Module 6: Visualization

Scott Thatcher

1/29/2020

Module Goals

Visualization is a necessary part of data exploration!

Anscombe's Quartet: Four data sets with identical statistics. Do these statistics tell the whole story?

Means:

- $\bar{x} = 3.$
- $\bar{y} = 7.5.$

Standard Deviations:

- $s_x = 11.$
- $s_y = 4.125.$

Correlation: $r = 0.816$.

Regression: $y = 0.5\ x + 3$.

Visualization is a necessary part of data exploration!

Anscombe's Quartet: Four data sets with identical statistics. Do these statistics tell the whole story?

Means:

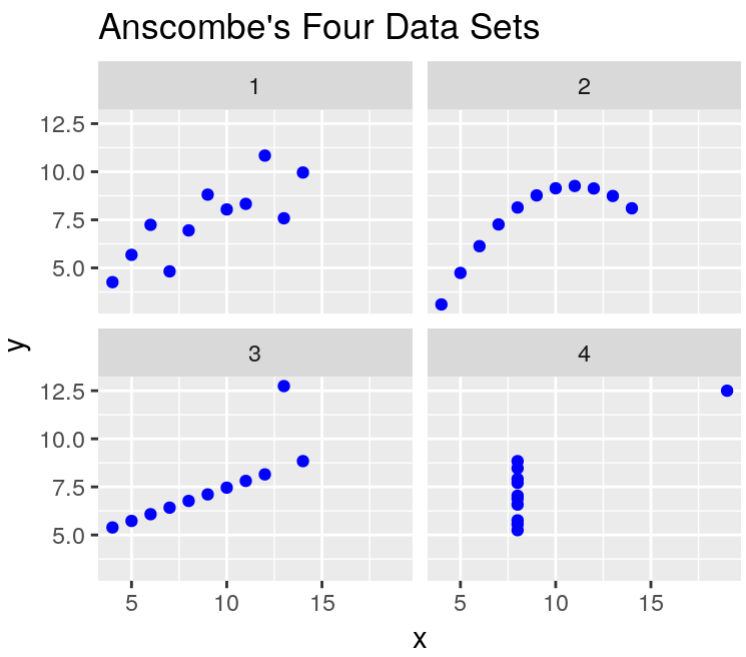
- $\bar{x} = 3.$
- $\bar{y} = 7.5.$

Standard Deviations:

- $s_x = 11.$
- $s_y = 4.125.$

Correlation: $r = 0.816$.

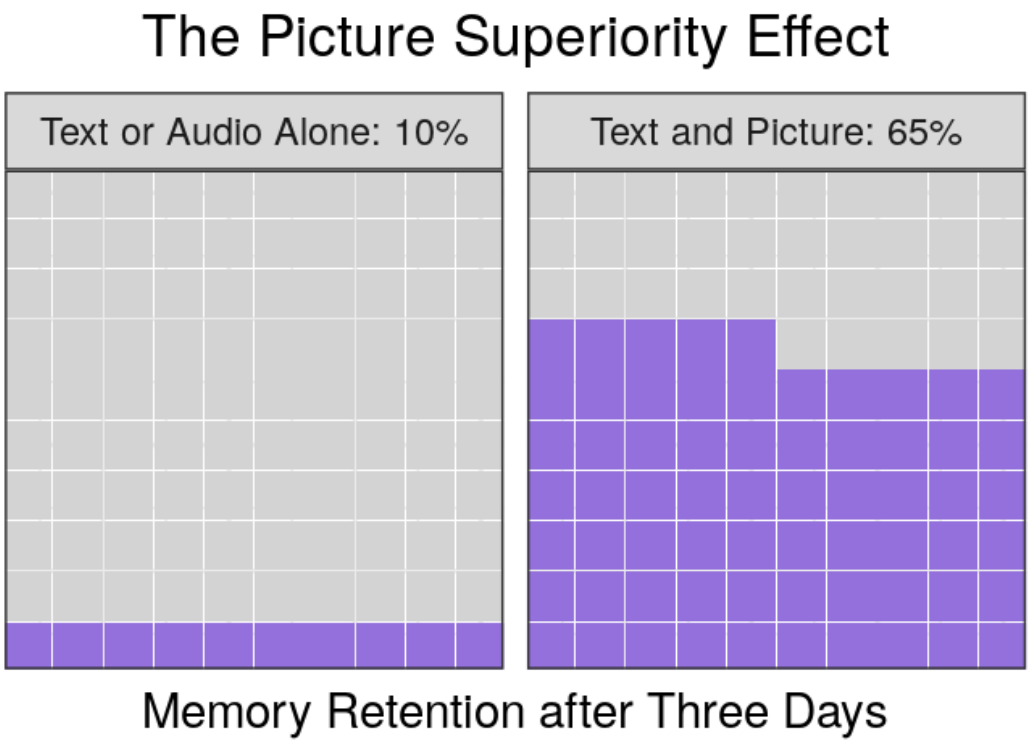
Regression: $y = 0.5\,x + 3$.



0:00 / 0:00

Subtitles

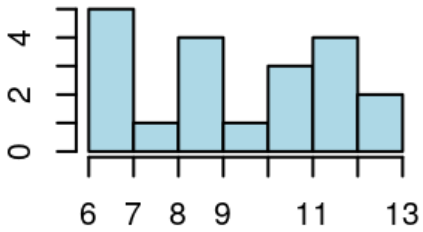
Visualization is an efficient and memorable way to present data.



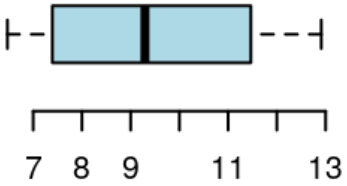
Graphic suggested by *Cool Infographics: Effective Communication with Data Visualization and Design*, Randy Krum, John Wiley & Sons, 2014.

The type of variable(s) determines appropriate graphical representations.

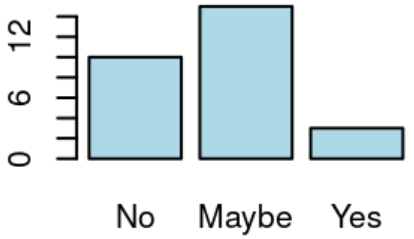
Numeric: Histogram



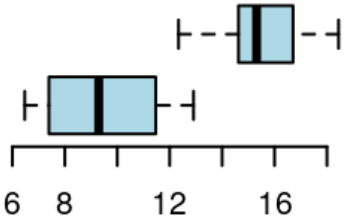
Numeric: Box Plot



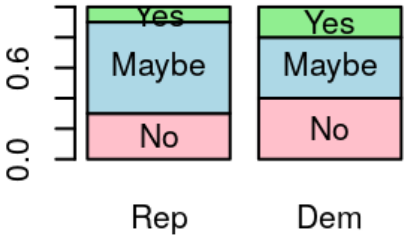
Categorical: Bar Graph



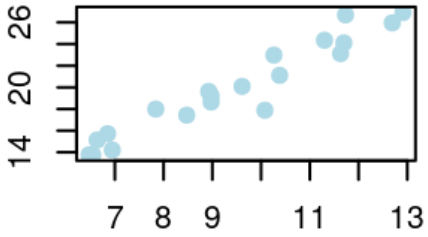
Cat. vs. Num.: Side-by-Side Box Plot



Cat. vs. Cat.: Stacked Bar Graph



Num. vs. Num.: Scatter Plot



Many software packages offer data visualization tools.

Often there's a compromise between ease-of-use and flexibility, but it's important to choose the right tool for the job.

- Spreadsheets (Excel, Libreoffice, Google Sheets)
- Statistical Software (SPSS, Minitab, …)
- Python and R (packages like `ggplot`, `plotly`, `shiny`, …)
- D3js (javascript)
- Tableau or Plotly (web-based GUI)

There are (at least) three major plotting systems in R.

