BigQuery Basics

Agenda

- **Overview**
  - **Why do you need a dataware warehouse?**
  - **Why BigQuery?**
- BigQuery organization
- Accessing BigQuery
- Google Analytics Export
- How to Query data
  - Query Samples
- Resources

Why Do You Need a Data Warehouse?

- A data warehouse is the most valuable asset of your BI team
- How it works:
  - Data are extracted on a periodic basis from source systems and moved to a dedicated server that contains the data warehouse
  - During this process, the data are cleaned, formatted, validated, reorganized, summarized, and integrated with other sources
- A data warehouse delivers value to companies through:
  - The generation of scheduled reports
  - Packaged analytical solutions
  - Adhoc reporting and analysis
  - Dynamic visualization
  - Storage of historical data
  - Data mining

Choosing a Data Warehouse

- There are many factors to consider when choosing a data warehouse:
  - Assets: generation of big data reports requires expensive servers
  - People: skilled database administrators are needed to manage data integrity
  - Cost: interacting with big data can be expensive, slow, and inefficient
  - Scale: how much storage is needed and will storage needs change over time?
  - Security: how is data protected to ensure availability and durability?

What is BigQuery?

- BigQuery is a service provided by Google Cloud Platform, a suite of products & services that includes application hosting, cloud computing, database services, etc on on Google's scalable infrastructure
- BigQuery is Google’s fully managed solution for companies who need a fully-managed and cloud based interactive query service for massive datasets

Why BigQuery?

- Service for interactive analysis of massive datasets (TBs)
  - Query billions of rows: seconds to write, seconds to return
  - Uses a SQL-style query syntax
  - It's a service, can be accessed by a API
- Reliable and Secure
  - Replicated across multiple sites
  - Secured through Access Control Lists
- Scalable
  - Store hundreds of terabytes
  - Pay only for what you use
- Fast (really)
  - Run ad hoc queries on multi-terabyte data sets in seconds

Agenda

- Overview
  - Why do you need a dataware warehouse?
  - Why BigQuery?
- **BigQuery organization**
- Accessing BigQuery
- Google Analytics Export
- How to Query data
  - Query Samples
- Resources

BigQuery Organization

- BigQuery is structured as a hierarchy with 4 levels:
  - Projects: Top-level containers in the Google Cloud Platform that store the data
  - Datasets: Within projects, datasets hold one or more tables of data
  - Tables: Within datasets, tables are row-column structures that hold actual data
  - Jobs: The tasks you are performing on the data, such as running queries, loading data, and exporting data

Projects

- Projects are the top-level containers that store the data
- Within the project, you can configure settings, permissions, and other metadata that describe your applications
- Each project has a name, ID, and number that you'll use as identifiers
- When billing is enabled, each project is associated with one billing account but multiple projects can be billed to the same account
  - This link provides more information on pricing options for BigQuery

Datasets

- Datasets allow you to organize and control access to your tables
- All tables must belong to a dataset. You must create a dataset before loading data into BigQuery
- You can configure permissions at the organization, project, and dataset level
  - See this link for more information on access control

Tables

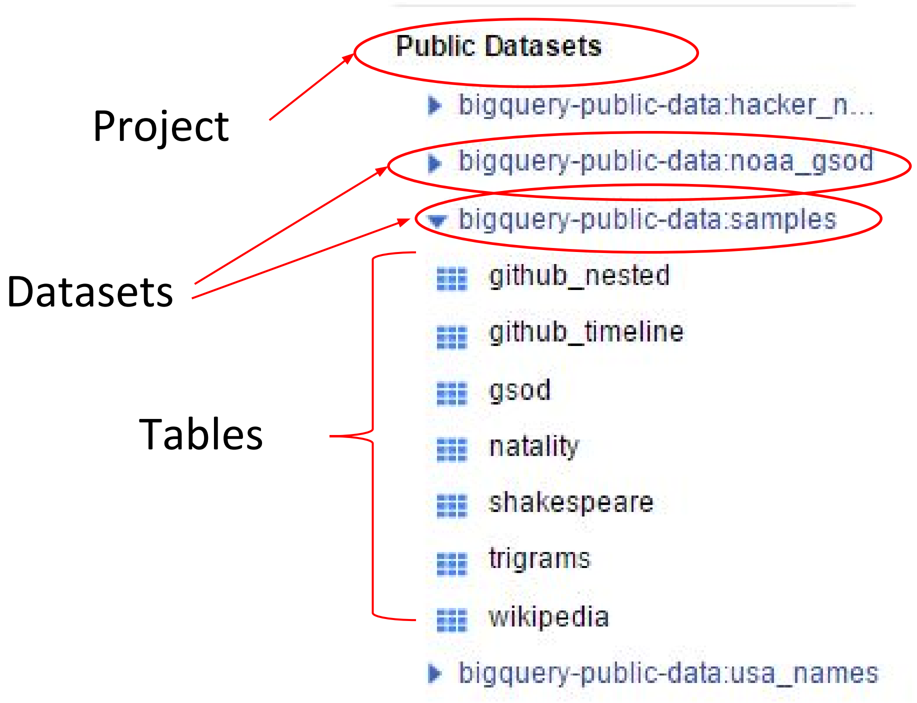
- Tables contain your data in BigQuery
- Each table has a schema that describes the data contained in the table, including field names, types, and descriptions
- BigQuery supports the following table types:
  - Native tables: tables backed by native BigQuery storage
  - External tables: tables backed by storage external to BigQuery
  - Views: virtual tables defined by a SQL query

Jobs

- Jobs are objects that manage asynchronous tasks such as running queries, loading data, and exporting data
    - You can run multiple jobs concurrently
    - Completed jobs are listed in the Jobs collection
- There are four types of jobs:
    - Load: load data into a table
    - Query: run a query against BigQuery data
    - Extract: export a BigQuery table to Google Cloud Storage
    - Copy: copy an existing table into another new or existing table

