Bayesian Statistics - Prediction of Movie Popularity Using Bayesian Modelling & Linear Regression

Setup

Load packages

```
library(ggplot2)
library(dplyr)
library(statsr)
library(BAS)
library(grid)
library(gridExtra)

set.seed(3514)
```

Load data

```
load("movies.Rdata")
```

Part 1: Data

The provided data consists of audience and critic review scores for 651 movies released during the years prior to 2016. The data comes from the Rotten Tomatoes and IMDb web sites. In addition to review scores, the data contains several other variables for descriptive information regarding each movie such as genre, running time, MPAA rating, production studio, Oscar nominations, and more.

Generalizability:

The raw data is not a complete list of all movies released prior to 2016. Instead, it is a random sample taken from the full data set. Hence, the results can be assumed to be generalizable to the population of all movies and that there is no bias introduced by the sampling method.

Causality:

As the raw data is a sample taken from existing data, this is an observational study and no causal relationships can be inferred or assumed from the conclusions drawn.

Part 2: Data manipulation

There are 32 variables provided in the raw data. In addition, we create the following variables for this analysis.

1. feature_film: “yes” if title_type is “Feature Film”, “no” otherwise
2. drama: “yes” if genre is “Drama”, “no” otherwise
3. mpaa_rating_R: “yes” if mpaa_rating is “R”, “no” otherwise
4. oscar_season: “yes” if movie is released in November, October, or December (based on thtr_rel_month), “no” otherwise
5. summer_season: “yes” if movie is released in May, June, July, or August (based on thtr_rel_month), “no” otherwise

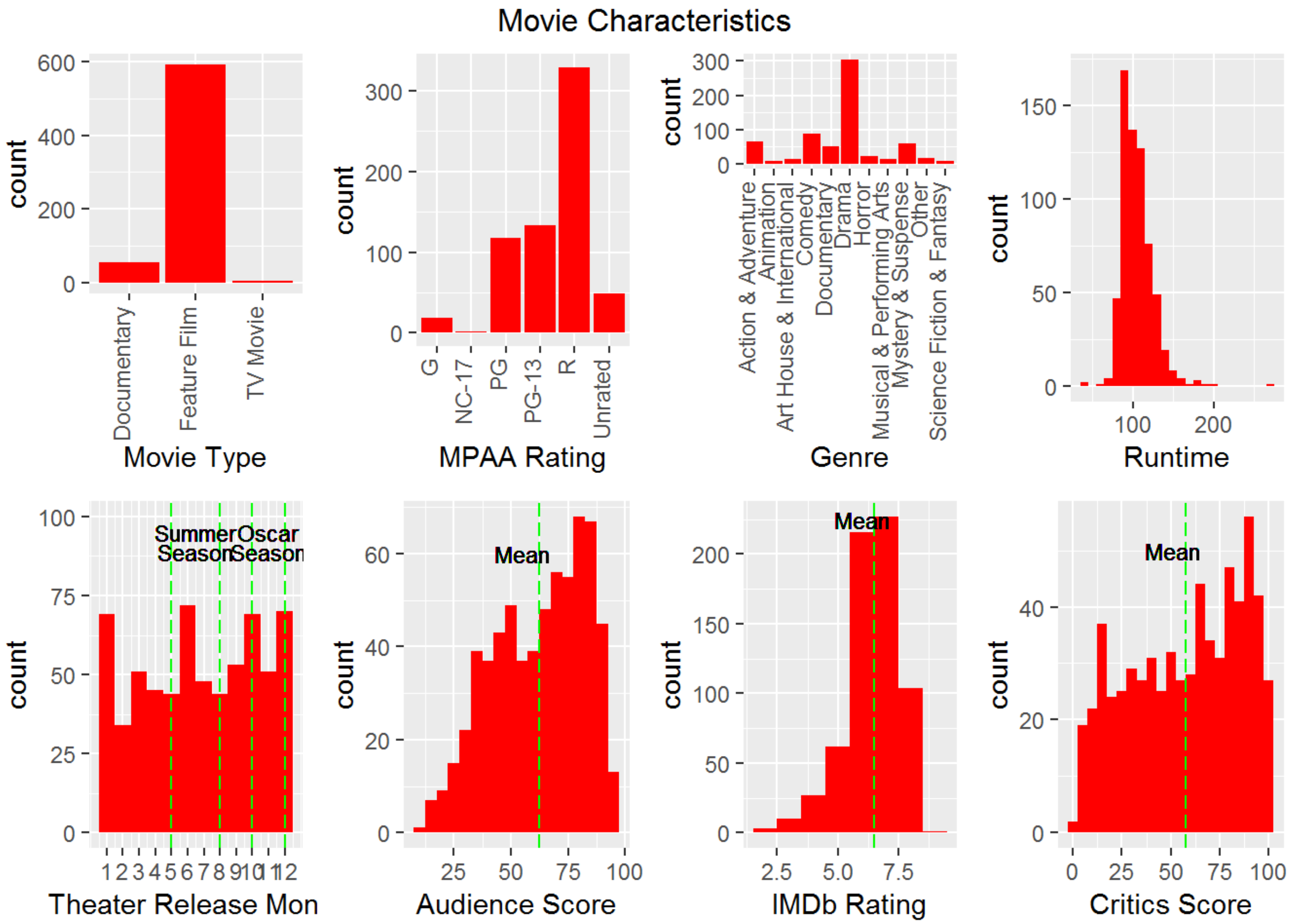
```
movies <- movies %>%
          mutate(feature_film=as.factor(ifelse(title_type == 'Feature Film', 'yes', '
no'))) %>%
          mutate(drama=as.factor(ifelse(genre == 'Drama', 'yes', 'no'))) %>%
          mutate(mpaa_rating_R=as.factor(ifelse(mpaa_rating == 'R', 'yes', 'no'))) %>
%
          mutate(oscar_season=as.factor(ifelse(thtr_rel_month %in% c(10:12), 'yes', '
no'))) %>%
          mutate(summer_season=as.factor(ifelse(thtr_rel_month %in% c(5:8), 'yes', 'n
o')))


# Remove N/A values in key variable in the data set.
movies <- filter(movies, !is.na(runtime))
```

