Multi-channel Advertising

Four methods for measuring advertising response

Dr. Junbum Kwon

29 March 2019

Author: Elea McDonnell Feit, Professor at Drexel University and Senior Fellow at Wharton Customer Analytics Initiative

Topics

Why advertising response?

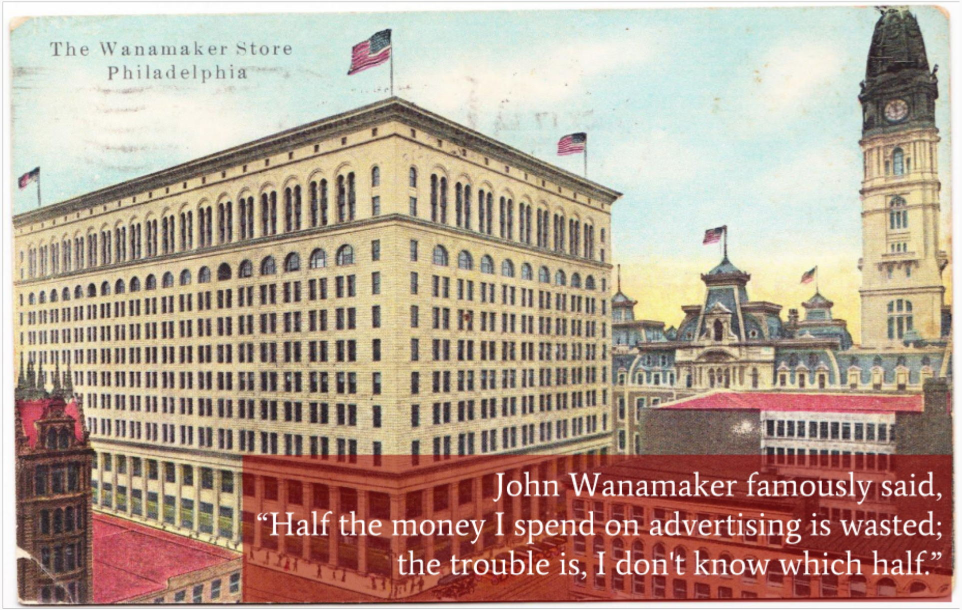
Attribution rules

Holdout testing

Marketing mix modeling

Model-based attribution

Why advertising response?



Source: Papergreat blog

Measuring advertising response

The goal of any marketing campaign is to increase sales (short-term or long-term).

Measuring advertising response

The goal of any marketing campaign is to increase sales (short-term or long-term).

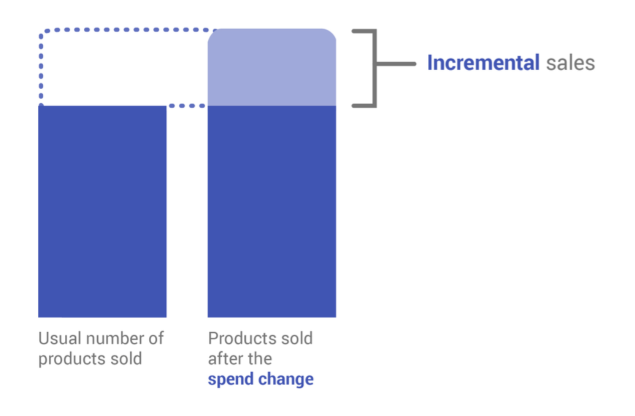
In theory, it should be easy to evaluate the performance of marketing. Each campaign or marketing channel should be evaluated based on the incremental profit that it produces relative to its cost.

$$ROI = \frac{\text{incremental profit due to advertising} - \text{cost of advertising}}{\text{cost of advertising}}$$

ROI = Return on Investment

Incremental sales

Incremental profit depends on incremental sales which are the additional sales we make with advertising over an above what we would have sold without advertising. Incremental profit is typically a function of the incremental sales.

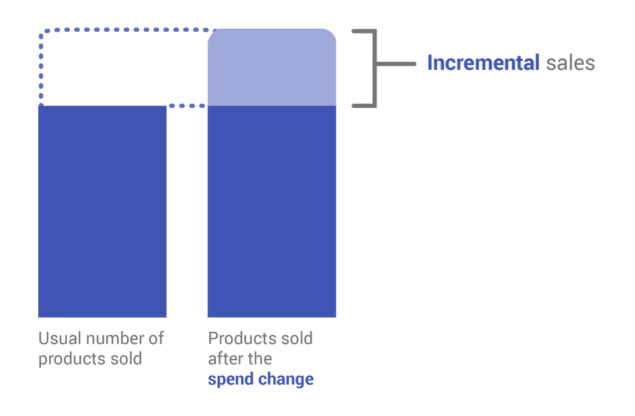


Source: Think with Google

Incremental sales

Incremental profit depends on incremental sales which are the additional sales we make with advertising over an above what we would have sold without advertising. Incremental profit is typically a function of the incremental sales.

With today's digital media – where we can track individual users – it is now possible to estimate incremental sales in many situations.



Source: Think with Google

Goals

Understand four popular techniques for estimating the incremental sales due to advertising.

- Attribution rules
- Holdout testing
- Marketing mix modeling (with aggregate spending and sales data)
- Attribution modeling (with user-level data)

Learn how each technique works using the Python.

Understand which techniques produce better estimates of incremental profits and why.

Why advertising response?

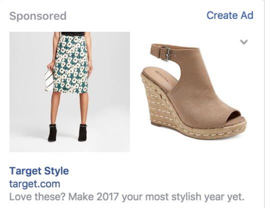
Data

Data: retail impressions and conversions

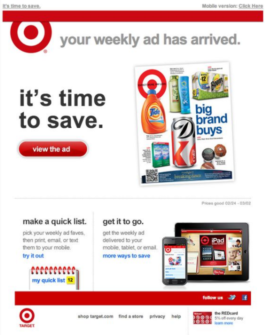
We will be analyzing a data set describing 10,000 customers and potential customers of a retailer. The retailer uses four different advertising channels.

Display

Social



Email



Direct (via mail)

Customer tracking begins when the customer is exposed to a display or social ad, visits the retailer website or makes a purchase.

Data structure

The data is organized in three files:

- `customer.csv`: each row is a customer, 10,000 rows
- `impressions.csv`: each row is an exposure of a marketing communication to a specific customer, 501,336 rows
- `transactions.csv`: each row is a transaction made by a customer

Data structure

The data is organized in three files:

- `customer.csv`: each row is a customer, 10,000 rows
- `impressions.csv`: each row is an exposure of a marketing communication to a specific customer, 501,336 rows
- `transactions.csv`: each row is a transaction made by a customer

This basic structure is typical of raw digital advertising data. The data is hypothetical, and has been designed to illustrate important points in data analysis.

Customer file

The customer file can be downloaded from goo.gl/fyxRE5. Each row describes a customer.

Columns:

`id`: an id number for the customer

`past purchase`: whether the customer has made a purchase prior to the observation period.

`email`: indicates whether the customer is eligible to receive emails, i.e. we have an email address and permission to mail

`direct`: indicates whether the customer is eligible to receive direct mail, i.e. we have an address and permission to mail

Read and inspect customer file

We can read the file directly into R:

```
> cust <- read.csv("https://goo.gl/mqy8NR")
> nrow(cust)
[1] 10000
> summary(cust)
       id        past.purchase        email            direct      
 Min.   :    1   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
 1st Qu.: 2501   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000  
 Median : 5000   Median :1.0000   Median :1.0000   Median :0.0000  
 Mean   : 5000   Mean   :0.5022   Mean   :0.6001   Mean   :0.4974  
 3rd Qu.: 7500   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000  
 Max.   :10000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
```

We have 10,000 customers and about half of those have made a purchase. About 60% are eligible for email and 50% are eligible for direct mail.

More inspection of customer file

```
> xtabs(~email+direct+past.purchase, data=cust)
, , past.purchase = 0
      direct
email     0    1
    0  2775  707
    1  1205  291
, , past.purchase = 1
      direct
email     0    1
    0    95  422
    1   951 3554
```

Most customers who have not made a purchase are not eligible for email or direct mail. Customers who have made a purchase are more likely to be eligible.

Impressions file

The impressions file can be downloaded from https://goo.gl/LT93AA. Each row in the file represents an exposure of one customer to an ad, i.e. an impression.

Columns:

`id`: id number for the customer

`date`: date of impression. Most files would have a date-time, but we have simplified for the workshop.

`channel`: channel of the ad exposure

`click`: indicates whether the customer clicked on the ad

Read and inspect impressions file

```
> impress <- read.csv("https://goo.gl/74qIxy")
> impress$date <- as.Date(impress$date) # change type
> nrow(impress)
[1] 501336
> summary(impress)
       id              date                      channel             click
 Min.   :    1   Min.   :2016-12-31   direct        :  9948   Min.   :0.00000
 1st Qu.: 2467   1st Qu.:2017-01-10   display       :216371   1st Qu.:0.00000
 Median : 4940   Median :2017-01-20   email         : 38426   Median :0.00000
 Mean   : 4960   Mean   :2017-01-22   email.holdout:  9582   Mean   :0.01854
 3rd Qu.: 7454   3rd Qu.:2017-01-31   social        :227009   3rd Qu.:0.00000
 Max.   :10000   Max.   :2017-02-27                           Max.   :1.00000
```

We have 501,336 impressions between 2016-12-31 to 2017-02-27. The majority of these impressions are social or display. Overall click rate is 1.85%, which is reasonable.

Summarize the cadence of the impressions

Cadence is the timing of advertising impressions. It is useful to understand the cadence of your ads before you do any other analysis. Email and direct are often sent out to users on specific dates. Display and social ads can be steady or can be turned on and off at specific times. We can summarize the cadence with the crosstab function in R.

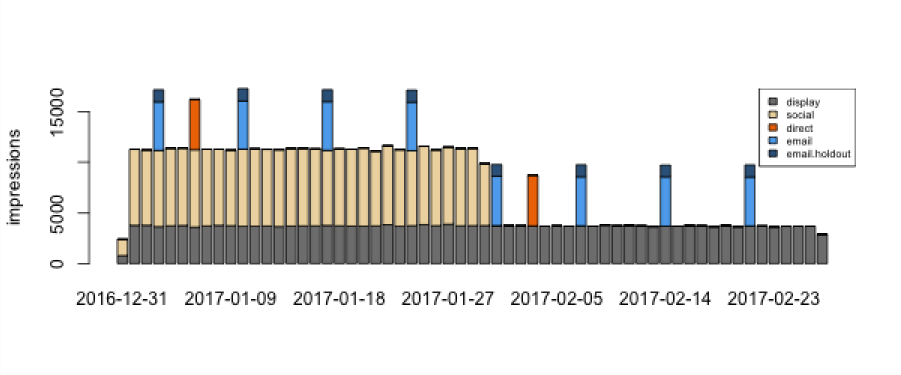
```
> (cadence <- xtabs(~date+channel, data=impress))
            channel
date         direct display email email.holdout social
  2016-12-31      0     788     0             0   1610
  2017-01-01      0    3786     0             0   7481
  2017-01-02      0    3792     0             0   7416
  2017-01-03      0    3656  4798          1203   7505
  2017-01-04      0    3731     0             0   7648
  ...
```

Visualizing the cadence

It is useful to create plot of the cadence data.

```
> cadence <- cadence[,c(2,5,1,3,4)] # reorder the columns
> mycols <- c("gray50", "wheat2", "darkorange2", "steelblue2", "steelblue4")
> barplot(t(cadence), col=mycols, ylab="impressions")
> legend("topright", legend=colnames(cadence), fill=mycols, cex=0.6)
```

Visualizing the cadence



Display impressions per day are steady across the observation window.

Social is steady in the first month and then stops.

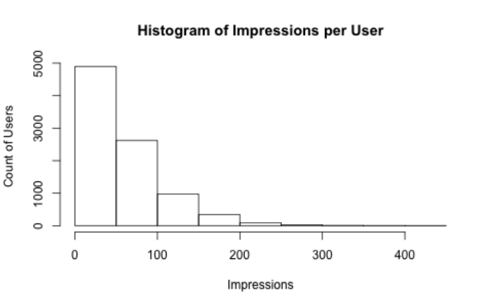
Emails are sent once per week and each campaign seems to have a holdout.

