



# Preliminaries



Data to decisions

# The goal

---

- This class describes the process whereby data are used to inform business decisions
- In a nutshell, it goes like this:
  1. State a precise question/problem
  2. Get appropriate data
  3. Validate the data (e.g. is sample representative of target population?)
  4. Select a model
  5. Estimate
  6. Answer question/optimize/make decision



# The tools

---

- Data processing (excel for small problems, e.g.)
- Probability theory
- Statistics
- And, finally, the more ad-hoc approaches people use in practice, such as classification



---

# Example I: Quality Control



# Calibrating a machine

---

- A company claims that a plastic injection press is properly calibrated
- Properly calibrated means  $\leq 1\%$  defect rate
- A 1,000-unit test-run produces 16 defective units
- Bad calibration or sample uncertainty?



# Hypothesis testing

---

- *Null Hypothesis*

$H_0$ : True defect rate is  $\pi \leq 1\%$

- Should  $H_0$  be *rejected* given the outcome of the test run?

- Does the evidence favor the *alternative hypothesis*

$H_a$ : True defect rate is  $> 1\%$  ?



# The idea

---

- Suppose I flip a supposedly fair coin 20 times
- If I did this 20-flip experiment over and over, I'd expect to see around 10 heads on average
- If I saw 12 heads in a particular sample, say, I would accept it as compatible with sample uncertainty
- But if I flip 20 heads in a row, I should probably reject the hypothesis that the coin is fair
- Because this should only happen in 0.0001% of the trials



# Back to quality control

---

- Assume  $\pi = 1\%$
- Then the sample defect rate  $\hat{\pi}$  is roughly normally distributed with *mean*  $\pi$  and *standard deviation*

$$\sqrt{\frac{\pi(1-\pi)}{n}}$$

where  $n = 1,000$  is the sample size

- This follows from the *Central Limit Theorem (CLT)*
- 





# Normal distribution facts

---

- With 95% probability ex-ante, a draw from a normal distribution is:
  1. within 1.96 standard deviations of the mean;
  2. no higher than mean + 1.645 standard deviations
  3. no lower than mean - 1.645 standard deviations
- Those sorts of facts underlie most of the tests we're going to learn about



# Critical test value

---

- When normality holds ( $\approx$  sample is large enough), sample mean ( $\hat{\pi}$ ) should be lower than population mean plus 1.645 times  $\sqrt{\frac{\pi(1-\pi)}{n}}$  in 95% of samples
- Here, this *critical value* is  $1\% + 1.645 \times 0.315\% \approx 1.52\%$
- Less than 5% chance of getting a sample with more than 15 defects if  $H_0$  is right
- So, “in all likelihood”,  $H_0$  is wrong



# Computing p-values

---

- Now  $\sqrt{\frac{\pi(1-\pi)}{n}} = \sqrt{\frac{1\%(99\%)}{1,000}} \approx 0.315\%$
- The probability of observing a 1.6% default rate or higher is, in excel-speak:  
 $1 - \text{normdist}(1.6\%, 1\%, 0.315\%, \text{TRUE}) \approx 2.83\%$
- That number is called the *p-value*
- Unlikely that sample uncertainty can explain away the high defect rate observed during the test run



# Classical estimation

---

- $\hat{\pi}$ , the sample mean rejection rate, is an *estimate* of  $\pi$
- $\sqrt{\frac{\pi(1-\pi)}{n}}$  is the *standard error of the estimate* (under the null), its precision so to speak
- When we don't know  $\pi$ , the standard error can itself be estimated as

$$\sqrt{\frac{\hat{\pi}(1 - \hat{\pi})}{n}}$$



# Confidence interval

---

- For  $n$  large enough, in roughly 95% of samples, the interval

$$\left[ \hat{\pi} - 1.96 \sqrt{\frac{\hat{\pi}(1 - \hat{\pi})}{n}}, \hat{\pi} + 1.96 \sqrt{\frac{\hat{\pi}(1 - \hat{\pi})}{n}} \right]$$

contains the true defect rate  $\pi$

- This is called a 95% *confidence interval* for the defect rate
- 



# Standard deviations vs standard errors

---

- Standard deviations are a measure of the variability of a random object
- Sample statistics/estimates are random objects (why?)
- So they have a standard deviation, which is called a standard error
- All standard errors are standard deviations
- But not all standard deviations are standard errors



---

# Example 2: Forecasting returns



# Asset pricing

---

- What return should I expect from IBM given how I expect the overall market (the S&P500) to perform?
- Data: historical returns for IBM and S&P500
- Would any other data be useful? Classic finance (CAPM) says no
- In fact, classic finance says that the “best” model for our purposes is a simple *linear regression* model:

$$r_{IBM} = a + \beta r_{S\&P} + \varepsilon$$

where  $\varepsilon$  is white noise (i.e. mean zero, normally distributed and independent of everything)

---





# Model selection issues

---

- Classic finance fails in practice
- So people fumble around for better models...
- ... adding *covariates (independent variables)* they find useful
- See Fama-French's data page for more



---

# Example 3: Marketing

---



# Spending forecast

---

- How much should I expect a new customer to spend on my service per year given their observed characteristics?
- Data: dataset of existing customer characteristics and spending history
- What model to select?
- This is the toughest question one ever asks in Stats
- It is full of pitfalls we will discuss at length, such as overfitting



---

# Example 4: Promotion Budgeting



# Promotion budget optimization

---

- Budget of  $\$B$  to boost new customer spending over the next year via advertising
- Potential target types:  $i = 1, 2, \dots, N$
- Select an amount  $c(i)$  to spend on type  $i$  subject to:

$$\sum_i c(i) \leq B$$

to maximize:

$$\sum_i [N(c(i)) - N(0)] S(i)$$

where

- $N(c(i))$  is the number of type  $i$  consumers who will join given  $c(i)$
- $S(i)$  is their expected spending if they join



# Task list and issues

---

1. Estimate  $N(c)$  (a “treatment effect” problem) using historical data and/or experimentation
2. Estimate  $S(i)$
3. This second step is fraught with selection problems:
  - a. Will new customers spend like our existing, observably similar customers?
  - b. Why were they not customers before the incentives?
4. Solve maximization problem (econ problem, we’ve got this)

