PrettyR Graphics with ggplot2

Jonathan Wells

Reed Empirical Workshop Series

4/23/20
Outline

1. Introduction
2. The Grammar of Graphics
3. The Core Graphics
4. Advanced Graphics
Section 1

Introduction
Background

- We assume our audience...
• We assume our audience...
  • Is familiar with general types of statistical graphics (histograms, scatterplots, etc.)
Background

• We assume our audience...
  • Is familiar with general types of statistical graphics (histograms, scatterplots, etc.)
  • Has some prior experience manipulating data, writing programs, or creating graphics in R
Background

- We assume our audience...
  - Is familiar with general types of statistical graphics (histograms, scatterplots, etc.)
  - Has some prior experience manipulating data, writing programs, or creating graphics in R
  - Does not have extensive prior experience with the ggplot2 package
Background

• We assume our audience... 
  • Is familiar with general types of statistical graphics (histograms, scatterplots, etc.)
  • Has some prior experience manipulating data, writing programs, or creating graphics in R
  • Does not have extensive prior experience with the ggplot2 package

• By the end of today’s workshop, our audience should be able to...
Background

- We assume our audience...
  - Is familiar with general types of statistical graphics (histograms, scatterplots, etc.)
  - Has some prior experience manipulating data, writing programs, or creating graphics in R
  - Does not have extensive prior experience with the ggplot2 package
- By the end of today’s workshop, our audience should be able to...
  - Create a variety of common statistical graphics using ggplot2
Background

- We assume our audience...
  - Is familiar with general types of statistical graphics (histograms, scatterplots, etc.)
  - Has some prior experience manipulating data, writing programs, or creating graphics in R
  - Does not have extensive prior experience with the ggplot2 package
- By the end of today’s workshop, our audience should be able to...
  - Create a variety of common statistical graphics using ggplot2
  - Customize aesthetic attributes and add context to graphics
Background

• We assume our audience... 
  • Is familiar with general types of statistical graphics (histograms, scatterplots, etc.)
  • Has some prior experience manipulating data, writing programs, or creating graphics in R
  • Does not have extensive prior experience with the ggplot2 package

• By the end of today’s workshop, our audience should be able to... 
  • Create a variety of common statistical graphics using ggplot2
  • Customize aesthetic attributes and add context to graphics
  • Incorporate some ggplot2 graphics in their own research
Background

• We assume our audience. . .
  • Is familiar with general types of statistical graphics (histograms, scatterplots, etc.)
  • Has some prior experience manipulating data, writing programs, or creating graphics in R
  • Does not have extensive prior experience with the ggplot2 package

• By the end of today’s workshop, our audience should be able to. . .
  • Create a variety of common statistical graphics using ggplot2
  • Customize aesthetic attributes and add context to graphics
  • Incorporate some ggplot2 graphics in their own research

• To follow along to this workshop live. . .
Background

• We assume our audience. . .
  • Is familiar with general types of statistical graphics (histograms, scatterplots, etc.)
  • Has some prior experience manipulating data, writing programs, or creating graphics in R
  • Does not have extensive prior experience with the ggplot2 package
• By the end of today’s workshop, our audience should be able to. . .
  • Create a variety of common statistical graphics using ggplot2
  • Customize aesthetic attributes and add context to graphics
  • Incorporate some ggplot2 graphics in their own research
• To follow along to this workshop live. . .
  • Navigate to rstudio.reed.edu/ (or open a local install)
Background

- We assume our audience... 
  - Is familiar with general types of statistical graphics (histograms, scatterplots, etc.)
  - Has some prior experience manipulating data, writing programs, or creating graphics in R
  - Does not have extensive prior experience with the ggplot2 package
- By the end of today’s workshop, our audience should be able to... 
  - Create a variety of common statistical graphics using ggplot2
  - Customize aesthetic attributes and add context to graphics
  - Incorporate some ggplot2 graphics in their own research
- To follow along to this workshop live... 
  - Navigate to rstudio.reed.edu/ (or open a local install)
  - Create a new .Rmd file and save to appropriate directory
Background

- We assume our audience...
  - Is familiar with general types of statistical graphics (histograms, scatterplots, etc.)
  - Has some prior experience manipulating data, writing programs, or creating graphics in R
  - Does not have extensive prior experience with the ggplot2 package
- By the end of today’s workshop, our audience should be able to...
  - Create a variety of common statistical graphics using ggplot2
  - Customize aesthetic attributes and add context to graphics
  - Incorporate some ggplot2 graphics in their own research
- To follow along to this workshop live...
  - Navigate to rstudio.reed.edu/ (or open a local install)
  - Create a new .Rmd file and save to appropriate directory
  - Add the following to the first code chunk:

```r
library(knitr)
library(tidyverse)
```
What is ggplot2?

