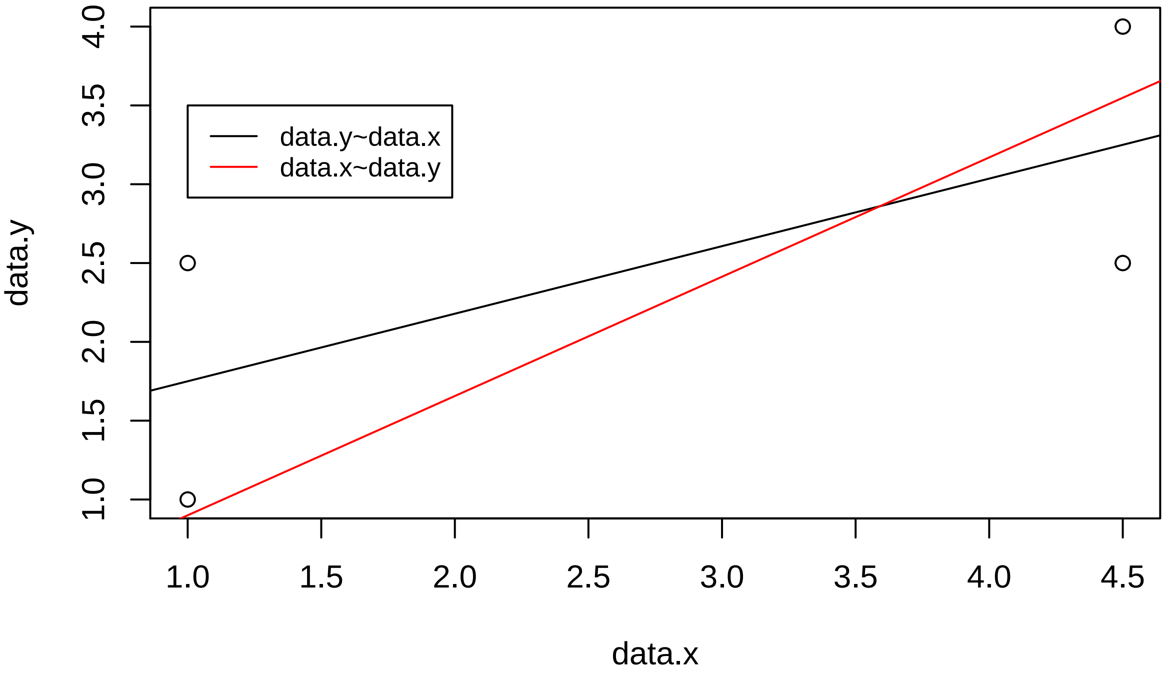
Exercise 1 solutions 2019-03-08

1.)

a.)

Q: In the plot below, draw the regression line for Y being the dependent and X being the independent variable and vice versa.

```r
plot(data.x, data.y)
abline(a=coeff[1], b=coeff[2])
abline(a=-coeff2[1]/coeff2[2], b=1/coeff[1], col="red")
legend(1, 3.5, legend=c("data.y~data.x", "data.x~data.y"),
       col=c("black", "red"), lty=1, cex=0.8)
```



b)

Q: In the plot below, we depicted for several farms the yearly income in Dollars versus the number of cows.

(i)

Q: Give the approximate equation for the least squares line.

Let the least square line be $\hat{y} = \hat{b_0} + \hat{b_1} * x$. From the graph one reads

$$\hat{b_0} = 675 \text{ USD}$$

$$\hat{b_1} = \frac{119 \text{ USD}}{6 \text{ cows}} = 19.83 \frac{\text{USD}}{\text{cows}}$$

Hence it follows for the least square equation approximately:

$$\hat{y} = 675 + 19.83 * x$$

where $\hat{y}$ is in units of USD, and x is in units of cows.

(ii)

Q: What is your estimate for the average deviation of the points with respect to the regression line?

The average deviation from the point with respect to the regression line vertically is around 0.

(iii)

Q: Estimate the income of a farm with 15 cows and of a farm with 100 cows? How meaningful are these estimates?

For 15 cows the estimated income is 972 USD. For 100 cows the estimated income is 2658. The value for 15 cows is more meaningful, since the data is only presented for farms between 0 and 18 cows. The datapoints (which were used to fit the regression line) are far from the 100-cow-farm, hence the extrapolation should be considered carefully and the estimated income for a 100 cow-farm is far less meaningful compared to the 15-cow-farm.

c)

Q: Show that sum of errors = 0 for any line that passes through the point of averages.

For any line it is $y_i = b_0 + b_1 * x_i + e_i$ where $i = 1, ..., n$

With the definition for the averages ($\bar{x} = \frac{1}{n}\sum_{i=1}^{n}(x_i)$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n}(y_i)$) follows:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n}(b_0 + b_1 * x_i + e_i)$$

$$= \frac{1}{n}\sum_{i=1}^{n}(b_0 + b_1 * x_i) + \frac{1}{n}\sum_{i=1}^{n}(e_i)$$

$$= \frac{1}{n}([b_0 + b_1 * x_1] + [b_0 + b_1 * x_2] + ... + [b_0 + b_1 * x_n]) + \frac{1}{n}\sum_{i=1}^{n}(e_i)$$

$$= \frac{1}{n} * b_0 * n + \frac{1}{n} * b_1 * \sum_{i=1}^{n}(x_i) + \sum_{i=1}^{n}(e_i)$$

$$= b_0 + b_1 * \bar{x} + \sum_{i=1}^{n}(e_i)$$

The first part before the sum must be by construction the average point $\bar{y} = b_0 + b_1 * \bar{x}$

$$= \bar{y} + \sum_{i=1}^{n}(e_i)$$

From this follows $\sum_{i=1}^{n}(e_i) = 0$

2.)

a)

Q: Compute the predicted value of y for x = 4.

```
b0 <- 2.1783
b1 <- 1.8232
res <- b0+b1*log(4)
res
```

```
## [1] 4.705792
```

The expected value for $x = 4$ is 4.705792.

b)

Q: If we compare two observations i and j where xi = 2xj , then the fitted value yˆi compared to yˆj is increased by a value . Please fill in the blank

From $\hat{y_i} - \hat{y_j} = (\hat{\beta}_0 + \hat{\beta}_1 * log(x_i)) - (\hat{\beta}_0 + \hat{\beta}_1 * log(x_j))$ and $x_i = 2 * x_j$ follows that $\hat{y_i} - \hat{y_j} = \hat{\beta}_1 * log(2)$, hence the fitted value is increased by a value of $log(2) * \hat{\beta}_1 = 1.263746$.

c)

Q: Compute the predicted value of y for x = 3.

```
b0 <- 1.12022
b1 <- 0.95966
res.log <- b0+b1*(3)
res <- exp(res.log)
res
```

```
## [1] 54.55449
```

The expected value for $x = 3$ is 54.55449.

d)

Q: If we compare two observations i and j where xi = xj + 1, then the fitted value yˆi compared to yˆj is multiplied by a value . Please fill in the blank.

Similar to b), but this time it is $log(\hat{y_i}) = \hat{\beta}_0 + \hat{\beta}_1 * x_i$ and $log(\hat{y_j}) = \hat{\beta}_0 + \hat{\beta}_1 * x_j$.

So it follows:

$$log(\hat{y_i}) - log(\hat{y_j}) = \hat{\beta}_1 * (x_i - x_j)$$

$$log(\hat{y_i}/\hat{y_j}) = \hat{\beta}_1 * (x_i - x_j)$$

Applying the exponential function $exp(\cdot)$ yields

$$\hat{y_i}/\hat{y_j} = exp(\hat{\beta}_1 * (x_i - x_j))$$

Since it is $x_i = x_j + 1$ follows this time:

$$\hat{y_i} = \hat{y_j} * exp(\hat{\beta_1})$$

The fitted value is multiplied by a value of 2.610809.

