Homework 6

Advanced Methods for Data Analysis (36-402)

Due Thursday February 28, 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: Bootstrapping a Cross-Validation

In Homework #5, we compared four models for predicting median house value from various predictors. Two of the models had R^2 and prediction errors that were very close, and it wasn't obvious that one model was better than the other. The bootstrap allows us to measure the uncertainty in the (cross-validation) mean-squared-error calculations to help decide whether one of the models is actually better than the other.

Part a

Read the two data files `housetrain.csv` and `housetest.csv` back into R and combine them into a single object using the `rbind()` function. This time, we are going to bootstrap the 5-fold cross-validation analysis for the models that were called Model 2 and Model 3 in Homework #5:

Model 2: A simple linear regression of `Median_house_value` on `Mean_household_income`.

Model 3: A multiple regression of `Median_house_value` on *both* `Median_household_income` and `Mean_household_income`.

Some of the code for 5-fold cross-validation from Homework #5 (with modifications) can be useful for this problem. Create $B = 200$ bootstrap samples each consisting of $n = 10605$ rows from the combined data set selected at random with replacement.

```
## some sample code
B <- 200; n <- nrow(dat)
boot_indices <- replicate(B, sample(1:n, n, replace=TRUE))
```

We will use the "resampling cases" form of the bootstrap (also called "resampling (X, Y) pairs" or nonparametric bootstrap).

For each bootstrap sample b, randomly divide the n observations into 5 disjoint sets of size 2121 each. Treating each of the 5 folds as test data and the other 4 as training data, calculate prediction error for each model and call the average of the 5 prediction errors for Model 2 $\widehat{MSE2}_b^*$. Call the average for Model 3 $\widehat{MSE3}_b^*$. Draw a histogram of the $\widehat{MSE2}_b^* - \widehat{MSE3}_b^*$ values. What visual evidence is there about whether one model is better?

Solution

```
traindat <- read.csv("housetrain.csv", header=TRUE)
testdat <- read.csv("housetest.csv", header=TRUE)
dat <- rbind(traindat, testdat)

B <- 200; n <- nrow(dat)
boot_indices <- replicate(B, sample(1:n, n, replace=TRUE))
get_errors <- function(boot_indices, nfold, dat) {
```

```
  n <- length(boot_indices)
  samp <- sample(rep(1:nfold, ceiling(n/nfold))[1:n])
  prederr1 <- prederr2 <- rep(NA, nfold)
  tempdata <- dat[boot_indices, ]

  for(j in 1:nfold) {
    traind <- tempdata[samp!=j, ]
    testd <- tempdata[samp==j, ]
    fit1 <- lm(Median_house_value~Mean_household_income, data=traind)
    pred1 <- predict(fit1, newdata=testd)
    prederr1[j] <- mean((pred1-testd$Median_house_value)^2)
    fit2 <- lm(Median_house_value~Mean_household_income+
                 Median_household_income, data=traind)
    pred2 <- predict(fit2,newdata=testd)
    prederr2[j] <- mean((pred2-testd$Median_house_value)^2)
  }

  return(mean(prederr1-prederr2))

}

test_errors <- apply(boot_indices, 2, get_errors, nfold=5, dat=dat)
print(mean(test_errors > 0))
```

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

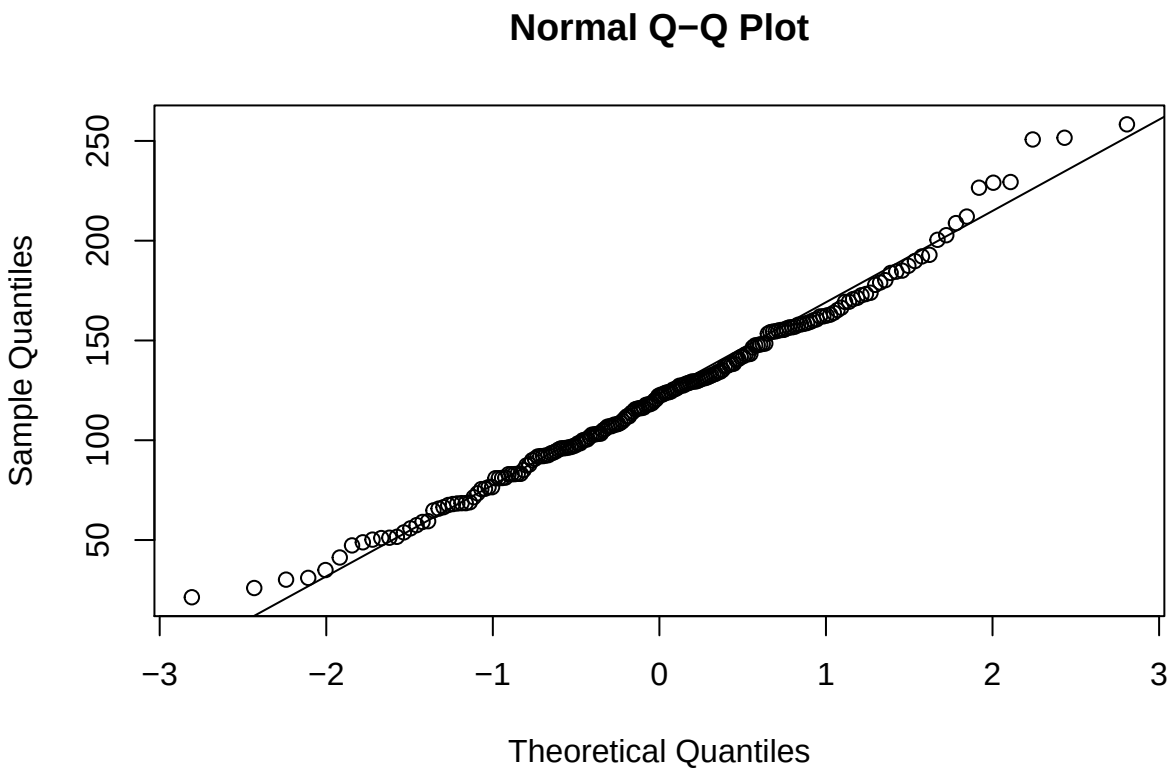
It looks like Model 3 is uniformly better than Model 2. Every difference is positive.

Part b

Let $T_b^* = \widehat{MSE2}_b^* - \widehat{MSE3}_b^*$ for $b = 1, \ldots, 200$. Draw a normal q-q plot of the T_b^* values and add the `qqline`. Do they look like a sample of normal random variables? Treat $T_1^*, \ldots, T_{200}^*$ as a random sample of normal random variables and test the null hypothesis that $\mathbb{E}(T_j^*) = 0$. Does one of the models look better than the other now?

Solution

```
qqnorm(test_errors)
qqline(test_errors)
```




```
t.test(test_errors)
```

```
##
##  One Sample t-test
##
## data:  test_errors
## t = 38.608, df = 199, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  115.8042 128.2705
## sample estimates:
## mean of x
##  122.0374
```

The q-q plot is remarkably straight, and the t test rejects the null hypothesis that $\mathbb{E}(T_j^*) = 0$ at virtually all levels. It looks like Model 3 is actually better at predicting.

Problem 2: Kernel Regression and the Bootstrap

This problem uses part of a set of data on abalone fishing in Australia (taken from http://archive.ics.uci.edu/ml/machine-learning-databases/abalone); available as `fishdata.csv` on Canvas with 1528 observations on 2 variables, with header. Each observation corresponds to a particular caught male abalone, and the columns correspond to the following attributes:

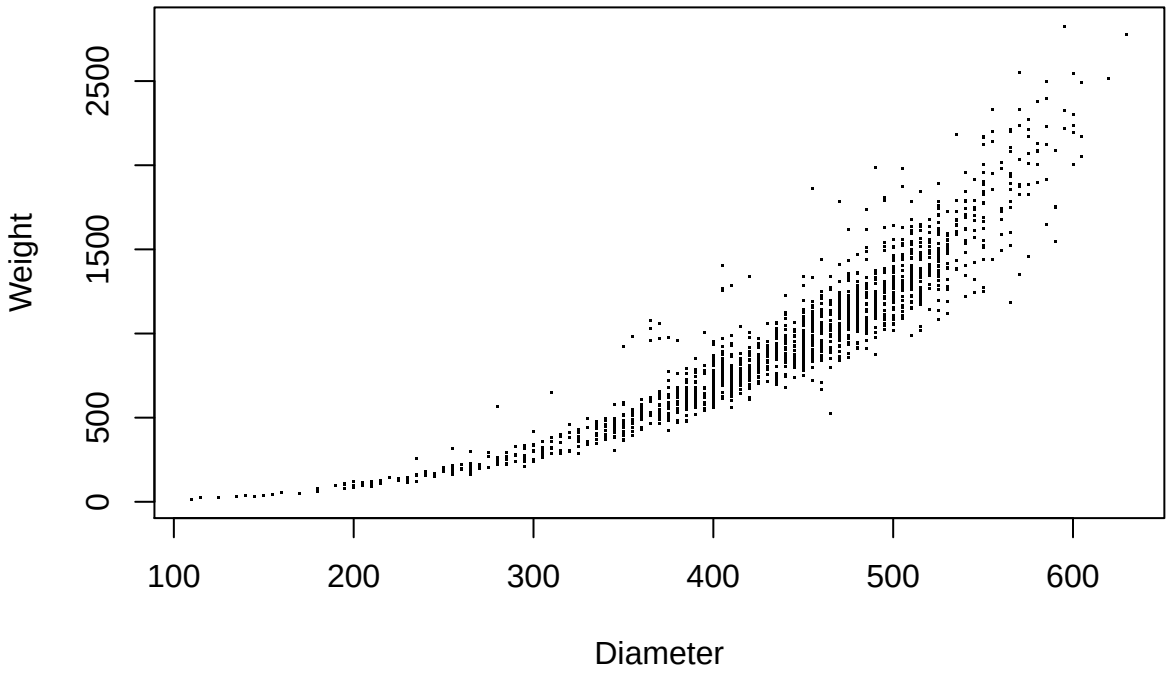
We will be predicting the `Weight` of male abalones from the diameter measurements. There are enough data values here to clutter a plot unless you use a small plotting symbol such as `pch="."`.

Part a

Plot the `Weight` versus `Diameter`. Does it look like a linear regression is likely to provide a good fit? Say why or why not.

Solution

```
dat <- read.csv("fishdata.csv", header=TRUE)
plot(x=dat$Diameter, y=dat$Weight, pch=".", xlab="Diameter", ylab="Weight")
```



There is an obvious curve here, so a linear regression will likely fit horribly without a transformation to the predictor and/or the response. Also, the variance of `Weight` seems to increase with `Diameter`, although some of the apparent increase might be due to there being more observations at some of the larger values of `Diameter`. There are ways to check if the variance really goes up. More on that later.

