CSE 4/587
Data Intensive Computing

Dr. Eric Mikida
epmikida@buffalo.edu
208 Capen Hall

Day 04
Cleaning, Algorithms, and Models

Announcements and Feedback

- Read chapter 3 in Doing Data Science, work through some examples

Recap from Last Class

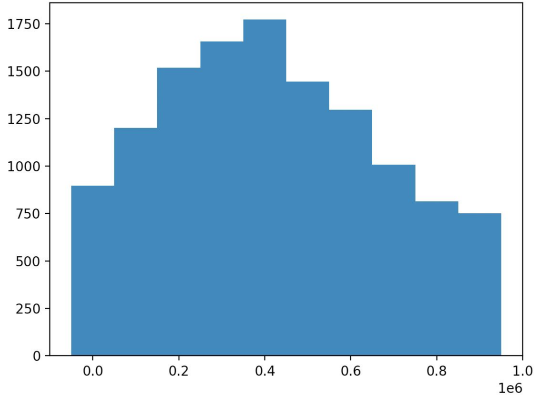
- Exploratory Data Analysis (EDA)
  - Get intuition about the nature of your data
  - Gather some basic stats/visualizations: min, max, mean, histograms, etc
  - Can be used to form some initial hypotheses
- Related to data cleaning, and feature extraction
  - We'll explore these two a bit more today
- Followed by more intensive modeling
  - We'll introduce a few modeling algorithms today

Data Cleaning and Munging

- Real-world data is almost always going to be *dirty*
  - Data will be missing/incomplete
  - Entries may contain errors
  - Entries may not be in the proper format
- Initial cleaning of the data will make the rest of the process smoother
  - Issues like formatting can often be dealt with immediately
  - Finding errors in the data may require EDA
  - EDA may reveal further cleaning that is required

Data Cleaning and Munging

- Examples (Ch 2 DDS, Ch 10 DSfS)
    - Clean up formatting for numbers
    - Remove nonsensical data (ie: sale prices of $0)
    - Check for outliers
    - Extract columns we want




```
def parse_num(f, s):
  return f(s.replace("$","").replace(",",""))

with open("rollingsales_brooklyn.csv", "r") as f:
  reader = csv.DictReader(f)
  for line in reader:
    data.append([
      parse_num(int,line["YEAR BUILT"]),
      parse_num(float,line["LAND SQUARE FEET"]),
      parse_num(float,line["GROSS SQUARE FEET"]),
      parse_num(float,line["SALE PRICE"])
    ])

plot_hist([d[3] for d in data if 0 < d[3] < 1000000], 100000)
```

Intro to Modeling Algorithms

- At this point, we have clean data, we have some intuition about it, and we've extracted just the parts of the data we are interested in
- Now, we can move to modeling to start getting useful information out of our data
- Two different types of algorithms/models
  - Optimization algorithms for parameter estimation
  - Machine learning algorithms

Optimization Algorithms

- These algorithms attempt to determine the parameters of the process from which the data is generated
- Once we have the parameters, we can use the resulting functions to predict new outcomes
- These algorithms also attempt to quantify the uncertainty; they attempt to give a measure of how good the prediction is
- Examples: Least squares, newton's methods, stochastic gradient descent

Machine Learning Algorithms

- These algorithms attempt to predict, classify, and cluster data
- Don't often make any claims about the degree of uncertainty
- Basis of AI

"Models" vs "Algorithms"?

- Distinction between the two is fuzzy at best
- Models come from the math side (statistics)...sort of
  - Equations which attempt to model the actual process at hand
  - Come with some measure of uncertainty
- Algorithms come from the computer science side (ML)...sort of
  - Set of steps required to achieve some result
  - Not designed (generally) to capture the underlying process, just to predict the outcome with the most accuracy

Linear Regression

- Very simple conceptually
- Expresses the relationship between two (or more) variables/attributes
- Assume a linear relationship between an outcome variable (also called dependent variable, response variable or label) and the predictor variable(s) (aka independent variables, features, explanatory variables)
- What if the relationship isn't linear?
    - Linear is a good starting point…
    - …but we can also look at other relationships after we get the basics down

