Chapter 3

Diagnostics and Remedial Measures for Simple Linear Regression

3.1 Diagnostics for Residuals

Before doing regression we must attempt to "verify" that the regression model assumptions from Section 1.4 hold. In theory you cannot actually, absolutely "verify" anything, let alone that the regression assumptions hold. According to the Popperian Principle of Falsification, one cannot conclusively affirm any hypothesis. Rather, one can only hope to bring forth sufficient evidence to reject a hypothesis. Therefore, when we attempt to "verify assumptions", what we really mean is we will attempt to show there is not sufficient evidence to conclude the assumptions are not met- we basically want to be satisfied to some degree that our data patterns are not inconsistent with the assumptions needed to build the model.

The assumptions in Section 1.4 can be summarized as

L: the conditional mean response $E[Y|X = x]$ is a LINEAR function of the predictor variable X;

I: the errors ε_i are INDEPENDENT random variables;

N: the errors ε_i are conditionally NORMAL random variables, given the predictor variable values;

E: the errors ε_i have EQUAL variance σ^2.

So if the LINE assumptions are met, the point pattern should look something like this:

Here are pictures indicating how the model assumptions might not be met in each case:

Remember it is very important that the regression model assumptions be met because the estimation and inferential methods we covered previously demand this. However, some of those methods are more sensitive to departure from the model assumptions than others:

1. All the hypothesis tests and intervals are very sensitive to even small departures from independence of the error terms.
2. All the hypothesis tests and intervals are sensitive to moderate departures from the equal variance assumption on the error terms.
3. The hypothesis tests and confidence intervals for β_0 and β_1 are fairly "robust" or forgiving when the normality assumption is violated.
4. The prediction intervals are very sensitive to departures from the normality assumption.

In this chapter we will learn ways to check on the assumptions as well as remedies to use when assumptions are not met. Much of our assumption checking involve looking at plots of the residuals (which are the observed error values) and running hypothesis tests about them. We will

1. plot residuals against the predictor values to check for LINEARITY;
2. plot residuals versus fits to check for EQUAL variance;
3. plot residuals against time or other variables to check for INDEPENDENCE;
4. make boxplots of residuals to softly check for NORMALITY;
5. make probability plots of the residuals to check for NORMALITY;
6. run Shapiro-Wilks tests and Anderson-Darling tests to check for NORMALITY;
7. run lack-of-fit tests to check for LINEARITY.

In addition to these things, we will also check for *outliers* (observations that lie far from the trend of the rest of the point pattern) and if there might be predictor variables that have been left out of the model.

Example 18 *Toluca Co., ALSM CH01TA01.*

```
vars <- c("lot.size", "work.hours")
data <- read.table("CH01TA01.txt", col.names = vars)
plot(data[,1], data[,2], main="Work Hours vs. Lot Size",
xlab="Lot Size", ylab="Work Hours")
out <- lm(data[,2] ~ data[,1])
abline(out)
summary(out)
residuals <- data[,2] - out$fit
head(residuals)
res <- out$residual
head(res)
windows()
plot(residuals ~ data[,1], xlab="Lot Size",
main="Residuals vs. Lot Size")
windows()
std.res <- residuals/sd(residuals)
plot(std.res ~ data[,1], main="Standardized
Residuals vs. Observed Lot Sizes", xlab="Lot Size")
run <- seq(1,nrow(data), 1)
data <- cbind(run, data)
windows()
plot(residuals ~ data$run, xlab="Run #", main =
"Residuals vs. Run #")
lines(data$run, residuals)
windows()
boxplot(residuals, horizontal=TRUE, outline=TRUE)
windows()
qqnorm(residuals)
qqline(residuals)
shapiro.test(residuals)
library(nortest)
ad.test(residuals)

####################

windows()
par(mfrow=c(2,3))
plot(data[,1], data[,2], main="Work Hours vs. Lot Size",
xlab="Lot Size", ylab="Work Hours")
plot(residuals ~ data[,1], xlab="Lot Size",
main="Residuals vs. Lot Size")
plot(std.res ~ data[,1], main="Standardized
Residuals vs. Observed Lot Sizes", xlab="Lot Size")
plot(residuals ~ data$run, xlab="Run #", main =
```



```
"Residuals vs. Run #")
lines(data$run, residuals)
boxplot(residuals, horizontal=TRUE, outline=TRUE, main
="Boxplot of Residuals")
qqnorm(residuals)
qqline(residuals)
```

