

Creating Data Democracy with Looker

Sajan Alexander, The Zebra



The Zebra

Simplify Buying Car Insurance

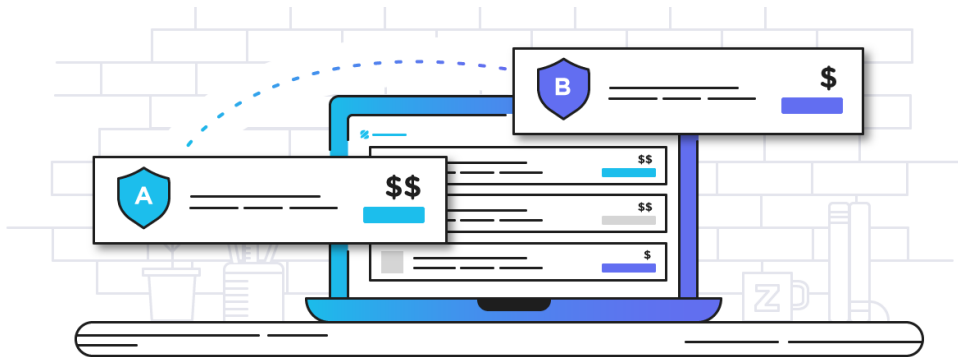




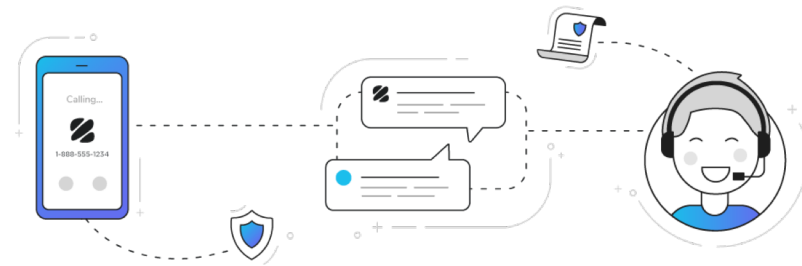
The Zebra

Two Ways to Find Insurance

SEARCH ENGINE



AGENCY

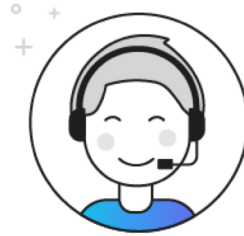




Agency Data



-
- Quote / Sales
 - Multiple databases



-
- Call Center Data
 - 3rd party application



Google Sheets

-
- Mapping rules
 - Updated by Agency

METRICS: Attribution, Cost, Revenue, Agent Performance



Phases in our evolution

STARTUP ANALYTICS

- Direct data pulls
- Rollups + Open Source Dashboards
- Slow metric development

ALL IN ONE

- Inflexible
- Slow to update
- Limited Connectors

ENTERPRISING ANALYST

- Daily downloads
- R Scripts to process
- Static Reports



Steps for Success

- Centralize data in one place
- Model the data properly
- Make data accessible



Centralize Data





Finding the right fit

ALOOMA

- Broad set of connectors
- Simple to add new datasources
- Transform & Consolidate

SNOWFLAKE

- Scale with data volume and analytics load
- Low administration cost
- Support for datatypes needed



Modeling Basics

- Organize around business concepts
 - Views
- Allow for connections
 - Joins
- Develop Dimensions & Measures
 - Columns & Aggregates



Modeling Data

looker



LookML: Making Modeling Easy

- Easy / Flexible
- Write SQL
- Sits on top of Snowflake

```
call_data ▾
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
::

select
  coalesce(am.email, ac.agent_email) as agent_email,
  ac.agent_first_name,
  ac.agent_last_name,
  coalesce(am."GROUP", ac.agent_group) as agent_group,
  (coalesce(am."GROUP", ac.agent_group) is not null and coalesce(am."GROUP", ac.agent_group) = 'Services') as services_call,
  ac.contacted
  and coalesce(am."GROUP", ac.agent_group) is not null
  as answered,
  ac.contacted
  and coalesce(am."GROUP", ac.agent_group) is not null
  and ((coalesce(am."GROUP", ac.agent_group) is null or coalesce(am."GROUP", ac.agent_group) <> 'Services'))
  as sales_answered,
  ac.contacted
  and coalesce(am."GROUP", ac.agent_group) is not null
  and coalesce(am."GROUP", ac.agent_group) = 'Services'
  as services_answered,
  ac.agency_channel,
  ac.ani,
  ac.state,
  ac.call_id,
  ac.call_time,
  ac.call_timestamp,
  ac.call_type,
  ac.campaign,
  ac.campaign_type,
  ac.contacted,
  ac.disposition,
  ac.missed,
  ac.dntls,
  ac.hold_time,
  ac.num_calls,
  ac.num_holds,
  ac.num_parks,
  ac.num_transfers,
  ac.num_voicemails,
  ac.talk_time,
  mc.marketing_cost,
  coalesce(mc.inbound_billable, false) as inbound_billable
from all_calls ac
left join marketing_costs mc on ac.call_id = mc.call_id and ac.call_timestamp = mc.call_timestamp and ac.call_type = mc.call_type
left join google_sheets.agents_mappings am on lower(trim(ac.agent_email)) = lower(am.email)
::
```

```
429 ▾
430
431
432
433
434
435
436
437 ▾
438
439
440
441 ▾
442
443
444
445
446

dimension: customer_state {
  label: "Customer State"
  description: "The state in which the customer resides"
  type: string
  map_layer_name: us_states
  sql: ${TABLE}.state ;;
}

measure: total_answered_calls {
  label: "Total Answered Calls"
  description: "The total number of calls that reached an agent"
  type: count
  filters: {
    field: answered
    value: "Yes"
  }
}
```



