Lab 3

Ve406

Due: **24 November 2020, 18:20am**

Instructions

- This lab is about unusual points, heteroskedasticity and correlated errors.

Task 1 (8 points)

The data `chem_pro` is the dataset about a particular chemical process we considered in class.

(a) (1 point)

Succesfully render this file.

(b) (1 point)

Clean `chem_pro.df` according to what we have discussed in class.

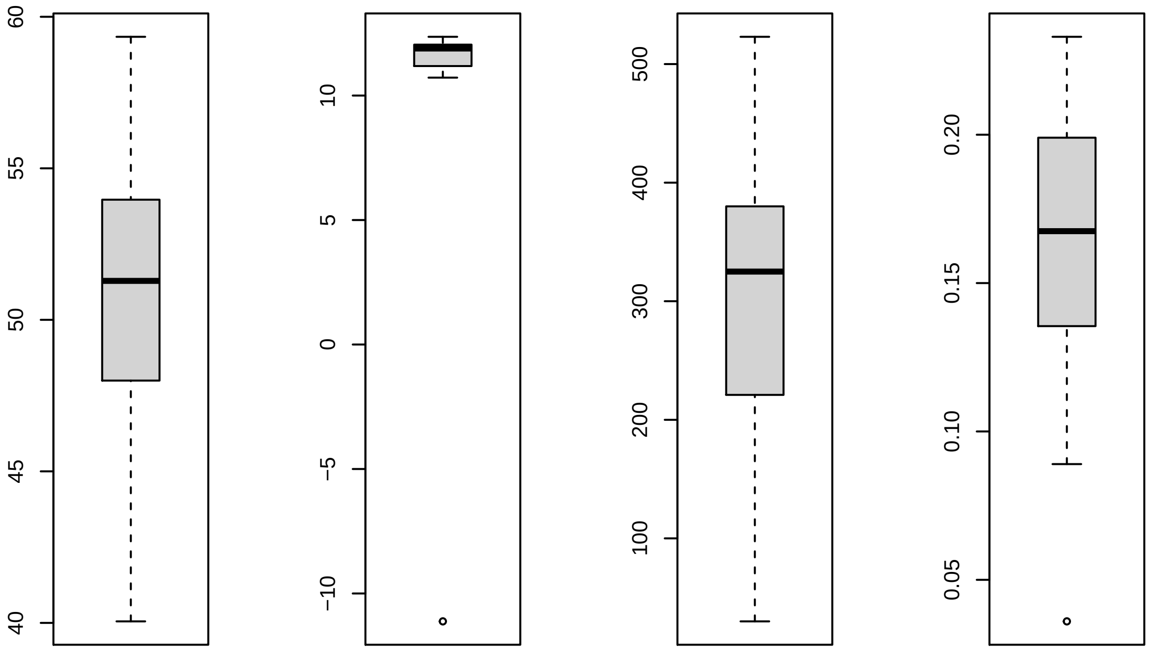
Data cleaning is discussed as "correcting obvious typos and reporting potential errors". Inconsistent data type indicates typo of *ratio*. Potential error is found through summary and boxplot.

```
chem_pro.df = read.table(file = chem_pro.csv, sep = ",", header = TRUE)

# typo of inconsistent data type and '0>163'
ratio_typo = which(chem_pro.df$ratio == "0>163")
```

```
chem_pro.df$ratio[ratio_typo] = "0.163"
chem_pro.df$ratio = as.double(chem_pro.df$ratio)

# identify potential problems
# summary(chem_pro.df)
par(mfrow = c(1, 4))
lapply(chem_pro.df, boxplot)
```




```r
par(mfrow = c(1, 1))
conversion_typo = which(chem_pro.df$conversion <= -10)
ratio_unusual = which(chem_pro.df$ratio <= 0.05)

# conversion typo
chem_pro.df$conversion[conversion_typo] = -chem_pro.df$conversion[conversion_typo]
```

The conversion should not be negative, which should be a typo. The unusual point of ratio, index is 6, should be reported.

(c) (1 point)

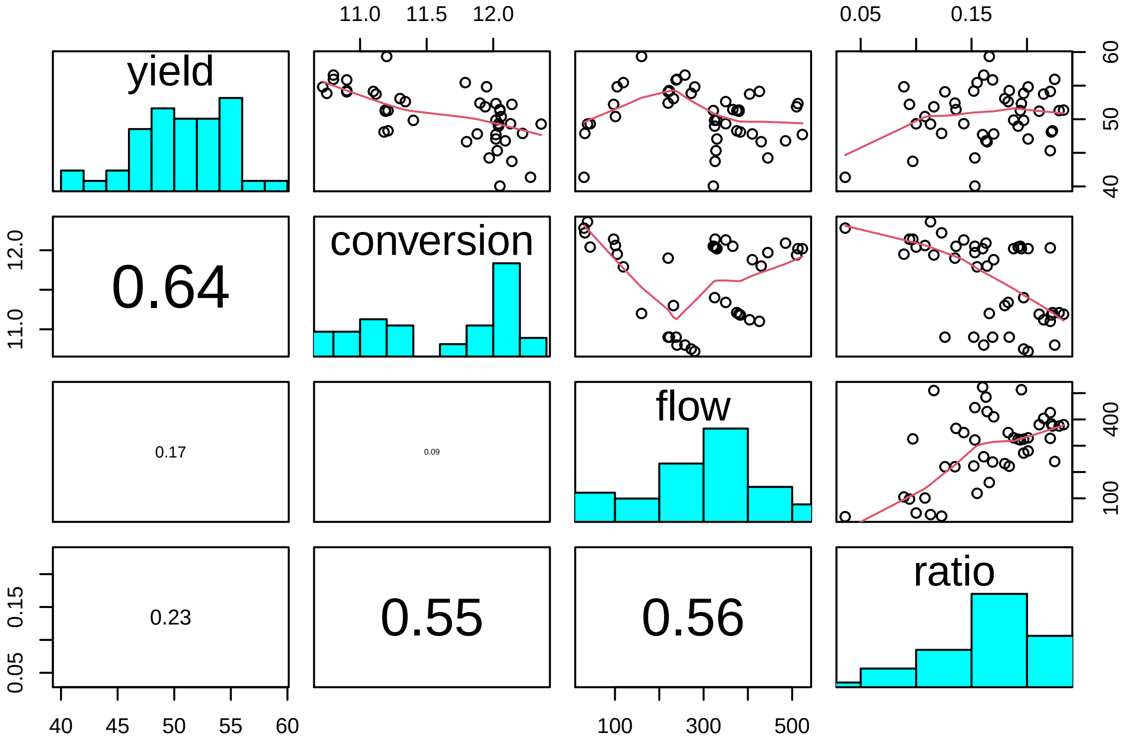
Produce the pairs plot of all the variables in `chem_pro.df` like the one I showed in class.

```r
## put histograms on the diagonal, from R official pairs doc
panel.hist <- function(x, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

## put (absolute) correlations, from R official pairs doc
## with size proportional to the correlations.
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
```

```
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

pairs(chem_pro.df, upper.panel = panel.smooth, diag.panel = panel.hist, lower.panel = panel.cor)
```



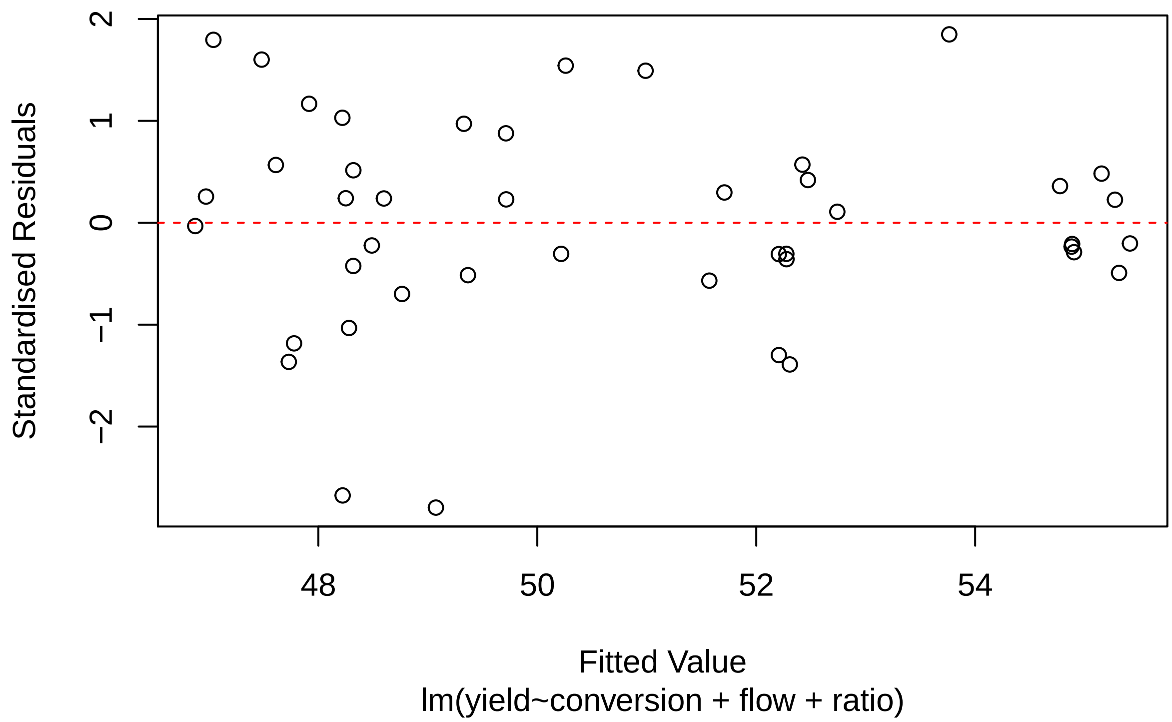
(d) (1 point)