• ggplot2 is an R package for efficiently creating publication-quality statistical graphics
• It is a core component of the tidyverse collection of R packages for Data Science which share an underlying data structure and design philosophy
• The package was created by Hadley Wickham in 2005 as an implementation of Leland Wilkinson's Grammar of Graphics, a scheme for decomposing a graphic into its constitute semantic elements: A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.
What is ggplot2?

• ggplot2 is an R package for efficiently creating publication-quality statistical graphics
What is ggplot2?

- ggplot2 is an R package for \textit{efficiently} creating \textit{publication-quality} statistical graphics
- It is a core component of the \texttt{tidyverse} collection of R packages for Data Science which share an underlying data structure and design philosophy
What is ggplot2?

• ggplot2 is an R package for **efficiently** creating **publication-quality** statistical graphics

• It is a core component of the tidyverse collection of R packages for Data Science which share an underlying data structure and design philosophy

• The package was created by Hadley Wickham in 2005 as an implementation of Leland Wilkinson's *Grammar of Graphics*, a scheme for decomposing a graphic into its constitute semantic elements:
What is ggplot2?

- ggplot2 is an R package for *efficiently* creating **publication-quality** statistical graphics.

- It is a core component of the tidyverse collection of R packages for Data Science which share an underlying data structure and design philosophy.

- The package was created by Hadley Wickham in 2005 as an implementation of Leland Wilkinson's *Grammar of Graphics*, a scheme for decomposing a graphic into its constitute semantic elements:

  *A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.*
Why use ggplot2?

- If you are new to creating graphics in R, you will need to learn the key commands for either ggplot2 or base R.
Why use ggplot2?

• If you are new to creating graphics in R, you will need to learn the key commands for either ggplot2 or base R

• Compared to base R, ggplot2 provides...
Why use ggplot2?

• If you are new to creating graphics in R, you will need to learn the key commands for either ggplot2 or base R

• Compared to base R, ggplot2 provides...
  • a unified collection of components and options allowing efficient creation and modification of graphics
Why use ggplot2?

• If you are new to creating graphics in R, you will need to learn the key commands for either ggplot2 or base R

• Compared to base R, ggplot2 provides...
  • a unified collection of components and options allowing efficient creation and modification of graphics
  • improved default aesthetics when quickly plotting results
Why use ggplot2?

- If you are new to creating graphics in R, you will need to learn the key commands for either ggplot2 or base R.

- Compared to base R, ggplot2 provides...
  - a unified collection of components and options allowing efficient creation and modification of graphics
  - improved default aesthetics when quickly plotting results
  - declarative programming, compared to base R's imperative programming
Why use ggplot2?

- If you are new to creating graphics in R, you will need to learn the key commands for either ggplot2 or base R

- Compared to base R, ggplot2 provides...
  - a unified collection of components and options allowing efficient creation and modification of graphics
  - improved default aesthetics when quickly plotting results
  - **declarative** programming, compared to base R's **imperative** programming
  - emphasis on the fundamental visual decisions required for interpreting data
Why use ggplot2?

- If you are new to creating graphics in R, you will need to learn the key commands for either ggplot2 or base R
- Compared to base R, ggplot2 provides...
  - a unified collection of components and options allowing efficient creation and modification of graphics
  - improved default aesthetics when quickly plotting results
  - **declarative** programming, compared to base R’s **imperative** programming
  - emphasis on the fundamental visual decisions required for interpreting data
  - efficient syntax for faceting plots, partition data via grouping, layering graphics, and adding legends
Why use ggplot2?

• If you are new to creating graphics in R, you will need to learn the key commands for either ggplot2 or base R

• Compared to base R, ggplot2 provides...
  • a unified collection of components and options allowing efficient creation and modification of graphics
  • improved default aesthetics when quickly plotting results
  • declarative programming, compared to base R’s imperative programming
  • emphasis on the fundamental visual decisions required for interpreting data
  • efficient syntax for faceting plots, partition data via grouping, layering graphics, and adding legends
  • a slightly steeper learning curve for creating simple graphics.
Base R vs ggplot2 (Simple)

For simple plots, both base R and ggplot2 offer comparable results:
For simple plots, both base R and ggplot2 offer comparable results:

**Base R**

```r
plot(mpg ~ hp, data = mtcars)
```

**ggplot2**

```r
ggplot(mtcars, aes(x = hp, y = mpg)) + geom_point()
```
Base R vs ggplot2 (Simple)

For simple plots, both base R and ggplot2 offer comparable results:

Base R

```r
plot(mpg ~ hp, data = mtcars)
```

![Base R plot](image1)

ggplot2

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

![ggplot2 plot](image2)

But both plots are rather lacking: color? legend? title? better axis names?
Base R vs ggplot2 (Advanced)

But ggplot requires less code to customize
But ggplot requires less code to customize