3)

a)

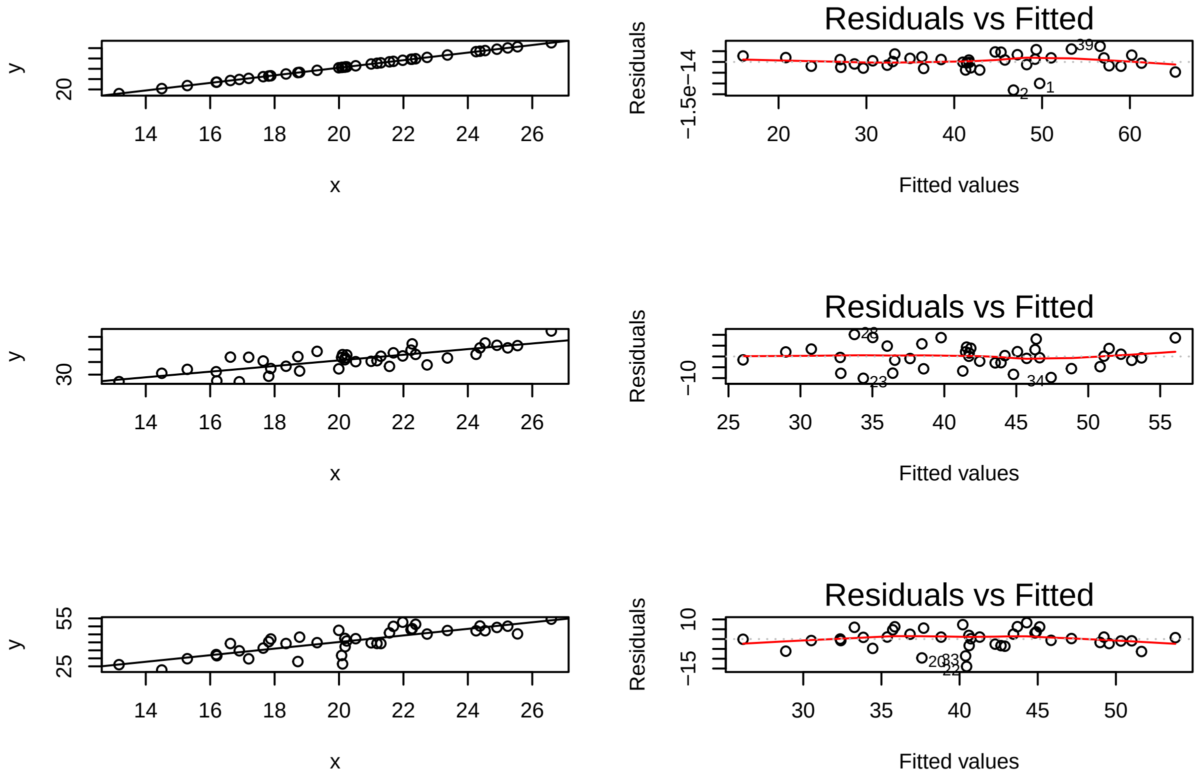
Q: Write a sequence of R-commands which randomly generates 100 times a vector. . . .

```
# simulation
set.seed(21)                          # initializes the random number generator
x <- rnorm(40, 20, 3)                 # generates x-values
nsim <- 100
hatbeta0 <- rep(NA, nsim)             # vector to store estimated beta_0 values
hatbeta1 <- rep(NA, nsim)             # vector to store estimated beta_0 values

for(i in 1:nsim){
  y <- 1 + 2 * x + 5 * rnorm(length(x)) # y-values = linear function(x) + error
  fit <- lm(y~x)              # fit linear regression
  hatbeta0[i] <- fit$coef[1]  # store estimated intercept
  hatbeta1[i] <- fit$coef[2]  # store estimated slope
}
```

b)

Q: For the first three generated y-vectors, plot y against x, and add the fitted regression line and construct the corresponding Tukey-Anscombe plot.

```
par(mfrow=c(3,2))     # plot 3x2
set.seed(21)   # same seed
for(i in 1:3){
  y <- 1 + 2 * x + 5 * rnorm(length(x)) # y-values = linear function(x) + error
  fit.new <- lm(y~x)              # fit linear regression
  plot(y~x)
  abline(fit.new)
  plot(fit.new, which=1)   # Tukey-Anscombe

}
```



c)

Q: Compute the empirical mean and standard deviation of the estimated slopes.

```
mean(hatbeta1)
```

```
## [1] 2.011819
```

```
sd(hatbeta1)
```

```
## [1] 0.2430534
```

d)

Q: Compute the theoretical variance of $\hat{\beta}_1$.

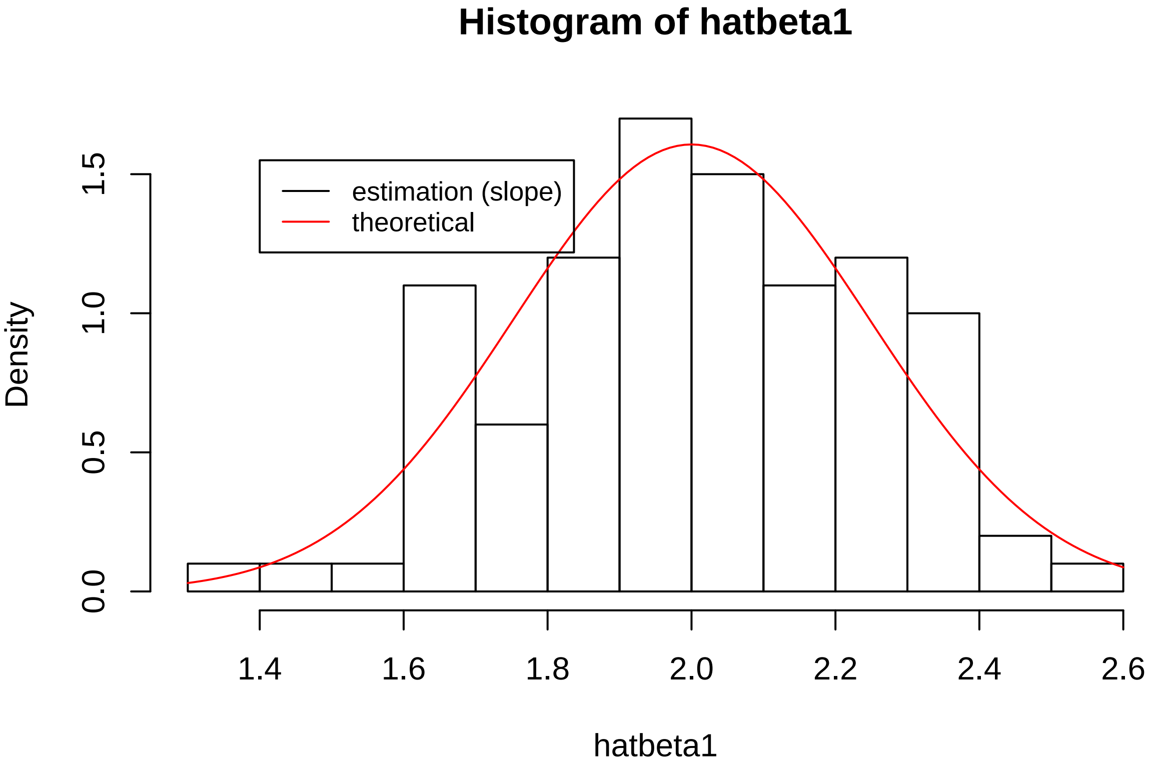
```
X = cbind(1,x)                            # design matrix
XtX.inv <- solve(t(X) %*% X)              # (X^T X)^{-1}
betahat.theo.var <- 5^2 * XtX.inv[2,2]
betahat.theo.var
```

```
## [1] 0.06164328
```

e)

Q: Draw a histogram of the 100 estimated slopes and add the normal density of the theoretical distribution of $\hat{\beta}_1$ to the histogram. What do you observe? Does it fit well?

```r
par(mfrow=c(1,1))      # plot 1x1
hist(hatbeta1, breaks=10, freq=FALSE)
betahat.theo.sd <- sqrt(betahat.theo.var)
betahat.theo <- 2  # from the model definition 1 + 2 * x + 5 * rnorm(length(x))
lines(seq(1.3, 2.6, by = 0.01), col="red",
      dnorm(seq(1.3, 2.6, by = 0.01), mean=betahat.theo , sd=betahat.theo.sd))
legend(1.4, 1.55, legend=c("estimation (slope)", "theoretical"),
       col=c("black", "red"), lty=1:1, cex=0.8)
```



Both, the estimated histogram and the theoretical curve are skewed to the left. The distribution of the estimated slopes (hatbeta1) follows the theoretical distribution well.

