Section: _____ Name:_____

Read the following directions carefully. DO NOT turn to the next page until the exam has started.

Write your name and section number at the top right of this page:

Class	**Section Number**
Miranda 8:50	001
Sahifa 12:05	002
Miranda 9:55	004

As you complete the exam, write your initials at the top right of each other page.

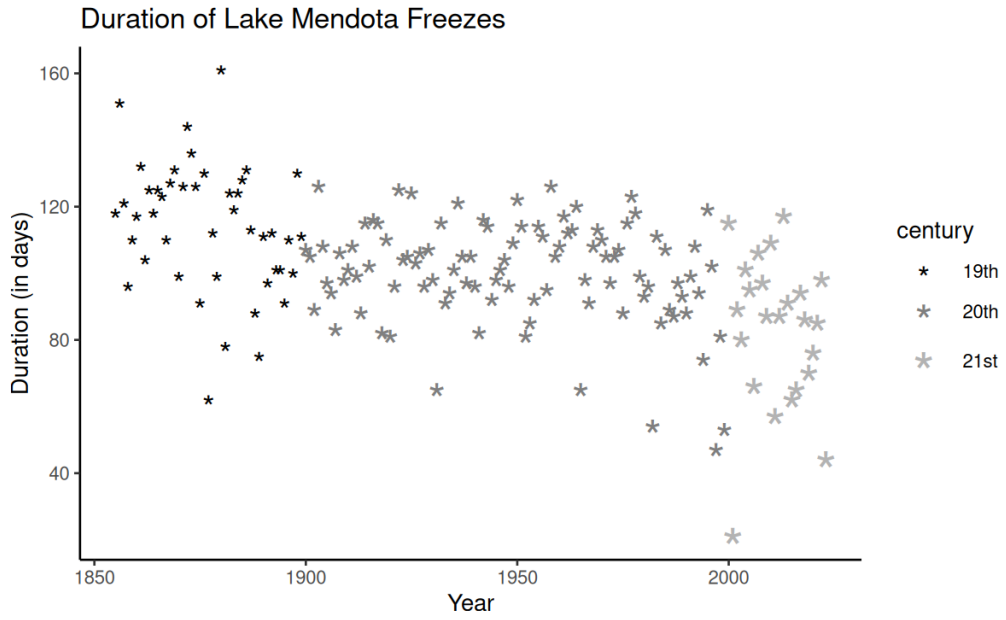
When the exam start time is called, you may turn the page and begin your exam.
If you need more room, there is a blank page at the end of the exam, or we can give you some scratch paper.

Some multiple choice questions are "Select ONE" while others are "Select ALL that apply". Pay attention to the question type and only mark one option if it says "Select ONE". Fill in the circles completely.

If you finish early, you can hand your exam to your instructor or TA and leave early.

Otherwise, stop writing and hand your exam to your instructor or TA when the exam stop time is called.

1. The scatter plot below was created with the `mendota` data from lecture. It shows the duration of Lake Mendota freezes over time, and also indicates the century each point belongs to.



(a) (5 points) For each aesthetic used in the plot, indicate whether it is a variable or constant aesthetic, or if it is not used in the plot. **Select ONE per aesthetic**.

- `x`: ● Variable ○ Constant
- `y`: ● Variable ○ Constant
- `color`: ● Variable ○ Constant
- `shape`: ○ Variable ● Constant
- `size`: ● Variable ○ Constant

(b) (3 points) Which two aesthetics correspond to the same variable in our data? What information is being conveyed by those two aesthetics?

`color` and `size` are both being used to refer to the century each data point belongs to.

(c) (3 points) Which line of code below create a red line parallel to the x-axis representing the mean duration? **Select ONE**.

- ● `geom_hline(yintercept = mean(mendota$duration), color = "red")`
- ○ `geom_vline(yintercept = mean(duration), color = "red")`
- ○ `geom_line(yintercept = mean(mendota$duration), color = "red")`
- ○ `geom_hline(yintercept = mean(duration), color = "red")`
- ○ `geom_vline(xintercept = mean(mendota$duration), color = red)`
- ○ `geom_line(xintercept = mean(duration), color = red)`

2. The dataframe `students` gives the name, gender, grade, and performance of 9 students.

```
> students
# A tibble: 9 × 4
  name      grade gender performance
  <chr>     <dbl> <chr>  <chr>
1 John       90   male   Excelled
2 Arya       87.3 female Excelled
3 Fred       NA   male   NA
4 Angie      67   female Performed well
5 Riley      79   female Performed well
6 Elizabeth  95   female Excelled
7 Farhan     92   male   Excelled
8 NA         20.5 NA     Failed
9 Henry      39.7 male   Failed
```

(a) (3 points) Which command keeps all rows of `students` where the variable `grade` is not missing, the variable `gender` is not "female", and the variable `perfomance` is not "Failed"? **Select ONE**.

