

# 02 - Basic Machine Learning Concepts

ml4econ, HUJI 2021

Itamar Caspi

March 14, 2021 (updated: 2021-03-14)

# Replicating this Presentation

R packages used to produce this presentation. We use the **pacman** package that loads and installs packages:

```
if (!require("pacman")) install.packages("pacman")

pacman::p_load(
  tidyverse,    # for data wrangling and visualization
  tidymodels,   # for data modeling
  RefManagerR,  # for bibliography
  truncnorm,    # for drawing from truncated normal
  dagitty,      # for drawing DAGs
  knitr         # for including graphics
)
```

# First Things First: "Big Data"

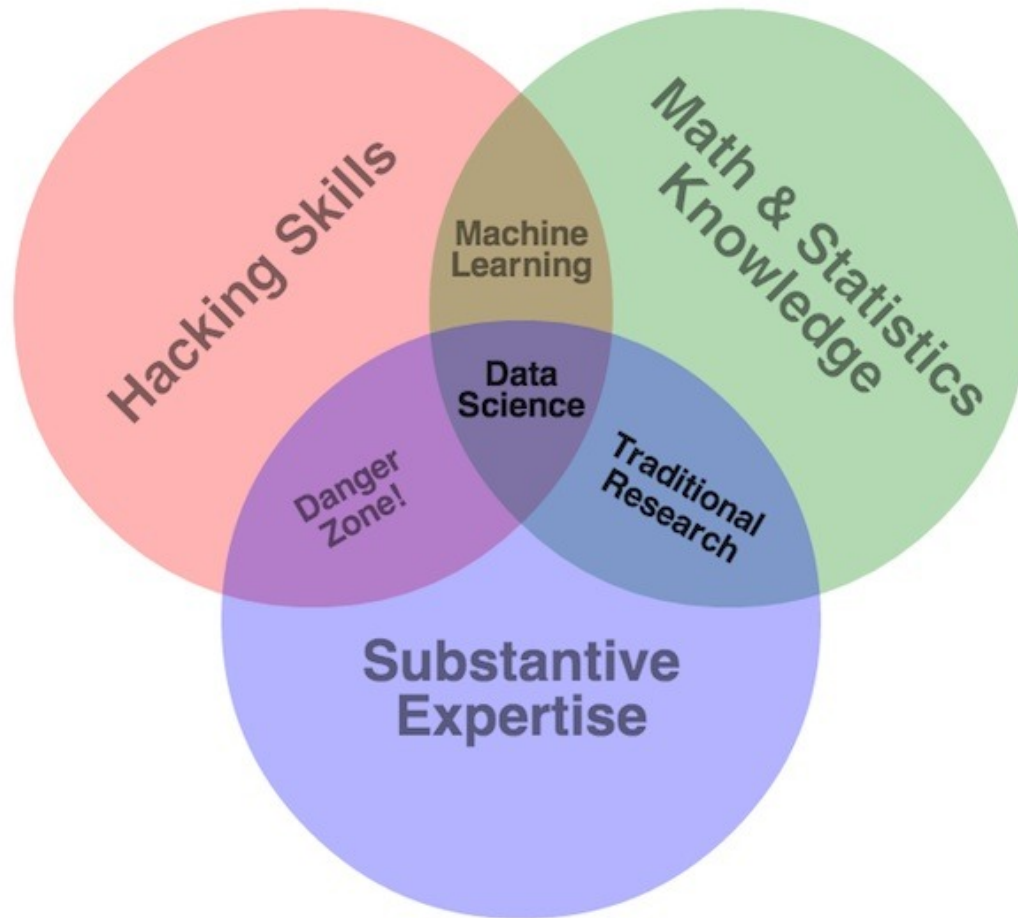
*"A billion years ago modern homo sapiens emerged. A billion minutes ago, Christianity began. A billion seconds ago, the IBM PC was released. A billion Google searches ago ... was this morning."*

Hal Varian (2013)

The 4 Vs of big data:

- Volume - Scale of data.
- Velocity - Analysis of streaming data.
- Variety - Different forms of data.
- Veracity - Uncertainty of data.

# "Data Science"



[\*] Hacking  $\approx$  coding

# Outline

1. What is ML?
2. The problem of overfitting
3. Too complex? Regularize!
4. ML and Econometrics

What is ML?

# So, what is ML?

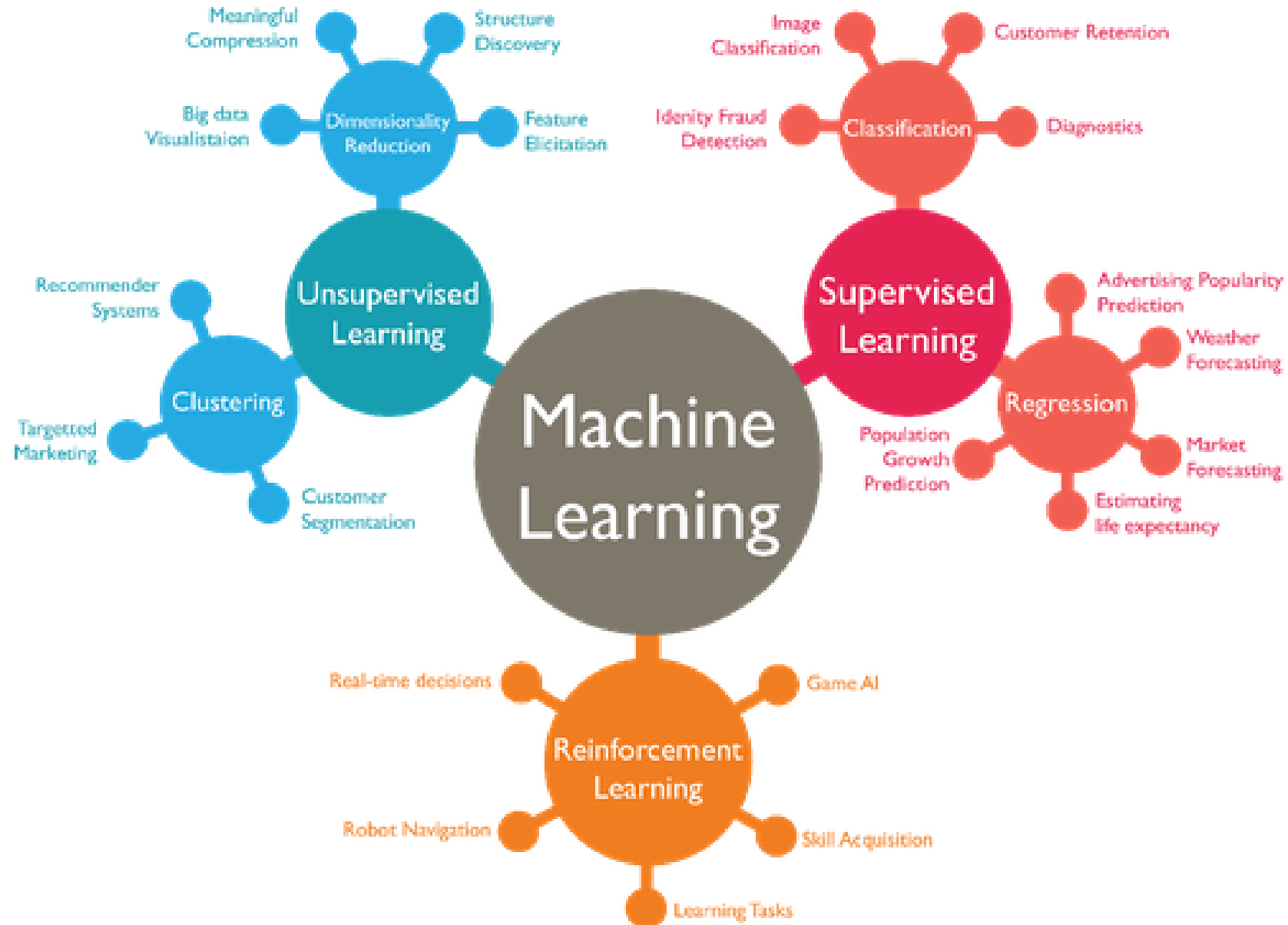
A concise definition by [Athey \(2018\)](#):

"...[M]achine learning is a field that develops algorithms designed to be applied to datasets, with the main areas of focus being prediction (regression), classification, and clustering or grouping tasks."

