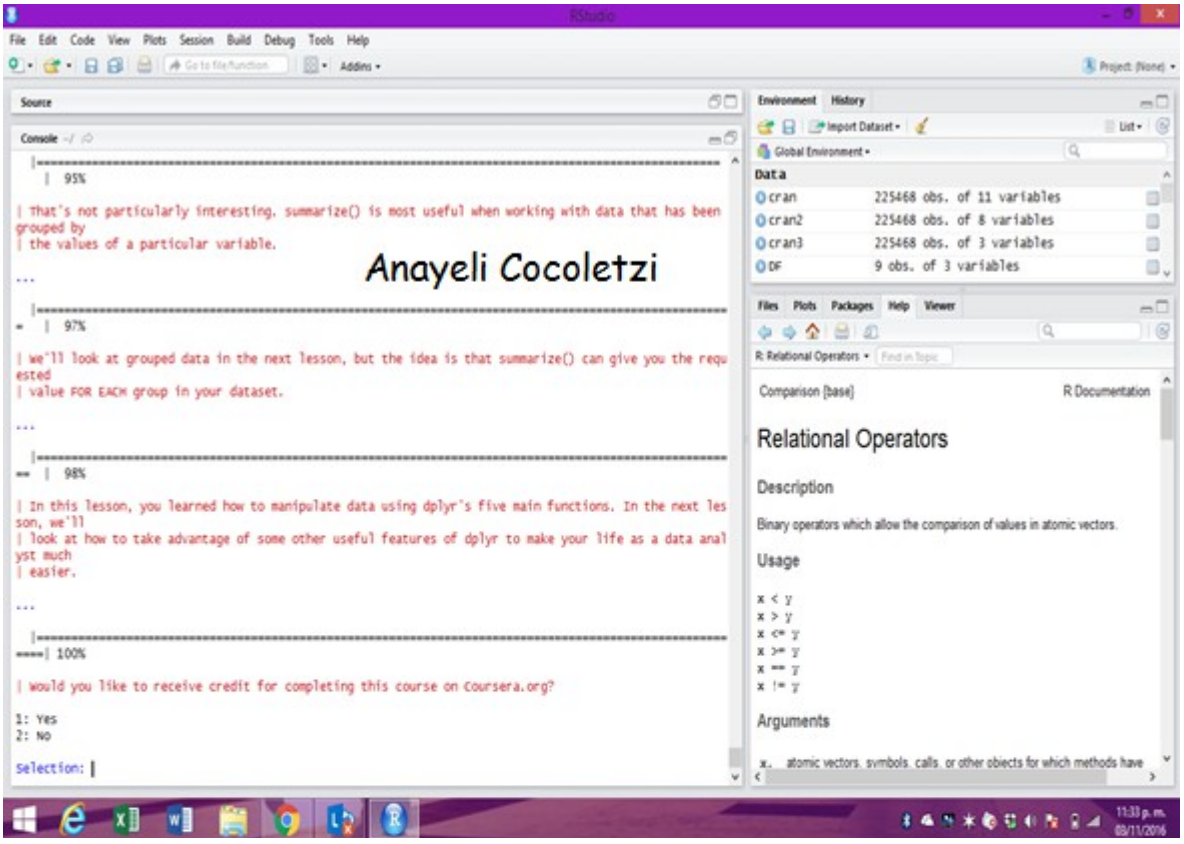
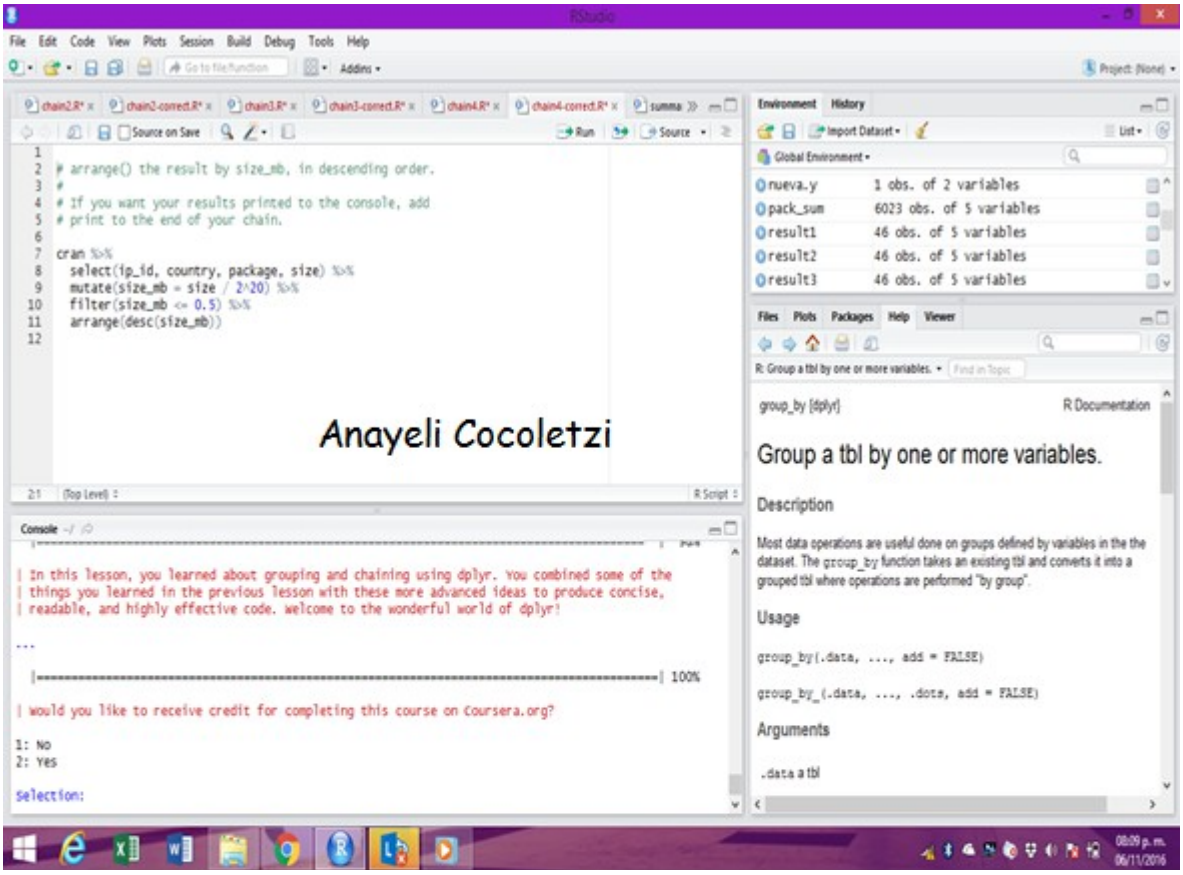
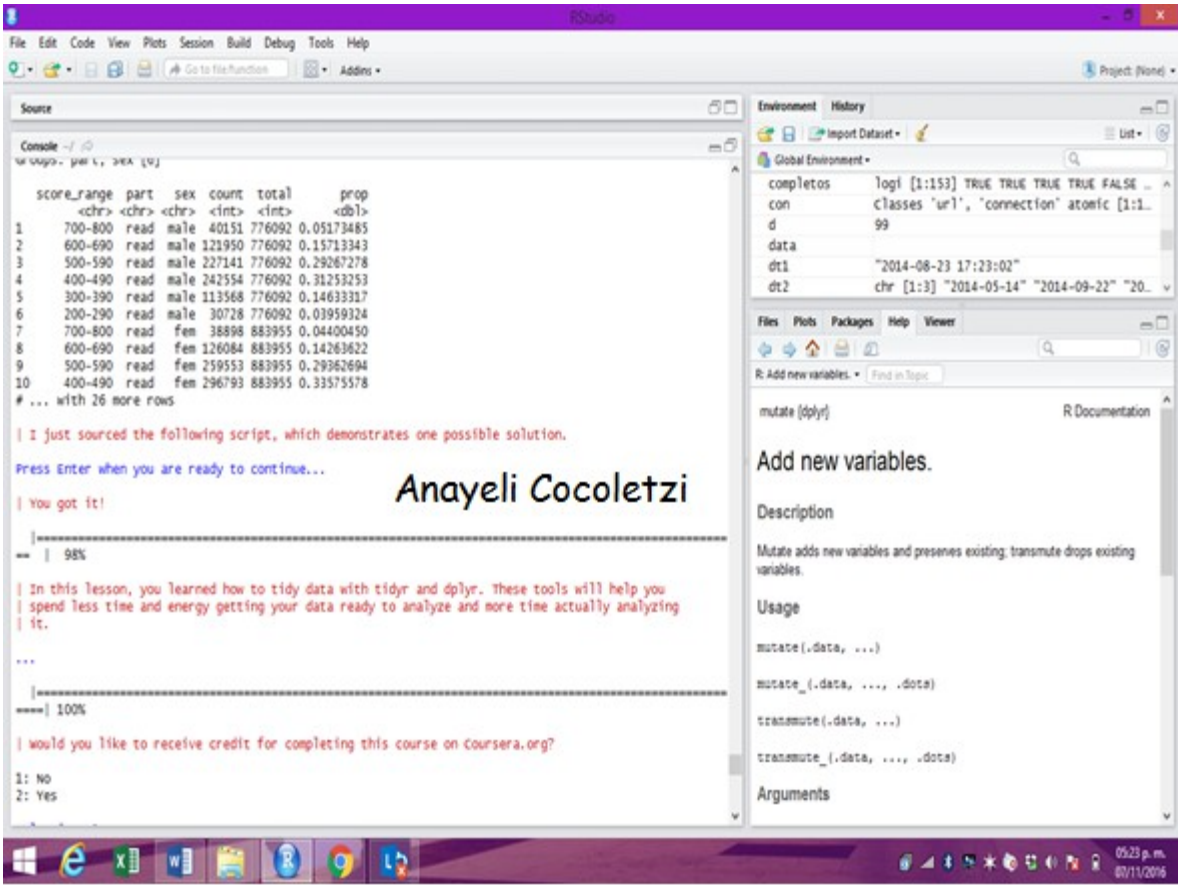
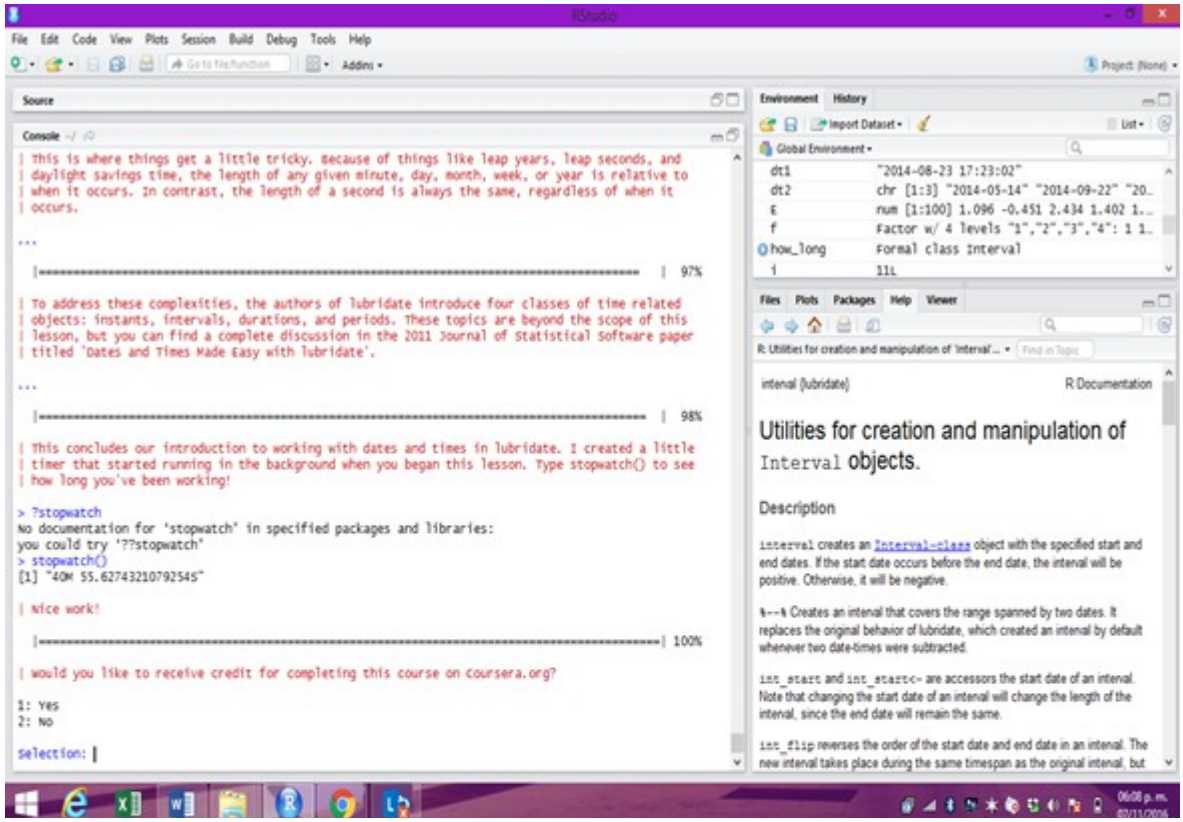

RStudio
File Edit Code View Plots Session Build Debug Tools Help
Source
Console
| That's not particularly interesting. summarize() is most useful when working with data that has been grouped by
| the values of a particular variable.
Anayeli Cocoletzi
| We'll look at grouped data in the next lesson, but the idea is that summarize() can give you the requested
| value FOR EACH group in your dataset.
| In this lesson, you learned how to manipulate data using dplyr's five main functions. In the next lesson, we'll
| look at how to take advantage of some other useful features of dplyr to make your life as a data analyst much
