Contents

1 Residual

A residual (or fitting error), on the other hand, is an observable estimate of the unobservable statistical error. Residual (or error) represents unexplained (or residual) variation after fitting a regression model. It is the difference (or left over) between the observed value of the variable and the value suggested by the regression model. Consider the previous example with men's heights and suppose we have a random sample of n people. The sample mean could serve as a good estimator of the population mean. Then we have:

The difference between the observed value of the dependent variable (y) and the predicted value () is called the residual (e). Each data point has one residual.

$$\text{Residual} = \text{Observed value} - \text{Predicted value}$$

$$e = y - \hat{y}$$

Both the sum and the mean of the residuals are equal to zero. .

The difference between the height of each man in the sample and the unobservable population mean is a statistical error, whereas The difference between the height of each man in the sample and the observable sample mean is a residual. Note that the sum of the residuals within a random sample is necessarily zero, and thus the residuals are necessarily not independent. The statistical errors on the other hand are independent, and their sum within the random sample is almost surely not zero.

1.1 Other uses of the word "error" in statistics

The use of the term "error" as discussed in the sections above is in the sense of a deviation of a value from a hypothetical unobserved value. At least two other uses also occur in statistics, both referring to observable prediction errors:

- Mean square error or mean squared error (abbreviated MSE) and root mean square error (RMSE) refer to the amount by which the values predicted by an estimator differ from the quantities being estimated (typically outside the sample from which the model was estimated).

- Sum of squared errors, typically abbreviated SSE or SSe, refers to the residual sum of squares (the sum of squared residuals) of a regression; this is the sum of the squares of the deviations of the actual values from the predicted values, within the sample used for estimation. Likewise, the sum of absolute errors (SAE) refers to the sum of the absolute values of the residuals, which is minimized in the least absolute deviations approach to regression.

1.2 Residual Plots

A residual plot is a graph that shows the residuals on the vertical axis and the independent variable on the horizontal axis. If the points in a residual plot are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data; otherwise, a non-linear model is more appropriate.

Below the table on the left shows inputs and outputs from a simple linear regression analysis, and the chart on the right displays the residual (e) and independent variable (X) as a residual plot.

The residual plot shows a fairly random pattern - the first residual is positive, the next two are negative, the fourth is positive, and the last residual is negative. This random pattern indicates that a linear model provides a decent fit to the data.

Below, the residual plots show three typical patterns. The first plot shows a random pattern, indicating a good fit for a linear model. The other plot patterns are non-random (U-shaped and inverted U), suggesting a better fit for a non-linear model.

In the next lesson, we will work on a problem, where the residual plot shows a non-random pattern. And we will show how to "transform" the data to use a linear model with nonlinear data.

2 Studentization

In statistics, a studentized residual is the quotient resulting from the division of a residual by an estimate of its standard deviation. Typically the standard deviations of residuals in a sample vary greatly from one data point to another even when the errors all have the same standard deviation, particularly in regression analysis; thus it does not make sense to compare residuals at different data points without first studentizing. It is a form of a Student's t-statistic, with the estimate of error varying between points.

This is an important technique in the detection of outliers. It is named in honor of William Sealey Gosset, who wrote under the pseudonym Student, and dividing by an estimate of scale is called studentizing, in analogy with standardizing and normalizing: see Studentization.

3 Residual Analysis for LME Models

In classical linear models model diagnostics have been become a required part of any statistical analysis, and the methods are commonly available in statistical packages and standard textbooks on applied regression. However it has been noted by several papers that model diagnostics do not often accompany LME model analyses.

Cite:Zewotir lists several established methods of analyzing influence in LME models. These methods include

- Cook's distance for LME models,

- likelihood distance,

- the variance (information) ration,

- the Cook-Weisberg statistic,

- the Andrews-Prebigon statistic.

