Week 1 Lectures

Part I: Introduction to data mining
Part II: Regression

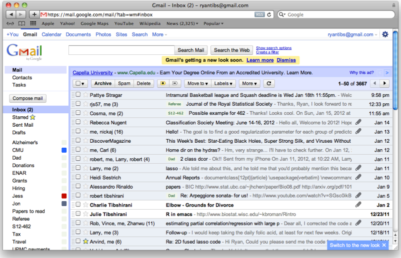
Prof. Alexandra Chouldechova
95-791: Data Mining

Spring 2019

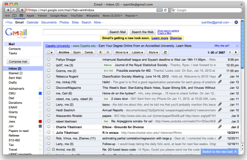
What is data mining?

Data mining is the science of discovering structure and making predictions in large or complex data sets

Spam filtering, Fraud detection, Event detection

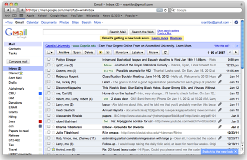


Spam filtering, Fraud detection, Outbreak detection



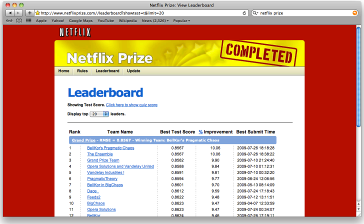
- How can we tell apart spam from real emails?
- How do we identify fraudulent transactions?
- Is the president's tweet going viral?

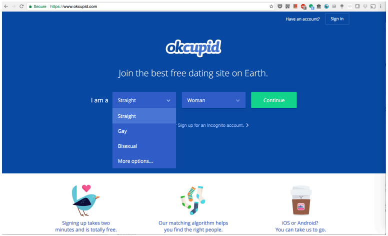
Spam filtering, Fraud detection, Outbreak detection

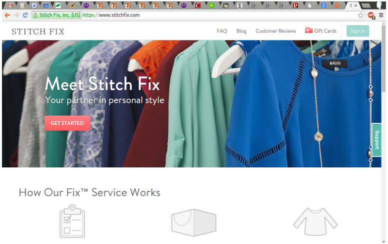


- How can we tell apart spam from real emails?
- How do we identify fraudulent transactions?
- Is the president's tweet going viral? Is the flu going viral?

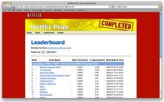
Recommendation systems

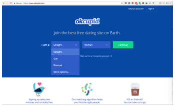


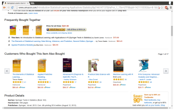


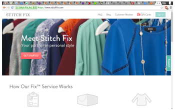


Recommendation systems



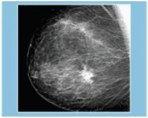






- Which movies should I recommend to my customers?
- How can I identify individuals with similar viewing/purchasing preferences?
- Which products should I recommend to my customers?
- Which promotional offers should I send out, and to whom?

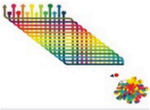
Precision medicine, health analytics



The Digital Mammography DREAM Challenge.

Spring 2016 **(Pre-registration Opens October 7)**

This Challenge, one of two large prize Coding4Cancer Challenges, seeks to improve the accuracy of breast cancer detection and reduce the current rate of patient callbacks.



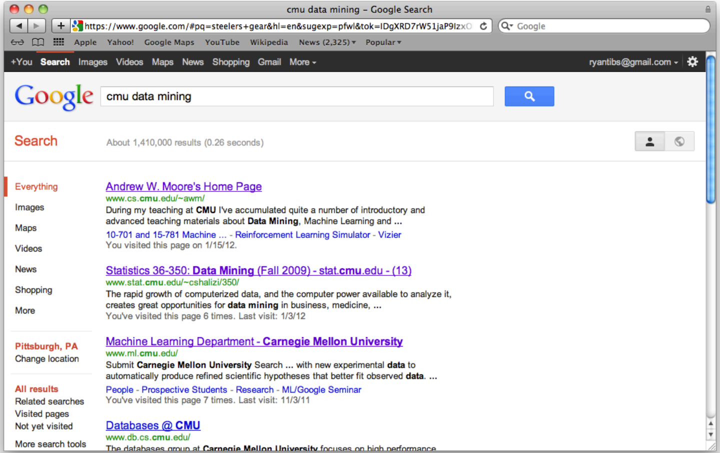
AstraZeneca-Sanger Drug Combination Prediction DREAM Challenge

Fall 2015 **(Now Open!)**

This Challenge is designed to explore fundamental traits that underlie effective combination treatments and synergistic drug behavior using baseline genomic data, i.e. data collected pretreatment.

23andMe

Search


cmu data mining – Google Search
https://www.google.com/#pq=steelers+gear&hl=en&sugexp=pfwl&tok=lDgXRD7rW51jaP9lzxO
Google
Apple Yahoo! Google Maps YouTube Wikipedia News (2,325) Popular
+You Search Images Videos Maps News Shopping Gmail More
ryantibs@gmail.com
Google
cmu data mining
Search
About 1,410,000 results (0.26 seconds)
Everything
Images
Maps
Videos
News
Shopping
More
Pittsburgh, PA
Change location
All results
Related searches
Visited pages
Not yet visited
More search tools
Andrew W. Moore's Home Page
www.cs.cmu.edu/~awm/
During my teaching at CMU I've accumulated quite a number of introductory and advanced teaching materials about Data Mining, Machine Learning and ...
10-701 and 15-781 Machine ... - Reinforcement Learning Simulator - Vizier
You visited this page on 1/15/12.
Statistics 36-350: Data Mining (Fall 2009) - stat.cmu.edu - (13)
www.stat.cmu.edu/~cshalizi/350/
The rapid growth of computerized data, and the computer power available to analyze it, creates great opportunities for data mining in business, medicine, ...
You've visited this page 6 times. Last visit: 1/3/12
Machine Learning Department - Carnegie Mellon University
www.ml.cmu.edu/
Submit Carnegie Mellon University Search ... with new experimental data to automatically produce refined scientific hypotheses that better fit observed data. ...
People - Prospective Students - Research - ML/Google Seminar
You've visited this page 7 times. Last visit: 11/3/11
Databases @ CMU
www.db.cs.cmu.edu/

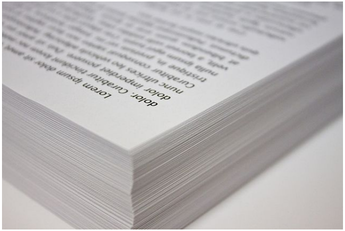
Content Tagging, Text mining

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with legos toy."

"boy is doing backflip on wakeboard."



In what sense can we think of *images* and *text* as data?

