

Data Visualizations for Surveys Using ggplot2 & ggsurvey AAPOR 2022 Short Course

Brittany Alexander Ph.D.
Associate Statistician
Ipsos Public Affairs

May 2, 2022

Outline

Course Objectives

Introduction to ggplot2 and ggsurvey

Using ggplot2

Using ggsurvey for bar charts

Using ggsurvey for box plots

Using ggsurvey for histograms

Using ggsurvey for Hexplots

Table of Contents

Course Objectives

Introduction to ggplot2 and ggsurvey

Using ggplot2

Using ggsurvey for bar charts

Using ggsurvey for box plots

Using ggsurvey for histograms

Using ggsurvey for Hexplots

Course Objectives

1. Learn about the Grammar of Graphics that ggplot2 uses
2. Learn how to use ggplot2 to visualize data
3. Learn how to use ggsurvey to create weighted bar and histograms
4. Learn how to visualize crosstabs using ggsurvey
5. Learn how to make other plots using ggsurvey and ggplot2

RCloud Setup

Go to [a](#) and create a free account. This will give you access to an R Studio project with all the course materials pre-installed.

Table of Contents

Course Objectives

Introduction to ggplot2 and ggsurvey

Using ggplot2

Using ggsurvey for bar charts

Using ggsurvey for box plots

Using ggsurvey for histograms

Using ggsurvey for Hexplots

Required Software (all available free)

1. R (version 3.5 or newer required)
2. RStudio highly recommended
3. ggplot2 R package
4. dplyr R package
5. survey & srvyr R package
6. anesr R package
7. ggsurvey R package
8. palmerpenguins R package
9. stringr R package

Introducing ggplot2

- ▶ ggplot2 is an R package created by Hadley Wickham that uses the grammar of graphics
- ▶ ggplot2 is a part of the tidyverse which is a collection of R packages
- ▶ ggplot2 is more flexible than base R plotting
- ▶ ggplot2 looks more modern than base R plotting

Grammar of Graphics

- ▶ The philosophy of ggplot2 is laid out in Wickham (2010), and builds upon previous work by Wilkinson, Anand, and Grossman (2005), and Bertin (1983). (1983)
- ▶ The grammar of graphics is a framework for visualizations to break apart the elements of a plot:
 1. the data and the mapping of the aesthetic in the plot
 2. the geometric objects of a plot, (is it a bar chart or histogram or something else)
 3. the scale of the plots
 4. the facet specification (are we making separate plots for each level of another variable)

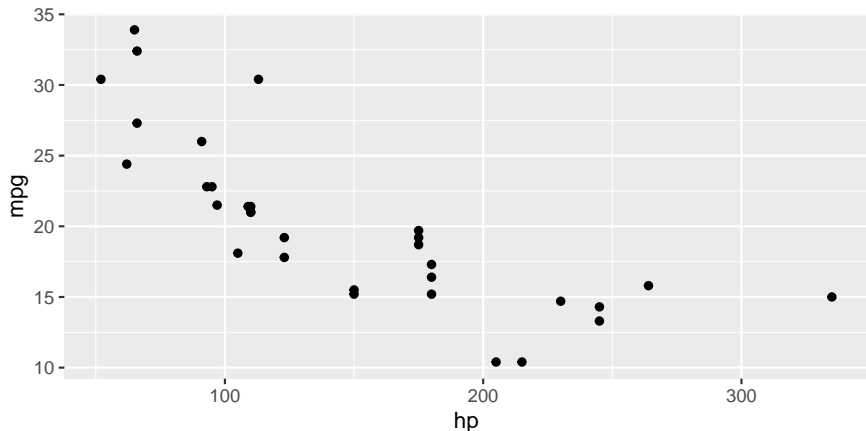
Dataset 1: mtcars

mtcars is a built in data set in R that takes data from the 1974 Motor Trend magazine with 11 variables relating to 32 automobile design and performance.

Variables: mpg (miles per gallon), cyl (number of cylinders), disp (displacement), hp (horsepower), drat (rear axle ratio), wt (weight in 1000lbs), qsec (0.25 mile time), vs (engine 0 = v-shape, 1 = straight), am (transmission (0=automatic, 1 = manual), gear (number of gears), carb (number of carburetors).

Example 1 ggplot2

```
library(ggplot2)
plot = ggplot(mtcars, aes(x = hp,
  y = mpg)) + geom_point()
plot
```



ggplot2 structure from the "Cheat Sheet"

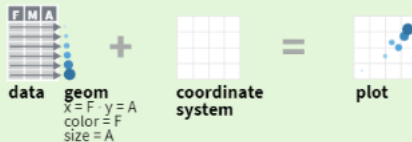
Tidyverse Cheat Sheets including ggplot2

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and **geoms**—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.



ggplot2 Options

ggplot2 book

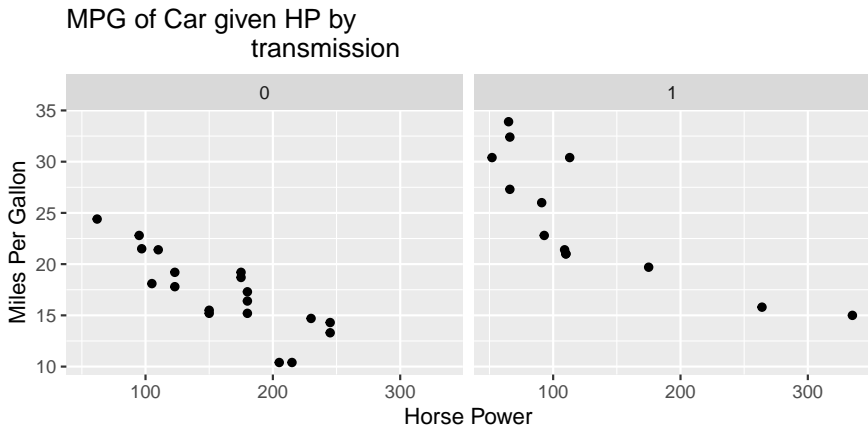
1. Plot types: density, histogram, boxplot, bar chart, violin plot, error bars for bar plots, maps
2. custom color schemes, add fill/color or shapes matching categorical variables
3. custom scales or coordinate styems for continuous variables
4. faceting to make plots for each level of a categorical variable
5. custom labels and legends

Example ggplot2 add facet and title

```
plotnew = plot + ggtitle("MPG of Car given HP by  
                        transmission") +  
  xlab("Horse Power") + ylab("Miles Per Gallon") +  
  facet_grid(~am)
```

Example ggplot2 add facet and title

```
plotnew
```



Why ggsurvey

- ▶ ggplot2 has all the necessary components to make visualizations of survey data but the code can get messy especially when handling weights or making crosstabs.
- ▶ ggplot2 defaults to showing categorical data via counts, but proportions are often of interest with survey data especially when making comparisons across groups of different size.
- ▶ ggsurvey creates functions to make common visualizations of survey data in a single function call while preserving some of the customization features in ggplot.
- ▶ ggsurvey retains the compatibility of ggplot2 with the rest of the tidyverse
- ▶ ggsurvey has functions to handle data frames (with weight as a variable) and functions that handle svy.design objects from survey package.

ggsurvey Features

- ▶ ggsurvey plot types: bar chart, histogram, boxplot, hexplot
- ▶ has faceting built in using 2d (one facets), 3d (two facets)
- ▶ uses weight argument in ggplot to handle weights
- ▶ ggsurvey is a series of "wrapper" functions that call ggplot or other tidyverse functions
- ▶ ggsurvey works with unweighted data if you add a "weight" variable that is 1 for every row

Dataset 2: apistrat from the survey package

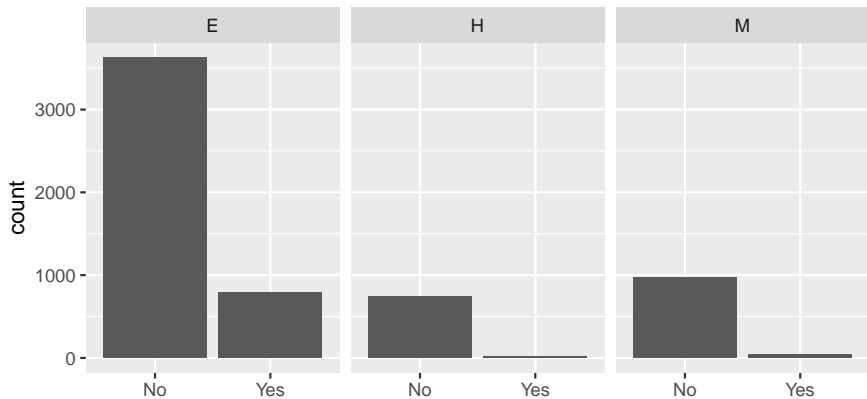
The second dataset we will use is the apistrat data set from the survey package. This is a stratified sample from a data set of all California schools with various characteristics of the schools such as type of school (stype), the Academic Performance index in 1999 and 2000 (api99, api00), if a school is year round (yr.rnd), percent of parents that are high school graduates (hsg)

```
library(survey)
data(api)
library(dplyr)
```

Example 2A Crosstabs with base ggplot

By default, ggplot does not adjust for varying sample sizes across graphics which doesn't always lead to good comparisons.

```
ggplot(apistrat, aes(yr.rnd)) +  
  geom_bar(aes(weight = pw)) +  
  facet_grid(cols = vars(stype))
```



Example 2A Crosstabs with base ggplot

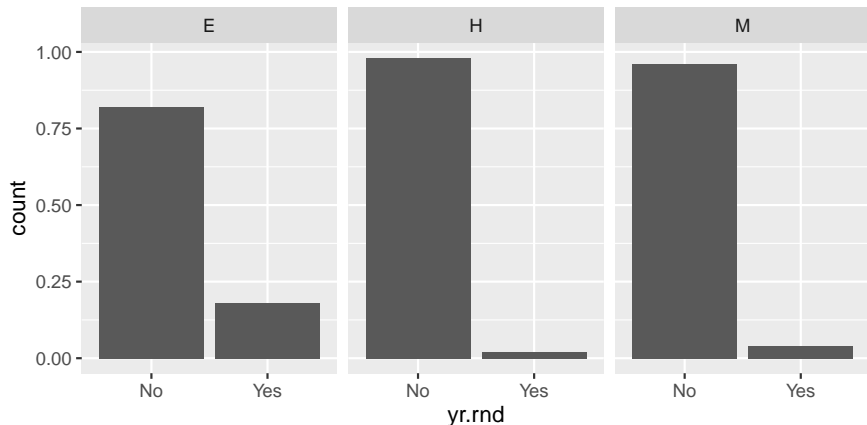
First let's look at the base ggplot2 code needed to produce a crosstab.

```
newdf = apistrat %>%  
  group_by(stype, yr.rnd) %>%  
  tally(, wt = pw) %>%  
  mutate(f = n/sum(n))  
plotnew = ggplot(newdf, aes(yr.rnd)) +  
  geom_bar(aes(weight = f)) +  
  facet_grid(cols = vars(stype))
```

Example 2A Crosstabs with base ggplot

First let's look at the base ggplot2 code needed to produce a crosstab.

```
plotnew
```



Example 2B Crosstabs with ggsurvey

```
library(ggsurvey)  
ggbarcrosstabs(apistrat, yr.rnd,  
               stype, pw)
```

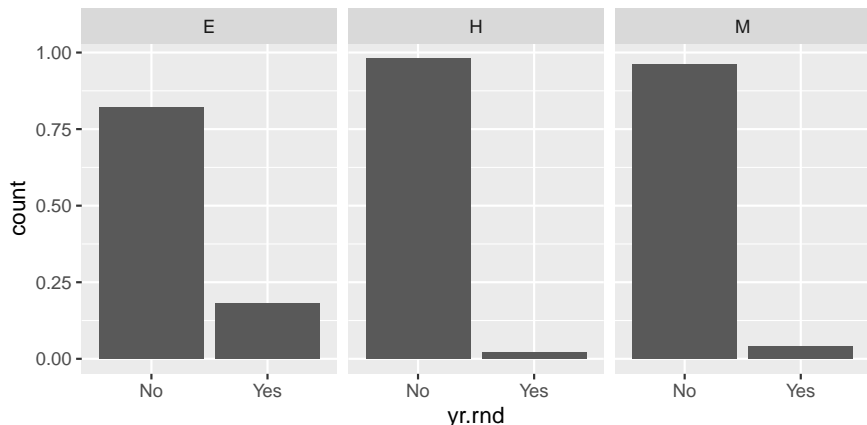


Table of Contents

Course Objectives

Introduction to ggplot2 and ggsurvey

Using ggplot2

Using ggsurvey for bar charts

Using ggsurvey for box plots

Using ggsurvey for histograms

Using ggsurvey for Hexplots

To understand the what can be done with ggsurvey, we will first look at how base ggplot2 works for non-survey data.

