Homework 02 Solutions

9/14/2018

Note: Question 4 is not graded. See the separate solution file for Question 4.

1. (Sheather 2.8.6)

(a)

Proof.

$$\begin{aligned}(y_i - \hat{y}_i) &= (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)) && \text{since } \hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i \\ &= (y_i - (\bar{y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 x_i)) && \text{since } \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \\ &= (y_i - \bar{y}) - \hat{\beta}_1 (x_i - \bar{x})\end{aligned}$$

□

(b)

Proof.

$$\begin{aligned}(\hat{y}_i - \bar{y}) &= ((\hat{\beta}_0 + \hat{\beta}_1 x_i) - \bar{y}) \\ &= ((\bar{y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 x_i) - \bar{y}) \\ &= \hat{\beta}_1 (x_i - \bar{x})\end{aligned}$$

□

(c)

Proof. Substituting the equalities from parts (a) and (b) above, we have:

$$\begin{aligned}\sum_{i=1}^{n} (y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) &= \sum_{i=1}^{n} [(y_i - \bar{y}) - \hat{\beta}_1 (x_i - \bar{x})] \hat{\beta}_1 (x_i - \bar{x}) \\ &= \hat{\beta}_1 \left[\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) - \hat{\beta}_1 \sum_{i=1}^{n} (x_i - \bar{x})^2 \right] \\ &= \hat{\beta}_1 \left[SXY - \frac{SXY}{SXX} SXX \right] \\ &= \hat{\beta}_1 [SXY - SXY] \\ &= 0\end{aligned}$$

□

2. (Gelman & Hill 2.6)

(a)

```
var1 <- rnorm(1000, 0, 1)
var2 <- rnorm(1000, 0, 1)
model <- lm(var2 ~ var1)
summary(model)
```

```
##
## Call:
## lm(formula = var2 ~ var1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7075 -0.6497 -0.0185  0.6384  3.5440
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02605    0.03062   0.851    0.395
## var1         0.01784    0.03014   0.592    0.554
##
## Residual standard error: 0.9676 on 998 degrees of freedom
## Multiple R-squared:  0.0003509,  Adjusted R-squared:  -0.0006507
## F-statistic: 0.3503 on 1 and 998 DF,  p-value: 0.5541
```

The slope is not significantly different from 0, as we can see from the p-value in the `var1` row, `Pr(>|t|)` column in the model summary.

(b)

```
library(arm)  # Needed for the se.coef() function.
z.scores <- rep(NA, 100)
for (k in 1:100) {
  var1 <- rnorm(1000, 0, 1)
  var2 <- rnorm(1000, 0, 1)
  fit <- lm(var2 ~ var1)
  z.scores[k] <- coef(fit)[2]/se.coef(fit)[2]
}

type1_errors <- sum(abs(z.scores) > 2)
print(type1_errors)
```

```
## [1] 5
```

There were 5 z-scores that were significant.

(c)

The rule of thumb to consider a z-score significant if the absolute value exceeds 2 is roughly equivalent to setting $\alpha = 0.05$, since 95% of the standard normal distribution falls within 1.96 standard deviations of

the mean of 0. Therefore, we should expect the coefficients to be significant 5 out of 100 times on average, corresponding to a Type 1 error rate of 0.05.

3. (Sheather 2.8.7)

The regression line is defined as $\mu(x) = E(Y|X = x)$. In other words, it's the mean value of Y at a given value of the covariates X. The confidence interval is for the regression line itself; that is, it's a confidence interval around an estimated mean. It's entirely possible for the data points themselves to fall outside the 95% confidence interval for a mean.

To gain some intuition, imagine that you have data from a normal distribution and you want to estimate the mean of that distribution. The more data you gather, the narrower the confidence interval around your estimate will be. The underlying distribution hasn't changed, however, so the more data you gather, the more of the data will fall outside the confidence interval for the mean.

5. (Sheather 3.4.8)

Part 1

(a)

Given the problem constraints, the only flexibility we have is in specifying the intercept. We could (1) fix the intercept to some value, or (2) estimate the intercept via least squares.

Option (1) typically means setting the intercept to 0, which can be nice for interpretability in cases where the value (0, 0) is meaningful and where a different intercept would not make sense. For example, suppose we were asked to regress the number of words typed in a paper against the number of hours of work put in. Zero hours of work would necessarily produce zero words, so in that case there is a good reason to set the intercept to 0.

