

# Machine Learning - Day 2

Data modelling

*Mariachiara Fortuna*

## Libraries

```
library(dplyr)

##
## 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(ggplot2)
library(here)

## here() starts at /home/chiara/dev/machineLearningClass
```

## DATA IMPORT

```
dc <- file.path(here(), "data", "dc-wikia-data.csv") %>%
  read.csv(na.strings = "")

max_year <- max(dc$YEAR, na.rm = T)

dc <- dc %>%
  mutate(active_years = max_year - YEAR)
```

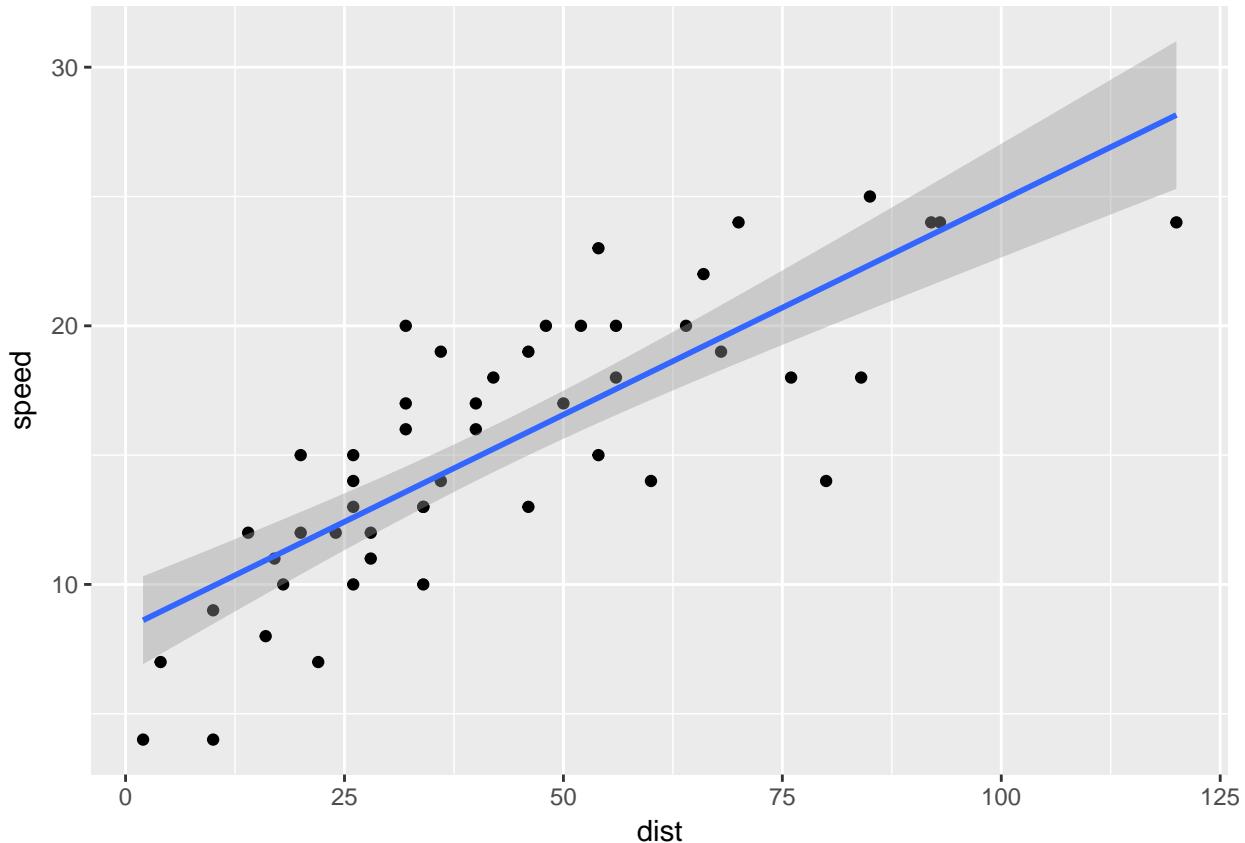
## DATA MODELING

### LINEAR REGRESSION

Simple linear regression

Easy ex

```
ggplot(data = cars, aes(x = dist, y = speed)) +
  geom_point() +
  geom_smooth(method='lm', formula=y~x)
```



```
m1 <- lm(data = cars, speed ~ dist)

summary(m1)

##
## Call:
## lm(formula = speed ~ dist, data = cars)
##
## Residuals:
##      Min    1Q   Median    3Q   Max 
## -7.5293 -2.1550  0.3615  2.4377  6.4179 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 8.28391   0.87438  9.474 1.44e-12 ***
## dist        0.16557   0.01749  9.464 1.49e-12 ***  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.156 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438 
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

```
residuals(m1)

##           1            2            3            4            5            6
## -4.61504079 -5.93958139 -1.94617594 -4.92639228 -2.93298684 -0.93958139
##           7            8            9            10           11           12
```

```

## -1.26412199 -2.58866258 -3.91320318 -0.09855441 -1.91979773 1.39814831
##          13          14          15          16          17          18
##  0.40474287 -0.25752743 -0.91979773  0.41133742 -0.91320318 -0.91320318
##          19          20          21          22          23          24
## -2.90001408  1.41133742 -0.24433833 -4.21796012 -7.52931161  3.40474287
##          25          26          27          28          29          30
##  2.41133742 -2.22455467  2.41793197  1.09339137  3.41793197  2.09339137
##          31          32          33          34          35          36
##  0.43771563  2.76225622  0.44431018 -2.86704131 -4.19158191  4.75566167
##          37          38          39          40          41          42
##  3.09998592 -0.54250072  6.41793197  3.76885078  3.10658048  2.44431018
##          43          44          45          46          47          48
##  1.11976958  2.78863443  5.77544533  4.12636413  0.48387749  0.31830992
##          49          50
## -4.15201460  2.64285051

```

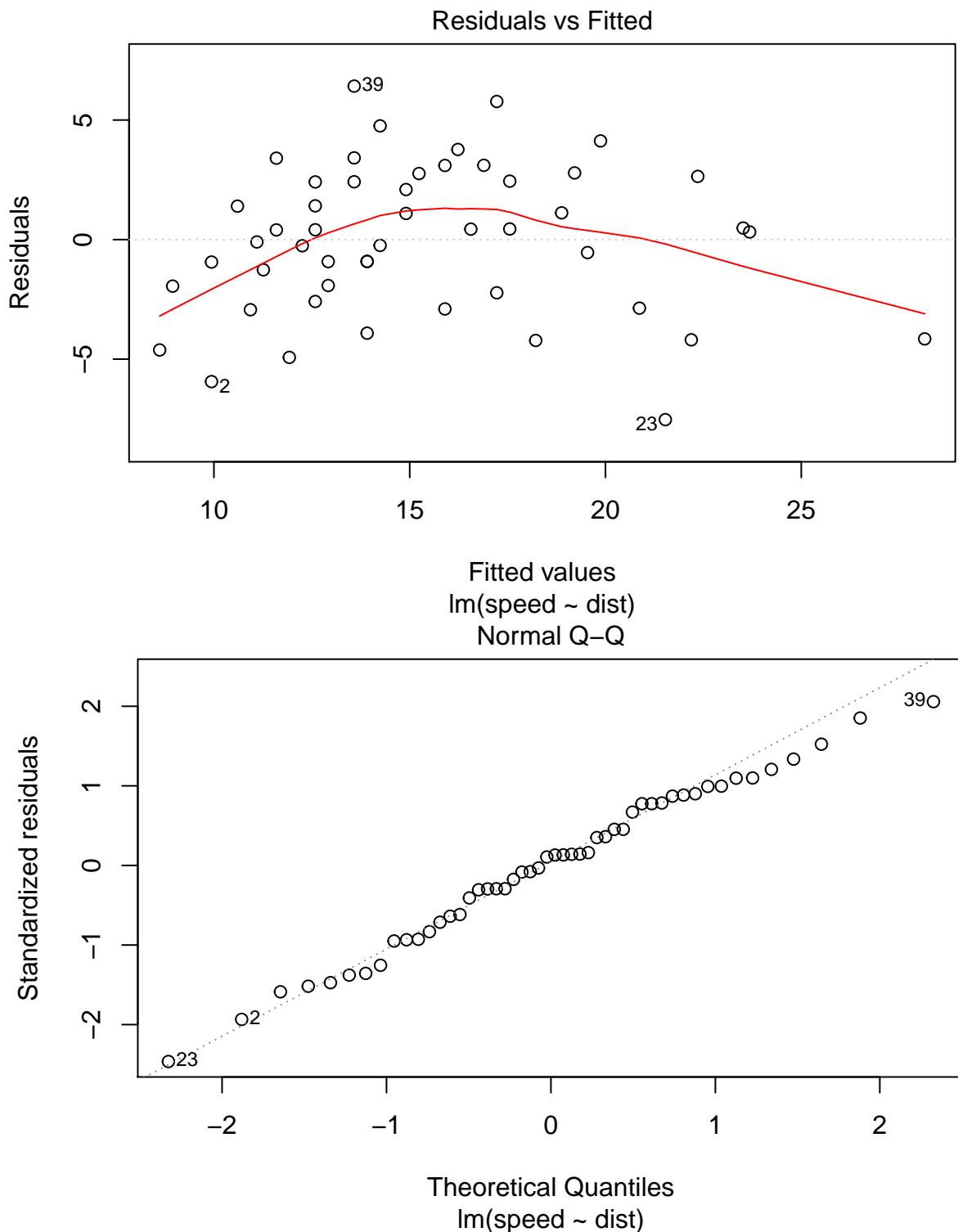
```
predict(m1)
```

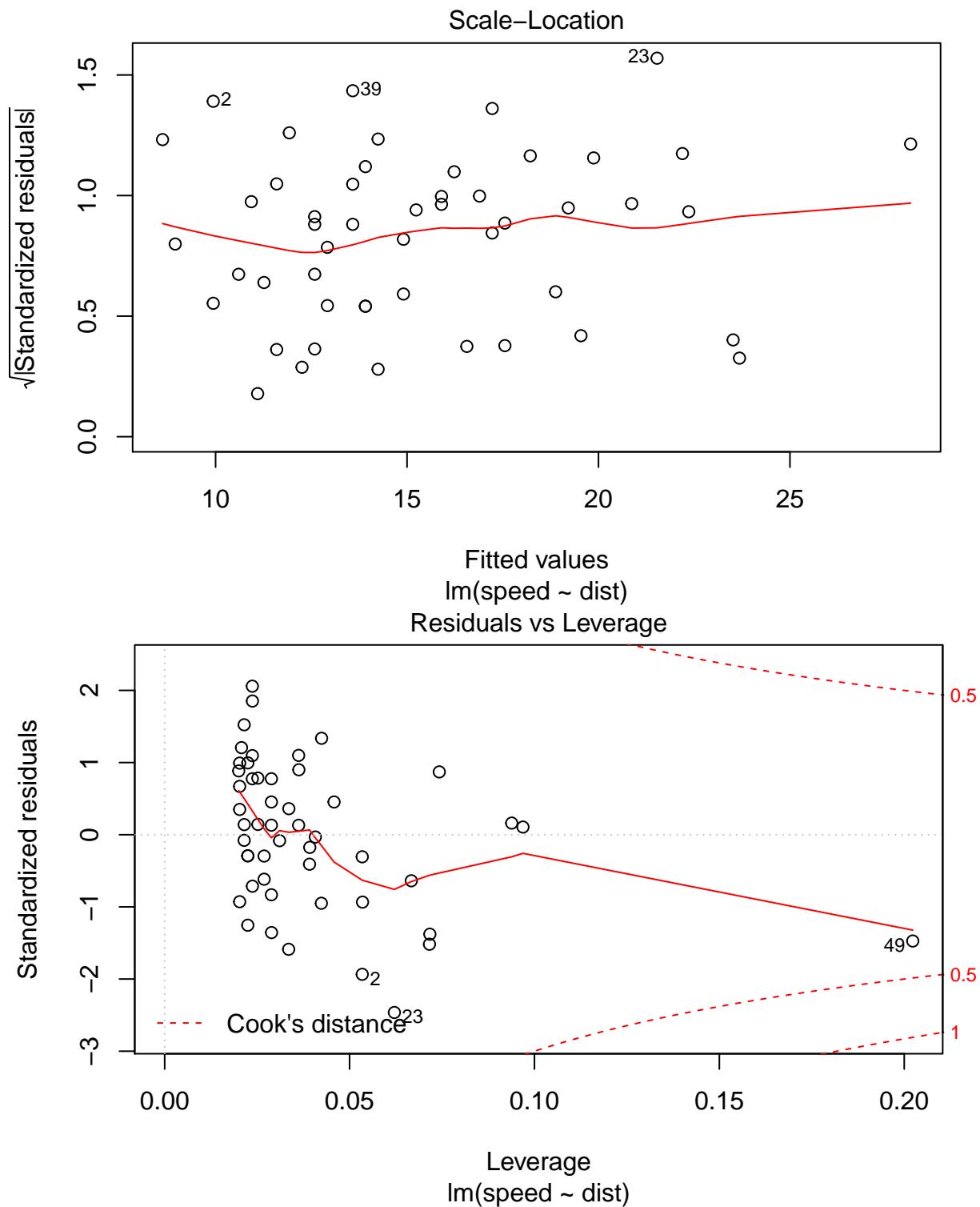
```

##          1          2          3          4          5          6          7
##  8.615041  9.939581  8.946176 11.926392 10.932987  9.939581 11.264122
##          8          9         10         11         12         13         14
## 12.588663 13.913203 11.098554 12.919798 10.601852 11.595257 12.257527
##          15         16         17         18         19         20         21
## 12.919798 12.588663 13.913203 13.913203 15.900014 12.588663 14.244338
##          22         23         24         25         26         27         28
## 18.217960 21.529312 11.595257 12.588663 17.224555 13.582068 14.906609
##          29         30         31         32         33         34         35
## 13.582068 14.906609 16.562284 15.237744 17.555690 20.867041 22.191582
##          36         37         38         39         40         41         42
## 14.244338 15.900014 19.542501 13.582068 16.231149 16.893420 17.555690
##          43         44         45         46         47         48         49
## 18.880230 19.211366 17.224555 19.873636 23.516123 23.681690 28.152015
##          50
## 22.357149

```

