

THE INTERNETRIX ADVANCED GUIDE TO

ATTRIBUTION MODELLING

WITH APPLICATIONS IN R

Dean Marchiori
Tiffany Chen



Internetrix

The Internetrix Advanced Guide to Attribution Modelling

With applications in R

Dean Marchiori - Head of Data Science, Internetrix

Tiffany Chen - Data Analyst, Internetrix

2019-05-07

Contents

1	Welcome	5
1.1	Contact Internetrix	5
2	Introduction	7
2.1	What is attribution modelling	7
2.2	Definitions	7
2.3	Scope of this paper	8
2.4	Where to go for further help	8
I	Online Attribution	9
3	Types of Attribution	11
3.1	Group vs. Individual Attribution	11
3.2	Simplistic vs. Fractional Modelling	11
3.3	Heuristic vs. Algorithmic Modelling	12
3.4	Classification of Methods	12
4	Methods	13
4.1	Simplistic Methods	13
4.2	Fractional Heuristic Methods	14
4.3	Fractional Algorithmic Methods	16
4.4	Comparison of Methods	21
5	Google Analytics Data	23
5.1	Survey of Basic Methods in Google Analytics	23
5.2	About Data Driven Attribution	23
5.3	BigQuery	24

II	Implementation Guides	25
6	Setup	27
6.1	R	27
6.2	BigQuery	29
7	Extracting Data from BigQuery	33
7.1	About the data	33
7.2	Get full event log	33
7.3	Identify those who converted	34
7.4	Converting touchpoints in last 7 days	34
7.5	Non-converting touchpoints in last 7 days	34
7.6	Complete Query	35
8	Heuristic Models	37
8.1	Transform Data	37
8.2	Heuristic Models	39
9	Algorithmic Methods	41
9.1	Markov Chains	41
9.2	Survival Analysis	43
III	Offline Attribution	49
10	About Offline Attribution	51
10.1	Scenario	51
10.2	Causal Inference Modelling	51
10.3	Comparing to other methods	54

Chapter 1

Welcome

The aim of this guide is to help marketing, business and analytics experts understand how to get the biggest impact from their marketing budget.

Our focus is on marketing attribution. That is, the way in which we can measure and assign credit to individual marketing channels, even when we are simultaneously advertising via many channels. Only by understanding this can we generate meaningful estimates on the return we get from each channel and therefore guide future spending decisions.

When we say marketing, the main focus is on digital marketing. In this guide we use Google Analytics and the Google Marketing Platform as the tools of choice. If you use some other digital analytics platform, you can still benefit from the content, but you will need to be a little more self-sufficient in exporting and preparing your data for analysis.

Finally, for more advanced and customised methods we use the R statistical programming language. These sections are aimed at data scientists and data analysts with some experience with scripted analysis. Hopefully the basic examples shown in this guide will help in getting up to speed and generating useful insights as quickly as possible.

We hope you enjoy and get some benefit from this guide. If you have any feedback or comments, or would like to discuss working together, please reach out using the contact information below.

1.1 Contact Internetrix

Internetrix Data Science

Web: <https://www.internetrix.com.au/services/data-science/>

Phone: +61242535300

email: irx.info@internetrix.com.au

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

Chapter 2

Introduction

2.1 What is attribution modelling

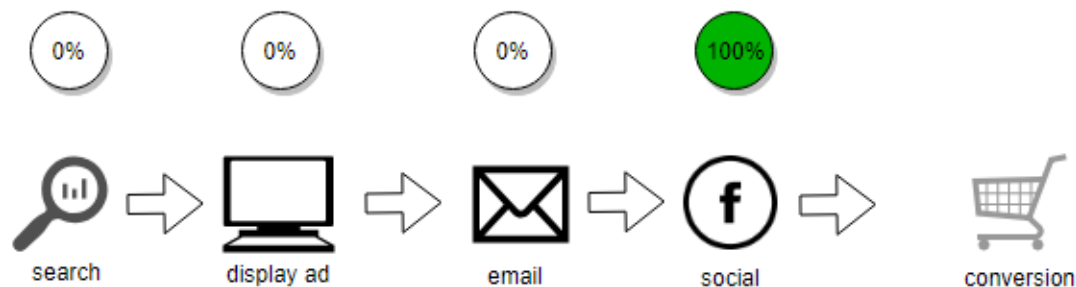
The ideal world for marketers is that customers will fall in love with your products or services and make a purchase the first time they see an advertisement. But, the truth is the customer might have hundreds of interactions with a brand before they make a purchase. Actually, they might get exposed to the products or services through different marketing channels such as social media, other web sites, search engines and apps etc.

So why do we care about this? The ultimate aim is to arm decision makers with the data needed to make complex decisions. We do this by measuring the return on investment (ROI) of marketing efforts. In Jayawardane et. al (2015) they discuss how customers who interact with a brand across multiple channels bring in more value. While it is typical to have parallel streams of marketing efforts, it is not straightforward to understand which of these channels actually helped make the conversion and to what extent.

We present a tour of methods that solve this problem with practical implementation guides and code examples. We aim to give decision makers the data to understand the true return from each channel and therefore guide marketing strategy in a data driven way.

2.2 Definitions

In this guide we talk about the ‘customer pathway’ or the ‘conversion journey’. We define this as the sequence of steps a customer takes when interacting with a brand which will ultimately lead to either the customer converting or not converting.



We say *conversion* to mean the user has completed a desired action. This may indeed be purchasing an item from a website eStore, however in other domains this may simply mean signing up to a service or newsletter or completing some other pre-defined action.

2.3 Scope of this paper

First we break down the types of models that solve this attribution problem in different ways.

For each type of approach we present an explanation and a practical example of how they work as well as a review of their strengths and weaknesses.

At Internetrix, we are accredited Google Partners and support clients with tracking and optimising their online presence using tools from the Google Marketing Platform. We review the existing methods that are available in tools like Google Analytics and Analytics 360.

However, this guide is largely aimed at users who want to customise and extend these methods and tailor them to their specific company goals. We present a toolkit using the R programming language, complete with reproducible code examples to showcase attribution modelling methods, from the very basic to more complicated and powerful algorithms.

In doing this, we use example data from Google Analytics 360 and BigQuery. We cover the often neglected steps of how to query and prepare this data for modelling. However, for users who collect their digital marketing data in other platforms, the implementation guides will still be relevant and valuable.

Finally we extend attribution modelling in a bonus section to cover *offline* attribution. That is, measuring the impact of channels outside the digital sphere like TV, radio, billboard etc. This unlocks some powerful tools to extend beyond what is tracked from your website.

2.4 Where to go for further help

While not a complete guide, we hope it provides some insights to the various methods that can help arm your organisation with the data to make effective marketing decisions.

For further help please reach out to us any time:

Internetrix Data Science

Web: <https://www.internetrix.com.au/services/data-science/>

Phone: +61242535300

email: irx.info@internetrix.com.au

Part I

Online Attribution

Chapter 3

Types of Attribution

3.1 Group vs. Individual Attribution

Historically, marketers got inspired by the marketing performance and strategies of offline channels such as newspaper and broadcasts, but there were some drawbacks. For example, they could not be used for investigating customer conversions at an *individual* level.

Marketing attribution in this case was largely done at the group level by aggregating the performance of all users responding via a particular channel. Furthermore it was difficult and often impossible to understand which customers experienced multiple touch points in these offline channels.

Currently, the prevalence of digital marketing provides the marketer with a better way to gain insights on how marketing activities perform at the individual level.

3.2 Simplistic vs. Fractional Modelling

3.2.1 Simplistic modelling

Simplistic models assign complete conversion credit to just a single touch point. A consequence of this is all other touch points in a user's conversion pathway are effectively ignored.

There are several types of simplistic attribution models, but we mainly focus on two of them, which have been widely adopted. In the **Last Touch** model, 100% credit for a conversion is assigned to the final touch point which happens before the conversion. In contrast, in the **First Touch** model, 100% credit is assigned to the first touch point a customer uses in their conversion pathway.

3.2.2 Fractional modelling

Unlike simplistic modelling techniques, fractional modelling determines the 'fractional' contribution of each touch point and helps the marketer apply appropriate weights to all channels in a client's decision-making journey.

Fractional attribution modelling is subdivided into heuristic modelling and algorithmic modelling as discussed in more detail below.

3.3 Heuristic vs. Algorithmic Modelling

3.3.1 The Heuristic model

The concept of the heuristic model, also known as rule-based fractional modelling, is summarised in Jayawardane et. al (2015) who define it as fixed rules to assign the credits to all touch points leading to the conversion.

These ‘rules’ that the heuristic modelling is based on include **Linear**, **Position Based**, **Time Decay** and others which we will explore further in later sections.

Compared to simplistic modelling, heuristic models better capture the contributions of multiple advertising and marketing channels.

3.3.2 The Algorithmic model

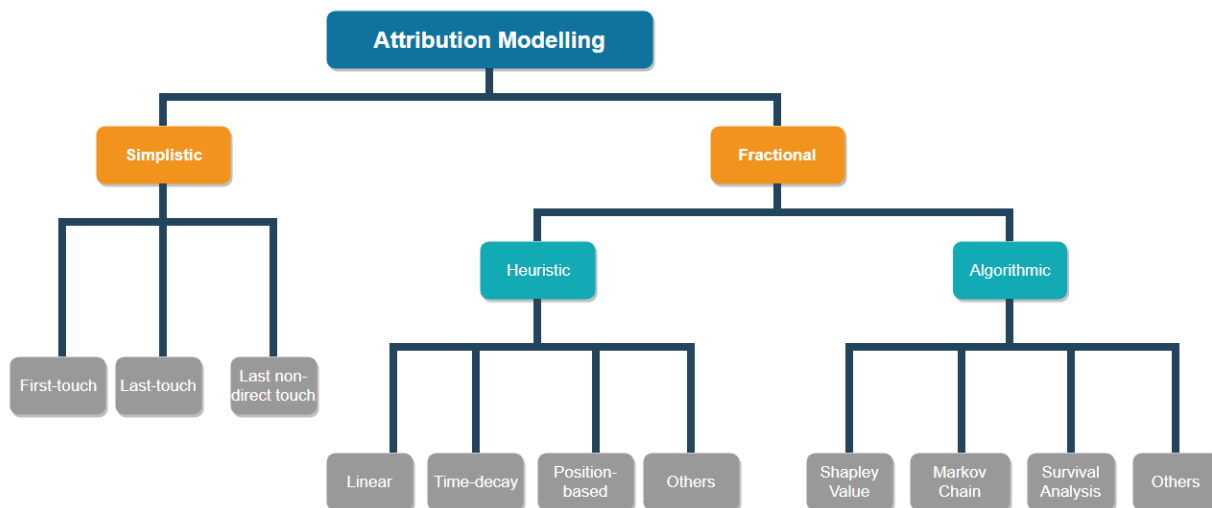
Extending the idea of fractional attribution is algorithmic attribution.

Rather than using a rule-of-thumb, an algorithmic model will analyse both converting and non-converting paths customers take and make a probabilistic assessment of each channel’s contribution to a conversion. Often using advanced statistics and machine learning algorithms, algorithmic methods develop custom weightings for each channel based on their effectiveness.

Currently, some popular approaches are Markov Chain models, the Shapley Value and survival analysis. We will take a detailed look into these methods and which are appropriate for your analysis goals.

3.4 Classification of Methods

Below is a useful taxonomy of the commonly used methods for marketing attribution modelling in eCommerce.