Specifically, there are three broad classifications of ML problems:

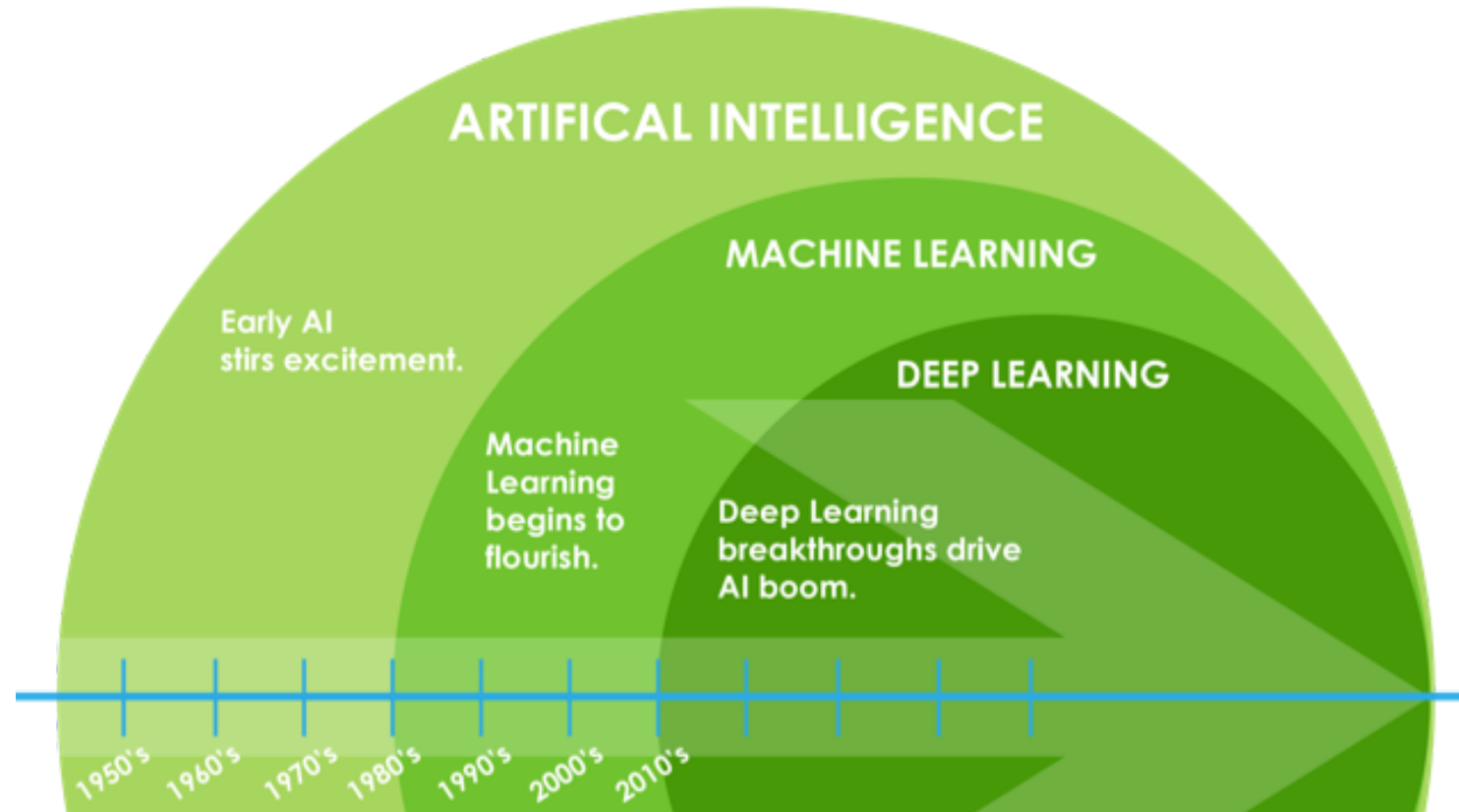
- supervised learning.
- unsupervised learning.
- reinforcement learning.

Most of the hype you hear about in recent years relates to supervised learning, and in particular, deep learning.





# An Aside: ML and Artificial Intelligence (AI)



# Unsupervised Learning

In *unsupervised* learning, the goal is to divide high-dimensional data into clusters that are **similar** in their set of features ( $X$ ).

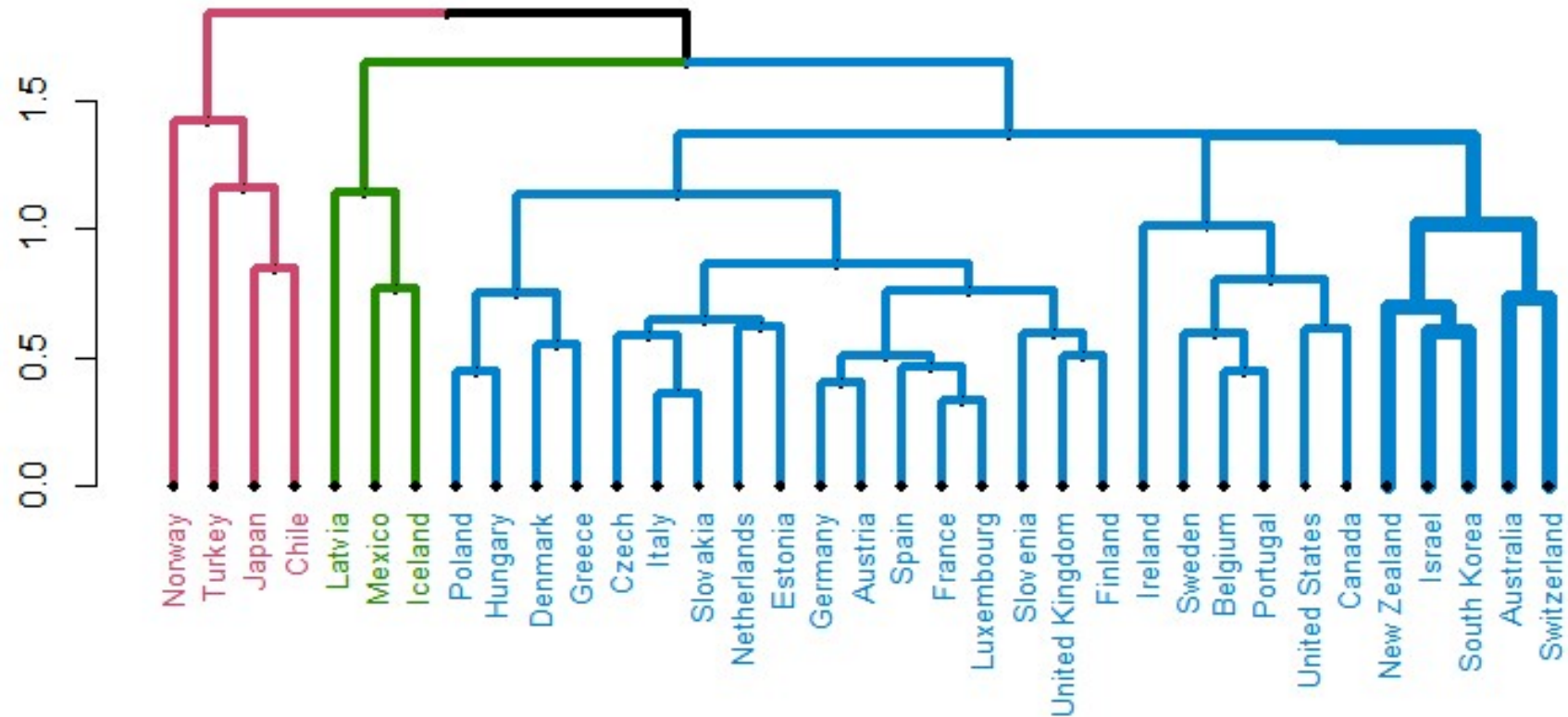
Examples of algorithms:

- principal component analysis
- $k$ -means clustering
- Latent Dirichlet Allocation (LDA)

Applications:

- image recognition
- cluster analysis
- topic modelling

# Example: Clustering OECD Inflation Rates



Source: Baudot-Trajtenberg and Caspi (2018).

# Reinforcement Learning (RL)

A definition by [Sutton and Barto \(2018\)](#):

*"Reinforcement learning is learning what to do—how to map situations to actions—so as to maximize a numerical reward signal. The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them."*

Prominent examples:

- Games (e.g., Chess, AlphaGo).
- Autonomous cars.

