Discussion 3

DSC 80

2024-04-19

Section 1

WI23 Midterm Problem 6

tv_excl

	Title	Year	Age	IMDb	Rotten Tomatoes	Service
0	Jersey Shore	2009	16+	3.6	54	Hulu
1	Henry Hugglemonster	2013	all	5.3	42	Disney+
2	Fast & Furious Spy Racers	2019	7+	5.5	62	Netflix
3	Atlanta	2016	18+	8.6	84	Hulu
4	Played	2013	NaN	6.4	45	Prime Video

counts

Service	Disney+	Hulu	Netflix	Prime Video
Age				
13+	NaN	4.0	2.0	1.0
16+	13.0	405.0	320.0	147.0
18+	NaN	223.0	445.0	134.0
7+	91.0	246.0	245.0	149.0
all	116.0	97.0	151.0	144.0

Problem 1

Given the above information, what does the following expression evaluate to:

```
tv excl.groupby(["Age", "Service"]).sum().shape[0]
```

- Which DataFrame can we use to give us the answer?

Problem 1

Given the above information, what does the following expression evaluate to:

```
tv excl.groupby(["Age", "Service"]).sum().shape[0]
```

- Which DataFrame can we use to give us the answer?
- What, conceptually, does the given expression evaluate to?

Problem 1

Given the above information, what does the following expression evaluate to:

```
tv excl.groupby(["Age", "Service"]).sum().shape[0]
```

- Which DataFrame can we use to give us the answer?
- What, conceptually, does the given expression evaluate to?
- Hint: What does each value in `counts` refer to?

Solution

The solution is to just count all of the non-null values in `counts`, since each one represents a combination of `Age` and `Service` from `tv_excl`. - That's why we noted that `counts` includes *every* valid combination.

Problem 2

Tiffany would like to compare the distribution of `Age` for Hulu and Netflix. Specifically, she'd like to test the following hypotheses:

- **Null Hypothesis:** The distributions of `Age` for Hulu and Netflix are drawn from the same population distribution, and any observed differences are due to random chance.
- **Alternative Hypothesis:** The distributions of `Age` for Hulu and Netflix are drawn from different population distributions.

Is this a hypothesis test, or a permutation test?

Hypothesis Testing

So what is a hypothesis test, anyway?

Let's say we have a result, and we'd like to know whether or not that result means anything.

In order to make sure, we simulate a similar picture of the dataset *assuming that nothing is happening*, and see how often an effect that large occurs. This is why we reject/fail to reject w.r.t. the null hypothesis, not the alternate.

Note: In practice, scientific papers rarely simulate to generate p-values

Operationalizing

So now, how do the pieces we're talking about today fit into that:

- **Test statistic:** the value we'll use to compare results
  - e.g. TVD, difference in means, the mean itself
  - Should capture the differences you care about!
- **p-value**: how unlikely the observed test statistic needs to be under H_0 to reject H_0 – you set this beforehand

Permutation Testing

A permutation test is a special case of hypothesis testing, where what we want to test is whether two samples were drawn from the same distribution.

Specifically, we're shuffling the group assignment as a method of generating samples under the null hypothesis!

Total Variation Distance

```
hn = counts[["Hulu", "Netflix"]]
# Note that distr has 2 rows and 5 columns.
distr = (hn / hn.sum()).T
```

To test the hypotheses above, Tiffany decides to use the total variation distance as her test statistic. Which of the following expressions **DO NOT** correctly compute the observed statistic for her test?

TVD (cont.)

- First – what is the DataFrame `distr` going to look like?

TVD (cont.)

- First – what is the DataFrame `distr` going to look like?
- Ignoring all of the potential given solutions, how might you calculate the TVD from there?

TVD (cont.)

- First – what is the DataFrame `distr` going to look like?
- Ignoring all of the potential given solutions, how might you calculate the TVD from there?
- Now we're going to go over each of the solutions in turn.

Things to Remember

- This is a real "you should be able to manipulate DataFrames in your head" type question!
- One big key to this question is knowing how the `axis` keyword works!
  - In most cases, the way I think about it is: `axis = 0` sums *vertically*, while `axis = 1` sums *horizontally*
  - Also, methods like `.diff()` and `.sum()` default to `axis = 0`, so even when the `axis` keyword isn't visibly present, you should still be aware of what's going on.

