ARTEA: DESIGNING TARGETING STRATEGIES

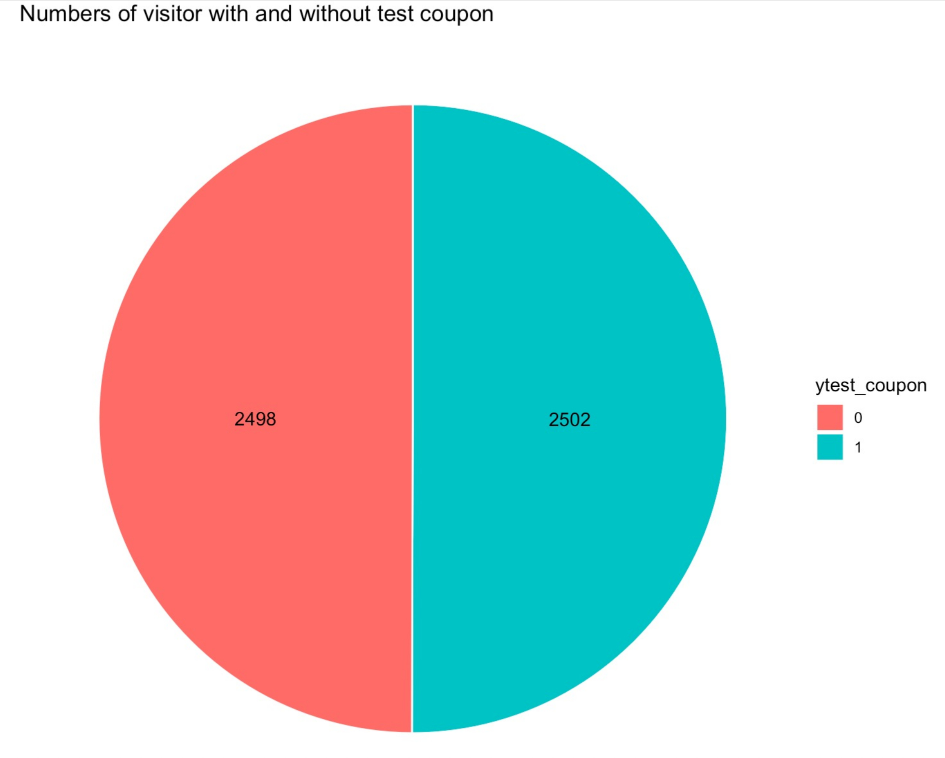
Tobias Jansen | Sinah-Nikola Kaefferlein | Leon Deng | Anne Huesges

Sourcecode and data:
https://github.com/tjbnde/abtest-artea

1) Proof of random manipulation

DIVIDING TEST-GROUP INTO red = NO_TEST_COUPON AND blue = COUPON

```
data$ytest_coupon=factor(data$test_coupon)
data$yshopping_cart=factor(data$shopping_cart)
no_test_coupon <- data %>%
  group_by(ytest_coupon) %>%
  summarise(n = n())

no_test_coupon
no_test_coupon_pos <- no_test_coupon %>%
  arrange(desc(ytest_coupon)) %>%
  mutate(lab.ypos = cumsum(n) - 0.5*n)
no_test_coupon_pos

ggplot(no_test_coupon_pos, aes(x="", y = n, fill = ytest_coupon)) +
  geom_bar(width = 1, stat= "identity", color= "white") +
  coord_polar("y", start = 0) +
  geom_text(aes(y = lab.ypos, label=n), color= "black") +
  theme_void() +
  ggtitle("Numbers of visitor with and without test coupon")
```



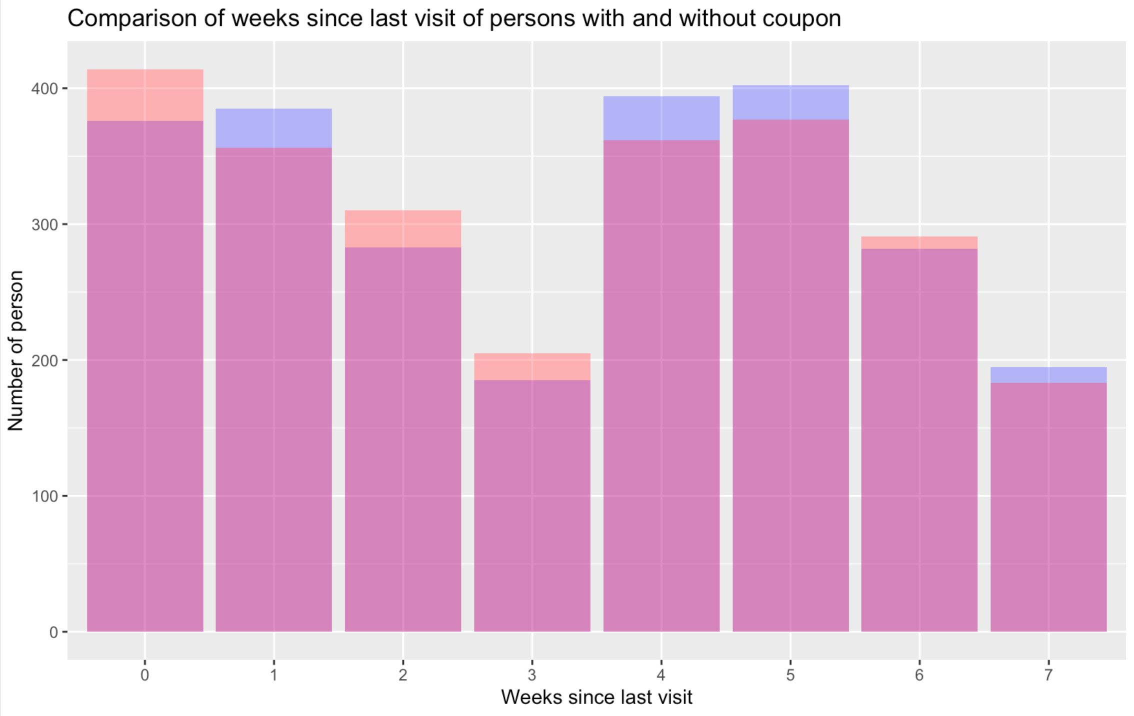
1) Proof of random manipulation

DIVIDING TEST-GROUP INTO red = NO_COUPON AND blue = COUPON (AND purple = OVERLAP)

COMPARING DIVIDED TEST GROUPS WITH SEVERAL VARIABLES

```r
data %>%
  group_by(test_coupon) %>%
  summarize(
    number = n(), mean(weeks_since_visit),
    sd(weeks_since_visit), std.error(weeks_since_visit)
  )

describeBy(data$weeks_since_visit, data$test_coupon)

ggplot(data, aes(x = weeks_since_visit, fill = test_coupon)) +
  geom_bar(data = data_coupon, fill = "red", alpha = 0.3) +
  geom_bar(data = data_no_coupon, fill = "blue", alpha = 0.3) +
  labs(title="Comparison of weeks since last visit of persons with and without coupon",
      x="Weeks since last visit", y="Number of person")
```

```
  test_coupon number `mean(weeks_since_visit)` `sd(weeks_since_visit)` `std.error(weeks_since_visit)`
        <int>  <int>                     <dbl>                    <dbl>                          <dbl>
1           0   2498                      3.18                     2.26                         0.0453
2           1   2502                      3.26                     2.25                         0.0451
>
> describeBy(data$weeks_since_visit, data$test_coupon)

 Descriptive statistics by group
group: 0
   vars    n mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 2498 3.18 2.26      3    3.14 2.97   0   7     7 0.05    -1.28 0.05
------------------------------------------------------------------------------------
group: 1
   vars    n mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 2502 3.26 2.25      4    3.22 2.97   0   7     7 0.01    -1.28 0.05
```

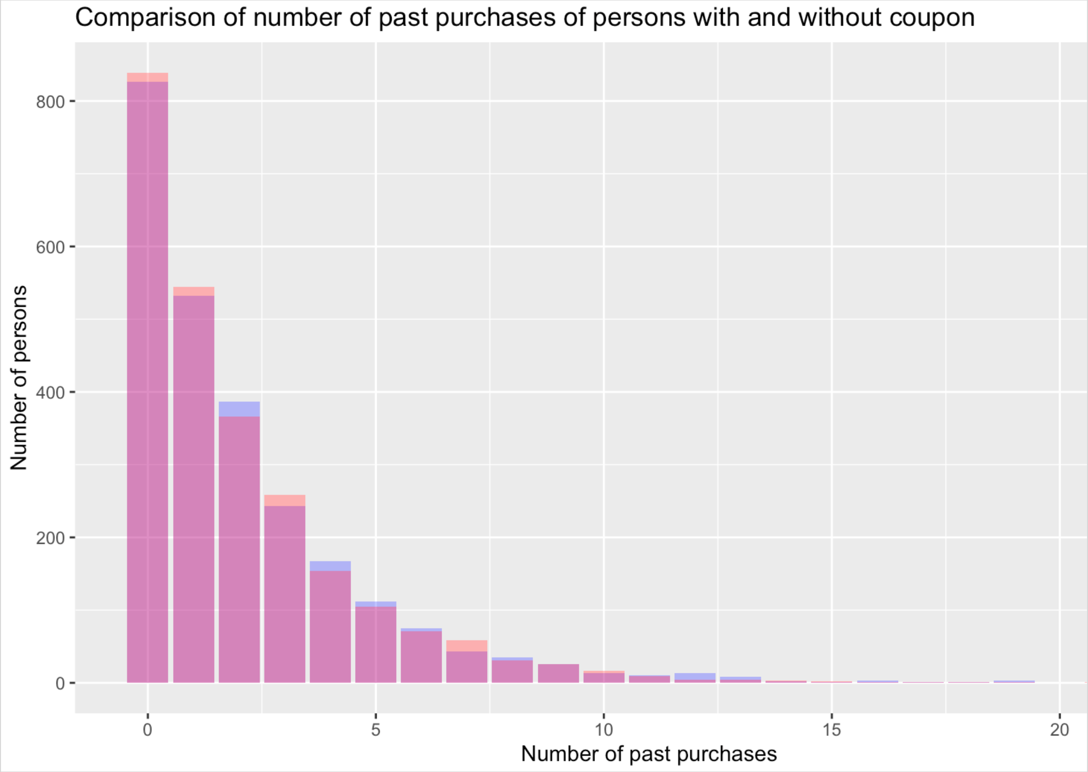


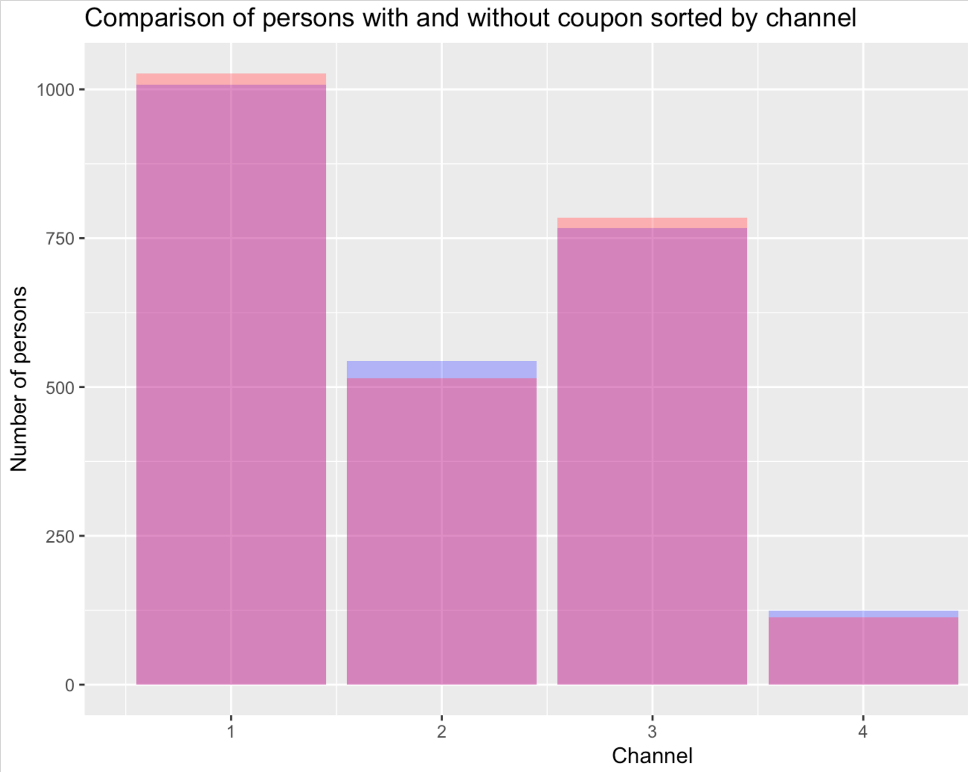
Almost no deviation → random manipulation

1) Proof of random manipulation

DIVIDING TEST-GROUP INTO red = NO_COUPON AND blue = COUPON (AND purple = OVERLAP)

COMPARING DIVIDED TEST GROUPS WITH SEVERAL VARIABLES





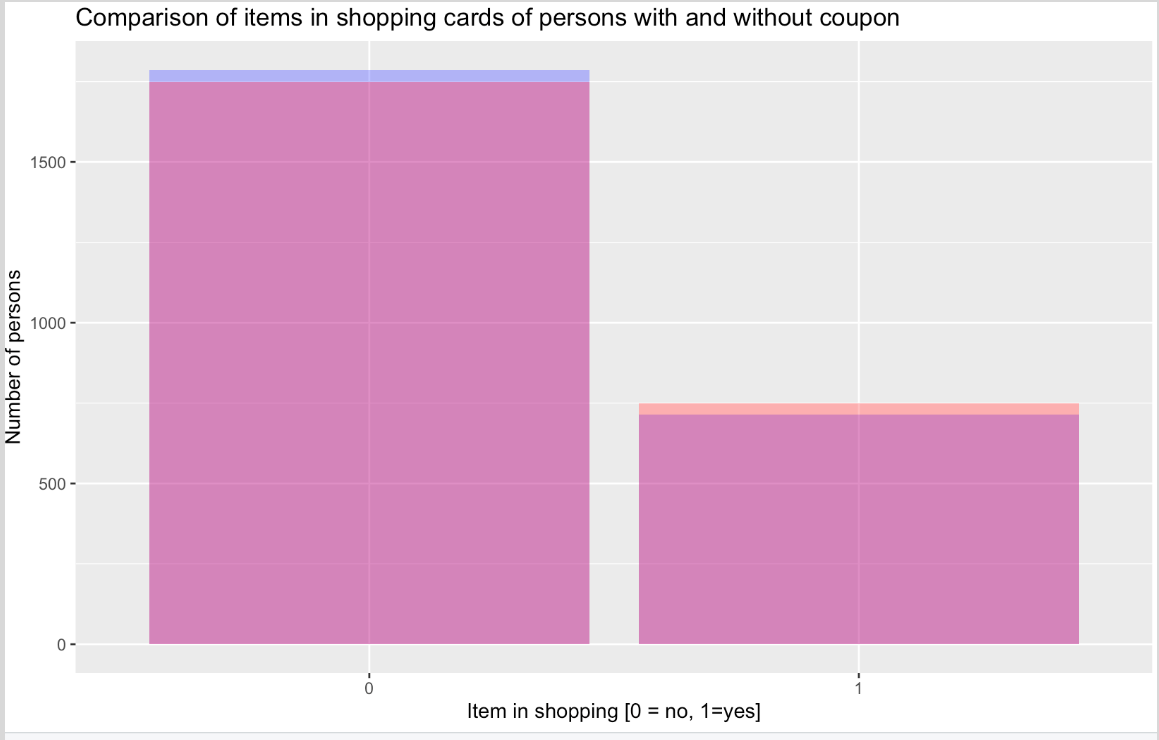
1=GOOGLE
2=FACEBOOK
3=INSTAGRAM
4=REFERRAL
5=OTHER

