

## A2Q3

### Undergraduate Student

```
enzyme <- read.delim("Enzyme.txt")
```

(i)

```
enzyme <- read.delim("Enzyme.txt")
enzyme$Y <- 1/enzyme$Y
enzyme$x <- 1/enzyme$x
init <- lm(Y~x, data = enzyme)
summary(init)
```

```
##
## Call:
## lm(formula = Y ~ x, data = enzyme)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.056684 -0.004123  0.000694  0.002766  0.063565
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.033759   0.006684   5.051 0.000118 ***
## x            0.454014   0.020061  22.632 1.41e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02175 on 16 degrees of freedom
## Multiple R-squared:  0.9697, Adjusted R-squared:  0.9678
## F-statistic: 512.2 on 1 and 16 DF,  p-value: 1.411e-13
```

```
# beta 0
alpha0 <- summary(init)$coefficient[1,1]
beta0 <- 1/alpha0
beta0
```

```
## [1] 29.62201
```

```
# beta 1
alpha1 <- summary(init)$coefficient[2,1]
beta1 <- alpha1*beta0
beta1
```

```
## [1] 13.44881
```

(ii)

```
enzyme <- read.delim("Enzyme.txt")
library(MASS)
par.ini = c(b0 = beta0, b1 = beta1)
nonlinear <- nls(Y~(b0*x)/(b1+x), data = enzyme, start = par.ini, trace = TRUE)
```

```
## 6.587434 : 29.62201 13.44881
## 4.303552 : 28.14230 12.59805
## 4.302271 : 28.13786 12.57534
## 4.302271 : 28.13708 12.57449
## 4.302271 : 28.13705 12.57445
```

```
summary(nonlinear)
```

```
##
## Formula: Y ~ (b0 * x)/(b1 + x)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## b0  28.1370    0.7280    38.65 < 2e-16 ***
## b1  12.5745    0.7631    16.48 1.85e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5185 on 16 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 4.347e-07
```

```
# beta 0
beta0 <- summary(nonlinear)$coefficient[1,1]
beta0
```

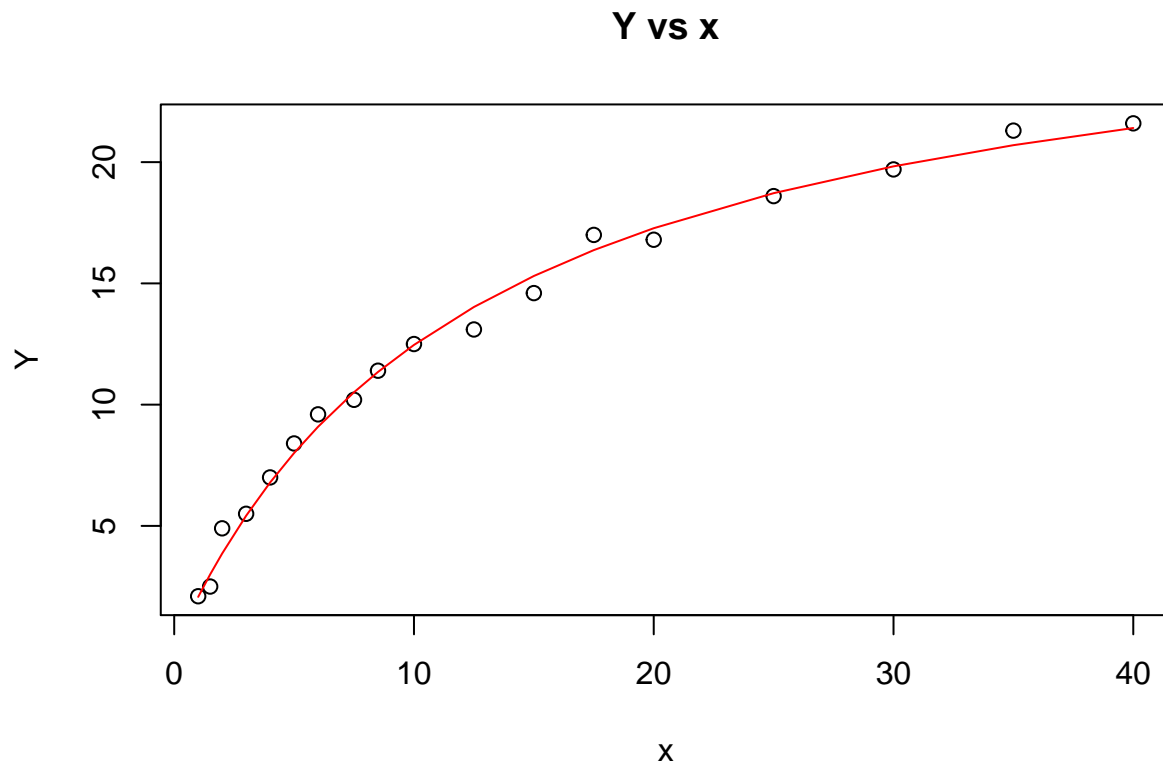
```
## [1] 28.13705
```

```
# beta 1
beta1 <- summary(nonlinear)$coefficient[2,1]
beta1
```

```
## [1] 12.57445
```

