Final Exam - DSC 10, Spring 2024

Full Name:	
PID:	
Seat you are in:	

Instructions:

- This exam consists of 16 questions, worth a total of 238 points.
- Write your PID in the top right corner of each page in the space provided.
- Please write **clearly** in the provided answer boxes; we will not grade work that appears elsewhere. Completely fill in bubbles and square boxes; if we cannot tell which option(s) you selected, you may lose points.

  ○ A bubble means that you should only **select one choice**.

  □ A square box means you should **select all that apply**.
- If your answer is a string, make sure to put it in quotes. If your answer is a float, make sure to include a decimal point.
- Aside from the provided Reference Sheet, you may not refer to any other resources or technology during the exam (no notes, no calculators).

By signing below, you are agreeing that you will behave honestly and fairly during and after this exam.

Signature:

Version A

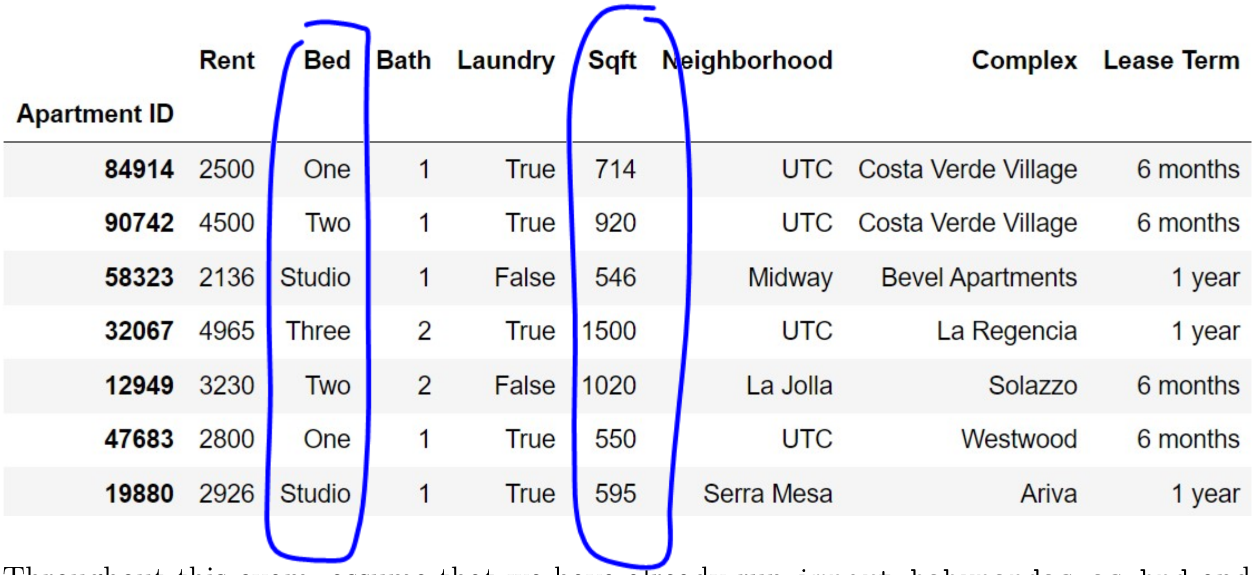
Please do not open your exam until instructed to do so.

San Diego Apartments

Today, we'll explore data on 800 different apartments available for rent in San Diego. Each row in the DataFrame `apts` corresponds to an individual apartment. The DataFrame is indexed by `"Apartment ID" (int)`, which is a unique identifier for the apartment. The columns of `apts` are as follows:

- `"Rent" (int)`: The monthly rent for the apartment, in dollars.
- `"Bed" (str)`: The number of bedrooms in the apartment. Values are `"Studio"`, `"One"`, `"Two"`, and `"Three"`.
- `"Bath" (str)`: The number of bathrooms in the apartment. Values are `"One"`, `"One and a half"`, `"Two"`, `"Two and a half"`, and `"Three"`.
- `"Laundry" (bool)`: If the apartment comes with an in-unit washer and dryer.
- `"Sqft" (int)`: The area of the apartment, in square feet.
- `"Neighborhood" (str)`: The neighborhood in which the apartment is located.
- `"Complex" (str)`: The complex the apartment is a part of.
- `"Lease Term" (str)`: The duration of the apartment's lease. Values are `"1 month"`, `"6 months"`, and `"1 year"`.

