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**
Sahifa 8:50	001
Miranda 9:55	002
Sahifa 2:25	003
Sahifa 3:30	004
Miranda 8:50	006

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.

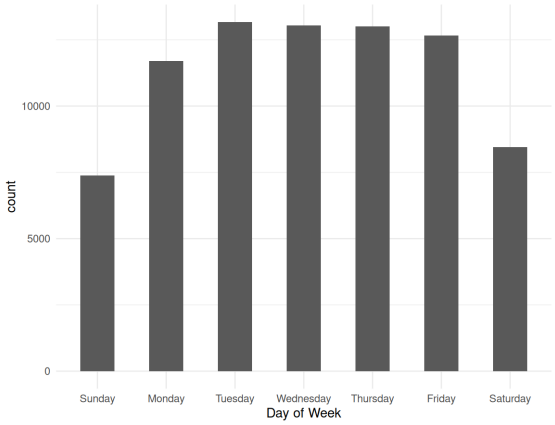
Some multiple choice questions are "Select ONE" while others are "Select ALL that apply". Pay attention to the question type and only mark one option if it says "Select ONE". Fill in the circles completely.

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 dataframe `births` (with 79,421 rows and 1 column) records the day of the week for 79,421 births. The code below (some details omitted) creates a graph to count the number of births for each day.

```
births %>%
  group_by(day_of_week) %>%
  summarize(count = n()) %>%
  ggplot(aes(x = day_of_week, y = count)) +
  geom_col()
```



(a) Explain why we need to create a new variable called `count`. Identify a different geom that could produce the same plot without needing to tidy the data or manually create this new variable.

> This is due to the fact that `geom_col` is a two variable bar graph and needs to be provided with both x and y variable. `geom_bar` can be used instead as it automatically creates the y variable for the count.

(b) You want to make the following changes to the plot: Add labels at the top of each bar displaying the count for that day, and add a line showing the count of the tallest bar. Which geoms do you need to add to your plot? **Select ALL that apply**.

- ○ `geom_point`
- ● `geom_text`
- ○ `geom_hline`
- ○ `geom_vline`

(c) You want to color the bar areas pastel blue. Which aesthetics do you need to add to your plot? **Select ALL that apply**.

- ○ `color` as a variable aesthetic
- ● `color` as a constant aesthetic
- ○ `fill` as a variable aesthetic
- ● `fill` as a constant aesthetic

2. Let `students` be a dataset that gives the `name` of the students, their `HW`, `Quiz` and `Final_Exam` grade in a class.

A tibble: 6 × 4

name <chr>	HW <dbl>	Quiz <dbl>	Final_Exam <dbl>
John	90.0	91	87.5
Arya	87.3	60	59.2
Fred	*NA*	75	82.7
Riley	92.0	88	73.0
Tyler	39.7	36	20.6
Dev	99.0	89	89.0

6 rows

Two column need to be added to `students` dataset that would contain the information about the final percentage and grade of the student in class. Fill in the blanks such that the output looks like the output above and has columns named `final_percentage` and `Grade`.

A tibble: 5 × 6

name <chr>	HW <dbl>	Quiz <dbl>	Final_Exam <dbl>	final_percentage <dbl>	Grade <chr>
John	90.0	91	87.5	89.05	A-
Arya	87.3	60	59.2	65.06	B-
Riley	92.0	88	73.0	81.30	A-
Tyler	39.7	36	20.6	29.04	F
Dev	99.0	89	89.0	91.00	A

5 rows

```
students %>%
  drop_na() %>%
  mutate(final_percentage = 0.2*HW+0.3*Quiz+0.5*Final_Exam,
  Grade = case_when(
  final_percentage >= 90 ~ "A",
  final_percentage < 90 & final_percentage >= 80 ~ "A-",
  final_percentage < 80 & final_percentage >= 70 ~ "B",
  final_percentage < 70 & final_percentage >= 60 ~ "B-",
  final_percentage < 60 & final_percentage >= 50 ~ "C",
  final_percentage < 50 & final_percentage >= 40 ~ "C-",
  final_percentage < 40 ~ "F",
))
```

3. In the dataframe, `students` used in question 2, the instructor of the course wants each assignment name to be placed beside the student name and their score beside it in the next column. That is the output should look like the sample output below.

A tibble: 15 × 3

name <chr>	Performance <chr>	value <dbl>
John	HW	90.0
John	Quiz	91.0
John	Final_Exam	87.5
Arya	HW	87.3
Arya	Quiz	60.0
Arya	Final_Exam	59.2

Fill in the blanks of the codes below such that you obtain the above output.

```
students %>%
   pivot_longer(-name, names_to = "Performance")
```

4. A data set `bm` has Boston Marathon data from the year 2010 with a row for each female runner who completed the race and variables `Age`, `Age_Range`, and `Time`. Identify code that calculates the mean time for all female runners between the ages of 35 and 39. Note: `Age_Range` equals "35 – 39" when `Age` is in this range.

   An answer is considered correct if it calculates the desired mean in addition to calculating additional summary statistics.

   Circle correct answers and cross out incorrect answers.