(iii)

```
plot(enzyme$x, enzyme$Y, xlab = "x", ylab = "Y", main = "Y vs x")
xx <- enzyme$x
yy <- (beta0*xx)/(beta1+xx)
points(xx, yy, "l", col = "red")
```

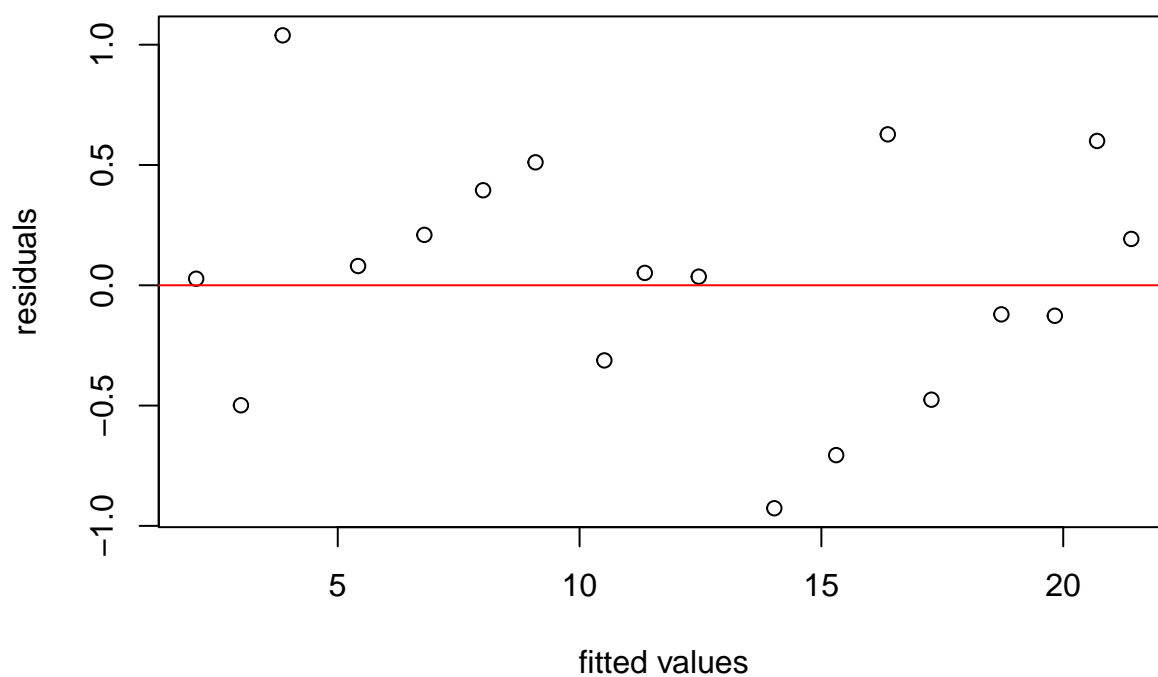


**Comments:** We see that the line matches the pattern of the data points. The model fit appears to be adequate.

(iv)

