Give Credit Where Credit is Due

•••

A Data-Driven Approach to Marketing Channel Attribution

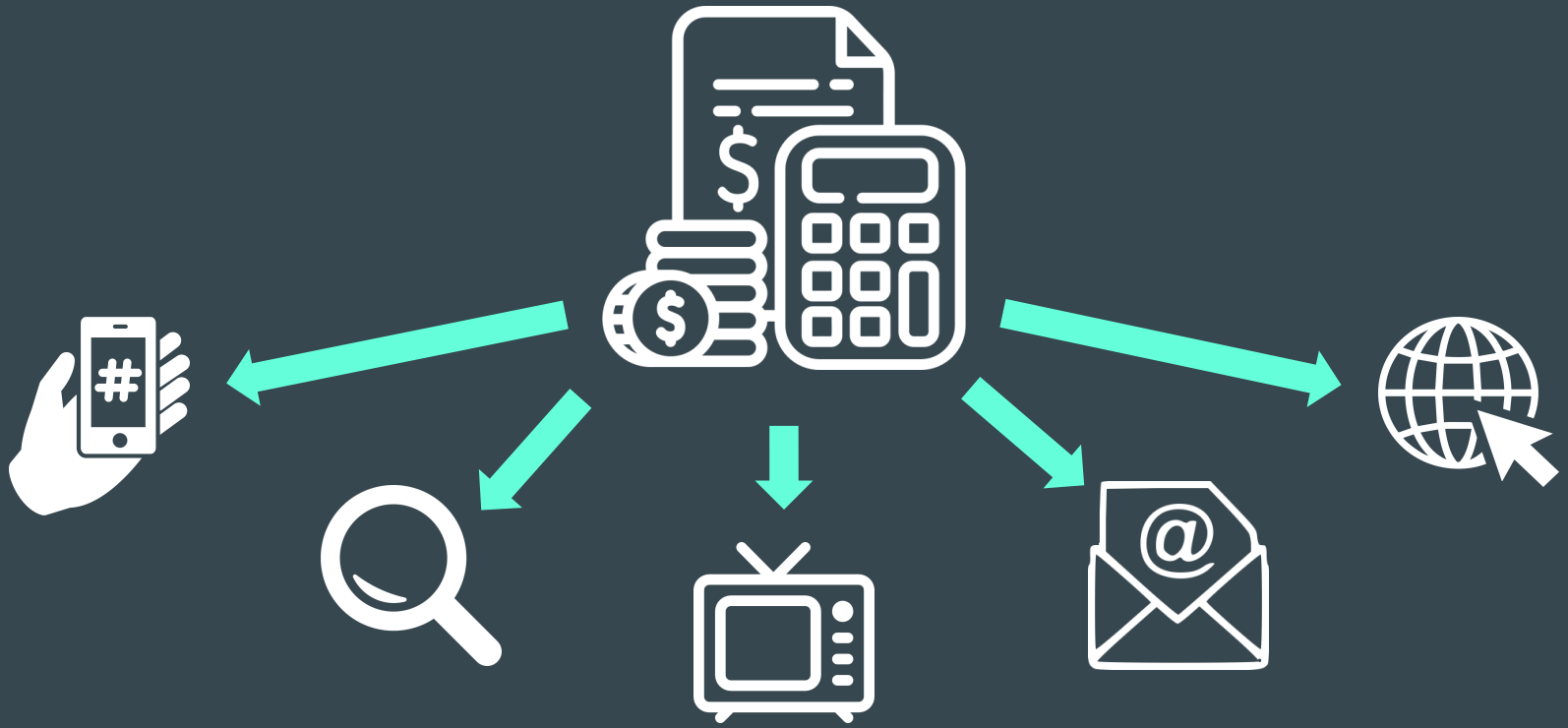
by Jessica Duncan
July 13, 2023

What is Marketing Channel Attribution?

An approach to understand which of the touchpoints along a customer journey drives the customer to conversion.



