ACCT 420: Linear Regression

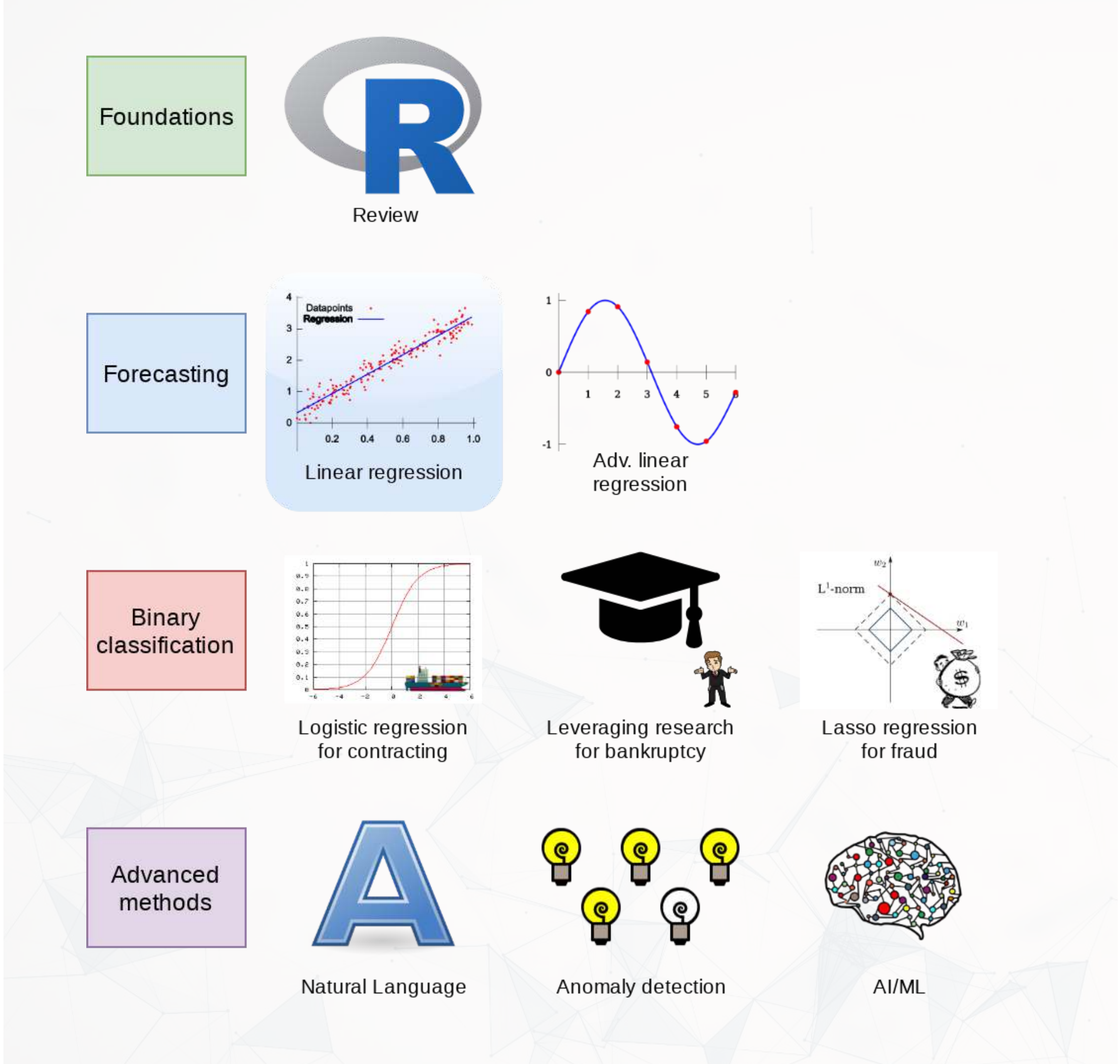
Dr. Richard M. Crowley

rcrowley@smu.edu.sg

https://rmc.link/

Front Matter

Learning objectives



Foundations

Review

Forecasting

Linear regression

Adv. linear regression

Binary classification

Logistic regression for contracting

Leveraging research for bankruptcy

Lasso regression for fraud

Advanced methods

Natural Language

Anomaly detection

AI/ML

- **Theory:**
  - Develop a logical approach to problem solving with data
    - Statistics
    - Causation
    - Hypothesis testing
- **Application:**
  - Predicting revenue for real estate firms
- **Methodology:**
  - Univariate stats
  - Linear regression
  - Visualization

Datacamp

- For next week:
  - 1 suggested chapters on tidyverse methods
- The full list of suggested Datacamp materials for the course is up on eLearn

> Datacamp is optional. If you find the coding difficult in today's lesson, you should go through the suggested datacamp chapters

R Installation

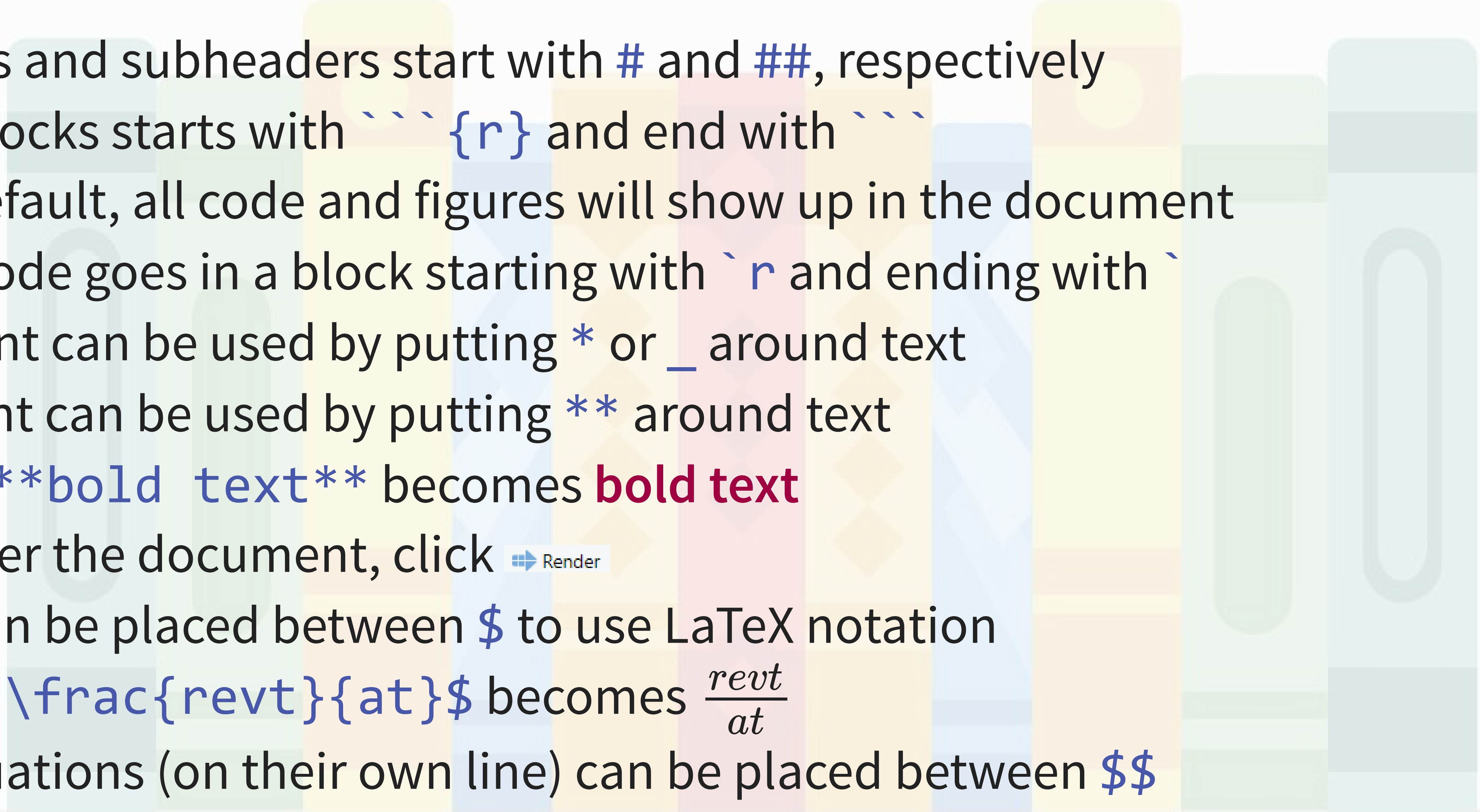
- If you haven’t already, make sure to install R, R Studio, and Quarto!
  - Instructions are in Session 1’s slides
  - You will need it for this week’s assignment
- Please install a few packages using the following code
  - These packages are also needed for the first assignment
  - You are welcome to explore other packages as well

```r
# Run this in the R Console inside RStudio
install.packages(c("tidyverse","plotly"))
```

- Assignments will be provided as Quarto files

> The format will generally all be filled out – you will just add to it, answer questions, analyze data, and explain your work. Instructions and hints are in the same file

R Markdown: A quick guide



- Headers and subheaders start with # and ##, respectively
- Code blocks starts with ```{r} and end with ```
  - By default, all code and figures will show up in the document
- Inline code goes in a block starting with `` `r `` and ending with `` ` ``
- Italic font can be used by putting * or _ around text
- Bold font can be used by putting ** around text
  - E.g.: `**bold text**` becomes **bold text**
- To render the document, click Render
- Math can be placed between $ to use LaTeX notation
  - E.g. `$\frac{revt}{at}$` becomes $\frac{revt}{at}$
- Full equations (on their own line) can be placed between $$
- A block quote is prefixed with >
- For a complete guide, see the Quarto tutorials or Datacamp's Quarto Cheat Sheet

Application: Revenue prediction

The question

> How can we predict revenue for a company, leveraging data about that company, related companies, and macro factors

- Specific application: Real estate companies

More specifically…

- Can we use a company’s own accounting data to predict it’s [future] revenue?
- Can we use other companies’ accounting data to better predict all of their future revenue?
- Can we augment this data with macro economic data to further improve prediction?
  - Singapore business sentiment data

Linear models

What is a linear model?

$$\hat{y} = \alpha + \beta\hat{x} + \varepsilon$$

- The simplest model is trying to predict some outcome $\hat{y}$ as a function of an input $\hat{x}$
  - $\hat{y}$ in our case is a firm's revenue in a given year
  - $\hat{x}$ could be a firm's assets in a given year
  - α and β are solved for
  - ε is the error in the measurement

> I will refer to this as an *OLS* model – **O**rdinary **L**east **S**quare regression

Example

Let’s predict UOL’s revenue for 2016

- Compustat has data for them since 1989
  - Complete since 1994
    - Missing CapEx before that

```r
# revt: Revenue, at: Assets
summary(uol[,c("revt", "at")])
```

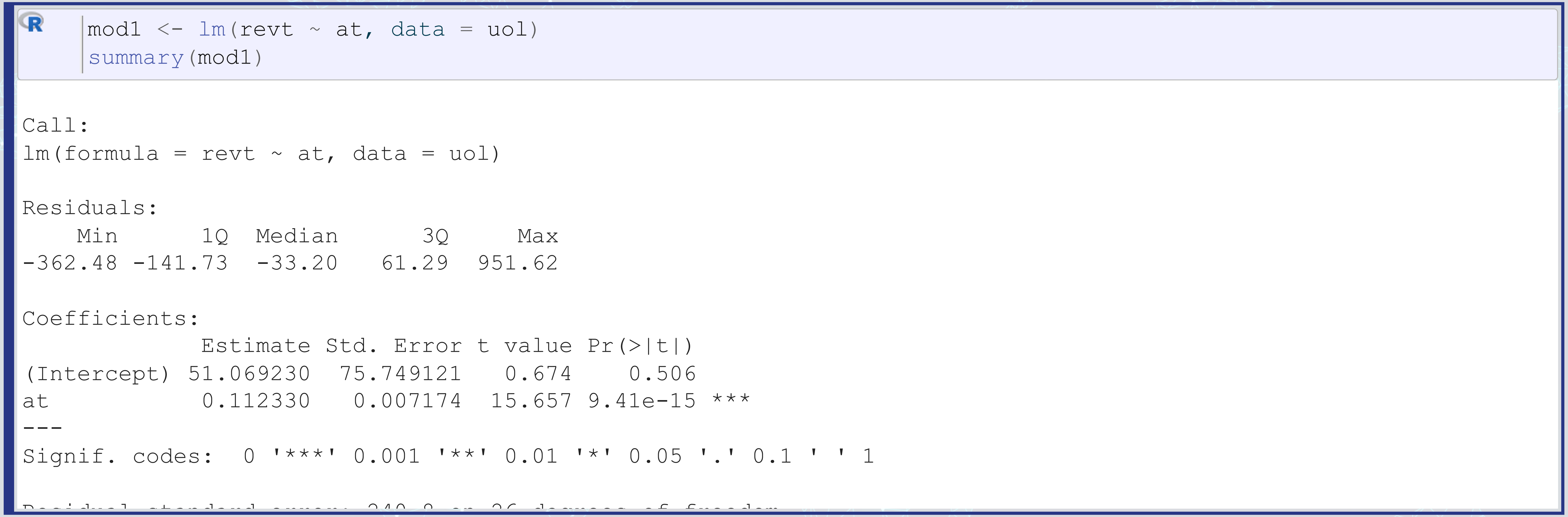
```
      revt              at       
 Min.   : 155.1   Min.   : 2366  
 1st Qu.: 347.1   1st Qu.: 3277  
 Median : 804.1   Median : 6138  
 Mean   : 999.2   Mean   : 8440  
 3rd Qu.:1380.7   3rd Qu.:11515  
 Max.   :2606.8   Max.   :21275  
```

Linear models in R

- To run a linear model, use `lm()`
  - The first argument is a formula for your model, where `~` is used in place of an equals sign
    - The left side is what you want to predict
    - The right side is inputs for prediction, separated by `+`
  - The second argument is the data to use
- Additional variations for the formula:
  - Functions transforming inputs (as vectors), such as `log()`
  - Fully interacting variables using `*`
    - I.e., `A*B` includes: `A`, `B`, and `A times B` in the model
  - Interactions using `:`
    - I.e., `A:B` only includes `A times B` in the model

```r
# Example:
lm(revt ~ at, data = uol)
```

Example: UOL

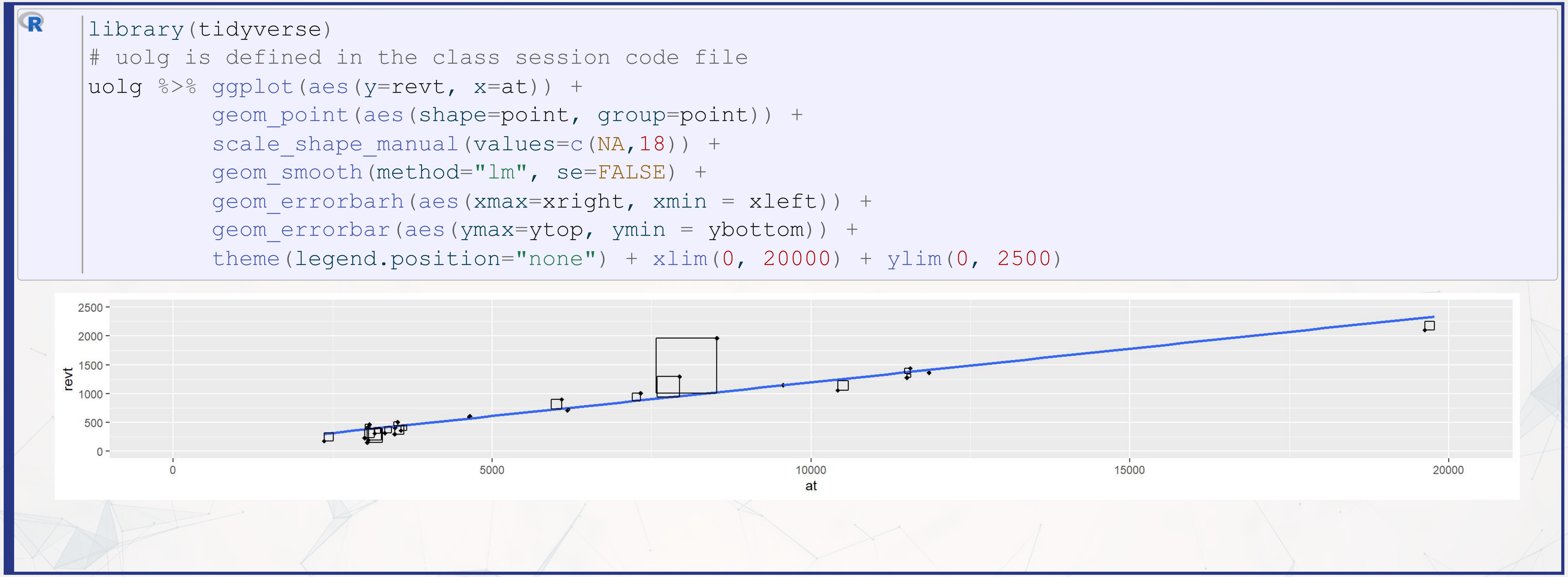


```r
mod1 <- lm(revt ~ at, data = uol)
summary(mod1)
```

```

Call:
lm(formula = revt ~ at, data = uol)

Residuals:
    Min      1Q  Median      3Q     Max 
-362.48 -141.73  -33.20   61.29  951.62 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 51.069230  75.749121   0.674    0.506    
at           0.112330   0.007174  15.657 9.41e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 240.8 on 26 degrees of freedom
```

> $1 more in assets leads to $0.12 more revenue

Why is it called Ordinary Least *Squares*?



```r
library(tidyverse)
# uolg is defined in the class session code file
uolg %>% ggplot(aes(y=revt, x=at)) +
         geom_point(aes(shape=point, group=point)) +
         scale_shape_manual(values=c(NA,18)) +
         geom_smooth(method="lm", se=FALSE) +
         geom_errorbarh(aes(xmax=xright, xmin = xleft)) +
         geom_errorbar(aes(ymax=ytop, ymin = ybottom)) +
         theme(legend.position="none") + xlim(0, 20000) + ylim(0, 2500)
```

Example: UOL

- This model wasn’t so interesting…
  - Bigger firms have more revenue – this is a given
  - Though it does tell us *something* about the relationship between assets and revenue

Abstracting a problem

If we don’t want to factor in firm size, we can use ratios to abstract away from it!