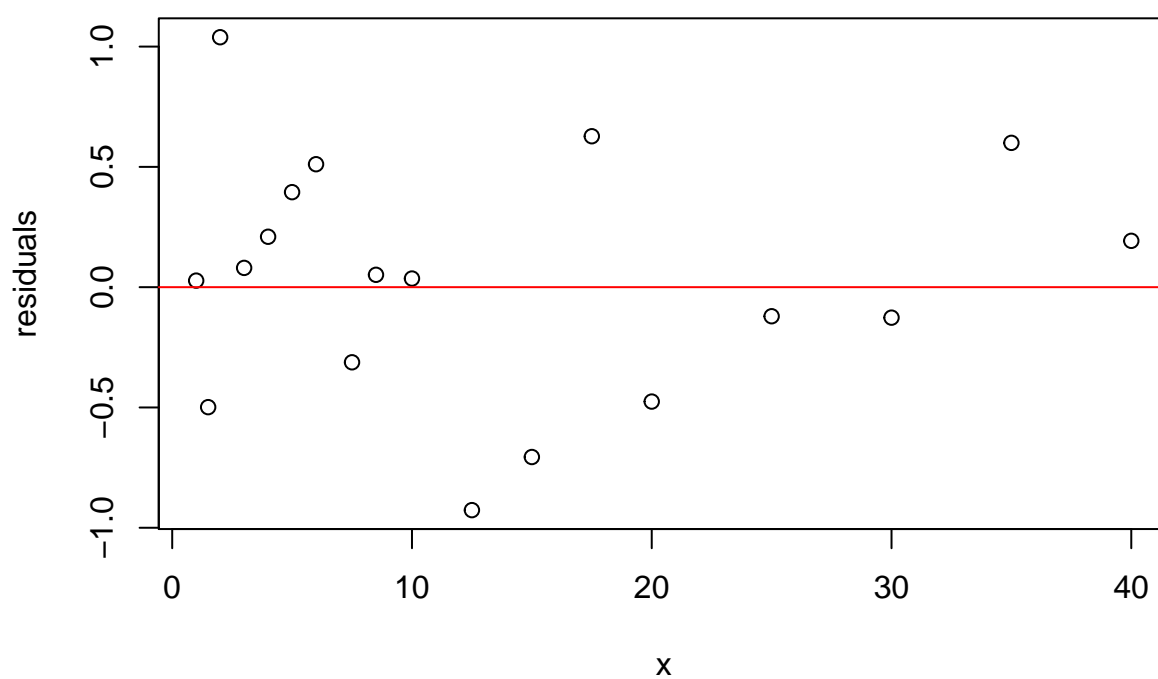
```
plot(fitted.values(nonlinear), residuals(nonlinear), xlab = "fitted values",  
     ylab = "residuals", main = "residuals vs fitted values")  
abline(h=0, col = 'red')
```

**residuals vs fitted values**



```
plot(enzyme$x, residuals(nonlinear), xlab = "x", ylab = "residuals",  
     main = "residuals vs x")  
abline(h=0, col = 'red')
```

