Handling Date & Time in R

Rsquared Academy

www.rsquaredacademy.com

Agenda

- → get current date/time
- → understand date/time classes in R
- → date arithmetic
- → time zones and daylight savings
- → date/time formats
- → parsing date/time
- → date/time components
- → create/update/verify
- → Intervals, duration and period

Prerequisites

→ basic knowledge of R & RStudio

→ laptop/desktop

→ working internet connection

Let's Begin to Learn

- **L**earning **M**anagement **S**ystem
  - Videos
  - Slides
  - Code
  - Data
  - Suggested Readings

- Need help?
  - Discussion forum
  - support@rsquaredacademy.com
  - +91-9606872504
  - Raise a ticket

Resources

→ Slides

→ Code & Data

→ RStudio Cloud

→ Online Course

→ Blog Post

Best Way to Learn

- → Go through the videos
- → Run the code (locally or on RStudio Cloud)
- → Experiment with the code
- → Go through the suggested readings
- → If you get stuck with the code or have any questions
    - → Use the discussion forum
    - → Mail us @ support@rsquaredacademy.com if you have any questions (**We believe no question is naive and every question is important.**) and we will try to reply at the earliest.

Module 1

Introduction

Current Date/Time

→ `Sys.Date() / today()`

→ `Sys.time() / now()`

→ `am()`

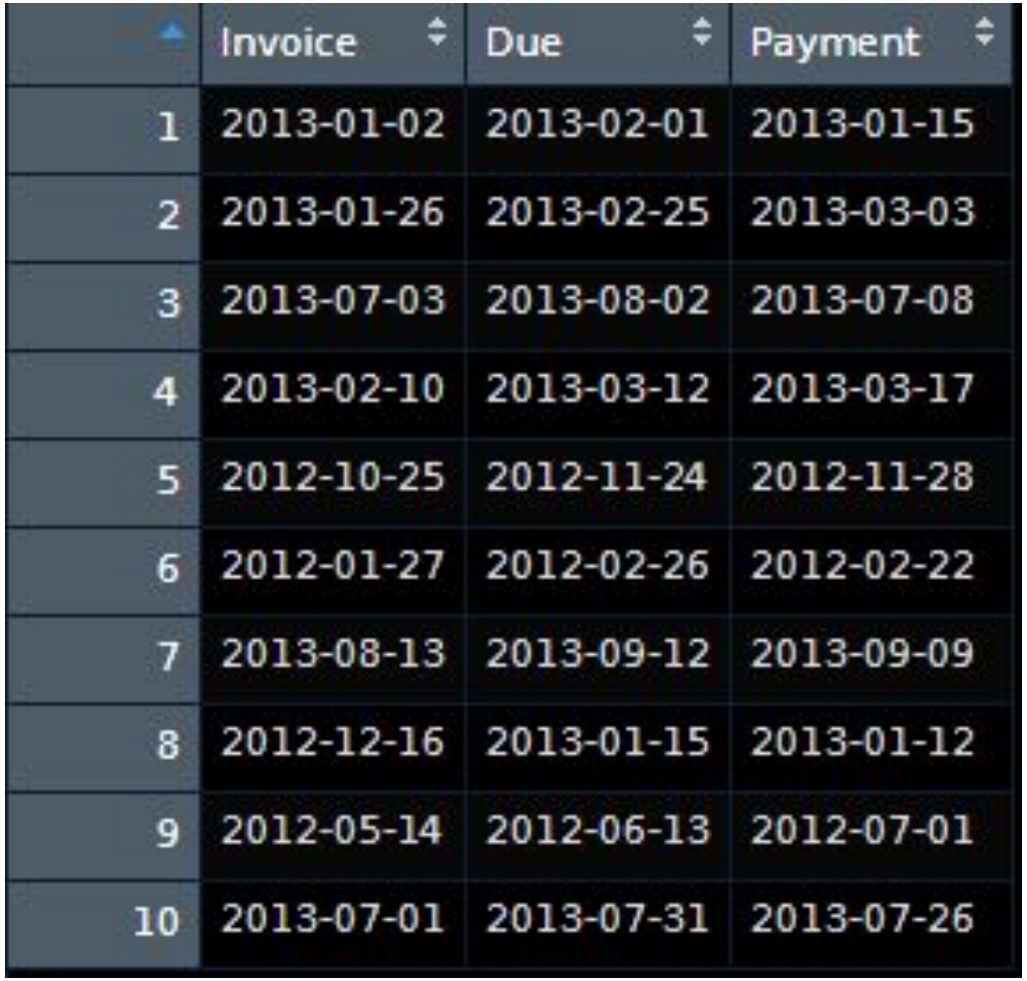
→ `pm()`

→ `leap_year()`

Your Turn...

- → get current date
- → get current time
- → check whether the time is in AM or PM?
- → check whether the following are leap years
  - → 2018
  - → 2016

Case Study - Data



	Invoice	Due	Payment
1	2013-01-02	2013-02-01	2013-01-15
2	2013-01-26	2013-02-25	2013-03-03
3	2013-07-03	2013-08-02	2013-07-08
4	2013-02-10	2013-03-12	2013-03-17
5	2012-10-25	2012-11-24	2012-11-28
6	2012-01-27	2012-02-26	2012-02-22
7	2013-08-13	2013-09-12	2013-09-09
8	2012-12-16	2013-01-15	2013-01-12
9	2012-05-14	2012-06-13	2012-07-01
10	2013-07-01	2013-07-31	2013-07-26

