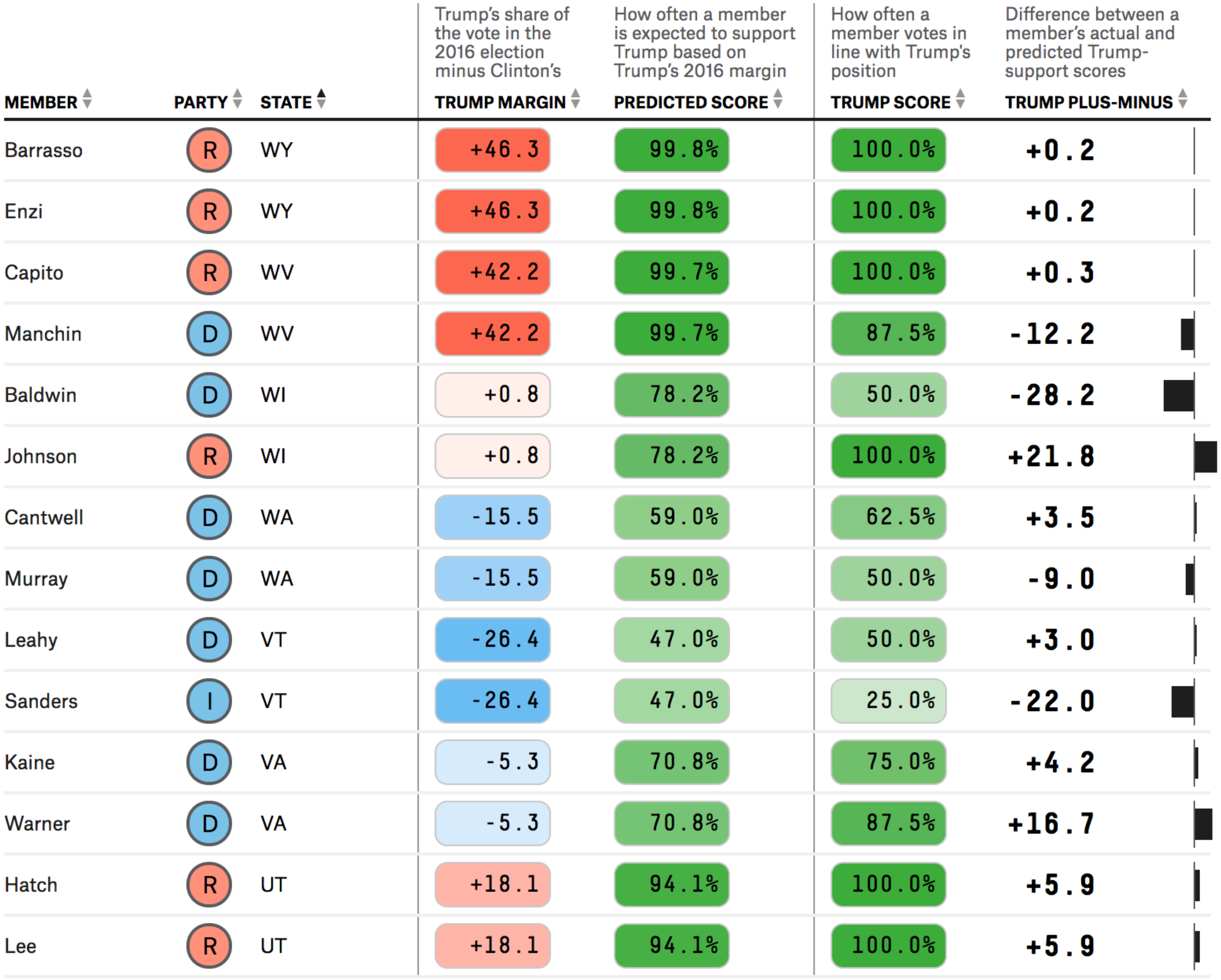
INFO 201

Joel Ross

Winter 2017

https://slides.com/joelross/info201w17-r-markdown/live

Data Report of the Day



MEMBER	PARTY	STATE	Trump's share of the vote in the 2016 election minus Clinton's TRUMP MARGIN	How often a member is expected to support Trump based on Trump's 2016 margin PREDICTED SCORE	How often a member votes in line with Trump's position TRUMP SCORE	Difference between a member's actual and predicted Trump-support scores TRUMP PLUS-MINUS
Barrasso	R	WY	+46.3	99.8%	100.0%	+0.2
Enzi	R	WY	+46.3	99.8%	100.0%	+0.2
Capito	R	WV	+42.2	99.7%	100.0%	+0.3
Manchin	D	WV	+42.2	99.7%	87.5%	-12.2
Baldwin	D	WI	+0.8	78.2%	50.0%	-28.2
Johnson	R	WI	+0.8	78.2%	100.0%	+21.8
Cantwell	D	WA	-15.5	59.0%	62.5%	+3.5
Murray	D	WA	-15.5	59.0%	50.0%	-9.0
Leahy	D	VT	-26.4	47.0%	50.0%	+3.0
Sanders	I	VT	-26.4	47.0%	25.0%	-22.0
Kaine	D	VA	-5.3	70.8%	75.0%	+4.2
Warner	D	VA	-5.3	70.8%	87.5%	+16.7
Hatch	R	UT	+18.1	94.1%	100.0%	+5.9
Lee	R	UT	+18.1	94.1%	100.0%	+5.9

