STAT 8051 HW 3

Jingxiang Li

October 2, 2014

Problem 6.7

(Data file: `fuel2001`) With the fuel consumption data, consider the following two models in WilkinsonRogers notation:

```
fuel ~ Tax + Dlic + Income + log(Miles)          (6.22)
fuel ~ log(Miles) + Income + Dlic + Tax          (6.23)
```

These models are of course the same, as they only differ by the order in which the regressors are written.

Problem 6.7.1

Show that the Type I anova for (6.22) and (6.23) are different. Provide an interpretation of each of the tests.

Solution

```
require(alr4)

data <- fuel2001
data$Dlic <- data$Drivers / data$Pop
data$Fuel <- 1000 * data$FuelC / data$Pop

m1 <- lm(Fuel ~ Tax + Dlic + Income + log(Miles), data = data)
m2 <- lm(Fuel ~ log(Miles) + Income + Dlic + Tax, data = data)
anova(m1)

## Analysis of Variance Table
##
## Response: Fuel
##            Df Sum Sq Mean Sq F value  Pr(>F)
## Tax         1  26635   26635    6.33  0.0155 *
## Dlic        1  79378   79378   18.85 7.7e-05 ***
## Income      1  61408   61408   14.58  0.0004 ***
## log(Miles)  1  34573   34573    8.21  0.0063 **
## Residuals  46 193700    4211
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m2)

## Analysis of Variance Table
##
```

```
## Response: Fuel
##             Df Sum Sq Mean Sq F value  Pr(>F)
## log(Miles)   1  70478   70478   16.74 0.00017 ***
## Income       1  49996   49996   11.87 0.00123 **
## Dlic         1  63256   63256   15.02 0.00034 ***
## Tax          1  18264   18264    4.34 0.04287 *
## Residuals   46 193700    4211
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The above anova results are based on Type I anova. Note that all the main effects of predictors are significant in two models, however, the levels of significance are different in two models. suggesting that the order of predictors influence the result of Type I anova.

Problem 6.7.2

Show that the Type II anova is the same for the two models. Which of the Type II tests are equivalent to Type I tests?

Solution

```
Anova(mod = m1, type = 2)

## Anova Table (Type II tests)
##
## Response: Fuel
##             Sum Sq Df F value  Pr(>F)
## Tax          18264  1    4.34 0.04287 *
## Dlic         56770  1   13.48 0.00063 ***
## Income       32940  1    7.82 0.00751 **
## log(Miles)   34573  1    8.21 0.00626 **
## Residuals   193700 46
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Anova(mod = m2, type = 2)

## Anova Table (Type II tests)
##
## Response: Fuel
##             Sum Sq Df F value  Pr(>F)
## log(Miles)   34573  1    8.21 0.00626 **
## Income       32940  1    7.82 0.00751 **
## Dlic         56770  1   13.48 0.00063 ***
## Tax          18264  1    4.34 0.04287 *
## Residuals  193700 46
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice that the Type II anova is the same for the two models.

In the first model, the Type II test of `log(Miles)` is equal to Type I; in the second model, the Type II test of `Tax` is equal to the Type I test.

Problem 6.8

Show that the overall F-test for multiple regression with an intercept can be written as

$$F = (\frac{n-p'}{p})\frac{R^2}{1-R^2}$$

Where R^2 is the proportion of variability explained by the regression. Thus, the F-statistic is just a transformation of R^2.

Solution

Following the definition of R^2 and F statistic, we have

$$R^2 = 1 - \frac{RSS}{SYY}$$

$$F = \frac{(SYY - RSS)/p}{RSS/(n-p-1)}$$

Then

$$\begin{aligned} F &= \frac{SYY - RSS}{RSS} \cdot \frac{n-p-1}{p} \\ &= \frac{1 - \frac{RSS}{SYY}}{\frac{RSS}{SYY}} \cdot \frac{n-p-1}{p} \\ &= \frac{R^2}{1-R^2} \cdot \frac{n-p-1}{p} \end{aligned}$$

Q.E.D.

