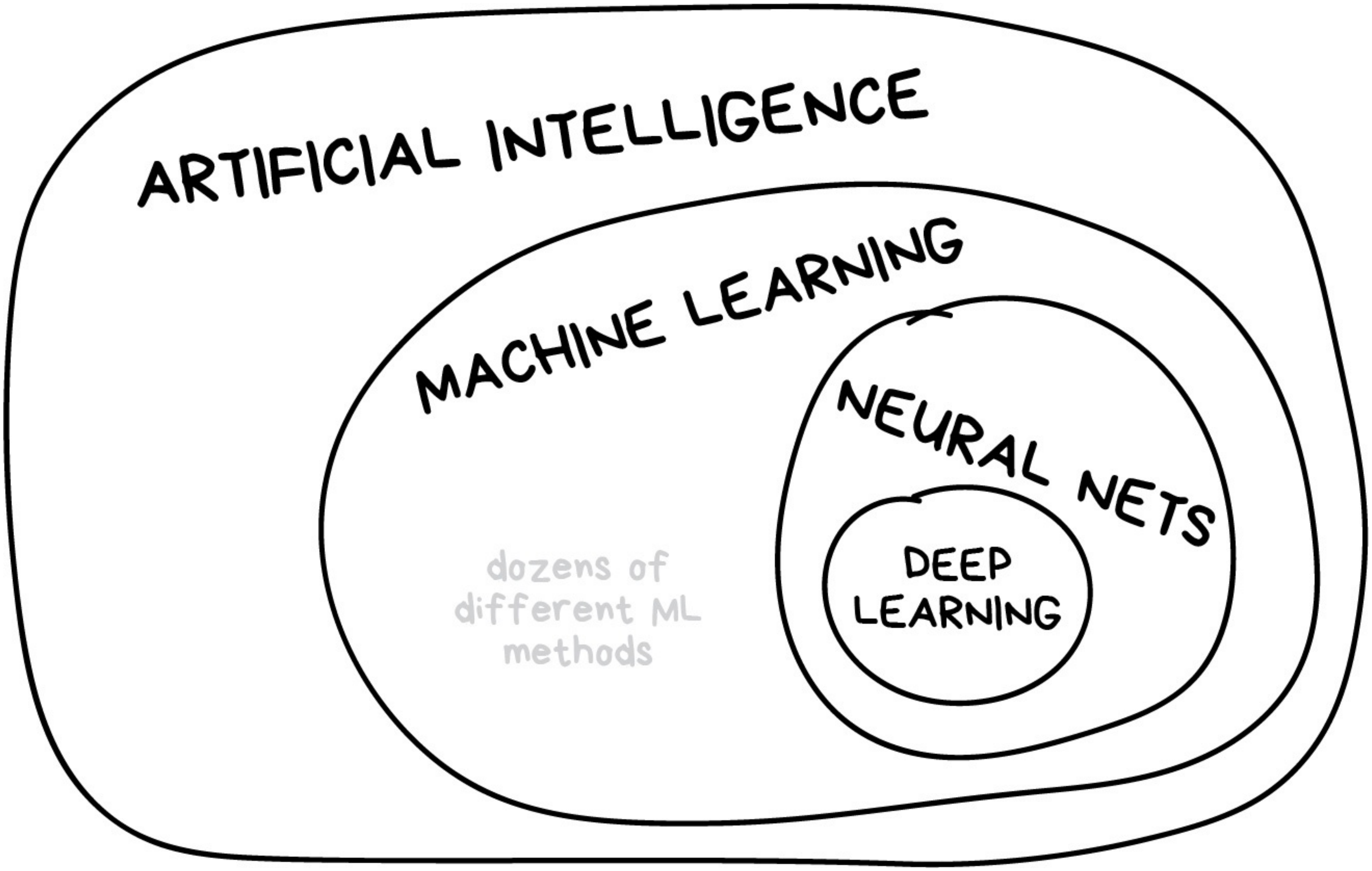
Intro To Modeling

Matthew McDonald

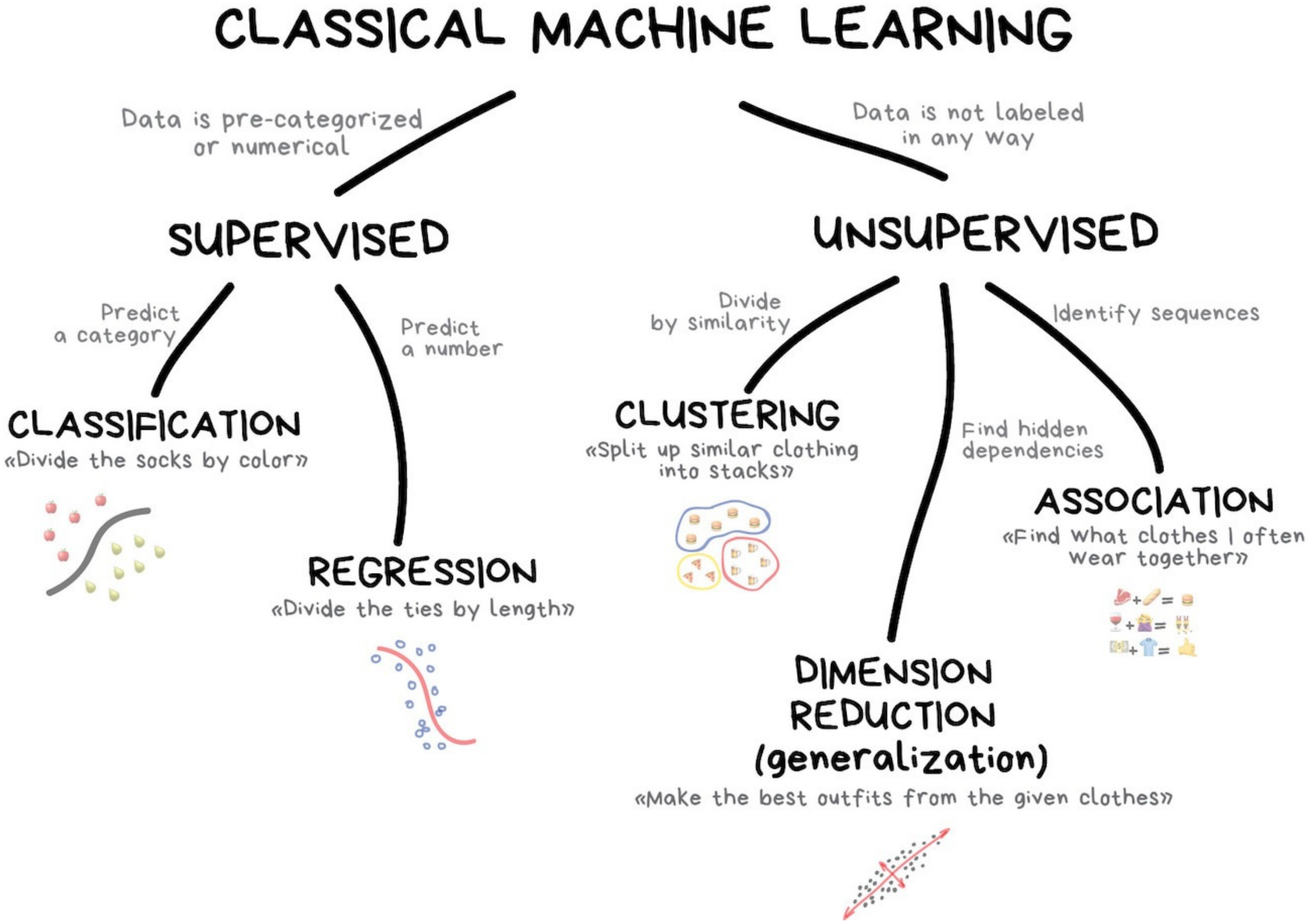
What is machine learning?

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

What is machine learning?



What is machine learning?


