Exercises

June 12, 2019

[3]:
```
# head(beaver1, 5)
# head(beaver2, 5)
# beaver1$temp[1:3]
data <- cbind(beaver1$temp[1:100], (5/8)*beaver1$temp[1:100] + 32,␣
 ↪beaver2$temp[1:100], (5/8)*beaver2$temp[1:100] + 32)
head(data, 5)
cov(data)
```

36.33	54.70625	36.58	54.86250
36.34	54.71250	36.73	54.95625
36.35	54.71875	36.93	55.08125
36.42	54.76250	37.15	55.21875
36.55	54.84375	37.23	55.26875

0.04116698	0.02572936	0.03449019	0.02155637
0.02572936	0.01608085	0.02155637	0.01347273
0.03449019	0.02155637	0.19962031	0.12476270
0.02155637	0.01347273	0.12476270	0.07797668

[4]:
```
data.mean <- matrix(data = 1, nrow=nrow(data)) %*% colMeans(data)
data.mean <- data - data.mean
# head(data.mean,10)
C <- (nrow(data) - 1)^-1 * t(data.mean) %*% data.mean
C
# eigen(cov(data))
# prcomp(data)
```

0.04116698	0.02572936	0.03449019	0.02155637
0.02572936	0.01608085	0.02155637	0.01347273
0.03449019	0.02155637	0.19962031	0.12476270
0.02155637	0.01347273	0.12476270	0.07797668

[5]:
```
eigen(C)
eigen(cov(data.mean))
data.pca <- prcomp(data, center = TRUE, scale = TRUE)
summary(data.pca)
```

```
eigen() decomposition
$values
[1]  2.875843e-01  4.726052e-02  4.857226e-17 -6.750473e-18
```

```
$vectors
           [,1]       [,2]          [,3]          [,4]
[1,] -0.1728705  0.8301909  0.000000e+00 -5.299989e-01
[2,] -0.1080441  0.5188693 -1.107443e-15  8.479983e-01
[3,] -0.8301909 -0.1728705 -5.299989e-01 -1.221245e-15
[4,] -0.5188693 -0.1080441  8.479983e-01  3.330669e-16



eigen() decomposition
$values
[1]  2.875843e-01  4.726052e-02  0.000000e+00 -6.591949e-17

$vectors
          [,1]       [,2]          [,3]          [,4]
[1,] 0.1728705  0.8301909  0.000000e+00 -5.299989e-01
[2,] 0.1080441  0.5188693 -1.022255e-15  8.479983e-01
[3,] 0.8301909 -0.1728705 -5.299989e-01 -1.221245e-15
[4,] 0.5188693 -0.1080441  8.479983e-01  1.110223e-16



Importance of components:
                          PC1    PC2       PC3       PC4
Standard deviation     1.6616 1.1131 2.326e-14 6.917e-15
Proportion of Variance 0.6902 0.3098 0.000e+00 0.000e+00
Cumulative Proportion  0.6902 1.0000 1.000e+00 1.000e+00
```

[6]:
```
str(data.pca)
```

```
List of 5
 $ sdev    : num [1:4] 1.66 1.11 2.33e-14 6.92e-15
 $ rotation: num [1:4, 1:4] 0.5 0.5 0.5 0.5 -0.5 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:4] "PC1" "PC2" "PC3" "PC4"
 $ center  : num [1:4] 36.9 55 37.6 55.5
 $ scale   : num [1:4] 0.203 0.127 0.447 0.279
 $ x       : num [1:100, 1:4] -4.89 -4.51 -4.01 -3.17 -2.36 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:4] "PC1" "PC2" "PC3" "PC4"
 - attr(*, "class")= chr "prcomp"
```

