



# PRACTICAL IOT DEVELOPMENT USING ORACLE BIG DATA AND ORACLE DV ...AND A WIFI KETTLE

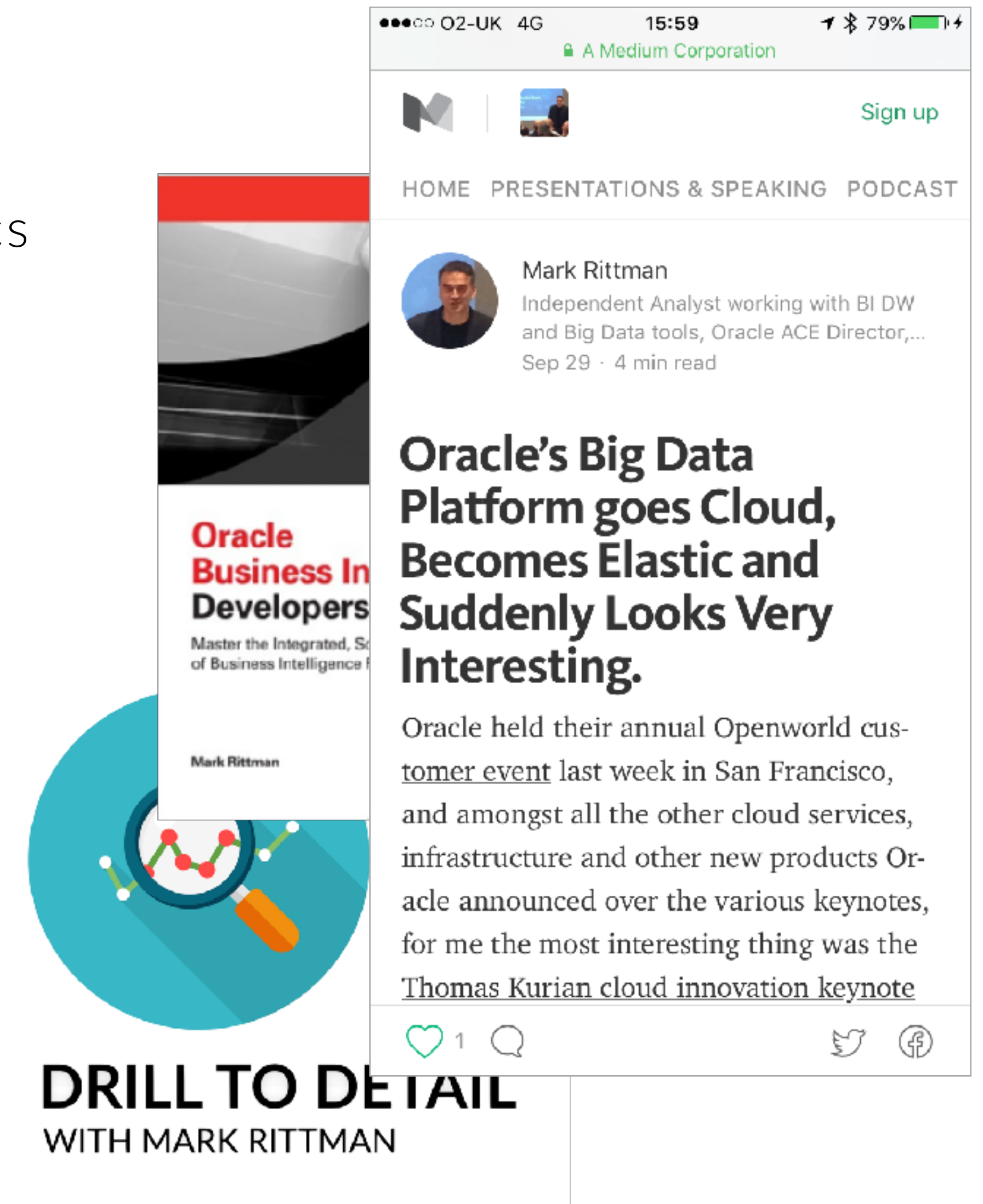
Mark Rittman, Oracle ACE Director & Independent Analyst  
MJR Analytics Ltd (<http://www.mjr-analytics.com>)

BIWA SUMMIT 2017, SAN FRANCISCO



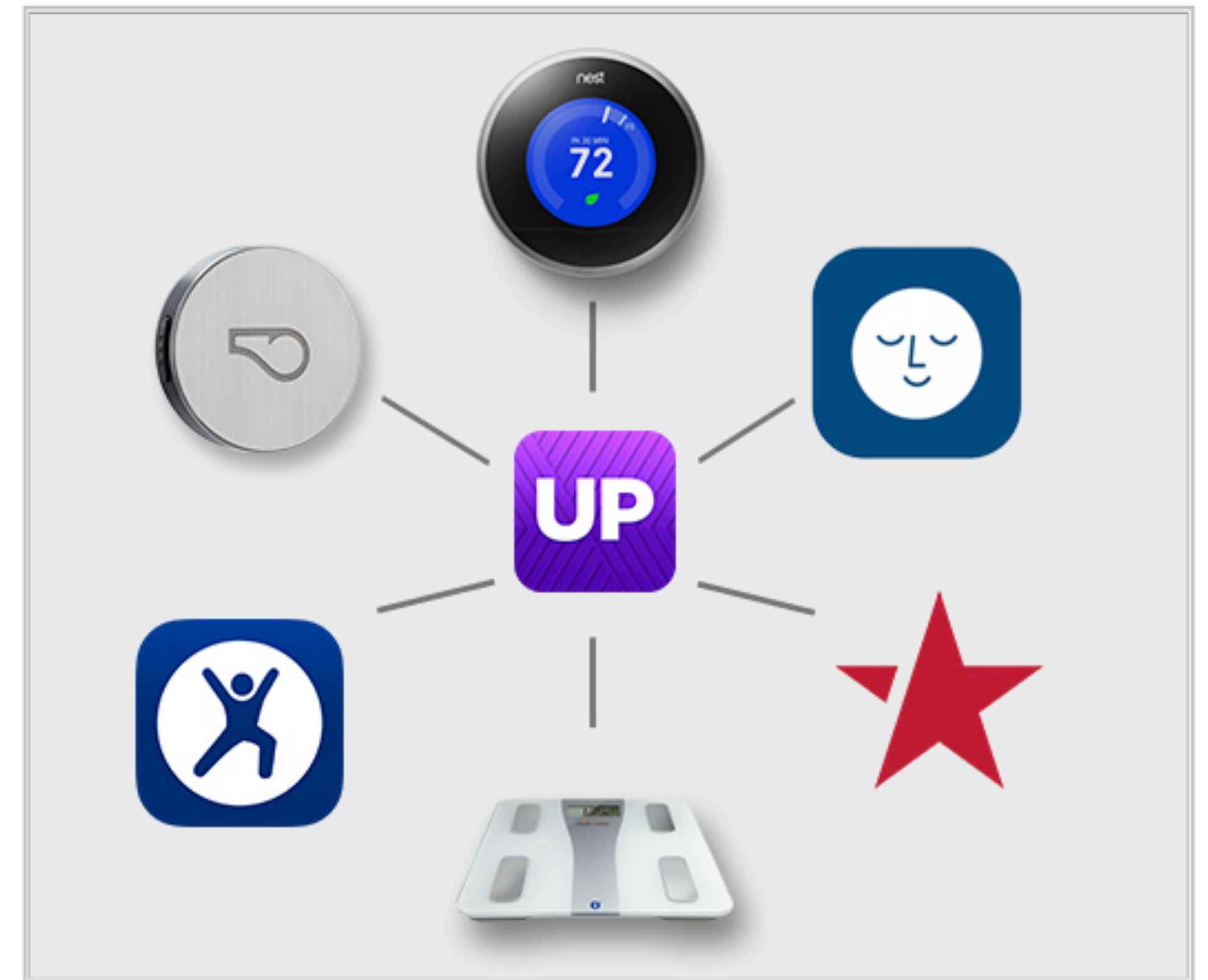
# About Mark Rittman

- Oracle ACE Director, Independent Analyst
- Company founder, Oracle ACE Director, product specialist
- Now working in product management around big data & analytics
- Regular columnist for Oracle Magazine, OTN
- Author of two books on Oracle BI & Engineered Systems
- 15+ Years in Oracle BI, DW, ETL + now Big Data
- Personal blog at [medium.com/mark-rittman](https://medium.com/mark-rittman)
- Podcast on iTunes and [drilltodetail.com](https://drilltodetail.com)
- Contact me at [mark@rittman.co.uk](mailto:mark@rittman.co.uk)



# Wearables & Smart Devices - Our Data Ecosystem

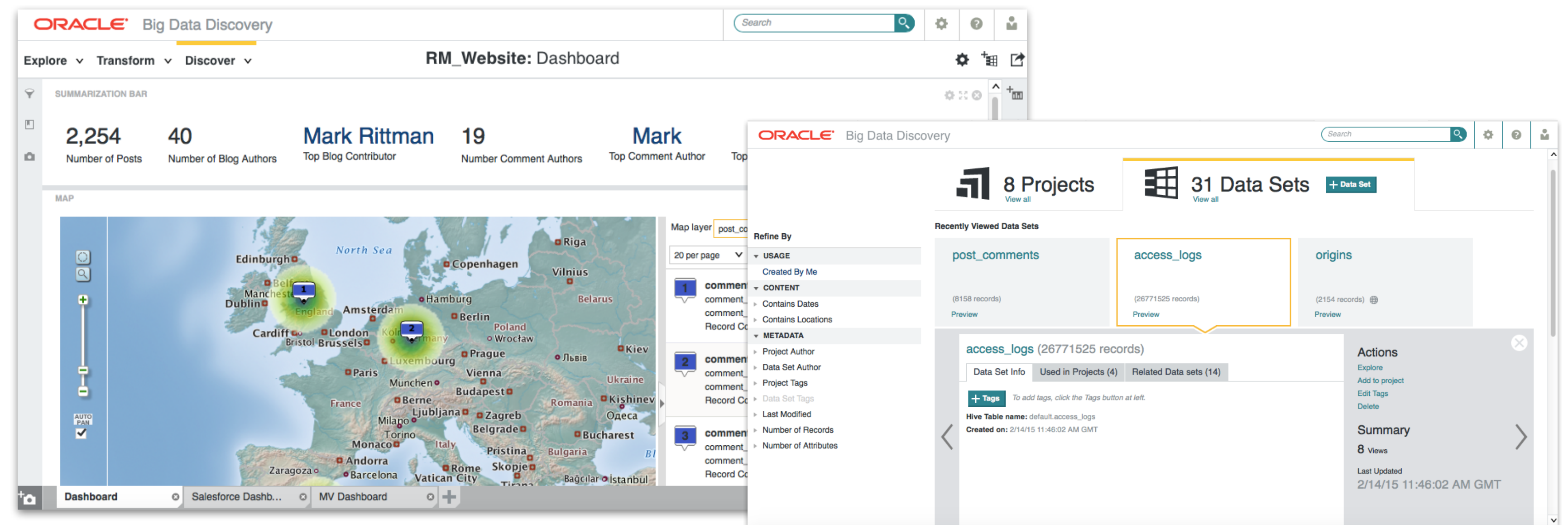
- How many of you are using health bands, smartphone apps, other life logging services?
- It's likely fair proportion of you log workouts, steps and other activities daily
- Some of you may have Nest, Hue or other home smart devices
- All of these services capture and generate useful data
- What if we could capture, combine and mine this data for insights, correlations, trends and patterns?
  - And what if we used Oracle Big Data Discovery to bring the data together, and mine for those insights?