CLASSICAL MACHINE LEARNING
Data is pre-categorized or numerical
Data is not labeled in any way
SUPERVISED
UNSUPERVISED
Predict a category
Predict a number
CLASSIFICATION
«Divide the socks by color»
REGRESSION
«Divide the ties by length»
Divide by similarity
Identify sequences
Find hidden dependencies
CLUSTERING
«Split up similar clothing into stacks»
ASSOCIATION
«Find what clothes I often wear together»
DIMENSION REDUCTION (generalization)
«Make the best outfits from the given clothes»

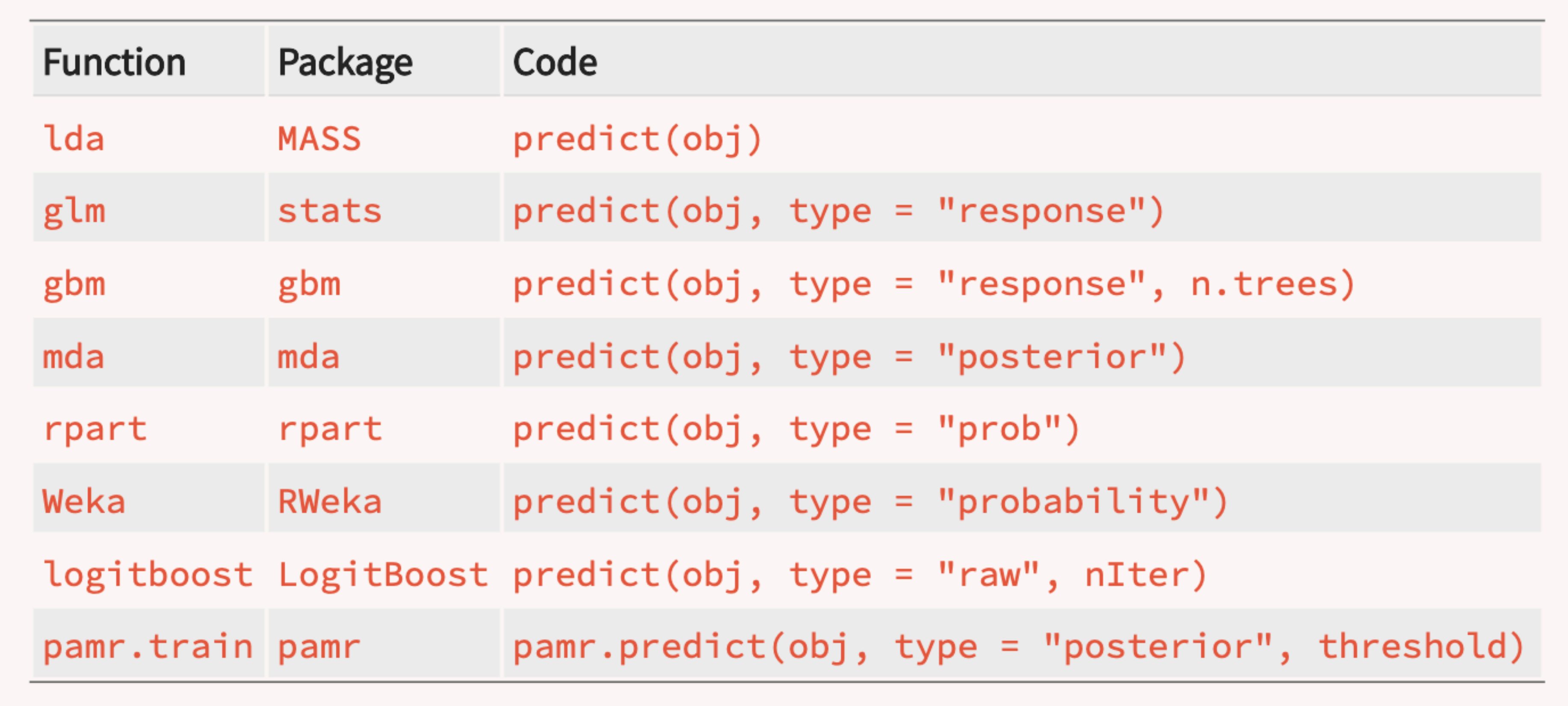
Why R For Modeling?