- How about… revenue *growth*?
- Then we can use *change* in assets in the model
  - i.e., Asset growth

$$\Delta x_t = \frac{x_t}{x_{t-1}} - 1$$

Calculating changes in R

- The easiest way is using tidyverse's dplyr
  - This has a `lag()` function
- The default way to do it is to create a vector manually

```r
# tidyverse
uol <- uol %>%
  mutate(revt_growth1 = revt / lag(revt) - 1)

# R way
uol$revt_growth2 = uol$revt / c(NA, uol$revt[-length(uol$revt)]) - 1

# Check that both ways are equivalent
identical(uol$revt_growth1, uol$revt_growth2)
```

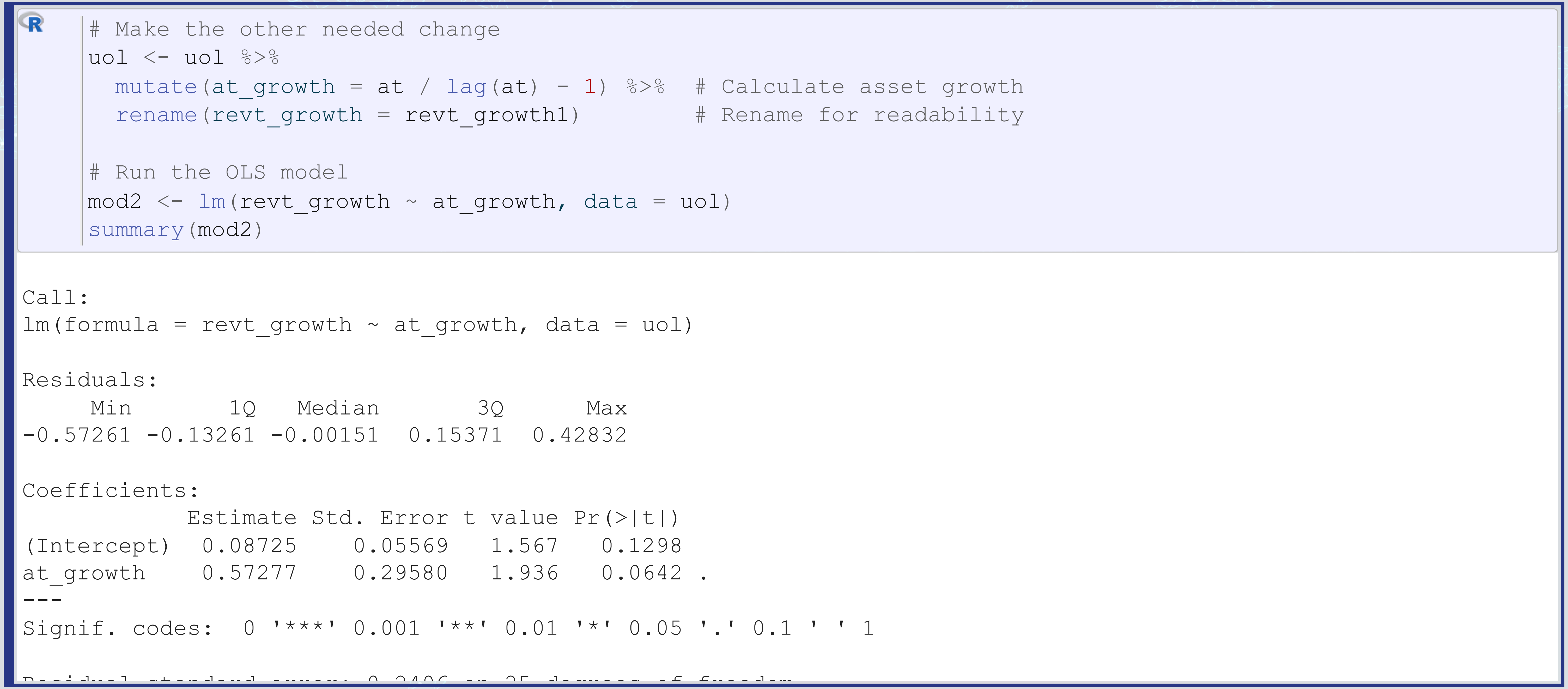
```
[1] TRUE
```

> You can use whichever you are comfortable with

A note on mutate()

- `mutate()` adds variables to an existing data frame
  - If you need to manipulate a bunch of columns at once:
    - `across()` applies a transformation to multiple columns in a data frame
    - You can mix in `starts_with()` or `ends_with()` to pick columns by pattern
- Mutate can be very powerful when making complex sets of variables
  - Examples:
    - Calculating growth within company in a multi-company data frame
    - Normalizing data to be within a certain range for multiple variables at once

Example: UOL with changes



```r
# Make the other needed change
uol <- uol %>%
  mutate(at_growth = at / lag(at) - 1) %>%  # Calculate asset growth
  rename(revt_growth = revt_growth1)         # Rename for readability

# Run the OLS model
mod2 <- lm(revt_growth ~ at_growth, data = uol)
summary(mod2)
```

```
Call:
lm(formula = revt_growth ~ at_growth, data = uol)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.57261 -0.13261 -0.00151  0.15371  0.42832 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  0.08725    0.05569   1.567   0.1298  
at_growth    0.57277    0.29580   1.936   0.0642 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2406 on 25 degrees of freedom
```

Example: UOL with changes

- ΔAssets doesn’t capture ΔRevenue so well
- Perhaps change in total assets is a bad choice?
- Or perhaps we need to expand our model?

Scaling up!

$$\hat{y} = \alpha + \beta_1 \hat{x}_1 + \beta_2 \hat{x}_2 + \ldots + \varepsilon$$

- OLS doesn't need to be restricted to just 1 input!
  - Not unlimited though (yet – we'll get there)
    - Number of inputs must be less than the number of observations minus 1
- Each $\hat{x}_i$ is an input in our model
- Each β_i is something we will solve for
- $\hat{y}$, α, and ε are the same as before

Scaling up our model

> We have… 464 variables from Compustat Global alone!

- Let’s just add them all?

- We only have 28 observations…
  - 28 << 464…

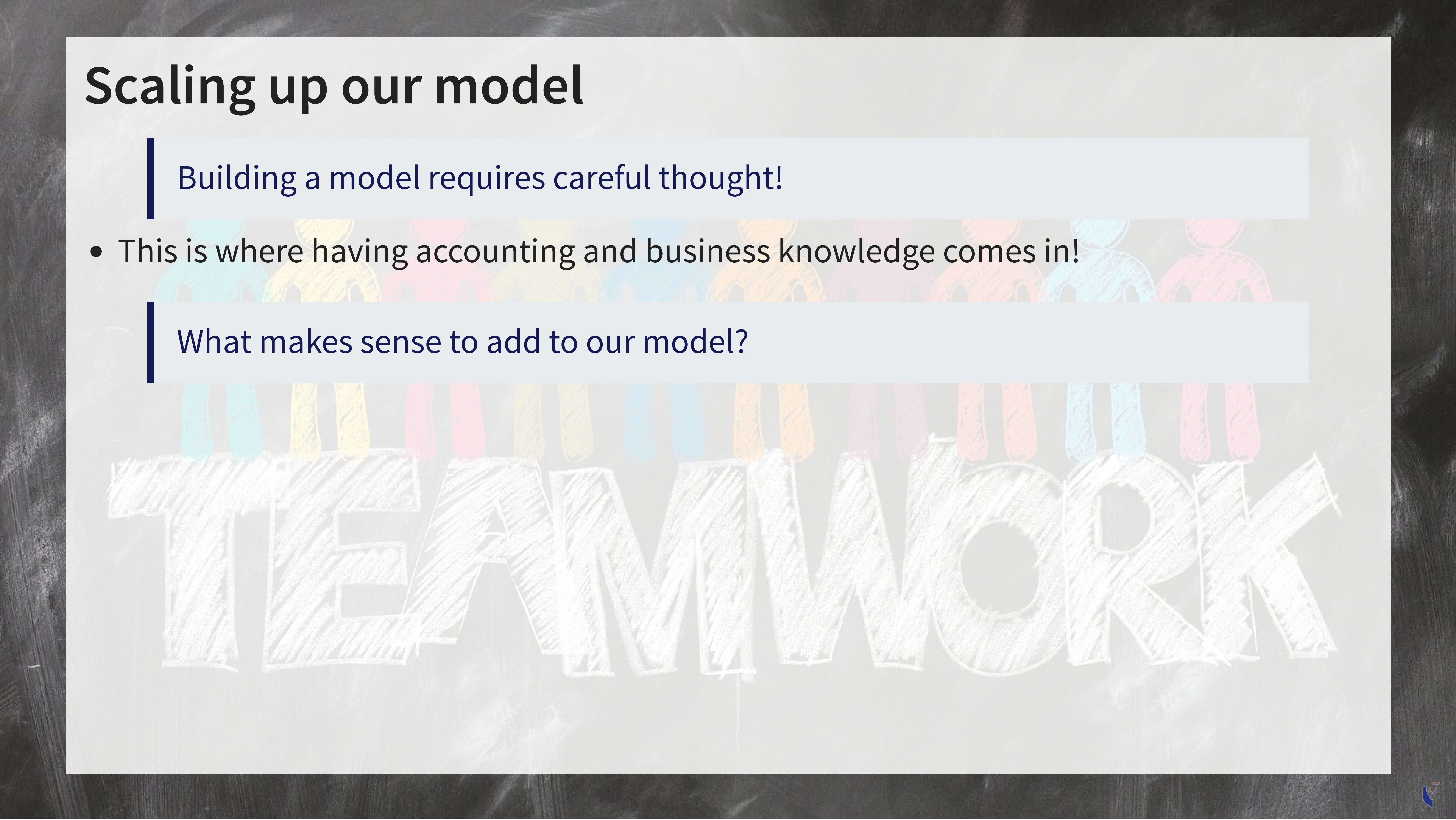
> Now what?

Scaling up our model

> Building a model requires careful thought!

- This is where having accounting and business knowledge comes in!

> What makes sense to add to our model?



Practice: mutate()

- This practice is to make sure you understand how to use mutate with lags
  - These are very important when dealing with business data!
- Do exercises 1 on today's R practice file:
  - R Practice
  - Shortlink: rmc.link/420r2

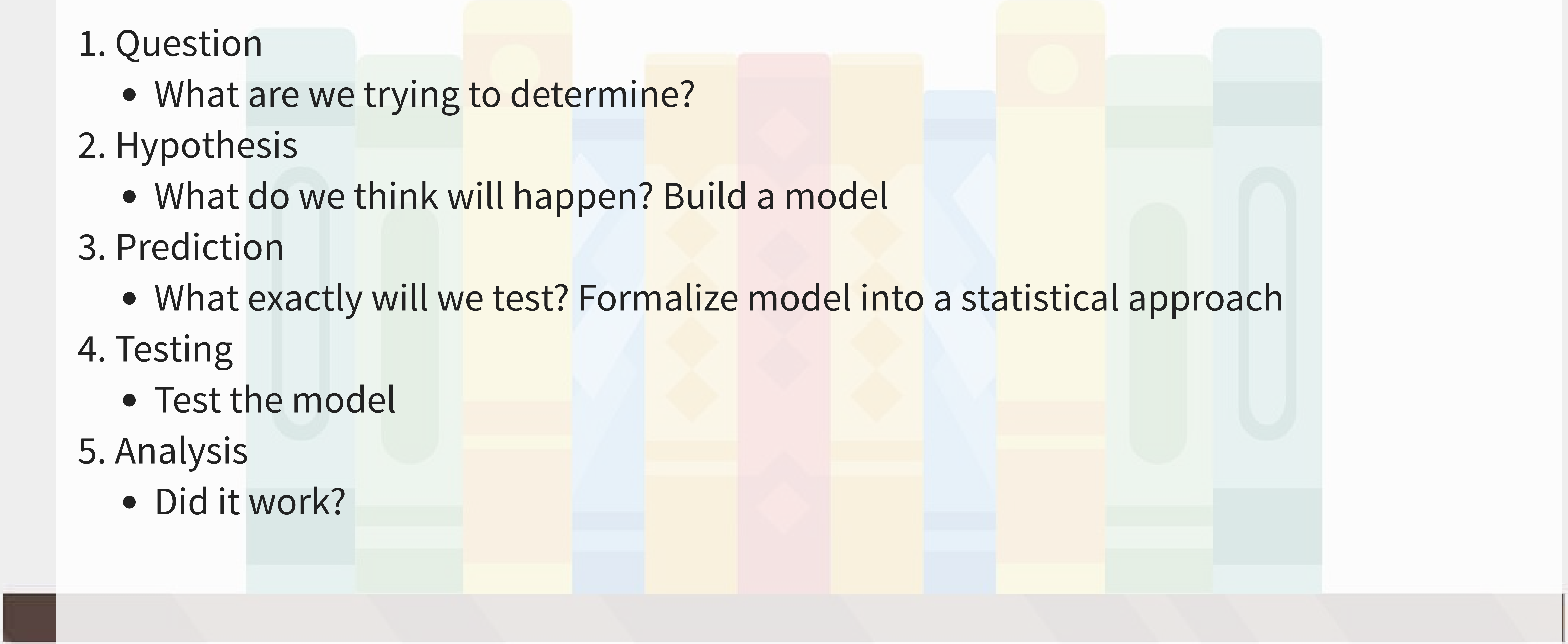
Formalizing frequentist testing

Why formalize?

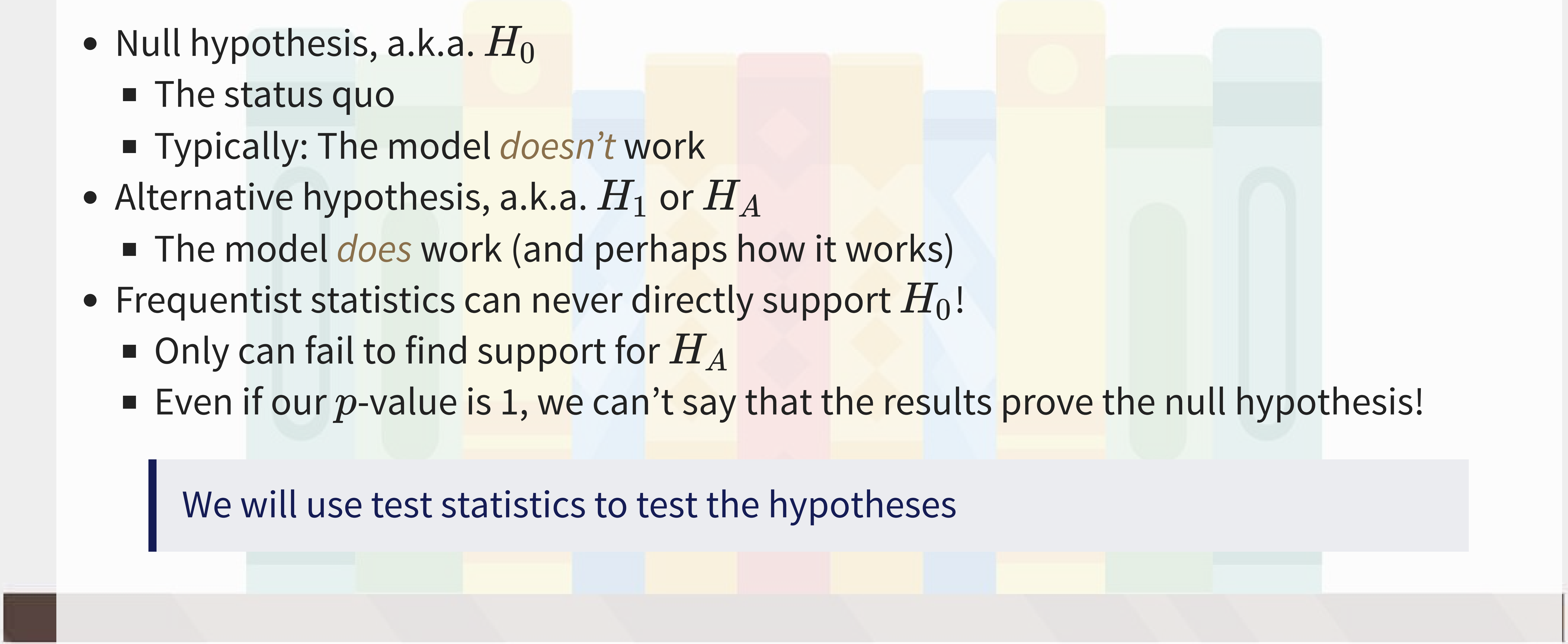
- Our current approach has been ad hoc
  - What is our goal?
  - How will we know if we have achieved it?
- Formalization provides more rigor

Scientific method

1. Question
   - What are we trying to determine?
2. Hypothesis
   - What do we think will happen? Build a model
3. Prediction
   - What exactly will we test? Formalize model into a statistical approach
4. Testing
   - Test the model
5. Analysis
   - Did it work?



