Homework 7

Advanced Methods for Data Analysis (36-402)

Due Friday March 22, 2019, at 6:00 PM

See the syllabus for general instructions regarding homework. Note that you should **always show all your work** and submit **exactly two files** (Rmd or R file as *code*, and knitted or scanned/merged pdf or html as *writeup*). Make sure that everything you submit is readable.

Problem 1: Cat Data Again

Part a

Fit a linear regression model for `Hwt`, in which `Bwt` interacts with `Sex` and the intercept is forced to be zero. Why is the forcing the intercept to be zero a reasonable thing to do?

Hint: There are some subtleties regarding how to force the intercept to be zero, make sure you are not accidentally introducing intercept terms by using the wrong `R` syntax.

Solution

```
library(MASS)
data(cats)

fit <- lm(Hwt~0+Bwt:Sex,data=cats)
summary(fit)
```

```
##
## Call:
## lm(formula = Hwt ~ 0 + Bwt:Sex, data = cats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4841 -0.9929 -0.1036  0.9879  5.2330
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## Bwt:SexF  3.88345    0.08925   43.51   <2e-16 ***
## Bwt:SexM  3.91461    0.05024   77.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.453 on 142 degrees of freedom
## Multiple R-squared:  0.9825, Adjusted R-squared:  0.9822
## F-statistic:  3982 on 2 and 142 DF,  p-value: < 2.2e-16
```

Forcing the intercept to be zero makes sense because we would expect that if body weight is equal to 0 so is heart weight.

Part b

Under the assumption that the residuals are normally distributed, conduct a statistical test to find out whether there is evidence that the slope coefficient for `Bwt` differs for female and male cats. Make sure you state the hypothesis, the null distribution, the value of the test statistic and the pvalue.

Solution

There are many parametrization of the null and alternative models. For example, we could have

$$\begin{aligned}
H_0 &: \mathbb{E}(Y|\texttt{Bwt}, \texttt{Sex}) = \alpha_1 \texttt{Bwt} \\
H_1 &: \mathbb{E}(Y|\texttt{Bwt}, \texttt{Sex}) = \alpha_1 \texttt{Bwt} + \alpha_2 \texttt{Bwt} \cdot \mathbb{1}\{Sex = Female\}
\end{aligned}$$

so we would be ineterested in testing the condition $\alpha_2 = 0$. Alternatively, we could specify models

$$\begin{aligned}
H_0 &: \mathbb{E}(Y|\texttt{Bwt}, \texttt{Sex}) = \alpha_1 \texttt{Bwt} \\
H_1 &: \mathbb{E}(Y|\texttt{Bwt}, \texttt{Sex}) = \alpha_1 \texttt{Bwt} \cdot \mathbb{1}\{Sex = Male\} + \alpha_2 \texttt{Bwt} \cdot \mathbb{1}\{Sex = Female\}
\end{aligned}$$

Notice that $\mathbb{1}\{Sex = Male\} = 1 - \mathbb{1}\{Sex = Female\}$:

$$\begin{aligned}
H_0 &: \mathbb{E}(Y|\texttt{Bwt}, \texttt{Sex}) = \alpha \texttt{Bwt} \\
H_1 &: \mathbb{E}(Y|\texttt{Bwt}, \texttt{Sex}) = \alpha_1 \texttt{Bwt} \cdot \mathbb{1}\{Sex = Male\} + \alpha_2 \texttt{Bwt} \cdot \mathbb{1}\{Sex = Female\} \\
&= \alpha_1 \texttt{Bwt} + (\alpha_2 - \alpha_1) \texttt{Bwt} \cdot \mathbb{1}\{Sex = Female\}
\end{aligned}$$

So, the two models are exactly the same, but only one parametrization makes the null model nested in the alternative model. Testing using the second set of models can be done via the `linearHypothesis` function in R. Ask Professor Lee or the TAs for an explanation of how this works. Alternatively, you could appeal to asymptotic normality of $\hat{\alpha}$ and compare the test statistic $\frac{\hat{\alpha_1} - \hat{\alpha_2}}{\sqrt{\hat{\text{Var}}(\hat{\alpha_1} - \hat{\alpha_2})}}$ to a standard normal distribution.

```
fit_null <- lm(Hwt~0+Bwt,data=cats)
fit_alt <- lm(Hwt~0+Bwt+Bwt:Sex,data=cats)
anova(fit_null, fit_alt)
```

```
## Analysis of Variance Table
##
## Model 1: Hwt ~ 0 + Bwt
## Model 2: Hwt ~ 0 + Bwt + Bwt:Sex
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    143 300.09
## 2    142 299.90  1   0.19544 0.0925 0.7614
```

The value of the t-statistic is 0.304, which, compared to a the t-distribution with $n - 1 = 142$ degrees of freedom, yields a p-value ≈ 0.76. Therefore, we fail to reject the null hypothesis that the slope for `Bwt` differs between males and females cats.

Part c

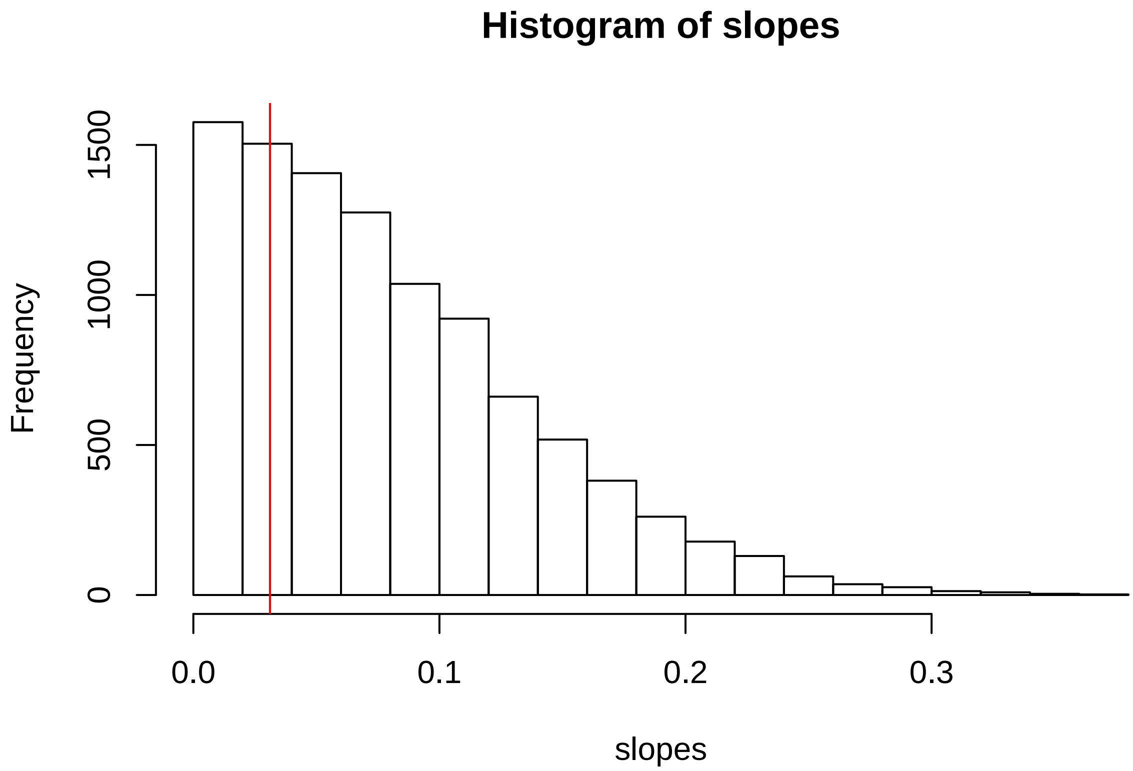
Answer the same hypothesis as in Part b, but this time perform a permutation test instead. You might find previous HWs and Canvas notes on permutation tests useful.

Solution

```
## Total number of permutations is too large
B <- 10000
sex_mat <- replicate(B, sample(cats$Sex, replace = FALSE))
get_slope <- function(sex){

  dat <- data.frame(Hwt=cats$Hwt, Bwt=cats$Bwt, Sex=as.factor(sex))
  dat$Sex <- relevel(dat$Sex, ref="F")

  slope <- abs(coef(lm(Hwt~0+Bwt+Bwt:Sex, data=dat))["Bwt:SexF"])

  return(slope)

}

slope_obs <- get_slope(cats$Sex)
print(slope_obs)
```

```
##  Bwt:SexF
## 0.0311567
```

```
slopes <- apply(sex_mat, 2, get_slope)
hist(slopes)
abline(v=slope_obs, col="red")
```



```
print(mean(slopes>= slope_obs))
```

```
## [1] 0.7599
```

Part d

Recall what we found in Question 2, Part c of HW 6: the test using the boottrap rejected the null hypothesis at $\alpha = 0.05$. What does this new finding suggest?

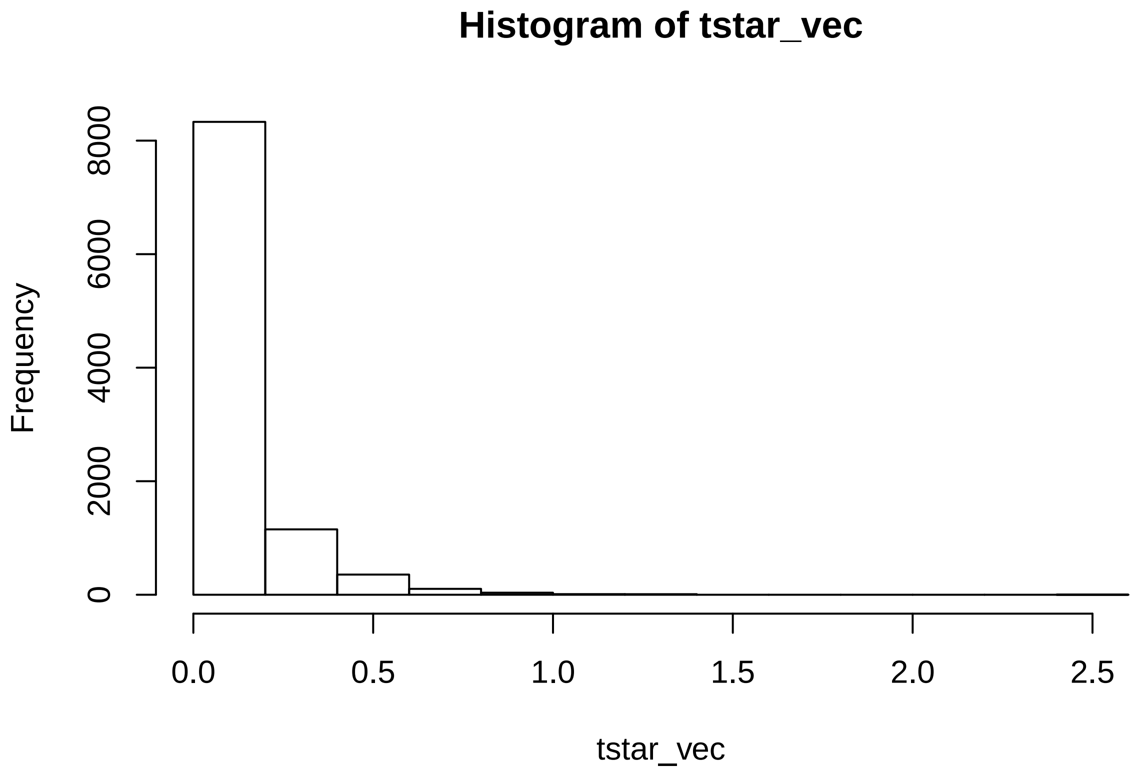
Hint: Think about what the null hypothesis was in HW 6 and what the null is now.

Solution

```
B <- 10000
get_tstar <- function(myform, xm, xf, betahat, resm, resf) {
  newym <- xm%*%betahat + sample(resm, length(resm), replace=TRUE)
  newfitm <- lm(formula=myform, data=data.frame(Hwt=newym,
                                              Bwt=cats$Bwt[cats$Sex == "M"]))
  boot_coeffsm <- coef(newfitm)

  newyf <- xf%*%betahat + sample(resf, length(resf), replace=TRUE)
  newfitf <- lm(formula=myform, data=data.frame(Hwt=newyf,
                                              Bwt=cats$Bwt[cats$Sex == "F"]))
  boot_coeffsf <- coef(newfitf)

  tstar <- (boot_coeffsf[1]-boot_coeffsm[1])^2
  names(tstar) <- NULL
  return(tstar)

}


## Testing just the intercept
myform_inter <- as.formula(Hwt~1)
catlm1=lm(myform_inter,data=cats)
modelf1=lm(myform_inter,data=cats[cats$Sex=="F",])
modelm1=lm(myform_inter,data=cats[cats$Sex=="M",])
resm1 <- resid(modelm1)
resf1 <- resid(modelf1)
bethat_null_inter <- catlm1$coef[1]

tstar_vec <- replicate(B, get_tstar(myform=myform_inter,
                                    xm=cbind(rep(1, length(resm1))),
                                    xf=cbind(rep(1, length(resf1))),
                                    betahat=bethat_null_inter,
                                    resm=resm1, resf=resf1))
tobs <- (modelf1$coef[1]-modelm1$coef[1])^2
names(tobs) <- NULL
print(tobs)
```

```
## [1] 4.496744
```

```
hist(tstar_vec)
abline(v=tobs, col="red")
```



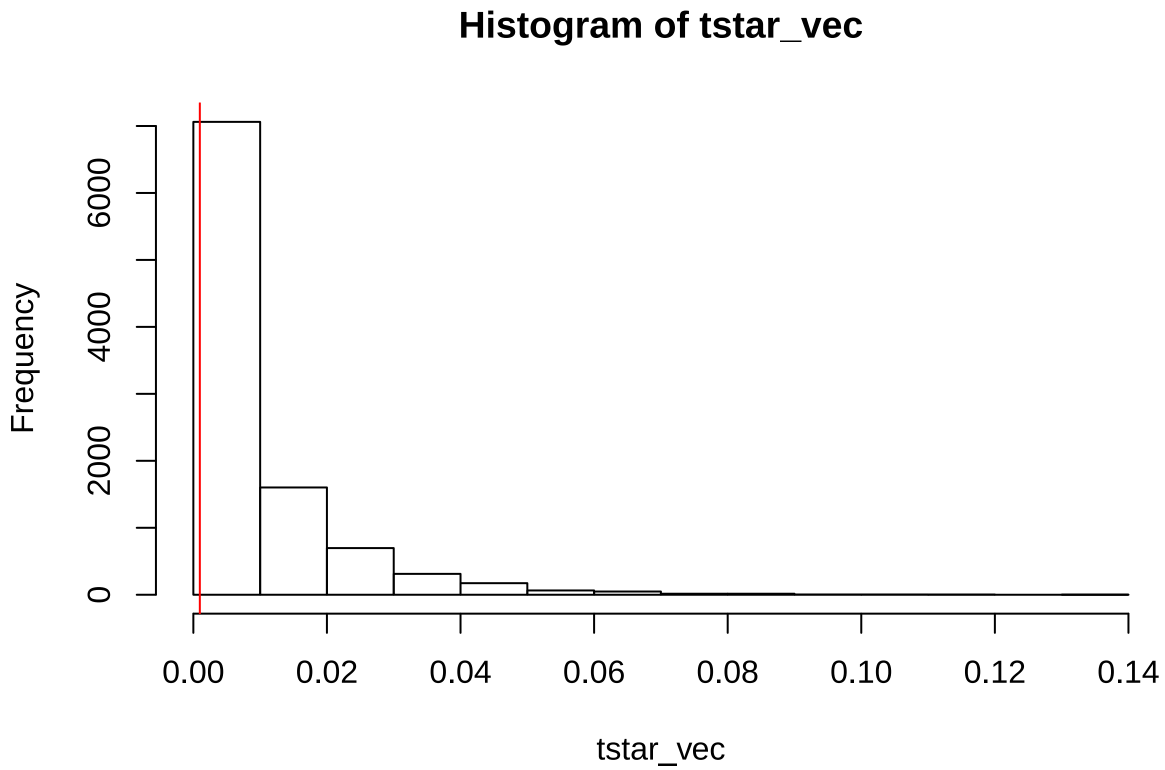
```
pvalue <- mean(tstar_vec >= tobs)
print(pvalue)
```

```
## [1] 0
```

```
## Testing just the slope
myform_slope <- as.formula(Hwt~0+Bwt)
catlm2=lm(myform_slope, data=cats)
modelf2=lm(myform_slope, data=cats[cats$Sex=="F",])
modelm2=lm(myform_slope, data=cats[cats$Sex=="M",])
resm2 <- resid(modelm2)
resf2 <- resid(modelf2)
bethat_null_slope <- catlm2$coef[1]

tstar_vec <- replicate(B, get_tstar(myform=as.formula(myform_slope),
                                    xm=cbind(cats$Bwt[cats$Sex=="M"]),
                                    xf=cbind(cats$Bwt[cats$Sex=="F"]),
                                    betahat=bethat_null_slope,
                                    resm=resm2, resf=resf2))
tobs <- (modelf2$coef[1]-modelm2$coef[1])^2
names(tobs) <- NULL
print(tobs)
```

```
## [1] 0.0009707399
```

```
hist(tstar_vec)
abline(v=tobs, col="red")
```



```
pvalue <- mean(tstar_vec >= tobs)
print(pvalue)
```

```
## [1] 0.7395
```

Part e

Look back at HW 6, Q2-Part c and check what the residual terms were in the null model. Is there a difference between the null models in the HW 6 question and the null model you fitted in Part a of this HW?

Problem 2: Problem 3 from HW 5 Revisited

For this problem, we use the `abalone` data set, which contains nine variables measured on 4173 abalones. Take a look at HW 5 for a description of the variables. Recall that we fitted the following models:

Model 1: A linear regression of log(`Shucked.weight`), on the logarithms of all three predictors.

Model 2: A kernel regression of `Shucked.weight`, on all three predictors: `Diameter`, `Length`, and `Height`.

Part a

Fit a smoothing spline to predict `Shucked.weight` from `Diameter` times `Length` times `Height`. Divide the product by 10^6 so that it is in dm^3. Just like in the previous HW, compute 95% pivotal confidence intervals for the regression function $r(x)$ at each x from 0 to 140 in steps of 2 (67 different values of x). Call this vector `x0`. Note that the `smooth.spline` command in R will choose the λ parameter by GCV, but we don't want to vary λ in each bootstrap sample. In later lectures, we will discuss how to properly tune λ, for now use $\lambda = 0.00004$.

For each bootstrap sample b, fit a smoothing spline, and predict the response `Shucked.weight` at each of the 111 values in x0. (We can call these $\hat{r}^*_b(x)$.) The command

```
pred=predict(ssmod,x0)$y
```

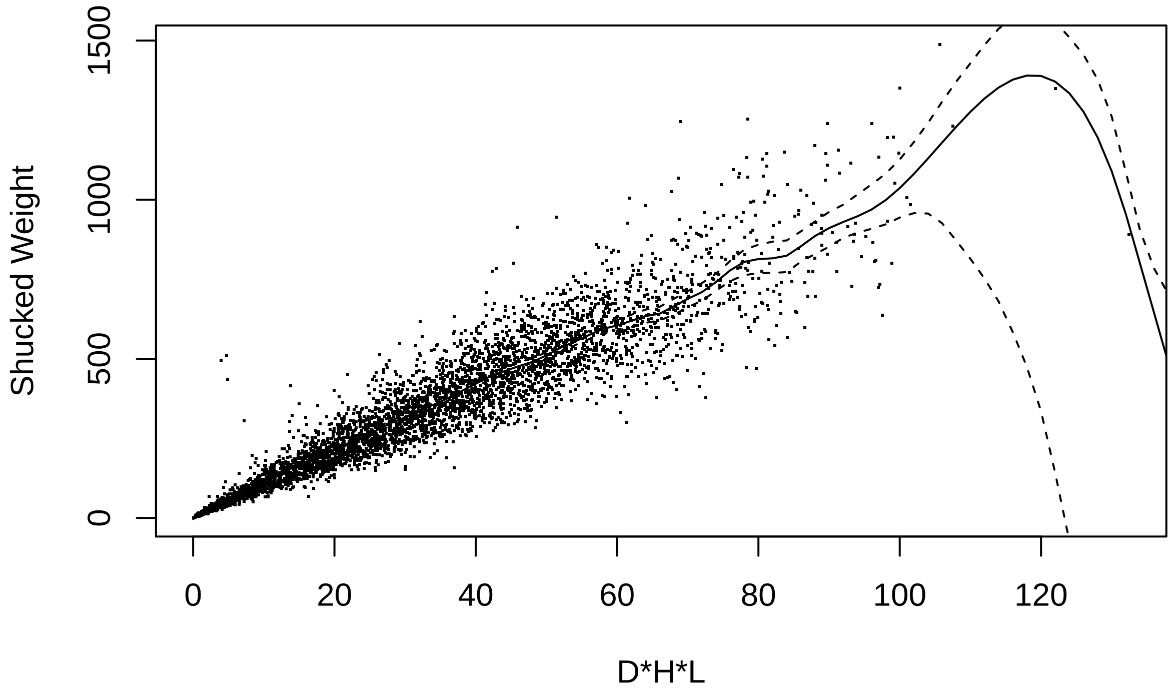
will produce the predictions requested, if the original fit is stored as `ssmod`. For each $x \in$ `x0`, compute a 95% pivotal bootstrap confidence interval for $r(x)$. Draw a plot with the original data and the estimated regression function $\hat{r}(x)$ for each x in `x0` added as a line. Finally, add to the plot the upper and lower endpoints of all of the pivotal bootstrap confidence intervals (using a different line type than used for $\hat{r}(x)$.)

Solution

```
abalone=read.csv("abalonemt.csv", header=TRUE)
lambda=0.00004
ssmod=smooth.spline(y=abalone$Shucked.weight,
                    x=abalone$Diameter*abalone$Height*abalone$Length/10^6, lambda=lambda)

# Get the predcitions from the original data
x0=seq(0, max(140), by=2)

pred=predict(ssmod,x0)$y

#  The bootstrap loop

bootfish=NULL

for(b in 1:1000){
      therows=sample(nrow(abalone),replace=TRUE)
      newfish=abalone[therows,]
      newfit=smooth.spline(y=newfish$Shucked.weight,
                           x=newfish$Diameter*newfish$Height*newfish$Length/10^6, lambda=lambda)
      bootfish=rbind(bootfish,predict(newfit,x=x0)$y)
}

#  Compute the quantiles

bootquant=apply(bootfish,2,quantile,prob=c(0.025,0.975))
dim(bootquant)
```

```
## [1]  2 71
```

```
#  Draw the requested plot

plot(x=abalone$Diameter*abalone$Heigh*abalone$Length/10^6,
     y=abalone$Shucked.weight,pch=".",xlab="D*H*L",ylab="Shucked Weight")
lines(x0,pred)
lines(x0,2*pred-bootquant[1,],lty=2)
lines(x0,2*pred-bootquant[2,],lty=2)
```



Part b

Now let's consider Model 1 again. Recall that one issue with transforming the response in a regression is the following. We defined $r(x)$ to be $\mathbb{E}[Y \mid X = x]$, the conditional mean of Y given $X = x$. In Model 1, we modeled

$$\log(Y) = \tilde{r}(x) + \varepsilon, \tag{1}$$

where ε is independent of X and has a distribution centered at 0. However, if the mean of ε is 0, $\exp\left(\tilde{r}(x)\right)$ is *not* the conditional mean of Y given $X = x$. We can still use $\exp\left(\widehat{\tilde{r}(x)}\right)$ as an estimator of $r(x) = \mathbb{E}[Y|X = x]$, but it will be biased. In the model of equation (1) assume only that the cases, i.e. (X, Y) pairs, are iid. Use the appropriate version of bootstrap to get an estimate of the bias of $\exp\left(\widehat{\tilde{r}(x)}\right)$ as an estimator of $r(x)$ for each three-dimensional vector x of predictors as defined by the rows of the following table:

Length	Diameter	Height
600	500	150
400	350	125
250	200	140

Say what assumption you are making about how the distribution of $\exp\left(\widehat{\tilde{r}(x)}\right)$ relates to the distribution of the bootstrap calculations $\exp\left(\widehat{\tilde{r}(x)}_b^*\right)$. Offer a reason for why the biases seem so different.

Note: In HW5 Part g (EC), we approximated $e^{\hat{r}(x)}$ via Taylor expansion so that it was easier to compute its variance. Now, we get rid of that approximating step by using the bootstrap.

Solution

Because of the appearent dependence of residual variance on overall level (see the plots of residuals versus fitted values,) the safest bet would be to resample cases. To make sure we know what we are doing, we are asked to compute biases for three estimators of three different "parameters." The parameters are $r(x)$ for each of the three x vectors in the table given. The corresponding estimators in Model 1 are $\widehat{r(x)} = \exp(\widehat{\tilde{r}(x)})$. In the other two models, the estimators are simply $\widehat{r(x)}$ computed directly without transformation. We will sample $B = 10000$ bootstrap samples of the data, and fit each of the three models with each sample b. We will then compute the $\widehat{r(x)}_b^*$ for each x and each model, noting that for Model 1 $\widehat{r(x)}_b^* = \exp(\widehat{\tilde{r}(x)}_b^*)$, where $\widehat{\tilde{r}(x)}_b^*$ is the prediction of $\log(Y)$ at x. We will be making the assumption that the distribution of $\exp(\widehat{\tilde{r}(x)}) - r(x)$ is similar to the distribution of $\widehat{r(x)}_b^*) - \widehat{r(x)}$ for each x and each model.

```
B=10000
x=data.frame(Length=c(600,400,250),Diameter=c(500,350,200),
    Height=c(150,125,140))
pred=NULL

# Loop through the bootstrap samples

for(b in 1:B){
      cases=sample(nrow(abalone),replace=TRUE)
      dataf=abalone[cases,]

# Fit and predict

    out=lm(log(Shucked.weight)~log(Length)+log(Diameter)+log(Height),
        data=dataf)
    pred=rbind(pred,exp(predict(out,newdata=x)))
}

# The predictions from the data
model1fit=lm(log(Shucked.weight)~log(Length)+log(Diameter)+log(Height),
data=abalone)
px1=exp(predict(model1fit,newdata=x))

# Bias estimates

apply(pred,2,mean)-px1
```

```
##           1           2           3
## -0.83336842 -0.57825511 -0.07422853
```

```
# Simulation standard error estimates

apply(pred,2,sd)/sqrt(B)
```

```
##          1          2          3
## 0.07023497 0.05189035 0.01481876
```

We see negative bias at all three x vectors, but the bias gets smaller as the predictors get smaller, which is where the fitted values are smaller and there is less variation in the response.

Have a great spring break!