4)

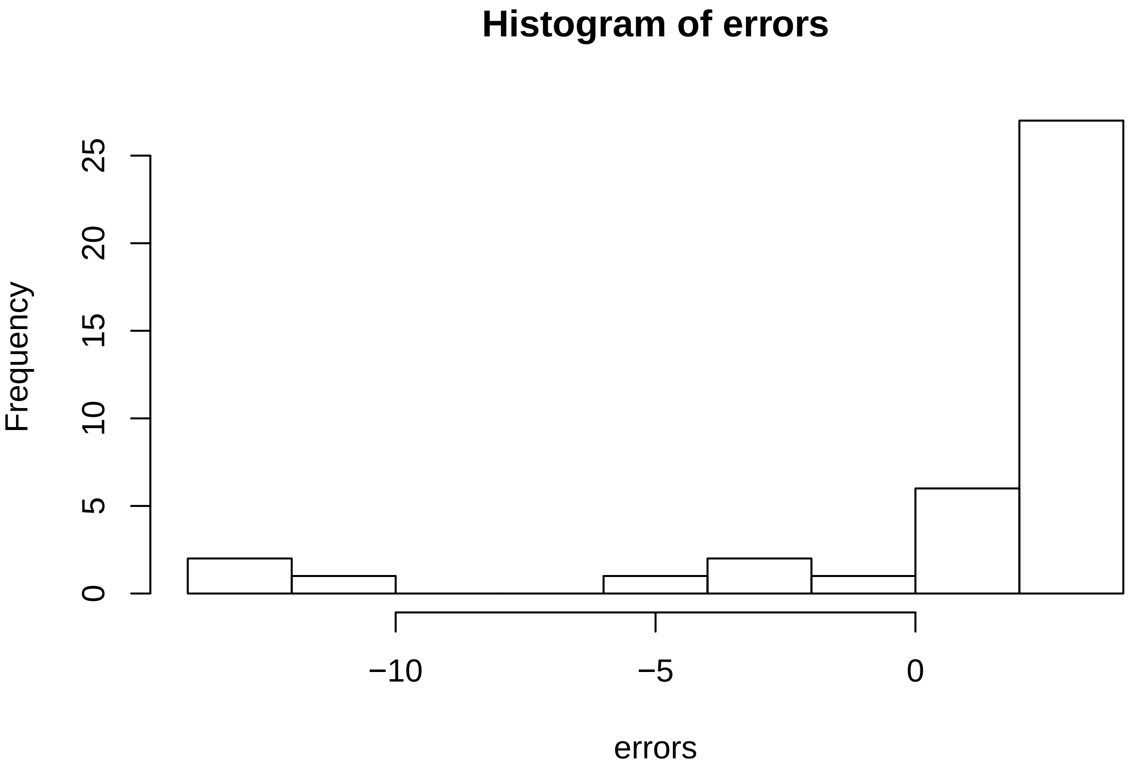
Q: Which (if any) assumptions are violated? What properties of the distribution of $\hat{\beta}_1$ are affected by this? Which part of the R output do you still trust?

a)

$y < -1 + 2 * x + 5 * (1 - rchisq(length(x), df = 1))/sqrt(2)$

```r
# simulation
set.seed(21)                          # initializes the random number generator
x <- rnorm(40, 20, 3)                 # generates x-values
nsim <- 100
hatbeta0 <- rep(NA, nsim)             # vector to store estimated beta_0 values
hatbeta1 <- rep(NA, nsim)             # vector to store estimated beta_0 values
```

```
for(i in 1:nsim){
  y <- 1 + 2 * x + 5 * (1 - rchisq(length(x), df = 1)) / sqrt(2)
  fit <- lm(y~x)              # fit linear regression
  hatbeta0[i] <- fit$coef[1]  # store estimated intercept
  hatbeta1[i] <- fit$coef[2]  # store estimated slope
}

par(mfrow=c(1,1))     # plot 3x2
errors <- 5 * (1 - rchisq(40, df = 1)) / sqrt(2)
hist(errors)
```



```
mean(errors)
```

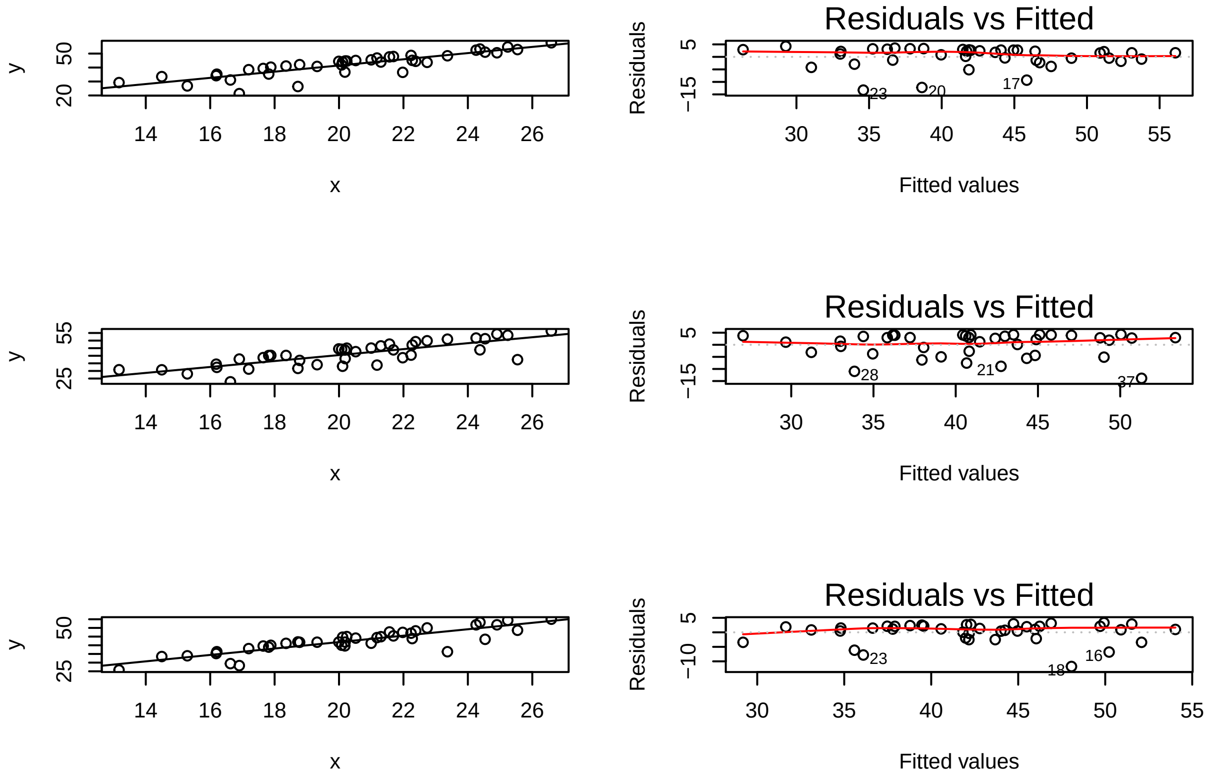
```
## [1] 0.9146039
```

```
var(errors)
```

```
## [1] 19.30535
```

