



**LEARN MORE 365**



# **GGPLOT - Data Visualization Lesson**



# DataVis Cheat Sheet

## Hello

If you're interested in data visualization, then working through this cheat sheet is a good place to start. You'll find examples and code that you can practice at home. The data that I use is available to you on your computer right now. If you want to see a comprehensive list of practice data sets on your computer, simply type `data()` into R Studio. As you install packages, the list of datasets available to you will increase.

The examples that I provide will 1) walk you through the basics of using ggplot to create a data visualization, 2) help you understand which plots to use, given different combinations of data that you might want to look at, and 3) provide some code for beautiful examples of data visualization that you might want to use in your own work.

This sheet won't cover everything. For a more comprehensive overview of data visualization using R, please visit [www.learnmore365.com](http://www.learnmore365.com)

## Load packages

If you haven't ever installed these packages, then you need to do so. You only ever have to install a package on your computer once using the function `install.packages("package_name")`

Here are all of the packages used to develop the graphics in this cheat sheet. I've also included a line of code that sets the theme for all plots to "black and white"

```
library(tidyverse)
library(ggribes)
library(patchwork)
library(viridis)
library(hrbrthemes)
library(gapminder)
theme_set(theme_bw())
```

## Using ggplot2

To create graphics using `ggplot2`, you need to understand "the grammar of graphics". All plots have three principle components:

1. data
2. mapping
3. geometry

### Data

The data is simply the dataset that you are using (nothing complicated about that).

## Mapping

Mapping refers to how each variable that you are going to use in your visualization relates to a particular aesthetic. This could be color, shape, size, x-axis, y-axis and others.

## Geometry

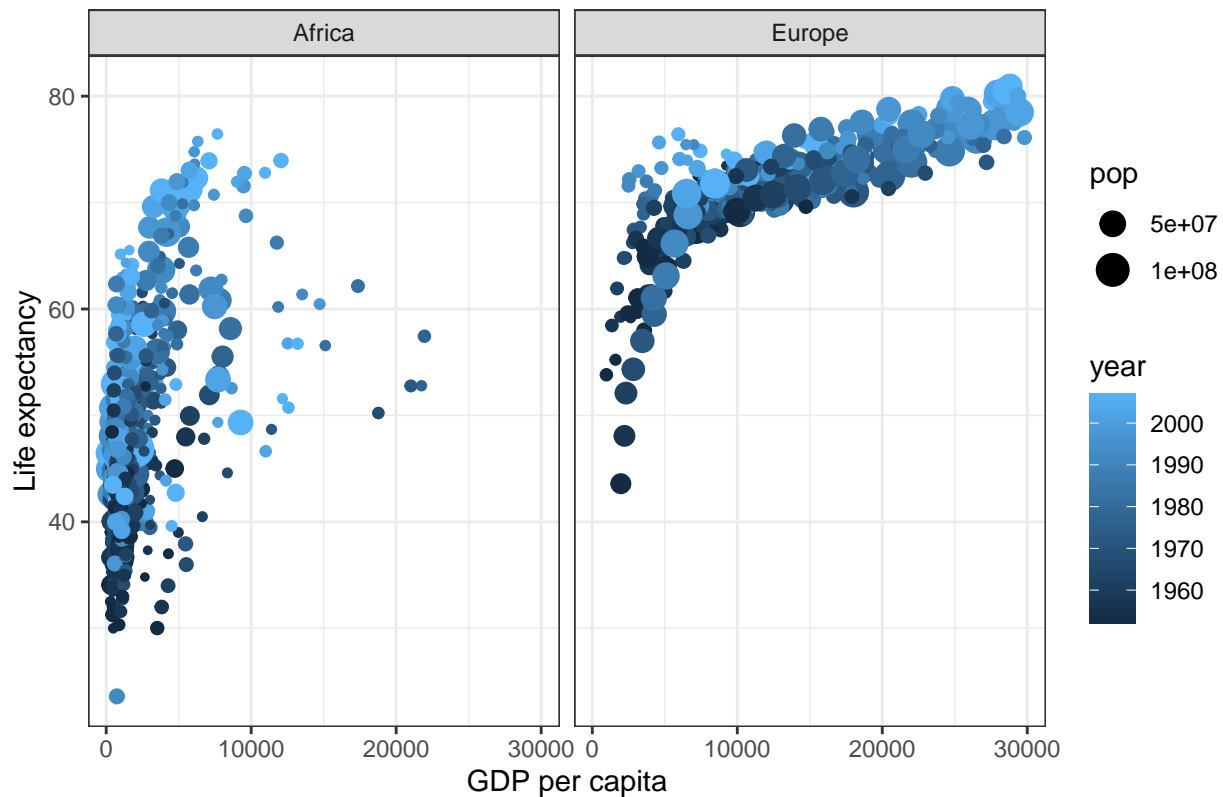
Geometry refers to the type of plot that will be used to represent the dataset. For example, you might want a boxplot, a histogram, a scatter plot, etc.

