Spotify 1: Intial Models for Predicting Number of Playlist Followers

Main conclusions:

- Adding audio attributes generally did not make a substantial difference in predicting number of playlist followers. Duration and time signature may have some statistically significant predictive contribution.
- Non-audio attributes (e.g the date songs were added, the popularity of the individual songs, total number of tracks, and whether the songs were "featured" by Spotify) are most significant in predicting number of followers.
- We see a reasonable model fit for predicting the log of the # of playlist followers: R^2 = 0.48

Read in the Data and drop unnecessary fields

```
train <- read.csv("/Users/lware/Harvard/spotify/capstone/playlist_data_with_audio_attributes_and_featur
                  header=TRUE,sep=',')
library(mgcv)
```

```
## Loading required package: nlme

## This is mgcv 1.8-15. For overview type 'help("mgcv-package")'.
```

```
dim(train)
```

```
## [1] 1720   21
```

```
drops <-c("names", "playlist_id","X")
train$added_date = as.Date(train$added_date)
train = train[ , !(names(train) %in% drops)]

# Add a field for log of followers, which will be a more appropriate response variable.
train$logfollowers = log(train$followers)
train$logfollowers[train$logfollowers<0]=0
train = na.omit(train)
dim(train)
```

```
## [1] 1634   19
```

```
str(train)
```

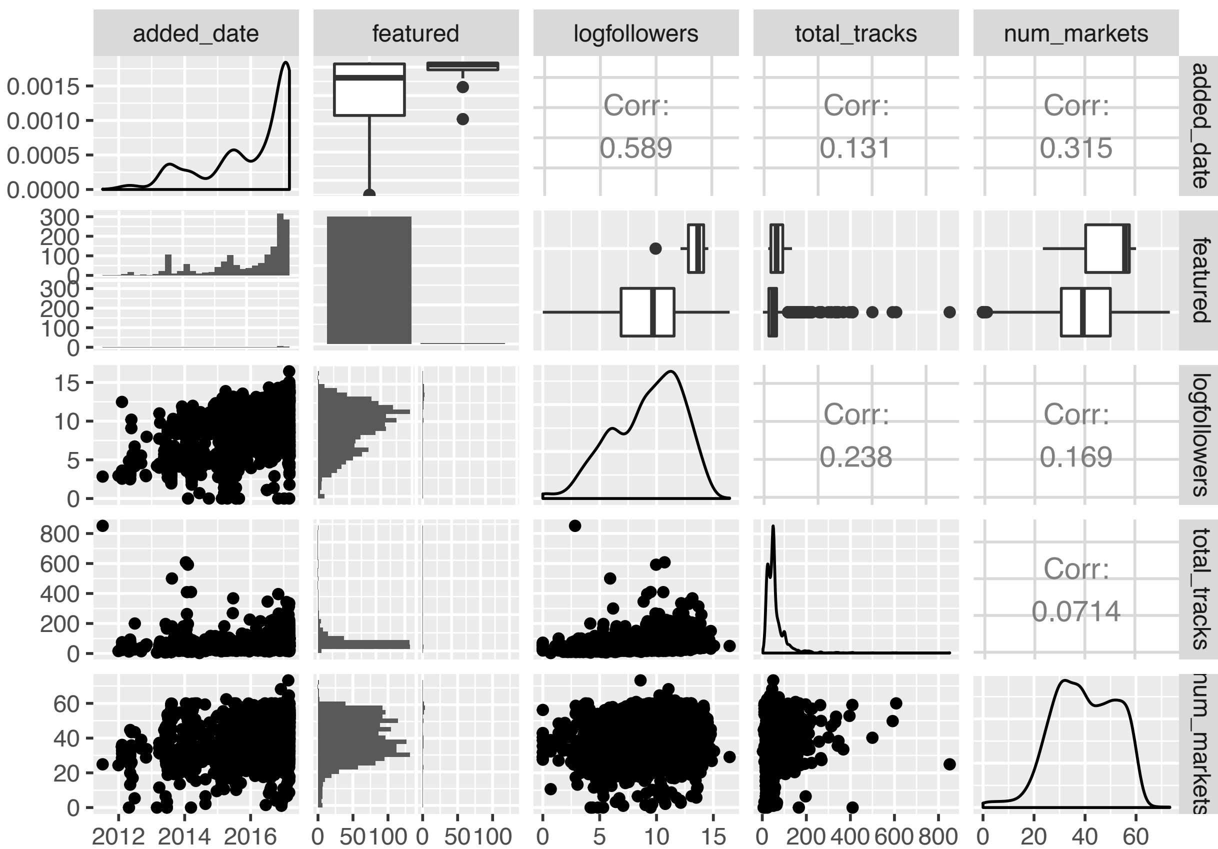
```
## 'data.frame':    1634 obs. of  19 variables:
##  $ acousticness    : num  0.2524 0.1989 0.1888 0.0962 0.1105 ...
##  $ added_date      : Date, format: "2016-12-09" "2016-08-10" ...
##  $ danceability    : num  0.552 0.595 0.643 0.684 0.699 ...
##  $ duration        : num  249831 228684 213474 213280 213630 ...
##  $ energy          : num  0.662 0.674 0.685 0.792 0.781 ...
##  $ featured        : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
##  $ followers       : int  899 7856 79961 17245 87715 345544 527653 170263 230130 432974 ...
##  $ instrumentalness: num  0.0123 0.07285 0.01586 0.00781 0.06818 ...
##  $ key             : num  5.53 4.95 5.66 5.31 4.93 ...
##  $ liveness        : num  0.192 0.167 0.157 0.184 0.194 ...
##  $ loudness        : num  -6.33 -7.84 -6.02 -4.96 -4.92 ...
```

```
##  $ mean_popularity : num  53.9 29.7 59.4 67.8 64.7 ...
##  $ mode            : num  0.673 0.723 0.57 0.56 0.543 ...
##  $ num_markets     : num  56.6 37.7 47.5 44.9 46 ...
##  $ tempo           : num  117 120 114 120 121 ...
##  $ time_signature  : num  3.9 3.96 4 3.99 4 ...
##  $ total_tracks    : int  49 83 138 135 46 40 60 53 40 60 ...
##  $ valence         : num  0.441 0.616 0.481 0.573 0.551 ...
##  $ logfollowers    : num  6.8 8.97 11.29 9.76 11.38 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:86] 49 53 115 134 153 154 165 166 210 212 ...
##   .. ..- attr(*, "names")= chr [1:86] "49" "53" "115" "134" ...
```

