Statistical Computing Report

Anthony Anderson

April 13, 2019

```r
imdb <- read.csv("imdb.csv")
#str(imdb)
library("pastecs")
#stat.desc(imdb)
#there are NA's making it difficult to run basic analysis, we'll omit the rows that contain NA's
imdb.clean<-na.omit(imdb)
summary(imdb.clean[,3:12])
```

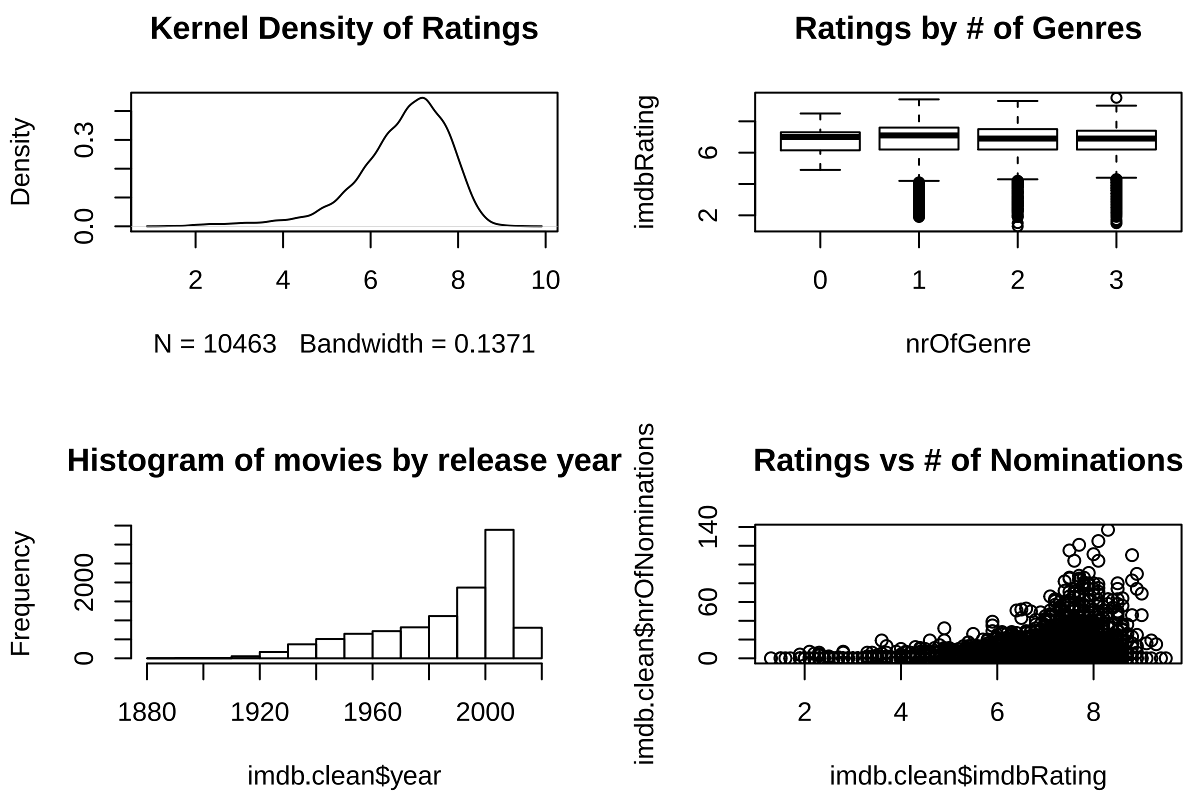
```
##    imdbRating     ratingCount         duration           year     
##  Min.   :1.300   Min.   :      5   Min.   :    2   Min.   :1888  
##  1st Qu.:6.200   1st Qu.:   1378   1st Qu.: 5400   1st Qu.:1972  
##  Median :6.900   Median :   6227   Median : 6000   Median :1996  
##  Mean   :6.738   Mean   :  32188   Mean   : 6159   Mean   :1987  
##  3rd Qu.:7.500   3rd Qu.:  29500   3rd Qu.: 6960   3rd Qu.:2006  
##  Max.   :9.500   Max.   :1183395   Max.   :46200   Max.   :2014  
##     nrOfWins        nrOfNominations     nrOfPhotos     nrOfNewsArticles
##  Min.   :  0.000   Min.   :  0.000   Min.   :  0.00   Min.   :    0.0  
##  1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.:  3.00   1st Qu.:    0.0  
##  Median :  1.000   Median :  0.000   Median : 12.00   Median :   23.0  
##  Mean   :  3.442   Mean   :  4.094   Mean   : 23.29   Mean   :  258.9  
##  3rd Qu.:  3.000   3rd Qu.:  4.000   3rd Qu.: 32.00   3rd Qu.:  141.0  
##  Max.   :137.000   Max.   :137.000   Max.   :407.00   Max.   :23660.0  
##  nrOfUserReviews    nrOfGenre    
##  Min.   :   0.0   Min.   :0.000  
##  1st Qu.:  18.0   1st Qu.:2.000  
##  Median :  54.0   Median :3.000  
##  Mean   : 138.2   Mean   :2.358  
##  3rd Qu.: 146.0   3rd Qu.:3.000  
##  Max.   :4928.0   Max.   :3.000  
```

```r
stat.desc(imdb.clean[3:12])
```

```
##                 imdbRating  ratingCount     duration         year
## nbr.val      1.046300e+04 1.046300e+04 1.046300e+04 1.046300e+04
## nbr.null     0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## nbr.na       0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## min          1.300000e+00 5.000000e+00 2.000000e+00 1.888000e+03
## max          9.500000e+00 1.183395e+06 4.620000e+04 2.014000e+03
## range        8.200000e+00 1.183390e+06 4.619800e+04 1.260000e+02
## sum          7.049990e+04 3.367801e+08 6.444146e+07 2.078984e+07
## median       6.900000e+00 6.227000e+03 6.000000e+03 1.996000e+03
## mean         6.738020e+00 3.218771e+04 6.158985e+03 1.986987e+03
## SE.mean      1.044243e-02 7.001077e+02 1.906294e+01 2.297082e-01
## CI.mean.0.95 2.046915e-02 1.372345e+03 3.736699e+01 4.502719e-01
## var          1.140931e+00 5.128448e+09 3.802208e+06 5.520892e+02
## std.dev      1.068144e+00 7.161318e+04 1.949925e+03 2.349658e+01
## coef.var     1.585248e-01 2.224861e+00 3.165984e-01 1.182523e-02
##                  nrOfWins nrOfNominations   nrOfPhotos nrOfNewsArticles
```

```
## nbr.val      1.046300e+04    1.046300e+04 1.046300e+04     1.046300e+04
## nbr.null     5.139000e+03    5.877000e+03 1.985000e+03     2.838000e+03
## nbr.na       0.000000e+00    0.000000e+00 0.000000e+00     0.000000e+00
## min          0.000000e+00    0.000000e+00 0.000000e+00     0.000000e+00
## max          1.370000e+02    1.370000e+02 4.070000e+02     2.366000e+04
## range        1.370000e+02    1.370000e+02 4.070000e+02     2.366000e+04
## sum          3.601500e+04    4.283900e+04 2.436440e+05     2.708418e+06
## median       1.000000e+00    0.000000e+00 1.200000e+01     2.300000e+01
## mean         3.442129e+00    4.094332e+00 2.328625e+01     2.588567e+02
## SE.mean      8.271697e-02    9.213452e-02 3.126164e-01     9.420644e+00
## CI.mean.0.95 1.621410e-01    1.806012e-01 6.127878e-01     1.846626e+01
## var          7.158887e+01    8.881800e+01 1.022539e+03     9.285759e+05
## std.dev      8.461020e+00    9.424330e+00 3.197716e+01     9.636264e+02
## coef.var     2.458077e+00    2.301799e+00 1.373221e+00     3.722624e+00
##              nrOfUserReviews    nrOfGenre
## nbr.val         1.046300e+04 1.046300e+04
## nbr.null        3.360000e+02 2.300000e+01
## nbr.na          0.000000e+00 0.000000e+00
## min             0.000000e+00 0.000000e+00
## max             4.928000e+03 3.000000e+00
## range           4.928000e+03 3.000000e+00
## sum             1.446248e+06 2.467300e+04
## median          5.400000e+01 3.000000e+00
## mean            1.382250e+02 2.358119e+00
## SE.mean         2.562669e+00 7.379531e-03
## CI.mean.0.95    5.023320e+00 1.446529e-02
## var             6.871338e+04 5.697885e-01
## std.dev         2.621324e+02 7.548434e-01
## coef.var        1.896418e+00 3.201040e-01
```

```r
rating.den<-density(imdb.clean$imdbRating)
par(mfrow=c(2,2))
plot(rating.den,main="Kernel Density of Ratings")
boxplot(imdbRating~nrOfGenre, data=imdb.clean,main="Ratings by # of Genres")
hist(imdb.clean$year, main="Histogram of movies by release year")
plot(imdb.clean$imdbRating,imdb.clean$nrOfNominations, main="Ratings vs # of Nominations")
```




```r
cor(imdb.clean[c(3:12)])
```

```
##                   imdbRating ratingCount  duration        year   nrOfWins
## imdbRating        1.00000000   0.2097127 0.1814005 -0.19978015 0.29136615
## ratingCount       0.20971270   1.0000000 0.1967420  0.21481437 0.48764290
## duration          0.18140050   0.1967420 1.0000000  0.09613700 0.21419085
## year             -0.19978015   0.2148144 0.0961370  1.00000000 0.16485246
## nrOfWins          0.29136615   0.4876429 0.2141908  0.16485246 1.00000000
## nrOfNominations   0.24396243   0.5790339 0.2363915  0.23191922 0.79815679
## nrOfPhotos        0.07332029   0.6434044 0.2188236  0.22993201 0.31642696
## nrOfNewsArticles  0.09198266   0.6076978 0.1173968  0.18213279 0.33352253
## nrOfUserReviews   0.13476475   0.8297693 0.2034154  0.19854090 0.44729374
## nrOfGenre        -0.01921206   0.1216737 0.1045556  0.03120071 0.03906636
##                  nrOfNominations nrOfPhotos nrOfNewsArticles
## imdbRating             0.2439624 0.07332029       0.09198266
## ratingCount            0.5790339 0.64340444       0.60769776
## duration               0.2363915 0.21882365       0.11739678
## year                   0.2319192 0.22993201       0.18213279
## nrOfWins               0.7981568 0.31642696       0.33352253
## nrOfNominations        1.0000000 0.46915118       0.46434594
## nrOfPhotos             0.4691512 1.00000000       0.49092618
## nrOfNewsArticles       0.4643459 0.49092618       1.00000000
## nrOfUserReviews        0.5543486 0.64710098       0.51020749
## nrOfGenre              0.0910109 0.18656237       0.08291106
##                  nrOfUserReviews   nrOfGenre
## imdbRating             0.1347648 -0.01921206
```

```
## ratingCount          0.8297693  0.12167372
## duration             0.2034154  0.10455555
## year                 0.1985409  0.03120071
## nrOfWins             0.4472937  0.03906636
## nrOfNominations      0.5543486  0.09101090
## nrOfPhotos           0.6471010  0.18656237
## nrOfNewsArticles     0.5102075  0.08291106
## nrOfUserReviews      1.0000000  0.12528402
## nrOfGenre            0.1252840  1.00000000
```

```r
table(imdb.clean$nrOfGenre)
```

```
##
##    0    1    2    3
##   23 1709 3229 5502
```

