

# R Notebook

```
{r, warning = FALSE} library(readr) library(tidyverse) library(dplyr)
library(leaps) library(car) library(MASS)

{r, echo = FALSE} df_spotify <- read_csv("spotify-2023.csv")
#glimpse(df_spotify) #names(df_spotify)

1. Summary Statistics of variables selected for MLR

# Streams was originally of chr type. Changed it to dbl type.
df_spotify <- df_spotify %>% mutate(streams = as.numeric(streams))

par(mfrow=c(3,3))
hist(df_spotify$danceability_`)
hist(df_spotify$bpm)
hist(df_spotify$streams)
hist(df_spotify$`speechiness_`)
hist(df_spotify$`valence_`)
hist(df_spotify$`energy_`)
hist(df_spotify$`acousticness_`)
hist(df_spotify$`liveness_`)
hist(df_spotify$`instrumentalness_`)
# Remove this from OG model - data points don't make sense in context

# List your predictor columns exactly as they appear in your data
predictor_cols <- c("danceability_",
"bpm", "streams", "valence_%", "energy_%", "acousticness_%", "liveness_%", "speechiness_",
"mode_binary", "in_spotify_playlists"
)

# Creating mode_binary variable which is 1 if mode is Major and 0 for Minor.
df_spotify <- df_spotify %>% mutate(mode_binary = ifelse(mode == 'Major', 1, 0))

df_pred <- df_spotify %>%
  mutate(
    streams = as.numeric(gsub(", ", "", as.character(streams))),
    mode_binary = as.numeric(as.character(mode_binary))
  ) %>%
```

```

  select(all_of(predictor_cols))

summary_table <- map_dfr(names(df_pred), function(v) {
  x <- df_pred[[v]]
  tibble(
    variable = v,
    n      = sum(!is.na(x)),
    mean   = round(mean(x, na.rm = TRUE), 2),
    sd     = round(sd(x, na.rm = TRUE), 2),
    min    = round(suppressWarnings(min(x, na.rm = TRUE)), 2),
    max    = round(suppressWarnings(max(x, na.rm = TRUE)), 2)
  )
})

summary_table

```

## 2. Data Cleaning for MLR

```

# Creating mode_binary variable which is 1 if mode is Major and 0 for Minor.
df_spotify <- df_spotify %>% mutate(mode_binary = ifelse(mode == 'Major', 1, 0))

# Streams was originally of chr type. Changed it to dbl type.
df_spotify <- df_spotify %>% mutate(streams = as.numeric(streams))

# Note that in streams, observation #575 had an inconsistency: instead of displaying # of streams
df_spotify <- df_spotify %>% filter(track_name != "Love Grows (Where My Rosemary Goes)")

Justification for interaction term prior to fitting model

major <- df_spotify[df_spotify$mode_binary == 1, ]
minor <- df_spotify[df_spotify$mode_binary == 0, ]

# Note that danceability is response variable
par(mfrow=c(2,2))
boxplot(df_spotify$danceability_`~ df_spotify$mode_binary)

# plotting relationship between Danceability % and bpm for major/minor
plot(df_spotify$danceability_`~ df_spotify$bpm, xlab="Bpm", ylab="Danceability %", type="p")
points(major$danceability_`~ major$bpm, col="blue")
points(minor$danceability_`~ minor$bpm, col="black")
lines(lowess(major$danceability_`~ major$bpm), col="blue")
lines(lowess(minor$danceability_`~ minor$bpm), col="black")

# plotting relationship between Danceability % and `valence_` for major/minor
plot(df_spotify$danceability_`~ df_spotify$valence_`, xlab="`valence_`", ylab="Danceability %", type="p")
points(major$danceability_`~ major$valence_`, col="blue")
points(minor$danceability_`~ minor$valence_`, col="black")

```



```

par(mfrow=c(2,2))
# Residuals vs each predictor
plot(x = df_spotify$bpm, y = y_value, main="Residual vs Bpm", xlab="Bpm", ylab="Residual")
plot(x = df_spotify$streams, y = y_value, main="Residual vs Streams", xlab="Streams", ylab="Residual")
plot(x = df_spotify$`valence_%`, y = y_value, main="Residual vs Valence %", xlab="Valence %", ylab="Residual")
plot(x = df_spotify$`energy_%`, y = y_value, main="Residual vs Energy %", xlab="Energy %", ylab="Residual")

par(mfrow=c(2,2))
plot(x = df_spotify$`acousticness_%`, y = y_value, main="Residual vs Acousticness %", xlab="Acousticness %", ylab="Residual")
plot(x = df_spotify$`liveness_%`, y = y_value, main="Residual vs Liveness %", xlab="Liveness %", ylab="Residual")
plot(x = df_spotify$`speechiness_%`, y = y_value, main="Residual vs Speechiness %", xlab="Speechiness %", ylab="Residual")

par(mfrow=c(1,2))
plot(x = df_spotify$in_spotify_playlists, y = y_value, main="Residual vs in_spotify_playlist", xlab="in_spotify_playlist", ylab="Residual")
# For categorical variable
boxplot(y_value ~ df_spotify$mode_binary, main="Residual vs Mode", xlab="Mode", ylab="Residual")

