DATA MANIPULATION

Load the gene expression data set from a file, "GenExp_data.txt". It consists of expression values for 200 genes belonging to 4 pathways (P1 to P4) observed for 5 consecutive days (D1 to D5).

Gene	D1	D2	D3	D4	D5	Path
Gene1	0.25	0.9	-0.59	-0.48	0.1	P1
Gene2	-0.12	1.66	-1.74	-0.43	-1.31	P1
Gene3	-1.34	0.04	-0.86	-0.59	-0.59	P1
Gene4	0.43	-1.69	-1.88	-0.39	0.1	P1

Prac1: Use of table() function.

```
gen=read.table("GenExp_data.txt",
         header=TRUE,
         stringsAsFactors=FALSE);
str(gen)
head(gen)
head(gen,10)
summary(gen)
colnames(gen)
table(gen$Path);
table(gen$Path,gen$D1);
```

- *head(gen)*: Displays top 6 rows of a data frame
- *summary(gen)*: Summarize the data frame
- *table(gen$Path)*: Count the number of genes in each pathway.
- *table(gen$Path,gen$D1)*: Returns the distribution of different gene expression values across all pathways(gen$Path) on day 1 (gen$D1).

Prac2: Use of apply() function

```
apply(gen[,2:6],1,mean)
apply(gen[,2:6],2,mean)
```

- *apply()* accepts a matrix as 1st argument. The 2nd argument, 1 or 2, suggests row or column wise operation while the 3rd argument is the function that will be applied to the data.
- ***apply(gen[,2:6],1,mean)*:** Each row of *gen[,2:6]* will be passed to *mean* function. It will return mean of each row from the passed matrix.
- ***apply(gen[,2:6],2,mean)*:**Each column will be passed. The result will be column means

Prac3: Filter rows using filter() function of dplyr package.

✓ Load external package dplyr

```
library(dplyr)
```

✓ Filter those genes whose gene expression values on Day1 (D1) > 1.5 (up regulated)

```
filter(gen,D1 >1.5)
```

✓ Filter those genes whose expression on Day1 > 1.5 and belong to Pathway P2.

```
filter(gen,D1 >1.5, Path=="P2")
```

- Using regular conditions: > and ==

✓ Filter genes whose D1 or D2 expression is > 1.5

```
filter(gen,D1 >1.5 | D2 >1.5 )
```

- Using OR condition: |

✓ Filter genes from pathway P2 whose D1 or D2 expression is > 1.5

```
filter(gen,D1 >1.5 | D2 >1.5, Path=="P2" )
```

Task:

1. Filter those genes whose difference of gene expressions on Day1 and Day2 is > 1.

Prac33: Row manipulations

✓ Select rows position wise using slice()

```
slice(gen, 5:10)
```

- Select 5th to 10th rows, here genes

✓ Arrange/reorders rows on the basis of values of columns.

```
arrange(gen,D1,D2)
arrange(gen,D1,desc(D2))
```

- ***arrange(gen,D1,D2)***: Arrange genes first as per expression values on D1 and then D2.
- ***arrange(gen,D1,desc(D2))***: D2 in descending order

Prac34: Sub setting columns using select()

```
select(gen,D1,Path);
select(gen,-D2)
select(gen,-c(D2,D1))
```

- ***select(gen,D1,Path)***: will fetch D1 and Path columns.
- ***select(gen,-D2):*** Fetches all columns excluding D2
- ***select(gen,-c(D2,D1))***: Fetches all columns excluding D1, D2

```
select(gen,starts_with("D"))
select(gen,ends_with("3"))
select(gen,contains("a"))
```

- Select columns by matching characters in their name (start, end, contains)

✓ Extract unique rows for chosen column(s)

```
distinct(select(gen,D1))
distinct(select(gen,D2,D3))
```

- In Line1, ***select()*** fetches the D1 column and ***distinct()*** finds unique values.

Prac35: Create new column(s) using existing

```
mutate(gen,difD1D2=D1-D2)
mutate(gen,difD1D2=D1-D2,pdifD1D2=difD1D2*100)
transmute(gen,difD1D2=D1-D2,pdifD1D2=difD1D2*100)
```

- ***mutate(gen,difD1D2=D1-D2)***: Adds a column named as "difD1D2", which consists of difference between values of D1 and D2 gene expressions, to the existing dataset.
- ***mutate(gen,difD1D2=D1-D2,pdifD1D2=difD1D2*100)*** Use newly created column difD1D2 to create another pdifD1D2. So, you can create and reuse newly created columns.
- ***transmute(gen, difD1D2=D1-D2, pdifD1D2 = difD1D2 * 100)*** Just keeps the new columns only.

Prac36: Grouped operations

```
gengrp=group_by(gen,Path)
summarize(gengrp,n())
summarize(gengrp,n(),mean(D1),sd(D1))
summarize(gengrp,n_distinct(D1))
```

- First create groups using group_by and use the grouped data for data manipulations.
- gengrp: contains the data grouped by pathways.
- *summarize(gengrp,n()):* n() number of genes in each pathway
- *summarize(gengrp,n(),mean(D1),sd(D1)):* Calculate mean and sd of D1 expression values in each pathway.
- *summarize(gengrp,n_distinct(D1)):* Number of unique gene expression values in each pathway.

```
slice(gengrp,1:2)
arrange(gengrp,D1,D2)
```

- *slice()*: select first 2 genes from each group.
- *arrange(gengrp,D1,D2)*: Arrange each pathway groups in terms of expression values of D1 followed by D2

Prac37: Piping/Chaining of data manipulation

```
gen %>%
  group_by(Path) %>%
  filter(D1>1.5) %>%
  summarise(mn=mean(D2))
```

- Begin with whole data in gen,
  - Group data by pathway,
    - For each pathway filter genes whose D1 value>1.5
      - Summarize each pathway by mean(D2)