Midterm\_Stat4310

Regis Ouedraogo

10/13/2022

Problem 1

years<- c(1976,1977,1978,1979,1980,1981,1982,1983,1984,1985,1986,1987,1988,1989,1990,1991)  
index<- c(16.7,17.1,18.2,18.1,17.2,18.2,16.0,17.2,18.0,17.2,16.9,17.1,18.2,17.3,17.5,16.6)  
days <- c(91,105,106,108,91,58,82,65,61,48,61,43,36)  
  
data <- cbind(years,days,index)

## Warning in cbind(years, days, index): number of rows of result is not a multiple  
## of vector length (arg 2)

data <- data.frame(data)  
data

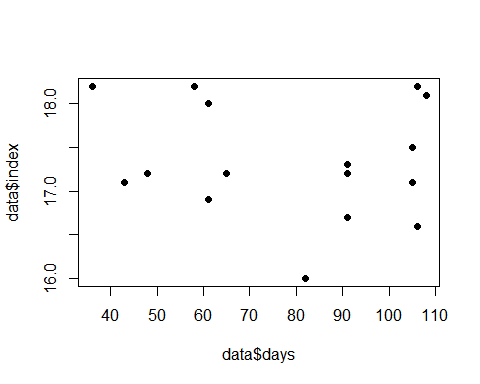
## years days index  
## 1 1976 91 16.7  
## 2 1977 105 17.1  
## 3 1978 106 18.2  
## 4 1979 108 18.1  
## 5 1980 91 17.2  
## 6 1981 58 18.2  
## 7 1982 82 16.0  
## 8 1983 65 17.2  
## 9 1984 61 18.0  
## 10 1985 48 17.2  
## 11 1986 61 16.9  
## 12 1987 43 17.1  
## 13 1988 36 18.2  
## 14 1989 91 17.3  
## 15 1990 105 17.5  
## 16 1991 106 16.6

attach(data)

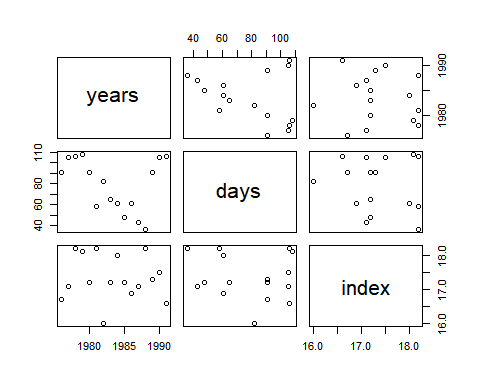
## The following objects are masked \_by\_ .GlobalEnv:  
##   
## days, index, years

Scatterplot

plot(data$days, data$index, pch=16)



pairs(data)

 Lets build a linear model

model <- lm(index ~ days, data = data)  
summary(model)

##   
## Call:  
## lm(formula = index ~ days, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.3312 -0.4074 -0.1226 0.6191 0.9567   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17.631265 0.564517 31.233 2.39e-14 \*\*\*  
## days -0.003660 0.006864 -0.533 0.602   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6683 on 14 degrees of freedom  
## Multiple R-squared: 0.0199, Adjusted R-squared: -0.0501   
## F-statistic: 0.2843 on 1 and 14 DF, p-value: 0.6023

# fitted values; prediction bands  
names(model)

## [1] "coefficients" "residuals" "effects" "rank"   
## [5] "fitted.values" "assign" "qr" "df.residual"   
## [9] "xlevels" "call" "terms" "model"

fitted <- model$fitted.values  
fitted

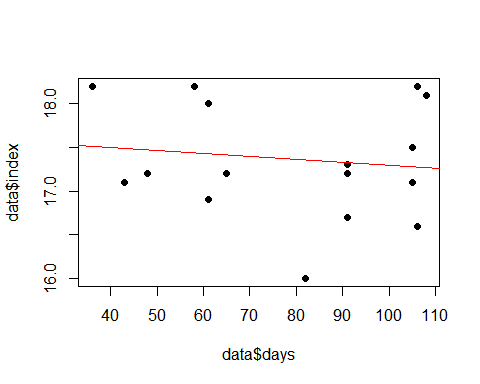
## 1 2 3 4 5 6 7 8   
## 17.29823 17.24700 17.24334 17.23602 17.29823 17.41900 17.33117 17.39338   
## 9 10 11 12 13 14 15 16   
## 17.40802 17.45560 17.40802 17.47390 17.49952 17.29823 17.24700 17.24334

attach(data)

## The following objects are masked \_by\_ .GlobalEnv:  
##   
## days, index, years

## The following objects are masked from data (pos = 3):  
##   
## days, index, years

plot(data$days, data$index,pch=16)  
abline(a=17.631265 , b= -0.003380, col="red")



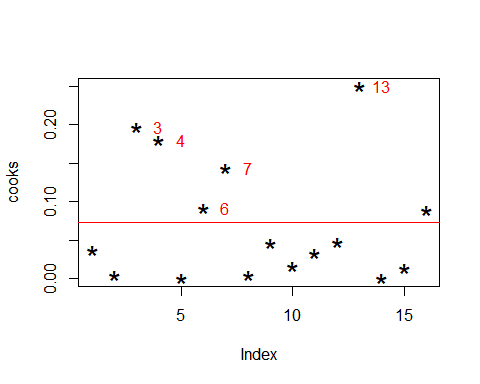
anova(model)

## Analysis of Variance Table  
##   
## Response: index  
## Df Sum Sq Mean Sq F value Pr(>F)  
## days 1 0.1270 0.12697 0.2843 0.6023  
## Residuals 14 6.2524 0.44660

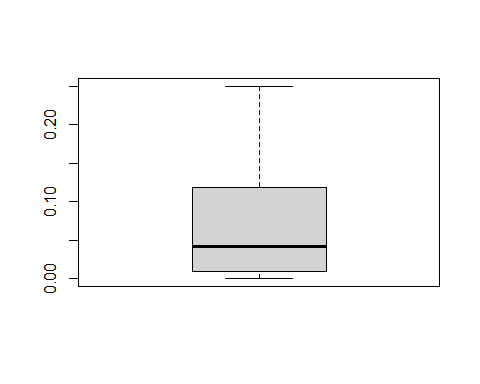
cooks <- cooks.distance(model)  
summary(cooks)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.200e-07 1.092e-02 4.137e-02 7.222e-02 1.049e-01 2.501e-01

plot(cooks, pch="\*", cex=2)  
abline(h = mean(cooks, na.rm=T), col="red") # add cutoff line  
text(x=1:length(cooks)+1, y=cooks, labels=ifelse(cooks>mean(cooks, na.rm=T),names(cooks),""), col="red")



boxplot(cooks)



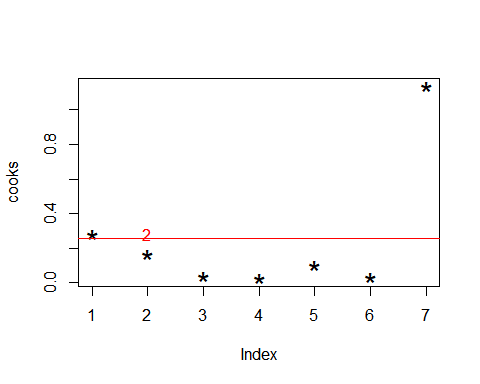
# Build new model on new data frame  
data <- data[ -c(3,4,7,6,13,15,9,14,1), ] #take out data points  
model2<- lm(index ~ days, data=data)  
summary(model2)

##   
## Call:  
## lm(formula = index ~ days, data = data)  
##   
## Residuals:  
## 2 5 8 10 11 12 16   
## 0.1694 0.2185 0.1239 0.0620 -0.1907 -0.0562 -0.3269   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17.312690 0.266210 65.034 1.63e-08 \*\*\*  
## days -0.003639 0.003411 -1.067 0.335   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.22 on 5 degrees of freedom  
## Multiple R-squared: 0.1855, Adjusted R-squared: 0.02256   
## F-statistic: 1.138 on 1 and 5 DF, p-value: 0.3348

cooks <- cooks.distance(model2)  
summary(cooks)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.02540 0.03417 0.10409 0.25425 0.22327 1.13539

plot(cooks, pch="\*", cex=2)  
abline(h = mean(cooks, na.rm=T), col="red") # add cutoff line  
text(x=1:length(cooks)+1, y=cooks, labels=ifelse(cooks>mean(cooks, na.rm=T),names(cooks),""), col="red")



prediction <- predict(model2, newdata=data.frame(data$days), se=T)

## Warning: 'newdata' had 7 rows but variables found have 13 rows

prediction

## $fit  
## 1 2 3 4 5 6 7 8   
## 16.98151 16.93056 16.92692 16.91964 16.98151 17.10161 17.01426 17.07613   
## 9 10 11 12 13   
## 17.09069 17.13800 17.09069 17.15620 17.18167   
##   
## $se.fit  
## 1 2 3 4 5 6 7   
## 0.10109950 0.13413604 0.13682866 0.14230633 0.10109950 0.09973407 0.08736947   
## 8 9 10 11 12 13   
## 0.08881275 0.09447102 0.12192749 0.09447102 0.13490204 0.15440513   
##   
## $df  
## [1] 5  
##   
## $residual.scale  
## [1] 0.2200148