```
plot(m1)
```



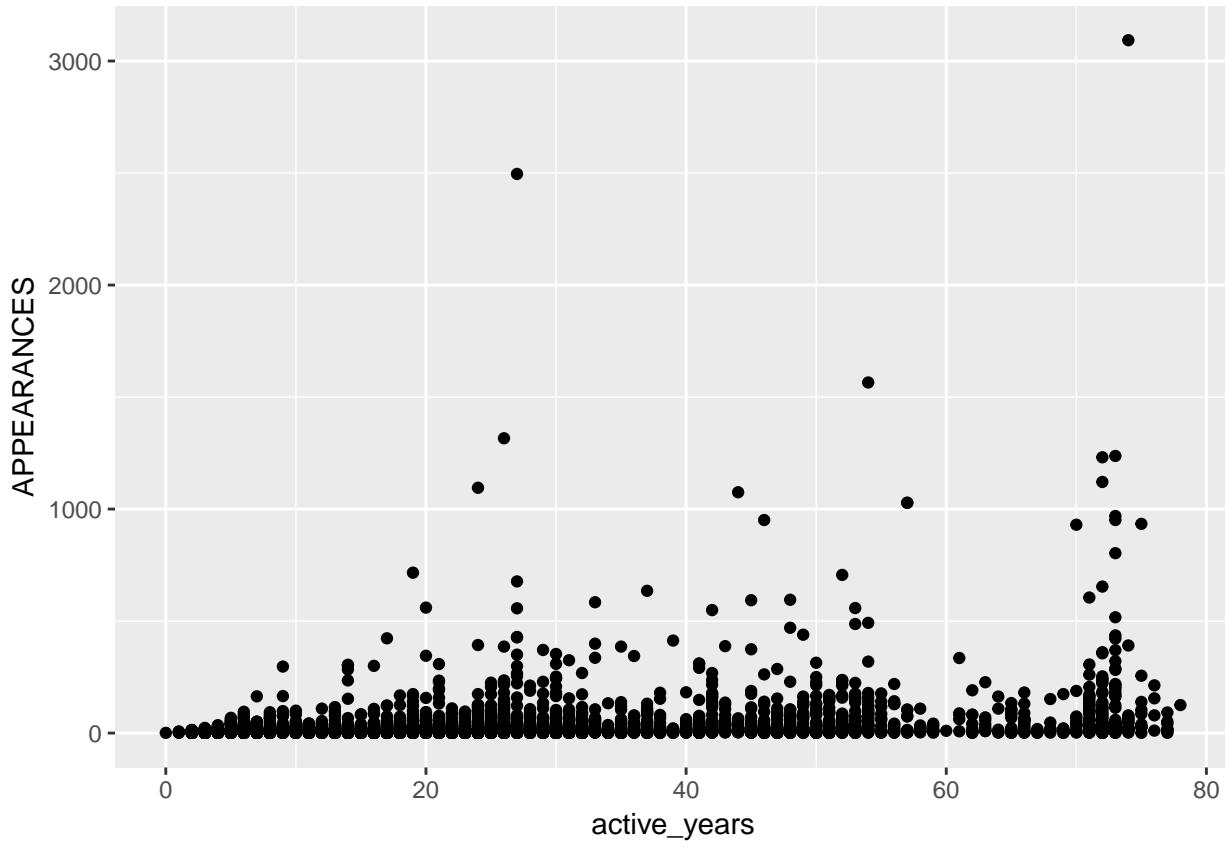


Comics example

```
dc_small <- dc %>%
  filter(APPEARANCES > 20)
```

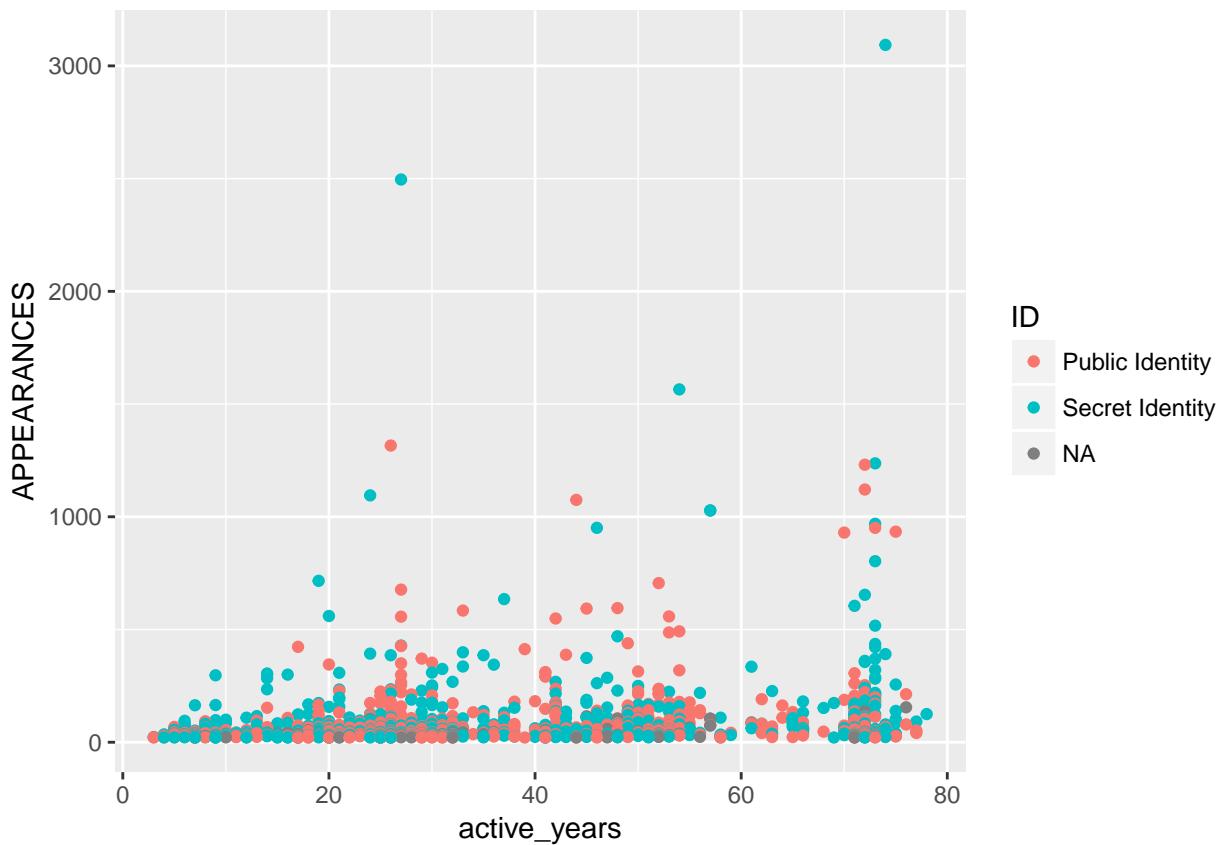
```
dc %>%
  ggplot(aes(x = active_years, y = APPEARANCES)) +
  geom_point()
```

## Warning: Removed 415 rows containing missing values (geom\_point).



```
dc_small %>%
  ggplot(aes(x = active_years, y = APPEARANCES, col = ID)) +
  geom_point()
```

## Warning: Removed 1 rows containing missing values (geom\_point).

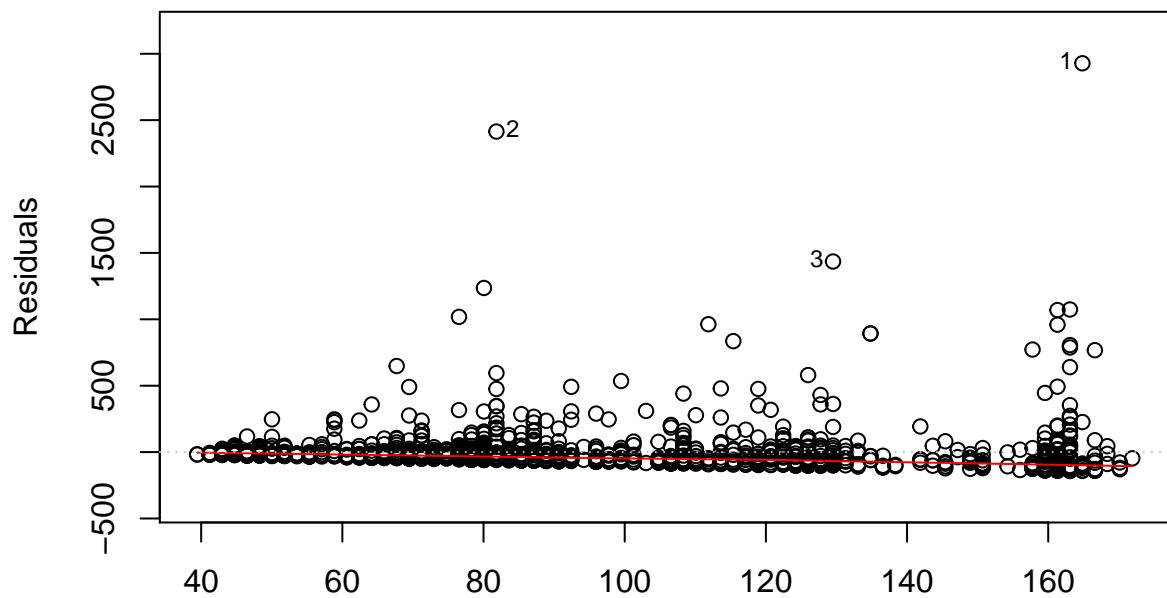


```
m1 <- lm(data=dc_small, APPEARANCES ~ active_years)

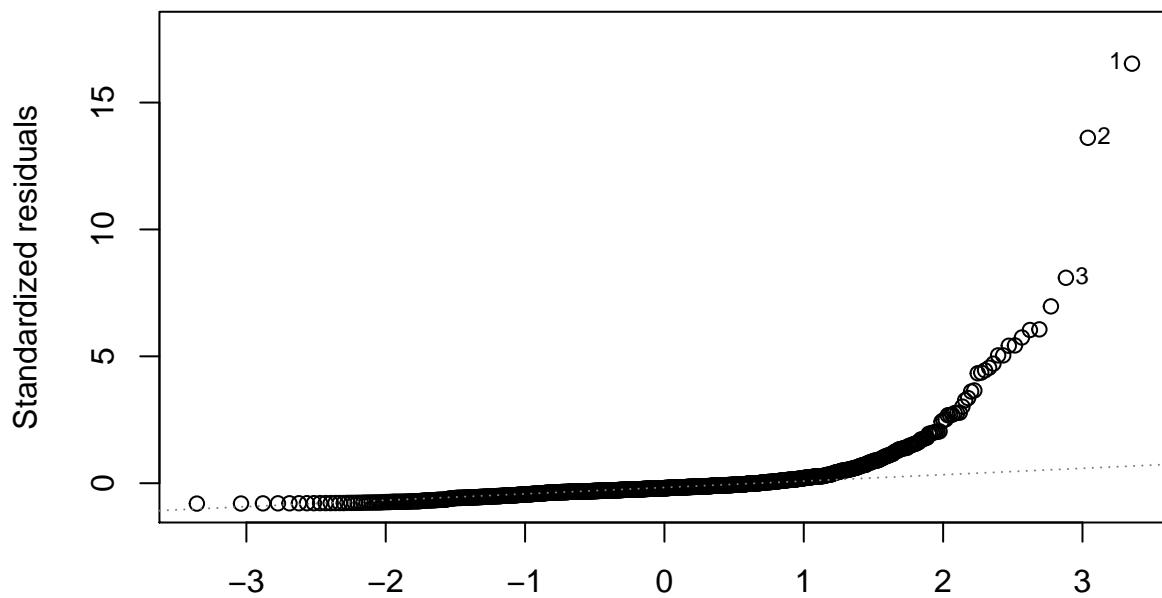
summary(m1)

