Lab 2

Sascha Strobl

7/8/2018

Preparation

Loading necessary dataset:

```
setwd("~/Documents/CGUClasses/ZOLDClasses/Statistical Learning/Lab2");
data = read.table("odor.txt", header = T);
odor_data<-data.matrix(data);
```

Problem 1

a) Generate a function to return LOOCV prediction MSE, with inputs X (matrix) and Y (column vector). Create function name. Open an editing board to generete my function.

```
CV = function (X,Y)
{n = length(Y); u=numeric(length(Y));S=X%*%solve(t(X)%*%X)%*%t(X);for(i in 1:n){u[i]=S[i,i]};mean(((Y-S%
}
```

b) Calculate the LOOCV prediction MSE for the 2 models. Prepare inputs.

```
model_1=lm(Odor~poly(Temp,2)+poly(Height,2)+poly(Ratio,2),data=data);
summary(model_1)
```

```
##
## Call:
## lm(formula = Odor ~ poly(Temp, 2) + poly(Height, 2) + poly(Ratio,
##     2), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.625  -9.625  -1.375   4.021  28.875
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         15.200      4.848   3.136   0.0139 *
## poly(Temp, 2)1     -34.295     18.775  -1.827   0.1052
## poly(Temp, 2)2      61.991     18.879   3.284   0.0111 *
## poly(Height, 2)1   -60.458     18.775  -3.220   0.0122 *
## poly(Height, 2)2    11.754     18.879   0.623   0.5509
## poly(Ratio, 2)1    -48.083     18.775  -2.561   0.0336 *
## poly(Ratio, 2)2     92.423     18.879   4.896   0.0012 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.77 on 8 degrees of freedom
## Multiple R-squared:  0.8683, Adjusted R-squared:  0.7695
## F-statistic: 8.789 on 6 and 8 DF,  p-value: 0.003616
```

```
model_2=lm(Odor~Height+poly(Ratio,2)+I(Temp^2),data=data);
summary(model_2)
```

```
##
```

```
## Call:
## lm(formula = Odor ~ Height + poly(Ratio, 2) + I(Temp^2), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.058  -8.572  -3.058   9.812  31.933
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.662      7.694  -0.216  0.83337
## Height            -21.375      7.187  -2.974  0.01395 *
## poly(Ratio, 2)1   -48.083     20.329  -2.365  0.03960 *
## poly(Ratio, 2)2    91.519     20.381   4.490  0.00116 **
## I(Temp^2)          31.615     10.548   2.997  0.01341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.33 on 10 degrees of freedom
## Multiple R-squared:  0.807,  Adjusted R-squared:  0.7297
## F-statistic: 10.45 on 4 and 10 DF,  p-value: 0.00135
```

```
X_model_1=cbind(rep(1,15), odor_data[,2:4], odor_data[,2]^2,odor_data[,3]^2,odor_data[,4]^2)
X_model_2=cbind(rep(1,15),odor_data[,3:4],odor_data[,2]^2,odor_data[,3]^2)
Y=data$Odor
```

The LOOCV prediction MSE for Model 1:

```
CV(X_model_1,Y)
```

```
## [1] 747.2333
```

The LOOCV prediction MSE for Model 2:

```
CV(X_model_2,Y)
```

```
## [1] 666.8952
```

Conlusion: The LOOCV prediction MSE is in favor of Model 2.

Problem 2

```
library(MASS)
```

a) Use help to check all data variables in "Boston".

```
help(Boston)
```

b) Calculate the median of "tax".

```
median(Boston$tax)
```

```
## [1] 330
```

c) Use bootstrap sampling method to estimate the standard deviation of the estimate of Boston tax median. Generate a function, in terms of bootstrap sample size B.

```
Se = function (X, B)
{n=length(X);
C=numeric(B);
for(i in 1:B){C[i]=median(sample(X,n,replace=TRUE))};
sd(C)};
```

```r
Se(Boston$tax,1000)
```

```
## [1] 13.80111
```

d)

Use the result in c) to find a 95% confidence interval for the median:

```r
upperbound=median(Boston$tax)+Se(Boston$tax,1000)
lowerbound=median(Boston$tax)-Se(Boston$tax,1000)
upperbound
```

```
## [1] 343.784
```

```r
lowerbound
```

```
## [1] 316.3013
```

