Linear Regression

AU STAT-427/627

Emil Hvitfeldt

2021-09-02

What is statistical learning?

- **Building a (statistical) model to represent relationships between different quantities**

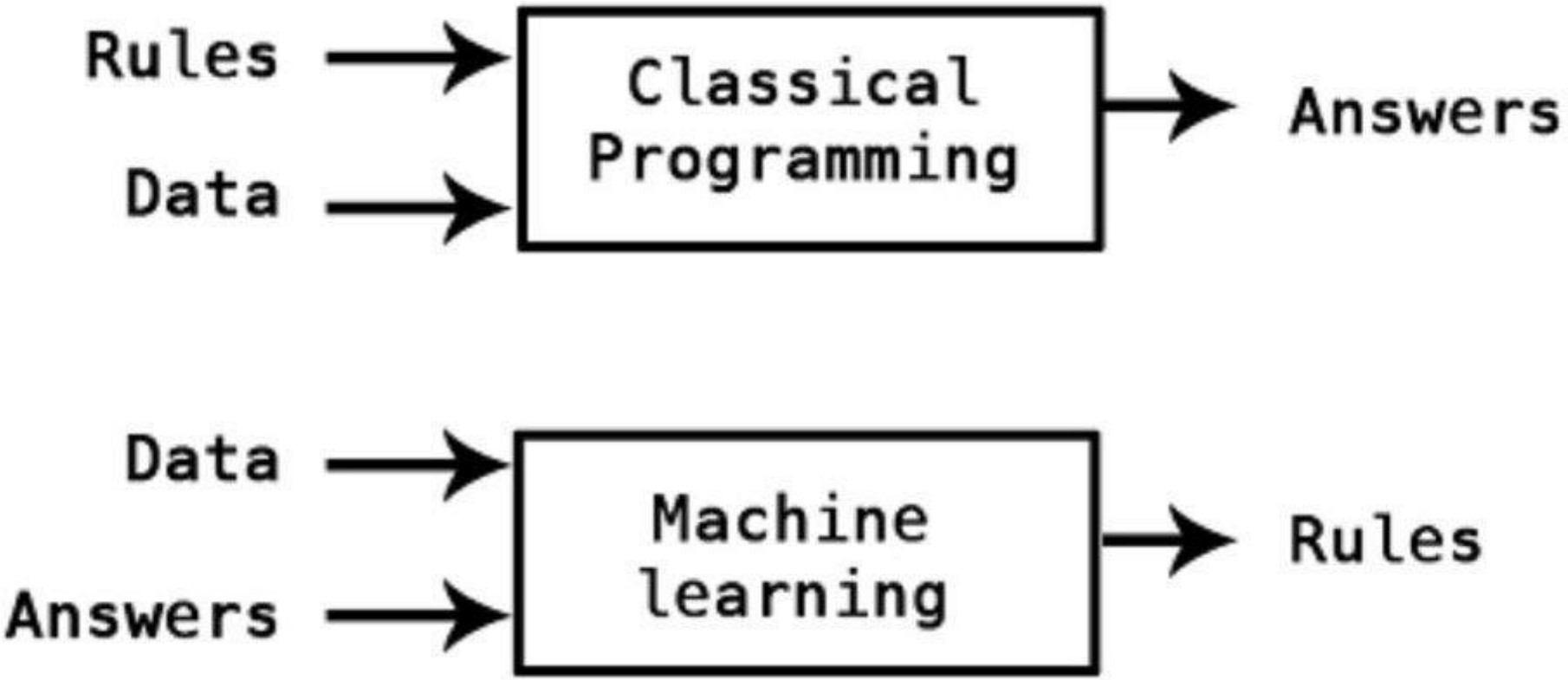
Quantities are defined here very broadly to be data or measurements

What is statistical learning?

- Building a (statistical) model to represent relationships between different quantities

- **Using data to create rules**

What is statistical learning?



What is statistical Learning?

Some machine learning methods have a statistical underpinning

This allows us to quantify the uncertainty

Examples of "non-statistical machine learning methods" are

- Genetic algorithms
- K-nearest neighbors

There is not a hard and fast distinction. Machine learning is about getting answers. Statistics is a great way to find answers.

Why use statistical Learning?

The main goals are

- Understanding/Inference
- Prediction

General setup

For response Y and p different predictors $X_1, X_2, \ldots, X_p$

Then the relationship between them can be written as

$$Y = f(X) + \epsilon$$

with ϵ being a random **error term**, independent from X and has mean 0.

This formulation is VERY general.
There is no assumption that f provides any information.

Our goal is to find f

Why estimate f?

If f is different than the null-model or monkey model.

- Prediction
- Inference

Prediction

Main thesis:

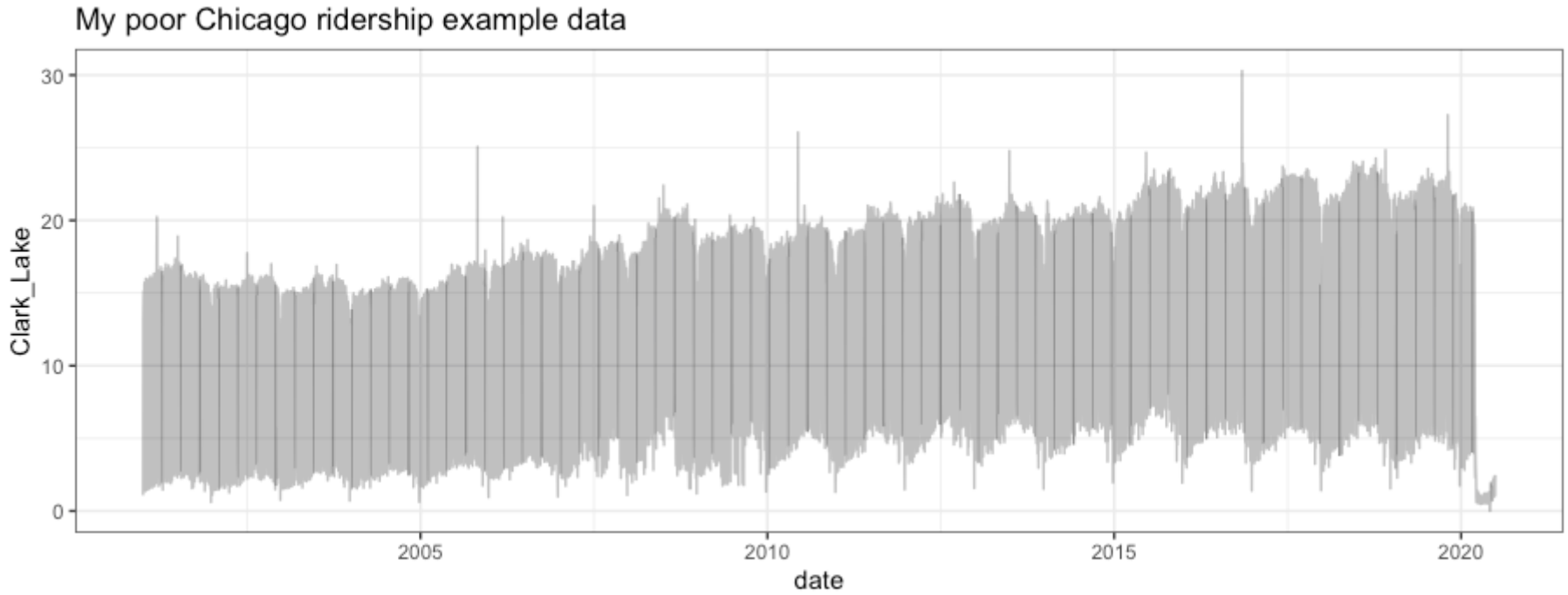
> If we can find f then we can predict the value of Y for different values of X

This holds a major assumption that the scenario in which we estimate f stays the same.