Attribution Drives Program Efficiency



Many Paths to Conversion



Many Paths to Conversion (or not)

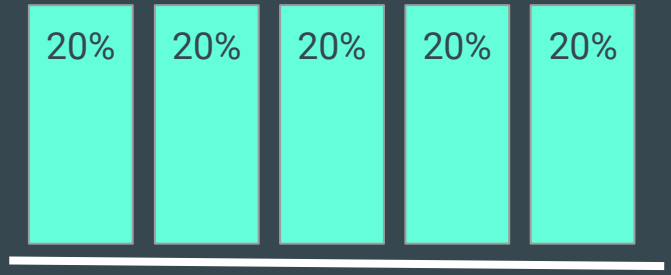


Marketing Attribution analyzes all these paths to optimize efficiency in marketing spend

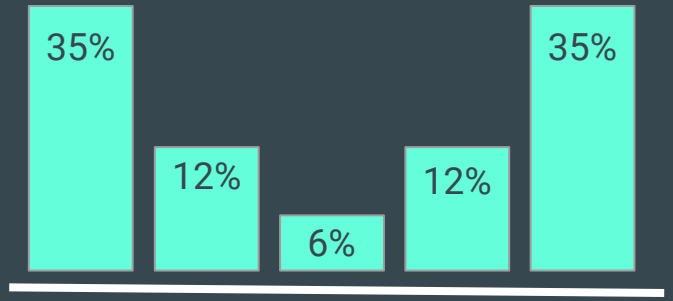
Rules Based, Single Touch Attribution



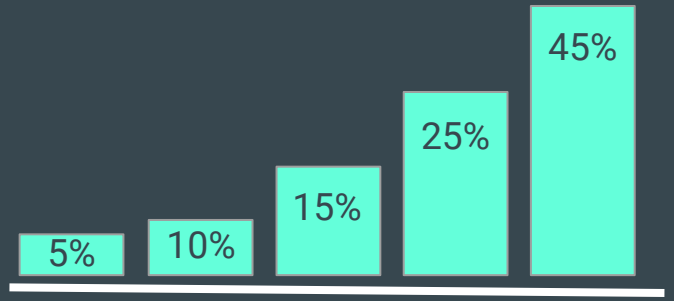
Rules Based, Multi Touch Attribution



Linear



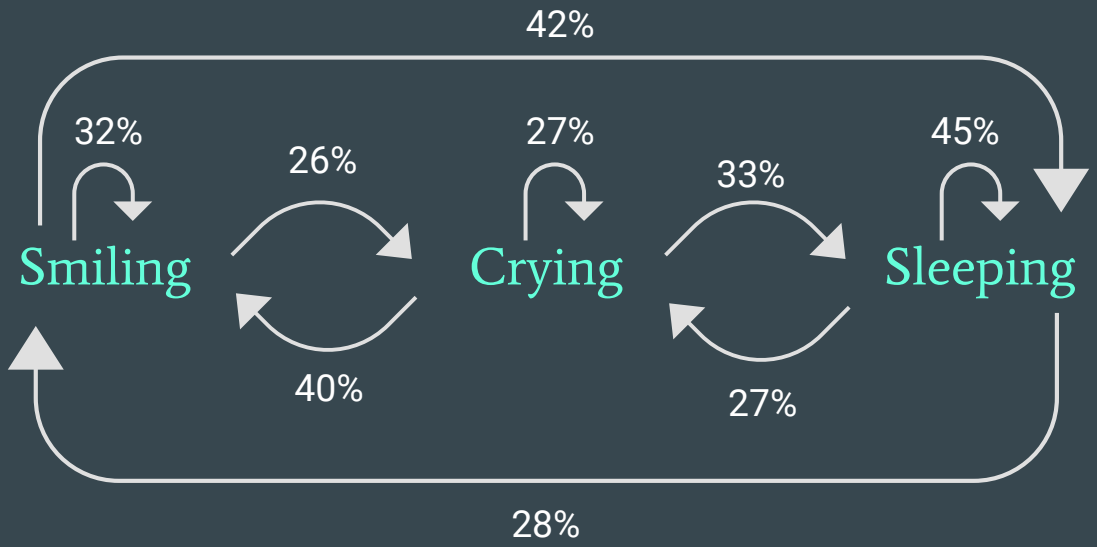
U-Shaped



Time Decay

Data-Driven Approach with Markov Chain

What's a Markov Chain? Modeling technique that aims to predict the probability of moving from one state to another, or staying in the same state.



