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.

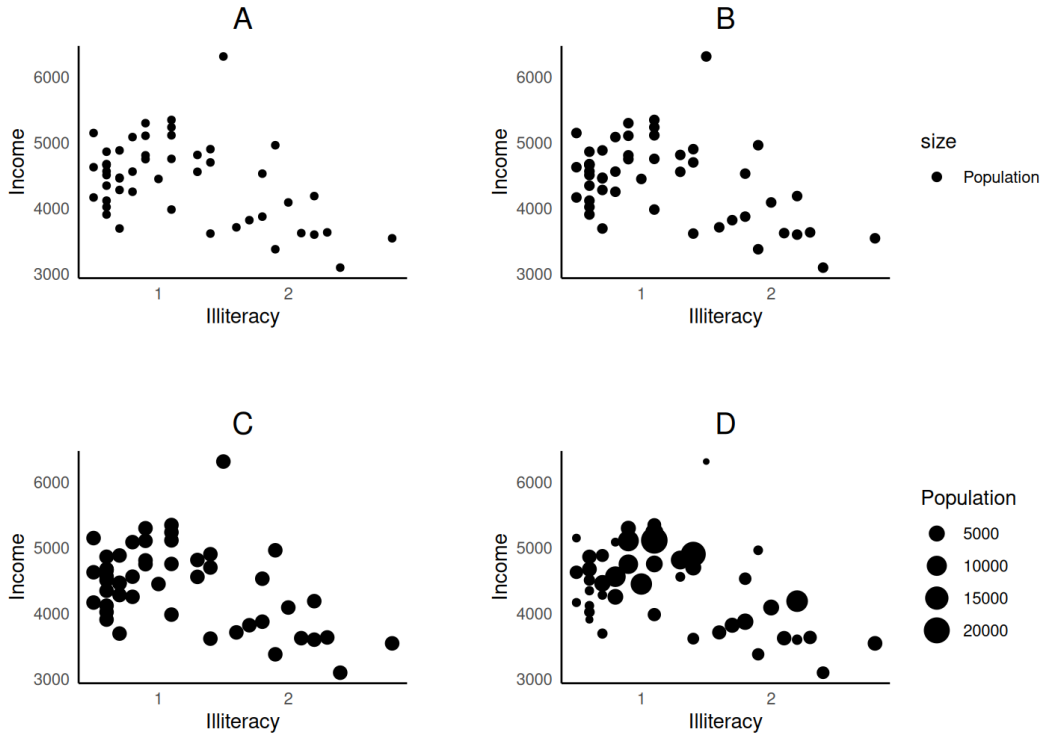
When the exam start time is called, you may turn the page and begin your exam.
If you need more room, there is a blank page at the end of the exam, or we can give you some scratch paper.

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



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

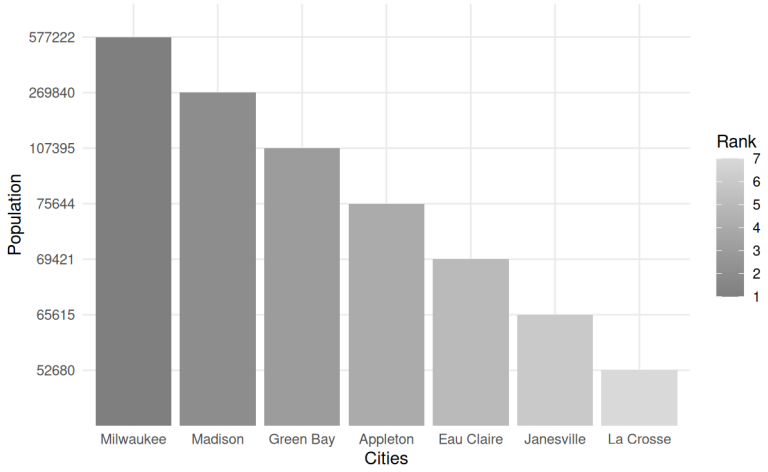
A `geom_point(aes(x = Illiteracy, y = Income))`

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)`

2. The `cities_WI` dataset has 7 rows and 3 columns, containing the name, population, and population rank 1-7 of Wisconsin's seven largest cities. We want to re-create the following bar plot:



(a) (3 points) Given that we know the population of the 7 cities, should the plot be created with `geom_bar()` or `geom_col()`? Briefly explain your answer.

The plot must have been made with `geom_col()`. We use `geom_col()` when we have both an x and y column and can therefore set the height of the bars (population in this case) manually. With `geom_bar()` we only provide one variable and the heights of the bars are calculated automatically based on counts of observations.