The first few rows of `apts` are shown below, though `apts` has many more rows than pictured, 800 in total. The data in `apts` is only a sample from the much larger population of **all** San Diego apartments.



	Rent	Bed	Bath	Laundry	Sqft	Neighborhood	Complex	Lease Term
Apartment ID								
84914	2500	One	1	True	714	UTC	Costa Verde Village	6 months
90742	4500	Two	1	True	920	UTC	Costa Verde Village	6 months
58323	2136	Studio	1	False	546	Midway	Bevel Apartments	1 year
32067	4965	Three	2	True	1500	UTC	La Regencia	1 year
12949	3230	Two	2	False	1020	La Jolla	Solazzo	6 months
47683	2800	One	1	True	550	UTC	Westwood	6 months
19880	2926	Studio	1	True	595	Serra Mesa	Ariva	1 year

Throughout this exam, assume that we have already run `import babypandas as bpd` and `import numpy as np`.

Important: Before proceeding, make sure to read the page called *San Diego Apartments*.

Question 1 (12 pts)

Yutian wants to rent a one-bedroom apartment, so she decides to learn about the housing market by plotting a density histogram of the monthly rent for all one-bedroom apartments in the `apts` DataFrame. In her call to the DataFrame method `.plot`, she sets the bins using the parameter

```
bins = np.arange(0, 10000, 100).
```

a) (4 pts) How many bins will this histogram have?

b) (4 pts) Suppose there are 300 one-bedroom apartments in the `apts` DataFrame, and 15 of them cost between \$2,300 (inclusive) and \$2,400 (exclusive). How tall should the bar for the bin `[2300, 2400)` be in the density histogram? Give your answer as a simplified fraction or exact decimal.

c) (4 pts) Suppose some of the one-bedroom apartments in the `apts` DataFrame cost more than \$5,000. Next, Yutian plots another density histogram with

```
bins = np.arange(0, 5000, 100).
```

Consider the bin `[2300, 2400)` in this new histogram. Is it taller, shorter, or the same height as in the old histogram, where the bins were `np.arange(0, 10000, 100)`?

○ Taller
○ Shorter
○ The same height
○ Not enough information to answer

Question 2 (16 pts)

Michelle and Abel are each touring apartments for where they might live next year. Michelle wants to be close to UCSD so she can attend classes easily. Abel is graduating and wants to live close to the beach so he can surf. Each person makes their own DataFrame (called `michelle` and `abel` respectively), to keep track of all the apartments that they toured. Both `michelle` and `abel` came from querying `apts`, so both DataFrames have the same columns and structure as `apts`.

Here are some more details about the apartments they toured:

- Michelle toured **one bedroom and studio** apartments at 12 different complexes, or 24 apartments total.
- Abel toured **one bedroom and two bedroom** apartments at 20 different complexes, or 40 apartments total.
- There are 8 complexes that are near both UCSD and the beach, and both Michelle and Abel toured these complexes.

We'll assume for this problem only that there is just one apartment of each size available at each complex, so that if they both tour a one bedroom apartment at the same complex, it is the exact same apartment with the same `"Apartment ID"`.

a) (4 pts) What does the following expression evaluate to?

```
michelle.merge(abel, left_index=True, right_index=True).shape[0]
```

b) (4 pts) What does the following expression evaluate to?

```
michelle.merge(abel, on = "Bed").shape[0]
```

c) (4 pts) What does the following expression evaluate to?

```
michelle.merge(abel, on = "Complex").shape[0]
```

d) (4 pts) What does the following expression evaluate to?

```
abel.merge(abel, on = "Bed").shape[0]
```

Question 3 (17 pts)

We wish to compare the average rent for studio apartments in different complexes.

a) (8 pts) Our goal is to create a DataFrame `studio_avg` where each complex with studio apartments appears once. The DataFrame should include a column named `"Rent"` that contains the average rent for all studio apartments in that complex. For each of the following strategies, determine if the code provided works as intended, gives an incorrect answer, or errors.