Exploration of relationships between features

```r
library(GGally)
subset = train[c(2,6,19,17, 14)]
ggpairs(subset)
```

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



Create some models and make predictions

```
library(ggplot2)

formula.1 = as.formula(paste0("followers ~ s(total_tracks) + s(num_markets) + energy + loudness +
                              s(mean_popularity) + danceability"))

formula.2 = as.formula(paste0("followers ~ s(acousticness) +
                              s(danceability) +
                              s(duration) +
                              s(energy) +
                              s(instrumentalness) +
                              s(key) +
                              s(liveness) +
                              s(loudness) +
                              s(mean_popularity) +
                              s(mode) +
                              s(num_markets) +
                              s(tempo) +
                              s(time_signature) +
                              s(total_tracks) +
                              s(valence)"))

formula.3 = as.formula(paste0("logfollowers ~ acousticness + added_date +
                              featured +
                              danceability +
                              duration +
                              energy +
                              instrumentalness +
                              key +
                              liveness +
                              loudness +
                              s(mean_popularity) +
                              mode +
                              s(num_markets) +
                              tempo +
                              time_signature +
                              s(total_tracks) +
                              valence"))

formula.4 = as.formula(paste0("logfollowers ~ added_date + featured + time_signature + duration +
                              s(mean_popularity) +
                              s(total_tracks)"))

formula.5 = as.formula(paste0("logfollowers ~ added_date + featured +
                              s(mean_popularity) +
                              s(total_tracks)"))

rsq = function(model, data, y) {
  y <- data[[y]]
  predict <- predict(model, newdata = data)
  predict[predict<0] = 0
  tss = sum((y - mean(y))^2)
```

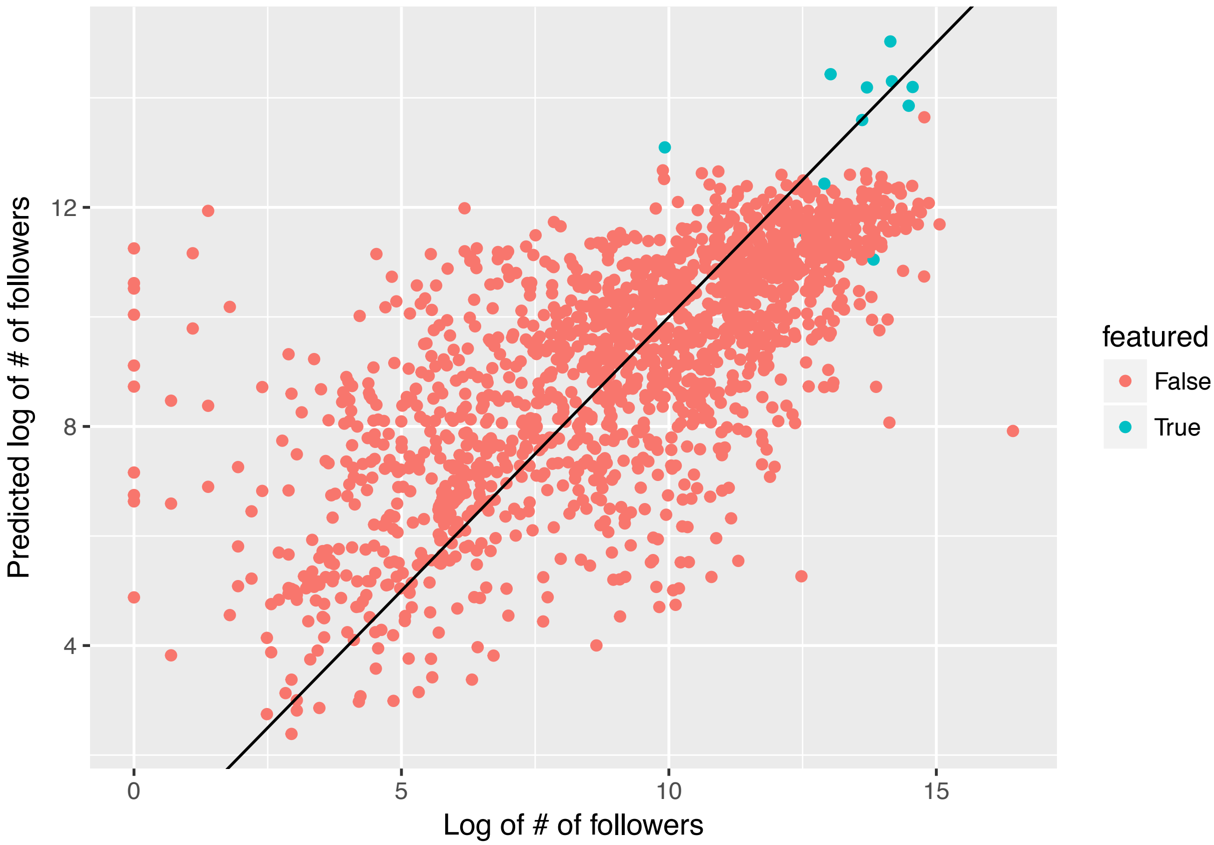
```r
  rss = sum((y-predict)^2)
  rsq_ = max(0, 1 - rss/tss)
  return(rsq_)
}

gam.results = function(form) {
  model = gam(form, data=train)
  cat("Train R^2: ",rsq(model, train, 19), "\n")
  return(model)
}
```