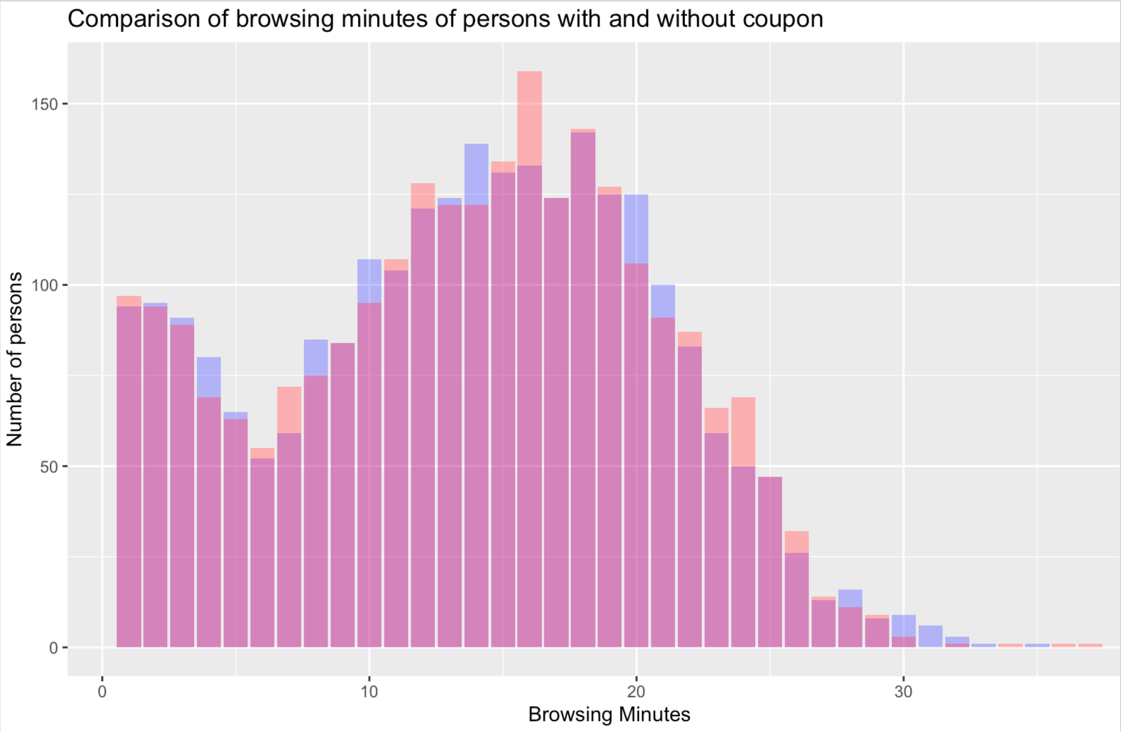
Almost no deviation → random manipulation

1) Proof of random manipulation

DIVIDING TEST-GROUP INTO red = NO_COUPON AND blue = COUPON (AND purple = OVERLAP)

COMPARING DIVIDED TEST GROUPS WITH SEVERAL VARIABLES





Almost no deviation → random manipulation

Q: IS THE MANIPULATION INDEED RANDOM?

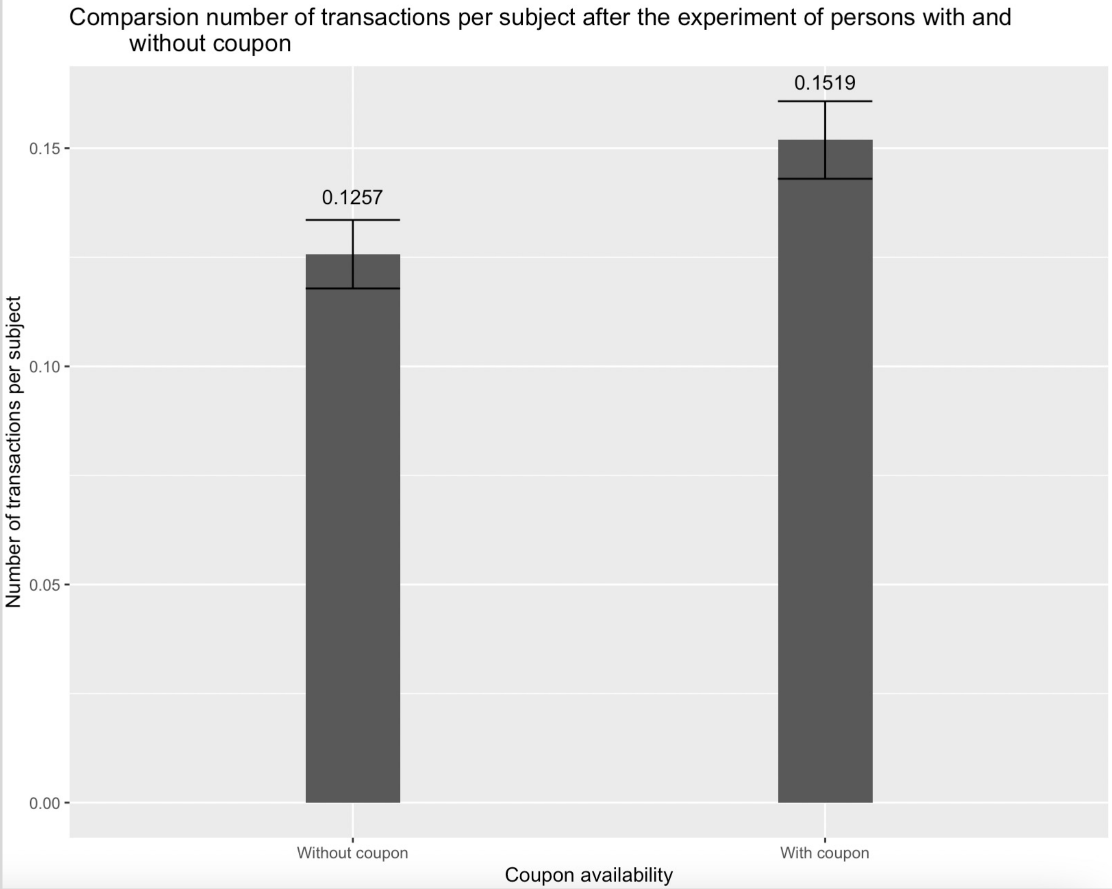
A: YES, IT IS RANDOM!

2) Effect of the Coupons

TRANSACTIONS PER PERSON

```r
effects_of_coupon <- data %>%
    group_by(test_coupon) %>%
    summarize(
        number = n(), revenue_per_subject = sum(revenue_after) / n(),
        transactions_per_subject = sum(trans_after) / n(), error_rev=std.error(revenue_after),
        error_trans=std.error(trans_after)
    )
# Plot transactions_per_subject
plot_data <- data.frame(
    coupon = factor(c("Without coupon", "With coupon"),
        levels = c("Without coupon", "With coupon")
    ),
    transactions_per_subject = effects_of_coupon$transactions_per_subject,
    error_trans = effects_of_coupon$error_trans,
    error_rev = effects_of_coupon$error_rev
)
ggplot(plot_data, aes(x = coupon, y = transactions_per_subject)) +
    geom_bar(stat = "identity", width = 0.2) +
    geom_errorbar(aes(ymin = transactions_per_subject - error_trans,
      ymax = transactions_per_subject + error_trans), width = .2,
      position = position_dodge(.9)) +
    geom_text(aes(label=round(transactions_per_subject, digits=4)), position=position_dodge(width=0.9),
            vjust=-3.5)+
    labs(title="Comparsion number of transactions per subject after the experiment of persons with and
        without coupon", x="Coupon availability", y="Number of transactions per subject")
```

```
> effects_of_coupon
# A tibble: 2 × 4
  test_coupon number revenue_per_subject transactions_per_subject
        <int>  <int>               <dbl>                    <dbl>
1           0   2498                7.78                    0.126
2           1   2502                7.54                    0.152
>
```



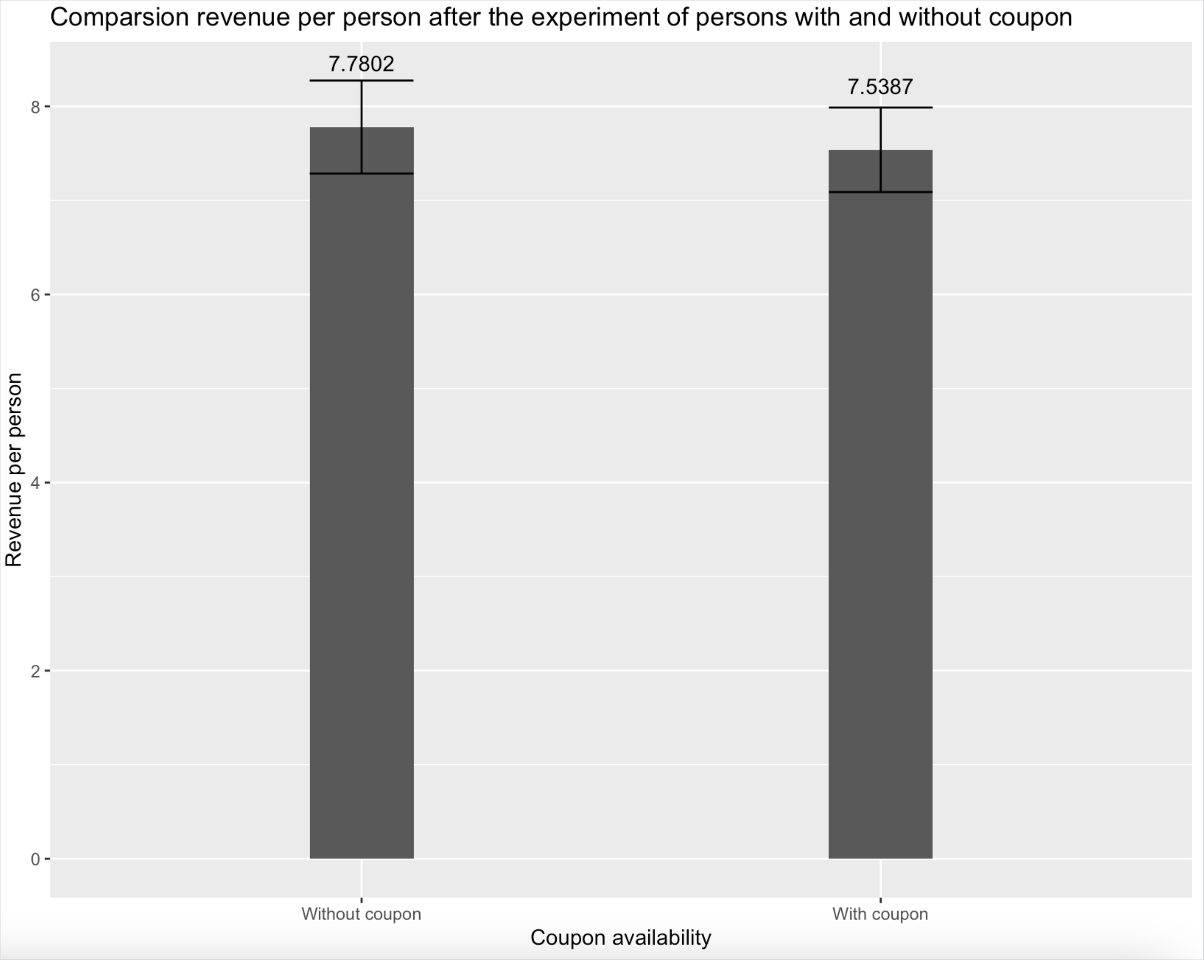
TRANSACTIONS PER CUSTOMER INCREASE

2) Effect of the Coupons

REVENUE PER PERSON

```
plot_data <- data.frame(
  coupon = factor(c("Without coupon", "With coupon"),
                  levels = c("Without coupon", "With coupon")
  ),
  revenue_per_subject = effects_of_coupon$revenue_per_subject
)
ggplot(plot_data, aes(x = coupon, y = revenue_per_subject)) +
  geom_bar(stat = "identity", width = 0.2) +
  geom_text(aes(label=round(revenue_per_subject, digits=4)), position=position_dodge(width=0.9), vjust=-0.25) +
  labs(title="Comparsion revenue per person after the experiment of persons with and without coupon",
       x="Coupon availability", y="Revenue per person")
```

```
> effects_of_coupon
# A tibble: 2 × 4
  test_coupon number revenue_per_subject transactions_per_subject
        <int>  <int>               <dbl>                    <dbl>
1           0   2498                7.78                    0.126
2           1   2502                7.54                    0.152
>
```



REVENUE PER CUSTOMER DECREASED
→ Also due to the 20% Coupon

Q: WERE THE COUPONS EFFECTIVE?

A: NO, THEY WERE INEFFECTIVE!

Q: DID TRANSACTIONS INCREASE?

A: YES,TRANSACTIONS PER CUSTOMER INCREASED!

Q: DID REVENUES INCREASE?

A: NO, REVENUES PER CUSTOMER DECREASED!