Agenda for Part I of this lecture

1. Course logistics
2. Goals and scope
3. A preview of recurring themes

Logistics: Class breakdown

There are two class components: Lecture(s) and 1 Lab session

- **Lectures**

- **Lab**: Friday 4:30PM - 5:30PM in HBH 1002
  - Hands-on data analysis practice designed to reinforce the week's lectures
  - Supervised by teaching staff
  - Attendance is mandatory

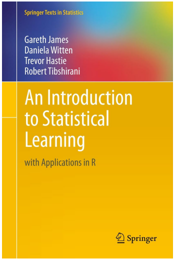
- **Teaching Staff**:
  - Instructor: Prof. Alexandra Chouldechova (HBH 2224)
  - TAs: Alton Lu, Andrew Olson, Min Heng (David) Wang, Jun Zhang

Logistics: Evaluation

- **Homework**: 5 weekly assignments — 20%
  - Due 2:50PM on Thursdays
  - Late Homework is not accepted
  - Lowest homework score gets dropped
- **Lab participation**: Friday lab attendance — 10%
  - There will be 5 regular lab sessions + 1 midterm lab session
  - Each regular lab you attend is worth 2.5 points
  - Your Lab score = $\min(10, \#\text{regular labs attended} \times 2.5)$
- **Midterm exam**: Friday during Lab — 15%
- **Final exam**: Written, closed book — 30%
- **Final project**: Team project — 25%

Logistics: Resources

- **Course website**
- **Canvas** for gradebook and turning in homework
- **Piazza** for forum
- **Required textbook:**

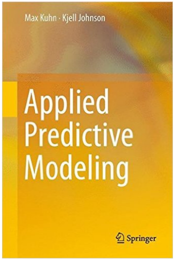


An Introduction to Statistical Learning (ISLR)
by *Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani*

- Available for FREE! here: `http://www-bcf.usc.edu/~gareth/ISL/`
- Supplementary video lectures, slides available here: `http://tinyurl.com/k7pq879`

Logistics: Resources

- **Highly recommended textbook:**

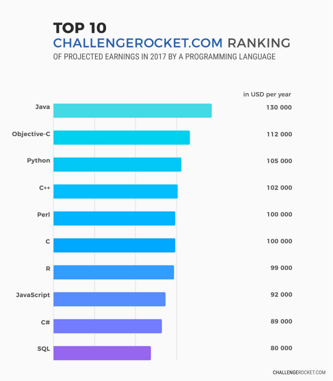


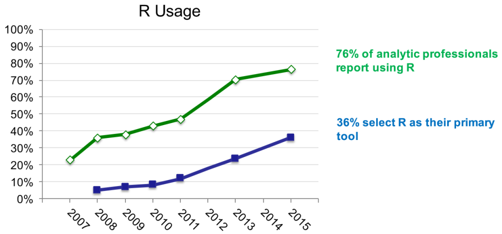
Applied Predictive Modeling (APM)
by *Max Kuhn and Kjell Johnson*

- Available for free from the CMU network through SpringerLink:
`http://tinyurl.com/zshm24z`
- SpringerLink will print you a black-and-white Softcover version for $24.99
- Supplementary materials available here:
`http://appliedpredictivemodeling.com/`

Logistics: Computing

- We'll use **R** / RStudio / R Markdown in this class

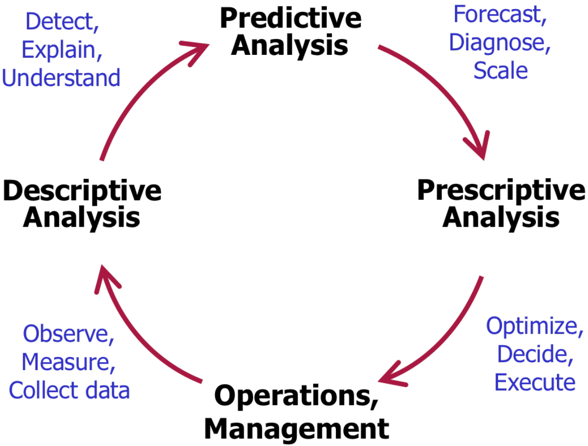




- I have posted a number of learning resources on the course website to help those of you who aren't yet familiar with R

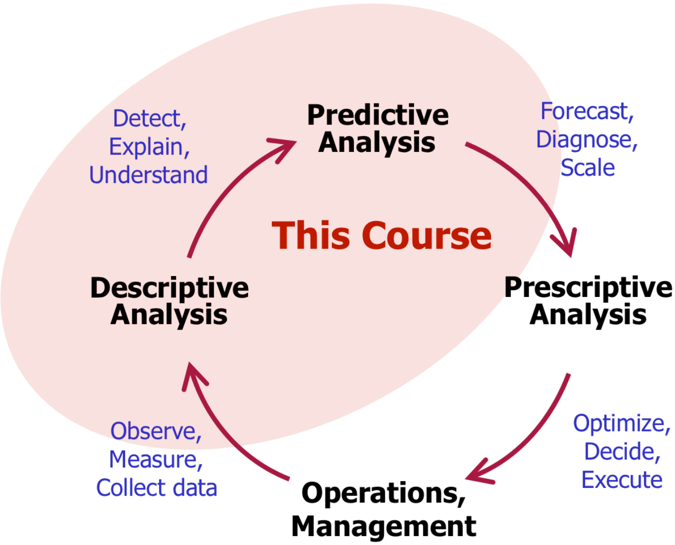
Scope of this class

The Data Analytics Cycle



Scope of this class

The Data Analytics Cycle



Thinking about Data Mining problems

Data mining problems are often divided into Predictive tasks and Descriptive tasks.

- **Predictive Analytics (Supervised learning)**:
  Given observed data $(X_1, Y_1), \ldots (X_n, Y_n)$, learn a model to predict Y from X.
  - If Y_i is a *continuous numeric* value, this task is called prediction (E.g., $Y_i =$ stock price, income, survival time)
  - If Y_i is a *discrete* or *symbolic* value, this task is called classification (E.g., $Y_i \in \{0, 1\}$, $Y_i \in$ {spam, email}, $Y_i \in \{1, 2, 3, 4\}$)

- **Descriptive Analytics (Unsupervised learning)**:

Thinking about Data Mining problems

Data mining problems are often divided into Predictive tasks and Descriptive tasks.

- **Predictive Analytics (Supervised learning)**:
  Given observed data $(X_1, Y_1), \ldots (X_n, Y_n)$, learn a model to predict Y from X.
  - If Y_i is a *continuous numeric* value, this task is called prediction (E.g., $Y_i =$ stock price, income, survival time)
  - If Y_i is a *discrete* or *symbolic* value, this task is called classification (E.g., $Y_i \in \{0, 1\}$, $Y_i \in$\{spam, email\}, $Y_i \in \{1, 2, 3, 4\}$)

- **Descriptive Analytics (Unsupervised learning)**:
  Given data $X_1, \ldots X_n$, identify some underlying patterns or structure in the data.

Thinking about Data Mining problems

Data mining problems are often divided into Predictive tasks and Descriptive tasks.

- **Predictive Analytics (Supervised learning)**:
  Q: To whom should I extend credit?
  - Task: Predict how likely an applicant is to repay loan.

  Q: What characterizes customers who are likely to churn?
  - Task: Identify variables that are predictive of churn.

  Q: How profitable will this subscription customer be?
  - Task: Predict how long customer will remain subscribed.