Chapter 4

Methods

We will now take a practical tour of some common models for marketing attribution.

A customer wants to buy some swimwear on your website. They start with a Google search that leads them to your website, but no purchase is made. A few days later they click on a display ad and go back to your website, perhaps to take another look at the product range and sign up for special offers. One week later they get an email campaign while at work and click through to check for special deals, but don't yet purchase. Later that evening, they see a Facebook ad and remember to grab their credit card and make a purchase.

So which channel caused this purchase? Was it the social channel that drove the customer to purchase, or was it the search result that made the customer initially aware of your brand, or was it the display and email campaigns that re-engaged the customer?

We will now look at a few different methods for making this decision.

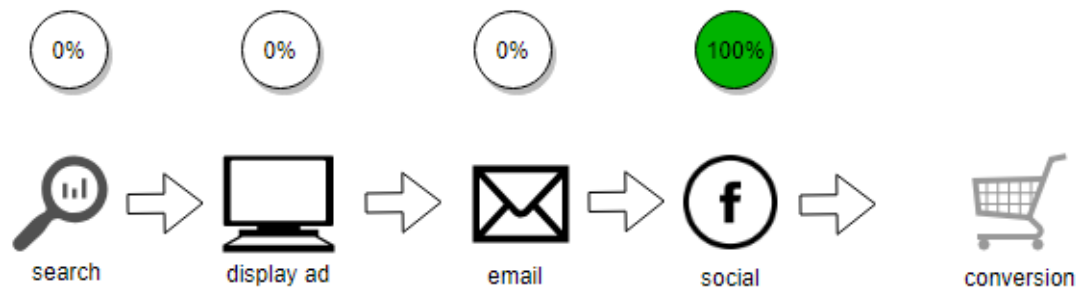
4.1 Simplistic Methods

Simplistic methods assign credit for a conversion to just one pre-determined touch point in the marketing journey. This is common and robust, but quite naive way to assess your marketing performance.

4.1.1 Last Touch

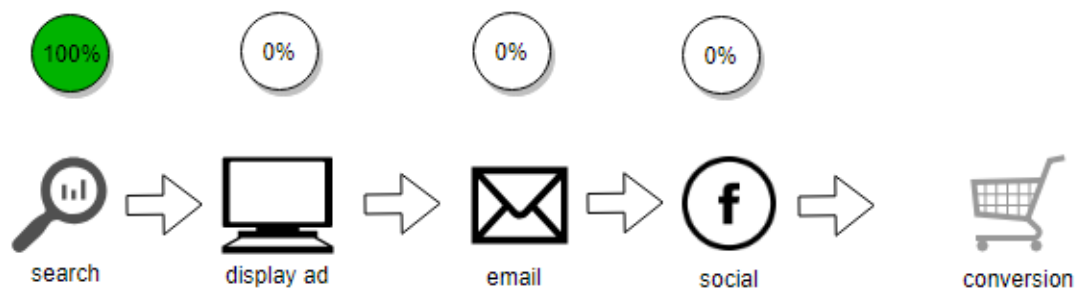
A classic example of a simplistic method is **Last Touch** attribution (sometimes also referred to as Last Interaction or Last Click).

In this model, all credit for the sale is attributed to the very last touch point in the customer's conversion pathway. In this case the social channel.



4.1.2 First Touch

A **First Touch** model assigns all credit for a sale to the first channel in a customer's conversion pathway. Here it is the search channel.



4.1.3 Others

There are a number of modifications to these that enable greater refinement. For example **Last non-direct click** is the same as Last Touch, but if the last touch is the direct channel, it ignores this and looks back to the last pure marketing channel.

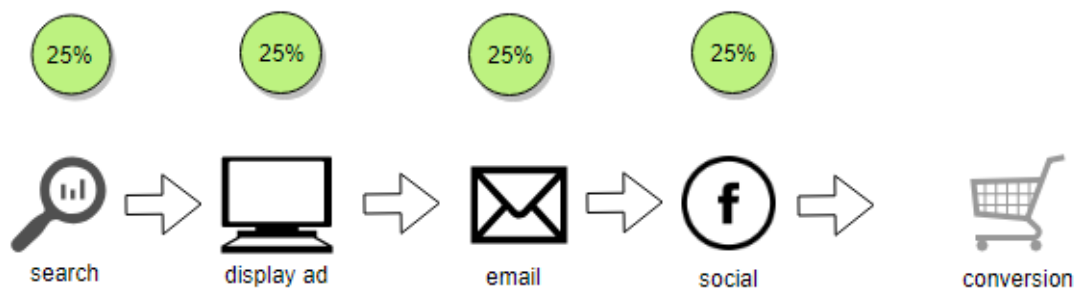
4.2 Fractional Heuristic Methods

In some specialised cases, a simplistic model may work well, however an intuitive next step is to split the credit across all or several channels that lead to the conversion. Just like a game of football, one person scores the goal, but its the entire team that works together to create the scoring opportunity.

4.2.1 Linear

The least opinionated method is known as the **Linear attribution model**. This method assigns equal credit across all marketing channels in the customer's conversion pathway.

In the example below all four touch points are assigned 25% credit for the value of the conversion.



4.2.2 Time Decay

In our scenario, considerable time passed between the first search, looking at the display ad, then getting an email campaign. A natural question is whether touch points from days or weeks ago really contribute equally to touch points made the day of a purchase? A model that accounts for this is the **Time Decay model**. This method adjusts the credit given to each channel based on how long ago the touch point occurred. This assigns less credit to distant touch points and more weighting to recent activity.



4.2.3 Position Based

Taking the concept of First Touch and Last Touch a step further, we can arbitrarily define attribution rules based on where in the conversion pathway a channel sits.

This is known as **Position Based** attribution. For example, below we have assigned 45% of the conversion credit to the first and last touch points, with equal amounts allocated to all touch points in between.



4.3 Fractional Algorithmic Methods

Algorithmic attribution models attempt to use historical data on actual conversion performance to measure or infer attribution, rather than rely on human defined rules and heuristics.



This is achieved through advanced statistical methods and algorithms that account for individual channel performance. Often these methods look at both the pathways for customers who converted and those who did not. This is a step up in sophistication and objectivity that can add tremendous value when simple rules are too naive to enable effective decision making.

4.3.1 Logistic Regression

Logistic regression is a commonly used statistical procedure developed in the 19th Century (Cramer, 2002). In simple terms, it uses a function known as the logistic function, to model a binary outcome. This may be yes/no, true/false or in our case converted/not converted. It bases this on a set of independent variables that may have contributed to the outcome of choice.

As a fractional or multi-touch method, in our case this means determining which marketing channels contributed to whether a customer converted or not. Typically this means using the various marketing channels

(email, display, search) as covariates in the model. For each customer journey, an indication will be made against each channel based on whether or not it was part of the customers conversion journey. This provides the basis for modelling the outcome of whether the customer converted or not.

customer	email	display	search	converted?
12345	TRUE	TRUE	FALSE	Yes
38546	FALSE	FALSE	TRUE	No
48379	TRUE	FALSE	FALSE	Yes
...

In practice, many of the channels will be related and gaining statistically rigorous estimates can be challenging. Other statistical techniques like ‘bagging’ have been used to improve accuracy with these methods (Shao and Li, 2011).

4.3.2 Shapley Value

The Shapley Value (Shapley, 1953) originated in 1953 by Nobel Prize winning mathematician Lloyd Shapley. It is a concept in Game Theory, specifically in systems where many factors need to cooperate to achieve a given outcome. The Shapley Value is a way of allocating credit for the total outcome achieved amongst these many cooperating factors.

In terms of a marketing example, we can see a comparison of two pathways below. Including ‘Display’ in the sequence increases the likelihood of a purchase by 2%. Therefore we can attribute this increase to ‘Display’ despite it being only a link in the sequence. To get the complete credit given to Display, more historical comparisons need to be made with Display occurring at different locations and working with different touch points.



For the technically inclined, the formula to calculate the Shapley value for a given contributor is:

$$\phi_i(v) = \frac{1}{|N|!} \sum_R [v(P_i^R \cup \{i\}) - v(P_i^R)]$$

To translate this, for any given marketing touch point (e.g. Display), what is the pay-off achieved where Display is part of the pathway sequence. Subtract from this the marginal contributions made by all the touch points preceding it. Add these contributions up across all permutations containing Display and divide by the total number of possible pathway permutations. This gives us our weighted marginal contribution to the overall outcome for a given marketing channel.

4.3.3 Markov Methods

A Markov Chain is a mathematical system that models events that transition from one ‘state’ to another. These states could be a touch point in a marketing journey, the current day’s weather or the health status of a patient. These concepts are explored further in (Gagniuc, 2017).

By recording the probability of moving from one state to another a prediction of future states can be made by knowing the present state of the system.

A Markov Chain is defined by three properties:

- The state space – the set of all the states in which process could potentially exist
- The transition matrix – the probability of moving from one state to other state
- current state probability distribution – probability distribution of being in any one of the states at the start of the process.

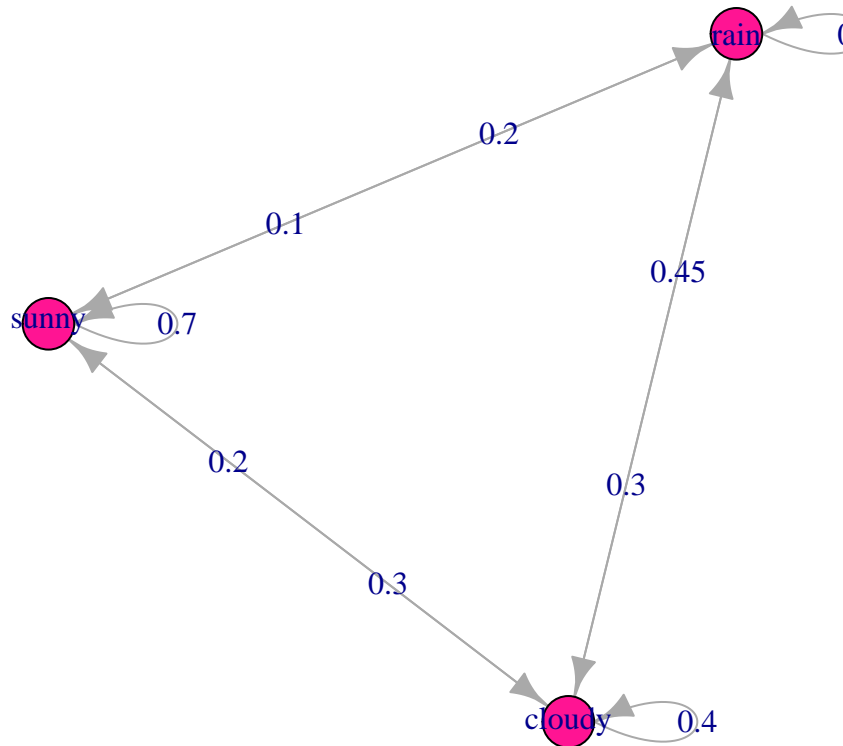
Markov chains are stochastic processes, but they have a characteristic of being ‘memory-less’. This means the predictions they make depend only on the present state of the system. Having said that, an extension to discrete time markov chains is the ability to use different ‘order’ models to account for a small number of previous states when making future predictions. We will explore this further in Section 9.1.