In base ggplot2 a plot is made with at least two functions: ggplot and an appropriate geom function. There are some geoms that are commonly used in addition to other geoms such as adding text or error bars.

Main Plot Geoms

	Geom function
Bar chart	geom_bar (for raw data), geom_col (for frequency tables)
Boxplot	geom_boxplot
Histogram	geom_histogram
Line Plot	geom_line
Scatterplot	geom_point
Hex plot	geom_hexbin (in hexbin package)
Violin plot	geom_violin

Supplemental Plot Geoms

	Geom
Error bars	geom_errorbars
Text	geom_text
Smoothing lines	geom_smooth
Line	geom_abline
horizontal line	geom_hline
vertical line	geom_vline

Dataset 3: Palmer Penguins

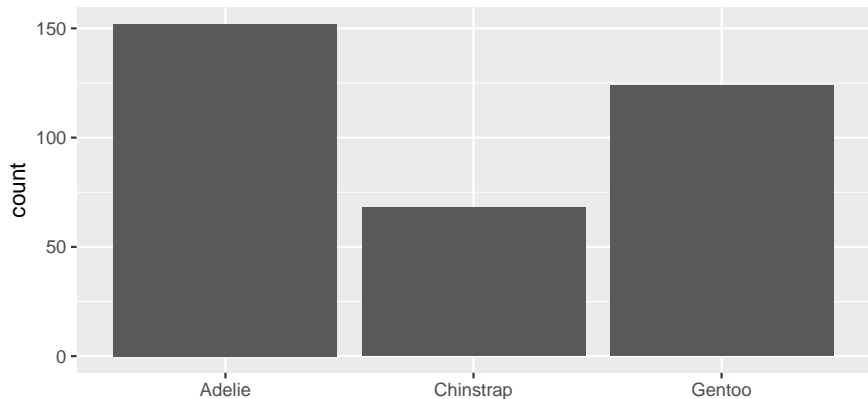
The Palmer penguins data set details information taken from penguins on the palmer archipelago. The variables are species, island, bill_length_mm, bill_depth_mm, flipper_depth_mm, flipper_length_mm, body_mass_g, sex, year.

```
library(palmerpenguins)
data(penguins)
```

Example 3: A Bar Chart

First let's use the `ggplot` and `geom_bar` functions to make a bar chart of the species of penguins in the data set.

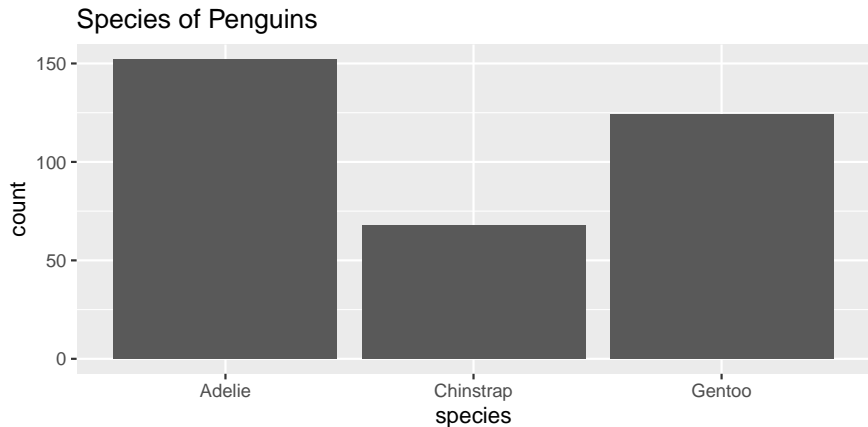
```
plot3 = ggplot(penguins, aes(species)) +  
  geom_bar()  
plot3
```



Title customization for Example 3

Next let's add a title to the previous plot with `ggtitle`.

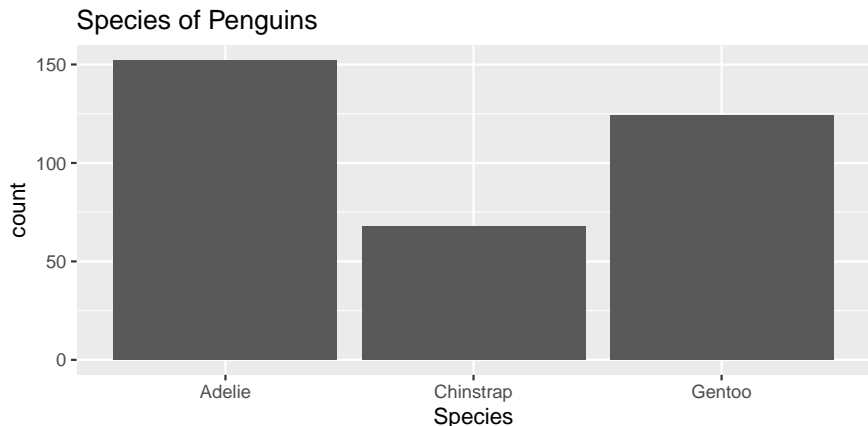
```
plot3 + ggtitle("Species of Penguins")
```



Label customization Part 1 for Example 3

Next let's add a x axis label with `xlab` adds a label to the x axis.

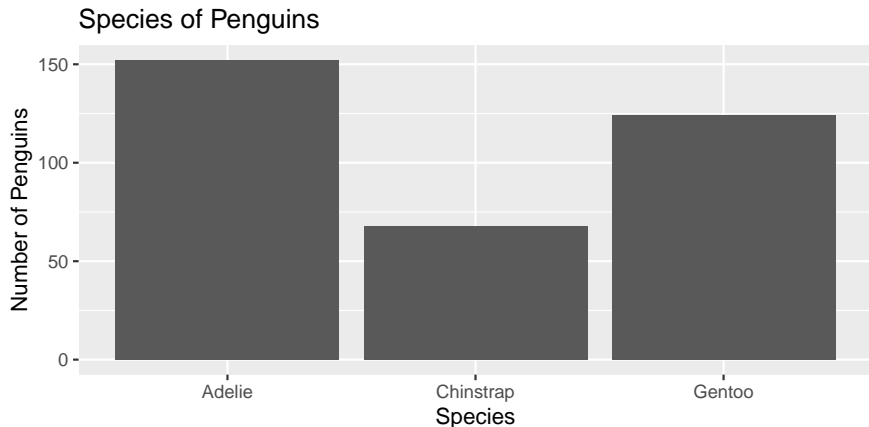
```
plot3 + ggtitle("Species of Penguins") +  
  xlab("Species")
```



Label customization Part 2 for Example 3

Next let's add a y axis label to the previous plot with `ylab`.

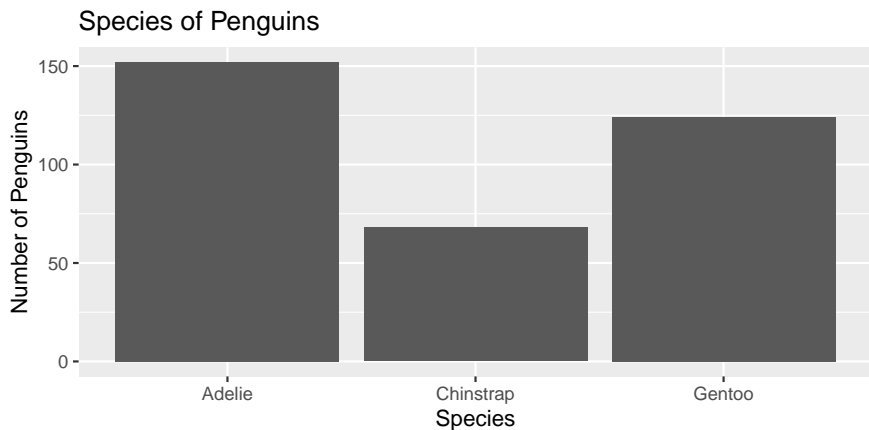
```
plot3title = plot3 + ggtitle("Species of Penguins") +  
  xlab("Species") + ylab("Number of Penguins")  
plot3title
```



Label customization Part 2 for Example 3

Next let's add a y axis label to the previous plot with ylab.

```
plot3title
```



Color customization

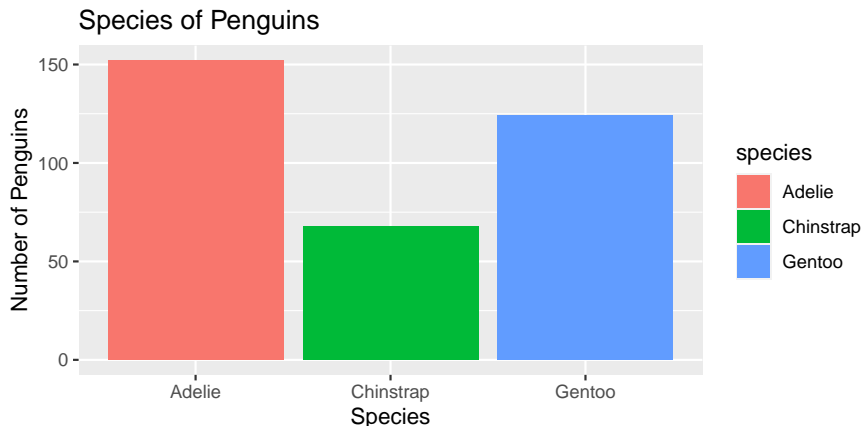
We can make the bars each a different color by passing `aes(fill = Species)` inside `geom_bar`

```
plot3color = ggplot(penguins, aes(species)) +  
  geom_bar(aes(fill = species)) +  
  ggtitle("Species of Penguins") +  
  xlab("Species") + ylab("Number of Penguins")
```

Color customization

We can make the bars each a different color by passing `aes(fill = Species)` inside `geom_bar`

```
plot3color
```



Faceting

Faceting is a common tool to compare the the distribution of one set of variables by considering another variable. Going back to Example 1, we compared the relationship between horsepower and mpg depending on the transmission of the car using `facet_grid`. Faceting helps to show the affect of a variable on multiple variables. Crosstabs are faceted graphs comparing the distribution of a survey question given an categorical variable.

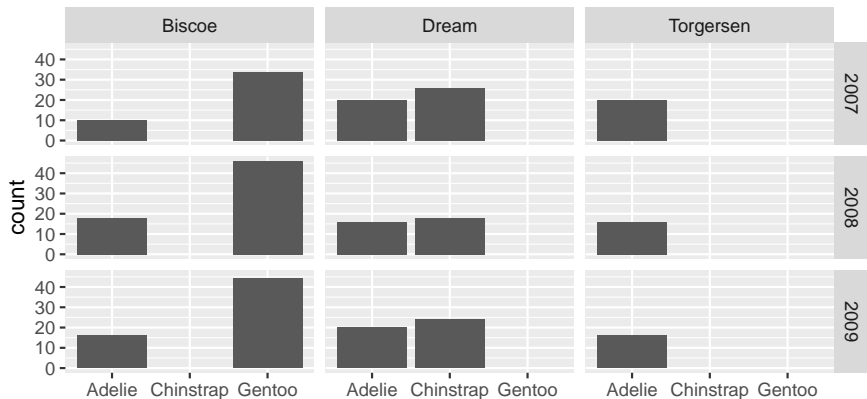
Faceting

```
plotnew = plot + ggtitle("MPG of Car given HP by  
                        transmission") +  
  xlab("Horse Power") + ylab("Miles Per Gallon") +  
  facet_grid(~am)
```

Example 4 Bar chart of Species with Facet Grid

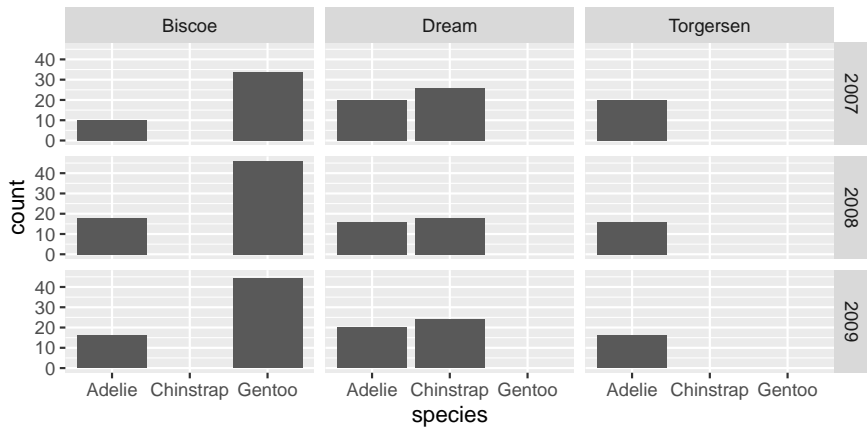
y x means x is on the horizontal axis

```
plot4 = ggplot(penguins, aes(species)) +  
  geom_bar() + facet_grid(year ~  
    island)  
plot4
```