**FOR THE PAST SIX MONTHS, I DID JUST THAT**

# Oracle Big Data Discovery - What Is It?

- A visual front-end to the Hadoop data reservoir, providing end-user access to datasets
- Data sampled and loaded from Hadoop (Hive) into NoSQL Dgraph engine for fast analysis
- Catalog, profile, analyse and combine schema-on-read datasets across the Hadoop cluster
- Visualize and search datasets to gain insights, potentially load in summary form into DW





# Key Features in Oracle Big Data Discovery 1.1.x

## VISUALISING AND TRANSFORMING DATA

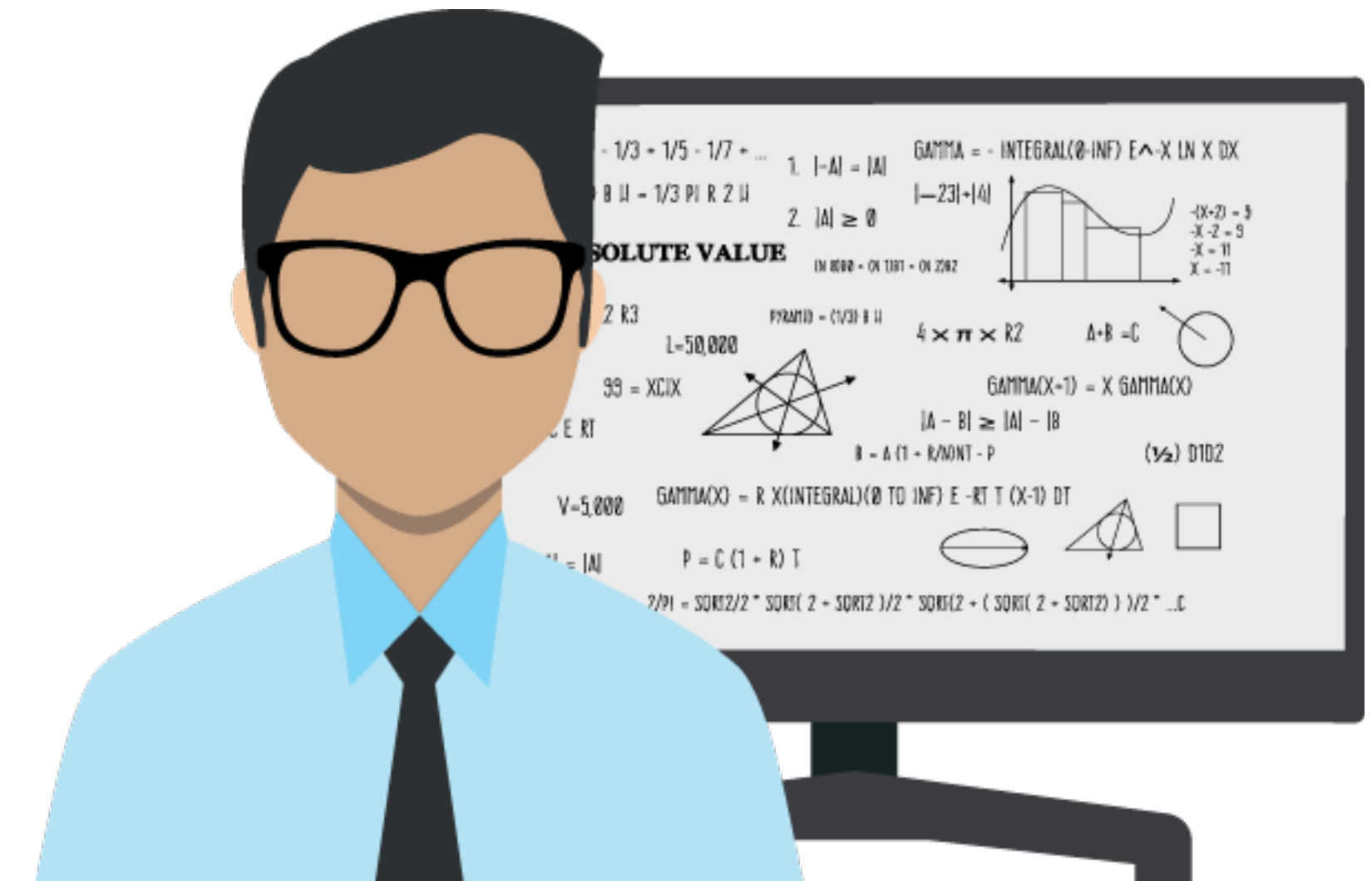


- Provide a visual catalog and search function across data in the data reservoir
- Profile and understand data, relationships, data quality issues
- Apply simple changes, transformations to data
- Add enrichment to incoming data including sentiment, geo-location

## COMMUNICATING AND BUNDLING



- Visualize datasets using rich chart types
- Join datasets at visualisation level
- Add data from JDBC + file sources
- Prepare more structured Hadoop datasets for use with other tools



# New Features In Oracle Big Data Discovery 1.2

## IMPORTING AND TIDYING DATA



- **Aggregation**
- **Materialised Joins**
- Better Pan and Zoom
- Speed and Scale

## METADATA AND DEVELOPER PRODUCTIVITY



- Metadata Curation
- Attribute-level Search from Catalog
- Activity Hub
- **Python Interface to BDD Datasets**