Models trained on data from a recession may not apply to data in a depression

Models trained on low-income houses might not work on high-income houses

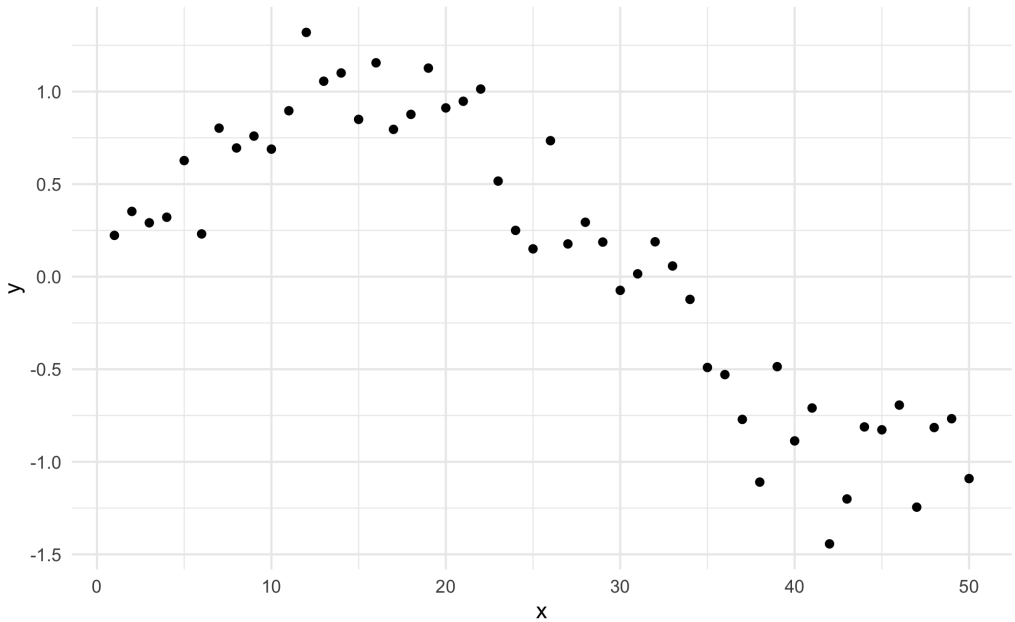
Covid-19 data

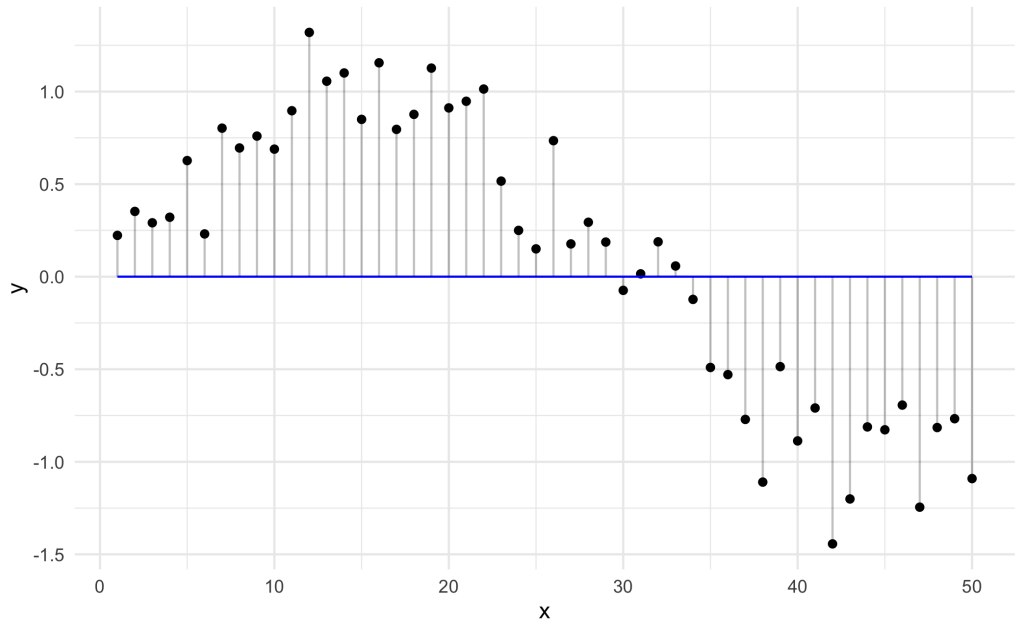


https://twitter.com/topepos/status/1333967942686543873

Prediction

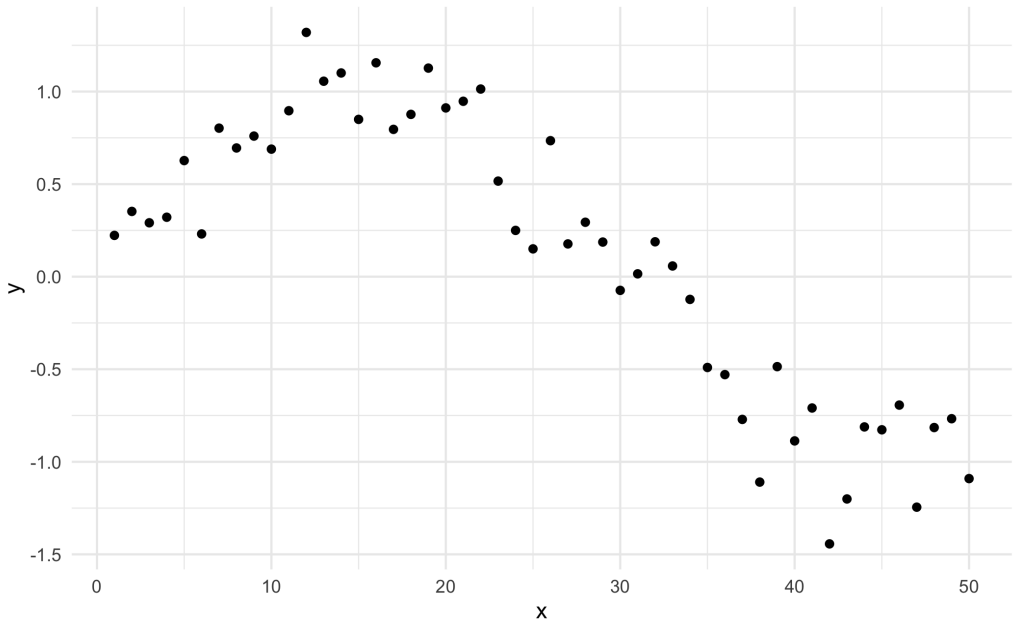
$$\hat{Y} = \hat{f}(X)$$

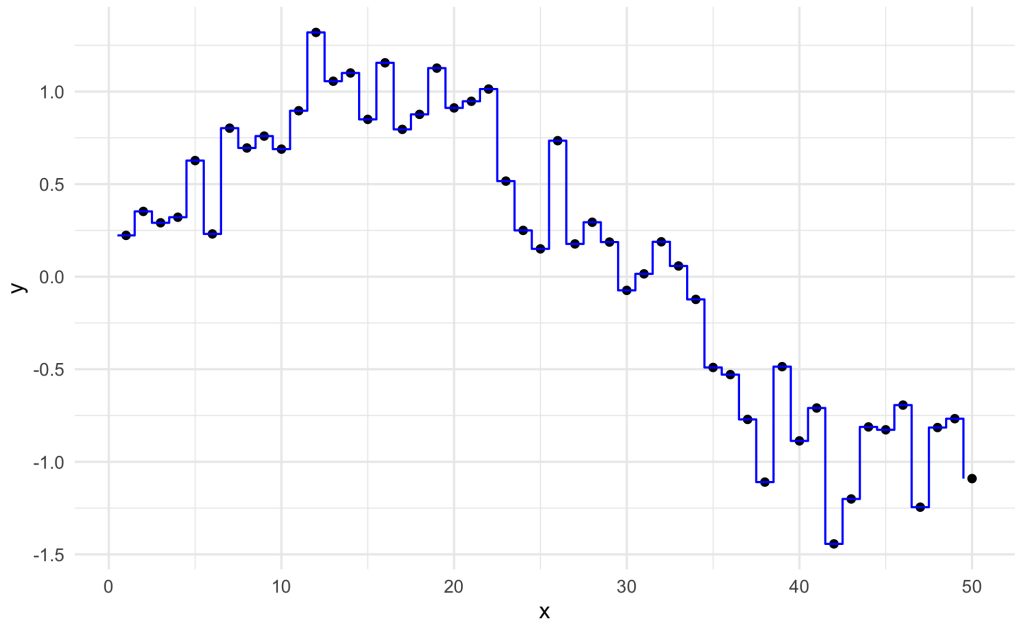




Prediction

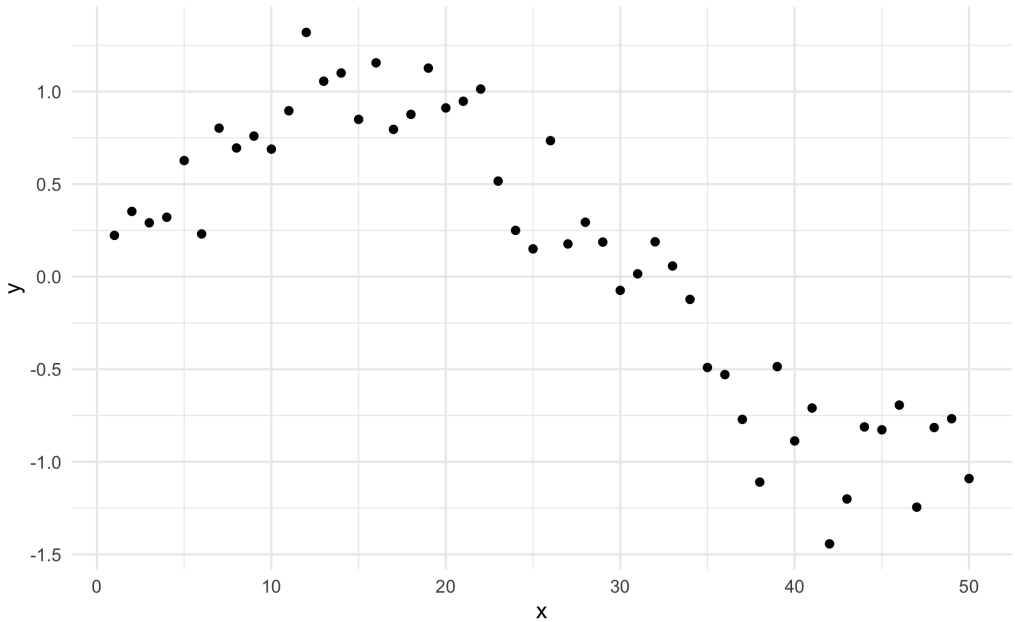
$$\hat{Y} = \hat{f}(X)$$

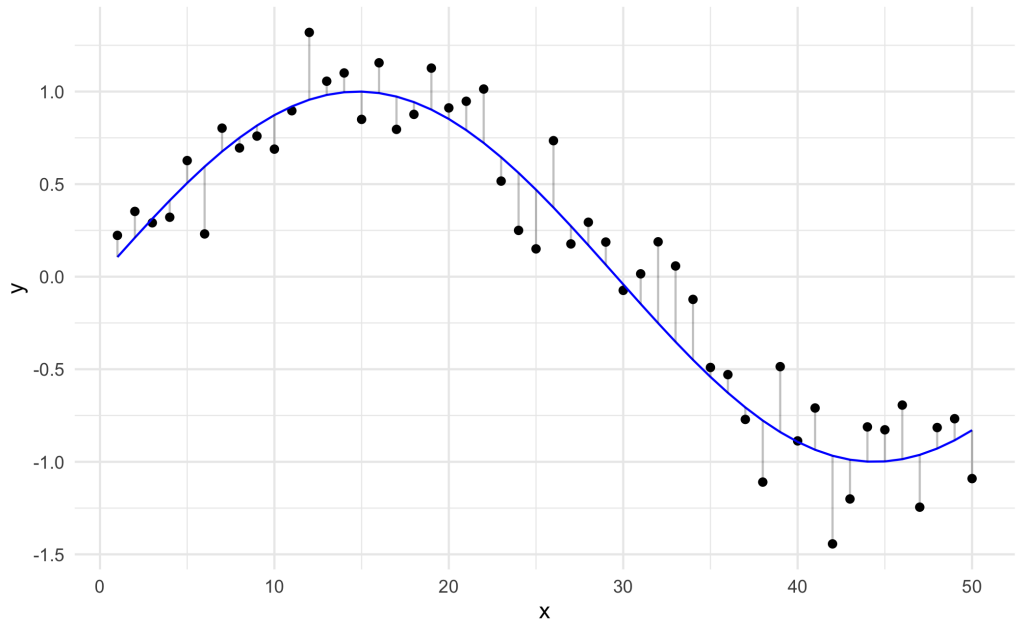




Prediction

$$\hat{Y} = \hat{f}(X)$$





Tradeoff between reducible error and irreducible error

The error is how much f is different from $\hat{f}$

We split this into **reducible** and **irreducible**

We will generally not be able to completely predict anything from a limited number of features
Any error left when a perfect statistical model is trained is the irreducible error

Sub-optimal estimates of $\hat{f}$ introduce error which could have been reduced.

(this hinges on a more philosophical basis. Is the world fully deterministic?)

Tradeoff between reducible error and irreducible error

If we could, we technically have a mathematical formula.

> amount of sales tax on an item

is generally not statistical you might need a complicated model, but you should be able to eliminate all the error

Tradeoff between reducible error and irreducible error

Examples with error:

bolt factory. Estimate the weight of the bolt.

Machines are calibrated, but things like, temperature, air quality, material quality, particles will still play (a small) factor.