Example 4 Bar chart of Species with Facet Grid

plot4

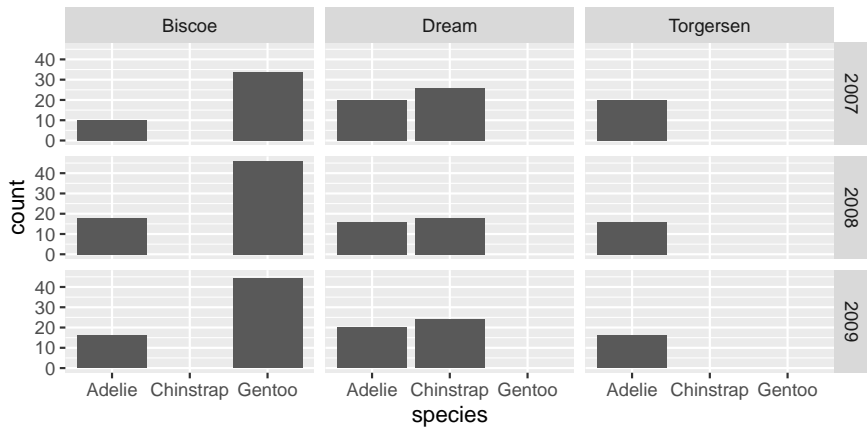


Example 4 Bar chart of Species with Facet Grid

```
plot4a = ggplot(penguins, aes(species)) +  
  geom_bar() + facet_grid(rows = vars(year),  
    cols = vars(island))
```

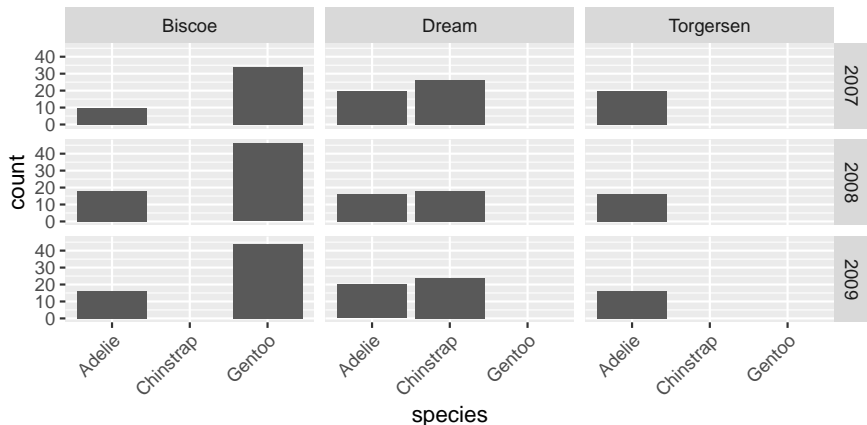
Example 4 Bar chart of Species with Facet Grid

plot4a



Example 4 Bar chart text wrapping

```
library(ggrepel)
plot4 + theme(axis.text.x = element_text(angle = 45,
  hjust = 1))
```



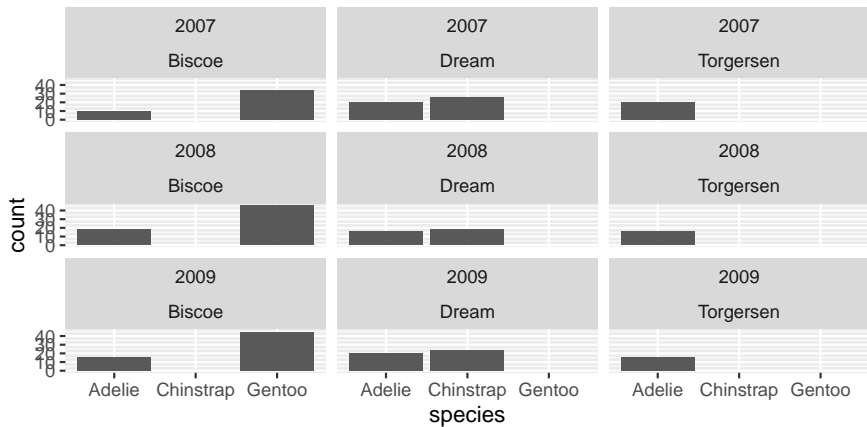
Facet Wrap

Facet Wrap wraps the facet for when there are too many columns/rows

```
plot4 = ggplot(penguins, aes(species)) +  
  geom_bar() + facet_wrap(vars(year,  
    island))
```

Facet Wrap

plot4

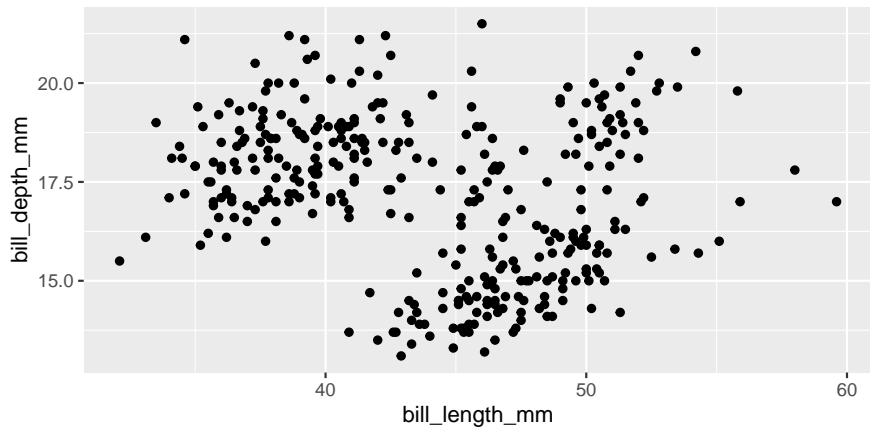


Example 5 Scatterplots

```
plot5 = ggplot(penguins, aes(x = bill_length_mm,  
  y = bill_depth_mm)) + geom_point()
```

Example 5 Scatterplots

plot5

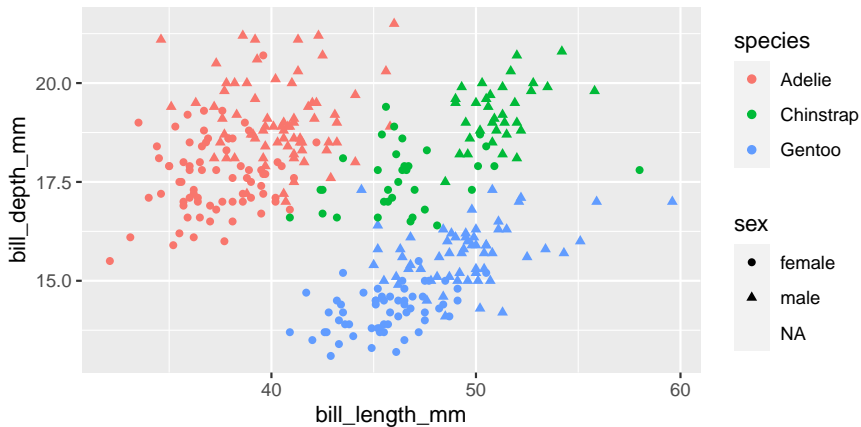


Example 5 with Species and Sex

```
plot5a = ggplot(penguins, aes(x = bill_length_mm,  
  y = bill_depth_mm)) + geom_point(aes(color = species,  
  shape = sex))
```

Example 5 with Species and Sex

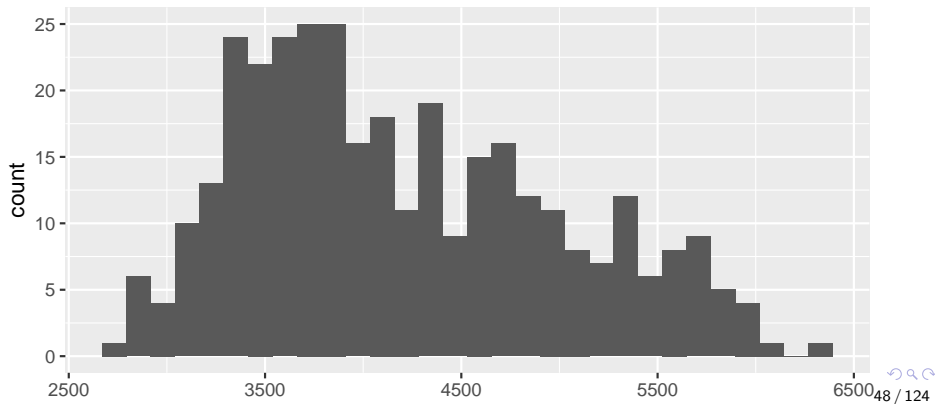
plot5a



Example 6A Histograms

A histogram takes a continuous variable and puts it into bins and then counts the number of data points in each bin. By default `geom_histogram()` makes 30 evenly sized bins.

```
ggplot(penguins, aes(x = body_mass_g)) +  
  geom_histogram()
```



Example 6B Histograms

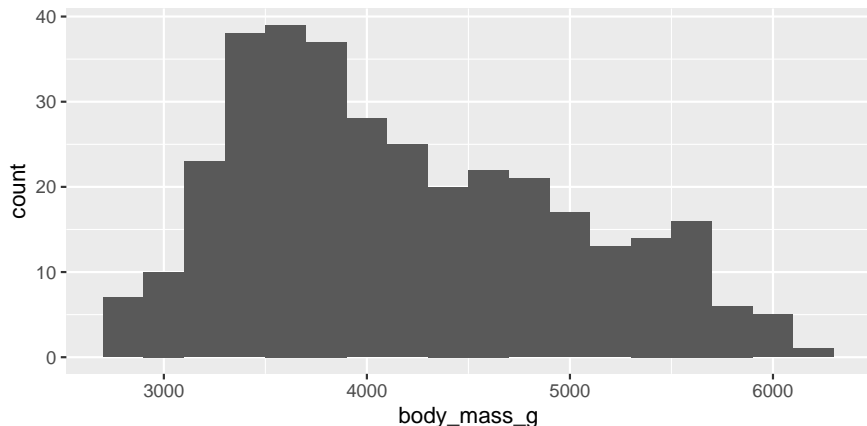
We can specify the size of the bins with `binwidth`. I prefer to try the default bins to get a sense of the range of the data.

```
plot6b = ggplot(penguins, aes(x = body_mass_g)) +  
  geom_histogram(binwidth = 200)
```

Example 6B Histograms

We can specify the size of the bins with `binwidth`. I prefer to try the default bins to get a sense of the range of the data.

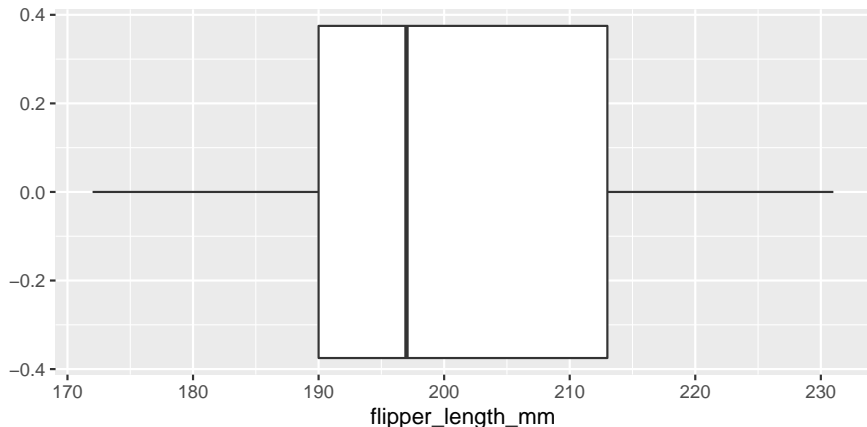
```
plot6b
```



Example 7 Boxplot

A boxplot shows the minimum, 25th percentile, median, 75th percentile, and maximum.

```
ggplot(penguins, aes(x = flipper_length_mm)) +  
  geom_boxplot()
```

