



join

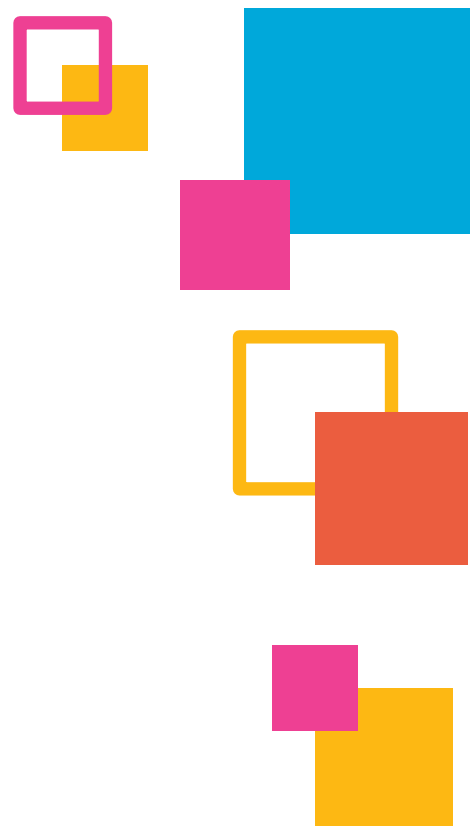


Hands-on Lab: Data Science in Action with Looker & R



Shingi Samudzi

Professional Services





looker.com/hol

Select the **Data Science in Action with Looker & R** lab in the drop-down



Shingi Samudzi

Consultant, Professional Services



Agenda

Introduction

Looker & R

Use cases

Exercise: Looker & R in action

Questions

The journey to data science with Looker



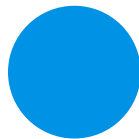
Is this right?

All data is stored on various disconnected Excel spreadsheets or databases



Are we tracking that?

Building clean data pipelines for Looker to model all data and be the single source of truth gives visibility to what is actually being tracked/measured and how



What does it really mean?

Simple statistical modeling helps create a picture of good vs bad conclusions to draw from data



What will happen tomorrow?

Strong modeling with minimized error allows for predictive analytics

What is data science, really?

Software + Statistics + Massive Datasets

Using software to scale the application of statistical models and tests onto massive datasets

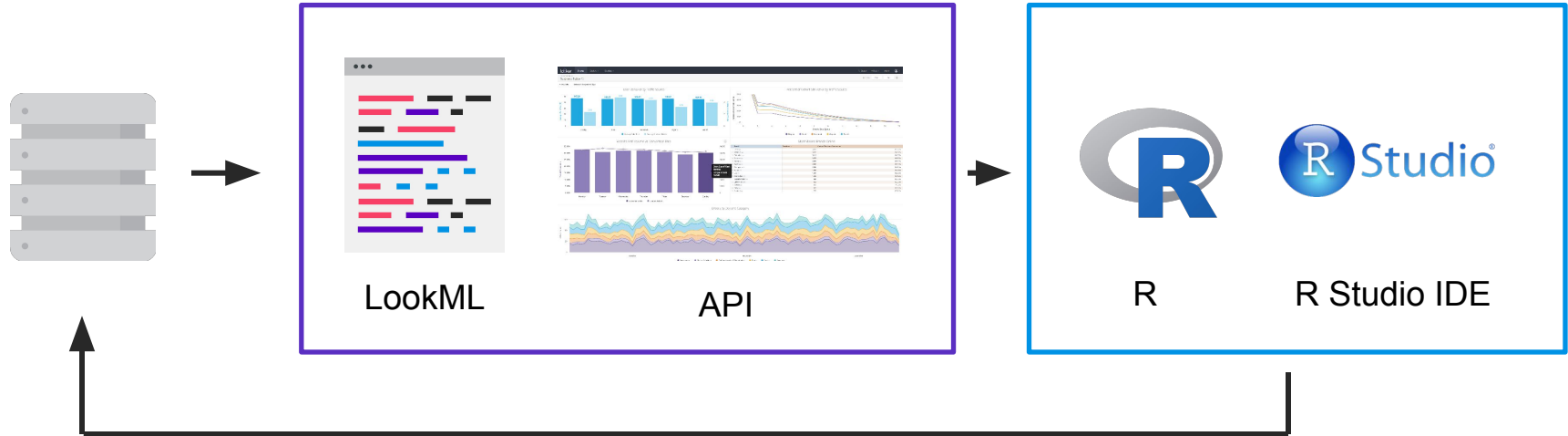
Can also include the data engineering that takes origin data thru ETL into a form appropriate for actual analysis