(i)

```
studio = apts[apts.get("Bed") == "Studio"]
studio_avg = studio.groupby("Complex").mean().reset_index()
```

○ Works as intended ○ Gives an incorrect answer ○ Errors

(ii)

```
studio_avg = apts.groupby("Complex").min().reset_index()
```

○ Works as intended ○ Gives an incorrect answer ○ Errors

(iii)

```
grouped = apts.groupby(["Bed", "Complex"]).mean().reset_index()
studio_avg = grouped[grouped.get("Bed") == "Studio"]
```

○ Works as intended ○ Gives an incorrect answer ○ Errors

(iv)

```
grouped = apts.groupby("Complex").mean().reset_index()
studio_avg = grouped[grouped.get("Bed") == "Studio"]
```

○ Works as intended ○ Gives an incorrect answer ○ Errors

b) (6 pts) Consider the DataFrame `alternate_approach` defined as follows

```
grouped = apts.groupby(["Bed", "Complex"]).mean().reset_index()
alternate_approach = grouped.groupby("Complex").min()
```

Suppose that the `"Rent"` column of `alternate_approach` has all the same values as the `"Rent"` column of `studio_avg`, where `studio_avg` is the DataFrame described in part (a). Which of the following are valid conclusions about `apts`? Select all that apply.

- □ No complexes have studio apartments.
- □ Every complex has at least one studio apartment.
- □ Every complex has exactly one studio apartment.
- □ Some complexes have only studio apartments.
- □ In every complex, the average price of a studio apartment is less than or equal to the average price of a one bedroom apartment.
- □ In every complex, the single cheapest apartment is a studio apartment.
- □ None of these.

c) (3 pts) Which data visualization should we use to compare the average prices of studio apartments across complexes?

○ Line chart ○ Bar chart ○ Scatter plot ○ Histogram

Question 4 (8 pts)

According to Chebyshev's inequality, at least 80% of San Diego apartments have a monthly parking fee that falls between $30 and $70.

a) (4 pts) What is the average monthly parking fee?

b) (4 pts) What is the standard deviation of monthly parking fees?

○ $\frac{20}{\sqrt{5}}$ ○ $\frac{40}{\sqrt{5}}$ ○ $20\sqrt{5}$ ○ $40\sqrt{5}$

Question 5 (6 pts)

You are given the following information about security deposits for a sample of 400 apartments in the Mission Hills neighborhood of San Diego:

- Mean deposit: $2,300
- Standard Deviation of deposits: $500

Using the fact that `scipy.stats.norm.cdf(-0.8)` evaluates to about `0.21`, construct a 58% confidence interval for the mean security deposit of all Mission Hills apartments. Below, give the endpoints of your confidence interval, both as integers.

Left endpoint: Right endpoint:

hyp test disguised "is the coin fair"

Question 6 (22 pts)

You want to use the data in `apts` to test both of the following pairs of hypotheses:

Pair 1:

- = **Null Hypothesis**: In San Diego, the number of one bedroom apartments available for rent is **equal** to the number of two bedroom apartments available for rent.
- > **Alternative Hypothesis**: In San Diego, the number of one bedroom apartments available for rent is **greater** than the number of two bedroom apartments available for rent.

prop_1br

Null | Alt →

Pair 2:

- = **Null Hypothesis**: In San Diego, the number of one bedroom apartments available for rent is **equal** to the number of two bedroom apartments available for rent.
- ≠ **Alternative Hypothesis**: In San Diego, the number of one bedroom apartments available for rent is **not equal** to the number of two bedroom apartments available for rent.

In `apts`, there are 467 apartments that are either one bedroom or two bedroom apartments. You perform the following simulation under the assumptions of the null hypothesis.