- Base graphics
  - A general-purpose `plot` command that does something different depending on what you plug in.
  - A collection of special-purpose plot commands.
- `ggplot2` and its extensions
  - Implements the “grammar of graphics.”
  - Emphasis on building a final visualization from a combination of layers.
- Grid and Lattice
  - A collection of special-purpose plotting commands.
  - Attention to situations where you’d like to make multiple graphs that show how the relationship between *x* and *y* changes as a third variable *z* changes.

Some base graphics commands create plots, and others add to them.

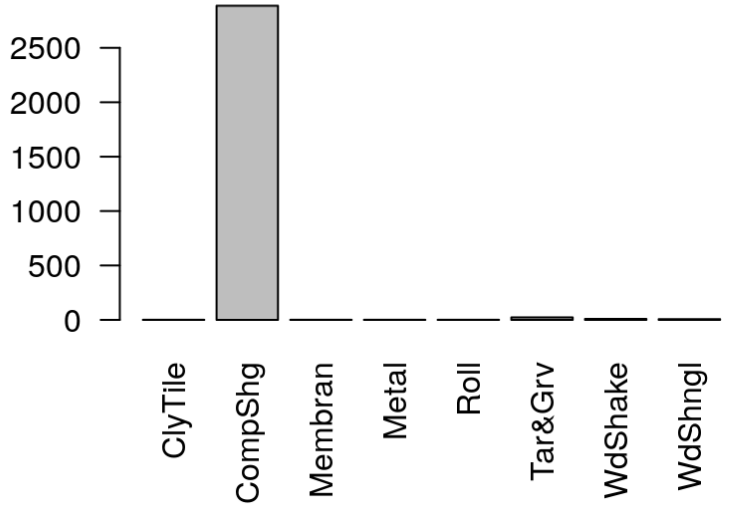
- The `par()` command sets graphics parameters prior to plot.
  - `mfrow=c(nrow, ncol)` groups multiple graphs.
  - `las` sets axis label rotation.
  - `mar=c(b, l, t, r)` changes margin spacing in graphs.
- High-level commands create new plots.

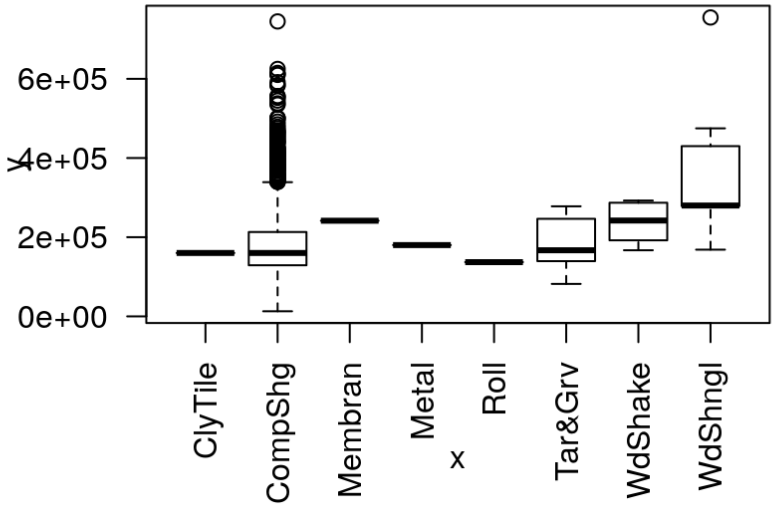
  Examples: `plot()`, `dotplot()`, `barplot()` and `boxplot()`
- Low-level commands add to existing plots.

  Examples: `axis()`, `lines()`, `text()`, `points()`

`plot()` has many "methods," depending on its input.

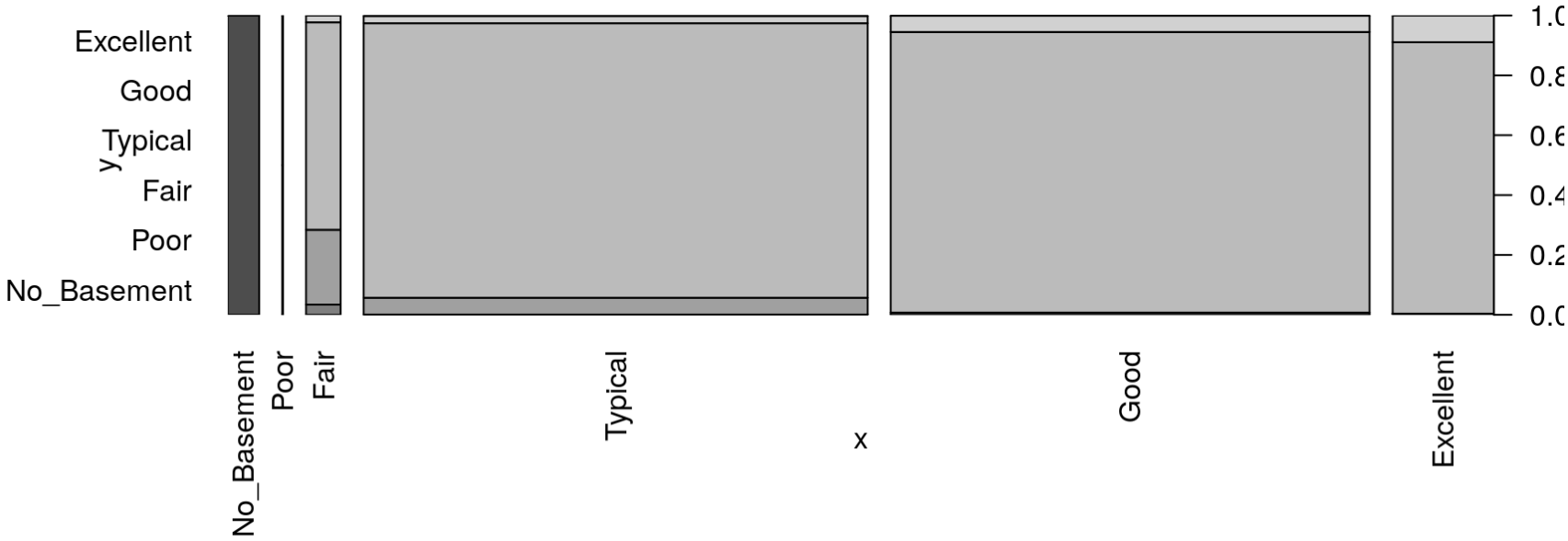
```
library(AmesHousing)
ames <- make_ordinal_ames()
par(mfrow=c(1,2), las=2)                    # Set parameters for plot
plot(x=ames$Roof_Matl)                      # One categorical var.
plot(x=ames$Roof_Matl, y=ames$Sale_Price)   # One categorical, one num.
```





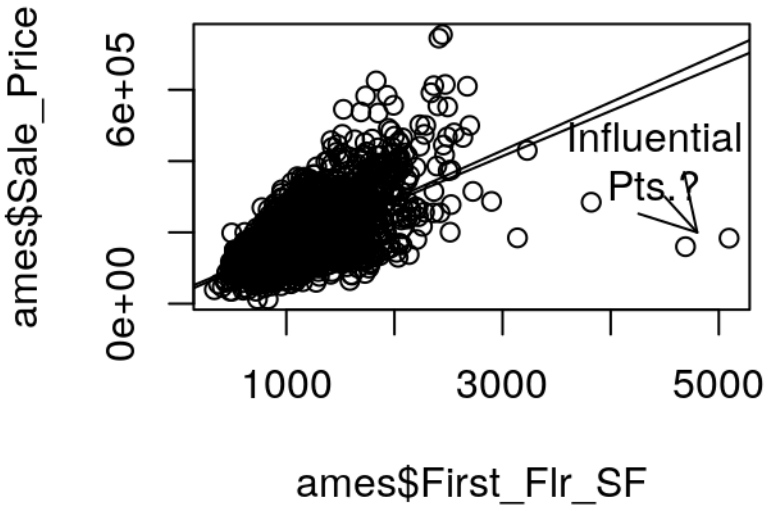
`plot()` has many "methods," depending on its input.

```
par(las=2, mar=c(7, 7, 7, 2))
plot(x=ames$Bsmt_Qual, y=ames$Bsmt_Cond)    # Two categorical variables.
```



Example: Added elements to annotate a graph.

