

# **R MIEF Skills Workshop**

## **Session 2**

**September 13, 2024**

# Today

- Learn how to:
  - Create a scatter plot, density plot, and bar chart using the `ggplot2` package
  - Create flexible and easy-to-read tables of any dataset using the `gt` package
  - Create simple academic-standard regression output tables using the `stargazer` package
- Practice the above!

# **Data Visualization — Descriptive Statistics — Plots**

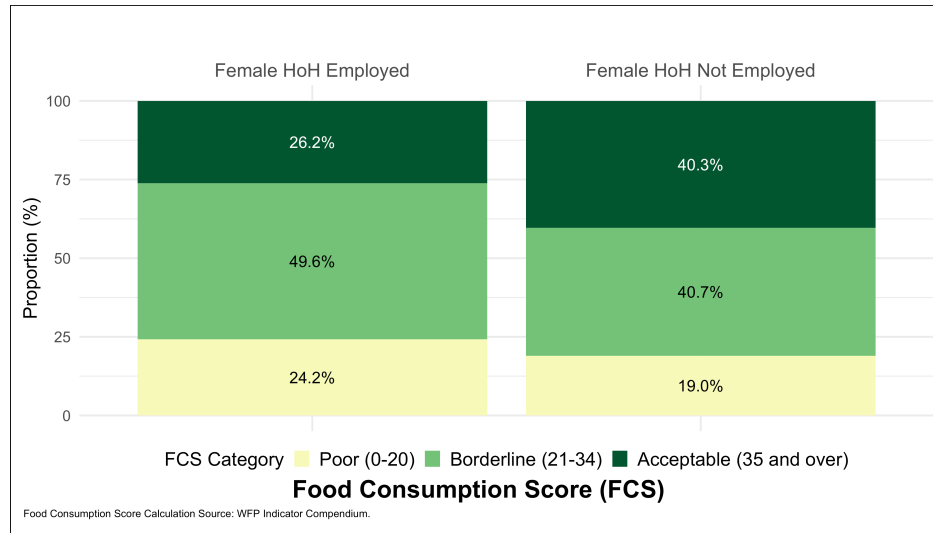
# Descriptive Stats Plots

`ggplot2` is the gold standard in data visualization in data work. It's one of the main reason that people use R over other programming languages.

Very simple syntax and allows you to add elements very easily.

You can use `ggplot2` to create any type of plot you can think of.

I've included a lot of links at the end of these slides to explore the possibilities of `ggplot2` further. Strongly recommend you use them or at least save them somewhere.



# The Magic of `ggplot2`

Using `ggplot2` to create plots is great because the **structure** it sets up makes plot creation intuitive.

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(  
    mapping = aes(<MAPPINGS>),  
    stat = <STAT>,  
    position = <POSITION>  
  ) +  
  <SCALE_FUNCTION> +  
  <FACET_FUNCTION> +  
  <THEME_FUNCTION>
```

1. **Data**: The data that you want to visualize
2. **Layers**: `geom_` and `stat_` → The geometric shapes and statistical summaries representing the data
3. **Aesthetics**: `aes()` → Aesthetic mappings of the geometric and statistical objects
4. **Scales**: `scale_` → Maps between the data and the aesthetic dimensions
5. **Facets**: `facet_` → The arrangement of the data into a grid of plots
6. **Visual themes**: `theme()` and `theme_` → The overall visual defaults of a plot

# Scatter Plot — Step-by-Step

---

Dataset

Convert to Plot

Add Something

---

Start with a dataset you want to visualize

```
head(mtcars)
```

##		mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
##	Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
##	Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
##	Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
##	Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
##	Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
##	Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

# Scatter Plot — Step-by-Step

Dataset

Convert to Plot

Add Something