Hypotheses



- Null hypothesis, a.k.a. H_0
  - The status quo
  - Typically: The model *doesn't* work
- Alternative hypothesis, a.k.a. H_1 or H_A
  - The model *does* work (and perhaps how it works)
- Frequentist statistics can never directly support H_0!
  - Only can fail to find support for H_A
  - Even if our p-value is 1, we can't say that the results prove the null hypothesis!

> We will use test statistics to test the hypotheses

Regression

- Regression (like OLS) has the following assumptions
  1. The data is generated following some model
     - E.g., a linear model
       - In two weeks, a logistic model
  2. The data conforms to some statistical properties as required by the test
  3. The model coefficients are something to precisely determine
     - I.e., the coefficients are constants
  4. p-values provide a measure of the chance of an error in a particular aspect of the model
     - For instance, the p-value on β_1 in $y = \alpha + \beta_1 x_1 + \varepsilon$ essentially gives the probability that the sign of β_1 is wrong

OLS Statistical properties

$$\begin{aligned} \text{Theory:} \quad & y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \varepsilon \\ \text{Data:} \quad & \hat{y} = \alpha + \beta_1 \hat{x}_1 + \beta_2 \hat{x}_2 + \ldots + \hat{\varepsilon} \end{aligned}$$

1. There should be a *linear* relationship between y and each x_i
   - I.e., y is [approximated by] a constant multiple of each x_i
   - Otherwise we **shouldn't** use a *linear* regression
2. Each $\hat{x}_i$ is normally distributed
   - Not so important with larger data sets, but a good to adhere to
3. Each observation is independent
   - We'll violate this one for the sake of *causality*
4. Homoskedasticity: Variance in errors is constant
   - **This is important** for the tests' reliability
5. Not too much multicollinearity
   - Each $\hat{x}_i$ should be relatively independent from the others (some dependence is OK)

Practical implications

> Models designed under a frequentist approach can only answer the question of “does this matter?”

- Is this a problem?

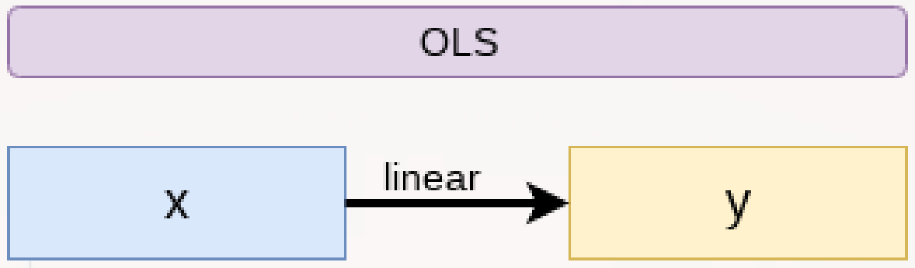
Linear model implementation

What exactly is a linear model?

- Anything OLS is linear
- Many transformations can be recast to linear
  - Ex.: $log(y) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 {x_1}^2 + \beta_4 x_1 \cdot x_2$
    - This is the same as $y' = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$ where:
      - $y' = log(y)$
      - $x_3 = {x_1}^2$
      - $x_4 = x_1 \cdot x_2$

> Linear models are *very* flexible

Mental model of OLS: 1 input



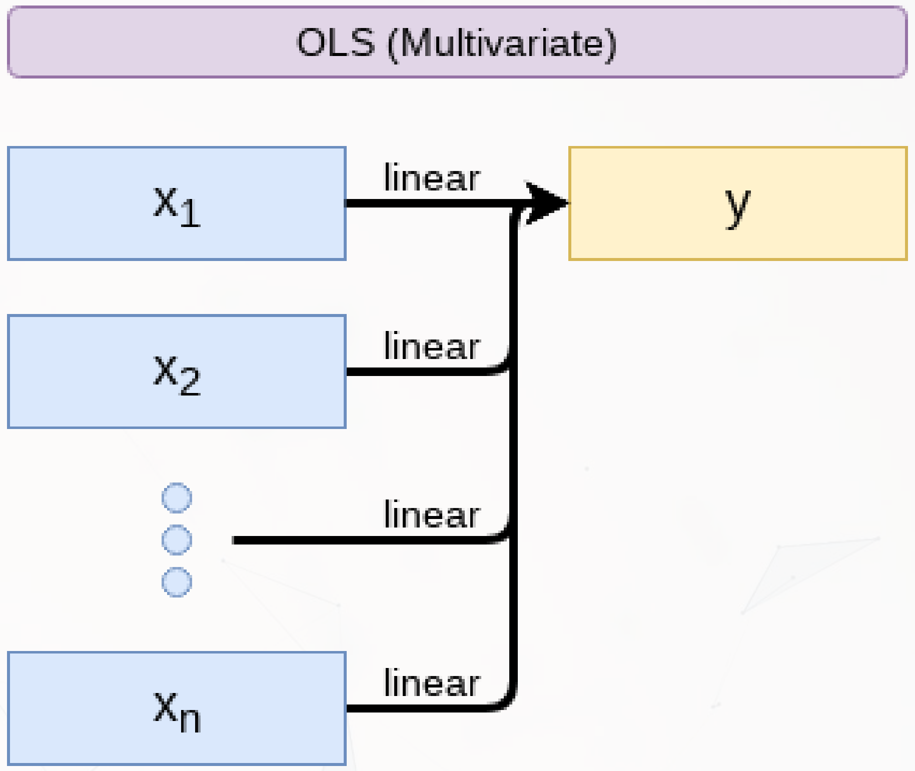
> Simple OLS measures a simple linear relationship between 1 input and 1 output

- E.g.: Our first regression this week: Revenue on assets

Mental model of OLS: Multiple inputs

> OLS measures simple linear relationships between a *set* of inputs and 1 output

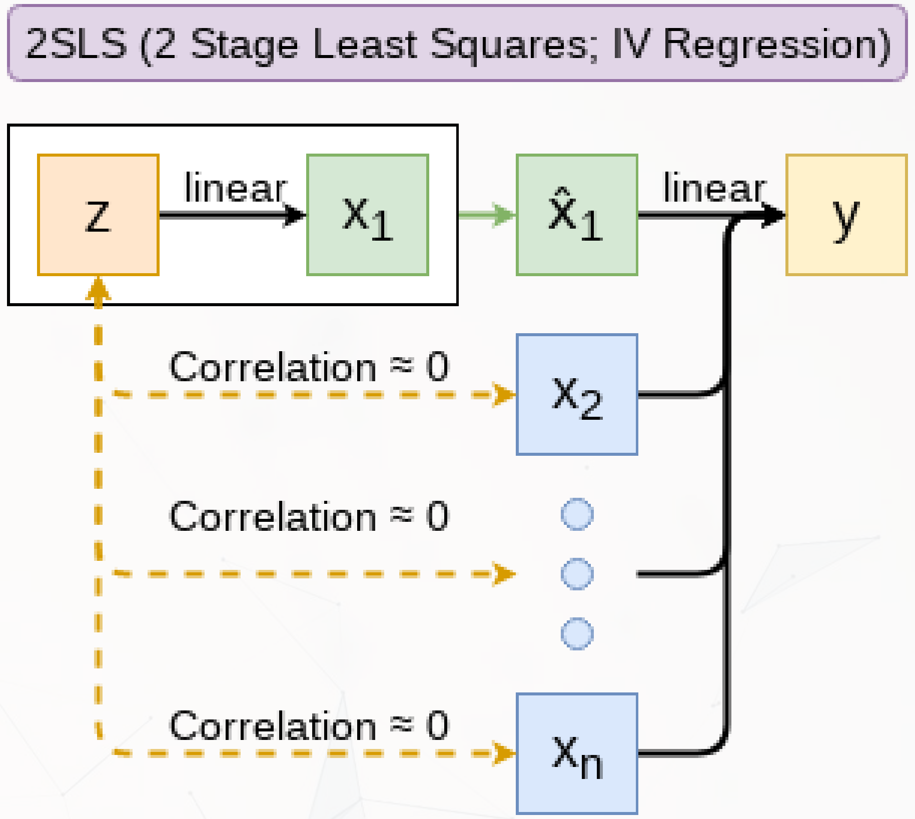
- E.g.: This is what we did when scaling up earlier this session



Other linear models: IV Regression (2SLS)

> IV/2SLS models linear relationships where the effect of some x_i on y may be *confounded by outside factors*.

- E.g.: Modeling the effect of management pay duration (like bond duration) on firms’ choice to issue earnings forecasts
  - Instrument with CEO tenure (Cheng, Cho, and Kim 2015)

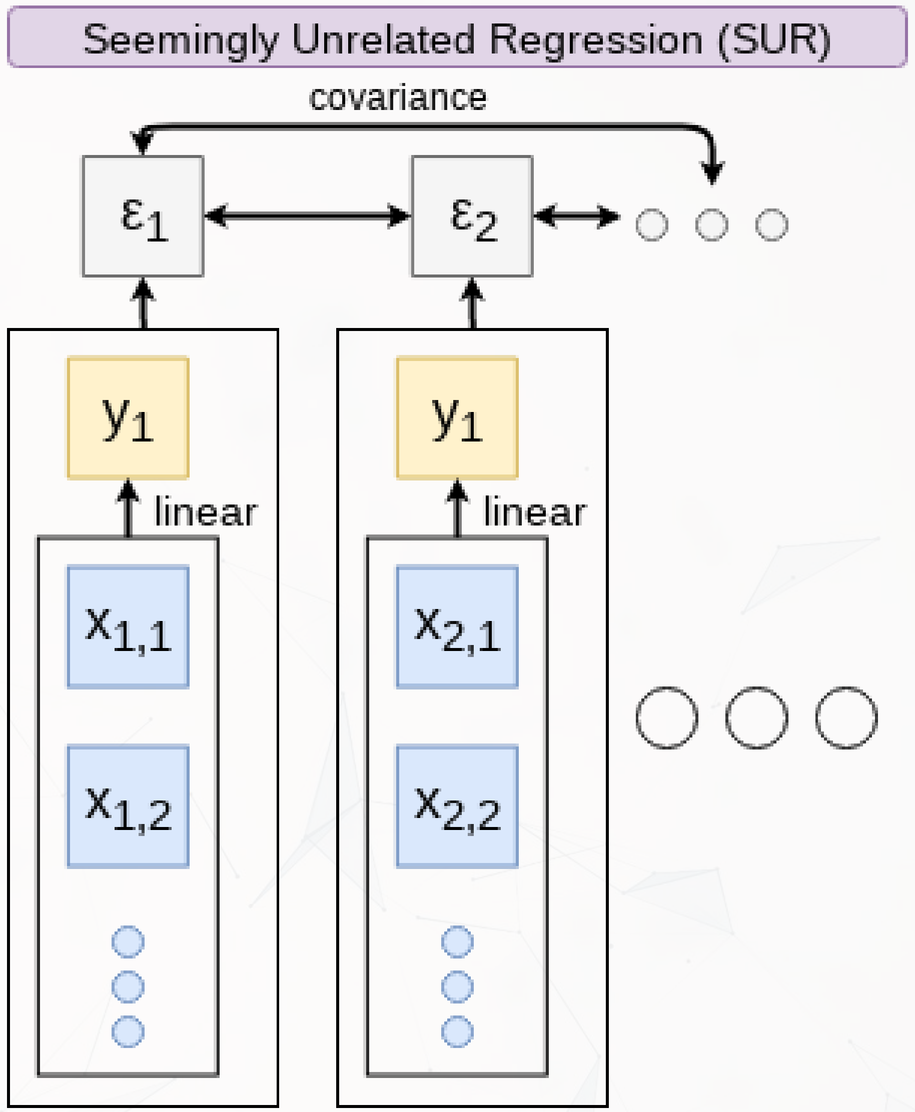


We won’t use this in this course, but you should know it exists.

Other linear models: SUR

> SUR models systems with *related error terms*

- E.g.: Modeling both revenue and earnings simultaneously

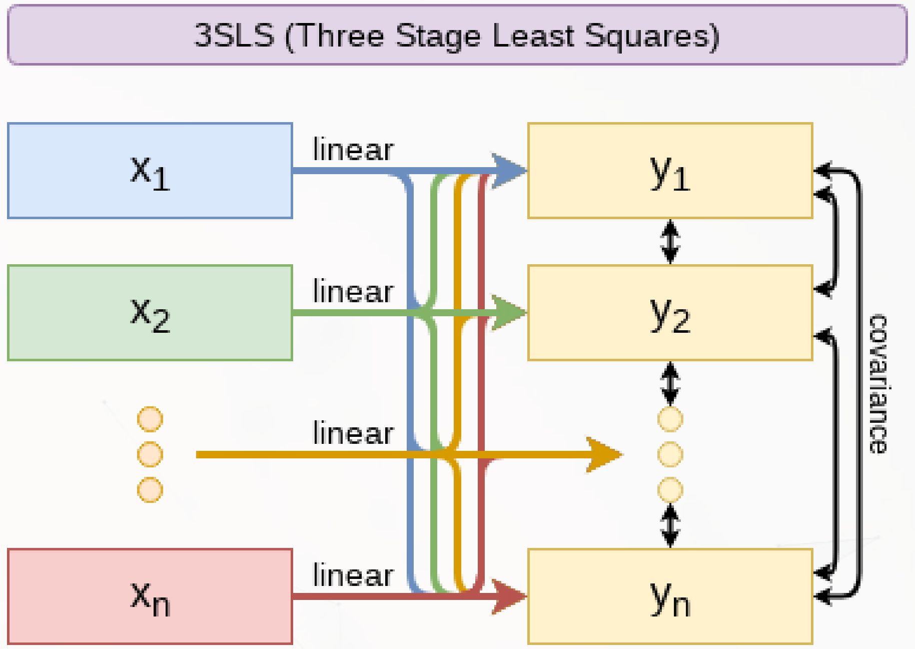


We won’t use this in this course, but you should know it exists.

Other linear models: 3SLS

> 3SLS models systems of equations with *related outputs*

- E.g.: Modeling stock return, volatility, and volume simultaneously

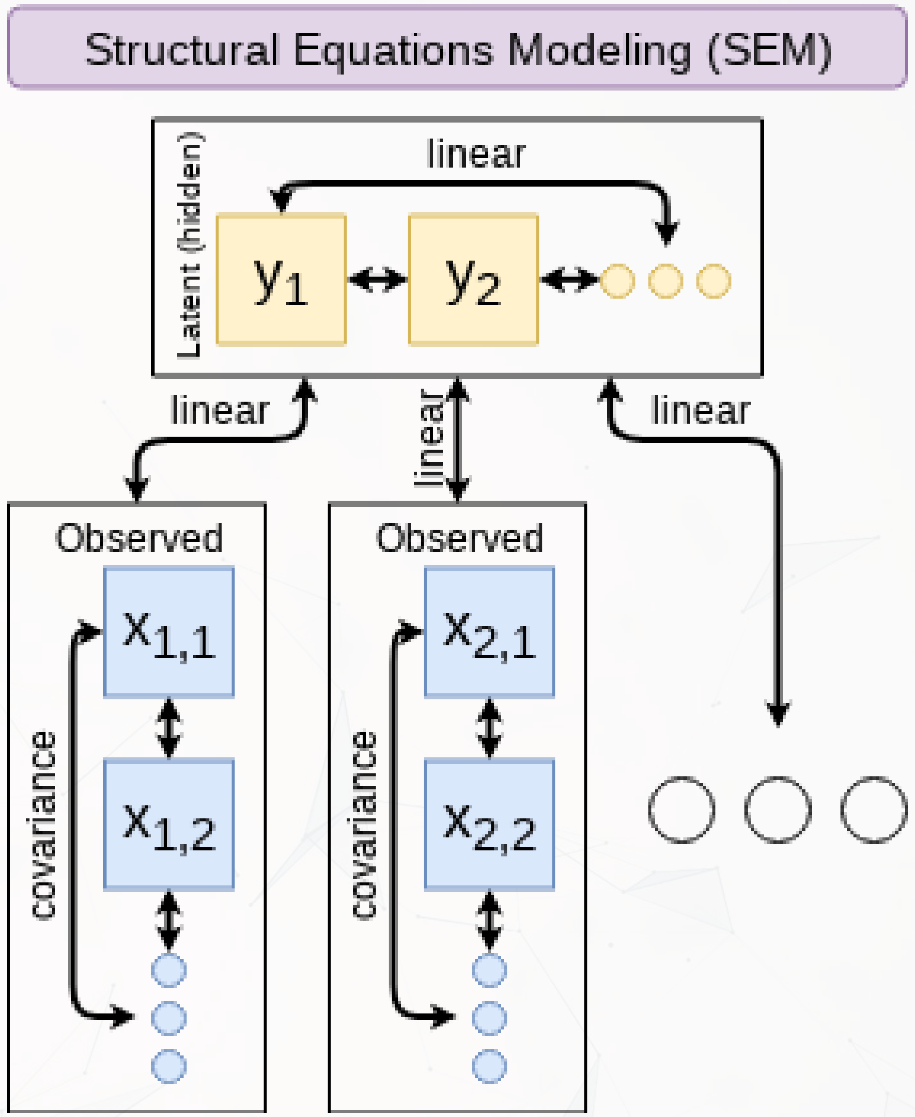


We won't use this in this course, but you should know it exists.

Other linear models: SEM

> SEM can model abstract and *multi-level relationships*

- E.g.: Showing that organizational commitment leads to higher job satisfaction, not the other way around (Poznanski and Bline 1999)



We won't use this in this course, but you should know it exists.