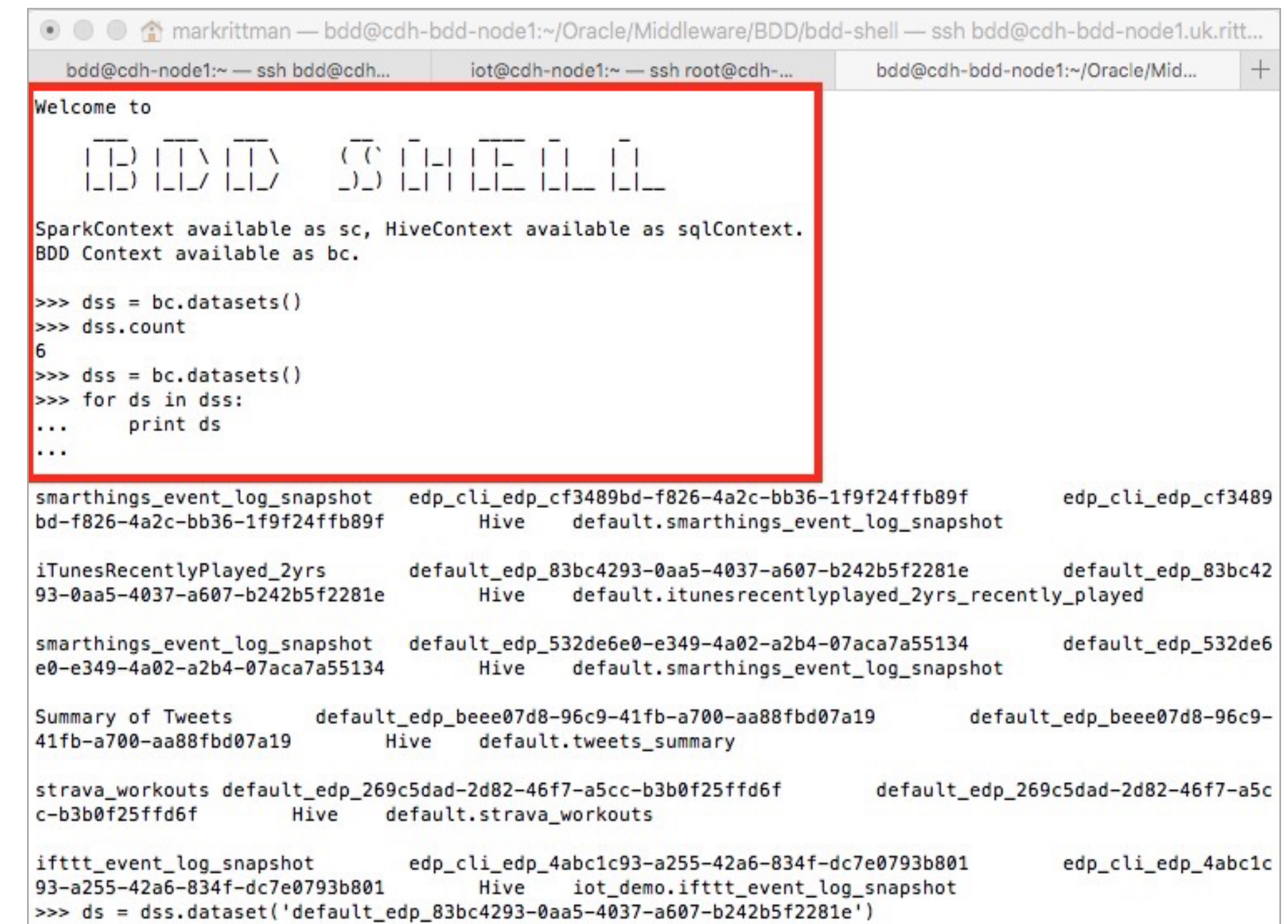
## COMMUNICATING AND BUNDLING



- Streamlined UI
- Faster Data Indexing
- Activity Hub
- Sunburst Visualization

# BDD Shell - pySpark Command-Line

- Interactive tool designed to work with BDD without using Studio's front-end
- Exposes all BDD concepts (views, datasets, data sources etc)
- Supports Apache Spark
- HiveContext and SQLContext exposed
- BDD Shell SDK for easy access to BDD features, functionality
- Access to third-party libraries such as Pandas, Spark ML, numPy
- Use with web-based notebook such as iPython, Jupyter, Zeppelin



```
markritman — bdd@cdh-bdd-node1:~/Oracle/Middleware/BDD/bdd-shell — ssh bdd@cdh-bdd-node1.uk.ritt...
bdd@cdh-node1:~ — ssh bdd@cdh...  iot@cdh-node1:~ — ssh root@cdh-...  bdd@cdh-bdd-node1:~/Oracle/Mid... +
Welcome to
  BDD SHELL

SparkContext available as sc, HiveContext available as sqlContext.
BDD Context available as bc.

>>> dss = bc.datasets()
>>> dss.count
6
>>> dss = bc.datasets()
>>> for ds in dss:
...     print ds
...

smarthings_event_log_snapshot  edp_cli_edp_cf3489bd-f826-4a2c-bb36-1f9f24ffb89f  edp_cli_edp_cf3489
bd-f826-4a2c-bb36-1f9f24ffb89f  Hive  default.smarthings_event_log_snapshot

iTunesRecentlyPlayed_2yrs  default_edp_83bc4293-0aa5-4037-a607-b242b5f2281e  default_edp_83bc42
93-0aa5-4037-a607-b242b5f2281e  Hive  default.itunesrecentlyplayed_2yrs_recently_played

smarthings_event_log_snapshot  default_edp_532de6e0-e349-4a02-a2b4-07aca7a55134  default_edp_532de6
e0-e349-4a02-a2b4-07aca7a55134  Hive  default.smarthings_event_log_snapshot

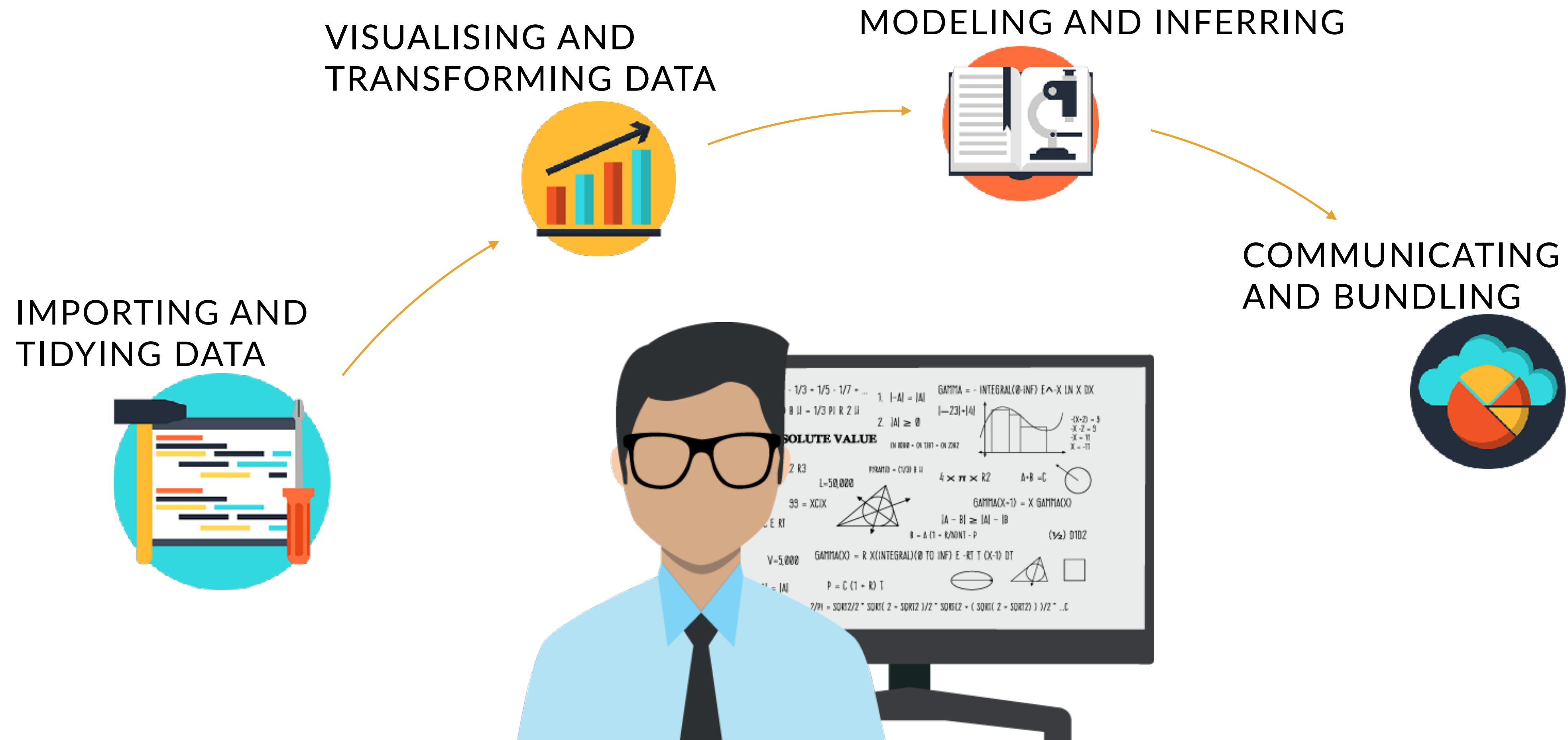
Summary of Tweets  default_edp_beee07d8-96c9-41fb-a700-aa88fbd07a19  default_edp_beee07d8-96c9-
41fb-a700-aa88fbd07a19  Hive  default.tweets_summary

strava_workouts  default_edp_269c5dad-2d82-46f7-a5cc-b3b0f25ffd6f  default_edp_269c5dad-2d82-46f7-a5c
c-b3b0f25ffd6f  Hive  default.strava_workouts

ifttt_event_log_snapshot  edp_cli_edp_4abc1c93-a255-42a6-834f-dc7e0793b801  edp_cli_edp_4abc1c
93-a255-42a6-834f-dc7e0793b801  Hive  iot_demo.ifttt_event_log_snapshot
>>> ds = dss.dataset('default_edp_83bc4293-0aa5-4037-a607-b242b5f2281e')
```

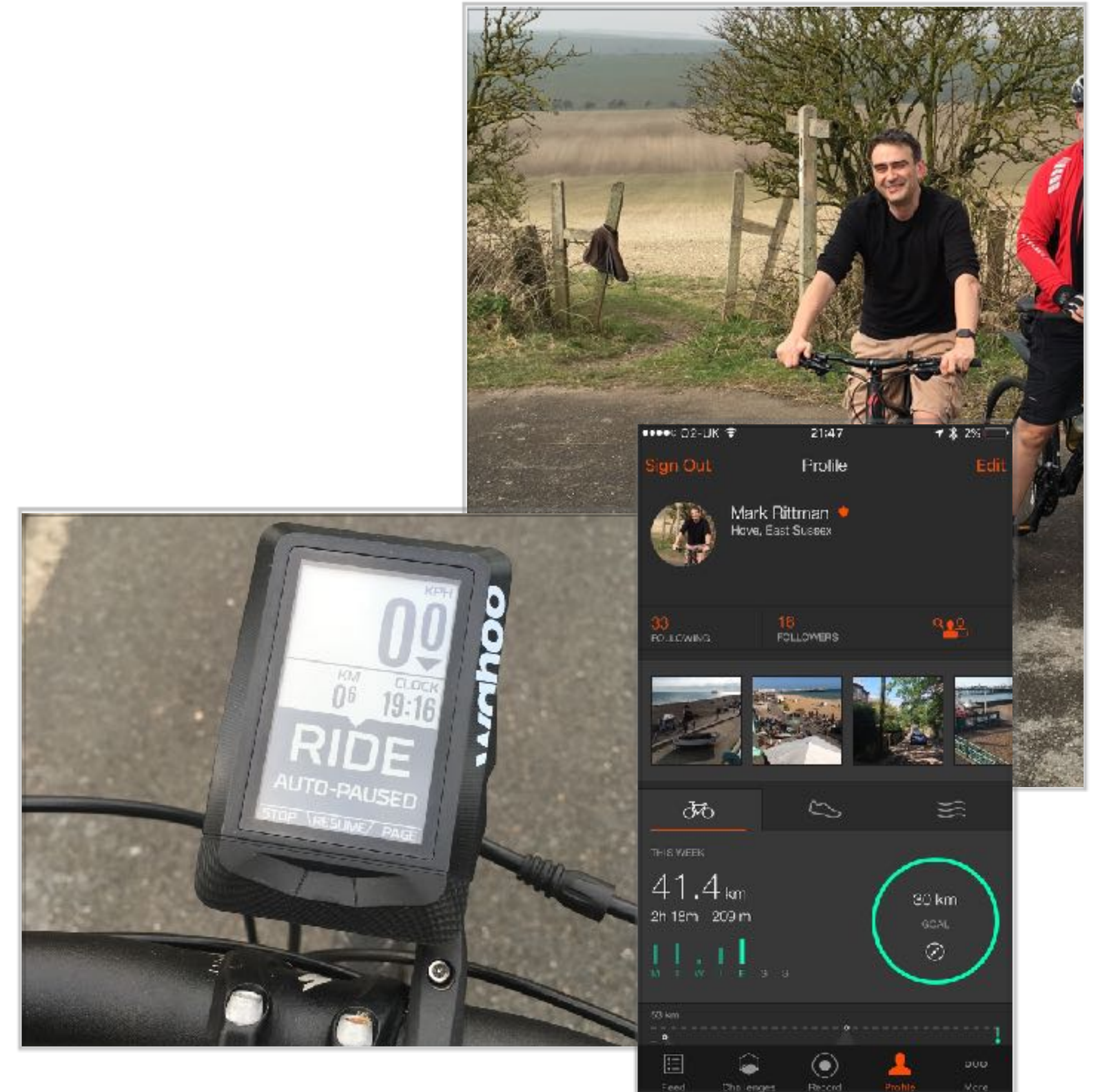


# Oracle Big Data Discovery as the Data Scientists' Toolkit



# Using Wearables To Enhance & Improve Workouts

- Over the past year or so, I've getting into cycling and generally trying to keep fit and lose weight
- Also using these activities as data sources for this project
  - Using Wahoo Elemnt + Strava for workout recording
  - Withings Wifi scales for weight + body fat measurement
  - Jawbone UP3 for steps, sleep, resting heart rate
  - All the time, collecting data and storing it in Hadoop







### Shades

Open the shades and let the morning light in.



### Electric Kettle

Start the electric kettle so the water is ready for your tea.



### Thermostat

Turn up the thermostat before you get out of bed.