0.52 →

0.02 →

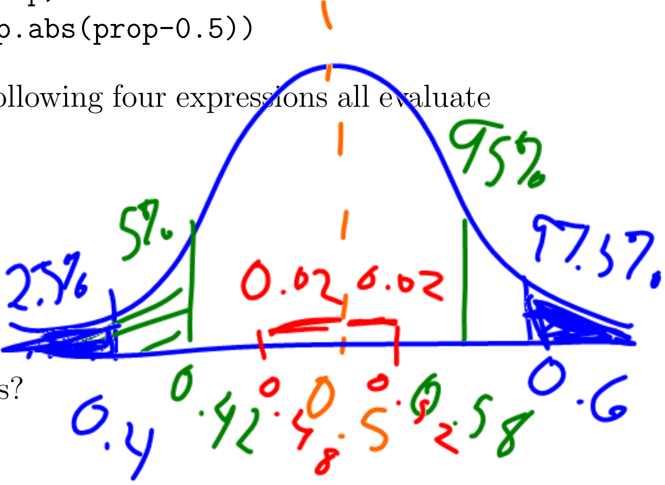
1br 2br

[0.52, 0.48]

```
prop_1br = np.array([])
abs_diff = np.array([])
for i in np.arange(10000):
    prop = np.random.multinomial(467, [0.5, 0.5])[0]/467
    prop_1br = np.append(prop_1br, prop)
    abs_diff = np.append(abs_diff, np.abs(prop-0.5))
```

You then calculate some percentiles of `prop_1br`. The following four expressions all evaluate to `True`.

```
np.percentile(prop_1br, 2.5) == 0.4
np.percentile(prop_1br, 5) == 0.42
np.percentile(prop_1br, 95) == 0.58
np.percentile(prop_1br, 97.5) == 0.6
```



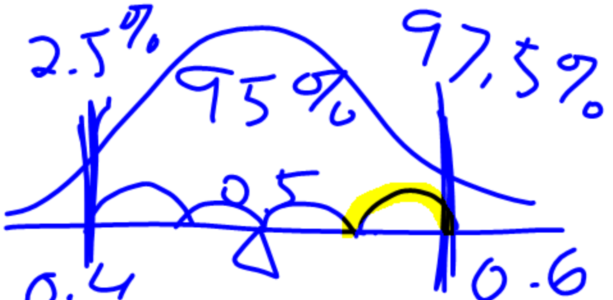
→ **a)** (4 pts) What is `prop_1br.mean()` to two decimal places?

$\boxed{0.5}$

prop_1br bell curve bc of CLT

→ **b)** (4 pts) What is `np.std(prop_1br)` to two decimal places?

$\boxed{0.05}$

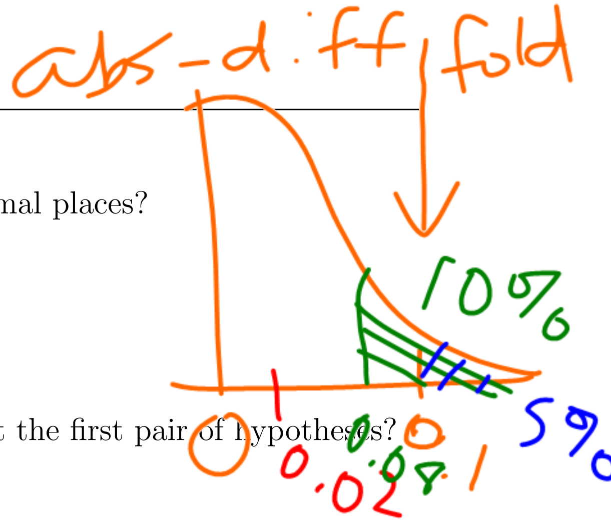


$$= \frac{0.6-0.4}{4}$$

0, 1, 1, 0, 1 ⇒ $\frac{3}{5}$

proportions are means

PID: ______________________



c) (4 pts) What is `np.percentile(abs_diff, 95)` to two decimal places?

0.1

d) (3 pts) Which simulated test statistics should be used to test the first pair of hypotheses?

☒ `prop_1br` ○ `abs_diff`

e) (3 pts) Which simulated test statistics should be used to test the second pair of hypotheses?