A basic way to think about markov chains is via weather. If we look at some historical data on whether it is sunny, cloudy or rainy we can form probabilities of moving from one of these states to another. Intuitively if it’s sunny today, we could reasonably expect a high likelihood of sunny weather tomorrow. Conversely, if it is raining today then cloud or rain are probably more likely than sunshine tomorrow.

We can view these probabilities as a ‘transition matrix’, and even plot them as a directed network graph.

	sunny	cloudy	rain
sunny	0.7	0.20	0.10
cloudy	0.3	0.40	0.30
rain	0.2	0.45	0.35

Weather transition matrix



If we know it is cloudy today we can represent that as:

$$initial = \begin{bmatrix} sunny = 0 \\ cloudy = 1 \\ rain = 0 \end{bmatrix}$$

By multiplying this into the transition matrix, we can get the probabilities of the the weather tomorrow. We can also simply read this result off our initial transition matrix.

sunny	cloudy	rain
0.3	0.4	0.3

We can continue this multiplication several times to forecast ahead say, seven days:

sunny	cloudy	rain
0.4622776	0.3188612	0.2188612

So how does this relate to marketing attribution? If we define the states in our model to be the various touch points, such as **email**, **paid search**, **display**, **social**, and include states such as **start**, **end**, **converted** we

can fully articulate our multi touch attribution model by observing the historical probability of customers moving from one state to another.

Once we have this graph defined we can measure the importance of each touch point by removing them one-by-one and simulating the resulting impact on conversions. This is known as the ‘Removal Effect’ (Anderl et al., 2014) and is covered in more detail in the practical section in Section 9.1. It is a simple way to extend our ideas of multi-touch attribution beyond basic rule based methods.

4.3.4 Survival Analysis

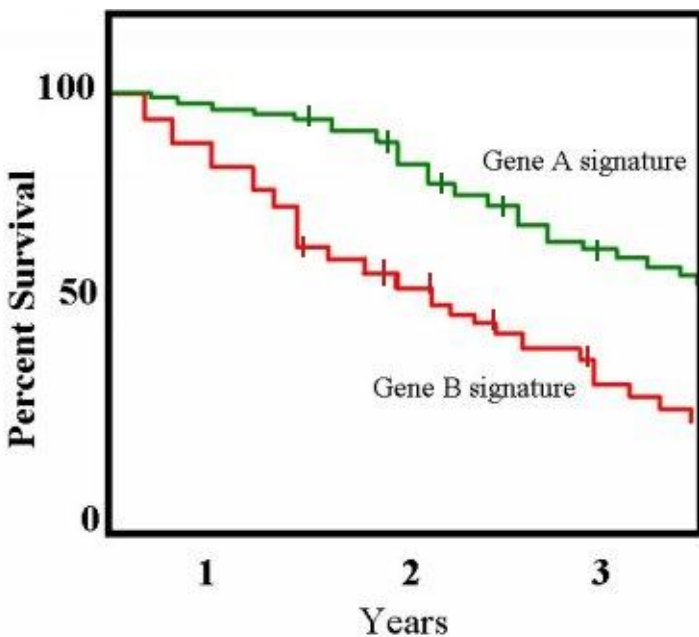
Survival analysis is a type of statistical analysis to model ‘time-to-event’ data. The name ‘survival’ evokes a medical application such as modelling how long diseased patients may survive under various treatments, however the method is applicable across a wide range of domains.

The key components of a survival analysis are (noa, 2019d):

- **Event:** This is the thing of interest, it may be death, machine failure, or sales conversion etc.
- **Time:** The time from the beginning of an analysis period to either the end of the study period or until the event of interest occurs, or disqualification from the analysis.
- **Censoring:** A key feature of survival analysis is censoring, which flags which observations did not experience the event in the time period of the analysis.
- **Survival Function:** The probability that a subject survives longer than time t .

The concept of censoring is a clear distinction with this method. While a customer may be exposed to a marketing campaign, if they do not convert in the time period of the analysis, we cannot conclude they won’t ever convert, just we have not observed a conversion event in our analysis time period.

A common method for estimating a survival function is with the use of a Kaplan-Meier curve (Kaplan and Meier, 1958) which can be used to display a chart of declining horizontal steps over time, which approximates the true survival function.



source: <https://upload.wikimedia.org/>

[wikipedia/commons/7/73/Km_plot.jpg](https://commons.wikimedia.org/wiki/File:Wikipedia/commons/7/73/Km_plot.jpg)

Mathematically this can be defined as

$$\hat{S}(t) = \prod_{i:t_i \leq t} (1 - \frac{d_i}{n_i})$$

Where $\hat{S}(t)$ is our estimator of survival (non-conversion) at time t . This formula looks at small time periods where events occur and calculates the survival probability based on the number of subjects who are still around at that time period.

There are many ways to extend this idea to marketing attribution. Rather than looking at assigning credit to touch points, we will take a slightly different approach.

In the practical section in section 9.2 we will compute an estimate for the survival function and stratify this by the channel in which our users converted. By doing this we can understand the likelihood of conversion through different channels at varying times throughout a typical conversion journey. This adds an interesting new dimension to analysing marketing performance.

4.4 Comparison of Methods

So which model is the right one to use? There is no clear cut answer. A one-size-fits-all model to attributing marketing credit does not exist (Shao and Li, 2011).

The choice should be determined by an understanding of the goals of your marketing activities and how they drive business outcomes. As a general rule, without substantive justification, heuristic and simplistic methods are very subjective and should only serve as a baseline over the data-driven methods.

Chapter 5

Google Analytics Data

Google Analytics users can benefit from a range of simple and heuristic attribution models for their digital data, build right into the platform. (noa, 2019b)

This is a great start, but they have one thing in common - they are heuristics. In other words, you ultimately make the decision on which model to use and this logic is hard-wired into the report. While these methods are quite intuitive, they are also quite naive.

5.1 Survey of Basic Methods in Google Analytics

- **Last Interaction**
- **First Interaction**
- **Linear**
- **Time Decay**
- **Last Non-Direct Click**
- **Last Google Ads Click**
- **Position Based**

A full overview is contained on the Google Help Pages

Google also has a useful feature called the Model Comparison Tool which allows the comparison of up to 3 attribution models at once.

5.2 About Data Driven Attribution

Google's Data-Driven Attribution is a feature only available in Google Analytics 360, part of Google Marketing Platform. Rather than using the position-based heuristics above, Data-Driven Attribution uses real data from your Google Analytics account to generate a custom model, driven by a more sophisticated algorithm.

5.2.1 How does it work?

The more basic, position-based methods are only interested in the paths that led to a conversion. Google's Data-Driven Attribution model analyses both converting and non-converting pathways. According to Google, it has two main steps:

1. Analyse all available path data to develop a probabilistic model of how customer journey's progress on your site.
2. Apply a sophisticated algorithm to this data to assign credit to particular marketing touch points.

The algorithm used in Data-Driven Attribution is based on a concept called the Shapley Value See Section 4.3.2, which is from the field of cooperative game theory. This method recognises the contribution a marketing touch point makes depends on where in the conversion pathway it occurs. By comparing many similar customer pathway sequences both with and without a given touch point included, a form of weighted contribution can be calculated. Put another way, intuitively this is like removing a particular marketing channel from a sequence of touch points in a user's journey and wondering what downstream impact it would have on conversions.

5.2.2 Limitations

There are some important factors to be aware of with Data-Driven Attribution. Firstly, the results presented are refreshed by Google on a weekly basis and look back at the last 28 days of conversion history at the time the model is trained. The benefit here is that models will evolve as your online activity does. Also, the model will only look back to a maximum of 4 interactions within the prior 90 days to each conversion.

Given this method learns from historical data, for the results to be meaningful there are thresholds imposed. These thresholds set the minimum amount of pathways and conversions to ensure there is enough data to train the model. At the time of writing these thresholds are:

- 400 conversions per conversion type with a path length of 2+ interactions (i.e., 400 conversions for a specific goal or transaction, not a sum of 400 over all conversion types) AND
- 10,000 paths in the selected reporting view (roughly equivalent to 10,000 users, although a single user may generate multiple paths)

5.3 BigQuery

BigQuery is a Google Developers tool that lets you run super-fast queries of large datasets.

While Google Analytics contains a plethora of online tool for analysis, when aiming to conduct more advanced digital analytics and attribution modelling, having *all* of your hit level data available is key. (noa, 2019a)

For customers that use Google Analytics 360 as part of the Google Marketing Platform, they have access to the BigQuery Export for Analytics.

Using BigQuery lets you access both session and hit level data using an SQL like syntax. Plus you benefit from the speed and scale of BigQuery along with the ability to create tables, data sets and manage jobs with your data.

Part II

Implementation Guides

Chapter 6

Setup

Before we start with the practical guides, it is important to ensure we correctly install and setup the required dependencies.

6.1 R

6.1.1 About R

The R programming language (R Core Team, 2018) is a popular and open-source tool for data analysis and statistical computing.

We will use R throughout this paper for the practical examples.



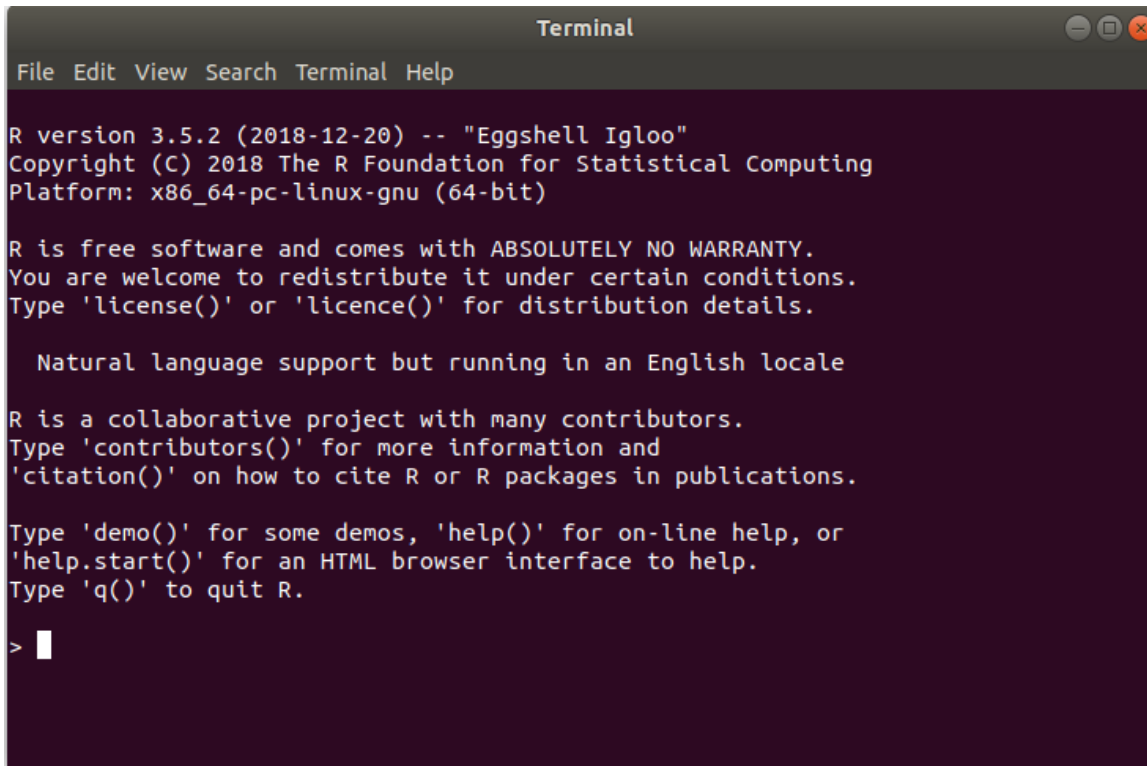
6.1.2 Install R

To install R, head over to <https://www.r-project.org/> and follow the instructions.

