task2

January 6, 2024

1 Quantium Virtual Internship - Retail Strategy and Analytics - Task 2

1.1 Load required libraries and datasets

[ ]:
```
library(data.table)
library(ggplot2)
library(tidyr)
```

Point the filePath to where you have downloaded the datasets to and

assign the data files to data.tables

[ ]:
```
data <- fread(paste0("QVI_data.csv"))
#### Set themes for plots
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
```

1.2 Select control stores

The client has selected store numbers 77, 86 and 88 as trial stores and want control stores to be established stores that are operational for the entire observation period. We would want to match trial stores to control stores that are similar to the trial store prior to the trial period of Feb 2019 in terms of : - Monthly overall sales revenue - Monthly number of customers - Monthly number of transactions per customer Let's first create the metrics of interest and filter to stores that are present throughout the pre-trial period.

[ ]:
```
#### Calculate these measures over time for each store
#### Add a new month ID column in the data with the format yyyymm.
data[, YEARMONTH := year(DATE)*100 + month(DATE)]
data
#### Next, we define the measure calculations to use during the analysis.
# For each store and month calculate total sales, number of customers,␣
 ↪transactions per customer, chips per customer and the average price per unit.
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR) ,
                            nTxnPerCust = uniqueN(TXN_ID)/
 ↪uniqueN(LYLTY_CARD_NBR),
```

```
                              nChipsPerTxn = sum(PROD_QTY)/uniqueN(TXN_ID),
                              avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY))
                          , by = c("STORE_NBR", "YEARMONTH")][order(STORE_NBR,␣
  ↪YEARMONTH) ]
#### Filter to the pre-trial period and stores with full observation periods
storesWithFullObs <- unique(measureOverTime[, .N, STORE_NBR][N == 12,␣
  ↪STORE_NBR])
preTrialMeasures <- measureOverTime[YEARMONTH < 201902 & STORE_NBR %in%
storesWithFullObs, ]
```

A data.table: 264834 × 13

LYLTY_CARD_NBR <int>	DATE <IDate>	STORE_NBR <int>	TXN_ID <int>	PROD_NBR <int>	PI <c
1000	2018-10-17	1	1	5	Na
1002	2018-09-16	1	2	58	Re
1003	2019-03-07	1	3	52	Gr
1003	2019-03-08	1	4	106	Na
1004	2018-11-02	1	5	96	W
1005	2018-12-28	1	6	86	Ch
1007	2018-12-04	1	7	49	In
1007	2018-12-05	1	8	10	RI
1009	2018-11-20	1	9	20	Do
1010	2018-09-09	1	10	51	Do
1010	2018-12-14	1	11	59	Ol
1011	2018-07-29	1	12	84	Gr
1011	2018-11-08	1	13	59	Ol
1011	2018-12-01	1	14	49	In
1011	2018-12-19	1	15	1	Sm
1012	2019-03-15	1	16	20	Do
1012	2019-06-19	1	17	3	Ke
1013	2019-03-04	1	18	93	Do
1013	2019-03-07	1	19	91	CC
1016	2019-04-19	1	20	74	To
1016	2019-06-09	1	21	63	Ke
1018	2018-09-03	1	22	3	Ke
1018	2018-11-28	1	23	97	RI
1018	2019-06-20	1	24	38	In
1019	2019-01-27	1	25	84	Gr
1020	2018-08-16	1	26	19	Sm
1020	2018-10-02	1	27	7	Sm
1020	2019-05-02	1	28	84	Gr
1022	2018-10-24	1	29	3	Ke
1023	2019-05-27	1	30	53	RI
...	...	...	...	...	...
2330031	2018-07-07	77	236716	102	Ke
2330041	2018-09-23	77	236718	24	Gr
2330051	2018-09-07	77	236719	78	Th
2330081	2019-06-22	77	236723	30	Do
2330121	2018-11-26	77	236729	10	RI
2330171	2019-06-20	77	236737	5	Na
2330191	2018-11-07	77	236740	67	RI
2330211	2018-07-17	77	236744	94	Bu
2330251	2018-11-29	77	236747	14	Sm
2330271	2019-06-29	77	236749	114	Ke
2330291	2018-11-09	77	236752	89	Ke
2330291	2018-11-10	77	236753	3	Ke
2330291	2019-06-18	77	236754	83	W
2330311	2018-11-09	77	236755	90	To
2330321	2018-07-30	77	236756	71	Tw
2330331	2018-11-18	77	236760	95	Su
2330431	2018-07-31	77	236770	50	To
2330461	2018-07-21	77	236777	87	In
2330501	2019-06-20	77	236780	63	Ke
2370001	2018-08-10	88	240064	102	Ke

Now we need to work out a way of ranking how similar each potential control store is to the trial store. We can calculate how correlated the performance of each store is to the trial store. Let's write a function for this so that we don't have to calculate this for each trial store and control store pair.

[ ]:
```
calculateCorrelation <- function(inputTable, metricCol, storeComparison) {
  calcCorrTable = data.table(Store1 = numeric(), Store2 = numeric(),␣
 ↪corr_measure =
  numeric())

  storeNumbers <- unique(inputTable[, STORE_NBR])

  for (i in storeNumbers) {
    calculatedMeasure = data.table("Store1" = storeComparison,
                                   "Store2" = i,
                                   "corr_measure" = cor( inputTable[STORE_NBR␣
 ↪== storeComparison,
                                                                    ␣
 ↪eval(metricCol)], inputTable[STORE_NBR == i,
                                                                          ␣
 ↪                eval(metricCol)]))
    calcCorrTable <- rbind(calcCorrTable, calculatedMeasure)
  }
  return(calcCorrTable)
}
```

Apart from correlation, we can also calculate a standardised metric based on the absolute difference between the trial store's performance and each control store's performance. Let's write a function for this.

[ ]:
```
calculateMagnitudeDistance <- function(inputTable, metricCol, storeComparison) {
calcDistTable = data.table(Store1 = numeric(), Store2 = numeric(), YEARMONTH =
numeric(), measure = numeric())
  storeNumbers <- unique(inputTable[, STORE_NBR])
  for (i in storeNumbers) {
  calculatedMeasure = data.table("Store1" = storeComparison
                                 , "Store2" = i
                                 , "YEARMONTH" = inputTable[STORE_NBR ==
storeComparison, YEARMONTH]
                                 , "measure" = abs(inputTable[STORE_NBR ==
storeComparison, eval(metricCol)]
                                                   - inputTable[STORE_NBR == i,
eval(metricCol)])
                                  )
    calcDistTable <- rbind(calcDistTable, calculatedMeasure)
}
#### Standardise the magnitude distance so that the measure ranges from 0 to 1
```

```
  minMaxDist <- calcDistTable[, .(minDist = min(measure), maxDist =␣
 ↪max(measure)),
by = c("Store1", "YEARMONTH")]
  distTable <- merge(calcDistTable, minMaxDist, by = c("Store1", "YEARMONTH"))
  distTable[, magnitudeMeasure := 1 - (measure - minDist)/(maxDist - minDist)]
  finalDistTable <- distTable[, .(mag_measure = mean(magnitudeMeasure)), by =
.(Store1, Store2)]
  return(finalDistTable)
}
```

Now let's use the functions to find the control stores! We'll select control stores based on how similar monthly total sales in dollar amounts and monthly number of customers are to the trial stores. So we will need to use our functions to get four scores, two for each of total sales and total customers.

[ ]:
```
trial_store <- 77
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales),␣
 ↪trial_store)
corr_nSales[order(-corr_measure)]
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers),␣
 ↪trial_store)
corr_nCustomers[order(-corr_measure)]
#### Then, use the functions for calculating magnitude.
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures,␣
 ↪quote(totSales),
trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures,
quote(nCustomers), trial_store)
```

A data.table: 260 × 3

Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>
77	77	1.0000000
77	71	0.9141060
77	233	0.9037742
77	119	0.8676644
77	17	0.8426684
77	3	0.8066436
77	41	0.7832319
77	50	0.7638658
77	157	0.7358932
77	162	0.7297401
77	257	0.7249273
77	234	0.6963248
77	115	0.6891588
77	84	0.6843478
77	167	0.6571104
77	265	0.6397594
77	192	0.6250302
77	63	0.5895506
77	254	0.5771085
77	237	0.5752003
77	33	0.5737002
77	60	0.5723209
77	105	0.5575083
77	27	0.5544333
77	202	0.5349696
77	53	0.5327640
77	250	0.5322723
77	184	0.5245841
77	113	0.5200153
77	111	0.5194727
...	...	...
77	240	-0.4115985
77	225	-0.4170938
77	89	-0.4289356
77	114	-0.4396363
77	180	-0.4447771
77	67	-0.4452270
77	61	-0.4505453
77	129	-0.4529198
77	227	-0.4574275
77	179	-0.4694154
77	267	-0.4705143
77	172	-0.4935916
77	22	-0.5005767
77	249	-0.5185074
77	147	-0.5379945
77	266	-0.5397460
77	102	-0.5508337
77	138	-0.5851740
77	169	-0.6301123
77	247	-0.6310496

A data.table: 260 × 3

Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>
77	77	1.0000000
77	233	0.9903578
77	119	0.9832666
77	254	0.9162084
77	113	0.9013480
77	84	0.8585712
77	41	0.8442195
77	3	0.8342074
77	35	0.7746471
77	88	0.7650480
77	26	0.7594519
77	71	0.7548167
77	33	0.7478944
77	17	0.7473078
77	248	0.7328657
77	157	0.7317167
77	115	0.7188818
77	167	0.7179126
77	14	0.7148412
77	162	0.6885772
77	27	0.6863864
77	111	0.6859257
77	145	0.6811551
77	230	0.6581783
77	250	0.6204840
77	237	0.6140960
77	105	0.6128367
77	57	0.6117599
77	50	0.6073908
77	53	0.6026804
...	...	...
77	133	-0.4081595
77	8	-0.4103574
77	38	-0.4112805
77	215	-0.4140223
77	24	-0.4252580
77	25	-0.4416044
77	16	-0.4636965
77	269	-0.4742925
77	220	-0.5018435
77	179	-0.5058964
77	138	-0.5348775
77	208	-0.5439011
77	258	-0.5457131
77	67	-0.5480536
77	2	-0.5720509
77	266	-0.5756655
77	124	-0.5859688
77	75	-0.5907354
77	163	-0.6044748
77	247	-0.6210342

We'll need to combine the all the scores calculated using our function to create a composite score to rank on. Let's take a simple average of the correlation and magnitude scores for each driver. Note that if we consider it more important for the trend of the drivers to be similar, we can increase the weight of the correlation score (a simple average gives a weight of 0.5 to the corr_weight) or if we consider the absolute size of the drivers to be more important, we can lower the weight of the correlation score.

