R Project

## R Markdown

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.6.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.6.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(statsr)

## Warning: package 'statsr' was built under R version 3.6.3

## Loading required package: BayesFactor

## Warning: package 'BayesFactor' was built under R version 3.6.3

## Loading required package: coda

## Warning: package 'coda' was built under R version 3.6.3

## Loading required package: Matrix

## \*\*\*\*\*\*\*\*\*\*\*\*  
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard Morey (richarddmorey@gmail.com).  
##   
## Type BFManual() to open the manual.  
## \*\*\*\*\*\*\*\*\*\*\*\*

library(gridExtra)

## Warning: package 'gridExtra' was built under R version 3.6.3

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

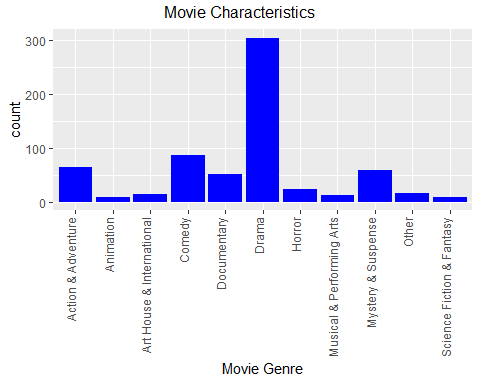
library(knitr)

## Warning: package 'knitr' was built under R version 3.6.3

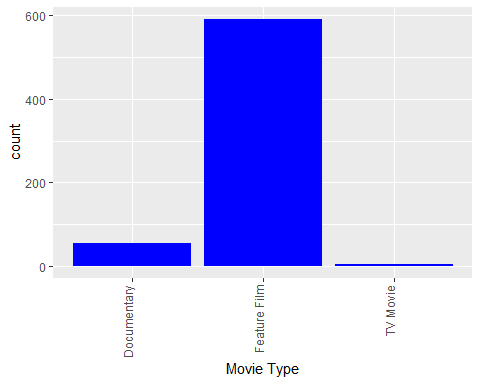
load("C:/Users/Sahil Shah/Desktop/movies.RData")  
head(movies)

## # A tibble: 6 x 32  
## title title\_type genre runtime mpaa\_rating studio thtr\_rel\_year thtr\_rel\_month  
## <chr> <fct> <fct> <dbl> <fct> <fct> <dbl> <dbl>  
## 1 Fill~ Feature F~ Drama 80 R Indom~ 2013 4  
## 2 The ~ Feature F~ Drama 101 PG-13 Warne~ 2001 3  
## 3 Wait~ Feature F~ Come~ 84 R Sony ~ 1996 8  
## 4 The ~ Feature F~ Drama 139 PG Colum~ 1993 10  
## 5 Male~ Feature F~ Horr~ 90 R Ancho~ 2004 9  
## 6 Old ~ Documenta~ Docu~ 78 Unrated Shcal~ 2009 1  
## # ... with 24 more variables: thtr\_rel\_day <dbl>, dvd\_rel\_year <dbl>,  
## # dvd\_rel\_month <dbl>, dvd\_rel\_day <dbl>, imdb\_rating <dbl>,  
## # imdb\_num\_votes <int>, critics\_rating <fct>, critics\_score <dbl>,  
## # audience\_rating <fct>, audience\_score <dbl>, best\_pic\_nom <fct>,  
## # best\_pic\_win <fct>, best\_actor\_win <fct>, best\_actress\_win <fct>,  
## # best\_dir\_win <fct>, top200\_box <fct>, director <chr>, actor1 <chr>,  
## # actor2 <chr>, actor3 <chr>, actor4 <chr>, actor5 <chr>, imdb\_url <chr>,  
## # rt\_url <chr>

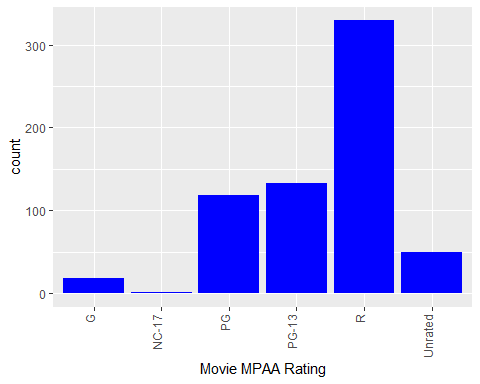
p1 <- ggplot(data=movies, aes(x=genre)) +   
 geom\_bar(fill="blue") +   
 xlab("Movie Genre") +  
 theme(axis.text.x=element\_text(angle=90, hjust=1, vjust=0))  
grid.arrange(p1,  
 top="Movie Characteristics")



p2 <- ggplot(data=movies, aes(x=title\_type)) +   
 geom\_bar(fill="blue") +   
 xlab("Movie Type") +  
 theme(axis.text.x=element\_text(angle=90, hjust=1, vjust=0))  
grid.arrange(p2)