##
## Call:
## lm(formula = APPEARANCES ~ active_years, data = dc_small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -142.08  -59.87  -34.17   0.34 2928.16 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 34.1321    10.2537   3.329 0.000898 ***
## active_years 1.7664     0.2494   7.084 2.33e-12 ***
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 177.4 on 1260 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0383, Adjusted R-squared:  0.03753 
## F-statistic: 50.18 on 1 and 1260 DF,  p-value: 2.331e-12
plot(m1)
```

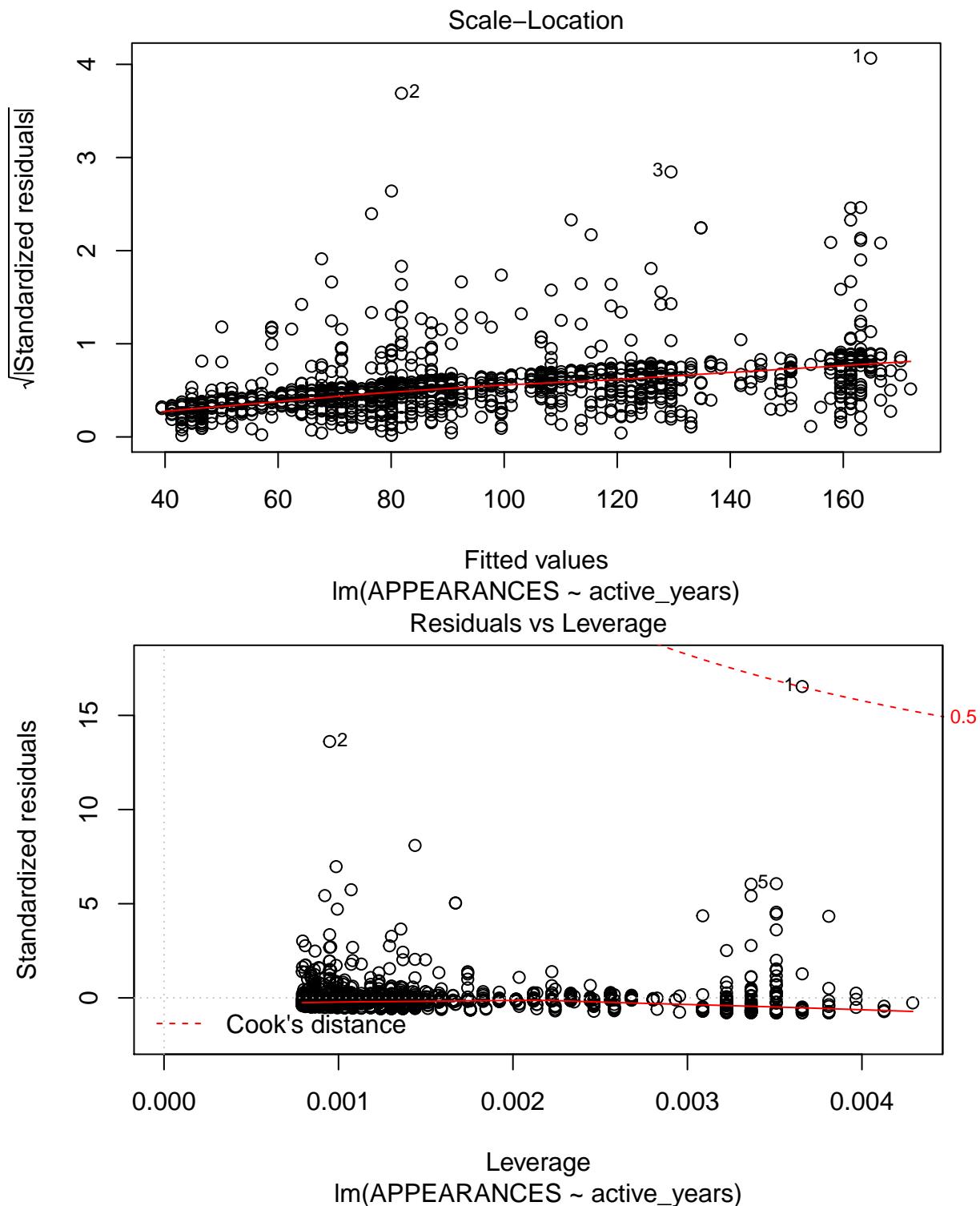
Residuals vs Fitted



Fitted values  
Im(APPEARANCES ~ active\_years)  
Normal Q-Q



Theoretical Quantiles  
Im(APPEARANCES ~ active\_years)



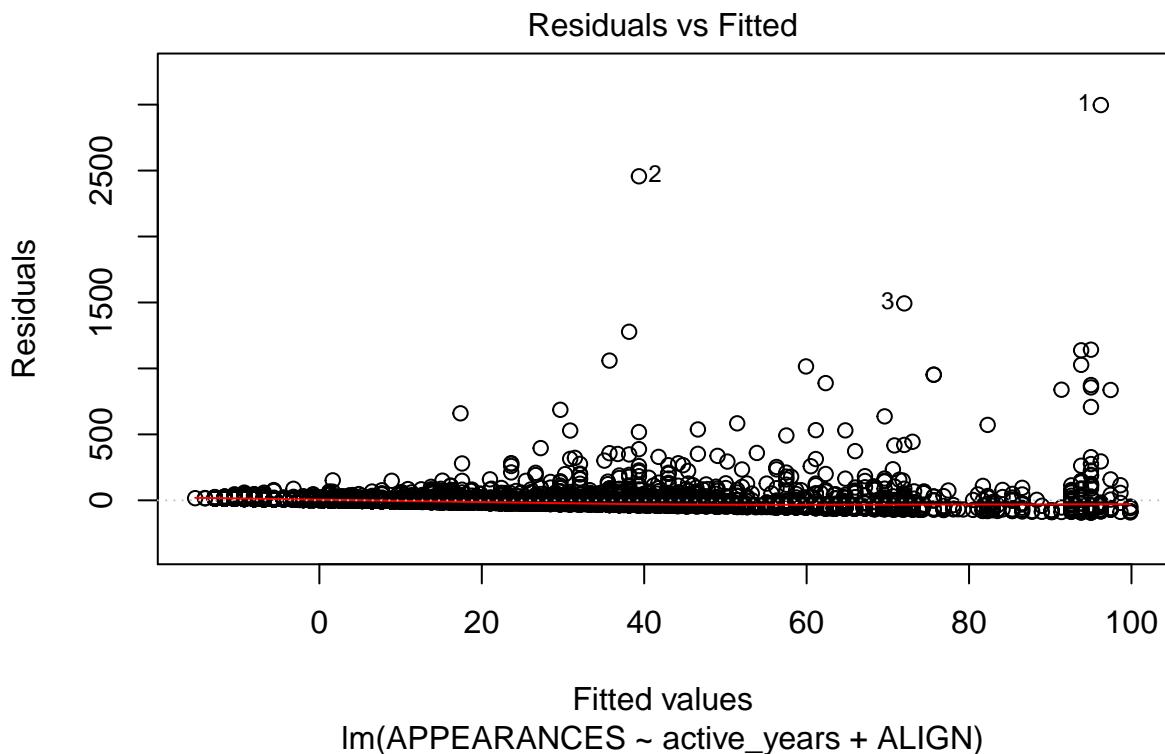
Multiple linear regression

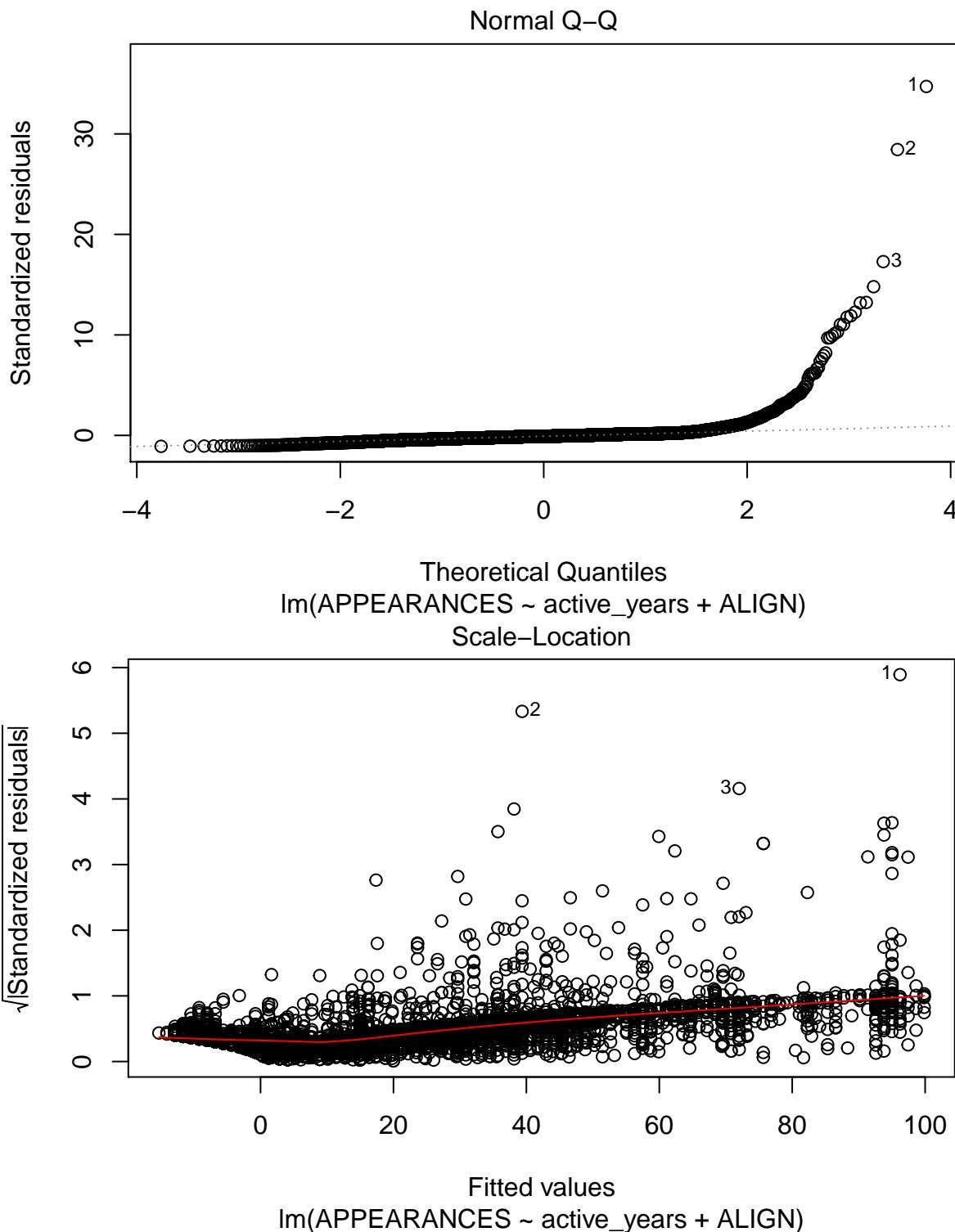
```
m3 <- lm(data=dc, APPEARANCES ~ active_years + ALIGN)
```

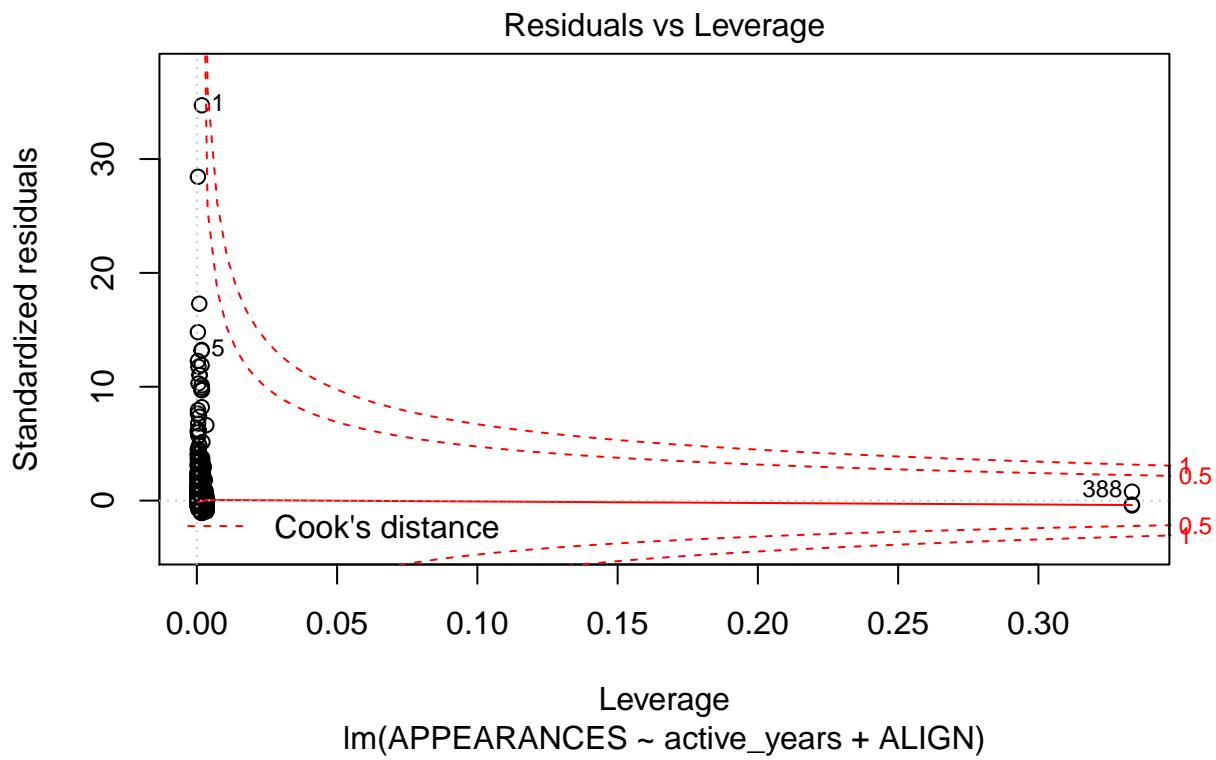
```
summary(m3)
```

```
##  
## Call:  
## lm(formula = APPEARANCES ~ active_years + ALIGN, data = dc)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -94.01  -22.47   -8.10    6.98 2996.78  
##  
## Coefficients:  
##                               Estimate Std. Error t value Pr(>|t|)  
## (Intercept)             -15.2929    2.2004 -6.950 4.05e-12 ***  
## active_years              1.2101    0.0675 17.927 < 2e-16 ***  
## ALIGNGood Characters     21.9737    2.3721  9.263 < 2e-16 ***  
## ALIGNNeutral Characters  10.4623    4.0650  2.574  0.0101 *  
## ALIGNReformed Criminals 27.2122   49.9045  0.545  0.5856  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 86.39 on 5918 degrees of freedom  
## (973 observations deleted due to missingness)  
## Multiple R-squared:  0.07101,    Adjusted R-squared:  0.07039  
## F-statistic: 113.1 on 4 and 5918 DF,  p-value: < 2.2e-16
```

