HW4

Chen Xupeng

2014012882

3.6

```
hardness<- read.table('CH01PR22_947709365.txt')
names(hardness) <- c("y","x")
is.data.frame(hardness)
```

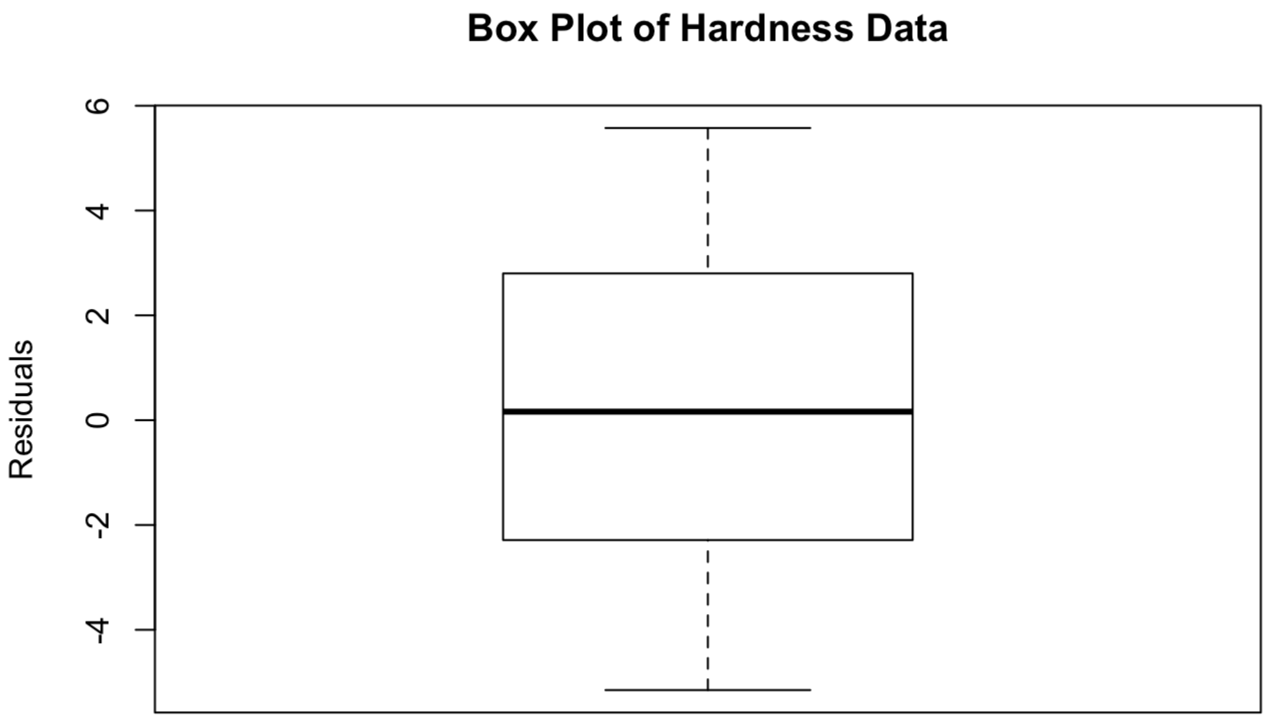
```
## [1] TRUE
```

```
hardness.fit <- lm(y~x,data=hardness)
sumtable<-summary(hardness.fit)
sumtable
```

```
##
## Call:
## lm(formula = y ~ x, data = hardness)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1500 -2.2188  0.1625  2.6875  5.5750
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 168.60000    2.65702   63.45  < 2e-16 ***
## x             2.03438    0.09039   22.51 2.16e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.234 on 14 degrees of freedom
## Multiple R-squared:  0.9731, Adjusted R-squared:  0.9712
## F-statistic: 506.5 on 1 and 14 DF,  p-value: 2.159e-12
```