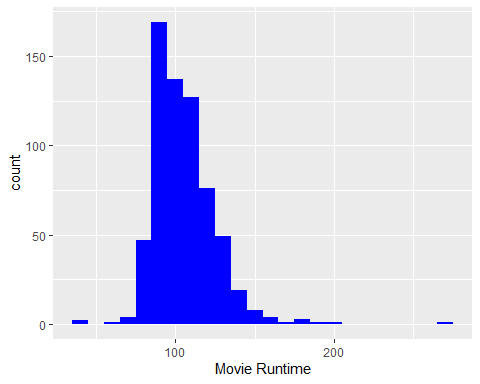


p3 <- ggplot(data=movies, aes(x=mpaa\_rating)) +   
 geom\_bar(fill="blue") +   
 xlab("Movie MPAA Rating") +  
 theme(axis.text.x=element\_text(angle=90, hjust=1, vjust=0))  
grid.arrange(p3)



p4 <- ggplot(data=movies, aes(x=runtime)) +   
 geom\_histogram(binwidth=10, fill="blue") +  
 xlab("Movie Runtime")  
grid.arrange(p4)

## Warning: Removed 1 rows containing non-finite values (stat\_bin).



movies <- movies %>% filter(title\_type=="Feature Film") %>%  
 filter(!(mpaa\_rating %in% c("NC-17", "Unrated")))  
summary(movies$audience\_score)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 11.00 44.00 62.00 60.17 77.00 97.00

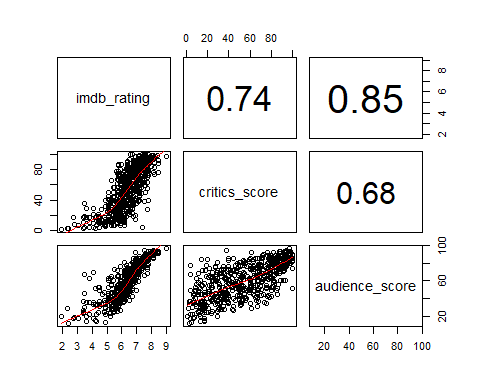
summary(movies$critics\_score)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 31.00 56.00 54.12 79.00 100.00

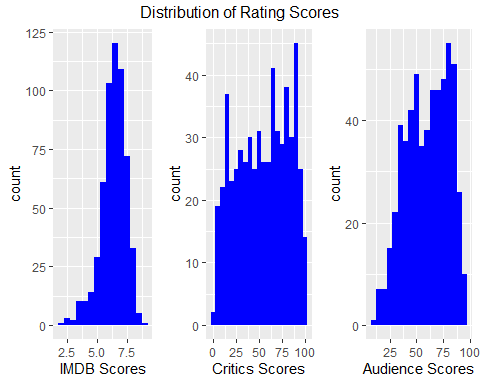
summary(movies$imdb\_rating)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.900 5.800 6.500 6.371 7.100 9.000

# Helper function for adding correlation coeficient values to a pairwise plot  
# (taken from pairs() help page).  
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)  
{  
 usr <- par("usr"); on.exit(par(usr))  
 par(usr = c(0, 1, 0, 1))  
 r <- abs(cor(x, y))  
 txt <- format(c(r, 0.123456789), digits = digits)[1]  
 txt <- paste0(prefix, txt)  
 if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)  
 text(0.5, 0.5, txt, cex = cex.cor \* r)  
}  
  
# Create pairwise plots of the movie rating scores to test for collinearity.  
# Using the helper function above, the linear correlation R value is included  
# on the chart.  
pairs(~ imdb\_rating + critics\_score + audience\_score,   
 data=movies, lower.panel=panel.smooth, upper.panel=panel.cor)



p1 <- ggplot(data=movies, aes(x=imdb\_rating)) +   
 geom\_histogram(binwidth=0.5, fill="blue") +  
 xlab("IMDB Scores")  
p2 <- ggplot(data=movies, aes(x=critics\_score)) +   
 geom\_histogram(binwidth=5, fill="blue") +  
 xlab("Critics Scores")  
p3 <- ggplot(data=movies, aes(x=audience\_score)) +   
 geom\_histogram(binwidth=5, fill="blue") +  
 xlab("Audience Scores")  
grid.arrange(p1, p2, p3, nrow=1,  
 top="Distribution of Rating Scores")



fullMod <- lm(imdb\_rating ~ genre + runtime + mpaa\_rating + thtr\_rel\_month +   
 best\_actor\_win + best\_actress\_win + best\_dir\_win, data=movies)  
summary(fullMod)