Looker + R system architecture

Data warehouse

Looker deployment

R deployment





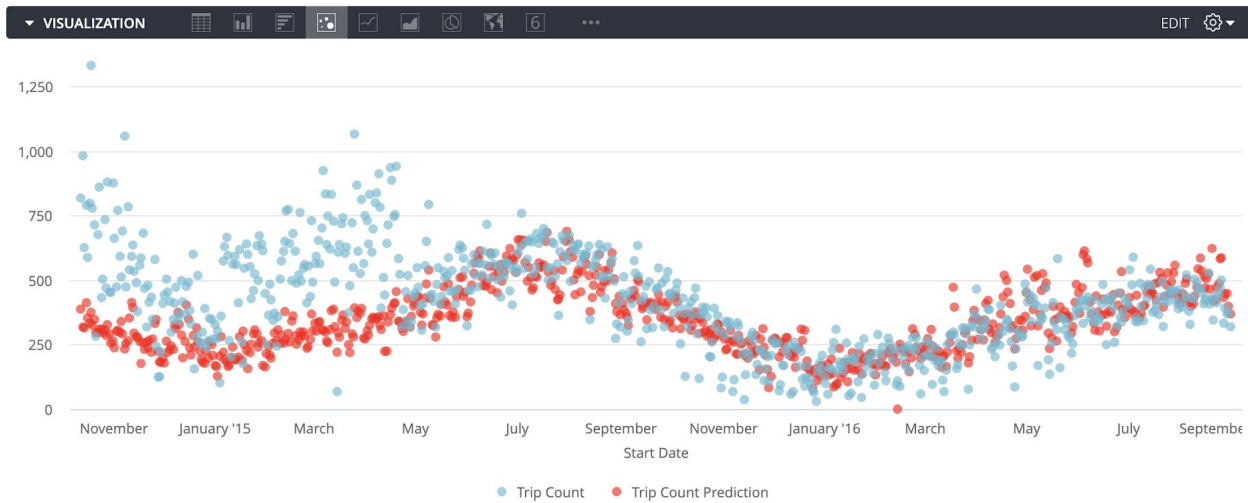
Use cases for Looker + R

- Predict a new customer's shopping choices based on demographic characteristics
- Predict how a customer will respond to a new product offering based on historical data
- Identify credit card fraud among millions of customer transactions
- Create inventory forecasts, financial projections, or event predictions from time series data

Let's see Looker & R in action

Use cases for Looker + R

- Driving alerts and actions through Looker based on predicted outcomes
- Flex staff scheduling based on predicted customer volume in a store



Step 1 — Model data in Looker

Use PDTs to build the model for Looks to ensure data pulled via API is the most up-to-date relative to your underlying dataset

The screenshot displays the Looker web application interface. At the top, the navigation bar includes 'Browse', 'Explore', 'Develop', and 'Admin' menus. The main header shows the project name 'census_analysis' and a search bar. On the left sidebar, under 'Shared Branch dev_shingai-2', there's a 'Commit Changes' button and a list of views including 'housingPop_analysis.view'. The central pane shows the 'housingPop_analysis' model being edited, with a LookML query that includes a derived table 'housingpop_analysis' and various columns and filters. The right sidebar contains a 'Quick Help' section with a link to documentation and a definition of the 'model' keyword.

```
1 # If necessary, uncomment the line below to include explore_source.
2 #include: "census_analysis.model.lkml"
3
4 view: housingpop_analysis {
5   derived_table: {
6     persist_for: "24 hours"
7     explore_source: ziptoall_flat {
8       column: zipcode { field: opp_zone_list.zipcode }
9       column: yr2017 { field: hpi_complete.yr2017 }
10      column: sum_total_population_with_income { field: income_flat.sum_total_population_with_income }
11      column: sum_total_adult_pop { field: demographics_flat.sum_total_adult_pop }
12      column: hpiChangeByLocation { field: hpi_complete.hpiChangeByLocation }
13      column: total_population_below_poverty { field: poverty_flat.total_population_below_poverty }
14      column: sum_total_pop { field: demographics_flat.sum_total_pop }
15      column: average_median_owner_occupied_house_value { field: homeownersvalue_flat.average_median_owner_occup
16      column: average_median_monthly_housing_cost_overall { field: income_housing_flat.average_median_monthly_h
17      column: total_occupied_units { field: housingstockage_flat.total_occupied_units }
18      column: average_median_income_overall { field: income_housing_flat.average_median_income_overall }
19      column: average_percent_of_units_newer_than_5yrs { field: housingstockage_flat.average_percent_of_units_n
20    }
21    filters: {
22      field: hpi_complete.yr2017
23      value: "NOT NULL"
24    }
25    filters: {
26      field: demographics_flat.sum_total_adult_pop
27      value: ">0"
28    }
29  }
30  # indexes: ["zipcode"]
31
32  measure: count {
33    type: count
```