Example: BigQuery, Datasets, and Tables

- Here is an example of the left-pane navigation within BigQuery
- Projects are identified by the project name, e.g. Public Datasets, and ID, e.g. bigquery-public-data
- You can expand projects to see the corresponding datasets, e.g. samples, and tables, e.g. github_nested
- Tables are referenced by their project and dataset as: <project>:<dataset>.<table>
  - e.g. bigquery-public-data:samples.natality



Project

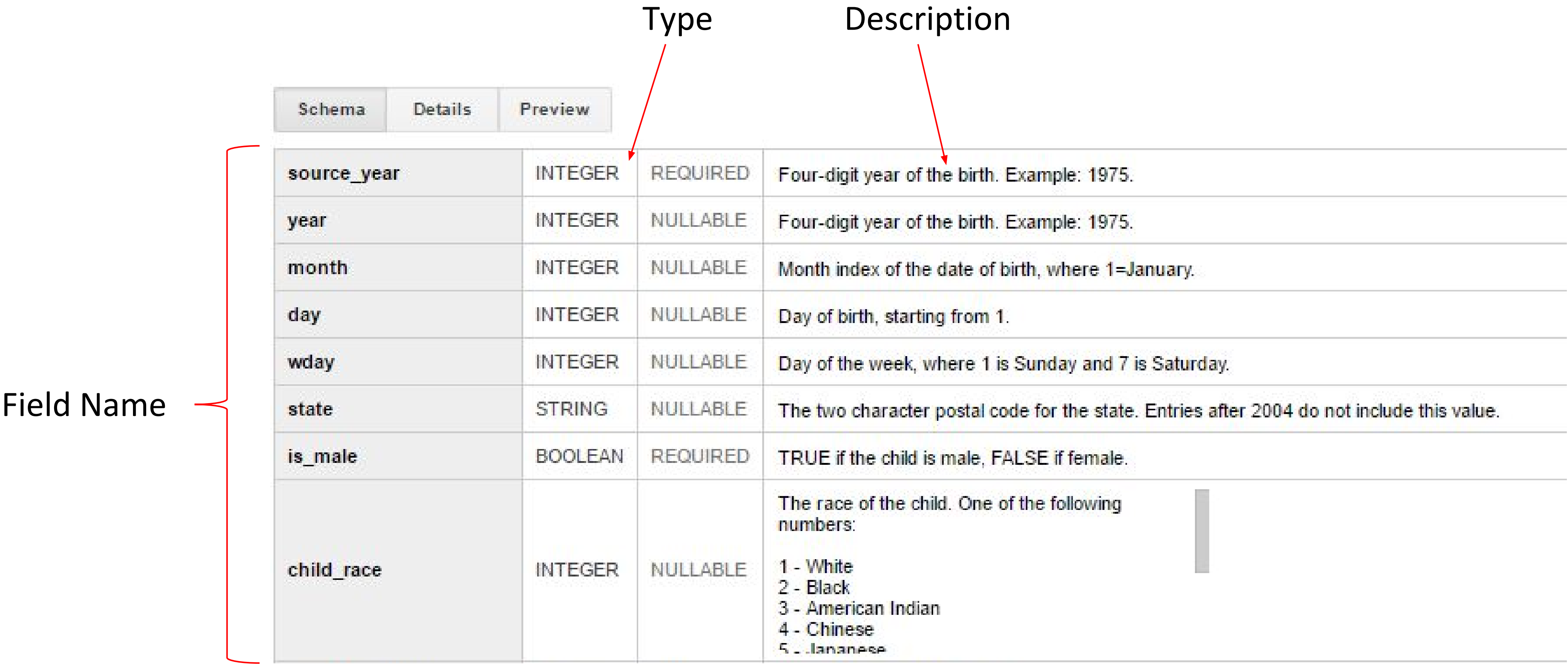
Datasets

Tables

Public Datasets
bigquery-public-data:hacker_n...
bigquery-public-data:noaa_gsod
bigquery-public-data:samples
github_nested
github_timeline
gsod
natality
shakespeare
trigrams
wikipedia
bigquery-public-data:usa_names

Example of Simple Schema

Schema for table Natality under Sample Datasets



source_year	INTEGER	REQUIRED	Four-digit year of the birth. Example: 1975.
year	INTEGER	NULLABLE	Four-digit year of the birth. Example: 1975.
month	INTEGER	NULLABLE	Month index of the date of birth, where 1=January.
day	INTEGER	NULLABLE	Day of birth, starting from 1.
wday	INTEGER	NULLABLE	Day of the week, where 1 is Sunday and 7 is Saturday.
state	STRING	NULLABLE	The two character postal code for the state. Entries after 2004 do not include this value.
is_male	BOOLEAN	REQUIRED	TRUE if the child is male, FALSE if female.
child_race	INTEGER	NULLABLE	The race of the child. One of the following numbers: 1 - White 2 - Black 3 - American Indian 4 - Chinese 5 - Japanese

Agenda

- Overview
  - Why do you need a dataware warehouse?
  - Why BigQuery?
- BigQuery organization
- **Accessing BigQuery**
- Google Analytics Export
- How to Query data
  - Query Samples
- Resources

Accessing BigQuery

- You can access BigQuery and run jobs from your web browser
- Developers can use bq command line tool
  - python-based tool that can access BigQuery from the command line
- Developers can also leverage the Service API
  - RESTful API to access BigQuery programmatically
  - Requires authorization by OAuth2
  - Google client libraries for Python, JavaScript, PHP, etc.
- Integration Possible with Third party Tools
  - Visualization and Statistical Tools tools like Tableau, QlikView, R, etc.
- You can export data in a .csv file, jason or to Google Cloud Storage