Linear Regression

- Specifically, we assume the underlying data is related in the real world by a function of the form: $y = f(x) = \beta_0 + \beta_1 x$

Linear Regression

- Specifically, we assume the underlying data is related in the real world by a function of the form: $y = f(x) = \beta_0 + \beta_1 x$

y represents the
outcome we are trying
to predict

Linear Regression

- Specifically, we assume the underlying data is related in the real world by a function of the form: $y = f(x) = \beta_0 + \beta_1 x$

y represents the outcome we are trying to predict

x is our independent variable

Linear Regression

- Specifically, we assume the underlying data is related in the real world by a function of the form: $y = f(x) = B_0 + B_1x$

y represents the outcome we are trying to predict

β_0 and β_1 are the parameters we are trying to solve for

x is our independent variable

A few examples

Subscriber Revenue

Take the following table of monthly revenue and subscriber count:

Subscribers (x)	Revenue (y)
5	125
10	250
15	375
20	500

A few examples

Subscriber Revenue

Take the following table of monthly revenue and subscriber count:

Subscribers (x)	Revenue (y)
5	125
10	250
15	375
20	500

In this case it's clear that $y=25x$.

Notice that in this case, we actually know the truth of the model. The website charges $25 for a subscription.

The model is attempting to capture that.

A few examples

Friends vs Time Spent

Now take the following table as a more complex example:

7	276
3	43
4	83
6	136
10	417

A few examples

Friends vs Time Spent

Now take the following table:

New Friends (x)	Time Spent (y)
7	276
3	43
4	83
6	136
10	417

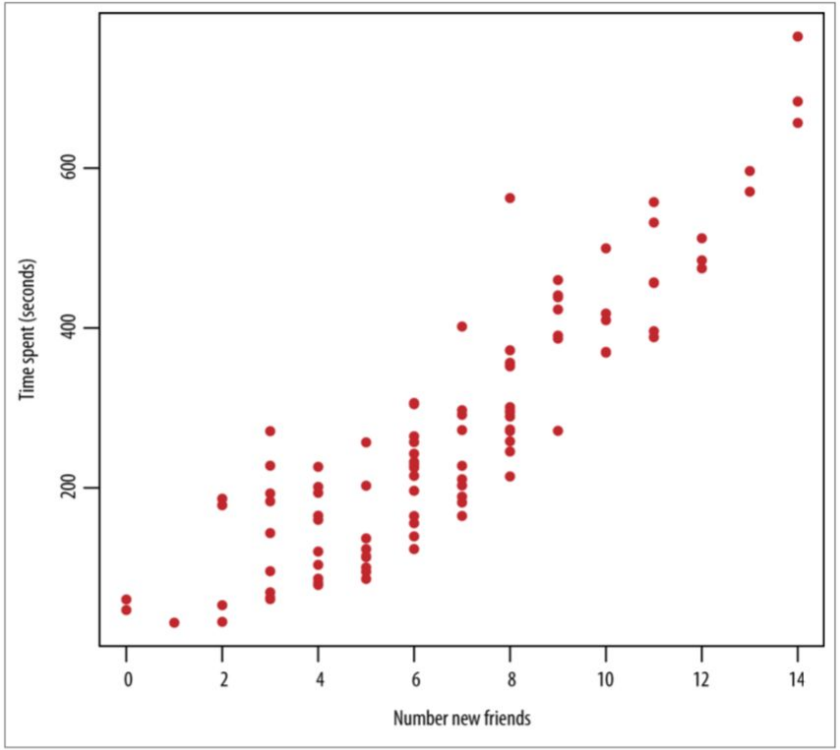
In this case, the data represents the amount of time a user spends on a social media site, compared to the number of new friends they've added this week.

What does our intuition say?

What does the data look like?