Build an upper and lower bounds for prediction

#Prediction and Prediction bands  
  
 #fitted values; prediction bands  
  
fitted <- model$fitted.values  
fitted

## 1 2 3 4 5 6 7 8   
## 17.29823 17.24700 17.24334 17.23602 17.29823 17.41900 17.33117 17.39338   
## 9 10 11 12 13 14 15 16   
## 17.40802 17.45560 17.40802 17.47390 17.49952 17.29823 17.24700 17.24334

upper<- fitted +abs( qt(.025, 16))\*prediction$se.fit

## Warning in fitted + abs(qt(0.025, 16)) \* prediction$se.fit: longer object length  
## is not a multiple of shorter object length

lower<- fitted -abs( qt(.025, 16))\*prediction$se.fit

## Warning in fitted - abs(qt(0.025, 16)) \* prediction$se.fit: longer object length  
## is not a multiple of shorter object length

lower

## 1 2 3 4 5 6 7 8   
## 17.08391 16.96264 16.95327 16.93434 17.08391 17.20758 17.14595 17.20511   
## 9 10 11 12 13 14 15 16   
## 17.20775 17.19712 17.20775 17.18792 17.17219 17.08391 16.96264 16.95327

upper

## 1 2 3 4 5 6 7 8   
## 17.51255 17.53135 17.53340 17.53769 17.51255 17.63043 17.51638 17.58166   
## 9 10 11 12 13 14 15 16   
## 17.60829 17.71407 17.60829 17.75988 17.82684 17.51255 17.53135 17.53340

Problem 2

library(readxl)  
data <- read\_excel("data-prob-2-18.xls")

## Warning: replacing previous import 'lifecycle::last\_warnings' by  
## 'rlang::last\_warnings' when loading 'tibble'

## Warning: replacing previous import 'lifecycle::last\_warnings' by  
## 'rlang::last\_warnings' when loading 'pillar'

attach(data)  
colnames(data)[2] ="budget"  
colnames(data)[3] ="impressions"  
data

## # A tibble: 21 x 3  
## Firm budget impressions  
## <chr> <dbl> <dbl>  
## 1 Miller Lite 50.1 32.1  
## 2 Pepsi 74.1 99.6  
## 3 Stroh's 19.3 11.7  
## 4 Federal Express 22.9 21.9  
## 5 Burger King 82.4 60.8  
## 6 Coca-Cola 40.1 78.6  
## 7 McDonald's 186. 92.4  
## 8 MCI 26.9 50.7  
## 9 Diet Cola 20.4 21.4  
## 10 Ford 166. 40.1  
## # ... with 11 more rows

Lets fit the linear model

model <- lm(impressions ~ budget ,data = data)  
summary(model)

##   
## Call:  
## lm(formula = impressions ~ budget, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -42.422 -12.623 -8.171 8.832 50.526   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 22.16269 7.08948 3.126 0.00556 \*\*  
## budget 0.36317 0.09712 3.739 0.00139 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 23.5 on 19 degrees of freedom  
## Multiple R-squared: 0.424, Adjusted R-squared: 0.3936   
## F-statistic: 13.98 on 1 and 19 DF, p-value: 0.001389

P-value (0.001389) < 0.05, Reject H0. there is a significant relationship between the amount that a company spends on advertising and retained impressions

fitted <- model$fitted.values  
fitted

## 1 2 3 4 5 6 7 8   
## 40.35771 49.07389 29.17195 30.47938 52.08824 36.72597 89.67675 31.93208   
## 9 10 11 12 13 14 15 16   
## 29.57144 82.52222 31.96839 38.72343 78.41836 23.97856 40.21244 31.93208   
## 17 18 19 20 21   
## 24.23279 24.92282 25.50389 33.92953 24.37806

upper<- fitted +abs( qt(.025, 21))\*prediction$se.fit

## Warning in fitted + abs(qt(0.025, 21)) \* prediction$se.fit: longer object length  
## is not a multiple of shorter object length

lower<- fitted -abs( qt(.025, 21))\*prediction$se.fit

## Warning in fitted - abs(qt(0.025, 21)) \* prediction$se.fit: longer object length  
## is not a multiple of shorter object length

upper

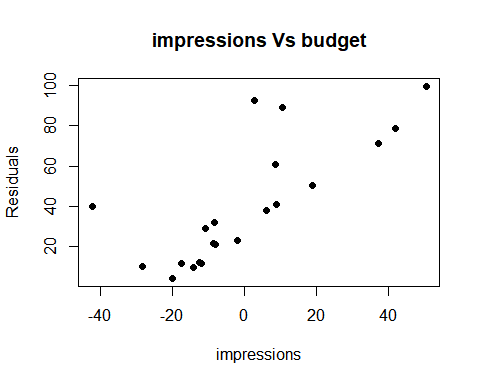
## 1 2 3 4 5 6 7 8   
## 40.56796 49.35284 29.45650 30.77532 52.29848 36.93338 89.85845 32.11677   
## 9 10 11 12 13 14 15 16   
## 29.76791 82.77579 32.16486 39.00398 78.73946 24.18881 40.49140 32.21663   
## 17 18 19 20 21   
## 24.52873 25.13306 25.71130 34.11123 24.56275

lower

## 1 2 3 4 5 6 7 8   
## 40.14747 48.79494 28.88740 30.18344 51.87799 36.51857 89.49506 31.74738   
## 9 10 11 12 13 14 15 16   
## 29.37498 82.26866 31.77193 38.44289 78.09725 23.76832 39.93349 31.64753   
## 17 18 19 20 21   
## 23.93684 24.71257 25.29649 33.74784 24.19336

Lets plot the residuals

model\_res <- model$residuals  
plot(model\_res, data$impressions, pch=16, ylab = "Residuals", xlab = "impressions", main = "impressions Vs budget")  
abline(0,0)



There are few influenced variables having high variance with the mean line.

1. 95% confidence interval for coefficient(Slope):

#ges("Rmisc")  
library(Rmisc)

## Warning: package 'Rmisc' was built under R version 4.1.3

## Loading required package: lattice

## Loading required package: plyr

## Warning: package 'plyr' was built under R version 4.1.3

CI(data$MCI, ci=0.95)

## Warning: Unknown or uninitialised column: `MCI`.

## Warning in mean.default(x): argument is not numeric or logical: returning NA

## Warning in qt(ci + (1 - ci)/2, df = n - 1): NaNs produced

## upper mean lower   
## NA NA NA

Problem 3.7:

#install.packages("MPV")  
library(MPV)

## Warning: package 'MPV' was built under R version 4.1.3

## Loading required package: KernSmooth

## KernSmooth 2.23 loaded  
## Copyright M. P. Wand 1997-2009

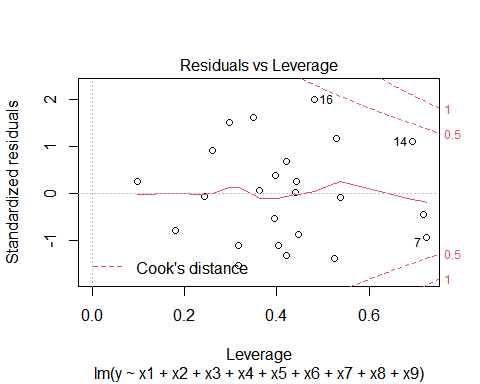
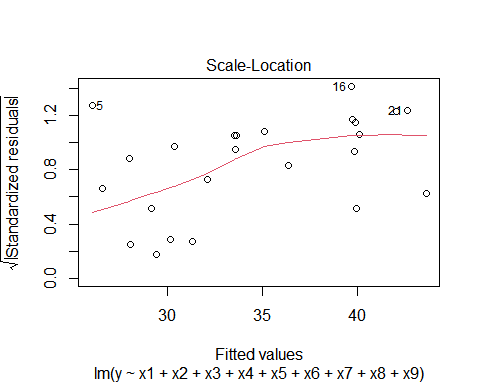
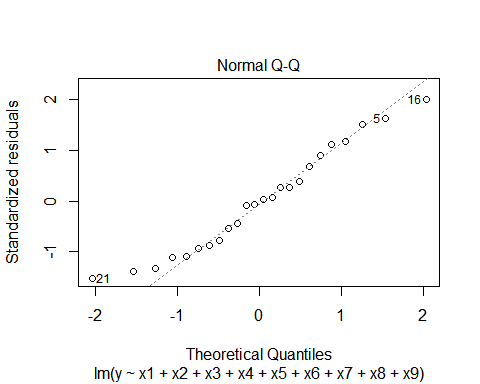
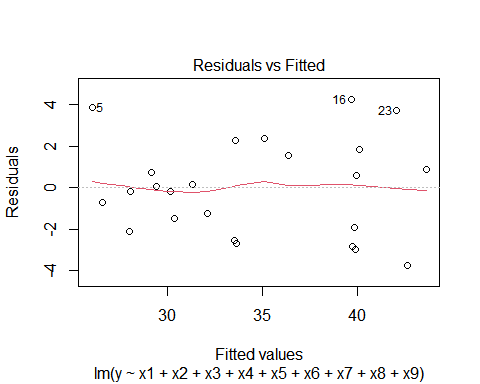
data(table.b4)  
attach(table.b4)  
model <- lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9, table.b4)  
summary(model)