Agenda

- Overview
    - Why do you need a dataware warehouse?
    - Why BigQuery?
- BigQuery organization
- Accessing BigQuery
- **Google Analytics Export**
- How to Query data
    - Query Samples
- Resources

Google Analytics Export

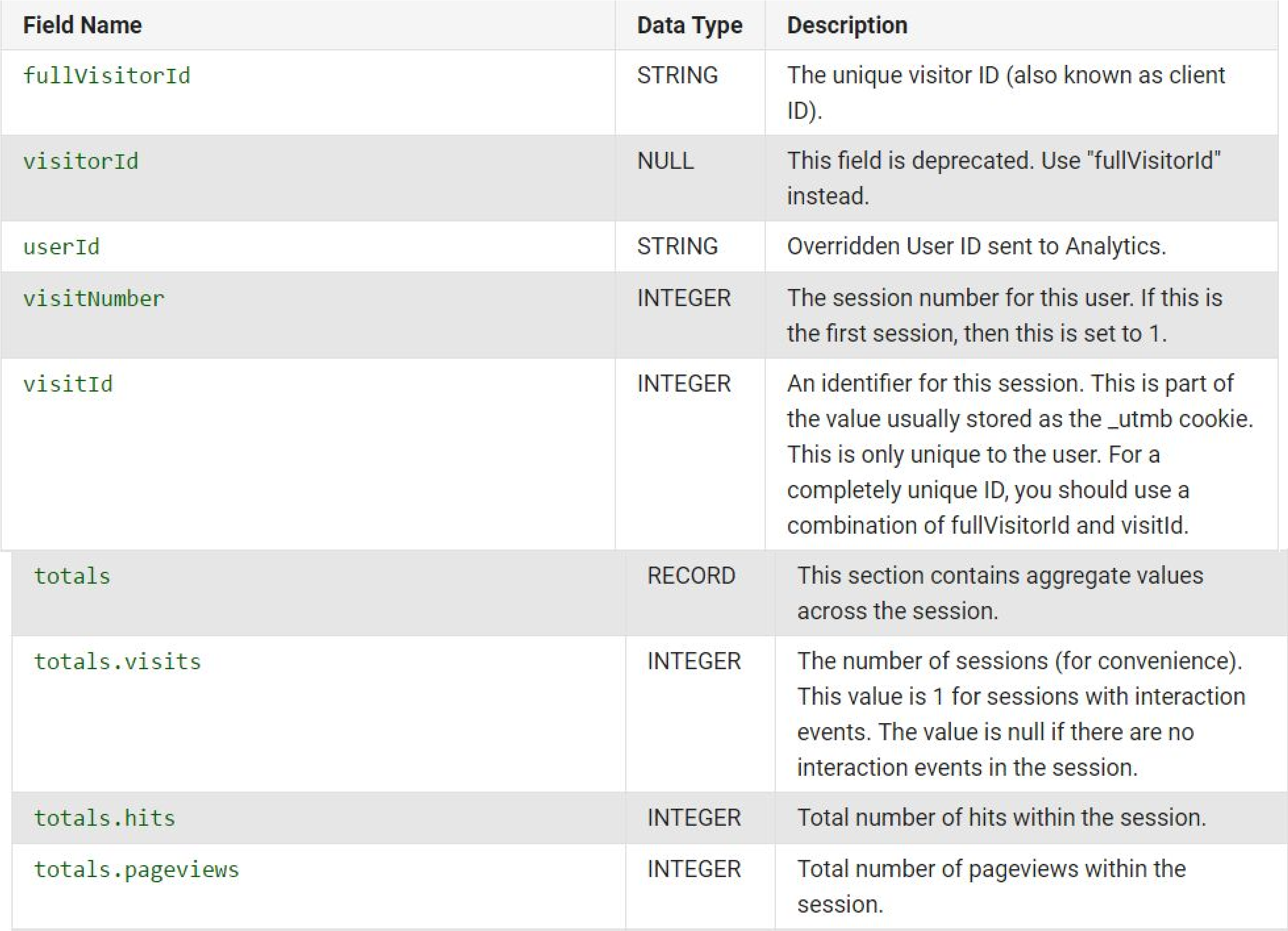
- *This feature is only available to Google Analytics Premium accounts.*
- You can export session and hit data from a Google Analytics account to BigQuery
  - Use SQL-like syntax to query
  - Unsampled, detailed Analytics logs automatically imported to BigQuery
- When data is exported to BigQuery, you own that data and you can use BigQuery Access Control Lists (ACLs) to manage permissions on projects and datasets
- Ability to integrate with data in multiple datasources
- Your Google Analytics 360 Account Manager will give you a monthly credit of $500 USD towards usage of BigQuery for this project

Google Analytics BigQuery Export Schema

- Datasets: For each Analytics view that is enabled for BigQuery integration, a dataset is added using the view ID as the name.
- Tables: Within each dataset, a table is imported for each day of export. These tables have the format "ga_sessions_YYYYMMDD".
- Rows: Each row within a table corresponds to a session in Google Analytics.
- Columns: Each column contains a value or set of nested values
  - Find the full list of columns by following the link here

