HW 9

Your Name Here

3/9/2023

Part 1: data drawings

This is our dataset. It is named `data`

(Note: I used a random name generator on the internet, and then randomly assigned favorite foods, major, and number of credits this semester!)

Note that the variables Name and Fav_food are character variables. Major is a factor variable (). The numbers in parentheses indicate the order. So, if we were actually looking at this data in R, it would just print that Nia's major is "art", not "art(1)"... but the level order 1 is stored internally.

Name	Fav_food	Major	Credits
Nia	ramen	art(1)	3.5
Lionel	pizza	chem(3)	4.0
Yan	ramen	art(1)	4.5
Jie	pizza	art(1)	3.0
Tom	pizza	bio(2)	4.0
Cloe	tacos	chem(3)	3.0
Carlo	pizza	bio(2)	4.0
Ping	tacos	dance(4)	3.5

For the following problems, copy the table above, and then modify it according to each instruction.

For example:

What does the dataset look like after running this code?

```
data %>%
  select(Name, Fav_food)
```

Answer:

Name	Fav_food
Nia	ramen
Lionel	pizza
Yan	ramen
Jie	pizza
Tom	pizza
Cloe	tacos
Carlo	pizza
Ping	tacos

1. What does the dataset look like after running this code?

```
data %>%
  filter(Fav_food == "ramen")
```

Answer:

Name	Fav_food	Major	Credits
Nia	ramen	art	3.5
Yan	ramen	art	4.5

2. What does the dataset look like after running this code?

```
data %>%
  count(Major)
```

Answer:

Major	n
art	3
bio	2
chem	2
dance	1

3. What does the dataset look like after running this code?

```
data %>%
  arrange(Name)
```

Answer:

Name	Fav_food	Major	Credits
Carlo	pizza	bio	4.0
Cloe	tacos	chem	3.0
Jie	pizza	art	3.0
Lionel	pizza	chem	4.0
Nia	ramen	art	3.5
Ping	tacos	dance	3.5
Tom	pizza	bio	4.0
Yan	ramen	art	4.5

3. What does the dataset look like after running this code?

```
data %>%
  mutate(cred4plus = Credits >=4)
```

Answer:

Name	Fav_food	Major	Credits	cred4plus
Nia	ramen	art	3.5	FALSE
Lionel	pizza	chem	4.0	TRUE
Yan	ramen	art	4.5	TRUE
Jie	pizza	art	3.0	FALSE
Tom	pizza	bio	4.0	TRUE
Cloe	tacos	chem	3.0	FALSE
Carlo	pizza	bio	4.0	TRUE

Name	Fav_food	Major	Credits	cred4plus
Ping	tacos	dance	3.5	FALSE

4. What does the dataset look like after running this code?

```
data %>%
  rename(credits_spring2023 = Credits)
```

Answer:

Name	Fav_food	Major	credits_spring2023
Nia	ramen	art	3.5
Lionel	pizza	chem	4.0
Yan	ramen	art	4.5
Jie	pizza	art	3.0
Tom	pizza	bio	4.0
Cloe	tacos	chem	3.0
Carlo	pizza	bio	4.0
Ping	tacos	dance	3.5

4. What does the dataset look like after running this code?

```
data %>%
  group_by(Major) %>%
  summarize(max_credits = max(Credits))
```

Answer:

Major	max_credits
art	4.5
bio	4.0
chem	4.0
dance	3.5

5. What does the dataset look like after running this code?

```
data %>%
  mutate(Major = fct_reorder(Major, Credits, .fun = max))
```

Answer:

Name	Fav_food	Major	Credits
Nia	ramen	art(1)	3.5
Lionel	pizza	chem(3)	4.0
Yan	ramen	art(1)	4.5
Jie	pizza	art(1)	3.0
Tom	pizza	bio(2)	4.0
Cloe	tacos	chem(3)	3.0
Carlo	pizza	bio(2)	4.0
Ping	tacos	dance(4)	3.5

Part 2: Spotify data

```
library(tidyverse)
```

```
bigspotify <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data
  mutate(year = parse_number(substr(track_album_release_date, 3, 4)),
         decade = (year %/% 10)*10,
         decade = str_c(as.character(decade), "s", sep = ""))
```

```
## Rows: 32833 Columns: 23
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): track_id, track_name, track_artist, track_album_id, track_album_na...
## dbl (13): track_popularity, danceability, energy, key, loudness, mode, speec...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
bigspotify %>% count(decade)
```

```
## # A tibble: 8 x 2
##   decade     n
##   <chr>  <int>
## 1 0s      4077
## 2 10s    23214
## 3 20s      785
## 4 50s        3
## 5 60s      172
## 6 70s      966
## 7 80s     1306
## 8 90s     2310
```

6. Above you can see that I've tried to create a variable to indicate the decade of a song... but it needs some help. First, modify the decade variable so that the aughts are "00s" instead of "0s". Print your result with count (see below):

```
bigspotify|>
  mutate(decade = as.factor(decade))|>
  mutate(decade = fct_recode(decade, "00s" = "0s"))|>
  count(decade)
```

```
## # A tibble: 8 x 2
##   decade     n
##   <fct>  <int>
## 1 00s     4077
## 2 10s    23214
## 3 20s      785
## 4 50s        3
## 5 60s      172
## 6 70s      966
## 7 80s     1306
## 8 90s     2310
```

7. Now add another line to the code above so that 00s, 10s, and 20s, come AFTER the 90s. Again, print your result with count.

```
bigspotify|>
  mutate(decade = as.factor(decade))|>
  mutate(decade = fct_recode(decade, "00s" = "0s"))|>
  mutate(decade = fct_relevel(decade, "90s"))|>
  count(decade)
```

```
## # A tibble: 8 x 2
##   decade     n
##   <fct>  <int>
## 1 90s     2310
## 2 00s     4077
## 3 10s    23214
## 4 20s      785
## 5 50s        3
## 6 60s      172
## 7 70s      966
## 8 80s     1306
```

8. Find the average track_popularity, danceability, and speechiness by decade. Write 1 sentence about what you observe for each variable. (If you got 6-7 to work, use that ordering for decade. If not, just use the original bigspotify!)

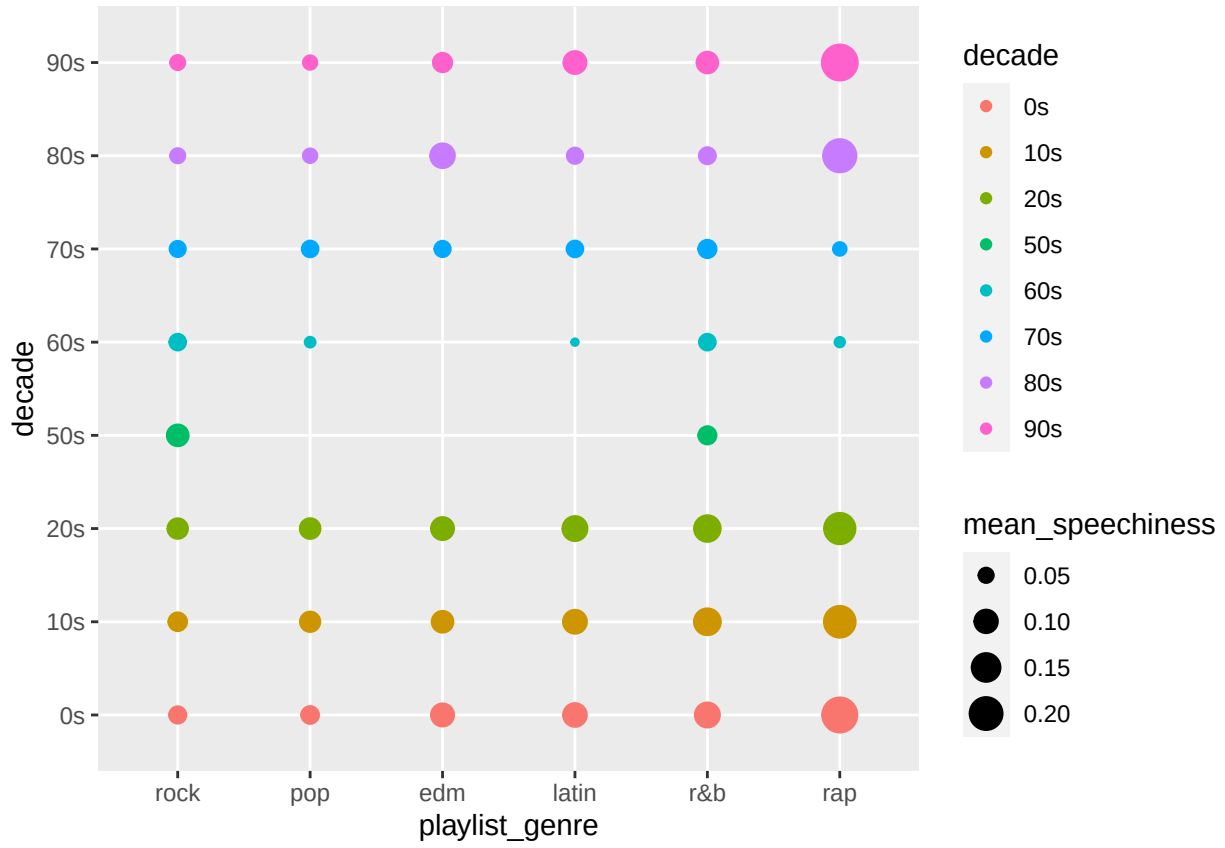
```
bigspotify|>
  mutate(decade = as.factor(decade))|>
  mutate(decade = fct_recode(decade, "00s" = "0s"))|>
  mutate(decade = fct_relevel(decade, "90s"))|>
  group_by(decade)|>
  summarise(mean_pop = mean(track_popularity), mean_dance = mean(danceability), mean_speech = mean(spee
```

```
## # A tibble: 8 x 4
##   decade mean_pop mean_dance mean_speech
##   <fct>     <dbl>      <dbl>       <dbl>
## 1 90s        40.0      0.670      0.119
## 2 00s        33.6      0.648      0.114
## 3 10s        43.8      0.663      0.110
## 4 20s        46.5      0.672      0.115
## 5 50s        44.3      0.562      0.0789
## 6 60s        49.3      0.511      0.0557
## 7 70s        46.7      0.539      0.0544
## 8 80s        44.6      0.603      0.0581
```

9. Create a plot that helps to visualize speechiness by playlist_genre and decade. Be sure that playlist_genre is in a meaningful order in your graph. Write 1-2 sentences about what you observe. (Note: you might want to drop the 50s since there are so few songs!)

```
bigspotify|>
  mutate(playlist_genre = fct_reorder(playlist_genre, speechiness, .fun = mean))|>
  group_by(playlist_genre, decade)|>
  summarise(mean_speechiness = mean(speechiness))|>
  ggplot(aes(playlist_genre, decade))+
  geom_point(aes(size = mean_speechiness, color = decade))
```

```
## `summarise()` has grouped output by 'playlist_genre'. You can override using
## the `.groups` argument.
```