##   
## Call:  
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9,   
## data = table.b4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.720 -1.956 -0.045 1.627 4.253   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 14.92765 5.91285 2.525 0.0243 \*  
## x1 1.92472 1.02990 1.869 0.0827 .  
## x2 7.00053 4.30037 1.628 0.1258   
## x3 0.14918 0.49039 0.304 0.7654   
## x4 2.72281 4.35955 0.625 0.5423   
## x5 2.00668 1.37351 1.461 0.1661   
## x6 -0.41012 2.37854 -0.172 0.8656   
## x7 -1.40324 3.39554 -0.413 0.6857   
## x8 -0.03715 0.06672 -0.557 0.5865   
## x9 1.55945 1.93750 0.805 0.4343   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.949 on 14 degrees of freedom  
## Multiple R-squared: 0.8531, Adjusted R-squared: 0.7587   
## F-statistic: 9.037 on 9 and 14 DF, p-value: 0.000185

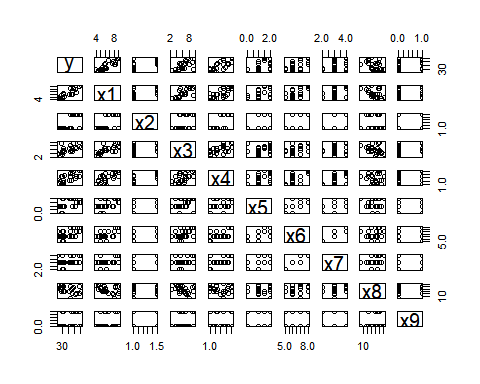
data

## # A tibble: 21 x 3  
## Firm budget impressions  
## <chr> <dbl> <dbl>  
## 1 Miller Lite 50.1 32.1  
## 2 Pepsi 74.1 99.6  
## 3 Stroh's 19.3 11.7  
## 4 Federal Express 22.9 21.9  
## 5 Burger King 82.4 60.8  
## 6 Coca-Cola 40.1 78.6  
## 7 McDonald's 186. 92.4  
## 8 MCI 26.9 50.7  
## 9 Diet Cola 20.4 21.4  
## 10 Ford 166. 40.1  
## # ... with 11 more rows

plot(model)



pairs(table.b4)



tstat\_x1 <- coef(summary(model))[2,1]/coef(summary(model))[2,2]  
tstat\_x1

## [1] 1.868841

2\*(1-pt(tstat\_x1, 14))

## [1] 0.08271059

tstat\_x2 <- coef(summary(model))[3,1]/coef(summary(model))[3,2]  
tstat\_x2

## [1] 1.62789

2\*(1-pt(tstat\_x2, 14))

## [1] 0.1258361

tstat\_x3 <- coef(summary(model))[4,1]/coef(summary(model))[4,2]  
tstat\_x3

## [1] 0.3042043

2\*(1-pt(tstat\_x3, 14))

## [1] 0.7654469

tstat\_x4 <- coef(summary(model))[5,1]/coef(summary(model))[5,2]  
tstat\_x4

## [1] 0.6245612

2\*(1-pt(tstat\_x4, 14))

## [1] 0.5423043

tstat\_x5 <- coef(summary(model))[6,1]/coef(summary(model))[6,2]  
tstat\_x5

## [1] 1.460991

2\*(1-pt(tstat\_x5, 14))

## [1] 0.1660965

tstat\_x6 <- coef(summary(model))[7,1]/coef(summary(model))[7,2]  
tstat\_x6

## [1] -0.1724264

2\*(1-pt(tstat\_x6, 14))

## [1] 1.13443

tstat\_x7 <- coef(summary(model))[8,1]/coef(summary(model))[8,2]  
tstat\_x7

## [1] -0.4132581

2\*(1-pt(tstat\_x7, 14))

## [1] 1.314322

tstat\_x8 <- coef(summary(model))[9,1]/coef(summary(model))[9,2]  
tstat\_x8

## [1] -0.5567916

2\*(1-pt(tstat\_x8, 14))

## [1] 1.413539

tstat\_x9 <- coef(summary(model))[10,1]/coef(summary(model))[10,2]  
tstat\_x9

## [1] 0.8048774

2\*(1-pt(tstat\_x9, 14))

## [1] 0.4343472

coef(summary(model))[2,2]

anova(model)

## Analysis of Variance Table  
##   
## Response: y  
## Df Sum Sq Mean Sq F value Pr(>F)   
## x1 1 636.16 636.16 73.1525 6.238e-07 \*\*\*  
## x2 1 29.18 29.18 3.3551 0.08836 .   
## x3 1 4.71 4.71 0.5416 0.47391   
## x4 1 0.03 0.03 0.0032 0.95537   
## x5 1 8.78 8.78 1.0091 0.33216   
## x6 1 13.03 13.03 1.4982 0.24115   
## x7 1 9.14 9.14 1.0515 0.32254   
## x8 1 0.64 0.64 0.0741 0.78943   
## x9 1 5.63 5.63 0.6478 0.43435   
## Residuals 14 121.75 8.70   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model)

##   
## Call:  
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9,   
## data = table.b4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.720 -1.956 -0.045 1.627 4.253   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 14.92765 5.91285 2.525 0.0243 \*  
## x1 1.92472 1.02990 1.869 0.0827 .  
## x2 7.00053 4.30037 1.628 0.1258   
## x3 0.14918 0.49039 0.304 0.7654   
## x4 2.72281 4.35955 0.625 0.5423   
## x5 2.00668 1.37351 1.461 0.1661   
## x6 -0.41012 2.37854 -0.172 0.8656   
## x7 -1.40324 3.39554 -0.413 0.6857   
## x8 -0.03715 0.06672 -0.557 0.5865   
## x9 1.55945 1.93750 0.805 0.4343   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.949 on 14 degrees of freedom  
## Multiple R-squared: 0.8531, Adjusted R-squared: 0.7587   
## F-statistic: 9.037 on 9 and 14 DF, p-value: 0.000185

Lets check for multicollinearity

#install.packages("car")  
#library(car)  
  
#vif(model)

As a rule of thumb, a vif score over 5 is a problem. So we should consider dropping the problematic variable from the regression model.

Problem 3.15

FROM TABLE B15

data <- read\_excel("data-table-B-15.xls")  
head(data)

## # A tibble: 6 x 7  
## City MORT PRECIP EDUC NONWHITE NOX SO2  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 San Jose, CA 791. 13 12.2 3 32 3  
## 2 Wichita, KS 824. 28 12.1 7.5 2 1  
## 3 San Diego, CA 840. 10 12.1 5.9 66 20  
## 4 Lancaster, PA 844. 43 9.5 2.9 7 32  
## 5 Minneapolis, MN 858. 25 12.1 3 11 26  
## 6 Dallas, TX 860. 35 11.8 14.8 1 1

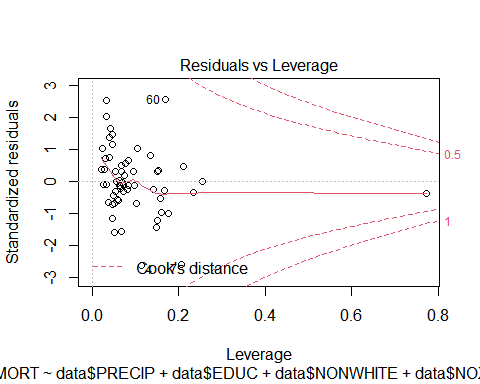
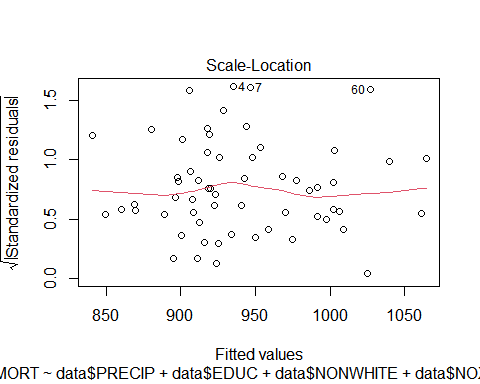
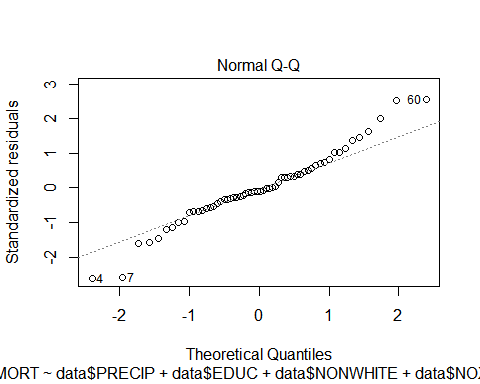
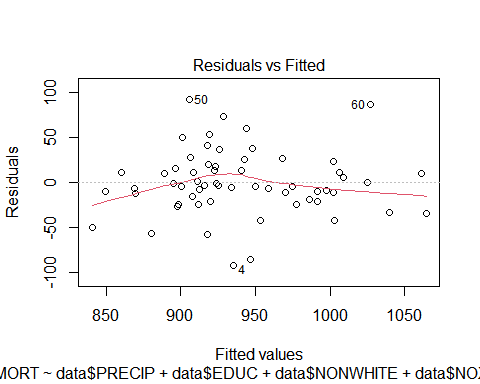
names(data)

