Unit 2 Quiz B

Randomly select 3 questions to ask members of the group requesting the next set of assignments. If they get all questions correct, give them the entire stack of quizzes and assignments so that they can quiz the next group. If they get any question wrong, send them back to their seats to review and then try again in 10 minutes or more.

1. In R, what is the key difference between these two lines of code?

```
people %>% filter(name == "Amy")
people %>% filter(str_detect(name, "Amy"))
```

Answer: `==` checks for exact matches only, while str_detect() finds any occurrences of "Amy" within the name (e.g., "Amy", "Amy Smith", "Amya"). The second line would return more results since it matches partial strings.

2. Explain how to use `arrange()`, and how to reverse the order it gives you.

   Answer: `arrange()` takes a data set and a variable, and sorts the data set based on that variable, lowest to highest. If you want highest to lowest, use `desc()`: `gapminder %>% arrange(desc(lifeExp))`

3. In this ggplot code, what's wrong and how would you fix it?

```
gapminder %>%
  ggplot(aes(x = gdpPercap, y = lifeExp)) %>%
  geom_point()
```

Answer: In ggplot, new layers are added with a `+` instead of piped.

4. Explain what an aesthetic mapping is in ggplot.

   Answer: It's a relation between variables in your data set and aesthetics in the plot. It tells ggplot which variable should be represented on the x-axis, which variable should be on the y-axis, and which variable should be represented by color, etc.

5. Name three ggplot geoms and explain when to use them.

   Answer: geom_point() draws a scatterplot: use it when your X and Y variables are both continuous and you want to visualize the nature of their relationship. `geom_histogram()` draws a histogram: use it when your X variable is continuous and you want to visualize its distribution. `geom_smooth()` draws a line of best fit: use it with geom_point() to add a line of best fit to a scatterplot.

6. Suppose you have two tables:

```
employees: id, name, department
salaries: id, salary, bonus
```

Write code to combine them to show each employee's name and compensation.

Answer: `employees %>% left_join(salaries, join_by(id))`