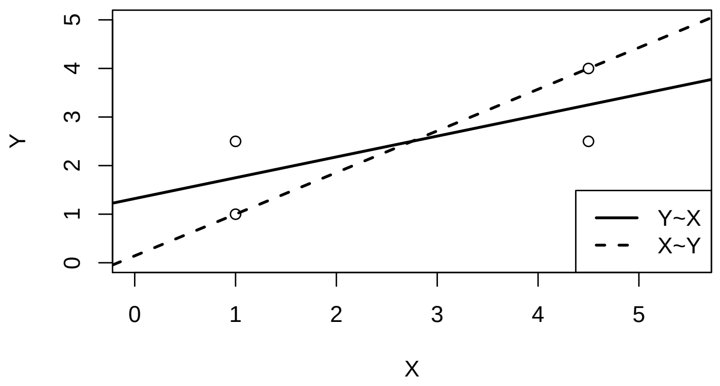
Solution to Series 1

1. **a)** One can see that for the LS-methods, it is important which variable is (in)dependent.



b)

(i) LS-line:

$$y \approx 695 + 20x \tag{1}$$

(ii) The average deviation of the points with respect to the regression line is the average value of the vertical distances of all points to the regression line. We just ask for an estimate, so the solution is not unique but should be in a reasonable range: around 100.

(iii) For a farm with $x = 15$ cows we estimate an income of $\hat{y} = 995$ using Equation (1). $x = 15$ lies in the range of the data, so the estimate is sensible.
For a farm with $x = 100$ cows we estimate an income of $\hat{y} = 2695$ using Equation (1). $x = 100$ lies far outside the range of the data, this estimate is dangerous! The linear relationship between x and y we have observed might only hold for the range of data that we actually observe.

c) Let $b_0 + b_1 x$ be any line that passes through the point of average $(\bar{x}, \bar{y})$. That is,

$$\bar{y} = b_0 + b_1 \bar{x}. \tag{2}$$

Therefore, we have

$$\begin{aligned}
\sum_{i=1}^{n} e_i &= \sum_{i=1}^{n} \Big(y_i - (b_0 + b_1 x_i)\Big) \\
&= \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} b_0 - \sum_{i=1}^{n} b_1 x_i \\
&= n\bar{y} - nb_0 - nb_1\bar{x} \\
&= n(\bar{y} - b_0 - b_1\bar{x}) \\
&= n(b_0 + b_1\bar{x} - b_0 - b_1\bar{x}) = 0,
\end{aligned}$$

where we used (2) to obtain the last equality.
Remark: One can also show that the LS line always passes through the point of averages. This implies that the residuals of LS regression always have exactly mean zero.

2. **a)** $y_{\text{predict}} = \hat{\beta}_0 + \hat{\beta}_1 * \log(4) = 2.1783 + 1.8232 * \log(4) \approx 4.71.$

b) Because $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 \log(x_i)$, we have

$$\begin{aligned}
\hat{y}_i - \hat{y}_j &= \hat{\beta}_0 + \hat{\beta}_1 \log x_i - (\hat{\beta}_0 + \hat{\beta}_1 \log x_j)\\
&= \hat{\beta}_1(\log x_i - \log x_j)\\
&= \hat{\beta}_1(\log(2x_j) - \log x_j)\\
&= \hat{\beta}_1(\log 2 + \log x_j - \log x_j)\\
&= \hat{\beta}_1 \log 2,
\end{aligned}$$

so the answers is $\hat{\beta}_1 \log(2) = 1.8232 \log(2) \approx 1.264$.

c) $y_{\text{predict}} = \exp(\hat{\beta}_0 + \hat{\beta}_1 * 3) = \exp(1.12022 + 0.95966 * 3) \approx 54.55$.

d) We have $\hat{y}_i = \exp(\hat{\beta}_0 + \hat{\beta}_1 x_i)$. Hence

$$\begin{aligned}
\hat{y}_i/\hat{y}_j &= \exp(\hat{\beta}_0 + \hat{\beta}_1 x_i)/\exp(\hat{\beta}_0 + \hat{\beta}_1 x_j)\\
&= \exp(\hat{\beta}_0 + \hat{\beta}_1 x_i - (\hat{\beta}_0 + \hat{\beta}_1 x_j))\\
&= \exp(\hat{\beta}_1(x_i - x_j))\\
&= \exp(\hat{\beta}_1(x_j + 1 - x_j))\\
&= \exp(\hat{\beta}_1),
\end{aligned}$$

and the answers is $\exp(\hat{\beta}_1) = \exp(0.95966) \approx 2.61$.

