Taking Google Analytics to the next level

Johann de Boer, digital analytics consultant and trainer

Johann de Boer

- Digital Analytics consultant and trainer
- Loves Data
- Master of Analytics (Marketing)
- Bachelor of Engineering (Computer Systems)
- From user experience and accessibility
- To analytics implementation and insights

Purpose of this talk

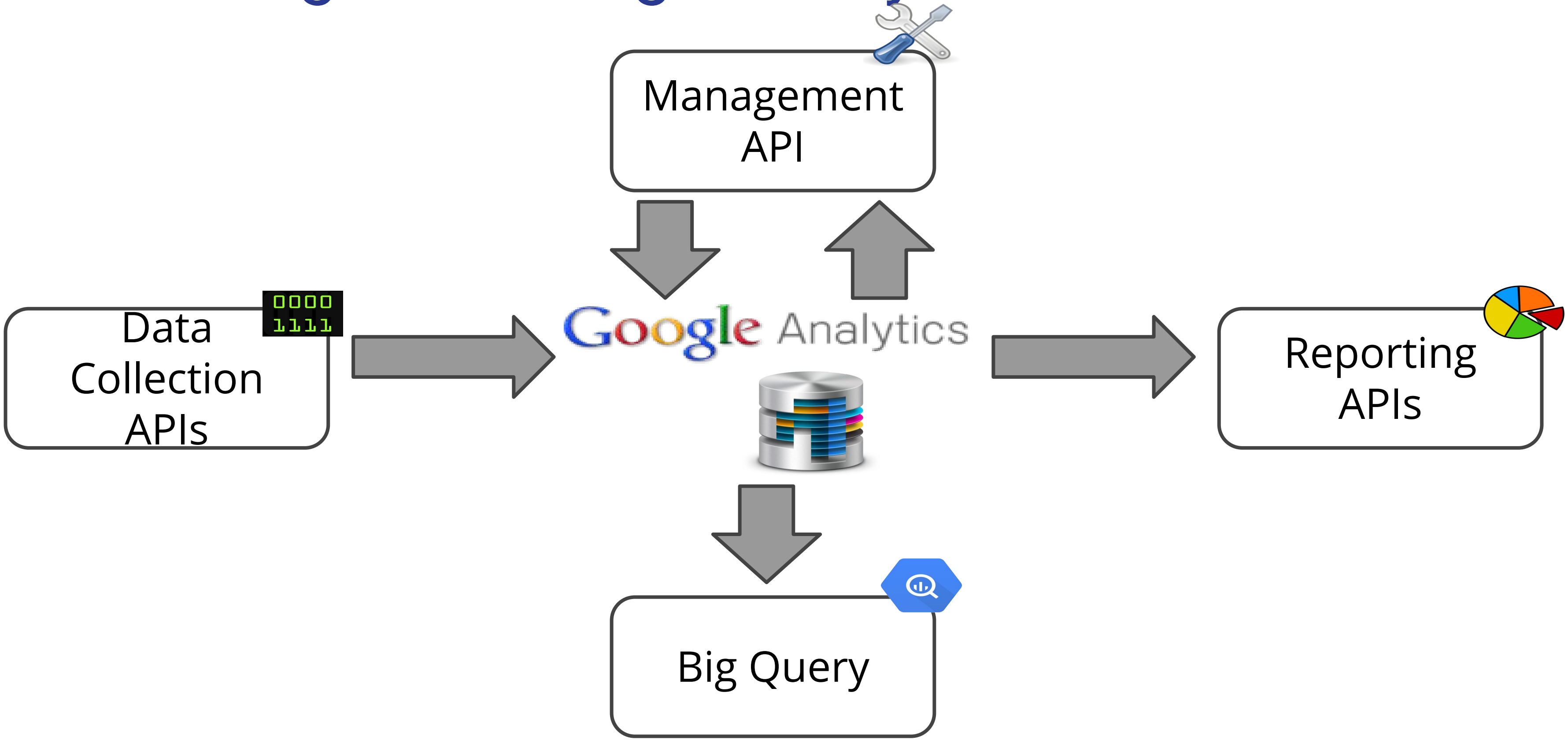
Encourage web data analysts to move to R and away from Excel!

Help those wanting to learn R to get started.

What is R?

- open-source programming language
- purpose built for data analysis
- World domination!

Interfacing with Google Analytics



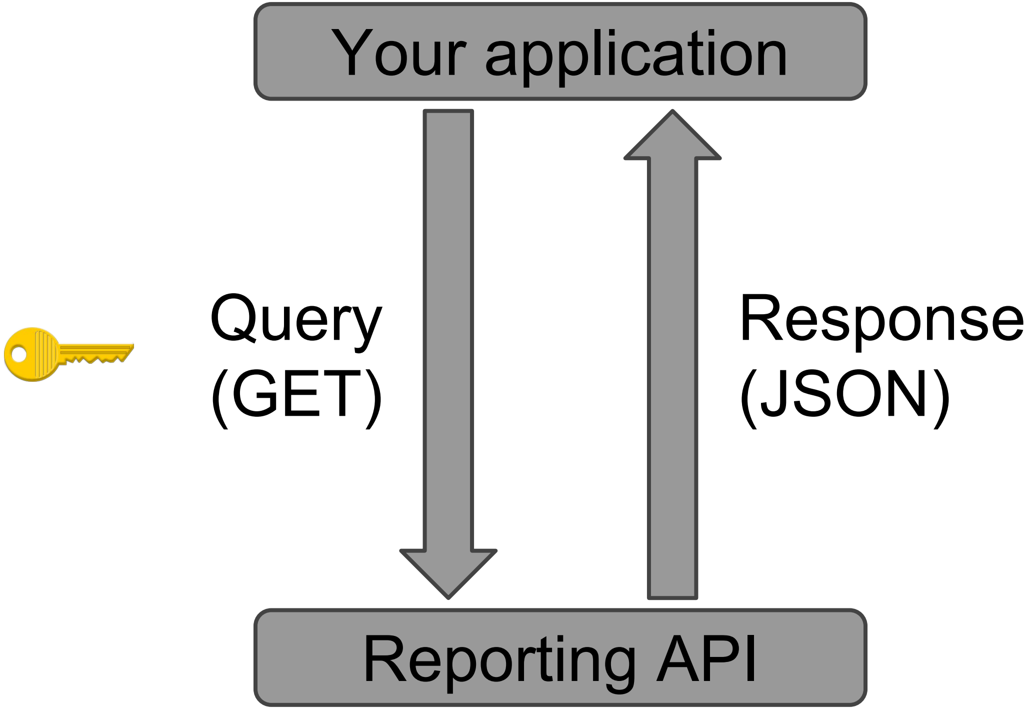
GA API platform overview

- **Collection APIs**
  - Collect and send data to Google Analytics for storage and processing
- Management API
  - Configure Google Analytics accounts
  - Import off-line data
- Reporting APIs
  - Query and extract processed data from Google Analytics for reporting

Why use the Reporting APIs?