```

## HOW TO FIX:

1. Check Multi-collinearity among predictors
2. Check points that simply don't make sense from descriptive stats
3. Transformations
4. Nested models

#1. Multi collinearity

```

# x includes all predictors in original model: in_spotify_playlists, streams, bpm, valence_% , energy_% , acousticness_% , speechiness_% , liveness_% , mode_binary
x <- cbind(df_spotify[, 7], df_spotify[, 9], df_spotify[, 15], df_spotify[, 19:21], df_spotify[, 23:25], df_spotify[, 27:29], df_spotify[, 31:33], df_spotify[, 35:37], df_spotify[, 39:41], df_spotify[, 43:45], df_spotify[, 47:49], df_spotify[, 51:53], df_spotify[, 55:57], df_spotify[, 59:61], df_spotify[, 63:65], df_spotify[, 67:69], df_spotify[, 71:73], df_spotify[, 75:77], df_spotify[, 79:81], df_spotify[, 83:85], df_spotify[, 87:89], df_spotify[, 91:93], df_spotify[, 95:97], df_spotify[, 99:101], df_spotify[, 103:105], df_spotify[, 107:109], df_spotify[, 111:113], df_spotify[, 115:117], df_spotify[, 119:121], df_spotify[, 123:125], df_spotify[, 127:129], df_spotify[, 131:133], df_spotify[, 135:137], df_spotify[, 139:141], df_spotify[, 143:145], df_spotify[, 147:149], df_spotify[, 151:153], df_spotify[, 155:157], df_spotify[, 159:161], df_spotify[, 163:165], df_spotify[, 167:169], df_spotify[, 171:173], df_spotify[, 175:177], df_spotify[, 179:181], df_spotify[, 183:185], df_spotify[, 187:189], df_spotify[, 191:193], df_spotify[, 195:197], df_spotify[, 199:201], df_spotify[, 203:205], df_spotify[, 207:209], df_spotify[, 211:213], df_spotify[, 215:217], df_spotify[, 219:221], df_spotify[, 223:225], df_spotify[, 227:229], df_spotify[, 231:233], df_spotify[, 235:237], df_spotify[, 239:241], df_spotify[, 243:245], df_spotify[, 247:249], df_spotify[, 251:253], df_spotify[, 255:257], df_spotify[, 259:261], df_spotify[, 263:265], df_spotify[, 267:269], df_spotify[, 271:273], df_spotify[, 275:277], df_spotify[, 279:281], df_spotify[, 283:285], df_spotify[, 287:289], df_spotify[, 291:293], df_spotify[, 295:297], df_spotify[, 299:301], df_spotify[, 303:305], df_spotify[, 307:309], df_spotify[, 311:313], df_spotify[, 315:317], df_spotify[, 319:319], df_spotify[, 321:321], df_spotify[, 323:323], df_spotify[, 325:325], df_spotify[, 327:327], df_spotify[, 329:329], df_spotify[, 331:331], df_spotify[, 333:333], df_spotify[, 335:335], df_spotify[, 337:337], df_spotify[, 339:339], df_spotify[, 341:341], df_spotify[, 343:343], df_spotify[, 345:345], df_spotify[, 347:347], df_spotify[, 349:349], df_spotify[, 351:351], df_spotify[, 353:353], df_spotify[, 355:355], df_spotify[, 357:357], df_spotify[, 359:359], df_spotify[, 361:361], df_spotify[, 363:363], df_spotify[, 365:365], df_spotify[, 367:367], df_spotify[, 369:369], df_spotify[, 371:371], df_spotify[, 373:373], df_spotify[, 375:375], df_spotify[, 377:377], df_spotify[, 379:379], df_spotify[, 381:381], df_spotify[, 383:383], df_spotify[, 385:385], df_spotify[, 387:387], df_spotify[, 389:389], df_spotify[, 391:391], df_spotify[, 393:393], df_spotify[, 395:395], df_spotify[, 397:397], df_spotify[, 399:399], df_spotify[, 401:401], df_spotify[, 403:403], df_spotify[, 405:405], df_spotify[, 407:407], df_spotify[, 409:409], df_spotify[, 411:411], df_spotify[, 413:413], df_spotify[, 415:415], df_spotify[, 417:417], df_spotify[, 419:419], df_spotify[, 421:421], df_spotify[, 423:423], df_spotify[, 425:425], df_spotify[, 427:427], df_spotify[, 429:429], df_spotify[, 431:431], df_spotify[, 433:433], df_spotify[, 435:435], df_spotify[, 437:437], df_spotify[, 439:439], df_spotify[, 441:441], df_spotify[, 443:443], df_spotify[, 445:445], df_spotify[, 447:447], df_spotify[, 449:449], df_spotify[, 451:451], df_spotify[, 453:453], df_spotify[, 455:455], df_spotify[, 457:457], df_spotify[, 459:459], df_spotify[, 461:461], df_spotify[, 463:463], df_spotify[, 465:465], df_spotify[, 467:467], df_spotify[, 469:469], df_spotify[, 471:471], df_spotify[, 473:473], df_spotify[, 475:475], df_spotify[, 477:477], df_spotify[, 479:479], df_spotify[, 481:481], df_spotify[, 483:483], df_spotify[, 485:485], df_spotify[, 487:487], df_spotify[, 489:489], df_spotify[, 491:491], df_spotify[, 493:493], df_spotify[, 495:495], df_spotify[, 497:497], df_spotify[, 499:499], df_spotify[, 501:501], df_spotify[, 503:503], df_spotify[, 505:505], df_spotify[, 507:507], df_spotify[, 509:509], df_spotify[, 511:511], df_spotify[, 513:513], df_spotify[, 515:515], df_spotify[, 517:517], df_spotify[, 519:519], df_spotify[, 521:521], df_spotify[, 523:523], df_spotify[, 525:525], df_spotify[, 527:527], df_spotify[, 529:529], df_spotify[, 531:531], df_spotify[, 533:533], df_spotify[, 535:535], df_spotify[, 537:537], df_spotify[, 539:539], df_spotify[, 541:541], df_spotify[, 543:543], df_spotify[, 545:545], df_spotify[, 547:547], df_spotify[, 549:549], df_spotify[, 551:551], df_spotify[, 553:553], df_spotify[, 555:555], df_spotify[, 557:557], df_spotify[, 559:559], df_spotify[, 561:561], df_spotify[, 563:563], df_spotify[, 565:565], df_spotify[, 567:567], df_spotify[, 569:569], df_spotify[, 