# Supervised Learning

Consider the following data generating process (DGP):

$$Y = f(\mathbf{X}) + \epsilon$$

where  $Y$  is the outcome variable,  $\mathbf{X}$  is a  $1 \times p$  vector of "features", and  $\epsilon$  is the irreducible error.

- **Training set** ("in-sample"):  $\{(x_i, y_i)\}_{i=1}^n$
- **Test set** ("out-of-sample"):  $\{(x_i, y_i)\}_{i=n+1}^m$



Typical assumptions: (1) independent observations; (2) stable DGP across training and test sets.

# The Goal of Supervised Learning

Use a labelled test set (  $X$  and  $Y$  are known) to construct  $\hat{f}(X)$  such that it *generalizes* to unseen test set (only  $X$  is known).

**EXAMPLE:** Consider the task of spam detection:

ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
ham	Ok lar... Joking wif u oni...
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
ham	U dun say so early hor... U c already then say...
ham	Nah I don't think he goes to usf, he lives around here though
spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, å£1.50 to rcv
ham	Even my brother is not like to speak with me. They treat me like aids patent.
ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune
spam	WINNER!! As a valued network customer you have been selected to receivea å£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030

In this case,  $Y = \{spam, ham\}$ , and  $X$  is the email text.

# Traditional vs. Modern Approach to Supervised Learning



# Traditional vs. Modern Approach to Supervised Learning

- Traditional: rules based, e.g., define dictionaries of "good" and "bad" words, and use it to classify text.
- Modern: learn from data, e.g., label text as "good" or "bad" and let the model estimate rules from (training) data.



# More Real World Applications of ML

task	outcome (Y)	features (X)
credit score	probability of default	loan history, payment record...
fraud detection	fraud / no fraud	transaction history, timing, amount...
voice recognition	word	recordings
sentiment analysis	good / bad review	text
image classification	cat / not cat	image
overdraft prediction	yes / no	bank-account history
customer churn prediction	churn / no churn	customer features

A rather mind blowing example: Amazon's **"Anticipatory Package Shipping"** patent (December 2013): Imagine Amazon's algorithms reaching such levels of accuracy, casing it to change its business model from shopping-then-shipping to shipping-then-shopping!

# Supervised Learning Algorithms

ML comes with a rich set of parametric and non-parametric prediction algorithms (approximate year of discovery in parenthesis):

- Linear and logistic regression (1805, 1958).
- Decision and regression trees (1984).
- K-Nearest neighbors (1967).
- Support vector machines (1990s).
- Neural networks (1940s, 1970s, 1980s, 1990s).
- Simulation methods (Random forests (2001), bagging (2001), boosting (1990)).
- etc.

# Recent Application

## Forecasting CPI Inflation Components with Hierarchical Recurrent Neural Networks

Oren Barkan<sup>b</sup>, Jonathan Benchimol<sup>c</sup>, Itamar Caspi<sup>c</sup>, Allon Hammer<sup>d,1</sup>, Noam Koenigstein<sup>d,1</sup>

<sup>a</sup>*Affiliation*

<sup>b</sup>*Ariel University*

<sup>c</sup>*Bank of Israel*

<sup>d</sup>*Tel-Aviv University*

---

### Abstract

We present a hierarchical architecture based on Recurrent Neural Networks (RNNs) for predicting disaggregated inflation components of the Consumer Price Index (CPI). While the majority of existing research is focused on predicting headline inflation, many economic and financial institutions are interested in its partial disaggregated components. To this end, we developed the novel Hierarchical Recurrent Neural Network (HRNN) model, which utilizes information from higher levels in the CPI hierarchy to improve predictions at the more volatile lower levels. Based on a large dataset from the US CPI-U index, our evaluations indicate that the HRNN model significantly outperforms a vast array of well-known inflation prediction baselines. Our methodology and results provide additional forecasting measures and possibilities to policy and market makers on sectoral and component-specific price changes.

Source: [Barkan et al. \(2021\)](#)

# So, Why Now?

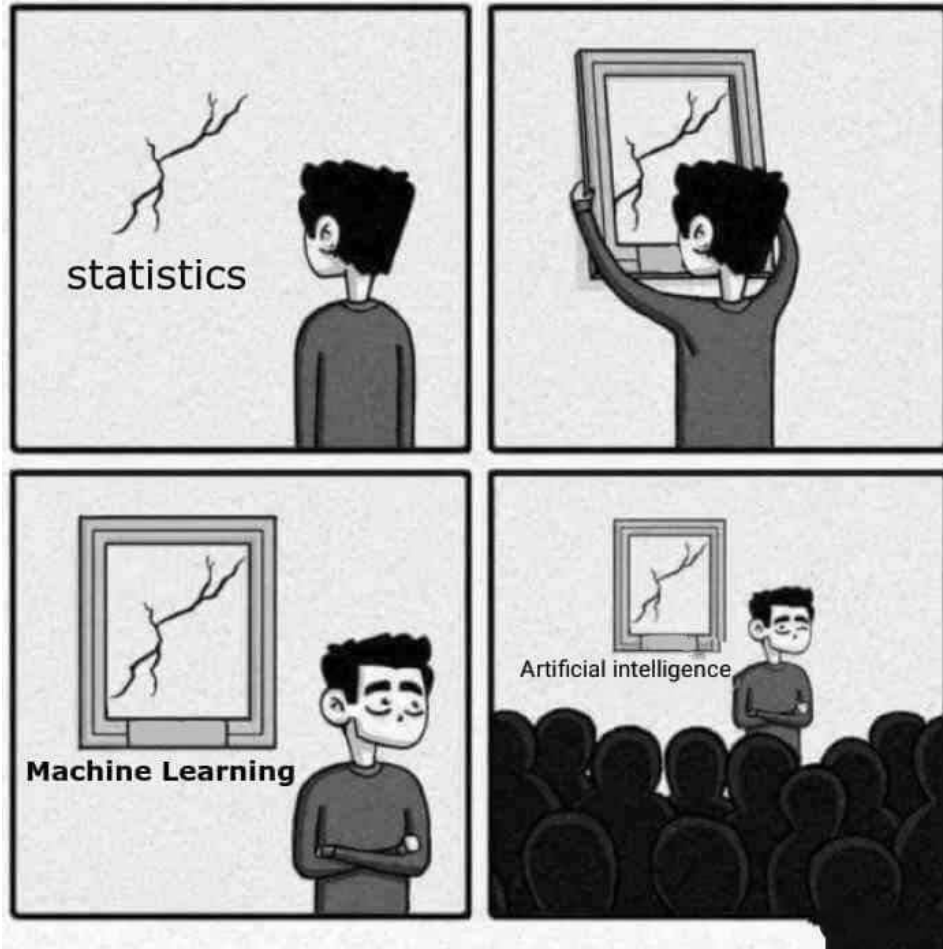
- ML methods are data-hungry and computationally expensive. Hence,  
 $\text{big data} + \text{computational advancements} = \text{the rise of ML}$
- Nevertheless,

*"[S]upervised learning [...] may involve high dimensions, non-linearities, binary variables, etc., but at the end of the day it's still just regression."*

— **Francis X. Diebold**



# Wait, is ML Just Glorified Statistics?



The "two cultures" ([Breiman, 2001](#)):

- Statisticians assume a data generating process and try to learn about it using data (parameters, confidence intervals, assumptions.)
- Computer scientists treat the data mechanism as unknown and try to predict or classify with the most accuracy.

# Overfitting

# Prediction Accuracy

Before we define overfitting, we need to be more explicit about what we mean by "good prediction."

- Let  $(x^0, y^0)$  denote a single realization from the (unseen) test set.
- Define a **loss function**  $L$  in terms of predictions  $\hat{y}^0 = \hat{f}(x^0)$  and the "ground truth"  $y^0$ , where  $\hat{f}$  is estimated using the *training* set.
- Examples
  - squared error (SE):  $L(\hat{y}^0, y^0) = (y^0 - \hat{f}(x^0))^2$
  - absolute error (AE):  $L(\hat{y}^0, y^0) = |y^0 - \hat{f}(x^0)|$
- There are other possible forms of loss function (e.g., in classification, as the probability of misclassifying a case, or in terms of economic costs.)

