line graphs, histograms & boxplots

Emily Malcolm-White

```
library(tidyverse)
```

Recall: The `mtcars` dataset was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). It's available inside the `ggplot` package which is already installed.

```
#load the data
data(mtcars)
```

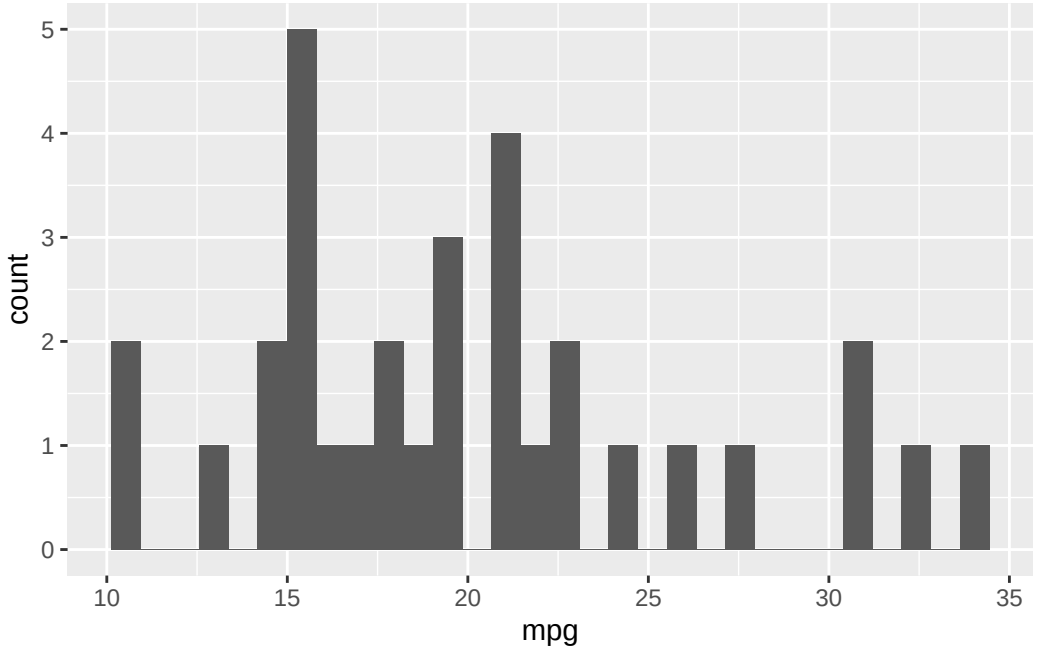
Code to update `mtcars` dataset so that `am` is treated as a factor rather than a continuous numeric variable

```
mtcars <- mtcars %>%
  mutate(am = as.factor(am))
```