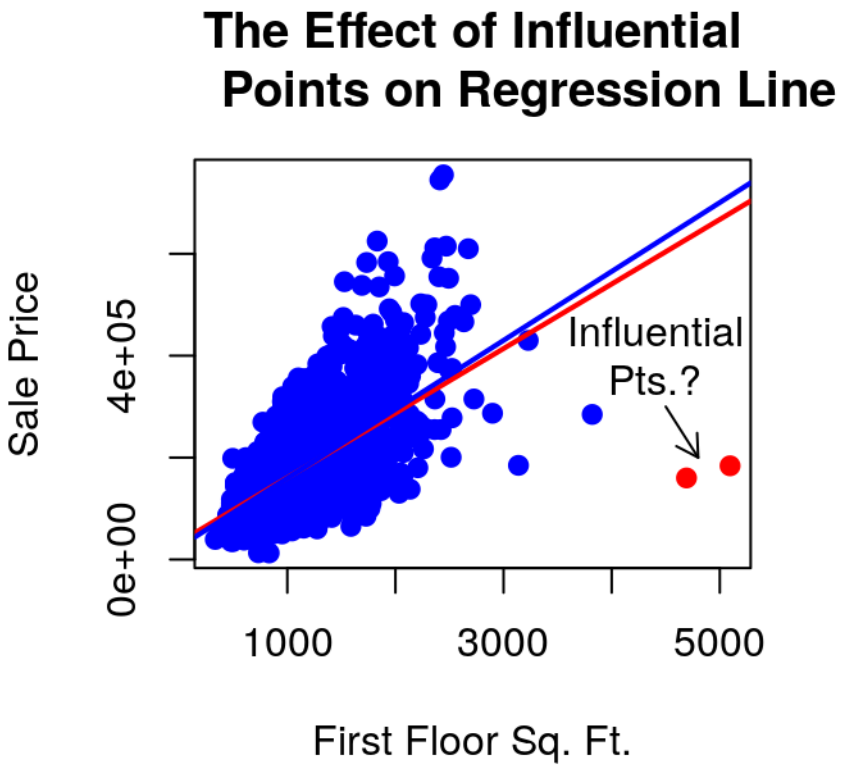
```
# Setup
ames.trim <- ames %>% filter(First_Flr_SF < 4000)
fit.full <- lm(Sale_Price ~ First_Flr_SF, ames)
fit.trim <- lm(Sale_Price ~ First_Flr_SF, ames.trim)


plot(x=ames$First_Flr_SF,
  y=ames$Sale_Price)
abline(fit.full)
abline(fit.trim)
text(label="Influential\nPts.?",
  x=4400, y=400000)
arrows(x0=4500, y0=300000,
  x1=4800, y1=200000)

# Other useful commands:
# points, lines, segments
```



Example: Options can fine tune the graph.

```
# Further Setup
inf.col <- if_else(ames$First_Flr_SF > 4000, "red", "blue")
```

```
plot(x=ames$First_Flr_SF,
  y=ames$Sale_Price,
  col=inf.col, pch=19,
  xlab="First Floor Sq. Ft.",
  ylab="Sale Price",
  main="The Effect of Influential
        Points on Regression Line")
abline(fit.full, col="red", lwd=2)
abline(fit.trim, col="blue", lwd=2)
text(label="Influential\nPts.?",
  x=4400, y=400000)
arrows(x0=4500, y0=300000,
  x1=4800, y1=200000, length=.1)
```



Two Base Graphics Cheat Sheets

- Try `?par`, `?plot`, `?hist`, etc. to be reminded of plot parameters.
- The internet is a good source—someone else has likely tackled your graphing task before.
- Below are links to two on-line cheat sheets.

(Gaston Sanchez) (Joyce Robbins)

In the *grammar of graphics*, every plot is built from several building blocks.

- The **data** to be used.
- **geom**etric objects that make up the graph.
- The **mapping** of variables to **aes**thetics of the graph.
- **stat**istical transformations of data for use by the geom.
- The **position** of elements in the graph.
- **facet**ing to create multiple graphs, changing one or more variables.
- **scale**s that define how values are mapped to an axis.
- Choice of **coordinate** system.
- Specification of **theme** to organize other aesthetic details.

Step 1: The data

```
library(tidyverse)
library(carData)
titanic <- TitanicSurvival
row.names(titanic) <- NULL
head(titanic)
```

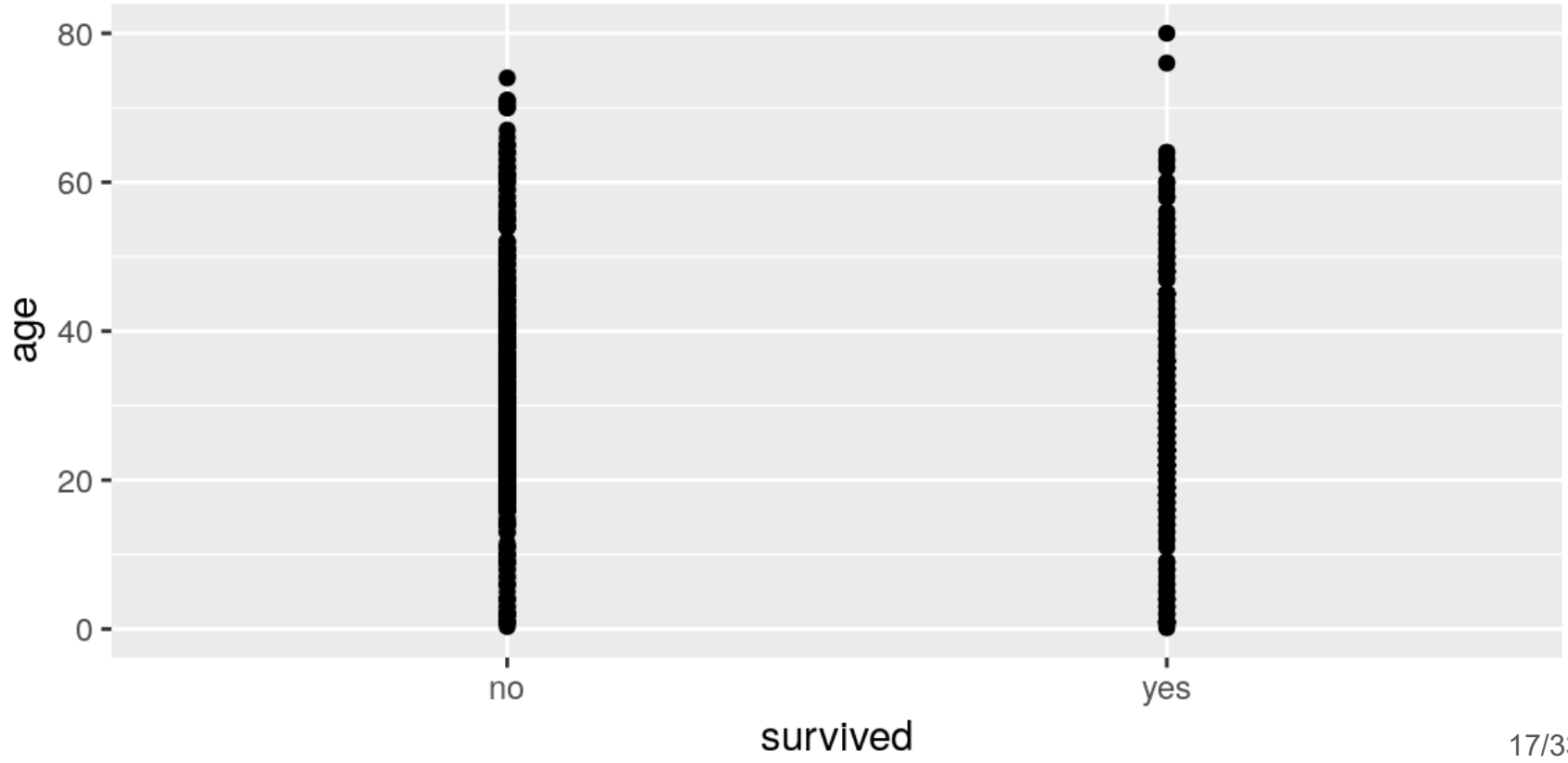
```
##   survived    sex     age passengerClass
## 1      yes female 29.0000            1st
## 2      yes   male  0.9167            1st
## 3       no female  2.0000            1st
## 4       no   male 30.0000            1st
## 5       no female 25.0000            1st
## 6      yes   male 48.0000            1st
```

```
ggplot(data = titanic)
```

