group project 2

Group 05

2022/3/15

Contents

```
film <- read.csv("dataset5.csv")
```

Introduction of dataset

question of dataset

Imagine you have been asked by a film producer to investigate the following question of interest:

- Which properties of films influence whether they are rated by IMDB as greater than 7 or not?

You should conduct an analysis to answer your question using a Generalised Linear Model (GLM). Following your analyses, you should then summarise your results in the form of a presentation.

explain each variables

- film.id – The unique identifier for the film
- year – Year of release of the film in cinemas
- length – Duration (in minutes)
- budget – Budget for the films production (in $1000000s)
- votes – Number of positive votes received by viewers
- genre – Genre of the film
- rating – IMDB rating from 0-10

data processing

create a col seperate the rate: >7(1), <=7(0)

```
film <- film %>%
  mutate(rating.large7 = cut(rating, breaks = c(0,7,Inf), labels=c(0,1))) %>%
  dplyr::select(-film_id, -rating)%>%
  na.omit()
```

Exploratory data analysis

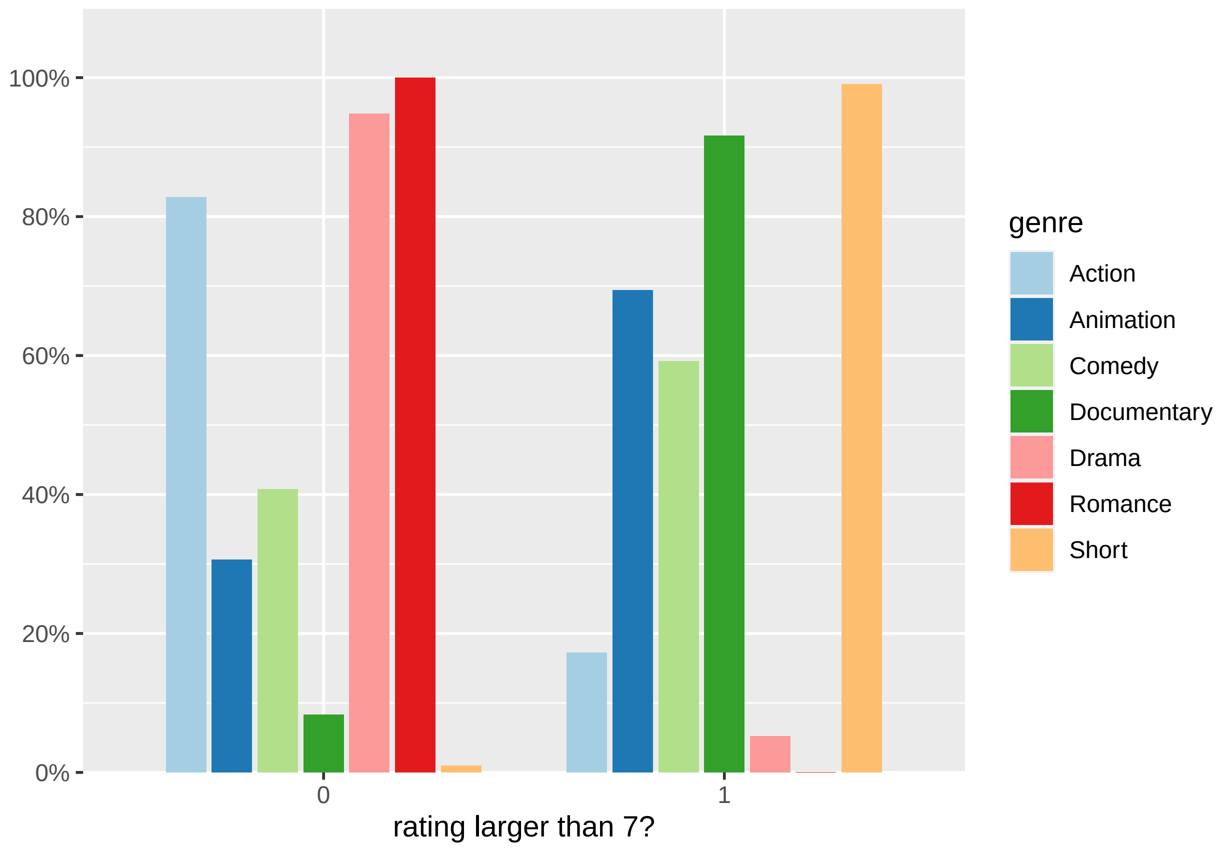
the distribution of rating.large7 by genre

```
film %>%
  group_by(genre, rating.large7)%>%
  summarise(n = n())
```

```
## ‘summarise()‘ has grouped output by ’genre’. You can override using the ‘.groups‘ argument.
```

```
## # A tibble: 13 x 3
## # Groups:   genre [7]
##    genre       rating.large7     n
##    <chr>       <fct>         <int>
##  1 Action      0               563
##  2 Action      1               117
##  3 Animation   0                49
##  4 Animation   1               111
##  5 Comedy      0               224
##  6 Comedy      1               325
##  7 Documentary 0                11
##  8 Documentary 1               121
##  9 Drama       0               620
## 10 Drama       1                34
## 11 Romance     0                15
## 12 Short       0                 1
## 13 Short       1               104
```