Direct mail is sent once per month.

Histogram of impressions

It is also useful to understand how many impressions each customer gets.

```
> hist(xtabs(~id, data=impress), xlab="Impressions", ylab="Count of Users",
+       main="Histogram of Impressions per User")
```



Some customers receive as many as 400 impressions in two months, but most get less than 100 impressions.

Click through rates by channel

```
> xtabs(click~channel, data=impress)/xtabs(~channel, data=impress)
channel
      direct      display        email email.holdout       social
 0.000000000  0.004783451  0.099672097   0.000000000  0.019519050
```

The click through rates are highest for email at 10.0% and lowest for display at 0.5%. As expected, there are no clicks for direct or email holdouts.

Transactions file

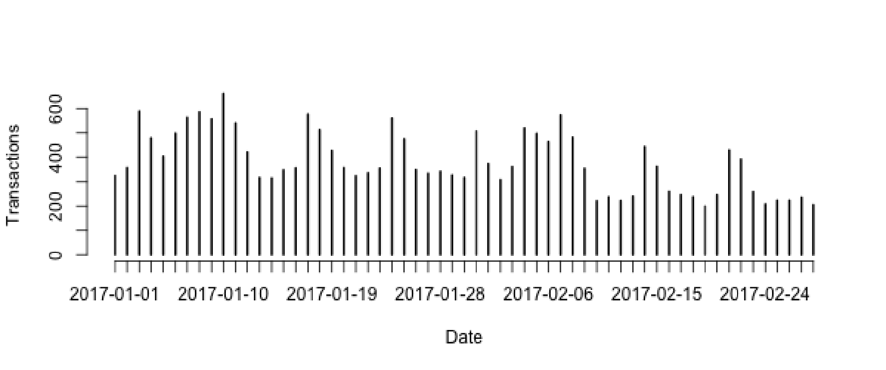
The transactions file can be downloaded from https://goo.gl/ZErZ6g. Each row in the file represents a purchase made by a customer.

Columns:
`id`: id number for the customer
`date`: date of transactions. Most files would have a date-time, but we have simplified for the workshop.
`last.touch`: channel of the last ad impression the customer saw before the transaction
`last.click`: channel of the last ad the customer clicked before the transaction

Purchase amounts can be used in holdout testing, attribution models or marketing mix models, but we will focus on transaction counts for this workshop.

Plot of transactions over time

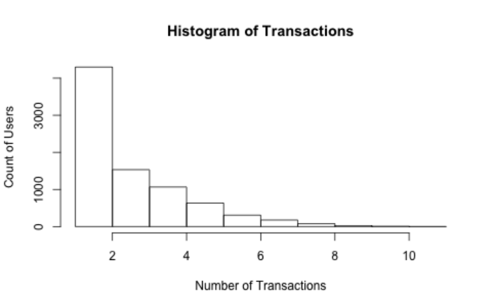
```
> plot(transbyday, ylab="Transactions", xlab="Date")
```



Transactions appear to be a bit higher in the first month and there are clear spikes around the time of emails and direct mail.

Histogram of transactions

```
> hist(xtabs(~id, data=trans), xlab="Number of Transactions", ylab="Count of Users",
+       main="Histogram of Transactions")
```



Most customers have less than 2 transactions in the two month observation period.

Customer 100

To understand how the files fit together, it is helpful to look at the impressions and transactions for one customer.

```
> cust[cust$id==100,]
     id past.purchase email direct
100 100             0     0      0
> impress[impress$id==100,]
[1] id      date    channel click
<0 rows> (or 0-length row.names)
> trans[trans$id==100,]
      X  id       date last.touch last.click
206 206 100 2017-01-18       none       none
207 207 100 2017-01-26       none       none
```

Customer 100 has no impressions and made two transactions.

Customer 300

```
> cust[cust$id==300,]
     id past.purchase email direct
300 300             1     1      1
> summary(impress[impress$id==300,])
       id            date                       channel        click
 Min.   :300   Min.   :2016-12-31   direct       : 2   Min.   :0
 1st Qu.:300   1st Qu.:2017-01-10   display      :81   1st Qu.:0
 Median :300   Median :2017-01-29   email        : 8   Median :0
 Mean   :300   Mean   :2017-01-28   email.holdout: 0   Mean   :0
 3rd Qu.:300   3rd Qu.:2017-02-14   social       : 0   3rd Qu.:0
 Max.   :300   Max.   :2017-02-26                      Max.   :0
> trans[trans$id==300,]
      X  id       date last.touch last.click
647 647 300 2017-01-03    display       none
648 648 300 2017-01-10      email       none
649 649 300 2017-02-04     direct       none
650 650 300 2017-02-19    display       none
```

Customer 300 has 8 email, 2 direct mail and 81 display impressions and made 4 transactions.

Steps in data inspection

1. For each file,
   1.1 Make sure you have the entire file read in by checking number of rows.
   1.2 Summarize individual variables in file mean, min, max, etc. Do they make sense?
   1.3 Summarize relationships between variables in file using crosstabs or scatterplots. Do they make sense?
2. Check that joins between files are as expected (skipped).
3. Summarize relationships between files. Do they make sense?

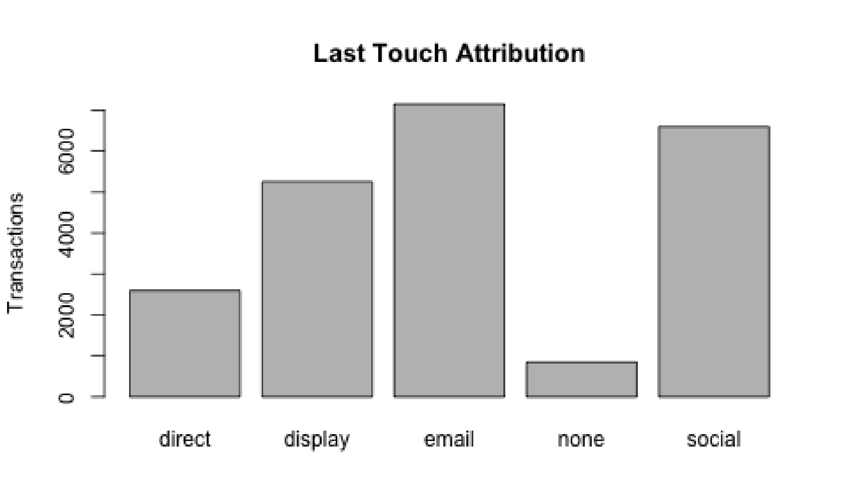
Attribution rules

Attribution rules

What is last click?

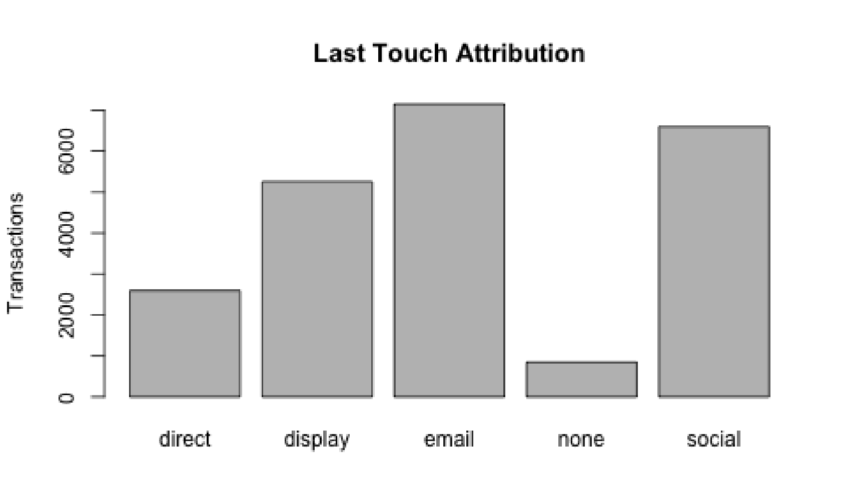
Last-touch attribution

Last-click (or last-touch) attribution looks backward from each conversion to find the last ad the user clicked on prior to the conversion.



Last-touch attribution

Last-click (or last-touch) attribution looks backward from each conversion to find the last ad the user clicked on prior to the conversion.



Attribution rules

Last touch analysis

Last touch analysis

The last click and the last touch are stored in the transaction file.

```
> head(trans)
  id       date last.touch last.click
1  2 2017-01-04      email       none
2  2 2017-02-12      email       none
3  3 2017-02-02      email       none
4  3 2017-02-14      email       none
5  5 2017-01-04    display      email
6  5 2017-01-13    display      email
```

One advantage of last-click is that it is easy to compute once this pre-processing is done.

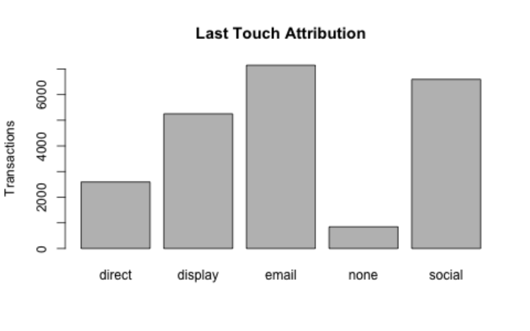
Last touch analysis

For example, if we want to compute the number of transactions that are attributed to each channel by last touch, we just do a quick crosstab on the transaction table.

```
> (last.touch.att <- xtabs(~last.touch, data=trans))
last.touch
 direct display   email    none  social
   2594    5252    7145     846    6596
> barplot(last.touch.att, ylab="Transactions",
+          main="Last Touch Attribution")
```

When we do this, we are ignoring all the customers who didn't transact.

Last touch analysis



Many people interpret this as meaning that the incremental sales for social are 6596.

Last touch for subgroups of transactions

It is also easy to compute the last touch for a subgroup of the transactions. Simply subset out the transactions of interest and then crosstab.

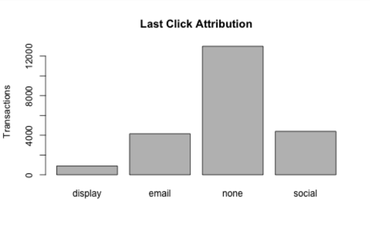
```
> xtabs(~last.touch, data=trans[trans$date>as.Date("2017-01-31"),])
last.touch
 direct display   email    none  social
   1669    2682    4081     258     328
```

In February, there were far fewer sales attributed to social, most likely because we ended the social ads at the end of January.

Last click analysis

You can do the same analysis based on the last ad a customer clicked, rather than the last touch/impression.

```
> (last.click.att <- xtabs(~last.click, data=trans))
last.click
display   email    none  social
    891    4151   12998    4393
> barplot(last.click.att, ylab="Transactions",
+         main="Last Click Attribution")
```

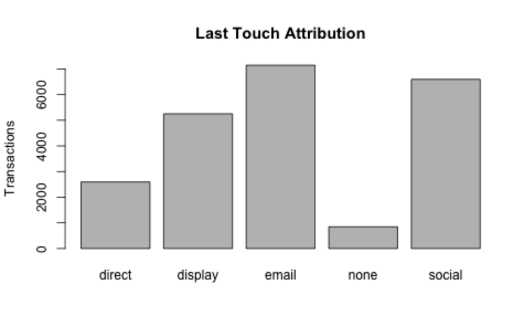


Since people don't click very much, the attributed sales are much lower. Measuring ad performance based on clicks doesn't make much sense for the advertiser.

Attribution rules

Limitations of last-click

Last touch analysis



Many people interpret this as meaning that the incremental sales for social are 6596.

Incorrect assumptions behind last touch

When we use last touch to estimate the incremental sales for an ad, we are making two mistakes:

1. Other ads may have influenced the customer and contributed to the sale.

Incorrect assumptions behind last touch

When we use last touch to estimate the incremental sales for an ad, we are making two mistakes:

1. Other ads may have influenced the customer and contributed to the sale.
    - This is the problem everyone seems to recognize.

