Graphing Data Using GGPlot

Jan-Philipp Kolb

25 Februar, 2020

GGplot vs Base

`ggplot` attempts to create a consistent framework for build graphs "layer by layer" in R.

You construct a graph by specifying:

1. The data.
2. An aesthetic (e.g., colors, line styles, the coordinate system, etc).
3. A graph "geometry" (e.g., boxplot, scatterplot, etc). This is where you specify the kind of graph you want.
4. Labels. The plot title, axis labels, etc.

A scatterplot

Creating the graph object and specifying the dataset.

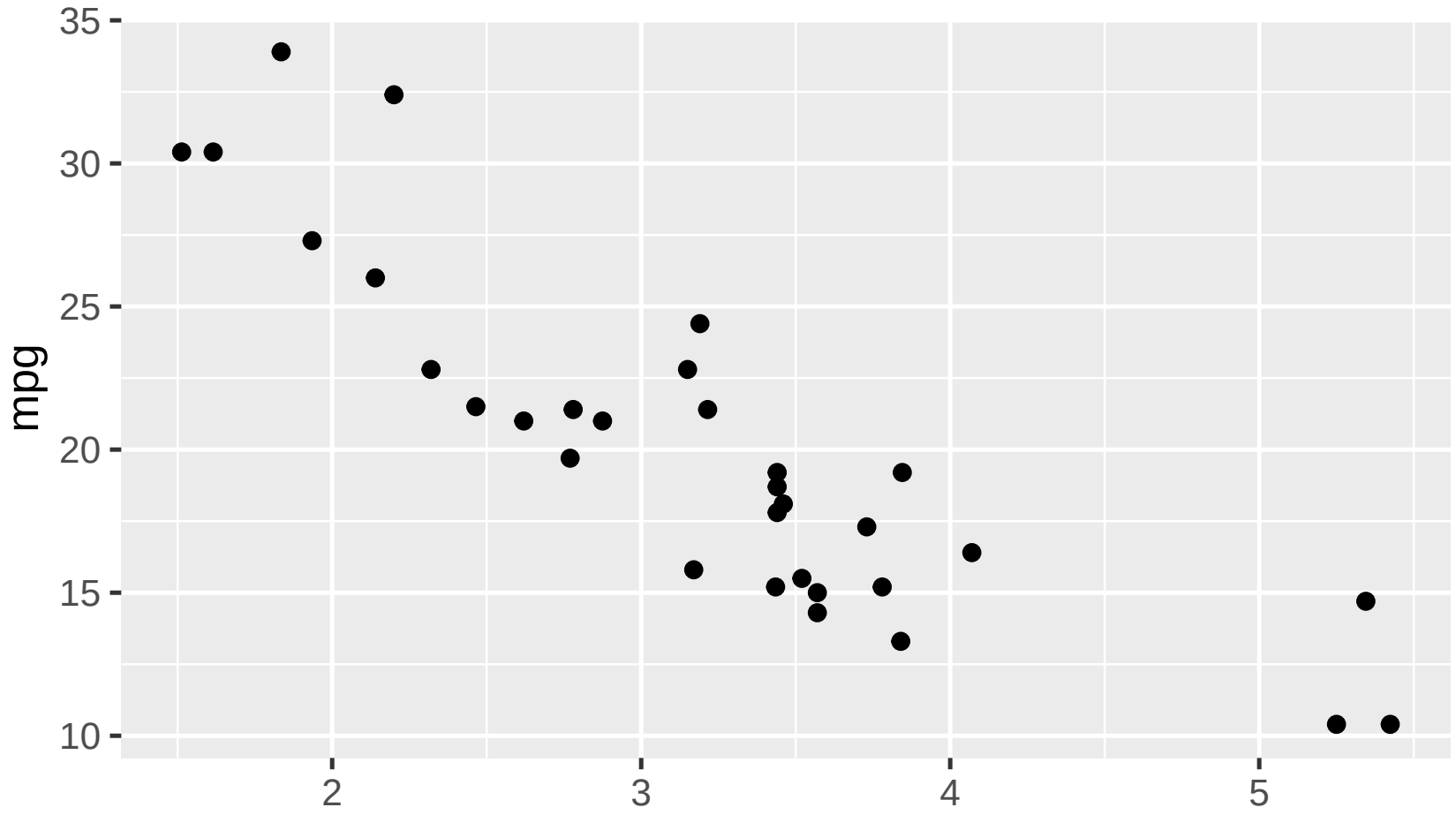
```
library(ggplot2)
car.graph <- ggplot(mtcars)
```

- Specifying what aesthetics to use.
- In this case, the coordinate system to use – meaning the x-y axis.
- This is sometimes also referred to as the "mapping" being used.

```
car.graph <- car.graph + aes(wt, mpg)
```

Specifying the plot

```
## Specifying the plot "geometry" in this case, a scatter plot.
car.graph <- car.graph + geom_point()

## Calling the graph object reproduces the plot.
car.graph
```

