Statistical inference : part 2, practice session 2

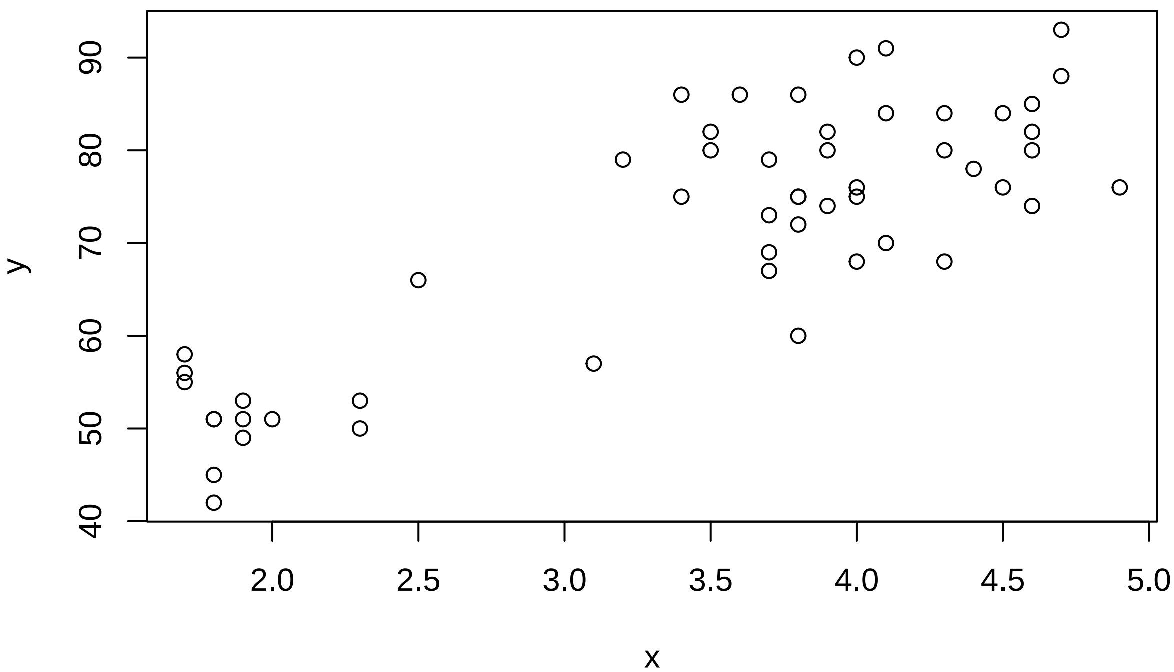
2023-11-30

Exercicse 6.14

Data importation:

```
geyser = read.csv(file = "geyser.csv", sep = ";")
```

```
plot(y ~ x, data = geyser)
```



(a) Implement the formulas given $\hat{\beta}_1$ and $\hat{\beta}_0$ (you can also use `lm` to fit the model `lm(y ~ x, data = geyser))`

```
x = geyser$x
y = geyser$y
xbar = mean(x)
ybar = mean(y)

beta1_hat = sum((x-xbar)*(y-ybar)) / sum((x - xbar)^2) ; beta1_hat
```

```
## [1] 11.3678
```

```
beta0_hat = ybar - beta1_hat * xbar ; beta0_hat
```

```
## [1] 31.75229
```

(b) Under $H_0 : \beta_1 = 0$, we have:

$$t = \frac{\hat{\beta}_1}{s/\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}} \sim t(n-2)$$

where $t(n-2)$ stands for the t distribution with $n-2$ degrees of freedom, and s^2 is the unbiaised estimator of the variance of the noise:

$$s^2 = \frac{1}{n-2}\sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{n-2}\sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

Thus, the course (p.132) says to reject H_0 if $|t| \geq t_{\alpha/2,n-2}$ where $t_{\alpha/2,n-2}$ is the upper $\alpha/2$ percentage point of the central t distribution with $n-2$ degrees of freedom. This is motivated by the fact that:

$$P_{H_0}(|t| \geq t_{\alpha/2,n-2}) = P_{H_0}(t \geq t_{\alpha/2,n-2}) + P_{H_0}(t \leq -t_{\alpha/2,n-2}) = 2P_{H_0}(t \geq t_{\alpha/2,n-2}) = 2 \times (\alpha/2) = \alpha$$

where the second equality comes from the symmetry of the student distribution. On R $t_{\alpha/2,n-2}$ can be obtained by `qt(alpha/2, n-2, lower.tail = F)`. Answer question by taking $\alpha = 0.05$.

```r
n = nrow(geyser)
alpha = 0.05
s2 = sum((y - beta0_hat - beta1_hat * x)^2) / (n-2)
t =  beta1_hat / (sqrt(s2) / sqrt(sum((x-xbar)^2))) ; t
```

```
## [1] 11.10859
```

```r
qt(alpha/2, n - 2, lower.tail = F)
```

```
## [1] 2.007584
```

```r
if (t >= qt(alpha/2, n - 2)){
  print(paste("We reject H0: beta1 is significantly different from 0"))
} else {
  print("We cannot reject H0: beta1 is not significantly different from 0")
}
```

```
## [1] "We reject H0: beta1 is significantly different from 0"
```