Here, however, the size of a diamond is presumably bounded away from 0. The notion of a zero-carat diamond doesn't make sense, and no one will manufacture a ring with a diamond that's so small as to be invisible to the naked eye. Indeed, the sizes of diamonds in the data range from 0.12 to 0.35 carats. If this is the range we're interested in modeling, then we may as well give the model the additional flexibility of a non-zero intercept, since the y-value at the intercept will not be meaningful regardless.

Hence, we take option (2) and estimate a model of the form

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

Although the problem dicatates that we use a linear model, you should always visualize the data before specifying a model to help you decide whether that model is appropriate. Since we only have two variables, let's simply look at a scatter plot of Price against Size (Figure 1).

```
diamonds <- read.table("diamonds.txt", header = TRUE)
plot(Price ~ Size, data = diamonds)
```

It looks like there's a more or less linear relationship, so let's fit the model.

```
model5a <- lm(Price ~ Size, data = diamonds)
```

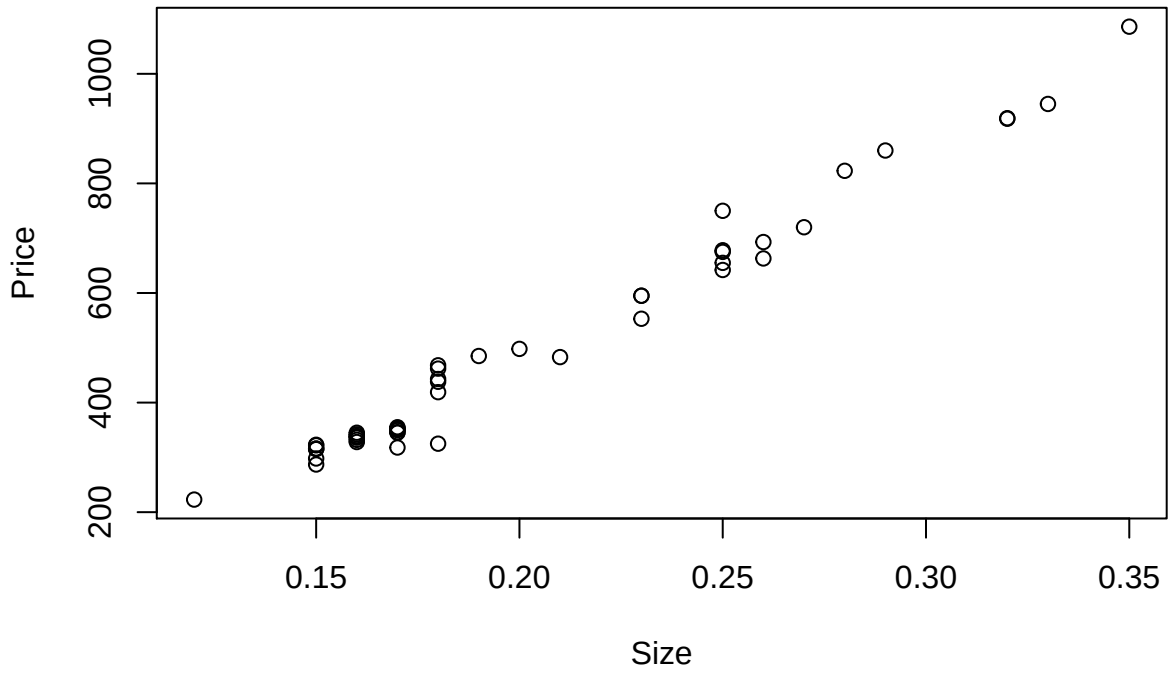

Figure 1: Scatter plot of diamond Price against Size

The coefficients of the model are

$\hat{\beta}_0$: -258.05

$\hat{\beta}_1$: 3715.02

Let's add the best fit line to the scatter plot (Figure 2).

```
plot(Price ~ Size, data = diamonds)
abline(a = coef(model5a)[1], b = coef(model5a)[2], col = "red", lty = "dashed")
```

The model looks reasonable. We'll examine diagnostic plots in Part 2 below.

(b)

One weakness of the model is that by definition, it cannot capture nonlinearities in the relationship. Additionally, it has only one predictor; additional predictors could provide additional information. It also cannot deal with nonconstant variance.

