



Realtime tuning of the ADaM parameters

with an example of the big actigraphy dataset

Mehrnoosh Oghbaie
March 8, 2023

Use of Actigraphy in Clinical trial

Impact of actigraphy in clinical research



Monitoring movement, sleep and wake patterns.

Evaluating subject physical activity level.



Evaluating medication sleep side effects among sleep disorders such as insomnia.

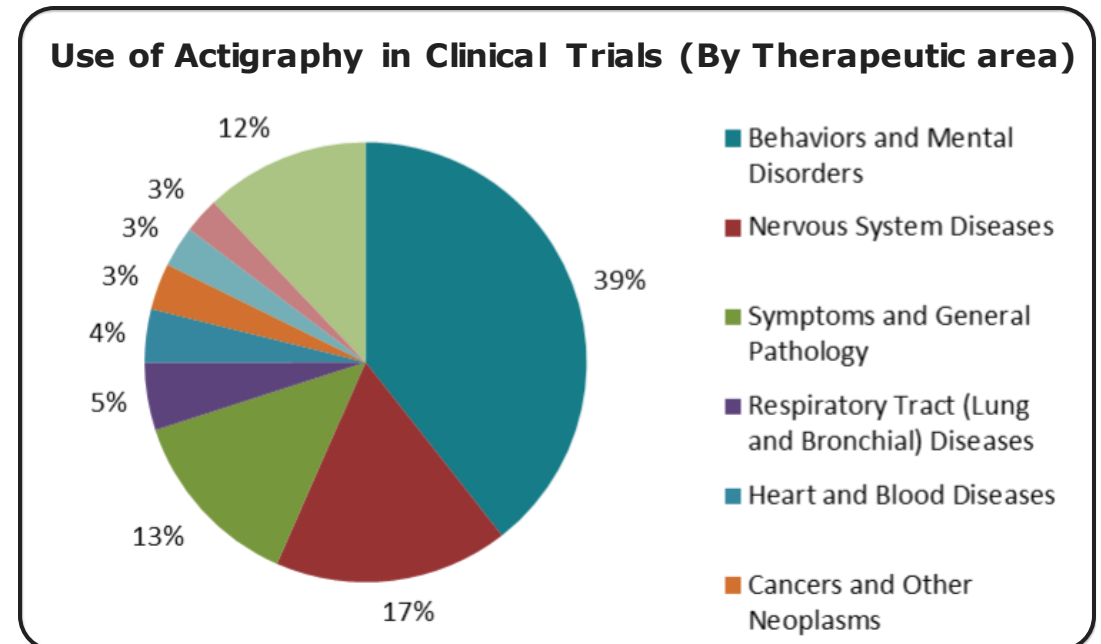


Figure 1: Use of Actigraphy in Clinical Trial by Therapeutic Area

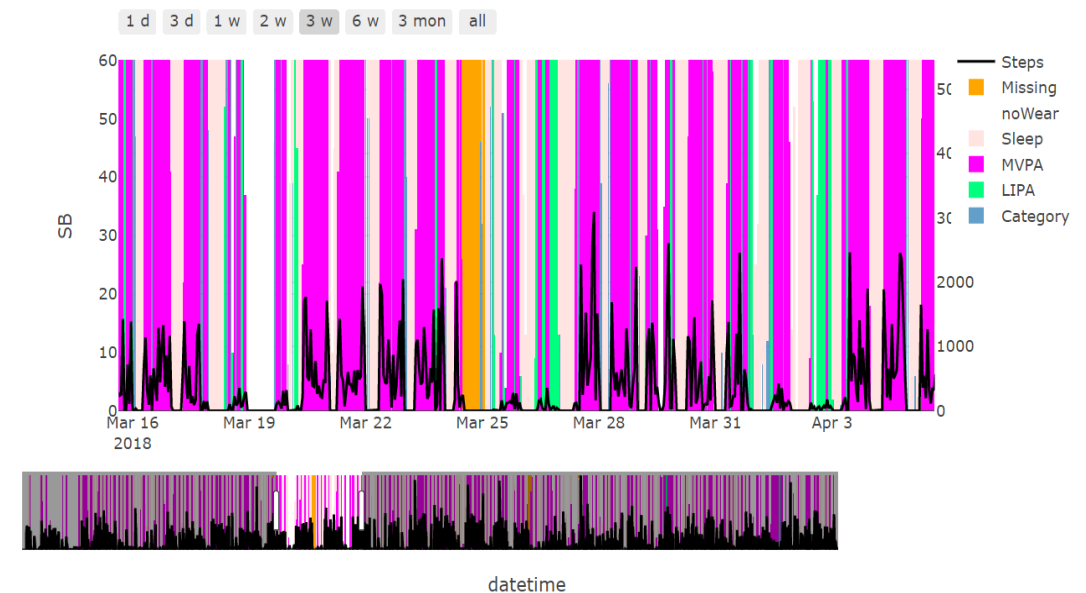


FDA requires that health care data to be **mapped** to **CDISC** standards; and that involves dealing with **inconsistent formats** and **sources**, differences in **terminology**, a wide range of methods to build datasets for aggregation, and variations in overall **data quality**

Why real-time tuning?

- ✓ ActiGraph GT9X is a high-resolution liquid crystal display (LCD) 3-axis accelerometer for measuring continuous secondary movement and position over a period.
- ✓ Analyzing actigraphy data could be cumbersome considering the number of records and size of the input data
