

Solution to practical part 7: Exercise on linear algebra

1. The matrices and vectors can be generated in R:

```
> A <- matrix(c(4,2,3,1,4,6),byrow=TRUE,nrow=2)
> B <- matrix(c(0,3,6,-1,-1,0),byrow=TRUE,nrow=2)
> x <- c(-1,2,-3)
> y <- c(5,3,-2)
```

a) `> 2*A`

```
      [,1] [,2] [,3]
[1,]     8     4     6
[2,]     2     8    12
```

b) `> A + B`

```
      [,1] [,2] [,3]
[1,]     4     5     9
[2,]     0     3     6
```

c) Calculating the results by hand:

$$\begin{bmatrix} 4 & 2 & 3 \\ 1 & 4 & 6 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 \\ 3 & -1 \\ 6 & 0 \end{bmatrix} = \begin{bmatrix} 4 \cdot 0 + 2 \cdot 3 + 3 \cdot 6 & 4 \cdot (-1) + 2 \cdot (-1) + 3 \cdot 0 \\ 1 \cdot 0 + 4 \cdot 3 + 6 \cdot 6 & 1 \cdot (-1) + 4 \cdot (-1) + 6 \cdot 0 \end{bmatrix} = \begin{bmatrix} 24 & -6 \\ 48 & -5 \end{bmatrix}$$

Checking with R:

```
> A%%t(B)
```

```
      [,1] [,2]
[1,]    24    -6
[2,]    48    -5
```

d) Again, start by hand:

$$\begin{bmatrix} 4 & 2 & 3 \\ 1 & 4 & 6 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 2 \\ -3 \end{bmatrix} = \begin{bmatrix} 4 \cdot (-1) + 2 \cdot 2 + 3 \cdot (-3) \\ 1 \cdot (-1) + 4 \cdot 2 + 6 \cdot (-3) \end{bmatrix} = \begin{bmatrix} -9 \\ -11 \end{bmatrix}$$

And check by R:

```
> A %%% x
```

```
      [,1]
[1,]    -9
[2,]   -11
```

e) Not defined, wrong dimensions.

f) Not defined, wrong dimensions.

g) `> A %*% t(A)`

```
      [,1] [,2]
[1,]    29    30
[2,]    30    53
```

h) `> t(A) %*% A`

```
      [,1] [,2] [,3]
[1,]    17    12    18
[2,]    12    20    30
[3,]    18    30    45
```

i) `> t(x)%*%x`

```
      [,1]
[1,]    14
```

j) `> x%*%t(x)`

```
      [,1] [,2] [,3]
[1,]     1    -2     3
[2,]    -2     4    -6
[3,]     3    -6     9
```

2.

a) In the lecture we have defined the covariate vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, the data matrix $\tilde{\mathbf{X}}$, the $\boldsymbol{\beta}$ vector, and the response vector:

```
> x1 <- c(0,1,2,3,4)
> x2 <- c(4,1,0,1,4)
> Xtilde <- matrix(c(rep(1,5),x1,x2),ncol=3)
> t.beta <- c(10,5,-2)
> t.y <- Xtilde%*%t.beta
> t.e <- rnorm(5,0,1)
> t.Y <- t.y + t.e
> r.lm <- lm(t.Y ~ x1 + x2)
> summary(r.lm)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	10.424761	1.1593521	8.991885	0.012143152
x1	4.996217	0.3895548	12.825454	0.006024432
x2	-2.494555	0.3292339	-7.576848	0.016976675

```
> solve(t(Xtilde) %*% Xtilde) %*% t(Xtilde) %*% t.Y
```

```
      [,1]
[1,] 10.424761
[2,]  4.996217
[3,] -2.494555
```

- b) Generate a matrix of true value `t.y` and a matrix of residual errors `t.E` (instead of generating a separate `t.e` each time). The 100 observed vectors are then stored in a 5×100 matrix `t.Y`:

```
> t.y <- matrix(rep(t.y,100),nrow=5,byrow=F)
> t.E <- matrix(rnorm(500,0,1),nrow=5)
> t.Y <- t.y + t.E
```

100 iterations of the regression:

```
> r.coef <- matrix(NA,ncol=3, nrow=100)
> for (i in 1:100) {
+   r.coef[i,] <- lm(t.Y[,i] ~x1 + x2)$coefficients
+ }
```

- c) We produce the required graphs using `ggplot`. Load the libraries:

```
> library(ggplot2)
> library(tidyr)
> library(dplyr)
```

For the histograms, first convert `r.coef` into a data frame and rename the columns:

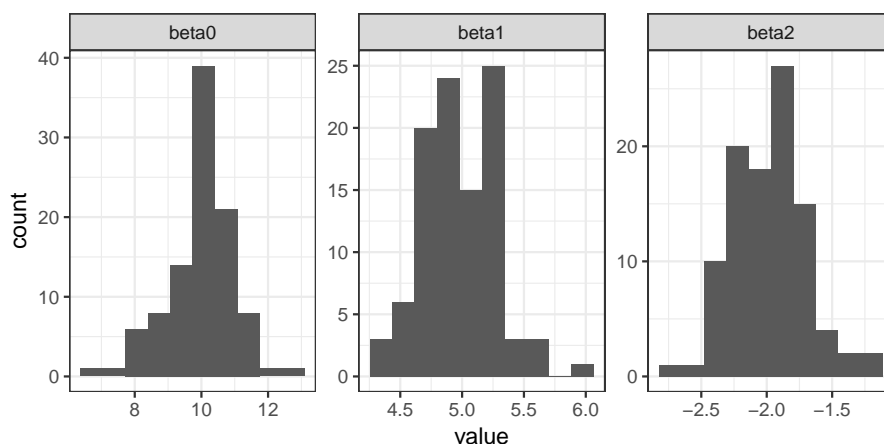
```
> r.coef <- data.frame(r.coef)
> names(r.coef) <- c("beta0", "beta1", "beta2")
```

Then either produce three separate plots using

```
> ggplot(r.coef,aes(x=beta0)) + geom_histogram()
> ggplot(r.coef,aes(x=beta1)) + geom_histogram()
> ggplot(r.coef,aes(x=beta2)) + geom_histogram()
```

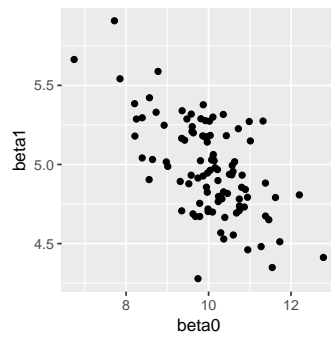
Or use the `gather()` function from the `tidyr` package, which has the advantage that you can then use `facet_wrap()`:

```
> ggplot(gather(r.coef, key=variable, value=value), aes(value)) +
+   geom_histogram(bins=10) + facet_wrap(~variable, scales = "free") +
+   theme_bw()
```

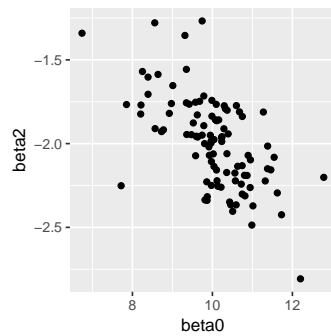


For the scatterplots we can continue to use `r.coef`:

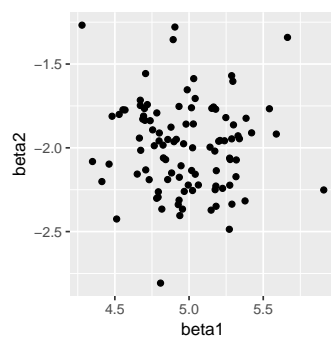
```
> ggplot(r.coef,aes(x=beta0,y=beta1)) + geom_point()
```



```
> ggplot(r.coef,aes(x=beta0,y=beta2)) + geom_point()
```



```
> ggplot(r.coef,aes(x=beta1,y=beta2)) + geom_point()
```



Observation: β_0 seems to be correlated with β_1 and β_2 .