[ ]:
```
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by =
          c("Store1","Store2"))[, scoreNSales := (corr_measure + mag_measure)/2␣
 ↪]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =
        c("Store1", "Store2"))[, scoreNCust := (corr_measure + mag_measure)/2]
```

[ ]:
```
score_nSales[order(-scoreNSales)]
```

	Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>	mag_measure <dbl>	scoreNSales <dbl>
	77	77	1.0000000	1.0000000	1.0000000
	77	233	0.9037742	0.9852649	0.9445195
	77	41	0.7832319	0.9651401	0.8741860
	77	50	0.7638658	0.9731293	0.8684976
	77	17	0.8426684	0.8806882	0.8616783
	77	115	0.6891588	0.9328321	0.8109955
	77	167	0.6571104	0.9591332	0.8081218
	77	265	0.6397594	0.9626629	0.8012111
	77	234	0.6963248	0.8903392	0.7933320
	77	84	0.6843478	0.8300852	0.7572165
	77	53	0.5327640	0.9754223	0.7540932
	77	254	0.5771085	0.9227714	0.7499399
	77	111	0.5194727	0.9654657	0.7424692
	77	20	0.5173500	0.9637886	0.7405693
	77	192	0.6250302	0.8394839	0.7322570
	77	121	0.5085674	0.9437334	0.7261504
	77	187	0.4606688	0.9705301	0.7155995
	77	35	0.5018257	0.9105502	0.7061880
	77	46	0.4356501	0.9747818	0.7052159
	77	27	0.5544333	0.8549272	0.7046802
	77	131	0.4032990	0.9749592	0.6891291
	77	71	0.9141060	0.4537816	0.6839438
	77	145	0.4306424	0.9368768	0.6837596
	77	264	0.3938364	0.9664660	0.6801512
	77	151	0.4113130	0.9285680	0.6699405
	77	195	0.3780449	0.9591473	0.6685961
	77	176	0.3691206	0.9643896	0.6667551
	77	202	0.5349696	0.7973091	0.6661394
	77	205	0.3565101	0.9723667	0.6644384
A data.table: 260 × 5	77	268	0.3447571	0.9607852	0.6527712
	...	...	...	...	...
	77	49	-0.2373630	0.3546084	0.058622684
	77	242	-0.6926643	0.8097966	0.058566134
	77	186	-0.8202139	0.9371006	0.058443366
	77	104	-0.4089696	0.5214802	0.056255304
	77	129	-0.4529198	0.5521541	0.049617183
	77	225	-0.4170938	0.5127414	0.047823786
	77	97	-0.3989686	0.4929226	0.046976960
	77	67	-0.4452270	0.5228072	0.038790097
	77	125	-0.2330665	0.3009422	0.033937884
	77	244	-0.7745129	0.8319823	0.028734697
	77	180	-0.4447771	0.4958254	0.025524154
	77	261	-0.2465929	0.2807964	0.017101757
	77	88	-0.1141987	0.1476075	0.016704401
	77	227	-0.4574275	0.4873566	0.014964556
	77	147	-0.5379945	0.5600254	0.011015411
	77	172	-0.4935916	0.5133446	0.009876494
	77	114	-0.4396363	0.4577496	0.009056645
	77	80	-0.4054377	0.4169541	0.005758211
	77	179	-0.4694154	0.4439293	-0.012743068
	77	95	-0.3325497	0.2965540	-0.017997858

[ ]:
```
score_nCustomers[order(-scoreNCust)]
```

Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>	mag_measure <dbl>	scoreNCust <dbl>
77	77	1.0000000	1.0000000	1.0000000
77	233	0.9903578	0.9927733	0.9915655
77	254	0.9162084	0.9371312	0.9266698
77	41	0.8442195	0.9746392	0.9094294
77	84	0.8585712	0.9241818	0.8913765
77	17	0.7473078	0.9624953	0.8549015
77	115	0.7188818	0.9659160	0.8423989
77	35	0.7746471	0.9069267	0.8407869
77	167	0.7179126	0.9493491	0.8336309
77	111	0.6859257	0.9660641	0.8259949
77	145	0.6811551	0.9542009	0.8176780
77	27	0.6863864	0.9488624	0.8176244
77	248	0.7328657	0.8789659	0.8059158
77	53	0.6026804	0.9577633	0.7802218
77	50	0.6073908	0.9250762	0.7662335
77	234	0.5823555	0.9447935	0.7635745
77	265	0.5734604	0.9479016	0.7606810
77	46	0.5507395	0.9607565	0.7557480
77	37	0.4940162	0.9237633	0.7088898
77	119	0.9832666	0.4266213	0.7049440
77	195	0.4708840	0.9257580	0.6983210
77	187	0.4721939	0.9100329	0.6911134
77	264	0.4312937	0.9399269	0.6856103
77	64	0.4017273	0.9507338	0.6762306
77	20	0.4092864	0.9370057	0.6731461
77	121	0.4222581	0.9201625	0.6712103
77	14	0.7148412	0.6265819	0.6707116
77	113	0.9013480	0.4197212	0.6605346
77	268	0.3695170	0.9399068	0.6547119
77	202	0.3738307	0.8966082	0.6352195
...	...	...	...	...
77	49	-0.2318959	0.3730064	7.055525e-02
77	116	-0.3058060	0.4394444	6.681924e-02
77	54	-0.7606047	0.8921440	6.576965e-02
77	155	-0.3156804	0.4164494	5.038450e-02
77	24	-0.4252580	0.5167814	4.576168e-02
77	95	-0.1877498	0.2767255	4.448788e-02
77	9	-0.7856990	0.8508400	3.257048e-02
77	172	-0.3965076	0.4528275	2.815993e-02
77	258	-0.5457131	0.5981580	2.622244e-02
77	165	-0.1265906	0.1738444	2.362692e-02
77	7	-0.3109767	0.3562471	2.263521e-02
77	80	-0.3299840	0.3681868	1.910139e-02
77	231	-0.2824438	0.3154430	1.649961e-02
77	208	-0.5439011	0.5437112	-9.495985e-05
77	114	-0.3530716	0.3503737	-1.348945e-03
77	125	-0.2829964	0.2746051	-4.195656e-03
77	55	-0.3954735	0.3797372	-7.868115e-03
77	267	-0.6487111	0.6155288	-1.659114e-02
77	19	-0.6334531	0.5322675	-5.059285e-02
77	4	-0.2956387	0.1895787	-5.303001e-02

A data.table: 260 × 5

Now we have a score for each of total number of sales and number of customers. Let's combine the two via a simple average.

[ ]:
```
score_Control <- merge(score_nSales, score_nCustomers, by =␣
↪c("Store1","Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
```

[ ]:
```
score_Control[order(-finalControlScore)]
```

A data.table: 260 × 9

Store1 <dbl>	Store2 <dbl>	corr_measure.x <dbl>	mag_measure.x <dbl>	scoreNSales <dbl>	corr_measure.y <dbl>	m <dbl>
77	77	1.0000000	1.0000000	1.0000000	1.0000000	1.
77	233	0.9037742	0.9852649	0.9445195	0.9903578	0.
77	41	0.7832319	0.9651401	0.8741860	0.8442195	0.
77	17	0.8426684	0.8806882	0.8616783	0.7473078	0.
77	254	0.5771085	0.9227714	0.7499399	0.9162084	0.
77	115	0.6891588	0.9328321	0.8109955	0.7188818	0.
77	84	0.6843478	0.8300852	0.7572165	0.8585712	0.
77	167	0.6571104	0.9591332	0.8081218	0.7179126	0.
77	50	0.7638658	0.9731293	0.8684976	0.6073908	0.
77	111	0.5194727	0.9654657	0.7424692	0.6859257	0.
77	265	0.6397594	0.9626629	0.8012111	0.5734604	0.
77	234	0.6963248	0.8903392	0.7933320	0.5823555	0.
77	35	0.5018257	0.9105502	0.7061880	0.7746471	0.
77	53	0.5327640	0.9754223	0.7540932	0.6026804	0.
77	27	0.5544333	0.8549272	0.7046802	0.6863864	0.
77	145	0.4306424	0.9368768	0.6837596	0.6811551	0.
77	46	0.4356501	0.9747818	0.7052159	0.5507395	0.
77	248	0.3661524	0.8885953	0.6273739	0.7328657	0.
77	20	0.5173500	0.9637886	0.7405693	0.4092864	0.
77	187	0.4606688	0.9705301	0.7155995	0.4721939	0.
77	121	0.5085674	0.9437334	0.7261504	0.4222581	0.
77	195	0.3780449	0.9591473	0.6685961	0.4708840	0.
77	264	0.3938364	0.9664660	0.6801512	0.4312937	0.
77	37	0.4375957	0.8650393	0.6513175	0.4940162	0.
77	119	0.8676644	0.4341303	0.6508974	0.9832666	0.
77	14	0.4355935	0.8432875	0.6394405	0.7148412	0.
77	268	0.3447571	0.9607852	0.6527712	0.3695170	0.
77	202	0.5349696	0.7973091	0.6661394	0.3738307	0.
77	205	0.3565101	0.9723667	0.6644384	0.2726114	0.
77	151	0.4113130	0.9285680	0.6699405	0.2674333	0.
...	...	...	...	...	...	...
77	267	-0.4705143	0.8293633	0.179424479	-0.64871112	0.
77	155	-0.3143641	0.4900639	0.087849875	-0.31568036	0.
77	9	-0.7029760	0.9097629	0.103393443	-0.78569901	0.
77	180	-0.4447771	0.4958254	0.025524154	-0.18164870	0.
77	244	-0.7745129	0.8319823	0.028734697	-0.39904387	0.
77	49	-0.2373630	0.3546084	0.058622684	-0.23189585	0.
77	201	-0.4109081	0.2809523	-0.064977859	0.13053712	0.
77	238	-0.3241948	0.2734532	-0.025370788	0.03729658	0.
77	258	-0.6508930	0.8322403	0.090673662	-0.54571312	0.
77	208	-0.3811432	0.6113168	0.115086826	-0.54390114	0.
77	7	-0.1938859	0.3744116	0.090262830	-0.31097668	0.
77	231	-0.1786914	0.3381136	0.079711083	-0.28244377	0.
77	172	-0.4935916	0.5133446	0.009876494	-0.39650765	0.
77	125	-0.2330665	0.3009422	0.033937884	-0.28299641	0.
77	95	-0.3325497	0.2965540	-0.017997858	-0.18774978	0.
77	80	-0.4054377	0.4169541	0.005758211	-0.32998404	0.
77	114	-0.4396363	0.4577496	0.009056645	-0.35307158	0.
77	269	-0.3157303	0.4521340	0.068201838	-0.47429252	0.
77	24	-0.7181123	0.5908516	-0.063630353	-0.42525802	0.
77	67	-0.4452270	0.5228072	0.038790097	-0.54805357	0.