571:571], df_spotify[, 573:573], df_spotify[, 575:575], df_spotify[, 577:577], df_spotify[, 579:579], df_spotify[, 581:581], df_spotify[, 583:583], df_spotify[, 585:585], df_spotify[, 587:587], df_spotify[, 589:589], df_spotify[, 591:591], df_spotify[, 593:593], df_spotify[, 595:595], df_spotify[, 597:597], df_spotify[, 599:599], df_spotify[, 601:601], df_spotify[, 603:603], df_spotify[, 605:605], df_spotify[, 607:607], df_spotify[, 609:609], df_spotify[, 611:611], df_spotify[, 613:613], df_spotify[, 615:615], df_spotify[, 617:617], df_spotify[, 619:619], df_spotify[, 621:621], df_spotify[, 623:623], df_spotify[, 625:625], df_spotify[, 627:627], df_spotify[, 629:629], df_spotify[, 631:631], df_spotify[, 633:633], df_spotify[, 635:635], df_spotify[, 637:637], df_spotify[, 639:639], df_spotify[, 641:641], df_spotify[, 643:643], df_spotify[, 645:645], df_spotify[, 647:647], df_spotify[, 649:649], df_spotify[, 651:651], df_spotify[, 653:653], df_spotify[, 655:655], df_spotify[, 657:657], df_spotify[, 659:659], df_spotify[, 661:661], df_spotify[, 663:663], df_spotify[, 665:665], df_spotify[, 667:667], df_spotify[, 669:669], df_spotify[, 671:671], df_spotify[, 673:673], df_spotify[, 675:675], df_spotify[, 677:677], df_spotify[, 679:679], df_spotify[, 681:681], df_spotify[, 683:683], df_spotify[, 685:685], df_spotify[, 687:687], df_spotify[, 689:689], df_spotify[, 691:691], df_spotify[, 693:693], df_spotify[, 695:695], df_spotify[, 697:697], df_spotify[, 699:699], df_spotify[, 701:701], df_spotify[, 703:703], df_spotify[, 705:705], df_spotify[, 707:707], df_spotify[, 709:709], df_spotify[, 711:711], df_spotify[, 713:713], df_spotify[, 715:715], df_spotify[, 717:717], df_spotify[, 719:719], df_spotify[, 721:721], df_spotify[, 723:723], df_spotify[, 725:725], df_spotify[, 727:727], df_spotify[, 729:729], df_spotify[, 731:731], df_spotify[, 733:733], df_spotify[, 735:735], df_spotify[, 737:737], df_spotify[, 739:739], df_spotify[, 741:741], df_spotify[, 743:743], df_spotify[, 745:745], df_spotify[, 747:747], df_spotify[, 749:749], df_spotify[, 751:751], df_spotify[, 753:753], df_spotify[, 755:755], df_spotify[, 757:757], df_spotify[, 759:759], df_spotify[, 761:761], df_spotify[, 763:763], df_spotify[, 765:765], df_spotify[, 767:767], df_spotify[, 769:769], df_spotify[, 771:771], df_spotify[, 773:773], df_spotify[, 775:775], df_spotify[, 777:777], df_spotify[, 779:779], df_spotify[, 781:781], df_spotify[, 783:783], df_spotify[, 785:785], df_spotify[, 787:787], df_spotify[, 789:789], df_spotify[, 791:791], df_spotify[, 793:793], df_spotify[, 795:795], df_spotify[, 797:797], df_spotify[, 799:799], df_spotify[, 801:801], df_spotify[, 803:803], df_spotify[, 805:805], df_spotify[, 807:807], df_spotify[, 809:809], df_spotify[, 811:811], df_spotify[, 813:813], df_spotify[, 815:815], df_spotify[, 817:817], df_spotify[, 819:819], df_spotify[, 821:821], df_spotify[, 823:823], df_spotify[, 825:825], df_spotify[, 827:827], df_spotify[, 829:829], df_spotify[, 831:831], df_spotify[, 833:833], df_spotify[, 835:835], df_spotify[, 837:837], df_spotify[, 839:839], df_spotify[, 841:841], df_spotify[, 843:843], df_spotify[, 845:845], df_spotify[, 847:847], df_spotify[, 849:849], df_spotify[, 851:851], df_spotify[, 853:853], df_spotify[, 855:855], df_spotify[, 857:857], df_spotify[, 859:859], df_spotify[, 861:861], df_spotify[, 863:863], df_spotify[, 865:865], df_spotify[, 867:867], df_spotify[, 869:869], df_spotify[, 871:871], df_spotify[, 873:873], df_spotify[, 875:875], df_spotify[, 877:877], df_spotify[, 879:879], df_spotify[, 881:881], df_spotify[, 883:883], df_spotify[, 885:885], df_spotify[, 887:887], df_spotify[, 889:889], df_spotify[, 891:891], df_spotify[, 893:893], df_spotify[, 895:895], df_spotify[, 897:897], df_spotify[, 899:899], df_spotify[, 901:901], df_spotify[, 903:903], df_spotify[, 905:905], df_spotify[, 907:907], df_spotify[, 909:909], df_spotify[, 911:911], df_spotify[, 913:913], df_spotify[, 915:915], df_spotify[, 917:917], df_spotify[, 919:919], df_spotify[, 921:921], df_spotify[, 923:923], df_spotify[, 925:925], df_spotify[, 927:927], df_spotify[, 929:929], df_spotify[, 931:931], df_spotify[, 933:933], df_spotify[, 935:935], df_spotify[, 937:937], df_spotify[, 939:939], df_spotify[, 941:941], df_spotify[, 943:943], df_spotify[, 945:945], df_spotify[, 947:947], df_spotify[, 949:949], df_spotify[, 951:951], df_spotify[, 953:953], df_spotify[, 955:955], df_spotify[, 957:957], df_spotify[, 959:959], df_spotify[, 961:961], df_spotify[, 963:963], df_spotify[, 965:965], df_spotify[, 967:967], df_spotify[, 969:969], df_spotify[, 971:971], df_spotify[, 973:973], df_spotify[, 975:975], df_spotify[, 977:977], df_spotify[, 979:979], df_spotify[, 981:981], df_spotify[, 983:983], df_spotify[, 985:985], df_spotify[, 987:987], df_spotify[, 989:989], df_spotify[, 991:991], df_spotify[, 993:993], df_spotify[, 995:995], df_spotify[, 997:997], df_spotify[, 999:999], df_spotify[, 1001:1001], df_spotify[, 1003:1003], df_spotify[, 1005:1005], df_spotify[, 