Histograms

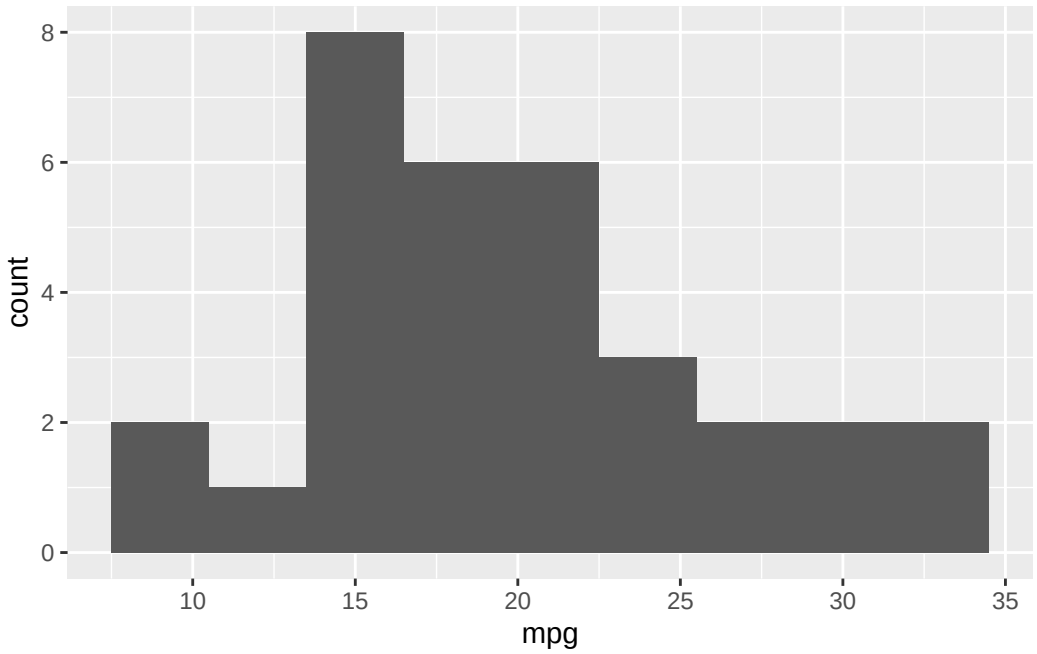
> Tip
>
> Histograms are great for looking at the distributions of numeric variables

```
mtcars %>%
  ggplot(aes(x=mpg)) +
  geom_histogram()
```



You can control the bin size with `binwidth`

```
mtcars %>%
  ggplot(aes(x=mpg)) +
  geom_histogram(binwidth=3)
```



Boxplots

> Tip
>
> Boxplots are good for displaying the spread, central tendency, and distribution of one numeric variable.

```
mtcars %>%
  ggplot(aes(y=mpg)) +
  geom_boxplot()
```