In the example below, I use the `gapminder` dataset (which is available to you once you've installed the `gapminder` package) to represent 5 variables at the same time. Four numeric variables (GDP per capita, Life expectancy, year and population size) and one categorical variable (continent).

Each of the numeric variables is mapped against a specific aesthetic and the categorical variable is used to disaggregate the data into facets.

```
gapminder %>%
  filter(continent %in% c("Africa", "Europe")) %>%
  filter(gdpPercap < 30000) %>%
  ggplot(aes(x= gdpPercap,
             y = lifeExp,
             size = pop,
             color = year)) +
  geom_point() +
  facet_wrap(~continent) +
  labs(title = "Life expectancy explained by GDP per capita",
       x = "GDP per capita",
       y = "Life expectancy")
```

## Life expectancy explained by GDP per capita



## How to visualize different combinations of data types

Here we are going to look at the `starwars` dataset. I'm going to provide you with examples of data visualization using different combinations of numeric and categorical variables. In this subset of the `starwars` data, I've included `name` as a unique identifier for each row (we won't be using the `name` variable in the analysis). `height` and `mass` are two numeric variables. `gender` and `hair_color` are categorical variables.

```
starwars %>%
  select(name, height, mass, gender, hair_color) %>%
  head()
```

```
## # A tibble: 6 x 5
##   name      height  mass gender  hair_color
##   <chr>      <int> <dbl> <chr>   <chr>
## 1 Luke Skywalker    172    77 masculine blond
## 2 C-3PO             167    75 masculine <NA>
## 3 R2-D2              96    32 masculine <NA>
## 4 Darth Vader       202   136 masculine none
## 5 Leia Organa       150    49 feminine brown
## 6 Owen Lars         178   120 masculine brown, grey
```

