websale Client Project

#Setup Loading packages and data

```
setwd("~/Websale_Technical_Exercise")

#load packages
library(pacman)
```

```
## Warning: package 'pacman' was built under R version 4.0.5
```

```
pacman::p_load(pacman,tidyverse, openxlsx, corrplot, lubridate, Rcpp, ggThemeAssist, ggthemes)

#load spreadsheets
addsToCart<- read.csv("DataAnalyst_Ecom_data_addsToCart.csv")
sessionCounts<- read.csv("DataAnalyst_Ecom_data_sessionCounts.csv")
```

Cleaning Data

Finding discrepancies in dataset and plotting them

```
#format dim_date to date type
sessionCounts$date<- as.Date(sessionCounts$dim_date, "%m/%d/%y")

#checking for instances with zero transactions but QTY over 1
sessionCounts%>%
  filter(transactions==0 & QTY>0)%>%
  summarise(n=n())
```

```
##     n
## 1 160
```

```
#checking for instances with more transactions than QTY
sessionCounts%>%
  filter(transactions>QTY)%>%
  summarise(n=n())
```

```
##     n
## 1 580
```

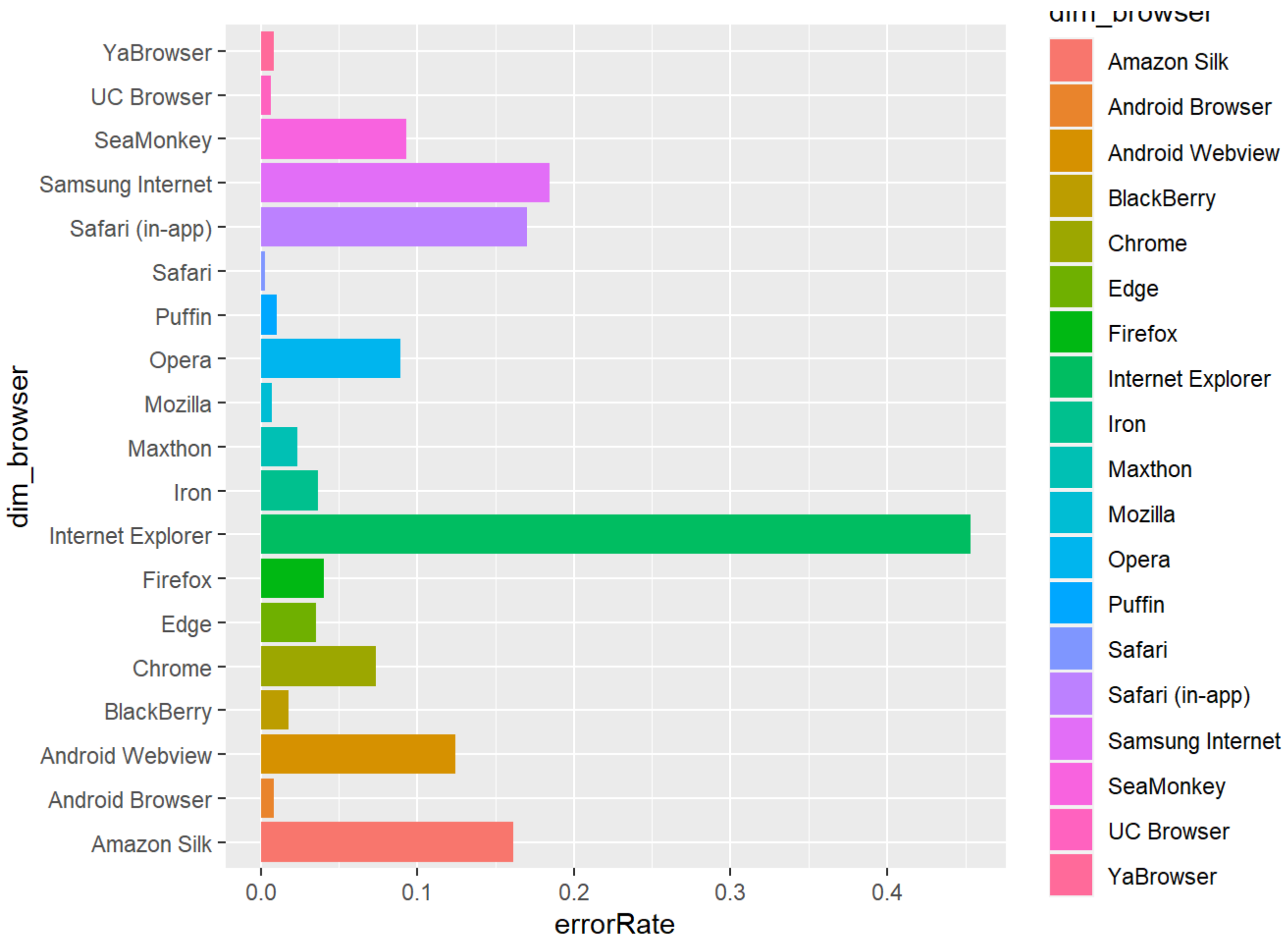
```
#checking for instances with more transactions than sessions
sessionCounts%>%
  filter(transactions>sessions)%>%
  summarise(n=n())
```

```
##   n
## 1 5
```

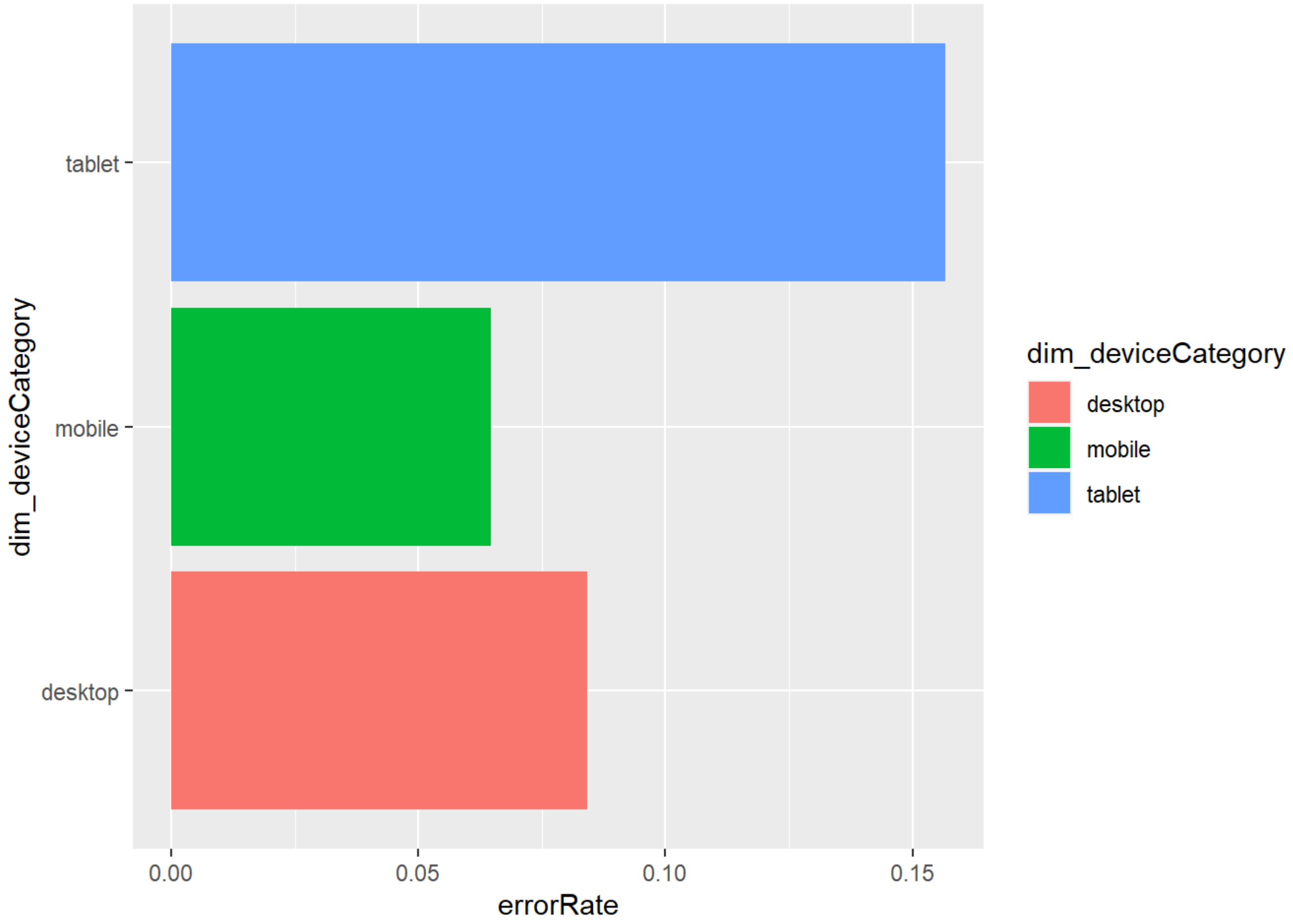
```
#checking for instances with zero sessions but transactions over 1
sessionCounts%>%
  filter(sessions==0 & QTY>0)%>%
  summarise(n=n())
```

```
##   n
## 1 4
```