Part 2

(a)

In order to decide whether and how to modify the model above, let's examine some diagnostic plots. First, the residuals against the predictor (Figure 3).

```
plot(model5a$residuals ~ diamonds$Size, xlab = "Size", ylab = "Residuals")
```

There's no clear relationship (e.g. the residuals don't seem to be increasing or decreasing with Size), so that's good. Now let's examine the diagnostic plots that R produces for us automatically (Figure 4).

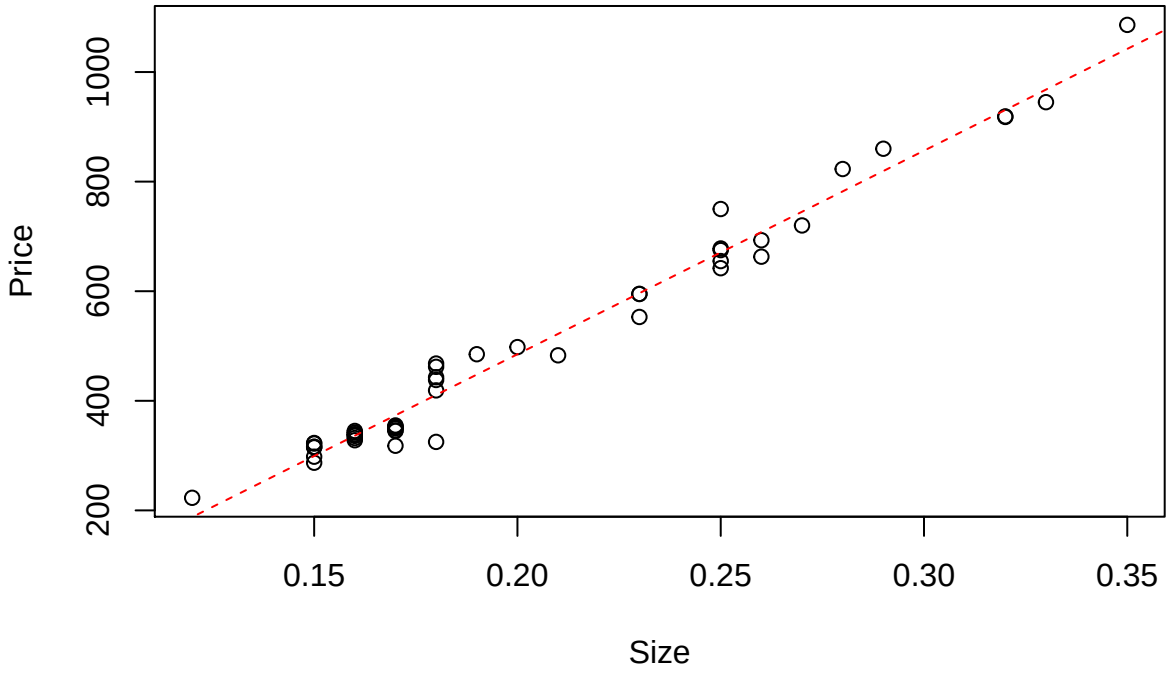

Figure 2: Scatter plot of diamond Price against Size with least squares line

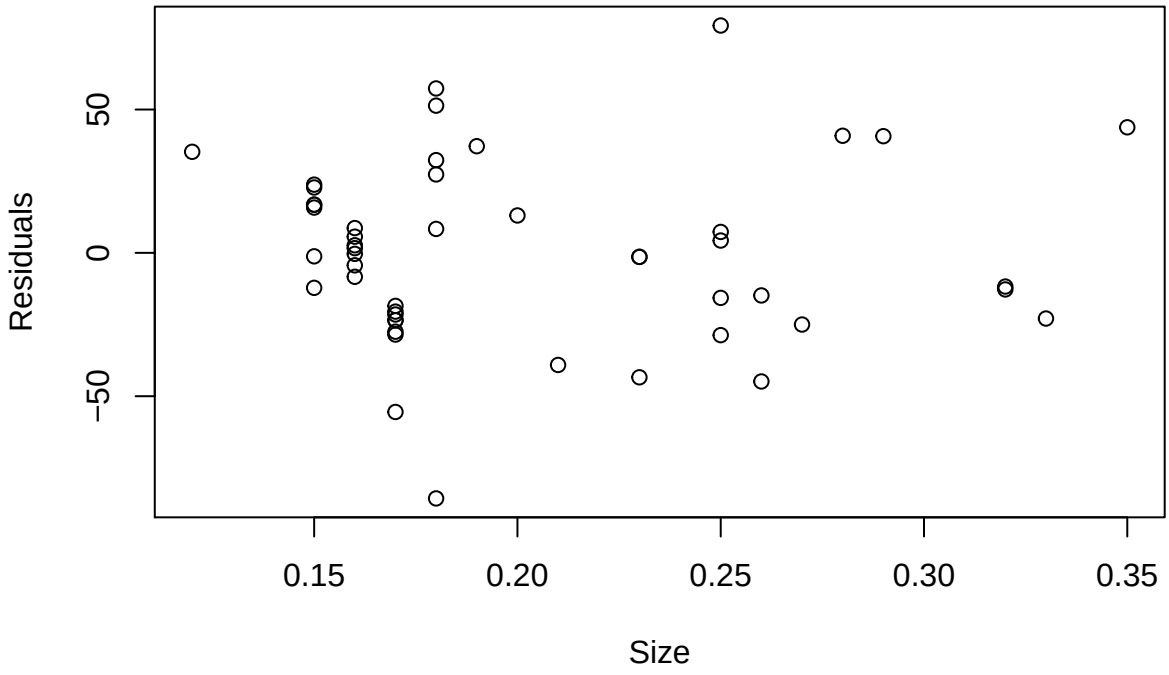

Figure 3: Model 5a residuals against predictor

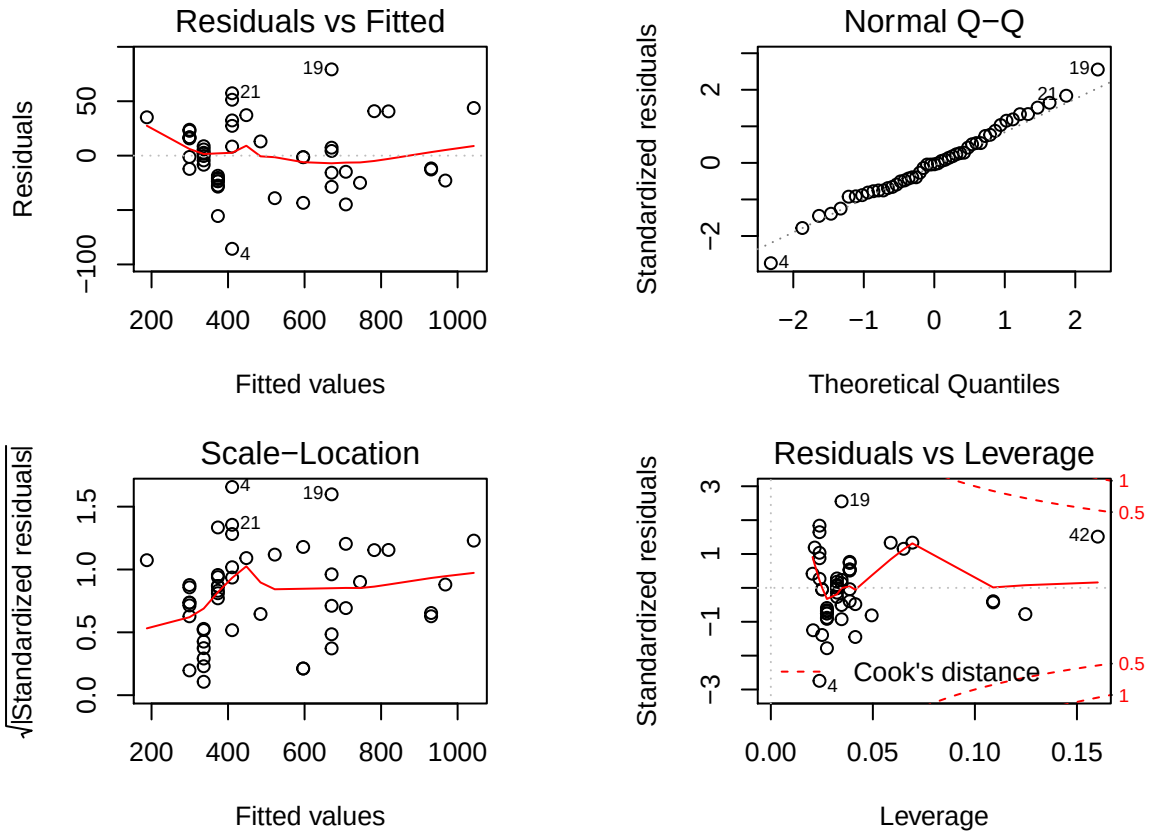

Figure 4: Diagnostic plots for model 5a

```
par(mfrow = c(2, 2), mar = c(4, 4.5, 2.5, 4)) # reducing the margins slightly
plot(model5a)                                 # generating diagnostic plots
```

Let's discuss each of these in turn:

- The top left plot shows no relationship between the residuals and the fitted values (the $\hat{y}_i$s), so that's good.
- The top right plot shows that the residuals are approximately normally distributed, which is also good.
- The nonparametric red line in the bottom left plot shows a slight increasing trend between the fitted values and the standardized residuals, but there aren't many data points on the right side of the plot, so I wouldn't take this too seriously. (Recall that the variance of the residuals $\hat{e}_i$ is nonconstant, *even if the variance of the underlying errors e_i is constant.* The standardized residuals have approximately constant variance, so it is useful to examine these instead of the raw residuals, particularly when points of high leverage exist.)
- The bottom right plot shows that there are a couple outliers in the sense defined in Sheather: points whose standardized residual falls outside the range [-2, 2]. R has helpfully labeled these points with their row numbers in the data, so we can examine them. Additionally, recall the rule of thumb that a point is "high leverage," or equivalently a "leverage point," if its leverage value h_{ii} is greater than $4/n$, where n is the number of data points. In this case, $4/n = 4/49 \approx 0.082$, so we have a couple of leverage points in the data. One of these, point 42, has a relatively high Cook's distance; it lies close to the contour line that represents a Cook's distance of 0.5.

Since R has helpfully labeled points 4, 19, and 42 for us in the bottom right diagnostic plot, let's examine these on the original graph (Figure 5).

