Sample solutions

Stat 8051 Homework 4

Problem 1: ALR Exercise 7.1

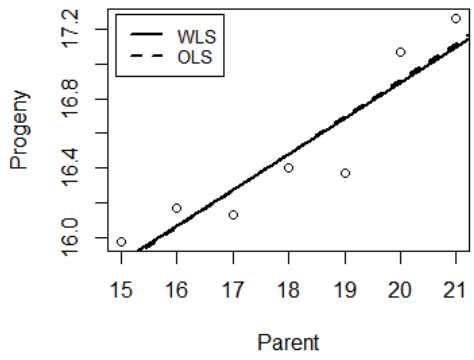
The basic structure for both the models is same: $Y = \mathbf{X}\boldsymbol{\beta}+\epsilon$. For this reason all coefficient estimates, standard errors and F-tests will be same in both the models.

The only difference between Sue and Joe's models is that they have $\epsilon \sim N(0, 2\sigma^2)$ and $\epsilon \sim N(0, \sigma^2)$. Thus the estimates of $\hat{\sigma}$ will be different.

Problem 2: ALR Exercise 7.7

7.7.1 and 7.7.2

```
> plot(Progeny ~ Parent, galtonpeas)
> m.weighted <- lm(Progeny ~ Parent,
+                  data=galtonpeas, weights= 1/SD^2)
> abline(m.weighted, lwd=2)
>
> abline(m.unweighted <- lm(Progeny ~ Parent,
+                           data=galtonpeas), lty=2, lwd=2)
> legend("topleft", c("WLS", "OLS"), lty=1:2 , lwd=2,
+        cex=.8, inset=.02)
```

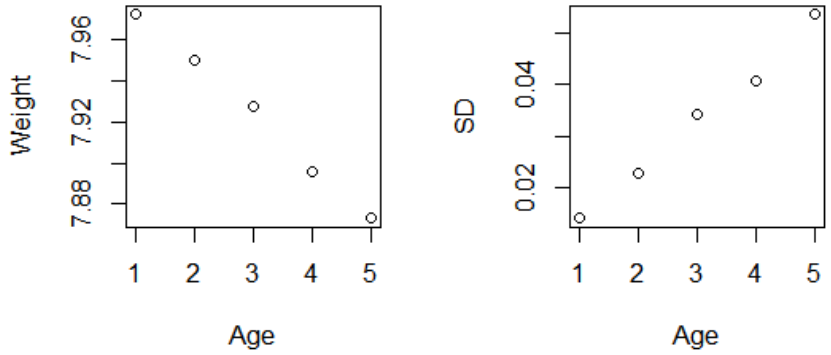


The weighted and unweighted fits look almost the same.

7.7.3 This should decrease the slope, and it could increase variances, making differences more difficult to detect.

Problem 3: ALR Exercise 7.8

7.8.1 There is a clear decreasing linear trend of weight vs. age, but the SD's are increasing with increasing age as well.



7.8.2

```
> mod2 = lm(Weight~Age, data=jevons, weights=SD^2/n)
> (z = summary(mod2))

Call:
lm(formula = Weight ~ Age, data = jevons, weights = SD^2/n)

Weighted Residuals:
         1          2          3          4          5
-5.211e-06 -5.230e-07  1.935e-05 -2.072e-05  8.662e-06

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.002701   0.005591  1431.3 7.52e-10 ***
Age         -0.026099   0.001292   -20.2 0.000265 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.738e-05 on 3 degrees of freedom
Multiple R-squared:  0.9927,Adjusted R-squared:  0.9903
F-statistic: 408.2 on 1 and 3 DF,  p-value: 0.000265
```

There is a significant effect of age in on coin weight. High value of multiple R^2 signifies a good linear fit.

7.8.3 We need to calculate the t-statistic manually here. The p-value is borderline. On face we fail to reject the null and conclude that the fitted regression coefficient is not different than the known standard weight, but it probably would not be a good idea to rely on it too much.

```
> tstat = (z$coef[1,1]-7.9876)/z$coef[1,2]
> 2*(1-pt(abs(tstat), 3))
[1] 0.07373166
```

7.8.4 Going by the hint, the first component mentioned is given by taking square of the variable SD, while the variance of the fitted value also depends on the number of samples in each age category:

$$\text{sefit}(\tilde{y}|x_j) = \frac{SD_j^2}{n_j^2}$$

Hence the overall prediction error $\text{sepred}(\tilde{y}|x_j) = SD_j^2 + SD_j^2/n_j^2$. We calculate this in R, and the required probabilities are given below:

```
> c0 = mod2$coef[1]
> c1 = mod2$coef[2]
> mean.vec = c0+(1:5)*c1
> se.vec = with(jevons, sqrt(SD^2 + SD^2/n^2))
> pnorm(7.9379, mean=mean.vec, sd=se.vec)
[1] 0.003010105 0.289557899 0.653105748 0.835036479 0.889928938
```

7.8.5 To obtain the standard error, we need to apply Delta method on the function (7.9379 − Intercept)/Coefficient for Age. This can be done in two ways:

Manually

```
> (age.at.min = (7.9379-c0)/c1)
(Intercept)
   2.482909
> grad = c(-1/c1, -(7.9379-c0)/c1^2)
> (se.age.at.min = sqrt(t(grad)%*%vcov(mod2)%*%grad))
           [,1]
[1,] 0.09657335
```

Using inbuilt function

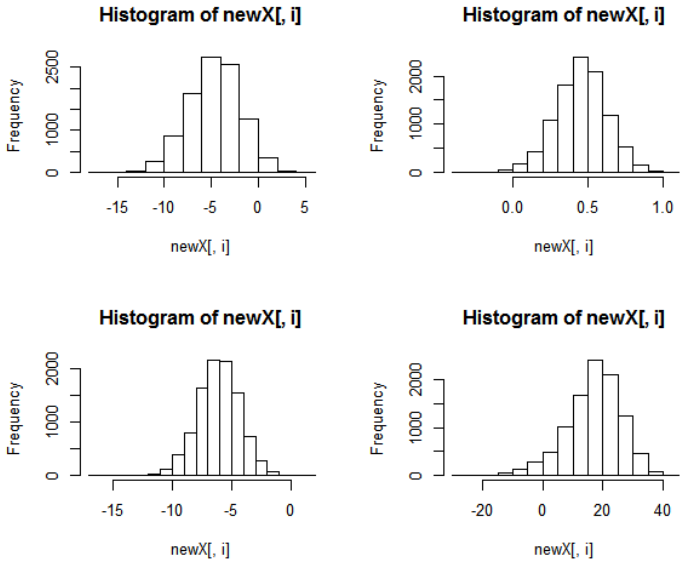
```
> deltaMethod(mod2, "(7.9379-Intercept)/Age")
                          Estimate         SE
(7.9379 - Intercept)/Age 2.482909 0.09657335
```

Problem 4: ALR Exercise 7.10

We use the function below to obtain bootstrap coefficients for `nsamp` number of samples.

```
bootcoefs = function(nsamp){
  n = nrow(fuel2001)
  coef.mat = matrix(0, nrow=nsamp, ncol=4)
  for(i in 1:nsamp){
    isamp = sample(1:n, n, replace=T)
    imod = update(m0, data=fuel2001[isamp,])
    coef.mat[i,] = coef(imod)[-1]
  }
  return(coef.mat)
}
```

The histograms look as follows: All other coefficients except from that for $\log(Miles)$



adhere to normality. Now we shall look at the confidence intervals. The OLS CI's are as below (using `confint`):

```
                2.5 %     97.5 %
Tax        -8.3144050 -0.1415614
Dlic        0.2131871  0.7305553
Income    -10.5508863 -1.7197756
logMiles    5.5174630 31.5730860
```

From the bootstrap coefficients we can get CI's in two ways: byt normal approximation or the actual CI. Here they are:

```
> # bootstrap approx
> mean.vec = apply(beta.matrix, 2, mean)
> sd.vec = apply(beta.matrix, 2, sd)
> cbind(mean.vec-1.96*sd.vec, mean.vec+1.96*sd.vec)
            [,1]       [,2]
[1,] -10.0617580  0.7387013
[2,]   0.1203924  0.7906267
[3,]  -9.6214058 -2.5908535
[4,]  -1.1835968 34.9410385
>
> # bootstrap actual
> qfun = function(x) quantile(x, c(.025,.975))
> t(apply(beta.matrix, 2, qfun))
           2.5%      97.5%
[1,] -10.312034  0.4767877
[2,]   0.103338  0.7838203
[3,]  -9.717586 -2.6421438
[4,]  -4.425712 32.6688512
```

It is better to use the actual CI's because we do see deviation from normality here. The actual and OLS CI's do differ from each other, and the lower limit is very different for the last coefficient.