Part 3: Exploratory data analysis

After removing the N/A values, there are 650 movies in the final data set for analysis. The following charts demonstrate the characterization of various movie characteristics.

```r
# Create histograms of some of the key movie characteristic data.
p1 <- ggplot(data=movies, aes(x=genre)) +
      geom_bar(fill="red") +
      xlab("Genre") +
      theme(axis.text.x=element_text(angle=90, hjust=1, vjust=0))
p2 <- ggplot(data=movies, aes(x=title_type)) +
      geom_bar(fill="red") +
      xlab("Movie Type") +
      theme(axis.text.x=element_text(angle=90, hjust=1, vjust=0))
p3 <- ggplot(data=movies, aes(x=mpaa_rating)) +
      geom_bar(fill="red") +
      xlab("MPAA Rating") +
      theme(axis.text.x=element_text(angle=90, hjust=1, vjust=0))
p4 <- ggplot(data=movies, aes(x=runtime)) +
      geom_histogram(binwidth=10, fill="red") +
      xlab("Runtime")
p5 <- ggplot(data=movies, aes(x=thtr_rel_month)) +
      geom_histogram(binwidth=1, fill="red") +
      scale_x_continuous(breaks=c(1:12)) +
      scale_y_continuous(limits=c(0, 100)) +
      geom_vline(xintercept=c(5, 8, 10, 12), colour='green', linetype='longdash') +
      geom_text(label='Summer', x=6.5, y=95, hjust='center', size=3) +
      geom_text(label='Oscar', x=11, y=95, hjust='center', size=3) +
      geom_text(label='Season', x=6.5, y=89, hjust='center', size=3) +
      geom_text(label='Season', x=11, y=89, hjust='center', size=3) +
      xlab("Theater Release Month")
p6 <- ggplot(data=movies, aes(x=audience_score)) +
      geom_histogram(binwidth=5, fill="red") +
      geom_vline(xintercept=mean(movies$audience_score), colour='green', linetype='lo
ngdash') +
      geom_text(label='Mean', x=55, y=60, hjust='center', size=3) +
      xlab("Audience Score")
p7 <- ggplot(data=movies, aes(x=imdb_rating)) +
      geom_histogram(binwidth=1, fill="red") +
      geom_vline(xintercept=mean(movies$imdb_rating), colour='green', linetype='longd
ash') +
      geom_text(label='Mean', x=6, y=225, hjust='center', size=3) +
      xlab("IMDb Rating")
p8 <- ggplot(data=movies, aes(x=critics_score)) +
      geom_histogram(binwidth=5, fill="red") +
      geom_vline(xintercept=mean(movies$critics_score), colour='green', linetype='lon
gdash') +
      geom_text(label='Mean', x=51, y=50, hjust='center', size=3) +
      xlab("Critics Score")
grid.arrange(p2, p3, p1, p4, p5, p6, p7, p8, nrow=2,
             top="Movie Characteristics")
```



