Solution to the homework

Exercise 2

(a) [0.5 points] We compute the eigenvalues by solving

$$\det(\mathbf{\Sigma} - \lambda I_2) = \det \begin{pmatrix} 1-\lambda & \rho \\ \rho & 1-\lambda \end{pmatrix} = (1-\lambda)^2 - \rho^2 = 0$$
$$\Leftrightarrow \lambda_1 = 1+\rho,\ \lambda_2 = 1-\rho.$$

When $\rho > 0$, we thus have $\lambda_1 > \lambda_2$, and when $\rho \leq 0$, $\lambda_1 \leq \lambda_2$.

(b) [0.5 points for each eigenvector] The eigenvector to eigenvalue λ_i is the solution v_i of $(\mathbf{\Sigma} - \lambda_i I)v_i = 0$. Here we obtain

1. for the eigenvalue $\lambda_1 = 1+\rho$

$$(\mathbf{\Sigma} - (1+\rho)I_2)v_1 = 0 \ \Leftrightarrow \ \begin{pmatrix} -\rho & \rho \\ \rho & -\rho \end{pmatrix} \begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} = 0 \ \Leftrightarrow v_{11} = v_{12}.$$

Since we are looking for standardised eigenvectors, the condition $1 = v_{11}^1 + v_{12}^2$ yields

$$1 = 2v_{11}^2 \ \Leftrightarrow v_{11} = \frac{1}{\sqrt{2}},$$

i.e. $v_1 = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})'$ (unique up to sign).

2. for the eigenvalue $\lambda_2 = 1-\rho$

$$(\mathbf{\Sigma} - (1-\rho)I_2)v_2 = 0 \ \Leftrightarrow \ \begin{pmatrix} \rho & \rho \\ \rho & \rho \end{pmatrix} \begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix} = 0 \ \Leftrightarrow v_{21} = -v_{22},$$

and the condition $1 = v_{21}^2 + v_{22}^2$ yields $v_2 = (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})'$ (unique up to sign).

(c) [1 point] The principal components are

1. *case* $\rho > 0$: $Z_1 = v_1'\mathbf{Y} = \frac{1}{\sqrt{2}}(Y_1 + Y_2)$ and $Z_2 = v_2'\mathbf{Y} = \frac{1}{\sqrt{2}}(Y_1 - Y_2)$ (0.5 **P**)
2. *case* $\rho < 0$: $Z_1 = v_2'\mathbf{Y} = \frac{1}{\sqrt{2}}(Y_1 - Y_2)$ and $Z_2 = v_1'\mathbf{Y} = \frac{1}{\sqrt{2}}(Y_1 + Y_2)$. (0.5 **P**)
3. *case* $\rho = 0$: the principal components are not unique, any vector of length 1 may be chosen.

(d) [0.5 points]

1. *case* $\rho > 0$: $\mathrm{Var}(Z_1) = \lambda_1 = 1+\rho,\ \mathrm{Var}(Z_2) = \lambda_2 = 1-\rho$ and the proportions of explained variance are $\kappa_1 = \frac{\lambda_1}{\lambda_1+\lambda_2} = \frac{1+\rho}{2},\ \kappa_2 = \frac{\lambda_2}{\lambda_1+\lambda_2} = \frac{1-\rho}{2}$.
2. *case* $\rho \leq 0$: $\mathrm{Var}(Z_1) = \lambda_2 = 1-\rho,\ \mathrm{Var}(Z_2) = \lambda_1 = 1+\rho$ and the proportions of explained variance are $\kappa_1 = \frac{\lambda_2}{\lambda_1+\lambda_2} = \frac{1-\rho}{2},\ \kappa_2 = \frac{\lambda_1}{\lambda_1+\lambda_2} = \frac{1+\rho}{2}$.

(e) [1 point] The larger the amount of explained variance of the first principal component is, the better the data is represented by it. (i.e., the closer the value of κ_1 is to 1, the better the data is represented by Z_1).

For $\rho > 0$ we have $\lim_{\rho \to 1} \kappa_1 = 1$, for $\rho < 0$ we have $\lim_{\rho \to -1} \kappa_1 = 1$, and $\lim_{\rho \to 0} \kappa_1 = \frac{1}{2}$ in both cases. Therefore, if $\rho \in \{-1, 1\}$, the first PC already explains 100% of the total variance.