## [1] "City" "MORT" "PRECIP" "EDUC" "NONWHITE" "NOX" "SO2"

model <- lm(data$MORT ~ data$PRECIP +   
 data$EDUC + data$NONWHITE +   
 data$NOX + data$SO2)  
summary(model)

##   
## Call:  
## lm(formula = data$MORT ~ data$PRECIP + data$EDUC + data$NONWHITE +   
## data$NOX + data$SO2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -91.38 -18.97 -3.56 16.00 91.83   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 995.63646 91.64099 10.865 3.35e-15 \*\*\*  
## data$PRECIP 1.40734 0.68914 2.042 0.046032 \*   
## data$EDUC -14.80139 7.02747 -2.106 0.039849 \*   
## data$NONWHITE 3.19909 0.62231 5.141 3.89e-06 \*\*\*  
## data$NOX -0.10797 0.13502 -0.800 0.427426   
## data$SO2 0.35518 0.09096 3.905 0.000264 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 37.09 on 54 degrees of freedom  
## Multiple R-squared: 0.6746, Adjusted R-squared: 0.6444   
## F-statistic: 22.39 on 5 and 54 DF, p-value: 4.407e-12

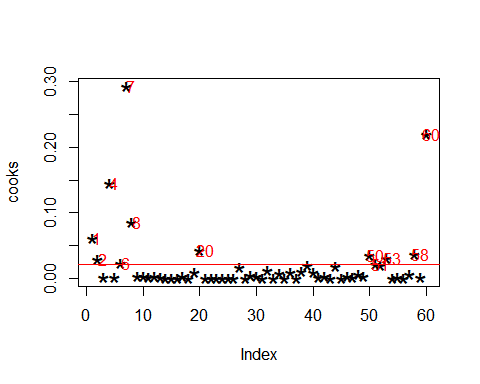
plot(model)



cooks <- cooks.distance(model)  
summary(cooks)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.300e-07 8.575e-04 3.546e-03 2.066e-02 1.720e-02 2.929e-01

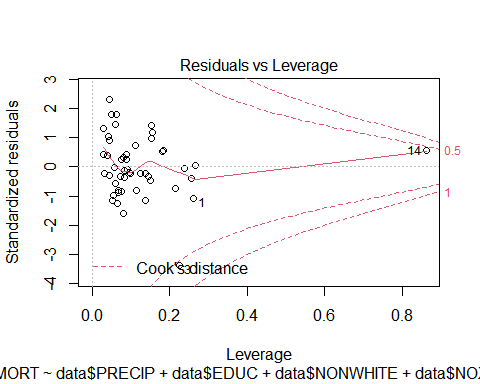
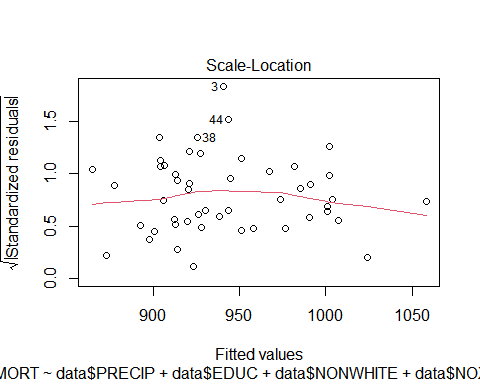
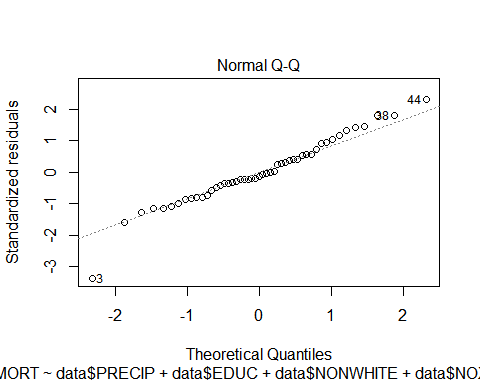
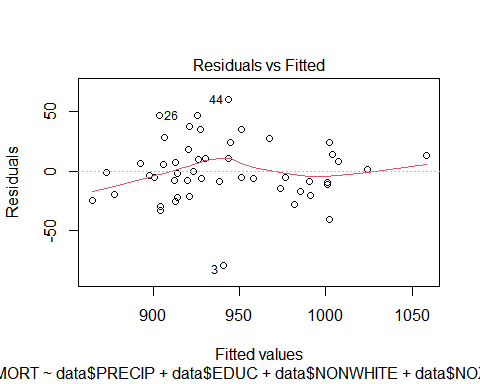
plot(cooks, pch="\*", cex=2)  
abline(h = mean(cooks, na.rm=T), col="red") # add cutoff line  
text(x=1:length(cooks)+1, y=cooks, labels=ifelse(cooks>mean(cooks, na.rm=T),names(cooks),""), col="red")



# Build new model on new data frame  
data <- data[ -c(1,2,6,8,20,50,51,53,58,60,4), ] #take out data points  
model2<- lm(data$MORT ~ data$PRECIP +   
 data$EDUC + data$NONWHITE +   
 data$NOX + data$SO2)  
summary(model2)

##   
## Call:  
## lm(formula = data$MORT ~ data$PRECIP + data$EDUC + data$NONWHITE +   
## data$NOX + data$SO2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -79.058 -14.829 -3.551 13.008 59.986   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1022.73634 75.13567 13.612 < 2e-16 \*\*\*  
## data$PRECIP 0.96424 0.54165 1.780 0.082112 .   
## data$EDUC -15.38738 6.01372 -2.559 0.014109 \*   
## data$NONWHITE 3.18595 0.57024 5.587 1.45e-06 \*\*\*  
## data$NOX -0.10688 0.19678 -0.543 0.589812   
## data$SO2 0.33071 0.07738 4.274 0.000105 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 26.71 on 43 degrees of freedom  
## Multiple R-squared: 0.7444, Adjusted R-squared: 0.7147   
## F-statistic: 25.04 on 5 and 43 DF, p-value: 9.83e-12

plot(model2)



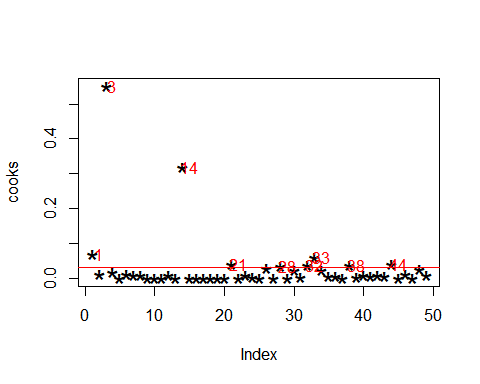
summary(model2)

##   
## Call:  
## lm(formula = data$MORT ~ data$PRECIP + data$EDUC + data$NONWHITE +   
## data$NOX + data$SO2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -79.058 -14.829 -3.551 13.008 59.986   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1022.73634 75.13567 13.612 < 2e-16 \*\*\*  
## data$PRECIP 0.96424 0.54165 1.780 0.082112 .   
## data$EDUC -15.38738 6.01372 -2.559 0.014109 \*   
## data$NONWHITE 3.18595 0.57024 5.587 1.45e-06 \*\*\*  
## data$NOX -0.10688 0.19678 -0.543 0.589812   
## data$SO2 0.33071 0.07738 4.274 0.000105 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 26.71 on 43 degrees of freedom  
## Multiple R-squared: 0.7444, Adjusted R-squared: 0.7147   
## F-statistic: 25.04 on 5 and 43 DF, p-value: 9.83e-12

cooks <- cooks.distance(model2)  
summary(cooks)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000018 0.0010455 0.0080081 0.0300762 0.0227189 0.5509157

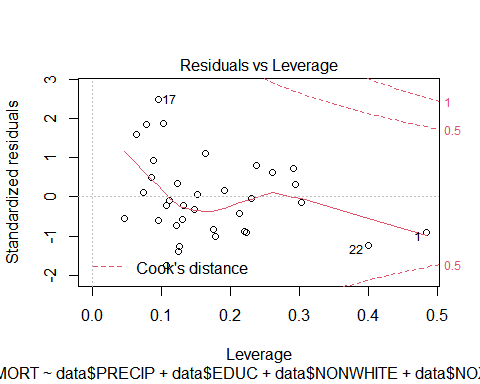
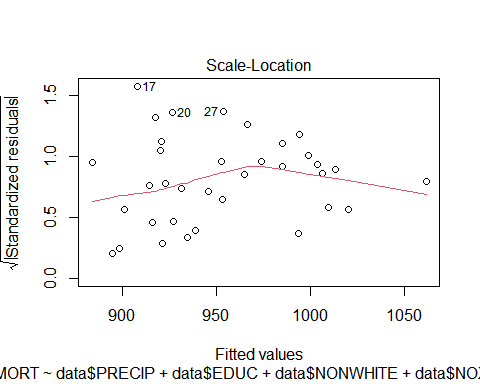
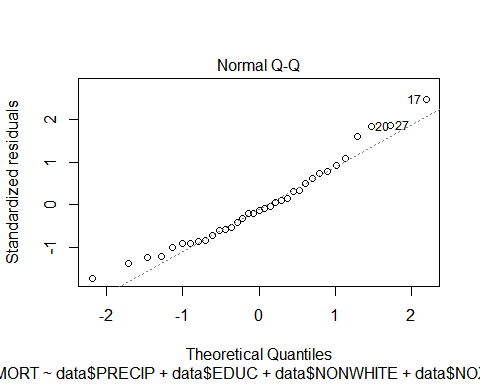
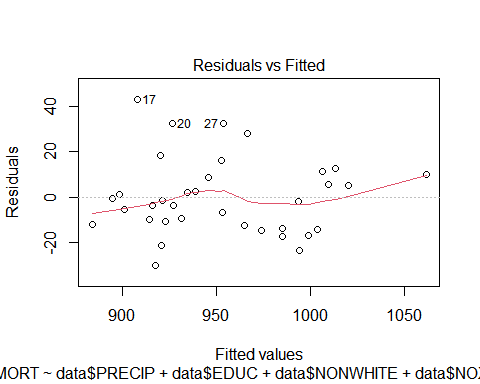
plot(cooks, pch="\*", cex=2)  
abline(h = mean(cooks, na.rm=T), col="red") # add cutoff line  
text(x=1:length(cooks)+1, y=cooks, labels=ifelse(cooks>mean(cooks, na.rm=T),names(cooks),""), col="red")



# Build new model on new data frame  
data <- data[ -c(1,2,6,8,20,50,51,53,58,60,4,21,28,  
 1,32,33,38,44,3,14), ] #take out data points  
model3<- lm(data$MORT ~ data$PRECIP +   
 data$EDUC + data$NONWHITE +   
 data$NOX + data$SO2)  
summary(model3)

##   
## Call:  
## lm(formula = data$MORT ~ data$PRECIP + data$EDUC + data$NONWHITE +   
## data$NOX + data$SO2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -29.798 -12.084 -2.046 9.309 42.791   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1015.8938 81.2521 12.503 3.32e-13 \*\*\*  
## data$PRECIP 0.6828 0.5720 1.194 0.2423   
## data$EDUC -13.5461 6.2980 -2.151 0.0400 \*   
## data$NONWHITE 3.5811 0.4447 8.053 7.01e-09 \*\*\*  
## data$NOX -0.7318 0.8955 -0.817 0.4205   
## data$SO2 0.4206 0.2014 2.089 0.0456 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.18 on 29 degrees of freedom  
## Multiple R-squared: 0.8684, Adjusted R-squared: 0.8457   
## F-statistic: 38.27 on 5 and 29 DF, p-value: 6.541e-12

plot(model3)



summary(model3)

##   
## Call:  
## lm(formula = data$MORT ~ data$PRECIP + data$EDUC + data$NONWHITE +   
## data$NOX + data$SO2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -29.798 -12.084 -2.046 9.309 42.791   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1015.8938 81.2521 12.503 3.32e-13 \*\*\*  
## data$PRECIP 0.6828 0.5720 1.194 0.2423   
## data$EDUC -13.5461 6.2980 -2.151 0.0400 \*   
## data$NONWHITE 3.5811 0.4447 8.053 7.01e-09 \*\*\*  
## data$NOX -0.7318 0.8955 -0.817 0.4205   
## data$SO2 0.4206 0.2014 2.089 0.0456 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.18 on 29 degrees of freedom  
## Multiple R-squared: 0.8684, Adjusted R-squared: 0.8457   
## F-statistic: 38.27 on 5 and 29 DF, p-value: 6.541e-12

Multiple R-squared: 0.8684 Therefore the model is performing a lot better

Lets calculate T-test for each regressor:

tstat\_precip <- coef(summary(model))[2,1]/coef(summary(model))[2,2]  
tstat\_precip

## [1] 2.04216

2\*(1-pt(tstat\_precip, 29))

## [1] 0.05032269

tstat\_educ <- coef(summary(model))[3,1]/coef(summary(model))[3,2]  
tstat\_educ

## [1] -2.10622

2\*(1-pt(tstat\_educ, 29))

## [1] 1.956043

tstat\_nonwhite <- coef(summary(model))[4,1]/coef(summary(model))[4,2]  
tstat\_nonwhite

## [1] 5.140683

2\*(1-pt(tstat\_nonwhite, 29))

## [1] 1.715609e-05

tstat\_nox <- coef(summary(model))[5,1]/coef(summary(model))[5,2]  
tstat\_nox

## [1] -0.7996351

2\*(1-pt(tstat\_nox, 29))

## [1] 1.569576

tstat\_so2 <- coef(summary(model))[6,1]/coef(summary(model))[6,2]  
tstat\_so2

## [1] 3.90474

2\*(1-pt(tstat\_so2, 29))

## [1] 0.000517846

# adjusted R²  
 summary(model)$adj.r.squared

## [1] 0.6444309

# R²  
 summary(model)$r.squared

## [1] 0.6745639

95% CI for model regressor “S02”

SO2\_CI <- confint(model, 'data$SO2', level=0.95)  
SO2\_CI

## 2.5 % 97.5 %  
## data$SO2 0.1728118 0.5375405

PROBLEM 4.5

library(readxl)  
data<- read\_excel("data-table-B4.XLS")  
head(data)

## # A tibble: 6 x 10  
## y x1 x2 x3 x4 x5 x6 x7 x8 x9  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 29.5 5.02 1 3.53 1.5 2 7 4 62 0  
## 2 27.9 4.54 1 2.28 1.18 1 6 3 40 0  
## 3 25.9 4.56 1 4.05 1.23 1 6 3 54 0  
## 4 29.9 5.06 1 4.46 1.12 1 6 3 42 0  
## 5 29.9 3.89 1 4.46 0.988 1 6 3 56 0  
## 6 30.9 5.90 1 5.85 1.24 1 7 3 51 1

#Lets check the data structure  
  
str(data)

## tibble [24 x 10] (S3: tbl\_df/tbl/data.frame)  
## $ y : num [1:24] 29.5 27.9 25.9 29.9 29.9 30.9 28.9 35.9 31.5 31 ...  
## $ x1: num [1:24] 5.02 4.54 4.56 5.06 3.89 ...  
## $ x2: num [1:24] 1 1 1 1 1 1 1 1 1 1 ...  
## $ x3: num [1:24] 3.53 2.27 4.05 4.46 4.46 ...  
## $ x4: num [1:24] 1.5 1.175 1.232 1.121 0.988 ...  
## $ x5: num [1:24] 2 1 1 1 1 1 0 2 1 1 ...  
## $ x6: num [1:24] 7 6 6 6 6 7 6 6 6 5 ...  
## $ x7: num [1:24] 4 3 3 3 3 3 3 3 3 2 ...  
## $ x8: num [1:24] 62 40 54 42 56 51 32 32 30 30 ...  
## $ x9: num [1:24] 0 0 0 0 0 1 0 0 0 0 ...