○ `prop_1br` ☒ `abs_diff`

f) (4 pts) Your observed data in `apts` is such that you reject the null for the first pair of hypotheses at the 5% significance level, but fail to reject the null for the second pair at the 5% significance level. What could the value of the following proportion have been?

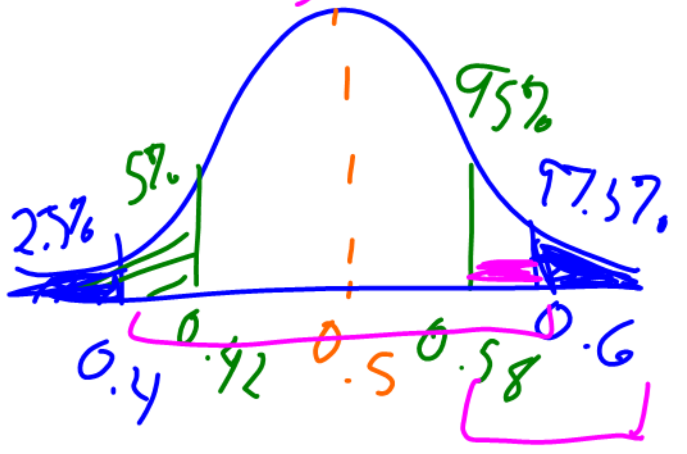
$$\frac{\text{\# of one bedroom apartments in apts}}{\text{\# of one bedroom apartments in apts} + \text{\# of two bedroom apartments in apts}}$$

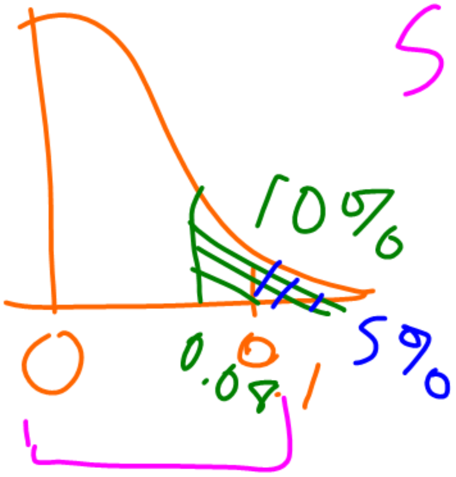
Give your answer as a number to two decimal places.

0.59

in rightmost 5%

not in rightmost 5%





observed prop ≥ 0.58

Question 7 (32 pts)

You want to know how much extra it costs, on average, to have a washer and dryer in your apartment. Since this cost is built into the monthly rent, it isn't clear how much of your rent will be going towards this convenience. You decide to bootstrap the data in `apts` to estimate the average monthly cost of having in-unit laundry.

a) (12 pts) Fill in the blanks to generate 10,000 bootstrapped estimates for the average monthly cost of in-unit laundry.

```
yes = apts[apts.get("Laundry")]
no = apts[apts.get("Laundry") == False]
laundry_stats = np.array([])
for i in np.arange(10000):
    yes_resample = yes.sample(__(a)__, __(b)__)
    no_resample = no.sample(__(c)__, __(d)__)
    one_stat = __(e)__
    laundry_stats = np.append(laundry_stats, one_stat)
```

a: b:

c: d:

e:

b) (8 pts) What if you wanted to instead estimate the average **yearly** cost of having in-unit laundry?

(i) Below, change blank `(e)`, such that the procedure now generates 10,000 bootstrapped estimates for the average **yearly** cost of in-unit laundry.

e:

(ii) Suppose you ran your original code from part (a) and used the results to calculate a confidence interval for the average **monthly** cost of in-unit laundry, which came out to be

$$[L_M, R_M].$$

Then, you changed blank `(e)` as you described above, and ran the code again to calculate a different confidence interval for the average **yearly** cost of-unit laundry, which came out to be

$$[L_Y, R_Y].$$

Which of the following is the best description of the relationship between the endpoints of these confidence intervals? Note that the symbol ≈ means "approximately equal".