Google Analytics BigQuery Export Schema

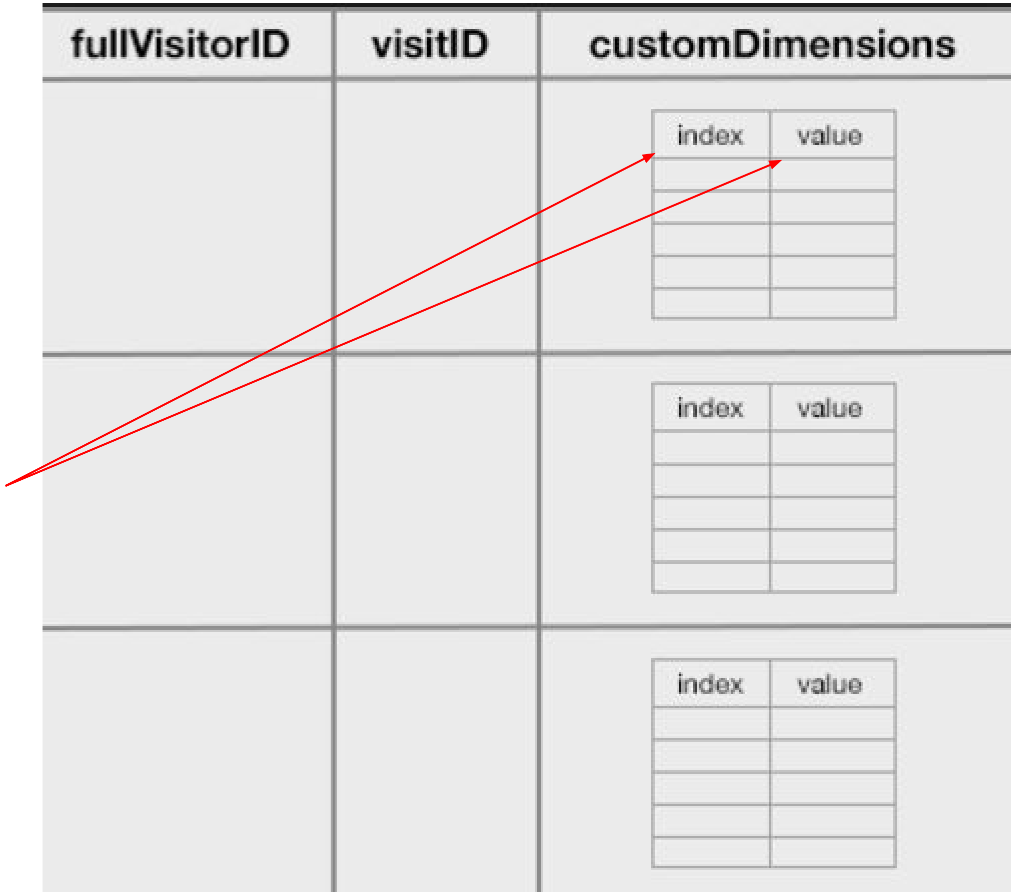
- Below is a subset of columns from the schema
- Many of the columns will be familiar to Google Analytics users, such as user ID, visits (sessions), hits, and pageviews
- For the full list, see this link



Field Name	Data Type	Description
`fullVisitorId`	STRING	The unique visitor ID (also known as client ID).
`visitorId`	NULL	This field is deprecated. Use "fullVisitorId" instead.
`userId`	STRING	Overridden User ID sent to Analytics.
`visitNumber`	INTEGER	The session number for this user. If this is the first session, then this is set to 1.
`visitId`	INTEGER	An identifier for this session. This is part of the value usually stored as the _utmb cookie. This is only unique to the user. For a completely unique ID, you should use a combination of fullVisitorId and visitId.
`totals`	RECORD	This section contains aggregate values across the session.
`totals.visits`	INTEGER	The number of sessions (for convenience). This value is 1 for sessions with interaction events. The value is null if there are no interaction events in the session.
`totals.hits`	INTEGER	Total number of hits within the session.
`totals.pageviews`	INTEGER	Total number of pageviews within the session.

Google Analytics BigQuery Export Schema

- Some columns within the export have nested fields
- Nested fields are referenced by using a period (.)
    - For example, within the customDimensions field, there are two nested fields, customDimensions.index and customDimensions.value



Google Analytics BigQuery Export Schema

- The schema also includes many useful columns that are not accessible within the GA user interface or Core Reporting API
- Some of these additional columns include:
  - fullVisitorId, the anonymous identifier used by the GA cookie
  - visitId, an identifier for the session
  - hits.hitNumber, the sequenced hit number
- Access to these three variable allows for deeper analysis at the user, session, and hit level

Agenda

- Overview
  - Why do you need a dataware warehouse?
  - Why BigQuery?
- BigQuery organization
- Accessing BigQuery
- Google Analytics Export
- **How to Query data**
  - **Query Samples**
- Resources

How to Query Data?

- BigQuery uses a SQL-like language for querying and manipulating data
- SQL statements are used to perform various database tasks, such as querying data, creating tables, and updating databases
    - For today, we’ll focus on SQL statements for querying data. These statements use the SELECT command
- Queried data is presented in a table called the result set