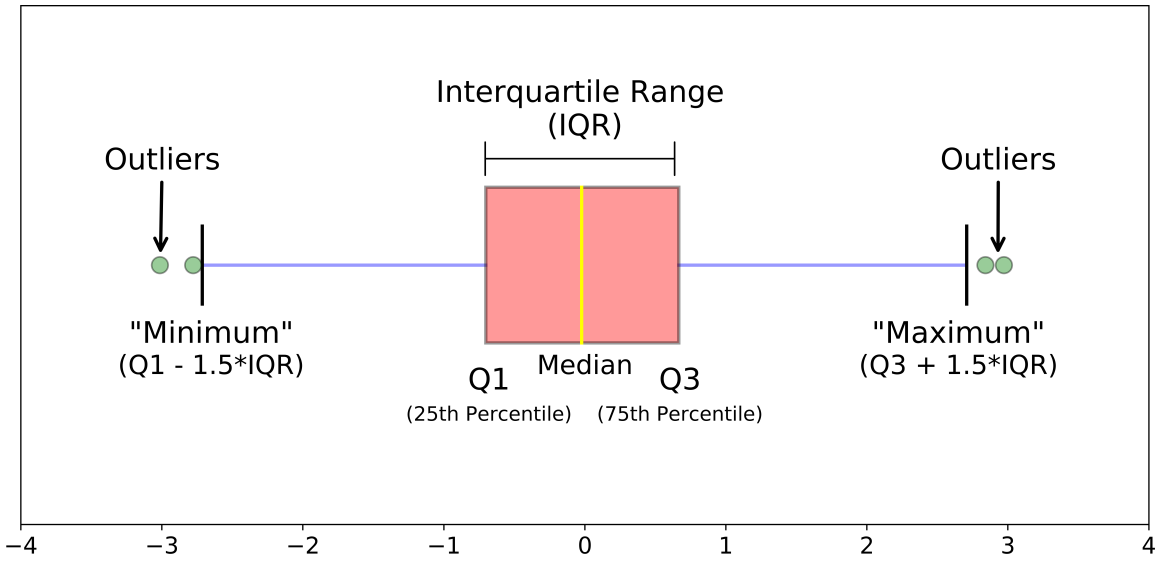
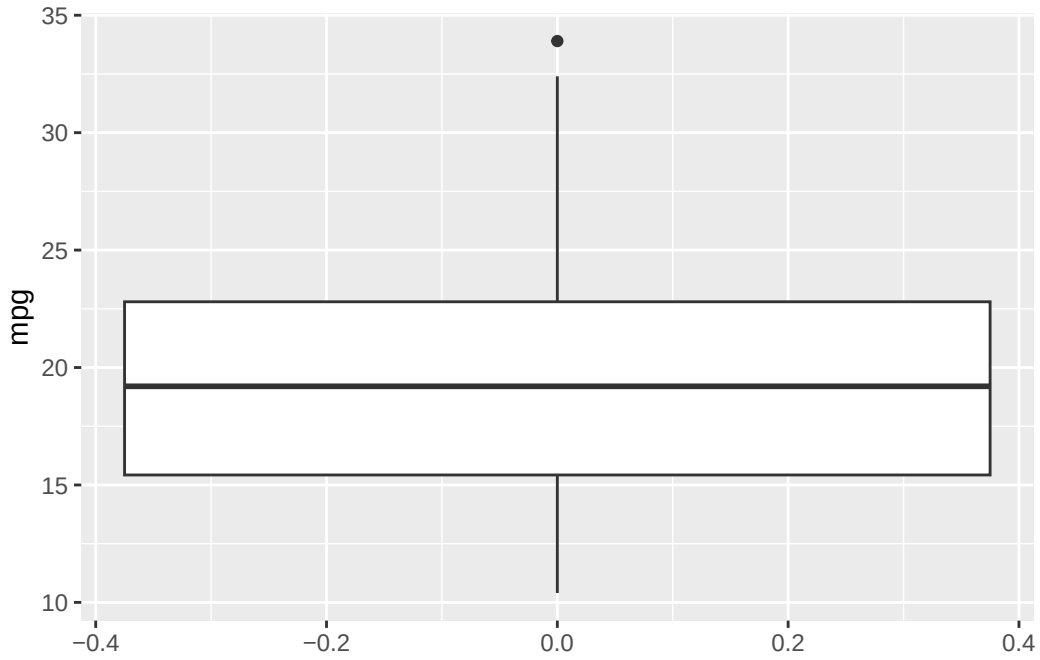