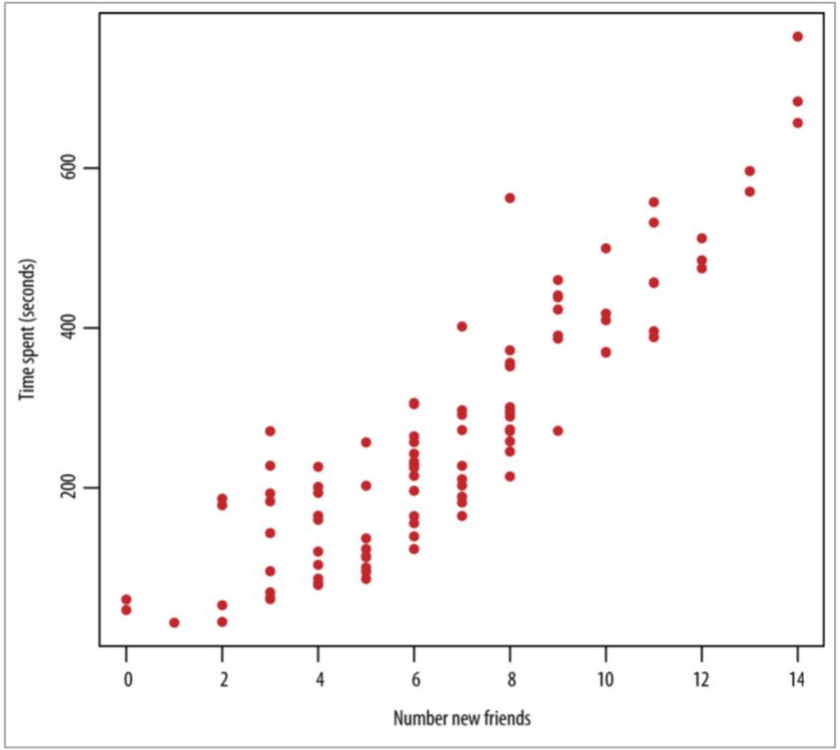
A few examples

- The right shows a plot of the dataset where the table came from
- We do see a generally linear looking relationship
- This time the model isn't deterministic…but can we estimate it?



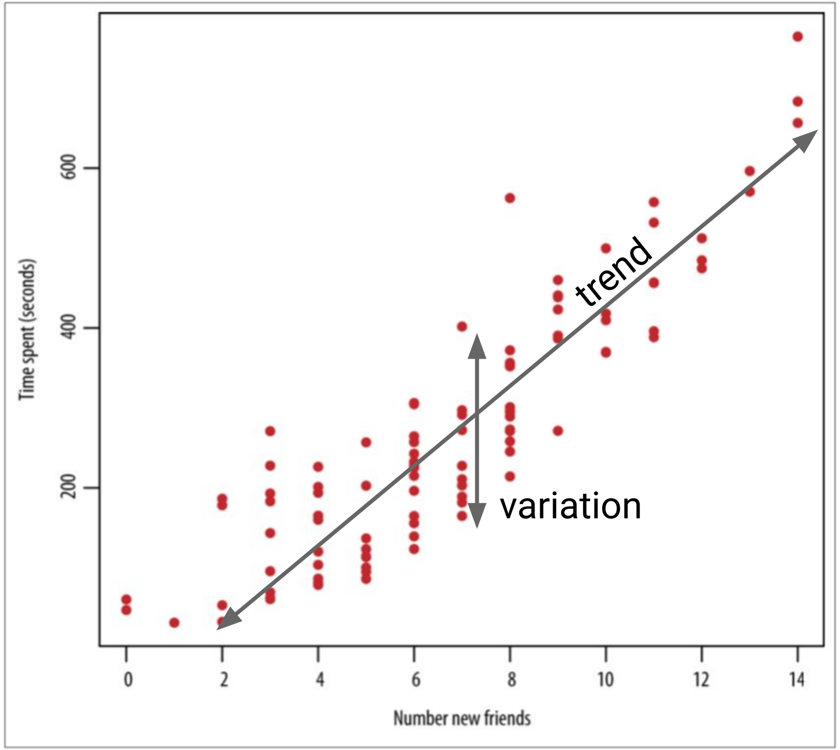
A few examples

- We want to capture 2 factors: trend and variation
- Assume a linear relationship ($y=\beta_0 + \beta_1 x$)
- Now we must *"fit"* the model - use an algorithm to find the best values of β_0 and β_1



