Week 3 solutions

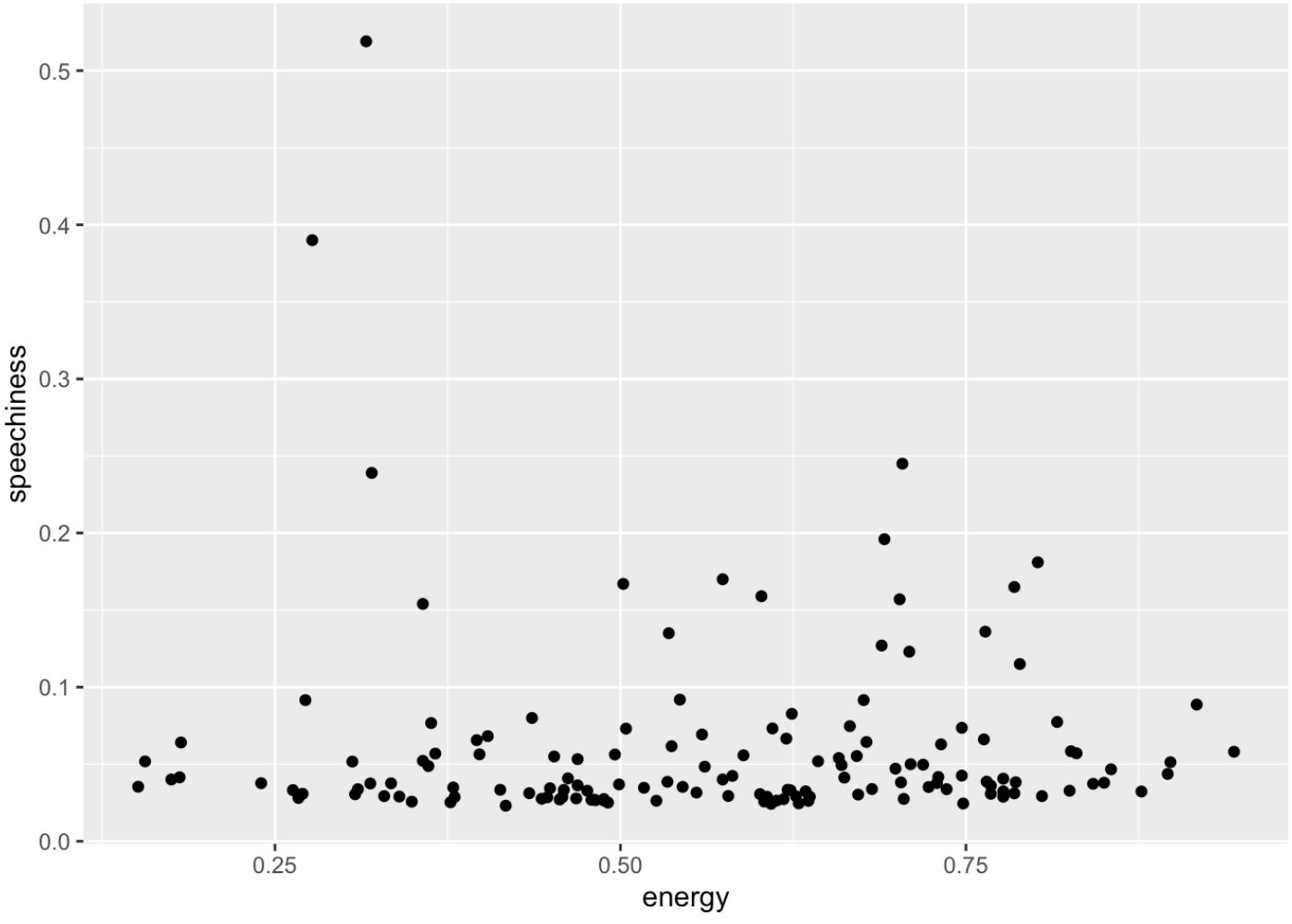
Before we start, let's load some packages and some data.

```
library(tidyverse)
library(lubridate)
```

```
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")
ghost <- read_csv("https://bit.ly/ghost-data-csv")
beyonce <- read_csv("https://bit.ly/beyonce-data-csv")
```

Q1: What's the relationship between energy and speechiness in Taylor Swift's albums?

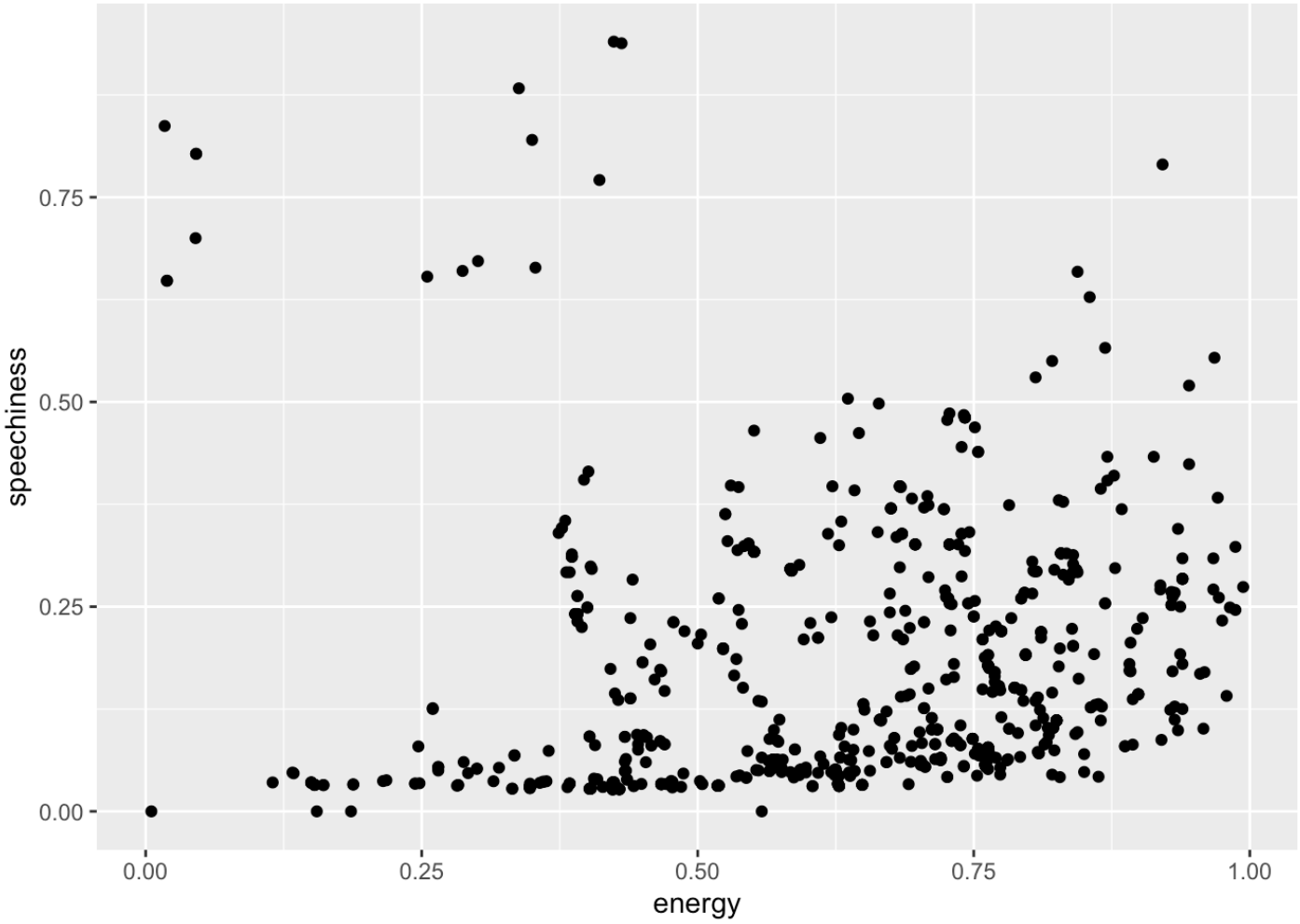
```
ggplot(data = taylor) +
  aes(x = energy, y = speechiness) +
  geom_point()
```



We can see that very few of Taylor Swift's songs have high speechiness, but there's one that's very high, up at the top left.