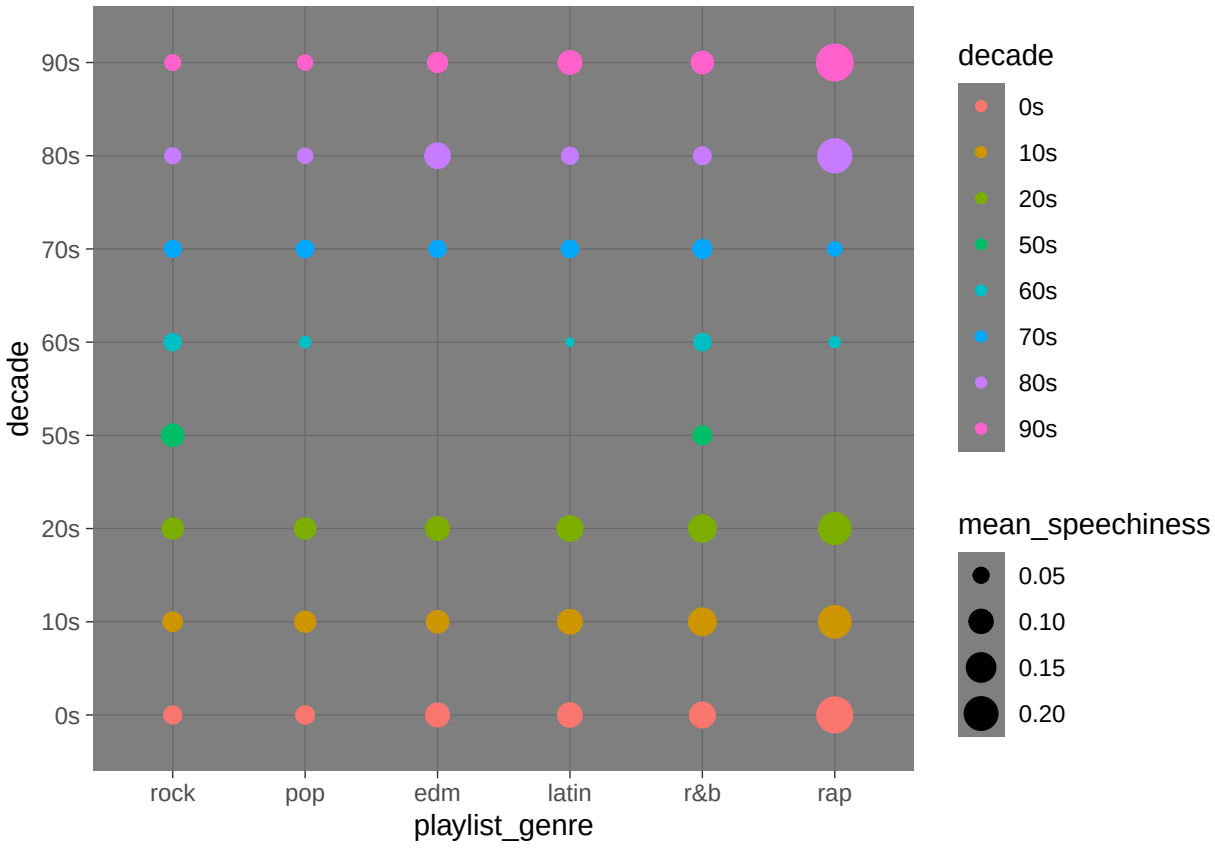


Ans: Rock has the lowest mean speechiness and the rap genre has the highest mean speechiness. Also the speechiness in rock genre is more evenly distributed in rock and r&b than other genres. There is not much change in speechiness with respect to changes in decades for a specific genre.

10. Pick a new theme and/or color scale for your plot above! (hint start typing theme_ then hit tab to see options!)

```
bigspotify|>
  mutate(playlist_genre = fct_reorder(playlist_genre, speechiness, .fun = mean))|>
  group_by(playlist_genre, decade)|>
  summarise(mean_speechiness = mean(speechiness))|>
  ggplot(aes(playlist_genre, decade))+
  geom_point(aes(size = mean_speechiness, color = decade))+
  theme_dark()
```

```
## `summarise()` has grouped output by 'playlist_genre'. You can override using
## the `.groups` argument.
```


decade
90s
80s
70s
60s
50s
20s
10s
0s
rock
pop
edm
latin
r&b
rap
playlist_genre
decade
0s
10s
20s
50s
60s
70s
80s
90s
mean_speechiness
0.05
0.10
0.15
0.20