# HOME AUTOMATION



# Home Automation and Smart 'IoT' Devices

- Another personal project has been home Automation, IoT and the "smart home"
- Started with Nest thermostat and Philips Hue lights
- Extended the Nest system to include Nest Protect and Nest Cam
- Used Apple HomeKit, HomeBridge, Apple TV for Siri voice control
- Added Samsung Smart Things hub for Z-wave, Zigbee compatibility





Smart homes

# English man spends 11 hours trying to make cup of tea with Wi-Fi kettle

Data specialist Mark Rittman spent an entire day attempting to set up his new appliance so that it would boil on command

Bonnie Malkin

@bonniemalkin

Wednesday 12 October 2016 02.29 BST



Shares 2,747 Comments 591



Mark Rittman sat about trying to make a cup of tea at 8pm but night had fallen by the time his new Wi-Fi

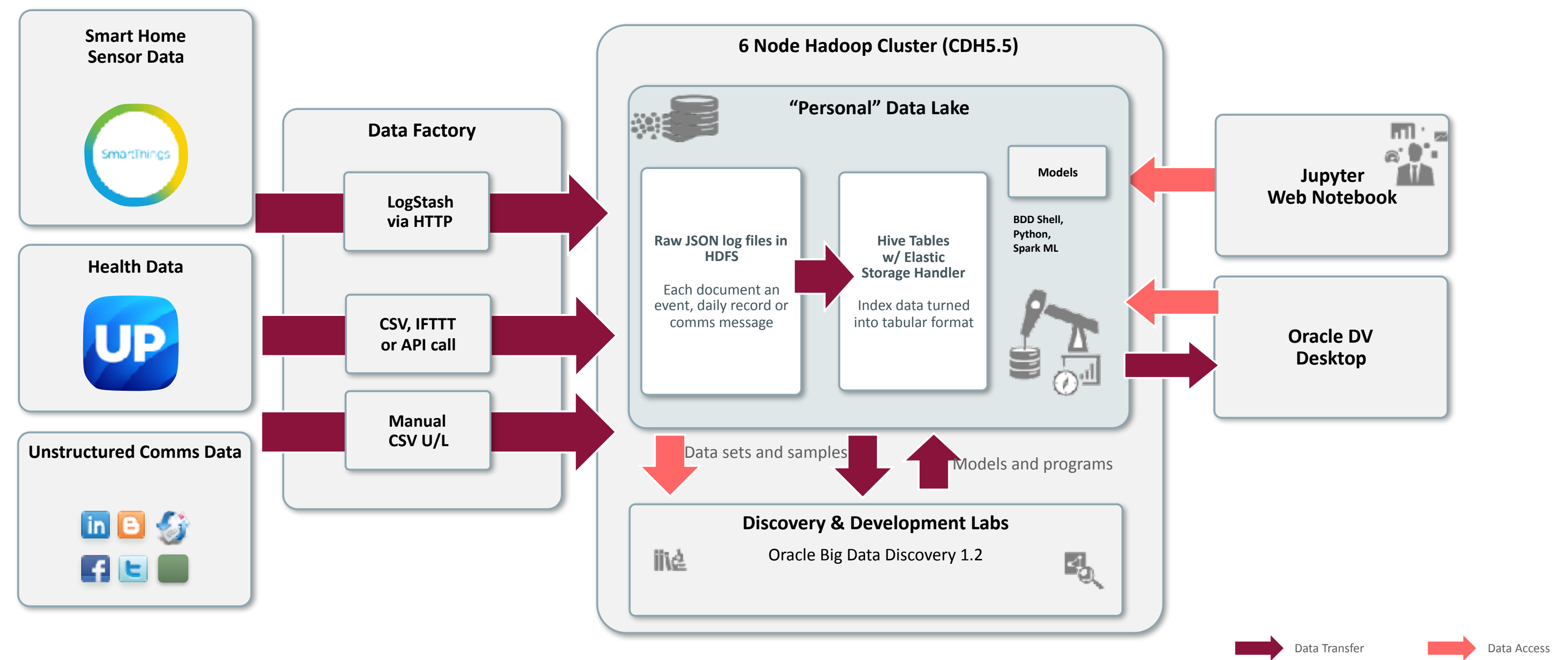
Advertisement





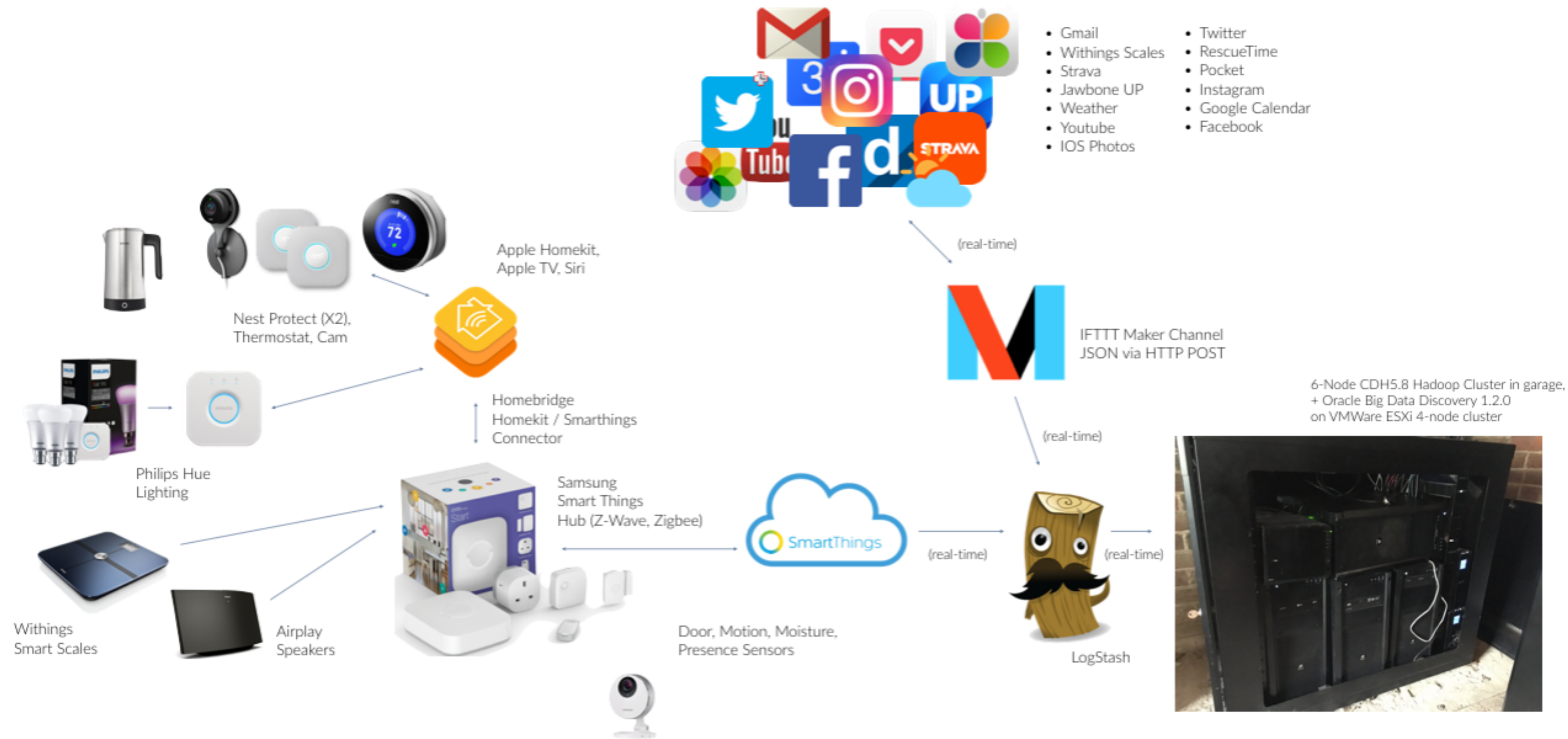
# And The Third Hobby : Land All That Data Into Hadoop

- Data extracted or transported to target platform using LogStash, CSV file batch loads
- Landed into HDFS as JSON documents, then exposed as Hive tables using Storage Handler
- Cataloged, visualised and analysed using Oracle Big Data Discovery + Python ML



