Relationships

Week 10

AEM 2850 / 5850 : R for Business Analytics
Cornell Dyson
Spring 2024

Acknowledgements: Andrew Heiss

Announcements

Reminders:

- Group project due April 19 (link)
  - We set up group-specific workspaces on Posit Cloud for the project to allow simultaneous collaborative editing
  - Instructions are posted there and on canvas
  - Make a plan and start early!
- Victor will help you work through this Thursday's example
- No lab-10 due to spring break

Questions before we get started?

Plan for today

Prologue: The dangers of dual y-axes

Visualizing relationships between a numerical and a categorical variable

Visualizing relationships between two numerical variables

- Visualizing correlations
- Visualizing regressions

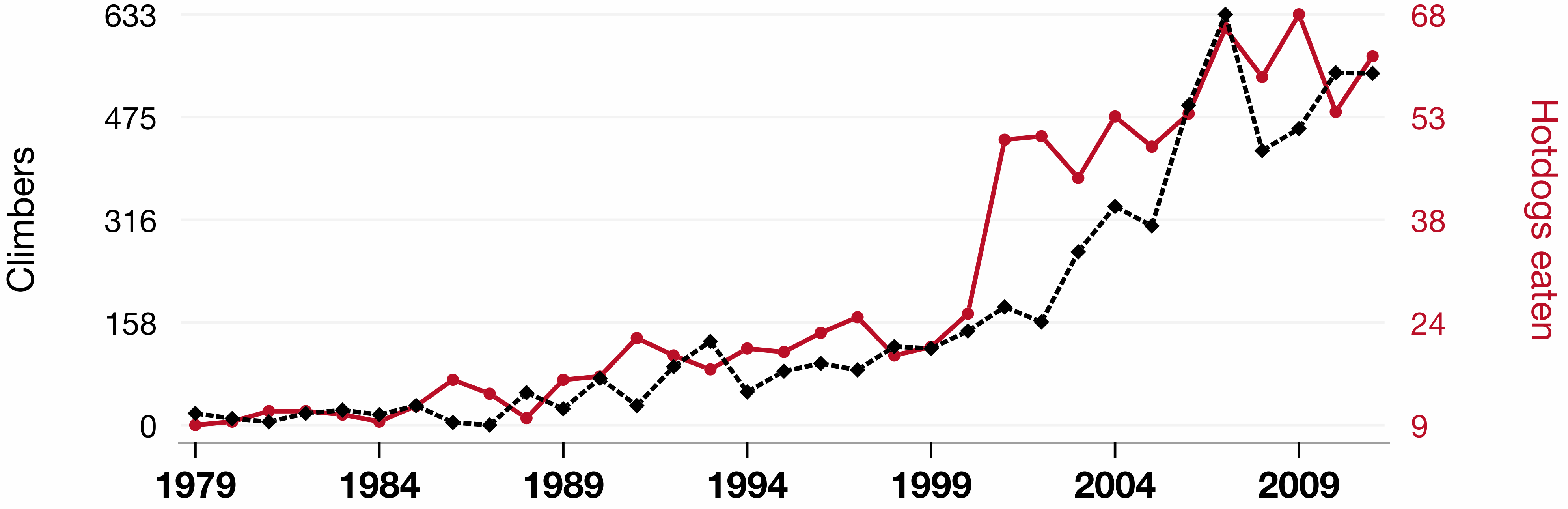
Prologue: The dangers of dual y-axes

Oh no!

Total Number of Successful Mount Everest Climbs

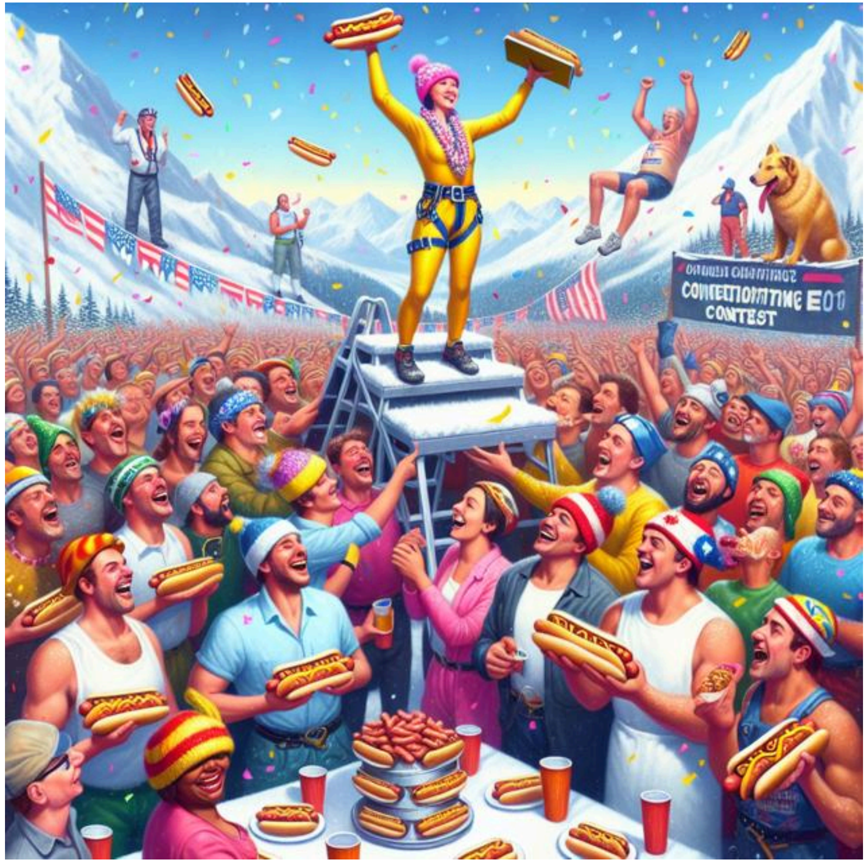
correlates with

Hotdogs consumed by Nathan's Hot Dog Eating Competition Champion



GPT 3.5 and DALL·E 3 explainer

"As the number of successful Mount Everest climbs rises, so does the peak appetite for adventure. This, in turn, creates a sausage-yetis-faction where competitors are relishing the thrill of the challenge like never before, and they're on a roll to claim the title. It's a summit showdown of epic proportions, where each contender is truly reaching their peak performance..."

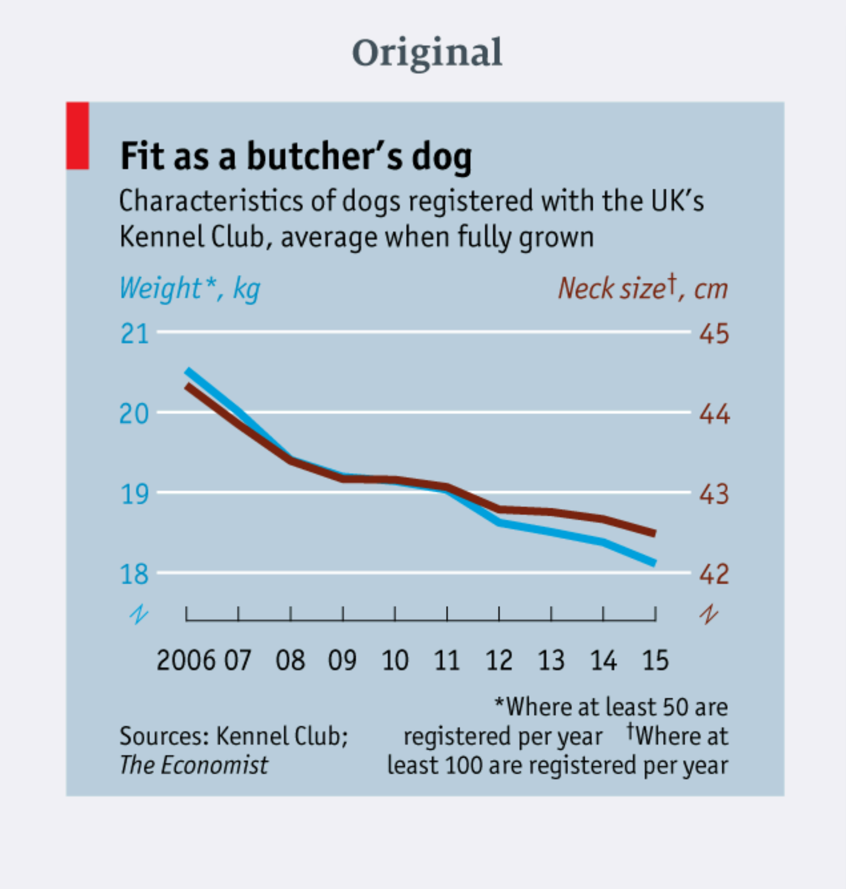


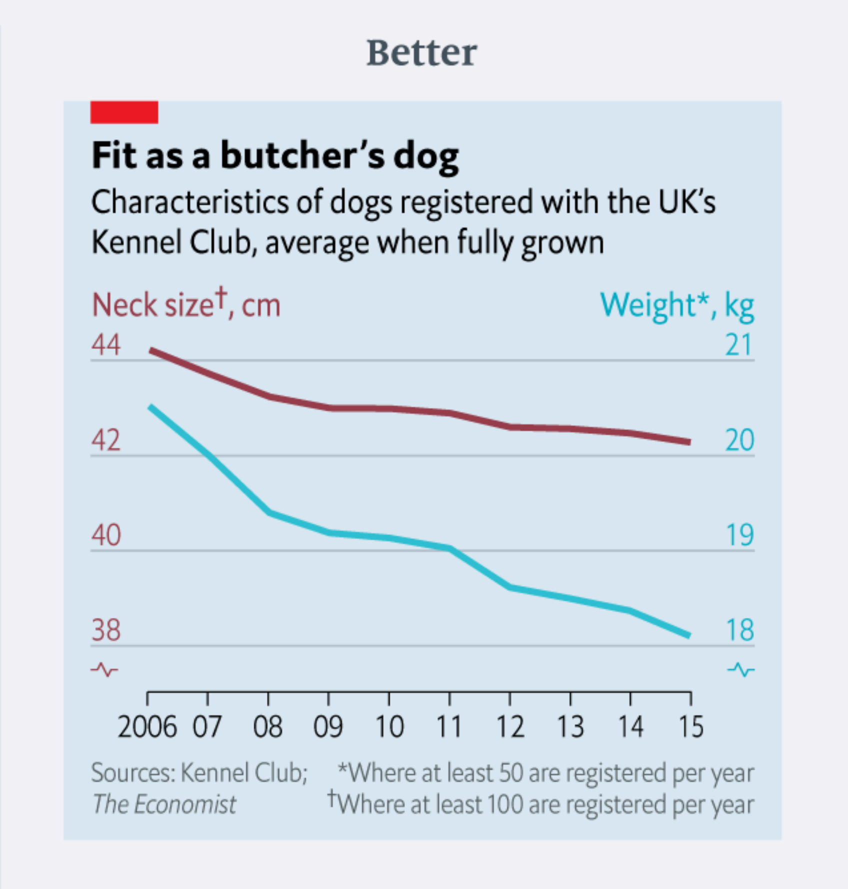
Why not use two y-axes?

You have to choose where the y-axes start and stop, which means…

…you can force the two trends to line up however you want!