→ It is better <u>not</u> to send coupons to every customer

3) Drivers of the Coupons – First Glance

Q: DOES THE CHANNEL OF AQUISITION IMPACT THE EFFICENCY?

```
data %>%
  group_by(test_coupon, channel_acq) %>%
  summarize(n(), mean(revenue_after), mean(trans_after))
```

test_coupon		channel_acq	`n()`	`mean(revenue_after)`	`mean(trans_after)`
<int>		<int>	<int>	<dbl>	<dbl>
0	GOOGLE	1	1027	4.92	0.0789
0	FACEBOOK	2	515	9.81	0.159
0	INSTAGRAM	3	785	9.17	0.146
0	REFERRAL	4	113	11.2	0.186
0	OTHERS	5	58	14.9	0.259
1	GOOGLE	1	1008	2.80	0.0585
1	FACEBOOK	2	544	10.9	0.221
1	INSTAGRAM	3	767	10.3	0.205
1	REFERRAL	4	125	11.5	0.24
1	OTHERS	5	58	12.8	0.241

A: A SELECTED CHANNEL OF AQUISITION HAS A POSITIVE EFFECT ON THE EFFICENCY OF THE COUPON

YES:

- INSTAGRAM & FACEBOOK
- REFERRAL: ONLY BECAUSE OF THE INCREASE OF TRANSACTIONS, BUT NO INCREASE OF THE REVENUE

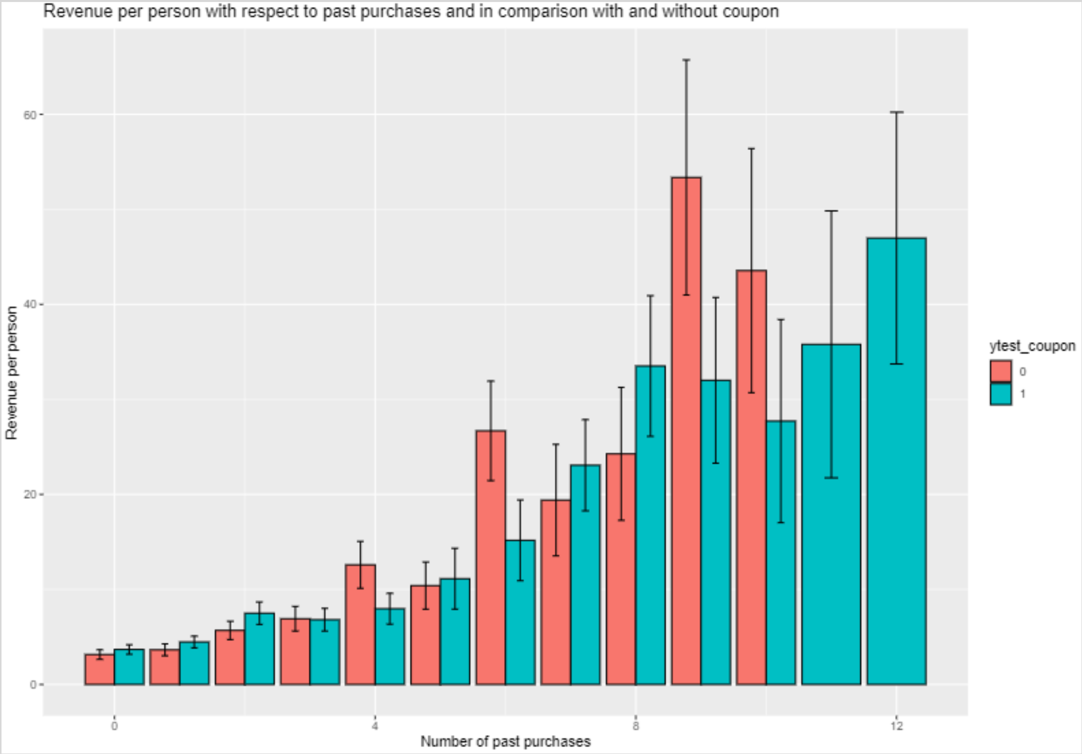
NO:

- GOOGLE & OTHERS

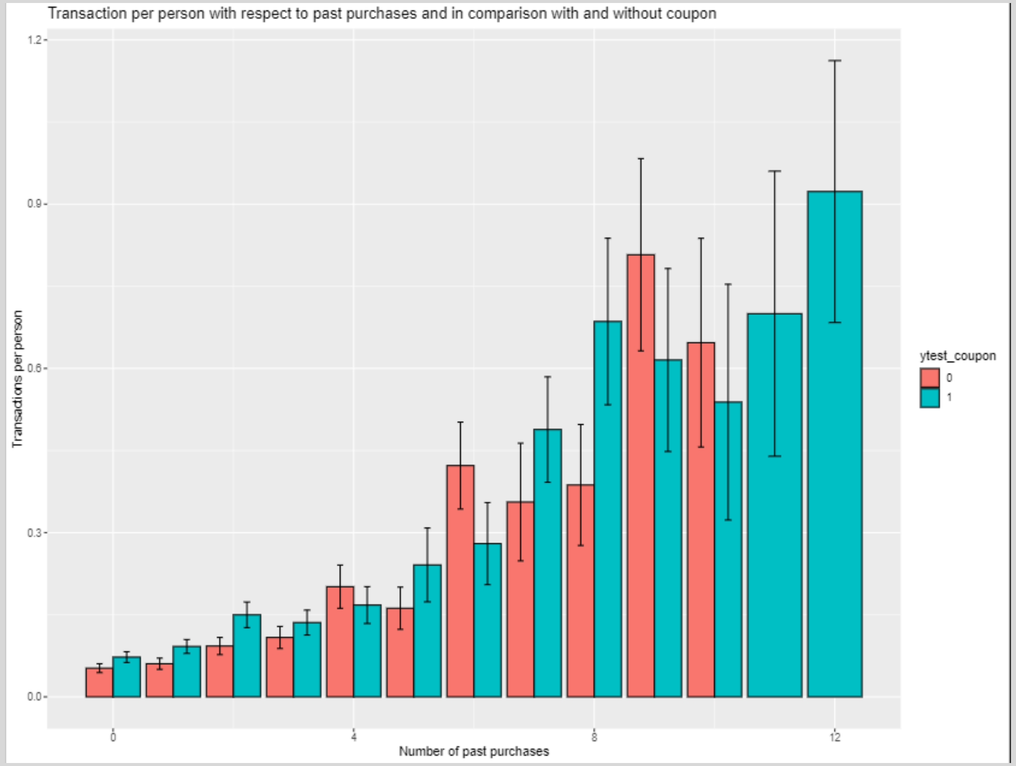
3) PAST PURCHASES

Revenue & Transactions with all acquisition types

```r
a <- data %>%
    group_by(test_coupon, num_past_purch) %>%
    summarize(
        number = n(), revenue_per_subject = sum(revenue_after) / n(),
        transactions_per_subject = sum(trans_after) / n(),
        error_revenue = std.error(revenue_after),
        error_trans = std.error(trans_after)
    ) %>% filter(number >= 10 )
a$ytest_coupon = factor(a$test_coupon)
print(n = 100, a)
```




```r
ggplot(a, aes(x = num_past_purch, y = revenue_per_subject, fill=ytest_coupon)) +
  geom_bar(stat="identity", color="black", position=position_dodge()) +
  geom_errorbar(aes(ymin = revenue_per_subject - error_revenue,
        ymax = revenue_per_subject + error_revenue), width = .2,
        position = position_dodge(.9)) + labs(title="Revenue per person with respect to past purchases and in comparison with
        and without coupon",  x = "Number of past purchases", y = "Revenue per person")
```



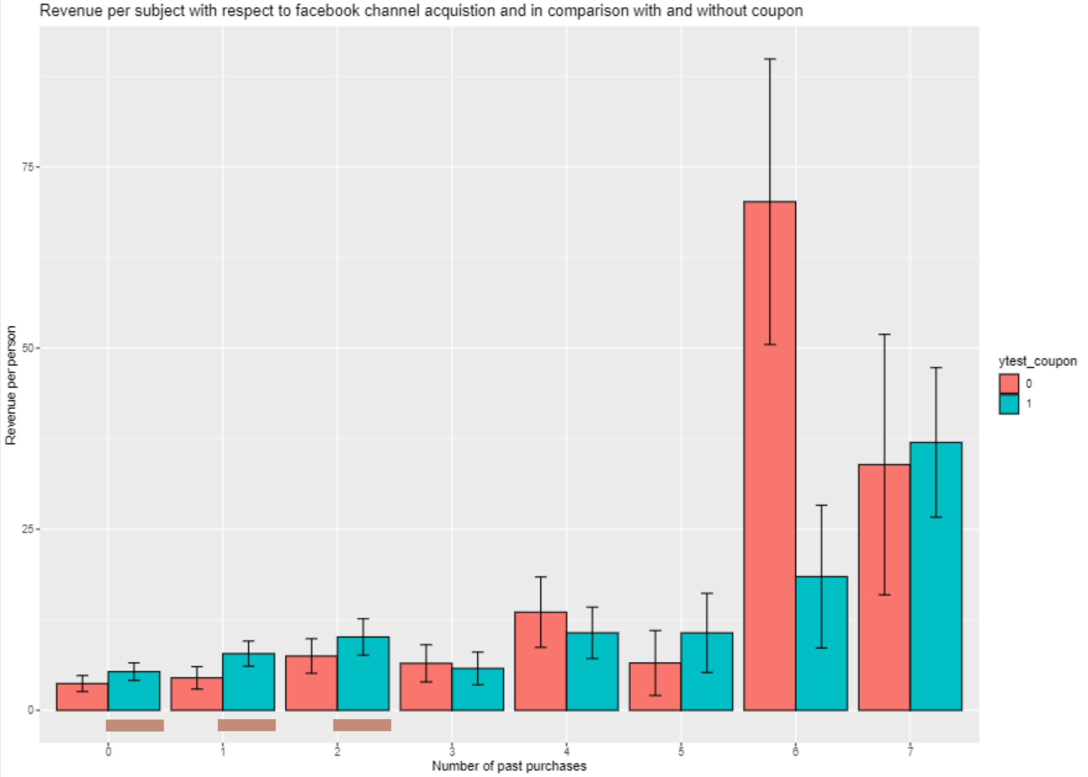

```r
ggplot(a, aes(x = num_past_purch, y = transactions_per_subject, fill=ytest_coupon)) +
  geom_bar(stat="identity", color="black", position=position_dodge()) +
  geom_errorbar(aes(ymin = transactions_per_subject - error_trans,
      ymax = transactions_per_subject + error_trans), width = .2,
      position = position_dodge(.9)) +
  labs(title="Transaction per person with respect to past purchases and in comparison with and without coupon",
     x = "Number of past purchases", y = "Transactions per person")
```

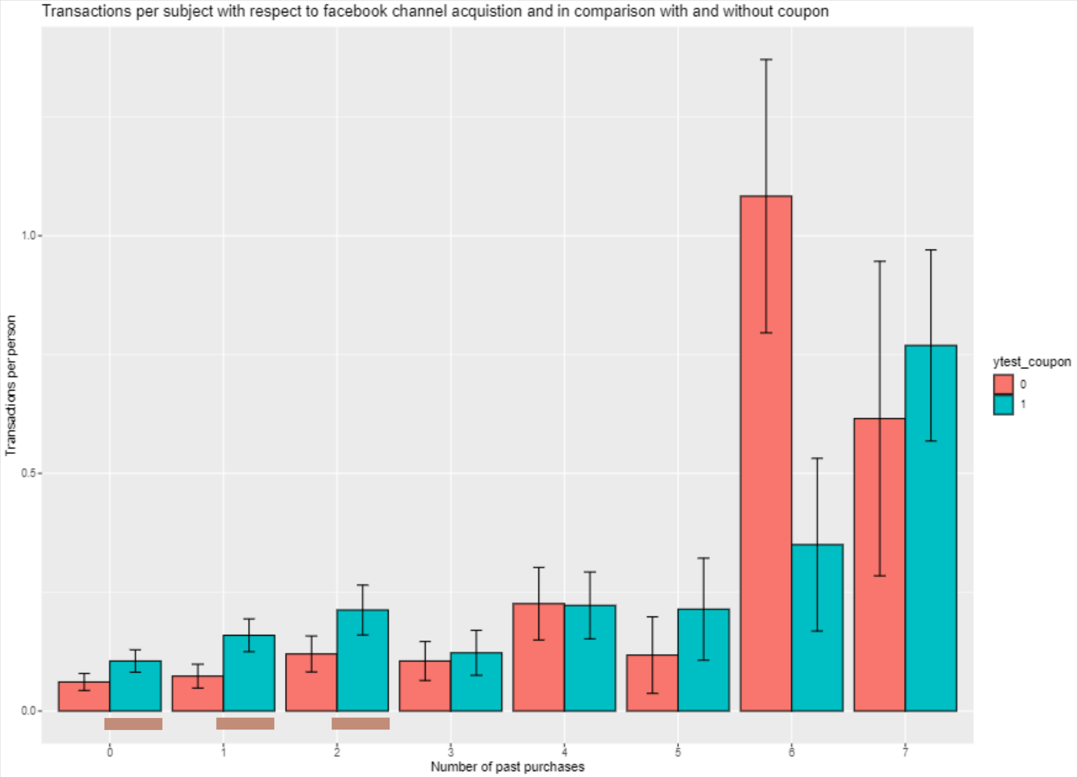
A: THE NUMBER OF PREVIOUS PURCHASES DRIVES THE REVENUE AND TRANSACTIONS PER CUSTOMER UNTIL THE 3. PURCHASE.
=> COUPON DRIVES EFFICENCY UNTIL 3. PURCHASE

```r
facebook <- data %>%
    filter(channel_acq == 2) %>%
    group_by(num_past_purch, test_coupon) %>%
    summarize(number = n(), revenue_per_subject = mean(revenue_after),
    error_revenue = std.error(revenue_after), transactions_per_subject = mean(trans_after),
    error_trans = std.error(trans_after)) %>%
    filter(number >= 10)
```

