STAT 5385: Lab 6

Willliam Ofosu Agyapong

26/03/2022

0.1 Data Read-in

```r
# Reading in required data
toluca <- read.table("../Data Sets/Chapter  1 Data Sets/CH01TA01.txt")
colnames(toluca) <- c("lot_size","Work_hours")
kable(head(toluca, 6), caption = "toluca data set")
```

Table 1: toluca data set

lot_size	Work_hours
80	399
30	121
50	221
90	376
70	361
60	224

```r
senic <- read.table("../Data Sets/Appendix C Data Sets/APPENC01.txt")
colnames(senic) <- c("ID","LOS","Age","Infec","Cul","Xray","beds","Med","region","avg","nurses","fands")
kable(head(senic,6), caption = "The SENIC data set")
```

Table 2: The SENIC data set

ID	LOS	Age	Infec	Cul	Xray	beds	Med	region	avg	nurses	fands
1	7.13	55.7	4.1	9.0	39.6	279	2	4	207	241	60
2	8.82	58.2	1.6	3.8	51.7	80	2	2	51	52	40
3	8.34	56.9	2.7	8.1	74.0	107	2	3	82	54	20
4	8.95	53.7	5.6	18.9	122.8	147	2	4	53	148	40
5	11.20	56.5	5.7	34.5	88.9	180	2	1	134	151	40
6	9.76	50.9	5.1	21.9	97.0	150	2	2	147	106	40

```r
# toluca %>% dfSummary() %>% view()
```

0.2 Basic matrix calculations and examples

```r
library(matlib)
library(MASS)

#-------- Using the toluca data
j <- rep(1,nrow(toluca)) # create a vector of ones for the intercept
X <- cbind(j,toluca$lot_size) # Create the design matrix
```

```r
y <- toluca$Work_hours # Extract response variable
kable(t(X))# making sure everything went right
```

j	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	80	30	50	90	70	60	120	80	100	50	40	70	90	20	110	100	30	50	90	110	30	90	40	80

```r
#--------- Using the SENIC data
names(senic) # just to remind ourselves of the variable names
```

```
##  [1] "ID"     "LOS"    "Age"    "Infec"  "Cul"    "Xray"   "beds"   "Med"   
##  [9] "region" "avg"    "nurses" "fands"
```

```r
j <- rep(1, nrow(senic))
X2 <- as.matrix(cbind(j, senic[, c(4,6,12)]))
kable(head(X2)) # looking at few rows to make sure everything went right
```

j	Infec	Xray	fands
1	4.1	39.6	60
1	1.6	51.7	40
1	2.7	74.0	20
1	5.6	122.8	40
1	5.7	88.9	40
1	5.1	97.0	40

```r
y2 <- senic$LOS
```

0.3 Regression matrix calculations

```r
# X'X
xpx <- t(X)%*%X # not X*X
#X'y
xpy <- t(X)%*%y # not X*y
#y'y
ypy <- t(y)%*%y # not y*y
#finding matrix inverse
solve(xpx);inv(xpx)
```

```
##           j          
## j  0.287475 -3.535e-03
##   -0.003535  5.051e-05
```

```
##                       
## [1,]  0.287475 -3.535e-03
## [2,] -0.003535  5.051e-05
```

```r
#beta vector
(beta <- inv(xpx)%*%xpy)
```

```
##        [,1]
## [1,] 62.368
## [2,]  3.573
```

```r
#now try for multivariate data
xpx2 <- t(X2)%*%X2
xpy2 <- t(X2)%*%y2
```

```
ypy2 <- t(y2)%*%y2

(beta2 <- inv(xpx2)%*%xpy2)
```

```
##         [,1]
## [1,] 4.80042
## [2,] 0.52852
## [3,] 0.01901
## [4,] 0.02292
```