3.1   LME REsiduals

Cox and Snell (1968, JRSS-B): general definition of residuals for models with single source of variability Hilden-Minton (1995, PhD thesis UCLA), Verbeke and Lesaffre (1997, CSDA) or Pinheiro and Bates (2000, Springer): extension to define three types of residuals that accommodate the extra source of variability present in linear mixed models, namely:

     i) Marginal residuals,

     predictors of marginal errors,

     ii) Conditional residuals,

$$be = yX\hat{\beta}Zbb = \hat{\sigma}Q\hat{y}$$

, predictors of conditional errors

$$e = yE[y|b] = yX\beta Zb$$

     iii) BLUP, Zbb, predictors of random effects,

$$Zb = E[y|b]E[y]$$

4   Residual Diagnostics

Consider a residual vector of the form $\hat{e} = \boldsymbol{PY}$, where $\boldsymbol{P}$ is a projection matrix, possibly an oblique projector. External studentization uses an estimate of Var that does not involve the ith observation.

     Externally studentized residuals are often preferred over studentized residuals because they have well known distributional properties in the standard linear models for independent data.

     Residuals that are scaled by the estimated variances of the responses are referred to as Pearson-type residuals.

     Standardization:

$$\frac{\hat{e}_i}{\sqrt{v_i}}$$

Studentization

$$\frac{\hat{e}_i}{\sqrt{\hat{v}_i}}$$

5   residuals.lme nlme- Extract lme Residuals

The residuals at level i are obtained by subtracting the fitted levels at that level from the response vector (and dividing by the estimated within-group standard error, if `type="pearson"`).

     The fitted values at level i are obtained by adding together the population fitted values (based only on the fixed effects estimates) and the estimated contributions of the random effects to the fitted values at grouping levels less or equal to i.

```
fm1 <- lme(distance ~ age + Sex,
data = Orthodont, random = ~ 1)
```

```
head(residuals(fm1, level = 0:1))
summary(residuals(fm1) /
residuals(fm1, type = "p"))

# constant scaling factor 1.432
```

6   Diagnostic Plots for Linear Models with `R`

Plot Diagnostics for an `lm` Object

    Six plots (selectable by `which`) are currently available:

1. a plot of residuals against fitted values,

2. a Scale-Location plot of *sqrt(— residuals —)* against fitted values,

3. a Normal Q-Q plot,

4. a plot of Cook's distances versus row labels,

5. a plot of residuals against leverages,

6. a plot of Cook's distances against leverage/(1-leverage).

By default, the first three and 5 are provided.

6.0.1   Residuals plots

lme allows to plot the residuals in the following ways:

```
res_lme=residuals(model_lme)
plot(res_lme)
qqnorm(res_lme)
qqline(res_lme)
plot(model_lme)
```

    When the `plot` function calls the model object, the residual plot is produced.

```
plot(JS.roy1, which=c(1) )
```

LME models assume that the residuals of the model are normally distributed. A Normal probability plot can be constructed to check this assumption. Commonly used `R` commands can be used to construct the plot.

```
qqnorm(resid(JS.roy1),pch="*",col="red")
qqline(resid(JS.roy1),col="blue")
```

```
table(dat$method[1:255])
##
##   J   S
## 255   0
table(dat$method[256:510])
##
##   J   S
##   0 255
```

```
library(predictMeans)
CookD(model, group=method, plot=TRUE, idn=5, newwd=FALSE)
```

```
> shapiro.test(resid(JS.roy1)[256:510])

Shapiro-Wilk normality test

data:  resid(JS.roy1)[256:510]
W = 0.9395, p-value = 9.503e-09
```

```
plot(roy.NLME, resid(., type = "p") ~ fitted(.) | method,
abline = 0, id=.05)
```

```
library(predictMeans)
CookD(model, group=method, plot=TRUE, idn=5, newwd=FALSE)
```

```
blood.red <- blood[!(blood$subject %in% c(68,78,80)),]
dim(blood.red)
# 27 observations should be removed.

blood.NLME.red <-lme(BP ~ method-1 , random=~1|subject,data = blood.red)
plot(blood.NLME.red, resid(., type = "p") ~ fitted(.) | method, abline = 0, id=.05)
```

```
> shapiro.test(resid(JS.roy1)[1:255])

Shapiro-Wilk normality test

data:  resid(JS.roy1)[1:255]
W = 0.9931, p-value = 0.2852
```

```
> shapiro.test(resid(JS.roy1)[256:510])

Shapiro-Wilk normality test

data:  resid(JS.roy1)[256:510]
W = 0.9395, p-value = 9.503e-09
```

```
data.frame( response = resid(JS.ARoy20091, type = "response"),
pearson  = resid(JS.ARoy20091, type = "pearson"),
normalized = resid(JS.ARoy20091, type = "normalized") )
```

```
response        pearson      normalized
1    -4.65805902 -0.761587227 -0.7615872269
2    -0.88701342 -0.145025661  0.0776238081
3    -5.16580898 -0.844603753 -0.8446037530
4     2.29041830  0.374480726  0.6450898404
5     7.87508366  1.287567009  1.2875670086
6    -6.57048659 -1.074266908 -1.5090772378
.........................................
```