1007:1007], df_spotify[, 1009:1009], df_spotify[, 1011:1011], df_spotify[, 1013:1013], df_spotify[, 1015:1015], df_spotify[, 1017:1017], df_spotify[, 1019:1019], df_spotify[, 1021:1021], df_spotify[, 1023:1023], df_spotify[, 1025:1025], df_spotify[, 1027:1027], df_spotify[, 1029:1029], df_spotify[, 1031:1031], df_spotify[, 1033:1033], df_spotify[, 1035:1035], df_spotify[, 1037:1037], df_spotify[, 1039:1039], df_spotify[, 1041:1041], df_spotify[, 1043:1043], df_spotify[, 1045:1045], df_spotify[, 1047:1047], df_spotify[, 1049:1049], df_spotify[, 1051:1051], df_spotify[, 1053:1053], df_spotify[, 1055:1055], df_spotify[, 1057:1057], df_spotify[, 1059:1059], df_spotify[, 1061:1061], df_spotify[, 1063:1063], df_spotify[, 1065:1065], df_spotify[, 1067:1067], df_spotify[, 1069:1069], df_spotify[, 1071:1071], df_spotify[, 1073:1073], df_spotify[, 1075:1075], df_spotify[, 1077:1077], df_spotify[, 1079:1079], df_spotify[, 1081:1081], df_spotify[, 1083:1083], df_spotify[, 1085:1085], df_spotify[, 1087:1087], df_spotify[, 1089:1089], df_spotify[, 1091:1091], df_spotify[, 1093:1093], df_spotify[, 1095:1095], df_spotify[, 1097:1097], df_spotify[, 1099:1099], df_spotify[, 1101:1101], df_spotify[, 1103:1103], df_spotify[, 1105:1105], df_spotify[, 1107:1107], df_spotify[, 1109:1109], df_spotify[, 1111:1111], df_spotify[, 1113:1113], df_spotify[, 1115:1115], df_spotify[, 1117:1117], df_spotify[, 1119:1119], df_spotify[, 1121:1121], df_spotify[, 1123:1123], df_spotify[, 1125:1125], df_spotify[, 1127:1127], df_spotify[, 1129:1129], df_spotify[, 1131:1131], df_spotify[, 1133:1133], df_spotify[, 1135:1135], df_spotify[, 1137:1137], df_spotify[, 1139:1139], df_spotify[, 1141:1141], df_spotify[, 1143:1143], df_spotify[, 1145:1145], df_spotify[, 1147:1147], df_spotify[, 1149:1149], df_spotify[, 1151:1151], df_spotify[, 1153:1153], df_spotify[, 1155:1155], df_spotify[, 1157:1157], df_spotify[, 1159:1159], df_spotify[, 1161:1161], df_spotify[, 1163:1163], df_spotify[, 1165:1165], df_spotify[, 1167:1167], df_spotify[, 1169:1169], df_spotify[, 1171:1171], df_spotify[, 1173:1173], df_spotify[, 1175:1175], df_spotify[, 1177:1177], df_spotify[, 1179:1179], df_spotify[, 1181:1181], df_spotify[, 1183:1183], df_spotify[, 1185:1185], df_spotify[, 1187:1187], df_spotify[, 1189:1189], df_spotify[, 1191:1191], df_spotify[, 1193:1193], df_spotify[, 1195:1195], df_spotify[, 1197:1197], df_spotify[, 1199:1199], df_spotify[, 1201:1201], df_spotify[, 1203:1203], df_spotify[, 1205:1205], df_spotify[, 1207:1207], df_spotify[, 1209:1209], df_spotify[, 1211:1211], df_spotify[, 1213:1213], df_spotify[, 1215:1215], df_spotify[, 1217:1217], df_spotify[, 1219:1219], df_spotify[, 1221:1221], df_spotify[, 1223:1223], df_spotify[, 1225:1225], df_spotify[, 1227:1227], df_spotify[, 1229:1229], df_spotify[, 1231:1231], df_spotify[, 1233:1233], df_spotify[, 1235:1235], df_spotify[, 1237:1237], df_spotify[, 1239:1239], df_spotify[, 1241:1241], df_spotify[, 1243:1243], df_spotify[, 1245:1245], df_spotify[, 1247:1247], df_spotify[, 1249:1249], df_spotify[, 1251:1251], df_spotify[, 1253:1253], df_spotify[, 1255:1255], df_spotify[, 1257:1257], df_spotify[, 1259:1259], df_spotify[, 1261:1261], df_spotify[, 1263:1263], df_spotify[, 1265:1265], df_spotify[, 1267:1267], df_spotify[, 1269:1269], df_spotify[, 1271:1271], df_spotify[, 1273:1273], df_spotify[, 1275:1275], df_spotify[, 1277:1277], df_spotify[, 1279:1279], df_spotify[, 1281:1281], df_spotify[, 1283:1283], df_spotify[, 1285:1285], df_spotify[, 1287:1287], df_spotify[, 1289:1289], df_spotify[, 1291:1291], df_spotify[, 1293:1293], df_spotify[, 1295:1295], df_spotify[, 1297:1297], df_spotify[, 1299:1299], df_spotify[, 1301:1301], df_spotify[, 1303:1303], df_spotify[, 1305:1305], df_spotify[, 1307:1307], df_spotify[, 1309:1309], df_spotify[, 1311:1311], df_spotify[, 1313:1313], df_spotify[, 1315:1315], df_spotify[, 1317:1317], df_spotify[, 1319:1319], df_spotify[, 1321:1321], df_spotify[, 1323:1323], df_spotify[, 1325:1325], df_spotify[, 1327:1327], df_spotify[, 1329:1329], df_spotify[, 1331:1331], df_spotify[, 1333:1333], df_spotify[, 1335:1335], df_spotify[, 1337:1337], df_spotify[, 1339:1339], df_spotify[, 1341:1341], df_spotify[, 1343:1343], df_spotify[, 1345:1345], df_spotify[, 1347:1347], df_spotify[, 1349:1349], df_spotify[, 1351:1351], df_spotify[, 1353:1353], df_spotify[, 1355:1355], df_spotify[, 1357:1357], df_spotify[, 1359:1359], df_spotify[, 1361:1361], df_spotify[, 1363:1363], df_spotify[, 1365:1365], df_spotify[, 1367:1367], df_spotify[, 1369:1369], df_spotify[, 1371:1371], df_spotify[, 1373:1373], df_spotify[, 1375:1375], df_spotify[, 1377:1377], df_spotify[, 1379:1379], df_spotify[, 1381:1381], df_spotify[, 1383:1383], df_spotify[, 1385:1385], df_spotify[, 1387:1387], df_spotify[, 1389:1389], df_spotify[, 1391:1391], df_spotify[, 1393:1393], df_spotify[, 1395:1395], df_spotify[, 1397:1397], df_spotify[, 1399:1399], df_spotify[, 1401:1401], df_spotify[, 1403:1403], df_spotify[, 1405:1405], df_spotify[, 1407:1407], df_spotify[, 