3. The command `set.seed` initializes the pseudo-random number generator, so that the results are reproducible.

a)
```
> set.seed(21)                                 ## initializes the rng
> nrep <- 100                                  ## number of repetitions
> slope <- numeric(nrep)                       ## initialization of vector
> x <- rnorm(40, 20, 3)                        ## x-values
> for (i in 1:nrep){
      y <- 1 + 2 * x + 5 * rnorm(length(x))    ## simulation of y-values
      reg <- lm(y ~ x)                         ## least squares regression
      slope[i] <- coefficients(reg)[2]         ## saves the slope
  }
```

b)
```
> par(mfrow = c(3, 2))                          ## display 6 figures as 3 rows and 2 columns
> set.seed(21)                                  ## initializes the rng
> x <- rnorm(40, 20, 3)                         ## x-values
> for (i in 1:3){                       ## now we only look at the first three simulations
      y <- 1 + 2 * x + 5 * rnorm(length(x))    ## simulation of y-values
      reg <- lm(y ~ x)                         ## least squares regression

      plot(y ~ x)
      abline(reg)
      plot(reg, which = 1)                     ## Tukey-Anscombe plot
  }
```

See Figure 1.

c) The mean and the standard deviation of the estimated slopes are 2.012 and 0.2431, respectively.

```
> mean(slope)                              ## mean and other information
> sd(slope)                                ## empirical standard deviation
```

d)
```
> X <- cbind(1, x)                         ## design matrix
> XtXinv <- solve(crossprod(X))            ## crossprod(X) <=> t(X) %*% X  = X'X
> tvar <- 5^2 * XtXinv[2, 2]
>
```

The theoretical variance of $\hat{\beta}_1$ is $\left[\sigma^2(X^\top X)^{-1}\right]_{22} = 0.0616$.

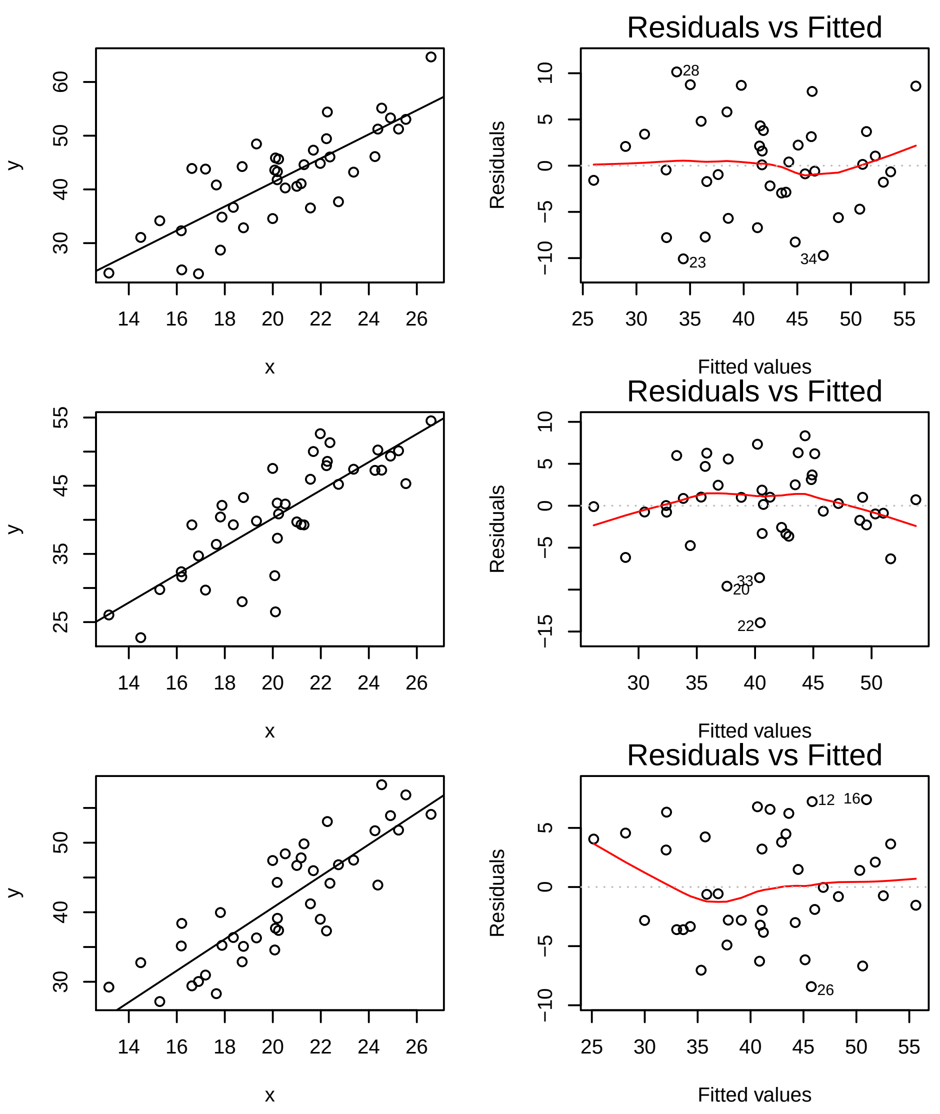


Figure 1: Plots of the observations and the regression line of a given data set and the corresponding Tukey-Anscombe plots for the first three times of simulations in problem 3 b).

e)
```
> par(mfrow = c(1, 1))                       ## now display only one figure
>  hist(slope, breaks = 15, freq = FALSE)             ## histogram
>  lines(seq(1.3, 2.6, by = 0.01),                    ## add theoretical density
         dnorm(seq(1.3, 2.6, by = 0.01), mean = 2, sd = sqrt(tvar) ) )
```

Under the simulation model (where $\epsilon \sim \mathcal{N}(0, 25I)$ and $\beta_1 = 2$), $\hat{\beta}_1$ has the following distribution $\mathcal{N}(\beta_1, \left[25(X^\top X)^{-1}\right]_{22}) = \mathcal{N}(2, 0.0616)$. We indeed see that the estimated $\hat{\beta}_1$'s closely follow this distribution; see Figure 2.