# The Bias-Variance Decomposition

Under a **squared error loss function**, an optimal predictive model is one that minimizes the *expected* squared prediction error.

It can be shown that if the true model is  $Y = f(X) + \epsilon$ , then

$$\begin{aligned}\mathbb{E} [\text{SE}^0] &= \mathbb{E} [(y^0 - \hat{f}(x^0))^2] \\ &= \underbrace{\left( \mathbb{E}(\hat{f}(x^0)) - f(x^0) \right)^2}_{\text{bias}^2} + \underbrace{\mathbb{E} [\hat{f}(x^0) - \mathbb{E}(\hat{f}(x^0))]^2}_{\text{variance}} + \underbrace{\mathbb{E} [y^0 - f(x^0)]^2}_{\text{irreducible error}} \\ &= \underbrace{\text{Bias}^2 + \mathbb{V}[\hat{f}(x^0)]}_{\text{reducible error}} + \sigma_\epsilon^2\end{aligned}$$

where the expectation is over the training set *and*  $(x^0, y^0)$ .



# Intuition Behind the Bias Variance Trade-off

Imagine you are a teaching assistant grading exams. You grade the first exam. What is your best prediction of the next exam's grade?

- the first test score is an unbiased estimator of the mean grade.
- but it is extremely variable.
- any solution?

Lets simulate it!

# Exam Grade Prediction Simulation

Let's Draw 1000 grade duplets from the following truncated normal distribution

$$g_i \sim \text{truncN}(\mu = 75, \sigma = 15, a = 0, b = 100), \quad i = 1, 2$$

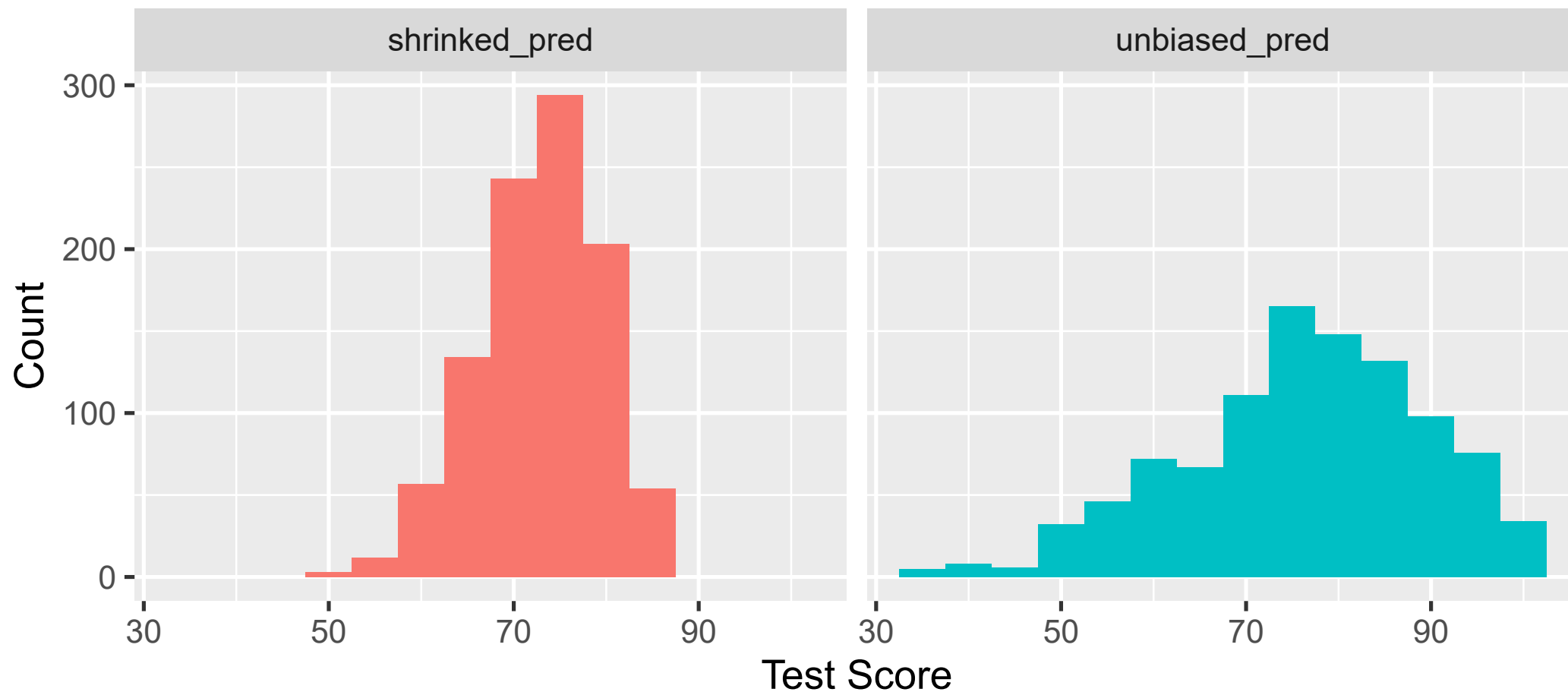
Next, calculate two types of predictions

- unbiased\_pred is the first exam's grade.
- shrinked\_pred is an average of the previous grade and a *prior* mean grade of 70.

Here a small sample from our simulated table:

attempt	grade1	grade2	unbiased_pred	shrinked_pred
921	83	82	83	76.5
27	62	66	62	66.0
165	77	67	77	73.5
841	69	47	69	69.5
781	72	77	72	71.0

# The Distribution of Predictions



# The MSE of Grade Predictions

unbiased_MSE	shrunked_MSE
327.573	215.348

Hence, the shrunk prediction turns out to be better (in the sense of MSE) than the unbiased one!

**QUESTION:** Is this a general result? Why?

# Regressions and the Bias-Variance Trade-off

Consider the following hypothetical DGP:

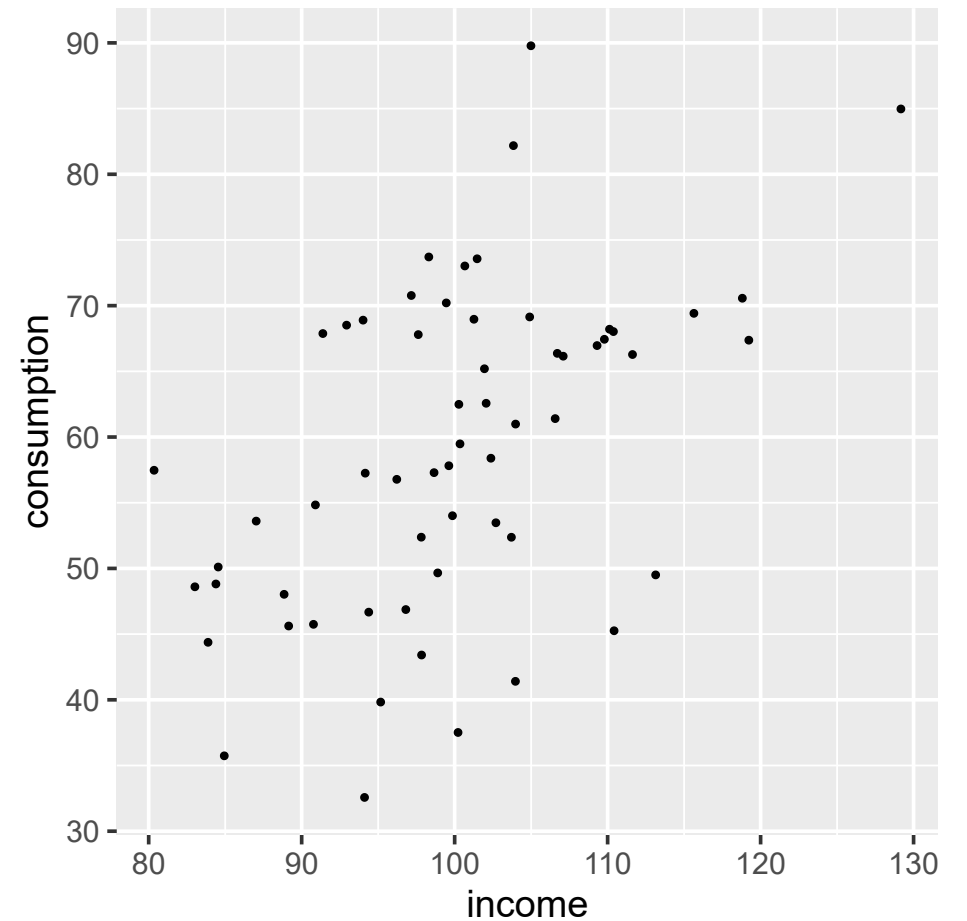
$$consumption_i = \beta_0 + \beta_1 \times income_i + \varepsilon_i$$

```
set.seed(1505) # for replicating the simulation

df <- crossing(economist = c("A", "B", "C"),
               obs = 1:20) %>%
  mutate(economist = as.factor(economist)) %>%
  mutate(income = rnorm(n(), mean = 100, sd = 10)) %>%
  mutate(consumption = 10 + 0.5 * income + rnorm(n(), sd = 10))
```