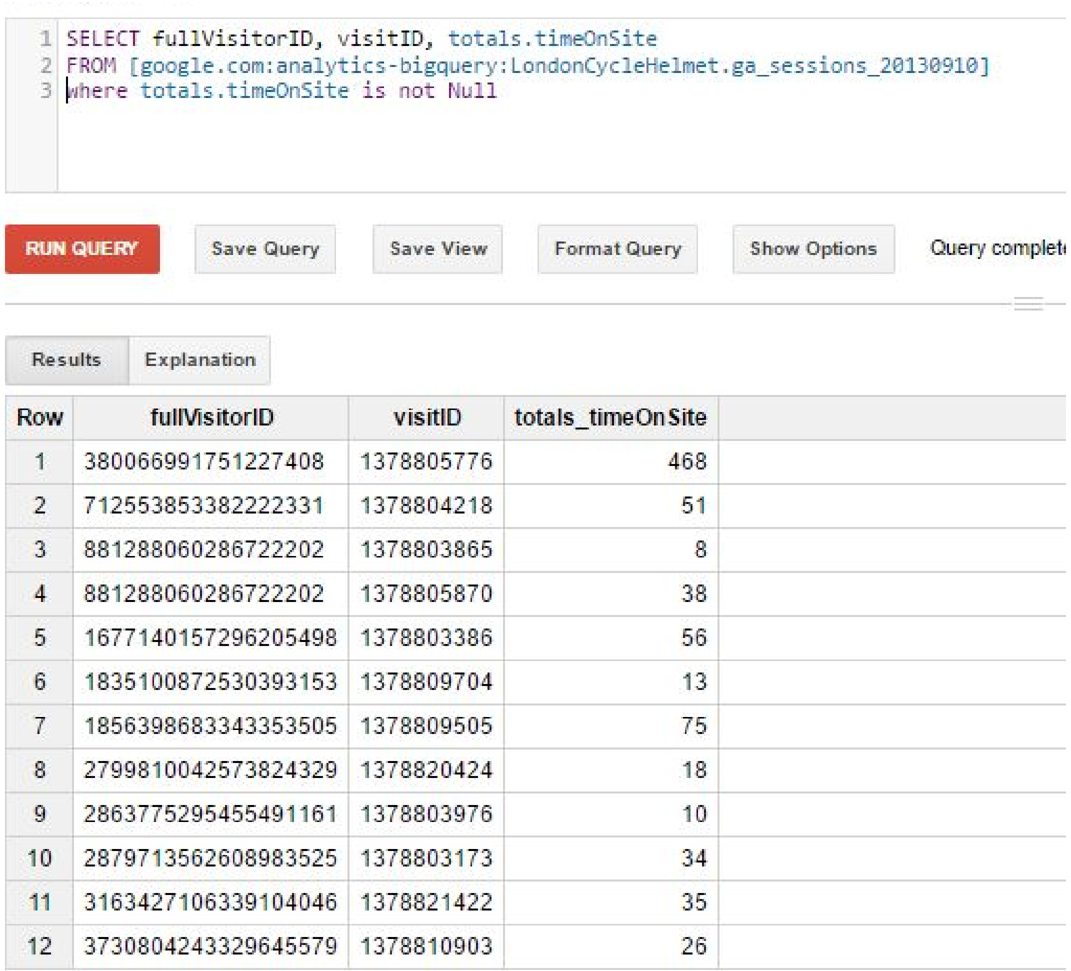
How to Query Data?

- Basic queries contain the following components:
  - SELECT (required): identifies the columns to be included in the query
  - FROM (required): the table that contains the columns in the SELECT statement
  - WHERE: a condition for filtering records
  - ORDER BY: how to sort the result set
  - GROUP BY: how to aggregate data in the result set
- Example query:

```
SELECT year, state, is_male, gestation_weeks
FROM [bigquery-public-data:samples.natality]
```

Query Sample :

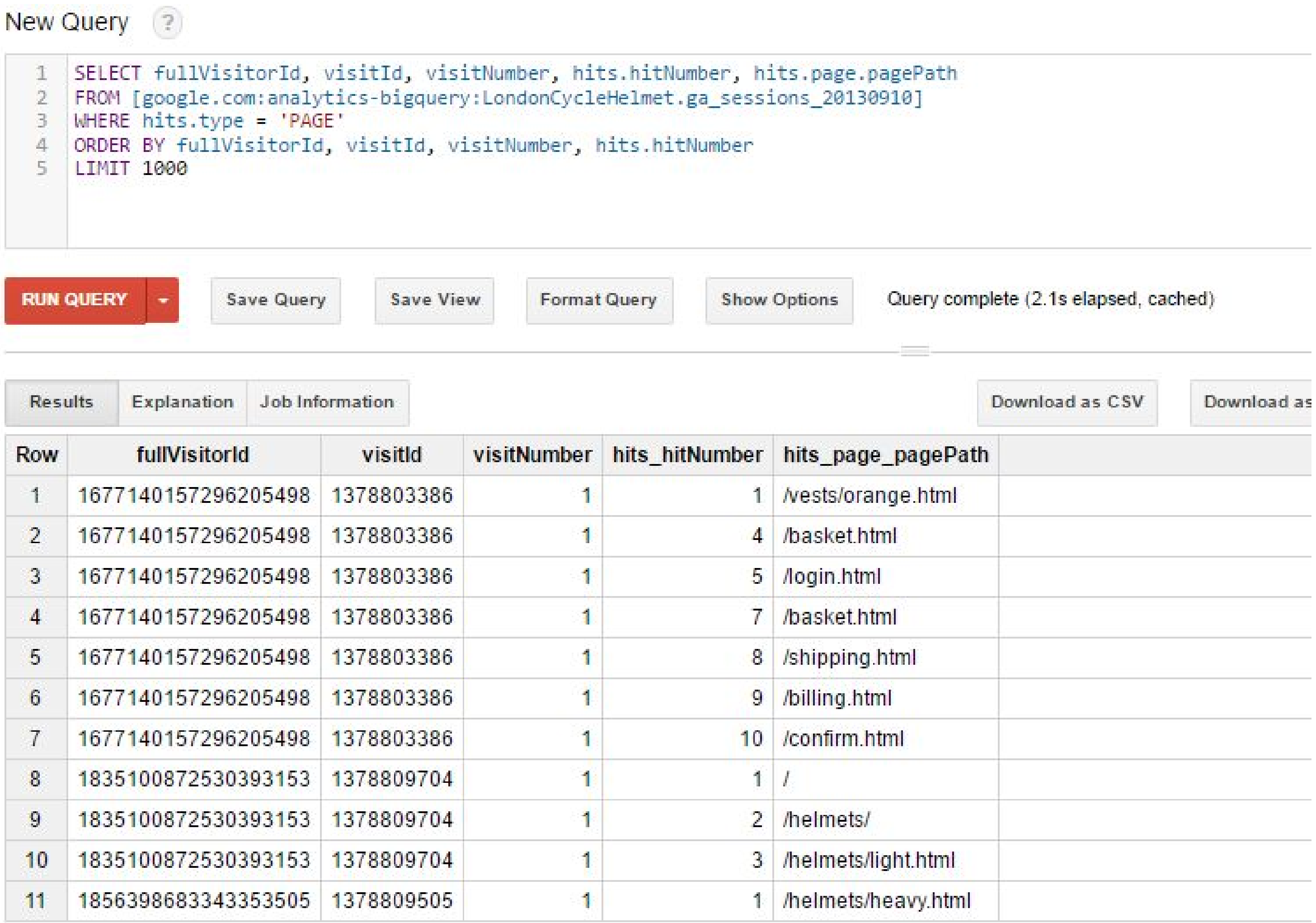
Time Spent Per session per user



```
SELECT fullVisitorID, visitID, totals.timeOnSite
FROM [google.com:analytics-bigquery:LondonCycleHelmet.ga_sessions_20130910]
where totals.timeOnSite is not Null
```