Example 19 *Maps and Ridership Increase, ALSM CH03TA01.*

```
vars <- c("RidershipIncr", "MapsDistr")
data <- read.table("CH03TA01.txt", header=FALSE, col.names = vars)
par(mfrow=c(1,2))
plot(data[,1] ~ data[,2], main="Increase in
Ridership (in 1000s) vs. Maps
Distributed (in 1000s)",
xlab="Maps Distributed (in 1000s)",
ylab="Increase in Ridership (in 1000s")
out <- lm(data)
summary(out) ######Note the r^2 value is .8751
abline(out)
res <- out$residual
sqrt((sd(res)^2)*7/6)
plot(res ~ data[,2], main="Residuals vs. Maps Distributed (1000s)",
xlab = "Maps Distributed (in 1000s)", ylab="Residuals")
abline(a=0, b=0)  ####Slope a and intercept b set to 0
```

Example 20 *Brain and Body Weight for 65 Animals: BrainandBodyWeight.*

```
> data = read.csv("BrainandBodyWeight.csv", header = TRUE)
> plot(data$brain ~ data$body)
> text(data$body, data$brain, labels=data$Animal, cex=0, pos=4)
> windows()
> data.mod <- data[1:60,]
> plot(data.mod$brain ~ data.mod$body)
> text(data.mod$body, data.mod$brain, labels=data.mod$Animal, cex=0, pos=4)
> out <- lm(data.mod$brain ~ data.mod$body)
> summary(out)
> abline(out)
> residuals <- data.mod$brain - out$fit
> windows()
> plot(residuals ~ data.mod$body)
> animals <- data.mod[-50,]
> windows()
> out <- lm(animals$brain ~ animals$body)
> summary(out)
```



```
> plot(animals$brain ~ animals$body)
> abline(out)
> residuals <- animals$brain - out$fit
> windows()
> plot(residuals ~ animals$body)
> abline(a=0, b=0)
> text(animals$body, residuals, labels=animals$Animal, cex=0, pos=1)
```

Example 21 *Hang Times of Little Parachutes.*

```
library(car)
h <- c(5, 5, 5, 5, 5, 10, 10, 10, 10, 10,
15, 15, 15, 15, 15, 20, 20, 20, 20, 20)
t <- c(3.20, 3.21, 3.19, 3.25, 3.26,
+ 6.05, 6.79, 7.06, 7.11, 7.31, 9.03, 10.12,
+ 9.55, 10.33, 10.97, 13.01, 14.98, 14.55, 15.28, 17.33)
plot(h, t, main="Hang Times (sec) of Little Parachutes
vs. Height (m)", xlab="Drop Height (m)", ylab="Hang Time (sec)")
leveneTest(t ~ h)
hh <- c("a", "a", "a", "a", "a", "b", "b",
"b", "b", "b", "c", "c", "c", "c", "c",
"d", "d", "d", "d", "d")
leveneTest(t ~ hh)   #nonparametric test. normality not needed
########### Not as powerful as Bartlett if normality is satisfied
bartlett.test(t~hh)  #parametric test, assumes normality.
```