3) PAST PURCHASES

Revenue & Transactions with Facebook acquisition types





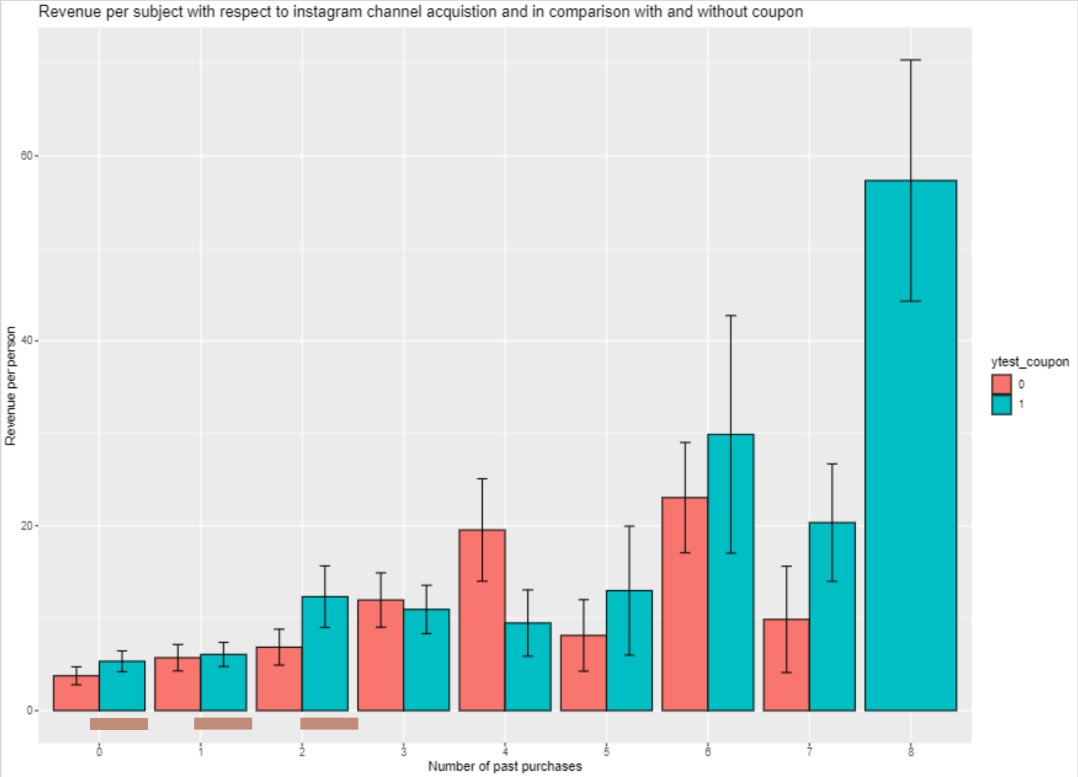

```r
facebook$ytest_coupon=factor(facebook$test_coupon)

ggplot(facebook, aes(x = num_past_purch, y = revenue_per_subject, fill=ytest_coupon)) +
  geom_bar(stat="identity", color="black", position=position_dodge()) +
  geom_errorbar(aes(ymin = revenue_per_subject - error_revenue,
        ymax = revenue_per_subject + error_revenue), width = .2,
        position = position_dodge(.9)) +
  labs(title="Revenue per subject with respect to facebook channel acquistion and in comparison with and without coupon",
       x = "Number of past purchases", y = "Revenue per subect")
```

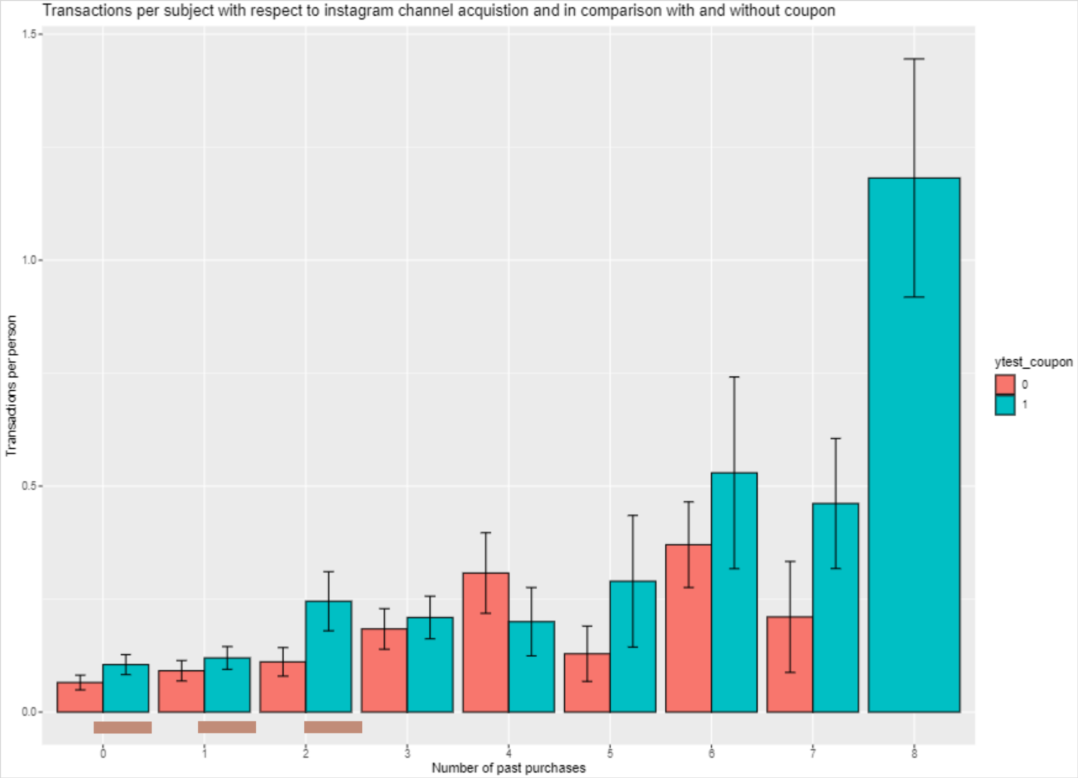
```r
ggplot(facebook, aes(x = num_past_purch, y = transactions_per_subject, fill=ytest_coupon)) +
  geom_bar(stat="identity", color="black", position=position_dodge()) +
  geom_errorbar(aes(ymin = transactions_per_subject - error_trans,
        ymax = transactions_per_subject + error_trans), width = .2,
        position = position_dodge(.9)) +
  labs(title="Transactions per subject with respect to facebook channel acquistion and in comparison with and without coupon",
       x = "Number of past purchases", y = "Transactions per subect")
```

```
instagram <- data %>%
    filter(channel_acq == 3) %>%
    group_by(num_past_purch, test_coupon) %>%
    summarize(number = n(), revenue_per_subject = mean(revenue_after),
    error_revenue = std.error(revenue_after), transactions_per_subject = mean(trans_after), error_trans= std.error(trans_after)) %>%
    filter(number >= 10)
```

3) PAST PURCHASES

Revenue & Transactions with Instagram acquisition types





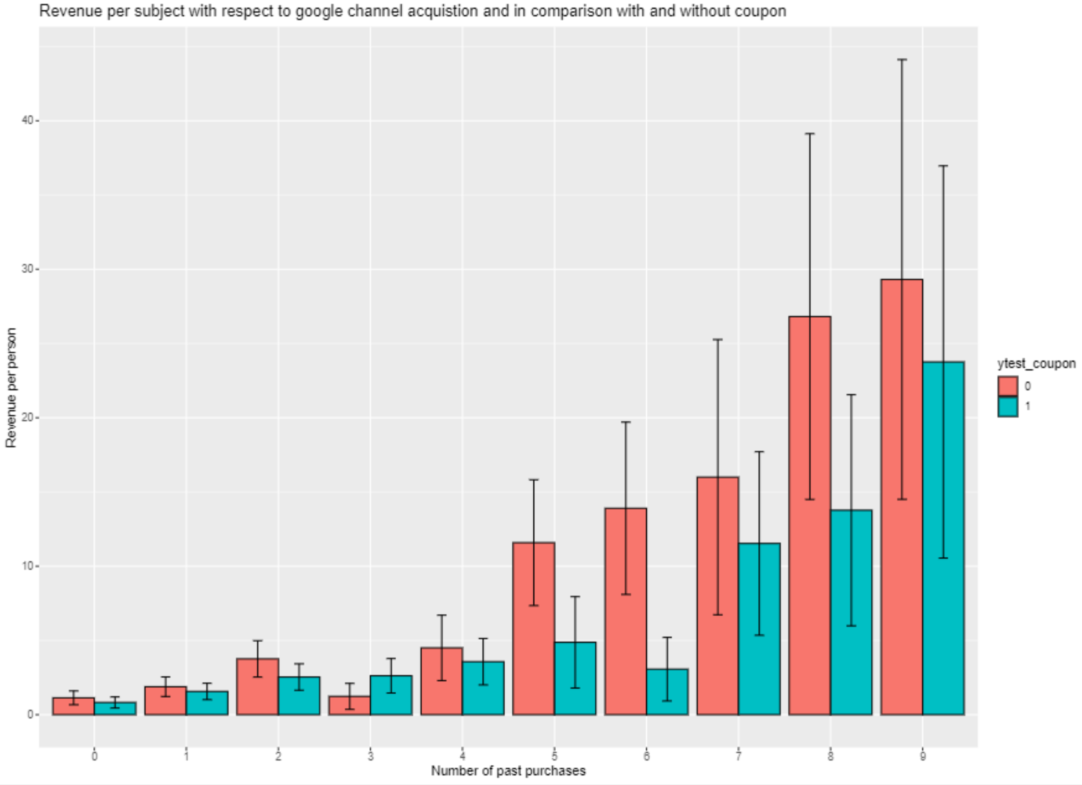

```
instagram$ytest_coupon=factor(instagram$test_coupon)

ggplot(instagram, aes(x = num_past_purch, y = revenue_per_subject, fill=ytest_coupon)) +
  geom_bar(stat="identity", color="black", position=position_dodge()) +
  geom_errorbar(aes(ymin = revenue_per_subject - error_revenue,
        ymax = revenue_per_subject + error_revenue), width = .2,
        position = position_dodge(.9)) +
  labs(title="Revenue per subject with respect to instagram channel acquistion and in comparison with and without coupon",
       x = "Number of past purchases", y = "Revenue per subect")
```

```
ggplot(instagram, aes(x = num_past_purch, y = transactions_per_subject, fill=ytest_coupon)) +
  geom_bar(stat="identity", color="black", position=position_dodge()) +
  geom_errorbar(aes(ymin = transactions_per_subject - error_trans,
        ymax = transactions_per_subject + error_trans), width = .2,
        position = position_dodge(.9)) +
  labs(title="Transactions per subject with respect to instagram channel acquistion and in comparison with and without coupon",
       x = "Number of past purchases", y = "Transactions per subect")
```

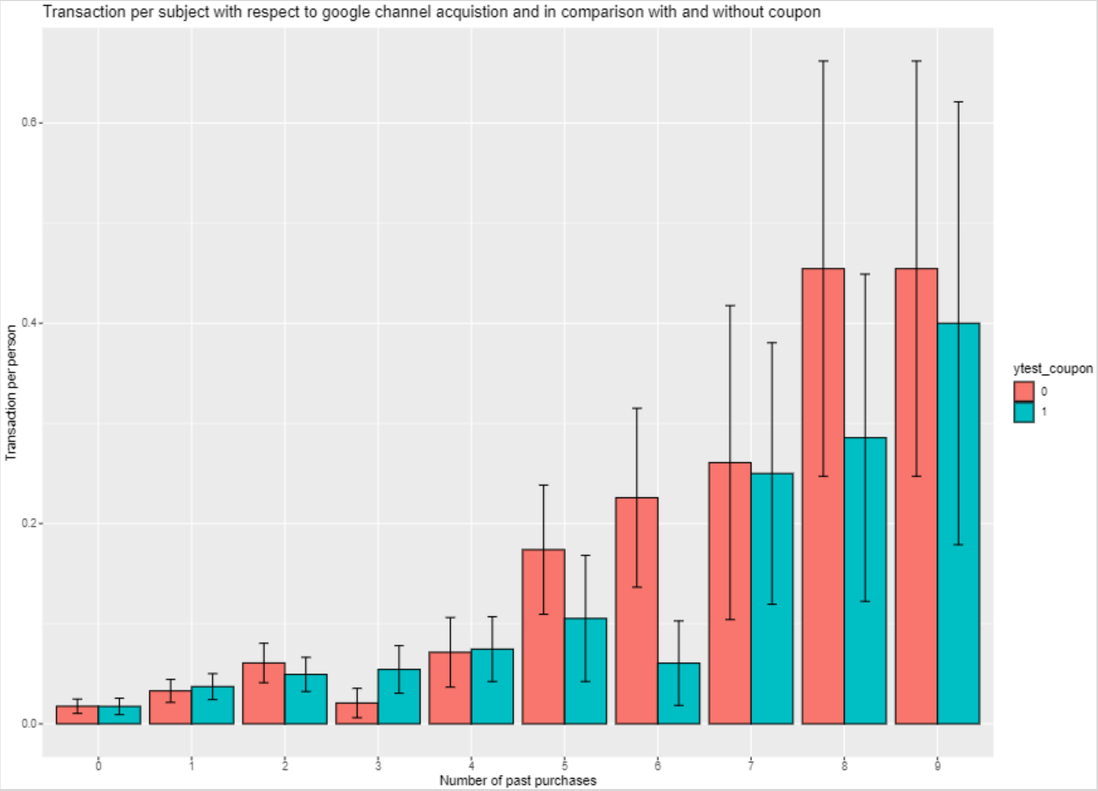
3) PAST PURCHASES

Revenue & Transactions with Google acquisition types

```
google <- data %>%
    filter(channel_acq == 1) %>%
    group_by(num_past_purch, test_coupon) %>%
    summarize(number = n(), revenue_per_subject = mean(revenue_after),
    error_revenue = std.error(revenue_after), transactions_per_subject = mean(trans_after), error_trans = std.error(trans_after)) %>%
    filter(number >= 10)
```




```
google$ytest_coupon=factor(google$test_coupon)

ggplot(google, aes(x = num_past_purch, y = revenue_per_subject, fill=ytest_coupon)) +
  geom_bar(stat="identity", color="black", position=position_dodge()) +
  geom_errorbar(aes(ymin = revenue_per_subject - error_revenue,
        ymax = revenue_per_subject + error_revenue), width = .2,
        position = position_dodge(.9)) +
  labs(title="Revenue per subject with respect to google channel acquistion and in comparison with and without coupon",
       x = "Number of past purchases", y = "Revenue per subect")
```