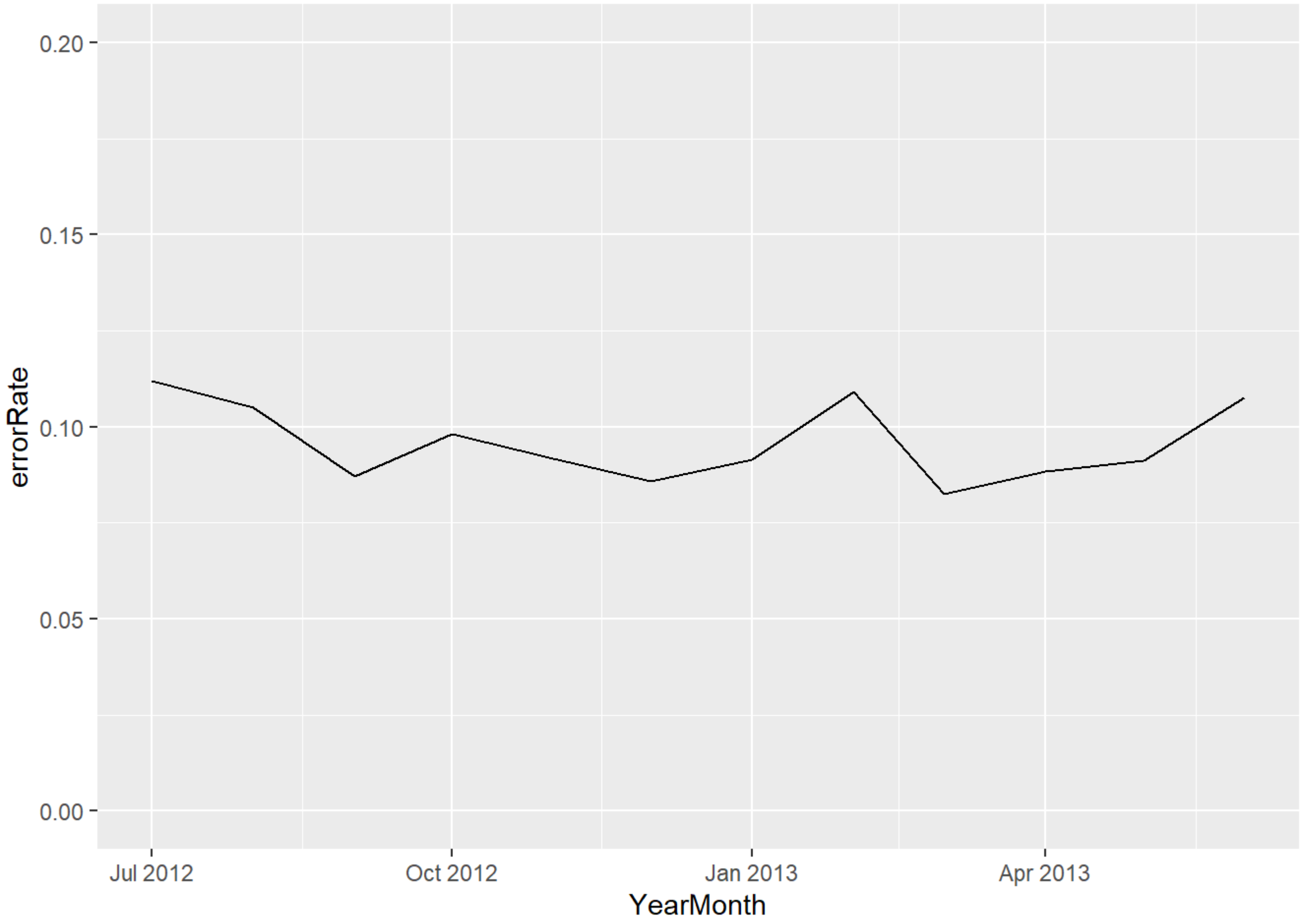
```
#creating column error to track distribution
sessionError<- sessionCounts%>%
  mutate(error=if_else((transactions==0 & QTY>0)|
                         (transactions>QTY)|
                         (transactions>sessions)|
                         (sessions==0 & QTY>0),
                         1, 0))

#plot of error rate by browser
sessionError%>%
  group_by(dim_browser)%>%
  summarise(errorRate=mean(error))%>%
  arrange(desc(errorRate))%>%
  #removing all browsers with zero errors, to many with
  filter(errorRate>0)%>%
  ggplot(aes(x=dim_browser, y=errorRate, fill=dim_browser))+
  geom_bar(stat = 'identity')+coord_flip()
```



```
#plot of error rate by device type
sessionError%>%
  group_by(dim_deviceCategory)%>%
  summarise(errorRate=mean(error))%>%
  ggplot(aes(x=dim_deviceCategory, y=errorRate, fill=dim_deviceCategory))+
  geom_bar(stat = 'identity')+coord_flip()
```



```
#plot of error rate over time
sessionError%>%
  #create YearMonth so error rate can be grouped by month
  mutate(YearMonth=floor_date(date,'month'))%>%
  group_by(YearMonth)%>%
  summarise(errorRate=mean(error))%>%
  ggplot(aes(x=YearMonth, y=errorRate))+
    geom_line()+
    coord_cartesian(ylim = c(0,.2))
```



#Creating Month * Device Data

```
#create Month * Device data
groupedCounts<- sessionError%>%
  rename(DeviceType = dim_deviceCategory)%>%
  #filter out errors
  filter(error==0)%>%
  #create column YearMonth that rounds each date down to the first of that month
  mutate(YearMonth=floor_date(date,'month'))%>%
  #group by device type and YearMonth
  group_by(DeviceType, YearMonth)%>%
  #remove unwanted columns
  select(-dim_browser, -dim_date, -date, -error)%>%
  #summarize all remaining columns that are not being grouped
  summarise(across(everything(), sum))%>%
  #create ECR column
  mutate(ECR=transactions/sessions,
  #create Average Quantity (AQ)
  AQ=QTY/transactions
         )
```

```
## `summarise()` has grouped output by 'DeviceType'. You can override using the
## `.groups` argument.
```

