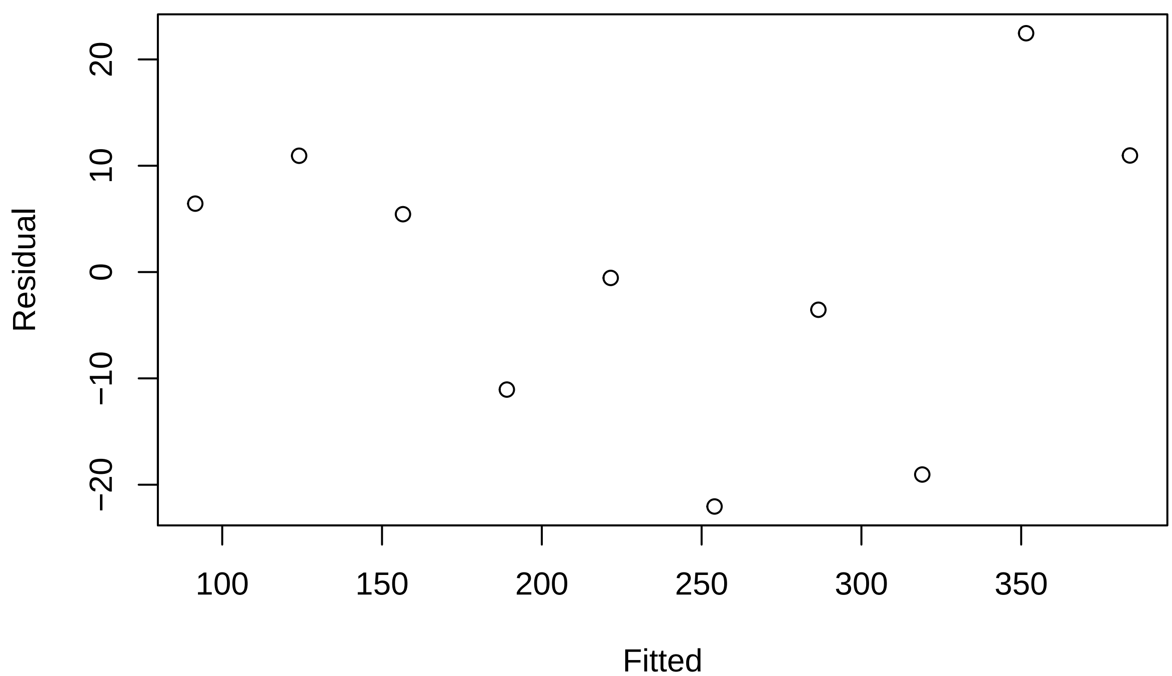
STAT_5044_HW3

zhengzhi lin

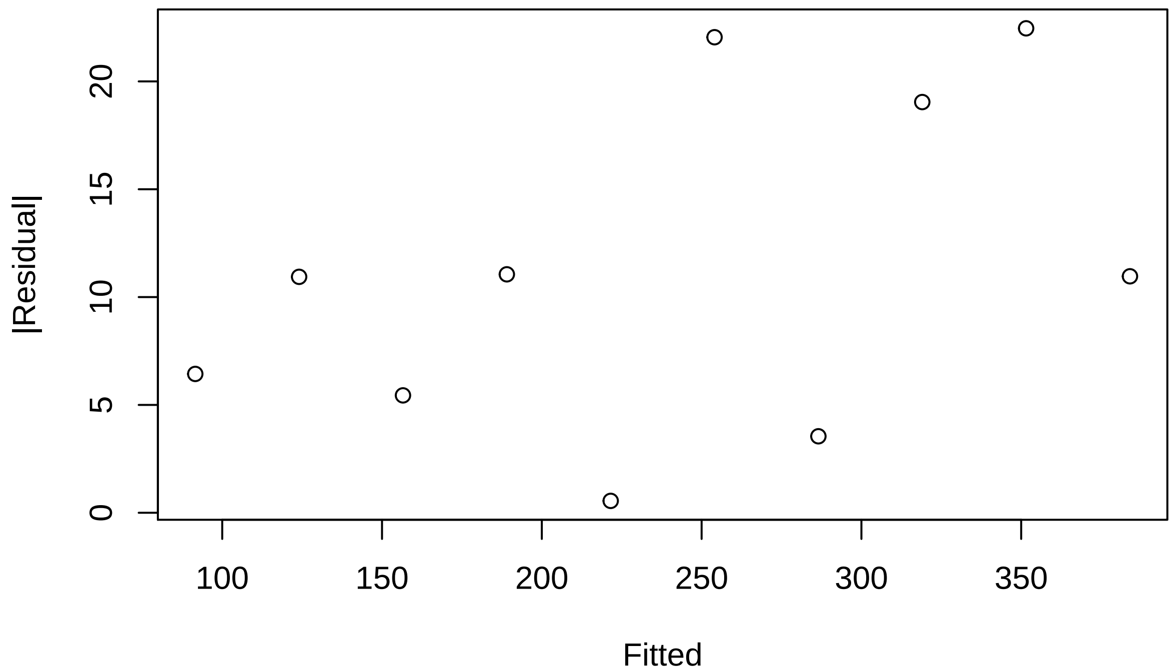
2019.10.15

Problem 1

```r
x <- c(0,1,2,3,4,5,6,7,8,9)
y <- c(98,135,162,178,221,232,283,300,374,395)


lmfit <- lm(y~x)
plot(fitted(lmfit),residuals(lmfit),xlab = "Fitted", ylab = "Residual")
```




```r
plot(fitted(lmfit),abs(residuals(lmfit)),xlab = "Fitted", ylab = "|Residual|")
```




```r
x <- cbind(rep(1,10),x)
r <- y - x %*% solve(t(x) %*% x) %*% t(x) %*% y  #residuals same as lm
#linearity verified

# n1=5=n2, rL=2, rU=10, runs=3, fails to reject randomness.

# BF test for homoscedasticity
r1 <- residuals(lmfit)[4:8]
r2 <- residuals(lmfit)[-c(4,5,6,7,8)]
n1 <- 5
n2 <- 5
r1nod <- median(r1)
r2nod <- median(r2)
d1 <- abs(r1 - r1nod)
d2 <- abs(r2 - r2nod)
s <-(sum((d1 - mean(d1))^2) + sum((d2 - mean(d2))^2))/(10-2)
t_bf <- (mean(d1) - mean(d2))/sqrt((1/n1 + 1/n2) * s) # t distribution with df 5+5-2=8

pt(t_bf, 8, lower.tail = FALSE, log.p = FALSE) # P-value is .159, fail to reject constant variance
```

```
## [1] 0.1587943
```

```r
#test for normality
shapiro.test(residuals(lmfit)) # fail to reject normality
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lmfit)
## W = 0.96123, p-value = 0.7998
```