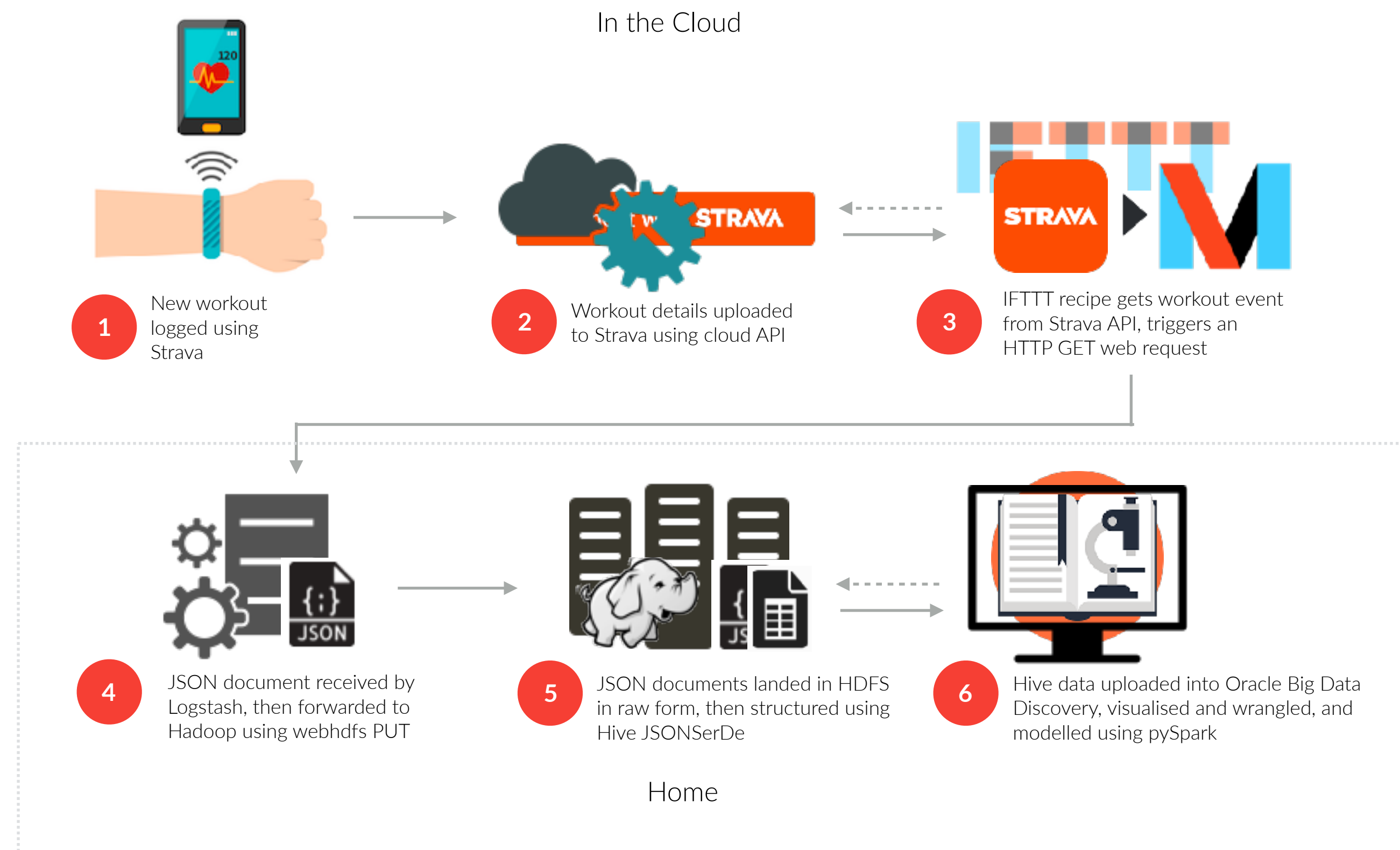


# Real-Time Logging of IoT + Wearable Activity Data



# Landing Wearables Data In Real-Time

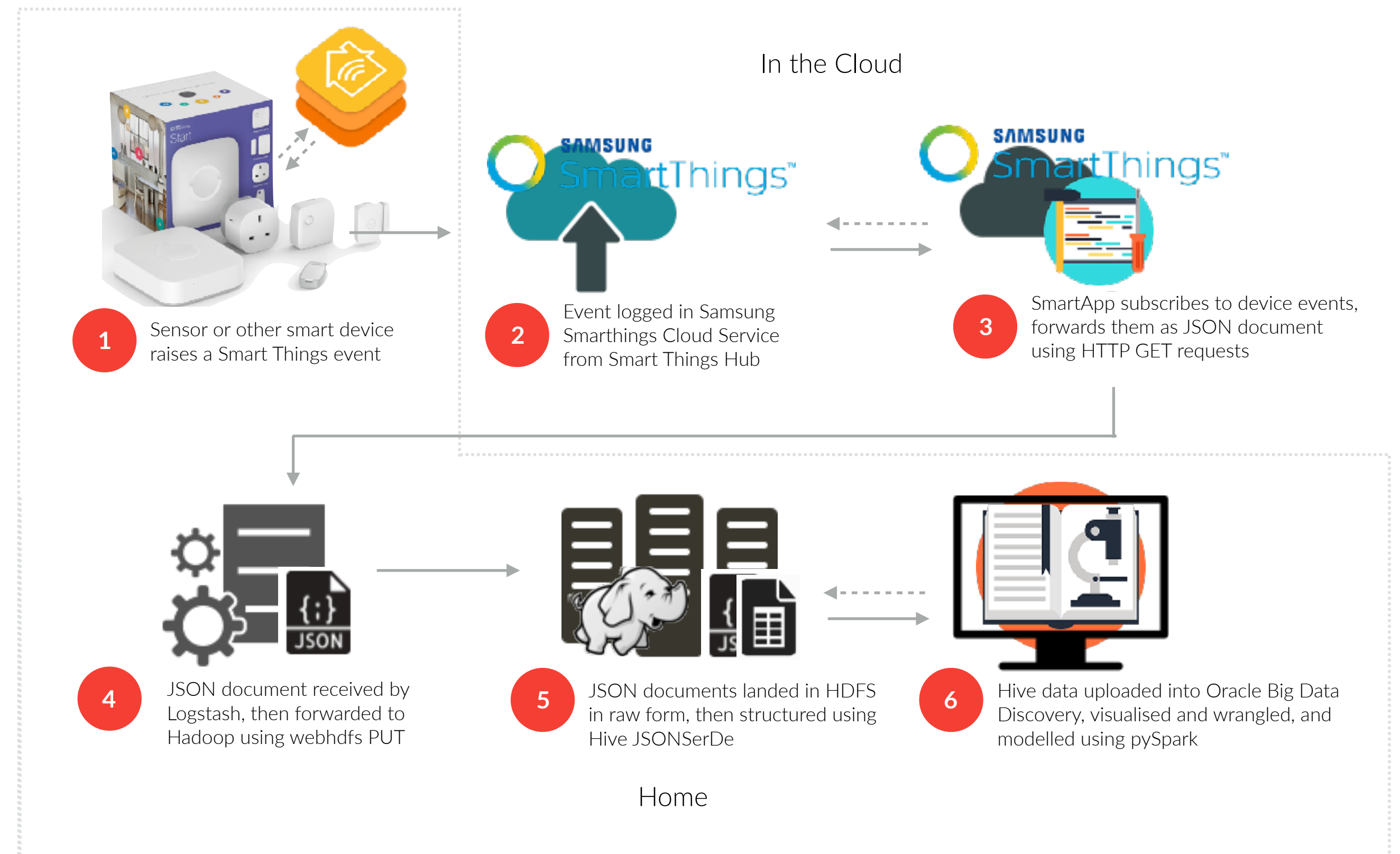
- Uses IFTTT cloud workflow service to subscribe to events on wearables' APIs
- Triggers HTTP GET request via IFTTT Maker Channel to Logstash running at home
- Event data sent as JSON documents, loaded into HDFS via webhdfs protocol
- Structured in Hadoop using Hive JSONSerDe
- Then loaded hourly into DGraph using Big Data Discovery dataprocessing CLI
- Event data automatically enriched, and can be joined to smart home data for analysis





# Landing Smart Home Data In Real-Time

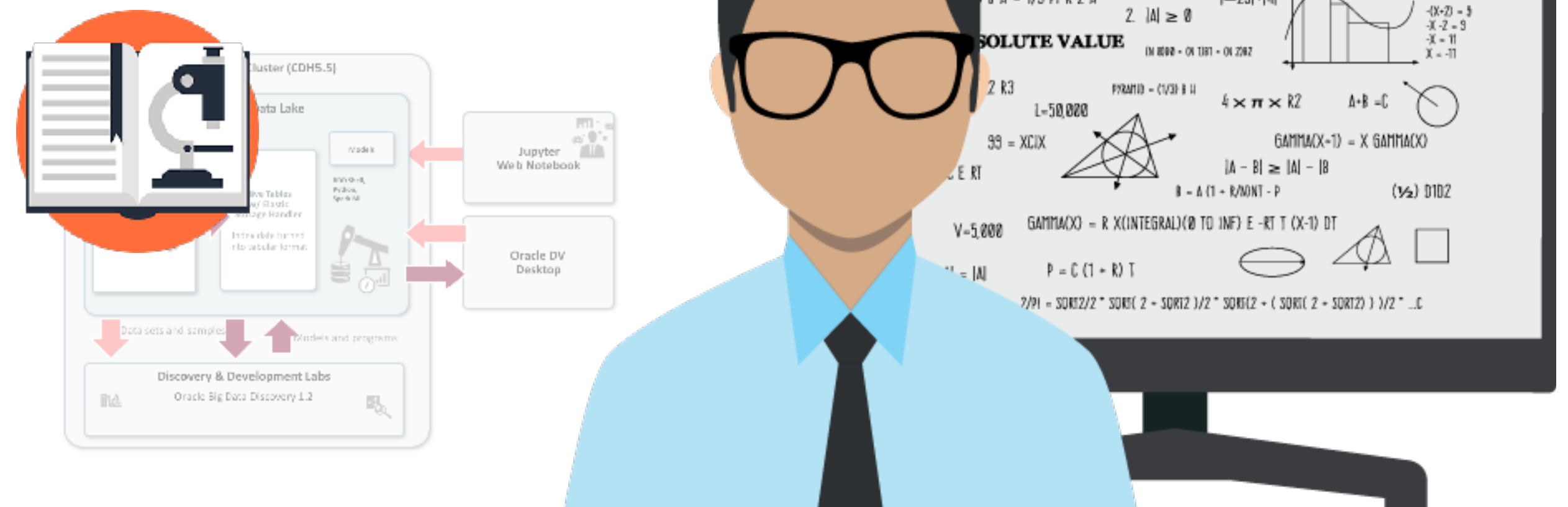
- All smart device events and sensor readings are routed through Samsung Smart Things hub
  - Including Apple HomeKit devices, through custom integration
- Event data uploads to Smart Things cloud service + storage
- Custom Groovy SmartApp subscribes to device events, transmits JSON documents to Logstash using HTTP GET requests
- Then process flow the same as with wearables and social media / comms data



# Initial Focus Area : What Drives Weight Gain/Loss?

- This combined dataset can potentially be used to answer some interesting questions
- For example ... “which of my daily activities or behaviours has most influence on my weight?”
  - Is it amount of exercise? amount of sleep? What I eat? How much work I’m doing in evenings?
- Objective is to work out which variable has the most influence on % weight change wk/wk
  - Will require tidying/reformatting of data feeds to standardise dates, bin and transform data
  - Dealing with nulls where workouts, weight readings were missed on certain days
  - Aggregating and joining different datasets
  - Build linear regression model to identify most influential variable