Construct the following model, then produce all the usual regression diagnostic plots for `chem_pro.LM`.

```
chem_pro.LM = lm(yield~conversion+flow+ratio, data = chem_pro.df)
```

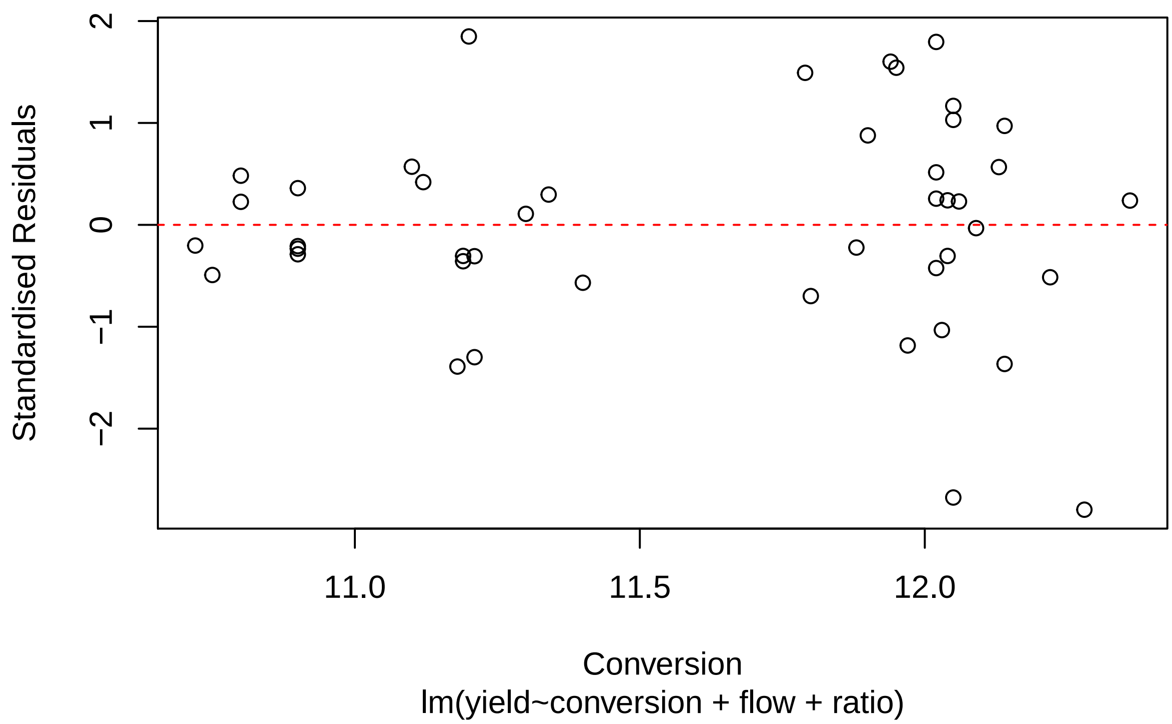
- Standardised residual Vs fitted value

```
fvs = fitted.values(chem_pro.LM)  # fitted value
sres = rstandard(chem_pro.LM)  # standardised residuals
plot(fvs, sres, xlab = "Fitted Value", ylab = "Standardised Residuals", sub = "lm(yield~conversion + fl
abline(h = 0, lty = 2, col = "red")
```



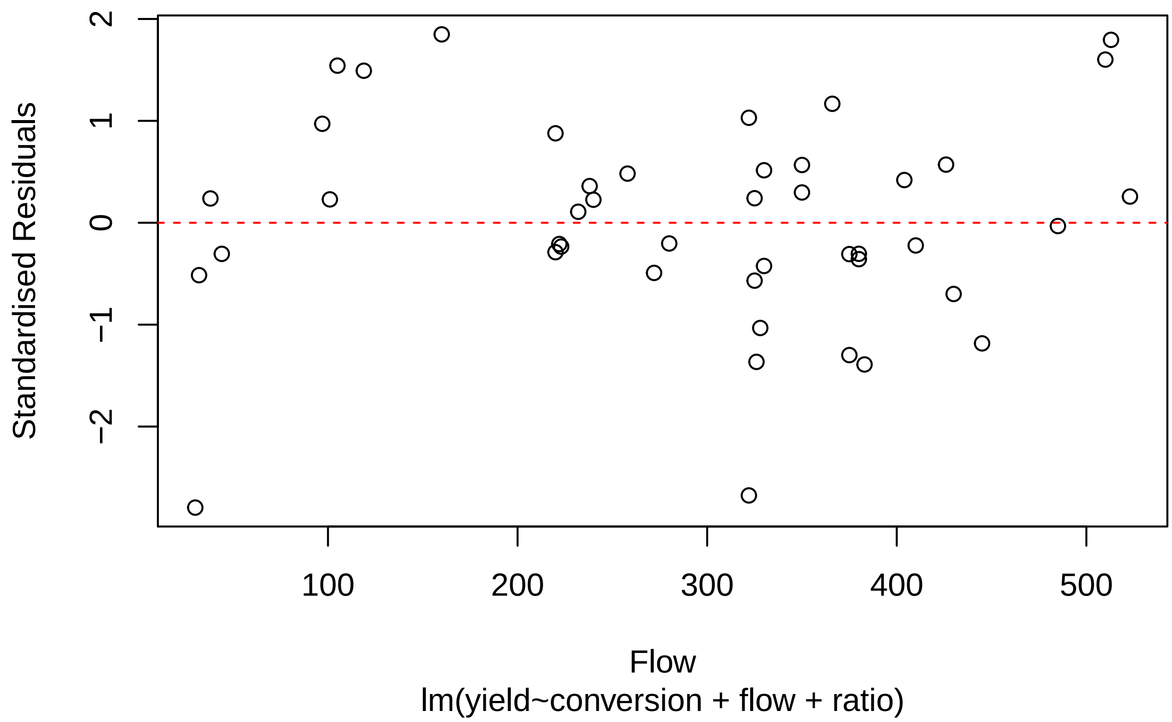
Standardised residual Vs conversion

```
plot(chem_pro.df$conversion, sres, xlab = "Conversion", ylab = "Standardised Residuals",  sub = "lm(yie
abline(h = 0, lty = 2, col = "red")
```



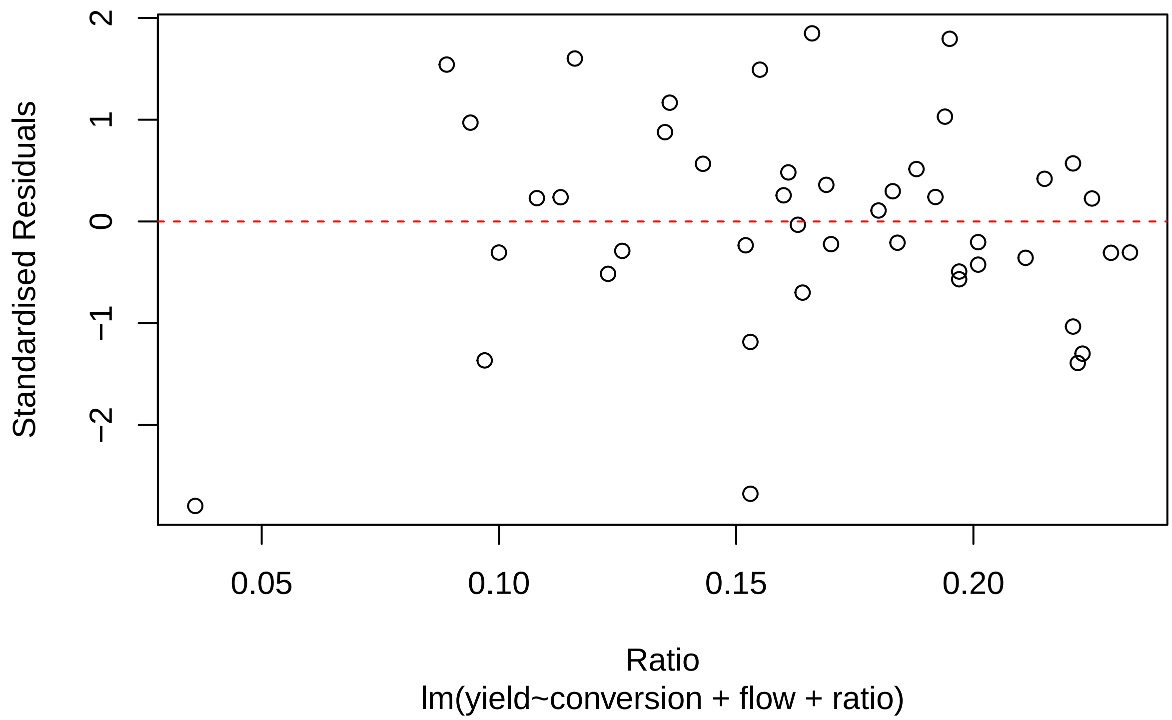
- Standardised residual Vs flow

```
plot(chem_pro.df$flow, sres, xlab = "Flow", ylab = "Standardised Residuals", sub = "lm(yield~conversion
abline(h = 0, lty = 2, col = "red")
```



