Introduction to the Practice of Statistics using R: Chapter 16

Ben Baumer Nicholas J. Horton*

April 8, 2013

Contents

Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Sixth Edition of *Introduction to the Practice of Statistics* (2002) by David Moore, George McCabe and Bruce Craig. More information about the book can be found at `http://bcs.whfreeman.com/ips6e/`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/ips6e`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

```
> install.packages('mosaic')              # note the quotation marks
```

The `#` character is a comment in R, and all text after that on the current line is ignored.

Once the package is installed (one time only), it can be loaded by running the command:

*Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> require(mosaic)
```

This needs to be done once per session.
We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3)
```

The specific goal of this document is to demonstrate how to replicate the analysis described in Chapter 10: Inference for Regression.

1 Simple Linear Regression

The first example from Chapter 10 is 10.4 (page 566), which assesses fuel economy for 60 cars.

```
> fuel = read.csv("http://math.smith.edu/ips6eR/ch10/eg10_001.csv")
> head(fuel)

  MILES  MPG  MPH LOGMPH  RESID
1 12457 14.8 18.6   2.92 -0.421
2 12658 15.1 19.5   2.97 -0.493
3 13439 17.5 24.2   3.19  0.206
4 13518 14.3 17.9   2.88 -0.619
5 13799 15.9 21.2   3.05 -0.352
6 14097 17.9 32.0   3.47 -1.594
```

In this case we are building a model for MPG as a function of $LOGMPG$, which is a pre-computed variable. Output similar to that shown in Figure 10.5 can be produced by applying the `summary()` command to an `lm` object.

```
> fm1 = lm(MPG ~ LOGMPH, data=fuel)
> summary(fm1)


Call:
lm(formula = MPG ~ LOGMPH, data = fuel)

Residuals:
   Min     1Q Median     3Q    Max
-3.717 -0.519  0.112  0.659  2.149

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -7.796      1.155   -6.75  7.7e-09 ***
LOGMPH         7.874      0.354   22.24  < 2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1 on 58 degrees of freedom
Multiple R-squared: 0.895,Adjusted R-squared: 0.893
F-statistic:  494 on 1 and 58 DF,  p-value: <2e-16
```

Note that R can compute the same model without using the precomputed variables, by applying the `log()` function to the MPH variables on-the-fly.

```
> fm1a = lm(MPG ~ log(MPH), data=fuel)
> summary(fm1a)


Call:
lm(formula = MPG ~ log(MPH), data = fuel)

Residuals:
   Min     1Q Median     3Q    Max 
-3.717 -0.519  0.112  0.659  2.149 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -7.796      1.155   -6.75  7.7e-09 ***
log(MPH)       7.874      0.354   22.24  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1 on 58 degrees of freedom
Multiple R-squared: 0.895,Adjusted R-squared: 0.893
F-statistic:  494 on 1 and 58 DF,  p-value: <2e-16
```

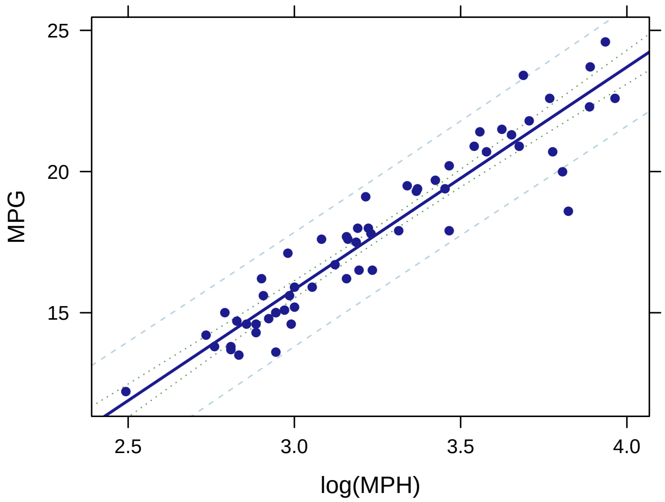
Like other statistical software packages, R performs a t-test for the null hypothesis that $\beta_i = 0$ for all coefficients β_i present in the model. The third column of the `summary()` output (labeled `t value`) gives the t-statistic, and the fourth column gives the corresponding p-value. Confidence intervals can be retrieved using the `confint()` command, which by default returns a 95% confidence interval.

```
> confint(fm1)

             2.5 % 97.5 %
(Intercept) -10.11  -5.48
LOGMPH        7.17   8.58
```

Confidence intervals for the mean response, as well as prediction intervals for future observations, can be plotted using the `panel.lmbands` argument to `xyplot()`. The following plot is a mashup of Figure 10.9 (page 573) and Figure 10.10 (page 575).

```
> xyplot(MPG ~ log(MPH), panel=panel.lmbands, data=fuel)
```



To retrieve the actual values, we can apply the `predict()` command to our regression model object, and specify whether we want confindence intervals or prediction intervals.

```
> # only show the first six rows for clarity
> head(predict(fm1, interval="confidence"))

   fit  lwr  upr
1 15.2 14.9 15.6
2 15.6 15.3 15.9
3 17.3 17.0 17.6
4 14.9 14.6 15.3
5 16.3 16.0 16.5
6 19.5 19.2 19.8

> # only show the first six rows for clarity
> head(predict(fm1, interval="predict"))

Warning:  Predictions on current data refer to _future_ responses

   fit  lwr  upr
1 15.2 13.2 17.3
2 15.6 13.6 17.6
3 17.3 15.3 19.3
4 14.9 12.9 17.0
5 16.3 14.2 18.3
6 19.5 17.5 21.5
```

2 More Detail about Simple Linear Regression

2.1 The ANOVA F-test

An ANOVA table similar to the one shown in Figure 10.12 (page 583) can be produced by applying the `anova()` command to a regression model object.

```
> anova(fm1)

Analysis of Variance Table

Response: MPG
          Df Sum Sq Mean Sq F value Pr(>F)
LOGMPH     1    494     494     494 <2e-16 ***
Residuals 58     58       1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.2 Inference for Correlation

We can test for zero correlation using the `cor.test()` command. In Example 10.22, a t-test for non-zero correlation is conducted between the MPG and $LOGMPH$ of 60 cars

```
> with(fuel, cor.test(MPG, LOGMPH))


Pearson's product-moment correlation

data:  MPG and LOGMPH
t = 22.2, df = 58, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.911 0.968
sample estimates:
  cor
0.946
```