Assignment

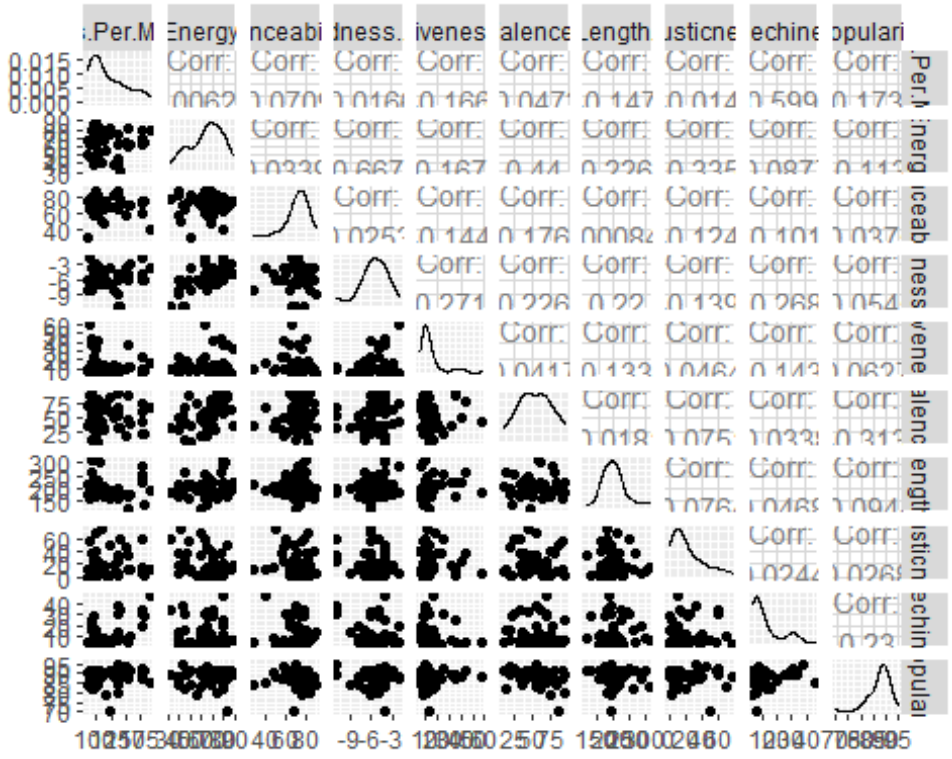
Introduction

The dataset contains the most popular songs TOP 50 in 2019, which can be downloaded in Kaggle webset (https:https://www.kaggle.com/leonardopena/top50spotify2019). The dataset includes 13 columns in which 3 string variables and 10 numerical variables. Variables in the dataset is shown as:

- Track.Name: Name of the songs
- Artist.Name: Name of singer
- Genre: Type of songs
- Beats.Per.Minute: The tempo of the song.
- Energy: The energy of a song - the higher the value, the more energtic.
- Danceability: The higher the value, the easier it is to dance to this song.
- Loudness..dB..: The higher the value, the louder the song.
- Liveness: The higher the value, the more likely the song is a live recording.
- Valence.: The higher the value, the more positive mood for the song.
- Length.: The duration of the song.
- Acousticness..: The higher the value the more acoustic the song is.
- Speechiness.: The higher the value the more spoken word the song contains.
- Popularity: The higher the value the more popular the song is.

Objective in this report is to predict popularity of a song by numerical features in this dataset. 2 NAs are existed in this dataset removed.

Exploratory Analysis



According to the paired scatter plot and density plot in the diagonal, it can be found that almost no outliers in the dataset. Besides, response variable, is almost normally distributed with a little left-skewed, which is not need to transformation.

Construction and selection for OLS model

Criterias for subsets models

Model	MSE	AIC	BIC	Cp	Adjr
Best model of one predictor	16.84	-0.96	2.79	-1.91	0.08
Best model of two predictors	15.94	-1.59	4.02	-2.17	0.11
Best model of three predictors	15.71	-0.28	7.20	-0.73	0.10
Best model of four predictors	15.42	0.82	10.17	0.54	0.10
Best model of five predictors	15.29	2.40	13.62	2.20	0.08
Best model of six predictors	15.21	4.16	17.26	4.01	0.07
Best model of seven predictors	15.21	6.15	21.11	6.00	0.04
Best model of eight predictors	15.21	8.14	24.98	8.00	0.02
Best model of nine predictors	15.21	10.14	28.85	10.00	-0.01