4. The following three lines of R code generate the same x values for all tasks. The comment part can be adapted for all the tasks (but note that the error term should be different for different tasks, and you can change the range when you draw plots, e.g. change (1.7, 2.3) to (1.5,2.5) when you draw the normal density line, or use a larger range of x-axis by "xlim=c(1, 7.4)" when you draw the histogram accocrding to your needs).

```
>  nrep <- 100            ## number of repetitions
>  set.seed(21)           ## initializes the rng
>  x <- rnorm(40,20,3)    ## x-values
>  X <- cbind(1, x)                   ## design matrix
>  XtXinv <- solve(crossprod(X))      ## crossprod(X) <=> t(X) %*% X  = X'X
>  tvar <- 5^2 * XtXinv[2, 2]
```

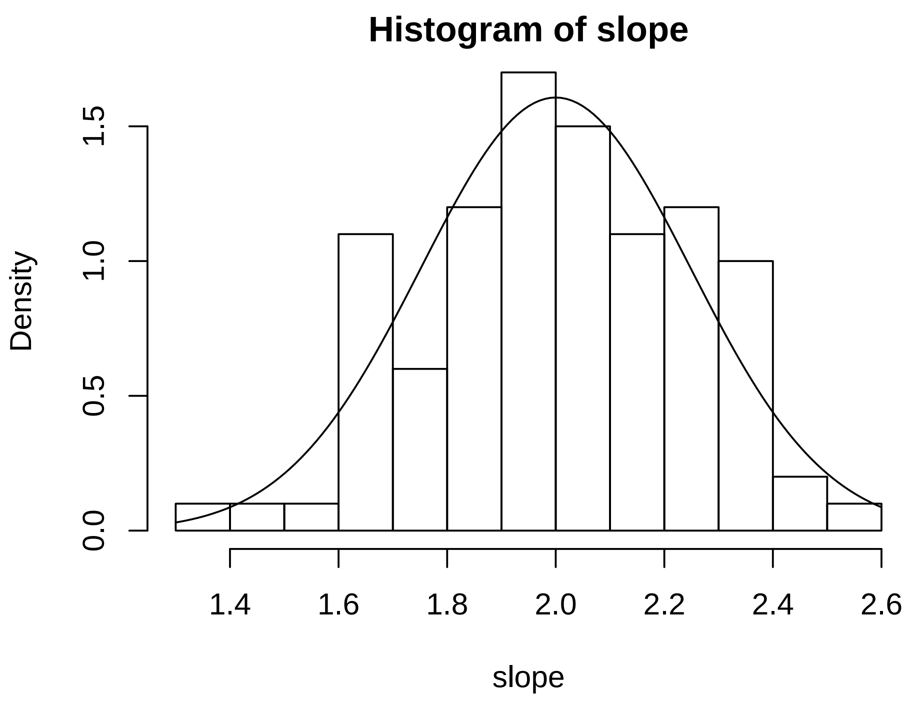


Figure 2: Histogram of the slopes in problem 3 b).

```
>
> # mean(slope)           ## mean
> # sd(slope)             ## empirical standard deviation
>
> # par(mfrow = c(3, 2))                      ## display 6 figures as 3 rows and 2 columns
> # set.seed(21)                              ## initializes the rng
> # x <- rnorm(40, 20, 3)                     ## x-values
> # for (i in 1:3){                        ## now we only look at the first three simulations
> #     y <- 1 + 2 * x + 5 * rnorm(length(x))  ## error should be different for different tasks
> #     reg <- lm(y ~ x)                      ## least squares regression
>
> #     plot(y ~ x)
> #     abline(reg)
> #     plot(reg, which = 1)                  ## Tukey-Anscombe plot
> # }
>
> # hist(slope, breaks = 15, freq=FALSE)         ## histogram
> # lines(seq(1.7, 2.3, by = 0.01),  ## add the same density
> #       dnorm(seq(1.7, 2.3, by = 0.01), mean = 2, sd = sqrt(tvar) ))
```

a)
```
> set.seed(29)                         ## initializes the rng
> slope <- numeric(nrep)               ## initialization of vector
> for (i in 1:nrep){
      y <- 1 + 2 * x + 5 * (1 - rchisq(length(x), df = 1)) / sqrt(2)
      reg <- lm(y ~ x)
      slope[i] <- coefficients(reg)[2]
  }
```

The errors are independent, have zero mean and constant variance $\sigma^2 = 25$, but they *are not normally distributed*. From Figure 3, one can see that the residuals are not symmetrically distributed (there are more positive points than negative points), this gives some evidence that the errors are not normally distributed. For the estimated slope $\hat{\beta}_1$, we have $\mathbf{E}\left[\hat{\beta}_1\right] = \beta_1$ and $\mathrm{Var}\left(\hat{\beta}_1\right) = \left[\sigma^2(X^\top X)^{-1}\right]_{22}$, but $\hat{\beta}_1$ is only approximately normally distributed. This approximation becomes better for large n (Central Limit Theorem) and the p-values are asymptotically correct. For $n = 40$, we see in Figure 4 that the distribution is starting to be like a Gaussian.

