Regression wokflow

Bhaskar

01/02/2021

https://stats.stackexchange.com/questions/100214/assumptions-of-linear-models-and-what-to-do-if-the-residuals-are-not-normally-di
https://stats.stackexchange.com/questions/173621/linear-regression-any-non-normal-distribution-giving-identity-of-ols-and-mle
https://stats.stackexchange.com/questions/152674/why-is-the-normality-of-residuals-barely-important-at-all-for-the-purpose-of-e

```
library(car)
```

```
## Loading required package: carData
```

```
df <- data.frame(Prestige)
head(df)
```

```
##                    education income women prestige census type
## gov.administrators      13.11  12351 11.16     68.8   1113 prof
## general.managers        12.26  25879  4.02     69.1   1130 prof
## accountants             12.77   9271 15.70     63.4   1171 prof
## purchasing.officers     11.42   8865  9.11     56.8   1175 prof
## chemists                14.62   8403 11.68     73.5   2111 prof
## physicists              15.64  11030  5.13     77.6   2113 prof
```

```
summary(df)
```

```
##    education          income          women          prestige
##  Min.   : 6.380   Min.   :  611   Min.   : 0.000   Min.   :14.80
##  1st Qu.: 8.445   1st Qu.: 4106   1st Qu.: 3.592   1st Qu.:35.23
##  Median :10.540   Median : 5930   Median :13.600   Median :43.60
##  Mean   :10.738   Mean   : 6798   Mean   :28.979   Mean   :46.83
##  3rd Qu.:12.648   3rd Qu.: 8187   3rd Qu.:52.203   3rd Qu.:59.27
##  Max.   :15.970   Max.   :25879   Max.   :97.510   Max.   :87.20
##      census       type
##  Min.   :1113   bc  :44
##  1st Qu.:3120   prof:31
##  Median :5135   wc  :23
##  Mean   :5402   NA's: 4
##  3rd Qu.:8312
##  Max.   :9517
```

```r
str(df)
```

```
## 'data.frame':    102 obs. of  6 variables:
##  $ education: num  13.1 12.3 12.8 11.4 14.6 ...
##  $ income   : int  12351 25879 9271 8865 8403 11030 8258 14163 11377 11023 ...
##  $ women    : num  11.16 4.02 15.7 9.11 11.68 ...
##  $ prestige : num  68.8 69.1 63.4 56.8 73.5 77.6 72.6 78.1 73.1 68.8 ...
##  $ census   : int  1113 1130 1171 1175 2111 2113 2133 2141 2143 2153 ...
##  $ type     : Factor w/ 3 levels "bc","prof","wc": 2 2 2 2 2 2 2 2 2 2 ...
```

```r
rownames(df)
```

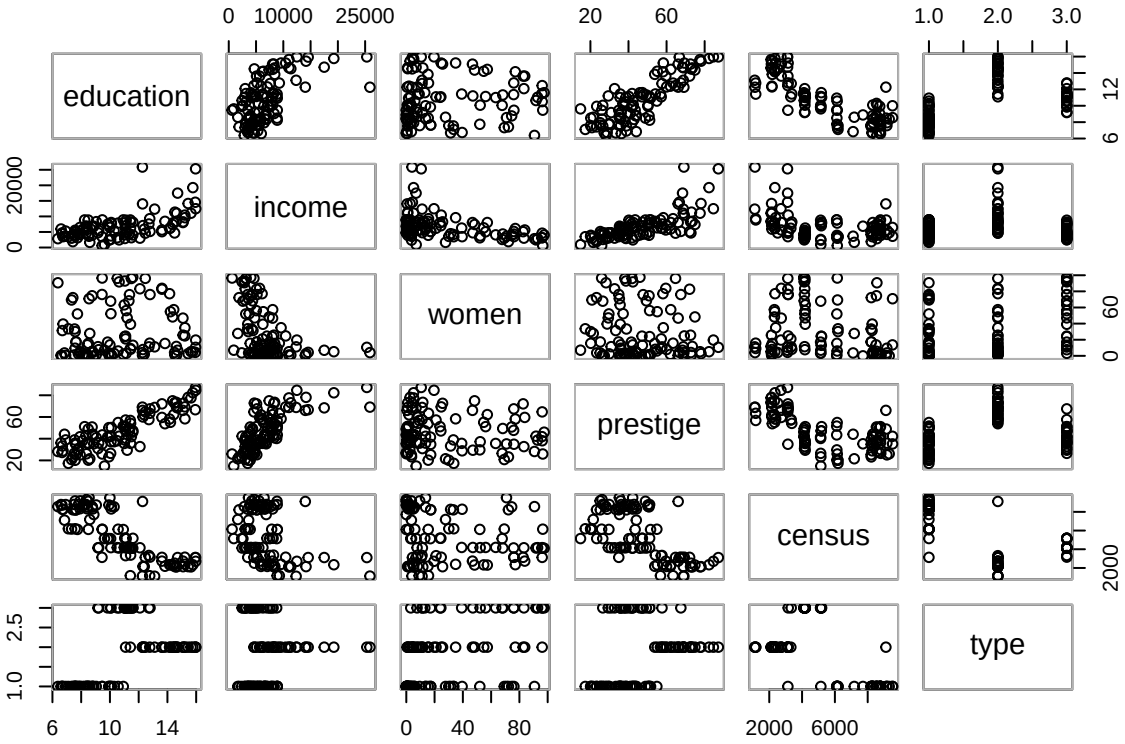
```
##   [1] "gov.administrators"        "general.managers"
##   [3] "accountants"               "purchasing.officers"
##   [5] "chemists"                  "physicists"
##   [7] "biologists"                "architects"
##   [9] "civil.engineers"           "mining.engineers"
##  [11] "surveyors"                 "draughtsmen"
##  [13] "computer.programers"       "economists"
##  [15] "psychologists"             "social.workers"
##  [17] "lawyers"                   "librarians"
##  [19] "vocational.counsellors"    "ministers"
##  [21] "university.teachers"       "primary.school.teachers"
##  [23] "secondary.school.teachers" "physicians"
##  [25] "veterinarians"             "osteopaths.chiropractors"
##  [27] "nurses"                    "nursing.aides"
##  [29] "physio.therapsts"          "pharmacists"
##  [31] "medical.technicians"       "commercial.artists"
##  [33] "radio.tv.announcers"       "athletes"
##  [35] "secretaries"               "typists"
##  [37] "bookkeepers"               "tellers.cashiers"
##  [39] "computer.operators"        "shipping.clerks"
##  [41] "file.clerks"               "receptionsts"
##  [43] "mail.carriers"             "postal.clerks"
##  [45] "telephone.operators"       "collectors"
##  [47] "claim.adjustors"           "travel.clerks"
##  [49] "office.clerks"             "sales.supervisors"
##  [51] "commercial.travellers"     "sales.clerks"
##  [53] "newsboys"                  "service.station.attendant"
##  [55] "insurance.agents"          "real.estate.salesmen"
##  [57] "buyers"                    "firefighters"
##  [59] "policemen"                 "cooks"
##  [61] "bartenders"                "funeral.directors"
##  [63] "babysitters"               "launderers"
##  [65] "janitors"                  "elevator.operators"
##  [67] "farmers"                   "farm.workers"
##  [69] "rotary.well.drillers"      "bakers"
##  [71] "slaughterers.1"            "slaughterers.2"
##  [73] "canners"                   "textile.weavers"
##  [75] "textile.labourers"         "tool.die.makers"
##  [77] "machinists"                "sheet.metal.workers"
##  [79] "welders"                   "auto.workers"
##  [81] "aircraft.workers"          "electronic.workers"
```

```
##  [83] "radio.tv.repairmen"        "sewing.mach.operators"
##  [85] "auto.repairmen"            "aircraft.repairmen"
##  [87] "railway.sectionmen"        "electrical.linemen"
##  [89] "electricians"              "construction.foremen"
##  [91] "carpenters"                "masons"
##  [93] "house.painters"            "plumbers"
##  [95] "construction.labourers"    "pilots"
##  [97] "train.engineers"           "bus.drivers"
##  [99] "taxi.drivers"              "longshoremen"
## [101] "typesetters"               "bookbinders"
```