https://projects.fivethirtyeight.com/congress-trump-score/

Today's Objectives

By the end of class, you should be able to

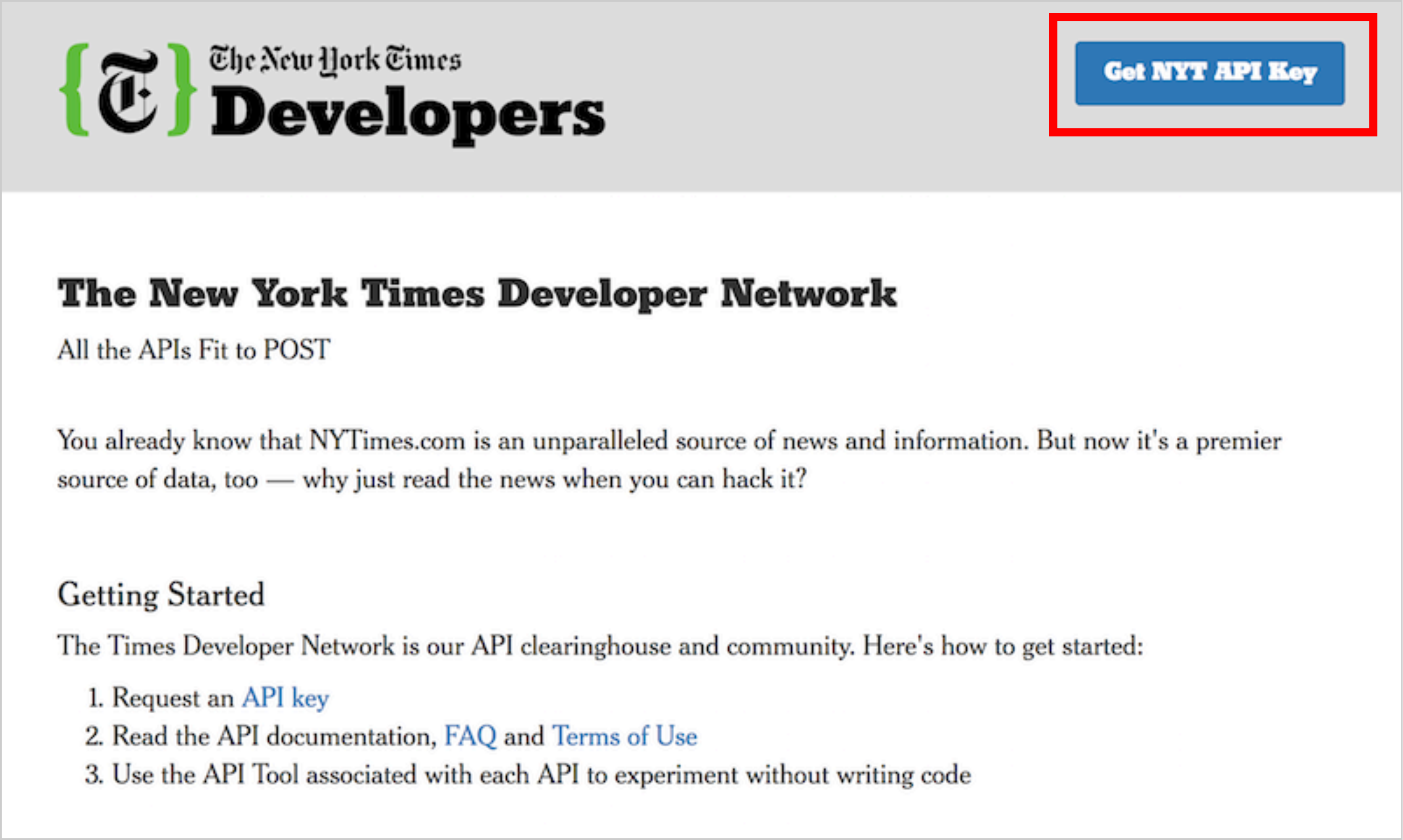
- Comfortably read and access **web APIs**
- Manage **api keys** (access tokens)
- Generate *dynamic reports* with **R Markdown**

API Review

Module 11 exercise-1

Module 11 exercise-2

Today's API



The New York Times Developer Network

All the APIs Fit to POST

You already know that NYTimes.com is an unparalleled source of news and information. But now it's a premier source of data, too — why just read the news when you can hack it?

Getting Started

The Times Developer Network is our API clearinghouse and community. Here's how to get started:

1. Request an API key
2. Read the API documentation, FAQ and Terms of Use
3. Use the API Tool associated with each API to experiment without writing code

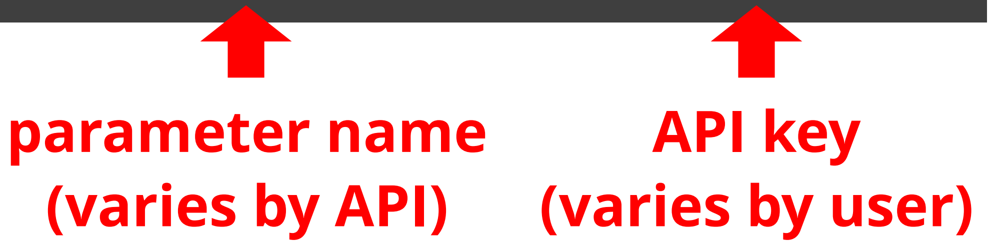
http://developer.nytimes.com/

Access Tokens

An **access token** (or **api key**) is a unique identifier for each developer who uses the API. These are used to control access (like user names or passwords).

Access tokens are usually sent as a **query parameter**:

```
https://api.github.com/user&access_token=12345678abcdefgh
```



parameter name (varies by API)

API key (varies by user)

Managing Access Tokens

Because **access tokens** are like passwords, we don't want to push them to GitHub!

Best practice is to store keys in a ***separate script*** which is added to your `.gitignore` file to avoid being committed.

```
### in apikey.R file ###
my.apikey <- "123456789abcdefg"  # save the key
```

```
### in script.R file ###
# load the script with the key
source("apikey.R")  # makes my.apikey available

# use the key
query.params <- list(access_token = my.apikey)
# ...
```

```
### in .gitignore file ###
apikey.R  # tell git to ignore!
```

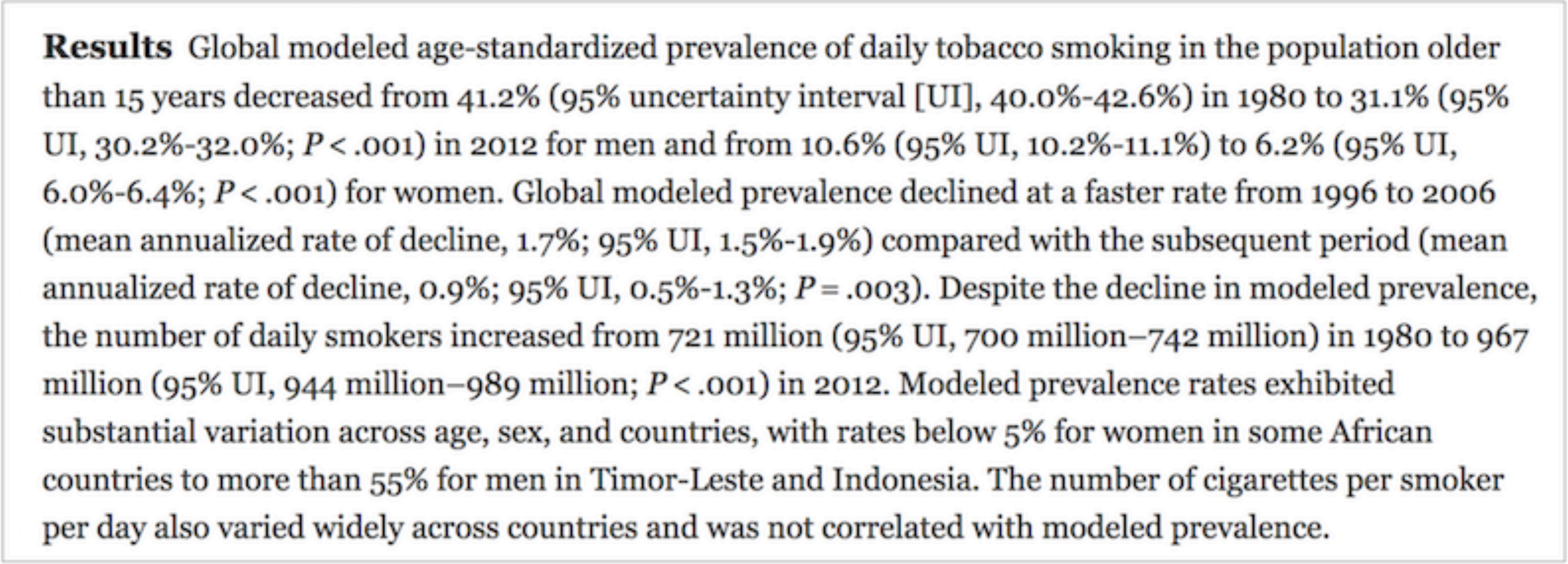
Module 11 exercise-4

Can copy/paste the exercise

CHECK THE README!