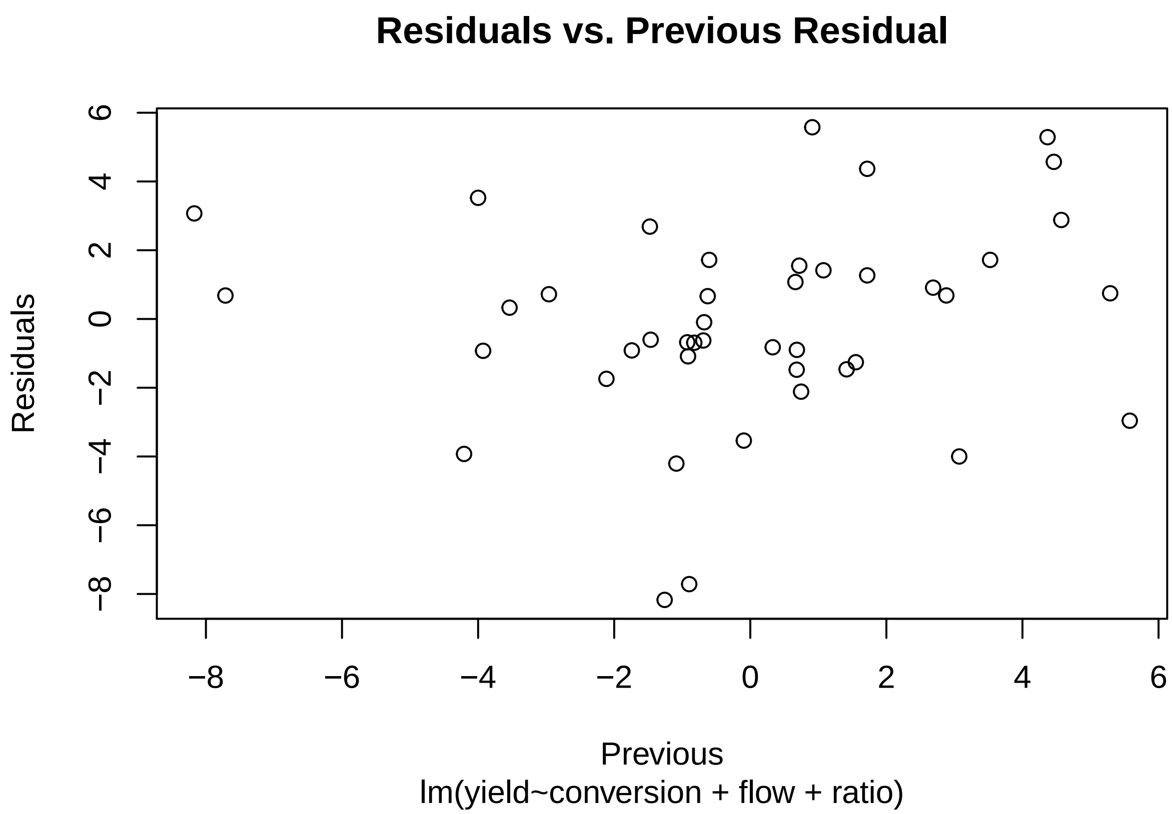
- Standardised residual Vs ratio

```
plot(chem_pro.df$ratio, sres, xlab = "Ratio", ylab = "Standardised Residuals", sub = "lm(yield~conversi
abline(h = 0, lty = 2, col = "red")
```



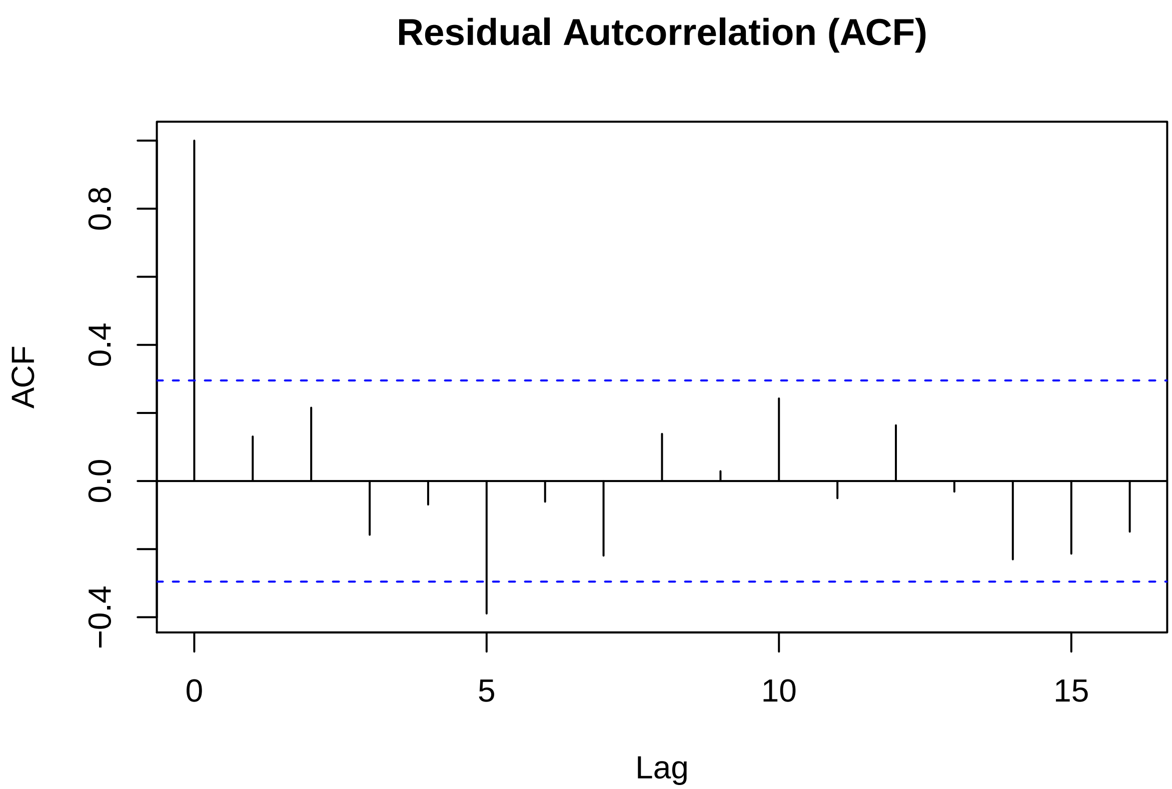
- Residual Vs Previous Residual

```
res = residuals(chem_pro.LM)
plot(res[-nrow(chem_pro.df)], res[-1], xlab = "Previous", ylab = "Residuals", main = "Residuals vs. Pre
```



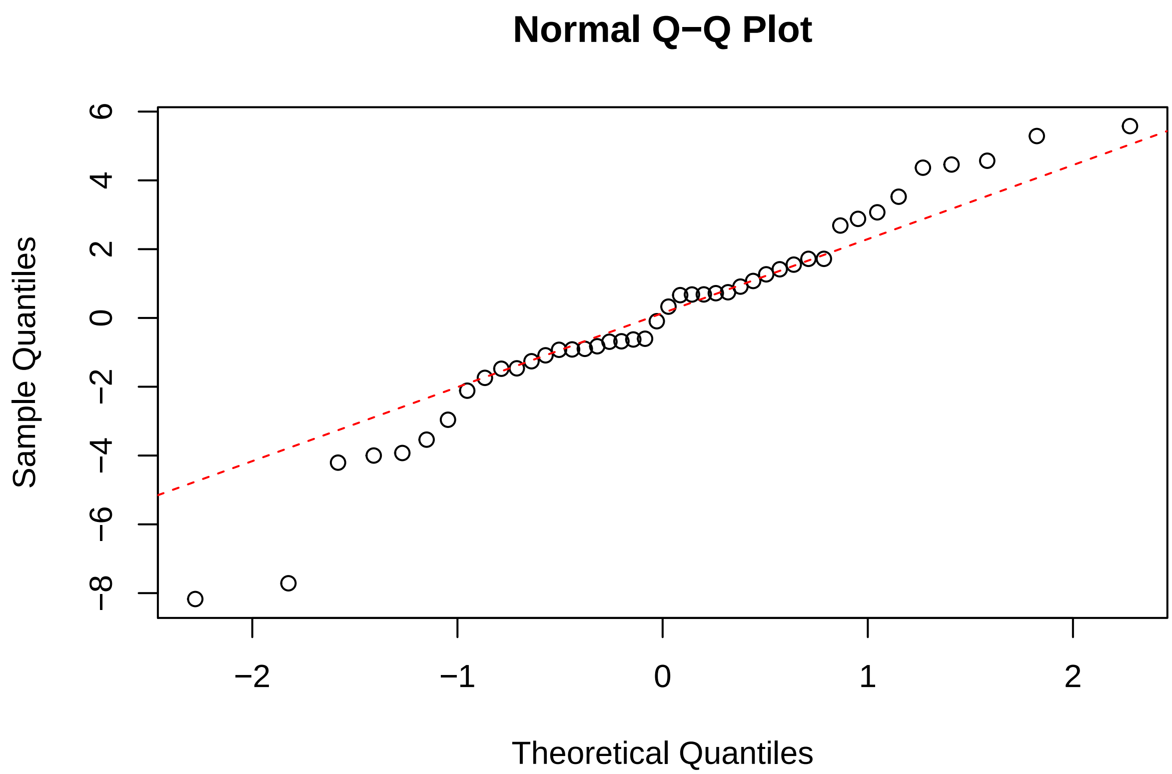
- Residual Autcorrelation (ACF)

```r
acf(res, main = "Residual Autcorrelation (ACF)")
```