Steps 2 and 3: The geom and the mapping

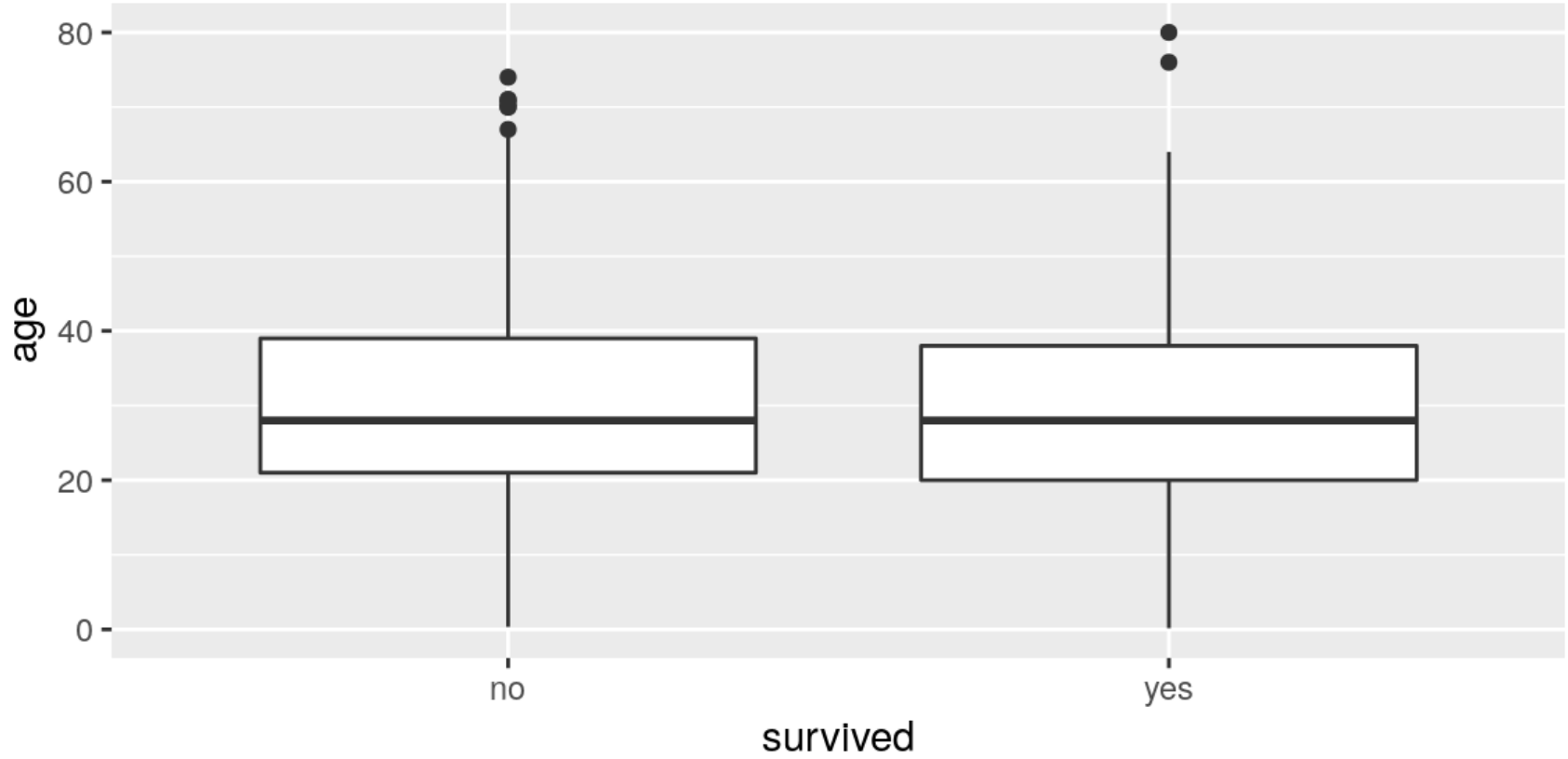
```
ggplot(data = titanic) +
  geom_point(mapping = aes(x=survived, y=age))
```

```
## Warning: Removed 263 rows containing missing values (geom_point).
```



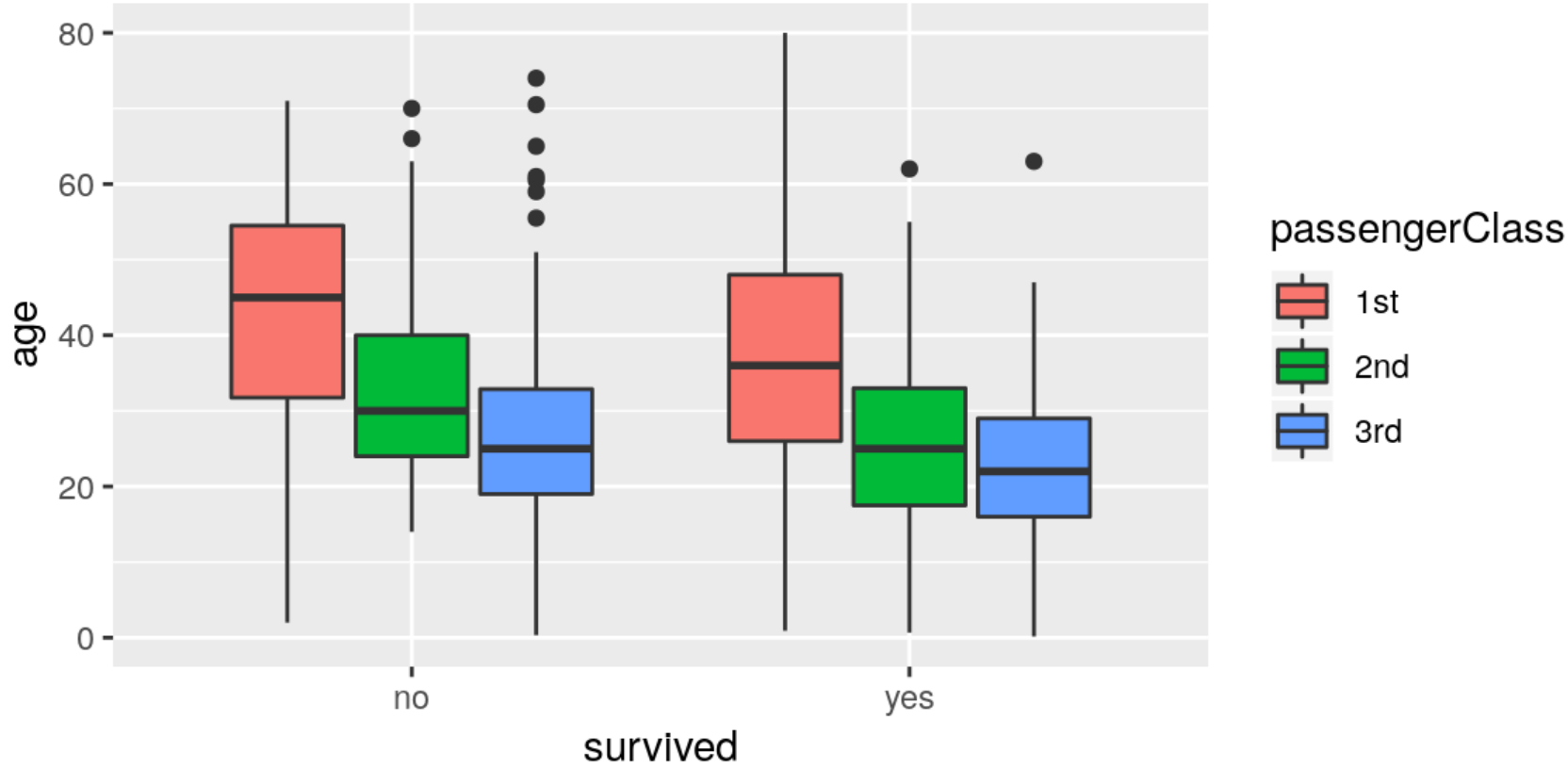
Steps 2 and 3: The geom and the mapping

```
ggplot(data = titanic) +
  geom_boxplot(mapping = aes(x=survived, y=age))
```