Inference

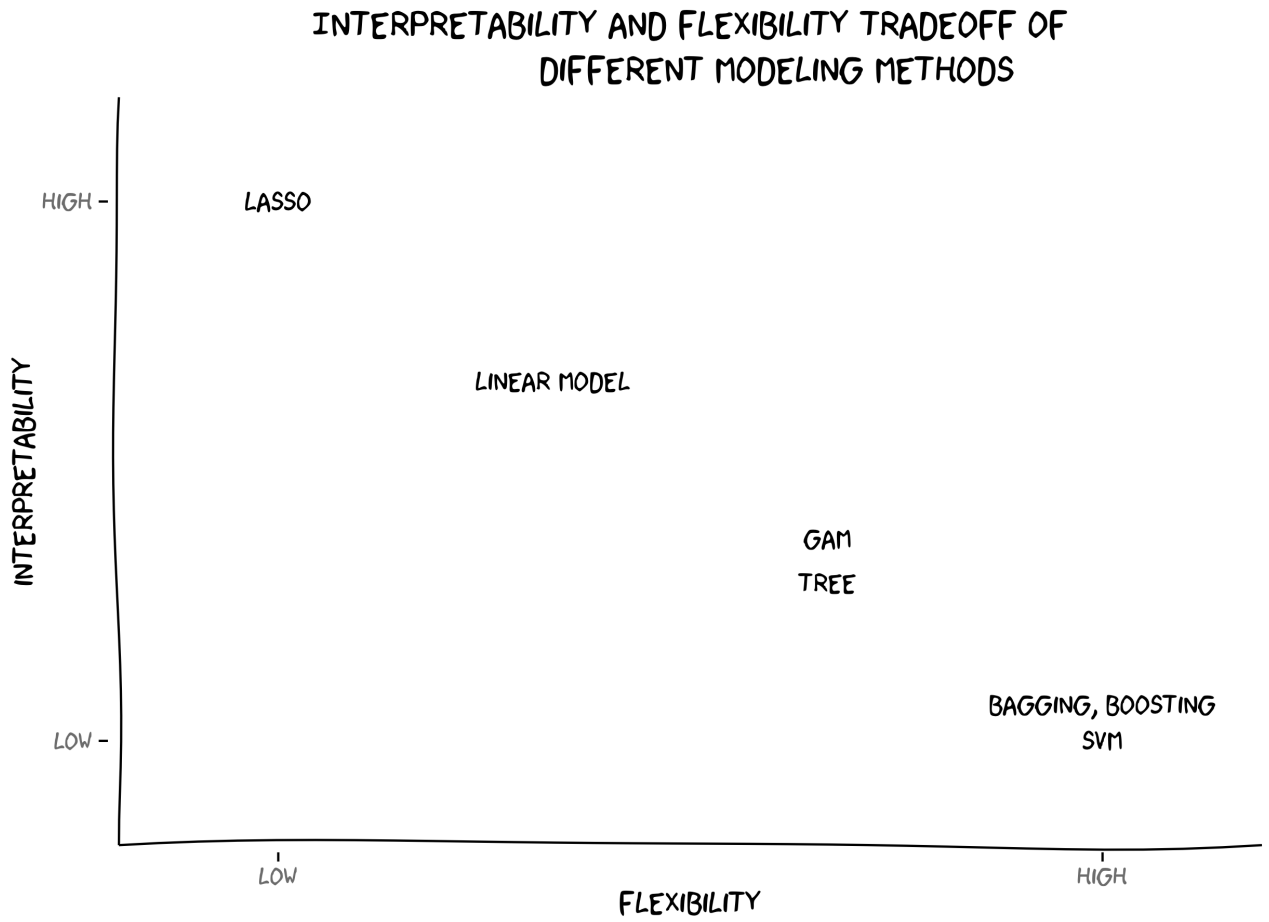
Understanding how Y is related to X

We want to understand the exact form

"What effect will changing price affect the rating of a product?"

This is inference. We are primarily interested in the effect, not the outcome

As we will see later, there is a trade-off between models that work well for prediction and easily explainable models



Inference vs prediction

hard to explain models can be good predictors but bad for inference

Certain fields will hold different weight on explainability/interpretability

Supervised vs Unsupervised Learning

Most of what we will are working on is going to be supervised.
The learning we are doing is based on a specific parameter Y we are working around

unsupervised learning on the other hand doesn't have an explicit goal or answer sheet

- pattern matching
- clustering

> "here is all our customer data, do they form groups?"

Model accuracy

the book covers **mean squared error**

There are many ways to access how well a model performs.

many of these will be related to how far away the prediction is from the observation

Linear models

We have seen this before so we are just freshening up

Start with simple

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Where X is a single predictor variable