- Q-Q Normal

```
qqnorm(res)
qqline(res, lty = 2, col = "red")
```



(e) (1 point)

Compute VIF for `chem_pro.LM` according to the definition, then compare it with the values found in class.

```
VIF <- rep(0, 3)
names(VIF) <- c("conversion", "flow", "ratio")
conversion.LM = lm(conversion ~ flow + ratio, data = chem_pro.df)
VIF[1] <- 1 / (1 - summary(conversion.LM)$r.squared)
flow.LM = lm(flow ~ conversion + ratio, data = chem_pro.df)
VIF[2] <- 1 / (1 - summary(flow.LM)$r.squared)
ratio.LM = lm(ratio ~ conversion + flow, data = chem_pro.df)
VIF[3] <- 1 / (1 - summary(ratio.LM)$r.squared)
VIF
```

```
## conversion       flow      ratio
##   1.580323   1.606860   2.276409
```

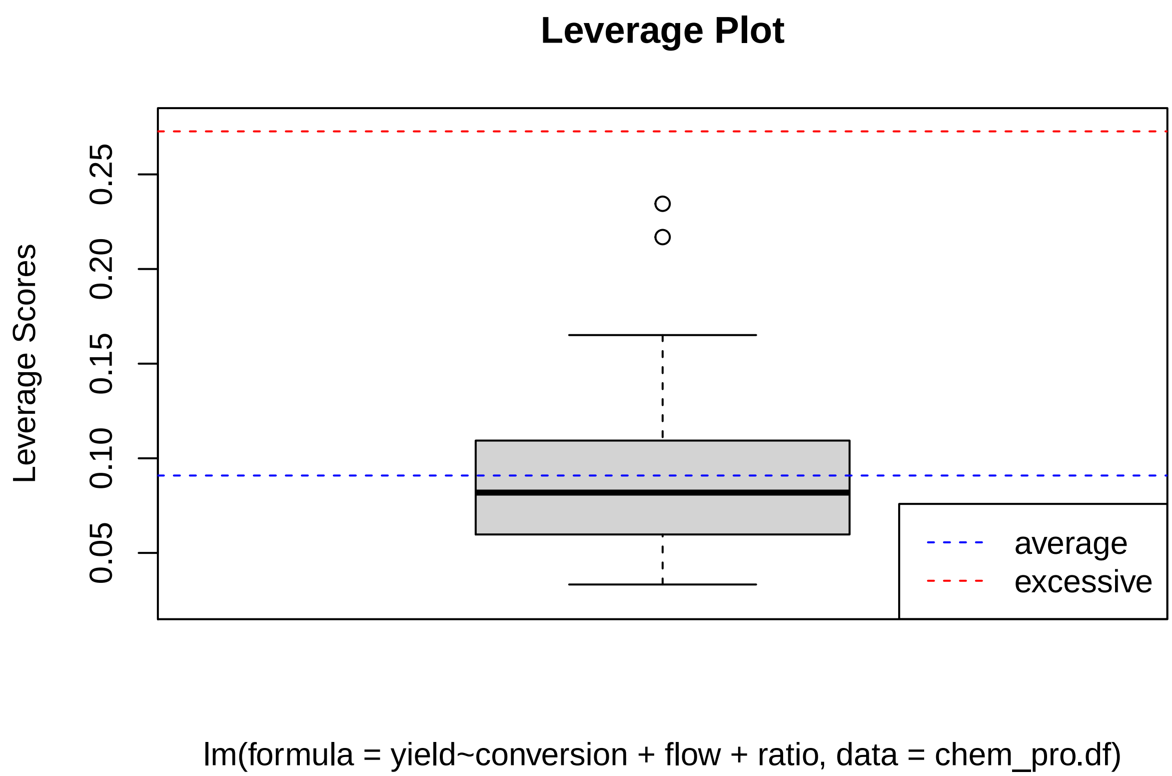
The VIF computed by definition gives the same result as in class.

(f) (1 point)

Produce a boxplot of Leverage Scores for `chem_pro.LM` like the one I showed in class.

```
pii.vec = hatvalues(chem_pro.LM)
boxplot(pii.vec, ylim = c(0.025, 0.275), xlab = "lm(formula = yield~conversion + flow + ratio, data = c
```

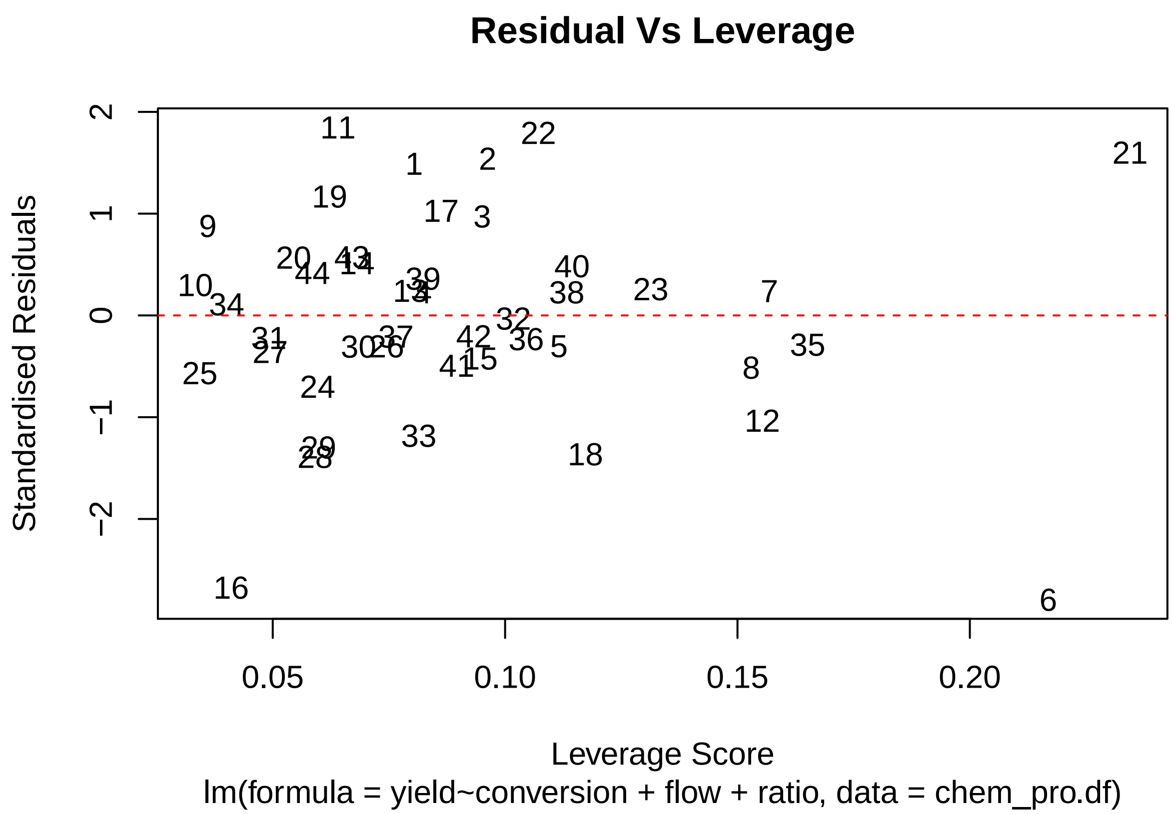
```r
# k = 3
abline(h = mean(pii.vec), lty = 2, col = "blue")
abline(h = 3 * (3 + 1) / 44, lty = 2, col = "red")
legend("bottomright", legend=c("average", "excessive"), col = c("blue", "red"), lty=2)
```



(g) (1 point)

Produce the plot of standardised residual Vs leverage score for `chem_pro.LM` like the one I showed in class.

```r
plot(pii.vec, sres,  type="n", xlab = "Leverage Score", ylab = "Standardised Residuals",
     main = "Residual Vs Leverage", sub = "lm(formula = yield~conversion + flow + ratio, data = chem_pro
text(pii.vec, sres, c(1:44))
abline(h = 0, lty = 2, col = "red")
```



(h) (1 point)

