MAP501 Coursework 2023

Preamble

```
library("tidyverse")
library("magrittr")
library("here")
library("janitor")
library("lubridate")
library("gridExtra")
library("readxl")
library("glmnet")
library("Lahman")
library("AER")
library("viridis")
library("lindia")
library("lme4")
library("caret")
library("pROC")
```

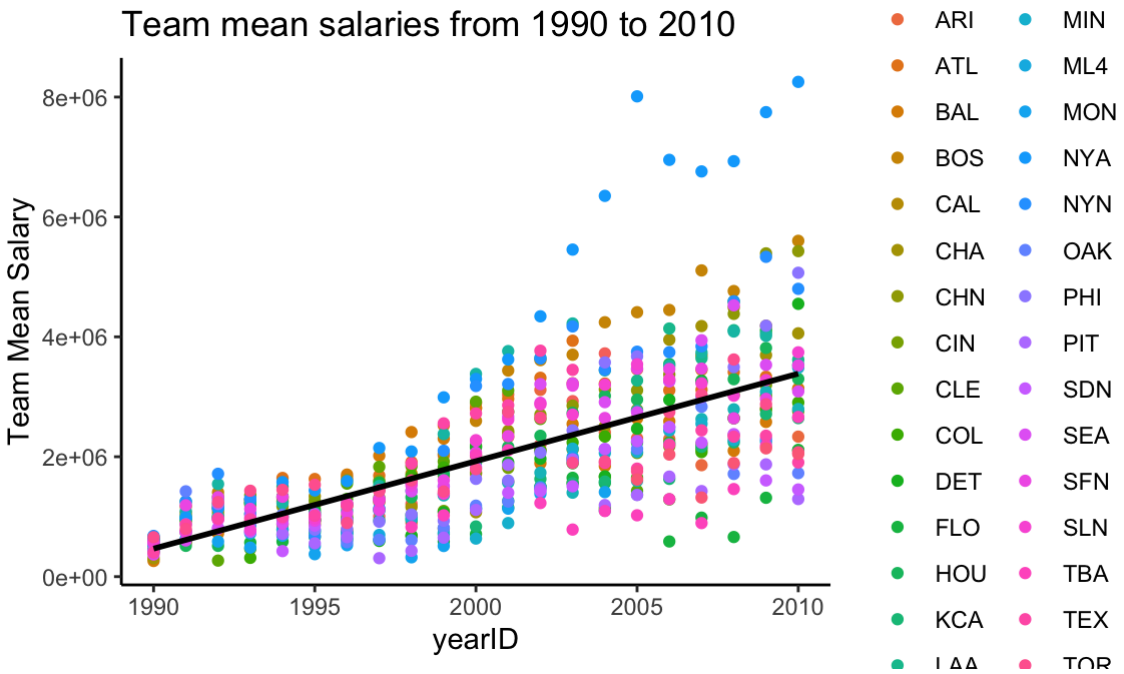
1. Simple Linear Regression

a. Create ‘df_MeanSalaries’ by taking the data from the years 1990 to 2010 from ‘Salaries’. Add the variable ‘meanSalary’ = the mean salary for each team per year. Ensure that there is a single row for each team per year. Use ‘df_MeanSalaries’ for the rest of question 1. [2 points]

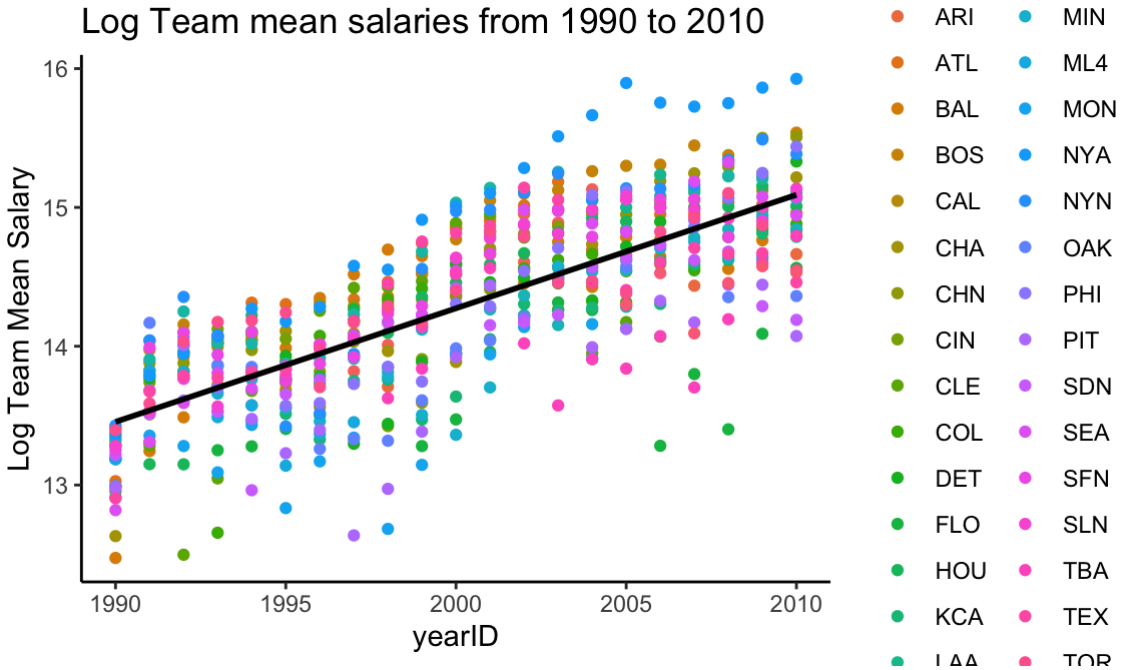
```
data("Salaries", package = "Lahman")

df_MeanSalaries <- Salaries %>%
  filter(yearID >= 1990 & yearID <= 2010) %>%
  group_by(yearID, teamID) %>%
  summarise(meanSalary = mean(salary))
```

b. Create one plot of team mean salaries over time from 1990 to 2010 and another of the log base 10 of team mean salaries over the same period. Comment and compare the two plots. [4 points]

```
df_MeanSalaries %>%
  ggplot(mapping = aes(x = yearID, y = meanSalary, colour = factor(teamID) )) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, colour = "black") +
  labs(x = "yearID", y = "Team Mean Salary") +
  ggtitle("Team mean salaries from 1990 to 2010") +
  theme_classic()
```




```
log_MeanSalaries <- df_MeanSalaries %>%
  mutate(logmeanSalary = log(meanSalary))

  log_MeanSalaries %>%
  ggplot(mapping = aes(x = yearID, y = logmeanSalary,
                       colour = factor(teamID) )) +
  geom_point() +
    geom_smooth(method = "lm", se = FALSE, colour = "black") +
  labs(x = "yearID", y = "Log Team Mean Salary") +
  ggtitle("Log Team mean salaries from 1990 to 2010") +
  theme_classic()
```



There isn't much of a linear relationship between 'year' and 'team mean salary.' On the second plot with the log transformation of the 'team mean salary' variable however, the scales of the newly derived variable align to form a much more obvious positive linear relationship. As yearID increases, log team mean salary also increases.

c. Fit a model of log_{10} of team mean salaries as a function of year. Report and interpret the results. Write the form of the fitted model (coefficients should be rounded to 2 significant figures). [10 points]

```
salarymod <- lm(logmeanSalary ~ yearID, data = log_MeanSalaries)
salarymod
summary(salarymod)
```

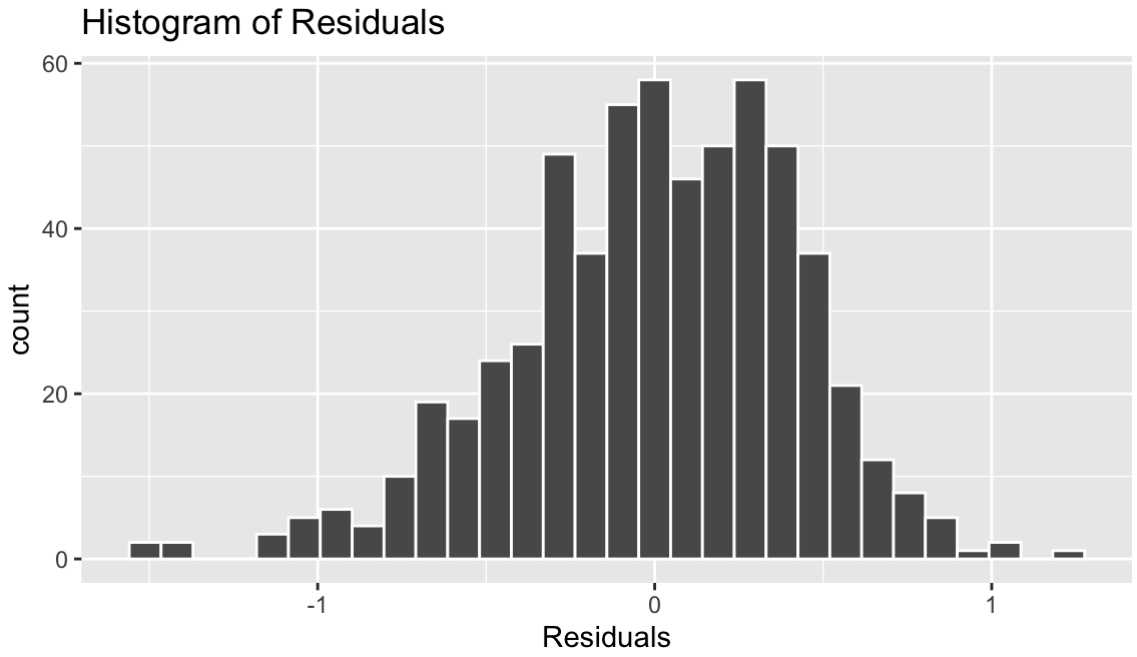
```
Call:
lm(formula = logmeanSalary ~ yearID, data = log_MeanSalaries)

Coefficients:
(Intercept)       yearID
 -149.28964      0.08178


Call:
lm(formula = logmeanSalary ~ yearID, data = log_MeanSalaries)

Residuals:
     Min       1Q   Median       3Q      Max
-1.52599 -0.26057  0.02626  0.31182  1.21434

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.493e+02  5.670e+00  -26.33   <2e-16 ***
yearID       8.178e-02  2.835e-03   28.85   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4186 on 606 degrees of freedom
Multiple R-squared:  0.5787,    Adjusted R-squared:  0.578
F-statistic: 832.3 on 1 and 606 DF,  p-value: < 2.2e-16
```

