

Fonts and tables

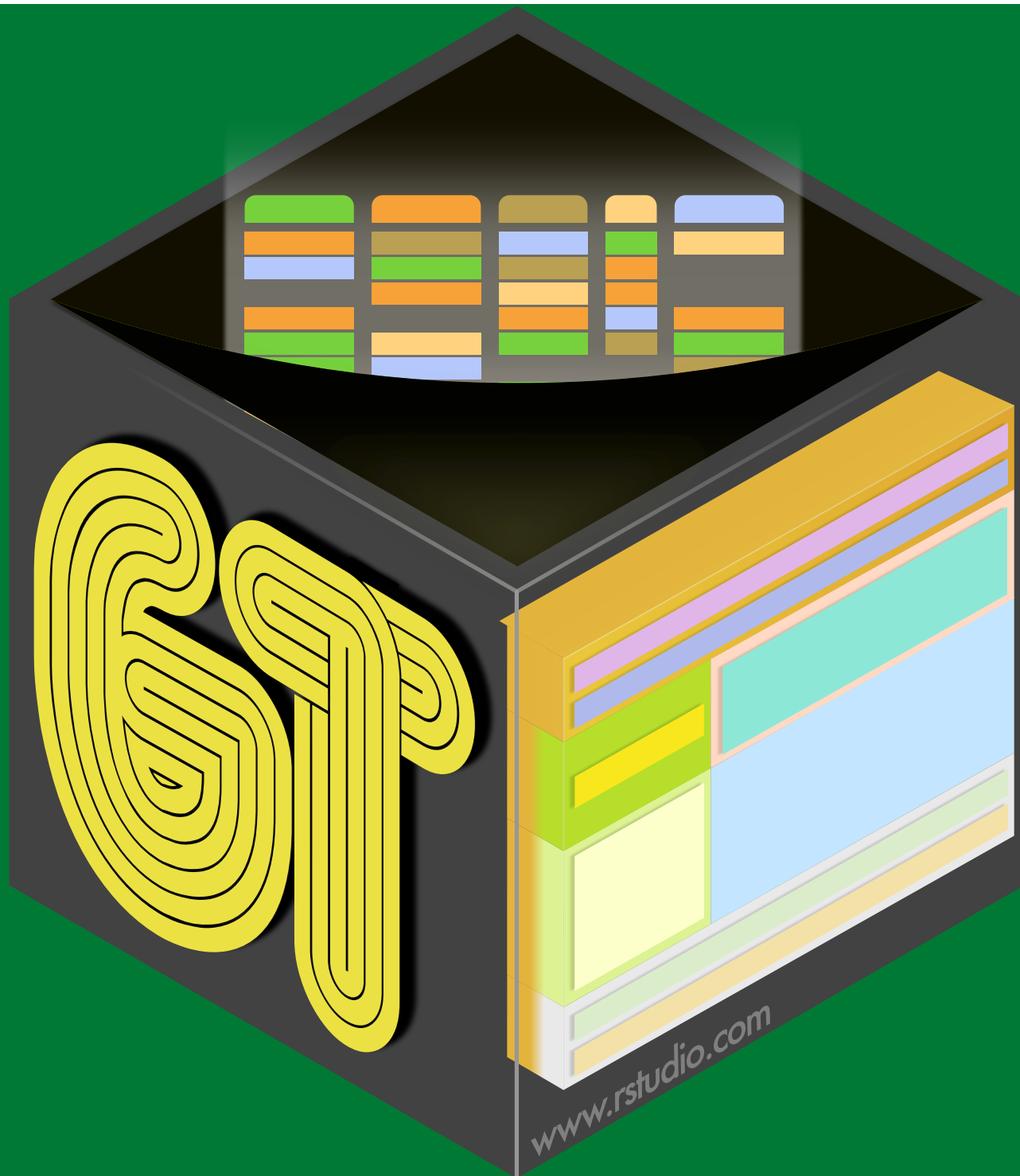
Daniel Anderson
Week 7, Class 2

Agenda

- Tables with `gt`
- Fonts with `showtext` and/or `extrafont`

Learning objectives

- Be comfortable with the basics of `gt`
 - create a table
 - format columns
 - create spanner heads
 - etc.
- Understand how to use additional fonts (if you so choose)