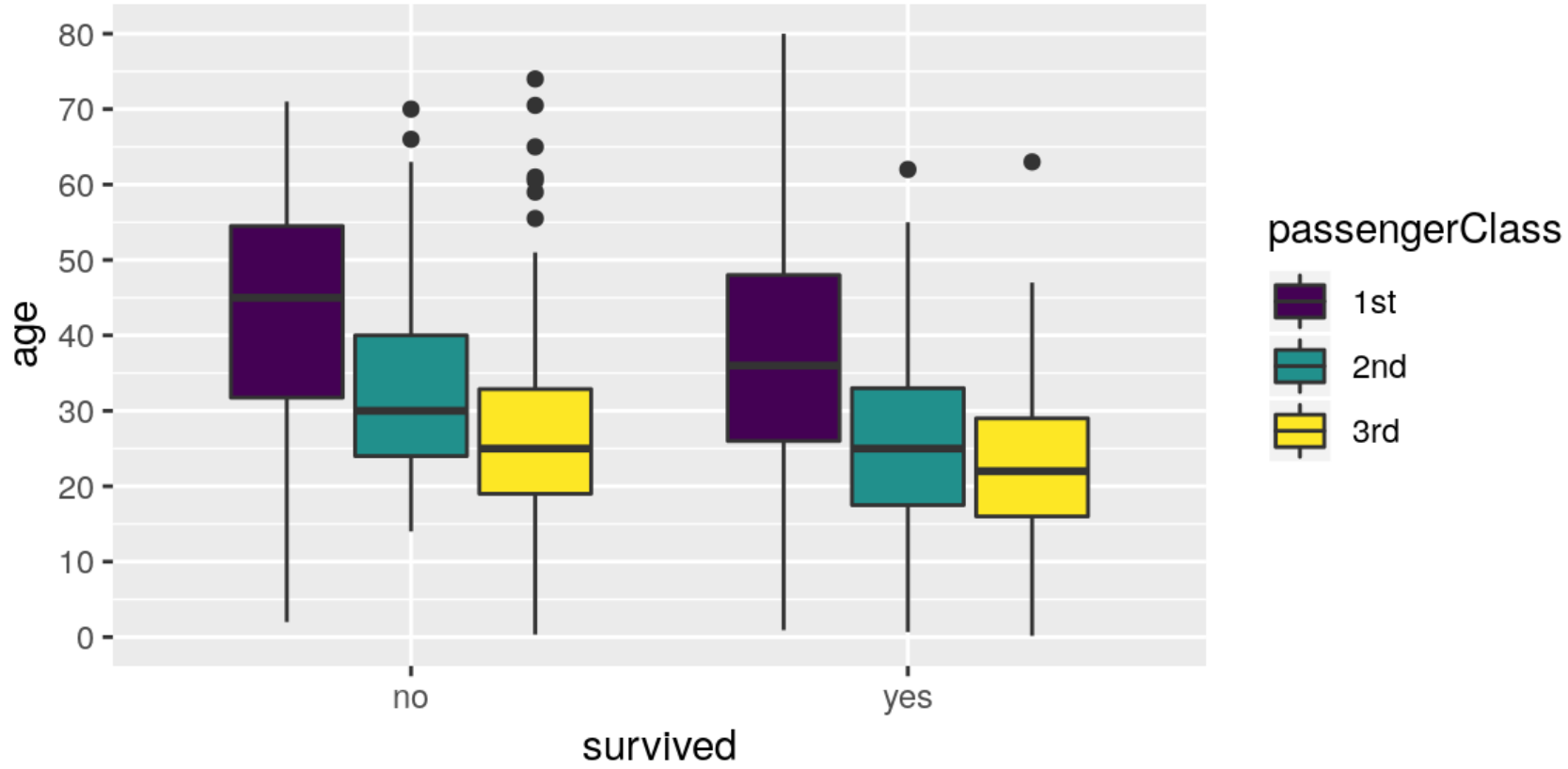
Step 3 (cont.): *Other* aesthetic mappings.

```
ggplot(data = titanic) +
  geom_boxplot(mapping = aes(x=survived, y=age, fill=passengerClass))
```



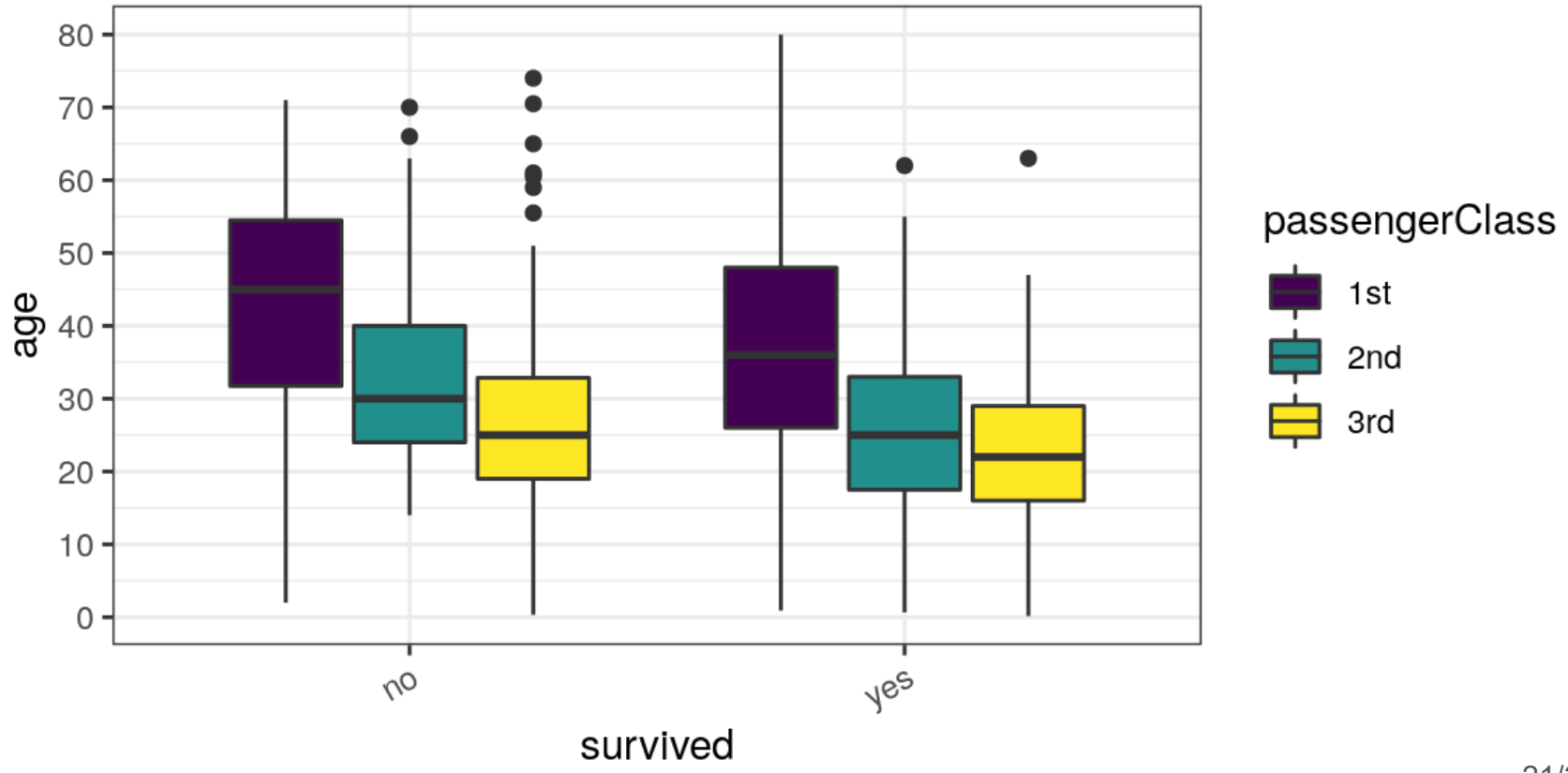
Scales control *how* data is mapped to aesthetics.

```
ggplot(data = titanic) +
  geom_boxplot(mapping = aes(x=survived, y=age, fill=passengerClass)) +
  scale_y_continuous(breaks=seq(0, 80, by=10)) + scale_fill_viridis_d()
```