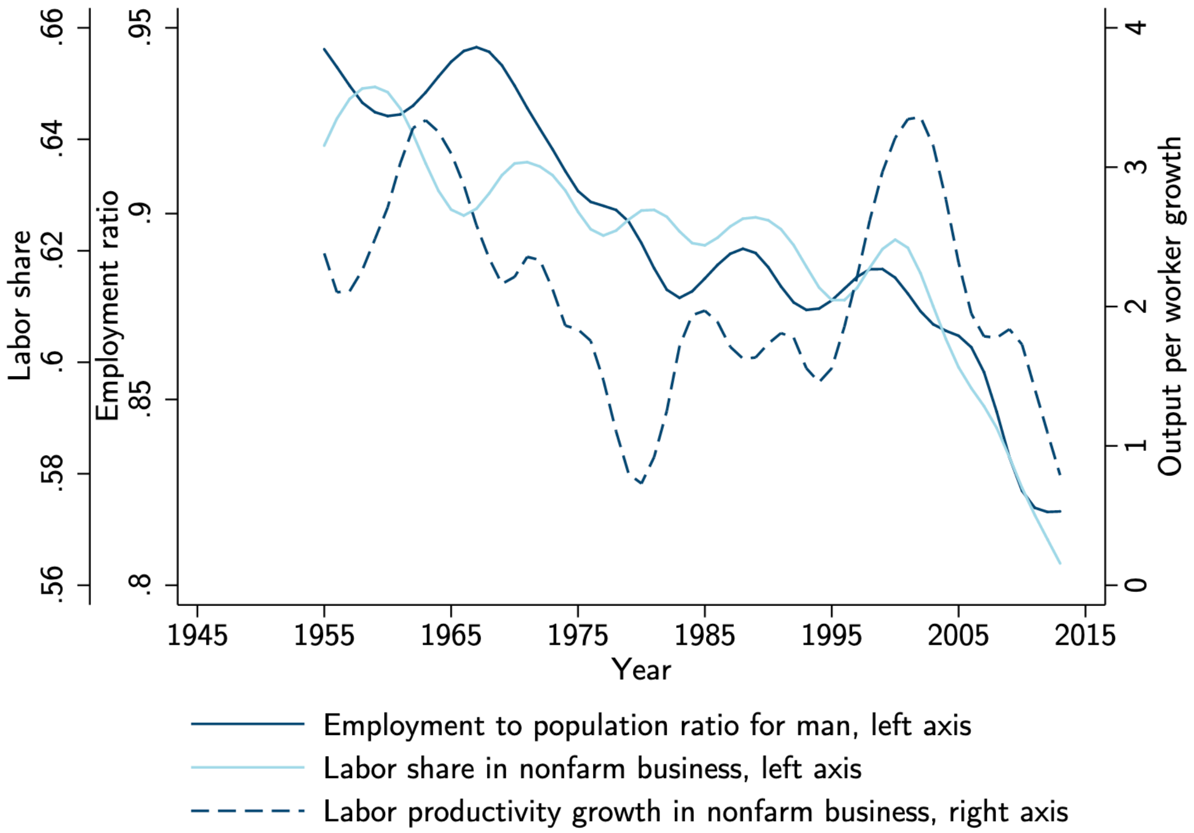
It even happens in *The Economist*!





The revised axes ranges reflect a comparable proportional change

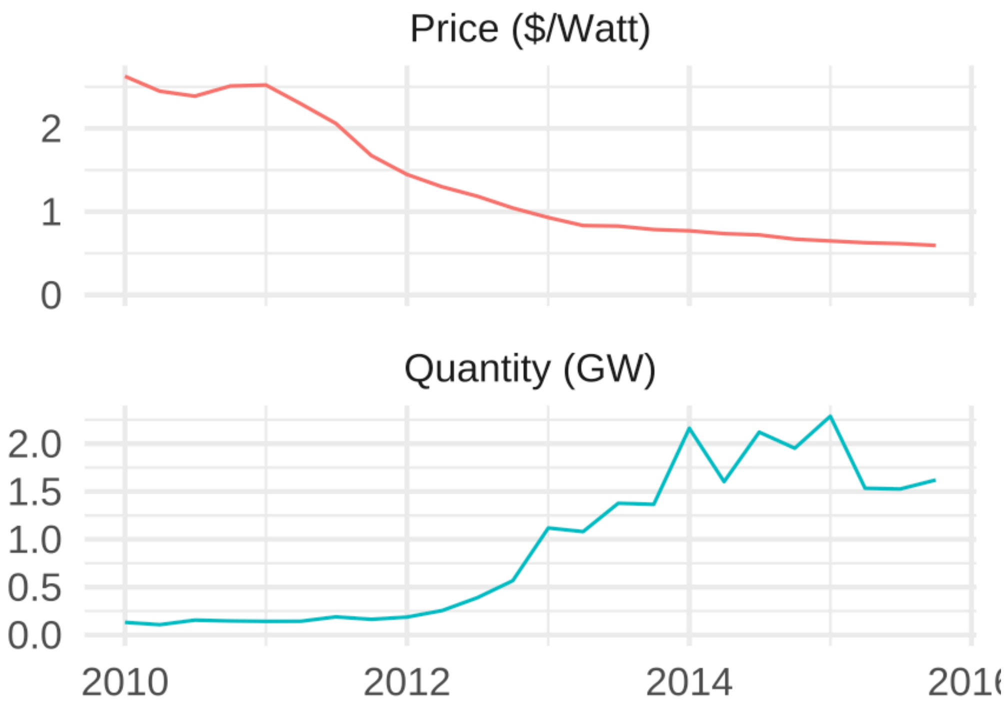
The rare triple y-axis



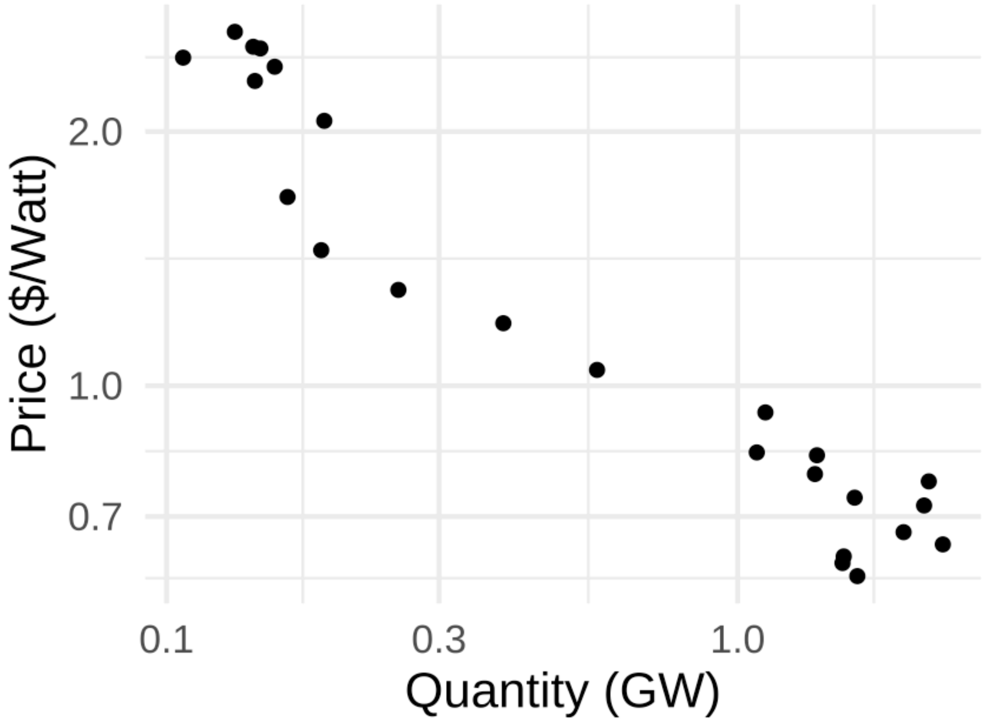
Source: Daron Acemoglu and Pascual Restrepo, "The Race Between Man and Machine: Implications of Technology for Growth, Factor Shares and Employment"

What could we do instead?

- Use multiple plots!



- Use scatter plots instead



How could we make multiple plots in R?

1. Facets are great when using a common geometry (we've already seen that)

2. Combining multiple plot objects can be more flexible

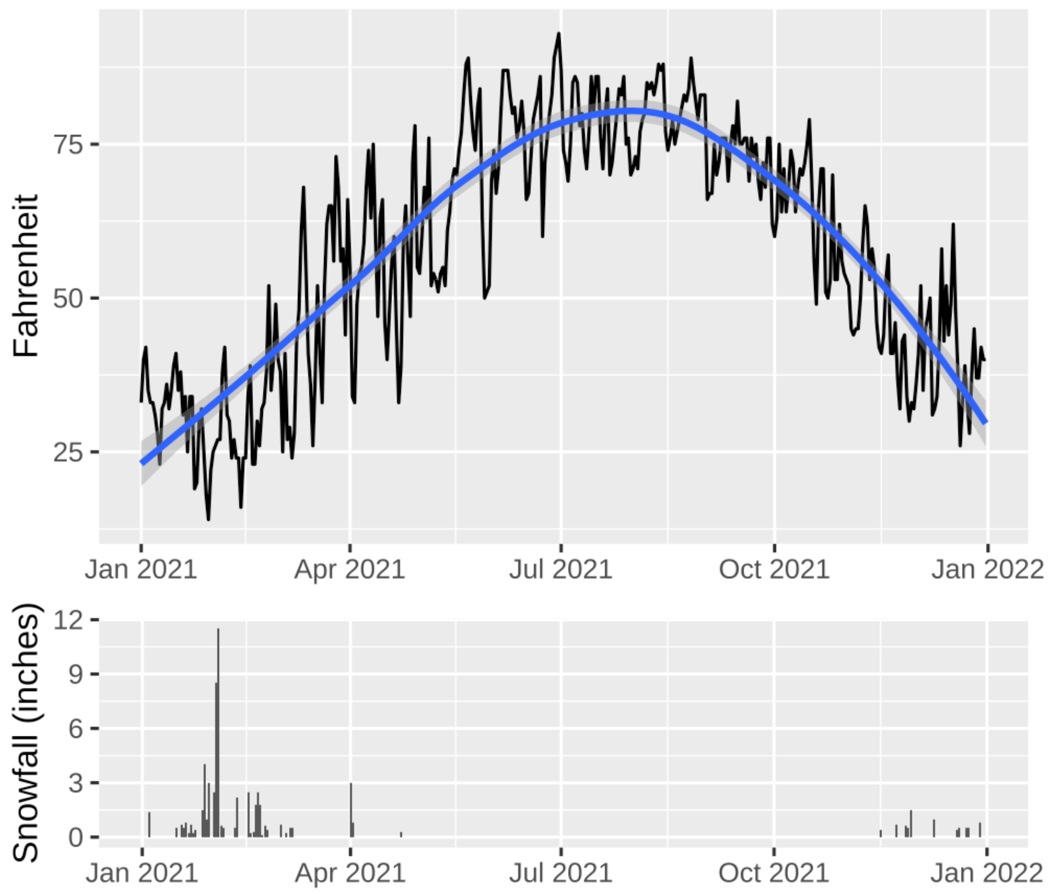
Let's use Ithaca weather data to see an example of combining plots:

```
ithaca_weather <- read_csv("data/ithaca-weather-2021.csv")

ithaca_weather |>
  select(STATION, NAME, DATE, TMAX, SNOW) |>
  head(3)
```

```
## # A tibble: 3 × 5
##   STATION     NAME                            DATE        TMAX  SNOW
##   <chr>       <chr>                           <date>     <dbl> <dbl>
## 1 USC00304174 ITHACA CORNELL UNIVERSITY, NY US 2021-01-01    33     0
## 2 USC00304174 ITHACA CORNELL UNIVERSITY, NY US 2021-01-02    40     0
## 3 USC00304174 ITHACA CORNELL UNIVERSITY, NY US 2021-01-03    42     0
```