# Scatterplot of the Data

```
df %>%  
  ggplot(aes(y = consumption,  
             x = income)) +  
  geom_point()
```

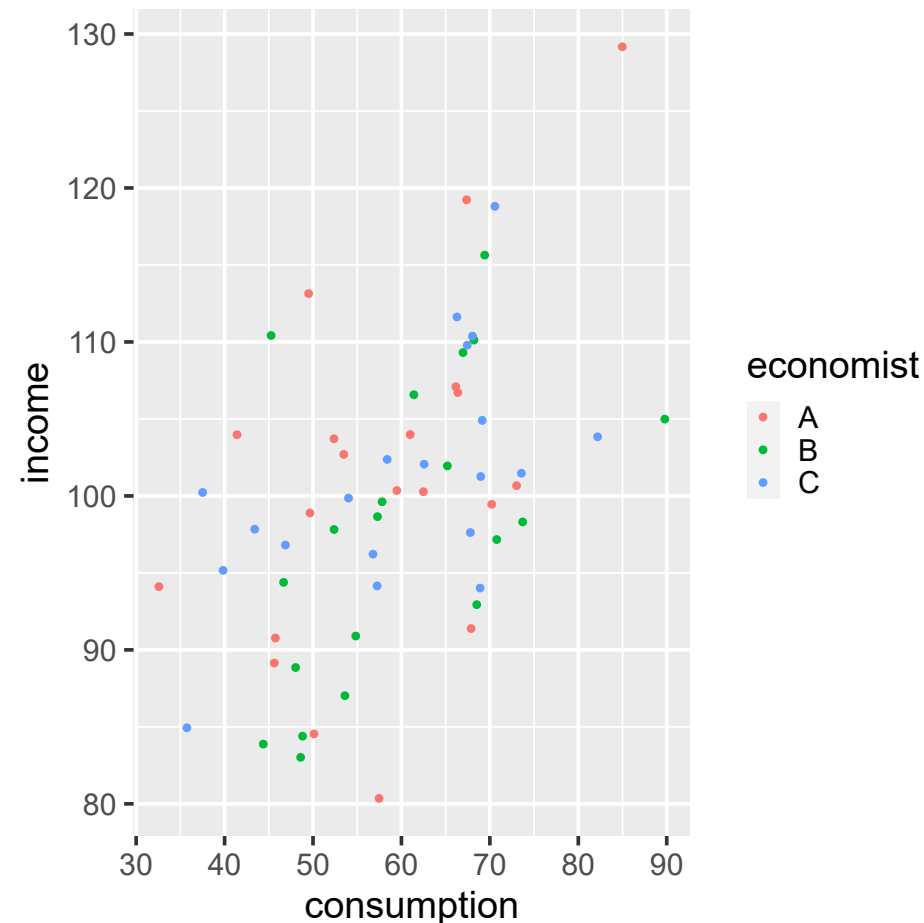


# Split the Sample Between Three Economists

```
df %>%  
  ggplot(aes(x = consumption,  
             y = income,  
             color = economist)) +  
  geom_point()
```

```
knitr::kable(sample_n(df,6), format = "html")
```

economist	obs	income	consumption
C	15	96.21703	56.77998
A	7	100.34957	59.47988
A	19	80.35239	57.47065
B	5	110.42183	45.25598
A	13	91.38508	67.87680
C	17	101.26113	68.96492

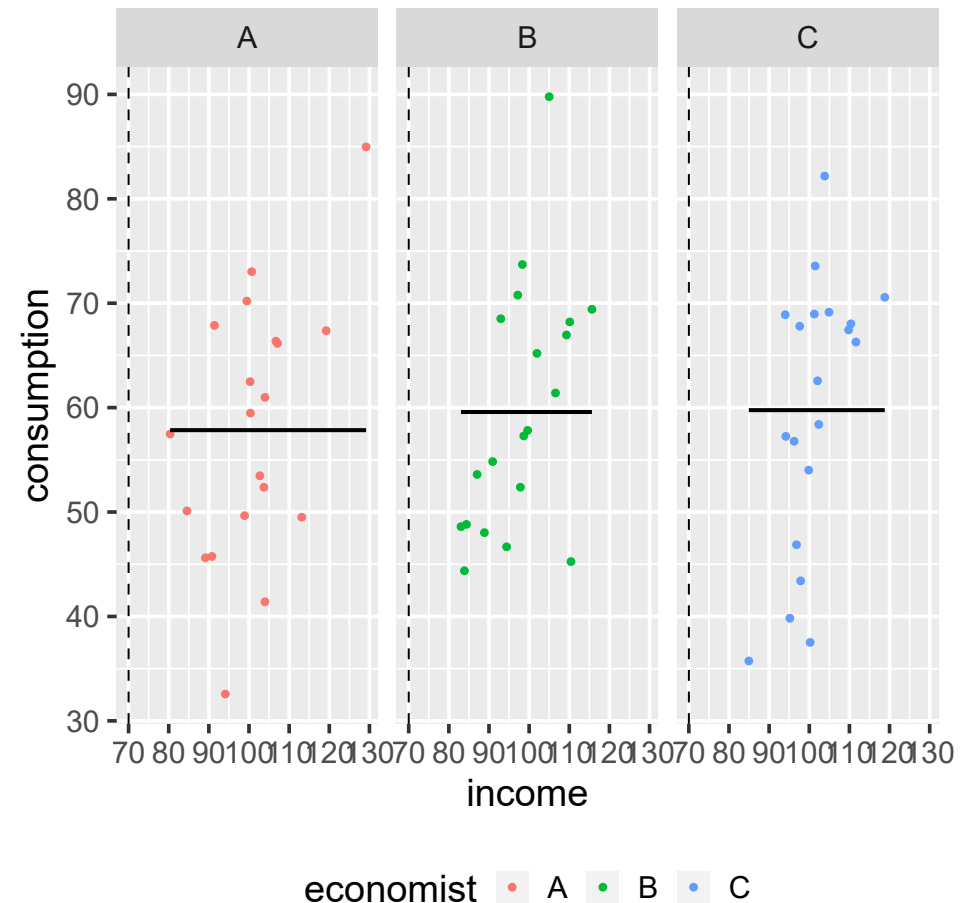


# Underfitting: High Bias, Low Variance

The model: unconditional mean

$$Y_i = \beta_0 + \varepsilon_i$$

```
df %>%  
  ggplot(aes(y = consumption,  
             x = income,  
             color = economist)) +  
  geom_point() +  
  geom_smooth(method = lm,  
             formula = y ~ 1,  
             se = FALSE,  
             color = "black") +  
  facet_wrap(~ economist) +  
  geom_vline(xintercept = 70, linetype = "dashed") +  
  theme(legend.position = "bottom")
```



