King's Super Math Saga experiment

We invite you to complete this practical assignment as part of your application for a Data Science role at King. Please send your solution back to us within one week of receipt. Candidates typically invest 3-6 hours across sittings to complete, but please spend as much or as little time as you feel is reasonable.

All completed assignments are reviewed by our Data Scientists, who will look for evidence of the skills detailed in the job description, particularly around your insight, communication, coding, SQL and statistical abilities. Please be ready to eventually discuss your assignment in an interview.

Problem statement

Super Math Saga is a fictional mobile game. Like other Saga games, players need to beat levels to progress through a map. At every level, players are faced with a math question they need to answer correctly to pass to the next level. The game is Free to Play meaning that players can download and play the game for free but can optionally buy hints on the levels.

In 2017 we ran an experiment (A/B Test) on the game, offering two different game experiences that we call A and B, **group A** being the control group where the experience is kept as is, and **group B** being the experiment group that is exposed to the new experience. We set the assignment process to randomly distribute players among the groups: 80% to group A (control) and 20% to group B (test). The experiment ran from 2017-05-04 to 2017-05-22.

Your assigment is to analyse and summarise the data provided, to determine which of the experiences makes Super Math Saga a better game. Send us a short (max 3 pages) report, suitable to present to the head of studio, with your findings and supporting plots, making recommendations for what the game should implement and any further analysis you think would be valuable. Markdown or notebook is the preferred report format.

Tools

We recommend to do the analysis in R or Python but you are free to use your preferred tools: what's important is to provide us with readable and reproducible code. You can access and query the data using one of the BigQuery libraries (Python, R, Other) with the provided project ID.

Two tables are provided in the **abtest** database in Google BigQuery:

The **assignment** table contains players assigned to the A/B test and attributes related to each player.

- playerid: Unique numeric identifier for each player
- abtest_group: The group the player was assigned to (A or B)
- assignment_date: The date when the player was assigned to the test
- install_date: The date when the player installed the game
- conversion_date: The date when the player made their first purchase

The **activity** table contains player activity for each day a player was active.

- playerid: Unique numeric identifier for each player
- activity_date: The date of activity
- purchases: Number of purchases made this day
- gameends: Number of gamerounds played this day

Instructions

1. Make sure you can access the provided Google Cloud Project. You can browse the data through the GUI by going to the *BigQuery* tab.
2. Set up the required BigQuery library above for your preferred language.
3. Do your thing! Based on your analysis, what do you recommend to the game team?
4. Send your report, code and plots back to us.

Suggested areas for analysis

- Which metrics do you consider important to look at in this case? What can you say about them in this A/B test?
- What sanity checks can you do to be confident in the validity of the test?
- Compute the average number of gamerounds per player for each group. Can you confidently state which group has higher engagement?
- Do different types of players react differently to the treatment?
- What type of change to the game do you think was tested here?

Notes

- Make sure to disable *Legacy SQL* when running queries, it is BigQuery's SQL dialect and can give different results from what you are expecting from standard SQL. This can be done in following ways:
  - **In queries**: Prefix your queries with `#standardSQL`
  - **Using libraries**: Set `use_legacy_sql=false` or similar flag when using the libraries
  - **Using BigQuery GUI**: Uncheck `Use Legacy SQL` under Options
- For python, if you want to connect to bigquery the setup we have found works best is:
  1. **Install Cloud SDK**: MacOS: https://cloud.google.com/sdk/docs/quickstart-macos Windows: https://cloud.google.com/sdk/docs/quickstart-windows Linux: https://cloud.google.com/sdk/docs/quickstart-linux
  2. Run `gcloud init` and use the gmail address you provided to us
  3. Run `gcloud auth application-default login`
  4. **Install python library**: https://pypi.python.org/pypi/google-cloud-bigquery
- For R we recommend **installing bigrquery library**: https://github.com/r-dbi/bigrquery
- Please do not try to use service accounts as you will get the following error: `"Access Denied: Project {project}: The user xxxxxx@{project}.iam.gserviceaccount.com does not have bigquery.jobs.create permission in project {project}"`.
- If you happen to encounter problems with the libraries, it is OK to run your queries through the BigQuery GUI, save the results as csv and import it in your code. Just remember to include the underlying queries in your code for reproduceability.

For any arising questions or issues, please contact datasciencetechtest@king.com.