The most common movie in the data set is R rated,is a feature film, a drama with a runtime of around 90 minutes. Approximately 32% of the movies were released during the summer season, and 29% of the movies were released during the Oscar season. The distribution of audience scores shows an interesting bi-modality. As this is the value to be predicted by the model, one would like to see a normal distribution for this.

Part 4: Modeling

We need to predict the popularity of a movie - as quantified by the audience_score variable - using a linear regression model and bayesian model averaging.

The initial set of predictors chosen for the model were runtime, thtr_rel_year, imdb_rating, imdb_num_votes, critics_score, best_pic_nom, best_pic_win, best_actor_win, best_actress_win, best_dir_win, top200_box, feature_film, drama, mpaa_rating_R, oscar_season, summer_season

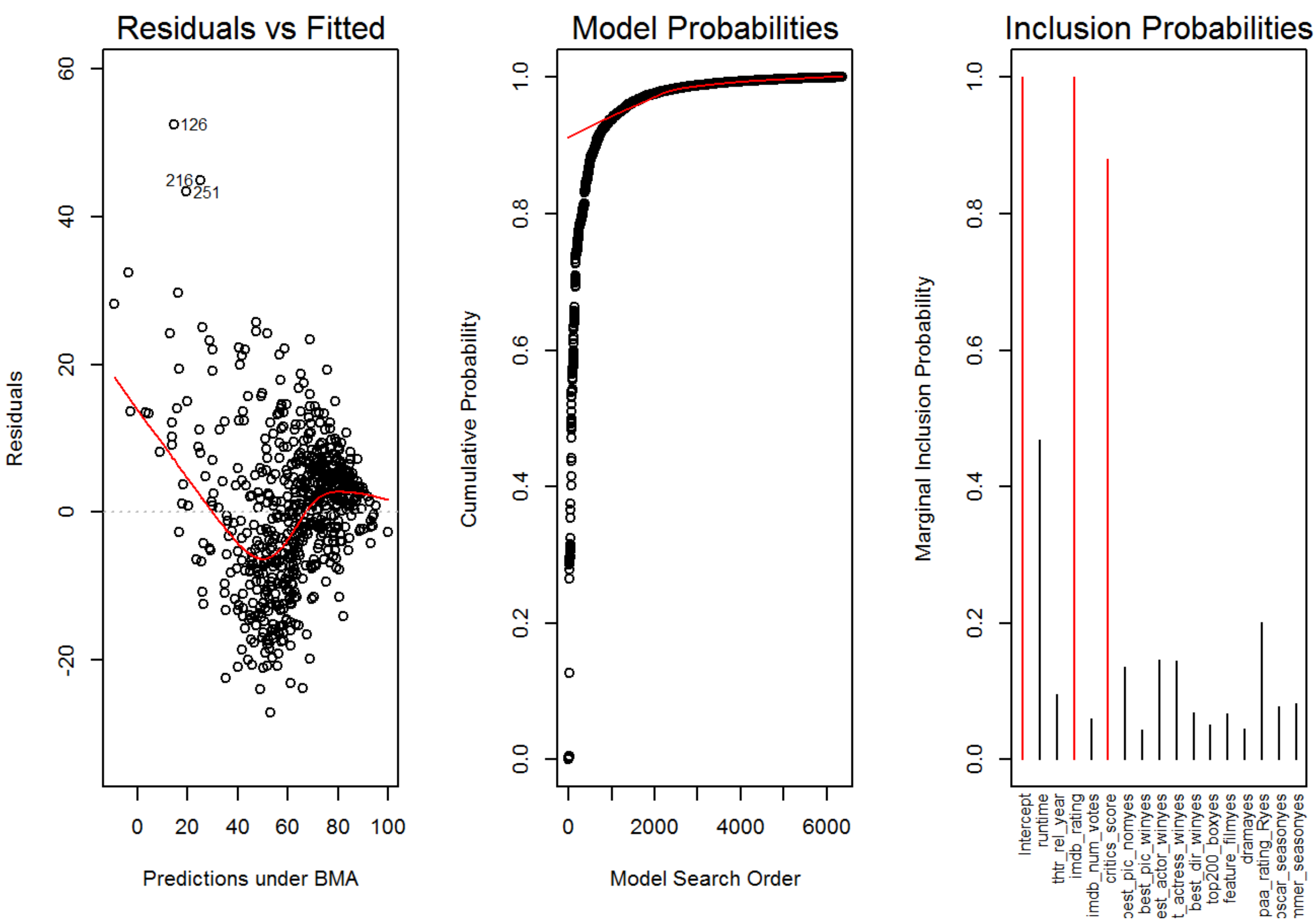
As the first step, we reduced the dataset to include only the response variable and the initial set of predictor variables.

```
movies <- select(movies, runtime, thtr_rel_year, imdb_rating, imdb_num_votes,
                 critics_score, audience_score, best_pic_nom, best_pic_win,
                 best_actor_win, best_actress_win, best_dir_win, top200_box,
                 feature_film, drama, mpaa_rating_R, oscar_season, summer_season)
```