Exercise 1

a) [0.5 points]

First, we load the library and have a look at the documentation of the dataset.

```
## starting httpd help server ... done
```

The format is a list containing two elements: data and labs. 'data' contains the expression levels on 6839 genes from 64 cancer cell lines; 'labs' is the corresponding cancer type (of the cancer cell lines)

```
nci_labs <- NCI60$labs
nci_data <- NCI60$data
dim(nci_data)
```

```
## [1]   64 6830
```

Running `head()` is not informative because it has more than 6000 columns. To be precise, we have 6830 gene expression measurements of 64 cancer cell lines. Instead, we have a look at the first 10 observations of the first five observed variables.

```
nci_data[1:10, 1:5]
```

```
##             1         2         3         4         5
## V1   0.300000  1.180000  0.550000  1.140000 -0.265000
## V2   0.679961  1.289961  0.169961  0.379961  0.464961
## V3   0.940000 -0.040000 -0.170000 -0.040000 -0.605000
## V4   0.280000 -0.310000  0.680000 -0.810000  0.625000
## V5   0.485000 -0.465000  0.395000  0.905000  0.200000
## V6   0.310000 -0.030000 -0.100000 -0.460000 -0.205000
## V7  -0.830000  0.000000  0.130000 -1.630000  0.075000
## V8  -0.190000 -0.870000 -0.450000  0.080000  0.005000
## V9   0.460000  0.000000  1.150000 -1.400000 -0.005000
## V10  0.760000  1.490000  0.280000  0.100000 -0.525000
```

b) [0.5 points]

With the function `table()` we can count how often which cancer type appears in the vector with the cancer types. Then we extract the names of those that appear more often than 3 times.

```
table(nci_labs)
```

```
## nci_labs
##      BREAST         CNS       COLON K562A-repro K562B-repro    LEUKEMIA
##           7           5           7           1           1           6
## MCF7A-repro MCF7D-repro    MELANOMA       NSCLC     OVARIAN    PROSTATE
##           1           1           8           9           6           2
##       RENAL     UNKNOWN
##           9           1
```

```
which(table(nci_labs) > 3)
```

```
##   BREAST      CNS    COLON LEUKEMIA MELANOMA    NSCLC  OVARIAN    RENAL
##        1        2        3        6        9       10       11       13
```

```
names(which(table(nci_labs)>3))
```

```
## [1] "BREAST"   "CNS"      "COLON"    "LEUKEMIA" "MELANOMA" "NSCLC"    "OVARIAN"
## [8] "RENAL"
```

```
chosen_rows <- nci_labs %in% names(which(table(nci_labs)>3))
chosen_rows
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
## [25]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
## [37]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [49] FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [61]  TRUE  TRUE  TRUE  TRUE
```

The last vector specifies for each row whether it is kept or not.

```
nci_red <- nci_data[chosen_rows,]
dim(nci_red)
```

```
## [1]   57 6830
```

c) [0.5 points]

Since we cannot have a look at all means and variances seperately (too many dimensions), we just have a look at their summary statistics.

```
summary(apply(nci_red, 2, var))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.03636  0.20893  0.31720  0.62533  0.64217 12.01224
```

```
summary(apply(nci_red, 2, mean))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.848773 -0.037804  0.009209  0.024343  0.060408  1.358420
```

The variances range between 0.03 and 12. That's a factor of 400, so we should better scale the variables to have unit variance when performing PCA.

d) [0.5 points]