Example 22 *Production time: ALSM CH03PR18.* This data set contains production times in hours (14.28, 8.80, 12.49, ...) for 111 production runs along with their corresponding lot sizes (15, 9, 7, ...)

```
pt <- read.table("CH03PR18.txt")
names(pt) <- c("ProductionHrs", "LotSize")
head(pt)
plot(pt$ProductionHrs ~ pt$LotSize)
out<-lm(pt$ProductionHrs ~ pt$LotSize)
abline(out, col="blue")

## Suppose we split the explanatory values into
## three sets: < 10; [10, 15); >= 15.  We want to
## check to see if the variances of the residuals
## are the same (or differ somehow) across the
## categories.  I'll demonstrate a couple ways to
## accomplish this- first with a loop, then with
## the cut() function.
```



```
## Way 1......

i=1
while(i <= nrow(pt)){
ifelse(pt[i,2] <10, pt[i,3]<-"low", ifelse(pt[i,2] < 15,
pt[i,3] <-"med", pt[i,3]<-"high"))
i <- i+1
}
names(pt) <- c("ProductionHrs", "LotSize", "LotSizeCode")
head(pt)
library(car)
leveneTest(out$residuals ~ factor(pt$LotSizeCode))
windows()
boxplot(out$residuals ~ pt$LotSizeCode)

## Way 2......

Other <- cut(pt$LotSize, c(-Inf, 9.999, 14.999, Inf),
labels = c("L", "M", "H"))
asdf <- cbind(pt, Other)
leveneTest(out$residuals ~ asdf$Other)


##Breusch-Pagan...
## Square the residuals
sqres <- out$residuals^2
BPout <- lm(sqres ~ pt$LotSize)
summary(BPout)
windows()
plot(sqres ~ pt$LotSize)
abline(BPout)
```


One can use the von Neumann-Durbin-Watson test to check for auto correlation in the residuals. Let i run through some index set (perhaps one that indicates the order in which the observations were made). The test statistic is

$$0 \leq \frac{\sum_{i=1}^{n}(e_{i+1} - e_i)^2}{\sum_{i=1}^{n} e_i^2} \leq 4.$$

If the residuals have mean zero, the expectation of the denominator equals the variance of the residuals. Also, the expectation of the numerator is

A von Neumann-Durbin–Watson test statistic much less than 2 indicates evidence of positive serial correlation. Test statistic values greater than 2 indicate successive error terms tend to differ from each other, and so are likely negatively correlated, which can also be problematic. Then if there is little to no autocorrelation, we expect to see a test statistic near 2. The von Neumann-Durbin-Watson test is in the R package called "car".

Example 23 *New production.* A business implements a new production system. Open the dataset "Production.csv".

```
data <- read.csv("Production.csv", header=TRUE)
head(data)
out <- lm(data$Sales ~ data$Production)
summary(out)
plot(data$Sales ~ data$Production)
abline(out)
```



```
residuals <- data$Sales - out$fit
windows()
par(mfrow=c(3,1))
plot(residuals ~ data$Production)
abline(a=0, b=0)
plot(data$Sales ~ data$Day)
plot(residuals ~ data$Day)
library(car)
durbinWatsonTest(out)
```

There are many tests for independence, constant variance, normality, etc., including runs tests, tests for rank correlation, many kinds of goodness-of-fit (GOF) tests, including the Kolmogorov-Smirnov test (KS test), Shapiro-Wilks, Anderson-Darling, etc.

3.2 Lack of Fit Test

This test requires multiple observations (called *replicates*) for at least one explanatory variable value, and we typically need several replicates for the test to have any power.

Let

y_{ij} = jth response at the ith x-value;

$\overline{y}_i$ = average of y-values at the ith x-value;

$\hat{y}_{ij}$ = predicted response for the jth measurement the ith x-value.

c = number of distinct x-values.

There are at least two reasons the data don't fall on the line: (1) the model is not good; (2) the random variation is the culprit. This suggests decomposing total error (namely, SSE) into (1) a lack of fit portion and (2) a random error portion. If most of the error is due to lack of fit, it suggests we should try a different model (like a polynomial). To this end, we will attempt to decompose SSE into a sum of the *pure error sum of squares* ($SSPE$) and the *lack of fit sum of squares* ($SSLF$):

$$SSE = SSPE + SSLF,$$

where

$$\begin{aligned} SSPE &= \sum_i \sum_j (y_{ij} - \overline{y}_i)^2, \\ SSLF &= \sum_i \sum_j (\overline{y}_i - \hat{y}_{ij})^2. \end{aligned}$$

If $SSLF$ is close to SSE then we can assume there might be a better model choice (like a polynomial).