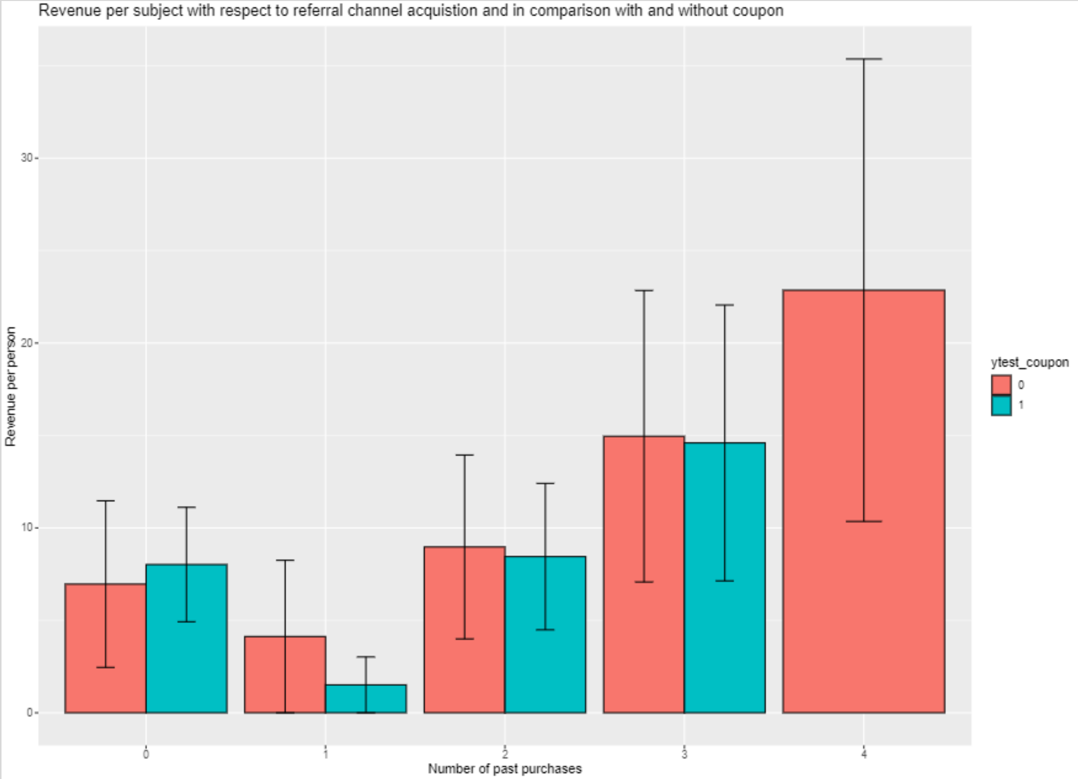


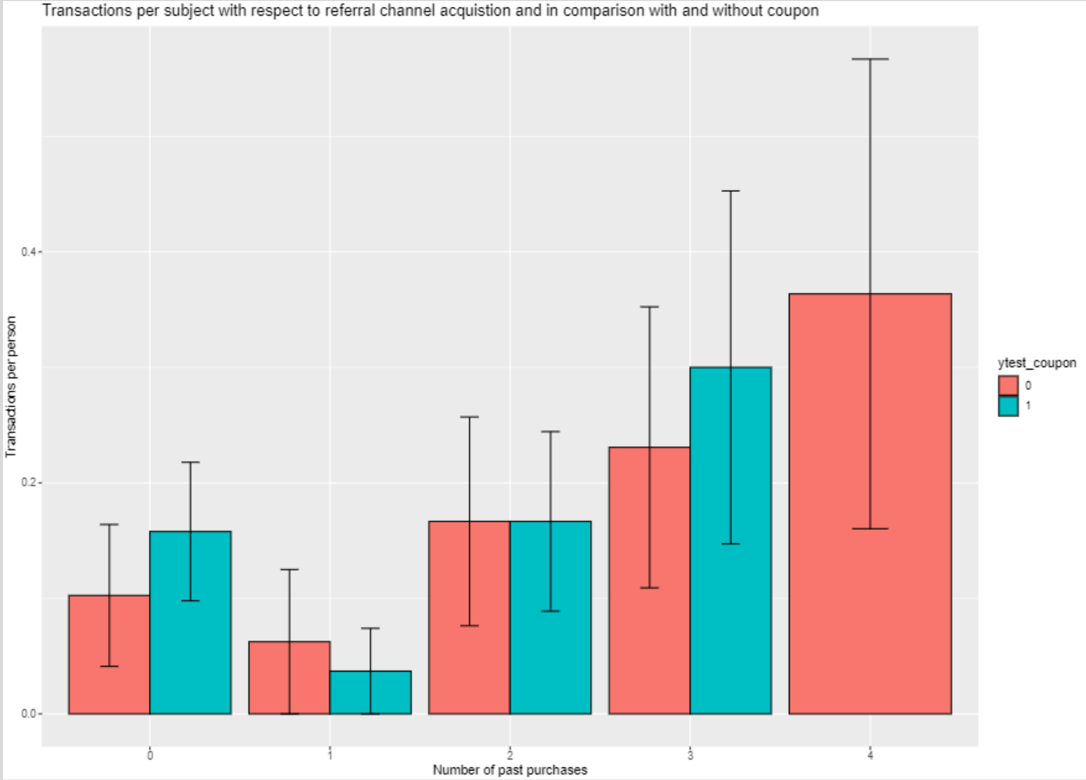

```
ggplot(google, aes(x = factor(num_past_purch), y = transactions_per_subject, fill=ytest_coupon)) +
  geom_bar(stat="identity", color="black", position=position_dodge()) +
  geom_errorbar(aes(ymin = transactions_per_subject - error_trans,
        ymax = transactions_per_subject + error_trans), width = .2,
        position = position_dodge(.9)) +
  labs(title="Transaction per subject with respect to google channel acquistion and in comparison with and without coupon",
       x = "Number of past purchases", y = "Transaction per person")
```

3) PAST PURCHASES

```r
referral <- data %>%
    filter(channel_acq == 4) %>%
    group_by(num_past_purch, test_coupon) %>%
    summarize(number = n(), revenue_per_subject = mean(revenue_after),
    error_revenue = std.error(revenue_after), transactions_per_subject = mean(trans_after), error_trans=std.error(trans_after)) %>%
    filter(number >= 10)
```

Revenue & Transactions with Referral acquisition types






```r
referral$ytest_coupon=factor(referral$test_coupon)

ggplot(referral, aes(x = num_past_purch, y = revenue_per_subject, fill=ytest_coupon)) +
  geom_bar(stat="identity", color="black", position=position_dodge()) +
  geom_errorbar(aes(ymin = revenue_per_subject - error_revenue,
        ymax = revenue_per_subject + error_revenue), width = .2,
        position = position_dodge(.9)) +
  labs(title="Revenue per subject with respect to referral channel acquistion and in comparison with and without coupon",
       x = "Number of past purchases", y = "Revenue per subect")
```

```r
ggplot(referral, aes(x = num_past_purch, y = transactions_per_subject, fill=ytest_coupon)) +
  geom_bar(stat="identity", color="black", position=position_dodge()) +
  geom_errorbar(aes(ymin = transactions_per_subject - error_trans,
        ymax = transactions_per_subject + error_trans), width = .2,
        position = position_dodge(.9)) +
  labs(title="Transactions per subject with respect to referral channel acquistion and in comparison with and without coupon",
       x = "Number of past purchases", y = "Transactions per subect")
```

3) Conclusion: PAST PURCHASES

IN GENERAL

THE NUMBER OF PREVIOUS PURCHASES DRIVES THE REVENUE AND TRANSACTIONS PER CUSTOMER INCLUDING 0, 1 AND 2 PURCHASES. THAT CAN BE ATTRIBUTED TO THE COUPON

-> COUPON DRIVES EFFICENCY UNTIL 2. PURCHASE

FOR THE DIFFERENT CHANNELS

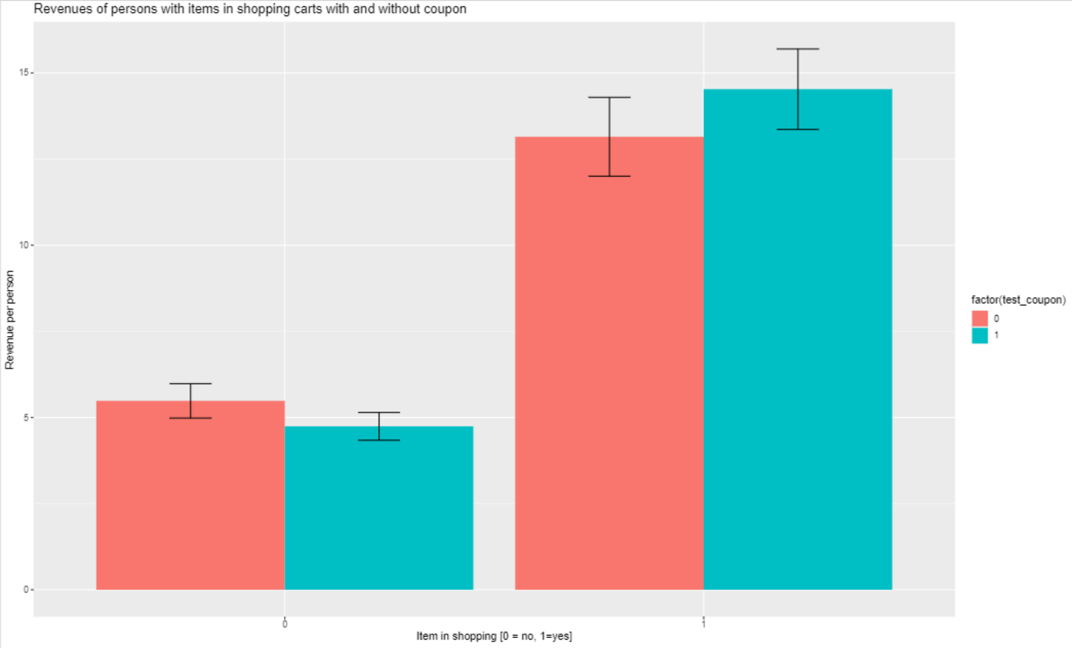
FOR THE FACEBOOK AND INSTAGRAM ACQUISTION CHANNELS AND FOR THE FIRST THREE PURCHASES THE REVENUE AND TRANSACTION ARE HIGHER WHEN A COUPON IS AVAILABLE

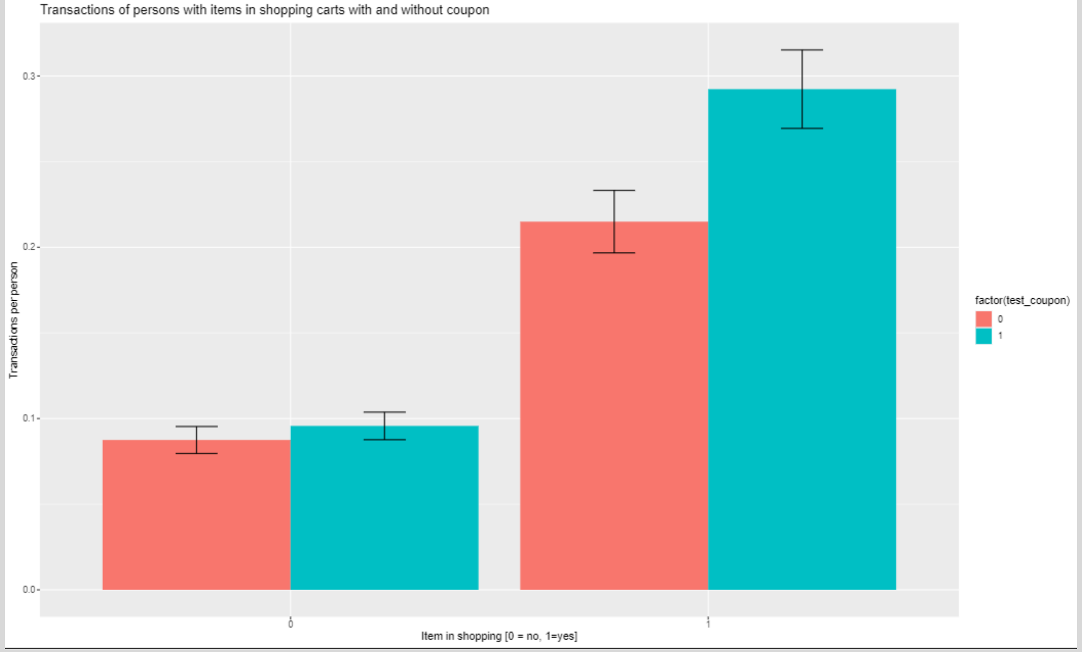
-> THIS IS GOING TO BE REFLECTED WHEN BUILDING UP THE NEW CAMPAIGN

3) SHOPPING CART

Revenue with all acquisition types

In general: If a person has an item in shopping cart and gets a coupon, both the transaction and revenue increase
Q: HOW IS THAT AFFECTED BY DIFFERENT CHANNELS?