As seen in c), we should set `scale = TRUE` in `prcomp()` to scale the observations (centering is done per default).

```
pc_out <- prcomp(nci_red, scale. = TRUE)

smry_pc <- summary(pc_out)
smry_pc
```

```
## Importance of components:
##                             PC1      PC2      PC3      PC4      PC5      PC6
## Standard deviation     27.3099 22.36530 19.46639 16.62983 16.00892 14.74726
## Proportion of Variance  0.1092  0.07324  0.05548  0.04049  0.03752  0.03184
## Cumulative Proportion   0.1092  0.18244  0.23792  0.27841  0.31593  0.34777
##                             PC7      PC8      PC9     PC10     PC11     PC12
## Standard deviation     14.48757 13.83538 13.27152 13.16853 12.93163 12.51617
## Proportion of Variance  0.03073  0.02803  0.02579  0.02539  0.02448  0.02294
## Cumulative Proportion   0.37850  0.40653  0.43232  0.45771  0.48219  0.50513
##                            PC13     PC14    PC15     PC16     PC17    PC18
## Standard deviation     12.01300 11.64955 11.3606 11.09790 11.03919 10.9334
## Proportion of Variance  0.02113  0.01987  0.0189  0.01803  0.01784  0.0175
## Cumulative Proportion   0.52626  0.54613  0.5650  0.58306  0.60090  0.6184
##                            PC19     PC20     PC21     PC22   PC23    PC24
## Standard deviation     10.80167 10.66492 10.60354 10.40594 9.9851 9.96200
## Proportion of Variance  0.01708  0.01665  0.01646  0.01585 0.0146 0.01453
```

```
## Cumulative Proportion   0.63548  0.65214  0.66860  0.68445 0.6990 0.71358
##                            PC25    PC26    PC27    PC28    PC29    PC30    PC31
## Standard deviation     9.89818 9.75035 9.49149 9.44121 9.30743 9.14528 8.99625
## Proportion of Variance 0.01434 0.01392 0.01319 0.01305 0.01268 0.01225 0.01185
## Cumulative Proportion  0.72793 0.74185 0.75504 0.76809 0.78077 0.79302 0.80486
##                            PC32    PC33    PC34    PC35    PC36    PC37   PC38
## Standard deviation     8.80696 8.65411 8.38351 8.33533 8.26887 8.19655 8.0984
## Proportion of Variance 0.01136 0.01097 0.01029 0.01017 0.01001 0.00984 0.0096
## Cumulative Proportion  0.81622 0.82719 0.83748 0.84765 0.85766 0.86750 0.8771
##                            PC39    PC40    PC41    PC42    PC43    PC44    PC45
## Standard deviation     7.86983 7.83016 7.79418 7.53768 7.37896 7.26338 7.24387
## Proportion of Variance 0.00907 0.00898 0.00889 0.00832 0.00797 0.00772 0.00768
## Cumulative Proportion  0.88617 0.89514 0.90404 0.91236 0.92033 0.92805 0.93574
##                          PC46    PC47    PC48    PC49    PC50    PC51    PC52
## Standard deviation     7.1104 7.01741 6.74126 6.70367 6.44671 6.31987 6.31465
## Proportion of Variance 0.0074 0.00721 0.00665 0.00658 0.00608 0.00585 0.00584
## Cumulative Proportion  0.9431 0.95035 0.95700 0.96358 0.96967 0.97551 0.98135
##                            PC53    PC54    PC55    PC56      PC57
## Standard deviation     6.22922 6.04133 5.77809 4.32175 2.595e-14
## Proportion of Variance 0.00568 0.00534 0.00489 0.00273 0.000e+00
## Cumulative Proportion  0.98703 0.99238 0.99727 1.00000 1.000e+00
```

e) [0.5 points each]

The importance matrix is one of the values in the list that is returned when running `summary()` on the `prcomp`-object. One can access it with $ \texttt{smry_pc$importance}$. The cumulative proportion is the third row within that matrix.

```
smry_pc$importance[3, ]
```

```
##     PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10
## 0.10920 0.18244 0.23792 0.27841 0.31593 0.34777 0.37850 0.40653 0.43232 0.45771
##    PC11    PC12    PC13    PC14    PC15    PC16    PC17    PC18    PC19    PC20
## 0.48219 0.50513 0.52626 0.54613 0.56502 0.58306 0.60090 0.61840 0.63548 0.65214
##    PC21    PC22    PC23    PC24    PC25    PC26    PC27    PC28    PC29    PC30
## 0.66860 0.68445 0.69905 0.71358 0.72793 0.74185 0.75504 0.76809 0.78077 0.79302
##    PC31    PC32    PC33    PC34    PC35    PC36    PC37    PC38    PC39    PC40
## 0.80486 0.81622 0.82719 0.83748 0.84765 0.85766 0.86750 0.87710 0.88617 0.89514
##    PC41    PC42    PC43    PC44    PC45    PC46    PC47    PC48    PC49    PC50
## 0.90404 0.91236 0.92033 0.92805 0.93574 0.94314 0.95035 0.95700 0.96358 0.96967
##    PC51    PC52    PC53    PC54    PC55    PC56    PC57
## 0.97551 0.98135 0.98703 0.99238 0.99727 1.00000 1.00000
```

```
which( smry_pc$importance[3, ] >= .8)[1]
```

```
## PC31
##   31
```

For example, the first criterion would suggest to use 31 PCs if we would want to explain 80 % of total variance.