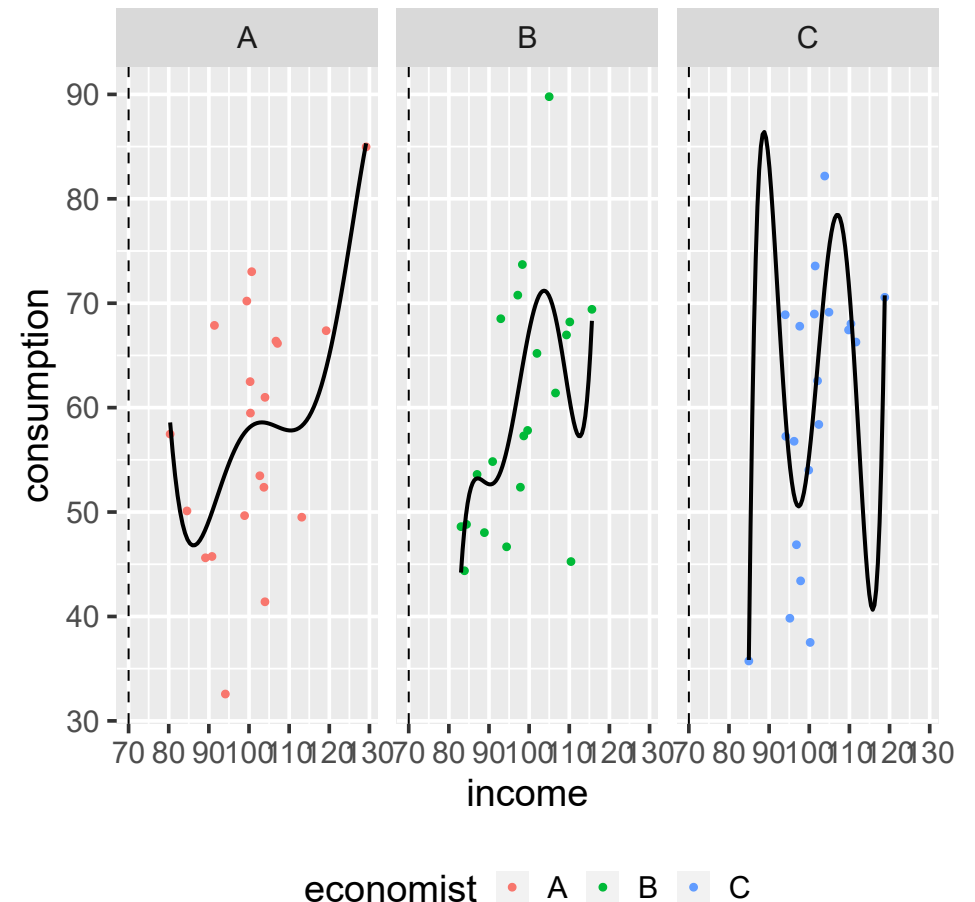


# Overfitting: Low Bias, High Variance

The model: high-degree polynomial

$$Y_i = \beta_0 + \sum_{j=1}^{\lambda} \beta_j X_i^j + \varepsilon_i$$

```
df %>%  
  ggplot(aes(y = consumption,  
             x = income,  
             color = economist)) +  
  geom_point() +  
  geom_smooth(method = lm,  
             formula = y ~ poly(x,5),  
             se = FALSE,  
             color = "black") +  
  facet_wrap(~ economist) +  
  geom_vline(xintercept = 70, linetype = "dashed") +  
  theme(legend.position = "bottom")
```

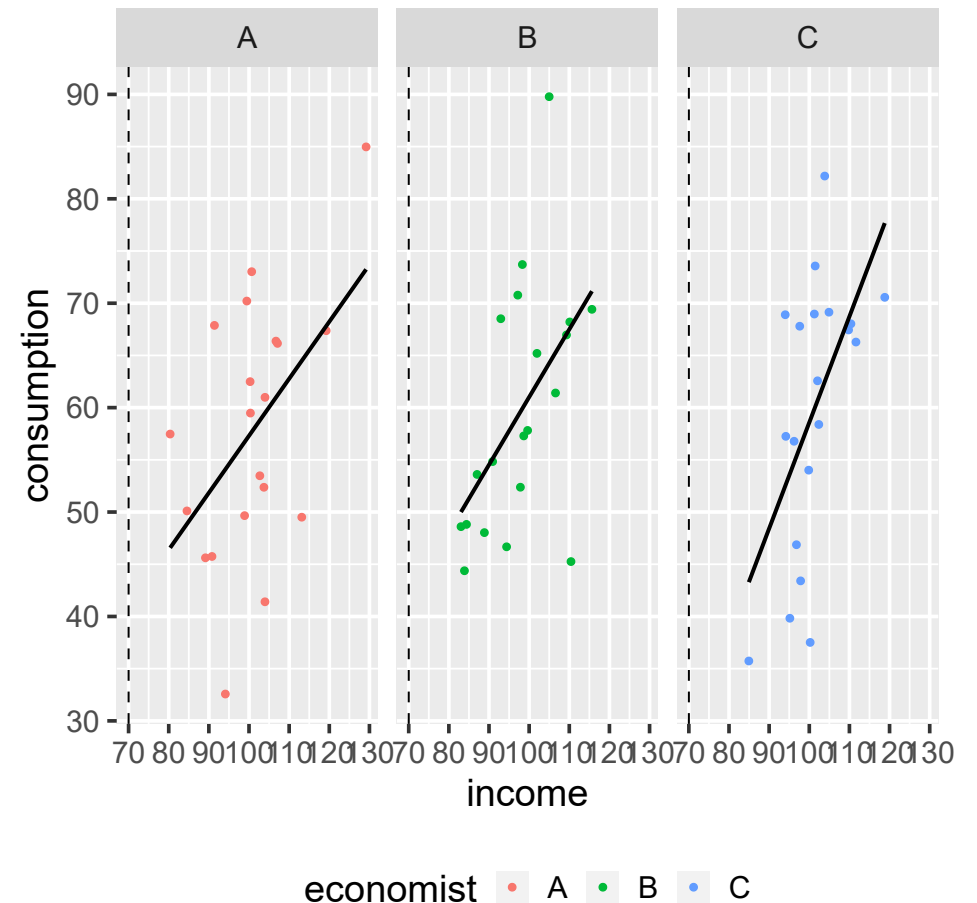


# "Justfitting": Bias and Variance are Just Right

The model: linear regression

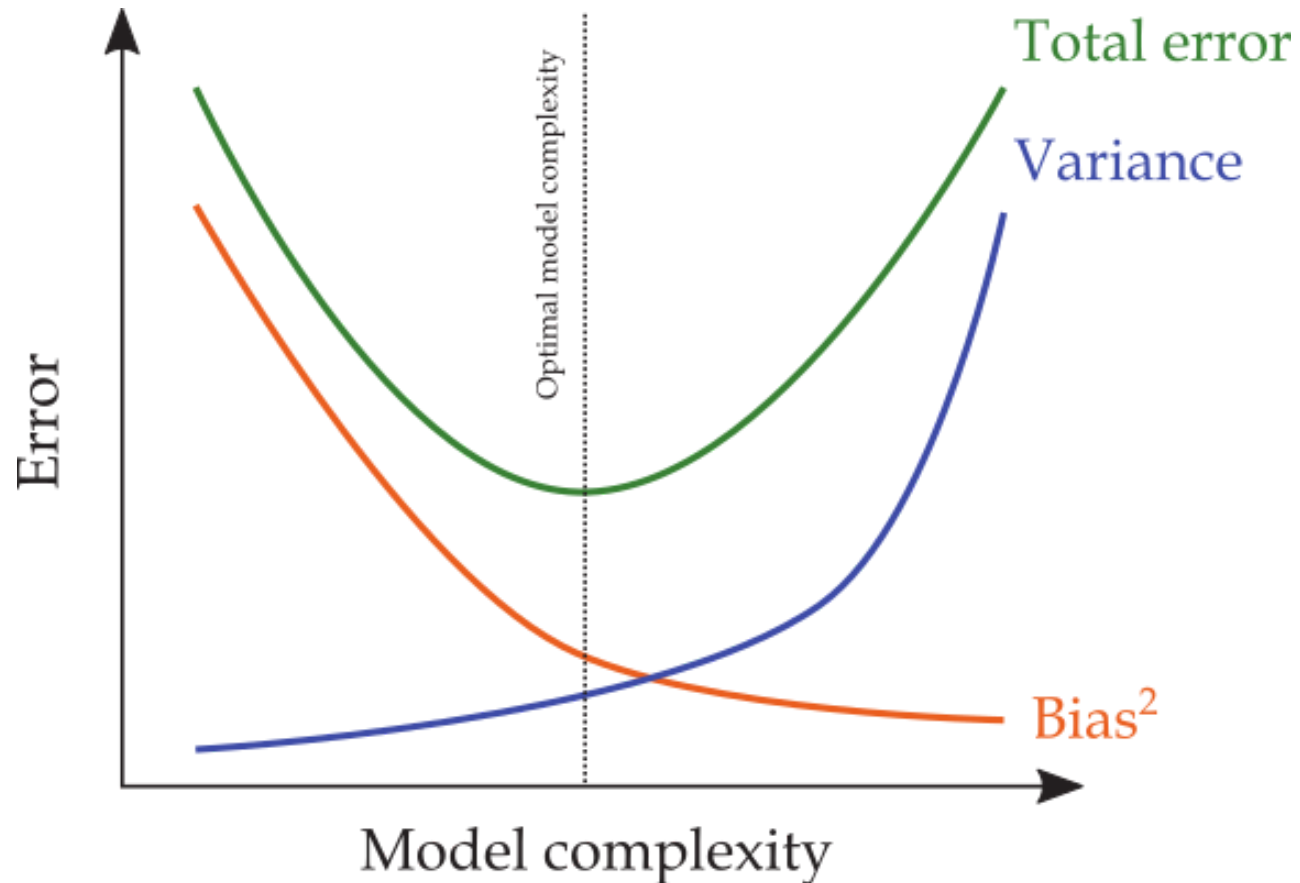
$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

```
df %>%  
  ggplot(aes(y = consumption,  
             x = income,  
             color = economist)) +  
  geom_point() +  
  geom_smooth(method = lm,  
             formula = y ~ x,  
             se = FALSE,  
             color = "black") +  
  facet_wrap(~ economist) +  
  geom_vline(xintercept = 70, linetype = "dashed") +  
  theme(legend.position = "bottom")
```



