Data frames

The data frame is a seminal concept in R. Most statistical operations expect one and they are the most common way to pass data in and out of R.
Although critical to understand, this is very, very easy to get.
What's a data frame? It's a table. That's it.

```r
df = read.csv("../Data/Rwanda_frame.csv")
```

Data frames

- Creating
- Referencing
- Ordering
- Adding new columns
- Subsetting
- Summarizing
- Merging

Creating a data frame - 1

```r
set.seed(1234)
Province = rep(c("Kigali", "Sud", "Ouest", "Nord", "Est")
               , 10)
N = length(Province)
Age = rnorm(N, mean = 40, sd=15)
Height = rnorm(N, mean=160, sd=10)
Weight = rnorm(N, mean=60, sd = 10)
Gender = c("Male", "Female") [sample(c(1,2), N
                                     , replace=TRUE)]
```

Creating a data frame - 2

```r
df = data.frame(Province, Age, Gender
                , Height, Weight, stringsAsFactors=FALSE)
```

Basic properties of a data frame - 1

```
summary(df)

##     Province              Age             Gender         
##  Length:50          Min.   : 4.81   Length:50         
##  Class :character   1st Qu.:25.18   Class :character  
##  Mode  :character   Median :31.97   Mode  :character  
##                     Mean   :33.20                     
##                     3rd Qu.:39.42                     
##                     Max.   :76.24                     
##      Height        Weight    
##  Min.   :142   Min.   :40.5  
##  1st Qu.:154   1st Qu.:54.1  
##  Median :160   Median :60.7  
##  Mean   :161   Mean   :60.2  
##  3rd Qu.:168   3rd Qu.:65.1  
##  Max.   :185   Max.   :80.6  
```

Basic properties of a data frame - 2

```
names(df)

## [1] "Province" "Age"      "Gender"   "Height"   "Weight'

colnames(df)

## [1] "Province" "Age"      "Gender"   "Height"   "Weight'
```

Basic properties of a data frame - 3

```
length(df)

## [1] 5

dim(df)

## [1] 50  5

nrow(df)

## [1] 50

ncol(df)

## [1] 5
```

Basic properties of a data frame - 4

```
head(df)
```

```
##   Province    Age Gender Height Weight
## 1   Kigali 21.894   Male  141.9  64.15
## 2      Sud 44.161 Female  154.2  55.25
## 3    Ouest 56.267 Female  148.9  60.66
## 4     Nord  4.815   Male  149.9  54.98
## 5      Est 46.437 Female  158.4  51.74
## 6   Kigali 47.591   Male  165.6  61.67
```

```
head(df, 2)
```

```
##   Province   Age Gender Height Weight
## 1   Kigali 21.89   Male  141.9  64.15
## 2      Sud 44.16 Female  154.2  55.25
```

```
tail(df)
```

```
##    Province   Age Gender Height Weight
## 45      Est 25.08   Male  155.0  65.14
## 46   Kigali 25.47   Male  163.6  63.99
## 47      Sud 23.39   Male  148.7  76.63
## 48    Ouest 21.22 Female  168.8  62.76
## 49     Nord 32.14 Female  169.7  65.06
## 50      Est 32.55   Male  181.2  63.48
```

Referencing

Very similar to referencing a vector, but now with row and column dimensions.

```
df[2, 3]
df[2]
df[2, ]
df[2, -1]
```

More referencing

```r
# The £ operator may be used to select a single column
df$Age
# Columns of a data frame may be treated as vectors
df$Age[3]
df[2:4, 1:2]
df[, "Age"]
df[, c("Age", "Province")]
```

Ordering

```
order(df$Age)

##  [1]  4 37 35 26 38 48  1 36 47 42 28 12 45 46 30 18 10
## [19] 19 13 33 25  7  9  8 49 17 34 50 22 11 32 40 23 39
## [37] 16 29 14 21  2  5 24  6 27 15  3 31 41 20

df = df[order(df$Age), ]
```

Altering and adding columns

```
df$BMI = df$Weight/(df$Height/100)^2

df$BMI = with(df, Weight/(Height/100)^2)
```

Eliminating columns

```r
df$BMI = NULL
df = df[, 1:2]
```

rbind, cbind

```
dfA = df[1:10, ]
dfB = df[11:20, ]
rbind(dfA, dfB)
dfC = dfA[, 1:2]
cbind(dfA, dfC)
```

Merging

```
df = data.frame(Province, Age, Gender
                , Height, Weight, stringsAsFactors=FALSE)
dfBeerIntake = data.frame(Province =c("Kigali", "Sud"
                                      , "Ouest", "Est", "No
                          , BeerIntake = c(400, 200, 300
                                           , 250, 300))
df = merge(df, dfBeerIntake)
```

Basically equivalent to a JOIN in SQL.

