Quantium Virtual Internship - Retail Strategy and Analytics - Task 2

Load required libraries and datasets before.

assign the data files to data.tables

```
#### Set themes for plots
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
data = read.csv("data_task2.csv")
data=data.table(data)
data
```

X <int>	LYLTY_CARD_... <int>	DATE <chr>	STORE_... <int>	TXN... <int>	PROD_... <int>	PROD_NAME <chr>
1	1000	2018-10-17	1	1	5	Natural Chip Compny SeaSalt175g
2	1002	2018-09-16	1	2	58	Red Rock Deli Chikn&Garlic Aioli 15
3	1003	2019-03-07	1	3	52	Grain Waves Sour Cream&Chives 2
4	1003	2019-03-08	1	4	106	Natural ChipCo Hony Soy Chckn17
5	1004	2018-11-02	1	5	96	WW Original Stacked Chips 160g
6	1005	2018-12-28	1	6	86	Cheetos Puffs 165g
7	1007	2018-12-04	1	7	49	Infuzions SourCream&Herbs Veg St
8	1007	2018-12-05	1	8	10	RRD SR Slow Rst Pork Belly 150g
9	1009	2018-11-20	1	9	20	Doritos Cheese Supreme 330g
10	1010	2018-09-09	1	10	51	Doritos Mexicana 170g

1-10 of 10,000 rows | 1-7 of 14 columns Previous **1** 2 3 4 5 6 ... 1000 Next

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.

monthYear <- format(as.Date(data$DATE),"%Y%m")
data[, YEARMONTH := monthYear]
data$YEARMONTH <- as.numeric(as.character(data$YEARMONTH))
#### Next, we define the measure calculations to use during the analysis.
#For each store and month calculate total sales, number of customers, transactions pe
r customer, chips per customer and the average price per unit.

#measure_over_time

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, YEARMONT
H) ]

#### Filter to the pre-trial period and stores with full observation periods
uniqueStores <- unique(measureOverTime$STORE_NBR)
storesWithFullObs <- numeric()

# Loop through unique stores
for (store in uniqueStores) {
  obsCount <- sum(measureOverTime$STORE_NBR == store)
  if (obsCount == 12) {
    storesWithFullObs <- c(storesWithFullObs, store)
  }
}

preTrialMeasures <- measureOverTime[measureOverTime$YEARMONTH < 201902 &
                                    measureOverTime$STORE_NBR %in% storesWithFullObs,
]
str(preTrialMeasures)
```

```
## Classes 'data.table' and 'data.frame':   1820 obs. of  7 variables:
##  $ STORE_NBR      : int  1 1 1 1 1 1 1 2 2 2 ...
##  $ YEARMONTH      : num  201807 201808 201809 201810 201811 ...
##  $ totSales       : num  207 176 279 188 193 ...
##  $ nCustomers     : int  49 42 59 44 46 42 35 39 39 36 ...
##  $ nTxnPerCust    : num  1.06 1.02 1.05 1.02 1.02 ...
##  $ nChipsPerTxn   : num  1.19 1.26 1.21 1.29 1.21 ...
##  $ avgPricePerUnit: num  3.34 3.26 3.72 3.24 3.38 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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.

```r
#### Let's define inputTable as a metric table with potential comparison stores, metr
icCol as the store metric used to calculate correlation on, and storeComparison as th
e store number of the trial store.

calculateCorrelation <- function(inputTable, metricCol, storeComparison) {
  # Create an empty table to store the results
  calcCorrTable = data.table(Store1 = numeric(), Store2 = numeric(), corr_measure = n
umeric())

  # Get a list of all the store numbers
  storeNumbers <- unique(inputTable[, STORE_NBR])

  # Loop through each store
  for (i in storeNumbers) {
    # Calculate the correlation between the trial store and the current store
    calculatedMeasure = data.table(
      "Store1" = storeComparison,
      "Store2" = i,
      "corr_measure" = cor(
        inputTable[STORE_NBR == storeComparison, eval(metricCol)],
        inputTable[STORE_NBR == i, eval(metricCol)]
      )
    )

    # Add the result to the table
    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.

```r
#### Create a function to calculate a standardised magnitude distance for a measure,
#### looping through each control store
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.

```r
### Use the function you created to calculate correlations against store 77 using tot
al sales and number of customers.
trial_store <- 77
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_st
ore)
#### Then, use the functions for calculating magnitude.
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales),
trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures,
quote(nCustomers), trial_store)
```

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.

```
####Create a combined score composed of correlation and magnitude, by first merging t
he correlations table with the magnitude table.
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales , by = c('Store1',"Store2"))[, sc
oreNSales := (corr_measure+mag_measure)/2]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c('Store1',"Sto
re2"))[, scoreNCust := (corr_measure+mag_measure)/2]
score_nSales[order(-scoreNSales)]
```

Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>	mag_measure <dbl>	scoreNSales <dbl>
77	77	1.000000000	1.00000000	1.000000000
77	233	0.903774188	0.98526489	0.944519541
77	41	0.783231868	0.96514010	0.874185985
77	50	0.763865842	0.97312929	0.868497568
77	17	0.842668360	0.88068824	0.861678301
77	115	0.689158820	0.93283212	0.810995469
77	167	0.657110366	0.95913323	0.808121796
77	265	0.639759375	0.96266286	0.801211116
77	234	0.696324778	0.89033921	0.793331995
77	84	0.684347845	0.83008517	0.757216508

1-10 of 260 rows Previous **1** 2 3 4 5 6 ... 26 Next

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

Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>	mag_measure <dbl>	scoreNCust <dbl>
77	77	1.000000000	1.00000000	1.000000e+00
77	233	0.990357788	0.99277331	9.915655e-01
77	254	0.916208390	0.93713119	9.266698e-01
77	41	0.844219490	0.97463924	9.094294e-01
77	84	0.858571239	0.92418181	8.913765e-01
77	17	0.747307760	0.96249530	8.549015e-01
77	115	0.718881754	0.96591604	8.423989e-01
77	35	0.774647081	0.90692675	8.407869e-01
77	167	0.717912623	0.94934912	8.336309e-01
77	111	0.685925661	0.96606414	8.259949e-01

1-10 of 260 rows Previous **1** 2 3 4 5 6 ... 26 Next

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.

```
#### Combine scores across the drivers by first merging our salesscores and customer scores into a single table
score_Control <- merge(score_nSales, score_nCustomers, by = c('Store1', 'Store2'))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
score_Control[order(-finalControlScore)]
```

Store1 <dbl>	Store2 <dbl>	corr_measure.x <dbl>	mag_measure.x <dbl>	scoreNSales <dbl>	corr_measure.y <dbl>	mag_measure. <dbl
77	77	1.000000000	1.00000000	1.000000000	1.000000000	1.0000000
77	233	0.903774188	0.98526489	0.944519541	0.990357788	0.9927733
77	41	0.783231868	0.96514010	0.874185985	0.844219490	0.9746392
77	17	0.842668360	0.88068824	0.861678301	0.747307760	0.9624953
77	254	0.577108489	0.92277135	0.749939920	0.916208390	0.9371311
77	115	0.689158820	0.93283212	0.810995469	0.718881754	0.9659160
77	84	0.684347845	0.83008517	0.757216508	0.858571239	0.9241818
77	167	0.657110366	0.95913323	0.808121796	0.717912623	0.9493491
77	50	0.763865842	0.97312929	0.868497568	0.607390794	0.9250762
77	111	0.519472674	0.96546566	0.742469166	0.685925661	0.9660641

1-10 of 260 rows | 1-8 of 9 columns Previous **1** 2 3 4 5 6 ... 26 Next

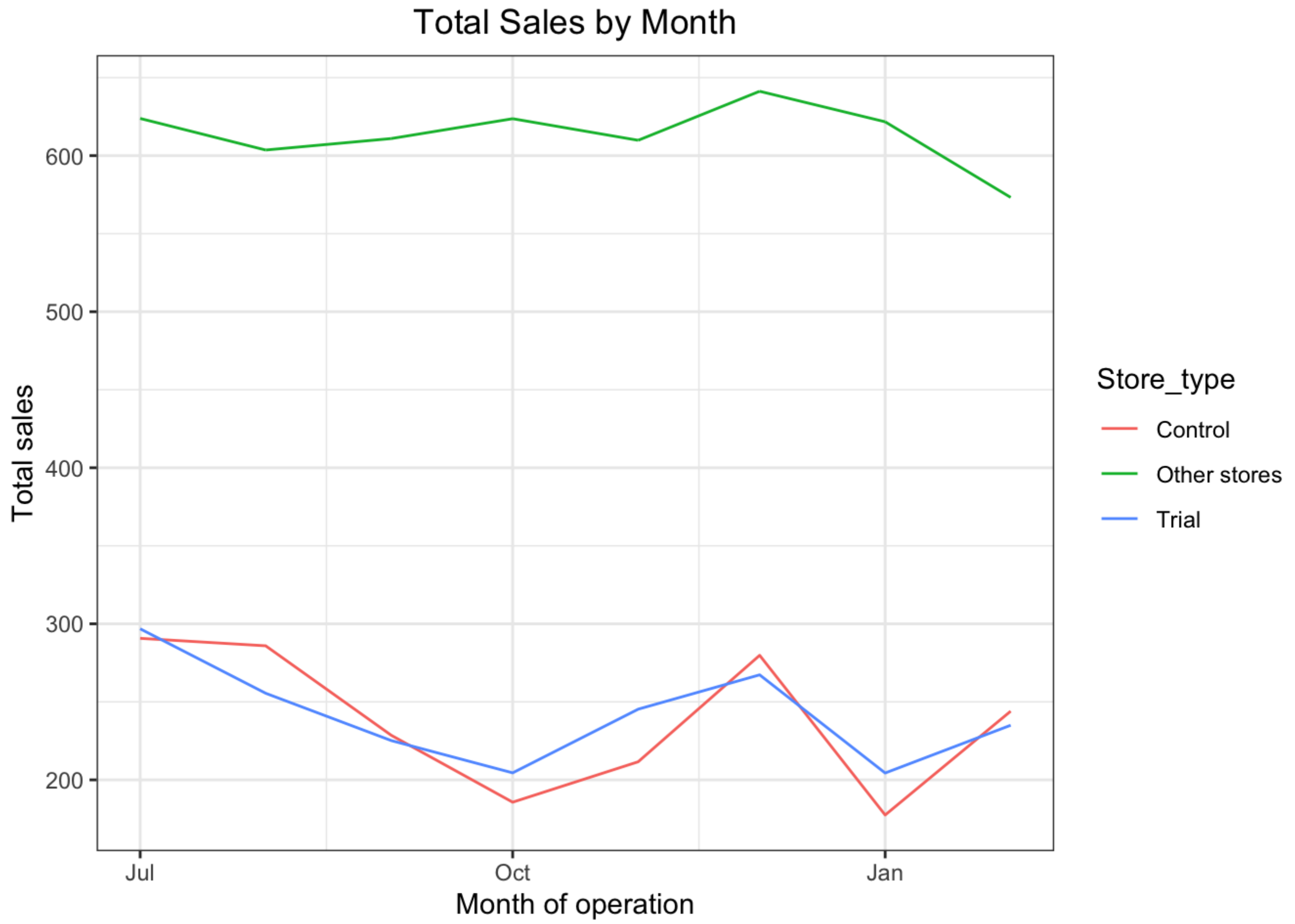
The store with the highest score is then selected as the control store since it is most similar to the trial store.

```
#### 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 the most appropriate control store for trial store 77 by finding the stor
e with the highest final score.
control_store <- score_Control[Store1 == trial_store, ][order(-finalControlScore)][2,
Store2]
control_store
```

```
## [1] 233
```

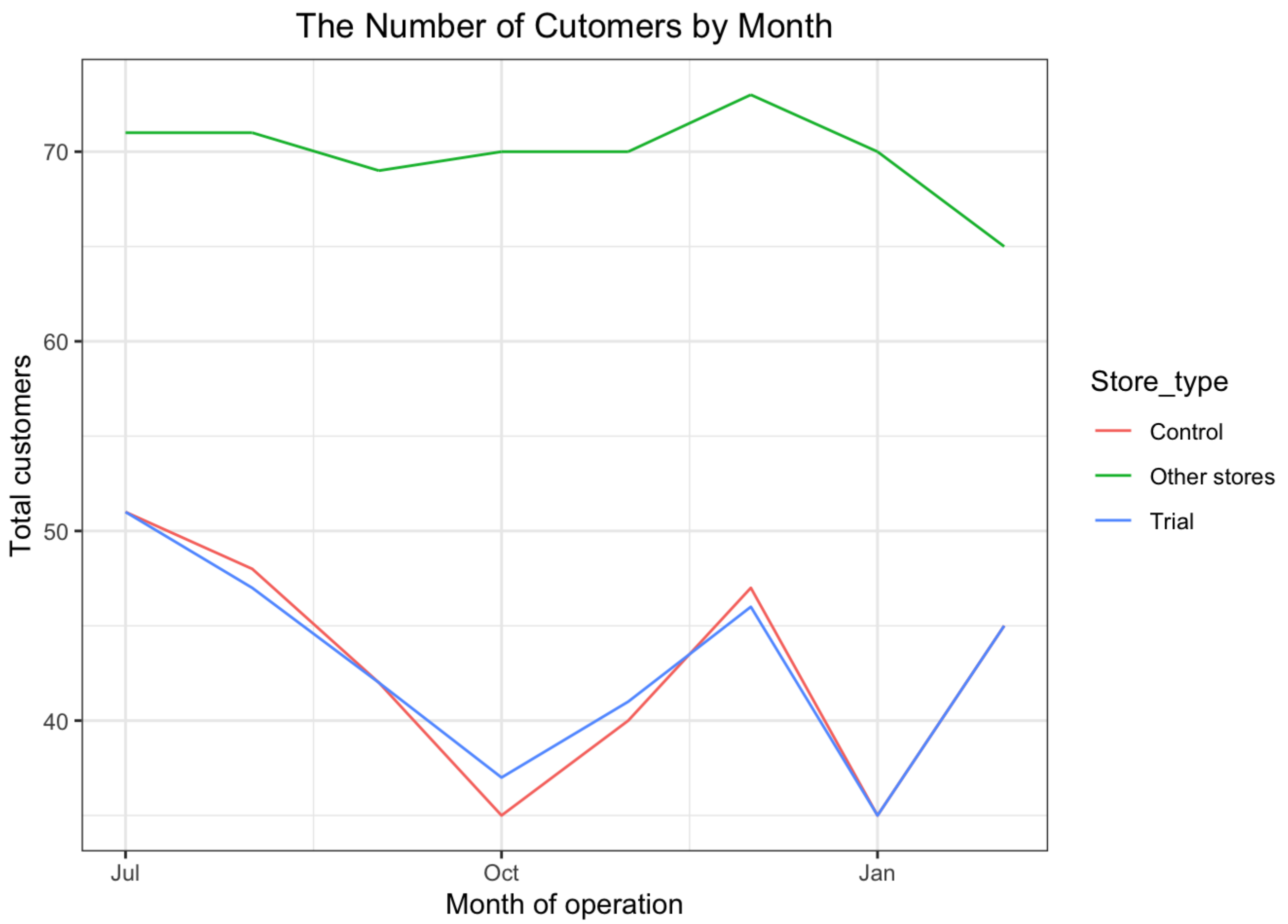
Now that we have found a control store, let's check visually if the drivers are indeed similar in the period before the trial. We'll look at total sales first.

```
#### Visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "T
rial", 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.

```
####Conduct visual checks on customer count trends by comparing the trial store to th
e control store and other stores.
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[,Store_type := ifelse(STORE_NBR == trial_store,
"Trial", ifelse(STORE_NBR == control_store, "Control", "Other stores"))][, nCustomers
:= round(mean(nCustomers)), by = c("YEARMONTH","Store_type")][, TransactionMonth := a
s.Date(paste(YEARMONTH %/%
100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
 ][YEARMONTH < 201903 , ]

measureOverTimeCusts[, nCustomers := as.numeric(nCustomers)]


ggplot(pastCustomers, aes(TransactionMonth,nCustomers , color = Store_type)) +
 geom_line() +
 labs(x = "Month of operation", y = "Total customers", title ="The Number of Cutomers
by Month", fill = "Store Type")
```



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.

```
#### Scale pre-trial control sales to match pre-trial trial store 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, ][ ,controlSal
es := 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.

```
####  Calculate the percentage difference between scaled control sales. and trial sal
es
percentageDiff <- merge(scaledControlSales[, c('YEARMONTH', "controlSales")],
                        measureOverTime[STORE_NBR == trial_store, c("YEARMONTH", "tot
Sales")],
                        by = "YEARMONTH")[, percentageDiff := abs(controlSales-totSal
es)/controlSales]
```

Let's see if the difference is significant!