- **Descriptive Analytics (Unsupervised learning)**:
  - Clustering customers into groups with similar spending habits
  - Learning association rules: E.g., 50% of clients who {recently got promoted, had a baby} want to {get a mortgage}

Over the course of this class, you will:

- Become familiar with common terminology
- Gain a working understanding of many of the most widely used data mining methods
- Learn about the advantages and disadvantages of the various methods
- Gain experience implementing various methods on real data using **R**
- Learn to compare the performance of different methods and to **validate** models

You will learn about:

- **Supervised learning** methods for prediction and classification (e.g., linear models, additive models, support vector machines, generative models, tree-based methods)
- **Unsupervised learning** methods (e.g., clustering, mixture models)
- **Feature selection** and **Dimensionality reduction** (e.g., PCA, MDS, featurizing text, regularized regression)
- **Model validation and selection** (e.g., Cross-validation, training-testing, ROC, precision-recall, bootstrap, permutation methods)

Central themes of this class

Predictive analytics: What are we trying to do?

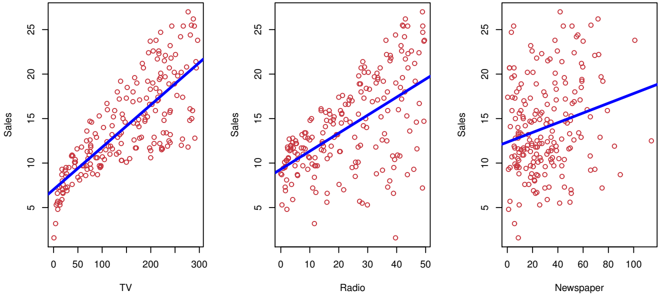
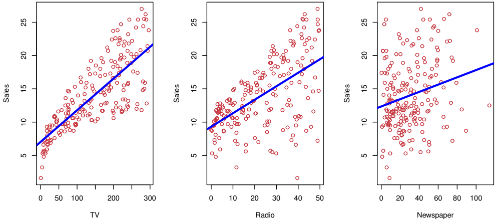


Figure: The Advertising data set, which contains data on n = 200 different markets. Each plot shows a linear regression line of Sales on the x-axis variable.

- Outcome Y_i = `Sales` in 1000's of units
- Covariates/inputs: Budgets for X_1 = `TV`, X_2 = `Radio`, and X_3 = `Newspaper` advertising budgets, in 1000's of dollars

Predictive analytics: What are we trying to do?

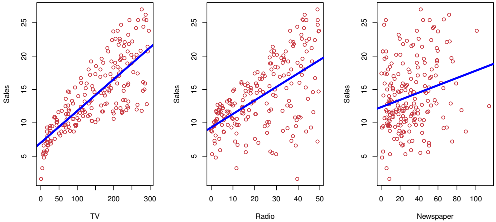


- Ideally, we would like to have a *joint* model of the form

$$\texttt{Sales} \approx f(\texttt{TV}, \texttt{Radio}, \texttt{Newspaper})$$

- We want to find a function f such that $f(\texttt{TV}, \texttt{Radio}, \texttt{Newspaper})$ is a good predictor of $\texttt{Sales}$.

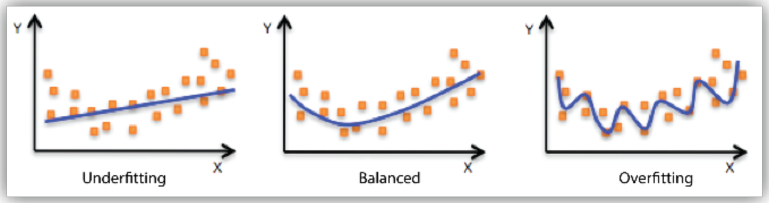
Predictive analytics: What are we trying to do?



What does it mean to be a good predictor?

Central theme 1: Generalizability

- We want to construct predictors that generalize well to unseen data
- i.e., we want predictors that:
  1. Capture useful trends in the data (don't *underfit*)
  2. Ignore meaningless random fluctuations in the data (don't *overfit*)



- We also want to avoid unjustifiably extrapolating beyond the scope of our data

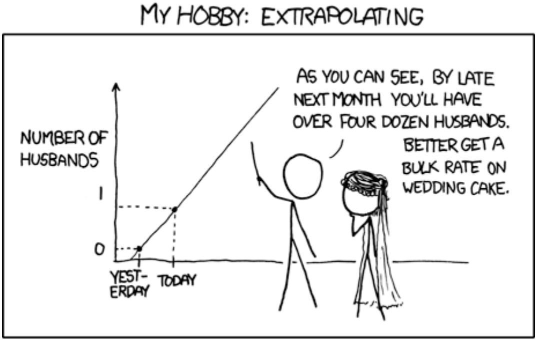
Central theme 1: Generalizability

- We want to construct predictors that generalize well to unseen data
- i.e., we want predictors that:
  1. Capture useful trends in the data (don't *underfit*)
  2. Ignore meaningless random fluctuations in the data (don't *overfit*)

- We also want to avoid unjustifiably extrapolating beyond the scope of our data

Central theme 1: Generalizability

- We want to construct predictors that generalize well to unseen data
- i.e., we want predictors that:
  1. Capture useful trends in the data (don't *underfit*)
  2. Ignore meaningless random fluctuations in the data (don't *overfit*)
- We also want to avoid unjustifiably extrapolating beyond the scope of our data



Randall Munroe, xkcd

Central theme 2: Bias-Variance Tradeoff

- We'll talk a lot about the Bias-Variance tradeoff, which relates to the fact that given a predictor $\hat{f}$,

$$\text{Expected-prediction-error}(\hat{f}) = \text{Variance}(\hat{f}) + \text{Bias}^2(\hat{f}) + \text{Noise}$$

- In the language of Theme 1:

Central theme 2: Bias-Variance Tradeoff

- We'll talk a lot about the Bias-Variance tradeoff, which relates to the fact that given a predictor $\hat{f}$,

$$\text{Expected-prediction-error}(\hat{f}) = \text{Variance}(\hat{f}) + \text{Bias}^2(\hat{f}) + \text{Noise}$$

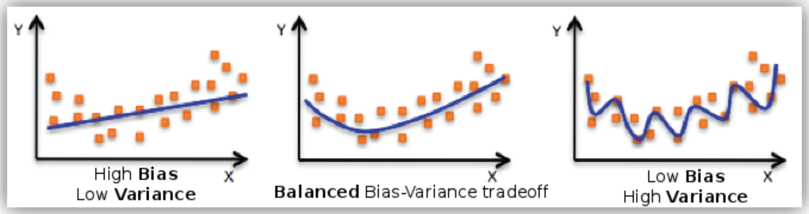
- In the language of Theme 1:

Central theme 2: Bias-Variance Tradeoff

- We'll talk a lot about the Bias-Variance tradeoff, which relates to the fact that given a predictor $\hat{f}$,

$$\text{Expected-prediction-error}(\hat{f}) = \text{Variance}(\hat{f}) + \text{Bias}^2(\hat{f}) + \text{Noise}$$

- In the language of Theme 1:



Central theme 3: Interpretability-Flexibility Tradeoff

- In this class we'll encounter both highly *structured*, interpretable models and highly flexible models
- The best predictor for a problem may turn out to be an uninterpretable or hard-to-interpret **black box**
- Depending on the purpose of the prediction, we may prefer a more interpretable, worse-performing model to a better-performing "black box".