CODE*

* the fun stuff

Generate Beautifully Clean, Hypothetical Data

```r
# define channels
channels <- c('A', 'B', 'C', 'D', 'E', 'F')

# create sample data
data <- data.frame(id = sample(c(1000:7500), 50000, replace = TRUE),
                   date = sample(c(1:185), 50000, replace = TRUE),
                   channel = sample(channels, 50000, replace = TRUE
                                    prob = c(0.34, 0.24, 0.25, 0.
        mutate(date = as.Date(date, origin = "2023-01-01"),
               channel = paste0('channel_', channel))
```

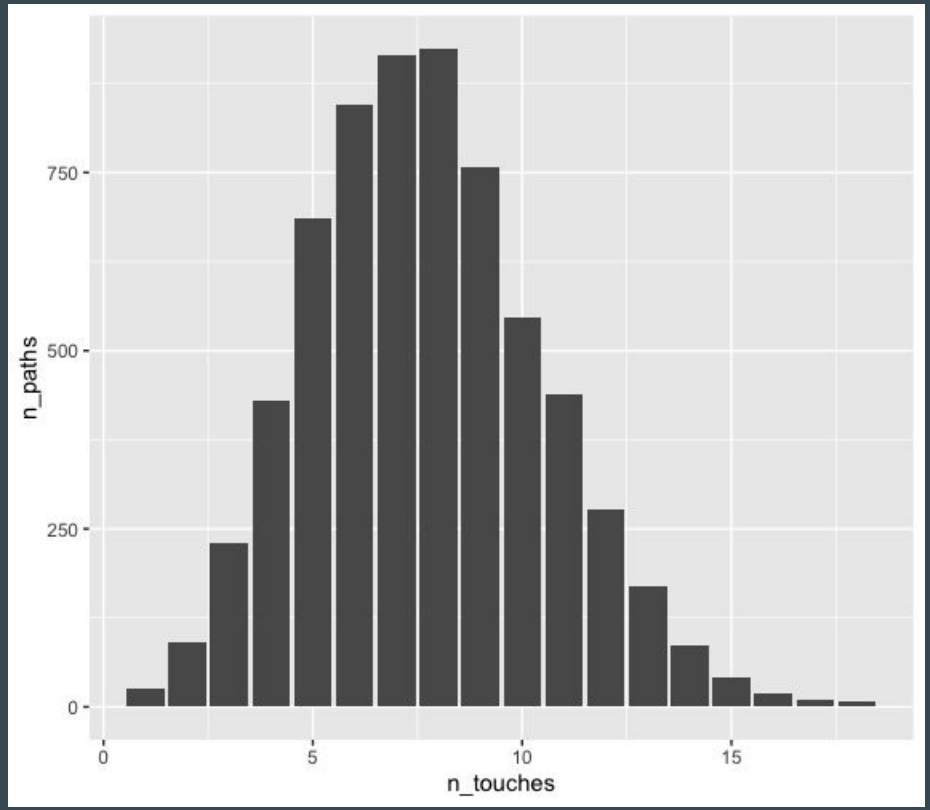
```
> head(data)
    id       date   channel
1 4138 2023-01-02 channel_C
2 5224 2023-04-24 channel_E
3 2961 2023-01-07 channel_C
4 7381 2023-02-22 channel_F
5 1230 2023-03-26 channel_E
6 1105 2023-01-07 channel_D
```

Dataframe Results

Simulates:

- ~6500 customers
- 50,000 marketing touchpoints
- 6 months of dat (Jan-June 2023)

```
data %>%
  group_by(id) %>%
  summarize(n_touches=n()) %>%
  ungroup() %>%
  group_by(n_touches) %>%
  summarise(n_paths = n()) %>%
  ggplot(aes(x=n_touches, y=n_paths)) +
    geom_col()
```



Convert dataframe to necessary structure

```r
# convert to necessary format
data_mod <- data %>%
  arrange(id, date) %>%
  group_by(id) %>%
  summarise(path = paste(channel, collapse = ' > ')) %>%
  ungroup() %>%
  mutat

  ungro
```

```
> head(data_mod)
# A tibble: 6 x 4
     id path                                                                  conv conv_null
  <int> <chr>                                                                <int>     <dbl>
1  1000 channel_B > channel_C > channel_B > channel_A > channel_D > channel_F …     1         0
2  1001 channel_A > channel_A > channel_A > channel_B > channel_A > channel_A …     1         0
3  1002 channel_E > channel_A > channel_A > channel_B > channel_E > channel_B …     1         0
4  1003 channel_C > channel_F > channel_D > channel_F > channel_E > channel_D       1         0
5  1004 channel_E > channel_B > channel_C > channel_D > channel_D                   0         1
6  1005 channel_F > channel_C > channel_E > channel_E > channel_F > channel_D …     0         1
```

Very Simple Code to Generate Model Results

Using the ChannelAttribution package, we get results for single touch, multi touch and Markov chain attribution in two very simple lines of code.

