ECE 20875

Python for Data Science

Chris Brinton and David Inouye

(Adapted from material developed by
Prof. Milind Kulkarni and Prof. Chris Brinton)

MWF, 12:30pm-1:20pm

Section 1: WALC 1055
Section II: FRNY G124

some data analysis examples

data analysis in “practice”

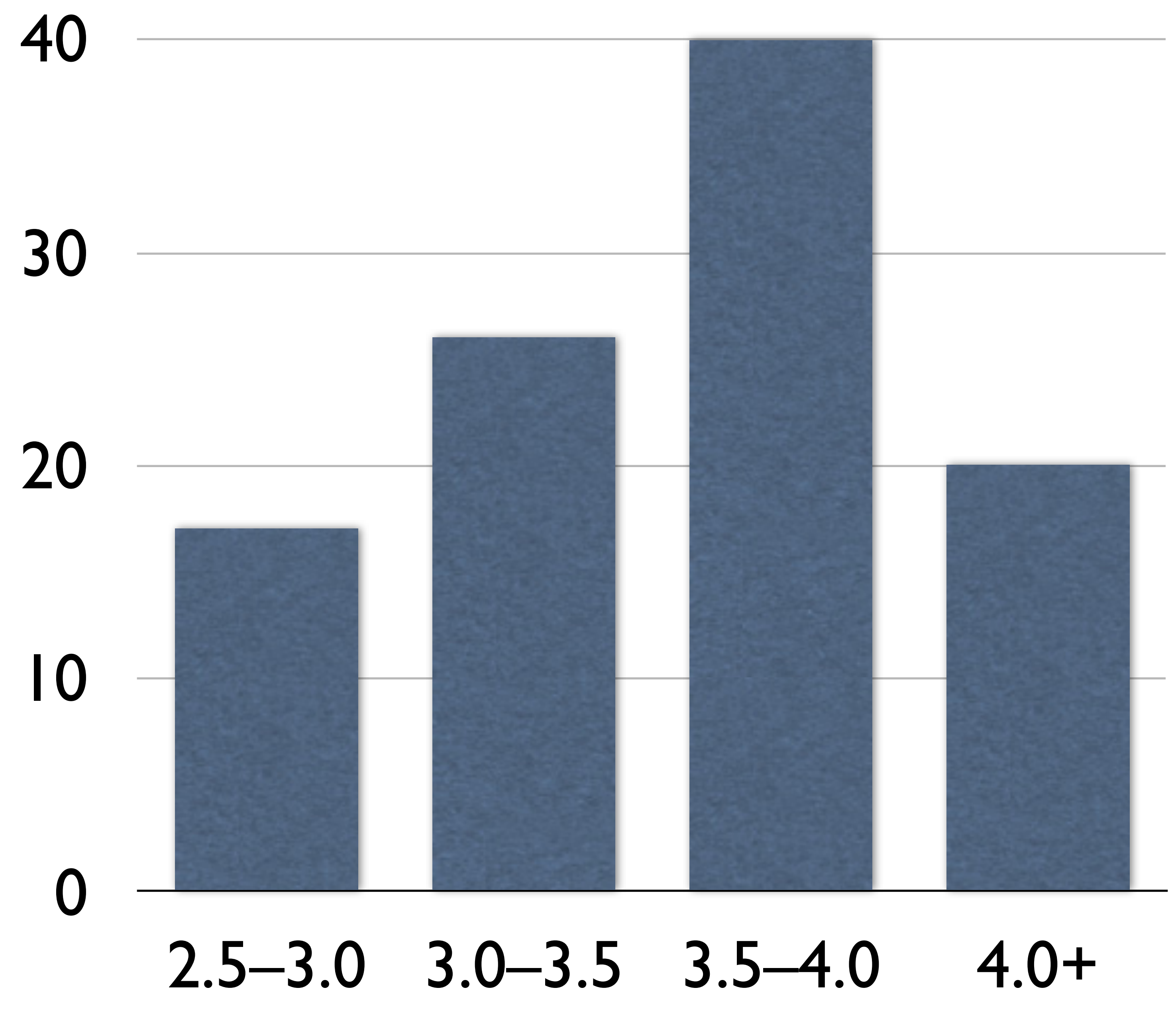
- Lets say we have a data set of applicants to Purdue

Name	High school GPA	SAT Math	SAT R/W	Residence
Jane Doe	4.7	760	700	Indiana
Purdue Pete	3.5	680	620	Indiana
B. O. ller	3.0	800	650	Michigan
Engy Neer	4.2	750	590	North Carolina
Mark Faller	3.8	780	550	New Jersey
…	…	…	…	…

- What might we want to learn about them?

descriptive statistics

- Which students come from which states?
- What is the distribution of GPAs? SAT scores?
  - GPAs may need to be *normalized* to a consistent range across all schools
- Can build *histograms*, e.g., for the GPAs
  - But how do we know how big to make the buckets?

