Biostatistics HomeWork 1

KIM SANG HYUN(202211545)

2025-03-14

Contents

0. Package and Data

```r
library(ISLR)
data("NCI60")
unique(NCI60$labs)
```

```
##  [1] "CNS"          "RENAL"        "BREAST"       "NSCLC"        "UNKNOWN"
##  [6] "OVARIAN"      "MELANOMA"     "PROSTATE"     "LEUKEMIA"     "K562B-repro"
## [11] "K562A-repro" "COLON"        "MCF7A-repro" "MCF7D-repro"
```

```r
dim(NCI60$data) # row = cancer cell lines(samples), col = gene expression
```

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

```r
str(NCI60)
```

```
## List of 2
##  $ data: num [1:64, 1:6830] 0.3 0.68 0.94 0.28 0.485 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:64] "V1" "V2" "V3" "V4" ...
##   .. ..$ : chr [1:6830] "1" "2" "3" "4" ...
##  $ labs: chr [1:64] "CNS" "CNS" "CNS" "RENAL" ...
```

```r
NCI60$data[1:5, 1:10]
```

```
##           1         2         3         4         5             6        7
## V1 0.300000  1.180000  0.550000  1.140000 -0.265000 -7.000000e-02 0.350000
## V2 0.679961  1.289961  0.169961  0.379961  0.464961  5.799610e-01 0.699961
## V3 0.940000 -0.040000 -0.170000 -0.040000 -0.605000  0.000000e+00 0.090000
## V4 0.280000 -0.310000  0.680000 -0.810000  0.625000 -1.387779e-17 0.170000
## V5 0.485000 -0.465000  0.395000  0.905000  0.200000 -5.000000e-03 0.085000
##            8           9          10
## V1 -0.315000 -0.45000000 -0.65498050
## V2  0.724961 -0.04003899 -0.28501950
## V3  0.645000  0.43000000  0.47501950
## V4  0.245000  0.02000000  0.09501949
## V5  0.110000  0.23500000  1.49001949
```

```r
sum(is.na(NCI60$data))
```

```
## [1] 0
```

1. For each cancer cell line, compute average gene expression values. Identify two cell lines that have the largest and the smallest mean values. Also, include the maximun and minimum mean values.

```r
sol_1 = function(x = NCI60$data, cell_name = NCI60$labs){

  avg_gene_expr_cancer_cell = apply(x, 1, mean)

  max_idx = which.max(avg_gene_expr_cancer_cell)
  min_idx = which.min(avg_gene_expr_cancer_cell)

  max_cell_name = cell_name[max_idx]
  min_cell_name = cell_name[min_idx]

  max_value = avg_gene_expr_cancer_cell[max_idx]
  min_value = avg_gene_expr_cancer_cell[min_idx]

  return(
     list(
      #avg_gene_expr_cancer_cell = avg_gene_expr_cancer_cell,
      max_cell_name = max_cell_name,
      max_value = max_value,
      min_cell_name = min_cell_name,
      min_value = min_value
    )
  )

}

sol_1()
```

```
## $max_cell_name
## [1] "BREAST"
##
## $max_value
##        V5
## 0.1485874
##
## $min_cell_name
## [1] "LEUKEMIA"
##
## $min_value
##        V39
## -0.1420865
```

2. For each gene, compute average gene expression values of 64 cancer cell lines, including "UNKNOWN" label. Identify top 5 gene that have the largest mean expression values and top 5 genes that have smallest mean expression values. Also, include their mean values with gene ID number(1~6830).

```r
sol_2 = function(x = NCI60$data){

  avg_gene_expr_gene = apply(x ,2, mean)

  top_5_max_idx = order(avg_gene_expr_gene, decreasing = TRUE)[1 : 5] # sort()
  top_5_min_idx = order(avg_gene_expr_gene)[1 : 5]

  top_5_max_gene = avg_gene_expr_gene[top_5_max_idx]
  top_5_min_gene = avg_gene_expr_gene[top_5_min_idx]

  return(
    list(
      top_5_max_gene = top_5_max_gene,
      top_5_min_gene = top_5_min_gene
    )
  )
}

sol_2()
```

```
## $top_5_max_gene
##      6393       256       257      4700      6391
## 1.1676457 1.1137491 1.0627335 0.9985928 0.9920303
##
## $top_5_min_gene
##       5869       5868       5984       3438        281
## -0.8621881 -0.7442193 -0.7360845 -0.7223447 -0.7109384
```