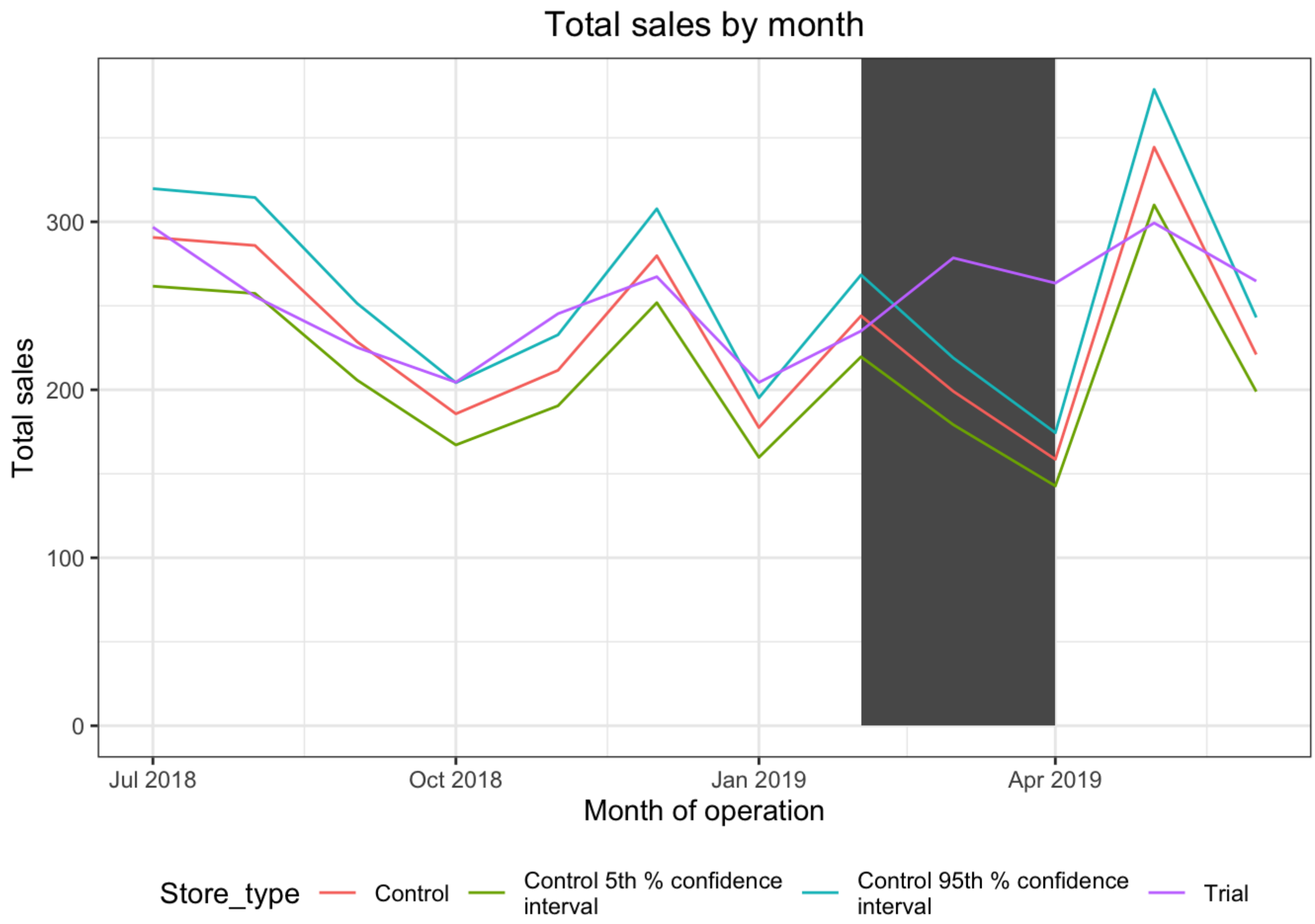
```
#### As our null hypothesis is that the trial period is the same as the pre-trial per
iod, lets take the standard deviation based on the scaled percentage difference in th
e pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
#### Note that there are 8 months in the pre-trial period
#### hence 8 - 1 = 7 degrees of freedom
degreesOfFreedom <- 7
#### We will test with a null hypothesis of there being 0 difference between trial an
d control stores.
#### Calculate the t-values for the trial months. After that, find the 95th percentil
e of the t distribution with the appropriate degrees of freedom
#### to check whether the hypothesis is statistically significant.
percentageDiff[, tValue := (percentageDiff - 0)/stdDev
               ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %%
100, 1,
                                                     sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth,tValue)]
```

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

3 rows

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
#### Trial and control store total sales
#### Create new variables Store_type, totSales and TransactionMonth in the data tabl
e.
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "T
rial",
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)
trialAssessment77 <- 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") +
  theme(legend.position = "bottom")
```

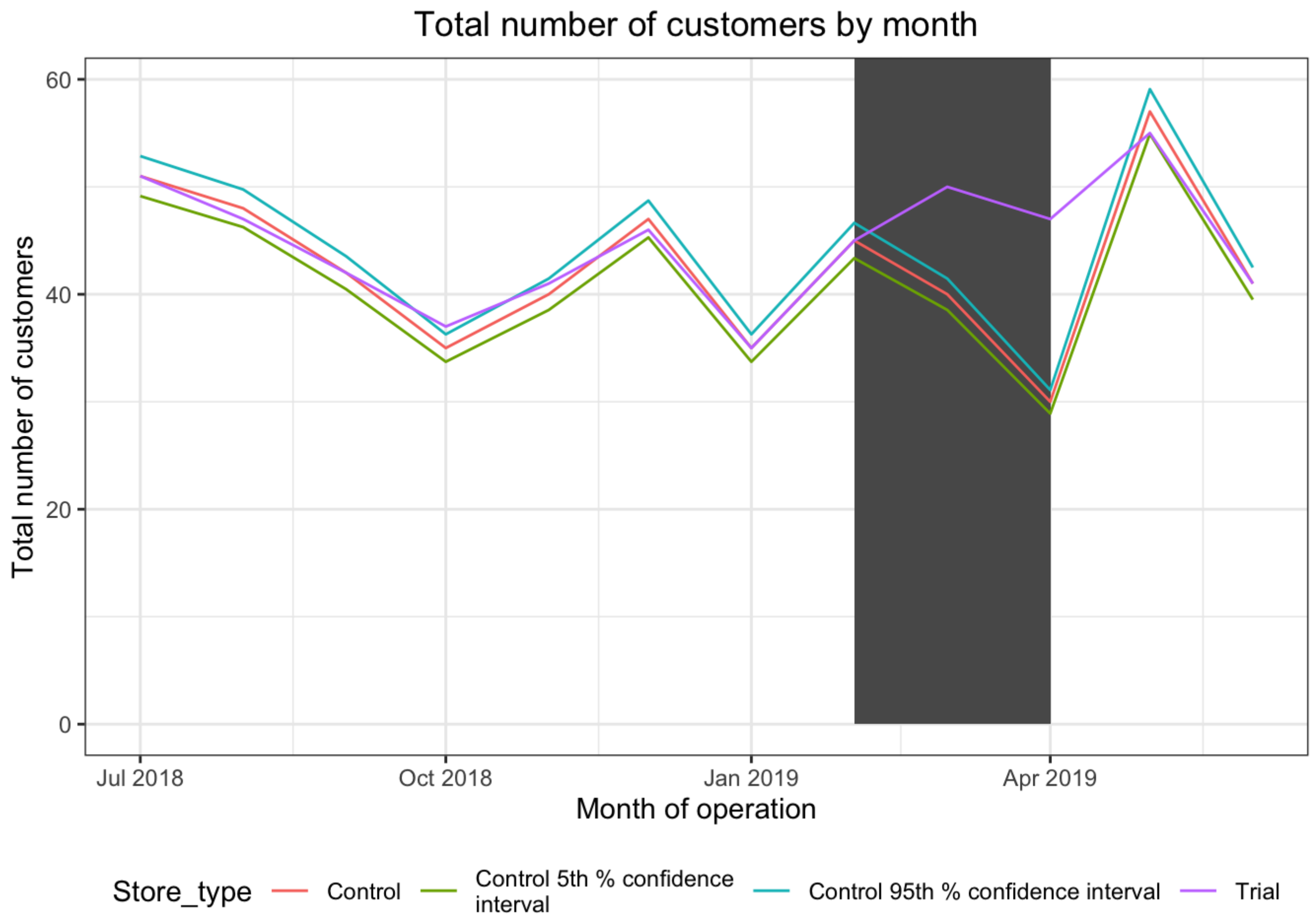


The results show that the trial in store 77 is significantly different to its control store in the trial period as the trial store performance lies outside the 5% to 95% confidence interval of the control store in two of the three trial months. Let’s have a look at assessing this for number of customers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

```r
#### This would be a repeat of the steps before for total sales
#### Scale pre-trial control customers to match pre-trial trial store customers
#### Compute a scaling factor to align control store customer counts to our trial sto
re.
#### Then, apply the scaling factor to control store customer counts.
#### Finally, calculate the percentage difference between scaled control store custom
ers and trial customers.
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", "controlCustomer
s")],
#measureOverTimeCusts[STORE_NBR == trial_store,c("nCustomers", "YEARMONTH")],
#by = "YEARMONTH"
#)[, percentageDiff := abs(controlCustomers-nCustomers)/controlCustomers]


# Merging the tables
mergedTableScale <- merge(
  scaledControlCustomers[, .(YEARMONTH, controlCustomers)],
  measureOverTimeCusts[STORE_NBR == trial_store, .(nCustomers, YEARMONTH)],
  by = "YEARMONTH"
)

# Calculating the percentage difference
percentageDiff <- mergedTableScale[, percentageDiff := abs(controlCustomers - nCustom
ers) / controlCustomers]
```

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

```
#### 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)

#### 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") +
    theme(legend.position = "bottom")
```



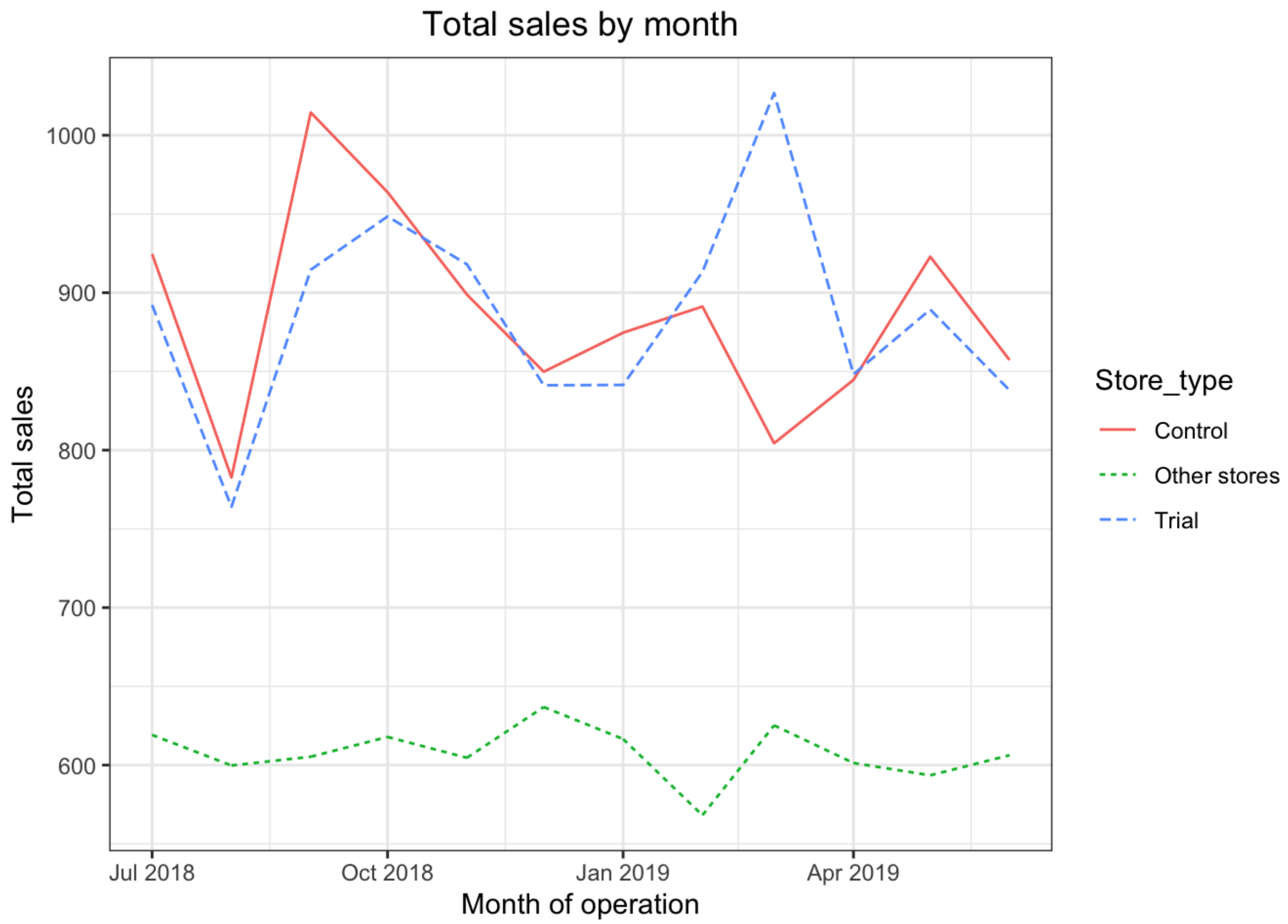
Let’s repeat finding the control store and assessing the impact of the trial for each of the other two trial stores. ## Trial store 86

```
#### Calculate the metrics below as we did for the first trial store.
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = (uniqueN(TXN_ID))/(uniqueN(LYLTY_CARD_NB
R)),
                            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_st
ore)
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), tri
al_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomer
s), 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"))[ , s
coreNSales := (corr_measure + mag_measure)/2]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "St
ore2"))[ , 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
```

```
## [1] 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.

```
#### Conduct visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type:= ifelse(STORE_NBR == trial_store, "Tr
ial", ifelse(STORE_NBR== control_store, "Control", "Other stores"))][, totSales := me
an(totSales), by = c("YEARMONTH", "Store_type")][, TransactionMonth:= as.Date(paste(Y
EARMONTH%/%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")
```