We create a Bayesian linear regression model using all of the initial predictor variables. We use the MCMC (Markov Chain Monte Carlo) method for sampling models during the fitting process, and the prior probabilities for the regression coefficients are assigned using the Zellner-Siow Cauchy distribution. A uniform distribution is used for the prior probabilities for all models.

```
basLM1 <- bas.lm(audience_score ~ ., data=movies, method='MCMC',
                 prior='ZS-null', modelprior=uniform())


par(mfrow=c(1,3))
plot(basLM1, which=c(1, 2), ask=FALSE)
plot(basLM1, which=4, ask=FALSE, cex.lab=0.5)
```



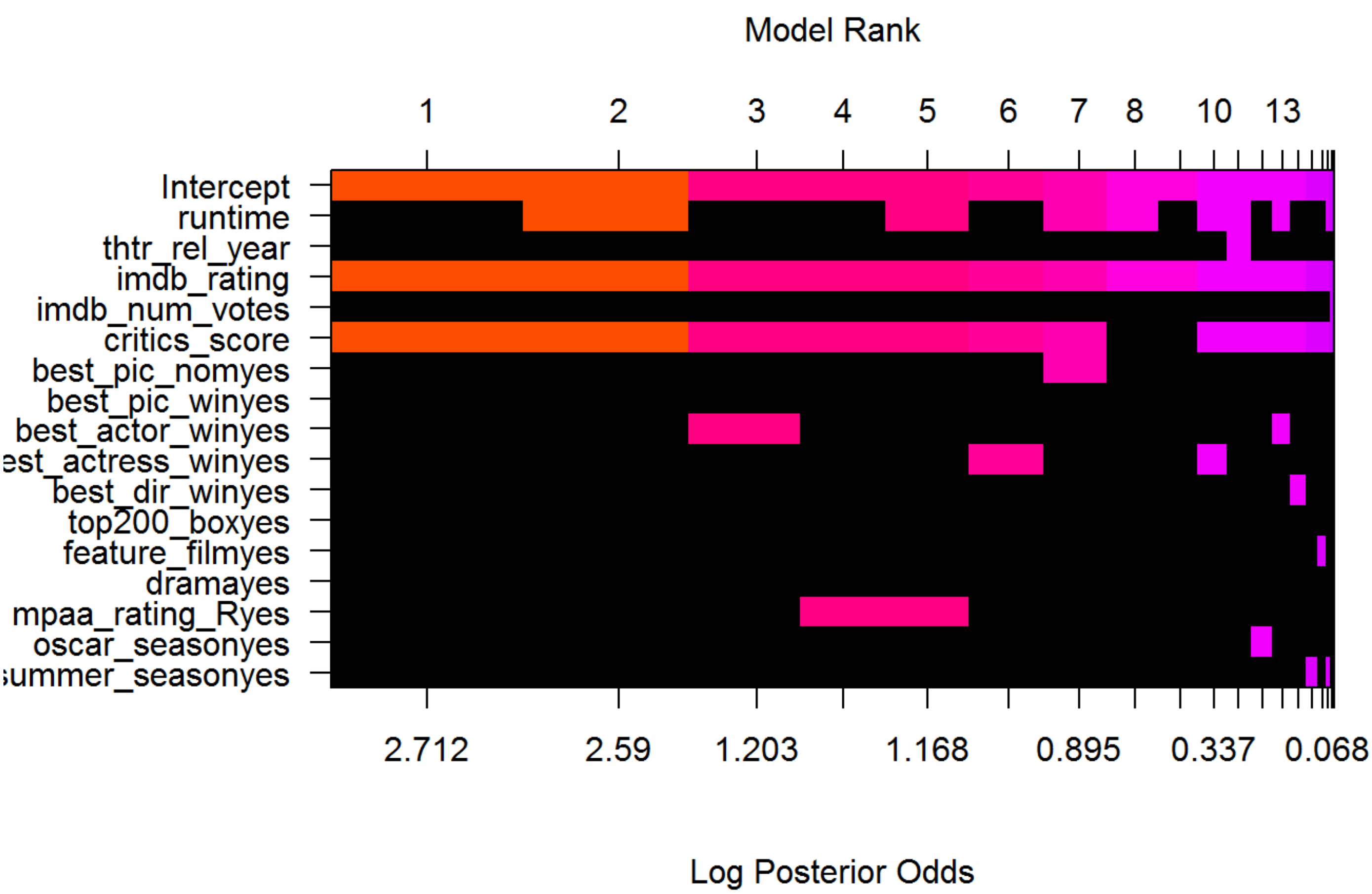
The above charts depict the diagnostic fitting process.

The left chart does not show a random scattering of the residuals versus the fitted values. The wave pattern indicates that the model tended to predict too low for ratings under a value of 30 and over a value of about 75. In between 30 and 75, the model tended to predict high. Above a rating of 75, the residuals appear to scatter randomly. This would indicate that perhaps some of the other predictor candidates should be further evaluated for inclusion in the model, or perhaps there is something else affecting the ratings that is not accounted for at all by the initial set of chosen predictors.

The middle chart shows that the model posterior probability density leveled off at 1 after approx. 3,000 (3,500 to be exact) model combinations had been sampled. The number of models was capped at this point rather than proceed with the evaluation of all 2^16 possible model combinations.