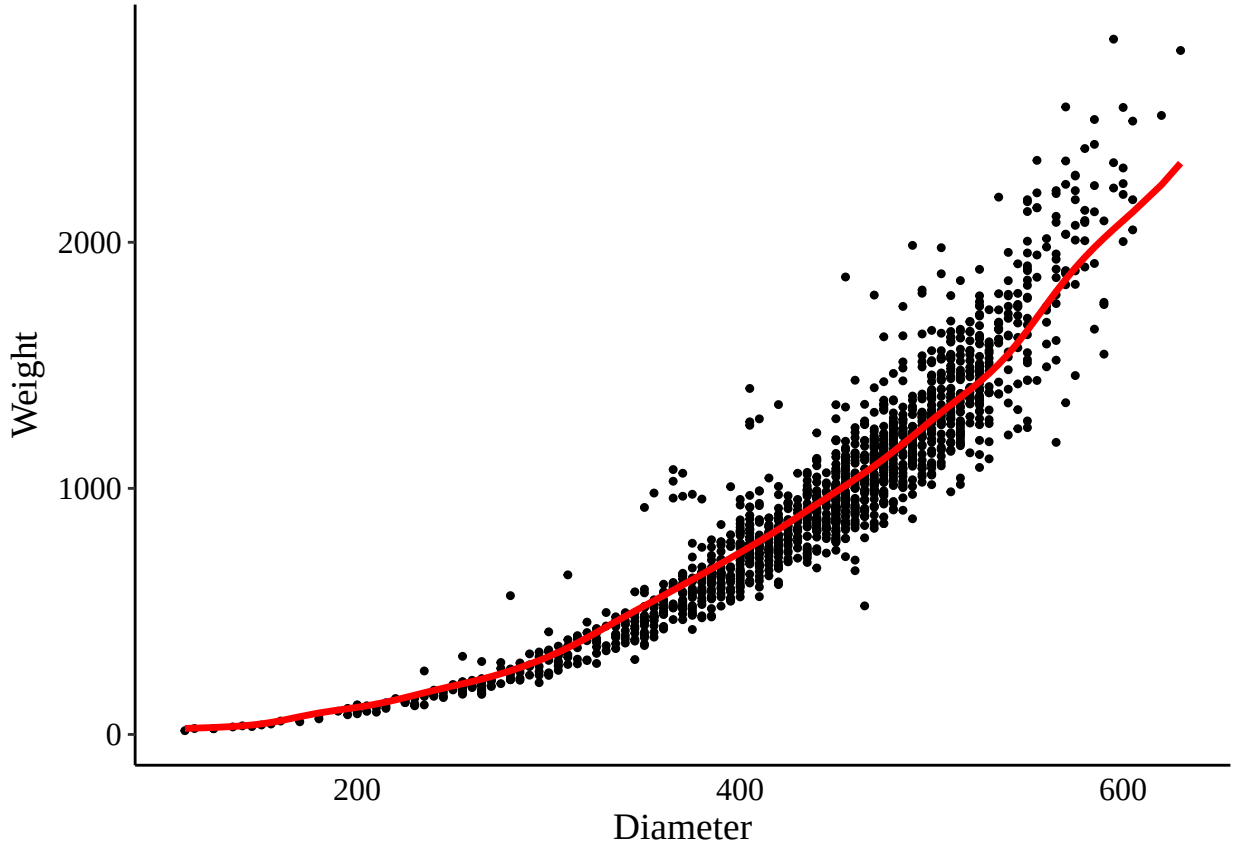
Part b

Fit a kernel regression of `Weight` on `Diameter`. As usual, let $r(x)$ denote the true regression function $\mathbb{E}\{Y|X=x\}$ and $\hat{r}(x)$ its nonparametric estimate. For this part and subsequent parts, use the heuristic choice for the bandwidth used in the previous HWs: the standard deviation of the predictor divided by $n^{1/5}$, where n is the size of dataset.

Solution

Name	Data Type	Meas.	Description
Diameter	continuous	mm	perpendicular to length
Weight	continuous	grams	whole abalone

```
library(np)
n <- nrow(dat)
fit <- npreg(Weight~Diameter, data=dat, bws=sd(dat$Diameter)/n^0.2)
library(ggplot2)
source("plot_theme.R")
fitted_vals <- fitted(fit)
ggplot(dat=dat, aes(x=Diameter, y=Weight)) +
  geom_point(size=0.9) +
  geom_line(aes(y=fitted_vals), color="red", size=1.2) +
  our_theme
```



Part c

In this problem, we will use the bootstrap to compute confidence intervals for $r(x)$ for each x from 100 to 650 in steps of 5 (111 different values of x). Call these x values `x0`. Use $B = 1000$ bootstrap samples of (`Diameter`,`Weight`) pairs drawn together from the empirical distribution of the data. (This corresponds to "resampling (X,Y) pairs" or "resampling cases" or nonparametric bootstrap in the language of bootstrapping regressions.) For each bootstrap sample b, fit a kernel regression as in part b, and predict the response `Weight` at each of the 111 values in `x0`. (We can call these $\hat{r}_b^*(x)$.)

For each x in `x0`, we are going to compute a 95% pivotal bootstrap confidence interval for $r(x)$. As you may recall from previous Statistics classes, a $1-\alpha$ confidence interval for a paremeter t_0 is a *random* interval that contains t_0 with probability $1-\alpha$. In other words, it is an interval that contains all the "plausible" values for t_0, where "plausible" means all values except those for which the probability of seeing the observed sample is less than or equal to α.

For each $x_i \in$ `x0`, you have B estimates $\hat{r}_b^*(x_i)$. Let $q_\tau(x_i)$ be the τ-quantile of the bootstrap distribution of $\hat{r}_b^*(x_i)$. A $1-\alpha$ pivotal bootstrap confidence interval fo $r(x_i)$ can be computed as

$$CI(x_i) = \left[2\hat{r}(x_i) - q_{\alpha/2}(x_i), 2\hat{r}(x_i) - q_{1-\alpha/2(x_i)}\right]$$

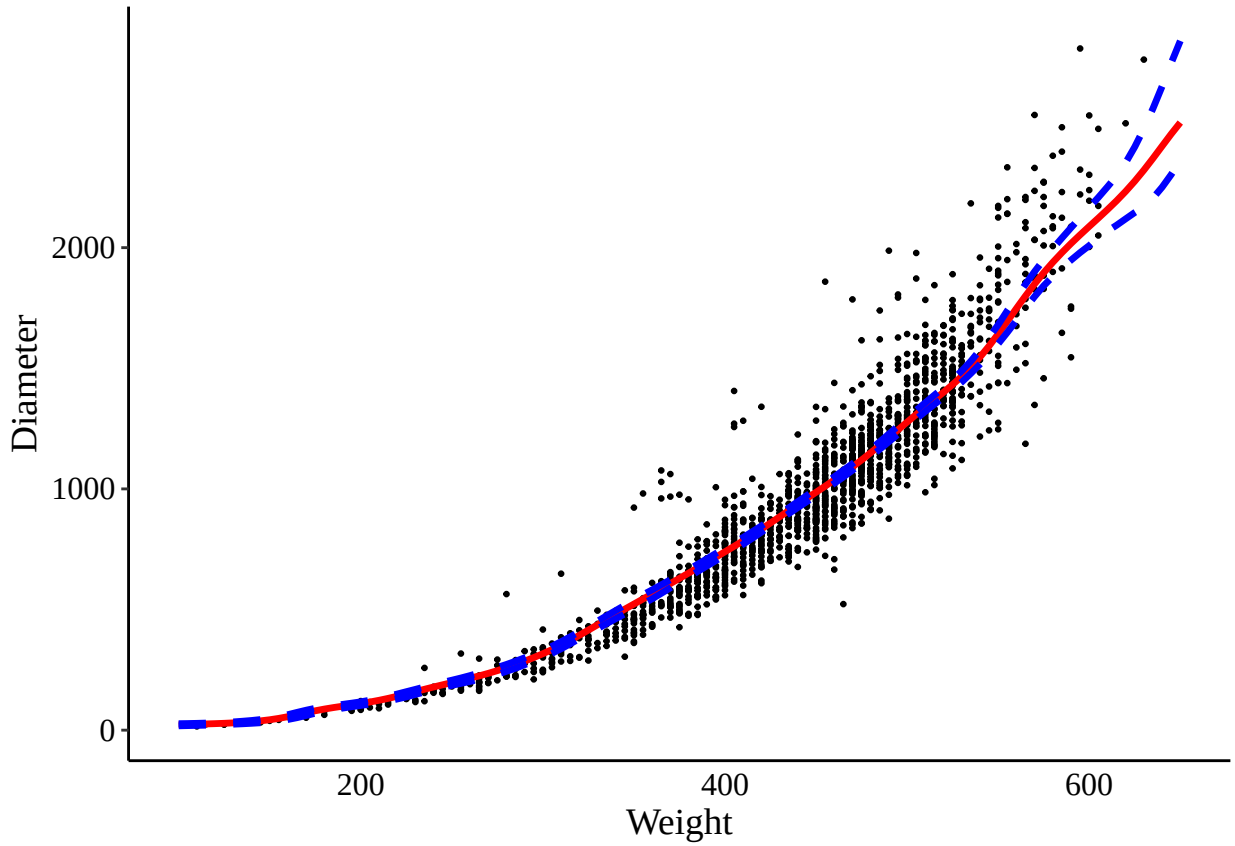
where $x_i \in$ `x0`.

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 $\hat{r}(x)$.

We will discuss in greater detail the pivotal bootstrap confidence interval in lecture on Tuesday. For now, take a look at Section 6.2.2 in Shalizi's book.

Solution

```
B <- 1000; n <- nrow(dat)
boot_indices <- replicate(B, sample(1:n, n, replace=TRUE))
xvals <- data.frame(Diameter=seq(100, 650, 5))
get_predictions <- function(boot_indices, xtest, dat) {

  bootdat <- dat[boot_indices, ]
  n <- nrow(bootdat)
  fit <- npreg(Weight~Diameter, data=bootdat, bws=sd(bootdat$Diameter)/n^0.2,
               newdata=xvals)
  return(fit$mean)

}
boot_preds <- apply(boot_indices, 2, get_predictions, xtest=xvals, dat=dat)
preds <- npreg(Weight~Diameter, data=dat, bws=sd(dat$Diameter)/n^0.2,
               newdata=xvals)
bootquant <- apply(boot_preds, 1, quantile, prob=c(0.025, 0.975))
ggplot() +
  geom_point(aes(x=dat$Diameter, y=dat$Weight), size=0.5) +
  geom_line(aes(x=xvals$Diameter, y=preds$mean), color="red", size=1.2) +
  geom_line(aes(x=xvals$Diameter, y=2*preds$mean-bootquant[1, ]),
            color="blue", size=1.2, linetype="dashed") +
  geom_line(aes(x=xvals$Diameter, y=2*preds$mean-bootquant[2, ]),
            color="blue", size=1.2, linetype="dashed") +
  labs(x="Weight", y="Diameter") +
  our_theme
```

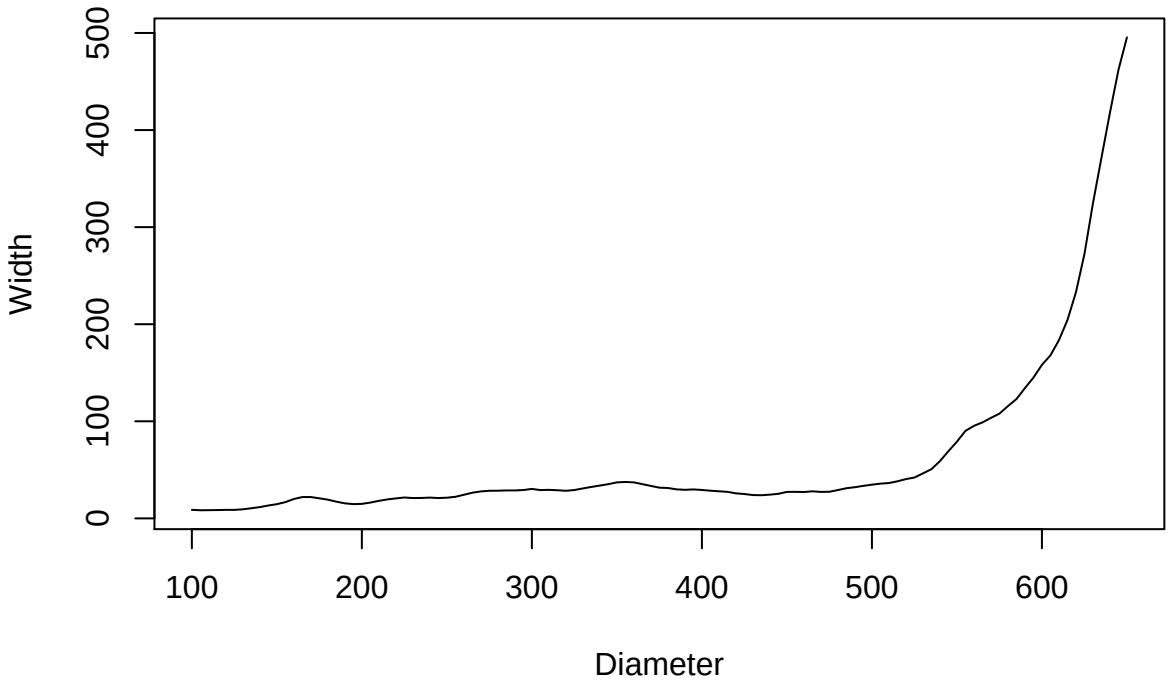


Part d (Extra Credit)

There are some x values where the confidence intervals are wider than they are for the other x values. Give a plausible explanation for this based on all of the analysis in the earlier parts of this problem.

Solution

```
plot(xvals$Diameter, bootquant[2,]-bootquant[1,],
     type="l",ylab="Width",xlab="Diameter")
```



The widths are essentially constant until `Diameter` gets a bit above 500, and then the width skyrockets. The numbers of observations at each `Diameter` value drops off quite a bit in this range, hence there is more uncertainty about $r(x)$ for large x.

Problem 3: Omitted Variables and Causal Regression

Return to the `cats` data that are available in `library(MASS)`. Check the lecture materials for some example calculations using these data. In particular, a linear regression of heart weight `Hwt` on body weight `Bwt` is explored. In this problem, we also consider predicting heart weight Y from body weight X, but we take into account the third variable `Sex` of the cats.