○ $L_Y = 12 * L_M$ and $R_Y = 12 * R_M$
○ $L_Y \approx 12 * L_M$ and $R_Y \approx 12 * R_M$
○ $L_M = 12 * L_Y$ and $R_M = 12 * R_Y$
○ $L_M \approx 12 * L_Y$ and $R_M \approx 12 * R_Y$
○ None of these.

c) (4 pts) You're concerned about the validity of your estimates because you think bigger apartments are more likely to have in-unit laundry. Instead, you decide to estimate the average monthly cost of in-unit laundry for one bedroom apartments only.

If your concern is valid and it is true that bigger apartments are more likely to have in-unit laundry, how will your bootstrapped estimates for the average monthly cost of in-unit laundry for one bedroom apartments only compare to the values you computed in part (a) based on all of `apts`?

○ The estimates will be about the same.
○ The estimates will be generally larger than those you computed in part (a).
○ The estimates will be generally smaller than those you computed in part (a).

d) (8 pts) Consider the distribution of `laundry_stats` as computed in part (a). How would this distribution change if we:

(i) Increased the number of repetitions to 100,000?

○ The distribution would be wider.
○ The distribution would be narrower.
○ The distribution would not change significantly.

(ii) Started with only half of the rows in `apts`?

○ The distribution would be wider.
○ The distribution would be narrower.
○ The distribution would not change significantly.

Question 8 (8 pts)

Make sure to answer this problem using your best printed handwriting. We need to be able to clearly read every letter you write!

a) (4 pts) The management of the Solazzo apartment complex is changing the complex's name to be the output of the following line of code. Write the new name of this complex as a string.

Note that the string method `.capitalize()` converts the first character of a string to uppercase.

```
("Solazzo".replace("z", "ala" * 2)
          .split("aa")[-1]
          .capitalize()
          .replace("o", "Jo"))
```

b) (4 pts) The management of the Renaissance apartment complex has decided to follow suit and rename their complex to be the output of the following line of code. Write the new name of this complex as a string.

```
(("Renaissance".split("n")[1] + "e") * 2).replace("a", "M")
```

Question 9 (20 pts)

For each expression below, determine the data type of the output and the value of the expression, if possible. Write the data type in the left box and the value in the right box. If there is not enough information to determine the expression's value, write "Unknown" in the right box.

a) (4 pts) `apts.get("Rent").iloc[43] * 4 / 2`

type: □ value: □

b) (4 pts) `apts.get("Neighborhood").iloc[2][-3]`

type: □ value: □

c) (4 pts) `(apts.get("Laundry") + 5).max()`

type: □ value: □

d) (4 pts) `apts.get("Complex").str.contains("Verde")`

type: □ value: □

e) (4 pts) `apts.get("Sqft").median() > 1000`

type: □ value: □

for perm test, need
2 cols of data
1) category
2) numbers

See 9am class

Question 10 (28 pts)

We want to use the data in `apts` to test the following hypotheses:

- **Null Hypothesis**: The rents of the apartments in UTC and the rents of the apartments in other neighborhoods come from the same distribution.
- **Alternative Hypothesis**: The rents of the apartments in UTC are **higher** than the rents of the apartments in other neighborhoods on average.

comparing #s across categories

While we could answer this question with a permutation test in this problem we will explore another way to test these hypotheses. Since this is a question of whether two samples come from the same unknown population distribution, we need to construct a "population" to sample from. We will construct our "population" in the same way as we would for a permutation test, except we will draw our sample differently. Instead of shuffling, we will draw our two samples **with replacement** from the constructed "population." We will use as our test statistic the difference in means between the two samples (in the order **UTC minus elsewhere**).

a) (13 pts) Suppose the data in `apts`, which has 800 rows, includes 85 apartments in UTC. Fill in the blanks below so that `p_val` evaluates to the p-value for this hypothesis test, which we test according to the strategy outlined above.

```
diffs = np.array([])
for i in np.arange(10000):
    utc_sample_mean = __(a)__
    elsewhere_sample_mean = __(b)__
    diffs = np.append(diffs, utc_sample_mean - elsewhere_sample_mean)
observed_utc_mean = __(c)__
observed_elsewhere_mean = __(d)__
observed_diff = observed_utc_mean - observed_elsewhere_mean
p_val = np.count_nonzero(diffs __(e)__ observed_diff) / 10000
```

a:

b:

c:

d:

e: ○ > ○ >= ○ < ○ <= ○ == ○ !=

b) (5 pts) Now suppose we tested the same hypotheses with a permutation test using the same test statistic. Which of your answers above (part a) would need to change? Select all that apply.

