Getting Started in R and data.table

Saghir Bashir
ilustat.com
17th May 2019

Relax

ExpeRiment

Make Mistakes

Learn

Enjoy

Outline

Basic R Concepts

Chick Weight Data

`data.table`

Summary

Basic R Concepts

Assign Values to Objects

Assign (<-) values to objects

```
age <- 10
age
```

```
## [1] 10
```

```
Name <- "Leo"
Name
```

```
## [1] "Leo"
```

R is case sensitive

```
AgE <- 50
AgE
```

```
## [1] 50
```

```
age
```

```
## [1] 10
```

R as a Calculator

```
3+2
```

```
## [1] 5
```

```
(8*5)/7
```

```
## [1] 5.714286
```

```
sqrt(5)
```

```
## [1] 2.236068
```

```
5**2
```

```
## [1] 25
```

```
log(5)
```

```
## [1] 1.609438
```

```
exp(1)
```

```
## [1] 2.718282
```

```
round(15.91531, 2)
```

```
## [1] 15.92
```

```
signif(7461, 2)
```

```
## [1] 7500
```

More R Concepts

Assign values to objects

```
roomLength <- 7.8
roomWidth <- 6.4
```

Create a new object from existing objects

```
roomArea <- roomLength * roomWidth
roomArea
```

```
## [1] 49.92
```

Vectors

So far we have seen "scalar" objects (e.g. age, roomArea)

By combining scalars we can create vectors using `c()`

```
a1 <- c(102, -14, 15, 89, 3, 75)
a1
```

```
## [1] 102 -14  15  89   3  75
```

```
b1 <- c("Portugal", "Brazil", "Angola", "Mozambique")
b1
```

```
## [1] "Portugal"   "Brazil"     "Angola"     "Mozambique"
```

Vectors: Other Methods

```
c1 <- 1:12
c1
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12
```

```
d1 <- rep(c(7:9), 4)
d1
```

```
##  [1] 7 8 9 7 8 9 7 8 9 7 8 9
```

```
e1 <- seq(from=10, to=12, by=0.5)
e1
```

```
## [1] 10.0 10.5 11.0 11.5 12.0
```

Vector Operations

You can do calculations on vectors (essential to R thinking)

```
e1*2
```

```
## [1] 20 21 22 23 24
```

```
round(sqrt(e1), 2)
```

```
## [1] 3.16 3.24 3.32 3.39 3.46
```

Missing Values: NA

NA is used to represent missing values in R

```
Year <- c(1996, NA, 2002, 1985, 1962, 1998)
Year
```

```
## [1] 1996   NA 2002 1985 1962 1998
```

```
city <- c("Lisbon", "Warsaw", NA, "Paris", "Rome", NA)
city
```

```
## [1] "Lisbon" "Warsaw" NA       "Paris"  "Rome"   NA
```

Comments

Comments are useful for the future you.

Comments start from a "#"

```
# This comment line is ignored by R
Year
```

```
## [1] 1996   NA 2002 1985 1962 1998
```

```
city  # This comment starts after the hash
```

```
## [1] "Lisbon" "Warsaw" NA       "Paris"  "Rome"   NA
```

Managing Objects

Use `ls()` to list the objects you have stored in R

```
ls()
```

```
##  [1] "a1"         "age"        "AgE"        "b1"         "c1"
##  [6] "city"       "d1"         "e1"         "Name"       "roomArea"
## [11] "roomLength" "roomWidth"  "Year"
```

Use `rm()` to remove objects

```
rm(a1, b1, c1, d1, e1, AgE)
ls()
```

```
## [1] "age"        "city"       "Name"       "roomArea"   "roomLength"
## [6] "roomWidth"  "Year"
```

Functions

We have already seen some R functions (e.g, `mean()`, `seq()`)

A function has a name and list of arguments

```
# Name: seq; Arguments: from, to & by
seq(from = 6, to = 9, by = 0.5)
```

```
## [1] 6.0 6.5 7.0 7.5 8.0 8.5 9.0
```

```
# Name: seq; Arguments: from, to & length.out
seq(from = 8, to = 12, length.out = 9)
```

```
## [1]  8.0  8.5  9.0  9.5 10.0 10.5 11.0 11.5 12.0
```