Model #1: Complex Model with Audio and non-Audio Attributes

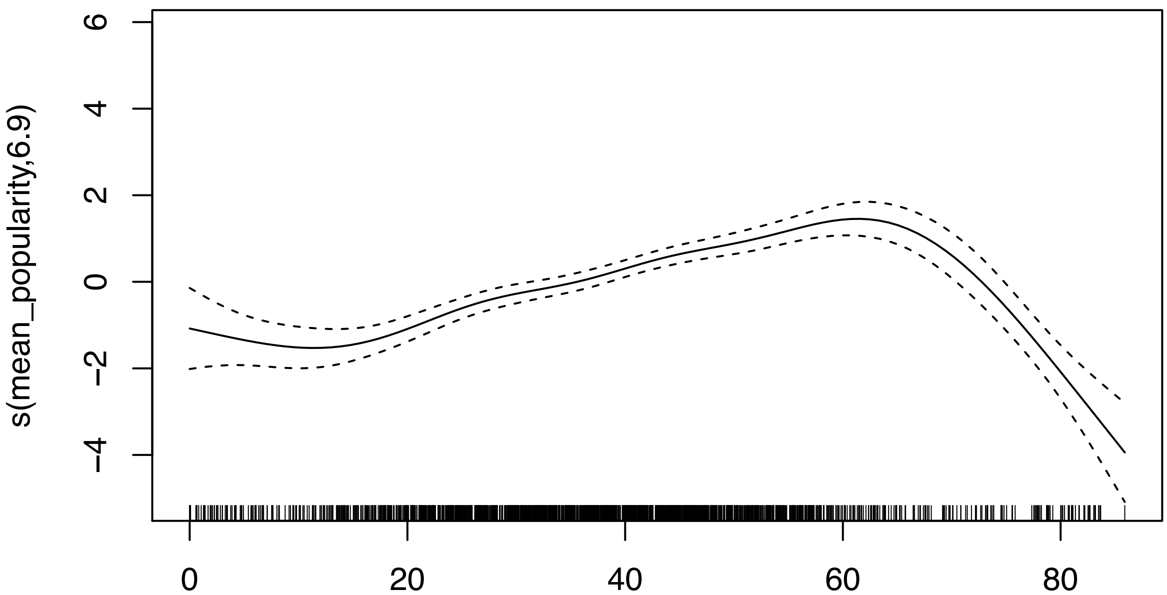
```r
results.3 = gam.results(formula.3)
```