```
plot(m3)
```







```
m4 <- lm(data=dc, APPEARANCES ~ active_years + ALIGN + active_years*ALIGN)
```

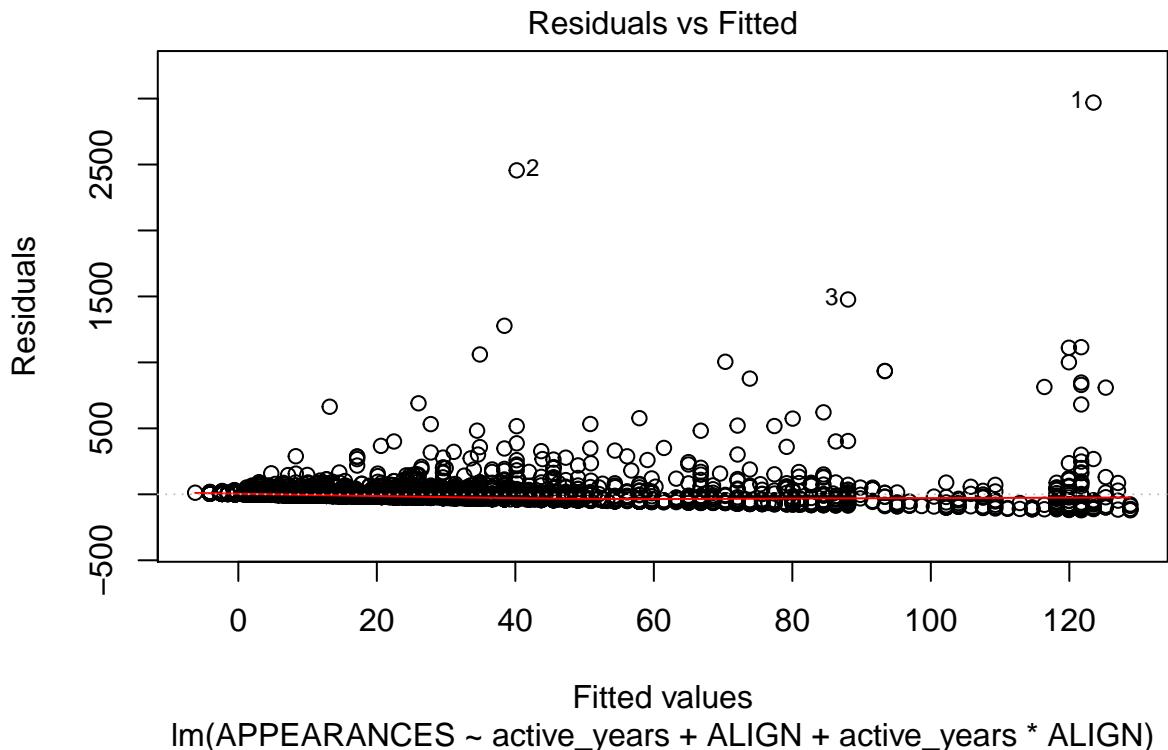
```
summary(m4)
```

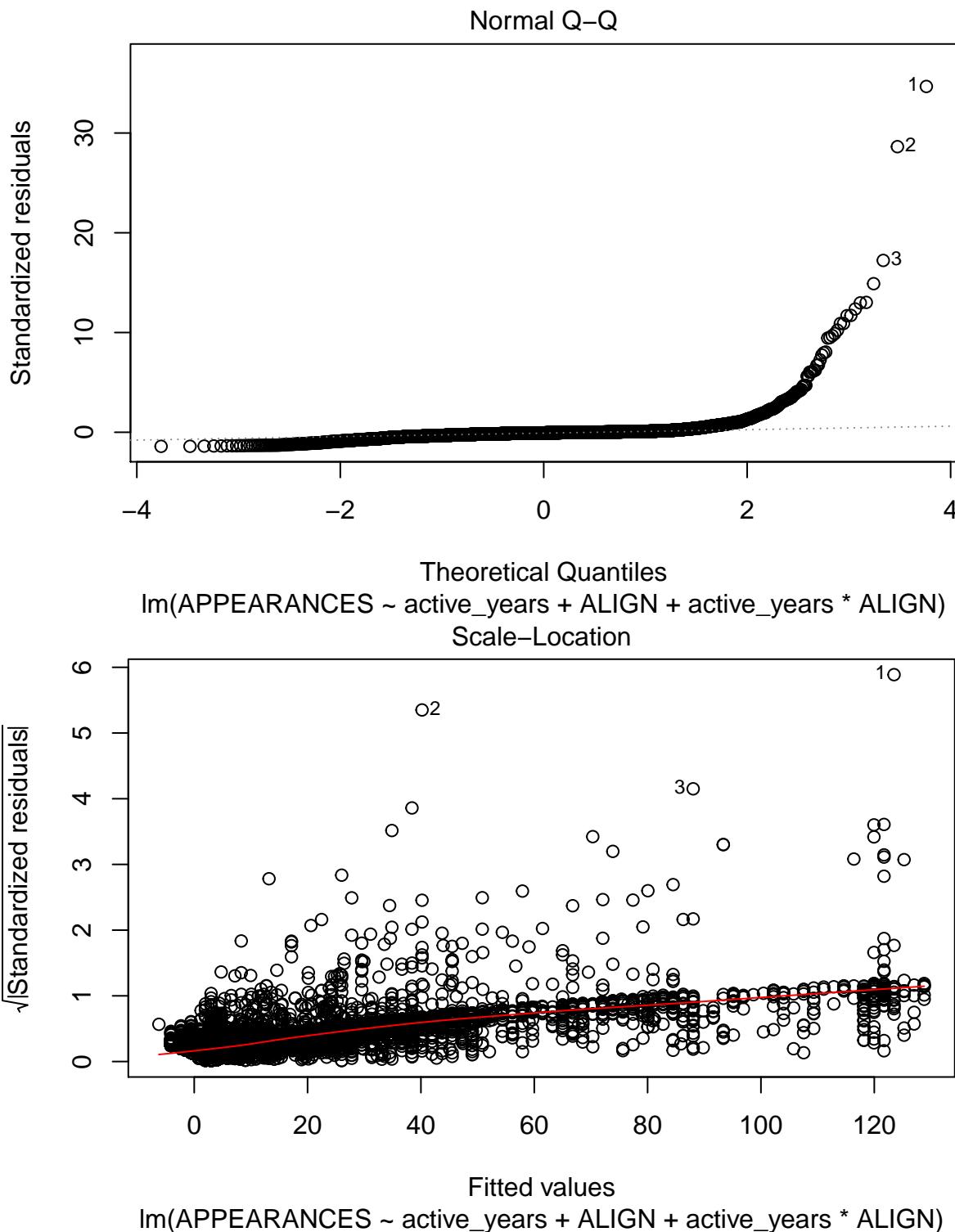
```
##  
## Call:  
## lm(formula = APPEARANCES ~ active_years + ALIGN + active_years *  
##       ALIGN, data = dc)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -120.80  -17.56   -5.56    2.56 2969.52  
##  
## Coefficients:  
##                               Estimate Std. Error t value Pr(>|t|)  
## (Intercept)                 0.7058    2.8269   0.250  0.80285  
## active_years                  0.4626    0.1073   4.311 1.65e-05  
## ALIGNGood Characters          -8.3553    4.0374  -2.069  0.03855  
## ALIGNNeutral Characters        -4.5558    6.8834  -0.662  0.50809  
## ALIGNReformed Criminals     127.6788   168.9829   0.756  0.44994  
## active_years:ALIGNGood Characters  1.3095    0.1416   9.247 < 2e-16  
## active_years:ALIGNNeutral Characters  0.7027    0.2554   2.751  0.00596  
## active_years:ALIGNReformed Criminals -7.1934   11.0144  -0.653  0.51373  
##  
## (Intercept)  
## active_years                   ***  
## ALIGNGood Characters             *  
## ALIGNNeutral Characters  
## ALIGNReformed Criminals  
## active_years:ALIGNGood Characters ***
```

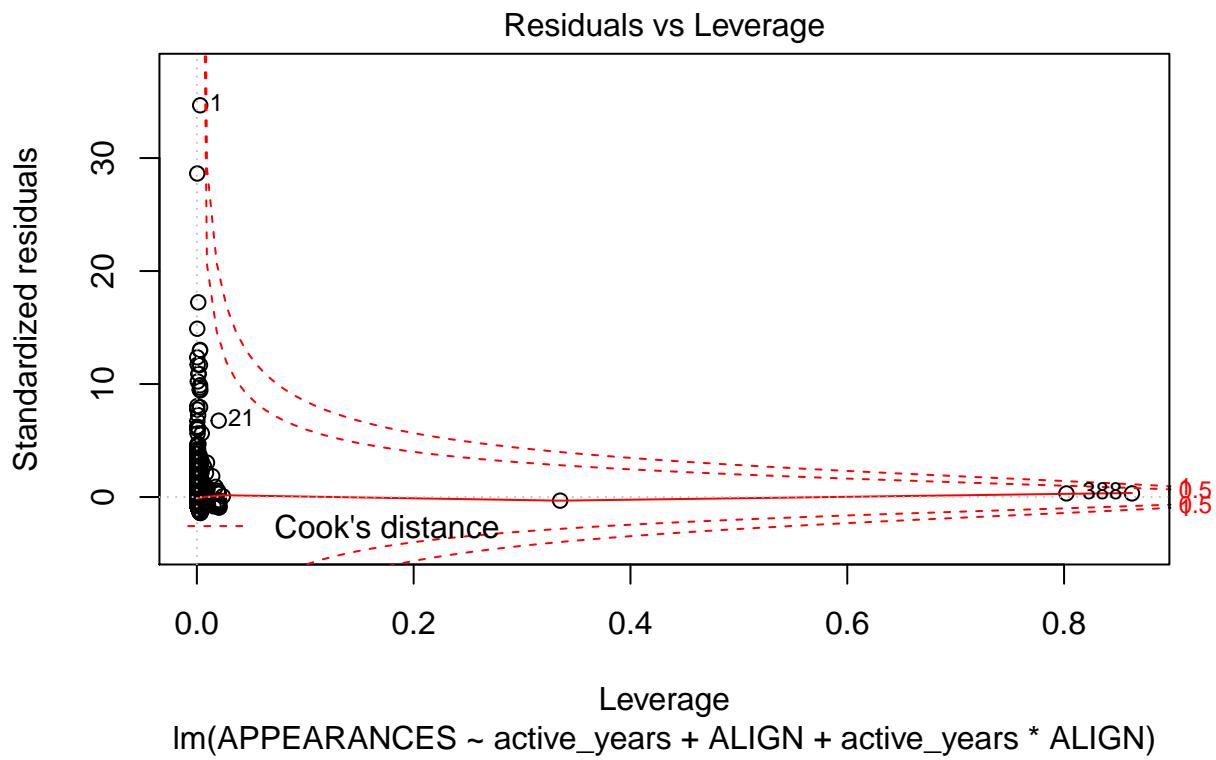
```

## active_years:ALIGNNeutral Characters **
## active_years:ALIGNReformed Criminals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 85.79 on 5915 degrees of freedom
##   (973 observations deleted due to missingness)
## Multiple R-squared:  0.08434,   Adjusted R-squared:  0.08325
## F-statistic: 77.83 on 7 and 5915 DF,  p-value: < 2.2e-16
plot(m4)

```



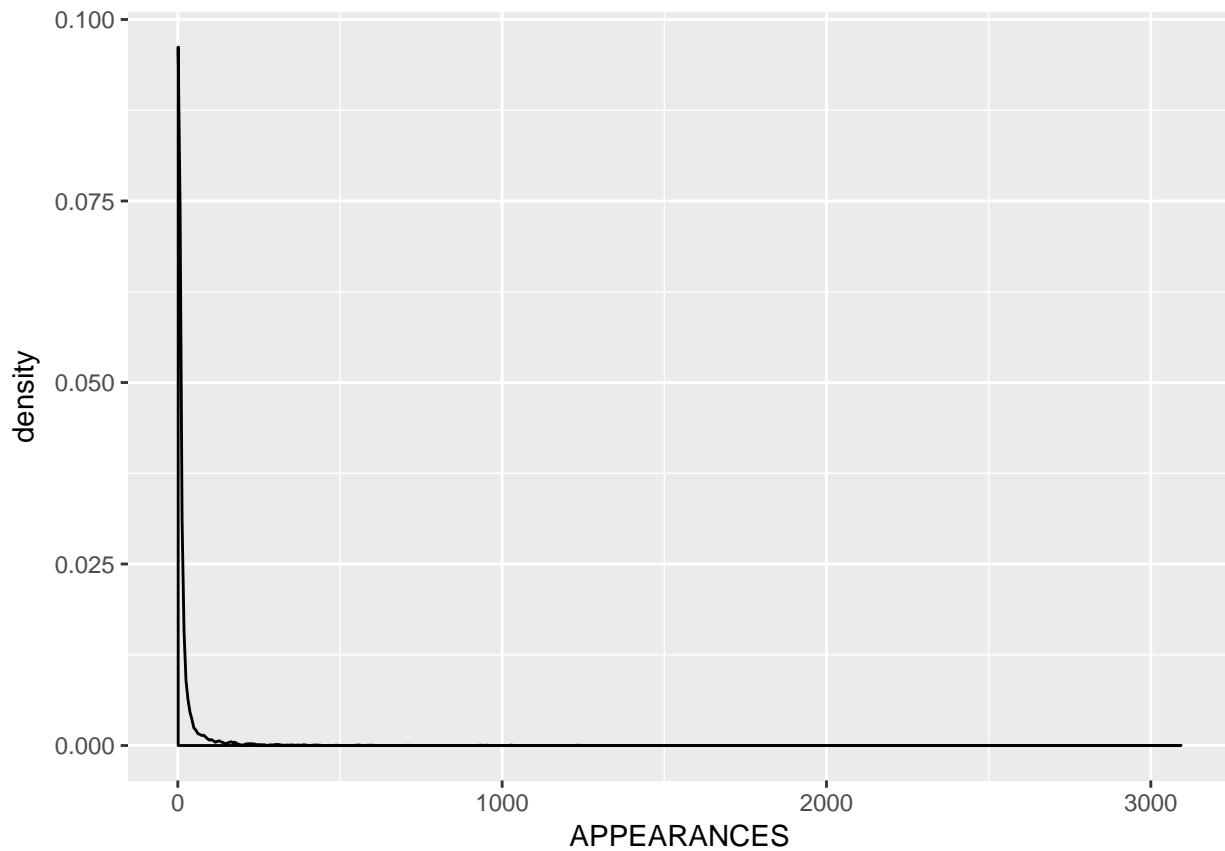




### Log-level regression

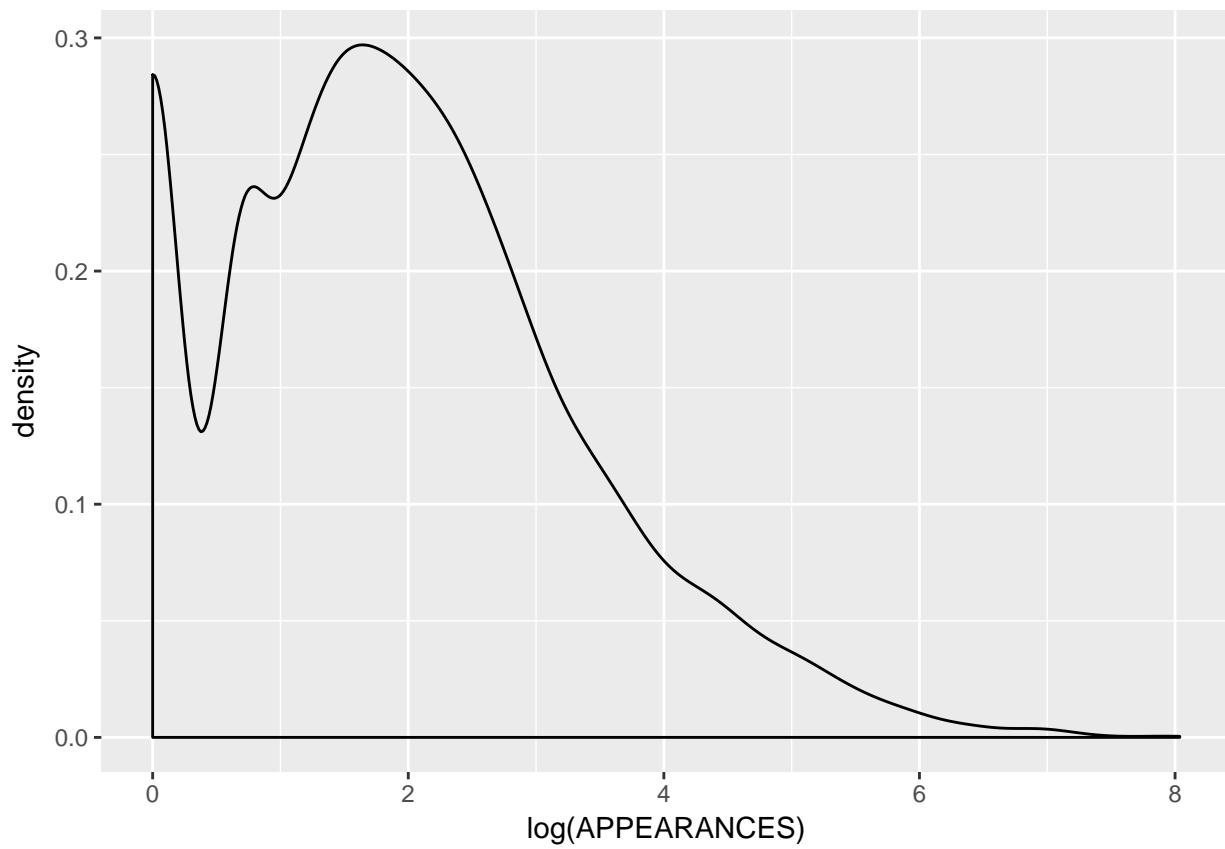
```
http://www.cazaar.com/ta/econ113/interpreting-beta
dc %>%
  ggplot(aes(x = APPEARANCES)) +
  geom_density()