A few examples

- We want to capture 2 factors: trend and variation
- Assume a linear relationship ($y=\beta_0 + \beta_1 x$)
- Now we must *"fit"* the model - use an algorithm to find the best values of β_0 and β_1

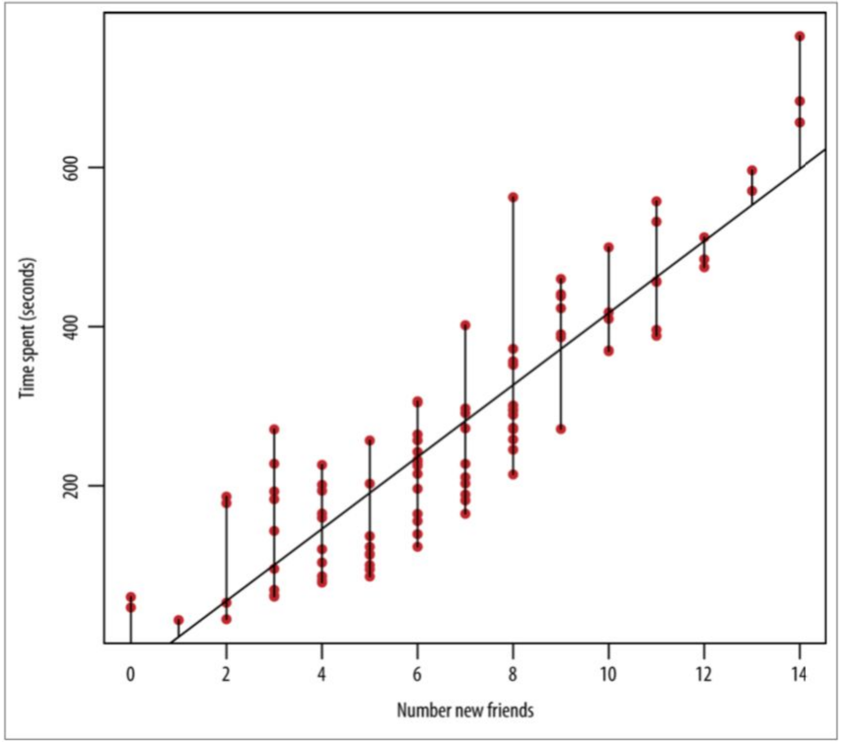


Fitting a Model

- Find the values of β_0 and β_1 that yield the "best" line
- What do we mean by "best"?
  - For now, the line that is on average closest to all the points
  - Closeness measured as vertical distance squared
- Therefore, we want the function that minimizes the sum of the squares for all points
  - This is called, unsurprisingly, *least squares estimation*

