
Midterm Exam - DSC 10, Fall 2026

Full Name:

TSN:

- Exam: ☐ 9AM HSS 1330 ☐ 10AM HSS 1330
 ☐ 10AM WLH 2005 ☐ 11AM WLH 2005

Instructions:

- This exam consists of 10 questions worth a total of 100 points.
- Write your TSN in the top right corner of each page in the space provided. Completing this is **worth 2 points**.
- 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.
- 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.
- You may use one handwritten sheet of notes. No calculators and no computers.

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

Signature:

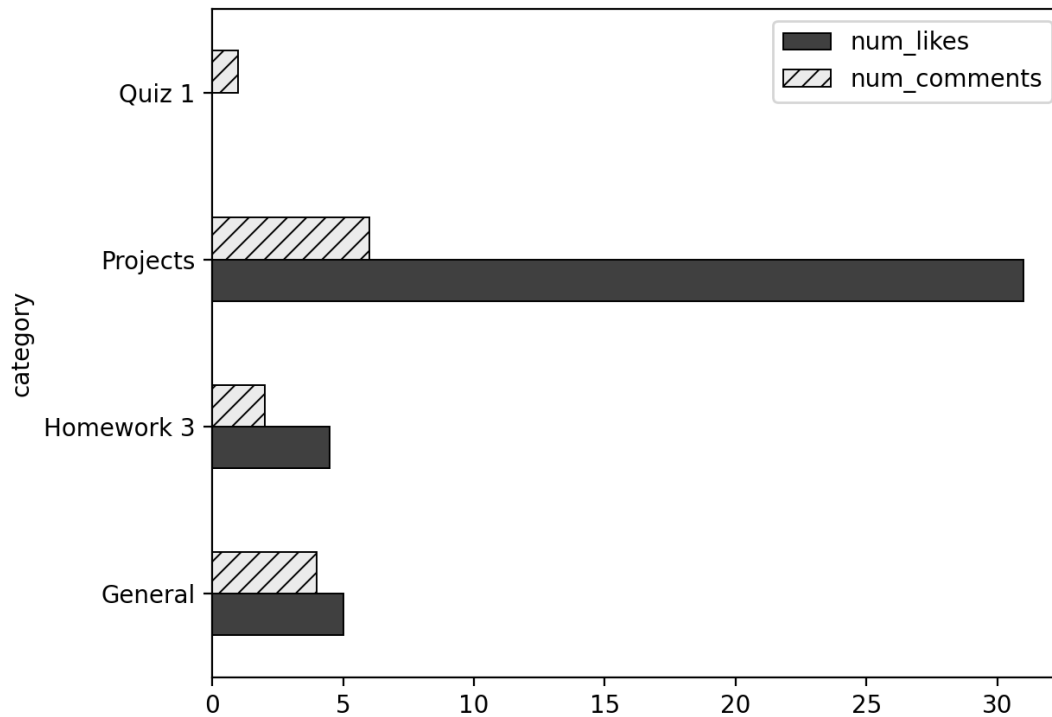
Version A

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

Rip off the last page of this exam packet and read the data description before proceeding.

Question 1 (7 pts)

Write **one line of code** to create a horizontal bar chart showing both the average "num_likes" and the average "num_comments" for each "category". On a DataFrame of just the 5 rows shown in the data description, it would look like the image below.



Question 2 (11 pts)

After a particularly difficult DSC 10 quiz, students flood Campuswire with questions. The course staff scramble to respond, but each new question has only a 40% chance of receiving a response within ten minutes, independently of other questions.

For all parts, give your answer as a mathematical expression. You do not need to simplify.

- a) (2 pts) What is the probability that the first two questions do not receive responses within ten minutes, but the third question does?

- b) (2 pts) What is the probability that exactly two of the first three questions receive a response within ten minutes?

- c) (2 pts) What is the probability that none of the first three questions receive responses within ten minutes?

- d) (2 pts) Given that exactly one of the first three questions receives a response within ten minutes, what is the probability that it is the third question?

- e) (3 pts) Let a , b , c , and d , represent the correct answers to parts (a), (b), (c), and (d), respectively.