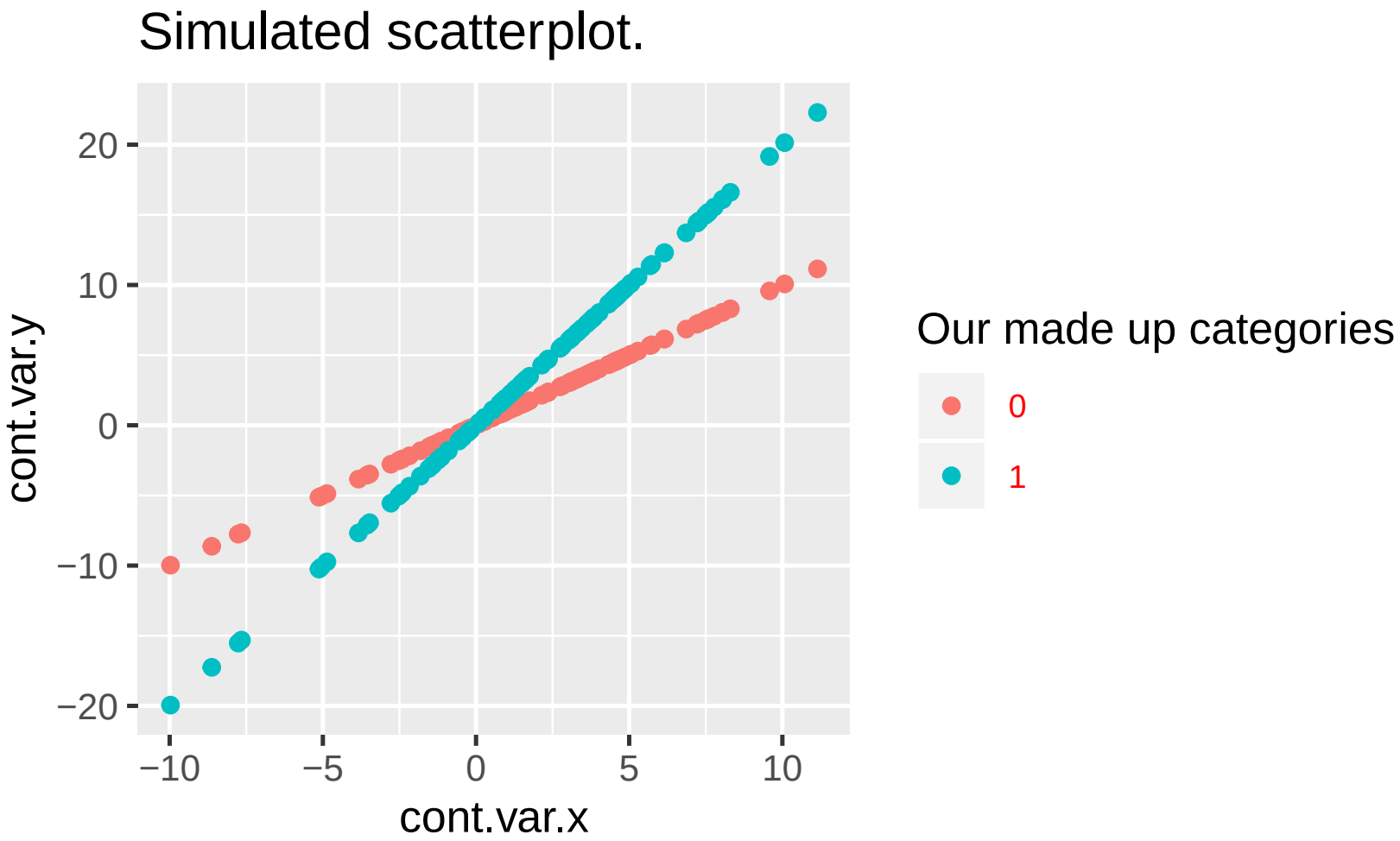


Exercise: A scatterplot

Use the code below to generate a random dataset. Then plot the results as a scatterplot, using ggplot2. Don't worry if your labels are different than mine. I'll cover some of those details after the exercise.

```
set.seed(42)
cat.var <- c(rep.int(0, 100), rep.int(1, 100))
cont.var.x <- rnorm(n=100, mean=2, sd=4)
cont.var.y <- cont.var.x + cont.var.x*cat.var
simulated.dataset <- data.frame(cont.var.x, cont.var.y, cat.var)
```