For the second criterion, we need the average proportion of explained variance, which is the mean of the second row of the importance matrix:

```
# average prop of explained variance
mean(smry_pc$importance[2, ])
```

```
## [1] 0.01754333
```

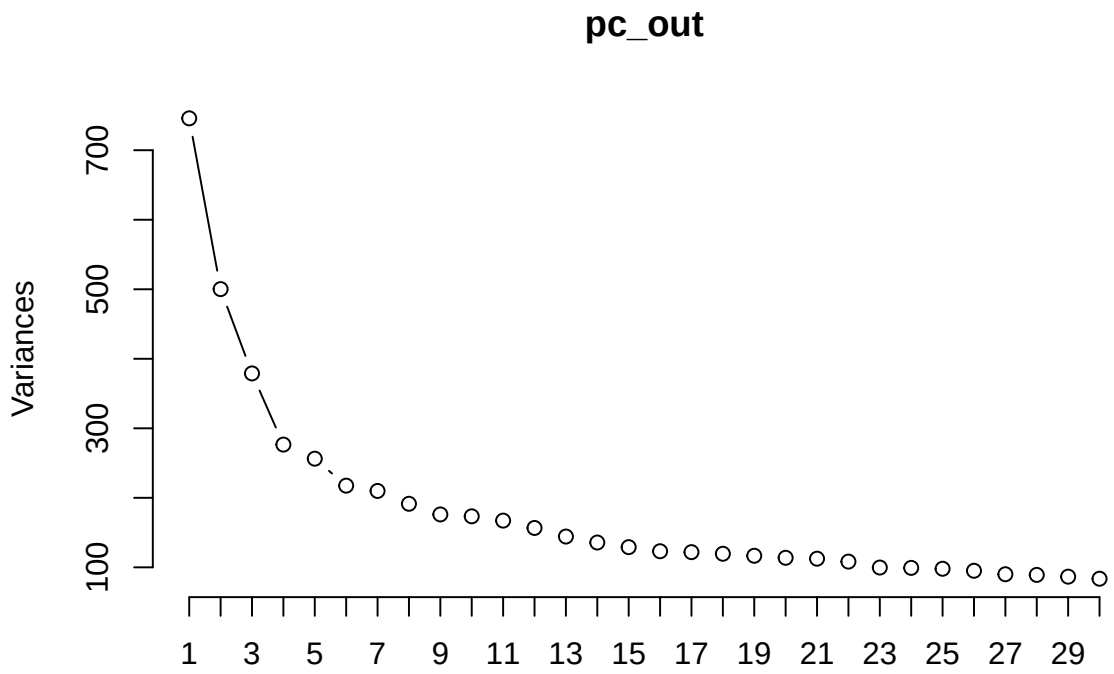
```
# those with larger than average proportion of explained variance
which(smry_pc$importance[2,] >= mean(smry_pc$importance[2, ]))
```

```
##  PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9 PC10 PC11 PC12 PC13 PC14 PC15 PC16
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## PC17
##   17
```

The second criterion would choose 17 PCs.

The third criterion is the screeplot, which can be generated with

```
screeplot(pc_out, type = "l", npcs = 30)
```



It would suggest to use 3 PCs.

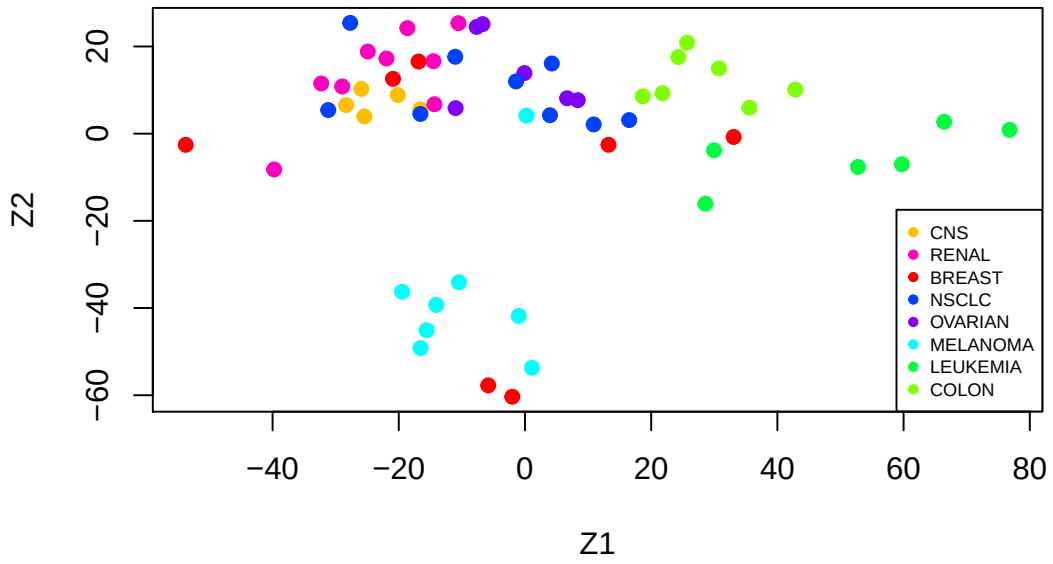
f) [0.5 points]