Since movies can be considered multiple categories at the same time (i.e. Action-Drama not just Action or Drama), doing analysis by genre is difficult. We could recode all the dummy variables into a single factored vector where each level indicates a movie's genre, or genre combination. So the first few levels would indicate the solo genre's, and after that each combination of two or more genre's. Of course since there's no upper limit to how many genre's a movie can be, this would also create a large number of factors, however the cleaned data summary shows the most genres any movie has is 3.

That being said, there seems to be a slight negative correlation with the number of genre's a movie belongs to and it's imdbRating. The most strongly correlated variables (excluding individual genres) with imdbRating is the number of nominations and the number of award wins, although they are both small, around .25 and .29 respectively. The imdbRatings are centered around roughly 7, according to the density plot and summary statistics.

I'm most interested in if certain genre's tend to have higher/lower ratings. As mentioned before, doing this efficiently will require some thought since genre's are coded as dummy variables but can have multiple genre's simultaneously, a possible solution is a single variable with multiple factors for each genre and genre combination. I'm also interested in if the number of user reviews or articles have any effect on nominations or wins. This would help show a true "power to the people" relationship between award nominated/winning films and the people who pay to watch it. There's also a notion of horror movies historically never winning any awards, so I'd be interested in seeing if this data supports that with applicable significance tests.

