ScriptForLab5.R

harrisonwolfe

2025-02-22

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
#Step 1
allentown.data = read_csv("data/essentia.data.allentown.csv")
```

```
## Rows: 1 Columns: 204
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): artist, album, track, chords_scale, chords_key, mode
## dbl (197): overall_loudness, spectral_energy, danceability, tempo, duration,...
## lgl   (1): key
##
## 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.
```

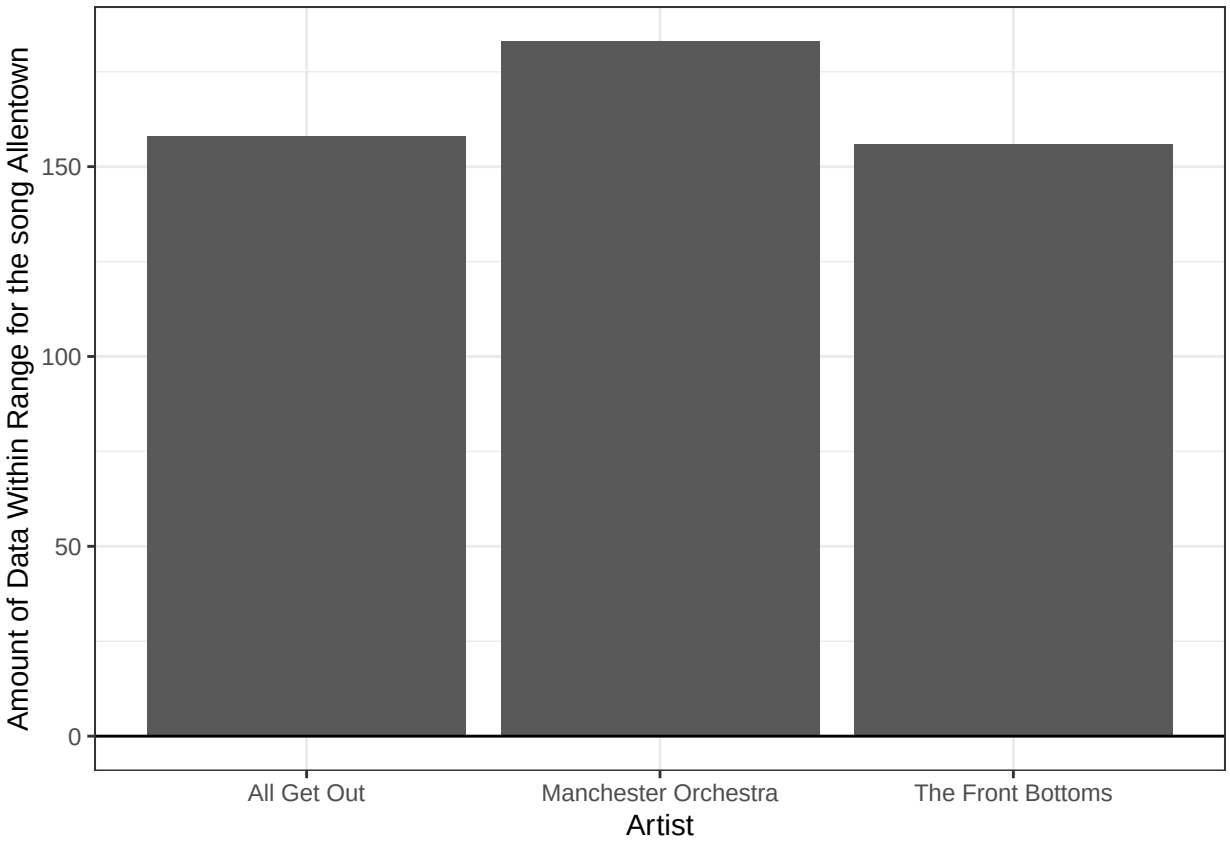
```
full.data = read_csv("data/essentia.data.csv")
```