Problem 6.9

(Data file: `Cakes`) For the cakes data in Section 5.3.1, we fit the full second-order model,

$$E(Y|X_1 = x_1, X_2 = x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_2 + \beta_4 x_2^2 + \beta_5 x_1 x_2$$

Compute and summarize the following three hypothesis tests.

$$\text{NH: } \beta_5 = 0 \text{ vs. AH: } \beta_5 \neq 0$$
$$\text{NH: } \beta_2 = 0 \text{ vs. AH: } \beta_2 \neq 0$$
$$\text{NH: } \beta_1 = \beta_2 = \beta_5 = 0 \text{ vs. AH: Not all } 0$$

Solution

```
data <- cakes
m_full <- lm(Y ~ (X1 + X2) ^ 2 + I(X1 ^ 2) + I(X2 ^ 2), data = data)
summary(m_full)
```

```
##
## Call:
## lm(formula = Y ~ (X1 + X2)^2 + I(X1^2) + I(X2^2), data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.491 -0.308  0.020  0.266  0.545
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.20e+03   2.42e+02   -9.13  1.7e-05 ***
## X1           2.59e+01   4.66e+00    5.56  0.00053 ***
## X2           9.92e+00   1.17e+00    8.50  2.8e-05 ***
## I(X1^2)     -1.57e-01   3.94e-02   -3.98  0.00408 **
## I(X2^2)     -1.20e-02   1.58e-03   -7.57  6.5e-05 ***
## X1:X2       -4.16e-02   1.07e-02   -3.88  0.00465 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.429 on 8 degrees of freedom
## Multiple R-squared:  0.949,Adjusted R-squared:  0.917
## F-statistic: 29.6 on 5 and 8 DF,  p-value: 5.86e-05
```

First Let's deal with the first two tests. Note that both of them are one coefficient two-sided t-test. Using R, it's easy to see the t-test result in the summary of full regression model.

As the result, for the first hypothesis test, we see a tiny p-value of 0.0047, suggesting that we should reject the null hypothesis, β_5 should be non-zero.

Similarly, for the second test, we see a p-value of 0.0041, suggesting that β_2 should be non-zero, the null hypothesis should be rejected.

```
m_test <- lm(Y ~ X2 + I(X2 ^ 2), data = data)
anova(m_test, m_full)
```

```
## Analysis of Variance Table
##
## Model 1: Y ~ X2 + I(X2^2)
## Model 2: Y ~ (X1 + X2)^2 + I(X1^2) + I(X2^2)
##   Res.Df   RSS Df Sum of Sq    F  Pr(>F)
## 1     11 11.47
## 2      8  1.47  3        10 18.1 0.00063 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Then we use anova to solve the last hypothesis test. the result shows a very tiny p-value, suggesting that the null hypothesis should be rejected, β_1, β_2 and β_5 are not all zero.

Problem 6.14

Testing for lack-of-fit (Data file: `MinnLand`) Refer to the Minnesota farm sales data introduced in Problem 5.4.

Problem 6.14.1

Fit the regression model `log(acrePrice)` ∼ `year` via ols, where `year` is not a factor, but treated as a continuous predictor. What does this model say about the change in price per acre over time? Call this model A.

Solution

```
data <- MinnLand
m_A <- lm(log(acrePrice) ~ year, data = data)
coef(m_A)
```

```
## (Intercept)        year
##   -193.8760      0.1005
```

model A says the `log(acrePrice)` increase 0.1005 per year on average.

Problem 6.14.2

Fit the regression model via `log(acrePrice)` ∼ `1 + fyear` via ols, where `fyear` is a factor with as many levels are there are years in the data, including the intercept in the model. What does this model say about the change in price per acre over time? Call this model B. (Hint: `fyear` is not included in the data file. You need to create it from the variable `year`.)

Solution

```
data$fyear <- factor(data$year)
m_B <- lm(log(acrePrice) ~ fyear, data = data)
coef(m_B)
```

