LESSON 16

DATABASES & SQL

Michael Twardos

What We Will Do Today

- Learn about databases and data warehouse design.
- Introduction to SQL and learn the Fundamental Growth Query.
- Look at product engagement data of a fictional company and use FGQ to compute retention curves.
- Apply convolution to the retention curve to project future active users.
- Build a model to predict the retention likelihood of individual customers.
- Think about how can a data science model **actually** be used.

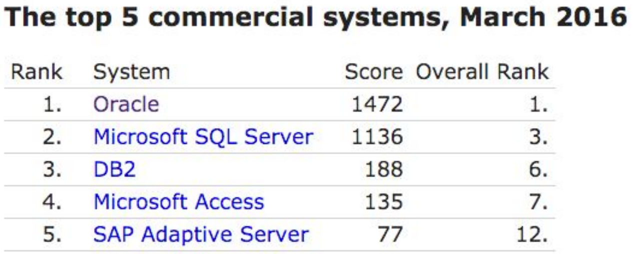
WHAT IS A DATABASE?

Why Databases?

- Databases are used as a repository of information. Allow for efficient storage and access.
- Types
    - Relational (MySQL, PostgreSQL, Redshift)
    - Key Value (Redis)
    - Document (Mongo)
    - Graph (Neo4j)
    - Time Series (Graphite)
    - Search (Elastic, Solr, Splunk)
    - Wide Column (Cassandra, HBase)

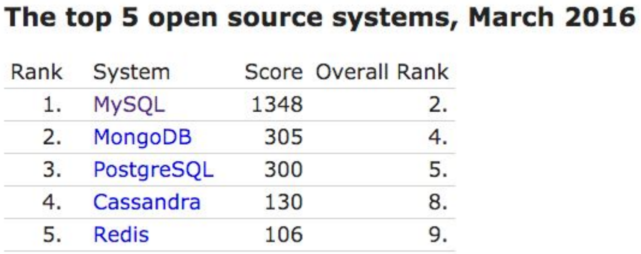
[redacted]

- DB Popularity



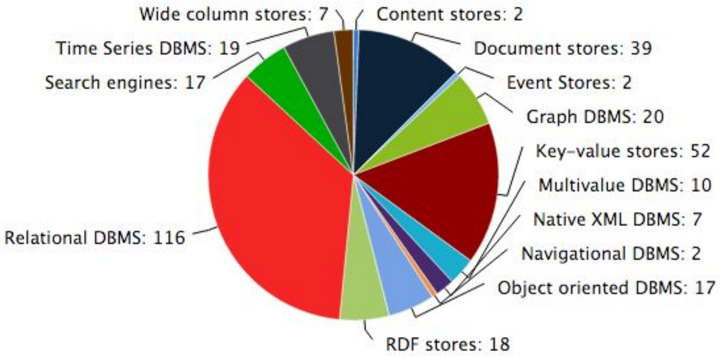
The top 5 commercial systems, March 2016

Rank	System	Score	Overall Rank
1.	Oracle	1472	1.
2.	Microsoft SQL Server	1136	3.
3.	DB2	188	6.
4.	Microsoft Access	135	7.
5.	SAP Adaptive Server	77	12.



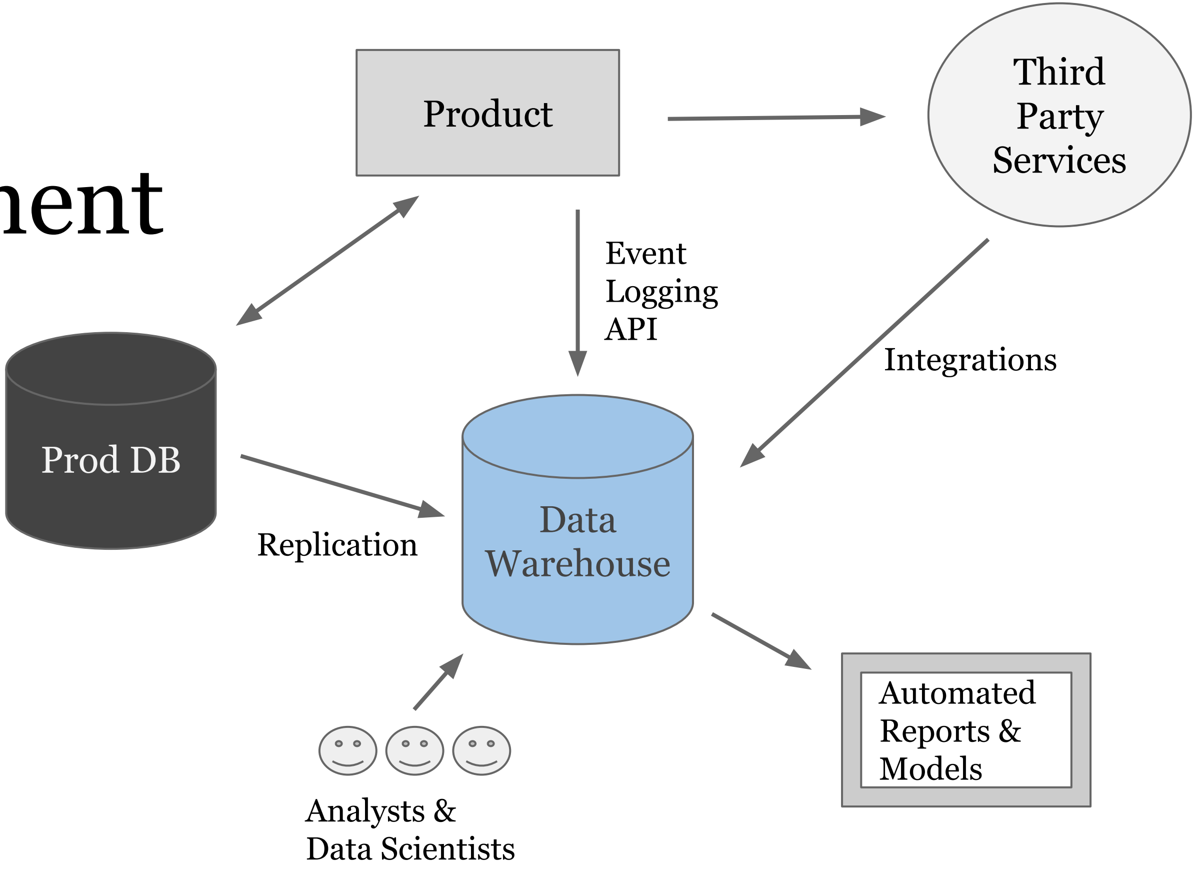
The top 5 open source systems, March 2016

