Final Exam - DSC 10, Spring 2026

Full Name:	Solutions
PID:	A12345678

Instructions:

- This exam consists of 13 questions, worth a total of 165 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**.
- For full credit, your solutions must use methods of the course.
- You may use one page of double-sided handwritten notes. Aside from this, you may not refer to any other resources or technology during the exam. No calculators!
- Throughout this exam, assume we have already run `import babypandas as bpd`, `import numpy as np`, and `import scipy`.
- At any point, feel free to use functions and variables you defined in earlier parts of the same question.

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.

Important: Before proceeding, make sure to tear off the last page of this exam packet and read the data description.

Question 1 (14 pts)

The column `"member_names"` is formatted in a specific way. If a band has two members, the string is of the form `"A and B"` with no commas. If the band has three or more members, names are comma-separated like `"A, B, and C"` for a three-person band or `"A, B, C, and D"` for a four-person band. Take a look at the first few rows of the DataFrame to make sure you understand how this column is formatted!

a) (6 pts) Define the Series `names` and the function `f` as follows.

```
names = bands.get("member_names")

def f(x):
    return max(x, 2)
```

Select all expressions below that evaluate to a Series containing the number of members in each band. Assume that no band member has a comma (`","`) in their name, but don't make any other assumptions about band members' names.

☐ `names.str.replace(", ", "").str.split(" ").apply(len) - 1`

■ `names.str.split(", ").apply(len).apply(f)`

■ `(names.str.count(", ") + 1).apply(f)`

☐ `names.str.replace("and", ", ").str.count(", ").apply(f)`

☐ None of the above.

b) (4 pts) Suppose a function `g` is defined such that `bands.get("member_names").apply(g)` evaluates to a Series containing the number of members in each band. What are the data types of the input and output to `g`?

input data type: `str` output data type: `int`

c) (4 pts) Fill in the `return` statement of the function `g` as described above.

```
def g(x):
    return __(a)__
```

`(a)`: `max(x.count(",") + 1, 2)` or `max(len(x.split(",")), 2)`

Note: For the remainder of the exam, assume that we have added to `bands` a new column called `"members"` containing the number of members in each band.

PID: _____

Question 2 (10 pts)

Recall that we have added to `bands` a new column called `"members"` containing the number of members in each band.

a) (4 pts) Ella is curious if bands with more members tend to make more albums. Write one line of code to add a new column to `bands` called `"albums_per_member"`, containing the ratio of albums to members for each band.

Solution: `bands = bands.assign(albums_per_member = bands.get("albums") / bands.get("members"))`

b) (6 pts) Find the year in which the band with the highest `"albums_per_member"` ratio was formed. **Select all** options that correctly evaluate to this year (or any such year in the case of a tie).

- ☐ `bands.groupby("albums_per_member").min().sort_values(by=index, ascending=True).iloc[-1]`
- ■ `bands.groupby("albums_per_member").min().reset_index().sort_values( by="albums_per_member", ascending=False).get("formed").iloc[0]`
- ☐ `bands.groupby("formed").max().sort_values(by="albums_per_member", ascending=False).get("formed").iloc[0]`
- ■ `bands.groupby("formed").max().sort_values(by="albums_per_member", ascending=False).index[0]`
- ■ `bands.sort_values(by="albums_per_member", ascending=True).get("formed").iloc[-1]`
- ☐ `bands.sort_values(by="formed", ascending=False).groupby("albums_per_member").max().index[0]`
- ☐ None of the above.

Question 3 (10 pts)

For each goal below, determine the most appropriate plot type.

a) (2 pts) Show the relationship between each band's formation year and `"top_album_sales"` to see if older bands tend to have higher sales.

◯ Scatter plot ◯ Line plot ◯ Bar chart

b) (2 pts) Compare the total number of albums released by bands from each country, with countries sorted from most albums to fewest albums.

◯ Scatter plot ◯ Line plot ◯ Bar chart

c) (2 pts) Display how the average number of members in a band has changed over time.

◯ Scatter plot ◯ Line plot ◯ Bar chart

d) (2 pts) Compare `"top_album_sales"` across different genres to identify which genre has the highest median sales.