**Base R**

```r
par(mar = c(4,4,2,.1))
plot(mpg~hp, data = subset(mtcars, am == 1),
    main="A comparison of hp and mpg",
    xlim=c(50,450),ylim=c(5,40) )
predicts(mpg~hp, col ="red",
data = subset(mtcars, am == 0))
legend(370, 38,
c(1", "0"), title ="transmission",
col = c("black", "red"),
pch = c(1,1))
```

**ggplot2**

```r
ggplot(mtcars, aes(x = hp, y = mpg, color = factor(am)))+
geom_point()+
labs(x = "horsepower", y = "miles per gallon",
title = "A comparison of hp and mpg",
color = "transmission")
```

(Not pictured: 20 minutes of frustration trying to reposition title and legend so they don't overlap with boundary)
Base R vs ggplot2 (Variant)

And ggplot2 doesn't necessitate a code rewrite when changing graphics
Base R vs ggplot2 (Variant)

And ggolot2 doesn’t necessitate a code rewrite when changing graphics

**Base R**

```r
par(par = c(2,2,2,2))
boxplot(mpg~ am, data = mtcars,
       main="A comparison of mpg and transmission",
       xlab="transmission",
       ylab = "miles per gallon"
)
```

(Note: almost all elements of the Base R code had to be modified to produce the above boxplot)

**ggplot2**

```r
ggplot(mtcars, aes( y = mpg, x = factor(am))) +
geom_boxplot()+
labs(x = "transmission", y = "miles per gallon",
     title = "A comparison of mpg and transmission")
```

(Here, we replaced `geom_point()` with `geom_boxplot`, and changed the `x = hp` to `x = am`)
Section 2

The Grammar of Graphics
The Grammar of Graphics provides a framework for assembling disparate elements of a statistical visualization into a coherent whole, under the guiding principle:

- **Data** consists of observations from the variables of interest (often encoded as a data frame)
- **Geometric objects** (or **geom**) are the distinct shapes present in the plot (lines, bars, points, etc.)
- **Aesthetic attributes** ([aes]) are the visual properties of the geometric objects, like x position, y position, color, fill, shape
- **Mappings** are the particular methods by which the values of the variables in the data control the attributes of the geometric objects
The Guiding Principle

The Grammar of Graphics provides a framework for assembling disparate elements of a statistical visualization into a coherent whole, under the guiding principle:

A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.
The Guiding Principle

The Grammar of Graphics provides a framework for assembling disparate elements of a statistical visualization into a coherent whole, under the guiding principle:

A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.

- **Data** consists of observations from the variables of interest (often encoded as a data frame)
The Guiding Principle

The Grammar of Graphics provides a framework for assembling disparate elements of a statistical visualization into a coherent whole, under the guiding principle:

*A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.*

- **Data** consists of observations from the variables of interest (often encoded as a data frame)
- **geometric objects** (or geoms) are the distinct shapes present in the plot (lines, bars, points, etc.)
The Grammar of Graphics provides a framework for assembling disparate elements of a statistical visualization into a coherent whole, under the guiding principle:

A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.

- **Data** consists of observations from the variables of interest (often encoded as a data frame)
- **geometric objects** (or geoms) are the distinct shapes present in the plot (lines, bars, points, etc.)
- **aesthetic attributes** are (aes) are the visual properties of the geometric objects, like x position, y position, color, fill, shape
The Guiding Principle

The Grammar of Graphics provides a framework for assembling disparate elements of a statistical visualization into a coherent whole, under the guiding principle:

*A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.*

- **Data** consists of observations from the variables of interest (often encoded as a data frame)
- **geometric objects** (or geoms) are the distinct shapes present in the plot (lines, bars, points, etc.)
- **aesthetic attributes** are (aes) are the visual properties of the geometric objects, like x position, y position, color, fill, shape
- **mappings** are the particular methods by which the values of the variables in the data control the attributes of the geometric objects
Deconstructing a graphic using GoG

1. What are the variables here?

2. What *geoms* do the variables map to?

3. What are the *aesthetics* of the *geom*? Which variables set the values of those *aesthetic*?
Anatomy of the ggplot2 graphic

- The code for graphics will (almost) always take the following general form:

```r
### ----- to be filled in by user
ggplot(data = -----, mapping = aes(-----)) +
   geom_-----(-----) +

   geom_-----(-----) +
   scale_-----(-----) +
   labs(-----) +
   coord_-----(-----)
```
Anatomy of the ggplot2 graphic

- The code for graphics will (almost) always take the following general form:

```r
### ----- to be filled in by user
ggplot(data = -----, mapping = aes(-----)) +
   geom_-----(-----) +

   geom_-----(-----) +
   scale_-----(-----) +
   labs(-----) +
   coord_-----(-----)
```

- Recall the guiding principle:

> A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.
Anatomy of the ggplot2 graphic