```
## Rows: 180 Columns: 204
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (7): artist, album, track, chords_scale, chords_key, key, mode
## dbl (197): overall_loudness, spectral_energy, danceability, tempo, duration,...
##
## 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.
```

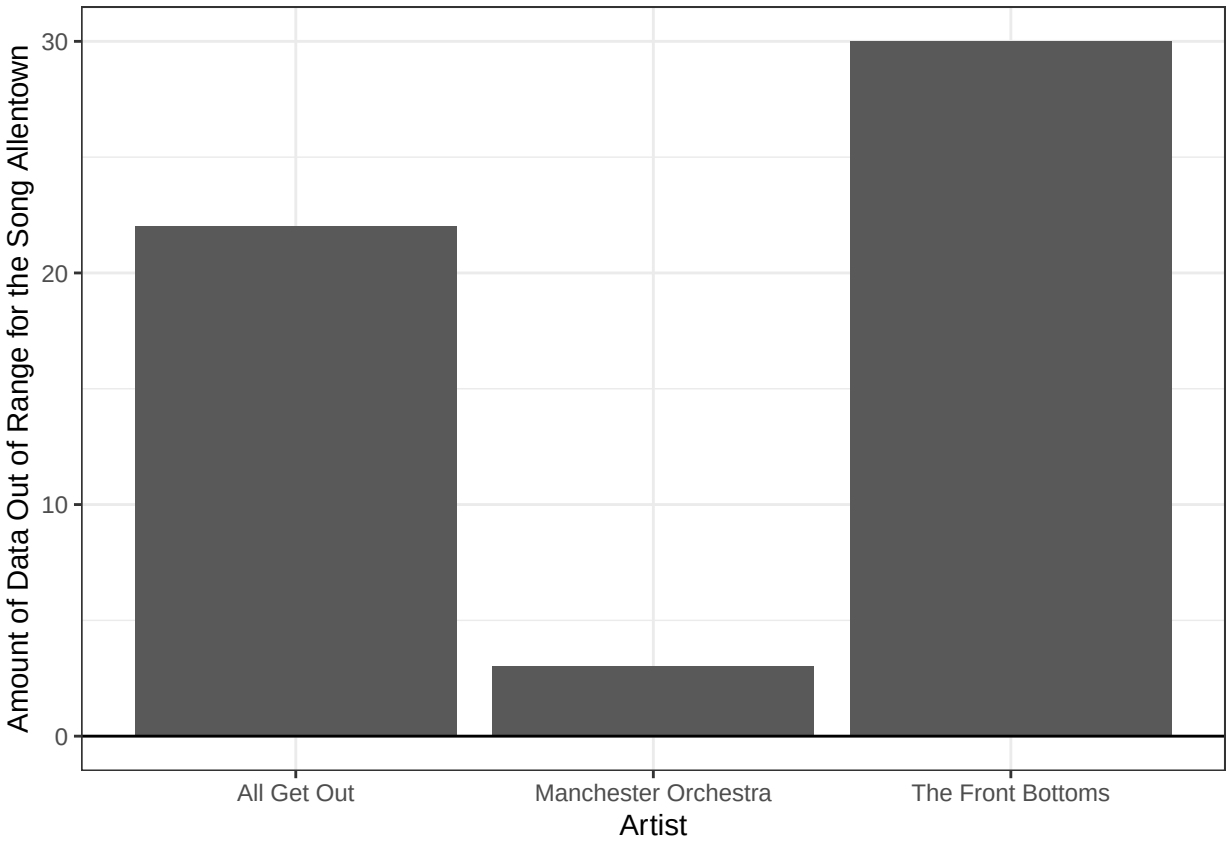
```
loudness.artist.data = full.data |>
  group_by(artist) |>
```