- R has cutting edge models.
    - Machine learning developers in some domains use R as their primary computing environment and their work often results in R packages.
- R and R packages are built by people who do data analysis.
- The S language is very mature.
- It is easy to port or link to other applications.
    - R doesn’t try to be everything to everyone. If you prefer models implemented in C, C++, tensorflow, keras, python, stan, or Weka, you can access these applications without leaving R.
- The machine learning environment in R is extremely rich.

Downsides to Modeling in R

- R is a data analysis language and is not C or Java. If a high performance deployment is required, R can be treated like a prototyping language.
- R is mostly memory-bound. There are plenty of exceptions to this though.
- The main issue is one of consistency of interface. For example:
  - There are two methods for specifying what terms are in a model1. Not all models have both.
  - 99% of model functions automatically generate dummy variables.
  - Sparse matrices can be used (unless they can't).

Syntax for Computing Predicted Class Probabilities



Function	Package	Code
`lda`	`MASS`	`predict(obj)`
`glm`	`stats`	`predict(obj, type = "response")`
`gbm`	`gbm`	`predict(obj, type = "response", n.trees)`
`mda`	`mda`	`predict(obj, type = "posterior")`
`rpart`	`rpart`	`predict(obj, type = "prob")`
`Weka`	`RWeka`	`predict(obj, type = "probability")`
`logitboost`	`LogitBoost`	`predict(obj, type = "raw", nIter)`
`pamr.train`	`pamr`	`pamr.predict(obj, type = "posterior", threshold)`

What is tidymodels?

tidymodels is a collection of packages.

```r
library(tidymodels)
```

```
── Attaching packages ────────────────────────────────────── tidymodels 1.1.1 ──

✔ broom        1.0.5     ✔ recipes      1.0.9
✔ dials        1.2.0     ✔ rsample      1.2.0
✔ dplyr        1.1.4     ✔ tibble       3.2.1
✔ ggplot2      3.4.4     ✔ tidyr        1.3.0
✔ infer        1.0.6     ✔ tune         1.1.2
✔ modeldata    1.3.0     ✔ workflows    1.1.3
✔ parsnip      1.1.1     ✔ workflowsets 1.0.1
✔ purrr        1.0.2     ✔ yardstick    1.3.0

── Conflicts ───────────────────────────────────────── tidymodels_conflicts() ──
✖ purrr::discard() masks scales::discard()
✖ dplyr::filter()  masks stats::filter()
✖ dplyr::lag()     masks stats::lag()
✖ recipes::step()  masks stats::step()
• Search for functions across packages at https://www.tidymodels.org/find/
```

