Regression Diagnostics

eCornell

7/28/2021

This R Markdown file demonstrates how to use residuals to evaluate a regression analysis. The example here evaluates the regression analyses done on the Prestige data set.

Step 1: Load the data and define colors.

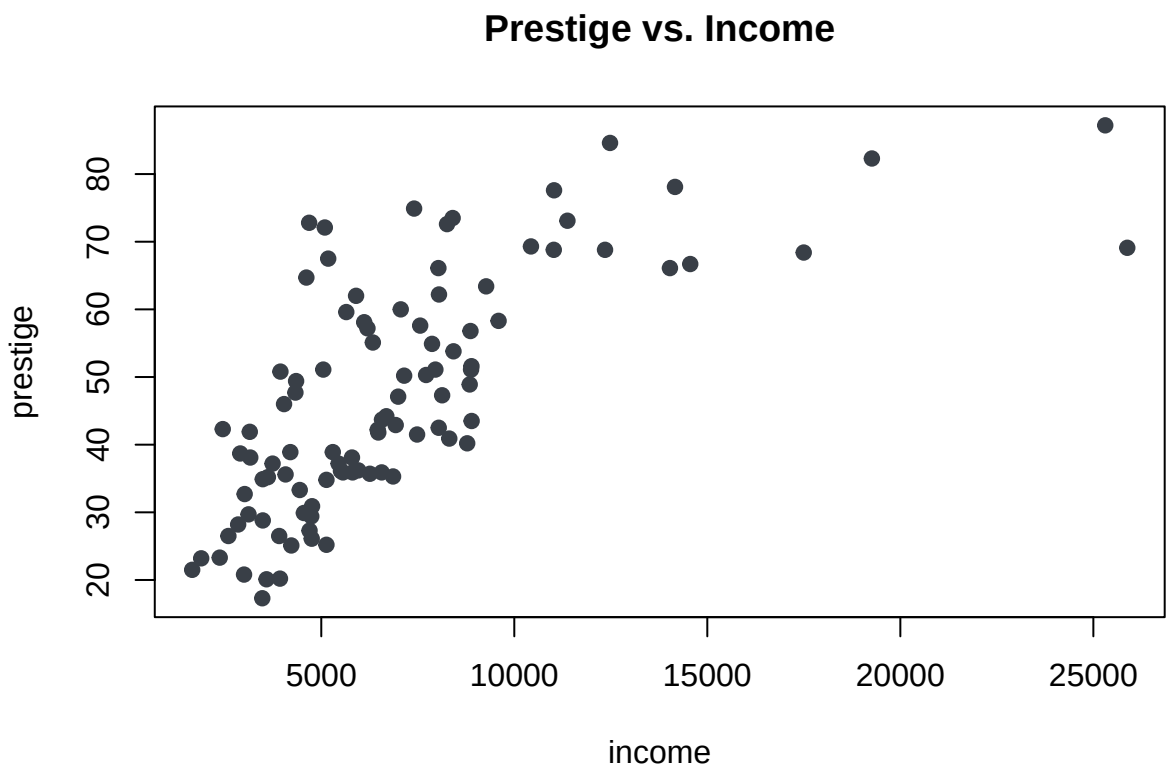
```
knitr::opts_chunk$set(echo = TRUE)
library(carData) # The Prestige data set is available in the carData library
data(Prestige) # Load the Prestige data set
Prestige = Prestige[!is.na(Prestige$type),] # exclude any observations that do not have an entry in the
Prestige$logincome <- log(Prestige$income) # add log(income) as a variable to the Prestige data frame

#eCornell Hex Codes:
crimson = '#b31b1b' #Crimson
lightGray = '#cecece' #lightGray
darkGray = '#606366'
skyBlue = '#92b2c4' #skyblue
gold = '#fbb040' #gold
ecBlack = '#393f47' #ecBlack
```

Step 2: Examine the plot of income vs. prestige.