[7]:
```
library(ggbiplot)
ggbiplot(data.pca)
```

```
ERROR while rich displaying an object: Error in FUN(X[[i]], ...): object
'varname' not found

Traceback:
1. FUN(X[[i]], ...)
2. tryCatch(withCallingHandlers({
 .     rpr <- mime2repr[[mime]](obj)
 .     if (is.null(rpr))
 .         return(NULL)
 .     prepare_content(is.raw(rpr), rpr)
 . }, error = error_handler), error = outer_handler)
3. tryCatchList(expr, classes, parentenv, handlers)
4. tryCatchOne(expr, names, parentenv, handlers[[1L]])
5. doTryCatch(return(expr), name, parentenv, handler)
6. withCallingHandlers({
 .     rpr <- mime2repr[[mime]](obj)
 .     if (is.null(rpr))
 .         return(NULL)
 .     prepare_content(is.raw(rpr), rpr)
 . }, error = error_handler)
7. mime2repr[[mime]](obj)
8. repr_text.default(obj)
9. paste(capture.output(print(obj)), collapse = "\n")
10. capture.output(print(obj))
11. evalVis(expr)
12. withVisible(eval(expr, pf))
13. eval(expr, pf)
14. eval(expr, pf)
15. print(obj)
16. print.ggplot(obj)
17. ggplot_build(x)
18. ggplot_build.ggplot(x)
19. by_layer(function(l, d) l$compute_aesthetics(d, plot))
20. f(l = layers[[i]], d = data[[i]])
21. l$compute_aesthetics(d, plot)
22. f(..., self = self)
23. scales_add_defaults(plot$scales, data, aesthetics, plot$plot_env)
24. lapply(aesthetics[new_aesthetics], rlang::eval_tidy, data = data)
25. FUN(X[[i]], ...)
```

[44]:
```
C <- cov(data)
e <- eigen(C)
R <- e$vectors; R
S <- diag(e$values); S
T <- R %*% S
T
```

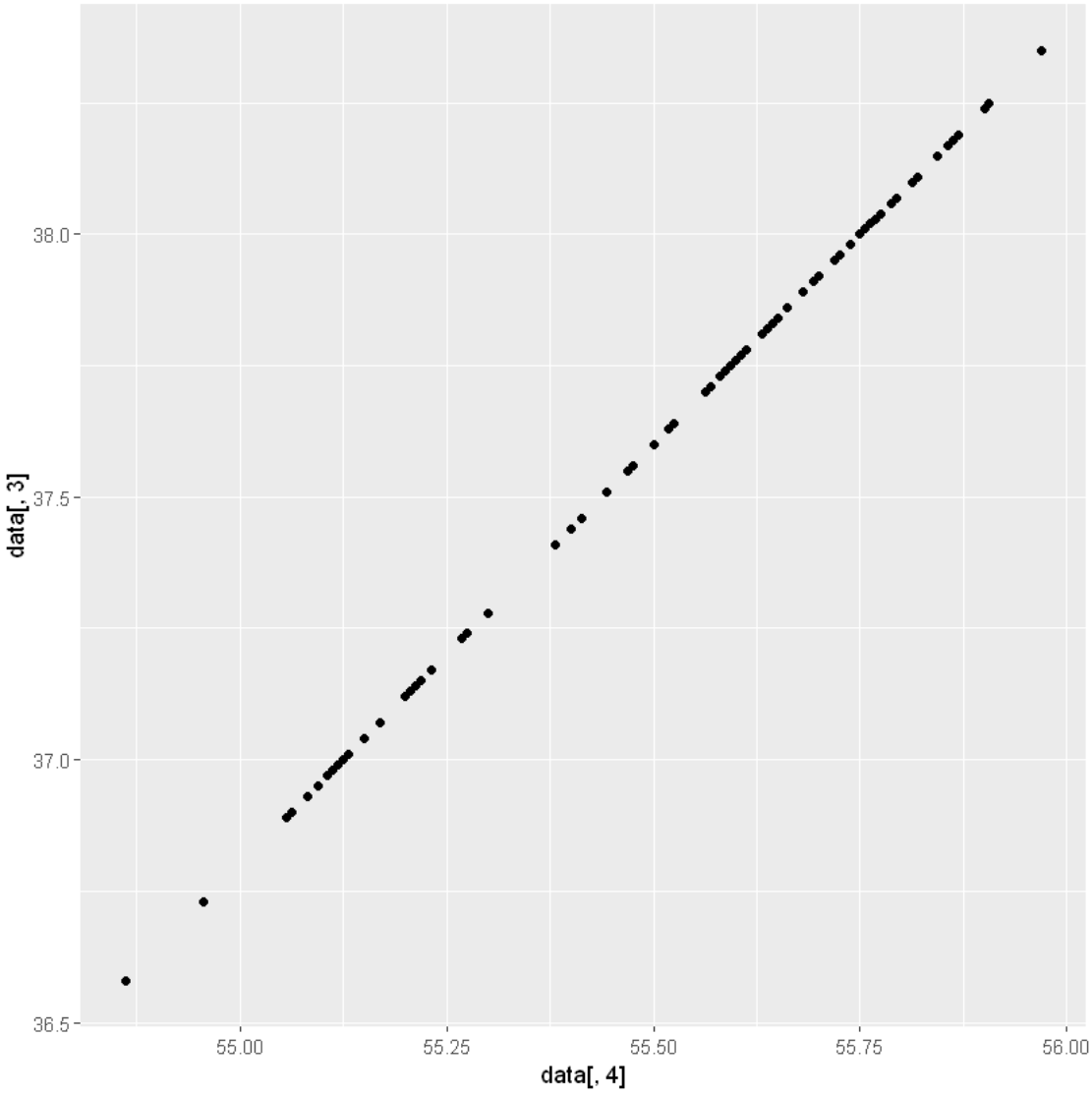
0.1728705	0.8301909	0.000000e+00	-5.299989e-01
0.1080441	0.5188693	-1.022255e-15	8.479983e-01
0.8301909	-0.1728705	-5.299989e-01	-1.221245e-15
0.5188693	-0.1080441	8.479983e-01	1.110223e-16
0.2875843	0.00000000	0	0.000000e+00
0.0000000	0.04726052	0	0.000000e+00
0.0000000	0.00000000	0	0.000000e+00
0.0000000	0.00000000	0	-6.591949e-17