Row	fullVisitorID	visitID	totals_timeOnSite
1	380066991751227408	1378805776	468
2	712553853382222331	1378804218	51
3	881288060286722202	1378803865	8
4	881288060286722202	1378805870	38
5	1677140157296205498	1378803386	56
6	1835100872530393153	1378809704	13
7	1856398683343353505	1378809505	75
8	2799810042573824329	1378820424	18
9	2863775295455491161	1378803976	10
10	2879713562608983525	1378803173	34
11	3163427106339104046	1378821422	35
12	3730804243329645579	1378810903	26

Query Sample :
Sequence of Pages Viewed by User



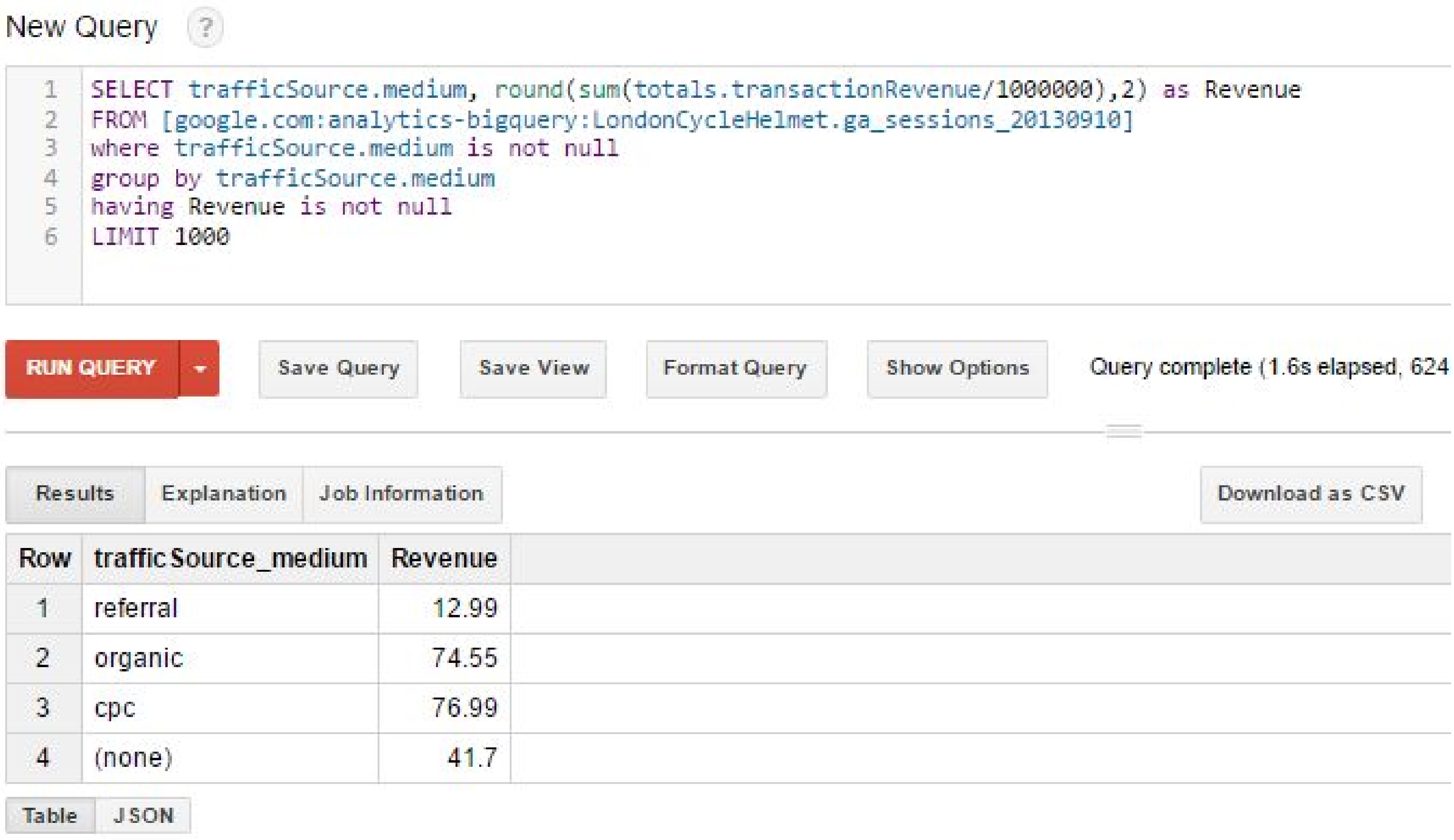
```
SELECT fullVisitorId, visitId, visitNumber, hits.hitNumber, hits.page.pagePath
FROM [google.com:analytics-bigquery:LondonCycleHelmet.ga_sessions_20130910]
WHERE hits.type = 'PAGE'
ORDER BY fullVisitorId, visitId, visitNumber, hits.hitNumber
LIMIT 1000
```