Rank	System	Score	Overall Rank
1.	MySQL	1348	2.
2.	MongoDB	305	4.
3.	PostgreSQL	300	5.
4.	Cassandra	130	8.
5.	Redis	106	9.

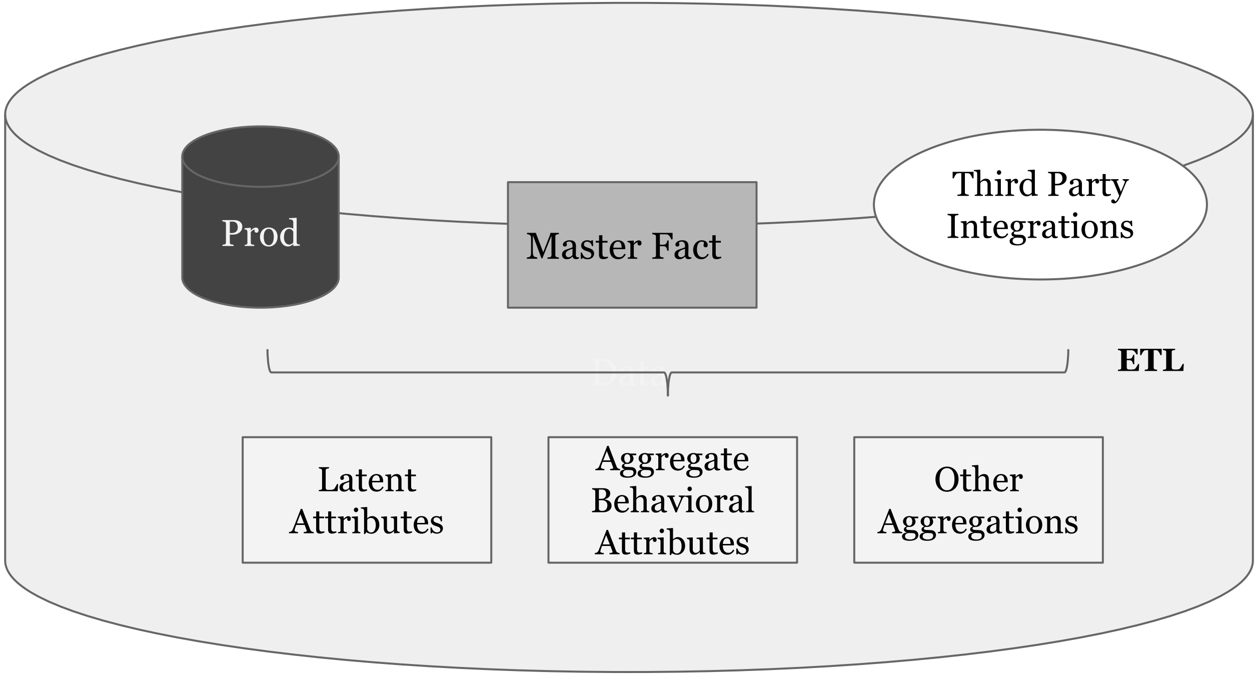


© 2016, DB-Engines.com

Data Management



Data Warehouse Design



Master Fact Table

- A comprehensive historical record of everything that happened. (Think who, what, where, when, how, why)
- Required fields:

entity_id, time_stamp, event_type

- Recommended fields

marketing campaign, location, device, browser, os…

Stories from Industry

What have you experienced and what was good or bad?

How was the data stored? How is it accessed?

Group Exercise: Choose One

1) How would you design Netflix's database? How would you use it to run their recommendation engine?

2) How would you store data to generate Facebook's Newsfeed?