Regression(Poisson): Can we predict winners or nominations based on the number of user reviews? Let's focus on just those movies that HAVE won awards or been nominated.

```r
Nom.only.imdb<-subset(imdb.clean,nrOfNominations>=1, select = 3:40)
Win.only.imdb<-subset(imdb.clean, nrOfWins>=1, select = 3:40)

nomination.fit <- glm(nrOfNominations~nrOfUserReviews,data=Nom.only.imdb,family=poisson())
win.fit <- glm(nrOfWins~nrOfUserReviews,data=Win.only.imdb,family=poisson())
summary(nomination.fit)
```

```
##
## Call:
## glm(formula = nrOfNominations ~ nrOfUserReviews, family = poisson(),
##     data = Nom.only.imdb)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -17.412   -2.406   -1.248    0.678   21.298
##
```

```
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.977e+00  5.611e-03   352.4   <2e-16 ***
## nrOfUserReviews 7.979e-04  6.036e-06   132.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 48298  on 4585  degrees of freedom
## Residual deviance: 38017  on 4584  degrees of freedom
## AIC: 54236
##
## Number of Fisher Scoring iterations: 5
```

```
summary(win.fit)
```

```
##
## Call:
## glm(formula = nrOfWins ~ nrOfUserReviews, family = poisson(),
##     data = Win.only.imdb)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -13.7301   -2.3589   -1.3818    0.2888   22.5626
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.680e+00  6.031e-03   278.6   <2e-16 ***
## nrOfUserReviews 7.993e-04  6.880e-06   116.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 50506  on 5323  degrees of freedom
## Residual deviance: 42629  on 5322  degrees of freedom
## AIC: 59405
##
## Number of Fisher Scoring iterations: 5
```

```
nomination.poly.fit <- glm(nrOfNominations~nrOfUserReviews+I(nrOfUserReviews^2),
                           data=Nom.only.imdb,family=poisson())
win.poly.fit <- glm(nrOfWins~nrOfUserReviews+I(nrOfUserReviews^2),
                    data=Win.only.imdb,family=poisson())

summary(nomination.poly.fit)
```

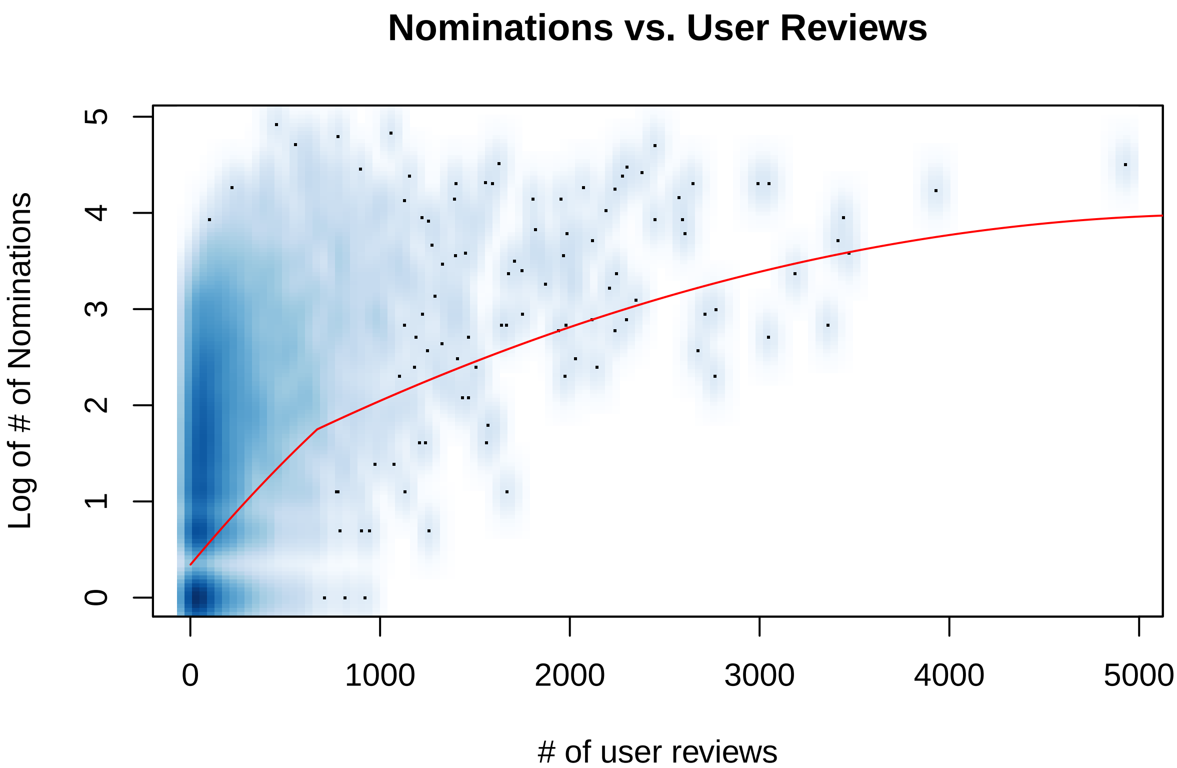
```
##
## Call:
## glm(formula = nrOfNominations ~ nrOfUserReviews + I(nrOfUserReviews^2),
##     family = poisson(), data = Nom.only.imdb)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -8.0576  -2.3031  -1.0768   0.7385  20.2591
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             1.749e+00  7.301e-03  239.56   <2e-16 ***
## nrOfUserReviews         1.851e-03  2.063e-05   89.72   <2e-16 ***
## I(nrOfUserReviews^2) -3.831e-07  8.281e-09  -46.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 48298  on 4585  degrees of freedom
## Residual deviance: 34708  on 4583  degrees of freedom
## AIC: 50929
##
## Number of Fisher Scoring iterations: 5
```

