

Programmieren mit R: Seminararbeit 3

Daniyar Akhmetov (5127348)

Marcelo Rainho Avila (4679876)

Xuan Son Le (4669361)

Abgabedatum: 08.02.2018

Contents

1	Part I: <i>Linear regression</i> (15 points)	2
1.1	<i>Raw implementation</i>	2
1.2	<i>Check equivalent</i>	3
1.3	<i>Extend implementation</i>	3
2	PartII: S3 for linear models	3

1 Part I: *Linear regression* (15 points)

1.1 *Raw implementation*

```
linModEst <- function(x,y) {
  ## compute temp=(x'x)^(-1) with x'=t(x)
  temp <- crossprod(x,y = NULL) #compute (x'x)
  temp <- solve(temp) #compute (x'x)^(-1)

  ## compute beta
  temp1 <- crossprod(x,y) #compute(x'y)
  beta <- crossprod(t(temp),temp1) #compute beta

  ## calculate degree of freedom
  df <- nrow(x) - ncol(x)

  ## calculate sigma^2
  SSR <- 0
  help <- 0
  for (i in (1:nrow(x))) {
    help <- (x[i,]) %*% beta
    SSR <- SSR + (y[i] - help)^2
  }
  sigma_power_2 <- as.double(SSR / df)
  sigma <- sqrt(sigma_power_2)

  # calculate covariance matrix
  vcov <- sigma_power_2 * temp

  # return results
  list(beta = beta, vcov = vcov, sigma = sigma, df = df)
}

data(cats, package = "MASS")
linModEst(x = cbind(1, cats$Bwt,as.numeric(cats$Sex) - 1),
          y = cats$Hwt)
```

```
## $beta
##           [,1]
## [1,] -0.41495263
## [2,]  4.07576892
## [3,] -0.08209684
##
## $vcov
##           [,1]      [,2]      [,3]
## [1,]  0.52900070 -0.20504763  0.06563743
## [2,] -0.20504763  0.08690026 -0.04696312
## [3,]  0.06563743 -0.04696312  0.09244480
##
## $sigma
## [1] 1.457138
##
## $df
```

```
## [1] 141
```

- By adding a new column with all values of 1, we transformed the given matrix x into X .
- `crossprod(x, y = NULL)` is equal to `t(x) %*% y`. Because `y = NULL` is taken to be the same matrix as `x`, the result will be `t(x) %*% x`. To find the inverse matrix of $x'x$ we used the function `solve()`.
- `x` is given as `cbind bla`, because the first column is used for β_0

1.2 Check equivalent

```
Hwt <- cats$Hwt
Bwt <- cats$Bwt
Sex <- as.numeric(cats$Sex) - 1

lm_cat <- lm(Hwt ~ Bwt + Sex, data = cats)
coef(lm_cat)
```

```
## (Intercept)      Bwt      SexM
## -0.41495263  4.07576892 -0.08209684
```

```
vcov(lm_cat)
```

```
##          (Intercept)      Bwt      SexM
## (Intercept)  0.52900070 -0.20504763  0.06563743
## Bwt         -0.20504763  0.08690026 -0.04696312
## SexM        0.06563743 -0.04696312  0.09244480
```

The same!!

1.3 Extend implementation

```
linMod <- function(formula, data){
  # still no idea :(
  lm(formula, data) # wäre schön ^^
}

linMod(Hwt ~ Bwt + Sex, data = cats)
```

```
##
## Call:
## lm(formula = formula, data = data)
##
## Coefficients:
## (Intercept)      Bwt      SexM
##   -0.4150      4.0758     -0.0821
```

2 PartII: S3 for linear models