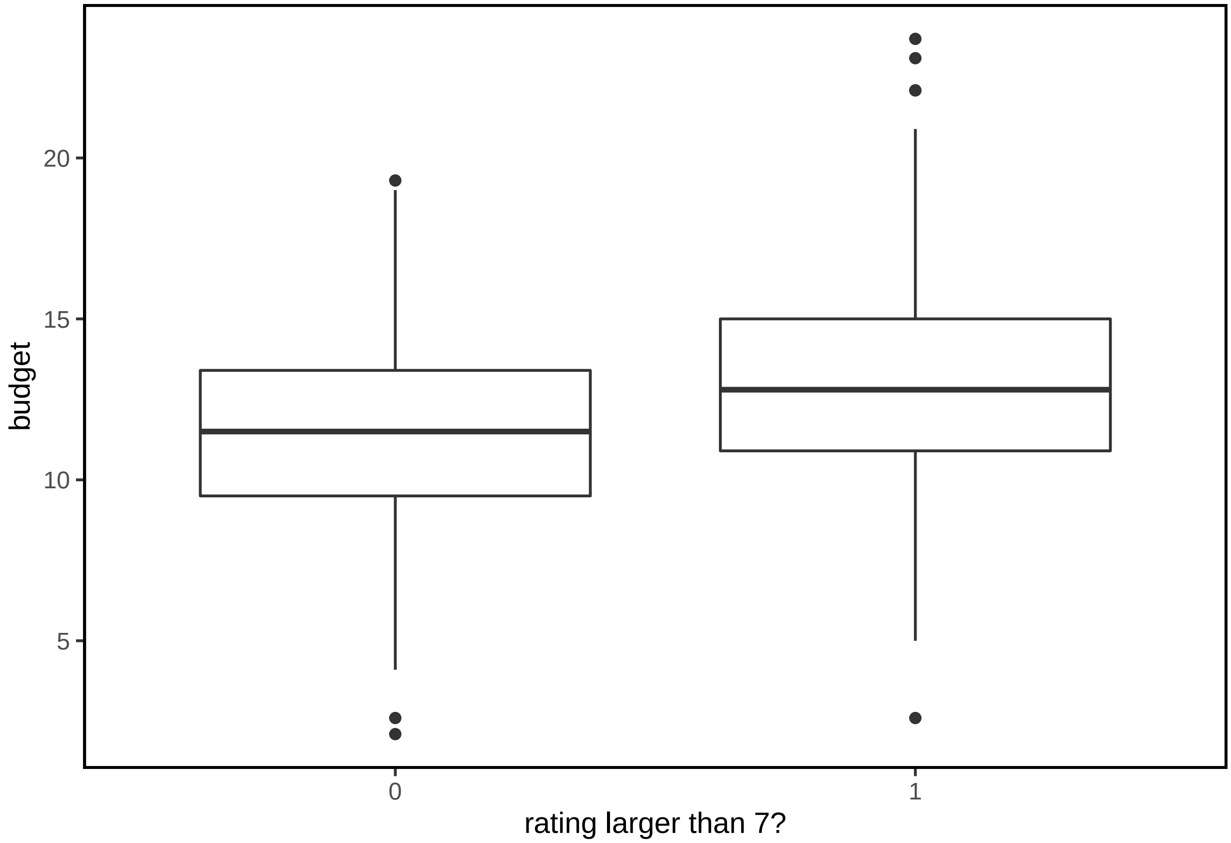
```
plot_xtab(film$rating.large7,film$genre,show.values =FALSE,show.total =FALSE,
axis.labels =c("0","1"),
axis.titles=c("rating larger than 7?"))
```



the distribution of rating.large7 by other numerical variables

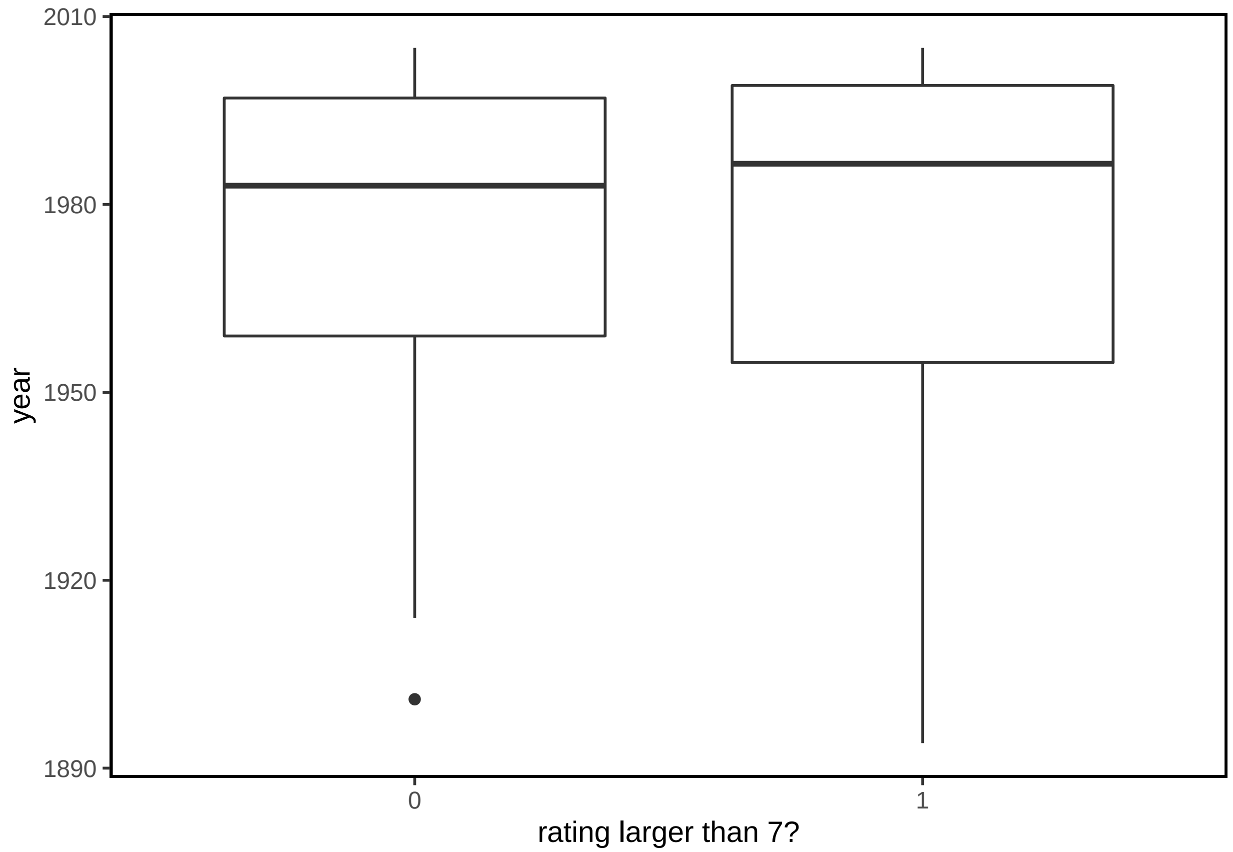
```
## budget
film.plot1<-ggplot(film, aes(y=budget,x=rating.large7))

film.plot1+geom_boxplot()+xlab("rating larger than 7?")+
theme(panel.background =element_rect(fill ="transparent",colour =NA),
plot.background =element_rect(fill ="transparent",colour =NA),
panel.border =element_rect(fill =NA,colour ="black",size =1))
```



