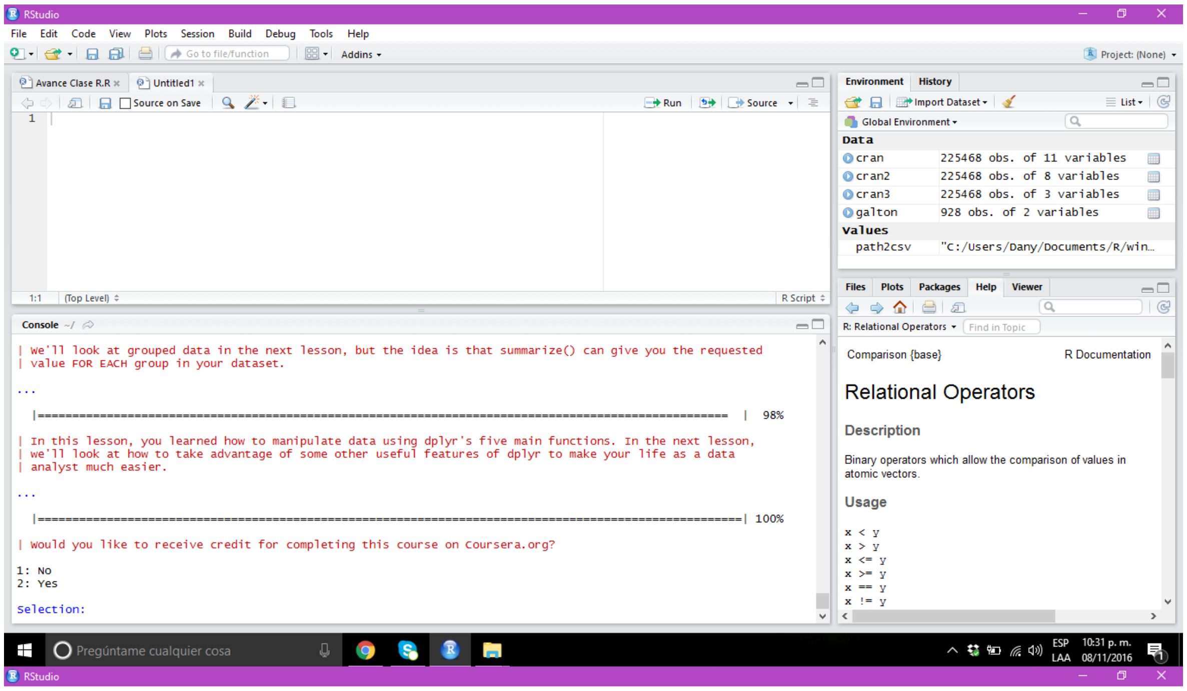
Getting and Cleaning Data

Daniel Reynaga Gil 201504086



```
1
```

Data	
cran	225468 obs. of 11 variables
cran2	225468 obs. of 8 variables
cran3	225468 obs. of 3 variables
galton	928 obs. of 2 variables
Values	
path2csv	"C:/Users/Dany/Documents/R/win…

```
| 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.

...

  |================================================================================================   |  98%

| 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.

...

  |===================================================================================================| 100%

| Would you like to receive credit for completing this course on Coursera.org?

1: No
2: Yes

Selection:
```