3. Suppose that group "A" contains "BREAST" and "NSCLC", group "B" has "MELANOMA", "OVARIAN" and "PROSTATE", and group "C" has "LEUKEMIA", "RENAL" and "UNKNOWN". The other 6 cell lines belong to group "D". For each cancer group, compute the mean expression values and the standard deviation of gene experssion values.

```r
cell_name = NCI60$labs

# grouping cells
A = c("BREAST","NSCLC")
B = c("MELANOMA", "OVARIAN", "PROSTATE")
C = c("LEUKEMIA", "RENAL", "UNKNOWN")

cell_group = ifelse(cell_name %in% A, "A",
               ifelse(cell_name %in% B, "B",
               ifelse(cell_name %in% C, "C", "D")))
```

sol 3__1

```r
groups = c("A", "B", "C", "D")

sol_3_1 = function(x = NCI60$data, group = groups){

  group_mean = sapply(group,
                      function(g) {
                        mean(x[cell_group == g, ])
                      })

  group_sd = sapply(group,
                    function(g) {
                      sd(x[cell_group == g, ])
                    })

  return(c_df = cbind(as.data.frame(group_mean), as.data.frame(group_sd)))
}

sol_3_1()
```

```
##    group_mean  group_sd
## A 0.025053741 0.8006589
## B 0.040698588 0.7368874
## C 0.008992817 0.8443378
## D 0.005848171 0.7924379
```

sol 3_2

```
sol_3_2= function(x = NCI60$data, group = groups){

  group_mean_ = NULL
  group_sd_ = NULL
  counter = 1

  for (g in group){
    tmp_mean = mean(x[cell_group == g,])
    tmp_sd = sd((x[cell_group == g,]))

    group_mean_[counter] = tmp_mean
    group_sd_[counter] = tmp_sd

    counter = counter + 1
  }
  names(group_mean_) = group
  names(group_sd_) = group

  list(
    group_mean = group_mean_,
    group_sd = group_sd_
  )

}

sol_3_2()
```

```
## $group_mean
##           A           B           C           D
## 0.025053741 0.040698588 0.008992817 0.005848171
##
## $group_sd
##         A         B         C         D
## 0.8006589 0.7368874 0.8443378 0.7924379
```

sol 3_3

```
sol_3_3 = function(x = NCI60$data, group = groups, cell_groups = cell_group){

  tmp_dataframe = data.frame(cell_group = cell_groups,
                      NCI60$data, check.names = FALSE)
  group_mean_ = by(tmp_dataframe[,-1] , cell_groups, function(x) mean(as.matrix(x)))
  group_sd_  = by(tmp_dataframe[,-1] , cell_groups, function(x) sd(as.matrix(x)))

  return(
    list(
      group_mean = group_mean_,
      group_sd = group_sd_
    )
  )
}

sol_3_3()
```

```
## $group_mean
## cell_groups: A
## [1] 0.02505374
## ------------------------------------------------------------
## cell_groups: B
## [1] 0.04069859
## ------------------------------------------------------------
## cell_groups: C
## [1] 0.008992817
## ------------------------------------------------------------
## cell_groups: D
## [1] 0.005848171
##
## $group_sd
## cell_groups: A
## [1] 0.8006589
## ------------------------------------------------------------
## cell_groups: B
## [1] 0.7368874
## ------------------------------------------------------------
## cell_groups: C
## [1] 0.8443378
## ------------------------------------------------------------
## cell_groups: D
## [1] 0.7924379
```

4. For each cancer group defined in Q3, compute the sample SD of gene expression values of individual genes. Find genes whose SD is less than 0.2 or greater than 2 for each cancer group, i.e., $SD < 0.2$ or $SD > 2$. How many genes are overlapped by 4 different cancer groups? How many genes are overlapped by exactly 3 different cancer groups? or exactly 2 different cancer groups? Also, how many genes are uniquely identified by only one cancer group? Summarize your answer, using the following table.