Central theme 3: Interpretability-Flexibility Tradeoff

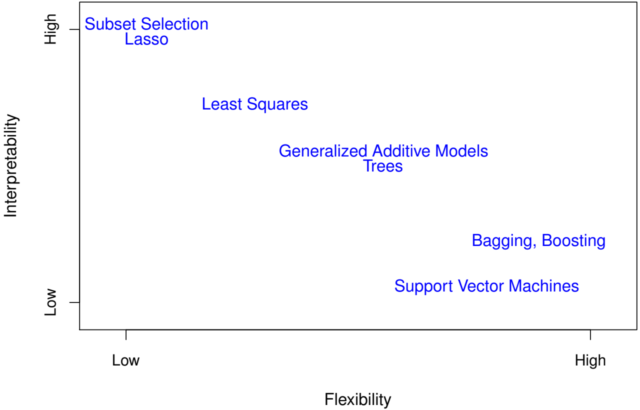


Figure: 2.7 from ISLR

Central theme 4: Feature engineering

"…some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used."
— Pedro Domingos, "A Few Useful Things to Know about Machine Learning"

"Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data."
— Jason Brownlee, Machine Learning Mastery

Central theme 4: Feature engineering

- Given unlimited data, sufficiently flexible models will be able to learn nearly arbitrarily complex patterns and structures.
- In reality, we have a limited number of observations, and often a large number of variables.
- We'll see that we can improve the performance of methods by constructing better features

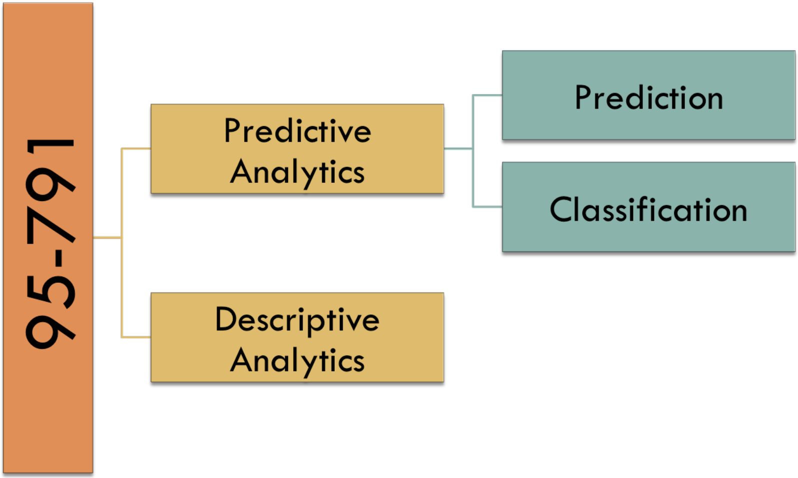
End of Part I

10 minute break

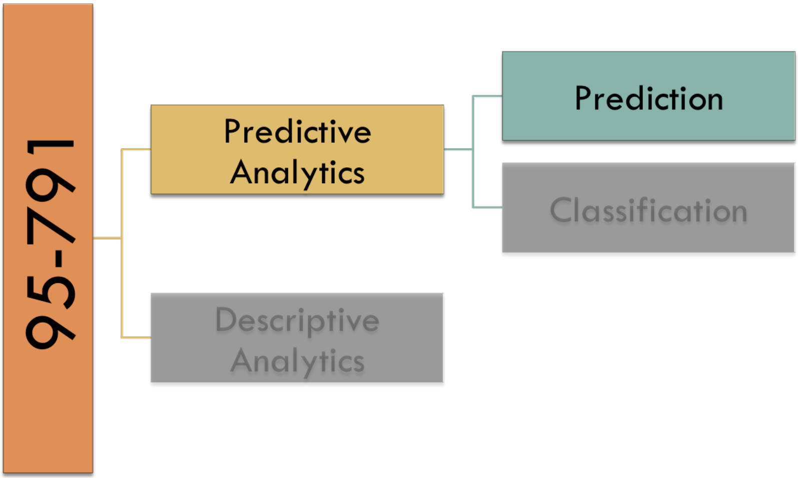
Agenda for Part II

- Prediction setup, terminology, notation
- What are models good for?
- What does it mean to "predict Y"?
- Methods: Linear and Additive models

Course Roadmap



Course Roadmap



What is the prediction task?

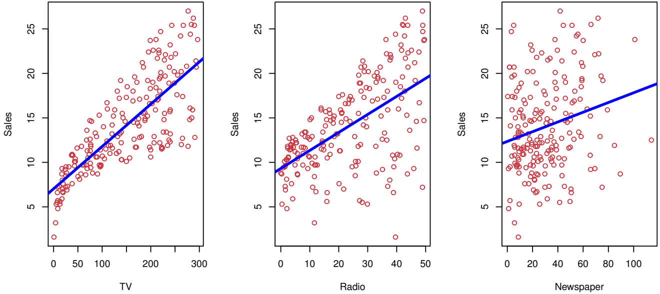


Figure: 2.1 from ISLR. $Y = \texttt{Sales}$ plotted against $\texttt{TV}$, $\texttt{Radio}$ and $\texttt{Newspaper}$ advertising budgets.

- We want a model, f, that describes $\texttt{Sales}$ as a function of the three advertising budgets.

$$\texttt{Sales} \approx f(\texttt{TV}, \texttt{Radio}, \texttt{Newspaper})$$

Notation and Terminology

- `Sales` is known as the response, or target, or outcome. It's the variable we wish to predict. We denote the response variable as Y.
- `TV` is a feature, or input, or predictor. We denote it by X_1
- Similarly, we denote X_2 = `Radio` and $X_3 =$ `Newspaper`
- We can put all the predictors into a single input vector

$$X = (X_1, X_2, X_3)$$

- Now we can write our model as

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

where ϵ captures measurement errors and other discrepancies between the response Y and the model f

What is $f(X)$ useful for?

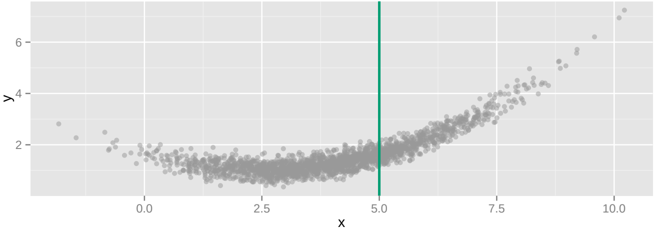
With a good model f, we can:

- Make predictions of Y at new points $X = x$.
- Understand which components of $X = (X_1, X_2, \ldots, X_p)$ are important for predicting Y.
    - We can look at which inputs are the most important in the model
    - E.g., If $Y = \texttt{Income}$ and X = (`Age`, `Industry`, `Favorite Color`, `Education`), we may find that $X_3 = \texttt{Favorite Color}$ doesn't help with predicting Y at all
- If f isn't too complex, we may be able to understand how each component X_j affects Y.[1]

[1] In this class, the statement "X_j affects Y" should *not* be interpreted as a causal claim.