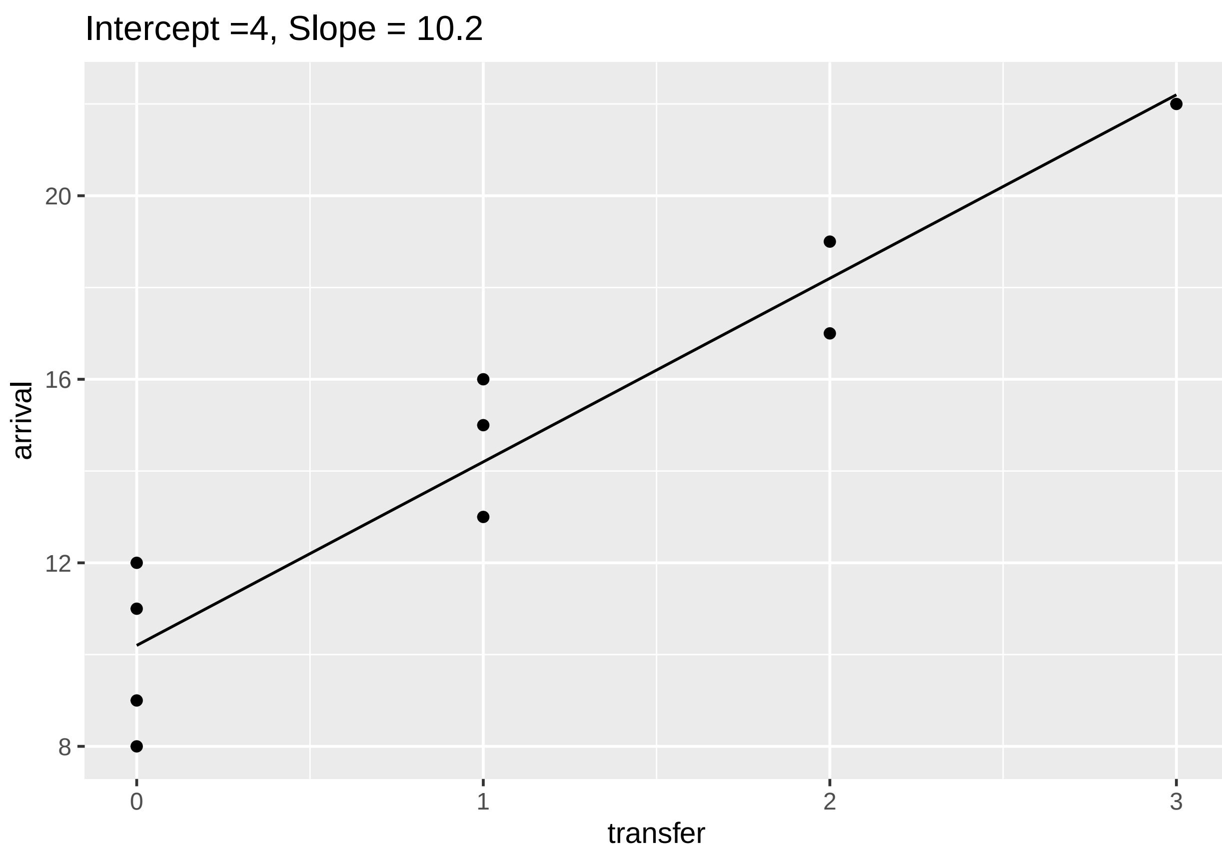
Problem 2

By chencking the residual plot, there is a linear relationship because residuals are spread equally along the ranges of predictors

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

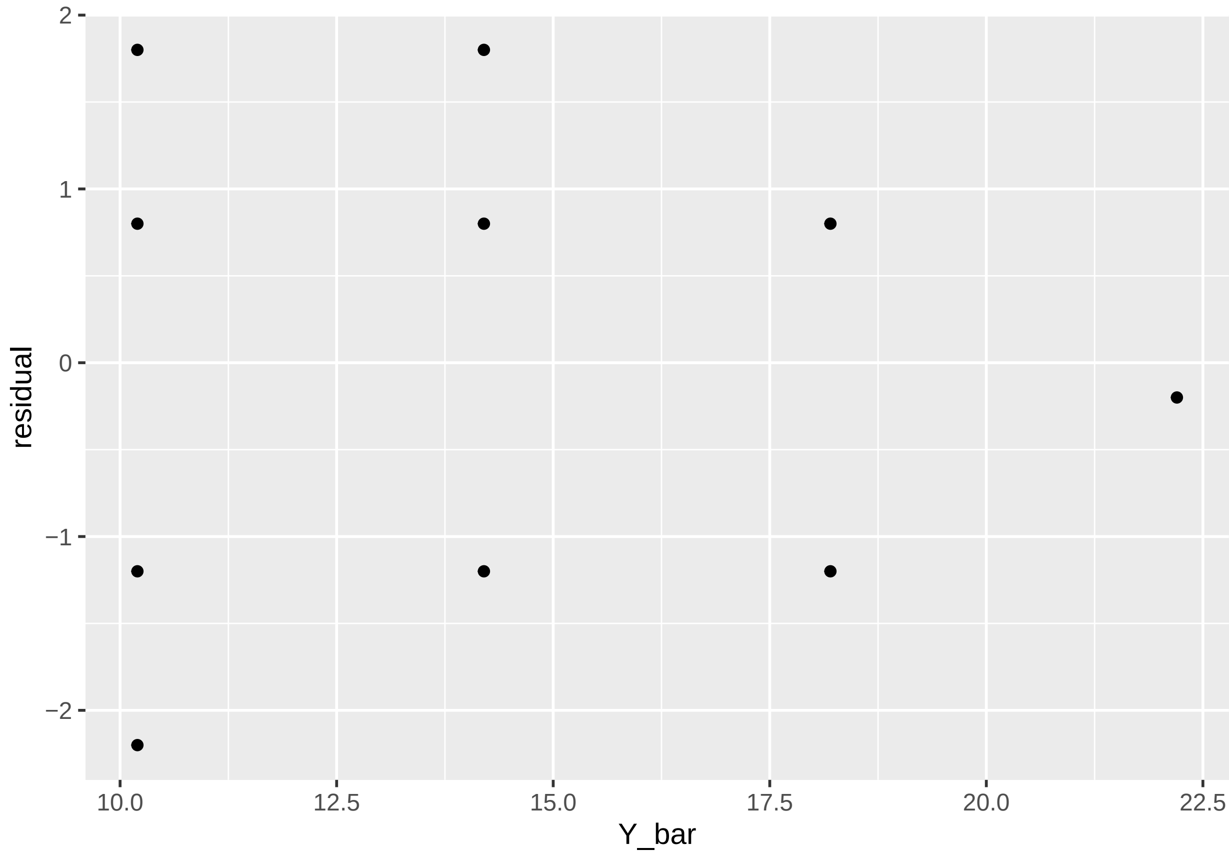
```r
library(ggplot2)
airfreight <- read.table("airfreight.txt")
airfreight <- airfreight[-1,]
airfreight <- airfreight %>% as.data.frame() %>%
  rename(transfer = V1, arrival = V2) %>%
  mutate_if(is.factor, as.character) %>%
  mutate_if(is.character, as.numeric)

#a

X <- cbind(rep(1,10),airfreight$transfer)
Y <- airfreight$arrival
h <- X %*% solve(t(X) %*% X) %*% t(X)
Sxx <- sum((X[,2] - mean(X[,2])) ^ 2)
Sxy <- sum((X[,2] - mean(X[,2])) * (Y - mean(Y)))
beta1_hat <- Sxy/Sxx                                      #estimator for beta1
beta0_hat <- mean(Y) - beta1_hat * mean(X[,2])            #estimator for beta0

ggplot(data = airfreight, aes(x = transfer, y = arrival)) + geom_point() +