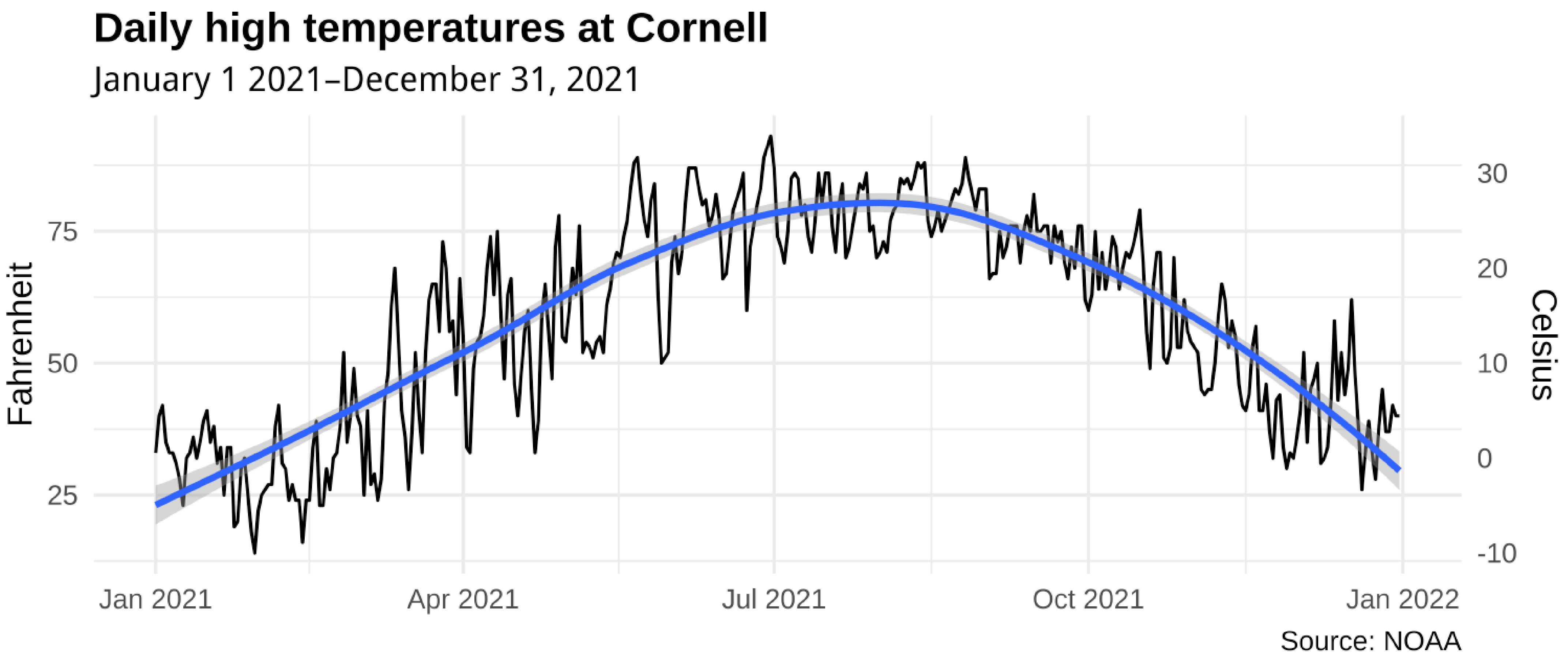
Combining multiple plots in R

```
library(patchwork)

# make a plot of temperatures
temp_plot <- ggplot(ithaca_weather,
                    aes(x = DATE, y = TMAX)) +
  geom_line() + geom_smooth() +
  labs(x = NULL, y = "Fahrenheit")

# make a plot of snowfall
snow_plot <- ggplot(ithaca_weather,
                    aes(x = DATE, y = SNOW)) +
  geom_col() +
  labs(x = NULL, y = "Snowfall (inches)")

# use patchwork to combine the two plots
temp_plot +     # simply use + to combine plots
  snow_plot +   # then add on custom options
  plot_layout(ncol = 1,
              heights = c(0.7, 0.3))
```



When are dual y-axes defensible?

When the two axes measure the same thing

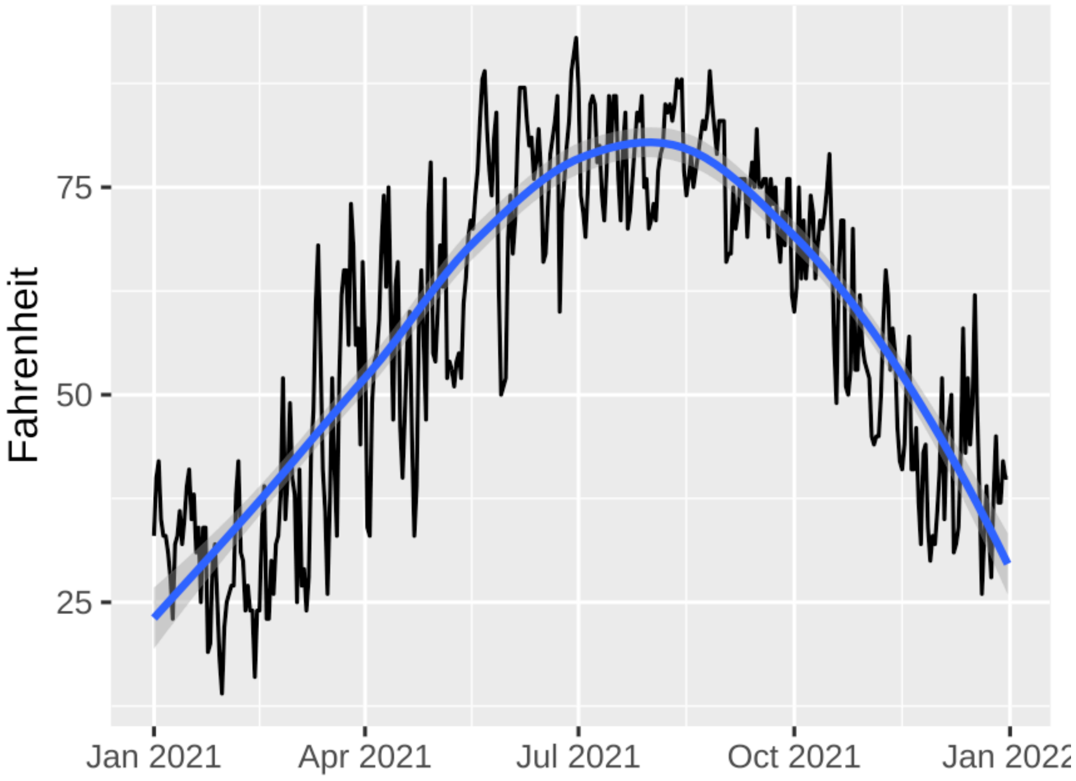


Making the base plot in R

```
ggplot(ithaca_weather,
       aes(x = DATE, y = TMAX)) +
  geom_line() +
  geom_smooth() +
  labs(x = NULL, y = "Fahrenheit")
```

How could we add a second axis?

Do any functions come to mind?

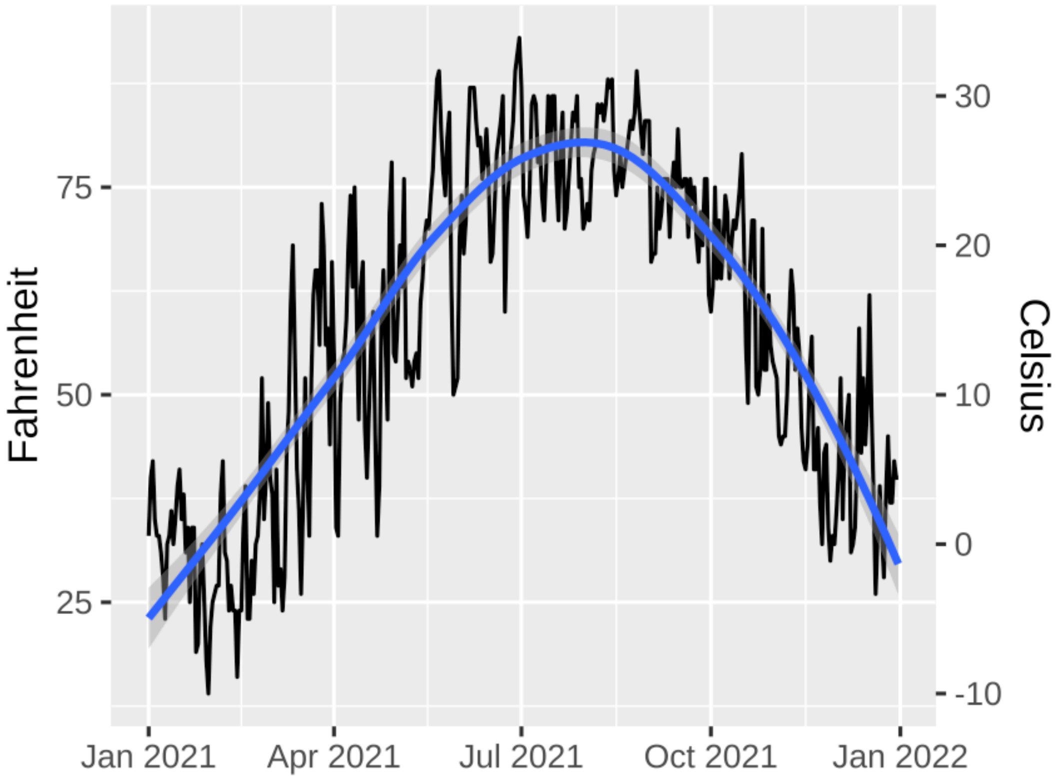


Adding a second scale in R

```
ggplot(ithaca_weather,
       aes(x = DATE, y = TMAX)) +
  geom_line() +
  geom_smooth() +
  scale_y_continuous(
    sec.axis =
      sec_axis(trans = ~ (. - 32) * 5/9,
               name = "Celsius")
  ) +
  labs(x = NULL, y = "Fahrenheit")
```