Overview

- New package (still very actively under development) by RStudio
- Really promising
 - Pipe-oriented
 - Beautiful tables easy
 - Spanner heads/grouping used to be a total pain - not so anymore
 - Renders to HTML/PDF without even thinking about it
- Has a few limitations relative to {papaja}
- May run into bumps because of the active development

Install

```
remotes::install_github("rstudio/gt")
```

The hard part

- Getting your data in the format you want a table in
- Utilize your `pivot_*` skills regularly

```
library(fivethirtyeight)
flying
```

```
## # A tibble: 1,040 x 27
##   respondent_id gender age   height children_under_18 household_income
##           <dbl> <chr> <ord> <ord>   <lgl>                <ord>
## 1      3436139758 <NA>   <NA>   <NA>    NA                <NA>
## 2      3434278696 Male    30-44 "6'3\... TRUE             <NA>
## 3      3434275578 Male    30-44 "5'8\... FALSE          $100,000 - $149...
## 4      3434268208 Male    30-44 "5'11... FALSE          $0 - $24,999
## 5      3434250245 Male    30-44 "5'7\... FALSE          $50,000 - $99,9...
## 6      3434245875 Male    30-44 "5'9\... TRUE           $25,000 - $49,9...
## 7      3434235351 Male    30-44 "6'2\... TRUE             <NA>
## 8      3434218031 Male    30-44 "6'0\... TRUE           $0 - $24,999
## 9      3434213681 <NA>   <NA>   "6'0\... TRUE             <NA>
## 10     3434172894 Male    30-44 "5'6\... FALSE          $0 - $24,999
## # ... with 1,030 more rows, and 21 more variables: education <ord>,
## #   location <chr>, frequency <ord>, recline_frequency <ord>,
## #   recline_obligation <lgl>, recline_rude <ord>, recline_eliminate <lgl>,
```

```
smry <- flying %>%
  count(gender, age, recline_frequency) %>%
  drop_na(age, recline_frequency) %>%
  pivot_wider(names_from = "age",
              values_from = "n")
```

smry

```
## # A tibble: 10 x 6
##   gender recline_frequency `18-29` `30-44` `45-60` `> 60`
##   <chr>   <ord>           <int>   <int>   <int>   <int>
## 1 Female Never             24      21      19      23
## 2 Female Once in a while    36      25      30      36
## 3 Female About half the time 10      22      18      17
## 4 Female Usually           13      22      26      28
## 5 Female Always            10      21      29      12
## 6 Male   Never             24      17      20      18
## 7 Male   Once in a while    19      39      40      29
## 8 Male   About half the time 11      11      16      11
## 9 Male   Usually           14      30      15      27
## 10 Male  Always            11      14      21      14
```

Turn into table

Disclaimer

These all look slightly different on the slides

```
library(gt)
smry %>%
  gt()
```

gender	recline_frequency	18-29	30-44	45-60
Female	Never	24	21	19
Female	Once in a while	36	25	30
Female	About half the time	10	22	18
Female	Usually	13	22	26
Female	Always	10	21	29
Male	Never	24	17	20
Male	Once in a while	19	39	40
Male	About half the time	11	11	16
Male	Usually	14	30	15
Male	Always	11	14	21

Add gender as a grouping variable

```
smry %>%  
  group_by(gender) %>%  
  gt()
```

recline_frequency	18-29	30-44	45-60	>60
Female				
Never	24	21	19	23
Once in a while	36	25	30	36
About half the time	10	22	18	17
Usually	13	22	26	28
Always	10	21	29	12
Male				
Never	24	17	20	18
Once in a while	19	39	40	29
About half the time	11	11	16	11
Usually	14	30	15	27
Always	11	14	21	14