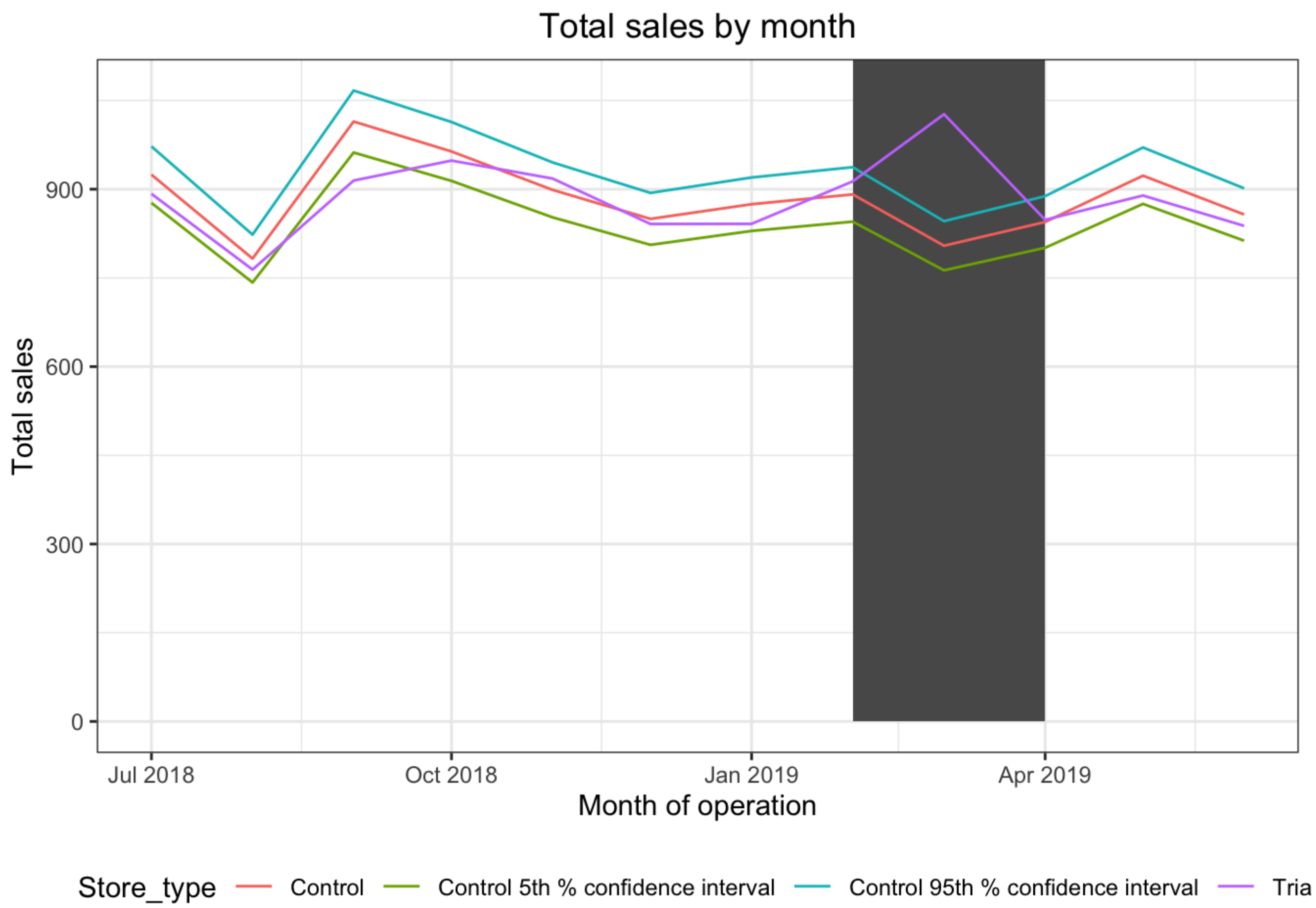
Great, sales are trending in a similar way. Next, number of customers.

```
####Conduct visual checks on trends based on the drivers
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_stor
e, "Trial", ifelse(STORE_NBR == control_store, "Control,", "Other store"))
 ][, 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.

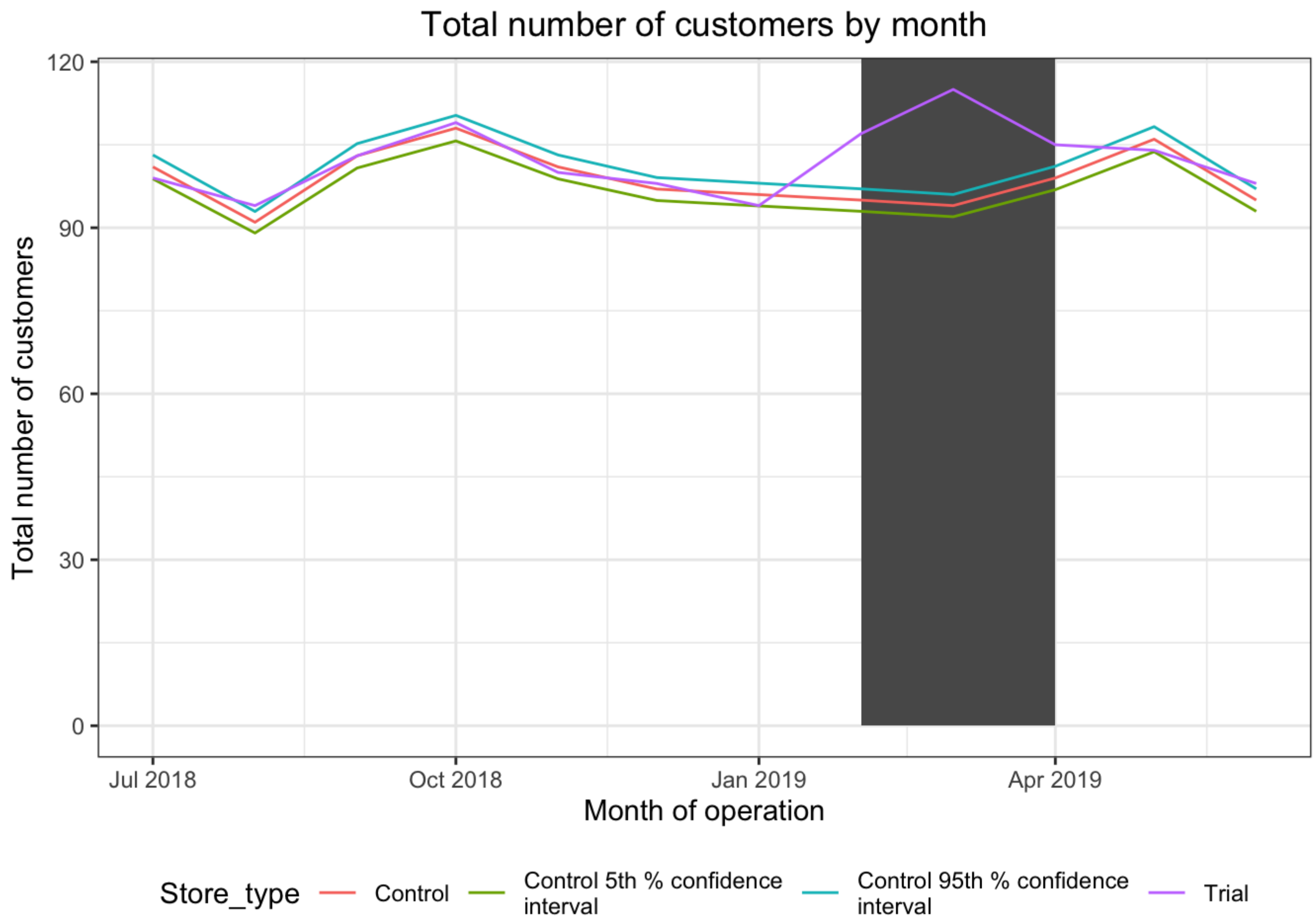
```
#### Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR == control_store &
YEARMONTH < 201902, sum(totSales)]
#### Apply the scaling factor
measureOverTimeSales <- 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 differe
nce
#### 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, "T
rial",
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 deviation
s away from the mean.
#### Recall that the variable stdDev earlier calculates standard deviation in percent
ages, 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)
trialAssessment86 <- 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")+
 theme(legend.position = "bottom")
```



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.

```
#### This would be a repeat of the steps before for total sales
#### Scale pre-trial control customers to match pre-trial trial store customers
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 per
iod, let's take the standard deviation based on the scaled percentage difference in t
he 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") +
    theme(legend.position = "bottom")
```



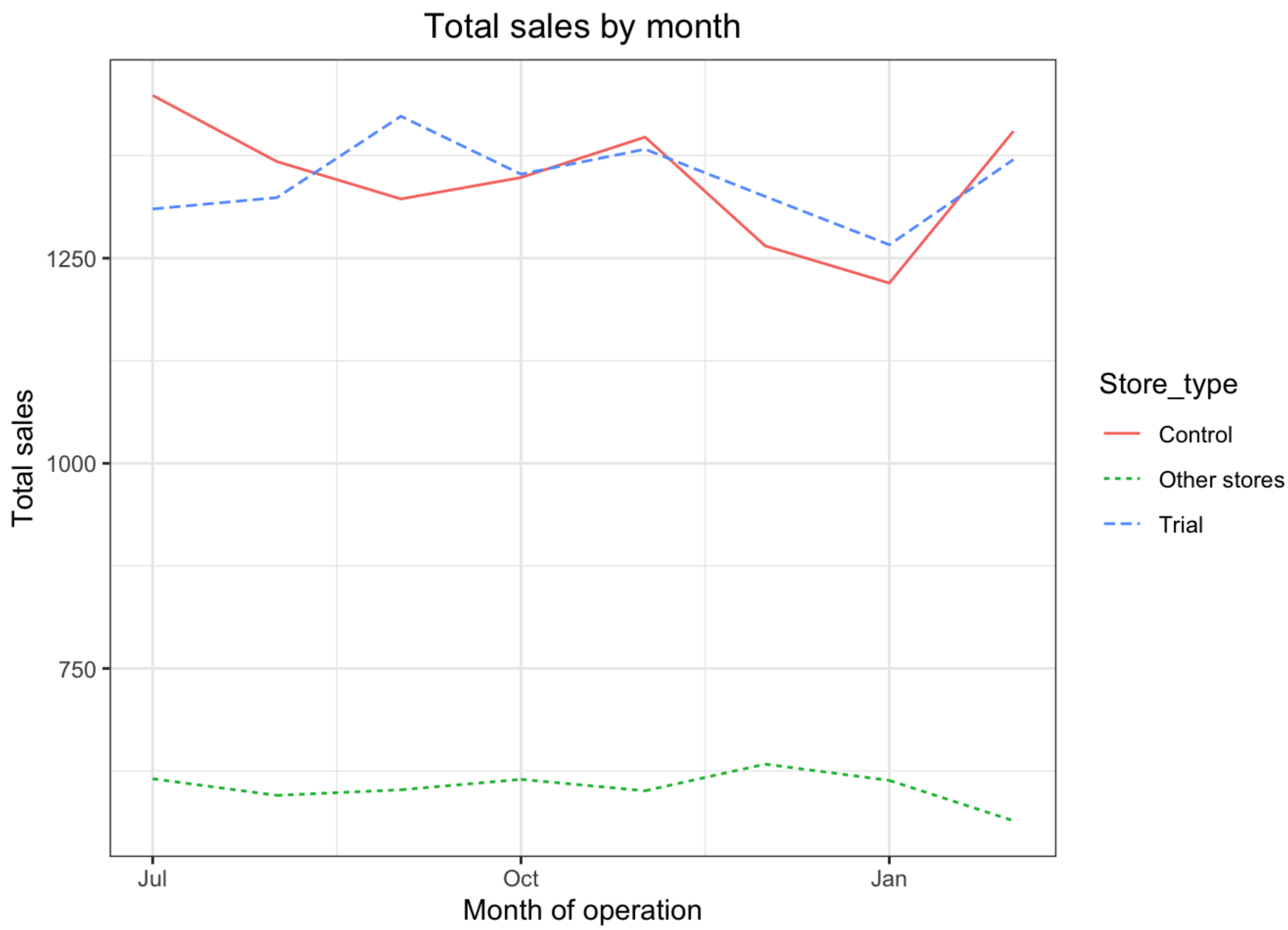
It looks like the number of customers is significantly higher in all of the three months. This seems to suggest that the trial had a significant impact on increasing the number of customers in trial store 86 but as we saw, sales were not significantly higher. We should check with the Category Manager if there were special deals in the trial store that were may have resulted in lower prices, impacting the results. ## Trial store 88

```
#### Conduct the analysis on trial store 88.
measureOverTime <-measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = (uniqueN(TXN_ID))/(uniqueN(LYLTY_CARD_NB
R)),
                            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 num
ber 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_st
ore)

#### 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), tri
al_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomer
s), trial_store)
#### Create a combined score composed of correlation and magnitude by merging the cor
relations table and the magnitudes table, for each driver.
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))[ , s
coreNSales := (corr_measure + mag_measure)/2]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "St
ore2"))[ , scoreNCust := (corr_measure + mag_measure)/2]
#### Combine scores across the drivers by merging sales scores and customer scores, a
nd 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
```

```
## [1] 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.

```
#### Visual checks on trends based on the drivers
#### For the period before the trial, create a graph with total sales of the trial st
ore for each month, compared to the control store and other stores.
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "T
rial",
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")
```

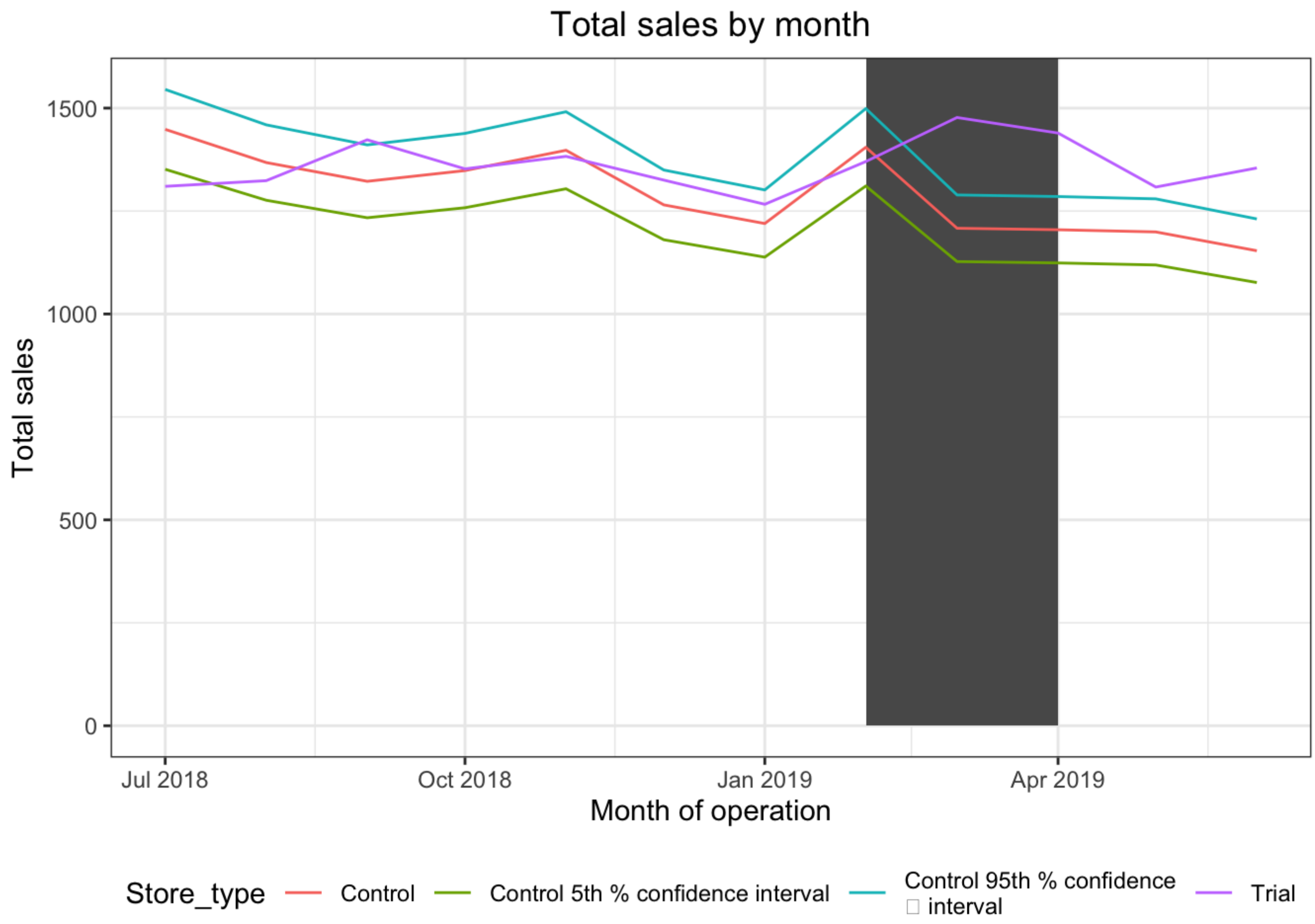


Great, the trial and control stores have similar total sales. Next, number of customers.

```
#### Visual checks on trends based on the drivers
#### For the period before the trial, create a graph with customer counts of the tria
l store for each month, compared to the control store and other stores.
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_stor
e, "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")
```

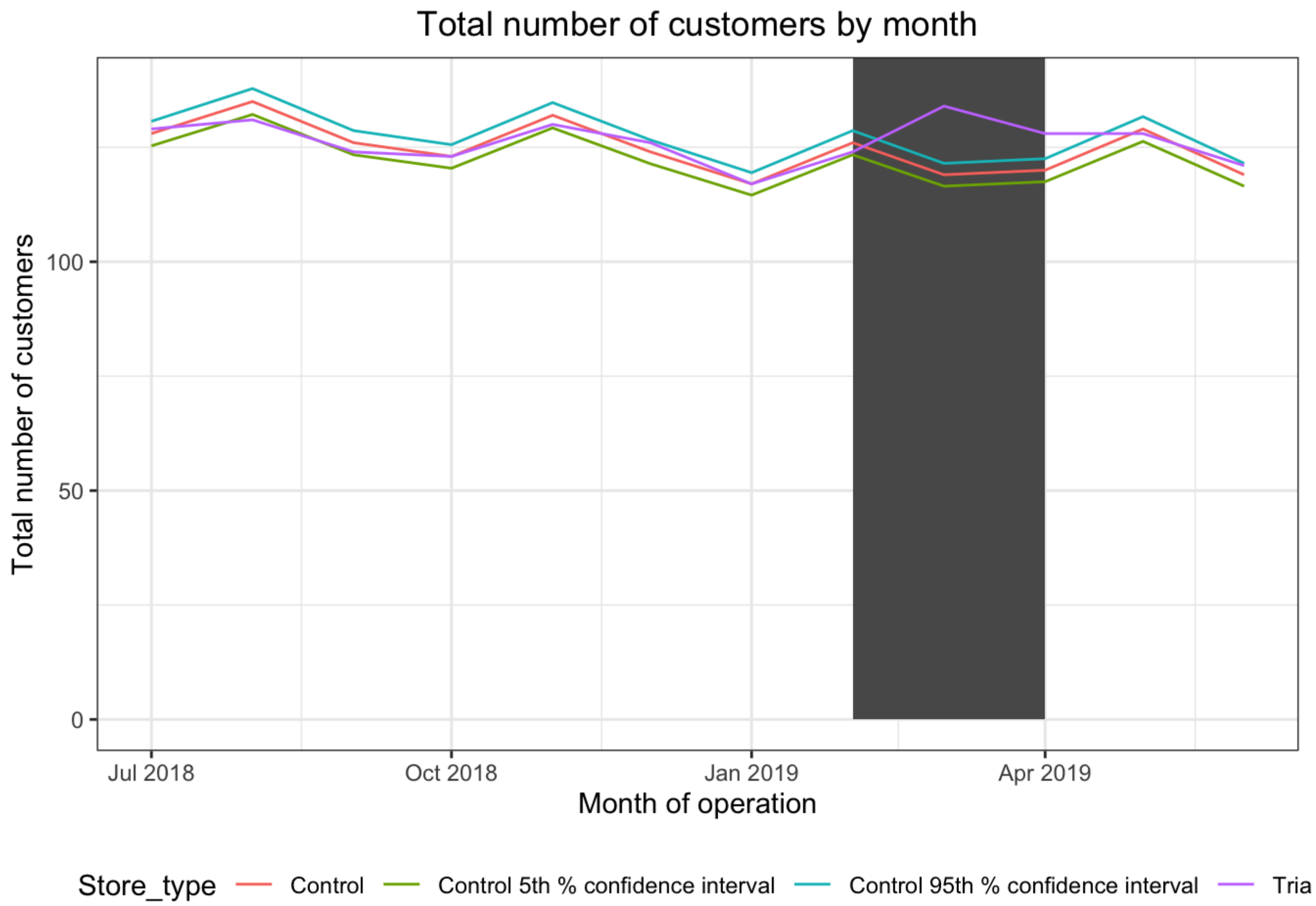
Total number of customers of the control and trial stores are also similar. Let’s now assess the impact of the trial on sales.

```
#### Scale pre-trial control sales to match pre-trial trial store 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, ][ , controlSa
les := totSales * scalingFactorForControlSales]
#### Calculate the percentage difference between scaled control sales and trial sales
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
measureOverTime[STORE_NBR == trial_store, c("totSales", "YEARMONTH")],by = "YEARMONT
H"
)[, percentageDiff := abs(controlSales-totSales)/controlSales]
#### As our null hypothesis is that the trial period is the same as thepre-trial peri
od, let's take the standard deviation based on the scaled percentage difference in th
e 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, "T
rial",
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)
trialAssessment88 <- 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")+
 theme(legend.position = "bottom")
```



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.

