Introduction to ggplot2

STA3100: Programming with Data

ggplot2

- Is an R package for producing statistical, or data, graphics.
- ggplot2 has an underlying grammar, based on the **G**rammar of **G**raphics, that allows you to compose graphs by combining independent components.
- You can create novel graphics that are tailored to your specific problem.
- Book: *ggplot2: Elegant Graphics for Data Analysis* by Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen (https://ggplot2-book.org/).

Differences with base graphics

- `ggplot()` is the main function used to make any plot.
- The function works with data frames and not individual vectors.
- You can keep enhancing a plot by adding more layers (and themes) to an existing plot created using the ggplot() function.

Example: Fuel economy data

Includes information about the fuel economy of popular car models in 1999 and 2008, collected by the US Environmental Protection Agency.

```r
library(ggplot2)
mpg
```

```
## # A tibble: 234 x 11
##    manufacturer model    displ  year   cyl trans   drv     cty   hwy fl    class
##    <chr>        <chr>    <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
##  1 audi         a4         1.8  1999     4 auto(l~ f        18    29 p     comp~
##  2 audi         a4         1.8  1999     4 manual~ f        21    29 p     comp~
##  3 audi         a4         2    2008     4 manual~ f        20    31 p     comp~
##  4 audi         a4         2    2008     4 auto(a~ f        21    30 p     comp~
##  5 audi         a4         2.8  1999     6 auto(l~ f        16    26 p     comp~
##  6 audi         a4         2.8  1999     6 manual~ f        18    26 p     comp~
##  7 audi         a4         3.1  2008     6 auto(a~ f        18    27 p     comp~
##  8 audi         a4 quat~   1.8  1999     4 manual~ 4        18    26 p     comp~
##  9 audi         a4 quat~   1.8  1999     4 auto(l~ 4        16    25 p     comp~
## 10 audi         a4 quat~   2    2008     4 manual~ 4        20    28 p     comp~
## # ... with 224 more rows
```

Example: Fuel economy data

The variables are mostly self-explanatory:

- `cty` and `hwy` record miles per gallon (mpg) for city and highway driving.
- `displ` is the engine displacement in liters (engine size).
- `drv` is the drivetrain: front wheel (f), rear wheel (r) or four wheel (4).
- `model` is the model of car. There are 38 models, selected because they had a new edition every year between 1999 and 2008.
- `class` is a categorical variable describing the “type” of car: two seater, SUV, compact, etc.

Tibbles vs Data Frames

- Tibbles are data frames with a tweak on some older behaviours:
  1. Refined print method shows only the first 10 rows and all the columns that fit on screen. Includes column name and type.
  2. Subsetting can be done with `$` by name, and `[[` by name or position.

```
mpg$cty
mpg[["cty"]]
mpg[[8]]
```

- Most packages use regular data frames (`data.frame()`). Use `as.data.frame()` to turn a tibble back to a data.frame.

- Use `as_tibble()` to coerce a data.frame to a tibble.

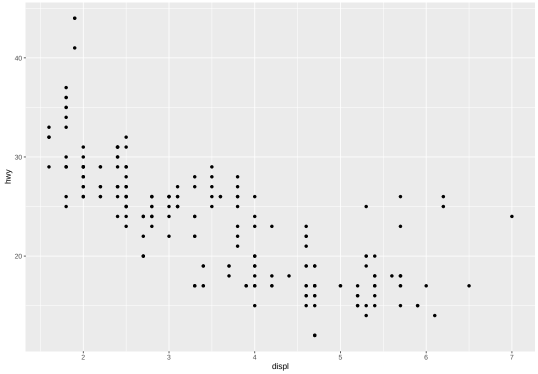
Key components ggplot2 plots

Every plot has three key components:

1. **data**,
2. A set of **aesthetic mappings** between variables in the data and visual properties, and
3. At least one layer which describes how to render each observation. Layers are usually created with a **geom** function.