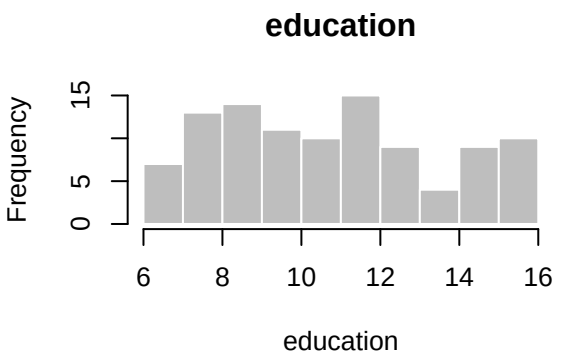
```
library(gclus)
```

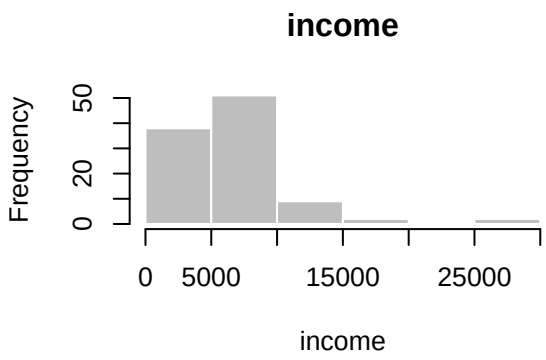
```
## Loading required package: cluster
```

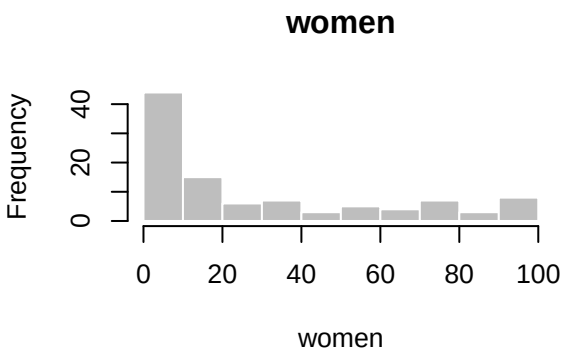
```
cpairs(df)
```

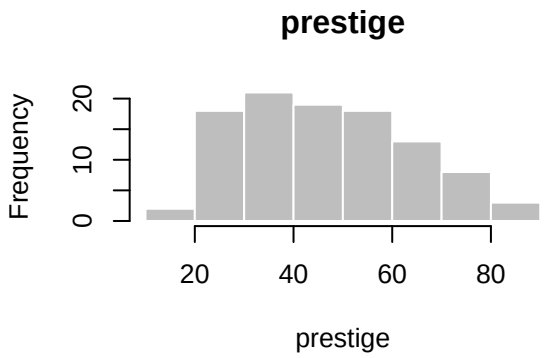



```
par(mfrow = c(2,2))
for (i in c("education","income","women","prestige")) {
  hist(df[,i], xlab = i, main = paste(i),
       col = "gray", border = "white")
}
```



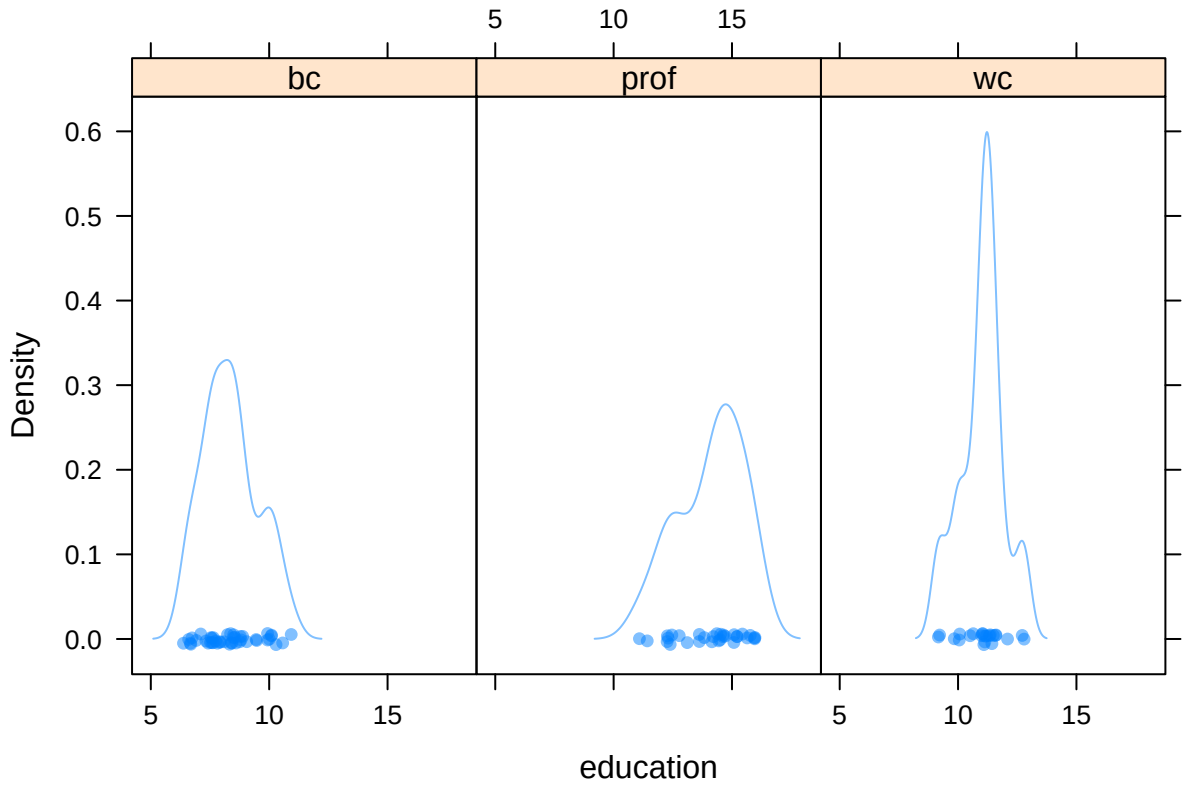







```r
library(lattice)
library(MASS)

densityplot( ~ education | type, data = df, pch = 16, alpha = 0.5)
```




```
par(mfrow= c(1,1), mar = rep(4,4), oma = rep(2,4))
library(corrplot) # for correlation plot, install if needed
```

```
## corrplot 0.84 loaded
```

```
library(gplots) # color interpolation, install if needed
```