## Warning: Removed 355 rows containing non-finite values (stat_density).
```



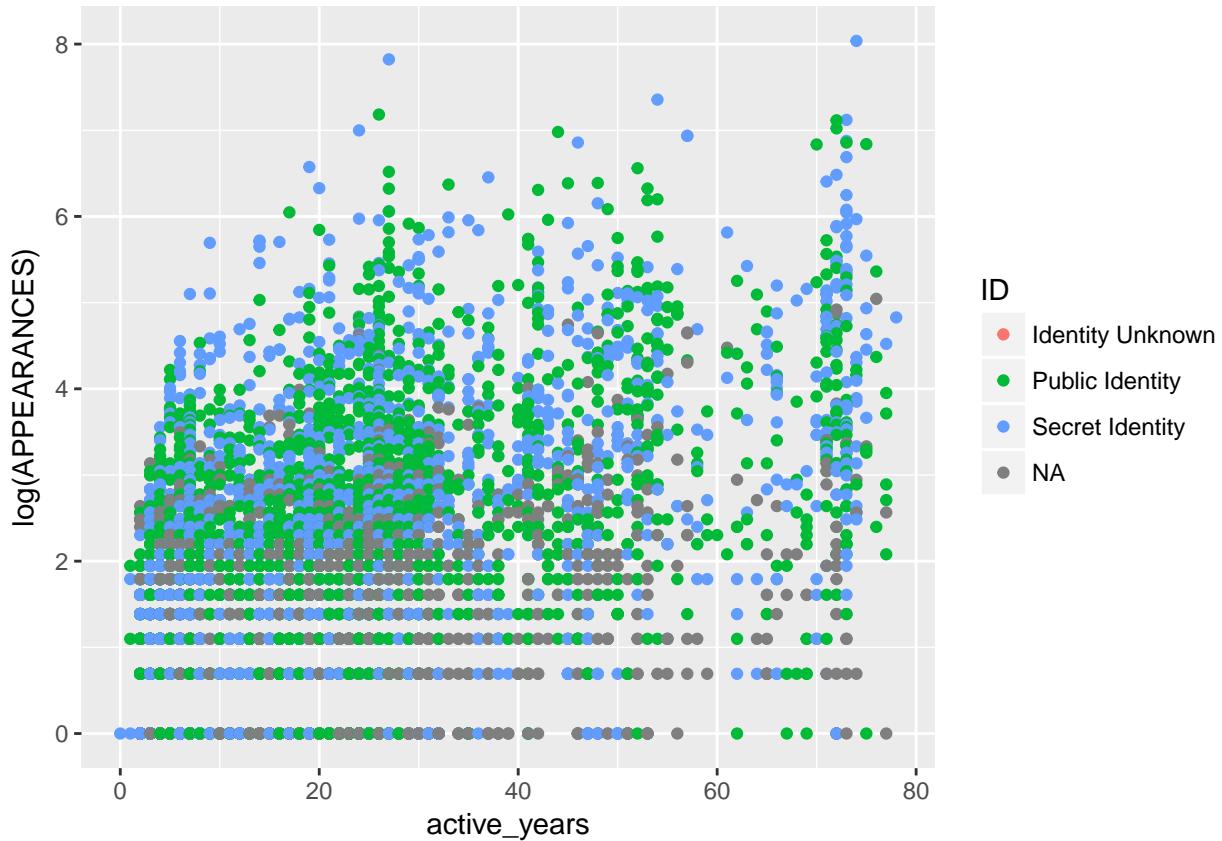
```
dc %>%
  ggplot(aes(x = log(APPEARANCES))) +
  geom_density()
```

```
## Warning: Removed 355 rows containing non-finite values (stat_density).
```



```
dc %>%
  ggplot(aes(x = active_years, y = log(APPEARANCES), col = ID)) +
  geom_point()
```

```
## Warning: Removed 415 rows containing missing values (geom_point).
```

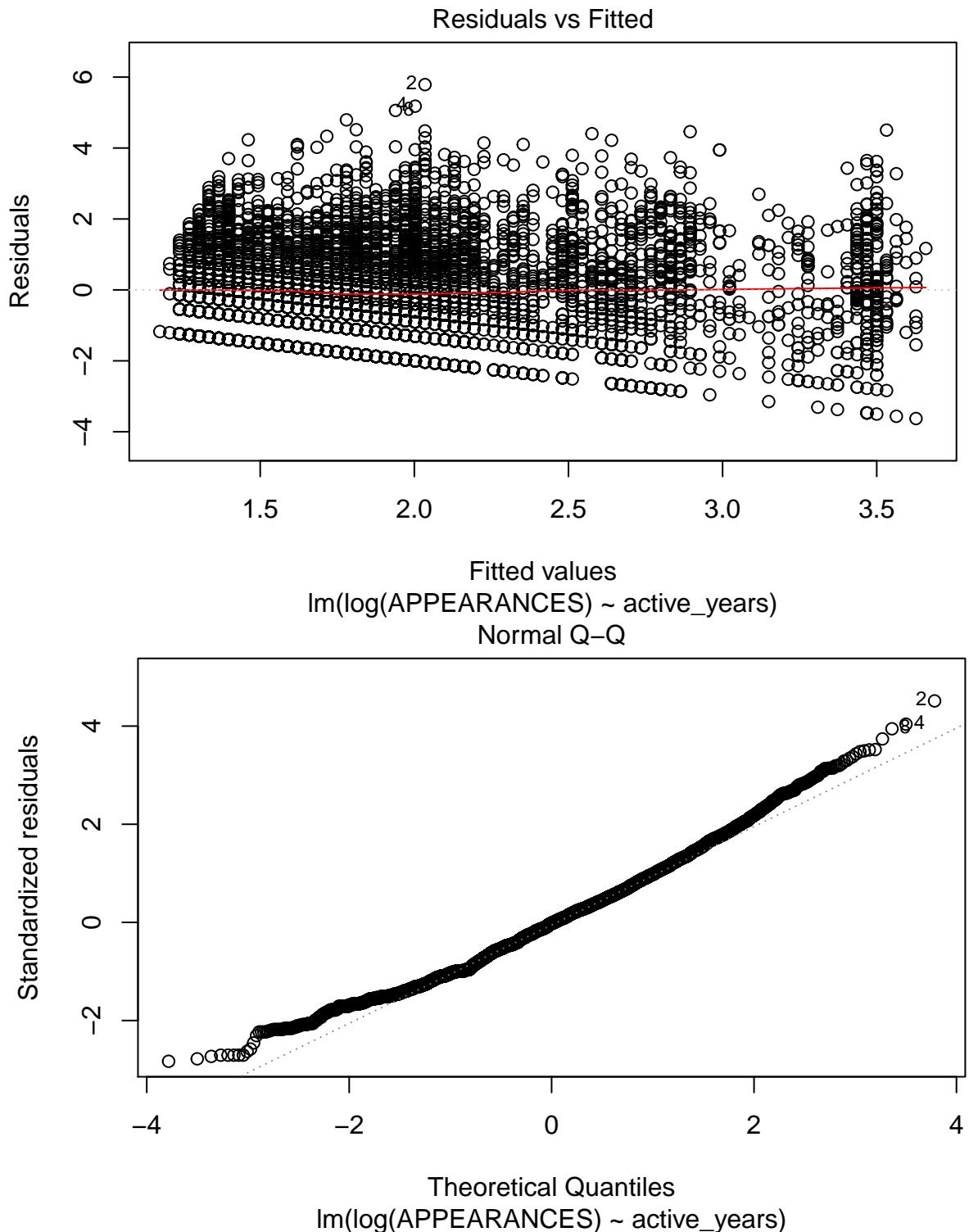


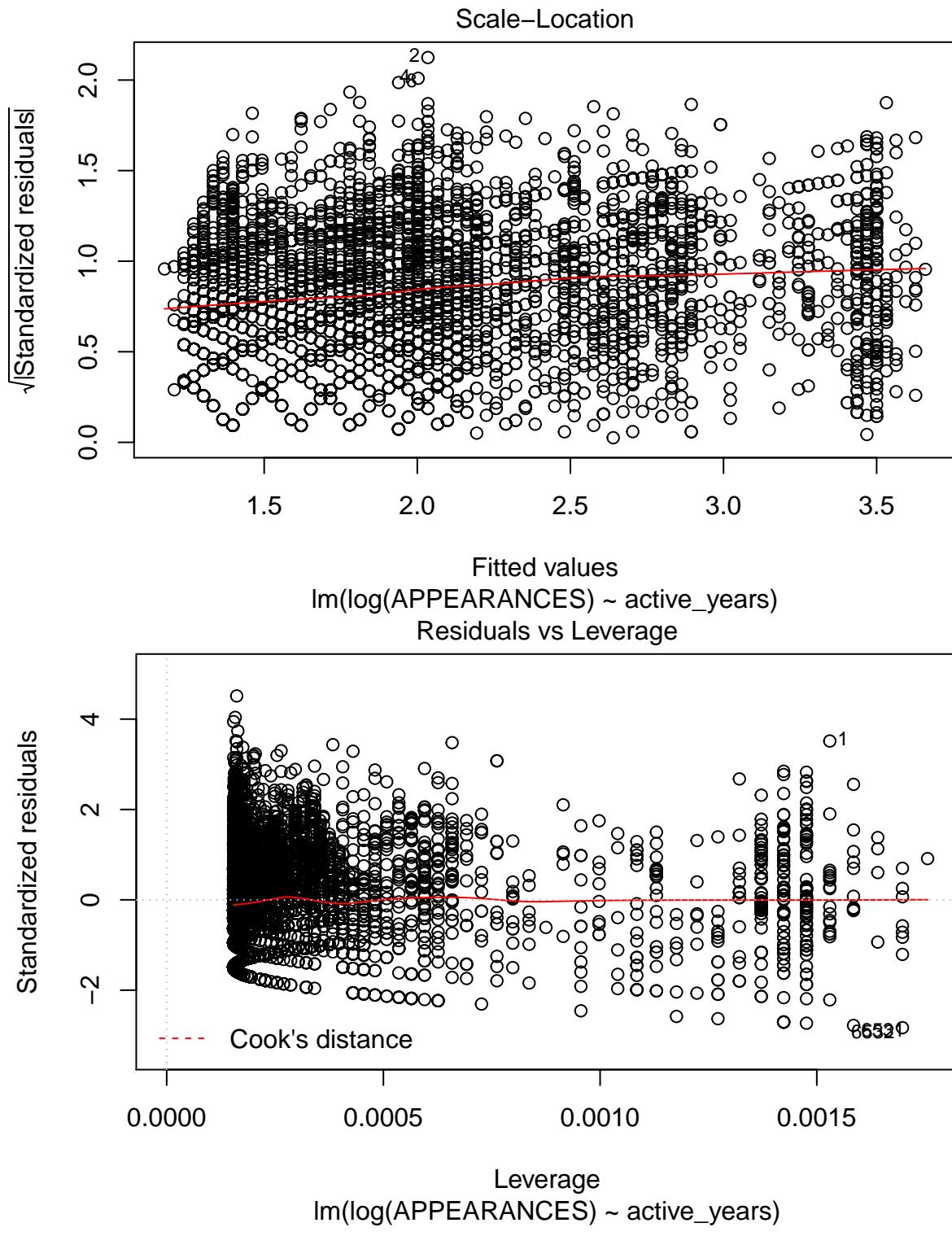
```
m2_1 <- lm(data=dc, log(APPEARANCES) ~ active_years)

summary(m2_1)

##
## Call:
## lm(formula = log(APPEARANCES) ~ active_years, data = dc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -3.6274 -0.9350 -0.0199  0.7997  5.7878 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.1745566  0.0270642  43.40 <2e-16 ***
## active_years 0.0318549  0.0009385  33.94 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.283 on 6479 degrees of freedom
##   (415 observations deleted due to missingness)
## Multiple R-squared:  0.151, Adjusted R-squared:  0.1508 
## F-statistic: 1152 on 1 and 6479 DF, p-value: < 2.2e-16

plot(m2_1)
```





Se aggiungiamo un altro anno di attività, ci aspettiamo che il numero di apparizioni cresca del 3%