Produce a table of influence measures for `chem_pro.LM` like the one I showed in class.

```r
im = influence.measures(chem_pro.LM)
im
```

```
## Influence measures of
##   lm(formula = yield ~ conversion + flow + ratio, data = chem_pro.df) :
##
##      dfb.1_  dfb.cnvr dfb.flow dfb.rati   dffit cov.r   cook.d    hat inf
## 1  -0.15379  0.171205 -0.37075  0.21414  0.4483 0.957 4.87e-02 0.0804
## 2   0.10348 -0.044898 -0.11423 -0.24833  0.5123 0.958 6.33e-02 0.0962
## 3  -0.02789  0.062869 -0.12017 -0.07752  0.3146 1.111 2.48e-02 0.0951
## 4  -0.01039  0.017236 -0.03453 -0.00626  0.0678 1.200 1.18e-03 0.0823
## 5   0.01094 -0.021314  0.06432  0.00689 -0.1070 1.234 2.93e-03 0.1116
## 6  -0.31180  0.115894  0.17695  0.97151 -1.6193 0.592 5.41e-01 0.2169   *
## 7  -0.05076  0.058095 -0.07330  0.03086  0.1015 1.305 2.64e-03 0.1569   *
## 8   0.10158 -0.115249  0.17135 -0.08036 -0.2165 1.272 1.19e-02 0.1530
## 9  -0.02284  0.038443 -0.03617 -0.02638  0.1692 1.062 7.20e-03 0.0360
## 10  0.02164 -0.022460  0.01842 -0.00746  0.0546 1.135 7.61e-04 0.0333
## 11  0.19537 -0.185560 -0.28038  0.04779  0.4993 0.827 5.85e-02 0.0640
## 12  0.35865 -0.342510  0.18667 -0.38167 -0.4433 1.176 4.90e-02 0.1553
## 13 -0.05581  0.054864 -0.02031  0.04898  0.0699 1.195 1.25e-03 0.0797
## 14 -0.10685  0.105545 -0.03182  0.08916  0.1380 1.156 4.85e-03 0.0681
## 15  0.10868 -0.105522  0.04425 -0.10339 -0.1357 1.200 4.70e-03 0.0946
## 16  0.29033 -0.321183 -0.09788 -0.01622 -0.6034 0.524 7.66e-02 0.0411   *
## 17 -0.25558  0.250764 -0.10094  0.22932  0.3167 1.087 2.50e-02 0.0862
```

```
## 18 -0.09085  0.045557 -0.29539  0.38361 -0.5032 1.036 6.19e-02 0.1173
## 19 -0.02490  0.044340  0.17963 -0.14403  0.3021 1.027 2.26e-02 0.0622
## 20 -0.04657  0.054612  0.05471 -0.02995  0.1349 1.133 4.63e-03 0.0545
## 21  0.28057 -0.240441  0.80109 -0.69167  0.9045 1.109 1.96e-01 0.2345
## 22 -0.28662  0.273128  0.35287  0.08608  0.6404 0.886 9.67e-02 0.1072
## 23 -0.00630  0.007837  0.08246 -0.03858  0.0987 1.265 2.50e-03 0.1313
## 24 -0.00246 -0.000810 -0.13055  0.06366 -0.1750 1.120 7.75e-03 0.0597
## 25  0.00460 -0.000362  0.01388 -0.04641 -0.1062 1.109 2.87e-03 0.0343
## 26  0.00934 -0.002575  0.01017 -0.05271 -0.0855 1.185 1.87e-03 0.0745
## 27 -0.01633  0.021263 -0.01283 -0.02067 -0.0806 1.149 1.66e-03 0.0494
## 28 -0.01828  0.043894 -0.01005 -0.15639 -0.3528 0.965 3.04e-02 0.0591
## 29  0.00710  0.017014  0.01493 -0.17090 -0.3307 0.991 2.69e-02 0.0599
## 30  0.00809 -0.001782  0.00932 -0.04943 -0.0825 1.177 1.74e-03 0.0685
## 31  0.01242 -0.013054 -0.02847  0.00527 -0.0501 1.158 6.43e-04 0.0492
## 32  0.00253 -0.002703 -0.00775  0.00261 -0.0107 1.232 2.94e-05 0.1018
## 33  0.01843 -0.029428 -0.27092  0.14880 -0.3542 1.045 3.11e-02 0.0814
## 34  0.00500 -0.005066 -0.01017  0.00589  0.0218 1.151 1.21e-04 0.0401
## 35 -0.11592  0.110300 -0.03173  0.09279 -0.1270 1.314 4.13e-03 0.1651   *
## 36 -0.06936  0.067273 -0.00626  0.04110 -0.0790 1.229 1.60e-03 0.1045
## 37 -0.03747  0.038115  0.01600  0.00120 -0.0594 1.193 9.04e-04 0.0765
## 38  0.01881 -0.023149 -0.03880  0.03878  0.0797 1.242 1.63e-03 0.1133
## 39  0.08636 -0.085527 -0.00198 -0.03544  0.1066 1.190 2.91e-03 0.0823
## 40  0.15475 -0.152393  0.03017 -0.08905  0.1717 1.221 7.51e-03 0.1144
## 41 -0.10567  0.110095  0.01319  0.00863 -0.1529 1.186 5.96e-03 0.0896
## 42 -0.04383  0.046000  0.00491  0.00218 -0.0645 1.215 1.06e-03 0.0933
## 43  0.03744 -0.047960  0.04730  0.02774  0.1516 1.148 5.85e-03 0.0670
## 44  0.02885 -0.035479  0.02799  0.01741  0.1034 1.155 2.73e-03 0.0585
```

Task 2 (6 points)

The data `USA_real_estate` is about the median price of houses sold in different areas of USA in 2006.

Variable	Description
`mppsf`	Median Price Per Square Foot
`ns`	Number Homes from which the Median Price is computed
`pnh`	Percentage of Homes sold that are build in 2005 or 2006
`pms`	Percentage of Mortgage Foreclosure Sales

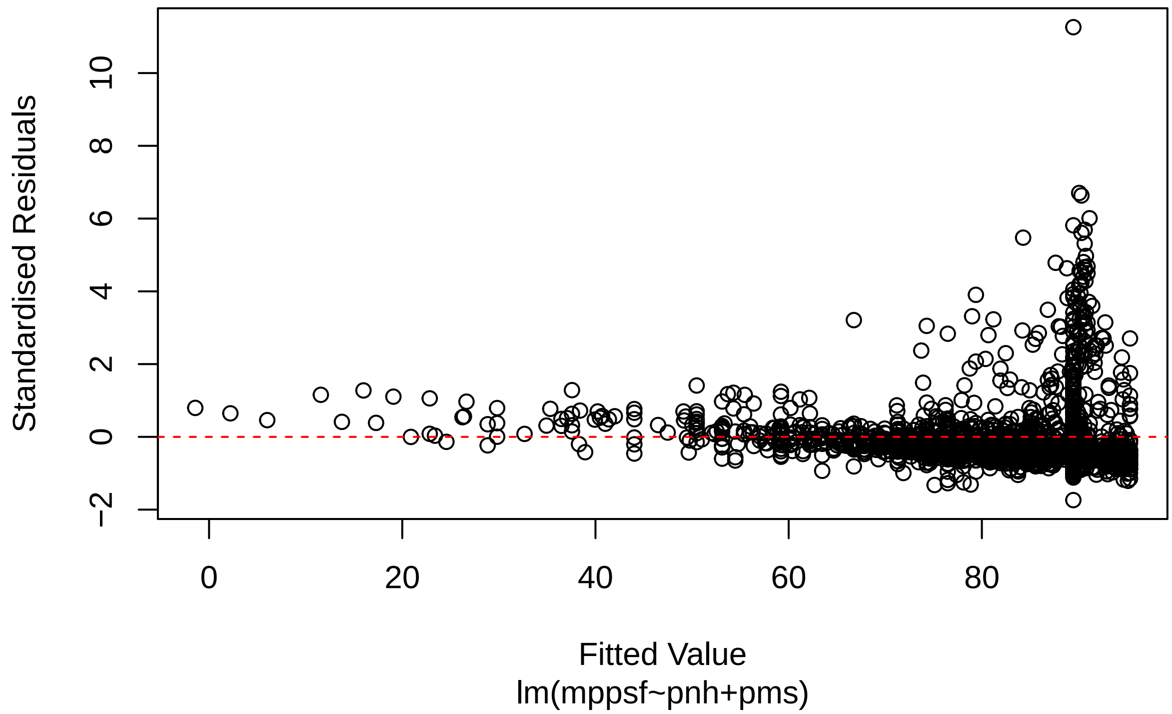
Each data point is for one such area of USA in 2006.

(a) (1 point)

Check for the presence of heteroskedasticity in the model `usare.LM`.

```r
usare.df = read.table(file = USA_real_estate.txt, sep = "", header = TRUE)
usare.LM = lm(mppsf~pnh+pms, data = usare.df)
```

```r
fvs2 = fitted.values(usare.LM)
sres2 = rstandard(usare.LM)
plot(fvs2, sres2, xlab = "Fitted Value", ylab = "Standardised Residuals", sub = "lm(mppsf~pnh+pms)")
abline(h = 0, lty = 2, col = "red")
```



From the above plot, we could see the variance of the residuals is not constant, heteroskedasticity presents in the model.

(b) (1 point)

Estimate the weights for using weighted least squares for the following linear model

$$\text{mppsf}_i = \beta_0 + \beta_1 \text{pnh}_i + \beta_2 \text{pms}_i + \sigma_i \varepsilon$$

```r
z = 2 * log(abs(usare.LM$residuals)) #  z is the auxiliary response,
auxiliary.LM = lm(z~pnh + pms, data = usare.df)  # Perform the auxiliary regression
v.vec = exp(auxiliary.LM$fitted.values) # transform back
w.vec = 1/v.vec
```

(c) (1 point)

Construct the linear model using weighted least squares with your estimated weights, name it `usare.WLS`.

$$\text{mppsf}_i = \beta_0 + \beta_1 \text{pnh}_i + \beta_2 \text{pms}_i + \sigma_i \varepsilon$$

```r
usare.WLS = lm(mppsf~pnh+pms, weights = w.vec, data = usare.df)
```

(d) (1 point)

Explain why `ns` might also be an appropriate estimate for the weights.