Add a spanner head

```
smry %>%  
  group_by(gender) %>%  
  gt() %>%  
  tab_spanner(label = "Age Range",  
              columns = vars(`18-29`, `30-44`, `45-60`, `> 60`))
```

recline_frequency	Age Range			
	18-29	30-44	45-60	> 60
Female				
Never	24	21	19	23
Once in a while	36	25	30	36
About half the time	10	22	18	17
Usually	13	22	26	28
Always	10	21	29	12
Male				
Never	24	17	20	18
Once in a while	19	39	40	29
About half the time	11	11	16	11
Usually	14	30	15	27
Always	11	14	21	14

Change column names

```
smry %>%  
  group_by(gender) %>%  
  gt() %>%  
  tab_spanner(label = "Age Range",  
              columns = vars(`18-29`, `30-44`, `45-60`, `> 60`)) %>%  
  cols_label(recline_frequency = "Recline")
```

Recline	Age Range			
	18-29	30-44	45-60	> 60
Female				
Never	24	21	19	23
Once in a while	36	25	30	36
About half the time	10	22	18	17
Usually	13	22	26	28
Always	10	21	29	12
Male				
Never	24	17	20	18
Once in a while	19	39	40	29
About half the time	11	11	16	11
Usually	14	30	15	27
Always	11	14	21	14

Align columns

```
smry %>%  
  group_by(gender) %>%  
  gt() %>%  
  tab_spanner(label = "Age Range",  
              columns = vars(`18-29`, `30-44`, `45-60`, `> 60`)) %>%  
  cols_label(recline_frequency = "Recline") %>%  
  cols_align(align = "left", columns = vars(recline_frequency))
```

Recline	Age Range			
	18-29	30-44	45-60	> 60
Female				
Never	24	21	19	23
Once in a while	36	25	30	36
About half the time	10	22	18	17
Usually	13	22	26	28
Always	10	21	29	12
Male				
Never	24	17	20	18
Once in a while	19	39	40	29
About half the time	11	11	16	11
Usually	14	30	15	27
Always	11	14	21	14

Add a title

```
smry %>%  
  group_by(gender) %>%  
  gt() %>%  
  tab_spanner(label = "Age Range",  
              columns = vars(`18-29`, `30-44`, `45-60`, `> 60`)) %>%  
  cols_label(recline_frequency = "Recline") %>%  
  cols_align(align = "left", columns = vars(recline_frequency)) %>%  
  tab_header(title = "Airline Passengers",  
            subtitle = "Leg space is limited, what do you do?")
```


Airline Passengers				
Leg space is limited, what do you do?				
Recline	Age Range			
	18-29	30-44	45-60	> 60
Female				
Never	24	21	19	23
Once in a while	36	25	30	36
About half the time	10	22	18	17
Usually	13	22	26	28
Always	10	21	29	12
Male				
Never	24	17	20	18
Once in a while	19	39	40	29
About half the time	11	11	16	11
Usually	14	30	15	27
Always	11	14	21	14

Format columns

```
smry %>%  
  mutate_at(vars(`18-29`, `30-44`, `45-60`, `> 60`), ~./100) %>%  
  group_by(gender) %>%  
  gt() %>%  
  tab_spanner(label = "Age Range",  
              columns = vars(`18-29`, `30-44`, `45-60`, `> 60`)) %>%  
  fmt_percent(vars(`18-29`, `30-44`, `45-60`, `> 60`),  
              decimals = 0) %>%  
  cols_label(recline_frequency = "Recline") %>%  
  cols_align(align = "left", columns = vars(recline_frequency)) %>%  
  tab_header(title = "Airline Passengers",  
            subtitle = "Leg space is limited, what do you do?")
```

Airline Passengers				
Leg space is limited, what do you do?				
Recline	Age Range			
	18-29	30-44	45-60	> 60
Female				
Never	24%	21%	19%	23%
Once in a while	36%	25%	30%	36%
About half the time	10%	22%	18%	17%
Usually	13%	22%	26%	28%
Always	10%	21%	29%	12%
Male				
Never	24%	17%	20%	18%
Once in a while	19%	39%	40%	29%
About half the time	11%	11%	16%	11%
Usually	14%	30%	15%	27%
Always	11%	14%	21%	14%