```
m3 <- lm(data=dc, log(APPEARANCES) ~ active_years + ALIGN)
```

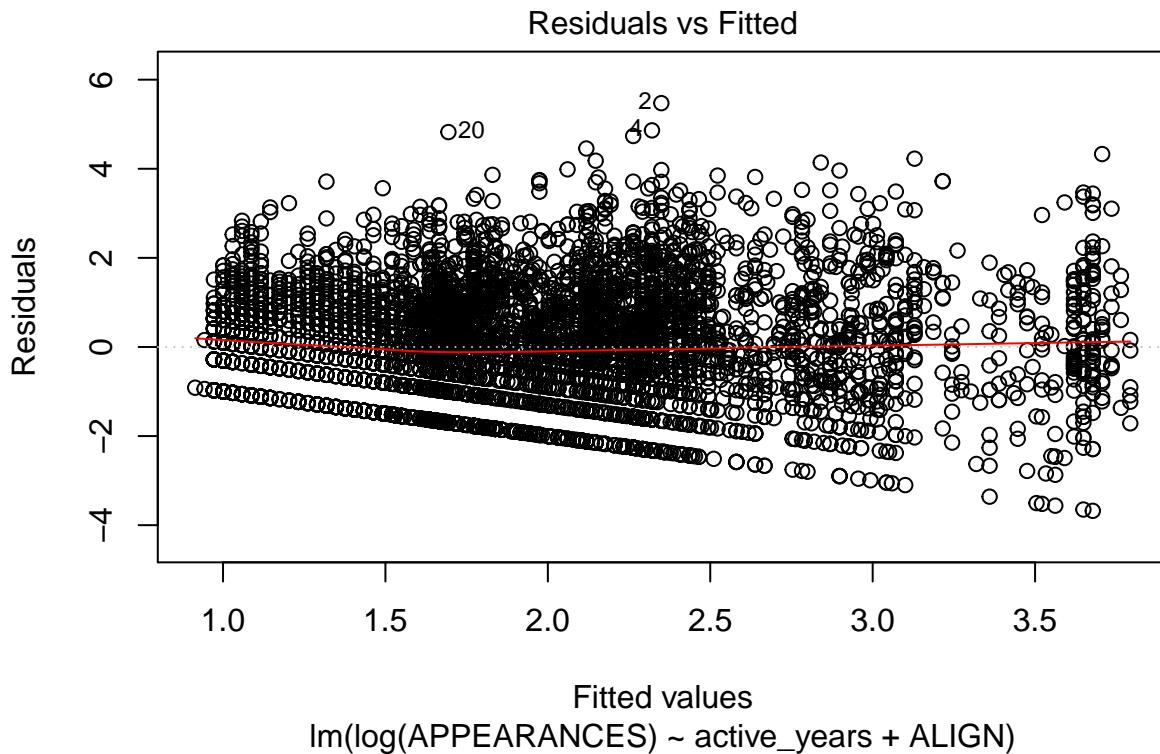
```
summary(m3)
```

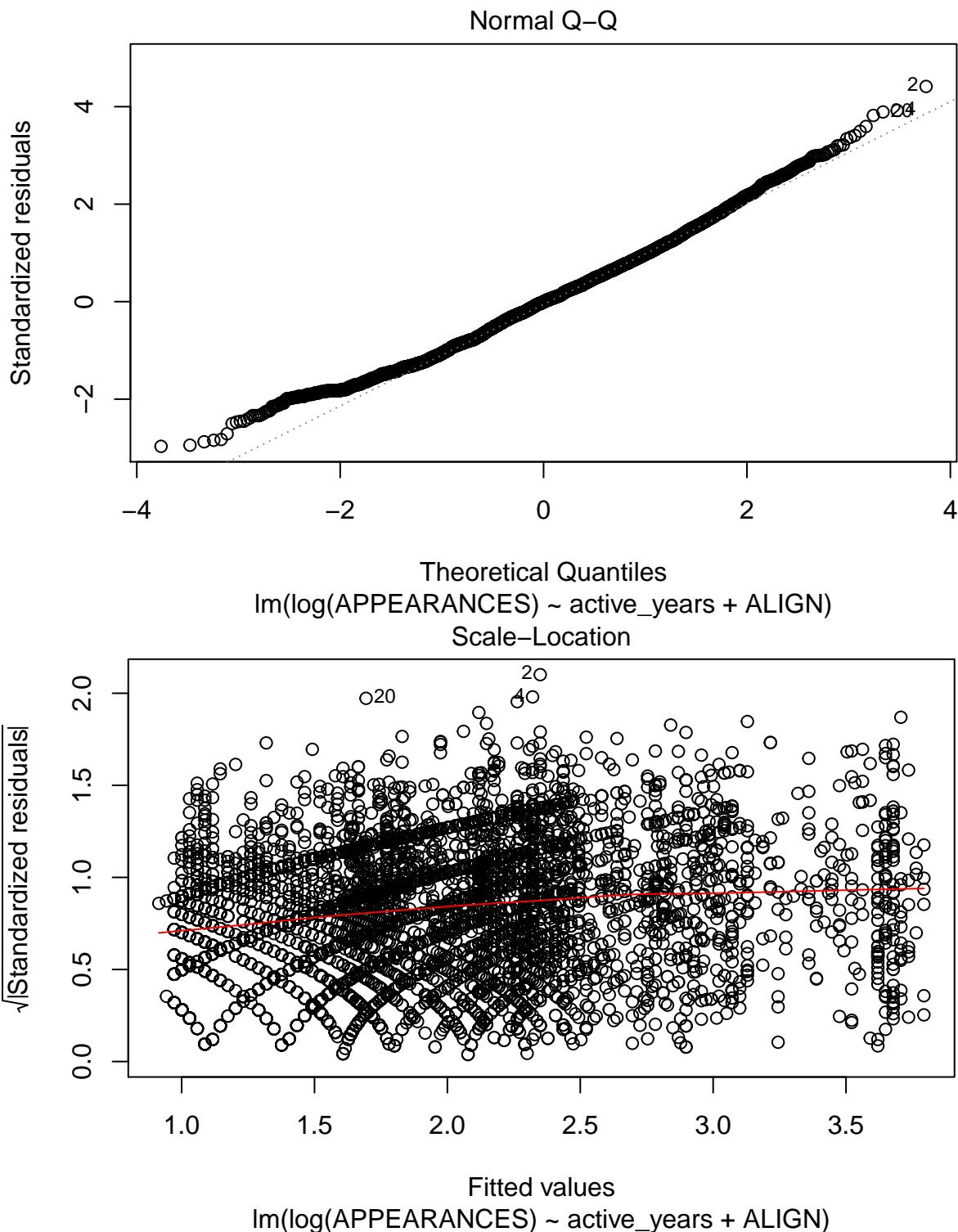
```
##
```

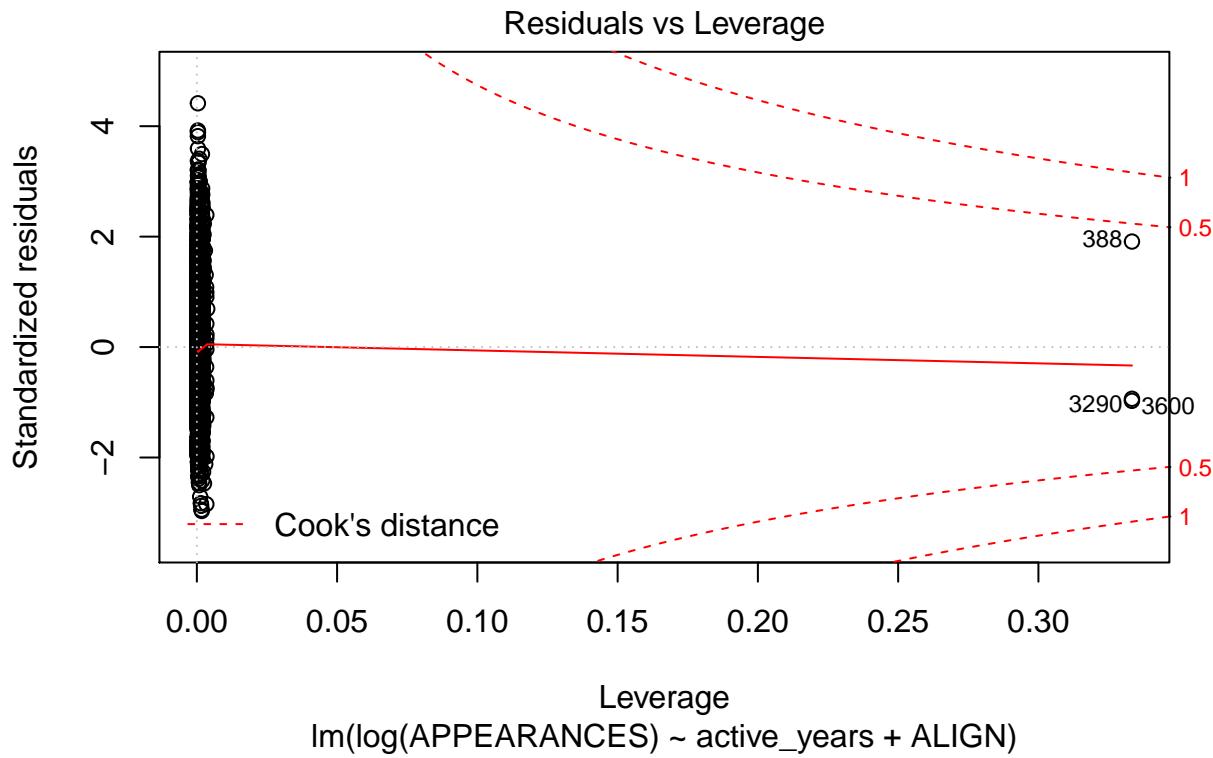
```

## Call:
## lm(formula = log(APPEARANCES) ~ active_years + ALIGN, data = dc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -3.6776 -0.9429 -0.0091  0.7971  5.4730 
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 0.9142744  0.0315861 28.945 <2e-16 ***
## active_years 0.0288729  0.0009689 29.800 <2e-16 *** 
## ALIGNGood Characters 0.6555994  0.0340508 19.254 <2e-16 *** 
## ALIGNNeutral Characters 0.5281561  0.0583511  9.051 <2e-16 *** 
## ALIGNReformed Criminals 1.2482256  0.7163524  1.742  0.0815 .  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.24 on 5918 degrees of freedom
## (973 observations deleted due to missingness)
## Multiple R-squared:  0.1936, Adjusted R-squared:  0.193 
## F-statistic: 355.1 on 4 and 5918 DF,  p-value: < 2.2e-16
plot(m3)

```







Se aggiungiamo un altro anno di attività, ci aspettiamo che il numero di apparizioni cresca del 3%. Se il personaggio è cattivo, però, ci aspettiamo un decremento del numero di apparizioni del 17%, mentre se è buono un incremento del 47%.

<https://www.youtube.com/watch?v=wXC2kViEGz8>

```
m4 <- lm(data=dc, log(APPEARANCES) ~ active_years + ALIGN + active_years*ALIGN)

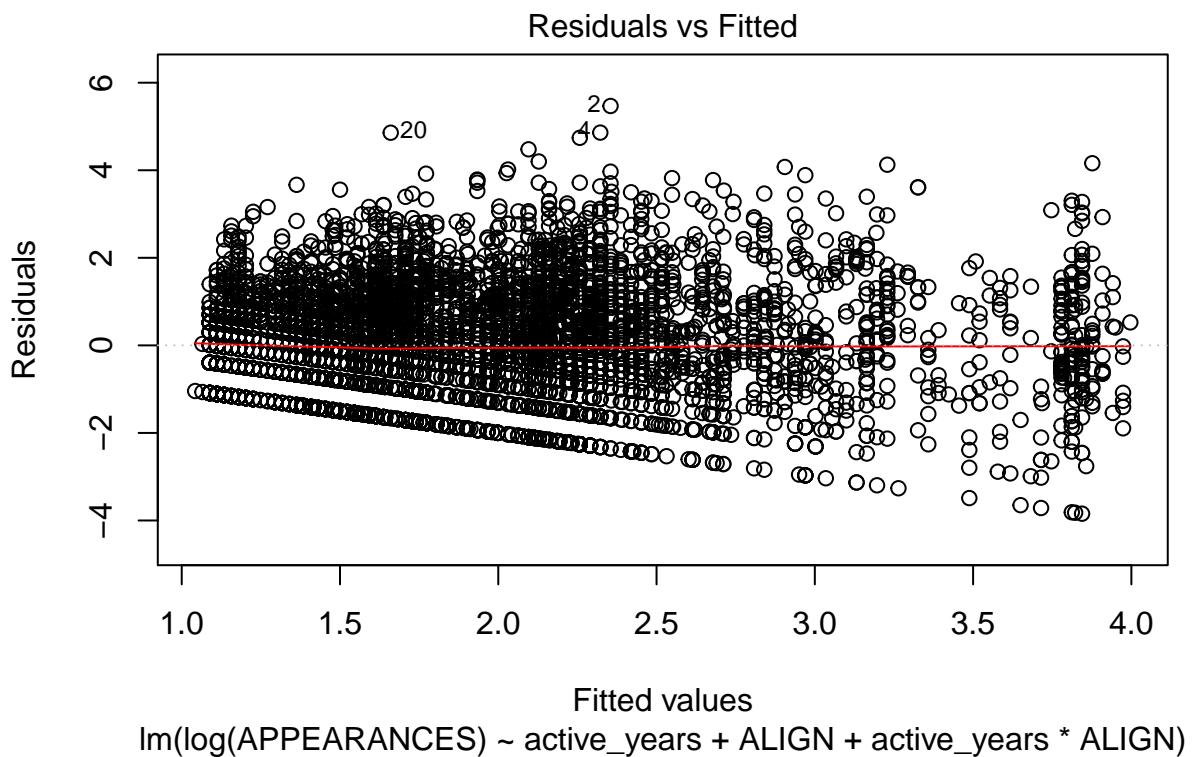
summary(m4)
```

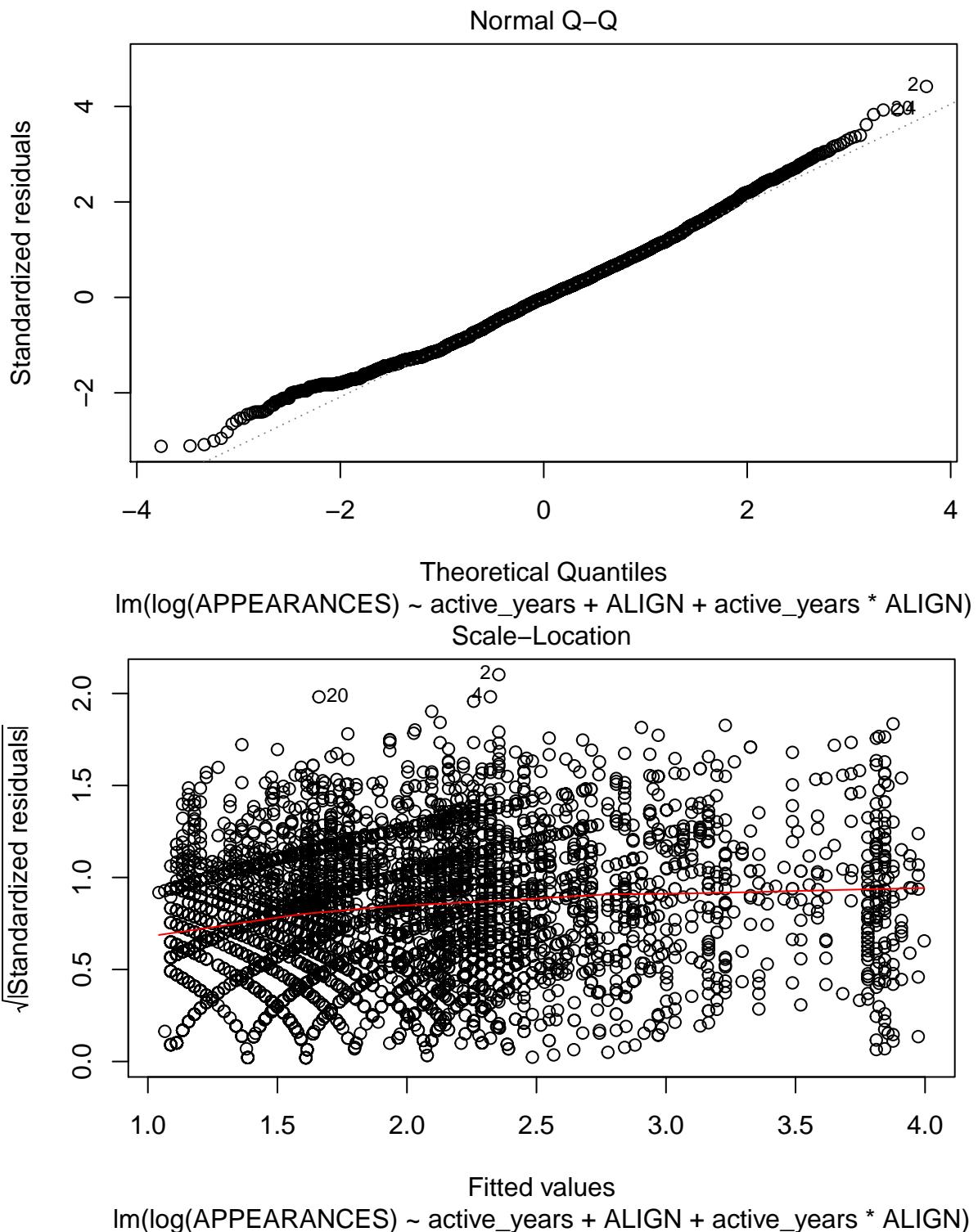
```
##
## Call:
## lm(formula = log(APPEARANCES) ~ active_years + ALIGN + active_years *
##     ALIGN, data = dc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -3.8442 -0.9037 -0.0126  0.8020  5.4677 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 1.042552  0.040777 25.567 < 2e-16  
## active_years                0.022880  0.001548 14.782 < 2e-16  
## ALIGNGood Characters        0.437866  0.058237  7.519 6.36e-14  
## ALIGNNeutral Characters    0.268121  0.099289  2.700  0.00694  
## ALIGNReformed Criminals   5.071996  2.437474  2.081  0.03749  
## active_years:ALIGNGood Characters 0.009501  0.002043  4.652 3.37e-06  
## active_years:ALIGNNeutral Characters 0.012005  0.003685  3.258  0.00113  
## active_years:ALIGNReformed Criminals -0.263465  0.158876 -1.658  0.09731 
## 
## (Intercept) ***
```

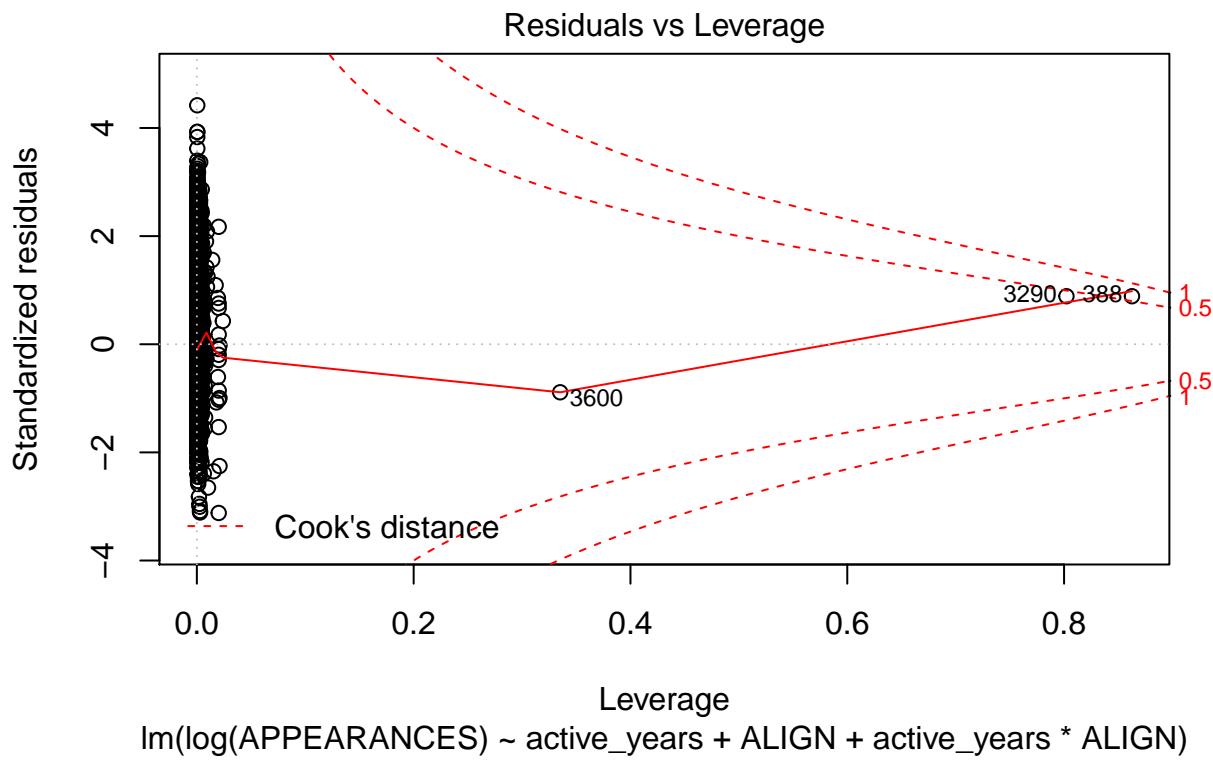
```