What does it mean to 'predict Y'?

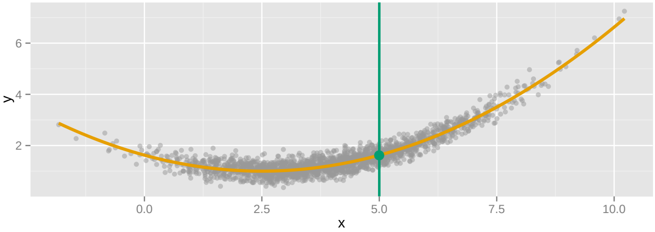
Here's some simulated data.



- Look at $X = 5$. There are many different Y values at $X = 5$.
- When we say *predict Y at* $X = 5$, we're really asking:

What is the expected value (average) of Y at $X = 5$?

The regression function



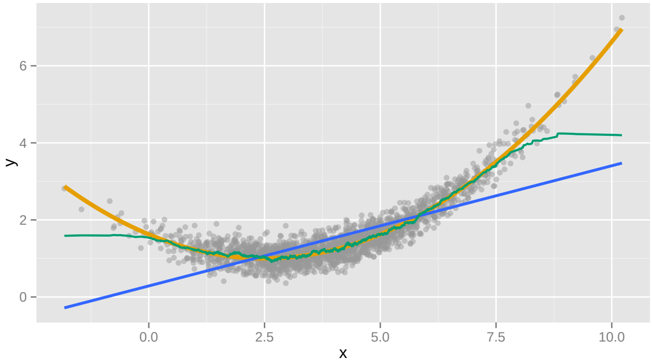
Definition: Regression function

Formally, the regression function is given by $\mathrm{E}(Y \mid X = x)$. This is the *expected value of* Y at $X = x$.

- The ideal or optimal predictor of Y based on X is thus

$$f(x) = \mathrm{E}(Y \mid X = x)$$

The prediction problem



regression function f linear regression $\hat{f}$ 50-nearest-neighbours $\hat{f}$

The prediction problem

We want to use the observed data to construct a predictor $\hat{f}(x)$ that is *a good estimate* of the regression function $f(x) = \mathrm{E}(Y \mid X = x)$.

Summary

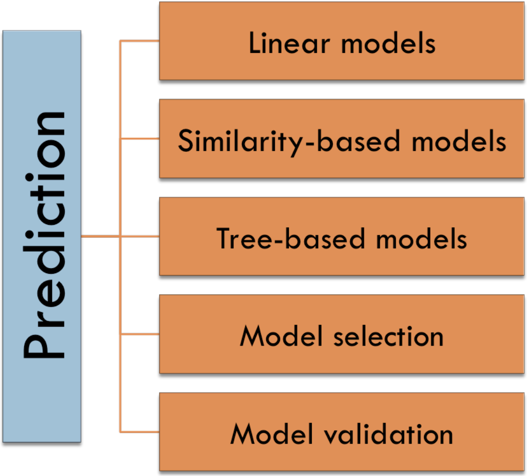
- The ideal predictor of a response Y given inputs $X = x$ is given by the regression function

$$f(x) = \mathrm{E}(Y \mid X = x)$$

- We *don't know* what f is, so the prediction task is to estimate the regression function from the available data.

- The various prediction methods we will talk about in this class are different ways of using data to construct estimators $\hat{f}$

Prediction topics



Prediction topics

- Prediction
  - Linear models
  - Similarity-based models
  - Tree-based models
  - Model selection
  - Model validation

Why are we learning about all these different methods?

Some of you might be thinking...

Prof C., can't you just teach us the **best** method?

Well…as it turns out…

Broad paraphrasing of Wolpert's No Free Lunch Theorem

Without any prior information about the modelling problem, there is no single model that will always do better than any other model. [a]

Alternatively: If we know nothing about the true regression function, all methods on average perform equally well (or poorly).

[a] To learn more, read this

Data mining in a No Free Lunch Theorem world

The reason we may prefer some methods over others is because we have found them to be good at capturing the types of structure that tend to arise in the problems we encounter.

- If the data you work with tends to have linear associations, you may be well-served by a linear model
- If you know that similar people like similar things, you may be well-served by a nearest-neighbours method

- Indeed, if we lived in a universe in which all relationships are linear, then linear regression would be all we'd ever really need

Linear models don't work for everything in our world, but they do work well in many cases. So today we're going to …

Via the Washington Post. wapo.st/1OjJ02P

Regression topics

- Linear regression from a prediction point of view
- Polynomial regression
- Step functions
- Next class: Splines
- Next class: Additive models

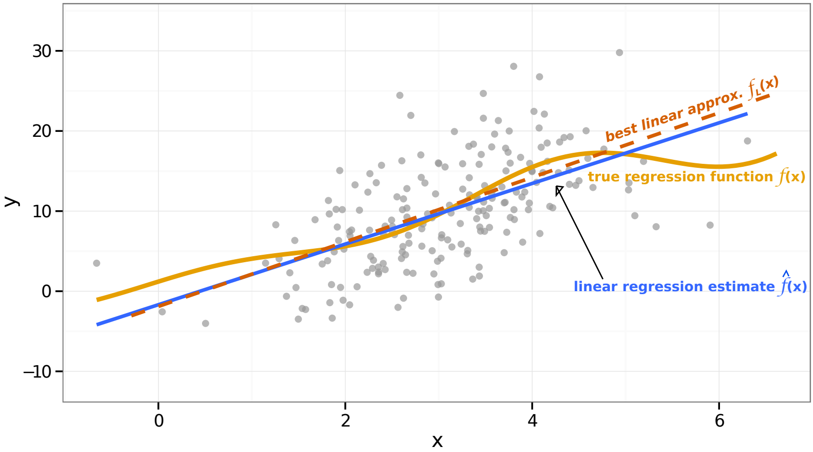
Linear regression refresher

- Linear regression is a *supervised learning approach* that models the dependence of Y on the covariates $X_1, X_2, \ldots, X_p$ as being linear:

$$
\begin{aligned}
Y &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \\
&= \underbrace{\beta_0 + \sum_{j=1}^{p} \beta_j X_j}_{f_L(X)} + \underbrace{\epsilon}_{\text{error}}
\end{aligned}
$$

- The true regression function $\mathrm{E}(Y \mid X = x)$ might not be linear (it almost never is)
- Linear regression aims to estimate $f_L(X)$: the best linear approximation to the true regression function

Best linear approximation



Linear regression

- Here's the linear regression model again:

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \epsilon$$

- The β_j, $j = 0, \ldots, p$ are called model coefficients or parameters
- Given estimates $\hat{\beta}_j$ for the model coefficients, we can predict the response at a value $x = (x_1, \ldots, x_p)$ via

$$\hat{y} = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_j$$

- The hat symbol denotes values estimated from the data

Estimation of the parameters by least squares

- Suppose that we have data (x_i, y_i), $i = 1, \ldots, n$

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \qquad X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}$$