**residuals vs x**

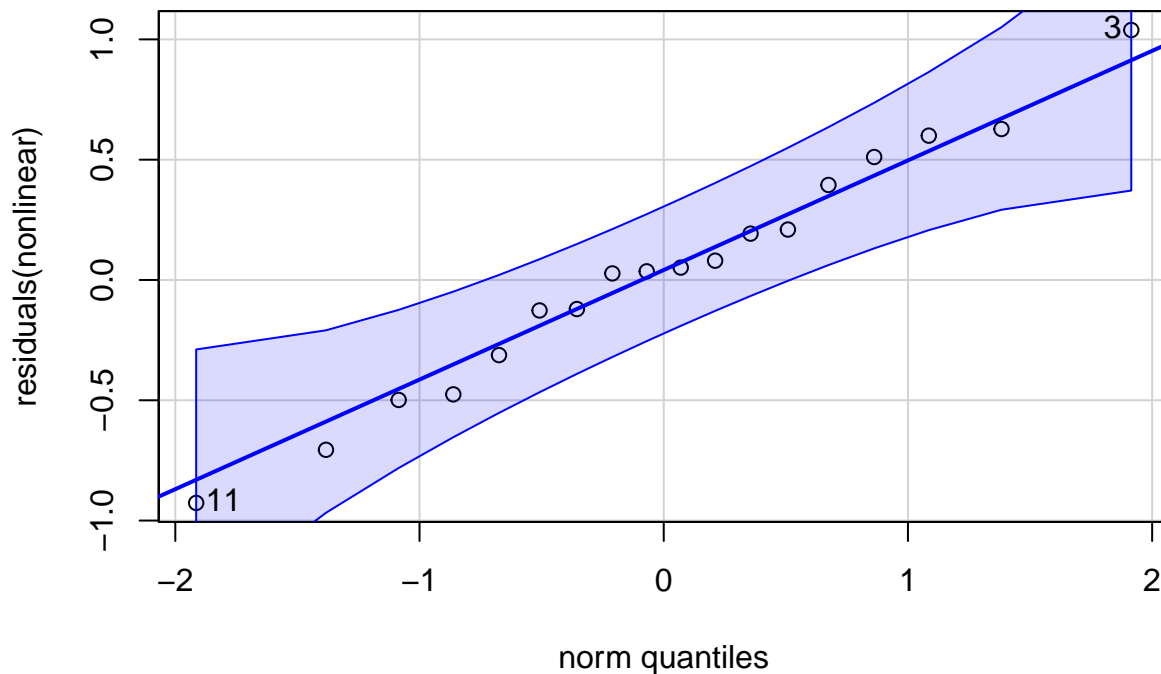


```
library(car)
```

```
## Warning: package 'car' was built under R version 4.0.5
```

```
## Loading required package: carData
```

```
qqPlot(residuals(nonlinear))
```



```
## [1] 3 11
```

**Comments:** Three graphs show that residuals are well-behaved and follow normality condition. Except for point 3 and point 11, these two points potentially can be outliers.

(v)

If we use fewer points with replications, since the some of data points have the same Y, the model can fit the data better when minimizing residuals. Therefore, the fitted line can be closer to all data points and reduce bias. On the other hand, since the fitted model has low bias, the model will have high variance. This means that the model does not generalized on the original data, or perform badly on original data. Also, since data are replicated, the residuals are not following normality conditions anymore. We can see a clear pattern on residual plot.

(vi)

(a)

```
confint(nonlinear, "b0", level=0.95)
```

```
## Waiting for profiling to be done...
```

```
## 4.995744 : 12.57445
## 4.386915 : 12.16642
## 4.386759 : 12.17273
## 4.651273 : 11.76952
## 4.65118 : 11.77432
## 4.65118 : 11.77431
## 5.095608 : 11.38687
## 5.095515 : 11.3916
## 5.095515 : 11.39157
## 5.719438 : 11.01932
## 5.719346 : 11.02394
## 5.719346 : 11.02389
## 6.522437 : 10.66606
## 6.522346 : 10.67056
## 6.522346 : 10.6705
## 4.995744 : 12.57445
## 4.382556 : 12.97078
## 4.382196 : 12.98069
## 4.382196 : 12.98078
## 4.623135 : 13.40008
## 4.623059 : 13.40472
## 4.623059 : 13.40478
## 5.024614 : 13.84084
## 5.024537 : 13.8456
## 5.024537 : 13.84567
## 5.586406 : 14.29941
## 5.586328 : 14.30427
## 5.586327 : 14.30436
## 6.308185 : 14.77672
## 6.308106 : 14.7817
## 6.308106 : 14.78181
## 7.189614 : 15.2738
## 7.189535 : 15.2789
## 7.189535 : 15.27903
```

```
##      2.5%      97.5%
## 26.64509 29.79721
```

The confidence interval is (26.64509, 29.79721).

(b)

$H_0 = 20$  vs  $H_A \neq 20$ :

```
tstat <- (beta1 - 20)/(summary(nonlinear)$coefficient[2,2]/sqrt(18))
tstat
```

```
## [1] -41.28666
```

```
pval <- 2*pt(tstat, df=18-1)  
pval
```

```
## [1] 1.720587e-18
```

**Comment:** Since the t-statistics is way smaller than the critical value 0.05, there is a strong evidence against the null hypothesis and we reject the null hypothesis. We conclude that beta 1 can not be 20.