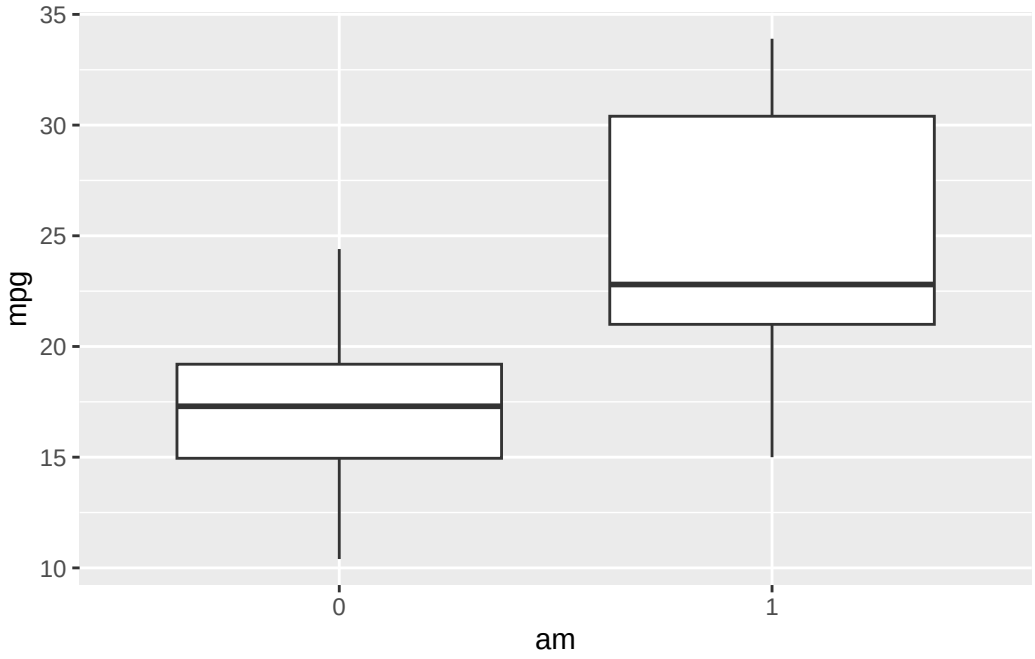
Figure 1: Credit: Michael Galarnyk



Tip

Side-by-side boxplots are good for displaying one categorical variable and one numeric variable. One advantage of boxplots over barplots is that they are able to show a bit about the spread and distribution of the numeric variable!

```
mtcars %>%
  ggplot(aes(x=am, y=mpg)) +
  geom_boxplot()
```



Line Graphs

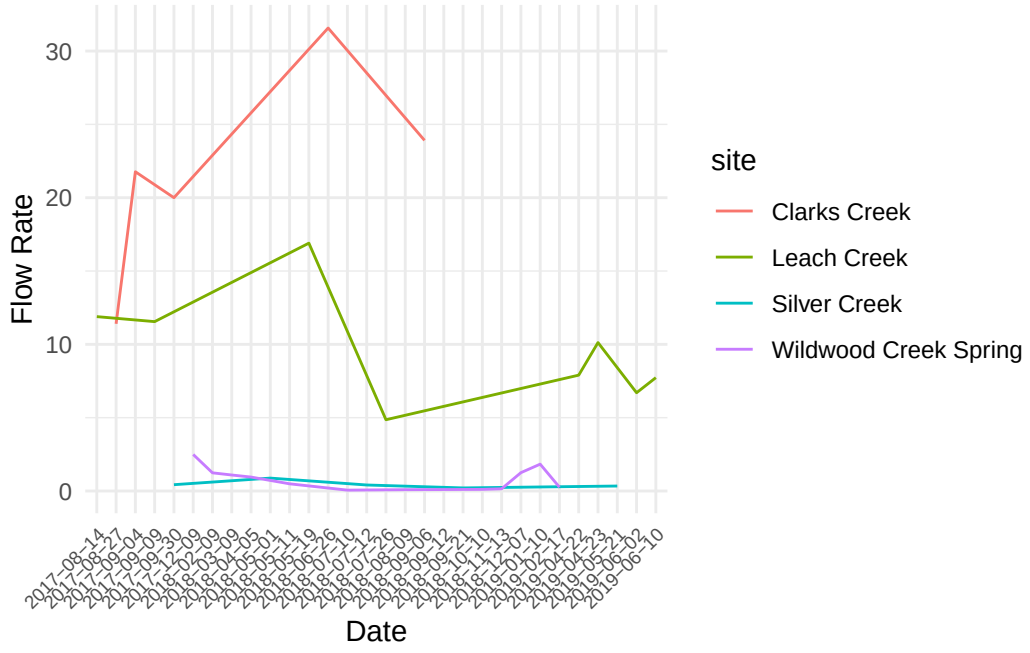
Tip

Line graphs are great for showing trends with respect to ordinal (ordered variables). Time is used quite commonly.