R Packages

R Packages are like apps to extend R

They contain funtions, data and documentation

For example, data.table is an R package

R Demo & Exercises

Practical: 50 minutes

Work with your neighbours through the

"Exercise Sheet: Base R Concepts"

Chick Weight Data

Chick Weight Data

Four variables: Chick ID, Diet, Time (days) and Weight (g)

578 observations (50 chicks & 4 diets)

First 6 observations

Chick	Diet	Time	weight
1	1	0	42
1	1	2	51
1	1	4	59
1	1	6	64
1	1	8	76
1	1	10	93

Last 6 observations

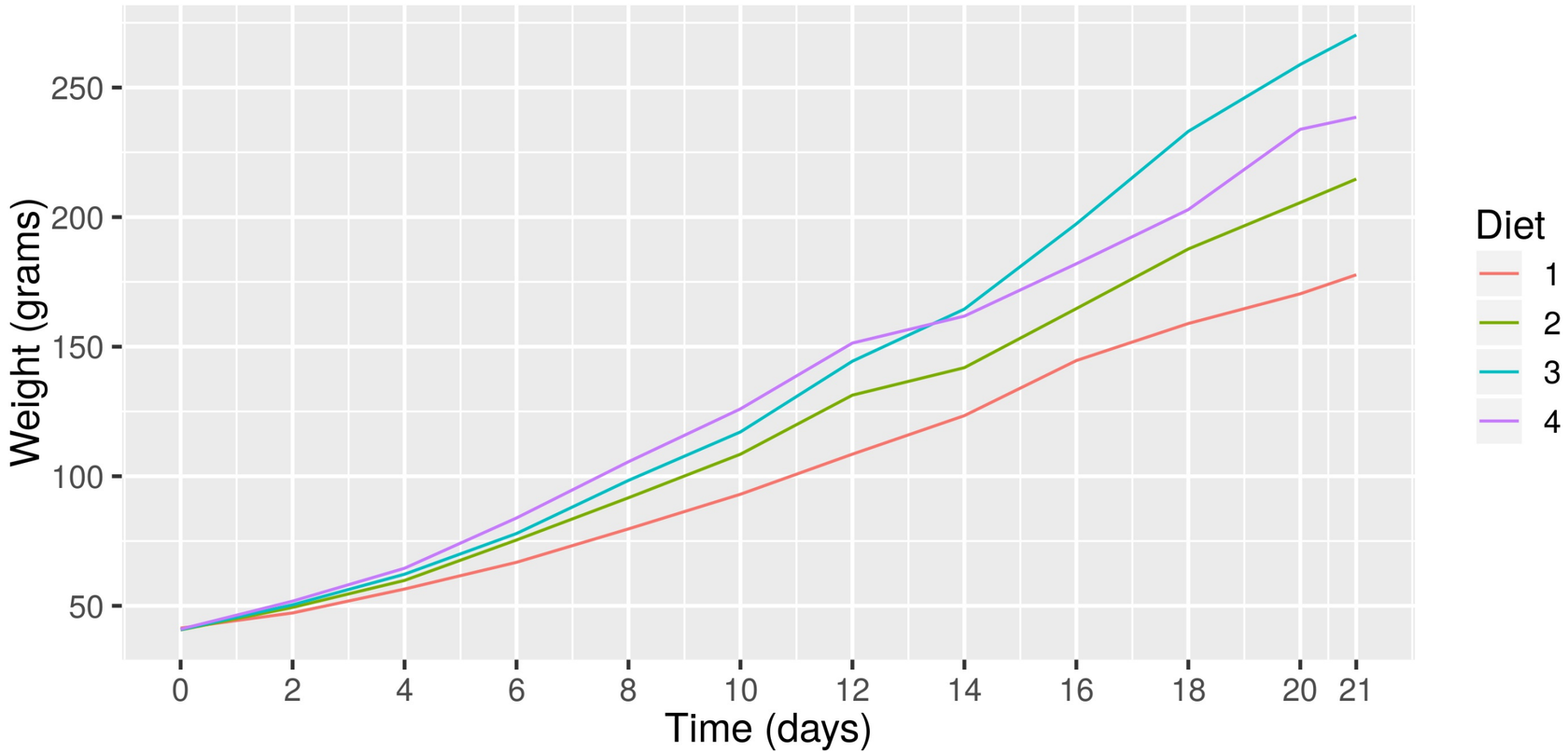
Chick	Diet	Time	weight
50	4	12	155
50	4	14	175
50	4	16	205
50	4	18	234
50	4	20	264
50	4	21	264