Case Study

→ extract date, month and year from Due

→ compute the number of days to settle invoice

→ compute days over due

→ check if due year is a leap year

→ check when due day in february is 29, whether it is a leap year

→ how many invoices were settled within due date

→ how many invoices are due in each quarter

→ what is the average duration between invoice date and payment date

Module 2

Date & Time Classes

Rsquared Academy

www.rsquaredacademy.com

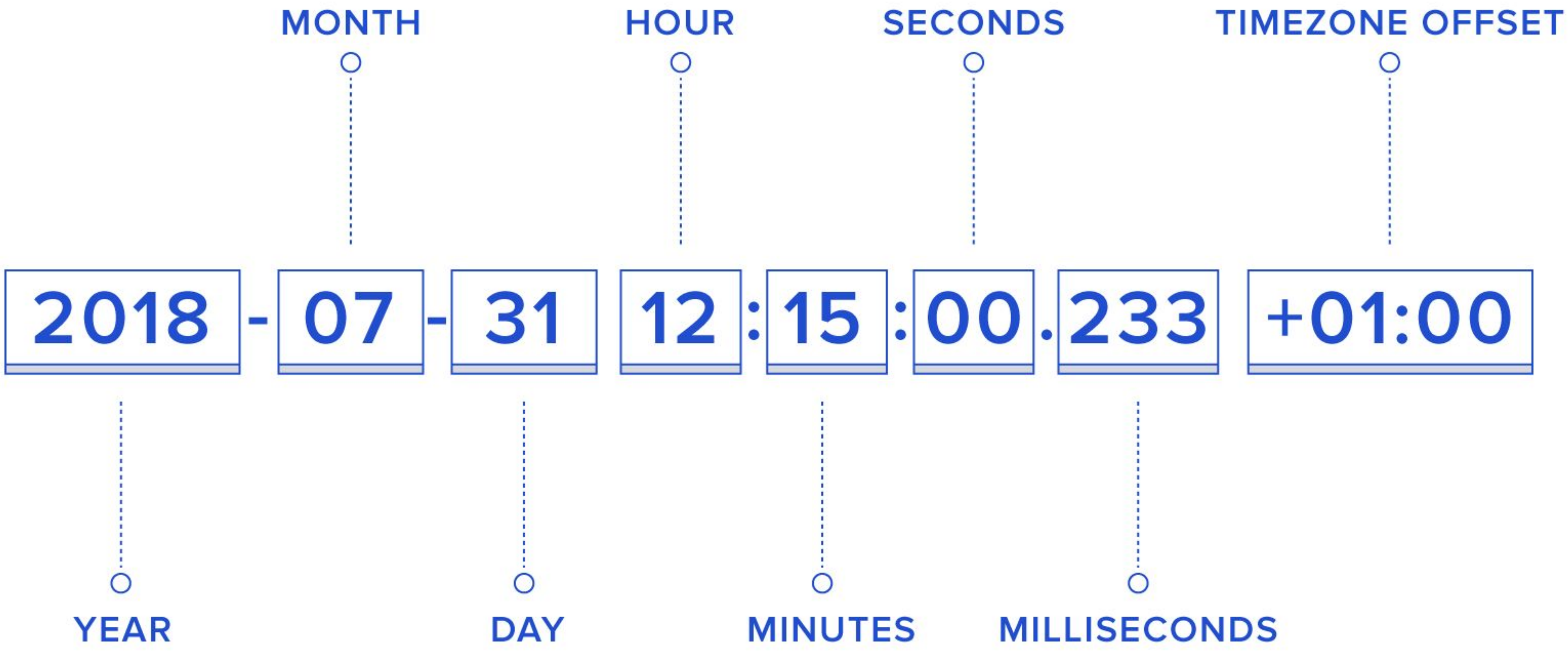
Date & Time Classes

→ Date

→ POSIXct

→ POSIXlt

ISO 8601 Standard



POSIXlt

→ sec

→ min

→ hour

→ mon

→ zone

→ wday

→ mday

→ year

→ yday

→ ist

→ gmtoff

Your Turn...