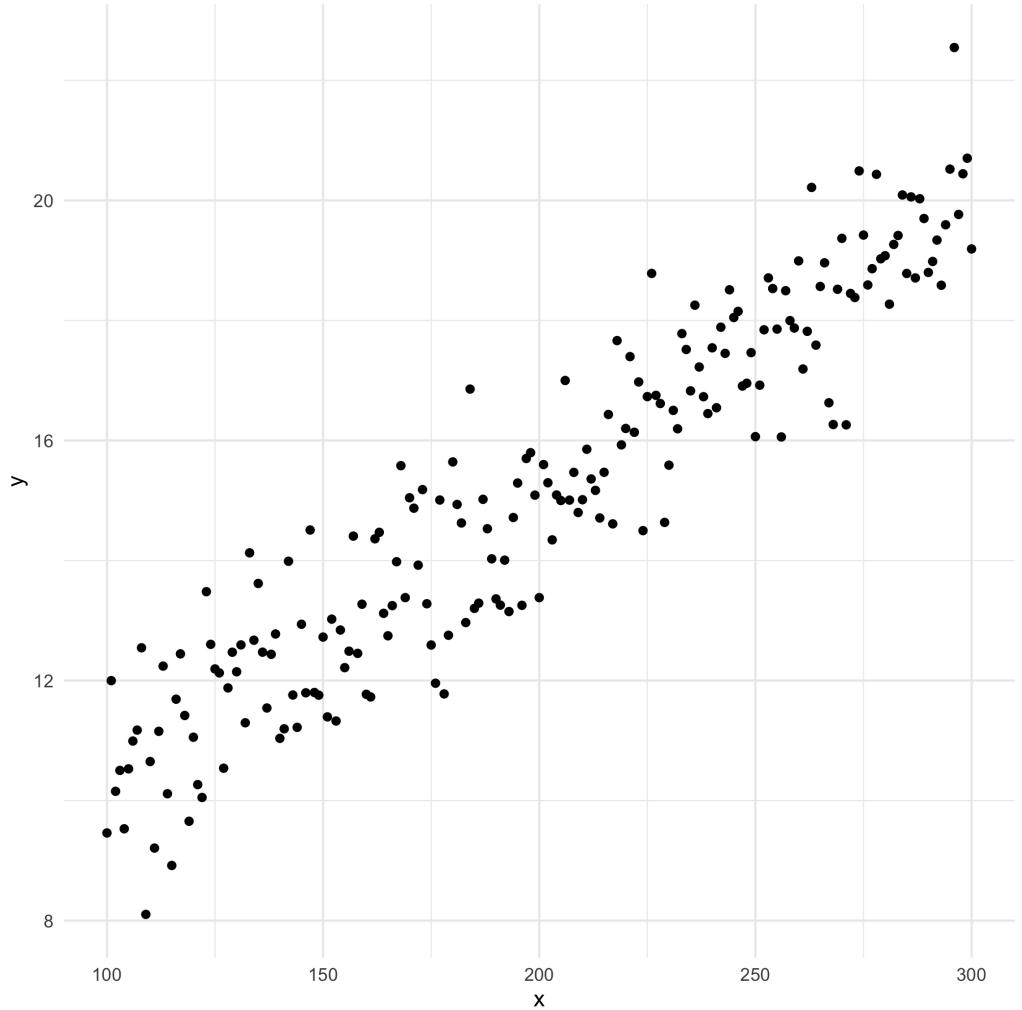
Notice this is

$$f(X) = \beta_0 + \beta_1 X$$

We need to find the values for the betas that makes it the line as close to the data as possible

Consider the data on the right

It appears to have a possible linear trend



Consider the data on the right

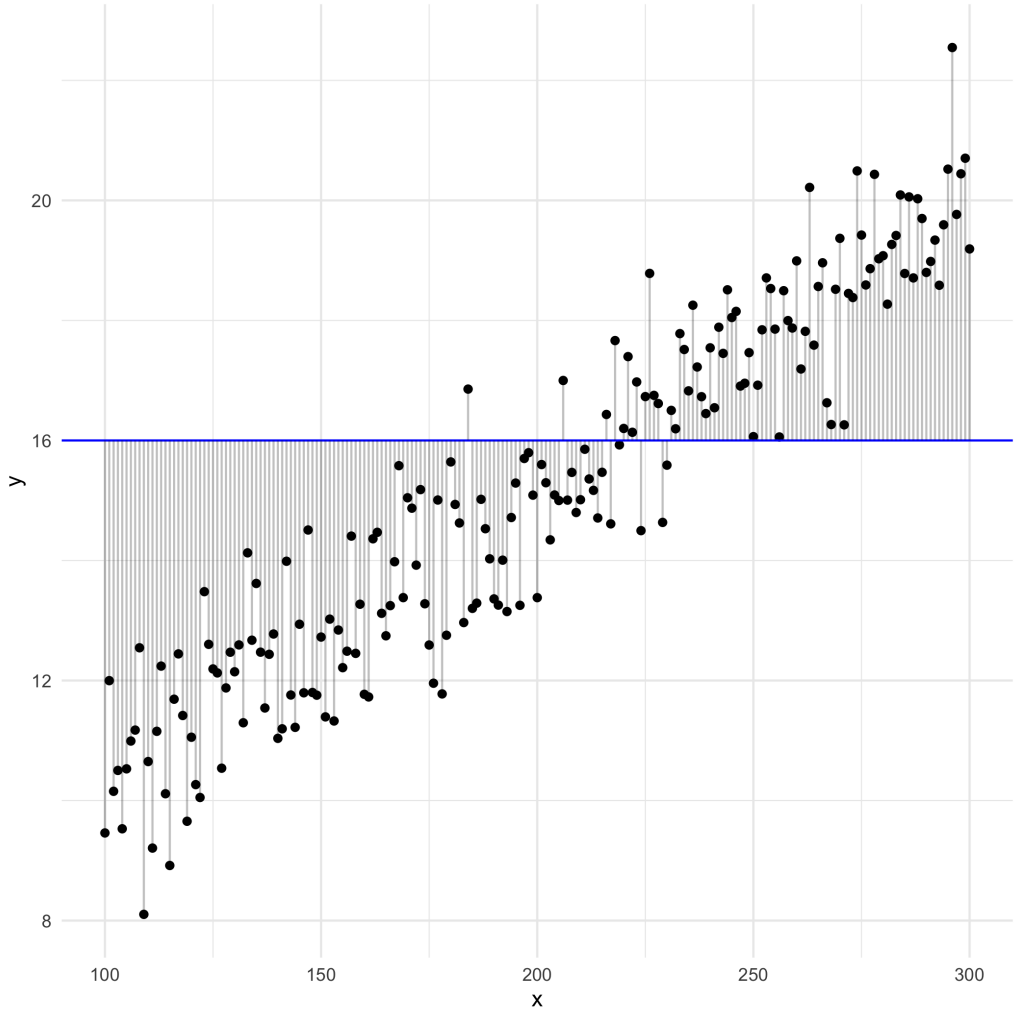
It appears to have a possible linear trend

If we draw a simple horizontal line for `y = 16`

This would be $\beta_0 = 16, \beta_1 = 0$

If we take the square of all the vertical lines and sum them we get

```
## # A tibble: 1 × 1
##     rss
##   <dbl>
## 1 1998.
```



