

DataManipulations

September 20, 2016

```
In [1]: #creating of dataframe by using 4 vectors
fy <- c(2010,2011,2012,2010,2011,2012,2010,2011,2012)
company <- c("Apple","Apple","Apple","Google","Google",
"Google","Microsoft","Microsoft","Microsoft")
revenue <- c(65225,108249,156508,29321,37905,50175,62484,
69943,73723)
profit <- c(14013,25922,41733,8505,9737,10737,18760,23150,
16978)
companiesData <- data.frame(fy, company, revenue, profit)
```

```
In [2]: #head(companiesData)
str(companiesData)
```

```
'data.frame':      9 obs. of  4 variables:
 $ fy      : num  2010 2011 2012 2010 2011 ...
 $ company: Factor w/ 3 levels "Apple","Google",...: 1 1 1 2 2 2 3 3 3
 $ revenue: num  65225 108249 156508 29321 37905 ...
 $ profit  : num  14013 25922 41733 8505 9737 ...
```

```
In [3]: summary(companiesData)
```

	fy	company	revenue	profit
Min.	:2010	Apple :3	Min. : 29321	Min. : 8505
1st Qu.:	2010	Google :3	1st Qu.: 50175	1st Qu.:10737
Median :	2011	Microsoft:3	Median : 65225	Median :16978
Mean :	2011		Mean : 72615	Mean :18837
3rd Qu.:	2012		3rd Qu.: 73723	3rd Qu.:23150
Max.	:2012		Max. :156508	Max. :41733

```
In [4]: #convert fy into factor
companiesData$fy <- factor(companiesData$fy,ordered = TRUE)
```

```
In [5]: # adding a column to an existing dataframe
companiesData$margin <- (companiesData$profit/companiesData$revenue)*100
```

```
In [6]: head(companiesData)
```

fy	company	revenue	profit	margin
2010	Apple	65225	14013	21.48409
2011	Apple	108249	25922	23.94664
2012	Apple	156508	41733	26.66509
2010	Google	29321	8505	29.00651
2011	Google	37905	9737	25.68790
2012	Google	50175	10737	21.39910

```
In [7]: str(companiesData)
```

```
'data.frame':      9 obs. of  5 variables:
 $ fy      : Ord.factor w/ 3 levels "2010"<"2011"<...: 1 2 3 1 2 3 1 2 3
 $ company: Factor w/ 3 levels "Apple","Google",...: 1 1 1 2 2 2 3 3 3
 $ revenue: num  65225 108249 156508 29321 37905 ...
 $ profit  : num  14013 25922 41733 8505 9737 ...
 $ margin  : num  21.5 23.9 26.7 29 25.7 ...
```

```
In [8]: companiesData$margin <- round(companiesData$margin, 1)
```

```
In [9]: head(companiesData)
```

fy	company	revenue	profit	margin
2010	Apple	65225	14013	21.5
2011	Apple	108249	25922	23.9
2012	Apple	156508	41733	26.7
2010	Google	29321	8505	29.0
2011	Google	37905	9737	25.7
2012	Google	50175	10737	21.4

```
In [10]: #Delete perticular column
companiesData <-companiesData[,c(1:4)]
```

```
In [11]: head(companiesData)
```

fy	company	revenue	profit
2010	Apple	65225	14013
2011	Apple	108249	25922
2012	Apple	156508	41733
2010	Google	29321	8505
2011	Google	37905	9737
2012	Google	50175	10737

Syntax 2: R's transform() function:

sum of two columns and store that into a new column with transform(), you would use code such as: **Syntax:** `dataFrame <- transform(dataFrame, newColumn = oldColumn1 + oldColumn2)`

```
In [12]: companiesData <- transform(companiesData, margin =
      round((profit/revenue) * 100, 1))
```

```
In [13]: companiesData
         ?transform
```

fy	company	revenue	profit	margin
2010	Apple	65225	14013	21.5
2011	Apple	108249	25922	23.9
2012	Apple	156508	41733	26.7
2010	Google	29321	8505	29.0
2011	Google	37905	9737	25.7
2012	Google	50175	10737	21.4
2010	Microsoft	62484	18760	30.0
2011	Microsoft	69943	23150	33.1
2012	Microsoft	73723	16978	23.0

```
In [14]: #data manipulate by using apply function
         apply(companiesData, 1, function(x) sum(x))
```

Error in sum(x): invalid 'type' (character) of argument
 Traceback:

```
1. apply(companiesData, 1, function(x) sum(x))
2. FUN(newX[, i], ...)
```

```
In [ ]: apply(companiesData[,c('revenue', 'profit')], 1,
             function(x) sum(x))
```

```
In [ ]: companiesData$sums <- apply(companiesData[,
                                   c('revenue', 'profit')], 1, function(x) sum(x))
```

```
In [ ]: companiesData
```

```
In [ ]: companiesData$margin <- apply(companiesData[,
                                       c('revenue', 'profit')], 1,
                                       function(x) { (x[2]/x[1]) * 100 } )
```

```
In [ ]: companiesData
```

```
In [ ]: highestMargin <- max(companiesData$margin)
         highestMargin
```

```
In [ ]: highestMargin <- companiesData[companiesData$margin == max(companiesData$ma
                                         highestMargin
                                         companiesData$margin == max(companiesData$margin)
```

```
In [ ]: #?subset
```

```
highestMargin <- subset(companiesData, margin == max(margin))
#airquality
#subset(airquality, Temp > 80, select = c(Ozone, Temp))
```

```
In [ ]: highestMargin
```

1 Dplyr

syntax: ddply(mydata, c('column name of a factor to group by',
'column name of the second factor to group by'), summarize
OR transform, newcolumn = myfunction(column name(s) I
want the function to act upon))

```
In [ ]: library("plyr")
        companiesData
```

```
In [ ]: highestProfitMargins <- ddply(companiesData, 'company', summarize, bestMargin = max(margin))
        highestProfitMargins
```

```
In [ ]: highestProfitMargins <- ddply(companiesData,
        'company', transform, bestMargin = max(margin))
```

```
In [ ]: highestProfitMargins
```

```
In [ ]:
```