## active_years      ***
## ALIGNGood Characters ***
## ALIGNNeutral Characters **
## ALIGNReformed Criminals *
## active_years:ALIGNGood Characters ***
## active_years:ALIGNNeutral Characters **
## active_years:ALIGNReformed Criminals .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 5915 degrees of freedom
##   (973 observations deleted due to missingness)
## Multiple R-squared:  0.1974, Adjusted R-squared:  0.1964
## F-statistic: 207.8 on 7 and 5915 DF,  p-value: < 2.2e-16
plot(m4)

```







## LOGISTIC REGRESSION

<https://datascienceplus.com/perform-logistic-regression-in-r/>

```
dc_class <- dc %>%
  filter((ALIGN == "Bad Characters" | ALIGN == "Good Characters") &
         (SEX == "Female Characters" | SEX == "Male Characters")) %>%
  select(name, ALIGN, SEX, APPEARANCES, active_years) %>%
  na.omit

l1 <- glm(data = dc_class, ALIGN ~ SEX, family = "binomial")

summary(l1)

##
## Call:
## glm(formula = ALIGN ~ SEX, family = "binomial", data = dc_class)
##
## Deviance Residuals:
##    Min      1Q      Median      3Q      Max 
## -1.378  -1.104  -1.104   1.253   1.253 
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) 0.45981   0.05364   8.572   <2e-16 ***
## SEXMale Characters -0.63598   0.06273 -10.139   <2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

## 
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7316.9  on 5277  degrees of freedom
## Residual deviance: 7212.0  on 5276  degrees of freedom
## AIC: 7216
##
## Number of Fisher Scoring iterations: 4
post_l1 <- predict(l1, type = "response")

ALIGN_pred <- ifelse(post_l1>0.5, "Good Characters", "Bad Characters")

misClasificError <- mean(ALIGN_pred != dc_class$ALIGN)
print(paste('Accuracy', 1-misClasificError))

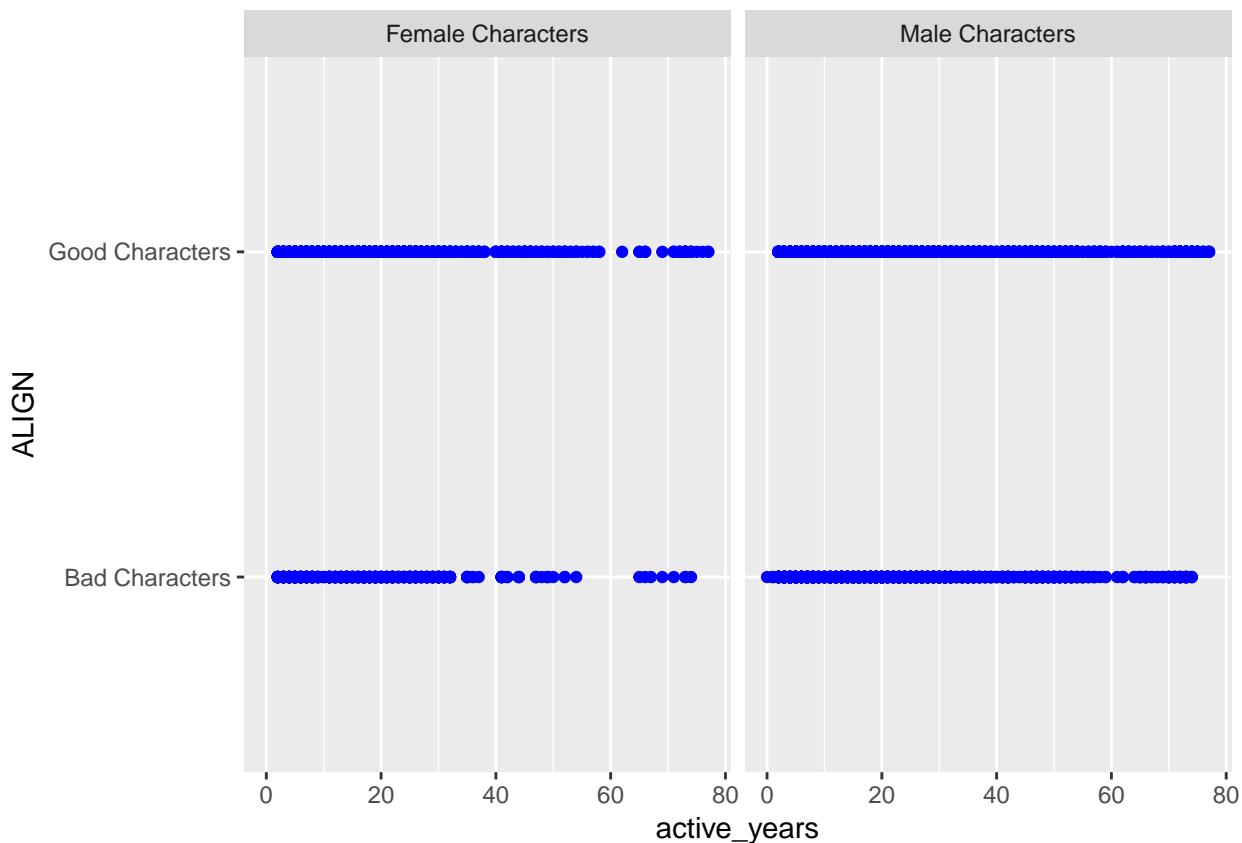
## [1] "Accuracy 0.170140204622963"
table(dc_class$ALIGN)

##
##      Bad Characters   Good Characters Neutral Characters
##             2641                  2637                   0
## Reformed Criminals
##                 0
table(ALIGN_pred)

## ALIGN_pred
## Bad Characters Good Characters
##       3813          1465
table(dc_class$ALIGN, ALIGN_pred)

##
##           ALIGN_pred
##           Bad Characters Good Characters
##   Bad Characters           2074          567
##   Good Characters          1739          898
##   Neutral Characters         0            0
##   Reformed Criminals        0            0
ggp <- ggplot(data = dc_class, mapping = aes(x = active_years, y = ALIGN)) +
  geom_point(colour="blue") +
  #geom_line(mapping = aes(x = active_years, y = ALIGN), colour="red") +
  facet_wrap(facets = ~SEX)
print(ggp)

```



```
12 <- glm(data = dc_class, ALIGN ~ SEX + active_years, family = "binomial")
summary(12)
```

```
##
## Call:
## glm(formula = ALIGN ~ SEX + active_years, family = "binomial",
##      data = dc_class)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -1.7874   -1.1054   -0.9442    1.1572    1.4225
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.122198  0.063490  1.925   0.0543 .
## SEXMale Characters -0.715899  0.063810 -11.219  <2e-16 ***
## active_years  0.016877  0.001713   9.851  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 7316.9 on 5277 degrees of freedom
## Residual deviance: 7111.5 on 5275 degrees of freedom
## AIC: 7117.5
##
```

```

## Number of Fisher Scoring iterations: 4
post_12 <- predict(l2, type = "response")

ALIGN_pred_12 <- ifelse(post_12>0.5, "Good Characters", "Bad Characters")

misClasificError <- mean(ALIGN_pred_12 != dc_class$ALIGN)
print(paste('Accuracy', 1-misClasificError))

## [1] "Accuracy 0.255778704054566"
table(dc_class$ALIGN, ALIGN_pred_12)

##          ALIGN_pred_12
##          Bad Characters Good Characters
##  Bad Characters        1723           918
##  Good Characters       1287          1350
##  Neutral Characters        0            0
##  Reformed Criminals       0            0

anova(l2, test="Chisq")

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: ALIGN
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
##  NULL             5277    7316.9
##  SEX              1   104.90     5276    7212.0 < 2.2e-16 ***
##  active_years     1   100.47     5275    7111.5 < 2.2e-16 ***
##  ---
##  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

13 <- glm(data = dc_class, ALIGN ~ SEX + active_years + APPEARANCES, family = "binomial")

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
summary(13)

##
## Call:
## glm(formula = ALIGN ~ SEX + active_years + APPEARANCES, family = "binomial",
##      data = dc_class)
##
## Deviance Residuals:
##      Min        1Q        Median         3Q        Max 
## -4.3715  -1.0378  -0.9422   1.1578   1.4308 
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) 0.068231  0.064948  1.051   0.293    
## SEXMale Characters -0.676379  0.064748 -10.446 < 2e-16 ***
## active_years  0.007528  0.001866   4.036 5.45e-05 ***

```

```

## APPEARANCES      0.014712   0.001346  10.929 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 7316.9 on 5277 degrees of freedom
## Residual deviance: 6888.4 on 5274 degrees of freedom
## AIC: 6896.4
##
## Number of Fisher Scoring iterations: 6
post_13 <- predict(13, type = "response")

ALIGN_pred_13 <- ifelse(post_13>0.5, "Good Characters", "Bad Characters")

misClasificError <- mean(ALIGN_pred_13 != dc_class$ALIGN)
print(paste('Accuracy', 1-misClasificError))

## [1] "Accuracy 0.257104964001516"
table(dc_class$ALIGN, ALIGN_pred_13)

##
##          ALIGN_pred_13
##          Bad Characters Good Characters
## Bad Characters           1862            779
## Good Characters          1280           1357
## Neutral Characters         0              0
## Reformed Criminals        0              0

anova(13, test="Chisq")

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: ALIGN
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL             5277    7316.9
## SEX              1    104.90    5276    7212.0 < 2.2e-16 ***
## active_years     1    100.47    5275    7111.5 < 2.2e-16 ***
## APPEARANCES     1    223.06    5274    6888.4 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