## Single numeric variable

Let's start with a single numeric variable (`height`). In this figure we've created a histogram, a density plot, a boxplot and a violin plot with the data. Here is the code and the outputs.

```
p1 <- starwars %>%
  ggplot(aes(x = height)) +
  geom_histogram(binwidth = 20,
                 show.legend = F,
                 alpha = .5) +
  labs(title = "Histogram",
       x = "Height",
       y = "Count")

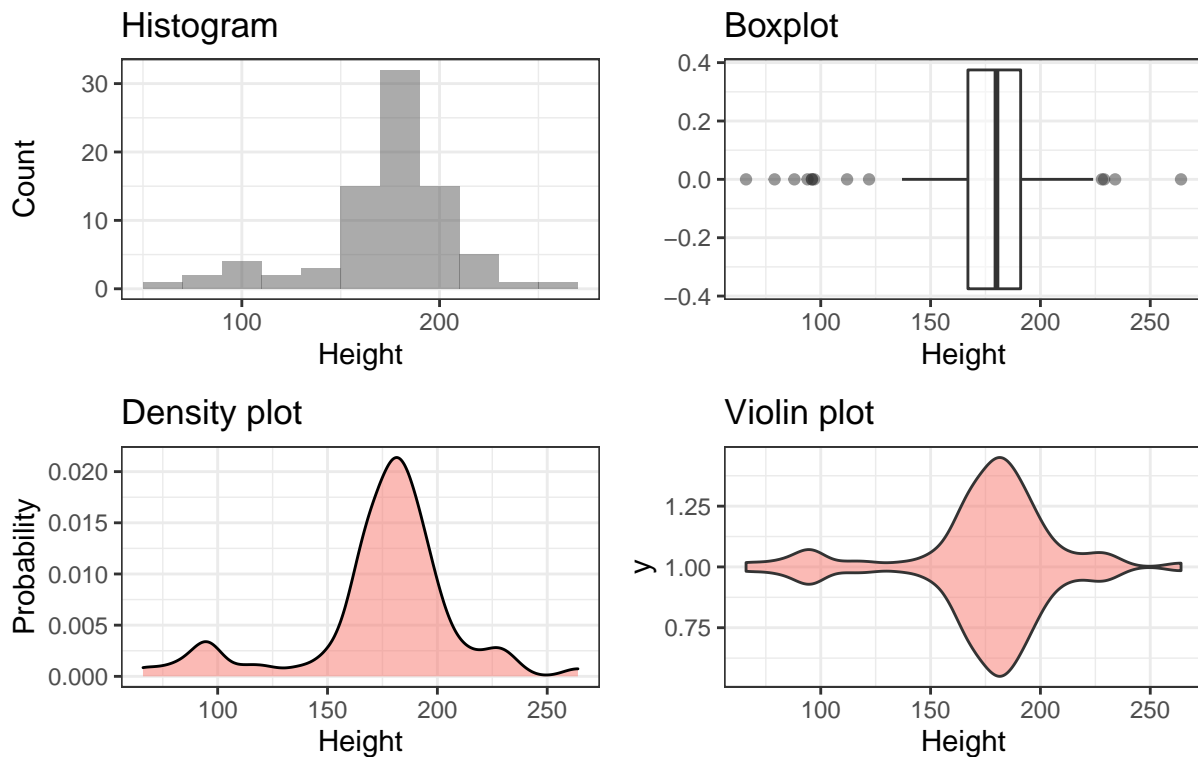
p2 <- starwars %>%
  ggplot(aes(x = height)) +
  geom_density(aes(fill = "blue"),
               show.legend = F,
               alpha = .5) +
  labs(title = "Density plot",
       x = "Height",
       y = "Probability")

p3 <- starwars %>%
  ggplot(aes(x = height)) +
  geom_boxplot(show.legend = F,
               alpha = .5) +
  labs(title = "Boxplot",
       x = "Height")

p3a <- starwars %>%
  ggplot(aes(x = height, y = 1)) +
  geom_violin(aes(fill = "blue"),
              show.legend = F,
              alpha = .5) +
  labs(title = "Violin plot",
       x = "Height")

(p1 / p2 | p3 / p3a) +
  plot_annotation(title = "Single numeric variable",
                  theme = theme(plot.title = element_text(size = 18,
                                                            colour = "blue"))) +
  theme(text = element_text('mono'))
```