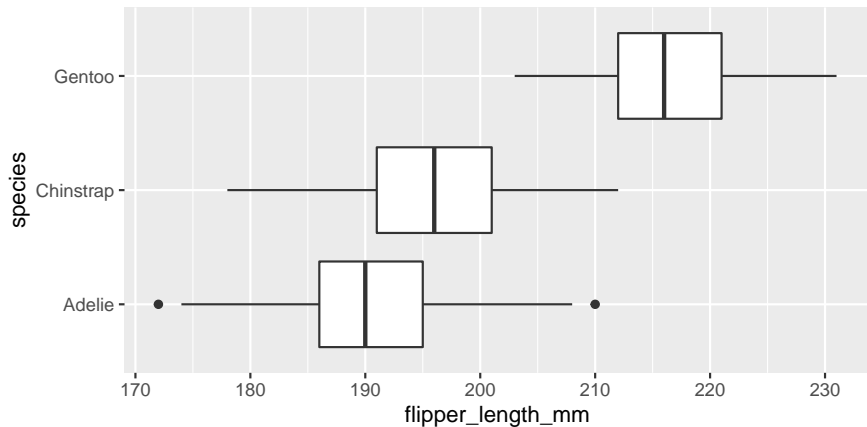


Example 7 Boxplot by Species

```
plot7 = ggplot(penguins, aes(x = flipper_length_mm,  
  y = species)) + geom_boxplot()
```

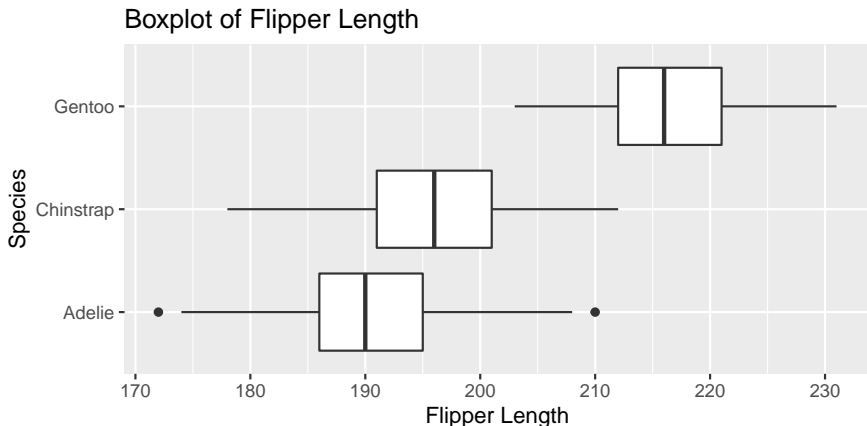
Example 7 Boxplot by Species

plot7



Example 7 with Title

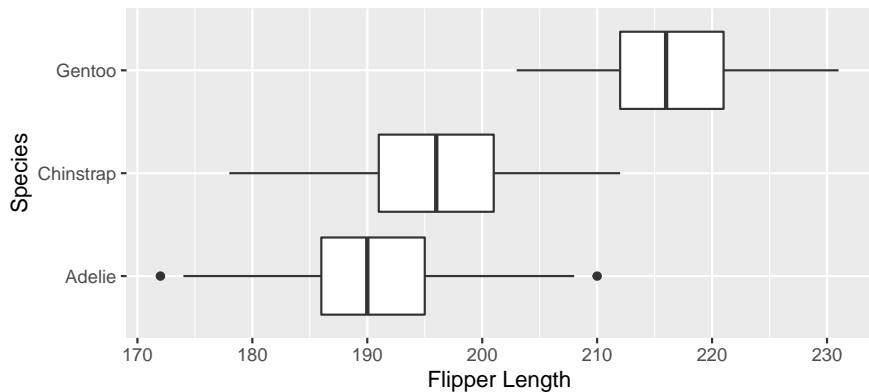
```
plot7new = plot7 + ggtitle("Boxplot of Flipper Length") +  
  xlab("Flipper Length") + ylab("Species")  
plot7new
```



Example 7 with Title

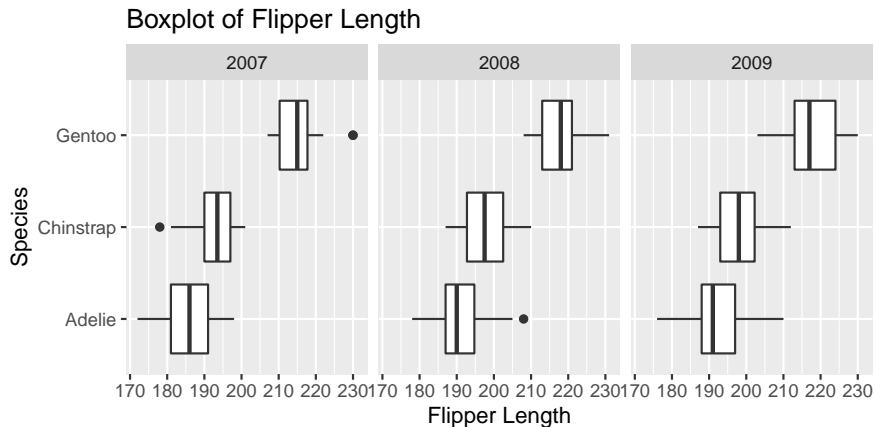
```
plot7new
```

Boxplot of Flipper Length



Example 7 with Facet for Year

```
plot7new + facet_grid(~year)
```



Any Questions?

Table of Contents

Course Objectives

Introduction to ggplot2 and ggsurvey

Using ggplot2

Using ggsurvey for bar charts

Using ggsurvey for box plots

Using ggsurvey for histograms

Using ggsurvey for Hexplots

Introduction to ggsurvey vs ggplot

- ▶ ggsurvey combines the ggplot+geom+facet_grid into one function to save time
- ▶ For each plot type ggsurvey specifies how to do it without a facet, with one facet, and with two facets
- ▶ You can add ggtitle, xlab, ylab just like regular ggplot2
- ▶ ggsurvey is compatible with dataframe and survey design objects (including srvyr objects for select functions)
- ▶ like in ggplot we don't put column names in quotes
- ▶ For data frames the survey weight should be a column in the data frame

Bar Chart Functions

Plot type	Data type	ggsurvey function
Bar Chart	data frame	<code>ggbarweight(df, x, weights, fill)</code>
Bar Chart	svydesign	<code>ggbarweight_svy(surveyobj, x, fill)</code>
Crosstab	data frame	<code>ggcrosstab(df, x, y, weights, fill, labeller)</code>
Crosstab	svydesign	<code>ggcrosstab_svy(surveyobj, x, y, fill, labeller)</code>
Crosstab with 2 variables	data frame	<code>ggcrosstab3d(df, x, y, z, weights, fill, labeller)</code>
Crosstab with 2 variables	svydesign	<code>ggcrosstab3d_svy(surveyobj, x, y, z, fill, labeller)</code>

Dataset 4: ANES

The American National Election Study is a large multi-mode multi-wave national survey conducted every two years with a broad range of questions related both to elections and broader policy issues. It is a collaboration between Stanford University and the University of Michigan and is funded by the National Science Foundation. It has been conducted since 1948. The population for the survey is US citizens.

[ANES information](#) The `anesr` package accesses ANES data, and we will use the 2020 time series data which has 8280 respondents.

Install and Load ANES

```
# install.packages('devtools')  
# library(devtools)  
# install_github('jamesmartherus/anesr')  
library(anesr)  
data("timeseries_2020")  
timeseries_2020 = haven::as_factor(timeseries_2020)
```

Variables in ANES

We will look at the following questions in the pre-election phase: V201114 (Right Track / Wrong Track), V201390 (Speed of Response of Federal Government to COVID-19), V201324 (State of Economy), V201151 feeling thermometer for Joe Biden, V201152 feeling thermometer for Trump. We will also consider crosstabs on the following variables V201549x (race), V201511x (education), V201600 (gender). The relevant weight is V200010a, strata V200010d, cluster is V200010c.

Rename variables and Drop Empty Levels

```
timeseries_2020$rightwrongtrack = timeseries_2020$V201114  
timeseries_2020$speedcovid = timeseries_2020$V201390  
timeseries_2020$economy = timeseries_2020$V201324  
timeseries_2020$race = timeseries_2020$V201549x  
timeseries_2020$gender = timeseries_2020$V201600  
timeseries_2020 = droplevels(timeseries_2020)
```


Removing Refusals on Demographics

To make the plots look a little cleaner we will remove individuals who refused or said didn't know about their race, education, or gender, or refused one of our questions of interest.

```
timeseries_2020 = timeseries_2020 %>%  
  filter(race != "-9. Refused",  
         race != "-8. Don't know",  
         gender != "-9. Refused",  
         rightwrongtrack != "-9. Refused",  
         rightwrongtrack != "-8. Don't know",  
         economy != "-9. Refused",  
         speedcovid != "-9. Refused")  
timeseries_2020 = droplevels(timeseries_2020)
```

Creating a Survey Design Object For ANES

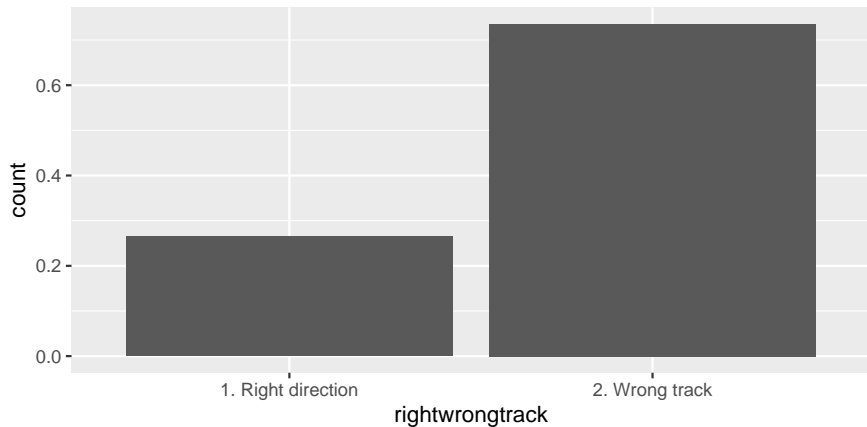
```
library(srvy)  
anes_survey = svydesign(id = ~V200001,  
  strata = ~V200010d, weights = ~V200010a,  
  data = timeseries_2020)  
anes_svy = as_survey(anes_survey)
```

Example 8A: ggsurvey Bar chart with Data Frame

```
plot8a = ggbarweight(timeseries_2020,  
  rightwrongtrack, V200010a)
```

Example 8A: ggsurvey Bar chart with Data Frame

plot8a

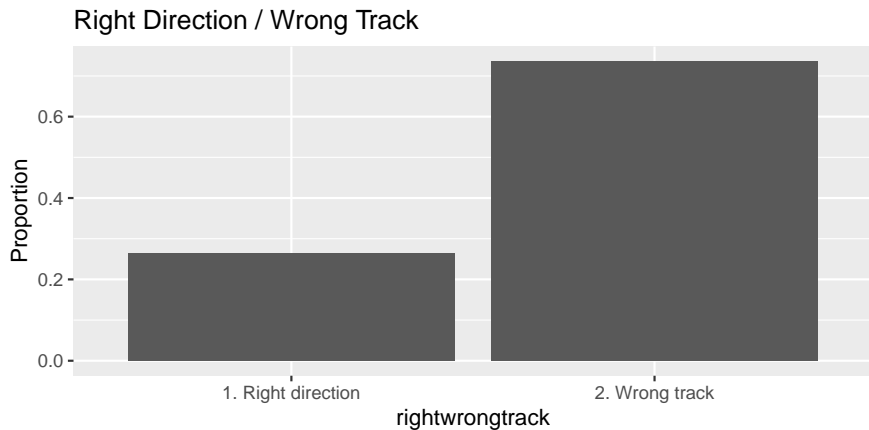


Example 8A: Adding titles to previous plot

```
plot8atitle = plot8a + ggtitle("Right Direction / Wrong Track")  
  ylab("Proportion")
```

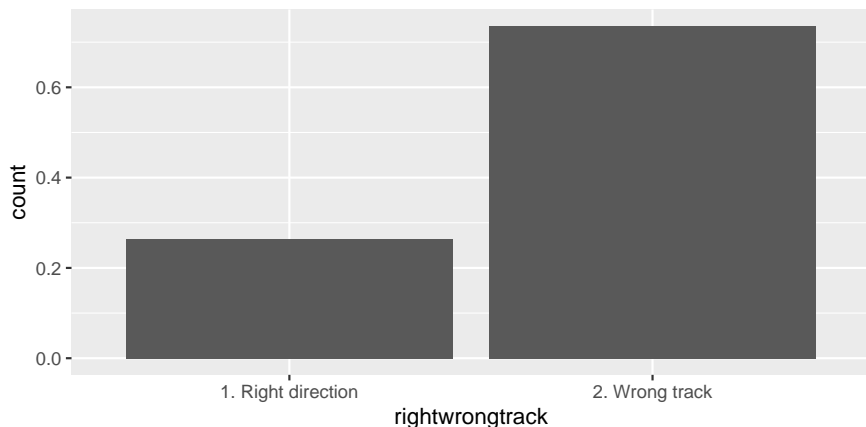
Example 8A: Adding titles to previous plot

```
plot8atitle
```