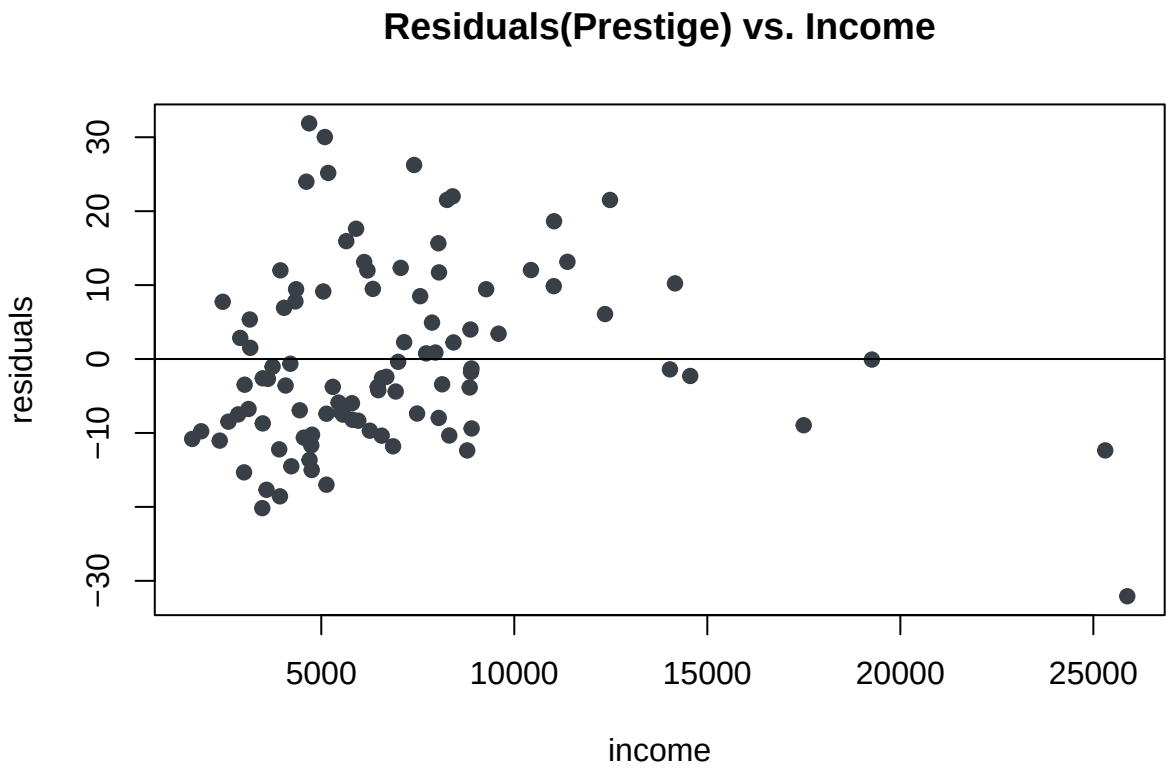
Plot prestige vs. income, then plot the regression line through the plot. Notice that the prestige score increases steeply at low levels of income, then stops increasing at high levels of income.

```
plot(prestige ~ income, data = Prestige, col = ecBlack,
     pch = 19, xlab = 'income', ylab = 'prestige',
     main = 'Prestige vs. Income')
```



##Step 3: Examine the residuals of income vs. prestige. Plot the residuals of prestige vs. income with a horizontal line at Y = 0, and notice that the residuals are not evenly distributed around the line.