- The code for graphics will (almost) always take the following general form:

```r
### ----- to be filled in by user
ggplot(data = -----, mapping = aes(-----)) +
geom_-----(-----) +
geom_-----(-----) +
scale_-----(-----) +
labs(-----) +
coord_-----(-----)
```

- Recall the guiding principle:

  *A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.*
A Short Example

• Consider the following data

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>diameter</th>
<th>rotation</th>
<th>rings</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>Terrestrial planet</td>
<td>0.382</td>
<td>58.64</td>
<td>FALSE</td>
<td>0.4</td>
</tr>
<tr>
<td>Venus</td>
<td>Terrestrial planet</td>
<td>0.949</td>
<td>-243.02</td>
<td>FALSE</td>
<td>0.7</td>
</tr>
<tr>
<td>Earth</td>
<td>Terrestrial planet</td>
<td>1.000</td>
<td>1.00</td>
<td>FALSE</td>
<td>1.0</td>
</tr>
<tr>
<td>Mars</td>
<td>Terrestrial planet</td>
<td>0.532</td>
<td>1.03</td>
<td>FALSE</td>
<td>1.5</td>
</tr>
<tr>
<td>Jupiter</td>
<td>Gas giant</td>
<td>11.209</td>
<td>0.41</td>
<td>TRUE</td>
<td>5.2</td>
</tr>
<tr>
<td>Saturn</td>
<td>Gas giant</td>
<td>9.449</td>
<td>0.43</td>
<td>TRUE</td>
<td>9.5</td>
</tr>
<tr>
<td>Uranus</td>
<td>Gas giant</td>
<td>4.007</td>
<td>-0.72</td>
<td>TRUE</td>
<td>19.2</td>
</tr>
<tr>
<td>Neptune</td>
<td>Gas giant</td>
<td>3.883</td>
<td>0.67</td>
<td>TRUE</td>
<td>30.1</td>
</tr>
</tbody>
</table>
A Short Example

• Consider the following data

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>diameter</th>
<th>rotation</th>
<th>rings</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>Terrestrial planet</td>
<td>0.382</td>
<td>58.64</td>
<td>FALSE</td>
<td>0.4</td>
</tr>
<tr>
<td>Venus</td>
<td>Terrestrial planet</td>
<td>0.949</td>
<td>-243.02</td>
<td>FALSE</td>
<td>0.7</td>
</tr>
<tr>
<td>Earth</td>
<td>Terrestrial planet</td>
<td>1.000</td>
<td>1.00</td>
<td>FALSE</td>
<td>1.0</td>
</tr>
<tr>
<td>Mars</td>
<td>Terrestrial planet</td>
<td>0.532</td>
<td>1.03</td>
<td>FALSE</td>
<td>1.5</td>
</tr>
<tr>
<td>Jupiter</td>
<td>Gas giant</td>
<td>11.209</td>
<td>0.41</td>
<td>TRUE</td>
<td>5.2</td>
</tr>
<tr>
<td>Saturn</td>
<td>Gas giant</td>
<td>9.449</td>
<td>0.43</td>
<td>TRUE</td>
<td>9.5</td>
</tr>
<tr>
<td>Uranus</td>
<td>Gas giant</td>
<td>4.007</td>
<td>-0.72</td>
<td>TRUE</td>
<td>19.2</td>
</tr>
<tr>
<td>Neptune</td>
<td>Gas giant</td>
<td>3.883</td>
<td>0.67</td>
<td>TRUE</td>
<td>30.1</td>
</tr>
</tbody>
</table>

• How would we construct a plot of planet size vs. distance from sun?
A Short Example

- Consider the following data

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>diameter</th>
<th>rotation</th>
<th>rings</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>Terrestrial planet</td>
<td>0.382</td>
<td>58.64</td>
<td>FALSE</td>
<td>0.4</td>
</tr>
<tr>
<td>Venus</td>
<td>Terrestrial planet</td>
<td>0.949</td>
<td>-243.02</td>
<td>FALSE</td>
<td>0.7</td>
</tr>
<tr>
<td>Earth</td>
<td>Terrestrial planet</td>
<td>1.000</td>
<td>1.00</td>
<td>FALSE</td>
<td>1.0</td>
</tr>
<tr>
<td>Mars</td>
<td>Terrestrial planet</td>
<td>0.532</td>
<td>1.03</td>
<td>FALSE</td>
<td>1.5</td>
</tr>
<tr>
<td>Jupiter</td>
<td>Gas giant</td>
<td>11.209</td>
<td>0.41</td>
<td>TRUE</td>
<td>5.2</td>
</tr>
<tr>
<td>Saturn</td>
<td>Gas giant</td>
<td>9.449</td>
<td>0.43</td>
<td>TRUE</td>
<td>9.5</td>
</tr>
<tr>
<td>Uranus</td>
<td>Gas giant</td>
<td>4.007</td>
<td>-0.72</td>
<td>TRUE</td>
<td>19.2</td>
</tr>
<tr>
<td>Neptune</td>
<td>Gas giant</td>
<td>3.883</td>
<td>0.67</td>
<td>TRUE</td>
<td>30.1</td>
</tr>
</tbody>
</table>

- How would we construct a plot of planet size vs. distance from sun?