R 1.0.0 was released on `2000-02-09 08:55:23 UTC`. Save it as

→ Date

→ Date using origin

→ POSIXct

→ POSIXlt and extract

  → month day

  → day of year

  → month

  → zone

→ ISODate

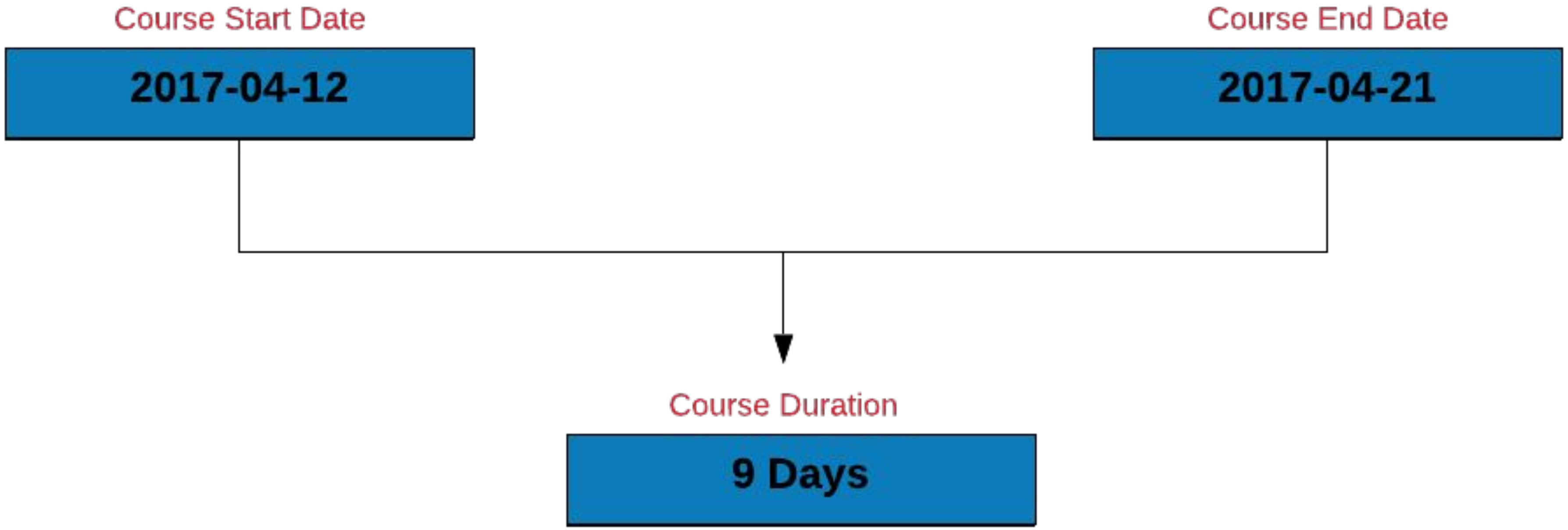
Module 3

Date Arithmetic

Rsquared Academy

www.rsquaredacademy.com

Course Duration



Course Start Date

2017-04-12

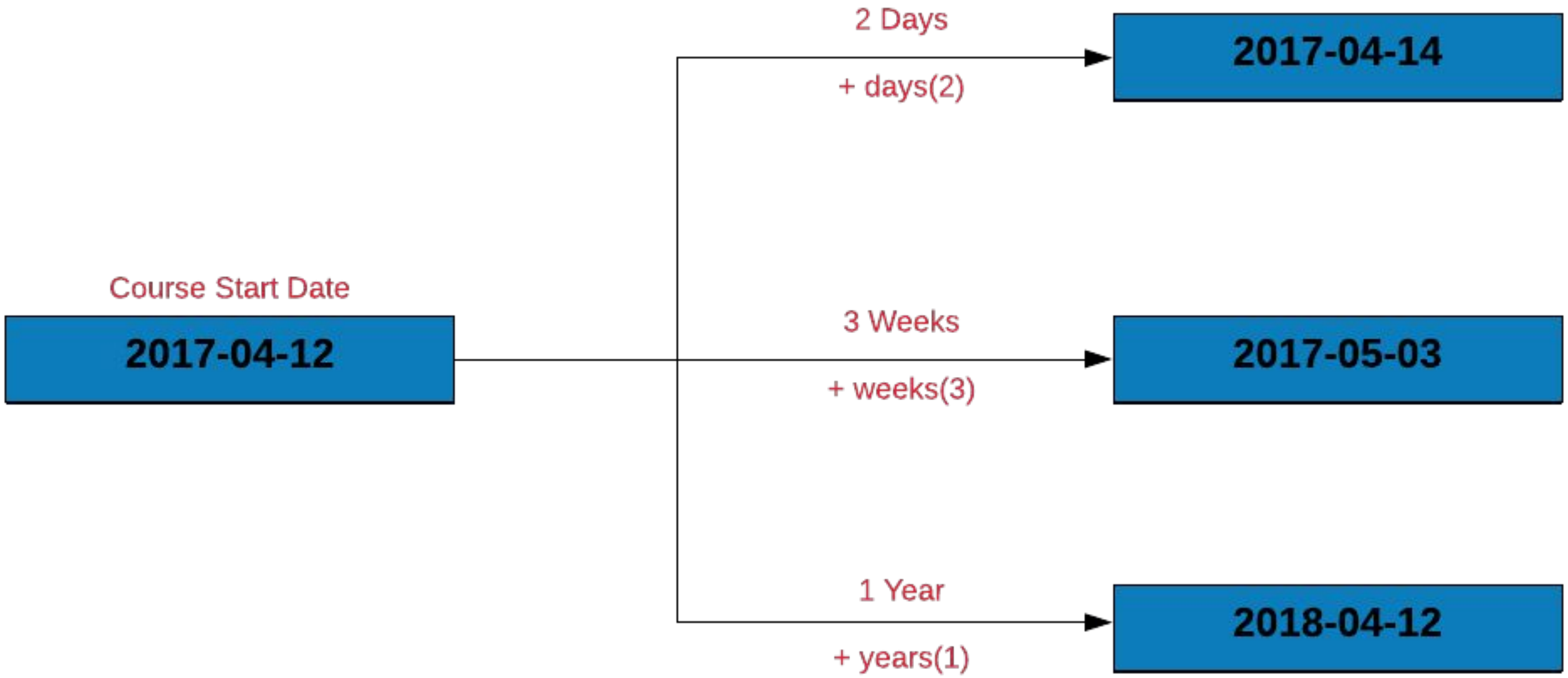
Course End Date

2017-04-21

Course Duration

9 Days

Shift Date



Course Start Date

2017-04-12

2 Days

+ days(2)

2017-04-14

3 Weeks

+ weeks(3)

2017-05-03

1 Year

+ years(1)

2018-04-12

Your Turn...

- → compute the length of a vacation which begins on `2020-04-19` and ends on `2020-04-25`
- → recompute the length of the vacation after shifting the vacation start and end date by **10 days** and **2 weeks**
- → compute the days to settle invoice and days overdue from the **receivables.csv** data set
- → compute the length of employment (only for those employees who have been terminated) from the **hr-data.csv** data set (use date of hire and termination)

Module 4

Time Zones & Daylight Savings

Rsquared Academy

www.rsquaredacademy.com

Time Zones

Daylight Savings

FALL
SPRING

Your Turn...

→ check the time zone you live in

→ check if daylight savings is in effect

→ check the current time in **UTC** or any other time zone

Module 5

Date & Time Formats

Rsquared Academy

www.rsquaredacademy.com

Formats

→ December 12, 2019

→ 12th Dec, 2019

→ Dec 12th, 19

→ 12-Dec-19

→ 12 December, 2019

→ 12.12.19

Conversion Specifications

Specification	Description	Example
%d	Day of the month (decimal number)	12
%m	Month (decimal number)	12
%b	Month (abbreviated)	Dec
%B	Month (full name)	December
%y	Year (2 digit)	19
%Y	Year (4 digit)	2019
%H	Hour	8
%M	Minute	5
%S	Second	3

Your Turn...

Use conversion specifications to specify the below dates using `as.Date()`

→ 05.07.19

→ 5-July, 2019

→ July 5th, 2019

→ July 05, 2019

→ 2019-July-05

→ 05/07/2019

→ 07/05/2019

→ 7/5/2019

→ 07/5/19

→ 2019-07-05

Module 6

Parse Date & Time

Rsquared Academy

www.rsquaredacademy.com

Parse Date & Time

→ Convert string/character to date using

  → `strptime()`

  → `parse_date_time()`

→ Convert date to string character using

  → `strftime()`

  → `as.character()`

  → `format()`

→ Use helper functions to convert string/character to date without using conversion specifications.

Your Turn...

Parse the below dates using `strptime()/parse_date_time()` or appropriate helper functions:

→ 05.07.19

→ 5-July, 2019

→ July 5th, 2019

→ July 05, 2019

→ 2019-July-05

→ 05/07/2019

→ 07/05/2019

→ 7/5/2019

→ 07/5/19

→ 2019-07-05

Module 7

Date & Time Components

Rsquared Academy

www.rsquaredacademy.com

Components

→ year

→ month

→ date

→ week

→ day

→ quarter

→ semester

→ hour

→ minute

→ second

→ timezone

Year, Month & Week

Function	Description
`year()`	Get year
`month()`	Get month (number)
`month(label = TRUE)`	Get month (abbreviated name)
`month(abbr = FALSE)`	Get month (full name)
`months()`	Get month
`week()`	Get week