We consider data on river flow rates collected by volunteers of the Pierce Conservation District in WA.

```
flow_rates <- read.csv("https://www.openintro.org/data/csv/flow_rates.csv")
```

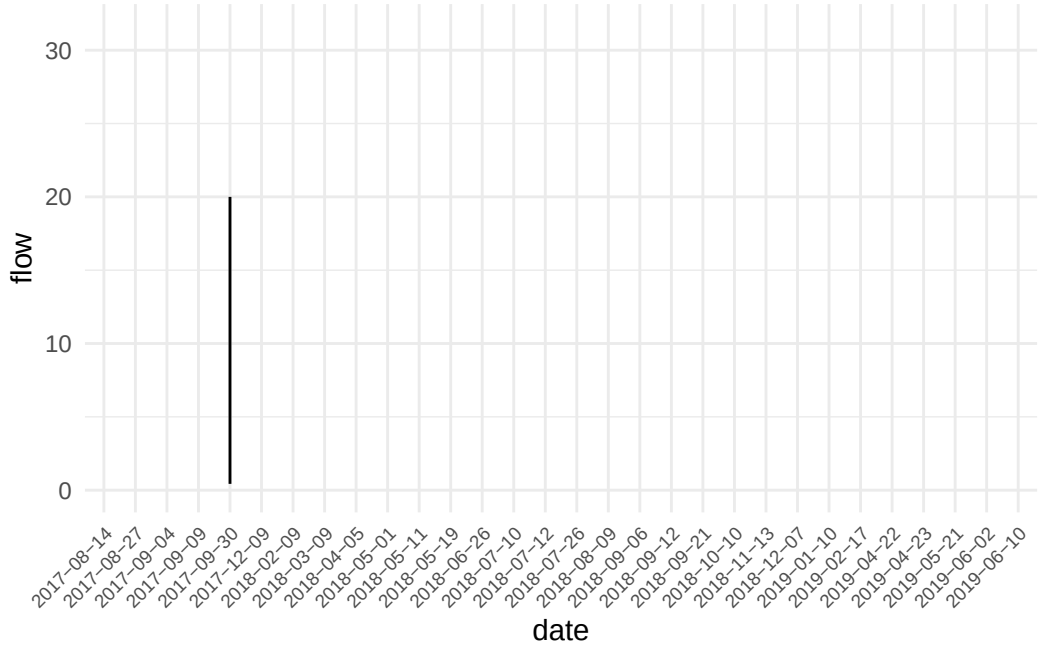
```
flow_rates %>%
  ggplot(aes(x=date, y=flow, group=site, color=site)) +
  geom_line() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size=7)) +
  labs(x="Date", y="Flow Rate")
```



Warning

What happens if we didn't have the `group=` command?

```
flow_rates %>%
  ggplot(aes(x=date, y=flow)) +
  geom_line() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size=7))
```