◯ Scatter plot ◯ Line plot ◯ Bar chart

e) (2 pts) Show the relationship between number of albums released and `"top_album_sales"` to see if one variable can be predicted from the other.

◯ Scatter plot ◯ Line plot ◯ Bar chart

PID: _____

Question 4 (8 pts)

Austin is analyzing the `"top_album_sales"` column of the `bands` DataFrame, which records each band's best-selling album sales in millions.

a) (4 pts) Fill in the blanks so that `[lower, upper]` is the interval that contains all values of `"top_album_sales"` that fall within 2 standard deviations of the mean.

```
sales = bands.get("top_album_sales")
lower = __(a)__
upper = __(b)__
```

(a): `sales.mean() - 2 * np.std(sales)`

(b): `sales.mean() + 2 * np.std(sales)`

Solution:

b) (4 pts) What proportion of `"top_album_sales"` fall in the interval `[lower, upper]`? Choose the best answer below.

◯ Approximately 0.68.

◯ Approximately 0.95.

● At least 0.75, but we don't know exactly.

◯ At least 0.88, but we don't know exactly.

◯ We cannot determine anything about this proportion.

Solution: $1 - \frac{1}{2^2} = 0.75$, by Chebyshev. The distribution is not necessarily normal so this is the most information we have.

Question 5 (17 pts)

Bianca standardizes the `"albums"` column, which has a mean of 7 and a standard deviation of 2 before standardization.

a) (4 pts) What are the mean and standard deviation of the `"albums"` column after standardization?

mean: 0 SD: 1

Solution: Standardization always produces mean 0 and standard deviation 1 by definition.

b) (4 pts) The band *Joy Division* has an `"albums"` value that is -2.5 standard units. How many albums did this band release?

2

Solution: Converting back from standard units: $x = \text{mean} - 2.5 \cdot \text{SD} = 7 - 2.5 \cdot 2 = 2$.

c) (4 pts) What proportion of bands have an `"albums"` value in the interval `[5, 9]`? Choose the best answer below.

◯ Approximately 0.68.

◯ Approximately 0.95.

◯ At least 0.75, but we don't know exactly.

◯ At least 0.88, but we don't know exactly.

● We cannot determine anything about this proportion.

Solution: Chebyshev's inequality does not guarantee anything within one standard deviation of the mean.

d) (5 pts) Which of the following **could not** be the value of a band's `"albums"` in standard units? **Select all** that apply.

■ -4

- ☐ -0.5
- ☐ 0
- ■ 1.75
- ☐ 8.5
- ☐ None of the above.

Question 6 (8 pts)

Histograms A, B, and C all display the distribution of `"top_album_sales"` for the 400 bands in `bands`, but they use different bins.

Histogram A uses `bins = [0, 10, 20, 30, 40, 50]`, with bar heights as follows:

Bin	Height
$[0, 10)$	0.03
$[10, 20)$	0.02
$[20, 30)$	0.01
$[30, 40)$	0.03
$[40, 50]$	0.01

a) (4 pts) Histogram B uses `bins = [0, 20, 40, 50]`. Note that not all bins have the same width. What is the height of the **tallest** bar in Histogram B? Give your answer as an **exact decimal**.

0.025

Solution:

Computing all three heights in Histogram B:

Bin	Proportion	Width	Height
$[0, 20)$	0.50	20	0.025
$[20, 40)$	0.40	20	0.020
$[40, 50]$	0.10	10	0.010

The $[0, 20)$ bar has the highest density (0.025) and is therefore the tallest.

b) (4 pts) Histogram C uses `bins = [0, 5, 10, 50]`, again of varying widths. Let h_1 be the height of the leftmost bar, $[0, 5)$. What are the minimum and maximum possible values of h_1? Give your answers as **exact decimals**.

minimum h_1: 0 maximum h_1: 0.06

Solution:

From Histogram A, the $[0, 10)$ bin has proportion 0.30 (area $= 0.03 \times 10 = 0.30$). When we split this bin, all of that 0.30 proportion must be distributed between $[0, 5)$ and $[5, 10)$. We don't know how it is split — a histogram does not show us the distribution *within* a bin.