As the histogram shows, the assumption that is violated is that $E(e_i) = 0$, so there **is** systematic error.

```
par(mfrow=c(3,2))     # plot 3x2
set.seed(21)    # same seed
for(i in 1:3){
  y <- 1 + 2 * x + 5 * (1 - rchisq(length(x), df = 1)) / sqrt(2)
  fit.new <- lm(y~x)                # fit linear regression
  plot(y~x)
  abline(fit.new)
  plot(fit.new, which=1)    # Tukey-Anscombe

}
```



```
mean(hatbeta1)
```

```
## [1] 1.994459
```

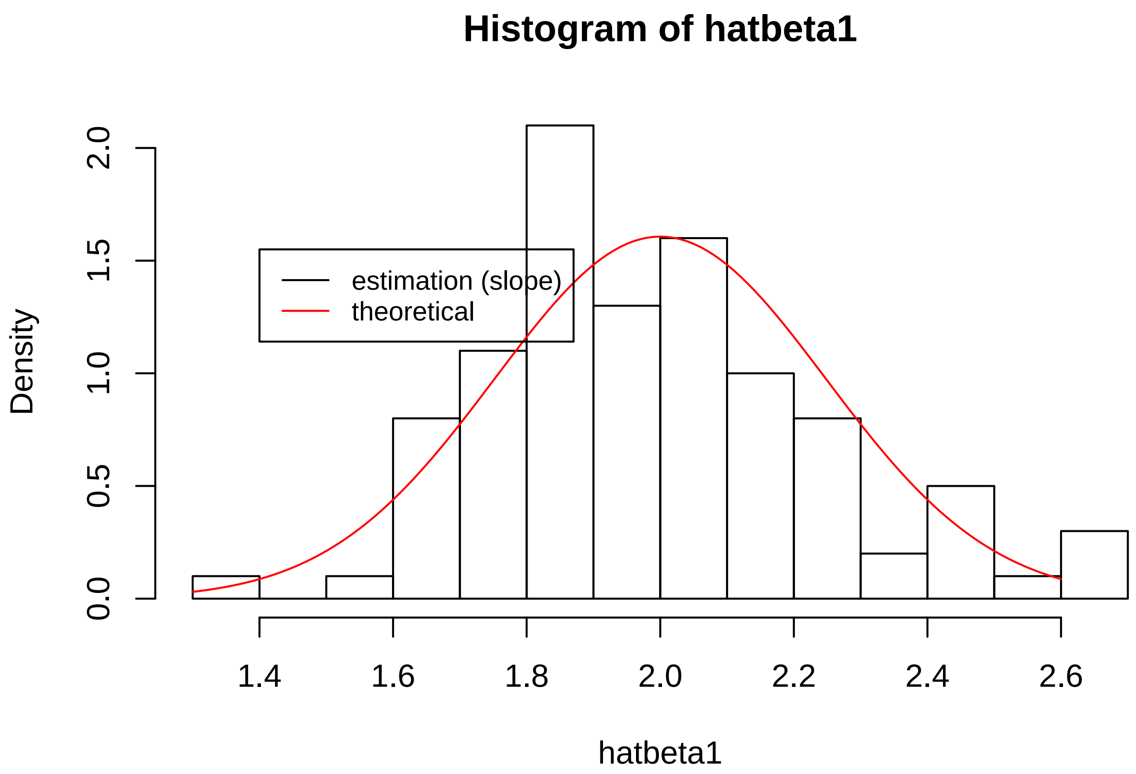
```
sd(hatbeta1)
```

```
## [1] 0.249094
```

```
X = cbind(1,x)                                  # design matrix
XtX.inv <- solve(t(X) %*% X)                    # (X^T X)^{-1}
betahat.theo.var <- 5^2 * XtX.inv[2,2]
betahat.theo.var
```

```
## [1] 0.06164328
```

```
par(mfrow=c(1,1))      # plot 1x1
hist(hatbeta1, breaks=10, freq=FALSE)
betahat.theo.sd <- sqrt(betahat.theo.var)
betahat.theo <- 2  # from the model definition 1 + 2 * x + 5 * rnorm(length(x))
lines(seq(1.3, 2.6, by = 0.01), col="red", dnorm(seq(1.3, 2.6, by = 0.01), mean=betahat.theo , sd=betaha
legend(1.4, 1.55, legend=c("estimation (slope)", "theoretical"), col=c("black", "red"), lty=1:1, cex=0.8
```

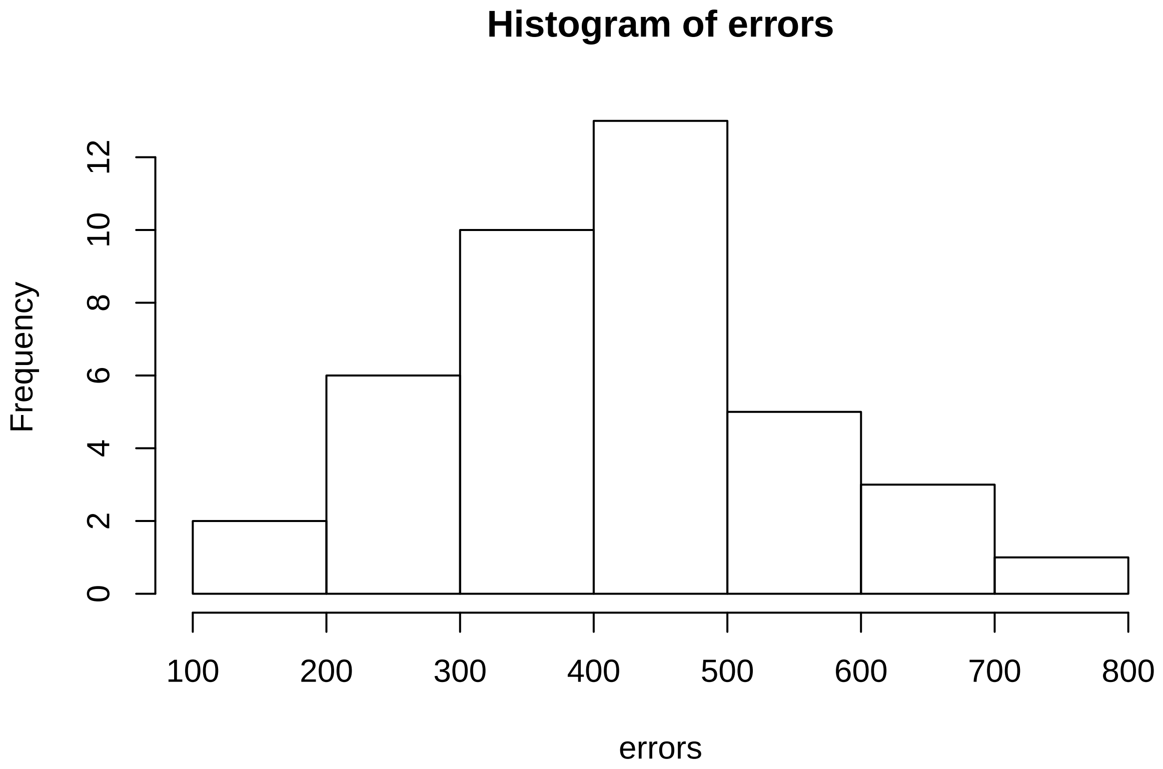


The distribution of the estimated slopes (hatbeta1) follows the theoretical distribution well.

b)

$y < -1 + 2 * x + 5 * rnorm(length(x), mean = x^2/5 - 1, sd = 1)$

```
# simulation
set.seed(21)                          # initializes the random number generator
x <- rnorm(40, 20, 3)                 # generates x-values
nsim <- 100
hatbeta0 <- rep(NA, nsim)             # vector to store estimated beta_0 values
hatbeta1 <- rep(NA, nsim)             # vector to store estimated beta_0 values

for(i in 1:nsim){
  y <- 1 + 2 * x + 5 * rnorm(length(x), mean = x^2 / 5 - 1, sd = 1)
  fit <- lm(y~x)              # fit linear regression
  hatbeta0[i] <- fit$coef[1]  # store estimated intercept
  hatbeta1[i] <- fit$coef[2]  # store estimated slope
}

par(mfrow=c(1,1))      # plot 3x2
errors <- 5 * rnorm(length(x), mean = x^2 / 5 - 1, sd = 1)
hist(errors)
```



```
mean(errors)
```