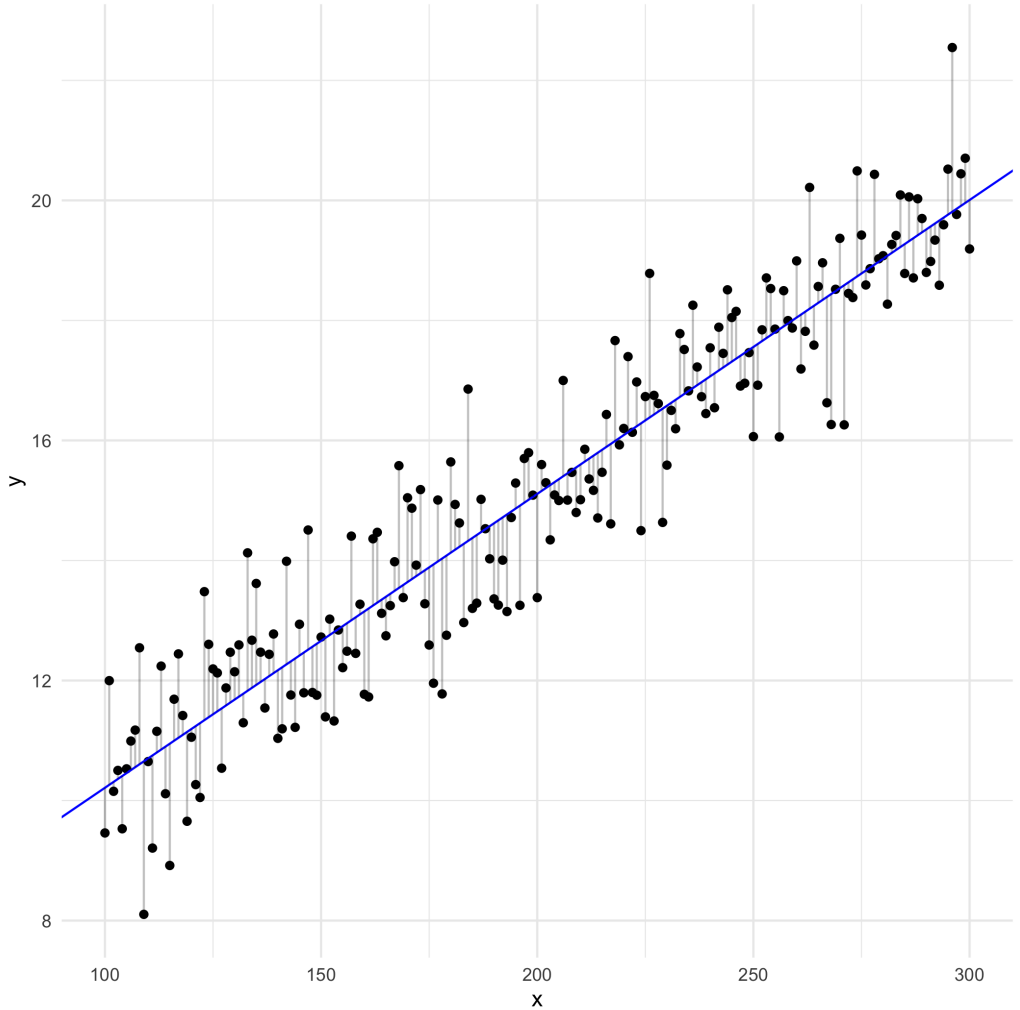
Consider the data on the right

It appears to have a possible linear trend

If we minimize the RSS then we would get $\beta_0 = 5.31537, \beta_1 = 0.04897$

With a resulting RSS of

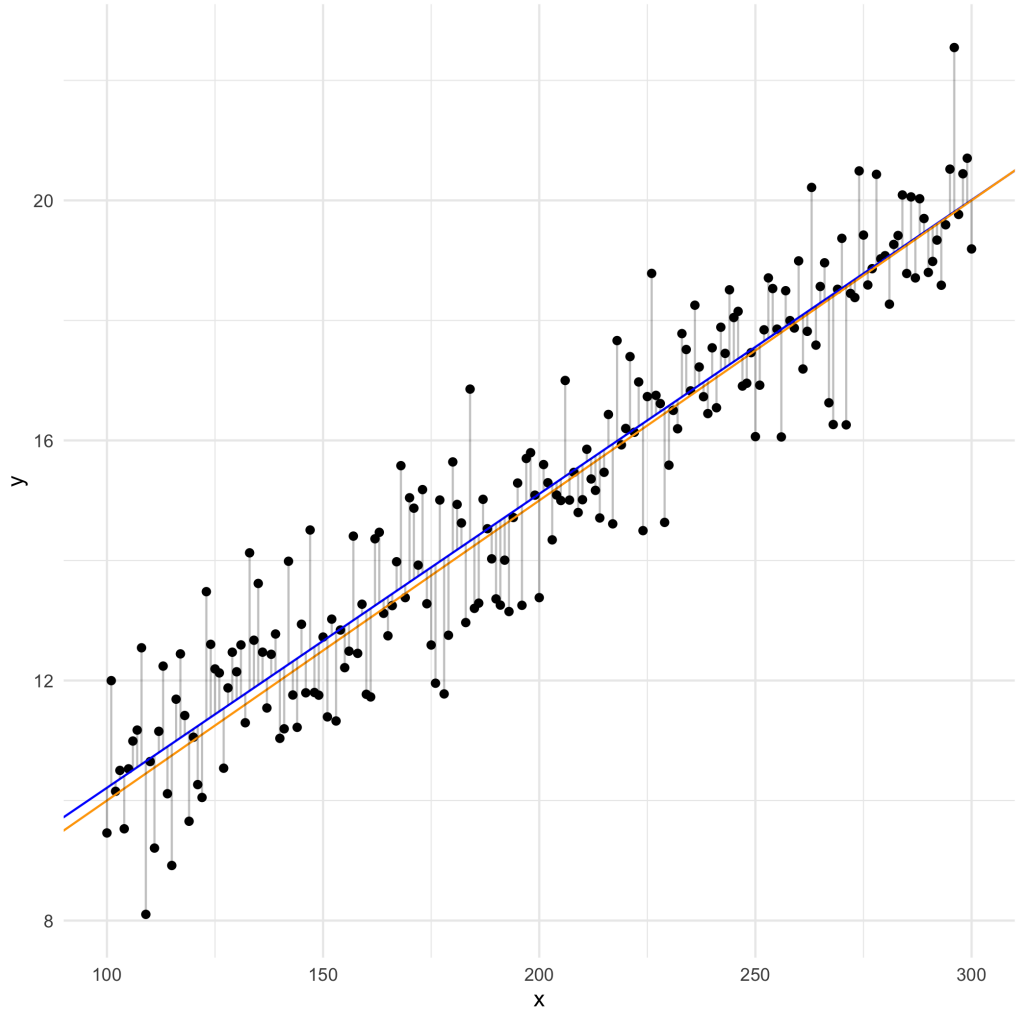
```
## # A tibble: 1 × 1
##     rss
##   <dbl>
## 1  216.
```