- ✓ A common approach is to analyze the summarized time series.
- ✓ While summarized data allows hypothesized testing, **a lot of information gets lost during this process**, including **optimum summarization rules** and **parameters** and how the **summary data is affected** by these **parameters' changes**.

```
Spark | Log | SQL | Help | Refresh Connection Data
local
epochsummarydata
Subject : int :
Timestamp : POSIXct, format: "2019-11-26 20:02:00" "2019"..
Steps : int 0 0 0 0 0
Lux : chr NA NA NA NA ...
HR : int 0 0 0 0 0
AxisXCounts : int 829 616 0 0 0
AxisYCounts : int 839 1381 0 0 0
AxisZCounts : int 578 875 0 0 0
Calories : num 0.777 1.677 0 0 0
Wear : logi TRUE TRUE TRUE TRUE T..
WearSense : logi FALSE FALSE FALSE FAL..
```



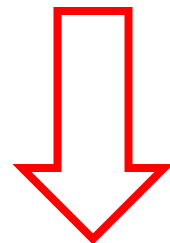
Technical difficulties

Our Solution

Epoch level data consists of millions
of records 2~10 Gb

Producing aDaM file pipeline with
default value takes 3-4 hours

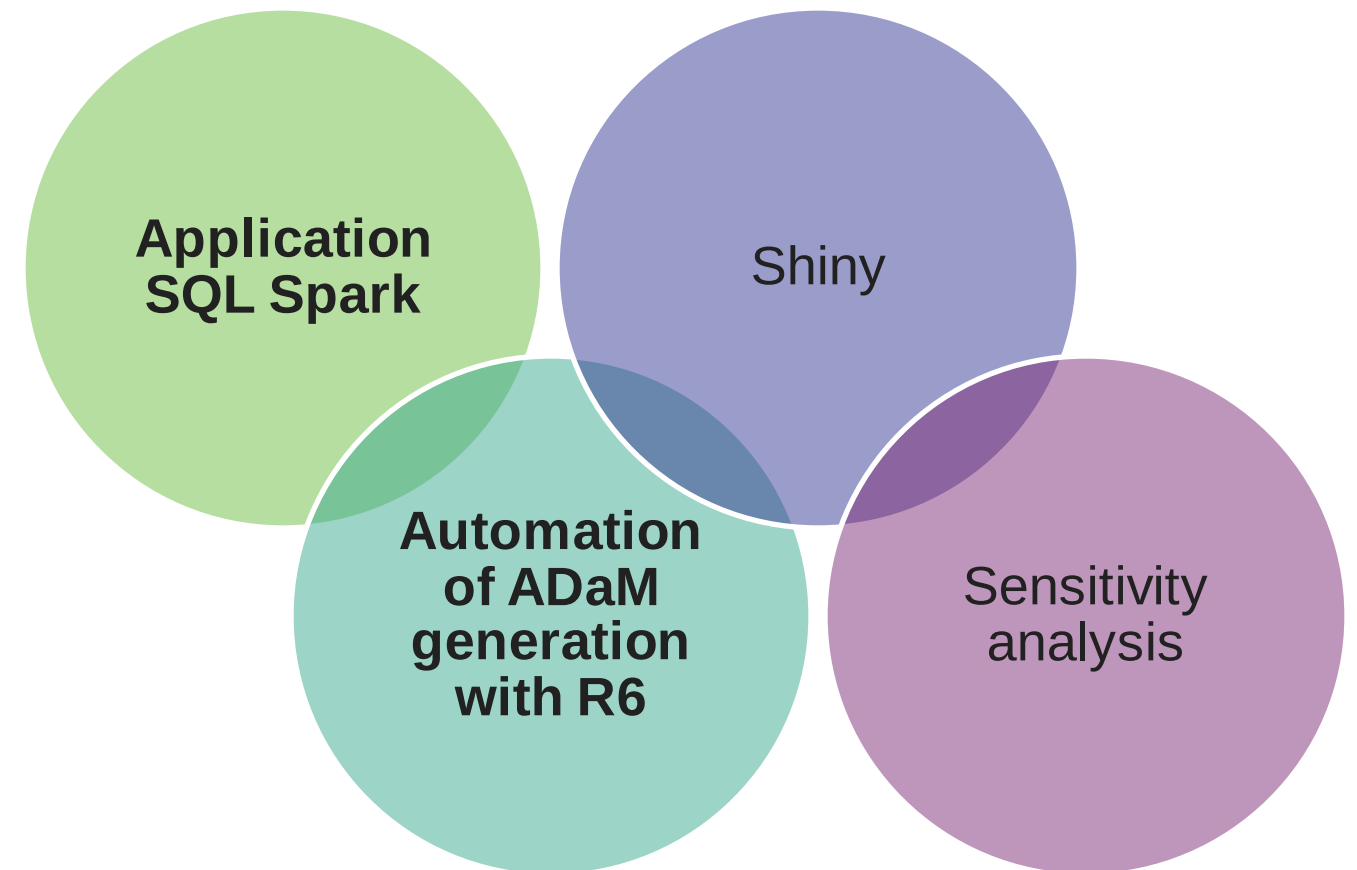
How to effectively compare different
scenarios