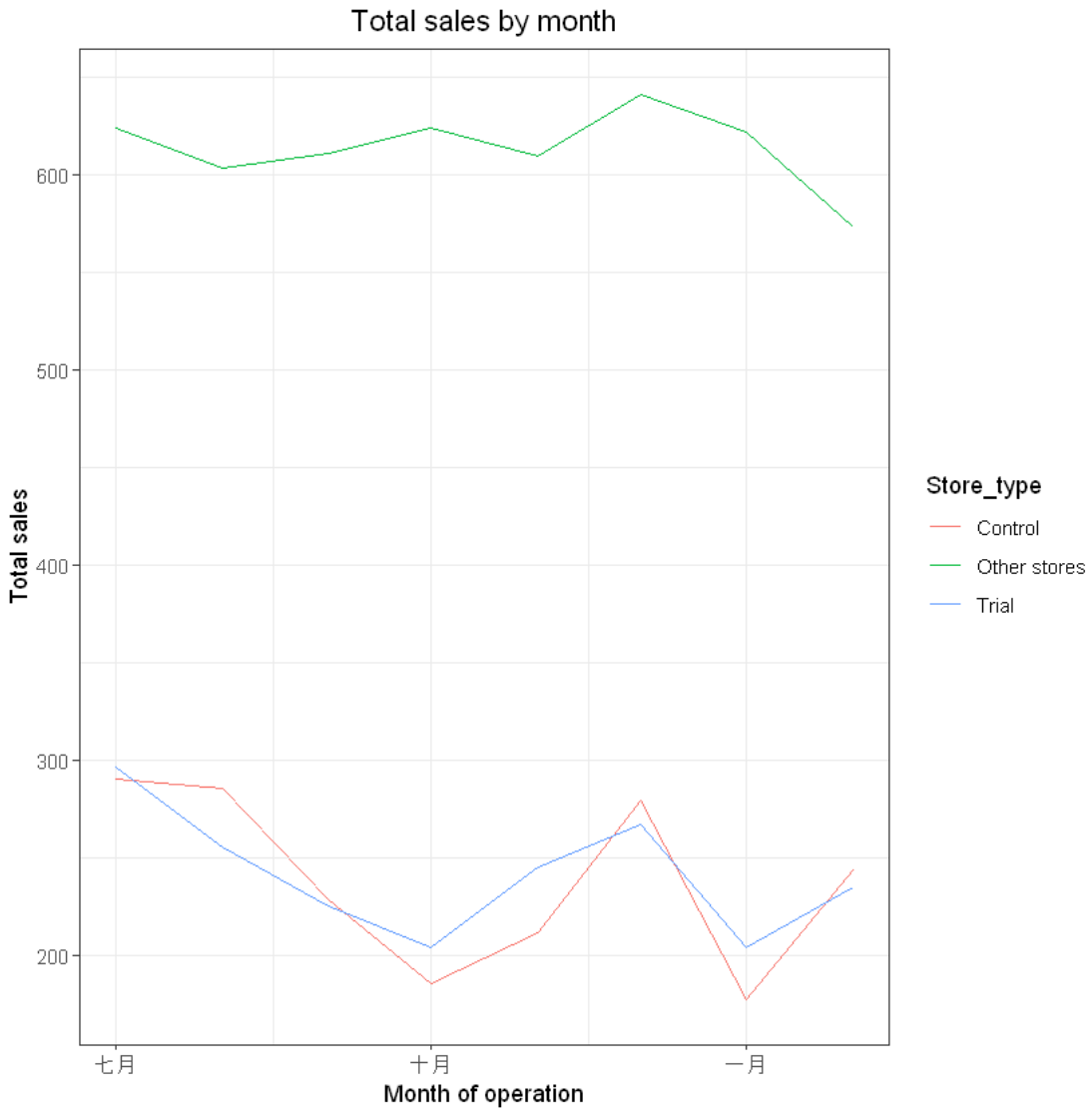
The store with the highest score is then selected as the control store since it is most similar to the trial store.

[ ]:
```
control_store <- score_Control[Store1 == trial_store,␣
↪][order(-finalControlScore)][2, Store2]
control_store
```

233

Now that we have found a control store, let's check visually if the drivers are indeed similar in the period before the trial. We'll look at total sales first.

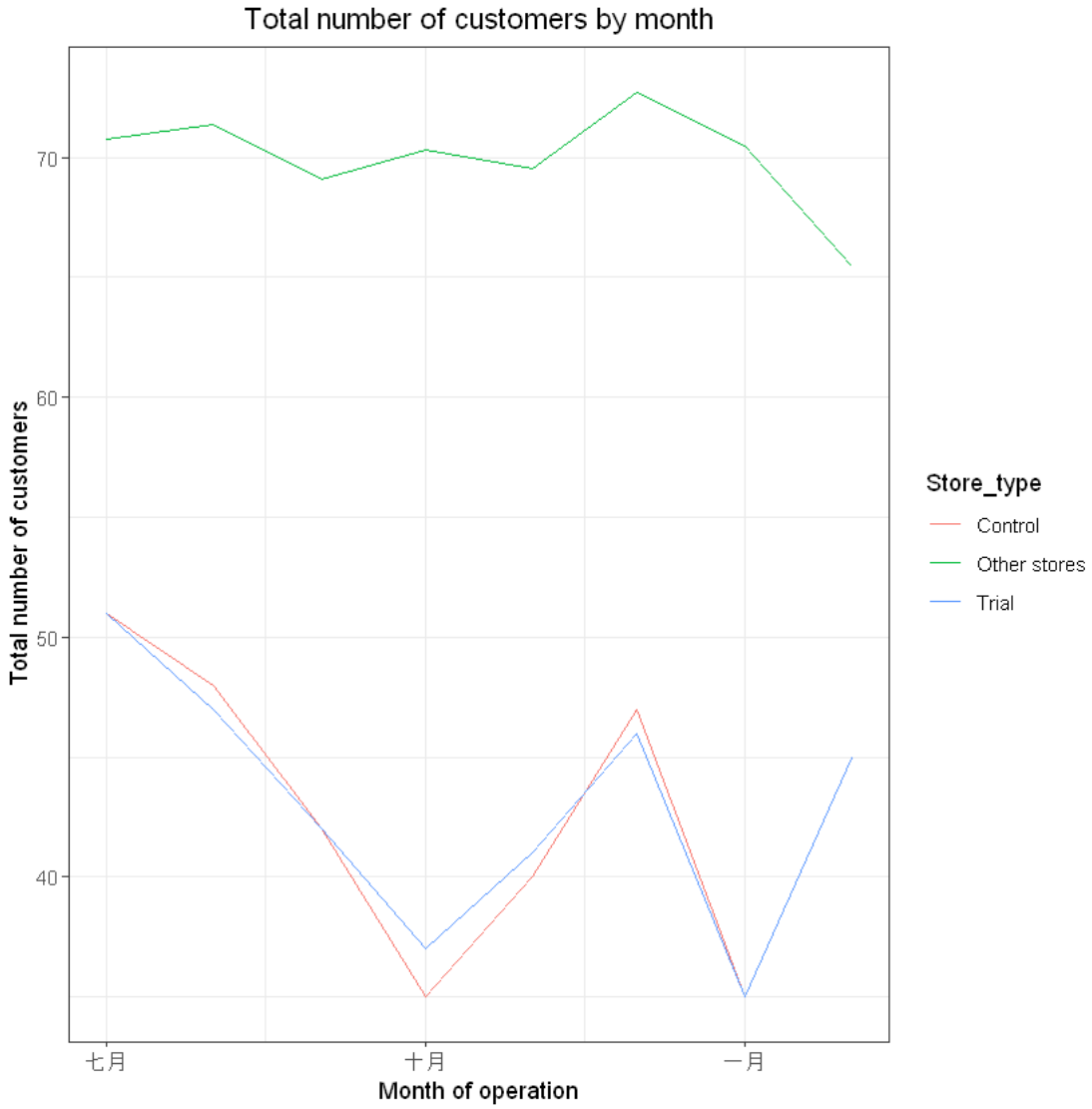
[ ]:
```
measureOverTimeSales <- measureOverTime

pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==␣
↪trial_store,
"Trial",
                                                ifelse(STORE_NBR == control_store,
"Control", "Other stores"))
                                ][, totSales := mean(totSales), by =␣
↪c("YEARMONTH",
"Store_type")
                                ][, TransactionMonth := as.Date(paste(YEARMONTH %/
↪%
100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
                                ][YEARMONTH < 201903 , ]
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by␣
↪month")
```



Next, number of customers.

[ ]:
```
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR ==␣
 ↪trial_store, "Trial",
                                        ifelse(STORE_NBR == control_store,␣
 ↪"Control", "Other stores"))
][, numberCustomers := mean(nCustomers), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,␣
 ↪sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201903 , ]
ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color =␣
 ↪Store_type)) +
```

```
  geom_line() +
  labs(x = "Month of operation", y = "Total number of customers", title =␣
↪"Total number of customers by month")
```



1.3 Assessment of trial

The trial period goes from the start of February 2019 to April 2019. We now want to see if there has been an uplift in overall chip sales. We'll start with scaling the control store's sales to a level similar to control for any differences between the two stores outside of the trial period.

[ ]:
```
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR == control_store &
YEARMONTH < 201902, sum(totSales)]
```

```
#### Apply the scaling factor
measureOverTimeSales <- measureOverTime
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,
controlSales := totSales * scalingFactorForControlSales]
```

Now that we have comparable sales figures for the control store, we can calculate the percentage difference between the scaled control sales and the trial store's sales during the trial period.

[ ]:
```
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
                        measureOverTime[STORE_NBR == trial_store, c("totSales",␣
 ↪"YEARMONTH")],
                        by = "YEARMONTH")[, percentageDiff :=␣
 ↪abs(controlSales-totSales)/controlSales]
```

[ ]:
```
percentageDiff # between control store sales and trial store sales
```

Let's see if the difference is significant!