## Single numeric variable



## One or more categorical variables

In this graphic I've included a bar plot for a single categorical variable. If a second categorical variable is added we can disaggregate each bar by either stacking, grouping or presenting a proportion using the second category.

```
p4 <- starwars %>%
  drop_na(eye_color) %>%
  filter(eye_color %in% c("black", "brown", "blue", "yellow")) %>%
  ggplot(aes(x = eye_color)) +
  geom_bar(stat = "count", alpha = 0.5) +
  labs(title = "Barplot",
       x = "Eye colour",
       y = "Count")

p5 <- starwars %>%
  drop_na(eye_color, gender) %>%
  filter(eye_color %in% c("black", "brown", "blue", "yellow")) %>%
  ggplot(aes(eye_color, fill = gender)) +
  geom_bar(stat = "count", alpha = .5,
          show.legend = F) +
  labs(title = "Stacked barplot",
       x = "Eye colour",
       y = "Count")
```

```

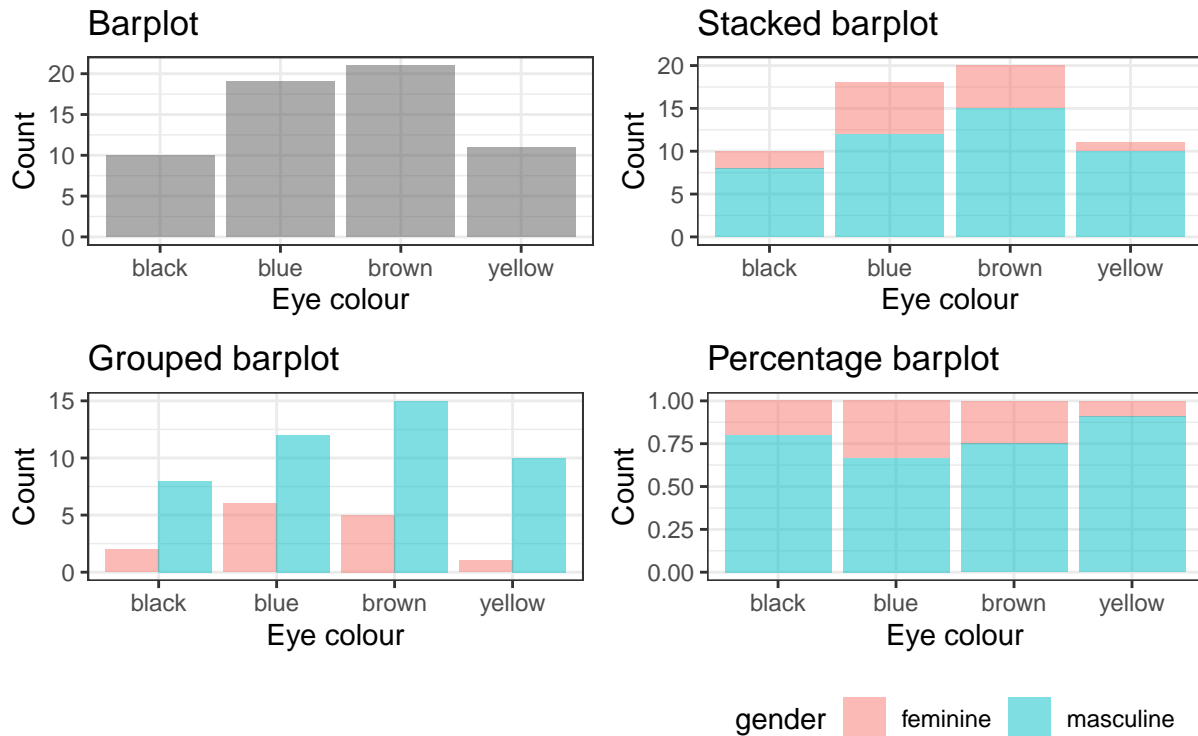
p5a <- starwars %>%
  drop_na(eye_color, gender) %>%
  filter(eye_color %in% c("black", "brown", "blue", "yellow")) %>%
  ggplot(aes(eye_color, fill = gender)) +
  geom_bar(stat = "count", alpha = .5,
           position="dodge",
           show.legend = F)+
  labs(title = "Grouped barplot",
       x = "Eye colour",
       y = "Count")

p5b <- starwars %>%
  drop_na(eye_color, gender) %>%
  filter(eye_color %in% c("black", "brown", "blue", "yellow")) %>%
  ggplot(aes(eye_color, fill = gender)) +
  geom_bar(stat = "count", alpha = .5,
           position="fill",
           show.legend = T) +
  labs(title = "Percentage barplot",
       x = "Eye colour",
       y = "Count") +
  theme(legend.position = "bottom")

((p4 | p5)/( p5a | p5b)) +
  plot_annotation(title = "One or more categorical variable",
                  theme = theme(plot.title = element_text(size = 18,
                                                            colour = "blue")))) +
  theme(text = element_text('mono'))