Example 8B: ggsurvey Bar Chart with Survey Object

```
plot8b = ggbarweight_svy(anes_survey,  
  rightwrongtrack)  
plot8b
```

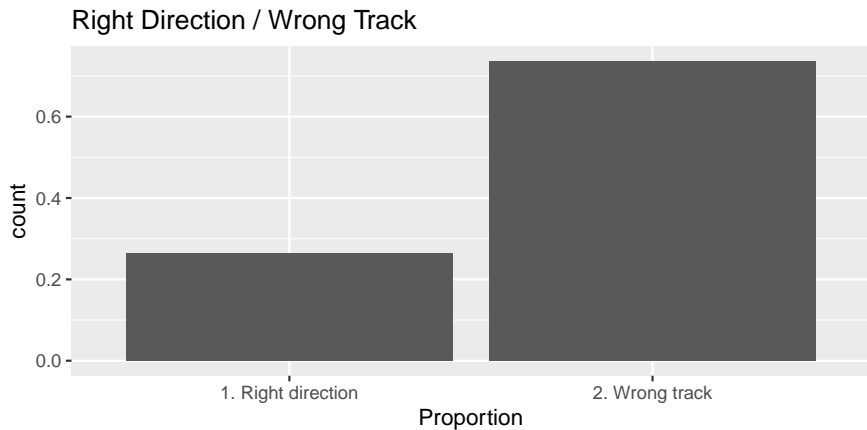


Example 8B: Adding titles to previous plot

```
plot8title = plot8b + ggtitle("Right Direction / Wrong Track")  
  xlab("Proportion")
```


Example 8B: Adding titles to previous plot

```
plot8title
```



Filtering: Advanced

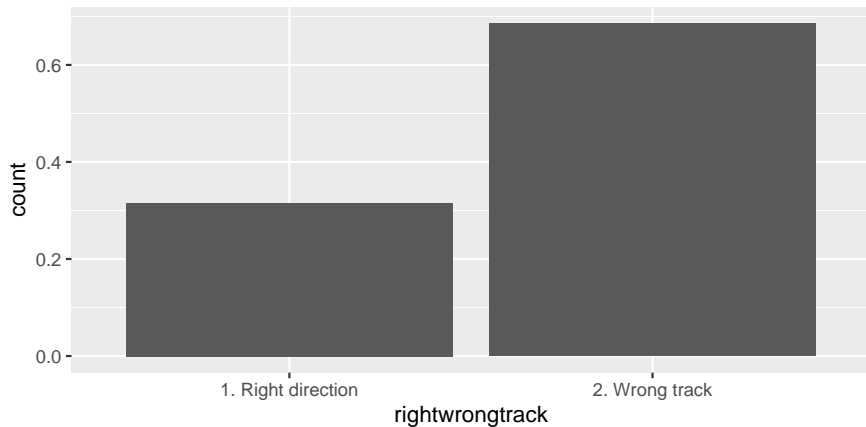
Sometimes we don't want to include a certain category in the plot. For instance we may want to remove the refusals from the graph. We can use the filter function from dplyr on a data frame to remove the refusals from the plot, and then pipe operator. The pipe operator passes the object on the left of the pipe into the function on the right as the first argument. At this time this only works with data frames with survey weights.

Removing Refusals and Don't Know

```
filterplot = timeseries_2020 %>%  
  filter(race == "1. White, non-Hispanic") %>%  
  ggbarweight(rightwrongtrack,  
    V200010a)
```

Removing Refusals and Don't Know

filterplot



Example 9A: Bar Chart with One Crosstab For Dataframe

Sometimes the text overlaps in ggplots.

```
ggbarcrosstabs(timeseries_2020,  
  rightwrongtrack, gender, V200010a)
```



Example 9A: Text Wrap Labels

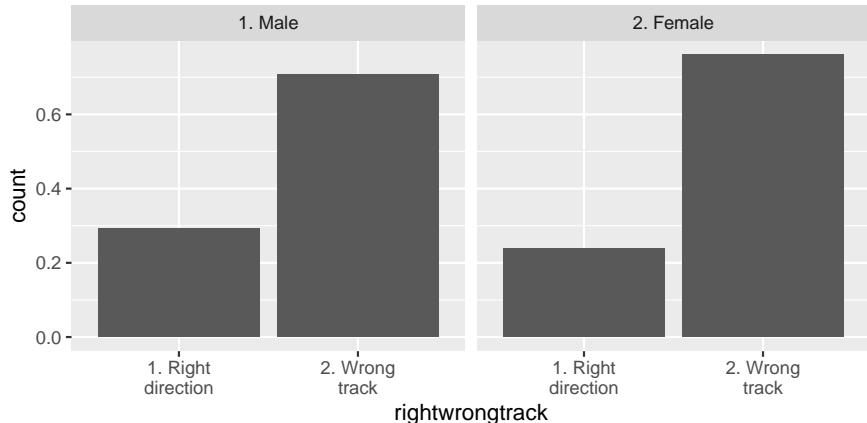
We can use the stringr package to use text wrapping for the labels on the bars.

```
library(stringr)
plot9Atext = ggbarcrossstabs(timeseries_2020,
  rightwrongtrack, gender, V200010a) +
  scale_x_discrete(labels = function(x) str_wrap(x,
    width = 10))
```

Example 9A: Text Wrap Labels

We can use the stringr package to use text wrapping for the labels on the bars.

```
library(stringr)
plot9Atext
```



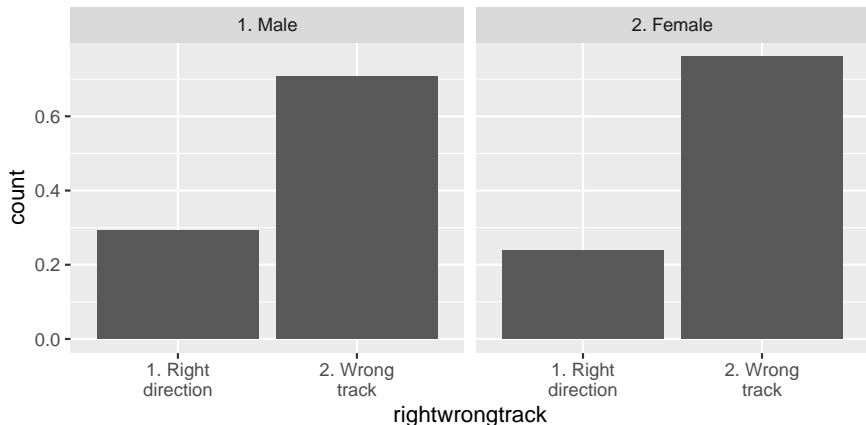
Example 9B: Bar Chart with One Crosstab For svydesign

```
ggbarcrosstabs_svy(anes_survey,  
  rightwrongtrack, gender)
```



Example 9B: Text Wrap Labels

```
ggbarcrossstabs_svy(anes_survey,  
  rightwrongtrack, gender) +  
  scale_x_discrete(labels = function(x) str_wrap(x,  
    width = 10))
```



Example 10A: Bar Chart with Two Crosstab For Dataframe

```
plot10A = ggbarcrosstabs3d(timeseries_2020,  
  rightwrongtrack, economy, gender,  
  V200010a)
```

Example 10A: Bar Chart with Two Crosstab For Dataframe

plot10A

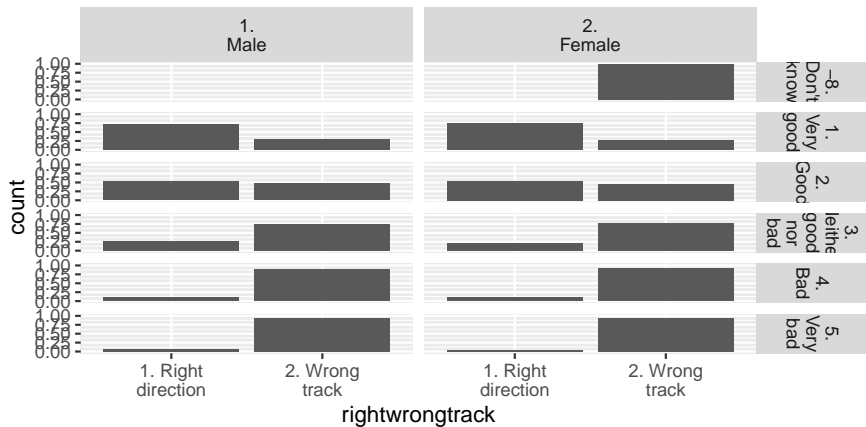


Example 10A: Add Text Wrap Labels

```
plot10Atext = ggbarcrossstabs3d(timeseries_2020,  
  rightwrongtrack, economy, gender,  
  V200010a, labeller = label_wrap_gen(width = 3,  
    multi_line = TRUE)) + scale_x_discrete(labels = function(x)  
  width = 10))
```

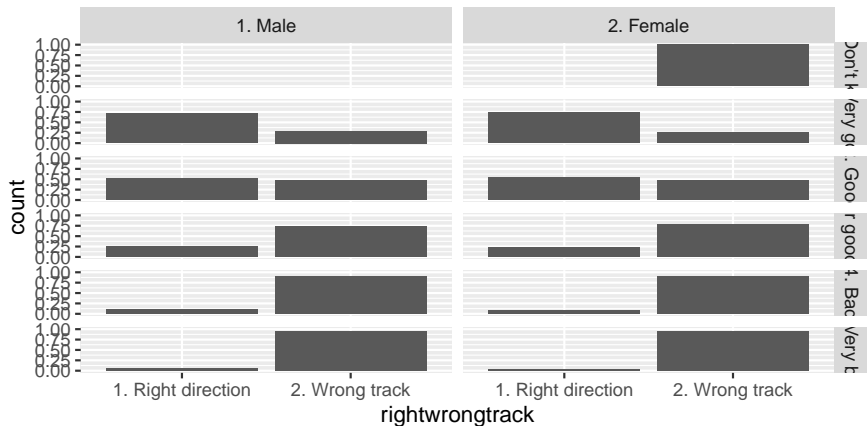
Example 10A: Add Text Wrap Labels

plot10Atext



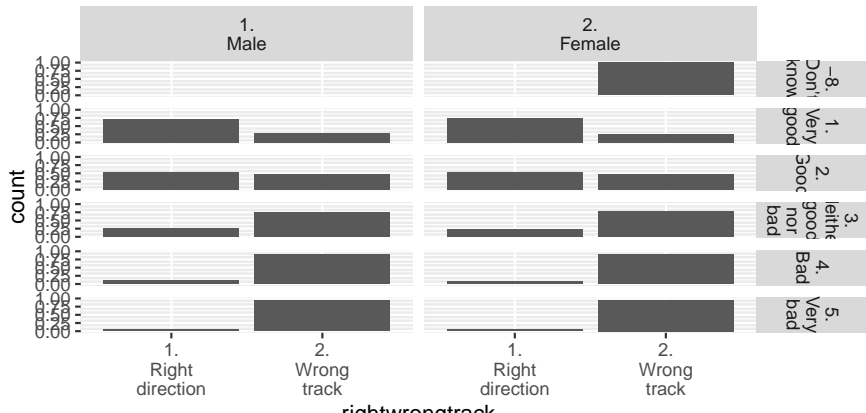
Example 10B: Bar chart with Two Crosstabs For svydesign

```
ggbarcrosstabs3d_svy(anes_survey,  
  rightwrongtrack, economy, gender)
```



Example 10B: Text Wrap Labels

```
ggbarcrossstabs3d_svy(anes_survey,
  rightwrongtrack, economy, gender,
  labeller = label_wrap_gen(width = 3,
    multi_line = TRUE)) + scale_x_discrete(labels = function(x)
    width = 3))
```



Any Questions?

Table of Contents

Course Objectives

Introduction to ggplot2 and ggsurvey

Using ggplot2

Using ggsurvey for bar charts

Using ggsurvey for box plots

Using ggsurvey for histograms

Using ggsurvey for Hexplots

Box Plots

A Boxplot is a way to visualize the distribution of a numeric variable such as a feeling thermometer. A boxplot highlights 5 key statistics: the minimum value, the 25th percentile, the median, the 75th percentile and the maximum value. In ggplot extreme values deemed to be outliers are marked with a dot. We will use the feeling thermometers for Joe Biden (V201151) and Donald Trump (V201152) in the 2020 ANES.

Process Data

We need to first remove the refusals and convert the factor to a numeric in the data frame.