a

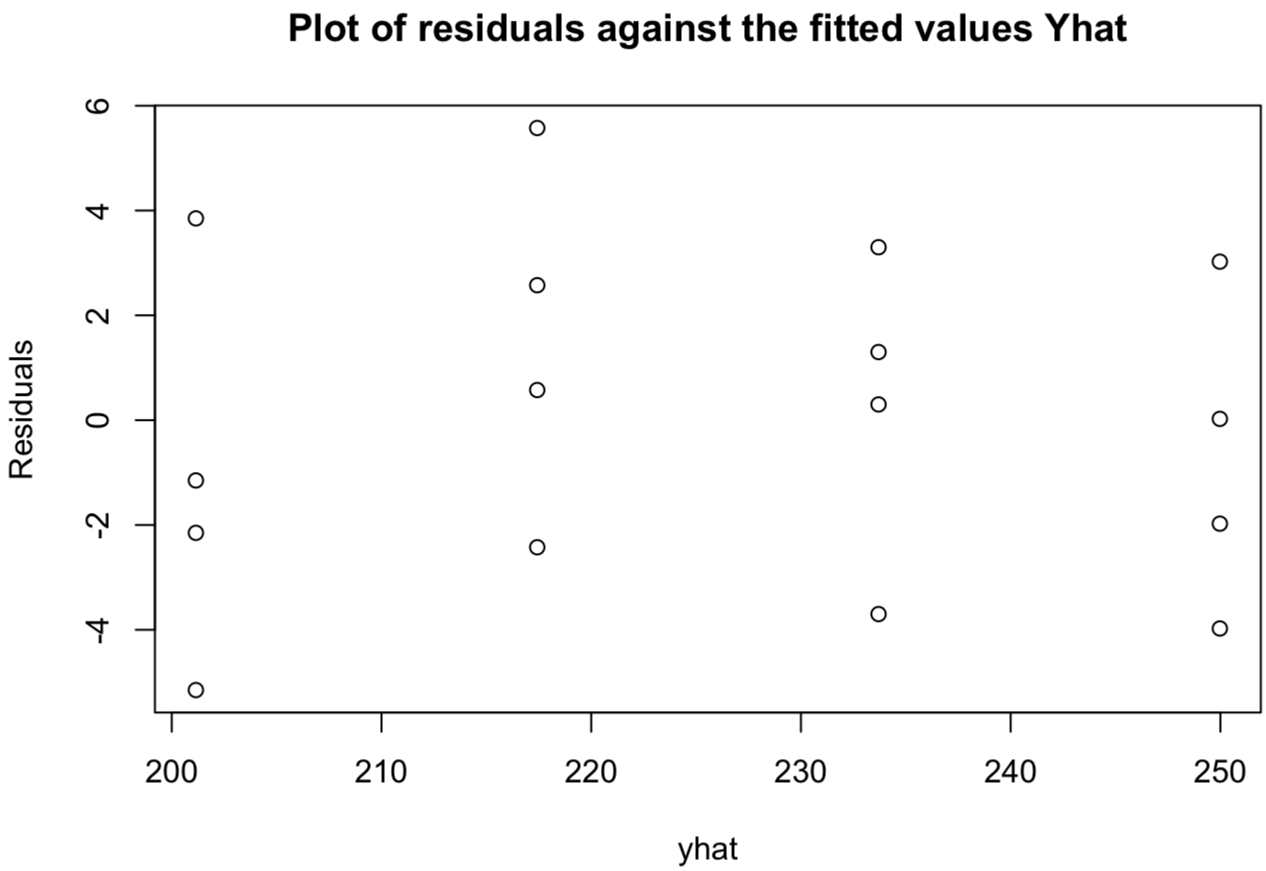
```
boxplot(resid(hardness.fit),main="Box Plot of Hardness Data", ylab="Residuals")
```