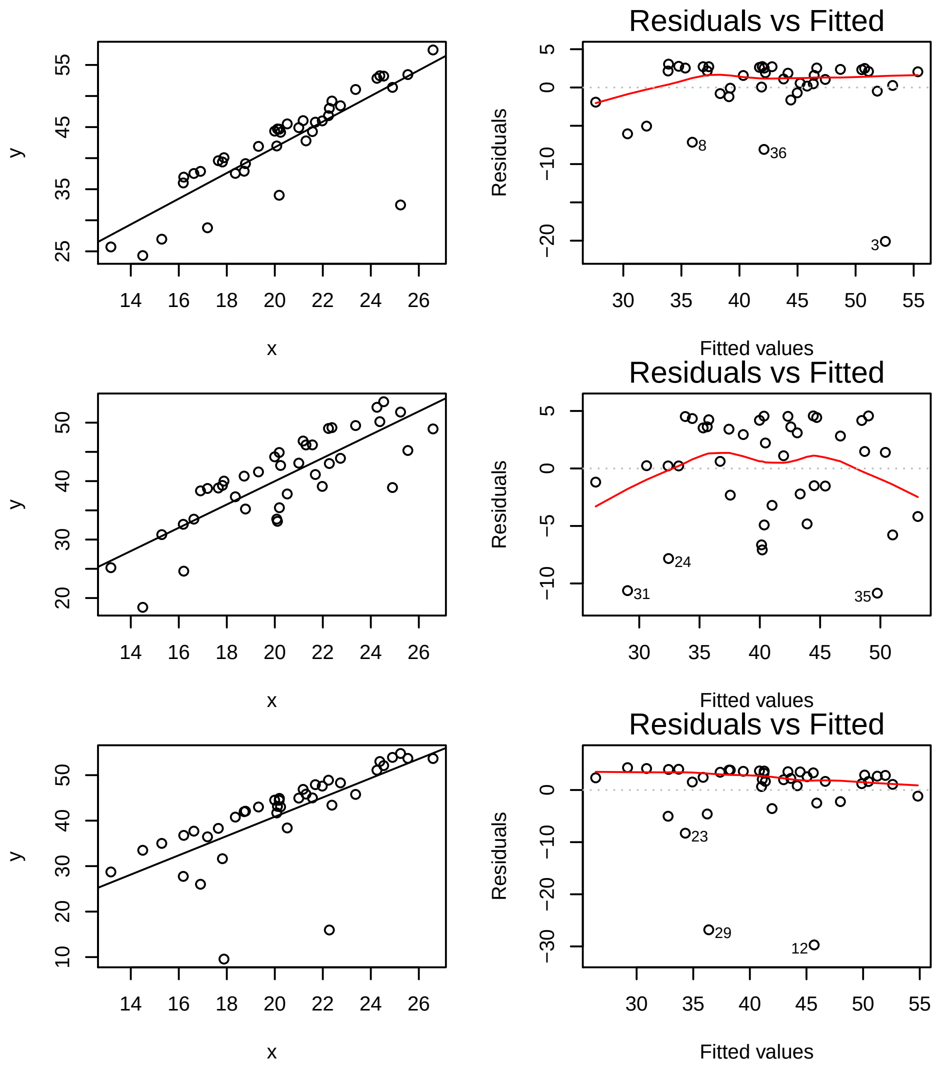


Figure 3: Plots of the observations and the regression line of a given data set and the corresponding Tukey-Anscombe plots for the first three times of simulations in problem 4 a).