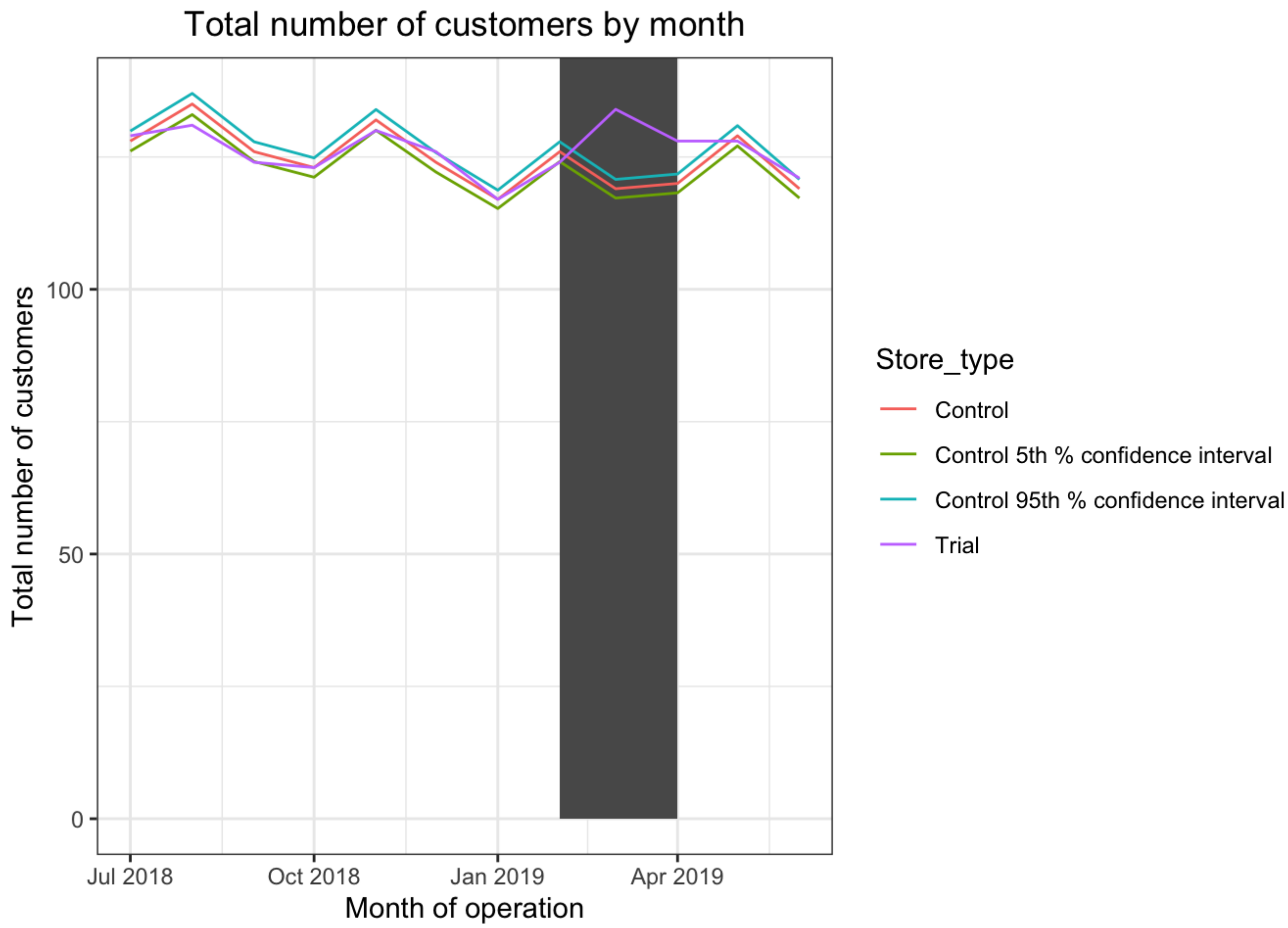
```
#### This would be a repeat of the steps before for total sales #### Scale pre-trial
control store customers to match pre-trial trial store customers scalingFactorForCont
rolCust <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(nCusto
mers)]/preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(nCustome
rs)]

#Apply the scaling factor

measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,][ , contro
lCustomers := nCustomers * scalingFactorForControlCust][, Store_type := ifelse(STORE_
NBR == trial_store, "Trial", ifelse(STORE_NBR == control_store,"Control", "Other stor
es"))]
#### Calculate the absolute percentage difference between scaled control sales and tr
ial sales


percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH","controlCustomers")],m
easureOverTime[STORE_NBR == trial_store, c("nCustomers", "YEARMONTH")], by = "YEARMON
TH")[, 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 t
he 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("YEARMONT
H", "Store_type")][Store_type %in% c("Trial", "Control"), ]

####Control store 95th percentile

pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",][, nCusts := nCust
s * (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_Contro
ls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) + geom_rec
t(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,], aes(xmin = min
(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 , ymax = Inf, color = NUL
L), show.legend = FALSE) + geom_line() + labs(x = "Month of operation", y = "Total nu
mber of customers", title = "Total number of customers by month")+
  theme(legend.position = "bottom")
```



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 month

hs. Let's have a look at assessing this for number of customers as well.

```
#### This would be a repeat of the steps before for total sales
#### Scale pre-trial control store customers to match pre-trial trial store customers
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 tr
ial sales
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH","controlCustomers")],m
easureOverTime[STORE_NBR == trial_store, c("nCustomers", "YEARMONTH")],
by = "YEARMONTH")[, percentageDiff := abs(controlCustomers-nCustomers)/controlCustome
rs]

#### 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 #### di
fference 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 fre
edom
#### Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by = c("YEARMONT
H", "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_Contro
ls5)

#### 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. 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.

How many customers did the trial stores raise compared to the control stores during the trial period?

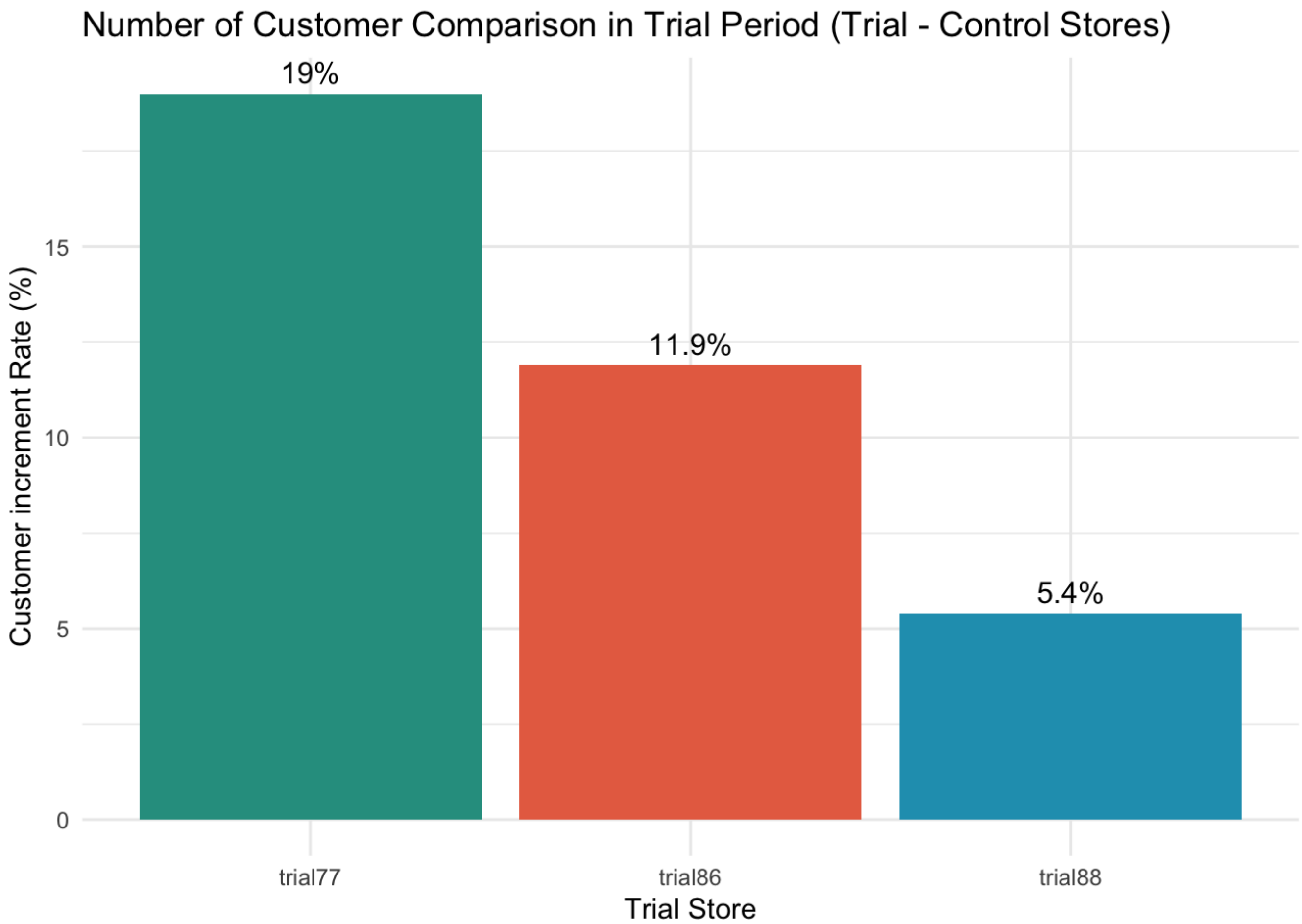
```r
trial_stores <- c(77, 86, 88)
control_stores <- c(233, 155, 237)

trialAssessment77_86 <- merge(trialAssessment77, trialAssessment86, all = TRUE)

# Then, merge the result with table3
trialAssessmentJoined <- merge(trialAssessment77_86, trialAssessment88,  all = TRUE)


# Filter data for the trial period
filtered_data <- trialAssessmentJoined[trialAssessmentJoined$YEARMONTH < 201905 & tri
alAssessmentJoined$YEARMONTH > 201901,]

sales_comparison <- filtered_data %>%
  filter(STORE_NBR %in% c(trial_stores, control_stores)) %>%
  group_by(STORE_NBR) %>%
  summarise(totSales = sum(totSales))

filtered_data <- filtered_data %>%
  filter(Store_type %in% c("Trial", "Control"))
```

How much sales did the trial stores raise compared to the control stores during the trial period?

```
# Calculate number of customer for trial and control stores
nCus_comparison <- filtered_data %>%
  filter(STORE_NBR %in% c(trial_stores, control_stores)) %>%
  group_by(STORE_NBR) %>%
  summarise(nCustomers = sum(nCustomers))

nCus_comparison <- nCus_comparison %>%
  mutate(Store_Type = ifelse(STORE_NBR %in% trial_stores, "Trial", "Control"))



nCus_rate <- data.frame(
  trial77 = numeric(0),
  trial86 = numeric(0),
  trial88 = numeric(0)
)
nCus_rate <- rbind(nCus_rate, c(NA, NA, NA))

# Extract sales for store 77 and store 155

nCus_77 <- nCus_comparison$nCustomers[nCus_comparison$STORE_NBR == 77]
nCus_233 <- nCus_comparison$nCustomers[nCus_comparison$STORE_NBR == 233]

nCus_86 <- nCus_comparison$nCustomers[nCus_comparison$STORE_NBR == 86]
nCus_155 <- nCus_comparison$nCustomers[nCus_comparison$STORE_NBR == 155]

nCus_88 <- nCus_comparison$nCustomers[nCus_comparison$STORE_NBR == 88]
nCus_237 <- nCus_comparison$nCustomers[nCus_comparison$STORE_NBR == 237]



# Perform the calculation
result_77_233 <- round(((nCus_77 - nCus_233) / nCus_77)*100, 1)
result_86_155 <- round(((nCus_86 - nCus_155) / nCus_86)*100, 1)
result_88_237 <- round(((nCus_88 - nCus_237) / nCus_88)*100, 1)

# Assign the result to the sales_rate data frame
nCus_rate$trial77 <- round(((nCus_77 - nCus_233) / nCus_77)*100, 1)
nCus_rate$trial86 <- round(((nCus_86 - nCus_155) / nCus_86)*100, 1)
nCus_rate$trial88 <- round(((nCus_88 - nCus_237) / nCus_88)*100, 1)





nCus_rate_long <- gather(nCus_rate, key = "trial_store", value = "rate")
nCus_rate_long <- na.omit(nCus_rate_long)

# Print the resulting data frame
print(nCus_rate_long)
```

```
##   trial_store rate
## 4     trial77 19.0
## 5     trial86 11.9
## 6     trial88  5.4
```

```
##############################

nCus_rate_chart = ggplot(nCus_rate_long, aes(x = trial_store, y = rate, fill = trial_
store)) +
  geom_bar(stat = "identity", position = "dodge", color = "transparent") +  # Set col
or to transparent
  scale_fill_manual(values = c("#2a9d8f", "#e76f51", "#219ebc")) +  # Colors for each
bar
  labs(title = "Number of Customer Comparison in Trial Period (Trial - Control Store
s)",
       x = "Trial Store",
       y = "Customer increment Rate (%)") +
  theme_minimal()+
  guides(fill = "none")   # Remove the legend

nCus_rate_chart + geom_text(position = position_dodge(width = 0.9),    # Adjust the p
osition as needed
            vjust = -0.5,                           # Adjust the vertical position
            size = 4,                               # Adjust the text size
            color = "black",                        # Text color
            aes(label = paste0(round(rate, 1), "%")),  # Add % symbol to the label
            show.legend = FALSE)
```