Fitting Our Example

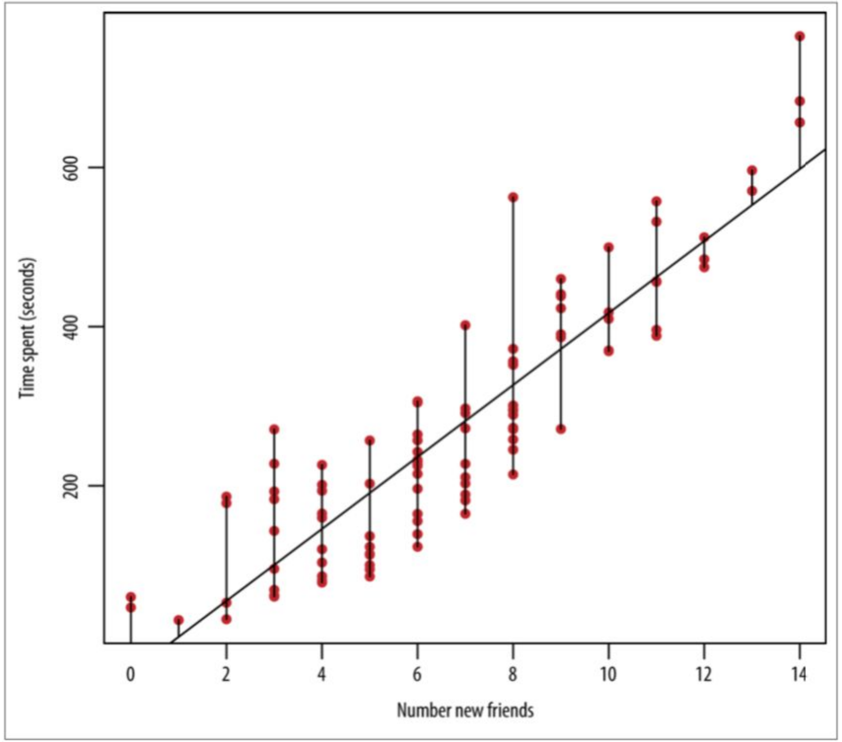
- Running the data through a solver yields β_0 = -32.08 and β_1= 45.92
- How confident are we in this model?
- If we have a new user, with 5 new friends, can we predict how much time they'll spend?



Fitting Our Example

- Running the data through a solver yields β_0 = -32.08 and β_1= 45.92
- How confident are we in this model?
- If we have a new user, with 5 new friends, can we predict how much time they'll spend?

...This, afterall, is the whole goal for modeling in the first place, right?

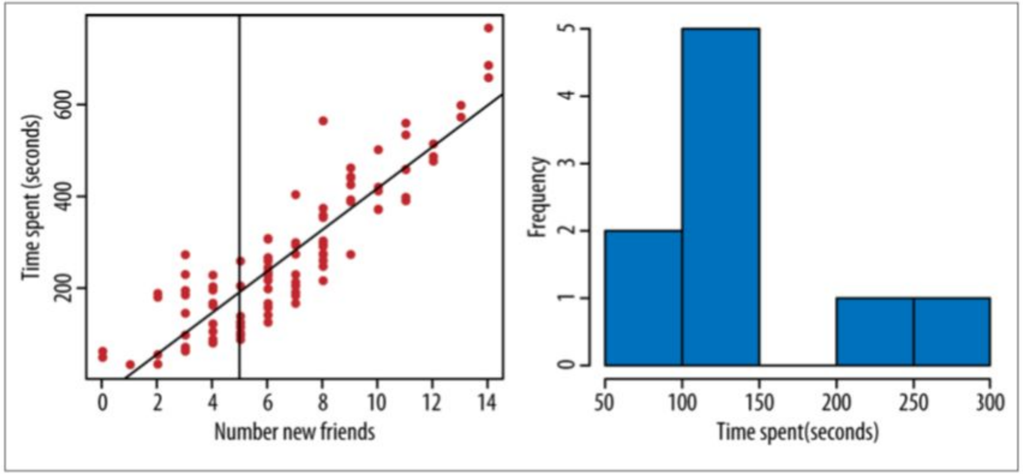


Next Steps...

- We have an initial model, how can we build on it?
    - Evaluate our model and add error terms
    - Add in more predictors
    - Transform the predictors

