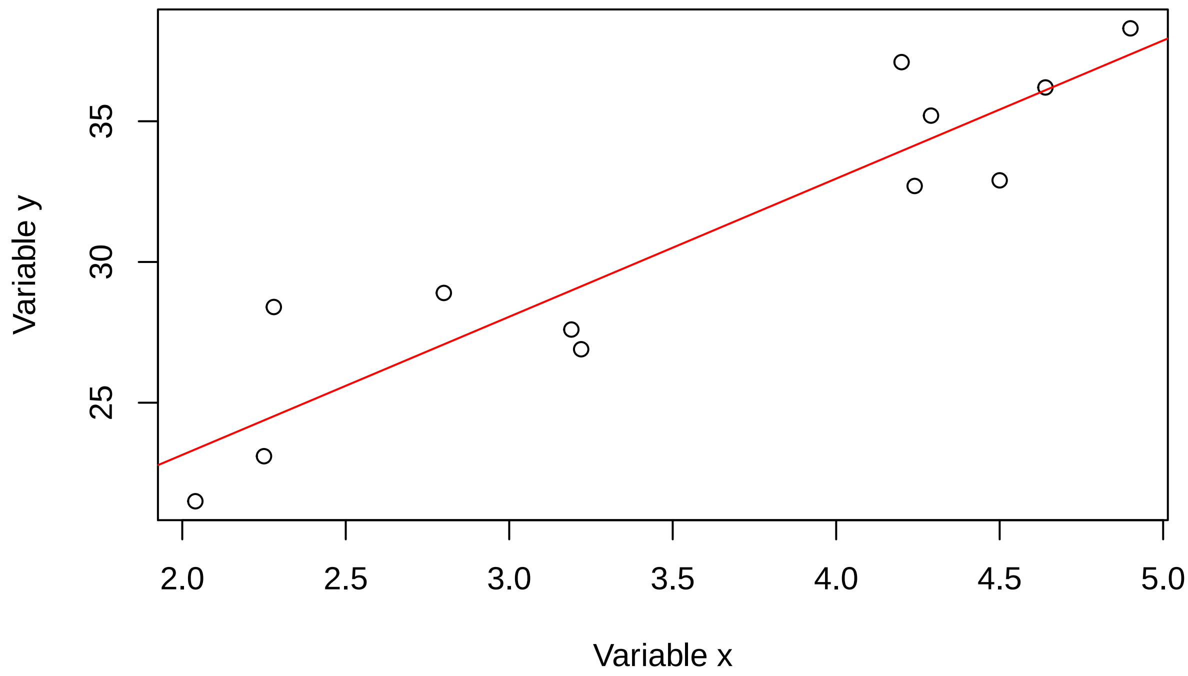
Devoir 1

Question No 1a)

```
df = data.frame(c(2.04, 21.5, 1/24),
                c(2.28, 28.4, 1/12),
                c(4.2, 37.1, 1/48),
                c(4.9, 38.3, 7/48),
                c(4.5, 32.9, 1/24),
                c(4.29, 35.2, 1/12),
                c(3.19, 27.6, 1/6),
                c(2.8, 28.9, 1/48),
                c(2.25, 23.1, 1/12),
                c(4.64, 36.2, 7/48),
                c(3.22, 26.9, 1/12),
                c(4.24, 32.7, 1/12))
colnames(df) = c(1:12)
rownames(df) = c("xi", "yi", "wi")
df = data.frame(t(df))


Q2a = lm(df$yi~df$xi)
plot(df[,2]~df[,1], xlab = "Variable x", ylab = "Variable y")
abline(Q2a, col ="red")
```




```r
summary(Q2a)
```

```
##
## Call:
## lm(formula = df$yi ~ df$xi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5159 -1.5408 -0.5885  1.1475  3.8788
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.3320     2.4070   5.539 0.000248 ***
## df$xi         4.9075     0.6539   7.505 2.05e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.241 on 10 degrees of freedom
## Multiple R-squared:  0.8492, Adjusted R-squared:  0.8342
## F-statistic: 56.33 on 1 and 10 DF,  p-value: 2.05e-05
```

On constate que les valeurs de "Estimate" pour Beta0 et Beta1 sont 13.3320 et 4.9075