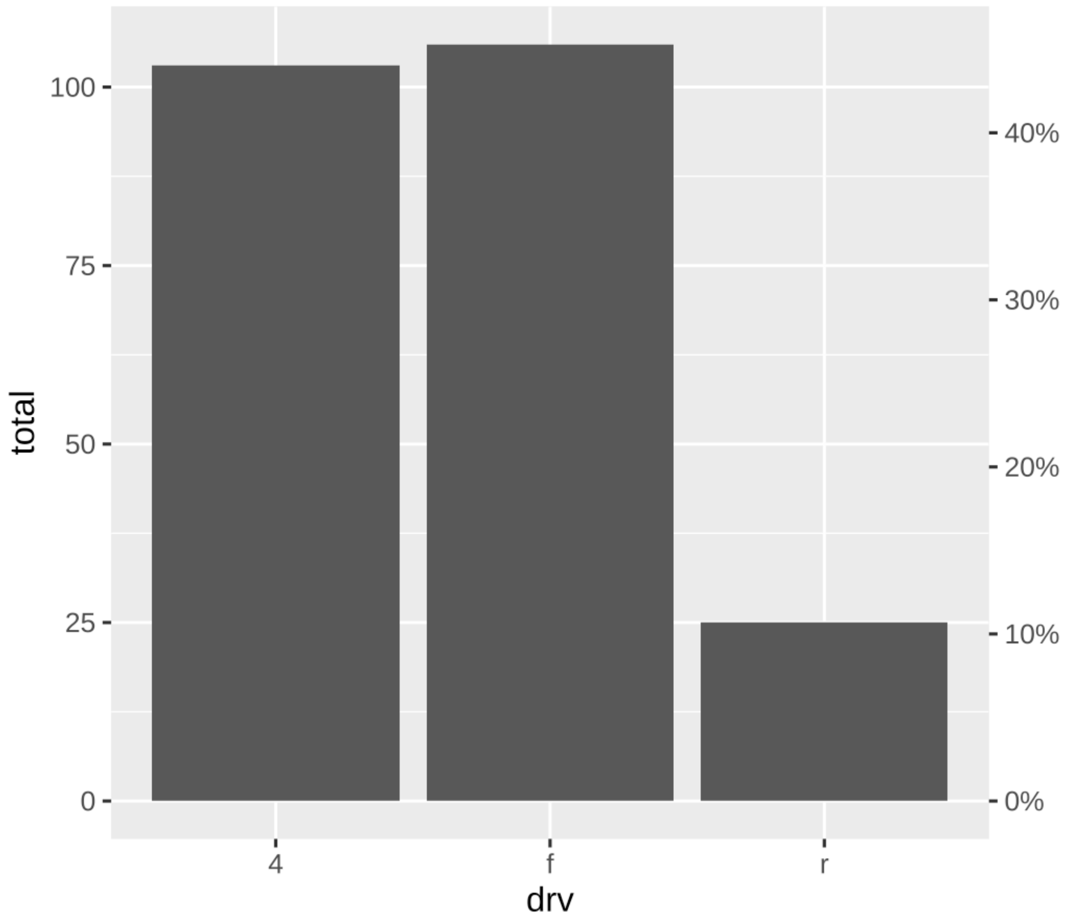
We provided this formula for the **trans**formation argument:

```
Celsius = (Fahrenheit - 32) * 5/9
```



Adding a second scale in R

```
car_counts <- mpg |>
  group_by(drv) |> summarize(total = n())

total_cars <- sum(car_counts$total)

car_counts |>
  ggplot(aes(x = drv, y = total)) +
  geom_col() +
  scale_y_continuous(
    sec.axis = sec_axis(
      trans = ~ . / total_cars,
      labels = scales::percent)) +
  guides(fill = "none")
```

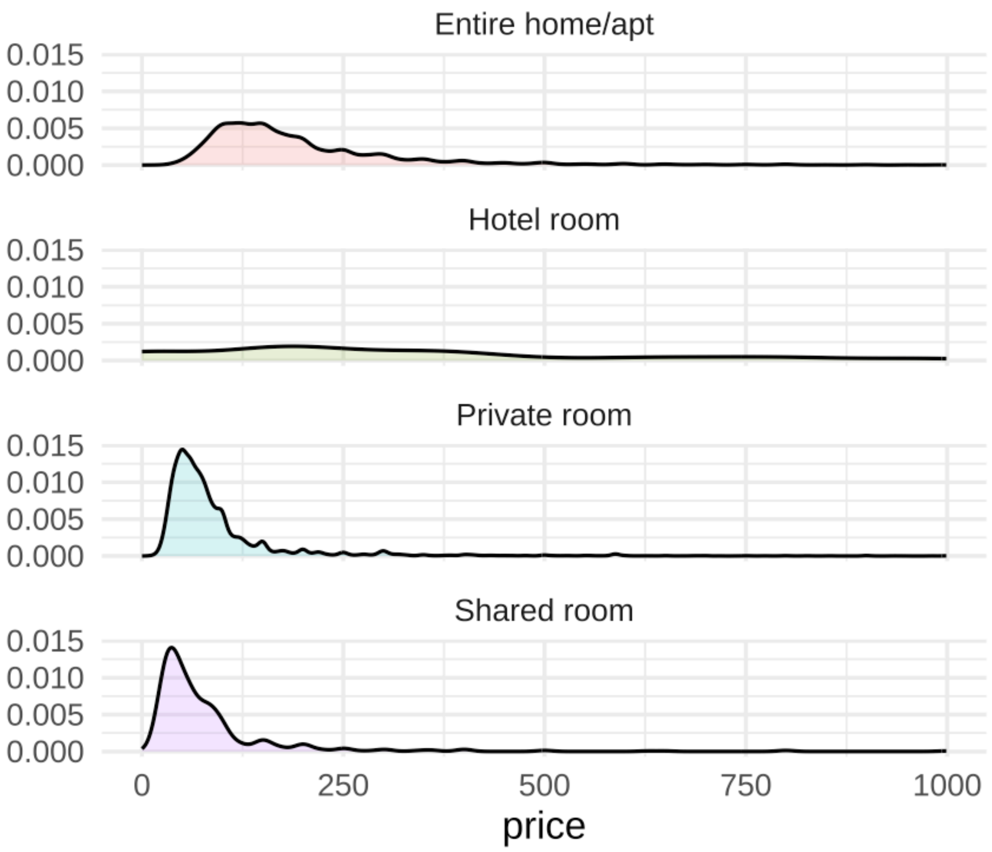


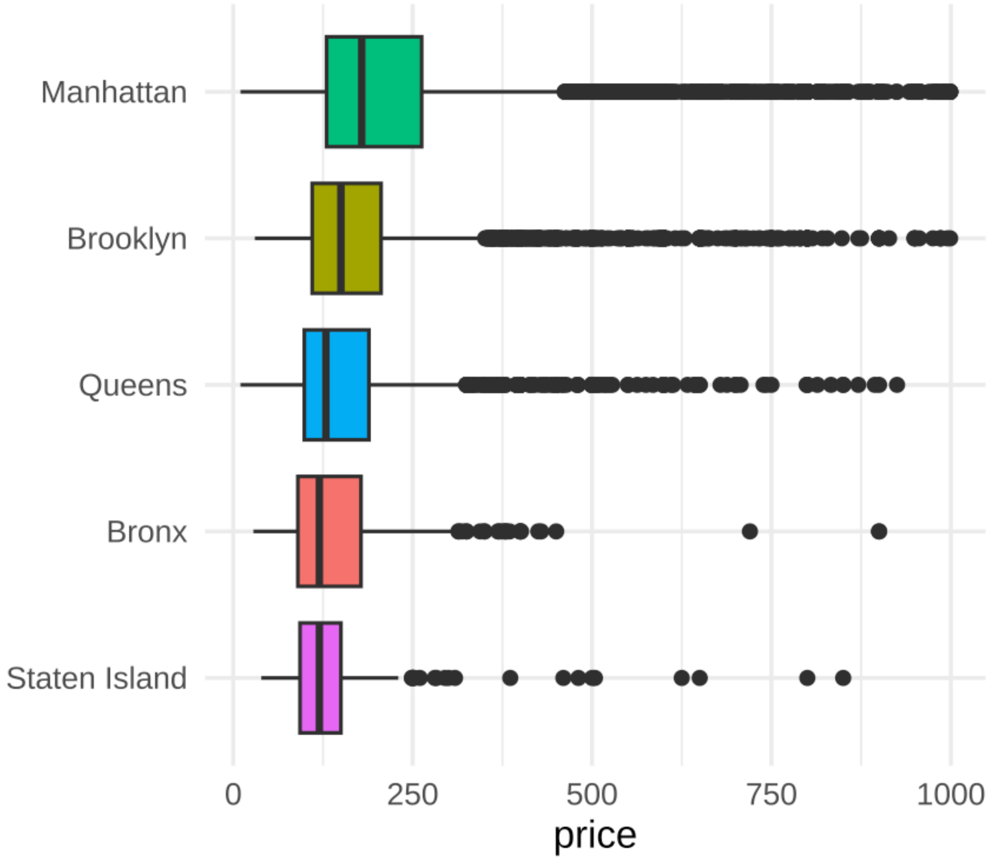
This makes it a lot easier to see proportions with side-by-side bars!

Note: **total_cars** is not in **car_counts**

Visualizing relationships between a numerical and a categorical variable

We already did this! When?





Visualizing relationships between two numerical variables

Visualizing correlations

What does "correlation" mean to you?

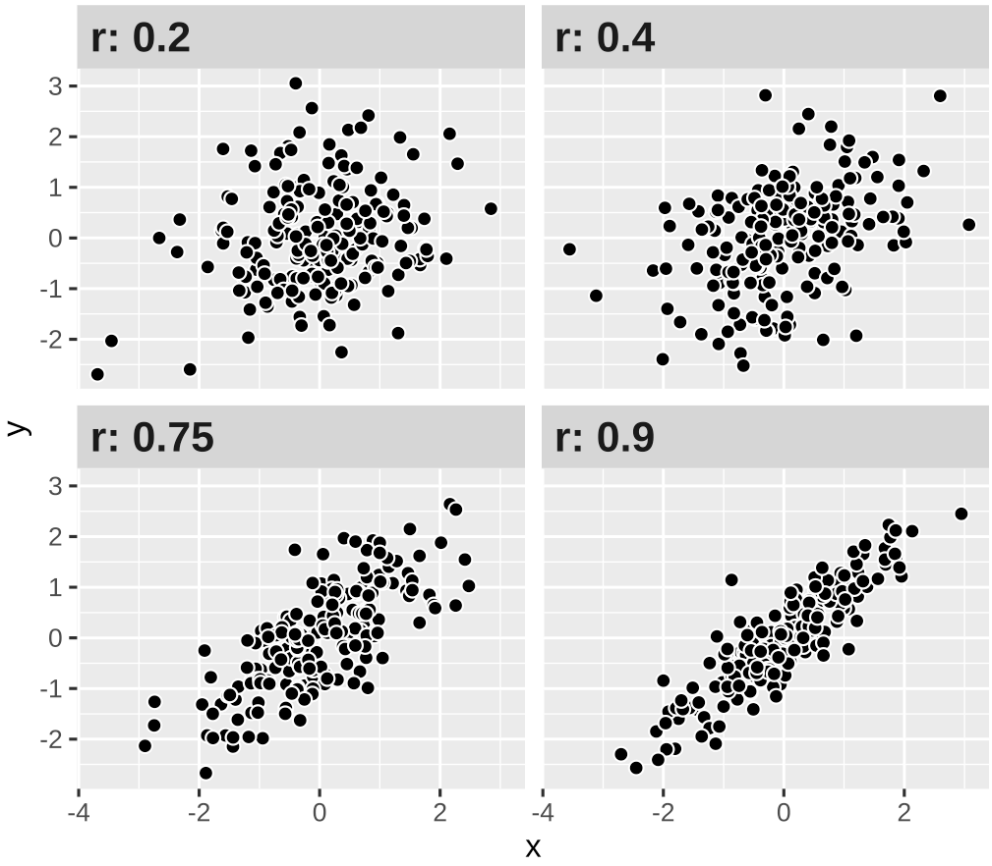
As the value of X goes up, Y is very / a little / not at all likely to go up (down)

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}$$

Says nothing about *how much* Y changes when X changes

Correlation values

ρ	**Rough meaning**
±0.1–0.3	Weak
±0.3–0.5	Moderate
±0.5–0.8	Strong
±0.8–0.9	Very strong



Scatter plots

The humble scatter plot is often the best place to start when studying the association between two variables

Example: max and min temperature in Ithaca each day of the year

- Do you think they are highly correlated, somewhat correlated, or not at all correlated?
- What sign do you think this correlation has?
- How would you make a scatter plot of these data in R?

