Week 9 – **Percentiles, CIs**

Slides by Suraj Rampure

Fall 2017

Hypothesis Testing (Again)

Hypothesis Testing

Null Hypothesis (H0): Observations are due to random chance

Alternative Hypothesis (H1): Something other than chance has influenced the observations

1. State null and alternative **hypotheses**
2. Define and compute a **test statistic** to help choose between hypotheses
3. The **probability distribution** of the test statistic
    1. What the test statistic might be if the null hypothesis were true
    2. Approximate the probability distribution by an empirical distribution
4. **Conclusion: Is the observed statistic consistent with the null distribution?**

Putting it All Together

Let’s say I flipped a coin (that I thought was fair) 100 times, and saw 65 heads and 35 tails. Suppose I wanted to test whether or not the coin was actually fair.

Null Hypothesis: The coin is fair, and any results are due to random chance.

Alternative Hypothesis: There is something other than random chance influencing outcomes – this coin is biased towards heads.

Putting it All Together

```
coin = [“H”, “T”]
outcomes = make_array()
for i in np.arange(10000):
  flips = np.random.choice(coin, 100)
  outcomes = np.append(outcomes, np.count_nonzero(flips == “H”))
p_value = np.count_nonzero(outcomes >= 65) / 10000
```

The value of `p_value` after running the above code will be the empirical probability of seeing 65 or more heads in 100 flips assuming the null hypothesis is true (assuming that the coin is fair).

Putting it All Together

If you run the previous code yourself, you'll see that `p_value` is usually some extremely small decimal value (~ 0.002). This means the chances of seeing 65 heads in 100 flips with a fair coin is under 0.2%, meaning that it is extremely unlikely that the coin is fair.

Since 0.2% is lower than any reasonable p-value cutoff (1% or 5%), we'd **reject the null hypothesis** in this case. This means we don't believe the coin is actually fair, and that it is biased towards heads.

Percentiles

Percentiles

Given some set of values, percentiles help us describe **how large a value is** with respect to the rest of the set.

If a value is at the **p-th percentile** (often denoted "p%ile"), this means that it is **the smallest value at least as large as p%** of the values.

eg. 45th percentile: Smallest value at least as large as 45% of the values

Finding Percentiles with Math

To find the **p**-th percentile of a list of **n** numbers:

$$k = (p / 100) * n$$

The element in the list at spot **k** (counting starting at 1) is the **p**-th percentile. If **k** is not an integer, round it up to the nearest integer.

Note that this means that all percentiles must actually be elements in the list, unlike some other definitions of percentile. (eg. If we have a list with an even number of elements, its 50th percentile is always the left-middle element, not the mean of the left and right middles).

Finding Percentiles with Code

```
from datascience import *
numbers = make_array(10, 30, 40, 70, 90)
per_25 = percentile(25, numbers)
per_80 = percentile(80, numbers)
```

The **`percentile`** function takes in two arguments: a percentile value and a list. It returns the element of the list that is at that percentile.

Question: What is another name for the 50th percentile?

The median.

25th percentile: “First quartile”

50th percentile: “Second quartile”

75th percentile: “Third quartile”

Interquartile range: 75th percentile value - 25th percentile value

Confidence Intervals

Bootstrapping and Confidence Intervals

When we **bootstrap** a sample some number of times, we yield **estimates** for a parameter of the **population** that the sample came from.

For example, if we want to estimate the median height of all Berkeley undergrads, we can easily find a sample. However, the median of the sample isn't a good estimate of the true median, so we "sample from the sample" (bootstrap) several times.

Bootstrapping and Confidence Intervals

Now that we have some number (eg. 5000) of estimates for the median, we need some way of figuring out what the population median is.

We look at the **"inner 95%"** of these new resampled medians, and call this a **95% confidence interval**.

n% confidence means that **"by using this estimation process, this interval will contain the parameter about n% of the time"**.

Finding Confidence Intervals with Code

```
resampled_medians = [. . . . . .]
lower = percentile(2.5, numbers)
upper = percentile(97.5, numbers)
#95% confidence interval is from lower to upper
```

A **95% confidence interval** is the inner 95% of values. To find this interval, we take everything from the **2.5%ile to the 97.5%ile**. By doing this, we've cut off the largest and smallest 2.5% of values.