Solutions, week 2

Loading packages

As always, the first thing I have to do is load the **tidyverse** package. Remember, as I have already installed the tidyverse package on this computer, I do not need to run **install.packages("tidyverse")**. I can go straight to the command below:

```
library(tidyverse)
```

Loading data

Let's load the data from the web!

```
taylor <- read_csv("https://bit.ly/taylor-data-csv")
adele <- read_csv("https://bit.ly/adele-data-csv")
mcr <- read_csv("https://bit.ly/mcr-data-csv")
```

Task - Taylor Swift

- Question 1 - How many of each of Taylor Swift's songs are in each key? (C sharp, E, etc)

The first thing I would do is look at the data, to find a variable that describes the key. I can do this using some of the code from last week. For example, I can look at the names of each variable:

```
names(taylor)
```

I can get a summary of the dataset.

```
summary(taylor)
```

I can look at the dataset in a window.

```
View(taylor)
```

And here's a bit of code I use all the time! Don't worry about understanding the mechanics of this code chunk. All you need to know is that it gives you the name of each variable and what type of variable it is (e.g. numeric, categorical, character etc.). You need to have tidyverse loaded to use it.

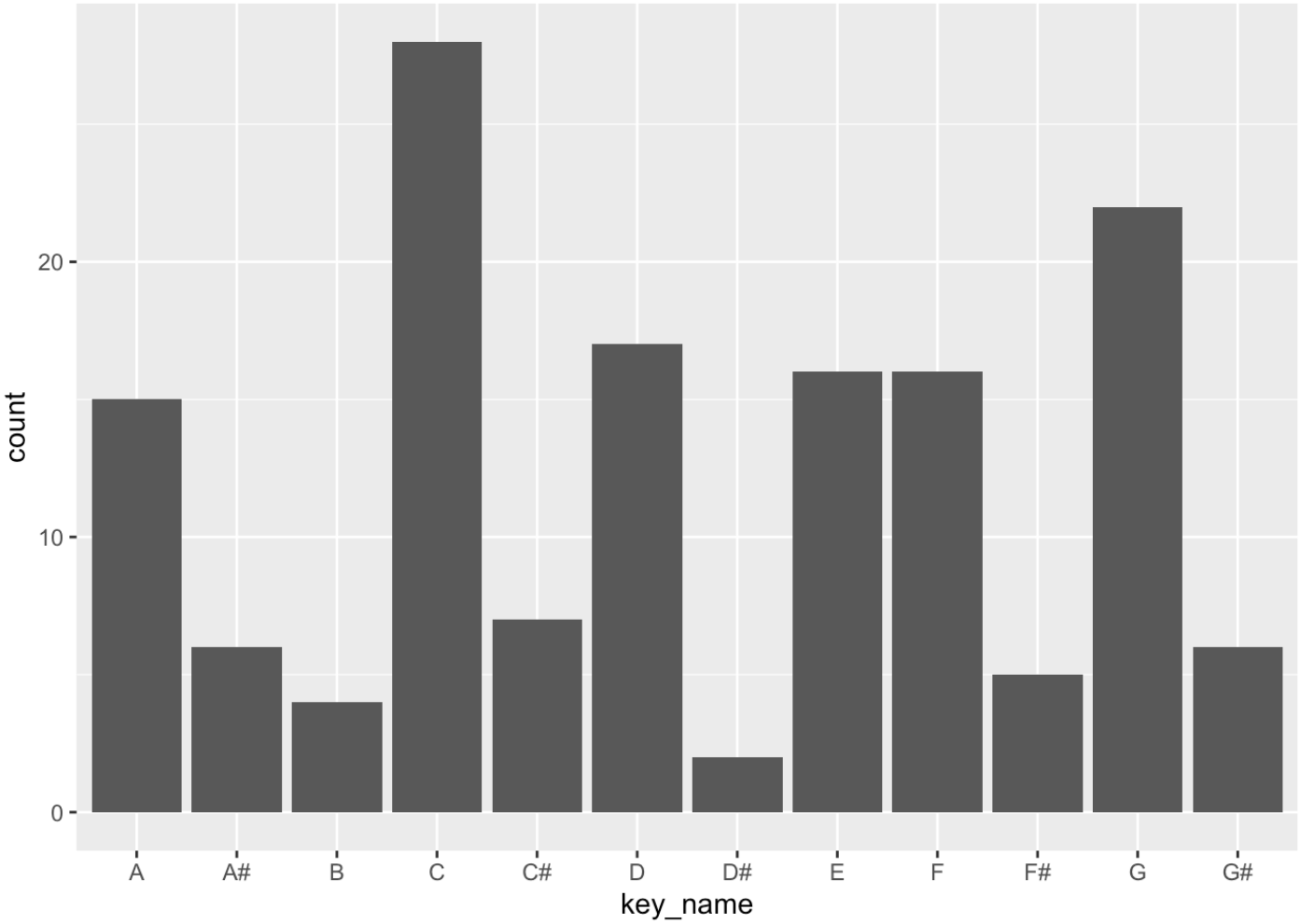
```
taylor %>%
  map_chr(class)
```

From this exploration of the data, I've found the right variable: key_name. I know this is the right one because it has a name which seems correct, and I've looked at the data and seen the observations contain letters that indicate keys (e.g. C#, D etc.). I could also read more about the variables at this link if I

wanted to be sure.

So let's make the graph.

```
ggplot(data = taylor) +
  aes(x = key_name) +
  geom_bar()
```