#Summary Statistics

```
#summary statistics
summary(groupedCounts)
```

```
##   DeviceType          YearMonth             sessions       transactions
##  Length:36          Min.   :2012-07-01   Min.   : 99933   Min.   : 1926
##  Class :character   1st Qu.:2012-09-23   1st Qu.:189799   1st Qu.: 3079
##  Mode  :character   Median :2012-12-16   Median :264353   Median : 4669
##                     Mean   :2012-12-16   Mean   :276858   Mean   : 6634
##                     3rd Qu.:2013-03-08   3rd Qu.:318521   3rd Qu.: 9259
##                     Max.   :2013-06-01   Max.   :528567   Max.   :18206
##       QTY             ECR                 AQ
##  Min.   : 3369   Min.   :0.009682   Min.   :1.682
##  1st Qu.: 5674   1st Qu.:0.013249   1st Qu.:1.812
##  Median : 8614   Median :0.022739   Median :1.854
##  Mean   :12430   Mean   :0.023122   Mean   :1.859
##  3rd Qu.:17768   3rd Qu.:0.032554   3rd Qu.:1.913
##  Max.   :34791   Max.   :0.039280   Max.   :2.029
```

```
#summary statistics for each device
groupedCounts %>%
  select(-YearMonth)%>%
  split(.$DeviceType) %>%
  map(summary)
```

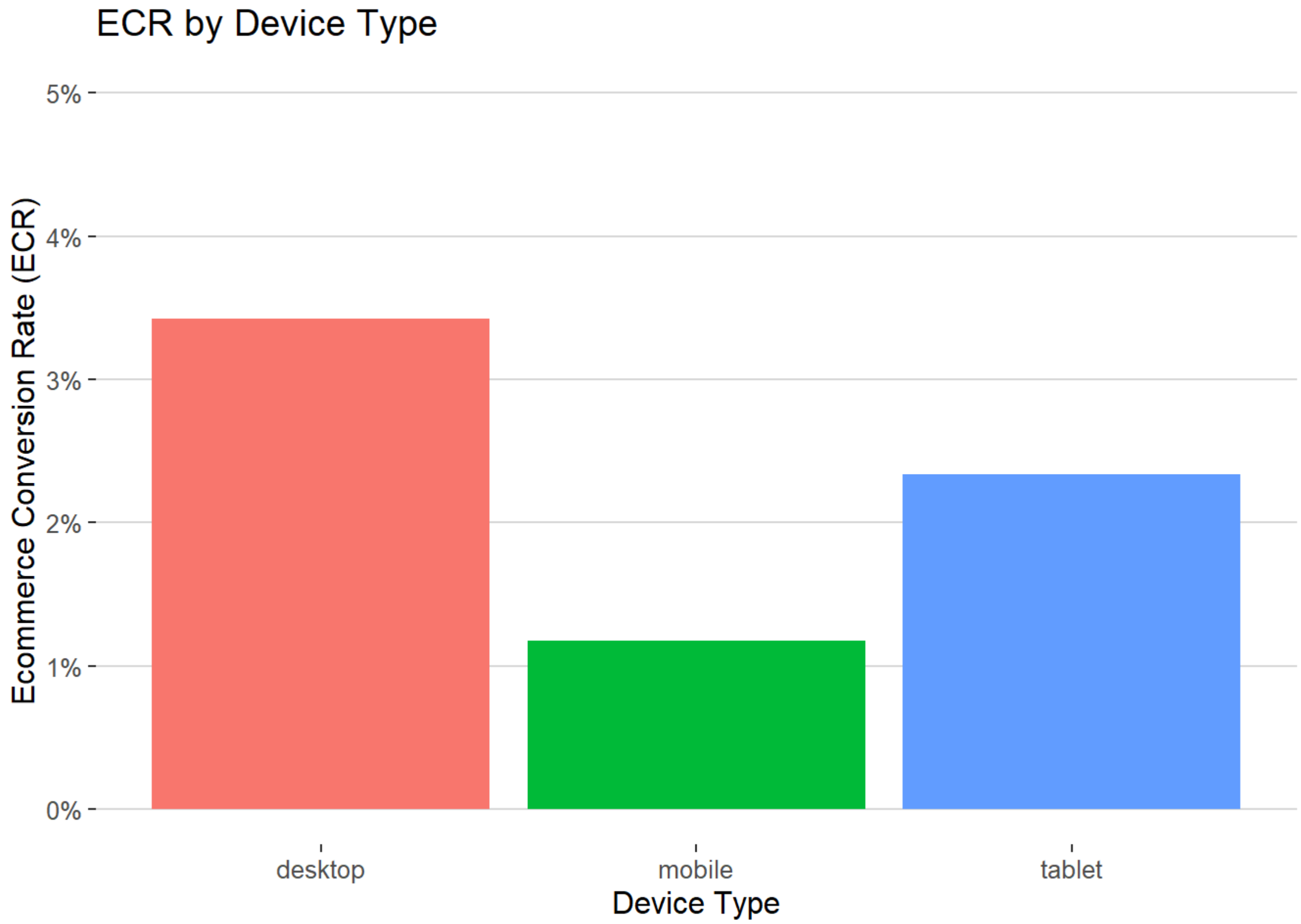
```
## $desktop
##   DeviceType           sessions       transactions        QTY
##  Length:12          Min.   :239867   Min.   : 8345   Min.   :16441
##  Class :character   1st Qu.:277522   1st Qu.: 9368   1st Qu.:18173
##  Mode  :character   Median :308368   Median :10512   Median :19235
##                     Mean   :353988   Mean   :12107   Mean   :23049
##                     3rd Qu.:408735   3rd Qu.:14272   3rd Qu.:27277
##                     Max.   :528567   Max.   :18206   Max.   :34791
##       ECR                AQ
##  Min.   :0.03138   Min.   :1.785
##  1st Qu.:0.03263   1st Qu.:1.887
##  Median :0.03385   Median :1.912
##  Mean   :0.03426   Mean   :1.905
##  3rd Qu.:0.03492   3rd Qu.:1.927
##  Max.   :0.03928   Max.   :1.990
##
## $mobile
##   DeviceType           sessions       transactions       QTY
##  Length:12          Min.   :171881   Min.   :1926   Min.   : 3369
##  Class :character   1st Qu.:222073   1st Qu.:2357   1st Qu.: 4307
##  Mode  :character   Median :264935   Median :3078   Median : 5551
##                     Mean   :293081   Mean   :3504   Mean   : 6266
##                     3rd Qu.:345708   3rd Qu.:4222   3rd Qu.: 7296
##                     Max.   :516679   Max.   :7347   Max.   :12948
##       ECR                 AQ
##  Min.   :0.009682   Min.   :1.682
##  1st Qu.:0.010653   1st Qu.:1.759
##  Median :0.011316   Median :1.787
##  Mean   :0.011712   Mean   :1.796
##  3rd Qu.:0.013107   3rd Qu.:1.820
##  Max.   :0.014220   Max.   :1.945
##
## $tablet
##   DeviceType           sessions       transactions       QTY
##  Length:12          Min.   : 99933   Min.   :2259   Min.   : 4449
##  Class :character   1st Qu.:144161   1st Qu.:3087   1st Qu.: 5802
##  Mode  :character   Median :162122   Median :4296   Median : 8024
##                     Mean   :183504   Mean   :4292   Mean   : 7975
##                     3rd Qu.:226846   3rd Qu.:4806   3rd Qu.: 8773
##                     Max.   :297765   Max.   :7523   Max.   :13614
##       ECR                AQ
##  Min.   :0.02057   Min.   :1.810
##  1st Qu.:0.02075   1st Qu.:1.836
##  Median :0.02274   Median :1.850
##  Mean   :0.02339   Mean   :1.875
##  3rd Qu.:0.02433   3rd Qu.:1.896
##  Max.   :0.03102   Max.   :2.029
```

```
#ECR is different for each device, desktop highest
```