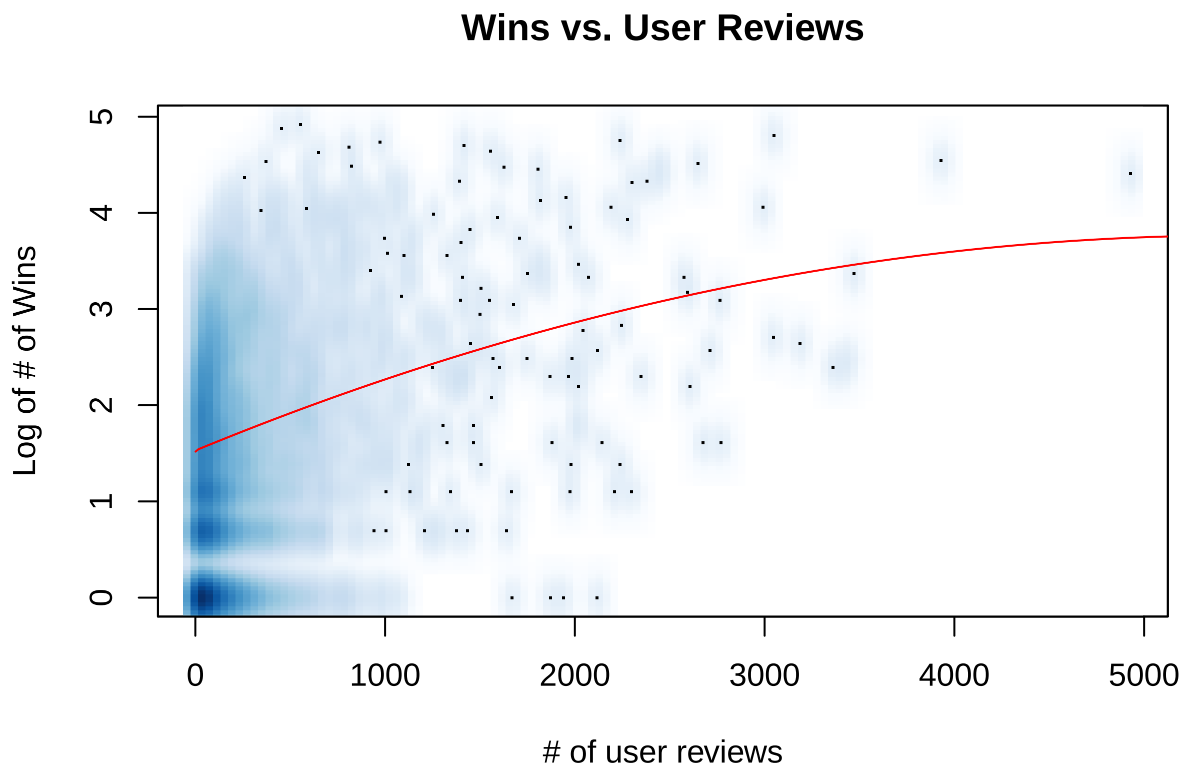
```r
summary(win.poly.fit)
```

```
##
## Call:
## glm(formula = nrOfWins ~ nrOfUserReviews + I(nrOfUserReviews^2),
##     family = poisson(), data = Win.only.imdb)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.1973  -2.1458  -1.3731   0.4728  21.3929
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             1.516e+00  7.500e-03  202.18   <2e-16 ***
## nrOfUserReviews         1.631e-03  2.197e-05   74.24   <2e-16 ***
## I(nrOfUserReviews^2) -2.956e-07  8.448e-09  -34.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 50506  on 5323  degrees of freedom
## Residual deviance: 40830  on 5321  degrees of freedom
## AIC: 57608
##
## Number of Fisher Scoring iterations: 5
```

```r
x.nom<-seq(1,5500,1)
y.nom<- predict(nomination.poly.fit,newdata=data.frame(nrOfUserReviews=x.nom))
y.win<- predict(win.poly.fit,newdata=data.frame(nrOfUserReviews=x.nom))
#y.nom<-(nomination.poly.fit$coefficients[1]
#        +nomination.poly.fit$coefficients[2]*Nom.win.only.imdb$nrOfUserReviews^2
#        +nomination.poly.fit$coefficients[3]*Nom.win.only.imdb$nrOfUserReviews^3)
smoothScatter(Nom.only.imdb$nrOfUserReviews, log(Nom.only.imdb$nrOfNominations),
              main="Nominations vs. User Reviews", xlab="# of user reviews",
              ylab="Log of # of Nominations")
lines(x.nom,sort(y.nom),col="red")
```




```
smoothScatter(Win.only.imdb$nrOfUserReviews, log(Win.only.imdb$nrOfWins),
              main="Wins vs. User Reviews", xlab="# of user reviews",
              ylab="Log of # of Wins")
lines(x.nom,sort(y.win),col="red")
```




```r
anova(nomination.fit,nomination.poly.fit, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: nrOfNominations ~ nrOfUserReviews
## Model 2: nrOfNominations ~ nrOfUserReviews + I(nrOfUserReviews^2)
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      4584      38017
## 2      4583      34708  1   3309.4 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(win.fit,win.poly.fit, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: nrOfWins ~ nrOfUserReviews
## Model 2: nrOfWins ~ nrOfUserReviews + I(nrOfUserReviews^2)
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      5322      42629
## 2      5321      40830  1     1799 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We are using a Poisson regression model:

$$log(\lambda) = \beta_0 + \beta_1(\text{numb. of User Reviews})$$

In the case of the polynomial models, we add $\beta_2(\text{numb. of User Reviews})^2$ to the models. The β_i are the coefficients for their respective variables, signifying an expected log change in Y for a unit change in the predictors (the rest held constant). The above output shows each model's respective coefficient estimates. This is a generalized linear model, so we are assuming a function of the mean, in this case $log(\lambda)$, is associated with the predictors.

The effects are individually significant compared to just empty models. The difference in deviance for nomination prediction and win prediction with the number of reviews and empty models are 10281 and 7877 respectively. Compared to a Chi-Squared distribution with 1 degree of freedom, these are obviously much larger, suggesting that the predictors are significantly different than 0.

It's also possible that "review bombing" is a factor in some of these movies, that is, the public thought it was so bad that those who would otherwise not write a review at all are compelled to write one so others do not see the movie. This could explain what seems to be a non-positive, non-linear relationship seen in the scatter plots. This lead to me using a degree 2 polynomial regression, and the p-values suggest they are significant to the model. The fits are overlaid the scatterplots for better visualization. The ANOVA output compares each single predictor model with the respective polynomial model, and the Chi-square test column shows that the result is significant, indicating the model with the degree 2 term is statistically better than the simple linear model, in both the nomination and win cases.

Regression: Are the genre's effects on rating different?