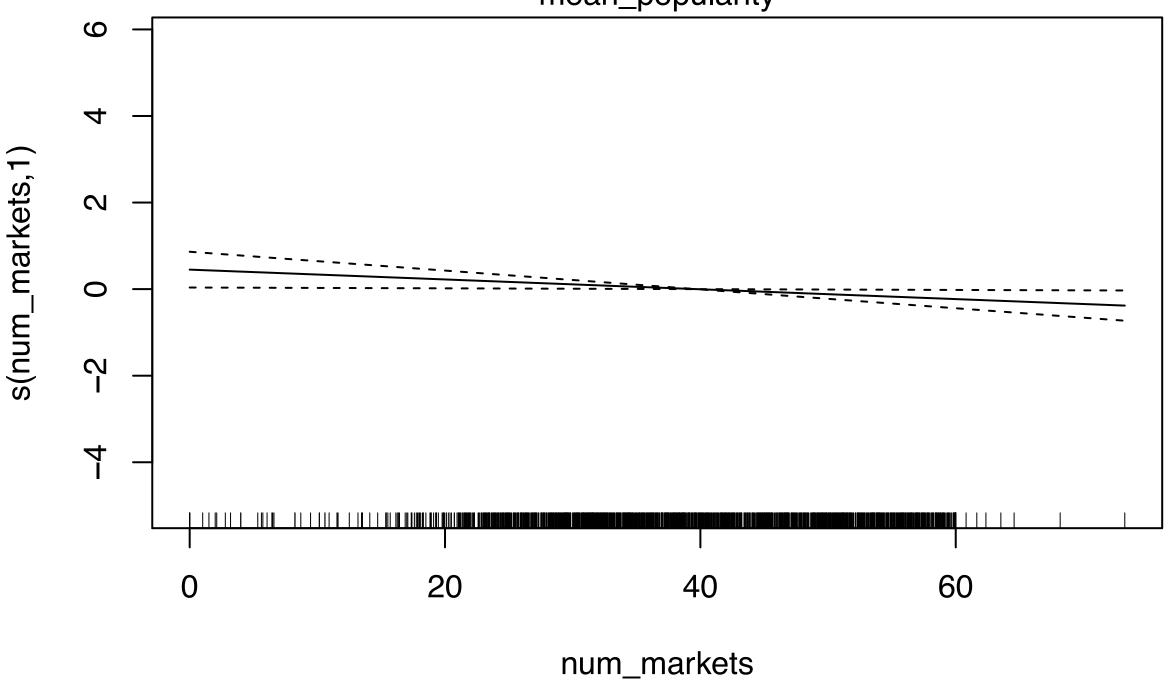
```
## Train R^2:  0.4943415
```

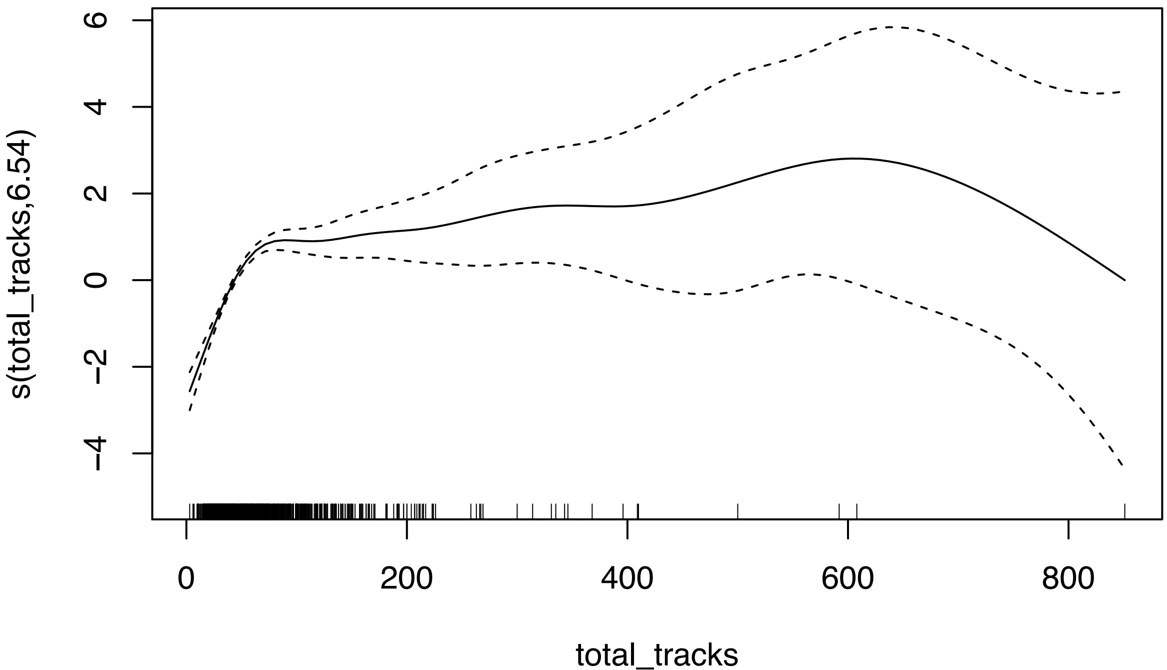
```r
preds = predict(results.3)
ggplot(train, mapping=aes(x=logfollowers, y=preds, color=featured)) + geom_point() + geom_abline(slope=
  scale_x_continuous(name="Log of # of followers") + scale_y_continuous(name="Predicted log of # of foll
```



```r
plot(results.3, se=TRUE)
```








```r
#coef(results.3)
summary(results.3)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## logfollowers ~ acousticness + added_date + featured + danceability +
##     duration + energy + instrumentalness + key + liveness + loudness +
##     s(mean_popularity) + mode + s(num_markets) + tempo + time_signature +
##     s(total_tracks) + valence
##
## Parametric coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -3.566e+01  4.078e+00  -8.745  < 2e-16 ***
## acousticness      5.436e-02  6.633e-01   0.082  0.93469
## added_date        2.798e-03  1.478e-04  18.934  < 2e-16 ***
## featuredTrue      1.857e+00  6.576e-01   2.824  0.00481 **
## danceability      1.018e+00  8.533e-01   1.193  0.23315
## duration          1.481e-06  5.115e-07   2.895  0.00384 **
## energy            2.975e-01  1.167e+00   0.255  0.79877
## instrumentalness  4.410e-01  3.914e-01   1.127  0.25998
## key               7.512e-02  9.300e-02   0.808  0.41936
## liveness          9.492e-01  9.597e-01   0.989  0.32277
## loudness         -5.837e-02  3.698e-02  -1.578  0.11470
## mode             -5.726e-01  4.638e-01  -1.235  0.21716
## tempo             3.971e-03  8.518e-03   0.466  0.64115
## time_signature   -1.252e+00  6.715e-01  -1.864  0.06245 .
## valence           6.208e-01  6.175e-01   1.005  0.31485
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

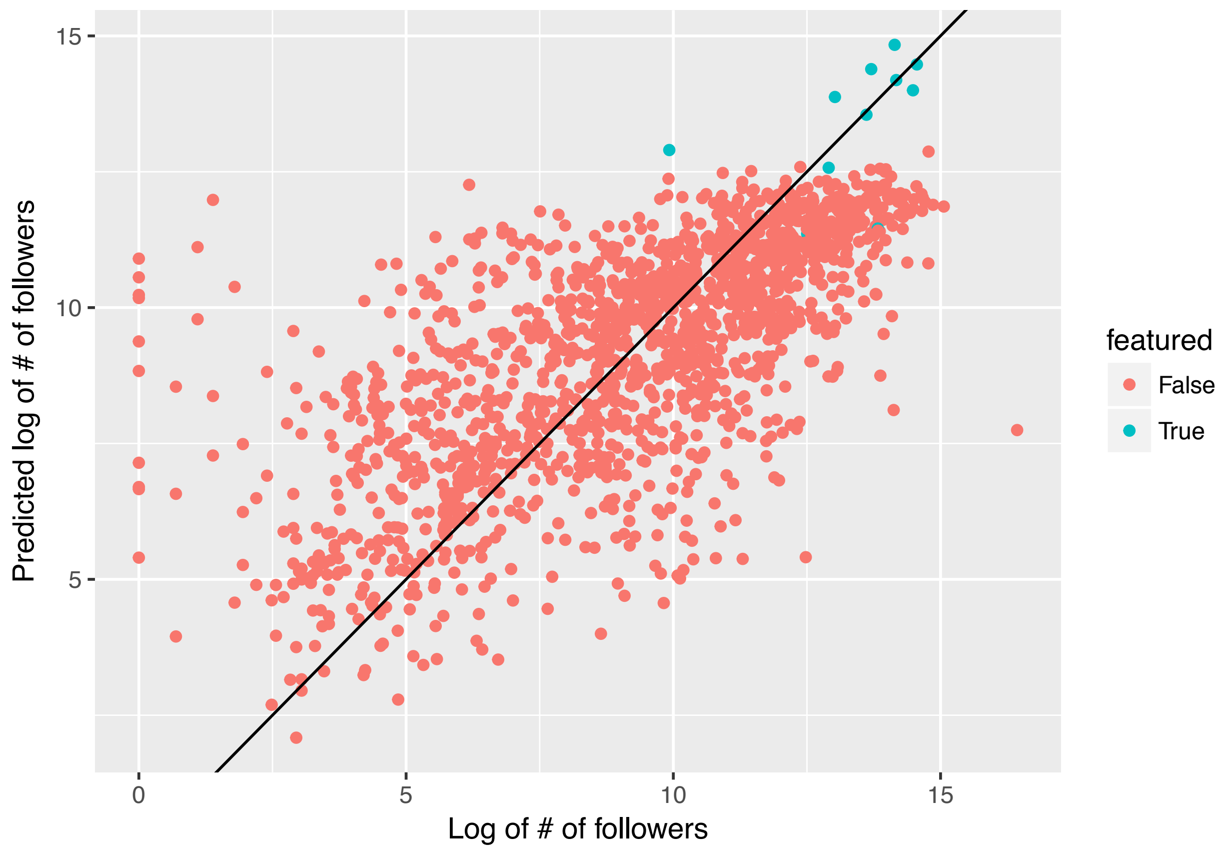
```
## Approximate significance of smooth terms:
##                        edf Ref.df      F p-value
## s(mean_popularity) 6.903  7.998 27.240  <2e-16 ***
## s(num_markets)     1.000  1.000  4.748  0.0295 *
## s(total_tracks)    6.544  7.564 27.072  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.485   Deviance explained = 49.4%
## GCV = 4.9387  Scale est. = 4.8497    n = 1634
```

Model #2: Simpler Model with only basic Audio Attributes

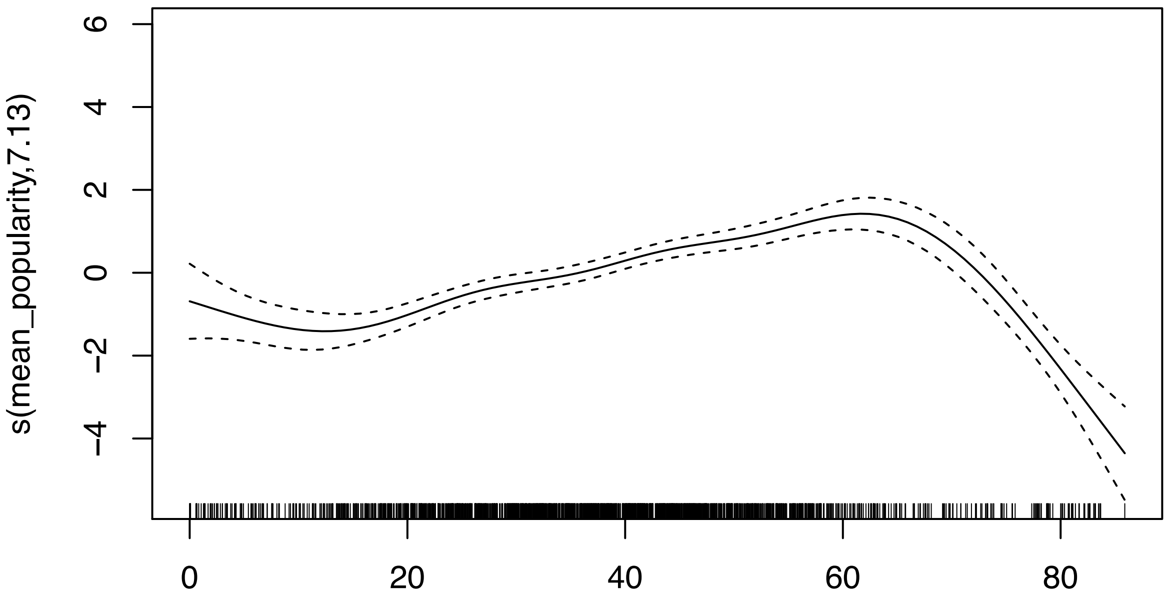
```
results.4 = gam.results(formula.4)
```