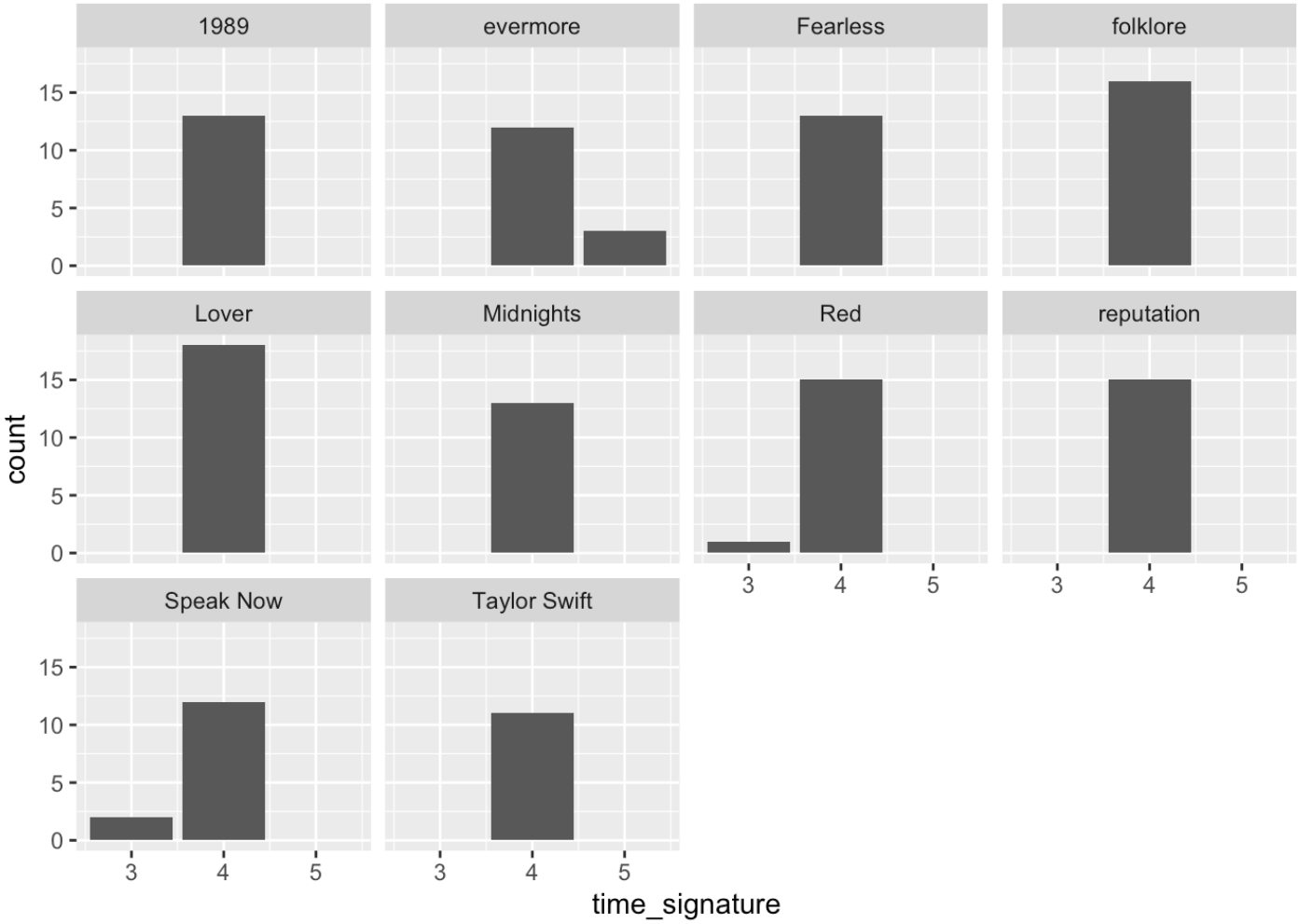


- Question 2 - On which Taylor Swift albums are there tracks with time signatures other than 4/4?

The relevant variable in this question is time_signature. But I also have to include album_name somewhere in my graph, as the question is asking for me to break it down by album.

