sparklyr

sparklyr — R interface for Apache Spark 의 설치와 사용법

- CentOS 6.7
- 사전 준비

```
sudo yum -y install libcurl-devel

install.packages("sparklyr" , repos = 'http://cran.nexr.com')

library(sparklyr)
# spark_install(version = "1.6.2")

if (nchar(Sys.getenv("SPARK_HOME")) < 1) {
  Sys.setenv(SPARK_HOME = "/home/goodmit/spark")
}
sc <- spark_connect(master = "local")
```

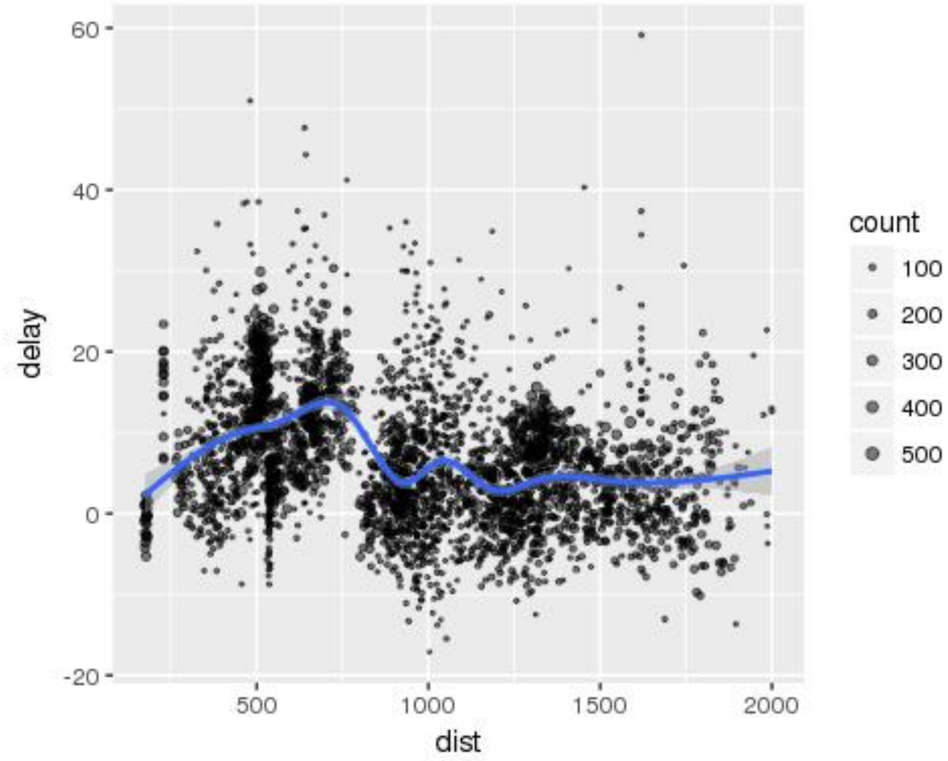
Reading Data

```
install.packages("dplyr", repos = "http://cran.nexr.com" )
install.packages("nycflights13", repos = "http://cran.nexr.com" )
install.packages("Lahman" , repos = "http://cran.nexr.com" )

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(nycflights13)

iris_tbl <- copy_to(sc, iris)

## The following columns have been renamed:
## - 'Sepal.Length' => 'Sepal_Length' (#1)
## - 'Sepal.Width'  => 'Sepal_Width'  (#2)
## - 'Petal.Length' => 'Petal_Length' (#3)
## - 'Petal.Width'  => 'Petal_Width'  (#4)
```

```
flights_tbl <- copy_to(sc, nycflights13::flights, "flights")
batting_tbl <- copy_to(sc, Lahman::Batting, "batting")

src_tbls(sc)

## [1] "batting" "flights" "iris"
```

Using dplyr

```
# filter by departure delay
flights_tbl %>% filter(dep_delay == 2)

## Source:   query [?? x 19]
## Database: spark connection master=local[1] app=sparklyr local=TRUE
## 
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1   2013     1     1      517            515         2      830
## 2   2013     1     1      542            540         2      923
## 3   2013     1     1      702            700         2     1058
## 4   2013     1     1      715            713         2      911
## 5   2013     1     1      752            750         2     1025
## 6   2013     1     1      917            915         2     1206
## 7   2013     1     1      932            930         2     1219
## 8   2013     1     1     1028           1026         2     1350
## 9   2013     1     1     1042           1040         2     1325
## 10  2013     1     1     1231           1229         2     1523
## # ... 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 <dbl>

delay <- flights_tbl %>%
  group_by(tailnum) %>%
  summarise(count = n(), dist = mean(distance), delay = mean(arr_delay)) %>%
  filter(count > 20, dist < 2000, !is.na(delay)) %>%
  collect()

install.packages('ggplot2', repos = 'http://cran.nexr.com')

# plot delays
library(ggplot2)
ggplot(delay, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  geom_smooth() +
  scale_size_area(max_size = 2)

## `geom_smooth()` using method = 'gam'
```