Persistent Derived Tables

- Transform Efficiently
- Trigger on new data
- Iterate quickly

call_data ▾

```
191     ac.num_calls,  
192     ac.num_holds,  
193     ac.num_parks,  
194     ac.num_transfers,  
195     ac.num_voicemails,  
196     ac.talk_time,  
197     mc.marketing_cost,  
198     coalesce(mc.inbound_billable, false) as inbound_billable  
199   from all_calls ac  
200   left join marketing_costs mc on ac.call_id = mc.call_id and ac.call_timestamp = mc.call_timestamp and ac.call_type = mc.call_type  
201   left join google_sheets.agents_mappings am on lower(trim(ac.agent_email)) = lower(am.email)  
202   ;;  
203   sql_trigger_value: select max(metadata__timestamp) from agency.call_data ;;  
204 }  
205
```

Connection	View	Type	Last Build Completed	Actions
prod-snowflake	call_data	trigger	11 minutes ago	LookML
<ul style="list-style-type: none">● Table name: LRSFMPERZIGAR5KX2FOSD4Q_call_data● Shared by models: online, offline● Latest build duration: 8.0s● Average build duration: 9.3s● Build Reason: Trigger value changed from 2018-03-06 15:50:16 +0000 to 2018-03-06 16:20:15 +0000● Trigger last checked: 2 minutes ago● Trigger Value: 2018-03-06 16:20:15 +0000				



Making Data Accessible

Explores

- Living data
- Slice and join
- Collaboration

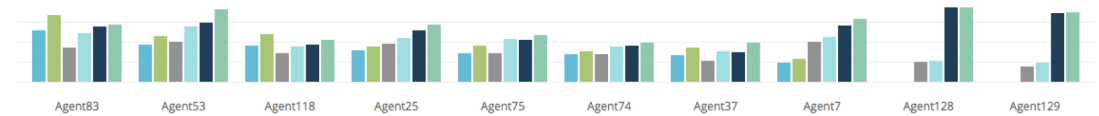
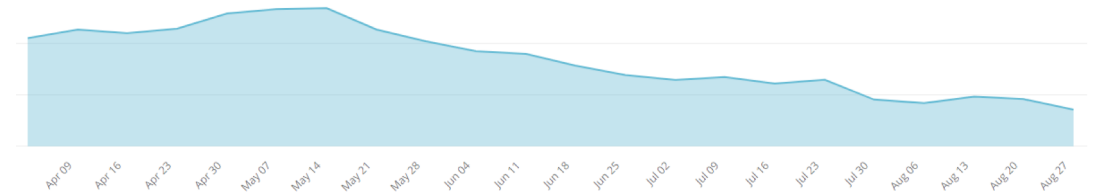
The screenshot shows the Looker 'Explore' interface. At the top, the 'Explore' menu is highlighted with an orange box and an arrow pointing to the 'DATA' tab. The interface displays a table with 21 rows of data. The table has two columns: 'Orders Created Date' and 'Orders Order Count'. The data shows a sequence of dates from 2017-02-18 to 2017-03-10, with corresponding order counts ranging from 1 to 344. The 'DATA' tab is selected, and the 'Row Limit' is set to 500. The 'Calculations' and 'Totals' buttons are visible at the top right of the table area.

Orders Created Date	Orders Order Count
2017-03-10	1
2017-03-09	265
2017-03-08	316
2017-03-07	344
2017-03-06	310
2017-03-05	267
2017-03-04	256
2017-03-03	303
2017-03-02	312
2017-03-01	311
2017-02-28	330
2017-02-27	333
2017-02-26	283
2017-02-25	256
2017-02-24	265
2017-02-23	315
2017-02-22	334
2017-02-21	321
2017-02-20	360
2017-02-19	279
2017-02-18	264



The Payoff

- Reduced marketing costs
- Align incentives to strategic goals
- Data in negotiation





Getting everyone involved

- Data Engineering
 - Exposing new data & modeling
 - Efficiency
- Analysts
 - Creation of KPIs / Dashboards
 - Deep Dive Analysis
- Everyone in the org
 - Exploration
 - Questions