There's a few ways to do this. First I'll show the most popular way in your submissions. And I think this is a great solution. So if you got it, well done!

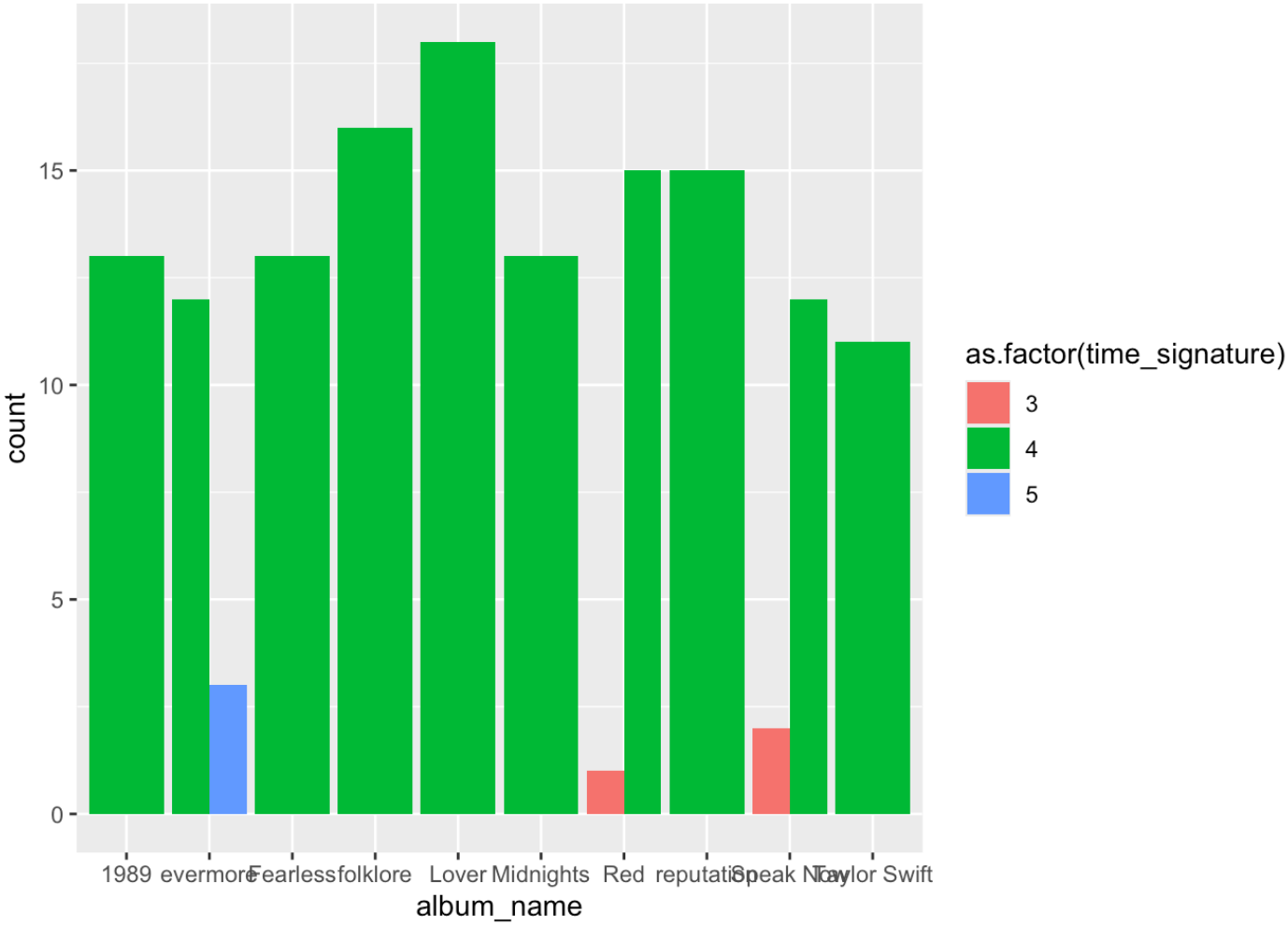
```
ggplot(data = taylor) +
  aes(x = time_signature) +
  geom_bar() +
  facet_wrap(~album_name)
```