Modeling choices: Model selection

> Pick what fits your problem!

- For forecasting a quantity:
  - Usually some sort of linear model regressed using OLS
  - The other model types mentioned are great for simultaneous forecasting of multiple outputs
- For forecasting a binary outcome:
  - Usually logit or a related model (we'll start this in 2 weeks)
- For forensics:
  - Usually logit or a related model

> There are many more model types though!

Modeling choices: Variable selection

- The options:
  1. Use your own knowledge to select variables
  2. Use a selection model to automate it

Own knowledge

- Build a model based on your knowledge of the problem and situation
- This is generally better
  - The result should be more interpretable
  - For prediction, you should know relationships better than most algorithms

Modeling choices: Automated selection

- Traditional methods include:
  - Forward selection: Start with nothing and add variables with the most contribution to Adj R^2 until it stops going up
  - Backward selection: Start with all inputs and remove variables with the worst (negative) contribution to Adj R^2 until it stops going up
  - Stepwise selection: Like forward selection, but drops non-significant predictors

- Newer methods include:
  - Lasso/Elastic Net based models
    - Optimize with high penalties for complexity (i.e., # of inputs)
    - These are proven to be better
    - We will discuss these in week 6

The overfitting problem

> Or: Why do we like simpler models so much?

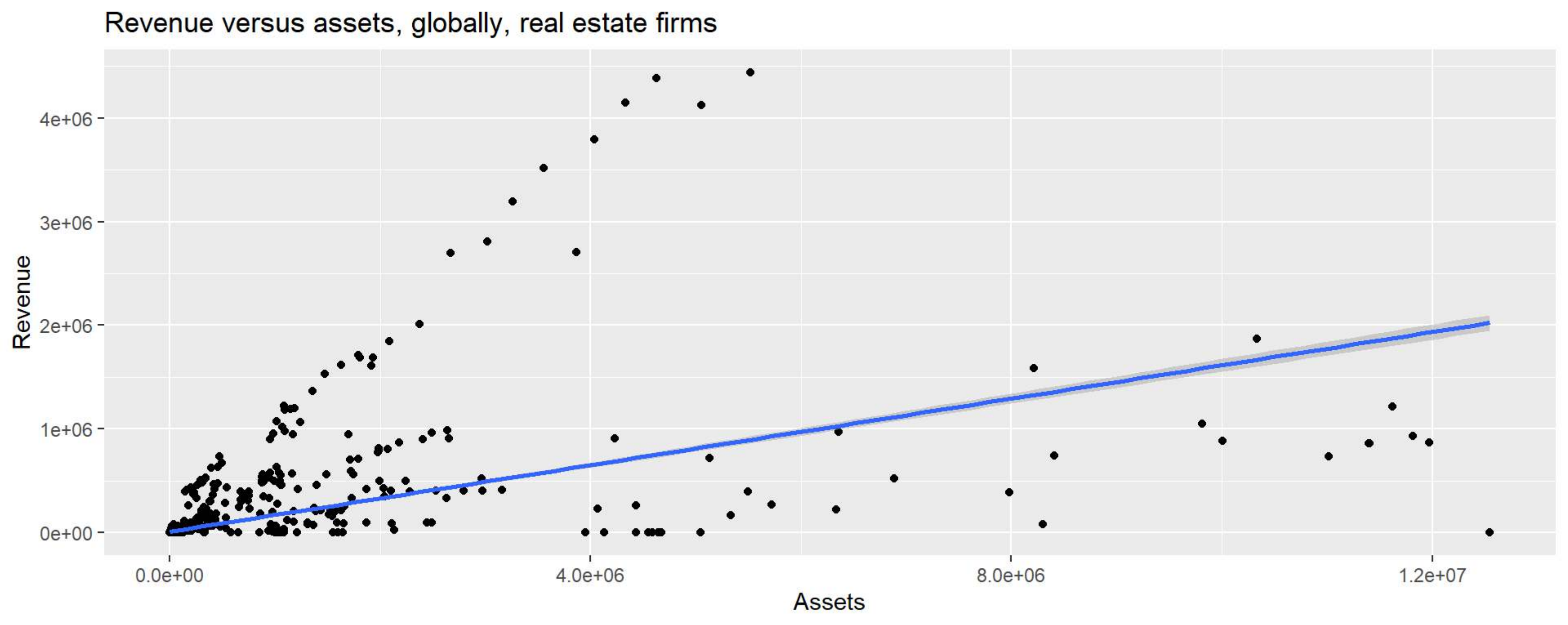
- Overfitting happens when a model fits in-sample data *too well*…
  - To the point where it also models any idiosyncrasies or errors in the data
  - This harms prediction performance
    - Directly harming our forecasts

> An overfitted model works really well on its own data, and quite poorly on new data

Statistical tests and interpretation

Coefficients

- In OLS: β_i
- A change in x_i by 1 leads to a change in y by β_i
- Essentially, the slope between x and y
- The blue line in the chart is the regression line for $\hat{Revenue} = \alpha + \beta_i \hat{Assets}$ for all real estate firms globally, 1994-2021



P-values

- p-values tell us the probability that an individual result is due to random chance

> "The P value is defined as the probability under the assumption of no effect or no difference (null hypothesis), of obtaining a result equal to or more extreme than what was actually observed."
> – Dahiru 2008

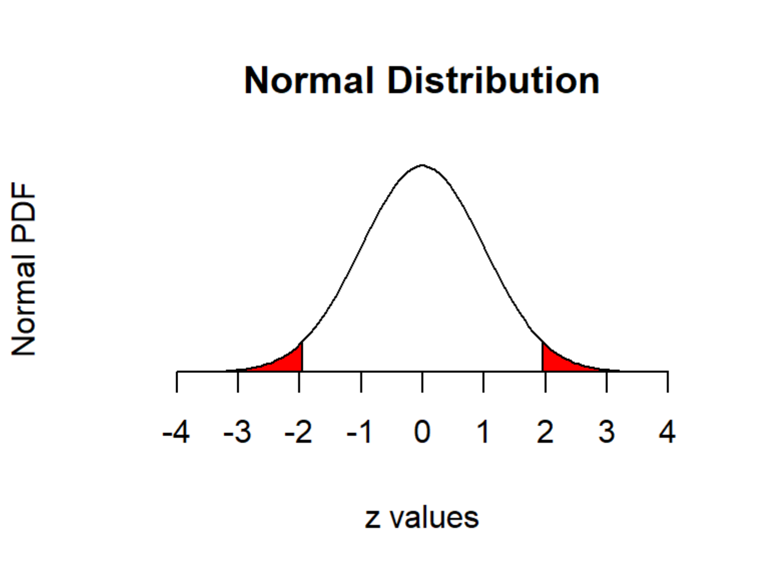
- These are very useful, particularly for a frequentist approach
- First used in the 1700s, but popularized by Ronald Fisher in the 1920s and 1930s

P-values: Rule of thumb

- If $p < 0.05$ and the coefficient sign matches our mental model, we can consider this as supporting our model
    - If $p < 0.05$ but the coefficient is opposite, then it is suggesting a problem with our model
    - If $p > 0.10$, it is rejecting the alternative hypothesis
- If $0.05 < p < 0.10$ it depends…
    - For a small dataset or a complex problem, we can use 0.10 as a cutoff
    - For a huge dataset or a simple problem, we should use 0.05
        - We may even set a lower threshold if we have a ton of data

One vs two tailed tests

- Best practice: **Use a two tailed test** with a p-value cutoff of 0.05 or 0.10
  - 0.05 for easier problems, 0.10 for harder/noisier problems
- Second best practice: use a 1-tailed test with a p-value cutoff of **0.025 or 0.05** - This is mathematically equivalent to the best practice, but roundabout
- Common but generally inappropriate:
  - Use a one tailed test with cutoffs of 0.05 or 0.10 because your hypothesis is directional



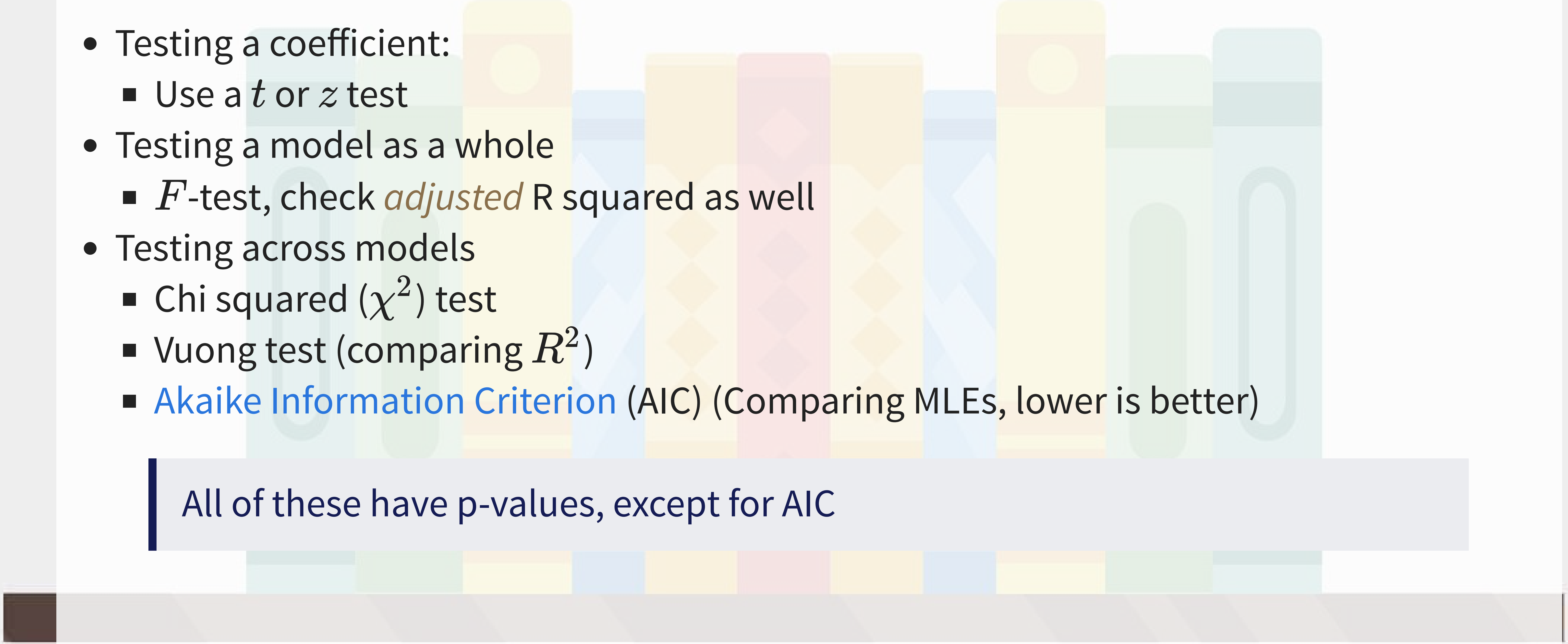
R^2

- R^2 = Explained variation / Total variation
  - Variation = difference in the observed output variable from its own mean
- A high R^2 indicates that the model fits the data very well
- A low R^2 indicates that the model is missing much of the variation in the output
- R^2 is technically a *biased* estimator
- Adjusted R^2 downweights R^2 and makes it unbiased
  - $R^2_{Adj} = PR^2 + 1 - P$
    - Where $P = \frac{n-1}{n-p-1}$
    - n is the number of observations
    - p is the number of inputs in the model

Test statistics

- Testing a coefficient:
  - Use a t or z test
- Testing a model as a whole
  - F-test, check *adjusted* R squared as well
- Testing across models
  - Chi squared (χ^2) test
  - Vuong test (comparing R^2)
  - Akaike Information Criterion (AIC) (Comparing MLEs, lower is better)

> All of these have p-values, except for AIC



Confusion from frequentist approaches

- Possible contradictions:
  - F test says the model is good yet nothing is statistically significant
  - Individual p-values are good yet the model isn't
  - One measure says the model is good yet another doesn't

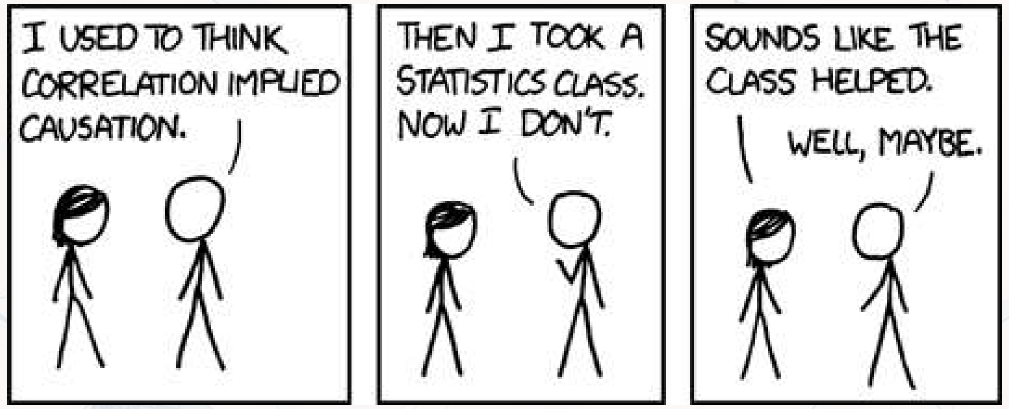
> There are many ways to measure a model, each with their own merits. They don't always agree, and it's on us to pick a reasonable measure.

Causality

What is causality?

$A \rightarrow B$

- Causality is A *causing* B
  - This means more than A and B are correlated
- I.e., If A changes, B changes. But B changing doesn't mean A changed
  - Unless B is 100% driven by A
- Very difficult to determine, particularly for events that happen [almost] simultaneously
- Examples of correlations that aren't causation


I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

Time and causality

$$A \rightarrow B \text{ or } A \leftarrow B?$$
$$A_t \rightarrow B_{t+1}$$

- If there is a separation in time, it's easier to say A caused B
  - Observe A, then see if B changes after
- Conveniently, we have this structure when forecasting
  - Consider a model like:

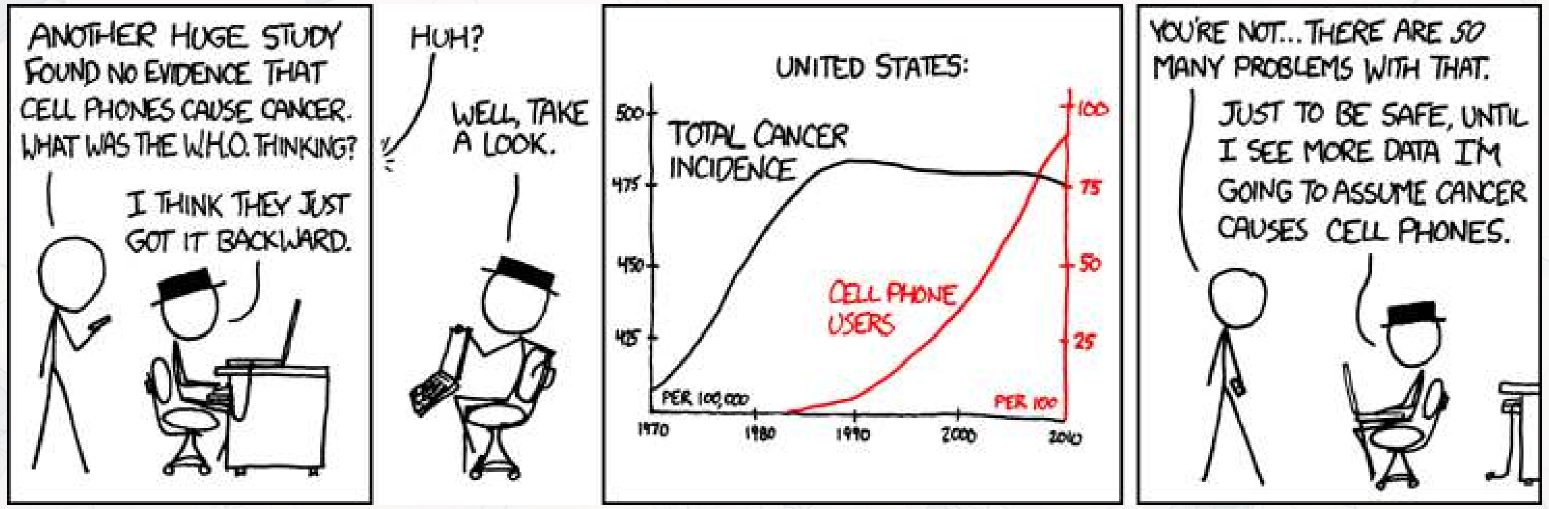
$$Revenue_{t+1} = Revenue_t + \ldots$$

> It would be quite difficult for $Revenue_{t+1}$ to cause $Revenue_t$

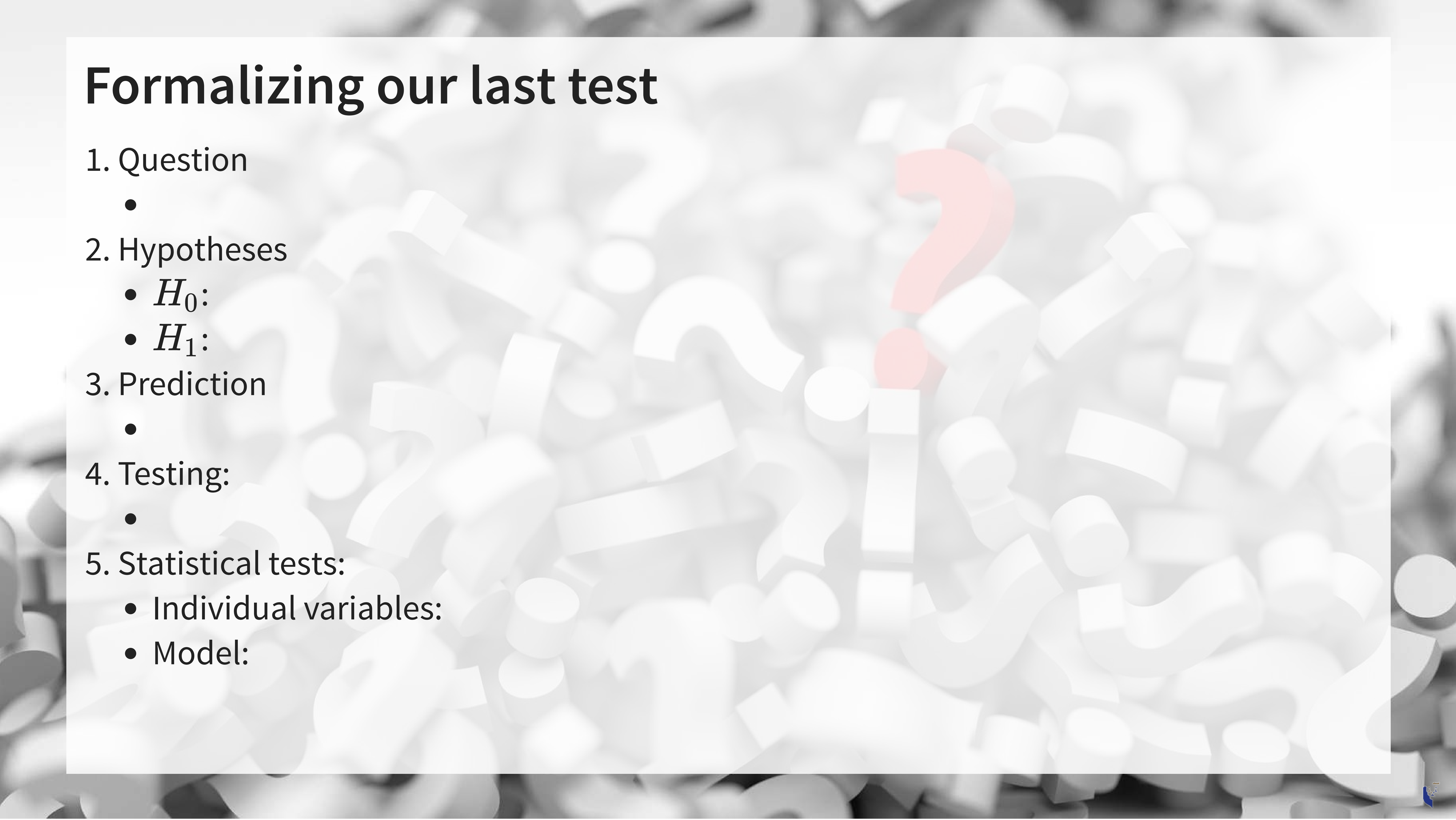
Time and causality break down

$$A_t \rightarrow B_{t+1}? \quad \text{OR} \quad C \rightarrow A_t \text{ and } C \rightarrow B_{t+1}?$$

- The above illustrates the *Correlated omitted variable problem*
  - A doesn’t cause B… Instead, some other force C causes both
  - This is the bane of social scientists everywhere
- It is less important for *predictive* analytics, as we care more about performance, but…
  - It can complicate interpreting your results
  - Figuring out C can help improve you model’s predictions (So find C!)


ANOTHER HUGE STUDY FOUND NO EVIDENCE THAT CELL PHONES CAUSE CANCER. WHAT WAS THE W.H.O. THINKING?
I THINK THEY JUST GOT IT BACKWARD.
HUH?
WELL, TAKE A LOOK.
UNITED STATES:
TOTAL CANCER INCIDENCE
CELL PHONE USERS
500
475
450
425
PER 100,000
100
75
50
25
PER 100
1970
1980
1990
2000
2010
YOU'RE NOT... THERE ARE SO MANY PROBLEMS WITH THAT.
JUST TO BE SAFE, UNTIL I SEE MORE DATA I'M GOING TO ASSUME CANCER CAUSES CELL PHONES.

Revisiting the previous problem



Formalizing our last test

1. Question
   -
2. Hypotheses
   - H_0:
   - H_1:
3. Prediction
   -
4. Testing:
   -
5. Statistical tests:
   - Individual variables:
   - Model:

Is this model better?

```r
anova(mod2, mod3, test="Chisq")
```

```
Analysis of Variance Table

Model 1: revt_growth ~ at_growth
Model 2: revt_growth ~ act_growth + che_growth + lct_growth
  Res.Df    RSS Df Sum of Sq Pr(>Chi)  
1     25 1.5580                        
2     23 1.2344  2   0.32359  0.04906 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

> A bit better at $p < 0.05$

- This means our model with change in current liabilities, cash, and EBIT appears to be better than the model with change in assets.

Scaling up

Expanding our methodology

- Why should we limit ourselves to 1 firm's data?
- The nature of data analysis is such:

> Adding more data usually helps improve predictions

- Assuming:
  - The data isn't of low quality (too noisy)
  - The data is relevant
  - Any differences can be reasonably controlled for

Fine tuning our question

- Previously: Can we predict revenue using a firm's accounting information?

⚠ **Problems with our original question**

1. We were using simultaneous $\hat{Y}$ and $\hat{X}$ variables
   - Thus, it was not forecasting
2. Simultaneous accounting data is very dependent/correlated
   - We were violating OLS regression assumptions

- Now: Can we predict *future* revenue using a firm's accounting information?
  - What do we need to change? $\hat{y}$ will need to be 1 year in the future

What this revised question does better

1. It is a proper prediction problem
   - We are using old data to predict a new outcome
2. We don't need to worry much about dependence

First things first

- When using a lot of data, it is important to make sure the data is clean
- In our case, we may want to remove any very small firms

```r
# Ensure firms have at least $1M (local currency), and have revenue
# df_full contains all real estate companies excluding North America
df_clean <- df_full %>%
  filter(at>1, revt>0)

# We cleaned out 596 observations!
print(c(nrow(df_full), nrow(df_clean)))
```

```
[1] 6152 5556
```

```r
# Another useful cleaning function:
# Replaces NaN, Inf, and -Inf with NA for all numeric variables in the data!
df_clean <- df_clean %>%
  mutate(across(where(is.numeric), ~replace(., !is.finite(.), NA)))
```

Looking back at the prior models

```r
uol <- uol %>% mutate(revt_lead = lead(revt))  # From dplyr
forecast1 <- lm(revt_lead ~ act + che + lct, data=uol)
library(broom)  # Lets us view bigger regression outputs in a tidy fashion
tidy(forecast1)  # Present regression output
```

```
# A tibble: 4 × 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept) 235.        139.        1.69  0.104
2 act           0.548       0.145     3.77  0.000999
3 che          -0.181       0.322    -0.561 0.580
4 lct          -0.0700      0.242    -0.289 0.775
```

```r
glance(forecast1)  # Present regression statistics
```

```
# A tibble: 1 × 12
  r.squared adj.r.squared sigma statistic       p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>         <dbl> <dbl>  <dbl> <dbl> <dbl>
1     0.826         0.803  337.      36.4 0.00000000675     3  -193.  397.  403.
# ℹ 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

> This model is ok, but we can do better.

Expanding the prior model

```r
forecast2 <-
  lm(revt_lead ~ revt + act + che + lct + dp + ebit , data=uol)
tidy(forecast2)
```

```
# A tibble: 7 × 5
  term        estimate std.error statistic   p.value
  <chr>          <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)  148.      119.         1.24  0.228
2 revt           1.63      0.311      5.22  0.0000414
3 act            0.317     0.165      1.92  0.0687
4 che            0.124     0.322      0.384 0.705
5 lct           -0.189     0.193     -0.981 0.338
6 dp            -3.66      3.39      -1.08  0.293
7 ebit          -3.63      0.995     -3.65  0.00159
```

- Revenue to capture stickiness of revenue
- Current assets & Cash (and equivalents) to capture asset base
- Current liabilities to capture payments due
- Depreciation to capture decrease in real estate asset values
- EBIT to capture operational performance

Expanding the prior model

```r
glance(forecast2)
```

```
# A tibble: 1 × 12
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
1     0.929         0.907  231.      43.4 1.97e-10     6  -181.  379.  389.
# ℹ 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

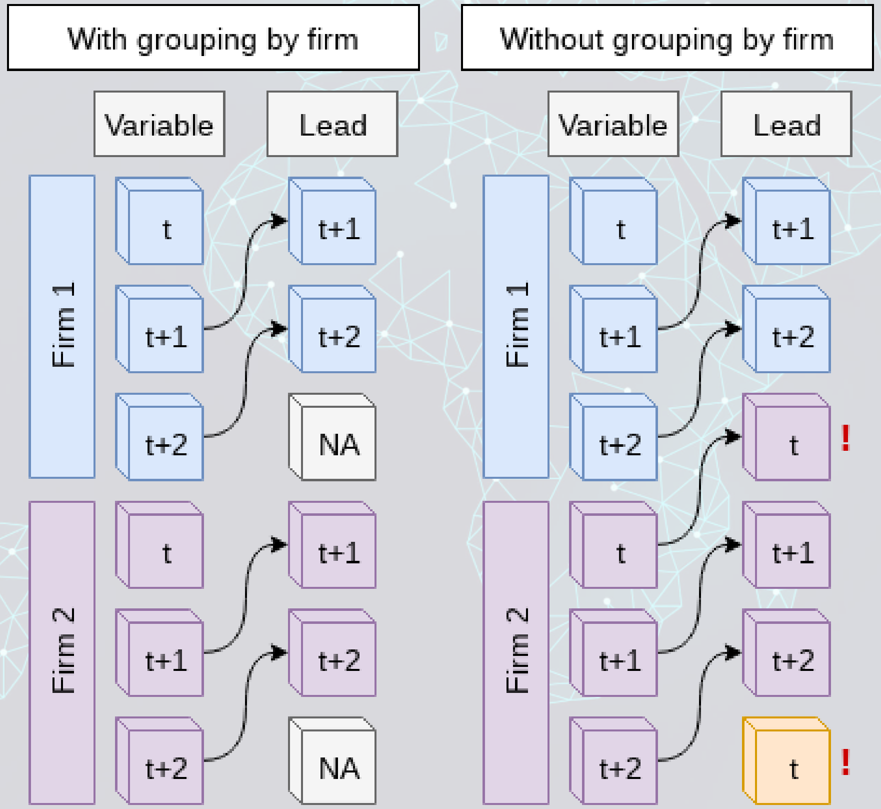
```r
anova(forecast1, forecast2, test="Chisq")
```

```
Analysis of Variance Table

Model 1: revt_lead ~ act + che + lct
Model 2: revt_lead ~ revt + act + che + lct + dp + ebit
  Res.Df     RSS Df Sum of Sq  Pr(>Chi)
1     23 2616067
2     20 1071637  3   1544429 2.439e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

> This is better (Adj. R^2, χ^2, AIC).

All Singapore real estate companies

```r
# Note the group_by -- without it, lead() will pull from the subsequent firm!
# ungroup() tells R that we finished grouping
df_clean <- df_clean %>%
  group_by(isin) %>%
  mutate(revt_lead = lead(revt)) %>%
  ungroup()
```



All Singapore real estate companies

```r
forecast3 <-
  lm(revt_lead ~ revt + act + che + lct + dp + ebit,
     data=df_clean[df_clean$fic=="SGP",])
tidy(forecast3)
```

```
# A tibble: 7 × 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   0.0134    7.86     0.00170 9.99e- 1
2 revt          0.652     0.0555  11.7     2.00e-27
3 act           0.154     0.0306   5.03    7.48e- 7
4 che           0.234     0.0807   2.90    3.98e- 3
5 lct           0.0768    0.0575   1.34    1.82e- 1
6 dp            1.63      0.748    2.17    3.04e- 2
7 ebit         -0.802     0.206   -3.90    1.15e- 4
```

All Singapore real estate companies

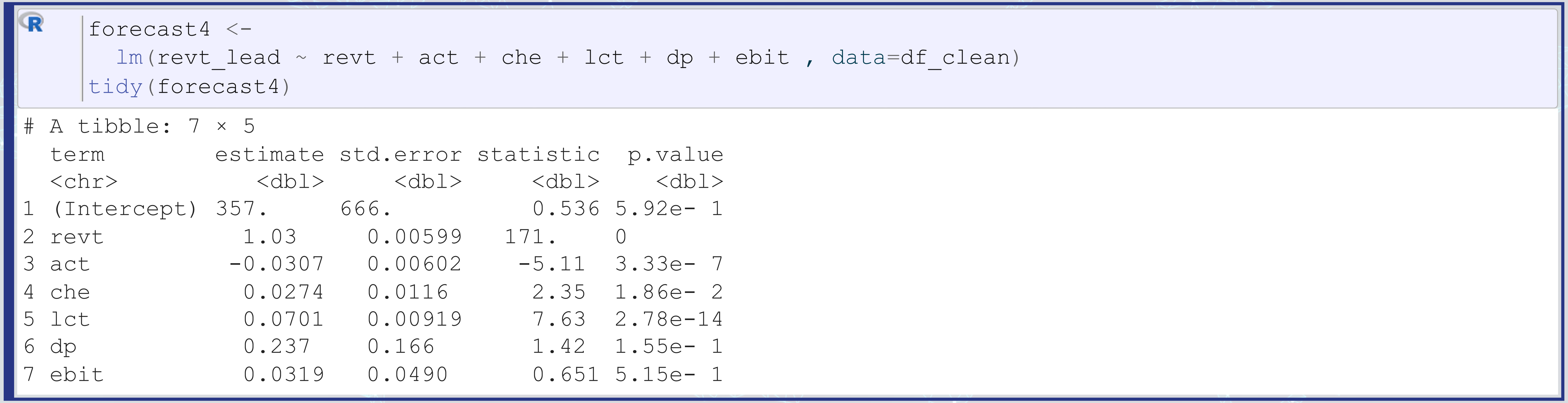
```r
glance(forecast3)
```

```
# A tibble: 1 × 12
  r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>  <dbl> <dbl> <dbl>
1     0.884         0.883  123.      488. 5.63e-176     6 -2427. 4870. 4902.
# ℹ 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

> Lower adjusted R^2 – This is worse? Why?

- Note: χ^2 can only be used for models on the same data
  - Same for AIC

Worldwide real estate companies



```r
forecast4 <-
  lm(revt_lead ~ revt + act + che + lct + dp + ebit , data=df_clean)
tidy(forecast4)
```

```
# A tibble: 7 × 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept) 357.      666.         0.536 5.92e- 1
2 revt          1.03      0.00599  171.    0
3 act          -0.0307    0.00602   -5.11  3.33e- 7
4 che           0.0274    0.0116     2.35  1.86e- 2
5 lct           0.0701    0.00919    7.63  2.78e-14
6 dp            0.237     0.166      1.42  1.55e- 1
7 ebit          0.0319    0.0490     0.651 5.15e- 1
```

Worldwide real estate companies

```r
glance(forecast4)
```

```
# A tibble: 1 × 12
  r.squared adj.r.squared  sigma statistic p.value    df  logLik     AIC     BIC
      <dbl>         <dbl>  <dbl>     <dbl>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>
1     0.948         0.948 46353.    15089.       0     6 -60617. 121249. 121302.
# ℹ 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

> Higher adjusted R^2 – better!

- Note: χ^2 can only be used for models on the same data
  - Same for AIC

Model accuracy

> Why is the UOL model better than the Singapore model?