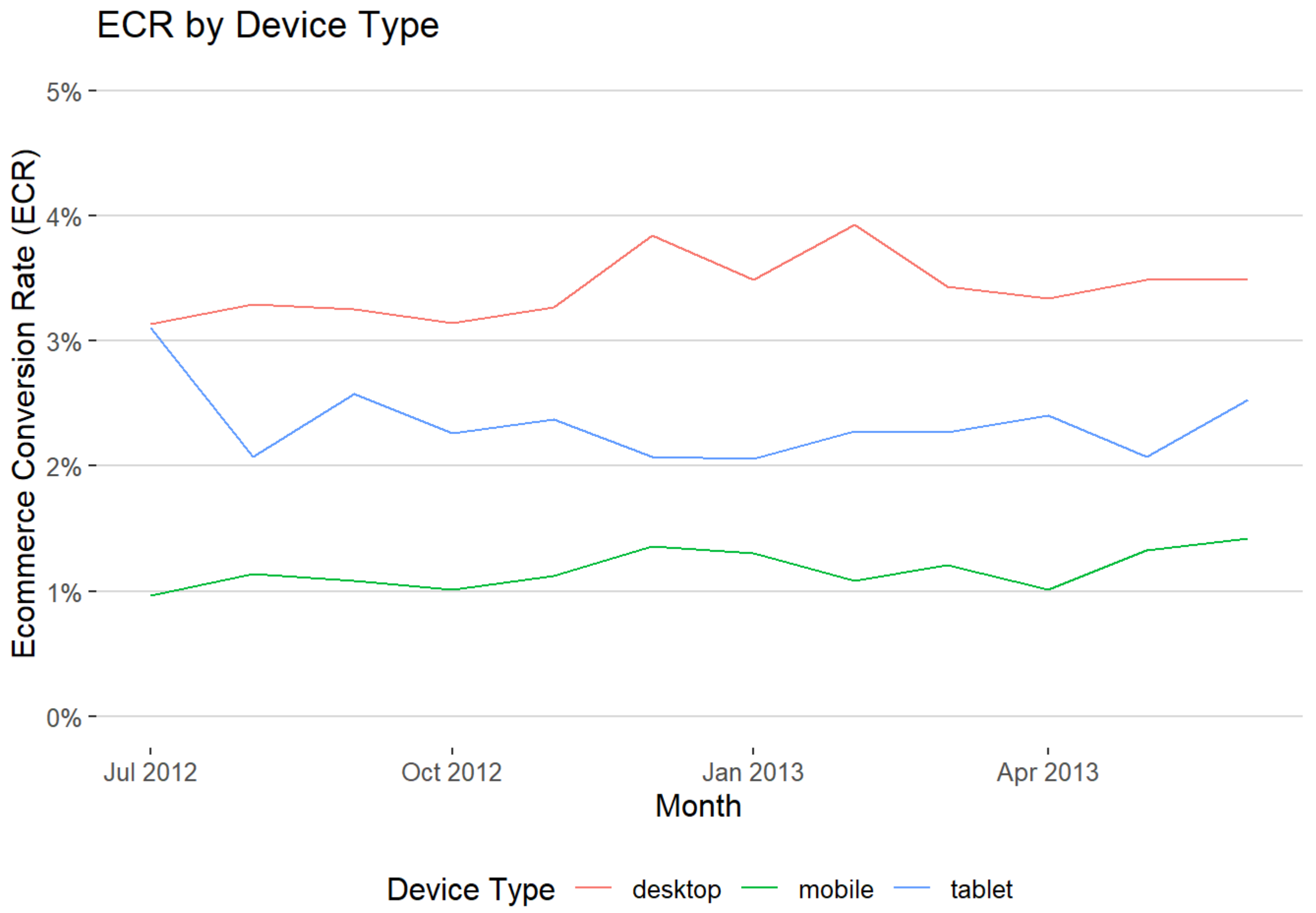
#Plots

```
##barplot of ECR by device type
ggplot(data=groupedCounts, aes(x=DeviceType, y= ECR, fill=DeviceType))+
  geom_bar(stat = "summary")+
  scale_y_continuous(labels = scales::percent_format(accuracy = 1))+
  coord_cartesian(ylim = c(0,.05))+
  theme_hc()+
  labs(title = "ECR by Device Type", x="Device Type", y="Ecommerce Conversion Rate (ECR)")+
  theme(legend.position="none")
```

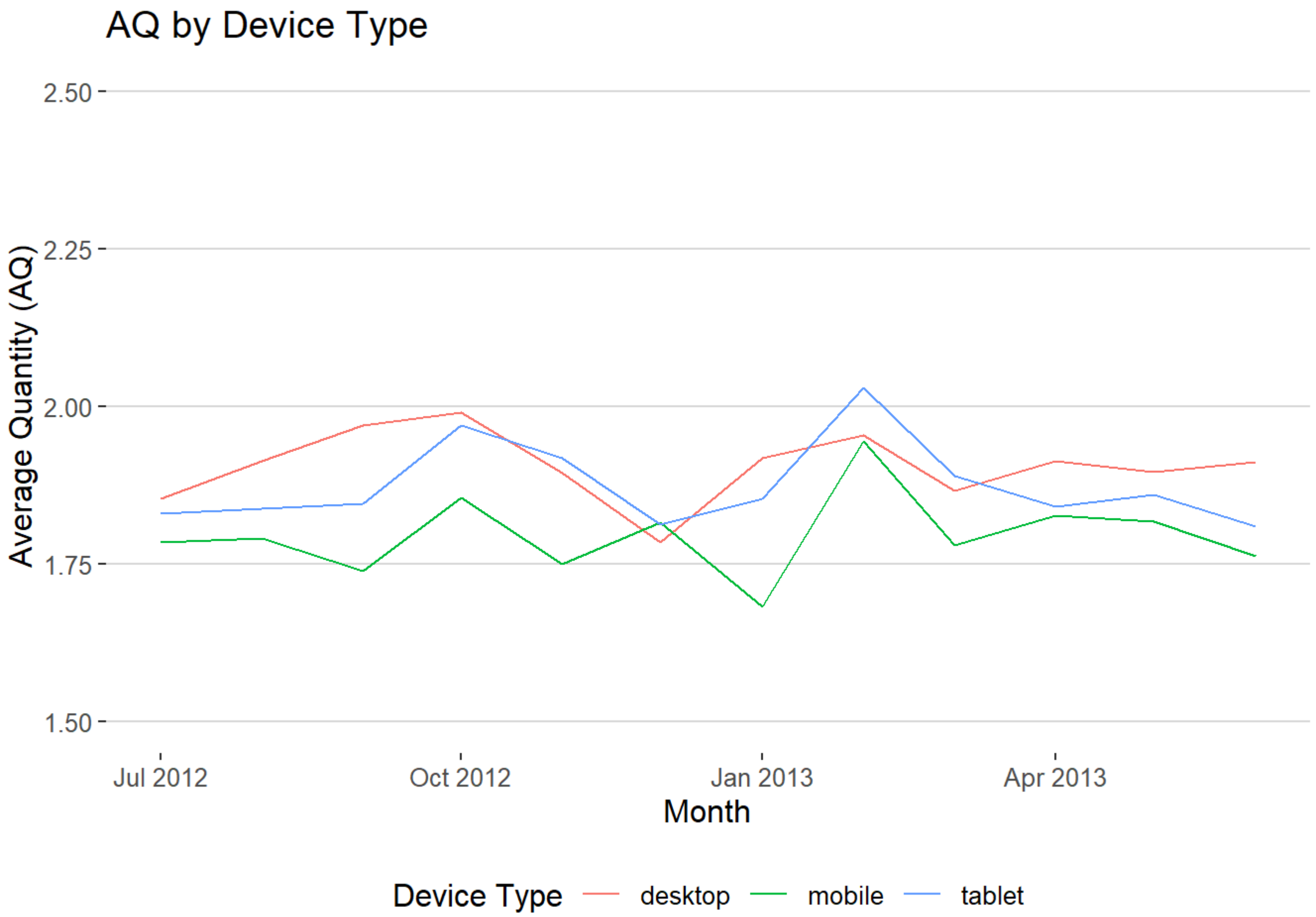
```
## No summary function supplied, defaulting to `mean_se()`
```