Example: Scatterplot of engine size vs hwy mileage

```r
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point()
```



Example: Scatterplot of engine size vs hwy mileage

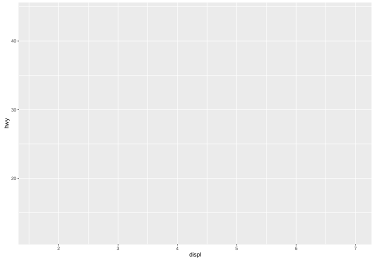
This produces a scatterplot defined by:

1. Data: mpg.
2. Aesthetic mapping: engine size mapped to `x` position, fuel economy to `y` position.
3. Layer: points.

Summary of ggplot2 plots

1. With `ggplot()`:
   - The first argument is always the name of the dataset you wish to use for plotting.
   - Next, you provide the variables from the dataset to be assigned to different aesthetic elements of the plot, such as the `x` and the `y` axes.

```r
ggplot(mpg, aes(x = displ, y = hwy))
```



Summary of ggplot2 plots (contd.)

2. Next, you tell `ggplot()` what type of visualization you would like to add to the blank template.
   - You add another layer to the `ggplot()` by adding a `+` at the end of the line, to indicate that you are adding a layer
   - then specify the **geom**etric object to be used to create the plot.

```r
ggplot(mpg, aes(displ, hwy)) +
  geom_point()
```

This tells `ggplot()` that each data point should be represented by one point on the plot. The first two unnamed arguments to aes() will always be mapped to x and y.

Other Plot geoms

You can substitute `geom_point()` for a different geom function to get a different type of plot.

- `geom_boxplot()` produces a box-and-whisker plot to summarise the distribution of a set of points. Use `geom_violin()` for violin plots.
- `geom_histogram()` and `geom_freqpoly()` show the distribution of continuous variables.
- `geom_bar()` shows the distribution of categorical variables.
- `geom_line()` draw lines between the data points.

Colour, size, shape and other aesthetic attributes

To add additional variables to a plot, we can use other aesthetics like colour, shape, and size into the call to aes():

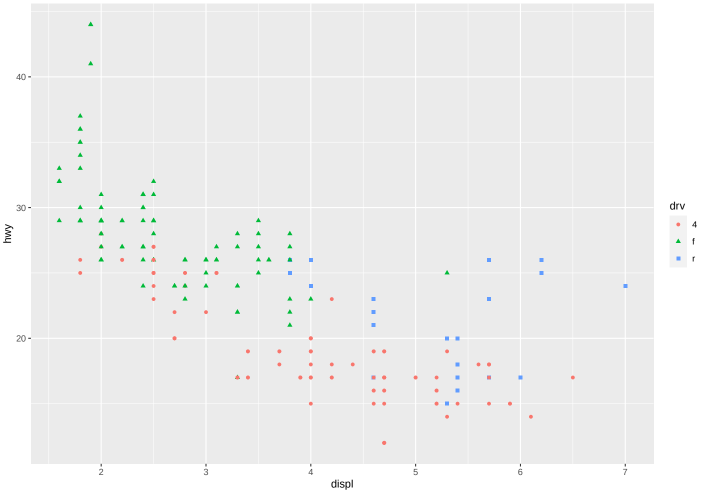
- `aes(displ, hwy, color = class)`, better with categorical.
- `aes(displ, hwy, shape = drv)`, better with categorical.
- `aes(displ, hwy, size = cyl)`, better with numerical.