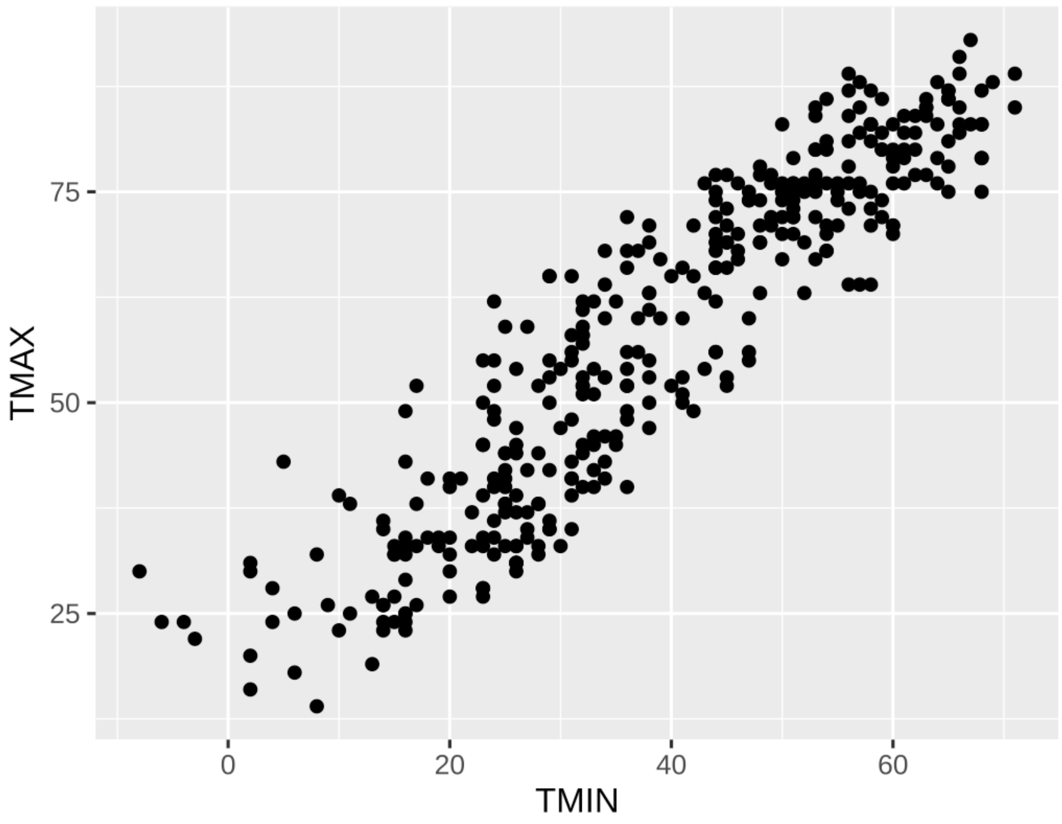
Scatter plots

```r
ithaca_weather |>
  ggplot(aes(x = TMIN, y = TMAX)) +
  geom_point()
```

```r
cor(ithaca_weather$TMIN,
    ithaca_weather$TMAX) |>
  round(2)
```

```
## [1] 0.92
```

Strong positive correlation



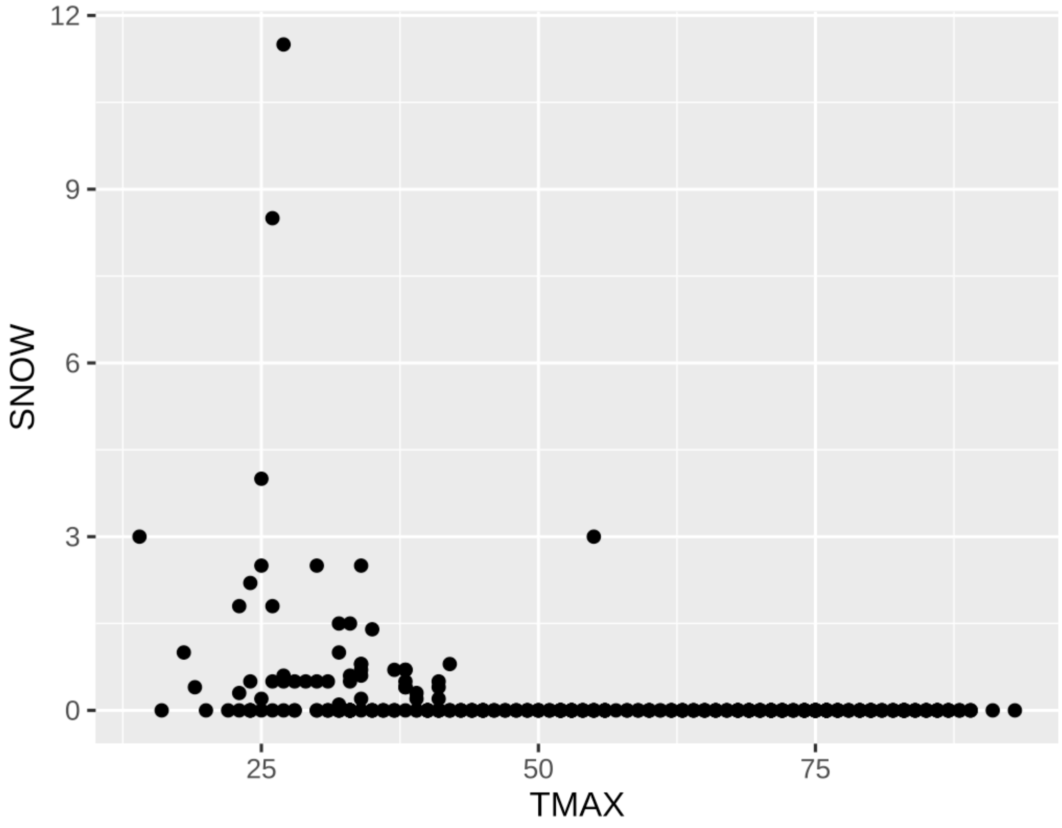
What about max temp and snowfall?

```
ithaca_weather |>
  ggplot(aes(x = TMAX, y = SNOW)) +
  geom_point()
```

```
cor(ithaca_weather$TMIN,
    ithaca_weather$SNOW) |>
  round(2)
```

```
## [1] -0.24
```