[ ]:
```
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
```

[ ]:
```
degreesOfFreedom <- 7
```

[ ]:
```
percentageDiff[, tValue := (percentageDiff - 0)/stdDev
               ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,␣
 ↪YEARMONTH %% 100, 1,
                                                     sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth,tValue)]
```

A data.table: 3 × 2

TransactionMonth <date>	tValue <dbl>
2019-02-01	1.183534
2019-03-01	7.339116
2019-04-01	12.476373

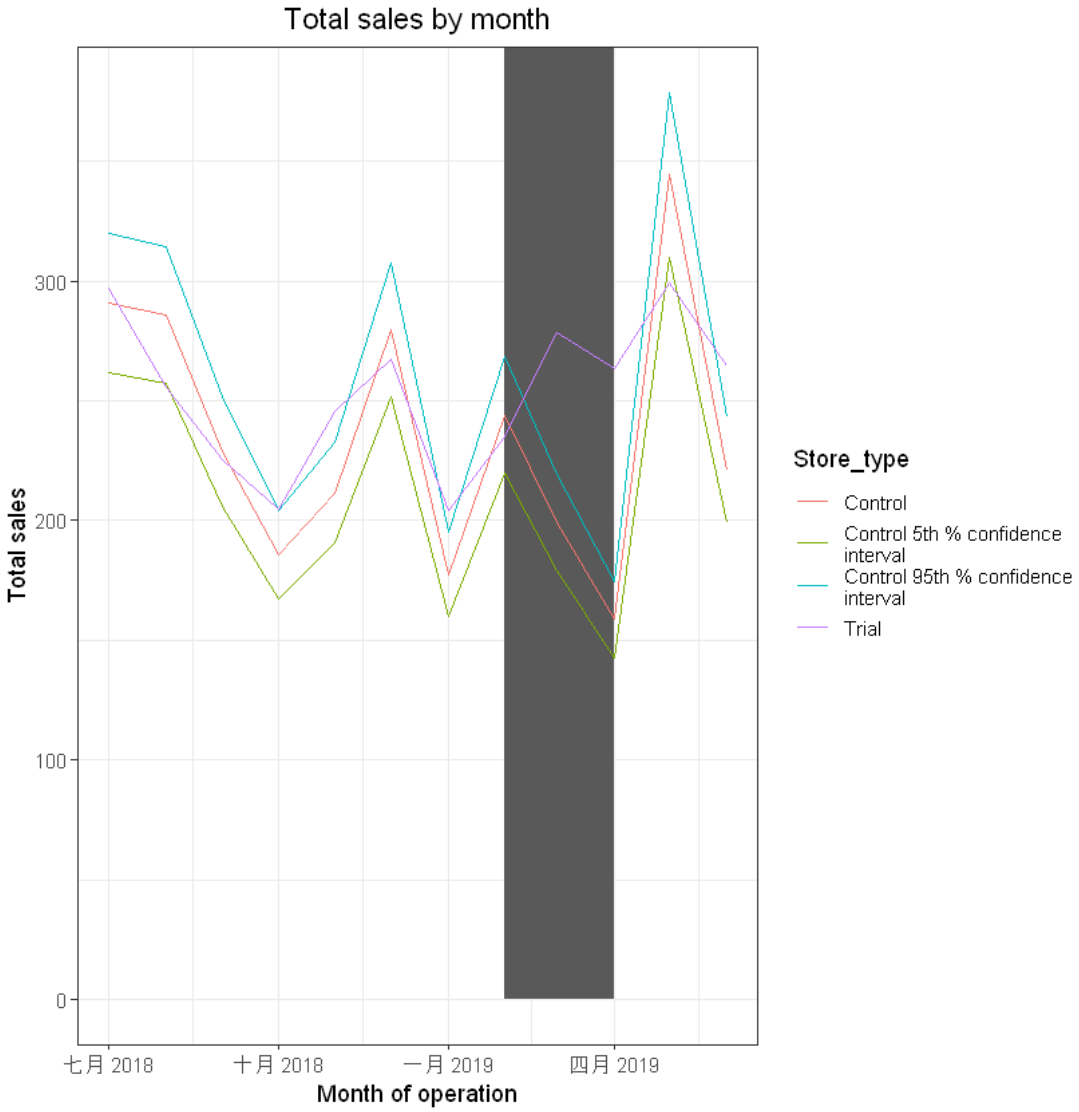
[ ]:
```
qt(0.95, df = degreesOfFreedom)
```

1.89457860509001

We can observe that the t-value is much larger than the 95th percentile value of the t-distribution for March and April - i.e. the increase in sales in the trial store in March and April is statistically greater than in the control store. Let's create a more visual version of this by plotting the sales of the control store, the sales of the trial stores and the 95th percentile value of sales of the control store.

[ ]:
```
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==␣
 ↪trial_store, "Trial",
ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
```

```
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,␣
 ↪sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ]

#### Control store 95th percentile
pastSales_Controls95 <- pastSales[Store_type == "Control",
 ][, totSales := totSales * (1 + stdDev * 2)
 ][, Store_type := "Control 95th % confidence
interval"]

#### Control store 5th percentile
pastSales_Controls5 <- pastSales[Store_type == "Control",
 ][, totSales := totSales * (1 - stdDev * 2)
 ][, Store_type := "Control 5th % confidence
interval"]
trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)

#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 , ymax␣
 ↪=
Inf, color = NULL), show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by␣
 ↪month")
```



The results show that the trial in store 77 is significantly different to its control store in the trial period as the trial store performance lies outside the 5% to 95% confidence interval of the control store in two of the three trial months. Let's have a look at assessing this for number of customers as well.

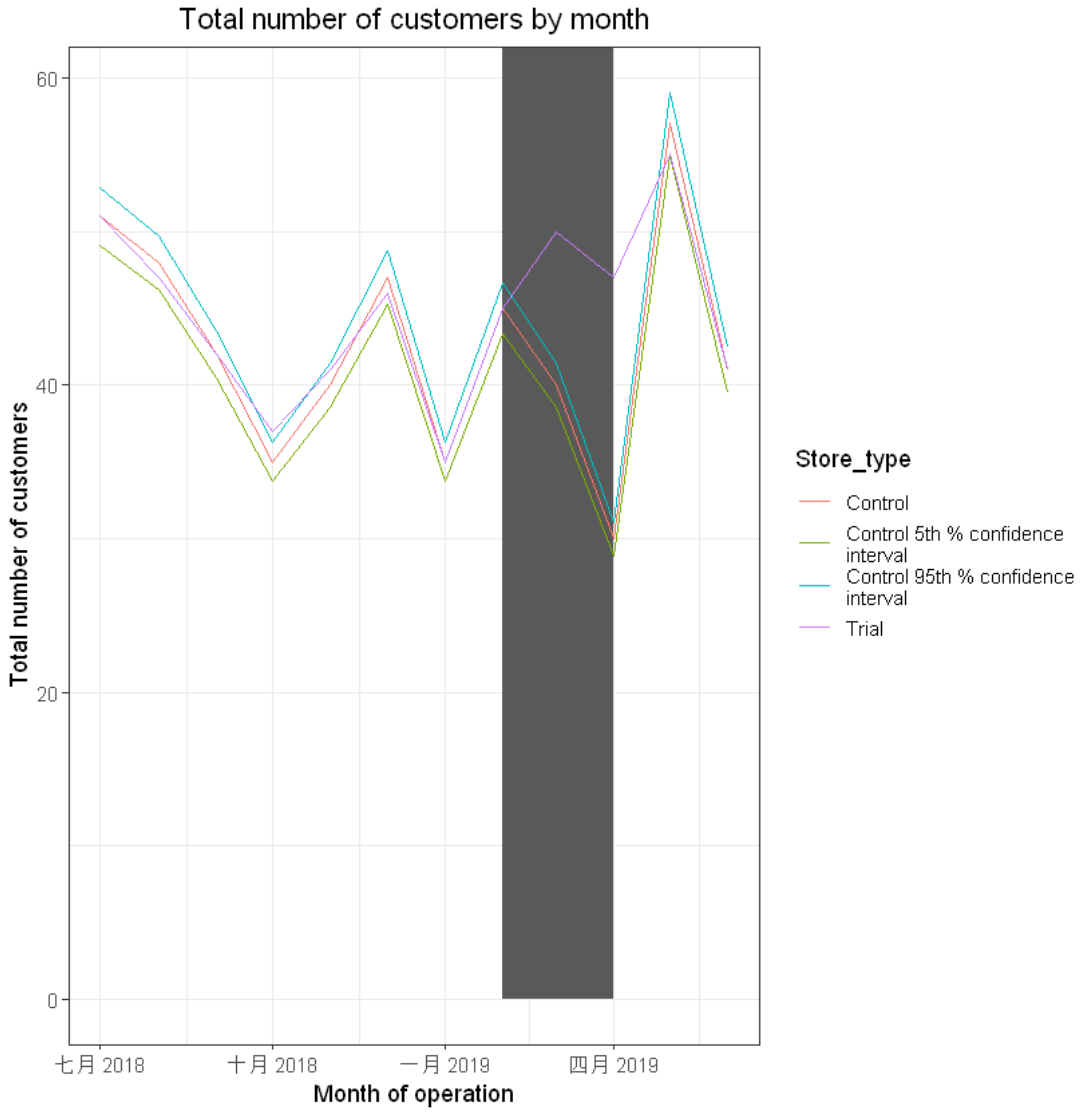
[ ]:
```
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(nCustomers)] / preTrialMeasures[STORE_NBR ==
control_store & YEARMONTH < 201902, sum(nCustomers)]
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
][ , controlCustomers := nCustomers * scalingFactorForControlCust
][, Store_type := ifelse(STORE_NBR ==trial_store, "Trial",
ifelse(STORE_NBR == control_store,"Control", "Other stores"))]
```

```
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH",␣
 ↪"controlCustomers")],
measureOverTimeCusts[STORE_NBR == trial_store,c("nCustomers", "YEARMONTH")],
by = "YEARMONTH"
)[, percentageDiff := abs(controlCustomers-nCustomers)/controlCustomers]
```

Let's again see if the difference is significant visually!

[ ]:
```
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
```

[ ]:
```
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by =
c("YEARMONTH", "Store_type")
                              ][Store_type %in% c("Trial", "Control"), ]
#### Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
                                  ][, nCusts := nCusts * (1 + stdDev * 2)
                                  ][, Store_type := "Control 95th % confidence
interval"]
#### Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
                                  ][, nCusts := nCusts * (1 - stdDev * 2)
                                  ][, Store_type := "Control 5th % confidence
interval"]
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
pastCustomers_Controls5)
#### Plot everything into one nice graph.
#### geom_rect creates a rectangle in the plot. Use this to highlight the
#### trial period in our graph.
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 ,
ymax = Inf, color = NULL), show.legend = FALSE) +
  geom_line() + labs(x = "Month of operation", y = "Total number of customers",␣
 ↪title = "Total number of customers by month")
```



Let's repeat finding the control store and assessing the impact of the trial for each of the other two trial stores. ## Trial store 86