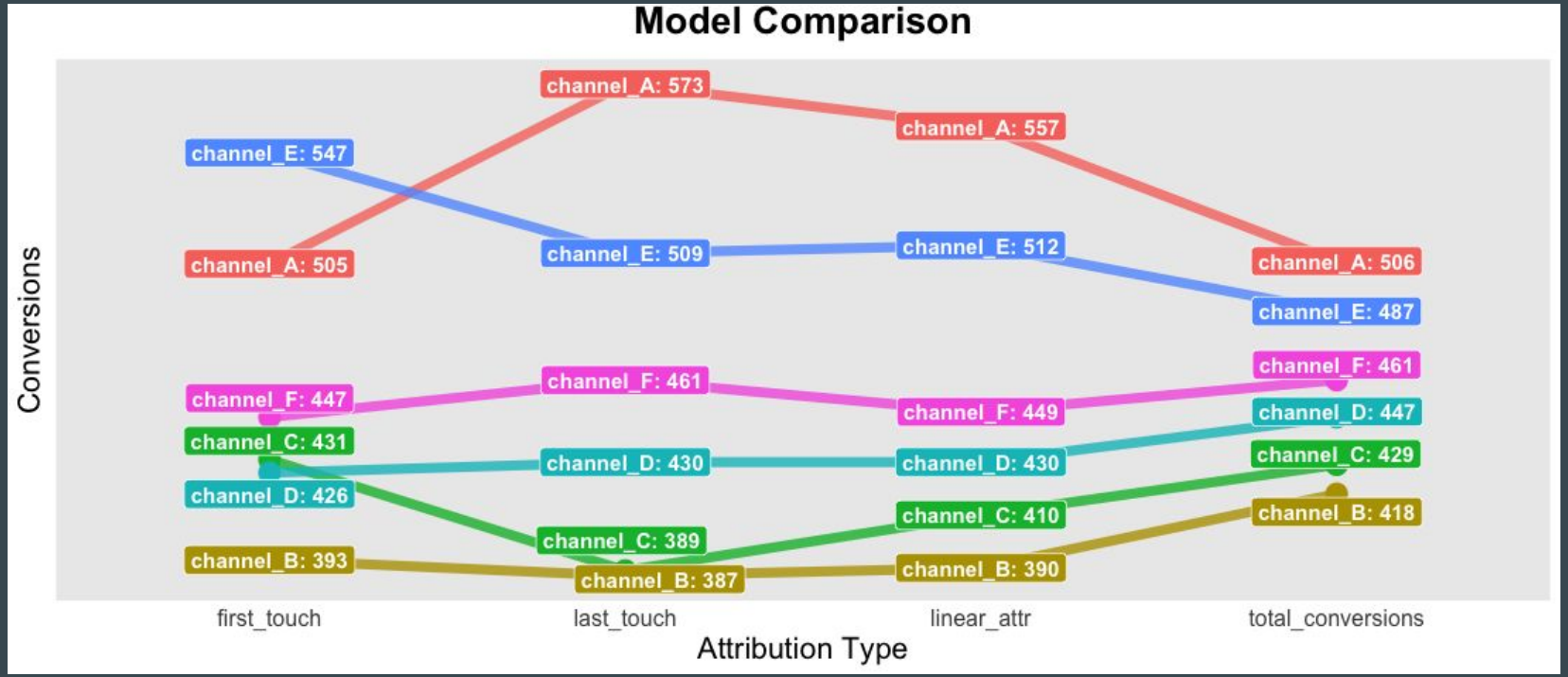
```
# single touch, multi touch attribution
h_mod <- heuristic_models(data_mod, var_path = 'path', var_conv = 'conv')


# markov attribution
m_mod <- markov_model(data_mod, var_path = 'path', var_conv = 'conv', var_null = 'conv_null', out_more = TRUE)
```