Add a source note

```
smry %>%
  mutate_at(vars(`18-29`, `30-44`, `45-60`, `> 60`), ~./100) %>%
  group_by(gender) %>%
  gt() %>%
  tab_spanner(label = "Age Range",
              columns = vars(`18-29`, `30-44`, `45-60`, `> 60`)) %>%
  fmt_percent(vars(`18-29`, `30-44`, `45-60`, `> 60`),
              decimals = 0) %>%
  cols_label(recline_frequency = "Recline") %>%
  cols_align(align = "left", columns = vars(recline_frequency)) %>%
  tab_header(title = "Airline Passengers",
             subtitle = "Leg space is limited, what do you do?") %>%
  tab_source_note(source_note = md("Data from [fivethirtyeight](https://five
```

Airline Passengers				
Leg space is limited, what do you do?				
Recline	Age Range			
	18-29	30-44	45-60	> 60
Female				
Never	24%	21%	19%	23%
Once in a while	36%	25%	30%	36%
About half the time	10%	22%	18%	17%
Usually	13%	22%	26%	28%
Always	10%	21%	29%	12%
Male				
Never	24%	17%	20%	18%
Once in a while	19%	39%	40%	29%
About half the time	11%	11%	16%	11%
Usually	14%	30%	15%	27%
Always	11%	14%	21%	14%
Data from fivethirtyeight				

Color cells

```
smry %>%
  mutate_at(vars(`18-29`, `30-44`, `45-60`, `> 60`), ~./100) %>%
  group_by(gender) %>%
  gt() %>%
  tab_spanner(label = "Age Range",
              columns = vars(`18-29`, `30-44`, `45-60`, `> 60`)) %>%
  fmt_percent(vars(`18-29`, `30-44`, `45-60`, `> 60`),
              decimals = 0) %>%
  cols_label(recline_frequency = "Recline") %>%
  data_color(vars(`18-29`, `30-44`, `45-60`, `> 60`),
            colors = scales::col_numeric(palette = c(c("#FFFFFF", "#FF0000")
  cols_align(align = "left", columns = vars(recline_frequency)) %>%
  tab_header(title = "Airline Passengers",
            subtitle = "Leg space is limited, what do you do?") %>%
  tab_source_note(source_note = md("Data from [fivethirtyeight](https://five
```

Airline Passengers				
Leg space is limited, what do you do?				
Recline	Age Range			
	18-29	30-44	45-60	> 60
Female				
Never	24%	21%	19%	23%
Once in a while	36%	25%	30%	36%
About half the time	10%	22%	18%	17%
Usually	13%	22%	26%	28%
Always	10%	21%	29%	12%
Male				
Never	24%	17%	20%	18%
Once in a while	19%	39%	40%	29%
About half the time	11%	11%	16%	11%
Usually	14%	30%	15%	27%
Always	11%	14%	21%	14%
Data from fivethirtyeight				

What else?

- Lots more it can do, and lots more in development
- See the [website](#)
- gtcars case study is worth going through



A few other table options

kableExtra

- Much more established
- Still really flexible
- Still mostly works with piping
- Renders in [pdf](#) or [html](#)
- Probably the best option currently, but I think {gt} will be the way to go moving forward

A few quick examples

Make sure to specify `results = "asis"` in your chunk options.

```
library(knitr)
library(kableExtra)
dt <- mtcars[1:5, 1:6]
kable(dt) %>%
  kable_styling("striped") %>%
  column_spec(5:7, bold = TRUE)
```

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875
Datsun 710	22.8	4	108	93	3.85	2.320
Hornet 4 Drive	21.4	6	258	110	3.08	3.215
Hornet Sportabout	18.7	8	360	175	3.15	3.440

```
kable(dt) %>%  
  kable_styling("striped") %>%  
  column_spec(5:7, bold = TRUE) %>%  
  row_spec(c(2, 4),  
    bold = TRUE,  
    color = "#EFF3F7",  
    background = "#71B0DE")
```