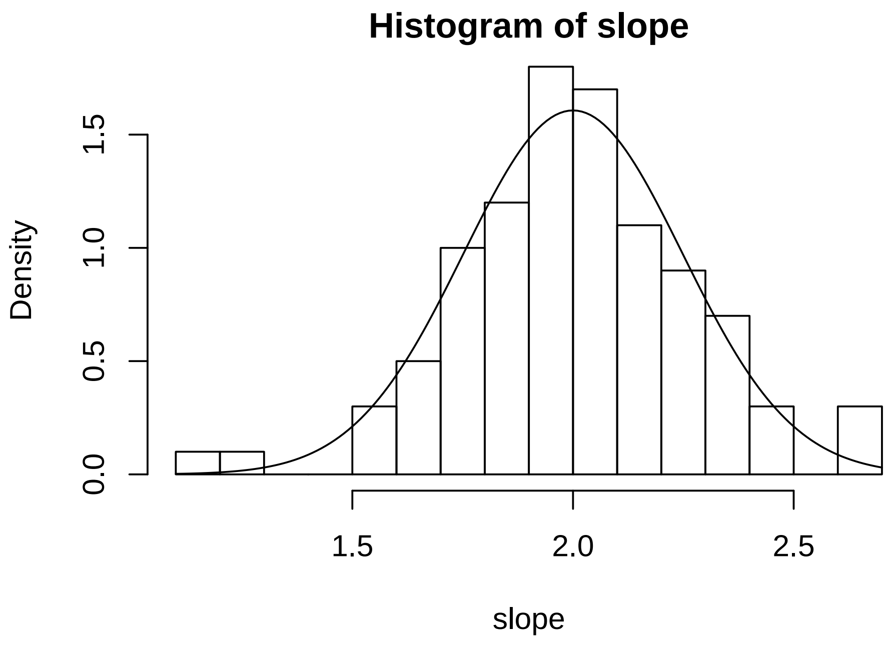


Figure 4: Histogram of the slopes in problem 4 a).

b)
```
> set.seed(245)                     ## initializes the rng
> slope <- numeric(nrep)            ## initialization of vector
> for (i in 1:nrep){
      y <- 1 + 2 * x + 5 * rnorm(length(x), mean = x^2 / 5 - 1, sd = 1)
      reg <- lm(y ~ x)
      slope[i] <- coefficients(reg)[2]
  }
```

The errors are independent with constant variance, but *do not have mean zero*: their expected value depends on x. This can be seen in Figure 5. Hence, the model is misspecified and the estimated $\hat{\beta}_1$ are severely biased. We clearly see this in Figure 6, where the estimated slopes are around 42.3 instead of 2. In the summary table of the R output, the least square estimation of the slope and intercept, the point estimation of error variance, and the resutls that depend on the distribution of $\hat{\beta}_1$ such as t-statistic and p-values are all not ok.

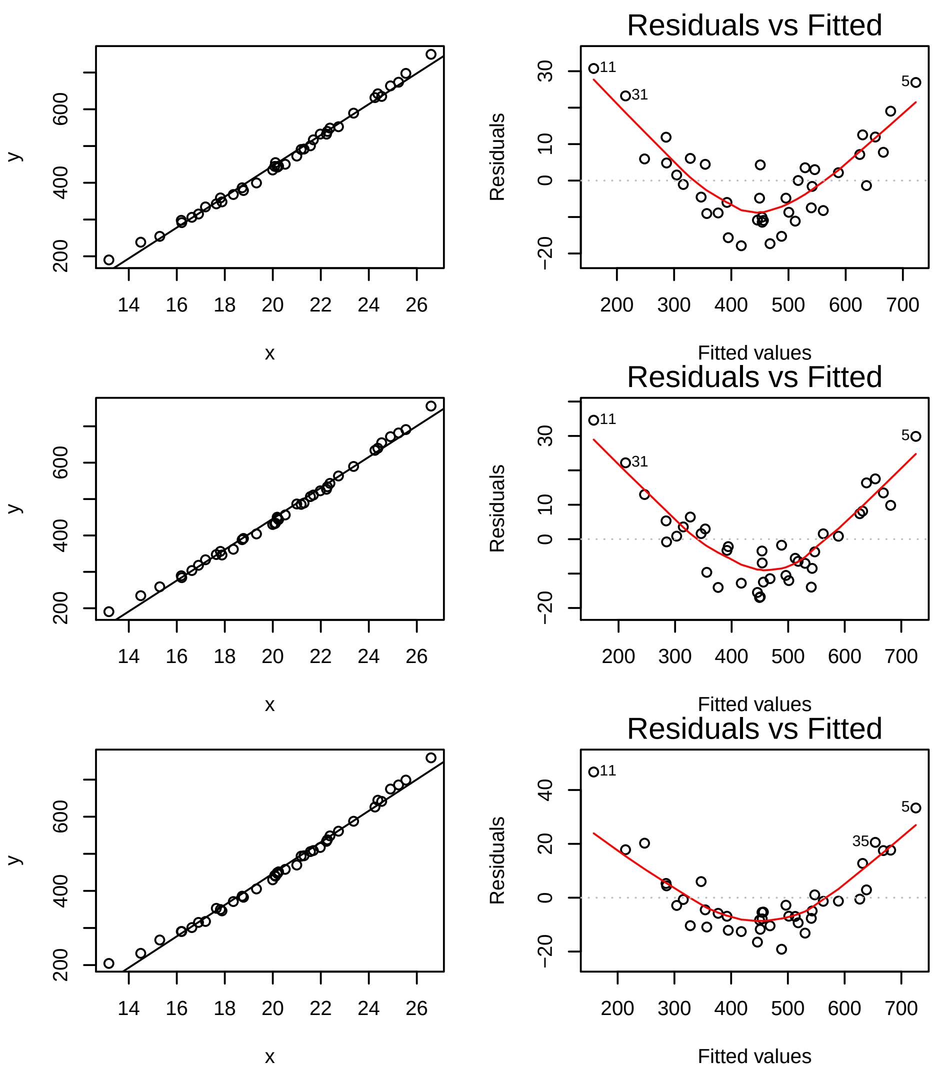


Figure 5: Plot of the observations and the regression line of a given data set and the corresponding Tukey-Anscombe plot in problem 4 b).

