CHRISTOPHER PRENER, PHD
FALL, 2017

WEEK 02
LECTURE 04

QUANTITATIVE ANALYSIS

STRUCTURING NOTEBOOKS

AGENDA

1. Motivation

2. R Notebooks

3. Markdown

4. Putting it all Together

1 MOTIVATION

ASSIGNING DATA CAN GET CUMBERSOME

```
> library(tidyverse)

> japaneseAutos <- mpg

> japaneseAutos <-
    select(japaneseAutos, model, cty, hwy)

> japaneseAutos <-
    rename(japaneseAutos, cityMpg = cty)

> japaneseAutos <-
    rename(japaneseAutos, hwyMpg = hwy)

> japaneseAutos <-
    filter(japaneseAutos, manufacturer == “honda” |
    manufacturer == “nissan” |
    manufacturer == “subaru” |
    manufacturer == “toyota”)

> japaneseAutos <-
    mutate(japaneseAutos, avgMpg =
    (cityMpg+hwyMpg)/2)

> japaneseAutos <- arrange(japaneseAutos, avgMpg)
```

```
> library(tidyverse)

> mpg %>%
    select(manufacturer,
        model, cty, hwy) %>%
    rename(cityMpg = cty) %>%
    rename(hwyMpg = hwy) %>%
    filter(manufacturer == “honda” |
       manufacturer == “nissan” |
       manufacturer == “subaru” |
       manufacturer == “toyota”) %>%
    mutate(avgMpg =
       (cityMpg+hwyMpg)/2) %>%
    arrange(avgMpg) -> japaneseAutos
```