  stat_function(fun = function(x) 10.2 + 4 * x) +
  labs(title = "Intercept =4, Slope = 10.2")
```




```r
#b

sigma_hat <- t(Y) %*% (diag(1,10,10) - h) %*% Y/8
t <- qt(.05/2,8,lower.tail = FALSE)
CI <- rbind(beta0_hat,beta1_hat)
CI <- CI %>% as.data.frame() %>%
  rename(parameter = V1) %>%
  mutate( uppertail = parameter +
           t * rbind(sqrt(sigma_hat * (1 / 10 + mean(X[,2]) ^ 2 / Sxx)), sqrt(sigma_hat/Sxx)),

          lowertail = parameter -
          t * rbind(sqrt(sigma_hat * (1 / 10 + mean(X[,2]) ^ 2 / Sxx)), sqrt(sigma_hat/Sxx)))
CI  #confidence interval
```

```
##   parameter uppertail lowertail
## 1      10.2 11.729630  8.670370
## 2       4.0  5.081612  2.918388
```

```r
#c

residual <- Y - X %*% rbind(beta0_hat,beta1_hat)
residual <- as.data.frame(cbind(residual,X %*% rbind(beta0_hat,beta1_hat)))
colnames(residual) <- c("residual","Y_bar")
ggplot(data = residual, aes(x=Y_bar,y=residual)) + geom_point() + geom_abline(intercept = 0)
```



