Introduction to sparklyr

We will largely follow chapters **2** and **3** of **Mastering Spark with R**, https://therinspark.com.

First install the following packages if you do not already have them already, and load them with the `library()` function:

```
library(sparklyr)
library(dplyr)
library(ggplot2)
library(knitr)
```

Preliminiaries

If you are working on EIDF, first make sure that the default working directory in RStudio is your folder. In RStudio, select Tools -> Global Options. Change the default working directory to be /work/eidf071/eidf071/.

To confirm the change has taken effect, close and then reopen RStudio, and type getpw() into the console. It should show your working directory correctly as above.

Connecting

```
sc = spark_connect master = 'local'
```

Examples:

```
#install.packages("nycflights13", "Lahman")
library(dplyr)
library(ggplot2)
library(jsonlite)
```

```
## Warning: package 'jsonlite' was built under R version 4.4.1
```

```
iris_tbl <- copy_to(sc, iris, overwrite = TRUE)
flights_tbl <- copy_to(sc, nycflights13::flights, "flights", overwrite = TRUE)
batting_tbl <- copy_to(sc, Lahman::Batting, "batting", overwrite = TRUE)
src_tbls(sc)
```

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

```
flights_tbl %>% filter(dep_delay == 2)
```

```
## # Source:   SQL [?? x 19]
## # Database: spark_connection
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      542            540         2      923            850
##  3  2013     1     1      702            700         2     1058           1014
##  4  2013     1     1      715            713         2      911            850
##  5  2013     1     1      752            750         2     1025           1029
##  6  2013     1     1      917            915         2     1206           1211
```

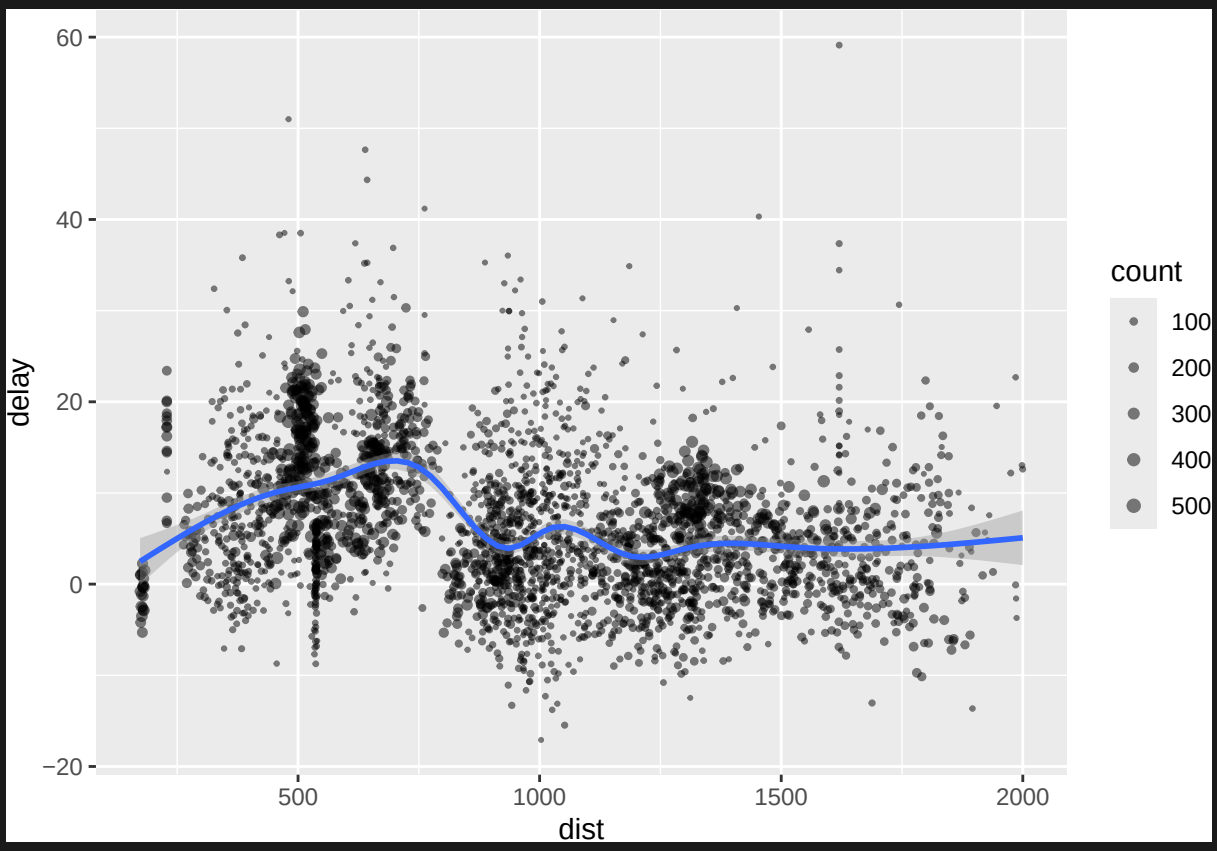
```
##  7  2013     1     1      932           930         2     1219           1225
##  8  2013     1     1     1028          1026         2     1350           1339
##  9  2013     1     1     1042          1040         2     1325           1326
## 10  2013     1     1     1231          1229         2     1523           1529
## # i more rows
## # i 11 more variables: 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>
```

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

# Saving as json
#write_json(delay, "file.json")

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' and formula = 'y ~ s(x, bs = "cs")'
```



The following code will take the built-in mtcars dataset, stored in an R dataframe, and put it into a spark dataframe. We will use this dataset in many of our examples.

```
cars = copy_to(sc, mtcars, overwrite = TRUE)
```

Data input/output

Write to a csv file

This will create a folder in your working directory called `cars.csv`. It contains a csv with the cars data in it.

```
spark_write_csv(cars, "cars.csv")
```

Note that running this more than once will result in an error because spark_write_csv will not overwrite a folder which is already created. You may need to delete the folder before running the code again.

Read from a csv file

```
spark_read_csv(sc, 'cars.csv') %>%
  head() %>%
  kable()
```

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

Data wrangling

Familiar commands from dplyr work as you would expect, but now they instead connect to Spark and would be run in parallel across the cluster.

Create a new column

```
cars = mutate(cars, transmission = ifelse(am == 0, 'automatic', 'manual'))
```

Select columns

```
select(cars, am, transmission) %>%
  head() %>%
  kable()
```

am	transmission
1	manual
1	manual
1	manual
0	automatic
0	automatic
0	automatic

Calculate the mean of each column

```
summarise_all(cars, mean, na.rm = TRUE) %>%
  kable()
```

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	transmission
20.09062	6.1875	230.7219	146.6875	3.596563	3.21725	17.84875	0.4375	0.40625	3.6875	2.8125	NA