# The Typical Bias-Variance Trade-off in ML

Typically, ML models strive to find levels of bias and variance that are "just right":



# When is the Bias-Variance Trade-off Important?

In low-dimensional settings (  $n \gg p$  )

- overfitting is highly **unlikely**
- training MSE closely approximates test MSE
- conventional tools (e.g., OLS) will perform well on a test set

INTUITION: As  $n \rightarrow \infty$ , insignificant terms will converge to their true value (zero).

In high-dimensional settings (  $n \ll p$  )

- overfitting is highly **likely**
- training MSE poorly approximates test MSE
- conventional tools tend to overfit

$n \ll p$  is prevalent in big-data

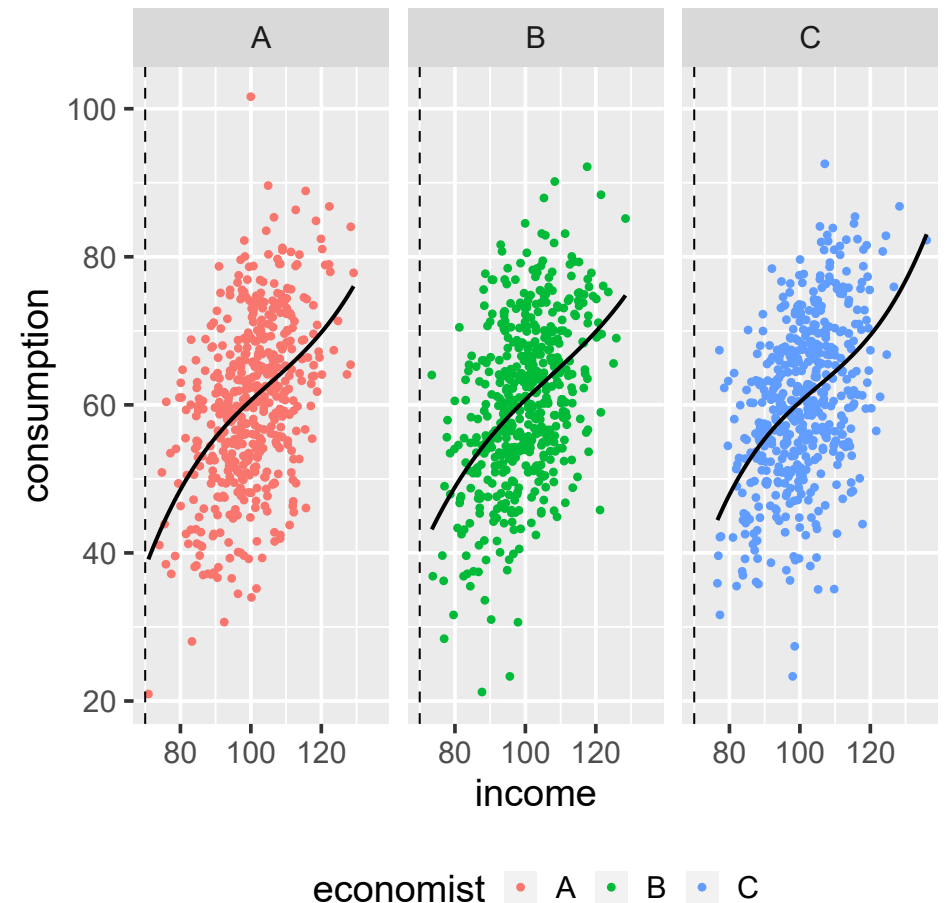
# Bias-Variance Trade-off in Low-dimensional Settings

The model is a 3rd degree polynomial

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \varepsilon_i$$

only now, the sample size for each economist increases to  $N = 500$ .

**INTUITION:** as  $n \rightarrow \infty$ ,  $\hat{\beta}_2$  and  $\hat{\beta}_3$  converge to their true value, zero.



# Regularization

# Regularization

- As the *complexity* of our model goes up, it will tend to overfit.
- **Regularization** is the act of penalizing models for their complexity level.

Regularization typically results in simpler and more accurate (though “wrong”) models.

# How to Penalize Overfit?

The test set MSE is *unobservable*. How can we tell if a model overfits?

- Main idea in machine learning: use *less* data!
- Key point: fit the model to a subset of the training set, and **validate** the model using the subset that was *not* used to fit the model.
- How can this "magic" work? Recall the stable DGP assumption.



# Validation

Split the sample to three folds: a training set, a validation set and a test set:



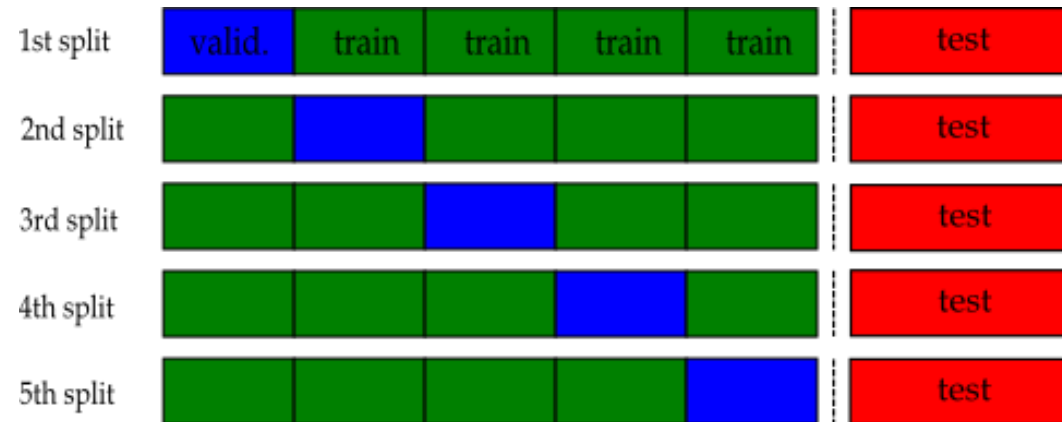
The algorithm:

1. Fit a model to the training set.
2. Use the model to predict outcomes from the validation set.
3. Use the mean of the squared prediction errors to approximate the test-MSE.

**CONCERNS:** (1) the algorithm might be sensitive to the choice of training and validation set; (2) the algorithm does not use all of the available information.

# k-fold Cross-validation

Split the training set into  $k$  roughly equal-sized parts (  $k = 5$  in this example):



Approximate the test-MSE using the mean of  $k$  split-MSEs

$$\text{CV-MSE} = \frac{1}{k} \sum_{j=1}^k \text{MSE}_j$$

# Which Model to Choose?

- Recall that the test-MSE is unobservable.
  - CV-MSE is our best guess.
  - CV-MSE is also a function of model complexity.
  - Hence, model selection amounts to choosing the complexity level that minimizes CV-MSE.
-

# Sounds Familiar?

- In a way, you probably already know this:
  - Adjusted  $R^2$ .
  - AIC, BIC (time series models).

The above two measures indirectly take into account the overfitting that may occur due to the complexity of the model (i.e., adding too many covariates or lags).