```
ggplot(mtcars)
```

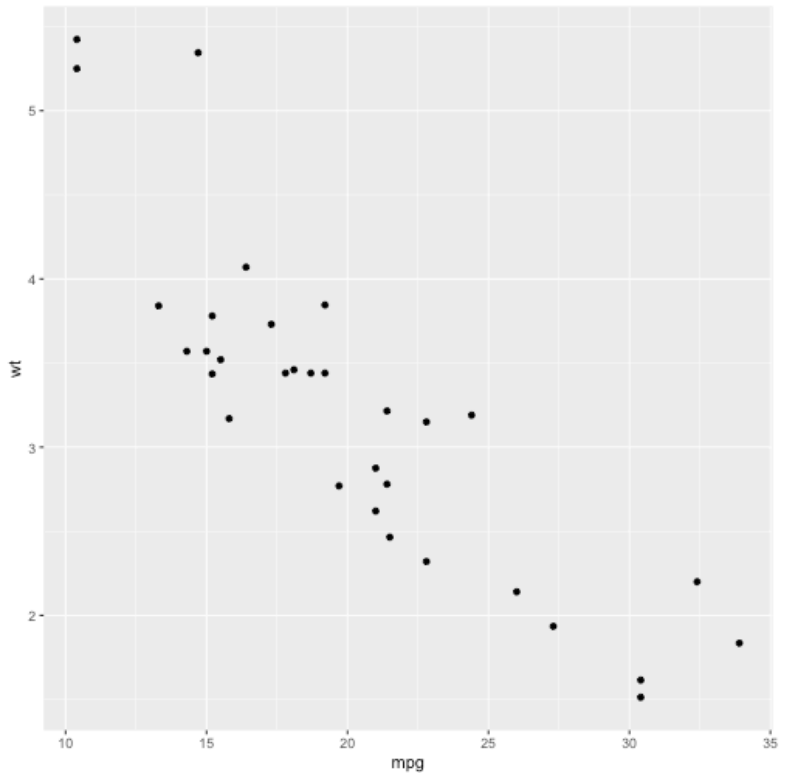
# Scatter Plot — Step-by-Step

Dataset

Convert to Plot

Add Something

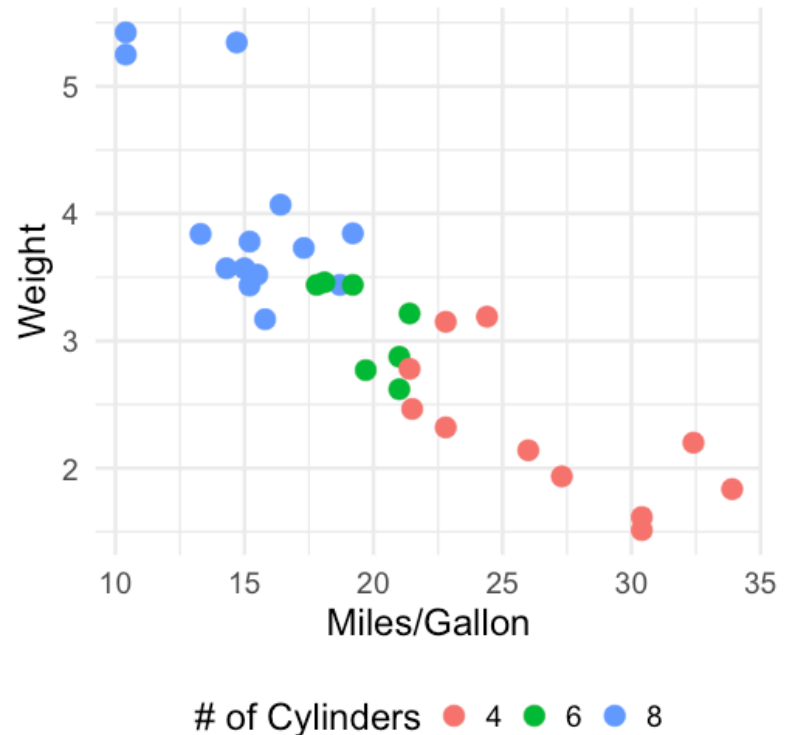
```
ggplot(mtcars) +  
  geom_point(  
    aes(x = mpg, y = wt)  
  )
```





# Scatter Plot — Make It Better