- Surpass the limits of the GA user interface
- Automate repetitive tasks
- Perform more powerful data analysis
- Easily reproduce and reuse analyses
- Combine with other data sources
- Develop innovative tools
- Feed into other tools and applications

API concepts



Your application

Query
(GET)

Response
(JSON)

Reporting API

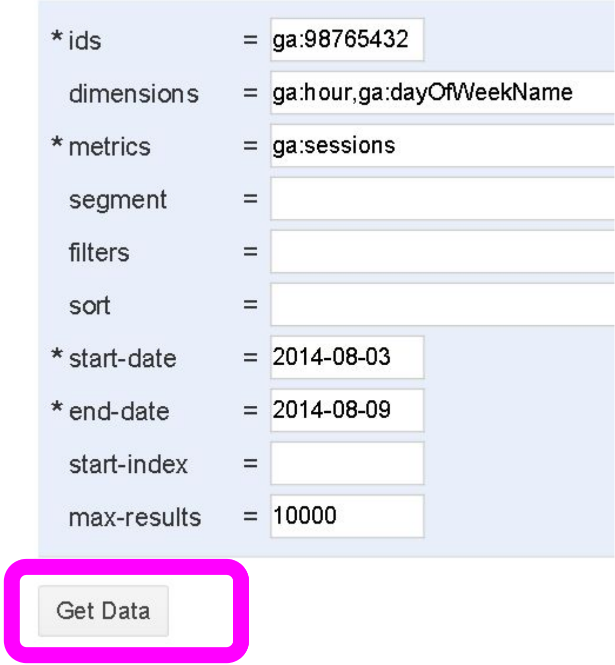
API Query

```
Authorization: Bearer {oauth2-token}

GET https://www.googleapis.com/analytics/v3/data/ga?
ids=ga:98765432&
start-date=2014-01-01&
end-date=2014-01-31&
metrics=ga:sessions&
dimensions=ga:landingPagePath&
sort=-ga:sessions&
filters=ga:medium==organic&
segment=users::condition::ga:goalCompletionsAll>0&
samplingLevel=HIGHER_PRECISION&
start-index=1&
max-results=10000
```

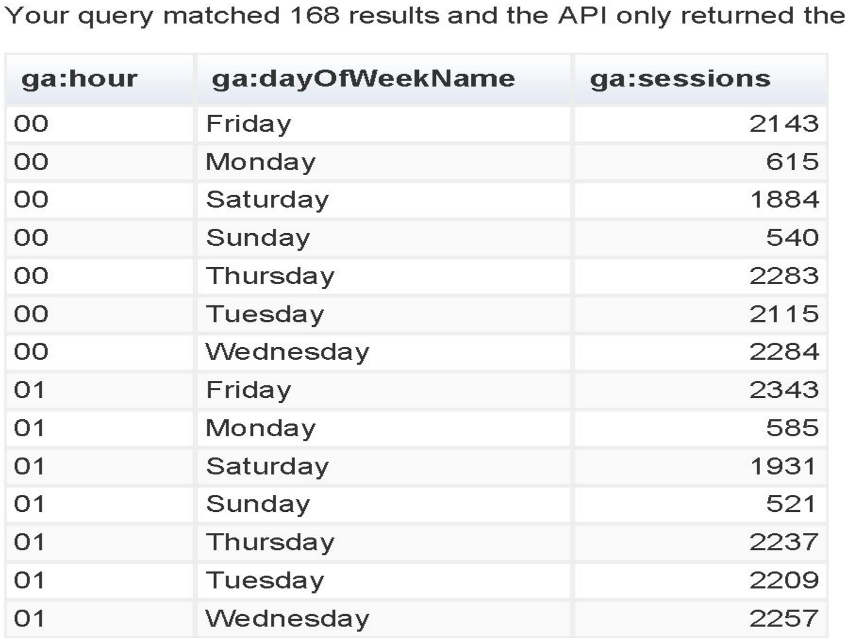
API Response

```
{
  "kind": "analytics#gaData",
  "id": string,
  "selfLink": string,
  "containsSampledData": boolean,
  "query": {
    ...
  },
  "itemsPerPage": integer,
  "totalResults": integer,
  "previousLink": string,
  "nextLink": string,
  "profileInfo": {
    ...
  },
  "columnHeaders": [
    {
      "name": string,
      "columnType": string,
      "dataType": string
    }, ...
  ],
  "rows": [
    [string, ...], ...
  ],
  "sampleSize": string,
  "sampleSpace": string,
  "totalsForAllResults": [{
    metricName: string,
    ...
  }]
}
```

Query Explorer



* ids	=	ga:98765432
dimensions	=	ga:hour,ga:dayOfWeekName
* metrics	=	ga:sessions
segment	=	
filters	=	
sort	=	
* start-date	=	2014-08-03
* end-date	=	2014-08-09
start-index	=	
max-results	=	10000



Your query matched 168 results and the API only returned the

ga:hour	ga:dayOfWeekName	ga:sessions
00	Friday	2143
00	Monday	615
00	Saturday	1884
00	Sunday	540
00	Thursday	2283
00	Tuesday	2115
00	Wednesday	2284
01	Friday	2343
01	Monday	585
01	Saturday	1931
01	Sunday	521
01	Thursday	2237
01	Tuesday	2209
01	Wednesday	2257

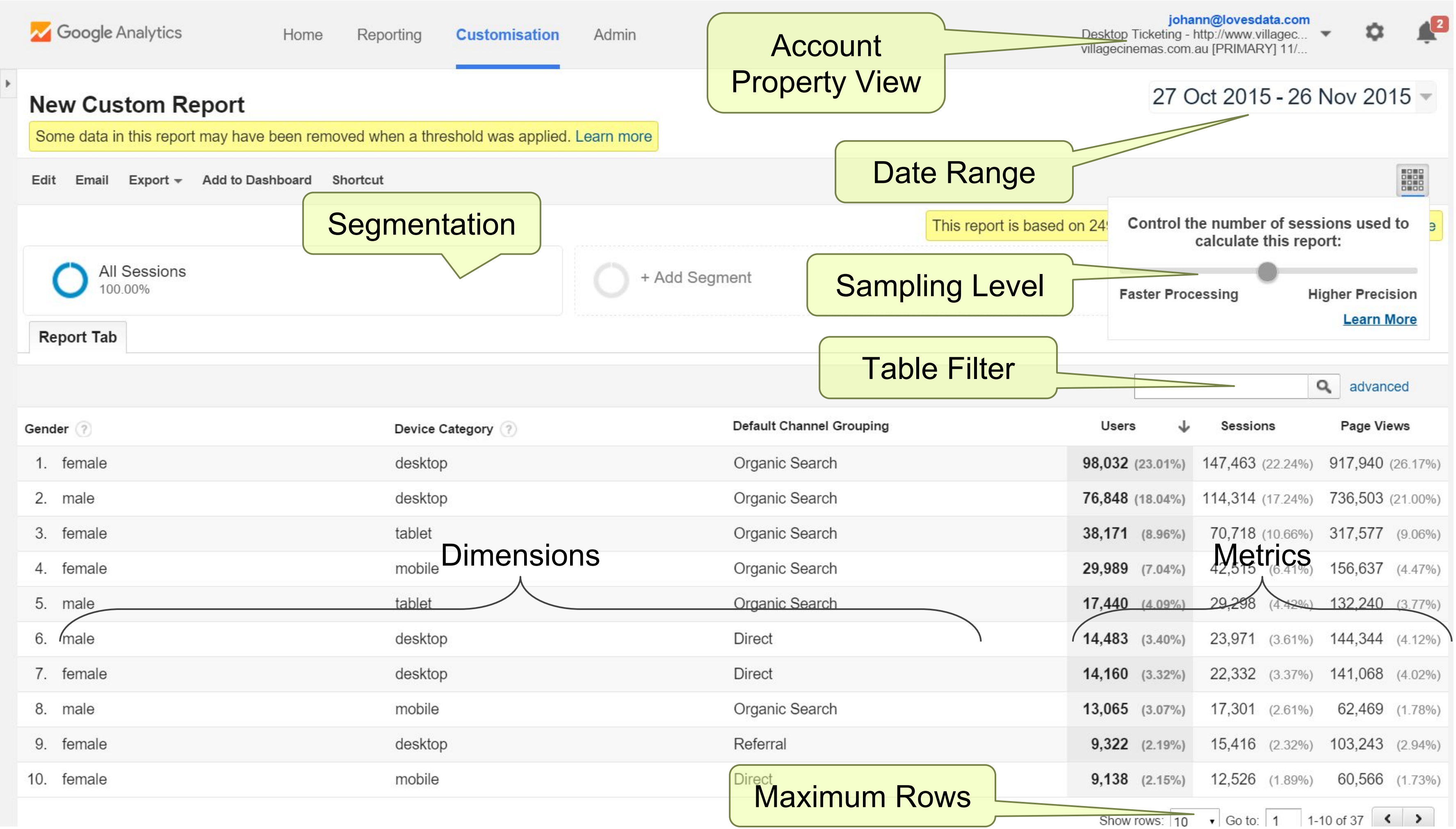
Expressing yourself with GA

Explaining the steps you followed to put together a data analysis report:

- precisely and accurately
- using natural language but without ambiguity
- so that others can reproduce that report without error, and
- can reuse your steps to save time in pursuing other analysis ideas

Google Analytics Reporting Query

1. **Account Property View**
2. **Date Range**: **Start Date** and **End Date**
3. **Variables** to form rows and columns: **Dimensions** and **Metrics**
4. **Maximum Rows** and which variables to **Sort By**
5. **Table Filters** to remove unwanted rows
6. One or more **Segments** to compare



Google Analytics Home Reporting **Customisation** Admin

johann@lovesdata.com
Desktop Ticketing - http://www.villagec...
villagecinemas.com.au [PRIMARY] 11/...

