TOOL

Assessing Linear Prediction Rules With Residuals

Once you've built a linear prediction rule, it is important to assess how well it fits your data. This step can help you evaluate or compare prediction rules. One way to assess fit is with residuals, which are measurements of the difference between the observed values of the response variable in your data set and the predicted values based on your linear prediction rule. This tool demonstrates how you can use residuals to determine whether a linear prediction rule is appropriate for your data, and shows how to use residuals to assess the goodness-of-fit of your prediction rule.

Using R With This Tool

The portions of this tool with a grey background are code text you can use to do the examples included in this tool. You can also modify them to use with your own data. In these examples:

- Commands are the lines of code that don't begin with a pound sign (#). Type these lines into R to carry out the command.
- Commented text begins with one pound sign and explains the lines of code.
- The example code output begins with two pound signs.

© 2021 Cornell University

Data Set Information

The **Prestige** data set contains information about different job types in Canada in 1971, and is part of the **carData** package in R. In this data set, **education** refers to the average number of years of education the employees that hold that job have had, and **prestige** refers to the Pineo-Porter prestige value which measures the pride an employee has in their job.

You can load and view the data set with the following code:

```
install.packages("carData")                    # Install the carData package
library(carData)                               # Load the carData package
data(Prestige)                                 # Load the Prestige data set
Prestige = Prestige[!is.na(Prestige$type),]  # Remove rows that do not
                              # contain information on the profession type
Prestige$logincome <- log(Prestige$income)  # Add log(income) as a
                                           # variable to the Prestige data
                                           # frame

head(Prestige)                 # View the first 6 rows of the data set
##                     education income women prestige census type logincome
## 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
```


© 2021 Cornell University

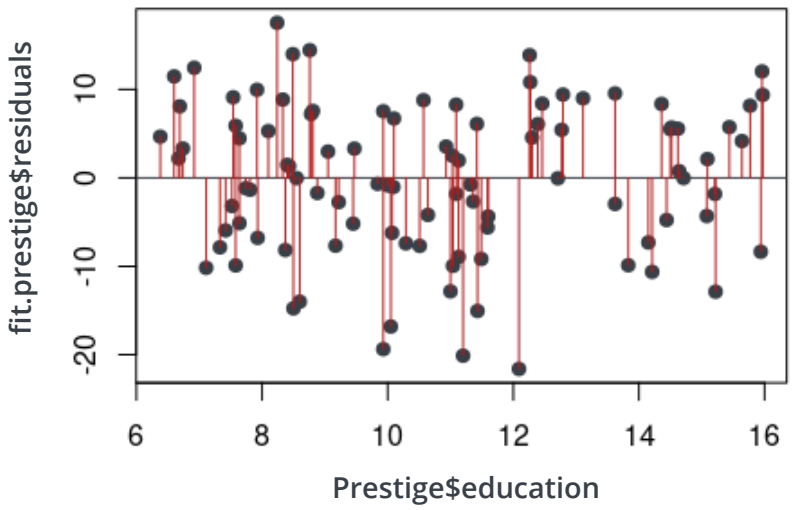
Detecting Nonlinearity With Regression Diagnostics

Building a linear prediction rule is appropriate if the response variable changes linearly with the predictor variables. This would happen, for example, when the average change in Y is roughly the same whether X changes from 1 to 2, or from 10 to 11. However, in many real world data sets the response changes at different rates, depending on the values of predictors. In such cases, a transformation of the predictor variable is needed — e.g., with a log or square root transformation — before fitting a linear regression line. Regression diagnostics are a systematic way to detect if there is some nonlinear pattern in your data that is not captured by your linear prediction rule.

For any simple linear regression, you can plot the residuals against each predictor variable. This plot should show that the residuals are distributed randomly across the entire range of the predictor variable, because it indicates there is no systematic nonlinear pattern in your data that your prediction rule failed to capture.

For example, the residuals for the linear prediction rule of **prestige** vs. **education** are distributed in this manner:

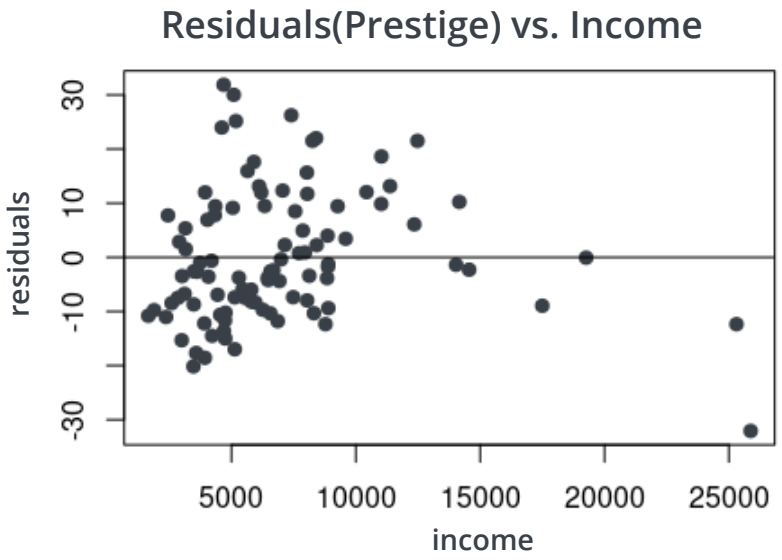
```
fit.prestige <- lm(prestige ~ education,    # Perform linear regression
    data = Prestige)
plot(Prestige$education,                    # Plot residuals
    fit.prestige$residuals,
    pch = 19, col = "black")
abline(h = 0, col = "black")           # Plot line at y = 0
for (i in 1:98){                       # Create lines to each residual point
  lines(rep(Prestige$education[i], 2),
        c(0, fit.prestige$residuals[i]),
        type = "l", col = "red", lwd = 1)
}
```