- data  
  - Planet Distance  
  - Planet Diameter

- aesthetics  
  - x position  
  - y position

- geometric object  
  - point  
  - point

Jonathan Wells  (Reed Empirical Workshop Series)  
PrettyR Graphics with ggplot2
A Short Example

```r
ggplot(data = planets, mapping = aes(x = distance, y = diameter)) + geom_point()
```
A Short Example

```r
ggplot(data = planets, mapping = aes(x = distance, y = diameter, color = type)) +
  geom_point() +
  geom_text(aes(label = name), size = 3) +
  labs(title = "The Planets")
```

![The Planets graph](image-url)
Section 3

The Core Graphics
The Data

Run the following code to import the biketown data into R:

```r
biketown <-
  read_csv("https://raw.githubusercontent.com/ChalkboardSonata/ERWS_ggplot/master/biketown10k.csv")
```
The Data

Run the following code to import the biketown data into R:

```r
biketown <- read_csv("https://raw.githubusercontent.com/ChalkboardSonata/ERWS_ggplot/master/biketown10k.csv")
```

The data consists of a subset of Summer 2017 bike rentals from Portland's bikeshare program.
The Data

Run the following code to import the biketown data into R:

```r
biketown <- read_csv("https://raw.githubusercontent.com/ChalkboardSonata/ERWS_ggplot/master/biketown10k.csv")
```

The data consists of a subset of Summer 2017 bike rentals from Portland’s bikeshare program.

`glimpse(biketown)`

```r
## Observations: 10,000
## Variables: 19
## $ RouteID <dbl> 4441750, 3626477, 4417818, 3834926, 3828348, ...
## $ PaymentPlan <chr> "Casual", "Casual", "Casual", "Subscriber", "...
## $ StartHub <chr> NA, "NW Everett at 22nd", "NW 11th at The Fie...
## $ StartLatitude <dbl> 45.52288, 45.52488, 45.53233, 45.52990, 45.529...
## $ StartLongitude <dbl> 45.52288, 45.52488, 45.53233, 45.52990, 45.555...
## $ StartLongidue <dbl> -122.6359, -122.6965, -122.6824, -122.6628, -122.66...
## $ StartDate <chr> "9/12/2017", "7/16/2017", "9/10/2017", "7/30/2...
## $ StartTime <time> 12:47:00, 12:19:00, 15:51:00, 21:42:00, 11:1...
## $ EndHub <chr> "SE 40th at Hawthorne", "NW Kearney at 23rd", "...
## $ EndLatitude <dbl> 45.51186, 45.52914, 45.55076, 45.51937, 45.558...
## $ EndLongitude <dbl> -122.6210, -122.6987, -122.6669, -122.6607, -122.6607...
## $ EndDate <chr> "9/12/2017", "7/16/2017", "9/10/2017", "7/30/2...
## $ EndTime <time> 13:02:00, 12:25:00, 16:17:00, 22:03:00, 12:0...
## $ TripType <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...  
## $ BikeID <dbl> 6163, 7182, 6165, 6098, 6379, 6167, 6554, 607...
## $ BikeName <chr> "0488 BIKETOWN", "0327 BIKETOWN", "0613 BIKET...
## $ Distance_Miles <dbl> 1.53, 0.42, 2.17, 0.80, 7.99, 0.13, 3.96, 2.1...
## $ Duration <time> 00:14:25, 00:06:20, 00:25:32, 00:21:08, 00:5...
## $ RentalAccessPath <chr> "keypad", "keypad", "keypad", "keypad_rfid_ca...
## $ MultipleRental <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...
```
The Bar Chart

- Investigate the distribution of bike use by month
The Bar Chart

- Investigate the distribution of bike use by month

```r
ggplot(data = biketown, mapping = aes(x = Month)) + geom_bar() + labs(title = "Bike use by month")
```

![Bike use by month](image_url)
Segmented / Stacked Bar Charts

- The biketown data includes 2 types of users: Casual Users and Subscribers. Do changes in monthly use vary by user type?
Segmented / Stacked Bar Charts

- The biketown data includes 2 types of users: Casual Users and Subscribers. Do changes in monthly use vary by user type?

```r
ggplot(data = biketown,
       mapping = aes(x = Month,
                      fill = PaymentPlan)) +
  geom_bar(position = "fill") +
  labs(title = "Segmented Bar Chart")
```

---

Jonathan Wells  (Reed Empirical Workshop Series)  PrettyR Graphics with ggplot2  4/23/20  20/42
Segmented / Stacked Bar Charts