##   
## Call:  
## lm(formula = imdb\_rating ~ genre + runtime + mpaa\_rating + thtr\_rel\_month +   
## best\_actor\_win + best\_actress\_win + best\_dir\_win, data = movies)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.8670 -0.5307 0.0430 0.6227 1.9998   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.033861 0.376917 13.355 < 2e-16 \*\*\*  
## genreAnimation -0.306365 0.373898 -0.819 0.412921   
## genreArt House & International 0.386649 0.321918 1.201 0.230233   
## genreComedy -0.071803 0.158142 -0.454 0.649977   
## genreDocumentary 0.790707 0.675727 1.170 0.242441   
## genreDrama 0.593064 0.134232 4.418 1.20e-05 \*\*\*  
## genreHorror -0.113862 0.241935 -0.471 0.638089   
## genreMusical & Performing Arts 0.922245 0.353553 2.609 0.009339 \*\*   
## genreMystery & Suspense 0.384125 0.175588 2.188 0.029113 \*   
## genreOther 0.736434 0.279093 2.639 0.008558 \*\*   
## genreScience Fiction & Fantasy -0.266850 0.333834 -0.799 0.424431   
## runtime 0.015921 0.002688 5.923 5.56e-09 \*\*\*  
## mpaa\_ratingPG -0.790634 0.284121 -2.783 0.005574 \*\*   
## mpaa\_ratingPG-13 -1.057501 0.289001 -3.659 0.000277 \*\*\*  
## mpaa\_ratingR -0.715151 0.282431 -2.532 0.011613 \*   
## thtr\_rel\_month 0.006145 0.011526 0.533 0.594129   
## best\_actor\_winyes -0.044683 0.114659 -0.390 0.696908   
## best\_actress\_winyes 0.065389 0.124816 0.524 0.600569   
## best\_dir\_winyes 0.358104 0.155371 2.305 0.021545 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9365 on 554 degrees of freedom  
## Multiple R-squared: 0.248, Adjusted R-squared: 0.2236   
## F-statistic: 10.15 on 18 and 554 DF, p-value: < 2.2e-16

anova(fullMod)

## Analysis of Variance Table  
##   
## Response: imdb\_rating  
## Df Sum Sq Mean Sq F value Pr(>F)   
## genre 10 95.24 9.524 10.8585 < 2.2e-16 \*\*\*  
## runtime 1 41.21 41.211 46.9851 1.913e-11 \*\*\*  
## mpaa\_rating 3 18.54 6.181 7.0468 0.0001178 \*\*\*  
## thtr\_rel\_month 1 0.25 0.252 0.2869 0.5924429   
## best\_actor\_win 1 0.10 0.096 0.1093 0.7410594   
## best\_actress\_win 1 0.27 0.272 0.3101 0.5778658   
## best\_dir\_win 1 4.66 4.659 5.3122 0.0215453 \*   
## Residuals 554 485.92 0.877   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

finalMod <- lm(imdb\_rating ~ genre + runtime + mpaa\_rating +   
 best\_dir\_win, data=movies)  
summary(finalMod)

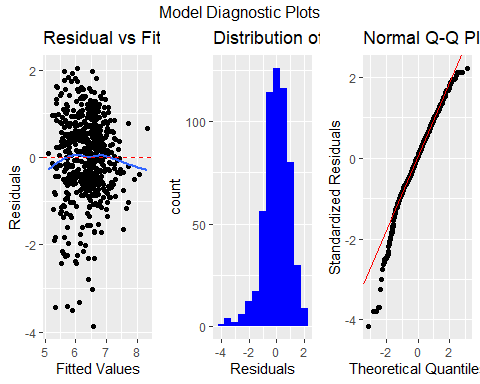
##   
## Call:  
## lm(formula = imdb\_rating ~ genre + runtime + mpaa\_rating + best\_dir\_win,   
## data = movies)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.8689 -0.5436 0.0404 0.6056 2.0432   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.035253 0.368299 13.672 < 2e-16 \*\*\*  
## genreAnimation -0.290502 0.371977 -0.781 0.435153   
## genreArt House & International 0.392689 0.320509 1.225 0.221017   
## genreComedy -0.059992 0.156688 -0.383 0.701958   
## genreDocumentary 0.779294 0.673510 1.157 0.247742   
## genreDrama 0.599714 0.132672 4.520 7.55e-06 \*\*\*  
## genreHorror -0.107019 0.241248 -0.444 0.657499   
## genreMusical & Performing Arts 0.924963 0.352691 2.623 0.008965 \*\*   
## genreMystery & Suspense 0.382923 0.173083 2.212 0.027347 \*   
## genreOther 0.731349 0.277479 2.636 0.008631 \*\*   
## genreScience Fiction & Fantasy -0.263124 0.332956 -0.790 0.429709   
## runtime 0.016264 0.002488 6.537 1.42e-10 \*\*\*  
## mpaa\_ratingPG -0.790039 0.283350 -2.788 0.005481 \*\*   
## mpaa\_ratingPG-13 -1.060774 0.288355 -3.679 0.000257 \*\*\*  
## mpaa\_ratingR -0.715358 0.281770 -2.539 0.011394 \*   
## best\_dir\_winyes 0.359052 0.155015 2.316 0.020907 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9346 on 557 degrees of freedom  
## Multiple R-squared: 0.2471, Adjusted R-squared: 0.2268   
## F-statistic: 12.19 on 15 and 557 DF, p-value: < 2.2e-16