Minimum h_1: It is possible that *none* of the data in $[0, 10)$ actually falls in $[0, 5)$ — all of it falls in $[5, 10)$. In that case, proportion in $[0, 5) = 0$, so $h_1 = 0/5 = \mathbf{0}$.

Maximum h_1: It is possible that all of the data in $[0, 10)$ falls in $[0, 5)$. In that case, proportion in $[0, 5) = 0.30$, so:

$$h_1 = \frac{0.30}{5} = 0.06$$

Question 7 (18 pts)

Zeyu is wondering whether active bands are more successful than inactive bands, in the sense that they have greater `"top_album_sales"`. To find out, he runs a permutation test on the `bands` DataFrame by comparing the average `"top_album_sales"` for active versus inactive bands using 10,000 shuffles of the `"active"` labels.

a) (4 pts) Which of the following best describes the **null hypothesis** for Zeyu's test?

◯ Active bands have higher top album sales than inactive bands, on average.

● Active bands and inactive bands have the same distribution of top album sales.

◯ In the full population of all bands, the average top album sales for active bands is higher than that for inactive bands.

◯ In the sample of data in `bands`, the average top album sales for active bands equals the average top album sales for inactive bands.

Solution: The permutation test's null hypothesis is that the two groups share the same distribution of top album sales, and any observed difference is due to chance label assignment. The first option is the alternative hypothesis; the third and fourth options are not the null.

b) (6 pts) Complete the code below to run Zeyu's permutation test.

```
active = bands[bands.get("active") == True].get("top_album_sales")
inactive = bands[bands.get("active") == False].get("top_album_sales")
obs_diff = active.mean() - inactive.mean()

sim_diffs = np.array([])

for i in np.arange(10000):
    shuffled_labels = __(a)__
    shuffled = bands.assign(active=shuffled_labels)
    sim_active = shuffled[shuffled.get("active") == True]
    sim_inactive = shuffled[shuffled.get("active") == False]
    sim_diff = (sim_active.get("top_album_sales").mean()
                - sim_inactive.get("top_album_sales").mean())
    sim_diffs = np.append(sim_diffs, sim_diff)

p_value = __(b)__ / 10000
```

(a): `np.random.permutation(bands.get("active"))`

(b): `np.count_nonzero(sim_diffs >= obs_diff)`

Solution: For (a), we shuffle the `"active"` labels to break any association with `"top_album_sales"` under the null. For (b), the p-value is the proportion of simulated differences at least as large as the observed difference, matching the one-sided alternative that active bands have higher sales.

c) (4 pts) Suppose Zeyu's p-value turns out to be 0.03. Which of the following is the correct conclusion of the test at the 0.05 significance level?

◯ We accept the null hypothesis; active and inactive bands have the same distribution of top album sales.

◯ We fail to reject the null hypothesis; we cannot determine whether the groups differ.

● We reject the null hypothesis; active bands have significantly higher top album sales.

◯ We reject the null hypothesis; active and inactive bands have different distributions of top album sales but we can't say which is greater.

Solution: Since $p = 0.03 < 0.05$, we reject the null. We never "accept" the null. Because the test was one-sided (`>= observed_diff`), rejecting it supports the specific alternative that active bands have higher sales.

d) (4 pts) In Zeyu's permutation test, suppose we change the code to instead shuffle the values in the `"top_album_sales"` column, and rerun the simulation. How would this affect the resulting null distribution?

◯ All 10,000 simulated values would be exactly the same as before.

● There would be slight variations in the null distribution but the histogram would look nearly identical.

◯ There would be substantial differences in the null distribution if we were to make this change. The histograms would be noticeably different.

◯ Without the full `bands` DataFrame available, we cannot predict how this change would affect the null distribution.

Solution: With fixed group sizes, shuffling the `"active"` labels and shuffling the `"top_album_sales"` values are statistically equivalent — both break the association between group and outcome. The two null distributions would look essentially the same, with only small differences from the randomness in the new run.