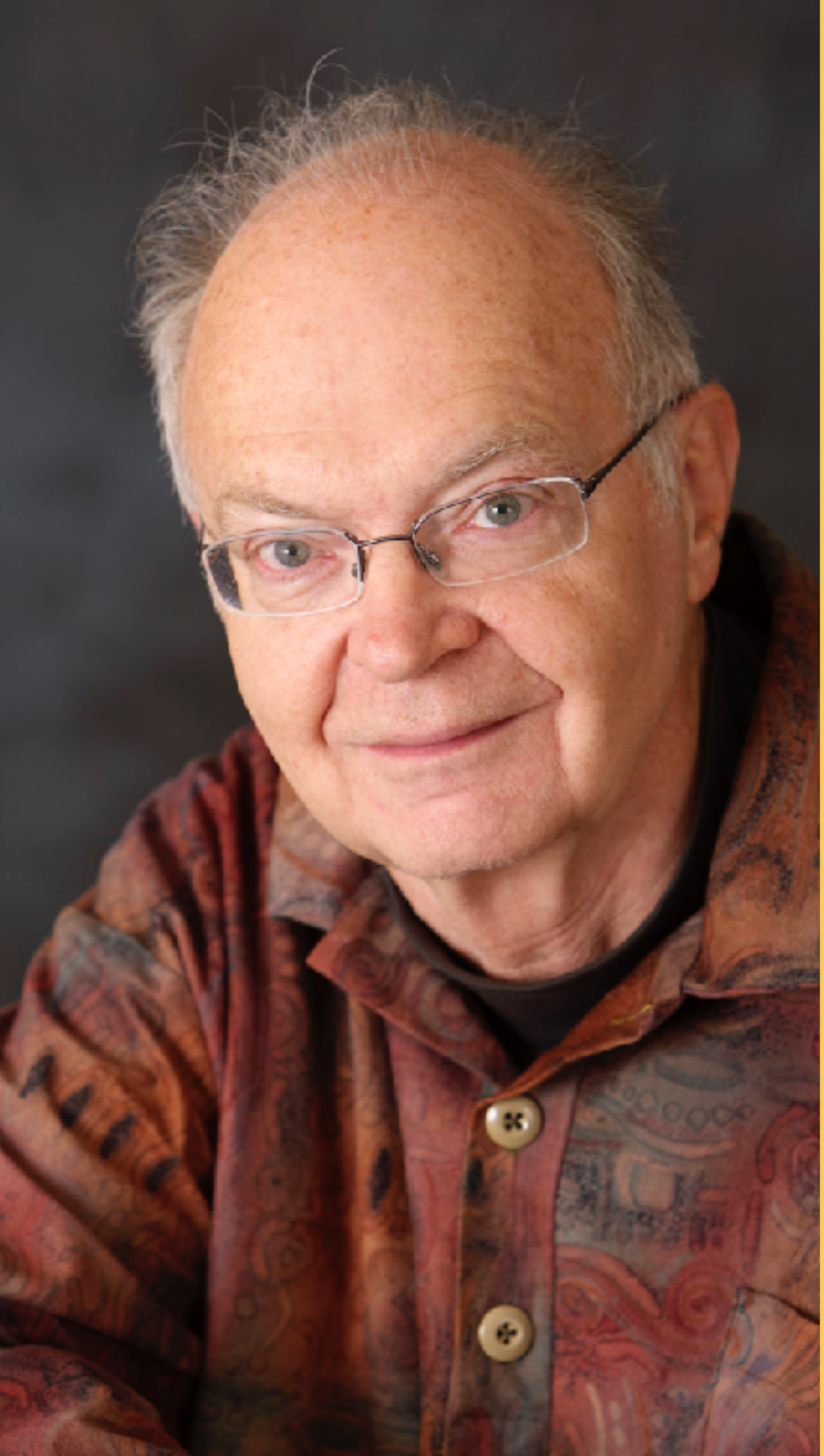
USING THE CONSOLE IS ALSO PROBLEMATIC

- When we type code into the console, it is hard to reuse
- Sometimes it is unavoidable, so knowing how to use the console is important, …
- …but we should look to other tools for writing code that can be saved and easily re-executed.

```
> library(tidyverse)

> mpg %>%
    select(manufacturer,
        model, cty, hwy) %>%
    rename(cityMpg = cty) %>%
    rename(hwyMpg = hwy) %>%
filter(manufacturer == “honda” |
    manufacturer == “nissan” |
    manufacturer == “subaru” |
    manufacturer == “toyota”) %>%
mutate(avgMpg =
    (cityMpg+hwyMpg)/2) %>%
arrange(avgMpg) -> japaneseAutos
```

2 R NOTEBOOKS



LET US CHANGE OUR TRADITIONAL ATTITUDE TO THE CONSTRUCTION OF PROGRAMS: INSTEAD OF IMAGINING THAT OUR MAIN TASK IS TO INSTRUCT A COMPUTER WHAT TO DO, LET US CONCENTRATE RATHER ON EXPLAINING TO HUMANS WHAT WE WANT THE COMPUTER TO DO.

Donald E. Knuth

Stanford University Computer Scientist

KNITR

- `knitr` is an R package designed to implement Knuth's logic
- R code is combined with narrative text that helps give our code more meaning
    - We use narrative to explain the motivation for why our code does what it does
- `knitr` allows us to "weave" code and narrative together and creates dynamic output for us

KNITR

When we combine knitr documents with piped code, we are stepping up to Donald Knuth's challenge to write code that is readable by humans *as well* as machines!

R NOTEBOOKS

week02.Rmd

Preview | Insert | Run

```
---
title: "Week 02 Notebook"
output: html_notebook
---

## Introduction

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

## Dependencies

The examples from week two require two packages - `dplyr` and `ggplot2`. These can be loaded individually or collectively as part of the `tidyverse` package.

To execute this code, click on the green right-facing arrow ("play button") on the right side of the gray code chunk below:

```{r}
library(tidyverse)
```

Example of Piped Code

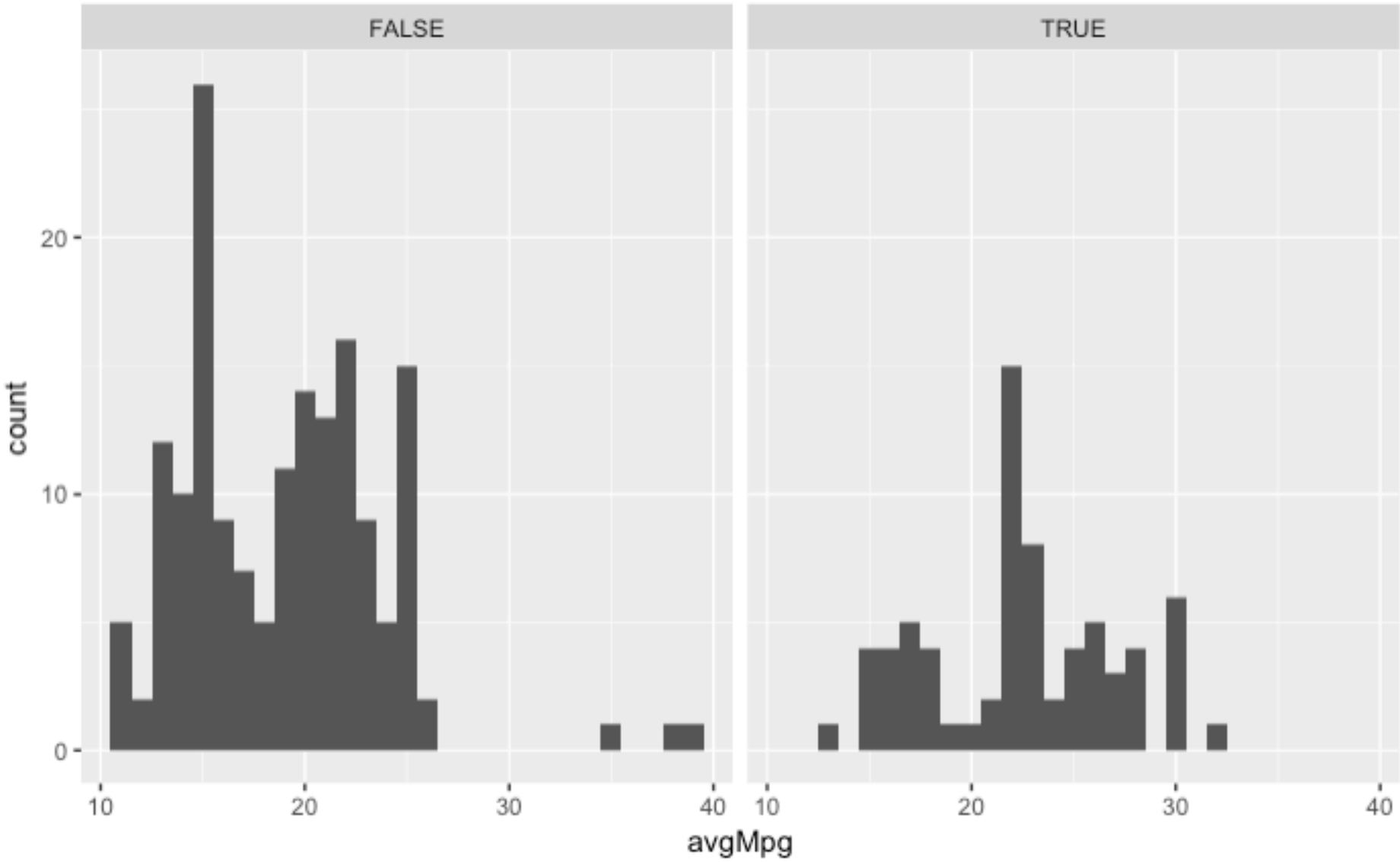
One of the concepts that was introduced this week was the idea of piping code to make it more readable. Pipes come from the [`magrittr`](http://magrittr.tidyverse.org) package, which is installed as part of the `tidyverse`. You do not need to explicitly load `magrittr` - it is automatically loaded as a dependency of `dplyr`.
```