Below is another solution some people used. They used a technique we did last week, **geom_bar(position = “dodge”)**. The album names aren’t readable, but we’ll get to that in another session.

There’s another problem though. R can get confused between different types of variables. The time_signature variable is indicated by numbers, so R thinks it is a numerical variable (it calls these a **double**). But actually time_signature is a categorical variable. There are a few different ways of storing categorical variables, but a very common one is as something called a **factor.** We’ll look at factors in detail in later weeks, so don’t worry if you didn’t get this. But in the code below you can see I’ve wrapped time_signature in a function called as.factor, which lets R know that it is a categorical variable.

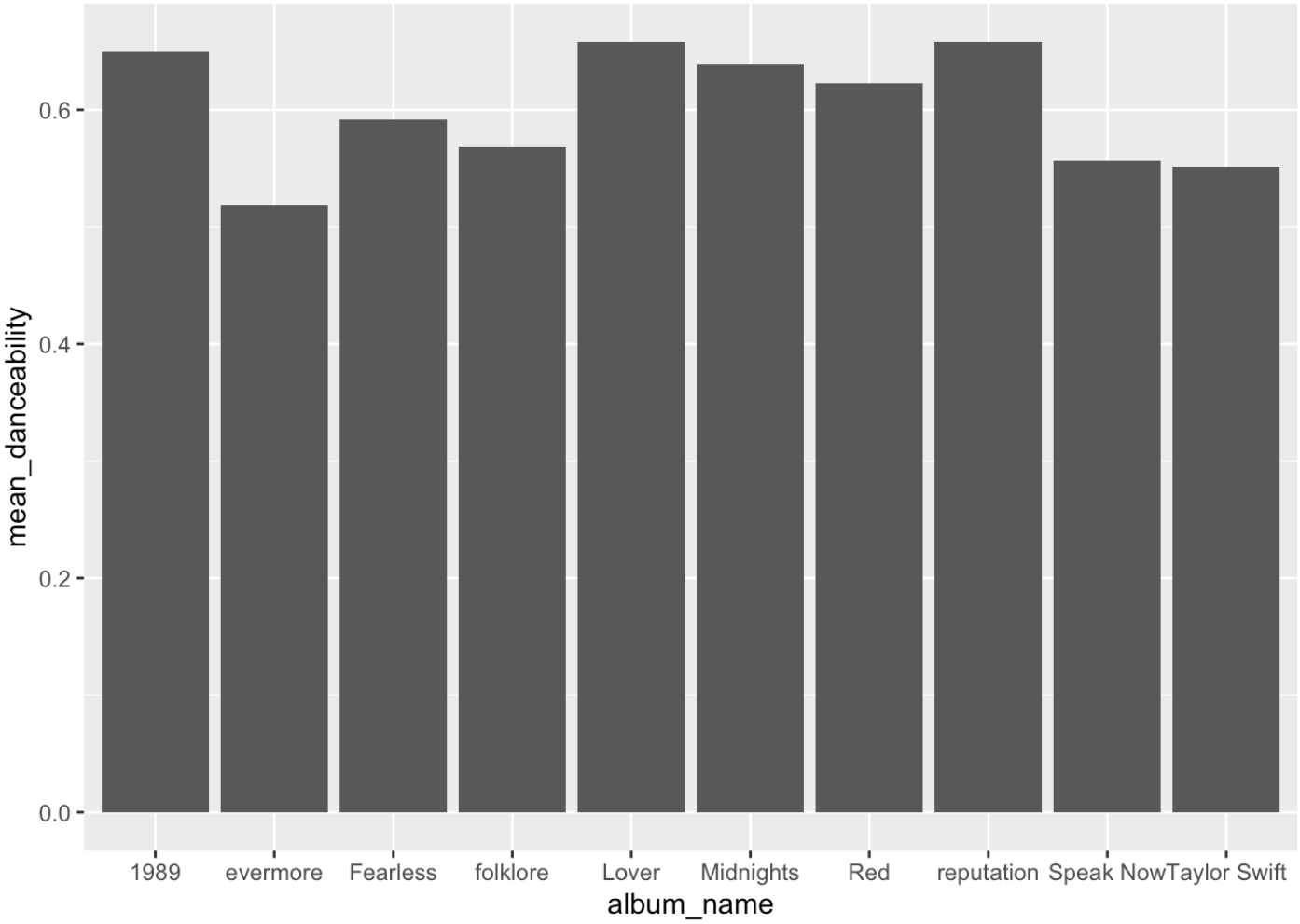
```
ggplot(data = taylor) +
  aes(x = album_name, fill = as.factor(time_signature)) +
  geom_bar(position = "dodge")
```



- Question 3 - Which Taylor Swift album has the highest average danceability?

The relevant variables here are danceability and album_name. But there's another key word, average. If you don't calculate an average per album, you will most likely produce a bar plot that sums all the values for danceability per album. Instead we need to use some dplyr commands to produce an average of danceability per album. The code below does this using group_by and summarise.