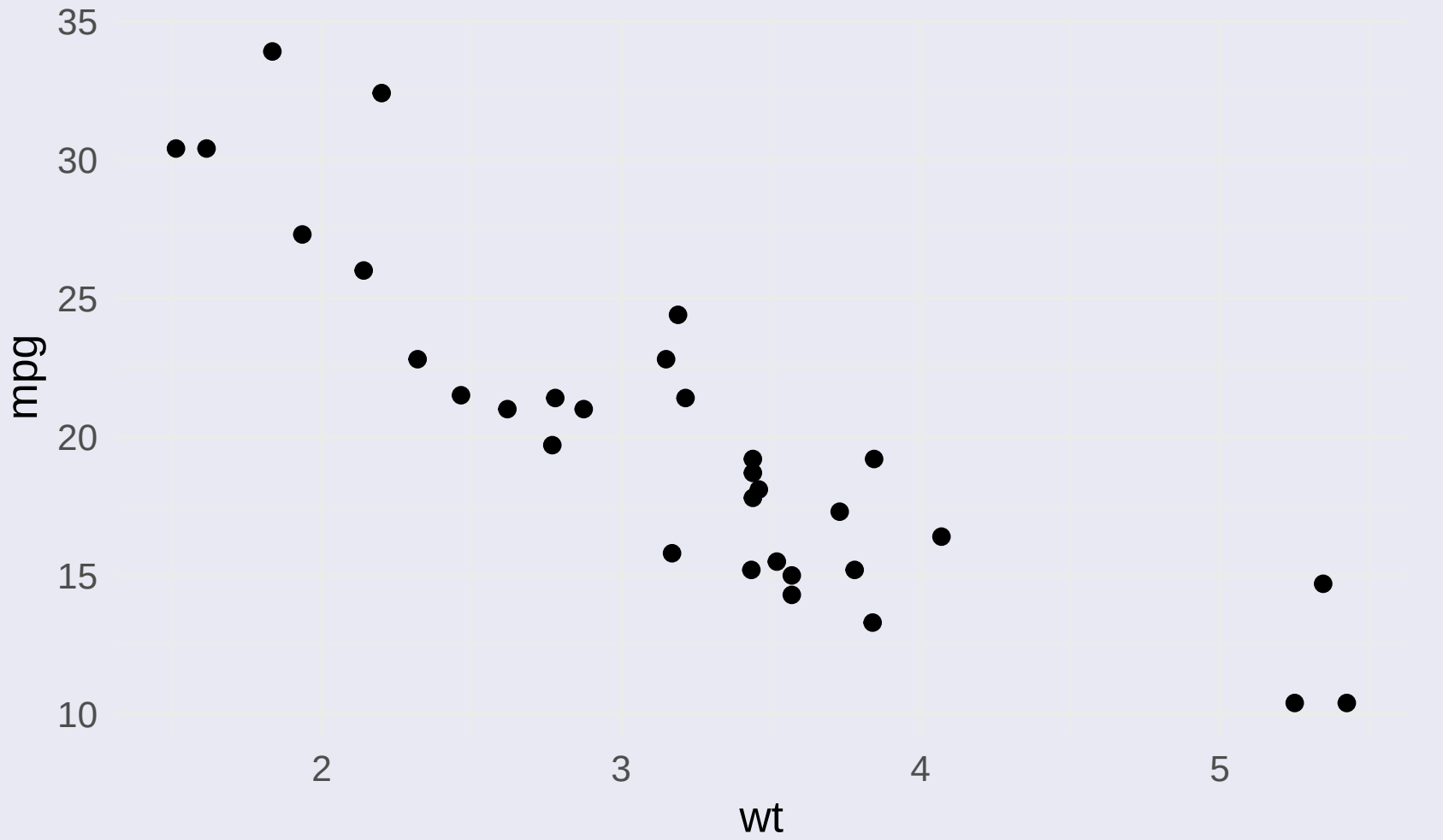
What the result should look like



Specifying different aesthetics

Themes

```
## Minimal theme
ggplot(mtcars) + aes(wt, mpg) + geom_point() +
    theme_minimal()
```



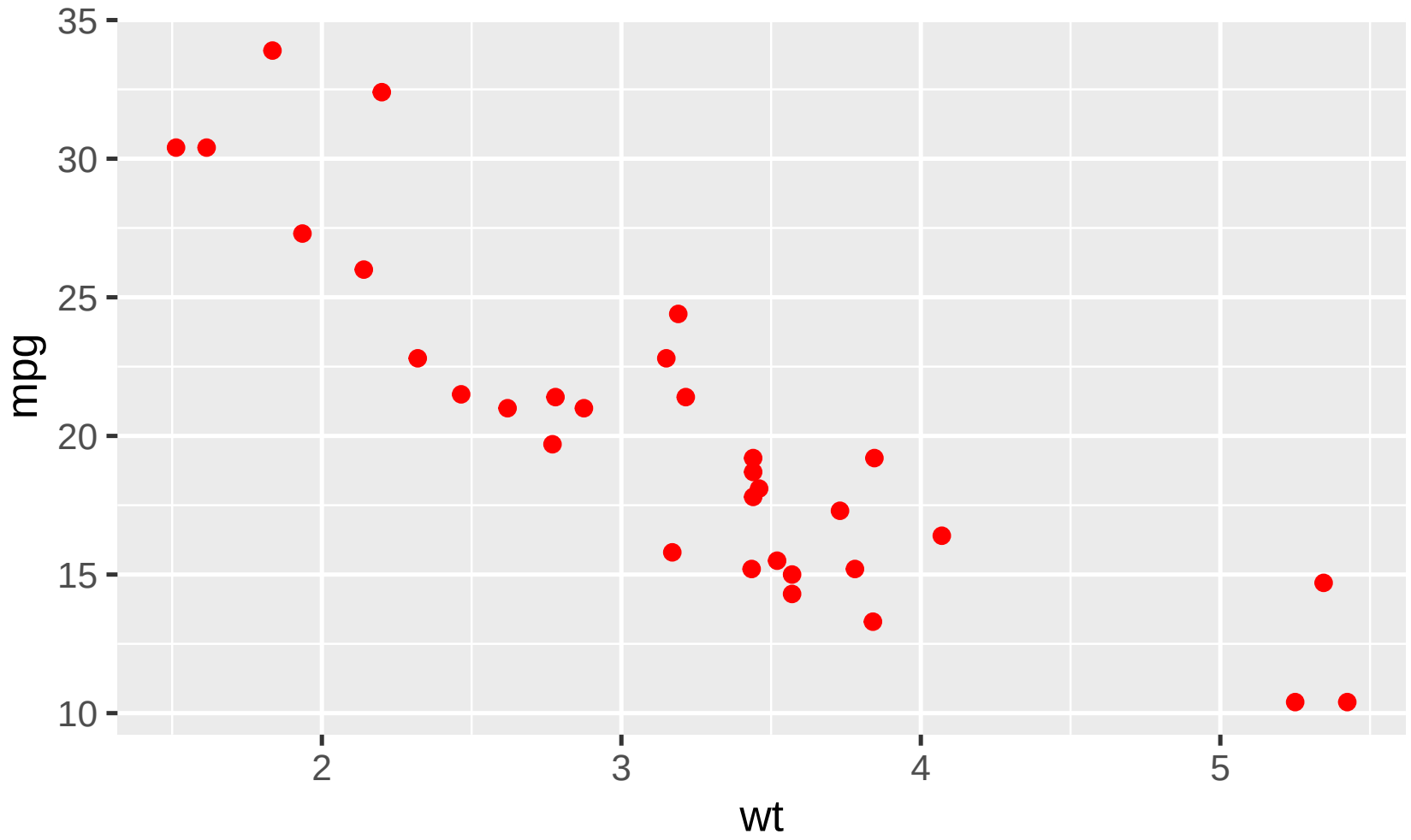
Themes

```
ggplot(mtcars) + aes(wt, mpg) + geom_point() + theme_gray() ## t
ggplot(mtcars) + aes(wt, mpg) + geom_point() + theme_dark() ## D
ggplot(mtcars) + aes(wt, mpg) + geom_point() + theme_classic() #
ggplot(mtcars) + aes(wt, mpg) + geom_point() + theme_void() ## U
```