#lets fit the model   
  
model =lm(y~.,data=data)  
summary(model)

##   
## Call:  
## lm(formula = y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.720 -1.956 -0.045 1.627 4.253   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 14.92765 5.91285 2.525 0.0243 \*  
## x1 1.92472 1.02990 1.869 0.0827 .  
## x2 7.00053 4.30037 1.628 0.1258   
## x3 0.14918 0.49039 0.304 0.7654   
## x4 2.72281 4.35955 0.625 0.5423   
## x5 2.00668 1.37351 1.461 0.1661   
## x6 -0.41012 2.37854 -0.172 0.8656   
## x7 -1.40324 3.39554 -0.413 0.6857   
## x8 -0.03715 0.06672 -0.557 0.5865   
## x9 1.55945 1.93750 0.805 0.4343   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.949 on 14 degrees of freedom  
## Multiple R-squared: 0.8531, Adjusted R-squared: 0.7587   
## F-statistic: 9.037 on 9 and 14 DF, p-value: 0.000185

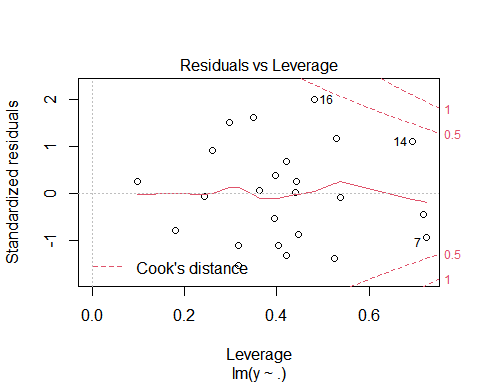
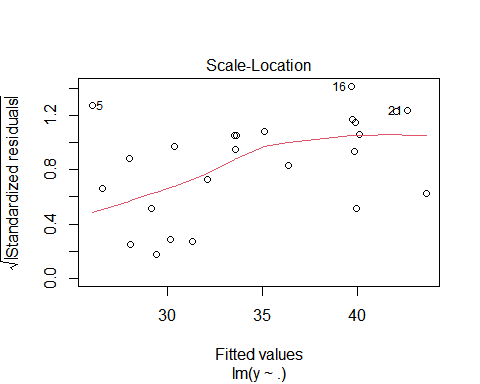
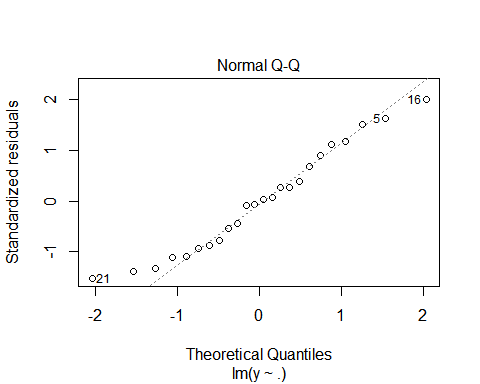
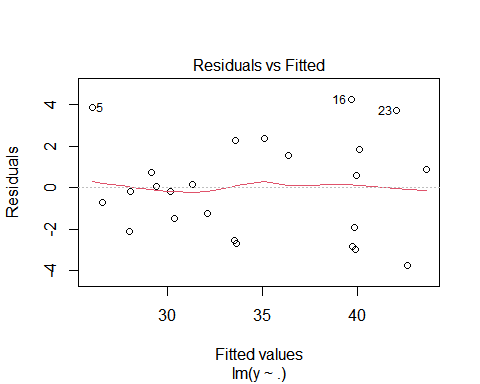
#Lets find the optimum regression equation with useful variables only  
model2 =step(lm(y~.,data=data))

## Start: AIC=58.97  
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9  
##   
## Df Sum of Sq RSS AIC  
## - x6 1 0.2585 122.01 57.025  
## - x3 1 0.8048 122.55 57.132  
## - x7 1 1.4852 123.23 57.265  
## - x8 1 2.6960 124.44 57.499  
## - x4 1 3.3922 125.14 57.633  
## - x9 1 5.6337 127.38 58.059  
## <none> 121.75 58.974  
## - x5 1 18.5622 140.31 60.379  
## - x2 1 23.0454 144.79 61.134  
## - x1 1 30.3724 152.12 62.319  
##   
## Step: AIC=57.02  
## y ~ x1 + x2 + x3 + x4 + x5 + x7 + x8 + x9  
##   
## Df Sum of Sq RSS AIC  
## - x3 1 0.889 122.90 55.199  
## - x8 1 2.731 124.74 55.556  
## - x4 1 3.312 125.32 55.667  
## - x9 1 5.889 127.90 56.156  
## - x7 1 9.249 131.26 56.778  
## <none> 122.01 57.025  
## - x5 1 18.798 140.81 58.464  
## - x2 1 26.774 148.78 59.786  
## - x1 1 40.508 162.51 61.905  
##   
## Step: AIC=55.2  
## y ~ x1 + x2 + x4 + x5 + x7 + x8 + x9  
##   
## Df Sum of Sq RSS AIC  
## - x8 1 3.245 126.14 53.824  
## - x4 1 4.744 127.64 54.108  
## - x9 1 8.718 131.61 54.844  
## - x7 1 9.501 132.40 54.986  
## <none> 122.90 55.199  
## - x5 1 17.987 140.88 56.477  
## - x2 1 25.930 148.83 57.793  
## - x1 1 53.969 176.87 61.936  
##   
## Step: AIC=53.82  
## y ~ x1 + x2 + x4 + x5 + x7 + x9  
##   
## Df Sum of Sq RSS AIC  
## - x4 1 4.935 131.07 52.745  
## - x9 1 5.839 131.98 52.910  
## <none> 126.14 53.824  
## - x5 1 19.015 145.16 55.194  
## - x7 1 22.338 148.48 55.737  
## - x2 1 24.770 150.91 56.127  
## - x1 1 95.846 221.99 65.390  
##   
## Step: AIC=52.75  
## y ~ x1 + x2 + x5 + x7 + x9  
##   
## Df Sum of Sq RSS AIC  
## - x9 1 5.540 136.62 51.739  
## <none> 131.07 52.745  
## - x5 1 16.852 147.93 53.648  
## - x7 1 17.471 148.55 53.748  
## - x2 1 39.893 170.97 57.122  
## - x1 1 158.792 289.87 69.793  
##   
## Step: AIC=51.74  
## y ~ x1 + x2 + x5 + x7  
##   
## Df Sum of Sq RSS AIC  
## <none> 136.62 51.739  
## - x5 1 19.643 156.26 52.963  
## - x7 1 21.024 157.64 53.174  
## - x2 1 47.648 184.26 56.919  
## - x1 1 157.504 294.12 68.142

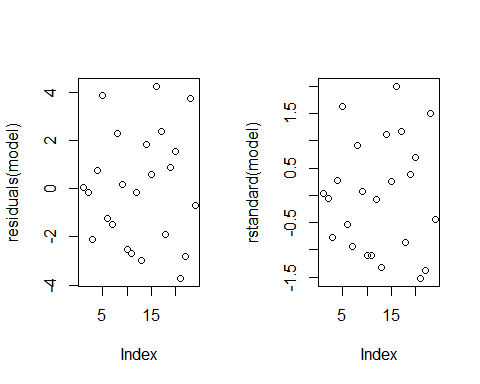
summary(model)

##   
## Call:  
## lm(formula = y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.720 -1.956 -0.045 1.627 4.253   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 14.92765 5.91285 2.525 0.0243 \*  
## x1 1.92472 1.02990 1.869 0.0827 .  
## x2 7.00053 4.30037 1.628 0.1258   
## x3 0.14918 0.49039 0.304 0.7654   
## x4 2.72281 4.35955 0.625 0.5423   
## x5 2.00668 1.37351 1.461 0.1661   
## x6 -0.41012 2.37854 -0.172 0.8656   
## x7 -1.40324 3.39554 -0.413 0.6857   
## x8 -0.03715 0.06672 -0.557 0.5865   
## x9 1.55945 1.93750 0.805 0.4343   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.949 on 14 degrees of freedom  
## Multiple R-squared: 0.8531, Adjusted R-squared: 0.7587   
## F-statistic: 9.037 on 9 and 14 DF, p-value: 0.000185

SSE\_Model=sum((model2$residuals)^2) #SSE of the optimal model  
  
plot(model)



par(mfrow = c(1,2))  
plot(residuals(model))  
plot(rstandard(model))



model <- lm(y~x1+sqrt(x2)+log(x3)+log(x4)+sqrt(x5)+log(x6)  
 +sqrt(x7)+log(x8)+sqrt(x9), data)  
summary(model)

##   
## Call:  
## lm(formula = y ~ x1 + sqrt(x2) + log(x3) + log(x4) + sqrt(x5) +   
## log(x6) + sqrt(x7) + log(x8) + sqrt(x9), data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.9846 -1.8454 0.2371 1.2531 3.7333   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 14.0197 16.3996 0.855 0.4070   
## x1 2.0550 0.9330 2.203 0.0449 \*  
## sqrt(x2) 13.2104 9.1648 1.441 0.1715   
## log(x3) 2.0664 2.7153 0.761 0.4593   
## log(x4) 4.1810 5.5492 0.753 0.4637   
## sqrt(x5) 3.5415 2.0793 1.703 0.1106   
## log(x6) -3.3610 15.8183 -0.212 0.8348   
## sqrt(x7) -5.7511 12.3059 -0.467 0.6474   
## log(x8) 0.1871 1.3797 0.136 0.8941   
## sqrt(x9) 1.2843 1.8772 0.684 0.5050   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.921 on 14 degrees of freedom  
## Multiple R-squared: 0.8559, Adjusted R-squared: 0.7633   
## F-statistic: 9.24 on 9 and 14 DF, p-value: 0.0001635

