Data Visualisation

ggplot2

ggplot2

- Easy to add legends
- Automatic colour scales
- Facets

```
ggplot(top_data, aes(rate, expression, color = nutrient)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
    facet_wrap(~name + systematic_name, scales = "free_y")
```

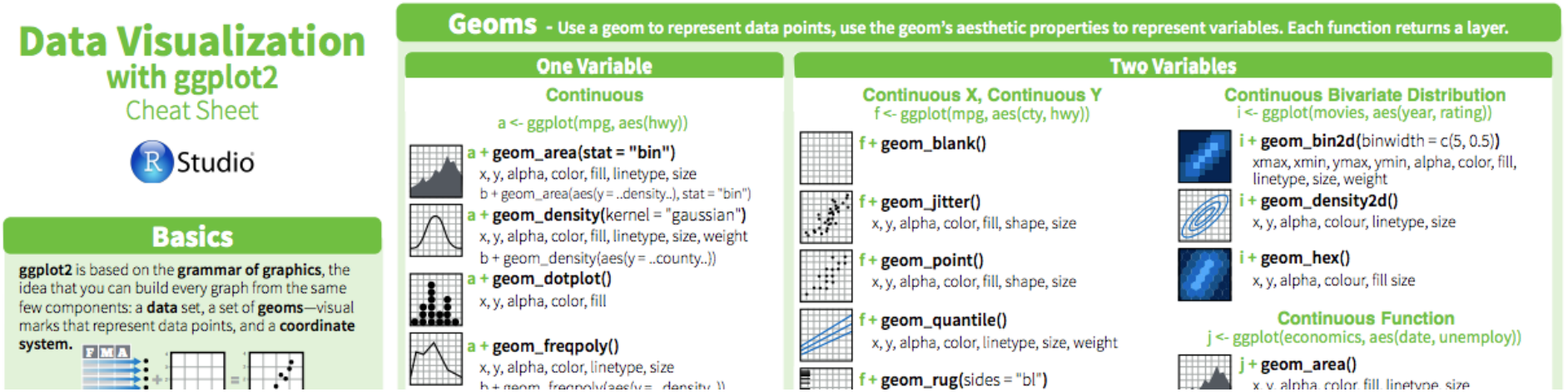
- Grammar of graphics

```
# What goes on x and y axes, and how do we communicate nutrient
# show the user raw observations
# but also show them linear trends
# 20 genes don't fit on 1 plot, show them separately
```

Useful Resources

- R for Data Science - https://r4ds.had.co.nz/

- Rstudio ggplot2 cheatsheet - https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf



Data Visualization with ggplot2 Cheat Sheet

RStudio

Basics

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

Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

One Variable

Continuous

a <- ggplot(mpg, aes(hwy))

a + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")

a + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..county..))

a + geom_dotplot()
x, y, alpha, color, fill

a + geom_freqpoly()
x, y, alpha, color, linetype, size

Two Variables

Continuous X, Continuous Y

f <- ggplot(mpg, aes(cty, hwy))

f + geom_blank()

f + geom_jitter()
x, y, alpha, color, fill, shape, size

f + geom_point()
x, y, alpha, color, fill, shape, size

f + geom_quantile()
x, y, alpha, color, linetype, size, weight

f + geom_rug(sides = "bl")

Continuous Bivariate Distribution

i <- ggplot(movies, aes(year, rating))

i + geom_bin2d(binwidth = c(5, 0.5))
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight

i + geom_density2d()
x, y, alpha, colour, linetype, size

i + geom_hex()
x, y, alpha, colour, fill size

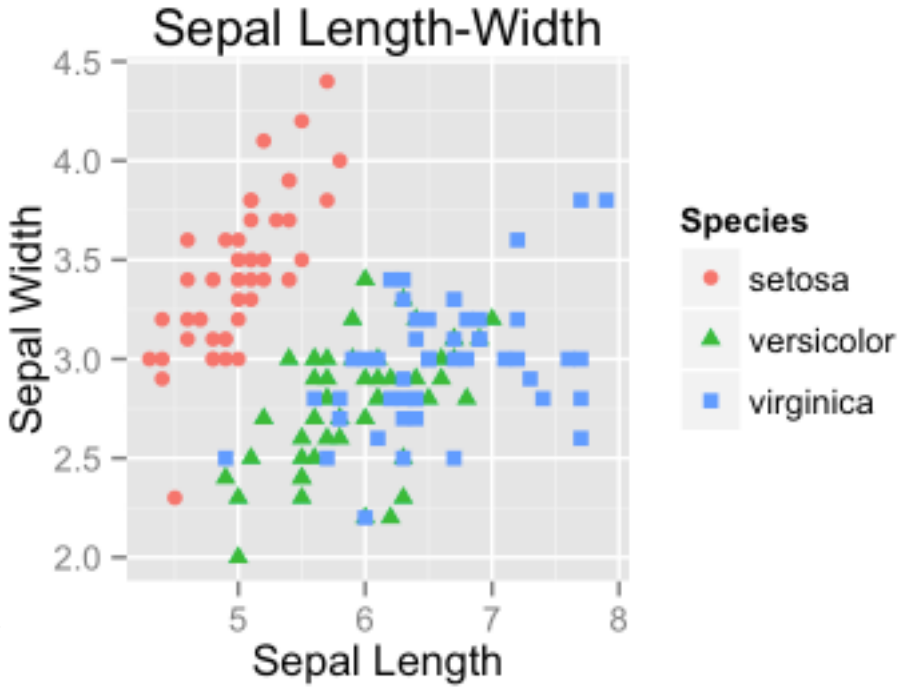
Continuous Function

j <- ggplot(economics, aes(date, unemploy))

j + geom_area()
x, y, alpha, color, fill, linetype, size

ggplot2 example

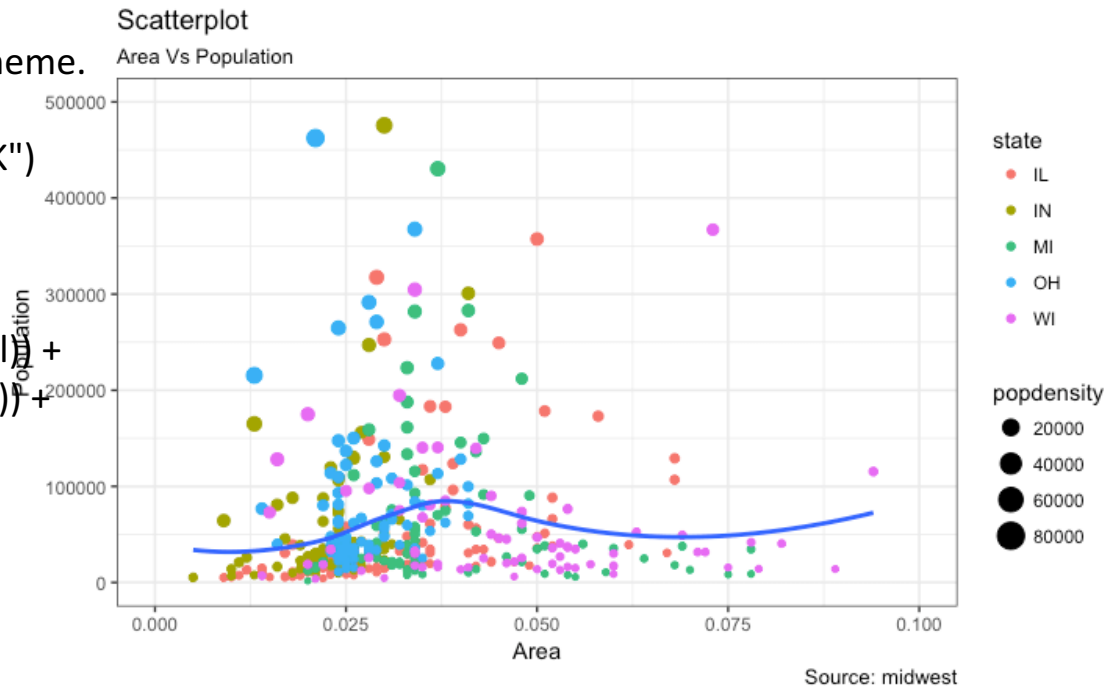
```
library(ggplot2)

ggplot(data=iris,
       aes(x = Sepal.Length,
           y = Sepal.Width)) +
 geom_point(aes(color=Species,
                shape=Species)) +
 xlab("Sepal Length") +
 ylab("Sepal Width") +
 ggtitle("Sepal Length-Width")
```