(b) (3 points) How should we write the call to `ggplot()` to make it so the bar interiors are colored based on the cities' `Rank`? The other plot aesthetics are intentionally left out. **Select ONE**.

- ● `ggplot(aes(..., fill = Rank))`
- ○ `ggplot(aes(..., col = Rank))`
- ○ `ggplot(aes(...), fill = Rank)`
- ○ `ggplot(aes(...), col = Rank)`

3. The `birthwt` dataset records birth weight and risk factors for 189 pregnancies. We are interested in columns `low` (0 or 1, indicating low birth weight), `smoke` (0 or 1, indicating mother's smoking status), and `bwt` (birth weight of newborn in grams).

```
> birthwt %>% select(low, smoke, bwt) %>% glimpse()
Rows: 189
Columns: 3
$ low   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ smoke <int> 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, ...
$ bwt   <int> 2523, 2551, 2557, 2594, 2600, 2622, 2637, 2637, 2663, 2665, 2722, ...
```

Use these function names as well as the column names to fill in the blanks below. You might not need to use every function/column, and you might need to use some more than once.

- `select`
- `filter`
- `group_by`
- `summarize`
- `count`
- `min`
- `slice_min`
- `median`
- `mean`

(a) (3 points) Fill in the blanks to count the number of rows for every combination of low birth weight (low vs normal) and mother's smoking status (yes vs no).

```
birthwt %>%
    __group_by__(low, __smoke__) %>%
    __summarize__(numRows = n())
```

(b) (5 points) Fill in the blanks to find the average and median weight of the 50 smallest newborns in the data.

```
birthwt %>%
    __slice_min__(__bwt__, n = 50) %>%
    __summarize__(avgWt = __mean__(bwt),
                  medWt = __median__(bwt))
```

4. (5 points) A company is creating a video game where the player fights increasingly difficult monsters through different layers of a dungeon. One team creates a dataset `monster_stats` with the combat statistics of different monsters. The column `name` gives the monster's name, `level` gives the difficulty level of the monster, and `atk` gives the monster's attack.

	name	level	atk
	<chr>	<dbl>	<dbl>
1	skeleton	1	30
2	ooze	2	45
3	haunt	4	75

A different team creates a dataset `monster_locations` that describes where different monsters appear. The column `name` gives the monster's name, `level` gives the layer of the dungeon where it lives, and `enviro` gives the monster's preferred environment.

	name	level	enviro
	<chr>	<dbl>	<chr>
1	chicken	0	mild
2	spider	1	humid
3	skeleton	1	arid
4	ooze	3	humid

The following R code is used to join the two dataframes based on the `name` column.

```
full_join(monster_stats, monster_locations, join_by(name))
```

Using the grid below, re-create the dataframe that would result from the call to `full_join()`. 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. The column names and the ordering of rows/columns do not need to be exact to get full credit, but the contents of the data must be accurate.

	name	level.x	level.y	atk	enviro
1	skeleton	1	1	30	arid
2	ooze	2	3	45	humid
3	haunt	4	NA	75	NA
4	chicken	NA	0	NA	mild
5	spider	NA	1	NA	humid

5. The dataset `students` gives the name, lecture section, homework, quiz, and exam grades of several students in a class. Below are 5 example rows from the full dataset.

	name	section	hw	quiz	exam
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	John	1	90	95	85
2	Arya	1	87	92	83
3	Fred	2	93	99	84
4	Riley	2	85	89	75
5	Tyler	2	67	70	61

(a) (4 points) We want to make a new column that combines `hw`, `quiz`, and `exam` into an overall final score, then converts that number into a categorical letter grade based on a scale.

For example, a student who scores between 92-100 should get an "A", a student who scores between 88-92 should get an "AB", and so on. Which `dplyr` commands should we use to accomplish this goal? **Select ALL that apply**.

- [x] `mutate()`
- [ ] `relocate()`
- [ ] `as.factor()`
- [x] `case_when()`

(b) (3 points) Starting from the original `students` data, we want to create a dataframe that has a column for "assignment type" and a column for "value" rather than three separate `hw, quiz, exam` columns. Which R code can be used to create this data?. **Select ONE**.

	name	section	assign_type	value
	<chr>	<dbl>	<chr>	<dbl>
1	John	1	hw	90
2	John	1	quiz	95
3	John	1	exam	85
4	Arya	1	hw	87

- [ ] `students %>% summarize(assign_type = c(hw, quiz, exam), value = n())`
- [ ] `students %>% pivot_wider(names_from = assign_type, values_from = value)`
- [x] `students %>% pivot_longer(c(hw, quiz, exam), names_to = assign_type)`
- [ ] `students %>% count(c(hw, quiz, exam))`