There are a number of other built in themes, but you get the idea. You can also create your own themes if one of the built in ones doesn't do what you want.

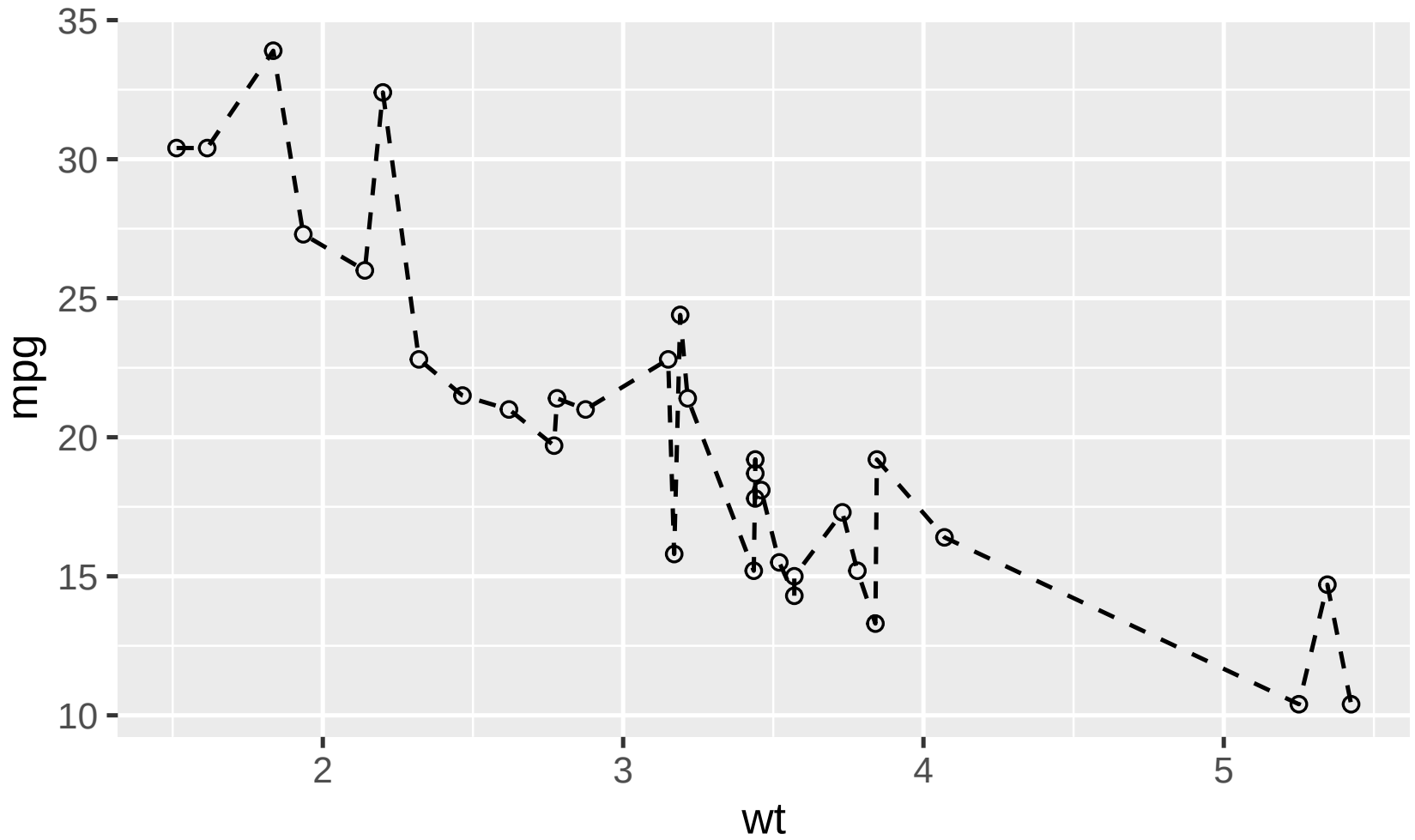
Colors

```
car.graph <- ggplot(mtcars) + aes(wt, mpg) +
    geom_point(color="red")
car.graph
```