The following code gives each point a unique color corresponding to its class:

```r
ggplot(mpg, aes(displ, hwy, color = class)) +
  geom_point()
```

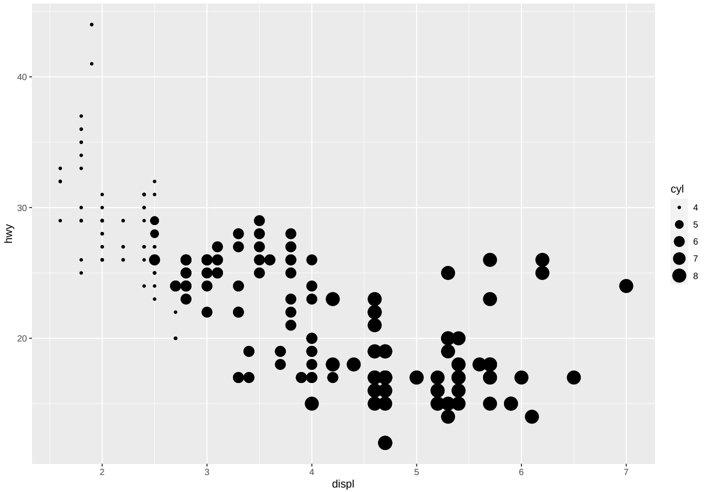
Example: Scatterplot of engine size vs hwy mileage by drivetrain

```r
ggplot(mpg, aes(displ, hwy, shape = drv, color = drv)) +
  geom_point()
```



Example: Scatterplot of engine size vs hwy mileage by cylinders

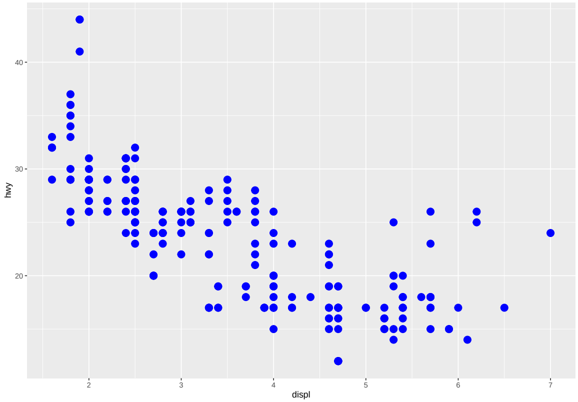
```r
ggplot(mpg, aes(displ, hwy, size = cyl)) +
  geom_point()
```



Example: Aesthetic to a fixed value

If you want to set an aesthetic to a fixed value, without scaling it, do so in the individual layer outside of `aes()`.

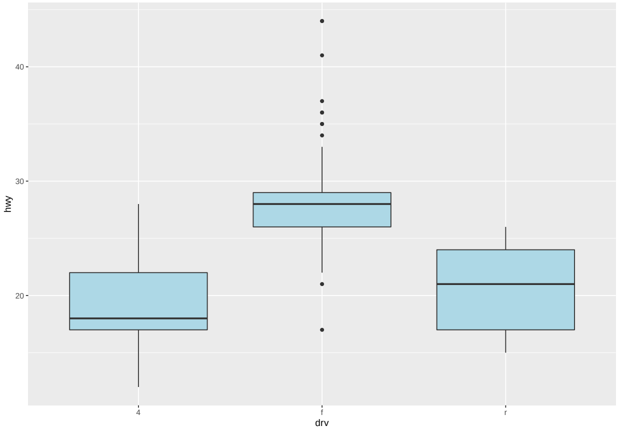
```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(colour = "blue", size=4)
```



Example: Boxplots of hwy mileage by drivetrain

Boxplots summarise the bulk of the distribution with the median, IQR and whiskers.

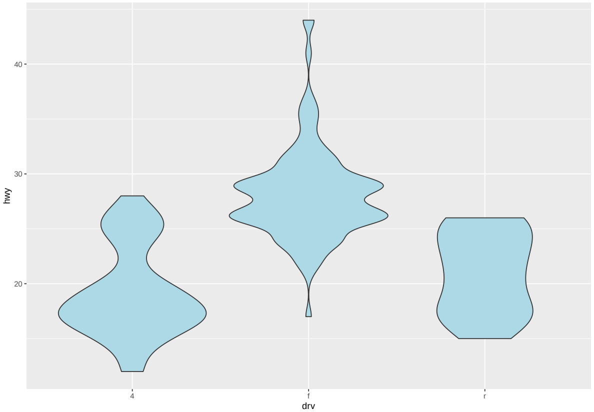
```r
ggplot(mpg, aes(drv, hwy)) +
  geom_boxplot(fill="lightblue")
```



Example: Violin plots of hwy mileage by drivetrain

Violin plots give the richest display, but rely on the calculation of a density estimate, which can be hard to interpret.