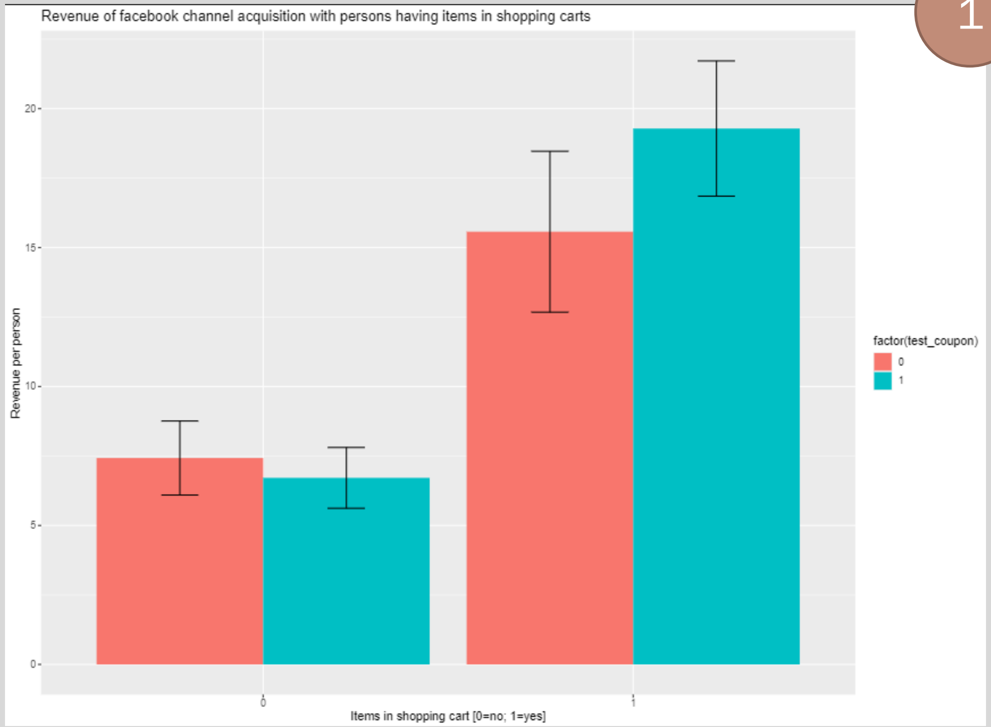

```r
shopping_cart_full <- data %>%
    group_by(shopping_cart, test_coupon) %>%
    summarize(number = n(), revenue = mean(revenue_after),
    error_revenue = std.error(revenue_after), transactions = mean(trans_after),
    error_trans = std.error(trans_after)) %>%
    filter(number >= 10)
```

```r
ggplot(shopping_cart_full, aes(fill = factor(test_coupon),
      y = transactions, x = factor(shopping_cart), group = test_coupon)) +
   geom_bar(position = "dodge", stat = "identity") +
   geom_errorbar(aes(ymin = transactions - error_trans,
      ymax = transactions + error_trans), width = .2,
      position = position_dodge(.9)) +
   labs(title="Transactions of persons with items in shopping carts with and without coupon",
      x = "Item in shopping [0 = no, 1=yes]", y = "Transactions per person")
```

3) SHOPPING CART

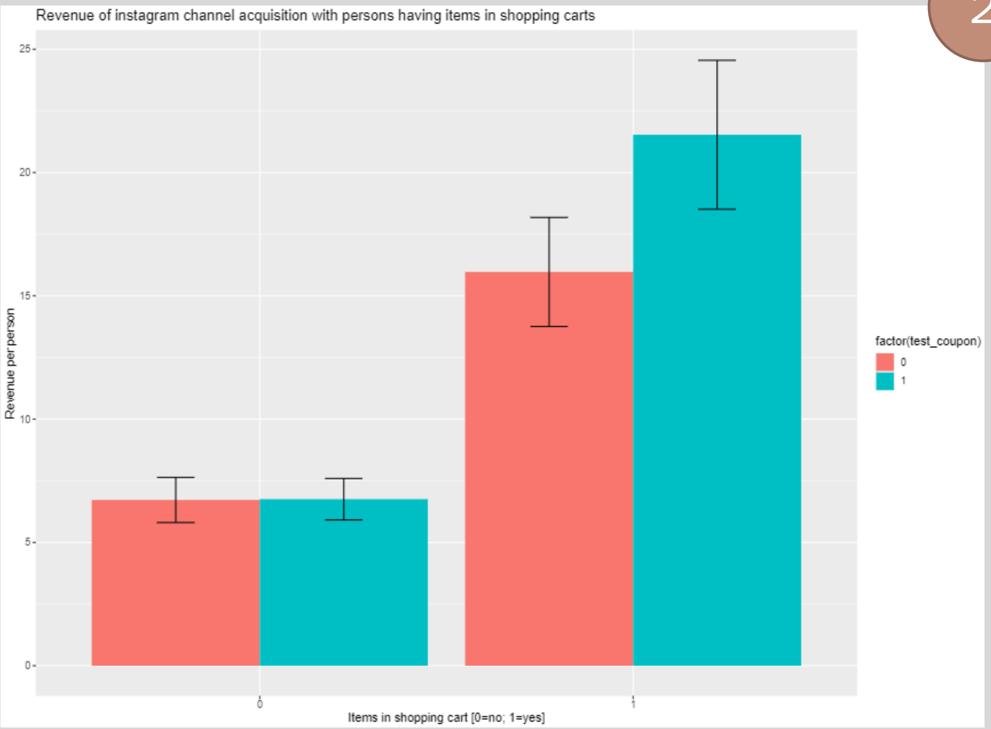
Revenue with Facebook and Instagram acquisition types

1




```r
ggplot(shopping_cart_fb, aes(fill = factor(test_coupon), y = revenue, x = factor(shopping_cart))) +
    geom_bar(position = "dodge", stat = "identity") +
    geom_errorbar(aes(ymin = revenue - error_revenue, ymax = revenue + error_revenue),
                  width = .2, position = position_dodge(.9)) +
    labs(title="Revenue of facebook channel acquisition with persons having items in shopping carts",
        x = "Items in shopping cart [0=no; 1=yes]", y = "Revenue per person")
```

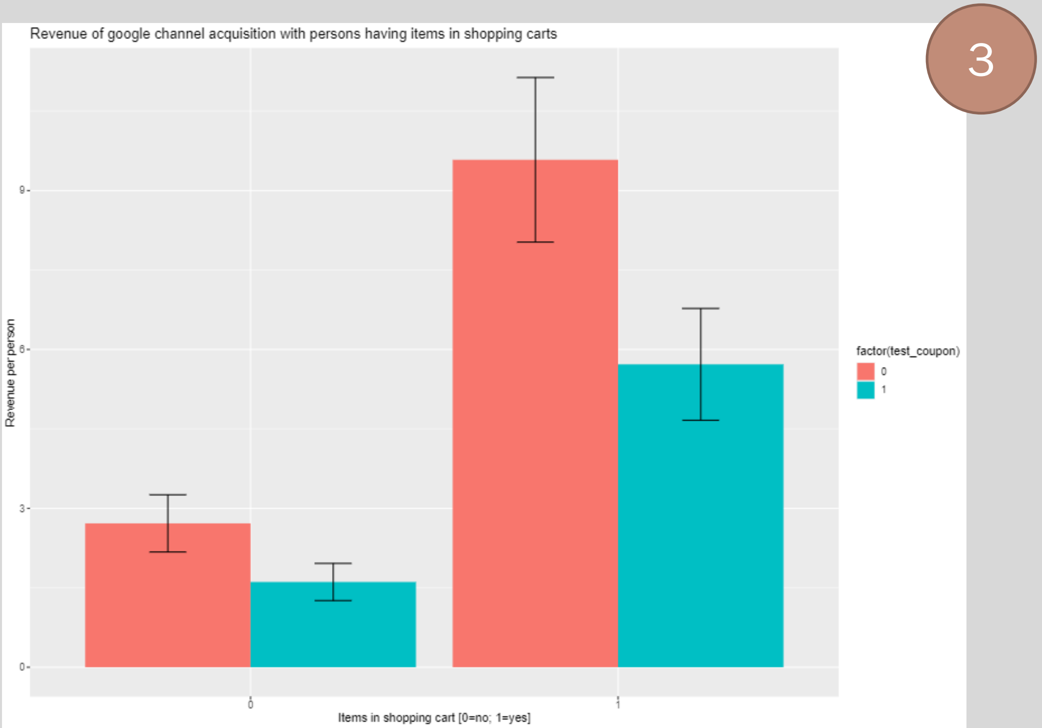
2



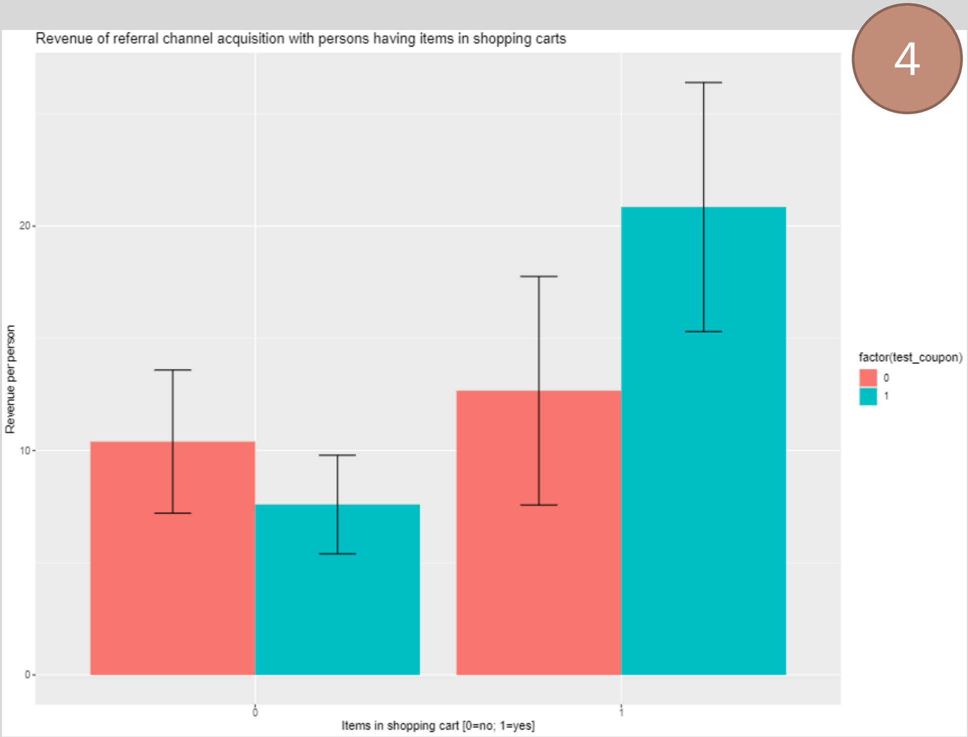

```r
shopping_cart_ig

ggplot(shopping_cart_ig, aes(fill = factor(test_coupon), y = revenue, x = factor(shopping_cart))) +
    geom_bar(position = "dodge", stat = "identity") +
    geom_errorbar(aes(ymin = revenue - error_revenue, ymax = revenue + error_revenue),
                  width = .2, position = position_dodge(.9)) +
    labs(title="Revenue of instagram channel acquisition with persons having items in shopping carts",
        x = "Items in shopping cart [0=no; 1=yes]", y = "Revenue per person")
```

3) SHOPPING CART

Revenue with Google and Referral acquisition types




```
ggplot(shopping_cart_google, aes(fill = factor(test_coupon), y = revenue, x = factor(shopping_cart))) +
    geom_bar(position = "dodge", stat = "identity") +
    geom_errorbar(aes(ymin = revenue - error_revenue, ymax = revenue + error_revenue),
                  width = .2, position = position_dodge(.9)) +
    labs(title="Revenue of google channel acquisition with persons having items in shopping carts",
         x = "Items in shopping cart [0=no; 1=yes]", y = "Revenue per person")
```




```
shopping_cart_ref

ggplot(shopping_cart_ref, aes(fill = factor(test_coupon), y = revenue, x = factor(shopping_cart))) +
    geom_bar(position = "dodge", stat = "identity") +
    geom_errorbar(aes(ymin = revenue - error_revenue, ymax = revenue + error_revenue),
                  width = .2, position = position_dodge(.9)) +
    labs(title="Revenue of referral channel acquisition with persons having items in shopping carts",
         x = "Items in shopping cart [0=no; 1=yes]", y = "Revenue per person")
```

3) Conclusion: SHOPPING CART AND CHANNEL

	shopping_cart	test_coupon	number	revenue	error_revenue	transactions	error_trans
	<int>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
1	0	0	74	10.4	3.19	0.176	0.0556
2	0	1	88	7.59	2.20	0.159	0.0454
3	1	0	39	12.7	5.09	0.205	0.0751
4	1	1	37	20.9	5.55	0.432	0.120

For Facebook (Graph: 1), Instagram (Graph: 2) and Referral (Graph: 4) channel acquisition the revenues increase when an item is in the shopping cart and when a coupon is available. For Google (Graph: 3) this does not hold true.

For Facebook (Graph: 1), the revenue even drops when a coupon is available but no item is in the shopping cart.

-> When building the new campaign coupon filter, persons from the referral channel acquisition type are not taken into account as only 37 persons coming from referral is too low for a statistical analysis.

3) QUESTION & ANSWER

WHAT DRIVES THE EFFECT OF THE COUPON & ARE THERE DIFFERENCES BETWEEN CHANNELS AND CUSTOMERS?

THE EFFECT OF THE COUPON IS MAINLY DRIVEN BY THE NUMBER OF PREVIOUS PURCHASES (INCLUDING 0, 1, 2 LAST PURCHAES) AS WELL AS ONE OR MORE ITEMS ALREADY IN THE SHOPPING CART.

THIS IS PARTICULARLY OBVIOUS FOR THE CUSTOMERS COMING FROM THE INSTAGRAM AND FACEBOOK ACQUISTION CHANNEL TYPES

FOR THIS REASON, THOSE ATTRIBUTES ARE REFLECTED IN THE NEW CAMPAIGN AND TO THOSE CUSTOMERS COUPONS WILL BE HANDED OUT

4) Marketing insights

Q: WHICH OF THE NEW CUSTOMERS SHOULD RECEIVE A COUPON?

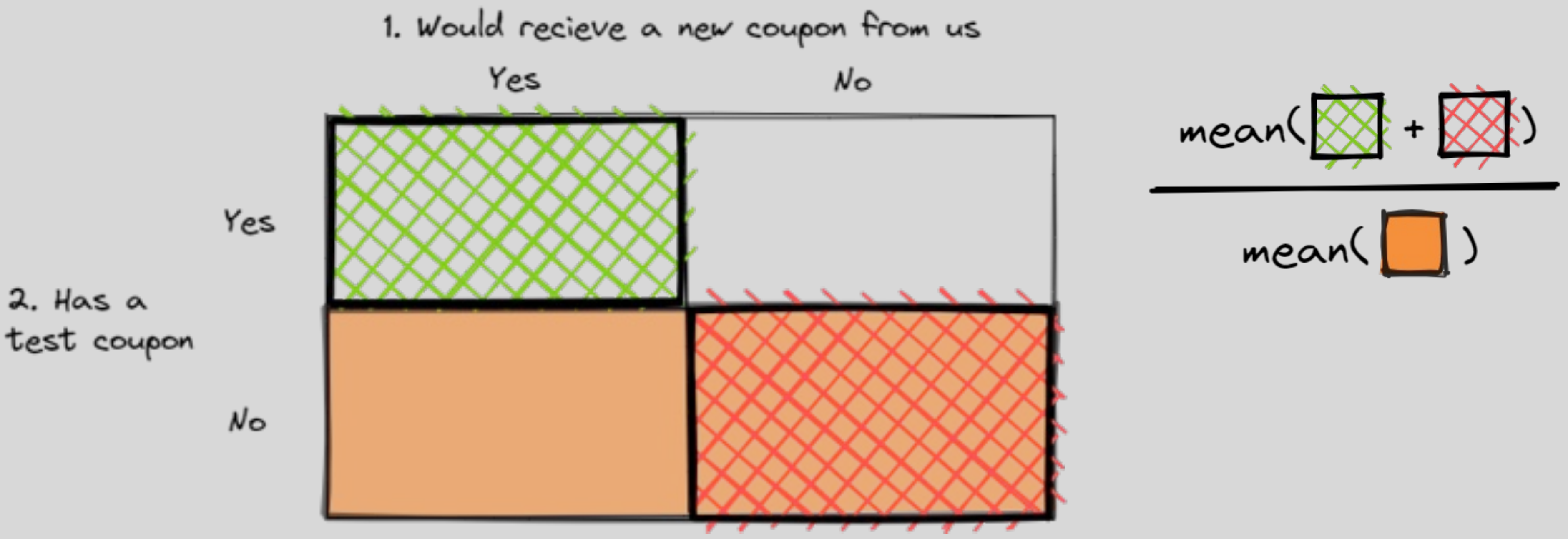
A: Based on the results of 3), the group of customers which are part of one of the following two groups should receive a coupon

- Facebook Users who have at least one item in their shopping cart, and have made less than 3 purchases in the past
- Instagram Users who have made less than 3 purchases in the path

4) Marketing insights

Q: WOULD ARTEA BE ABLE TO INCREASE TRANSACTIONS / REVENUES WITH THIS CAMPAIGN IF IT TARGETS THOSE CUSTOMERS? BY HOW MUCH?