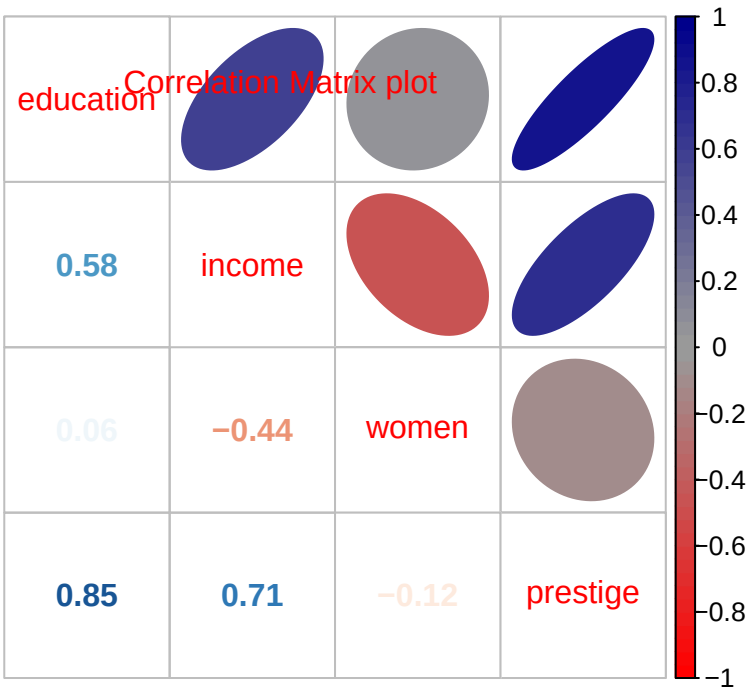
```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
corr_df = cor(df[,c("education","income","women","prestige")])

corrplot.mixed(corr=corr_df,
               upper="ellipse", tl.pos='d',
               upper.col = colorpanel(50, "red", "gray60", "blue4"))

title(main = "Correlation Matrix plot",
      cex.main = 1, font.main= 1, col.main= "red", adj = 0.5)
```




```
corr_df = cor(df[,c("education","income","women","prestige")])
corr_df
```

```
##             education     income       women   prestige
## education 1.00000000  0.5775802  0.06185286  0.8501769
## income    0.57758023  1.0000000 -0.44105927  0.7149057
## women     0.06185286 -0.4410593  1.00000000 -0.1183342
## prestige  0.85017689  0.7149057 -0.11833419  1.0000000
```

```
df$log_income = log(df$income)
head(df)
```

```
##                     education income women prestige census type log_income
## gov.administrators      13.11  12351 11.16     68.8   1113 prof   9.421492
## general.managers        12.26  25879  4.02     69.1   1130 prof  10.161187
## accountants             12.77   9271 15.70     63.4   1171 prof   9.134647
## purchasing.officers     11.42   8865  9.11     56.8   1175 prof   9.089866
## chemists                14.62   8403 11.68     73.5   2111 prof   9.036344
## physicists              15.64  11030  5.13     77.6   2113 prof   9.308374
```

```
mod1 <- lm(formula = prestige ~ education + income + women + type, data = df)
summary(mod1)
```

```
##
## Call:
## lm(formula = prestige ~ education + income + women + type, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -14.7485  -4.4817   0.3119   5.2478  18.4978
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.8139032  5.3311558  -0.153 0.878994
## education    3.6623557  0.6458300   5.671 1.63e-07 ***
## income       0.0010428  0.0002623   3.976 0.000139 ***
## women        0.0064434  0.0303781   0.212 0.832494
## typeprof     5.9051970  3.9377001   1.500 0.137127
## typewc      -2.9170720  2.6653961  -1.094 0.276626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.132 on 92 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.8349, Adjusted R-squared:  0.826
## F-statistic: 93.07 on 5 and 92 DF,  p-value: < 2.2e-16
```

```
mod2 <- lm(formula = prestige ~ education + log_income + type+women, data = df)
summary(mod2)
```

```
##
## Call:
## lm(formula = prestige ~ education + log_income + type + women,
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8762  -4.0579   0.5503   4.2129  16.6400
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -115.67219   18.80181  -6.152 1.96e-08 ***
## education      2.97384    0.60205   4.940 3.49e-06 ***
## log_income    14.65518    2.31151   6.340 8.42e-09 ***
## typeprof       5.29186    3.55585   1.488   0.1401
## typewc        -3.21599    2.40654  -1.336   0.1847
## women          0.08382    0.03223   2.601   0.0108 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.44 on 92 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.8654, Adjusted R-squared:  0.8581
## F-statistic: 118.3 on 5 and 92 DF,  p-value: < 2.2e-16
```

After log transforming the variable income, R squared value has increased and women variable became statistically significant.

```
mod3 <- lm(formula = prestige ~ education + log_income + women, data = df)
summary(mod3)
```