```
## (Intercept)   fyear2003   fyear2004   fyear2005   fyear2006   fyear2007   fyear2008
##     7.27175    -0.00155     0.14794     0.36026     0.39392     0.47682     0.68364
##   fyear2009   fyear2010   fyear2011
##     0.71407     0.75733     0.72071
```

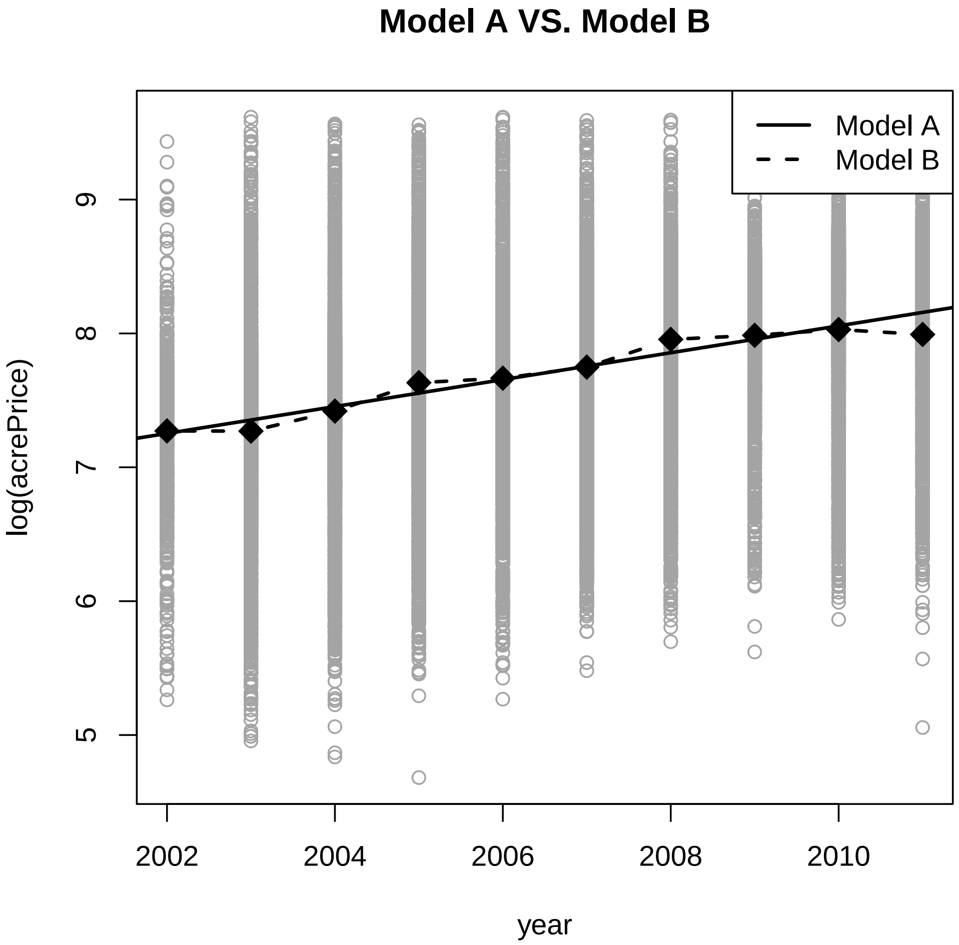
In this model the intercept is the mean value of `log(acrePrice)` in 2002, and each coefficient of any other years represents the change of `log(acrePrice)` compared with that of 2002.

Problem 6.14.3

Show that model A is a special case of model B, and so a hypothesis test of NH : model A versus AH : model B is reasonable.

Solution

```
mean_value <- tapply(X = log(data$acrePrice), INDEX = data$year, FUN = mean)
plot(x = data$year, y = log(data$acrePrice), col = "#A4A4A4",
     xlab = "year", ylab = "log(acrePrice)", main = "Model A VS. Model B")
lines(x = 2002:2011, y = mean_value, type = 'b', pch = 18, cex = 2, lwd = 2, lty = 2)
abline(coef(m_A), lwd = 2, lty = 1)
legend("topright", c("Model A", "Model B"), lty = c(1,2), lwd = c(2,2), bg = "white")
```



We show two fitted regression model in the graph above. Note that model B (dashed line in the graph above) can be viewed as a bunch of broken lines connected with each other on the mean value of response for each year; and model A (solid line in the graph above) can be viewed as a non-broken line go through all the points. Thus model A is just the special case of model B, if the mean values of response in each year are all on an unbroken line.

Problem 6.14.4

A question of interest is whether or not model A provides an adequate description of the change in `log(acrePrice)` over time. The hypothesis test of NH : model A versus AH : model B addresses this question, and it can be called a *lack-of-fit* test for model A. Perform the test and summarize results.

Solution

It's easy to find that $df_A = 18698$, $df_B = 18690$, $RSS_A = 8666.9333$, $RSS_B = 8579.2475$. Hence we can construct a F-statistic:

$$F = \frac{(RSS_A - RSS_B)/(df_A - df_B)}{RSS_B/df_B} \sim \quad F(df_A - df_B, df_B)$$

So that we can use anova to do the *lack-of-fit* test.