```
## Train R^2:  0.4889698
```

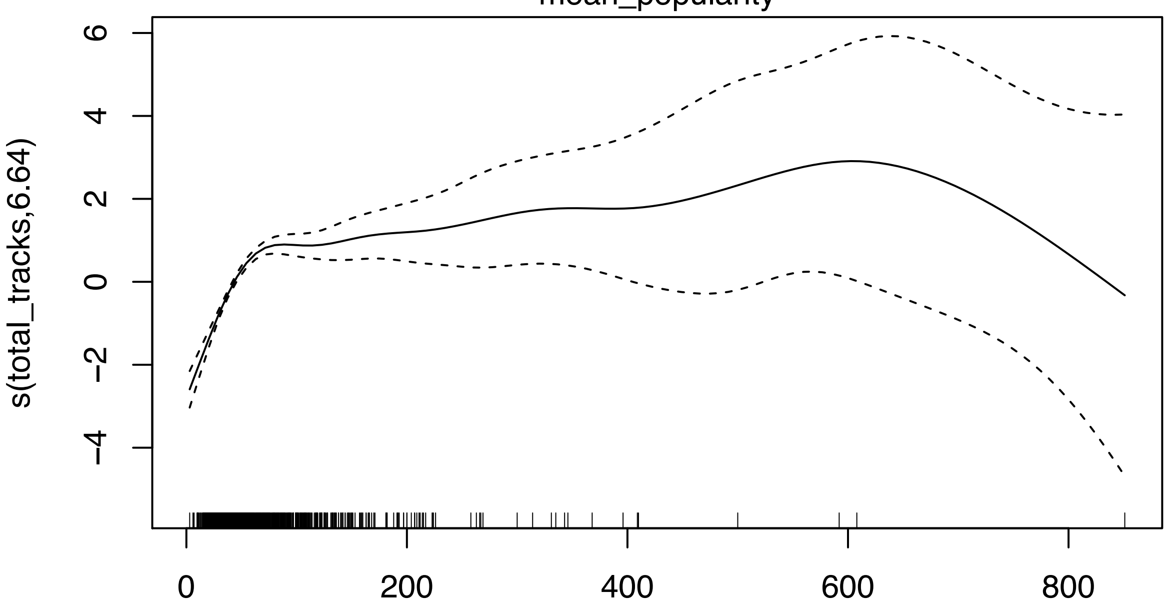
```
preds = predict(results.4)
ggplot(train, mapping=aes(x=logfollowers, y=preds, color=featured)) + geom_point() + geom_abline(slope=
  scale_x_continuous(name="Log of # of followers") + scale_y_continuous(name="Predicted log of # of fol
```




```
#scale_y_continuous(limits=c(-1,1)) + scale_x_continuous(limits=c(-1,1)) + coord_fixed(ratio = 1)
#summary(new.results)