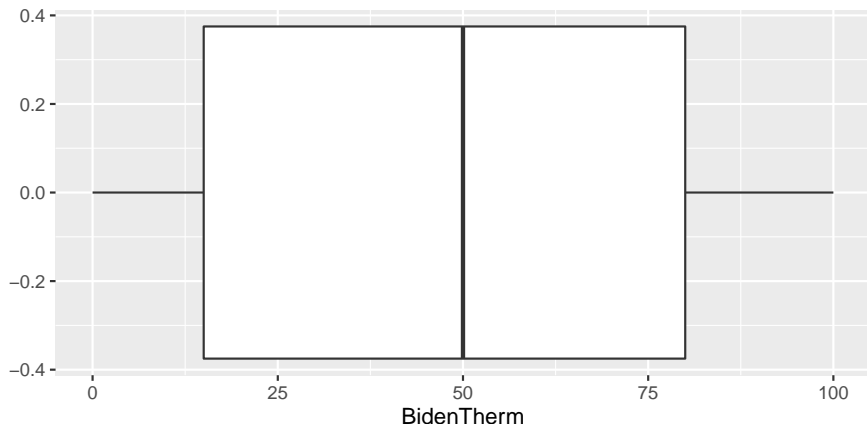
```
timeseries_2020$BidenTherm = as.numeric(as.character(
  timeseries_2020$V201151))
timeseries_2020$TrumpTherm = as.numeric(as.character(
  timeseries_2020$V201152))
anes_svy = anes_svy %>% mutate(BidenTherm =
  as.numeric(as.character(timeseries_2020$V201151)))
anes_svy = anes_svy %>% mutate(TrumpTherm =
  as.numeric(as.character(timeseries_2020$V201152)))
```

ggsurvey boxplot Functions

Plot type	Data type	ggsurvey function
Box Plot	data frame	<code>ggboxweight(df, x, weights)</code>
Box Plot	svydesign	<code>ggboxweight_svy(surveyobj, x)</code>
Box plot with 1 Facet	data frame	<code>ggboxweight2d(df, x, y, weights)</code>
Box plot with 1 Facet	svydesign	<code>ggboxweight2d_svy(surveyobj, x, y)</code>
Box plot with 2 Facets	data frame	<code>ggboxweight3d(df, x, y, z, weights)</code>
Box plot with 2 Facets	svydesign	<code>ggboxweight3d_svy(surveyobj, x, y, z)</code>

Example 11A: Boxplot with data frame object

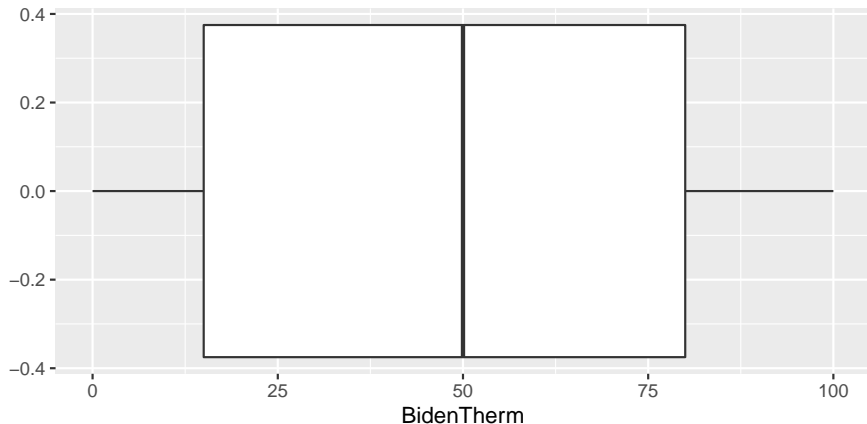
```
ggboxweight(timeseries_2020, BidenTherm,  
            V200010a)
```



Example 11B: Boxplot with survey design object

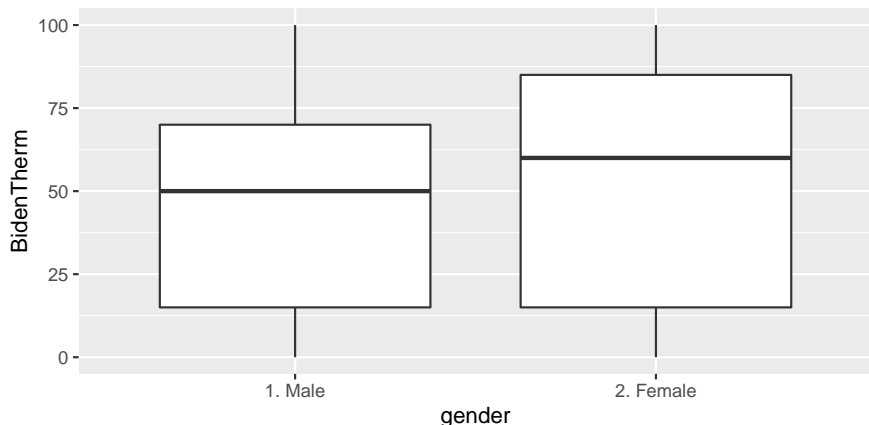
Recall that we stored the survey design object for the ANES in `anes_svy`.

```
ggboxweight_svy(anes_svy, BidenTherm)
```



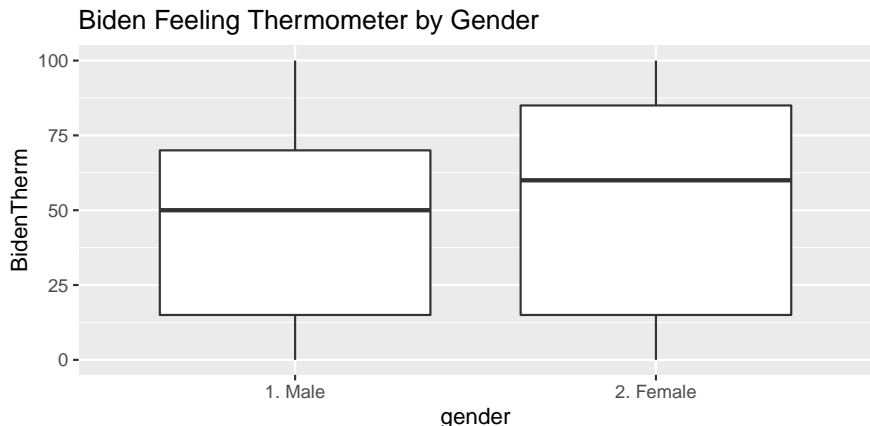
Example 12A: Boxplot with one facet with data frame object

```
ggboxweight2d(timeseries_2020,  
  BidenTherm, gender, V200010a)
```



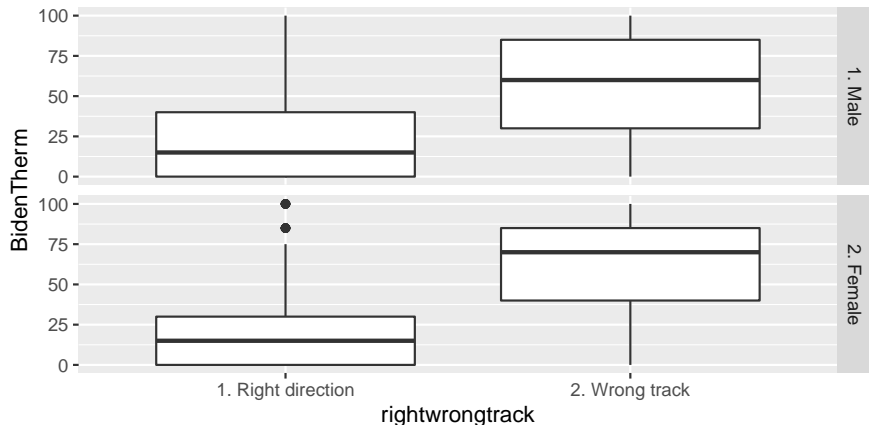
Example 12B: Boxplot with one facet with survey design object

```
ggboxweight2d_svy(anes_svy, BidenTherm,  
  gender) + ggtitle("Biden Feeling Thermometer by Gender")
```



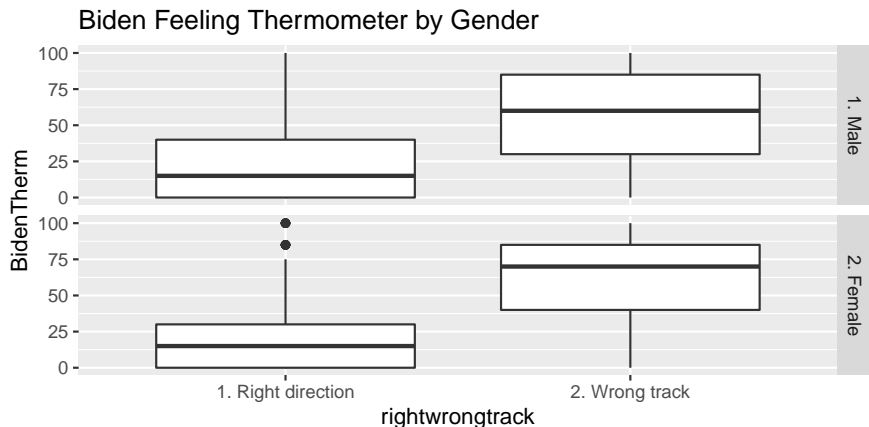
Example 12A: Boxplot with two facets with data frame object

```
ggboxweight3d(timeseries_2020,  
  BidenTherm, rightwrongtrack,  
  gender, V200010a)
```



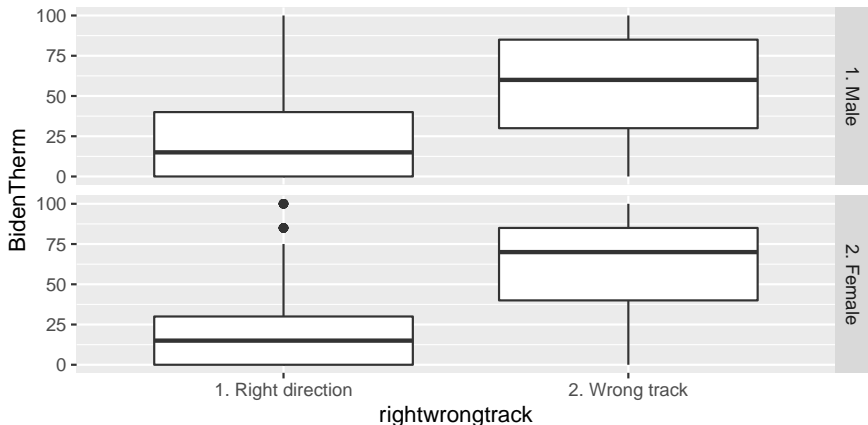
Example 12B: Boxplot with two facets with survey design object

```
ggboxweight3d_svy(anes_svy, BidenTherm,  
  rightwrongtrack, gender) +  
  ggtitle("Biden Feeling Thermometer by Gender")
```



Base ggplot solution for 12A

```
ggplot(timeseries_2020, aes(x = rightwrongtrack,  
  y = BidenTherm)) + geom_boxplot(aes(weight = V200010a)) +  
  facet_grid(rows = vars(gender))
```



Any Questions?

Table of Contents

Course Objectives

Introduction to ggplot2 and ggsurvey

Using ggplot2

Using ggsurvey for bar charts

Using ggsurvey for box plots

Using ggsurvey for histograms

Using ggsurvey for Hexplots

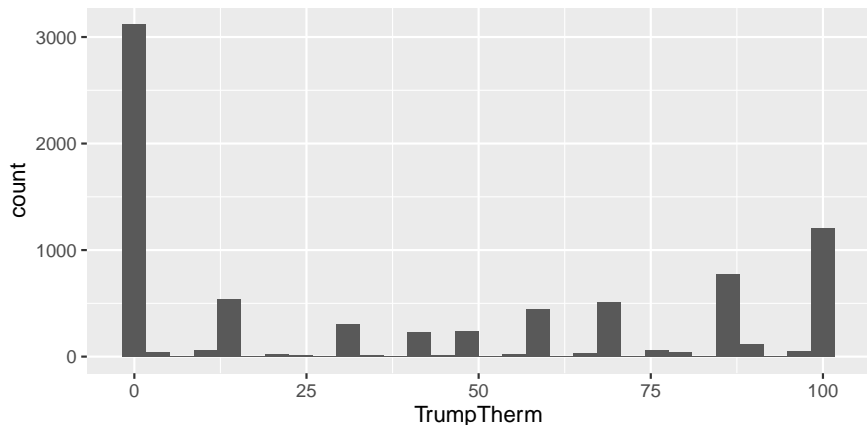
Histograms in ggsurvey

We will work with Trump's feeling thermometer question to make histograms.

