Exam 1 Answer Key

Econ 217

2/12/2019

Problem 1

Part A

First, we restrict the data as described in the question:

```
options(warn=-1)
library(foreign)
d<-read.dta("https://people.ucsc.edu/~aspearot/Econ_217/org_example.dta")

d$hourslw<-ifelse(is.na(d$hourslw),0,d$hourslw)

subd13<-subset(d,year==2013&nilf==0)
subd93<-subset(d,year==1993&nilf==0)
```

Run the real wage models using glm

```
realwage13<-glm(log(rw)~as.factor(age)*female+educ+wbho+female,subd13,family="gaussian")
realwage93<-glm(log(rw)~as.factor(age)*female+educ+wbho+female,subd93,family="gaussian")
```

We will simply copy output here to avoid the long print out - the education coefficients for 2013 are:

```
#educHS                 0.232269   0.007988  29.077  < 2e-16 ***`
#educSome college       0.350032   0.008094  43.247  < 2e-16 ***`
#educCollege            0.722504   0.008374  86.276  < 2e-16 ***`
#educAdvanced           0.940278   0.009000 104.476  < 2e-16 ***`
```

the education coefficients for 1993 are:

```
#educHS                 0.2000987  0.0057882  34.570  < 2e-16 ***`
#educSome college       0.3312980  0.0060501  54.759  < 2e-16 ***`
#educCollege            0.6095148  0.0065777  92.663  < 2e-16 ***`
#educAdvanced           0.7906654  0.0076687 103.103  < 2e-16 ***`
```

The college and advanced degree premia are larger in 2013 than 1993. Specifically, wages are 11% higher in 2013 than 1993 when comparing somebody with a college degree to somebody with less than a high school degree. For an advanced degree, the wage gap increased 15% over this period. Further, the gap between an advanced degree and college degree is 22% in 2013, higher than 18% in 1993. So, it appears that the wage benefits of college and advanced education are rising.

Grading Criteria Correct coefficient/table [6]
Correct interpretation (log, percentage change) [4]
Note: if you get different result (say, due to improper data cleaning), part or all of the coefficient score will be removed but no further punishment afterwards.

Part B

To generate the predictions, create new data frames, one for men, one for women, with the ages of interest. Specifically, the following code creates a data frame of ages 20 through 70, of while college educated men and women, respectively.

```
newmale<-data.frame(age=seq(20,70,by=1),female=0,wbho="White",educ="College")
newfemale<-data.frame(age=seq(20,70,by=1),female=1,wbho="White",educ="College")
```

Then, create a figure that will report all the results for this question:

```
plot(I(predict(realwage93,newfemale)-predict(realwage93,newmale))~newmale$age,type="l",col='blue',lwd=2,
     ylim=c(-0.50,0),main="Female-Male (log) wage gaps",xlab='age',ylab='log wage gap')
lines(I(predict(realwage13,newfemale)-predict(realwage13,newmale))~newmale$age,type="l",col='red',lwd=2)
```

The first plot shows the life cycle of the wage gap in 1993 and 2013. 2013 is in Red, and 1993 is in Blue. The first thing to note from these figures is that the age gap is largest somewhere around age 45-60. However, though the wage gap is still large during these years, it has fallen substantially since 1993 during the middle and late years of work. Specifically, the wage gap is just shy of 40% for in 1993 for 50-55 year old women, it's around 25% for the same age range in 2013. So, while the gap has fallen, females still make less than men, and at the largest percetnage, this is around 25% in 2013.

Grading Criteria Correct figure [12]
Comments on gaps over age based on your own result [4]
Comments on gaps over time, expect to see a relative difference (DID style) [4]
Note: ideally you may generate "faked" data, i.e., female/male, college, with age grid [20,70]. It's fine if you use in-sample prediction, but the shape of graph must be identical to get credit.

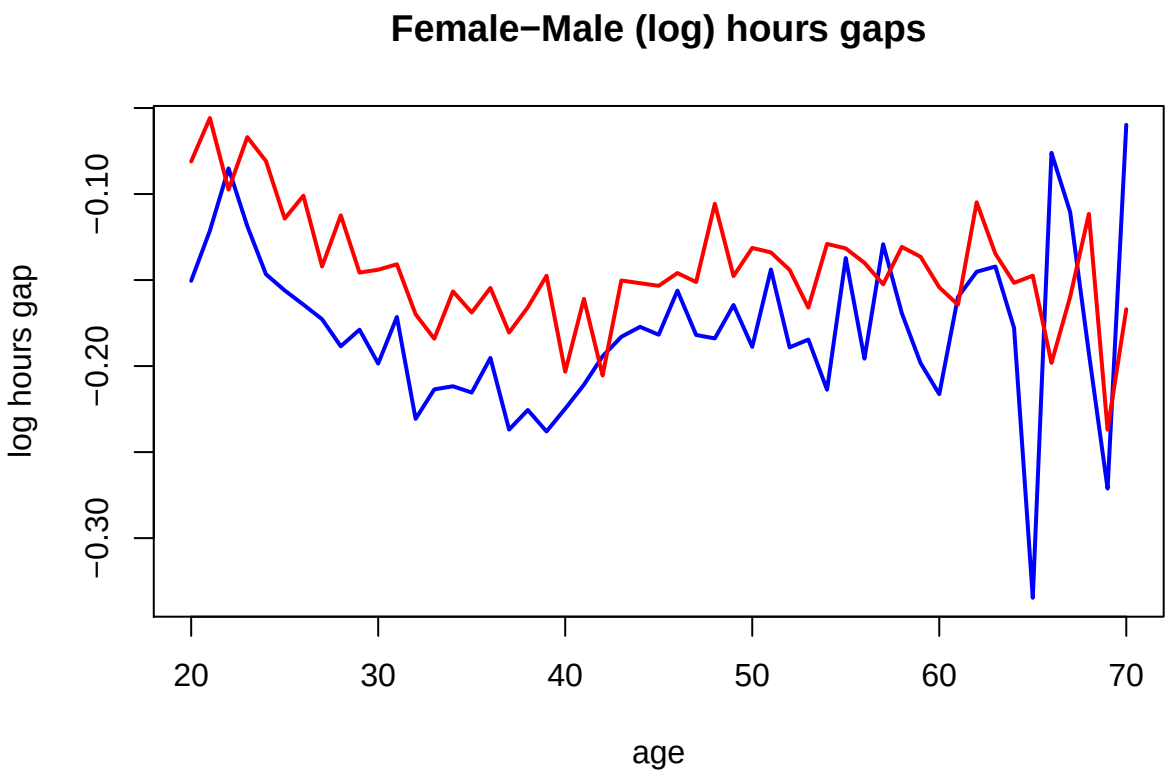
Part C

For this model, we will use a poisson GLM with a log link. This is ensure that predicted values of hour worked are positive, and zero values can be handled (which we imposed earlier). The two models are the following:

```
hours13<-glm(hourslw~as.factor(age)*female+educ+wbho+female,subd13,family=poisson(link='log'))
hours93<-glm(hourslw~as.factor(age)*female+educ+wbho+female,subd93,family=poisson(link='log'))
```

We use a similar approach to ploting the results.

```
plot(I(predict(hours93,newfemale)-predict(hours93,newmale))~newmale$age,col='blue',type="l",lwd=2,
     main="Female-Male (log) hours gaps",xlab='age',ylab='log hours gap')
lines(I(predict(hours13,newfemale)-predict(hours13,newmale))~newmale$age,col='red',type="l",lwd=2)
```



Note that since this is a poisson model, and we are NOT writing "type=response" in the prediction, we will be given predictions in log hours worked. The gap seems less than in part b, though again this gap is largest during middle-late ages of work, and has fallen between 1993 and 2013.

Grading Criteria Correct figure [12]
Comments on gaps over age based on your own result [4]
Comments on gaps over time, expect to see a relative difference (DID style) [4]

Problem 2

Part A

Load a sample dataset for the answer key

```
    d<-read.csv("https://people.ucsc.edu/~aspearot/Econ_217/ATNHPIUS11244Q.csv",header=TRUE)
```

For all datasets, we applied the following cleaning to make them the most recent 160 periods

```
d<-d[(nrow(d)-159):nrow(d),]
d$time<-seq(1,nrow(d),by=1)
names(d)<-c("quarter","price","time")
```

Now, we execute the cross validation to find the optimal span. This is nearly

```
    rm(results)
    for(h in seq(0.05,1,by=0.05)){
    for(i in 1:nrow(d)){
```

```
      smalldrop<-d[i,]
      smallkeep<-d[-i,]
      fit<-loess(price~time,smallkeep, family="gaussian",span=h, degree=1)
      dropfit<-predict(fit,smalldrop,se=FALSE)
      sqrerr<-(smalldrop$price-as.numeric(dropfit))^2
      if(exists('results')==TRUE){results<-rbind(results,data.frame(h,i,sqrerr))}
      if(exists('results')==FALSE){results<-data.frame(h,i,sqrerr)}

    }
    print(h)
    }
```

```
## [1] 0.05
## [1] 0.1
## [1] 0.15
## [1] 0.2
## [1] 0.25
## [1] 0.3
## [1] 0.35
## [1] 0.4
## [1] 0.45
## [1] 0.5
## [1] 0.55
## [1] 0.6
## [1] 0.65
## [1] 0.7
## [1] 0.75
## [1] 0.8
## [1] 0.85
## [1] 0.9
## [1] 0.95
## [1] 1
```

Report the different prediction errors for each span, and choose the span with the minimum predition error

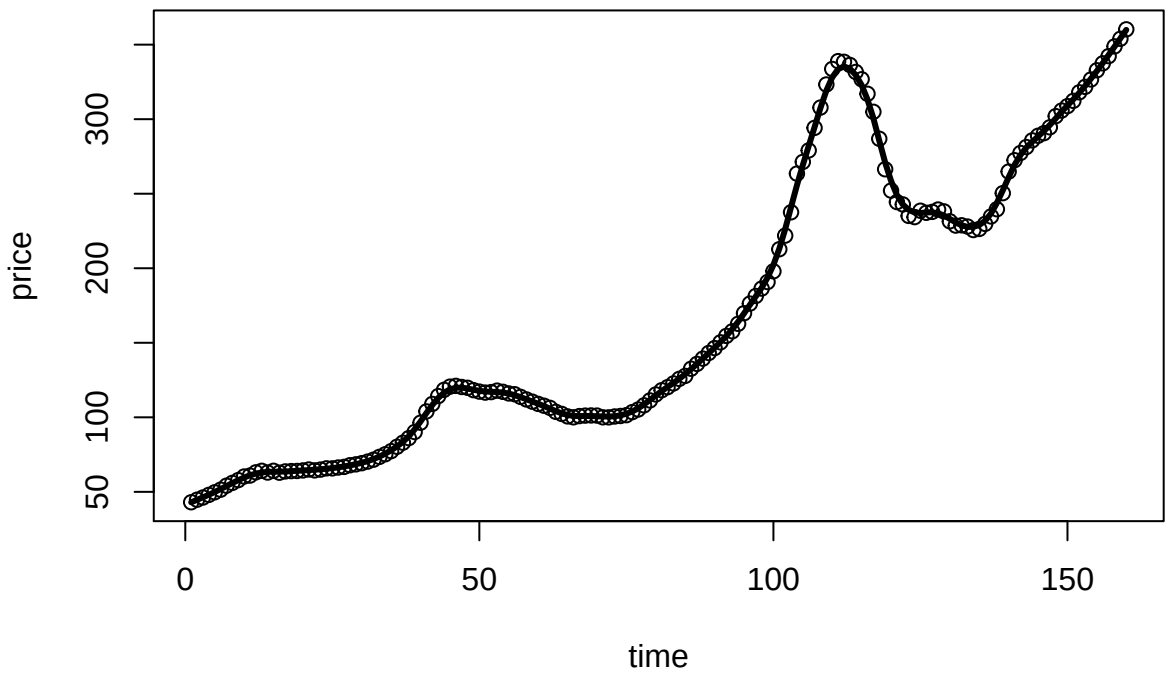
```
    tapply(results$sqrerr,results$h,FUN=sum,na.rm=TRUE)
```

```
##        0.05         0.1        0.15         0.2        0.25         0.3
##    764.5492   4170.3667  12371.0287  24683.2216  41095.3239  60838.2476
##        0.35         0.4        0.45         0.5        0.55         0.6
##  82824.2656 109645.4724 133112.0943 148984.7692 162098.3701 173293.2839
##        0.65         0.7        0.75         0.8        0.85         0.9
## 179883.4667 188932.4615 195190.8753 201192.5829 205291.8101 208048.2568
##        0.95           1
## 209912.9110 211241.6573
```

```
    h_ans<-as.numeric(names(which.min(tapply(results$sqrerr,results$h,FUN=sum,na.rm=TRUE))))
```

calculate the fit at the optimal span, and add to a plot:

```
    fit_opt<-loess(price~time,data=d, family="gaussian",span=h_ans, degree=1)

  plot(price~time,d,main=paste("Part A:  Span = ",h_ans))
  lines(d$time,predict(fit_opt),type="l",lwd=3,ylim=c(-2,2))
```



Grading Criteria Correct figure [5]
Correct code and/or (your tapply result with span) [5]

Part B

With knots starting at 10, there are 15 knots. There would also be one at 160, but since we cannot identify changes after 160, it is not needed here

```
knots<-c(10,20,30,40,50,60,70,80,90,100,110,120,130,140,150)
```

Define the spline around the knots

```
d$k1<-ifelse(d$time>(knots[1]),(d$time-knots[1])^3,0)
d$k2<-ifelse(d$time>(knots[2]),(d$time-knots[2])^3,0)
d$k3<-ifelse(d$time>(knots[3]),(d$time-knots[3])^3,0)
d$k4<-ifelse(d$time>(knots[4]),(d$time-knots[4])^3,0)
d$k5<-ifelse(d$time>(knots[5]),(d$time-knots[5])^3,0)
d$k6<-ifelse(d$time>(knots[6]),(d$time-knots[6])^3,0)
d$k7<-ifelse(d$time>(knots[7]),(d$time-knots[7])^3,0)
d$k8<-ifelse(d$time>(knots[8]),(d$time-knots[8])^3,0)
d$k9<-ifelse(d$time>(knots[9]),(d$time-knots[9])^3,0)
d$k10<-ifelse(d$time>(knots[10]),(d$time-knots[10])^3,0)
d$k11<-ifelse(d$time>(knots[11]),(d$time-knots[11])^3,0)
d$k12<-ifelse(d$time>(knots[12]),(d$time-knots[12])^3,0)
d$k13<-ifelse(d$time>(knots[13]),(d$time-knots[13])^3,0)
d$k14<-ifelse(d$time>(knots[14]),(d$time-knots[14])^3,0)
d$k15<-ifelse(d$time>(knots[15]),(d$time-knots[15])^3,0)
```

Add the quadratic and cubic terms

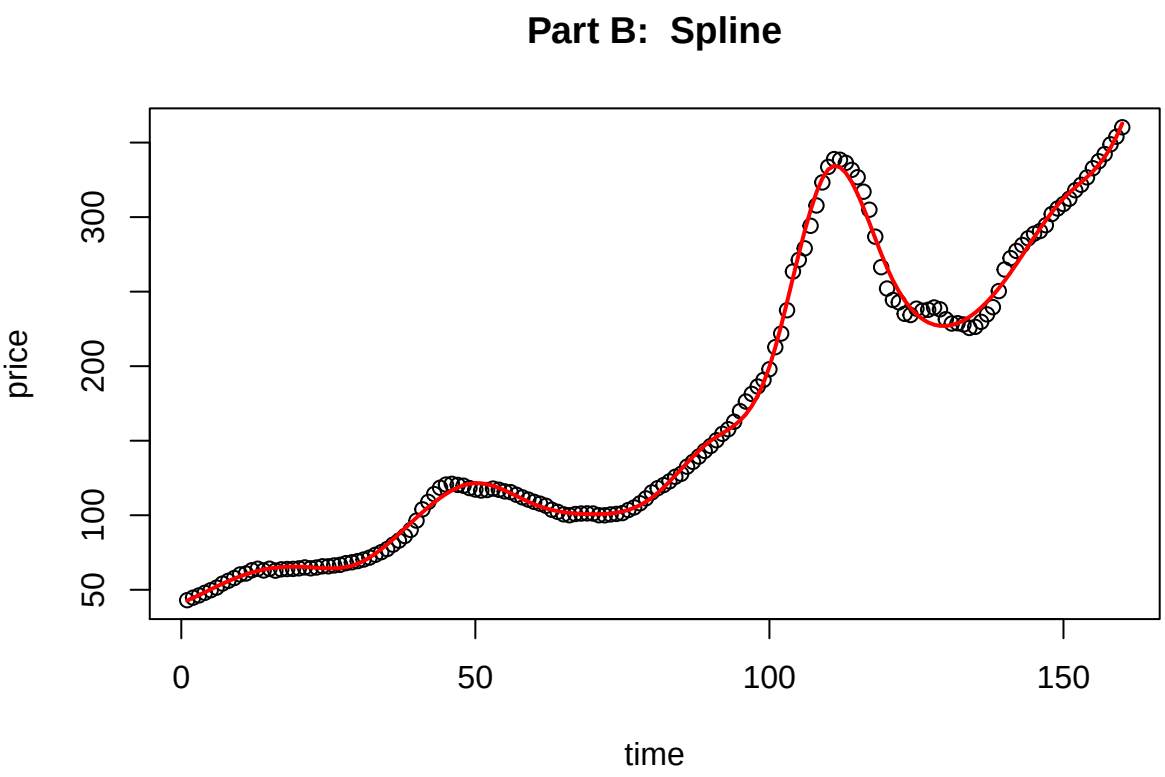
```
d$time2<-d$time^2
d$time3<-d$time^3
```

Estimate the spline

```
splinelm<-lm(price~time+time2+time3+k1+k2+k3+k4+k5+k6+k7+k8+k9+k10+k11+k12+k13+k14+k15,d)
```

And add to the new plot

```
plot(price~time,d,main="Part B:  Spline")
lines(predict(splinelm,data=d)~d$time,col=2,lwd=2)
```



Grading Criteria Correct code for generating spline [8]
Correct figure [12]

Part C

For this question, answers should allow for 16 possible knots. We will start from placement at 0 and iterate the starting point higher, choosing the best starting oint via cross validation.

```
knots0<-c(0,10,20,30,40,50,60,70,80,90,100,110,120,130,140,150)

rm(results)
    for(h in 0:10){
    for(i in 1:nrow(d)){

    knots<-knots0+h

    sd<-d[i,]
    dd<-d[-i,]

        dd$k1<-ifelse(dd$time>(knots[1]),(dd$time-knots[1])^3,0)
        dd$k2<-ifelse(dd$time>(knots[2]),(dd$time-knots[2])^3,0)
        dd$k3<-ifelse(dd$time>(knots[3]),(dd$time-knots[3])^3,0)
        dd$k4<-ifelse(dd$time>(knots[4]),(dd$time-knots[4])^3,0)
        dd$k5<-ifelse(dd$time>(knots[5]),(dd$time-knots[5])^3,0)
```

```
            dd$k6<-ifelse(dd$time>(knots[6]),(dd$time-knots[6])^3,0)
            dd$k7<-ifelse(dd$time>(knots[7]),(dd$time-knots[7])^3,0)
            dd$k8<-ifelse(dd$time>(knots[8]),(dd$time-knots[8])^3,0)
            dd$k9<-ifelse(dd$time>(knots[9]),(dd$time-knots[9])^3,0)
            dd$k10<-ifelse(dd$time>(knots[10]),(dd$time-knots[10])^3,0)
            dd$k11<-ifelse(dd$time>(knots[11]),(dd$time-knots[11])^3,0)
            dd$k12<-ifelse(dd$time>(knots[12]),(dd$time-knots[12])^3,0)
            dd$k13<-ifelse(dd$time>(knots[13]),(dd$time-knots[13])^3,0)
            dd$k14<-ifelse(dd$time>(knots[14]),(dd$time-knots[14])^3,0)
            dd$k15<-ifelse(dd$time>(knots[15]),(dd$time-knots[15])^3,0)
            dd$k16<-ifelse(dd$time>(knots[16]),(dd$time-knots[16])^3,0)

        fit<-lm(price~time+time2+time3+k1+k2+k3+k4+k5+k6+k7+k8+k9+k10+k11+k12+k13+k14+k15+k16,dd)

            sd$k1<-ifelse(sd$time>(knots[1]),(sd$time-knots[1])^3,0)
            sd$k2<-ifelse(sd$time>(knots[2]),(sd$time-knots[2])^3,0)
            sd$k3<-ifelse(sd$time>(knots[3]),(sd$time-knots[3])^3,0)
            sd$k4<-ifelse(sd$time>(knots[4]),(sd$time-knots[4])^3,0)
            sd$k5<-ifelse(sd$time>(knots[5]),(sd$time-knots[5])^3,0)
            sd$k6<-ifelse(sd$time>(knots[6]),(sd$time-knots[6])^3,0)
            sd$k7<-ifelse(sd$time>(knots[7]),(sd$time-knots[7])^3,0)
            sd$k8<-ifelse(sd$time>(knots[8]),(sd$time-knots[8])^3,0)
            sd$k9<-ifelse(sd$time>(knots[9]),(sd$time-knots[9])^3,0)
            sd$k10<-ifelse(sd$time>(knots[10]),(sd$time-knots[10])^3,0)
            sd$k11<-ifelse(sd$time>(knots[11]),(sd$time-knots[11])^3,0)
            sd$k12<-ifelse(sd$time>(knots[12]),(sd$time-knots[12])^3,0)
            sd$k13<-ifelse(sd$time>(knots[13]),(sd$time-knots[13])^3,0)
            sd$k14<-ifelse(sd$time>(knots[14]),(sd$time-knots[14])^3,0)
            sd$k15<-ifelse(sd$time>(knots[15]),(sd$time-knots[15])^3,0)
            sd$k16<-ifelse(sd$time>(knots[16]),(sd$time-knots[16])^3,0)

        dropfit<-predict(fit,sd,se=FALSE)
        sqrerr<-(sd$price-as.numeric(dropfit))^2
        if(exists('results')==TRUE){results<-rbind(results,data.frame(h,i,sqrerr))}
        if(exists('results')==FALSE){results<-data.frame(h,i,sqrerr)}
        }}
```

Here we choose the best starting point.

```
        tapply(results$sqrerr,results$h,FUN=sum,na.rm=TRUE)
```

```
##        0        1        2        3        4        5        6        7
## 4273.891 3586.488 4017.735 5376.059 7179.834 8638.638 9330.301 9003.324
##        8        9       10
## 8845.459 5868.498 4273.891
```

```
    use_h<-as.numeric(names(which.min(tapply(results$sqrerr,results$h,FUN=sum,na.rm=TRUE))))
```

Define knots starting from the best starting point

```
    knots<-knots0+use_h

    d$k1<-ifelse(d$time>(knots[1]),(d$time-knots[1])^3,0)
    d$k2<-ifelse(d$time>(knots[2]),(d$time-knots[2])^3,0)
    d$k3<-ifelse(d$time>(knots[3]),(d$time-knots[3])^3,0)
    d$k4<-ifelse(d$time>(knots[4]),(d$time-knots[4])^3,0)
    d$k5<-ifelse(d$time>(knots[5]),(d$time-knots[5])^3,0)
    d$k6<-ifelse(d$time>(knots[6]),(d$time-knots[6])^3,0)
```