© 2021 Cornell University

In contrast, if you plot the residuals for the linear prediction rule of prestige vs. income, you can see that they are not distributed evenly around the line y = 0. Instead, at lower values of X they are mostly negative, at intermediate values of X they are mostly positive, and at very high values of X they are mostly negative:

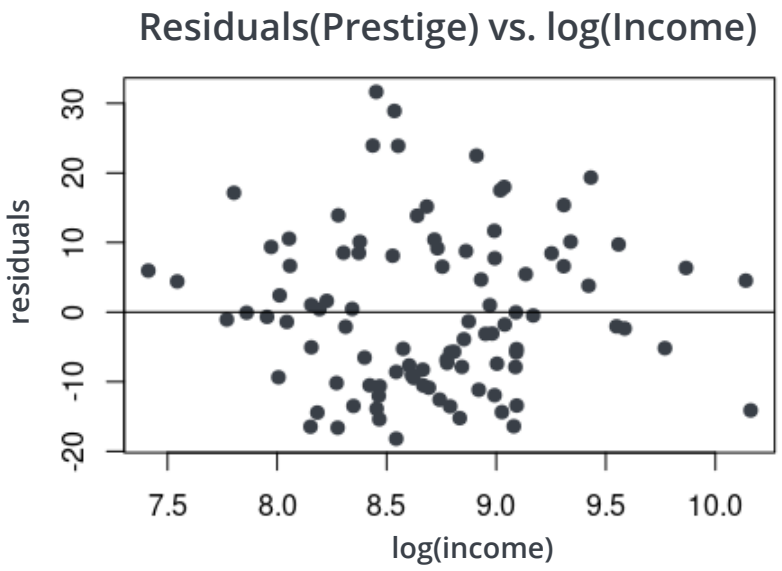
```
fit.inc <- lm(prestige ~ income, data = Prestige) # Perform linear regression
plot(Prestige$income,                             # Plot residuals
     fit.inc$residuals,
     col = "black",
     pch = 19, xlab = "income",
     ylab = "residuals",
     main = "Residuals(Prestige) vs. Income")
abline(h = 0)                                     # Plot line at y = 0
```



© 2021 Cornell University

One way to fix the systemic pattern in your data is to transform the predictor value to ensure they are linear. Transforming the data by taking the log of each observation causes the residuals to be more haphazardly distributed around the y = 0 line:

```
fit.loginc <- lm(prestige ~ logincome, data = Prestige)
plot(Prestige$logincome,
fit.loginc$residuals, col ="black" , pch = 19, xlab = "log(income)",
     ylab = "residuals", main = "Residuals(Prestige) vs. log(Income)")
abline(h = 0)
```



© 2021 Cornell University

Measuring Goodness-of-fit of a Linear Prediction Rule

When you use linear regression to create a linear prediction rule, you are condensing all of the information from the observations in the data set you're analyzing into one prediction rule, then using that same rule to describe each observation. The prediction rule might be very accurate, meaning it explains the data well, or it might not be accurate, meaning it does not explain the data very well. It is helpful to be able to quantify how well a prediction rule fits your data, because it can help you compare different models. This measurement is called the goodness-of-fit, and one way to quantify it is to use residuals to calculate the Mean Squared Error (MSE) of a prediction rule, and then use MSE to calculate the R^2 value of a prediction rule.

The R^2 value of a prediction rule quantifies the proportion of the variability in your response variable that is explained by the prediction rule. It is calculated with the following formula:

$$R^2 = 1 - \frac{MSE}{Variance(Y)}$$

R^2 is a value between 0 and 1. The higher the R^2 value, the better the fit of your prediction rule.

R^2 values can be used to compare prediction rules or to determine how much a predictor variable improves a prediction rule. The example below uses the prediction rules for the Prestige data set to demonstrate how to calculate this value and use it to compare prediction rules.

First, calculate the R^2 value for the linear prediction rule that uses education to predict prestige score of a job:

```
fit.edu <- lm(prestige ~ education, data = Prestige)
MSE.edu <- mean(fit.edu$residuals^ 2)
# Calculate variance:
y = Prestige$prestige
var.y = mean((y - mean(y))^2)
# Calculate Residuals:
R2.edu = 1 - MSE.edu/var.y
R2.edu

## [1] 0.7507872
```

The R^2 value for this model is 0.75, which means that 75% of the variation in prestige score across different jobs is explained by its linear association with education.

Next, calculate the R^2 value for the linear prediction rule that uses both **`education`** and log(**`income`**) to predict **`prestige`** score:

© 2021 Cornell University

```
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
```

The R^2 value for this model is 0.84, which means that 84% of the variation in prestige score across different jobs is explained by its association with education log(**income**).

Finally, you can compare the R^2 values from the two models to show that adding log(**income**) improved the fit of the model because the model that includes both predictor variables had a greater R^2 value by about 9%.

© 2021 Cornell University