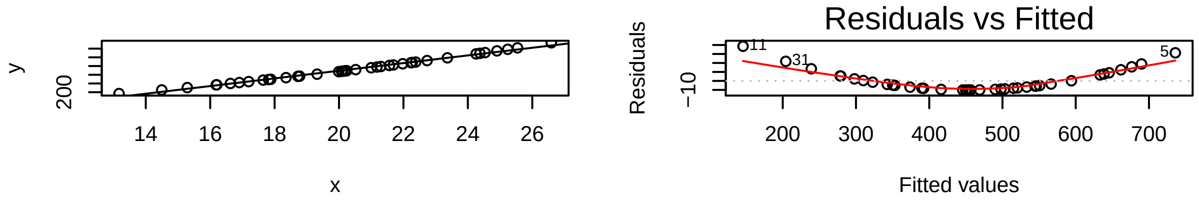
```
## [1] 418.8571
```

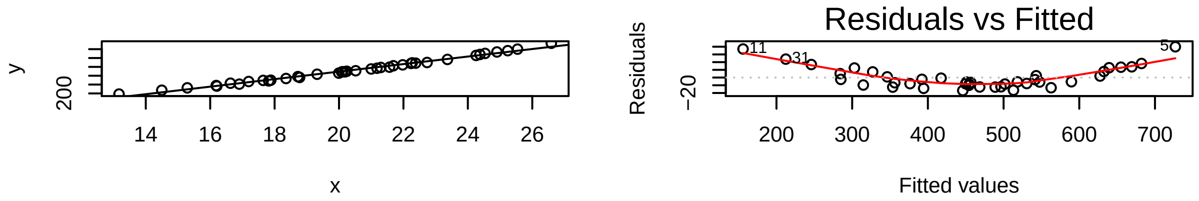
```
var(errors)
```

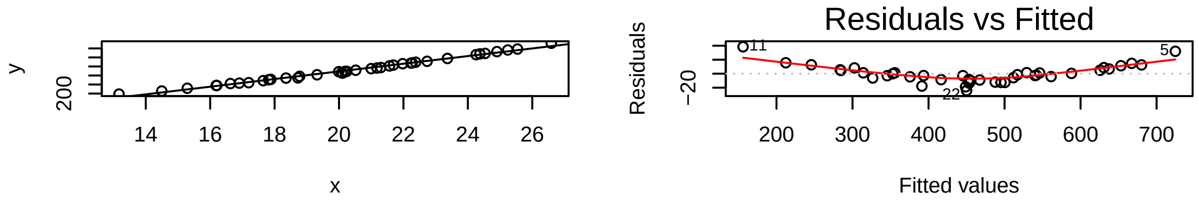
```
## [1] 17122.38
```

As the histogram shows, the assumption that is violated is that $E(e_i) = 0$ for $i = 1, ..., n$, so there **is** systematic error. Also the assumption $Var(e_i) = \sigma^2$ for $i = 1, ..., n$ is violated.

```
par(mfrow=c(3,2))      # plot 3x2
set.seed(21)    # same seed
for(i in 1:3){
  y <- 1 + 2 * x + 5 * rnorm(length(x), mean = x^2 / 5 - 1, sd = 1)
  fit.new <- lm(y~x)                # fit linear regression
  plot(y~x)
  abline(fit.new)
  plot(fit.new, which=1)    # Tukey-Anscombe

}
```







Low and high y-value points tend to lie above the fitted curve, whereas points in the middle region tend to fall below the fitted curve

```
mean(hatbeta1)
```

```
## [1] 42.31647
```

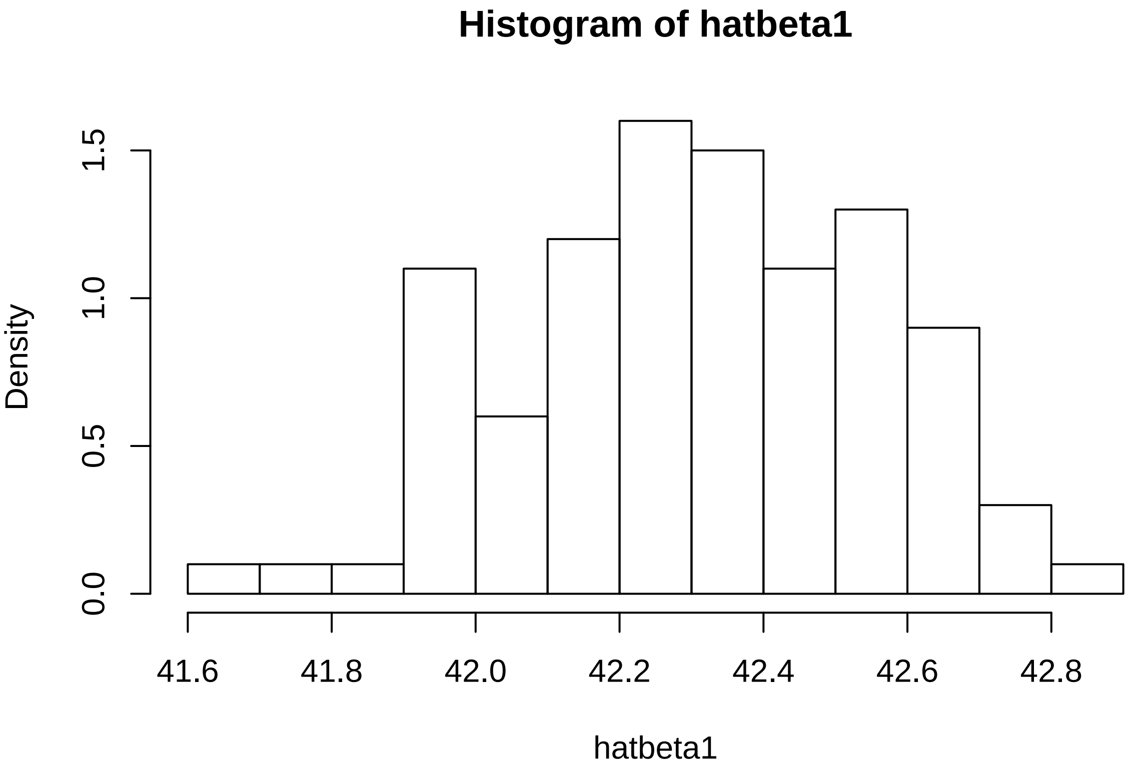
```
sd(hatbeta1)
```

```
## [1] 0.2430534
```

```
X = cbind(1,x)                          # design matrix
XtX.inv <- solve(t(X) %*% X)            # (X^T X)^{-1}
betahat.theo.var <- 5^2 * XtX.inv[2,2]
betahat.theo.var
```

```
## [1] 0.06164328
```

```
par(mfrow=c(1,1))     # plot 1x1
hist(hatbeta1, breaks=10, freq=FALSE)
betahat.theo.sd <- sqrt(betahat.theo.var)
betahat.theo <- 2  # from the model definition 1 + 2 * x + 5 * rnorm(length(x))
lines(seq(1.3, 2.6, by = 0.01), col="red",
      dnorm(seq(1.3, 2.6, by = 0.01), mean=betahat.theo , sd=betahat.theo.sd))
legend(1.4, 1.55, legend=c("estimation (slope)", "theoretical"),
       col=c("black", "red"), lty=1:1, cex=0.8)
```



The distribution of the estimated slopes (hatbeta1) is completely off with respect to the theoretical distribution.

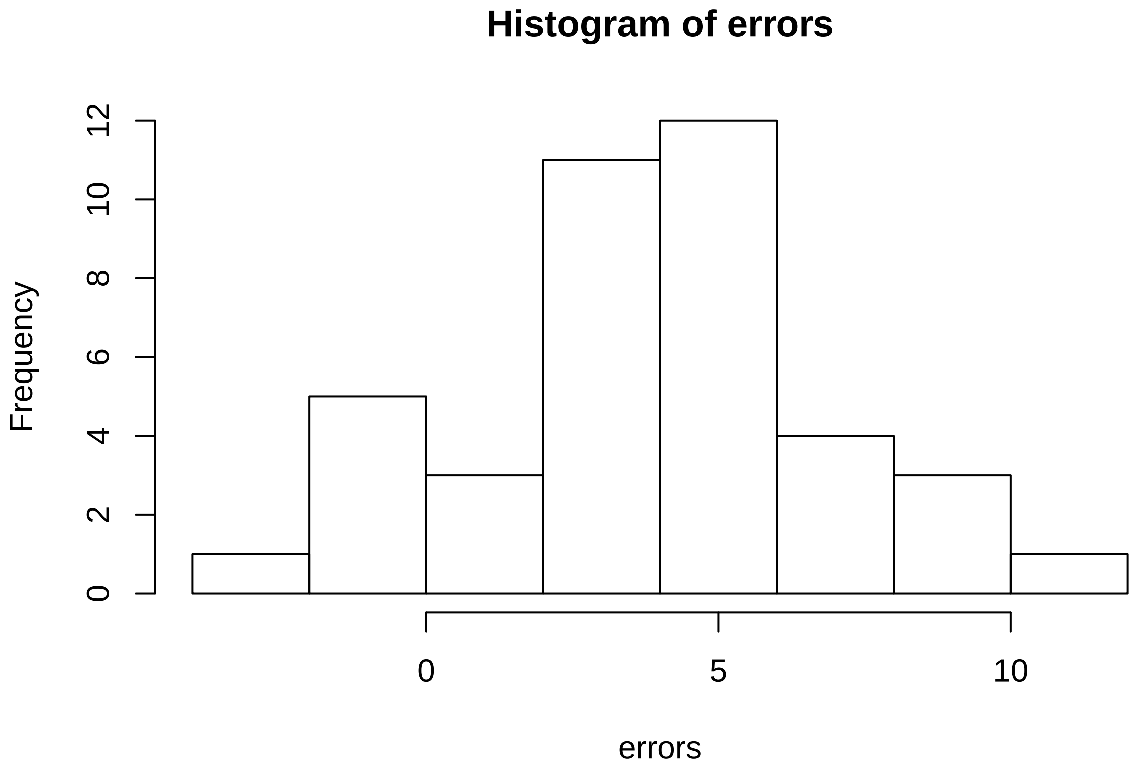
c)

$y < -1 + 2 * x + 5 * mvrnorm(n = 1, mu = rep(0, length(x)), Sigma = Sigma)$

```
# simulation
set.seed(21)                          # initializes the random number generator
x <- rnorm(40, 20, 3)                 # generates x-values
nsim <- 100
hatbeta0 <- rep(NA, nsim)             # vector to store estimated beta_0 values
hatbeta1 <- rep(NA, nsim)             # vector to store estimated beta_0 values

for(i in 1:nsim){
  require(MASS)
  Sigma <- matrix(0.7,40,40)
  diag(Sigma) <- 1
  y   <- 1 + 2 * x + 5 * mvrnorm(n = 1, mu = rep(0, length(x)), Sigma = Sigma)
  fit <- lm(y~x)                  # fit linear regression
  hatbeta0[i] <- fit$coef[1]  # store estimated intercept
  hatbeta1[i] <- fit$coef[2]  # store estimated slope
}
```

```
## Loading required package: MASS
```

```
par(mfrow=c(1,1))     # plot 3x2
errors <- 5 * mvrnorm(n = 1, mu = rep(0, length(x)), Sigma = Sigma)
hist(errors)
```



```
mean(errors)
```

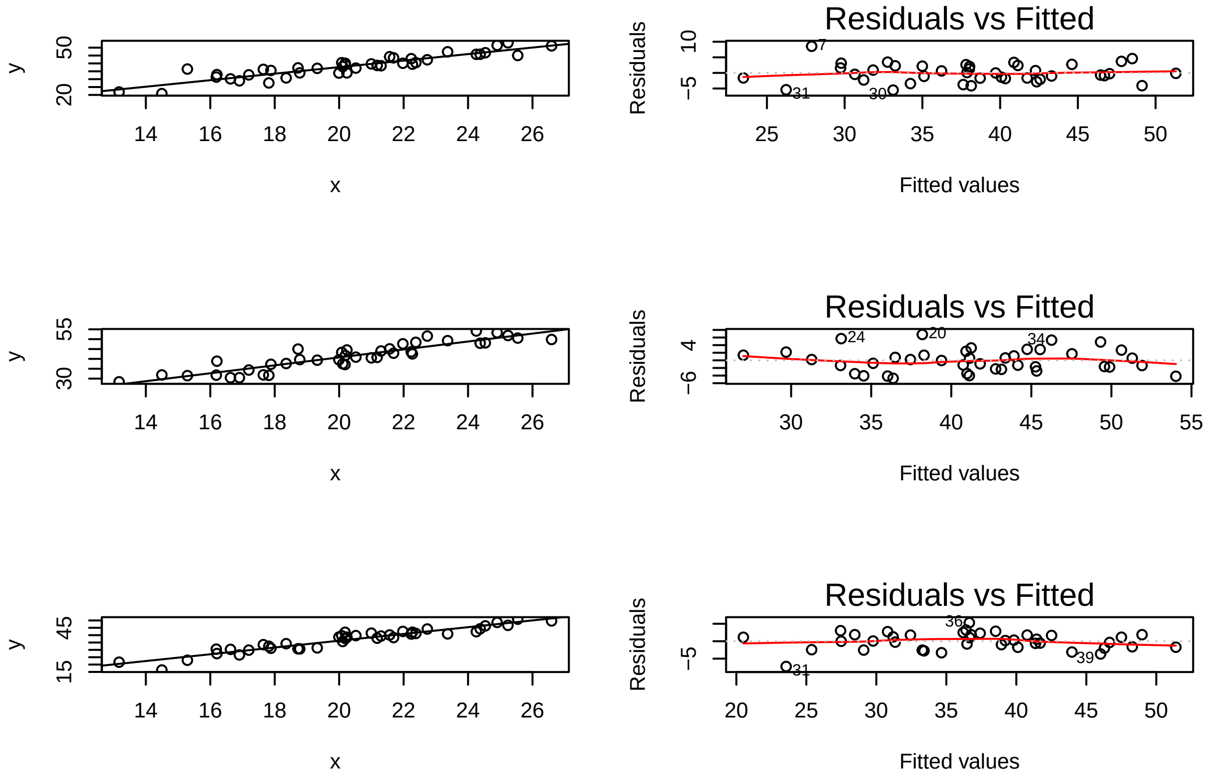
```
## [1] 3.763886
```

```
var(errors)
```

```
## [1] 9.561033
```

As the histogram shows, the assumption that is violated is that $E(e_i) = 0$ for $i = 1, ..., n$, so there **is** systematic error. Also the assumption $Var(e_i) = \sigma^2$ for $i = 1, ..., n$ is violated.

```
par(mfrow=c(3,2))      # plot 3x2
set.seed(21)    # same seed
for(i in 1:3){
  require(MASS)
  Sigma <- matrix(0.7,40,40)
  diag(Sigma) <- 1
  y   <- 1 + 2 * x + 5 * mvrnorm(n = 1, mu = rep(0, length(x)), Sigma = Sigma)
  fit.new <- lm(y~x)                # fit linear regression
  plot(y~x)
  abline(fit.new)
  plot(fit.new, which=1)    # Tukey-Anscombe

}
```



```r
mean(hatbeta1)
```

```
## [1] 1.965306
```