```
anova(m_A, m_B)
```

```
## Analysis of Variance Table
##
## Model 1: log(acrePrice) ~ year
## Model 2: log(acrePrice) ~ fyear
##   Res.Df  RSS Df Sum of Sq    F Pr(>F)
## 1  18698 8667
## 2  18690 8579  8      87.7 23.9 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that the p-value is very small, seems suggesting that we should reject NH, which means model A is lack of fit.

However, the small p-value may result not from the model, but the huge sample size. Note that the sample size in this problem is 18700, and hence the power of this F-test is so big that even small differences in RSS will result in large significance. In addition, from the graph we can see that, this two models are very close to each other. Therefore in this problem, though the F-test suggests statistical significance, it can't be translated to practical significance.

Problem 8.2

(Data file: `stopping`) We reconsider the stopping distance data used in Problem 7.6.

Problem 8.2.1

Using `Speed` as the only regressor, find an appropriate transformation for `Distance` that can linearize this regression.

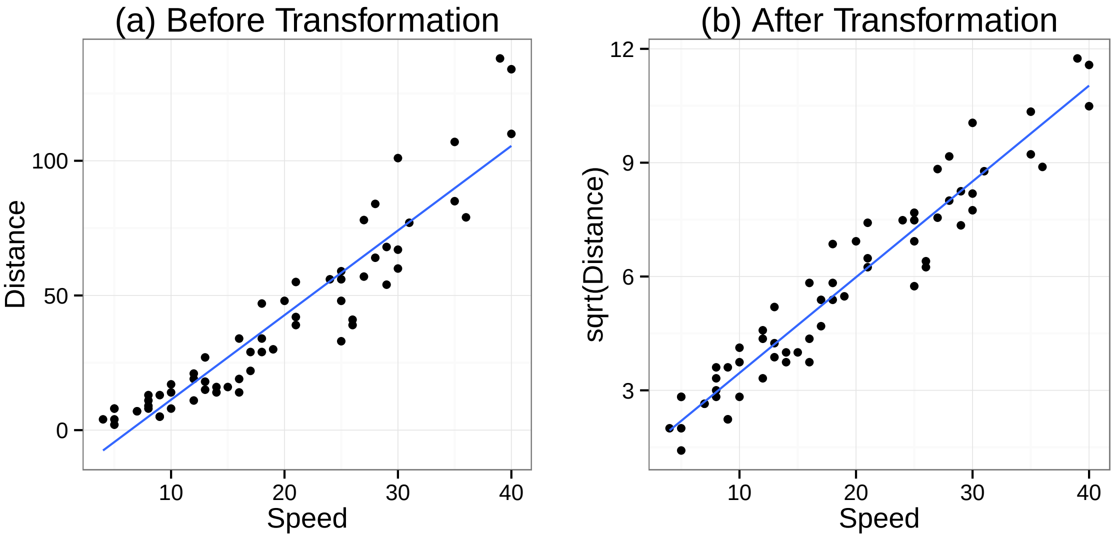
Solution

```
require(package = "gridExtra")
require(package = "ggplot2")
data <- stopping

p1 <- ggplot(data, aes(x = Speed, y = Distance))
p1 <- p1 + geom_point()
p1 <- p1 + geom_smooth(formula = y ~ x, method = "lm", se = FALSE)
p1 <- p1 + theme_bw() + theme(text = element_text(size = 14))
p1 <- p1 + ggtitle("(a) Before Transformation")

p2 <- ggplot(data, aes(x = Speed, y = sqrt(Distance)))
p2 <- p2 + geom_point()
p2 <- p2 + geom_smooth(formula = y ~ x, method = "lm", se = FALSE)
p2 <- p2 + theme_bw() + theme(text = element_text(size = 14))
p2 <- p2 + ggtitle("(b) After Transformation")

grid.arrange(p1, p2, ncol=2)
```



We first draw the Scatterplot of `Distance` against `Speed`, as graph (a) above, and see that the relationship between response and predictor is approximately quadratic. Then let $\sqrt{\texttt{Speed}}$

as the new response and draw a new scatterplot (b), we see that the relationship between new response and predictor is almost linear. Thus the transformation we need should be square root.