Weak negative correlation



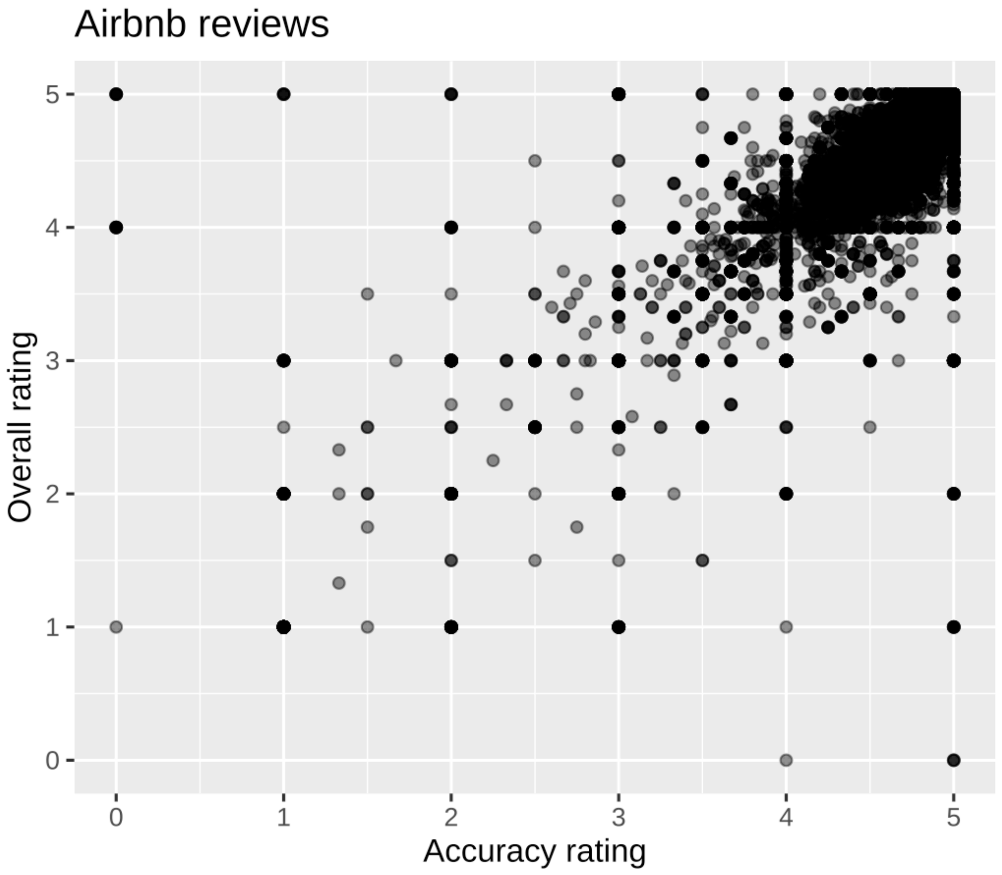
Visualizing regressions

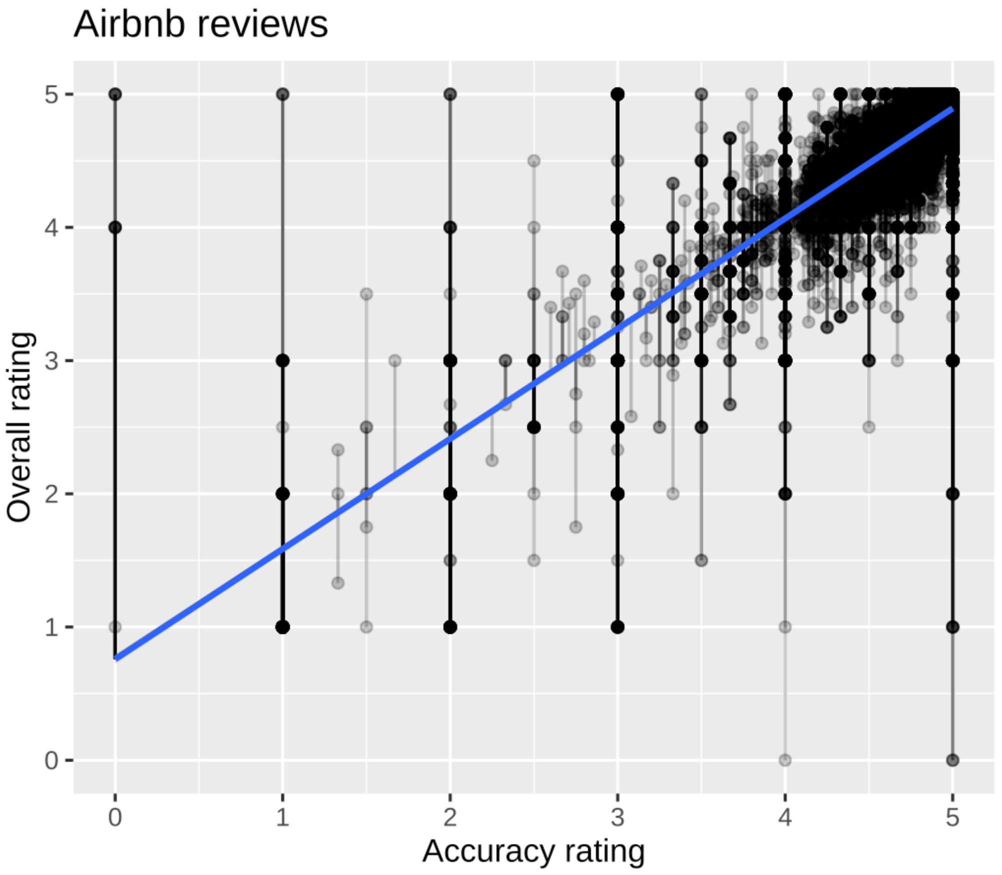
Linear regression reminder

$$y = \beta_0 + \beta_1 x_1 + \varepsilon$$

y	Outcome variable (DV)
x_1	Explanatory variable (IV)
β_1	Slope
β_0	y-intercept
ε	Error (residuals)

Linear regression is just drawing lines





Building models in R

Base R has some basic modeling tools:

```
name_of_model <- lm(<Y> ~ <X>, data = <DATA>) # use lm to fit simple linear models

summary(name_of_model) # see model details
```

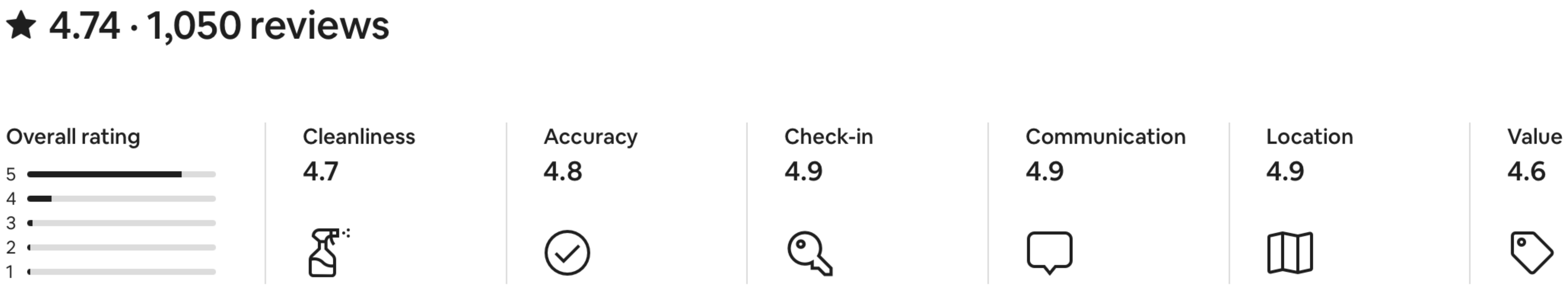
The `broom` package provides helpful tools for tidying model output:

```
library(broom)

# convert model estimates to a data frame for plotting
tidy(name_of_model)

# return a data frame that includes predictions, residuals, etc.
augment(name_of_model)
```

Modeling Airbnb reviews

Let's use some real-world data to explore linear regression

Put yourself in the shoes of a landlord trying to decide how much to invest in improvements across these categories:



Let's see how well "accuracy" reviews predict an Airbnb's overall rating

Modeling Airbnb reviews

$$\text{rating} = \beta_0 + \beta_1 \text{accuracy} + \varepsilon$$

```r
review_model <- lm(
  rating ~ accuracy,
  data = reviews
  )
```

Note how we didn't write anything for the β_0 or ε terms

What do you think the sign on β_1 is?

```r
review_model
```

```
##
## Call:
## lm(formula = rating ~ accuracy, data = reviews)
##
## Coefficients:
## (Intercept)     accuracy
##      0.7590       0.8271
```

Modeling Airbnb reviews

```
summary(review_model)
```

```
##
## Call:
## lm(formula = rating ~ accuracy, data = reviews)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8943 -0.0648  0.0608  0.1057  4.2410
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.758952   0.017156   44.24   <2e-16 ***
## accuracy    0.827067   0.003597  229.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2996 on 28159 degrees of freedom
##   (10116 observations deleted due to missingness)
## Multiple R-squared:  0.6525,    Adjusted R-squared:  0.6525
## F-statistic: 5.287e+04 on 1 and 28159 DF,  p-value: < 2.2e-16
```

Modeling Airbnb reviews

```r
tidy(review_model, conf.int = TRUE)
```

```
## # A tibble: 2 × 7
##   term        estimate std.error statistic p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)    0.759   0.0172       44.2       0    0.725     0.793
## 2 accuracy       0.827   0.00360     230.        0    0.820     0.834
```

Interpretation for a continuous variable

$$y = \beta_0 + \beta_1 x_1 + \varepsilon$$

On average, a one unit increase in x_1 is *associated* with a β_1 change in y

$$\text{rating} = \beta_0 + \beta_1 \text{accuracy} + \varepsilon$$

$$\widehat{\text{rating}} = 0.76 + 0.83 \times \text{accuracy}$$

On average, a one unit increase in accuracy rating is associated with 0.83 higher overall rating

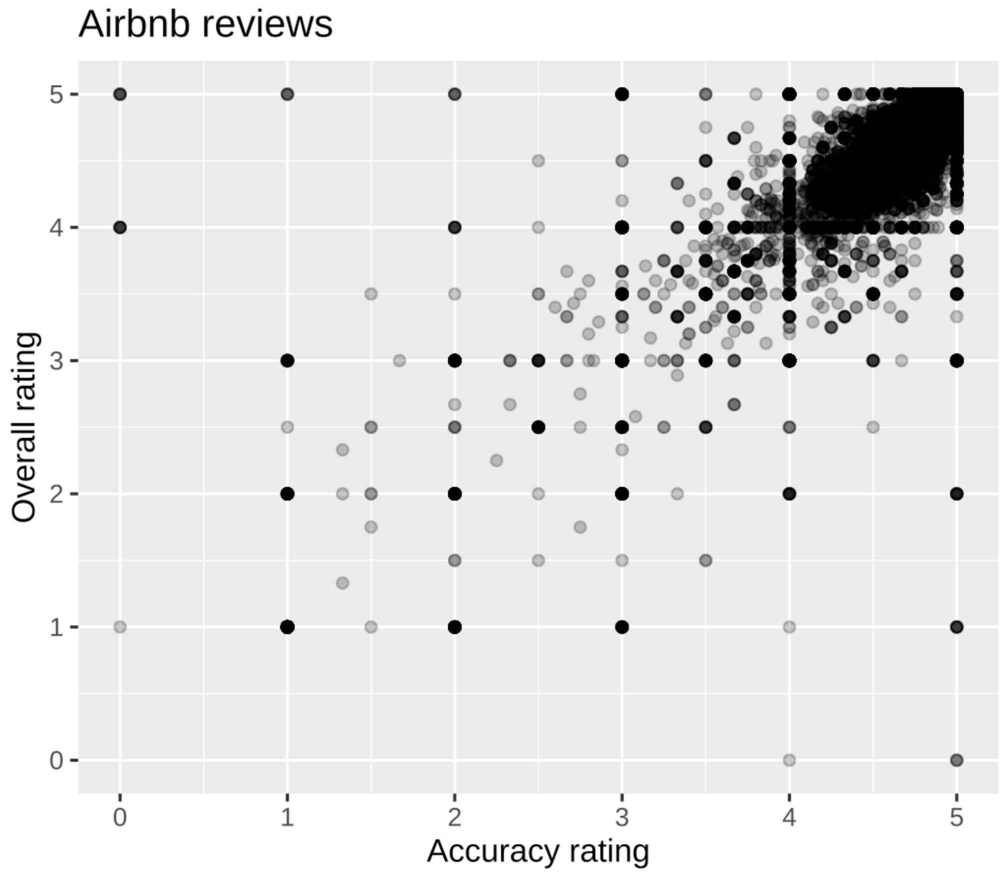
This is easy to visualize: it's a line!

Visualization of a continuous variable

```
tidy(review_model) |>
  select(term, estimate)
```

```
## # A tibble: 2 × 2
##   term        estimate
##   <chr>          <dbl>
## 1 (Intercept)    0.759
## 2 accuracy       0.827
```

$$\widehat{\text{rating}} = 0.76 + 0.83 \times \text{accuracy}$$



Visualization of a continuous variable

```
tidy(review_model) |>
  select(term, estimate)
```

```
## # A tibble: 2 × 2
##   term        estimate
##   <chr>          <dbl>
## 1 (Intercept)    0.759
## 2 accuracy       0.827
```

$$\widehat{\text{rating}} = 0.76 + 0.83 \times \text{accuracy}$$

Visualization of a continuous variable

Reminder: `geom_smooth(method = "lm")` allows us to skip the estimation step!

```
reviews |>
  ggplot(aes(x = accuracy, y = rating)) +
  geom_point(alpha = 0.25) +
  geom_smooth(
    method = "lm",      # smoothing function
    se = FALSE          # omit confidence bands
  )
```

Multiple regression

We're not limited to just one explanatory variable!

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$$

```r
review_model_big <- lm(rating ~ accuracy + cleanliness +
                         communication + location + checkin + value,
                       data = reviews)
```

$$\begin{aligned}
\widehat{\text{rating}} =& \widehat{\beta}_0 + \widehat{\beta}_1 \text{accuracy} + \widehat{\beta}_2 \text{cleanliness} + \\
& \widehat{\beta}_3 \text{communication} + \widehat{\beta}_4 \text{location} + \\
& \widehat{\beta}_5 \text{checkin} + \widehat{\beta}_6 \text{value}
\end{aligned}$$

Multiple regression

We started by estimating this **univariate** (aka **bivariate**) regression model:

$$\text{rating} = \beta_0 + \beta_1 \text{accuracy} + \varepsilon$$

Now we are estimating this **multivariate** regression model:

$$\begin{aligned}\text{rating} = &\beta_0 + \beta_1 \text{accuracy} + \beta_2 \text{cleanliness} + \\ &\beta_3 \text{communication} + \beta_4 \text{location} + \\ &\beta_5 \text{checkin} + \beta_6 \text{value} + \varepsilon\end{aligned}$$

Do you think the coefficient on `accuracy` will be smaller, larger, or the same as in the simpler model? Why?