Quick Help →

A **model** references a combination of related explores. Unlike other LookML elements, a model is not declared explicitly with the **model** keyword.

```
model: {
  access_grant: identifier
  case_sensitive: yes or no
  connection: "string"
  datagroup: identifier
  explore: identifier
  fiscal_month_offset: number
  include: "string"
  label:
    possibly-localized-string
  map_layer: identifier
  named_value_format: identifier
  persist_for: "string"
  persist_with: datagroup-ref
  view: identifier
  week_start_day: monday or ...
}
```

Step 2 — Using LookR to access Looks

Create a `looker.ini` file to store your API connection details

```
1 [Looker]
2 # API version is required
3 api_version=3.0
4 # Base URL for API. Do not include /api/* in the url
5 base_url=https://data.asoba.co:19999
6 # API 3 client id
7 client_id=v53XyrTCYKrqxkRDwt4y
8 # API 3 client secret
9 client_secret=J949yjq7hP5x8ggkSQWmW2Xx
10 # Optional embed secret for SSO embedding
11 #embed_secret=your_embed_SSO_secret
12 # Optional user_id to impersonate
13 user_id=1
14 # Set to false if testing locally against self-signed certs. Otherwise leave True
15 verify_ssl=True
```

Then install the LookR R package and establish a connection with your Looker instance

```
1 #Installing Lookr from dev branch
2 #devtools::install_github("looker/lookr", ref = "dev", force=TRUE)
3
4 #Initialize the LookerAPI connection
5 setwd("~/Projects/looker-sdk/")
6
7 library(lookr)
8
9 sdk <- LookerSDK$new(configFile = "looker.ini")
10
```

Step 3 — Build/train ML model

Load libraries you plan on using, and then use the API to load data from your Look as a dataframe

```
19 #Make sure that the Looker Connection is initialized
20 source(file=~ /Projects/initializeLooker.R")
21
22 setwd("~/Projects/DemandModeling")
23 #Load predictive R library
24 source(file="lib/predictiveR2.R")
25 source(file="lib/crossvalidation.R")
26
27 #Load all of the key libraries
28 library(plyr)
29 library(dplyr)
30 library(quantmod)
31 library(purrr)
32 library(data.table)
33 library(tidyr)
34 library(MVLM)
35 library(car)
36 library(ggplot2)
37 library(GGally)
38 library(scatterplot3d)
39 library(rrr)
40 library(caret)
41 library(leaps)
42
43 #pull data from the Looker API
44 base_data <- sdk$runLook(lookId = 51)
45
```

Step 3 — Build/train ML model

Load libraries you plan on using, and then use the API to load data from your Look as a dataframe

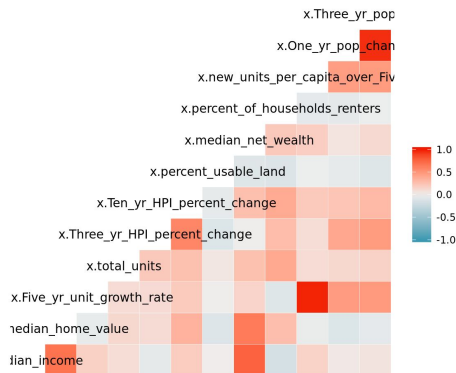
```
19 #Make sure that the Looker Connection is initialized
20 source(file=~ /Projects/initializeLooker.R")
21
22 setwd("~/Projects/DemandModeling")
23 #Load predictive R library
24 source(file="lib/predictiveR2.R")
25 source(file="lib/crossvalidation.R")
26
27 #Load all of the key libraries
28 library(plyr)
29 library(dplyr)
30 library(quantmod)
31 library(purrr)
32 library(data.table)
33 library(tidyr)
34 library(MVLM)
35 library(car)
36 library(ggplot2)
37 library(GGally)
38 library(scatterplot3d)
39 library(rrr)
40 library(caret)
41 library(leaps)
42
43 #pull data from the Looker API
44 base_data <- sdk$runLook(lookId = 51)
45
```


Step 3 — Build/train ML model

Start with an exploratory data analysis to narrow down the set of variables you will use for your model

EDA techniques like **ggpairs** allow you to spot highly correlated variables

```
106 model_plot <- ggpairs(data=base_df2, columns=1:11, title="Real Estate Demand Variables",  
107 progress=NULL)  
108 #second form of variable analysis to assess correlations between predictor variables  
109  
110 model_x <- base_df2 %>%  
111   select(starts_with("x"))  
112 model_y <- base_df2 %>%  
113   select(starts_with("y"))  
114  
115 GGally::ggcorr(model_x)  
116 GGally::ggcorr(model_y)
```