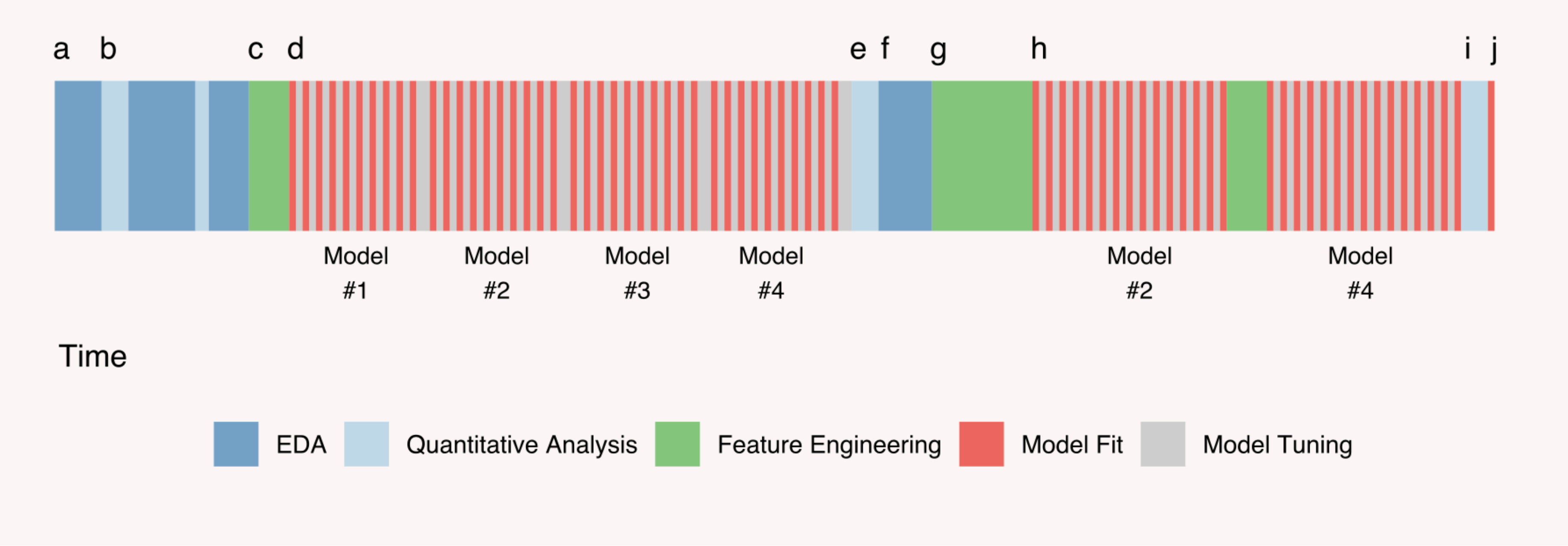
The Modeling Process

Common steps during model building are:

- estimating model parameters (i.e. training models)
- determining the values of tuning parameters that cannot be directly calculated from the data
- model selection (within a model type) and model comparison (between types)
- calculating the performance of the final model that will generalize to new data

Many books and courses portray predictive modeling as a short sprint. A better analogy would be a marathon or campaign (depending on how hard the problem is).

