001-About-dataframes

March 19, 2024

1 TP 02 - About Data Frames - 1/2

- Written by Alexandre Gazagnes
- Last update: 2024-02-01
- Based on https://www.w3schools.com/r/default.asp

1.1 Vectors

[ ]:
```
# Create vectors
countries = c("France", "Suisse", "Espagne", "Norway")
ue = c("Oui", "Non", "Oui", "Non")
pop = c(66000, 8000, 48000, 5000)
super = c(643, 41, 505, 323)
```

1.2 Creating DataFrame

[ ]:
```
# Build the data frame
df = data.frame(UE=ue, population=pop, superficie=super, row.names =countries )
df
```

[ ]:
```
# Using str
str(df)  # compact display of the structure of an R object
```

1.3 Accessing to elements

[ ]:
```
# Head and tail
head(df, 2)  # first 2 rows
```

[ ]:
```
tail(df, 2)  # last 2 rows
```

[ ]:
```
sample(df, 2)  # random 2 rows
```

[ ]:
```
# Access to specific elements
df[2,3] # 2nd row, 3rd column
df[2,] # 2nd row, all columns
df[,3] # all rows, 3rd column
df[,] # all rows, all columns
```

[ ]:
```
row_index = c(1, 3)  # => [1, 3]
col_index = c(1, 3)  # => [1, 3]
df[row_index, col_index]  # 1st and 3rd rows, 1st and 3rd columns
```

[ ]:
```
row_index = c(3, 1)  # => [1, 3]
col_index = c(1, 3)  # => [1, 3]
df[row_index, col_index]  # 1st and 3rd rows, 1st and 3rd columns
```

[ ]:
```
df[c(1, 4), c(1, 4)]  # 1st and 4th rows, 1st and 4th columns => error !
```

[ ]:
```
df[1:3]  # 1st to 3rd columns

df[1:2, 1:2]  # 1st to 2nd rows, 1st to 2nd columns
```

[ ]:
```
df["France",]  # all columns of France
df["France"]  # all columns of France
```

[ ]:
```
df["France", "population"] # population of France
df[, "population"] # population of all countries
```

[ ]:
```
# Access to specific colomns
df.population # error :(
df$population # ok
```

[ ]:
```
## Addind / Deleting / Modifiyng Values
```

[ ]:
```
# Adding a new column
df$capitale = c("Paris", "Berne", "Madrid", "Oslo")
df
```

[ ]:
```
# Adding a new column
df$dummy = c("foo", "bar", "tree", "tom")
df
```

[ ]:
```
# Delete a column
df$dummy = NULL
df
```

[ ]:
```
# Delete a column
df$dummy = c("foo", "bar", "tree", "tom")
df = df[, -5] # delete the 5th column
df
```

[ ]:
```
# What about ?
df$dummy=NA
```

[ ]:
```
# delete a row
df[-2,]  # delete the 2nd row
df
```

[ ]:
```
df[-c(2, 3),]  # delete the 2nd and 3rd rows
df
```

[ ]:
```
# Modifying a value
df[1, 1] = "Maybe"  # change the value of the 1st row, 1st column
df
```

1.4 Filtering

[ ]:
```
# Using an indexor
row_indexor = (df$UE == "Oui")
df[row_indexor,] # only countries in the UE
df
```

[ ]:
```
# The oposite
df[!row_indexor,] # only countries not in the UE
```

[ ]:
```
# Better :
df[df$UE == "Oui",]
```

[ ]:
```
# Using Subset
df[df$population > 10000,]  # only countries with a population > 10000
df # do  or do not work ?
```

[ ]:
```
# Better :
tmp = subset(df, population > 10000)  # only countries with a population > 10000
tmp  # do works
```

[ ]:
```
# Better :
tmp = subset(
    df, population > 10000, select=c("UE", "population")
)  # only countries with a population > 10000
tmp  # do works
```

1.5 Answers

[ ]:
```
# Population of espana
df["Espagne", "population"]  # population of Espana
# df[rownames(df) == "Espagne", "population"] # population of Espana
```

[ ]:
```
# Population and superficie of 3 first countries
df[1:3, c("population", "superficie")]  # population and superficie of 3 first␣
 ↪countries
```

[ ]:
```
# Add new col number of habitants per km2
df$hab_km2 = df$population / df$superficie # compute the number of habitants␣
 ↪per km2
df
```

[ ]:
```
# Better :)
df$hab_km2 = round(df$population / df$superficie, 2) # compute the number of␣
 ↪habitants per km2
df
```

[ ]:
```
# Cols 2,3,4 of UE countries
row_indexor = (df$UE == "Oui")
column_indexor = c(2,3,4)
df[row_indexor, column_indexor] # cols 2,3,4 of UE countries
```