New Custom Report

Some data in this report may have been removed when a threshold was applied. Learn more

27 Oct 2015 - 26 Nov 2015

Edit Email Export Add to Dashboard Shortcut

This report is based on 24...

Control the number of sessions used to calculate this report:

Faster Processing Higher Precision

Learn More

All Sessions
100.00%

+ Add Segment

Report Tab

advanced

	Gender	Device Category	Default Channel Grouping	Users ↓	Sessions	Page Views
1.	female	desktop	Organic Search	98,032 (23.01%)	147,463 (22.24%)	917,940 (26.17%)
2.	male	desktop	Organic Search	76,848 (18.04%)	114,314 (17.24%)	736,503 (21.00%)
3.	female	tablet	Organic Search	38,171 (8.96%)	70,718 (10.66%)	317,577 (9.06%)
4.	female	mobile	Organic Search	29,989 (7.04%)	42,515 (6.41%)	156,637 (4.47%)
5.	male	tablet	Organic Search	17,440 (4.09%)	29,298 (4.42%)	132,240 (3.77%)
6.	male	desktop	Direct	14,483 (3.40%)	23,971 (3.61%)	144,344 (4.12%)
7.	female	desktop	Direct	14,160 (3.32%)	22,332 (3.37%)	141,068 (4.02%)
8.	male	mobile	Organic Search	13,065 (3.07%)	17,301 (2.61%)	62,469 (1.78%)
9.	female	desktop	Referral	9,322 (2.19%)	15,416 (2.32%)	103,243 (2.94%)
10.	female	mobile	Direct	9,138 (2.15%)	12,526 (1.89%)	60,566 (1.73%)

Show rows: 10 Go to: 1 1-10 of 37

Account Property View

Date Range

Segmentation

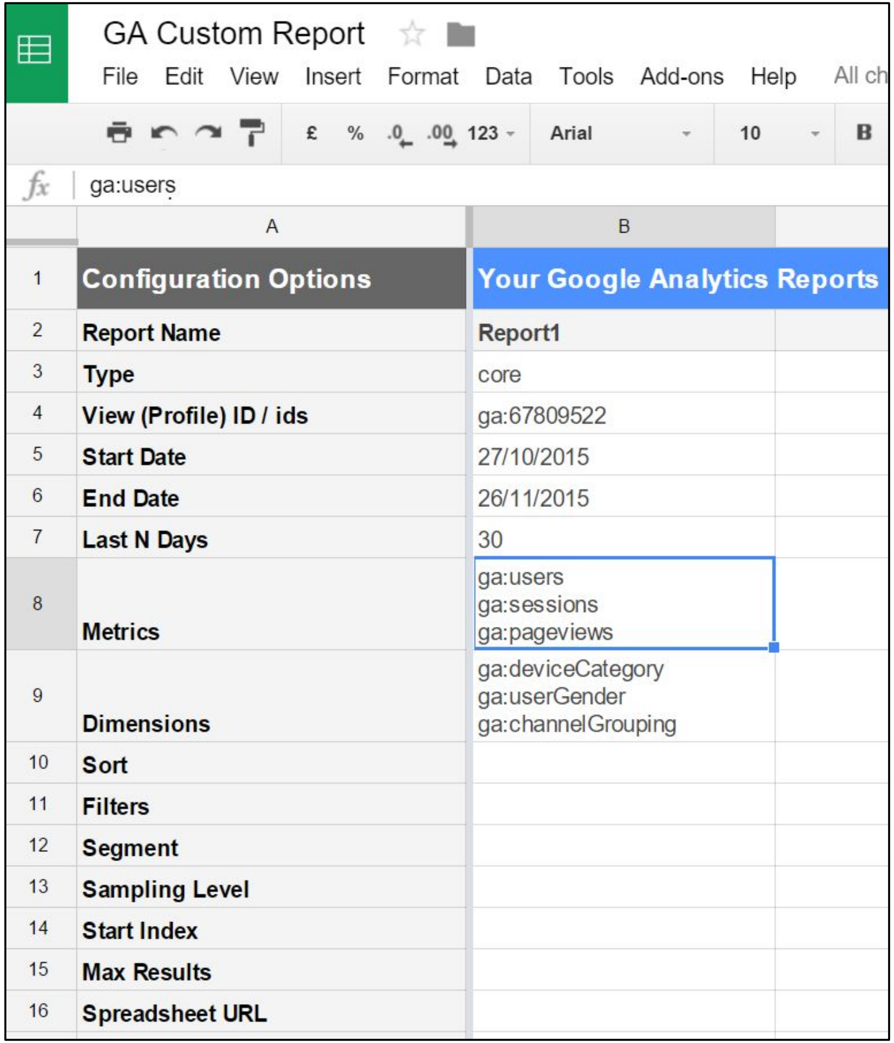
Sampling Level

Table Filter

Dimensions

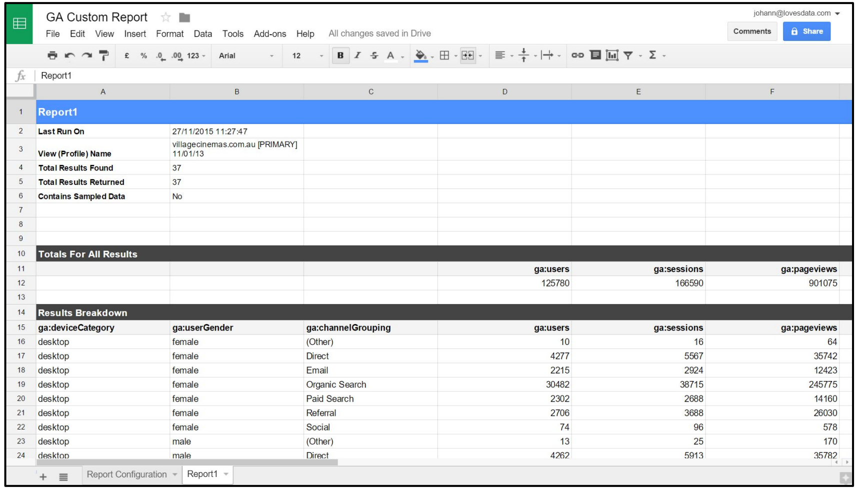
Metrics

Maximum Rows


GA Custom Report
File Edit View Insert Format Data Tools Add-ons Help All ch
£ % .0 .00 123 Arial 10 B