Incorrect assumptions behind last touch

When we use last touch to estimate the incremental sales for an ad, we are making two mistakes:

1. Other ads may have influenced the customer and contributed to the sale.
   - This is the problem everyone seems to recognize.
2. It counts all the sales as incremental, i.e. it assumes that customers who saw ads would not have bought if they hadn't seen the ads. (Recall treatment vs control group)

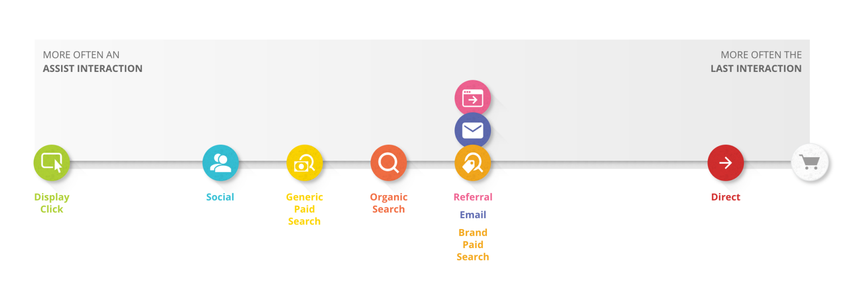
Incorrect assumptions behind last touch

When we use last touch to estimate the incremental sales for an ad, we are making two mistakes:

1. Other ads may have influenced the customer and contributed to the sale.
   - This is the problem everyone seems to recognize.
2. It counts all the sales as incremental, i.e. it assumes that customers who saw ads would not have bought if they hadn't seen the ads. (Recall treatment vs control group)
   - This issue is less well recognized.

Attribution as a payment mechanism

Because last-touch is so simple to compute, it has been used as a rule for allocating payments between channels on pay-for-conversion ads.



Source: Think with Google

Last touch unfairly favors channels that tend to show ads towards the end of the path to purchase such as search and retargeting.

Other attribution rules ("models")

Perhaps because everyone intuitively understood that last touch didn't account for "assists" and was unfair to channels that tend to occur early in the path to purchase they came up with other arbitrary rules.

Source: Across Health

Other attribution rules ("models")

Perhaps because everyone intuitively understood that last touch didn't account for "assists" and was unfair to channels that tend to occur early in the path to purchase they came up with other arbitrary rules.

Source: Across Health

These improved rules may help problem 1, but they don't help with problem 2.

Last touch is largely dead

There are many vendors such as Convertro, VisualIQ and MarketShare that have provided model-based attribution solutions.

Source: Google Analytics 360

In Google Analytics 360, released in 2016, Google also abandoned rule-based attribution in favor of model-based approaches.

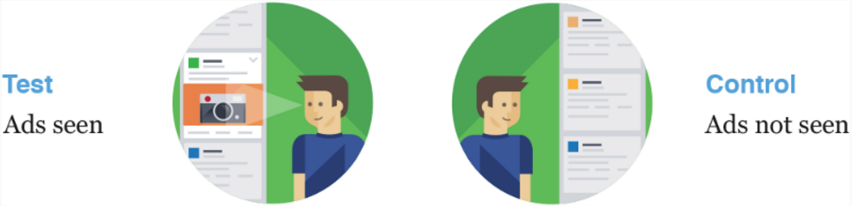
Holdout testing

Holdout testing

What is a holdout test?

Holdout testing

The gold standard for measuring incremental sales is an experiment, where we randomly assign customers to be exposed or not exposed to an ad.



Source: Designing with Science, medium.com

For example, with email we can take our list of target emails and randomly select a set of them to not receive an email. Facebook also has a tool for randomized holdouts (for large advertisers.)

Why holdout testing works

The magic behind a holdout test is the randomization. Another name for holdout testing is randomized controlled trial.

Why holdout testing works

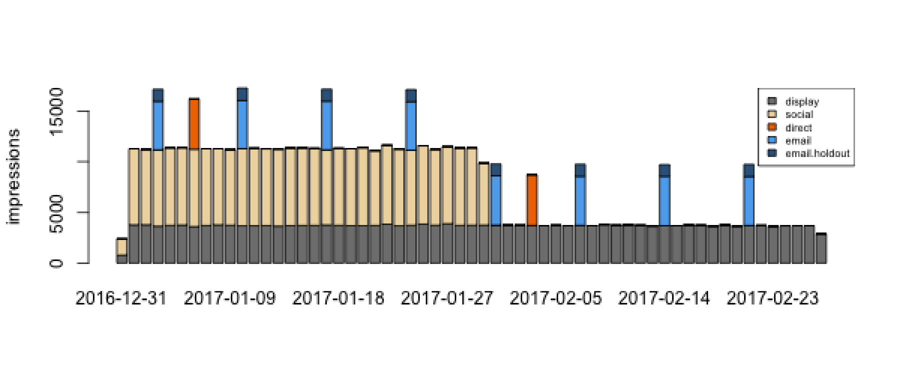
The magic behind a holdout test is the randomization. Another name for holdout testing is randomized controlled trial.

By randomly selecting the customers for the control group, we assure that the two groups are the same on average. The treatment and control groups will be similar in their propensity to purchase and response to ads. Statisticians call this probabilistically equivalent.

Holdout testing

Holdout test analysis

Email holdout tests in the example data



There was an email on 2017-01-03 that included a holdout group. Let's analyze this test.

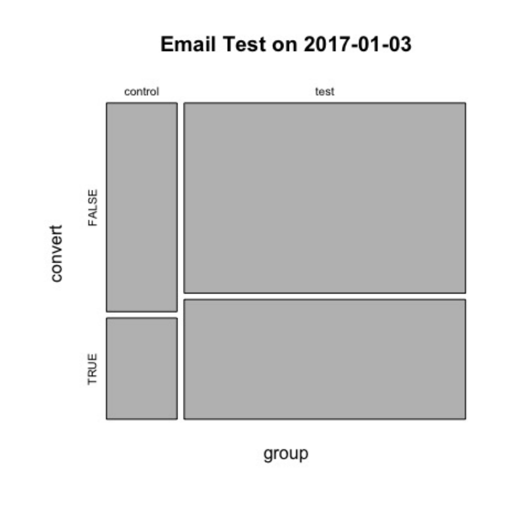
Function for analyzing holdout tests

```
holdout.test <- function(test.date, delay=0, window, impress, trans) {
  test.ids <- unique(impress$id[impress$channel=="email" & impress$date==test.date])
  control.ids <- unique(impress$id[impress$channel=="email.holdout" & impress$date==test.date])
  tdata <- data.frame(ids=c(test.ids, control.ids))
  tdata$group[tdata$id %in% test.ids] <- "test"
  tdata$group[tdata$id %in% control.ids] <- "control"
  in.window <- trans$date>=(test.date+delay) & trans$date<(test.date+window+delay)
  tdata$convert <- tdata$id %in% trans$id[in.window]
  ttable <- xtabs(~ group + convert, data=tdata)
  mosaicplot(~ group + convert, data=tdata,
             main=paste("Email Test on", test.date))
  proptest <- prop.test(x=ttable[,"TRUE"], n=xtabs(~group, data=tdata))
  diff.conv <- c(diff=proptest$estimate[2]-proptest$estimate[1], ci=-proptest$conf.int)
  out <- list(diff.conv, ttable, proptest)
}
```

Analysis of email holdout test

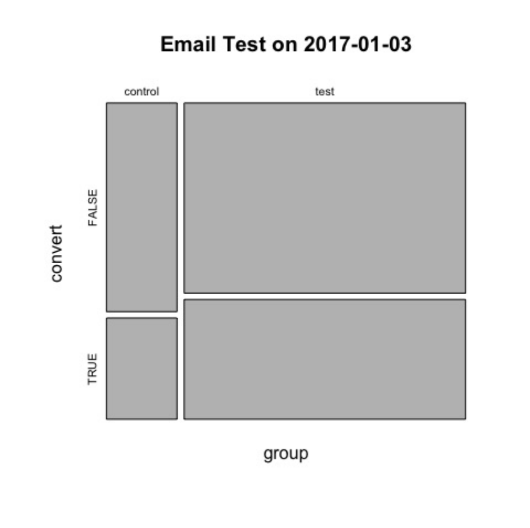
```
> (holdout.test(test.date=as.Date("2017-01-03"), window=7, impress=impress,
                trans=trans))
[[2]]
         convert
group      FALSE TRUE
  control    810  393
  test      2944 1854

[[3]]

        2-sample test for equality of proportions with
        continuity correction

data:  ttable[, "TRUE"] out of xtabs(~group, data = tdata)
X-squared = 14.395, df = 1, p-value = 0.0001482
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.09011755 -0.02933788
sample estimates:
   prop 1    prop 2
0.3266833 0.3864110
```

Reporting of email holdout test



The test group had a 38.6% conversion rate in the 7 days after the email was sent, versus a 32.7% conversion rate for the control group.

Reporting of email holdout test



The test group had a 38.6% conversion rate in the 7 days after the email was sent, versus a 32.7% conversion rate for the control group.

The email on 2017-01-03 produced incremental sales. The incremental increase in conversion rate is between 5.9%.

Another analysis of email holdout test on 2017-01-01

```
 > (holdout.test(test.date=as.Date("2017-01-03"), window=3, impress=impress, trans=trans))
[[2]]
         convert
group      FALSE TRUE
  control  1063  140
  test     3850  948

[[3]]
        2-sample test for equality of proportions with continuity correction

data:  ttable[, "TRUE"] out of xtabs(~group, data = tdata)
X-squared = 42.187, df = 1, p-value = 0.00000000008295
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.10306428 -0.05934891
sample estimates:
   prop 1    prop 2
0.1163757 0.1975823
```

Setting the response window

If you change the response window, you will get a different answer about how much advertising increases sales.

Setting the response window

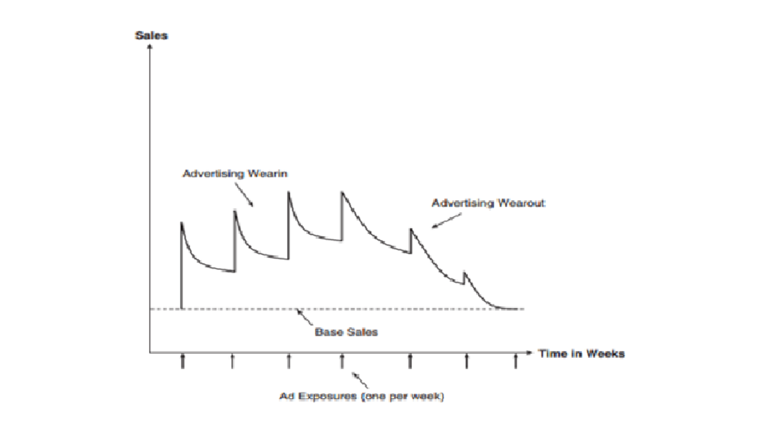
If you change the response window, you will get a different answer about how much advertising increases sales.

- For the email test on 2017-01-03, a response window of 3 days indicated a increase in conversion of 8.1% versus 6.0% for a 7 day window.

What is happening here?

Checking ad response over time

Ad response is often greatest just after exposure and then falls off over time.