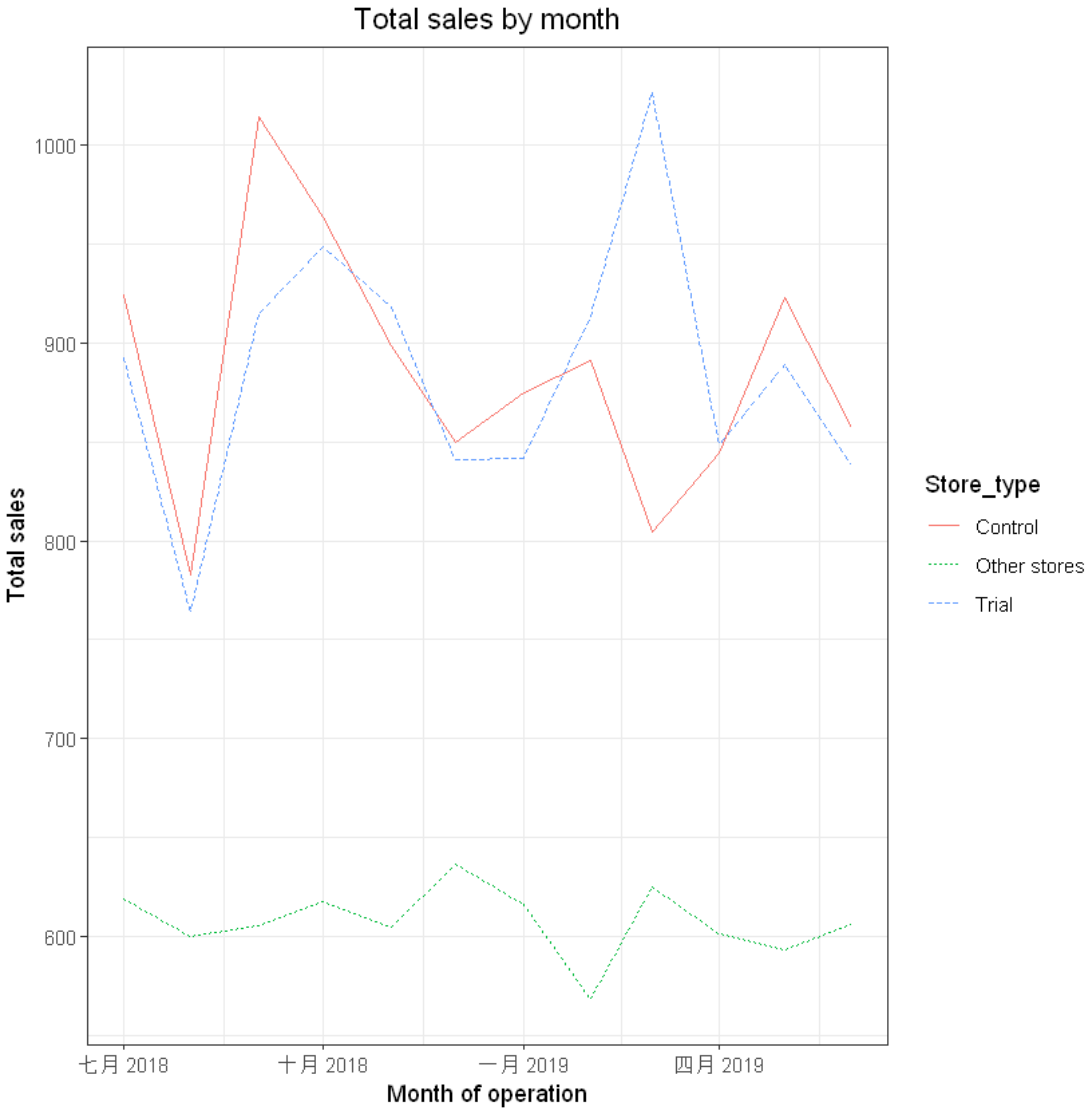
[ ]:
```
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = (uniqueN(TXN_ID))/
 ↪(uniqueN(LYLTY_CARD_NBR)),
                            nChipsPerTxn = (sum(PROD_QTY))/(uniqueN(TXN_ID)) ,
                            avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY) ) ,␣
 ↪by = c("STORE_NBR", "YEARMONTH")][order(STORE_NBR, YEARMONTH)]

#### Use the functions we created earlier to calculate correlations and␣
 ↪magnitude for each potential control store
```

```
trial_store <- 86
corr_nSales <- calculateCorrelation(preTrialMeasures,␣
 ↪quote(totSales),trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers),␣
 ↪trial_store)
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures,␣
 ↪quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures,␣
 ↪quote(nCustomers), trial_store)
#### Now, create a combined score composed of correlation and magnitude
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1",␣
 ↪"Store2"))[ , scoreNSales := (corr_measure + mag_measure)/2]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =␣
 ↪c("Store1", "Store2"))[ , scoreNCust := (corr_measure + mag_measure)/2]

#### Finally, combine scores across the drivers using a simple average.
score_Control <- merge(score_nSales, score_nCustomers, by =␣
 ↪c("Store1","Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
#### Select control stores based on the highest matching store
#### (closest to 1 but not the store itself, i.e. the second ranked highest␣
 ↪store)
#### Select control store for trial store 86
control_store <- score_Control[Store1 == trial_store,
][order(-finalControlScore)][2, Store2]
control_store
```

155

Looks like store 155 will be a control store for trial store 86. Again, let's check visually if the drivers are indeed similar in the period before the trial. We'll look at total sales first.

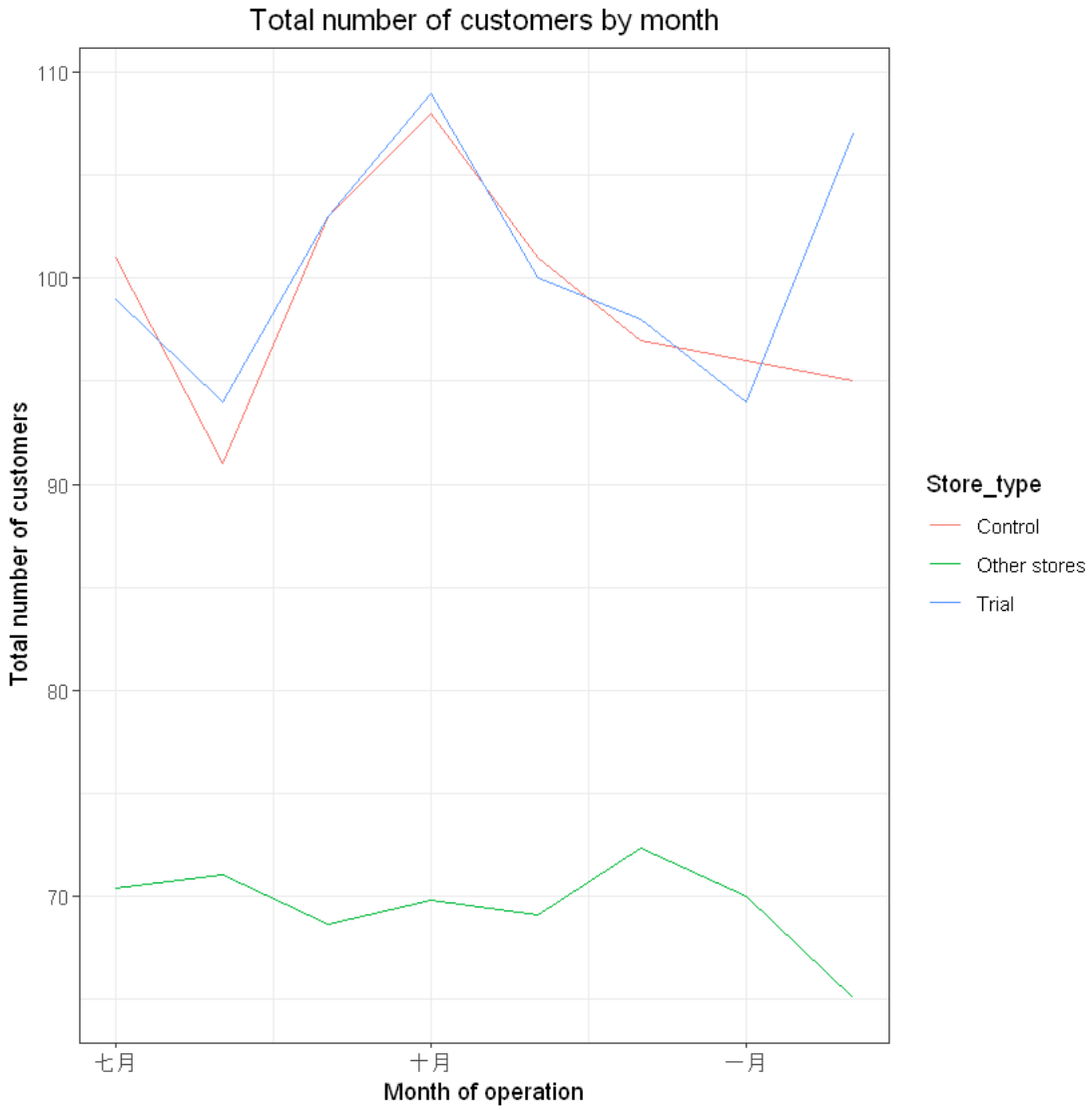
[ ]:
```
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type:= ifelse(STORE_NBR ==␣
 ↪trial_store, "Trial", ifelse(STORE_NBR== control_store, "Control", "Other␣
 ↪stores"))][, totSales := mean(totSales), by = c("YEARMONTH",␣
 ↪"Store_type")][, TransactionMonth:= as.Date(paste(YEARMONTH%/%100,␣
 ↪YEARMONTH%% 100, 1, sep = "-"), "%Y-%m-%d")][YEARMONTH <210903]

ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by␣
 ↪month")
```



Great, sales are trending in a similar way. Next, number of customers.

[ ]:
```
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR ==␣
 ↪trial_store, "Trial",
ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, numberCustomers := mean(nCustomers), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/%
                                      100, YEARMONTH %% 100, 1, sep = "-"),␣
 ↪"%Y-%m-%d")
][YEARMONTH < 201903 , ]
```

```
ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color =␣
↪Store_type)) +
 geom_line() +
 labs(x = "Month of operation", y = "Total number of customers", title =␣
↪"Total number of customers by month")
```

