**R programming**

**EDA**

Basics:

|  |  |
| --- | --- |
| **Functionality** | **Code Syntax** |
| Import Dataset read csv | Variable = read.csv(“dataset/location”)  Variable -> read.csv(“dataset\\location”) |
| Converting String values as factors | Read.csv(“c:\\dataset\\location”,stringsAsFactors = TRUE) ->variable |
| Export dataset write csv 🡪 download dataset | Write.csv(dataset, “c:\\dataset\\location\\to\\get\\save”, row.names = False) |
| Print the dataset | View(dataset) 🡪 V should be capital |
| Data types | Str(dataset) |
| Describe 🡪 statistical values of data | Summary(fish) |
| Null values | Is.na(dataset) |
| Sum of null values | Sum(is.na(dataset)) |
| Null values w.r.t to columns | Colsums(is,na(dataset)) |
| All column names of dataset | Names(dataset)  Colnames(dataset) |
| Row index or names if exists | Row.names(dataset) |
| Count of rows | Nrow(dataset) |
| Count of columns | Ncol(dataset) |
| Length of the dataset | Length(dataset) |
| Required Percentage of the dataset to display | Sample\_frac(dataset, percentage\_in\_fractions) -> sample\_data  Sample\_frac(dataset, 0.7) ->sample\_data  o.7 refers to 70% of the dataset |
| Grouping | Group\_by(dataset,column) ->col\_group  We can use this group to filter other columns.  Ex: Filter(group\_col, sum(col2) |
| Grouping with original code | Group\_by(alpha,gender)->gender\_group  Summarise(gender\_group,mean(tenure))  So here extracting avg of tenure w.r.t gender |

Extractive Values:

|  |  |
| --- | --- |
| Particular row | Dataset[45,] |
| Particular value w.r.t row and column | Dataset[45,3] row🡪45, col🡪3 |
| Sample dataset from the population dataset | Dataset[1:10,2:5]—1 to 10 rows with 2 to 5 col |
| Random sampling extracting random sample data | Sample\_n(dataset, number) -> sample\_data  Sample\_n(dataset,40) -> sample\_dataset  🡪 here 40 means 40 random rows in dataset |
| Particular rows and columns | Dataset[c(2,6,8,13,43),c(2,3,4) rows , columns |
| Get values of the column | Dataset$column\_name |
|  |  |

Library dplyr:

Select:

|  |  |
| --- | --- |
| Install library dplyr | Go to packages and Install library manually |
| Import library | library(dplyr) | Go to packages search - check it |
| Extract values w.r.t columns (we can extract w.r.t columns only) in this dplyr library | W.r.t column index  Select(dataset,1:5) 🡪 1 to 5 columns get printed |
| Extract w.r.t column names | Select(dataset,col1,col3,col7) |
| With starting letter of columns | Select(dataset,start\_with(“A”),start\_with(“b”) |

Filter: General Way

|  |  |
| --- | --- |
| Calling particular column | Dataset[‘col1’] |
| Filtering the col with condition and print | Dataset[dataset$col1 > True, False] -> new\_var  Fish[fish $ col1 > 30, 1] |

Filter:

|  |  |
| --- | --- |
| Filter the column with condition | filter(dataset, col1 > 30) -> new\_var |
| Filter the column with condition method 2 | Dataset %>% filter(condition) ->new\_var  Alpha %>% filter(SeniorCitizen == 1) ->seniors |
| Multiple condition | Filter(dataset, col condition, col condition)  Filter(load, gender == ‘Male’, credit\_score > 780) |

Mutate:

|  |  |
| --- | --- |
| Add new column with condition  2 conditions | Mutate(datset, col = if\_else(col condition, True, False)) -> new\_var  Mutate(dataset, rating = if\_else(credit\_score > 800, Good, Poor)) -> new\_var |
| Multiple conditions (if, else\_if, else) | Mutate(dataset, col = if\_else(condition, True, (if\_else(condition & condition , True, False)))) -> new\_var |
| Encoding (factorising the values) assigning the numbers to the string values | Dataset$col2 = as.factor(dataset$col2) |
| Add new column with manipulation of column | Mutate(dataset,col=col2/100) -> new\_var |

Ggplot:

|  |  |
| --- | --- |
| Import ggplot library | Library(ggplot2) |
| Bar plot | Ggplot(data = dataset, aes(x =col1))+ geom\_bar() |
| Bar plot with fill | Ggplot(data=dataset,aes(x=col1, fill = col2)) + geom\_bar() |
| Histogram plot | Ggplot(data=dataset,aes(x=col1,fill=col2))+geom\_histogram(bin=10) |
| Scatter plot | Ggplot(data=dataset,aes(x=col1, y=col2, col=col3)) +geom\_point() |
| Box plot (Horizontal) | Ggplot(data=dataset,aes(x=col1)) + geom\_boxplot() |
| Box plot (vertical) | Ggplot(data=dataset, aes(x=factor(0), y=col1) +geom\_boxplot() |
| Box plot for multiple variables of same column | Ggplot(data=dataset, aes(x=col1, y=col2) +geom\_boxplot()+state\_summary(fun= ‘mean’, color= ‘red’)+ labs(x = ‘x-lable\_name’, y= ‘y-lable\_name’) |

Plotly:

|  |  |
| --- | --- |
| Bar plot | Plot\_ly(data =dataset, x=~col1, y=~col2, type= ‘bar’) |
| Bar plot with colors | Plot\_ly(data = dataset, x=~col1, y=~col2, color =col3, type= ‘bar’) |
| Scatter plot | Plot\_ly(data=dataset, x=~col1, y=~col2, color = col3, type = ‘scatter’) |
| Histogram plot | Plot\_ly(data=dataset, x=~col1, y=~col2, color = col3, type = ‘histogram’) |
| Scatter plot 3D | Plot\_ly(data=dataset, x=~col1, y=~col2, z=col3, color = col3, type = ‘scatter3d’) |
| Mesh 3D | Plot\_ly(data=dataset, x=~col1, y=~col2, z=col3, color = col3, type = ‘mesh3d’) |