Problem 3

Apply Poisson regression.

```r
crab = read.table("crab.txt", header = F)
colnames(crab)=c("Obs","C","S","W","Wt","Sa")
Expectation=glm(Sa~W+Wt, data=crab, family=poisson(link=log))
summary(Expectation)
```

```
##
## Call:
## glm(formula = Sa ~ W + Wt, family = poisson(link = log), data = crab)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9308  -1.9705  -0.5481   0.9700   4.9905
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.29168    0.89929  -1.436  0.15091
## W            0.04590    0.04677   0.981  0.32640
## Wt           0.44744    0.15864   2.820  0.00479 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 632.79  on 172  degrees of freedom
## Residual deviance: 559.89  on 170  degrees of freedom
## AIC: 921.18
##
## Number of Fisher Scoring iterations: 6
```

Conclusion: from the p-values, only Crab$Wt significantly explains the value of the response Crab$Sa.

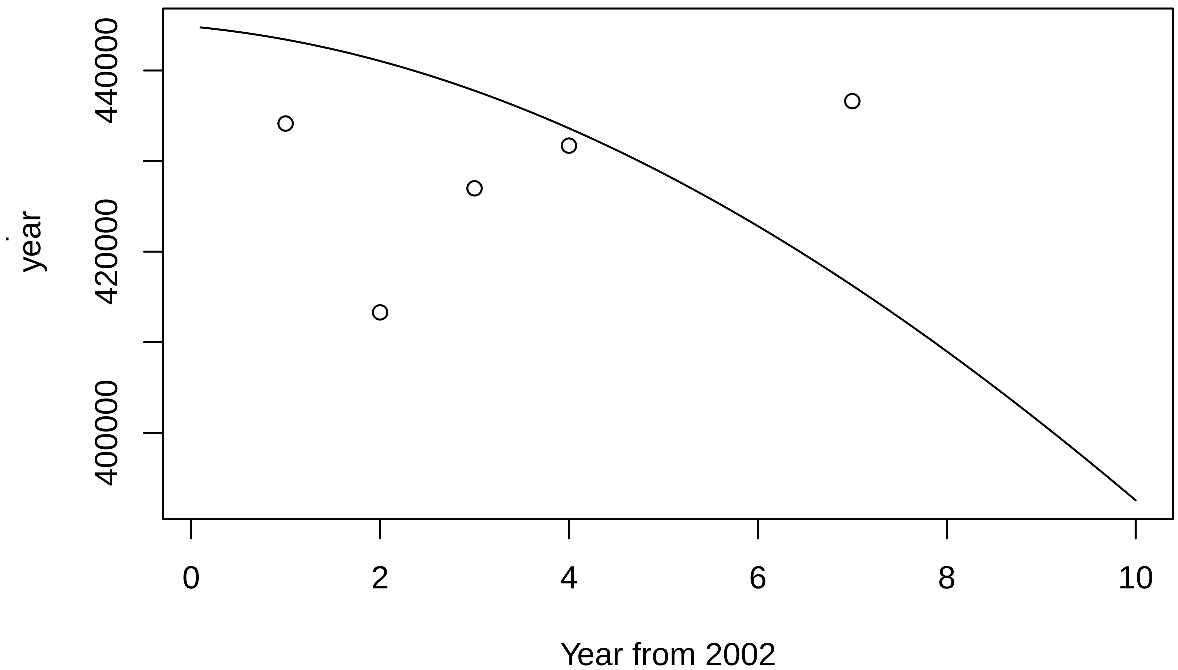
Problem 4

Produce Bass curve as well as the estmates of all parameters, using nonlinear least squares. Attention: here M,P,Q are initial inputs of m,p,q. P=0.5, Q=0.65 are initial values (the initial values should be well-chosen. We have made many tries to reduce the p-value).