- The biketown data includes 2 types of users: Casual Users and Subscribers. Do changes in monthly use vary by user type?

```r
ggplot(data = biketown,
       mapping = aes(x = Month,
                      fill = PaymentPlan)) +
geom_bar() +
labs(title = "Stacked Bar Chart")
```

- Each bar divided into count by `fill` variable.
Segmented / Stacked Bar Charts

• The biketown data includes 2 types of users: Casual Users and Subscribers. Do changes in monthly use vary by user type?

```r
ggplot(data = biketown, 
mapping = aes(x = Month, 
    fill = PaymentPlan)) + 
geom_bar() + 
labs(title = "Stacked Bar Chart")
```

• Each bar divided into count by `fill` variable.
• Hard to make direct comparisons
Segmented / Stacked Bar Charts

- The biketown data includes 2 types of users: Casual Users and Subscribers. Do changes in monthly use vary by user type?

```r
ggplot(data = biketown, 
      mapping = aes(x = Month, 
                      fill = PaymentPlan)) + 
geom_bar() + 
labs(title = "Stacked Bar Chart")
```

- Each bar divided into count by `fill` variable.
- Hard to make direct comparisons

```r
ggplot(data = biketown, 
      mapping = aes(x = Month, 
                      fill = PaymentPlan)) + 
geom_bar(position = "fill") + 
labs(title = "Segmented Bar Chart")
```
Segmented / Stacked Bar Charts

- The biketown data includes 2 types of users: Casual Users and Subscribers. Do changes in monthly use vary by user type?

```r
# Stacked Bar Chart
ggplot(data = biketown, 
       mapping = aes(x = Month, 
                      fill = PaymentPlan)) +
  geom_bar() +
  labs(title = "Stacked Bar Chart")
```

```r
# Segmented Bar Chart
ggplot(data = biketown, 
       mapping = aes(x = Month, 
                      fill = PaymentPlan)) +
  geom_bar(position = "fill") +
  labs(title = "Segmented Bar Chart")
```

- Each bar divided into count by `fill` variable.
- Hard to make direct comparisons

- Each bar divided into proportion by `fill` variable.
The Histogram

- What is the distribution of distance of bike trips?
• What is the distribution of distance of bike trips?

```r
ggplot(data = biketown, mapping = aes(x = Distance_Miles)) + geom_histogram()
```

```r
# `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```
Customizing Histograms

- The size of bins can be specified by the `bins =` or the `binwidth =` argument.

```r
ggplot(data = biketown, mapping = aes(x = Distance_Miles)) + geom_histogram(bins = 10)
```
Customizing Histograms

- The size of bins can be specified by the \texttt{bins =} or the \texttt{binwidth =} argument.

```r
ggplot(data = biketown, mapping = aes(x = Distance_Miles)) + geom_histogram(bins = 10)
```

![Histogram with bins = 10]

```r
ggplot(data = biketown, mapping = aes(x = Distance_Miles)) + geom_histogram(binwidth = 1)
```

![Histogram with binwidth = 1]
Customizing Histograms

- Improved readability can often be achieved by changing the fill and border color.
Customizing Histograms

- Improved readability can often be achieved by changing the fill and border color.

```r
ggplot(data = biketown, mapping = aes(x = Distance_Miles)) + geom_histogram(binwidth = 1, color = "white", fill = "steelblue")
```
Customizing Histogram

- To plot proportion rather than count, add `y = ..density..` to the aesthetic mapping.
Customizing Histogram

• To plot proportion rather than count, add `y = ..density..` to the aesthetic mapping.

• To superimpose a smoothed density curve, add a `geom_density()` layer.
Customizing Histogram

• To plot proportion rather than count, add `y = ..density..` to the aesthetic mapping.

• To superimpose a smoothed density curve, add a `geom_density()` layer

```r
ggplot(data = biketown, mapping = aes(x = Distance_Miles, y = ..density..)) + geom_histogram(binwidth = 1, color = "white", fill = "steelblue") + geom_density()
```

![Graph showing histogram and density curve](image-url)
The Boxplot

- How do distances of casual users and subscribers compare?
The Boxplot

- How do distances of casual users and subscribers compare?

```r
ggplot(data = biketown, mapping = aes(y = Distance_Miles, x = PaymentPlan)) + geom_boxplot() + labs(title = "Distance versus Rider Type")
```

Distance versus Rider Type

![Boxplot showing comparisons between casual and subscriber distances](image)
The Boxplot

- Vertically oriented boxplots can waste a lot of screen real estate. Create horizontally oriented plots by adding a `coord_flip()` layer.
Vertically oriented boxplots can waste a lot of screen real estate. Create horizontally oriented plots by adding a `coord_flip()` layer.

```r
ggplot(data = biketown, mapping = aes(y = Distance_Miles, x = PaymentPlan)) + geom_boxplot() + labs(title = "Distance versus Rider Type") + coord_flip()
```
The Scatterplot

- Do bike rides covering large distances have a longer duration?
The Scatterplot

- Do bike rides covering large distances have a longer duration?