```
  #geom_text(aes( label = paste(round(.data[["PERCENT"]], 1), "%")), size = 3, color
= "black")
```

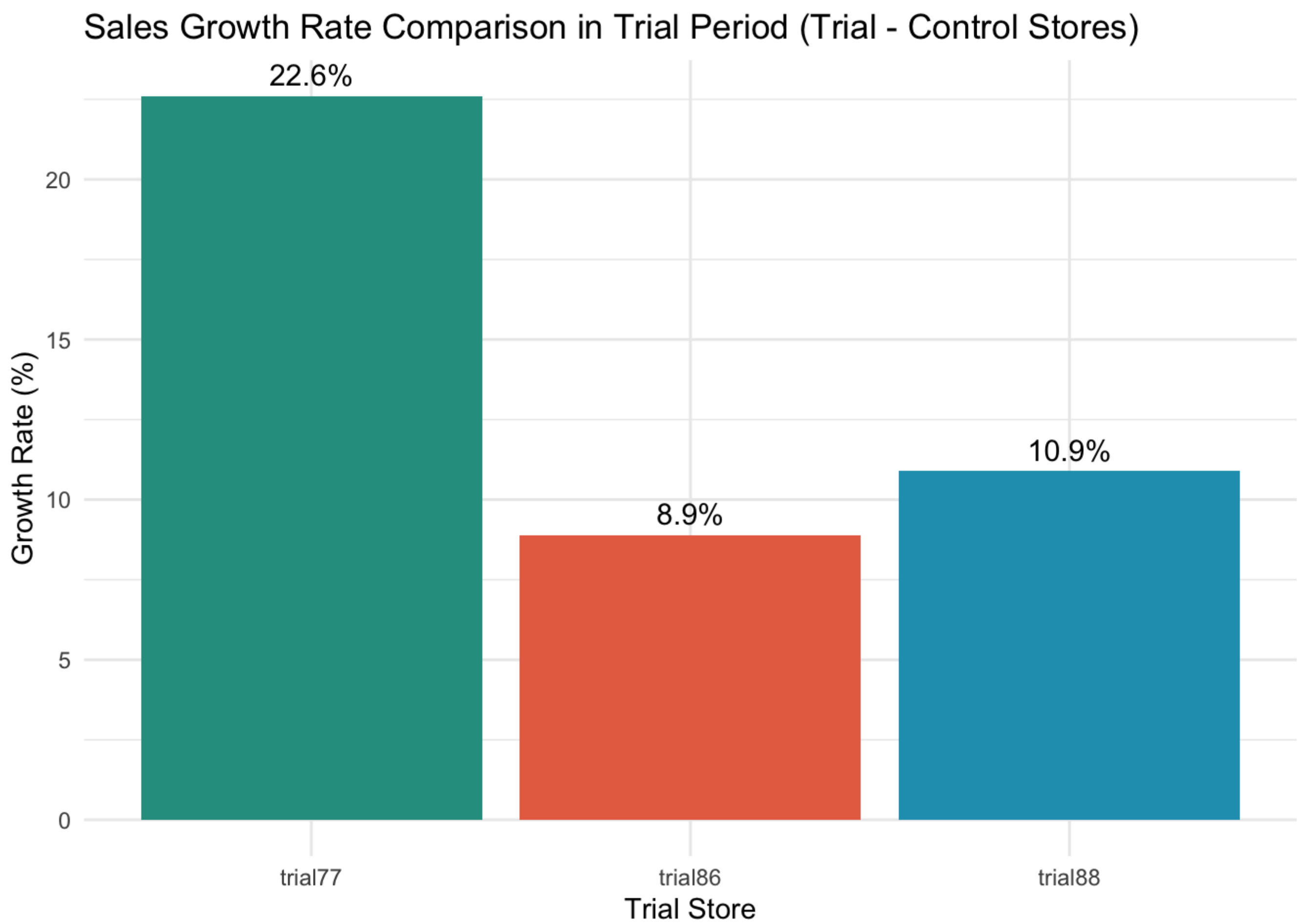
How many customers did the trial stores raise compared to the control stores during the trial period?

```r
# Calculate sales for trial and control stores

sales_comparison <- filtered_data %>%
  filter(STORE_NBR %in% c(trial_stores, control_stores)) %>%
  group_by(STORE_NBR) %>%
  summarise(totSales = sum(totSales))

sales_comparison <- sales_comparison %>%
  mutate(Store_Type = ifelse(STORE_NBR %in% trial_stores, "Trial", "Control"))



sales_rate <- data.frame(
  trial77 = numeric(0),
  trial86 = numeric(0),
  trial88 = numeric(0)
)
sales_rate <- rbind(sales_rate, c(NA, NA, NA))


# Extract sales for store 77 and store 155
sales_77 <- sales_comparison$totSales[sales_comparison$STORE_NBR == 77]
sales_233 <- sales_comparison$totSales[sales_comparison$STORE_NBR == 233]

sales_86 <- sales_comparison$totSales[sales_comparison$STORE_NBR == 86]
sales_155 <- sales_comparison$totSales[sales_comparison$STORE_NBR == 155]

sales_88 <- sales_comparison$totSales[sales_comparison$STORE_NBR == 88]
sales_237 <- sales_comparison$totSales[sales_comparison$STORE_NBR == 237]

# Perform the calculation
result_77_233 <- round(((sales_77 - sales_233) / sales_77)*100, 1)
result_86_155 <- round(((sales_86 - sales_155) / sales_86)*100, 1)
result_88_237 <- round(((sales_88 - sales_237) / sales_88)*100, 1)

# Assign the result to the sales_rate data frame
sales_rate$trial77 <- result_77_233
sales_rate$trial86 <- result_86_155
sales_rate$trial88 <- result_88_237


sales_rate_long <- gather(sales_rate, key = "trial_store", value = "rate")
sales_rate_long <- na.omit(sales_rate_long)

# Print the resulting data frame
print(sales_rate_long)
```

```
##   trial_store rate
## 4     trial77 22.6
## 5     trial86  8.9
## 6     trial88 10.9
```

```
# make a chart

ggplot(sales_rate_long, aes(x = trial_store, y = rate, fill = trial_store)) +
  geom_bar(stat = "identity", position = "dodge", color = "transparent") +  # Set col
or to transparent
  scale_fill_manual(values = c("#2a9d8f", "#e76f51", "#219ebc")) +  # Colors for each
bar
  labs(title = "Sales Growth Rate Comparison in Trial Period (Trial - Control Store
s)",
       x = "Trial Store",
       y = "Growth Rate (%)") +
  theme_minimal() +
  guides(fill = FALSE) + # Remove the legend
  geom_text(position = position_dodge(width = 0.9),    # Adjust the position as neede
d
            vjust = -0.5,                              # Adjust the vertical position
            size = 4,                                  # Adjust the text size
            color = "black",                           # Text color
            aes(label = paste0(round(rate, 1), "%")), # Add % symbol to the label
            show.legend = FALSE)                       # Exclude from legend
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instea
d as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



The average price spent per customer transition

```
avg_comparison <- filtered_data %>%
  filter(STORE_NBR %in% c(trial_stores, control_stores)) %>%
  group_by(STORE_NBR) %>%
  summarise(average_price_per_cust = sum(totSales)/sum(nCustomers))

avg_comparison <- avg_comparison %>%
  mutate(Store_Type = ifelse(STORE_NBR %in% trial_stores, "Trial", "Control"))



# Extract sales for store 77, 86, 88 and store 233, 155, 237

avg_price_77 <- avg_comparison$average_price_per_cust[avg_comparison$STORE_NBR == 77]
avg_price_233 <- avg_comparison$average_price_per_cust[avg_comparison$STORE_NBR == 23
3]

avg_price_86 <- avg_comparison$average_price_per_cust[avg_comparison$STORE_NBR == 86]
avg_price_155 <- avg_comparison$average_price_per_cust[avg_comparison$STORE_NBR == 15
5]

avg_price_88 <- avg_comparison$average_price_per_cust[avg_comparison$STORE_NBR == 88]
avg_price_237 <- avg_comparison$average_price_per_cust[avg_comparison$STORE_NBR == 23
7]


# Create an empty data frame
avg_price_rate <- data.frame(
  trial77 = numeric(0),
  trial86 = numeric(0),
  trial88 = numeric(0)
)
avg_price_rate <- rbind(avg_price_rate, c(NA, NA, NA))

# Assign the result to the avg_price_rate data frame
avg_price_rate$trial77 <- round(((avg_price_77 - avg_price_233) / avg_price_77)*100,
1)
avg_price_rate$trial86 <- round(((avg_price_86 - avg_price_155) / avg_price_86)*100,
1)
avg_price_rate$trial88 <- round(((avg_price_88 - avg_price_237) / avg_price_88)*100,
1)


avg_price_rate_long <- gather(avg_price_rate, key = "trial_store", value = "rate")
avg_price_rate_long <- na.omit(avg_price_rate_long)

# make a chart

ggplot(avg_price_rate_long, aes(x = trial_store, y = rate, fill = trial_store)) +
  geom_bar(stat = "identity", position = "dodge", color = "transparent") +  # Set col
or to transparent
  scale_fill_manual(values = c("#2a9d8f", "#e76f51", "#219ebc")) +  # Colors for each
bar
  labs(title = "Average Price Spent per Customer Comparison (Trial - Control Store
s)",
       x = "Trial Store",
```

```
      y = "Growth Rate (%)") +
  theme_minimal() +
  guides(fill = FALSE) + # Remove the legend
  geom_text(position = position_dodge(width = 0.9),    # Adjust the position as neede
d
            vjust = -0.5,                              # Adjust the vertical position
            size = 4,                                  # Adjust the text size
            color = "black",                           # Text color
            aes(label = paste0(round(rate, 1), "%")),  # Add % symbol to the label
            show.legend = FALSE)                       # Exclude from legend
```

Deep dive into 86

```
avg_price_rate_long
```

	trial_store <chr>	rate <dbl>
4	trial77	4.4
5	trial86	-3.4
6	trial88	5.8

3 rows

```r
colnames(avg_price_rate_long)[c(2)] <- c("avg_price_rate")

sales_rate_long
```

	trial_store <chr>	rate <dbl>
4	trial77	22.6
5	trial86	8.9
6	trial88	10.9

3 rows

```r
colnames(sales_rate_long)[c(2)] <- c("sales_rate")

nCus_rate_long
```

	trial_store <chr>	rate <dbl>
4	trial77	19.0
5	trial86	11.9
6	trial88	5.4

3 rows

```r
colnames(nCus_rate_long)[c(2)] <- c("nCus_rate")

growth_rate_joined <- merge(merge(sales_rate_long, nCus_rate_long, by = "trial_stor
e"), avg_price_rate_long, by = "trial_store")

library(reshape2)
```

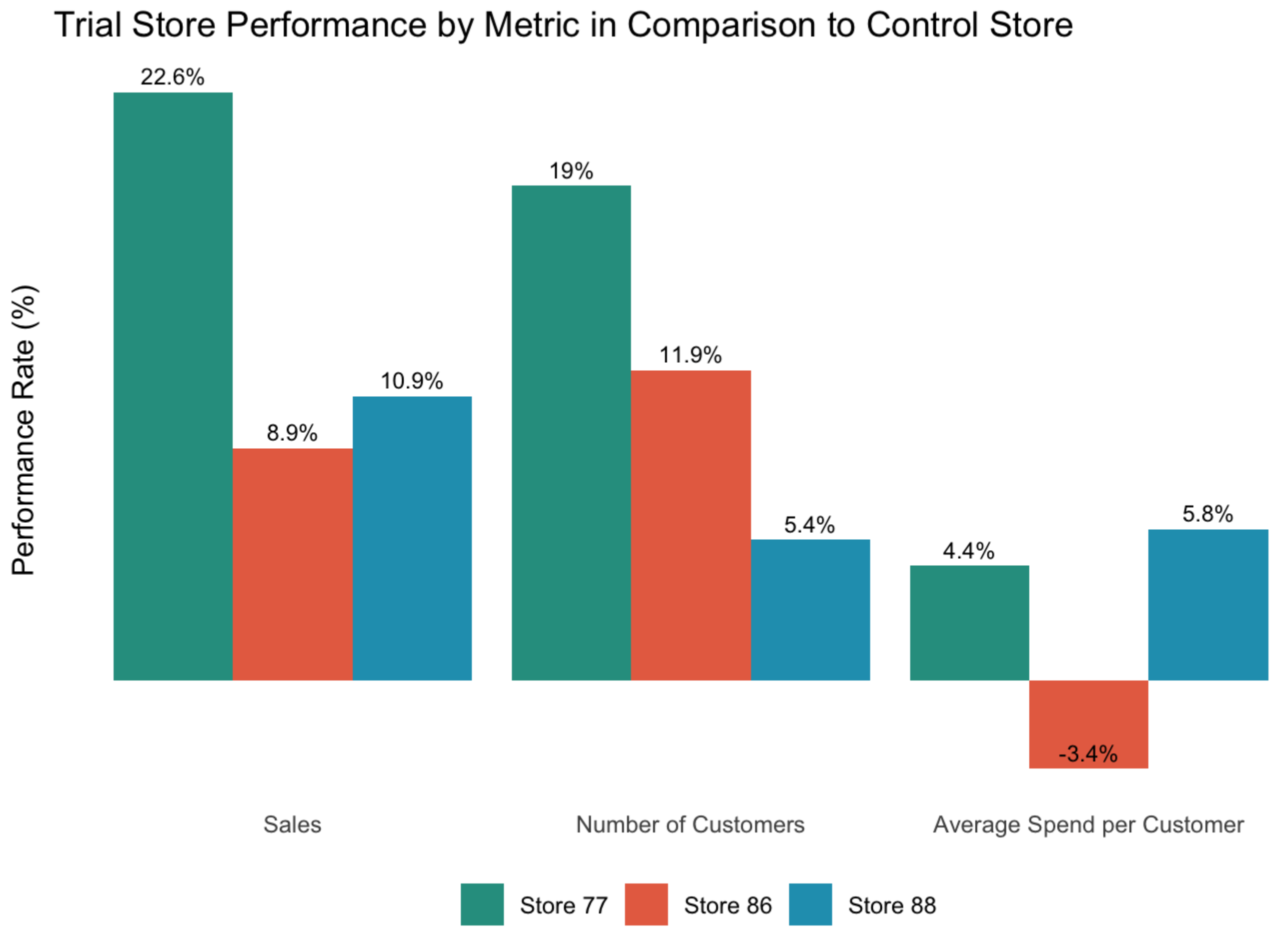
```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

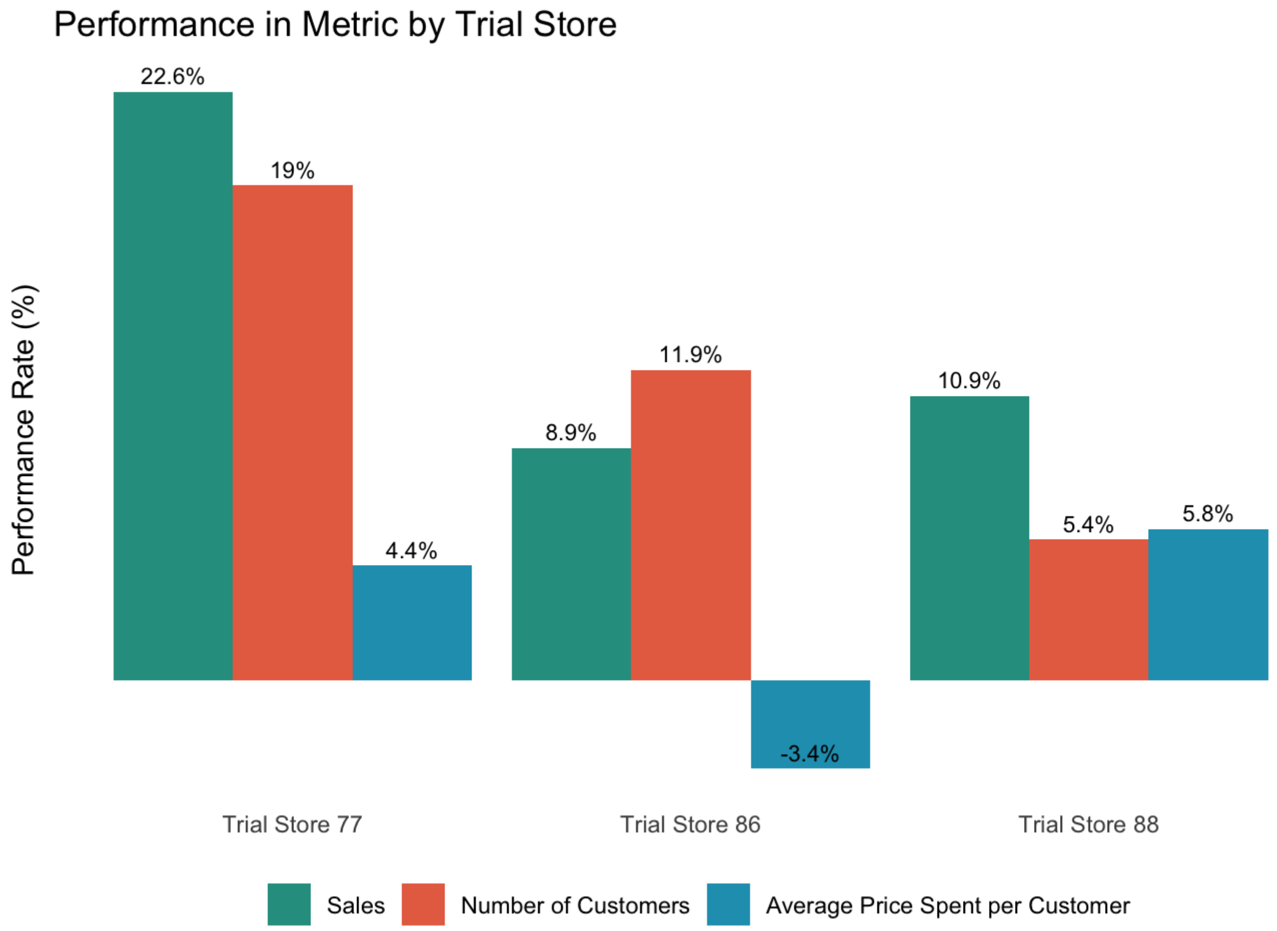
```r
# Melt the data frame for easy plotting
growth_rate_joined_melted <- reshape2::melt(growth_rate_joined, id.vars = "trial_stor
e")