- In ML we use the data to tune the level of complexity such that it maximizes prediction accuracy.
-

# ML and Causality

# ML vs. econometrics

Apart from jargon (Training set vs. in-sample, test-set vs. out of sample, learn vs. estimate, etc.) here is a summary of some of the key differences between ML and econometrics:

Machine Learning	Econometrics
prediction	causal inference
$\hat{Y}$	$\hat{\beta}$
minimize prediction error	unbiasedness, consistency, efficiency
---	statistical inference
stable environment	counterfactual analysis
black-box	structural

# ML and causal inference

*"Data are profoundly dumb about causal relationships."*

— **Pearl and Mackenzie, *The Book of Why***

Intervention violates the stable DGP assumption:

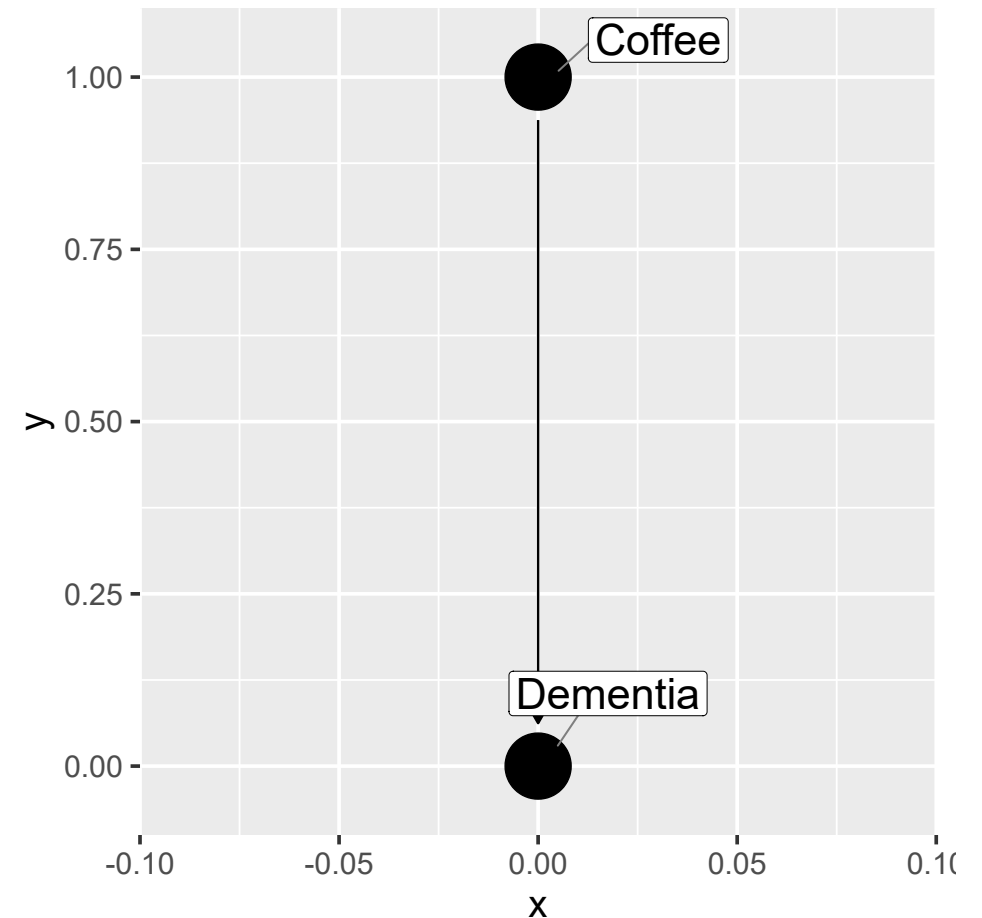
$$P(Y|X = x) \neq P(Y|do(X = x)),$$

where

- $P(Y|X = x)$  is the the probability of  $Y$  given that we *observe*  $X = x$ .
- $P(Y|do(X = x))$  is the the probability of  $Y$  given that we *intervene* to set  $X = x$

# "A new study shows that..."

**TOY PROBLEM:** Say that we find in the data that coffee is a good predictor of dementia. Is avoiding coffee a good idea?

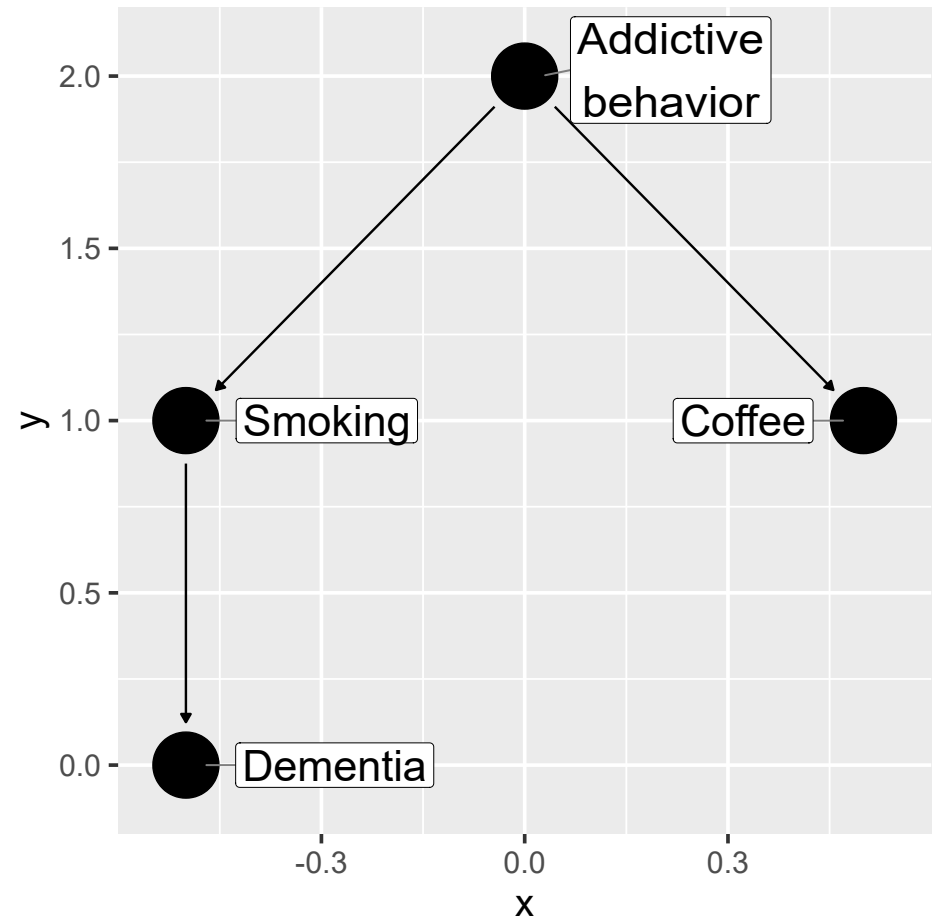




# To explain or to predict?

- In this example coffee is a good *predictor* of dementia, despite not having any causal link to it.
- Controlling for smoking will give us the causal effect of coffee on dementia, which is zero.

In general, causal inference always and everywhere depends on *assumptions* about the DGP (i.e., the data never "speaks for itself").



# ML in aid of econometrics

Consider the standard "treatment effect regression":

$$Y_i = \alpha + \underbrace{\tau D_i}_{\text{low dimensional}} + \underbrace{\sum_{j=1}^p \beta_j X_{ij}}_{\text{high dimensional}} + \varepsilon_i, \quad \text{for } i = 1, \dots, n$$

where

- An outcome  $Y_i$
- A treatment assignment  $D_i \in \{0, 1\}$
- A vector of  $p$  control variables  $X_i$

Our object of interest is often  $\hat{\tau}$ , the average treatment effect (ATE).

```
slides::end()
```

 [Source code](#)

# References

- [1] A. Agrawal, J. Gans, and A. Goldfarb. *Prediction Machines: The Simple Economics of Artificial Intelligence*. Harvard Business Review Press, 2018.
- [2] S. Athey. "The impact of machine learning on economics". In: *The Economics of Artificial Intelligence: An Agenda*. University of Chicago Press, 2018.
- [3] L. Breiman. "Statistical modeling: The two cultures (with comments and a rejoinder by the author)". In: *Statistical science* 16.3 (2001), pp. 199-231.
- [4] G. Shmueli. "To explain or to predict?" In: *Statistical science* 25.3 (2010), pp. 289-310.
- [5] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

# References