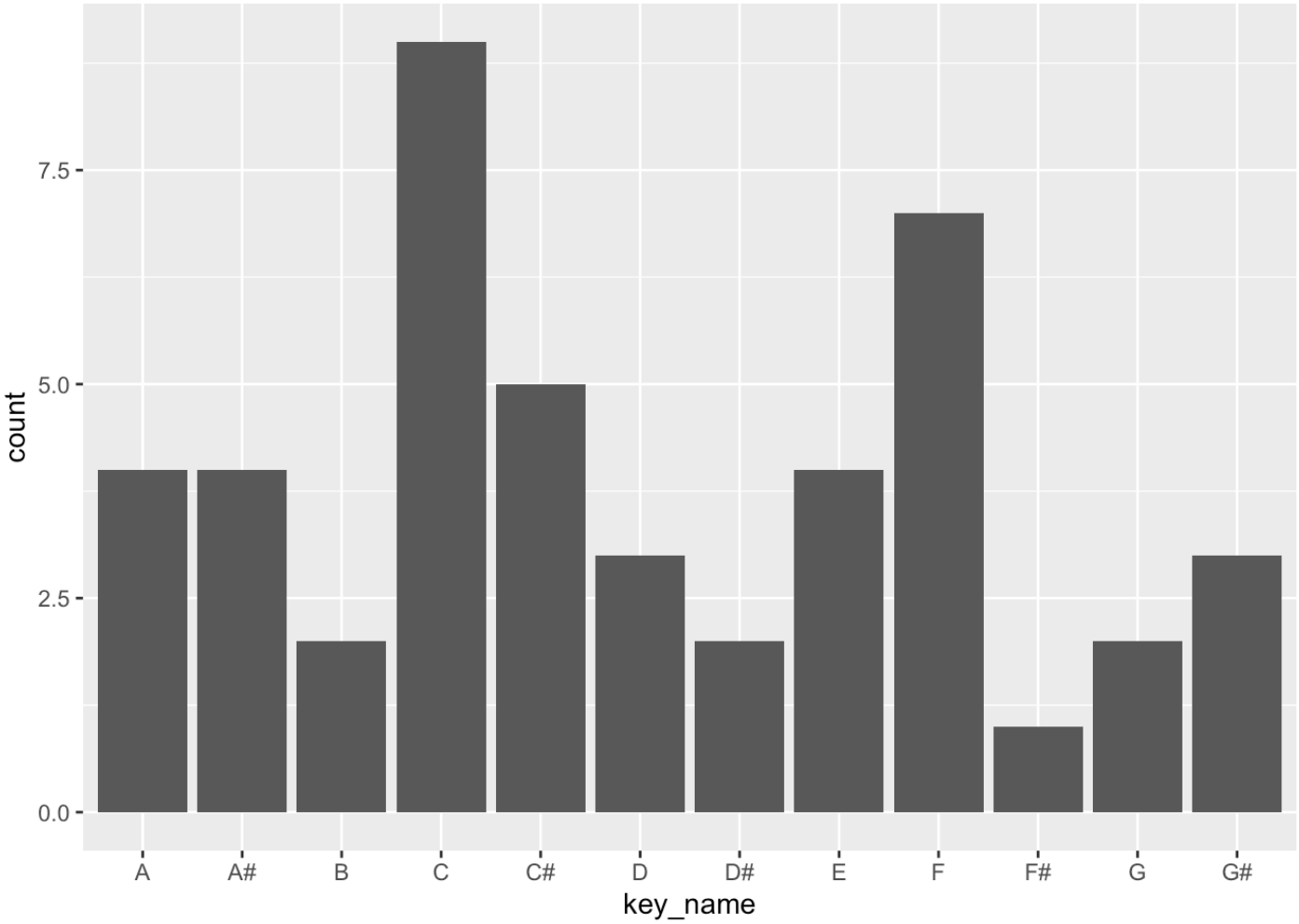
```
taylor %>%
  group_by(album_name) %>%
  summarise(mean_danceability = mean(danceability)) %>%
  ggplot() +
  aes(x = album_name, y = mean_danceability) +
  geom_col()
```