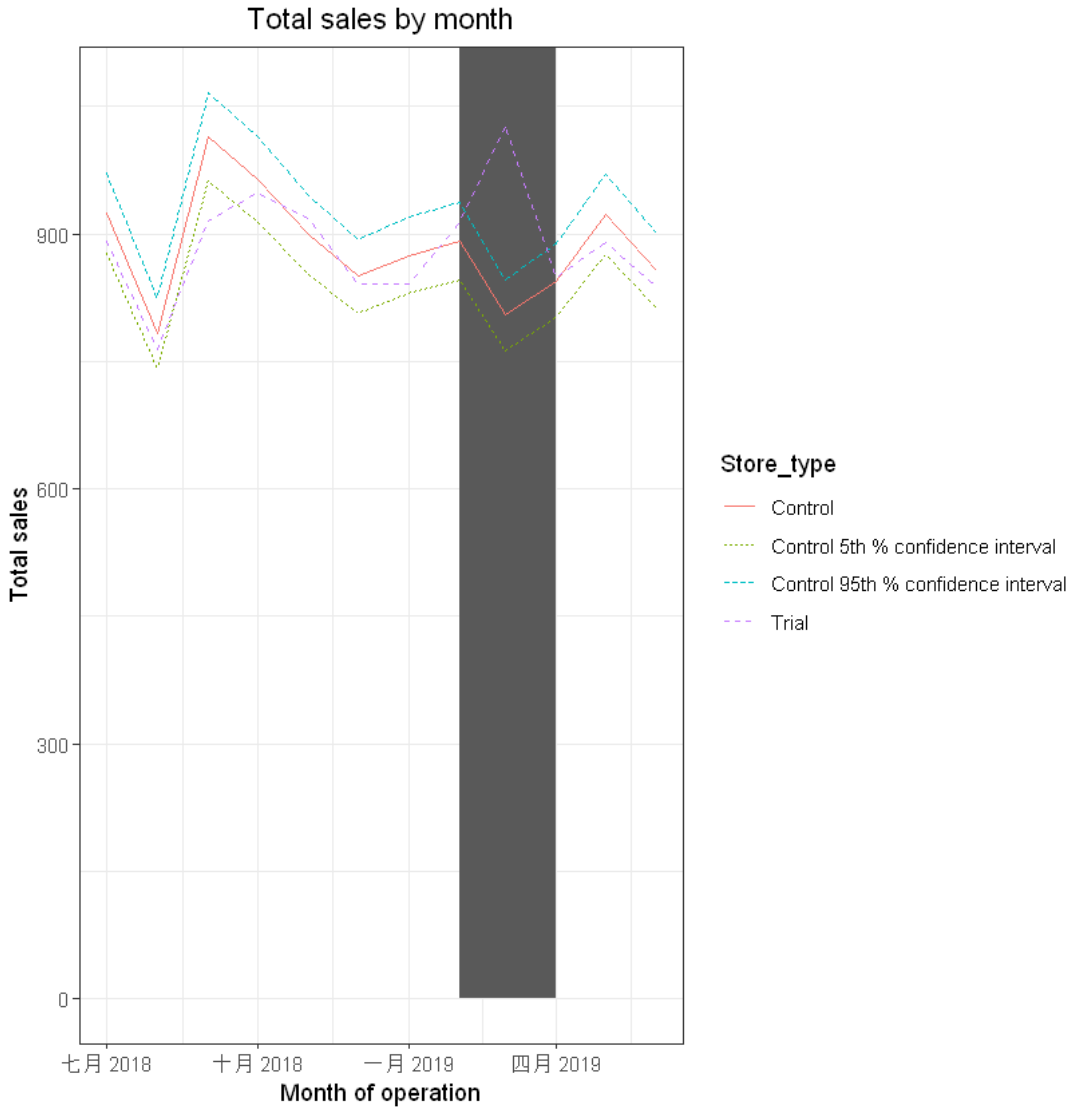


Good, the trend in number of customers is also similar. Let's now assess the impact of the trial on sales.

[ ]:
```
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR == control_store &
YEARMONTH < 201902, sum(totSales)]
#### Apply the scaling factor
```

```
measureOverTimeSales <- measureOverTime
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,
controlSales := totSales * scalingFactorForControlSales]
#### Calculate the percentage difference between scaled control sales and trial␣
 ↪sales
#### When calculating percentage difference, remember to use absolute difference
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
measureOverTime[STORE_NBR == trial_store, c("totSales", "YEARMONTH")],
by = "YEARMONTH"
)[, percentageDiff := abs(controlSales-totSales)/controlSales]

#### As our null hypothesis is that the trial period is the same as the␣
 ↪pre-trial
#### period, let's take the standard deviation based on the scaled percentage␣
 ↪difference
#### in the pre-trial period
#### Calculate the standard deviation of percentage differences during the␣
 ↪pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7

#### Trial and control store total sales
#### Create a table with sales by store type and month.
#### We only need data for the trial and control store.
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==␣
 ↪trial_store, "Trial",
ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/%100, YEARMONTH %% 100, 1,␣
 ↪sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ]

#### Calculate the 5th and 95th percentile for control store sales.
#### The 5th and 95th percentiles can be approximated by using two standard␣
 ↪deviations away from the mean.
#### Recall that the variable stdDev earlier calculates standard deviation in␣
 ↪percentages, and not dollar sales.
#### Control store 95th percentile
pastSales_Controls95 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence interval"]

#### Control store 5th percentile
pastSales_Controls5 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 - stdDev * 2)
```

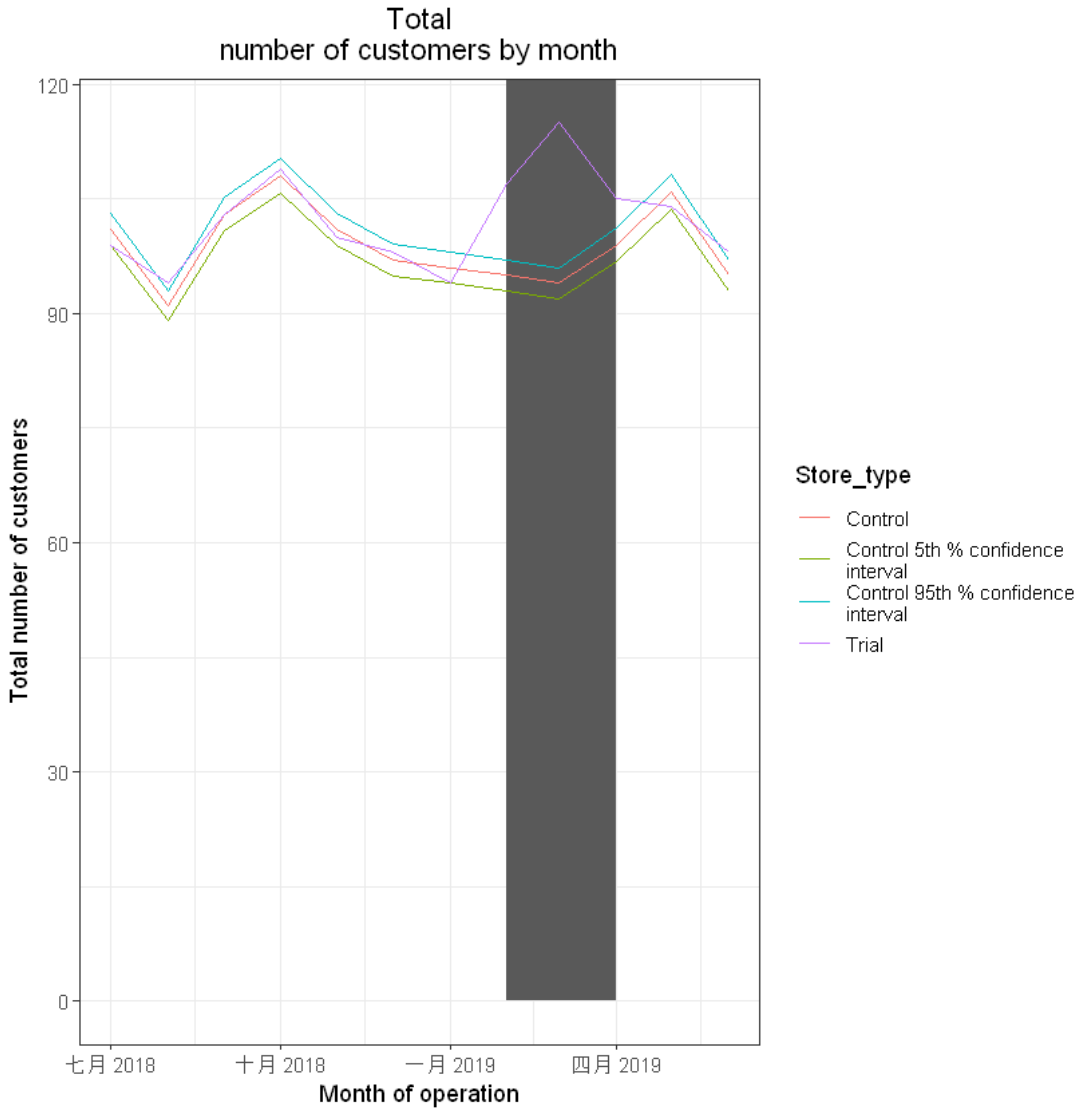
```
][, Store_type := "Control 5th % confidence interval"]

#### Then, create a combined table with columns from pastSales,␣
 ↪pastSales_Controls95 and pastSales_Controls5
trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 , ymax␣
 ↪=
Inf, color = NULL), show.legend = FALSE) +
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by␣
 ↪month")
```



The results show that the trial in store 86 is not significantly different to its control store in the trial period as the trial store performance lies inside the 5% to 95% confidence interval of the control store in two of the three trial months. Let's have a look at assessing this for the number of customers as well.

[ ]:
```
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(nCustomers)]/preTrialMeasures[STORE_NBR ==␣
 ↪control_store &
YEARMONTH < 201902, sum(nCustomers)]
#### Apply the scaling factor
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
                                               ][ , controlCustomers := nCustomers
* scalingFactorForControlCust
                                               ][, Store_type := ifelse(STORE_NBR
== trial_store, "Trial",
                                        ifelse(STORE_NBR == control_store,
"Control", "Other stores"))
]
#### Calculate the percentage difference between scaled control sales and trial␣
 ↪sales
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH",
"controlCustomers")],
                        measureOverTime[STORE_NBR == trial_store,␣
 ↪c("nCustomers",
"YEARMONTH")],
                        by = "YEARMONTH"
                        )[, percentageDiff :=
abs(controlCustomers-nCustomers)/controlCustomers]
#### As our null hypothesis is that the trial period is the same as the␣
 ↪pre-trial
#### period, let's take the standard deviation based on the scaled percentage␣
 ↪difference
#### in the pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by =
c("YEARMONTH", "Store_type")
                                ][Store_type %in% c("Trial", "Control"), ]
#### Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
                                  ][, nCusts := nCusts * (1 + stdDev * 2)
                                  ][, Store_type := "Control 95th % confidence
interval"]
```

```
#### Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
                                         ][, nCusts := nCusts * (1 - stdDev * 2)
                                         ][, Store_type := "Control 5th % confidence
interval"]
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
pastCustomers_Controls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 , ymax␣
  ↪=
Inf, color = NULL), show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of operation", y = "Total number of customers", title = "Total
number of customers by month")
```



It looks like the number of customers is significantly higher in all of the three months. This seems to suggest that the trial had a significant impact on increasing the number of customers in trial store 86 but as we saw, sales were not significantly higher. We should check with the Category Manager if there were special deals in the trial store that were may have resulted in lower prices, impacting the results.