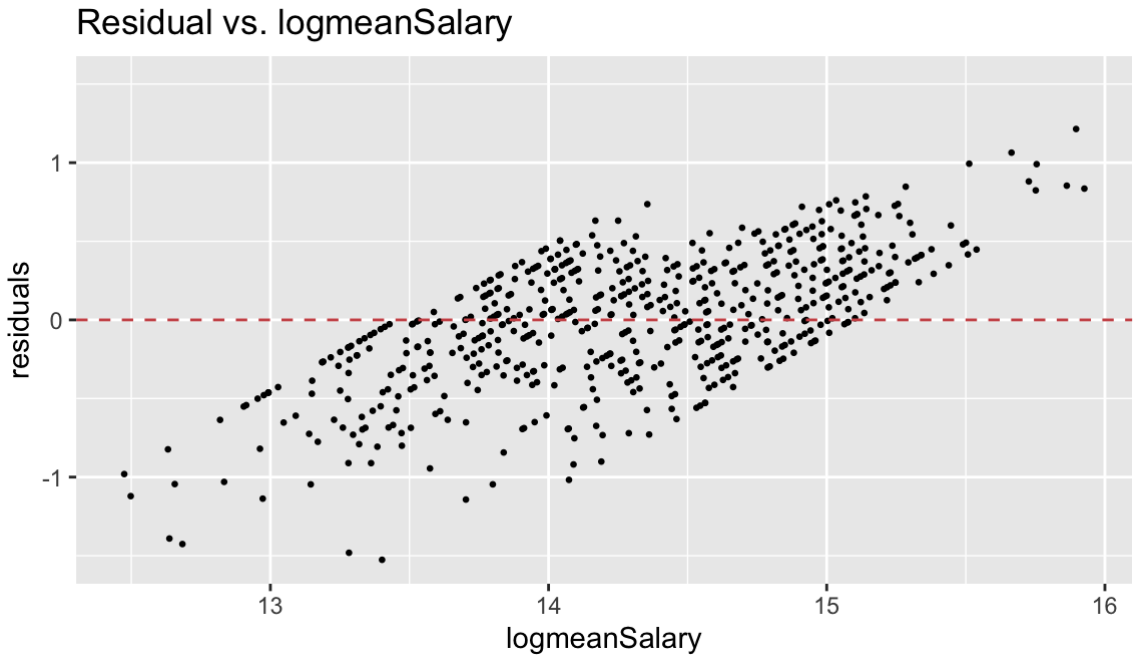
The form of the fitted model is: logmeanSalary ~ N(-150 + 0.082 x yearID, 0.42)

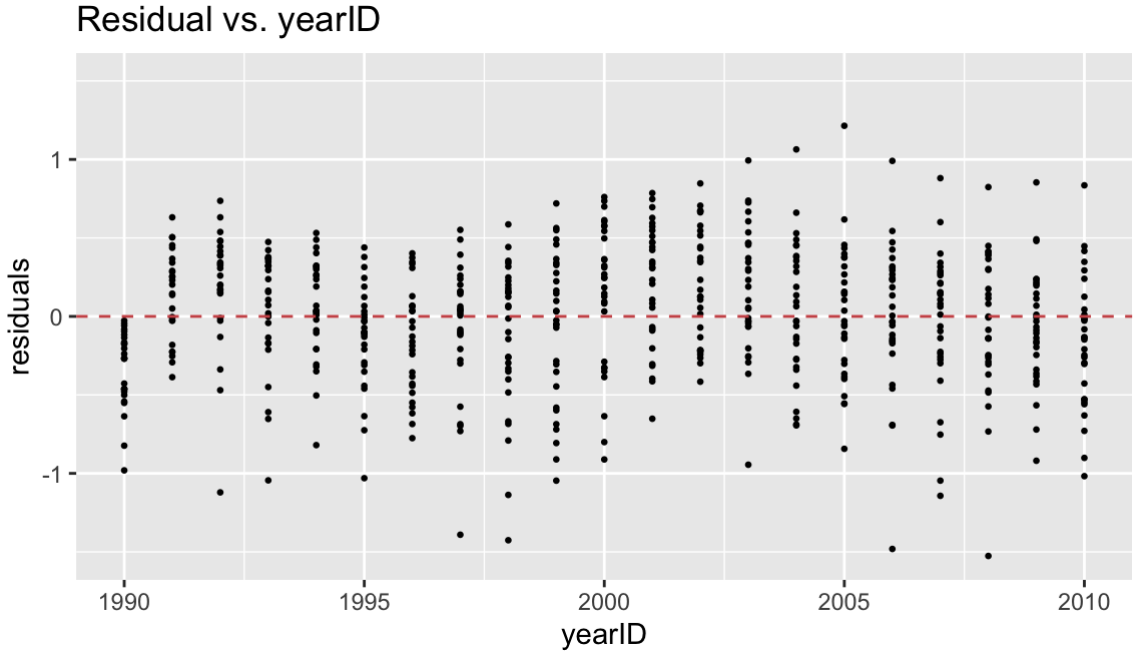
There is a very strong positive relationship between yearID and logmeanSalary. As the yearID increases, logmeanSalary increases. The p value at 2e-16 is statistically significant as it is very close to 0 (much smaller than the 0.05 (5%) threshold). This puts the null hypothesis in the rejection zone and we say that statistically, yearID is very unlikely not to be important in determining logmeanSalary. As the year increases by 1 year, logmeanSalary increases by 1 percent. The multiple R squared is 0.58. This implies that 58% of the variance in logmeanSalary is explained by differences in yearID. So yearID is definitely explanatory, but there are still other uncontrolled factors that are also influencing logmeanSalary.

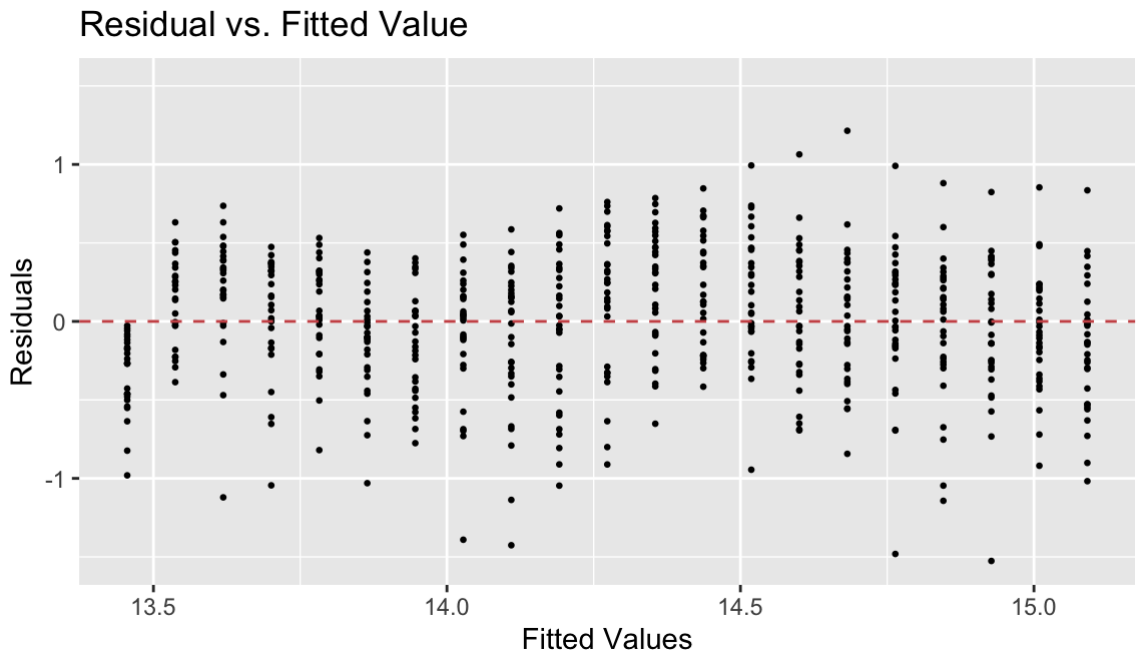
d. State and evaluate the assumptions of the fitted model. [9 points]

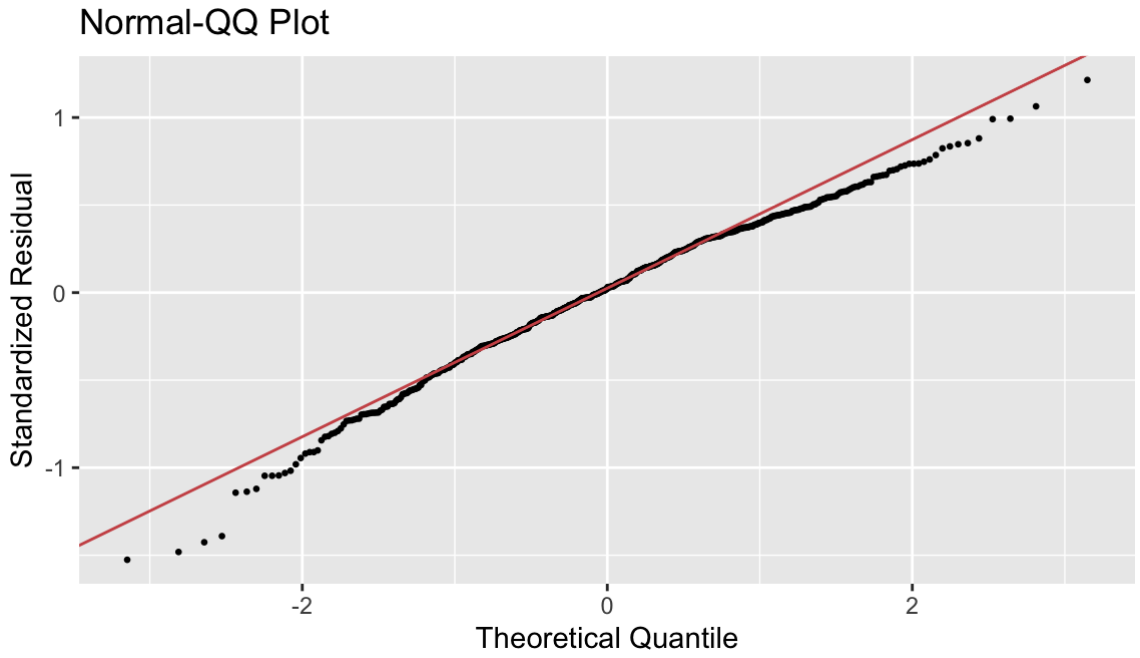
```
salarymod %>%
  gg_diagnose(max.per.page = 1)
```