Problem 8.2.2

Using Distance as the response, transform the predictor Speed using a power transformation with each $\lambda \in \{-1, 0, 1\}$ and show that none of these transformations is adequate.

Solution

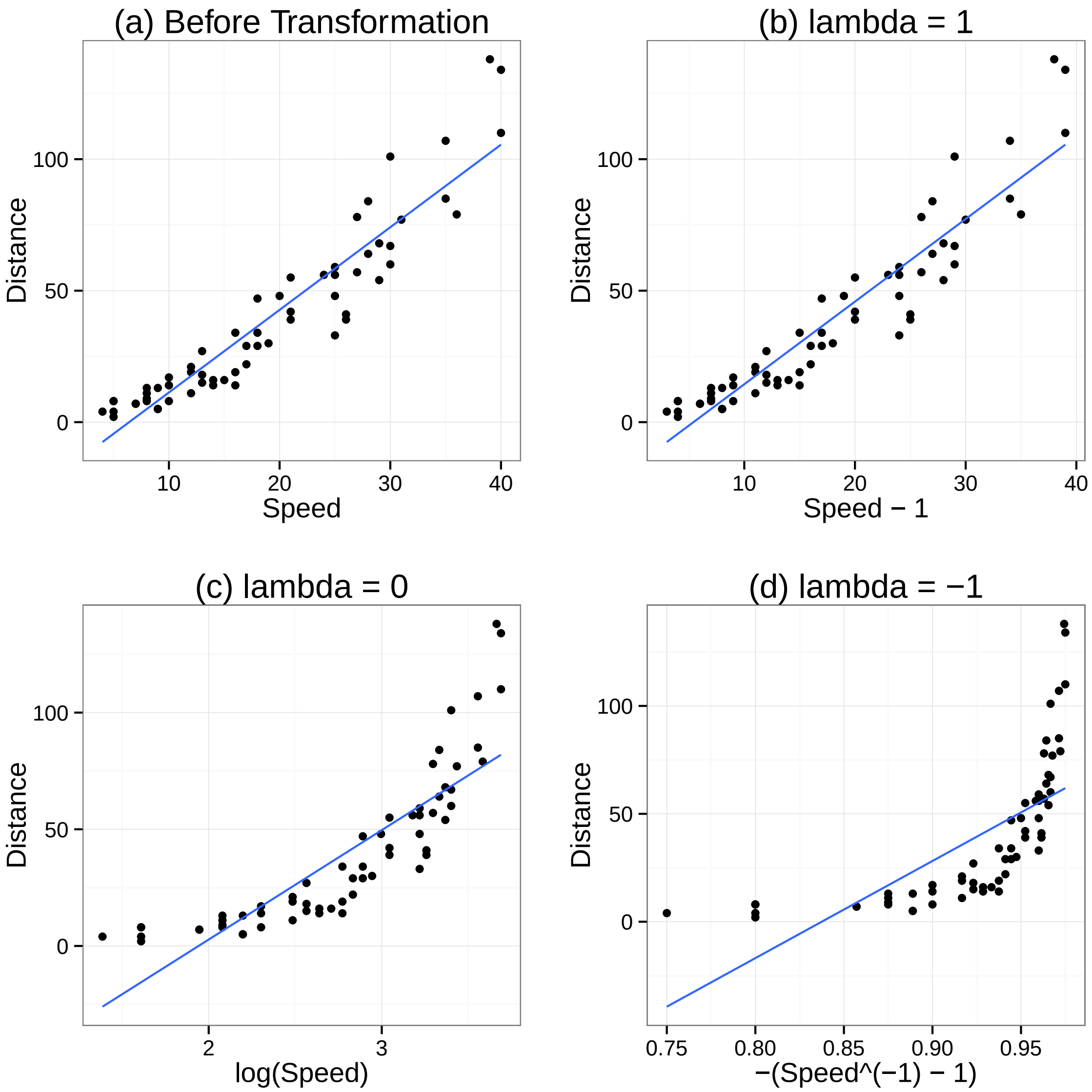
```
p1 <- ggplot(data, aes(x = Speed, y = Distance))
p1 <- p1 + geom_point()
p1 <- p1 + geom_smooth(formula = y ~ x, method = "lm", se = FALSE)
p1 <- p1 + theme_bw() + theme(text = element_text(size = 14))
p1 <- p1 + ggtitle("(a) Before Transformation")

p2 <- ggplot(data, aes(x = Speed - 1, y = Distance))
p2 <- p2 + geom_point()
p2 <- p2 + geom_smooth(formula = y ~ x, method = "lm", se = FALSE)
p2 <- p2 + theme_bw() + theme(text = element_text(size = 14))
p2 <- p2 + ggtitle("(b) lambda = 1")

p3 <- ggplot(data, aes(x = log(Speed), y = Distance))
p3 <- p3 + geom_point()
p3 <- p3 + geom_smooth(formula = y ~ x, method = "lm", se = FALSE)
p3 <- p3 + theme_bw() + theme(text = element_text(size = 14))
p3 <- p3 + ggtitle("(c) lambda = 0")

p4 <- ggplot(data, aes(x = -(Speed ^ (-1) - 1), y = Distance))
p4 <- p4 + geom_point()
p4 <- p4 + geom_smooth(formula = y ~ x, method = "lm", se = FALSE)
p4 <- p4 + theme_bw() + theme(text = element_text(size = 14))
p4 <- p4 + ggtitle("(d) lambda = -1")

grid.arrange(p1, p2, p3, p4, ncol = 2, nrow = 2)
```