It will prompt you to choose a CRAN Mirror. This is an archive server that contains the latest R version.

There are instructions to install R on Linux, Mac and Windows.

This will install the base R software, which on its own is just a command line-like interface.



```

Terminal
File Edit View Search Terminal Help

R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>

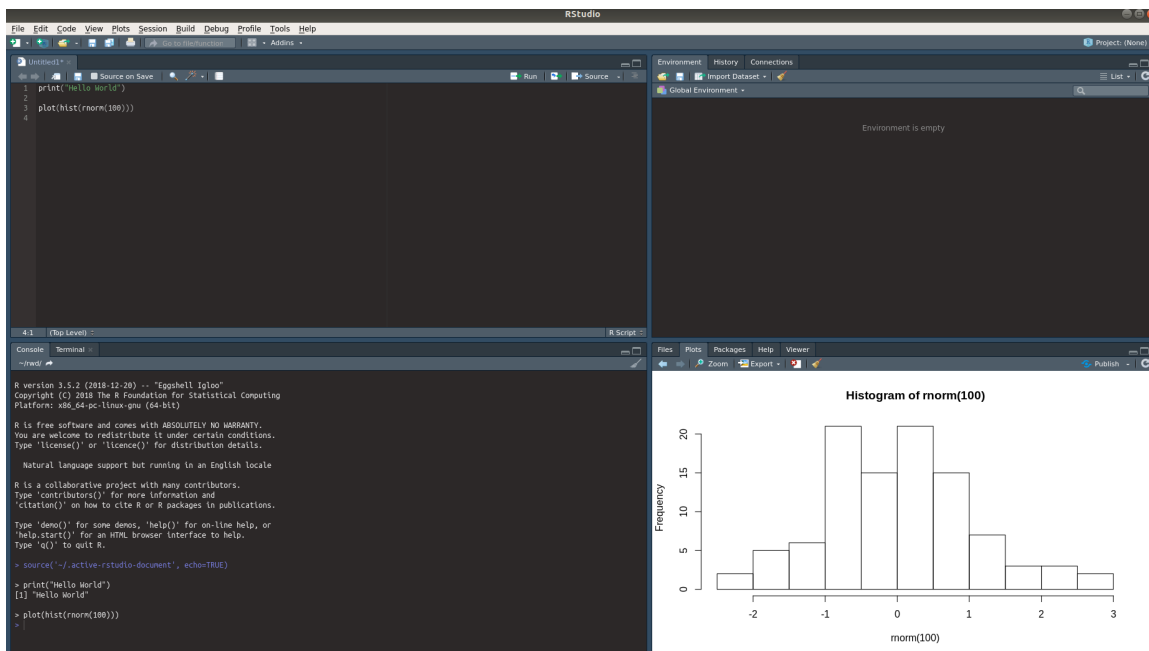
```

6.1.3 Install RStudio

To get the most out of R we recommend installing the RStudio (noa, 2019c) integrated development environment. It is an environment that provides a console with a range of features to enrich the use of R.

To download, visit <https://www.rstudio.com/products/rstudio/> and follow the instructions for your system.

The end results should look similar to this:



6.1.4 Install Required R Packages

R has a ecosystem of open source packages that work like add-ins. These packages typically contain specialised functionality that may be required when performing analysis. You can install R packages from the R console and they will be downloaded onto your system and available whenever you use R.

The main R packages we will use in this paper are:

- tidyverse - The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures. (Wickham, 2017)
- bigrquery - The bigrquery package makes it easy to work with data stored in Google BigQuery by allowing you to query BigQuery tables and retrieve metadata about your projects, data sets, tables, and jobs. (Wickham, 2019)
- ChannelAttribution - Implements a Markov Model for the Online Multi-Channel Attribution Problem. (Altomare and Loris, 2018)
- survival - Contains the core survival analysis routines, including definition of Surv objects, Kaplan-Meier and Aalen-Johansen (multi-state) curves, Cox models, and parametric accelerated failure time models. (Therneau, 2019)
- CausalImpact - Implements a Bayesian approach to causal impact estimation in time series, as described in Brodersen et al. (2015)

To install these packages, and some other supporting packages used in this paper, you can run the command below

```
install.packages(c("tidyverse", "bigrquery", "ChannelAttribution", "survival",  
                  "lubridate", "survminer", "DBI", "CausalImpact"))
```

6.2 BigQuery

The first step to more advanced modelling is extracting and cleaning the data. Here we will go through a basic setup and familiarisation guide to get your Analytics data from BigQuery. It is expected that you have some experience using SQL and knowledge of basic database operations.

6.2.1 Setup

If you are an Analytics 360 customer, you can setup BigQuery for use with Google Analytics data. Refer to the Google Help Pages for detailed instructions.

If you don't yet have Analytics 360, you can still benefit from these practical code examples by using the free Google Analytics Sample Dataset. This is a complete Google Analytics 360 data set from the Google Merchant Store, a real eCommerce platform.

To access the data set follow the instructions provided by Google:

1. Go to <http://bigquery.cloud.google.com>.
2. If you're new to BigQuery, or you don't have a project set up yet, you'll need to create a project.
3. Select this link to go directly to the data set.

4. Click **Query Table** to run a query.

In the future you can access the data set within BigQuery by selecting the `bigquery-public-data` project from the left-hand navigation panel, then select the `ga_sessions` table under the `google_analytics_sample` data set.

Please be mindful that using BigQuery can incur a cost. Please refer to Google's billing and quota information before use.

6.2.2 Export Schema

The way Google Analytics structures its data export into BigQuery is known as the export schema. This schema is able to take advantage of data structures that may be counter-intuitive for users who are familiar with normalised relational databases. BigQuery supports de-normalised tables, where instead of joining lots of flat, normalised tables, you can have one table with nested records.



In the bigquery data set, there is one table per day, defined at the session level that contains all Analytics related data nested within, such as hits and events. Below we will demonstrate some basic queries on this data.

For more information you can refer to the BigQuery Export Schema

6.2.3 Running Queries

6.2.3.1 Hello World

This is a basic query to run that simply sums the page views over a given day. You will note the use of `#standardSQL` in the header. This lets bigquery know you are using the 'standard' SQL flavour over it's older 'legacy' flavour.

```
/* Hello World Example */
#standardSQL
SELECT SUM(totals.pageviews) as TotalPageviews
FROM `bigquery-public-data.google_analytics_sample.ga_sessions_20170101`
```


Row	TotalPageviews
1	5362

6.2.3.2 Using Table Ranges

As the GA Export Schema provides one table per day, to scan across date ranges we need to specify a table range to query.

Notice the `*` in the FROM clause and the inclusion of `_TABLE_SUFFIX`.

```
/* Using Table Ranges */
#standardSQL
SELECT  date,
        SUM(totals.pageviews) as TotalPageviews

FROM    `bigquery-public-data.google_analytics_sample.ga_sessions_`

WHERE   _TABLE_SUFFIX BETWEEN '20170725' AND '20170801'

GROUP BY date

ORDER BY date
```

6.2.3.3 Hit Level Data

Accessing ‘hit’ level data, that is, individual hits within each user’s session requires ‘un-nesting’ the hits data and joining it to the `ga_sessions_` table

```
/* Now at the hit level */
#standardSQL
SELECT  fullVisitorId,
        visitNumber,
        date,
        totals.hits,
        hits.hitNumber,
        hits.type,
        hits.time,
        hits.page.pagePath

FROM    `bigquery-public-data.google_analytics_sample.ga_sessions_20170101`,
        UNNEST(hits) as hits

WHERE   fullVisitorId = '6170732910727440668' /*Random Visitor Selected*/
```

Now that we are equipped with the basics we can further our understanding in future chapters to extract the data we need for modelling.

6.2.4 Using R with BigQuery

We can also interact with the BigQuery API using the `bigrquery` package in R.

To set up we load this package, along with the DBI package which helps with database interfacing.

```
library(bigrquery)
library(DBI)
```

Next we establish a connection to BigQuery. Here you would replace `bq_test_project()` with the project id you have established in Google Cloud Console.

```
con <- dbConnect(
  bigrquery::bigrquery(),
  project = "bigquery-public-data",
  dataset = "google_analytics_sample",
  billing = bq_test_project()
)
```

We can save any SQL queries and forward these for execution as text strings.

```
sql <- "
  /* Using Table Ranges */
  #standardSQL
  SELECT date,
         SUM(totals.pageviews) as TotalPageviews

  FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`

  WHERE _TABLE_SUFFIX BETWEEN '20170725' AND '20170801'

  GROUP BY date

  ORDER BY date
"
```

Here the query is executed and saved as a temp table. The `bq_table_download` function returns the results to the console.

```
query <- bq_project_query(x = bq_test_project(), sql)
bq_table_download(query)
```

date	TotalPageviews
20170725	10728
20170726	11200
20170727	10175
20170728	9359
20170729	6293
20170730	7258
20170731	11115
20170801	10939

Chapter 7

Extracting Data from BigQuery

The first step before we implement our own attribution models in R is to extract the data from BigQuery. Here we assume you are a Google Analytics 360 customer and have set up the BigQuery export. If you are using some other form of multi-touch conversion pathway data, you can skip this section. For this example we will be using the `bigquery-public-data` data set which contains a sample of Analytics 360 data from the Google Merchant Store (See Section 6.2 for setup info).

7.1 About the data

When we conduct this analysis, a key consideration is how much data to use.

It's usually not feasible, or sensible to use all available clickstream data in your organisation. Here we have made some decisions around how much data we want to base our models on.

- Extract the last 30 days of sessions
- Keep both converting and non-converting users
- Look back 7 days of touch points from the visitor's most recent session.
- If the visitor converted, disregard any subsequent visits.

These numbers are arbitrary and should be considered in light of your organisation. Some websites with a longer conversion funnel may want to analyse 6 months of data, using a look back window of one month for example.

7.2 Get full event log

First we want to extract a log of all sessions during a time period. Here we have selected 30 days. Note how we use the `*` suffix in the `FROM` clause so we can scan multiple dates at once.