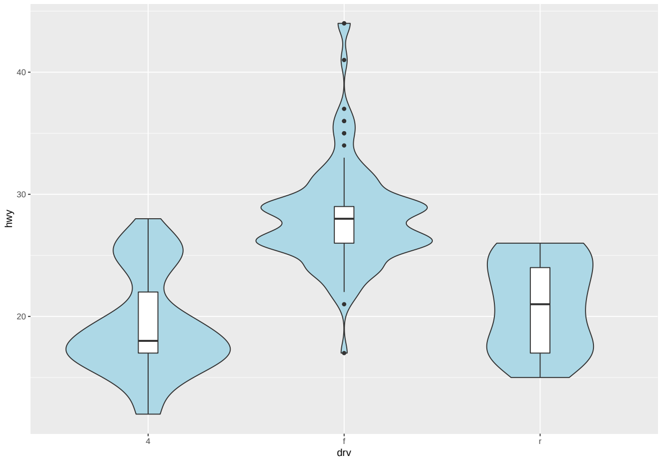
```r
p <- ggplot(mpg, aes(drv, hwy)) +
        geom_violin(fill="lightblue")
p
```



Example: Adding layers to plots

Violin plots of hwy mileage by drivetrain with boxplots

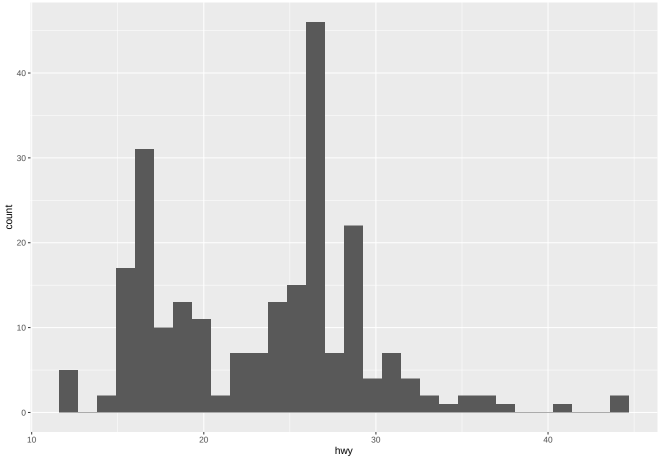
```
p + geom_boxplot(width=0.1)
```



Example: Histograms of highway mpg

Histograms show the distribution of a single numeric variable.

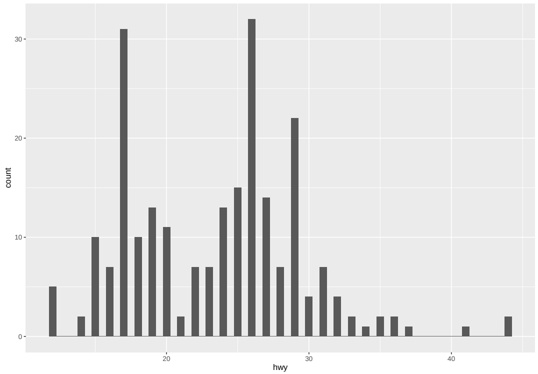
```r
ggplot(mpg, aes(hwy)) + geom_histogram()
```