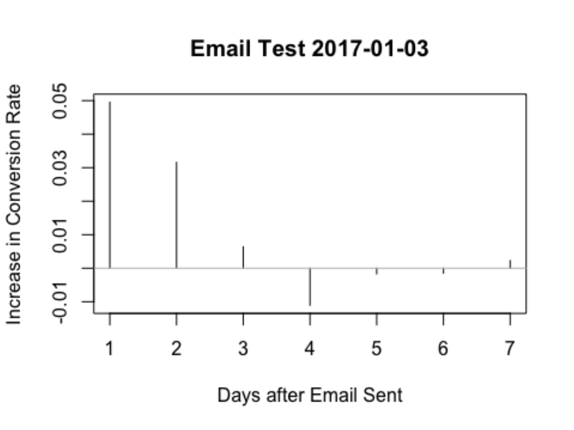
Source: Michael Wolfe

Ad response over time for 2017-01-03 email test

With a holdout test, you can study the ad response over time to learn what the pattern looks like over time.

```
day1 <- holdout.test(test.date=as.Date("2017-01-03"), delay=0, window=1,
impress=impress, trans=trans)
...
day7 <- holdout.test(test.date=as.Date("2017-01-03"), delay=6, window=1,
impress=impress, trans=trans)
incr.conv <- c(day1[[1]][1], day2[[1]][1], day3[[1]][1], day4[[1]][1], day5[[1]][1],
               day6[[1]][1], day7[[1]][1])
plot(incr.conv, type="h", xlab="Days after Email Sent", ylab="Increase in Conversion Rate")
abline(h=0)
```

Ad response over time for 2017-01-03 email test



In the test on 2017-01-03, the lift in conversion rate falls to zero in about three days.

Holdout testing

Planning a holdout test

Designing a holdout test

How do you measure response? For how long?

- Conversion (binary), Sales (number), or Site Visits (count)
- During or after campaign

What advertising do the treatment and control groups receive?

- How many ads does the treatment group see? Usual? Heavy up?
- What is used for the control? "Go dark"? alternative ad? Usual?

Which customers are included in the test?

- Current customers, good customers or prospects?

How should customers be assigned to treatment and control groups?

- Always random, but doesn't have to be an equal split.

Holdout testing

Example: Johnson, Lewis and Reiley (2017)

Display advertising holdout test at Yahoo!

Source: Johnson, Lewis and Reiley (2017)

During two weeks in Spring 2010, a national apparel retailer ran a test on Yahoo! to measure response to display ads on Yahoo!.

The ads displayed the retailer brand and brands of apparel firm, with slideshow transitions between photographs and text.

This test is described in detail in Johnson, Lewis and Reiley (2017) "When less is more: Data and power in advertising experiments", *Marketing Science* 36(1), 43-53.

Display advertising holdout test: design

How do you measure response? For how long?

- Sales ($) at the target retailer during the two week campaign

Display advertising holdout test: design

How do you measure response? For how long?

- Sales ($) at the target retailer during the two week campaign

What advertising do the treatment and control groups receive?

- Full: all retailer ads at all opportunities
- Control: house Yahoo! ads at all opportunities
- Half: half retailer and half house ads

Display advertising holdout test: design

How do you measure response? For how long?

- Sales ($) at the target retailer during the two week campaign

What advertising do the treatment and control groups receive?

- Full: all retailer ads at all opportunities
- Control: house Yahoo! ads at all opportunities
- Half: half retailer and half house ads

Which customers are included in the test?

- Registered Yahoo! users who were also customers of the retailer were assigned to groups at the start of the two week period. Many of these users never visit Yahoo! during the campaign and don't have the opportunity to see the ads.
- Database match allows us to track users between Yahoo! and the retailer.

Display advertising holdout test: design

How do you measure response? For how long?

- Sales ($) at the target retailer during the two week campaign

What advertising do the treatment and control groups receive?

- Full: all retailer ads at all opportunities
- Control: house Yahoo! ads at all opportunities
- Half: half retailer and half house ads

Which customers are included in the test?

- Registered Yahoo! users who were also customers of the retailer were assigned to groups at the start of the two week period. Many of these users never visit Yahoo! during the campaign and don't have the opportunity to see the ads.
- Database match allows us to track users between Yahoo! and the retailer.

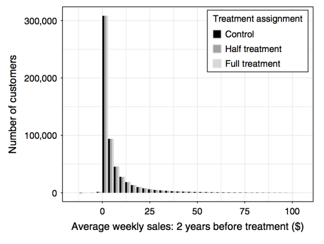
How should customers be assigned to treatment and control groups?

- Randomly, with one third in each group

Display advertising holdout test: randomization checks

It is a good idea to check that the treatment and control groups looked similar prior to the test.

	Treatment group			
	Full	Half	Control	*p*-value
Sample size	1,032,204	1,032,074	1,032,299	0.988
Female (mean)	68.5%	68.5%	68.5%	0.794
Age (mean)	43.6	43.6	43.6	0.607
Yahoo! page views[a] (mean)	245.8	244.4	243.5	0.132[d]
Pretreatment sales (two years, mean)	$857.74	$859.30	$855.54	0.475
Pretreatment sales (two weeks, mean)	$19.34	$19.24	$19.10	0.517
Treated subsample				
Exposed sample	572,574	571,222	570,908	0.254
Yahoo! page views (mean)	412.2	411.5	410.1	0.108
Ad views (mean)	33.42	33.41	33.66	0.164
Ad views (median)	15	15	15	
Retailer ad views (mean)	33.42	16.69	—	0.801
Control ad views (mean)	—	16.72	33.66	0.165
Retailer ad click-through rate[b] (%)	0.19	0.24	—	
Retailer ad clicker rate[c] (%)	4.91	3.39	—	



Source: Johnson, Lewis and Reiley (2017)

The three groups appear to be similar. So, the randomization worked and we are comparing "apples to apples".

Display advertising holdout test: result

Sales Increase for Full and Half Treatments ($)

	(1)	(2)
Subset of users[a]	Everyone	Treated
Sales after first ad exposure[b]		
Full treatment ($)	0.673**	0.525**
	(0.317)	(0.237)
Half treatment ($)	0.0248	0.189
	(0.311)	(0.235)
Constant ($)	15.52***	15.53***
	(0.122)	(0.166)

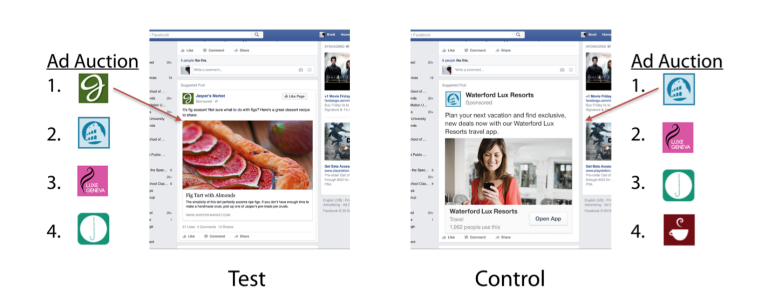
Looking at all the customers, the treatment increased sales by $0.67 (95% CI = [0.05, 1.29]). Among those who actually were exposed to ads, the effect was $0.52 (95% CI = [0.06,0.97]).

Holdout testing

Example: Gordon, Zettelmeyer, Bahargava and Chapsky (2016WP)

Social advertising holdout test

During two weeks in the first half of 2015, an omni-channel retailer conducted a test to determine the lift Facebook ads in the newsfeed.



Source: Gordon et al. (2016 WP)

This test is reported in Gordon, Zettelmeyer, Bhargava and Chapsky (2016WP) "A Comparison of Approaches to Advertising Measurement: Evidence from Big Field Experiments at Facebook"

Search advertising holdout test design

Which customers are included in the test?

- Facebook users that met the targeting criteria for the campaign

Search advertising holdout test design

Which customers are included in the test?

- Facebook users that met the targeting criteria for the campaign

How do you measure response? For how long?

- Purchases at the retailers website measured via Facebook conversion pixel during the campaign period and up to several weeks after the study ended.

Search advertising holdout test design

Which customers are included in the test?

- Facebook users that met the targeting criteria for the campaign

How do you measure response? For how long?

- Purchases at the retailers website measured via Facebook conversion pixel during the campaign period and up to several weeks after the study ended.

How should customers be assigned to treatment and control groups?

- Randomly, 70% test and 30% control

Search advertising holdout test design

Which customers are included in the test?

- Facebook users that met the targeting criteria for the campaign

How do you measure response? For how long?

- Purchases at the retailers website measured via Facebook conversion pixel during the campaign period and up to several weeks after the study ended.

How should customers be assigned to treatment and control groups?

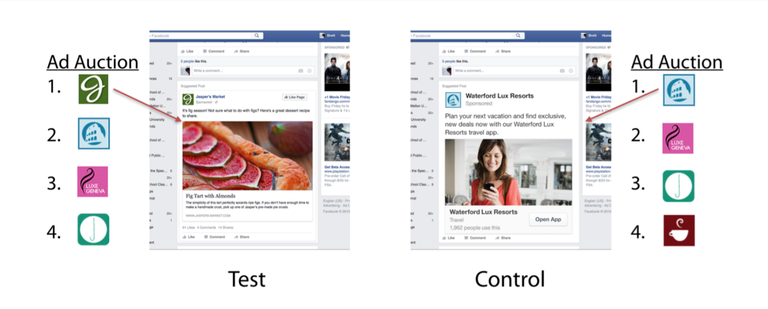
- Randomly, 70% test and 30% control

What advertising do the treatment and control groups receive?

- Test: user is eligible to see ads and will if the retailer wins the Facebook auction
- Control: competitor ads

Social advertising holdout test: Control condition

Since purchasing house ads or public service announcements (PSAs) is expensive, the Facebook experiment ran a competitor ad for the control.



Source: Gordon et al. (2016 WP)

By using the “next advertiser” as the control condition, the sales lift we estimate is relative to “what would have happened had we not done the campaign.”

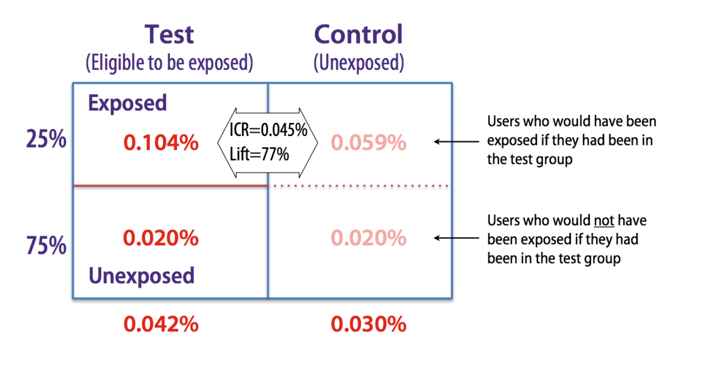
Social holdout test: Randomization check

Variable	Control group	Test group	p-value
Average user age	31.7	31.7	0.33
% of users who are male	17.2%	17.2%	0.705
Length of time using FB (days)	2,288	2,287	0.24
% of users with status "married"	19.6	19.6	0.508
% of users status "engaged"	13.8	13.8	0.0892
% of users status "single"	14.0	14.0	0.888
# of FB friends	485.7	485.7	0.985
# of FB uses in last 7 days	6.377	6.376	0.14
# of FB uses in last 28 days	25.5	25.5	0.172

Source: Gordon et al. (2016 WP)

The test and control groups appear to be similar. Randomization worked.

Social holdout test: Results



Source: Gordon et al. (2016 WP)

There is a 0.045% increase in conversions when comparing the users exposed to ads to the users in the control group that would have been exposed.

Facebook holdout testing tool: Conversion Lift

What is conversion lift?

Conversion lift accurately captures the impact that Facebook ads have in driving business for marketers. Here's how it works:

1. When creating a Facebook campaign, a randomized test group (people that see ads) and control group (people that don't) are established.
2. The advertiser securely shares conversion data from the campaign with Facebook. Typically, this data comes from sources like the Facebook Custom Audiences pixel, conversion pixel or secure point-of-sale (POS) data.
3. Facebook determines additional lift generated from the campaign by comparing conversions in the test and control groups.
4. The results of the study are made available in Ads Manager.