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875
Datsun 710	22.8	4	108	93	3.85	2.320
Hornet 4 Drive	21.4	6	258	110	3.08	3.215
Hornet Sportabout	18.7	8	360	175	3.15	3.440

```
kable(dt) %>%
  kable_styling("striped", full_width = F) %>%
  pack_rows("Group 1", 1, 3,
    label_row_css = "background-color: #666; color: #fff;") %>%
  pack_rows("Group 2", 4, 5,
    label_row_css = "background-color: #666; color: #fff;")
```

	mpg	cyl	disp	hp	drat	wt
Group 1						
Mazda RX4	21.0	6	160	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875
Datsun 710	22.8	4	108	93	3.85	2.320
Group 2						
Hornet 4 Drive	21.4	6	258	110	3.08	3.215
Hornet Sportabout	18.7	8	360	175	3.15	3.440

KableExtra wrapup

Many other options, please see the documentation. Works well for PDF and HTML.
What about Microsoft Word?

flextable

flextable **0.5.8**

Overview

Selectors

Layout

Format visual properties

Format Content

Render as image

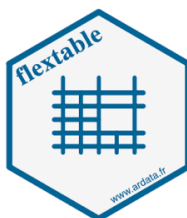
Examples

Function reference



flextable R package

build passing BUILD PASSING CRAN 0.5.8 downloads 16K/month repo status Active



The flextable package provides a framework for easily create tables for reporting and publications. Tables can be embedded within:

- R Markdown documents with support for HTML, Word and PowerPoint documents.
- Microsoft Word or PowerPoint documents.
- PDF documents with package `pagedown` (it's only HTML)

Tables can also be rendered as R plots or graphic files (png, pdf and jpeg).

Getting Started

An API is available to let R users create tables for reporting and control their formatting properties and their layout. A `flextable` object is a `data.frame` representation, it can be manipulated with functions that give control over:

Links

Download from CRAN at
<https://cloud.r-project.org/package=flextable>

Report a bug at
<https://github.com/davidgohel/flextable/issues>

Visit [ARDATA](https://www.ardata.fr) website at
<https://www.ardata.fr>

License

[GPL-3](#)









Developers

David Gohel
Author, maintainer

[All authors...](#)

2019 Women's World Cup Predictions

Soccer Power Index (SPI) ratings and chances of advancing for every team

TEAM	GROUP	Team Rating			Chance of Finishing Group Stage In ...			Knockout Stage Chances				WIN WORLD CUP
		SPI	OFF.	DEF.	1ST PLACE	2ND PLACE	3RD PLACE	MAKE ROUND OF 16	MAKE QTR-FINALS	MAKE SEMIFINALS	MAKE FINAL	
 USA 6 pts.	F	98.3	5.5	0.6	83%	17%	—	✓	78%	47%	35%	24%
 France 6 pts.	A	96.3	4.3	0.5	>99%	<1%	<1%	✓	78%	42%	30%	19%
 Germany 6 pts.	B	93.8	4.0	0.7	98%	2%	—	✓	89%	48%	28%	12%
 Canada 6 pts.	E	93.5	3.7	0.6	39%	61%	—	✓	59%	36%	20%	9%
 England 6 pts.	D	91.9	3.5	0.6	71%	29%	—	✓	69%	43%	16%	8%
 Netherlands 6 pts.	E	92.7	3.9	0.7	61%	39%	—	✓	59%	37%	19%	8%
 Australia 3 pts.	C	92.8	4.2	0.9	13%	54%	34%	>99%	54%	26%	10%	5%
 Brazil 3 pts.	F	92.4	3.8	0.8	17%	63%	20%	✓	67%	30%	10%	4%

Many others

- [huxtable](#)
- [formattable](#)
- [DT](#) (particularly helpful when using [shiny](#))
- [rhandsontable](#)

Particularly helpful for modeling

- [stargazer](#)
- [pixiedust](#)
- [modelsummary](#)

For descriptives

- [gtsummary](#)

Fonts

General advice

- Use different fonts to distinguish things
 - Specifically code
 - Consider different fonts for different heading levels, and/or to distinguish headers from the body
- **Always** choose a sans-serif font for code
- Explore and try - it makes a big impact on the overall look/feel (bigger than you may expect if you haven't played with fonts much before)
- Try not to get sucked into too deep of a rabbit hole

Google fonts

<https://fonts.google.com>

- Open source, designed for the web
- Good place to explore fonts
- Can be incorporated via the `{showtext}` package!