week02.Rmd

Preview | Insert | Run

```{r}
mpg %>%
  select(manufacturer,model, cty, hwy) %>%
  mutate(japan = ifelse(manufacturer == "honda" | manufacturer == "nissan" |
                          manufacturer == "subaru" | manufacturer == "toyota", TRUE, FALSE)) %>%
  mutate(avgMpg = (cty+hwy)/2) %>%
  ggplot() +
    geom_histogram(mapping = aes(avgMpg)) +
    facet_grid(~ japan)
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



Week 02 Notebook

Code

Introduction

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

Dependencies

The examples from week two require two packages - `dplyr` and `ggplot2`. These can be loaded individually or collectively as part of the `tidyverse` package.

To execute this code, click on the green right-facing arrow ("play button") on the right side of the gray code chunk below:

Hide

```
library(tidyverse)
```

```
Loading tidyverse: ggplot2
Loading tidyverse: tibble
Loading tidyverse: tidyr
Loading tidyverse: readr
Loading tidyverse: purrr
Loading tidyverse: dplyr
Conflicts with tidy packages ----------------------------------------------------------------------------------
-----
filter(): dplyr, stats
lag():    dplyr, stats
```

Example of Piped Code

One of the concepts that was introduced this week was the idea of piping code to make it more readable. Pipes come from the `magrittr` package, which is installed as part of the `tidyverse`. You do not need to explicitly load `magrittr` - it is automatically loaded as a dependency of `dplyr`.

Example 1 - Creating a New Data Frame

This is an example of a simple pipe of code to create a limited data frame from `ggplot2`'s `mpg` data set that is saved to the object `japaneseAutos`:

Hide

3 MARKDOWN

RMARKDOWN

- rmarkdown works with knitr to provide tools for writing in Markdown syntax

- Markdown is a “markup” language, with special symbols used to indicate formatting:


```
*italicized text*

**bold text**
```

italicized text

bold text

BASIC SYNTAX

Syntax	Output
`# Largest heading`	# Largest heading
`## Second largest heading`	## Second largest heading
`###### Smallest heading`	###### Smallest heading
`*italicized text*`	*italicized text*
`**bold text**`	**bold text**
`> this is a quote`	> this is a quote
`` `dataframe` ``	`dataframe`

LISTS

Syntax	Output

`* item 1`
`* item 2`
`* item 3`
	• item 1
	• item 2
	• item 3

`- item 1`
`- item 2`
`- item 3`
	• item 1
	• item 2
	• item 3

`1. item 1`
`2. item 2`
`3. item 3`
	1. item 1
	2. item 2
	3. item 3

HYPERLINKS AND PICTURES

Syntax	Output
`[Github](https://github.com)`	Github
`![picture](https://desktop.github.com/images/desktop-icon.svg)`	

4 PUTTING IT ALL TOGETHER