plot(results.4, se=TRUE)
```



mean_popularity



total_tracks

```r
#coef(results.4)
summary(results.4)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## logfollowers ~ added_date + featured + time_signature + duration +
##     s(mean_popularity) + s(total_tracks)
##
## Parametric coefficients:
```

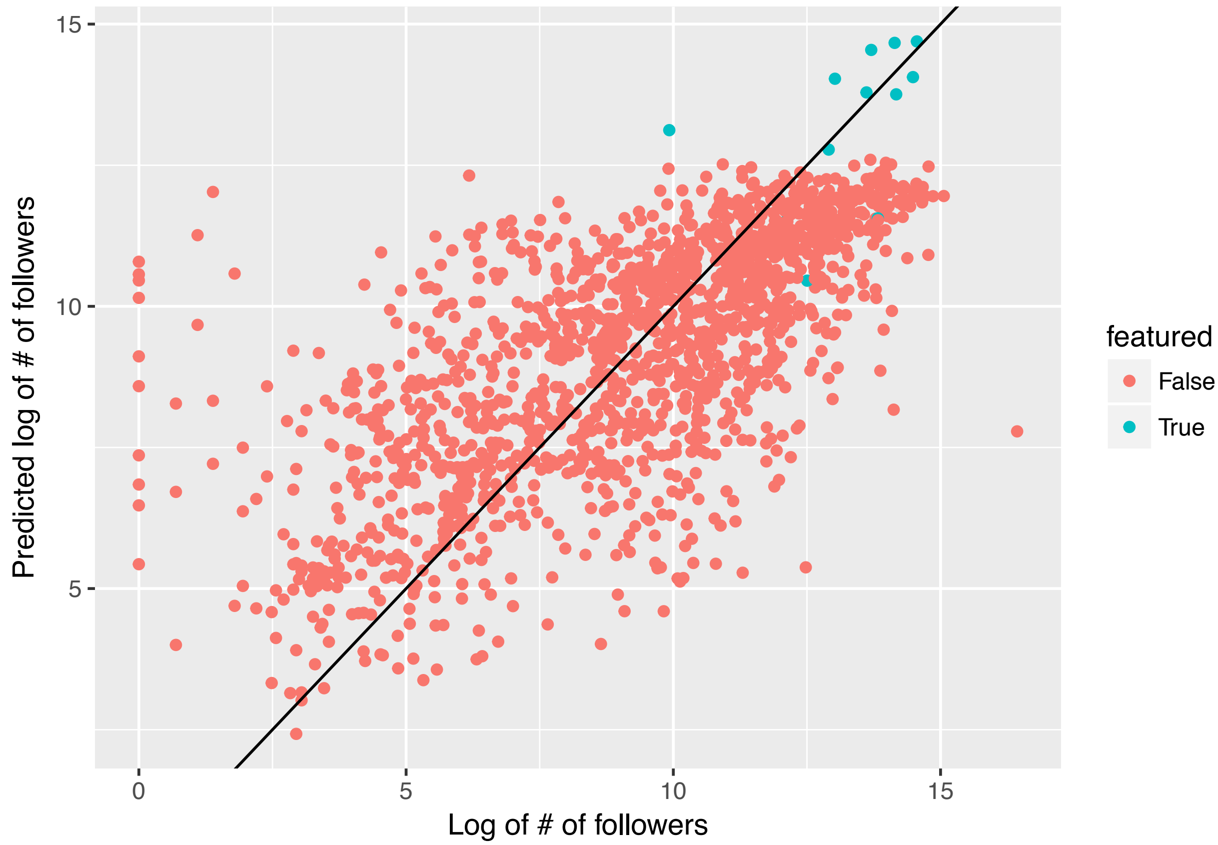
```
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.241e+01  3.324e+00  -9.752  < 2e-16 ***
## added_date      2.729e-03  1.399e-04  19.507  < 2e-16 ***
## featuredTrue    1.998e+00  6.498e-01   3.074 0.002144 **
## time_signature -1.204e+00  5.019e-01  -2.399 0.016532 *
## duration        1.655e-06  4.719e-07   3.507 0.000466 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                       edf Ref.df     F p-value
## s(mean_popularity) 7.128  8.173 26.94  <2e-16 ***
## s(total_tracks)    6.640  7.649 27.35  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.483   Deviance explained = 48.9%
## GCV = 4.9254  Scale est. = 4.8688    n = 1634
```

Model #3: Simpler Model with no Audio Attributes

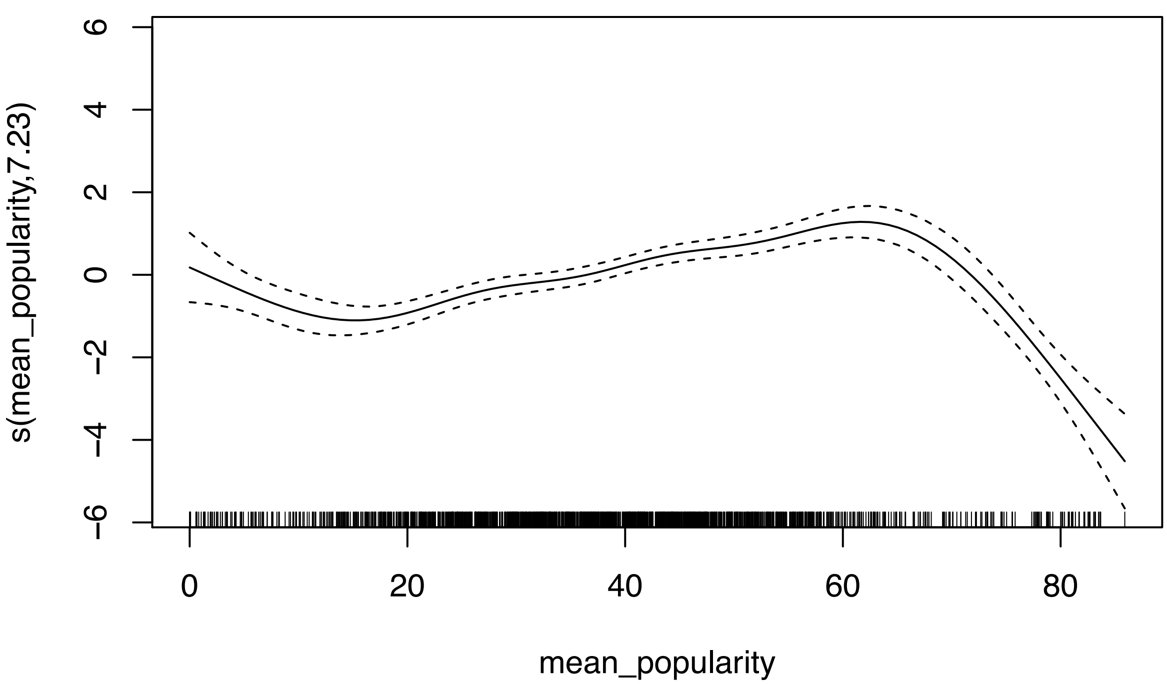
```
results.5 = gam.results(formula.5)
```

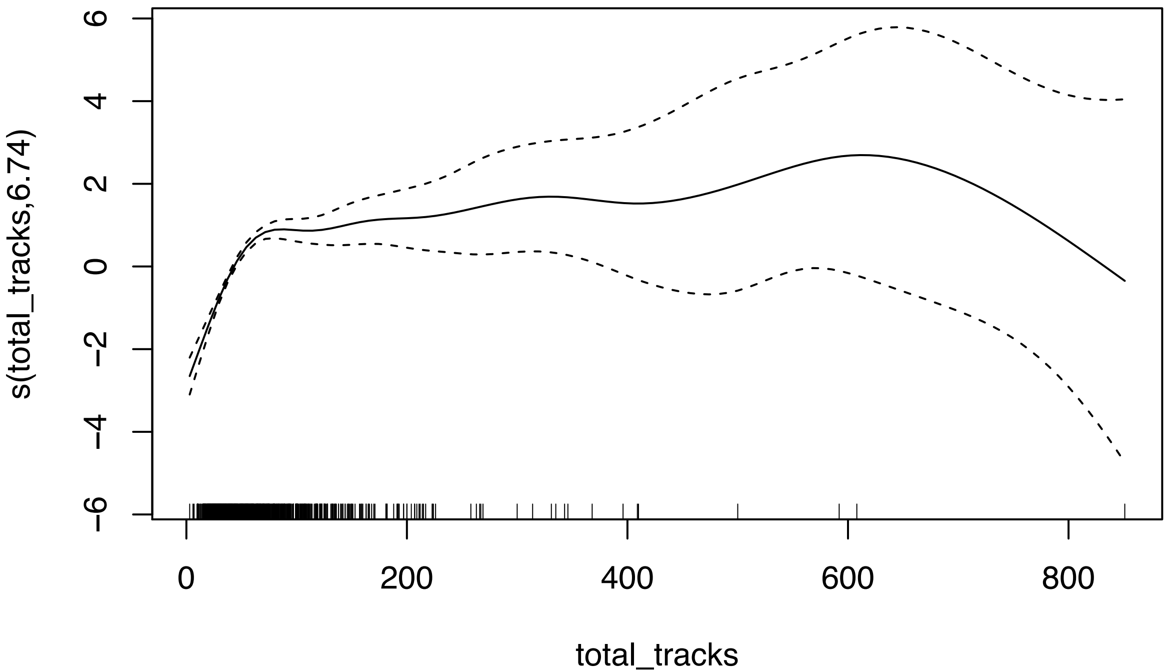
```
## Train R^2:  0.4821185
```

