Factors and Summarizing data

> Created by Woo Hyun (Ray) Hwang

Factors and summarizing data

Factors

Factor is a type of vector that characters are stored. Used for categorical variables. Use the `factor()` function to make a factor. Use `levels()` to check the categories.

```
set.seed(1)
alpha <- sample(c("A", "B", "C"), 25, replace=T) #n=25 sample with replacement
f <- factor(alpha); str(f) #shows the levels
z <- sample(1:25, 25, replace=T)
g <- factor(z); str(g)
str(data.frame(f=f, g=g)) #f and g are both factors
```

`table()`

Creates a frequency table according to the factors.

```
table(f)
table(f, g) #frequency table with both factors from f and g
addmargins(table(f)) #adds the sum margin
addmargins(table(f, g))
tab <- table(f, g)
class(tab); dim(tab); rownames(tab); colnames(tab)
```

`tapply()`

`tapply(x, f, function)` seperates x into f factors, and applies function.

```
set.seed(2); x <- round(rnorm(25, 50, 10))
head(data.frame(x=x, f=f), 7)
tapply(x, f, median) #applies median to x by factors of f
tapply(x, f, function(r) max(r)-min(r))
```

`split()`

`split(df, f)` : splits the df into f factors.

```
split(data.frame(x=x, z=z), f) #A, B, C / data frames within these factors
s <- split(data.frame(x=x, z=z), f); class(s)
s$A #retrieve data frame under factor A
class(s$A) #data frame
```

```
#Then use the sapply() function
sapply(s, apply, 2, median) #apply the median function to the cols of s
#this gives the same result as tapply(x, f, median)
```

`aggregate()`

`aggregate(x, list(f, g), function)` : splits x into a combination of f and g, then applies the function. Results are given in data frame formate with a random name for the new column.

```
aggregate(data.frame(x=x, z=z), list(f), sum) #only f factors
aggregate(data.frame(x=x, z=z), list(f, g), sum) #both f and g (ABC 123)
```

`cut()`

`cut(x, breaks)` function ‘cuts’ numeric x into groups of ‘breaks’, factorizing them. This is also known as ‘binning’.

```
x <- runif(100, 0, 10)
y <- 5 + 0.5*(x-5) + rnorm(100)
x.cut <- cut(x, 0:10) #0~10 classes (breaks)
head(cbind(x, x.cut), 5)
y.local <- aggregate(y, list(x.cut), mean)
  #applied mean to group of breaks from y
y.local
plot(x, y, ylim=c(0, 10), main="x vs. y")
segments(0:9, y.local$x, 1:10, y.local$x, lwd=2)
abline(v=1:9, lty="dotted", col="blue")
```

Application to MLB

```
library(Lahman)
data("Salaries")
str(Salaries)
#Get the 2013 data
salaries.2013 <- subset(Salaries, yearID==2013)
attach(salaries.2013)
table(teamID)
tab <- table(teamID)
teamID.1 <- factor(teamID)
levels(teamID.1) <- names(tab)[tab>0] #getting rid of those that have 0 value
table(teamID.1)
tapply(salary/1000, teamID.1, median) #apply median to salary/1000 by teamID.1
barplot(tapply(salary/1000, teamID.1, median), horiz=T, las=1,
        main="Median salary (in 1,000) of 2013")
barplot(sort(tapply(salary/1000, teamID.1, median)), horiz=T, las=1,
        main="Median salary (in 1,000) of 2013") #from largest to smallest
```

Spliting the teams, calculation for individual players in certain team

```
salaries.2013.s <- split(salaries.2013, teamID.1)
class(salaries.2013.s); names(salaries.2013.s)
detach(salaries.2013)
attach(salaries.2013.s$LAN)
names(salary) <- playerID
barplot(sort(salary/1000), horiz=T, las=1, main="LA Dodgers salary (in 1,000)")
```