The mean of residuals is zero

b

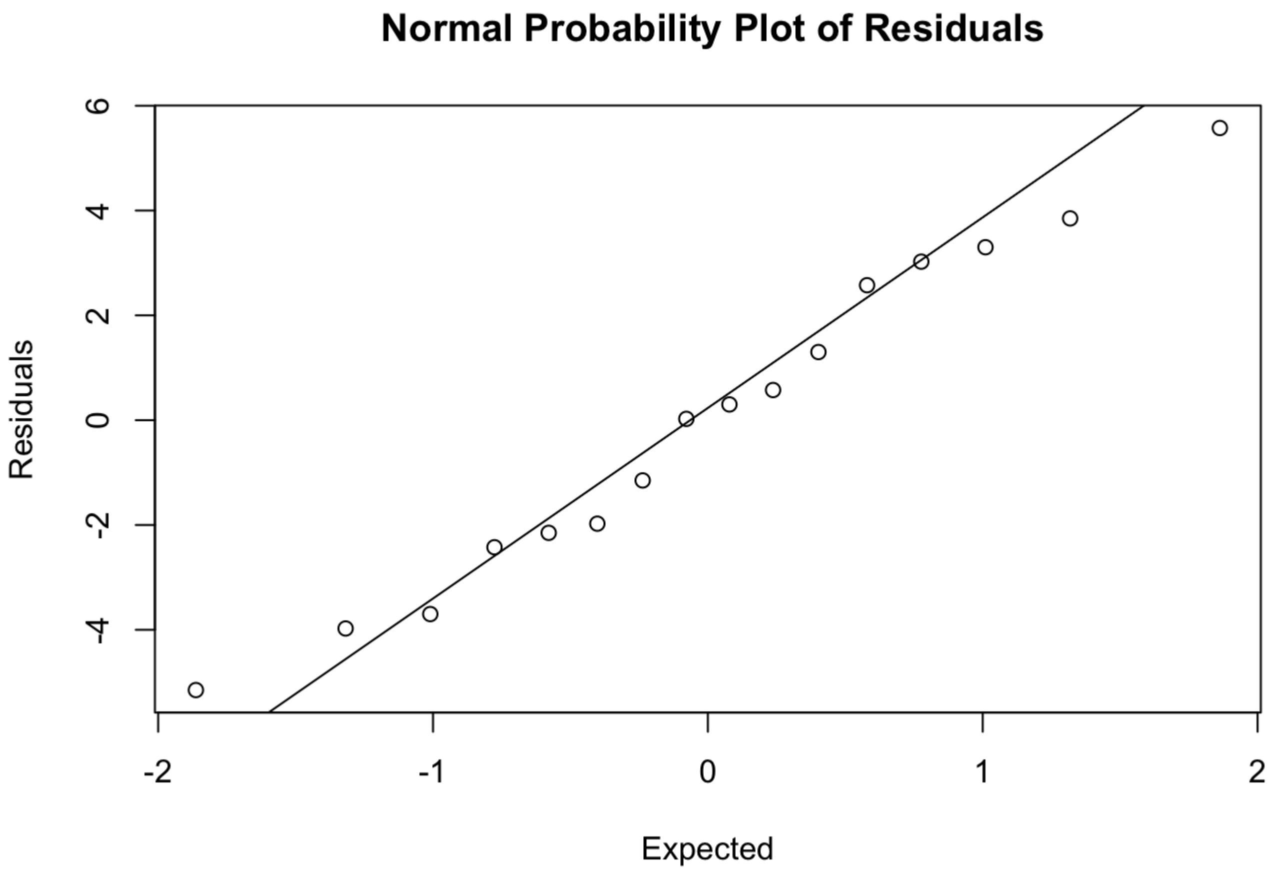
```
yhat <- fitted(hardness.fit); resid <- resid(hardness.fit)
plot(yhat,resid,main="Plot of residuals against the fitted values Yhat",
     ylab="Residuals",xlab='yhat')
```



There is one residual equals to 5.575 a little higher than others.

C

```
hardness.stdres = rstandard(hardness.fit)
qqnorm(resid,
       ylab="Residuals",
       xlab="Expected",
       main=" Normal Probability Plot of Residuals")
qqline(resid)
```




```
StdErr = summary(hardness.fit)$sigma
n = 16
ExpVals = sapply(1:n, function(k) StdErr * qnorm((k-.375)/(n+.25)))
cor(ExpVals,sort(resid))
```

```
## [1] 0.9916733
```

With n=16, from Table B.6, the critical value for the coefficient of correlation between the ordered residuals and the expected values under normality when the distribution of error terms is normal using a 0.05 significance level is 0.941. Since 0.9916733 > 0.941, the assumption of normality appeared reasonable.