| easier.
| Would you like to receive credit for completing this course on Coursera.org?
1: Yes
2: No
Selection:
Environment History
Import Dataset
Global Environment
Data
cran 225468 obs. of 11 variables
cran2 225468 obs. of 8 variables
cran3 225468 obs. of 3 variables
DF 9 obs. of 3 variables
Files Plots Packages Help Viewer
R: Relational Operators
Comparison [base] R Documentation
Relational Operators
Description
Binary operators which allow the comparison of values in atomic vectors.
Usage
x < y
x > y
x <= y
x >= y
x == y
x != y
Arguments
x, atomic vectors, symbols, calls, or other objects for which methods have


RStudio
File Edit Code View Plots Session Build Debug Tools Help
chain2.R* chain2-correct.R* chain3.R* chain3-correct.R* chain4.R* chain4-correct.R* summa
Source on Save
Run
Source
arrange() the result by size_mb, in descending order.
#
If you want your results printed to the console, add
print to the end of your chain.
cran %>%
select(ip_id, country, package, size) %>%
mutate(size_mb = size / 2^20) %>%
filter(size_mb <= 0.5) %>%
arrange(desc(size_mb))
Anayeli Cocoletzi
2:1 (Top Level)
R Script
Console
| In this lesson, you learned about grouping and chaining using dplyr. You combined some of the
| things you learned in the previous lesson with these more advanced ideas to produce concise,
| readable, and highly effective code. Welcome to the wonderful world of dplyr!
...
| 100%
| Would you like to receive credit for completing this course on Coursera.org?
1: No
2: Yes
Selection:
Environment History
Import Dataset
List
Global Environment
nueva_y 1 obs. of 2 variables
pack_sum 6023 obs. of 5 variables
result1 46 obs. of 5 variables
result2 46 obs. of 5 variables
result3 46 obs. of 5 variables
Files Plots Packages Help Viewer
R: Group a tbl by one or more variables.
group_by {dplyr} R Documentation
Group a tbl by one or more variables.
Description
Most data operations are useful done on groups defined by variables in the the dataset. The group_by function takes an existing tbl and converts it into a grouped tbl where operations are performed "by group".
Usage
group_by(.data, ..., add = FALSE)
group_by_(.data, ..., .dots, add = FALSE)
Arguments
.data a tbl
08:09 p. m.