sol 4_1

```r
sol_4 = function(){
  group_sd = lapply(groups,
                    function(grp) {
                      apply(NCI60$data[cell_group == grp, ], 2, sd)
                    })
  names(group_sd) = c("A", "B", "C", "D")
  filtered_genes = lapply(group_sd,
                          function(sd_value){
                            names(sd_value[sd_value < 0.2 | sd_value > 2])
                          })

  unique_genes = unique(c(filtered_genes[["A"]],
                          filtered_genes[["B"]],
                          filtered_genes[["C"]],
                          filtered_genes[["D"]]))

  num_overlapped = factor(apply(sapply(filtered_genes,
                                       function(g_list) {
                                         unique_genes %in% g_list
                                       }), 1, sum), levels = 1:4)

  sol_df = data.frame(
    x4 = length(num_overlapped[num_overlapped == "4"]),
    x3 = length(num_overlapped[num_overlapped == "3"]),
    x2 = length(num_overlapped[num_overlapped == "2"]),
    x1  = length(num_overlapped[num_overlapped == "1"])
  )
  rownames(sol_df) = "The number of genes"
  colnames(sol_df) = c("4 groups", "3 groups", "2 groups", "1 groups")
  return(
    list(
      #filtered_genes = filtered_genes,
      sol_df = sol_df
      ))
}

sol_4()
```

```
## $sol_df
##                     4 groups 3 groups 2 groups 1 groups
## The number of genes       14       17       71      220
```

sol 4_2

```r
tmp_df = data.frame(cell_group = cell_group,
                      NCI60$data, check.names = FALSE)

tmp_grp_values = by(tmp_df[,-1], cell_group, function(x) apply(x, 2, sd))

tmp_outlier = lapply(tmp_grp_values, function(tmp){
  names(tmp[tmp < 0.2 | tmp> 2])
})

tmp_unique_genes = unique(c(tmp_outlier[["A"]],
                            tmp_outlier[["B"]],
                            tmp_outlier[["C"]],
                            tmp_outlier[["D"]]))

sol = NULL
counter = 1
num = 0

for (i in tmp_unique_genes){
  num = 0
  for (j in groups){
    if(i %in% tmp_outlier[[j]]){
      num = num + 1
    }
  }
  sol[counter] = num
  counter = counter + 1
}

tmp = as.data.frame(table(factor(sol, levels = 1:4)))
tmp_t =  as.data.frame(t(tmp))
sol_4_2 = tmp_t[2, , drop = FALSE]

rownames(sol_4_2) = "The number of genes"
colnames(sol_4_2) = c("4 groups", "3 groups", "2 groups", "1 groups")

sol_4_2
```

```
##                     4 groups 3 groups 2 groups 1 groups
## The number of genes      220       71       17       14
```

sol 4_3

```r
tmp_df = data.frame(cell_group = cell_group,
                      NCI60$data, check.names = FALSE)

tmp_grp_values = by(tmp_df[,-1], cell_group, function(x) apply(x, 2, sd))

tmp_outlier = lapply(tmp_grp_values, function(tmp){
  names(tmp[tmp < 0.2 | tmp> 2])
})

tmp_unique_genes = unique(c(tmp_outlier[["A"]],
                            tmp_outlier[["B"]],
                            tmp_outlier[["C"]],
                            tmp_outlier[["D"]]))

############## double for => double sapply ######################
tmp_sol = sapply(tmp_unique_genes, function(gene) {
  sum(sapply(groups, function(gr) gene %in% tmp_outlier[[gr]]))
})
#################################################################

tmp =  as.data.frame(table(factor(tmp_sol, levels = 1:4)))
tmp_t =  as.data.frame(t(tmp))
sol_4_3 = tmp_t[2, , drop = FALSE]

rownames(sol_4_3) = "The number of genes"
colnames(sol_4_3) = c("4 groups", "3 groups", "2 groups", "1 groups")

sol_4_3
```

```
##                     4 groups 3 groups 2 groups 1 groups
## The number of genes      220       71       17       14
```

5. For each gene, compute the pairwise difference in mean expression values among 4 different cancer groups. Note that there are a total of 6 pairs among 4 cancer groups. Which gene and which cancer group pair have the largest difference in mean expression values? You should report the numerical value of the largest difference along with the gene ID number. Also, identify the corresponding cancer group pair that has the largest difference.