We have selected the `fullVisitorId` as the unique identifier. A time stamp of the `visitStartTime` and the `channelGrouping` which will show the Default Channel Group associated with an end user's session for this View.

```

SELECT fullVisitorId,
       TIMESTAMP_SECONDS(SAFE_CAST(visitStartTime AS INT64)) AS visitStartTime,
       channelGrouping

FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`

WHERE _TABLE_SUFFIX BETWEEN '20170101' AND '20170131'

ORDER BY fullVisitorId, visitStartTime

```

7.3 Identify those who converted

Next we identify all visitors who made a transaction during our date range, and the date of this transaction. We have specified a ‘transaction’ as our conversion goal, however there is no reason why this can’t be another goal, such as a sign up event etc.

```

SELECT fullVisitorId,
       MIN(TIMESTAMP_SECONDS(SAFE_CAST(visitStartTime AS INT64))) AS purchasetime

FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`

WHERE _TABLE_SUFFIX BETWEEN '20170101' AND '20170131'
      AND totals.transactions IS NOT NULL

GROUP BY fullVisitorId

ORDER BY fullVisitorId

```

7.4 Converting touchpoints in last 7 days

Now we can identify our converting path touch points. These are taken from the event log we defined in Step 1, however we ensure only visitors who converted appear in our results. Furthermore we restrict the results to only those touch points that occur on or before the purchase time (obviously touch points after this don’t influence the conversion outcome). Finally we implement a look-back period of 7 days, so any touch points older than this are disregarded.

```

SELECT a.*,
       'conversion' AS outcome

FROM event_log a
      INNER JOIN conversions b ON a.fullVisitorId = b.fullVisitorId

WHERE a.visitStartTime <= b.purchasetime
      AND DATE_DIFF(DATE(b.purchasetime), DATE(a.visitStartTime), DAY) <= 7

```

7.5 Non-converting touchpoints in last 7 days

For those users who did not convert, we need to identify the most recent session start time so we can calculate where to start our 7 day look back period.

```

SELECT fullVisitorId,
       MAX(visitStartTime) AS maxvisittime

FROM event_log

GROUP BY fullVisitorId

```

Now we can query the touch points of our non-converting visitors and apply our 7 day time window.

```

SELECT a.*,
       'non_conversion' AS outcome

FROM event_log a
     INNER JOIN maxtimes b ON a.fullVisitorId = b.fullVisitorId

WHERE a.fullVisitorId NOT IN (SELECT DISTINCT fullVisitorId FROM conversions)
     AND DATE_DIFF(
         DATE(b.maxvisittime),
         DATE(a.visitStartTime),
         DAY) <= 7

```

7.6 Complete Query

We can bundle this together to run as one query:

```

#standardSQL

WITH
  /* EVENT LOG */
  event_log AS (
    SELECT fullVisitorId,
           TIMESTAMP_SECONDS(SAFE_CAST(visitStartTime AS INT64)) AS visitStartTime,
           channelGrouping

    FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`

    WHERE _TABLE_SUFFIX BETWEEN '20170101' AND '20170131'

    ORDER BY fullVisitorId, visitStartTime
  ),
  /* VISITORS WHO CONVERTED */
  conversions AS (
    SELECT fullVisitorId,
           MIN(TIMESTAMP_SECONDS(SAFE_CAST(visitStartTime AS INT64))) AS purchasetime

    FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`

    WHERE _TABLE_SUFFIX BETWEEN '20170101' AND '20170131'
           AND totals.transactions IS NOT NULL

    GROUP BY fullVisitorId

    ORDER BY fullVisitorId

```

```

),

/* LATEST VISIT TIME FOR ALL VISITORS */
maxtimes AS (
  SELECT fullVisitorId,
    MAX(visitStartTime) AS maxvisittime

  FROM event_log

  GROUP BY fullVisitorId
)

/*== MAIN QUERY THAT UNIONS CONVERTING AND NON CONVERTING PATHS WITHA GIVEN TIME WINDOW ==*/

/* CONVERTING PATHS */
SELECT a.*,
  'conversion' AS outcome

FROM event_log a
  INNER JOIN conversions b ON a.fullVisitorId = b.fullVisitorId

WHERE a.visitStartTime <= b.purchasetime
  AND DATE_DIFF(DATE(b.purchasetime), DATE(a.visitStartTime), DAY) <= 7

UNION ALL

/* NON CONVERTING PATHS */
SELECT a.*,
  'non_conversion' AS outcome

FROM event_log a
  INNER JOIN maxtimes b ON a.fullVisitorId = b.fullVisitorId

WHERE a.fullVisitorId NOT IN (SELECT DISTINCT fullVisitorID FROM conversions)
  AND DATE_DIFF(DATE(b.maxvisittime), DATE(a.visitStartTime), DAY) <= 7

```

Chapter 8

Heuristic Models

Here we will implement some non-algorithmic methods as a baseline. To do this we will use the ChannelAttribution R package

8.1 Transform Data

The ChannelAttribution package requires the data structured in a certain way. In this case it is in the form:

path	conversion	non-conversions
direct > social > search	10	154
direct > direct > direct	2	234
referral > direct	7	187

Here the touch points are transformed from into a single string path, separated by the > character. For each path we aggregate the total number of conversions that resulted from this pathway, and also the number of non-conversions.

If you are using marketing data from a system other than BigQuery, you will need to prepare your data per the above.

Now we can look at the steps required to transform our data from Chapter 7.

In this case we have the results saved as a CSV, but these may also be queried directly in R as per Section 6.2.4

```
library(tidyverse)
library(lubridate)

paths_raw <- read_csv('bigquery/bq-results.csv')
```

Below is a snapshot of the top 20 rows. We can see from BigQuery it is in a standard 'long' format with one row per touch point based on the time stamp.

fullVisitorId	visitStartTime	channelGrouping	outcome
07184911138250312	2017-01-18 06:57:50 UTC	(Other)	non_conversion
07184911138250312	2017-01-18 07:40:31 UTC	(Other)	non_conversion
07184911138250312	2017-01-18 08:18:50 UTC	(Other)	non_conversion
3112985461863519829	2017-01-25 20:42:23 UTC	(Other)	non_conversion
4720404071621394560	2017-01-18 07:02:40 UTC	(Other)	non_conversion
6060076679741207514	2017-01-18 06:59:27 UTC	(Other)	non_conversion
0003297619580760716	2017-01-08 05:50:50 UTC	Direct	non_conversion
00035794135966385	2017-01-20 12:46:25 UTC	Direct	non_conversion
0004867638405459898	2017-01-15 14:22:26 UTC	Direct	non_conversion
0005604256236421547	2017-01-24 21:04:26 UTC	Direct	non_conversion
0006746295360194683	2017-01-24 05:16:18 UTC	Direct	non_conversion
0009834325573666752	2017-01-24 22:43:34 UTC	Direct	non_conversion
001324382917654255	2017-01-10 18:46:32 UTC	Direct	non_conversion
0013701781325366363	2017-01-25 15:32:52 UTC	Direct	non_conversion
0014256672578655164	2017-01-24 17:39:56 UTC	Direct	non_conversion
0015731153666510386	2017-01-25 22:37:42 UTC	Direct	non_conversion
0016316356325418630	2017-01-09 03:08:40 UTC	Direct	non_conversion
0016883628233932470	2017-01-04 18:02:46 UTC	Direct	non_conversion
0017373815580187343	2017-01-25 11:59:15 UTC	Direct	non_conversion
0018094491063949293	2017-01-18 15:18:19 UTC	Direct	non_conversion

We are using the **tidyverse** conventions here to make the interpretation easier. To translate we can see we start by formatting the time stamp correctly.

Next we rank the sessions by this time stamp for each visitor so we know which order the touch points occurred. We next restructure the data by summarising the touch points into one path string. Finally we count the occurrence of conversions and non-conversions.

```
paths <- paths_raw %>%
  mutate(visitStartTime = ymd_hms(visitStartTime)) %>%
  group_by(fullVisitorId, outcome) %>%
  arrange(visitStartTime) %>%
  summarise(path = paste(channelGrouping, collapse = ' > ')) %>%
  ungroup() %>%
  count(outcome, path, name = "n") %>%
  spread(outcome, n) %>%
  replace_na(list(conversion = 0, non_conversion = 0)) %>%
  arrange(desc(conversion))
```

path	conversion	non_conversion
Referral	178	4015
Organic Search	137	21659
Direct	71	9308
Referral > Referral	55	553
Organic Search > Organic Search	39	1507
Direct > Direct	26	803
Referral > Referral > Referral	26	132
Paid Search	21	1348
Display	12	256
Direct > Referral	11	60

8.2 Heuristic Models

Now that the data are in the correct format we can use the `ChannelAttribution::heuristic_models()` function to compare three common models: First Touch, Last Touch and Linear.

This function will automatically calculate the total number of conversion attributed to each channel using the above models.

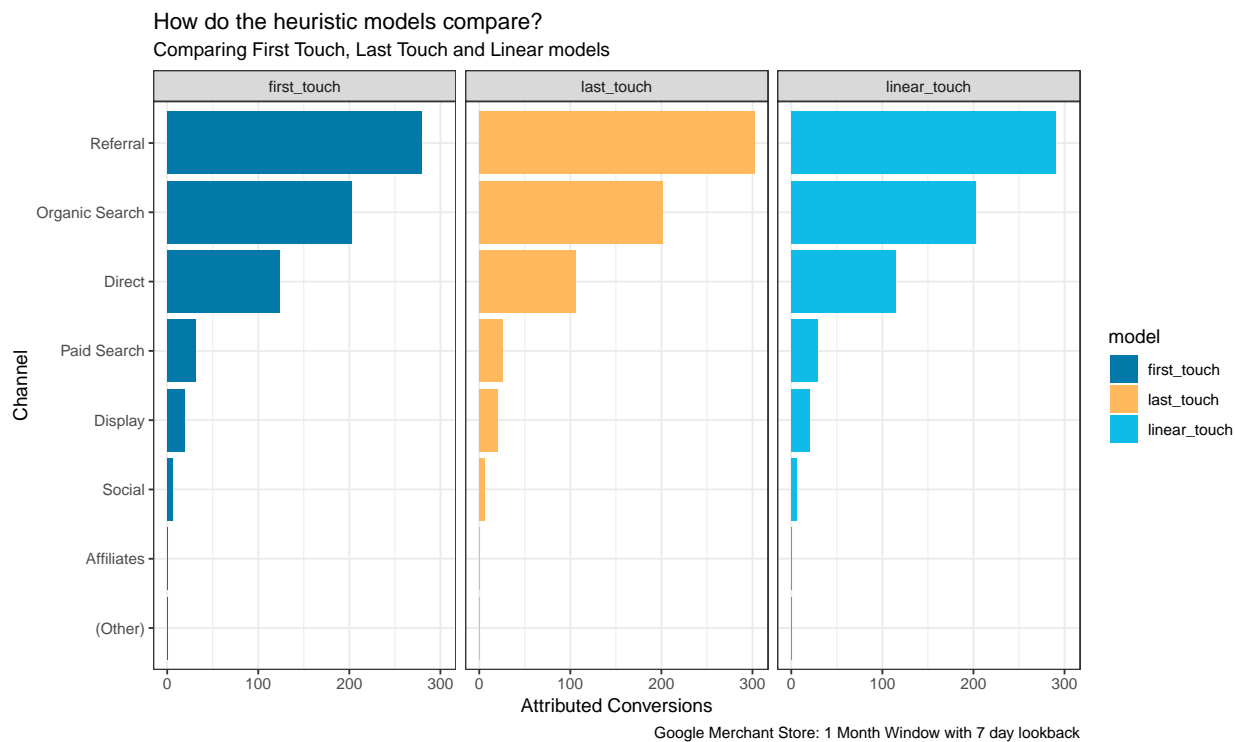
The results across these three methods are very similar. We can see 'Referral' is the channel that is attributed the most credit for conversions, followed by 'Organic Search'.

The Last Touch method provides slightly more credit to 'Referral' than other methods, in contrast to 'Direct', which is attributed more credit when using the First Touch method.

```
library(ChannelAttribution)
```

```
fit_h <- heuristic_models(Data = paths, var_path = 'path', var_conv = 'conversion')
```

channel_name	first_touch	last_touch	linear_touch
Referral	279	302	290.46667
Organic Search	203	202	202.28333
Direct	124	106	114.31667
Paid Search	31	26	29.08333
Display	19	20	19.85000
Social	6	6	6.00000
(Other)	0	0	0.00000
Affiliates	0	0	0.00000



Chapter 9

Algorithmic Methods

Now we move into algorithmic methods for modelling marketing attribution.