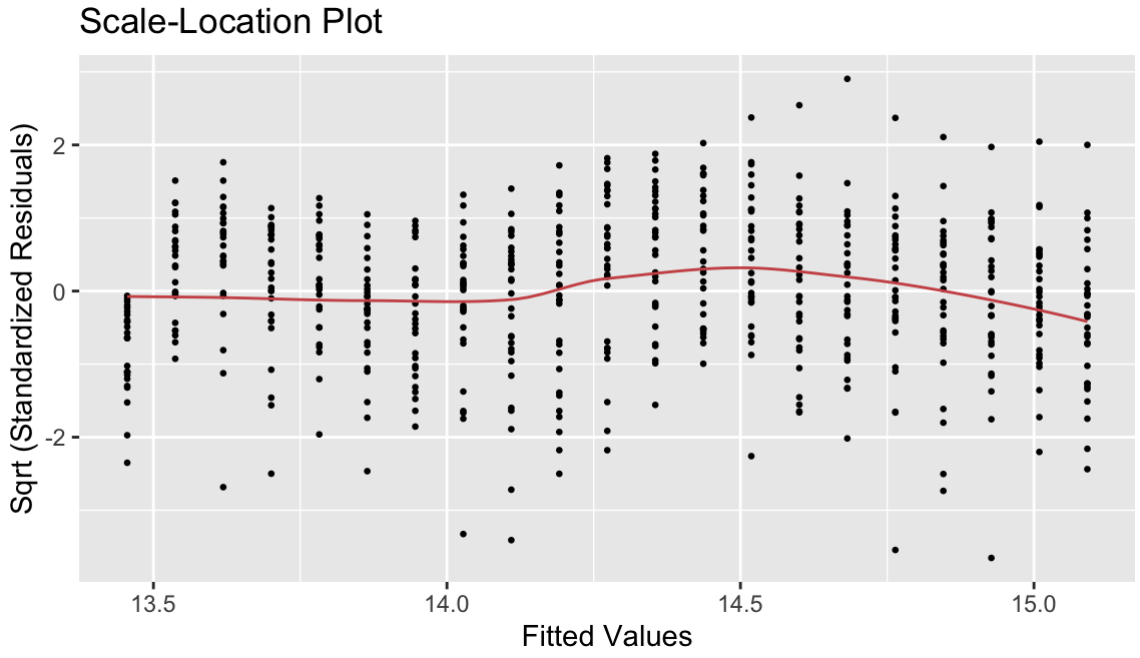

Histogram of Residuals
count
Residuals
60
40
20
0
-1
0
1

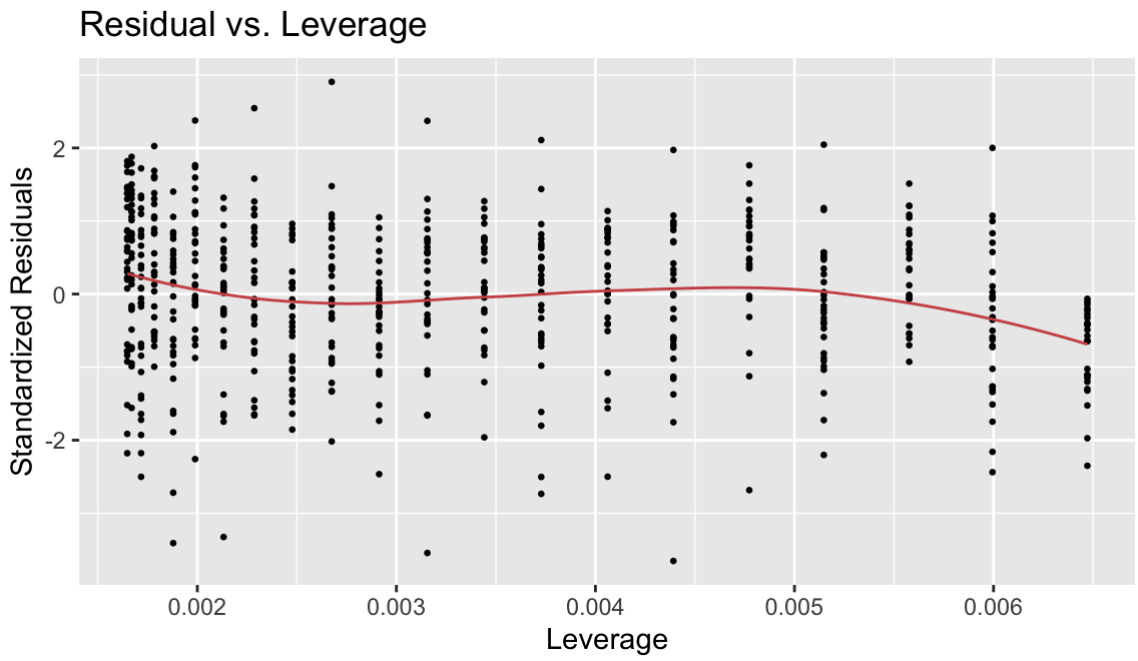

Residual vs. logmeanSalary
residuals
logmeanSalary
1
0
-1
13
14
15
16

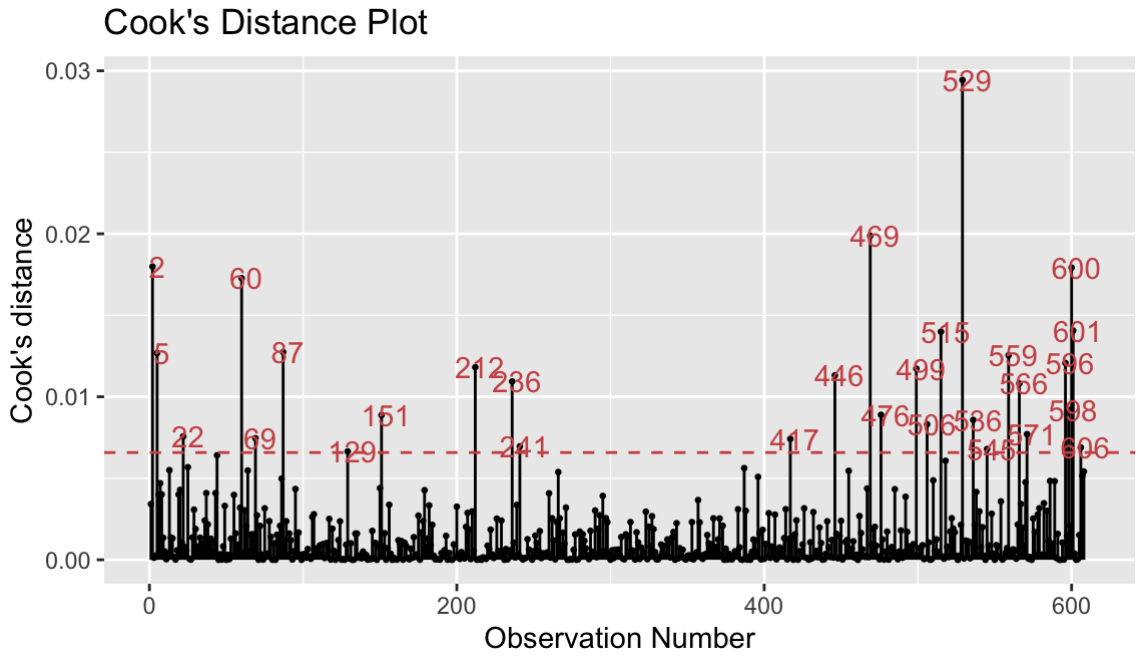

Residual vs. yearID
residuals
yearID
1
0
-1
1990
1995
2000
2005
2010


Residual vs. Fitted Value
Residuals
1
0
-1
13.5
14.0
14.5
15.0
Fitted Values


Normal-QQ Plot
Standardized Residual
1
0
-1
-2
0
2
Theoretical Quantile


Scale-Location Plot
Sqrt (Standardized Residuals)
2
0
-2
13.5
14.0
14.5
15.0
Fitted Values





1. Linearity: Looking at the scatterplot of yearID versus log team mean salary, there is a good positive linear relationship between the two variables. As the yearID increases, log team mean salary increases.

2. Homoscedasticity :The scatter of residuals versus logmeanSalary reveals an obvious trend. The points show a definite positive relationship between the pair. This is a display of heteroscedasticity. This may mean that the standard deviation in the model does depend on the values of the predictor (yearID) variable. However, looking at the residuals vs leverage plot, there seems to be no pattern, with all the points evenly spread out, and indicating that residuals are randomly distributed across all levels of leverage. This contradicts the scatter of residuals versus logmeanSalary.

The scatter of residuals versus yearID on the other hand is a lot more random. The points are a lot more evenly spread out though there is a very slight display of ‘snaking’.

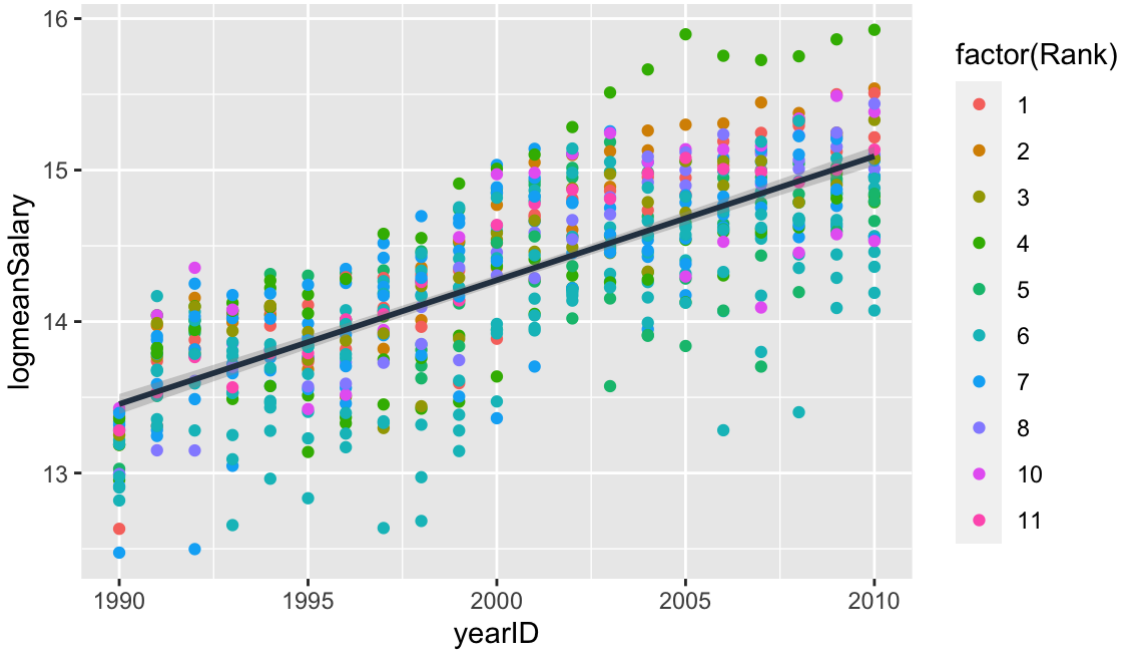
3. Normality : The histogram of residuals, though has a very subtle left skew, can be said to be normaly distributed.This points to the fact that the choice of the gaussian distribution is good here.

The qq plot of residuals looks roughly like a straight line, though there are slight deviations at both ends of the line.The straight line shape also points to the fact that the choice of a gaussian distribution is good.

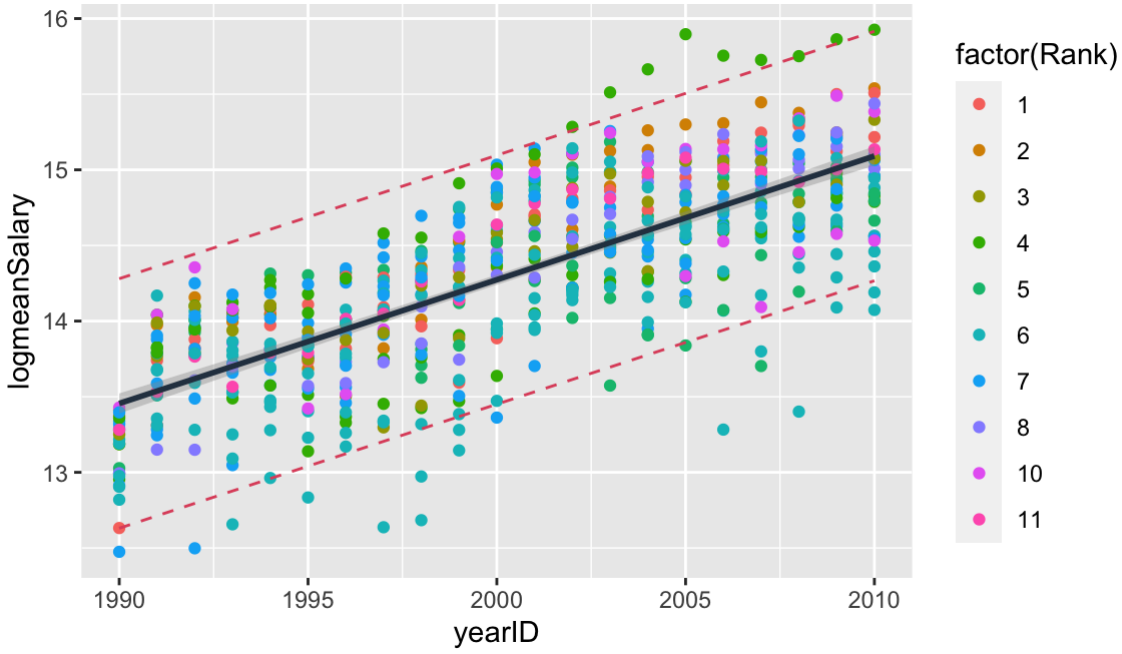
e. Plot confidence and prediction bands for this model. Colour the points according to a third variable from any of Lahman dataset (apart from ‘Salaries’). Comment on what you find. Find out what teams do

they relate to the points that appear outside the prediction band. [10 points]

```
teamsrank <- Teams[, c("teamID", 'Rank')]
teamsrank <- inner_join(log_MeanSalaries, teamsrank, by = "teamID")

teamsrank <- teamsrank %>%
distinct(teamID, .keep_all = TRUE)

ggplot(teamsrank, aes(x = yearID,y = logmeanSalary, color = factor(Rank))) +
  geom_point() +
  geom_smooth(method = lm, color = '#2C3E50')
```




```
predb <- predict(salarymod, interval = "prediction")

lmsalaries <- cbind(teamsrank, predb)
ggplot(lmsalaries, aes(x = yearID, y = logmeanSalary, color = factor(Rank))) +
  geom_point() +
  geom_smooth(method = lm, color = '#2C3E50') +
  geom_line(aes(y = lwr), color = 2,lty = 2) +
  geom_line(aes(y = upr), color = 2,lty = 2)
```




```
outside_band <- lmsalaries$logmeanSalary < lmsalaries$lwr |
  lmsalaries$logmeanSalary > lmsalaries$upr

points_outside_band <- lmsalaries[outside_band,]
print(points_outside_band)

teams_outside_band <- unique(points_outside_band$teamID)
print(teams_outside_band)
```

```
# A tibble: 27 × 8
# Groups:   yearID [17]
   yearID teamID meanSalary logmeanSalary  Rank   fit   lwr   upr
    <int> <fct>       <dbl>         <dbl> <int> <dbl> <dbl> <dbl>
 1   1990 BAL       261624.          12.5     7  13.5  12.6  14.3
 2   1992 CLE       267801.          12.5     7  13.6  12.8  14.4
 3   1993 COL       313742.          12.7     6  13.7  12.9  14.5
 4   1995 MON       374667.          12.8     6  13.9  13.0  14.7
 5   1997 PIT       307762.          12.6     6  14.0  13.2  14.9
 6   1998 MON       322470.          12.7     6  14.1  13.3  14.9
 7   1998 PIT       430429.          13.0     6  14.1  13.3  14.9
 8   1999 FLO       585694.          13.3     6  14.2  13.4  15.0
 9   1999 MON       511514.          13.1     6  14.2  13.4  15.0
10   2000 MIN       635365.          13.4     7  14.3  13.5  15.1
# ℹ 17 more rows
 [1] BAL CLE COL MON PIT FLO MIN NYA TBA SDN
149 Levels: ANA ARI ATL BAL BOS CAL CHA CHN CIN CLE COL DET FLO HOU KCA ... WSU
```

The confidence band is very narrow, implying that there is relatively low uncertainty or variability in the estimation of logmeanSalary. The points that fall outside the prediction band however suggest that these observations deviate from the expected pattern predicted by the model.These points may be influencial or outliers and are most likely responsible for the heteroscedasticity observed in the residuals versus fitted plot explored in part d.

The Rank variable introduced to the plot does not reveal any obvious trend. It appears that there isnt an obvious correlation between rank, yearID and logmeanSalary. The only observation worthy of note is that the outlier teams are consistently ranked between 4th and 7ths positions across the years observed.

10 teams appear outside the band. They are: BAL CLE COL MON PIT FLO MIN NYA TBA SDN

2. Multiple Regression for Count Data [35 points]

a. Create a dataset ‘df_FieldingData’ from ‘Fielding’ by

i. selecting data of the two years 1990 and 2015 (note that it is not all the years 1990 to 2015).
ii. selecting playerID, year and position.

Then create a dataset ‘df_BattingData’ from the dataset ‘Batting’ by:

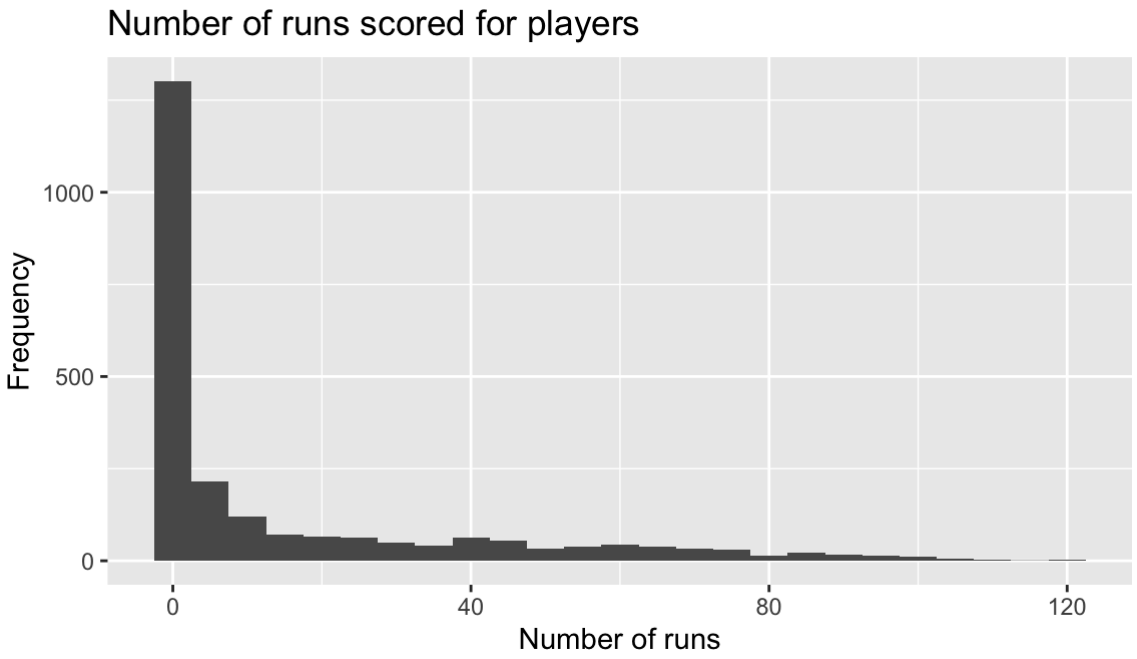
iii. selecting data of the two years 1990 and 2015,
iv. adding height, weight, birthYear of players from ‘People’.
v. adding position played from ‘df_FieldingData’
vi. creating a new variable ‘age’ equal to each player’s age in the relevant year,
vii. dropping incomplete cases from the dataset and dropping unused levels of any categorical variable.
viii. remove duplication in players (i.e., each player’s data is in a single row).

Use ‘df_BattingData’ for the rest of question 2. Note: use one code chunk for a.

