SESSION 2: ANALYSING GENOMIC AND TRANSCRIPTOMIC DATA IN R

A mini-course

DATA EXPLORATION AND BASIC PLOTTING

Objectives

1. Data exploration with Dplyr
2. Statistics

Create a project

Logical operators in R

== Equal to

!=:Not equal to

< Less than

<= Less than or equal to

> Greater than

>= Greater than or equal to

& And

| Or

! Not

What is dplyr?

- A package that allows for easy data manipulation compared to base R
- Use the 'pipe' %>% to chain functions – imagine a flow of data from left to right

select(): Select columns

filter(): Filter rows

mutate(): Add new variables

arrange(): Sort data

summarise(): Summarise data

Select columns

Positive selection

```
df1_selected = select(df1, col1, col3)
```

OR

```
df1 %>%
    select(col1, col3) -> df1_selected
```

Negative selection

```
df1_selected = select(df1, -col2)
```

OR

```
df1 %>%
    select(-col2) -> df1_selected
```

Filter

```
df1 %>%
      filter(col3 > 2.0) -> df1_filtered

df1 %>%
      filter(col2 < 100 & col3 > 2.0) -> df1_filtered

df1 %>%
      filter(col2 < 100 | col3 > 2.0) -> df1_filtered
```

Mutate

```
df1 %>%
    mutate(new_col = col3 * 2.0) -> df1_mutated
```

```
df1 %>%
    mutate(new_col = col3 * 2.0) %>%
    mutate(col4 = new_col/col3) -> df1_mutated
```

```
df1 %>%
    mutate(new_col = col3 * 2.0, col4 = new_col/col3) -> df1_mutated
```

Arrange

Arrange based on 1 column

```
df1_sorted = arrange(df1, col1)
```

OR

```
arrange(df1, col1) -> df1_sorted
```

OR

```
df1 %>%
    arrange(col1) -> df1_sorted
```

Arrange based on multiple columns

```
df1_sorted = arrange(df1, col1, col2)
```

OR

```
df1 %>%
    arrange(col1, col2) -> df1_sorted
```

Summarise

- One column:

summarise(df1, sum(col1)) or summarise(df1, mean(col2))

- Several columns:

summarise(df1, sum(col1), mean(col2), sd(col3))

- Assign column names:

```
df1 %>%
      summarise(sum_col1 = sum(col1), mean_col2 = mean(col2), sd_col3 =
sd(col3)) -> df1_sum
```

Note: **summarise()**can be used separately, but typically used on grouped data created by **group_by()** - see next slides.

Useful functions for summarisation

- Center: mean(), median()
- Spread: sd(), IQR()
- Range: min(), max(), quantile()
- Position: first(), last()
- Count: n(), n_distinct()
- Logical: any(), all()

Group by

Group by one variable

```
df1 %>% group_by(col2)
```

Summarise data grouped by one variable

```
df1 %>%
    group_by(col2) %>%
    summarise(sum_col4 = sum(col4)) -> df1_grouped
```

Summarise data grouped by two variables

```
df1 %>%
    group_by(col2, col3) %>%
    summarise(sum_col4 = sum(col4)) -> df1_grouped
```

Grouped by new variable and summarised

```
df1 %>%
    group_by(new_col2 = toupper(col2)) %>%
    summarise(sum_col4 = sum(col4)) -> df1_grouped
```

Exercise 1

Objectives

✓Data exploration with Dplyr

2. Statistics

Measures of Central Tendency

- Mean
- Median
- Mode

Mean

$\bar{x}$ *sample mean*

μ *population mean*

$$\overline{X} = \frac{\sum_{i=1}^{n} x_i}{n}$$

mean()

Median

Median()

More robust than the mean

Interquartile range

- Difference between the first quartile (25th percentile) and the third quartile (75th percentile).
- Allows detection of outliers

IQR()

quantile()

Exercise 2

Objectives

- ✓Data exploration with Dplyr
- ✓Statistics and distributions