Data Presentation

Data Presentation

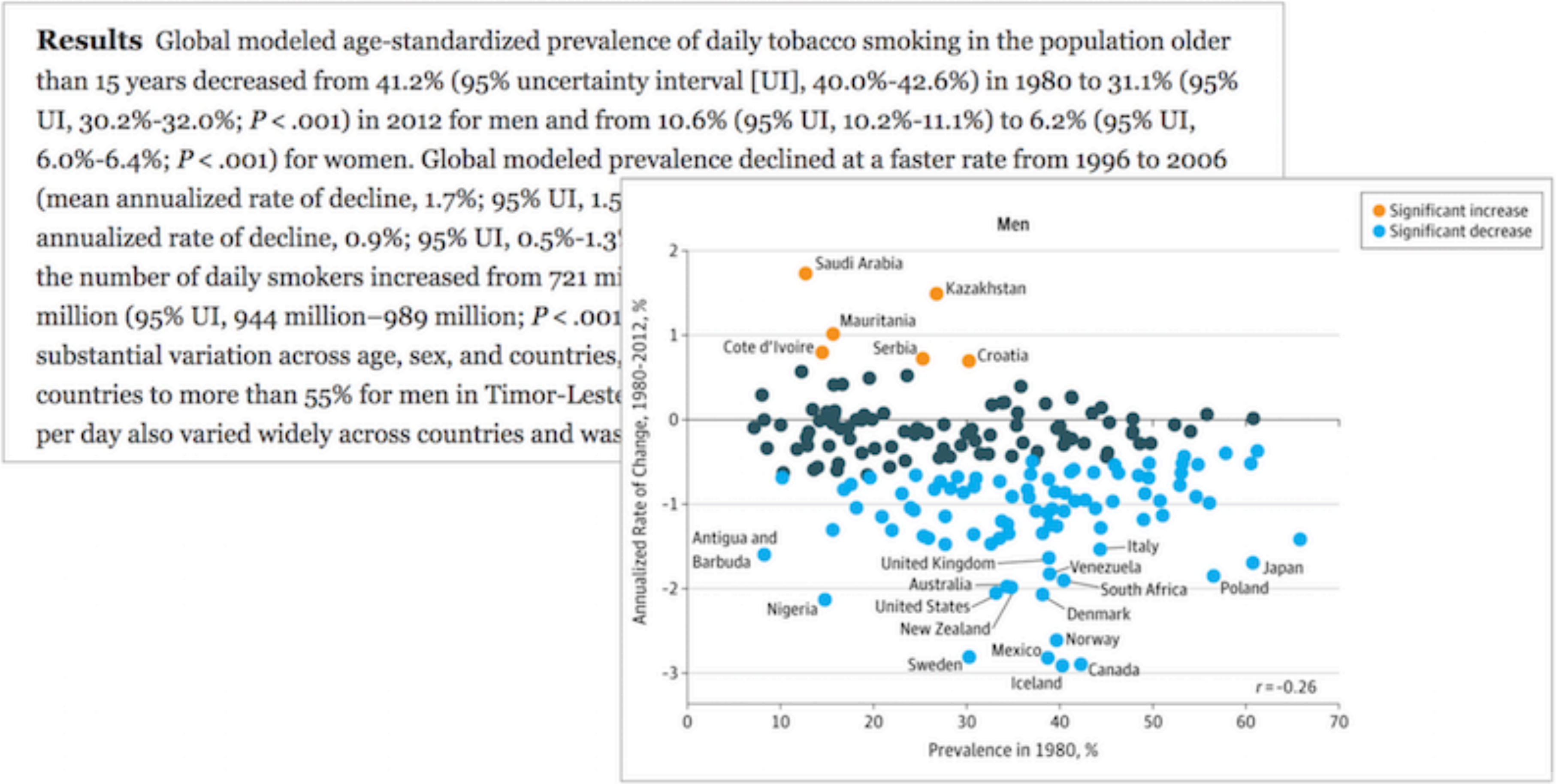


Results Global modeled age-standardized prevalence of daily tobacco smoking in the population older than 15 years decreased from 41.2% (95% uncertainty interval [UI], 40.0%-42.6%) in 1980 to 31.1% (95% UI, 30.2%-32.0%; *P* < .001) in 2012 for men and from 10.6% (95% UI, 10.2%-11.1%) to 6.2% (95% UI, 6.0%-6.4%; *P* < .001) for women. Global modeled prevalence declined at a faster rate from 1996 to 2006 (mean annualized rate of decline, 1.7%; 95% UI, 1.5%-1.9%) compared with the subsequent period (mean annualized rate of decline, 0.9%; 95% UI, 0.5%-1.3%; *P* = .003). Despite the decline in modeled prevalence, the number of daily smokers increased from 721 million (95% UI, 700 million–742 million) in 1980 to 967 million (95% UI, 944 million–989 million; *P* < .001) in 2012. Modeled prevalence rates exhibited substantial variation across age, sex, and countries, with rates below 5% for women in some African countries to more than 55% for men in Timor-Leste and Indonesia. The number of cigarettes per smoker per day also varied widely across countries and was not correlated with modeled prevalence.

How many numbers are in this summary?

Data Reports

Data reports have hundreds (thousands!) of variables, dozens of representations (tables or graphics)



Results Global modeled age-standardized prevalence of daily tobacco smoking in the population older than 15 years decreased from 41.2% (95% uncertainty interval [UI], 40.0%-42.6%) in 1980 to 31.1% (95% UI, 30.2%-32.0%; $P < .001$) in 2012 for men and from 10.6% (95% UI, 10.2%-11.1%) to 6.2% (95% UI, 6.0%-6.4%; $P < .001$) for women. Global modeled prevalence declined at a faster rate from 1996 to 2006 (mean annualized rate of decline, 1.7%; 95% UI, 1.5