To calculate a prediction for a mean revenue / transactions, we need to compare the mean values of the groups which are shown in the following diagram, using the displayed formula:



4) MARKETING INSIGHTS

```
chosen_cust_data <- data %>%
    filter(((((channel_acq == 2 & shopping_cart == 1) |  channel_acq == 3) & num_past_purch < 3) & test_coupon == 1) |
        (!((((channel_acq == 2 & shopping_cart == 1) |  channel_acq == 3) & num_past_purch < 3)) & test_coupon == 0)) %>%
    summarize(n(), mean_revenue = mean(revenue_after), std.error(revenue_after), mean_trans = mean(trans_after), std.error(trans_after))
```

```
# Group didn't recieve a coupon (control group)
control_cust_data <- data %>%
    filter(test_coupon == 0) %>%
    summarize(n(), mean_revenue = mean(revenue_after), std.error(revenue_after), mean_trans = mean(trans_after), std.error(trans_after))
```

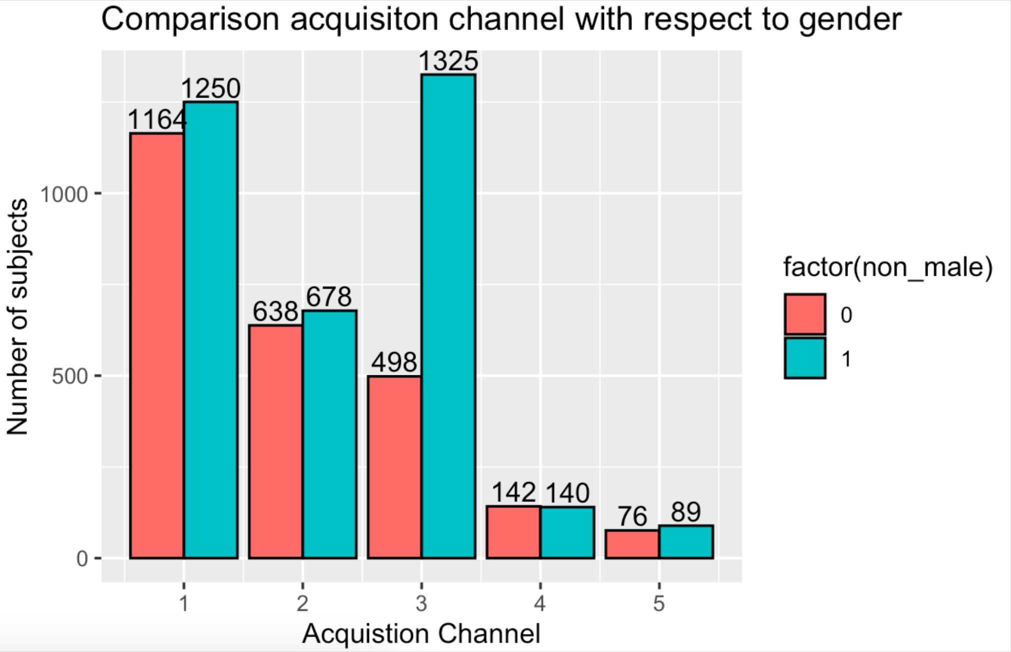
```
> increase <- (chosen_cust_data$mean_revenue / control_cust_data$mean_revenue) - 1
> print(increase)
[1] 0.09752805
```

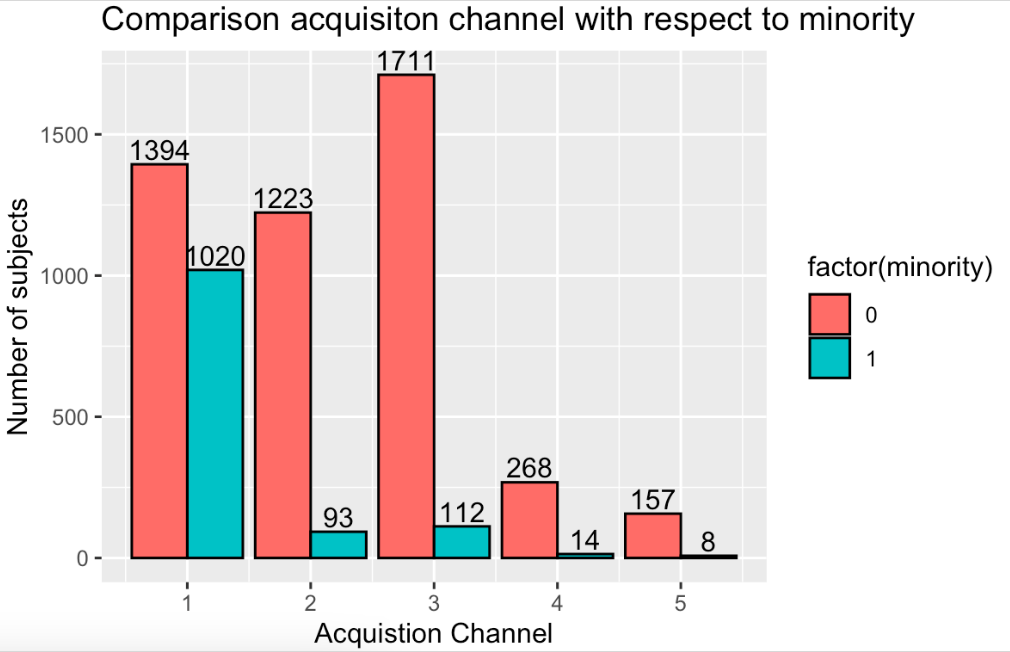
```
> increase <- (chosen_cust_data$mean_trans / control_cust_data$mean_trans) - 1
> print(increase)
[1] 0.1551261
```

A: Artea would be able to increase revenues with this campaign by 9.75% and transactions by 15.51% if it targets those customers

5) PROOF OF COUPON ALLOCATION

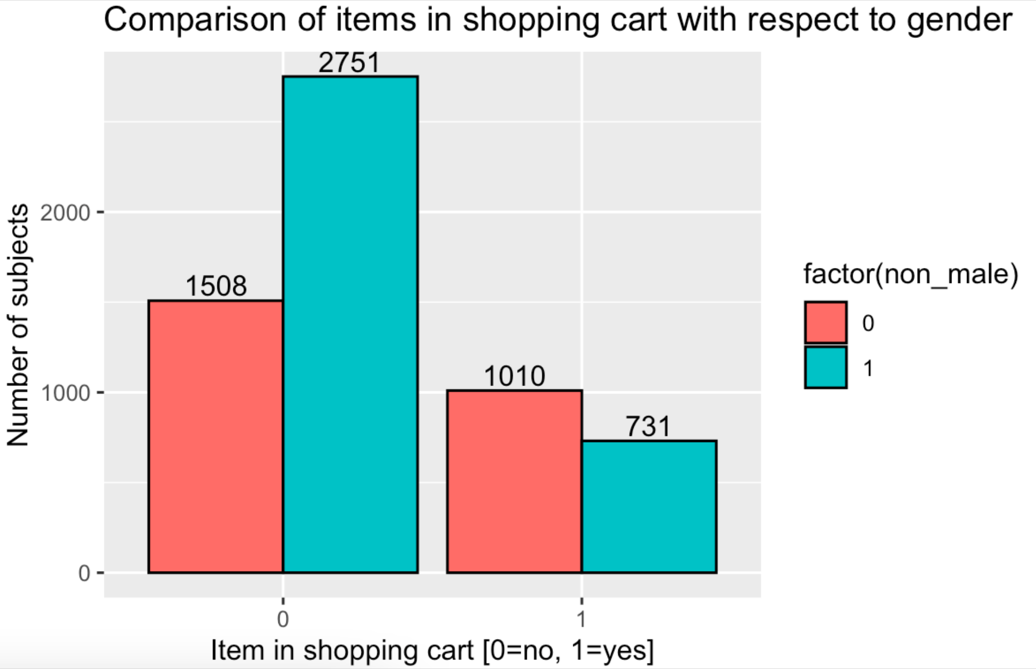
Q: WHAT DO WE LEARN FROM THE NEW DATA?

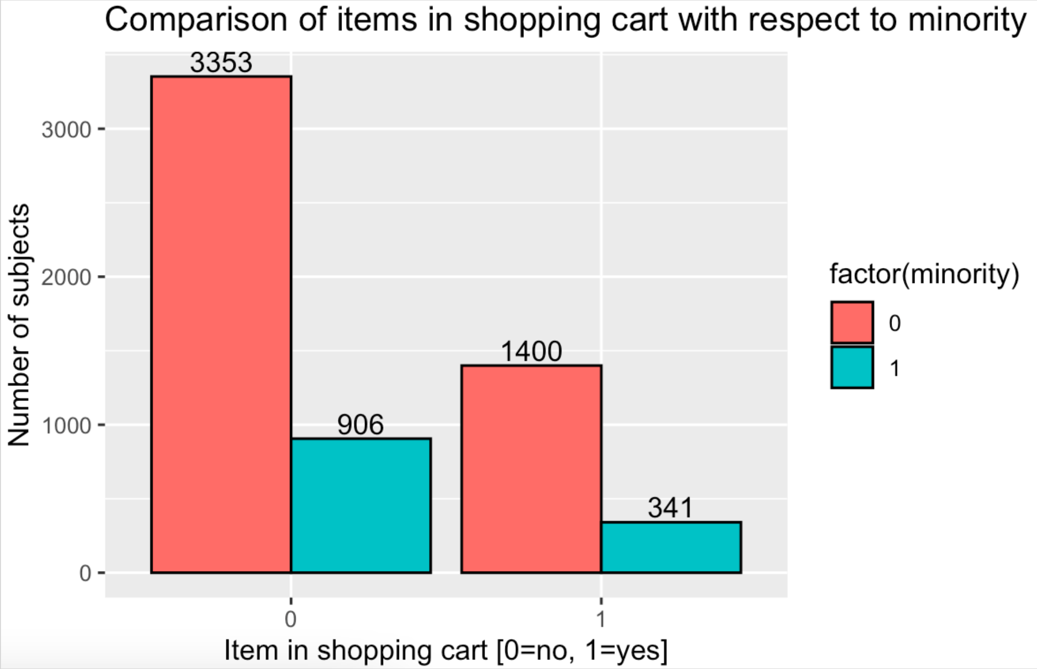




A (1/4): The number of non-male subjects for acquisition channel 1 (Google), 2 (Facebook) and especially 3 (Instagram) in the underlying data for different genders and minorities vary