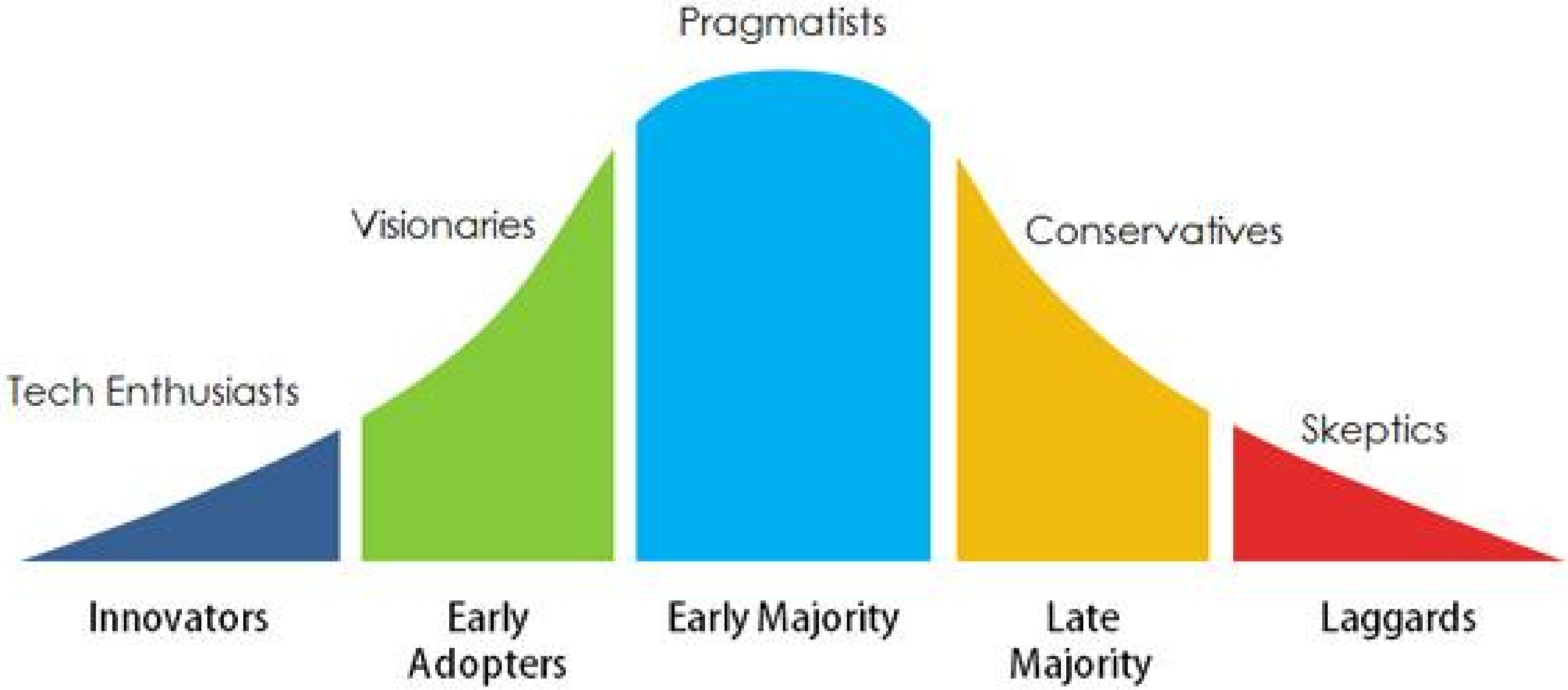
SQL Read

```
SELECT      a
FROM        b
WHERE       c
GROUP BY    d
ORDER BY    e
LIMIT       f
;
```