Example: Histograms of highway mpg

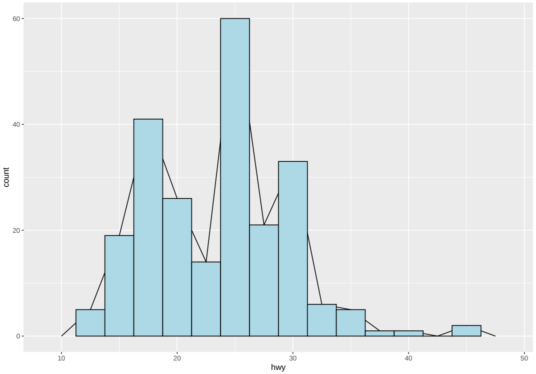
- You can control the width of the bins with the binwidth argument.
- The default just splits your data into 30 bins (binwidth = 1).
- Try many bin widths to tell the full story of your data.

```r
ggplot(mpg, aes(hwy)) + geom_histogram(binwidth = 0.5)
```



Example: Frequency polygons of highway mpg

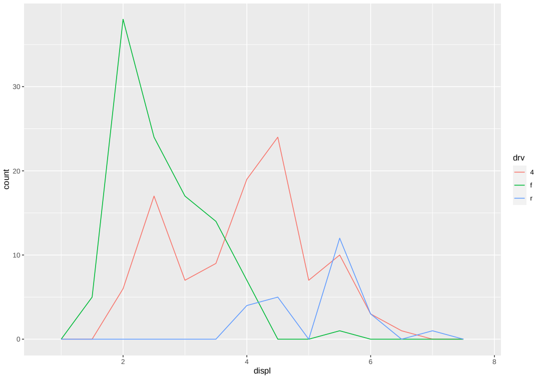
```r
ggplot(mpg, aes(hwy)) + geom_freqpoly(binwidth = 2.5) +
  geom_histogram(binwidth = 2.5, color = "black",
                 fill = "lightblue")
```



Example: Frequency polygons of engine size by drivetrain

To compare the distributions of different subgroups, you can map a categorical variable to either `fill` (for `geom_histogram()`) or `color` (for `geom_freqpoly()`):

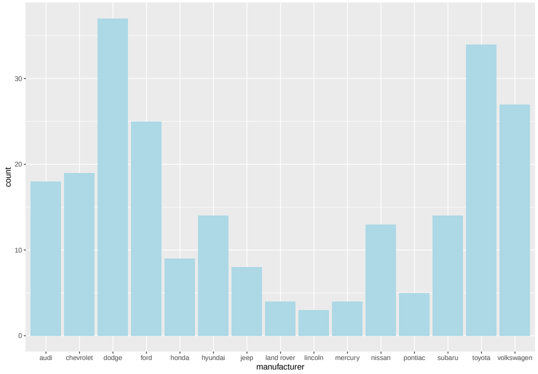
```r
ggplot(mpg, aes(displ, color = drv)) +
  geom_freqpoly(binwidth = 0.5)
```



Example: Bar chart of manufacturer

The discrete analogue of the histogram is the bar chart:

```r
ggplot(mpg, aes(manufacturer)) +
  geom_bar(fill = "lightblue")
```



- Note that we are using unsummarised data.

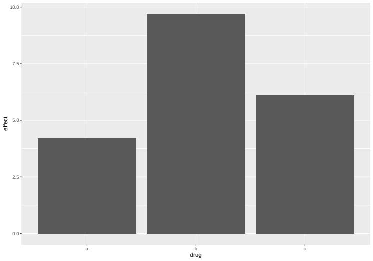
Example: Bar chart for summarised data

- Toy example of data frame

```r
drugs <- data.frame(drug = c("a", "b", "c"),
  effect = c(4.2, 9.7, 6.1))
```

- To display this sort of data, you need to tell geom_bar() to not run the default stat which bins and counts the data:

```r
ggplot(drugs, aes(drug, effect)) +
  geom_bar(stat = "identity")
```


effect
a
b
c
drug