## DECISION TREE

<https://gormanalysis.com/decision-trees-in-r-using-rpart/>

```
library(rpart)
```

```



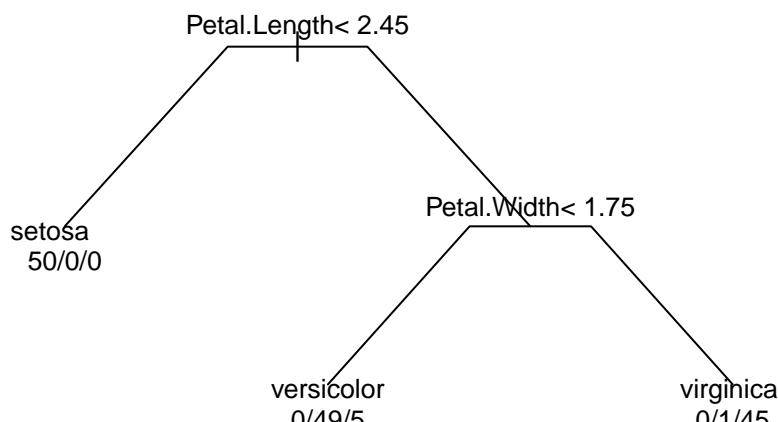
```

```

## 
## Node number 3: 100 observations,    complexity param=0.44
##   predicted class=versicolor  expected loss=0.5  P(node) =0.6666667
##   class counts:      0     50     50
##   probabilities: 0.000 0.500 0.500
##   left son=6 (54 obs) right son=7 (46 obs)
## Primary splits:
##   Petal.Width < 1.75 to the left,  improve=38.969400, (0 missing)
##   Petal.Length < 4.75 to the left,  improve=37.353540, (0 missing)
##   Sepal.Length < 6.15 to the left,  improve=10.686870, (0 missing)
##   Sepal.Width < 2.45 to the left,  improve= 3.555556, (0 missing)
## Surrogate splits:
##   Petal.Length < 4.75 to the left,  agree=0.91, adj=0.804, (0 split)
##   Sepal.Length < 6.15 to the left,  agree=0.73, adj=0.413, (0 split)
##   Sepal.Width < 2.95 to the left,  agree=0.67, adj=0.283, (0 split)
##
## Node number 6: 54 observations
##   predicted class=versicolor  expected loss=0.09259259  P(node) =0.36
##   class counts:      0     49      5
##   probabilities: 0.000 0.907 0.093
##
## Node number 7: 46 observations
##   predicted class=virginica   expected loss=0.02173913  P(node) =0.3066667
##   class counts:      0      1     45
##   probabilities: 0.000 0.022 0.978

plot(iris_tree, compress = T, margin = 0.2, branch = 0.3)
text(iris_tree, use.n = T, digits = 3, cex = 0.8)

```



```

printcp(iris_tree)

##
## Classification tree:
## rpart(formula = Species ~ ., data = iris, method = "class")
##
## Variables actually used in tree construction:
## [1] Petal.Length Petal.Width
##
## Root node error: 100/150 = 0.666667
##
## n= 150

```

```

## CP nsplit rel error xerror      xstd
## 1 0.50      0     1.00  1.14 0.052307
## 2 0.44      1     0.50  0.66 0.060795
## 3 0.01      2     0.06  0.07 0.025833
iris_pred <- predict(iris_tree, type = "class")

table(iris_pred, iris$Species)

##
## iris_pred      setosa versicolor virginica
##   setosa        50         0         0
##   versicolor     0        49         5
##   virginica      0         1        45
misClasificError <- mean(iris_pred != iris$Species)
print(paste('Accuracy', 1-misClasificError))

## [1] "Accuracy 0.96"
align_tree <- rpart(ALIGN ~ SEX + active_years + APPEARANCES, method = "class", data = dc_class)

summary(align_tree)

## Call:
## rpart(formula = ALIGN ~ SEX + active_years + APPEARANCES, data = dc_class,
##       method = "class")
## n= 5278
##
##          CP nsplit rel error xerror      xstd
## 1 0.2396663      0 1.0000000 1.0383011 0.01376540
## 2 0.0242700      1 0.7603337 0.7675389 0.01339582
## 3 0.0100000      2 0.7360637 0.7425104 0.01330854
##
## Variable importance
## APPEARANCES           SEX active_years
##            75             16            9
##
## Node number 1: 5278 observations,    complexity param=0.2396663
##   predicted class=Bad Characters  expected loss=0.4996211  P(node) =1
##   class counts: 2641 2637
##   probabilities: 0.500 0.500
##   left son=2 (3608 obs) right son=3 (1670 obs)
## Primary splits:
##   APPEARANCES < 11.5 to the left,  improve=175.64190, (0 missing)
##   SEX           splits as R-L-,    improve= 52.10742, (0 missing)
##   active_years < 22.5 to the left,  improve= 35.48814, (0 missing)
## Surrogate splits:
##   active_years < 39.5 to the left,  agree=0.722, adj=0.12, (0 split)
##
## Node number 2: 3608 observations,    complexity param=0.02427
##   predicted class=Bad Characters  expected loss=0.4118625  P(node) =0.6835923
##   class counts: 2122 1486
##   probabilities: 0.588 0.412
##   left son=4 (2640 obs) right son=5 (968 obs)

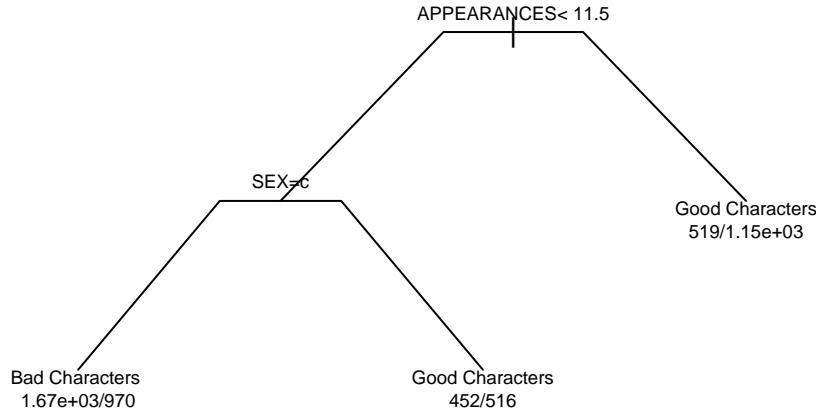
```

```

## Primary splits:
##   SEX           splits as R-L-,    improve=38.86330, (0 missing)
##   APPEARANCES < 4.5 to the left, improve=20.28753, (0 missing)
##   active_years < 22.5 to the left, improve=14.12867, (0 missing)
##
## Node number 3: 1670 observations
##   predicted class=Good Characters  expected loss=0.3107784 P(node) =0.3164077
##   class counts: 519 1151
##   probabilities: 0.311 0.689
##
## Node number 4: 2640 observations
##   predicted class=Bad Characters  expected loss=0.3674242 P(node) =0.5001895
##   class counts: 1670 970
##   probabilities: 0.633 0.367
##
## Node number 5: 968 observations
##   predicted class=Good Characters  expected loss=0.4669421 P(node) =0.1834028
##   class counts: 452 516
##   probabilities: 0.467 0.533

plot(align_tree, uniform = T, compress = T, margin = 0.2, branch = 0.3)
text(align_tree, use.n = T, digits = 3, cex = 0.6)

```



```
align_pred_tree <- predict(align_tree, type = "class")
```

```
table(align_pred_tree, dc_class$ALIGN)
```

```

##
## align_pred_tree      Bad Characters Good Characters Neutral Characters
##   Bad Characters          1670            970              0
##   Good Characters          971            1667              0
##   Neutral Characters         0              0              0
##   Reformed Criminals        0              0              0
##
## align_pred_tree      Reformed Criminals
##   Bad Characters             0
##   Good Characters             0
##   Neutral Characters             0
##   Reformed Criminals             0

```

```
misClasificError <- mean(align_pred_tree != dc_class$ALIGN)
print(paste('Accuracy', 1-misClasificError))
```

```
## [1] "Accuracy 0.632247063281546"
```

## Training and test dataset

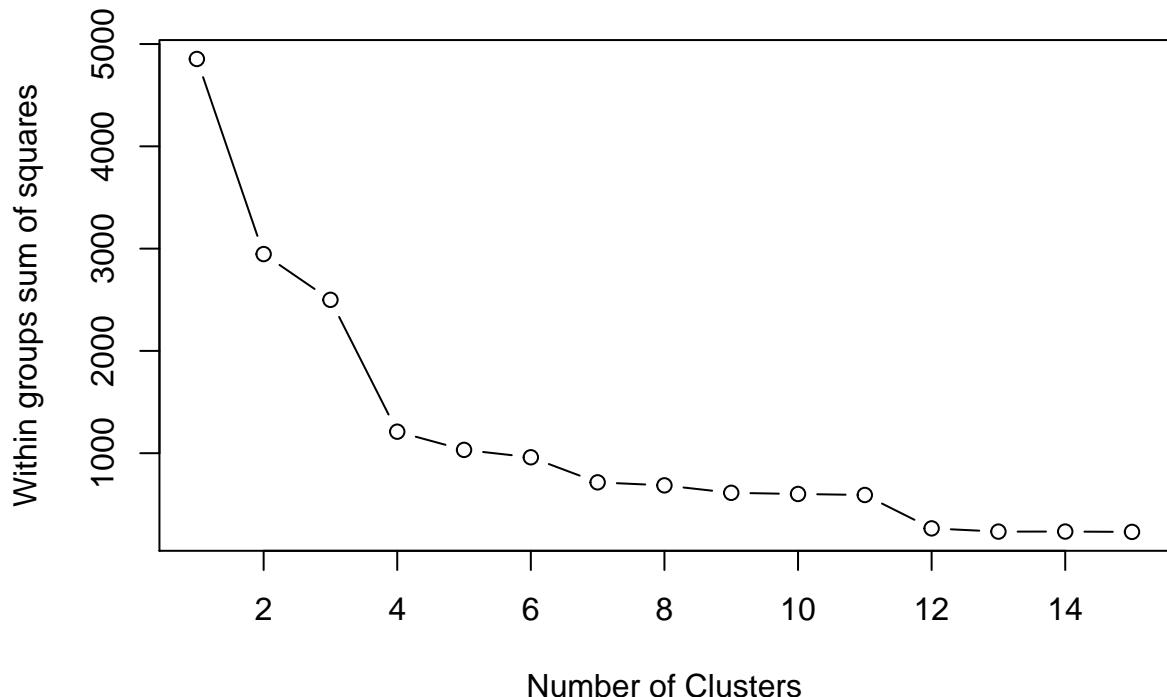
```
train <- sample(nrow(dc_class), 4800)
dc_train <- dc_class[train,]
dc_test <- dc_class[-train,]
```

## CLUSTER

```
dc_full <- dc %>%
  select(name, ALIGN, EYE, HAIR, SEX, ALIVE, APPEARANCES, active_years, YEAR) %>%
  na.omit()

dc_cluster <-dc_full %>%
  select(active_years, APPEARANCES) %>%
  scale() %>%
  as.data.frame()

# Finding cluster number through Within groups sum of squares
wss <- (nrow(dc_cluster)-1)*sum(apply(dc_cluster,2,var))
for (i in 2:15){
  wss[i] <- sum(kmeans(dc_cluster, centers=i)$withinss)
}
plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")
```



```

# Let's try 4

set.seed(1234)

# K-Means Cluster Analysis
fit <- kmeans(dc_cluster, 4) # 4 cluster solution
# append cluster assignment
dc_full <- data.frame(dc_full, cluster = as.factor(fit$cluster))

# CLUSTERS MEANS

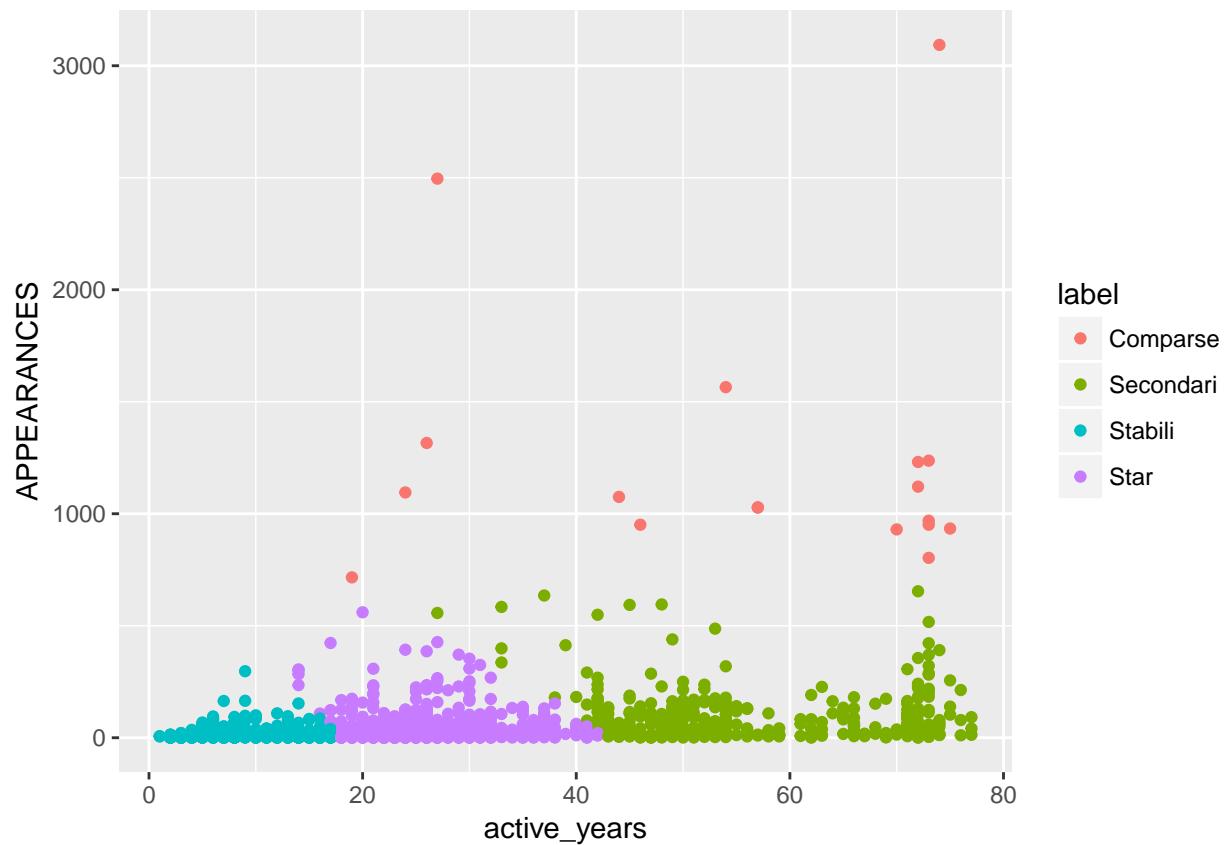
# Stats
cluster_stats <- dc_full %>%
  group_by(cluster) %>%
  summarise(count = n(),
            perc = paste0(round(n()/nrow(dc_full)*100,2), "%"),
            avg_appear = mean(APPEARANCES),
            avg_year = mean(active_years))

### LABELS MAY BE DIFFERENT!!

dc_full <- dc_full %>%
  mutate(label = case_when(
    cluster == 1 ~ "Stabili",
    cluster == 2 ~ "Star",
    cluster == 3 ~ "Comparse",
    cluster == 4 ~ "Secondari"
  ))

ggplot(data = dc_full, aes(x = active_years, y = APPEARANCES, col = label)) +
  geom_point()

```



```
star <- dc_full %>%
  filter(label == "Star")
```