[ ]:
```
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
nCustomers = uniqueN(LYLTY_CARD_NBR),
nTxnPerCust = uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
nChipsPerTxn = sum(PROD_QTY)/uniqueN(TXN_ID),
avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY))
, by = c("STORE_NBR", "YEARMONTH")][order(STORE_NBR, YEARMONTH)]
```

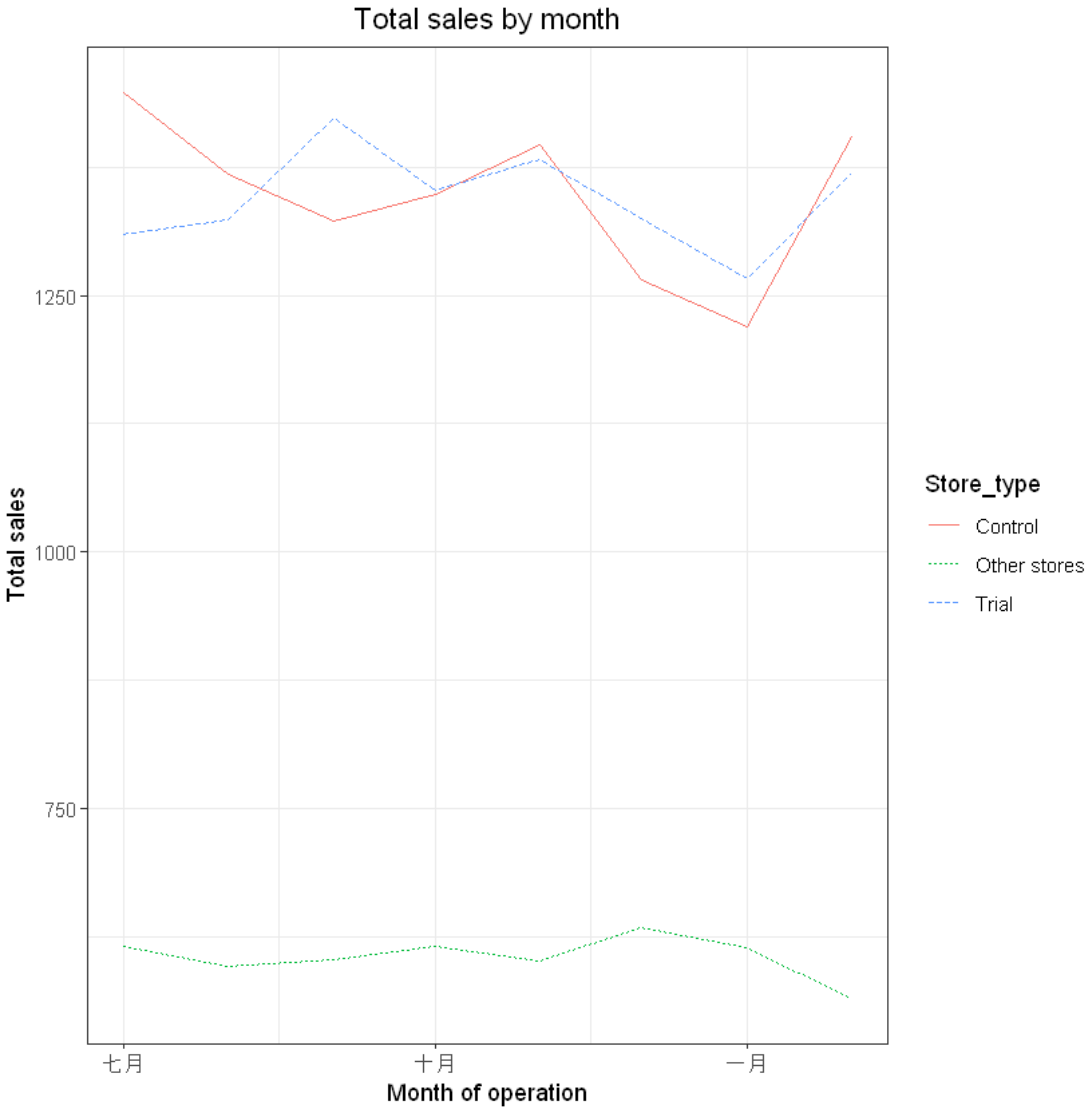
```
#### Use the functions from earlier to calculate the correlation of the sales␣
 ↪and number of customers of each potential control store to the trial store
trial_store <- 88
corr_nSales <- calculateCorrelation(preTrialMeasures,␣
 ↪quote(totSales),trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers),␣
 ↪trial_store)
#### Use the functions from earlier to calculate the magnitude distance of the␣
 ↪sales and number of customers of each potential control store to the trial␣
 ↪store
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures,␣
 ↪quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures,␣
 ↪quote(nCustomers), trial_store)
#### Create a combined score composed of correlation and magnitude by merging␣
 ↪the correlations table and the magnitudes table, for each driver.
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1",␣
 ↪"Store2"))[ , scoreNSales := (corr_measure + mag_measure)/2]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =␣
 ↪c("Store1", "Store2"))[ , scoreNCust := (corr_measure + mag_measure)/2]

#### Combine scores across the drivers by merging sales scores and customer␣
 ↪scores, and compute a final combined score.
score_Control <- merge(score_nSales, score_nCustomers, by =␣
 ↪c("Store1","Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
#### Select control stores based on the highest matching store
#### (closest to 1 but not the store itself, i.e. the second ranked highest␣
 ↪store)
#### Select control store for trial store 88
control_store <- score_Control[Store1 == trial_store,␣
 ↪][order(-finalControlScore)][2, Store2]
control_store
```

237

We've now found store 237 to be a suitable control store for trial store 88. Again, let's check visually if the drivers are indeed similar in the period before the trial. We'll look at total sales first.

[ ]:
```
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==␣
 ↪trial_store, "Trial",
ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, totSales := mean(totSales), by = c("YEARMONTH","Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,␣
 ↪sep = "-"), "%Y-%m-%d")
```

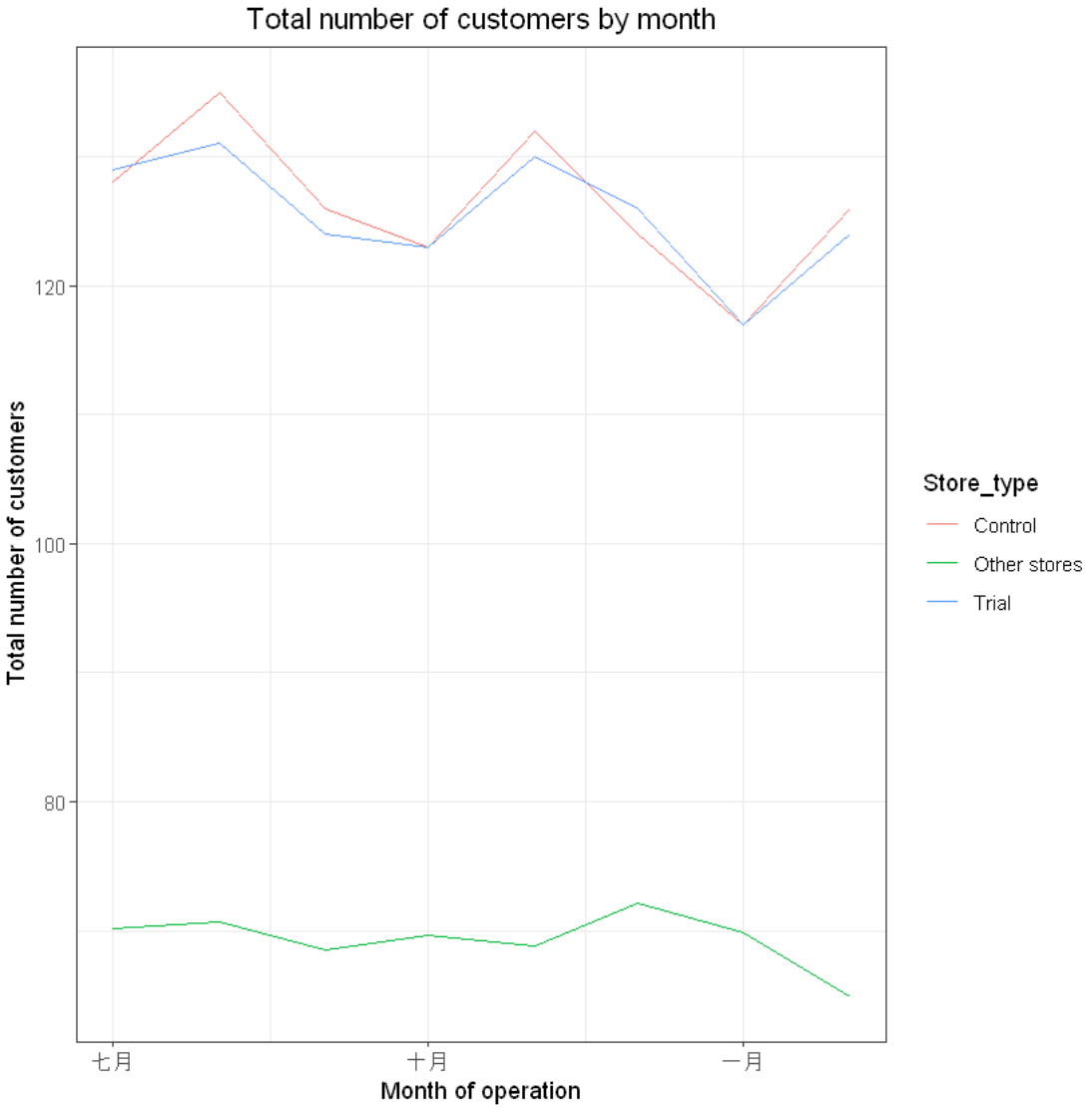
```
][YEARMONTH < 201903 , ]
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
geom_line(aes(linetype = Store_type)) +
labs(x = "Month of operation", y = "Total sales", title = "Total sales by␣
 ↪month")
```



Great, the trial and control stores have similar total sales. Next, number of customers.