Altering column names

```
df$BeerPerGram = with(df, BeerIntake/Weight)
names(df)

## [1] "Province"    "Age"         "Gender"
## [4] "Height"      "Weight"      "BeerIntake"
## [7] "BeerPerGram"

colnames(df)[7] = "BeerPerKg"
colnames(df)

## [1] "Province"   "Age"        "Gender"     "Height"
## [5] "Weight"     "BeerIntake" "BeerPerKg"
```

Subsetting - The easy way

```r
dfKigali = subset(df, Province == "Kigali")
dfOld = subset(df, Age > 50)
```

Subsetting - The hard(ish) way

```
dfKigali = df[df$Province == "Kigali", ]
dfOld = df[df$Age > 50, ]
```

Subsetting - Yet another way

```r
whichProvince = df$Province == "Kigali"
dfKigali = df[whichProvince, ]

whichAge = df$Age > 50
dfOld = df[whichAge, ]
```

Subsetting

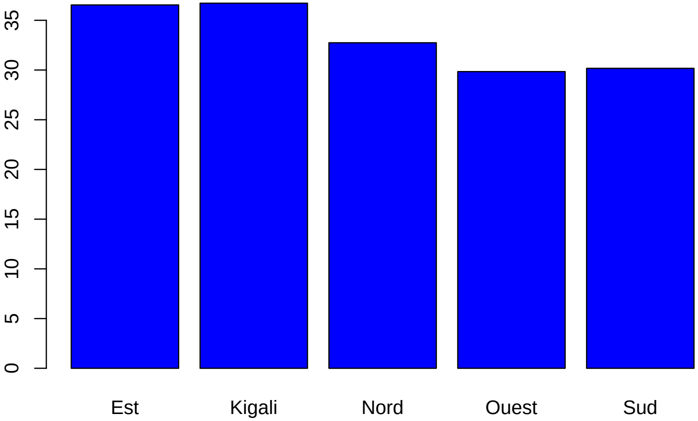
I use each of these three methods routinely. They're all good.

Summarizing

```
mean(df$Age)

## [1] 33.2

mean(df$Age[df$Province == "Kigali"])

## [1] 36.72

aggregate(df[, -c(1, 3)], list(df$Province), mean)

##   Group.1   Age Height Weight BeerIntake BeerPerKg
## 1     Est 36.55  162.0  59.76        250     4.275
## 2  Kigali 36.72  161.9  61.82        400     6.505
## 3    Nord 32.74  163.8  63.48        300     4.763
## 4   Ouest 29.84  159.2  57.63        300     5.413
## 5     Sud 30.17  160.1  58.32        200     3.507
```

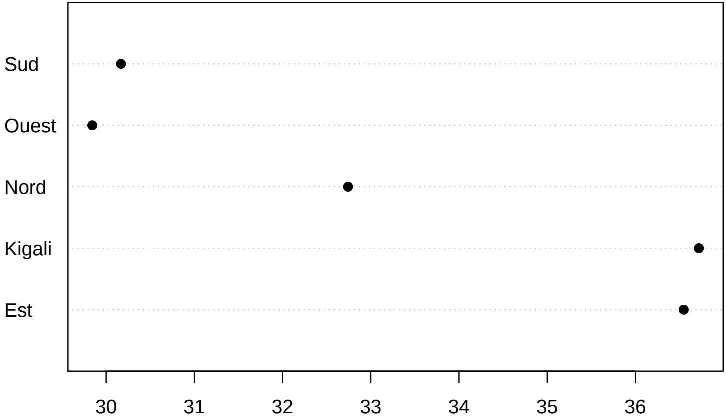
Summarizing visually - 1

```
dfByProvince = aggregate(df$Age, list(df$Province), mean)
colnames(dfByProvince) = c("Province", "Age")
barplot(dfByProvince$Age, names.arg = dfByProvince$Province,
    col = "blue")
```



Summarizing visually - 2

```r
dotchart(dfByProvince$Age, dfByProvince$Province, pch = 19)
```



Advanced data frame tools

- plyr
- reshape2
- data.table
- doBy

Questions

- Construct a random data frame where the weights differ for male and female
- Which subject has the largest weight? The largest BMI?
- Create a data frame for females only