```
ggplot(mtcars) +  
  geom_point(  
    aes(  
      x = mpg, y = wt,  
      color = factor(cyl)  
    ),  
    size = 6  
  ) +  
  xlab("Miles/Gallon") +  
  ylab("Weight") +  
  scale_color_discrete(  
    name = "# of Cylinders"  
  ) +  
  theme_minimal(base_size = 24) +  
  theme(  
    legend.position = "bottom"  
  )
```



# Bar Plot — Step-by-Step

---

Dataset

Convert to Plot

Add Something

Fix Class Issue

---

Start with a dataset you want to visualize

```
mtcars_summary <- mtcars %>%  
  group_by(cyl) %>%  
  summarize(  
    mpg = mean(mpg)  
  ) %>%  
  ungroup()  
  
mtcars_summary
```

```
## # A tibble: 3 × 2  
##   cyl    mpg  
##   <dbl> <dbl>  
## 1     4  26.7  
## 2     6  19.7  
## 3     8  15.1
```

# Bar Plot — Step-by-Step

Dataset

Convert to Plot

Add Something

Fix Class Issue

```
ggplot(mtcars_summary)
```

# Bar Plot — Step-by-Step

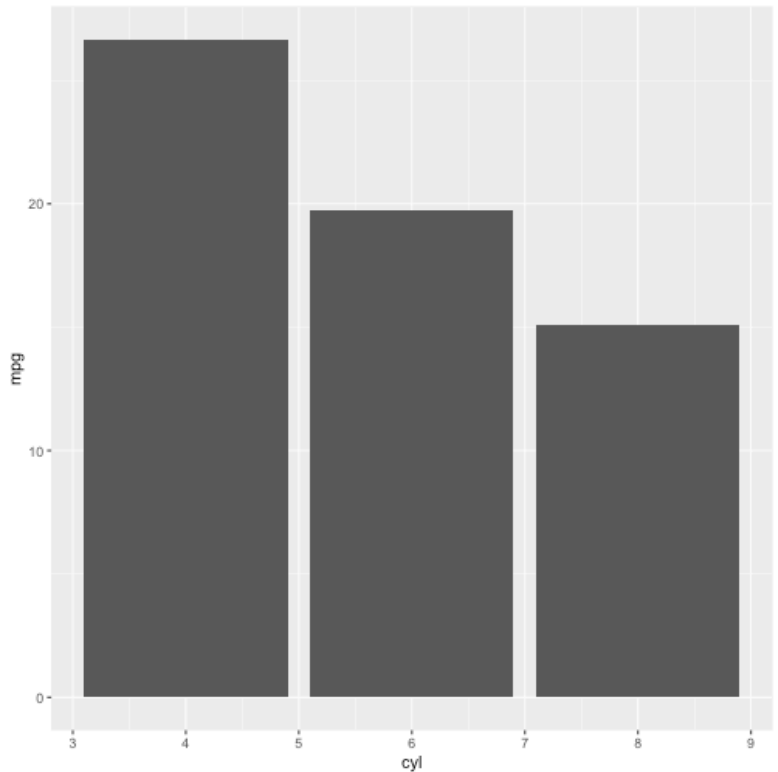
Dataset

Convert to Plot

Add Something

Fix Class Issue