```

## One or more categorical variable



## One numeric and two categorical variables

Here a single numeric variable can be represented by with a density plot or boxplot and then disaggregated by one categorical variable using color or two categorical variables using color and facets.

```
p14a <- starwars %>%
  drop_na(gender) %>%
  ggplot(aes(height, fill = gender)) +
  geom_boxplot(alpha = 0.3) +
  labs(title = "Boxplot of a numeric variable",
       subtitle = "disaggregated by one categorical variable",
       x = "Height") +
  theme(legend.position = "none")

p14b <- starwars %>%
  drop_na(gender) %>%
  ggplot(aes(height, fill = gender)) +
  geom_density(alpha = 0.3) +
  labs(title = "Density plot of a numeric variable",
       subtitle = "disaggregated by one categorical variable",
       x = "Height",
       y = "Probability") +
  theme(legend.position = "none")
```



```

p15 <- starwars %>%
  drop_na(hair_color, gender) %>%
  filter(hair_color %in% c("black", "brown")) %>%
  ggplot(aes(height, fill = gender)) +
  geom_density(alpha = 0.3) +
  facet_wrap(~hair_color) +
  labs(title = "Density plot of a numeric variable",
        subtitle = "disagregated by two categorical variables",
        x = "Height",
        y = "Probability") +
  theme(legend.position = "none")

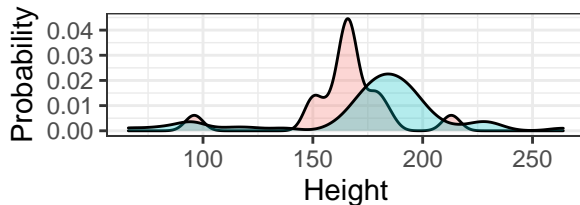
p16 <- starwars %>%
  filter(height > 140 & height < 200) %>%
  drop_na(hair_color, gender) %>%
  filter(hair_color %in% c("black", "brown")) %>%
  ggplot(aes(height, fill = gender)) +
  geom_boxplot(alpha = 0.3) +
  facet_wrap(~hair_color) +
  labs(title = "Boxplot of a numeric variable",
        subtitle = "disagregated by two categorical variable",
        x = "Height")+
  theme(legend.position = "bottom")

((p14b/p14a)|(p15 / p16)) +
  plot_annotation(title = "One numeric and two categorical variable",
                  theme = theme(plot.title = element_text(size = 18,
                                                            colour = "blue")) +
  theme(text = element_text('mono'))

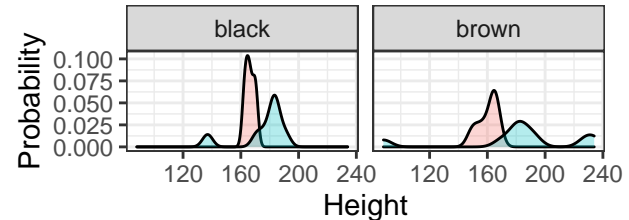
```

## One numeric and two categorical variable

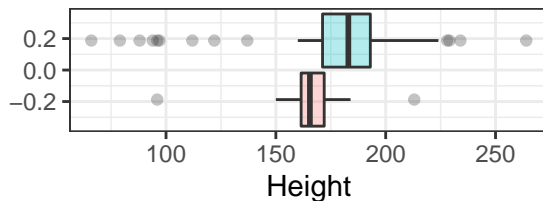
Density plot of a numeric variable  
disaggregated by one categorical variable



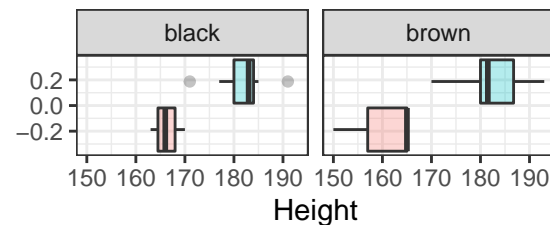
Density plot of a numeric variable  
disaggregated by two categorical variable



Boxplot of a numeric variable  
disaggregated by one categorical variable



Boxplot of a numeric variable  
disaggregated by two categorical variable



gender  feminine  masculine

## Two numeric and one categorical variable

Here we have a scatter plot of two numeric variables on the left. I've included a smoothed linear model with standard error margins.