   ◯ `bm %>% filter(Age_Range == "35 – 39") %>% summarize(mean = mean(Time))`

   ◯ `bm %>% group_by (Age_Range) %>% summarize(mean = mean(Time))`

   ◯ `bm %>% mutate(mean = mean(Time)) %>% filter(between(Age, 35, 39))`

   ◯ `bm %>% select(Age_Range == "35 – 39") %>% summarize(mean = mean(Time))`

5. (a) A data set `grocery_price` has variables named `item`, `type`, and `price`. A data set named `grocery_list` has variables named `item` and `n`. The values in the columns `item` match if the same item is part of both data sets. Some items in `grocery_price` may not by in `grocery_list` and some items in `grocery_list` may not by in `grocery_price`. Which description matches the contents of `grocery` after executing the following code? No items are repeated within either data set. **Circle one answer.**

```
  grocery = grocery_list %>%
full_join(grocery_price, by = "item")
```

- ○ A data frame with one row for each item in both data sets and columns `item` and `n` only.
- ○ A data frame with one row for each item in both data sets and columns `item`, `n`, `type` and `price`.
- ○ A data frame with one row for each item in `grocery_list` and columns `item`, `n`, `type` and `price`.
- ● A data frame with one row for each item in either data set, columns `item`, `n`, `price` and `type`, and the value `NA` in columns `n`, `type`, and `price` in rows where this information was missing.

(b) Fill in the blacks to obtain the 3 cheapest `item` in the fruit and vegetable category, in the `grocery` dataset.

```
  cheapest = grocery %>%
group_by(type) %>%
slice_min(price, n=3)
```

6. We start by a dataset called `grocery1` which looks like the output displayed below:

A tibble: 8 × 5

item <chr>	n <dbl>	price <dbl>	type <chr>	total_cost <dbl>
mango	1	0.77	fruit	0.8085
avocado	4	0.77	vegetable	3.2340
pineapple	1	2.99	fruit	3.1395
onion	5	1.27	vegetable	6.6675
sweet potato	*NA*	1.35	vegetable	*NA*
banana	8	0.27	fruit	2.2680
apple	6	0.85	fruit	5.3550
strawberry	*NA*	2.99	fruit	*NA*

8 rows

(a) Write out the output you get from the following code:

```
grocery1 %>%
  select(type, total_cost) %>%
  group_by(type) %>%
  summarize(max = max(total_cost, na.rm = T), min = min(total_cost, na.rm = T))
```

>

A tibble: 2 × 3

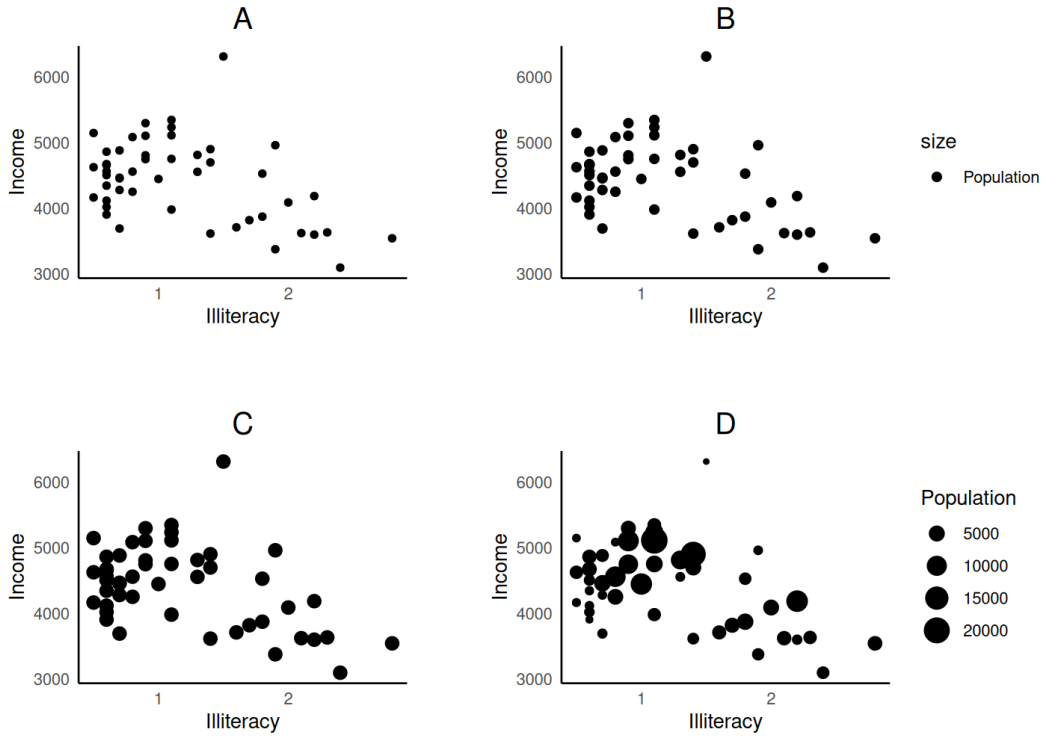
type <chr>	max <dbl>	min <dbl>
fruit	5.3550	0.8085
vegetable	6.6675	3.2340

2 rows

(b) What dplyr function could be an alternative to the `na.rm = T` argument in `max()` and `min()`.

> We can use `drop_na()` or `filter(!is.na(n), !is.na(total_cost))`

7. Below are four different `geom_point()` plots of illiteracy rate versus income for US states. Some plots also show the states' population.



Match the four plots (A, B, C, D) to the four different `geom_point()` calls below.

A `geom_point(aes(x = Illiteracy, y = Income))`

D `geom_point(aes(x = Illiteracy, y = Income, size = Population))`

B `geom_point(aes(x = Illiteracy, y = Income, size = "Population"))`

C `geom_point(aes(x = Illiteracy, y = Income), size = 3)`