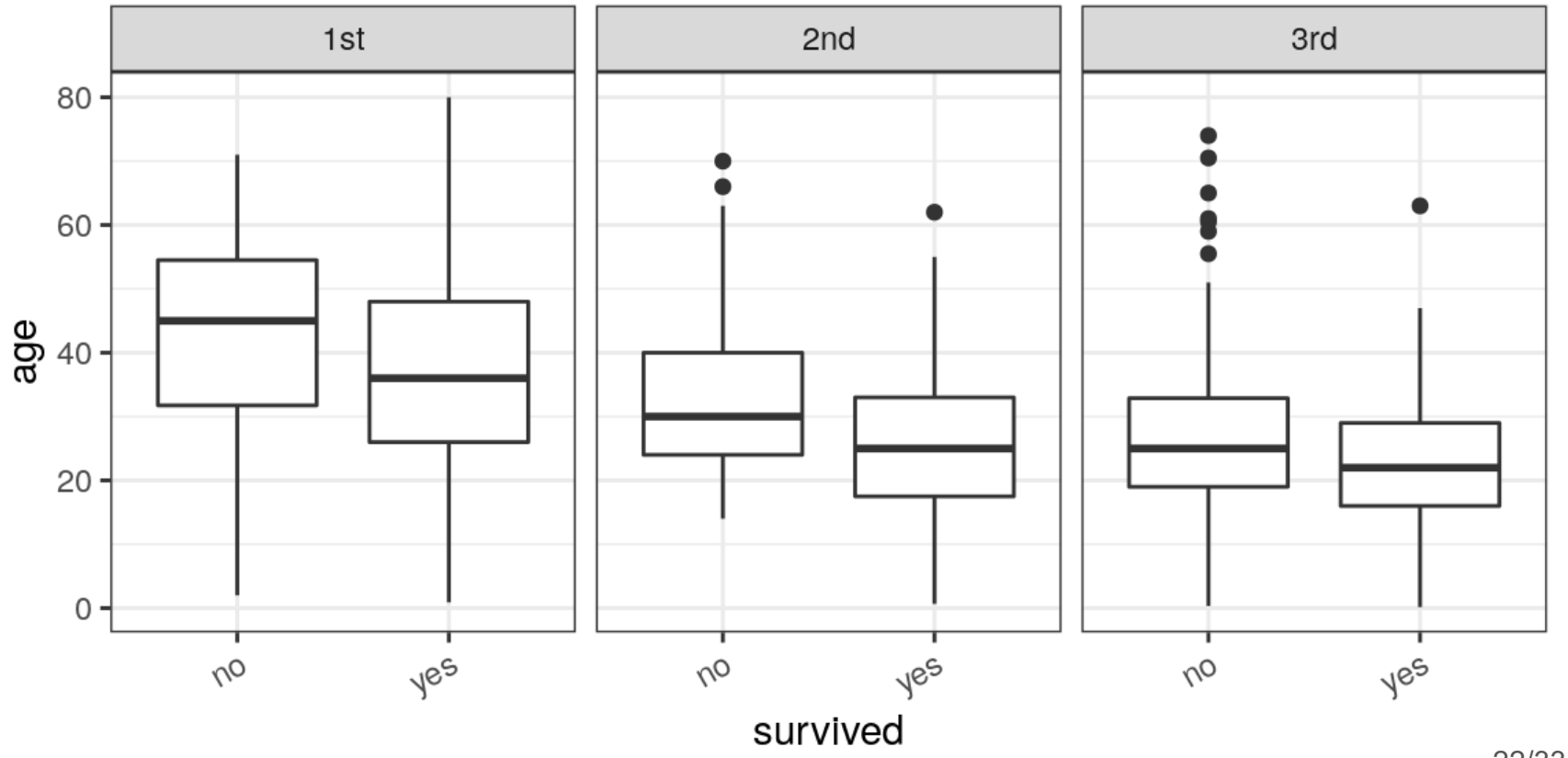
Themes control non-data aspects of the graph.

```
ggplot(data = titanic) +
  geom_boxplot(mapping = aes(x=survived, y=age, fill=passengerClass)) +
  scale_y_continuous(breaks=seq(0, 80, by=10)) + scale_fill_viridis_d() +
  theme_bw() + theme(axis.text.x = element_text(angle=30, hjust=1))
```



Facets view multivariate data in separate panels.

```
ggplot(data = titanic) +
  geom_boxplot(mapping = aes(x=survived, y=age)) +
  facet_wrap("passengerClass", ncol=3) +
  theme_bw() + theme(axis.text.x = element_text(angle=30, hjust=1))
```