Question of Interest

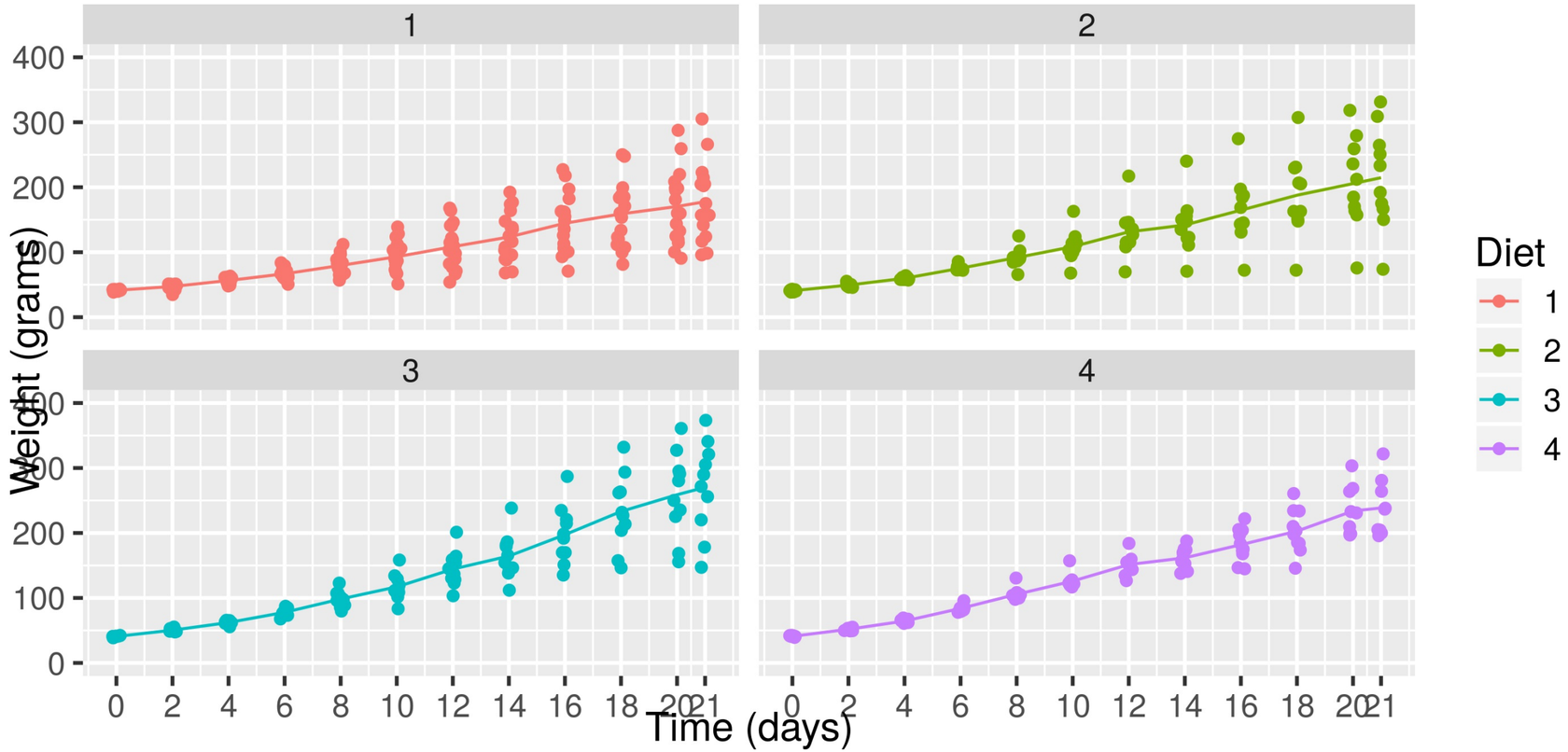
"The body weights of the chicks were measured at birth and every second day thereafter until day 20. They were also measured on day 21. There were four groups of chicks on different protein diets."

Which of the fours diets leads to the most body weight gain?