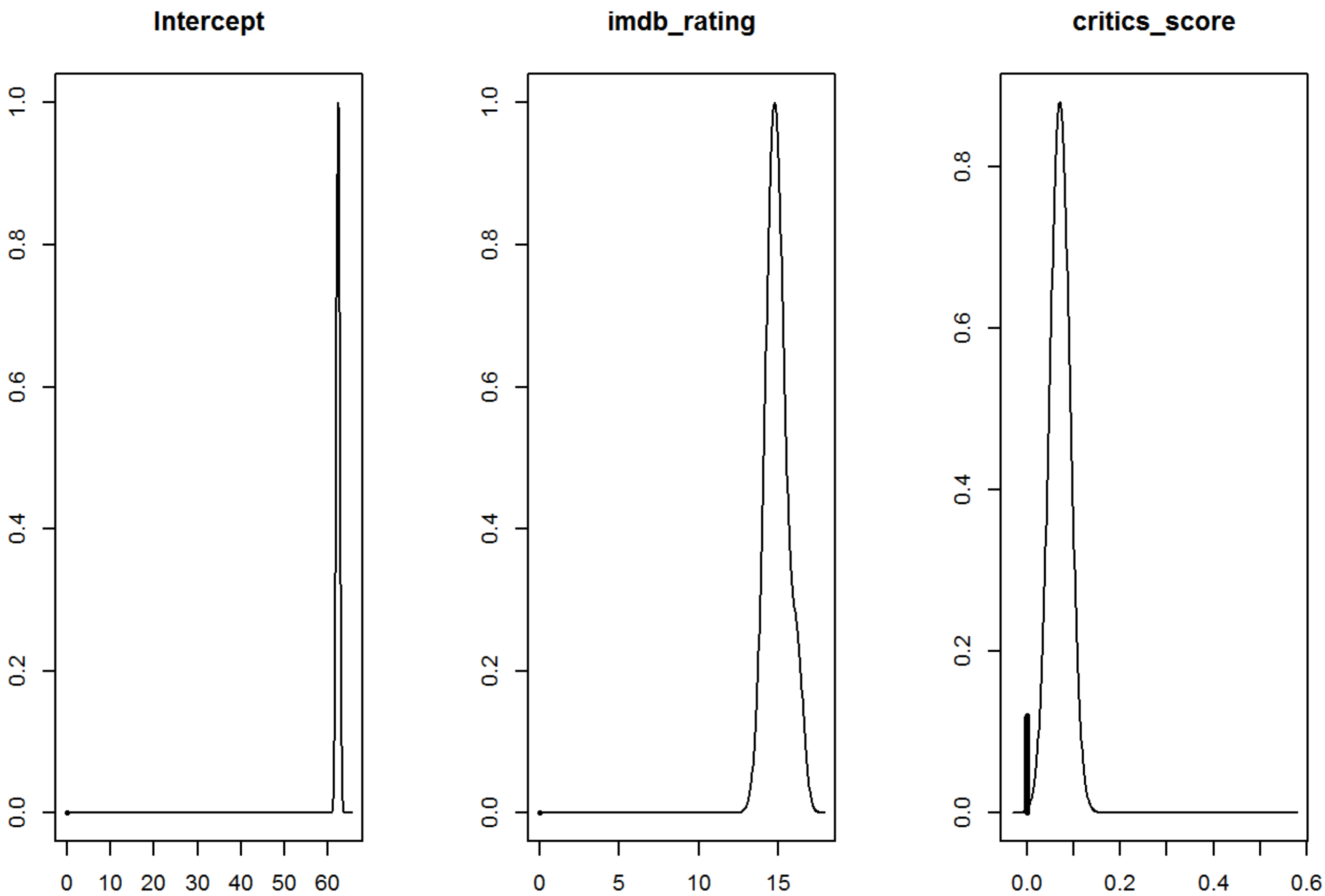
The right chart shows the inclusion probabilities for each of the predictors. How each predictor was used is shown graphically below for the top 20 models tested.

```
image(basLM1, rotate=FALSE)
```



The model with the highest posterior probability (0.1404) contains only two of the predictors: imbd_rating and critics_score. The posterior distributions of the regression coefficients are shown below.

```
par(mfrow=c(1,3))
plot(coefficients(basLM1), subset=c(1, 4, 6), ask=FALSE)
```