annualized rate of decline, 0.9%; 95% UI, 0.5%-1.3

the number of daily smokers increased from 721 mi

million (95% UI, 944 million–989 million; $P < .001$

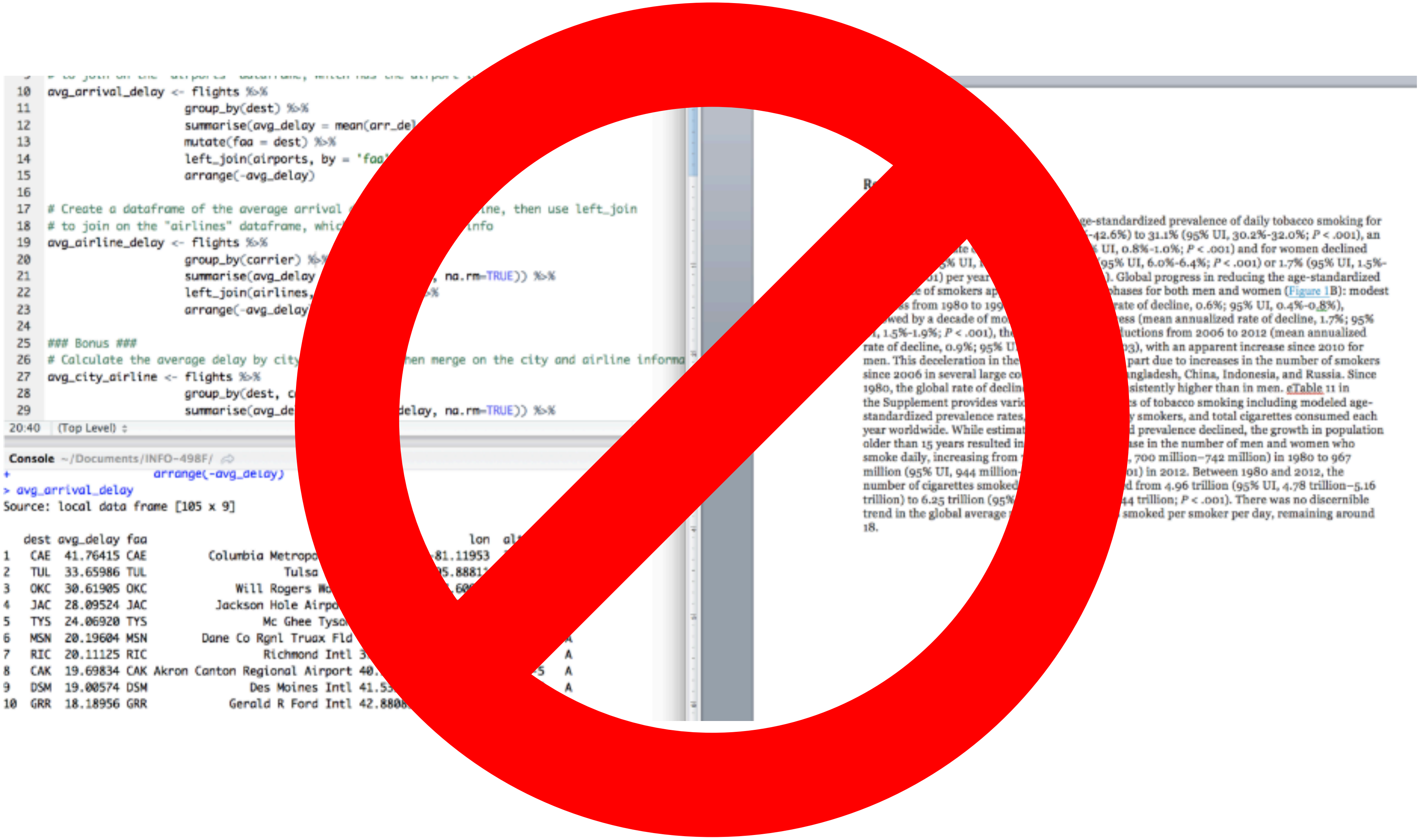
substantial variation across age, sex, and countries,

countries to more than 55% for men in Timor-Leste

per day also varied widely across countries and was

How can we update our report when the data or analysis changes?

Copy and Paste?



```
avg_arrival_delay <- flights %>%
                     group_by(dest) %>%
                     summarise(avg_delay = mean(arr_de
                     mutate(faa = dest) %>%
                     left_join(airports, by = 'faa'
                     arrange(-avg_delay)

# Create a dataframe of the average arrival                   ine, then use left_join
# to join on the "airlines" dataframe, whic                   info
avg_airline_delay <- flights %>%
                     group_by(carrier) %>%
                     summarise(avg_delay                      , na.rm=TRUE)) %>%
                     left_join(airlines,                      %
                     arrange(-avg_delay

### Bonus ###
# Calculate the average delay by city                         hen merge on the city and airline informa
avg_city_airline <- flights %>%
                     group_by(dest, c
                     summarise(avg_de                         delay, na.rm=TRUE)) %>%
```

20:40 (Top Level)

Console ~/Documents/INFO-498F/

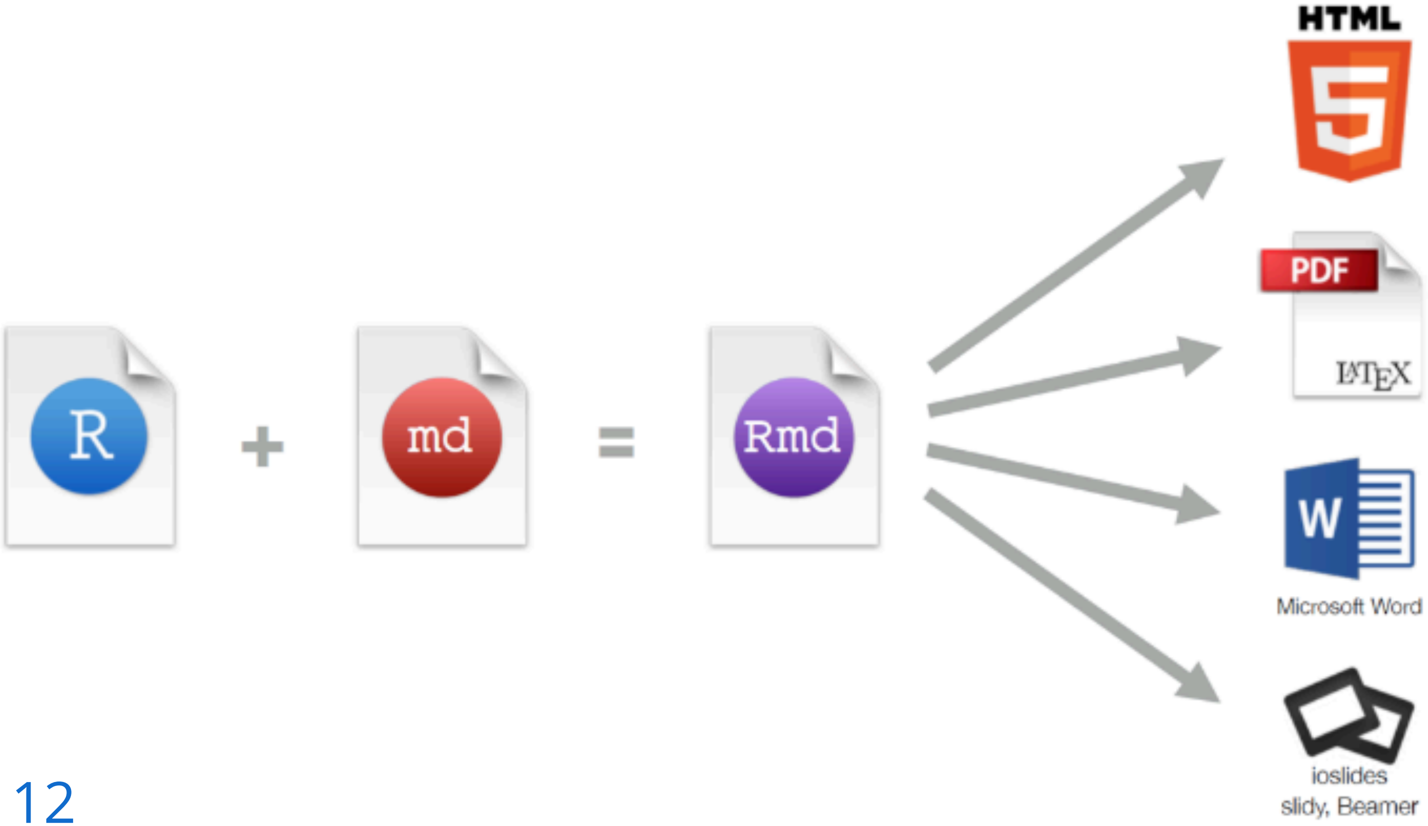
```
+                    arrange(-avg_delay)
> avg_arrival_delay
Source: local data frame [105 x 9]

   dest avg_delay faa                                          lon  al
1   CAE  41.76415 CAE       Columbia Metropo                  -81.11953
2   TUL  33.65986 TUL                  Tulsa                   95.8881
3   OKC  30.61905 OKC         Will Rogers Wo                      .60
4   JAC  28.09524 JAC        Jackson Hole Airpo
5   TYS  24.06920 TYS              Mc Ghee Tyso
6   MSN  20.19604 MSN       Dane Co Rgnl Truax Fld                       A
7   RIC  20.11125 RIC             Richmond Intl 3                        A
8   CAK  19.69834 CAK Akron Canton Regional Airport 40.               5  A
9   DSM  19.00574 DSM            Des Moines Intl 41.53                   A
10  GRR  18.18956 GRR         Gerald R Ford Intl 42.8808
```