c)
```
> require(MASS)
> Sigma <- matrix(0.7,40,40)
> diag(Sigma) <- 1
> set.seed(61)                      ## initializes the rng
> slope <- numeric(nrep)            ## initialization of vector
> for (i in 1:nrep){
      y <- 1 + 2 * x + 5 * mvrnorm(n = 1, mu = rep(0, length(x)), Sigma = Sigma)
      reg <- lm(y ~ x)
```

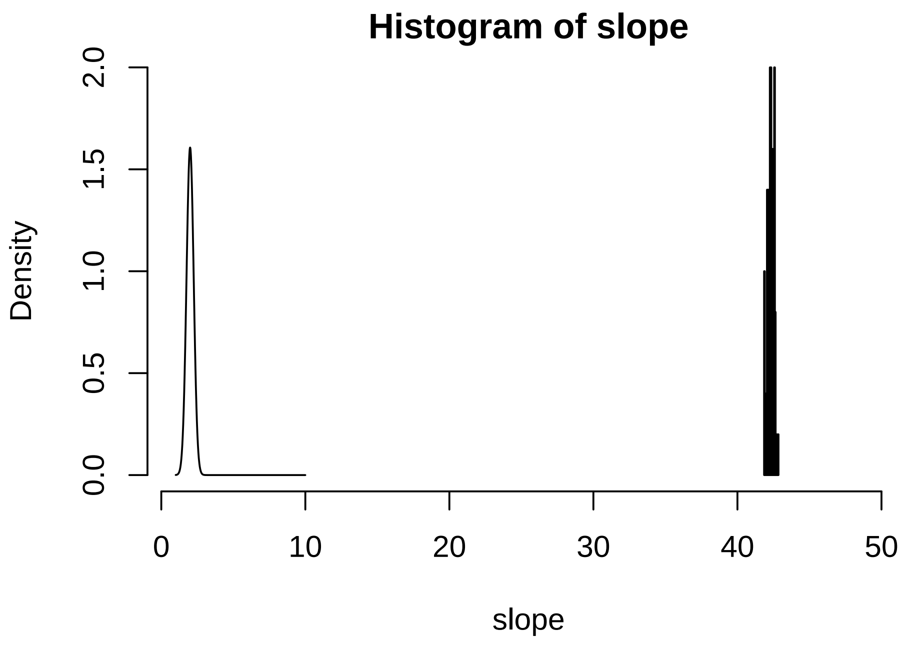


Figure 6: Histogram of the slopes in problem 4 b).

```
    slope[i] <- coefficients(reg)[2]
}
```

The errors are normally distributed with mean zero and constant variance $\sigma^2 = 25$, but they are *correlated*. Note that we *can not* detect this in Tukey-Anscombe plot (Figure 7). The estimated slopes are unbiased, but the standard error formula that is obtained under the uncorrelated error assumption does not hold. We indeed clearly see in Figure 8 that the estimated slopes have a Gaussian shape with mean 2, but a different variance compared with the normal density line that corresponds to distribution $\mathcal{N}(\beta_1, \left[25(X^\top X)^{-1}\right]_{22}) = \mathcal{N}(2, 0.0616)$. As a result, we do not trust the outputted standard errors, p-values, or any other inference.

d)
```
>  set.seed(997)                        ## initializes the rng
> slope <- numeric(nrep)              ## initialization of vector
> for (i in 1:nrep){
   y   <- 1 + 2 * x + 5 * rnorm(length(x), mean = 0, sd = (x-15)^2 / 30)
   reg <- lm(y ~ x)
   slope[i] <- coefficients(reg)[2]
 }
```

The errors are independent, have mean zero, and have *non constant variance*. This can be clearly seen in the Tukey-Anscombe plot; compare with Figure 9. Hence, the estimated slopes are unbiased, but the standard error formula does not hold. We indeed see in Figure 10 that the true variation in the estimated slopes is larger than the theoretical variation computed under the assumption of independent constant variance errors. As a result, we do not trust the outputted standard errors, p-values, or any other inference.