Python SQL Workshop

- Import data into SQLite
- Run basic SQL queries
- Explore SQL functions

Product Adoption Curve



CONSUMPTION SPREADS FASTER TODAY

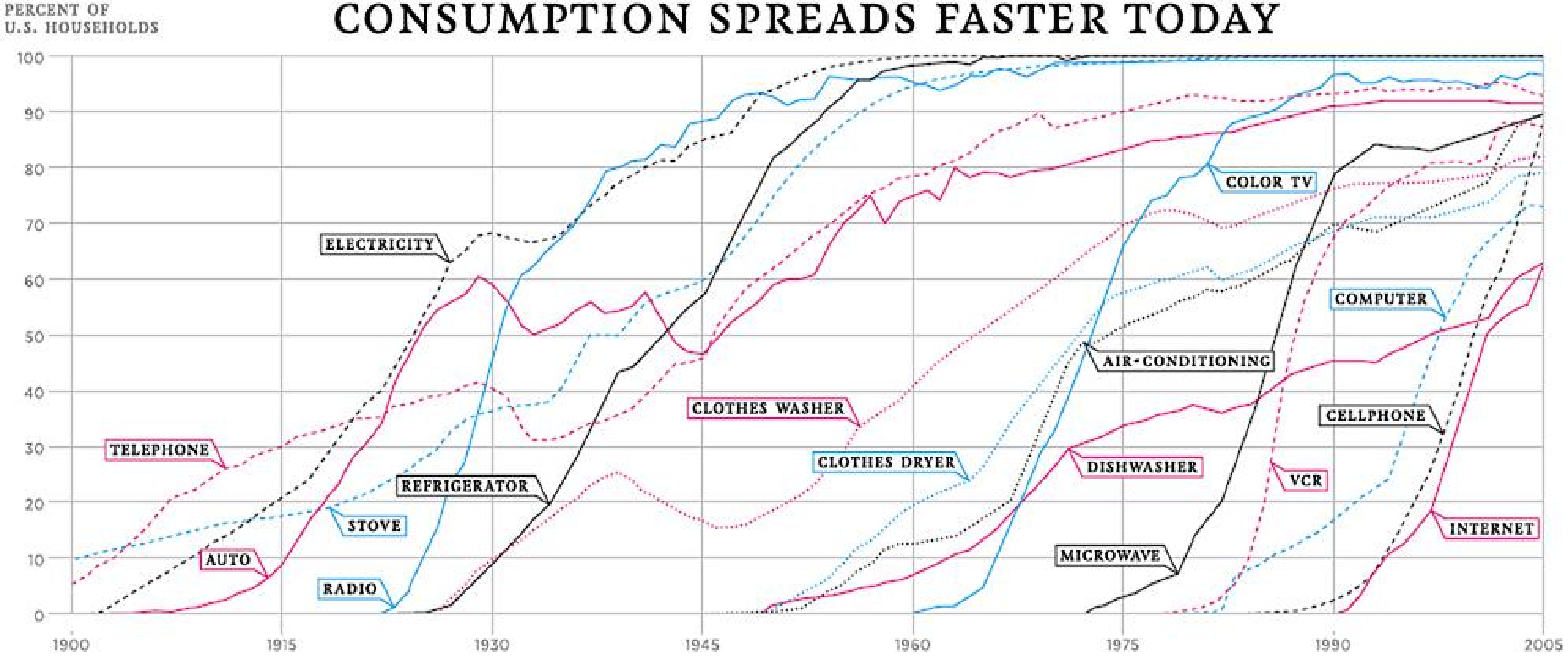
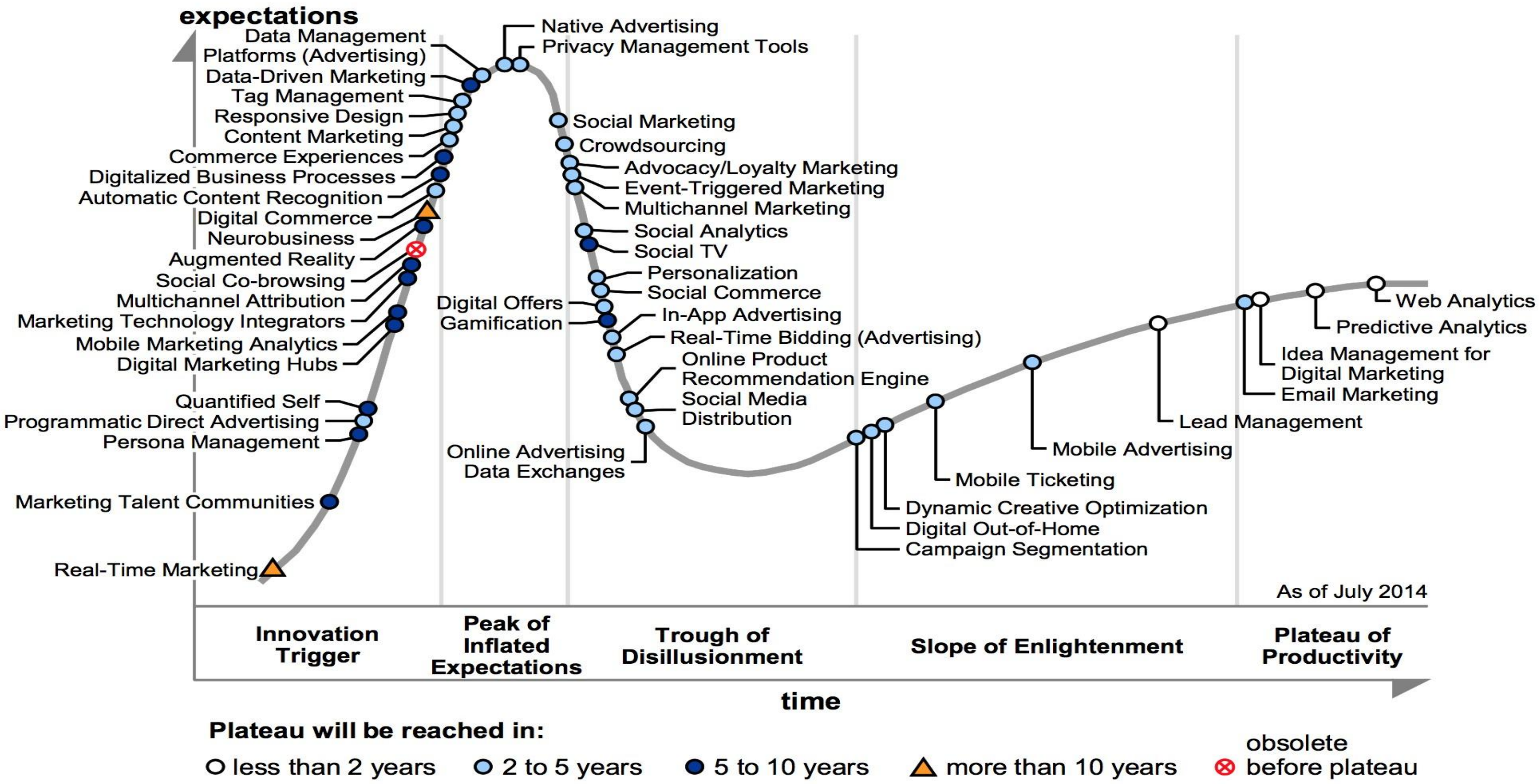


Figure 1. Hype Cycle for Digital Marketing, 2014



Source: Gartner (July 2014)

Stories of Product Adoption

Task: Think of a product / service / company you started using/interacting with in the last year and answer these questions.

1) How did you hear about it?

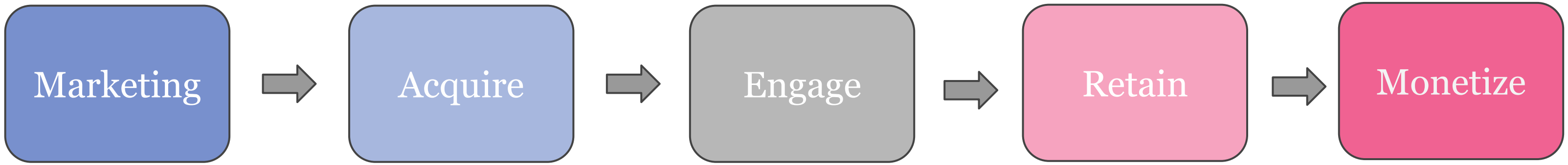
2) What was your first experiences with it? Was it memorable?

3) How easy was the product to use? How valuable? How enjoyable?

4) How often do you use this product months after you first started to if at all?

5) Did you pay? Would you pay again?