R

ge-standardized prevalence of daily tobacco smoking for -42.6%) to 31.1% (95% UI, 30.2%-32.0%; $P < .001$), an te UI, 0.8%-1.0%; $P < .001$) and for women declined % UI, (95% UI, 6.0%-6.4%; $P < .001$) or 1.7% (95% UI, 1.5%- 01) per year). Global progress in reducing the age-standardized e of smokers ap phases for both men and women (Figure 1B): modest ss from 1980 to 199 rate of decline, 0.6%; 95% UI, 0.4%-0.8%), wed by a decade of mo ess (mean annualized rate of decline, 1.7%; 95% , 1.5%-1.9%; $P < .001$), th ductions from 2006 to 2012 (mean annualized rate of decline, 0.9%; 95% U 03), with an apparent increase since 2010 for men. This deceleration in the part due to increases in the number of smokers since 2006 in several large co angladesh, China, Indonesia, and Russia. Since 1980, the global rate of declin sistently higher than in men. eTable 11 in the Supplement provides vari s of tobacco smoking including modeled age-standardized prevalence rates, y smokers, and total cigarettes consumed each year worldwide. While estimat d prevalence declined, the growth in population older than 15 years resulted in se in the number of men and women who smoke daily, increasing from , 700 million–742 million) in 1980 to 967 million (95% UI, 944 million- 01) in 2012. Between 1980 and 2012, the number of cigarettes smoked d from 4.96 trillion (95% UI, 4.78 trillion–5.16 trillion) to 6.25 trillion (95% 44 trillion; $P < .001$). There was no discernible trend in the global average smoked per smoker per day, remaining around 18.

R Markdown

An R package (framework) for dynamically generating documents from code. Formatted **text**, executed **code**, and displayed **graphics** are seamlessly integrated.

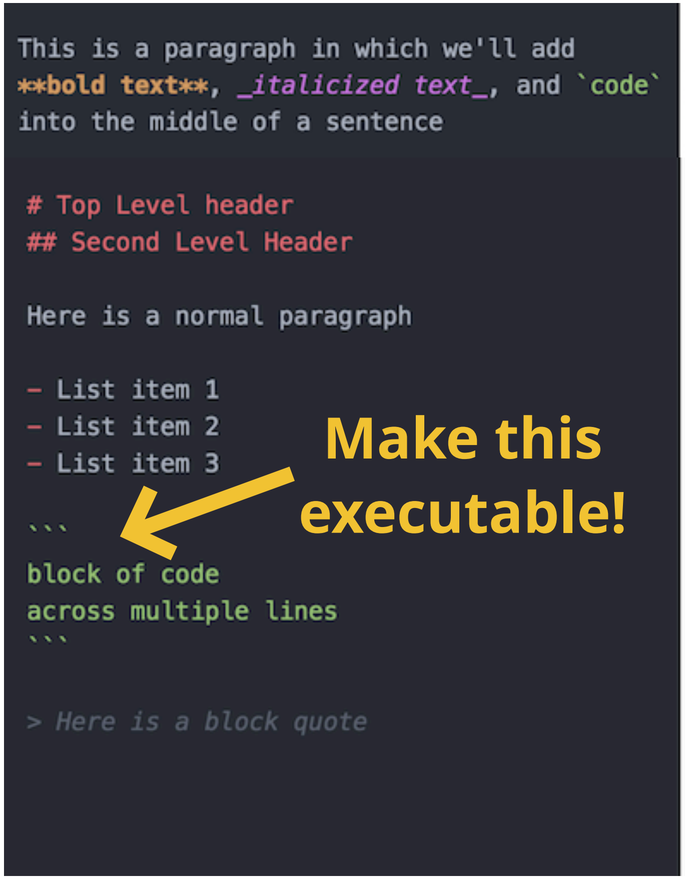


module 12

RECALL

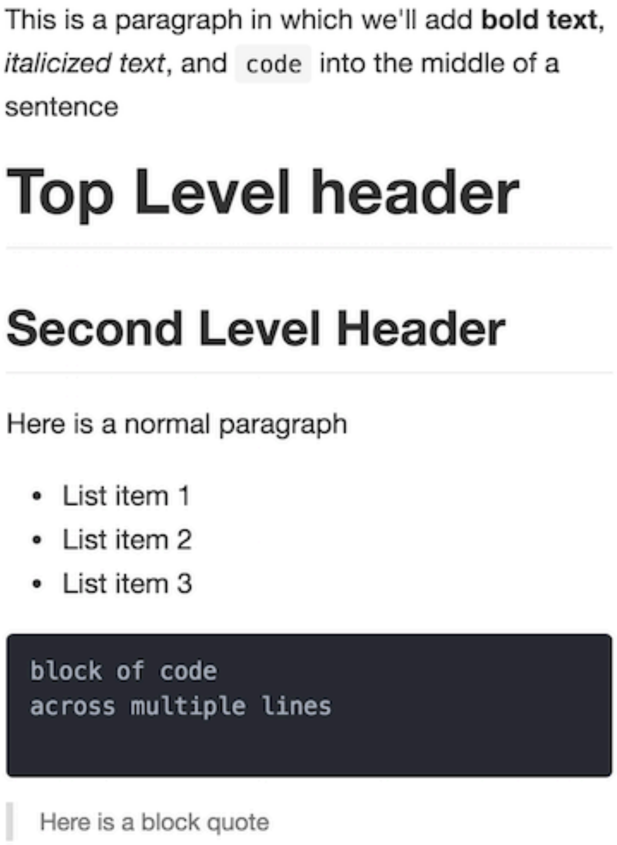
Markdown

Markdown is a simple **syntax** for specifying how plain text should be formatted.



```
This is a paragraph in which we'll add
**bold text**, _italicized text_, and `code`
into the middle of a sentence

# Top Level header
## Second Level Header

Here is a normal paragraph

- List item 1
- List item 2
- List item 3