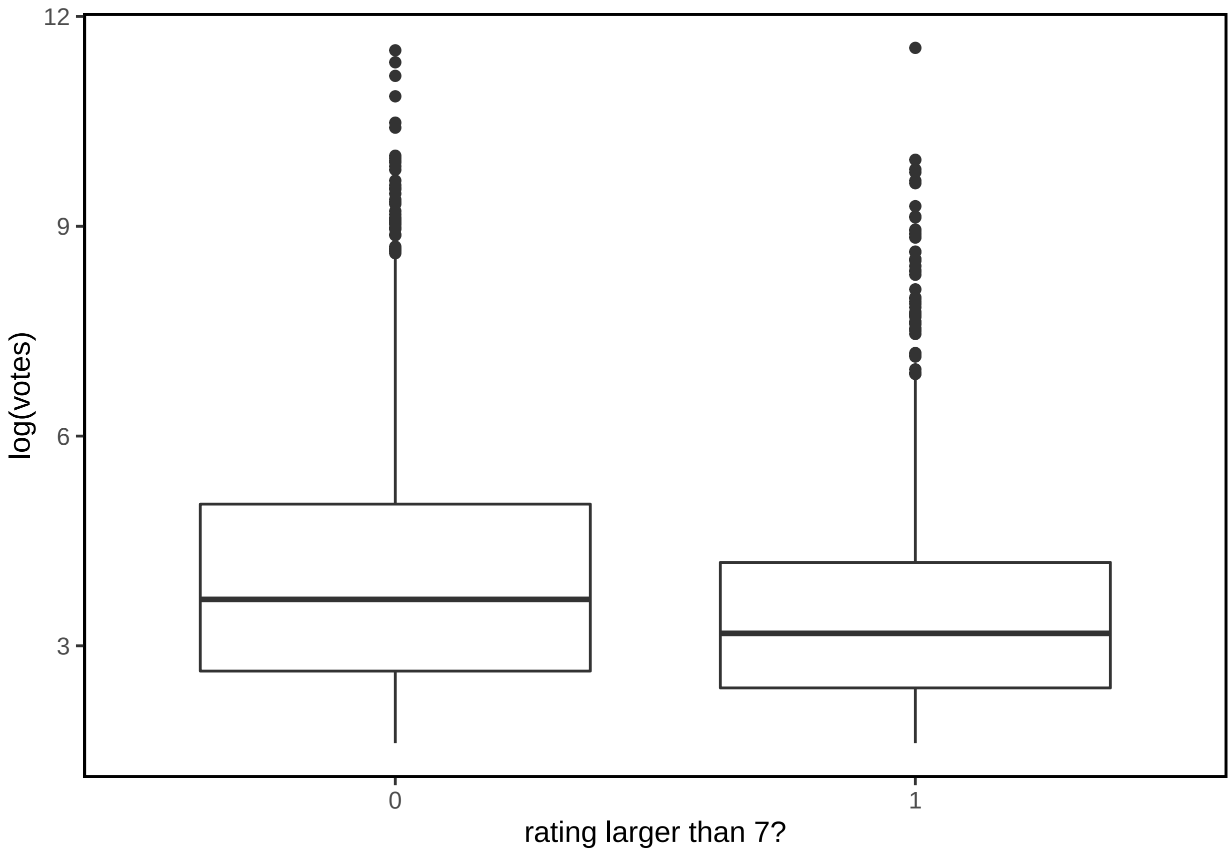
```
## year
film.plot2<-ggplot(film, aes(y=year,x=rating.large7))

film.plot2+geom_boxplot()+xlab("rating larger than 7?")+
theme(panel.background =element_rect(fill ="transparent",colour =NA),
plot.background =element_rect(fill ="transparent",colour =NA),
panel.border =element_rect(fill =NA,colour ="black",size =1))
```



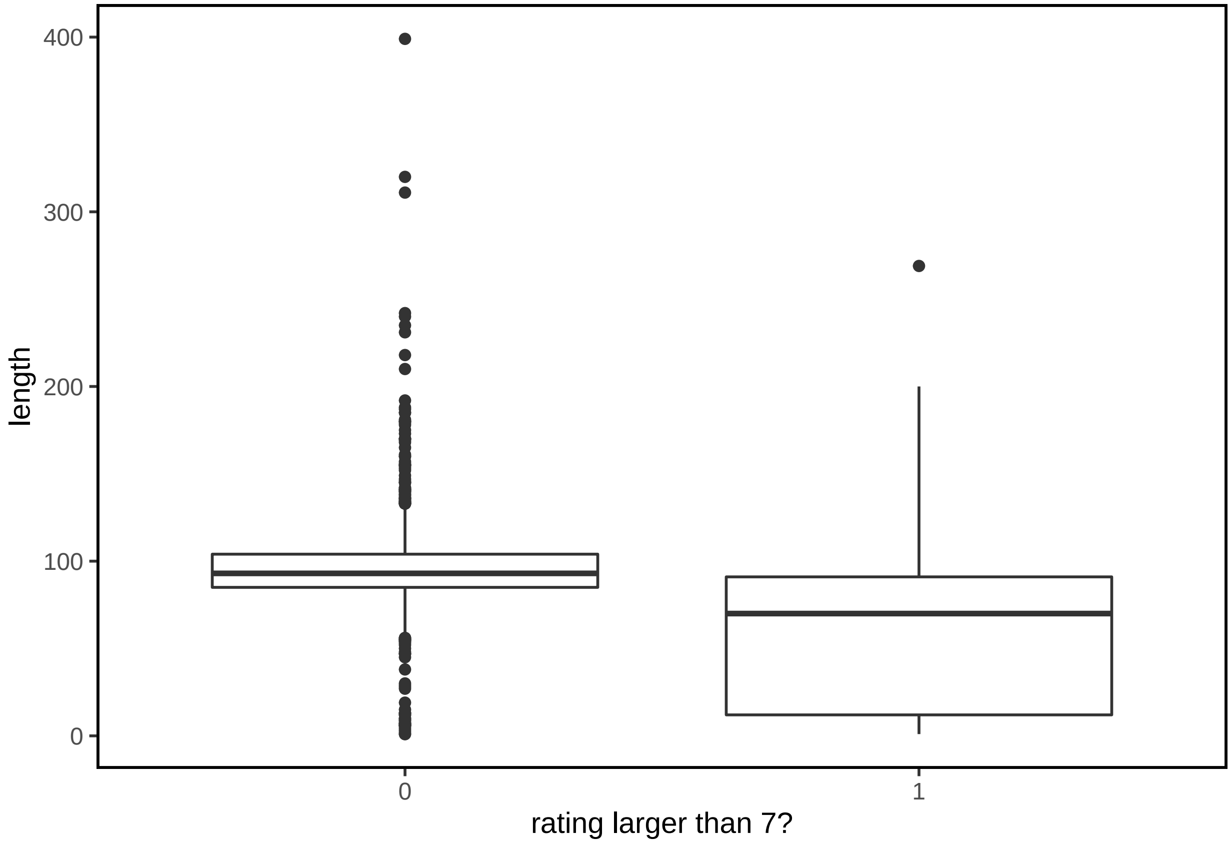
```
## votes
film.plot3<-ggplot(film, aes(y=log(votes),x=rating.large7))

film.plot3+geom_boxplot()+xlab("rating larger than 7?")+
theme(panel.background =element_rect(fill ="transparent",colour =NA),
plot.background =element_rect(fill ="transparent",colour =NA),
panel.border =element_rect(fill =NA,colour ="black",size =1))
```



```
## length
film.plot4<-ggplot(film, aes(y=length,x=rating.large7))

film.plot4+geom_boxplot()+xlab("rating larger than 7?")+
theme(panel.background =element_rect(fill ="transparent",colour =NA),
plot.background =element_rect(fill ="transparent",colour =NA),
panel.border =element_rect(fill =NA,colour ="black",size =1))
```