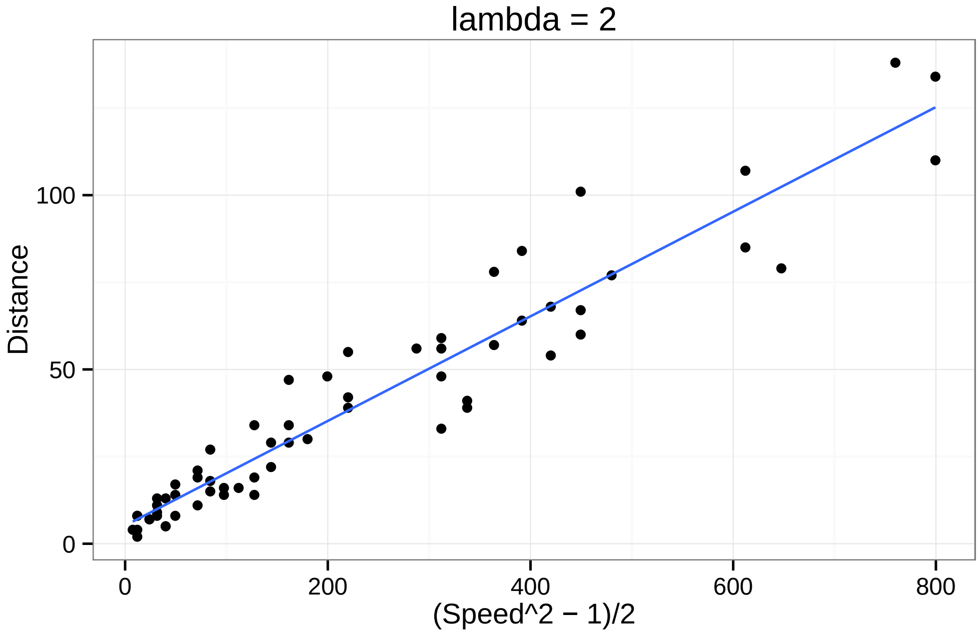
graph (b), (c), (d) above are scatterplots where the predictor for each graph is respectively transformed with power $\lambda \in \{-1, 0, 1\}$. We can see that none of the above scatterplots can be well fitted by a linear model, suggesting that none of these transformations is adequate.

Problem 8.2.3

Show that using $\lambda = 2$ does match the data well. This suggests using a quadratic polynomial for regressors, including both Speed and Speed^2.

Solution

```
p1 <- ggplot(data, aes(x = (Speed ^ 2 - 1)/2, y = Distance))
p1 <- p1 + geom_point()
p1 <- p1 + geom_smooth(formula = y ~ x, method = "lm", se = FALSE)
p1 <- p1 + theme_bw() + theme(text = element_text(size = 12))
p1 <- p1 + ggtitle("lambda = 2")
p1
```



The graph above is the scatterplot of `Distance` against `Speed`, where `Speed` is transformed with $\lambda = 2$. It shows that under such transformation, the points can be fitted well by a linear model.