Source	DF	SS	MS	F
Regression	1	$SSR = \sum_{i=1}^{c}\sum_{j=1}^{n_i}(\hat{y}_{ij} - \overline{y})^2$	$MSR = \frac{SSR}{1}$	$F^* = \frac{MSR}{MSE}$
Residual	$n-2$	$SSE = \sum_{i=1}^{c}\sum_{j=1}^{n_i}(y_{ij} - \hat{y}_{ij})^2$	$MSE = \frac{SSE}{n-2}$	
Lack-of-Fit	$c-2$	$SSLF = \sum_{i=1}^{c}\sum_{j=1}^{n_i}(\overline{y}_i - \hat{y}_{ij})^2$	$MSLF = \frac{SSLF}{c-2}$	$F^* = \frac{MSLF}{MSPE}$
Pure Error	$n-c$	$SSPE = \sum_{i=1}^{c}\sum_{j=1}^{n_i}(y_{ij} - \overline{y}_i)^2$	$MSPE = \frac{SSPE}{n-c}$	
Total	$n-1$	$SSTO = \sum_{i=1}^{c}\sum_{j=1}^{n_i}(y_{ij} - \overline{y})^2$		

It turns out

$$\begin{aligned} E[MSLF] &= \sigma^2 + \frac{\sum_{i=1}^{c} n_i(\mu_i - (\beta_0 + \beta_1 x_i))^2}{c-2}, \\ E[MSPE] &= \sigma^2. \end{aligned} \tag{3.1}$$

If the null hypothesis is true (that is, there exists a linear relationship between the variables), then $\mu_i = \beta_0 + \beta_1 x_i$, and the term on the right in (3.1) is 0. In this case we would expect $MSLF/MSPE$ to be close to 1. It turns out then that under the null hypothesis, $MSLF/MSPE$ behaves according to an F distribution with $c-2$ numerator and $n-c$ denominator degrees of freedom.

Example 24 Enter these data into R:

```
x <- c(1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4, 6, 6,
7, 7, 7, 10, 10, 10, 10)
y <- c(4, 6, 8, 12, 13, 12, 14, 15, 15.5, 15,
15.9, 16, 16.5, 15, 14.5, 11, 3, 4,4.5, 8)
plot(x,y)
model1 <- lm(y~x)
anova(model1)
library(alr3)
pureErrorAnova(model1)
abline(model1, lwd=2, col="blue")
```

You can tell from the plot that a straight-line model is probably no good and we should probably fit a parabola. Also the output gives $SSE = 385.47$ and $SSLF = 338.07$, indicating most of the residual error is due to a bad model (rather than $SSPE$). Note indeed that $SSE = SSLF + SSPE$.

Change the response values and redo:

```
y <- c(2.8, 3.2, 5.5, 6.3, 6, 8.7, 8.9, 9,
9.3, 12.2, 11.6, 17.7, 18.1, 21.1, 21.8, 21,
```



```
29.5, 30, 30.5, 30.8)
windows()
plot(x,y)
model2 <- lm(y~x)
anova(model2)
pureErrorAnova(model2)
abline(model2, lwd=2, col="blue")
```

Note the $SSLF$ is very small compared to the $SSPE$ and SSE.

Note the p-value was very small for the lack-of-fit test for the first data set, but not significantly small for the lack-of-fit test on the second. These p-values correspond well to the plots.

3.3 Transformations

Sometimes when it seems a good linear relationship does not exist between two variables, there might actually exist a good linear relationship between a function of one of the variables and a function of the other. That is, by transforming one or more of the variables might result in uncovering a linear pattern to the data by "correcting" for skewness of the errors (to make them more normally distributed), unequal error variances, and general nonlinearity of the data pattern. It is important to realize after transforming variables, that the linear model will attempt to describe linear relationships between the transformed variables.