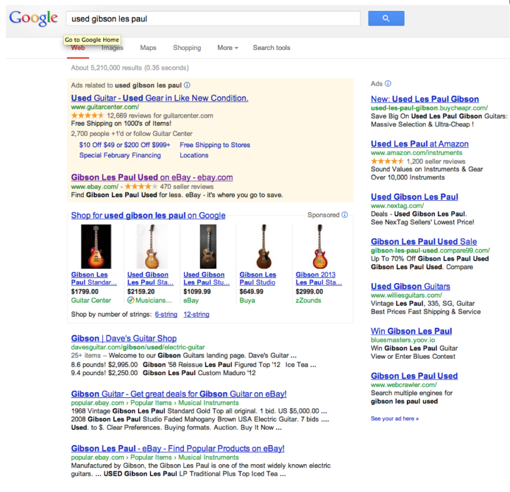
Source: Facebook Conversion Lift


There were 5,000 additional buyers as a result of your ads, an increase of 3.00%.
+ 5,000
Incremental Buyers
+ 3.00%
Conversion Lift
GROUP SIZE
CONVERSION RATE
BUYERS
TEST
800K
25.0%
200K
INCREMENTAL BUYERS
CONTROL
200K
4.0x
5,000
± 1,102
SCALED CONTROL
800K
24.4%
195K
STEP 1: COMPARE THE GROUPS
Scaling the control group to the same size as the test group provides an equal comparison between groups.
STEP 2: APPLY THE CONVERSION RATES
Applying the conversion rate from the control group to the scaled control shows how many people would have bought if the control group were the same size as the test group.
STEP 3: FIND THE INCREMENTAL BUYERS
Subtracting the buyers in the scaled control group from the buyers in the test group gives you the incremental buyers, the number of buyers directly attributable to your ads.
View Conversion Lift
Close this card

Holdout testing

Example: Blake, Nosko and Tadellis (2015)

Search advertising holdout test



Source: Blake, Nosko and Tadellis (2015)

In April-July 2012, eBay ran a holdout experiment to determine the sales lift of their search advertising targeting keywords other than “eBay”.

This test is reported in detail in Blake, Nosko and Tadelis (2015) “Consumer Heterogeneity and Paid Search Effectiveness: A Large-Scale Field Experiment”, *Econometrica* 83(1), 155-174.

Search advertising holdout test: Design

Which customers are included in the test?

- Google AdWords does not provide a testing tool that will randomize search ad exposure at the user level. Instead, they conducted a geo-test with 30% of the DMAs in the US market.

Search advertising holdout test: Design

Which customers are included in the test?

- Google AdWords does not provide a testing tool that will randomize search ad exposure at the user level. Instead, they conducted a geo-test with 30% of the DMAs in the US market.

How do you measure response? For how long?

- eBay sales ($) shipped to the DMA

Search advertising holdout test: Design

Which customers are included in the test?

- Google AdWords does not provide a testing tool that will randomize search ad exposure at the user level. Instead, they conducted a geo-test with 30% of the DMAs in the US market.

How do you measure response? For how long?

- eBay sales ($) shipped to the DMA

What advertising do the treatment and control groups receive?

- Test: keep search advertising at usual level
- Control: “go dark”

Search advertising holdout test: Design

Which customers are included in the test?

- Google AdWords does not provide a testing tool that will randomize search ad exposure at the user level. Instead, they conducted a geo-test with 30% of the DMAs in the US market.

How do you measure response? For how long?

- eBay sales ($) shipped to the DMA

What advertising do the treatment and control groups receive?

- Test: keep search advertising at usual level
- Control: "go dark"

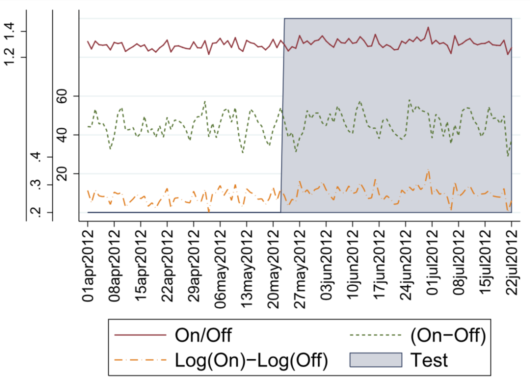
How should DMAs be assigned to treatment and control groups?

- DMAs were matched based on prior eBay sales and then randomly assigned to groups.

Search advertising holdout test: Findings

Ratio of Sales

Before and During Test Period

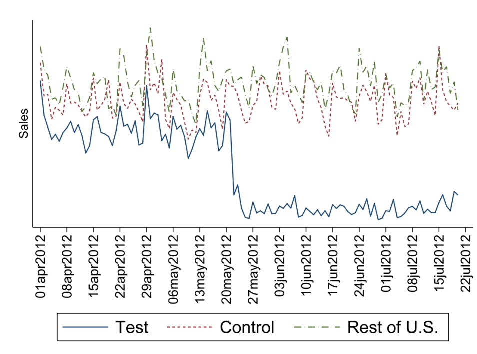


Source: Blake, Nosko and Tadellis (2015)

Search advertising does not increase eBay sales. As a result, eBay has drastically reduced search advertising.

Search advertising holdout test: "Last Touch"

"Last Touch" Attributed Sales
(sale within 24 hours of search ad exposure)
Before and During Test Period



Source: Blake, Nosko and Tadellis (2015)

If you were using last touch attribution, you would think that search advertising made a big difference.

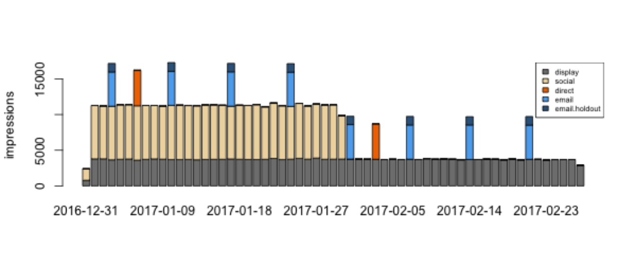
Marketing mix modeling

Marketing mix modeling

What is marketing mix modeling?

Marketing mix modeling

Before advertisers had access to user-level ad exposure data (like our example data set) they only had data on what they had spend on advertising and (sometimes) the number of people who saw the ad. This type of spending data is similar to the aggregate impressions data we looked at for our data set.



To find the correlation between total sales in each day/week/month to advertising spending or impressions on that same day/week/month, you estimate regression model.

A simple marketing mix model

A regression is an equation relating a response ("dependent variable") to one or more other variables ("independent variables").

A simple marketing mix model

A regression is an equation relating a response ("dependent variable") to one or more other variables ("independent variables").

A simple marketing mix model is:

$$\text{sales}_t = \beta_0 + \beta_1 \text{display}_t + \beta_2 \text{social}_t + \beta_3 \text{email}_t + \beta_4 \text{direct}_t + \epsilon_t$$

A simple marketing mix model

A regression is an equation relating a response ("dependent variable") to one or more other variables ("independent variables").

A simple marketing mix model is:

$$\text{sales}_t = \beta_0 + \beta_1 \text{display}_t + \beta_2 \text{social}_t + \beta_3 \text{email}_t + \beta_4 \text{direct}_t + \epsilon_t$$

The words represent data we have for each time period and the β's represent the unknown relationship between ad impressions and sales. For example β_1 is the increase in sales we get for each additional display impression.

A simple marketing mix model

A regression is an equation relating a response ("dependent variable") to one or more other variables ("independent variables").

A simple marketing mix model is:

$$\text{sales}_t = \beta_0 + \beta_1 \text{display}_t + \beta_2 \text{social}_t + \beta_3 \text{email}_t + \beta_4 \text{direct}_t + \epsilon_t$$

The words represent data we have for each time period and the β's represent the unknown relationship between ad impressions and sales. For example β_1 is the increase in sales we get for each additional display impression.

Using observations of the sales and advertising data, when can estimate these unknown parameters of the model.

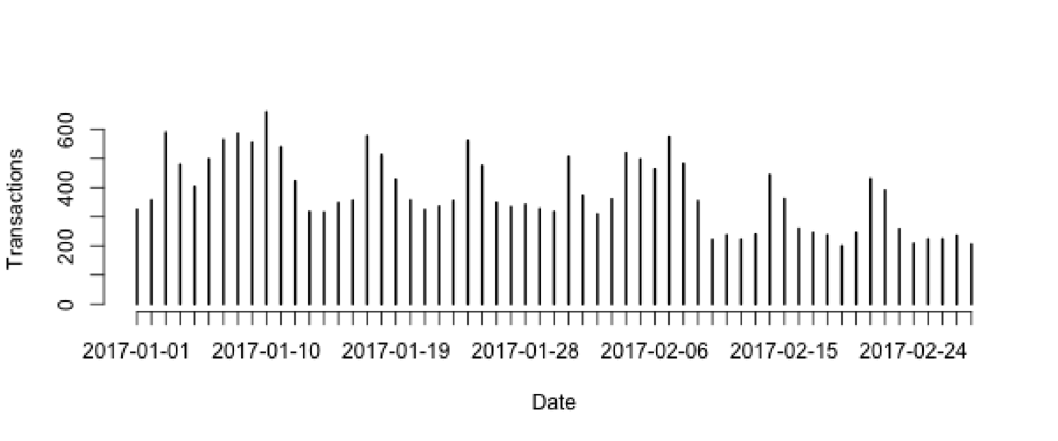
Marketing mix modeling

Aggregating the user-level sales data

As a reminder, we already computed the aggregate transactions for each day from our raw user-level data using a crosstab.

```
> (transbyday <- xtabs(~date, data=trans))
date
2017-01-01 2017-01-02 2017-01-03 2017-01-04 2017-01-05 2017-01-06 2017-01-07
       325        357        589        479        403        498        564
2017-01-08 2017-01-09 2017-01-10 2017-01-11 2017-01-12 2017-01-13 2017-01-14
       586        556        660        539        422        317        315
2017-01-15 2017-01-16 2017-01-17 2017-01-18 2017-01-19 2017-01-20 2017-01-21
       348        356        577        512        428        357        324
```

Sales by day



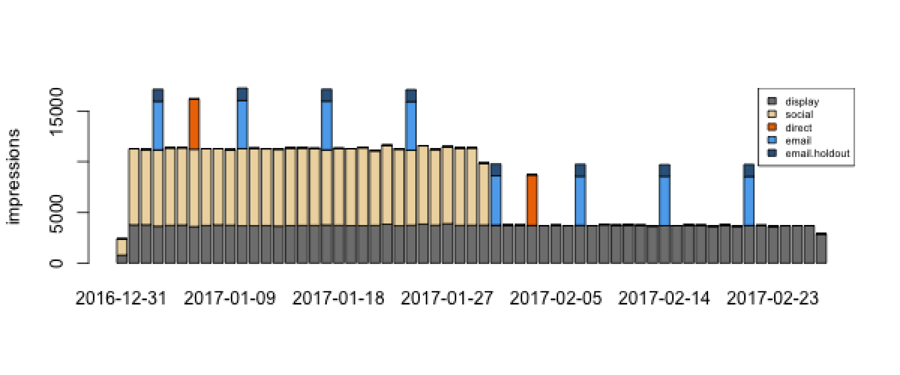
Transactions
0
200
400
600
2017-01-01
2017-01-10
2017-01-19
2017-01-28
2017-02-06
2017-02-15
2017-02-24
Date

Aggregating the user-level advertising data

We also computed the aggregate number of impressions on each day.

```
> (cadence <- xtabs(~date+channel, data=impress))
            channel
date          direct display email email.holdout social
  2016-12-31       0     788     0             0   1610
  2017-01-01       0    3786     0             0   7481
  2017-01-02       0    3792     0             0   7416
  2017-01-03       0    3656  4798          1203   7505
  2017-01-04       0    3731     0             0   7648
  2017-01-05       0    3770     0             0   7620
  2017-01-06    4974    3611     0             0   7614
  2017-01-07       0    3719     0             0   7552
  2017-01-08       0    3780     0             0   7504
  2017-01-09       0    3744     0             0   7446
```

Advertising by day



Combining the sales and impressions data

When we fit models, it will be convenient to put the sales and impressions data together in the same data frame.