```
ggplot(mtcars_summary) +  
  geom_col(  
    aes(  
      x = cyl,  
      y = mpg  
    )  
  )  
)
```



# Bar Plot — Step-by-Step

Dataset

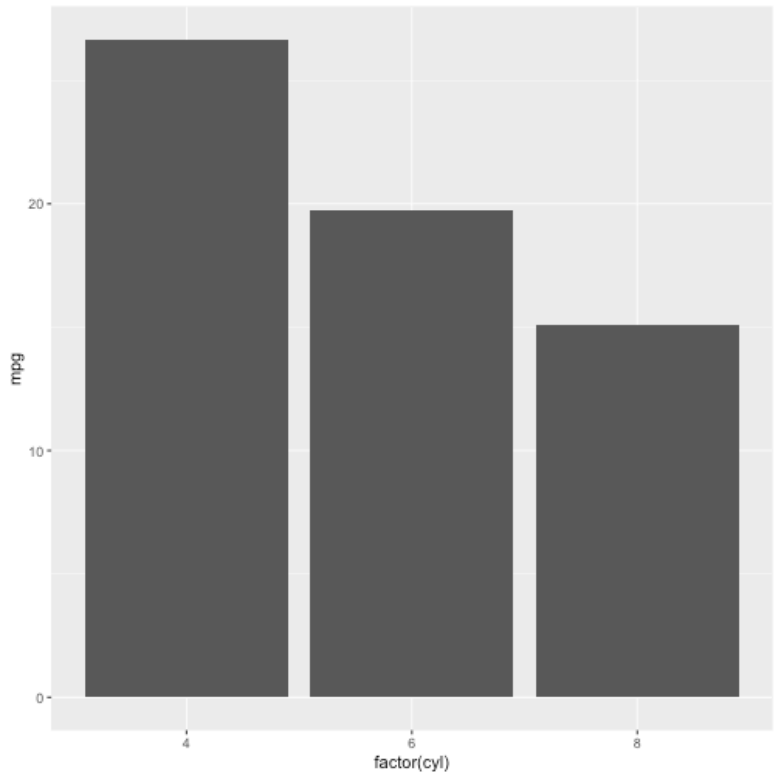
Convert to Plot

Add Something

Fix Class Issue

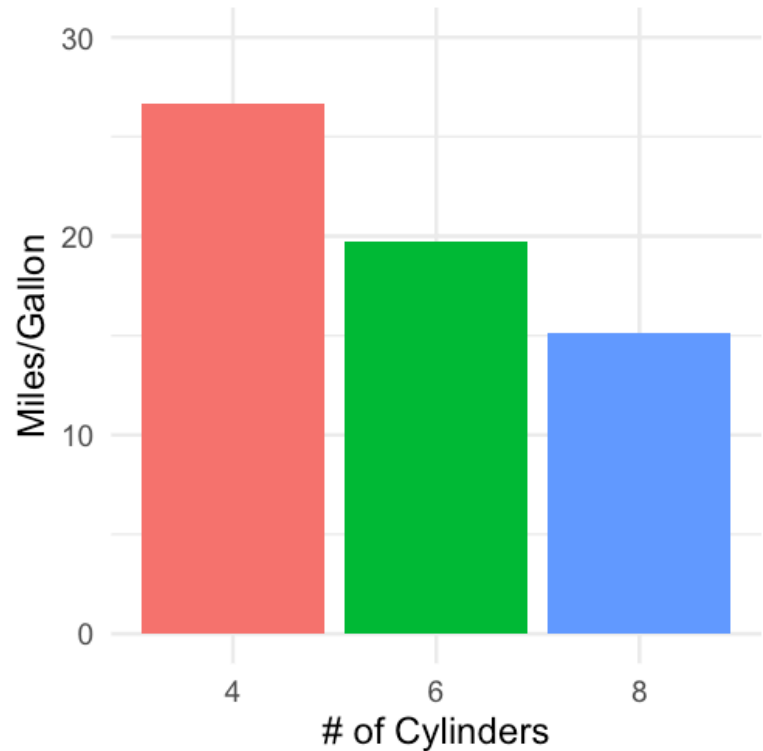
```
ggplot(mtcars_summary) +  
  geom_col(  
    aes(  
      x = factor(cyl),  
      y = mpg  
    )  
  )
```

`cyl` categorizes cars by number of cylinders. Although the values are numbers, it is a **categorical** variable. We communicate this to `ggplot()` using the `factor()` function.



# Bar Plot — Make It Better