{showtext} example

```
devtools::install_github("yixuan/showtext")
```

```
library(showtext)
```

```
font_add_google('Monsieur La Doulaise', "mld")
```

```
font_add_google('Special Elite', "se")
```

```
showtext_auto()
```

```
quartz()
```

```
ggplot(mtcars, aes(displacement, mpg)) +
```

```
  geom_point() +
```

```
  labs(title = "An amazing title",
```

```
        subtitle = "with the world's most boring dataset") +
```

```
  theme(plot.subtitle = element_text(size = 18, family = "se"),
```

```
        plot.title = element_text(size = 22, family = "mld"),
```

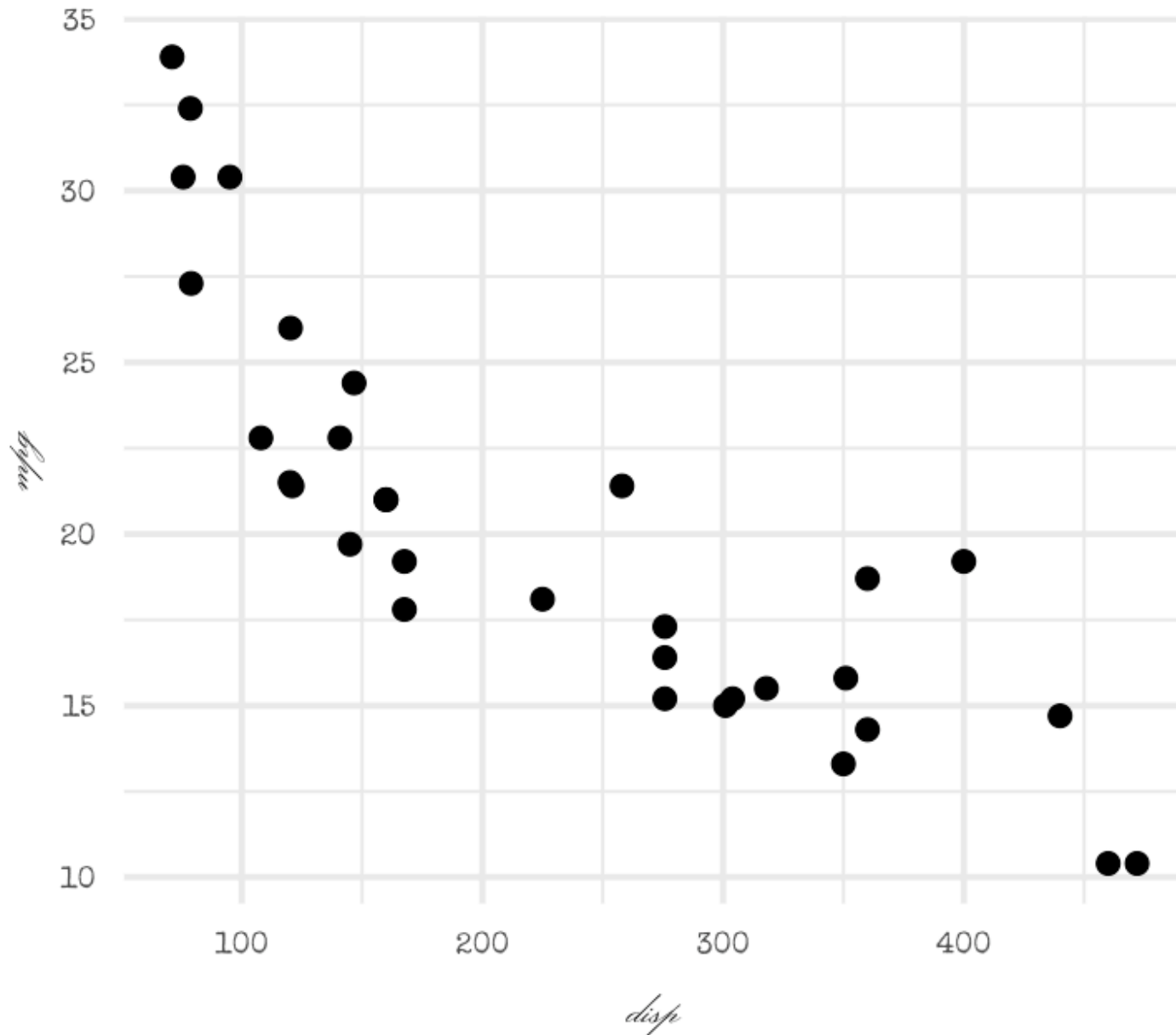
```
        axis.title = element_text(size = 18, family = "mld"),
```

```
        axis.text.x = element_text(size = 12, family = "se"),
```

```
        axis.text.y = element_text(size = 12, family = "se"))
```