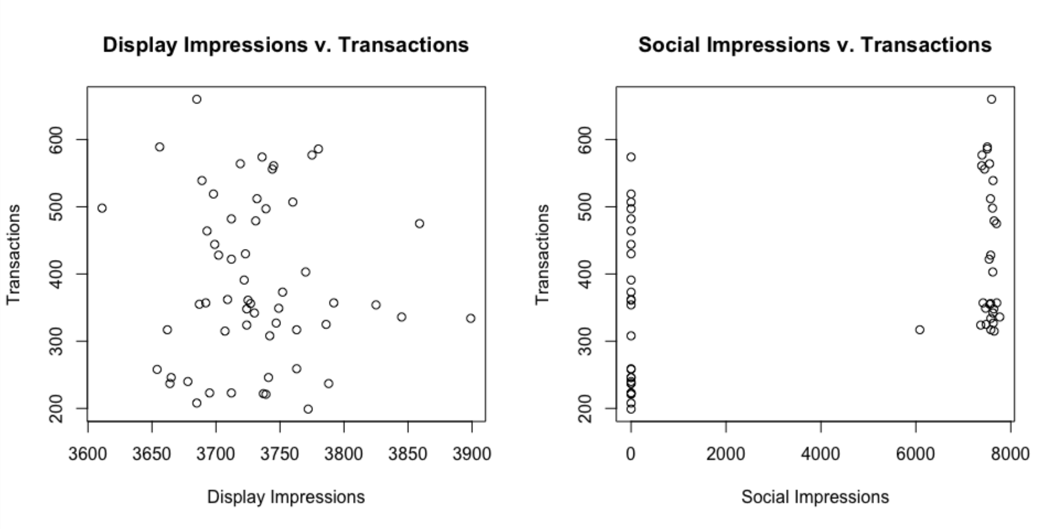
```
> mdata <- as.data.frame(cbind(trans=transbyday[1:57], cadence[2:58,])) # aligining
> head(mdata)
           trans direct display email email.holdout social
2017-01-01   325      0    3786     0             0   7481
2017-01-02   357      0    3792     0             0   7416
2017-01-03   589      0    3656  4798          1203   7505
2017-01-04   479      0    3731     0             0   7648
2017-01-05   403      0    3770     0             0   7620
2017-01-06   498   4974    3611     0             0   7614
```

Data exploration

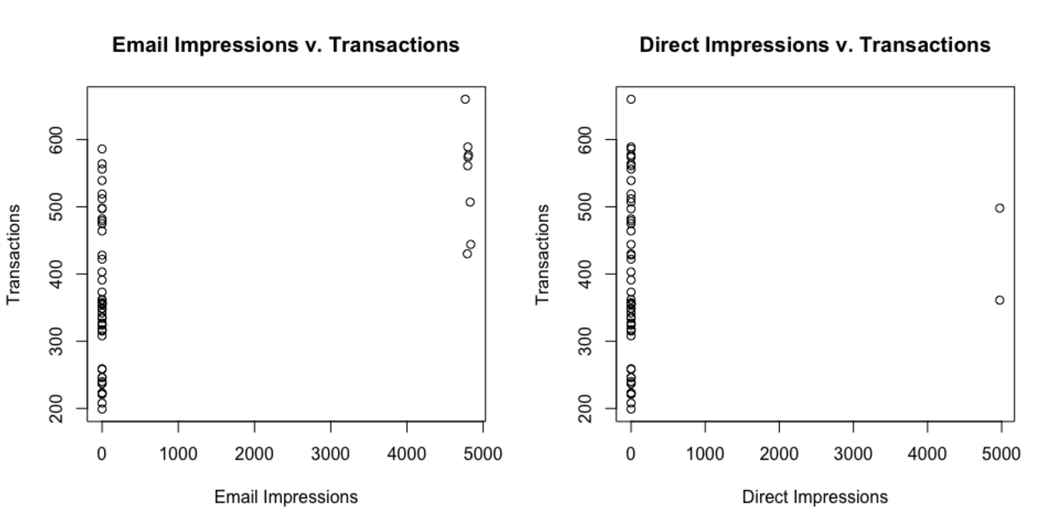
Now that we have the data aggregated, we should do some exploration before modeling.

```
> plot(x=mdata$direct, y=mdata$trans, xlab="Direct Impressions",
+      ylab="Transactions", main="Direct Impressions v. Transactions")
> plot(x=mdata$email, y=mdata$trans, xlab="Email Impressions",
+      ylab="Transactions", main="Email Impressions v. Transactions")
> plot(x=mdata$display, y=mdata$trans, xlab="Display Impressions",
+      ylab="Transactions", main="Display Impressions v. Transactions")
> plot(x=mdata$social, y=mdata$trans, xlab="Social Impressions",
+      ylab="Transactions", main="Social Impressions v. Transactions")
```

Daily display and social impressions versus transactions



Daily email and direct mpressions versus transactions



Running a simple media mix model

```
> m1 <- lm(trans~direct+social+email+display+social+email.holdout, data=mdata)
> summary(m1)
...
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   438.501429 940.622551   0.466 0.643070
direct          0.014529   0.013853   1.049 0.299206
social          0.012663   0.003379   3.747 0.000457 ***
email          -0.161643   0.321735  -0.502 0.617541
display        -0.035299   0.252371  -0.140 0.889313
email.holdout   0.805475   1.290198   0.624 0.535211
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 92.53 on 51 degrees of freedom
Multiple R-squared:  0.4644,    Adjusted R-squared:  0.4119
F-statistic: 8.845 on 5 and 51 DF,  p-value: 0.00000427
```

Things to notice in the model output

```
> m1 <- lm(trans~direct+social+email+display+social+email.holdout,
           data=mdata)
> summary(m1)

Call:
lm(formula = trans ~ direct + social + email + display + social +
    email.holdout, data = mdata)

Residuals:
    Min      1Q  Median      3Q     Max
-106.35  -67.79  -41.56   54.42  211.04

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   438.501429 940.622551   0.466 0.643070
direct          0.014529   0.013853   1.049 0.299206
social          0.012663   0.003379   3.747 0.000457 ***
email          -0.161643   0.321735  -0.502 0.617541
display        -0.035299   0.252371  -0.140 0.889313
email.holdout   0.805475   1.290198   0.624 0.535211
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 92.53 on 51 degrees of freedom
Multiple R-squared:  0.4644,    Adjusted R-squared:  0.4119
F-statistic: 8.845 on 5 and 51 DF,  p-value: 0.00000427
```

The only statistically significant effect is for social impressions, and we get 0.012663 additional transactions for each social impression.

Things to notice in the model output

```
> m1 <- lm(trans~direct+social+email+display+social+email.holdout,
           data=mdata)
> summary(m1)

Call:
lm(formula = trans ~ direct + social + email + display + social +
    email.holdout, data = mdata)

Residuals:
    Min      1Q  Median      3Q     Max
-106.35  -67.79  -41.56   54.42  211.04

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   438.501429 940.622551   0.466 0.643070
direct          0.014529   0.013853   1.049 0.299206
social          0.012663   0.003379   3.747 0.000457 ***
email          -0.161643   0.321735  -0.502 0.617541
display        -0.035299   0.252371  -0.140 0.889313
email.holdout   0.805475   1.290198   0.624 0.535211
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 92.53 on 51 degrees of freedom
Multiple R-squared:  0.4644,    Adjusted R-squared:  0.4119
F-statistic: 8.845 on 5 and 51 DF,  p-value: 0.00000427
```

The only statistically significant effect is for social impressions, and we get 0.012663 additional transactions for each social impression.

The standard error for display is very large, which means we don't have a precise estimate of the effect of display. This happened because daily display impressions are pretty much the same every day. The data is not informative!

Things to notice in the model output

```
> m1 <- lm(trans~direct+social+email+display+social+email.holdout,
           data=mdata)
> summary(m1)

Call:
lm(formula = trans ~ direct + social + email + display + social +
    email.holdout, data = mdata)

Residuals:
    Min      1Q  Median      3Q     Max
-106.35  -67.79  -41.56   54.42  211.04

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   438.501429 940.622551   0.466 0.643070
direct          0.014529   0.013853   1.049 0.299206
social          0.012663   0.003379   3.747 0.000457 ***
email          -0.161643   0.321735  -0.502 0.617541
display        -0.035299   0.252371  -0.140 0.889313
email.holdout   0.805475   1.290198   0.624 0.535211
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 92.53 on 51 degrees of freedom
Multiple R-squared:  0.4644,    Adjusted R-squared:  0.4119
F-statistic: 8.845 on 5 and 51 DF,  p-value: 0.00000427
```

The only statistically significant effect is for social impressions, and we get 0.012663 additional transactions for each social impression.

The standard error for display is very large, which means we don't have a precise estimate of the effect of display. This happened because daily display impressions are pretty much the same every day. The data is not informative!

The estimated effect of email and display is negative (but not significant).

Adding other variables

In addition to advertising impressions, we might want to include other predictors of transactions in our model, such as the day of the week.

```
> mdata$dayofweek <- weekdays(as.Date(rownames(mdata)))
> m2 <- lm(trans~email+direct+display+social+dayofweek, data=mdata)
```

Our second model

```
> summary(m2)

Call:
lm(formula = trans ~ email + direct + display + social + dayofweek,
    data = mdata)

Residuals:
   Min      1Q  Median      3Q     Max
-87.09 -51.78 -21.70   25.46  221.46

Coefficients:
                      Estimate  Std. Error t value Pr(>|t|)
(Intercept)         348.192012  891.949059   0.390 0.698063
email                -0.960694    1.463482  -0.656 0.514809
direct                0.029537    0.014165   2.085 0.042632 *
display              -0.031124    0.239069  -0.130 0.896986
social                0.012754    0.003077   4.144 0.000145 ***
dayofweekMonday      71.405198   45.616341   1.565 0.124357
dayofweekSaturday    64.445464   45.601326   1.413 0.164318
dayofweekSunday      55.293919   44.598357   1.240 0.221330
dayofweekThursday    67.663736   45.341037   1.492 0.142440
dayofweekTuesday   4877.229334 7029.997193   0.694 0.491313
dayofweekWednesday  171.086874   45.336050   3.774 0.000459 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 83.94 on 46 degrees of freedom
Multiple R-squared:  0.6024,    Adjusted R-squared:  0.516
F-statistic:  6.97 on 10 and 46 DF,  p-value: 0.000001582
```

When we add other variables, it can change the coefficients of the model.

Our second model

```
> summary(m2)

Call:
lm(formula = trans ~ email + direct + display + social + dayofweek,
    data = mdata)

Residuals:
   Min     1Q  Median     3Q    Max
-87.09 -51.78 -21.70  25.46 221.46

Coefficients:
                      Estimate   Std. Error t value Pr(>|t|)
(Intercept)         348.192012  891.949059   0.390 0.698063
email                -0.960694    1.463482  -0.656 0.514809
direct                0.029537    0.014165   2.085 0.042632 *
display              -0.031124    0.239069  -0.130 0.896986
social                0.012754    0.003077   4.144 0.000145 ***
dayofweekMonday      71.405198   45.616341   1.565 0.124357
dayofweekSaturday    64.445464   45.601326   1.413 0.164318
dayofweekSunday      55.293919   44.598357   1.240 0.221330
dayofweekThursday    67.663736   45.341037   1.492 0.142440
dayofweekTuesday   4877.229334 7029.997193   0.694 0.491313
dayofweekWednesday  171.086874   45.336050   3.774 0.000459 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 83.94 on 46 degrees of freedom
Multiple R-squared:  0.6024,    Adjusted R-squared:  0.516
F-statistic:  6.97 on 10 and 46 DF,  p-value: 0.000001582
```

When we add other variables, it can change the coefficients of the model.

With our new model we find a significant association between direct impressions and transactions (29.5 additional transactions / 1000 direct impressions) in addition to social (12.8 additional transactions / 1000 impressions).

Summary: marketing mix models

A marketing mix model is a regression relating advertising spending or total impressions to some response like sales.

To build a marketing mix model, you assemble data on advertising spending or impressions by day/week/month and conversions or sales by day/week/month and then fit a model using any regression tool.

The coefficients of this model tell us the correlations between daily/weekly/monthly levels of advertising and sales.