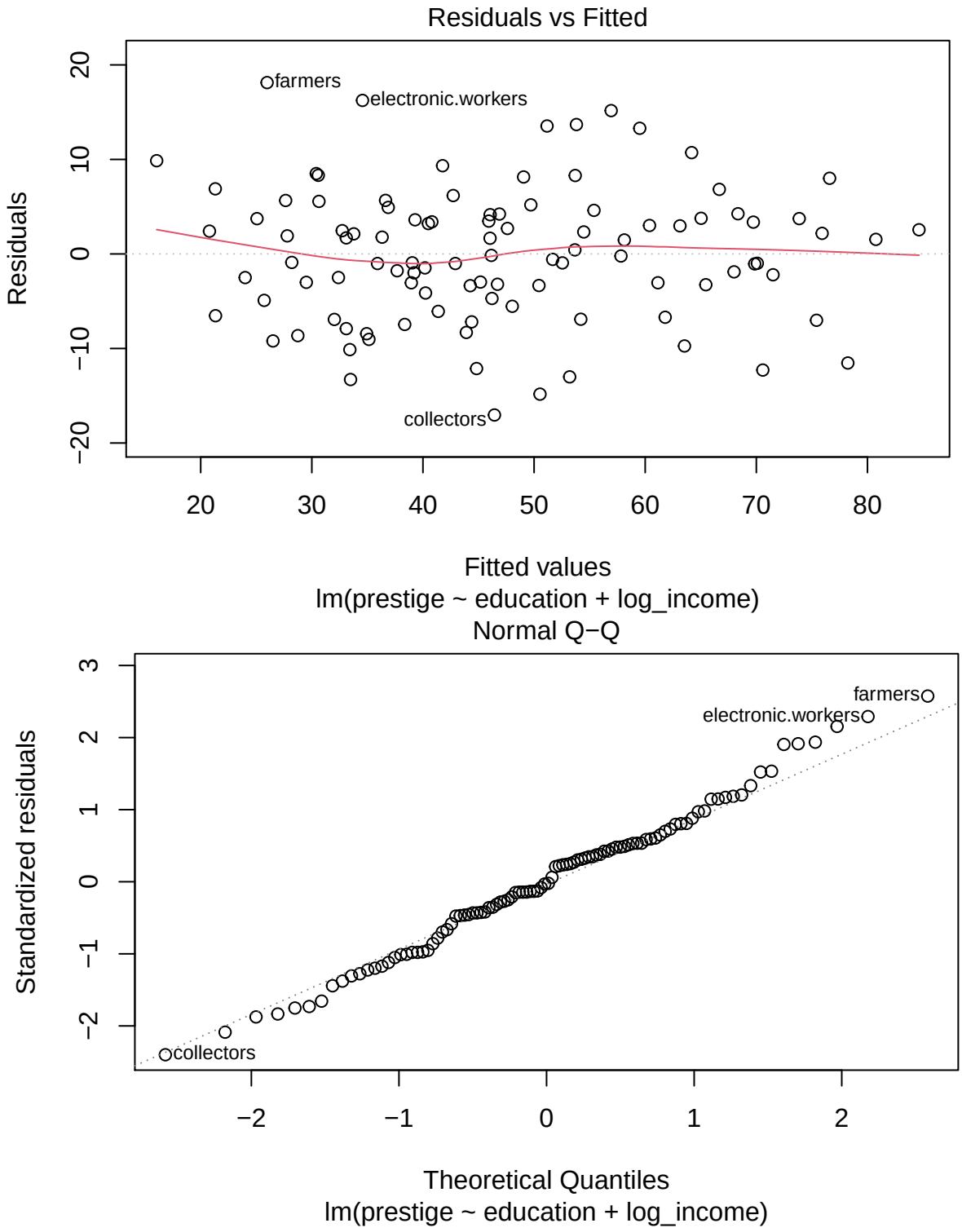
```
##
```

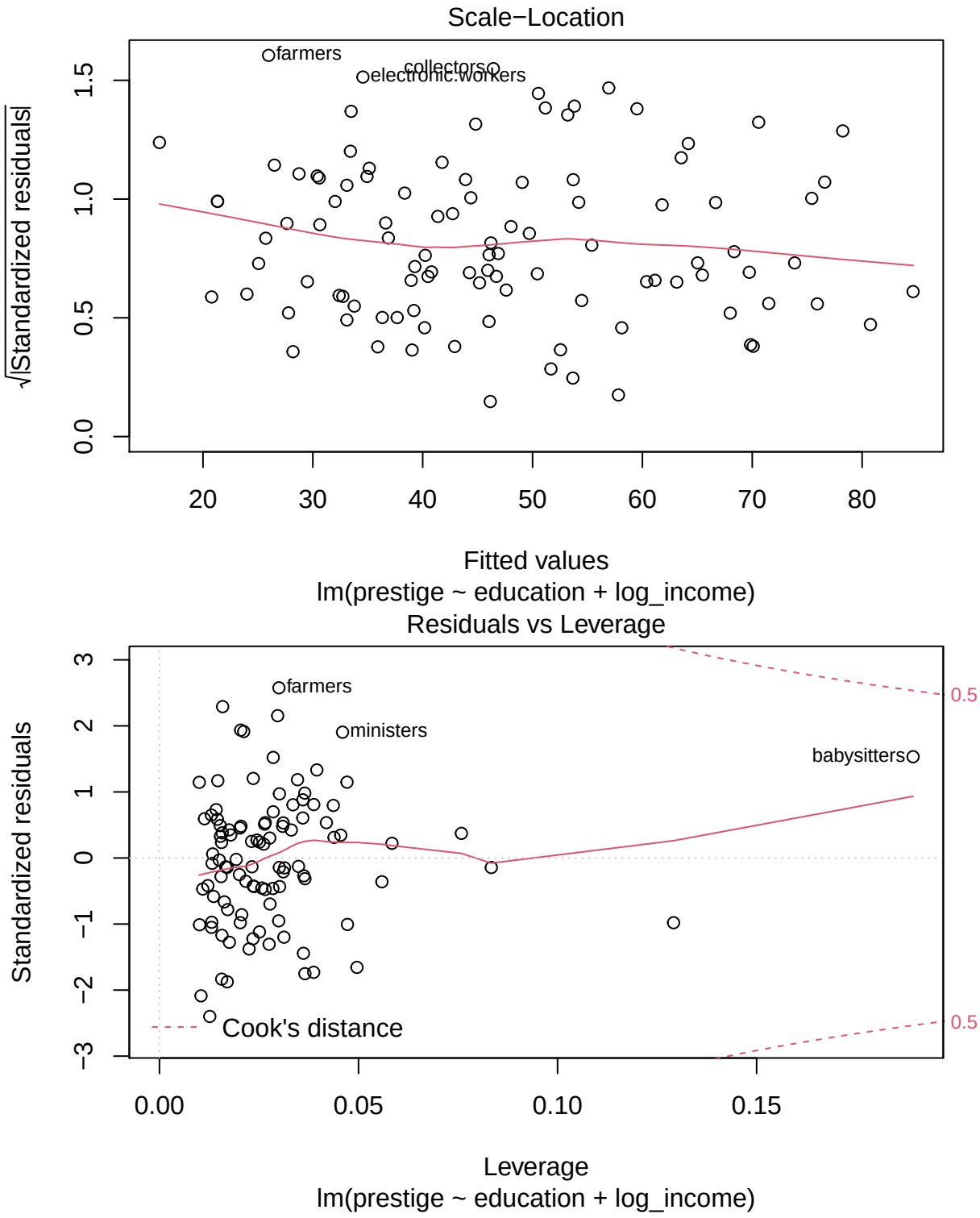
```
## Call:
## lm(formula = prestige ~ education + log_income + women, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.364  -4.429  -0.101   4.316  19.179
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -110.9658    14.8429  -7.476 3.27e-11 ***
## education      3.7305     0.3544  10.527  < 2e-16 ***
## log_income    13.4382     1.9138   7.022 2.90e-10 ***
## women          0.0469     0.0299   1.568     0.12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.093 on 98 degrees of freedom
## Multiple R-squared:  0.8351, Adjusted R-squared:   0.83
## F-statistic: 165.4 on 3 and 98 DF,  p-value: < 2.2e-16
```

```
mod4 <- lm(formula = prestige ~ education + log_income, data = df)
summary(mod4)
```

```
##
## Call:
## lm(formula = prestige ~ education + log_income, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.0346  -4.5657  -0.1857   4.0577  18.1270
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -95.1940    10.9979  -8.656 9.27e-14 ***
## education     4.0020     0.3115  12.846  < 2e-16 ***
## log_income   11.4375     1.4371   7.959 2.94e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.145 on 99 degrees of freedom
## Multiple R-squared:  0.831,  Adjusted R-squared:  0.8275
## F-statistic: 243.3 on 2 and 99 DF,  p-value: < 2.2e-16
```

```
plot(mod4)
```


Residuals vs Fitted
Residuals
Fitted values
lm(prestige ~ education + log_income)
farmers
electronic.workers
collectors
Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(prestige ~ education + log_income)
farmers
electronic.workers
collectors



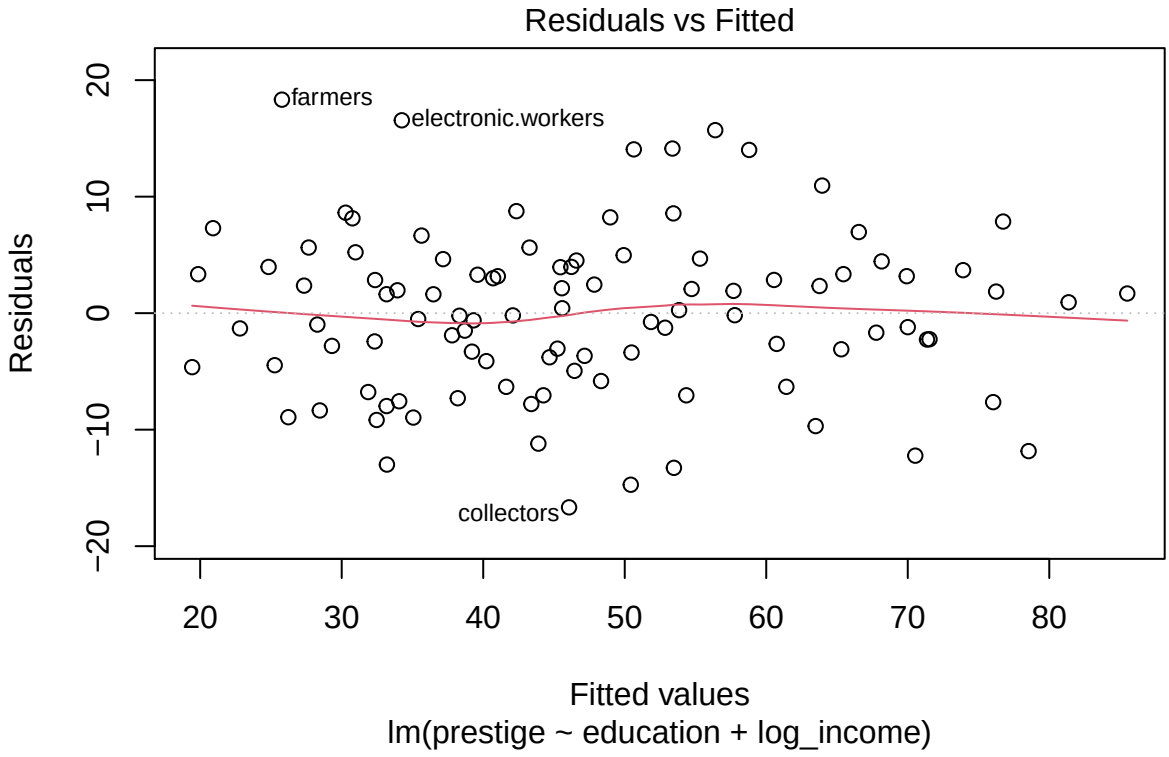
Farmers and Electronic Workers have higher prestige than the predicted values from our model. This suggests the prestige is high for Farmers and Electronic Workers than other other professions having similar levels of income and education.