(e) (1 point)

Construct the linear model using weighted least squares with the weights based on `ns`, name it `usare.ns.WLS`.

$$\text{mppsf}_i = \beta_0 + \beta_1 \text{pnh}_i + \beta_2 \text{pms}_i + \sigma_i \varepsilon$$

```
usare.ns.WLS = lm(mppsf~pnh+pms, weights = ns, data=usare.df)
summary(usare.ns.WLS)
```

```
##
## Call:
## lm(formula = mppsf ~ pnh + pms, data = usare.df, weights = ns)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -271.74  -67.88  -32.78   15.85 1901.11
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   88.767      1.104  80.383   <2e-16 ***
## pnh            4.262      2.754   1.548    0.122
## pms          -98.019      6.296 -15.568   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 162.6 on 1919 degrees of freedom
## Multiple R-squared:  0.1252, Adjusted R-squared:  0.1243
## F-statistic: 137.4 on 2 and 1919 DF,  p-value: < 2.2e-16
```

(f) (1 point)

Compare `usare.WLS` with `usare.ns.WLS`. Which of the two models do you prefer? Explain your answer.

Task 3 (5 points)

The data `grossboxoffice` is about yearly gross box office receipts from moives screened in Australia.

(a) (1 point)

Load the data file `grossboxoffice.txt` into R, and construct the following model, name it as `gbo.LM`.

$$\text{GrossBoxOffice}_i = \beta_0 + \beta_1 \text{year}_i + \varepsilon$$

Comment on the validity of `gbo.LM`.

```
gbo.df = read.table(file = "grossboxoffice.txt", sep="", header = TRUE)
gbo.LM = lm(GrossBoxOffice ~ year, data = gbo.df)
summary(gbo.LM)
```

```
##
## Call:
## lm(formula = GrossBoxOffice ~ year, data = gbo.df)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -116.382  -79.197    6.083   62.260  121.697
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -58386.485   2952.825  -19.77   <2e-16 ***
## year            29.534      1.483   19.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 77.44 on 30 degrees of freedom
## Multiple R-squared:  0.9297, Adjusted R-squared:  0.9274
## F-statistic: 396.8 on 1 and 30 DF,  p-value: < 2.2e-16
```

(b) (1 point)

Explore the possibility of using AR(1), AR(2), and AR(3).

- Check the posibility of using AR(1)

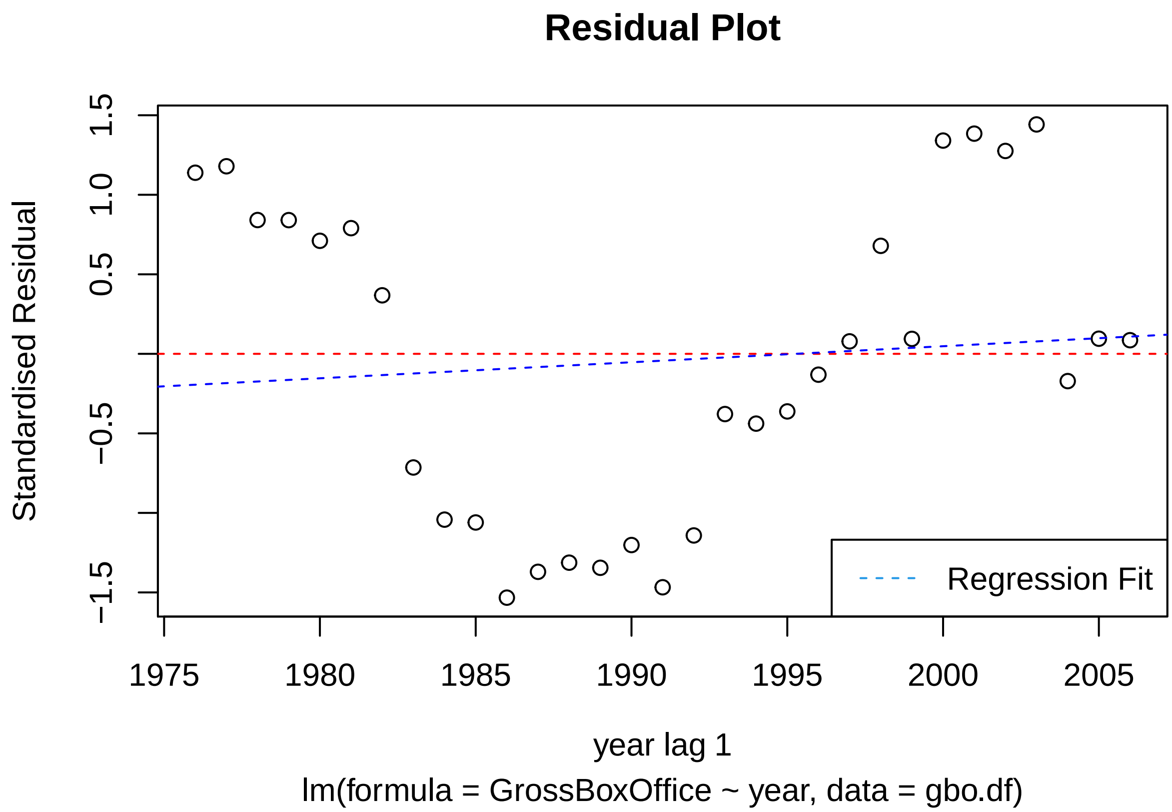
```r
res3 = residuals(gbo.LM)
res.lag.df = data.frame(x = res3[-length(res3)], y = res3[-1])
auxiliary.LM3 = lm(y~x, data = res.lag.df)
summary(auxiliary.LM3)
```

```
##
## Call:
## lm(formula = y ~ x, data = res.lag.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -97.854 -14.930   4.111  18.335  98.292
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.75798    6.61120  -0.568    0.574
## x            0.83474    0.08679   9.618 1.59e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.81 on 29 degrees of freedom
## Multiple R-squared:  0.7613, Adjusted R-squared:  0.7531
## F-statistic:  92.5 on 1 and 29 DF,  p-value: 1.586e-10
```

Because of the extremely small p-value, there is no evidence against that the auxiliary model is valid.

```r
ar_res_plot = function(lag = 1) {
  index = 0:(lag - 1) - nrow(gbo.df)
  x = gbo.df$year[index]
  y = rstandard(gbo.LM)[-(1:lag)]
  plot(x, y, xlab = bquote("year lag"~.(lag)), ylab = "Standardised Residual",
       main = "Residual Plot", sub = deparse(gbo.LM$call))
  abline(a = 0, b = 0, lty = 2, col = "red")
```

```r
  abline(lm(y~x), lty = 2, col = "blue")
  legend("bottomright", "Regression Fit", lty = 2, col = 4)
}
ar_res_plot(1)
```



From the plot, we see that the residual do not show a random scatter, but it is highly likely that ε_i is not only depend on X_{i-1}. We might need to consider a polynomial formula.

- Check the posibility of using AR(2)

```r
res.lag2.df = data.frame(x = res3[-(32:31)], y = res3[-(1:2)])
auxiliary2.LM = lm(y~x, data = res.lag2.df)
summary(auxiliary2.LM)
```

```
##
## Call:
## lm(formula = y ~ x, data = res.lag2.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -92.203 -28.922   6.103  27.199 104.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -6.5155     8.7028  -0.749     0.46
## x             0.7197     0.1124   6.403 6.23e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.67 on 28 degrees of freedom
```

```
## Multiple R-squared:  0.5942, Adjusted R-squared:  0.5797
## F-statistic:    41 on 1 and 28 DF,  p-value: 6.23e-07
```

Because of the extremely small p-value, there is no evidence against that the auxiliary model is valid.

- Check the posibility of using AR(3)

```r
res.lag3.df = data.frame(x = res3[-(32:30)], y = res3[-(1:3)])
auxiliary3.LM = lm(y~x, data = res.lag3.df)
summary(auxiliary3.LM)
```

```
##
## Call:
## lm(formula = y ~ x, data = res.lag3.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -102.295  -44.392    3.305   34.007  107.118
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.0467    10.7252  -0.937 0.357200
## x             0.5664     0.1363   4.157 0.000292 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.76 on 27 degrees of freedom
## Multiple R-squared:  0.3903, Adjusted R-squared:  0.3677
## F-statistic: 17.28 on 1 and 27 DF,  p-value: 0.0002917
```

Because of the extremely small p-value, there is no evidence against that the auxiliary model is valid. However, the adjusted R-squared are decreaseing for these three models.

(c) (1 point)

Obtain a final model for predicting `GrossBoxOffice` for `year=1975`, name it as `gbo.final.M`.