formal analysis

change the votes variable to log votes

```
film <- film %>%
  mutate(log.votes = log(votes))
```

stepwise: choosing which variables need to be removed.

Since Model2(removed year) has lowest AIC=1310.1 and deviance D=1290.1 we will go ahead and compared the three link function in our model

```
model <-  glm(rating.large7 ~ year + length + budget + log.votes + genre, family = binomial(link = "log

logit.step <- step(model,direction='both')
```

```
## Start:  AIC=1304.73
## rating.large7 ~ year + length + budget + log.votes + genre
##
##             Df Deviance    AIC
## - year       1   1283.2 1303.2
## <none>           1282.7 1304.7
## - log.votes  1   1293.0 1313.0
## - length     1   1595.2 1615.2
## - budget     1   1658.2 1678.2
## - genre      6   2101.0 2111.0
##
## Step:  AIC=1303.21
## rating.large7 ~ length + budget + log.votes + genre
##
##             Df Deviance    AIC
## <none>           1283.2 1303.2
## + year       1   1282.7 1304.7
## - log.votes  1   1294.3 1312.3
## - length     1   1602.0 1620.0
## - budget     1   1659.2 1677.2
## - genre      6   2118.6 2126.6
```

```
summary(logit.step)
```

```
##
## Call:
## glm(formula = rating.large7 ~ length + budget + log.votes + genre,
##     family = binomial(link = "logit"), data = film)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7981  -0.3973  -0.1132   0.2672   4.3137
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -3.902725   0.454750  -8.582  < 2e-16 ***
## length            -0.054295   0.003818 -14.220  < 2e-16 ***
## budget             0.499131   0.031641  15.775  < 2e-16 ***
## log.votes          0.139433   0.041932   3.325 0.000884 ***
## genreAnimation    -0.335711   0.346575  -0.969 0.332719
## genreComedy        2.677837   0.184023  14.552  < 2e-16 ***
## genreDocumentary   4.908172   0.414732  11.835  < 2e-16 ***
## genreDrama        -2.081558   0.259941  -8.008 1.17e-15 ***
## genreRomance     -14.705910 513.365991  -0.029 0.977147
## genreShort         4.192245   1.051548   3.987 6.70e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2982.5  on 2294  degrees of freedom
## Residual deviance: 1283.2  on 2285  degrees of freedom
## AIC: 1303.2
##
## Number of Fisher Scoring iterations: 15
```

compare different link functions

the AIC and BIC in model1 is the smallest, and the Pseudo-R^2 is the largest. Hence we choose 'logit' link function to fit our model

Model 1: logit link

```
model1 <- glm(rating.large7 ~ length + budget + log.votes + genre, family = binomial(link = "logit"), da
summary(model1)
```

```
##
## Call:
## glm(formula = rating.large7 ~ length + budget + log.votes + genre,
##     family = binomial(link = "logit"), data = film)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7981  -0.3973  -0.1132   0.2672   4.3137
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -3.902725   0.454750  -8.582  < 2e-16 ***
## length             -0.054295   0.003818 -14.220  < 2e-16 ***
## budget              0.499131   0.031641  15.775  < 2e-16 ***
## log.votes           0.139433   0.041932   3.325 0.000884 ***
## genreAnimation     -0.335711   0.346575  -0.969 0.332719
## genreComedy         2.677837   0.184023  14.552  < 2e-16 ***
## genreDocumentary    4.908172   0.414732  11.835  < 2e-16 ***
## genreDrama         -2.081558   0.259941  -8.008 1.17e-15 ***
## genreRomance      -14.705910 513.365991  -0.029 0.977147
## genreShort          4.192245   1.051548   3.987 6.70e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2982.5  on 2294  degrees of freedom
## Residual deviance: 1283.2  on 2285  degrees of freedom
## AIC: 1303.2
##
## Number of Fisher Scoring iterations: 15
```

```
summ(model1)
```

Model 2: probit link

```
model2 <- glm(rating.large7 ~ length + budget + log.votes + genre, family = binomial(link = "probit"), d
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model2)
```

```
##
## Call:
## glm(formula = rating.large7 ~ length + budget + log.votes + genre,
##     family = binomial(link = "probit"), data = film)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2418  -0.4281  -0.0836   0.2941   5.1505
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.25801    0.24005  -9.407  < 2e-16 ***
## length           -0.02704    0.00190 -14.227  < 2e-16 ***
## budget            0.26517    0.01612  16.452  < 2e-16 ***
## log.votes         0.07415    0.02294   3.233  0.00123 **
## genreAnimation    0.01144    0.18502   0.062  0.95071
## genreComedy       1.43664    0.09722  14.776  < 2e-16 ***
## genreDocumentary  2.67883    0.20897  12.819  < 2e-16 ***
## genreDrama       -1.12547    0.13240  -8.501  < 2e-16 ***
## genreRomance     -4.79462   76.38417  -0.063  0.94995
## genreShort        2.30958    0.44871   5.147 2.64e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2982.5  on 2294  degrees of freedom
## Residual deviance: 1308.2  on 2285  degrees of freedom
## AIC: 1328.2
##
## Number of Fisher Scoring iterations: 14
```

```
summ(model2)
```