```
block of code
across multiple lines
```


> Here is a block quote
```

Make this executable!



This is a paragraph in which we'll add **bold text**, *italicized text*, and `code` into the middle of a sentence

Top Level header

Second Level Header

Here is a normal paragraph

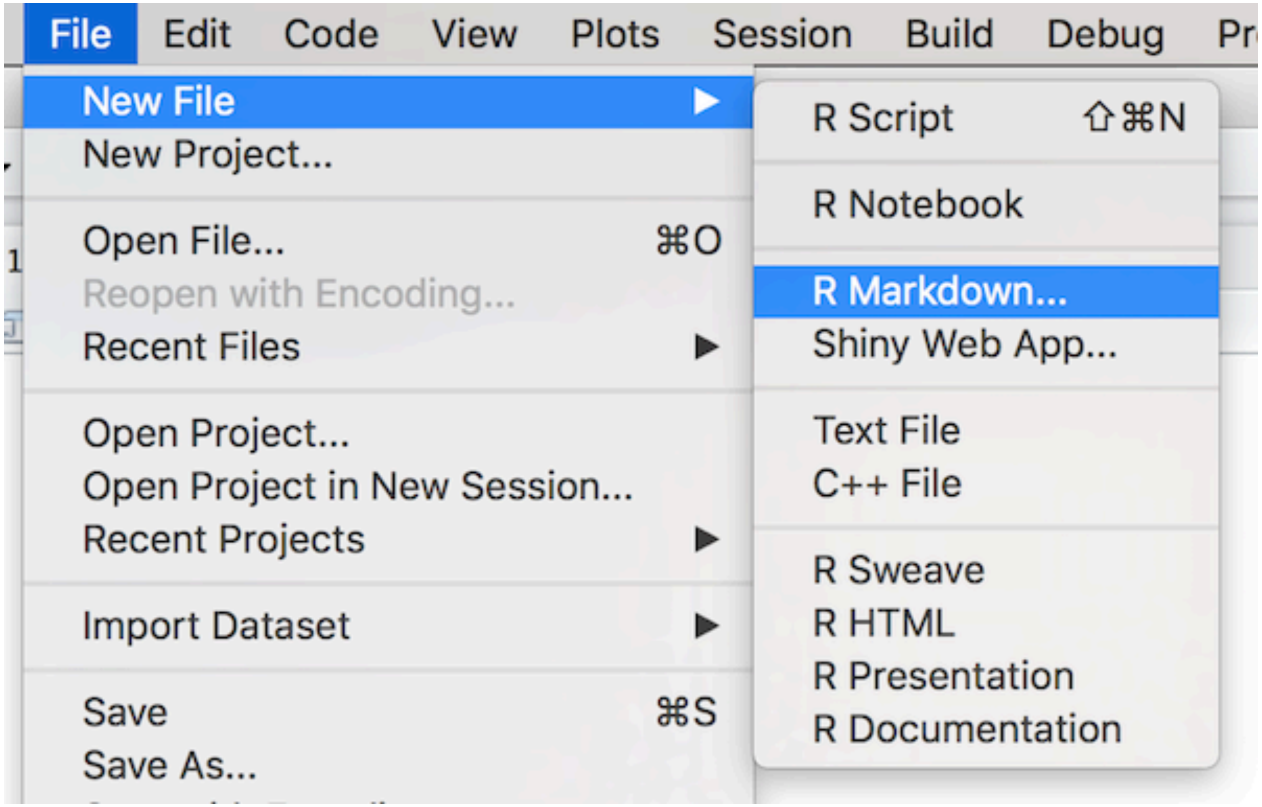
- List item 1
- List item 2
- List item 3

```
block of code
across multiple lines
```

> Here is a block quote

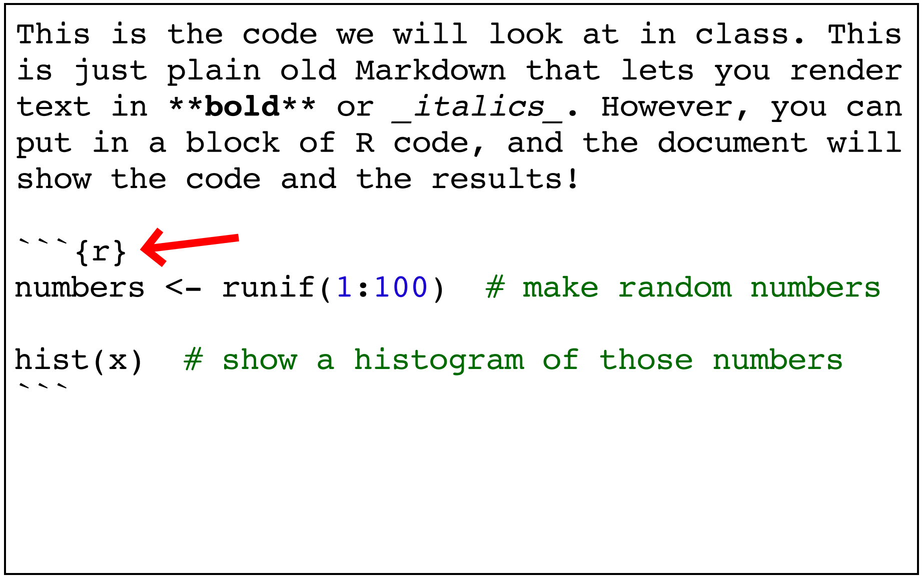
Rmd Files

R Markdown document source code is written in `.Rmd` files. These can easily be created through R Studio.