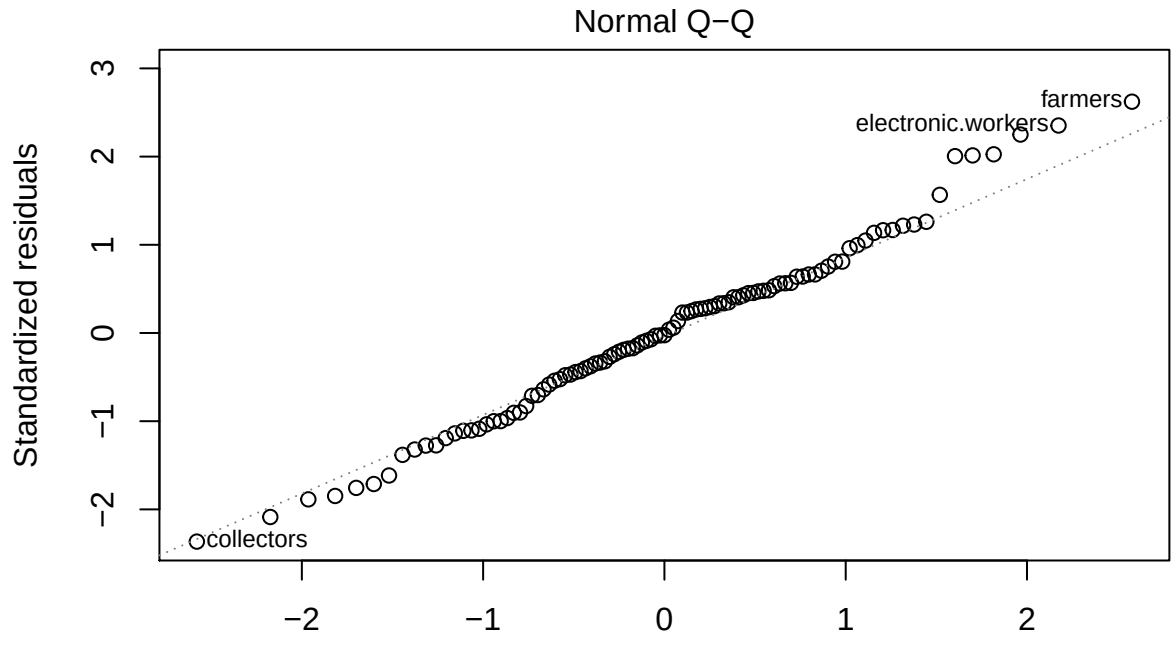
Collectors have lower prestige levels than peers having similar education and income.

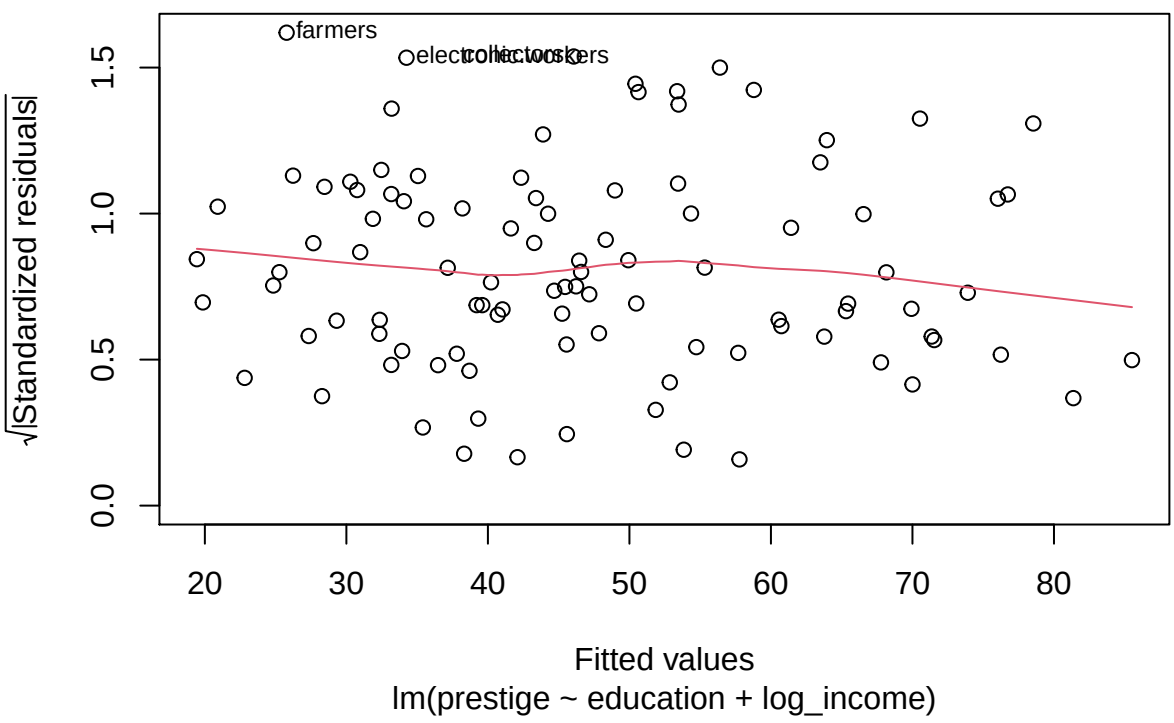
```r
mod5 <- lm(formula = prestige ~ education + log_income, data = df[!(row.names(df)) %in% c("babysitters")
summary(mod5)
```

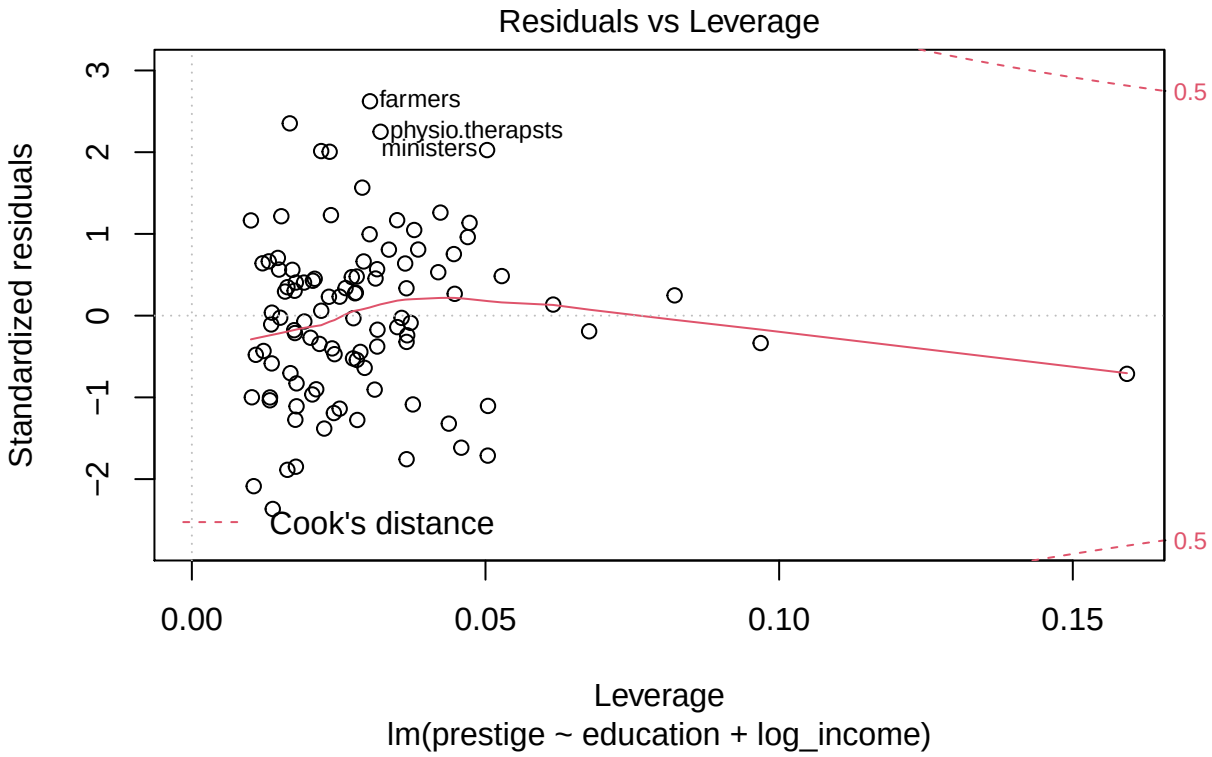
```
##
## Call:
## lm(formula = prestige ~ education + log_income, data = df[!(row.names(df)) %in%
##     c("babysitters"), ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.6658  -4.4662  -0.1762   3.9665  18.3236
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -103.1453    12.0753  -8.542 1.75e-13 ***
## education      3.9002     0.3163  12.330  < 2e-16 ***
## log_income    12.4680     1.5755   7.913 3.88e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.095 on 98 degrees of freedom
## Multiple R-squared:  0.8325, Adjusted R-squared:  0.8291
## F-statistic: 243.5 on 2 and 98 DF,  p-value: < 2.2e-16
```

```r
plot(mod5)
```