Given that at least one of the first three questions receives a response within ten minutes, what is the probability that the third question is the only one to receive a response within ten minutes?

☐ $\frac{d}{a+b}$ ☐ $\frac{a}{1-c}$ ☐ $\frac{d}{1-c}$ ☐ $\frac{a}{a+b}$

Question 3 (10 pts)

Define a Series containing the first five values in the "num_comments" column:

```
five = posts.take(np.arange(5)).get("num_comments")
```

Determine the value of each of the following expressions.

a) (3 pts) `10 + five.sum()`

b) (3 pts) `(10 + five).sum()`

c) (4 pts) `((five + (five >= 5).sum()) >= 5).sum()`

Question 4 (6 pts)

- a) (3 pts) Suppose we want the function `convert_to_month` to take a string in the format of "YYYY-MM-DD" and return the month as an integer. Fill in the blanks in the code below.

```
def convert_to_month(x):  
    return __ (a) __ (x.__ (b) __ [1])
```

(a):

(b):

- b) (3 pts) Next, we want `post_months` to evaluate to a Series with the month (as an integer) that each post was made on Campuswire, in the same order as the rows appear in the DataFrame. Fill in the blanks in the code below.

```
post_months = posts.__ (c) __.__ (d) __
```

(c):

(d):

Question 5 (12 pts)

- a) (5 pts) Which of the following evaluates to the title of the **question** with the third-highest number of views? **Select all that apply.**

- ☐ `posts.sort_values("num_views", ascending=False).get("title").iloc[2]`
- ☐ `posts[posts.get("is_question") == True].get(["title", "num_views"]).set_index("title").sort_values("num_views", ascending=False).index[2]`
- ☐ `posts[posts.get("is_question")].sort_values("num_views", ascending=False).get("title").loc[2]`
- ☐ `posts[posts.get("is_question") == True].sort_values("num_views", ascending=False).get("title").iloc[2]`
- ☐ `posts[posts.get("is_question")].sort_values("num_views", ascending=False).get("title").iloc[-3]`
- ☐ None of the above.

- b) (7 pts) Which of the following evaluates to the number of views of the post with "post_id" 104? **Select all that apply.**

- ☐ `posts.get("num_views").loc[3]`
- ☐ `posts.get("num_views").iloc[3]`
- ☐ `posts.set_index("post_id").get("num_views").loc[104]`
- ☐ `posts.set_index("post_id").get("num_views").iloc[104]`
- ☐ `posts.sort_values("num_views").get("num_views").iloc[3]`
- ☐ `posts.sort_values("num_views").get("num_views").loc[3]`
- ☐ `posts[posts.get("post_id") == 104].get("num_views").iloc[0]`
- ☐ None of the above.

Question 6 (10 pts)

Suppose we divide student-authored posts into four categories as follows:

1. public, resolved student-authored posts
2. public, unresolved student-authored posts
3. private, resolved student-authored posts
4. private, unresolved student-authored posts

Fill in the blanks in the code below to create a DataFrame called `max_likes` with four rows and a single column containing the **largest number of likes on a single post** in each of these four categories. The rows can be in any order (not necessarily matching the order given above).

```
student_posts = posts[__(a)___]  
max_likes = student_posts.get(__(b)___).groupby(__(c)___).__(d)___
```

(a):

(b):

(c):

(d):

Question 7 (7 pts)

Vanessa authored five of the posts included in `posts`. We define a DataFrame called `vanessa` containing only Vanessa's posts, and keeping only certain columns. This DataFrame is shown in full below.

	<code>post_id</code>	<code>num_likes</code>	<code>num_comments</code>	<code>num_views</code>
0	201	2	0	2
1	312	3	0	3
2	273	0	0	0
3	188	1	2	2
4	241	0	3	3

What does the following expression evaluate to?

```
vanessa.merge(vanessa, left_on="num_likes", right_on="num_comments").shape[0]
```

Question 8 (10 pts)

Suppose the array `assignments` contains the names of all assignments in a course. For example, `assignments[0]` might be "Homework 1".

