Subsetting Tutorial

Jennifer Liberto

9/27/2017

set up a dummy data frame

```
df <- data.frame(SubjectID = c("ABCD","EFGH","IJKL","MNOP"), age = c(12,14,10,16),
gender = c("MALE", "FEMALE", "MALE", "MALE"), Pre.FEV1.Meas = c(2.95,NA,1.52,2.04)
, X = c(1,2,3,4))
print(df)
```

```
##   SubjectID age gender Pre.FEV1.Meas X
## 1      ABCD  12   MALE          2.95 1
## 2      EFGH  14 FEMALE            NA 2
## 3      IJKL  10   MALE          1.52 3
## 4      MNOP  16   MALE          2.04 4
```

subset by row number or column number

by row number

```
df[c(1:3), ]
```

```
##   SubjectID age gender Pre.FEV1.Meas X
## 1      ABCD  12   MALE          2.95 1
## 2      EFGH  14 FEMALE            NA 2
## 3      IJKL  10   MALE          1.52 3
```

```
df[c(1,3:4),]
```

```
##   SubjectID age gender Pre.FEV1.Meas X
## 1      ABCD  12   MALE          2.95 1
## 3      IJKL  10   MALE          1.52 3
## 4      MNOP  16   MALE          2.04 4
```

by column number

```
df[ ,c(1:3)]
```

```
##   SubjectID age gender
## 1      ABCD  12   MALE
## 2      EFGH  14 FEMALE
## 3      IJKL  10   MALE
## 4      MNOP  16   MALE
```

```
df[ ,c(1,3:4)]
```

```
##   SubjectID gender Pre.FEV1.Meas
## 1      ABCD   MALE          2.95
## 2      EFGH FEMALE            NA
## 3      IJKL   MALE          1.52
## 4      MNOP   MALE          2.04
```

subset rows by condition

most times when working with our data, we want to subset based on a condition:

Singluar conditions

```
# age > 12
df[which(df$age > 12), ]
```

```
##   SubjectID age gender Pre.FEV1.Meas X
## 2      EFGH  14 FEMALE            NA 2
## 4      MNOP  16   MALE          2.04 4
```

```
# Pre.FEV1.Meas existing
df[which(!is.na(df$Pre.FEV1.Meas) == TRUE), ]
```

```
##   SubjectID age gender Pre.FEV1.Meas X
## 1      ABCD  12   MALE          2.95 1
## 3      IJKL  10   MALE          1.52 3
## 4      MNOP  16   MALE          2.04 4
```

Combined Conditions

```
# age > 12 and gender is male
df[which(df$age < 12 & df$gender == "MALE"), ]
```

```
##   SubjectID age gender Pre.FEV1.Meas X
## 3      IJKL  10   MALE          1.52 3
```

```
# age >= 14 or gender is female
df[which(df$age >= 14 | df$gender == "FEMALE"), ]
```

```
##   SubjectID age gender Pre.FEV1.Meas X
## 2      EFGH  14 FEMALE            NA 2
## 4      MNOP  16   MALE          2.04 4
```

subset columns by names

Now we have learned to subset the rows, we can subset the columns by including or excluding columns based on their name

```
# only include the three columns in our output
include <- c("SubjectID","age","gender")
df[, names(df) %in% include]
```

```
##   SubjectID age gender
## 1      ABCD  12   MALE
## 2      EFGH  14 FEMALE
## 3      IJKL  10   MALE
## 4      MNOP  16   MALE
```

```
# exclude the two columns in our output
exclude <- c("gender","X")
df[, !(names(df) %in% exclude)]
```

```
##   SubjectID age Pre.FEV1.Meas
## 1      ABCD  12          2.95
## 2      EFGH  14            NA
## 3      IJKL  10          1.52
## 4      MNOP  16          2.04
```

we can also combine our conditional row subsetting with our column subsetting

ex. I want the data frame of SubjectID, age, and gender where gender is always female and age is at least 10.

```
include <- c("SubjectID","age","gender")
df[which(df$age >= 10 & df$gender == "FEMALE"), names(df) %in% include]
```

```
##   SubjectID age gender
## 2      EFGH  14 FEMALE
```

ex. I want the all the data possible but not age or X where the gender is male or FEV1 exists

```
# note the ! before anything means the opposite
exclude <- c("age", "X")
df[which(df$gender != "FEMALE" | is.na(df$Pre.FEV1.Meas) == FALSE), !(names(df) %i
n% exclude)]
```

```
##   SubjectID gender Pre.FEV1.Meas
## 1      ABCD   MALE          2.95
## 3      IJKL   MALE          1.52
## 4      MNOP   MALE          2.04
```

Good luck, you’ll master this skill in no time!