In this section, all subset regression are searched exhaustively and MSE, AIC, BIC, Mallow's Cp as well as Adjusted R^2 are also calculated to evaluate these subset models. Finally, best models are selected from one-predictor to full-predictors thus 9 model are constructed as shown above. Due to AIC and Adjusted R^2 are prefer to model with two predictors, as well as MSE is decrease quickly while number of predictors less than 2 but slowly while it more than 2. Thus, this model is selected as final OLS model with predictors as Valence. and Speechiness.. Thus final model is shown as:

```
##
## Call:
## lm(formula = Popularity ~ Valence. + Speechiness., data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1380  -1.7692   0.6544   2.2282   6.1851
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  89.81491    1.72212  52.154   <2e-16 ***
## Valence.     -0.05960    0.02686  -2.219   0.0316 *
## Speechiness.  0.08474    0.05313   1.595   0.1178
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.124 on 45 degrees of freedom
## Multiple R-squared:  0.1463, Adjusted R-squared:  0.1084
## F-statistic: 3.857 on 2 and 45 DF,  p-value: 0.02844
```

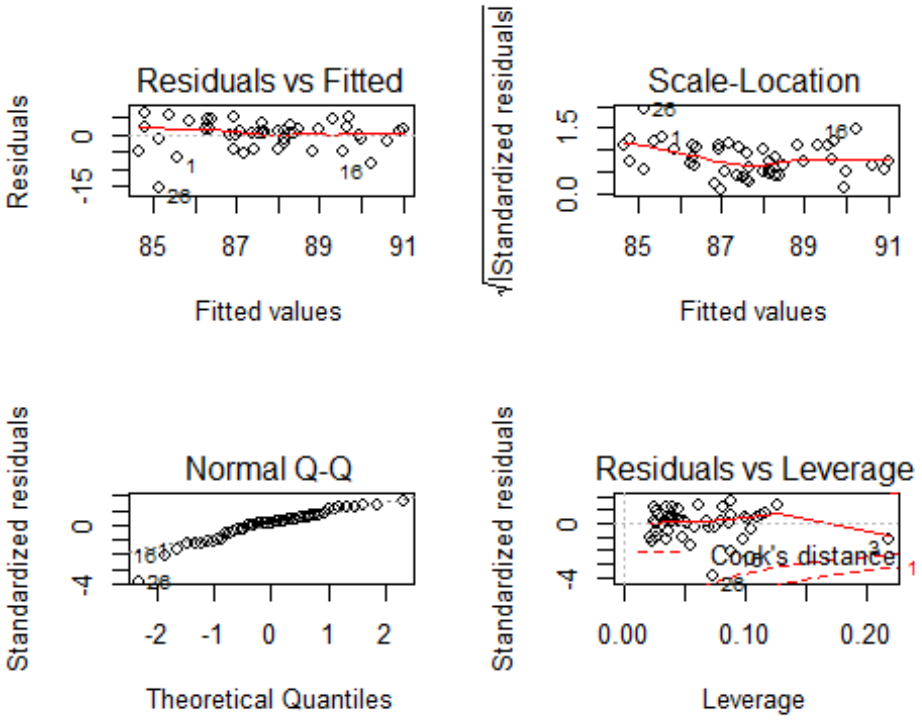
Model diagnose

To reveal whether the best OLS model is as well as full OLS model, ANOVA is performed, the result is shown as:

ANOVA for full and reduced models

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
38	729.91				
45	765.15	-7	-35.24	0.26	0.96

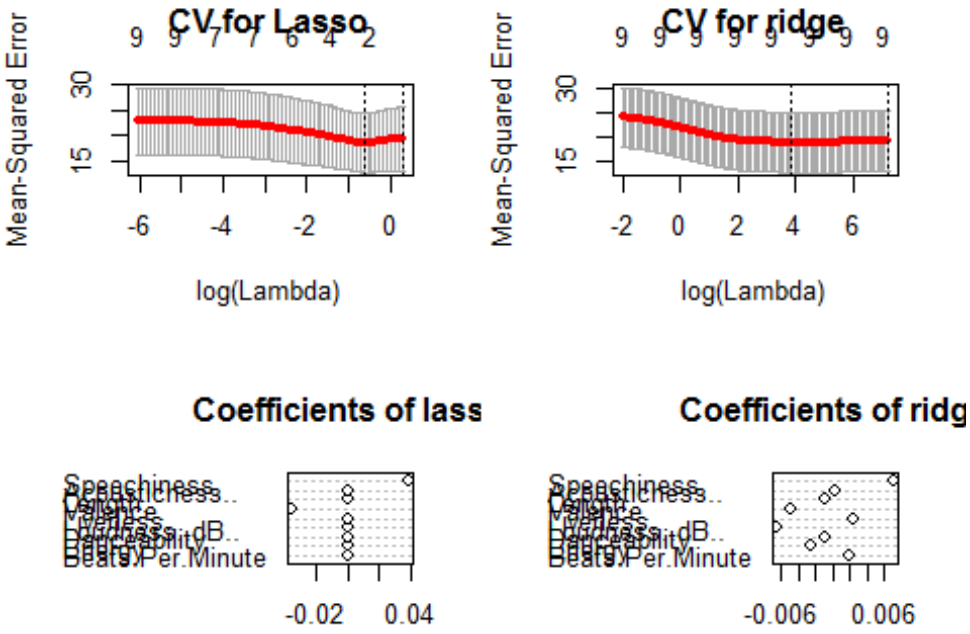
The P value is more than 0.05, which indicates that the godness of fitting and variance explaining of model with two predictors is not different with full model. Diagnose plots for assumption of linear regression model are also visualized as well as influent and high leverage points.




```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  scale(mod.ols$residuals)
## D = 0.16865, p-value = 0.1159
## alternative hypothesis: two-sided
```