# Performance by metric
ggplot(growth_rate_joined_melted, aes(x = variable, y = value, fill = trial_store)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Trial Store Performance by Metric in Comparison to Control Store",
       x = NULL,
       y = "Performance Rate (%)") +
  scale_fill_manual(
    values = c("#2a9d8f", "#e76f51", "#219ebc"),
    labels = c("Store 77", "Store 86", "Store 88")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5)) +
  guides(fill = guide_legend(title = NULL)) +
  coord_cartesian(clip = 'off') +
  theme(legend.position = "bottom") +
  scale_x_discrete(labels = c("Sales", "Number of Customers", "Average Spend per Cust
omer"))+
   coord_cartesian(clip = 'off') +
    theme(axis.text.y = element_blank(),  # Remove y-axis labels
        axis.ticks.y = element_blank(),  # Remove y-axis ticks
      panel.grid.major = element_blank(),  # Remove major grid lines
       panel.grid.minor = element_blank(),  # Remove minor grid lines
        legend.position = "bottom")  +    # Place legend at the bottom
  geom_text(position = position_dodge(width = 0.9),    # Adjust the position as neede
d
            vjust = -0.5,                              # Adjust the vertical position
            size = 3,                                  # Adjust the text size
            color = "black",                           # Text color
            aes(label = paste0(round(value, 1), "%")),  # Add % symbol to the label
            show.legend = FALSE)                       # Exclude from legend
```

```
## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.
```




```
# Performance by trial store
ggplot(growth_rate_joined_melted, aes(x = trial_store, y = value, fill = variable)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Performance in Metric by Trial Store",
       x = NULL,
       y = "Performance Rate (%)") +
  scale_fill_manual(
    values = c("#2a9d8f", "#e76f51", "#219ebc"),
    labels = c("Sales", "Number of Customers", "Average Price Spent per Customer")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5)) +
  guides(fill = guide_legend(title = NULL)) +
  coord_cartesian(clip = 'off') +
  theme(legend.position = "bottom") +
  scale_x_discrete(labels = c("Trial Store 77", "Trial Store 86", "Trial Store 88"))+
   coord_cartesian(clip = 'off') +
    theme(axis.text.y = element_blank(),  # Remove y-axis labels
        axis.ticks.y = element_blank(),  # Remove y-axis ticks
      panel.grid.major = element_blank(),  # Remove major grid lines
       panel.grid.minor = element_blank(),  # Remove minor grid lines
        legend.position = "bottom")  +    # Place legend at the bottom
  geom_text(position = position_dodge(width = 0.9),    # Adjust the position as neede
d
            vjust = -0.5,                             # Adjust the vertical position
            size = 3,                                 # Adjust the text size
            color = "black",                          # Text color
            aes(label = paste0(round(value, 1), "%")),  # Add % symbol to the label
            show.legend = FALSE)                      # Exclude from legend
```

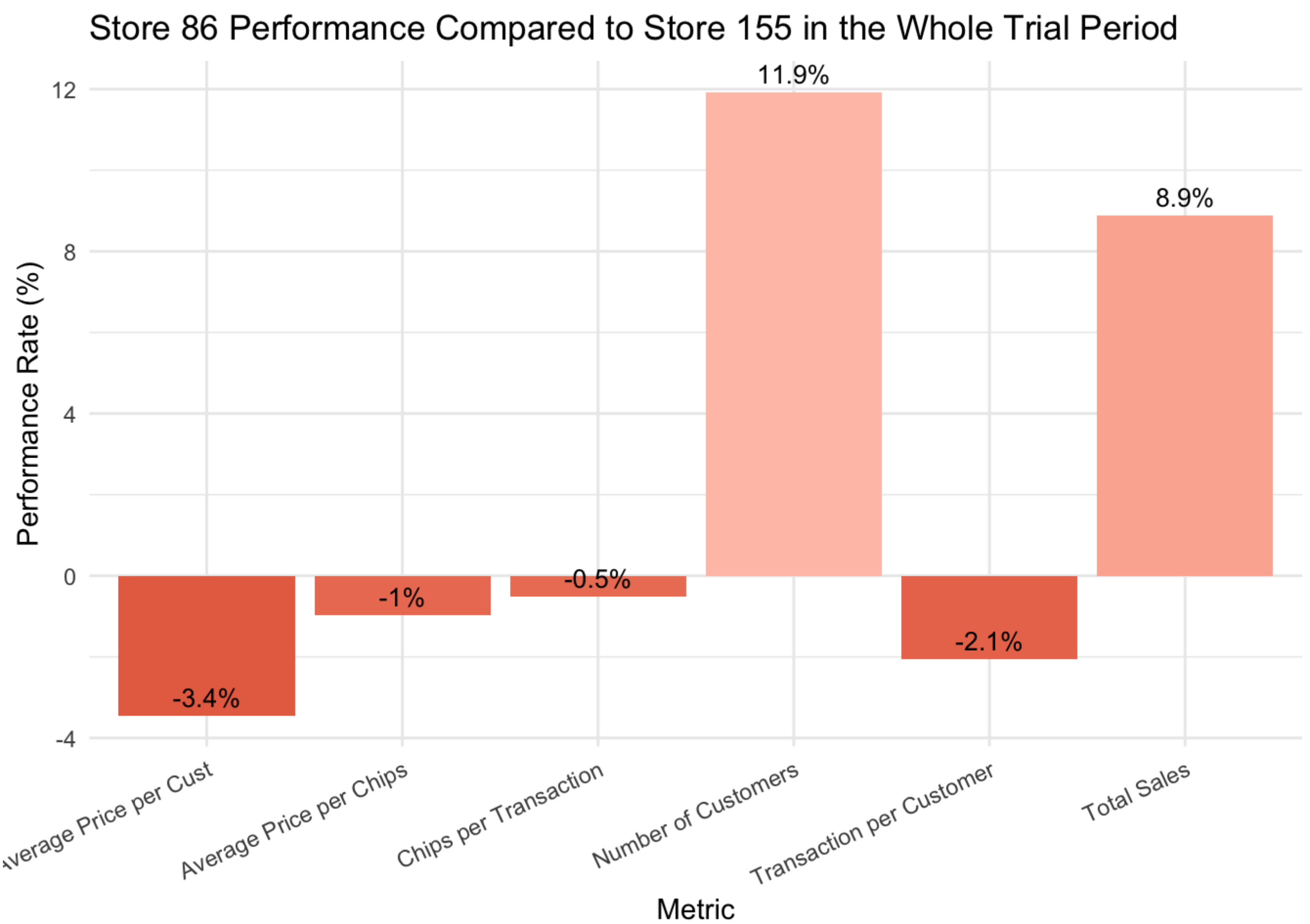
```
## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.
```



compare the performance of 86 to 155 in the whole trial period

```
assembled_trial_period = filtered_data
assembled_trial_period <- select(assembled_trial_period, -c(numberCustomers.x, number
Customers.y))

assembled_trial_period$avgPricePerCust = assembled_trial_period$totSales/assembled_tr
ial_period$nCustomers

assembled_trial_period = assembled_trial_period %>%
  group_by(STORE_NBR) %>%
  summarise(
    totSales = sum(totSales),
    nTxnPerCust = sum(nTxnPerCust)/3,
    nChipsPerTxn = sum(nChipsPerTxn)/3,
    avgPricePerUnit = sum(avgPricePerUnit)/3,
    avgPricePerCust = sum(avgPricePerCust)/3,
    nCustomers =  sum(nCustomers)
    )

assembled_86_155 = assembled_trial_period %>%
  filter(STORE_NBR == 86| STORE_NBR == 155)

your_data_restructured <- pivot_longer(assembled_86_155, cols = -c(STORE_NBR), names_
to = "Metric", values_to = "Value")


# Filter rows
desired_rows <- c("totSales", "nTxnPerCust", "nChipsPerTxn", "avgPricePerUnit", "avgP
ricePerCust", "nCustomers")
your_data_restructured <- your_data_restructured %>% filter(Metric %in% desired_rows)

# Add a new column for the store-specific column names
your_data_restructured$STORE_NBR <- paste("STORE_NBR_", your_data_restructured$STORE_
NBR, sep = "")

# Pivot again to get the desired structure
final_result <- pivot_wider(your_data_restructured, names_from = STORE_NBR, values_fr
om = Value)


result_86_155 <- final_result


# make a column to see the store 86 performance compared to 155 at each metric
result_86_155[5, "STORE_NBR_86"] <- avg_price_86
result_86_155[5, "STORE_NBR_155"] <- avg_price_155


result_86_155 <- result_86_155 %>%
  mutate(Performance_86 = ((`STORE_NBR_86` - `STORE_NBR_155`) / `STORE_NBR_86`)*100)

result_86_155 = result_86_155 %>%
  arrange(Metric)

# make a chart
ggplot(result_86_155, aes(x = Metric, y = Performance_86, fill = Performance_86)) +
```

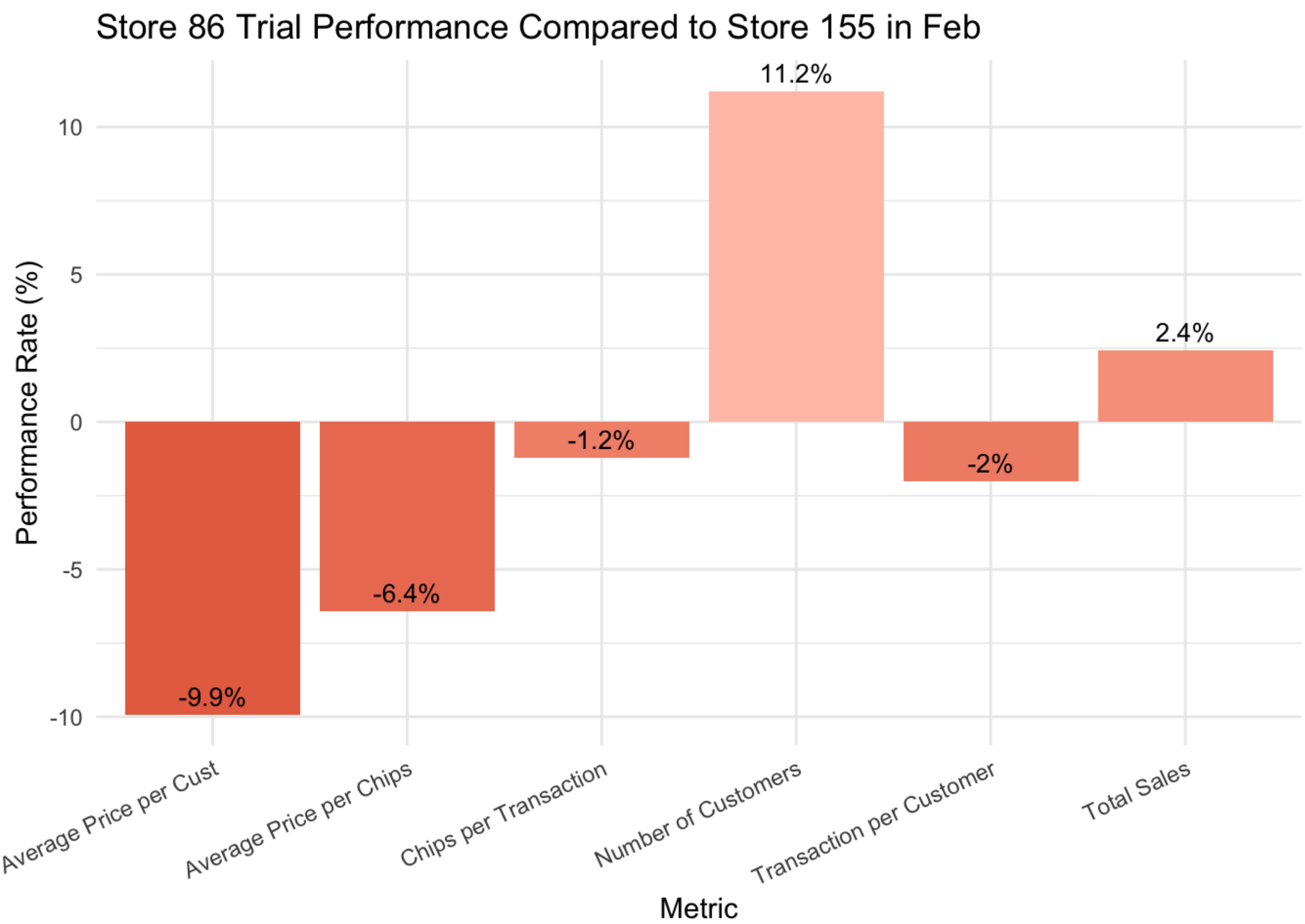
```
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Store 86 Performance Compared to Store 155 in the Whole Trial Perio
d",
       x = "Metric",
       y = "Performance Rate (%)") +
  theme_minimal() +
  guides(fill = FALSE) + # Remove the legend
  theme(axis.text.x = element_text(angle = 25, hjust = 0))  +
    scale_fill_gradient(low = "#e76f51", high = "#ffc4b5") +
    geom_text(position = position_dodge(width = 0.9),    # Adjust the position as nee
ded
              vjust = -0.5,                               # Adjust the vertical position
              size = 3.5,                                   # Adjust the text size
              color = "black",                            # Text color
                 aes(label = paste0(round(Performance_86, 1), "%")),
              show.legend = FALSE)+
  theme(axis.text.x = element_text(hjust = 1)) +
  scale_x_discrete(labels = c("Average Price per Cust", "Average Price per Chips", "C
hips per Transaction", "Number of Customers", "Transaction per Customer", "Total Sale
s"))
```



compare the performance of 86 to 155 in Feb

```
assembled_feb_86_155 = filtered_data %>%
  filter(YEARMONTH == 201902 & STORE_NBR == 86 | YEARMONTH == 201902 & STORE_NBR == 1
55)

assembled_feb_86_155 <- select(assembled_feb_86_155, -c(numberCustomers.x, numberCust
omers.y, Store_type, TransactionMonth, YEARMONTH))

assembled_feb_86_155$avgPricePerCust = assembled_feb_86_155$totSales/assembled_feb_86
_155$nCustomers



assembled_feb_86_155_restructured <- pivot_longer(assembled_feb_86_155, cols = -c(STO
RE_NBR), names_to = "Metric", values_to = "Value")

# Filter rows
desired_rows_feb <- c("totSales", "nTxnPerCust", "nChipsPerTxn", "avgPricePerUnit",
"avgPricePerCust", "nCustomers")
assembled_feb_86_155_restructured <- assembled_feb_86_155_restructured %>% filter(Met
ric %in% desired_rows_feb)

# Add a new column for the store-specific column names
assembled_feb_86_155_restructured$STORE_NBR <- paste("STORE_NBR_", assembled_feb_86_1
55_restructured$STORE_NBR, sep = "")