As our goal is to predict, although observing a decreased adjusted R squared, the t-statistic make it reasonable to push to AR(3) model.

```r
lag = 3
index = 0:(lag - 1) - nrow(gbo.df)
GrossBoxOffice = gbo.df$GrossBoxOffice[-(1:lag)]
year = gbo.df$year[-(1:lag)]
df.lag = gbo.df$GrossBoxOffice[index]
gbo_lag.df = data.frame(GrossBoxOffice, year, df.lag)
gbo.final.M = lm(GrossBoxOffice ~ year + poly(df.lag, 2), data = gbo_lag.df)
summary(gbo.final.M)
```

```
##
## Call:
## lm(formula = GrossBoxOffice ~ year + poly(df.lag, 2), data = gbo_lag.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -112.100  -30.311    3.406   33.331   70.108
##
```

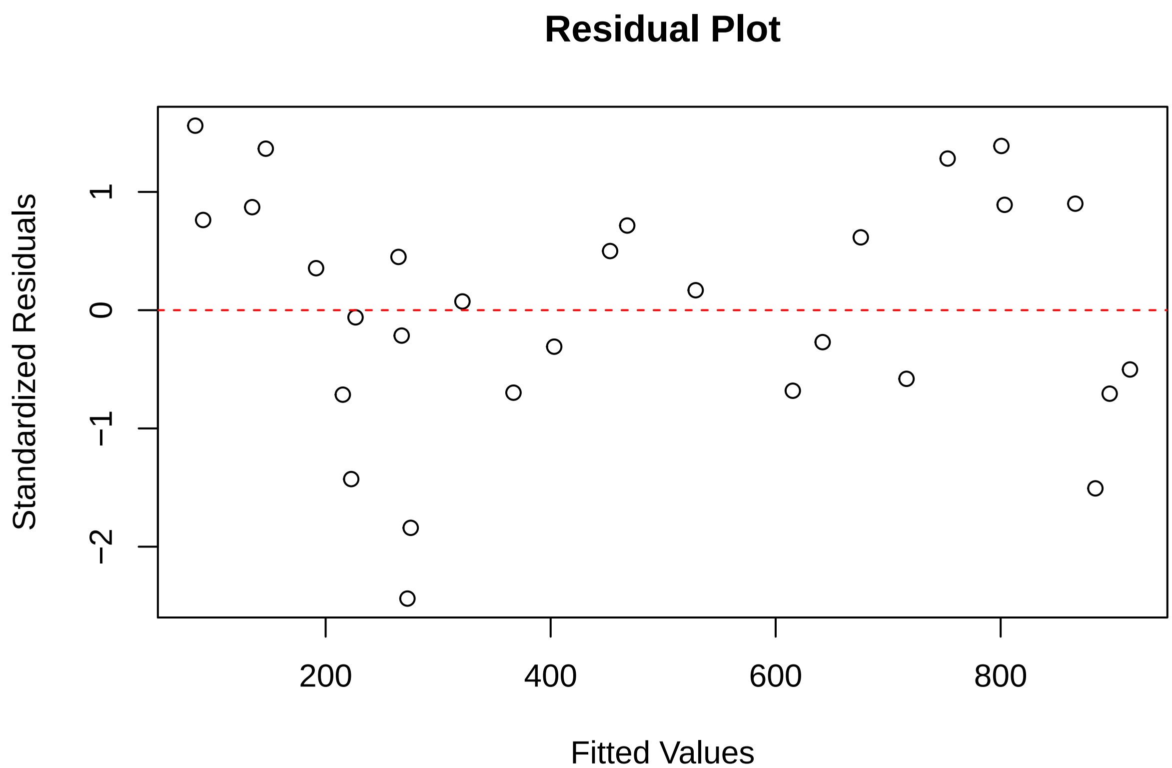
```
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       -10200.332  10064.372  -1.014   0.3205    
## year                   5.352      5.050   1.060   0.2994    
## poly(df.lag, 2)1    1210.508    222.307   5.445 1.18e-05 ***
## poly(df.lag, 2)2    -190.540     69.086  -2.758   0.0107 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.25 on 25 degrees of freedom
## Multiple R-squared:  0.9724, Adjusted R-squared:  0.9691 
## F-statistic: 293.4 on 3 and 25 DF,  p-value: < 2.2e-16
```

(d) (1 point)

Produce diagnostic plots to justify your choice of model.

- Residual Plot

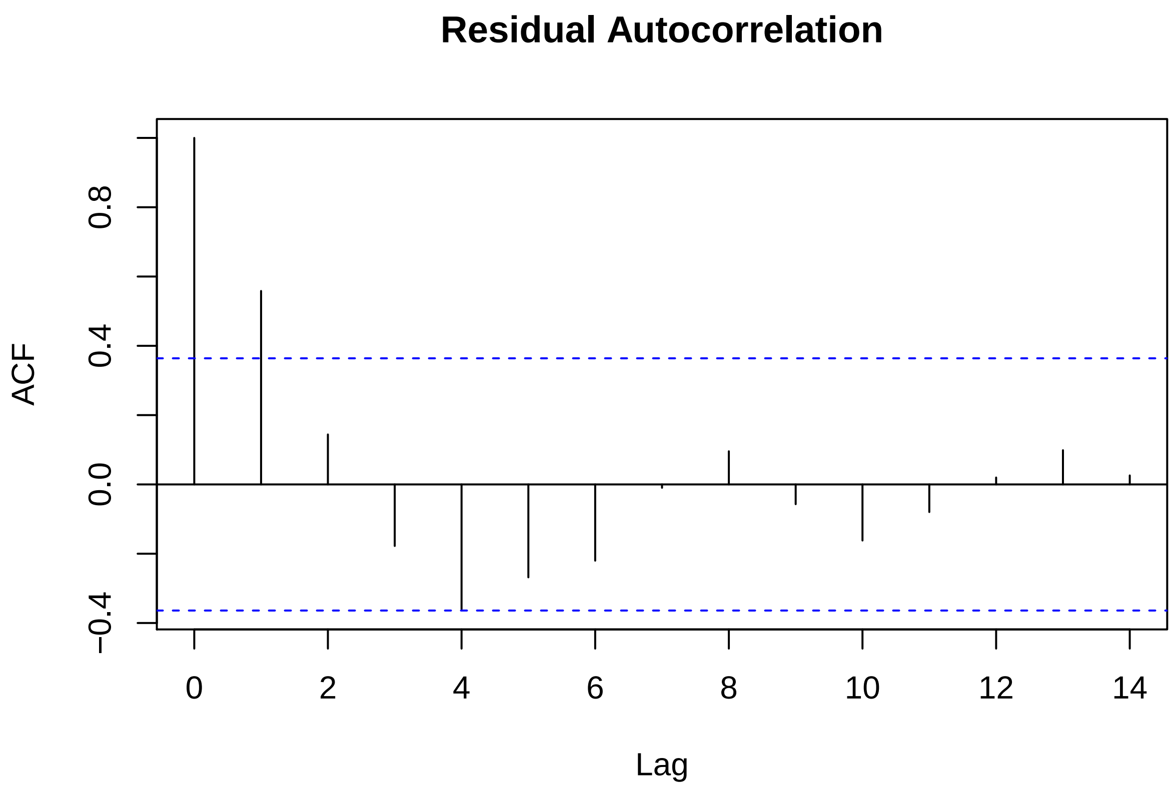
```
fvs3d = fitted.values(gbo.final.M)
sres3d = rstandard(gbo.final.M)
plot(fvs3d, sres3d, xlab = "Fitted Values", ylab = "Standardized Residuals", main = "Residual Plot")
abline(h = 0, lty = 2, col = "red")
```



There is no evidence for non-constant variance. The residuals seem like have a bit underlying pattern, but considering the small sample size and the pattern is not so obvious, we may continue and check other assumptions.

- ACF Plot

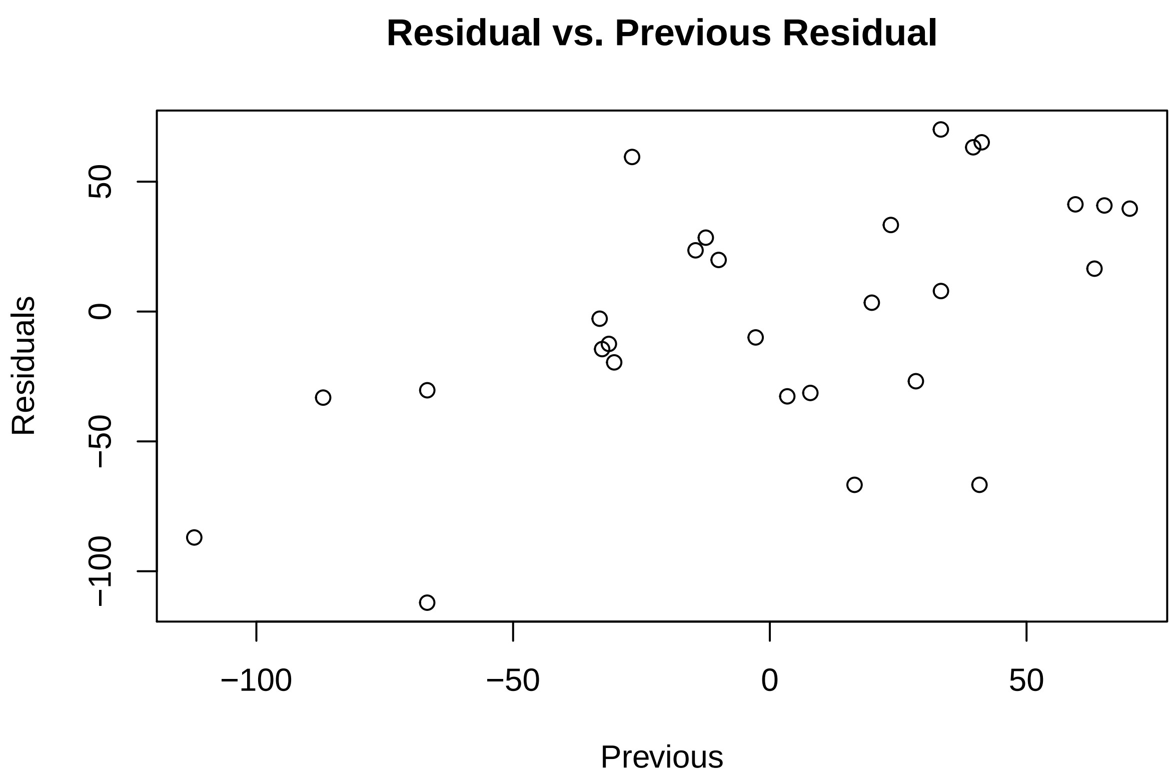
```
acf(gbo.final.M$residuals, main="Residual Autocorrelation")
```



There is no evidence that the residuals for our final model is correlated.