fx ga:users

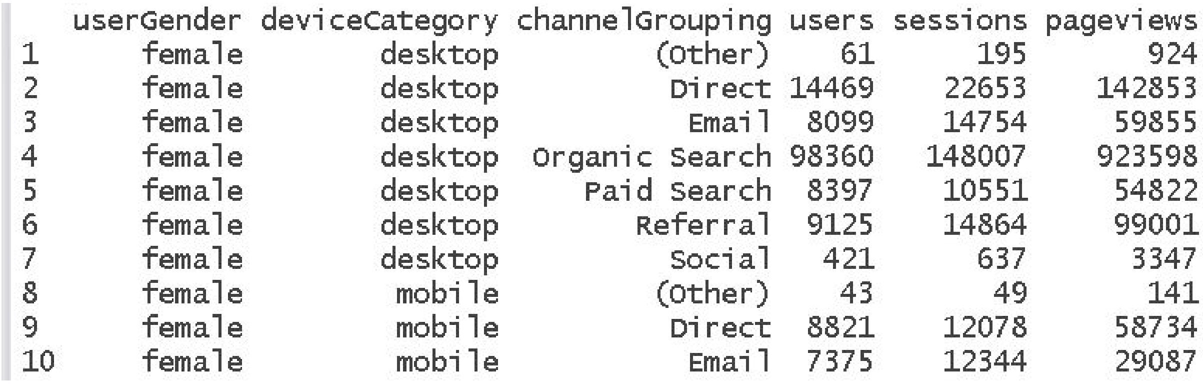
	A	B
1	**Configuration Options**	**Your Google Analytics Reports**
2	**Report Name**	Report1
3	**Type**	core
4	**View (Profile) ID / ids**	ga:67809522
5	**Start Date**	27/10/2015
6	**End Date**	26/11/2015
7	**Last N Days**	30
8	**Metrics**	ga:users ga:sessions ga:pageviews
9	**Dimensions**	ga:deviceCategory ga:userGender ga:channelGrouping
10	**Sort**	
11	**Filters**	
12	**Segment**	
13	**Sampling Level**	
14	**Start Index**	
15	**Max Results**	
16	**Spreadsheet URL**	


GA Custom Report
File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive
johann@lovesdata.com
Comments
Share
£ % .0 .00 123 Arial 12 B I S A
Report Configuration
Report1

fx Report1

	A	B	C	D	E	F
1	**Report1**					
2	**Last Run On**	27/11/2015 11:27:47				
3	**View (Profile) Name**	villagecinemas.com.au [PRIMARY] 11/01/13				
4	**Total Results Found**	37				
5	**Total Results Returned**	37				
6	**Contains Sampled Data**	No				
7						
8						
9						
10	**Totals For All Results**					
11				**ga:users**	**ga:sessions**	**ga:pageviews**
12				125780	166590	901075
13						
14	**Results Breakdown**					
15	**ga:deviceCategory**	**ga:userGender**	**ga:channelGrouping**	**ga:users**	**ga:sessions**	**ga:pageviews**
16	desktop	female	(Other)	10	16	64
17	desktop	female	Direct	4277	5567	35742
18	desktop	female	Email	2215	2924	12423
19	desktop	female	Organic Search	30482	38715	245775
20	desktop	female	Paid Search	2302	2688	14160
21	desktop	female	Referral	2706	3688	26030
22	desktop	female	Social	74	96	578
23	desktop	male	(Other)	13	25	170
24	desktop	male	Direct	4262	5913	35782

```r
library(ganalytics)
my_creds <- GoogleApiCreds("johann@lovesdata.com")

q <- GaQuery(view = 67809522, creds = my_creds)

DateRange(q) <- c("2015-10-27", "2015-11-26")
Dimensions(q) <- c("userGender", "deviceCategory", "channelGroup")
Metrics(q) <- c("users", "sessions", "pageviews")

GetGaData(q)
```



```
   userGender deviceCategory channelGrouping users sessions pageviews
1      female        desktop         (Other)    61      195       924
2      female        desktop          Direct 14469    22653    142853
3      female        desktop           Email  8099    14754     59855
4      female        desktop  Organic Search 98360   148007    923598
5      female        desktop     Paid Search  8397    10551     54822
6      female        desktop        Referral  9125    14864     99001
7      female        desktop          Social   421      637      3347
8      female         mobile         (Other)    43       49       141
9      female         mobile          Direct  8821    12078     58734
10     female         mobile           Email  7375    12344     29087
```

Conditions for Table Filters and Segments

Conditions are expressed as:

```
~ <variable> <comparator> <operand>
```

Examples:

```
~ UserGender == 'female'
~ Bounces > 0
~ Hour in 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
```

Using conditions to form expressions

Combine one or more conditions with **AND**, **OR**

Define **User** and **Session** level segments

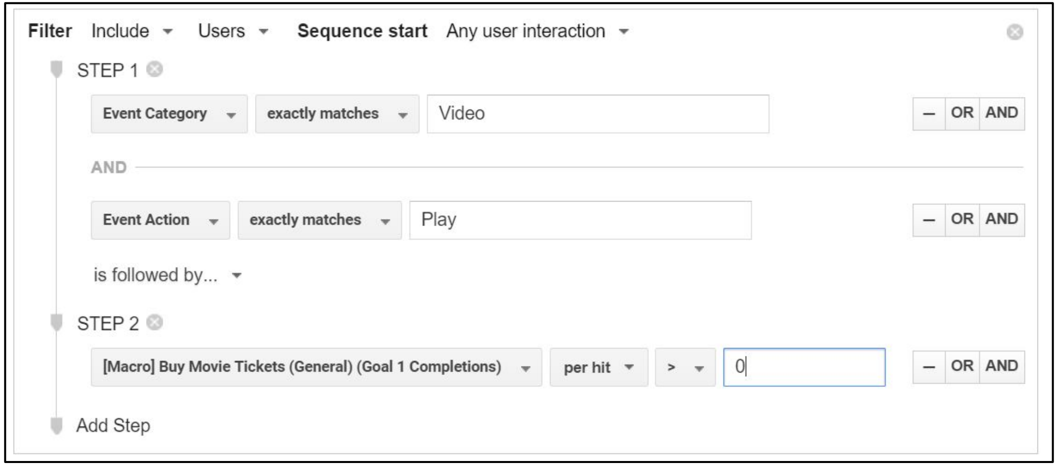
Define **Sequences** used for segmentation

- **First**, **Then**, **Later**

Scope conditions to **hit**, **session** or **user** level

Sequence segments

```
played_video <-
  Expr(~EventCategory == "Video") &
  Expr(~EventAction == "Play")

purchased_ticket <-
  PerHit(
    Expr(~goal1completions > 0)
  )

journey <- PerUser(Sequence(
  played_video,
  Then(purchased_ticket)
))
```



Filter Include Users Sequence start Any user interaction

STEP 1

Event Category exactly matches Video

AND

Event Action exactly matches Play

is followed by...

STEP 2

[Macro] Buy Movie Tickets (General) (Goal 1 Completions) per hit > 0

Add Step

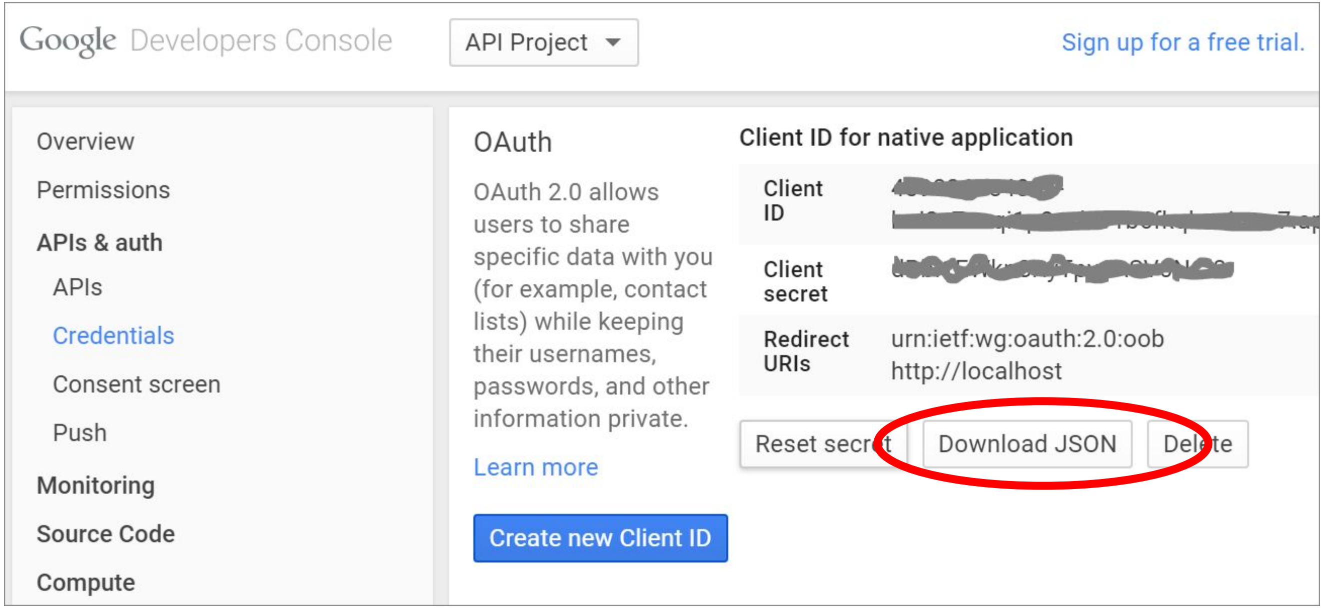
```
users::sequence::ga:eventCategory==Video;ga:
eventAction==Play;->perHit::ga:
goal1Completions>0
```

ganalytics for R

Goals of the ‘ganalytics’ R package

- Access to all Google Analytics features
- Easy to use, familiar to R users and in GA lingo
- Abstract away from API technicalities
- Interactively build and manipulate queries
- Detect and correct query definition errors
- Integrate with R classes, functions and other packages

Getting started



OAuth

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.