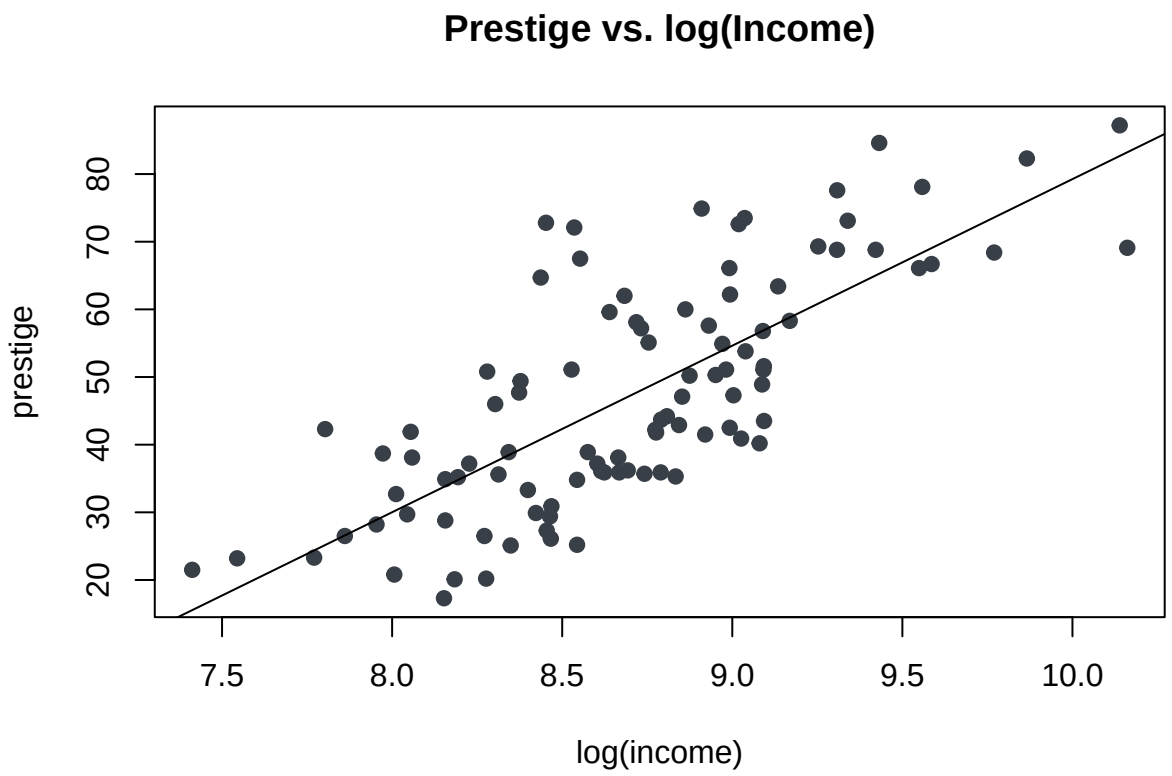
```
fit.inc <- lm(prestige ~ income, data = Prestige)
plot(Prestige$income, fit.inc$residuals, col = ecBlack,
     pch = 19, xlab = 'income', ylab = 'residuals',
     main = 'Residuals(Prestige) vs. Income')
abline(h = 0)
```