What the Modeling Process Usually Looks Like

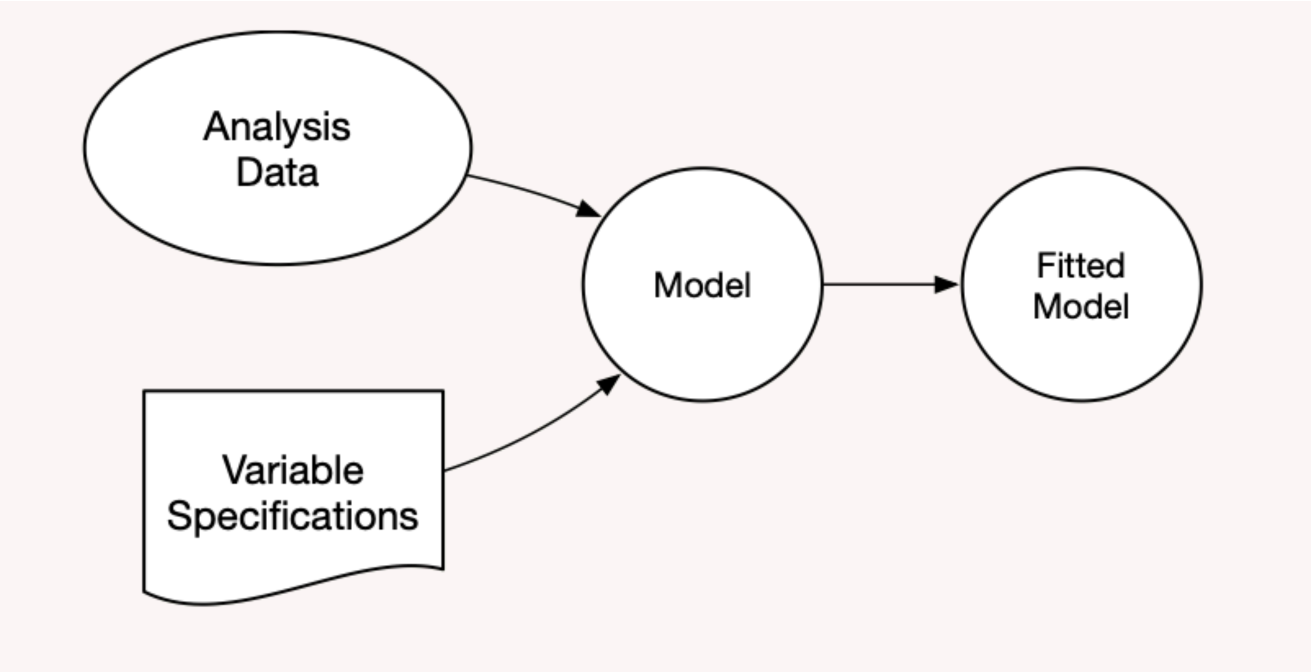


What Are We Doing With The Data?

We often think of the model as the only real data analysis step in this process.

However, there are other procedures that are often applied before or after the model fit that are data-driven and have an impact.

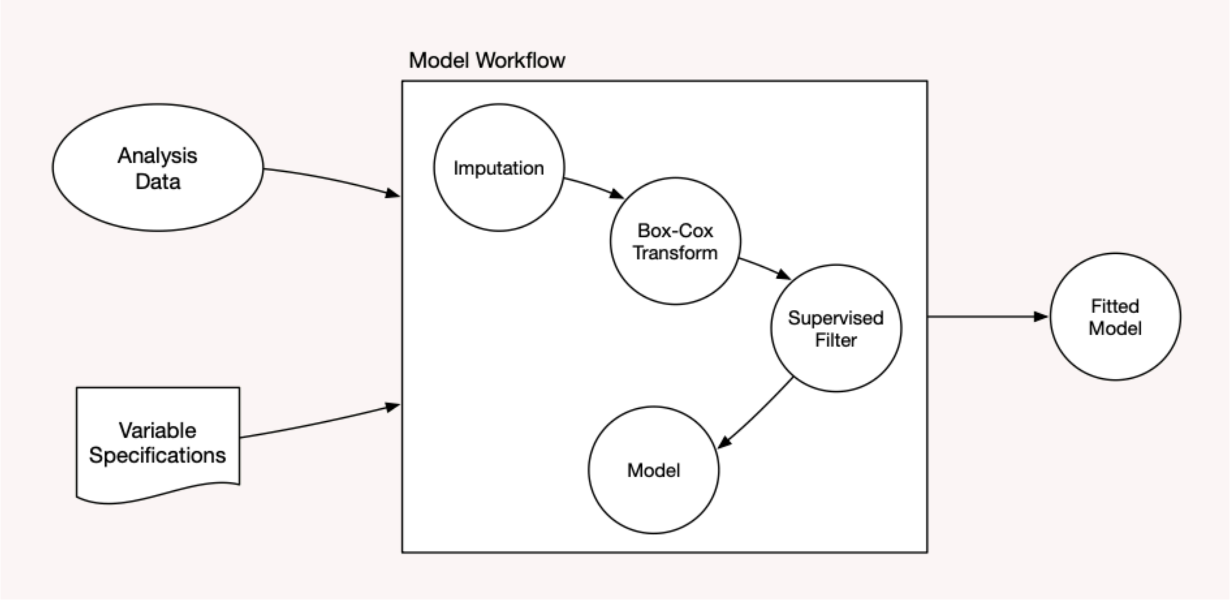
If we only think of the model as being important, we might end up accidentally overfitting to the data in-hand. This is very similar to the problem of “p-hacking”.



Define The Data Analysis Process

tidymodels conceptualizes a workflow that involves all of the steps where the data are analyzed in a significant way. The includes the model but might also include other estimation steps:

- data preparation steps (e.g. imputation, encoding, transformations, etc)
- selection of which terms go into the model
- and so on.