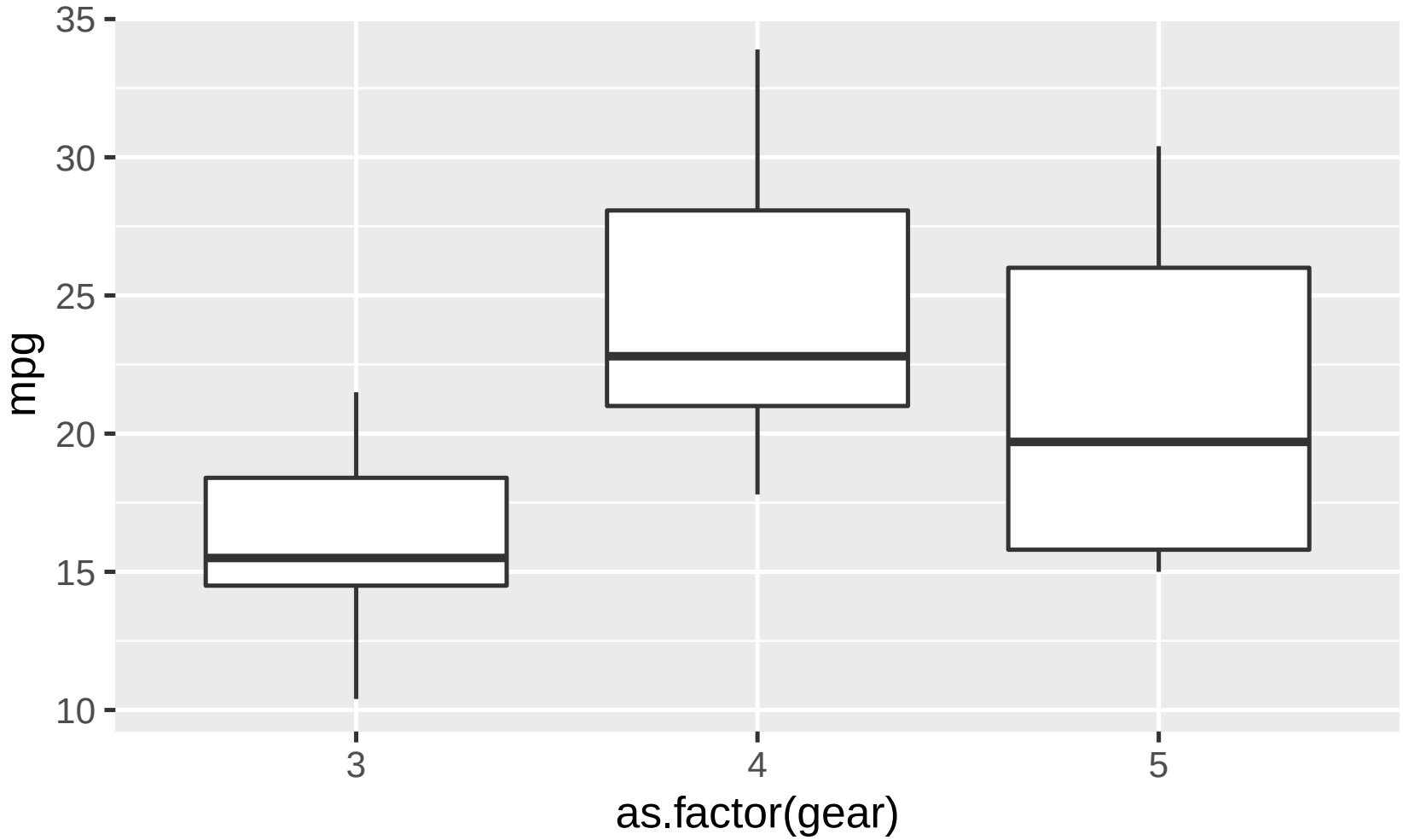
POINT/LINE STYLES.

```
car.graph <- ggplot(mtcars) + aes(wt, mpg) +
    geom_point(shape=21) + geom_line(linetype=2)
car.graph
```