On the right we have the same data disaggregated by color and then by color and facets.

```
p6 <- starwars %>%
  filter( mass < 250) %>%
  ggplot(aes(x = height,
             y = mass)) +
  geom_point(size = 2,
             alpha = 0.7) +
  geom_smooth()+
  labs(title = "Scatter plot",
       subtitle = "with smoothed linear model",
       x = "Height",
       y = "Mass")

p7 <- starwars %>%
  filter( mass < 250) %>%
  drop_na(gender) %>%
  ggplot(aes(height, mass, colour = gender)) +
  geom_point(size = 2, show.legend = T) +
  labs(title = "Scatter plot",
```

```

    subtitle = "disagregated by colour",
    x = "Height",
    y = "Mass")

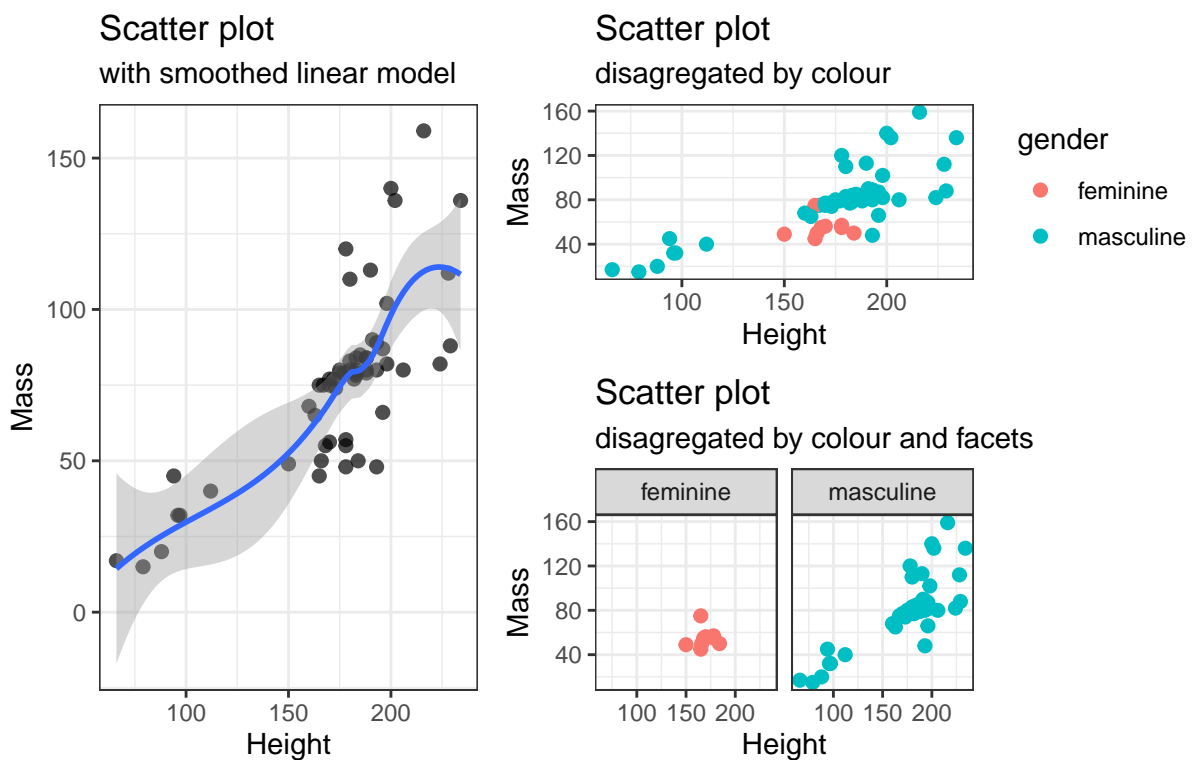
p7a <- starwars %>%
  filter( mass < 250) %>%
  drop_na(gender) %>%
  ggplot(aes(height, mass, colour = gender))+
  geom_point(size = 2, show.legend = F)+
  facet_wrap(~gender) +
  labs(title = "Scatter plot",
       subtitle = "disagregated by colour and facets",
       x = "Height",
       y = "Mass")

(p6 | p7 / p7a) +
  plot_annotation(title = "Two numeric and one categorical variable",
                  theme = theme(plot.title = element_text(size = 18,
                                                            colour = "blue")) +
                  theme(text = element_text('mono')))

## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'