What can go wrong with marketing mix modeling?

There isn't enough variation in the advertising over time or two different types of advertising tend to happen at the same time.

- Model estimates will show large standard errors on the coefficients. You'll be stuck, but at least you'll know it.

What can go wrong with marketing mix modeling?

There isn't enough variation in the advertising over time or two different types of advertising tend to happen at the same time.

- Model estimates will show large standard errors on the coefficients. You'll be stuck, but at least you'll know it.

Advertising happens during periods of peak demand, reversing the causality.

- Model estimates may appear reasonable but will overestimate advertising response. You'll be wrong, but you won't know it.

What can go wrong with marketing mix modeling?

There isn't enough variation in the advertising over time or two different types of advertising tend to happen at the same time.

- Model estimates will show large standard errors on the coefficients. You'll be stuck, but at least you'll know it.

Advertising happens during periods of peak demand, reversing the causality.

- Model estimates may appear reasonable but will overestimate advertising response. You'll be wrong, but you won't know it.

The analyst can twittle with the model until she gets it to say what she likes or what you want to hear.

Model-based attribution

Model-based attribution

What is model-based attribution?

Model-based attribution

Model-based attribution is similar to marketing mix modeling, but we do the analysis at the user-level, relating a user's transactions or conversions to her prior advertising exposures. A simple attribution model is:

$$conversion_{it} = \beta_0 + \beta_1 \text{display}_{it} + \beta_2 \text{social}_{it} + \beta_3 \text{email}_{it} + \beta_4 \text{direct}_{it} + \epsilon_{it}$$

The key difference from a marketing mix model is that now all the variables are indexed by i in addition to t.

Model-based attribution

Model-based attribution

Model-based attribution

To run this regression, we need to create a data frame where each row is a user-day and summarizes the user's impressions and transactions on that day. We have to transform the raw data to this:

```
> head(adata)
  id       date direct display email email.holdout social trans past.purchase has.email has.direct
1  1 2017-01-01      0       0     0             0      1     0             0         0          1
2  1 2017-01-02      0       0     0             0      1     0             0         0          1
3  1 2017-01-03      0       0     0             0      0     0             0         0          1
4  1 2017-01-04      0       0     0             0      0     0             0         0          1
5  1 2017-01-05      0       0     0             0      1     0             0         0          1
6  1 2017-01-06      1       0     0             0      2     0             0         0          1
```

My R code for doing this is a bit messy and takes several steps. I'm sure you could do better with tidyverse.

Data transform step 1: Summary of impressions by user and day

First, we create a data frame that has a daily summary of each user's transactions.

```
> adata1 <- as.data.frame(xtabs(~ id + date + channel, data=impress), stringsAsFactors=FALSE)
> adata1$id <- as.integer(adata1$id)
> adata1$date <- as.Date(adata1$date)
> adata1$channel <- as.factor(adata1$channel)
> dimnames(adata1)[[2]][4] <- "impr"
> head(adata1)
  id       date channel impr
1  1 2016-12-31  direct    0
2  2 2016-12-31  direct    0
3  3 2016-12-31  direct    0
4  5 2016-12-31  direct    0
5  6 2016-12-31  direct    0
6  7 2016-12-31  direct    0
```

Data transform step 2: Adding in users with zero impressions

Because some users don't appear in the impressions file, we have to add in rows (with zeros) for those users.

```
> pop <- unique(cust$id)
> no.impress.ids <- pop[!(pop %in% unique(impress$id))]
> dates <- sort(unique(impress$date))
> channels <- unique(impress$channel)
> no.impress.obs <- data.frame(id=rep(no.impress.ids, each=length(dates)*length(channels)),
+                              date=rep(rep(dates, each=length(channels)), length(no.impress.ids)),
+                              channel=rep(channels, length(no.impress.ids)*length(dates)),
+                              impr=rep(0, length(dates)*length(no.impress.ids)*length(channels)),
+                              stringsAsFactors=FALSE)
> no.impress.obs$channel <- as.factor(no.impress.obs$channel)
> adatal <- rbind(adatal, no.impress.obs)
> summary(adatal)
       id              date                         channel             impr
 Min.   :    1   Min.   :2016-12-31   direct        :590000   Min.   : 0.0000
 1st Qu.: 2501   1st Qu.:2017-01-14   display       :590000   1st Qu.: 0.0000
 Median : 5000   Median :2017-01-29   email         :590000   Median : 0.0000
 Mean   : 5000   Mean   :2017-01-29   email.holdout:590000   Mean   : 0.1699
 3rd Qu.: 7500   3rd Qu.:2017-02-13   social        :590000   3rd Qu.: 0.0000
 Max.   :10000   Max.   :2017-02-27                           Max.   :16.0000
```

Data transform step 3: Switch to wide format

Next, we unstack the impressions column by the channel.

```
> adata <- reshape(adatal, direction="wide", v.names="impr", idvar=c("id", "date"),
+                  timevar="channel", new.row.names=NULL)
> sum(adata$impr.direct) == length(impress$channel[impress$channel=="direct"]) #quick check
[1] TRUE
> nrow(adata)
[1] 590000
> head(adata)
  id       date impr.direct impr.display impr.email impr.email.holdout impr.social
1  1 2016-12-31           0            0          0                  0           0
2  2 2016-12-31           0            0          0                  0           0
3  3 2016-12-31           0            0          0                  0           0
4  5 2016-12-31           0            1          0                  0           0
5  6 2016-12-31           0            0          0                  0           0
6  7 2016-12-31           0            0          0                  0           0
```

Data transformation step 4: Add the daily transactions for each user

Finally, we summarize the transactions by customer-day and merge with impressions.

```
> atrans <- as.data.frame(xtabs(~ id + date, data=trans), stringsAsFactors=FALSE)
> atrans$id <- as.integer(atrans$id)
> atrans$date <- as.Date(atrans$date)
> dimnames(atrans)[[2]][3] <- "trans"
> adata <- merge(adata, atrans, by=c("id", "date"), all=TRUE)
> adata$trans[is.na(adata$trans)] <- 0  # fill in zeros for transactions
> head(adata)
  id       date impr.direct impr.display impr.email impr.email.holdout impr.social trans
1  1 2016-12-31           0            0          0                  0           0     0
2  1 2017-01-01           0            0          0                  0           1     0
3  1 2017-01-02           0            0          0                  0           1     0
4  1 2017-01-03           0            0          0                  0           0     0
5  1 2017-01-04           0            0          0                  0           0     0
6  1 2017-01-05           0            0          0                  0           1     0
```

Data transformation step 5: Final tidy up of attribution modeling data

```
> # Remove first and last days (which are incomplete)
> adata <- adata[adata$date!="2016-12-31" & adata$date != "2017-02-28" & adata$date != "2017-02-27",]
# Add customer info from cust table
> adata <- merge(adata, cust, by=c("id"))
# Tidy up column names
> dimnames(adata)[[2]][3:11] <- c("direct", "display", "email", "email.holdout", "social",
              "trans", "past.purchase", "has.email", "has.direct")
> rm(adatal, atrans)
```

Summary of attribution modeling data

```
> summary(adata)
       id              date                direct            display             email        
 Min.   :    1   Min.   :2017-01-01   Min.   :0.00000   Min.   : 0.0000   Min.   :0.00000  
 1st Qu.: 2501   1st Qu.:2017-01-15   1st Qu.:0.00000   1st Qu.: 0.0000   1st Qu.:0.00000  
 Median : 5000   Median :2017-01-29   Median :0.00000   Median : 0.0000   Median :0.00000  
 Mean   : 5000   Mean   :2017-01-29   Mean   :0.01745   Mean   : 0.3731   Mean   :0.06741  
 3rd Qu.: 7500   3rd Qu.:2017-02-12   3rd Qu.:0.00000   3rd Qu.: 0.0000   3rd Qu.:0.00000  
 Max.   :10000   Max.   :2017-02-26   Max.   :1.00000   Max.   :13.0000   Max.   :1.00000  
 email.holdout         social            trans          past.purchase      has.email     
 Min.   :0.00000   Min.   : 0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:0.00000   1st Qu.: 0.0000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :0.00000   Median : 0.0000   Median :0.00000   Median :1.0000   Median :1.0000  
 Mean   :0.01681   Mean   : 0.3954   Mean   :0.03858   Mean   :0.5022   Mean   :0.6001  
 3rd Qu.:0.00000   3rd Qu.: 0.0000   3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.0000  
 Max.   :1.00000   Max.   :16.0000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000  
   has.direct    
 Min.   :0.0000  
 1st Qu.:0.0000  
 Median :0.0000  
 Mean   :0.4974  
 3rd Qu.:1.0000  
 Max.   :1.0000  
```

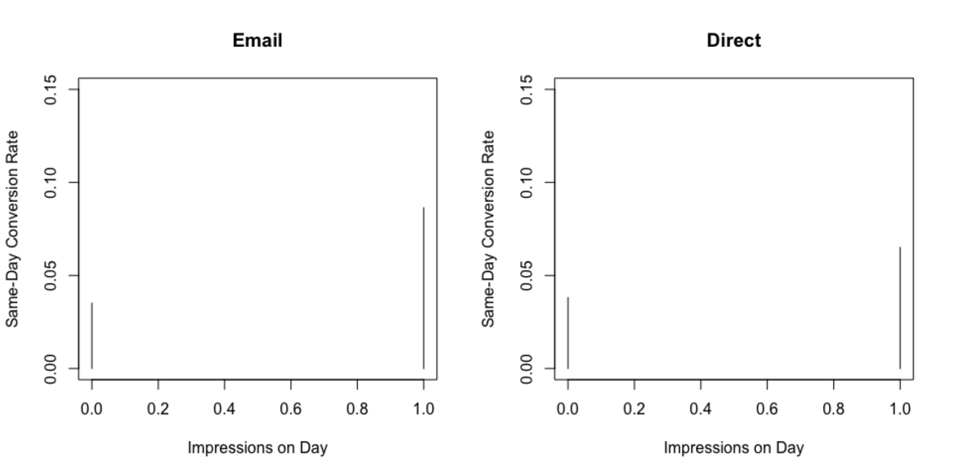
Summary of attribution modeling data on a specific day

```
> summary(adata[adata$date=="2017-01-03",])
       id              date                direct     display            email         email.holdout
 Min.   :    1   Min.   :2017-01-03   Min.   :0   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.: 2501   1st Qu.:2017-01-03   1st Qu.:0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
 Median : 5000   Median :2017-01-03   Median :0   Median :0.0000   Median :0.0000   Median :0.0000
 Mean   : 5000   Mean   :2017-01-03   Mean   :0   Mean   :0.3656   Mean   :0.4798   Mean   :0.1203
 3rd Qu.: 7500   3rd Qu.:2017-01-03   3rd Qu.:0   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
 Max.   :10000   Max.   :2017-01-03   Max.   :0   Max.   :9.0000   Max.   :1.0000   Max.   :1.0000
     social            trans         past.purchase      has.email        has.direct
 Min.   : 0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
 Median : 0.0000   Median :0.0000   Median :1.0000   Median :1.0000   Median :0.0000
 Mean   : 0.7505   Mean   :0.0589   Mean   :0.5022   Mean   :0.6001   Mean   :0.4974
 3rd Qu.: 1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   :12.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

Visualizing the relationship between impressions and transactions

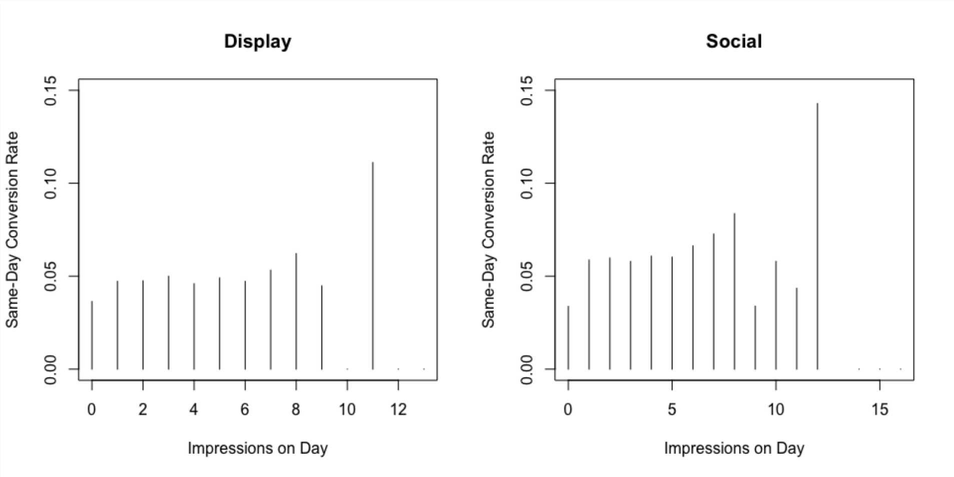
```
> plot(aggregate(trans~direct, data=adata, FUN=mean), type="h", ylim=c(0,0.15),
+      xlab="Impressions on Day", ylab="Same-Day Conversion Rate", main="Direct")
> plot(aggregate(trans~email, data=adata, FUN=mean), type="h", ylim=c(0,0.15),
+      xlab="Impressions on Day", ylab="Same-Day Conversion Rate", main="Email")
> plot(aggregate(trans~display, data=adata, FUN=mean), type="h", ylim=c(0,0.15),
+      xlab="Impressions on Day", ylab="Same-Day Conversion Rate", main="Display")
> plot(aggregate(trans~social, data=adata, FUN=mean), type="h", ylim=c(0,0.15),
+      xlab="Impressions on Day", ylab="Same-Day Conversion Rate", main="Social")
```

Direct and email daily impressions versus transactions



Looking at user-level data, it is easy to see that users convert more on days they get emails or direct mail.

Display and social daily impressions versus transactions



A simple attribution model