A Customer's Lifecycle



Marketing → Acquire → Engage → Retain → Monetize

- Cost
- Impressions
- Clicks

- Impress
- Submit
- Convert
- Onboard

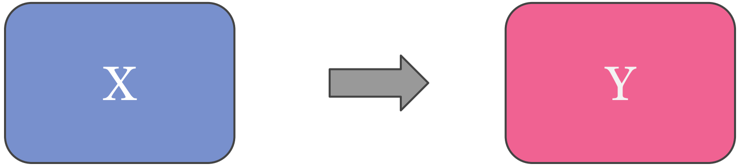
- Feature Use
- Product Adoption

- Logins
- Feature Engagement

- Payments
- Churn

- Events are captured through a customer's life.
- The period of time between events can define a user 'state'. A state is the atomic unit of measurement for growth.

Customer Transitions



$$Y(t) = XG(t)$$

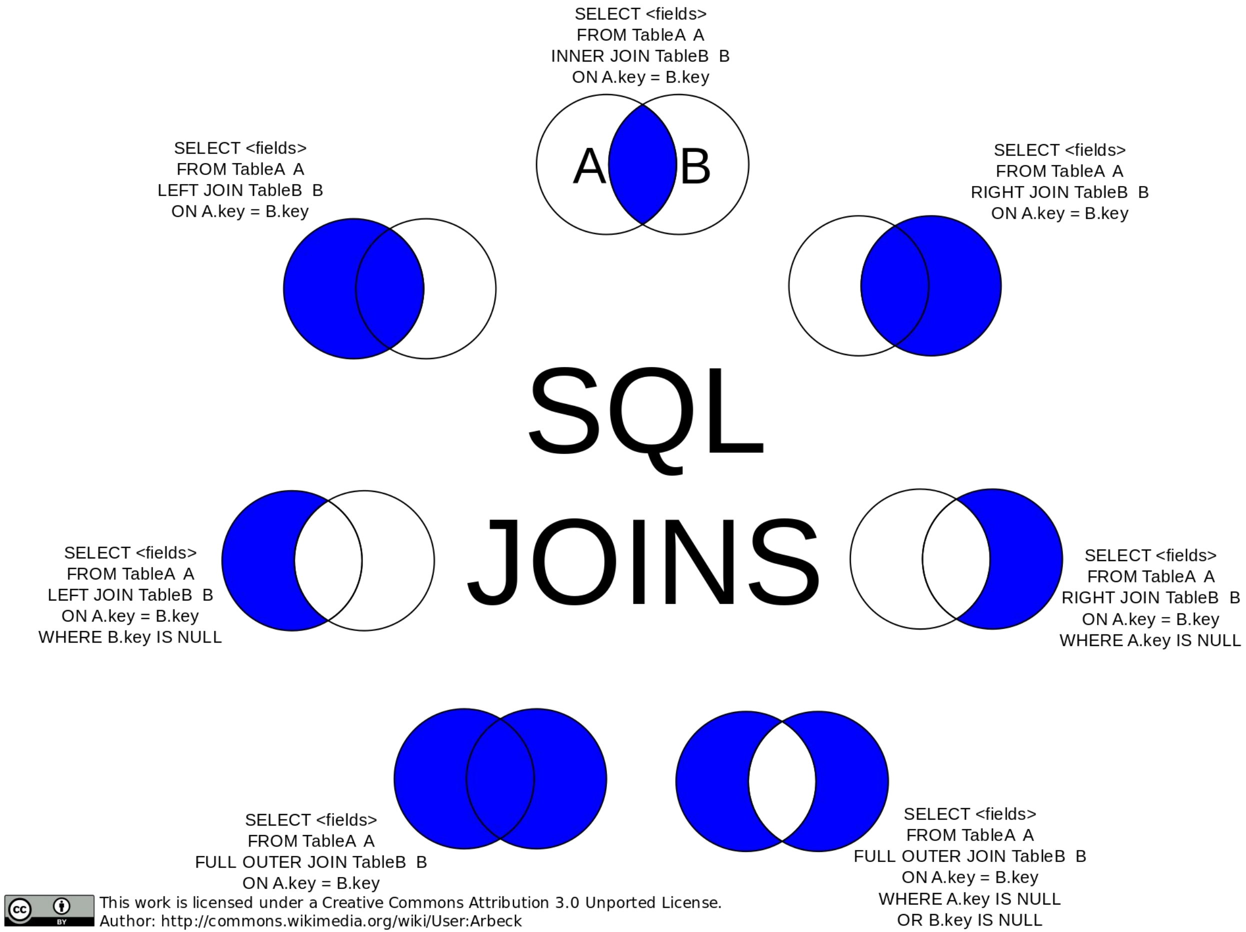
The time dependent 'susceptibility' of Y to X.

- X and Y are behaviors captured as events.
- We measure the transitions of customers as defined by these events.
- Y's relationship to X, can be dependent or independent, singular or multiple. Y can also be the same event as X.
- Measuring G(t) tells us about the rate of change or growth between X and Y.

Join Example

```
SELECT Count(DISTINCT( b.customer_id ))
FROM   master_fact x
       JOIN master_fact y
         ON x.customer_id = y.customer_id
WHERE  x.event_type = 'X'
       AND y.event_type = 'Y'
```

JOINS



```
SELECT <fields>
FROM TableA  A
INNER JOIN TableB  B
ON A.key = B.key
```

```
SELECT <fields>
FROM TableA  A
LEFT JOIN TableB  B
ON A.key = B.key
```

```
SELECT <fields>
FROM TableA  A
RIGHT JOIN TableB  B
ON A.key = B.key
```

```
SELECT <fields>
FROM TableA  A
LEFT JOIN TableB  B
ON A.key = B.key
WHERE B.key IS NULL
```

```
SELECT <fields>
FROM TableA  A
RIGHT JOIN TableB  B
ON A.key = B.key
WHERE A.key IS NULL
```