Learn more

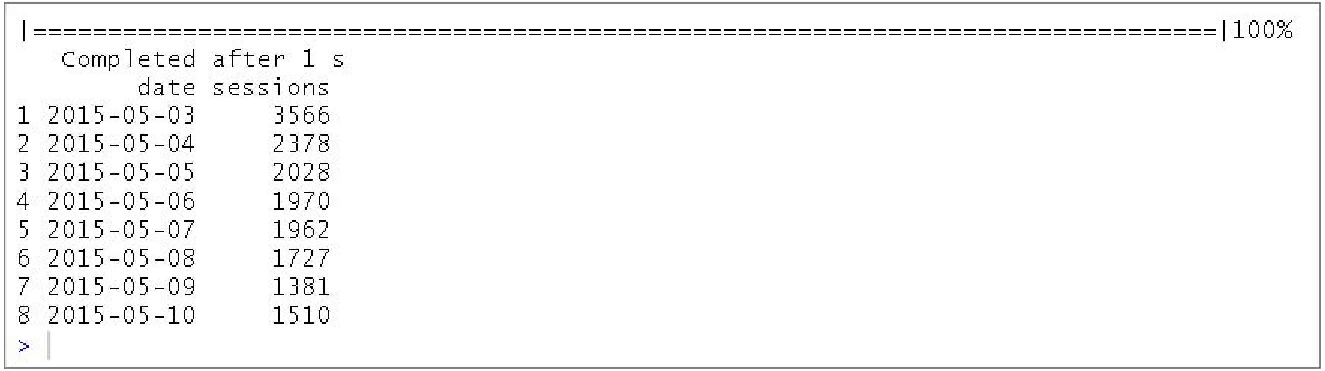
Client ID for native application

Client ID	[redacted]
Client secret	[redacted]
Redirect URIs	urn:ietf:wg:oauth:2.0:oob http://localhost

Installation and “Hello World!”

```
devtools::install_github("jdeboer/ganalytics")

library(ganalytics)

creds <- GoogleApiCreds(
 "you@domain.com", # Your Google username
 "client_secret.json" # From Google APIs console
)

# Default view of first account and property
view <- NA

query <- GaQuery(view, creds)

GetGaData(query)
```

“Hello World!” of ganalytics



```
|===============================================================================|100%
   Completed after 1 s
        date sessions
1 2015-05-03     3566
2 2015-05-04     2378
3 2015-05-05     2028
4 2015-05-06     1970
5 2015-05-07     1962
6 2015-05-08     1727
7 2015-05-09     1381
8 2015-05-10     1510
> 
```

Defining your query and getting data into R with ganalytics

Example query

```r
query <- GaQuery(
  view = 94605596,
  startDate = "2015-01-01", endDate = "2015-01-31",
  metrics = c("users", "sessions", "pageviews"),
  dimensions = c("deviceCategory", "dayOfWeekName"),
  filter = Expr(~deviceCategory %matches% "mobile|tablet"),
  segment = Expr(~country == "Australia")
)

response <- GetGaData(query)
```

Example query response

```
   deviceCategory dayOfWeekName users sessions pageviews
1          mobile        Friday   429      500       816
2          mobile        Monday   273      329       466
3          mobile      Saturday   339      399       640
4          mobile        Sunday   256      301       487
5          mobile      Thursday   420      475       689
6          mobile       Tuesday   314      358       542
7          mobile     Wednesday   304      357       571
8          tablet        Friday   209      235       424
9          tablet        Monday   160      190       355
10         tablet      Saturday   195      225       435
11         tablet        Sunday   128      151       334
12         tablet      Thursday   217      266       455
13         tablet       Tuesday   169      194       374
14         tablet     Wednesday   161      178       356
```

Conditional expressions

```
expr1 <- Expr(~variable comparator value)
expr2...

expr3 <- expr1 | !expr2
expr5 <- expr3 & expr4
expr6 <- xor(expr1, expr2)
```

From R to an API call

```r
e1 <- Expr(~keyword %starts_with% 'buy')
e2 <- Expr(~city %matches% '^(Sydney|Melbourne)$')
e3 <- Expr(~deviceCategory == 'tablet')

e4 <- e1 & (e2 | e3)

as(e4, 'character')
```

```
# ga:keyword=@buy;ga:city=~^(sydney|melbourne)$,ga:deviceCategory==tablet
```

How does traffic from desktop, mobile and tablet users change throughout the day and over the week?

Average number of visits per hour and day – split by desktop, mobile and tablet

Step 1: Define the query

```
query <- GaQuery(view = '98765432')

DateRange(query) <- c("2014-01-01", "2014-12-31")

Dimensions(query) <- c(
  "deviceCategory", "dayOfWeekName", "hour", "date"
)

Metrics(query) <- "sessions"

MaxResults(query) <- 30000
```

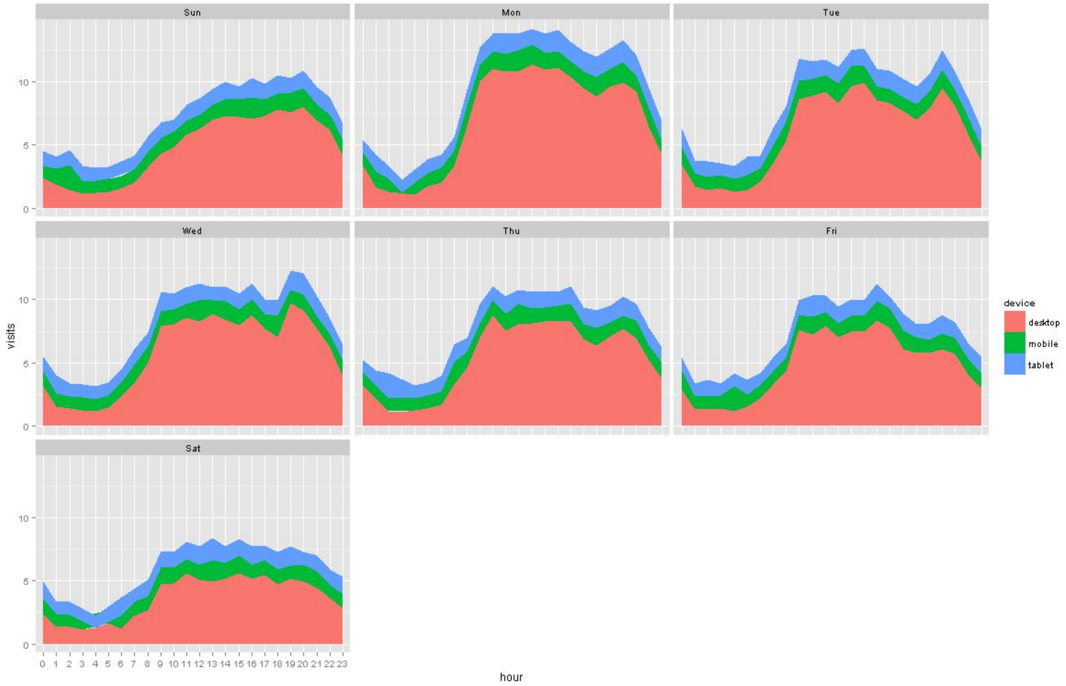
Step 2: Get data and summarise

```
data <- GetGaData(query)

library(dplyr)

weekly_data <- tbl_df(data) %>%
  group_by(deviceCategory, dayOfWeekName, hour) %>%
  summarise(avg_sessions_per_day = mean(sessions))

#   deviceCategory dayOfWeekName hour avg_sessions_per_day
#1         desktop        Friday    0             83.83673
#2         desktop        Friday    1             81.79167
#3         desktop        Friday    2             77.29167
#4         desktop        Friday    3             80.35417
#5         desktop        Friday    4             91.60417
#..            ...           ...  ...                  ...
```