Multiple regression

```
tidy(review_model_big, conf.int = TRUE)
```

```
## # A tibble: 7 × 7
##   term          estimate std.error statistic   p.value conf.low conf.high
##   <chr>            <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)    -0.124     0.0178     -6.96 3.43e- 12  -0.159    -0.0892
## 2 accuracy        0.217     0.00531    40.8  0           0.206     0.227
## 3 cleanliness     0.227     0.00356    63.9  0           0.220     0.234
## 4 communication   0.169     0.00507    33.4  1.45e-239   0.159     0.179
## 5 location        0.0384    0.00428     8.97 3.25e- 19   0.0300    0.0468
## 6 checkin         0.0578    0.00521    11.1  1.37e- 28   0.0476    0.0680
## 7 value           0.313     0.00476    65.8  0           0.304     0.323
```

$$
\begin{aligned}
\widehat{\text{rating}} = & -0.12 + 0.22 \times \text{accuracy} + 0.23 \times \text{cleanliness} + \\
& 0.17 \times \text{communication} + 0.04 \times \text{location} + \\
& 0.06 \times \text{checkin} + 0.31 \times \text{value}
\end{aligned}
$$

Interpretation for continuous variables

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$$

Holding everything else constant, a one unit increase in x_n is *associated* with a β_n change in y, on average

$$\begin{aligned}\widehat{\text{rating}} = & -0.12 + 0.22 \times \text{accuracy} + 0.23 \times \text{cleanliness} + \\ & 0.17 \times \text{communication} + 0.04 \times \text{location} + \\ & 0.06 \times \text{checkin} + 0.31 \times \text{value}\end{aligned}$$

On average, a one unit increase in accuracy rating is associated with 0.22 higher overall rating, holding everything else constant

For the earlier model we had said

> On average, a one unit increase in accuracy rating is associated with 0.83 higher overall rating

Good luck visualizing all this!

You can't just draw a single line! There are too many moving parts!

Main challenges

Each coefficient has its own estimate and standard errors

Solution: Plot the coefficients and their errors with a *coefficient plot*

The results can change as you move each slider (continuous variable) up and down and flip each switch (categorical variable) on and off

Solution: Plot the *marginal effects* for the coefficients you're interested in

Coefficient plots

Convert the model results to a data frame with `tidy()`

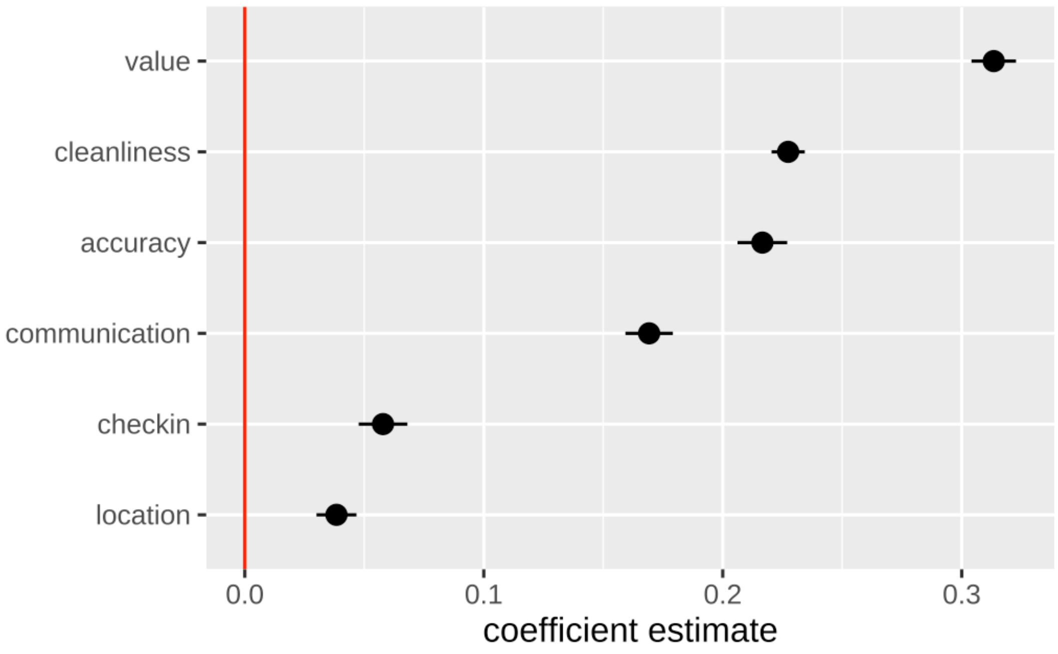
```r
# tidy the estimates (reformatting names is not required)
review_coefs <- tidy(review_model_big, conf.int = TRUE) |>
  filter(term!="(Intercept)")

review_coefs
```

```
## # A tibble: 6 × 7
##   term          estimate std.error statistic   p.value conf.low conf.high
##   <chr>            <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 accuracy        0.217    0.00531     40.8  0            0.206     0.227
## 2 cleanliness     0.227    0.00356     63.9  0            0.220     0.234
## 3 communication   0.169    0.00507     33.4  1.45e-239    0.159     0.179
## 4 location        0.0384   0.00428      8.97 3.25e- 19    0.0300    0.0468
## 5 checkin         0.0578   0.00521     11.1  1.37e- 28    0.0476    0.0680
## 6 value           0.313    0.00476     65.8  0            0.304     0.323
```

Coefficient plots

Plot the point estimate and confidence intervals with `geom_pointrange()`

```
review_coefs |>
  ggplot(aes(x = estimate,
             y = fct_reorder(term, estimate))) +
  geom_pointrange(aes(xmin = conf.low,
                      xmax = conf.high)) +
  geom_vline(xintercept = 0, color = "red") +
  labs(x = "coefficient estimate",
       y = NULL)
```



What do you take away from this?

Should this inform where you decide to focus your investment as a landlord?

Marginal effects plots

Remember that we interpret individual coefficients while holding the others constant

We move one slider while leaving all the other sliders and switches alone

Same principle applies to visualizing the effect

Plug a bunch of values into the model and find the predicted outcome

Plot the values and predicted outcome

Marginal effects plots

Create a data frame of values you want to manipulate and values you want to hold constant

Must include all the explanatory variables in the model

Marginal effects plots

```
reviews_new_data <- reviews |>
  select(rating, accuracy, cleanliness, checkin, communication, location, value) |>
  mutate(
    across(
      c(cleanliness, checkin, communication, location, value),
      ~ mean(.x, na.rm = TRUE)
    )
  )

head(reviews_new_data)
```

```
## # A tibble: 6 × 7
##   rating accuracy cleanliness checkin communication location value
##    <dbl>    <dbl>       <dbl>   <dbl>         <dbl>    <dbl> <dbl>
## 1   4.7      4.72        4.61    4.81          4.81     4.75  4.65
## 2   4.45     4.58        4.61    4.81          4.81     4.75  4.65
## 3   4.52     4.22        4.61    4.81          4.81     4.75  4.65
## 4   5        5           4.61    4.81          4.81     4.75  4.65
## 5   4.21     4.21        4.61    4.81          4.81     4.75  4.65
## 6   4.91     4.83        4.61    4.81          4.81     4.75  4.65
```

Marginal effects plots

Plug each of those rows of data into the model with `augment()`

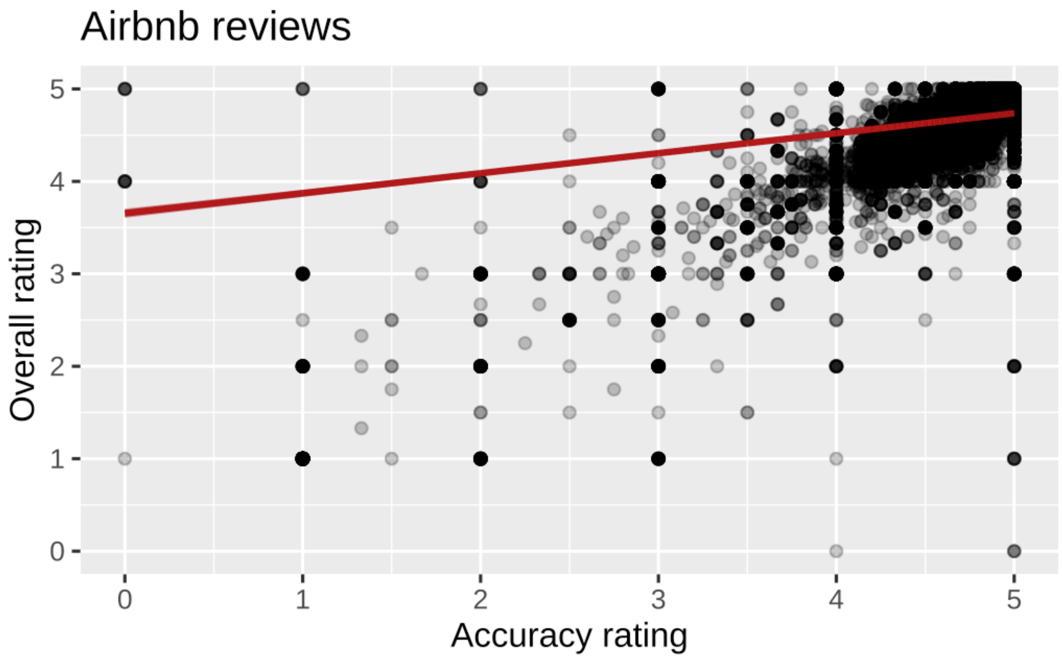
```
predicted_reviews <- augment(review_model_big,              # our estimated model
                             newdata = reviews_new_data,  # our new data for plotting
                             interval = "confidence")     # add confidence intervals

head(predicted_reviews)
```

```
## # A tibble: 6 × 11
##   rating accuracy cleanliness checkin communication location value .fitted
##    <dbl>    <dbl>       <dbl>   <dbl>         <dbl>    <dbl> <dbl>   <dbl>
## 1   4.7      4.72        4.61    4.81          4.81     4.75  4.65    4.68
## 2   4.45     4.58        4.61    4.81          4.81     4.75  4.65    4.65
## 3   4.52     4.22        4.61    4.81          4.81     4.75  4.65    4.57
## 4   5        5           4.61    4.81          4.81     4.75  4.65    4.74
## 5   4.21     4.21        4.61    4.81          4.81     4.75  4.65    4.57
## 6   4.91     4.83        4.61    4.81          4.81     4.75  4.65    4.70
## # ℹ 3 more variables: .lower <dbl>, .upper <dbl>, .resid <dbl>
```

Marginal effects plots

Plot the fitted values for each row

```
mfx_plot <- predicted_reviews |>
  ggplot(aes(x = accuracy, y = rating)) +
  geom_point(alpha = 0.25) +
  geom_line( # multivariate regression
    aes(y = .fitted),
    color = "#B31B1B",
    linewidth = 1
  ) +
  geom_ribbon(aes(ymin = .lower,
                  ymax = .upper),
              fill = "#B31B1B",
              alpha = 0.5) +
  labs(x = "Accuracy rating",
       y = "Overall rating",
       title = "Airbnb reviews")
mfx_plot
```