- Linear regression estimates the parameters β_j by finding the parameter values that *minimize* the residual sum of squares (RSS):

$$\begin{aligned} \mathrm{RSS}(\hat{\beta}) &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} \left(y_i - \left[\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \beta_p x_{ip} \right] \right)^2 \end{aligned}$$

- The quantity $e_i = y_i - \hat{y}_i$ is called a residual

Least squares picture in 1-dimension

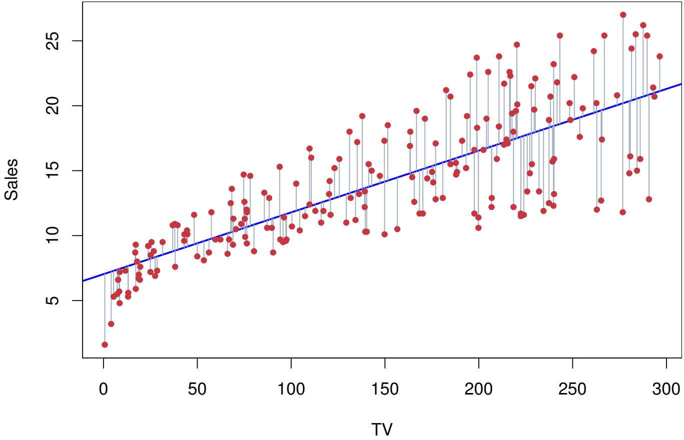


Figure: 3.1 from ISLR. Blue line shows least squares fit for the regression of `Sales` onto `TV`. Lines from observed points to the regression line illustrate the residuals. For any other choice of slope or intercept, the *sum of squared vertical distances* between that line and the observed data would be larger than that of the line shown here.

Least squares picture in 2-dimensions

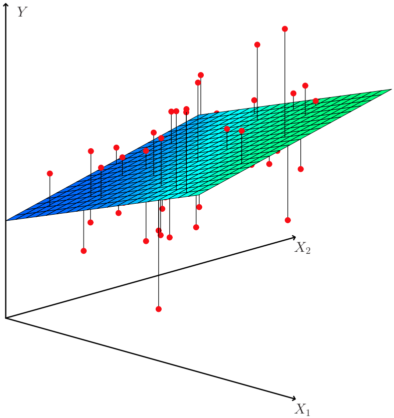
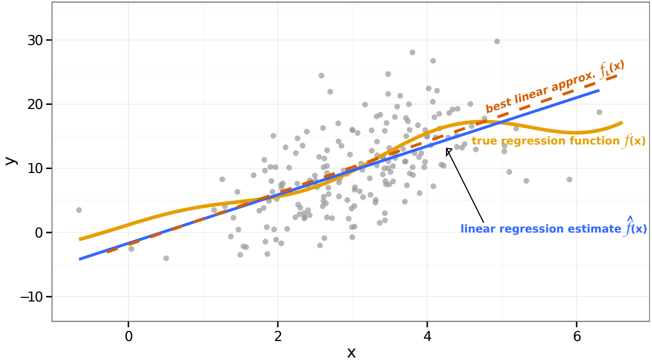


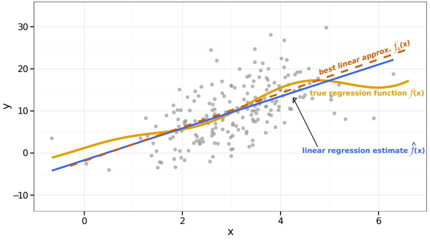
Figure: 3.4 from ISLR. The 2-dimensional place is the least squares fit of Y onto the predictors X_1 and X_2. If you tilt this plane in any way, you would get a larger *sum of squared vertical distances* between the plane and the observed data.

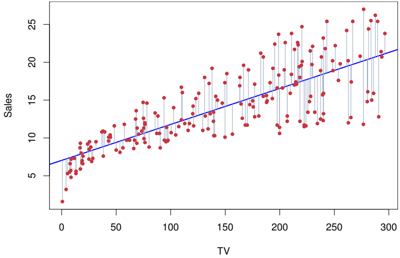
Summary



- Linear regression aims to predict the response Y by estimating the best linear predictor: the linear function that is closest to the true regression function f.
- The parameter estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ are obtained by minimizing the residual sum of squares

Summary





- Linear regression aims to predict the response Y by estimating the best linear predictor: the linear function that is closest to the true regression function f.
- The parameter estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ are obtained by minimizing the residual sum of squares

$$\mathrm{RSS}(\hat{\beta}) = \sum_{i=1}^{n} \left(y_i - \left[\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right] \right)^2$$

- Once we have our parameter estimates, we can predict y at a new value of $x = (x_1, \ldots, x_p)$ with

Linear regression is easily* interpretable

(*As long as the # of predictors is small)

- In the Advertising data, our model is

$$\texttt{sales} = \beta_0 + \beta_1 \times \texttt{TV} + \beta_2 \times \texttt{radio} + \beta_3 \times \texttt{newspaper} + \epsilon$$

- The coefficient β_1 tells us the expected change in `sales` per unit change of the `TV` budget, with all other predictors held fixed
- Using the `lm` function in **R**, we get:

	Coefficient	Std. Error	t-statistic	p-value
`Intercept`	2.939	0.3119	9.42	< 0.0001
`TV`	0.046	0.0014	32.81	< 0.0001
`radio`	0.189	0.0086	21.89	< 0.0001
`newspaper`	-0.001	0.0059	-0.18	0.8599

- So, holding the other budgets fixed, for every $1000 spent on `TV` advertising, `sales` on average increase by (1000×0.046) = 46 units sold [2]

[2] `sales` is recorded in 1000's of units sold

The perils of over-interpreting regression coefficients

- A regression coefficient β_j estimates the expected change in Y per unit change in X_j, assuming all other predictors are held fixed
- But predictors typically *change together*!
- Example: A firm might not be able to increase the `TV` ad budget without *reallocating* funds from the `newspaper` or `radio` budgets
- Example:[3] $Y =$ total amount of money in your pocket; $X_1 =$ # of coins; $X_2 =$ # pennies, nickels and dimes.
  - By itself, a regression of $Y \sim \beta_0 + \beta_2 X_2$ would have $\hat{\beta}_2 > 0$. But how about if we add X_1 to the model?

[3] *Data Analysis and Regression*, Mosteller and Tukey 1977

In the words of a famous statistician…

"Essentially, all models are wrong, but some are useful."

—George Box

- As an analyst, you can make your models more useful by
  1. Making sure you're solving useful problems
  2. Carefully interpreting your models in meaningful, practical terms
- So that just leaves one question…

How can we make our models less wrong?