Run time was reduced from 3 hours to less than a min

Outline

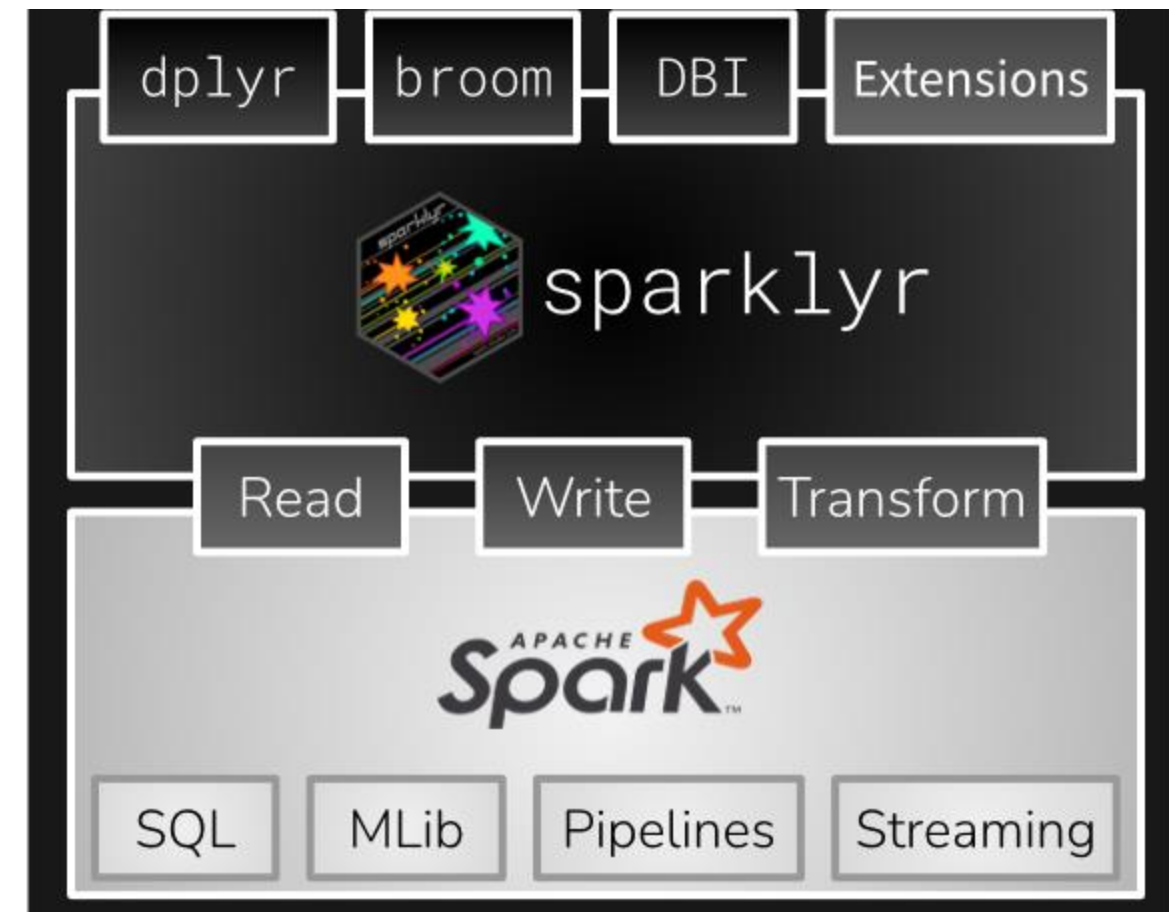
- Why Spark?
- Object Oriented Programming and R6 objects
- Solution at a Glance!
 - Using Spark
 - Dividing functions to static & dynamic
 - Using R6 to implement functions
 - Passing R6 object to shiny module through reactiveValues
 - Updating R6 object with new parameters