4.5

a

```
interval <- function(dat){
  names(dat) <- c("y","x")
  is.data.frame(dat)
  data.fit <- lm(y~x,data=dat)
  sumtable<-summary(data.fit)
  coef <- sumtable$coefficients
  signif<-1-(1-0.9)/4.0
  tvalue <-qt(signif, 14)
  beta_0 = list(coef[1]-tvalue*coef[3],coef[1]+tvalue*coef[3])
  beta_1 = list(coef[2]-tvalue*coef[4],coef[2]+tvalue*coef[4])
  return(list(beta_0,beta_1))
}
interval(hardness)
```

```
## [[1]]
## [[1]][[1]]
## [1] 162.9013
##
## [[1]][[2]]
## [1] 174.2987
##
##
## [[2]]
## [[2]][[1]]
## [1] 1.8405
##
## [[2]][[2]]
## [1] 2.22825
```

From R result, $b_0 = 168.6, s(b_0)=2.65702, b_1 = 2.03438, s(b_1)=0.09039.$ Since t(0.975, 14) = 2.145, Bonferroni joint confidence intervals for $β_0$ and $β_1$, using a 90% percent family confidence coefficient, are 168.6±2.145(2.65702) = [162.901, 174.299] for $β_0$ and 2.03438±2.145(0.09039) = [1.840, 2.228] for $β_1$. At least 90% of the time, both coefficients will be within the limits stated.

c

The 90% joint confidence interval means that both will be in the interval at least 90% of the time. Restated, at least one of them will be out of the interval no more than 10% of the time. We cannot get more specific than this.

4.9

a

For Bonferroni, use $b_0+b_1X_j±t(1−0.1/6, 14)s\{\hat Y_h\}$, with t(1−.10/6, 14) = 2.35982.

```
meaninterval <-function(X_h){
mse <- mean(sumtable$residuals^2)
sYh <- (mse * ( 1/16.0 + (X_h -ave(hardness$x)[1])**2/(sum((hardness$x - ave(hardness
$x))**2)/16.0)) )**0.5
beta_0 = coef[1]
beta_1 = coef[3]
list(beta_0 +beta_1*X_h - qt(1-.10/6, 14) * sYh,beta_0 +beta_1*X_h + qt(1-.10/6, 14)
* sYh)
}
```

Using this function we have CI of 20,30,40 are [215.1106, 228.3704], [245.9163, 250.7052], [265.1384, 284.6236] respectively.

The 90% joint confidence interval means that all three mean hardness will be in their respective interval at least 90% of the time.Restated, at least one of them will be out of the interval no more than 10% of the time. We cannot get more specific than this.

4.12

```
galleys.x <- c(7, 12, 10, 10, 14, 25, 30, 25, 18, 10, 4, 6)
cost.y <- c(128, 213, 191, 178, 250 , 446, 540, 457, 324, 177, 75, 107)
```