Diet Means Over Time

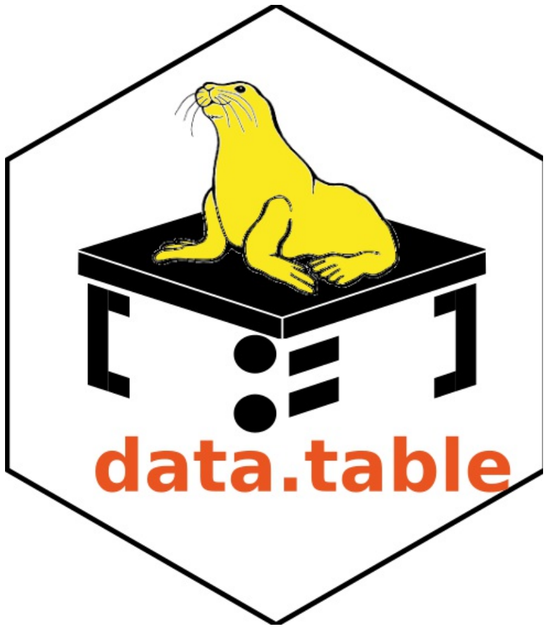


Diet Means Over Time



Summary Statistics: Days 0 & 21

Group	Time	N	Mean	SDev	Median	Min	Max
Diet 1	0	20	41.4	1.0	41.0	39	43
	21	16	177.8	58.7	166.0	96	305
Diet 2	0	10	40.7	1.5	40.5	39	43
	21	10	214.7	78.1	212.5	74	331
Diet 3	0	10	40.8	1.0	41.0	39	42
	21	10	270.3	71.6	281.0	147	373
Diet 4	0	10	41.0	1.1	41.0	39	42
	21	9	238.6	43.3	237.0	196	322


data.table

Install `data.table` **package**

```
install.packages("data.table")
```

Import the data: `fread()`

First load `data.table` using the `library()` function

```
library(data.table)
cw <- fread("ChickWeight.csv")
cw
```

```
##       Chick Diet Time weight
##    1:     1    1    0     42
##    2:     1    1    2     51
##    3:     1    1    4     59
##    4:     1    1    6     64
##    5:     1    1    8     76
##   ---
## 574:    50    4   14    175
## 575:    50    4   16    205
## 576:    50    4   18    234
## 577:    50    4   20    264
## 578:    50    4   21    264
```

Data Structure

```
str(cw)
```

```
## Classes 'data.table' and 'data.frame':    578 obs. of  4 variables:
##  $ Chick : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Diet  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Time  : int  0 2 4 6 8 10 12 14 16 18 ...
##  $ weight: int  42 51 59 64 76 93 106 125 149 171 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
dt[i, j, by]
```

```
dt[rows, columns, groups]
```

`dt[i]`: Filter Rows


```
cw[Time==21]
```

```
##     Chick Diet Time weight
##  1:     1    1   21    205
##  2:     2    1   21    215
##  3:     3    1   21    202
##  4:     4    1   21    157
##  5:     5    1   21    223
## ---
## 41:    46    4   21    238
## 42:    47    4   21    205
## 43:    48    4   21    322
## 44:    49    4   21    237
## 45:    50    4   21    264
```

```
cw
```

```
##      Chick Diet Time weight
##   1:     1    1    0     42
##   2:     1    1    2     51
##   3:     1    1    4     59
##   4:     1    1    6     64
##   5:     1    1    8     76
##  ---
## 574:    50    4   14    175
## 575:    50    4   16    205
## 576:    50    4   18    234
## 577:    50    4   20    264
## 578:    50    4   21    264
```

`cw` did not change!

`dt[i]`: Filter Rows


```
cw[Time==21 & weight>=300]
```

```
##    Chick Diet Time weight
## 1:     7    1   21    305
## 2:    21    2   21    331
## 3:    29    2   21    309
## 4:    32    3   21    305
## 5:    34    3   21    341
## 6:    35    3   21    373
## 7:    40    3   21    321
## 8:    48    4   21    322
```

```
cw
```

```
##       Chick Diet Time weight
##    1:     1    1    0     42
##    2:     1    1    2     51
##    3:     1    1    4     59
##    4:     1    1    6     64
##    5:     1    1    8     76
##   ---
## 574:    50    4   14    175
## 575:    50    4   16    205
## 576:    50    4   18    234
## 577:    50    4   20    264
## 578:    50    4   21    264
```

`cw` did not change!