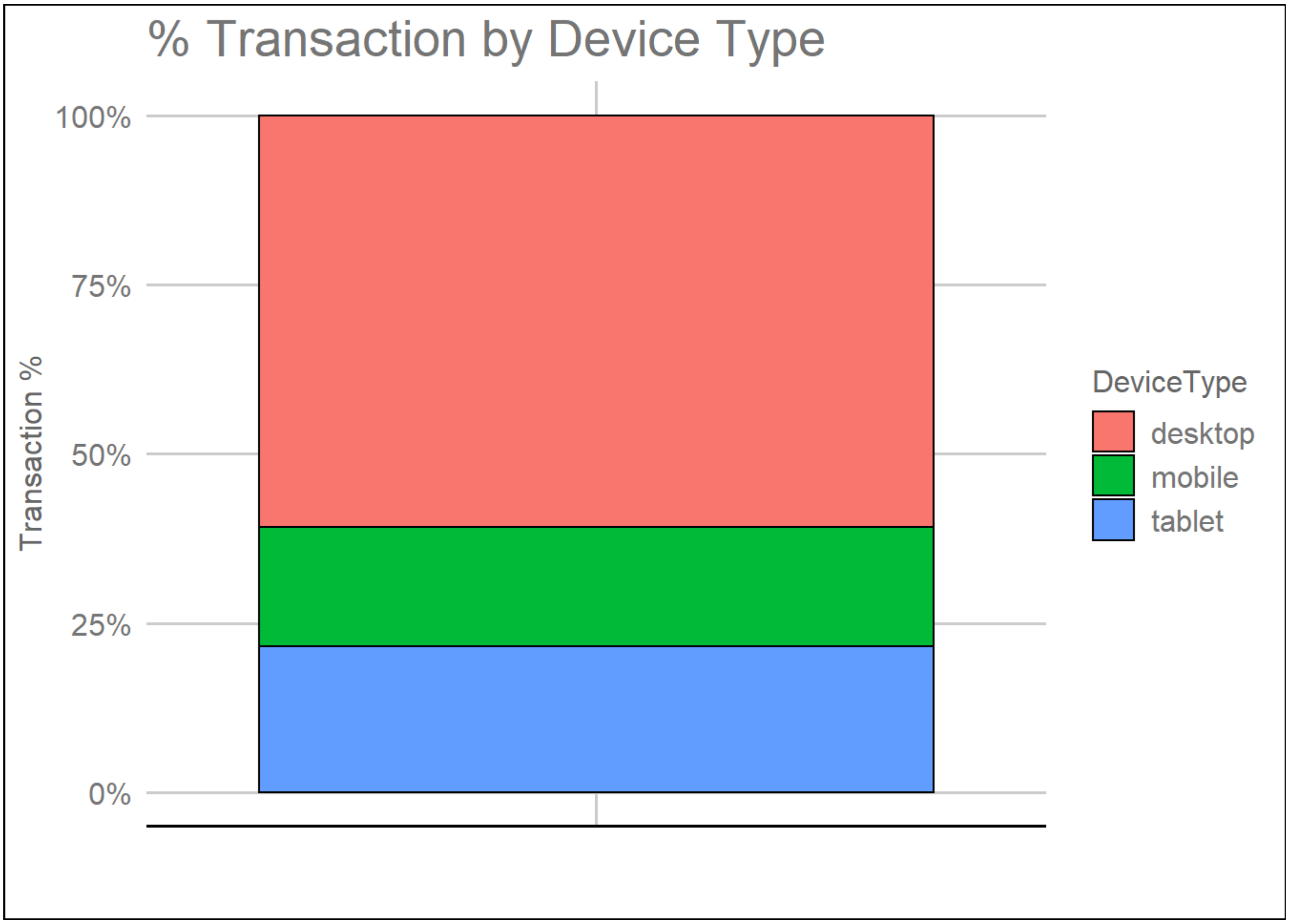
```
##plot of Devices ECR by Month
ggplot(data = groupedCounts, aes(x=YearMonth, y=ECR))+
  geom_line(aes(group=DeviceType, color= DeviceType))+
  scale_y_continuous(labels = scales::percent_format(accuracy = 1))+
  coord_cartesian(ylim = c(0,.05))+
  theme_hc()+
  labs(title = "ECR by Device Type", x="Month", y="Ecommerce Conversion Rate (ECR)", color= "Dev
ice Type")
```



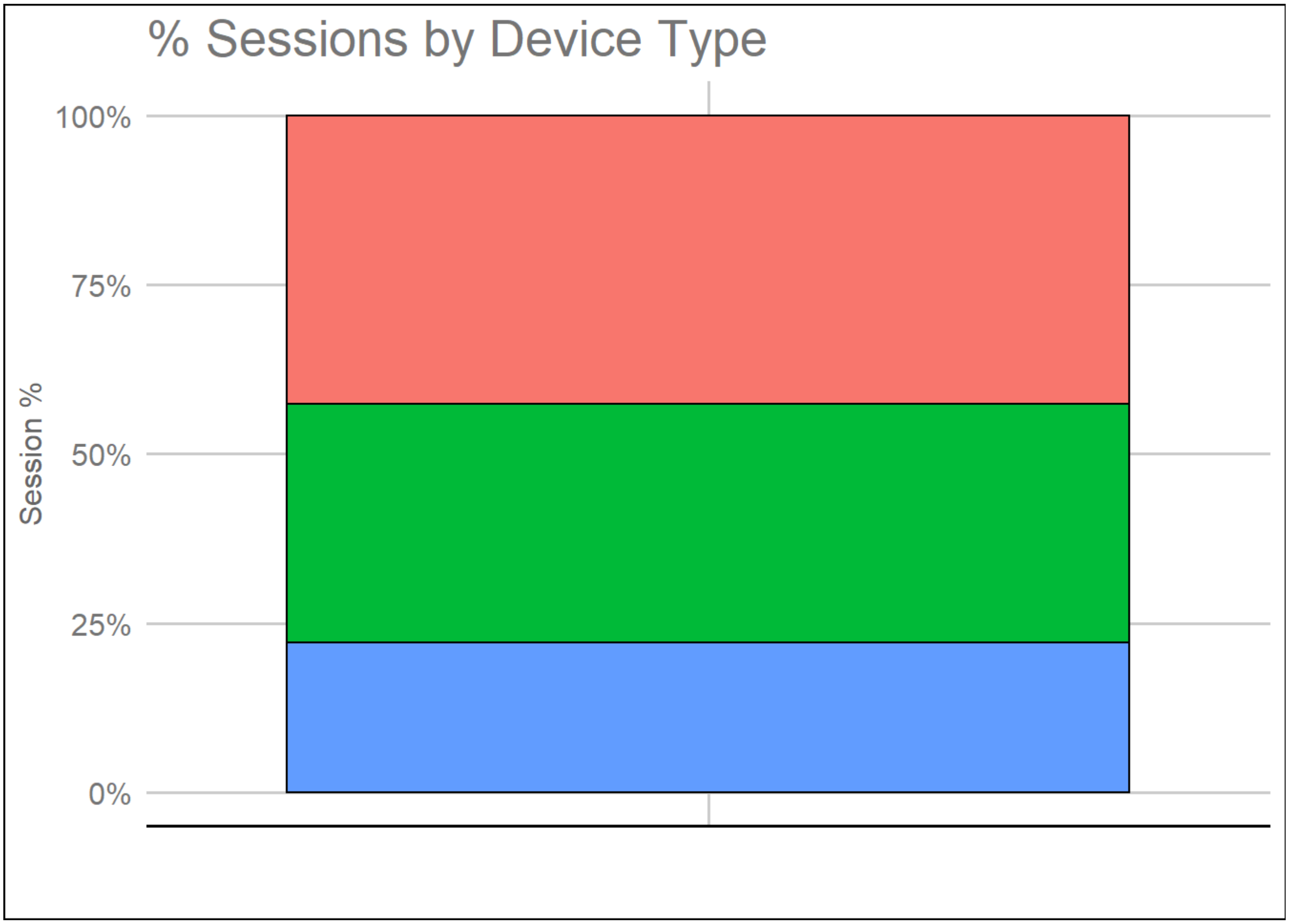
```
#Plot of AQ for Each Device by Month
ggplot(data = groupedCounts, aes(x=YearMonth, y=AQ))+
  geom_line(aes(group=DeviceType, color= DeviceType))+
  coord_cartesian(ylim = c(1.5,2.5))+
  theme_hc()+
  labs(title = "AQ by Device Type", x="Month", y="Average Quantity (AQ)", color= "Device Type")
```