Example 25 *Brain and body weight.* Open BrainandBodyWeight.csv data.

```
animals <- read.csv("BrainandBodyWeight.csv", header=TRUE)
animals
animals <- animals[1:60,] ##Get rid of dinosaurs...
plot(animals$brain ~ animals$body)
animals <- animals[-50,]  ## Get rid of humans...
plot(animals$brain ~ animals$body)
text(animals$body, animals$brain, labels=animals$Animal, cex=.7, pos=4)
out <- lm(animals$brain ~ animals$body)
summary(out)
animals$logbrain <- log(animals$brain)
animals$logbody <- log(animals$body)
logout <- lm(animals$logbrain ~ animals$logbody)
summary(logout)
windows()
plot(animals$logbrain ~ animals$logbody, main="Log of Brain
Mass (g) vs. Log of Body Mass (kg)",
xlab="Log of Body Mass (kg)", ylab="Log of Brain Mass (g)")
abline(logout, col="blue")
pureErrorAnova(logout)
windows()
plot(animals$logbody, logout$residual)
windows()
plot(animals$logbody, logout$residual)
hist(logout$residual)
shapiro.test(logout$residual)
```

So we can say there seems to exist a strong positive association between the natural logs of the brain and body weights of animals:

$$\widehat{\log(\text{brainweight})} = 0.73528 \log(\text{bodyweight}) + 2.11392$$

Some common functions for attempting to try to stabilize variance:

$$\sqrt{y}, \quad \log y, \quad \log_{10}(y+1), \quad \sin^{-1}\sqrt{y}, \quad 1/y.$$

Which function or functions to use in transforming data takes experience.

The *Box-Cox* transformation is given by

$$y_i' = \frac{(y_i^\lambda - 1)}{\lambda},$$

where the λ value is chosen to maximize the log-likelihood function of a normal distribution (taken at its maximum in) for the transformed data:
Note that $y_i' \to_{\lambda \to 0} \log(y)$, and so we take the Box-Cox transformation to be equivalent to the natural log transformation when $\lambda = 0$. The Box-Cox transformation can be applied to one or all variables in the model. Once λ is found, we often simply just transform data from $y \to y^\lambda$ because subtracting 1 and dividing by λ would not change the overall shape of the data. Note that several of the transformations in (3.2) are actually Box-Cox transformations.

Box-Cox transformations can only be performed on non-negative data. If the data are less than zero, you can shift the data set by a constant and then apply Box-Cox.

Example 26 *Trees.* Open the trees data in R by simply typing "trees" at the prompt (without the quotes). This data set provides measurements of the girth, height and volume of timber in 31 felled black cherry trees. Girth is the diameter of the tree (in inches) measured at 4.5 feet above the ground.

```
trees
attach(trees)
reg1 <- lm(Volume ~ Height + Girth)
##### Here is our first model with two predictors!
plot(Girth, rstandard(reg1))
par(mfrow=c(2,1))
hist(reg1$resid)
qqnorm(reg1$resid)
qqline(reg1$resid)
library(moments)     ###### For skewness function
skewness(reg1$resid)
library(MASS)        ######  For Box Cox transformation
trans <- boxcox(Volume ~ Height + Girth)
trans
lambda <- trans$x
loglh <- trans$y
boxcox <- cbind(lambda, loglh)
```



```
boxcox[order(-loglh),] #Using log-likelihood to optimize lambda
reg2 <- lm (Volume^.3030303 ~ Height + Girth)
plot(reg2)
par(mfrow = c(2,1))
hist(reg2$resid)
qqnorm(reg2$resid)
qqline(reg2$resid)
skewness(reg2$resid)
windows()
par(mfrow=c(2,1))
plot(Girth, rstandard(reg1))
plot(Girth, rstandard(reg2))
summary(reg2)
```


3.4 Homework

1. Open the GPA/ACT data from CH03PR03.txt. This data set is the data set from Chapter 1 on GPA vs. ACT, but includes observations on two additional variables, namely intelligence test scores (third column) and high school class rank percentile (fourth column). We want to know which of the three explanatory variables (ACT, intelligence test score, high school class rank percentile) can best be used to make a linear model for predicting GPA. So you will build and compare three simple linear regression models for

   1. GPA vs. ACT;
   2. GPA vs. intelligence test score;
   3. GPA vs. class rank percentile.

   So for each of these three cases, do the following:

   (a) obtain the linear model (lm(y $\sim$ x)) and output (summary()) and compare the r^2 values;

   (b) make scatter plots that include the regression lines. Identify any potential outliers and influential observations. Decide whether or not to remove them before moving on.