A boxplot

```
ggplot(mtcars) + aes(as.factor(gear), mpg) +geom_boxplot()
```

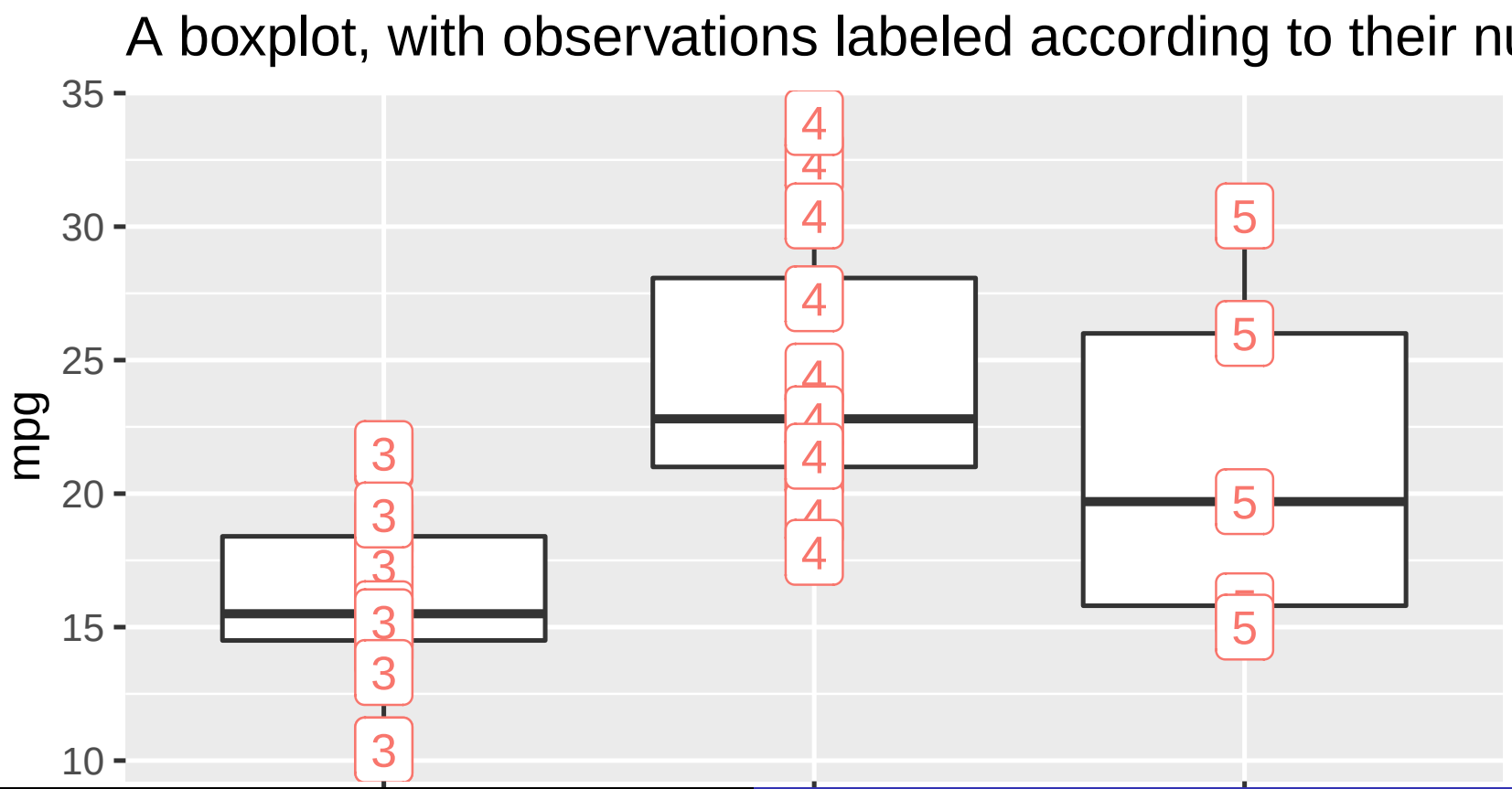


Adding a title/changing labels

- geom_text() adds text to a plot.
- geom_label() adds stuff to make the text easier to read (e.g., a box around the text).
- labs() modifies your labels/title.
- theme() lets us manipulate stuff like the inclusion of a legend, its position, etc.

Adding a title/changing labels

```
ggplot(mtcars) + aes(as.factor(gear), mpg) +
    geom_boxplot() +
    geom_text(aes(label=as.factor(gear), col="red")) +
    geom_label(aes(label=as.factor(gear), col="red"))+
    labs(x="A different label than earlier.", title="A boxplot,
    theme(legend.position="none")
```