`dt[i]`: Order Rows


```
cw[order(weight)]
```

```
##       Chick Diet Time weight
##    1:    18    1    2     35
##    2:     3    1    2     39
##    3:    18    1    0     39
##    4:    27    2    0     39
##    5:    28    2    0     39
##   ---
## 574:    21    2   21    331
## 575:    35    3   18    332
## 576:    34    3   21    341
## 577:    35    3   20    361
## 578:    35    3   21    373
```

```
cw
```

```
##       Chick Diet Time weight
##    1:     1    1    0     42
##    2:     1    1    2     51
##    3:     1    1    4     59
##    4:     1    1    6     64
##    5:     1    1    8     76
##   ---
## 574:    50    4   14    175
## 575:    50    4   16    205
## 576:    50    4   18    234
## 577:    50    4   20    264
## 578:    50    4   21    264
```

`cw` did not change!

`dt[i]`: Order Rows - Descending


```
cw[order(-weight)]
```

```
##       Chick Diet Time weight
##    1:    35    3   21    373
##    2:    35    3   20    361
##    3:    34    3   21    341
##    4:    35    3   18    332
##    5:    21    2   21    331
##   ---
## 574:    29    2    0     39
## 575:    33    3    0     39
## 576:    36    3    0     39
## 577:    48    4    0     39
## 578:    18    1    2     35
```

```
cw
```

```
##       Chick Diet Time weight
##    1:     1    1    0     42
##    2:     1    1    2     51
##    3:     1    1    4     59
##    4:     1    1    6     64
##    5:     1    1    8     76
##   ---
## 574:    50    4   14    175
## 575:    50    4   16    205
## 576:    50    4   18    234
## 577:    50    4   20    264
## 578:    50    4   21    264
```

`cw` did not change!

`dt[, j]`: Add New Column

`:=` adds a column (`cw` changes)

```
cw[, weightKg := weight/1000]
cw
```

```
##       Chick Diet Time weight weightKg
##    1:     1    1    0     42    0.042
##    2:     1    1    2     51    0.051
##    3:     1    1    4     59    0.059
##    4:     1    1    6     64    0.064
##    5:     1    1    8     76    0.076
##   ---
## 574:    50    4   14    175    0.175
## 575:    50    4   16    205    0.205
## 576:    50    4   18    234    0.234
## 577:    50    4   20    264    0.264
## 578:    50    4   21    264    0.264
```

dt[, j]: Add Multiple Columns

Use := to add multiple columns (cw changes)

```
cw[, `:=`(Hours=24*Time, LongD=paste0("Diet ", Diet))]
cw
```

```
##       Chick Diet Time weight weightKg Hours  LongD
##   1:      1    1    0     42    0.042     0 Diet 1
##   2:      1    1    2     51    0.051    48 Diet 1
##   3:      1    1    4     59    0.059    96 Diet 1
##   4:      1    1    6     64    0.064   144 Diet 1
##   5:      1    1    8     76    0.076   192 Diet 1
##  ---
## 574:     50    4   14    175    0.175   336 Diet 4
## 575:     50    4   16    205    0.205   384 Diet 4
## 576:     50    4   18    234    0.234   432 Diet 4
## 577:     50    4   20    264    0.264   480 Diet 4
## 578:     50    4   21    264    0.264   504 Diet 4
```

Drop Columns


```
# Drop One Column
cw[, weightKg:=NULL]
cw
```

```
##       Chick Diet Time weight Hours  LongD
##    1:     1    1    0     42     0 Diet 1
##    2:     1    1    2     51    48 Diet 1
##    3:     1    1    4     59    96 Diet 1
##    4:     1    1    6     64   144 Diet 1
##    5:     1    1    8     76   192 Diet 1
##   ---
## 574:    50    4   14    175   336 Diet 4
## 575:    50    4   16    205   384 Diet 4
## 576:    50    4   18    234   432 Diet 4
## 577:    50    4   20    264   480 Diet 4
## 578:    50    4   21    264   504 Diet 4
```

```
# Drop Multiple Columns
cw[, c("Diet", "Hours"):=NULL]
cw
```

```
##       Chick Time weight  LongD
##    1:     1    0     42 Diet 1
##    2:     1    2     51 Diet 1
##    3:     1    4     59 Diet 1
##    4:     1    6     64 Diet 1
##    5:     1    8     76 Diet 1
##   ---
## 574:    50   14    175 Diet 4
## 575:    50   16    205 Diet 4
## 576:    50   18    234 Diet 4
## 577:    50   20    264 Diet 4
## 578:    50   21    264 Diet 4
```