Task - do the same for another artist

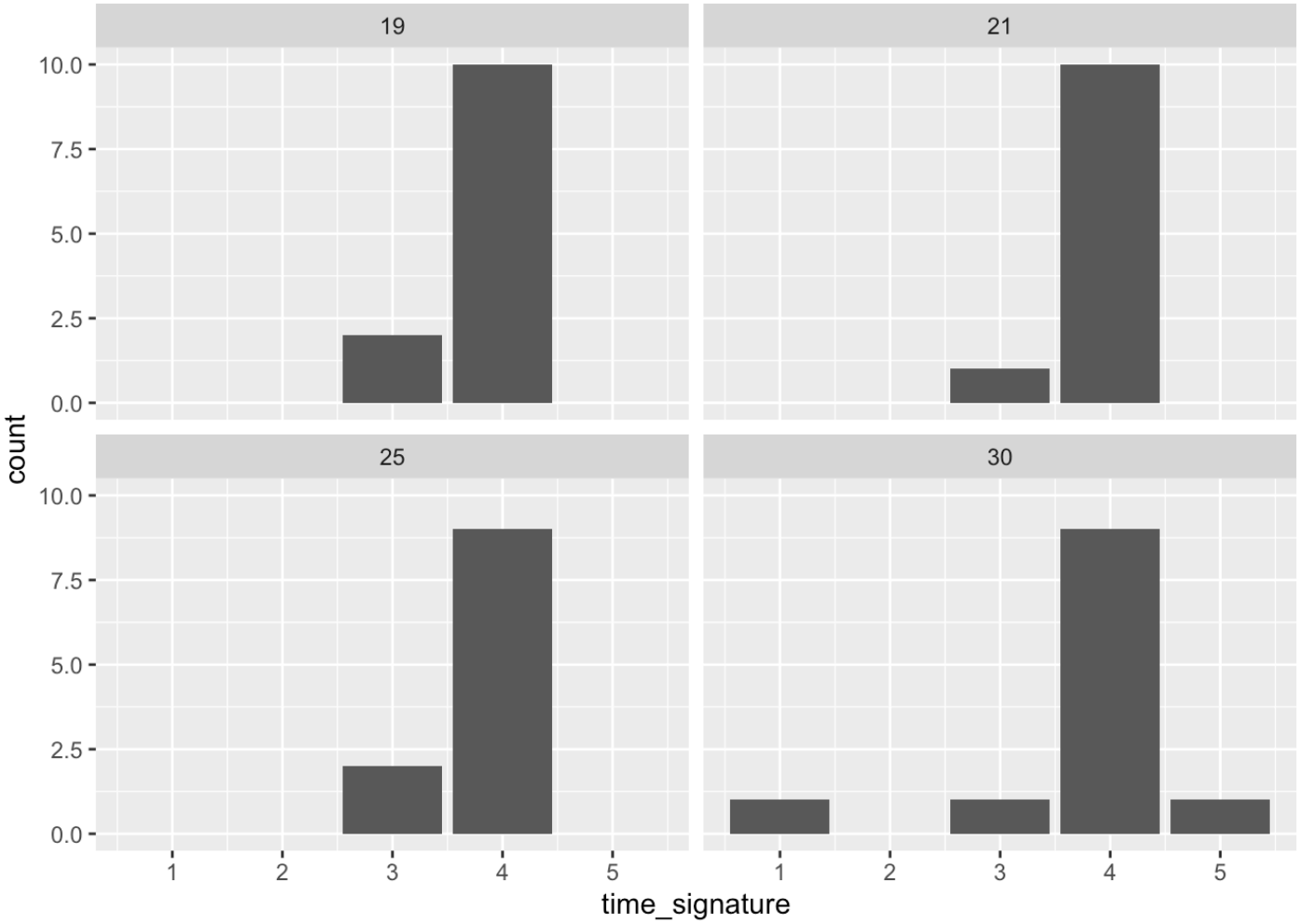
- Question 4 - How many of each of Adele's songs are in each key? (C sharp, E, etc)

```
ggplot(data = adele) +
  aes(x = key_name) +
  geom_bar()
```