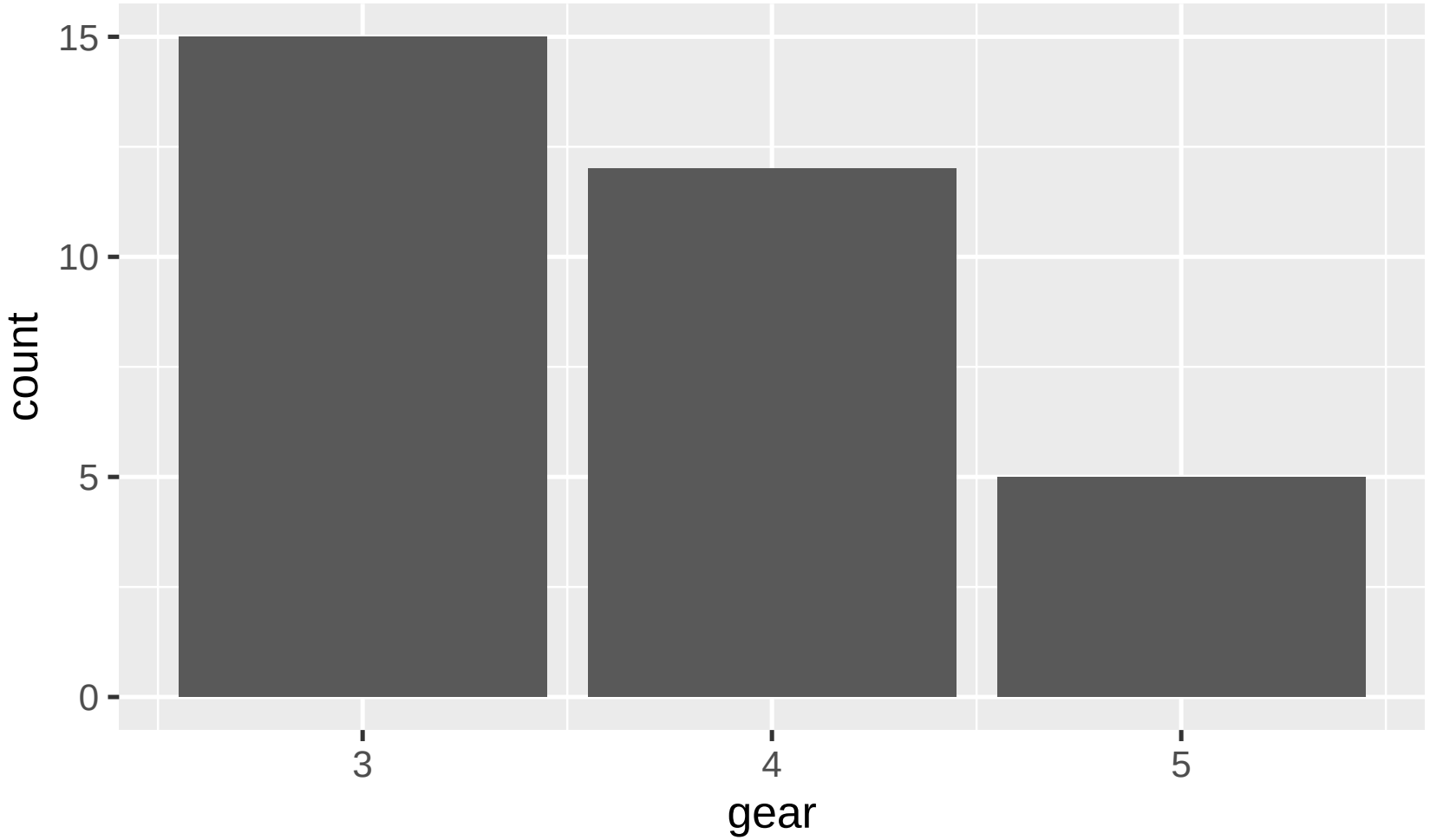


Saving a graph

```
## Saving to pdf
ggsave(car.graph, file="car_graph.pdf")

## Saving to pdf, while specifying dimensions of plot
ggsave(car.graph, file="car_graph.pdf", width = 20,
       height = 20, units = "cm")

## Saving to png
ggsave(car.graph, file="car_graph.png")

## Other formats are possible.
# See ?ggsave for more information.
```

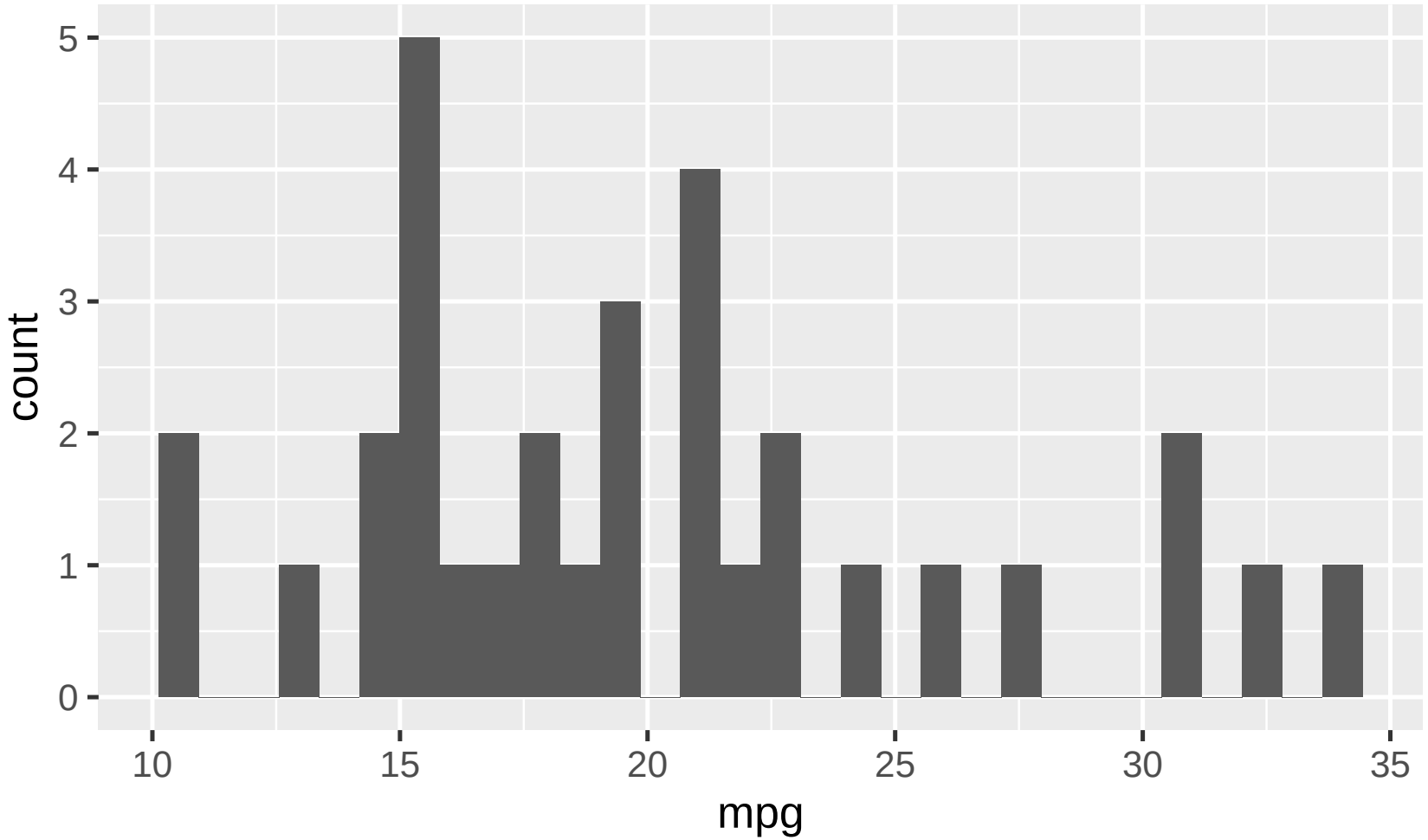
A barplot

```
## Data + aesthetics + geometry.
ggplot(mtcars)+aes(gear)+geom_bar()
```