Problem 8.2.4

Hald (1960) suggested on the basis of a theoretical argument using a quadratic mean function for `Distance` given `Speed`, with $\text{Var}(\texttt{Distance}|\texttt{Speed}) = \sigma^2\texttt{Speed}^2$. Draw the plot of `Distance` versus `Speed`, and add a line on the plot of the fitted curve from Halds model. Then obtain the fitted values from the fit of the transformed `Distance` on `Speed`, using the transformation you found in Problem 8.2.1. Transform these fitted values to the `Distance` scale (for example, if you fit the regression `sqrt(Distance)` $\sim$ `Speed`, then the fitted values would be in square-root scale and you would square them to get the original `Distance` scale). Add to your plot the line corresponding to these transformed fitted values. Compare the fit of the two models.

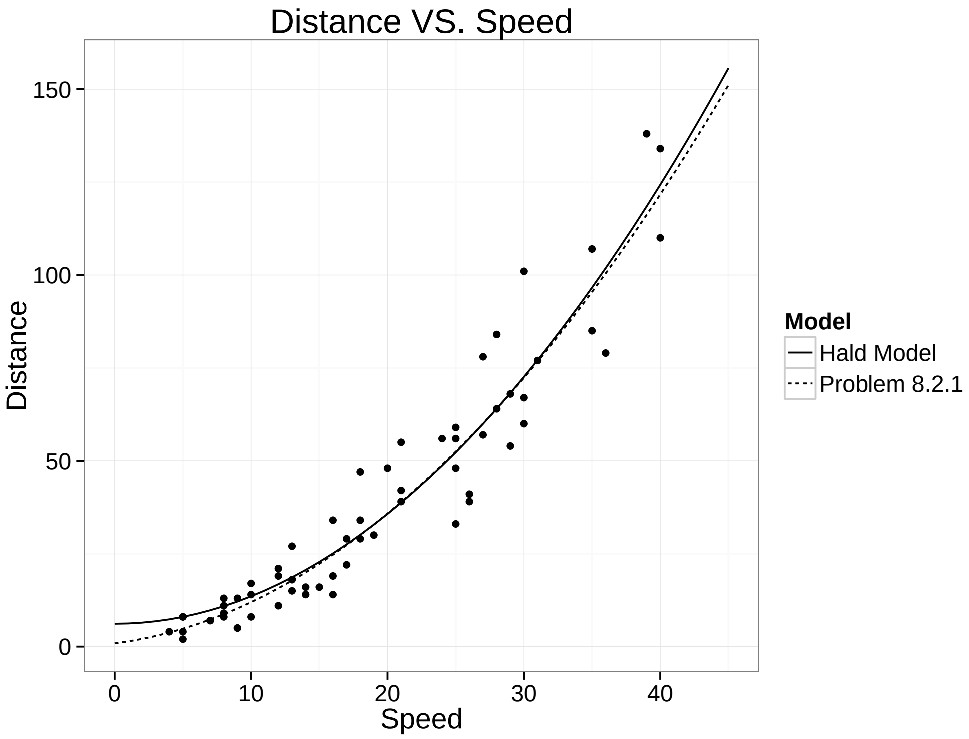
Solution

```
require(reshape2)
data <- stopping
m_h <- lm(Distance ~ I(Speed^2), data = data, weights = Speed^2)
m_1 <- lm(sqrt(Distance) ~ Speed, data = data)

x = 0:45
y_h = x^2 * coef(m_h)[2] + coef(m_h)[1]
y_1 = (x * coef(m_1)[2] + coef(m_1)[1])^2

data_plot <- data.frame(x, y_Hald = y_h, y_P1 = y_1)
data_plot <- melt(data_plot, id.vars = 1)
colnames(data_plot)[2] <- "group"

p <- ggplot()
p <- p + geom_point(aes(x = data$Speed, y = data$Distance))
p <- p + geom_line(aes(x = data_plot$x, y = data_plot$value,
                       linetype = data_plot$group))
p <- p + theme_bw() + theme(text = element_text(size = 16))
p <- p + ggtitle("Distance VS. Speed") + xlab("Speed") + ylab("Distance")
p <- p + scale_linetype_discrete(name  ="Model",
                              breaks=c("y_Hald", "y_P1"),
                              labels=c("Hald Model", "Problem 8.2.1"))
p
```



In this problem we fit Hold's model with weighted parameter Speed^2, and then draw the Hold's model and model fitted in problem 8.2.1 in the same graph above. We see that, according to the graph above, these two models are almost the same. The only difference is that Hold's model is better in fitting points where predictor value is high, and model in problem 8.2.1 is better in fitting points where predictor value is low.