Dropping columns changes `cw`

`setnames()`: **Renaming Columns**


```
# Current cw
cw
```

```
##      Chick Time weight  LongD
##   1:     1    0     42 Diet 1
##   2:     1    2     51 Diet 1
##   3:     1    4     59 Diet 1
##   4:     1    6     64 Diet 1
##   5:     1    8     76 Diet 1
##  ---
## 574:    50   14    175 Diet 4
## 575:    50   16    205 Diet 4
## 576:    50   18    234 Diet 4
## 577:    50   20    264 Diet 4
## 578:    50   21    264 Diet 4
```

```
# Rename: weight to Weight & LongD to Group
setnames(cw, c("weight", "LongD"),
             c("Weight", "Group"))
cw
```

```
##      Chick Time Weight  Group
##   1:     1    0     42 Diet 1
##   2:     1    2     51 Diet 1
##   3:     1    4     59 Diet 1
##   4:     1    6     64 Diet 1
##   5:     1    8     76 Diet 1
##  ---
## 574:    50   14    175 Diet 4
## 575:    50   16    205 Diet 4
## 576:    50   18    234 Diet 4
## 577:    50   20    264 Diet 4
## 578:    50   21    264 Diet 4
```

`setnames()` **changes** `cw` 😀

`dt[, j]`: Column Order


```
cw[, .(Weight, Time, Group, Chick)]
```

```
##      Weight Time  Group Chick
##   1:     42    0 Diet 1     1
##   2:     51    2 Diet 1     1
##   3:     59    4 Diet 1     1
##   4:     64    6 Diet 1     1
##   5:     76    8 Diet 1     1
##  ---
## 574:    175   14 Diet 4    50
## 575:    205   16 Diet 4    50
## 576:    234   18 Diet 4    50
## 577:    264   20 Diet 4    50
## 578:    264   21 Diet 4    50
```

```
cw
```

```
##      Chick Time Weight  Group
##   1:     1    0     42 Diet 1
##   2:     1    2     51 Diet 1
##   3:     1    4     59 Diet 1
##   4:     1    6     64 Diet 1
##   5:     1    8     76 Diet 1
##  ---
## 574:    50   14    175 Diet 4
## 575:    50   16    205 Diet 4
## 576:    50   18    234 Diet 4
## 577:    50   20    264 Diet 4
## 578:    50   21    264 Diet 4
```

`cw` did not change!

`setcolorder()`: Column Order


```
setcolorder(cw, c("Group", "Chick", "Time", "Weight"))
cw
```

```
##       Group Chick Time Weight
##   1: Diet 1     1    0     42
##   2: Diet 1     1    2     51
##   3: Diet 1     1    4     59
##   4: Diet 1     1    6     64
##   5: Diet 1     1    8     76
##  ---
## 574: Diet 4    50   14    175
## 575: Diet 4    50   16    205
## 576: Diet 4    50   18    234
## 577: Diet 4    50   20    264
## 578: Diet 4    50   21    264
```

`setcolorder()` **changes** `cw` 😀

`dt[, j, by]`: **Summary Statistics**


```
cw[, .(Mean=mean(Weight), SD=sd(Weight))
   , by = .(Group, Time)]
```

```
##      Group Time      Mean         SD
##  1: Diet 1    0  41.40000  0.9947229
##  2: Diet 1    2  47.25000  4.2781575
##  3: Diet 1    4  56.47368  4.1280668
##  4: Diet 1    6  66.78947  7.7572829
##  5: Diet 1    8  79.68421 13.7761978
## ---
## 44: Diet 4   14 161.80000 15.7324859
## 45: Diet 4   16 182.00000 25.3026130
## 46: Diet 4   18 202.90000 33.5574135
## 47: Diet 4   20 233.88889 37.5680863
## 48: Diet 4   21 238.55556 43.3477540
```

```
# `cw` did not change!
cw
```

```
##       Group Chick Time Weight
##   1: Diet 1     1    0     42
##   2: Diet 1     1    2     51
##   3: Diet 1     1    4     59
##   4: Diet 1     1    6     64
##   5: Diet 1     1    8     76
##  ---
## 574: Diet 4    50   14    175
## 575: Diet 4    50   16    205
## 576: Diet 4    50   18    234
## 577: Diet 4    50   20    264
## 578: Diet 4    50   21    264
```