a

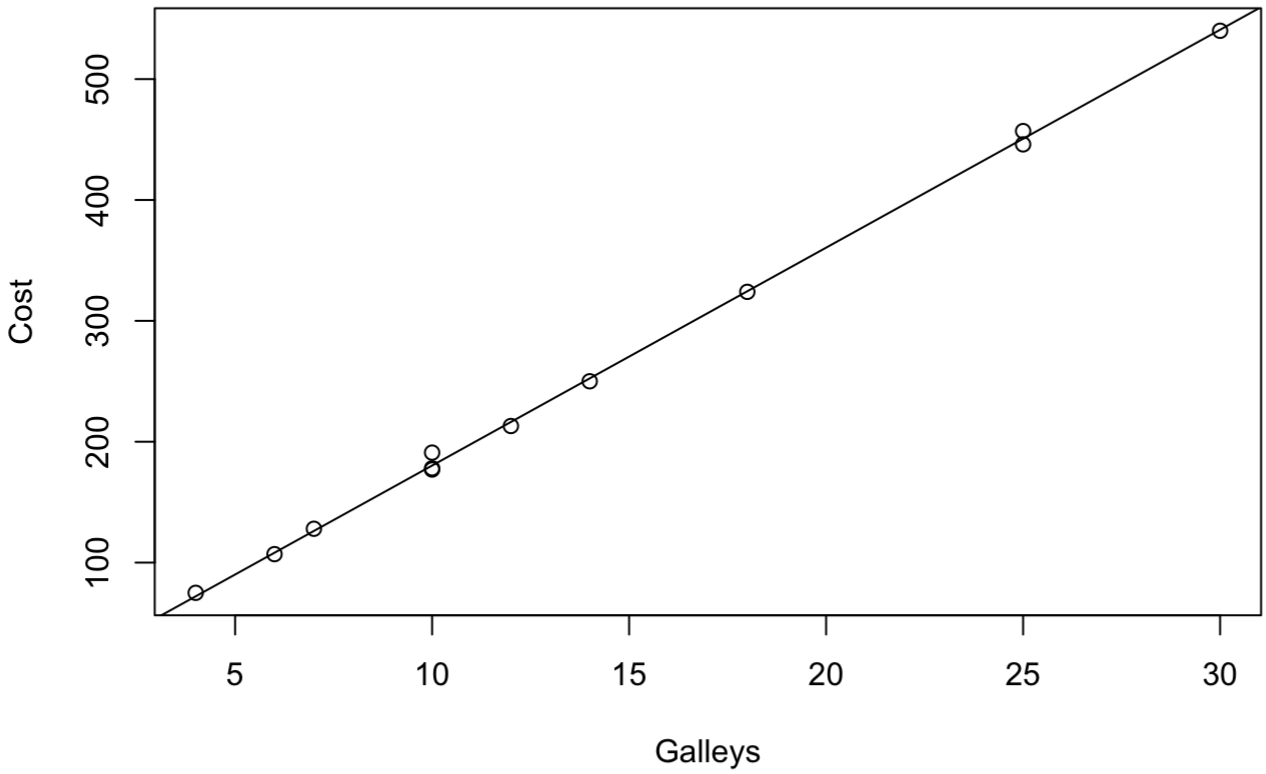
```
typos.lm <- lm(cost.y~galleys.x-1)
typos.lm
```

```
##
## Call:
## lm(formula = cost.y ~ galleys.x - 1)
##
## Coefficients:
## galleys.x
##     18.03
```

\[\hat Y_h=18.03X \]

b

```
plot(galleys.x,cost.y,xlab= "Galleys", ylab="Cost")
abline(typos.lm)
```



It appears that the model fits good

c

```
historical.norm <- data.frame(galleys.x=1)
alpha <- 0.02
typos.int <- predict.lm(typos.lm, newdata = historical.norm , interval = "confidence"
, level = 1-alpha)
typos.int
```

```
##       fit      lwr      upr
## 1 18.0283 17.81226 18.24435
```

Alternatives: $H_0:E[Y]=\beta_{10}=17.50$

$H_0:E[Y]\neq\beta_{10}=17.50$

CI: $17.81226\leq E[Y_h]\leq 18.24435$

Decision rule:

If β_{10} falls within the confidence interval for $E[Y_h]$, conclude H_0; If β_{10} does not fall within the confidence interval for $E[Y_h]$, conclude H_A

Conclusion:

Since 17.50<17.81226; therefore, accept H_A.

d

```
newdata.galleys <- data.frame(galleys.x=10)
typos.pred <- predict(typos.lm, newdata.galleys, level=0.98, interval = "predict", s
e.fit = TRUE)
typos.pred
```

```
## $fit
##       fit      lwr     upr
## 1 180.283 167.8441 192.722
##
## $se.fit
## [1] 0.7948375
##
## $df
## [1] 11
##
## $residual.scale
## [1] 4.506806
```

$\hat Y_h=180.283$

s[pred]=4.506806

180.283±2.738769(4.506806)

$167.8441≤Y_{h(new)}≤192.722$