These methods use more advanced statistical algorithms to create a data driven model for assigning conversion credit to channels in a multi-touch point conversion pathway.

9.1 Markov Chains

The first method we will review are markov chains. We introduced these in section 4.3.3.

First we load the required packages. In this case we again use **ChannelAttribution** as well as the **tidyverse** and **lubridate** packages for data manipulation and date handling. We also read in our raw results from BigQuery that contains the event log and timestamps of visits, complete with channel and outcome.

```
library(tidyverse)
library(lubridate)
library(ChannelAttribution)

paths_raw <- read_csv('bigquery/bq-results.csv')
```

fullVisitorId	visitStartTime	channelGrouping	outcome
07184911138250312	2017-01-18 06:57:50 UTC	(Other)	non_conversion
07184911138250312	2017-01-18 07:40:31 UTC	(Other)	non_conversion
07184911138250312	2017-01-18 08:18:50 UTC	(Other)	non_conversion
3112985461863519829	2017-01-25 20:42:23 UTC	(Other)	non_conversion
4720404071621394560	2017-01-18 07:02:40 UTC	(Other)	non_conversion
6060076679741207514	2017-01-18 06:59:27 UTC	(Other)	non_conversion
0003297619580760716	2017-01-08 05:50:50 UTC	Direct	non_conversion
00035794135966385	2017-01-20 12:46:25 UTC	Direct	non_conversion
0004867638405459898	2017-01-15 14:22:26 UTC	Direct	non_conversion
0005604256236421547	2017-01-24 21:04:26 UTC	Direct	non_conversion

Next we need to transform the data. We use the same procedure as in Section 7.1. We now have one row per conversion path, with total conversions/non-conversions.

path	conversion	non_conversion
Referral	178	4015
Organic Search	137	21659
Direct	71	9308
Referral > Referral	55	553
Organic Search > Organic Search	39	1507
Direct > Direct	26	803
Referral > Referral > Referral	26	132
Paid Search	21	1348
Display	12	256
Direct > Referral	11	60

We now call the `markov_model()` function from the `ChannelAttribution` package. It accepts arguments for the data frame, the variable that contains the conversion path, the variable that encodes both number of conversions and non conversions and the order of the markov model.

Below we can see it's output is a list of distinct channels with the total attributed conversions per channel. The channel that receives the most credit is 'Referral', followed by 'Organic Search'.

As a marketer we could multiply these by the average conversion value to get a total attributed value for each channel. By comparing this to the cost of marketing in each channel we get a robust calculation for ROI.

In fact, if the actual sales revenue per customer is recorded we can go one step further and have this model calculate the attributed value without having to estimate using the average value. This is handled with the argument `var_value`.

```
fit_m <- markov_model(Data = paths,
                      var_path = 'path',
                      var_conv = 'conversion',
                      var_null = 'non_conversion',
                      order = 1)

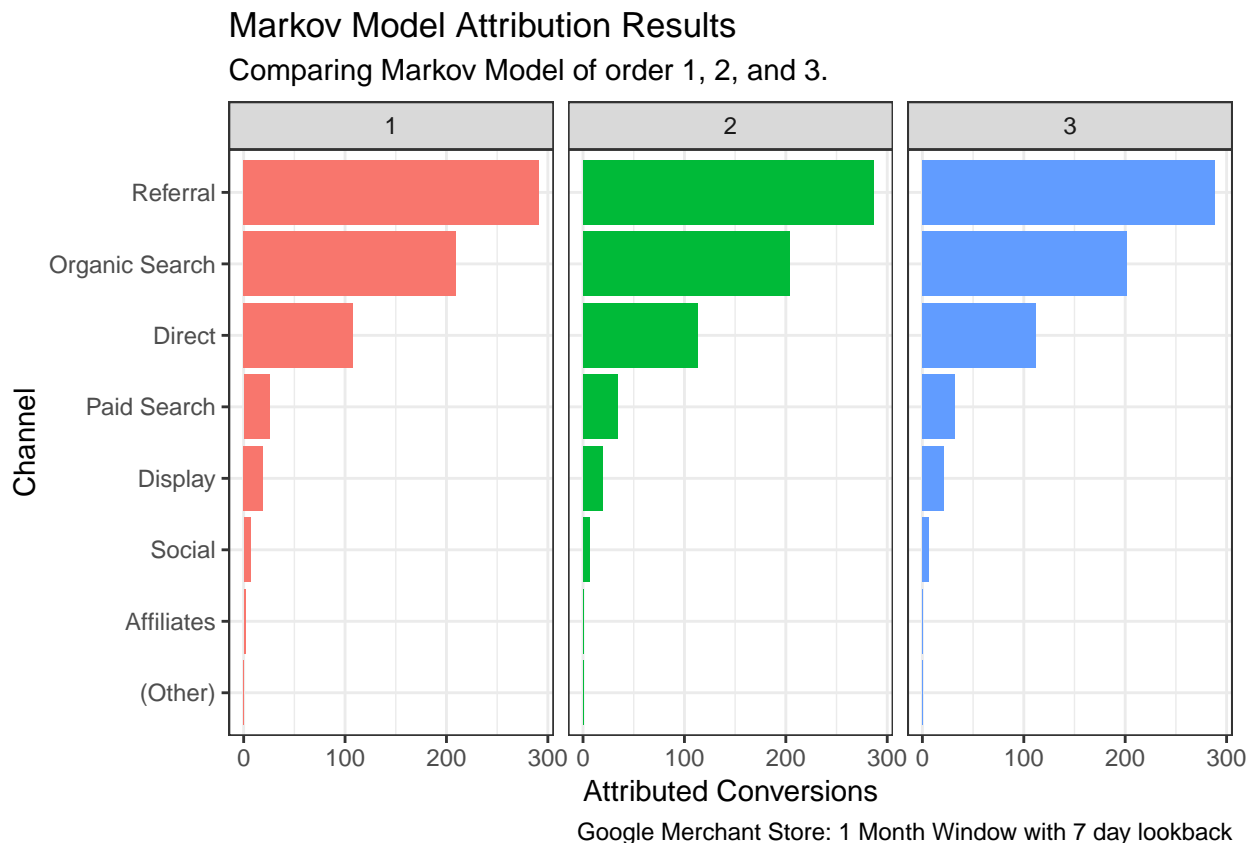
fit_m
```

```
##      channel_name total_conversions
## 1      Referral      289.7406085
## 2 Organic Search      209.9964297
## 3        Direct      106.2056815
## 4    Paid Search       27.3863707
## 5        Display       19.7819000
## 6         Social        7.2961813
## 7         (Other)        0.1541447
## 8    Affiliates        1.4386836
```

We can also iterate on this by calculating a 1, 2, and 3 order markov model.

Below we display a chart of the results. As we can see, there is not much difference in the results.

```
fit_mult <- map_dfr(.x = c(1, 2, 3),
                   .f = ~markov_model(Data = paths,
                                       var_path = 'path',
                                       var_conv = 'conversion',
                                       var_null = 'non_conversion',
                                       order = .x),
                   .id = "order")
```



9.2 Survival Analysis

The next model we demonstrate is survival analysis. Here we define ‘survival’ as non-conversion and the event of interest is when a customer converts.

As usual we start by loading the data and the required packages.

```
library(tidyverse)
library(lubridate)
library(survival)
library(survminer)

paths_raw <- read_csv('bigquery/bq-results.csv')
```

fullVisitorId	visitStartTime	channelGrouping	outcome
07184911138250312	2017-01-18 06:57:50 UTC	(Other)	non_conversion
07184911138250312	2017-01-18 07:40:31 UTC	(Other)	non_conversion
07184911138250312	2017-01-18 08:18:50 UTC	(Other)	non_conversion
3112985461863519829	2017-01-25 20:42:23 UTC	(Other)	non_conversion
4720404071621394560	2017-01-18 07:02:40 UTC	(Other)	non_conversion
6060076679741207514	2017-01-18 06:59:27 UTC	(Other)	non_conversion
0003297619580760716	2017-01-08 05:50:50 UTC	Direct	non_conversion
00035794135966385	2017-01-20 12:46:25 UTC	Direct	non_conversion
0004867638405459898	2017-01-15 14:22:26 UTC	Direct	non_conversion
0005604256236421547	2017-01-24 21:04:26 UTC	Direct	non_conversion

The data transformation steps here are a little different.

We want to condense our data into one row per customer. For our analysis, three key pieces of information are required:

- 1) The interval of time:
 - a) For converting customers, between the first visit and the purchase time.
 - b) For non-converting customers, between the first visit and the last recorded visit.
- 2) The outcome: 1 for converted, 0 for non-converted.
- 3) The channel used to convert.

Firstly, this analysis is slightly different. We aren't strictly attributing credit between channels, but rather analysing at various points in time, what is the probability of a customer converting through any given channel.

Secondly, we have made some assumptions around excluding (or censoring) customers who don't convert at the point of the most recent visit. In effect we are declaring these customer lost to follow up. An alternative method would be to calculate the time interval for non-converters right up until the end of the analysis period. Both are ok, but given we constrained our look back period to just 7 days we will go with our chosen method.

```
surv_data <- paths_raw %>%
  mutate(visitStartTime = ymd_hms(visitStartTime)) %>%
  group_by(fullVisitorId) %>%
  mutate(mindate = min(visitStartTime),
         maxdate = max(visitStartTime)) %>%
  filter(visitStartTime == maxdate) %>%
  mutate(time = (maxdate - mindate)/3600,
         status = ifelse(outcome == "conversion", 1, 0)) %>%
  dplyr::select(fullVisitorId, time, status, channelGrouping)
```

```
## # A tibble: 10 x 4
## # Groups:   fullVisitorId [10]
##   fullVisitorId      time      status channelGrouping
##   <chr>          <time>    <dbl> <chr>
## 1 48853089148264994~ 0.00000000~      0 Social
## 2 77424453797616436~ 0.00000000~      0 Organic Search
## 3 35382961118733745~ 0.00000000~      0 Direct
## 4 22905878572329286~ 0.00000000~      0 Organic Search
## 5 49866419995749112~ 0.00000000~      0 Organic Search
## 6 853089376150635567 0.00000000~      0 Organic Search
## 7 81055338799633676~ 0.00000000~      0 Referral
## 8 55543663053734243~ 0.00000000~      0 Referral
## 9 44838988143636314~ 0.01555556~      0 Social
## 10 836224165758780524 0.00000000~      0 Social
```

Next we create a special object called a survival object using the `Surv` function.

```
surv_object <- Surv(time = surv_data$time, event = surv_data$status)
```

We can now compute our estimate for a survival curve using the `survfit` function.