- ○ `students %>% filter(grade != NA, gender != "female", performance != "Failed")`
- ○ `students %>% filter(grade != NA, gender == "male", performance == !"Failed")`
- ○ `students %>% filter(drop_na(grade), gender == "male", performance == !"Failed")`
- ● `students %>% filter(!is.na(grade), gender != "female", performance != "Failed")`

(b) (3 points) After performing the operation in (a), how many rows are printed?

2 rows (rows 1 and 7)

(c) (3 points) After performing the following operation, how many rows are printed?

```
students %>% drop_na()
```

7 rows (all rows except 3 and 8 have no missing values)

3. The `cabbages` dataset gives the cultivar type, date planted, head weight (in kg) and vitamin C content of 60 cabbages.

```
> glimpse(cabbages)
Rows: 60
Columns: 4
$ Cult   <fct> c39, c39, c39, c39, c39, c39, c39, c39, c39, c39, c39, c39, c39, c39,  ...
$ Date   <fct> d16, d16, d16, d16, d16, d16, d16, d16, d16, d16, d20, d20, d20, d20,  ...
$ HeadWt <dbl> 2.5, 2.2, 3.1, 4.3, 2.5, 4.3, 3.8, 4.3, 1.7, 3.1, 3.0, 2.8, 2.8, 2.7,  ...
$ VitC   <int> 51, 55, 45, 42, 53, 50, 50, 52, 56, 49, 65, 52, 41, 51, 41, 45, 51, 45,  ...
```

Use these function names to fill in the blanks below.

- `relocate`
- `select`
- `group_by`
- `summarize`
- `max`
- `slice_max`
- `n`
- `count`
- `mean`

(a) (3 points) Fill in the blanks to remove the `VitC` column, then print the largest cabbage for each of the cultivar types.

```
cabbages %>%
          __select__(-VitC) %>%
          _group_by_(Cult) %>%
          _slice_max_(HeadWt)
```

(b) (4 points) Fill in the blanks to count the number of cabbages for each cultivar-date combination, and calculate the average vitamin C content for each cultivar-date combination.

```
cabbages %>%
          _group_by_(Cult, Date) %>%
          _summarize_(counts = ___n___(),
                      avg_vitc = __mean__(VitC))
```

4. (5 points) Researchers perform two experiments. In the first experiment, researchers record the participants' names and 8 other variables, which are stored in `df1`. In the second experiment, researchers record participants' names and 6 other variables, which are stored in `df2`. The name column is the only variable that appears in both dataframes.

   `df1` has 15 rows and `df2` has 25 rows (one row per participant). 5 people participated in both experiments, so they appear in both dataframes.

   Give the number of rows and the number of columns that result from the following join operations.

Join	Number of rows	Number of columns
`inner_join(df1, df2)`	5	15
`left_join(df1, df2)`	15	15
`right_join(df1, df2)`	25	15
`full_join(df1, df2)`	35	15
`anti_join(df1, df2)`	10	9

5. (5 points) In the dataframe `cakes`, a bakery records the flavor and occasion for each cake order they receive each week. For example, this week they received 4 orders for chocolate birthday cakes.

```
> cakes
# A tibble: 8 × 3
  flavor    occasion num_ordered
  <chr>     <chr>          <dbl>
1 chocolate birthday           4
2 vanilla   birthday           3
3 vanilla   wedding            1
4 redvelvet wedding            1
5 chocolate wedding            2
6 carrot    birthday           2
7 vanilla   other              4
8 redvelvet other              2
```

The bakery wants to structure the data such that there is a separate column for each occasion type, rather than a single "occasion" column.

Using the grid below, re-create the dataframe that would result from the call to `pivot_wider`. Use the top shaded row for the column names and the un-shaded cells for the data values. Be sure to account for any missing data.

```
pivot_wider(cakes, names_from = "occasion", values_from = "num_ordered")
```

	flavor	birthday	wedding	other
1	chocolate	4	2	NA
2	vanilla	3	1	4
3	redvelvet	NA	1	2
4	carrot	2	NA	NA