Residual standard error: 0.06682 Multiple R-squared: 0.9761 The model is performing a lot better now after the box cox transformation

PROBLEM 5.10 \*\*\*\* TABLE B9 \*\*\*\*

data <- read\_excel("data-table-B9.XLS")  
  
head(data)

## # A tibble: 6 x 5  
## x1 x2 x3 x4 y  
## <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2.14 10 0.34 1 28.9  
## 2 4.14 10 0.34 1 31   
## 3 8.15 10 0.34 1 26.4  
## 4 2.14 10 0.34 0.246 27.2  
## 5 4.14 10 0.34 0.379 26.1  
## 6 8.15 10 0.34 0.474 23.2

Lets check the regression model

attach(data)

## The following objects are masked from table.b4:  
##   
## x1, x2, x3, x4, y

model=lm(y ~ ., data = data) # regression model  
summary(model)

##   
## Call:  
## lm(formula = y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.9958 -3.3092 -0.2419 3.3924 10.5668   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.89453 4.32508 1.363 0.17828   
## x1 -0.47790 0.34002 -1.406 0.16530   
## x2 0.18271 0.01718 10.633 3.78e-15 \*\*\*  
## x3 35.40284 11.09960 3.190 0.00232 \*\*   
## x4 5.84391 2.90978 2.008 0.04935 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.014 on 57 degrees of freedom  
## Multiple R-squared: 0.6914, Adjusted R-squared: 0.6697   
## F-statistic: 31.92 on 4 and 57 DF, p-value: 5.818e-14

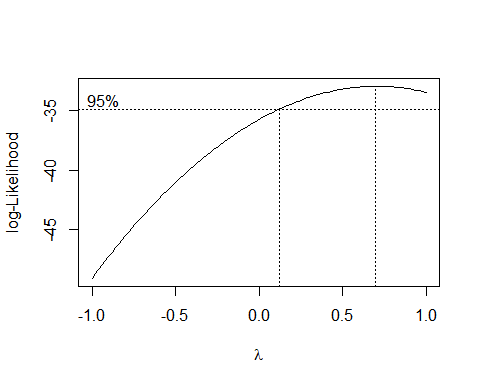
Lets perform a box cox

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:MPV':  
##   
## cement

bc=boxcox(model, lambda=seq(-1.0, 1.0, .1), plotit=TRUE) # Box cox

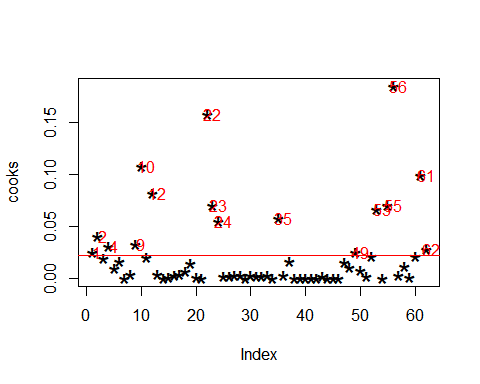


Cooks distance for outliers analysis

cooks <- cooks.distance(model)  
summary(cooks)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 6.040e-06 1.167e-03 4.515e-03 2.255e-02 2.464e-02 1.853e-01

plot(cooks, pch="\*", cex=2)  
abline(h = mean(cooks, na.rm=T), col="red") # add cutoff line  
text(x=1:length(cooks)+1, y=cooks, labels=ifelse(cooks>mean(cooks, na.rm=T),names(cooks),""), col="red")

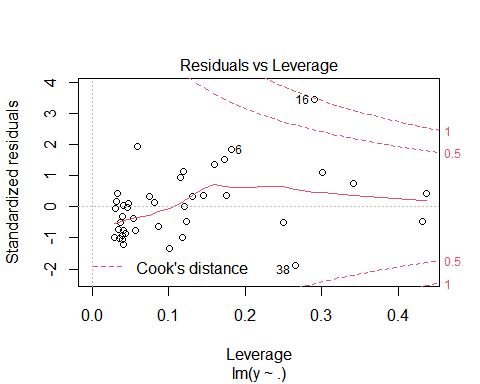
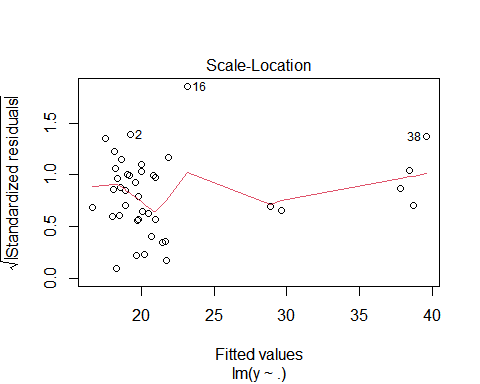
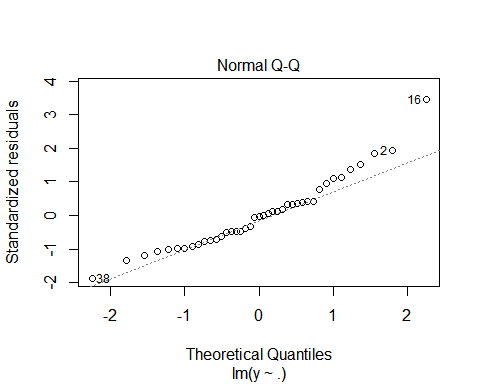
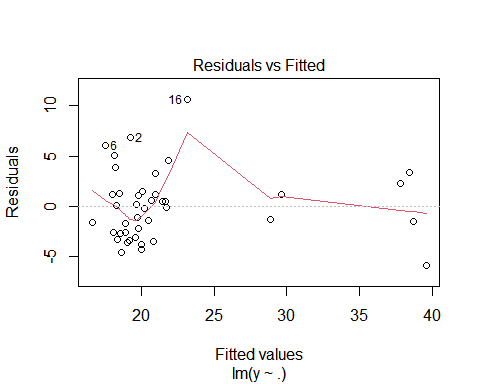


#take out data points

# Build new model on new data frame  
data <- data[ -c(1,2,4,9,10,12,23,24,25,35,49,48,47,60,52,37,54,53,61,62,56), ]  
model2<- lm(y ~ ., data = data)  
summary(model2)

##   
## Call:  
## lm(formula = y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.8827 -2.6497 -0.1083 1.2446 10.6285   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.51054 4.33125 0.811 0.42297   
## x1 -0.43334 0.35257 -1.229 0.22701   
## x2 0.17050 0.01829 9.323 3.91e-11 \*\*\*  
## x3 38.78465 11.45507 3.386 0.00173 \*\*   
## x4 6.95314 3.20346 2.171 0.03664 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.659 on 36 degrees of freedom  
## Multiple R-squared: 0.7543, Adjusted R-squared: 0.727   
## F-statistic: 27.64 on 4 and 36 DF, p-value: 1.548e-10

plot(model2)

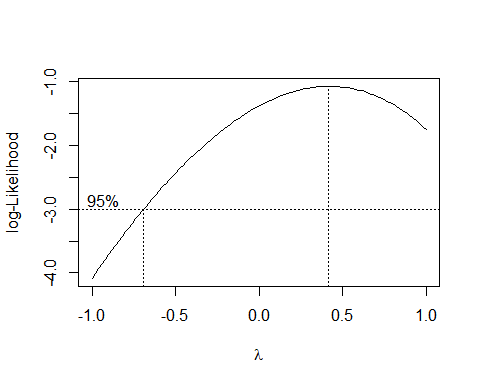


summary(model2)

##   
## Call:  
## lm(formula = y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.8827 -2.6497 -0.1083 1.2446 10.6285   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.51054 4.33125 0.811 0.42297   
## x1 -0.43334 0.35257 -1.229 0.22701   
## x2 0.17050 0.01829 9.323 3.91e-11 \*\*\*  
## x3 38.78465 11.45507 3.386 0.00173 \*\*   
## x4 6.95314 3.20346 2.171 0.03664 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.659 on 36 degrees of freedom  
## Multiple R-squared: 0.7543, Adjusted R-squared: 0.727   
## F-statistic: 27.64 on 4 and 36 DF, p-value: 1.548e-10

LETS PERFORM ANOTHER BOX COX ON NEW SUBSET

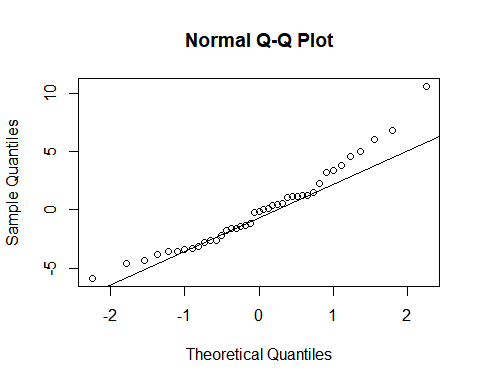
bc=boxcox(model2, lambda=seq(-1.0, 1.0, .1), plotit=TRUE) # Box cox



