36-401 Modern Regression HW #4 Solutions

DUE: 9/29/2017

Note: In the body of this document I omit the vast majority of the code I used in Problems 2-4, primarily because most of it serves the purpose of making the plots prettier and I do not want to confuse those just beginning to learn R. However, if you are interested, I provide my code in the appendix with `eval = FALSE`.

Problem 1 [10 points]

In Homework 2 we showed

$$\widehat{\beta} = \frac{\sum_{i=1}^n Y_i X_i}{\sum_{i=1}^n X_i^2}. \tag{1}$$

We can rewrite this as

$$\begin{aligned}
\widehat{\beta} &= \frac{\sum_{i=1}^n Y_i X_i}{\sum_{i=1}^n X_i^2} \\
&= \frac{\sum_{i=1}^n \left(\beta X_i + \epsilon_i\right) X_i}{\sum_{i=1}^n X_i^2} \\
&= \frac{\beta \sum_{i=1}^n X_i^2 + \sum_{i=1}^n X_i \epsilon_i}{\sum_{i=1}^n X_i^2} \\
&= \beta + \frac{\sum_{i=1}^n X_i \epsilon_i}{\sum_{i=1}^n X_i^2}.
\end{aligned}$$

At this point you can invoke the fact that a linear combination of Gaussian random variables is also Gaussian (something you proved in 36-225). Therefore, $\widehat{\beta}$ follows a Gaussian distribution with mean

$$\begin{aligned}
\mathbb{E}[\widehat{\beta}] &= \mathbb{E}\left[\beta + \frac{\sum_{i=1}^n X_i \epsilon_i}{\sum_{i=1}^n X_i^2}\right] \\
&= \beta + \frac{\sum_{i=1}^n X_i \mathbb{E}[\epsilon_i]}{\sum_{i=1}^n X_i^2} \\
&= \beta
\end{aligned}$$

and variance

$$\begin{aligned}
\mathrm{Var}(\widehat{\beta}) &= \mathrm{Var}\left(\beta + \frac{\sum_{i=1}^n X_i \epsilon_i}{\sum_{i=1}^n X_i^2}\right) \\
&= \frac{1}{\left(\sum_{i=1}^n X_i^2\right)^2} \mathrm{Var}\left(\sum_{i=1}^n X_i \epsilon_i\right) \\
&= \frac{1}{\left(\sum_{i=1}^n X_i^2\right)^2} \sum_{i=1}^n X_i^2 \mathrm{Var}(\epsilon_i) \\
&= \frac{1}{\left(\sum_{i=1}^n X_i^2\right)^2} \sum_{i=1}^n X_i^2 \sigma^2 \\
&= \frac{\sigma^2}{\sum_{i=1}^n X_i^2}.
\end{aligned}$$

Alternate Solution

Recall from 36-225 that the distribution of a random variable has a one-to-one correspondence with its moment generating function. Above we showed

$$\widehat{\beta} = \beta + \frac{\sum_{i=1}^n X_i \epsilon_i}{\sum_{i=1}^n X_i^2}.$$

The MGF of $\widehat{\beta}$ is

$$\begin{aligned}
M_{\widehat{\beta}}(t) &= \mathbb{E}\left[e^{\widehat{\beta} t}\right] \\
&= \mathbb{E}\left[e^{\left(\beta + \frac{\sum_{i=1}^n X_i \epsilon_i}{\sum_{i=1}^n X_i^2}\right) t}\right] \\
&= e^{\beta t} \mathbb{E}\left[\prod_{i=1}^n e^{\frac{X_i \epsilon_i t}{\sum_{i=1}^n X_i^2}}\right] \\
&= e^{\beta t} \prod_{i=1}^n \mathbb{E}\left[e^{\frac{X_i \epsilon_i t}{\sum_{i=1}^n X_i^2}}\right] && \text{by independence of } \epsilon_i \\
&= e^{\beta t} \prod_{i=1}^n M_{\epsilon_i}\left(\frac{X_i t}{\sum_{i=1}^n X_i^2}\right) \\
&= e^{\beta t} \prod_{i=1}^n e^{\frac{1}{2}\sigma^2 t^2 \frac{X_i^2}{\left(\sum_{i=1}^n X_i^2\right)^2}} && \text{plugging in Gaussian MGF} \\
&= e^{\beta t + \frac{1}{2}\frac{\sigma^2 t^2}{\sum_{i=1}^n X_i^2}},
\end{aligned}$$

which we recognize as a Gaussian MGF. In particular,

$$\widehat{\beta} \sim N\left(\beta,\ \frac{\sigma^2}{\sum_{i=1}^n X_i^2}\right).$$

Problem 2 [20 points]

(a) (6 pts.)

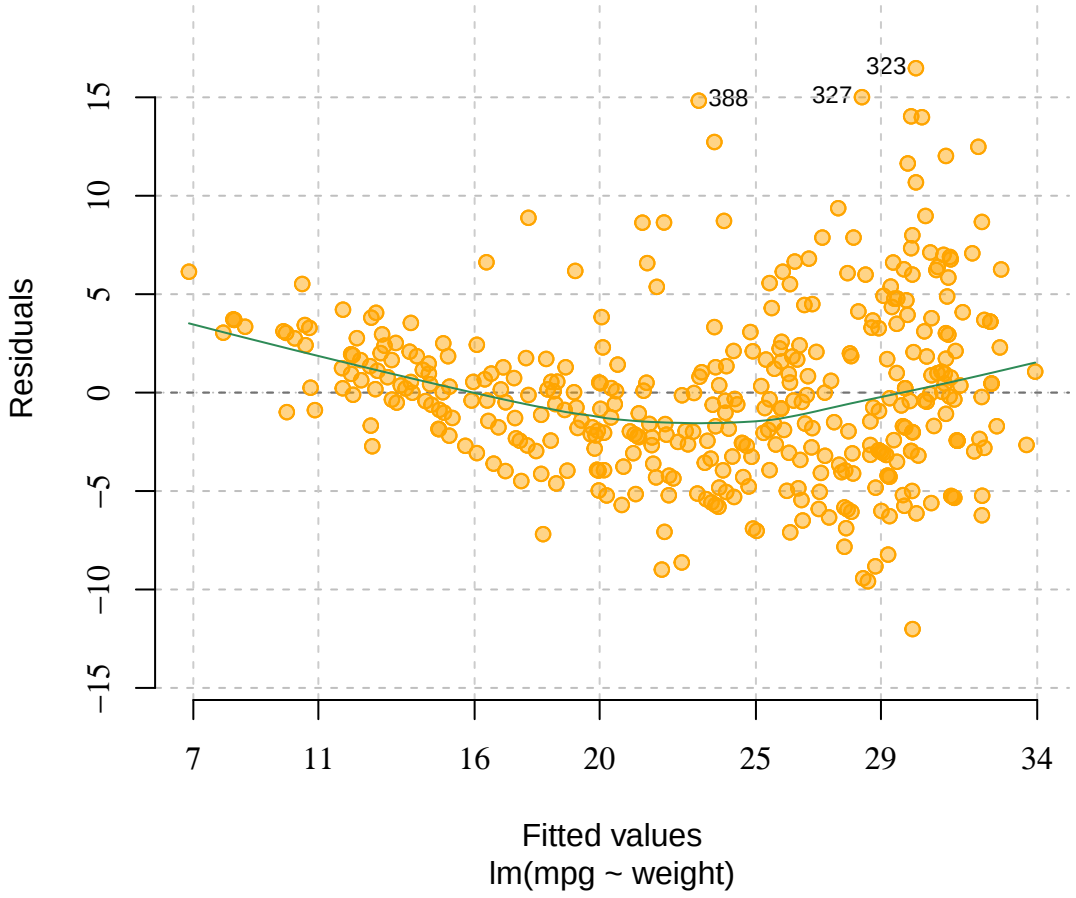


Figure 1: Residuals vs. predicted MPG for regression on car weight

The residuals are clearly not symmetric about zero for all values of $\widehat{Y}$, which suggests the linearity assumption is violated.