```
#(title,url,imdbRating,ratingCount,duration,year,nrOfWins,
# nrOfNominations,nrOfPhotos,nrOfNewsArticles,nrOfUserReviews,nrOfGenre)

genre.col<-c(seq(13,40))
imdb.clean[genre.col]<-lapply(imdb.clean[genre.col],as.factor)


genre.fit <- lm(imdbRating~Action+Adult+Adventure+Animation+Biography+Comedy+Crime+Documentary
                +Drama+Family+Fantasy+FilmNoir+GameShow+History+Horror+Music+Musical+Mystery
                +News+RealityTV+Romance+SciFi+Short+Sport+TalkShow+Thriller+War+Western, data=imdb.clean
summary(genre.fit)
```

```
##
## Call:
## lm(formula = imdbRating ~ Action + Adult + Adventure + Animation +
##     Biography + Comedy + Crime + Documentary + Drama + Family +
##     Fantasy + FilmNoir + GameShow + History + Horror + Music +
##     Musical + Mystery + News + RealityTV + Romance + SciFi +
##     Short + Sport + TalkShow + Thriller + War + Western, data = imdb.clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2137 -0.4927  0.1040  0.6355  3.8157
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.690527   0.036768 181.968  < 2e-16 ***
## Action1      -0.272484   0.029078  -9.371  < 2e-16 ***
## Adult1       -0.715584   0.245376  -2.916 0.003550 **
## Adventure1   -0.057851   0.031750  -1.822 0.068468 .
## Animation1    0.537043   0.047510  11.304  < 2e-16 ***
## Biography1    0.151577   0.045977   3.297 0.000981 ***
## Comedy1      -0.215347   0.026495  -8.128 4.87e-16 ***
## Crime1       -0.003184   0.029679  -0.107 0.914557
## Documentary1  0.406655   0.047114   8.631  < 2e-16 ***
```

```
## Drama1        0.398857   0.026049  15.312  < 2e-16 ***
## Family1      -0.369174   0.042270  -8.734  < 2e-16 ***
## Fantasy1     -0.187365   0.040567  -4.619 3.91e-06 ***
## FilmNoir1     0.277601   0.074673   3.718 0.000202 ***
## GameShow1     0.724820   0.691359   1.048 0.294479
## History1      0.106645   0.049095   2.172 0.029861 *
## Horror1      -0.646763   0.039102 -16.540  < 2e-16 ***
## Music1        0.069887   0.062029   1.127 0.259904
## Musical1      0.209258   0.056203   3.723 0.000198 ***
## Mystery1      0.061217   0.039411   1.553 0.120382
## News1         0.124880   0.691867   0.180 0.856766
## RealityTV1    0.409473   0.691827   0.592 0.553949
## Romance1      0.018613   0.028567   0.652 0.514708
## SciFi1       -0.335500   0.040965  -8.190 2.92e-16 ***
## Short1       -0.144115   0.053334  -2.702 0.006901 **
## Sport1       -0.142855   0.070041  -2.040 0.041416 *
## TalkShow1     1.423395   0.564899   2.520 0.011759 *
## Thriller1    -0.017320   0.033710  -0.514 0.607411
## War1          0.275717   0.049536   5.566 2.67e-08 ***
## Western1      0.036378   0.061266   0.594 0.552678
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.977 on 10434 degrees of freedom
## Multiple R-squared:  0.1656, Adjusted R-squared:  0.1634
## F-statistic: 73.96 on 28 and 10434 DF,  p-value: < 2.2e-16
```

Model: $Y = \beta_0 + \sum_{i=1}^{28} \beta_i(X_i)$ where each coefficient β_i is the expected change in Y, the imdbRating, for belonging to their respective genre, assuming all other genres are held "constant". This is a simple linear regression, so we have the standard 4 assumptions: errors are distributed iid normal with mean 0, each movie is independent of each other, there is a linear relationship between the genres and the imdbRating, and the variance of the random errors do not vary across the 2 levels of genre.

The adj-R^2 is only .1634, meaning only roughly 16% of the variability in imdbRatings is explained by the various genres. Of them, only a few are considered statistically significant at the 5% level, namely any genre with 1 or more stars (*) in the above output. Not all of them are, suggesting that the effects of each genre do vary, which is supported by the F-statistic of 73.96, with a p-value of essentially 0. So we would reject the null hypothesis that each genre's effect=0. At LEAST one is not.

ANOVA:Are horror movies significantly different in rating?

```r
genre.aov <- aov(imdbRating~Action+Adult+Adventure+Animation+Biography+Comedy+Crime+Documentary
                 +Drama+Family+Fantasy+FilmNoir+GameShow+History+Horror+Music+Musical+Mystery
                 +News+RealityTV+Romance+SciFi+Short+Sport+TalkShow+Thriller+War+Western, data=imdb.clean
summary(genre.aov)
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)
## Action          1    181   180.5 189.121  < 2e-16 ***
## Adult           1     10    10.0  10.521  0.00118 **
## Adventure       1     15    14.9  15.578 7.97e-05 ***
## Animation       1     36    36.0  37.678 8.65e-10 ***
## Biography       1    107   106.5 111.591  < 2e-16 ***
## Comedy          1    189   189.1 198.132  < 2e-16 ***
## Crime           1     16    15.7  16.460 5.00e-05 ***
## Documentary     1     34    33.8  35.366 2.82e-09 ***
## Drama           1    790   790.0 827.655  < 2e-16 ***
```

```
## Family          1     55    55.2  57.837 3.09e-14 ***
## Fantasy         1     16    15.6  16.342 5.33e-05 ***
## FilmNoir        1     18    18.5  19.369 1.09e-05 ***
## GameShow        1      1     1.4   1.497  0.22120
## History         1     18    18.3  19.215 1.18e-05 ***
## Horror          1    348   348.3 364.893  < 2e-16 ***
## Music           1      2     1.6   1.656  0.19816
## Musical         1     17    16.6  17.379 3.09e-05 ***
## Mystery         1      3     3.0   3.116  0.07754 .
## News            1      0     0.0   0.011  0.91716
## RealityTV       1      0     0.4   0.461  0.49714
## Romance         1      1     1.4   1.456  0.22758
## SciFi           1     70    70.2  73.563  < 2e-16 ***
## Short           1      8     7.8   8.197  0.00421 **
## Sport           1      5     5.3   5.511  0.01892 *
## TalkShow        1      6     5.9   6.197  0.01281 *
## Thriller        1      1     1.0   1.012  0.31444
## War             1     29    29.3  30.660 3.15e-08 ***
## Western         1      0     0.3   0.353  0.55268
## Residuals   10434   9960     1.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(genre.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = imdbRating ~ Action + Adult + Adventure + Animation + Biography + Comedy + Crime +
##
## $Action
##           diff        lwr        upr p adj
## 1-0 -0.3484983 -0.3981723 -0.2988243     0
##
## $Adult
##           diff       lwr        upr     p adj
## 1-0 -0.7927359 -1.271883 -0.3135888 0.0011862
##
## $Adventure
##           diff        lwr         upr    p adj
## 1-0 -0.1046941 -0.1585878 -0.05080052 0.000141
##
## $Animation
##          diff       lwr       upr p adj
## 1-0 0.2577257 0.1730992 0.3423522     0
##
## $Biography
##          diff       lwr       upr p adj
## 1-0 0.4536163 0.3692112 0.5380215     0
##
## $Comedy
##           diff       lwr       upr p adj
## 1-0 -0.2832125 -0.323321 -0.243104     0
##
## $Crime
```