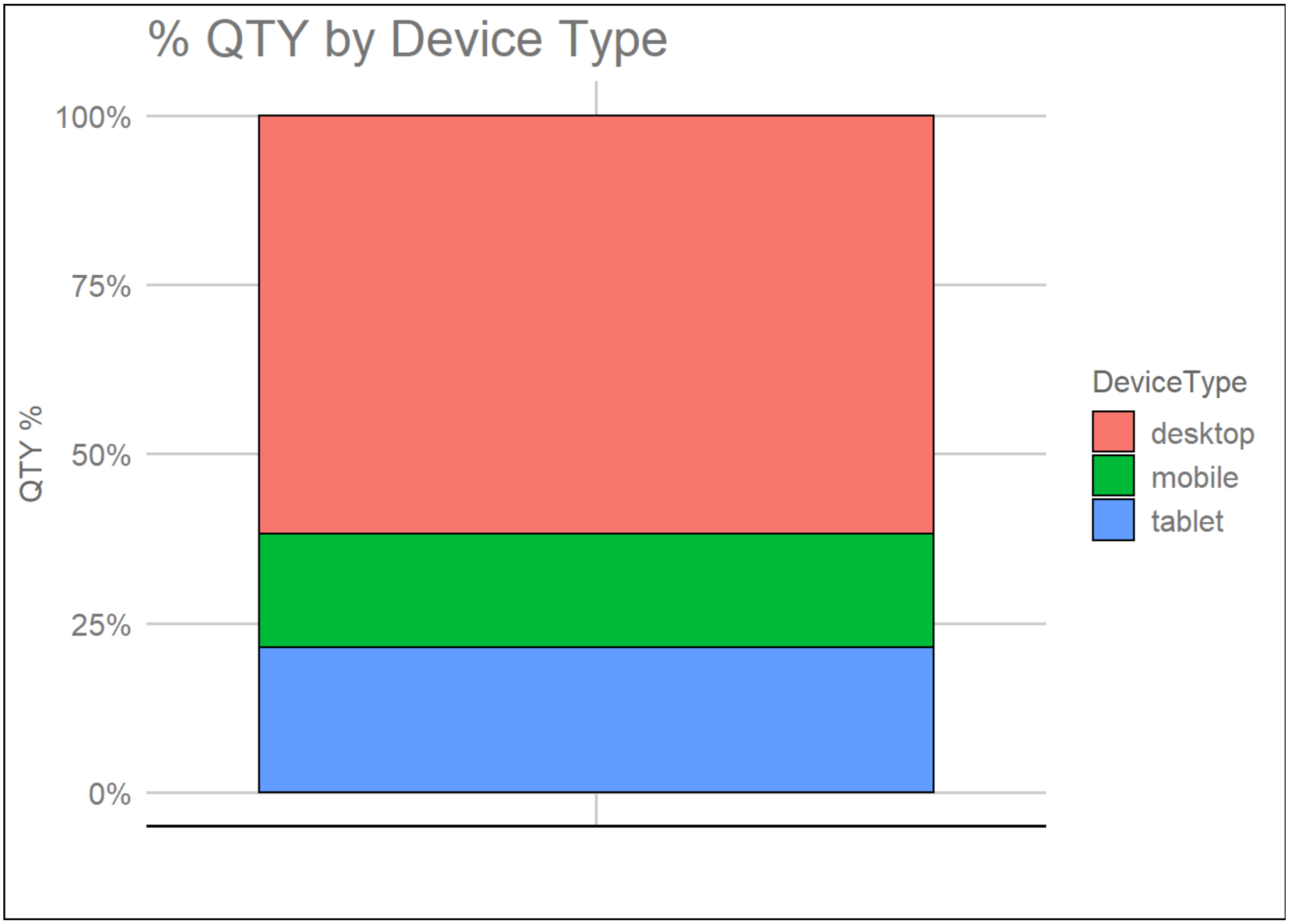
```
#Manipulating data to see percent of sessions, transactions, and QTY each
##device responsible for
groupedPerc<-groupedCounts%>%
  group_by(DeviceType)%>%
  select(-YearMonth, -ECR, -AQ)%>%
  summarise(across(everything(), sum))%>%
  mutate(PercTransaction=transactions/sum(transactions),
         PercSession=sessions/sum(sessions),
         PercQTY=QTY/sum(QTY))%>%
  select(-sessions, -transactions, -QTY)

#collumn plot of Transaction percent by Device
ggplot(data= groupedPerc, aes(x = "", y = PercTransaction, fill = DeviceType)) +
  geom_col(color = "black")+
  ##making scale percent
  scale_y_continuous(labels = scales::percent_format(accuracy = 1))+
  theme_gdocs()+
  labs(title = "% Transaction by Device Type", x="", y="Transaction %", color= "Device Type")
```



```
#collumn plot of Session percent by Device
ggplot(data= groupedPerc, aes(x = "", y = PercSession, fill = DeviceType)) +
  geom_col(color = "black")+
  scale_y_continuous(labels = scales::percent_format(accuracy = 1))+
  theme_gdocs()+
  theme(legend.position="none")+
  labs(title = "% Sessions by Device Type", x="", y="Session %", color= "Device Type")
```



```
#collumn plot of QTY percent by Device
ggplot(data= groupedPerc, aes(x = "", y = PercQTY, fill = DeviceType, label=PercQTY)) +
  geom_col(color = "black")+
  scale_y_continuous(labels = scales::percent_format(accuracy = 1))+
  theme_gdocs()+
  labs(title = "% QTY by Device Type", x="", y="QTY %", color= "Device Type")
```