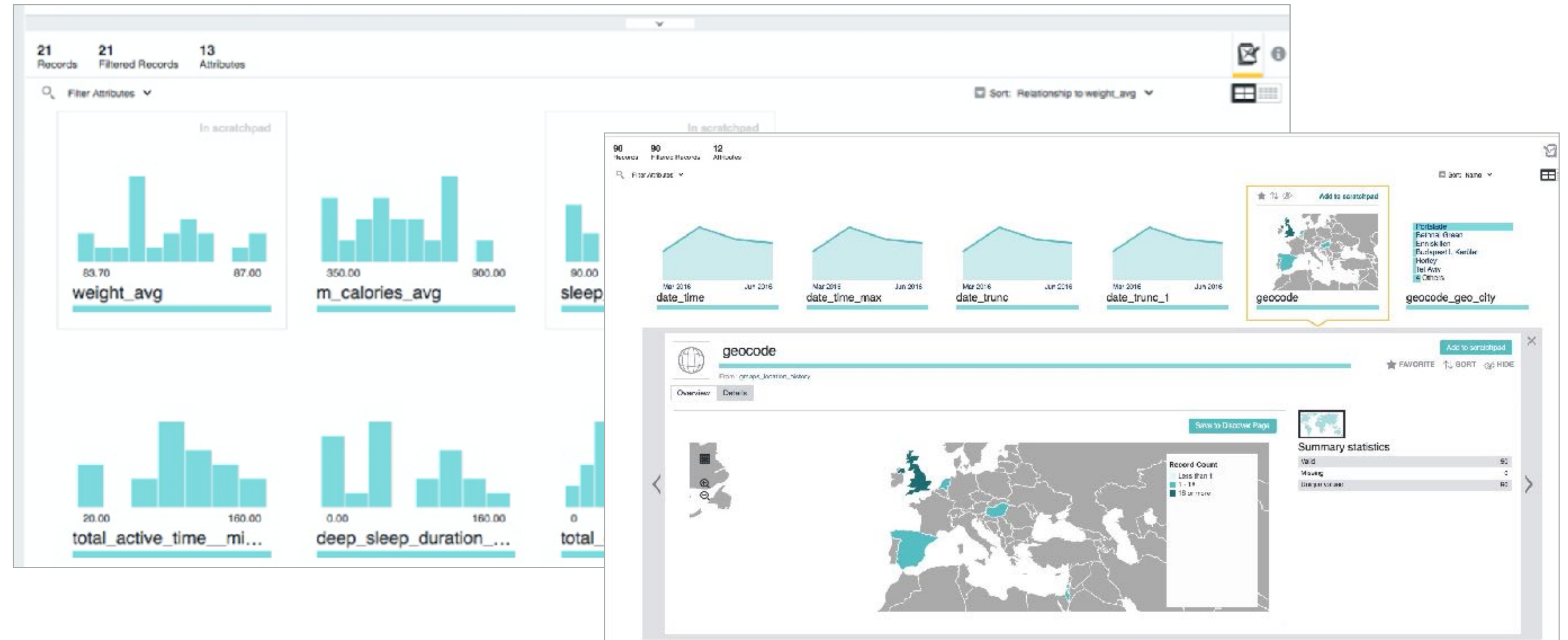
## MODELING AND INFERRING





# Perform Exploratory Analysis On Data

- Understand the “spread” of data using histograms
- Use box-plot charts to identify outliers and range of “usual” values
- Sort attributes by strongest correlation to a target attribute



# Transform (“Wrangle”) Data To Standardise & Tidy

- Initial row-wise preparation and transformation of data using Groovy transformations

58587  
RecordsFiltered RecordsAttributes

BasicConvertAdvancedShaping

Select one or more attributes below to create transformations

Filter Attributes

#	cycling_duration_h...	#	Distance KM_sum	#	Energy Kcal_sum
0		0		0	
0		0		0	
0		0		0	
0		0		0	
0		0		0	
0		0		0	
0		0		0	
0		0		0	
0		0		0	
0		0.09		5	28.7833.4797222222
0		0		0	29.8366.1755555556
0		0		0	27.0323.9719444444

129 of 36612923  
Records SampledFiltered RecordsAttributes

BasicConvertAdvancedShaping

Use refinement state as a conditional statement

1toDate(toString(date), 'yyyyMMdd')

ShapingEditor

Filter rowsAggregateJoin

ExploreTransformDiscover

58587  
RecordsFiltered RecordsAttributes

BasicConvertAdvancedShaping

Use refinement state as a conditional statement

1(tag == 'sleep' ? duration : null)

allows you to remove rows from your data set based on the current set of selected refinements, and any parameters that you add here. Rows will be removed from the data set in this project but the original data set in will be unaffected.

☒ Enable automatic typeahead

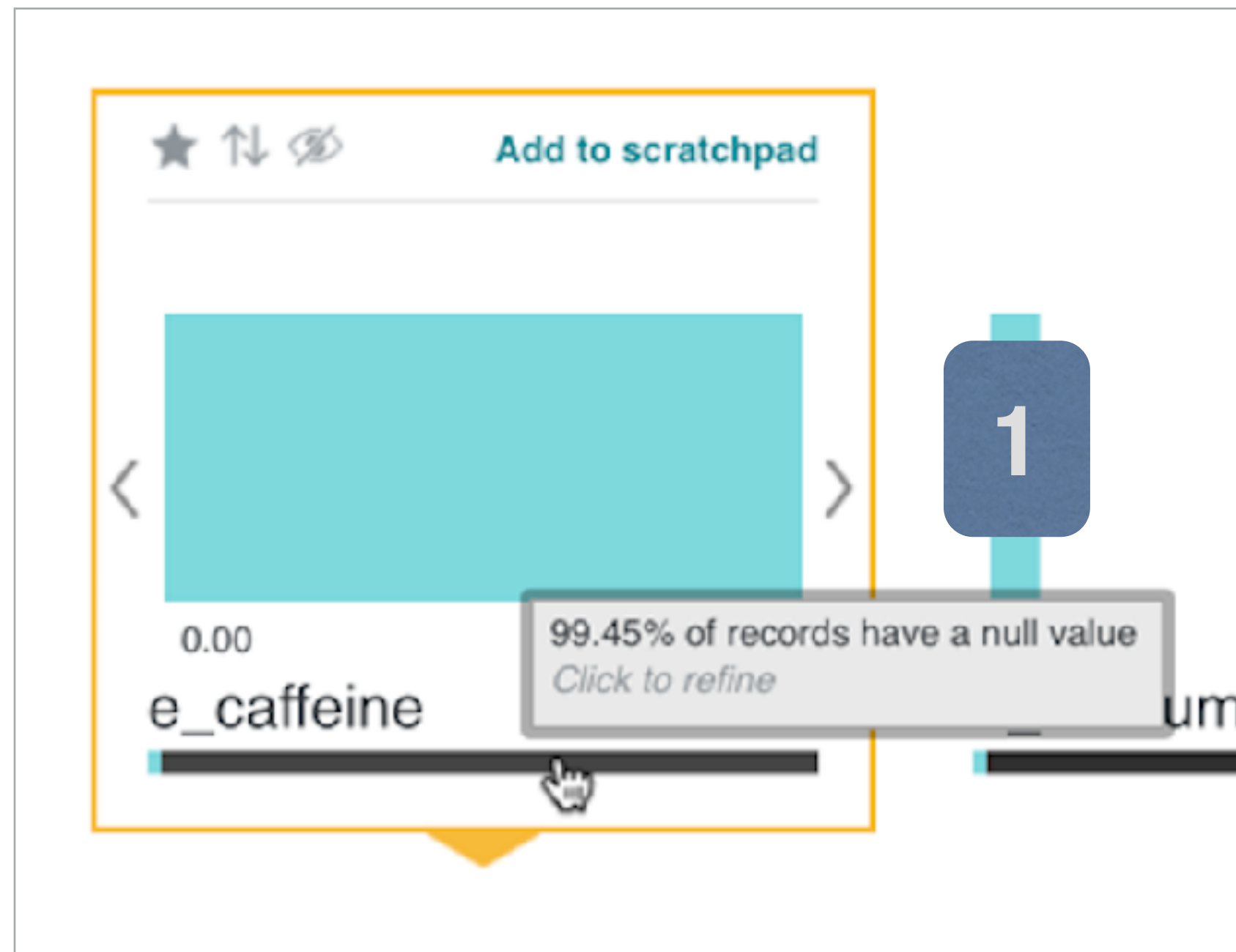
yyy-MM-dd HH:mm:ss z') && date\_dt <= toDate('2016-06-19 23:59:59 GMT', 'yyyy-MM-dd HH:mm:ss z'))

00:00 UTC	189.875
00:00 UTC	190.26
00:00 UTC	189.38
00:00 UTC	186.4
00:00 UTC	185.41
00:00 UTC	191.8
00:00 UTC	191.58
00:00 UTC	191.415



# Dealing With Missing Recordings In The Data

- Very typical with self-recorded healthcare and workout data
- Most machine-learning algorithms expect every attribute to have a value per row
- Self-recorded data is typically sporadically recorded, lots of gaps in data
- Need to decide what to do with columns of poorly populated values



A screenshot of a data management tool interface. The main table displays data for 'Fat Percentage...' and 'humidity\_avg'. A blue box with the number '2' is positioned over the first column. A blue box with the number '3' is positioned over the second column. A sidebar on the right contains various tools and settings. The top section shows '129 of 366 Records Sampled', '129 Filtered Records', and '23 Attributes'. Below this are tabs for 'Basic', 'Convert', 'Advanced', 'Shaping', and 'Editor'. The 'Filter rows' section shows a filter expression: `'yyyy-MM-dd HH:mm:ss z') && date_dt <= toDate('2016-06-19 23:59:59 GMT', 'yyyy-MM-dd HH:mm:ss z'))]`. The 'Manage null values' section shows options for 'Choose a null-handling technique', including 'Replace with Static Value', 'Most Frequent Value: 5', 'Attribute Mean: 7.33333333333333', and 'Null filling value:'. A checkbox 'Create null indicator attribute for m\_distance' is checked.



# Joining Wearables Data With Comms + Smart Devices

- Previous versions of BDD allowed you to create joins for views
  - Used in visualisations, equivalent to a SQL view i.e. SELECT only
- BDD 1.2.x allows you to add new joined attributes to data view, i.e. materialise
- In this instance, use to bring in data on emails, and on geolocation

[[ Sleep Level (H/M/L)	[[ Steps Intensity (H/M...	# Total Active Time (M...	# Total_Emails_Recei...	# weight
L	High	47		
H	Medium	16		
L	Medium	112		
L	Medium	50		
L	Medium	107		
L	Medium	114		
L	Low	10		
L	Medium	22		

BasicConvertAdvancedShapingEditor

Filter rowsAggregateJoin

Select the join key(s)  
Select one or more pairs of attributes to join the data sets

Jawbone UP Mark Rittman 2016 + Emails

Jawbone UP Mark Rittman 2016 a...

Filter...Show...