We first define a function that returns a color for each element of a vector:

```
Cols <- function(vec){
  cols <- rainbow(length(unique(vec)))
  return(cols[as.numeric(as.factor(vec))])
}
```

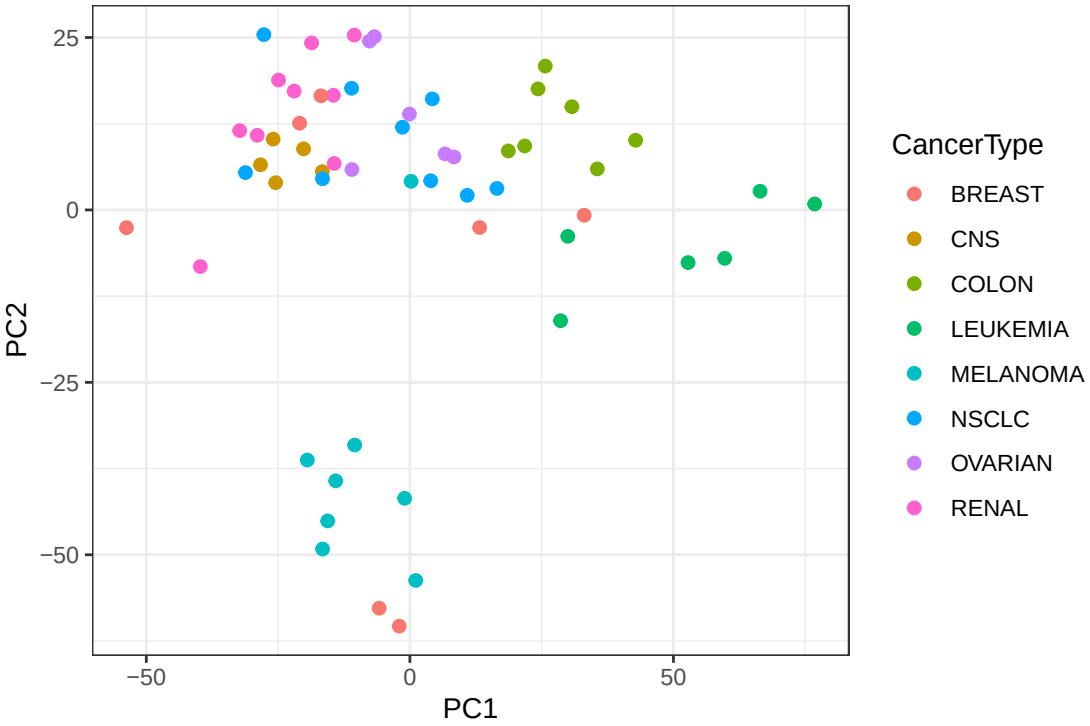
The scores of the first two PCs are returned in the list element x of the `prcomp` object.

```
plot(pc_out$x[,1:2], col = Cols(nci_labs[chosen_rows]), pch = 19,
     xlab="Z1",ylab="Z2")
legend("bottomright",  legend = unique(nci_labs[chosen_rows]), col = Cols(unique(nci_labs[chosen_rows])),
       pch = 19, cex = 0.6)
```



For `ggplot()` we define a dataframe with the scores and the corresponding cancer type first.

```
library(ggplot2)
df <- data.frame(pc_out$x[, 1:2])
df$CancerType <- nci_labs[chosen_rows]
ggplot(df, aes( x = PC1, y  = PC2, color = CancerType)) +
  geom_point(size = 2) +
   theme_bw()   # for black and white color theme
```


25
0
−25
−50
PC2
−50
0
50
PC1
CancerType
BREAST
CNS
COLON
LEUKEMIA
MELANOMA
NSCLC
OVARIAN
RENAL