An amazing title

with the world's most boring dataset



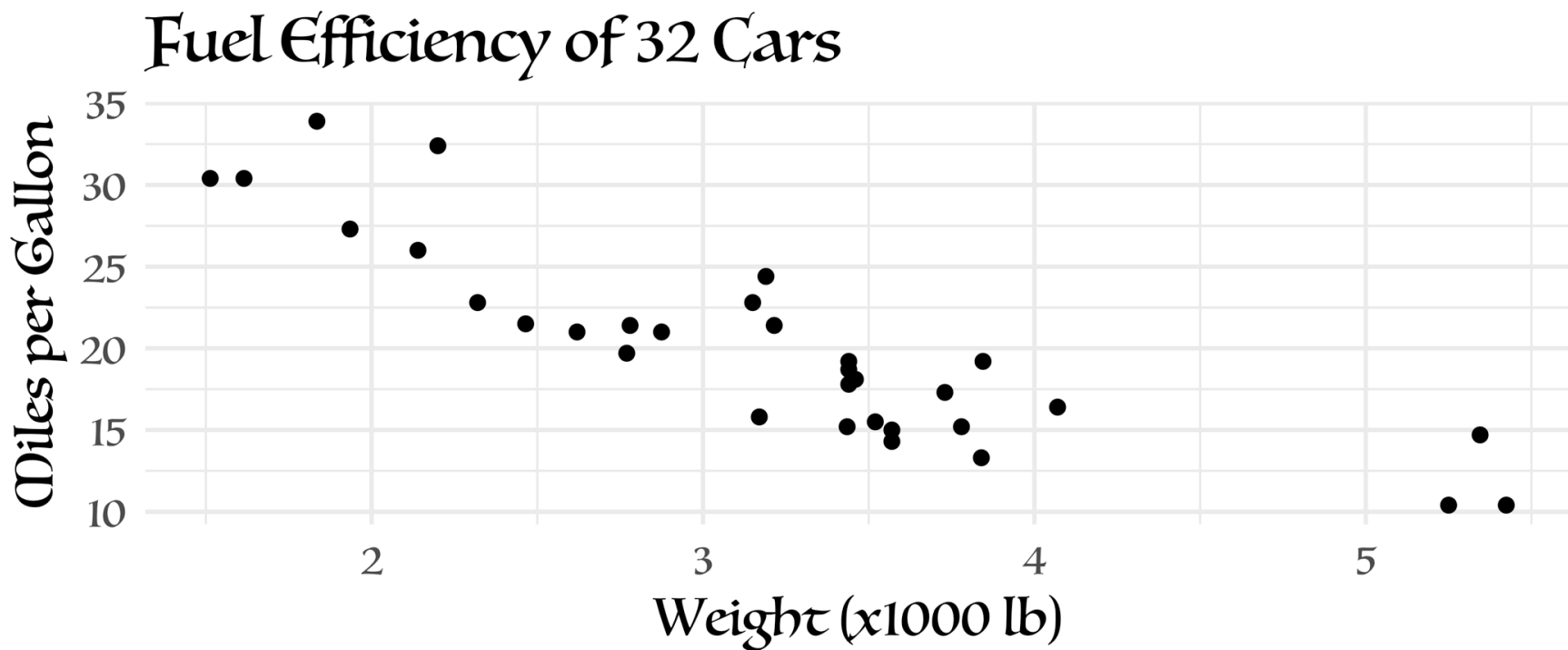
extrafont

- Primary downside - you have to have the fonts installed on your computer
- Look at the install documentation - it's pretty good and fairly comprehensive

```
library(extrafont)
# font_import() Only need to run once
fonts() # list fonts
```

```
## [1] ".Keyboard" "System Font"
## [3] ".SF NS Rounded" "Andale Mono"
## [5] "Apple Braille" "AppleMyungjo"
## [7] "Arial Black" "Arial"
## [9] "Arial Narrow" "Arial Rounded MT Bold"
## [11] "Arial Unicode MS" "Bodoni Ornaments"
## [13] "Bodoni 72 Smallcaps" ""
## [15] "Brush Script MT" "Comic Sans MS"
## [17] "Courier New" "DIN Alternate"
## [19] "DIN Condensed" "Fira Code iScript"
## [21] "Georgia" "Impact"
## [23] "Khmer Sangam MN" "Lao Sangam MN"
## [25] "Luminari" "Microsoft Sans Serif"
## [27] "Tahoma" "Times New Roman"
## [29] "Trattatello" "Trebuchet MS"
## [31] "Verdana" "Webdings"
```

```
ggplot(mtcars, aes(wt, mpg)) +  
  geom_point() +  
  labs(title = "Fuel Efficiency of 32 Cars",  
        x = "Weight (x1000 lb)",  
        y = "Miles per Gallon") +  
  theme(text = element_text(family = "Luminari", size = 30))
```



Why fonts matter

A few examples of epic fails

h/t Will Chase

MegaFlicks



The image shows two light gray sticky notes with yellow tabs at the top, pinned to a dark gray background. The left note has the text 'I will always find you' in a red, lowercase, cursive font. The right note has the text 'I WILL ALWAYS FIND YOU' in a red, uppercase, all-caps font. Below the notes, the text 'FONT MATTERS.' is written in a white, uppercase, sans-serif font.

I will always
find you

I WILL ALWAYS
FIND YOU

FONT MATTERS.



Quick aside