```
##          diff       lwr       upr    p adj
## 1-0 0.1034963 0.0521123 0.1548803 7.93e-05
##
## $Documentary
##          diff       lwr       upr p adj
## 1-0 0.2082705 0.1369897 0.2795512     0
##
## $Drama
##          diff       lwr       upr p adj
## 1-0 0.4794308 0.4419809 0.5168806     0
##
## $Family
##           diff       lwr        upr p adj
## 1-0 -0.2779745 -0.353616 -0.2023331     0
##
## $Fantasy
##           diff        lwr         upr    p adj
## 1-0 -0.1515192 -0.2270074 -0.07603102 8.39e-05
##
## $FilmNoir
##          diff       lwr       upr    p adj
## 1-0 0.3038633 0.1636467 0.4440798 2.18e-05
##
## $GameShow
##          diff        lwr      upr     p adj
## 1-0 0.8449929 -0.5093327 2.199319 0.2213552
##
## $History
##          diff       lwr       upr    p adj
## 1-0 0.1965413 0.1055945 0.2874881 2.29e-05
##
## $Horror
##           diff        lwr       upr p adj
## 1-0 -0.5744013 -0.6400337 -0.508769     0
##
## $Music
##          diff         lwr       upr     p adj
## 1-0 0.0761626 -0.04213508 0.1944603 0.2069721
##
## $Musical
##          diff       lwr       upr    p adj
## 1-0 0.2232738 0.1164594 0.3300881 4.21e-05
##
## $Mystery
##           diff         lwr      upr     p adj
## 1-0 0.06221679 -0.01018541 0.134619 0.0921275
##
## $News
##           diff       lwr      upr     p adj
## 1-0 0.07180024 -1.282525 1.426126 0.9172344
##
## $RealityTV
##          diff      lwr      upr     p adj
## 1-0 0.4688056 -0.88552 1.823131 0.4974516
```

```
##
## $Romance
##            diff         lwr        upr     p adj
## 1-0 0.02874419 -0.02178376 0.07927214 0.2648299
##
## $SciFi
##            diff       lwr      upr p adj
## 1-0 -0.2970365 -0.370343 -0.22373     0
##
## $Short
##            diff        lwr         upr     p adj
## 1-0 -0.1324341 -0.2293796 -0.03548867 0.0074233
##
## $Sport
##            diff        lwr         upr     p adj
## 1-0 -0.1598555 -0.2949434 -0.02476757 0.0203828
##
## $TalkShow
##          diff       lwr      upr     p adj
## 1-0 1.402942 0.2970869 2.508797 0.0129055
##
## $Thriller
##             diff         lwr        upr     p adj
## 1-0 -0.02643362 -0.08436019 0.03149294 0.3710783
##
## $War
##           diff       lwr       upr p adj
## 1-0 0.2511219 0.1583419 0.3439018 1e-07
##
## $Western
##            diff         lwr       upr     p adj
## 1-0 0.03205891 -0.08067938 0.1447972 0.5772584
```

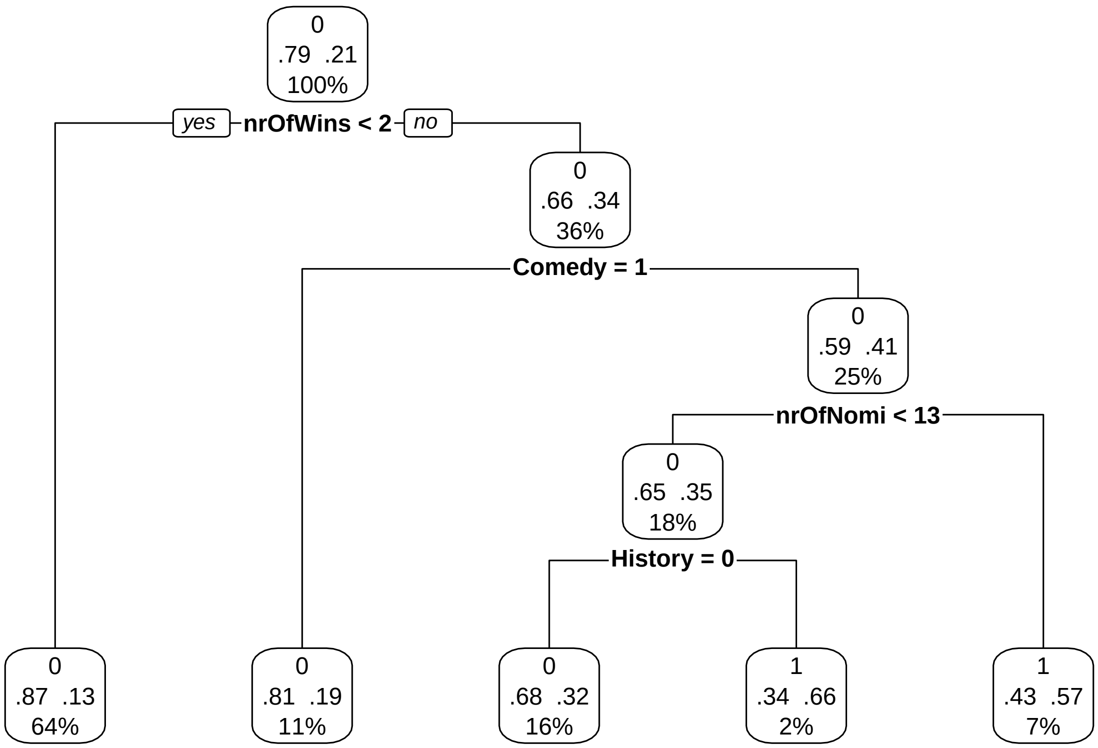
In ANOVA we assume the response, imdbRatings, is distributed normal within each genre, and the variances are equal across the genres. The Tukey HSD output, which uses the ANOVA model, suggests that horror movies have a statistically significant difference in imdbRatings. The contrast being calculated is horror movies - non horror movies, with a difference of -.57, suggesting that on average horror movies have a .57 reduction in imdbRating, all other genres held "constant". This dataset suggests there is some truth to horror movies being less well received.

Classification: Long movies, defined as being 2 hours or longer- can we classify it by the other predictors? Is there a tendency for movies to be shorter over time as society prefers shorter more digestible content?