```
ggplot(data = biketown, mapping = aes(x = Duration,
    y = Distance_Miles)) +
    geom_point() +
    geom_smooth(method = "lm") +
    labs(x = "Duration (seconds)", y = "Distance (miles)"")
```
The Scatterplot

- To fix overplotting, adjust transparency of points using the `alpha =` argument
The Scatterplot

- To fix overplotting, adjust transparency of points using the `alpha =` argument

```r
ggplot(data = biketown, mapping = aes(x = Duration,
                                       y = Distance_Miles)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm") +
  labs(x = "Duration (seconds)", y = "Distance (miles)")
```
The Scatterplot

- Does behavior change for different user types?
The Scatterplot

- Does behavior change for different user types?
- We can change the range of axes by adding a `xlim` or `ylim` layer
The Scatterplot

- Does behavior change for different user types?

- We can change the range of axes by adding a `xlim` or `ylim` layer

```r
ggplot(data = biketown, mapping = aes(x = Duration, y = Distance_Miles, color = PaymentPlan)) +
geom_point(alpha = 0.1) +
xlim(0, 2*60*60) +
ylim(0,10) +
labs(x = "Duration (seconds)", y = "Distance (miles)")
```

![Scatterplot diagram](image-url)
Summary

• The guiding principle of the Grammar of Graphics:

A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.
Summary

• The guiding principle of the Grammar of Graphics:

  A *statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.*

• Create statistical graphics by deciding which geometric objects and which aesthetic attributes you want to represent which variables
Summary

• The guiding principle of the Grammar of Graphics:

  A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.

• Create statistical graphics by deciding which geometric objects and which aesthetic attributes you want to represent which variables

• Iteratively apply layers to add elements to a plot
The guiding principle of the Grammar of Graphics:

*A statistical graphic is a mapping of data variables to aesthetic attributes of geometric objects.*

- Create statistical graphics by deciding which geometric objects and which aesthetic attributes you want to represent which variables
- Iteratively apply layers to add elements to a plot
- Modify graphics by tweaking values of arguments of each layer
Further Resources

- For helpful reminders (also available under the help tab in RStudio):
  
  The ggplot2 cheatsheet
Further Resources

- For helpful reminders (also available under the help tab in RStudio):
  
  The ggplot2 cheatsheet

- For a more thorough overview of the ggplot2 package:
  
  Ch. 3 in Wickham’s R for Data Science
Further Resources

- For helpful reminders (also available under the help tab in RStudio):
  The ggplot2 cheatsheet

- For a more thorough overview of the ggplot2 package:
  Ch. 3 in Wickham’s R for Data Science

- For detailed instructions for creating particular plots:
  Chang’s R Graphics Cookbook
Further Resources

- For helpful reminders (also available under the help tab in RStudio):
  
  The ggplot2 cheatsheet

- For a more thorough overview of the ggplot2 package:
  
  Ch. 3 in Wickham’s R for Data Science

- For detailed instructions for creating particular plots:
  
  Chang’s R Graphics Cookbook

- For the theoretical foundation of ggplot2:
  
  Wickham’s original paper A Layered Grammar of Graphics
Further Resources

• For helpful reminders (also available under the help tab in RStudio):
  The ggplot2 cheatsheet

• For a more thorough overview of the ggplot2 package:
  Ch. 3 in Wickham’s R for Data Science

• For detailed instructions for creating particular plots:
  Chang’s R Graphics Cookbook

• For the theoretical foundation of ggplot2:
  Wickham’s original paper A Layered Grammar of Graphics

• For other questions about today’s presentation, feel free to email me:
  wellsj@reed.edu
Section 4

Advanced Graphics
Facets

- How does the distribution of travel distance vary by month?
Facets

- How does the distribution of travel distance vary by month?
- Adding a `facet_wrap()` layer separates data into subplots for each value of the faceted variable
Facets

- How does the distribution of travel distance vary by month?
- Adding a `facet_wrap()` layer separates data into subplots for each value of the faceted variable.

```r
ggplot(data = biketown, mapping = aes(x = Distance_Miles, y = ..density..)) + geom_histogram(binwidth = 1, color = "white", fill = "steelblue") + facet_wrap(~Month)
```
Transformations

• How do averages and medians for bike distance compare between Casual Users and Subscribers?