Model 3: complementary log-log link

```
model3 <- glm(rating.large7 ~ length + budget + log.votes + genre, family = binomial(link = "cloglog"),
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model3)
```

```
##
## Call:
## glm(formula = rating.large7 ~ length + budget + log.votes + genre,
##     family = binomial(link = "cloglog"), data = film)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.7671  -0.4993  -0.2349   0.2710   3.1957
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.060e+00  2.770e-01 -11.048  < 2e-16 ***
## length           -2.613e-02  2.018e-03 -12.948  < 2e-16 ***
## budget            2.727e-01  1.770e-02  15.407  < 2e-16 ***
## log.votes         6.313e-02  2.702e-02   2.336   0.0195 *
## genreAnimation    2.914e-01  2.000e-01   1.457   0.1452
## genreComedy       1.618e+00  1.181e-01  13.702  < 2e-16 ***
## genreDocumentary  2.821e+00  1.889e-01  14.929  < 2e-16 ***
## genreDrama       -1.423e+00  1.921e-01  -7.411 1.25e-13 ***
## genreRomance     -2.446e+01  7.992e+04   0.000   0.9998
## genreShort        2.352e+00  3.408e-01   6.902 5.11e-12 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2982.5  on 2294  degrees of freedom
## Residual deviance: 1382.1  on 2285  degrees of freedom
## AIC: 1402.1
##
## Number of Fisher Scoring iterations: 25
```

```
summ(model3)
```

Model	link function	AIC	BIC
model1	$g(p_i) = log(\frac{p_i}{1-p_i})$	1303.21	1360.60
model2	$g(p_i) = \phi^{-1}(pi), p_i = \phi(\frac{x_i-\mu}{\sigma})$	1328.18	1385.57
model3	$g(p_i) = log[-log(1-p_i)]$	1402.13	1459.51

D0-D1 = 2982.5-2357.5 = 1692.401 > x^2(0.95, 9) = 16.91898, so we reject H0, we can say that the model1 fits the data better than saturated model.

$$D_0 - D_1 = 2982.5 - 2357.5 = 1692.401 > \chi^2(0.95, 9) = 16.91898$$

So we reject H0, and we can say that the model1 fits the data better than saturated model.

```
model1$null.deviance - model1$deviance
```

```
## [1] 1699.256
```

```
df = model1$df.null - model1$df.residual
qchisq(p=0.95, df = df)
```

```
## [1] 16.91898
```