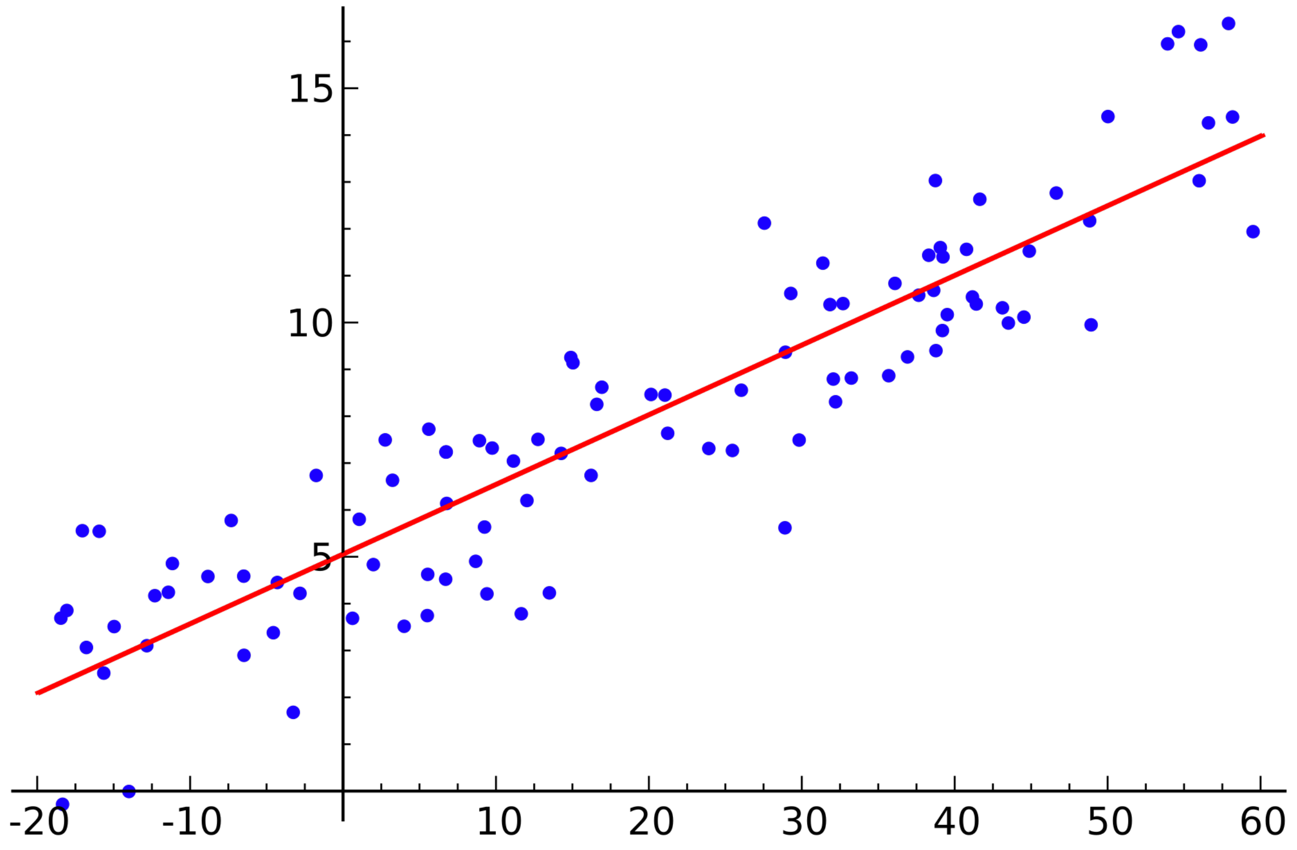


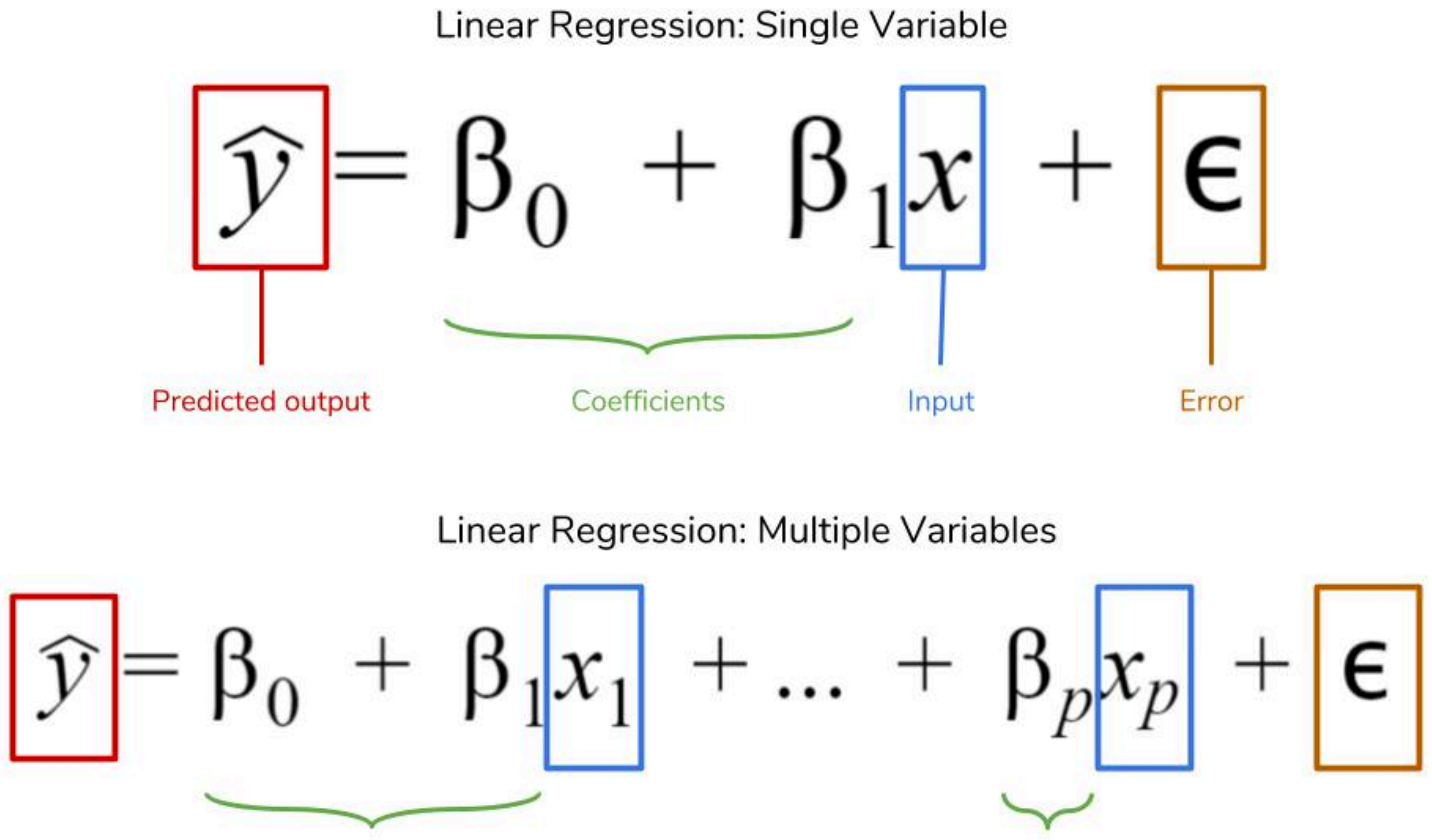
reasoning about data

- How do Purdue applicants compare to the national average?
  - *Mean* GPA of applicants: 3.6
- Is this high or low?
  - Can *sample* GPA of all high school students
- Suppose we collect 1000 GPAs and find a mean of 3.4
  - Does this mean Purdue students have a higher GPA on average?
- Need more information! In particular …
  - Was the sampling method we used *unbiased*?
  - What is the *variance* of the sample collected (i.e., the spread of GPAs)?
  - What *confidence interval* can be built for the population mean (i.e., what is the likely range of the true mean GPA)?

making predictions

- Can we predict how successful a particular applicant might be at Purdue?
  - How do we define success? GPA?
- Idea: Look at the application statistics of the *current seniors* and see if there is a relationship between these statistics and their current GPA
- One way to find a relationship is using *linear regression*
  - Might tell you something like: "a Purdue student's GPA can be predicted mostly by their high school GPA, with their SAT score having a lighter influence"
- Many other prediction algorithms exist too