Change the font of your R Markdown!

Create a new text file, import the text there, write a tiny bit of css.

```
@import url('https://fonts.googleapis.com/css?family=Akronim&display=swap');  
  
body {  
  font-family: 'Akronim', cursive;  
}
```

Modify Rmd YAML

You need to have the CSS file in the same directory as your Rmd, then reference it with

```
output:  
  html_document:  
    css: custom.css
```

Here I'm assuming the file you saved before is called "custom.css".

Render!

Untitled

Daniel Anderson

2/18/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more information, see <http://rmarkdown.rstudio.com>.

When you click the Knit button a document will be generated that includes both content as well as the output of any embedded R code chunks like this:

```
summary(cars)
```

```
##           speed           dist
##  Min.      : 4.0      Min.      : 2.00
##  1st Qu.:12.0      1st Qu.: 26.00
##  Median :15.0      Median : 36.00
##  Mean   :15.4      Mean   : 42.98
##  3rd Qu.:19.0      3rd Qu.: 56.00
##  Max.    :25.0      Max.     :120.00
```

Resource for learning more

- I'm not an expert on fonts. I have mostly just picked what looks nice to me.

off down the rabbit hole again



Best I've heard of is [practical typography](#)



BUTTERICK'S PRACTICAL TYPOGRAPHY

2ND EDITION

Typography in ten minutes

Summary of key rules

Start

foreword by Erik Spiekermann

introduction

how to use this book

acknowledgments

about Matthew Butterick

legal

Please pay for this book

how to pay for this book

why you should pay

MB fonts

Why typography matters

what is typography?

who is typography for?

why does typography
matter?

what is good typography?

where do the rules come
from?

Next time

Create your own blog!