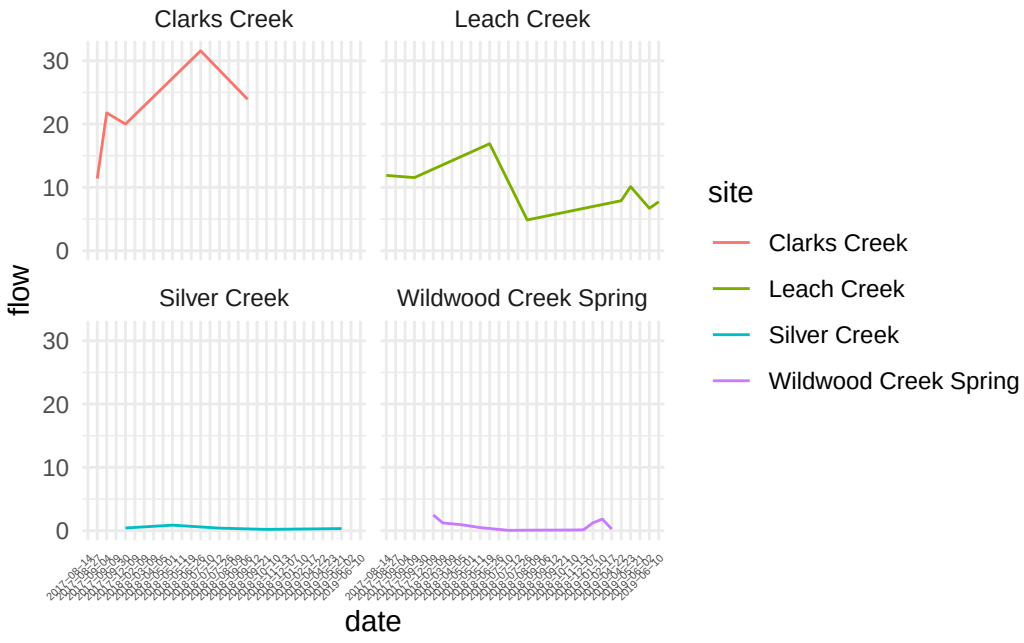


Yikes!
If there is grouping in the data, you need to either: - seperate by group by using the `group=` command - Merge together (add? average?) the groups using `summarize()` first before trying to graph

Facet Wrap

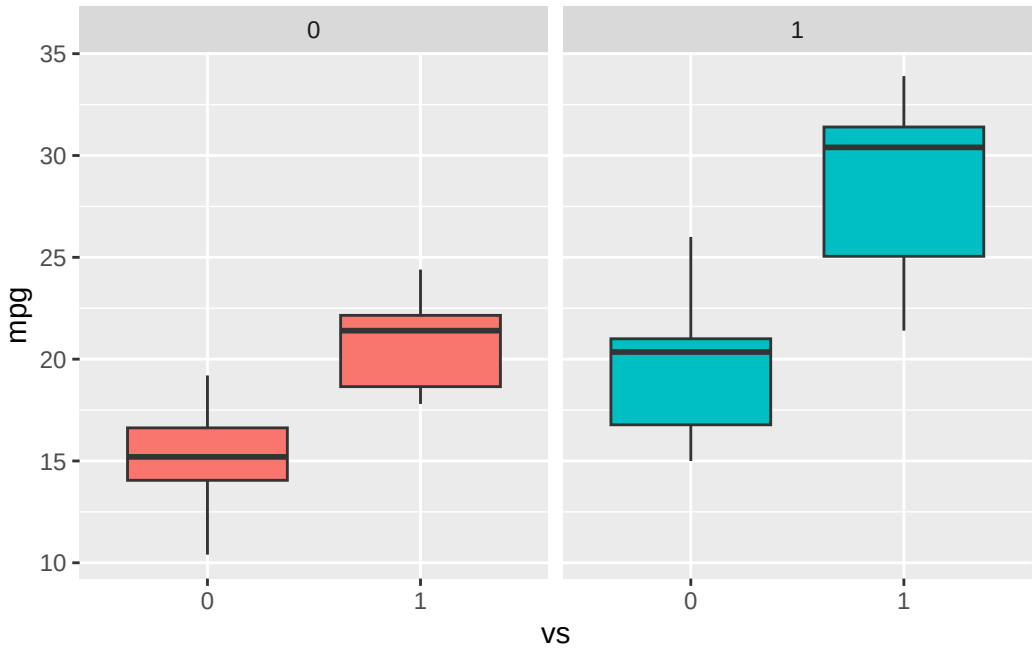
`facet_wrap()` is a function in the `ggplot2` package that allows you to create a multi-panel plot showing a similar plot over different subsets of the data, usually different values of a categorical variable.

```
# Example of Facet Wrap with `flow_rates` dataset
flow_rates %>%
  ggplot(aes(x=date, y=flow, group=site, color=site)) +
  geom_line() +
  facet_wrap(~site) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size=4))
```



You can create facets over any categorical variable in the dataset!

```
#Example of Facet Wrap with `mtcars` dataset
mtcars %>%
  mutate(vs=as.factor(vs)) %>% #engine shape 0=v-shaped, 1=manual
  ggplot(aes(x=vs, y=mpg, fill=am)) +
  geom_boxplot() +
  facet_wrap(~am) +
  theme(legend.position="none")
```


0
1
35
30
25
20
15
10
mpg
0
1
0
1
vs