```
ggplot(mtcars_summary) +  
  geom_col(  
    aes(  
      x = factor(cyl),  
      y = mpg,  
      fill = factor(cyl)  
    )  
  ) +  
  xlab("# of Cylinders") +  
  ylab("Miles/Gallon") +  
  scale_y_continuous(  
    limits = c(0, 30)  
  ) +  
  theme_minimal(base_size = 24) +  
  theme(  
    legend.position = "none"  
  )
```



# Plot Standards

## 1. Your plot should be **properly labeled**:

- The plot should have a title describing its content
- Axes should be labeled
- Legend (if any) should have a title and labels

## 2. Your plot should be **properly formatted**:

- Axis dimensions should be appropriate. What is appropriate varies depending on context, but usually you should aim to fill the plot space with data
- Text size should be large enough for text to be legible

## 3. Your plot should be **self contained**. People should be able to understand your plot and its data without any other context or explanatory text. That means:

- A caption note that includes data source and any important data construction notes
- Title and subtitle that deliver the plot's *message*

# **Data Visualization — Descriptive Statistics — Tables**



# Descriptive Statistics Tables

Thankfully, not every RA position requires academic-standard tables or use of LaTeX.

It is still useful, however, to be able to communicate descriptive statistics about data.

There are countless R packages to help do this. Today, we're looking at the `gt` package. It's simple to use and it's very easy to create good-looking tables using it.

`gt` exports into .png, .pdf, or .html. You can add interactive elements, plots within columns.

GEWE El Salvador Baseline					
Module L -- Female HoH -- Time Use -- Weekdays					
	Mean (SD)	Median (Q1, Q3)	Min - Max	# Obs (% Group)	# in Group
Daily Time Spent (Hours)					
Agriculture (Household)	0.24 (0.81)	0 (0, 0)	0 - 10	1,275 (100%)	1,275
Childcare	0.97 (1.49)	0 (0, 1.6)	0 - 14.25	1,275 (100%)	1,275
Chores	4.76 (2.54)	4.58 (3, 6.48)	0 - 13.75	1,275 (100%)	1,275
Collecting Water	0.17 (0.48)	0 (0, 0)	0 - 4.25	1,275 (100%)	1,275
Collecting Wood	0.2 (0.54)	0 (0, 0)	0 - 4.42	1,275 (100%)	1,275
Eating	2 (1.06)	1.75 (1.29, 2.5)	0 - 9.25	1,275 (100%)	1,275
Leisure/Religion	2.39 (2.04)	2 (0.75, 3.5)	0 - 14.5	1,275 (100%)	1,275

# Descriptive Statistics Table — Step-by-Step

We will mainly use the example in the script for this. To summarize, the steps are:

- Create a dataset you want to export
- Run the dataset through the `gt()` function to create a gt object
- Customize the table using functions from the `gt` package (see online for further things you can do). Examples of what you can do include:
  - Modify column names — `cols_label()`
  - Modify borders — `tab_style()`, `cell_borders()`
  - Add colors conditional on cell value — `data_color()`
  - Add title/subtitle — `tab_header()`
- Export the table using `gtsave()`

# **Data Visualization — Simple Regression Table**

# Regression Tables

Regression tables are very common in economic/policy analysis.

They're very simple to create using R and a software called **LateX** (pronounced latek).

Unless you're getting into academic research, you don't need to know how to properly use LaTeX. Just enough to:

- Export the LaTeX script from R
- Copy/paste it into a LaTeX-reading software, e.g. Overleaf
- Export the pdf or png to share

Predicted Consumption per Capita (2019 PPP USD)

	Any Treatment vs. Control (1)	Women Working Treatment vs. Any Treatment (2)
Any Treatment	12.049** (5.330)	12.155* (6.600)
Women Working Treatment		-0.222 (8.463)
Baseline Control	0.249** (0.101)	0.249** (0.101)
Constant	22.788*** (3.483)	22.791*** (3.489)
Control Mean	27.91	27.91
Observations	761	761
R <sup>2</sup>	0.028	0.028
Adjusted R <sup>2</sup>	0.025	0.024
Residual Std. Error	44.983 (df = 758)	45.013 (df = 757)
F Statistic	10.925*** (df = 2; 758)	7.275*** (df = 3; 757)

Note:

\*p<0.1; \*\*p<0.05; \*\*\*p<0.01

# Regression Table — Step by Step

---

Run Regression in R

---

Convert to Exportable Table

---

```
# Simplest regression format in R
```

```
reg_example ← lm(  
  outcome_variable ~ independent_variable + control_variables,  
  data = dataset  
)
```

```
# Observe results
```

```
reg_example %>% summary()
```

# Regression Table — Step by Step

---

Run Regression in R

Convert to Exportable Table

---

Simply do one of these!