```

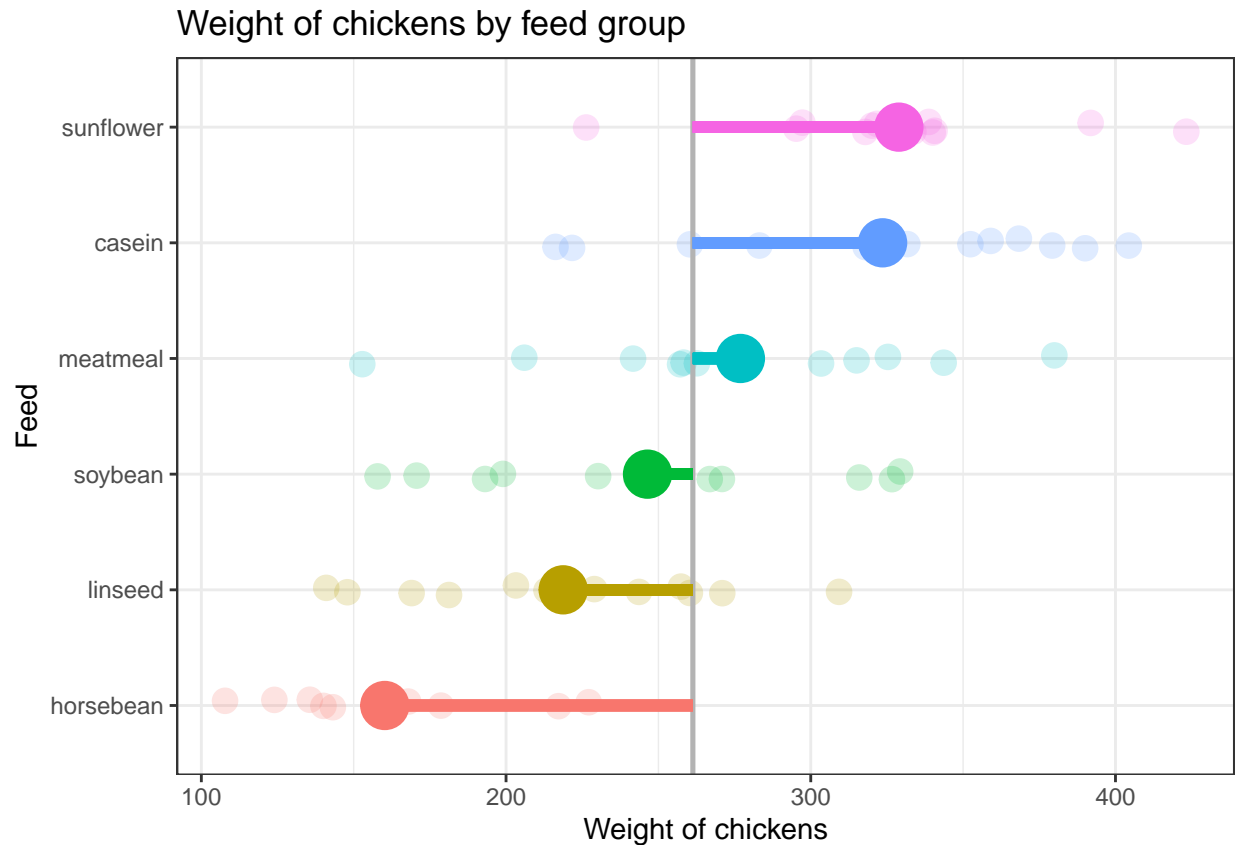
## Two numeric and one categorical variable



## Lolipop graphic

Instead of plotting a simple boxplot for each of the categories, I've plotted individual data points, the average for each category and then joined the average with a line that represents the average for the entire dataset.

```
chickwts %>%
  group_by(feed) %>%
  mutate(mean_by_feed = mean(weight)) %>%
  ungroup() %>%
  mutate(feed = fct_reorder(feed, mean_by_feed)) %>%
  ggplot(aes(feed, weight, colour = feed,
             show.legend = F)) +
  coord_flip() +
  geom_jitter(show.legend = F,
             size = 4,
             alpha = 0.2,
             width = 0.05) +
  stat_summary(fun = mean, geom = "point", size = 8, show.legend = F) +
  geom_hline(aes(yintercept = mean(weight)),
            colour = "gray70",
            size = 0.9) +
  geom_segment(aes(x = feed, xend = feed,
                  y = mean(weight), yend = mean_by_feed),
              size = 2, show.legend = F) +
  labs(title = "Weight of chickens by feed group",
       x = "Feed",
       y = "Weight of chickens") +
  theme(legend.position = "none") +
  theme_bw()
```