Problem 9.11

(Data file: `fuel2001`) In the fuel consumption data, consider fitting the mean function

$$E(\texttt{Fuel}|X) = \beta_0 + \beta_1\texttt{Tax} + \beta_2\texttt{Dlic} + \beta_3\texttt{Income} + \beta_4\texttt{log(Miles)}$$

For this regression, we find $\hat{\sigma} = 64.891$ with 46 df, and the diagnostic statistics for four states and the District of Columbia were the following:

	`Fuel`	$\hat{e}_i$	h_{ii}
Alaska	514.279	-163.145	0.256
New York	374.164	-137.599	0.162
Hawaii	426.349	-102.409	0.206
Wyoming	842.792	183.499	0.084
District of Columbia	317.492	-49.452	0.415

Compute D_i and t_i for each of these cases, and test for one outlier. Which is most influential?

Solution

```
data <- fuel2001
data$Dlic <- data$Drivers / data$Pop
data$Fuel <- 1000 * data$FuelC / data$Pop

test_outlier <- function(data, index)
{
    m_tmp <- lm(Fuel ~ Tax + Dlic + Income + log(Miles), data = data,
                x = TRUE, y = TRUE)
    X = m_tmp$x
    y = m_tmp$y

    m <- lm(Fuel ~ Tax + Dlic + Income + log(Miles), data = data, subset = -index)
    y_predict <- predict(object = m, newdata = data[index, ])

    sigma <- sqrt(sum(m$residuals^2) / m$df.residual)
    t = (y[index] - y_predict) /
        (sigma * sqrt(1 + t(X[index, ]) %*% solve(t(X[-index, ])
                                %*% X[-index, ]) %*% X[index, ]))
    ## Need Notice, we test 5 states, so the df = 51 - 5, and then we use Bonferroni p-v
    p_value = min((1 - pt(abs(t), df = 46)) * 2 * 51, 1)
    data.frame(t, p_value)
}

test_influence <- function(data, index)
```

```
{
    m_full <- lm(Fuel ~ Tax + Dlic + Income + log(Miles), data = data)
    m_noindex <- lm(Fuel ~ Tax + Dlic + Income + log(Miles),
                    data = data, subset = -index)

    y_predict_full <- predict(object = m_full, newdata = data)
    y_predict_noindex <- predict(object = m_noindex, newdata = data)

    sigma_2 <- sum(m_full$residuals^2) / m_full$df.residual

    D <- t(y_predict_full - y_predict_noindex) %*%
        (y_predict_full - y_predict_noindex) /
        (length(coef(m_full)) * sigma_2)

    data.frame(D)
}

result <- rbind(test_outlier(data, match("AK", rownames(data))),
test_outlier(data, match("NY", rownames(data))),
test_outlier(data, match("HI", rownames(data))),
test_outlier(data, match("WY", rownames(data))),
test_outlier(data, match("DC", rownames(data))))

result <- cbind(result, rbind(test_influence(data, match("AK", rownames(data))),
test_influence(data, match("NY", rownames(data))),
test_influence(data, match("HI", rownames(data))),
test_influence(data, match("WY", rownames(data))),
test_influence(data, match("DC", rownames(data)))))

rownames(result) <- c("AK", "NY", "HI", "WY", "DC")
result

##          t p_value      D
## AK -3.1930  0.1296 0.5850
## NY -2.4382  0.9527 0.2081
## HI -1.8144  1.0000 0.1624
## WY  3.2461  0.1115 0.1596
## DC -0.9962  1.0000 0.1408
```

Here we define two functions `test_outlier` and `test_influence` to get the D_i, t_i and the p-value derived from t_i. The p-values derived from the outlier test suggest that there is no evidence to say that any of these states is outlier. The most influential point is Alaska.