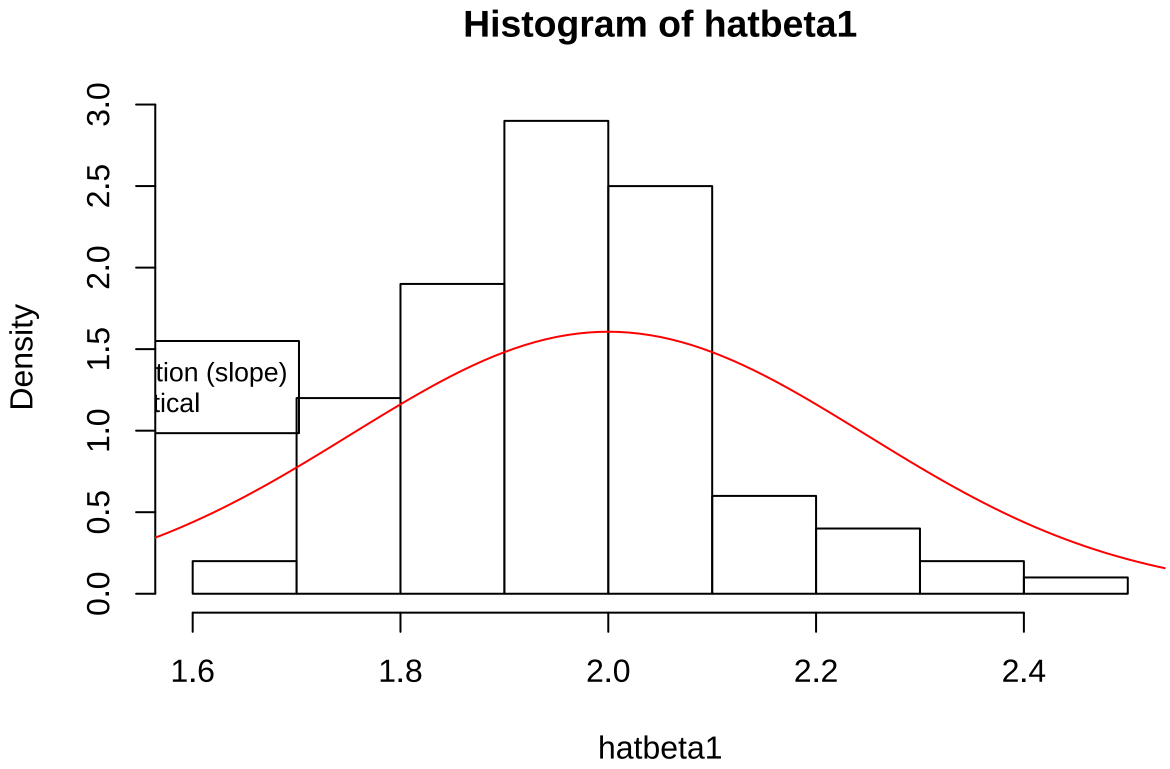
```r
sd(hatbeta1)
```

```
## [1] 0.1458017
```

```r
X = cbind(1,x)                          # design matrix
XtX.inv <- solve(t(X) %*% X)            # (X^T X)^{-1}
betahat.theo.var <- 5^2 * XtX.inv[2,2]
betahat.theo.var
```

```
## [1] 0.06164328
```

```r
par(mfrow=c(1,1))      # plot 1x1
hist(hatbeta1, breaks=10, freq=FALSE)
betahat.theo.sd <- sqrt(betahat.theo.var)
betahat.theo <- 2  # from the model definition 1 + 2 * x + 5 * rnorm(length(x))
lines(seq(1.3, 2.6, by = 0.01), col="red",
      dnorm(seq(1.3, 2.6, by = 0.01), mean=betahat.theo , sd=betahat.theo.sd))
legend(1.4, 1.55, legend=c("estimation (slope)", "theoretical"),
       col=c("black", "red"), lty=1:1, cex=0.8)
```

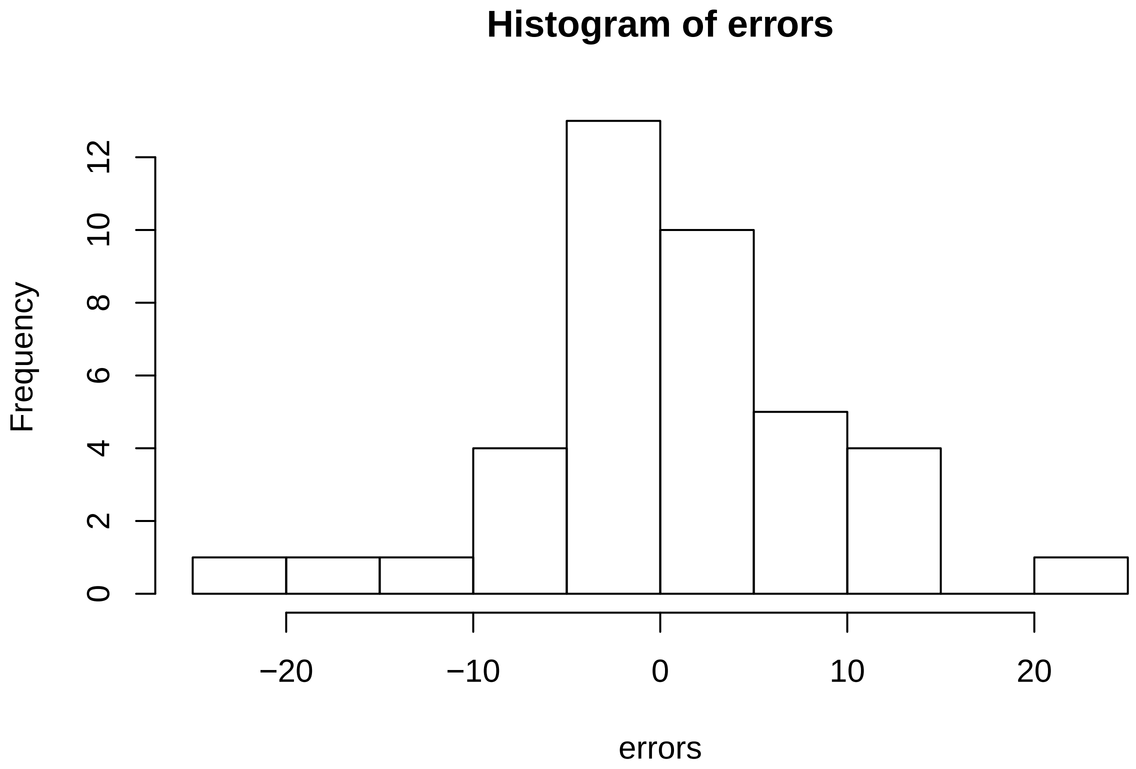


The distribution of the estimated slopes (hatbeta1) is skewed to the left with respect to the theoretical distribution.

d)

$y < -1 + 2 * x + 5 * mvrnorm(n = 1, mu = rep(0, length(x)), Sigma = Sigma)$

```
# simulation
set.seed(21)                          # initializes the random number generator
x <- rnorm(40, 20, 3)                 # generates x-values
nsim <- 100
hatbeta0 <- rep(NA, nsim)             # vector to store estimated beta_0 values
hatbeta1 <- rep(NA, nsim)             # vector to store estimated beta_0 values

for(i in 1:nsim){
  y <- 1 + 2 * x + 5 * rnorm(length(x), mean = 0, sd = (x-15)^2 / 30)
  fit <- lm(y~x)               # fit linear regression
  hatbeta0[i] <- fit$coef[1]  # store estimated intercept
  hatbeta1[i] <- fit$coef[2]  # store estimated slope
}

par(mfrow=c(1,1))      # plot 3x2
errors <-  5 * rnorm(length(x), mean = 0, sd = (x-15)^2 / 30)
hist(errors)
```



```r
mean(errors)
```