□ blank `(a)`
□ blank `(b)`
□ blank `(c)`
□ blank `(d)`
□ blank `(e)`
□ None of these.

c) (10 pts) Now suppose we test the following pair of hypotheses.

- **Null Hypothesis**: The rents of the apartments in UTC and the rents of the apartments in other neighborhoods come from the same distribution.
- **Alternative Hypothesis**: The rents of the apartments in UTC are **different** than the rents of the apartments in other neighborhoods on average.

Then we can test this pair of hypotheses by constructing a 95% confidence interval for a parameter and checking if some particular number, x, falls in that confidence interval. To do this:

(i) What parameter should we construct a 95% confidence interval for? Your answer should be a phrase or short sentence.

(ii) What is the value of x? Your answer should be a number.

(iii) Suppose x is in the 95% confidence interval we create. Select all valid conclusions below.

□ We reject the null hypothesis at the 5% significance level.
□ We reject the null hypothesis at the 1% significance level.
□ We fail to reject the null hypothesis at the 5% significance level.
□ We fail to reject the null hypothesis at the 1% significance level.

Question 11 (8 pts)

Next year, six of the DSC tutors (Kathleen, Sophia, Kate, Ashley, Claire, and Vivian) want to rent a 3-bedroom apartment together. Each person will have a single roommate with whom they'll share a bedroom. They determine the bedroom assignments randomly such that each possible arrangement is equally likely.

For both questions below, give your answer as a simplified fraction or exact decimal.

Hint: Both answers can be expressed in the form $\frac{1}{k}$ for an integer value of k.

a) (4 pts) What is the probability that Kathleen and Sophia are roommates?
Hint: Think about the problem from the perspective of Kathleen.

b) (4 pts) What is the probability that the bedroom assignments are the following: Kathleen with Sophia, Kate with Ashley, and Claire with Vivian?

Question 12 (20 pts)

Suppose you know the following information.

- The average monthly rent for the apartments in `apts` is \$3,000 with a standard deviation of \$400.
- The average size of the apartments in `apts` is 2,000 square feet, with a standard deviation of 100 square feet.
- The correlation coefficient between rent and square footage is 0.9.

For all parts of this question, give your answer as an **integer**.

a) (4 pts) Suppose the rents are normally distributed. What is the rent below which 84% of apartments are priced?

b) (4 pts) Sophie's apartment rents for \$5,000. What is this rent in standard units?

c) (4 pts) Based on what you know about the rent of Sophie's apartment, use the regression line to predict the square footage of Sophie's apartment.

d) (4 pts) Sophie's apartment is actually 2,300 square feet. What is the residual of your prediction?

e) (4 pts) Cici's apartment is 1,800 square feet. Based on this information, use the regression line to predict the rent of Cici's apartment.

Question 13 (13 pts)

a) (3 pts) Values in the "Bath" column are "One", "One and a half", "Two", "Two and a half", and "Three". Fill in the blank in the function float_bath that will convert any string from the "Bath" column into the corresponding number of bathrooms, as a float. For example float_bath("One and a half") should return 1.5.

```
def float_bath(s):
    if "One" in s:
        n_baths =  1
    elif "Two" in s:
        n_baths = 2
    else:
        n_baths = 3
    if "and a half" in s:
        __(a)__
    return n_baths
```

What goes in blank (a)?

b) (6 pts) Values in the "Lease Term" column are "1 month", "6 months", and "1 year". Fill in the blanks in the function int_lease() that will convert any string from the "Lease Term" column to the corresponding length of the lease, in months, as an integer.

```
def int_lease(s):
    if s[-1] == "r":
        return __(b)__
    else:
        return __(c)__
```

What goes in blank (b)?