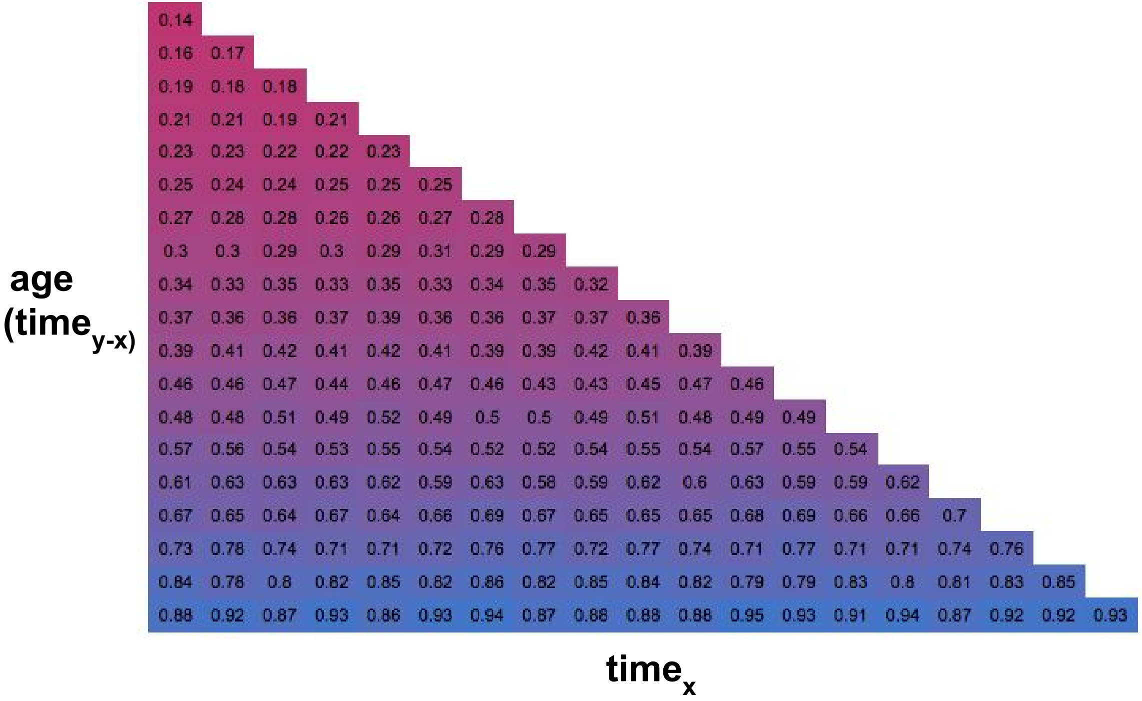
```
SELECT <fields>
FROM TableA  A
FULL OUTER JOIN TableB  B
ON A.key = B.key
```

```
SELECT <fields>
FROM TableA  A
FULL OUTER JOIN TableB  B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL
```

The Fundamental Growth Query

```
SELECT b.date_created :: DATE - a.date_created ::
DATE AS "Age",
       Count(DISTINCT( b.customerid ))
AS customers
FROM   master_fact a
       left join master_fact b
              ON a.customer_id = b.customer_id
WHERE  a.eventtype = 'X'
       AND b.eventtype = 'Y'
GROUP  BY 1
```

Time Dependant Susceptibility



0.14																		
0.16	0.17																	
0.19	0.18	0.18																
0.21	0.21	0.19	0.21															
0.23	0.23	0.22	0.22	0.23														
0.25	0.24	0.24	0.25	0.25	0.25													
0.27	0.28	0.28	0.26	0.26	0.27	0.28												
0.3	0.3	0.29	0.3	0.29	0.31	0.29	0.29											
0.34	0.33	0.35	0.33	0.35	0.33	0.34	0.35	0.32										
0.37	0.36	0.36	0.37	0.39	0.36	0.36	0.37	0.37	0.36									
0.39	0.41	0.42	0.41	0.42	0.41	0.39	0.39	0.42	0.41	0.39								
0.46	0.46	0.47	0.44	0.46	0.47	0.46	0.43	0.43	0.45	0.47	0.46							
0.48	0.48	0.51	0.49	0.52	0.49	0.5	0.5	0.49	0.51	0.48	0.49	0.49						
0.57	0.56	0.54	0.53	0.55	0.54	0.52	0.52	0.54	0.55	0.54	0.57	0.55	0.54					
0.61	0.63	0.63	0.63	0.62	0.59	0.63	0.58	0.59	0.62	0.6	0.63	0.59	0.59	0.62				
0.67	0.65	0.64	0.67	0.64	0.66	0.69	0.67	0.65	0.65	0.65	0.68	0.69	0.66	0.66	0.7			
0.73	0.78	0.74	0.71	0.71	0.72	0.76	0.77	0.72	0.77	0.74	0.71	0.77	0.71	0.71	0.74	0.76		
0.84	0.78	0.8	0.82	0.85	0.82	0.86	0.82	0.85	0.84	0.82	0.79	0.79	0.83	0.8	0.81	0.83	0.85	
0.88	0.92	0.87	0.93	0.86	0.93	0.94	0.87	0.88	0.88	0.88	0.95	0.93	0.91	0.94	0.87	0.92	0.92	0.93