https://www.mailman.columbia.edu/sites/default/files/media/fdawg_ggplot2.html

Scatter graph example

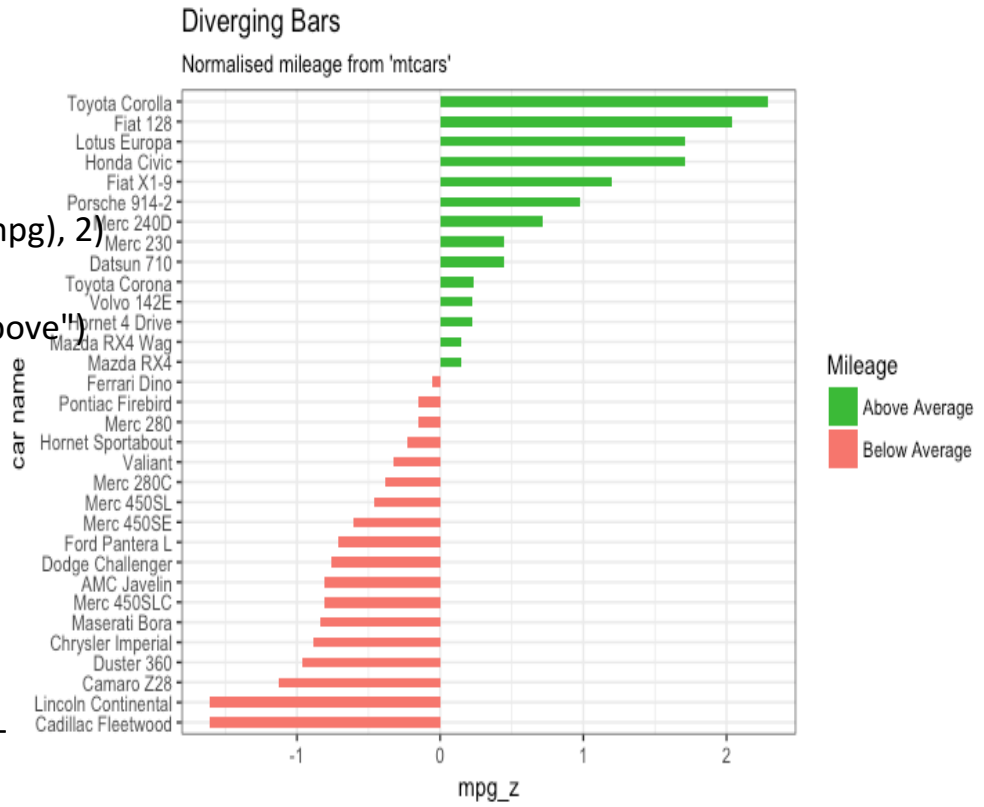
```
# install.packages("ggplot2")  # load package and data
options(scipen=999)  # turn-off scientific notation like 1e+48
library(ggplot2)
theme_set(theme_bw())  # pre-set the bw theme.
data("midwest", package = "ggplot2")
# midwest <- read.csv("http://goo.gl/G1K41K")
# bkup data source

# Scatterplot
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point(aes(col=state, size=popdensity)) +
  geom_smooth(method="loess", se=F) +
  xlim(c(0, 0.1)) +
  ylim(c(0, 500000)) +
  labs(subtitle="Area Vs Population",
       y="Population",
       x="Area",
       title="Scatterplot",
       caption = "Source: midwest")

plot(gg)
```