Step 3: Plot the summary

```
library(ggplot2)

ggplot(weekly_data) +
  aes(
    x = hour,
    y = avg_sessions_per_day,
    fill = deviceCategory,
    group = deviceCategory
  ) +
  geom_area(position = "stack") +
  facet_wrap(~dayOfWeekName)
```

ggplot2 + dplyr + ganalytics =



Which permutation of two different device types are signed-in users most likely to transition between?

Define sequences to segment users by permutations of device type and compare against total users of each type of device.

Step 1: Setup query (needs UserID)

```
Dimensions(query) <- NULL

Metrics(query) <- c("users")

DateRange(query) <- c("2015-01-01", "2015-03-31")
# Maximum of 90 days for user-based segmentation
```

Step 2: Define sequence segments

```
devices <- list(
  desktop = Expr(~deviceCategory == "desktop"),
  mobile = Expr(~deviceCategory == "mobile"),
  tablet = Expr(~deviceCategory == "tablet")
)

device_sequences <- lapply(devices, function(from) {
  lapply(devices, function(to) {
    SegmentFilters(
      Sequence(First(from), Then(to)),
      scope = "users"
    )
  })
})
```

Step 3: Get data for each segment

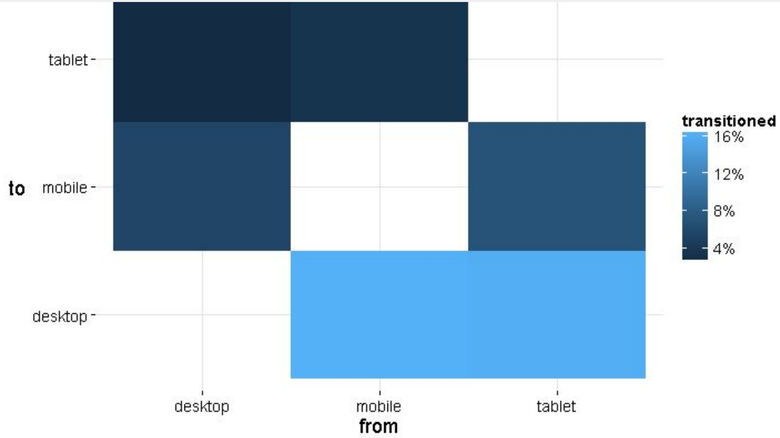
```
data <- lapply(seq_along(device_sequences), function(from_index){
  from_name <- names(device_sequences)[from_index]
  from_seq <- device_sequences[[from_index]]
  lapply(seq_along(from_seq), function(to_index){
    to_name <- names(from_seq)[to_index]
    Segment(query) <- from_seq[[to_index]]
    df <- GetGaData(query)
    df <- cbind(from = from_name, to = to_name, df)
  })
})

data <- unlist(data, recursive = FALSE)
data <- do.call(rbind, data)
```

Step 4: Summarise the data

```
library(dplyr)

benchmarks <- data %>%
  subset(from == to) %>%
  select(from, benchmark = users)

data <- data %>%
  subset(from != to) %>%
  inner_join(benchmarks, by = "from") %>%
  group_by(from, to) %>%
  summarise(transitioned = users / benchmark)

#       from      to transitioned
# 1 desktop  mobile   0.05565673
# 2 desktop  tablet   0.02547988
# 3  mobile desktop   0.16147748
# 4  mobile  tablet   0.03634899
# 5  tablet desktop   0.15945559
# 6  tablet  mobile   0.07034457
```

Step 5: Plot the results

```
library(ggplot2, scales)

ggplot(data) +
  aes(x = from, y = to, fill = transitioned) +
  geom_tile() +
  scale_fill_continuous(labels = percent) +
  theme_minimal(base_size = 18) +
  theme(axis.title.y = element_text(angle = 0))
```

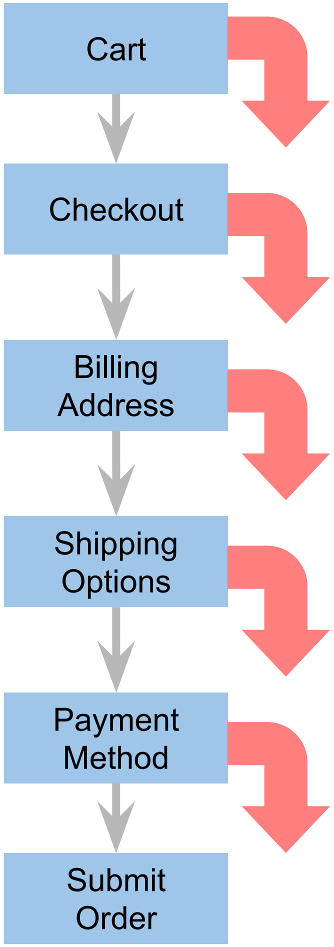


Custom Funnels

Where are shoppers likely to drop out of your ecommerce website?

Custom Funnels: GA Premium

- A linear sequence of stages like a goal funnel:
    - Restricted to only five stages
- Apply retrospectively
- Beyond just pageviews, e.g. campaigns, events
- Session-based like goals or user-based
- Like the enhanced ecommerce funnel reports:
    - Create segments on-the-go



Custom funnels for everyone else

1. Create segments for:
   a. each stage in the funnel
   b. the sequence of steps required to reach that stage
2. Get the number of users for each segment
3. Calculate:
   a. Entrances and exits for each stage
   b. Abandonment and conversion rates

Count traffic at each funnel stage

- Cart

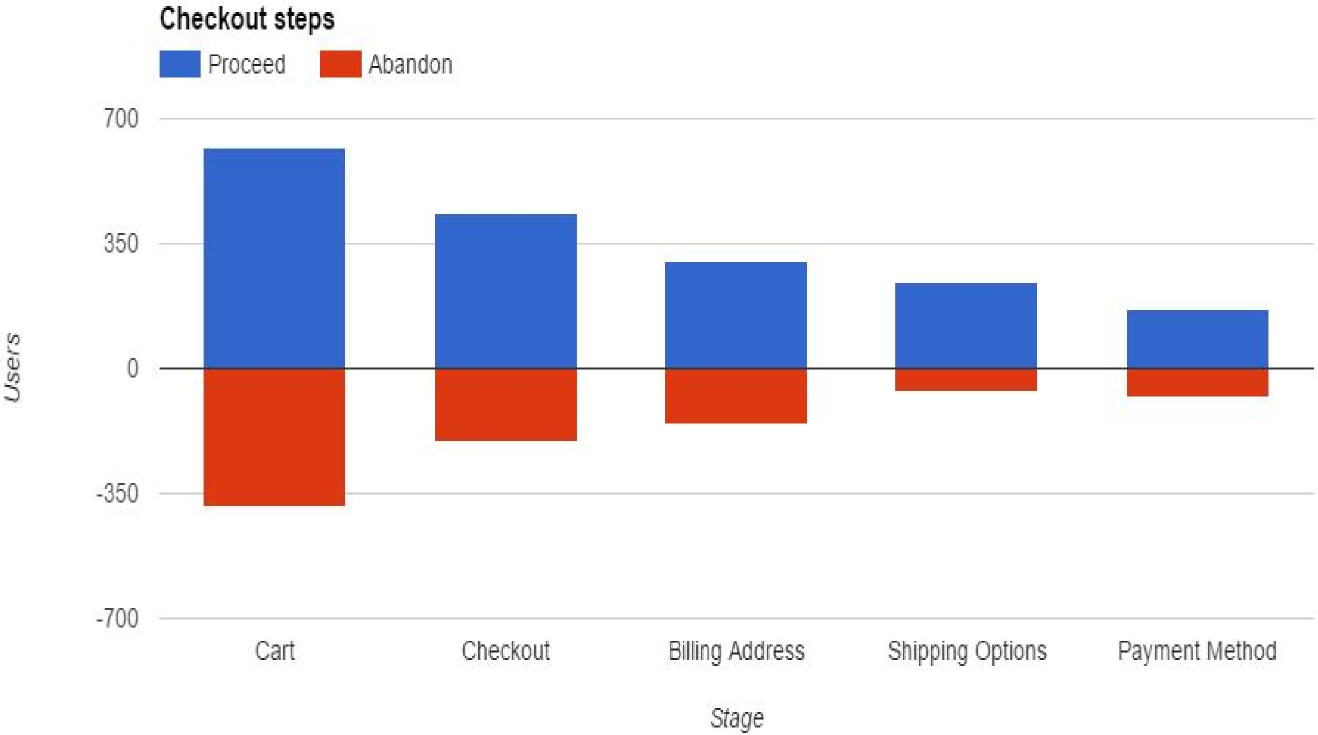
  ```
  users::condition::pagePath=~cart$
  ```