Linear Regression: Single Variable

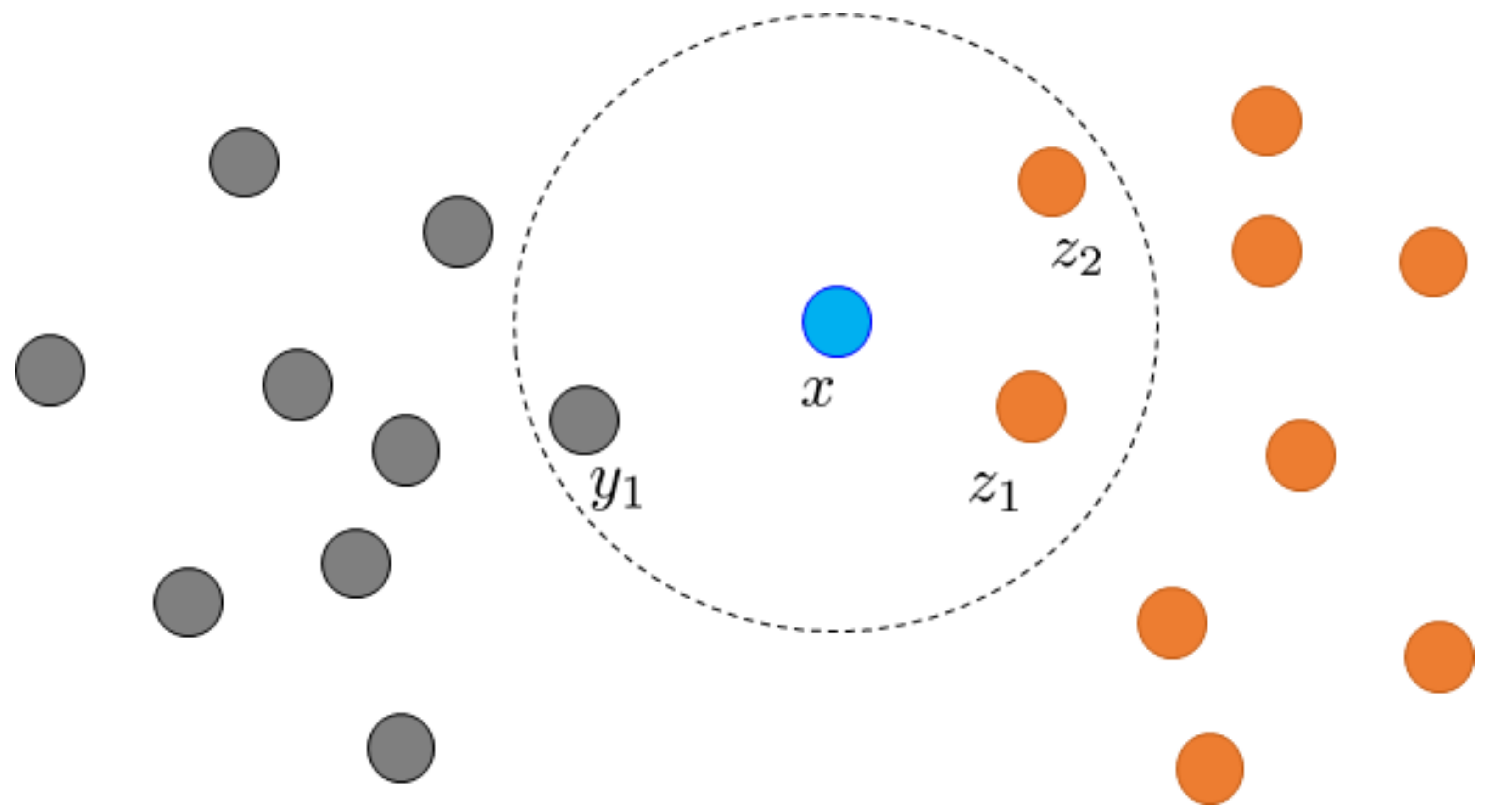
$$\underset{\text{Predicted output}}{\boxed{\hat{y}}} = \underbrace{\beta_0 + \beta_1}_{\text{Coefficients}} \underset{\text{Input}}{\boxed{x}} + \underset{\text{Error}}{\boxed{\epsilon}}$$

Linear Regression: Multiple Variables

$$\boxed{\hat{y}} = \underbrace{\beta_0 + \beta_1}_{} \boxed{x_1} + \dots + \underbrace{\beta_p}_{} \boxed{x_p} + \boxed{\epsilon}$$

classification

- Can we make admissions decisions quicker through automation?
- Idea: Compare each applicant's statistics to past applicants that were admitted, and to those that were rejected
- Train a *classifier* to analyze these past applicants and maximize the ability to predict whether a student would be accepted or not
  - For example, a *k-nearest neighbor* classifier would assess whether a given applicant is more similar to the pool of admitted applicants or to the rejected applicants
  - Why might we run into trouble here?



clustering

- What if we want to identify groups of students beyond "admitted" vs. "rejected"?
- Idea: See if students cluster together according to some measure of *distance*
  - Some students look more like "nearby" students than students that are "far away"
- Important question: What *features* of students should be considered for the clustering?
  - E.g., maybe don't consider something like hair color!
- With *k-means clustering*, *k* groups of students would be extracted based on "closeness"