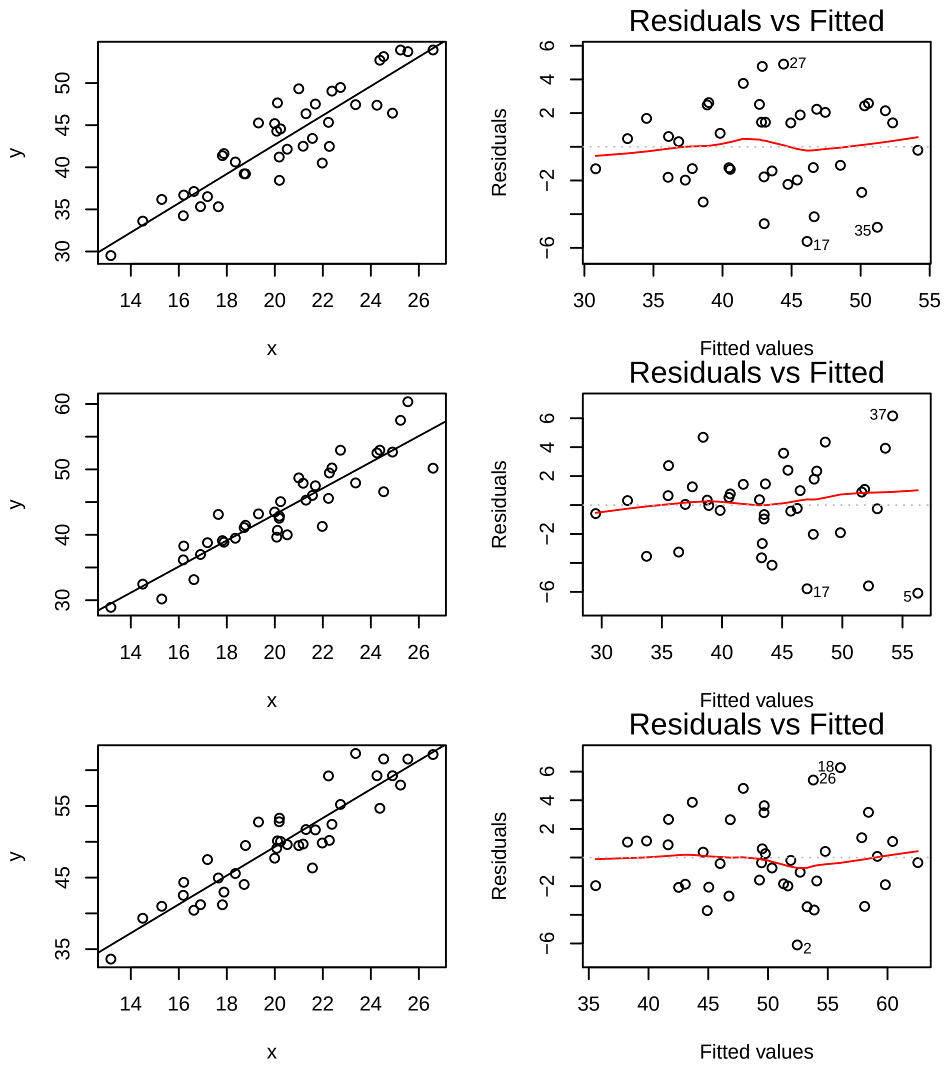


Figure 7: Plot of the observations and the regression line of a given data set and the corresponding Tukey-Anscombe plot in problem 4 c).

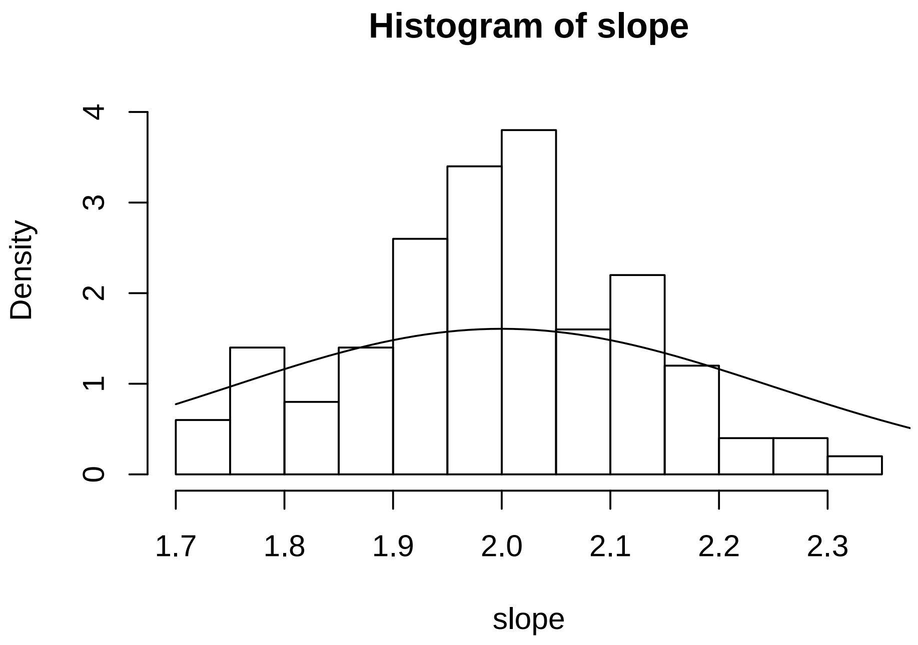


Figure 8: Histogram of the slopes in problem 4 c).