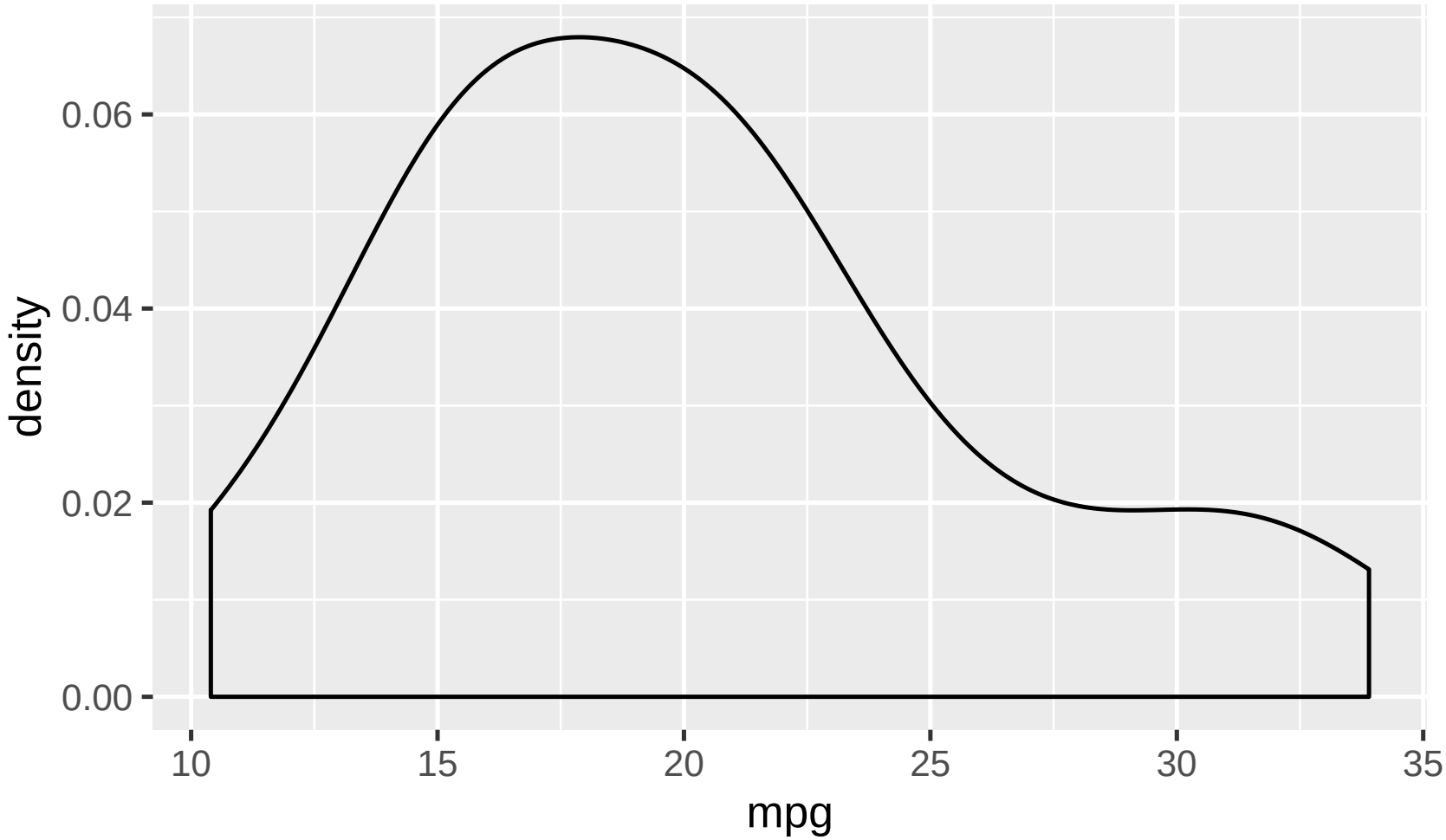
A histogram

```
## Data + aesthetics + geometry.
ggplot(mtcars)+aes(mpg)+geom_histogram()
```



A density plot

```
ggplot(mtcars)+aes(mpg)+geom_density()
```



Exercise: Histogram and density plot

Using your data from the scatterplot exercise, produce a histogram for cont.var.x, and a density plot for cont.var.y.

Links to read on

- R cookbook for graphs