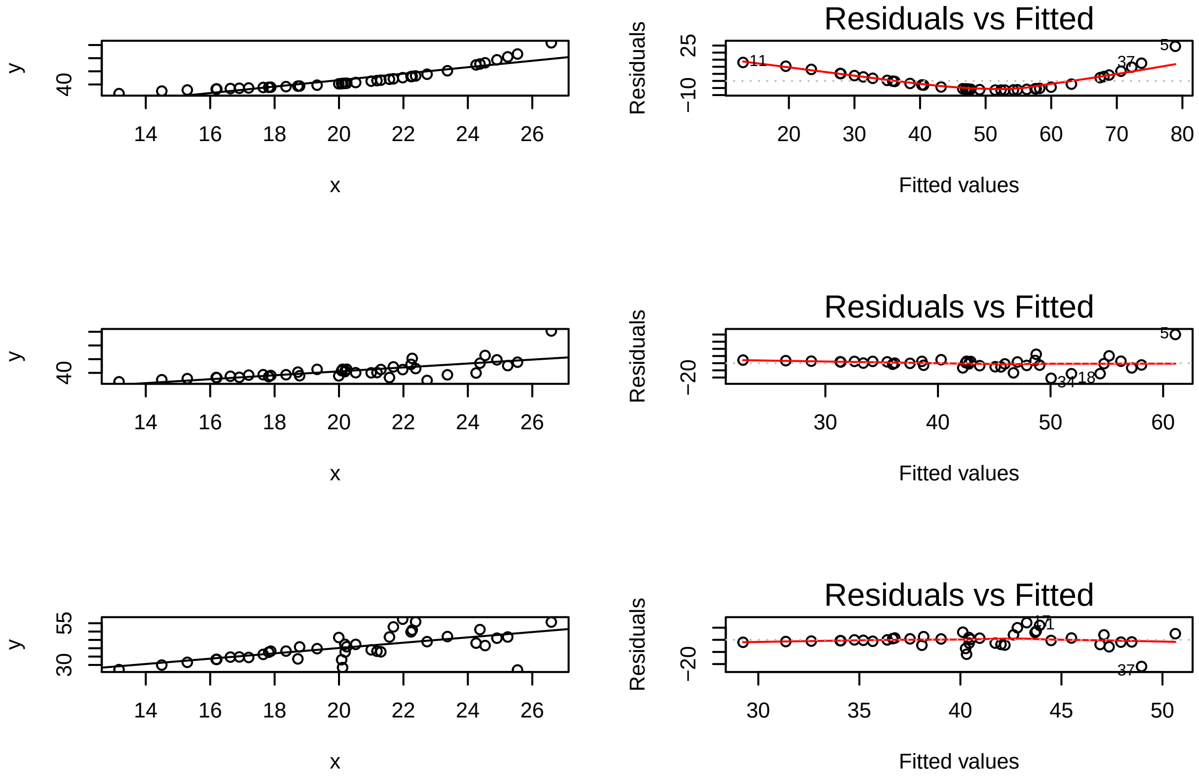
```
## [1] 0.6638852
```

```r
var(errors)
```

```
## [1] 65.71991
```

As the histogram shows, the assumption that is violated is that $E(e_i) = 0$ for $i = 1, ..., n$, so there **is** systematic error. Also the assumption $Var(e_i) = \sigma^2$ for $i = 1, ..., n$ is violated.

```r
par(mfrow=c(3,2))     # plot 3x2
set.seed(21)    # same seed
for(i in 1:3){
  require(MASS)
  Sigma <- matrix(0.7,40,40)
  diag(Sigma) <- 1
  y <- 1 + 2 * x + 5 * rnorm(length(x), mean = 0, sd = (x-15)^2 / 30)
  fit.new <- lm(y~x)                # fit linear regression
  plot(y~x)
  abline(fit.new)
  plot(fit.new, which=1)    # Tukey-Anscombe

}
```



```r
mean(hatbeta1)
```

```
## [1] 2.127485
```

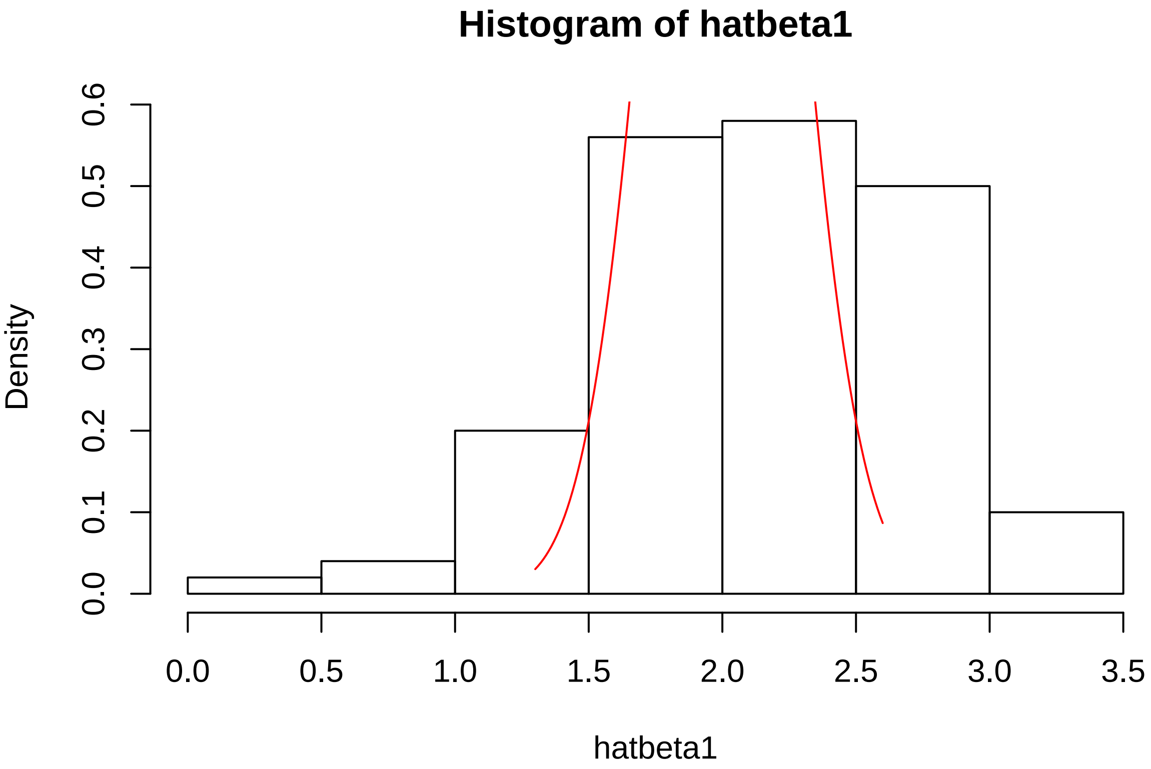
```r
sd(hatbeta1)
```

```
## [1] 0.5797268
```

```r
X = cbind(1,x)                              # design matrix
XtX.inv <- solve(t(X) %*% X)                # (X^T X)^{-1}
betahat.theo.var <- 5^2 * XtX.inv[2,2]
betahat.theo.var
```

```
## [1] 0.06164328
```

```r
par(mfrow=c(1,1))       # plot 1x1
hist(hatbeta1, breaks=10, freq=FALSE)
betahat.theo.sd <- sqrt(betahat.theo.var)
betahat.theo <- 2  # from the model definition 1 + 2 * x + 5 * rnorm(length(x))
lines(seq(1.3, 2.6, by = 0.01), col="red",
      dnorm(seq(1.3, 2.6, by = 0.01), mean=betahat.theo , sd=betahat.theo.sd))
legend(1.4, 1.55, legend=c("estimation (slope)", "theoretical"),
       col=c("black", "red"), lty=1:1, cex=0.8)
```



The distribution of the estimated slopes (hatbeta1) does not match at all with the theoretical distribution.