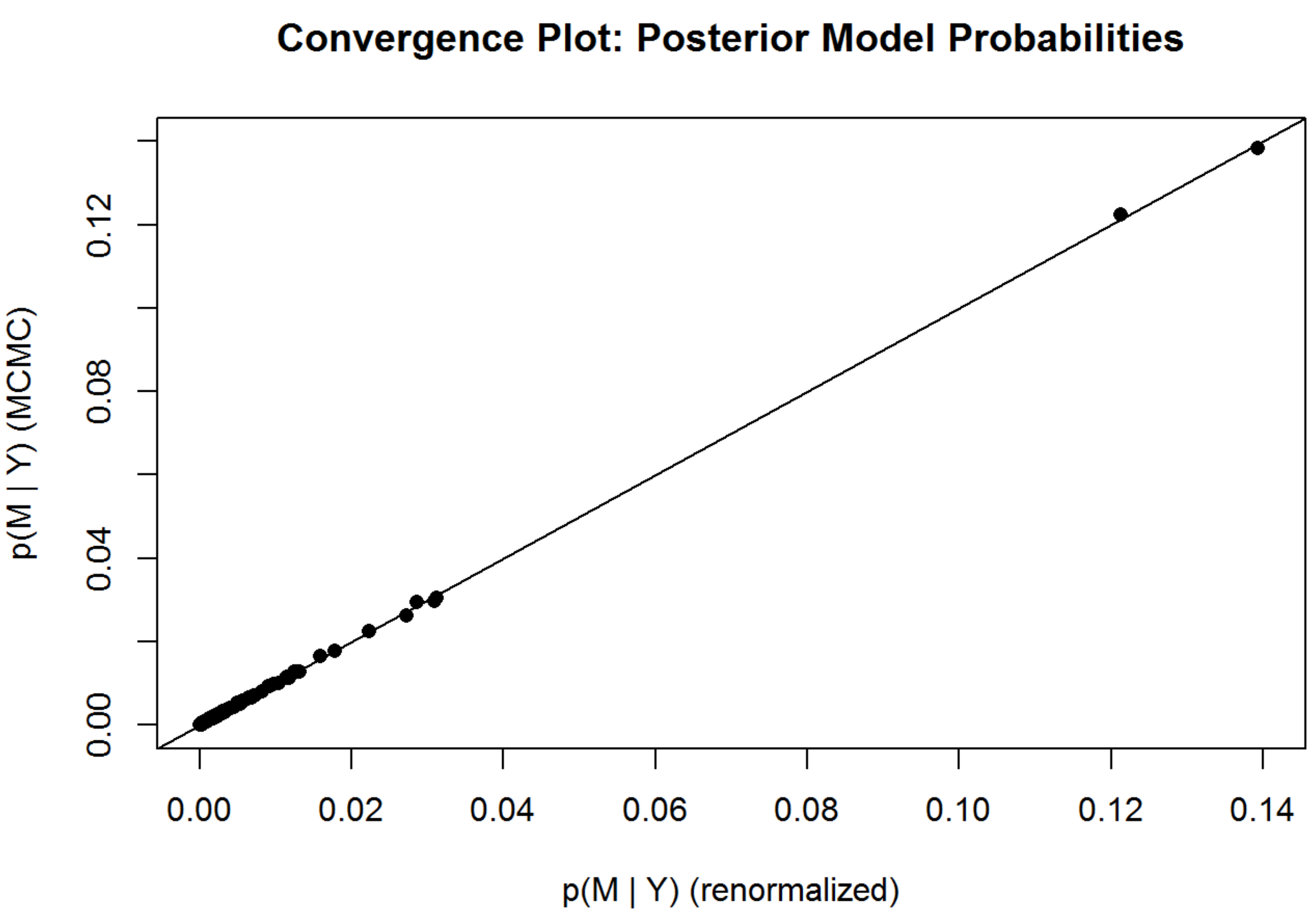
The distribution for imdb_rating is not entirely normal as there appears to be a small non-symmetric bump on the right side. This could be related to the wave pattern shown on the residuals plot above or the non-normal distribution of audience score values. Also, the regression intercept is right at the median of the range of audience_score with only positive coefficients for the other regression coefficients - meaning that a predicted audience score cannot be below 65. This almost certainly contributes to the wave pattern in the residuals plot above.