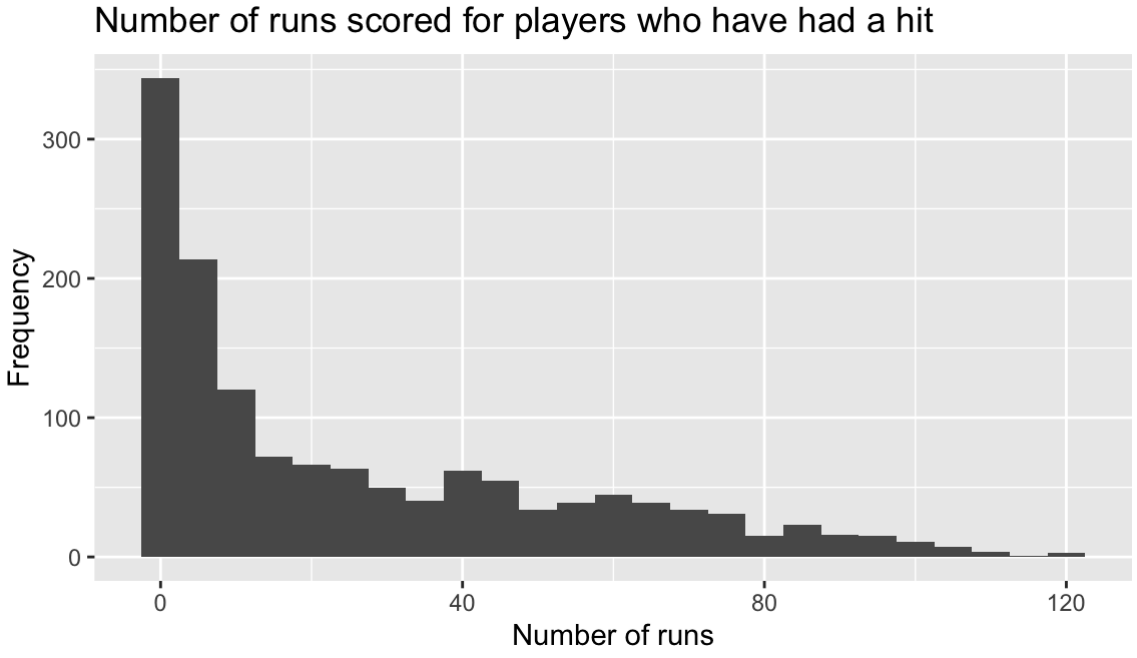
```
#i. & ii.
df_FieldingData <- Fielding %>%
  filter(yearID %in% c(1990, 2015)) %>%
  select(playerID, yearID, POS)

df_BattingData <- Batting %>%
  filter(yearID %in% c(1990, 2015))

df_PeopleData <- People %>%
  select(playerID, height, weight, birthYear)

df_FieldingData_P <- df_FieldingData %>%
  select(playerID, POS)

df_BattingData<- left_join(df_BattingData, df_PeopleData, by = "playerID")
df_BattingData <- left_join(df_BattingData, df_FieldingData_P, by = "playerID")

df_BattingData <- df_BattingData %>%
  mutate(age = yearID - birthYear)

df_BattingData <- df_BattingData %>%
  drop_na()

df_BattingData <- droplevels(df_BattingData)

df_BattingData <- df_BattingData %>%
  distinct(playerID, .keep_all = TRUE)

#glimpse(df_BattingData)
```

b. Create a histogram of the number of runs scored for players. Next create a histogram of the number of runs for all players who have had a hit. Why it is more reasonable to create a Poisson data for the second set than the first. [3 points]

```
df_BattingData %>%
  ggplot(aes(R))+
  geom_histogram(binwidth = 5)+
  labs(x="Number of runs",y="Frequency",
       title="Number of runs scored for players")
```




```
df_BattingData %>%
  filter(H > 0) %>%
  ggplot(aes(R))+
  geom_histogram(binwidth = 5)+
  labs(x="Number of runs",y="Frequency",
       title="Number of runs scored for players who have had a hit")
```



It is more reasonable to create a Poisson data for the set where players have had a hit because the players who have had no hit are not relevant to scoring runs.

c. Excluding players who have had no hit, construct a Poisson model of the number of runs as a function of the number of hits, the year as a factor, position played and player height and age in the relevant

year. Interpret the results and write the form of the fitted model (coefficients should be rounded to 2 significant figures). [8 points]

```
poissonmod <- glm(R ~ H + as.factor(yearID) + POS + height + age,
data = df_BattingData[!df_BattingData$H == 0,], family="poisson")
summary(poissonmod)
```

```
Call:
glm(formula = R ~ H + as.factor(yearID) + POS + height + age,
    family = "poisson", data = df_BattingData[!df_BattingData$H ==
        0, ])

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)           1.533e+00  2.026e-01   7.567 3.83e-14 ***
H                     1.412e-02  9.792e-05 144.243  < 2e-16 ***
as.factor(yearID)2015 8.320e-03  1.050e-02   0.792 0.428269
POS2B                -7.885e-03  1.761e-02  -0.448 0.654228
POS3B                -8.870e-02  2.244e-02  -3.952 7.74e-05 ***
POSC                 -1.724e-01  2.444e-02  -7.054 1.74e-12 ***
POSOF                 5.021e-02  1.357e-02   3.700 0.000216 ***
POSP                 -1.714e+00  4.696e-02 -36.497  < 2e-16 ***
POSSS                -6.123e-02  2.279e-02  -2.687 0.007208 **
height                7.357e-03  2.657e-03   2.769 0.005620 **
age                   5.081e-03  1.382e-03   3.677 0.000236 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 42577.6  on 1402  degrees of freedom
Residual deviance:  5748.2  on 1392  degrees of freedom
AIC: 11496

Number of Fisher Scoring iterations: 5
```

The form of the fitted model is: R ~ Pois (exp (1.5 + 0.014 x H + 0.0083 x isas.factor(yearID)2015 - 0.0079 x isPOS2B - 0.089 x isPOS3B - 0.17 x isPOSC + 0.05 x isPOSOF - 1.7 x isPOSP - 0.061 x isPOSSS + 0.0074 x height + 0.0051 x age))

The variable H has a p value very close to 0 at 2e-16. It is extremely significant in determining runs scored. We can deduce from the model that as hits by batters increase by 1 (and other variables held constant), the mean of runs scored increase by 1.412e-02. The variable age is also very significant with a p value at 0.00024. We can also deduce that as age increase by 1 year (and other variables held constant), the mean of runs scored increase by 5.081e-3. On the other hand, though height with a p value of 0.0056 is not as significant as H and age, it is also significant in determining runs scored as it is way below the 0.05 threshold. We deduce that as height increase by 1 inch (and other variables held constant), the mean of runs scored increase by 7.357e-03.

We also see that as.factor(yearID)2015 is statistically not significant in determining runs scored, with a p value of 0.43. This is far above the 0.05 (5%) threshold.

It can also be seen that POS largely has a negative relationship with runs scored. The only position with a strong positive relationship is POSOF with a p value of 0.00022 and a positive coefficient. The positions with the strongest negative relationship by reason of their p values are POSP, POSC and POS3B, with 2e-16,

1.74e-12 and 7.74e-5 respectively. POSSS has the least negative relationship with runs scored with a p value of 0.0072, though it is worth noting that this is still way below the 0.05 threshold. POS2B with a p value of 0.65 is the only position with no statistical significance in determining runs scored.

d. Find the p-value for each of the predictor variables in this model using analysis of variance. Interpret the results and mathematically explain what is meant by the p-value associated to each predictor. [5 points]

```
poissonmod <- glm(R ~ H + as.factor(yearID) + POS + height + age,
  family = "poisson", data = df_BattingData[!df_BattingData$H == 0,])
Anova(poissonmod)
```

```
Analysis of Deviance Table (Type II tests)

Response: R
                  LR Chisq Df Pr(>Chisq)
H                  22326.0  1  < 2.2e-16 ***
as.factor(yearID)      0.6  1  0.4282241
POS                 2340.6  6  < 2.2e-16 ***
height                 7.7  1  0.0055982 **
age                   13.5  1  0.0002416 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p values for the variables H and POS are the least for all the predictor variables, at 2.2e-16. This is very close to 0, indicating that the Hits by batters and Position are statistically very unlikely to not be important in determining runs scored. For these variables, we reject the null hypothesis that the coefficients are equal to 0.

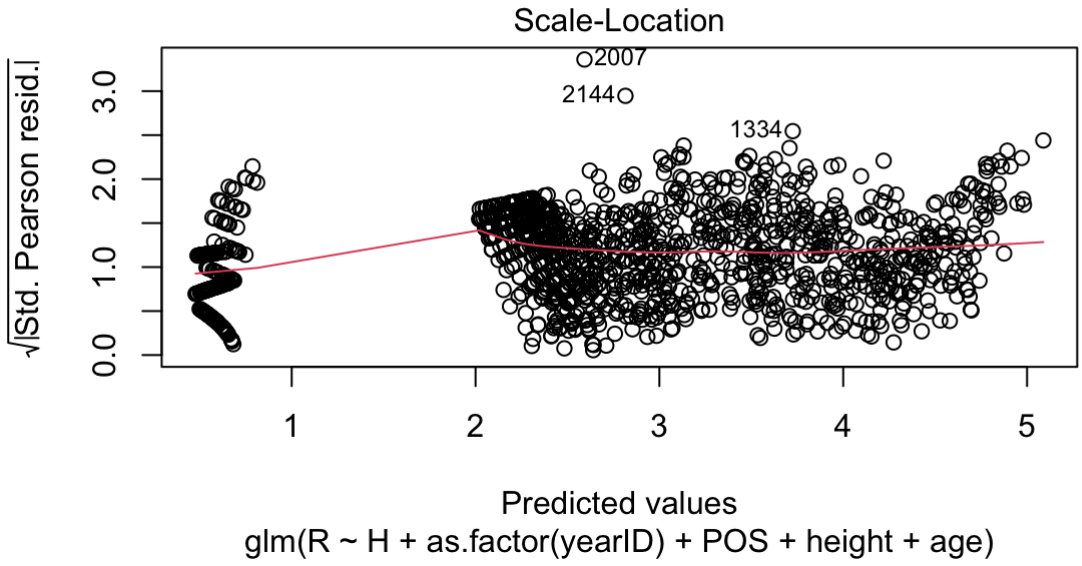
The p value for the variable as.factor(yearID) is 0.43. This is way above the 5% cut-off at 0.05 and in the accepting region. For this reason we accept the null hypothesis that the coefficient is equal to 0 and conclude that yearID is statistically unimportant in determining runs scored.

The p values for the variable height is 0.0056. Though this is close to the 0.05 cut-off, it is still below it and therefore a significant predictor variable in determining runs scored. We therefore reject the null hypothesis stating that the coefficient is equal to 0.

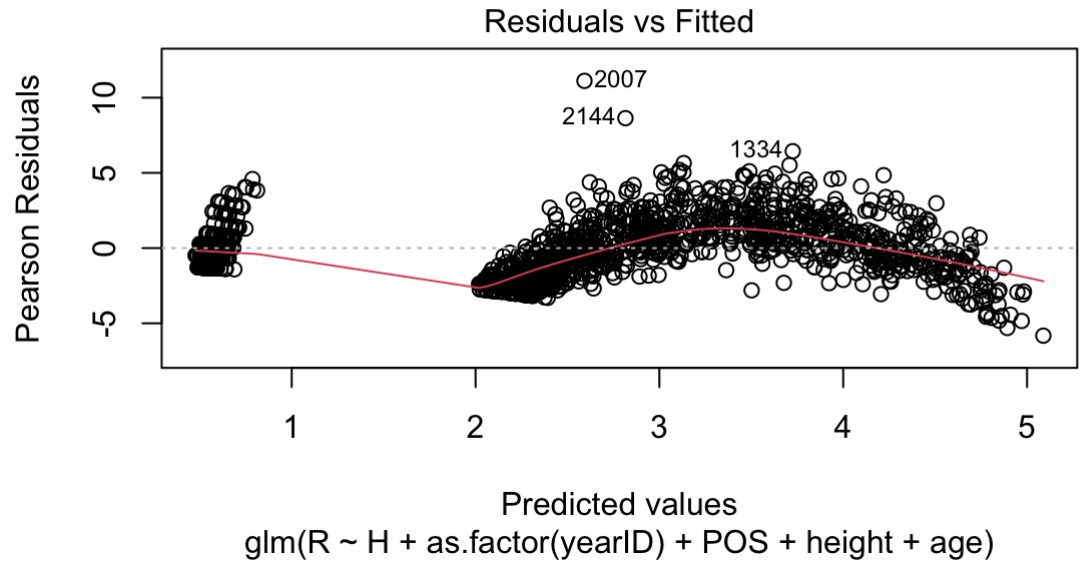
The p value for age is also very small at 0.00024. This is also close to 0 and we conclude that age is statistically very unlikely to not be important in determining runs scored. We reject the null hypothesis that the coefficient is equal to 0.

e. State and evaluate the assumptions of Poisson model. Comment on any weird pattern. [9 points]

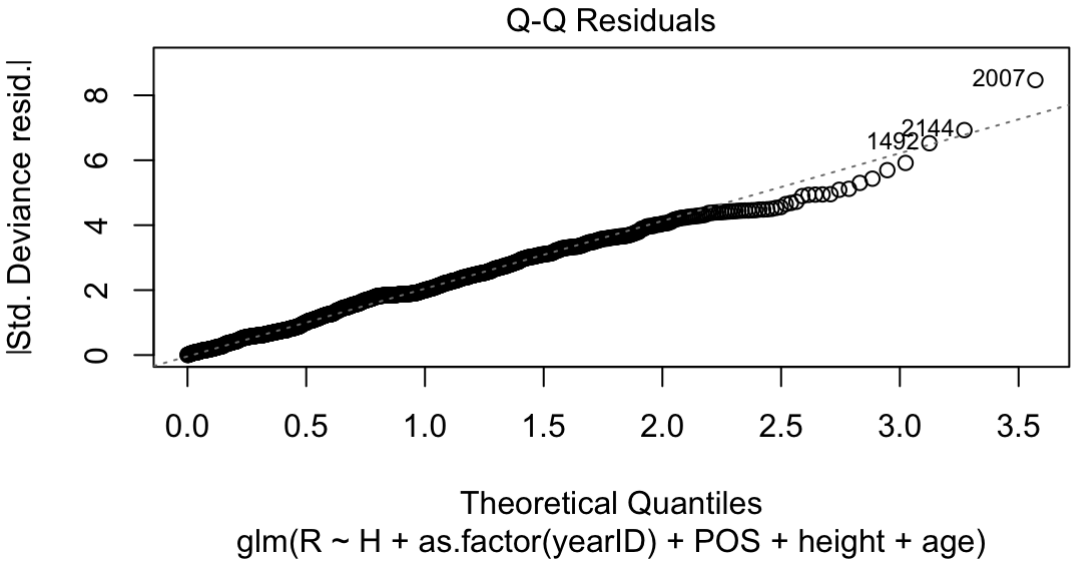
```
plot(poissonmod,which=3)
```