[ ]:
```
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR ==␣
 ↪trial_store, "Trial",
ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, numberCustomers := mean(nCustomers), by = c("YEARMONTH", "Store_type")
```

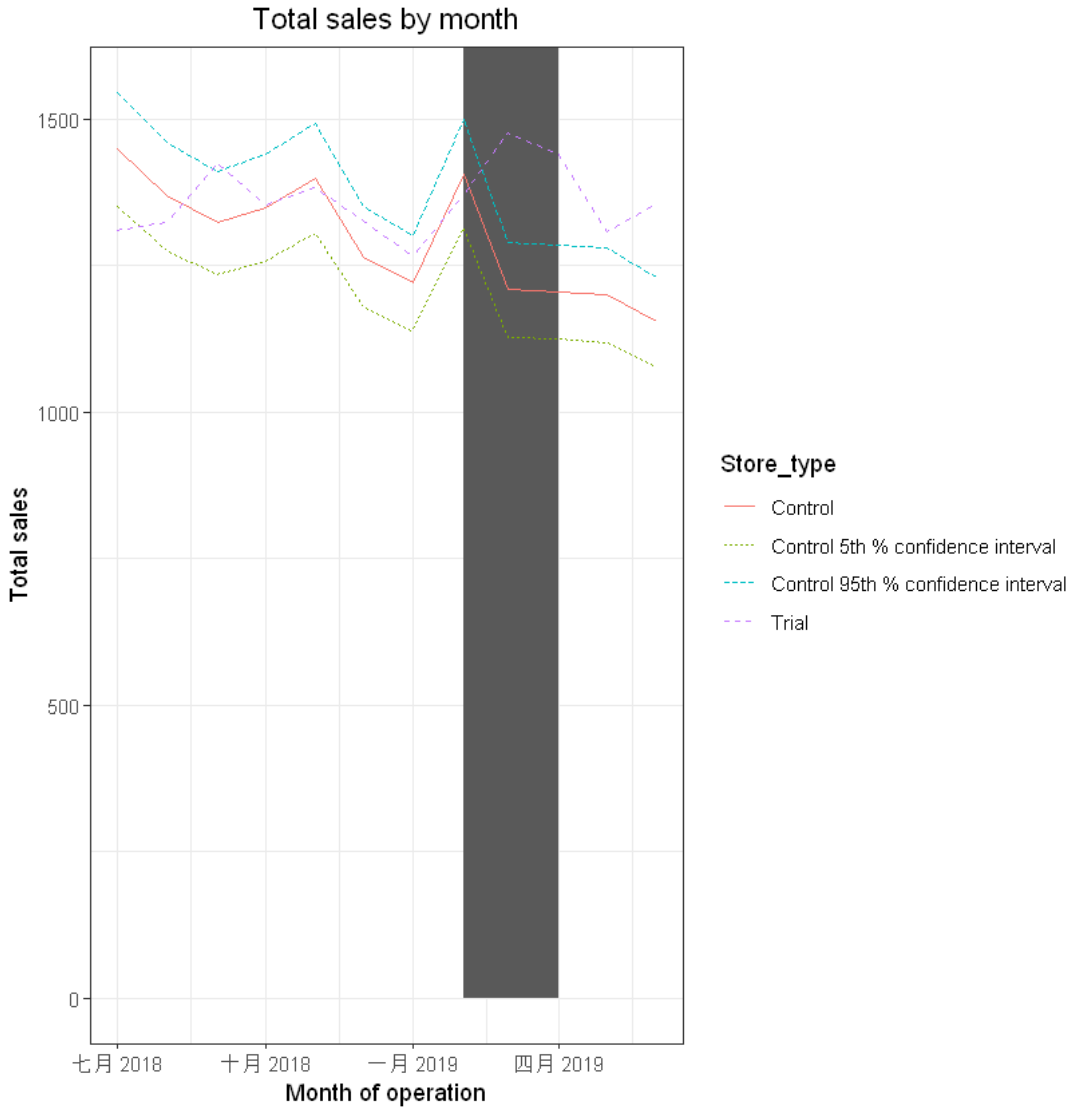
```
][, TransactionMonth := as.Date(paste(YEARMONTH %/%
                                      100, YEARMONTH %% 100, 1, sep = "-"),␣
 ↪"%Y-%m-%d")
][YEARMONTH < 201903 , ]
ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color =␣
 ↪Store_type)) +
  geom_line() + labs(x = "Month of operation", y = "Total number of customers",␣
 ↪title = "Total number of customers by month")
```



Total number of customers of the control and trial stores are also similar. Let's now assess the impact of the trial on sales.

[ ]:
```
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR ==
control_store & YEARMONTH < 201902, sum(totSales)]

#### Apply the scaling factor
measureOverTimeSales <- measureOverTime
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][␣
 ↪,controlSales := totSales * scalingFactorForControlSales]

#### Calculate the absolute percentage difference between scaled control sales␣
 ↪and trial sales
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH",␣
 ↪"controlSales")],measureOverTime[STORE_NBR == trial_store, c("totSales",␣
 ↪"YEARMONTH")],by = "YEARMONTH")[, percentageDiff :=␣
 ↪abs(controlSales-totSales)/controlSales]

#### As our null hypothesis is that the trial period is the same as the␣
 ↪pre-trial period,
#### let's take the standard deviation based on the scaled percentage␣
 ↪difference in the pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store total sales
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==␣
 ↪trial_store, "Trial",
ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/%100, YEARMONTH %% 100, 1,␣
 ↪sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ]
#### Control store 95th percentile
pastSales_Controls95 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence interval"]
#### Control store 5th percentile
pastSales_Controls5 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidence interval"]
trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type)) +
geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 ,
ymax = Inf, color = NULL), show.legend = FALSE) +
  geom_line(aes(linetype = Store_type)) +
```

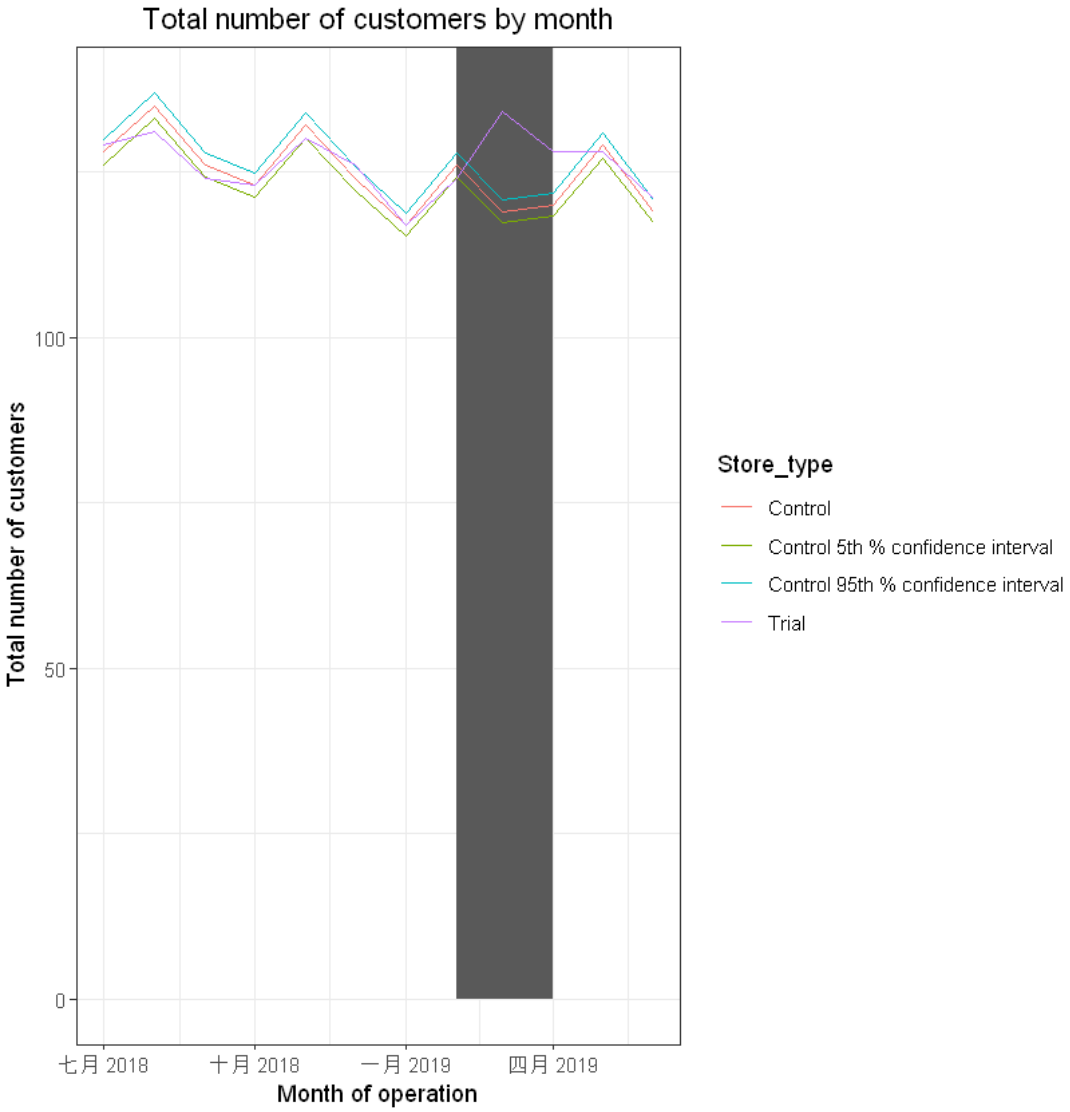
```
 labs(x = "Month of operation", y = "Total sales", title = "Total sales by␣
↪month")
```



The results show that the trial in store 88 is significantly different to its control store in the trial period as the trial store performance lies outside of the 5% to 95% confidence interval of the control store in two of the three trial months. Let's have a look at assessing this for number of customers as well.

[ ]:
```
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(nCustomers)]/preTrialMeasures[STORE_NBR ==
control_store & YEARMONTH < 201902, sum(nCustomers)]

#### Apply the scaling factor
```

```
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
][ , controlCustomers := nCustomers * scalingFactorForControlCust
][, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
ifelse(STORE_NBR == control_store,"Control", "Other stores"))
]
#### Calculate the absolute percentage difference between scaled control sales␣
 ↪and trial sales
percentageDiff <- merge(scaledControlCustomers[,␣
 ↪c("YEARMONTH","controlCustomers")],measureOverTime[STORE_NBR == trial_store,␣
 ↪c("nCustomers", "YEARMONTH")],
by = "YEARMONTH")[, percentageDiff := abs(controlCustomers-nCustomers)/
 ↪controlCustomers]

#### As our null hypothesis is that the trial period is the same as the␣
 ↪pre-trial
#### period, let's take the standard deviation based on the scaled percentage␣
 ↪#### difference in the pre-trial period

stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
# note that there are 8 months in the pre-trial period hence 8 - 1 = 7 degrees␣
 ↪of freedom
#### Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by =␣
 ↪c("YEARMONTH", "Store_type")
][Store_type %in% c("Trial", "Control"), ]

#### Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence interval"]
#### Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidence interval"]
#### Combine the tables pastSales, pastSales_Controls95, pastSales_Controls5
trialAssessment <- rbind(pastCustomers,␣
 ↪pastCustomers_Controls95,pastCustomers_Controls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 ,
ymax = Inf, color = NULL), show.legend = FALSE) + geom_line() +
labs(x = "Month of operation", y = "Total number of customers", title = "Total␣
 ↪number of customers by month")
```



Total number of customers in the trial period for the trial store is significantly higher than the control store for two out of three months, which indicates a positive trial effect. ## Conclusion Good work! We've found control stores 233, 155, 237 for trial stores 77, 86 and 88 respectively. The results for trial stores 77 and 88 during the trial period show a significant difference in at least two of the three trial months but this is not the case for trial store 86. We can check with the client if the implementation of the trial was different in trial store 86 but overall, the trial shows a significant increase in sales. Now that we have finished our analysis, we can prepare our presentation to the Category Manager.