Section 2

WI23 Final Problem 2.5

WI23 Final Problem 2.5

$\text{TVD}(\vec{a}, \vec{b}) = \frac{1}{2} \sum_{i=1}^{n} |a_i - b_i|$

$\text{dis1}(\vec{a}, \vec{b}) = \vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2 + ... + a_n b_n$

$\text{dis2}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} = \frac{a_1 b_1 + a_2 b_2 + ... + a_n b_n}{\sqrt{a_1^2 + a_2^2 + ... + a_n^2}\sqrt{b_1^2 + b_2^2 + ... + b_n^2}}$

$\text{dis3}(\vec{a}, \vec{b}) = 1 - \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|}$

Yikes! Math! But if you're familiar with the dot product, you should be alright.

Dot Products

- The dot product is a very common vector similarity metric.
  - What's $(3,3) \cdot (2,2)$?
  - What's $(3,3) \cdot (-2,-2)$?
- But the dot product increases when vectors are scaled – we might not want that!

Dot Products

- The dot product is a very common vector similarity metric.
  - What's $(3, 3) \cdot (2, 2)$?
  - What's $(3, 3) \cdot (-2, -2)$?
- But the dot product increases when vectors are scaled – we might not want that!
- We can normalize the dot product by dividing by the length of each vector

Dot Products

- The dot product is a very common vector similarity metric.
  - What's $(3,3) \cdot (2,2)$?
  - What's $(3,3) \cdot (-2,-2)$?
- But the dot product increases when vectors are scaled – we might not want that!
- We can normalize the dot product by dividing by the length of each vector
- By length, I mean the Euclidean norm – if you feel like looking up some math, look up the definition of a p-norm, it's somewhat interesting.

Dot Products

- The dot product is a very common vector similarity metric.
  - What's $(3, 3) \cdot (2, 2)$?
  - What's $(3, 3) \cdot (-2, -2)$?
- But the dot product increases when vectors are scaled – we might not want that!
- We can normalize the dot product by dividing by the length of each vector
- By length, I mean the Euclidean norm – if you feel like looking up some math, look up the definition of a p-norm, it's somewhat interesting.
- So, our solution is `dis3` – it's the only one that's both normalized to (0,1), and where, like TVD, smaller values are more similar.

Section 3

FA23 Midterm Problem 4

Donkey Data

We're working with the DataFrame `donkeys`, described below.

	id	BCS	Age	Weight	WeightAlt
0	d01	3.0	<2	77	NaN
1	d02	2.5	<2	100	NaN
2	d03	1.5	<2	74	NaN

- `id` A unique identifier for each donkey (`d01`, `d02`, etc.).
- `BCS` Body condition score: from 1 (emaciated) to 3 (healthy) to 5 (obese) in increments of 0.5.
- `Age` Age in years: <2, 2–5, 5–10, 10–15, 15–20, and over 20 years.
- `Weight` Weight in kilograms.
- `WeightAlt` Second weight measurement taken for 30 donkeys. NaN if the donkey was not reweighed.

Problem

Alan wants to see whether donkeys with BCS ≥ 3 have larger `Weight` values on average compared to donkeys that have BCS < 3. Select all the possible test statistics that Alan could use to conduct this hypothesis test. Let μ_1 be the mean weight of donkeys with BCS ≥ 3 and μ_2 be the mean weight of donkeys with BCS < 3.

Solution

- We want a test statistic that compares the `Weight` of donkeys with `BCS` both $<$ and $\geq$ than 3.

Solution

- We want a test statistic that compares the `Weight` of donkeys with `BCS` both $<$ and $\geq$ than 3.
- Specifically, we want to know whether donkeys with `BCS` ≥ 3 have **larger** `Weight` values on average

Solution

- We want a test statistic that compares the `Weight` of donkeys with `BCS` both $<$ and $\geq$ than 3.
- Specifically, we want to know whether donkeys with `BCS` ≥ 3 have **larger** `Weight` values on average
- There are two options that work here: $\mu_1 - \mu_2$, and $2\mu_2 - \mu_1$

Solution

- We want a test statistic that compares the `Weight` of donkeys with `BCS` both $<$ and $\geq$ than 3.
- Specifically, we want to know whether donkeys with `BCS` ≥ 3 have **larger** `Weight` values on average
- There are two options that work here: $\mu_1 - \mu_2$, and $2\mu_2 - \mu_1$
- $2\mu_2 - \mu_1 = \mu_2 + (\mu_2 - \mu_1)$, so it's pretty much the same thing, just shifted upwards

Section 4

Attendance

Attendance

Once I give you a number, fill out the following Google form:
https://forms.gle/wP6ybKhG6H5E2wYH6