# Pivot again to get the desired structure
final_assembled_feb_86_155_ <- pivot_wider(assembled_feb_86_155_restructured, names_f
rom = STORE_NBR, values_from = Value)



final_86_155_feb <- final_assembled_feb_86_155_






final_86_155_feb <- final_86_155_feb %>%
  mutate(Performance_86 = ((`STORE_NBR_86` - `STORE_NBR_155`) / `STORE_NBR_86`)*100)

final_86_155_feb = final_86_155_feb %>%
  arrange(Metric)
# make a chart
ggplot(final_86_155_feb, aes(x = Metric, y = Performance_86, fill = Performance_86))
+
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Store 86 Trial Performance Compared to Store 155 in Feb",
       x = "Metric",
       y = "Performance Rate (%)") +
  theme_minimal() +
  guides(fill = FALSE) + # Remove the legend
 theme(axis.text.x = element_text(angle = 25, hjust = 0))  +
    scale_fill_gradient(low = "#e76f51", high = "#ffc4b5") +
    geom_text(position = position_dodge(width = 0.9),    # Adjust the position as nee
ded
            vjust = -0.5,                               # Adjust the vertical position
            size = 3.5,                                   # Adjust the text size
            color = "black",                            # Text color
```

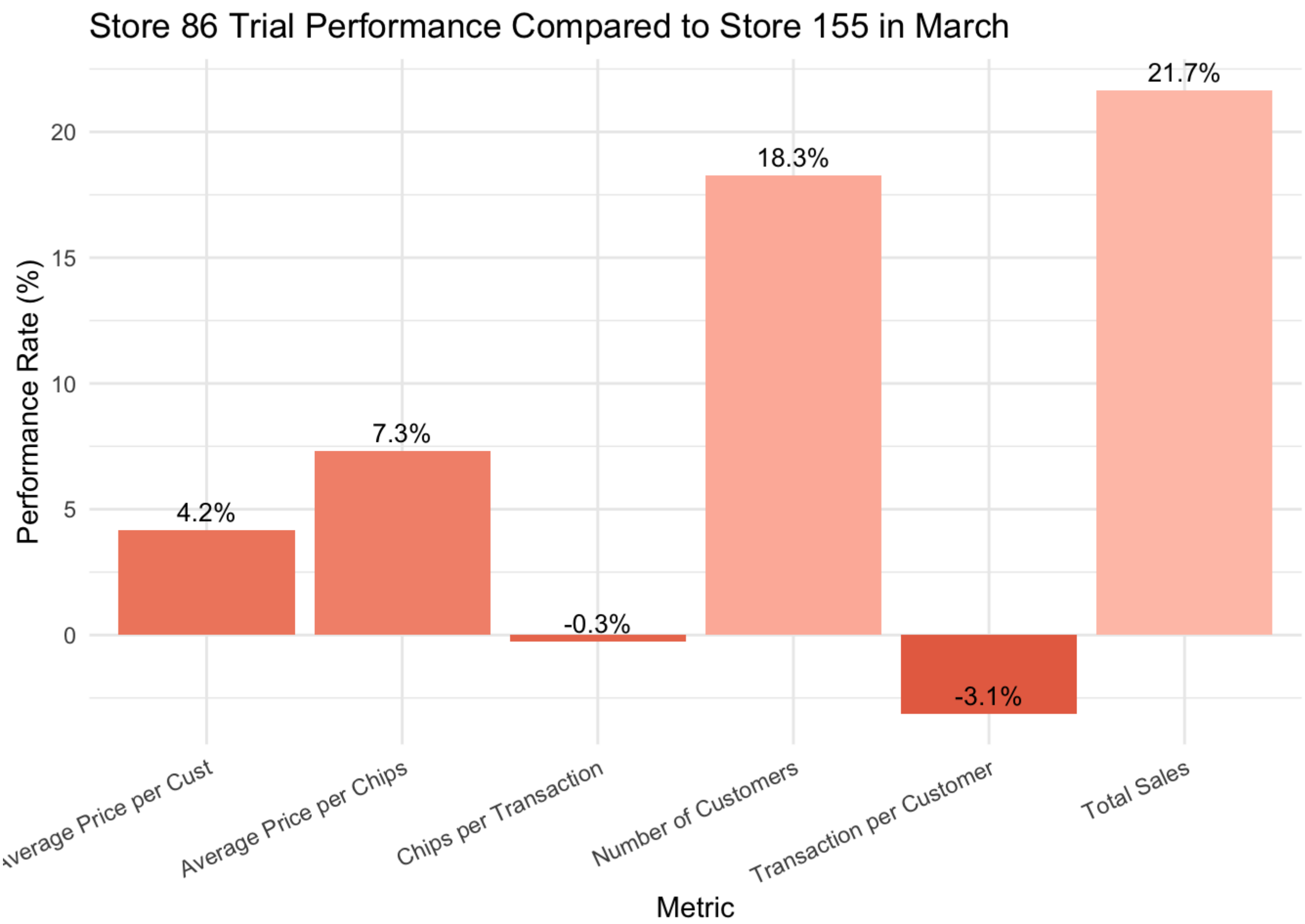
```
          aes(label = paste0(round(Performance_86, 1), "%")),  # Add % symbol to th
e label
          show.legend = FALSE)+
  theme(axis.text.x = element_text(hjust = 1)) +
   scale_x_discrete(labels = c("Average Price per Cust", "Average Price per Chips",
"Chips per Transaction", "Number of Customers", "Transaction per Customer", "Total Sa
les"))
```



Get the grouped average of each column data by stores in March

```
assembled_march_86_155 = filtered_data %>%
  filter(YEARMONTH == 201903 & STORE_NBR == 86 | YEARMONTH == 201903 & STORE_NBR == 1
55)

assembled_march_86_155 <- select(assembled_march_86_155, -c(numberCustomers.x, number
Customers.y, Store_type, TransactionMonth, YEARMONTH))

assembled_march_86_155$avgPricePerCust = assembled_march_86_155$totSales/assembled_ma
rch_86_155$nCustomers



assembled_march_86_155_restructured <- pivot_longer(assembled_march_86_155, cols = -c
(STORE_NBR), names_to = "Metric", values_to = "Value")

# Filter rows
desired_rows_march <- c("totSales", "nTxnPerCust", "nChipsPerTxn", "avgPricePerUnit",
"avgPricePerCust", "nCustomers")
assembled_march_86_155_restructured <- assembled_march_86_155_restructured %>% filter
(Metric %in% desired_rows_march)

# Add a new column for the store-specific column names
assembled_march_86_155_restructured$STORE_NBR <- paste("STORE_NBR_", assembled_march_
86_155_restructured$STORE_NBR, sep = "")

# Pivot again to get the desired structure
final_assembled_march_86_155_ <- pivot_wider(assembled_march_86_155_restructured, nam
es_from = STORE_NBR, values_from = Value)


final_86_155_march <- final_assembled_march_86_155_





final_86_155_march <- final_86_155_march %>%
  mutate(Performance_86 = ((`STORE_NBR_86` - `STORE_NBR_155`) / `STORE_NBR_86`)*100)

final_86_155_march = final_86_155_march %>%
  arrange(Metric)
# make a chart
ggplot(final_86_155_march, aes(x = Metric, y = Performance_86, fill = Performance_8
6)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Store 86 Trial Performance Compared to Store 155 in March",
       x = "Metric",
       y = "Performance Rate (%)") +
  theme_minimal() +
  guides(fill = FALSE) + # Remove the legend
 theme(axis.text.x = element_text(angle = 25, hjust = 0))  +
    scale_fill_gradient(low = "#e76f51", high = "#ffc4b5") +
    geom_text(position = position_dodge(width = 0.9),    # Adjust the position as nee
ded
            vjust = -0.5,                              # Adjust the vertical position
            size = 3.5,                                  # Adjust the text size
            color = "black",                           # Text color
```

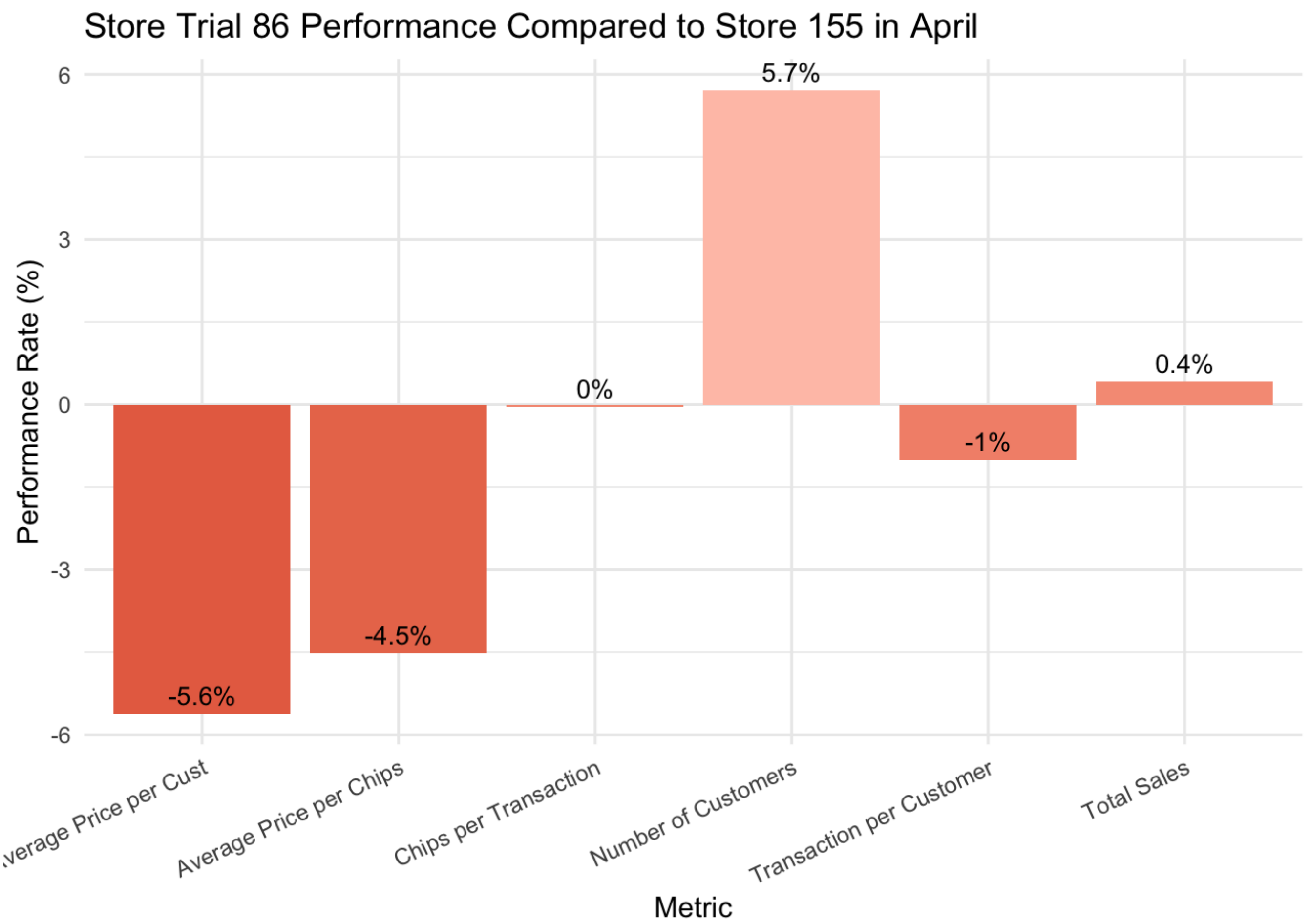
```
            aes(label = paste0(round(Performance_86, 1), "%")),# Add % symbol to the
label
            show.legend = FALSE)+
  theme(axis.text.x = element_text(hjust = 1)) +
   scale_x_discrete(labels = c("Average Price per Cust", "Average Price per Chips",
"Chips per Transaction", "Number of Customers", "Transaction per Customer", "Total Sa
les"))
```



Get the grouped average of each column data by stores in April

```
assembled_april_86_155 = filtered_data %>%
  filter(YEARMONTH == 201904 & STORE_NBR == 86 | YEARMONTH == 201904 & STORE_NBR == 1
55)

assembled_april_86_155 <- select(assembled_april_86_155, -c(numberCustomers.x, number
Customers.y, Store_type, TransactionMonth, YEARMONTH))

assembled_april_86_155$avgPricePerCust = assembled_april_86_155$totSales/assembled_ap
ril_86_155$nCustomers



assembled_april_86_155_restructured <- pivot_longer(assembled_april_86_155, cols = -c
(STORE_NBR), names_to = "Metric", values_to = "Value")

# Filter rows
desired_rows_march <- c("totSales", "nTxnPerCust", "nChipsPerTxn", "avgPricePerUnit",
"avgPricePerCust", "nCustomers")
assembled_april_86_155_restructured <- assembled_april_86_155_restructured %>% filter
(Metric %in% desired_rows_march)

# Add a new column for the store-specific column names
assembled_april_86_155_restructured$STORE_NBR <- paste("STORE_NBR_", assembled_april_
86_155_restructured$STORE_NBR, sep = "")

# Pivot again to get the desired structure
final_assembled_april_86_155_ <- pivot_wider(assembled_april_86_155_restructured, nam
es_from = STORE_NBR, values_from = Value)


final_86_155_april <- final_assembled_april_86_155_






final_86_155_april <- final_86_155_april %>%
  mutate(Performance_86 = ((`STORE_NBR_86` - `STORE_NBR_155`) / `STORE_NBR_86`)*100)

final_86_155_april = final_86_155_april %>%
  arrange(Metric)
# make a chart
ggplot(final_86_155_april, aes(x = Metric, y = Performance_86, fill = Performance_8
6)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Store Trial 86 Performance Compared to Store 155 in April",
       x = "Metric",
       y = "Performance Rate (%)") +
  theme_minimal() +
  guides(fill = FALSE) + # Remove the legend
 theme(axis.text.x = element_text(angle = 25, hjust = 0))  +
    scale_fill_gradient(low = "#e76f51", high = "#ffc4b5") +
    geom_text(position = position_dodge(width = 0.9),    # Adjust the position as nee
ded
            vjust = -0.5,                                # Adjust the vertical position
            size = 3.5,                                    # Adjust the text size
```

```
            color = "black",                           # Text color
            aes(label = paste0(round(Performance_86, 1), "%")),# Add % symbol to the
label
            show.legend = FALSE)+
  theme(axis.text.x = element_text(hjust = 1)) +
   scale_x_discrete(labels = c("Average Price per Cust", "Average Price per Chips",
"Chips per Transaction", "Number of Customers", "Transaction per Customer", "Total Sa
les"))
```



combine the three months sales data (get only performance_86 column from each)

```
performance_86 <- list(final_86_155_feb, final_86_155_march, final_86_155_april)

performance_86 = data.frame(Metric = c("totSales", "nTxnPerCust", "nChipsPerTxn", "av
gPricePerUnit", "avgPricePerCust", "nCustomers"))
combined_86 <- bind_rows(
  mutate(final_86_155_feb, Month = "February"),
  mutate(final_86_155_march, Month = "March"),
  mutate(final_86_155_april, Month = "April")
)
combined_86_wide <- pivot_wider(combined_86, names_from = Month, values_from = Perfor
mance_86)
combined_86_wide <- combined_86_wide %>% filter(Metric %in% desired_rows)


combined_86_wide <- select(combined_86_wide, -c(STORE_NBR_86, STORE_NBR_155))
combined_86_wide <- combined_86_wide %>% mutate_all(~replace(., is.na(.), 0))


feb_86 <- slice(combined_86_wide, 1:6)
march_86 <- slice(combined_86_wide, 7:12)
april_86 <- slice(combined_86_wide, 13:18)


feb_86$March <- march_86$March
feb_86$April <- april_86$April

compare_86_months <- feb_86
compare_86_months
```

Metric <chr>	February <dbl>	March <dbl>	April <dbl>
avgPricePerCust	-9.918159	4.1579292	-5.61045494
avgPricePerUnit	-6.427859	7.2971043	-4.51320346
nChipsPerTxn	-1.233213	-0.2625925	-0.03937008
nCustomers	11.214953	18.2608696	5.71428571
nTxnPerCust	-2.021358	-3.1155015	-1.01010101
totSales	2.409111	21.6595247	0.42442820

6 rows

```
pretrial_86_metric =  measureOverTime %>%
  filter(STORE_NBR == 86, YEARMONTH < 201902) %>%
  select(STORE_NBR, totSales,nCustomers, nTxnPerCust, nChipsPerTxn, avgPricePerUnit)

pretrial_86_metric = pretrial_86_metric %>%
  group_by(STORE_NBR) %>%
  summarise(
     nTxnPerCust = sum(nTxnPerCust)/7,
    nChipsPerTxn = sum(nChipsPerTxn)/7,
    avgPricePerUnit = sum(avgPricePerUnit)/7,
    avgPricePerCust = sum(totSales)/sum(nCustomers),
      totSales = sum(totSales),
    nCustomers =  sum(nCustomers)
  )


# Melt the data frame for easy plotting
pretrial_86_metric_melted = reshape2::melt(pretrial_86_metric, id.vars = "STORE_NBR")
pretrial_86_metric_melted
```

STORE_NBR <int>	variable <fct>	value <dbl>
86	nTxnPerCust	1.256935
86	nChipsPerTxn	2.001391
86	avgPricePerUnit	3.492118
86	avgPricePerCust	6.134947
86	totSales	4276.057957
86	nCustomers	697.000000

6 rows