(b) (8 pts.)

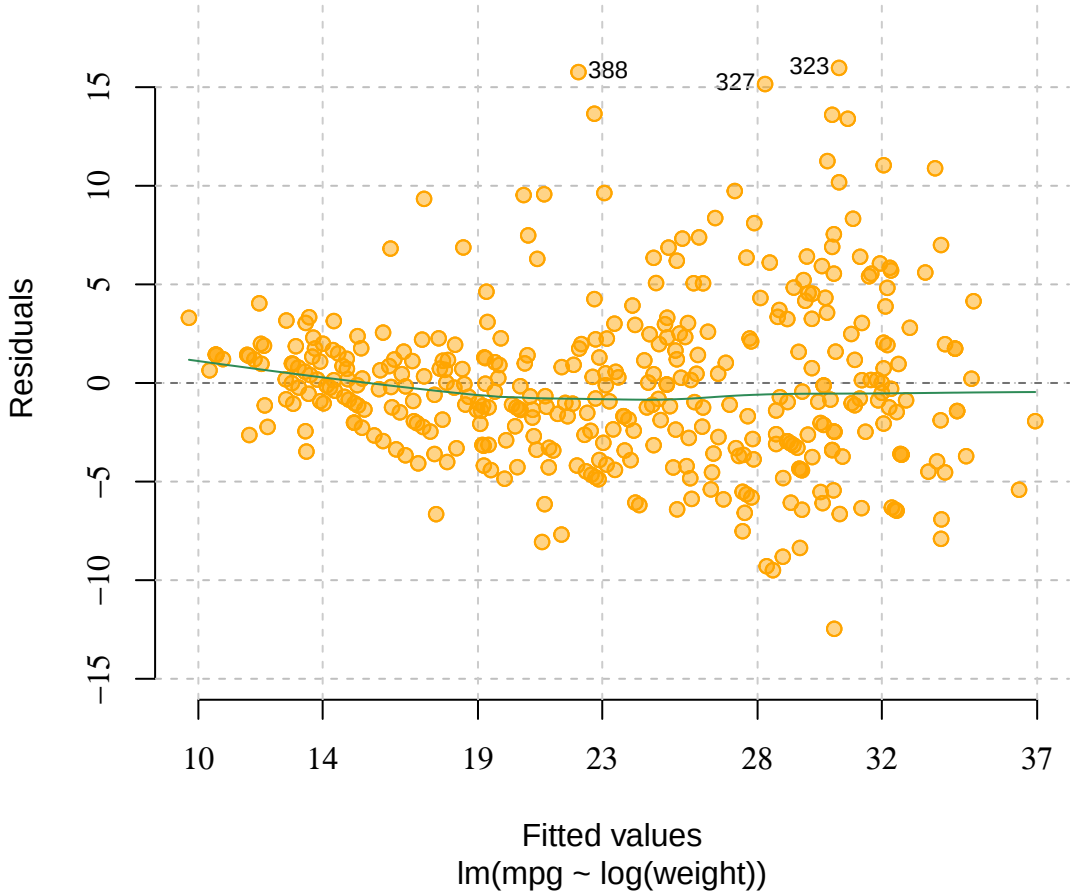


Figure 2: Residuals vs. predicted mpg for regression on log(weight)

Figure 2 shows that applying a log transformation to `weight` has improved the linear fit. Nevertheless, the linear assumption is still violated (albeit less egregiously) as the residuals can be seen to be systematically not centered at zero for all values of $\widehat{Y}$, e.g. $\widehat{Y} \approx 11$, $\widehat{Y} \approx 21$. Morever, Figure 2 shows a clear violation of the homoskedasticity assumption.

(c) (6 pts.)

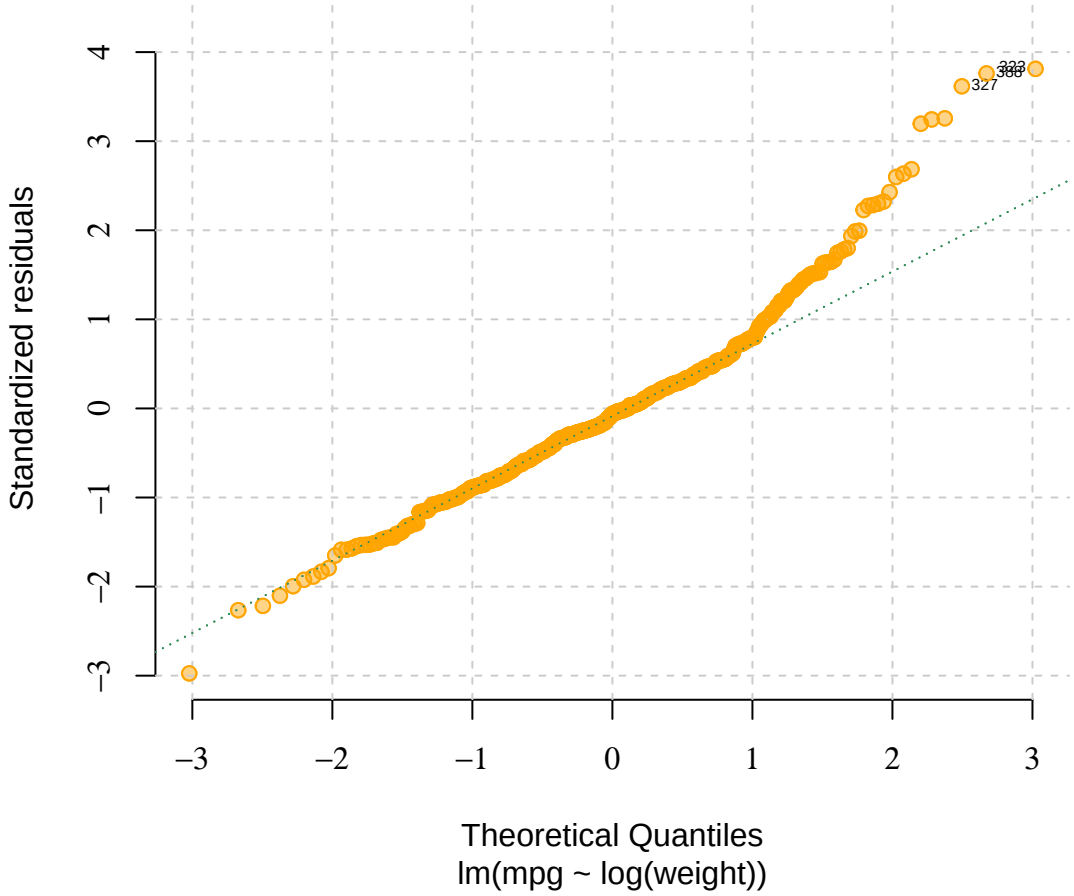


Figure 3: Q-Q plot of mpg vs log(weight) residuals

Figure 3 shows strong evidence against a Gaussian assumption on the noise. In particular, the right tail (the positive residuals) is much too fat.

Problem 3 [20 points]

We will use the transformation log(`weight`) for our predictor but, before we proceed with an analysis, let's check the Box-Cox power transformation to see if we should transform `mpg` as well.

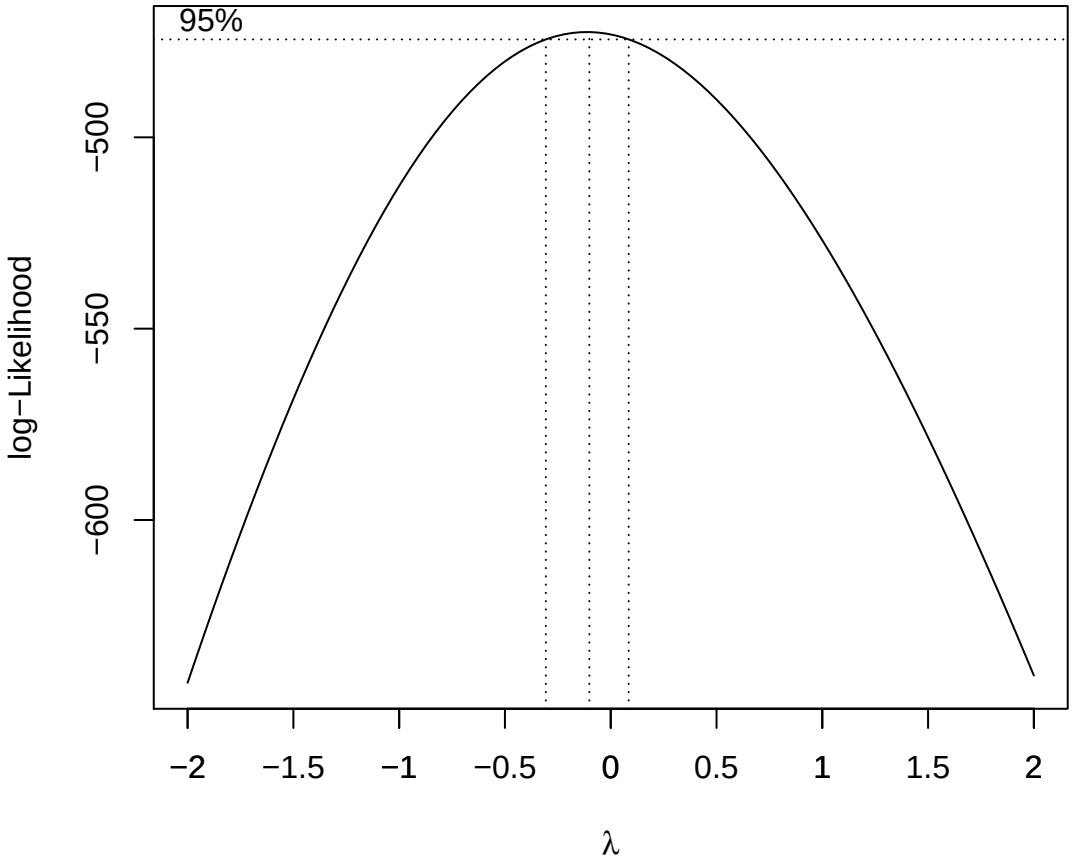


Figure 4: Box-Cox power transformation for mpg on weight

Figure 4 suggests $\lambda = 0$ is a reasonable choice, so we let our response be log(`mpg`). The residuals of the regression are shown in Figure 5.

(a) (5 pts.)

```
transformed.weight <- log(dat$weight)
transformed.mpg <- log(dat$mpg)
model3 <- lm(transformed.mpg ~ transformed.weight)
coefficients(model3)
```

$\widehat{\beta}_0 = 11.521907$ is an empirical estimate for

$$\mathbb{E}[\log(\texttt{mpg}) | \log(\texttt{weight}) = 0]$$

and $\widehat{\beta}_1 = -1.058268$ is an empirical estimate for

$$\mathbb{E}[\log(\texttt{mpg}) | \log(\texttt{weight}) = x + 1] - \mathbb{E}[\log(\texttt{mpg}) | \log(\texttt{weight}) = x], \quad x \in \text{Support}(\log(\texttt{weight})),$$

i.e., it is an estimate for the slope of $\mathbb{E}[\log(\texttt{mpg}) | \log(\texttt{weight})]$.

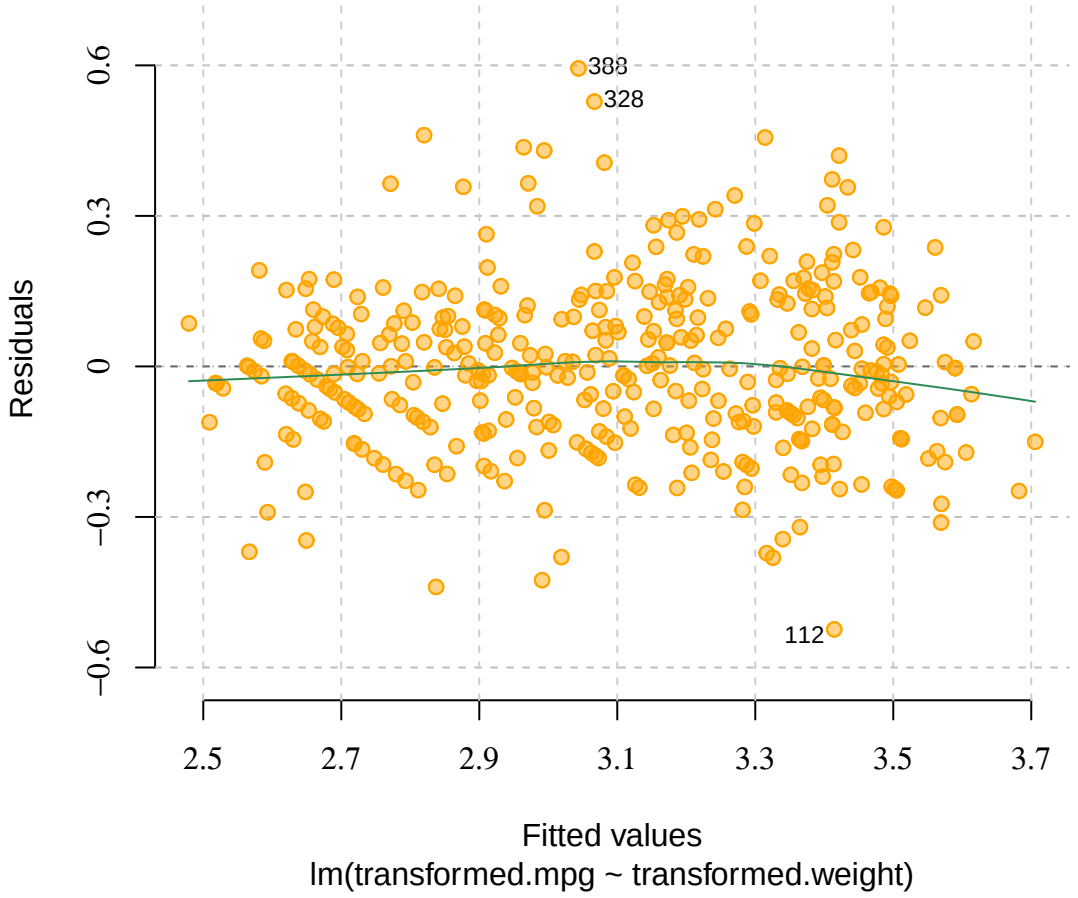


Figure 5: Residuals vs fitted values for log(mpg) vs log(weight) regression

(b) (10 pts.)

The hypothesis being tested is:

$$\begin{aligned} H_0 &: \beta_1 = 0 \qquad \big(\log(\texttt{mpg}) \text{ and } \log(\texttt{weight}) \text{ have no linear association}\big) \\ H_1 &: \beta_1 \neq 0 \quad \big(\log(\texttt{mpg}) \text{ and } \log(\texttt{weight}) \text{ have a non-zero linear association}\big). \end{aligned}$$

We reject H_0 when:

$$\frac{|\widehat{\beta}_1 - 0|}{\widehat{\text{se}}(\widehat{\beta}_1)} > t_{n-2}(0.025).$$

The right-hand side is

```
qt(0.025, df = 396, lower.tail = FALSE)
```

```
## [1] 1.965973
```

and the left-hand side is given by

```
t <- abs(summary(model3)$coefficients[2,1]) / summary(model3)$coefficients[2,2]
print(t)
```

```
## [1] 35.87373
```

so we reject H_0.

The p-value is given by

```
2 * pt(t, df = 396, lower.tail = FALSE)
```

```
## [1] 1.752225e-126
```

Notice that `lm` also stores the p-value for this test.

```
summary(model3)$coefficients[2,4]
```

```
## [1] 1.752225e-126
```

(c) (5 pts.)

An approximate 90% confidence interval for β_1 is

$$\left(\widehat{\beta}_1 - t_{n-2}(0.05) \cdot \widehat{\text{se}}(\widehat{\beta}_1),\ \widehat{\beta}_1 + t_{n-2}(0.05) \cdot \widehat{\text{se}}(\widehat{\beta}_1)\right).$$

```
tmp <- qt(0.05, df = 396, lower.tail = FALSE) * summary(model3)$coefficients[2,2]
L <- summary(model3)$coefficients[2,1] - tmp
U <- summary(model3)$coefficients[2,1] + tmp
print(list(L,U))
```

```
## [[1]]
## [1] -1.106905
##
## [[2]]
## [1] -1.009631
```

$$C = \left(-1.106905,\ -1.009631\right)$$

You can also make the `confint` do the work for you.

```
confint(model3, level = 0.9, parm = "transformed.weight")
```

```
##                            5 %      95 %
## transformed.weight -1.106905 -1.009631
```

Sample Data Analysis Practice (50 points)

Introduction

The term *abalone* is a common name for a broad class of marine snails, ranging from small (fractions of a millimeter) to potentially very large ($\sim$ 12 inches)[1,2]. The age of an abalone is determined through a tedious process of counting the number of rings on its shell through a microscope. In particular, it is believed an abalone's age (in years) corresponds to one-and-a-half plus the number of rings on its shell. Because of the inconvenience associated with counting an abalone's rings, it would be helpful if age could be approximately estimated by a more easily obtainable feature. In this anaylsis, we will seek to build a predictive model of abalone age from height (in mm.).

Exploratory Data Analyis & Initial Modeling

The data set that we use comes from the University of California, Irvine Machine Learning Repository[1]. This data set documents the height and number of rings of 4177 abalones observed in Tasmania in 1994. Summary statistics for each of the two variables are shown in Table 1. The full marginal distribution of each variable is more clearly shown in Figure 6. The vast majority of the observed abalones fall in the height range of [0, 0.25 mm.], however, there is a pair of relatively large cases, one in particular measuring 1.13 mm. The ring count among the observed abalones ranges from one to 29, with a mode of nine.

Table 1: Summary Statistics for Abalone Data Set

	Minimum	Mean	Median	Maximum	Variance
Height (mm)	0	0.1395	0.14	1.13	0.0017
Rings	1	9.9337	9.00	29.00	10.3953

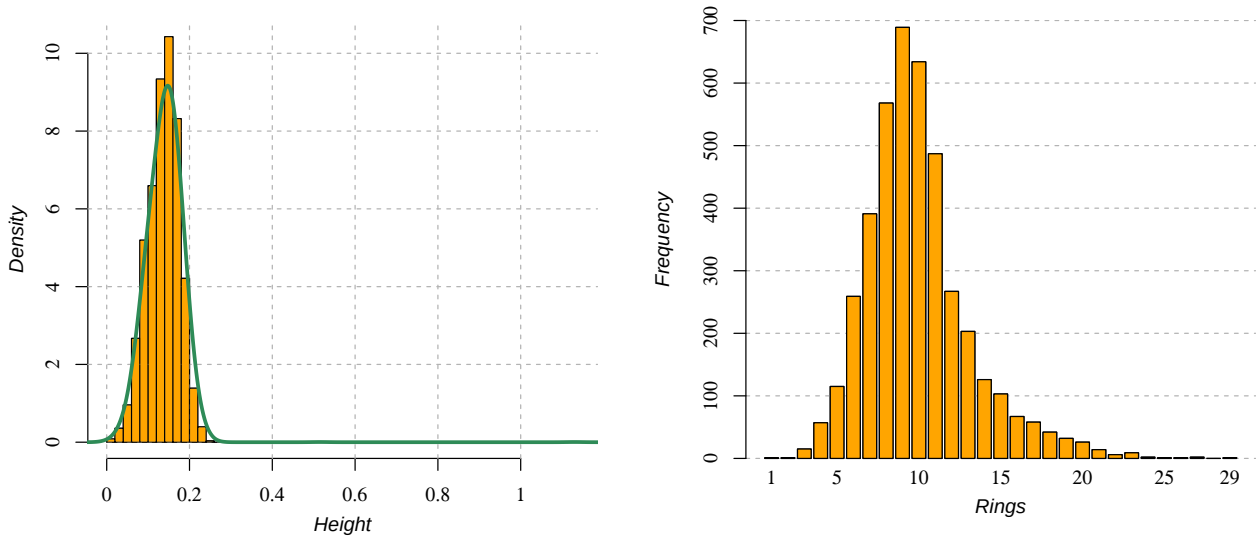


Figure 6: Marginal Distributions of Abalone Height and Ring Count

The results of a naive simple linear regression directly modeling abalone ring count vs. height are shown in Figure 7. The two cases with exceptionally large heights seem to significantly influence the fit. Furthermore,

[1] https://archive.ics.uci.edu/ml/datasets/Abalone
[2] https://en.wikipedia.org/wiki/Abalone

even if we restrict our attention to the non-extreme cases, it does not appear that a linear model will be suitable on this scale.

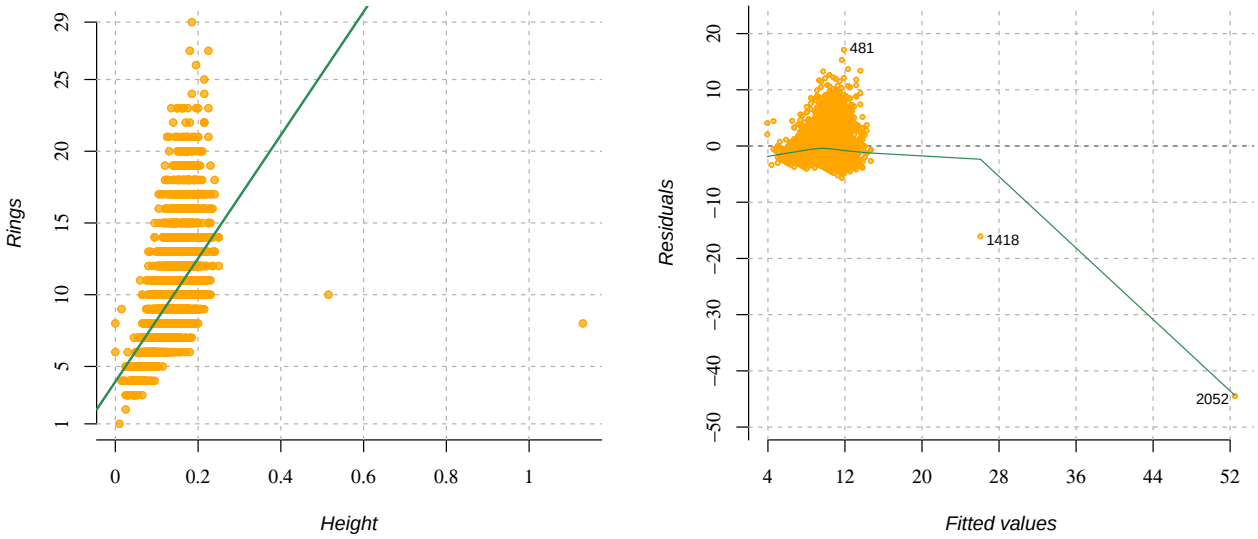


Figure 7: Results of Naive Linear Regression of Abalone Ring Count vs. Height

Modeling and Diagnostics

Given a mere two observations of abalones measuring greater than 0.25 mm. in height were collected in this data set, we remove these observations and focus on building a predictive model of ring count for the smaller abalones. We also discard two observations that were recorded with a height of exactly zero since this is clearly due to limited precision in the measurement instrument. After some preliminary modeling (omitted here), we also found it best to omit the abalones with the three smallest non-zero heights. While these three abalones had approximately the same height (0.01, 0.015, and 0.015 mm.), their ring counts were drastically different (1, 4, and 9, respectively). This suggests there may be a large population variance among abalones in this height range. This is particulary undesirable because these observations lie at the boundary of the height range, and thus possess high leverage on the fit. That is, their high variability will significantly influence $\widehat{\beta}_1$ and se$(\widehat{\beta}_1)$.

When modeling the remaining observations on the height interval [0.02,;0.25 mm.] no transformation of `Height` yields a more linear trend than the originally observed scale. However, as previously noted in our EDA, a linear model is likely not a reasonable assumption on this scale. A Box-Cox transformation of the number of rings yields reasonably healthy residuals; however, modeling on a transformed scale disallows[3] us from doing inference on $\mathbb{E}[\texttt{Rings}|\texttt{Height}]$ (which we are specifically instructed to do), so we do not perform such a transformation[4]. Therefore, for the purpose of answering our given scientific questions, are final chosen model will be of the form

$$\widehat{\texttt{Rings}} = \widehat{\beta}_0 + \widehat{\beta}_1 \cdot \texttt{Height}, \quad \texttt{Height} \in [0.02,\ 0.25 \text{ mm.}]. \tag{2}$$

[3]We cannot back-transform confidence intervals on $\mathbb{E}[g(\texttt{Rings})|\texttt{Height}]$ because $g^{-1}\big(\mathbb{E}[g(\texttt{Rings})|\texttt{Height}]\big) \neq \mathbb{E}[\texttt{Rings}|\texttt{Height}]$, in general.

[4]The diagnostic plots for this model are however provided in the appendix.

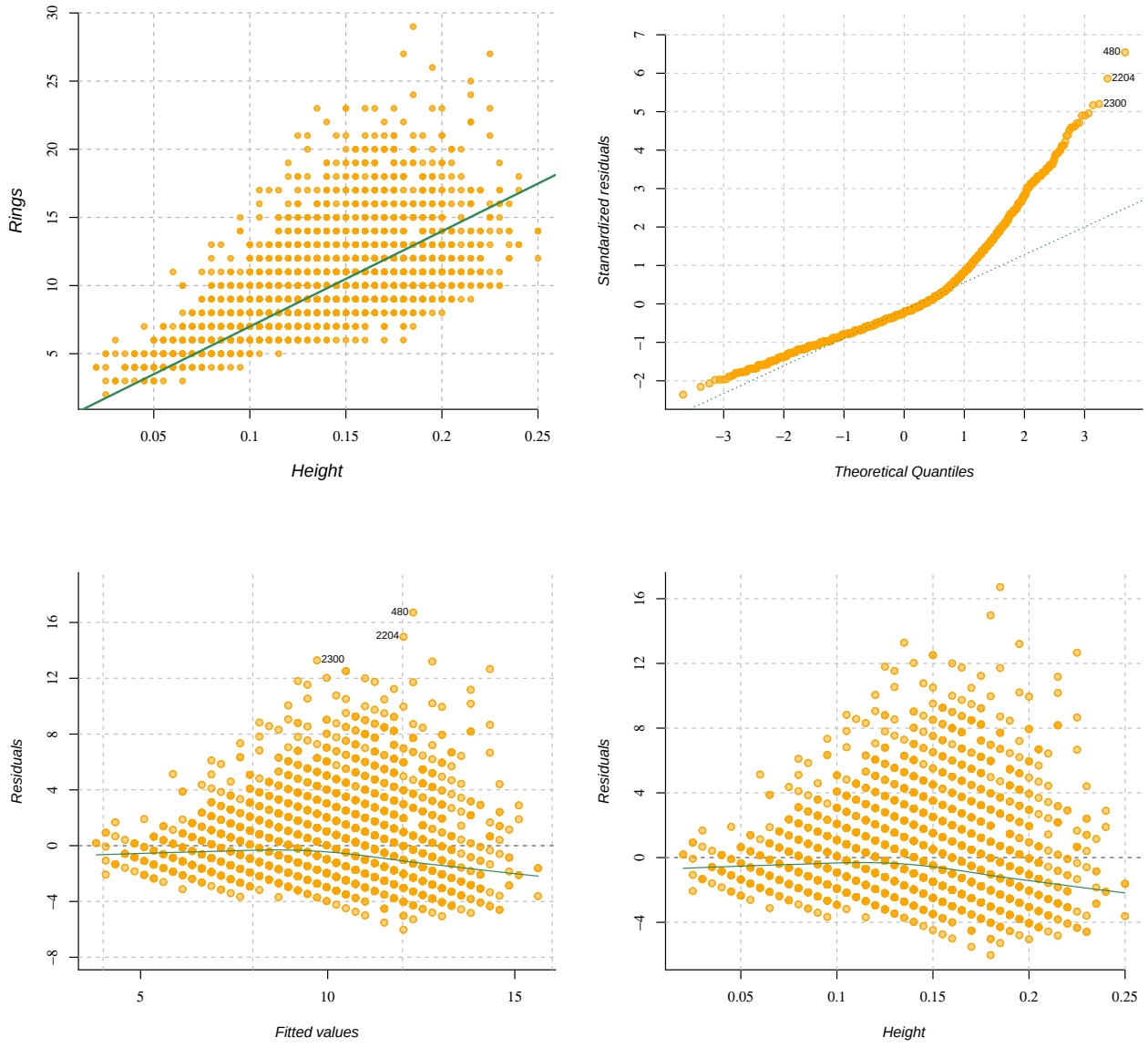


Figure 8: Final model fit and diagnostic plots

Inference and Results

The fit and diagnostic plots for the final model are shown in Figure 8. Due to our inability to transform the ring count, we are left with heteroskedacity in the residuals, which means our confidence interval for the conditional mean ring count will not have correct empirical coverage. The parameter estimates and corresponding confidence intervals are shown in Table 2. Since `Height` $= 0$ has no scientific interpretation, $\widehat{\beta}_0 = 2.7856$ is more appropriately regarded as the baseline number of rings for an abalone, after which a 0.01 mm. increase in height is associated with a 0.51 increase to an abalone's ring count. A t-test for a significant association between ring count and height yields a p-value less than 2×10^{-16} so we can confidently say there is a significant relationship between the height and number of rings of abalones.

Table 2: Parameter Estimates and 95% Confidence Intervals

	Estimate	CI.Lower	CI.Upper
Intercept	2.7856	2.4925	3.0787
Slope	51.3379	49.3095	53.3663

A 95% confidence interval for the average number of rings for abalones with height at 0.128 mm. is (9.275917, 9.437726). The heteroskedasticity of the residuals likely makes this interval wider than it could be. A better approach would be to construct the confidence interval via bootstrapping (36-402). A 99% confidence interval for the number of rings for a *single* abalone with height at 0.132 mm. is (4.55298, 14.57137), with the point estimate 9.562173 mm.

Conclusions and Discussions

The height of an abalone bears significant power in predicting its number of rings, and thus its age. Since the average ring count of abalones as a function of height (or some transformation of height) is our primary interest, we were not able to transform the ring count to homoskedastize the residuals, however this relatively naive model still reveals a very strong association between the two variables of interest. In order to also get confidence intervals for $\mathbb{E}[\texttt{Rings}|\texttt{Height}]$ with correct coverage we recommend applying the Bootstrap (will learn in 36-402). Furthermore, even though the linear mean fit is not egregious, it could be made better by allowing for a more flexible model. In particular, a more complex linear model on `Height` could be used (covered later in 36-401), or a nonparametric model (covered in 36-402).

Appendix

```
# This is a function I found online to adjust the transparancy of colors when plotting
addTrans <- function(color,trans)
{
  # This function adds transparancy to a color.
  # Define transparancy with an integer between 0 and 255
  # 0 being fully transparant and 255 being fully visable
  # Works with either color and trans a vector of equal length,
  # or one of the two of length 1.

  if (length(color)!=length(trans)&!any(c(length(color),length(trans))==1)){
    stop("Vector lengths not correct")}
  if (length(color)==1 & length(trans)>1) color <- rep(color,length(trans))
  if (length(trans)==1 & length(color)>1) trans <- rep(trans,length(color))

  num2hex <- function(x)
  {
    hex <- unlist(strsplit("0123456789ABCDEF",split=""))
    return(paste(hex[(x-x%%16)/16+1],hex[x%%16+1],sep=""))
  }
  rgb <- rbind(col2rgb(color),trans)
  res <- paste("#",apply(apply(rgb,2,num2hex),2,paste,collapse=""),sep="")
  return(res)
}
```

Problem 2

(a)

```
dat <- read.csv("http://www.stat.cmu.edu/~larry/=stat401/auto-mpg.csv")
model <- with(dat, lm(mpg ~ weight))
plot(model, which = 1, col = NA, pch = 19, axes = FALSE,
     add.smooth = FALSE, caption = "")
abline(h = seq(-15,15,5), col = "gray75", lty = 2)
abline(v = round(seq(min(fitted(model)), max(fitted(model)), length = 7)),
       col = "gray80", lty = 2)
abline(0,0, lty = 2, col = "gray45")
#rug(fitted(model), lwd = 0.2, ticksize = 0.025)
axis(side = 1, at = round(seq(min(fitted(model)), max(fitted(model)),
                              length = 7)), as.character(round(seq(min(fitted(model)),
                                                                   max(fitted(model)),
                                                                   length = 7))),
     font = 5)
axis(side = 2, at = seq(-15,15,5), labels = as.character(seq(-15,15,5)),
     font = 5)
points(fitted(model), residuals(model), col = addTrans("orange",120),
       ch = 19)
points(fitted(model), residuals(model), col = "orange")
panel.smooth(fitted(model), residuals(model), col = "orange",cex = 1,
             col.smooth = "seagreen", span = 2/3, iter = 3)
```

(b)

```
model2 <- with(dat, lm(mpg ~ log(weight)))
plot(model2, which = 1, col = NA, pch = 19, axes = FALSE,
     add.smooth = FALSE, caption = "")
axis(side = 1, at = round(seq(min(fitted(model2)), max(fitted(model2)),
                              length = 7)), as.character(round(seq(min(fitted(model2)),
                                                                   max(fitted(model2)),
                                                                   length = 7))),
     font = 5)
axis(side = 2, at = seq(-15,15,5), labels = as.character(seq(-15,15,5)),
     font = 5)
abline(h = seq(-15,15,5), col = "gray75", lty = 2)
abline(v = round(seq(min(fitted(model2)), max(fitted(model2)), length = 7)),
       col = "gray80", lty = 2)
abline(0,0, lty = 2, col = "gray45")
points(fitted(model2), residuals(model2), col = addTrans("orange",120),
       ch = 19)
points(fitted(model2), residuals(model2), col = "orange")
panel.smooth(fitted(model2), residuals(model2), col = "orange",cex = 1,
             col.smooth = "seagreen", span = 2/3, iter = 3)
```

(c)

```
plot(model2, which = 2, col = addTrans("orange",120), pch = 19, axes = FALSE,
     qqline = FALSE, cex.id = 0.5, caption = "")
axis(side = 1, at = seq(-3,3,1), labels = as.character(seq(-3,3,1)), font = 5)
axis(side = 2, at = seq(-3,4,1), labels = as.character(seq(-3,4,1)), font = 5)
qqline(scale(residuals(model2)), col = "seagreen", lty = 3)
secondpts <- qqnorm(scale(residuals(model2)), plot.it = FALSE)
abline(v = seq(-3,3,1), lty = 2, col = "gray80")
abline(h = seq(-3,4,1), lty = 2, col = "gray80")
points(secondpts, col = addTrans("orange",120))
points(secondpts, col = "orange")
qqline(scale(residuals(model2)), col = "seagreen", lty = 3)
```

Problem 3

```
library(MASS)
transformed.weight <- log(dat$weight)
with(dat, boxcox(mpg ~ transformed.weight))
axis(side = 1, at = seq(-2,2,0.5), labels = as.character(seq(-2,2,0.5)))
```

```
transformed.mpg <- log(dat$mpg)
model3 <- lm(transformed.mpg ~ transformed.weight)
plot(model3, which = 1, col = NA, pch = 19, axes = FALSE,
     add.smooth = FALSE, caption = "")
axis(side = 1, at = signif(seq(min(fitted(model3)), max(fitted(model3)), length = 7),
                           digits = 2), as.character(signif(seq(min(fitted(model3)),
                                                                max(fitted(model3)),
                                                                length = 7),
                                                            digits = 2)),
     font = 5)
axis(side = 2, at = seq(-0.6,0.6,0.3), labels = as.character(seq(-0.6,0.6,0.3)),
     font = 5)
abline(h = seq(-0.6,0.6,0.3), col = "gray75", lty = 2)
abline(v = signif(seq(min(fitted(model3)), max(fitted(model3)), length = 7),
                  digits = 2), col = "gray80", lty = 2)
abline(0,0, lty = 2, col = "gray45")
points(fitted(model3), residuals(model3), col = addTrans("orange",120),
       pch = 19)
points(fitted(model3), residuals(model3), col = "orange")
panel.smooth(fitted(model3), residuals(model3), col = "orange",cex = 1,
             col.smooth = "seagreen", span = 2/3, iter = 3)
```