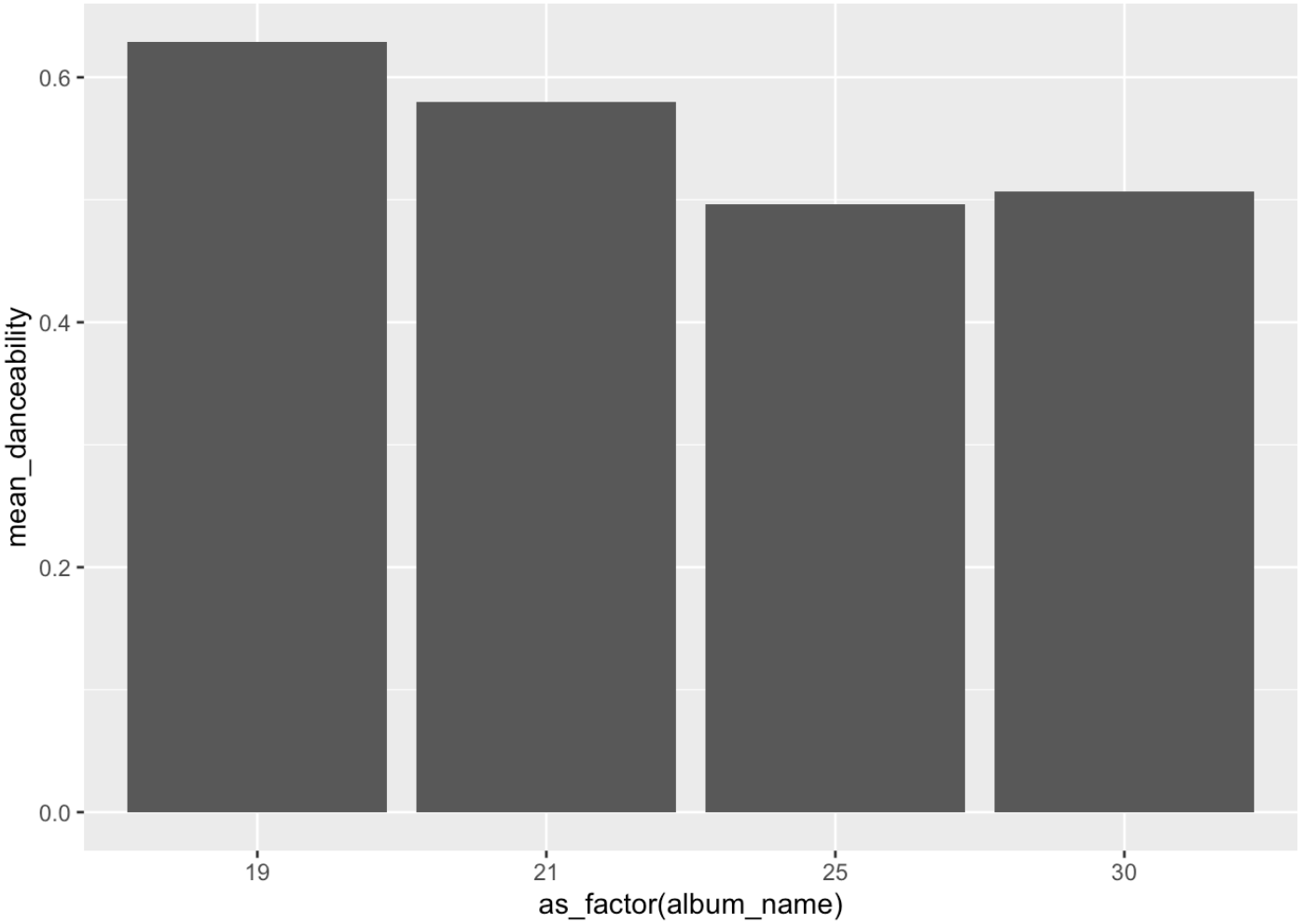
- Question 5 - On which Adele albums are there tracks with time signatures other than 4/4?

```
ggplot(data = adele) +
  aes(x = time_signature) +
  geom_bar() +
  facet_wrap(~album_name)
```



- Question 6 - Which Adele album has the highest average danceability?

```
adele %>%
  group_by(album_name) %>%
  summarise(mean_danceability = mean(danceability)) %>%
  ggplot() +
  aes(x = as_factor(album_name), y = mean_danceability) +
  geom_col()
```



Note that I've wrapped album_name in as_factor again. That's because all Adele's albums are named after numbers, so R assumes that you're working with a numeric variable. Instead, we need to explicitly state that it's a factor.

- Final question: show me something interesting!

There was a great mix here!

Common errors in the submissions

- Not reading the question properly! Please submit a graph for each question, many submissions had graphs missing
- Not including the album_name variable in question 2. If you produced a bar chart with time_signature on the x-axis, but didn't include album_name in some way (e.g. as a facet, or a fill aesthetic), then this will count the time signatures across all albums
- Not using group_by and summarise to produce an average for question 3. If you don't produce the average first, the height of the bar will be the sum of danceability values per album
- Using the wrong variables. Use the techniques at the top of this document to find the right variables. And read any documentation that accompanies a dataset
- Not connecting lines of code. Use the pipe %>% to connect your dplyr commands. Use the + sign to connect your ggplot commands. And format your lines of code as I have done here i.e. the first line of code starts immediately, then each line after that is indented