Comparison {base} R Documentation

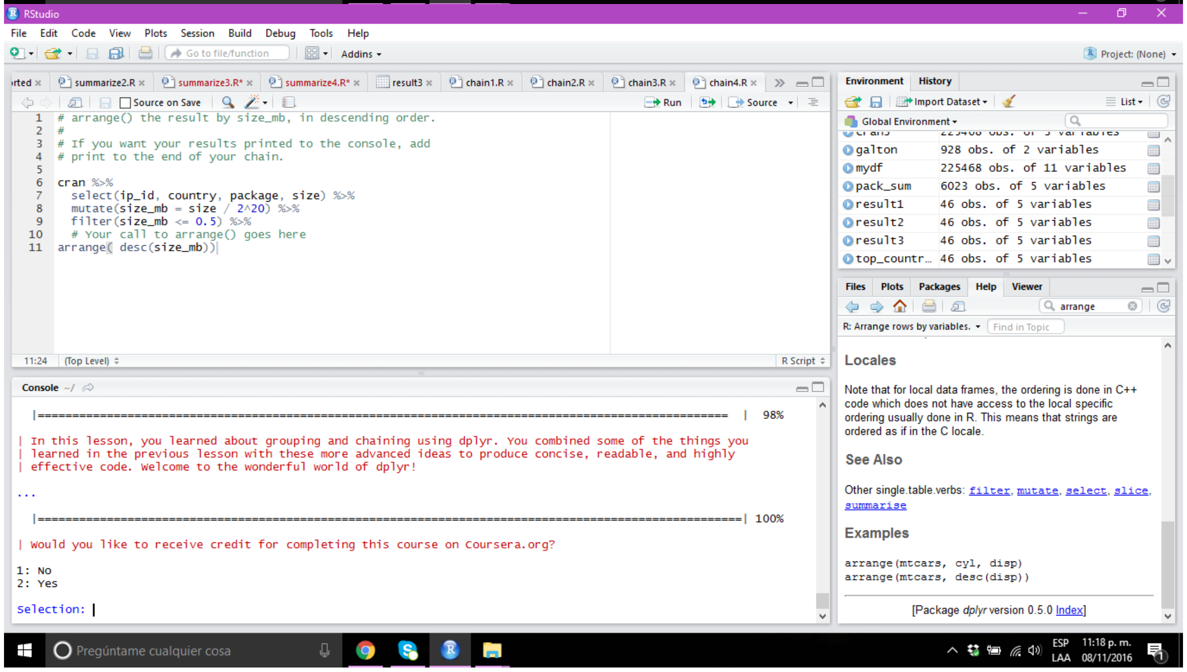
Relational Operators

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
```



```
# 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) %>%
  # Your call to arrange() goes here
arrange( desc(size_mb))
```

galton	928 obs. of 2 variables
mydf	225468 obs. of 11 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
top_countr…	46 obs. of 5 variables

```
  |================================================================================================   |  98%

| 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: |
```

Locales

Note that for local data frames, the ordering is done in C++ code which does not have access to the local specific ordering usually done in R. This means that strings are ordered as if in the C locale.

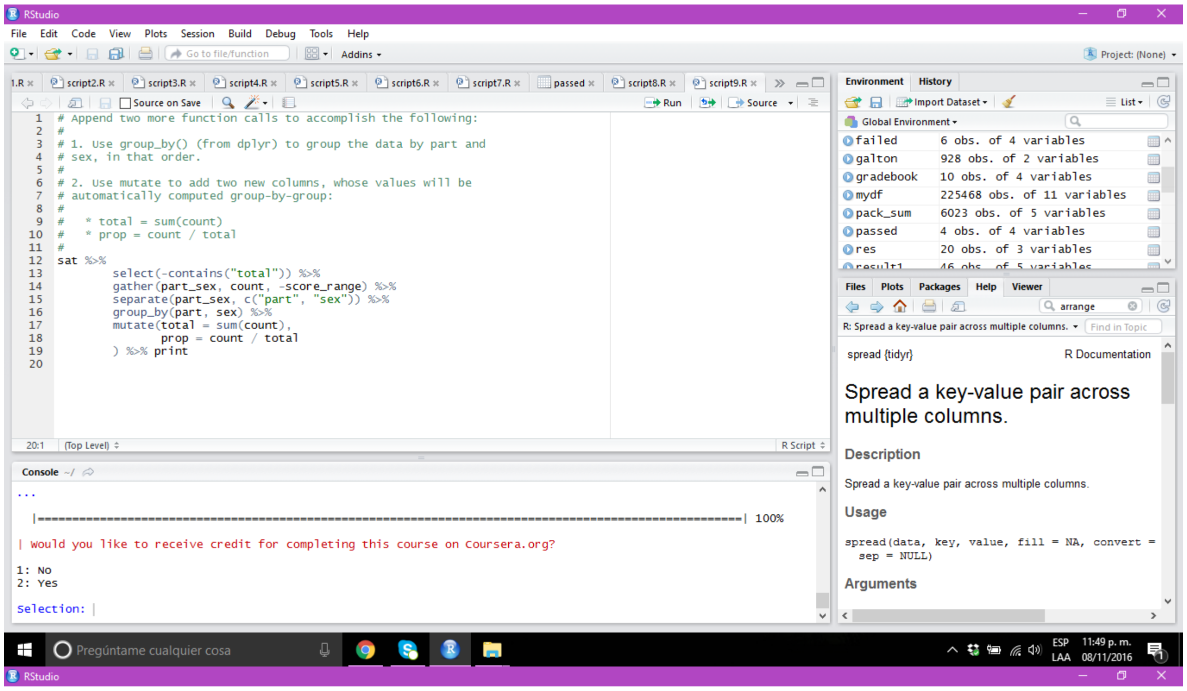
See Also

Other single.table.verbs: filter, mutate, select, slice, summarise

Examples

```
arrange(mtcars, cyl, disp)
arrange(mtcars, desc(disp))
```

[Package *dplyr* version 0.5.0 Index]



```r
# Append two more function calls to accomplish the following:
#
# 1. Use group_by() (from dplyr) to group the data by part and
# sex, in that order.
#
# 2. Use mutate to add two new columns, whose values will be
# automatically computed group-by-group:
#
#   * total = sum(count)
#   * prop = count / total
#
sat %>%
        select(-contains("total")) %>%
        gather(part_sex, count, -score_range) %>%
        separate(part_sex, c("part", "sex")) %>%
        group_by(part, sex) %>%
        mutate(total = sum(count),
               prop = count / total
        ) %>% print

```

```
...

  |=============================================================================================| 100%

| Would you like to receive credit for completing this course on Coursera.org?