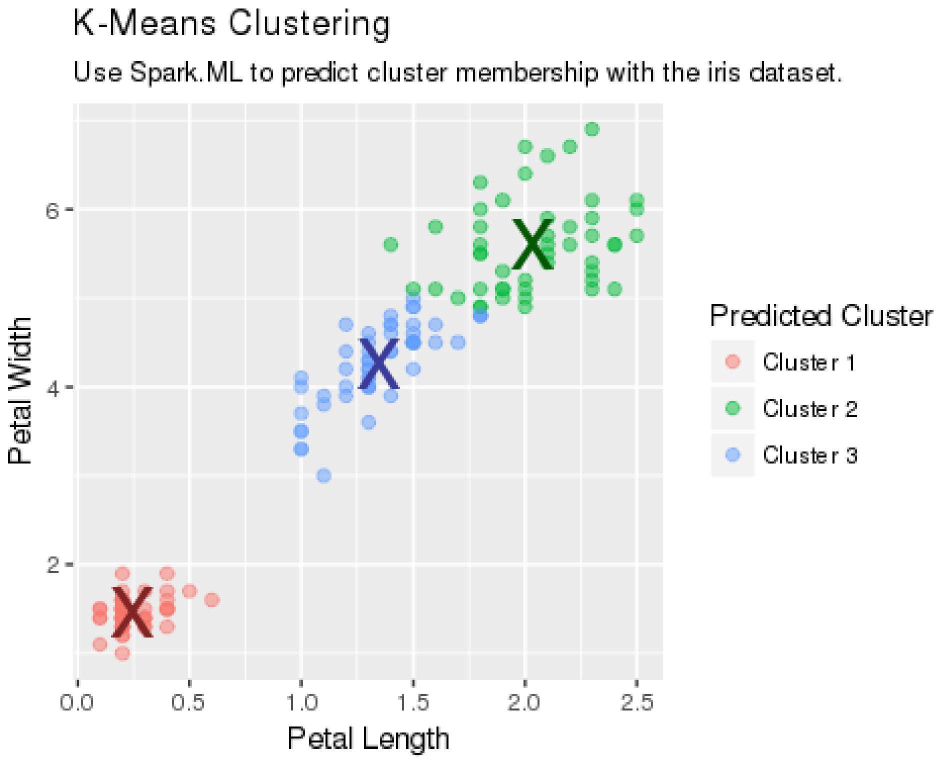
spark + R 의 장점

- Spark 에서 병렬처리가 가능하도록 구현한 머신러닝 LIB 을 사용하고, 결과를 R 의 시각화 패키지로 보여줌.

K-Means Clustering

```
kmeans_model <- iris_tbl %>%
  select(Petal_Width, Petal_Length) %>%
  ml_kmeans(centers = 3)

# print our model fit
print(kmeans_model)

## K-means clustering with 3 clusters
##
## Cluster centers:
##   Petal_Width Petal_Length
## 1    0.246000     1.462000
## 2    2.037500     5.595833
## 3    1.342308     4.269231
##
## Within Set Sum of Squared Errors =  31.37136
```

```
predicted <- sdf_predict(kmeans_model, iris_tbl) %>%  collect

table(predicted$Species, predicted$prediction)

##
##                0  1  2
##   setosa     50  0  0
##   versicolor  0  2 48
##   virginica   0 46  4

# plot cluster membership
sdf_predict(kmeans_model) %>%
  collect() %>%
  ggplot(aes(Petal_Length, Petal_Width)) +
  geom_point(aes(Petal_Width, Petal_Length, col = factor(prediction + 1)),
             size = 2, alpha = 0.5) +
  geom_point(data = kmeans_model$centers, aes(Petal_Width, Petal_Length),
             col = scales::muted(c("red", "green", "blue")),
             pch = 'x', size = 12) +
  scale_color_discrete(name = "Predicted Cluster",
                       labels = paste("Cluster", 1:3)) +
  labs(
    x = "Petal Length",
    y = "Petal Width",
    title = "K-Means Clustering",
    subtitle = "Use Spark.ML to predict cluster membership with the iris data
set."
  )
```



K-Means Clustering

Use Spark.ML to predict cluster membership with the iris dataset.