Plot type	Data type	ggsurvey function
Histogram	data frame	<code>gghistweight(df,x,weights,binwidth)</code>
Histogram	svydesign	<code>gghistweight_svy(surveyobj,x,binwidth)</code>
Histogram with 1 Facet	data frame	<code>gghistweight2d(df,x,y,weights,binwidth)</code>
Histogram with 1 Facet	svydesign	<code>gghistweight2d_svy(surveyobj,x,y,binwidth)</code>
Histogram with 2 Facets	data frame	<code>gghistweight3d(df,x, y,z,weights,binwidth)</code>
Histogram with 2 Facets	svydesign	<code>gghistweight3d_svy(surveyobj,x,y,z,binwidth)</code>

Example 13A: Histogram with data frame object

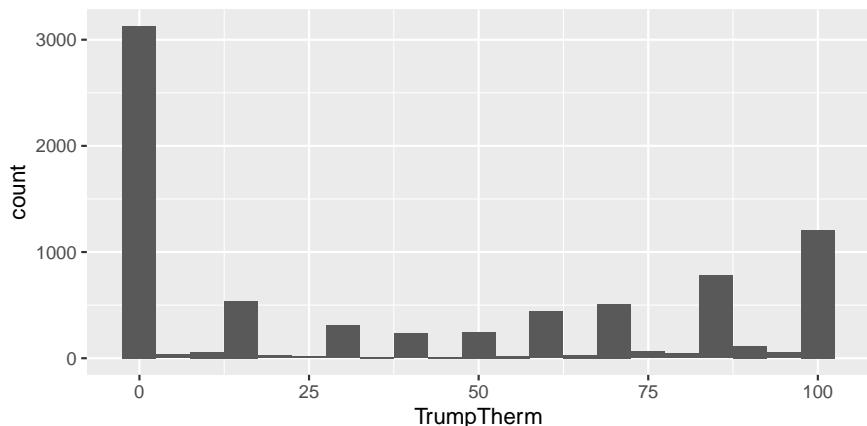
```
gghistweight(timeseries_2020, TrumpTherm,  
             V200010a)
```



Example 13A: Histogram with data frame object

Try a binwidth of 5.

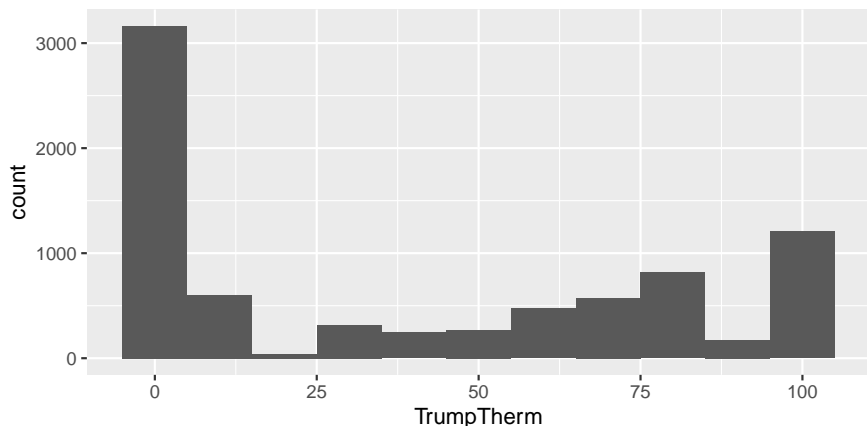
```
gghistweight(timeseries_2020, TrumpTherm,  
             V200010a, binwidth = 5)
```



Example 13A: Histogram with data frame object

Try a binwidth of 10. This is what I prefer, and will use in the next plots.

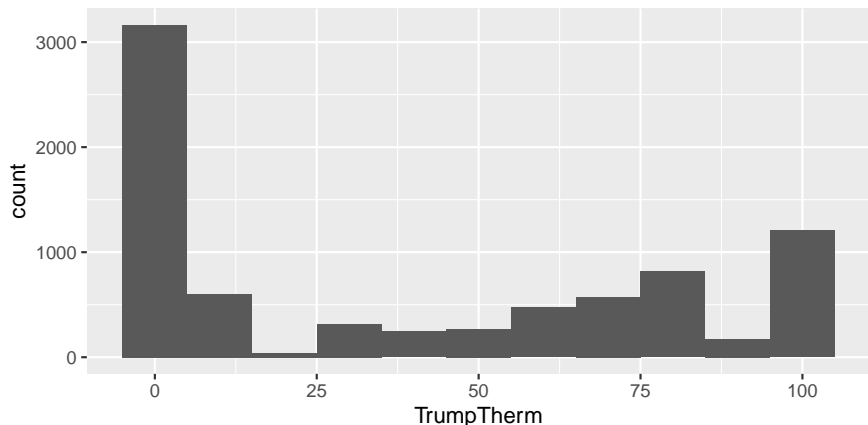
```
gghistweight(timeseries_2020, TrumpTherm,  
             V200010a, binwidth = 10)
```



Example 13B: Histogram with survey design object

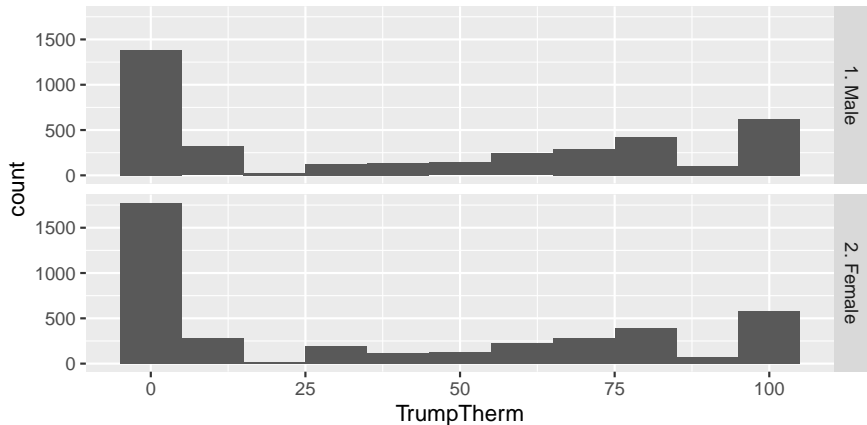
Recall that we stored the survey design object for the ANES in `anes_svy`.

```
gghistweight_svy(anes_svy, TrumpTherm,  
  binwidth = 10)
```



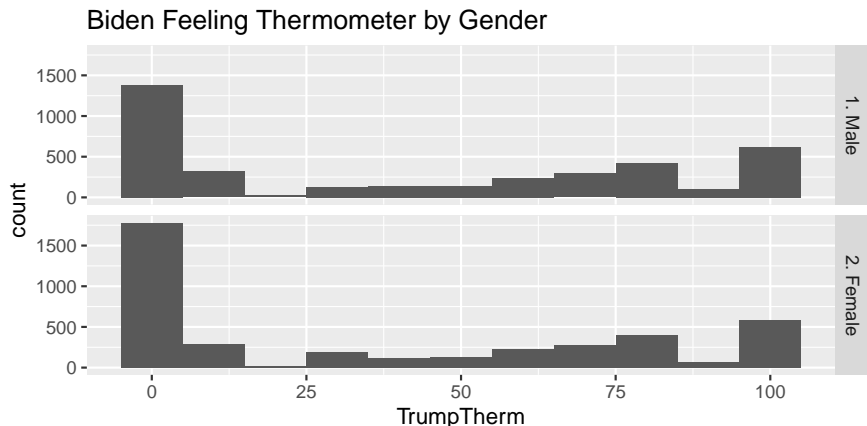
Example 14A: Histogram with one facet with data frame object

```
gghistweight2d(timeseries_2020,  
  TrumpTherm, gender, V200010a,  
  binwidth = 10)
```



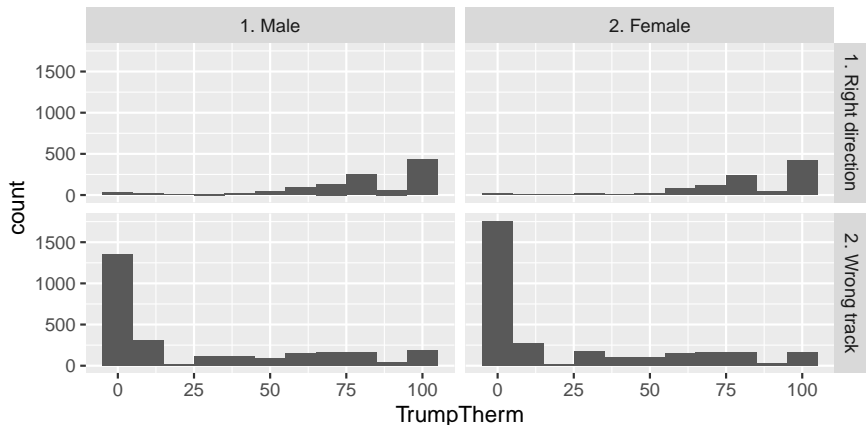
Example 14B: Histogram with one facet with survey design object

```
gghistweight2d_svy(anes_svy, TrumpTherm,  
  gender, binwidth = 10) + ggtitle("Biden Feeling Thermometer
```



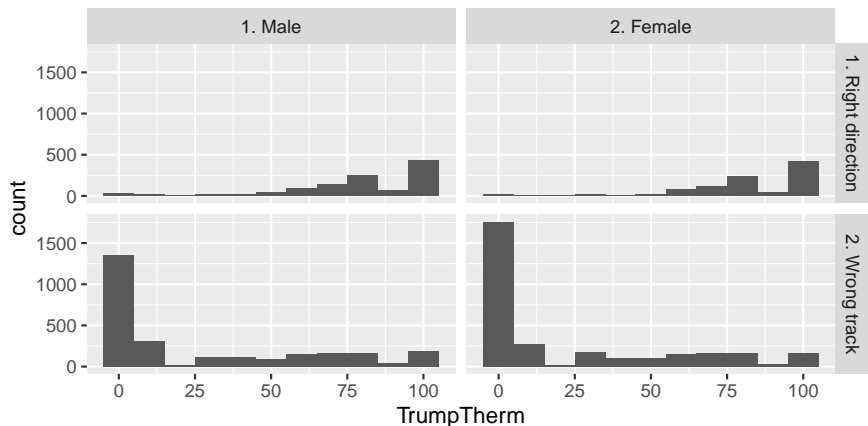
Example 15A: Histogram with two facets with data frame object

```
gghistweight3d(timeseries_2020,  
  TrumpTherm, rightwrongtrack,  
  gender, V200010a, binwidth = 10)
```



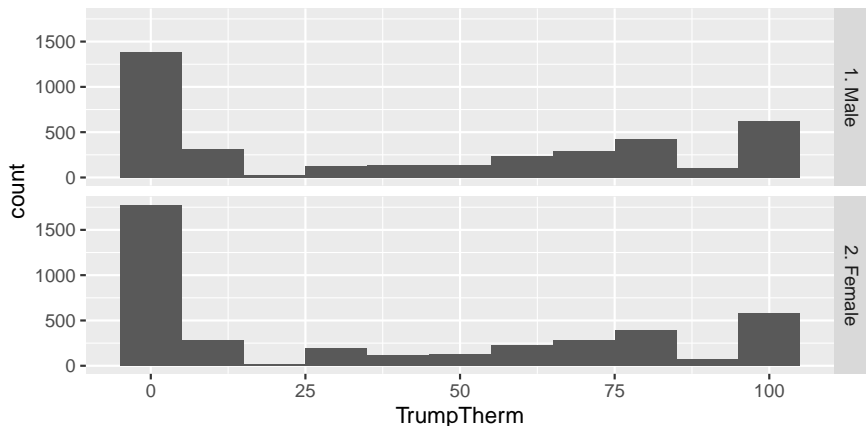
Example 15B: Histogram with two facets with survey design object

```
gghistweight3d_svy(anes_svy, TrumpTherm,  
  rightwrongtrack, gender, binwidth = 10)
```



Base ggplot solution for 15A

```
ggplot(timeseries_2020, aes(TrumpTherm)) +  
  geom_histogram(aes(weight = V200010a),  
    binwidth = 10) + facet_grid(rows = vars(gender))
```



Any Questions?