1409:1409], df_spotify[, 1411:1411], df_spotify[, 1413:1413], df_spotify[, 1415:1415], df_spotify[, 1417:1417], df_spotify[, 1419:1419], df_spotify[, 1421:1421], df_spotify[, 1423:1423], df_spotify[, 1425:1425], df_spotify[, 1427:1427], df_spotify[, 1429:1429], df_spotify[, 1431:1431], df_spotify[, 1433:1433], df_spotify[, 1435:1435], df_spotify[, 1437:1437], df_spotify[, 1439:1439], df_spotify[, 1441:1441], df_spotify[, 1443:1443], df_spotify[, 1445:1445], df_spotify[, 1447:1447], df_spotify[, 1449:1449], df_spotify[, 1451:1451], df_spotify[, 1453:1453], df_spotify[, 1455:1455], df_spotify[, 1457:1457], df_spotify[, 1459:1459], df_spotify[, 1461:1461], df_spotify[, 1463:1463], df_spotify[, 1465:1465], df_spotify[, 1467:1467], df_spotify[, 1469:1469], df_spotify[, 1471:1471], df_spotify[, 1473:1473], df_spotify[, 1475:1475], df_spotify[, 1477:1477], df_spotify[, 1479:1479], df_spotify[, 1481:1481], df_spotify[, 1483:1483], df_spotify[, 1485:1485], df_spotify[, 1487:1487], df_spotify[, 1489:1489], df_spotify[, 1491:1491], df_spotify[, 1493:1493], df_spotify[, 1495:1495], df_spotify[, 1497:1497], df_spotify[, 1499:1499], df_spotify[, 1501:1501], df_spotify[, 1503:1503], df_spotify[, 1505:1505], df_spotify[, 1507:1507], df_spotify[, 1509:1509], df_spotify[, 1511:1511], df_spotify[, 1513:1513], df_spotify[, 1515:1515], df_spotify[, 1517:1517], df_spotify[, 1519:1519], df_spotify[, 1521:1521], df_spotify[, 1523:1523], df_spotify[, 1525:1525], df_spotify[, 1527:1527], df_spotify[, 1529:1529], df_spotify[, 1531:1531], df_spotify[, 1533:1533], df_spotify[, 1535:1535], df_spotify[, 1537:1537], df_spotify[, 1539:1539], df_spotify[, 1541:1541], df_spotify[, 1543:1543], df_spotify[, 1545:1545], df_spotify[, 1547:1547], df_spotify[, 1549:1549], df_spotify[, 1551:1551], df_spotify[, 1553:1553], df_spotify[, 1555:1555], df_spotify[, 1557:1557], df_spotify[, 1559:1559], df_spotify[, 1561:1561], df_spotify[, 1563:1563], df_spotify[, 1565:1565], df_spotify[, 1567:1567], df_spotify[, 1569:1569], df_spotify[, 1571:1571], df_spotify[, 1573:1573], df_spotify[, 1575:1575], df_spotify[, 1577:1577], df_spotify[, 1579:1579], df_spotify[, 1581:1581], df_spotify[, 1583:1583], df_spotify[, 1585:1585], df_spotify[, 1587:1587], df_spotify[, 1589:1589], df_spotify[, 1591:1591], df_spotify[, 1593:1593], df_spotify[, 1595:1595], df_spotify[, 1597:1597], df_spotify[, 1599:1599], df_spotify[, 1601:1601], df_spotify[, 1603:1603], df_spotify[, 1605:1605], df_spotify[, 1607:1607], df_spotify[, 1609:1609], df_spotify[, 1611:1611], df_spotify[, 1613:1613], df_spotify[, 1615:1615], df_spotify[, 1617:1617], df_spotify[, 1619:1619], df_spotify[, 1621:1621], df_spotify[, 1623:1623], df_spotify[, 1625:1625], df_spotify[, 1627:1627], df_spotify[, 1629:1629], df_spotify[, 1631:1631], df_spotify[, 1633:1633], df_spotify[, 1635:1635], df_spotify[, 1637:1637], df_spotify[, 1639:1639], df_spotify[, 1641:1641], df_spotify[, 1643:1643], df_spotify[, 1645:1645], df_spotify[, 1647:1647], df_spotify[, 1649:1649], df_spotify[, 1651:1651], df_spotify[, 1653:1653], df_spotify[, 1655:1655], df_spotify[, 1657:1657], df_spotify[, 1659:1659], df_spotify[, 1661:1661], df_spotify[, 1663:1663], df_spotify[, 1665:1665], df_spotify[, 1667:1667], df_spotify[, 1669:1669], df_spotify[, 1671:1671], df_spotify[, 1673:1673], df_spotify[, 1675:1675], df_spotify[, 1677:1677], df_spotify[, 1679:1679], df_spotify[, 1681:1681], df_spotify[, 1683:1683], df_spotify[, 1685:1685], df_spotify[, 1687:1687], df_spotify[, 1689:1689], df_spotify[, 1691:1691], df_spotify[, 1693:1693], df_spotify[, 1695:1695], df_spotify[, 1697:1697], df_spotify[, 1699:1699], df_spotify[, 1701:1701], df_spotify[, 1703:1703], df_spotify[, 1705:1705], df_spotify[, 1707:1707], df_spotify[, 1709:1709], df_spotify[, 1711:1711],
```

```

# leverage statistic
h_jj <- hatvalues(model)
# standardized residuals
r_j <- rstandard(model)
# cook's distance
D_j <- cooks.distance(model)
# dffits
dffits_j <- dffits(model)
# dfbetas
dfbetas_j <- dfbetas(model)

# number of predictors and observations
n <- nobs(model)
p <- length(coef(model)) - 1

# leverage cutoff
hcut <- 2 * ((p + 1) / n)
# cook's distance cutoff
cookcut <- qf(0.5, df1 = p + 1, df2 = n - p - 1)
# dffits cutoff
fitcut <- 2 * sqrt((p+1)/n)
# dfbeta cutoff
betacut <- 2 / (sqrt(n))

# which observations are leverage points?
which(h_jj > hcut)
# which observations are regression outliers?
which(r_j > 4 | r_j < -4)
# which observations are influential by cook's distance?
which(D_j > cookcut)
# which observations are influential by dffits?
which(abs(dffits_j) > fitcut)

# After initial check of problematic observations, we realized that high influential points
# NO RATIONALE TO REMOVE ANY SONGS.

#3. Transformation of Variables
Normality
p_box <- powerTransform(cbind(df_spotify[, 18], df_spotify[, 7], df_spotify[, 9], df_spotify[, 10]), method = "Box-Cox")
#summary(p_box)

# fix worst transformation first, check residual plots after every transformation, avoid too many
# 1. Normality