0.4 Computations from Chapter 5 notes

```
#first estimate a lm object
mod0 <- lm(LOS~Infec+Xray+fands,data=senic)
summary(mod0)
```

```
##
## Call:
## lm(formula = LOS ~ Infec + Xray + fands, data = senic)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.678 -0.882 -0.202  0.697  7.976
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.80088    0.74230    6.47  2.9e-09 ***
## Infec        0.52862    0.13672    3.87  0.00019 ***
## Xray         0.01916    0.00868    2.21  0.02935 *
## fands        0.02274    0.01082    2.10  0.03789 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.58 on 109 degrees of freedom
## Multiple R-squared:  0.336,  Adjusted R-squared:  0.318
## F-statistic: 18.4 on 3 and 109 DF,  p-value: 9.78e-10
```

```
anova(mod0)
```

```
## Analysis of Variance Table
##
## Response: LOS
##            Df Sum Sq Mean Sq F value  Pr(>F)
## Infec       1  116.4   116.4   46.74 4.9e-10 ***
## Xray        1   10.2    10.2    4.09   0.046 *
## fands       1   11.0    11.0    4.42   0.038 *
## Residuals 109  271.6     2.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vcov(mod0) # variance covariance matrix for the beta coefficients
```

```
##             (Intercept)      Infec       Xray      fands
## (Intercept)    0.551016 -0.0117535 -4.199e-03 -3.129e-03
## Infec         -0.011753  0.0186937 -5.338e-04 -6.043e-04
## Xray          -0.004199 -0.0005338  7.532e-05  8.689e-06
## fands         -0.003129 -0.0006043  8.689e-06  1.170e-04
```

```
#now the matrix calculations
MSE <- 2.492
MSE*inv(xpx2) # the same as using vcov(mod0)
```

```
##
## [1,]  0.551126 -0.0117558 -4.200e-03 -3.129e-03
## [2,] -0.011756  0.0186974 -5.339e-04 -6.044e-04
## [3,] -0.004200 -0.0005339  7.533e-05  8.697e-06
## [4,] -0.003129 -0.0006044  8.697e-06  1.170e-04
```

```
#some needed extra cals
ypJy <- t(y2)%*%matrix(1,nrow(senic),nrow(senic))%*%y2
hatmat <- X2%*%xpx2%*%t(X2) # dim: n by n
resid <- y2-X2%*%beta2

kable(head(cbind(resid, mod0$residuals)), col.names = c("By hand", "From lm model"),
      caption = "Comparing residuals")
```

Table 5: Comparing residuals

By hand	From lm model
-1.9656	-1.9612
1.2741	1.2733
0.2471	0.2392
-2.0619	-2.0736
0.7798	0.7731
-0.4971	-0.5049

```
#sums of squares
(SSTO <- ypy2-1/nrow(senic)*ypJy)
```

```
##       [,1]
## [1,] 409.2
```

```
(SSE <- t(resid)%*%resid)
```

```
##       [,1]
## [1,] 271.6
```

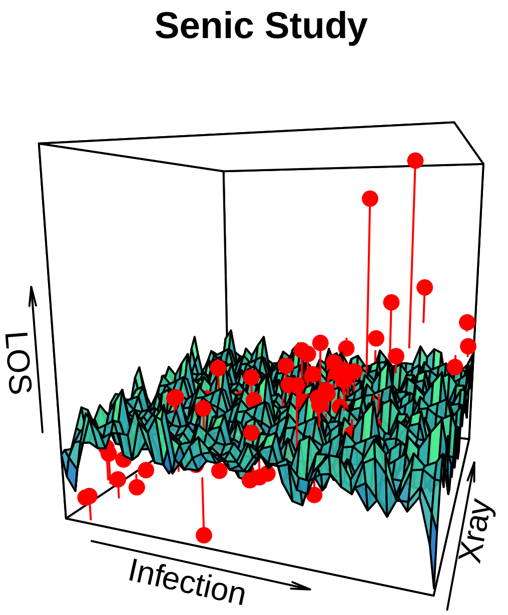
```
(SSR <- t(beta2)%*%xpy2-1/nrow(senic)*ypJy)
```

```
##       [,1]
## [1,] 132.2
```