Step 3 — Build/train ML model

Once you have narrowed down your variables, you can use a technique like Best Subsets Regression to measure regression model quality. In this example, I am measuring using three different metrics - Adjusted R^2 , Bayesian information criteria, and Mallows Cp

```
237 #for 1yr HPI Change
238 var_subset2 <- regsubsets(y.One_yr_HPI_percent_change~., data = model_x4, nvmax = 14)
239 res_subset2<-summary(var_subset2)
240 res_sum2<-data.frame(
241   Adj.R2 = which.max(res_subset2$adjr2),
242   CP = which.min(res_subset2$cp),
243   BIC = which.min(res_subset2$bic)
```

Step 3 — Build/train ML model

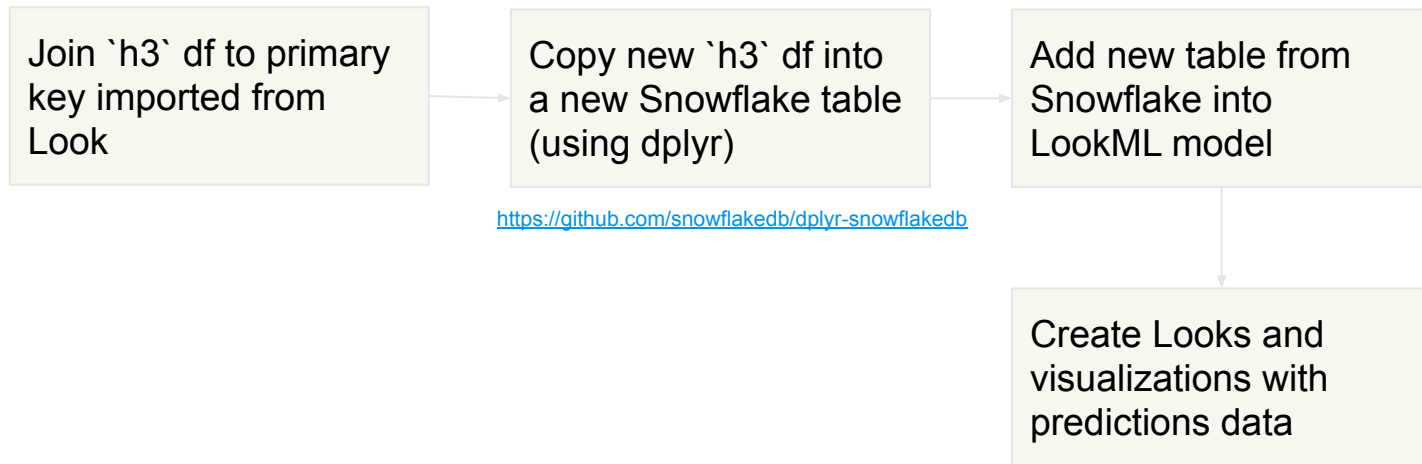
If you use multiple metrics in best subsets regression and each suggests a different “best” model, cross-validation error can give more insight into which set of variables most accurately predicts against your actual train/test data

```
253 # Compute cross-validation error for HPI
254 model.ids <- 1:5
255 cv.errors <- map(model.ids, get_model_formula, var_subset, "y.HPI") %>%
256   map(get_cv_error, data = model_x3) %>%
257   unlist()
258 cv.errors
259
260 # Select the model that minimize the CV error
261 which.min(cv.errors)
262
263 #Based on the response, select the best set of variables based on the number with lowest CV error
264 get_model_formula(5, var_subset, "y.HPI")
265
266 #This is the H3 model for predicting HPI
267 h3 <- lm(y.HPI ~ x.median_home_value + x.Three_yr_HPI_percent_change +
268   x.Ten_yr_HPI_percent_change + x.median_net_wealth + x.percent_of_households_renters, data
   =base_df2)
```

Dataframe `h3`
contains our predictor
values

Step 4 — Make data available to Looker

Let's assume that we are using Snowflake as our data warehouse. Here is a simple process outlining how to make our predicted values available for modelling back within LookML.



The journey to data science with Looker



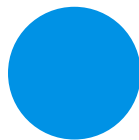
Is this right?

All data is stored on various disconnected Excel spreadsheets or databases



Are we tracking that?

Building clean data pipelines for Looker to model all data and be the single source of truth gives visibility to what is actually being tracked/measured and how



What does it really mean?

Simple statistical modeling helps create a picture of good vs bad conclusions to draw from data



What will happen tomorrow?

Strong modeling with minimized error allows for predictive analytics

Questions?

The background is a solid purple color with a pattern of thin, light-purple lines radiating from a central point. There are three small, dark-purple geometric shapes: a square of dots in the upper right, a vertical bar of horizontal lines in the middle left, and a horizontal bar of vertical lines in the lower right.



looker





Thank you

Rate this session in
the JOIN mobile app