Normal Q−Q
Standardized residuals
farmers
electronic.workers
collectors
−2
−1
0
1
2
3
Theoretical Quantiles
lm(prestige ~ education + log_income)


Scale−Location
√|Standardized residuals|
farmers
0.0
0.5
1.0
1.5
20
30
40
50
60
70
80
Fitted values
lm(prestige ~ education + log_income)



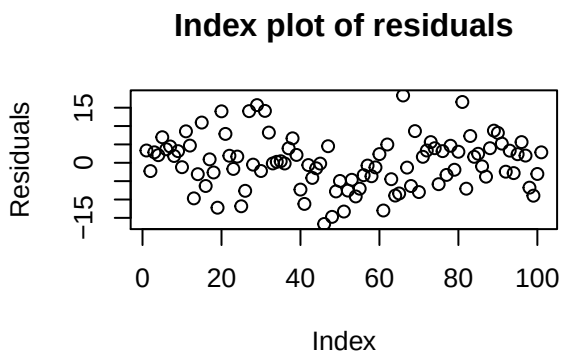

```
par(mfrow = c(2,2))
plot(mod5$residuals,ylab="Residuals",main="Index plot of residuals")
profession <- row.names(df)
profession <- profession[!profession %in% c("babysitters")]
identify(1:101,mod5$res,profession)
```

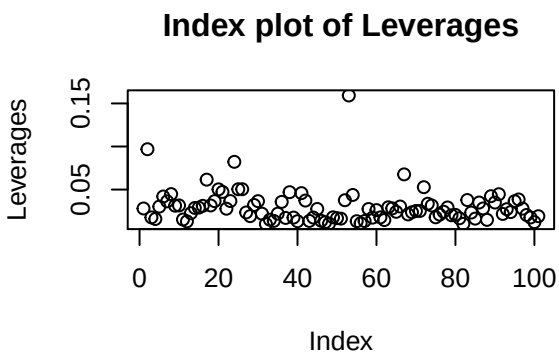
```
## integer(0)
```

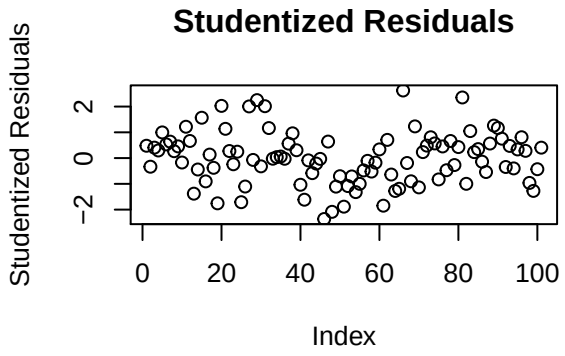
```
x <- model.matrix(mod5)
lev <- hat(x)
plot(lev,ylab="Leverages",main="Index plot of Leverages")
abline(h=2*5/50)
```

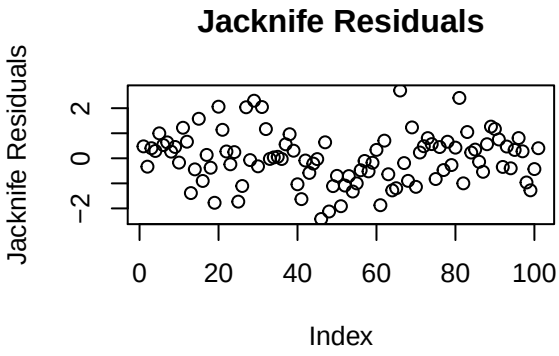
```
gs <- summary(mod5)
stud <- mod5$res/(gs$sig*sqrt(1-lev))
plot(stud,ylab="Studentized Residuals",main="Studentized Residuals")
```

```
jack <- rstudent(mod5)
plot(jack,ylab="Jacknife Residuals",main="Jacknife Residuals") > jack[abs(jack)==max(abs(jack))]
```










```
## logical(0)
```

```
str(profession)
```

```
##  chr [1:101] "gov.administrators" "general.managers" "accountants" ...
```

```
df_res <- data.frame(sort(mod5$res))
head(df_res)
```

```
##                         sort.mod5.res.
## collectors                   -16.66578
## travel.clerks                -14.72608
## commercial.travellers        -13.27587
## bartenders                   -12.99615
## vocational.counsellors       -12.23180
## veterinarians                -11.84041
```

```
tail(df_res)
```