```

```

t_y <- (df_spotify$`danceability_%`)^2 # value of lambda obtained by BoxCox

model_1 <- lm(t_y ~ bpm + streams + `valence_%` + `energy_%` + `acousticness_%` + `liveness_`)

# Now we check residual plots

par(mfrow=c(2,2))
y_value_1 <- resid(model_1)
x_value_1 <- fitted(model_1)
plot(x = x_value_1, y = y_value_1, main="Residual vs Fitted Model 1", xlab="Fitted", ylab="Residual")

# Residuals vs each predictor
plot(x = df_spotify$bpm, y = y_value_1, main="Residual vs Bpm", xlab="Bpm", ylab="Residual")
plot(x = df_spotify$streams, y = y_value_1, main="Residual vs Streams", xlab="Streams", ylab="Residual")
plot(x = df_spotify$`valence_%`, y = y_value_1, main="Residual vs Valence %", xlab="Valence %")

par(mfrow=c(2,2))
plot(x = df_spotify$`energy_%`, y = y_value_1, main="Residual vs Energy %", xlab="Energy %")

plot(x = df_spotify$`acousticness_%`, y = y_value_1, main="Residual vs Acousticness %", xlab="Acousticness %")
plot(x = df_spotify$`liveness_%`, y = y_value_1, main="Residual vs Liveness %", xlab="Liveness %")
plot(x = df_spotify$`speechiness_%`, y = y_value_1, main="Residual vs Speechiness %", xlab="Speechiness %")

par(mfrow=c(2,2))
plot(x = df_spotify$in_spotify_playlists, y = y_value_1, main="Residual vs in_spotify_playlists", xlab="in_spotify_playlists", ylab="Residual")

# For categorical variable
boxplot(y_value ~ df_spotify$mode_binary, main="Residual vs Mode", xlab="Mode", ylab="Residual")

# Check normality just in case
# QQ Normality plots
qqnorm(resid(model_1), main = "Normal Q-Q Plot")
qqline(resid(model_1))

# random noise at tails
Constant Variance
In_spotify_playlist

# From Residual plots, we see that multiple predictors exhibit non constant variance. First we will log the variable

t_playlist <- log(df_spotify$in_spotify_playlists)

model_2 <- lm(t_y ~ bpm + streams + `valence_%` + `energy_%` + `acousticness_%` + `liveness_`)

# Now we check for residuals