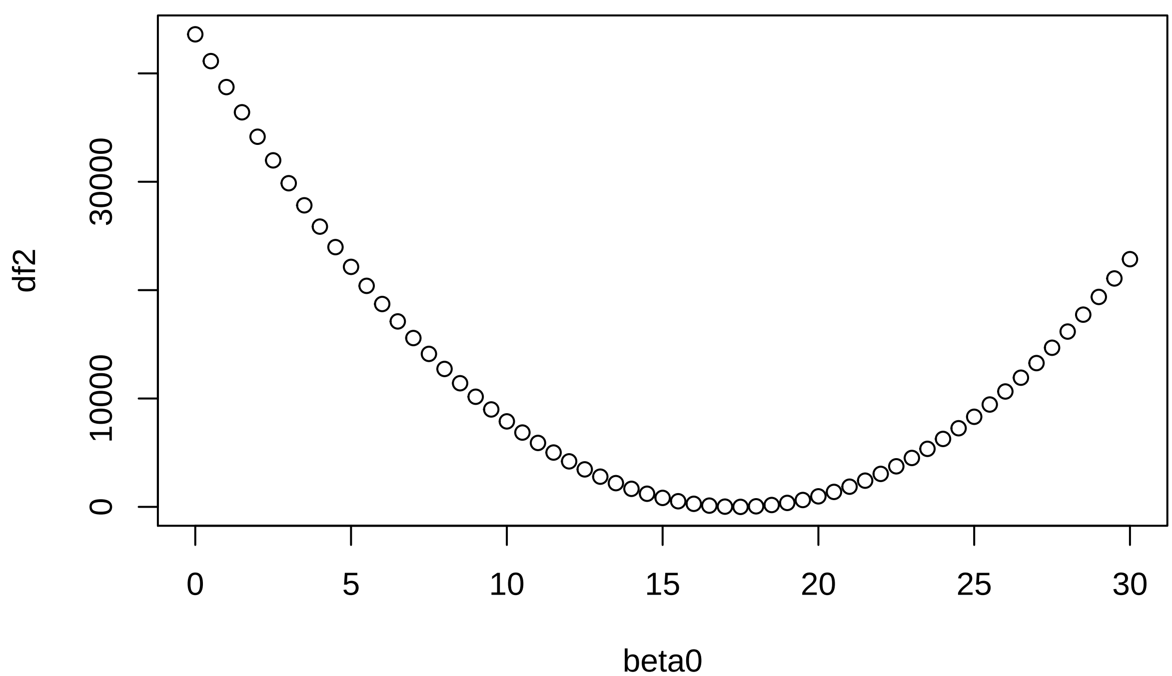
Question No 1b)

```r
anova(Q2a)
```

```
## Analysis of Variance Table
##
## Response: df$yi
##           Df  Sum Sq Mean Sq F value   Pr(>F)
## df$xi      1 282.820 282.820  56.331 2.05e-05 ***
## Residuals 10  50.207   5.021
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
beta1 = Q2a$coefficients[2]
beta0 = seq(0,30,by=.5)
df2 = data.frame()
datalist = list()
for (i in 1:length(beta0)) {
  rss = sum(df$yi-Q2a$coefficients[1] - beta0[i])^2
  datalist[[i]] = rss
}

df2 = do.call(rbind, datalist)
plot(df2~beta0)
```



On constate que le point le plus bas du graphique se trouve autour de beta 0 = 13, ce qui correspond à la valeur du coefficient de beta0 chapeau (13.3320).