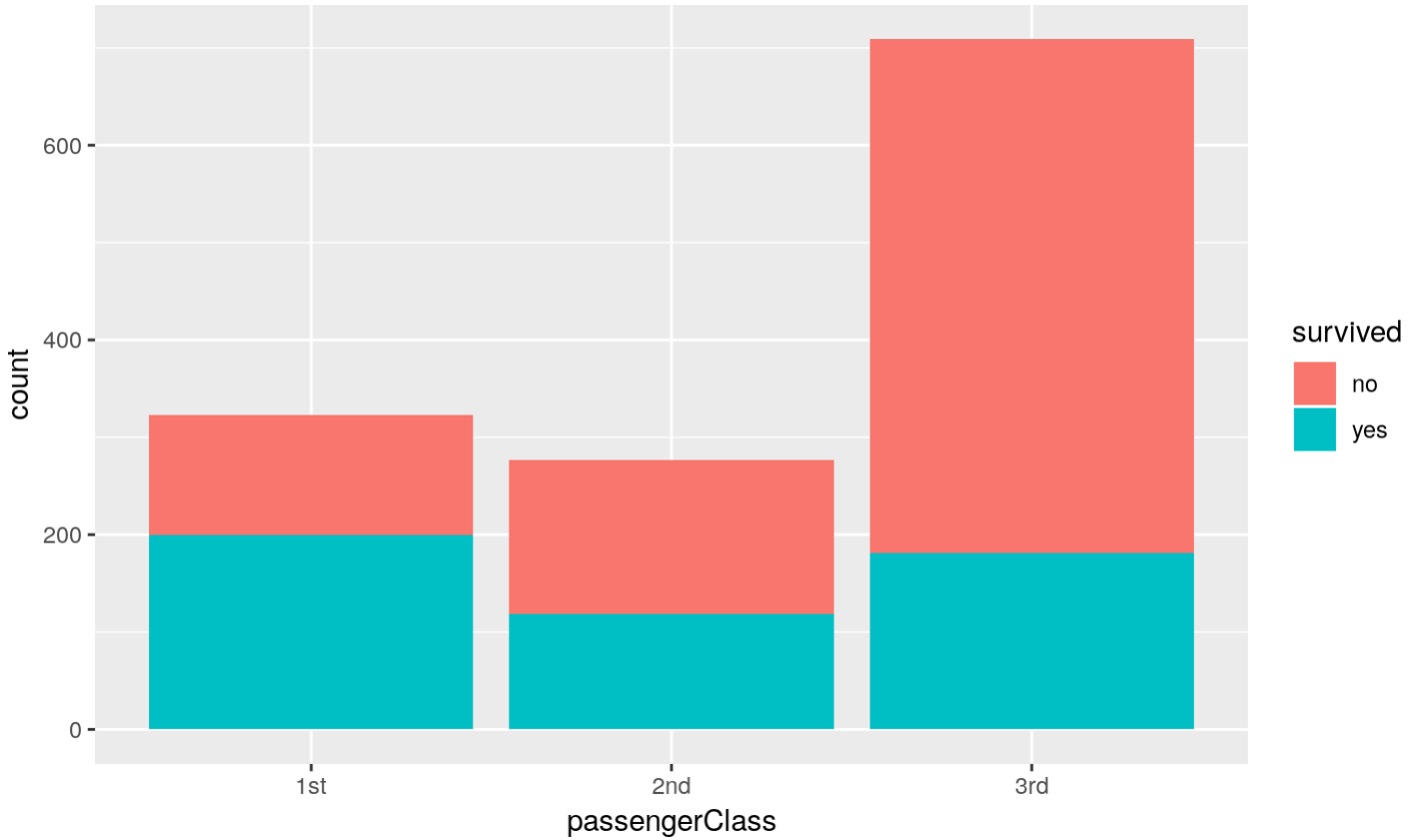


Every `geom` has a default `stat`, but you can use `stat`'s separately as well.

```
ggplot(data = titanic) +
  geom_boxplot(mapping = aes(x=survived, y=age)) +
  stat_summary(mapping = aes(x=survived, y=age), color="red",
    fun.y = mean, geom="point")
```

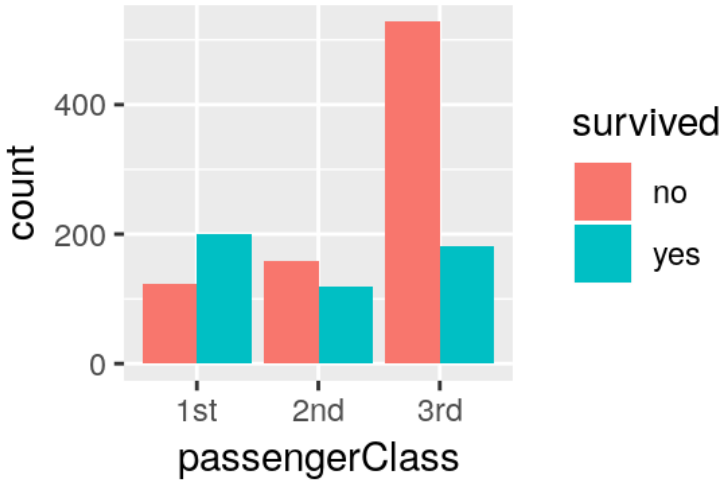
A bar graph can explain why the age pattern is only visible when passenger class is included.

```
ggplot(data = titanic) +
  geom_bar(mapping = aes(x=passengerClass, fill=survived))
```



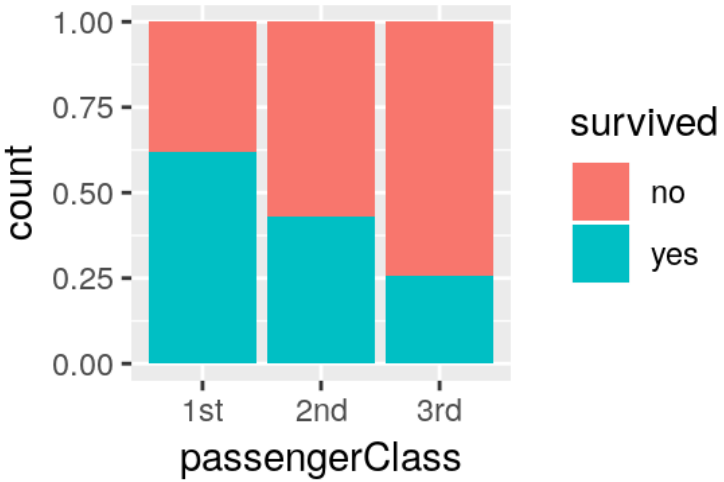
Changing the `position` of the bar segments gives better graphs.

```
ggplot(data = titanic) +
  geom_bar(mapping = aes(
             x=passengerClass,
             fill=survived),
           position="dodge")
```



Here we emphasize the raw counts in each category.

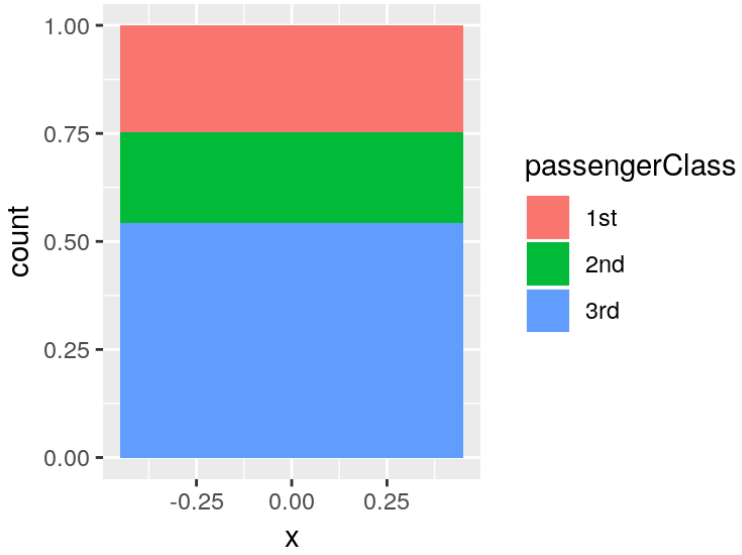
```
ggplot(data = titanic) +
  geom_bar(mapping = aes(
             x=passengerClass,
             fill=survived),
           position="fill")
```



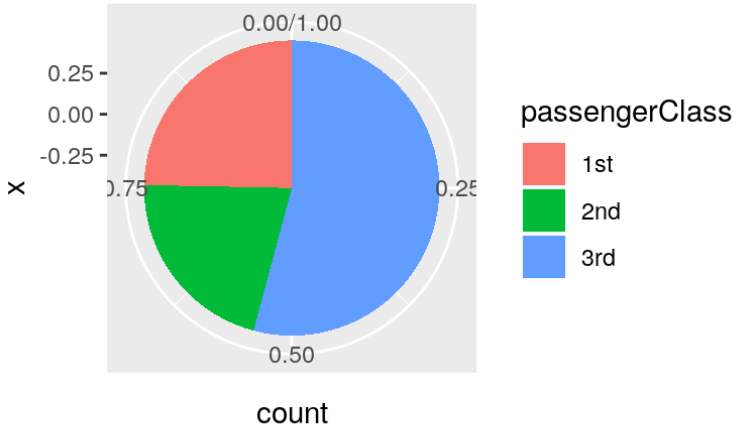
Most often you'll want to focus on percentages within each category.