7

For the J observations, the variance is 6.116252 whereas for the S observations, the denominator is 9.118144. (with the expected ratio of 1.490806)

```
> pearson %>%
+    as.numeric %>%
+    matrix(nrow=85) %>%
+    round(4)
[,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] -0.7616  0.2194  0.3829 -0.2983  0.3597 -0.0790
[2,] -0.1450  0.1820 -0.1450 -0.5014  0.1567  0.2663
[3,] -0.8446  0.4634  0.1364 -0.1630 -0.2727  0.1660
[4,]  0.3745 -0.2795 -0.2795 -0.2658 -0.2658  0.6115
[5,]  1.2876 -0.6744 -0.6744  0.8935 -0.0935 -0.8612
[6,] -1.0743  1.8687 -0.7473 -0.0383  0.2908 -0.3673

.............................................
```

We can plot the residuals against the fitted values, to assess the assumption of constant variance.

```
# standardized residuals versus fitted values
plot(JS.ARoy20091, resid(., type = "pearson") ~ fitted(.) ,
abline = 0, id = 0.05)
```

```
par(mfrow=c(1,2))
qqnorm((resid(JS.ARoy20091)[1:255]),
pch="*",col="red",
ylim=c(-40,40),
main="Method J")
qqline(resid(JS.ARoy20091)[1:255],col="blue")
qqnorm((resid(JS.ARoy20091)[256:510]),
pch="*",col="red",
ylim=c(-40,40),
main="Method S")
qqline(resid(JS.ARoy20091)[256:510],col="blue")
par(mfrow=c(1,1))
```

6.0.2   Residuals plots

When the `plot` function calls the model object, the residual plot is produced.

```
plot(JS.roy1, which=c(1) )
```

LME models assume that the residuals of the model are normally distributed. A Normal probability plot can be constructed to check this assumption. Commonly used R commands can be used to construct the plot.

```
qqnorm(resid(JS.roy1),pch="*",col="red")
qqline(resid(JS.roy1),col="blue")
```

```
table(dat$method[1:255])
##
##   J   S
## 255   0
table(dat$method[256:510])
##
##   J   S
##   0 255
```

```
plot(roy.NLME, resid(., type = "p") ~ fitted(.) | method,
abline = 0, id=.05)
```

```
data.frame( response = resid(JS.ARoy20091, type = "response"),
pearson  = resid(JS.ARoy20091, type = "pearson"),
normalized = resid(JS.ARoy20091, type = "normalized") )
```

```
response        pearson      normalized
1    -4.65805902 -0.761587227 -0.7615872269
2    -0.88701342 -0.145025661  0.0776238081
3    -5.16580898 -0.844603753 -0.8446037530
4     2.29041830  0.374480726  0.6450898404
5     7.87508366  1.287567009  1.2875670086
6    -6.57048659 -1.074266908 -1.5090772378
.......................................
```

9

For the J observations, the variance is 6.116252 whereas for the S observations, the denominator is 9.118144. (with the expected ratio of 1.490806)

```
> pearson %>%
+   as.numeric %>%
+   matrix(nrow=85) %>%
+   round(4)
[,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] -0.7616  0.2194   0.3829 -0.2983   0.3597 -0.0790
[2,] -0.1450  0.1820  -0.1450 -0.5014   0.1567  0.2663
[3,] -0.8446  0.4634   0.1364 -0.1630  -0.2727  0.1660
[4,]  0.3745 -0.2795  -0.2795 -0.2658  -0.2658  0.6115
[5,]  1.2876 -0.6744  -0.6744  0.8935  -0.0935 -0.8612
[6,] -1.0743  1.8687  -0.7473 -0.0383   0.2908 -0.3673
.......................................
```