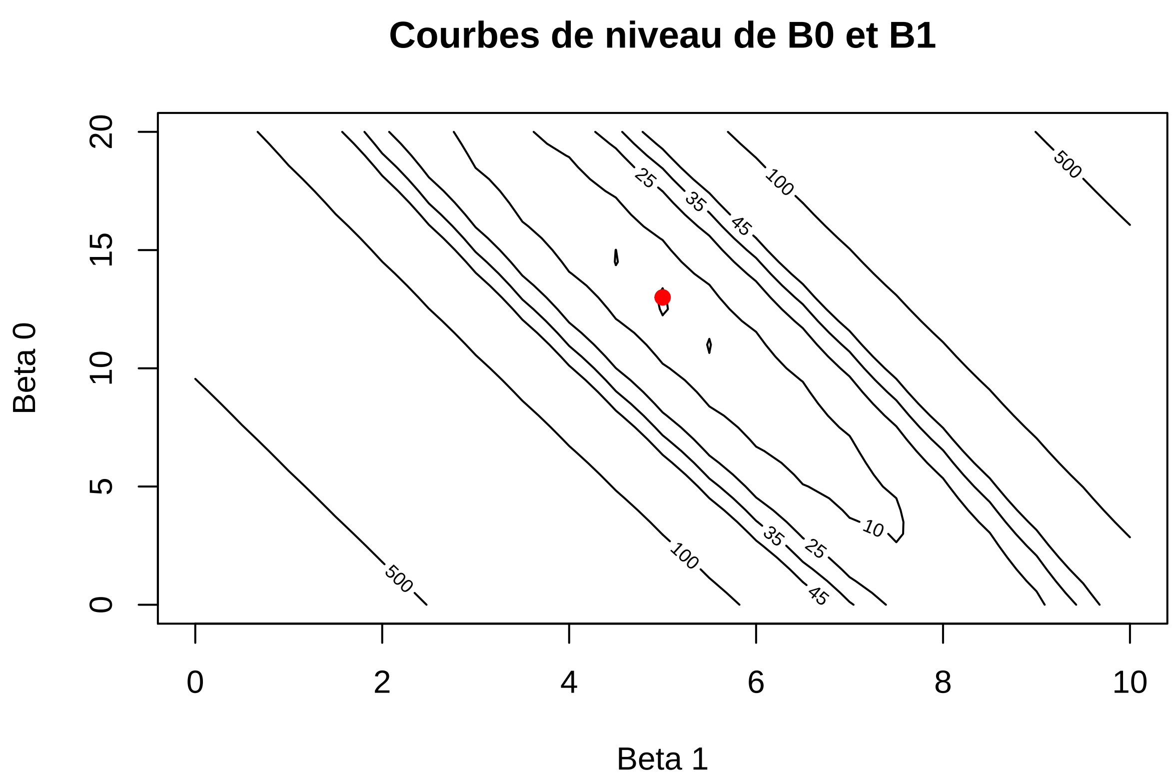
Question No 1c)

```r
beta0 = seq(0,20, by = .5)
  beta1 = seq(0,10, by = .5)

  m = matrix(data = NA, nrow = length(beta1), ncol = length(beta0))
  q2c = function(b0, b1) {
    ssrbw = (sum(df$wi*(df$yi-b0-b1*df$xi)^2))
  }

  for (j in 1:length(beta0)) {
    for (i in 1:length(beta1)) {
      m[i,j] = q2c(beta0[j], beta1[i])
    }
  }

  ind = which(m == min(m), arr.ind = TRUE)
  rname = beta1[ind[,1]]
  cname = beta0[ind[,2]]
  dot = as.numeric(c(rname, cname))
  contour(x = beta1, y = beta0, z = m, levels = c(3.5,10,25,35,45,100,500), xlab = "Beta 1", ylab = "Be
  points(x = dot[1], y = dot[2], col = "red", pch = 19)
```



On constate que la fonction de Beta 0 et Beta sont trouve son minimum lorsque Beta 0 ~ 13 et que Beta 1 ~ 5, ce qui correspond aux valeur calculées avec la méthode numérique (13.3320 et 4.9075).

Question No 1d)

```r
beta0 = seq(0,20, by = .5)
beta1 = seq(0,10, by = .5)

rss_b0 = function(b0) {
  rss1 = (sum(df$yi-b0)^2)
}

rss_b0_b1 = function(b0,b1) {
  rss2 = (sum(df$wi*(df$yi-b0-b1*df$xi)^2))
}

rss_b0_min = optimize(rss_b0, interval = c(0,30))
rss_b0_b1_min = nlm(rss_b0_b1, c(0, 30), c(0,30))
print(rss_b0_min)
```

```
## $minimum
## [1] 29.99993
##
## $objective
## [1] 77.45443
```

```r
print(rss_b0_b1_min)
```

```
## $minimum
## [1] 317.9449
##
## $estimate
## [1]  26.98571 -90.70370
##
## $gradient
## [1] -2.558255e-08 -6.561481e-09
##
## $code
## [1] 1
##
## $iterations
## [1] 6
```

Pour cette section, il semble que les résultats obtenus par les fonctions optimize et nlm n'ont pas permis d'obtenir les mêmes résultats que la minimisation des RSS pour produire des estimateurs valables. Sauf indication contraire, les méthodes vues précédemment semblent plus fiables.