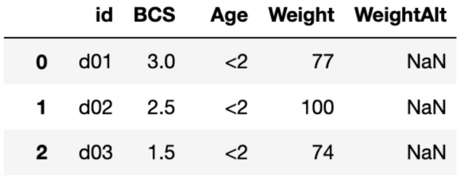
DSC 80 Discussion 4 Worksheet

Name: ________________________

1 FA23 Midterm Problem 3

The `donkeys` table contains data from a research study about donkey health. The researchers measured the attributes of 544 donkeys. The next day, they selected 30 donkeys to reweigh. The first few rows of the `donkeys` table are shown below (left), and the table contains the following columns (right):



	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.

Consider the following scenarios for how the researchers chose the 30 donkeys to reweigh. In each scenario, select if the missing mechanism for the `WeightAlt` column is NMAR, MAR, or MCAR.

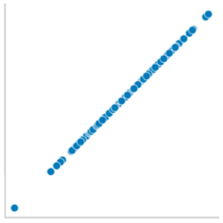
Note: Although the missing data are missing by design from the perspective of the original researchers, since we can't directly recover the missing values from our other data, we can treat the missing data as NMAR, MAR, or MCAR.

- A. The researchers chose the 30 donkeys with the largest `'Weight'` values to reweigh.
- B. The researchers drew 30 donkeys uniformly at random without replacement from the donkeys with `BCS` score of 44 or greater.
- C. The researchers set `i` as a number drawn uniformly at random between 0 and 514, then reweighed the donkeys in `donkeys.iloc[i:i+30]`.
- D. The researchers reweighed all the donkeys, but deleted all the values in `'WeightAlt'` except for the 30 lowest values.
- E. The researchers split up the donkeys into the 6 different age groups, then sampled 5 donkeys uniformly at random without replacement within each age group.

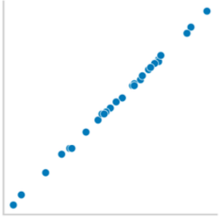
For this next question, assume that the researchers chose the 30 donkeys to reweigh by drawing **a simple random sample of 30 underweight donkeys: donkeys with BCS values of 1, 1.5, or 2.** The researchers weighed these 30 donkeys one day later and stored the results in `'WeightAlt'`.

Which of the following shows the scatter plot of `'WeightAlt' - 'Weight'` on the y-axis and `'Weight'` on the x-axis? Assume that missing values are not plotted.

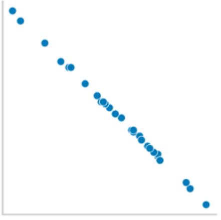
A)



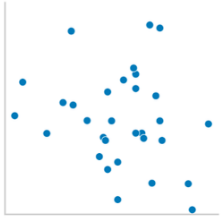
B)



C)

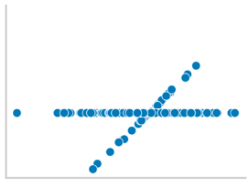


D)

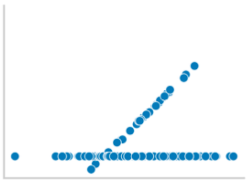


Suppose we use mean imputation to fill in the missing values in `'WeightAlt'`. Select the scatter plot `'WeightAlt'` on `'Weight'` after imputation.

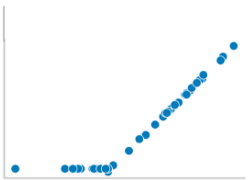
A)



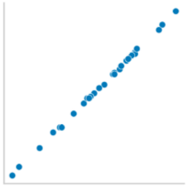
B)



C)

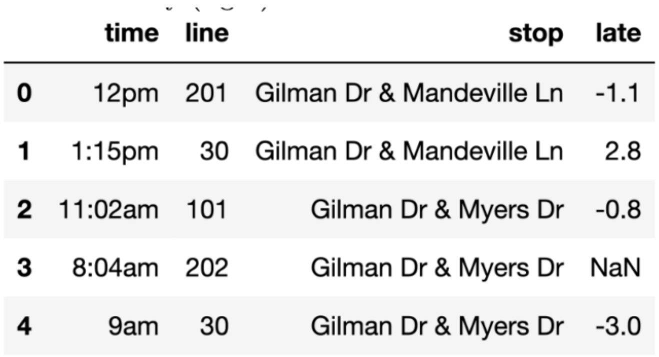


D)



2 FA23 Final Problem 3

The `bus` table (left) records bus arrivals over 1 day for all the bus stops within a 2 mile radius of UCSD. The data dictionary (right) describes each column.



	time	line	stop	late
0	12pm	201	Gilman Dr & Mandeville Ln	-1.1
1	1:15pm	30	Gilman Dr & Mandeville Ln	2.8
2	11:02am	101	Gilman Dr & Myers Dr	-0.8
3	8:04am	202	Gilman Dr & Myers Dr	NaN
4	9am	30	Gilman Dr & Myers Dr	-3.0

`time` Time of arrival (`str`). Note that the times are inconsistently entered (e.g. `12pm` vs. `1:15pm`).