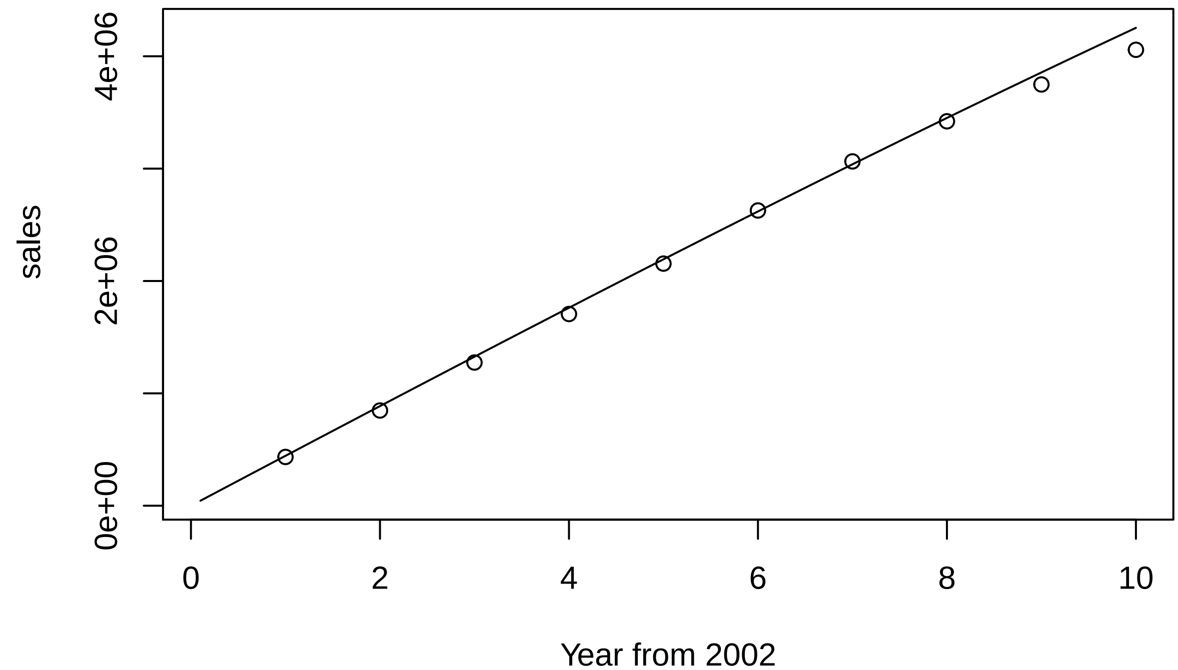
```
library(minpack.lm)
ts = read.table("ToyotaSales.txt", header = T);
Camry=ts$CamrySales
Cruiser=ts$FJCruiserSales;
Cruiser=Cruiser[-c(1,2,3,4)];
Cruiser<-as.numeric(paste(Cruiser))
```

a) Use Bass curve to fit Toyota. Give estimates of its m; p; q and provide its Bass curve. Does Bass model fit Camry?

```
T02=1:15;
Tdelt =(1:100) / 10;
Bass1.nls= nlsLM(Camry ~ M *(((P+Q)^2/P)*exp(-(P+Q)*T02))/(1+(Q/P)*exp(-(P+Q)*T02))^2,start = list(M=sur

## Warning in nls.lm(par = start, fn = FCT, jac = jac, control = control, lower = lower, : lmdif: info =

summary(Bass1.nls);

##
## Formula: Camry ~ M * (((P + Q)^2/P) * exp(-(P + Q) * T02))/(1 + (Q/P) *
##     exp(-(P + Q) * T02))^2
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## M 1.323e+07  7.442e+06   1.777   0.1008
## P 3.363e-02  1.645e-02   2.045   0.0634 .
## Q 3.144e-02  4.862e-02   0.647   0.5300
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52490 on 12 degrees of freedom
##
## Number of iterations till stop: 50
## Achieved convergence tolerance: 1.49e-08
## Reason stopped: Number of iterations has reached `maxiter' == 50.

Bcoef=coef(Bass1.nls);
Cusales=cumsum(Camry);
m=Bcoef[1];
p=Bcoef[2];
q=Bcoef[3];
ngete = exp(-(p+q) * Tdelt);
Bpdf=m * ( (p+q)^2 / p ) * ngete / (1 + (q/p) * ngete)^2;
plot(Tdelt, Bpdf, xlab = "Year from 2002",ylab = "Sales per
     year", type='l');
points(T02, Camry);
```




```
Bcdf= m * (1 - ngete)/(1 + (q/p)*ngete);
plot(Tdelt, Bcdf, xlab = "Year from 2002",ylab = "Cumulative
     sales", type='l');
points(T02, Cusales);
```



