Reading **hflights** data. Throughout the tutorial we shall be making extensive use of it. By default, it is a data frame.

```
library(hflights)

class(hflights)

## [1] "data.frame"
```

Converting it to a data table.

```
mydata <- as.data.table(hflights)
class(mydata)

## [1] "data.table" "data.frame"
```

Task: Filtering 1st 3 rows:

Both the lines will yield to same output (not adding a comma also leads to filtering of rows)

```
mydata[1:3]
mydata[1:3,]
```

Task: Excluding 3rd and 4th row:

```
mydata[!3:4,]
mydata[!(3:4),]
mydata[-(3:4),]
```

Writing in the following manner is an incorrect way and will lead to error

```
mydata[-3:4,]  #incorrect
```

Task: Select all rows except 1 through 5 and 15 through 20

```
exclude_some <- mydata[-c(1:5,15:20)]
exclude_some
```

.N in data.table - It gives the number of rows in a data.table (similar to n() in dplyr)

Task: Select all rows except the first and last

```
not_first_last <- mydata[-c(1,.N)]
not_first_last
```

Task: Removing last 15 rows

```
nrow(mydata)

## [1] 227496
```

```
subset_hflights = mydata[1:(.N-15),]
nrow(subset_hflights)

## [1] 227481
```

Let us try to determine how many unique origin are in the data.

```
unique(mydata$Origin)

## [1] "IAH" "HOU"
```

Task: Filter the rows where Origin is “IAH”

```
#base package
subset_hflights = mydata[mydata$Origin == "IAH",]

#dplyr way
subset_hflights = filter(mydata,Origin == "IAH")

#data.table way
subset_hflights = mydata[Origin == "IAH",]
```

Task: Filter the rows where Origin is “IAH” but Destination is not “BOS”

```
#base package
subset_hflights = mydata[mydata$Origin == "IAH" & mydata$Dest != "BOS",]

#data.table way
subset_hflights = mydata[Origin == "IAH" & Dest != "BOS",]
```

%like% in data.table - It allows you to search for a pattern in a character vector.

Task: Subset all rows where Destination starts with “A"

```
#Base package
subset_hflights = mydata[grepl("^A",mydata$Dest),]


#data.table way
subset_hflights = mydata[Dest %like% "^A"]
unique(subset_hflights$Dest)

## [1] "AUS" "ATL" "ABQ" "ASE" "AEX" "AVL" "AMA" "AGS" "ANC"
```

%between% in data.table – It allows you to search for values in closed interval [a,b]

Task: Subset all rows where Distance is between 500 - 1000 units

```
#base package
subset_hflights = mydata[mydata$Distance >= 500 & mydata$Distance <= 1000,]
```

```
#data.table way
subset_hflights = mydata[Distance %between% c(500,1000)]
sort(unique(subset_hflights$Distance))
```

```
## [1] 501 502 517 519 528 542 562 570 571 595 643 657 666 667 668 670 677 687 689
## [20] 696 744 759 771 772 781 787 788 802 809 816 817 821 828 834 837 838 844 845
## [39] 848 849 851 853 854 861 862 871 878 883 886 912 913 914 925 926 927 928 929
## [58] 935 936 937 956 957 964 965 975 978 979 984 985 986 987
```

%chin% in data.table – It is similar to %in% but is much faster and is used only for character vectors.

Task: Subset all rows where Destination is either “DFW” or “MIA”

```
mydata[Dest %chin% c("DFW","MIA")]
#is faster than
mydata[mydata$Dest %in% c("DFW","MIA")]
```