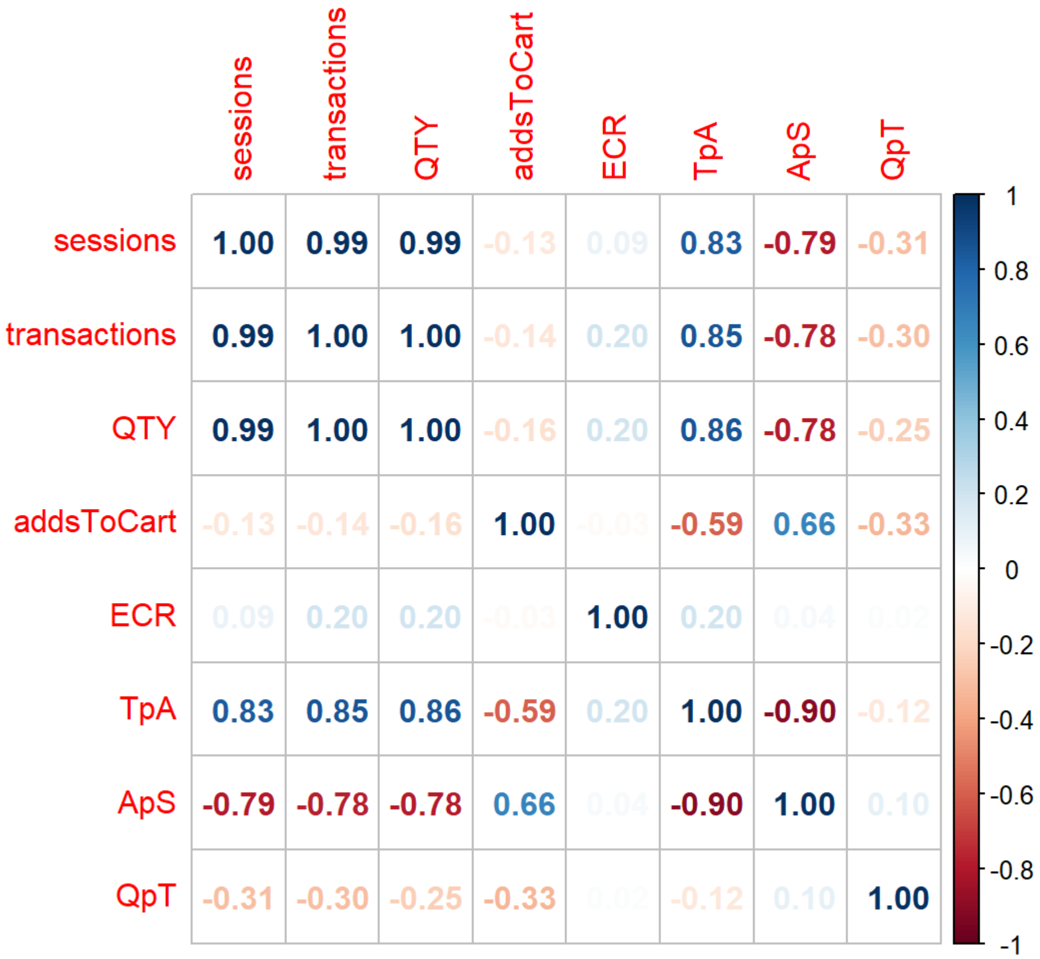
#Joining Datasets

```
#format date using year and month for addsToCart
addsToCart$YearMonth<-as.Date(with(addsToCart,paste(dim_year,dim_month,1,sep="-")),"%Y-%m-%d")

MonthJoin<- sessionCounts%>%
  #create column YearMonth that rounds each date down to the first of that month
  mutate(YearMonth=floor_date(date,'month'))%>%
  #group by device type and YearMonth
  group_by(YearMonth)%>%
  #remove unwanted columns
  select(-dim_browser, -dim_date, -date, -dim_deviceCategory)%>%
  #summarize all remaining columns that are not being grouped
  summarise(across(everything(), sum))%>%
  inner_join(addsToCart)%>%
  select(-dim_year, -dim_month)%>%
  arrange(YearMonth)%>%
  mutate(ECR=transactions/sessions, TpA=transactions/addsToCart,     ApS=addsToCart/sessions, Qp
T=QTY/transactions)
```

```
## Joining, by = "YearMonth"
```

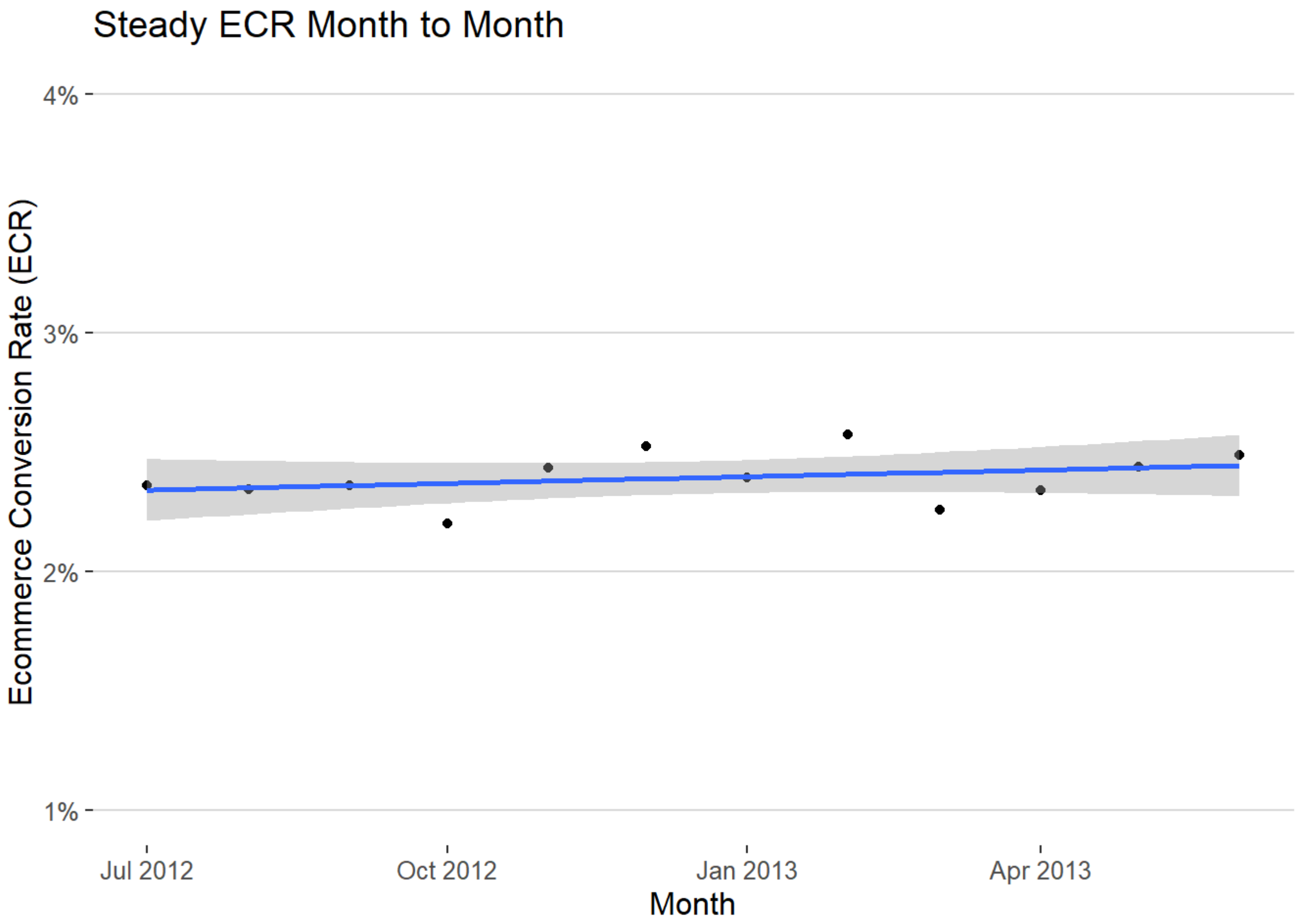
#Plots with Merged Data

```
#correlation plot
corrplot(cor(MonthJoin%>% select(-YearMonth)), method='number')
```