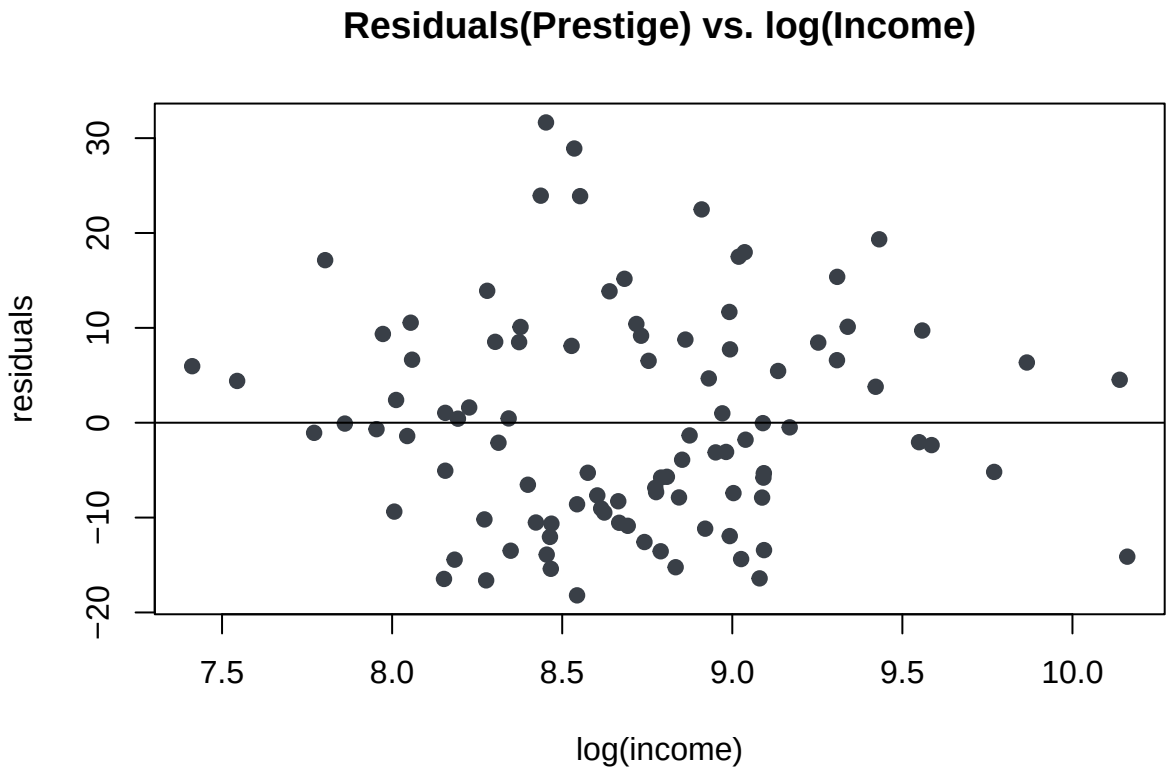
Step 4: Examine how residuals change after log-transforming income.

Remake the same plots, but this time with log(income). Notice that the plot of prestige vs. log(income) is linear, and that the residuals are now randomly distributed around the line at Y = 0.

```
plot(prestige ~ logincome, data = Prestige, col = ecBlack,
     pch = 19, xlab = 'log(income)', ylab = 'prestige',
     main = 'Prestige vs. log(Income)')
fit.loginc <- lm(prestige ~ logincome, data = Prestige)
abline(fit.loginc)
```




```
plot(Prestige$logincome, fit.loginc$residuals, col = ecBlack,
     pch = 19, xlab = 'log(income)', ylab = 'residuals',
     main = 'Residuals(Prestige) vs. log(Income)')
abline(h = 0)
```



Step 5: Compare prediction rules based on their R-Squared values.

R-squared values indicate how much of the variation in a data set is explained by a prediction rule. You can use them to compare two or more prediction rules. Better prediction rules explain more of the variation in your data, so they have higher R-squared values.

```
fit.edu <- lm(prestige ~ education, data = Prestige)
MSE.edu <- mean(fit.edu$residuals^2)

y = Prestige$prestige
var.y = mean((y - mean(y))^2)

R2.edu = 1 - MSE.edu/var.y
R2.edu
```

```
## [1] 0.7507872
```

The R-squared value of 75% indicates that 75% of the variation in prestige score across different jobs is explained by its linear association with education.

```
fit.edu.inc <- lm(prestige ~ education + logincome, data = Prestige)
MSE.edu.inc <- mean(fit.edu.inc$residuals^2)

R2.edu.inc = 1 - MSE.edu.inc/var.y
R2.edu.inc
```

```
## [1] 0.8389468
```

When log(income) is included in our prediction rule, the percentage of explained variation went up from 75% to ~84%. This means that including log(income) in our prediction rule increases how much of the variation in our data set we can explain by ~9%.