Problem 3

bootstrap is asymptotically more consistent than the standard intervals obtained using assumptions of normality.

bootstrap automatically makes assumption of independence.

```
beta1 <- rep(0,1000)
for (i in 1:1000) {
  dat_boostrap <- airfreight[sample(nrow(airfreight),20,replace = TRUE),]
  X1 <- cbind(rep(1,20),dat_boostrap$transfer)
  Y1 <- dat_boostrap$arrival
  beta1[i] <- c(0,1) %*% solve(t(X1) %*% X1) %*% t(X1) %*% Y1
}
beta1_boostrap <- sum(beta1)/1000

beta1_boostrap          #estimator for beta1 by using bootstrap
```

```
## [1] 4.026573
```

```
quantile(beta1, probs = c(0.025,0.975))
```

```
##     2.5%    97.5%
## 3.463538 4.625128
```

Problem 4

```r
for (i in 1:5) {  #do BF test 5 times
  r1 <- sample(residuals(lmfit),5,replace = FALSE)
  r2 <- setdiff(residuals(lmfit),r1)
  n1 <- 5
  n2 <- 5
  r1nod <- median(r1)
  r2nod <- median(r2)
  d1 <- abs(r1 - r1nod)
  d2 <- abs(r2 - r2nod)
  s <-(sum((d1 - mean(d1))^2) + sum((d2 - mean(d2))^2))/(10-2)
  t_bf <- (mean(d1) - mean(d2))/sqrt((1/n1 + 1/n2) * s)
  # t distribution with df 5+5-2=8

  print(pt(t_bf, 8, lower.tail = FALSE, log.p = FALSE))
  # P-value is .159, fail to reject constant variance
}
```

```
## [1] 0.8025373
## [1] 0.7515058
## [1] 0.3585945
## [1] 0.4829292
## [1] 0.4514695
```

```r
ssr_star <- sum((x %*% solve(t(x) %*% x) %*% t(x) %*% (r^2) - mean(r^2))^2)

sse <- sum((x %*% solve(t(x) %*% x) %*% t(x) %*% y - y)^2)
(ssr_star/2) / ((sse/10)^2)
```

```
## [1] 1.277558
```

```r
pchisq((ssr_star/2) / ((sse/10)^2), 1, ncp = 0, lower.tail = F)
```

```
## [1] 0.2583537
```

```r
# p-value, fail to reject constant variance assumptiom
```