```
preds = predict(results.5)
ggplot(train, mapping=aes(x=logfollowers, y=preds, color=featured)) + geom_point() + geom_abline(slope=
  scale_x_continuous(name="Log of # of followers") + scale_y_continuous(name="Predicted log of # of fol
```




```r
plot(results.5, se=TRUE)
```






```
#coef(results.4)
summary(results.5)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## logfollowers ~ added_date + featured + s(mean_popularity) + s(total_tracks)
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.843e+01  2.324e+00 -16.534  < 2e-16 ***
## added_date    2.833e-03  1.382e-04  20.497  < 2e-16 ***
## featuredTrue  2.257e+00  6.473e-01   3.487 0.000502 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                      edf Ref.df     F p-value
## s(mean_popularity) 7.227  8.247 24.52  <2e-16 ***
## s(total_tracks)    6.743  7.740 27.05  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.477   Deviance explained = 48.2%
## GCV = 4.9803  Scale est. = 4.9286    n = 1634
```

Likelihood Ratio Test to compare models

```
anova(results.5, results.4, results.3, test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: logfollowers ~ added_date + featured + s(mean_popularity) + s(total_tracks)
## Model 2: logfollowers ~ added_date + featured + time_signature + duration +
##     s(mean_popularity) + s(total_tracks)
## Model 3: logfollowers ~ acousticness + added_date + featured + danceability +
##     duration + energy + instrumentalness + key + liveness + loudness +
##     s(mean_popularity) + mode + s(num_markets) + tempo + time_signature +
##     s(total_tracks) + valence
##   Resid. Df Resid. Dev     Df Deviance Pr(>Chi)
## 1    1615.0     7969.7
## 2    1613.2     7864.3  1.835  105.435 1.47e-05 ***
## 3    1602.4     7781.6 10.739   82.665  0.09709 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Only statistically significant difference is between models 1 and 2 (adding time signature and duration). Adding additional audio attributes beyond these two only has signifance at the p<0.1 level.