odds ratios of model1

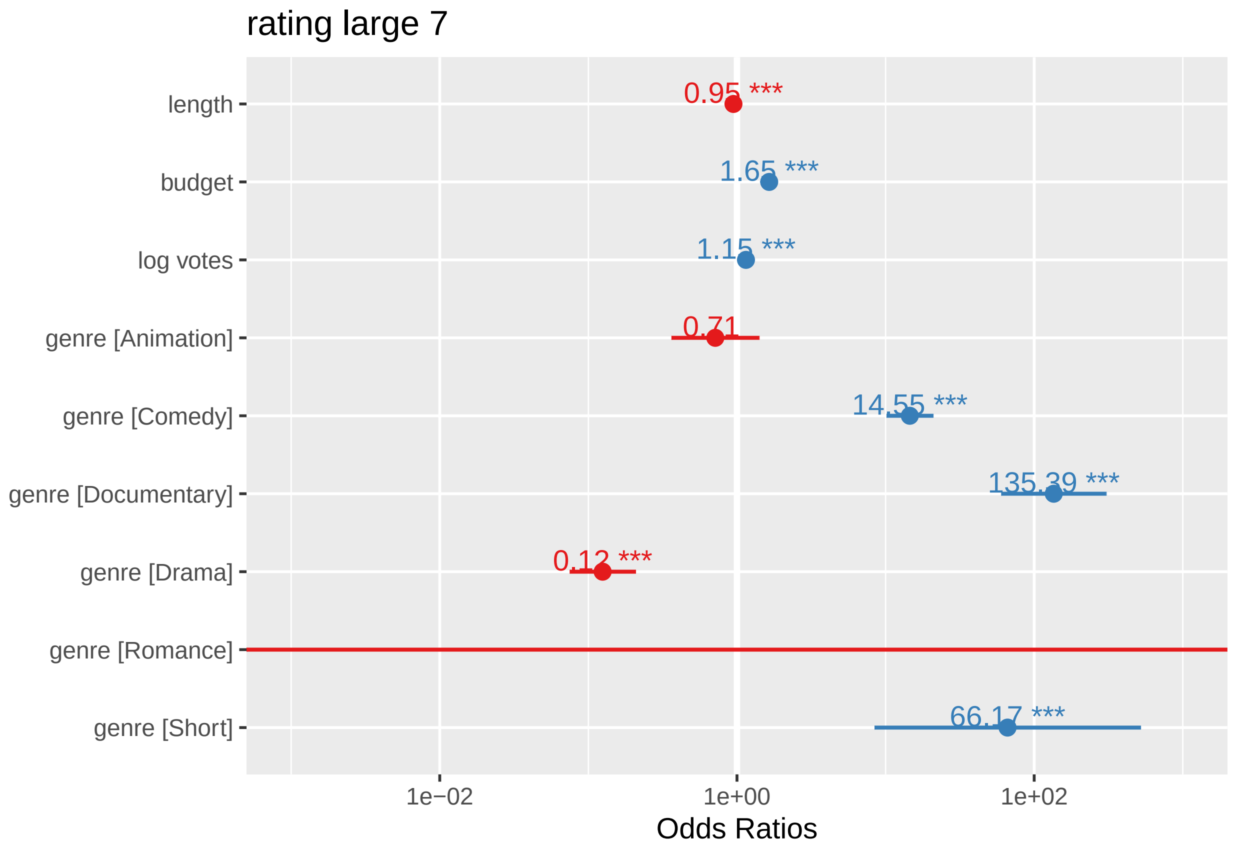
```
plot_model(model1,show.values=TRUE)+
  scale_y_log10(limits = c(0.001, 1000))
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

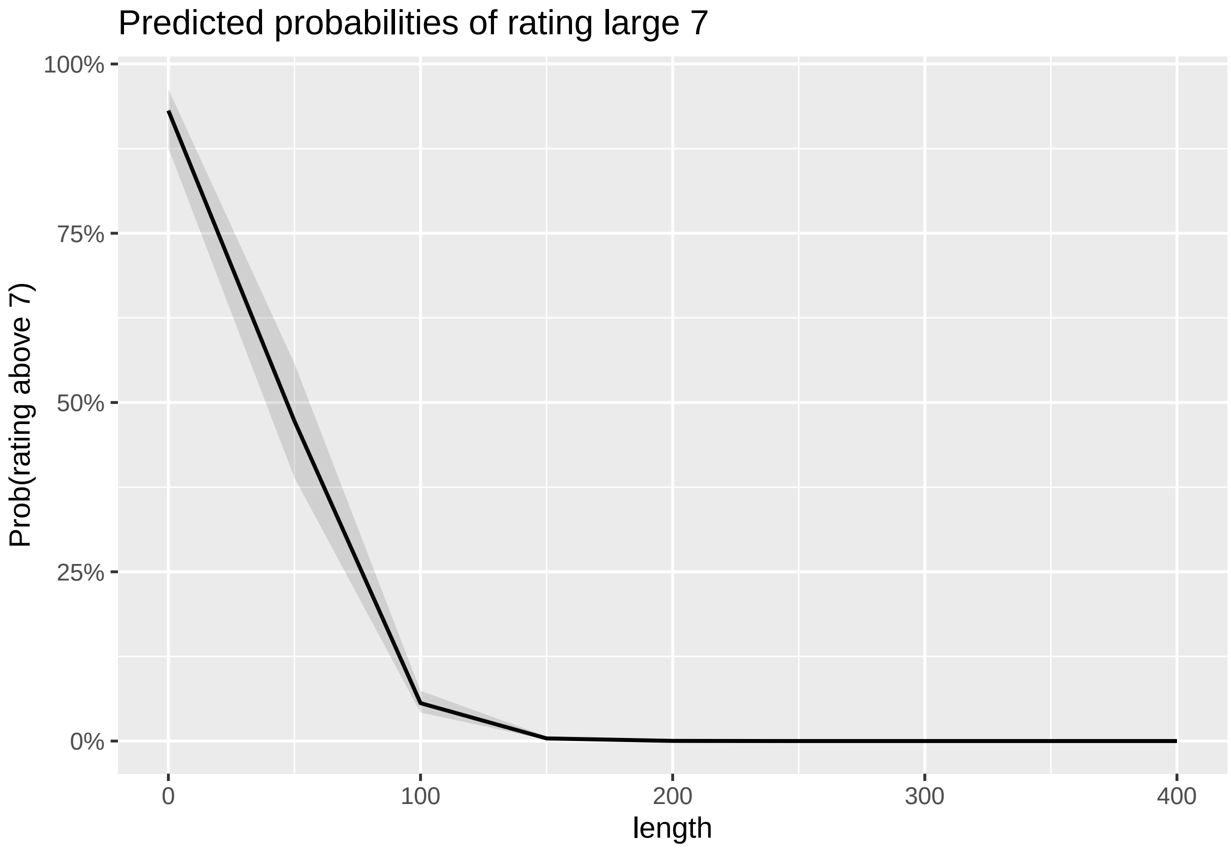
```
## Warning: Removed 1 rows containing missing values (geom_text).
```



prediction

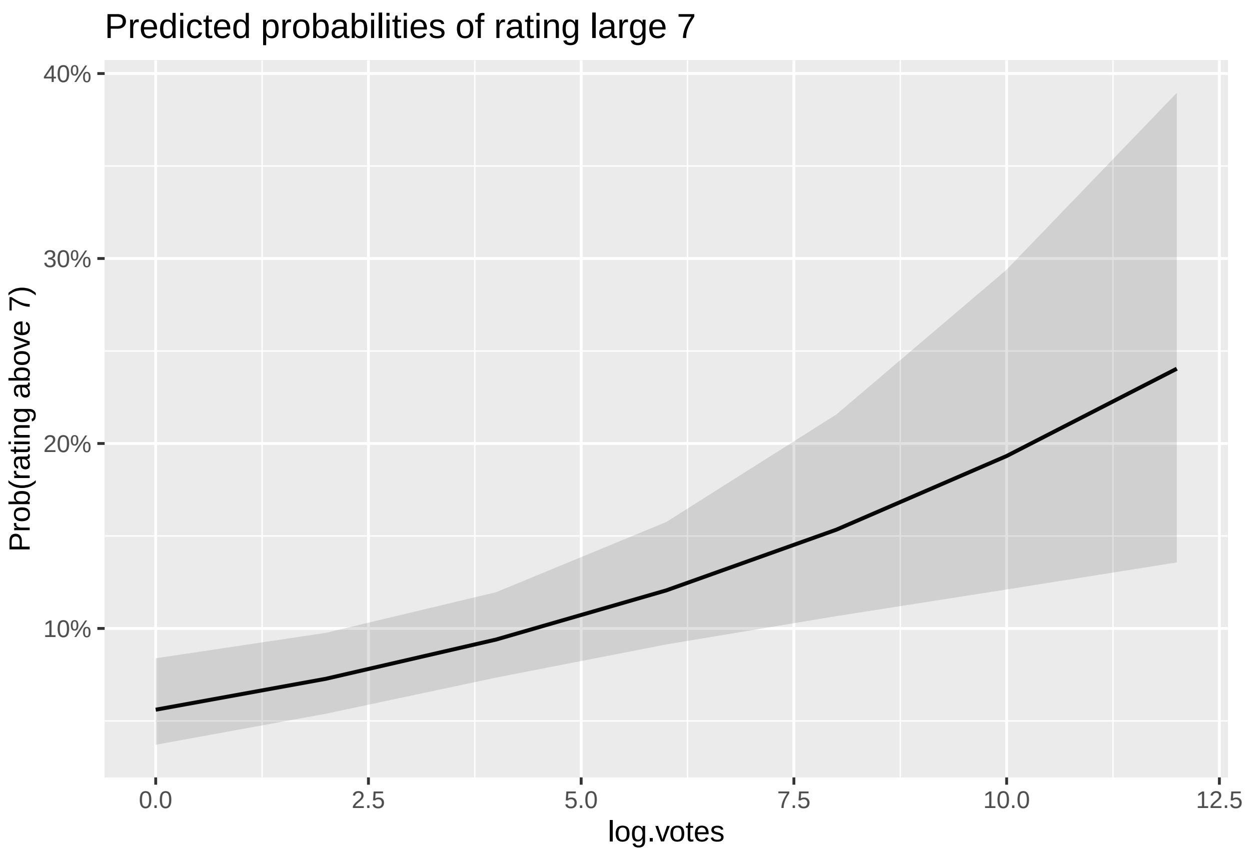
```
plot_model(model1,type="pred",terms=c("length"),axis.title=c("length","Prob(rating above 7)"))
```