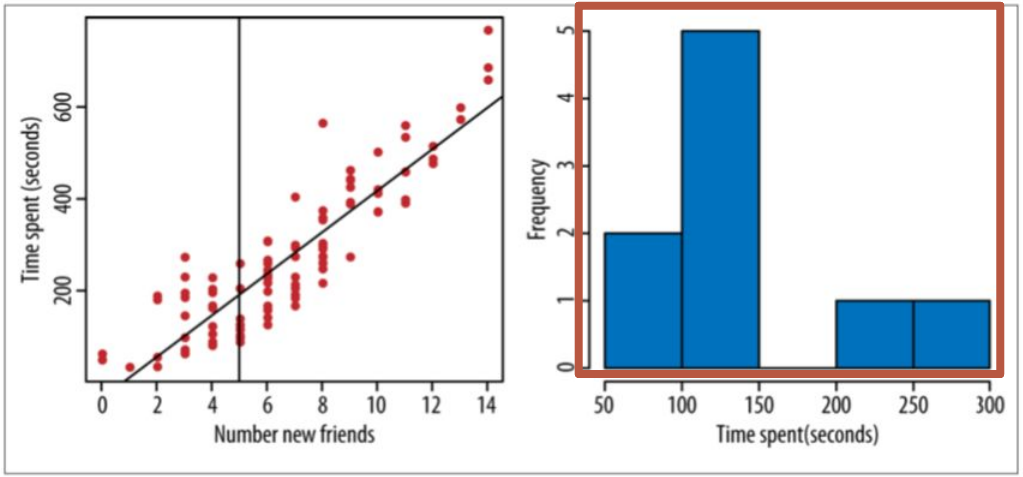
Capturing Variability

- With our model so far, predictions are *deterministic*
  - We claim that for a given x, the outcome will be y
  - However, our data has some amount of variability



Capturing Variability

- With our model so far, predictions are *deterministic*
  - We claim that for a given x, the outcome will be y
  - However, our data has some amount of variability



How do we capture this variability?

Capturing Variability

- Add in an error term, ϵ: $y = \beta_0 + \beta_1 x + \epsilon$
    - Referred to as *noise*
    - Represents relationships you have not accounted for
    - This term captures the difference between our observations, and the *true* regression line

Capturing Variability

- Add in an error term, ϵ: $y = \beta_0 + \beta_1 x + \epsilon$
  - Referred to as *noise*
  - Represents relationships you have not accounted for
  - This term captures the difference between our observations, and the *true* regression line

Remember, our data is just a trace of the real world. It is incomplete. It has uncertainty. We can only estimate the true regression line. Noise attempts to capture this fact.

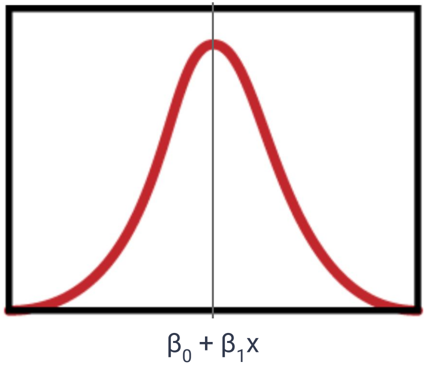
Finding Noise

- A common first assumption is that noise follows a normal distribution
    - $\epsilon \sim N(0, \sigma^2)$
    - It then follows that $p(y|x) \sim N(\beta_0 + \beta_1 x, \sigma^2)$
    - We have already found β_0 and β_1
    - σ^2 is the mean squared error (roughly the sum of all of the observed error squared, divided by n-2)

Finding Noise

- A common first assumption is that noise follows a normal distribution
  - $\epsilon \sim N(0, \sigma^2)$
  - It then follows that $p(y|x) \sim N(\beta_0 + \beta_1 x, \sigma^2)$
  - We have already found β_0 and β_1
  - σ^2 is the mean squared error (roughly the sum of all of the observed error squared, divided by n-2)

Our prediction now becomes: Given $x = 5$, we predict y is a random variable with the distribution shown to the right.



Evaluating Our Model

- How can we be certain our model is good?
- Many solvers will compute a few heuristics to help
    - R^2 captures the amount of the variance explained by our model
        - High R^2 means we've captured most of the variance
    - p-values captures the likelihood that our coefficients are "unimportant"
        - Low p-values means our coefficients are likely significant
- We can also cross validate ourselves!
    - Divide the data into training data and test data.
    - Fit the model on the training data to find β and ϵ
    - Calculate mean squared error on the test data and see if it's consistent

Extending Our Model

- Add more predictors…
  - $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \epsilon$
  - Fit using the package of your choice
  - May even have interaction between predictors
- Transformation on predictors
  - Why did we assume linear…what about $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots$
  - We can still use linear regression:
    - assume $z = x^2$
    - Now do a linear regression based on z