```
reg_example_ht ← reg_example %>%  
  huxtable::huxreg()
```

OR

```
reg_example_sg ← reg_example %>%  
  stargazer::stargazer() # Many options to make prettier
```

# Regression Table — Step by Step

---

Export Huxtable Table

---

Export Stargazer Table

---

Some simple options for the Huxtable table:

```
huxtable::quick_latex(  
  reg_example_ht,  
  file = "filepath/filepath/filepath/reg_example_ht.tex"  
)  
  
huxtable::quick_pdf(  
  reg_example_ht,  
  file = "filepath/filepath/filepath/reg_example_ht.pdf"  
)  
  
huxtable::quick_html(  
  reg_example_ht,  
  file = "filepath/filepath/filepath/reg_example_ht.html"  
)
```

# Regression Table — Step by Step

Export Huxtable Table

Export Stargazer Table

*# You can export a LaTeX script using the 'writeLines' function*

```
writeLines(  
  reg_example_sg,  
  "filepath/filepath/filepath/reg_example_sg.tex"  
)
```

To visualize your table, the easiest solution is to:

- Create a free Overleaf account on [overleaf.com](https://overleaf.com)
- Open a new document
- Copy/paste your .tex output in between the `begin{document}` and `end{document}` lines
- Click compile and then save!

You can also install the `tinytex` package and use `pdftolatex` to save a PDF file.



# **Practical Exercise — Using the World Values Survey Dataset**

# World Values Survey

## Background

*"The survey, which started in 1981, seeks to use the most rigorous, high-quality research designs in each country. The WVS consists of nationally representative surveys conducted in almost 100 countries which contain almost 90 percent of the world's population, using a common questionnaire. [...] WVS seeks to help scientists and policy makers understand changes in the beliefs, values and motivations of people throughout the world."*

## Survey Contents

- Social values, attitudes & stereotypes
- Societal well-being
- Social capital, trust and organizational membership
- Economic values
- Corruption
- Migration
- Post-materialist index
- Science & technology
- Religious values
- Security
- Ethical values & norms
- Political interest and political participation
- Political culture and political regimes
- Demography

# Today's practical component

1. Successfully run the code in the `session_2.R` script
2. Attempt the challenges at the bottom of the script!

# Mid-Course Assignment (Due on Thursday, September 26)

1. Find an open-source dataset. Anything will do, there is ton of data out there. If you are looking for inspiration, you can explore the [Tidy Tuesday Github repository](#).
2. Create a new folder called "**your name** R Course Assignment". Replace **your name** with your name.
3. Set up that folder for data use: - Create empty "code", "data", and "output" folders within it - Create an R Project in that folder. Refer back to last week's slides if you're not sure on how to do this.
4. Add your open-source dataset (I strongly recommend downloading or converting it to CSV format) to the "data" folder.
5. Download the "r\_template.R" script from the "Course Resources" module on Canvas and place it in the "code" folder.
6. Write a script that produces and saves a plot using your open-source dataset.

# Grading criteria

You will submit a .zip file of the "**your name** R Course Assignment" folder.

- Setup
  - The script's purpose should be stated at the top of the script
  - The script should load required packages for it to run
  - The script should successfully import the data used for the data visualization
- Plot
  - The plot should convey an interesting relationship between variables. For instance, you could use a scatter plot to show how two variables are positively or negatively correlated. Or you could use a bar plot to show the evolution of a variable over time.
  - The plot should meet the "plot standards" laid out in [slide 11](#).
  - The script should end with the plot being saved in the "output" folder.

To receive top grades, the script must fully run in one click without any errors.

# Links

## Tables

Marek Hlavac, [“stargazer: beautiful LATEX, HTML and ASCII tables from R statistical output”](#)

Thomas Mock, [“gt - a \(G\)rammar of \(T\)ables”](#)

## Plots

Alicia Horsch, [“A quick introduction to ggplot2”](#)

RStudio, [RStudio Cheatsheets](#)