`dt[, j, by]`: Summary Statistics


```
cwSum <- cw[, .(Mean = mean(Weight), SD = sd(Weight)), by = .(Group, Time)]
cwSum
```

```
##      Group Time      Mean         SD
##  1: Diet 1    0  41.40000  0.9947229
##  2: Diet 1    2  47.25000  4.2781575
##  3: Diet 1    4  56.47368  4.1280668
##  4: Diet 1    6  66.78947  7.7572829
##  5: Diet 1    8  79.68421 13.7761978
## ---
## 44: Diet 4   14 161.80000 15.7324859
## 45: Diet 4   16 182.00000 25.3026130
## 46: Diet 4   18 202.90000 33.5574135
## 47: Diet 4   20 233.88889 37.5680863
## 48: Diet 4   21 238.55556 43.3477540
```

Assigned summary statistics to `cwSum` 😀

Chaining: Multiple Steps

To do multiple steps together use `cw[...][...][...]...`

```
cw[Time %in% c(0, 21)][
  , .(N = .N, Mean = mean(Weight), SD = sd(Weight)), by = .(Group, Time)][
    order(Group, Time)]
```

```
##     Group Time  N  Mean     SD
## 1: Diet 1    0 20  41.4  0.995
## 2: Diet 1   21 16 177.8 58.702
## 3: Diet 2    0 10  40.7  1.494
## 4: Diet 2   21 10 214.7 78.138
## 5: Diet 3    0 10  40.8  1.033
## 6: Diet 3   21 10 270.3 71.623
## 7: Diet 4    0 10  41.0  1.054
## 8: Diet 4   21  9 238.6 43.348
```

Final Summary Statistics


```
cws <- cw[Time %in% c(0, 21),
  .(N      = .N,
    Mean   = mean(Weight),
    SDev   = sd(Weight),
    Median = median(Weight),
    Min    = min(Weight),
    Max    = max(Weight)  ),
  by=.(Group, Time)][
    order(Group, Time)]
cws
```

```
##     Group Time  N Mean  SDev Median Min Max
## 1: Diet 1    0 20   41  0.99     41  39  43
## 2: Diet 1   21 16  178 58.70    166  96 305
## 3: Diet 2    0 10   41  1.49     40  39  43
## 4: Diet 2   21 10  215 78.14    212  74 331
## 5: Diet 3    0 10   41  1.03     41  39  42
## 6: Diet 3   21 10  270 71.62    281 147 373
## 7: Diet 4    0 10   41  1.05     41  39  42
## 8: Diet 4   21  9  239 43.35    237 196 322
```

Prettier Table

Summary Statistics for the Chick Weight Data

Group	Time	N	Mean	SDev	Median	Min	Max
Diet 1	0	20	41	1.0	41	39	43
	21	16	178	58.7	166	96	305
Diet 2	0	10	41	1.5	40	39	43
	21	10	215	78.1	212	74	331
Diet 3	0	10	41	1.0	41	39	42
	21	10	270	71.6	281	147	373
Diet 4	0	10	41	1.1	41	39	42
	21	9	239	43.3	237	196	322

R Markdown Demo

data.table Practical

World Population Data

Download: https://ilustat.com/shared/GSiRdt.zip

Double Click on "World-Popn-dt.Rproj"

Summary

Assigning (`<-`) values to objects (e.g. vectors, data.table)

Manipulating Vectors

`data.table`: `dt[i, j, by]`

`dt[i, ]` **to filter or order rows**

`dt[, j]` **to create, rename and select columns**

`dt[, j, by]` **to do grouping operations on columns** `j`

data.table - Additional Exercises

Women in Parliament

https://ilustat.com/shared/WiP-rdatatable.zip