We can plot the residuals against the fitted values, to assess the assumption of constant variance.

```
# standardized residuals versus fitted values
plot(JS.ARoy20091, resid(., type = "pearson") ~ fitted(.) ,
abline = 0, id = 0.05)
```

```
par(mfrow=c(1,2))
qqnorm((resid(JS.ARoy20091)[1:255]),
pch="*",col="red",
ylim=c(-40,40),
main="Method J")
qqline(resid(JS.ARoy20091)[1:255],col="blue")
qqnorm((resid(JS.ARoy20091)[256:510]),
pch="*",col="red",
ylim=c(-40,40),
main="Method S")
qqline(resid(JS.ARoy20091)[256:510],col="blue")
par(mfrow=c(1,1))
```

This code will allow you to make QQ plots for each level of the random effects. LME models assume that not only the within-cluster residuals are normally distributed, but that each level

of the random effects are as well. Depending on the model, you can vary the level from 0, 1, 2 and so on

```
qqnorm(JS.ARoy20091, ~ranef(.))

#  qqnorm(JS.ARoy20091, ~ranef(.,levels=1))
```

Figure 1:

```
data.frame( response = resid(JS.roy1, type = "response"),
pearson  = resid(JS.roy1, type = "pearson"),
normalized = resid(JS.roy1, type = "normalized") )
```

```
response         pearson     normalized
1    -4.65805902 -0.761587227 -0.7615872269
2    -0.88701342 -0.145025661  0.0776238081
3    -5.16580898 -0.844603753 -0.8446037530
4     2.29041830  0.374480726  0.6450898404
5     7.87508366  1.287567009  1.2875670086
6    -6.57048659 -1.074266908 -1.5090772378
.........................................
```

For the J observations, the variance is 6.116252 whereas for the S observations, the denominator is 9.118144. (with the expected ratio of 1.490806)

```
> pearson %>%
+    as.numeric %>%
+    matrix(nrow=85) %>%
+    round(4)
[,1]    [,2]     [,3]     [,4]     [,5]     [,6]
[1,] -0.7616  0.2194   0.3829 -0.2983   0.3597 -0.0790
[2,] -0.1450  0.1820  -0.1450 -0.5014   0.1567  0.2663
[3,] -0.8446  0.4634   0.1364 -0.1630  -0.2727  0.1660
[4,]  0.3745 -0.2795  -0.2795 -0.2658  -0.2658  0.6115
[5,]  1.2876 -0.6744  -0.6744  0.8935  -0.0935 -0.8612
[6,] -1.0743  1.8687  -0.7473 -0.0383   0.2908 -0.3673
.........................................
```

We can plot the residuals against the fitted values, to assess the assumption of constant variance.

```
# standardized residuals versus fitted values
plot(JS.roy1, resid(., type = "pearson") ~ fitted(.) ,
abline = 0, id = 0.05)
```

Figure 2:

- The **Scale-Location** plot, also called Spread-Location or S-L plot, takes the square root of the absolute residuals in order to diminish skewness (sqrt(—E—)) is much less skewed than — E — for Gaussian zero-mean E).

- The **Residual-Leverage** plot shows contours of equal Cook's distance, for values of cook.levels (by default 0.5 and 1) and omits cases with leverage one with a warning. If the leverages are constant (as is typically the case in a balanced aov situation) the plot uses factor level combinations instead of the leverages for the x-axis. (The factor levels are ordered by mean fitted value.)

```
par(mfrow=c(4,1))
plot(fittedmodel)
par(opar)
```

Residuals for LME Models

7   Diagnostic Tools for the nlme package

With the nlme package, the generic function `lme()` fits a linear mixed-effects model in the formulation described in Laird and Ware (1982) but allowing for nested random effects.

The within-group errors are allowed to be correlated and/or have unequal variances, which is very important in fitting the models for Roy's Tests

The nlme package has a limited set of diagnostic tools that can be used to assess the model fit. A review of the package manual is sufficient to get a sense of the package's capability in that regard.

8   Computation and Notation

with $\boldsymbol{V}$ unknown, a standard practice for estimating $\boldsymbol{X\beta}$ is the estime the variance components σ_j^2, compute an estimate for $\boldsymbol{V}$ and then compute the projector matrix A, $\boldsymbol{X\hat{\beta}} = \boldsymbol{AY}$.

Application to Method Comparison Studies