Bigquery

```
library(bigrquery)
library(tidyverse)
```

```
## -- Attaching packages ------ tidyverse 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts --------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
con <- NULL  # to avoid knit error
```

When using bigrquery interactively, you'll be prompted to authorize bigrquery in the browser.

- login the Google Cloud Platform
- create a new project
- enable BigQuery API
- add public data

```
bigrquery::bq_auth()
```

```
# replace it with your project id
project <- "adept-vigil-269305"
```

Some test data.

```
result <- bq_project_query(
  project,
  "SELECT * FROM `bigquery-public-data.samples.gsod` LIMIT 100;")
```

```
bq_table_download(result)
```

Upload dataset

You could upload via the web interface or using `bq_` functions.

```
mydataset <- bq_dataset(project, "mydataset")
bq_dataset_create(mydataset)
bq_dataset_exists(mydataset)
```

Let's try to upload the `mtcars` dataset and pretend that it is huge.

```
ta <- bq_table(mydataset, "mtcars")
bq_table_create(ta)
bq_table_exists(ta)

cars <- mtcars %>%
  mutate(cyl = as_factor(cyl), vs = as_factor(vs), am = as_factor(am))
bq_table_upload(ta, cars, fields = as_bq_fields(cars))
```

Now, let's have some fun.

There are three interfaces provided by `bigrquery`. - Low level API overy REST - DBI - dplyr

*bq_*

```
result <- bq_project_query(
  project,
  "SELECT * FROM `adept-vigil-269305.mydataset.mtcars` where `mpg` < 30")
bq_table_download(result)
```

```
library(DBI)
con <- dbConnect(
  bigquery(),
  project = project,
  dataset = "mydataset"
)
```

DBI

```
con %>% dbGetQuery("SELECT * FROM `adept-vigil-269305.mydataset.mtcars` WHERE `mpg` < 30")
```

dplyr

```
con %>% tbl("mtcars") %>%
  filter(mpg < 30) %>%
  collect()
```

```
SELECT * FROM `adept-vigil-269305.mydataset.mtcars` WHERE `mpg` < 30;
```

Running linear regression in Bigquery

Create a column which indicated if the data should be trained.

```
CREATE OR REPLACE TABLE `adept-vigil-269305.mydataset.mtcars2` AS
  SELECT *,
  RAND() < 0.9 as `train`
  FROM `adept-vigil-269305.mydataset.mtcars`
```

```
CREATE OR REPLACE MODEL `adept-vigil-269305.mydataset.mtcars_model`
OPTIONS
  (model_type='linear_reg',
    input_label_cols=['mpg']) AS
SELECT
  `mpg`,
  `cyl`,
  `disp`,
  `hp`,
  CAST(`gear` AS string) AS `gear`
FROM
  `adept-vigil-269305.mydataset.mtcars2`
WHERE
  `train` = true
```

If you want to delete the model

```
DROP MODEL `adept-vigil-269305.mydataset.mtcars_model`;
```

To do prediction

```
SELECT * FROM ML.PREDICT(MODEL `adept-vigil-269305.mydataset.mtcars_model`, (
  SELECT
    `cyl`,
    `disp`,
    `hp`,
    CAST(`gear` AS string) AS `gear`
  FROM `adept-vigil-269305.mydataset.mtcars2` WHERE `train` = false
))
```

Running logisitic regression in Bigquery

```
library(kernlab)
data(spam)
ta <- bq_table(mydataset, "spam")
bq_table_create(ta)
bq_table_exists(ta)
bq_table_upload(ta, spam, fields = as_bq_fields(spam))
```

```
CREATE OR REPLACE VIEW
  `adept-vigil-269305.mydataset.spam_view` AS
SELECT
  `all` AS `a`,
  `over` AS `o`,
```

```
  `order` AS `ord`,
  * EXCEPT (`all`, `over`, `order`),
  CASE
    WHEN MOD(ROW_NUMBER() OVER(), 10) < 8 THEN 'training'
    WHEN MOD(ROW_NUMBER() OVER(), 10) = 8 THEN 'evaluation'
    WHEN MOD(ROW_NUMBER() OVER(), 10) = 9 THEN 'prediction'
  END AS dataframe
FROM
  `adept-vigil-269305.mydataset.spam`
```

```
CREATE OR REPLACE MODEL
  `adept-vigil-269305.mydataset.spam_model`
OPTIONS
  ( model_type='LOGISTIC_REG',
    input_label_cols=['type'],
    max_iterations=15) AS
SELECT
  *
FROM
  `adept-vigil-269305.mydataset.spam_view`
WHERE
  dataframe = 'training'
```

Evalute the model

```
SELECT
  *
FROM
  ML.EVALUATE (MODEL `adept-vigil-269305.mydataset.spam_model`,
    (
    SELECT
      *
    FROM
      `adept-vigil-269305.mydataset.spam_view`
    WHERE
      dataframe = 'evaluation'
    )
  )
```

Prediction

```
SELECT
  *
FROM
  ML.PREDICT (MODEL `adept-vigil-269305.mydataset.spam_model`,
    (
    SELECT
      * EXCEPT(`type`)
    FROM
```

```
    `adept-vigil-269305.mydataset.spam_view`
  WHERE
    dataframe = 'prediction'
  )
)
```

Reference

BigQuery: https://cloud.google.com/bigquery-ml/docs/reference/standard-sql