```
> m1 <- lm(trans ~ direct + display + email + social, data=adata)
> summary(m1)
...
Coefficients:
             Estimate Std. Error t value            Pr(>|t|)    
(Intercept) 0.0296060  0.0003024   97.91 <0.0000000000000002 ***
direct      0.0306548  0.0019423   15.78 <0.0000000000000002 ***
display     0.0041618  0.0002759   15.09 <0.0000000000000002 ***
email       0.0520472  0.0010144   51.31 <0.0000000000000002 ***
social      0.0085516  0.0002542   33.64 <0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1919 on 569995 degrees of freedom
Multiple R-squared:  0.007249,  Adjusted R-squared:  0.007242
F-statistic:  1041 on 4 and 569995 DF,  p-value: < 0.00000000000000022
```

Key things to notice about our attribution model

```
> m1 <- lm(trans ~ direct + display + email + social, data=adata)
> summary(m1)
...
Coefficients:
             Estimate Std. Error t value            Pr(>|t|)
(Intercept) 0.0296060  0.0003024   97.91 <0.0000000000000002 ***
direct      0.0306548  0.0019423   15.78 <0.0000000000000002 ***
display     0.0041618  0.0002759   15.09 <0.0000000000000002 ***
email       0.0520472  0.0010144   51.31 <0.0000000000000002 ***
social      0.0085516  0.0002542   33.64 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1919 on 569995 degrees of freedom
Multiple R-squared:  0.007249,  Adjusted R-squared:  0.007242
F-statistic:  1041 on 4 and 569995 DF,
p-value: < 0.00000000000000022
```

We have gotten reasonable estimates of advertising response even for this simple model that only looks at same-day response. For example, 1 email impressions is associated with 0.052 additional transactions on the same day.

Key things to notice about our attribution model

```
> m1 <- lm(trans ~ direct + display + email + social, data=adata)
> summary(m1)
...
Coefficients:
             Estimate Std. Error t value            Pr(>|t|)
(Intercept) 0.0296060  0.0003024   97.91 <0.0000000000000002 ***
direct      0.0306548  0.0019423   15.78 <0.0000000000000002 ***
display     0.0041618  0.0002759   15.09 <0.0000000000000002 ***
email       0.0520472  0.0010144   51.31 <0.0000000000000002 ***
social      0.0085516  0.0002542   33.64 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1919 on 569995 degrees of freedom
Multiple R-squared:  0.007249,  Adjusted R-squared:  0.007242
F-statistic:  1041 on 4 and 569995 DF,
p-value: < 0.00000000000000022
```

We have gotten reasonable estimates of advertising response even for this simple model that only looks at same-day response. For example, 1 email impressions is associated with 0.052 additional transactions on the same day.

The probability of transacting on a day where the user is exposed to no advertising is 3%. It would be bad to assume that all sales are due to advertising.

A model that includes user-characteristics

Because we have a user-level model, we can bring in user characteristics.

```
> m2 <- lm(trans ~ direct + display + email + social + past.purchase, data=adata)
> summary(m2)
Coefficients:
               Estimate Std. Error t value             Pr(>|t|)
(Intercept)   0.0142352  0.0003889   36.60 <0.0000000000000002 ***
direct        0.0203596  0.0019427   10.48 <0.0000000000000002 ***
display       0.0041846  0.0002749   15.22 <0.0000000000000002 ***
email         0.0433566  0.0010205   42.49 <0.0000000000000002 ***
social        0.0086049  0.0002533   33.97 <0.0000000000000002 ***
past.purchase 0.0320725  0.0005130   62.52 <0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1912 on 569994 degrees of freedom
Multiple R-squared:  0.01401,   Adjusted R-squared:  0.014
F-statistic:  1620 on 5 and 569994 DF,  p-value: < 0.00000000000000022
```

A model that includes user-characteristics

```
> m2 <- lm(trans ~ direct + display + email + social +
>                  past.purchase, data=adata)
> summary(m2)
Coefficients:
               Estimate Std. Error t value          Pr(>|t|)
(Intercept)   0.0142352  0.0003889   36.60 <0.0000000000000002 ***
direct        0.0203596  0.0019427   10.48 <0.0000000000000002 ***
display       0.0041846  0.0002749   15.22 <0.0000000000000002 ***
email         0.0433566  0.0010205   42.49 <0.0000000000000002 ***
social        0.0086049  0.0002533   33.97 <0.0000000000000002 ***
past.purchase 0.0320725  0.0005130   62.52 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1912 on 569994 degrees of freedom
Multiple R-squared:  0.01401,   Adjusted R-squared:  0.014
F-statistic:  1620 on 5 and 569994 DF,
p-value: < 0.00000000000000022
```

The model suggests that customers who have not made a purchase before are much less likely to make a purchase (1.4% versus 4.6% baseline rate without advertising).

A model that includes user-characteristics

```
> m2 <- lm(trans ~ direct + display + email + social +
>                  past.purchase, data=adata)
> summary(m2)
Coefficients:
               Estimate Std. Error t value            Pr(>|t|)
(Intercept)   0.0142352  0.0003889   36.60 <0.0000000000000002 ***
direct        0.0203596  0.0019427   10.48 <0.0000000000000002 ***
display       0.0041846  0.0002749   15.22 <0.0000000000000002 ***
email         0.0433566  0.0010205   42.49 <0.0000000000000002 ***
social        0.0086049  0.0002533   33.97 <0.0000000000000002 ***
past.purchase 0.0320725  0.0005130   62.52 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1912 on 569994 degrees of freedom
Multiple R-squared:  0.01401,   Adjusted R-squared:  0.014
F-statistic:  1620 on 5 and 569994 DF,
p-value: < 0.00000000000000022
```

The model suggests that customers who have not made a purchase before are much less likely to make a purchase (1.4% versus 4.6% baseline rate without advertising).

The estimates of advertising effects are smaller than in our first model. This is because people who have made a purchase are more likely to be exposed to advertising. When we control for past purchase, we can see that the effects of advertising are smaller than we might have thought based on the first model.

A model that includes user-characteristics

```
> m2 <- lm(trans ~ direct + display + email + social +
>                  past.purchase, data=adata)
> summary(m2)
Coefficients:
               Estimate Std. Error t value            Pr(>|t|)
(Intercept)   0.0142352  0.0003889   36.60 <0.0000000000000002 ***
direct        0.0203596  0.0019427   10.48 <0.0000000000000002 ***
display       0.0041846  0.0002749   15.22 <0.0000000000000002 ***
email         0.0433566  0.0010205   42.49 <0.0000000000000002 ***
social        0.0086049  0.0002533   33.97 <0.0000000000000002 ***
past.purchase 0.0320725  0.0005130   62.52 <0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1912 on 569994 degrees of freedom
Multiple R-squared:  0.01401,   Adjusted R-squared:  0.014
F-statistic:  1620 on 5 and 569994 DF,
p-value: < 0.00000000000000022
```

The model suggests that customers who have not made a purchase before are much less likely to make a purchase (1.4% versus 4.6% baseline rate without advertising).

The estimates of advertising effects are smaller than in our first model. This is because people who have made a purchase are more likely to be exposed to advertising. When we control for past purchase, we can see that the effects of advertising are smaller than we might have thought based on the first model.

The controls you choose to add to the model can be critical.

Logistic regression

Those of you who learned regression from a careful teacher may have noticed that I was violating a key assumption of the linear regression model.

Logistic regression

Those of you who learned regression from a careful teacher may have noticed that I was violating a key assumption of the linear regression model.

Our outcome variable is binary and so it would be better to use a logistic regression for binary outcomes.

Logistic regression for attribution

```
> m3 <- glm(trans ~ direct + display + email + social + past.purchase,
+           family=binomial(), data=adata)
> summary(m3)
...
Coefficients:
               Estimate Std. Error  z value               Pr(>|z|)
(Intercept)   -4.022893   0.014013 -287.078 <0.0000000000000002 ***
direct         0.412554   0.041688    9.896 <0.0000000000000002 ***
display        0.100630   0.006581   15.291 <0.0000000000000002 ***
email          0.762355   0.020010   38.099 <0.0000000000000002 ***
social         0.180079   0.005356   33.621 <0.0000000000000002 ***
past.purchase  0.957207   0.015672   61.079 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 186297  on 569999  degrees of freedom
Residual deviance: 178869  on 569994  degrees of freedom
AIC: 178881
```

Logistic regression for attribution

```
> m3 <- glm(trans ~ direct + display + email + social + past.purchase,
+           family=binomial(), data=adata)
> summary(m3)
...
Coefficients:
               Estimate Std. Error  z value            Pr(>|z|)
(Intercept)   -4.022893   0.014013 -287.078 <0.0000000000000002 ***
direct         0.412554   0.041688    9.896 <0.0000000000000002 ***
display        0.100630   0.006581   15.291 <0.0000000000000002 ***
email          0.762355   0.020010   38.099 <0.0000000000000002 ***
social         0.180079   0.005356   33.621 <0.0000000000000002 ***
past.purchase  0.957207   0.015672   61.079 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 186297  on 569999  degrees of freedom
Residual deviance: 178869  on 569994  degrees of freedom
AIC: 178881

Number of Fisher Scoring iterations: 6
```

While this model is better, the logistic regression coefficients can be difficult to interpret.

Logistic regression for attribution

```
> m3 <- glm(trans ~ direct + display + email + social + past.purchase,
+           family=binomial(), data=adata)
> summary(m3)
...
Coefficients:
               Estimate Std. Error  z value             Pr(>|z|)
(Intercept)   -4.022893   0.014013 -287.078 <0.0000000000000002 ***
direct         0.412554   0.041688    9.896 <0.0000000000000002 ***
display        0.100630   0.006581   15.291 <0.0000000000000002 ***
email          0.762355   0.020010   38.099 <0.0000000000000002 ***
social         0.180079   0.005356   33.621 <0.0000000000000002 ***
past.purchase  0.957207   0.015672   61.079 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 186297  on 569999  degrees of freedom
Residual deviance: 178869  on 569994  degrees of freedom
AIC: 178881

Number of Fisher Scoring iterations: 6
```

While this model is better, the logistic regression coefficients can be difficult to interpret.

We are still seeing that direct and email have a bigger effect than a single display or social impression.