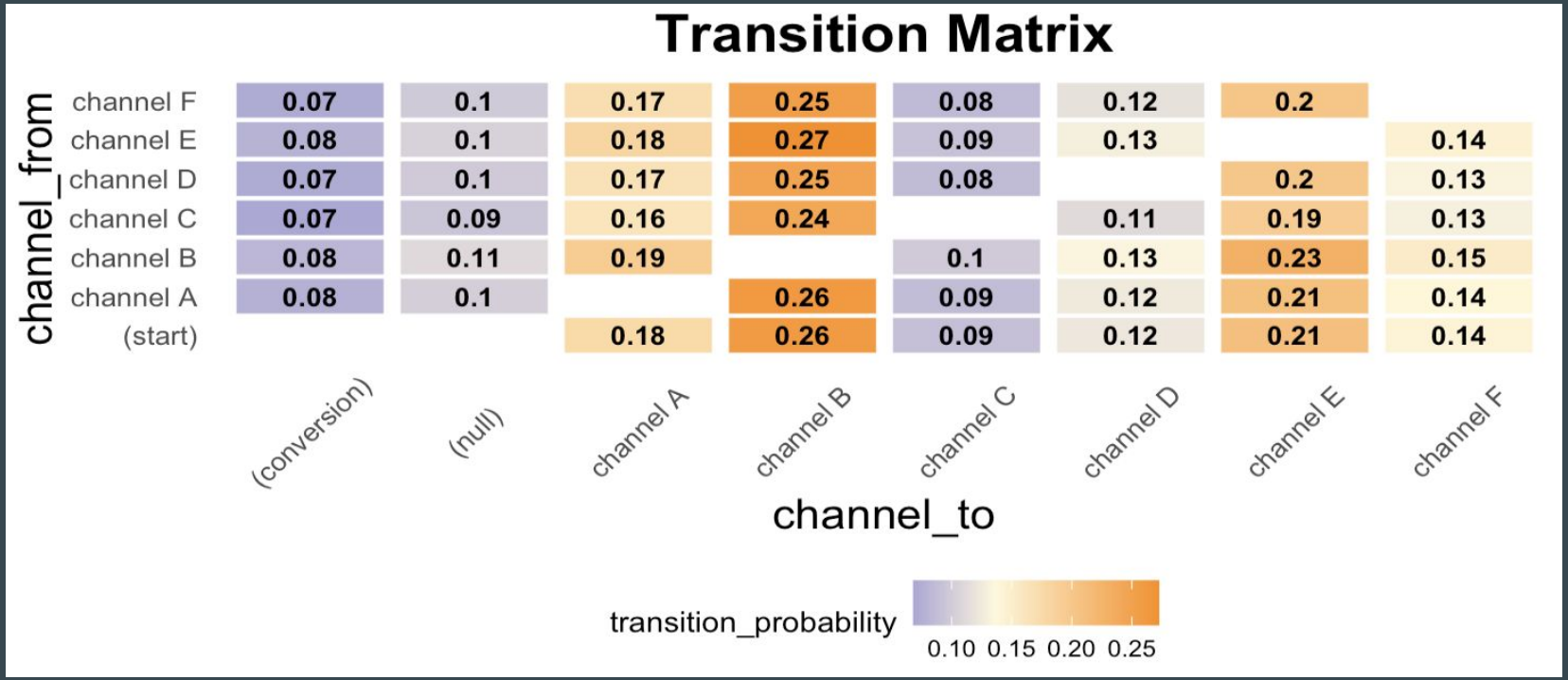
```
> h_mod
  channel_name first_touch last_touch linear_touch
1    channel_B         393        387     389.8151
2    channel_C         431        389     409.9723
3    channel_A         505        573     557.0717
4    channel_D         426        430     430.2783
5    channel_F         447        461     449.4014
6    channel_E         547        509     512.4611
```

```
> m_mod
$result
  channel_name total_conversions
1    channel_B          418.4474
2    channel_C          428.8089
3    channel_A          506.1221
4    channel_D          447.3402
5    channel_F          460.8899
6    channel_E          487.3916
```

Result: Attribution Comparison using ggplot2



Result: Transition Matrix using ggplot2



Considerations

- Can you get access to the data you need, or can you estimate it?
- Markov chain assumes each state is independent from the last and the next; if your customer journey has dependent steps, consider modeling before and/or after those steps.
- Determine when a customer journey restarts: 6 months? 3 years? After each purchase?

Resources/ References

GitHub: JessicaMDuncan/2023_NYR

Packages

- dplyr
- stringr
- ggplot2
- reshape2
- ggrepel
- ChannelAttribution

Reference: AnalyzeCore by Serhii Bryl