Part a

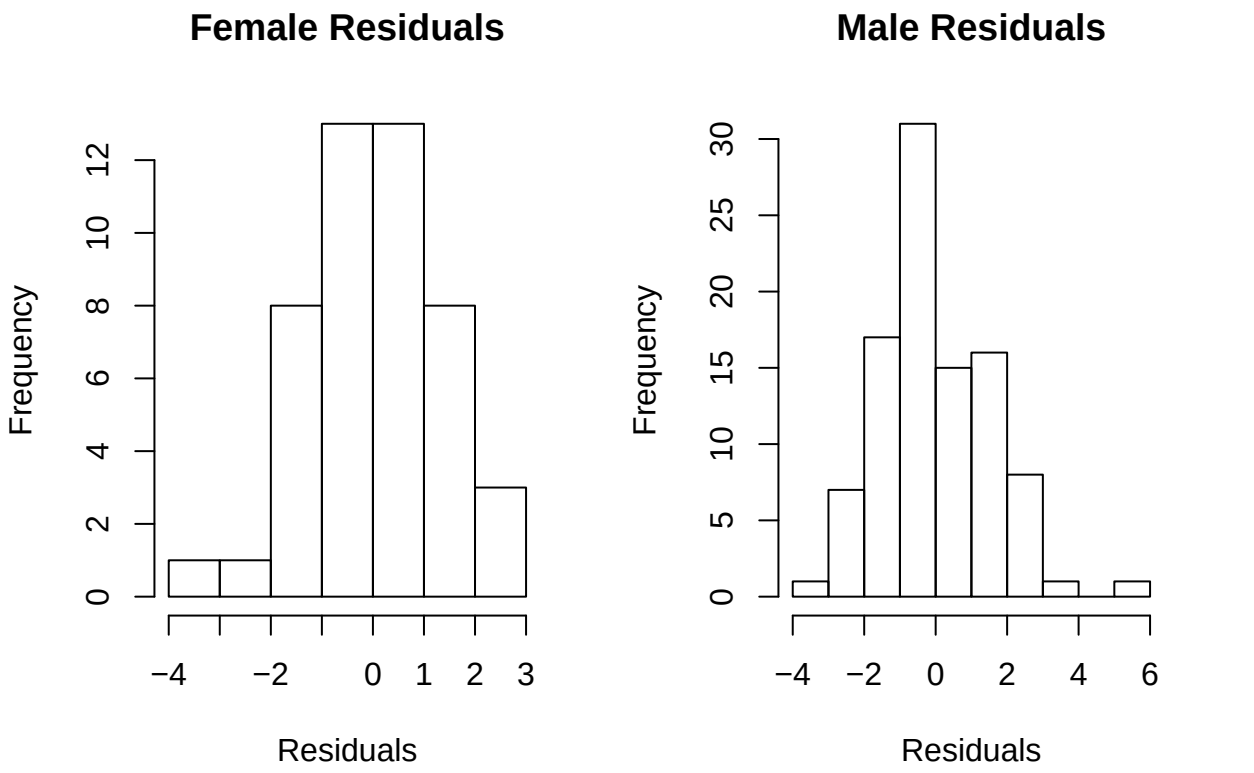
Fit the linear regression of `Hwt` on `Bwt`. Examine the residuals separately for male and female cats, and comment on what you see.

Examine the empirical distribution of body weight separately for male and female cats, and comment on what you see.

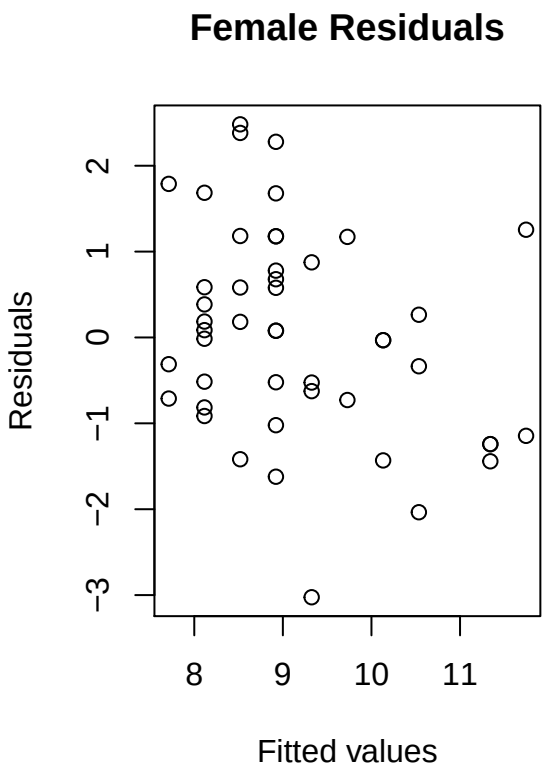
Solution

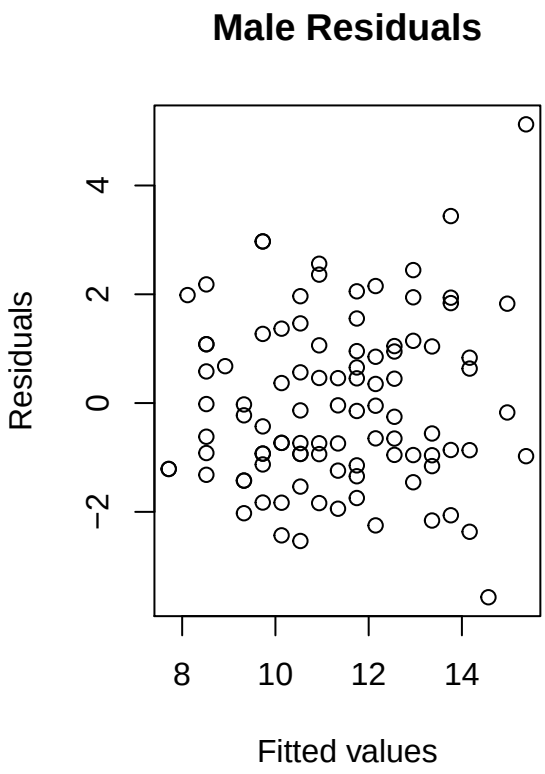
```
library(MASS)
data(cats)

catlm1=lm(Hwt~Bwt,data=cats)
```

```r
# Residuals
par(mfrow=c(1,2))
hist(catlm1$resid[cats$Sex=="F"],xlab="Residuals",main="Female Residuals")
hist(catlm1$resid[cats$Sex=="M"],xlab="Residuals",main="Male Residuals")
```




```r
plot(catlm1$fitted[cats$Sex=="F"],catlm1$resid[cats$Sex=="F"],
xlab="Fitted values",ylab="Residuals",main="Female Residuals")
plot(catlm1$fitted[cats$Sex=="M"],catlm1$resid[cats$Sex=="M"],
xlab="Fitted values",ylab="Residuals",
main="Male Residuals")
```





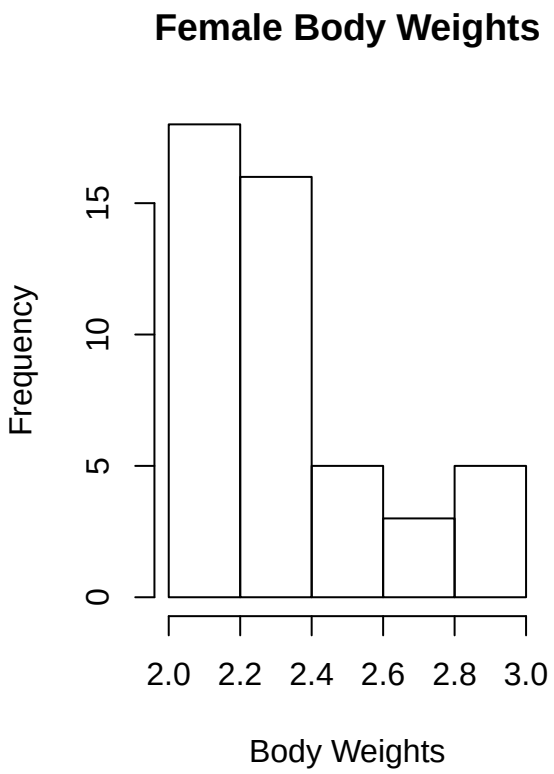

```
sd(catlm1$resid[cats$Sex=="F"])
```

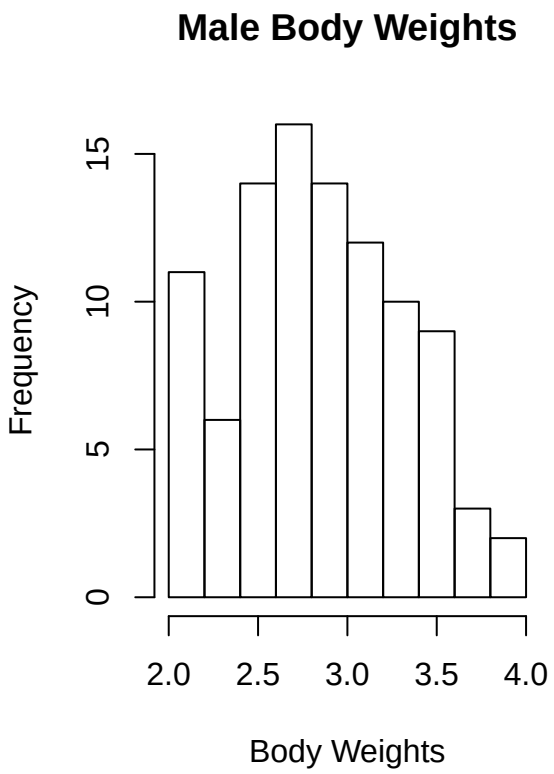
```
## [1] 1.211658
```

```
sd(catlm1$resid[cats$Sex=="M"])
```

```
## [1] 1.554186
```

```
# Body weights
hist(cats$Bwt[cats$Sex=="F"],xlab="Body Weights",main="Female Body Weights")
hist(cats$Bwt[cats$Sex=="M"],xlab="Body Weights",main="Male Body Weights")
```






```
par(mfrow=c(1,1))

summary(cats$Bwt[cats$Sex=="F"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    2.15    2.30    2.36    2.50    3.00
```

```
summary(cats$Bwt[cats$Sex=="M"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     2.0     2.5     2.9     2.9     3.2     3.9
```

The histograms show a distribution of residuals for male cats slightly more spread out than for female cats. The standard deviations reflect this also. The residuals for female cats seem to decrease slightly as the fitted values increase. This suggests that the wrong slope is fit for Female cats. The residual plot for male cats looks fairly uniform. The male cats mostly have more body weight than the female cats. The sample has more male cats than female cats.

Part b

Treat `Sex` as an omitted variable that could be a potential confounder in studying the relation between `Hwt` and `Bwt`. Go back to the in-class discussion of causal regression (you may find previous HWs helpful as well). Assume that `Sex` is the only confounder, and estimate the causal regression function $\theta(x)$ of Y on X, where the argument x stands for different values of body weight.

Draw a scatter plot of `Hwt` (on vertical) versus `Bwt` with the following four lines added so that we can tell them apart:

(i) the simple linear regression of `Hwt` on `Bwt`,

(ii) the causal regression line,

(iii) the simple regression of `Hwt` on `Bwt` for Females only, and

(iv) the simple regression of `Hwt` on `Bwt` for Males only.

Comment on when, if ever, each line would be most appropriate for making predictions about heart weight from body weight.

Solution

We are now going to apply the methodology of adjusting for a confounder, which will be `Sex`, which we label Z to match the notation from lectures. Along those lines, let X denote `Bwt`, and let Y denote the response `Hwt`. First, we estimate the conditional regression function of Y on X given Z for each value of Z, which is defined as

$$r(x|z) = \mathbb{E}(Y|X = x, Z = z).$$

There are only two values of Z (M and F), so this is not very complicated. There are multiple ways to do this in R. One could either fit separate linear models for the two levels of Z or fit a "crossed design" model.

```
# Crossed design
catlm2=lm(Hwt~Bwt*Sex,data=cats)
summary(catlm2)
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt * Sex, data = cats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7728 -1.0118 -0.1196  0.9272  4.8646
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.9813     1.8428   1.618 0.107960
## Bwt           2.6364     0.7759   3.398 0.000885 ***
## SexM         -4.1654     2.0618  -2.020 0.045258 *
## Bwt:SexM      1.6763     0.8373   2.002 0.047225 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.442 on 140 degrees of freedom
## Multiple R-squared:  0.6566, Adjusted R-squared:  0.6493
## F-statistic: 89.24 on 3 and 140 DF,  p-value: < 2.2e-16
```

```
# Separate regressions
modelf=lm(Hwt~Bwt,data=cats[cats$Sex=="F",])
modelm=lm(Hwt~Bwt,data=cats[cats$Sex=="M",])
summary(modelf)
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt, data = cats[cats$Sex == "F", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.00871 -0.68599 -0.04506  0.79583  2.21858
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.9813     1.4855   2.007 0.050785 .
## Bwt           2.6364     0.6254   4.215 0.000119 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.162 on 45 degrees of freedom
## Multiple R-squared:  0.2831, Adjusted R-squared:  0.2671
## F-statistic: 17.77 on 1 and 45 DF,  p-value: 0.0001186
```

```
summary(modelm)
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt, data = cats[cats$Sex == "M", ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7728 -1.0478 -0.2976  0.9835  4.8646
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.1841     0.9983  -1.186    0.239
## Bwt           4.3127     0.3399  12.688   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.557 on 95 degrees of freedom
## Multiple R-squared:  0.6289, Adjusted R-squared:  0.625
## F-statistic:   161 on 1 and 95 DF,  p-value: < 2.2e-16
```

Although the two fits are "equivalent" in the sense that the fitted values are the same for both, the parameterizations are different. The separate regressions is more convenient for doing the next step in finding the causal regression function. The needed calculation is

$$\hat{\theta}(x) = \hat{r}(x|F)\widehat{Pr(F)} + \hat{r}(x|M)\widehat{\Pr(M)},$$

where

- $\widehat{\Pr(F)}$ is the fraction of females in the sample,
- $\widehat{\Pr(M)}$ is the fraction of males in the sample,
- $\hat{r}(x|F)$ is the fitted regression for females, and
- $\hat{r}(x|M)$ is the fitted regression for males.

```
# The fractions
Prob=c(mean(cats$Sex=="F"),mean(cats$Sex=="M"))
Prob
```

```
## [1] 0.3263889 0.6736111
```

```
# Estimate theta (via coefficients)
thetahat=Prob[1]*modelf$coef+Prob[2]*modelm$coef
thetahat
```

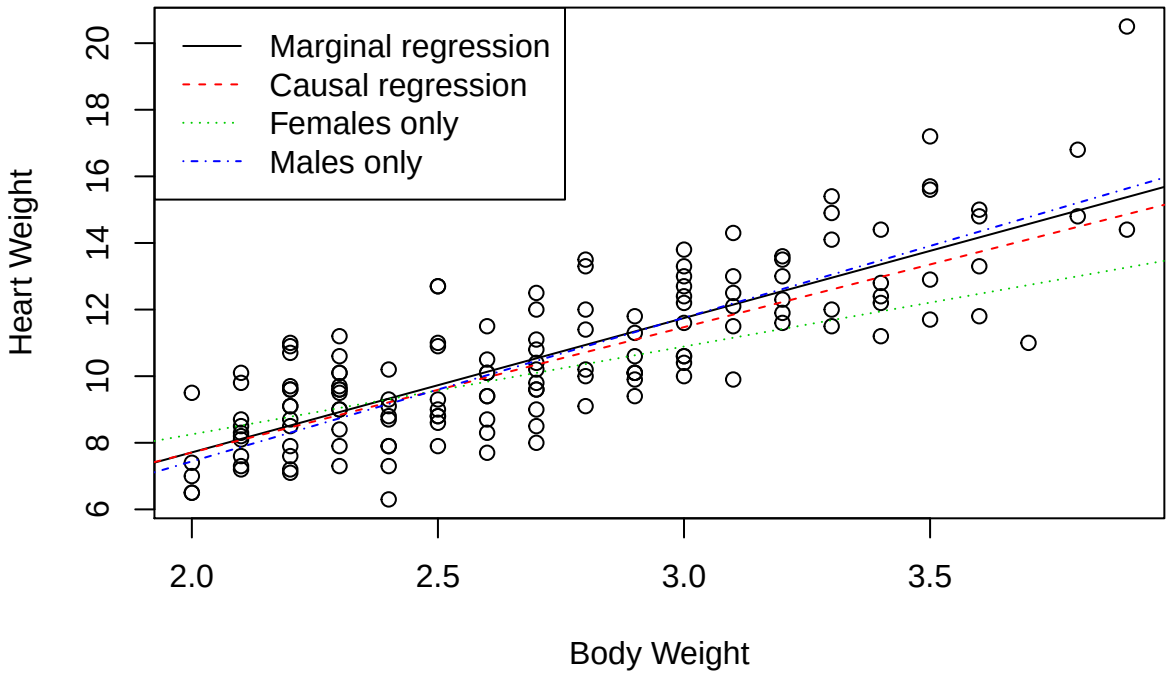
```
## (Intercept)         Bwt
##   0.1754524   3.7655646
```

The estimated causal regression line is

$$\hat{\theta}(x) = 0.1755 + 3.766x.$$

We summarize this analysis with the requested plot.

```
plot(cats$Bwt,cats$Hwt,xlab="Body Weight",ylab="Heart Weight")
abline(catlm1$coef,lty=1,col=1)
abline(thetahat,lty=2,col=2)
abline(modelf$coef,lty=3,col=3)
abline(modelm$coef,lty=4,col=4)
legend("topleft",
legend=c("Marginal regression","Causal regression","Females only","Males only"),
lty=c(1,2,3,4),col=c(1,2,3,4))
```



The marginal line would be useful if we wanted to predict heart weight for a cat drawn at random from a population like the one from which the sample was drawn.

The causal line would be appropriate for predicting heart weight for a cat drawn from a population that was about 33% females.

The female-only line would be appropriate for predicting heart weight for a a female cat. The male-only line would be appropriate for predicting heart weight for a a male cat.

Part c

Assume a model of the form

$$Y = \beta_{0,S} + \beta_{1,S}X + \varepsilon,$$

where S stands for `Sex`, and ε has a distribution that might depend on `Sex`, but is otherwise unspecified. You may have already fit this model earlier in this problem if you ran separate regressions for males and females. If not, do it now. For each sex separately, run a bootstrap by resampling residuals so that we can examine the different distributions of $(\hat{\beta}_{0,\text{Male}}, \hat{\beta}_{1,\text{Male}})$ and $(\hat{\beta}_{0,\text{Female}}, \hat{\beta}_{1,\text{Female}})$. To do this, take a look at the code below (we will talk more about when to use which type of bootstrap in lecture next week):

```
## You will need to store the results and complete the code below
## datm and datf are the datasets partitioned by sex
## modelm and modelf are the linear regression models by sex
## betahat_null is the betahat vector of coefficients under the null hypothesis
resm <- resid(modelm)
resf <- resid(modelf)
B <- 10000
for (bb in 1:B) {
  newym <- cbind(1, cats$Bwt[cats$Sex=="M"])%*%betahat_null +
    sample(resm, length(resm), replace=TRUE)
  newfitm <- lm(Hwt~Bwt, data=data.frame(Hwt=newym,
                                         Bwt=cats$Bwt[cats$Sex == "M"]))
  boot_coeffs <- coef(newfitm)
  ## Similar for modelf
}
```

In particular, we will test the single null hypothesis

$$H_0: \text{ Both } \beta_{0,\text{Male}} = \beta_{0,\text{Female}} \text{ and } \beta_{1,\text{Male}} = \beta_{1,\text{Female}}.$$

The alternative hypothesis is that H_0 is false, i.e., either the slopes differ or the intercepts differ or both.

Use the following form of the test:

- Define the test statistic

$$T = (\hat{\beta}_{0,\text{Male}} - \hat{\beta}_{0,\text{Female}})^2 + (\hat{\beta}_{1,\text{Male}} - \hat{\beta}_{1,\text{Female}})^2.$$

- Reject H_0 at level α if $T > q_\alpha$, where q_α is the $1-\alpha$ quantile of the null distribution of T, that is, the distribution of T when $\beta_{0,\text{Male}} = \beta_{0,\text{Female}}$ and $\beta_{1,\text{Male}} = \beta_{1,\text{Female}}$.

To bootstrap from the null distribution, replace the estimated male and female regression parameters by a common set of parameters, e.g., the marginal regression line from part (a) or the causal regression line from part (b). It won't matter which one you use because of the form of the statistic T. Be sure to replace the ε's by sampled values from the appropriate residuals. Perform the test by bootstrapping the p-value. That is, find $\Pr(\hat{T}^* \geq T)$, where T is the test statistic computed from the observed data and $\hat{T}^*$ is drawn computed from a generic bootstrap sample from the null distribution.

Solution

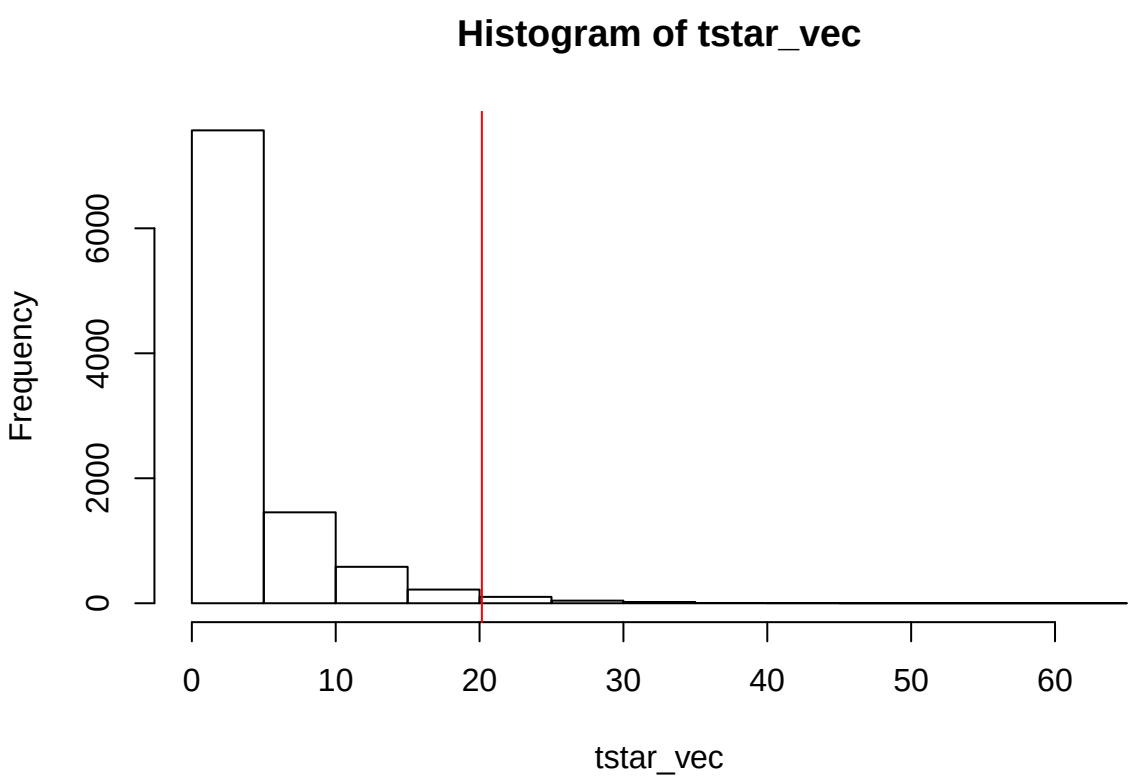
We end with a test of whether the male and female regression lines are the same. The distribution from which we need bootstrap is one in which the two regression lines are the same. We were told to set them both equal to the regression line found in part (a).

```
B <- 10000
resm <- resid(modelm)
resf <- resid(modelf)
bethat_null <- c(catlm1$coef[1], catlm1$coef[2])
```

```r
get_tstar <- function() {
  newym <- cbind(1, cats$Bwt[cats$Sex=="M"])%*%bethat_null +
    sample(resm, length(resm), replace=TRUE)
  newfitm <- lm(Hwt~Bwt, data=data.frame(Hwt=newym,
                                         Bwt=cats$Bwt[cats$Sex == "M"]))
  boot_coeffsm <- coef(newfitm)

  newyf <- cbind(1, cats$Bwt[cats$Sex=="F"])%*%bethat_null +
    sample(resf, length(resf), replace=TRUE)
  newfitf <- lm(Hwt~Bwt, data=data.frame(Hwt=newyf,
                                         Bwt=cats$Bwt[cats$Sex == "F"]))
  boot_coeffsf <- coef(newfitf)

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

}

tstar_vec <- replicate(B, get_tstar())
# The test statistic from the data
tobs <- (modelf$coef[1]-modelm$coef[1])^2+(modelf$coef[2]-modelm$coef[2])^2
names(tobs) <- NULL
print(tobs)
```

```
## [1] 20.16042
```

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




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

```
## [1] 0.0171
```

The p-value is the probability in the upper tail of the distribution of T^* (supposedly the bootstrap distribution of T_b^*) at or above the observed statistic $t_{\text{obs}} = 20.16$.
From output above, we see that the p-value is 0.0171. This is small enough to reject H_0 at all levels greater than 0.0171. At level $\alpha = 0.01$, we would not reject.