Markdown and Code

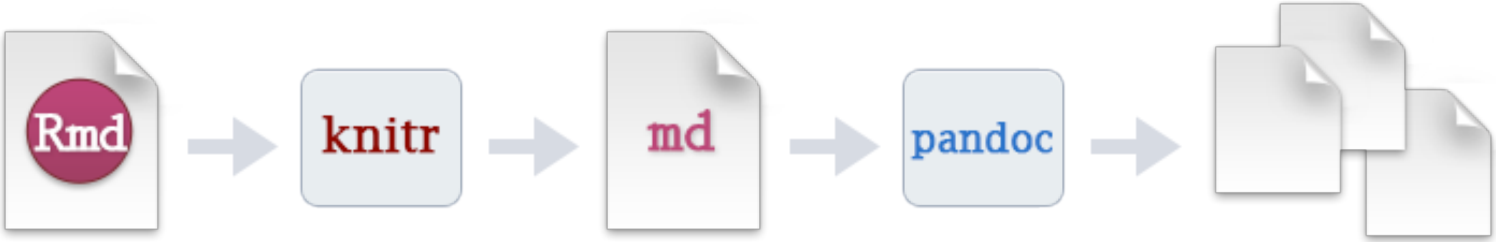
We write Markdown code as normal in the document, but include `{r}` next to code blocks we want to execute!

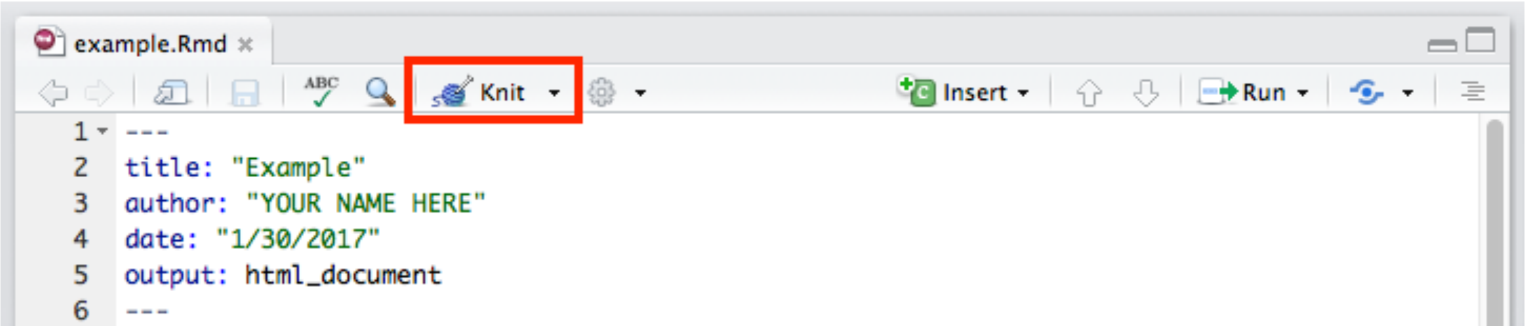


~~~
This is the code we will look at in class. This
is just plain old Markdown that lets you render
text in **bold** or _italics_. However, you can
put in a block of R code, and the document will
show the code and the results!

```{r}
numbers <- runif(1:100)  # make random numbers

hist(x)  # show a histogram of those numbers
```
~~~

Knitting

R Markdown files are converted into readable documents (e.g., HTML) using the `knitr` library. This library handles the code execution and producing the output.

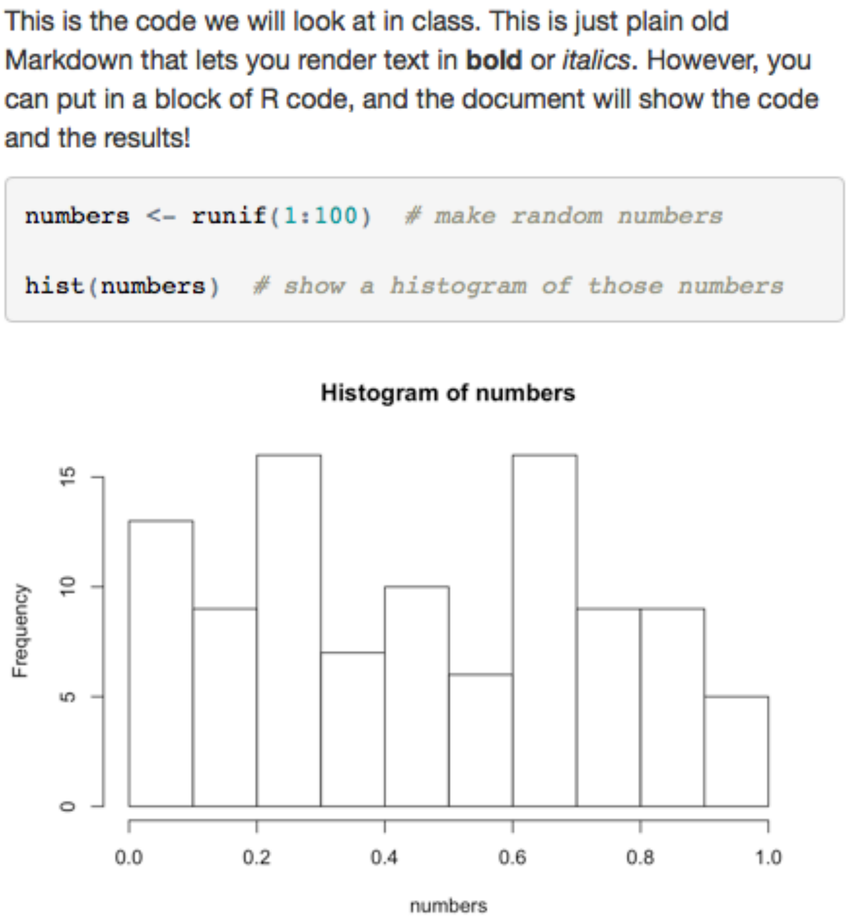




```
---
title: "Example"
author: "YOUR NAME HERE"
date: "1/30/2017"
output: html_document
---
```

Markdown and Code

~~~
This is the code we will look at in class. This
is just plain old Markdown that lets you render
text in **bold** or _italics_. However, you can
put in a block of R code, and the document will
show the code and the results!


```{r}
numbers <- runif(1:100)  # make random numbers

hist(x)  # show a histogram of those numbers
```
~~~



This is the code we will look at in class. This is just plain old Markdown that lets you render text in **bold** or *italics*. However, you can put in a block of R code, and the document will show the code and the results!