Count traffic at each funnel stage

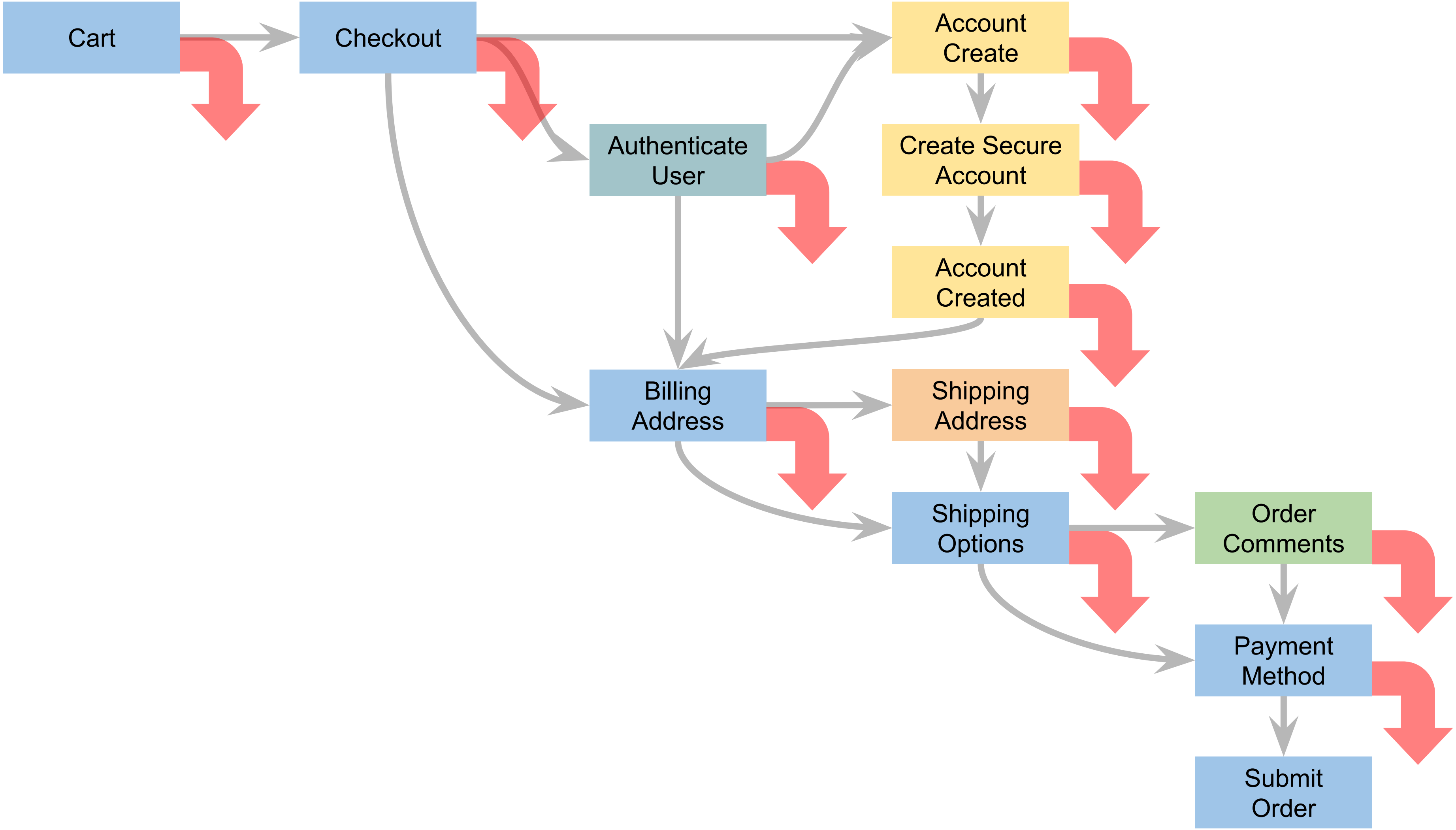
- Cart = `users::condition::pagePath=~cart$`
- Checkout = `users::condition::pagePath=~checkout$`
- Billing Address = `users::condition::pagePath=~billingaddress$`
- Shipping Options = `users::condition::pagePath=~shippingoptions$`
- Payment Method = `users::condition::pagePath=~paymentmethod$`
- Submit Order = `users::condition::pagePath=~submitorder$`