PID: _____

Question 8 (16 pts)

a) (6 pts) Sofia is doing research on bands in the UK. She writes the code below to estimate the mean number of members in UK bands using a bootstrapped 85% confidence interval. Fill in the blanks to complete the code.

```
uk = bands[bands.get("origin") == "UK"]
means = np.array([])

for i in np.arange(5000):
    resample = uk.sample(__(a)__, replace=True)
    means = np.append(means, resample.get("members").mean())

ci_low = np.percentile(means, __(b)__)
ci_high = np.percentile(means, __(c)__)
```

(a): `uk.shape[0]`

Solution: The bootstrap resample should have the same size as the original sample `uk`, so the correct choice is `uk.shape[0]`.

(b): `7.5`

(c): `92.5`

Solution: For a 85% confidence interval, we leave 15% in the tails, with 12.5% in each tail. Therefore the endpoints are the 7.5th and 82.5th percentiles.

b) (6 pts) Sofia decides to generalize her code in a function called `bootstrap_ci(df, confidence)` which takes as input the queried DataFrame and the confidence level, and returns a two-element list with the endpoints of the resulting confidence interval for the mean number of band members. For example, `bootstrap_ci(uk, 85)` corresponds to the procedure in part (a). Sofia defines some DataFrames and makes calls to the function as follows:

```
uk = bands[bands.get("origin") == "UK"]
uk_90 = bootstrap_ci(uk, 90)
uk_width = uk_90[1] - uk_90[0]
```

```
uk_sweden = bands[(bands.get("origin") == "UK") |
                  (bands.get("origin") == "Sweden")]
uk_sweden_90 = bootstrap_ci(uk_sweden, 90)
uk_sweden_width = uk_sweden_90[1] - uk_sweden_90[0]
```

(i) Suppose you know that the following expression evaluates to `True`.

```
np.std(uk.get("members")) > np.std(uk_sweden.get("members"))
```

What would you expect the relationship between `uk_width` and `uk_sweden_width` to be in this case?

◯ `uk_width < uk_sweden_width`

◯ `uk_width == uk_sweden_width`

● `uk_width > uk_sweden_width`

◯ We do not have enough information to determine the relationship.

Solution: Adding Sweden decreases the sample standard deviation and increases the sample size, both of which make the confidence interval narrower. Therefore `uk_sweden_width` should be smaller.

(ii) Suppose you know that the following expression evaluates to `True`.

```
np.std(uk.get("members")) < np.std(uk_sweden.get("members"))
```

What would you expect the relationship between `uk_width` and `uk_sweden_width` to be in this case?

◯ `uk_width < uk_sweden_width`

◯ `uk_width == uk_sweden_width`

◯ `uk_width > uk_sweden_width`

● We do not have enough information to determine the relationship.

Solution: Increasing the sample standard deviation tends to widen the confidence interval, while increasing the sample size tends to narrow it. Since these effects compete, we cannot determine which interval is wider from the given information alone.

c) (4 pts) Suppose `uk_90` comes out to be [3.8, 5.2]. Sofia's friend Elise says:

"There is a 90% probability that the true mean number of members in UK bands is between 3.8 and 5.2."

Which of the following is the most accurate criticism of Elise's statement?

◯ The statement should refer to individual bands, not the mean of all UK bands.

◯ The statement should say 95%, not 90%.

● The true mean is a fixed number and the confidence interval is also fixed, so the true mean either is or isn't in the interval $[3.8, 5.2]$.

◯ The interval is too wide to be useful. We need to lower the confidence level to get a more precise interval.

Solution: The correct answer is the second choice. Once an interval has been computed, it is a fixed interval and the parameter is a fixed value—there is no randomness left. It does not make sense to assign a probability to whether a fixed number falls in a fixed interval. The correct interpretation is that the *procedure* that produced the interval (repeated bootstrap sampling + taking the middle 90%) captures the true parameter approximately 90% of the time if we were to repeat the whole process many times with new samples from the same population.

PID: _____

Question 9 (17 pts)

Recall our assumption that `bands` is a **simple random sample of 400 bands** from the population of all bands.

a) (4 pts) You want to use the data in `bands` to create a 95% CLT-based confidence interval for the mean `"top_album_sales"` of all bands in the population. The data in `bands` was drawn **without replacement** but the CLT is a statement about samples drawn with replacement. What makes it reasonable to still use the CLT despite this discrepancy?

