Additional Code Supplement

Jonathan Neimann

2024-12-10

Additional Code

Popularity means by genre

```
spotify_popularity %>%
  group_by(playlist_genre) %>%
  summarize(
    n = n(),
    avg_popularity = mean(track_popularity, na.rm = TRUE)
  ) %>%
  arrange(desc(avg_popularity))
```

```
## # A tibble: 6 × 3
##   playlist_genre     n avg_popularity
##   <chr>          <int>          <dbl>
## 1 latin           4601           52.6
## 2 pop             4994           52.5
## 3 rap             5040           49.1
## 4 rock            4209           48.7
## 5 r&b             4635           47.9
## 6 edm             5040           41.7
```

ANOVA Test

```
anova_result <- aov(track_popularity ~ as.factor(key), data =
spotify_popularity)
summary(anova_result)
```

```
##                   Df   Sum Sq Mean Sq F value   Pr(>F)
## as.factor(key)    11    21583  1962.1   4.666 3.67e-07 ***
## Residuals      28507 11987776   420.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Mean Square Error for models

```
# Calculate predictions for each model
original_predictions <- predict(lm_model, spotify_popularity)
interaction_predictions <- predict(lm_model_interaction, spotify_popularity)
genre_predictions <- predict(lm_model_genre, spotify_popularity)

# Calculate residuals for each model
original_residuals <- spotify_popularity$track_popularity -
original_predictions
```

```
interaction_residuals <- spotify_popularity$track_popularity -
interaction_predictions
genre_residuals <- spotify_popularity$track_popularity - genre_predictions

# Calculate MSE for each model
original_mse <- mean(original_residuals^2)
interaction_mse <- mean(interaction_residuals^2)
genre_mse <- mean(genre_residuals^2)

# Print the MSE values
cat("MSE for Original Model:", original_mse, "\n")

## MSE for Original Model: 394.5611

cat("MSE for Interaction Model:", interaction_mse, "\n")

## MSE for Interaction Model: 391.6193

cat("MSE for Model with Playlist Genre:", genre_mse, "\n")

## MSE for Model with Playlist Genre: 385.812
```

Mean vs Varience for poisson

```
mean_popularity <- mean(spotify_popularity$track_popularity)
variance_popularity <- var(spotify_popularity$track_popularity)

# Print the results
cat("Mean of Track Popularity:", mean_popularity, "\n")

## Mean of Track Popularity: 48.69273

cat("Variance of Track Popularity:", variance_popularity, "\n")

## Variance of Track Popularity: 421.115
```

Track Artist poportion on Varience

```
spotify_popularity <- spotify_popularity %>%
  mutate(across(c(energy, valence, loudness, instrumentalness, duration_ms,
danceability), scale))
# Calculate the overall variance in track_popularity
total_variance <- var(spotify_popularity$track_popularity)

# Calculate the mean popularity for each artist
artist_means <- aggregate(track_popularity ~ track_artist, data =
spotify_popularity, FUN = mean)

# Calculate variance between artists
between_variance <- var(artist_means$track_popularity)

# Calculate variance within artists
```

```
within_variance <- total_variance - between_variance

# Proportion of variance explained by artist
proportion_between <- between_variance / total_variance
proportion_within <- within_variance / total_variance

# Print results
cat("Total Variance:", total_variance, "\n")

## Total Variance: 421.115

cat("Between-Artist Variance:", between_variance, "\n")

## Between-Artist Variance: 262.2521

cat("Within-Artist Variance:", within_variance, "\n")

## Within-Artist Variance: 158.863

cat("Proportion of Variance Explained by Artist (Between):",
proportion_between, "\n")

## Proportion of Variance Explained by Artist (Between): 0.6227564
```

AIC & BIC Comparison

```
AIC(lm_model, lm_model_genre, mixed_model)

##                df      AIC
## lm_model        8 251429.6
## lm_model_genre 13 250800.0
## mixed_model    14 244922.0

BIC(lm_model, lm_model_genre, mixed_model)

##                df      BIC
## lm_model        8 251495.6
## lm_model_genre 13 250907.4
## mixed_model    14 245037.6
```