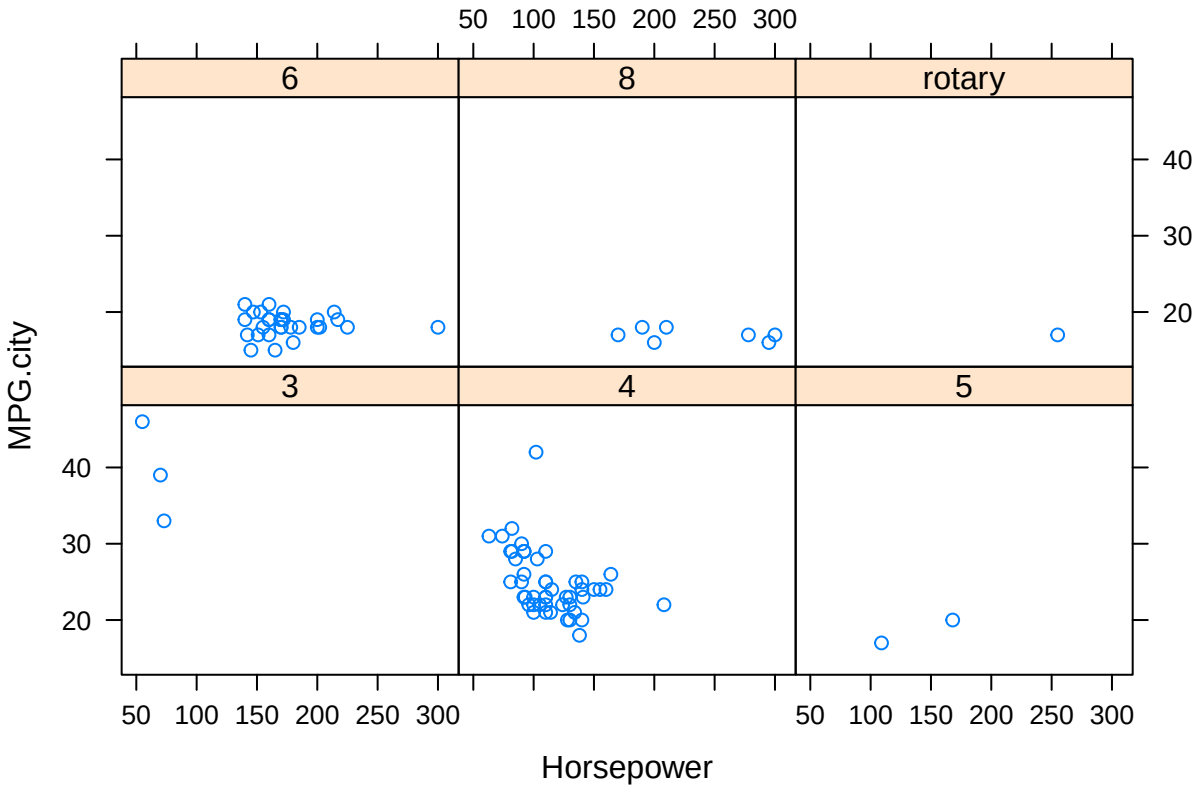
```
##                       sort.mod5.res.
## ministers                   14.00906
## nurses                      14.05852
## medical.technicians         14.12879
## physio.therapsts            15.70526
## electronic.workers          16.55178
## farmers                     18.32364
```

```r
profession[!profession %in% c("babysitters")]
```

```
##   [1] "gov.administrators"        "general.managers"
##   [3] "accountants"               "purchasing.officers"
##   [5] "chemists"                  "physicists"
##   [7] "biologists"                "architects"
##   [9] "civil.engineers"           "mining.engineers"
##  [11] "surveyors"                 "draughtsmen"
##  [13] "computer.programers"       "economists"
##  [15] "psychologists"             "social.workers"
##  [17] "lawyers"                   "librarians"
##  [19] "vocational.counsellors"    "ministers"
##  [21] "university.teachers"       "primary.school.teachers"
##  [23] "secondary.school.teachers" "physicians"
##  [25] "veterinarians"             "osteopaths.chiropractors"
##  [27] "nurses"                    "nursing.aides"
##  [29] "physio.therapsts"          "pharmacists"
##  [31] "medical.technicians"       "commercial.artists"
##  [33] "radio.tv.announcers"       "athletes"
##  [35] "secretaries"               "typists"
##  [37] "bookkeepers"               "tellers.cashiers"
##  [39] "computer.operators"        "shipping.clerks"
##  [41] "file.clerks"               "receptionsts"
##  [43] "mail.carriers"             "postal.clerks"
##  [45] "telephone.operators"       "collectors"
##  [47] "claim.adjustors"           "travel.clerks"
##  [49] "office.clerks"             "sales.supervisors"
##  [51] "commercial.travellers"     "sales.clerks"
##  [53] "newsboys"                  "service.station.attendant"
##  [55] "insurance.agents"          "real.estate.salesmen"
##  [57] "buyers"                    "firefighters"
##  [59] "policemen"                 "cooks"
##  [61] "bartenders"                "funeral.directors"
##  [63] "launderers"                "janitors"
##  [65] "elevator.operators"        "farmers"
##  [67] "farm.workers"              "rotary.well.drillers"
##  [69] "bakers"                    "slaughterers.1"
##  [71] "slaughterers.2"            "canners"
##  [73] "textile.weavers"           "textile.labourers"
##  [75] "tool.die.makers"           "machinists"
##  [77] "sheet.metal.workers"       "welders"
##  [79] "auto.workers"              "aircraft.workers"
##  [81] "electronic.workers"        "radio.tv.repairmen"
##  [83] "sewing.mach.operators"     "auto.repairmen"
##  [85] "aircraft.repairmen"        "railway.sectionmen"
##  [87] "electrical.linemen"        "electricians"
##  [89] "construction.foremen"      "carpenters"
##  [91] "masons"                    "house.painters"
##  [93] "plumbers"                  "construction.labourers"
##  [95] "pilots"                    "train.engineers"
##  [97] "bus.drivers"               "taxi.drivers"
##  [99] "longshoremen"              "typesetters"
## [101] "bookbinders"
```

```
library(lattice)
library(MASS)

xyplot(MPG.city ~ Horsepower | Cylinders, data = Cars93)
```




```
df[,c("education","income","women","prestige","log_income")]
```

```
##                          education income women prestige log_income
## gov.administrators           13.11  12351 11.16     68.8   9.421492
## general.managers             12.26  25879  4.02     69.1  10.161187
## accountants                  12.77   9271 15.70     63.4   9.134647
## purchasing.officers          11.42   8865  9.11     56.8   9.089866
## chemists                     14.62   8403 11.68     73.5   9.036344
## physicists                   15.64  11030  5.13     77.6   9.308374
## biologists                   15.09   8258 25.65     72.6   9.018938
## architects                   15.44  14163  2.69     78.1   9.558388
## civil.engineers              14.52  11377  1.03     73.1   9.339349
## mining.engineers             14.64  11023  0.94     68.8   9.307739
## surveyors                    12.39   5902  1.91     62.0   8.683047
## draughtsmen                  12.30   7059  7.83     60.0   8.862059
## computer.programers          13.83   8425 15.33     53.8   9.038959
## economists                   14.44   8049 57.31     62.2   8.993303
## psychologists                14.36   7405 48.28     74.9   8.909911
## social.workers               14.21   6336 54.77     55.1   8.754003
## lawyers                      15.77  19263  5.13     82.3   9.865941
## librarians                   14.15   6112 77.10     58.1   8.718009
## vocational.counsellors       15.22   9593 34.89     58.3   9.168789
```

```
## ministers                    14.50   4686  4.14     72.8   8.452335
## university.teachers          15.97  12480 19.59     84.6   9.431883
## primary.school.teachers      13.62   5648 83.78     59.6   8.639057
## secondary.school.teachers    15.08   8034 46.80     66.1   8.991438
## physicians                   15.96  25308 10.56     87.2  10.138876
## veterinarians                15.94  14558  4.32     66.7   9.585896
## osteopaths.chiropractors     14.71  17498  6.91     68.4   9.769842
## nurses                       12.46   4614 96.12     64.7   8.436850
## nursing.aides                 9.45   3485 76.14     34.9   8.156223
## physio.therapsts             13.62   5092 82.66     72.1   8.535426
## pharmacists                  15.21  10432 24.71     69.3   9.252633
## medical.technicians          12.79   5180 76.04     67.5   8.552560
## commercial.artists           11.09   6197 21.03     57.2   8.731821
## radio.tv.announcers          12.71   7562 11.15     57.6   8.930891
## athletes                     11.44   8206  8.13     54.1   9.012621
## secretaries                  11.59   4036 97.51     46.0   8.303009
## typists                      11.49   3148 95.97     41.9   8.054523
## bookkeepers                  11.32   4348 68.24     49.4   8.377471
## tellers.cashiers             10.64   2448 91.76     42.3   7.803027
## computer.operators           11.36   4330 75.92     47.7   8.373323
## shipping.clerks               9.17   4761 11.37     30.9   8.468213
## file.clerks                  12.09   3016 83.19     32.7   8.011687
## receptionsts                 11.04   2901 92.86     38.7   7.972811
## mail.carriers                 9.22   5511  7.62     36.1   8.614501
## postal.clerks                10.07   3739 52.27     37.2   8.226573
## telephone.operators          10.51   3161 96.14     38.1   8.058644
## collectors                   11.20   4741 47.06     29.4   8.464003
## claim.adjustors              11.13   5052 56.10     51.1   8.527539
## travel.clerks                11.43   6259 39.17     35.7   8.741776
## office.clerks                11.00   4075 63.23     35.6   8.312626
## sales.supervisors             9.84   7482 17.04     41.5   8.920255
## commercial.travellers        11.13   8780  3.16     40.2   9.080232
## sales.clerks                 10.05   2594 67.82     26.5   7.860956
## newsboys                      9.62    918  7.00     14.8   6.822197
## service.station.attendant     9.93   2370  3.69     23.3   7.770645
## insurance.agents             11.60   8131 13.09     47.3   9.003439
## real.estate.salesmen         11.09   6992 24.44     47.1   8.852522
## buyers                       11.03   7956 23.88     51.1   8.981682
## firefighters                  9.47   8895  0.00     43.5   9.093245
## policemen                    10.93   8891  1.65     51.6   9.092795
## cooks                         7.74   3116 52.00     29.7   8.044305
## bartenders                    8.50   3930 15.51     20.2   8.276395
## funeral.directors            10.57   7869  6.01     54.9   8.970686
## babysitters                   9.46    611 96.53     25.9   6.415097
## launderers                    7.33   3000 69.31     20.8   8.006368
## janitors                      7.11   3472 33.57     17.3   8.152486
## elevator.operators            7.58   3582 30.08     20.1   8.183677
## farmers                       6.84   3643  3.60     44.1   8.200563
## farm.workers                  8.60   1656 27.75     21.5   7.412160
## rotary.well.drillers          8.88   6860  0.00     35.3   8.833463
## bakers                        7.54   4199 33.30     38.9   8.342602
## slaughterers.1                7.64   5134 17.26     25.2   8.543640
## slaughterers.2                7.64   5134 17.26     34.8   8.543640
## canners                       7.42   1890 72.24     23.2   7.544332
```

```
## textile.weavers              6.69   4443 31.36     33.3   8.399085
## textile.labourers            6.74   3485 39.48     28.8   8.156223
## tool.die.makers             10.09   8043  1.50     42.5   8.992557
## machinists                   8.81   6686  4.28     44.2   8.807771
## sheet.metal.workers          8.40   6565  2.30     35.9   8.789508
## welders                      7.92   6477  5.17     41.8   8.776013
## auto.workers                 8.43   5811 13.62     35.9   8.667508
## aircraft.workers             8.78   6573  5.78     43.7   8.790726
## electronic.workers           8.76   3942 74.54     50.8   8.279443
## radio.tv.repairmen          10.29   5449  2.92     37.2   8.603187
## sewing.mach.operators        6.38   2847 90.67     28.2   7.954021
## auto.repairmen               8.10   5795  0.81     38.1   8.664751
## aircraft.repairmen          10.10   7716  0.78     50.3   8.951051
## railway.sectionmen           6.67   4696  0.00     27.3   8.454466
## electrical.linemen           9.05   8316  1.34     40.9   9.025937
## electricians                 9.93   7147  0.99     50.2   8.874448
## construction.foremen         8.24   8880  0.65     51.1   9.091557
## carpenters                   6.92   5299  0.56     38.9   8.575273
## masons                       6.60   5959  0.52     36.2   8.692658
## house.painters               7.81   4549  2.46     29.9   8.422663
## plumbers                     8.33   6928  0.61     42.9   8.843326
## construction.labourers       7.52   3910  1.09     26.5   8.271293
## pilots                      12.27  14032  0.58     66.1   9.549096
## train.engineers              8.49   8845  0.00     48.9   9.087608
## bus.drivers                  7.58   5562  9.47     35.9   8.623713
## taxi.drivers                 7.93   4224  3.59     25.1   8.348538
## longshoremen                 8.37   4753  0.00     26.1   8.466531
## typesetters                 10.00   6462 13.58     42.2   8.773694
## bookbinders                  8.55   3617 70.87     35.2   8.193400
```