- Ranking:
  1. Worldwide real estate model
  2. UOL model
  3. Singapore real estate model

Practice: group_by()

- This practice is to make sure you understand how to use mutate with leads and lags when there are multiple companies in the data
    - We’ll almost always work with multiple companies!
- Do exercises 2 and 3 on today’s R practice file:
    - R Practice
    - Shortlink: rmc.link/420r2

Expanding our problem with macro data

Macro data sources

- For Singapore: Data.gov.sg
  - Covers: Economy, education, environment, finance, health, infrastructure, society, technology, transport
- For real estate in Singapore: URA’s REALIS system
  - Access through the library
- WRDS has some as well
- For US: data.gov, as well as many agency websites
  - Like BLS or the Federal Reserve

Wharton UNIVERSITY of PENNSYLVANIA wrds WHARTON RESEARCH DATA SERVICES

DATA.GOV

Our Macro data

```r
library(tidyverse)
# First column, first 10 rows...
read_csv("../../Data/Session_2-Macro.csv") %>%
  .[1:10, 1] %>%
  DT::datatable()
```

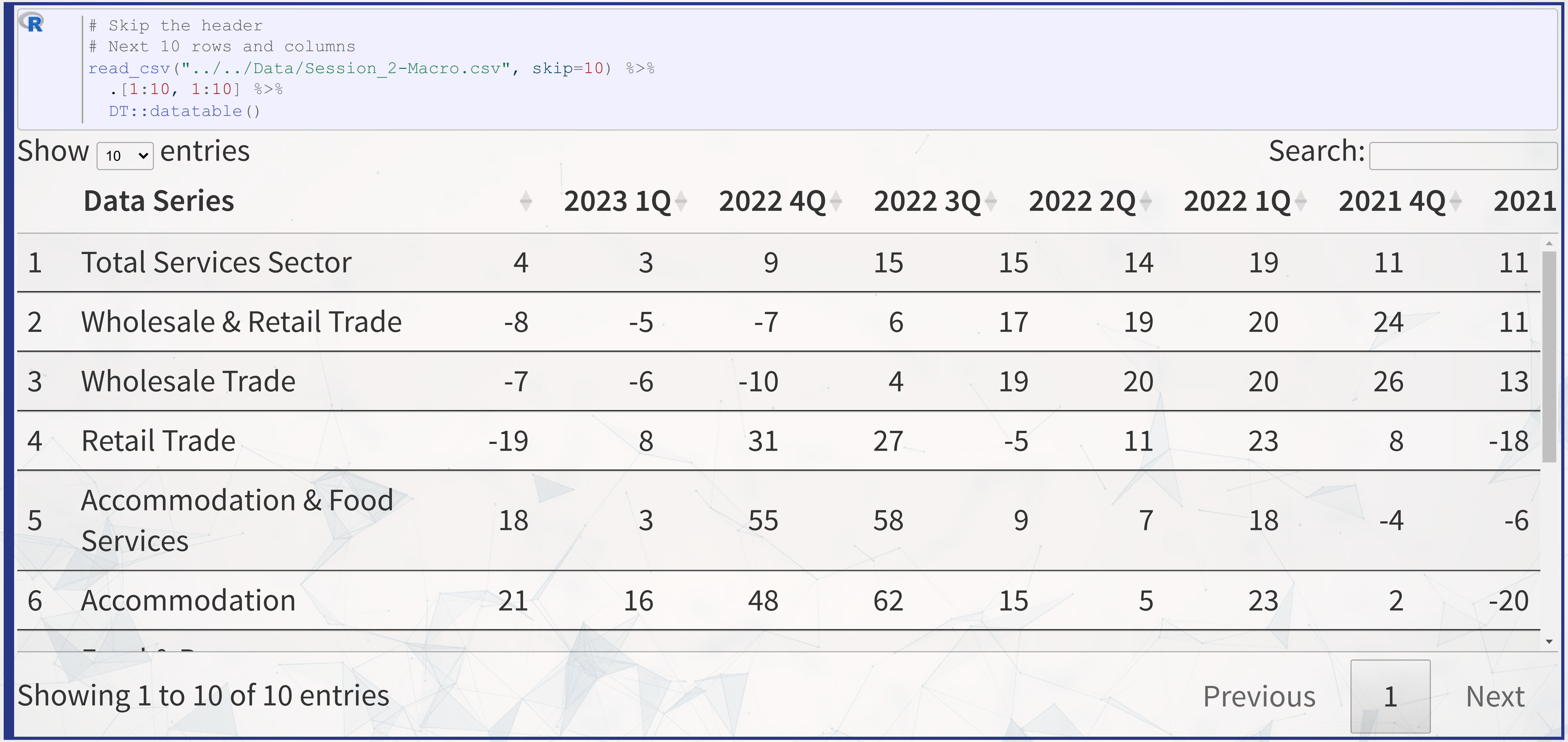
Show 10 entries Search:

	...1
1	Theme: Industry
2	Subject: Business Expectations
3	Topic: Services Sector
4	Table Title: Business Expectations For The Services Sector - General Business Outlook For The Next 6 Months, Net Weighted Balance, Quarterly
5	
6	Data last updated: 28/04/2023
7	Source: SINGAPORE DEPARTMENT OF STATISTICS

Showing 1 to 10 of 10 entries

Previous 1 Next

Our macro data



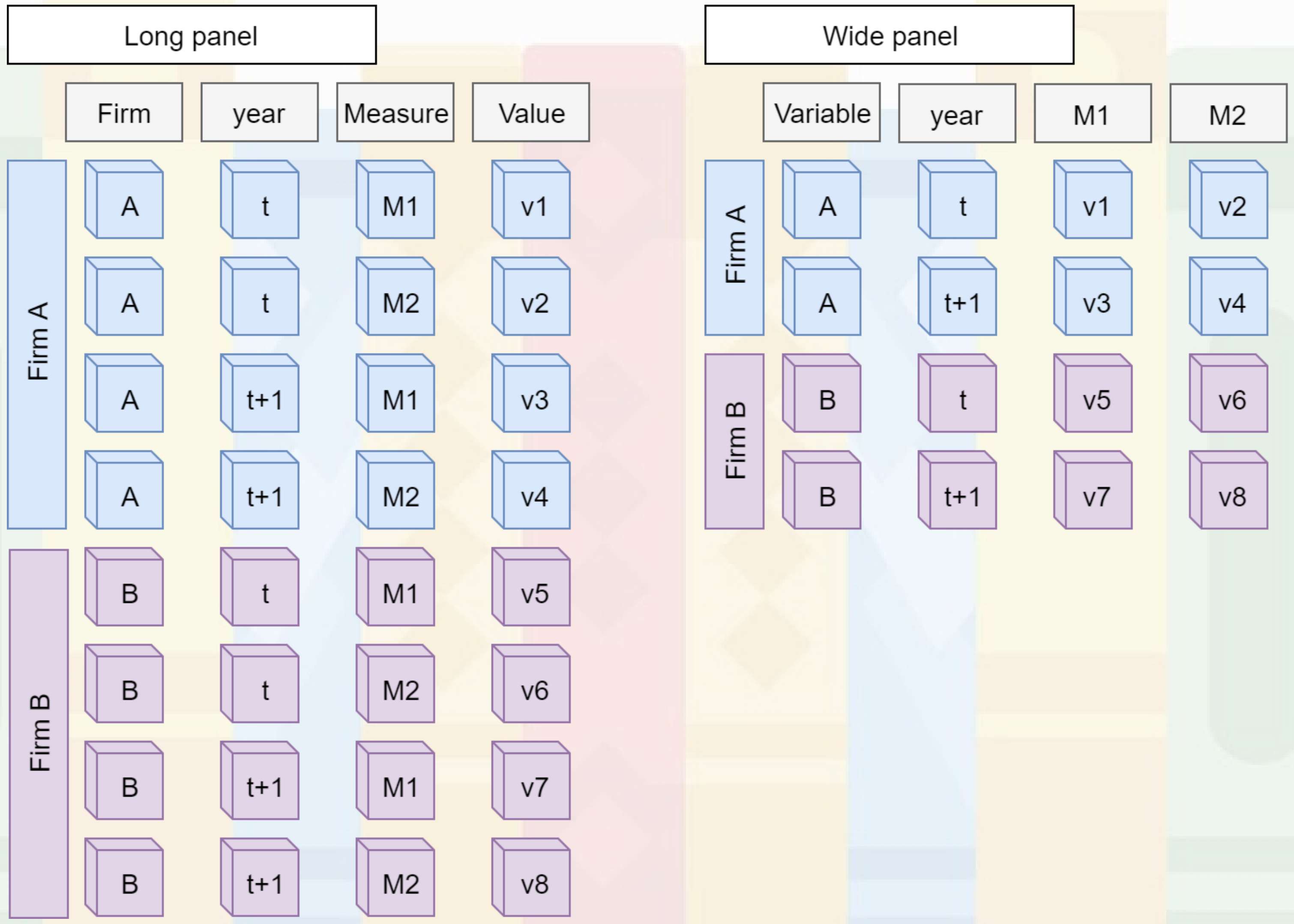
```r
# Skip the header
# Next 10 rows and columns
read_csv("../../Data/Session_2-Macro.csv", skip=10) %>%
  .[1:10, 1:10] %>%
  DT::datatable()
```

	Data Series		2023 1Q	2022 4Q	2022 3Q	2022 2Q	2022 1Q	2021 4Q	2021
1	Total Services Sector	4	3	9	15	15	14	19	11	11
2	Wholesale & Retail Trade	-8	-5	-7	6	17	19	20	24	11
3	Wholesale Trade	-7	-6	-10	4	19	20	20	26	13
4	Retail Trade	-19	8	31	27	-5	11	23	8	-18
5	Accommodation & Food Services	18	3	55	58	9	7	18	-4	-6
6	Accommodation	21	16	48	62	15	5	23	2	-20

Panel data

- Panel data refers to data with the following characteristics:
  - There is a time dimension
  - There is at least 1 other dimension to the data (firm, country, etc.)
- Special cases:
  - A panel where all dimensions have the same number of observations is called *balanced*
    - Otherwise we call it *unbalanced*
  - A panel missing the time dimension is *cross-sectional*
  - A panel missing the other dimension(s) is a *time series*
- Format:
  - Long: Indexed by all dimensions
  - Wide: Indexed only by some dimensions

Panel data



Long panel

	Firm	year	Measure	Value
Firm A	A	t	M1	v1
	A	t	M2	v2
	A	t+1	M1	v3
	A	t+1	M2	v4
Firm B	B	t	M1	v5
	B	t	M2	v6
	B	t+1	M1	v7
	B	t+1	M2	v8

Wide panel

	Variable	year	M1	M2
Firm A	A	t	v1	v2
	A	t+1	v3	v4
Firm B	B	t	v5	v6
	B	t+1	v7	v8

> Data frames are usually wide panels

Loading macro data

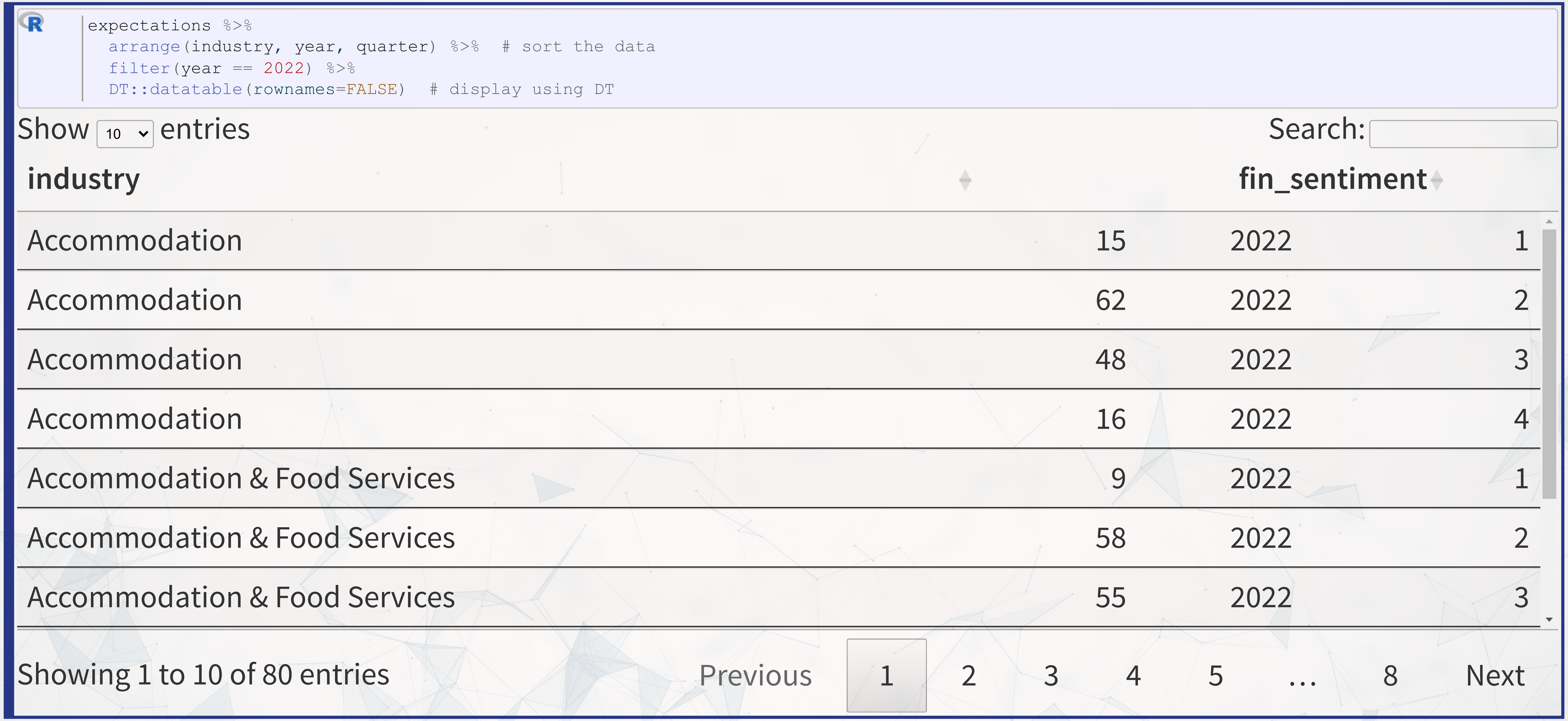
- Singapore business expectations data (from SingStat

```r
expectations <- read_csv("../../Data/Session_2-Macro.csv",
                         skip=10, na="na") %>%          # Needed to load file
  filter(row_number() < 21) %>%                          # Drop the footer
  rename(industry=`Data Series`) %>%                     # Rename column
  pivot_longer(!industry, names_to='yearQ',
               values_to='fin_sentiment') %>%          # Cast wide to long
  mutate(year = as.numeric(substr(yearQ, 1, 4))) %>%    # split out year
  mutate(quarter = as.numeric(substr(yearQ, 6, 6))) %>% # split out quarter
  select(-yearQ)                                         # Remove measure

# extract out Q1, finance only
expectations_re <- expectations %>%
  filter(quarter == 1,                               # Keep only the Q1
         industry == "Real Estate")                  # Keep only real estate
```

Casting between data frame shapes

The `pivot_wider()` and `pivot_longer()` functions work well. See the dplyr documentation for more details. In the code above, the first argument is the columns to turn into rows. `!industry` means all columns except `industry`. `names_to` (`value_to`) specifies the variable name to contain the column names (data) after transforming to long.

What was in the macro data?



```r
expectations %>%
  arrange(industry, year, quarter) %>%  # sort the data
  filter(year == 2022) %>%
  DT::datatable(rownames=FALSE)  # display using DT
```

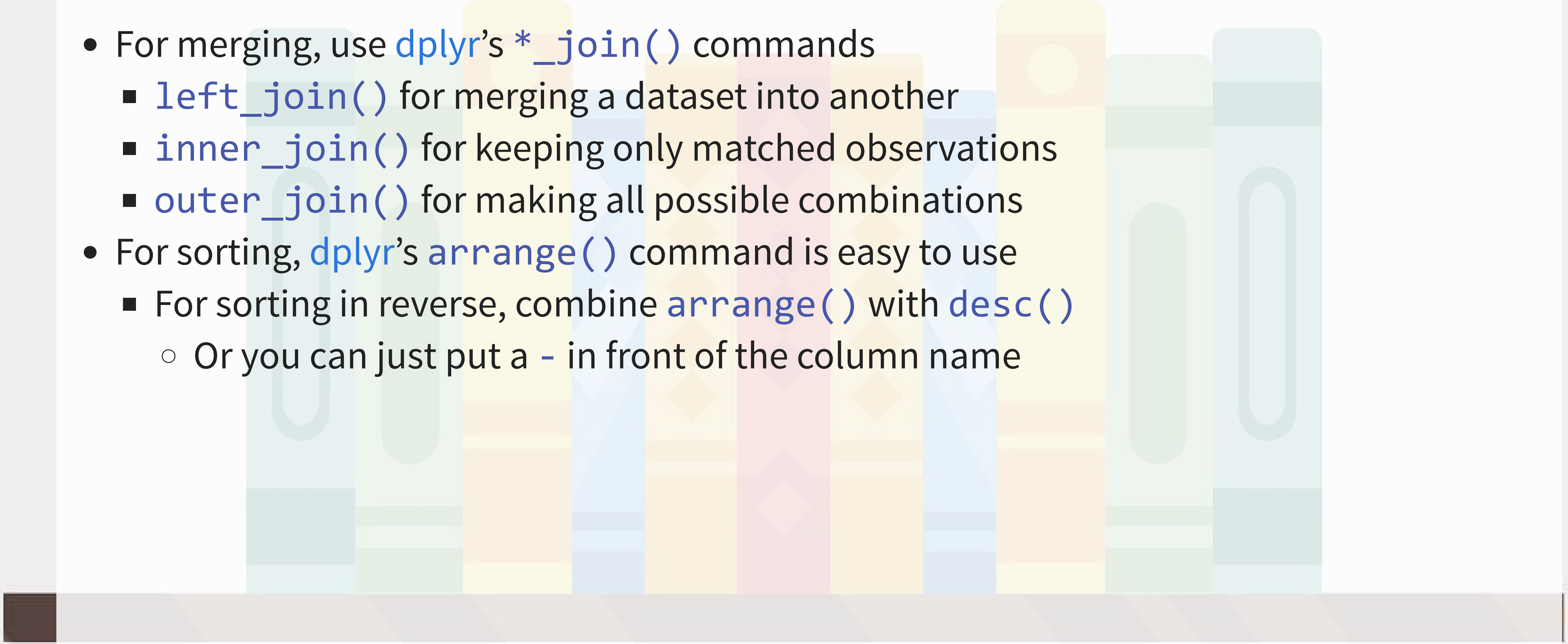
Show 10 entries

Search:

industry		fin_sentiment	
Accommodation	15	2022	1
Accommodation	62	2022	2
Accommodation	48	2022	3
Accommodation	16	2022	4
Accommodation & Food Services	9	2022	1
Accommodation & Food Services	58	2022	2
Accommodation & Food Services	55	2022	3

Showing 1 to 10 of 80 entries

dplyr makes merging easy



- For merging, use dplyr’s `*_join()` commands
  - `left_join()` for merging a dataset into another
  - `inner_join()` for keeping only matched observations
  - `outer_join()` for making all possible combinations
- For sorting, dplyr’s `arrange()` command is easy to use
  - For sorting in reverse, combine `arrange()` with `desc()`
    - Or you can just put a `-` in front of the column name