- ◯ The sample standard deviation is a good approximation of the population standard deviation.
- ◯ The statistic of interest is a sample mean.
- ◯ The population distribution of `"top_album_sales"` is likely normal.
- ● The sample is much smaller than the population.
- ◯ The CLT is less computational than bootstrapping, so we do not need to resample with replacement.

b) (4 pts) Your 95% CLT-based confidence interval comes out to be [20.0, 22.0]. Based on this information, estimate `np.std(bands.get("top_album_sales"))` to the **nearest integer**.

10.0

c) (4 pts) You now want to use the same data to generate another 95% CLT-based confidence interval for a new parameter: the proportion of bands with four or more members. What is the **maximum possible width** of this new confidence interval? Give your answer as an **exact decimal or simplified fraction**.

0.1 or $\frac{1}{10}$

d) (5 pts) You want to estimate several parameters using only the data in `bands`. Which of the statements are true? **Select all** that apply.

- [x] A CLT-based 95% confidence interval for the mean `"top_album_sales"` of all bands can be constructed if you don't have access to all of `bands` but you do know the mean, standard deviation, and length of the `"top_album_sales"` column.
- [x] To create a 95% confidence interval for the median `"top_album_sales"` of all bands, you would need to resample rows of `bands`; the CLT could not be used.
- [ ] A bootstrapped 95% confidence interval for the mean number of band `"members"` can be constructed if you don't have access to all of `bands` but you do know the mean, standard deviation, and length of the `"members"` column.
- [x] A 95% confidence interval for the proportion of `"active"` bands in the population can be calculated using either the CLT or bootstrapping.
- [ ] A 95% confidence interval for the largest value of `"members"` in the population can be calculated using at least one of the following methods: CLT, bootstrapping.
- [ ] None of the above.

PID: _____

Question 10 (12 pts)

We asked each DSC 10 tutor for their **top ten** favorite bands, and then counted how many had a `"genre"` of `"Rock"` and how many had a `"genre"` of `"Pop"`. Here are those counts for Michelle, Pranav, and Kate:

	Rock	Pop
Michelle	4	3
Pranav	2	4
Kate	x	y

For each tutor, we created a ten-row DataFrame containing just their top ten bands. Each was queried directly from `bands`, so each tutor's DataFrame has the same columns as `bands`, just with fewer rows.

a) (4 pts) Suppose we merge Michelle's ten-row DataFrame with Pranav's ten-row DataFrame, on the column `"genre"`. What are the minimum and maximum possible number of rows in the result?

min: $4 * 2 + 3 * 4 = 20$ max: $4 * 2 + 3 * 4 + 3 * 4 = 32$

b) (4 pts) Now, suppose we again merge Michelle and Pranav's DataFrame, but this time on the column `"band"`. What are the minimum and maximum possible number of rows in the result?

min: 0 max: $2 + 3 + 3 = 8$

c) (4 pts) Finally, suppose we merge Michelle's ten-row DataFrame with Kate's ten-row DataFrame, on the column `"genre"`. What are the minimum and maximum possible number of rows in the result? Give your answers as **unsimplified algebraic expressions** involving x and y.

min: $4x + 3y$ max: $4x+3y+3(10-x-y) = x+30$

Question 11 (8 pts)

Ray recently got into K-Pop and he wants to estimate the median number of albums among all K-Pop bands. Consider what would happen if he could somehow **repeat the following process 500 times**:

1. Take a simple random sample of K-Pop bands from the population.
2. Bootstrap resample from that sample 10,000 times, computing the median number of albums each time.
3. Construct an 80% confidence interval from the bootstrapped medians.

a) (4 pts) **How many** of Ray's confidence intervals would we expect to contain the true median number of albums among all K-Pop bands? Give your answer as an **integer**.

400

Solution: An 80% confidence interval procedure should capture the true parameter approximately 80% of the time. Since Ray repeats the process 500 times, we expect $0.8 \times 500 = 400$ intervals to contain the true median.

b) (4 pts) Which of the following is the most significant barrier to Ray actually completing this procedure?

◯ The computation time required to bootstrap so many times.