Suppose also that the array `due_dates` contains the deadlines for all assignments in the course, in the same order as in `assignments`. In particular, both arrays have the same length. The entries of `due_dates` are strings formatted like "YYYY-MM-DD". For example, `due_dates[0]` might be "2026-01-12".

Our goal in this problem is to create a string called `reminders` that contains sentences describing when each assignment is due, with one space after each sentence. For example, in a class with only three assignments, the output may look like this:

```
"Homework 1 is due on 2026-01-12. Homework 2 is due on 2026-01-17. Midterm
Project is due on 2026-01-25. "
```

Complete the for-loop below so that `reminders` is a string formatted like the above example, using all the assignments and deadlines in `assignments` and `due_dates`.

```
reminders = __(a)__
for i in __(b)__:
    reminders = __(c)__
```

(a):

(b):

(c):

Question 9 (13 pts)

The course tutors are constantly competing to give the best, most thoughtful answers on Campuswire. To determine who is the nicest tutor, the course staff run a weekly **Kindness Check**: they select **4 posts at random, without replacement**, from **posts** and read every tutor reply on them. Since only questions receive replies, Jeffrey is hoping that the Kindness Check lands on more questions than notes. That way, he has more chances to show everyone just how nice he can be.

Fill in the blanks below so that `prob_more_questions` is an estimate of the probability that a Kindness Check contains **more questions than notes**.

```
repetitions = 10000
count = 0
for i in np.arange(repetitions):
    check = np.random.choice(__(a)__)
    if ____(b)__:
        count = count + 1
prob_more_questions = count / repetitions
```

a) (4 pts) Which of the following correctly fills in blank (a)? **Select one.**

- ☐ `posts, 4, replace=True`
- ☐ `posts, 4, replace=False`
- ☐ `posts.get("is_question"), 4, replace=True`
- ☐ `posts.get("is_question"), 4, replace=False`
- ☐ `np.array([True, False]), 4, replace=True`
- ☐ `np.array([True, False]), 4, replace=False`

b) (5 pts) Which of the following **could** go in blank (b)? **Select all that apply.**

- ☐ `np.count_nonzero(check) >= 3`
- ☐ `np.count_nonzero(check) >= 2`
- ☐ `np.count_nonzero(check) > 0.5`
- ☐ `check.sum() > (4 - (check).sum())`
- ☐ `check.mean() > 0.5`
- ☐ None of the above.

- c) (4 pts) Jeffrey copies the correct simulation into his own notebook, but in his excitement, he accidentally moves the line `count = 0` to the inside of the `for`-loop:

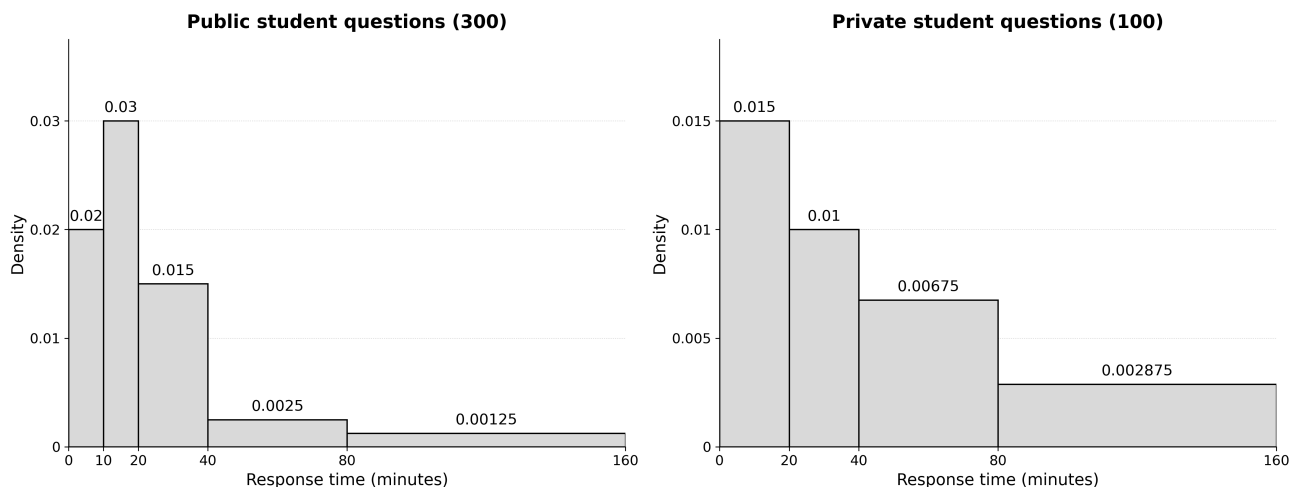
```
repetitions = 10000
for i in np.arange(repetitions):
    count = 0
    check = np.random.choice(__(a)__)
    if __(b)__:
        count = count + 1
prob_more_questions = count / repetitions
```