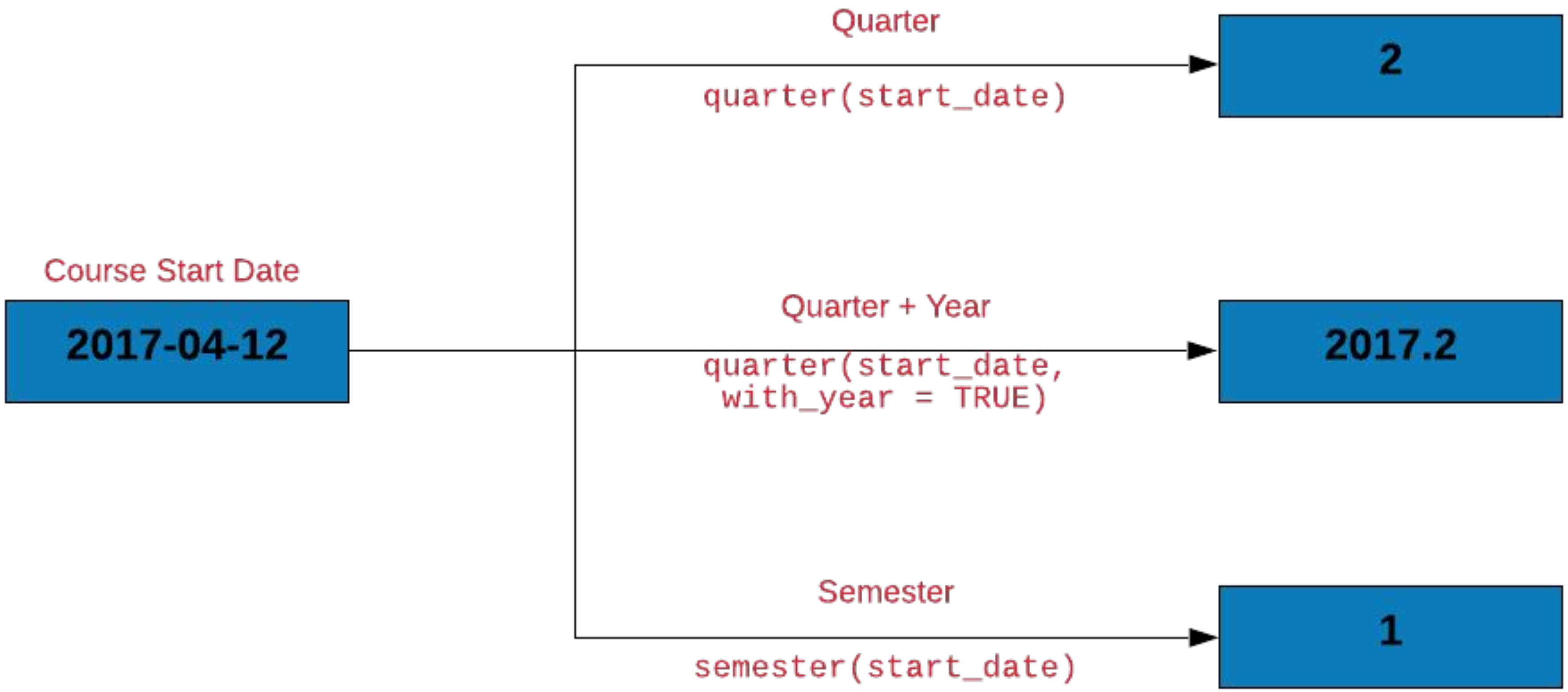
Day

Function	**Description**
`day()`	Get day
`mday()`	Day of the month
`wday()`	Day of the week
`qday()`	Day of quarter
`yday()`	Day of year
`weekdays()`	Day of week
`days_in_month()`	Days in the month

HMS

Function	Description
`hour()`	Get hour
`minute()`	Get minute
`second()`	Get second
`seconds()`	Number of seconds since `1970-01-01`

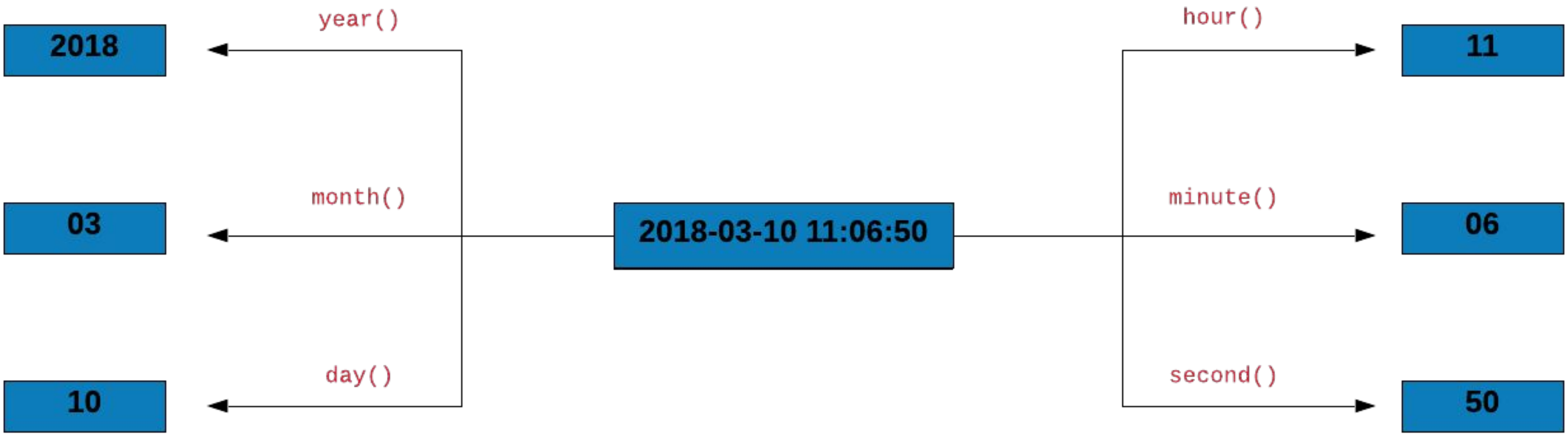
Quarter & Semester



Quarter & Semester

Function	Description
`quarter()`	Get quarter
`quarter(with_year = TRUE)`	Quarter with year
`quarter(fiscal_start = 4)`	Fiscal starts in April
`quarters()`	Get quarter
`semester()`	Get semester

Case Study - Components



Case Study - Data Sanitization

Let us do some data sanitization. If the due day happens to be February 29, let us ensure that the due year is a leap year. Below are the steps to check if the due year is a leap year:

- we will extract the following from the due date:
  - day
  - month
  - year
- we will then create a new column is_leap which will have be set to TRUE if the year is a leap year else it will be set to FALSE
- filter all the payments due on 29th Feb
- select the following columns:
  - Due
  - is_leap

Your Turn...