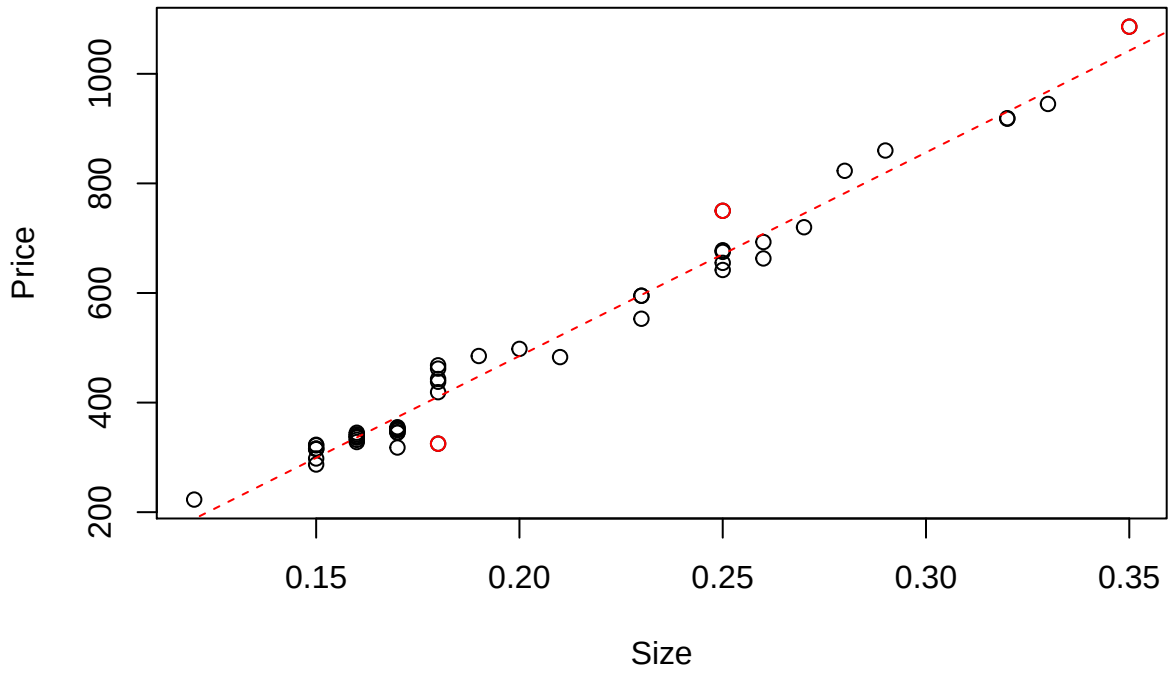

Figure 5: Scatter plot of diamond Price against Size with least squares line and points 4, 19, 42 highlighted

```
plot(Price ~ Size, data = diamonds)
abline(a = coef(model5a)[1], b = coef(model5a)[2], col = "red", lty = "dashed")
points(diamonds[c(4, 19, 42), ], col = "red")
```

None of these points is obviously problematic. Let's check one more diagnostic, the inverse response plot:

```
# install.packages(alr3)  # if you don't already have it installed
library(alr3)