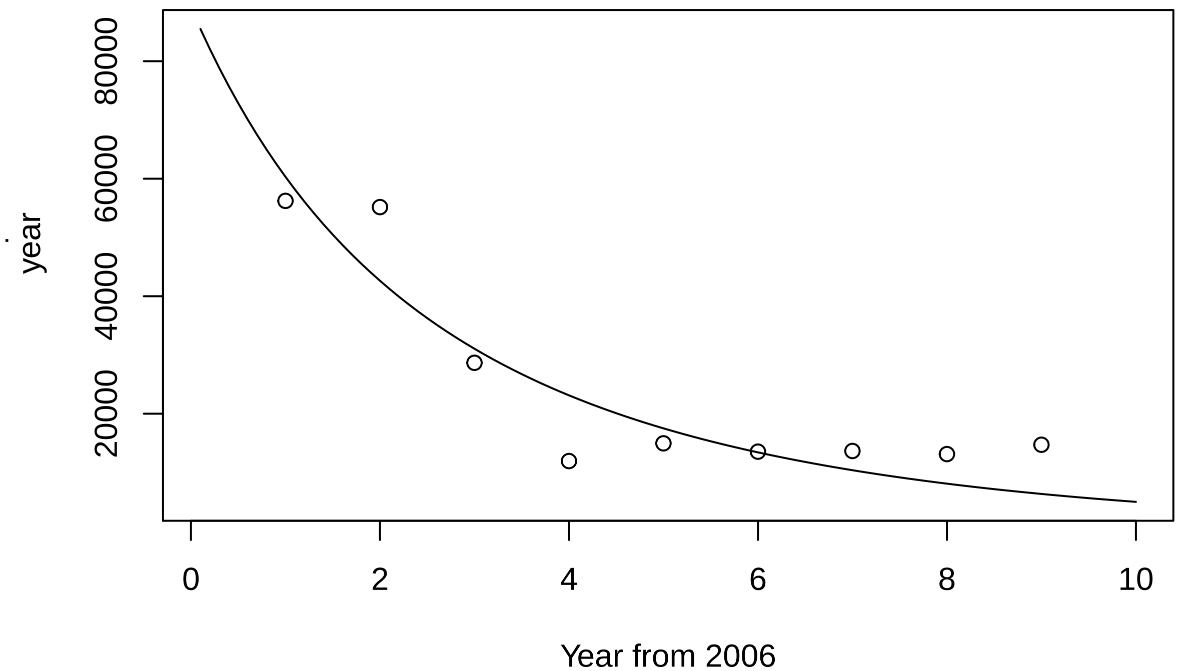
b) Use Bass curve to fit Cruiser. Give estimates of its m; p; q and provide its Bass curve. Compare the 2 Bass curves and conclude: which series has better sales performance?

```
T=1:11
set.seed(1)
Bass2.nls= nls(Cruiser ~ M *(((P+Q)^2/P)*exp(-(P+Q)*T))/(1+(Q/P)*exp(-(P+Q)*T))^2,start = list(M=sum(Cru
```

```r
summary(Bass2.nls)
```

```
##
## Formula: Cruiser ~ M * (((P + Q)^2/P) * exp(-(P + Q) * T))/(1 + (Q/P) *
##     exp(-(P + Q) * T))^2
##
## Parameters:
##     Estimate Std. Error t value Pr(>|t|)
## M  2.788e+05  5.039e+04   5.532 0.000553 ***
## P  3.197e-01  7.847e-02   4.073 0.003566 **
## Q -9.798e-02  2.475e-01  -0.396 0.702564
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7548 on 8 degrees of freedom
##
## Number of iterations to convergence: 8
## Achieved convergence tolerance: 4.839e-06
```

```r
Bcoef=coef(Bass2.nls)
Cusales=cumsum(Cruiser)
Tdelt =(1:100) / 10
m=Bcoef[1]
p=Bcoef[2]
q=Bcoef[3]
ngete = exp(-(p+q) * Tdelt)
Bpdf=m * ( (p+q)^2 / p ) * ngete / (1 + (q/p) * ngete)^2
plot(Tdelt, Bpdf, xlab = "Year from 2006",ylab = "Sales per
     year", type='l')
points(T, Cruiser)
```




```r
Bcdf= m * (1 - ngete)/(1 + (q/p)*ngete)
plot(Tdelt, Bcdf, xlab = "Year from 2006",ylab = "Cumulative
```

```
      sales", type='l')
points(T, Cusales)
```