Consider the data on the right

Overlaying the true relationship in orange

since we are only receiving a sample of the underlying distribution, we are not able to completely determine the right slope and intercept



Least Square Criterion

We are minimizing the residual sum of squares (RSS)

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Where $\bar{x}$ and $\bar{y}$ are sample means of x and y

Hypothesis tests

Since this model is built on certain assumptions, we can calculate standard error estimates for each parameter estimate.

These standard errors can be used to determine if the estimates are significantly different from 0

An inverse relationship between the size of effect and number of observations

Assessing model accuracy

How much does the model fit the data?

We want to know **how well** the model is performing

Again a measure of how far away the predictions are away from the actual model

Assessing model accuracy

Remember how **residuals squared sum** (RSS) depended on the number of observations?

$$RSE = \sqrt{\frac{1}{n-2}RSS} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

residual standard error takes care of this by normalization

Interpretation:

> RSE is the average amount that the response will deviate from the true regression line

RSE measures the lack of fit. Smaller values are better

Assessing model accuracy

$$R^2 = 1 - \frac{RSS}{TSS}$$

where $TSS = \sum(y_i - \bar{y})^2$ is the total sum of squares

Interpretation:

> R^2 is the proportion of variance explained

takes values between 0 and 1, higher being better

Multiple linear regression

This is a simple extension,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

All the previous questions apply here with slightly different answers

Model selection

When $p = 1$ we have the question

> Is there an association between Y and X

but when $p > 1$ then question becomes

> Is at least one of the predictors $X_1, X_2, \ldots, X_p$ useful in predicting the response

and

> Which of the X's have an association with Y

F statistics

> Is at least one of the predictors $X_1, X_2, \ldots, X_p$ useful in predicting the response

$$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)}$$

If F-statistic is close to 1 then we suspect there is no relationship between the response and predictors

Model selection

- Forward selection
- Backward selection
- Mixed selection

Qualitative predictors

We will come back to this later

Model assumptions

The linear model works well in a lot of cases.

But there are assumptions

- Linear relationship: There exists a linear relationship between the independent variable, x, and the dependent variable, y.

- Independence: The residuals are independent. In particular, there is no correlation between consecutive residuals in time series data.

- Homoscedasticity: The residuals have constant variance at every level of x.

- Normality: The residuals of the model are normally distributed.

Model assumptions

If the assumptions are not met then the model will not be sound

If the error terms are correlated, we may have an unwarranted sense of confidence in our model

Outliers

You should be careful to throw out data that does not fit well into your model

- Don't remove points because they result in a bad fit
- Remove data if they are wrongly collected
- Consider using a model that isn't affected much by outliers

domain, linear models perform badly if some of the observations are FAR away from the other points

User-facing problems in modeling in R

- Data must be a matrix (except when it needs to be a data.frame)
- Must use formula or x/y (or both)
- Inconsistent naming of arguments (ntree in randomForest, num.trees in ranger)
- na.omit explicitly or silently
- May or may not accept factors

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)`

blatantly stolen from Max Kuhn


parsnip
www.rstudio.com

The goals of `parsnip` is...

- Decouple the **model classification** from the **computational engine**
- Separate the definition of a model from its evaluation
- Harmonize argument names
- Make consistent predictions (always tibbles with na.omit=FALSE)

```
model_lm <- lm(mpg ~ disp + drat + qsec, data = mtcars)
```

```r
library(parsnip)
model_lm <- linear_reg() %>%
  set_mode("regression") %>%
  set_engine("lm")
model_lm
```

```
## Linear Regression Model Specification (regression)
##
## Computational engine: lm
```

```r
fit_lm <- model_lm %>%
  fit(mpg ~ disp + drat + qsec, data = mtcars)
fit_lm
```

```
## parsnip model object
##
## Fit time:  41ms
##
## Call:
## stats::lm(formula = mpg ~ disp + drat + qsec, data = data)
##
## Coefficients:
## (Intercept)         disp         drat         qsec
##    11.52439     -0.03136      2.39184      0.40340
```

Tidy prediction

```
predict(fit_lm, mtcars)
```

```
## # A tibble: 32 × 1
##    .pred
##    <dbl>
##  1  22.5
##  2  22.7
##  3  24.9
##  4  18.6
##  5  14.6
##  6  19.2
##  7  14.3
##  8  23.8
##  9  25.7
## 10  23.0
## # … with 22 more rows
```