Sample Data Analysis Practice

```
dat <- read.csv("http://www.stat.cmu.edu/~larry/=stat401/abalone.csv")
library(knitr)
data.summary <- data.frame(Minimum = sapply(X = 1:2, FUN = function(X) min(dat[,X])),
                           Mean = sapply(X = 1:2, FUN = function(X) mean(dat[,X])),
                           Median = sapply(X = 1:2, FUN = function(X) median(dat[,X])),
                           Maximum = sapply(X = 1:2, FUN = function(X) max(dat[,X])),
                           Variance = sapply(X = 1:2, FUN = function(X) var(dat[,X])))
row.names(data.summary) <- c("Height (mm)", "Rings")
kable(data.summary, row.names = TRUE, digits = 4,
      caption = "Summary Statistics for Abalone Data Set")
```

```
par(mfrow=c(1,2))
hist(dat$Height, breaks = 75, xaxt = "n", yaxt = "n", main = "", xlab="",
     ylab="", col = NA, axes = FALSE, probability = TRUE)
axis(side = 1, at = seq(0,1,0.2), as.character(seq(0,1,0.2)), font = 5)
axis(side = 2, at = seq(0,10,2), as.character(seq(0,10,2)), font = 5)
abline(h = seq(0,10,2), v = seq(0,1,0.2), col = "gray70", lty = 2)
hist(dat$Height, breaks = 75, xaxt = "n", yaxt = "n", main = "", xlab="",
     ylab="", col = "orange", axes = FALSE, probability = TRUE, add = TRUE,
     border = addTrans("black",150))
dens <- density(dat$Height, bw = 0.02)
lines(x = dens$x, y = dens$y, col = "seagreen", lwd = 3)
mtext(side = 1, text = "Height", font = 3, line = 2.25)
mtext(side = 2, text = "Density", font = 3, line = 2.5)

barplot(c(table(dat$Rings)[1:27],0,table(dat$Rings)[28]), col = NA, xlab = "",
        ylab = "", ylim = c(0,700), xaxt = "n", yaxt = "n")
abline(h = seq(0,700,100),col = "gray70", lty = 2)
barplot(c(table(dat$Rings)[1:27],0,table(dat$Rings)[28]), col = "orange",
        xlab = "", ylab = "", add = TRUE, ylim = c(0,700), xaxt = "n", yaxt = "n")
mids <- barplot(c(table(dat$Rings)[1:27],0,table(dat$Rings)[28]),
                plot = FALSE)
axis(side = 1, at = mids[c(1,seq(5,25,5),29)], labels = c(1,seq(5,25,5),29),
     font = 5, tick = FALSE, line = -0.75)
axis(side = 2, at = seq(0,700,100), labels = as.character(seq(0,700,100)),
     font = 5)
mtext(side = 1, text = "Rings", font = 3, line = 1.5)
mtext(side = 2, text = "Frequency", font = 3, line = 3)
```

```
par(mfrow=c(1,2))
with(dat, plot(Height, Rings, col = addTrans("orange",120), pch = 19,
               cex = 0.8, axes = FALSE, xlab="",ylab=""))
axis(side = 1, at = seq(-0.2,1.2,0.2), as.character(seq(-0.2,1.2,0.2)),
     ont = 5)
axis(side = 2, at = c(1,seq(5,25,5),29), as.character(c(1,seq(5,25,5),29)),
     font = 5)
abline(v = seq(0,1,0.2), h = c(1,seq(5,25,5),29), col = "gray70", lty = 2)
mtext(side = 2, text = "Rings", font = 3, line = 3)
mtext(side = 1, text = "Height", font = 3, line = 3)
with(dat, points(Height, Rings, col = addTrans("orange",120), cex = 0.8,
                 pch = 19))
with(dat, points(Height, Rings, col = "orange", pch = 1, cex = 0.8))
```

```
model4 <- with(dat, lm(Rings ~ Height))
abline(model4, col = "seagreen", lwd = 2)

plot(model4, which = 1, col = NA, pch = 19, axes = FALSE,
     add.smooth = FALSE, font.lab= 3, caption = "")
axis(side = 1, at = round(seq(min(fitted(model4)), max(fitted(model4)), length = 7)),
     as.character(round(seq(min(fitted(model4)), max(fitted(model4)), length = 7))),
     font = 5)
axis(side = 2, at = seq(-70,30,10), labels = as.character(seq(-70,30,10)),
     font = 5)
abline(h = seq(-70,30,10), col = "gray70", lty = 2)
abline(v = round(seq(min(fitted(model4)), max(fitted(model4)), length = 7)),
       col = "gray70", lty = 2)
abline(0,0, lty = 2, col = "gray45")
points(fitted(model4), residuals(model4), col = addTrans("orange",120), pch = 19,
       cex = 0.5)
points(fitted(model4), residuals(model4), col = "orange", cex = 0.5)
panel.smooth(fitted(model4), residuals(model4), col = "orange",cex = 0.5,
             col.smooth = "seagreen", span = 2/3, iter = 3)
```

```
par(mfrow=c(1,2))
dat[c(which(dat$Height > 0.4 | dat$Height < 0.0184)),] <- NA
dat <- na.omit(dat)
transformed.height <- log(dat$Height)
log.likelihood <- with(dat, boxcox(Rings ~ transformed.height))
lambda.opt <- log.likelihood$x[which.max(log.likelihood$y)]

transformed.rings <- dat$Rings ^ (lambda.opt)
plot(transformed.height, transformed.rings, col = addTrans("orange",120), pch = 19,
     cex = 0.8, axes = FALSE, xlab="",ylab="")
axis(side = 1, at = seq(-5,0,1), as.character(seq(-5,0,1)), font = 5)
axis(side = 2, at = seq(0.30,1,0.1), as.character(seq(0.3,1,0.1)), font = 5)
abline(v = seq(-5,0,1), h = seq(0.3,1,0.1), col = "gray70", lty = 2)
mtext(side = 2, text = "Rings transformation", font = 3, line = 3)
mtext(side = 1, text = "Height transformation", font = 3, line = 3)
with(dat, points(transformed.height, transformed.rings, col = addTrans("orange",120),
                 cex = 0.8, pch = 19))
with(dat, points(transformed.height, transformed.rings, col = "orange", pch = 1,
                 cex = 0.8))
model4 <- lm(transformed.rings ~ transformed.height)
abline(model4, col = "seagreen", lwd = 2)
```

```
layout(matrix(c(1,2,3,3), 2, 2, byrow = T), widths = c(4,4,4))
model5 <- lm(transformed.rings ~ transformed.height)
plot(model5, which = 1, col = NA, pch = 19, axes = FALSE,
     add.smooth = FALSE, caption = "", font.lab = 3)
axis(side = 1, at = seq(0.35,0.8,0.05),as.character(seq(0.35,0.8,0.05)),font = 5)
axis(side = 2, at = seq(-0.3,0.2,0.05), labels = as.character(seq(-0.3,0.2,0.05)),
     font = 5)
abline(h = seq(-0.3,0.2,0.05), col = "gray75", lty = 2)
abline(v = seq(0.35,0.8,0.05), col = "gray75", lty = 2)
abline(0,0, lty = 2, col = "gray45")
points(fitted(model5), residuals(model5), col = addTrans("orange",120), pch = 19)
points(fitted(model5), residuals(model5), col = "orange")
```

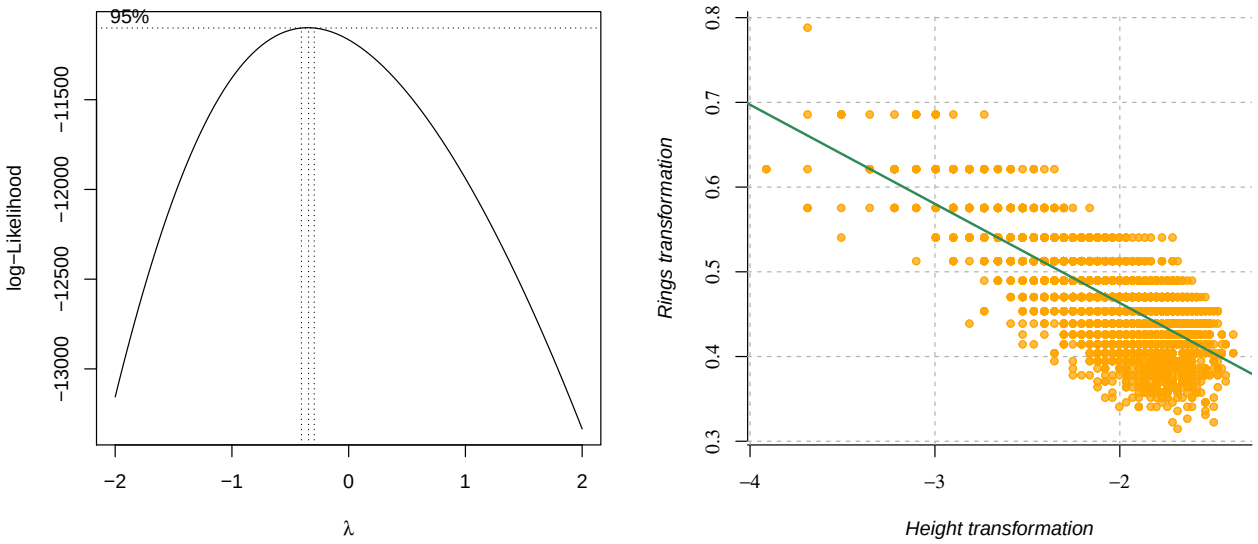
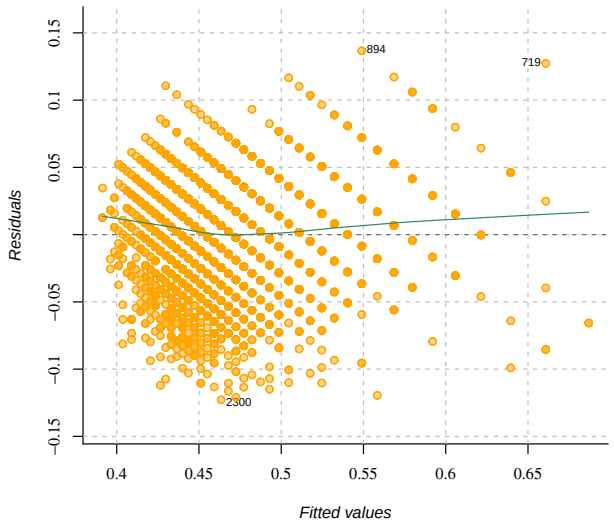


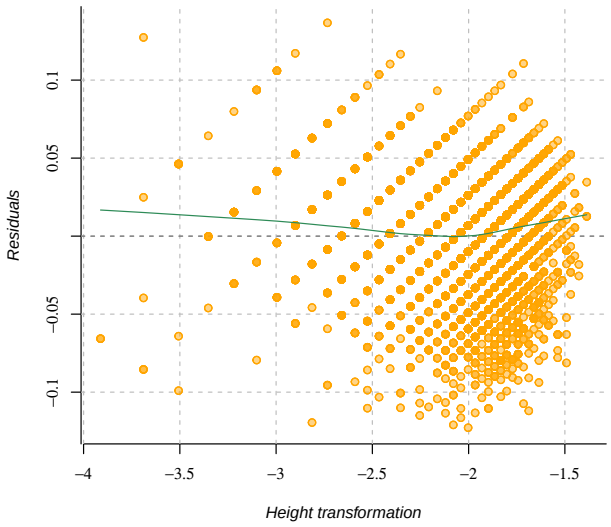
Figure 9: Log-likelihood of Box-Cox Power Transformation and Fitted Regression using the Optimal Power

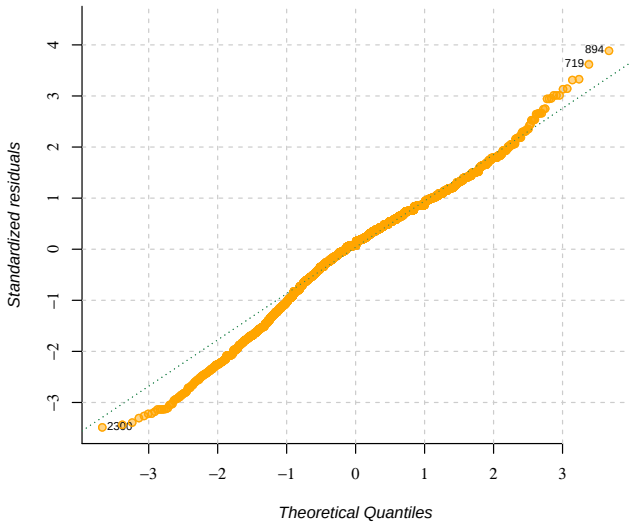
```
panel.smooth(fitted(model5), residuals(model5), col = "orange",cex = 1,
             col.smooth = "seagreen", span = 2/3, iter = 3)

plot(transformed.height, residuals(model5),
     col = NA, pch = 19, axes = FALSE, xlab = "", ylab = "")
axis(side = 1, at = seq(-4,-1,0.5),as.character( seq(-4,-1,0.5)),font = 5)
axis(side = 2, at = seq(-0.15,0.15,0.05), labels = as.character(seq(-0.15,0.15,0.05)),
     font = 5)
abline(h = seq(-0.15,0.15,0.05), col = "gray75", lty = 2)
abline(v = seq(-4,-1,0.5), col = "gray75", lty = 2)
abline(0,0, lty = 2, col = "gray45")
points(transformed.height, residuals(model5),
       col = addTrans("orange",120), pch = 19)
points(transformed.height, residuals(model5),
       col = "orange")
mtext(side = 2, text = "Residuals", font = 3, line = 3, cex = 0.85)
mtext(side = 1, text = "Height transformation", font = 3, line = 3, cex = 0.85)
panel.smooth(transformed.height, residuals(model5),
             col = "orange",cex = 1,col.smooth = "seagreen", span = 2/3, iter = 3)

par(mar = c(5,20,4,18) + 0.1)
plot(model5, which = 2, col = addTrans("orange",120), pch = 19, axes = FALSE,
     qqline = FALSE, cex.id = 0.75, caption = "", font.lab = 3)
axis(side = 1, at = seq(-8,3,1), labels = as.character(seq(-8,3,1)), font = 5)
axis(side = 2, at = seq(-8,6,1), labels = as.character(seq(-8,6,1)), font = 5)
qqline(scale(residuals(model5)), col = "seagreen", lty = 3)
secondpts <- qqnorm(scale(residuals(model5)), plot.it = FALSE)
abline(v = seq(-5,3,1), lty = 2, col = "gray80")
abline(h = seq(-8,6,1), lty = 2, col = "gray80")
points(secondpts, col = addTrans("orange",120))
points(secondpts, col = "orange")
qqline(scale(residuals(model5)), col = "seagreen", lty = 3)
```








```
dat[c(which(dat$Height > 0.4 | dat$Height < 0.0184)),] <- NA
dat <- na.omit(dat)
transformed.rings <- dat$Rings
transformed.height <- dat$Height
par(mfrow=c(2,2))
plot(transformed.height, transformed.rings, col = addTrans("orange",120), pch = 19,
     cex = 0.8, axes = FALSE, xlab="",ylab="")
axis(side = 1, at = seq(0,0.25,0.05), as.character(seq(0,0.25,0.05)), font = 5)
axis(side = 2, at = seq(0,30,5), as.character(seq(0,30,5)), font = 5)
abline(v = seq(0,0.25,0.05), h = seq(0,30,5), col = "gray70", lty = 2)
mtext(side = 2, text = "Rings", font = 3, line = 3)
mtext(side = 1, text = "Height", font = 3, line = 3)
with(dat, points(transformed.height, transformed.rings, col = addTrans("orange",120),
                 cex = 0.8, pch = 19))
with(dat, points(transformed.height, transformed.rings, col = "orange", pch = 1,
                 cex = 0.8))
model4 <- lm(transformed.rings ~ transformed.height - 1)
abline(model4, col = "seagreen", lwd = 2)
```

```
model5 <- lm(transformed.rings ~ transformed.height)
plot(model5, which = 2, col = addTrans("orange",120), pch = 19, axes = FALSE,
     qqline = FALSE, cex.id = 0.75, caption = "", font.lab = 3)
axis(side = 1, at = seq(-4,4,1), labels = as.character(seq(-4,4,1)), font = 5)
axis(side = 2, at = seq(-4,7,1), labels = as.character(seq(-4,7,1)), font = 5)
qqline(scale(residuals(model5)), col = "seagreen", lty = 3)
secondpts <- qqnorm(scale(residuals(model5)), plot.it = FALSE)
abline(v = seq(-4,4,1), lty = 2, col = "gray80")
abline(h = seq(-4,7,1), lty = 2, col = "gray80")
points(secondpts, col = addTrans("orange",120))
points(secondpts, col = "orange")
qqline(scale(residuals(model5)), col = "seagreen", lty = 3)

#layout(matrix(c(1,2,3,3), 2, 2, byrow = T), widths = c(4,4,4))
plot(model5, which = 1, col = NA, pch = 19, axes = FALSE,
     add.smooth = FALSE, caption = "", font.lab = 3)
axis(side = 1, at = seq(0,20,5),as.character(seq(0,20,5)),font = 5)
axis(side = 2, at = seq(-12,20,4), labels = as.character(seq(-12,20,4)),
     font = 5)
abline(h = seq(-8,20,4), col = "gray75", lty = 2)
abline(v = seq(-8,20,4), col = "gray75", lty = 2)
abline(0,0, lty = 2, col = "gray45")
points(fitted(model5), residuals(model5), col = addTrans("orange",120), pch = 19)
points(fitted(model5), residuals(model5), col = "orange")
panel.smooth(fitted(model5), residuals(model5), col = "orange",cex = 1,
             col.smooth = "seagreen", span = 2/3, iter = 3)

plot(transformed.height, residuals(model5),
     col = NA, pch = 19, axes = FALSE, xlab = "", ylab = "")
axis(side = 1, at = seq(0,0.25,0.05),as.character( seq(0,0.25,0.05)),font = 5)
axis(side = 2, at = seq(-8,20,4), labels = as.character(seq(-8,20,4)),
     font = 5)
abline(h = seq(-8,20,4), col = "gray75", lty = 2)
abline(v = seq(0,0.25,0.05), col = "gray75", lty = 2)
abline(0,0, lty = 2, col = "gray45")
points(transformed.height,
       residuals(model5),
       col = addTrans("orange",120), pch = 19)
points(transformed.height,
       residuals(model5),
       col = "orange")
mtext(side = 2, text = "Residuals", font = 3, line = 3, cex = 0.85)
mtext(side = 1, text = "Height", font = 3, line = 3, cex = 0.85)
panel.smooth(transformed.height,
             residuals(model5),
             col = "orange",cex = 1,col.smooth = "seagreen", span = 2/3, iter = 3)
#par(mar = c(5,20,4,18) + 0.1)
```