Why Spark?

Sparklyr: Big data enabler for R

- ✓ Spark is a fast and open-source framework designed for real-time analysis, and machine learning algorithms
- ✓ Sparklyr interacts with Spark using familiar R interfaces, such as dplyr, broom or DBI.
- ✓ Five SQL functions in dplyr:
 - select() ~ SELECT
 - filter() ~ WHERE
 - arrange() ~ ORDER
 - summarise() ~ aggregators: sum, min, sd, etc.
 - mutate() ~ operators: +, *, log, etc.
- ✓ Gain access to Spark's distributed Machine Learning libraries.
- ✓ Run distributed R code inside Spark.



Object Oriented Programming and R6 objects

Object-Oriented Programming revolves around **objects**. We create these objects from blueprints named **classes**. Each class has **properties** and **methods** that describe how each object behaves

OOP works with four principles:

- **Encapsulation**
- **Inheritance**
- **Abstraction**
- **Polymorphism**

```
<Dog>
Public:
  clone: function (deep = FALSE)
  initialize: function (name = NA, age = 1)
  make_sound: function (sound)
Private:
  age: 4
  hair_color: black
  name: Milo
```

```
Dog <- R6Class(
  classname = "Dog",
  public = list(
    initialize = function(name = NA, age = NA, hair_color = NA) {
      private$name = name
      private$age = age
      private$hair_color = hair_color
    }
  ),
  private = list(
    name = NULL,
    age = NULL,
    hair_color = NULL
  )
)

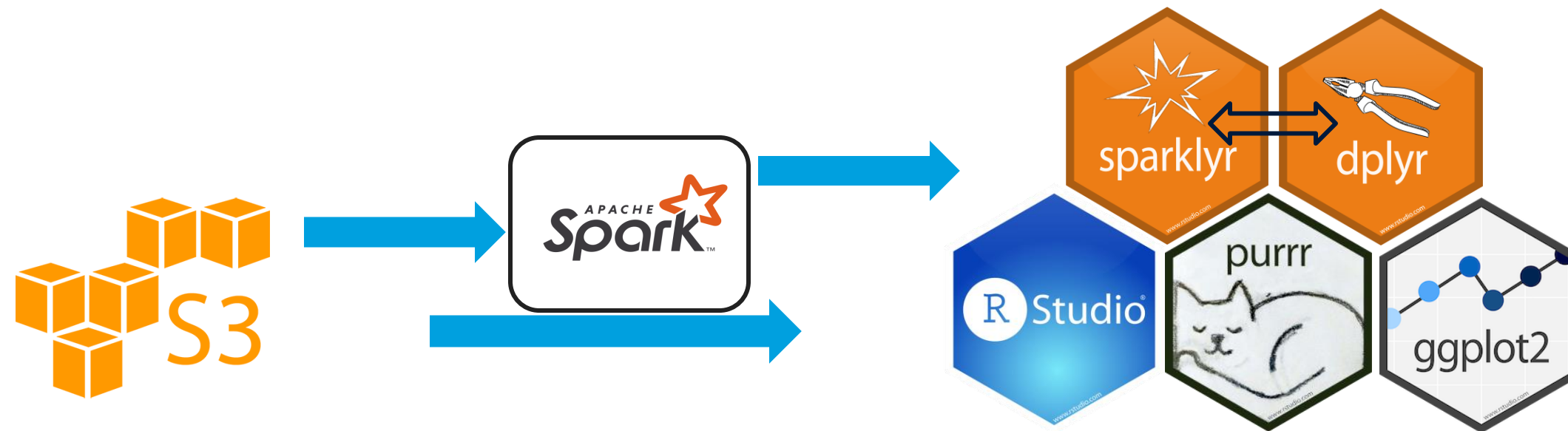
Dog$set("public", "make_sound", function(sound) {
  cat(private$name, " says ", sound, "\n", sep = "")
  invisible(self)
})

d <- Dog$new(name = "Milo", age = 4, hair_color = "black")
d$name
...
```

```
NULL
Milo says WOOF!
```