The first figure indicates that no obvious trend is existed between fitted value and residuals except 26th song, which indicates that linear assumption is almost satisfied except 26th song. Similarly, trend line for fitted value vesus squared root of standardized residuals is also a horizontal straight line, which indicates that homoscedasticity assumption is also satisfied almost. Scatters in Normal Q-Q plot are near to the straight line, which represent that normality assimption is also most satisfided. Normality assumption is also tested by KS-test as shown above, in which P value is more than 0.05, which indicates that residuals are really normaly distributed. The last figure indicates that the third song with high leverage but not large residual. The 26th songs are near to influent point, of which cook's distance is close to 0.5.

Comparison with Lasso and Ridge



According to the figures, λ parameters are selected by gird search and 10 folds cross validation. The λ is selected while MSE reach lowest in corss validation. The coefficients of models are shown in dotcharts. In lasso regression, the two predictors retained is Speechiness and Valence which is consistent with OLS model. As for ridge regression model, top 3 predictors of absolute coefficient are Speechiness, Loudness and Valence.

Conclusion

According to the models, the best important predictor used to predict popularity of a song in 2019 are Valence and Speechiness, in which coefficient of Valence is less than 0 and coefficient of Speechiness is larger tha 0. Thus a popular songs should include spoken words. However, coefficient of Valence is less than 0, which represent popular songs should not enriched with positive mood. In other words, sadness or angry may make songs more artistic. Loudness is important in ridge regression, of which coefficient is less than 0. Thus singing a popular song should not too loudly.

Appendix

```
# exploatory analysis
library(GGally)
```

```
df<-read.csv('top50.csv')
df<-na.omit(df)
ggpairs(df[,4:13])

# OLS model constructed and selection
library(leaps)
# First function: Combine subset regression build and calculating all
criteria
best_select<-function(formula,data,...){
    mod.set<-regsubsets(formula,data=data,...)
    crit<-summary(mod.set)
    k<-apply(crit$which,1,sum)
    crit$aic<-crit$bic-k*(log(nrow(data))-2)
    crit$mse<-crit$rss/nrow(data)

    comp_tab<-data.frame(Model=c('Best model of one predictor',
                                 'Best model of two predictors',
                  'Best model of three predictors',
                  'Best model of four predictors',
                  'Best model of five predictors','Best model of six
predictors',
                  'Best model of seven predictors',
                  'Best model of eight predictors',
                  'Best model of nine predictors'),
                  MSE=crit$mse,AIC=crit$aic,BIC=crit$bic,
                  Cp=crit$cp,Adjr=crit$adjr2)
    return(list(comp_tab=comp_tab,
                mod_list=crit$which))
}

res<-
best_select(Popularity~.,data=df[,4:13],method="exhaustive",nbest=1,nvm
ax=9)
knitr::kable(res$comp_tab,caption = 'Criterias for subsets
models',digits = 2)

# F-test
options(knitr.kable.NA = '')
knitr::kable(anova(full.ols,mod.ols),
             caption = 'ANOVA for full and reduced models',digits = 2)

# Diagnose plots
layout(matrix(1:4,nrow=2))
plot(mod.ols)
# KS test
ks.test(scale(mod.ols$residuals),pnorm)

# Lasso regression and Ridge regression
library(glmnet)
```

```
set.seed(1234)
X<-model.matrix(formula,data=df[,4:13])[,-1]
y<-df$Popularity
# second function: combine build lasso and ridge regression as well as
# cross validation and visualization of CV and coefficient
glmnet_comb<-function(X,y) {
     cv.lasso<-cv.glmnet(X,y)
     cv.ridge<-cv.glmnet(X,y,alpha=0)
     fit.lasso<-glmnet(X,y,lambda=cv.lasso$lambda.min)
     fit.ridge<-glmnet(X,y,alpha=0,lambda=cv.ridge$lambda.min)
     layout(matrix(1:4,nrow=2,byrow=T))
     plot(cv.lasso,main='CV for Lasso')
     plot(cv.ridge,main='CV for ridge')
     beta.lasso<-as.numeric(fit.lasso$beta)
     names(beta.lasso)<-fit.lasso$beta@Dimnames[[1]]
     beta.ridge<-as.numeric(fit.ridge$beta)
     names(beta.ridge)<-fit.ridge$beta@Dimnames[[1]]
     dotchart(beta.lasso,main='Coefficients of lasso')
     dotchart(beta.ridge,main='Coefficients of ridge')
     return(list(fit.lasso=fit.lasso,
                 fit.ridge=fit.ridge))
}
res<-glmnet_comb(X,y)
```