```

```

par(mfrow=c(2,2))
y_value_1 <- resid(model_2)
x_value_1 <- fitted(model_2)
plot(x = x_value_1, y = y_value_1, main="Residual vs Fitted Model 1", xlab="Fitted", ylab="Residual")

# Residuals vs each predictor
plot(x = df_spotify$bpm, y = y_value_1, main="Residual vs Bpm", xlab="Bpm", ylab="Residual")
plot(x = df_spotify$streams, y = y_value_1, main="Residual vs Streams", xlab="Streams", ylab="Residual")
plot(x = df_spotify$`valence_%`, y = y_value_1, main="Residual vs Valence %", xlab="Valence %")

par(mfrow=c(2,2))
plot(x = df_spotify$`energy_%`, y = y_value_1, main="Residual vs Energy %", xlab="Energy %")

plot(x = df_spotify$`acousticness_%`, y = y_value_1, main="Residual vs Acousticness %", xlab="Acousticness %")
plot(x = df_spotify$`liveness_%`, y = y_value_1, main="Residual vs Liveness %", xlab="Liveness %")
plot(x = df_spotify$`speechiness_%`, y = y_value_1, main="Residual vs Speechiness %", xlab="Speechiness %")

par(mfrow=c(2,2))
plot(x = t_playlist, y = y_value_1, main="Residual vs in_spotify_playlists", xlab="in_spotify_playlists")
# For categorical variable
boxplot(y_value ~ df_spotify$mode_binary, main="Residual vs Mode", xlab="Mode", ylab="Residual")

# Check normality just in case
# QQ Normality plots
qqnorm(resid(model_2), main = "Normal Q-Q Plot")
qqline(resid(model_2))

Streams

t_streams <- log(df_spotify$streams) # boxcox said 0.14 and log looks like cluster so better

model_3 <- lm(t_y ~ bpm + t_streams + `valence_%` + `energy_%` + `acousticness_%` + `liveness_%`)

# Now we check for residuals

par(mfrow=c(2,2))
y_value_1 <- resid(model_3)
x_value_1 <- fitted(model_3)
plot(x = x_value_1, y = y_value_1, main="Residual vs Fitted Model 1", xlab="Fitted", ylab="Residual")

# Residuals vs each predictor
plot(x = df_spotify$bpm, y = y_value_1, main="Residual vs Bpm", xlab="Bpm", ylab="Residual")
plot(x = t_streams, y = y_value_1, main="Residual vs Streams", xlab="Streams", ylab="Residual")
plot(x = df_spotify$`valence_%`, y = y_value_1, main="Residual vs Valence %", xlab="Valence %")

par(mfrow=c(2,2))
plot(x = df_spotify$`energy_%`, y = y_value_1, main="Residual vs Energy %", xlab="Energy %")

