STOR 664 HW 2

Brian N. White

9/9/2020

Problem 20

```
library(data.table)
dmark <- fread('http://rls.sites.oasis.unc.edu/faculty/rs/source/Data/dmark.dat')
#pull in data from website
head(dmark)
```

```
##    V1    V2    V3
## 1:  1 1.153 1.529
## 2:  2 1.152 1.530
## 3:  3 1.161 1.490
## 4:  4 1.161 1.483
## 5:  5 1.159 1.457
## 6:  6 1.154 1.460
```

```
colnames(dmark)[] <- c("week", "mark", "pound")
#give variables descriptive names
colnames(dmark)
```

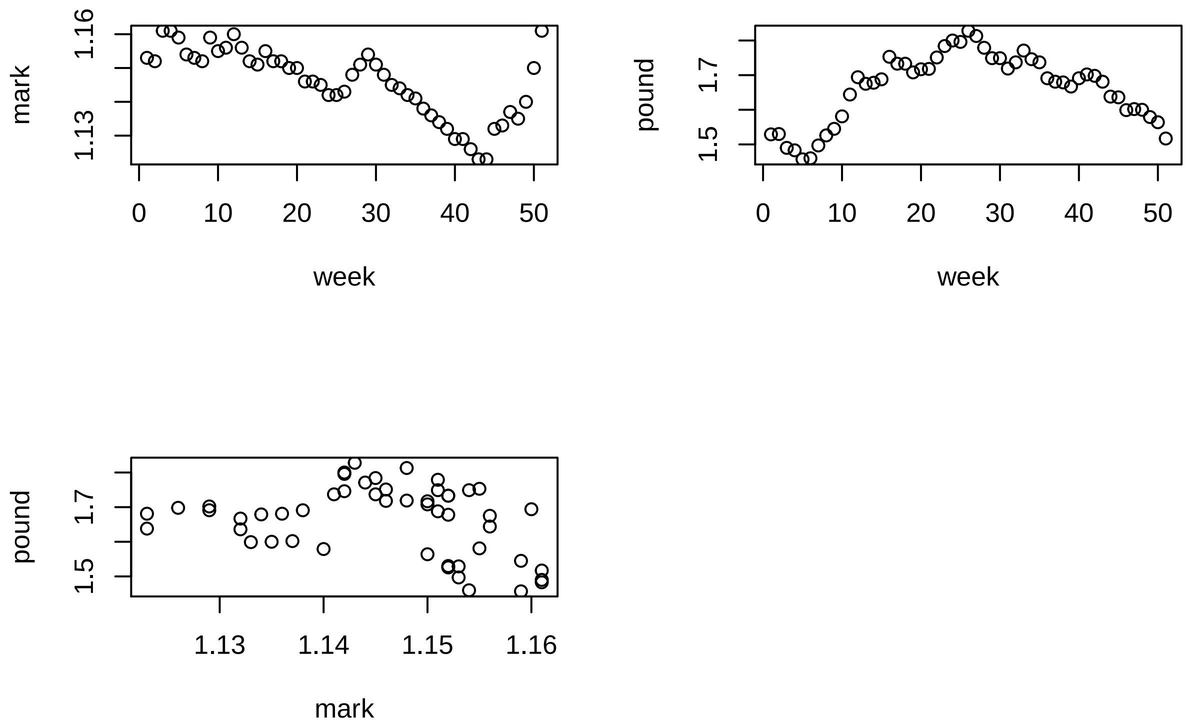
```
## [1] "week"  "mark"  "pound"
```

```
attach(dmark)
#column names of mydat recognized independently
```

(a)

Note, there is strong visual evidence of autocorrelation in the time series.

```
par(mfrow=c(2, 2))
plot(mark~week)
plot(pound~week)
plot(pound~mark)
```



(b)

Assume the individaul weekly observaions are independent. The linear regression equation $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ is computed via the code below. A point estimate for $\hat{\beta}_1$ is approximately -2.9 with a 90% confidence interval of $[-5.12, -0.69]$. Consider the following hypotheses: $H_0 : \beta_1 = 0$ vs $H_1 : \beta_1 \neq 0$ with $\alpha = 0.1$. Observe that 0 is not an element of $[-5.12, -0.69]$, the 90% confidence interval for β_1. Thus, H_0 is rejected; there is evidence to suggest that $\beta_1 \neq 0$.

```
mp_lm <- lm(pound~mark)
summary(mp_lm)
```

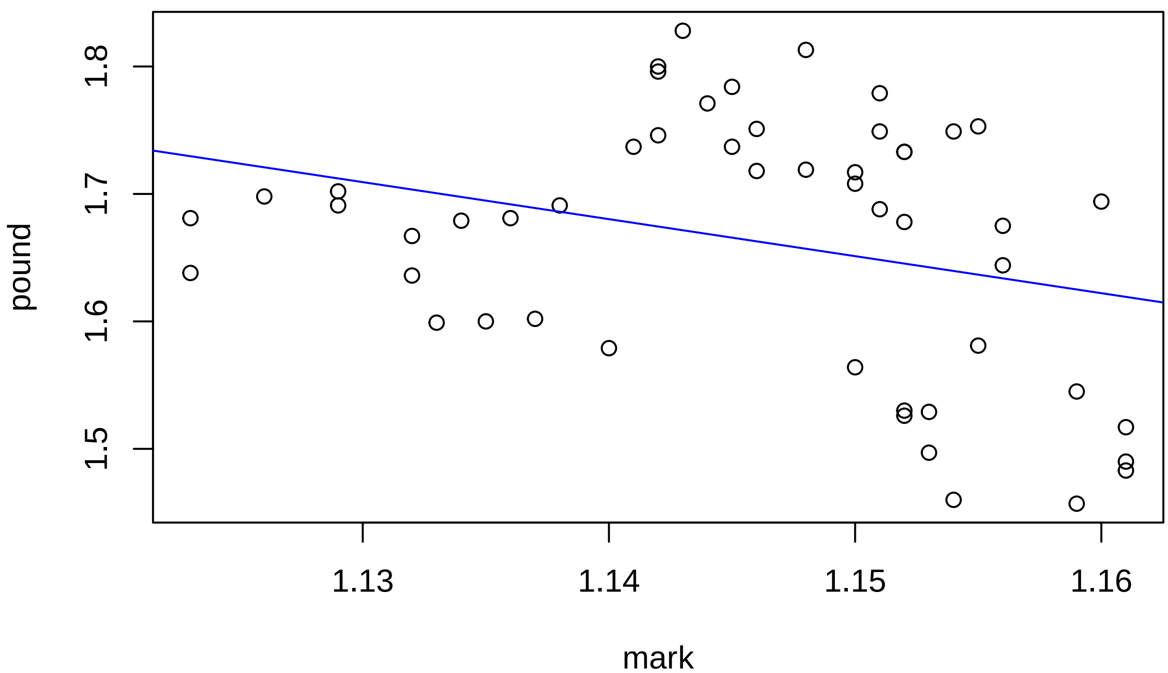
```
##
## Call:
## lm(formula = pound ~ mark)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.179567 -0.089384  0.004972  0.079741  0.156491
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.991      1.516   3.292  0.00185 **
## mark          -2.904      1.323  -2.195  0.03293 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09665 on 49 degrees of freedom
```

```
## Multiple R-squared:  0.08952,    Adjusted R-squared:  0.07094
## F-statistic: 4.818 on 1 and 49 DF,  p-value: 0.03293
```

```
confint(mp_lm, level=0.90)
```

```
##                     5 %       95 %
## (Intercept)  2.449006  7.5321114
## mark        -5.121743 -0.6858684
```

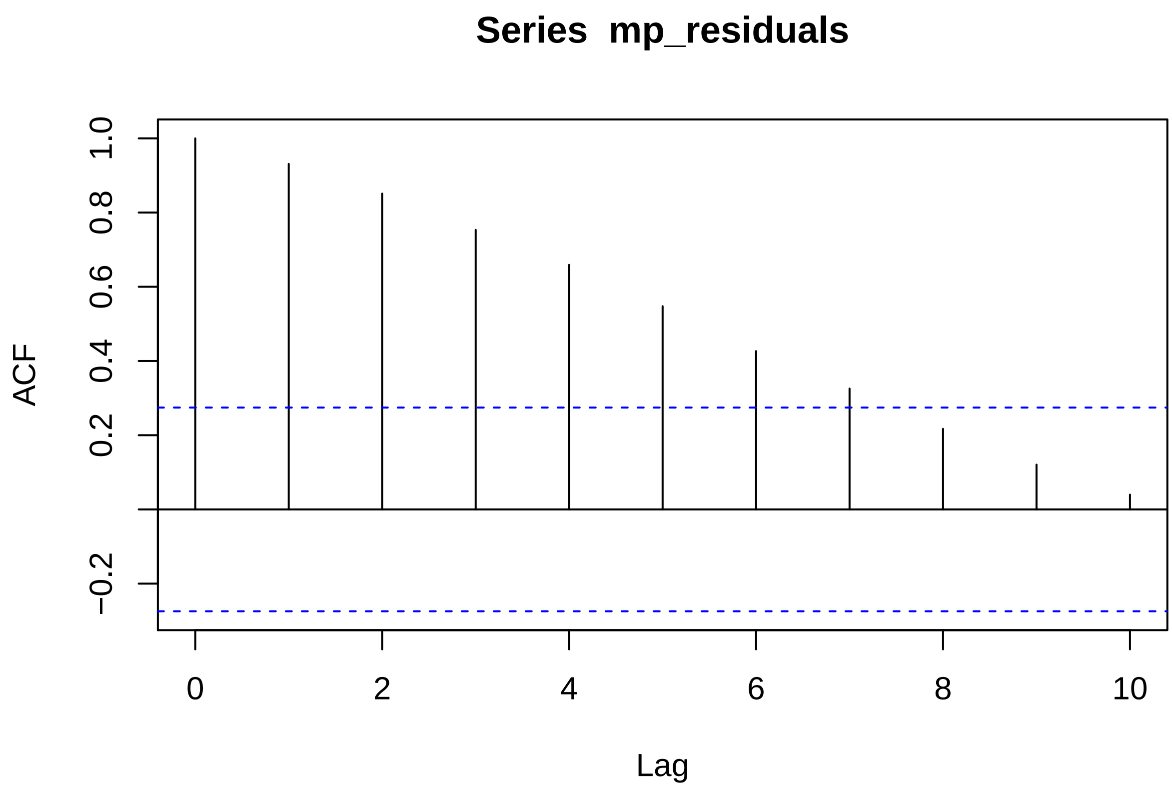
```
plot(pound~mark)
abline(mp_lm, col="blue")
```



(c)

Inspection of the first 10 autocorrelation coeffecients are computed from the residuals via the acf(.) command. Inspection of these, together with the heuristic $\frac{2}{\sqrt{n}} = \frac{2}{\sqrt{51}} \approx 0.28$, suggests that the first 8 autocorrelations are statistically significant. There is clearly evidence that autocorrelation is present. A more precise test, such as the Durbin-Watson could be performed to confirm this heuristic argument.

```
mp_residuals <- mp_lm$residuals #the residuals of the linear model in question
mp_ac <- acf(mp_residuals, lag.max=10) #the first 10 serial correlations are computed
```




```
#with approximate 95% error bounds if the true time series is independent.
mp_ac
```

```
##
## Autocorrelations of series 'mp_residuals', by lag
##
##     0     1     2     3     4     5     6     7     8     9    10
## 1.000 0.931 0.851 0.753 0.659 0.548 0.427 0.326 0.217 0.121 0.040
```

```
#the heursitic used to determine statistical significance of the autocorrelations
heuristic<-2/sqrt(51)
#the indices of the autocorrelations that are statistically significant.
which(abs(mp_ac$acf)>heuristic)
```

```
## [1] 1 2 3 4 5 6 7 8
```

```
#Code to compute the Durbin Watson test statistic
dw_num <- rep(0,50)
for(i in 2:51){
  dw_num[i] <- (mp_residuals[i]-mp_residuals[i-1])^2
}

D=sum(dw_num)/sum(mp_residuals^2)
D
```

```
## [1] 0.08659338
```

(d) As autocorrelation is present, the standard deviation of the least squared estimates must be corrected. In particular, the corrected standard deviation of $\hat{\beta}_1$ is computed below for K=8. Note, this corrected value is about 3.77, in contrast to the original value of 1.32. With this new value, observe that the test statistic for the previously considerd hypotheses is $t = -\frac{2.195}{3.77} \approx -0.771$. The p-value associated with this test statistic, with respect to a t_{n-2} distribution, is greater than the 0.1 threshold. Thus, in contrast to the previous conclusion, the evidence does not support the rejection of H_0. In otherwords, there is not statistically significant evidence to suggest that $\beta_1 \neq 0$.

```r
summary(mp_lm)
```

```
##
## Call:
## lm(formula = pound ~ mark)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.179567 -0.089384  0.004972  0.079741  0.156491
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.991      1.516   3.292  0.00185 **
## mark          -2.904      1.323  -2.195  0.03293 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09665 on 49 degrees of freedom
## Multiple R-squared:  0.08952,    Adjusted R-squared:  0.07094
## F-statistic: 4.818 on 1 and 49 DF,  p-value: 0.03293
```

```r
se_slope=1.323

#confirm uncorrected standard error of slope estimate
rse <- sum((mp_lm$residuals)^2)/49
beta1_se=sqrt(rse/sum((mark-mean(mark))^2))
beta1_se
```

```
## [1] 1.322917
```

```r
#compute corrected standard deviation of slope estimate

#the denominator of r_x(k) for all k=1,..,8
k_denom <- vector()
  for(i in 1:51){
    k_denom[i] <- (mark[i]-mean(mark))^2
  }

#compute the r_x(k) values for k=1,...,8. e.g. k1 is a vector
#containing the terms of the sum in the numerator of the ratio that defines r_x(k)
 k1 <-vector()
 for(i in 1:50){
```

```
    k1[i] <- (mark[i]-mean(mark))*(mark[i+1]-mean(mark))
 }

k2 <-vector()
 for(i in 1:49){
    k2[i] <- (mark[i]-mean(mark))*(mark[i+2]-mean(mark))
 }

k3 <-vector()
 for(i in 1:48){
    k3[i] <- (mark[i]-mean(mark))*(mark[i+3]-mean(mark))
 }

k4 <-vector()
 for(i in 1:47){
    k4[i] <- (mark[i]-mean(mark))*(mark[i+4]-mean(mark))
 }

k5 <-vector()
 for(i in 1:46){
    k5[i] <- (mark[i]-mean(mark))*(mark[i+5]-mean(mark))
 }

k6 <-vector()
 for(i in 1:45){
    k6[i] <- (mark[i]-mean(mark))*(mark[i+6]-mean(mark))
 }

k7 <-vector()
 for(i in 1:44){
    k7[i] <- (mark[i]-mean(mark))*(mark[i+7]-mean(mark))
 }

k8 <-vector()
 for(i in 1:43){
    k8[i] <- (mark[i]-mean(mark))*(mark[i+8]-mean(mark))
 }

#sum over the terms to compute r_x(k) for k=1,..,8
rx1 <-sum(k1)/sum(k_denom)
rx2 <-sum(k2)/sum(k_denom)
rx3 <-sum(k3)/sum(k_denom)
rx4 <-sum(k4)/sum(k_denom)
rx5 <-sum(k5)/sum(k_denom)
rx6 <-sum(k6)/sum(k_denom)
rx7 <-sum(k7)/sum(k_denom)
rx8 <-sum(k8)/sum(k_denom)

#create vector where kth entry is kth sample autocorrelation of the {mark_i} process.
a <- c(rx1, rx2, rx3, rx4, rx5, rx6, rx7, rx8)

#create vector of terms in sum in the second term of
#the scaling factor for the corrected variance of slope estimate
```

```r
b <- vector()
for(i in 1:8){
  b[i] <- (mp_ac$acf[i])*(a[i])
}

#the slope estimate standard deviation corrected for
#the presence of autocorrelation in the residuals.
beta1_se_corrected <- sqrt(((beta1_se)^2)*(1+2*sum(b)))

beta1_se_corrected
```

```
## [1] 3.765597
```

```r
beta1_se
```

```
## [1] 1.322917
```

```r
t=(mp_lm$coefficients[2])/beta1_se_corrected
t
```

```
##       mark
## -0.7711408
```

```r
qt(.05, 49)
```

```
## [1] -1.676551
```

```r
2*pt(t, 49)
```

```
##      mark
## 0.4443263
```

Problem 21

```r
marathon <- fread('http://rls.sites.oasis.unc.edu/faculty/rs/source/Data/marathon.dat')
#pull in data from website
head(marathon)
```

```
##          V1      V2
## 1: 1.000178  9402.5
## 2: 1.000178  9404.0
## 3: 1.000178  9402.0
## 4: 1.000178  9403.0
## 5: 0.000000 12163.5
## 6: 0.000000 14981.5
```

```r
colnames(marathon)[] <- c("length", "count")
#give variables descriptive names
colnames(marathon)
```

```
## [1] "length" "count"
```

```r
attach(marathon)
#column names of mydat recognized independently
```