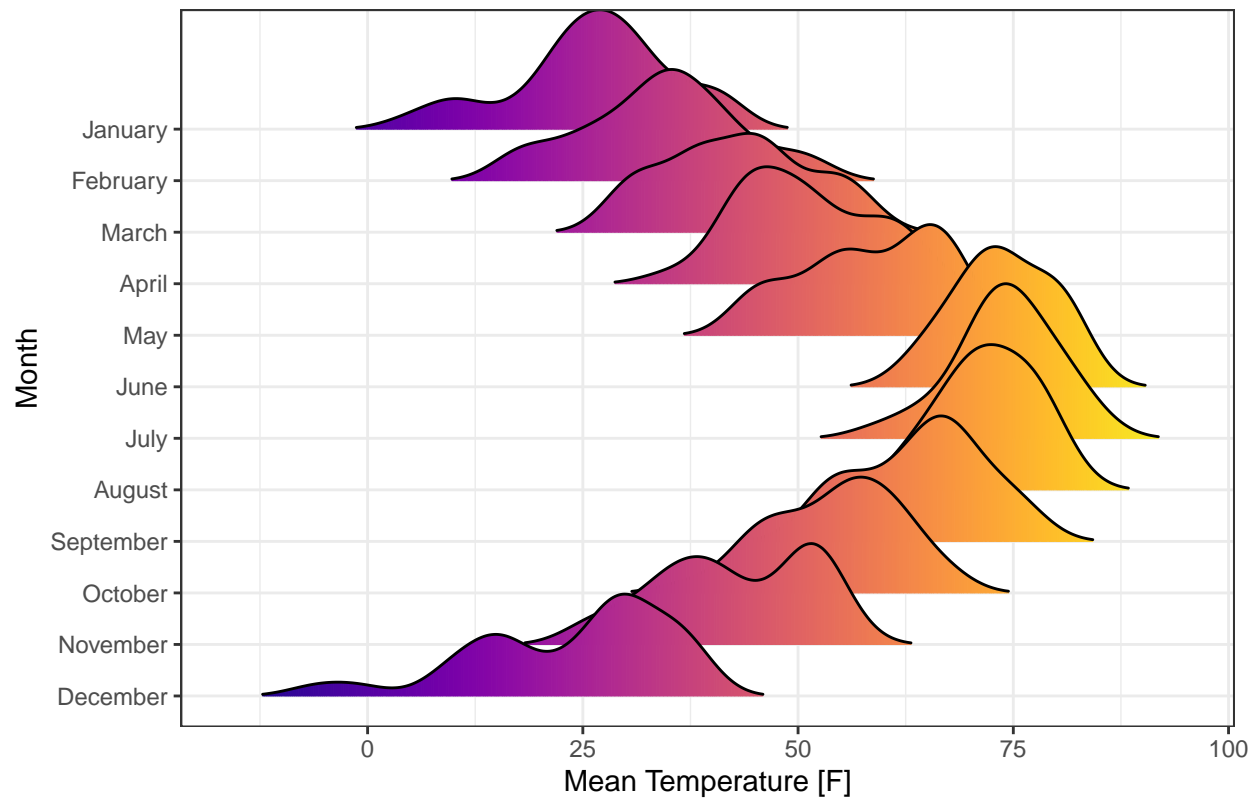
## Using ridges

In this visualization I've shown distribution of temperatures that occurred within a given month (as a density plot) and then compared each month by plotting the density plots of each month on the same canvas.

```
ggplot(lincoln_weather, aes(x = `Mean Temperature [F]`, y = `Month`, fill = ..x..)) +
  geom_density_ridges_gradient(scale = 3, rel_min_height = 0.01,
                              alpha = 5) +
  scale_fill_viridis(name = "Temp. [F]", option = "C") +
  labs(title = 'Temperatures in Lincoln NE in 2016') +
  theme_bw() +
  theme(
    legend.position="none",
    panel.spacing = unit(0.1, "lines"),
    strip.text.x = element_text(size = 8)
  )
```

## Picking joint bandwidth of 3.37

Temperatures in Lincoln NE in 2016



# Learn More 365

*Share this cheat sheet with others*



**Unlock a wealth of FREE resources TODAY!**

Click on the links below to enrich your knowledge in these topics:

 [Statistics & Research Methods Resource Library](#)

 [R Programming & Data Visualization Resource Library](#)

 [Discover the Public Health & Epidemiology Resource Library](#)

Explore video lessons on **Statistics & Research Methods,**  
**R Programming & Data Visualization**  
**Public Health & Epidemiology,** and more  
at **Learn More 365!**

Click [HERE](#) or scan the QR code below to  
start learning!