- Residual vs Previous Residual

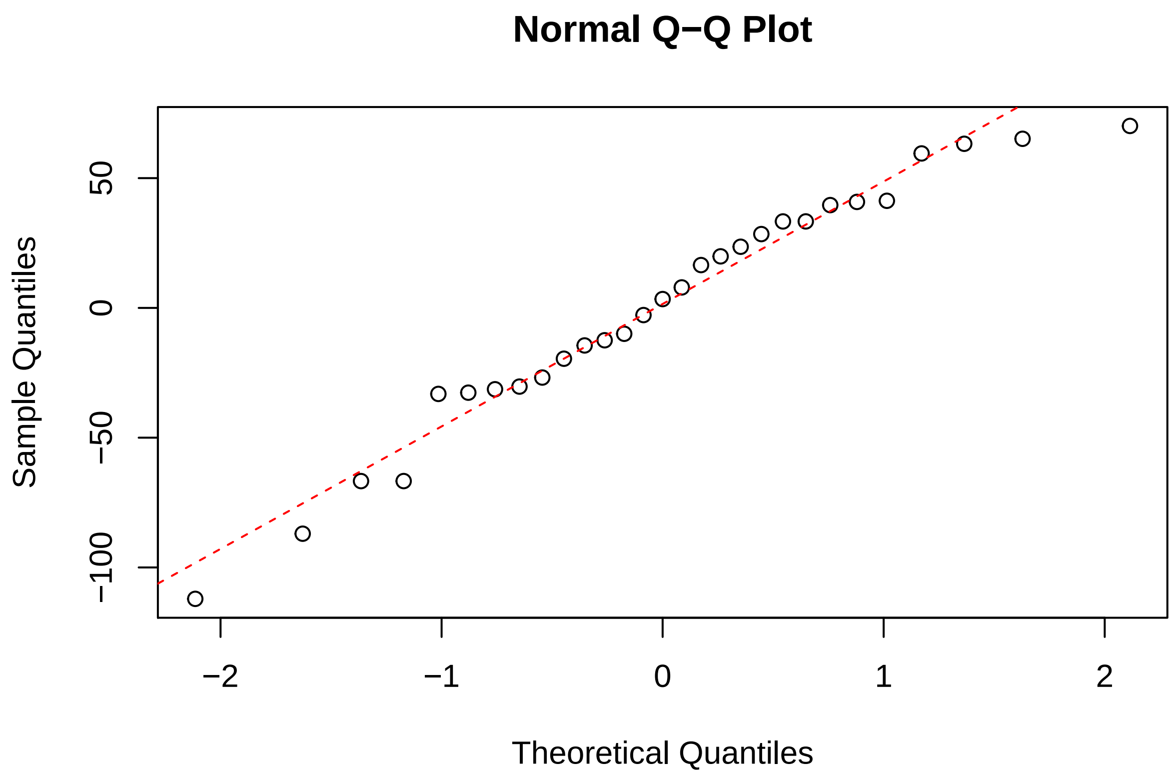
```
plot(gbo.final.M$residuals[-length(gbo.final.M$residuals)], gbo.final.M$residuals[-1], xlab="Previous",
```



The plot doesn't show obvious pattern, the residuals basically randomly scatter around mean=0. So no evidence show the model is inappropriate.

- Nomarlity

```r
qqnorm(gbo.final.M$residuals)
qqline(gbo.final.M$residuals, lty=2, col="red")
```




```r
shapiro.test(gbo.final.M$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  gbo.final.M$residuals
## W = 0.96151, p-value = 0.358
```

We check the Q-Q plot and there is no violation of our assumption. Due to the small sample size, we check the Shapiro-Wilk normality test as well. There is no evidence against the normality.

- Overall, the model choice is reasonable.

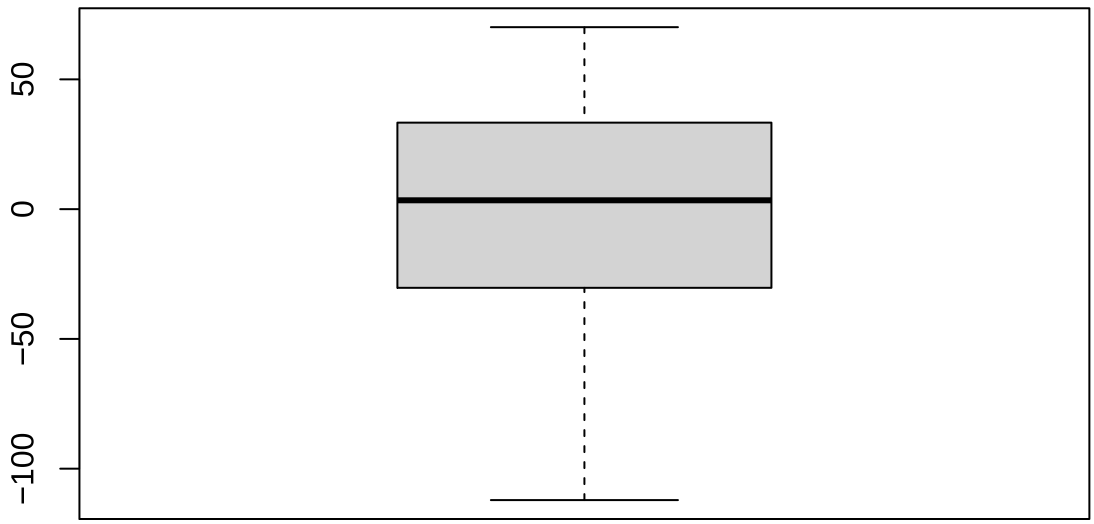
(e) (1 point)

Describe any weakness in your `gbo.final.M`.

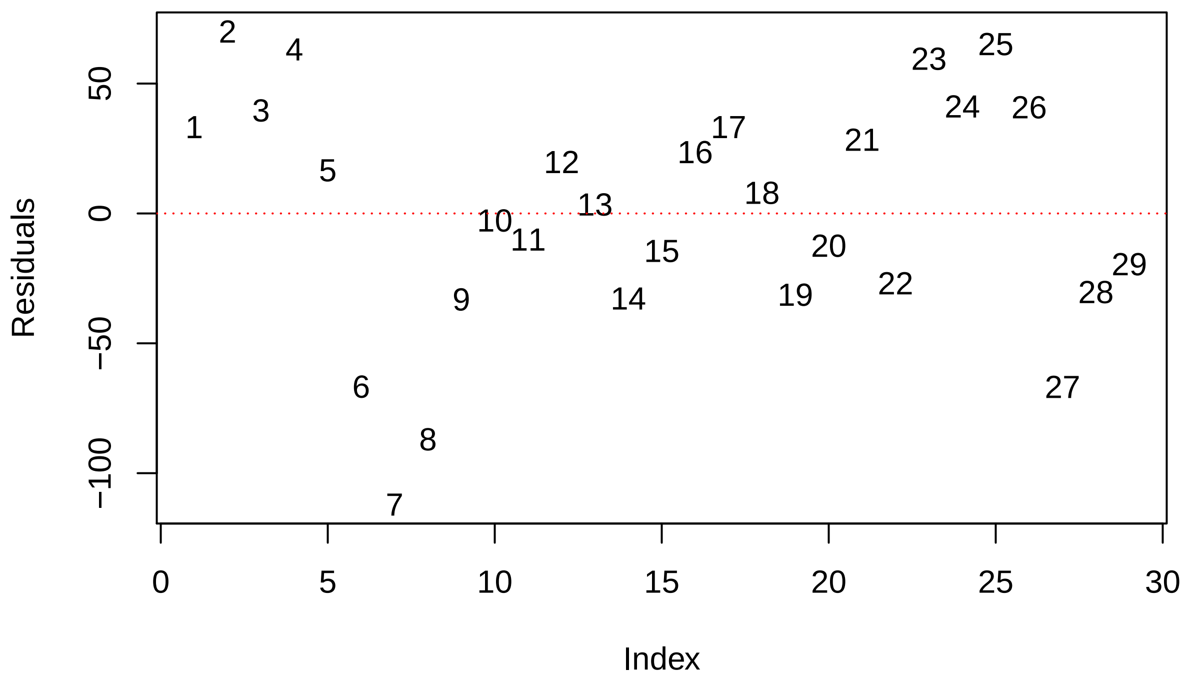
With the small data size, the evidence is not so strong. ## (f) (1 point)

Use your model `gbo.final.M` to identify any outliers.

```r
boxplot(gbo.final.M$residuals)
```




```r
plot(gbo.final.M$residuals, type="n",
ylab = "Residuals")
text(gbo.final.M$residuals);
abline(h=0,lty=3, col="red")
```




```r
which(gbo.final.M$residuals< -100)
```

```
## 7
## 7
```