invisible(inverseResponsePlot(model5a))
```

Based on this plot, it looks like the original model, represented by the light blue line ($\lambda = 1$) fits the data well. It doesn't look like we would do better transforming the response variable, at least not with a scaled power transformation.

Given these results, it seems reasonable to keep the original model in 5(a)

(b)

Since the model hasn't changed, see 5(a).

Part 3

The model did not change.

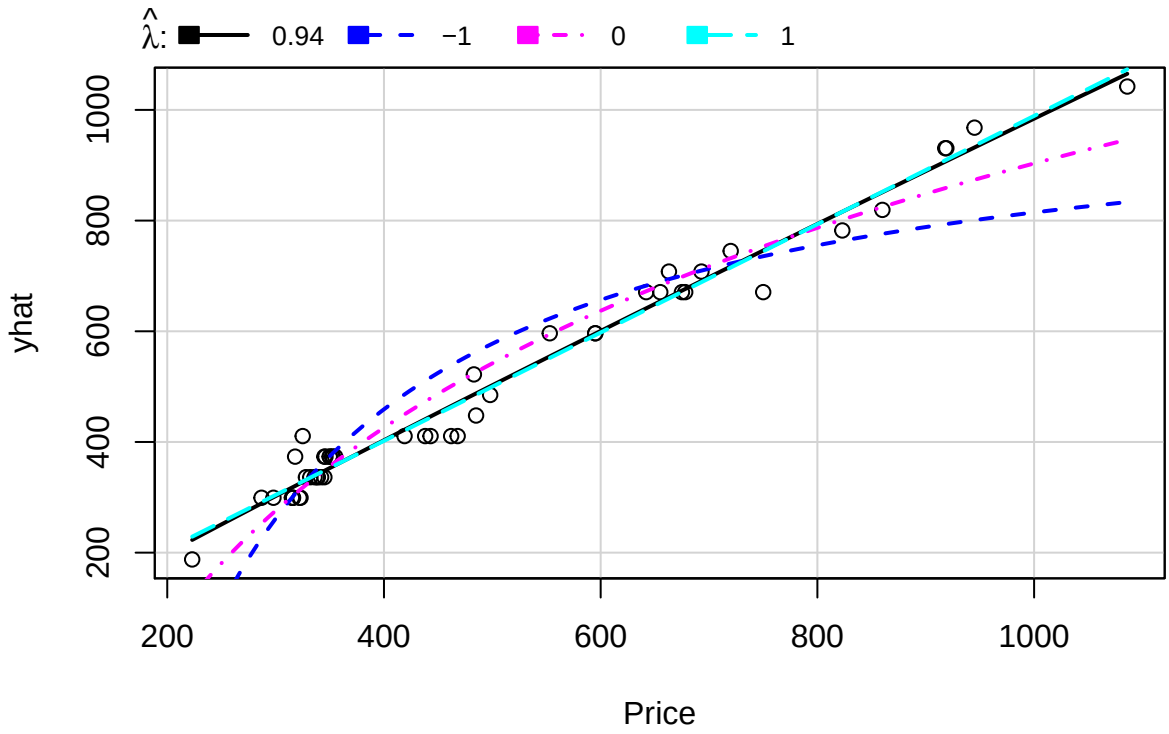

Figure 6: Inverse response plot for the diamonds model5a