Assumptions of Linear Regression model

There are several assumptions when a linear model is fitted to data. The first is that the relationship between the predictors and the outcomes is linear. If the relationship is not linear, then the model will make systematic errors. For example, if we generate data where y is a function of the square of x and then fit a linear model y x, this will draw a straight line through a cloud of points that is curved.

Another assumption of a linear model is that prediction errors—the parts of the data that do not exactly fit the model—are normally distributed and look like random noise with no pattern. One way to examine this is to plot the model's fitted values (the predictions) versus the residuals (the prediction errors).

The second plot in the lower left of Fig. 7.5 is similar to the first, except that instead of plotting the raw residual value, it plots the square root of the standardized residual. Again, there should be no clear pattern; if there were it might indicate a non-linear relationship. Observations with high residuals are flagged as potential outliers, and R labels them with row numbers in case we wish to inspect them in the data frame.

A QQ plot helps you see **whether the residuals follow a normal distribution**, another key assumption (see Sect. 3.4.3). It compares the values that residuals would be expected to take if they are normally distributed, versus their actual values. When the model is appropriate, these points are similar and fall close to a diagonal line; when the relationship between the variables is non-linear or otherwise does not match the assumption, the points deviate from the diagonal line.

Quantile-quantile (QQ) plots are a good way to check one's data against a distribution that you think it should come from. Some common statistics such as the correlation coefficient r (to be precise, the Pearson product-moment correlation coefficient) are interpreted under an assumption that data are normally distributed. A QQ plot can confirm that the distribution is, in fact, normal by plotting the observed quantiles of your data against the quantiles that would be expected for a normal distribution.

The final plot in the lower right panel of helps to **identify potential outliers**, observations that may come from a different distribution than the others. Outliers are a problem because, if they are far from other points, they unduly influence the fitted line. We do not want one or a very few observations to have a large effect on the coefficients. The lower right plot plots the leverage of each point, a measure of how much influence the point has on the model coefficients. When a point has a high residual and high leverage, it indicates that the point has both a different pattern (residual) and undue influence (leverage). One measure of the leverage of a data point is Cook's distance, an estimate of how much predicted (y) values would change if the model were re-estimated with that point eliminated from the data. If you have observations with high Cook's distance, this chart would show dotted lines for the distances; in the present case, there are none.

Studentized Deleted Residuals and Outliers
{Statistics for Business and Economics, Sweeney|Anderson|Williams, section 15.8, Page 676}

Suppose the ith observation is deleted from the data set and a new estimated regression equation is developed with the remaining n 1 observations. Let s(i) denote the standard error of the estimate based on the data set with the ith observation deleted. If we compute the standard deviation of residual i using $s_{(i)}$ instead of s, and then compute the standardized residual for observation i using the revised $s_{y-\hat{y}}$ value,the resulting standardized residiual is called a ***studentized deleted residual***. If the ith observation is an outlier, $s_{(i)}$ will be less than s. The absolute value of the ith studentized deleted residual therefore will be larger than the absolute value of the standardized residual. In this sense, studentized deleted residuals may detect outliers that standardized residuals do not detect.

Using Cook's Distance Measure to Identify Influential Observations

An observation can be identified as having high leverage and not necessarily be influential in terms of the resulting estimated regression equation.

Cook's distance measure uses both the leverage of observation i, h_i, and the residual for observation i, $(y_i - \hat{y})$, to determine whether the observation is influential.

COOK'S DISTANCE MEASURE

$D_i = \frac{(y_i - \hat{y})^2}{(p+1)s^2} \frac{h_i}{(1-h_i)^2}$

Where
D_i = Cook's distance measure for observation i
$(y_i - \hat{y})$ = the residual for observation i
h_i = the leverage for observation i
p = the number of independent variables
s = the standard error of the estimate

The value of Cook's distance measure will be large and indicate an influential observation if the residual or the leverage is large. As a rule of thumb, values of $D_i > 1$ indicate that the ith observation is influential and should be studied further.

Log Transform

The log transform is a powerful tool for dealing with positive numbers with a heavy- tailed distribution. (A heavy-tailed distribution places more probability mass in the tail range than a Gaussian distribution.) It compresses the long tail in the high end of the distribution into a shorter tail, and expands the low end into a longer head. {***Feature Engineering for Machine Learning***}

Feature Scaling or Normalization

Some features, such as latitude or longitude, are bounded in value. Other numeric features, such as counts, may increase without bound. Models that are smooth functions of the input, such as linear regression, logistic regression, or anything that involves a matrix, are affected by the scale of the input.
If your model is sensitive to the scale of input features, feature scaling could help. As the name suggests, feature scaling changes the scale of the feature. Sometimes people also call it feature normalization. Feature scaling is usually done individually to each feature.

Several types of common scaling operations, each resulting in a different distribution of feature values.

1. Min-Max Scaling
Let x be an individual feature value (i.e., a value of the feature in some data point), and min(x) and max(x), respectively, be the minimum and maximum values of this feature over the entire dataset. Min-max scaling squeezes (or stretches) all feature values to be within the range of [0, 1]. Figure 2-15 demonstrates this concept. The formula for min-max scaling is:
$\overline{x} = \frac{x - min(x)}{max(x) - min(x)}$

2. Standardization (Variance Scaling)