The Ames Housing Data

The data set contains information on 2,930 properties in Ames, Iowa, including columns related to:

- house characteristics (bedrooms, garage, fireplace, pool, porch, etc.)
- location (neighborhood)
- lot information (zoning, shape, size, etc.)
- ratings of condition and quality
- sale price

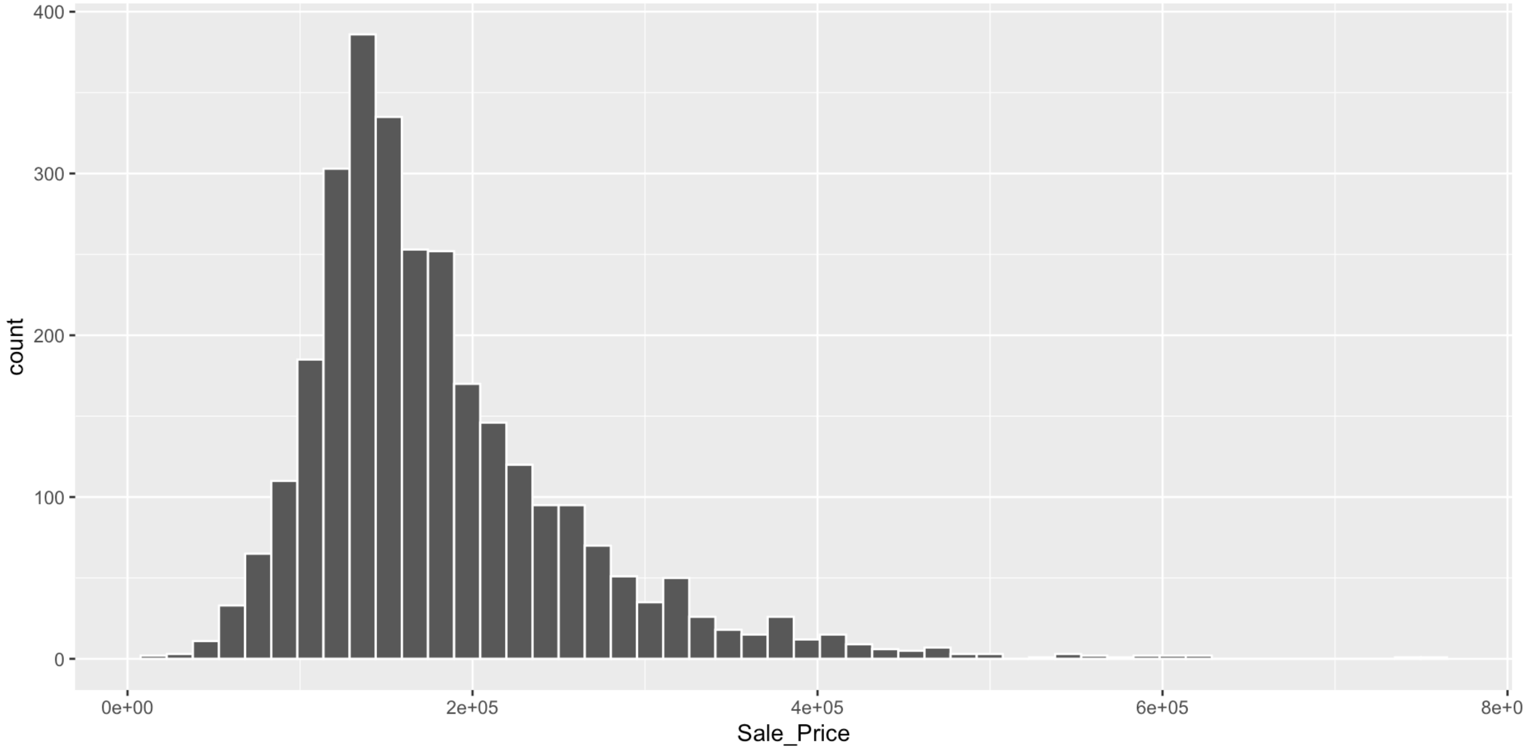
Loading the Ames Data set

```r
library(modeldata) # This is also loaded by the tidymodels package
data(ames)

dim(ames)
```

```
[1] 2930   74
```