```
dispersiontest(poissonmod,trafo=1)
plot(poissonmod,which=1)
```




```
plot(poissonmod,which=2)
```




```
    Overdispersion test

data:  poissonmod
z = 17.462, p-value < 2.2e-16
alternative hypothesis: true alpha is greater than 0
sample estimates:
   alpha
2.763484
```

1. Dispersion assumption- Equality of variance and mean (Variance = Mean): In order to see if the assumption that variance = mean is reasonable for this dataset, we create a plot of the absolute value of residuals versus predicted means. This should look flat but looking at the plot, the red line is definitely not flat. Also if we look into the equidispersion assumption, with the dispersiontest() method, we notice that there is a slight overdispersion in the data. Judging from the scale-location plot, the overdispersion initially increases linearly with the prediction, and then seem to flatten out with progression. The reported alpha is greater than 0, at 2.76, corroborating these findings. This slight overdispersion suggests that we have not accounted for all of the important predictors in our model.

2. Linearity: This assumes that there is a linear relationship between the numeric variables on the right hand side of the equation, and the log (count) on the left hand side. This assumption can be checked by plotting the (deviance) residuals vs fitted and seeing if the plot looks fairly flat. Looking at the plot however, we see that the red line starts off with a sharp negative slope and ends with a parabolic curve.Therefore the assumption of linearity is violated.

3. Distribution: With the distribution assumption, we assume that the errors (deviance residuals) are poisson distributed. For this we make a plot of Q-Q residuals. This actually doesn't look bad. The line is pretty straight and follows the dotted black line, and only starts to deviate slightly at the upper tail end. This is likely due to a few outliers.

4. Independence: This assumption refers to the errors (deviance residuals) as a function of order of datapoints being independent of each other. This assumes that the outcomes observed are independent of one another. This assumption is not statistically. Instead, the study design of data collection is checked to ensure that the outcomes are independent.

f. Now create a new model that includes teamID as a random effect. Ensure there are no fit warnings. What does the results tell us about the importance of team on number of runs that players score? [4

points]

```
new_poissonmod <- glmer(R ~ H + as.factor(yearID) + POS + height + age +
                          (1|teamID), family = "poisson",
                        data = df_BattingData[!df_BattingData$H == 0,],
                        nAGQ=0)
summary(new_poissonmod)
```

```
Generalized linear mixed model fit by maximum likelihood (Adaptive
  Gauss-Hermite Quadrature, nAGQ = 0) [glmerMod]
 Family: poisson  ( log )
Formula: R ~ H + as.factor(yearID) + POS + height + age + (1 | teamID)
   Data: df_BattingData[!df_BattingData$H == 0, ]

     AIC      BIC   logLik deviance df.resid
 11417.2  11480.1  -5696.6  11393.2     1391

Scaled residuals:
    Min      1Q  Median      3Q     Max
-5.6078 -1.3711 -0.3106  1.0765 10.8655

Random effects:
 Groups Name        Variance Std.Dev.
 teamID (Intercept) 0.004298 0.06556
Number of obs: 1403, groups:  teamID, 33

Fixed effects:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)            1.690e+00  2.081e-01   8.122 4.60e-16 ***
H                      1.425e-02  9.991e-05 142.671  < 2e-16 ***
as.factor(yearID)2015  2.399e-02  1.123e-02   2.136  0.03270 *
POS2B                 -1.191e-02  1.787e-02  -0.666  0.50536
POS3B                 -9.941e-02  2.287e-02  -4.347 1.38e-05 ***
POSC                  -1.660e-01  2.463e-02  -6.741 1.57e-11 ***
POSOF                  4.932e-02  1.367e-02   3.607  0.00031 ***
POSP                  -1.690e+00  4.722e-02 -35.793  < 2e-16 ***
POSSS                 -6.615e-02  2.295e-02  -2.883  0.00394 **
height                 5.179e-03  2.723e-03   1.902  0.05717 .
age                    4.434e-03  1.411e-03   3.141  0.00168 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) H      a.(ID) POS2B  POS3B  POSC   POSOF  POSP   POSSS
H           -0.024
as.(ID)2015  0.052  0.018
POS2B       -0.465 -0.022 -0.020
POS3B       -0.160 -0.098 -0.021  0.309
POSC        -0.130  0.192  0.006  0.271  0.173
POSOF       -0.228 -0.048 -0.005  0.493  0.351  0.308
POSP         0.022  0.217  0.001  0.098  0.066  0.128  0.138
POSSS       -0.196 -0.112  0.006  0.322  0.233  0.173  0.356  0.062
height      -0.976 -0.016 -0.085  0.423  0.131  0.099  0.184 -0.046  0.165
age         -0.261 -0.067 -0.007  0.157  0.092  0.024  0.090  0.000  0.107
            height
H
as.(ID)2015
POS2B
POS3B
POSC
POSOF
POSP
POSSS
```

```
height
age          0.067
```

We can see from the intercept of the first model that the overall mean runs scored is 1.53, whereas the new model has an overall mean of 1.69. This implies that team have a positive relationship with the mean number of runs scored.

This new model is marginally better than the first. The yearID variable is now significant, with a p value of 0.033, though age and height are now slightly less significant than they were. It can also be observed that the AIC for the new model is slightly less than for the initial model, at 11417 against 11496 for the first model. This implies a slight improvement in the model's goodness of fit and avoidance of overfitting.

g. What is the mean number of runs could you expect 27-year-old, 85-inch-tall outfielders playing for the Cleveland Indians in 2015 with 50 hits to have scored? comment on the result. [2 points]

```
predict(new_poissonmod, newdata = data.frame(age = 27, height = 85,
        POS = "OF", yearID = 2015, teamID = "CLE", H = 50 ))
```