(you can check your results by making `summary(lm(y ~ x, data = geyser))` and looking at the p-value in the summary, you reject the test at risk α if p-value $\leq \alpha$)

```r
summary(lm(y ~ x, data = geyser))
```

```
##
## Call:
## lm(formula = y ~ x, data = geyser)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.9499  -4.8132  -0.8132   5.1868  15.5972
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   31.752      3.689   8.606 1.66e-11 ***
## x             11.368      1.023  11.109 3.17e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.442 on 51 degrees of freedom
## Multiple R-squared:  0.7076, Adjusted R-squared:  0.7018
## F-statistic: 123.4 on 1 and 51 DF,  p-value: 3.172e-15
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   31.752      3.689   8.606 1.66e-11 ***
x             11.368      1.023  11.109 3.17e-15 ***
```

The p-value is `3.17e-15` thus we reject H_0 at risk $\alpha = 0.05$.

It can also be computed in the following way: $2P_{H_0}(t > |t_{obs}|)$ where t is the test statistic following a t distribution with $n-2$ degrees of freedom under H_0, and t_{obs} is the value of the test statistic observed on the dataset at hand (denoted by t in previous code).

```r
2 * pt(t, n-2, lower.tail = F)
```

```
## [1] 3.172252e-15
```

(c) Now come back to

$$t = \frac{\hat{\beta}_1 - \beta_1}{s/\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}} \sim t(n-2)$$

(here we do not assume $H_0 : \beta_1 = 0$ any more)

Thus the formula can be obtained page 133 of the book: the following $100(1-\alpha)$ confidence interval for β_1 is

$$\hat{\beta}_1 \pm t_{\alpha/2,n-2} \frac{s}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

Compute the confidence interval with $\alpha = 0.05$.

```r
beta1_hat + c(-1,1) * qt(alpha/2, n - 2, lower.tail = F) * sqrt(s2) / sqrt(sum((x-xbar)^2))
```

```
## [1]  9.313375 13.422234
```

(you can check you results with `confint(lm(y~x, data = geyser))`)

```r
confint(lm(y~x, data = geyser))
```

```
##                 2.5 %   97.5 %
## (Intercept) 24.345472 39.15910
## x            9.313375 13.42223
```

(d)

$$r^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

(you can also check that $r^2 = \left(\frac{\text{cov}(x,y)}{\sigma_x \sigma_y}\right)^2$) where $\frac{\text{cov}(x,y)}{\sigma_x \sigma_y}$ is the linear correlation coefficient (you can use `cor(geyser$x, geyser$y)` to get this coefficient)

```r
yp = beta0_hat + beta1_hat * x
SSR = sum((yp - ybar)^2)
SST = sum((y - ybar)^2)
r2 = SSR / SST ;
paste("r2 =", r2)
```

```
## [1] "r2 = 0.707570192126858"
```

```r
correlation = cov(x,y)/(sd(x)*sd(y)) ; paste("correlation =",correlation)
```

```
## [1] "correlation = 0.841171915916632"
```

```r
paste("correlation^2 =",correlation^2)
```

```
## [1] "correlation^2 = 0.707570192126858"
```

(you can also look at `Multiple R-squared` value when making `summary(lm(y ~ x, data = geyser))`)

```
summary(lm(y ~ x, data = geyser))$r.squared
```

```
## [1] 0.7075702
```

Exercise 7.53

Import the data:

```
gas = read.csv("gas.csv",sep=";")
head(gas)
```

```
##    y x1 x2   x3   x4
## 1 29 33 53 3.32 3.42
## 2 24 31 36 3.10 3.26
## 3 26 33 51 3.18 3.18
## 4 22 37 51 3.39 3.08
## 5 27 36 54 3.20 3.41
## 6 21 35 35 3.03 3.03
```

```
X = cbind(1,as.matrix(gas[,-1])) # add a first column of 1 for the intercept
head(X)
```

```
##        x1 x2   x3   x4
## [1,] 1 33 53 3.32 3.42
## [2,] 1 31 36 3.10 3.26
## [3,] 1 33 51 3.18 3.18
## [4,] 1 37 51 3.39 3.08
## [5,] 1 36 54 3.20 3.41
## [6,] 1 35 35 3.03 3.03
```

```
n = nrow(X)
y = gas[,1]
```