Making linear regression great (again)

- Linear regression imposes two key restrictions on the model: We assume the relationship between the response Y and the predictors $X_1, \ldots, X_p$ is:
  1. Linear
  2. Additive

- The truth is almost never linear; but often the linearity and additivity assumptions are *good enough*

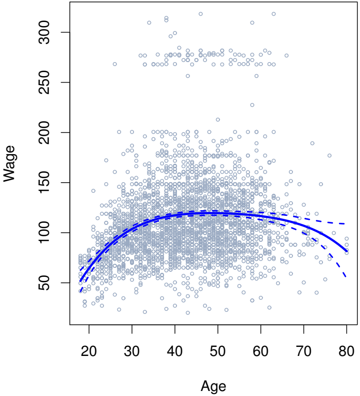
- When we think linearity might not hold, we can try…
  - Polynomials
  - Step functions
  - Splines (Next class)
  - Local regression
  - Generalized additive models (Next class)

- When we think the additivity assumption doesn't hold, we can incorporate interaction terms

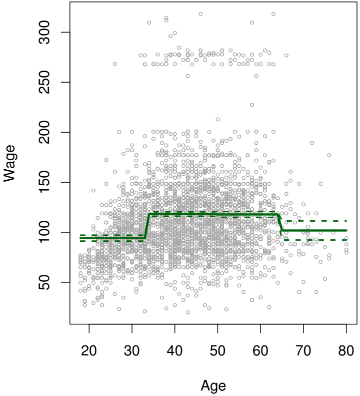
- These variants offer increased flexibility, while retaining much of the ease and interpretability of ordinary linear regression

Polynomial regression, Step functions

Polynomials and Step functions are simple forms of feature engineering



(a) Degree-4 polynomial



(b) Step function (cuts at 35, 65)

Polynomial regression

- Start with a variable X. E.g., $X = \texttt{Age}$
- Create new variables ("features")

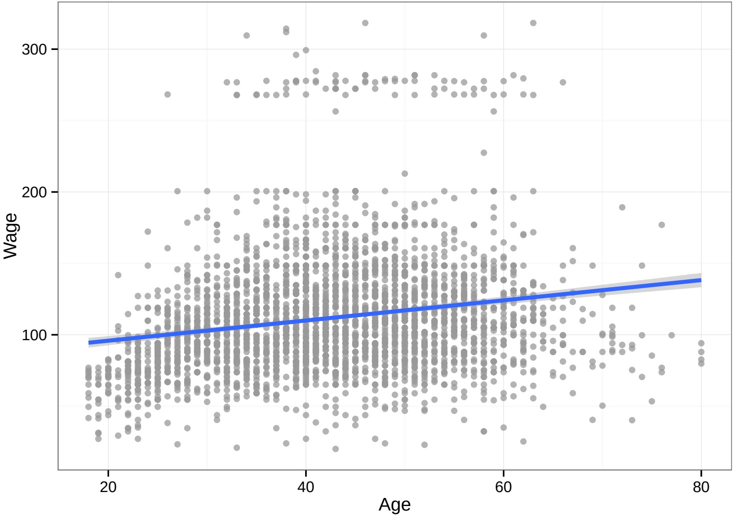
$$X_1 = X\ , \quad X_2 = X^2, \quad \ldots \quad , X_k = X^k$$

- Fit linear regression model with new variables x_1, x_2, …, x_k

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik} + \epsilon_i \\ &= \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_k x^k + \epsilon_i \end{aligned}$$

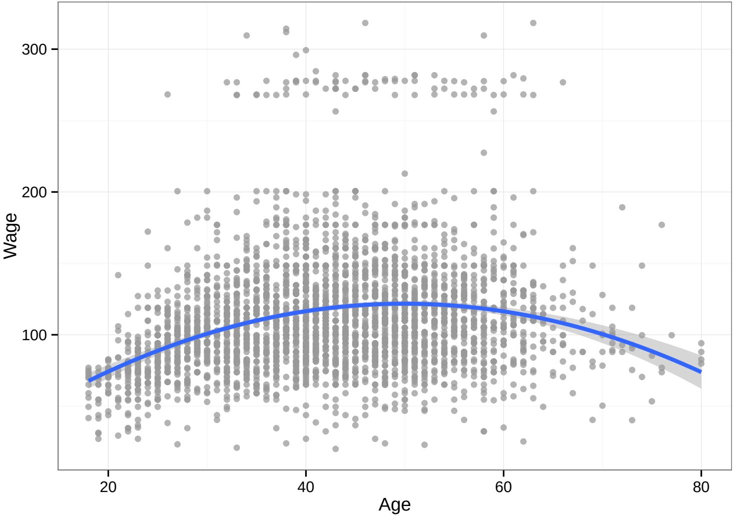
Coding tip: In **R** you can use the syntax `poly(x, k)` in your regression formula to fit a degree-`k` polynomial in the variable `x`.

Polynomial regression



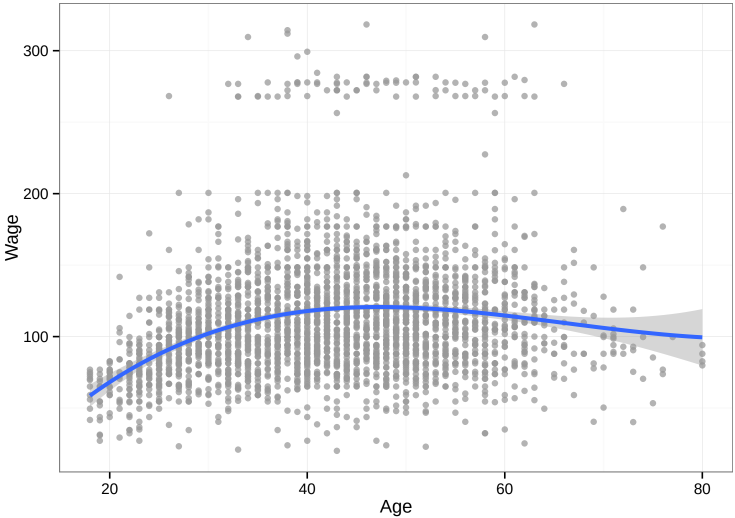

```
lm(wage ~ age, data = Wage)
```

Polynomial regression



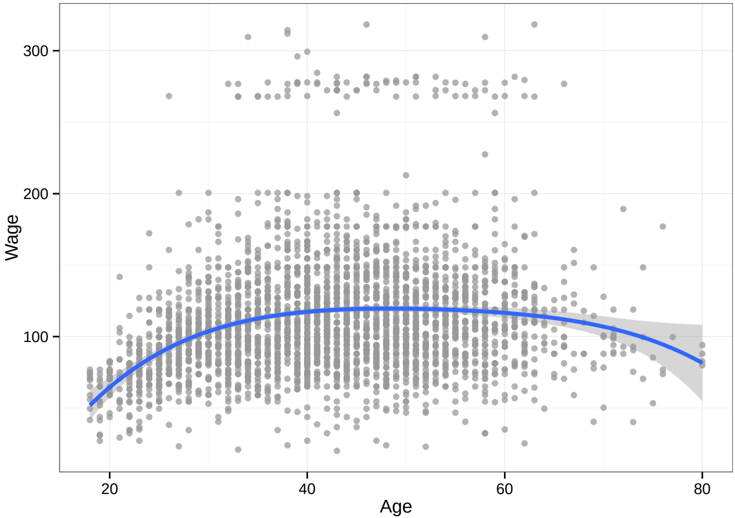

```
lm(wage ∼ poly(age, 2), data = Wage)
```