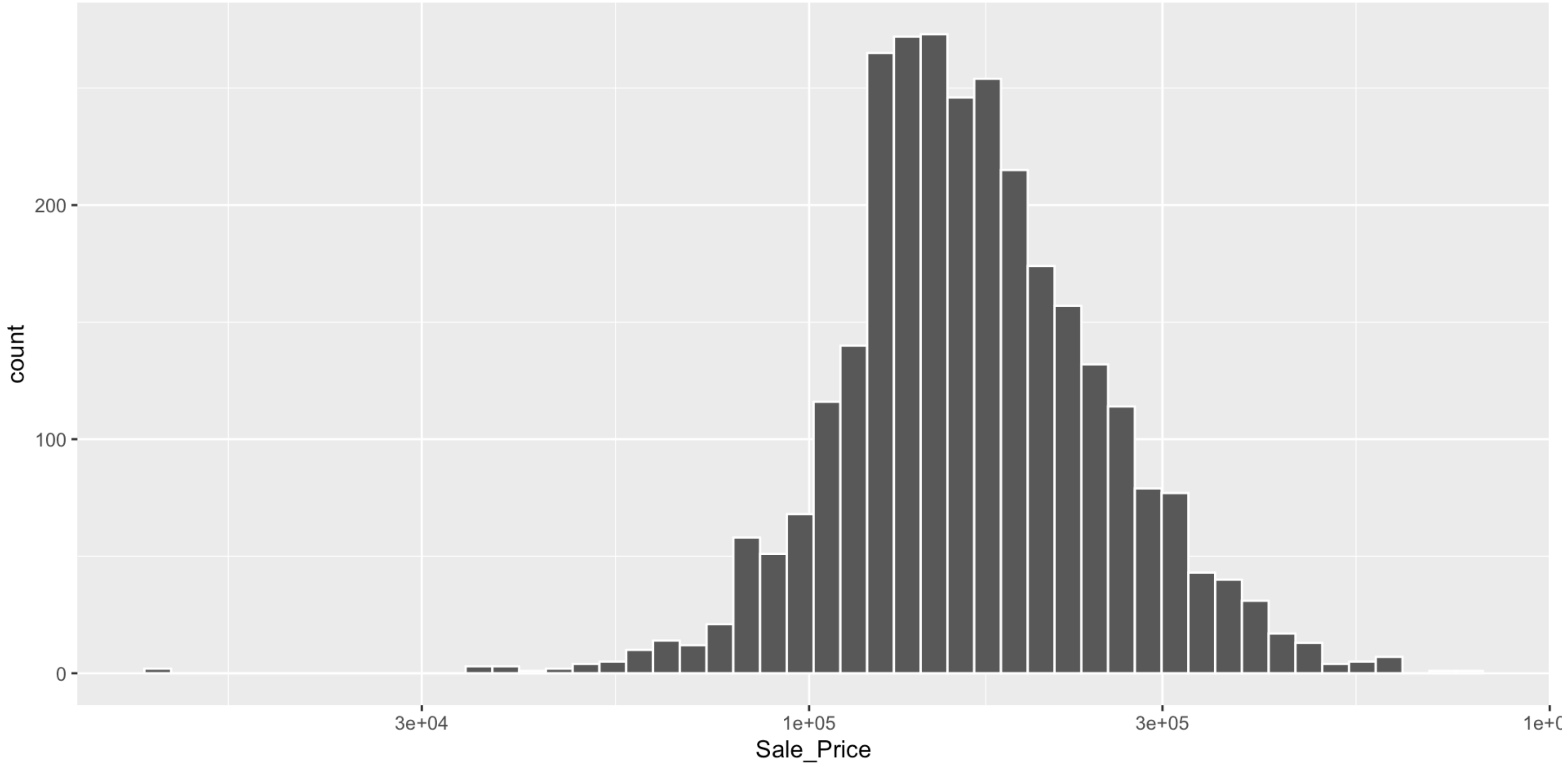
Sale Price

```r
library(ggplot2)

ggplot(ames, aes(x = Sale_Price)) +
  geom_histogram(bins = 50, col= "white")
```



Log Sale Price

```r
ggplot(ames, aes(x = Sale_Price)) +
  geom_histogram(bins = 50, col= "white") +
  scale_x_log10()
```


count
3e+04
1e+05
3e+05
1e+0
Sale_Price