0.04971484	0.039235254	0	3.493726e-17
0.03107178	0.024522033	0	-5.589962e-17
0.23874987	-0.008169950	0	8.050387e-32
0.14921867	-0.005106219	0	-7.318534e-33

[45]:
```
library(ggplot2)
```

```
Warning message:
```

[51]:
```
data <- as.data.frame(data)
ggplot(data = data) + geom_point(aes(data[,4], data[,3]))
```



[2]:
```
# # Prepare data
# data()
```

```r
beaver1C <- beaver1$temp
beaver1F <- (9/5)*beaver1$temp + 32
beaver2C <- beaver2$temp
beaver2F <- (9/5)*beaver1$temp + 32

beavers <- data.frame(beaver1C[1:100], beaver1F[1:100], beaver2C[1:100],␣
 ↪beaver2F[1:100])

# Calculate covariance
columnMeans <- c(mean(beavers$beaver1C.1.100.),
                 mean(beavers$beaver1F.1.100.),
                 mean(beavers$beaver2C.1.100.),
                 mean(beavers$beaver2F.1.100.))

subtracted <- scale(beavers, columnMeans, FALSE)

covarianceMatrix <- (t(subtracted)%*%subtracted)/99
covarianceMatrix

# Check with cov()
cov(beavers)

# Calculate eigenvectors of covariance matrix
ev <- eigen(covarianceMatrix)
eigenValue <- ev$values
eigenVectors <- ev$vectors


# Check with prcomp()
prcomp(cov(beavers))
```

	beaver1C.1.100.	beaver1F.1.100.	beaver2C.1.100.	beaver2F.1.100.
beaver1C.1.100.	0.04116698	0.07410056	0.03449019	0.07410056
beaver1F.1.100.	0.07410056	0.13338101	0.06208235	0.13338101
beaver2C.1.100.	0.03449019	0.06208235	0.19962031	0.06208235
beaver2F.1.100.	0.07410056	0.13338101	0.06208235	0.13338101
	beaver1C.1.100.	beaver1F.1.100.	beaver2C.1.100.	beaver2F.1.100.
beaver1C.1.100.	0.04116698	0.07410056	0.03449019	0.07410056
beaver1F.1.100.	0.07410056	0.13338101	0.06208235	0.13338101
beaver2C.1.100.	0.03449019	0.06208235	0.19962031	0.06208235
beaver2F.1.100.	0.07410056	0.13338101	0.06208235	0.13338101

```
Standard deviations (1, .., p=4):
[1] 8.375758e-02 4.327370e-02 1.150339e-17 1.374194e-18

Rotation (n x k) = (4 x 4):
                       PC1        PC2            PC3           PC4
```

```
beaver1C.1.100.  0.1950403 0.3092719 -9.269180e-01  8.445781e-02
beaver1F.1.100.  0.3510725 0.5566895  3.216407e-01  6.807291e-01
beaver2C.1.100. -0.8458460 0.5334271 -5.551115e-16 -5.551115e-17
beaver2F.1.100.  0.3510725 0.5566895  1.933137e-01 -7.276501e-01
```

Na kolokwium rozklad svd, $U * D * V^T$, gdzie D jest macierz diagonaln, a U, T maj ortogonalne kolumny

zmiany znaków w V -> zmiana znaków w U

znaki musz odpowiada sobie parami??