Week 1 Lectures

Part I: Introduction to data mining
Part II: Regression

Prof. Alexandra Chouldechova
95-791: Data Mining

Spring 2019
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.
Content Tagging, Text mining

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 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/](http://www-bcf.usc.edu/~gareth/ISL/)
- Supplementary **video lectures**, **slides** available here: [http://tinyurl.com/k7pq879](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

**TOP 10 CHALLENGEROCKET.COM RANKING**

<table>
<thead>
<tr>
<th>Language</th>
<th>Projected Earnings in 2017 (USD per year)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>130,000</td>
</tr>
<tr>
<td>Objective-C</td>
<td>112,000</td>
</tr>
<tr>
<td>Python</td>
<td>105,000</td>
</tr>
<tr>
<td>C++</td>
<td>102,000</td>
</tr>
<tr>
<td>Perl</td>
<td>100,000</td>
</tr>
<tr>
<td>C</td>
<td>100,000</td>
</tr>
<tr>
<td>R</td>
<td>99,000</td>
</tr>
<tr>
<td>JavaScript</td>
<td>92,000</td>
</tr>
<tr>
<td>C#</td>
<td>89,000</td>
</tr>
<tr>
<td>SQL</td>
<td>80,000</td>
</tr>
</tbody>
</table>

**R Usage**

- 76% of analytic professionals report using R
- 36% select R as their primary tool
Scope of this class

The Data Analytics Cycle

Predictive Analysis
- Detect, Explain, Understand
- Forecast, Diagnose, Scale

Descriptive Analysis
- Observe, Measure, Collect data

Operations, Management
- Optimize, Decide, Execute

Prescriptive Analysis
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., \(Y_i \in \{0, 1\}, Y_i \in \{\text{spam, email}\}, Y_i \in \{1, 2, 3, 4\}\) )

- **Descriptive Analytics (Unsupervised learning):**
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., \(Y_i \in \{0, 1\}\), \(Y_i \in \{\text{spam, email}\}\), \(Y_i \in \{1, 2, 3, 4\}\))

- **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** \( Y_i = \text{Sales} \) in 1000’s of units
- **Covariates/inputs:** Budgets for \( X_1 = \text{TV} \), \( X_2 = \text{Radio} \), and \( X_3 = \text{Newspaper} \) 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

\[ \text{Sales} \approx f(\text{TV}, \text{Radio}, \text{Newspaper}) \]

• We want to find a function \( f \) such that \( f(\text{TV}, \text{Radio}, \text{Newspaper}) \) is a good predictor of 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:
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

95-791

Predictive Analytics

Prediction

Classification

Descriptive Analytics
What is the prediction task?

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

- We want a model, \( f \), that describes Sales as a function of the three advertising budgets.

\[
\text{Sales} \approx f(\text{TV}, \text{Radio}, \text{Newspaper})
\]
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 $X_2 = \text{Radio}$ and $X_3 = \text{Newspaper}$.
- 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 $\epsilon$ 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 = $ Income and $X = (Age, Industry, Favorite Color, Education)$, we may find that $X_3 = $ 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 $E(Y \mid X = x)$. This is the expected value of $Y$ at $X = x$.

- The ideal or optimal predictor of $Y$ based on $X$ is thus $f(x) = E(Y \mid X = x)$.
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 $f(x) = \mathbb{E}(Y \mid X = x)$. 

**regression function** $f$  **linear regression** $\hat{f}$  **50-nearest-neighbours** $\hat{f}$
• The ideal predictor of a response $Y$ given inputs $X = x$ is given by the regression function

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

• 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

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

- 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. 

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

*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 …
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:

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

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

  $$= f_L(X) + \text{error}$$

- The **true** regression function $E(Y \mid X = x)$ 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
Here’s the linear regression model again:

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

The \( \beta_j, j = 0, \ldots, p \) 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 \((x_i, y_i), i = 1, \ldots, n\)

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

• Linear regression estimates the parameters \(\beta_j\) by finding the parameter values that minimize the residual sum of squares (RSS):

\[
\text{RSS}(\hat{\beta}) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2
\]

\[
= \sum_{i=1}^{n} \left(y_i - \left[\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}\right]\right)^2
\]

• The quantity \(e_i = y_i - \hat{y}_i\) is called a residual
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.
Figure: 3.4 from ISLR. The 2-dimensional plane 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

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

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

• 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

\[ \text{sales} = \beta_0 + \beta_1 \times \text{TV} + \beta_2 \times \text{radio} + \beta_3 \times \text{newspaper} + \epsilon \]

• The coefficient \( \beta_1 \) tells us the expected change in sales per unit change of the TV budget, with all other predictors held fixed

• Using the \texttt{lm} function in \texttt{R}, we get:

\[
\begin{array}{|c|c|c|c|c|}
\hline
\text{Coefficient} & \text{Std. Error} & \text{t-statistic} & \text{p-value} \\
\hline
\text{Intercept} & 2.939 & 0.3119 & 9.42 & < 0.0001 \\
\text{TV} & 0.046 & 0.0014 & 32.81 & < 0.0001 \\
\text{radio} & 0.189 & 0.0086 & 21.89 & < 0.0001 \\
\text{newspaper} & -0.001 & 0.0059 & -0.18 & 0.8599 \\
\hline
\end{array}
\]

• So, holding the other budgets fixed, for every $1000 spent on TV advertising, sales on average increase by \((1000 \times 0.046) = 46 \text{ units sold}^2\)

\(^2\text{sales is recorded in 1000’s of units sold}\)
The perils of over-interpreting regression coefficients

- A regression coefficient $\beta_j$ 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: $Y =$ 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., $X = \text{Age}$
- Create new variables (“features”)
  
  $X_1 = X$, $X_2 = X^2$, ..., $X_k = X^k$

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

  
  $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik} + \epsilon_i$

  $= \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_k x^k + \epsilon_i$

**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

\[
\text{lm(wage} \sim \text{age, data = Wage)}
\]
Polynomial regression

\[ \text{lm}(\text{wage} \sim \text{poly}(\text{age}, 2), \text{data} = \text{Wage}) \]
Polynomial regression

\[ \text{lm}(wage \sim \text{poly}(age, 3), \text{data} = \text{Wage}) \]
Polynomial regression

\texttt{lm(wage \sim \text{poly}(age, 4), data = Wage)}
Polynomial regression

\[
\text{lm(wage} \sim \text{poly(age, 10), data = Wage)}
\]
Step functions

• Start with a variable $X$. E.g., $X = \text{Age}$

• 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

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

<table>
<thead>
<tr>
<th>Age</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>67</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>54</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

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