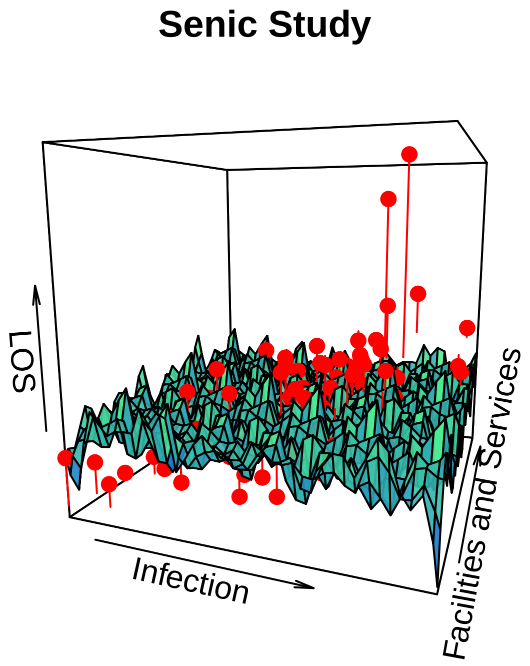
0.5 Now some multivariate modeling

```
library(plot3D)

# set the variables
x1 <- senic$Infec
x2 <- senic$Xray
x3 <- senic$fands
y <- senic$LOS

# Compute the linear regression
fit <- lm(y ~ x1 + x2 + x3)
```

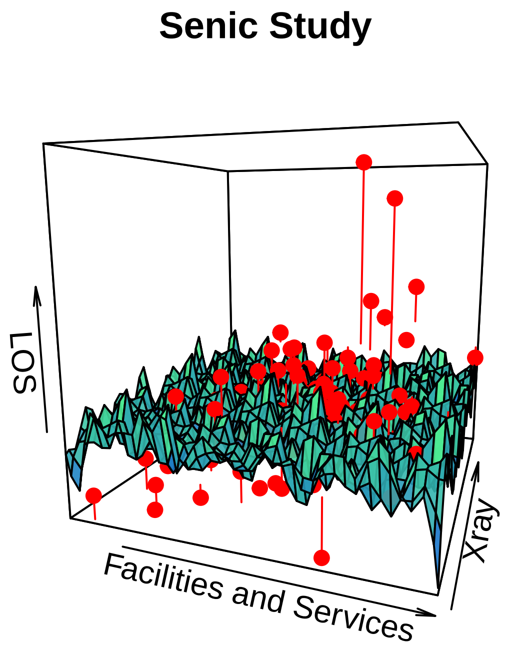
```
# create a grid from the x and y values (min to max) and predict values for every point
# this will become the regression plane
grid.lines = 40
x1.pred <- seq(min(x1), max(x1), length.out = grid.lines)
x2.pred <- seq(min(x2), max(x2), length.out = grid.lines)
x3.pred <- seq(min(x3), max(x3), length.out = grid.lines)
x1x2 <- expand.grid( x = x1.pred, y = x2.pred)
y.pred <- matrix(predict(fit, newdata = x1x2),nrow = grid.lines, ncol = grid.lines)
x1x3 <- expand.grid( x = x1.pred, y = x3.pred)
y.pred <- matrix(predict(fit, newdata = x1x3),nrow = grid.lines, ncol = grid.lines)
x3x2 <- expand.grid( x = x3.pred, y = x2.pred)
y.pred <- matrix(predict(fit, newdata = x3x2),nrow = grid.lines, ncol = grid.lines)
# create the fitted points for droplines to the surface
fitpoints <- predict(fit)

# scatter plot with regression plane
scatter3D(x1, x2, y, pch = 19, cex = 1,colvar = NULL, col="red",
          theta = 20, phi = 10, bty="b",
          xlab = "Infection", ylab = "Xray", zlab = "LOS",
          surf = list(x = x1.pred, y = x2.pred, z = y.pred,
                      facets = TRUE, fit = fitpoints, col=ramp.col (col = c("dodgerblue3","seagreen2"), n
```



```
scatter3D(x1, x3, y, pch = 19, cex = 1,colvar = NULL, col="red",
          theta = 20, phi = 10, bty="b",
          xlab = "Infection", ylab = "Facilities and Services", zlab = "LOS",  surf = list(x = x1.pred, y
```



```
scatter3D(x3, x2, y, pch = 19, cex = 1,colvar = NULL, col="red",
          theta = 20, phi = 10, bty="b",
          xlab = "Facilities and Services", ylab = "Xray", zlab = "LOS",  surf = list(x = x3.pred, y = x2.
```


Senic Study
LOS
Facilities and Services
Xray