Get the R release dates using `r_versions()` from the **rversions** package and tabulate the following

→ year
→ month with label
→ weekday with label
→ hour
→ and quarter

Module 8

Create, Update & Verify

Create, Update & Verify

- → `make_date()`
- → `make_datetime()`
- → `update()`

Date Sequence

Function	**Description**
`from`	Starting date of the sequence
`by`	End date of the sequence
`to`	Date increment of the sequence
`length.out`	Length of the sequence
`along.with`	Use length of this value as length of sequence

Your Turn…

R 2.0.0 was released on `2004-10-04 14:24:38`. Create this date using

- `make_date()`
- `make_datetime()`
- and update to `2013-04-03 07:12:36` (**R 3.0.0 release date**)

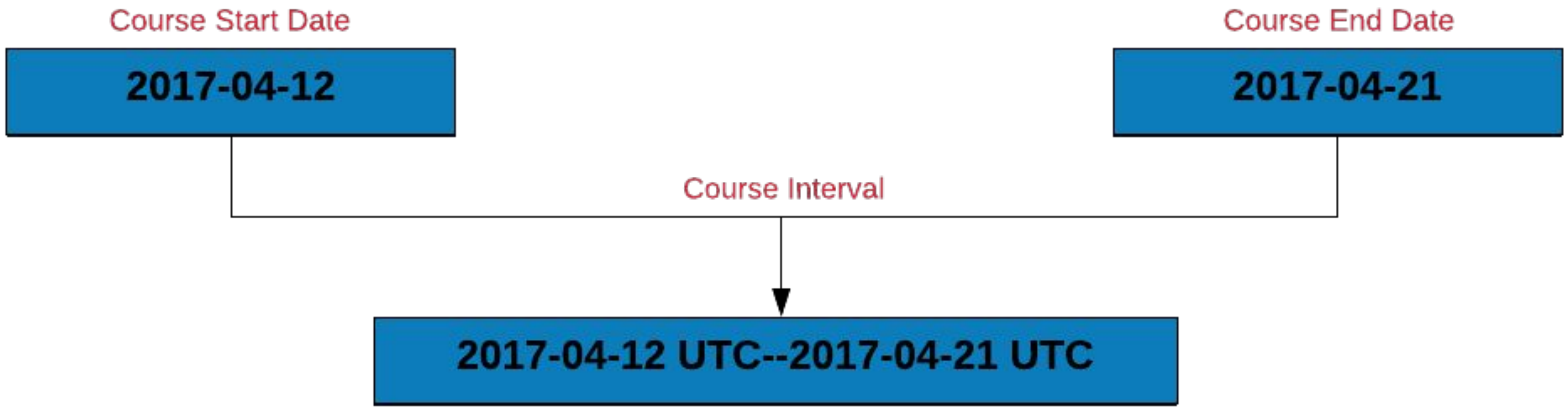
Module 9

Intervals, Duration & Period

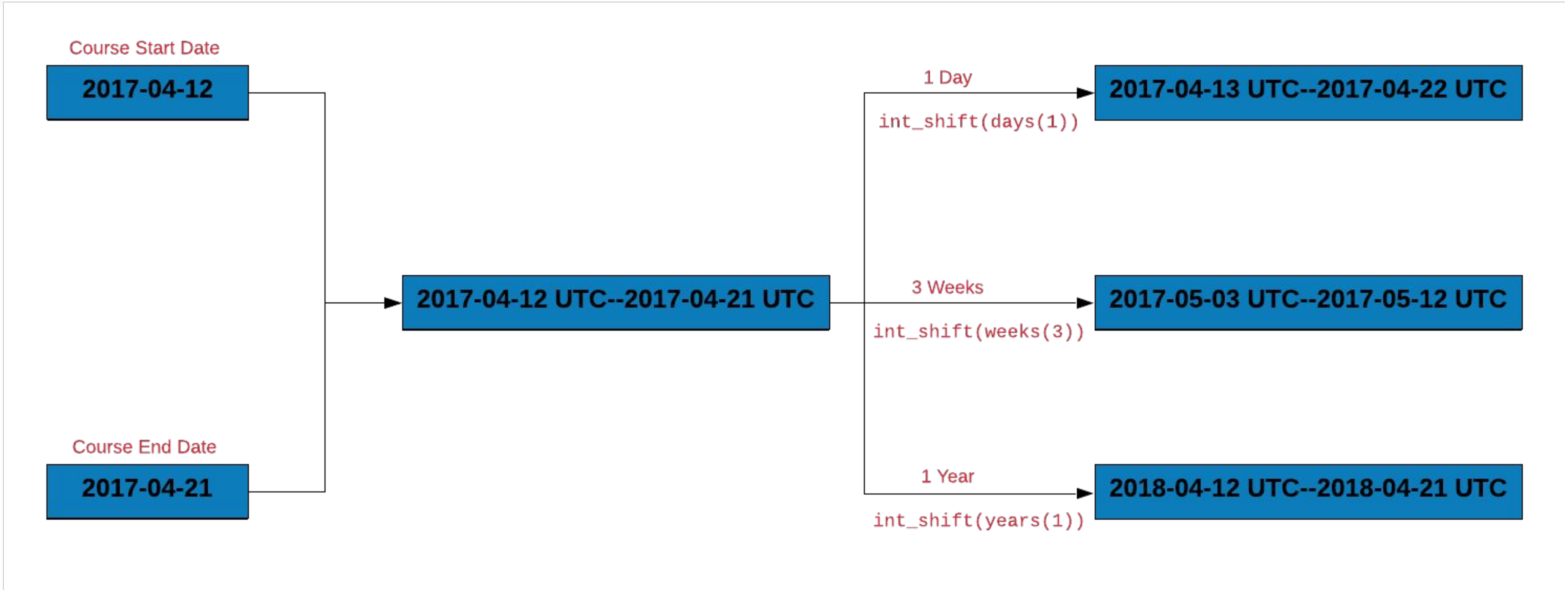
Rsquared Academy

www.rsquaredacademy.com

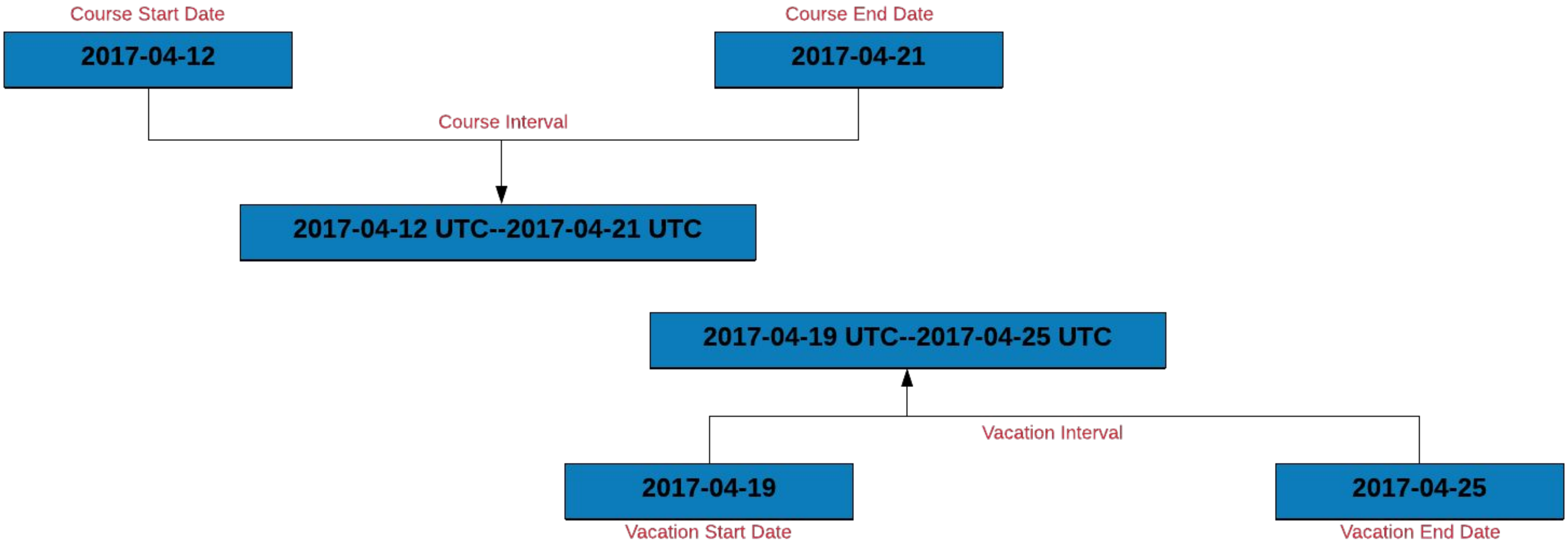
Interval


Course Start Date
2017-04-12
Course End Date
2017-04-21
Course Interval
2017-04-12 UTC--2017-04-21 UTC

Shift Interval



Interval Overlap

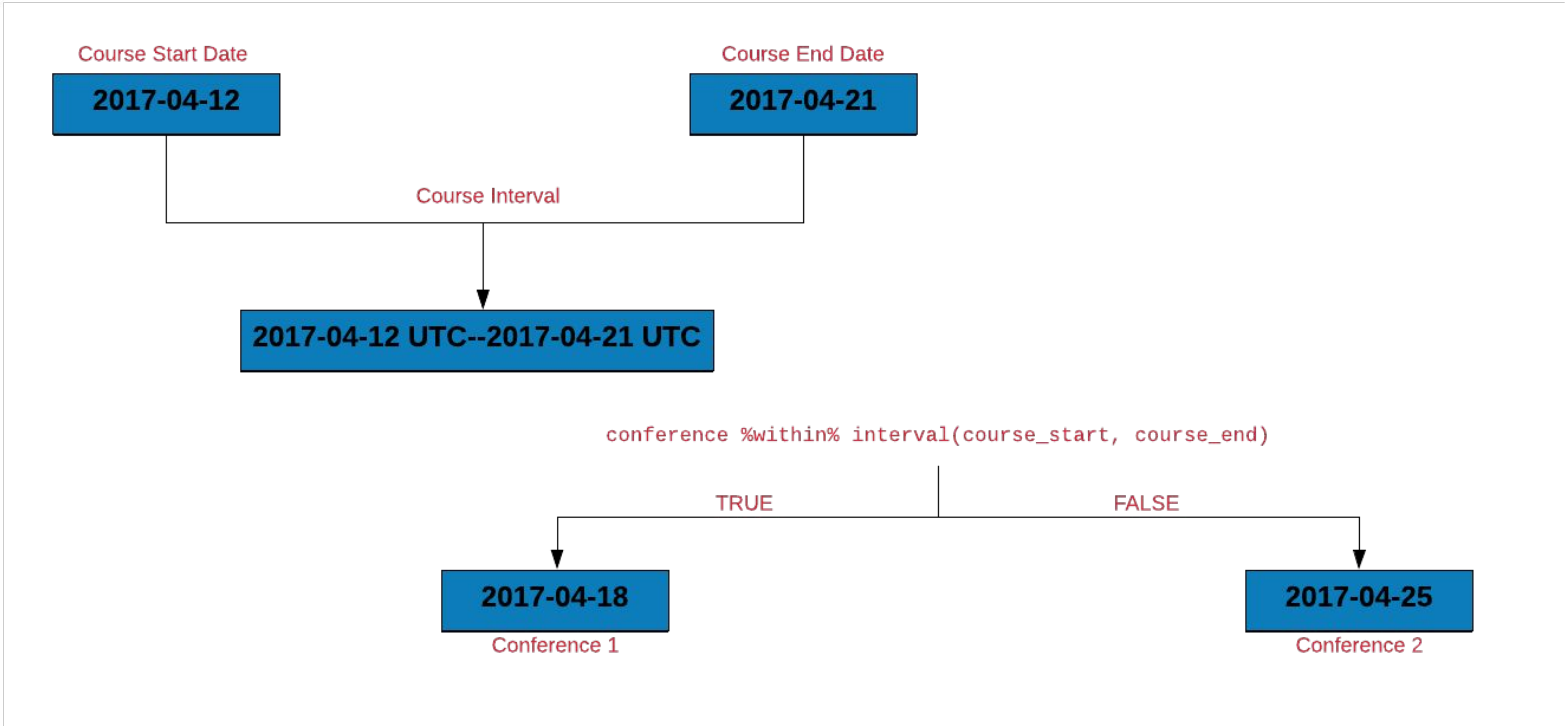


Case Study

Let us use intervals to count the number of invoices that were settled within the due date. To do this, we will:

- create an interval for the invoice and due date
- create a new column `due_next` by incrementing the due date by 1 day
- another interval for `due_next` and the payment date
- if the intervals overlap, the payment was made within the due date

Within



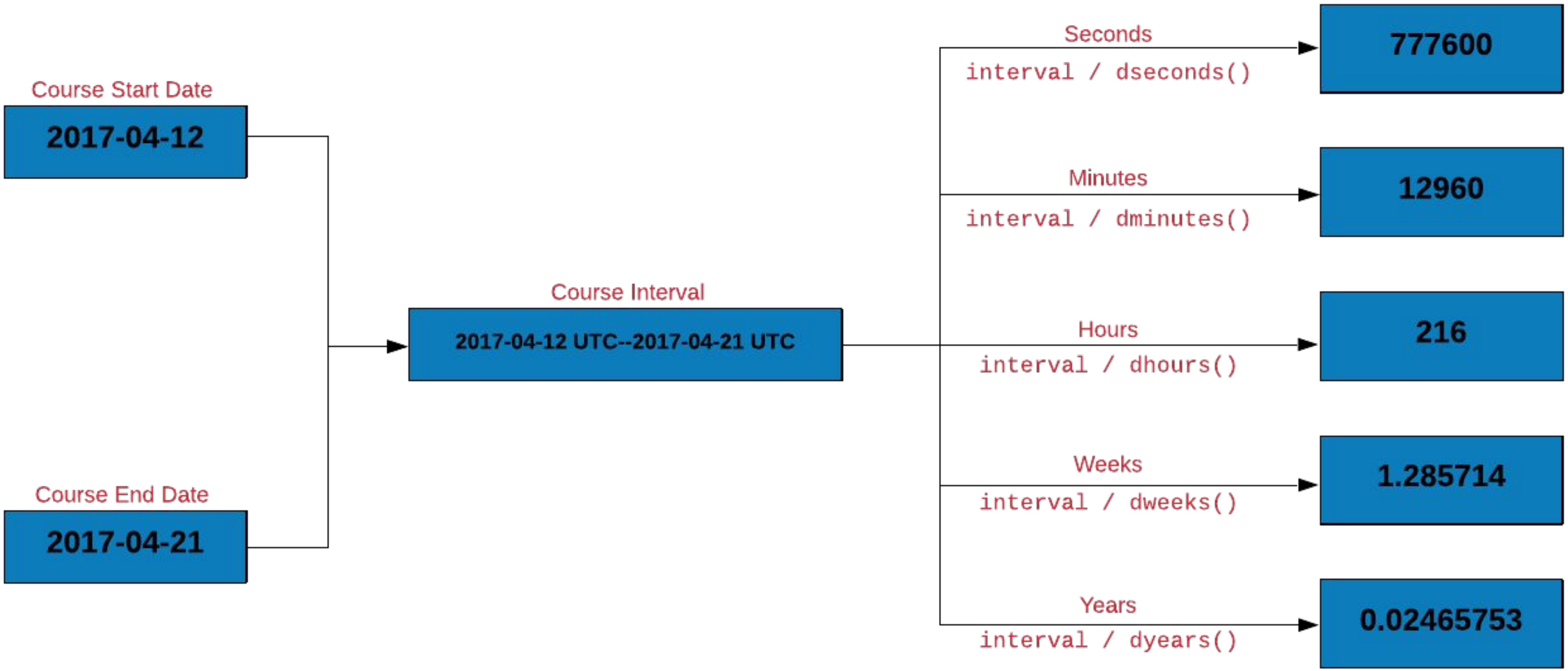
```
conference %within% interval(course_start, course_end)
```

Within

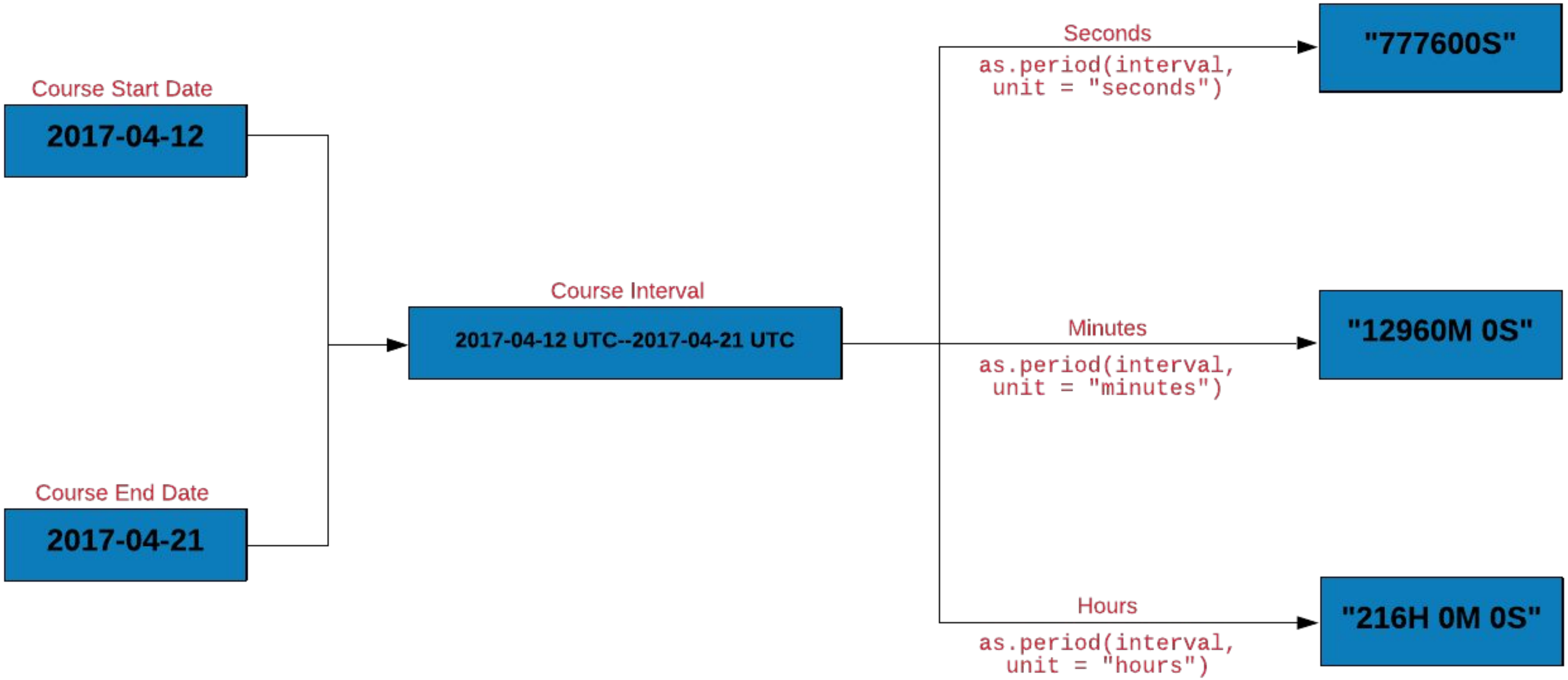
Let us use **`%within%`** to count the number of invoices that were settled within the due date. We will do this by:

- ➢ creating an interval for the invoice and due date
- ➢ check if the payment date falls within the above interval

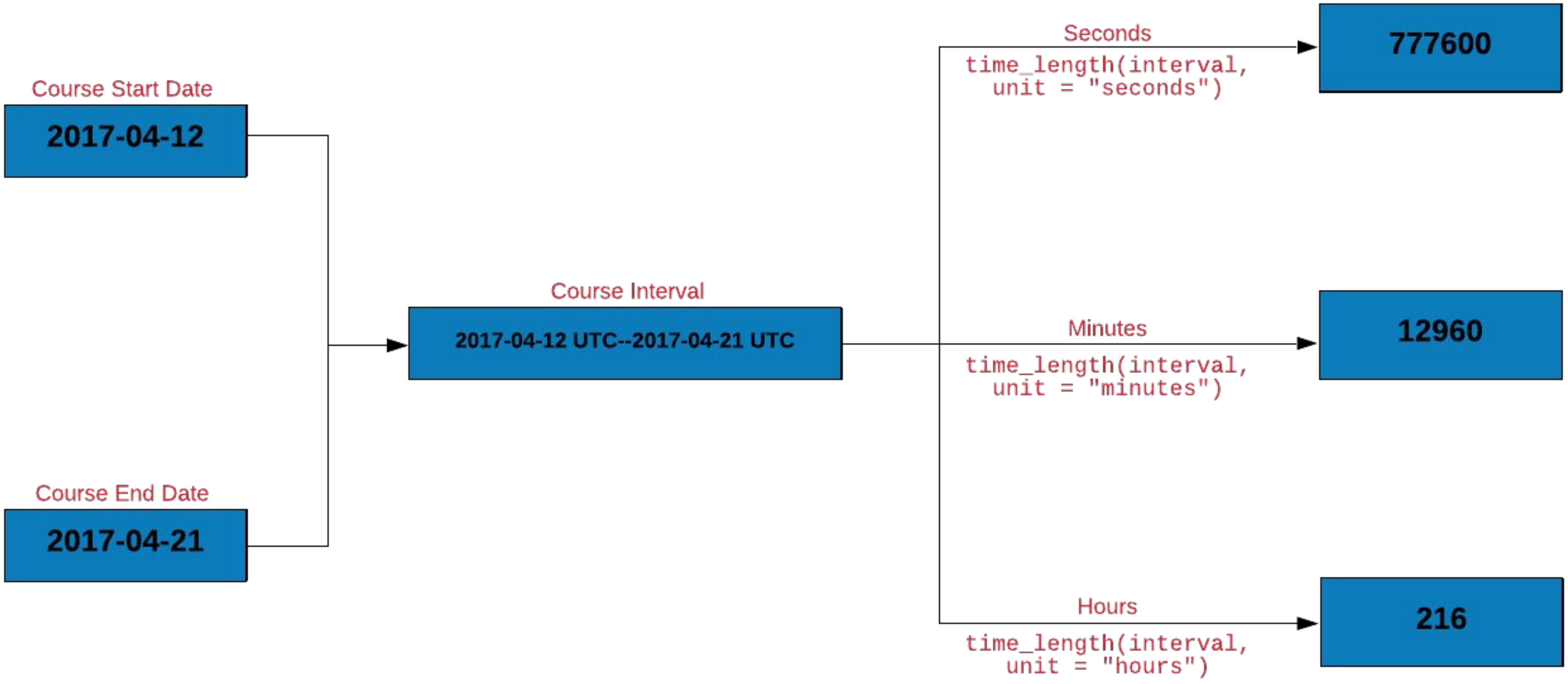
Duration


Course Start Date
2017-04-12
Course End Date
2017-04-21
Course Interval
2017-04-12 UTC--2017-04-21 UTC
Seconds
interval / dseconds()
777600
Minutes
interval / dminutes()
12960
Hours
interval / dhours()
216
Weeks
interval / dweeks()
1.285714
Years
interval / dyears()
0.02465753

Period


Course Start Date
2017-04-12
Course End Date
2017-04-21
Course Interval
2017-04-12 UTC--2017-04-21 UTC
Seconds
as.period(interval,
unit = "seconds")
"777600S"
Minutes
as.period(interval,
unit = "minutes")
"12960M 0S"
Hours
as.period(interval,
unit = "hours")
"216H 0M 0S"

Time Length


Course Start Date
2017-04-12
Course End Date
2017-04-21
Course Interval
2017-04-12 UTC--2017-04-21 UTC
Seconds
time_length(interval,
unit = "seconds")
777600
Minutes
time_length(interval,
unit = "minutes")
12960
Hours
time_length(interval,
unit = "hours")
216

Module 10

Round & Rollback

Rsquared Academy

www.rsquaredacademy.com

Rounding Dates

We will explore functions for rounding dates

- to the nearest value using `round_dates()`
- down using `floor_date()`
- up using `ceiling_date()`

Rounding Dates

The unit for rounding can be any of the following:

- second
- minute
- hour
- day
- week
- month
- bimonth
- quarter
- season
- halfyear
- and year

Your Turn...

- round up R release dates to hours
- round down R release dates to minutes
- rollback R release dates to the beginning of the month

References

- https://lubridate.tidyverse.org/
- https://r4ds.had.co.nz/dates-and-times.html
- https://en.wikipedia.org/wiki/Daylight_saving_time
- https://en.wikipedia.org/wiki/Time_zone
- https://www.worldtimebuddy.com/
- https://en.wikipedia.org/wiki/POSIX

Connect With Us

- → Website
- → Free Online R Courses
- → R Packages
- → Shiny Apps
- → Blog
- → GitHub
- → YouTube
- → Twitter
- → Facebook
- → Linkedin

Thank You

For more information please visit our website
www.rsquaredacademy.com