Table of Contents

Course Objectives

Introduction to ggplot2 and ggsurvey

Using ggplot2

Using ggsurvey for bar charts

Using ggsurvey for box plots

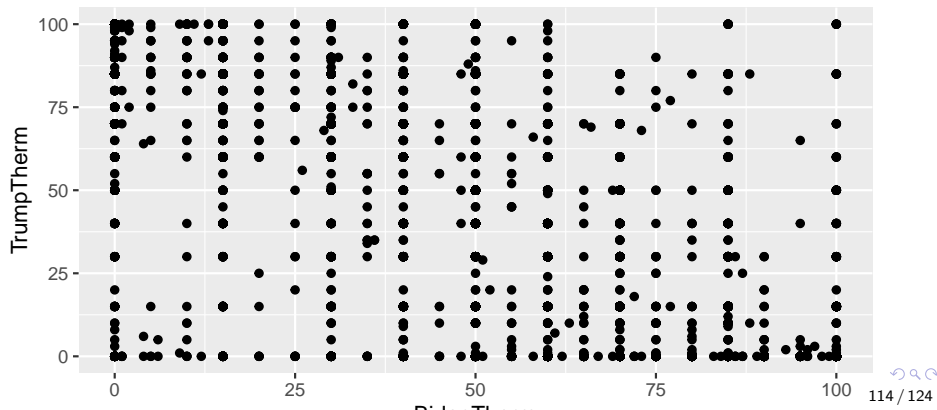
Using ggsurvey for histograms

Using ggsurvey for Hexplots

Motivation for Hexplot

Sometimes you have multiple data points that overlap. When we plot the Trump Feeling Thermometer by the Biden Feeling Thermometer we see less points than number of observations.

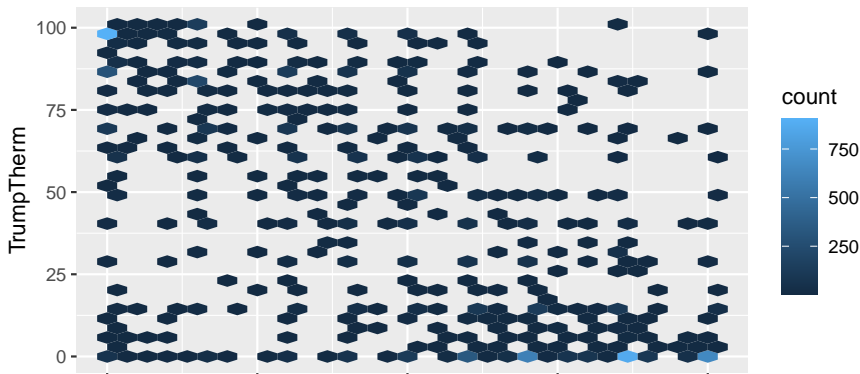
```
ggplot(timeseries_2020, aes(x = BidenTherm,  
y = TrumpTherm)) + geom_point()
```



Example 16A: Hexplot

A hexplot creates bins of data that is close to overlapping and the intensity of shade in the bin denotes how many data points are in each bin. Ignoring survey weights this is done using the hexbin package.

```
library(hexbin)
ggplot(timeseries_2020, aes(x = BidenTherm,
  y = TrumpTherm)) + geom_hex()
```



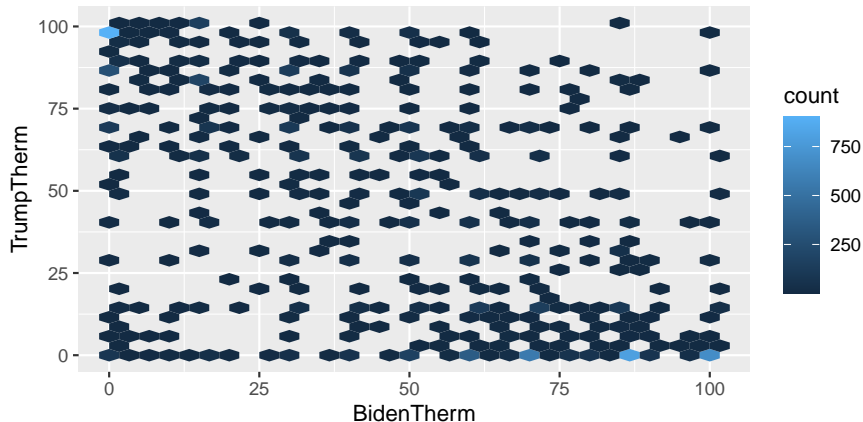
Hexplot ggsurvey functions

Plot type	Data type	ggsurvey function
Hexplot	data frame	<code>gghexweight(df, x, y, weights)</code>
Hexplot	svydesign	<code>gghexweight_svy(surveyobj, x, y)</code>
Hexplot with 1 Facet	data frame	<code>gghexweight2d(df, x, y, z, weights)</code>
Hexplot with 1 Facet	svydesign	<code>gghexweight2d_svy(surveyobj, x, y, z)</code>
Hexplot with 2 Facet	data frame	<code>gghexweight3d(df, x, y, a, b weights)</code>
Hexplot with 2 Facet	svydesign	<code>gghexweight3d_svy(surveyobj, x,y, a, b)</code>

Example 16B: Weighted Hexplot With Dataframe

In example 16 we did not account for the survey weights, but `gghexweight` does.

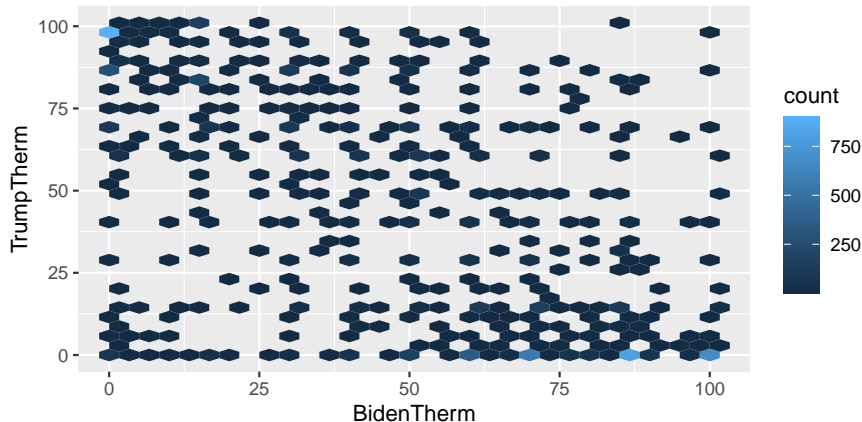
```
gghexweight(timeseries_2020, BidenTherm,  
            TrumpTherm, weight = V200010a)
```



Example 16C: Using a Survey object

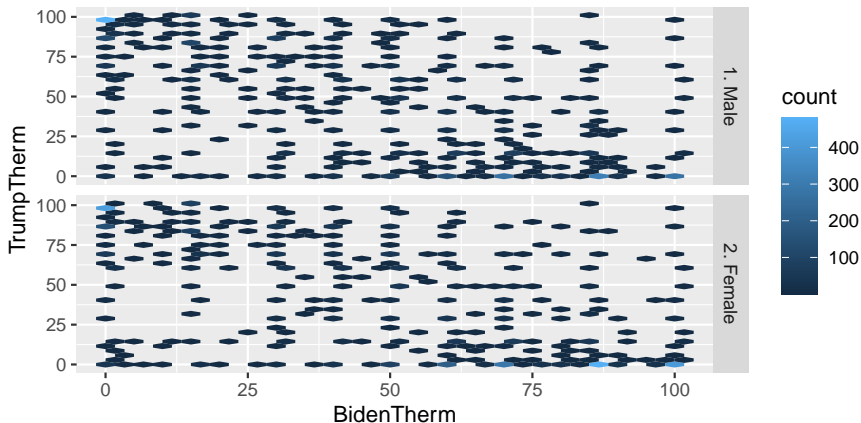
In example 16 we did not account for the survey weights, but `gghexweight` does.

```
gghexweight_svy(anes_svy, BidenTherm,  
  TrumpTherm)
```



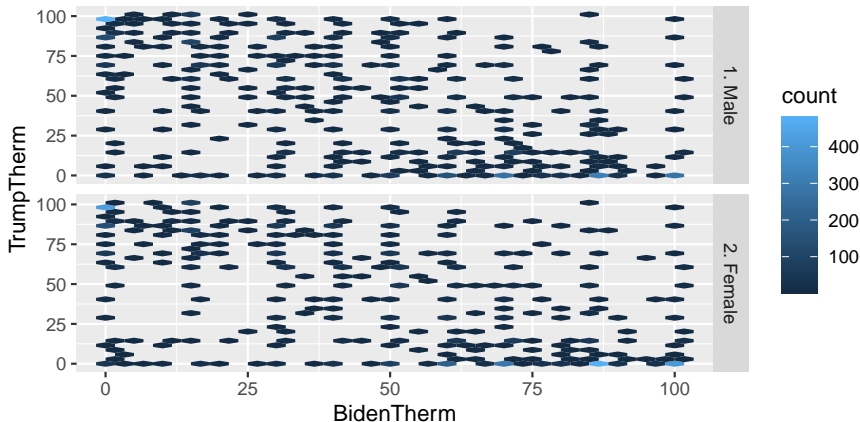
Example 17A: Weighted Hexplot with Facet For Data Frame

```
gghexweight2d(timeseries_2020,  
  BidenTherm, TrumpTherm, gender,  
  weight = V200010a)
```



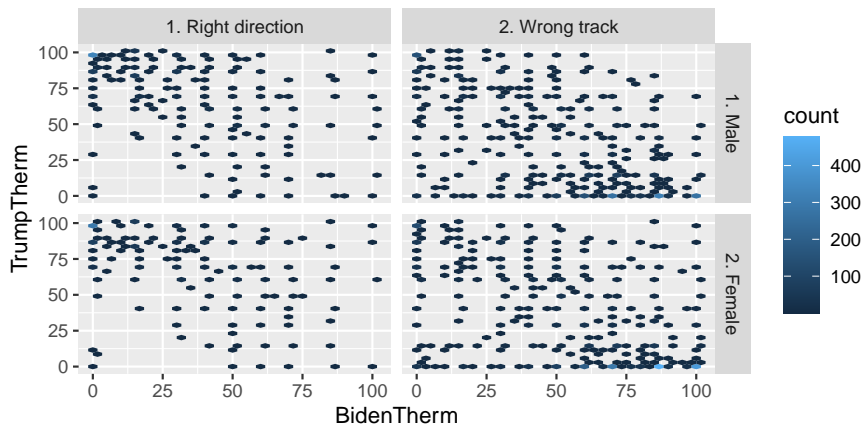
Example 17B: Weighted Hexplot with Facet For svydesign

```
gghexweight2d_svy(anes_svy, BidenTherm,  
  TrumpTherm, gender)
```



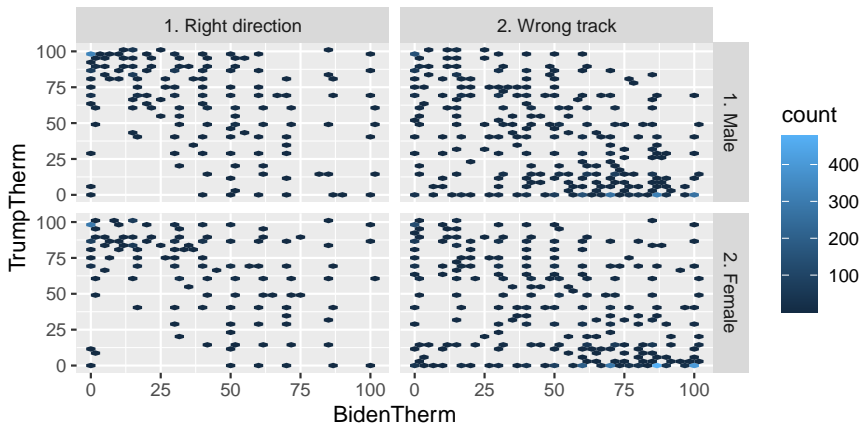
Example 18A: Weighted Hexplot with facet

```
gghexweight3d(timeseries_2020,  
  BidenTherm, TrumpTherm, gender,  
  rightwrongtrack, weight = V200010a)
```



Example 18B: Weighted Hexplot with facet

```
gghexweight3d_svy(anes_svy, BidenTherm,  
  TrumpTherm, gender, rightwrongtrack)
```



Conclusion

- ▶ ggsurvey is still in early development, and new features should be coming, including more customization for histograms
- ▶ Full compatibility with srvyr is in progress.
- ▶ In general ggsurvey is compatible with most "add-on" functions from ggplot2 and related packages (change color scheme, scales, titles), except geoms and facet_grid.
- ▶ ggplot2 and ggsurvey provide an open source way to create high quality visualizations of complex survey data
- ▶ [package website to download on your machine and follow updates](#)

References

- ▶ The [American National Election Studies](#) . These materials are based on work supported by the National Science Foundation under grant numbers SES 1444721, 2014-2017, the University of Michigan, and Stanford University.
- ▶ Horst AM, Hill AP, Gorman KB (2020). [palmerpenguins](#): Palmer Archipelago (Antarctica) penguin data. R package version 0.1.0. doi: 10.5281/zenodo.3960218.
- ▶ Hadley Wickham (2010) A Layered Grammar of Graphics, Journal of Computational and Graphical Statistics, 19:1, 3-28, DOI: 10.1198/jcgs.2009.07098