```
       1
3.057155
```

The point estimate of 3.1 for this dataset is the mean number of runs expected. This is much higher than the overall mean of 1.7 predicted by the model.

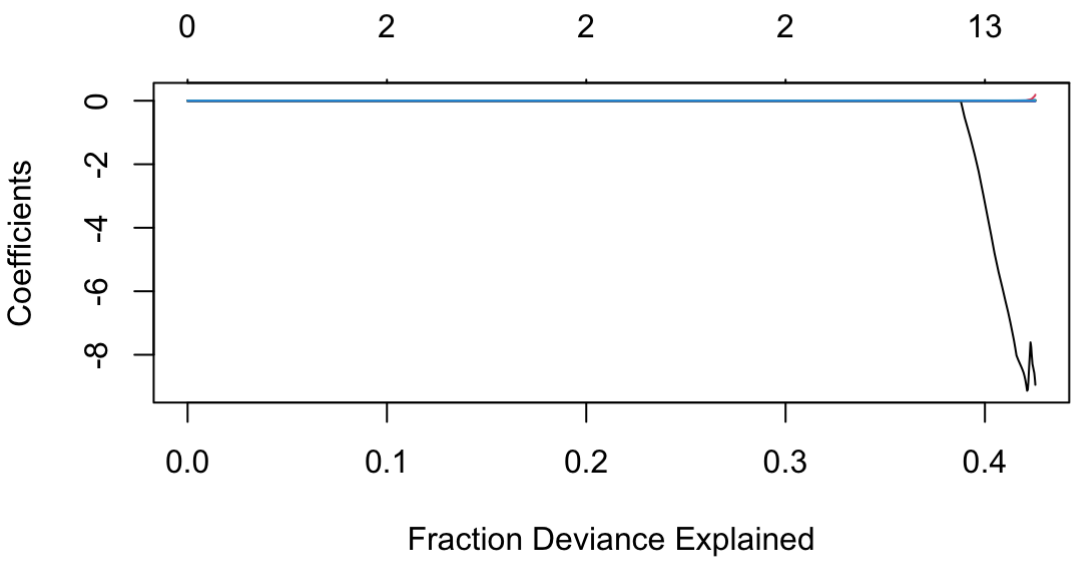
3. Lasso Regression for Logistic Regression [30 points]

a. From 'Teams' dataset, create a new dataset, df_DivWinners, by choosing data from the years 1990 to 2015 and removing all the variables that are team identifiers in the dataset, as well as 'lgID', 'Rank','franchID','divID', 'WCWin','LgWin','WSwin','name' and 'park'. Drop incomplete cases from the dataset 'df_DivWinners'. Split the resulting into a 80% training and a 20% testing set so that the variable 'DivWin' is balanced between the two datasets. Use the seed 123. [3 points]

```
df_DivWinners <- Teams %>%
  filter(yearID >= 1990 & yearID <= 2015 ) %>%
  select(-lgID, -Rank, -franchID, -divID, -WCWin, -LgWin, -WSWin, -name, -park,
         -teamIDretro, -teamIDlahman45, -teamIDBR, -teamID) %>%
  drop_na()

set.seed(123)
training_samples <- df_DivWinners$DivWin %>%
  createDataPartition(p = 0.8, list = FALSE)
training_data <- df_DivWinners[training_samples,]
test_data <- df_DivWinners[-training_samples,]
```

b. Use the training data to fit a logistic regression model using the 'glmnet' command. Plot residual deviance against number of predictors. Comment on the result. [3 points]

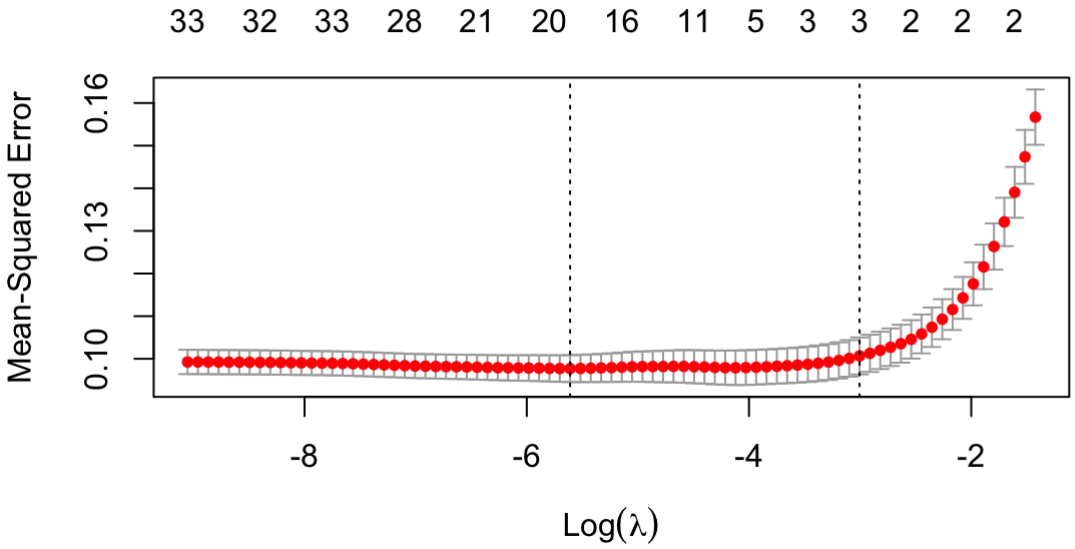
```
Divwin <- as.numeric(factor(training_data$DivWin,
                            levels = c("N", "Y"),
                            labels = c(0, 1)))
Divwinnerspredict <- model.matrix(~. -1 -DivWin, data
                                  = training_data, family = "binomial" )
Divwinnersfit <- glmnet(Divwinnerspredict, Divwin)

plot(Divwinnersfit,xvar="dev")
```



This tells us that with just 13 nonzero coefficients other than intercept, already 40% of the deviance is explained.

c. Now use cross-validation to choose a moderately conservative model. State the variables you will include. [3 points]

```
Divwinnersfit

set.seed(123)
Divwinnerscv <- cv.glmnet(Divwinnerspredict,Divwin)
plot(Divwinnerscv)
```




```
# the coefficients when lambda is 0.007022
Divwinners13 <-coef(Divwinnersfit, s = 0.007022)
Divwinners13@Dimnames[[1]][1+Divwinners13@i]

Divwinnersopt <-coef(Divwinnersfit,s=Divwinnerscv$lambda.1se)

Divwinnersmax<-coef(Divwinnersfit,s=Divwinnerscv$lambda.min)
Divwinnersmax@Dimnames[[1]][1+Divwinnersmax@i]
```

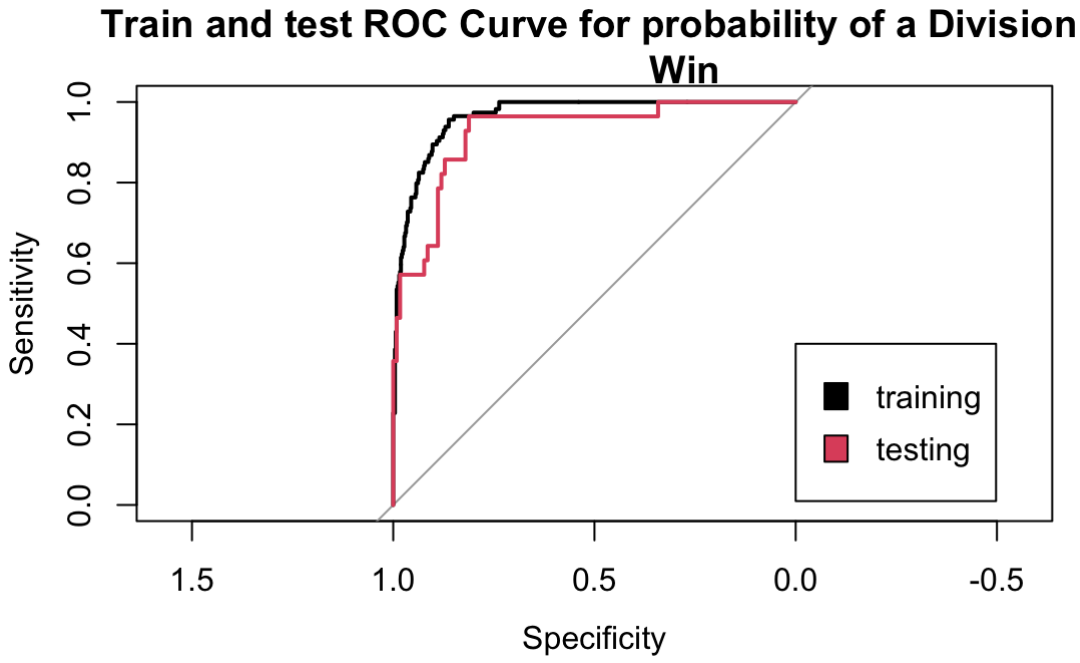
```
Call:  glmnet(x = Divwinnerspredict, y = Divwin)

   Df  %Dev   Lambda
1   0  0.00 0.240900
2   2  6.38 0.219500
3   2 11.70 0.200000
4   2 16.12 0.182200
5   2 19.79 0.166000
6   2 22.84 0.151300
7   2 25.37 0.137800
8   2 27.47 0.125600
9   2 29.21 0.114400
10  2 30.66 0.104300
11  2 31.86 0.095010
12  2 32.86 0.086570
13  2 33.69 0.078880
14  2 34.38 0.071870
15  2 34.95 0.065490
16  3 35.50 0.059670
17  3 36.04 0.054370
18  3 36.48 0.049540
19  3 36.85 0.045140
20  3 37.16 0.041130
21  3 37.42 0.037480
22  3 37.63 0.034150
23  3 37.80 0.031110
24  3 37.95 0.028350
25  4 38.10 0.025830
26  4 38.24 0.023540
27  4 38.35 0.021440
28  5 38.45 0.019540
29  5 38.56 0.017800
30  5 38.65 0.016220
31  5 38.73 0.014780
32  5 38.79 0.013470
33 10 38.98 0.012270
34 11 39.26 0.011180
35 11 39.49 0.010190
36 11 39.69 0.009283
37 12 39.92 0.008458
38 13 40.13 0.007707
39 13 40.32 0.007022
40 14 40.48 0.006398
41 16 40.69 0.005830
42 16 40.87 0.005312
43 17 41.03 0.004840
44 17 41.16 0.004410
45 17 41.28 0.004018
46 20 41.37 0.003661
47 20 41.45 0.003336
48 20 41.52 0.003040
49 21 41.58 0.002770
50 23 41.67 0.002524
51 23 41.76 0.002299
```

```
52 22 41.83 0.002095
53 21 41.89 0.001909
54 22 41.94 0.001739
55 21 41.99 0.001585
56 23 42.03 0.001444
57 25 42.08 0.001316
58 26 42.12 0.001199
59 27 42.15 0.001092
60 27 42.18 0.000995
61 27 42.20 0.000907
62 28 42.23 0.000826
63 28 42.25 0.000753
64 28 42.27 0.000686
65 29 42.29 0.000625
66 30 42.30 0.000570
67 30 42.32 0.000519
68 33 42.34 0.000473
69 33 42.38 0.000431
70 32 42.43 0.000393
71 31 42.45 0.000358
72 32 42.45 0.000326
73 32 42.47 0.000297
74 32 42.48 0.000271
75 32 42.49 0.000247
76 32 42.50 0.000225
77 32 42.50 0.000205
78 33 42.51 0.000187
79 33 42.52 0.000170
80 33 42.52 0.000155
81 33 42.52 0.000141
82 33 42.53 0.000129
83 33 42.53 0.000117
 [1] "(Intercept)" "yearID"      "W"           "L"           "R"
 [6] "X2B"         "CS"          "HBP"         "CG"          "HA"
[11] "HRA"         "BBA"         "FP"          "attendance"  "BPF"
 [1] "(Intercept)" "yearID"      "W"           "L"           "R"
 [6] "AB"          "X2B"         "X3B"         "HR"          "SO"
[11] "SB"          "CS"          "HBP"         "CG"          "HA"
[16] "HRA"         "BBA"         "DP"          "FP"          "attendance"
[21] "BPF"
```

I have decided to go with the coefficients recommended based on the minimum lambda (and therefore minimum Mean Squared Error). The coefficients recommended are “(Intercept)”, “yearID”, “W”, “L”, “R”, “AB”, “X2B”, “X3B”, “HR”, “SO”, “SB”, “CS”, “HBP”, “CG”, “HA”, “HRA”, “BBA”, “DP”, “FP”, “attendance”, “BPF”

d. Fit the model on the training data using glm(), interpret the results and write the form of the model (coefficients should be rounded to 2 significant figures). Then predict on the testing data. Plot comparative ROC curves and summarise your findings. [9 points]