```
imdb.clean$Long<- rep(0,nrow(imdb.clean))
imdb.clean$Long[imdb.clean$duration >=7200]<- 1
imdb.clean$Long<-as.factor(imdb.clean$Long)
#leave year as numeric, see the trend as it increases

library(rpart)
set.seed(42)
train<- sample(nrow(imdb.clean),.75*nrow(imdb.clean))
imdb.clean.train<-imdb.clean[train,]
imdb.clean.test<-imdb.clean[-train,]
long.class.tree<- rpart(Long~imdbRating+ratingCount+year+nrOfWins+nrOfNominations
                        +nrOfPhotos+nrOfNewsArticles+nrOfUserReviews
```

```r
                        +Action+Adult+Adventure+Animation+Biography+Comedy+Crime+Documentary
                +Drama+Family+Fantasy+FilmNoir+GameShow+History+Horror+Music+Musical+Mystery
                +News+RealityTV+Romance+SciFi+Short+Sport+TalkShow+Thriller+War+Western,
                data=imdb.clean.train, method="class",parms=list(split="information"))
#(summary(long.class.tree)
#plotcp(long.class.tree)
pruned.long.class.tree<-prune(long.class.tree,cp=.012)
library(rpart.plot)
prp(pruned.long.class.tree,type=2,extra=104,fallen.leaves=TRUE)
```




```r
pruned.tree.prob<-predict(pruned.long.class.tree,imdb.clean.test,type="class")
pruned.tree.pred<-table(imdb.clean.test$Long,pruned.tree.prob,dnn=c("Actual","predicted"))
pruned.tree.pred
```

```
##       predicted
## Actual    0    1
##      0 1979   92
##      1  398  147
```

```r
performance <- function(table, n=2){
if(!all(dim(table) == c(2,2)))
stop("Must be a 2 x 2 table")
tn = table[1,1]
fp = table[1,2]
fn = table[2,1]
tp = table[2,2]
sensitivity = tp/(tp+fn)
```

```r
specificity = tn/(tn+fp)
ppp = tp/(tp+fp)
npp = tn/(tn+fn)
hitrate = (tp+tn)/(tp+tn+fp+fn)
result <- paste("Sensitivity = ", round(sensitivity, n) ,
"\nSpecificity = ", round(specificity, n),
"\nPositive Predictive Value = ", round(ppp, n),
"\nNegative Predictive Value = ", round(npp, n),
"\nAccuracy = ", round(hitrate, n), "\n", sep="")
cat(result)
}
performance(pruned.tree.pred)
```

```
## Sensitivity = 0.27
## Specificity = 0.96
## Positive Predictive Value = 0.62
## Negative Predictive Value = 0.83
## Accuracy = 0.81
```

The pruned classification tree doesn't suggest year as a very good classifier for long movies, in fact it doesn't even use it as a splitting variable. Compared to the other variables, specifically the number of nominations, and genres, year doesn't split the data well. Using the performance function defined in class, we can see that the pruned tree does not perform very well. It has a very low sensitivity, and PPV at .26 and .57 respectively, meaning it misclassified a majority of long movies, and of those that are classified as long, only 57% of them are actually long.

```r
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```r
set.seed(42)
forest.fit <- randomForest(Long~imdbRating+ratingCount+year+nrOfWins+nrOfNominations
                        +nrOfPhotos+nrOfNewsArticles+nrOfUserReviews
                        +Action+Adult+Adventure+Animation+Biography+Comedy+Crime+Documentary
                +Drama+Family+Fantasy+FilmNoir+GameShow+History+Horror+Music+Musical+Mystery
                +News+RealityTV+Romance+SciFi+Short+Sport+TalkShow+Thriller+War+Western,
                data=imdb.clean.train, importance=TRUE)
forest.fit
```

```
##
## Call:
##  randomForest(formula = Long ~ imdbRating + ratingCount + year +      nrOfWins + nrOfNominations + n
##               Type of random forest: classification
##                     Number of trees: 500
## No. of variables tried at each split: 6
##
##         OOB estimate of  error rate: 16.97%
## Confusion matrix:
##      0   1 class.error
## 0 6016 218  0.03496952
## 1 1114 499  0.69063856
```

```r
importance(forest.fit,type=2)
```

```
##                   MeanDecreaseGini
```

```
## imdbRating           230.72709274
## ratingCount          261.07001060
## year                 238.70309045
## nrOfWins             175.57175579
## nrOfNominations      180.01886036
## nrOfPhotos           212.77816241
## nrOfNewsArticles     188.46866350
## nrOfUserReviews      236.69653073
## Action                35.41160769
## Adult                  0.76306664
## Adventure             24.42207819
## Animation             14.88769108
## Biography             24.10133858
## Comedy                51.86015202
## Crime                 25.65457788
## Documentary           11.80270304
## Drama                 92.18600331
## Family                12.58711164
## Fantasy               16.97025942
## FilmNoir               4.51452824
## GameShow               0.92106666
## History               57.98972470
## Horror                17.39878307
## Music                 13.58282683
## Musical               36.05501506
## Mystery               17.19122810
## News                   0.05264584
## RealityTV              0.00000000
## Romance               32.77476635
## SciFi                 10.58965700
## Short                  7.31844479
## Sport                  7.07875960
## TalkShow               0.76430963
## Thriller              19.84794261
## War                   19.12003054
## Western               12.21558603
```

```
forest.prob<-predict(forest.fit,imdb.clean.test,type="class")
forest.pred<-table(imdb.clean.test$Long,forest.prob,dnn=c("Actual","predicted"))
forest.pred
```

```
##       predicted
## Actual    0    1
##      0 2006   65
##      1  364  181
```