```r
ggplot(data = biketown, mapping = aes(y = Distance_Miles, x = PaymentPlan)) + geom_boxplot() + labs(title = "Distance versus Rider Type") + coord_flip() + stat_summary(fun = "mean", geom = "point", size = 3, color = "red")
```
Transformations

- How do averages and medians for bike distance compare between Casual Users and Subscribers?
- A stat builds a new variable from the existing data
Transformations

- How do averages and medians for bike distance compare between Casual Users and Subscribers?

- A stat builds a new variable from the existing data

```r
ggplot(data = biketown, mapping = aes(y = Distance_Miles, x = PaymentPlan)) + geom_boxplot() + labs(title = "Distance versus Rider Type") + coord_flip() + stat_summary(fun = "mean", geom = "point", size = 3, color = "red")
```
Annotations

How strong is the linear relationship between distance and duration?
How strong is the linear relationship between distance and duration?

```r
model <- lm(Distance_Miles ~ Duration, biketown)
summary(model)
```

```r
## Call:
## lm(formula = Distance_Miles ~ Duration, data = biketown)
## ## Residuals:
##    Min     1Q  Median     3Q    Max
## -32.332 -0.742 -0.284  0.491 17.859
## ## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.173e+00  1.776e-02  66.05  <2e-16 ***
## Duration   5.078e-04  6.070e-06  83.66  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ## Residual standard error: 1.456 on 9991 degrees of freedom
## (7 observations deleted due to missingness)
## Multiple R-squared:  0.4119, Adjusted R-squared:  0.4119
## F-statistic: 6999 on 1 and 9991 DF,  p-value: < 2.2e-16
```
Annotations

We can use an `annotate()` layer to manually add the $R^2$ value to the plot.
We can use an `annotate()` layer to manually add the $R^2$ value to the plot

```r
ggplot(data = biketown, mapping = aes(x = Duration, y = Distance_Miles)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm") +
  labs(x = "Duration (seconds)", y = "Distance (miles)") +
  annotate(geom = "text", x = 15*3600, y = 15, label = "$R^2 = .41"\n```

**Annotations**

Jonathan Wells (Reed Empirical Workshop Series)
Annotations

Or can change the output to a formula by adding the `parse = T` argument

```r
ggplot(data = biketown, mapping = aes(x = Duration, y = Distance_Miles)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm") +
  labs(x = "Duration (seconds)", y = "Distance (miles)") +
  annotate(geom = "text", x = 15*3600, y = 15, label = "R^2 == .41", parse = T)
```

R^2 = 0.41

![Graph showing the relationship between duration and distance with a line of best fit and R^2 value.]
Most bike trips are short, but a few are very long. It may be informative to plot Distance and Duration on a log-log scale.
• Most bike trips are short, but a few are very long. It may be informative to plot Distance and Duration on a log-log scale.

```r
ggplot(data = biketown, mapping = aes(x = as.numeric(Duration), y = Distance_Miles)) +
    geom_point(alpha = 0.1) +
    geom_smooth(method = "lm") +
    labs(x = "Duration (seconds)", y = "Distance (miles)") +
    scale_x_log10() +
    scale_y_log10()
```
How will the output of the following two code chunks differ?

```
ggplot(data = biketown, mapping = aes(x = Duration, y = Distance_Miles)) + geom_point(alpha = 0.2, mapping = aes(color = PaymentPlan))
```

```
ggplot(data = biketown, mapping = aes(x = Duration, y = Distance_Miles)) + geom_point(alpha = 0.2, color = "blue")
```
How will the output of the following two code chunks differ?

```r
ggplot(data = biketown, mapping = aes(x = Duration, y = Distance_Miles)) + geom_point(alpha = 0.2, mapping = aes(color = PaymentPlan))
```

```
ggplot(data = biketown, mapping = aes(x = Duration, y = Distance_Miles)) + geom_point(alpha = 0.2, color = "blue")
```
• When mapping a variable to the color aesthetic, `ggplot2` uses a default color scale. But this can be modified by adding a `scale` layer:
When mapping a variable to the color aesthetic, ggplot2 uses a default color scale. But this can be modified by adding a scale layer:

```r
ggplot(data = biketown, mapping = aes(x = Month, fill = PaymentPlan)) +
  geom_bar(position = "fill") +
  labs(title = "Segmented Bar Chart") +
  scale_fill_brewer(palette = "Pastel2")
```
• When mapping a variable to the color aesthetic, ggplot2 uses a default color scale. But this can be modified by adding a scale layer:

```r
ggplot(data = biketown, mapping = aes(x = Month, fill = PaymentPlan)) + geom_bar(position = "fill") + labs(title = "Segmented Bar Chart") + scale_fill_brewer(palette = "Pastel2")
```

• For the complete list of palettes, run the code chunk:

```r
library(RColorBrewer)
display.brewer.all()
```
Themes

- Suppose you want to alter the overall appearance of plot according to a premade theme:
• Suppose you want to alter the overall appearance of plot according to a premade theme:

```r
library(ggthemes)

ggplot(data = biketown, mapping = aes(x = Distance_Miles )) + geom_histogram( bins = 20, color = "white") + labs(title = "Distance Traveled") + theme_tufte()
```

![Histogram of Distance Traveled](chart.png)