(a)

Use the formula

$$\hat{\beta} = (X'X)^{-1}X'y$$

With R, the function `t` stands for transposition, `solve` for matrix inversion, and the operator `%*%` stand for matrix multiplication.

```
beta_hat = solve(t(X) %*% X) %*% t(X) %*% y ; beta_hat
```

```
##           [,1]
##     1.01501756
## x1 -0.02860886
## x2  0.21581693
## x3 -4.32005167
## x4  8.97488928
```

And s^2 is given by

$$s^2 = \frac{1}{n-k-1}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

```r
yp = as.vector(X %*% beta_hat)
s2 = sum((y - yp)^2) / (n - ncol(X)) ; s2
```

```
## [1] 7.452874
```

You can compare you results with the results of `lm`:

```r
reg = lm(y ~ ., data = gas)
reg$coefficients # hat beta
```

```
## (Intercept)          x1          x2          x3          x4
##  1.01501756 -0.02860886  0.21581693 -4.32005167  8.97488928
```

```r
reg_summary = summary(reg)
reg_summary$sigma # s
```

```
## [1] 2.729995
```

```r
reg_summary$sigma^2 # s2
```

```
## [1] 7.452874
```

(b) We know that $\text{cov}(\hat{\beta}) = \sigma^2(X'X)^{-1}$ and σ^2 can be estimated by s^2 thus

$$\widehat{\text{cov}(\hat{\beta})} = s^2(X'X)^{-1}$$

Thus do the computation of $\widehat{\text{cov}(\hat{\beta})}$ using previous results:

```r
hat_cov_beta_hat = s2 * solve(t(X) %*% X)
hat_cov_beta_hat
```

```
##                          x1           x2         x3          x4
##     3.46446861  0.014493274 -0.063835928 -1.1619793  1.07233308
## x1  0.01449327  0.008208638 -0.001925963 -0.1630247  0.07835194
## x2 -0.06383593 -0.001925963  0.004585728  0.1039042 -0.12500572
## x3 -1.16197929 -0.163024654  0.103904189  8.1280146 -7.20448052
## x4  1.07233308  0.078351940 -0.125005720 -7.2044805  7.68748532
```

You can check the value of $(X'X)^{-1}$

```r
reg_summary$cov.unscaled
```

```
##              (Intercept)            x1            x2          x3          x4
## (Intercept)  0.464850012  0.0019446557 -0.0085652766 -0.15591023  0.14388182
## x1           0.001944656  0.0011014057 -0.0002584188 -0.02187407  0.01051298
## x2          -0.008565277 -0.0002584188  0.0006152966  0.01394149 -0.01677282
## x3          -0.155910227 -0.0218740653  0.0139414927  1.09058795 -0.96667144
## x4           0.143881820  0.0105129832 -0.0167728207 -0.96667144  1.03147930
```

or directly obtain $s^2(X'X)^{-1}$ by making:

```r
vcov(reg)
```

```
##             (Intercept)           x1           x2         x3          x4
## (Intercept)  3.46446861  0.014493274 -0.063835928 -1.1619793  1.07233308
## x1           0.01449327  0.008208638 -0.001925963 -0.1630247  0.07835194
## x2          -0.06383593 -0.001925963  0.004585728  0.1039042 -0.12500572
## x3          -1.16197929 -0.163024654  0.103904189  8.1280146 -7.20448052
## x4           1.07233308  0.078351940 -0.125005720 -7.2044805  7.68748532
```

The diagonal elements gives the estimated variance of each parameter.

(c)

Find R^2 and R_a^2, you can use the formulas of exercise 7.29:

$$R^2 = 1 - SSE/\sum_i (y_i - \bar{y})^2$$

and

$$R_a^2 = 1 - \frac{SSE/(n-k-1)}{\sum_i (y_i - \bar{y})^2/(n-1)}$$

```
ybar = mean(y)
SSE = sum((y - yp)^2)
SST = sum((y - ybar)^2)
R2 = 1 - SSE/SST ; R2
```

```
## [1] 0.92606
```

```
R2a = 1 - (SSE/(n - ncol(X)))/(SST/(n-1)) ; R2a
```

```
## [1] 0.915106
```

You can check you results:

```
reg_summary$r.squared
```

```
## [1] 0.92606
```

```
reg_summary$adj.r.squared
```

```
## [1] 0.915106
```

where `Multiple R-squared` gives the R^2 and `Adjusted R-squared` gives R_a^2