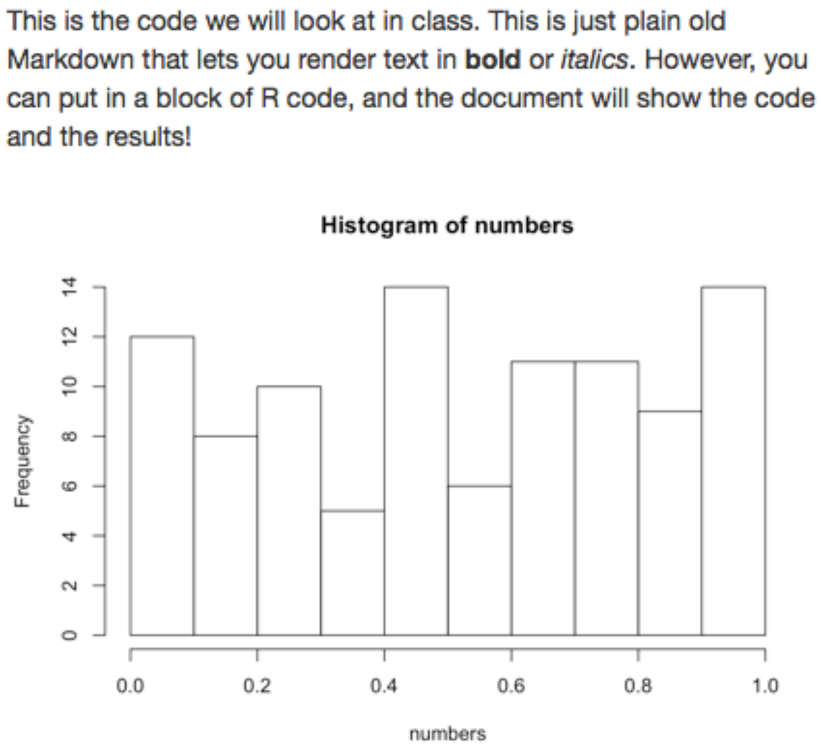
```
numbers <- runif(1:100)  # make random numbers

hist(numbers)  # show a histogram of those numbers
```

knitr Options

Specify options after a comma in the `{r}` to specify what content should be rendered.

~~~
This is the code we will look at in class. This
is just plain old Markdown that lets you render
text in **bold** or _italics_. However, you can
put in a block of R code, and the document will
show the code and the results!


```{r, echo=FALSE}
numbers <- runif(1:100)  # make random numbers

hist(x)  # show a histogram of those numbers
```
~~~

Do not echo (show) the code, just output

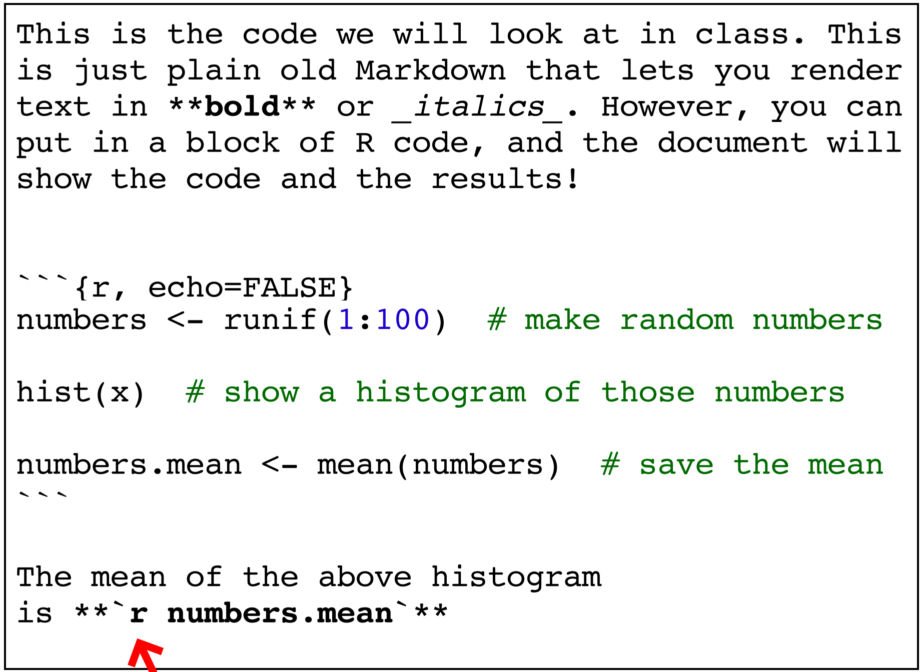


This is the code we will look at in class. This is just plain old Markdown that lets you render text in **bold** or *italics*. However, you can put in a block of R code, and the document will show the code and the results!

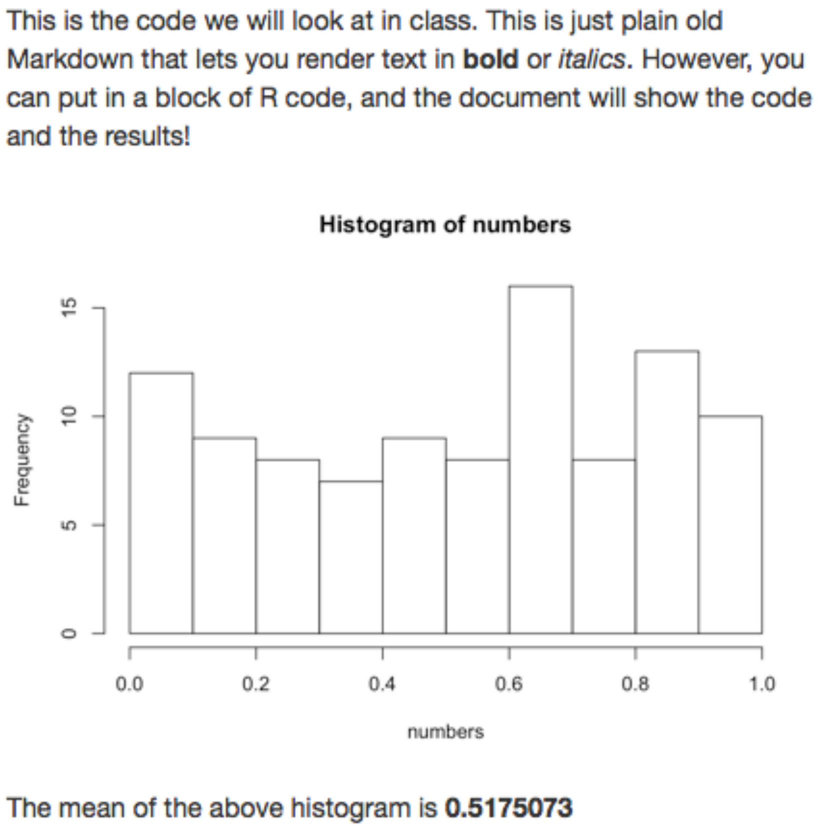
https://yihui.name/knitr/options/

Inline Code

Include expressions (e.g., variables) in **inline** code blocks by prepending them with `r`



~~~
This is the code we will look at in class. This
is just plain old Markdown that lets you render
text in **bold** or _italics_. However, you can
put in a block of R code, and the document will
show the code and the results!


```{r, echo=FALSE}
numbers <- runif(1:100)  # make random numbers

hist(x)  # show a histogram of those numbers

numbers.mean <- mean(numbers)  # save the mean
```


The mean of the above histogram
is **`r numbers.mean`**
~~~



This is the code we will look at in class. This is just plain old Markdown that lets you render text in **bold** or *italics*. However, you can put in a block of R code, and the document will show the code and the results!

The mean of the above histogram is **0.5175073**

Module 12 exercise-1

CHECK THE README!

Questions on anything so far?

Action Items!

- Be comfortable with **module 11-12**
- Assignment 5 due ***Tuesday before class***

Tuesday: Making pretty diagrams!