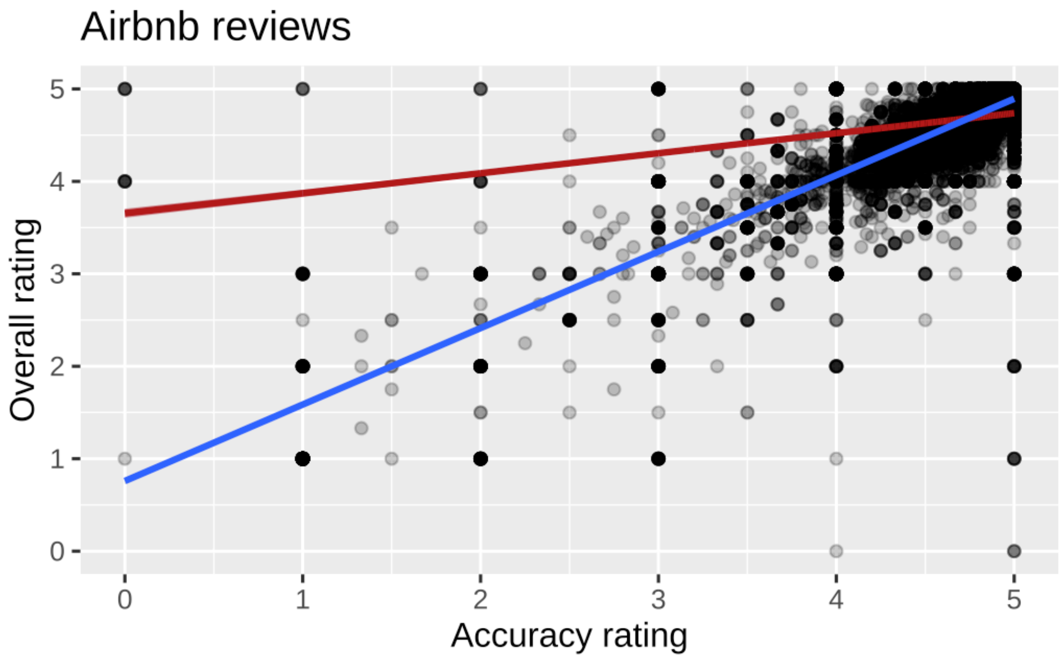
Marginal effects plots

How does this regression line compare to our univariate regression line?

```
mfx_plot +
  geom_smooth( # univariate regression
    method = "lm"
    )
```

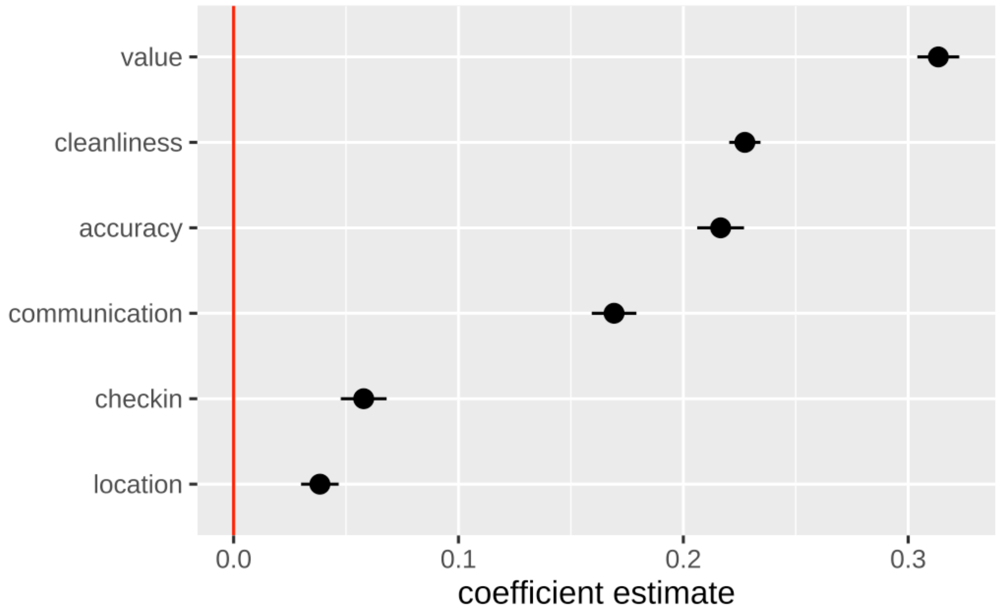
What do you take away from this?

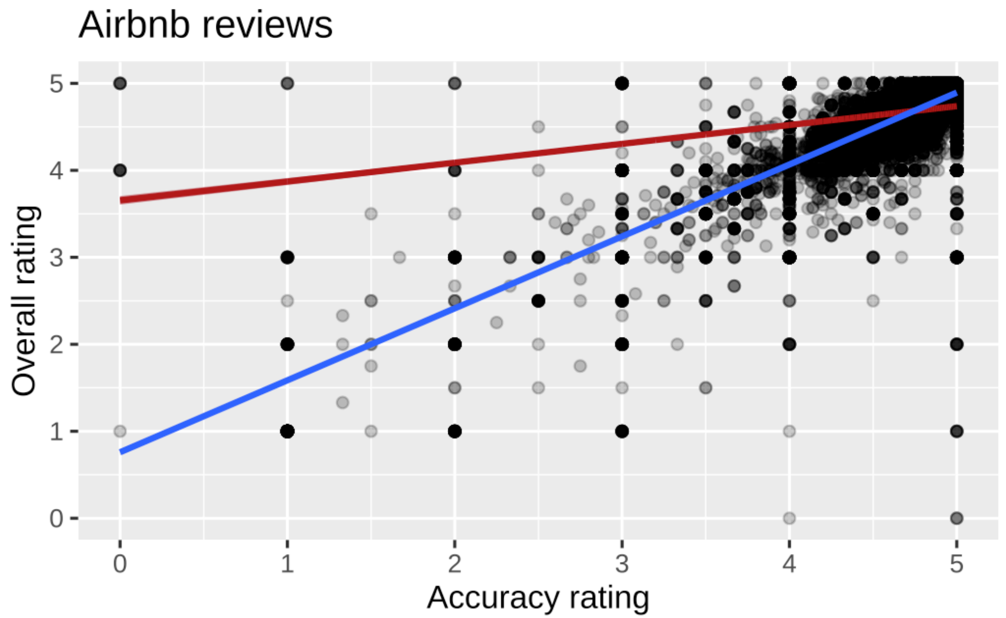
Should this affect how much you invest in accuracy?



Stepping back

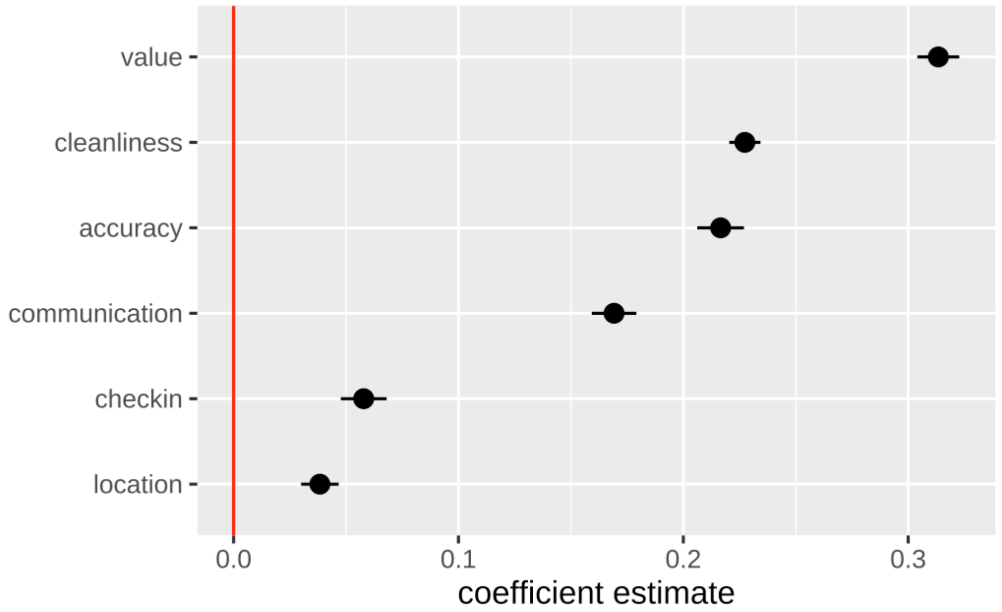
Which of these plots would be more useful to Airbnb landlords? Why?

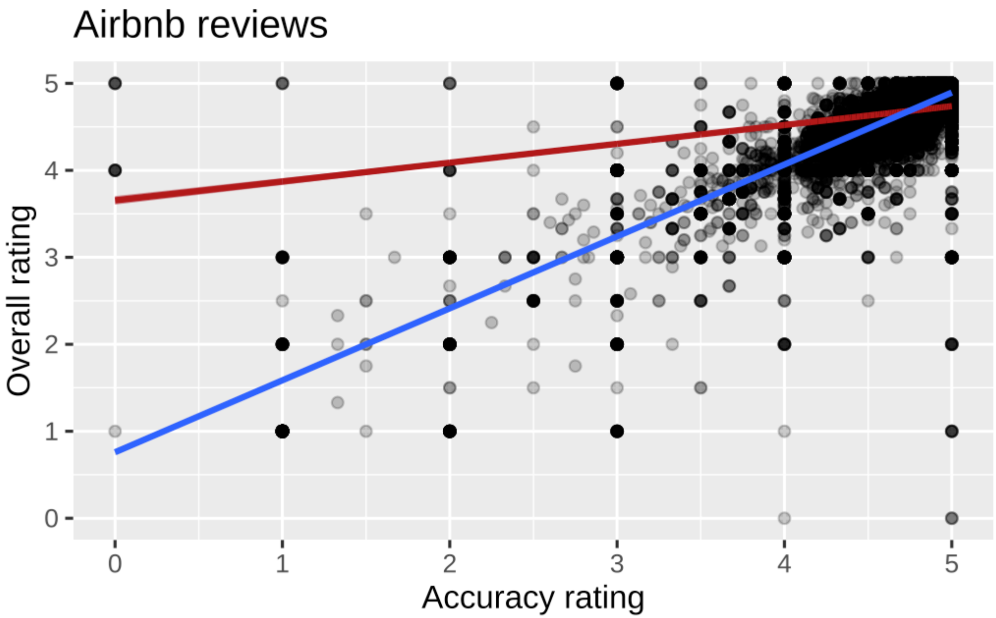




Not just OLS!

These plots are for an OLS model built with `lm()`





Any type of statistical model

The same techniques work for pretty much any model R can run

- OLS with high-dimensional fixed effects
- Logistic, probit, and multinomial regression (ordered and unordered)
- Multilevel (i.e., mixed and random effects) regression
- Bayesian models
- Machine learning models

If it has coefficients and/or makes predictions, you can (and should) visualize it!