Solution at a Glance!

Using Spark



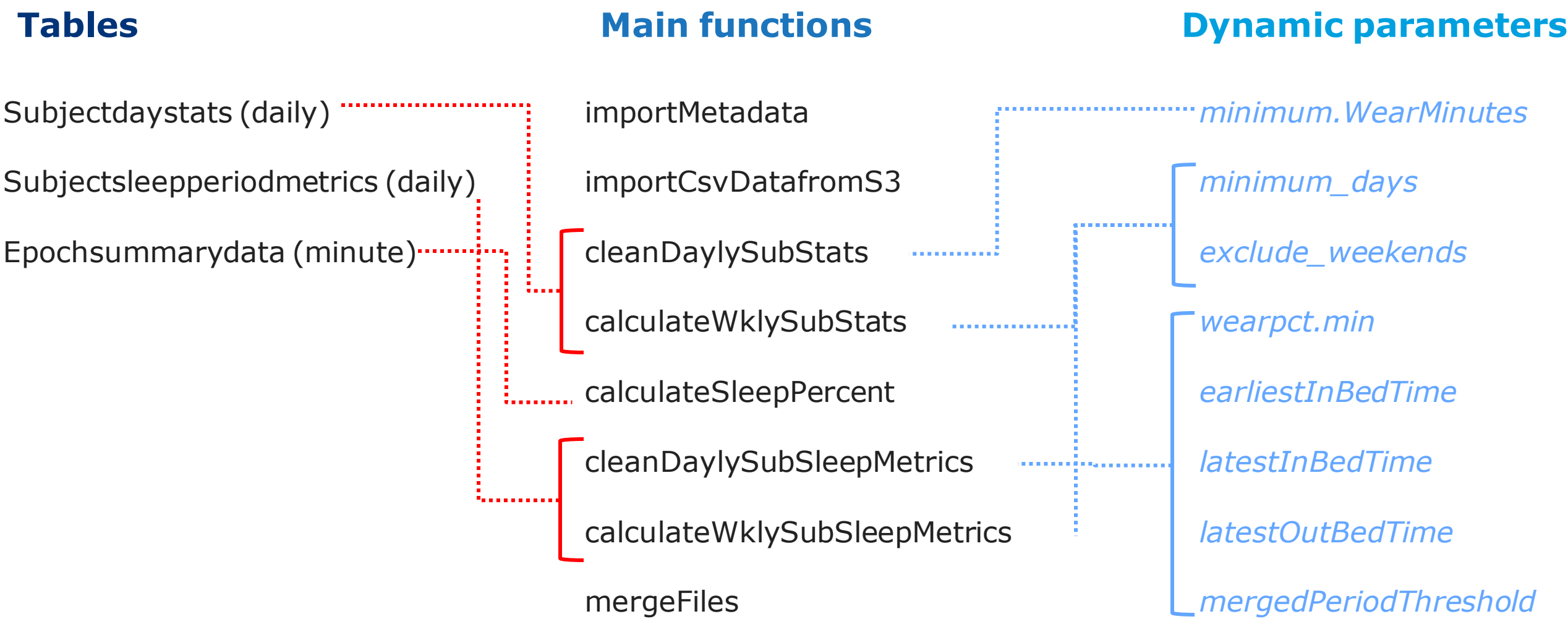
```
## user system elapsed  
## 8.110 0.281 65.036
```

```
## user system elapsed  
## 131.314 57.797 203.569
```

16 times

Solution at a Glance!