1: No
2: Yes

Selection: 
```

```
failed      6 obs. of 4 variables
galton      928 obs. of 2 variables
gradebook   10 obs. of 4 variables
mydf        225468 obs. of 11 variables
pack_sum    6023 obs. of 5 variables
passed      4 obs. of 4 variables
res         20 obs. of 3 variables
result1     46 obs. of 5 variables
```

R: Spread a key-value pair across multiple columns.

spread {tidyr} R Documentation

Spread a key-value pair across multiple columns.

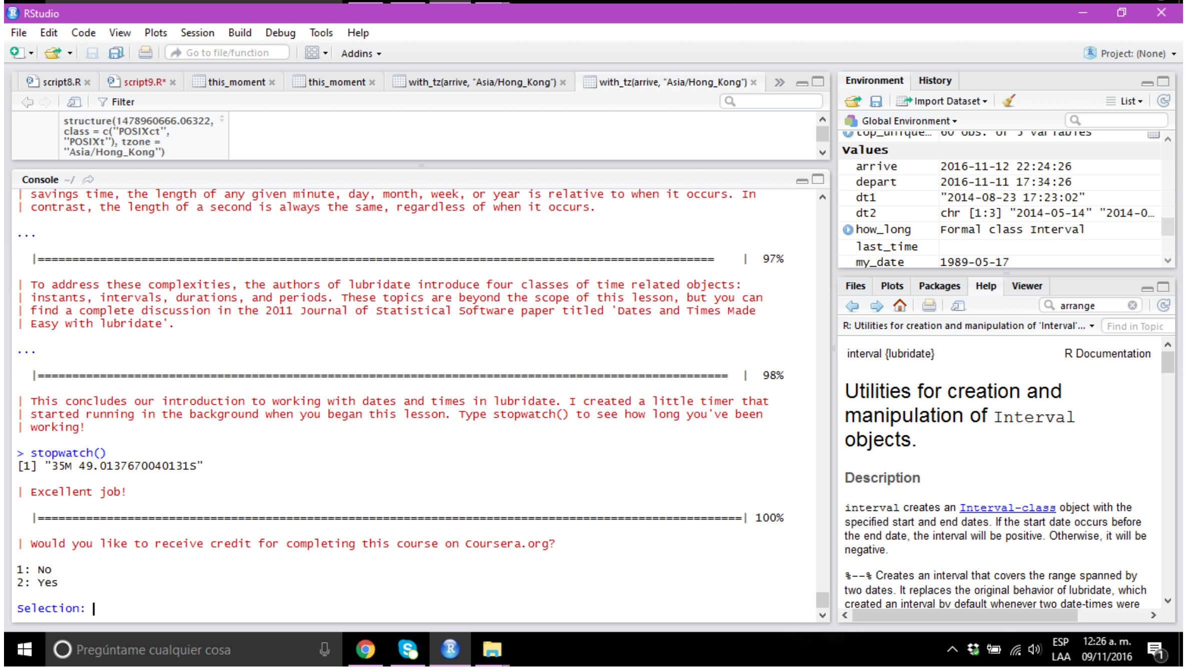
Description

Spread a key-value pair across multiple columns.

Usage

```r
spread(data, key, value, fill = NA, convert =
  sep = NULL)
```

Arguments



```
structure(1478960666.06322,
class = c("POSIXct",
"POSIXt"), tzone =
"Asia/Hong_Kong")
```

```
| 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()
[1] "35M 49.0137670040131S"

| Excellent job!

  |=============================================================================================| 100%

| Would you like to receive credit for completing this course on Coursera.org?

1: No
2: Yes

Selection: |
```

```
top_unique…  60 obs. of 3 variables
Values
arrive      2016-11-12 22:24:26
depart      2016-11-11 17:34:26
dt1         "2014-08-23 17:23:02"
dt2         chr [1:3] "2014-05-14" "2014-0…
how_long    Formal class Interval
last_time
my_date     1989-05-17
```

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

RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Addins
Project: (None)
script8.R script9.R* this_moment this_moment with_tz(arrive, "Asia/Hong_Kong") with_tz(arrive, "Asia/Hong_Kong")
Filter
Console ~/
Environment History
Import Dataset
List
Global Environment
Files Plots Packages Help Viewer
arrange
Find in Topic
Pregúntame cualquier cosa
ESP LAA
12:26 a. m.
09/11/2016