```
  summarize(#Finds the stats
    IQR = quantile(overall_loudness, .75) - quantile(overall_loudness, 0.25),
    Min = min(overall_loudness),
    LF = quantile(overall_loudness, 0.25) - 1.5*IQR,
    UF = quantile(overall_loudness, .75) + 1.5*IQR,
    Max = max(overall_loudness)) |> #Puts all the stats in columns after to make it easier to look at
  mutate(out.of.range = ifelse(allentown.data$overall_loudness > Max | allentown.data$overall_loudness
  mutate(unusual = ifelse(allentown.data$overall_loudness > UF | allentown.data$overall_loudness < LF,
  mutate(description = case_when(out.of.range == "True" ~ "Out of Range",
                                 unusual == "True" ~ "Outlying",
                                 TRUE ~ "Within Range"))

compare = function(feature){#This function takes all the data and puts it into a data frame and returns
  artist.data = full.data |>
    group_by(artist) |>
    summarize(
      IQR = quantile(get(feature), .75, na.rm=T) - quantile(get(feature), 0.25, na.rm=T),
      Min = min(get(feature), na.rm=T),
      LF = quantile(get(feature), 0.25, na.rm=T) - 1.5*IQR,
      UF = quantile(get(feature), .75, na.rm=T) + 1.5*IQR,
      Max = max(get(feature))) |>
    mutate(out.of.range = ifelse(allentown.data[[feature]] > Max | allentown.data[[feature]] < Min, "Tru
    mutate(unusual = ifelse(allentown.data[[feature]] > UF | allentown.data[[feature]] < LF, "True", "Fa
    mutate(!!feature := case_when(out.of.range == "True" ~ "Out of Range",
                                  unusual == "True" ~ "Outlying",
                                  TRUE ~ "Within Range")) |>
    select(artist, !!sym(feature))
  return(artist.data)

}

#Step 2
#I need a for loop to show



#This makes a dataframe of all the numeric values
features = full.data |>
  select(-c("artist", "album","track","chords_scale","chords_key","key","mode")) |>
  colnames()

#This creates the data frame for the artists to merge
comparing.data = full.data |>
  group_by(artist) |>
  summarize() |>
  select(artist)

#This makes the features into one dataframe with the outlying compared data using the function
for(i in 1:length(features)){
  results = compare(features[i])
  comparing.data = comparing.data |>
    right_join(results, by = "artist", suffix = c("", ""))
}
```

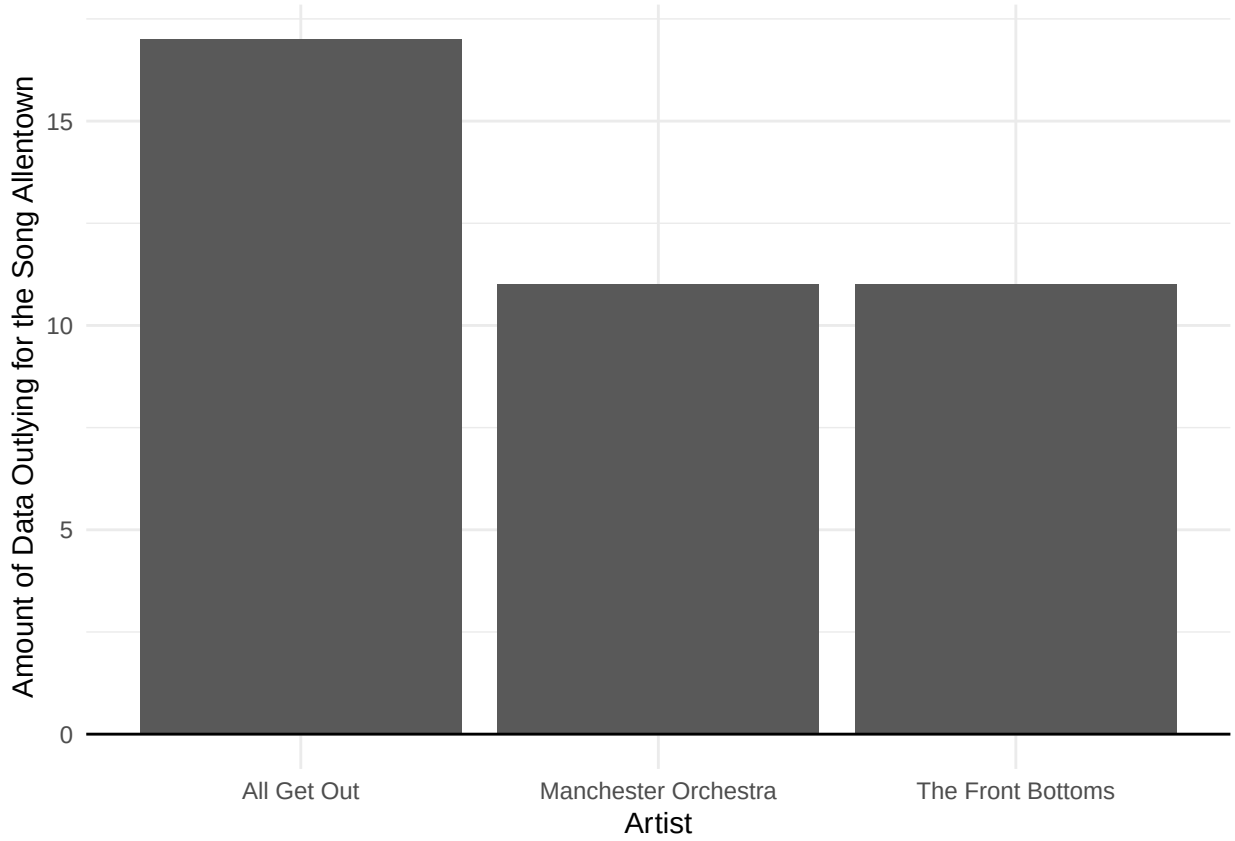
```r
#This is to start the count at 0 for these rows
comparing.data = comparing.data |>
  mutate(Within.Range = 0) |>
  mutate(Out.of.Range = 0) |>
  mutate(Outlying = 0)