# bmr# body\_fat# date# date\_dt

Emails attributes

Filter...Date...

date\_fid

Selected Key(s)  
date\_dt = date\_fid

+  
Select Join Type  
Determine how the records will be filtered when creating the join

☒ Left Outer Join  
Includes all of the primary data set plus matching secondary set

☐ Inner Join  
Only records matching both primary and secondary data set

☐ Full Outer Join  
All records from both data sets

PreviewCancel

Jawbone UP Mark Rittman 2016 attributes

date_dt	# m_steps_3am	# n_quality	# n_count	# min_bg	# s_light
---------	---------------	-------------	-----------	----------	-----------



# Aggregate Data Up To The Week Level

- Only sensible option when looking at change in weight compared to prior period
  - Change compared to previous day too granular

BasicConvertAdvancedShapingEditor

Filter rowsAggregateJoin

Available attributes

1

Filter...  
# Number of Records  
# body\_fat  
# date\_dt  
# day\_of\_week  
# day of week type

Grouping attributes

week

Aggregated attributes

2

# m\_workout\_... (average)  
# m\_calories\_... (average)  
# Total Active ... (average)  
# Light Sleep ... (average)  
# Deep Sleep ... (average)

Available attributes preview

3

Select one or more source attributes to preview values here

Aggregation preview

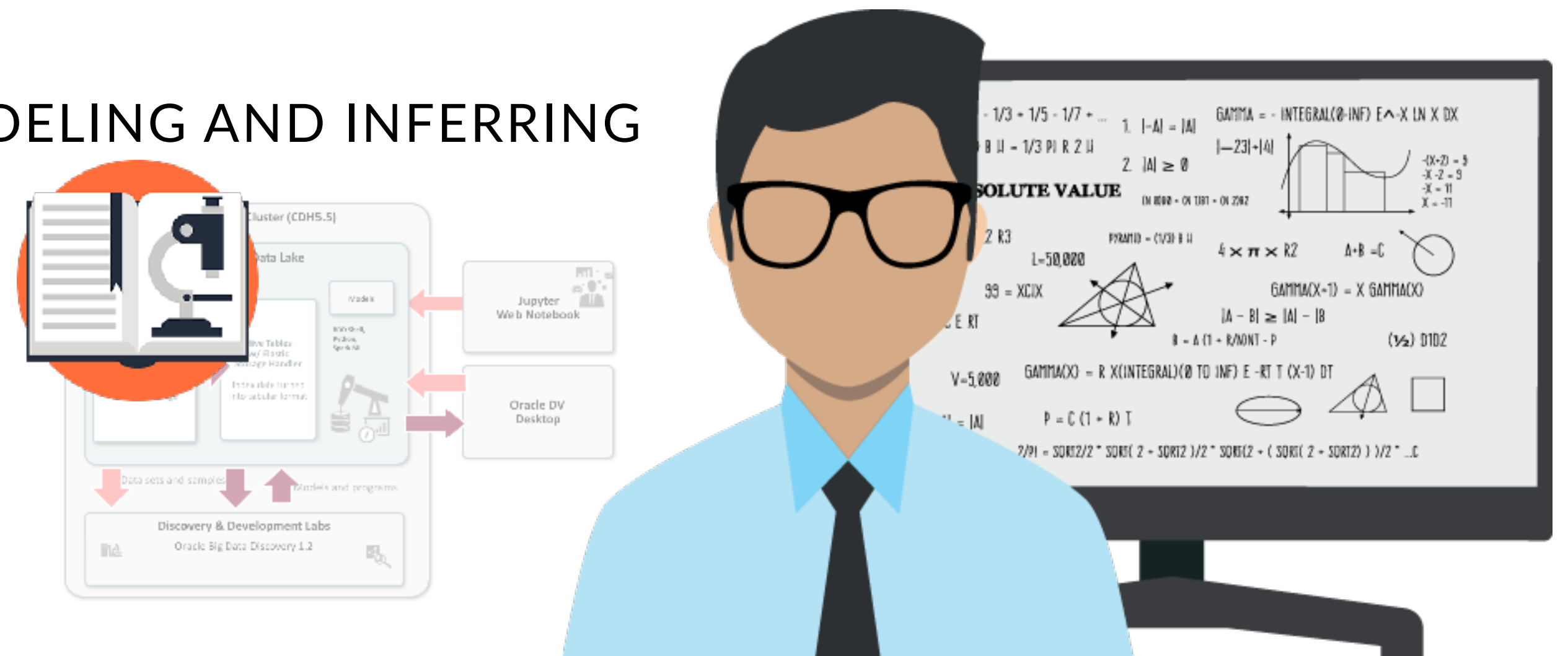
week	# m_workout_count_avg	# m_calories_avg	# Total Active Time (Mi...	# Light
2016-11	0	580.6	109.2	84.6
2016-12	0	678.2857142857143	110.57142857142857	99.57142857142857
2016-13	0	411.14285714285717	70.28571428571429	50.857142857142854
2016-14	0.2857142857142857	450.42857142857144	84.42857142857143	162.85714285714286

CancelPrAdd to

# Use of BDD Shell, Python Pandas + Jupyter Notebook

- Now we have the data organised into weekly reading rows, we now switch to **Python Pandas**
- Use this Python statistics and data visualisation library to **calculate w/o/w weight change**, and **identify most influential variable** (i.e. action, activity type I've recorded)
- Use BDD Shell to connect to BDD SDK from pySpark environment
- Work with BDD datasets as Spark dataframes
- Import and use Python Pandas and SparkML packages
- Shape and transform dataframes further if needed
- Use visualizations to understand correlations between variables
- Create linear regression ML model to identify most influential variable

## MODELING AND INFERRING





# Use BDD Shell API to Identify Main Dataset ID

```
In [1]: execfile('ipython/00-bdd-shell-init.py')
```

```
In [31]: dss = bc.datasets()
         dss.count
```

```
Out[31]: 76
```

```
In [32]: for ds in dss:
         print('Name: %s\t'
```

```
Name: ifttt_comms_ema
Name: ifttt_comms_tw
e8c8240
Name: ifttt_health_ev
Name: Combined Health
Name: Combined Health
7b-df15d8580b29
Name: Jawbone UP Mark
Name: Combined Health
70-ac05e7699f50
Name: ifttt_comms_eve
Name: smarthings_log_
Name: Combined Aggreg
eec6914
Name: Combined Health
Key:default_edp_4f930
Name: Jawbone UP Mark
Name: ifttt_health_ev
Name: Heart Rate - Ma
Name: ifttt_comms_fac
73f6e60
```

```
In [34]: ds = dss.dataset('default_edp_07c07ea5-891e-40ae-b2ca-b6d85f68b9e1')
```

```
import json
print json.dumps(ds.properties(),indent=2,sort_keys=True)

{
  "accessType": "public_default",
  "attributeCount": "14",
  "attributeDisplayNames": "workout_duration__mins__avg",
  "attributeKeys": "workout_duration__mins__avg",
  "attributeNotes": null,
  "attributeSemanticTypes": null,
  "attributeTags": null,
  "authorizedGroup": null,
  "authorizedReadGroup": null,
  "authorizedReadUser": "10098",
  "authorizedUser": "10098",
  "collectionIdToBeReplaced": null,
  "collectionKey": "default_edp_07c07ea5-891e-40ae-b2ca-b6d85f68b9e1",
  "creationTime": "2016-06-23T22:50:52.787Z",
  "curated": "false",
  "databaseKey": "default_edp_07c07ea5-891e-40ae-b2ca-b6d85f68b9e1",
  "dateTimePresent": "false",
```



# Use Python PANDAS to Calculate % CHG W/w

```
In [198]: pandas_df['weight_pct_chg'] = (pandas_df.weight_avg.pct_change
```

```
In [199]: pandas_df.head()
```

```
Out[199]:
```

	body_fat_avg	deep_sleep_duration_hours_avg	hour_last_email_sent_a
4	29.500000	82.500000	11
16	30.571429	43.857143	16
7	30.285714	0.714286	14
14	29.000000	105.142857	2
5	29.000000	45.666667	1

```
In [200]: pandas_df.shape
```

```
Out[200]: (21, 15)
```

```
In [161]: spark_df2 = sqlContext.createDataFrame(pandas_df)
```

```
In [162]: spark_df2.write.saveAsTable('default.health_data_weekly_with_pct_chg')
```

```
In [163]: spark_df2.printSchema()
```

```
root
|-- body_fat_avg: double (nullable = true)
|-- deep_sleep_duration_hours_avg: double (nullable = true)
|-- hour_last_email_sent_avg: long (nullable = true)
|-- light_sleep_duration_hours_avg: double (nullable = true)
|-- m_calories_avg: double (nullable = true)
|-- m_workout_count_avg: double (nullable = true)
|-- number_of_emails_sent_avg: long (nullable = true)
|-- sleep_duration_hours_avg: double (nullable = true)
|-- total_active_time_mins_avg: double (nullable = true)
|-- total_emails_received_sent_avg: double (nullable = true)
|-- week: string (nullable = true)
|-- weight_avg: double (nullable = true)
|-- workout_duration_mins_avg: double (nullable = true)
|-- PRIMARY_KEY: string (nullable = true)
|-- weight_pct_chg: double (nullable = true)
```

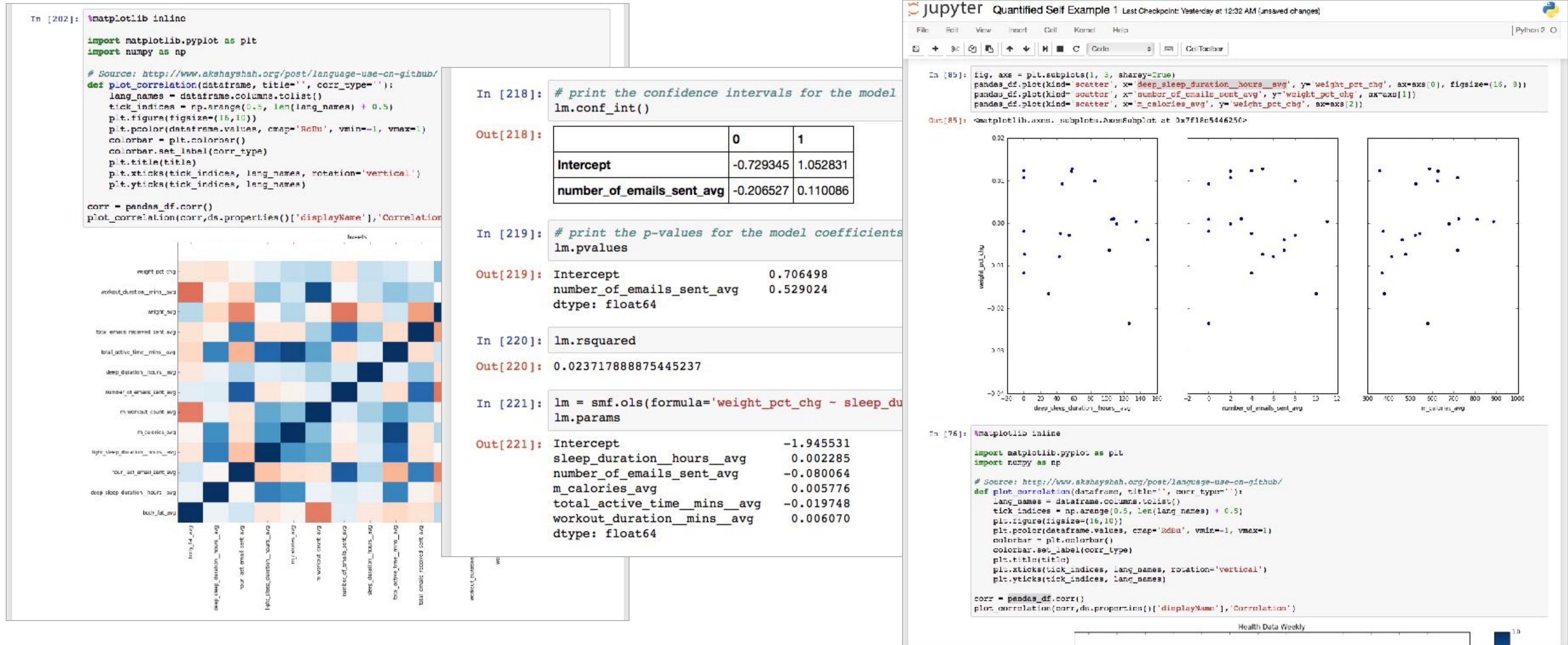
```
In [178]: pandas_df3 = spark_df2.toPandas()
pandas_df3 = pandas_df3.sort_values(['week'], ascending=[True])
pandas_df3.head()
```

```
Out[178]:
```

	body_fat_avg	deep_sleep_duration_hours_avg	hour_last_email_sent_avg	light_sleep_duration_hours_avg	nr
16	30.571429	43.857143	16	103.428571	5
7	30.285714	0.714286	14	1.428571	4
14	29.000000	105.142857	2	144.571429	8
5	29.000000	45.666667	1	43.833333	5
18	29.400000	126.200000	5	84.600000	5