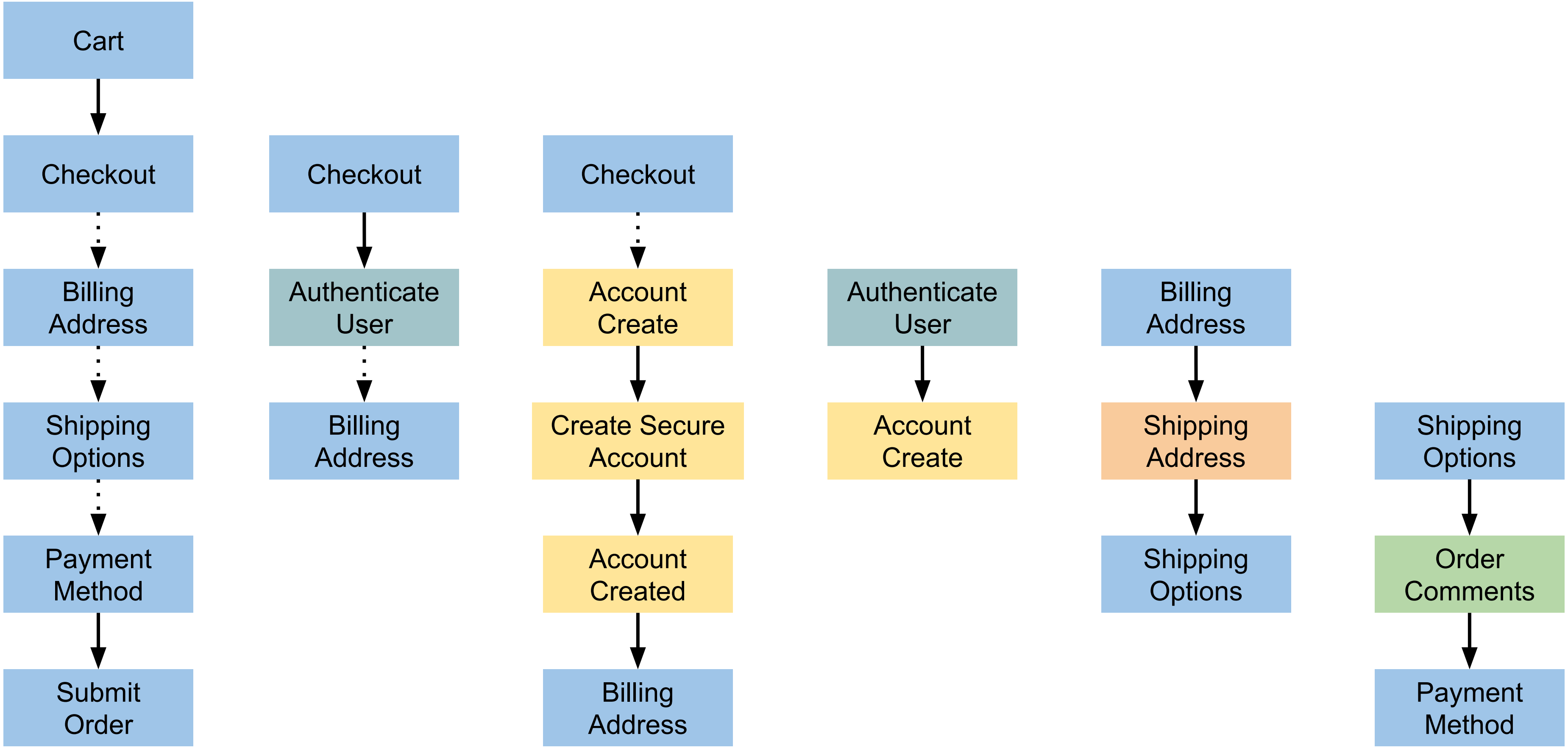
Count path completions

- users::**sequence**::
- pagePath=~cart$
- **;->>**pagePath=~checkout$
- **;->>**pagePath=~billingaddress$
- **;->>**pagePath=~shippingoptions$
- **;->>**pagePath=~paymentmethod$
- **;->>**pagePath=~submitorder$

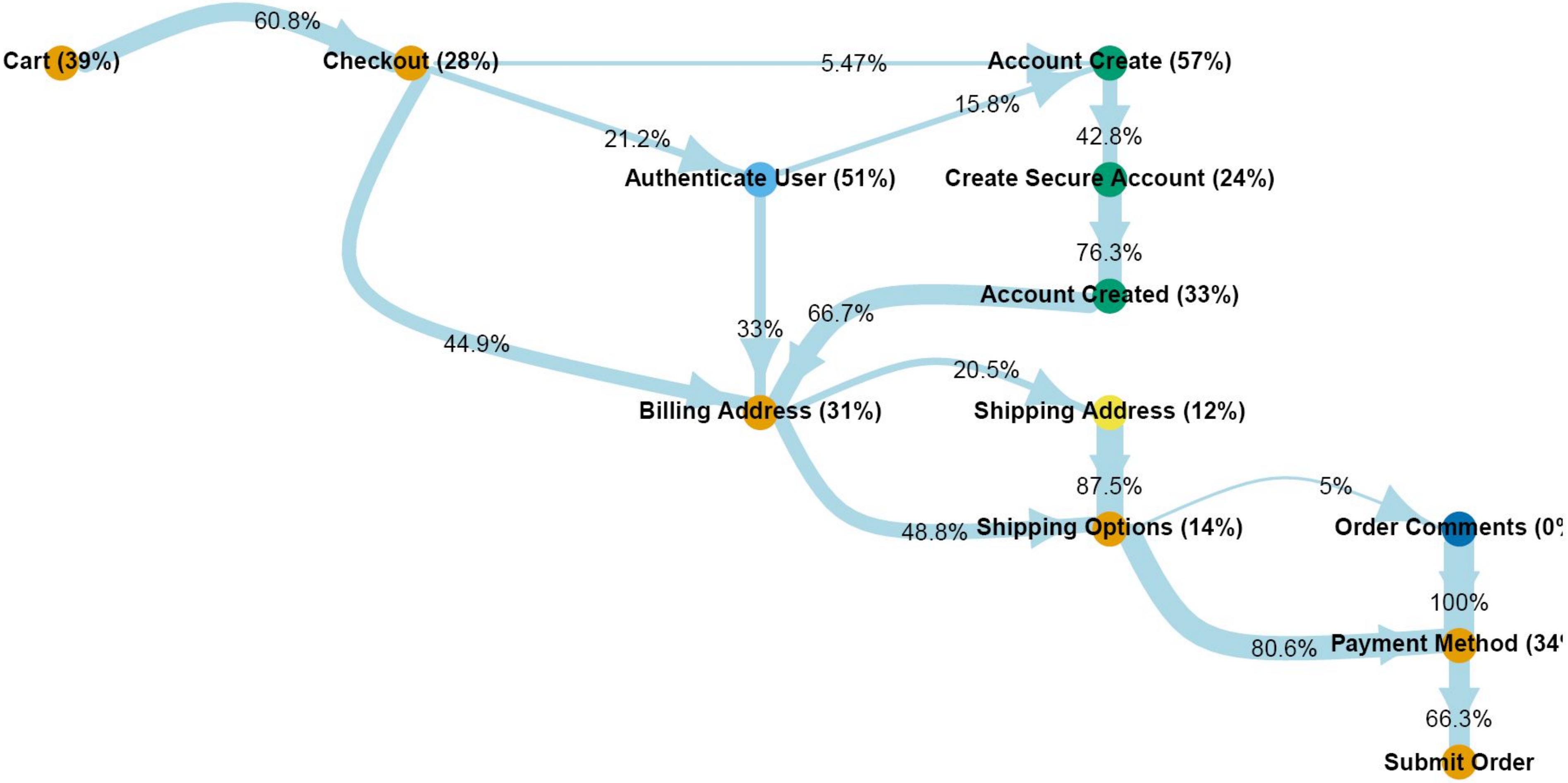

Checkout steps
Proceed
Abandon
700
350
0
-350
-700
Users
Cart
Checkout
Billing Address
Shipping Options
Payment Method
Stage

Conversion funnel analysis with R


Cart
Checkout
Account Create
Authenticate User
Create Secure Account
Account Created
Billing Address
Shipping Address
Shipping Options
Order Comments
Payment Method
Submit Order



16 transitions (5 are shortcuts), and 16 routes from start to end.


Cart (39%)
60.8%
Checkout (28%)
5.47%
Account Create (57%)
15.8%
21.2%
42.8%
Authenticate User (51%)
Create Secure Account (24%)
76.3%
Account Created (33%)
66.7%
33%
44.9%
20.5%
Billing Address (31%)
Shipping Address (12%)
87.5%
5%
48.8%
Shipping Options (14%)
Order Comments (0
100%
80.6%
Payment Method (34
66.3%
Submit Order

Using R

```
query <- GaQuery(view = 98765432)
Segment(query) <- Sequence(
  Cart,
  Checkout,
  BillingAddress,
  ShippingOptions,
  PaymentMethod,
  SubmitOrder
)
GetData(query)
```

Using ganalytics with other Google Analytics APIs

Reporting APIs

`GaQuery()`	Core Reporting API	Standard reporting over a date range
`McfQuery()`	Multi-channel funnels reporting API	Requires conversion tracking to be set up (goals or ecommerce)
`RtQuery()`	Real-time reporting API	Request developer access first

Other APIs

MetaData API		Internal package use only

Management API	`GaAccounts()`
	`GaUserSegments()`
		Work in progress so please use with caution!

Google Tag Manager API	`GtmAccounts()`	

Get involved!

Open source R package development is fun!

Remaining work to complete

- Package documentation
- Adding examples and demos
- Further error handling and messages
- Testing and finding bugs to fix
- Submitting to CRAN
- Code optimisation and tidy up
- Feature enhancements and improvements
- Occasional updates for API changes
- Internationalisation

Shiny demo

https://mark.shinyapps.io/GA-dashboard-demo/

How to get involved ...

- Go to the Github site: github.com/jdeboer/ganalytics
- Submit feedback via the issues tracker
- Get in touch on twitter: @johannux