A Thought Experiment

Day 3 of Launch

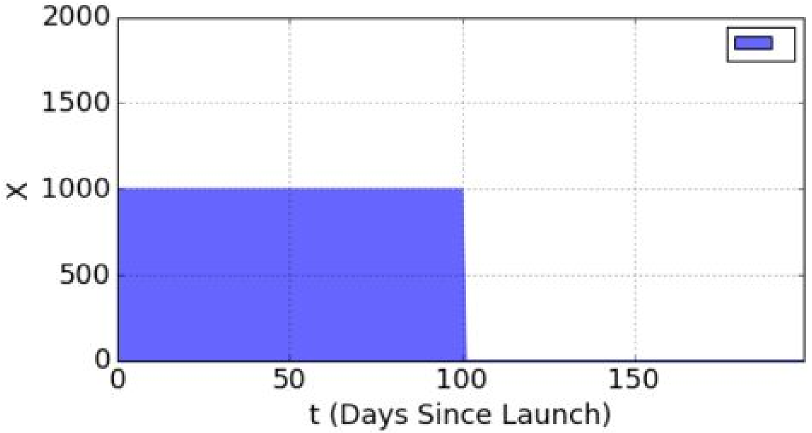
- 1000 new customers per day
- 50% return day after signup, 25% return 2 days later, none return after.
- How many users do we expect on day 5?

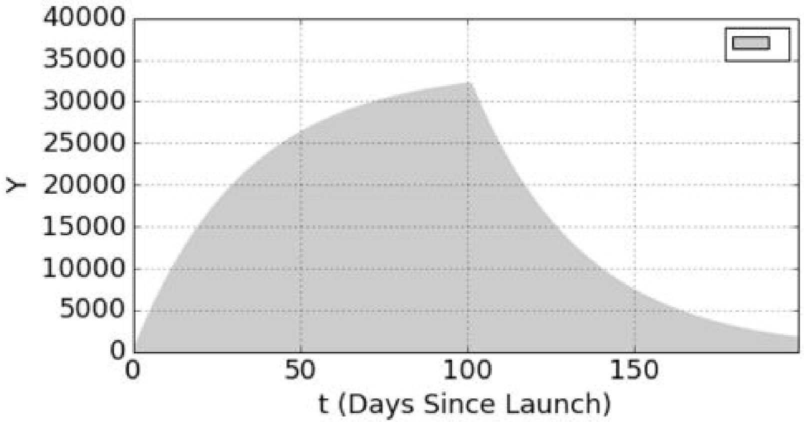
Generalize: If N(t) customers join each day and G(a) return how many do we expect to be active T days from now?

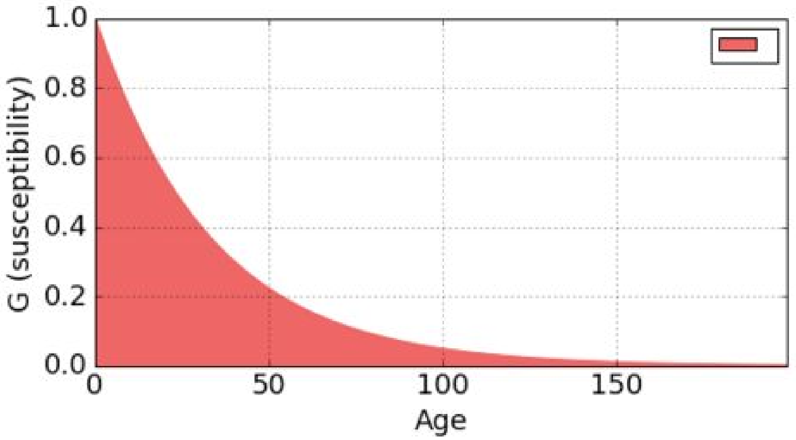
Forecast Simulations

$$Y(t) = XG(t)$$

$$Y(t) = \sum_a X(t-a)G(a)$$



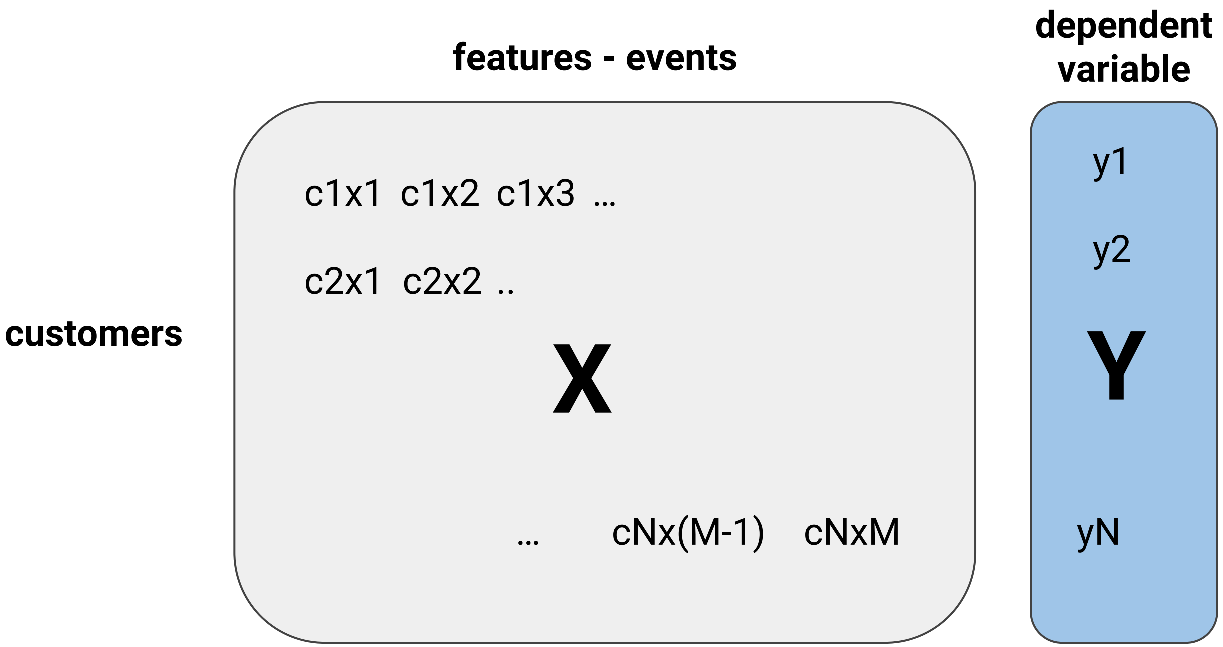




Python SQL Workshop #2

- Import master_fact table from eliflo.
- Explore how to do joins.
- Measure retention curves of a few cohorts.
- Look at retention performance over time.
- Convolve the retention curve with new users to compute future daily active users.