We include a grouping variable of the converting channel. This will calculate one survival curve per group so we have a basis for comparison.

```
fit_surv <- survfit(surv_object ~ channelGrouping, data = surv_data)
```

```
## Call: survfit(formula = surv_object ~ channelGrouping, data = surv_data)
##
##
```

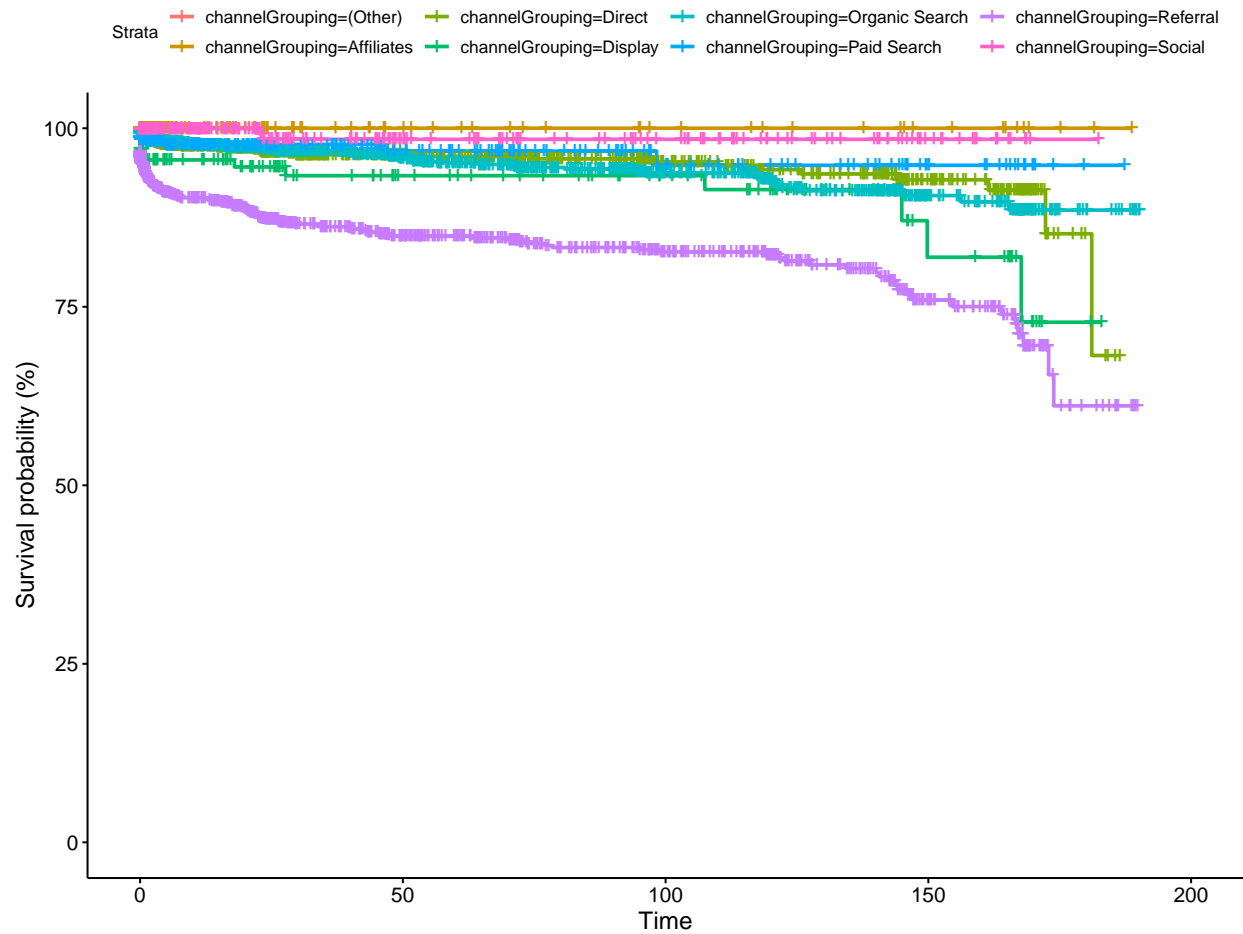
	n	events	median
## channelGrouping=(Other)	2	0	NA
## channelGrouping=Affiliates	963	0	NA
## channelGrouping=Direct	10555	106	NA
## channelGrouping=Display	408	20	NA
## channelGrouping=Organic Search	24208	202	NA
## channelGrouping=Paid Search	1769	26	NA
## channelGrouping=Referral	5372	302	NA
## channelGrouping=Social	9764	6	NA

```
##
## 0.95LCL 0.95UCL
## channelGrouping=(Other) NA NA
## channelGrouping=Affiliates NA NA
## channelGrouping=Direct 181 NA
## channelGrouping=Display NA NA
## channelGrouping=Organic Search NA NA
## channelGrouping=Paid Search NA NA
## channelGrouping=Referral 174 NA
## channelGrouping=Social NA NA
```

The results are best viewed graphically.

It is important to note that the terminology ‘survival’ in this context means ‘non-conversion’. It shows, up to a given time along the x-axis, what is the probability of a customer *not* converting through a particular channel.

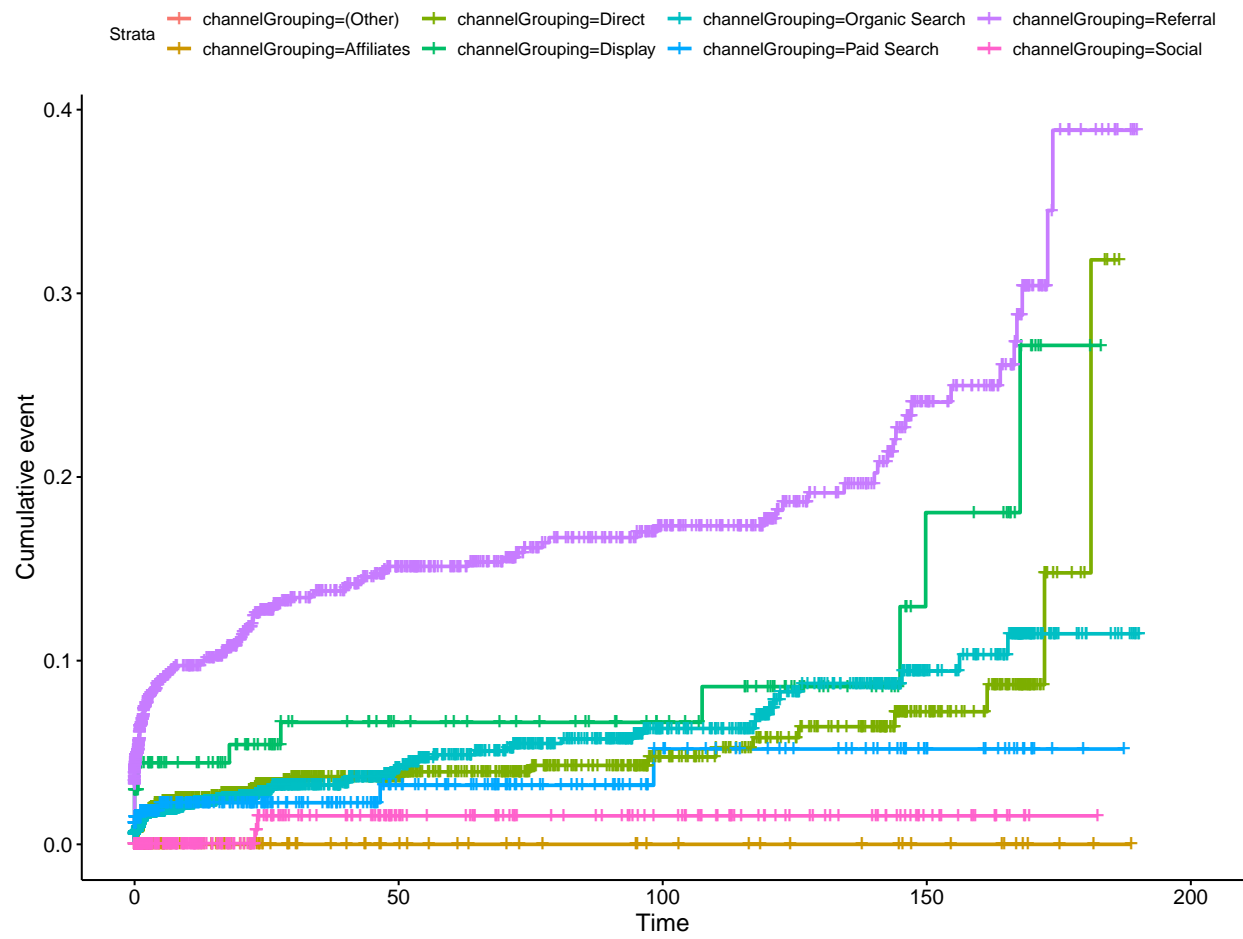
Survival Curve for each converting channel



What we want is the complement of this, that is, the cumulative event plot.

We can see that Referral channel achieves high levels of conversion very quickly, representing a valuable channel. This is followed by Display, however as time goes on, after 150 hours (~6 days) from first visit, channels such as 'Direct' and 'Organic Search' have a strong conversion probability. This indicates that advertising and referral traffic are essential to raising awareness early on, and after this, customer recall of the brand is high with customers returning days later via direct and organic channels.

Cumulative Event Plot for Customer Conversion



Part III

Offline Attribution

Chapter 10

About Offline Attribution

So far we have looked exclusively at digital marketing. However offline media channels also play an important part for many businesses. Generally speaking, the offline media channels include TV, broadcast, newspaper, coupons and outdoor advertising etc. Compared to online attribution, it is more difficult to measure the impact of offline marketing interventions.

When a randomised controlled experiment is not possible, an inferential method can assist. Below we demonstrate one such method for causal inference.

Developed by Google, the **CausalImpact** R package implements methods for causal inference using Bayesian structural time-series models (Brodersen et al., 2015).

Lets see how this works.

10.1 Scenario

You run a business that sells widgets. Throughout the year your product demand and website traffic goes up and down based on a variety of seasonal factors.

You decide to run a TV commercial to promote your product coming into a busy time of year. This advertisement kicks off on a specified launch date and runs in just one of your markets (not all regions).

How can we measure the impact of this TV advertisement on website traffic?

The intuitive answer is measure the change in sessions before and after the launch date of the commercial. If we want to get more sophisticated we might try to compare the change in our advertising region vs. all others.

However, a key problem here is recognising what would have happened if we **didn't** run our TV commercial. If we timed our campaign to coincide with a busy time of year, how much of our uplift is due to the advertising versus just organic uplift? Are we giving the advertising channel too much credit and inflating our ROI estimates?

10.2 Causal Inference Modelling

The solution uses a three step process:

- 1) Identify a *control group* which in that case could be website sessions from another unaffected region.

- 2) Using historical data from our advertising region, construct a model that predicts what would have happened in our advertising region during the campaign period if no action was taken. This is called the *counterfactual*.
- 3) Compare this counterfactual prediction with the actual number of website sessions to calculate the actual uplift attributable to our TV commercial.

It is important that the control group selected is not impacted by the campaign in any way, otherwise the results may be misleading.