Dividing functions to static & dynamic



Solution at a Glance!

Using R6 to implement main functions

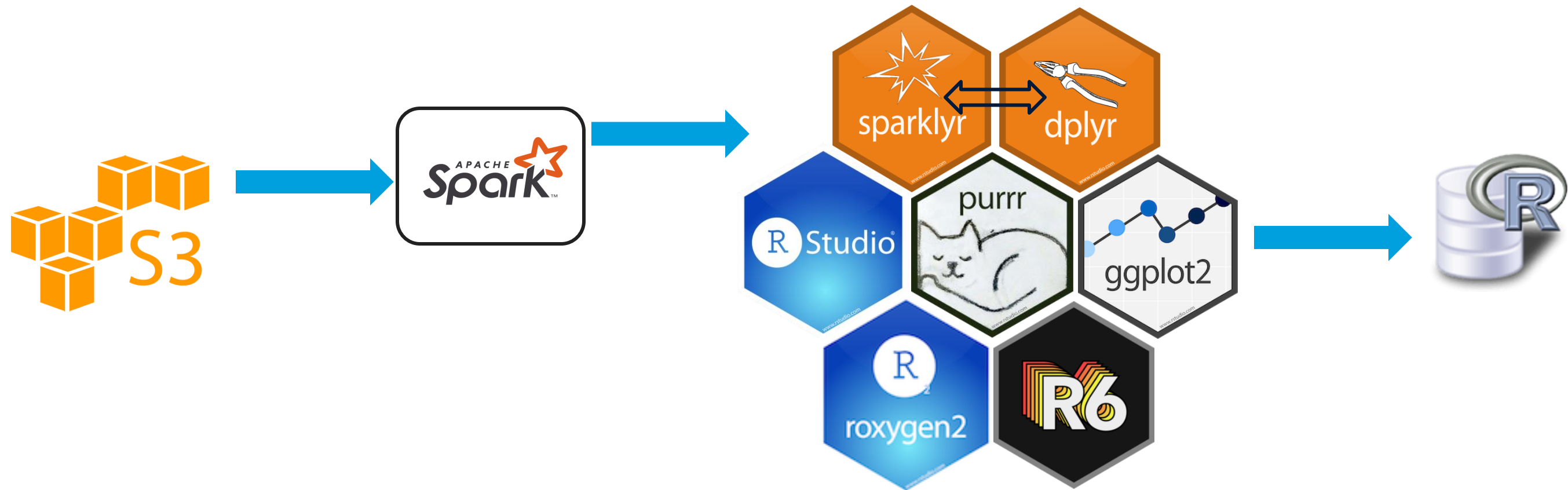
```
actigraph <- Template.project$  
  new()$  
  importMetadata(metadata)$  
  importCsvDatafromS3(sc, bucket, prefix,  
    transfer_date)$  
  cleanDailySubStats(minimum.WearMinutes)$  
  calculateWklySubStats(minimum_days,  
    exclude_weekends)$  
  calculateSleepPercent()$  
  cleanDailySubSleepMetrics(earliestInBedTime,  
    latestInBedTime, latestOutBedTime, wearpct.min,  
    mergedPeriodThreshold)$  
  calculateWklySubSleepMetrics(minimum_days,  
    exclude_weekends)$  
  mergeFiles()
```

Updating R6 object with new parameters

```
actigraph <- actigraph$  
  cleanDailySubStats(minimum.WearMinutes)$  
  calculateWklySubStats(minimum_days,  
    exclude_weekends)$  
  cleanDailySubSleepMetrics(earliestInBedTime,  
    latestInBedTime, latestOutBedTime, wearpct.min,  
    mergedPeriodThreshold)$  
  calculateWklySubSleepMetrics(minimum_days,  
    exclude_weekends)$  
  mergeFiles()
```

Solution at a Glance!

Using R6 to implement them



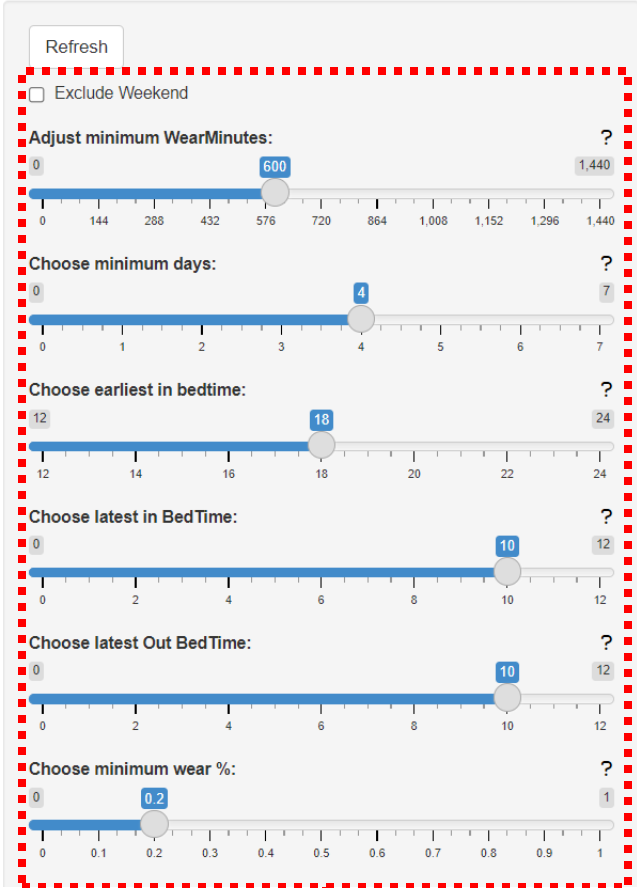
Solution at a Glance!

Passing R6 object to shiny module through reactiveValues

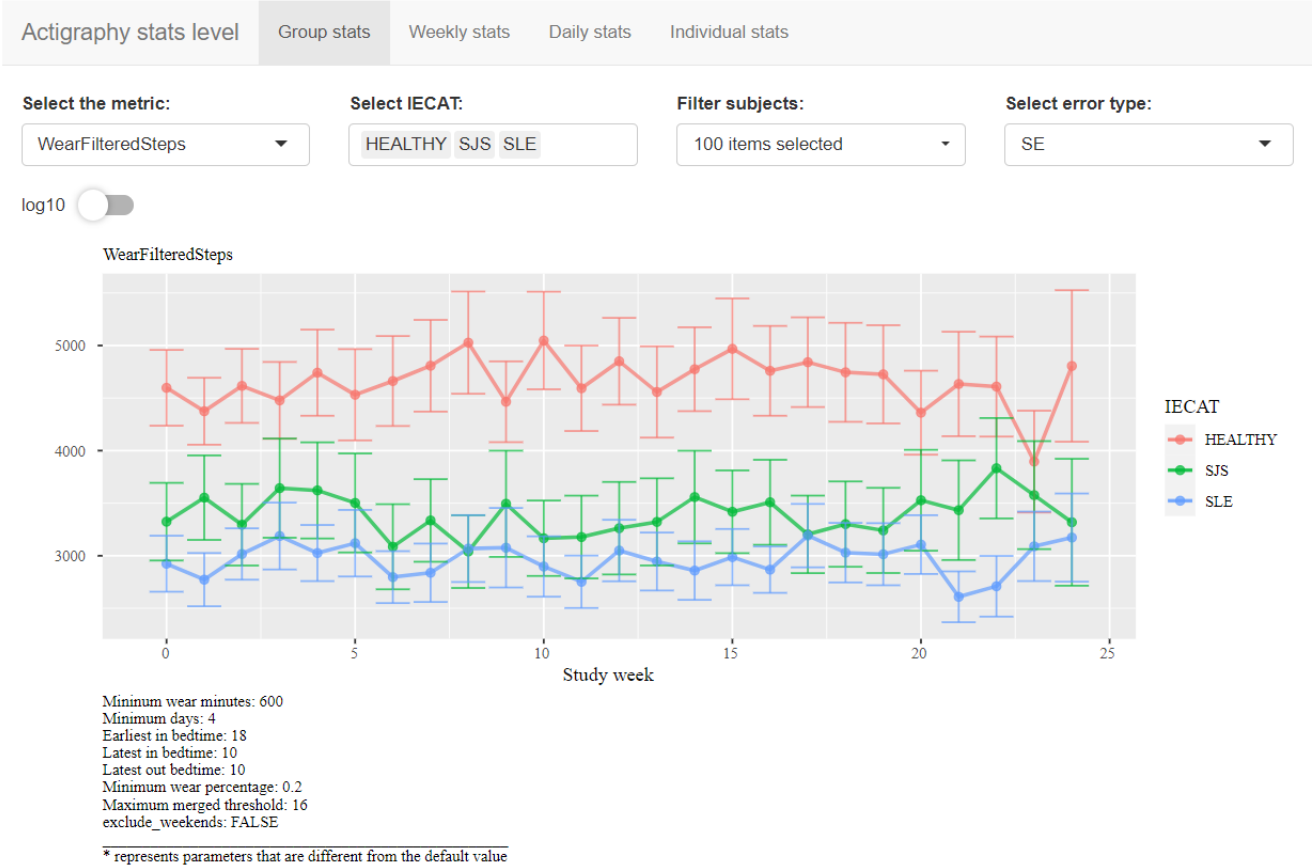


Pass the reactiveValues to shiny modules

Actigraphy parameters tuning



Input parameters



Solution at a Glance!

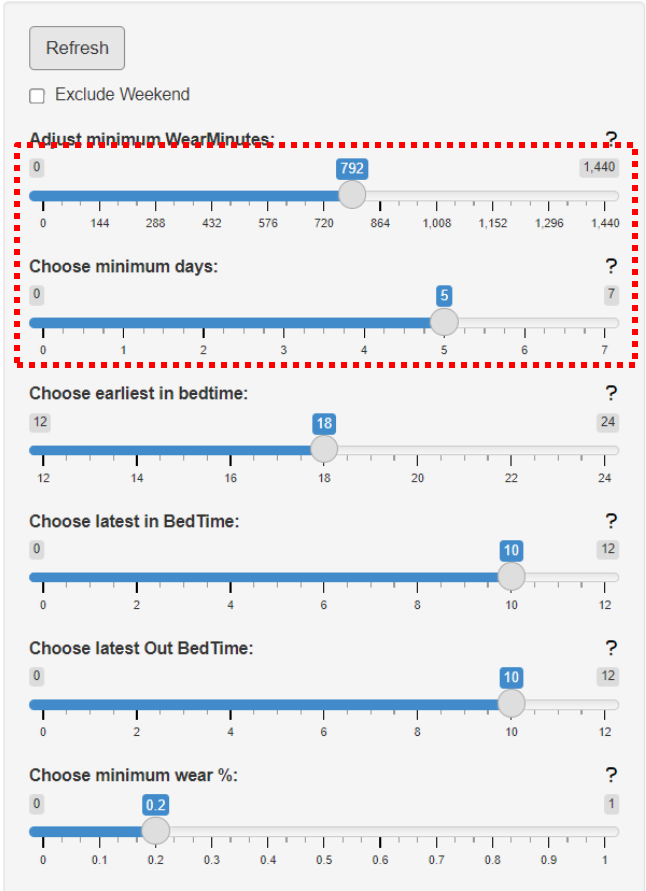
Passing input parameter changes and R6 object to each module



Update the R6 object with new parameters and pass it to shiny module

Minimum wear minutes: 600 -> 792 *
Minimum days: 4 -> 5 *
Earliest in bedtime: 18
Latest in bedtime: 10
Latest out bedtime: 10
Minimum wear percentage: 0.2
Maximum merged threshold: 16
exclude_weekends: FALSE

Actigraphy parameters tuning



References

1. Attalides, N. (n.d.). *Purrr or not to purrr*. Retrieved from R bloggers: <https://www.r-bloggers.com/2018/05/to-purrr-or-not-to-purrr/>
2. GT9X. (n.d.). Retrieved from Actigraph: <https://www.actigraph.nl/en/product/72/actigraph-link-gt9x.html>
3. Ilancheran, M., & Zammit, G. K. (2016, February). Impact of Actigraphy in Clinical Research. *Applied Clinical Trials*.
4. Privé, F. (n.d.). *A guide to parallelism in R*. Retrieved from <https://privefl.github.io/blog/a-guide-to-parallelism-in-r/>
5. *R interface to Apache Spark*. (n.d.). Retrieved from Spark: <https://spark.rstudio.com/>
6. Wickham, H. (n.d.). *Advanced R*. Retrieved from <http://adv-r.had.co.nz/Functional-programming.html>



moghbaie@its.jnj.com