◯ The fact that his parameter of interest is a median and not a mean so the CLT does not apply.

● The logistics of obtaining new random samples from the population.

◯ The fact that he wants an 80% confidence interval instead of a standard 95% confidence interval which is easier to compute.

Solution: The biggest challenge is obtaining many new random samples from the population. Bootstrapping itself is computationally feasible (for any confidence level), and bootstrapped confidence intervals can still be constructed for medians.

PID: _____

Question 12 (19 pts)

If a band has released a large number of studio albums, is their best-selling album more likely to be successful? To answer this question, you decide to use linear regression. Suppose:

- `"albums"` has mean 7 and standard deviation 2,
- `"top_album_sales"` has mean 21 and standard deviation S, and
- the correlation coefficient between `"albums"` and `"top_album_sales"` is 0.6.

a) (4 pts) The band *Coldplay* has released 10 albums. How many standard deviations above the mean is *Coldplay*'s **predicted** top album sale, using our regression line? Give your answer as an **exact decimal or simplified fraction**.

$\frac{9}{10} = 0.9$

b) (4 pts) If a band has 0 albums, its top album sale should *also* be 0. Suppose our regression line predicts that a band with 0 albums will have a top album sale of exactly 0. What is the standard deviation S of `"top_album_sales"`? Give your answer as an **exact decimal or simplified fraction**.

10

c) (7 pts) Imagine for this question only that we've just discovered a mistake in our data: each studio album also has a companion live album, so the true album count for each band is **double** what's recorded. To address this, you multiply all values in the `"albums"` column by 2 and re-fit the regression line predicting `"top_album_sales"` from the new (doubled) `"albums"` variable. Which of the following statements about the **new** regression line are true? **Select all** that apply.

- ☐ The new slope will be exactly double the old slope.
- ■ The new slope will be exactly half the old slope.
- ☐ The new intercept will be less than the old intercept.
- ☐ The new intercept will be greater than the old intercept.
- ■ The correlation coefficient r will stay the same.
- ■ The RMSE of the new line is the same as the RMSE of the original line.
- ☐ The predicted value for a band with a new (after doubling) album count of 30 equals **twice** the original line's prediction for a band with 15 albums.
- ☐ None of the above.

d) (4 pts) The DSC 10 tutors split in half and form two bands, *The Null Hypotheticals* and *P-Value and the Rejects*. They track the number of albums each band produces, as well as the sales of each band's top albums, storing this information in a two-row DataFrame called `tutors`. Instead of using the data in `bands`, they create a regression line predicting top album sales from the number of albums using **only** the data in `tutors`.

(i) **True or False:** The regression line formed using the data in `bands` **cannot** have the same RMSE as the regression line formed using only the bands in `tutors`.

◯ True ◯ False

(ii) **True or False:** If both the `tutors` and `bands` regression lines are exactly the same line, then the `bands` DataFrame **must** include bands with the same number of albums and top album sales as the two bands in the `tutors` DataFrame.

◯ True ◯ False

PID: _____

Question 13 (8 pts)

Below is one line of code and the DataFrame it produces.

```
bands.get(["active", "genre", "members"]).groupby(["active", "genre"]).count()
```

		members
active	**genre**	
False	**Alternative**	30
	Metal	90
	Pop	90
	Rock	60
	Soul	30
True	**Alternative**	10
	Hip-Hop	20
	Metal	20
	Pop	20
	Rock	30

a) (4 pts) Using this DataFrame, calculate the total variation distance (TVD) between the `"genre"` distribution of active bands and the `"genre"` distribution of inactive bands. Give your answer as an **exact decimal or simplified fraction**.

$$0.3 = \frac{3}{10}$$

b) (4 pts) Suppose you randomly draw one band uniformly from `bands`. You are told that this band is **not** a rock band. What is the conditional probability that the band is active? Give your answer as an **exact decimal or simplified fraction**.

$$\frac{70}{310} = \frac{7}{31}$$

Feel free to draw us a picture or tell us about your fondest memory from DSC 10 this quarter.

Solution:

Before turning in your exam, please make sure that your PID is on every page.

Congratulations on finishing DSC 10!