```
## Data were 'prettified'. Consider using `terms="length [all]"` to get smooth plots.
```



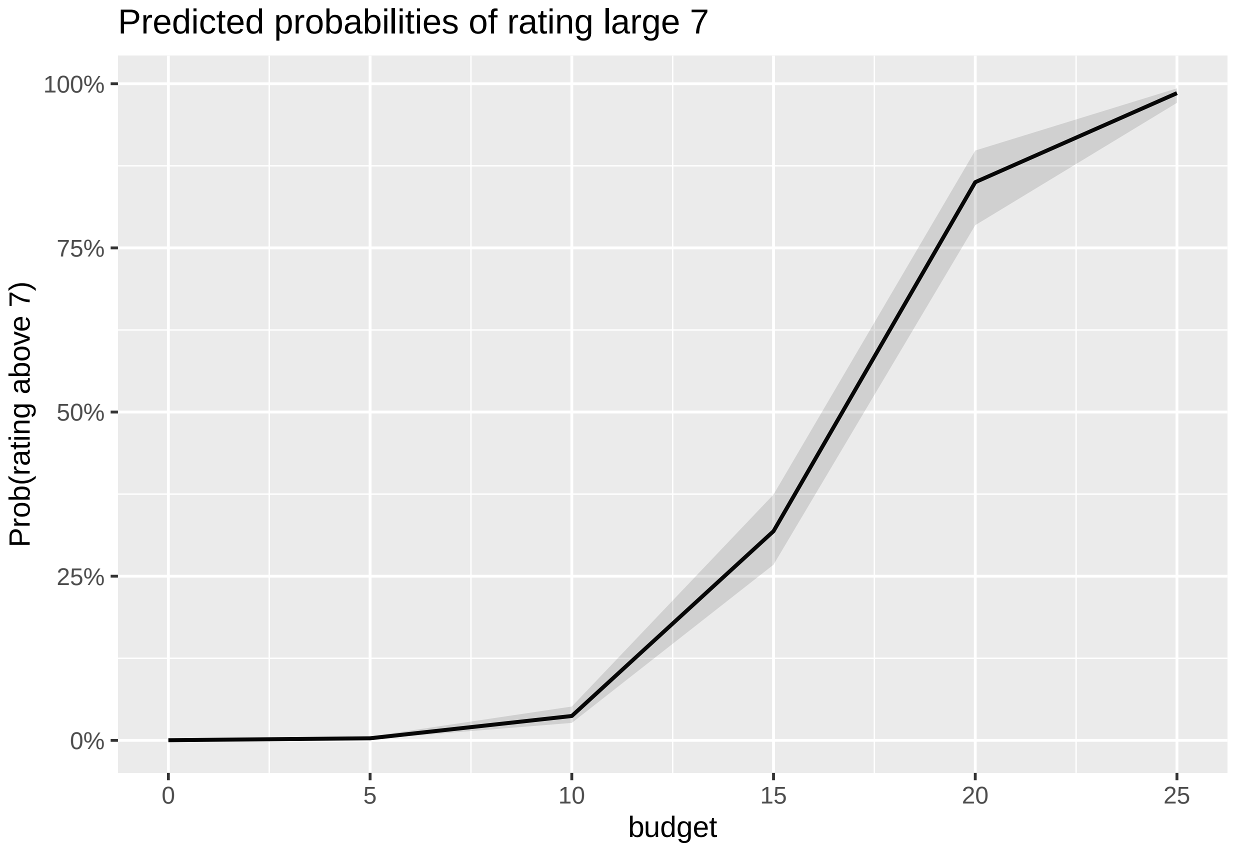
```
plot_model(model1,type="pred",terms=c("log.votes"),axis.title=c("log.votes","Prob(rating above 7)"))
```