06/11/2016


File Edit Code View Plots Session Build Debug Tools Help
Addins
Project: (None)
Source
Console
Environment History
Import Dataset
List
Global Environment
completos logi [1:153] TRUE TRUE TRUE TRUE FALSE ...
con classes 'url', 'connection' atomic [1:1...
d 99
data
dt1 "2014-08-23 17:23:02"
dt2 chr [1:3] "2014-05-14" "2014-09-22" "20...
Files Plots Packages Help Viewer
R: Add new variables.
mutate (dplyr) R Documentation
Add new variables.
Description
Mutate adds new variables and preserves existing; transmute drops existing variables.
Usage
mutate(.data, ...)
mutate_(.data, ..., .dots)
transmute(.data, ...)
transmute_(.data, ..., .dots)
Arguments
score_range part sex count total prop
<chr> <chr> <chr> <int> <int> <dbl>
1 700-800 read male 40151 776092 0.05173485
2 600-690 read male 121950 776092 0.15713343
3 500-590 read male 227141 776092 0.29267278
4 400-490 read male 242554 776092 0.31253253
5 300-390 read male 113568 776092 0.14633317
6 200-290 read male 30728 776092 0.03959324
7 700-800 read fem 38898 883955 0.04400450
8 600-690 read fem 126084 883955 0.14263622
9 500-590 read fem 259553 883955 0.29362694
10 400-490 read fem 296793 883955 0.33575578
... with 26 more rows
| I just sourced the following script, which demonstrates one possible solution.
Press Enter when you are ready to continue...
| You got it!
Anayeli Cocoletzi
98%
| In this lesson, you learned how to tidy data with tidyr and dplyr. These tools will help you
| spend less time and energy getting your data ready to analyze and more time actually analyzing
| it.
...
100%
| Would you like to receive credit for completing this course on Coursera.org?
1: No
2: Yes
05:23 p. m.
07/11/2016


File Edit Code View Plots Session Build Debug Tools Help
Source
Console
| This is where things get a little tricky. Because of things like leap years, leap seconds, and
| daylight savings time, the length of any given minute, day, month, week, or year is relative to
| when it occurs. In contrast, the length of a second is always the same, regardless of when it
| occurs.
...
|======== | 97%
| To address these complexities, the authors of lubridate introduce four classes of time related
| objects: instants, intervals, durations, and periods. These topics are beyond the scope of this
| lesson, but you can find a complete discussion in the 2011 Journal of Statistical Software paper
| titled 'Dates and Times Made Easy with lubridate'.
...
|======== | 98%
| This concludes our introduction to working with dates and times in lubridate. I created a little
| timer that started running in the background when you began this lesson. Type stopwatch() to see
| how long you've been working!
> ?stopwatch
No documentation for 'stopwatch' in specified packages and libraries:
you could try '??stopwatch'
> stopwatch()
[1] "40M 55.6274321079254S"
| Nice work!
|======== | 100%
| Would you like to receive credit for completing this course on Coursera.org?
1: Yes
2: No
Selection:
Environment History
Import Dataset
Global Environment
dt1 "2014-08-23 17:23:02"
dt2 chr [1:3] "2014-05-14" "2014-09-22" "20..
E num [1:100] 1.096 -0.451 2.434 1.402 1..
f Factor w/ 4 levels "1","2","3","4": 1 1..
how_long Formal class Interval
i 11L
Files Plots Packages Help Viewer
R: Utilities for creation and manipulation of 'interval'...
interval {lubridate} R Documentation
Utilities for creation and manipulation of Interval objects.
Description
interval creates an Interval-class object with the specified start and end dates. If the start date occurs before the end date, the interval will be positive. Otherwise, it will be negative.
%--% Creates an interval that covers the range spanned by two dates. It replaces the original behavior of lubridate, which created an interval by default whenever two date-times were subtracted.
int_start and int_start<- are accessors the start date of an interval. Note that changing the start date of an interval will change the length of the interval, since the end date will remain the same.
int_flip reverses the order of the start date and end date in an interval. The new interval takes place during the same timespan as the original interval, but