```

```

plot(x = df_spotify$`acousticness_%`, y = y_value_1, main="Residual vs Acousticness %", xlab="Acousticness %", ylab="Residual")
plot(x = df_spotify$`liveness_%`, y = y_value_1, main="Residual vs Liveness %", xlab="Liveness %", ylab="Residual")
plot(x = df_spotify$`speechiness_%`, y = y_value_1, main="Residual vs Speechiness %", xlab="Speechiness %", ylab="Residual")

par(mfrow=c(2,2))
plot(x = t_playlist, y = y_value_1, main="Residual vs in_spotify_playlists", xlab="in_spotify_playlists", ylab="Residual")
# For categorical variable
boxplot(y_value ~ df_spotify$mode_binary, main="Residual vs Mode", xlab="Mode", ylab="Residual")

# Check normality just in case
# QQ Normality plots
qqnorm(resid(model_3), main = "Normal Q-Q Plot")
qqline(resid(model_3))

```

## Recall from week 9 - BEST model selection

```

best <- regsubsets(t_y ~ bpm + t_streams + `valence_%` + `energy_%` + `acousticness_%` + `liveness_%` + `speechiness_%` + `liveness_%` * `speechiness_%`)

summary(best)
#4. Nested Models

summary(model_3) # By looking at summary, we see that t_streams, and interaction term are not significant

# Thus, we build reduced model without those
# Further supported by best selection model

model_3_reduced <- lm(t_y ~ bpm + t_playlist + `valence_%` + `energy_%` + `acousticness_%` + `liveness_%` + `speechiness_%` + `liveness_%` * `speechiness_%`)

anova(model_3_reduced, model_3)
# small p value = drop = full model better
# large p value = reduced model better

summary(model_3_reduced)
# Now we check for residuals

par(mfrow=c(2,2))
y_value_1 <- resid(model_3_reduced)
x_value_1 <- fitted(model_3_reduced)
plot(x = x_value_1, y = y_value_1, main="Residual vs Fitted Model 1", xlab="Fitted", ylab="Residual")

# Residuals vs each predictor
plot(x = df_spotify$bpm, y = y_value_1, main="Residual vs Bpm", xlab="Bpm", ylab="Residual")

```

```

plot(x = t_streams, y = y_value_1, main="Residual vs Streams", xlab="Streams", ylab="Residual")
plot(x = df_spotify$valence_%, y = y_value_1, main="Residual vs Valence %", xlab="Valence %")

par(mfrow=c(2,2))
plot(x = df_spotify$energy_%, y = y_value_1, main="Residual vs Energy %", xlab="Energy %")

plot(x = df_spotify$acousticness_%, y = y_value_1, main="Residual vs Acousticness %", xlab="Acousticness %")
plot(x = df_spotify$liveness_%, y = y_value_1, main="Residual vs Liveness %", xlab="Liveness %")
plot(x = df_spotify$speechiness_%, y = y_value_1, main="Residual vs Speechiness %", xlab="Speechiness %")

par(mfrow=c(2,2))
plot(x = t_playlist, y = y_value_1, main="Residual vs in_spotify_playlists", xlab="in_spotify_playlists")
# For categorical variable
boxplot(y_value ~ df_spotify$mode_binary, main="Residual vs Mode", xlab="Mode", ylab="Residual")

# Check normality just in case
# QQ Normality plots
qqnorm(resid(model_3_reduced), main = "Normal Q-Q Plot")
qqline(resid(model_3_reduced))

```

## Model Selection

```

# Full transformed model (same as model_3)
full_model <- lm(
  t_y ~ bpm + t_streams + `valence_%` + `energy_%` +
    `acousticness_%` + `liveness_%` + `speechiness_%` +
    mode_binary + t_playlist + mode_binary*`speechiness_%` ,
  df_spotify
)

# Null model, intercept only
null_model <- lm(t_y ~ 1, data = df_spotify)

# Forward Selection with AIC
forward_model <- stepAIC(
  null_model,
  scope = list(
    lower = null_model,
    upper = full_model
  ),
  direction = "forward",
  trace = TRUE
)

summary(forward_model)

```

```

# Backward

backward_model <- stepAIC(
  full_model,
  direction = "backward",
  trace = TRUE
)

summary(backward_model)

# Stepwise

stepwise_model <- stepAIC(
  full_model,
  direction = "both",
  trace = TRUE
)

summary(stepwise_model)

AIC(full_model, forward_model, backward_model, stepwise_model)

BIC(full_model, forward_model, backward_model, stepwise_model)

c(
  full = summary(full_model)$adj.r.squared,
  forward = summary(forward_model)$adj.r.squared,
  backward = summary(backward_model)$adj.r.squared,
  stepwise = summary(stepwise_model)$adj.r.squared
)

# all selection algorithms give same model as each other and as the anova test. and has better

```