http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html

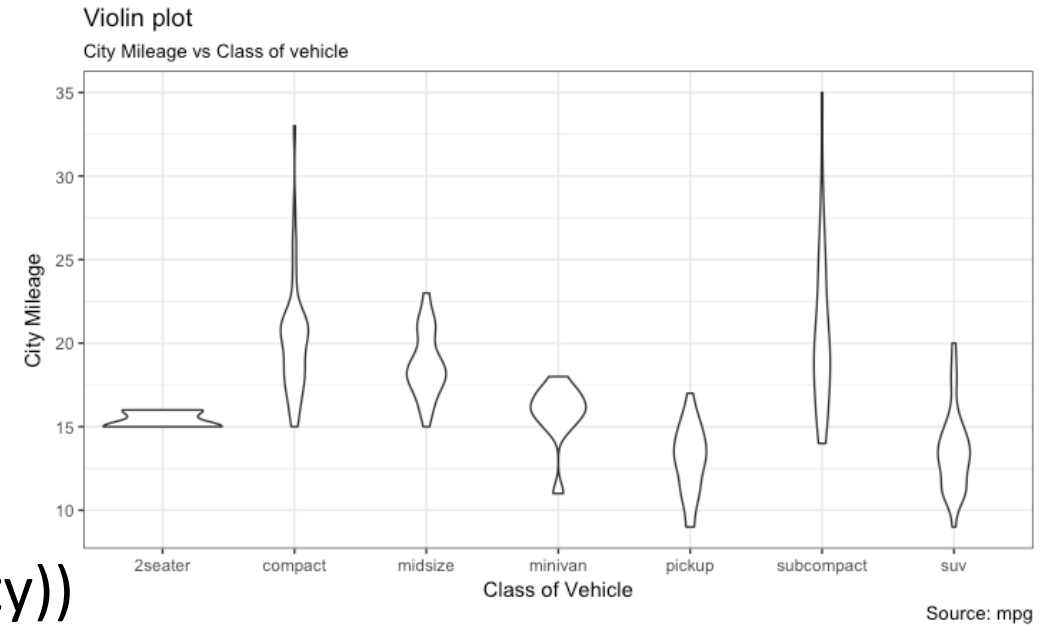
Diverging Bars example

```
library(ggplot2)
theme_set(theme_bw())

# Data Prep
data("mtcars")  # load data
mtcars$`car name` <- rownames(mtcars)
# create new column for car names
mtcars$mpg_z <- round((mtcars$mpg –
                       mean(mtcars$mpg))/sd(mtcars$mpg), 2)
# compute normalized mpg
mtcars$mpg_type <- ifelse(mtcars$mpg_z < 0, "below", "above")
# above / below avg flag
mtcars <- mtcars[order(mtcars$mpg_z), ]  # sort
mtcars$`car name` <- factor(mtcars$`car name`,
                            levels = mtcars$`car name`)
# convert to factor to retain sorted order in plot.


# Diverging Barcharts
ggplot(mtcars, aes(x=`car name`, y=mpg_z, label=mpg_z)) +
  geom_bar(stat='identity', aes(fill=mpg_type), width=.5)  +
  scale_fill_manual(name="Mileage",
                    labels = c("Above Average", "Below Average"),
                    values = c("above"="#00ba38", "below"="#f8766d")) +
  labs(subtitle="Normalised mileage from 'mtcars'",
       title= "Diverging Bars") +
  coord_flip()
```



http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html

Violin Plot example




```
library(ggplot2)
theme_set(theme_bw())
```

```
# plot
g <- ggplot(mpg, aes(class, cty))
g + geom_violin() +
  labs(title="Violin plot",
       subtitle="City Mileage vs Class of vehicle",
       caption="Source: mpg",
       x="Class of Vehicle",
       y="City Mileage")
```

http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html