Query complete (2.1s elapsed, cached)

Row	fullVisitorId	visitId	visitNumber	hits_hitNumber	hits_page_pagePath
1	1677140157296205498	1378803386	1	1	/vests/orange.html
2	1677140157296205498	1378803386	1	4	/basket.html
3	1677140157296205498	1378803386	1	5	/login.html
4	1677140157296205498	1378803386	1	7	/basket.html
5	1677140157296205498	1378803386	1	8	/shipping.html
6	1677140157296205498	1378803386	1	9	/billing.html
7	1677140157296205498	1378803386	1	10	/confirm.html
8	1835100872530393153	1378809704	1	1	/
9	1835100872530393153	1378809704	1	2	/helmets/
10	1835100872530393153	1378809704	1	3	/helmets/light.html
11	1856398683343353505	1378809505	1	1	/helmets/heavy.html

Query Sample :

Revenue per medium



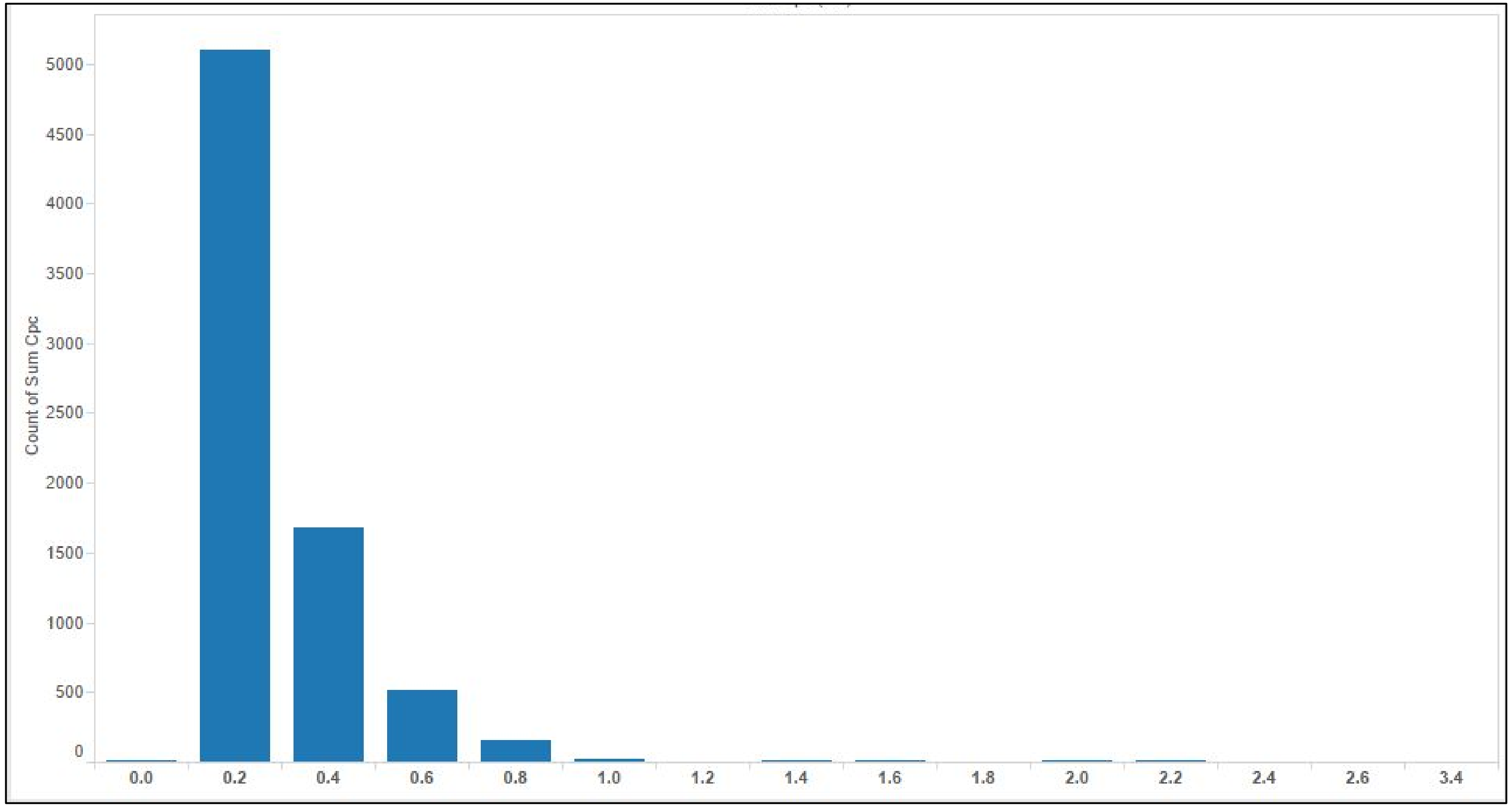
```
SELECT trafficSource.medium, round(sum(totals.transactionRevenue/1000000),2) as Revenue
FROM [google.com:analytics-bigquery:LondonCycleHelmet.ga_sessions_20130910]
where trafficSource.medium is not null
group by trafficSource.medium
having Revenue is not null
LIMIT 1000
```

Query complete (1.6s elapsed, 624

Row	trafficSource_medium	Revenue
1	referral	12.99
2	organic	74.55
3	cpc	76.99
4	(none)	41.7

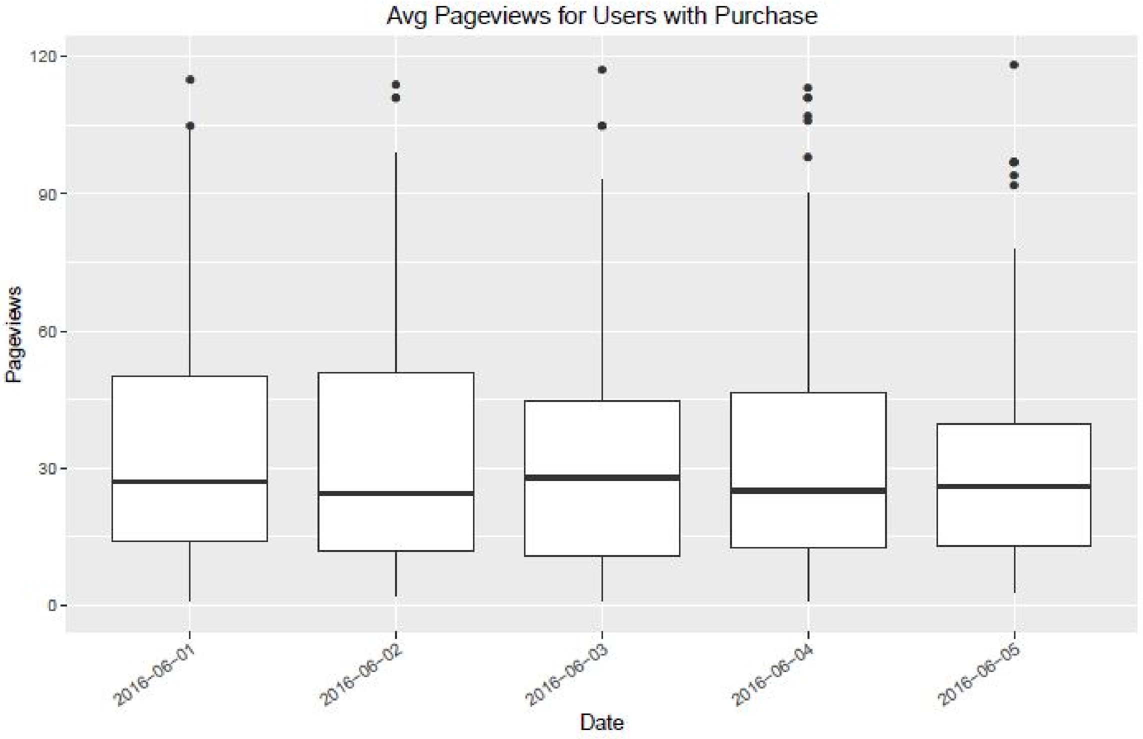
Visualization Example - Tableau

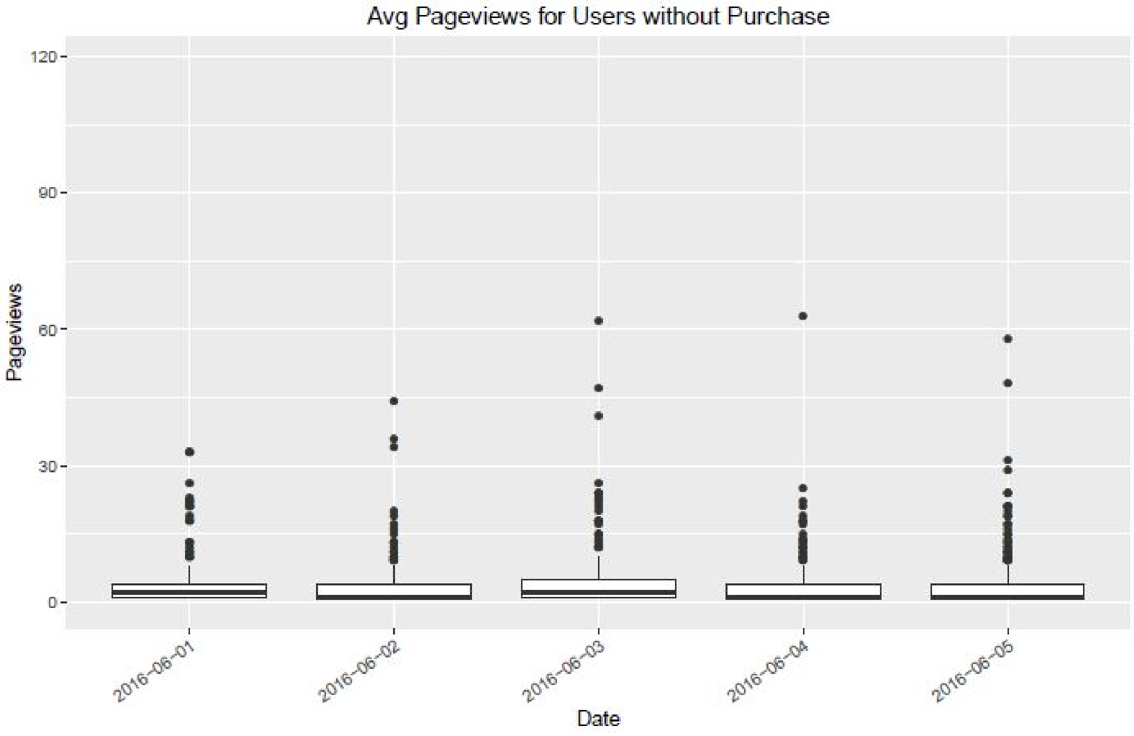
This bar chart shows the distribution of cpc costs per user



Visualization Example - R

These boxplots show the difference in the number of pageviews for sessions with and without purchases





Agenda

- Overview
  - Why do you need a dataware warehouse?
  - Why BigQuery?
- BigQuery organization
- Accessing BigQuery
- Google Analytics Export
- How to Query data
  - Query Samples
- **Resources**

Resources

- Google BigQuery Documentation
- Google Analytics Premium + Google BigQuery for Predictive Digital Marketing
- SQL tutorial

Thank You!