What goes in blank (c)?

c) (4 pts) Values in the "Bed" column are "Studio", "One", "Two", and "Three". The function int_bed provided below converts any string from the "Bed" column to the corresponding number of bedrooms, as an integer. Note that "Studio" apartments count as having 0 bedrooms.

```
def int_bed(s):
    if s == "Studio":
        return 0
    elif s == "One":
        return 1
    elif s == "Two":
        return 2
    return 3
```

Using the provided int_bed function, write one line of code that modifies the "Bed" column of the apts DataFrame so that it contains integers instead of strings.

Important: We will assume throughout the rest of this exam that we have converted the "Bed" column of apts so that it now contains ints.

Question 14 (6 pts)

Consider the following four slopes.

1. The slope of the regression line predicting "Rent" from "Sqft".
2. The slope of the regression line predicting "Sqft" from "Rent".
3. The slope of the regression line predicting "Rent" from "Bed".
4. The slope of the regression line predicting "Bed" from "Rent".

Note that we don't have enough information to calculate all of these slopes, but you should be able to answer the questions below based not on calculation, but on the interpretation of what these slopes represent in the context of housing.

a) (3 pts) Which of the above slopes do you expect to be the **largest**?

○ 1 ○ 2 ○ 3 ○ 4

b) (3 pts) Which of the above slopes do you expect to be the **smallest**?

○ 1 ○ 2 ○ 3 ○ 4

Question 15 (8 pts)

Imagine a DataFrame constructed from `apts` called `bedrooms`, which has one row for each bedroom in an apartment in `apts`. More specifically, a one bedroom apartment in `apts` will appear as one row in `bedrooms`, a two bedroom apartment in `apts` will appear as two rows in `bedrooms`, and a three bedroom apartment in `apts` will appear as three rows in `bedrooms`. Studio apartments will not appear in `bedrooms` at all.

The `"Apartment ID"` column of `bedrooms` contains the `"Apartment ID"` of the apartment in `apts`. Notice that this is not the index of `bedrooms` since these values are no longer unique. The `"Cost"` column of `bedrooms` contains the rent of the apartment divided by the number of bedrooms. All rows of `bedrooms` with a shared `"Apartment ID"` should therefore have the same value in the `"Cost"` column.

a) (3 pts) Recall that `apts` has 800 rows. How many rows does `bedrooms` have?

○ 800
○ More than 800.
○ Less than 800.
○ Not enough information.

b) (5 pts) Suppose `no_studio` is defined as follows. (Remember, we previously converted the `"Beds"` column to integers.)

```
no_studio = apts[apts.get("Bed") != 0]
```

Which of the following statements evaluates to the same value as the expression below?

```
bedrooms.get("Cost").mean()
```

Select all that apply.

□ `no_studio.get("Rent").mean()`
□ `no_studio.get("Rent").sum() / apts.get("Bed").sum()`
□ `(no_studio.get("Rent") / no_studio.get("Bed")).mean()`
□ `(no_studio.get("Rent") / no_studio.get("Bed").sum()).sum()`
□ `no_studio.get("Rent").mean() / no_studio.get("Bed").mean()`
□ None of these.

Question 16 (14 pts)

The table below shows the proportion of apartments of each type in each of three neighborhoods. Note that each column sums to 1.

Type	**North Park**	**Chula Vista**	**La Jolla**
Studio	0.30	0.15	0.40
One bedroom	0.40	0.35	0.30
Two bedroom	0.20	0.25	0.15
Three bedroom	0.10	0.25	0.15

a) (4 pts) Find the total variation distance (TVD) between North Park and Chula Vista. Give your answer as an exact decimal.

b) (4 pts) Which pair of neighborhoods is most similar in terms of types of housing, as measured by TVD?

○ North Park and Chula Vista

○ North Park and La Jolla

○ Chula Vista and La Jolla

c) (6 pts) 25% of apartments in Little Italy are one bedroom apartments. Based on this information, what is the minimum and maximum possible TVD between North Park and Little Italy? Give your answers as exact decimals.

Minimum: □ Maximum: □