Merging example

> Merge in the finance sentiment data to our accounting data

```r
# subset out our Singaporean data, since our macro data is Singapore-specific
df_SG <- df_clean %>% filter(fic == "SGP")

# Create year in df_SG (date is given by datadate as YYYYMMDD)
df_SG$year = round(df_SG$datadate / 10000, digits=0)

# Combine datasets
# Notice how it automatically figures out to join by "year"
df_SG_macro <- left_join(df_SG, expectations_re[,c("year","fin_sentiment")])
```

Predicting with macro data

Building in macro data

- First try: Just add it in

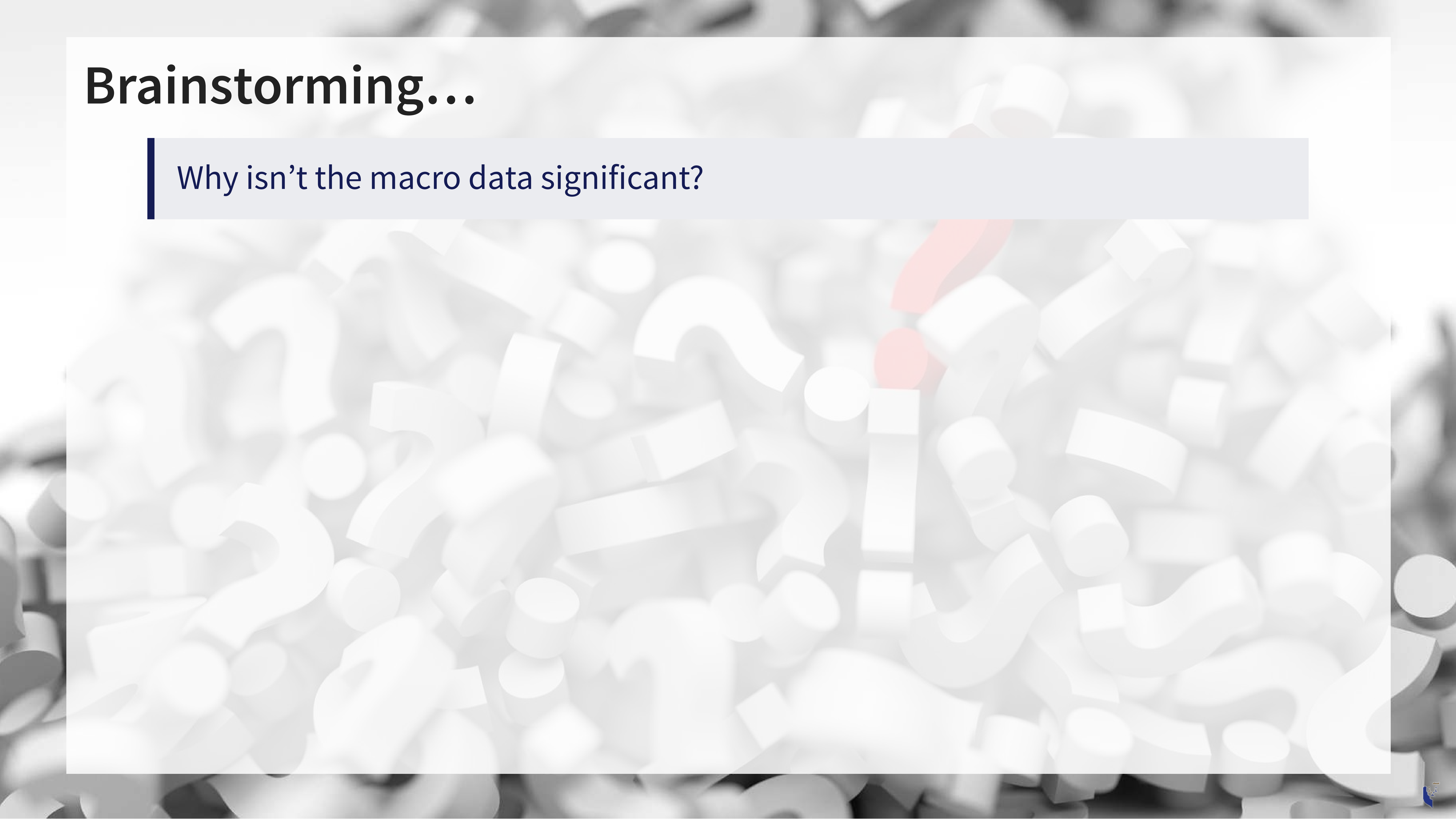
```r
macro1 <- lm(revt_lead ~ revt + act + che + lct + dp + ebit + fin_sentiment,
             data=df_SG_macro)
library(broom)
tidy(macro1)
```

```
# A tibble: 8 × 5
  term          estimate std.error statistic  p.value
  <chr>            <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)     0.119     8.00      0.0149 9.88e- 1
2 revt            0.652     0.0563   11.6    1.01e-26
3 act             0.155     0.0316    4.90   1.41e- 6
4 che             0.231     0.0823    2.81   5.23e- 3
5 lct             0.0755    0.0582    1.30   1.96e- 1
6 dp              1.63      0.761     2.15   3.25e- 2
7 ebit           -0.804     0.208    -3.86   1.35e- 4
8 fin_sentiment   0.0174    0.177     0.0980 9.22e- 1
```

> It isn’t significant. Why is this?

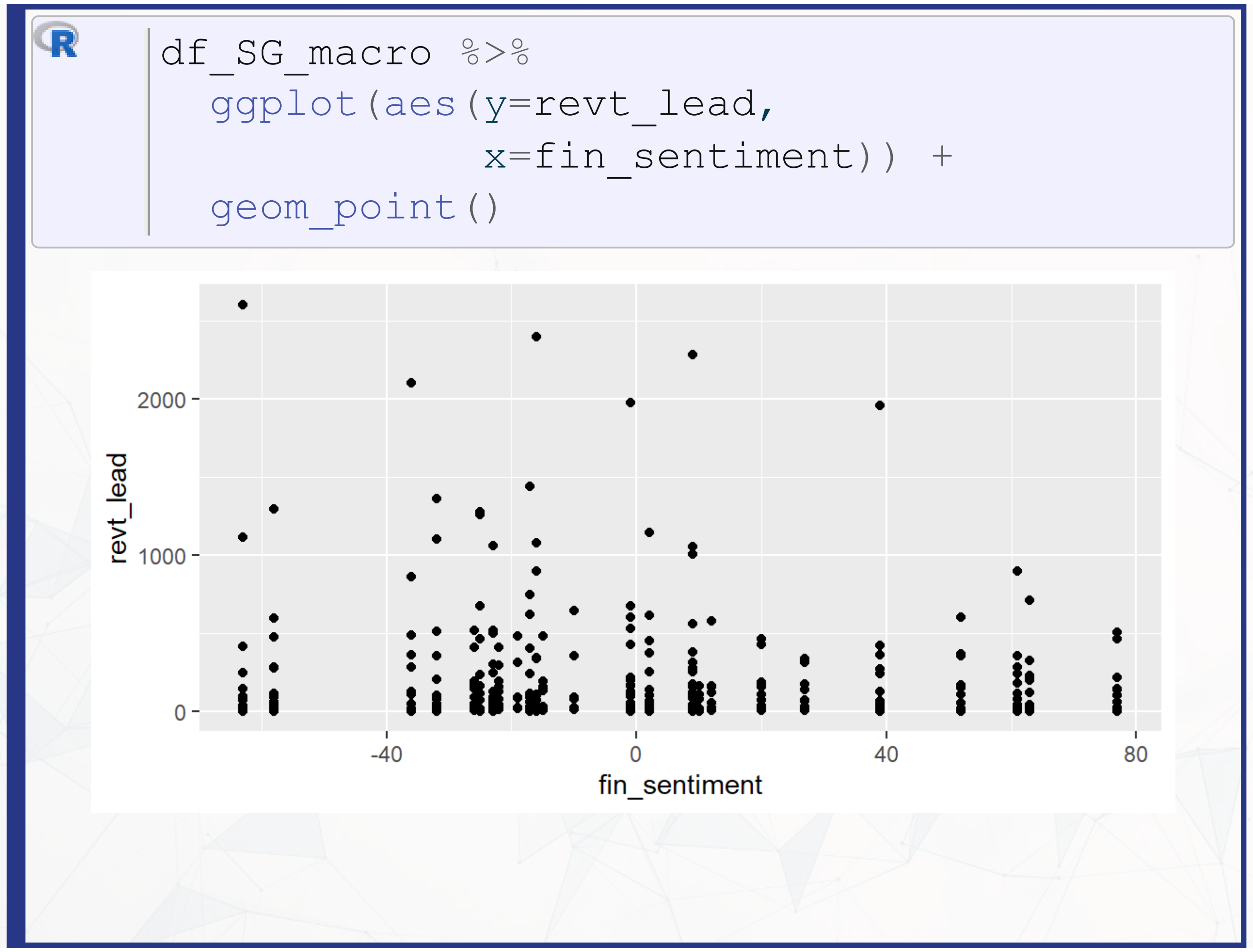
Brainstorming…

> Why isn’t the macro data significant?

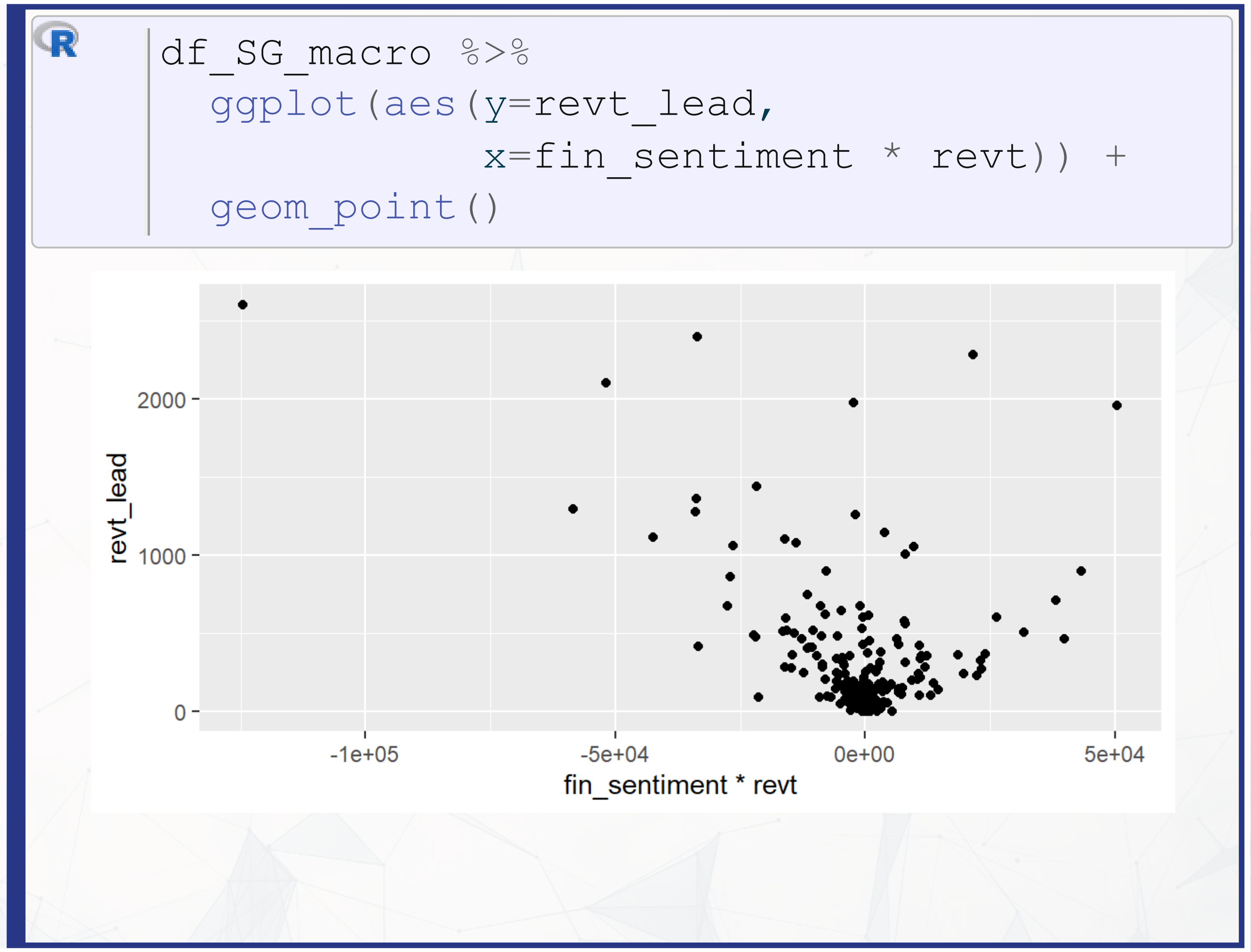


Scaling matters

- All of our firm data is on the same terms as revenue: dollars within a given firm
- But `fin_sentiment` is a constant scale…
  - We need to scale this to fit the problem
    - The current scale would work for revenue growth



```
df_SG_macro %>%
  ggplot(aes(y=revt_lead,
             x=fin_sentiment)) +
  geom_point()
```



```
df_SG_macro %>%
  ggplot(aes(y=revt_lead,
             x=fin_sentiment * revt)) +
  geom_point()
```

Scaled macro data

- Scale by revenue

```r
macro3 <-
  lm(revt_lead ~ revt + act + che + lct + dp + ebit + fin_sentiment:revt,
     data=df_SG_macro)
tidy(macro3)
```

```
# A tibble: 8 × 5
  term                 estimate std.error statistic  p.value
  <chr>                   <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)           1.83     7.91         0.231 8.18e- 1
2 revt                  0.655    0.0556      11.8   1.63e-27
3 act                   0.133    0.0316       4.21  3.21e- 5
4 che                   0.267    0.0821       3.25  1.24e- 3
5 lct                   0.0619   0.0577       1.07  2.84e- 1
6 dp                    1.94     0.757        2.57  1.06e- 2
7 ebit                 -0.804    0.206       -3.90  1.12e- 4
8 revt:fin_sentiment -0.00175  0.000596    -2.94  3.51e- 3
```

```r
glance(macro3)
```

```
# A tibble: 1 × 12
  r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>  <dbl> <dbl> <dbl>
1     0.887         0.885  123.      421. 1.28e-173     7 -2388. 4794. 4830.
# ℹ 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Model comparisons

```r
# Ensure that we use the same data (fin_sentiment is missing in 1994)
baseline <-
  lm(revt_lead ~ revt + act + che + lct + dp + ebit,
     data=df_SG_macro[!is.na(df_SG_macro$fin_sentiment),])
glance(baseline)
```

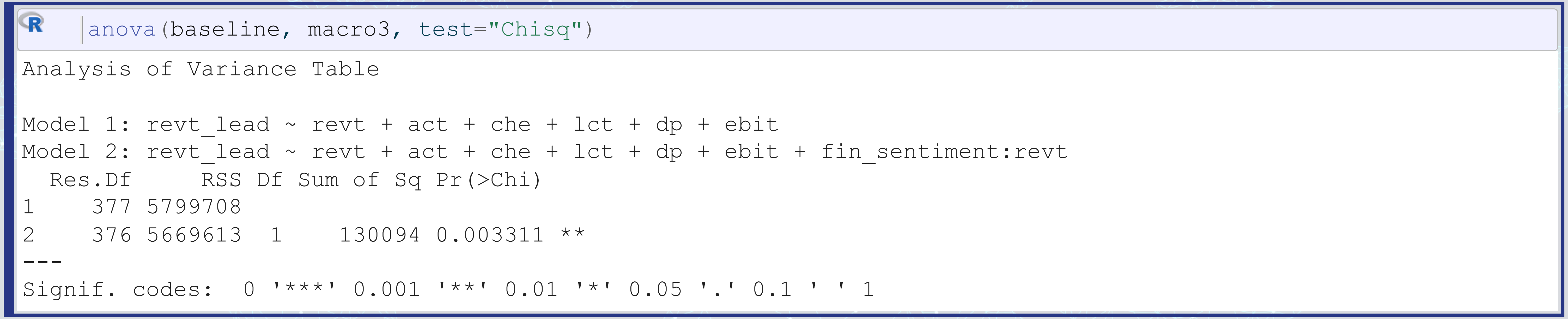
```
# A tibble: 1 × 12
  r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>  <dbl> <dbl> <dbl>
1     0.884         0.882  124.      480. 3.97e-173     6 -2392. 4801. 4832.
# ℹ 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
glance(macro3)
```

```
# A tibble: 1 × 12
  r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>  <dbl> <dbl> <dbl>
1     0.887         0.885  123.      421. 1.28e-173     7 -2388. 4794. 4830.
# ℹ 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