Lets first load the packages required:

```
library(CausalImpact)
library(tidyverse)
```

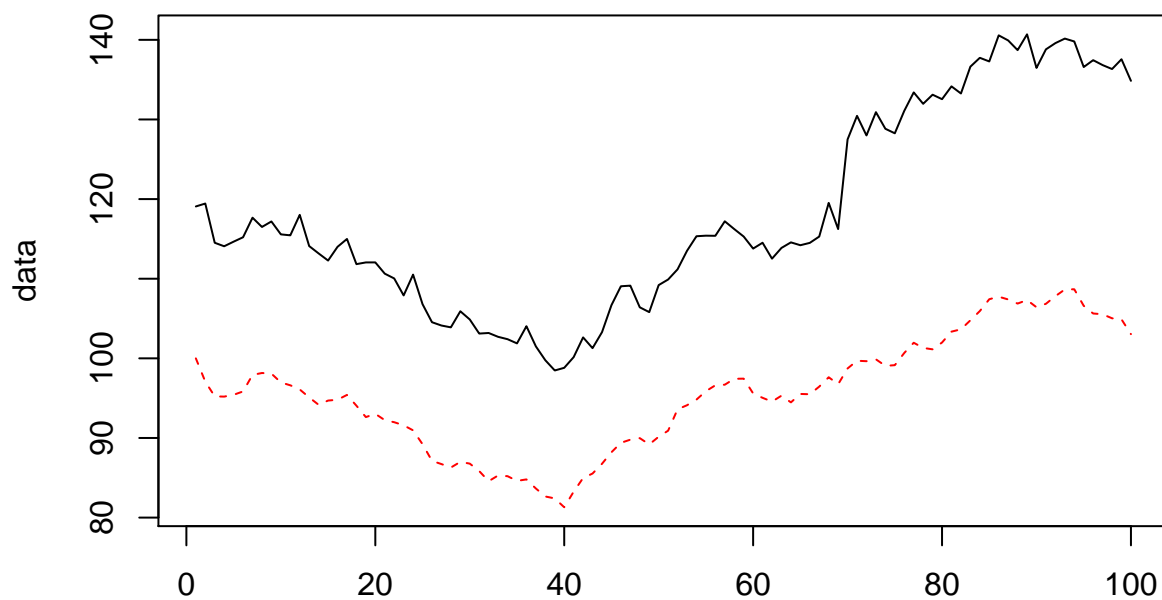
For this example we will simulate some time series data.

The red dashed line is the web traffic for our control region. The solid black line is the web traffic for our region where the TV commercial is airing.

We will select our TV campaign period to be from day 70 - 100 with an uplift of 10 sessions per day.

```
set.seed(2018)
x1 <- 100 + arima.sim(model = list(order = c(1,1,0), ar = 0.7), n = 99)
y <- 1.2 * x1 + rnorm(100)
y[70:100] <- y[70:100] + 10
data <- cbind(y, x1)

matplot(data, type = "l")
```



```
pre.period <- c(1, 70)
post.period <- c(71, 100)
```

We can now fit the causal inference model using the `CausalImpact()` function.

```
impact <- CausalImpact(data, pre.period, post.period)
```

```
## Posterior inference {CausalImpact}
##
##               Average      Cumulative
## Actual          135          4062
## Prediction (s.d.) 125 (0.89) 3765 (26.64)
## 95% CI           [124, 127] [3712, 3817]
##
## Absolute effect (s.d.) 9.9 (0.89) 297.2 (26.64)
## 95% CI           [8.2, 12]  [245.0, 350]
##
## Relative effect (s.d.) 7.9% (0.71%) 7.9% (0.71%)
## 95% CI           [6.5%, 9.3%] [6.5%, 9.3%]
##
## Posterior tail-area probability p: 0.001
## Posterior prob. of a causal effect: 99.8997%
##
## For more details, type: summary(impact, "report")
```

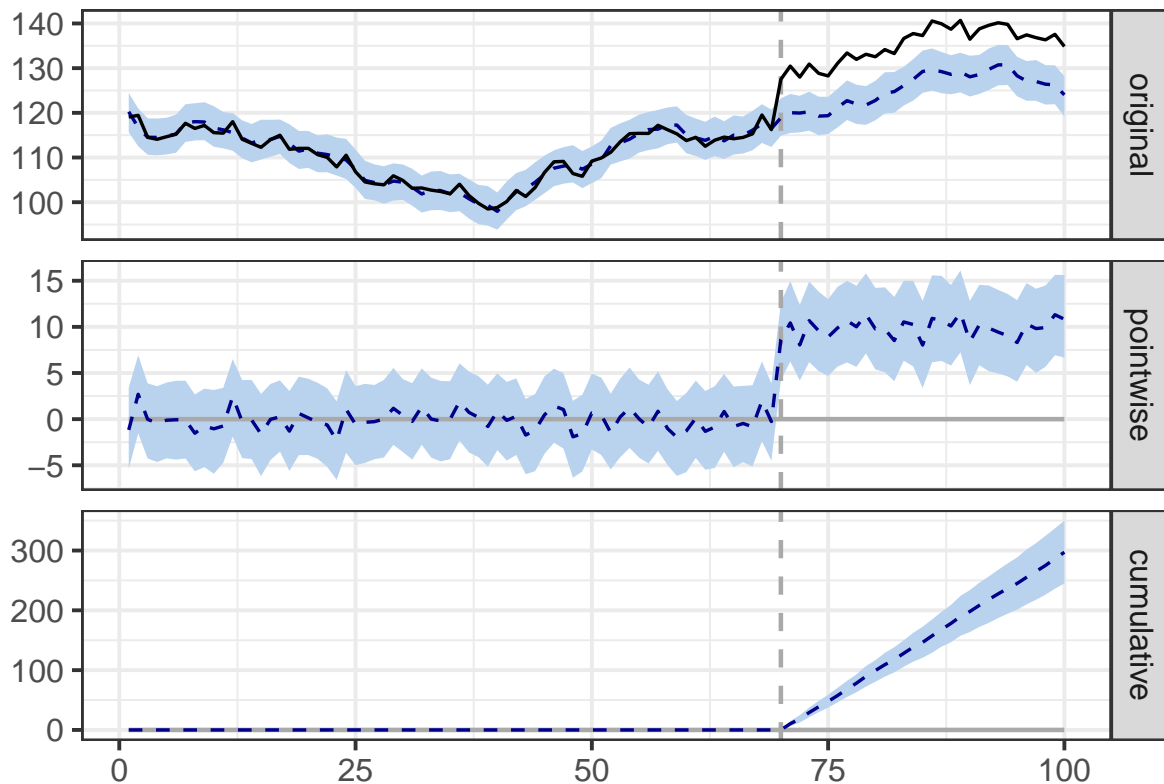
We can also view the results graphically.

The ‘original’ facet shows the actual website visits in the black solid line and predicted values without marketing intervention in the blue dashed line. The period of intervention is shown as vertical dashed lines. The confidence intervals is shaded in blue.

The second, ‘pointwise’ graph basically shows the difference between the actual values and the predicted values.

The third, ‘cumulative’ graph shows the summed effect of the marketing intervention after accumulating the differences caused by the marketing activity since the start date of the intervention.

```
plot(impact)
```



We can see that the modelled counterfactual increases in the campaign period, so too does the actual website sessions. Rather than rely on prior period comparisons we are able to extract the pointwise and cumulative uplift in a more reasoned way.

We can also generate a written analysis report. We can see the estimate effect size is 9.91 extra sessions per day, very close to the +10 we injected in our made-up example.

```
summary(impact, "report")
```

```
## Analysis report {CausalImpact}
##
##
## During the post-intervention period, the response variable had an average value of approx. 135.40. By
##
## Summing up the individual data points during the post-intervention period (which can only sometimes be
##
## The above results are given in terms of absolute numbers. In relative terms, the response variable sh
##
## This means that the positive effect observed during the intervention period is statistically signifi
##
## The probability of obtaining this effect by chance is very small (Bayesian one-sided tail-area probab
```

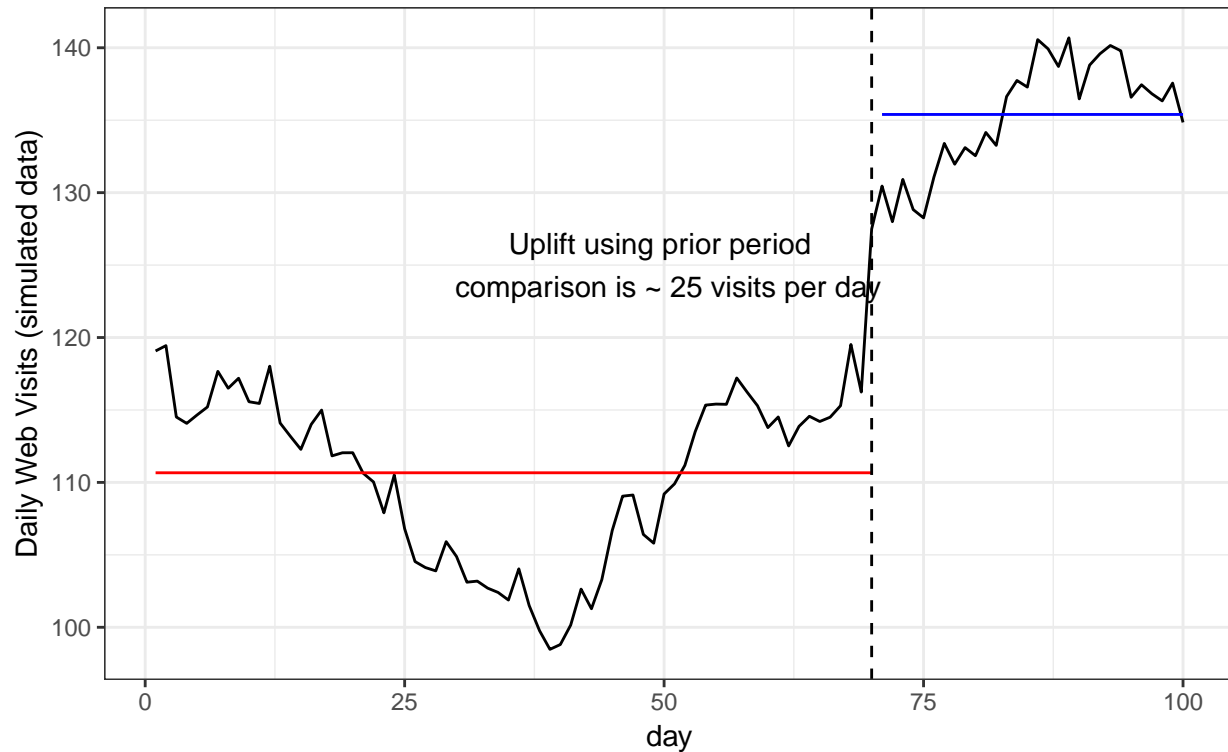
10.3 Comparing to other methods

If we rely on a simple, more naive method in this case, we can overestimate the impact of our TV commercial.

By comparing the average daily web sessions before vs. after the campaign started we see an uplift near +25 visits per day. We know this is an over-estimation as we only injected an uplift of +10 when we created the synthetic data.

How does our measure of uplift change using more basic methods?

By comparing average daily visits before and after the tv ad leads to overestimating uplift



Bibliography

- (2019a). BigQuery - Analytics Data Warehouse | BigQuery.
- (2019b). Google Marketing Platform - Unified Advertising and Analytics.
- (2019c). RStudio.
- (2019d). Survival analysis. Page Version ID: 888519729.
- Altomare, D. and Loris, D. (2018). *ChannelAttribution: Markov Model for the Online Multi-Channel Attribution Problem*. R package version 1.14.
- Anderl, E., Becker, I., Wangenheim, F., and Schumann, J. (2014). Mapping the customer journey: A graph-based framework for online attribution modeling.
- Brodersen, K. H., Gallusser, F., Koehler, J., Remy, N., and Scott, S. L. (2015). Inferring causal impact using bayesian structural time-series models. *Annals of Applied Statistics*, 9:247–274.
- Cramer, J. S. (2002). The origins of logistic regression.
- Gagniuc, P. A. (2017). *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons.
- Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481.
- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Shao, X. and Li, L. (2011). Data-driven multi-touch attribution models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 258–264. ACM.
- Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.
- Therneau, T. M. (2019). *survival: Survival Analysis*. R package version 2.44-1.1.
- Wickham, H. (2017). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.2.1.
- Wickham, H. (2019). *bigquery: An Interface to Google's 'BigQuery' 'API'*. R package version 1.1.0.