`line` Bus line (`int`). There are multiple buses per bus line each day.

`stop` Bus stop (`str`).

`late` The number of minutes the bus arrived after its scheduled time. Negative numbers mean that the bus arrived early (`float`). Some entries in this column are missing.

For each of the following questions, select the correct procedure to simulate a single sample under the null hypothesis, and the correct test statistic for the hypothesis test. Assume that the `'time'` column of the `bus` DataFrame has already been parsed into timestamps.

Are buses equally likely to be early or late?

- [] `np.random.choice([-1, 1], bus.shape[0])`
- [] `np.random.choice(bus['late'], bus.shape[0], replace = True)`
- [] Randomly permute the `'late'` column

Test statistic:

`[] Number of values below 0 [] np.mean [] np.std [] TVD [] K-S statistic`

Is the `'late'` column MAR dependent on the `'line'` column?

- [] `np.random.choice([-1, 1], bus.shape[0])`
- [] `np.random.choice(bus['late'], bus.shape[0], replace = True)`
- [] Randomly permute the 'late' column

Test statistic:

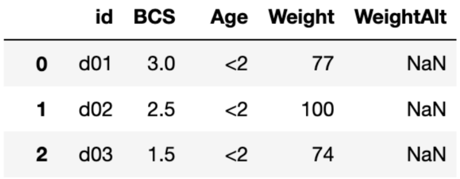
`[] Absolute difference in means [] Absolute difference in proportions [] TVD [] K-S statistic`

DSC 80 Discussion 5 Worksheet

Name: ______________________________

1 FA23 Midterm Problem 4

In this question, we will continue to work with the `donkeys` dataset from Problem 3. The first few rows of the table column descriptions are shown again below for convenience.



	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.

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)**:

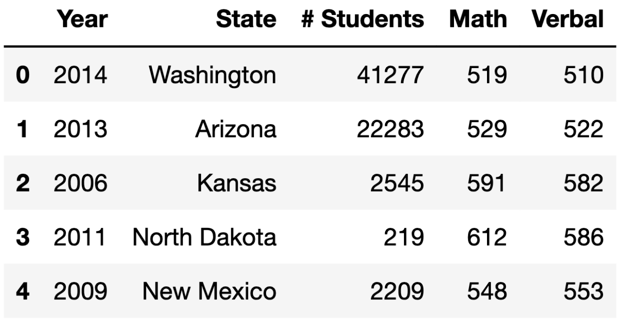
- Resample 744 donkeys with replacement from donkeys.
- Resample 372 donkeys with replacement from donkeys with `BCS` < 3, and another 372 donkeys with `BCS` ≥ 3.
- Randomly permute the `Weight` column.

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.

```
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())
```

2 WI23 Final Exam Problem 1

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.

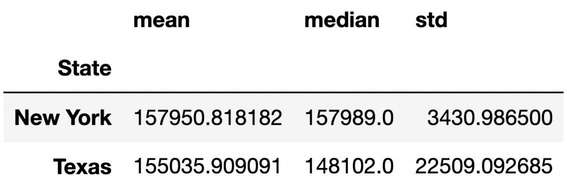


	Year	State	# Students	Math	Verbal
0	2014	Washington	41277	519	510
1	2013	Arizona	22283	529	522
2	2006	Kansas	2545	591	582
3	2011	North Dakota	219	612	586
4	2009	New Mexico	2209	548	553

The data description stated that there is one row in `sat` for most combinations of `Year` (between 2005 and 2015, inclusive) and `State`. 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`.

```
state_only = sat.groupby("State").filter(______________________)
merged = sat["Year"].value_counts().to_frame().merge(
    state_only,
   _________________________________________________________________
   )
missing_years =
   ____________________________________________________________.
   to_numpy()
```

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.



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

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

- A. mean number of students in Texas – mean number of students in New York
- B. |mean number of students in Texas – mean number of students in New York|
- C. |median number of students in Texas – median number of students in New York|
- D. The Kolmogorov-Smirnov statistic