Credible intervals are determined by using the model to predict the same audience scores as were used to fit the model originally. Below are shown the intervals for the fitted and predicted values:

```
BMA_basLM1 = predict(basLM1, estimator="BMA", se.fit=TRUE)
BMA_confint_fit = confint(BMA_basLM1, parm="mean")
BMA_confint_pred = confint(BMA_basLM1, parm="pred")
head(cbind(BMA_confint_fit, BMA_confint_pred), 10)
```

```
##             2.5%    97.5%     mean     2.5%     97.5%     pred
##  [1,] 45.26351 49.09597 47.20259 26.93508  66.49927 47.20259
##  [2,] 75.10285 78.98445 77.14570 57.40756  97.01032 77.14570
##  [3,] 79.49173 83.81131 81.55113 61.93788 101.61468 81.55113
##  [4,] 70.41754 75.70754 73.23432 53.19594  93.11153 73.23432
##  [5,] 38.52551 41.79076 40.23987 20.75839  60.74609 40.23987
##  [6,] 82.55648 87.38474 84.95281 65.01187 104.26905 84.95281
##  [7,] 69.58878 74.73521 72.24687 52.18748  91.65782 72.24687
##  [8,] 42.20789 47.41918 44.84076 24.13632  63.86494 44.84076
##  [9,] 78.12377 82.27512 80.13070 60.97483 100.76163 80.13070
## [10,] 62.98777 67.38028 65.34738 45.33075  84.76524 65.34738
```

```
diagnostics(basLM1, type="model",  pch=16)
```



The chart shows that the posterior model probabilities follow a normal distribution.

Part 5: Prediction

We tested the predictive capability of the model using data for the movie Zootopia. The information for this movie was obtained from the IMDb and Rotten Tomatoes web sites in order to be consistent with the analysis data.

```
dfZootopia <- data.frame(runtime=108,
                         thtr_rel_year=2016,
                         imdb_rating=8.0,
                         imdb_num_votes=359424,
                         critics_score=97,
                         audience_score=92,
                         best_pic_nom=factor("yes", levels=c("no", "yes")),
                         best_pic_win=factor("yes", levels=c("no", "yes")),
                         best_actor_win=factor("no", levels=c("no", "yes")),
                         best_actress_win=factor("no", levels=c("no", "yes")),
                         best_dir_win=factor("no", levels=c("no", "yes")),
                         top200_box=factor("yes", levels=c("no", "yes")),
                         feature_film=factor("yes", levels=c("no", "yes")),
                         drama=factor("no", levels=c("no", "yes")),
                         mpaa_rating_R=factor("no", levels=c("no", "yes")),
                         oscar_season=factor("no", levels=c("no", "yes")),
                         summer_season=factor("no", levels=c("no", "yes")))

BMA_basLM1_DP <- predict(basLM1, newdata=dfZootopia, estimator="BMA", se.fit=TRUE)

BMA_basLM1_predME <- qt(0.95, df=BMA_basLM1_DP$se.bma.pred[1]) *
                     mean(BMA_basLM1_DP$se.bma.pred)

df <- data.frame(t="Zootopia",
                 p=sprintf("%2.1f", BMA_basLM1_DP$Ybma),
                 i=sprintf("%2.1f - %2.1f", BMA_basLM1_DP$Ybma - BMA_basLM1_predME,
                           BMA_basLM1_DP$Ybma + BMA_basLM1_predME),
                 r=92)
colnames(df) <- c("Movie Title", "Predicted Rating", "95% Prediction Interval",
                  "Actual Rating")
print(df)
```

```
##   Movie Title Predicted Rating 95% Prediction Interval Actual Rating
## 1    Zootopia             88.1            69.5 - 106.7            92
```

The true audience score for the movie is 92. The model prediction is 88.1 with a 95% prediction interval of 69.5 - 106.7.

Part 6: Conclusion

We created a parsimonious linear regression model using bayesian model averaging that was proved to have some capability for predicting movie popularity based on certain movie characteristics.

Shortconigs: There is much room for further analysis in at least the following areas:

- What is the nature of the wave pattern in the residuals plot?
- Would it be better to create separate models for each movie type or genre? Would doing so eliminate the non-normal distribution of audience score values in the analysis data?

THANK YOU