Polynomial regression



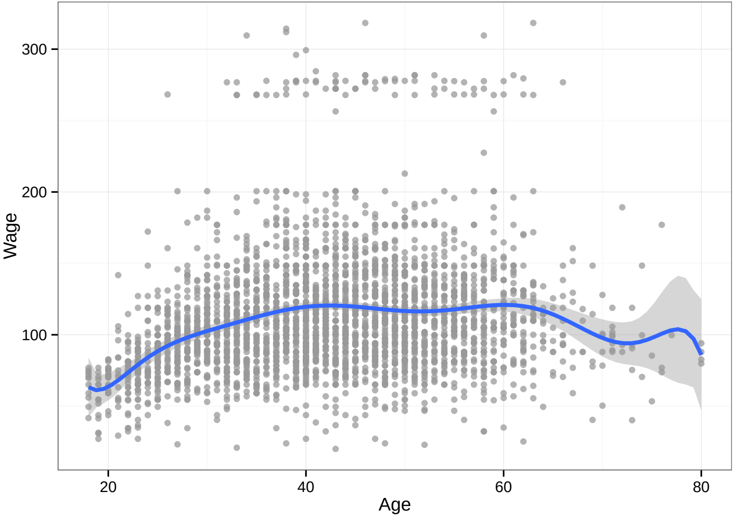

```
lm(wage ~ poly(age, 3), data = Wage)
```

Polynomial regression




```
lm(wage ~ poly(age, 4), data = Wage)
```

Polynomial regression




```
lm(wage ~ poly(age, 10), data = Wage)
```

Step functions

- Start with a variable X. E.g., $X = \texttt{Age}$
- Create new dummy indicator variables by *cutting* or *binning* X:
  $C_1 = I(X < t_1)$,
  $C_2 = I(t_1 \leq X < t_2)$, ...,
  $C_k = I(X > t_{k-1})$
- $I(\cdot)$ is called the indicator function
  - $I(\cdot) = 1$ if the condition holds, and 0 if it doesn't

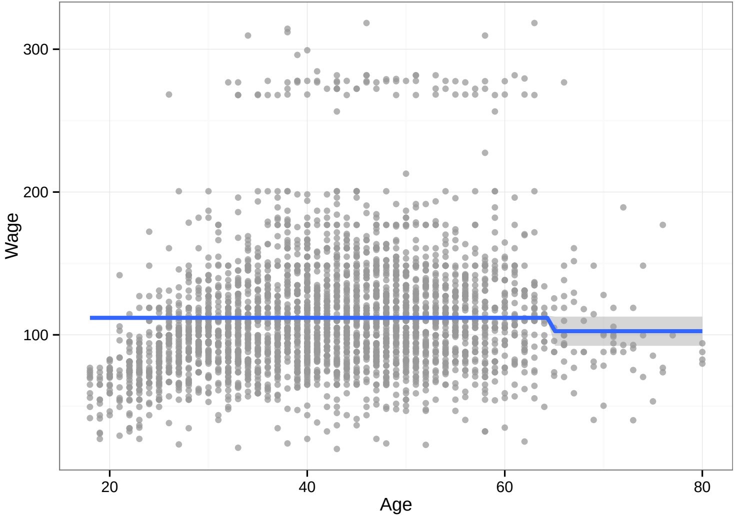
Step functions: Example

- $C_1 = I(\texttt{Age} < 35)$
- $C_2 = I(35 \leq \texttt{Age} < 65)$
- $C_3 = I(\texttt{Age} \geq 65)$

Age	C_1	C_2	C_3
18	1	0	0
24	1	0	0
45	0	1	0
67	0	0	1
54	0	1	0
⋮	⋮	⋮	⋮

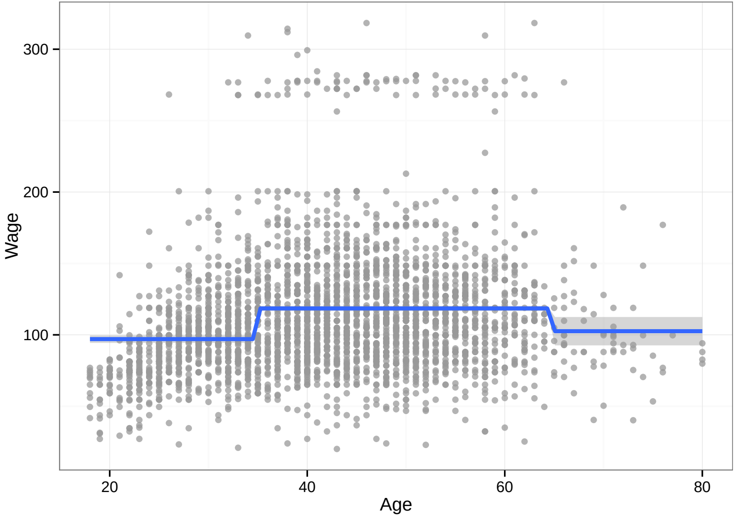
Coding tip: In **R** you can use the syntax `cut(x, breaks)` in your regression formula to fit a step function in the variable `x` with breakpoints given by the vector `breaks`.

Step functions



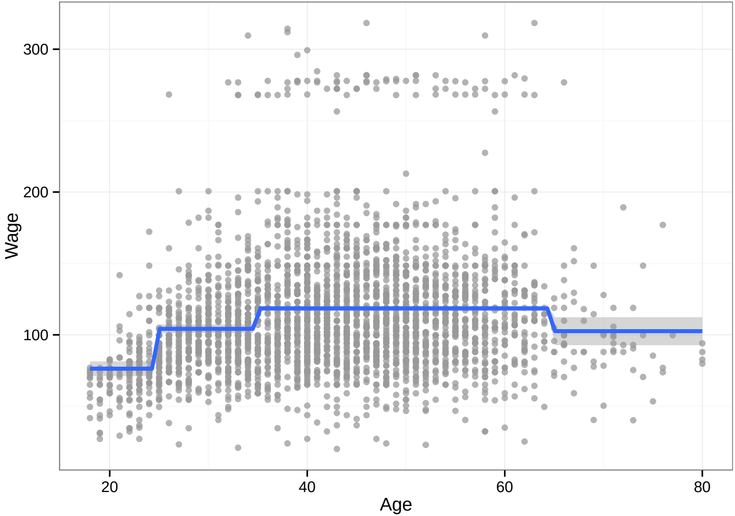

```
lm(wage ~ cut(age, breaks = c(-Inf, 65, Inf)), data = Wage)
```

Step functions



`lm(wage ~ cut(age, breaks = c(-Inf, 35, 65, Inf)), data = Wage)`

Step functions




```
lm(wage ~ cut(age, breaks = c(-Inf, 25, 35, 65, Inf)), data = Wage)
```

Acknowledgements

All of the lectures notes for this class feature content borrowed with or without modification from the following sources:

- 36-462/36-662 Lecture notes (Prof. Tibshirani, Prof. G'Sell, Prof. Shalizi)
- 95-791 Lecture notes (Prof. Dubrawski)
- *An Introduction to Statistical Learning, with applications in R* (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani