Spotify Project Markdown

Andrew Gregory

2022-06-29

```
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("ggExtra")
library("ggthemes")
library("ggpubr")
library("magrittr")
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

Data loading

```
# Load data for the 2010's
df10 <- read_csv(file = "Project_Data/dataset-of-10s.csv")
```

```
## Rows: 6398 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): track, artist, uri
## dbl (16): danceability, energy, key, loudness, mode, speechiness, acousticne...
##
## 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.
```

Feature Engineering

```
# Change to seconds for the sake of interpretability
df10 %<>%
  mutate(duration_s= duration_ms / 1000 )
```

```
# Change these columns to factors
df10 %<>%
  mutate(across(c(mode,key,time_signature), ~ factor(.x)))
# List factor column
df10 %>%
  select(where(is.factor))
```

```
## # A tibble: 6,398 x 3
##    key   mode  time_signature
##    <fct> <fct> <fct>
##  1 1     0     4
##  2 5     0     3
##  3 9     0     4
##  4 0     0     4
##  5 1     1     4
##  6 0     1     4
##  7 0     1     4
##  8 2     1     4
##  9 7     1     4
## 10 8     1     4
## # ... with 6,388 more rows
```

```
# Delete uninformative columns
df10 %<>%
  select(-track,-artist,-uri,-target,-duration_ms)
```

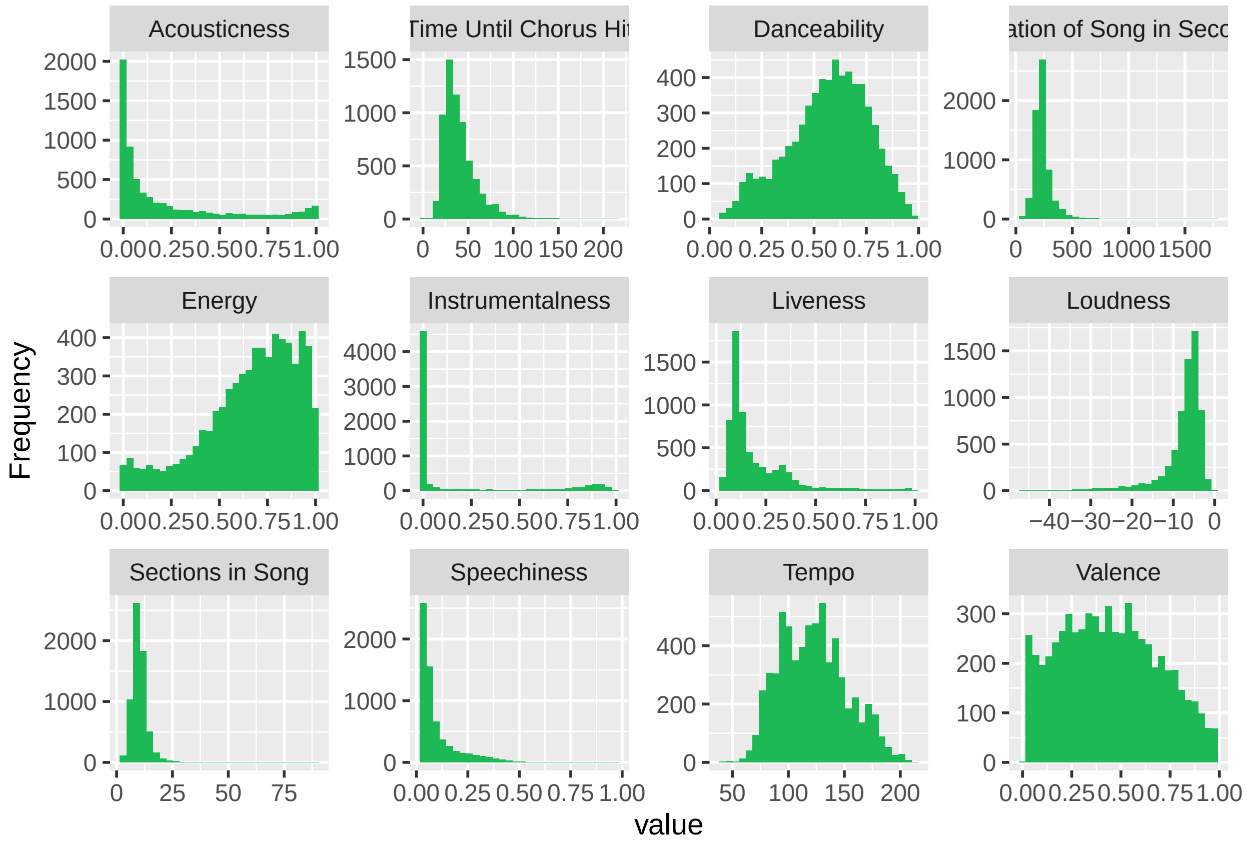
Define Spotify green for the sake of plotting

```
spotify_green <- "#1DB954"
```

Univariate plots

```
# numeric plots
numeric_labels <- c(`sections` = "Sections in Song", `chorus_hit` = "Time Until Chorus Hit"
                    ,`duration_s` = "Duration of Song in Seconds",`tempo` =  "Tempo",
  `valence` = "Valence",`liveness` =  "Liveness",  `instrumentalness` = "Instrumentalness",
  `acousticness` = "Acousticness", `speechiness` = "Speechiness", `loudness` = "Loudness",
  `energy` = "Energy", `danceability` = "Danceability")
df10 %>%
  select(where(is.numeric)) %>%
  gather() %>%
  ggplot(aes(value)) +
  geom_histogram(fill = spotify_green) +
  ylab("Frequency") +
  facet_wrap(~key, scales = 'free',labeller = as_labeller(numeric_labels))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Acousticness
2000
1500
1000
500
0
0.00 0.25 0.50 0.75 1.00
Time Until Chorus Hi
1500
1000
500
0
0 50 100 150 200
Danceability
400
300
200
100
0
0.00 0.25 0.50 0.75 1.00
ation of Song in Seco
2000
1000
0
0 500 1000 1500
Energy
400
300
200
100
0
0.00 0.25 0.50 0.75 1.00
Instrumentalness
4000
3000
2000
1000
0
0.00 0.25 0.50 0.75 1.00
Liveness
1500
1000
500
0
0.00 0.25 0.50 0.75 1.00
Loudness
1500
1000
500
0
−40 −30 −20 −10 0
Sections in Song
2000
1000
0
0 25 50 75
Speechiness
2000
1000
0
0.00 0.25 0.50 0.75 1.00
Tempo
400
200
0
50 100 150 200
Valence
300
200
100
0
0.00 0.25 0.50 0.75 1.00
Frequency
value