Discussion 5

DSC 80

2024-05-03

Section 1

FA23 Midterm Problem 4

Problem

	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. To generate a single sample under his null hypothesis, Alan should **(choose one)**:

- Resample 744 donkeys with replacement from donkeys.
- Resample 372 donkeys with replacement from donkeys with `BCS` < 3, and another 372 donkeys with `BCS` >= 3.
- Randomly permute the `Weight` column.

Solution

- What's the null hypothesis?

Solution

- What's the null hypothesis?
- What do answers (A) and (B) try to do?

Solution

- What's the null hypothesis?
- What do answers (A) and (B) try to do?
- Why might part (C) work?

Solution

- What's the null hypothesis?
- What do answers (A) and (B) try to do?
- Why might part (C) work?
- The bootstrapping procedure, part (A), will yield a similar distribution, but adds additional randomness

Problem

Doris wants to use multiple imputation to fill in missing values in `WeightAlt`.

She knows that `WeightAlt` is MAR on `BCS` and `Age`, so she will perform multiple imputation conditional on `BCS` and `Age` – each missing value will be filled in with values from a random `WeightAlt` value from a donkey with the same `BCS` and `Age`. Assume that all `BCS` and `Age` combinations have observed `WeightAlt` values.

Fill in the blanks in the code below to estimate the median of `WeightAlt` using multiple imputation conditional on `BCS` and `Age` with 100 repetitions.

Fill in the blanks

```
def impute(col):
    col = col.copy()
    n = _________
    fill = np.random.choice(_____)
    col[_____] = fill
return col

results = []
for i in range(___):
    imputed = donkeys._____(_______)['WeightAlt'
        ]._______(_________)
    results.append (imputed.median())
```

Comments

- These kinds of problems are sort of like doing a crossword or sudoku puzzle!
- Fill in the things you're most confident about, propagate those changes, see whether it makes sense at each step

Section 2

WI23 Final Exam Problem 1

States

The DataFrame `sat` contains one row for most combinations of `Year` and `State`, where `Year` ranges between 2005 and 2015 and `State` is one of the 50 states (not including the District of Columbia). Assume `sat` does not contain any duplicate rows — that is, there is only one row for every unique combination of `Year` and `State` that is in sat – and that `sat` does not contain any null values.

Data

	Year	State	# Students	Math	Verbal
0	2014	Washington	41277	519	510
1	2013	Arizona	22283	529	522
2	2006	Kansas	2545	591	582
3	2011	North Dakota	219	612	586
4	2009	New Mexico	2209	548	553

Problem

It turns out that there are 11 rows in sat for all 50 states, except for one state.

Fill in the blanks below so that missing years evaluates to an array, sorted in any order, containing the years for which that one state does not appear in `sat`.

More Blanks

```
state_only = sat.groupby("State").filter(___(a)___)

merged = sat["Year"].value_counts().to_frame().merge(
    state_only, ___(b)___
)

missing_years = ___(c)___.to_numpy()
```

Solution

- This problem gives you a bit more structure – you should be able to generally tell what each step is doing without filling in the blanks!

Test Statistics

The following DataFrame contains summary statistics for all SAT takers in New York and Texas from 2005 to 2015. Suppose we want to run a statistical test to assess whether the distributions of the number of students between 2005 and 2015 in New York and Texas are significantly different.

State	mean	median	std
New York	157950.818182	157989.0	3430.986500
Texas	155035.909091	148102.0	22509.092685

Problem

Given the above DataFrame, which test statistic is **most likely** to yield a significant difference?

- mean # of students in Texas – mean # of students in New York
- |mean # of students in Texas – mean # of students in New York|
- |median # of students in Texas – median # of students in New York|
- The Kolmogorov-Smirnov statistic

Solution

- We talked last week about how summary statistics only capture certain things about data

Solution

- We talked last week about how summary statistics only capture certain things about data
- We want to know which test statistic is more likely to yield a significant result – what does that mean?

Solution

- We talked last week about how summary statistics only capture certain things about data
- We want to know which test statistic is more likely to yield a significant result – what does that mean?
- Which test statistic has captures the difference between distributions the best?

Section 3

Attendance

Attendance

Once I give you a number, fill out the following Google form: