Big Data avec R

S.Bord, T.Mary-Huard

22 juin 2018

Contents

1 Load packages

```r
install.packages("nycflights13",  repos='https://cran.univ-paris1.fr/')
install.packages("sparklyr",  repos='https://cran.univ-paris1.fr/')
library(sparklyr)
spark_install(version = "2.2.0")
library(sparklyr)
library(dplyr)
library(pryr)
```

```
## package 'nycflights13' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\sbord\AppData\Local\Temp\RtmpIPhZOD\downloaded_packages
```

```
##
##   There is a binary version available but the source version is
##   later:
##          binary source needs_compilation
## sparklyr  0.7.0  0.8.4             FALSE
```

2 Parameters

```r
RepName <- 'C:/Users/sbord/Dropbox/Big_Data/TP'
setwd(RepName)
```

3 Local space connection

```
sc <- spark_connect("local", version = "2.2.0")
```

3.1 Add a table in “sc” : copy_to

```
flights_tbl <- dplyr::copy_to(sc, nycflights13::flights, "flights", overwrite=T)
dplyr::src_tbls(sc)
```

```
## [1] "flights"
```

```
ls()
```

```
## [1] "flights_tbl" "RepName"     "sc"
```

```
class(flights_tbl)
```

```
## [1] "tbl_spark" "tbl_sql"   "tbl_lazy"  "tbl"
```

3.2 Data manipulation : fligths example

3.2.1 Selection of fligths

```
flight.sel <- flights_tbl %>%  filter(carrier %in% c("B6", "DL", "EV"))
class(flight.sel)
```

```
## [1] "tbl_spark" "tbl_sql"   "tbl_lazy"  "tbl"
```

3.2.2 Save the selection in “sc” space

```
sdf_register(flight.sel, "flightsel")
```

```
## # Source:   table<flightsel> [?? x 19]
## # Database: spark_connection
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      544            545     -1.00     1004
##  2  2013     1     1      554            600     -6.00      812
##  3  2013     1     1      555            600     -5.00      913
##  4  2013     1     1      557            600     -3.00      709
##  5  2013     1     1      557            600     -3.00      838
##  6  2013     1     1      558            600     -2.00      849
##  7  2013     1     1      558            600     -2.00      853
##  8  2013     1     1      559            559      0         702
##  9  2013     1     1      600            600      0         851
## 10  2013     1     1      601            600      1.00      844
## # ... with more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
src_tbls(sc)
```

```
## [1] "flights"   "flightsel"
```

3.2.3 Save the selection in "R" environment

```
flightsel.inR <- flight.sel %>% collect()
```

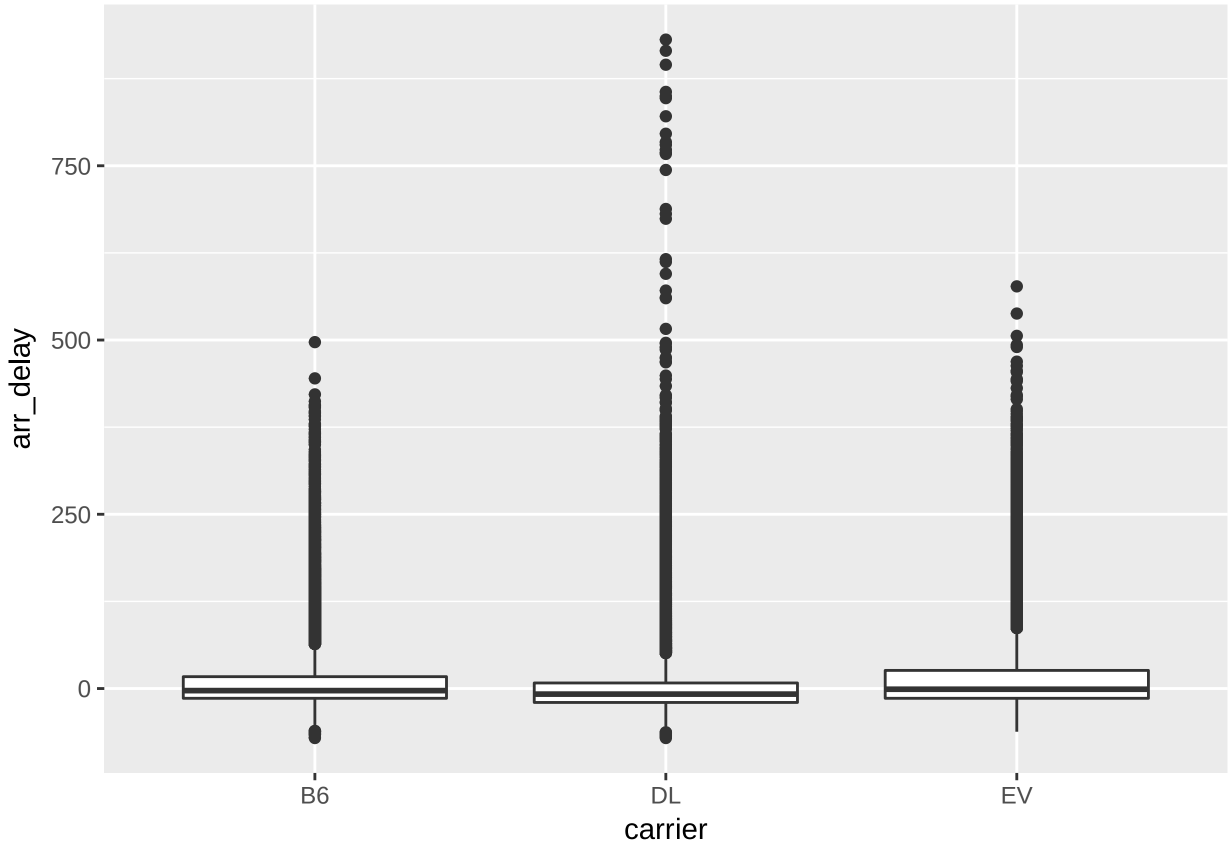
3.2.4 Graph: boxplot of delay by carrier

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
ggplot(flightsel.inR, aes(x=carrier, y=arr_delay)) +  geom_boxplot()
```

```
## Warning: Removed 4103 rows containing non-finite values (stat_boxplot).
```



3.2.5 ANOVA of delay by carrier

```
resanova <- flight.sel %>% na.omit() %>% ml_linear_regression(x = .,
    response = "arr_delay", features = "carrier")
```

```
## * Dropped 4200 rows with 'na.omit' (156918 => 152718)
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

```
summary(resanova)
```

```
## Deviance Residuals (approximate):
##     Min      1Q  Median      3Q     Max
## -77.800 -24.800 -12.639   7.361 913.361
##
## Coefficients:
## (Intercept)  carrier_B6  carrier_EV
##    1.638578    7.821559   14.161606
##
## R-Squared: 0.01523
## Root Mean Squared Error: 45.78
```

3.2.6 Principal component analysis (PCA)

3.2.6.1 Selection of variables for PCA

```
test_tbl <- flights_tbl %>%
  select(one_of(c('dep_time','sched_dep_time','dep_delay',
                  'arr_time','sched_arr_time','arr_delay','carrier')))


DataForPca <- test_tbl %>%  na.omit() %>% filter(carrier=='UA') %>%  select(1:6)
```

```
## * Dropped 9430 rows with 'na.omit' (336776 => 327346)
```

3.2.6.2 Correlation Matrix between PCA variables

```
CorrMat <- ml_corr(DataForPca, method = "pearson")
class(CorrMat)
```

```
## [1] "data.frame"
```

```
CorrMat
```

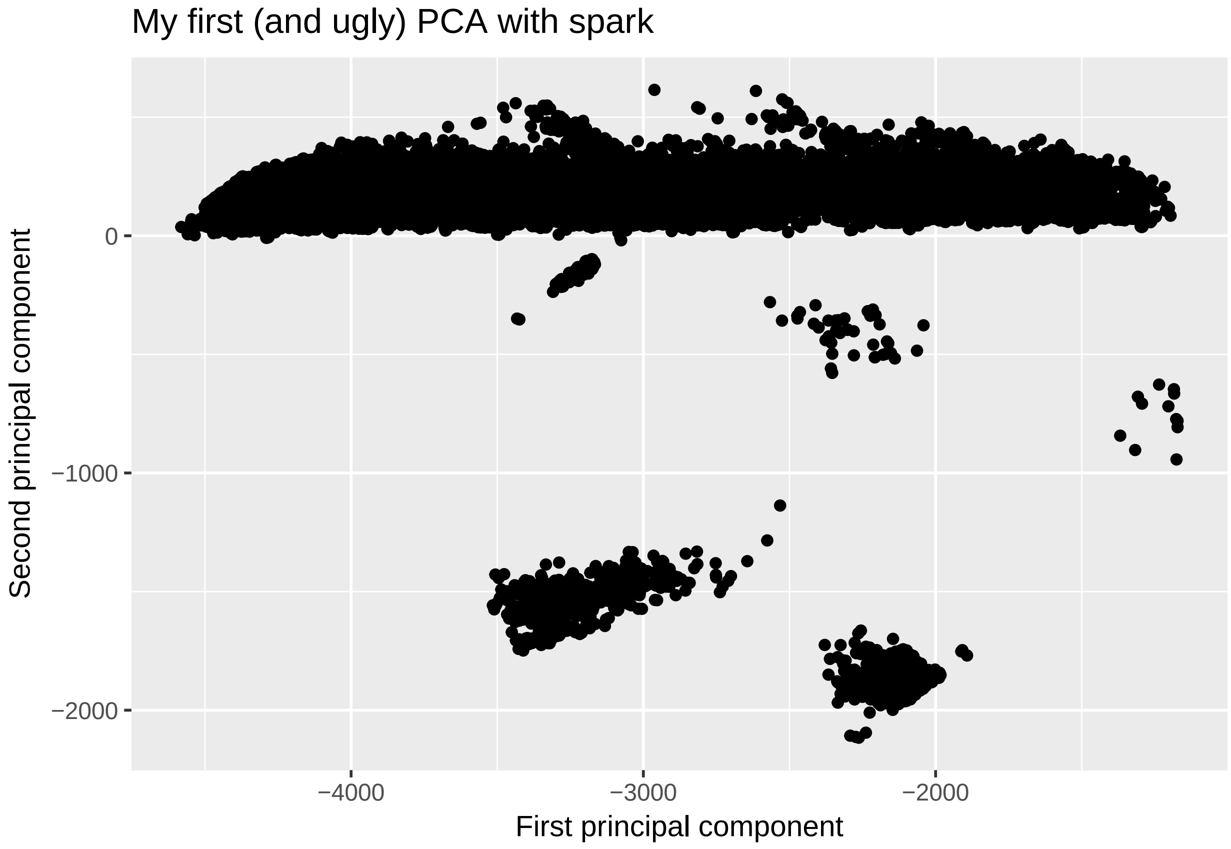
```
##    dep_time sched_dep_time  dep_delay   arr_time sched_arr_time arr_delay
## 1 1.0000000      0.9839321 0.28886830 0.65822090      0.7955738 0.2594648
## 2 0.9839321      1.0000000 0.20242720 0.66496117      0.8042313 0.1811069
## 3 0.2888683      0.2024272 1.00000000 0.02980382      0.1612802 0.8853862
## 4 0.6582209      0.6649612 0.02980382 1.00000000      0.7872813 0.0456680
## 5 0.7955738      0.8042313 0.16128020 0.78728129      1.0000000 0.1547960
## 6 0.2594648      0.1811069 0.88538623 0.04566800      0.1547960 1.0000000
```

3.2.6.3 PCA

```
PcaResults <- ml_pca(DataForPca)

Coordinates <- sdf_project(PcaResults, DataForPca, features = rownames(PcaResults$pc)) %>%
    select(starts_with("PC"))

Coordinates %>% select(1:2) %>% collect() %>% ggplot(aes(PC1, PC2)) +
    geom_point(aes(PC1, PC2)) + labs(x = "First principal component",
    y = "Second principal component", title = "My first (and ugly) PCA with spark")
```



```
PcaResults
```

```
## Explained variance:
##
##          PC1          PC2          PC3          PC4          PC5
## 0.8297183201 0.1193196281 0.0445887135 0.0046865005 0.0015258961
##          PC6
## 0.0001609417
##
## Rotation:
##                         PC1         PC2          PC3          PC4
## dep_time       -0.498075838 -0.47652499 -0.232778127  0.550318170
## sched_dep_time -0.482546608 -0.44444822 -0.199644019 -0.606353053
## dep_delay      -0.007150391 -0.02617198  0.008342857  0.381268435
## arr_time       -0.507439946  0.74338599 -0.434952353  0.015994794
## sched_arr_time -0.511333789  0.14660631  0.846539874  0.008437846
## arr_delay      -0.007782056 -0.02426102  0.008584814  0.428711134
##                        PC5          PC6
## dep_time        0.40865689  0.028428728
## sched_dep_time -0.40183848 -0.024872274
## dep_delay      -0.49357427 -0.781161387
## arr_time       -0.02064289 -0.004056941
## sched_arr_time  0.01848889  0.001245919
## arr_delay      -0.65356090  0.623170967
```

3.2.6.4 Kmeans

```
KmeansResults <- ml_kmeans(DataForPca, centers = 3, iter.max = 100,
    features = rownames(PcaResults$pc))
arrange(KmeansResults$centers, arr_delay)
```

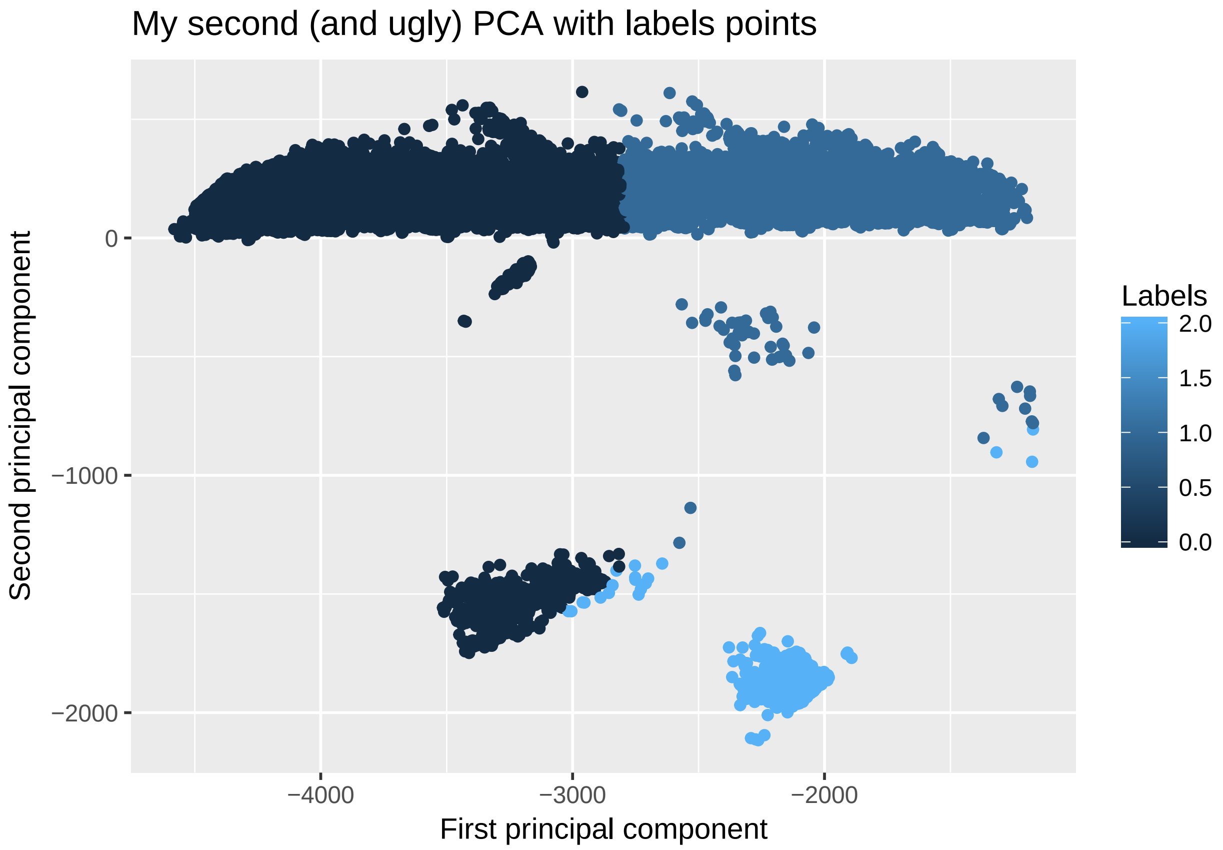
```
##    dep_time sched_dep_time dep_delay   arr_time sched_arr_time arr_delay
## 1  865.7841        867.640  4.588382 1123.46301     1135.04209 -4.523637
## 2 1705.1535       1676.461 17.882919 1880.37853     1936.08586 10.083142
## 3 2112.5663       2065.514 33.342752   68.87715       87.98649 21.382064
```

```
Labels <- ml_predict(KmeansResults, DataForPca) %>% select(prediction) %>%
    collect()
table(Labels$prediction)
```

```
##
##     0     1     2
## 30526 26442   814
```

```
CoordinatesWithLabels <- Coordinates %>% collect() %>% mutate(Labels = Labels$prediction)

CoordinatesWithLabels %>% ggplot(aes(PC1, PC2)) + geom_point(aes(PC1,
    PC2, color = Labels)) + labs(x = "First principal component",
    y = "Second principal component", title = "My second (and ugly) PCA with labels points")
```



3.3 Export “sc” data in a file

```
filename_to_save <- paste0(RepName, "/export_exemple.csv")
spark_write_csv(DataForPca, path = filename_to_save, header = FALSE,
    delimiter = ";")
```

4 Memory allocation

```
sc %>% spark_context %>% invoke("getRDDStorageInfo")
```

```
## [[1]]
## <jobj[468]>
##   org.apache.spark.storage.RDDInfo
##   RDD "In-memory table `flights`" (9) StorageLevel: StorageLevel(memory, deserialized, 1 replicas); C
##
## [[2]]
## <jobj[469]>
##   org.apache.spark.storage.RDDInfo
##   RDD "*Filter (carrier#38 IN (B6,DL,EV) && AtLeastNNulls(n, year#29,month#30,day#31,dep_time#32,sche
## +- InMemoryTableScan [year#29, month#30, day#31, dep_time#32, sched_dep_time#33, dep_delay#34, arr_ti
##       +- InMemoryRelation [year#29, month#30, day#31, dep_time#32, sched_dep_time#33, dep_delay#34, a
```

```
url <- sc %>% spark_context %>% invoke("uiWebUrl") %>% invoke("get")
browseURL(paste(url, "storage", sep = "/"))
app_id <- sc %>% spark_context %>% invoke("applicationId")
httr::GET(paste(url, "api", "v1", "applications", app_id, "storage",
    "rdd", sep = "/"))
```

```
## Response [http://127.0.0.1:4040/api/v1/applications/local-1529921815383/storage/rdd]
##   Date: 2018-06-25 10:17
##   Status: 200
##   Content-Type: application/json
##   Size: 1.44 kB
## [ {
##   "id" : 52,
##   "name" : "*Filter (carrier#38 IN (B6,DL,EV) && AtLeastNNulls(n, year#2...
##   "numPartitions" : 1,
##   "numCachedPartitions" : 1,
##   "storageLevel" : "Memory Deserialized 1x Replicated",
##   "memoryUsed" : 10471128,
##   "diskUsed" : 0
## }, {
##   "id" : 9,
## ...
```

5 Local space disconnection

```
spark_disconnect(sc)
```