anova(finalMod)

## Analysis of Variance Table  
##   
## Response: imdb\_rating  
## Df Sum Sq Mean Sq F value Pr(>F)   
## genre 10 95.24 9.524 10.9040 < 2.2e-16 \*\*\*  
## runtime 1 41.21 41.211 47.1820 1.735e-11 \*\*\*  
## mpaa\_rating 3 18.54 6.181 7.0763 0.0001129 \*\*\*  
## best\_dir\_win 1 4.69 4.686 5.3650 0.0209069 \*   
## Residuals 557 486.51 0.873   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Supplement the model data to make it easier to produce the diagnostic plots.  
pMod <- fortify(finalMod)  
  
# Create residuals scatter plot.  
p1 <- ggplot(pMod, aes(x=.fitted, y=.resid))+geom\_point() +  
 geom\_smooth(se=FALSE)+geom\_hline(yintercept=0, col="red", linetype="dashed") +  
 xlab("Fitted Values")+ylab("Residuals") +  
 ggtitle("Residual vs Fitted Plot")  
  
# The following is a bunch of extra code to get around ggplot not being able  
# to automatically draw a normal distribution line on a QQ plot.  
# This code comes from a blog post at http://mgimond.github.io/ES218/Week06a.html  
pMod$.qqnorm <- qqnorm(pMod$.stdresid, plot.it=FALSE)$x   
y <- quantile(pMod$.stdresid, c(0.25, 0.75)) # Find the 1st and 3rd quartiles  
x <- quantile(pMod$.qqnorm, c(0.25, 0.75)) # Find the 1st and 3rd quartiles  
slope <- diff(y) / diff(x) # Compute the line slope  
int <- y[1] - slope \* x[1] # Compute the line intercept  
  
# Create residuals QQ plot.  
p2 <- ggplot(pMod, aes(.qqnorm, .stdresid)) +  
 geom\_point(na.rm = TRUE) +  
 geom\_abline(intercept=int, slope=slope, color="red") +  
 xlab("Theoretical Quantiles")+ylab("Standardized Residuals") +  
 ggtitle("Normal Q-Q Plot")  
  
# Create residuals histogram plot.  
p3 <- ggplot(data=pMod, aes(x=.resid)) +   
 geom\_histogram(binwidth=0.5, fill="blue") +  
 xlab("Residuals") +  
 ggtitle("Distribution of Residuals")  
  
grid.arrange(p1, p3, p2, nrow=1, top="Model Diagnostic Plots")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



# Use the final model to generate rating predictions for Dirty Grandpa released  
# in January 2016 and for Deadpool released in February 2016.  
dataDG <- data.frame(genre="Comedy", runtime=102, mpaa\_rating="R", best\_dir\_win="no")  
predDG <- predict(finalMod, dataDG, interval="predict")  
  
dataDead <- data.frame(genre="Action & Adventure", runtime=108, mpaa\_rating="R", best\_dir\_win="no")  
predDead <- predict(finalMod, dataDead, interval="predict")  
  
# Show prediction results.  
df <- data.frame(t=c("Dirty Grandpa", "Deadpool"),  
 p=c(sprintf("%2.1f", predDG[1]),   
 sprintf("%2.1f", predDead[1])),  
 i=c(sprintf("%2.1f - %2.1f", predDG[2], predDG[3]),   
 sprintf("%2.1f - %2.1f", predDead[2], predDead[3])),  
 r=c("6.0", "8.1"))  
kable(df, col.names=c("Movie Title", "Predicted Rating", "95% Prediction Interval", "Actual Rating"))

|  |  |  |  |
| --- | --- | --- | --- |
| Movie Title | Predicted Rating | 95% Prediction Interval | Actual Rating |
| Dirty Grandpa | 5.9 | 4.1 - 7.8 | 6.0 |
| Deadpool | 6.1 | 4.2 - 7.9 | 8.1 |