```
Divwinlogimod <-glm(as.factor(DivWin) ~ yearID + W + L + R + AB + X2B + X3B
+ HR + SO + SB + CS + HBP + CG + HA + HRA + BBA + DP + FP + attendance + BPF,       f
amily="binomial",data= training_data)
summary(Divwinlogimod)

test_predictdivwin <- predict(Divwinlogimod, newdata = test_data,
                              type = "response")

train_predictdivwin <- predict(Divwinlogimod, type = "response")

roc_train <- roc(response = training_data$DivWin,
                 predictor = train_predictdivwin, plot = TRUE,
                 main ="Train and test ROC Curve for probability of a Division
                 Win",auc = TRUE)
roc_test <- roc(response = test_data$DivWin,
                 predictor = test_predictdivwin, plot = TRUE, auc = TRUE,
                add = TRUE, col = 2)

legend(0,0.4,legend=c("training","testing"),fill=1:2)
```



```
Call:
glm(formula = as.factor(DivWin) ~ yearID + W + L + R + AB + X2B +
    X3B + HR + SO + SB + CS + HBP + CG + HA + HRA + BBA + DP +
    FP + attendance + BPF, family = "binomial", data = training_data)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.098e+02  1.118e+02  -0.982 0.326087
yearID       9.217e-02  5.243e-02   1.758 0.078755 .
W            2.169e-01  1.244e-01   1.744 0.081211 .
L           -2.364e-01  1.326e-01  -1.782 0.074687 .
R           -9.789e-03  5.448e-03  -1.797 0.072332 .
AB          -3.324e-03  3.282e-03  -1.013 0.311235
X2B         -2.765e-03  8.564e-03  -0.323 0.746772
X3B          4.748e-03  2.386e-02   0.199 0.842276
HR           6.281e-03  9.396e-03   0.668 0.503868
SO          -8.811e-04  2.024e-03  -0.435 0.663293
SB           4.357e-03  6.746e-03   0.646 0.518385
CS          -2.950e-02  2.036e-02  -1.449 0.147400
HBP         -2.784e-02  1.520e-02  -1.831 0.067046 .
CG           8.173e-02  4.720e-02   1.731 0.083374 .
HA           1.327e-02  3.691e-03   3.595 0.000324 ***
HRA         -3.287e-03  1.116e-02  -0.295 0.768327
BBA          4.300e-03  3.637e-03   1.182 0.237086
DP          -2.149e-02  1.196e-02  -1.796 0.072539 .
FP          -7.513e+01  7.378e+01  -1.018 0.308588
attendance   6.764e-07  3.038e-07   2.227 0.025961 *
BPF          4.689e-02  4.247e-02   1.104 0.269526
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 577.06  on 584  degrees of freedom
Residual deviance: 221.05  on 564  degrees of freedom
AIC: 263.05

Number of Fisher Scoring iterations: 8
```

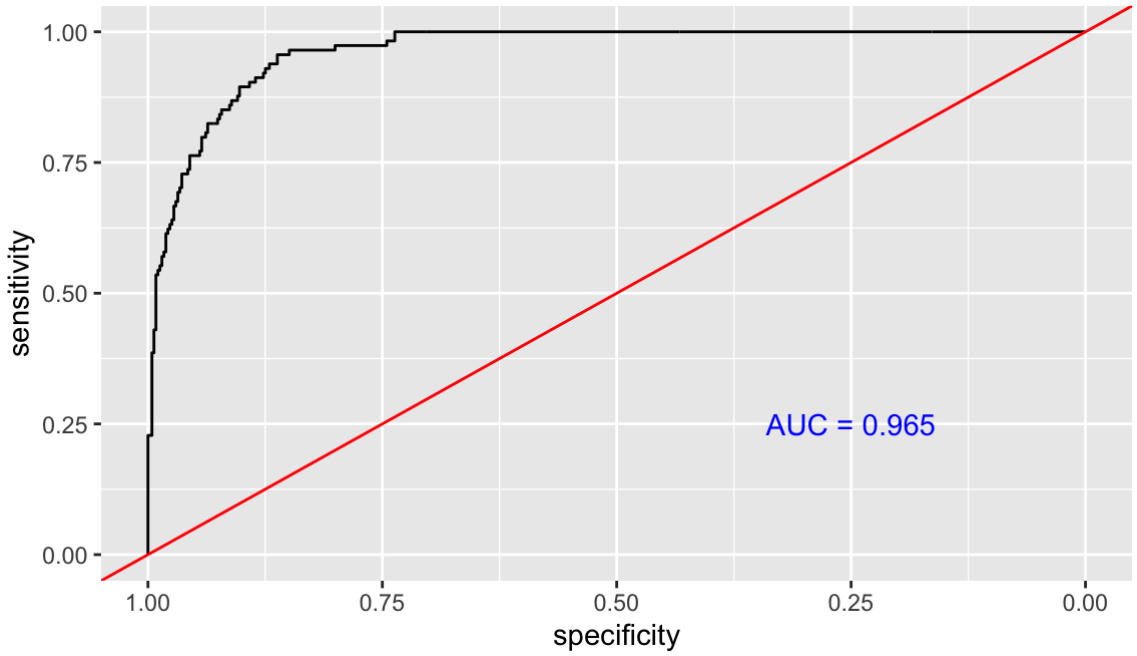
From the divwinlogimod model, we see that only one variable is significantly important (statistically it is very unlikely to not be important). This is the HA (Hits allowed) variable with a p-value very close to 0. The second most significant variable is attandance with a p-value roughly half of 0.05 at 0.026. Other variables worth noting with p-values slightly above the 0.05 threshold are yearID, W (wins), L (Losses), R (Runs scored), HBP (Batters hit by pitch), CG (Complete games), and DP (Double plays).

The form of the fitted model is: DivWin ~ B (inverse logit (-110 + 0.092 x yearID + 0.22 x W -0.24 x L - 0.0098 x R - 0.0032 x AB - 0.0028 x X2B + 0.0047 x X3B + 0.0063 x HR - 0.00088 x SO + 0.0044 x SB - 0.030 x CS - 0.028 x HBP - 0.082 x CG + 0.013 x HA - 0.0033 x HRA + 0.0043 x BBA - 0.021 x DP - 75 x FP + 0.00000068 x attendance + 0.047 x BPF), 1)

The training and test ROC curves are reasonably similar. They both follow the same pattern, particularly at the start and end tails. The discrepancies begin to show around the curve, where the training ROC is smooth and the test ROC is a little rougher and shorter. Overall, it can be concluded that the model is good and there is no overfitting.

e. Find Youden's index for the training data and calculate confusion matrices at this cutoff for both training and testing data. Comment on the quality of the model. [6 points]

```
roc_train <- roc(response = training_data$DivWin,
                 predictor = train_predictdivwin, auc=TRUE)
ggroc(roc_train) +
 geom_abline(aes(intercept = 1, slope = 1),colour="red") +
 annotate(geom = "text", x = 0.25, y = 0.25, label = paste("AUC =", round (auc(roc_tr
ain), 3)), colour = "blue" )
```




```
youdentrain <- coords(roc_train, "b", best.method = "youden", transpose = TRUE)
youdentrain
youdentrain[2]+youdentrain[3]


predict_train <- ifelse (predict(Divwinlogimod, type = "response")
                   >= 0.14, "Yes", "No")

predict_test <- ifelse (predict(Divwinlogimod, newdata = test_data, type = "respons
e") >= 0.14, "Yes", "No")

train_table <- table(training_data$DivWin, predict_train )
train_table

test_table <- table(test_data$DivWin, predict_test)
test_table
#names(dimnames(traincm)<- list("Actual","Predicted")
```

```
  threshold specificity sensitivity
  0.1353870   0.8619958   0.9561404
specificity
   1.818136
   predict_train
     No Yes
  N 406  65
  Y   6 108
   predict_test
    No Yes
  N 96  21
  Y  3  25
```

The best threshold is right around 0.14, and this achieves a sensitivity + specificity of 1.82 on the training data. This is very good, considering that 1 is achieved at random. The area under the ROC curve (auc) is 0.965. This is very close to 1, which is for a perfect model. Taking these figures into account, the model is a good fit to the data.

f. Calculate the sensitivity+specificity on the testing data as a function of divID and plot as a bar chart. Comment on the result. [6 points]

```
divss <- Teams[, c("attendance", 'divID')]
divss <- inner_join(test_data, divss, by = "attendance")

divss <- divss %>%
distinct(attendance, .keep_all = TRUE)

sum_divss <- function(pred, true_lab, positive_class) {
  roc_new <- roc(ifelse(true_lab == positive_class, 1, 0), pred)

  cut_off <- 0.14

  sensitivity_value <- sensitivity(roc_new, cut_off = cut_off)
  specificity_value <- specificity(roc_new, cut_off = cut_off)

  return(c(Sensitivity = sensitivity_value, Specificity = specificity_value))
}
#classW <- sum_divss(predict(Divwinlogimod, newdata = divss, type = "response"),
                    #Divwinlogimod$DivID, positive_class = "W")

#classE <- sum_divss(predict(Divwinlogimod, newdata = divss, type = "response"),
                    #Divwinlogimod$DivID, positive_class = "E")

#classC <- sum_divss(predict(Divwinlogimod, newdata = divss, type = "response"),
                    #Divwinlogimod$DivID, positive_class = "C")

#add_divss <- sapply(list(classW, classE, classC), sum)

#ssdf <- data.frame(Class = c("W", "E", "C"), Sum_SS = add_divss)

#ggplot(ssdf, aes(x = Class, y = Sum_SS, fill = Class)) +
  #geom_bar(stat = "identity") +
  #labs(title = "Sum of Sensitivity and Specificity for Each DivID class",
       #x = "Class", y = "Sum of Sensitivity and Specificity") +
  #theme_minimal()
```