```
performance(forest.pred)
```

```
## Sensitivity = 0.33
## Specificity = 0.97
## Positive Predictive Value = 0.74
## Negative Predictive Value = 0.85
## Accuracy = 0.84
```

The random forest also doesn't seem to perform so well. It has a low sensitivity just like the pruned tree, but does accurately predict movies as "long" better than the pruned tree.

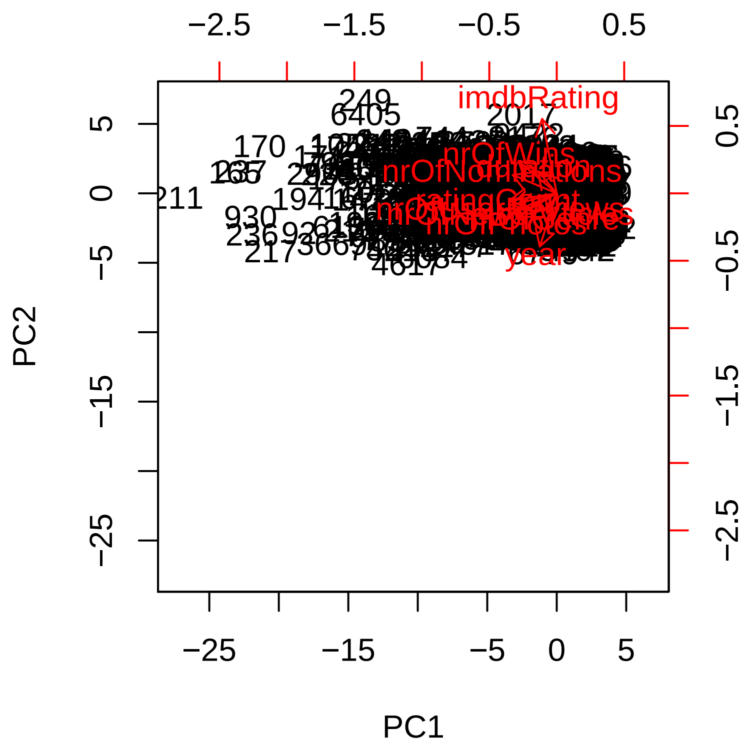
The most important variable, based on Gini index, is the ratingCount, followed by year, suggesting year is actually a very important variable in classifying movies as "long".

Principal Component Analysis:

```
pr.out <-prcomp(imdb.clean[,3:12],scale=TRUE)
pr.out$rotation
```

```
##                          PC1         PC2          PC3         PC4
## imdbRating       -0.13578507  0.69003733 -0.094544984 -0.09841140
## ratingCount      -0.43750675 -0.06946799  0.007466662 -0.23722915
## duration         -0.17148001  0.20355156 -0.392771611  0.59151166
## year             -0.15685587 -0.49340156  0.256397932  0.56467875
## nrOfWins         -0.35448358  0.29912473  0.228173436  0.26537214
## nrOfNominations  -0.40707102  0.17035002  0.174721528  0.20057023
## nrOfPhotos       -0.37478644 -0.21918688 -0.150233762 -0.16844030
## nrOfNewsArticles -0.34595300 -0.14777826  0.068496355 -0.27626951
## nrOfUserReviews  -0.41979456 -0.10980432 -0.019515149 -0.22112558
## nrOfGenre        -0.09165067 -0.18360641 -0.812873024  0.03232798
##                          PC5         PC6          PC7         PC8
## imdbRating        0.05081980 -0.68778741 -0.007097570  0.06526676
## ratingCount       0.12249235 -0.06547914  0.149802040 -0.39004103
## duration          0.58665926  0.24217946 -0.123366802 -0.08153799
## year             -0.03694577 -0.58461279  0.001536761 -0.05138606
## nrOfWins         -0.41974041  0.24852239  0.031572883  0.01300594
## nrOfNominations  -0.33263462  0.21534281 -0.032769252  0.16204237
## nrOfPhotos        0.21099608 -0.02153849  0.278115345  0.78721082
## nrOfNewsArticles  0.10715434 -0.01805710 -0.858624892  0.01298975
## nrOfUserReviews   0.14448350  0.04851504  0.375391681 -0.42546396
## nrOfGenre        -0.52266070 -0.10468224 -0.068649904 -0.08368518
##                           PC9        PC10
## imdbRating       -0.049835571  0.06602614
## ratingCount       0.254801784 -0.69902798
## duration          0.009048068 -0.01536643
## year              0.008487394  0.03869250
## nrOfWins          0.608902628  0.22914296
## nrOfNominations  -0.695502536 -0.25776806
## nrOfPhotos        0.134556114  0.02030516
## nrOfNewsArticles  0.029117241  0.16552991
## nrOfUserReviews  -0.242635588  0.59866390
## nrOfGenre         0.011277666  0.01026691
```

```
biplot(pr.out,scale=0)
```




```r
pr.var<-pr.out$sdev^2
pr.var
```

```
##  [1] 4.0064350 1.3024583 1.0372999 0.9714679 0.8205488 0.5968967 0.5412869
##  [8] 0.3899444 0.1823793 0.1512826
```

```r
prop.var.explain<-pr.var/sum(pr.var)
prop.var.explain
```

```
##  [1] 0.40064350 0.13024583 0.10372999 0.09714679 0.08205488 0.05968967
##  [7] 0.05412869 0.03899444 0.01823793 0.01512826
```

The procomp function uses the “scale” argument since the variables all have different units, so to run genuine PCA, we scale each to have standard deviation 0. The biplot, although very messy, shows us that the loading for imdbRating on the first component is about -.15 and .7 for the second. All the black numbers are the movies index, while the red are the numeric variables. The scaling here opposite the “PC1” and “PC2” labels indicate how much the variable is loaded for that respective component, which is why imdbRating is centered roughly at (-.15,.7) indicating the “weight”, loosely speaking, of that variable in (PC1,PC2). So imdbRatings explains a lot of variation in PC2, and less in PC1. The negative sign of course does not indicate a negative variability, which is not possible, PCA is a purely mathematical and unsupervised technique. The negative value is indicating a negative correlation in that PC. The output of the rotation matrix gives us the actual loadings more precisely for each variable and the respective PC.

The “prop.var.explain” line is the amount of variability explained in the data by the respective principal component, the first number PC1, and so on. The first component explains 40% of the variability, the next about 13%, and so on. We can explain roughly 73% of the variability in the data using only the first 4 PC together.