> Adjusted R^2 and AIC are slightly better with macro data

Model comparisons



```
anova(baseline, macro3, test="Chisq")
```

```
Analysis of Variance Table

Model 1: revt_lead ~ revt + act + che + lct + dp + ebit
Model 2: revt_lead ~ revt + act + che + lct + dp + ebit + fin_sentiment:revt
  Res.Df     RSS Df Sum of Sq Pr(>Chi)
1    377 5799708
2    376 5669613  1    130094 0.003311 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

> Macro model definitely fits better than the baseline model!

Takeaway

1. Adding macro data can help explain some exogenous variation in a model
    - Exogenous meaning outside of the firms, in this case
2. Scaling is very important
    - Not scaling properly can suppress some effects from being visible

> Interpretating the macro variable

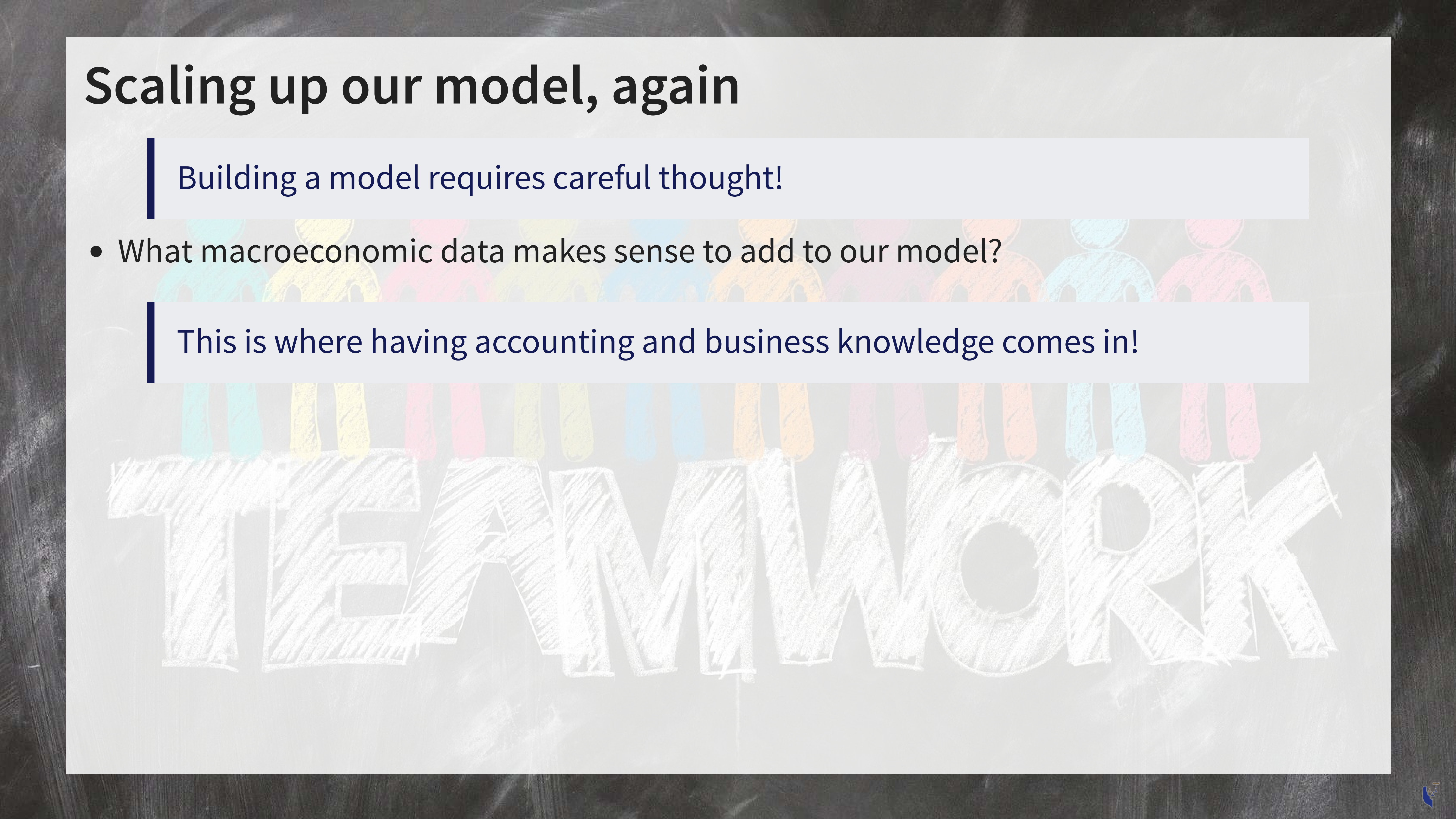
- All else equal, the average firm has revenue stickiness of 65.55%
- For every 1 S.D. increase in `fin_sentiment` (36.1 points)
    - Revenue stickiness changes by ~-6.32%
- Over the range of sentiment data (-63 to 77)…
    - Revenue stickiness changes from +11.04% to -13.49%

Scaling up our model, again

> Building a model requires careful thought!

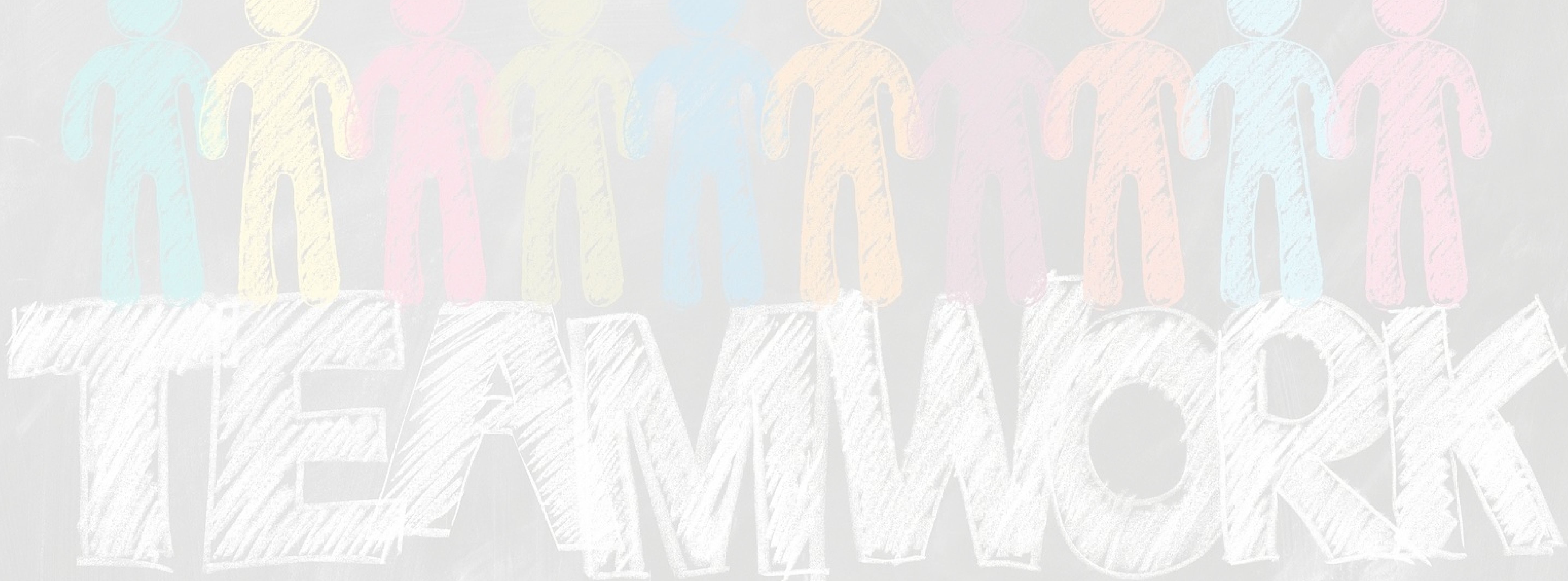
- What macroeconomic data makes sense to add to our model?

> This is where having accounting and business knowledge comes in!



Brainstorming…

> What other macro data would you like to add to this model?



Validation: Is it better?

Validation

- Ideal:
  - Withhold the last year (or a few) of data when building the model
  - Check performance on *hold out sample*
  - This is *out of sample* testing
- Sometimes acceptable:
  - Withhold a random sample of data when building the model
  - Check performance on *hold out sample*

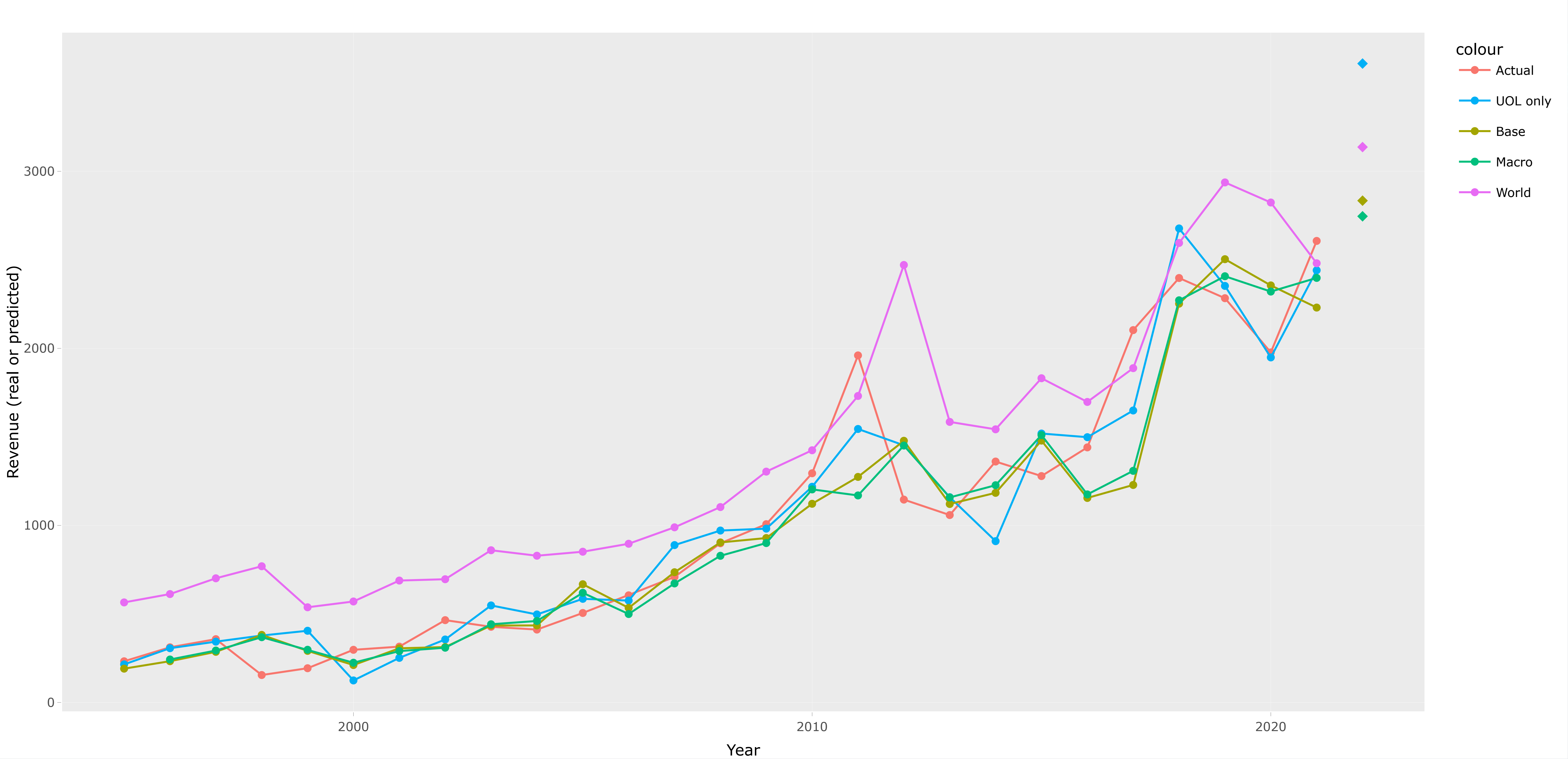
Estimation

- As we never constructed a hold out sample, let’s end by estimating UOL’s *2022* year revenue
  - Announced in 2023

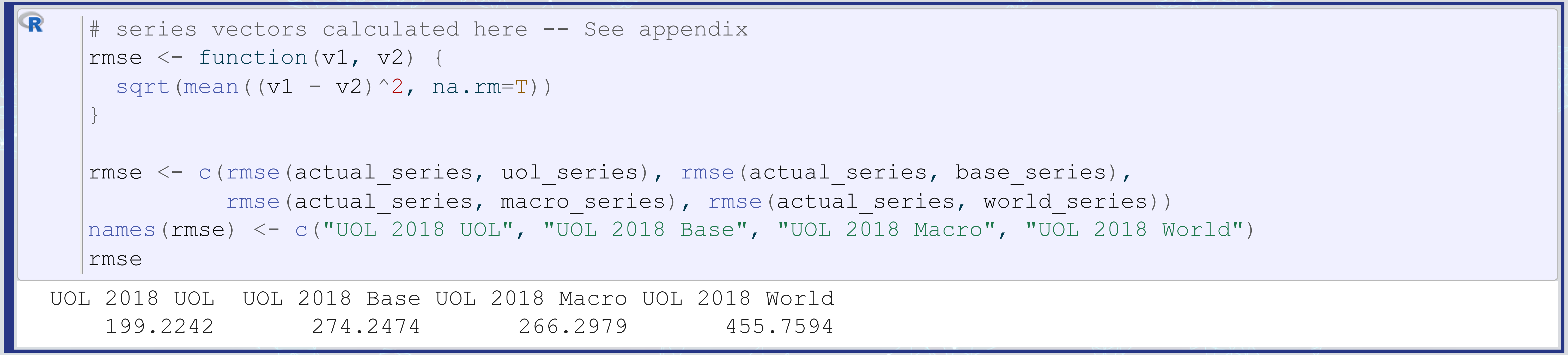
```r
p_uol <- predict(forecast2, uol[uol$fyear==2021,])
p_base <- predict(baseline,
  df_SG_macro[df_SG_macro$isin=="SG1S83002349" & df_SG_macro$fyear==2021,])
p_macro <- predict(macro3,
  df_SG_macro[df_SG_macro$isin=="SG1S83002349" & df_SG_macro$fyear==2021,])
p_world <- predict(forecast4,
  df_clean[df_clean$isin=="SG1S83002349" & df_clean$fyear==2021,])
preds <- c(p_uol, p_base, p_macro, p_world)
names(preds) <- c("UOL 2022 UOL", "UOL 2022 Base", "UOL 2022 Macro",
                  "UOL 2022 World")
preds
```

```
UOL 2022 UOL  UOL 2022 Base UOL 2022 Macro UOL 2022 World 
    3608.571       2834.237       2745.834       3136.901 
```

Visualizing our prediction



In Sample Accuracy



```r
# series vectors calculated here -- See appendix
rmse <- function(v1, v2) {
  sqrt(mean((v1 - v2)^2, na.rm=T))
}

rmse <- c(rmse(actual_series, uol_series), rmse(actual_series, base_series),
          rmse(actual_series, macro_series), rmse(actual_series, world_series))
names(rmse) <- c("UOL 2018 UOL", "UOL 2018 Base", "UOL 2018 Macro", "UOL 2018 World")
rmse
```

```
 UOL 2018 UOL  UOL 2018 Base UOL 2018 Macro UOL 2018 World 
     199.2242       274.2474       266.2979       455.7594 
```

> Why is UOL the best for *in sample*?

> UOL is trained to minimize variation only in that context. It is potentially overfitted, meaning it won’t predict well *out of sample*. Out of sample prediction is much more useful than in sample, however.

Actual Accuracy

> UOL posted a $3.2B in revenue in 2022.

```r
preds
```

```
UOL 2022 UOL  UOL 2022 Base UOL 2022 Macro UOL 2022 World 
    3608.571       2834.237        2745.834        3136.901 
```

> Why is the global model better?

- Consider UOL’s business model
  - 2022 annual report

End Matter

Wrap up

- For next week:
  - 2 chapters on Datacamp (optional)
  - First assignment
    - Turn in on eLearn before class in 2 weeks
    - You can work on this in *pairs* or *individually*
- Survey on the class session at this QR code:

Packages used for these slides

- broom
- DT
- downlit
- fixest
- kableExtra
- knitr
- plotly
- quarto
- revealjs
- tidyverse

Custom code

```r
# Graph showing squared error (slide 4.6)
uolg <- uol[,c("at","revt")]
uolg$resid <- mod1$residuals
uolg$xleft <- ifelse(uolg$resid < 0,uolg$at,uolg$at - uolg$resid)
uolg$xright <- ifelse(uolg$resid < 0,uolg$at - uolg$resid, uol$at)
uolg$ytop <- ifelse(uolg$resid < 0,uolg$revt - uolg$resid,uol$revt)
uolg$ybottom <- ifelse(uolg$resid < 0,uolg$revt, uolg$revt - uolg$resid)
uolg$point <- TRUE

uolg2 <- uolg
uolg2$point <- FALSE
uolg2$at <- ifelse(uolg$resid < 0,uolg2$xright,uolg2$xleft)
uolg2$revt <- ifelse(uolg$resid < 0,uolg2$ytop,uolg2$ybottom)

uolg <- rbind(uolg, uolg2)

uolg %>% ggplot(aes(y=revt, x=at, group=point)) +
         geom_point(aes(shape=point)) +
         scale_shape_manual(values=c(NA,18)) +
         geom_smooth(method="lm", se=FALSE) +
         geom_errorbarh(aes(xmax=xright, xmin = xleft)) +
         geom_errorbar(aes(ymax=ytop, ymin = ybottom)) +
         theme(legend.position="none")
```

```r
# Chart of mean revt_lead for Singaporean firms (slide 12.6)
df_clean %>%                                         # Our data frame
  filter(fic=="SGP") %>%                             # Select only Singaporean firms
  group_by(isin) %>%                                 # Group by firm
  mutate(mean_revt_lead=mean(revt_lead, na.rm=T)) %>% # Determine each firm's mean revenue (lead)
  slice(1) %>%                                       # Take only the first observation for each group
  ungroup() %>%                                      # Ungroup (we don't need groups any more)
  ggplot(aes(x=mean_revt_lead)) +                    # Initialize plot and select data
  geom_histogram(aes(y = ..density..)) +             # Plots the histogram as a density so that geom_density is visible
  geom_density(alpha=.4, fill="#FF6666")             # Plots smoothed density
```