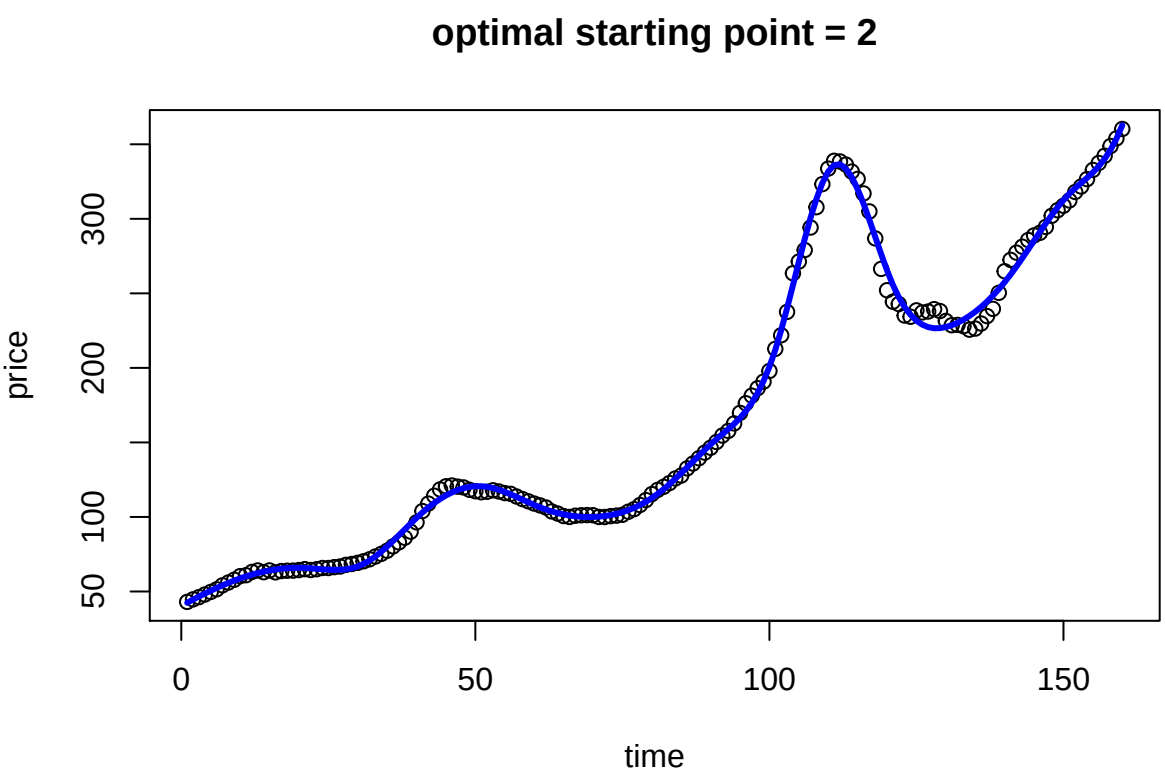
```
    d$k7<-ifelse(d$time>(knots[7]),(d$time-knots[7])^3,0)
    d$k8<-ifelse(d$time>(knots[8]),(d$time-knots[8])^3,0)
    d$k9<-ifelse(d$time>(knots[9]),(d$time-knots[9])^3,0)
    d$k10<-ifelse(d$time>(knots[10]),(d$time-knots[10])^3,0)
    d$k11<-ifelse(d$time>(knots[11]),(d$time-knots[11])^3,0)
    d$k12<-ifelse(d$time>(knots[12]),(d$time-knots[12])^3,0)
    d$k13<-ifelse(d$time>(knots[13]),(d$time-knots[13])^3,0)
    d$k14<-ifelse(d$time>(knots[14]),(d$time-knots[14])^3,0)
    d$k15<-ifelse(d$time>(knots[15]),(d$time-knots[15])^3,0)
    d$k16<-ifelse(d$time>(knots[16]),(d$time-knots[16])^3,0)

    d$time2<-d$time^2
    d$time3<-d$time^3
```

Estimate the spline at the optimal starting point.

```
    splinelm<-lm(price~time+time2+time3+k1+k2+k3+k4+k5+k6+k7+k8+k9+k10+k11+k12+k13+k14+k15+k16,d)
```

And add to the new plot

```
    plot(price~time,d,main=paste("optimal starting point =",use_h+1))
    lines(predict(splinelm,data=d)~d$time,lwd=3,col='blue')
```



Grading Criteria Correct code for generating 10-period increments [8]
Correct figure [12]