Q2: …and Beyonce's?

```
ggplot(data = beyonce) +
  aes(x = energy, y = speechiness) +
  geom_point()
```

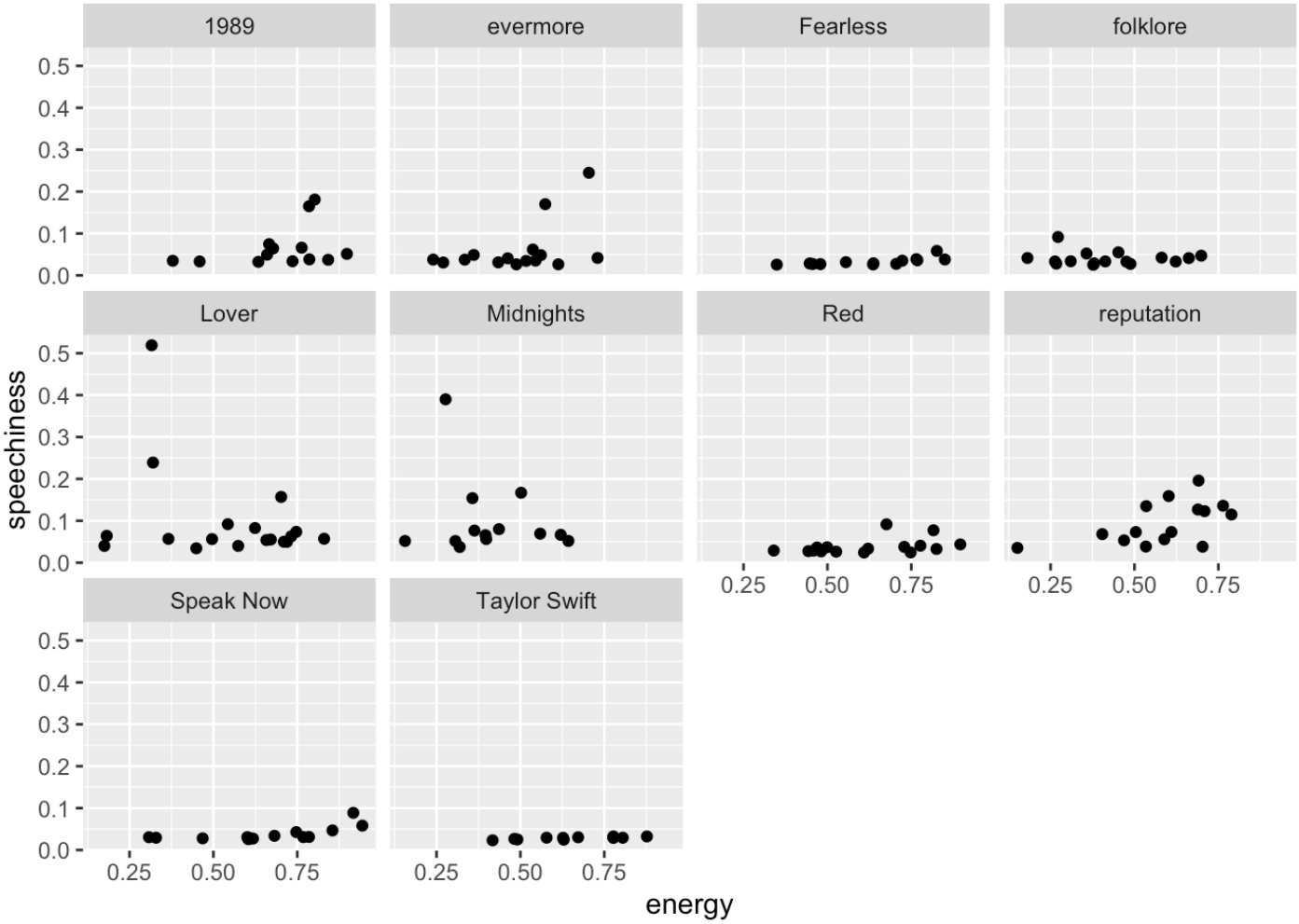


This is the figure that you generate when you use all of Beyonce's songs. You can see there's an "island" of sorts up at the top left with tracks with high speechiness and low energy.

Q3: How do these relationships vary by album?

For simplicity, let's look at Taylor Swift again.

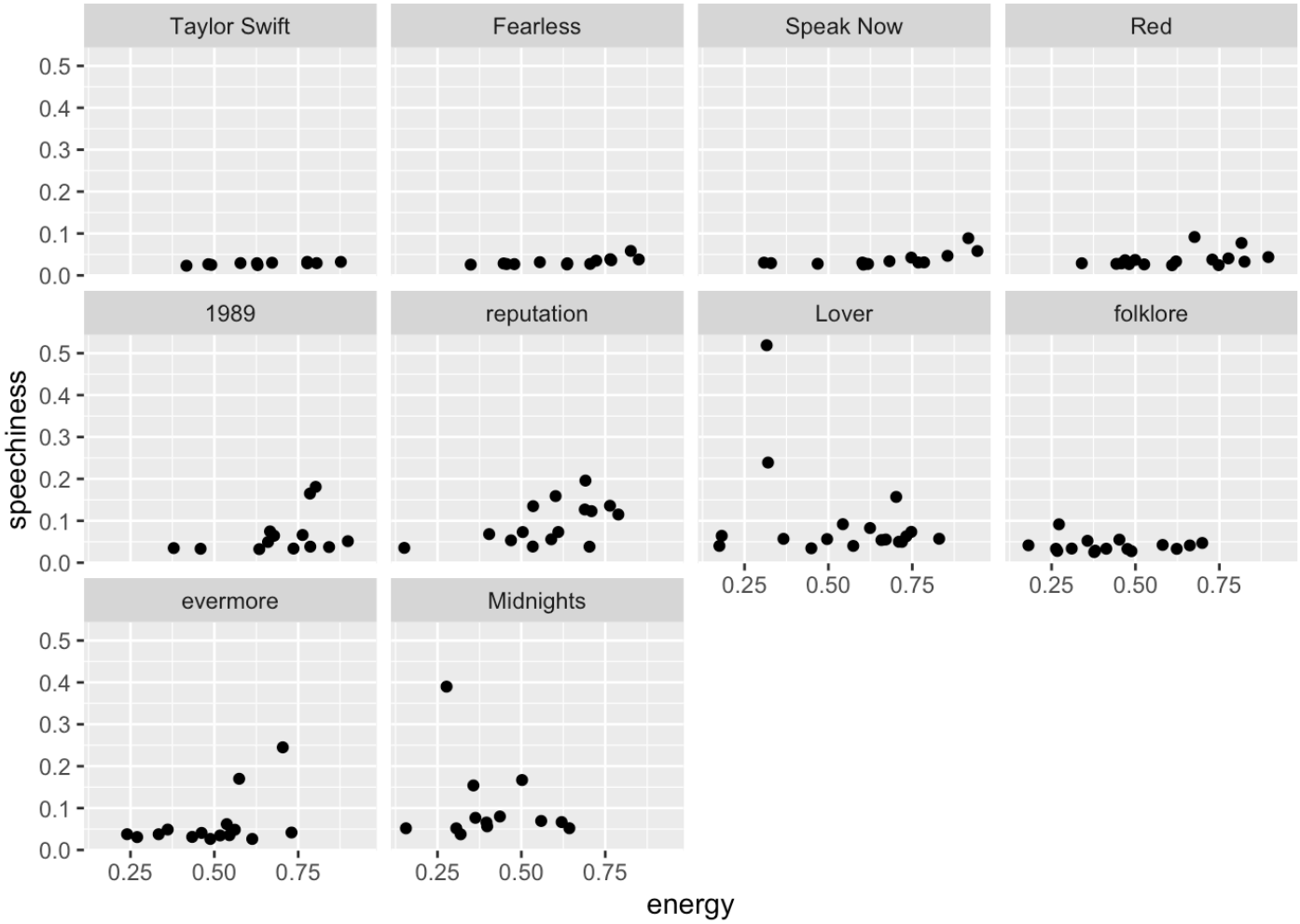
```
ggplot(data = taylor) +
  aes(x = energy, y = speechiness) +
  geom_point() +
  facet_wrap(~ album_name)
```



This shows us that the majority of Taylor Swift's albums have super-low speechiness. However, as before, it's hard to interpret this data give the albums are in alphabetical order. Let's reorganise the plot so we can see how this has changed over her career, by generating a **levels** object again.

```
taylor_levels <-
  c("Taylor Swift",
    "Fearless",
    "Speak Now",
    "Red",
    "1989",
    "reputation",
    "Lover",
    "folklore",
    "evermore",
    "Midnights")

ggplot(data = taylor) +
  aes(x = energy, y = speechiness) +
  geom_point() +
  facet_wrap(~ factor(album_name,
                      levels = taylor_levels))
```



This shows that her earlier albums had super-low speechiness, while her more recent albums have been more of a mix of high and low speechiness, with Lover featuring the track we saw earlier with super-high speechiness.

Q4: What about for My Chemical Romance?

In this case, I asked you to put the albums in chronological order. However, the data's inconsistent. Let's have a look at the data:

```
head(mcr)
```

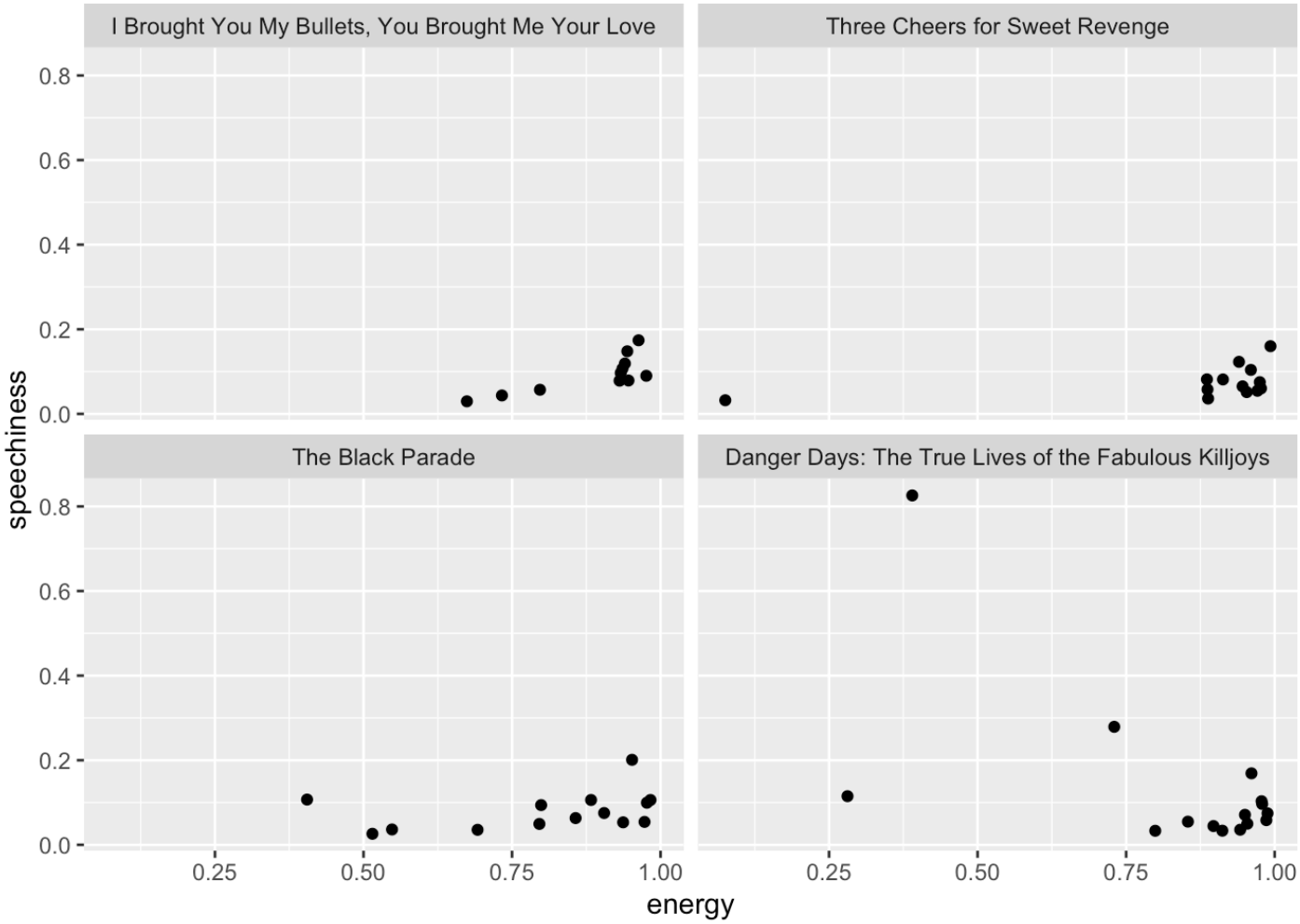
```
## # A tibble: 6 × 22
##   artist_name        album_release_date danceability energy   key loudness  mod
e
##   <chr>              <chr>                      <dbl>  <dbl> <dbl>    <dbl> <dbl
>
## 1 My Chemical Roman… 2002                       0.311  0.733     4   -18.0
0
## 2 My Chemical Roman… 2002                       0.383  0.946     1    -8.12
0
## 3 My Chemical Roman… 2002                       0.327  0.963     4    -8.87
0
## 4 My Chemical Roman… 2002                       0.306  0.94      9    -6.48
1
## 5 My Chemical Roman… 2002                       0.271  0.931     1    -6.81
0
## 6 My Chemical Roman… 2002                       0.291  0.976     2    -6.44
1
## # ℹ 15 more variables: speechiness <dbl>, acousticness <dbl>,
## #   instrumentalness <dbl>, liveness <dbl>, valence <dbl>, tempo <dbl>,
## #   time_signature <dbl>, duration_ms <dbl>, explicit <lgl>, track_name <chr>,
## #   track_number <dbl>, album_name <chr>, key_name <chr>, mode_name <chr>,
## #   key_mode <chr>
```

You can see that **album_release_date** here is a character variable, rather than a date: that's because their first album is just specified as having been released at some point in 2002, rather than sharing the specific date.

There's a few ways we can resolve this issue, but the most straightforward is just to order the tracks ourselves manually, as we did for Taylor Swift. If you don't know what order these albums were released in, the easiest thing is to just Google it.

```
mcr_levels <-
  c("I Brought You My Bullets, You Brought Me Your Love",
    "Three Cheers for Sweet Revenge",
    "The Black Parade",
    "Danger Days: The True Lives of the Fabulous Killjoys")

ggplot(data = mcr) +
  aes(x = energy, y = speechiness) +
  geom_point() +
  facet_wrap(~ factor(album_name,
                      levels = mcr_levels))
```

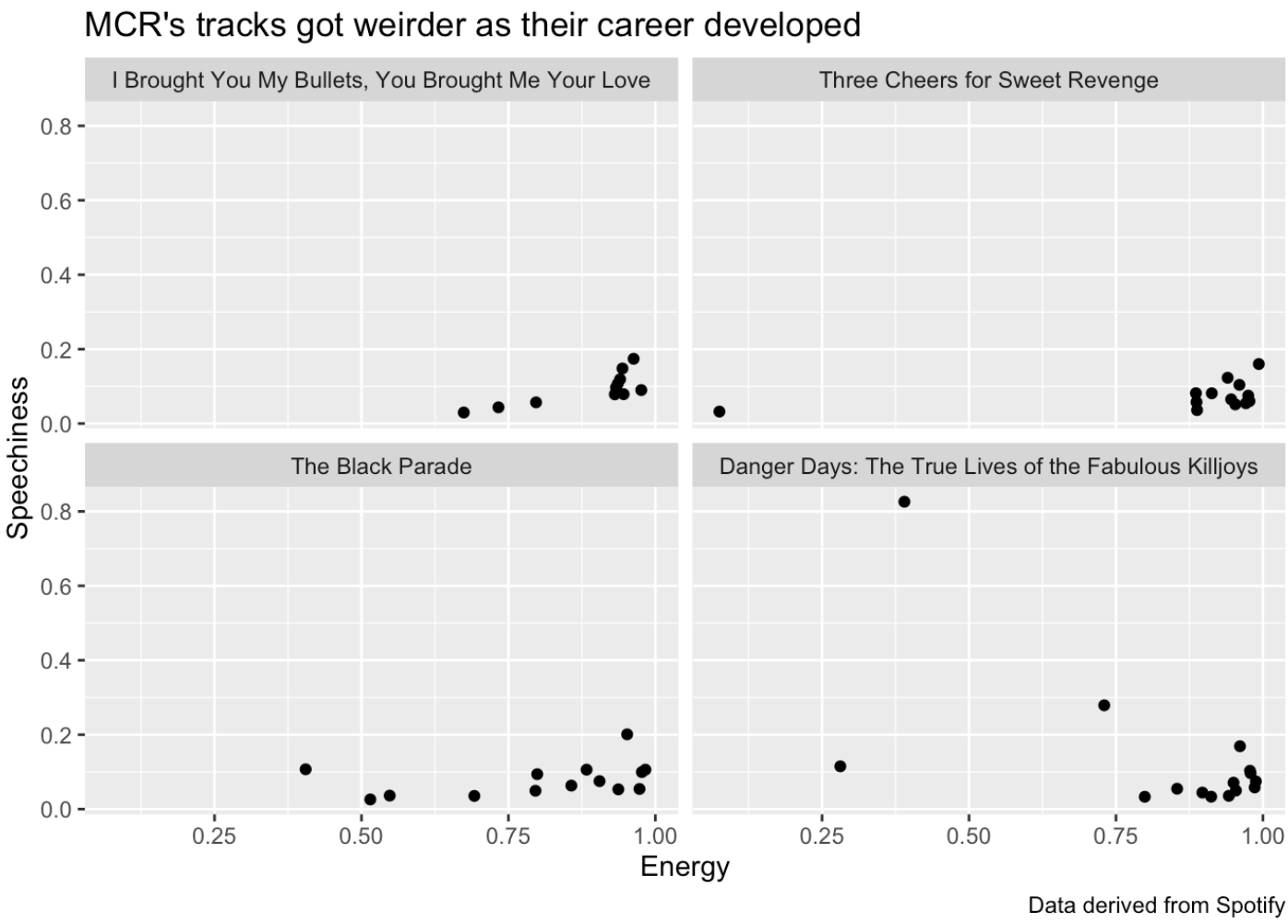


We can see some extreme outliers here – on Three Cheers for Sweet Revenge, the majority of tracks have high energy and low speechiness, but there's one track with super-low energy.

Let's clean up

We need to add labels! Let's do that now.

```
ggplot(data = mcr) +
  aes(x = energy, y = speechiness) +
  geom_point() +
  facet_wrap(~ factor(album_name,
                      levels = mcr_levels)) +
  labs(x = "Energy",
       y = "Speechiness",
       title = "MCR's tracks got weirder as their career developed",
       caption = "Data derived from Spotify")
```



General feedback

- Remember to make your submissions ‘stand-alone’, with all of the code required to replicate them. For example, if you use a vector called ‘taylor_levels’, make sure you include the code used to create that vector in your submission.
- When using technical terms to interpret the plot, you might also want to think about how you could interpret the plot for a more general audience (e.g. how would you describe it in a blog post for Taylor Swift fans compared to to academics?)
- Remember to include your interpretation of the plots — it’s common to underestimate how much practice is needed picking out the interesting things from data visualisations.
- Remember that outliers and clusters (‘islands’) can be interesting things to point out when interpreting data visualisations, not least because they are the most obvious in data visualisations and the least obvious when they are ‘hiding’ in the data itself.
- Not everyone had a go at practicing adding labels to their plots — I’d encourage you to have a go at this because it can help you learn how to write good active titles, etc.