A pie chart is just a bar graph in polar coordinates.

```
ggplot(data = titanic) +
  geom_bar(mapping = aes(x=0,
    fill=passengerClass),
    position="fill") +
  coord_cartesian()
```




```
ggplot(data = titanic) +
  geom_bar(mapping = aes(x=0,
    fill=passengerClass),
    position="fill") +
  coord_polar(theta="y")
```



Remember: Use with extreme caution. They made this hard on purpose!

More on Mappings

- Mappings specified in the `ggplot` command are inherited by the geoms.

```
ggplot(data = titanic, mapping = aes(x=survived, y=age)) +
  geom_boxplot(mapping = aes(fill=passengerClass))
```

is the same as

```
ggplot(data = titanic) +
  geom_boxplot(mapping = aes(x=survived, y=age, fill=passengerClass))
```

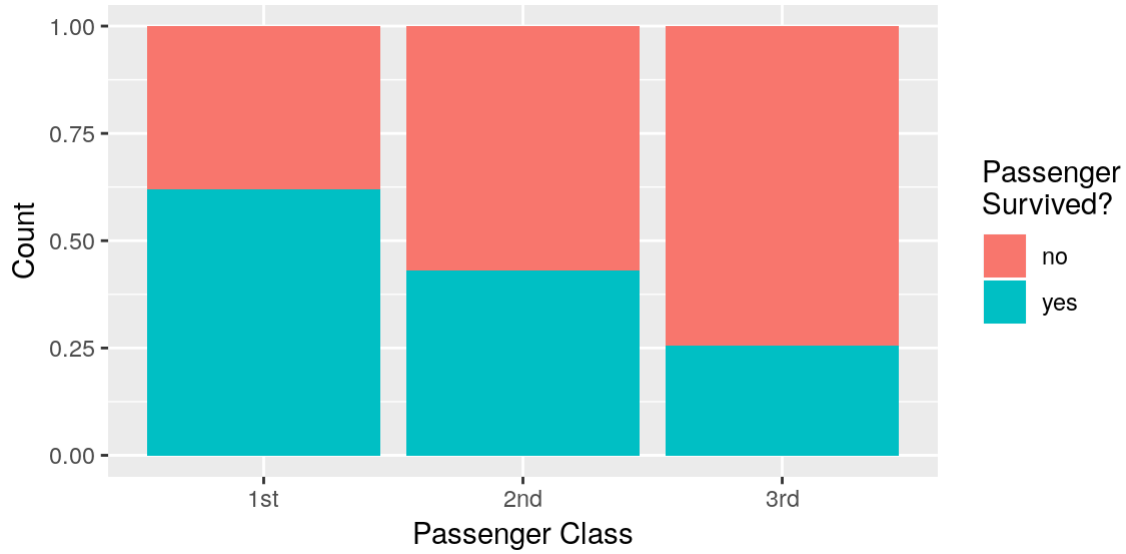
- Aesthetics specified outside of the `aes` are constants.

```
# This one makes red boxes.
ggplot(data = titanic) +
  geom_boxplot(mapping = aes(x=survived, y=age), fill="red")

# This one won't do what you want.
ggplot(data = titanic) +
  geom_boxplot(mapping = aes(x=survived, y=age), fill=passengerClass)
```

There are cross-cutting helper functions for common tasks.

- `xlab`, `ylab`, `labs` for labeling axes.
- `guides` for more easily adjusting legends.

```
ggplot(data = titanic) +
  geom_bar(mapping = aes( x=passengerClass, fill=survived), position="fill") +
  xlab("Passenger Class") + ylab("Count") +
  guides(fill = guide_legend("Passenger\nSurvived?"))
```



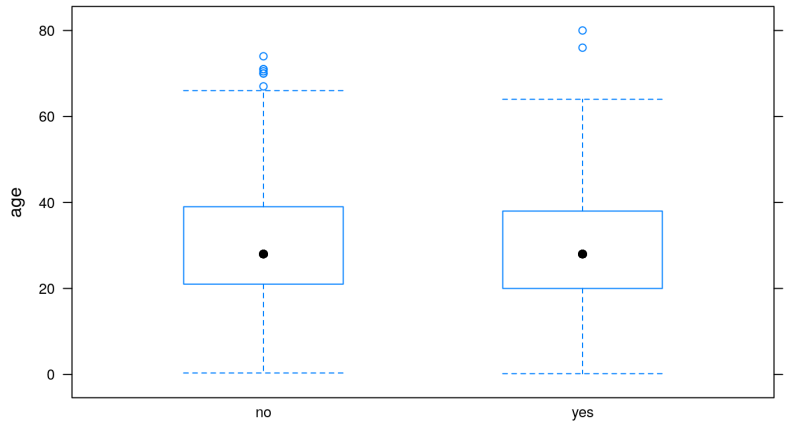
Patterns for Getting Help

- Read the book(s)!
  - ggplot2: Elegant Graphics for Data Analysis, Hadley Wickham
  - R for Data Science, Garrett Grolemund and Hadley Wickham
- Use built-in help
  - Think about whether it's a geom, stat, scale, theme question.
  - For example, `?geom`+TAB to be reminded of all geoms.
  - Or, `?theme` to find out that `axis.text.x` controls the *x* axis and that `element_text()` is the actual function that sets attributions. Then `?element_text()` to find its options.
- Google. Someone has almost surely already wanted to do what you want to do.

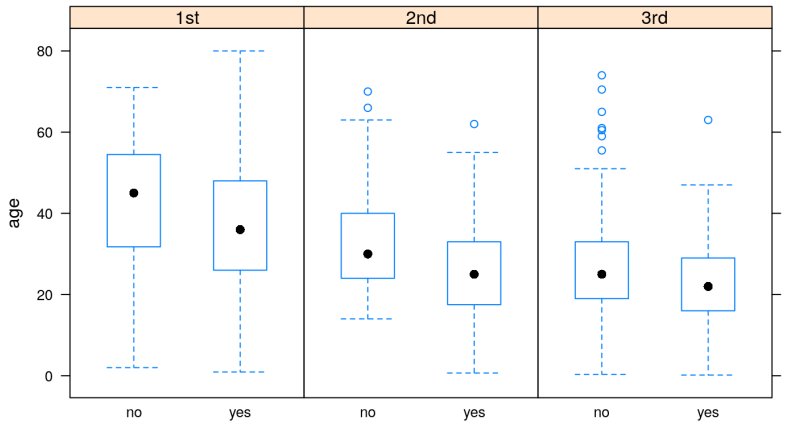
Get the `ggplot` cheat sheet from R Studio!

The `lattice` package has graph commands for each plot type, using formulas to generate multiple plots.

```
library(lattice)
bwplot(age~survived, data=titanic)
```




```
bwplot(age~survived|passengerClass,
  data=titanic)
```



Some lattice commands and base R equivalents.

Lattice Function	Description	Formula Example(s)	Base Analog
barchart()	bar graph	x ~ A or A ~ x	barplot()
bwplot ()	box plot	x ~ A or A ~ x	boxplot()
densityplot()	kernal density plot	~ x | A * B	plot.density()
dotplot()	dot plot	~x | A	dotchart()
histogram()	histogram	~ x	hist()
stripplot()	strip plots	A ~ x or x ~ A	stripchart()
xyplot()	scatter plot	y ~ x | A	plot()
contourplot()	3D contour plot	z ~ x * y	contour()
cloud()	3D scatter plot	z ~ x * y | A	NA
levelplot()	3D level plot	z ~ y * x	image()
parallel()	parallel coordinates plot	data frame	NA
splom()	scatter plot matrix	data frame	pairs()
wireframe()	3D surface graph	z ~ y * x	persp()

An exercise will have them do this to answer why average condition costs more. (This isn't part of the lecture!)