#This is going to make a count for each of the rows for each thing it detects
for(feature in features){
  comparing.data = comparing.data |>
    rowwise() |>
    mutate(Within.Range = ifelse(get(feature) == "Within Range", Within.Range + 1, Within.Range)) |>
    mutate(Out.of.Range = ifelse(get(feature) == "Out of Range", Out.of.Range + 1, Out.of.Range)) |>
    mutate(Outlying = ifelse(get(feature) == "Outlying", Outlying + 1, Outlying))


}


#Step 4

#Data where it is in the range (In the IQR)
(within.range.bar.plot = ggplot(comparing.data)+
  geom_col(aes(x = artist, y = Within.Range))+
  geom_hline(yintercept = 0)+
  xlab("Artist")+
  ylab("Amount of Data Within Range for the song Allentown")+
    theme_bw())
```




```
#Data for out of range (It will be above the min or below the max)
(out.of.range.bar.plot = ggplot(comparing.data)+
    geom_col(aes(x = artist, y = Out.of.Range))+
    geom_hline(yintercept = 0)+
    xlab("Artist")+
    ylab("Amount of Data Out of Range for the Song Allentown")+
    theme_bw())
```




```
#Box plot for the outlying data (It is an outliar)
(outlying.bar.plot = ggplot(comparing.data)+
    geom_col(aes(x = artist, y = Outlying))+
    geom_hline(yintercept = 0)+
    xlab("Artist")+
    ylab("Amount of Data Outlying for the Song Allentown")+
    theme_minimal())
```




```
library(xtable)

#Step
#\begin{table}[ht]
#\centering
#\begin{tabular}{|c|ccc|}
#\hline
#Artist & Within Range & Out of Range & Outlying \\
#\hline
#All Get Out & 158.00 & 22.00 & 17.00 \\
#Manchester Orchestra & 183.00 & 3.00 & 11.00 \\
#The Front Bottoms & 156.00 & 30.00 & 11.00 \\
#\hline
#\end{tabular}
#\end{table}
#Commented out so code will still run
```