sol 5_1

```r
group_means = sapply(groups,
                     function(grp){
                       colMeans(NCI60$data[cell_group == grp, ])
                     })

pair = combn(1:4, 2) # col = # of pairs, row = combin of group

group_diffs = apply(pair, 2, # each pair goes to arg of func like -> 1/(1, 2), 2/(1, 3)....
      function(pair){
        group_means[, pair[1]] - group_means[, pair[2]]
      })

colnames(group_diffs)=c("A-B", "A-C", "A-D", "B-C", "B-D", "C-D")

ans = which(abs(group_diffs) == max(abs(group_diffs)), arr.ind = TRUE) # arr.ind !!!!

cat("sol 5_1", "\n",
    "The largest difference is", group_diffs[ans[1, 1], ans[1, 2]], "\n",
    "ID of Gene:", ans[1, 1], "\n",
    "Group Pair:", colnames(group_diffs)[ans[1, 2]], "\n")
```

```
## sol 5_1
##  The largest difference is 2.915624
##  ID of Gene: 6415
##  Group Pair: A-B
```

sol 5_2

```r
tmp_df = data.frame(cell_group = cell_group,
                    NCI60$data, check.names = FALSE)

tmp = by(tmp_df[,-1], cell_group, function(x) apply(x, 2, mean))

pair = combn(c("A", "B", "C", "D"), 2)

tmp_diff = apply(pair, 2,function(x){
  tmp[[x[1]]] - tmp[[x[2]]]
})

colnames(tmp_diff)=c("A-B", "A-C", "A-D", "B-C", "B-D", "C-D")

head(tmp_diff)
```

```
##             A-B        A-C         A-D        B-C         B-D         C-D
## 1  0.04624878 0.26187256  0.11562743 0.21562378  0.06937865 -0.14624513
## 2 -0.24875122 0.32874753 -0.21374878 0.57749875  0.03500244 -0.54249631
## 3  0.23218628 0.30718506  0.07156494 0.07499878 -0.16062134 -0.23562012
## 4  0.27593628 0.81968506 -0.22968506 0.54374878 -0.50562134 -1.04937012
## 5 -0.29750122 0.09562256  0.02750243 0.39312378  0.32500366 -0.06812012
## 6 -0.01500122 0.09124756  0.12187744 0.10624878  0.13687866  0.03062988
```

```r
ans = which(abs(group_diffs) == max(abs(group_diffs)), arr.ind = TRUE)

cat("sol 5_2", "\n",
    "The largest difference is", group_diffs[ans[1, 1], ans[1, 2]], "\n",
    "ID of Gene:", ans[1, 1], "\n",
    "Group Pair:", colnames(group_diffs)[ans[1, 2]], "\n")
```

```
## sol 5_2
##  The largest difference is 2.915624
##  ID of Gene: 6415
##  Group Pair: A-B
```

6. Only 9 different cancer cell lines have at least 2 samples. For each of these 9 cell lines, compute the pairwise distance of expression values between two samples (i, j) s.t

$$dist(i,j) = \sqrt{\sum_{k=1}^{p}(x_{ik} - x_{jk})^2}$$

where x_{ik} stands for the gene expression value of the i-th sample and the k-th gene, and $p = 6,830$. Which cell line and two samples have the smallest pairwise distance? Include the nummerical value of the smallest distance with two sample ID number (1 ~ 64) and the name of the corresponding cancer cell line. Note that computation of distance between two samples is limited the same cancer cell line.

```r
filter_cell = names(table(NCI60$labs)[table(NCI60$labs) >= 2])
compr = Inf
ans_samples = NULL
ans_cell = NULL

for(cell in filter_cell){

  cell_id = which(NCI60$labs == cell)

  if(length(cell_id) >= 2){

    for(i in 1:(length(cell_id) - 1)){

      for(j in (i + 1):length(cell_id)){

        dist_cell = sqrt(sum((NCI60$data[cell_id[i], ] - NCI60$data[cell_id[j], ])^2))

        if(dist_cell < compr){

          compr = dist_cell

          ans_samples = c(cell_id[i], cell_id[j])

          ans_cell = cell
        }
      }

    }
  }
}

cat("Smallest dist:", compr, "\n",
    "Sample ID:", ans_samples, "\n",
    "Name of cancer cell line:", ans_cell)
```

```
## Smallest dist: 39.10562
##  Sample ID: 57 58
##  Name of cancer cell line: BREAST
```

```
print("Hello world")
```