```
## Data were 'prettified'. Consider using `terms="log.votes [all]"` to get smooth plots.
```

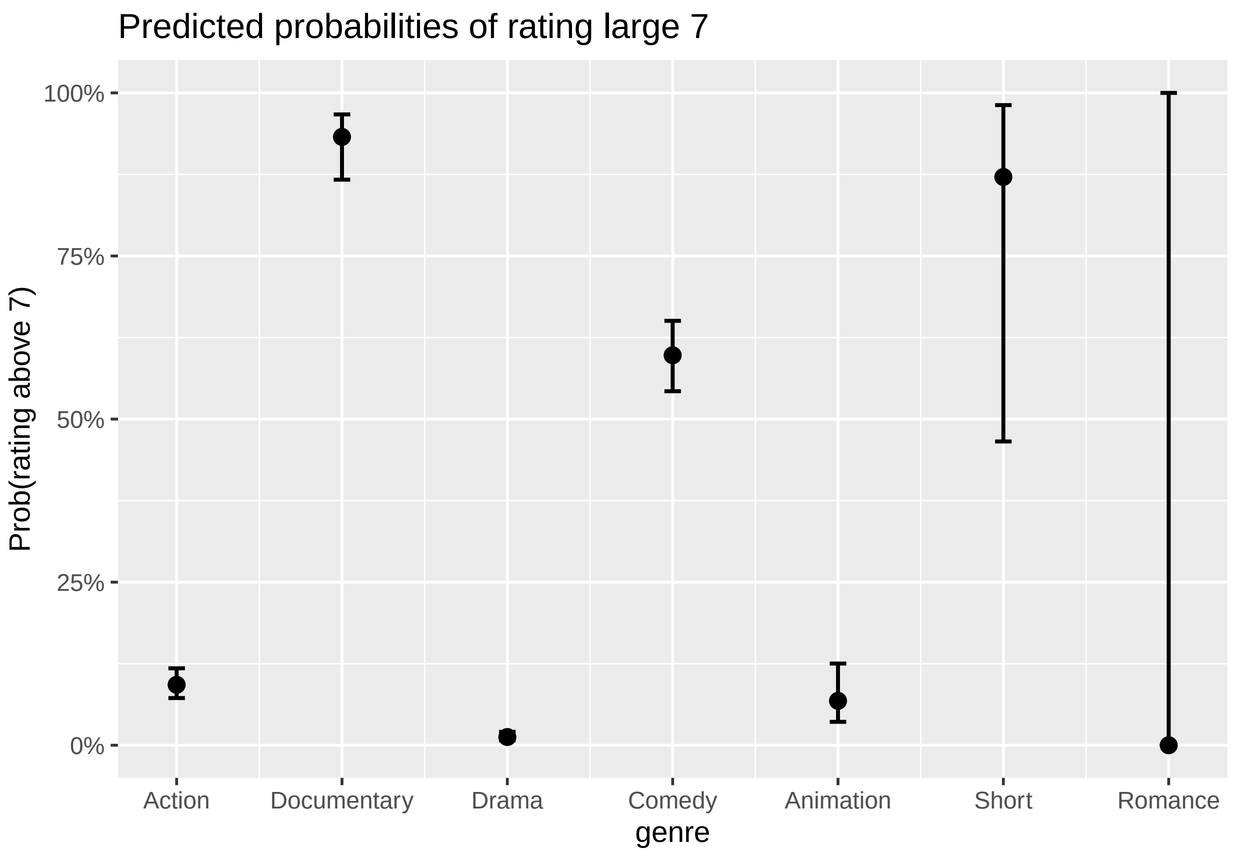


```
plot_model(model1,type="pred",terms=c("budget"),axis.title=c("budget","Prob(rating above 7)"))
```

```
## Data were 'prettified'. Consider using `terms="budget [all]"` to get smooth plots.
```



```
plot_model(model1,type="pred",terms=c("genre"),axis.title=c("genre","Prob(rating above 7)"))
```

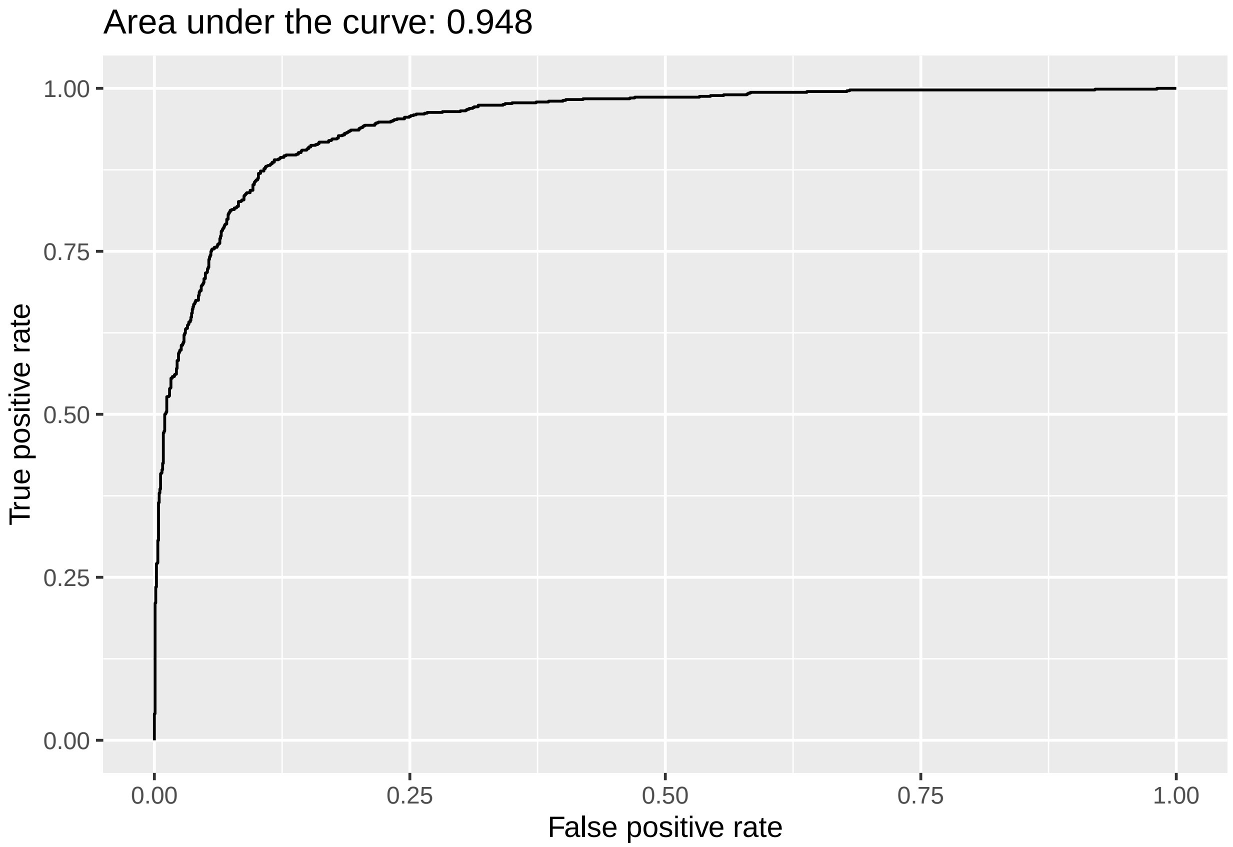


Predicted probabilities of rating large 7

Model checking and diagnostics

ROC and AUC

```
film$Prid <- predict(model1, film, type="response")
score <- prediction(film$Prid,film$rating.large7)
perf <- performance(score,"tpr","fpr")
auc <- performance(score,"auc")
perfd <- data.frame(x= perf@x.values[1][[1]], y=perf@y.values[1][[1]])
p4<- ggplot(perfd, aes(x= x, y=y)) + geom_line() +
xlab("False positive rate") + ylab("True positive rate") +
ggtitle(paste("Area under the curve:", round(auc@y.values[[1]], 3)))
p4
```



AUC = 0.948 indicated that model 1 is very good at predicting the films rating greater than 7 given all predictor variables.