Assuming that blanks (a) and (b) are filled in correctly, which of the following **could** `prob_more_questions` evaluate to after Jeffrey's code finishes running **Select all that apply**.

- ☐ 0.0
- ☐ 0.0001
- ☐ 0.5
- ☐ 1.0
- ☐ None of the above.

Question 10 (12 pts)

The density histograms below show response times for **300 public** and **100 private** student questions. All values are exact.



- a) (4 pts) How many **private** questions took **at least 40 minutes** to receive a response? Give your answer as an **integer**.

- b) (4 pts) Suppose we make a single density histogram for all 400 questions, with one bin being **[0, 10)**. What would the height of this bar be?

- ☐ $\frac{3}{200} = 0.015$
☐ $\frac{7}{400} = 0.0175$
☐ $\frac{1}{50} = 0.02$
- ☐ $\frac{11}{400} = 0.0275$
☐ $\frac{7}{200} = 0.035$
☐ Not enough information.

- c) (4 pts) Suppose we make a single density histogram for all 400 questions, with one bin being **[20, 40)**. What would the height of this bar be?

- ☐ $\frac{1}{80} = 0.0125$
☐ $\frac{11}{800} = 0.01375$
☐ $\frac{3}{200} = 0.015$
- ☐ $\frac{1}{40} = 0.025$
☐ $\frac{11}{40} = 0.275$
☐ Not enough information.



Campuswire is an online discussion forum used by many college courses, including DSC 10. Students and instructional staff use it to ask questions, share announcements, and discuss course material.

The DataFrame `posts` has one row for every post made on a course's Campuswire forum. The columns are:

- `"post_id"` (int): A unique ID number for the post.
- `"title"` (str): The title of the post.
- `"date"` (str): The date the post was made, formatted as "YYYY-MM-DD", for example "2026-02-03".
- `"author_role"` (str): The role of the author, either `"student"` or `"staff"`.
- `"category"` (str): The category the post was filed under, such as `"General"`, `"Quiz 1"`, `"Projects"`, or `"Homework 3"`.
- `"is_private"` (bool): `True` if the post is visible only to the author and course staff, and `False` if it is visible to the whole class.
- `"is_question"` (bool): `True` if the post is a question, and `False` if it is a note (such as an announcement or reminder).
- `"is_resolved"` (bool): `True` if the post has been marked as resolved, and `False` otherwise.
- `"num_views"` (int): The number of times the post has been viewed.
- `"num_likes"` (int): The number of likes the post has received.
- `"num_comments"` (int): The number of comments on the post.

The first few rows of `posts` are shown below, though `posts` has many more rows than pictured.

	post_id	title	date	author_role	category	is_private	is_question	is_resolved	num_views	num_likes	num_comments
0	101	Office hours this week?	2026-02-03	student	General	False	True	True	87	5	4
1	102	Grade discrepancy on Quiz 1	2026-02-05	student	Quiz 1	True	True	True	3	0	1
2	103	Reminder: Project 2 due Friday	2026-02-06	staff	Projects	False	False	False	214	31	6
3	104	How do I merge two DataFrames?	2026-02-07	student	HW 3	False	True	True	56	9	3
4	105	Extension request for medical reason	2026-02-09	student	HW 3	True	False	True	2	0	1

Throughout this exam, we will refer to `posts` repeatedly. Assume that we have already run `import baby pandas as bpd` and `import numpy as np`.