Extending to Multiple X - Supervised Learning



features - events

customers

c1x1 c1x2 c1x3 …

c2x1 c2x2 ..

X

… cNx(M-1) cNxM

dependent variable

y1

y2

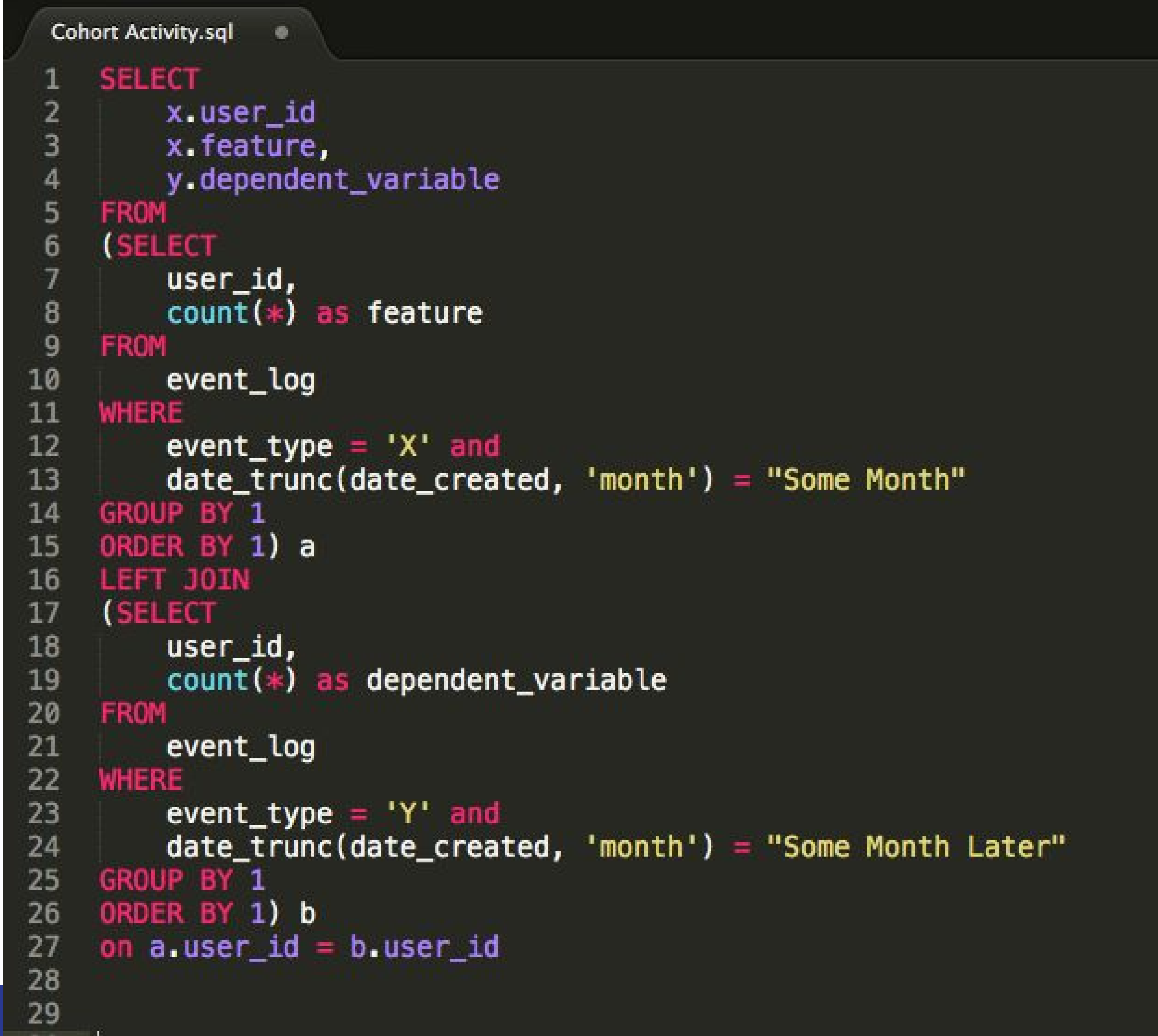
Y

yN

- **Matrix X contains a customer's behavioral profile.**
- **Rows are customers.**
- **Features are behavioral measures on those customers.**
- **Dependent variable, Y, is the customer's 'state' we want to predict.**

Predictive Models

Can generalize to more than one feature



```sql
SELECT
    x.user_id
    x.feature,
    y.dependent_variable
FROM
(SELECT
    user_id,
    count(*) as feature
FROM
    event_log
WHERE
    event_type = 'X' and
    date_trunc(date_created, 'month') = "Some Month"
GROUP BY 1
ORDER BY 1) a
LEFT JOIN
(SELECT
    user_id,
    count(*) as dependent_variable
FROM
    event_log
WHERE
    event_type = 'Y' and
    date_trunc(date_created, 'month') = "Some Month Later"
GROUP BY 1
ORDER BY 1) b
on a.user_id = b.user_id
```

Python SQL Workshop #3

- Predict whether a customer will be active in their 3rd month.
- Write a query to create a data set you can use to build this model.