```
colnames(pretrial_86_metric_melted)[3] <- "value_86"

pretrial_155_metric =  measureOverTime %>%
  filter(STORE_NBR == 155, YEARMONTH < 201902) %>%
  select(STORE_NBR, totSales,nCustomers, nTxnPerCust, nChipsPerTxn, avgPricePerUnit)

pretrial_155_metric = pretrial_155_metric %>%
  group_by(STORE_NBR) %>%
  summarise(
     nTxnPerCust = sum(nTxnPerCust)/7,
    nChipsPerTxn = sum(nChipsPerTxn)/7,
    avgPricePerUnit = sum(avgPricePerUnit)/7,
    avgPricePerCust = sum(totSales)/sum(nCustomers),
      totSales = sum(totSales),
    nCustomers =  sum(nCustomers)
  )


pretrial_155_metric_melted = reshape2::melt(pretrial_155_metric, id.vars = "STORE_NB
R")

colnames(pretrial_155_metric_melted)[3] <- "value_155"
pretrial_155_metric_melted
```

STORE_NBR	variable	value_155
<int>	<fct>	<dbl>
155	nTxnPerCust	1.291266
155	nChipsPerTxn	2.004789
155	avgPricePerUnit	3.495530
155	avgPricePerCust	6.134947
155	totSales	4276.057957
155	nCustomers	697.000000

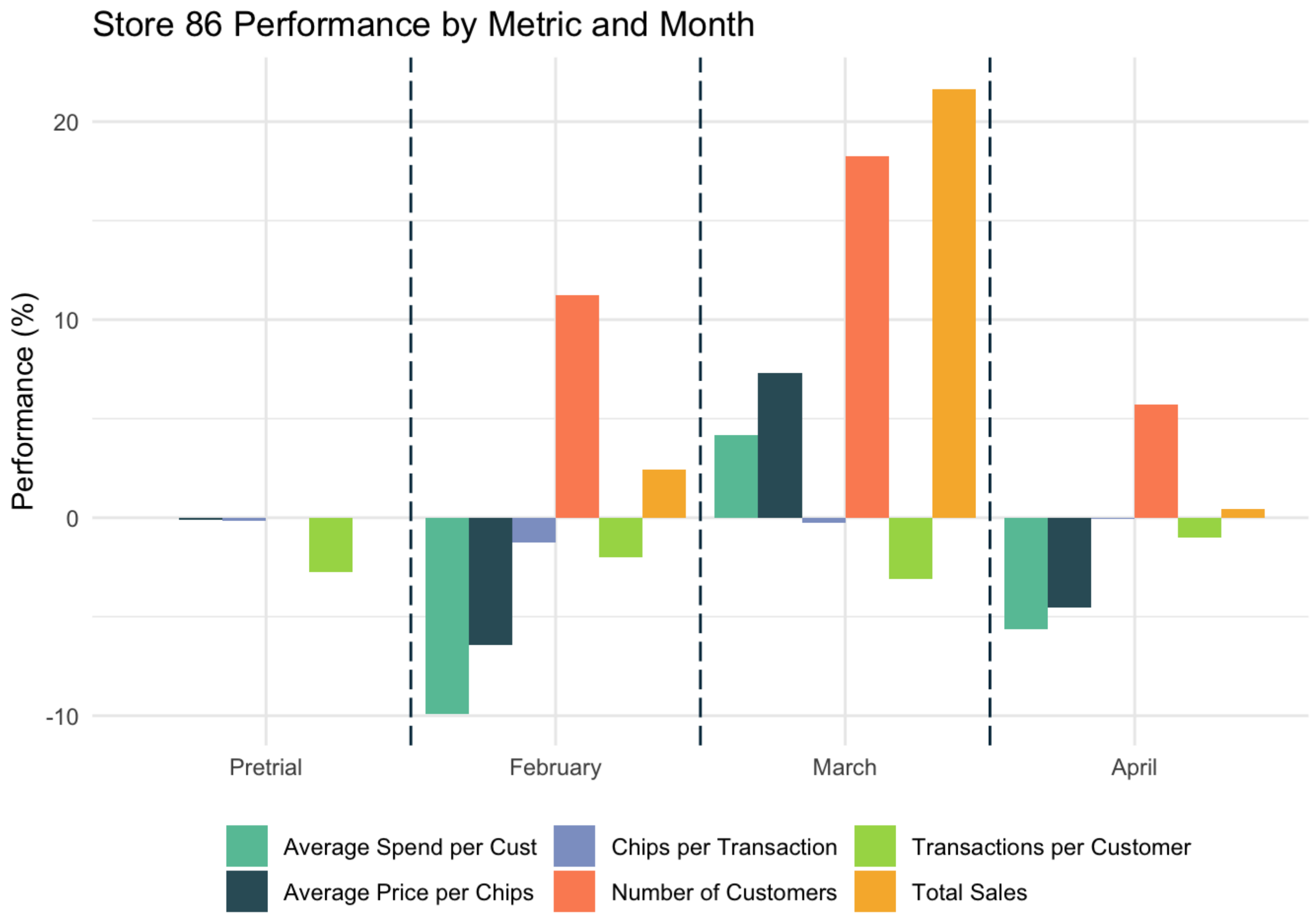
6 rows

```
pretrial_86_metric_melted <- pretrial_86_metric_melted[, c("variable", "value_86")]

pretrial_155_metric_melted <- pretrial_155_metric_melted[, c("variable", "value_15
5")]

pretrial_86_155_metrics <- merge(pretrial_86_metric_melted, pretrial_155_metric_melte
d, by = "variable")

pretrial_86_155_metrics = pretrial_86_155_metrics %>%
  mutate(Performance_86_pretrial = ((`value_86` - `value_155`) / `value_86`)*100)
pretrial_86_155_metrics <- pretrial_86_155_metrics %>% arrange(desc(variable))

names(pretrial_86_155_metrics)[names(pretrial_86_155_metrics) == "variable"] <- "Metr
ic"
names(pretrial_86_155_metrics)[names(pretrial_86_155_metrics) == "Performance_86_pret
rial"] <- "Pretrial"

#pretrial_86_155_metrics <- pretrial_86_155_metrics %>%
#  select(-c(value_86, value_155))
pretrial_86_155_metrics
```

Metric <fct>	value_86 <dbl>	value_155 <dbl>	Pretrial <dbl>
nCustomers	697.000000	697.000000	0.00000000
totSales	4276.057957	4276.057957	0.00000000
avgPricePerCust	6.134947	6.134947	0.00000000
avgPricePerUnit	3.492118	3.495530	-0.09772447
nChipsPerTxn	2.001391	2.004789	-0.16976045
nTxnPerCust	1.256935	1.291266	-2.73131866

6 rows

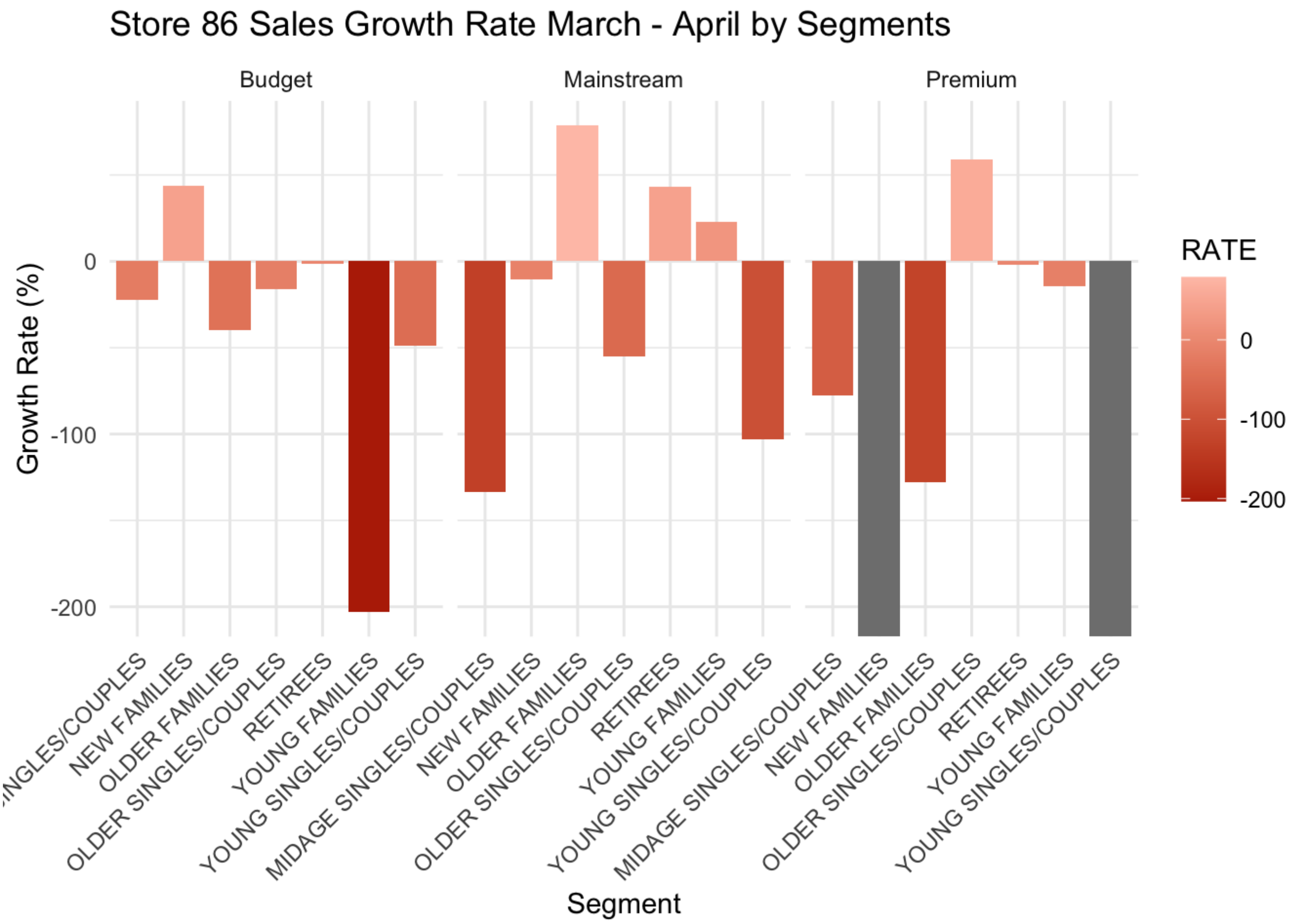
```
compare_86_pretrial_months <- merge(compare_86_months, pretrial_86_155_metrics, by =
"Metric", all = TRUE)


#make a long table for making a chart
compare_86_months_pre_long <- tidyr::pivot_longer(compare_86_pretrial_months, cols =
-Metric, names_to = "Month", values_to = "Performance")



# choose only "Pretrial", "February", "March", "April"

compare_86_months_pre_long = compare_86_months_pre_long %>%
  filter(Month %in% c("Pretrial", "February", "March", "April"))

# arrange the order
compare_86_months_pre_long <- compare_86_months_pre_long %>%
  mutate(Month = factor(Month, levels = c("Pretrial", "February", "March", "April")))
%>%
  arrange(Month)

# Create a grouped bar chart with lines

ggplot(compare_86_months_pre_long, aes(x = Month, y = Performance, fill = Metric)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Store 86 Performance by Metric and Month",
       x = NULL,
       y = "Performance (%)") +
  scale_fill_manual(
    values = c("#66c2a5", "#335c67", "#8da0cb", "#fc8d62", "#a6d854", "#f7b538"),
    name = "Metrics",
    labels = c("Average Spend per Cust", "Average Price per Chips", "Chips per Transa
ction", "Number of Customers", "Transactions per Customer", "Total Sales")
    ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5)) +
  guides(fill = guide_legend(title = NULL)) +
  geom_vline(xintercept = c(1.5, 2.5, 3.5), linetype = "longdash", color = "#023047")
+
  coord_cartesian(clip = 'off') +
  theme(legend.position = "bottom")
```



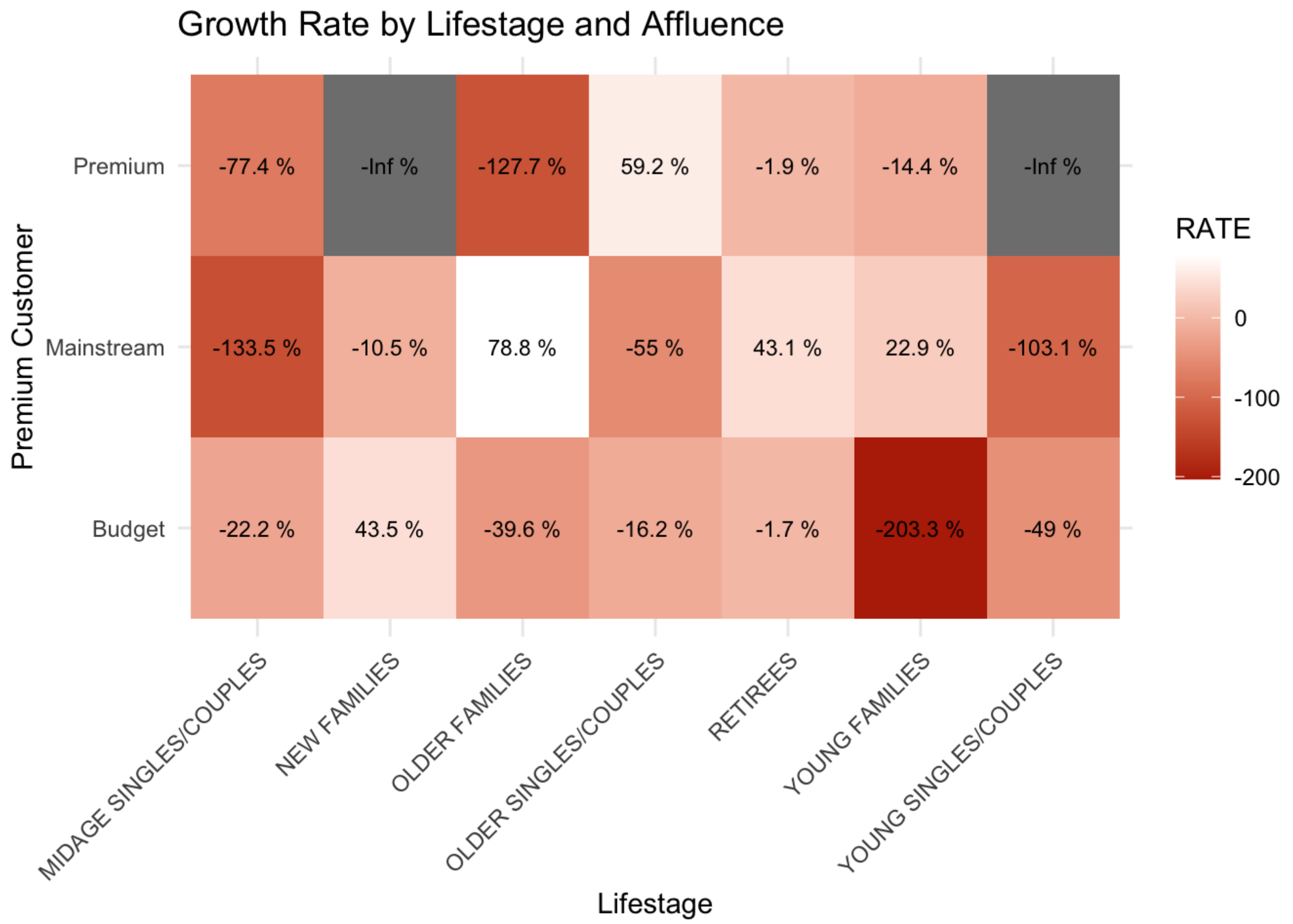
Compare the number of customers of store 86 in pre-trial (average), February, March and April by segments

+Compare the sales of store 86 in March and April by segments

```
# get the sales in March by segments
sales_march_86 <- data %>%
  filter(STORE_NBR == 86, YEARMONTH == 201903 ) %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(SALES = sum(TOT_SALES), .groups = "keep")

# get the sales in April by segments
sales_april_86 <- data %>%
  filter(STORE_NBR == 86, YEARMONTH == 201904 ) %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(SALES = sum(TOT_SALES), .groups = "keep")

# compare the sales between March and April

combined_sales_86 <- full_join(sales_march_86, sales_april_86, by = c("LIFESTAGE", "P
REMIUM_CUSTOMER"))

combined_sales_86 <- combined_sales_86 %>%
  mutate(SALES.y = ifelse(is.na(SALES.y), 0, SALES.y))

combined_sales_86$RATE <- ((combined_sales_86$SALES.y - combined_sales_86$SALES.x) /
combined_sales_86$SALES.y) * 100



#chart option 1: bars


  ggplot(combined_sales_86, aes(x = interaction(LIFESTAGE, PREMIUM_CUSTOMER), y = RAT
E, fill = RATE)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Store 86 Sales Growth Rate March - April by Segments",
       x = "Segment",
       y = "Growth Rate (%)") +
  theme_minimal() +
  facet_wrap(~PREMIUM_CUSTOMER, scales = "free_x", ncol = 3) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_x_discrete(labels = function(x) str_remove(x, "\\..*$")) +
  scale_fill_gradient(low = "#b82b04", high = "#ffc4b5")
```




```
#chart option 2: heatmap

heatmap_86 = ggplot(combined_sales_86, aes(x = LIFESTAGE, y = PREMIUM_CUSTOMER, fill
= RATE)) +
  geom_tile() +
  scale_fill_gradient(low = "#b82b04", high = "white") +
  labs(title = "Growth Rate by Lifestage and Affluence",
       x = "Lifestage",
       y = "Premium Customer") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

heatmap_86 + geom_text(aes(x = LIFESTAGE, y = PREMIUM_CUSTOMER, label = paste(round(.
data[["RATE"]], 1), "%")), size = 3, color = "black")
```



Growth Rate by Lifestage and Affluence
Premium Customer
Premium
Mainstream
Budget
-77.4 %
-Inf %
-127.7 %
59.2 %
-1.9 %
-14.4 %
-Inf %
-133.5 %
-10.5 %
78.8 %
-55 %
43.1 %
22.9 %
-103.1 %
-22.2 %
43.5 %
-39.6 %
-16.2 %
-1.7 %
-203.3 %
-49 %
MIDAGE SINGLES/COUPLES
NEW FAMILIES
OLDER FAMILIES
OLDER SINGLES/COUPLES
RETIREES
YOUNG FAMILIES
YOUNG SINGLES/COUPLES
Lifestage
RATE
0
-100
-200