# Ejercicios: Pandas, GCS y BigQuery

## Pre-Ejercicio 1: Accedé a la nube de Google

Si tenés usuario y contraseña de Gmail, es fácil. Entrá a https://console.cloud.google.com/ y registrate. No te preocupes, aunque pida un número de tarjeta de crédito, Google nos dice que no cobrará nada a menos que le demos “Ok” para cuando se nos acaben nuestros $300usd de *trial.*

## Pre-Ejercicio 2: Crea un Dataset en BigQuery

Crear un Dataset en BigQuery usando la UI de la nube. Llamalo “*dataset_dev*” (o el nombre que quieras). **IMPORTANTE: Elegí que sea un Dataset en la región “us-east1” para que sea lo más barato posible.**

## Pre-Ejercicio 3: Crea un Bucket

Crear un Bucket en el servicio Cloud Storage. Ponele el nombre descriptivo que quieras. **IMPORTANTE: Elegí que sea un Dataset en la región “us-east1” para que sea lo más barato posible.**

## Pre-Ejercicio 4: Logueate en tu terminal a tu usuario de Google

Acá vas a usar la terminal para que cualquier código o acción que corras tanto en la terminal como en Python, use tus credenciales y la nube y los recursos que creaste anteriormente. Para loguearte, vas a tener que usar la CLI de *gcloud*, guiate con esto*:* *https://cloud.google.com/sdk/docs/install?hl=es-419*

Una vez que tengas la CLI instalada, asegurate de que estar bien logueado haciendo:

gcloud auth login

y

gcloud auth application-default login

Ambos te deberían llevar al navegador para loguearte con tu cuenta de Google, el resto se hace solo.

# Ejercicio 1

Cree un script que:

1. Transforme a parquet cualquiera de los siguientes datasets https://data.world/datasets/csv
2. Suba el archivo parquet al bucket de Cloud Storage.
3. Corra un script SQL en BigQuery para que BQ tome este archivo parquet y lo cargue en una tabla exitosamente.

# Ejercicio 2

Analice las diferencias entre subir un dataframe directo a una tabla en BigQuery, a subir un archivo de tipo parquet y hacer que BigQuery lo cargue desde Cloud Storage. Cuál le parece que deberías ser más rápido? Por qué? Hay beneficios en tener el archivo parquet en la nube? Cuáles?

**Deje sus respuestas a estas preguntas en el channel #foro de slack.**

# Ejercicio 3 (opcional):

Analizá, de la forma que vos quieras, la eficiencia que tienen los diferentes algoritmos de compresión bajo el parámetro **compression** de la función **to_parquet** de pandas. Podés hacer gráficos, una tabla o simplemente imprimir en consola tus conclusiones. A su vez, podés hacer un jupyter notebook contando un poco tu *línea de pensamiento*.

Investigá qué hace el algoritmo que encontraste ser el más eficiente y por qué le fue tan bien.

**Mostrá tus conclusiones al resto de la clase usando el channel #foro de slack!**