   (c) check for normality of the residuals with Shapiro-Wilk tests (shapiro.test());

   (d) make boxplots and histograms of the residuals to help check for normality;

   (e) make normal probability plots for the residuals (to check for normality);

   (f) Split the data set into two groups: students with ACT scores less than 26 and students with ACT scores at least 26. Run Levene's test for equality of variance of the residuals (leveneTest() requires the car package) on the response for these two groups. Remark on what you observe. The command subset() can be useful for splitting data sets.

   (g) Repeat part (f) for the intelligence test score model, splitting at < 120 and ≥ 120. Do the residual variances for the two groups appear to differ?

   (g) plot the residuals vs. the predictor variable values to check for equality of variance;

   (h) Remark on which of the three explanatory variables seems to be the most useful in building a linear model for predicting first-year GPA.

2. Adapted from ALSM 3.15. A chemist wanted to model the evolution of a solution concentration over time. To do this, she randomly assigned three solutions to measure after one hour, three solutions to measure after three hours, three to measure after five hours, three to measure after seven horus, and three to measure after nine hours. The data are in CH03PR15.

   (a) Find the equation of the least-squares regression line.

   (b) Run the F-test for lack of linear fit. Use the significance level $\alpha = 0.025$. State your p-value and provide your conclusion.

   (c) When a lack-of-linear fit test indicates there is a lack of linear fit, does it suggest exactly what kind of function would be appropriate? Explain.

3. Adapted from ALSM 3.16. Use the data from the previous problem.

   (a) Make a scatterplot of the data, with concentration as the response variable. Based on the scatterplot, what kind of data transformation do you suggest to adjust for the non-constant variance and/or non-linearity?

   (b) Use the log-likelihood method in R we used for selecting the Box-Cox λ, and apply the transformation to the data. Then build a new regression model (using lm()) for the transformed data, make a fitted-line plot, and discuss how the new relationship compares with the old one.

   (c) Apply the log_{10} transformation and get the new regression line equation. Plot this model on a scatterplot of the transformed data. Compare the results of the Box-Cox transformation with those of the log_{10} transformation. Which do you like better and why?

   (d) Plot the residuals against the fitted values. Also make a normal probability plot. What do these plots indicate?

   (e) Express your estimated regression functions in the original units.

4. Adapted from ALSM 3.17. A marketing manager studied annual product sales figures over a ten year period. The data (years and sales in thousands of units) are in the file CH03PR17.

   (a) Make a scatterplot. Is the linearity assumption reasonable?

   (b) Apply the maximum likelihood Box-Cox method (like we did in the *Trees* example) to get an appropriate power transformation of the response (sales). What is the value of SSE in this case?

   (c) Try using the square-root transformation and get a new regression line.

(d) Plot the regression line from the previous part on a scatterplot of the transformed data. Does this line seem to fit the transformed data well?

(e) Make a plot of the residuals vs. fits. Also make a normal probability plot. What do these plots indicate for your transformed data?

(f) Express the regression models in the original units.

5. The Blaisdell Company wanted to use industry sales to predict its sales. Adjusted quarterly sales data for 1998 - 2002 are in the ALSM data set CH12TA02. The first column of data are observations of Blaisdell's sales, and the second column contain industry sales. The *very* first column, the one with the row numbers, is a time index: 1 means first quarter of 1998, 2 means 2nd quarter of 1998, etc.

(a) Make a scatter plot of company sales using industry sales as the predictor (with R of course). Describe the apparent relationship between the two variables.

(b) Make a scatter plot of company sales versus the time index. You might have to create a new column for the time values or figure out how to reference the row numbers in R.

(c) Use R to run a Neumann-Durbin-Watson to check for autocorrelation of company sales over time:

$$\begin{aligned} H_0: &\quad \rho = 0, \\ H_a: &\quad \rho \neq 0. \end{aligned}$$

(d) How would you suggest to proceed in modelling the relationship between these variables?

6. Complete the following lack-of-fit ANOVA table:

Source	DF	SS	MS	F^*	p-value
Regression	??	34.783	??	??	??
Residual	??	??	??		
Lack-of-Fit	5	??	??	??	??
Pure Error	??	2.110	??		
Total	21	41.85			