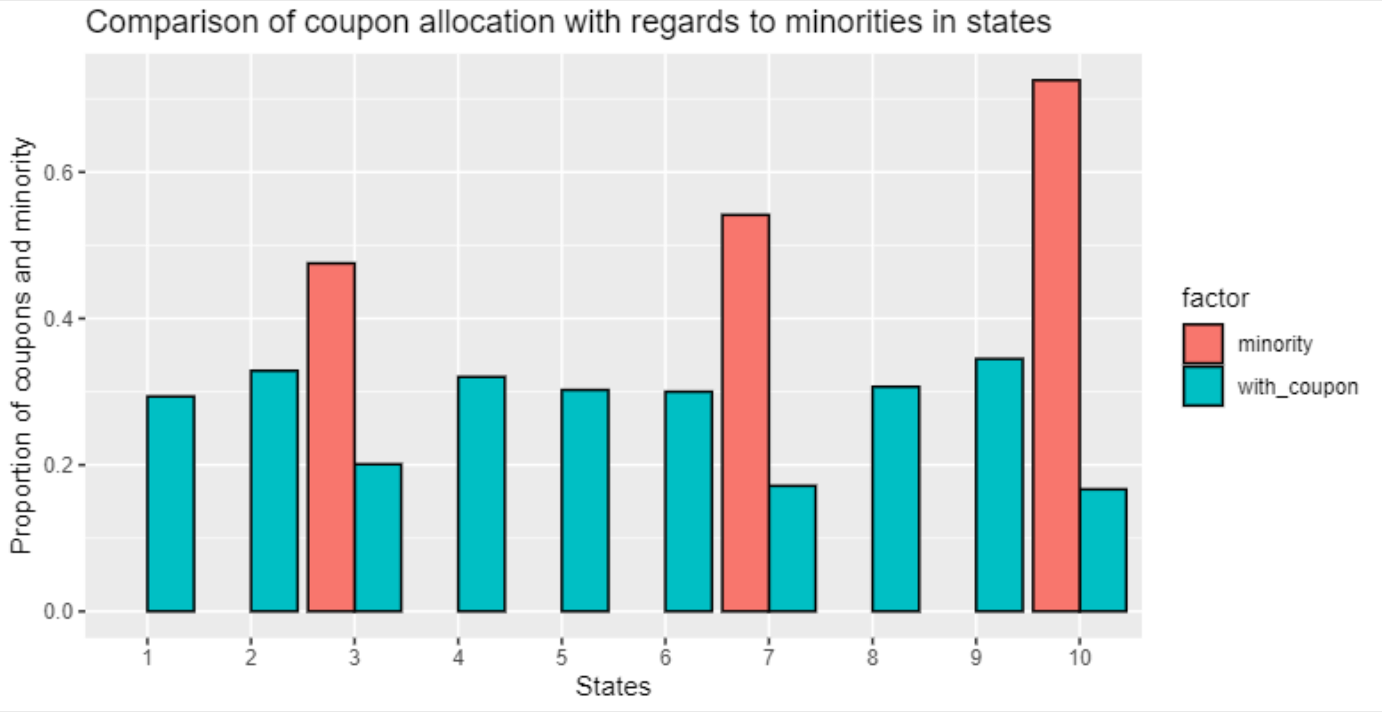
5) PROOF OF COUPON ALLOCATION





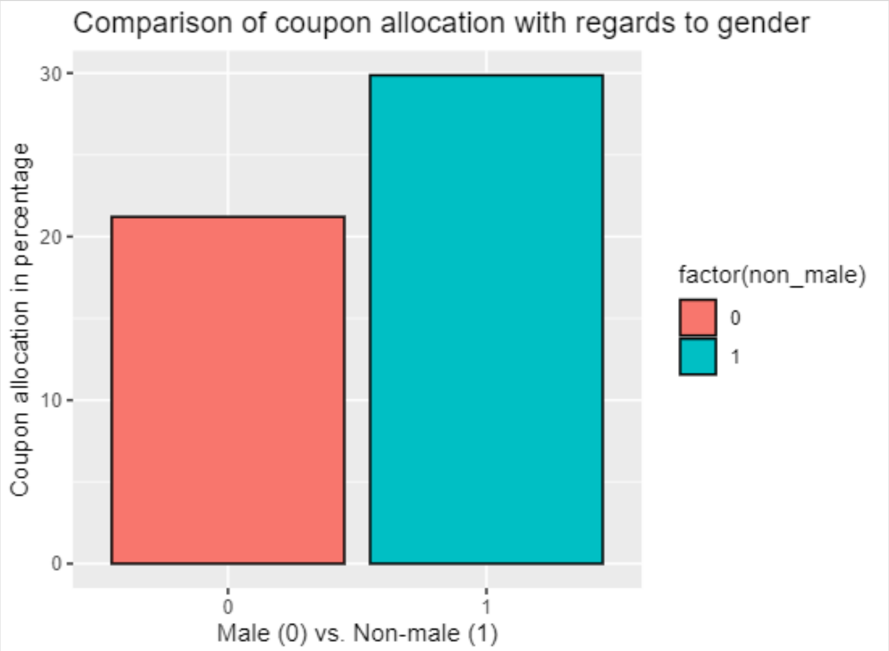
A (2/4): Non-males are more likely to have items in the shopping card. For minorities we don‘t see a big difference for this aspect (27.34% of non-minorities have no items in the shopping cart, compared to 29.46% for minorities)

5) PROOF OF COUPON ALLOCATION



A (3/4): According to the data, there are only certain states where a minority exists. In states with minorities, our coupon allocation is significantly less in terms of percentage

5) PROOF OF COUPON ALLOCATION



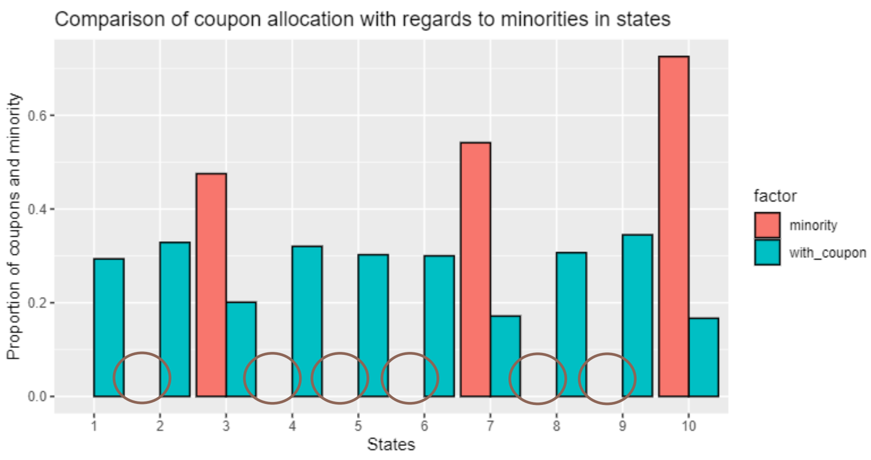
A (4/4): With our coupon allocation, the percentage of males who get a coupon is lower than the percentage of non-males getting a coupon. Same applies to non-minorities and minorities. Therefore, higher shares of non-males and non-minorities are receiving coupons.

5) PROOF OF COUPON ALLOCATION

Q: HOW SHOULD WE REACT TO THESE DATA PATTERN? WOULD WE BE DISSCRIMINATING IF WE TARGETED THE PROMOTIONS BASED ON CHANNEL OF ACQUISITION? WAS IST LEGAL AND ETHICAL TO DO SO? HOW SHOULD THE TARGETING POLICY CHANGE? WHAT SCHOULD WE DO, MOVING FORWARD?

A: Our allocation seems to be favoring non-males and non-minorities. But from our analysis, this does not appear to be an issue, because the reason for this favoring is that non-males more often are having items in their shopping cart and non-minorities more often are using Instagram and Facebook.
It seems to be an inherent gap in these dimensions of our data, which we are not actively targeting but which as a result is reflected in the final data as well. Additionally, everyone is free to use social networks or our shopping cart. Therefore, from our understanding this would be legal and ethical, and we would use our defined targeting groups as planned.

5) PROOF OF COUPON ALLOCATION



Q: WOULD WE ADVISE TO BUY DEMOGRAPHICS DATA?

A: No, we would not, because they do not change our selection of the target group for our customers, and therefore don‘t provide any value for us. Furthermore, the quality of the data seems to be lacking, when it comes to the distribution of minority groups amongst states, with most states even having an amount of people from minorities of zero. This could be due to either missing or inaccurate data, hence strengthening our conclusion.

Q: SHOULD WE USE DEMOGRAPHICS DATA FOR TARGETING?

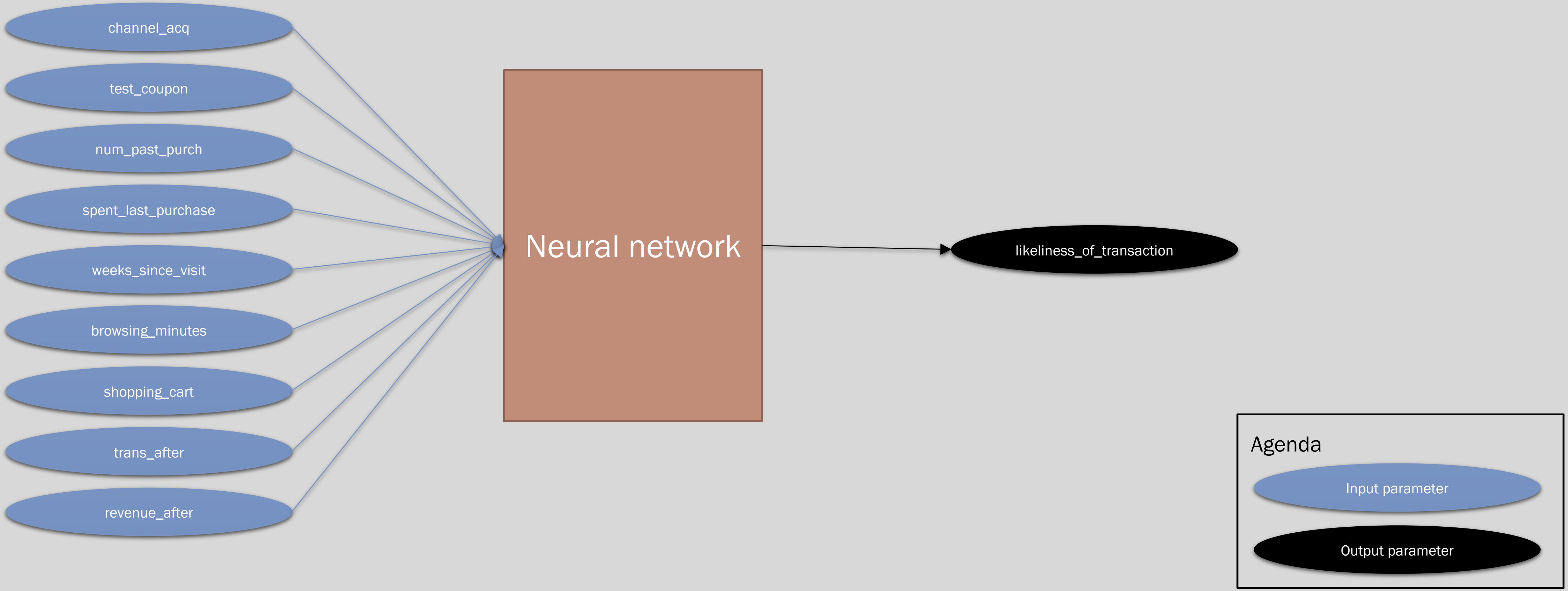
A: Using demographics data or data which is directly linked to demogaphics data e.g. the states for targeting would lead to discrimination, because in this case, the users wouldn‘t be able to influence their parameters. Therefore, we should not use demographics data for targeting.

ANOTHER APPROACH: USING MACHINE LEARNING TO PREDICT THE EFFICIENCY OF SENDING COUPONS TO CUSTOMERS

Implementation (Jupyter Notebook): https://github.com/tjbnde/abtest-artea/blob/master/Tensorflow-version/machinelearning_implementation.ipynb

Idea 1/2

◦ The idea is to train a machine learning model using the given data. This neural network uses the following inputs to calculate the likeliness of a transaction for a given user and outputs this value as an output parameter as shown below.



Idea 2/2

- We can use this model, to predict the likeliness of a purchase for the same customer two times:
  - One time with the parameter test_coupon set to 0
  - One time with the parameter test_coupon set to 1
- This allows us to compare the likeliness of a purchase for this user, in case we send him a coupon, and in case we don’t send him a coupon
- Based on those probabilities, we can decide, whether we want to send a coupon to the customer or not

Step 1: Importing test- and trainingdata

```
# Columns of importet data: channel_acq,test_coupon,num_past_purch,spent_last_purchase,weeks_since_visit,browsing_minut
path = 'trainingdata.csv'
trainingdata = read_csv(path, header=None)
path = 'testdata.csv'
testdata = read_csv(path, header=None)

# split into input and output columns
X, y = trainingdata.values[:, :-2], trainingdata.values[:, -2]

# ensure all data are floating point values
X = X.astype('float32')
# encode strings to integer
y = LabelEncoder().fit_transform(y)
# split into train and test datasets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1)
```

Step 2: Training the model

- We are using half of the given dataset to train our model. This dataset includes the inputs, which were shown before, and the likeliness of a transaction (based on column “trans_after” of our dataset)

```
n_features = X_train.shape[1]
# define model
model = Sequential()
model.add(Dense(10, activation='relu', kernel_initializer='he_normal', input_shape=(n_features,)))
model.add(Dense(8, activation='relu', kernel_initializer='he_normal'))
model.add(Dense(1, activation='sigmoid'))
# compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
# fit the model
model.fit(X_train, y_train, epochs=150, batch_size=32, verbose=0)
# evaluate the model
loss, acc = model.evaluate(X_test, y_test, verbose=0)
print('Test Accuracy: %.3f' % acc)
```

```
Test Accuracy: 0.900
```

Step 3: Making predictions

- After training our model, we can calculate a prediction for a given user for both cases: when sending him a coupon and when not sending him a coupon

Input parameter: test_coupon

```
# make a prediction
input_without_coupon = [5,0,6,73.49,4,8,0]
output_without_coupon = model.predict([input_without_coupon])
input_with_coupon = [5,1,6,73.49,4,8,0]
output_with_coupon = model.predict([input_with_coupon])
print('Predicted value (likeliness_of_transaction) without coupon: %.3f' % output_without_coupon)
print('Predicted value (likeliness_of_transaction) with coupon: %.3f' % output_with_coupon)
```

```
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 37ms/step
Predicted value (likeliness_of_transaction) without coupon: 0.470
Predicted value (likeliness_of_transaction) with coupon: 0.433
```

Output parameters: likeliness_of_transaction

- In this example, we can see, that the likeliness of a transaction is slightly higher for the given customer if we would not send him a coupon (0.47) compared to if we would send him a coupon (0.433). **Therefore, we would not send a coupon to this given user.**

Evaluating the quality of our neural network

- To evaluate the quality of our neural network, we used the other half of our dataset (which wasn't used for training the model) and did predictions for all customers in this half of the dataset. We exported those predictions to a csv, and also added the correct trans_after value to the exported file using the following lines of code:

```python
# Calculate predictions for data in testdata.csv and export them to outputfile (outputdata.csv)
outputdata = []
X, y = testdata.values[:, :-2], testdata.values[:, -2:]
for i in range(len(testdata)):
    yhat = model.predict([list(X[i])])
    outputdata.append(numpy.append(numpy.append(X[i], y[i]), yhat))

numpy.savetxt("outputdata.csv", outputdata, delimiter=",", fmt='%s')
```

- We then imported this csv into R and ran some further analysis on those predictions

Evaluating the quality of our neural network

- To check if the overall predictions made by our neural network are qualitative, we compared the mean value of trans_after for all customers to the mean value of trans_after for customers where our neural network predicted a likeliness_of_transaction of 0.3 or higher.

```
> mean(data$trans_after[data$prediction_using_neural_network > 0.3])
[1] 0.6133829
> mean(data$trans_after)
[1] 0.1428
```

- It turned out, that for those customers, the mean value of trans_after is significantly higher than the mean value of trans_after for the whole dataset.
- This proofs that the predictions made by our neural network are actually able to rate the likeliness of a transaction for different users with different parameters. We can therefore say that the neural network can be used to rate the likeliness for a transaction for our user and that those results are fairly reliable.
- The question to answer is, whether the input parameter “test_coupon” actually would be reliable enough to produce predictions which would increase the transactions made after sending coupons to customers which were selected using the neural network.
  - To validate this, we would need to run another AB test with group A being a set of customers where we send coupons to customers who we select using the neural network and don’t send coupons to customers which weren’t selected by the neural network, and group B being a control group of customers, who we don’t send a coupon to.

Thank you!

- Team:
  - Tobias Jansen (tobias.jansen@sap.com)
  - Sinah-Nikola Kaefferlein (sinah-nikola.kaefferlein@mercedes-benz.com)
  - Leon Deng (leon.deng@hpe.com)
  - Anne Huesges (anne.huesges@hpe.com)
- Sourcecode and data: https://github.com/tjbnde/abtest-artea