# Identify Correlations Between Attributes





# Use Linear Regression on BDD Dataset via Python

- To answer the question - which metric is the most influential when it comes to weight change?

```
In [214]: lm = smf.ols(formula='weight_pct_chg ~ number_of_emails_sent_avg', data=pandas_df3).fit()

# print the coefficients
lm.params

X_new = pd.DataFrame({'number_of_emails_sent_avg': [50]})
X_new.head()
lm.predict(X_new)
```

```
Out[214]: array([-2.24926045])
```

```
In [215]: X_new = pd.DataFrame({'number_of_emails_sent_avg': [pandas_df3.number_of_emails_sent_avg,
X_new.head()

Out[215]:
```

	number_of_emails_sent_avg
0	0
1	11

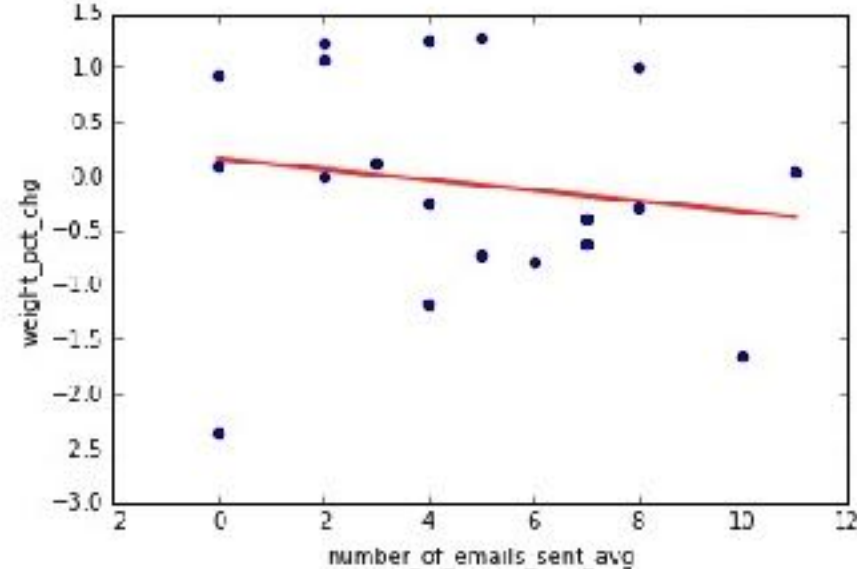
```
In [216]: preds = lm.predict(X_new)
preds

Out[216]: array([ 0.16174297, -0.36867955])
```

```
In [217]: # First, plot the observed data
pandas_df3.plot(kind='scatter', x='number_of_emails_sent_avg', y='weight_pct_chg')

# then, plot the least squares line
plt.plot(X_new, preds, c='red', linewidth=2)

Out[217]: <matplotlib.lines.Line2D at 0x7f18ac3dd110>
```



The scatter plot shows the relationship between the average number of emails sent and the percentage change in weight. The x-axis, 'number\_of\_emails\_sent\_avg', ranges from 0 to 12. The y-axis, 'weight\_pct\_chg', ranges from -3.0 to 1.5. Data points are represented by blue dots, and a solid red line indicates the linear regression fit. The data points are scattered around the line, which has a slight negative slope.

```
In [218]: # print the confidence intervals for the model coefficients
lm.conf_int()
```

```
Out[218]:
```

	0	1
Intercept	-0.729345	1.052831
number_of_emails_sent_avg	-0.206527	0.110086

```
In [219]: # print the p-values for the model coefficients
lm.pvalues

Out[219]: Intercept                0.706498
number_of_emails_sent_avg         0.529024
dtype: float64
```

```
In [220]: lm.rsquared

Out[220]: 0.023717888875445237
```

```
In [221]: lm = smf.ols(formula='weight_pct_chg ~ sleep_duration_hours_avg + number_of_emails_sent_avg +
lm.params

Out[221]: Intercept                -1.945531
sleep_duration_hours_avg           0.002285
number_of_emails_sent_avg         -0.080064
m_calories_avg                     0.005776
total_active_time_mins_avg        -0.019748
workout_duration_mins_avg          0.006070
dtype: float64
```



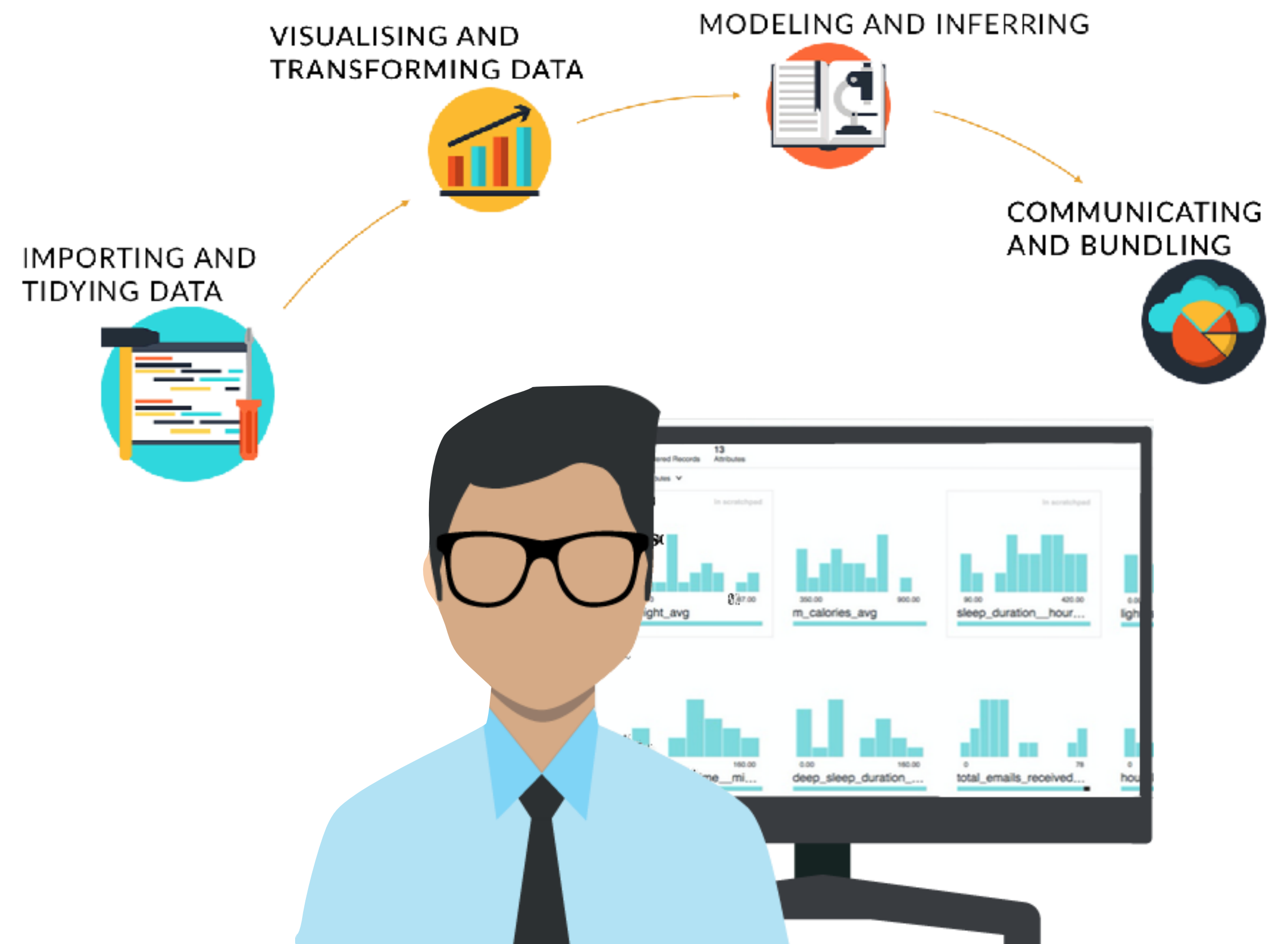
# The Answer? ... Hours of Sleep Most Influential Activity

- Most influential variable/attribute in my weight / loss gain is hours of sleep per week
  - The more sleep i get, the more likely I am to exercise, walk somewhere, eat properly and lose weight
  - Weeks where less sleep recorded led to eating more carbs, driving rather than walking, weight gain
- Environment (internal, external) had less influence this time, but influential variables were:
  - Comms activity - emails sent late night, Facebook likes, Instagram photos - proxy for working/play
  - Heat/Temperature inside house - indicates warm/cold outside, driver of exercise activity
  - Geo-location - am I on holiday? At work that week?
  - Diet? Although fairly constant over perio



# How Did Oracle BDD Help With This Project?

- Visual, graphic way to understand shape, data distribution and outliers/completeness
- Simple user-driven graphical tools for data tidying and transformation
- Join and aggregate datasets to get to one row of data = set of weekly readings
- Enrich and bring in additional datasets to add comms and environment activity data
- Enable use of wide range of industry-standard stats and ML libraries on final dataset







# PRACTICAL IOT DEVELOPMENT USING ORACLE BIG DATA AND ORACLE DV ...AND A WIFI KETTLE

Mark Rittman, Oracle ACE Director & Independent Analyst  
MJR Analytics Ltd (<http://www.mjr-analytics.com>)

BIWA SUMMIT 2017, SAN FRANCISCO