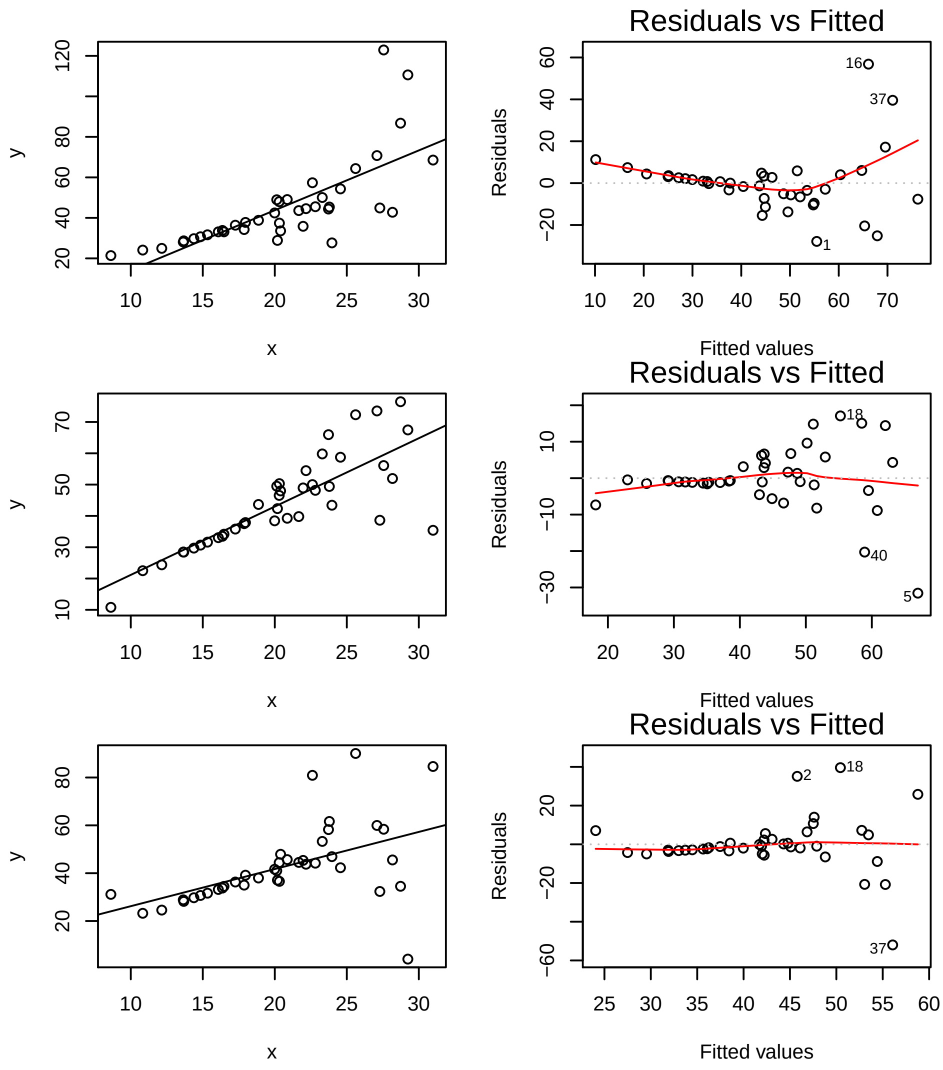


Figure 9: Plot of the observations and the regression line of a given data set and the corresponding Tukey-Anscombe plot in problem 4 d).

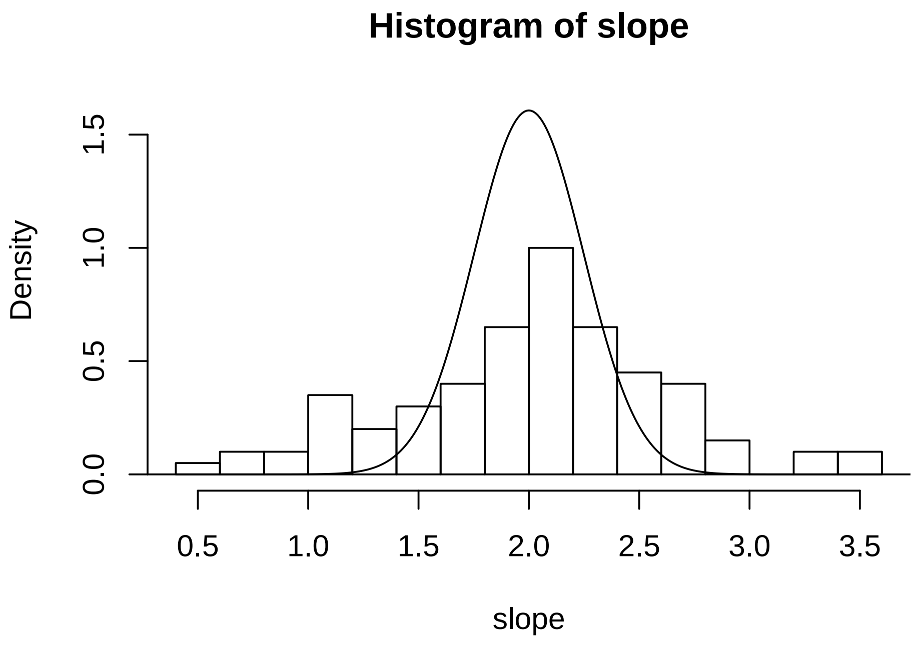


Figure 10: Histogram of the slopes in problem 4 d).