Model is performing a lot better

Lets check for residuals

qqnorm(model2$residuals)  
qqline(model2$residuals)



Problem 5.21

mix\_Rate <- c(150, 175 ,200, 225 ,3129, 3200, 2800, 2600)  
Tensile\_Strength <- c(3000, 3065, 3300, 2975, 2900, 2985 ,2700 ,2600 ,3190 ,3150 ,3050 ,2765)  
Tensile\_Strength

## [1] 3000 3065 3300 2975 2900 2985 2700 2600 3190 3150 3050 2765

A=c(3129,3000,3065,3190)  
B=c(3200,3300,2975,3150)  
C=c(2800,2900,2985,3050)  
D=c(2600,2700,2600,2765)  
  
Dt=matrix(c(A,B,C,D),ncol=4,byrow=FALSE)  
Dt

## [,1] [,2] [,3] [,4]  
## [1,] 3129 3200 2800 2600  
## [2,] 3000 3300 2900 2700  
## [3,] 3065 2975 2985 2600  
## [4,] 3190 3150 3050 2765

a=4 # Number of treatments in experiment  
  
Yi.=colSums(Dt)  
Y..=sum(Dt)  
N=length(A)+length(B)+length(C)+length(D) # Total number of observations.  
CT=Y..^2/N # Correction Term  
  
TSS=sum(Dt^2)-CT # Total Sum of square  
TSS

## [1] 706210.9

SSTr=(1/4)\*sum(Yi.^2)-CT # Sum of square due to treatment  
SSTr

## [1] 575802.7

MSTr=SSTr/(a-1) # Mean Sum of square due to treatment  
  
SSE=TSS-SSTr # Sum of square due to error  
SSE

## [1] 130408.2

MSSE=SSE/(N-a) # Mean Sum of square due to error  
  
Fcal=MSTr/MSSE # F-statistic value  
Fcal

## [1] 17.66154

Ftab=qf(0.95,a-1,N-a) # F-tabulated value can chosen from table with appropriate degrees pf freedem  
Ftab

## [1] 3.490295

Problem(Polynomial Regression)

[ Dataset 7.18]

data <- read\_excel("data-prob7-18.xlsx")  
data

## # A tibble: 26 x 4  
## y x1 x2 x3  
## <dbl> <dbl> <dbl> <dbl>  
## 1 0.222 7.3 0 0   
## 2 0.395 8.7 0 0.3  
## 3 0.422 8.8 0.7 1   
## 4 0.437 8.1 4 0.2  
## 5 0.428 9 0.5 1   
## 6 0.467 8.7 1.5 2.8  
## 7 0.444 9.3 2.1 1   
## 8 0.378 7.6 5.1 3.4  
## 9 0.494 10 0 0.3  
## 10 0.456 8.4 3.7 4.1  
## # ... with 16 more rows

attach(data)

## The following objects are masked from data (pos = 4):  
##   
## x1, x2, x3, y

## The following objects are masked from table.b4:  
##   
## x1, x2, x3, y

Lets perform a full quadratic regression

model = lm(y ~ x1+x2+x3+x1^2+x2^2+x3^2, data=data)  
summary(model)

##   
## Call:  
## lm(formula = y ~ x1 + x2 + x3 + x1^2 + x2^2 + x3^2, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.09188 -0.04688 -0.01395 0.03138 0.12708   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.369350 0.143675 -2.571 0.0174 \*   
## x1 0.086514 0.016006 5.405 1.99e-05 \*\*\*  
## x2 0.024417 0.010228 2.387 0.0260 \*   
## x3 -0.028581 0.004478 -6.382 2.02e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.06583 on 22 degrees of freedom  
## Multiple R-squared: 0.8667, Adjusted R-squared: 0.8485   
## F-statistic: 47.68 on 3 and 22 DF, p-value: 8.555e-10

T test for x1 x2 and x3

tstat\_x1 <- coef(summary(model))[2,1]/coef(summary(model))[2,2]  
tstat\_x1

## [1] 5.405082

2\*(1-pt(tstat\_x1, 22))

## [1] 1.987083e-05

tstat\_x2 <- coef(summary(model))[3,1]/coef(summary(model))[3,2]  
tstat\_x2

## [1] 2.387331

2\*(1-pt(tstat\_x2, 22))

## [1] 0.02599389

tstat\_x3 <- coef(summary(model))[4,1]/coef(summary(model))[4,2]  
tstat\_x3

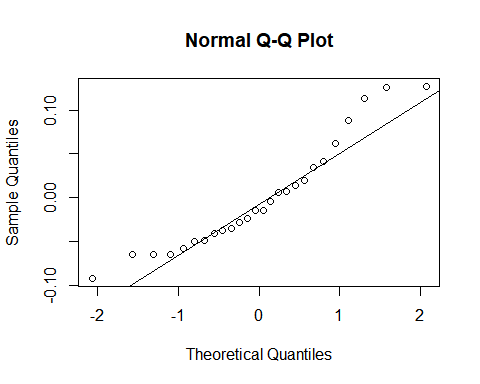
## [1] -6.382038

2\*(1-pt(tstat\_x3, 22))

## [1] 1.999998

lets plot residuals

qqnorm(model$residuals)  
qqline(model$residuals)



library(rsm)

## Warning: package 'rsm' was built under R version 4.1.3

## lets convert to integer as RSM function only takes integers  
as.integer(data$x1)

## [1] 7 8 8 8 9 8 9 7 10 8 9 7 9 7 8 9 7 7 7 10 7 7 7 7 7  
## [26] 7

as.integer(data$x2)

## [1] 0 0 0 4 0 1 2 5 0 3 3 2 4 2 2 2 2 2 3 1 3 3 4 6 2 7

as.integer(data$x3)

## [1] 0 0 1 0 1 2 1 3 0 4 2 7 2 6 6 5 7 7 8 4 8 6 9 10 5  
## [26] 20

as.integer(data$y)

## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

tools.rsm <- rsm(y ~ SO(x1,x2,x3), data = data)  
summary(tools.rsm)

##   
## Call:  
## rsm(formula = y ~ SO(x1, x2, x3), data = data)  
##   
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.76846225 1.28727791 -1.3738 0.18845   
## x1 0.42059352 0.29424173 1.4294 0.17212   
## x2 0.22244974 0.13077280 1.7010 0.10829   
## x3 -0.12801311 0.07026172 -1.8219 0.08721 .  
## x1:x2 -0.01987630 0.01204027 -1.6508 0.11826   
## x1:x3 0.00915226 0.00762307 1.2006 0.24738   
## x2:x3 0.00257867 0.00704092 0.3662 0.71898   
## x1^2 -0.01931284 0.01680075 -1.1495 0.26723   
## x2^2 -0.00744859 0.01205056 -0.6181 0.54520   
## x3^2 0.00082358 0.00144144 0.5714 0.57570   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Multiple R-squared: 0.9169, Adjusted R-squared: 0.8702   
## F-statistic: 19.62 on 9 and 16 DF, p-value: 5.062e-07  
##   
## Analysis of Variance Table  
##   
## Response: y  
## Df Sum Sq Mean Sq F value Pr(>F)  
## FO(x1, x2, x3) 3 0.61987 0.206622 55.6424 1.098e-08  
## TWI(x1, x2, x3) 3 0.02386 0.007954 2.1421 0.1350  
## PQ(x1, x2, x3) 3 0.01206 0.004019 1.0822 0.3848  
## Residuals 16 0.05941 0.003713   
## Lack of fit 16 0.05941 0.003713 NaN NaN  
## Pure error 0 0.00000 NaN   
##   
## Stationary point of response surface:  
## x1 x2 x3   
## 13.290212 -1.675598 6.495148   
##   
## Eigenanalysis:  
## eigen() decomposition  
## $values  
## [1] 0.001959153 -0.002139059 -0.025757953  
##   
## $vectors  
## [,1] [,2] [,3]  
## x1 0.2818074 0.4238920 0.8607556  
## x2 -0.1682371 -0.8613854 0.4792822  
## x3 0.9446062 -0.2798762 -0.1714304

canonical(tools.rsm)

## Near-stationary-ridge situation detected -- stationary point altered  
## Change 'threshold' if this is not what you intend

## $xs  
## x1 x2 x3   
## 8.197039 4.564240 -1.632544   
##   
## $eigen  
## eigen() decomposition  
## $values  
## [1] 0.00000000 0.00000000 -0.02575795  
##   
## $vectors  
## [,1] [,2] [,3]  
## x1 0.2818074 0.4238920 0.8607556  
## x2 -0.1682371 -0.8613854 0.4792822  
## x3 0.9446062 -0.2798762 -0.1714304

xs <- canonical(tools.rsm)$xs

## Near-stationary-ridge situation detected -- stationary point altered  
## Change 'threshold' if this is not what you intend

contour(tools.rsm, ~ x1+x2+x3, image=TRUE, at=xs)