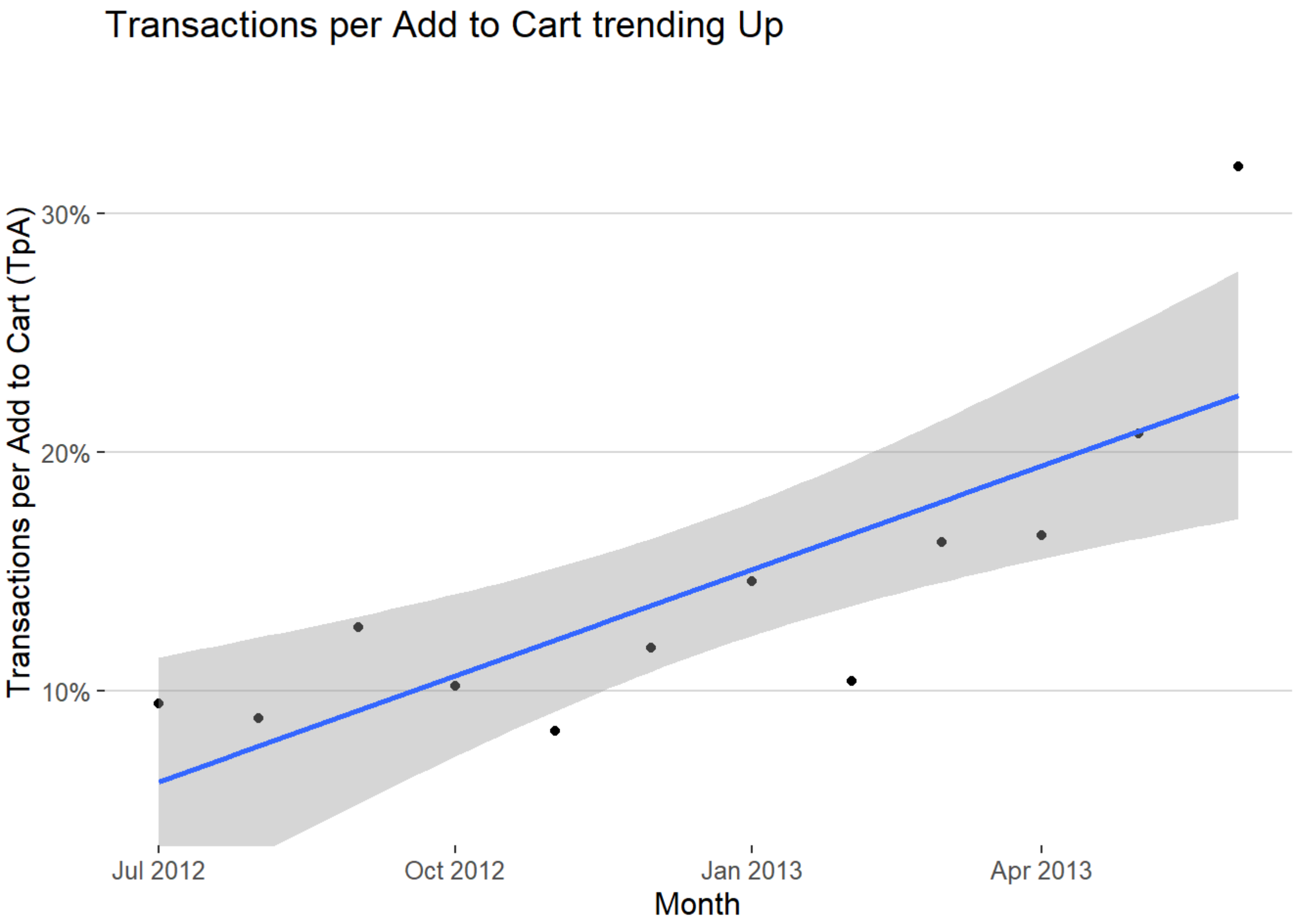
```
#scatterplot of ECR by month
ggplot(data=MonthJoin, aes(x=YearMonth, y= ECR))+
  geom_point()+
  geom_smooth(method="lm")+
  #formating y axis limits
  coord_cartesian(ylim = c(.01,.04))+
  #making y axis pecent
  scale_y_continuous(labels = scales::percent_format(accuracy = 1))+
  theme_hc()+
  labs(title = "Steady ECR Month to Month", x="Month", y="Ecommerce Conversion Rate (ECR)")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



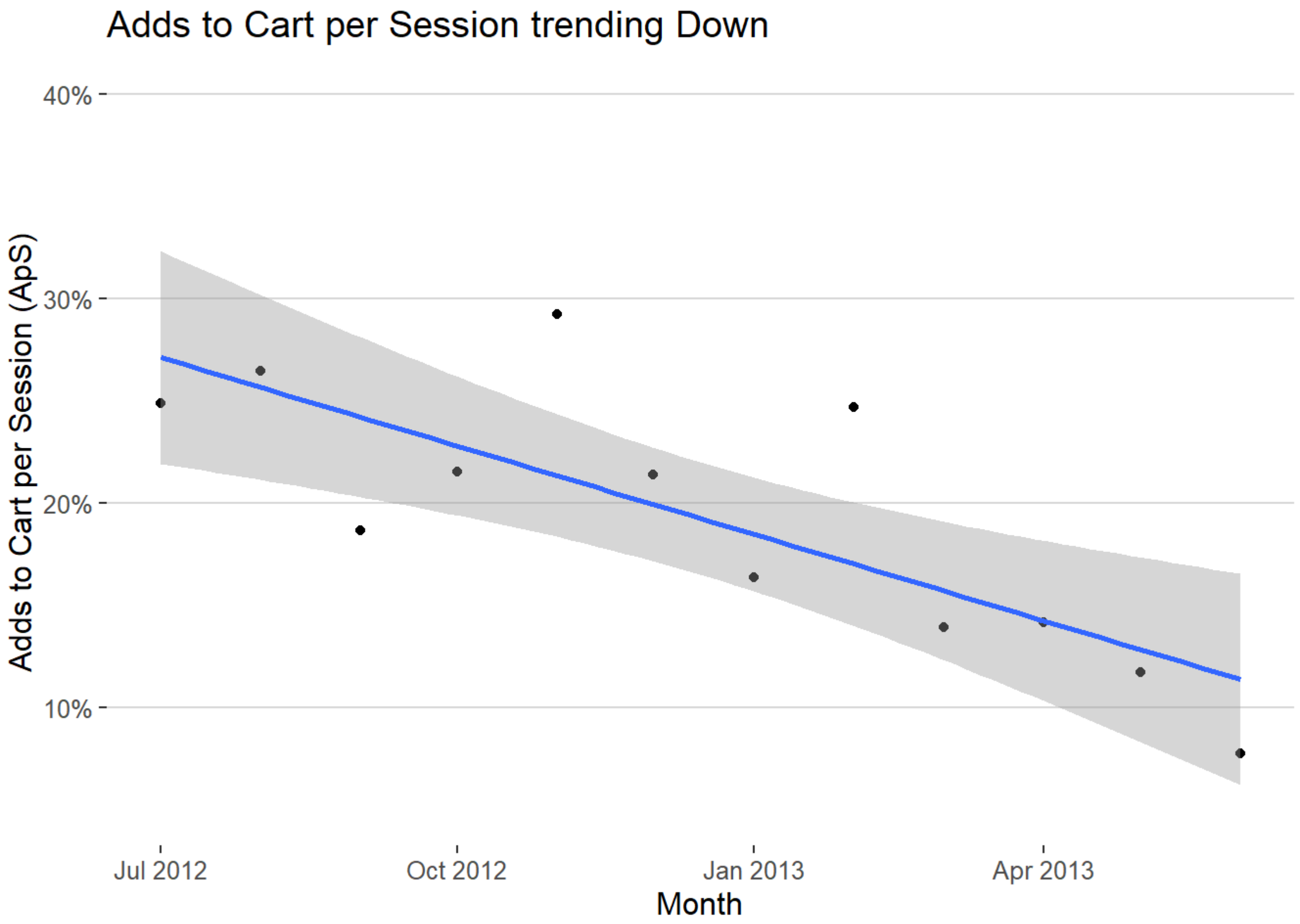
```
#scatterplot of Transactions/AddsToCart by month
ggplot(data=MonthJoin, aes(x=YearMonth, y= TpA))+
  geom_point()+
  geom_smooth(method="lm")+
  coord_cartesian(ylim = c(.05,.35))+
  scale_y_continuous(labels = scales::percent_format(accuracy = 1))+
  theme_hc()+
  labs(title = "Transactions per Add to Cart trending Up", x="Month", y="Transactions per Add to
Cart (TpA)")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
#scatterplot of AddsToCart/Sessions by month
ggplot(data=MonthJoin, aes(x=YearMonth, y= ApS))+
  geom_point()+
  geom_smooth(method="lm")+
  coord_cartesian(ylim = c(.05,.4))+
  scale_y_continuous(labels = scales::percent_format(accuracy = 1))+
  theme_hc()+
  labs(title = "Adds to Cart per Session trending Down", x="Month", y="Adds to Cart per Session (ApS)")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
#linear model of transactions by adds to cart
model1=lm(transactions~addsToCart, data=MonthJoin)
summary(model1)
```

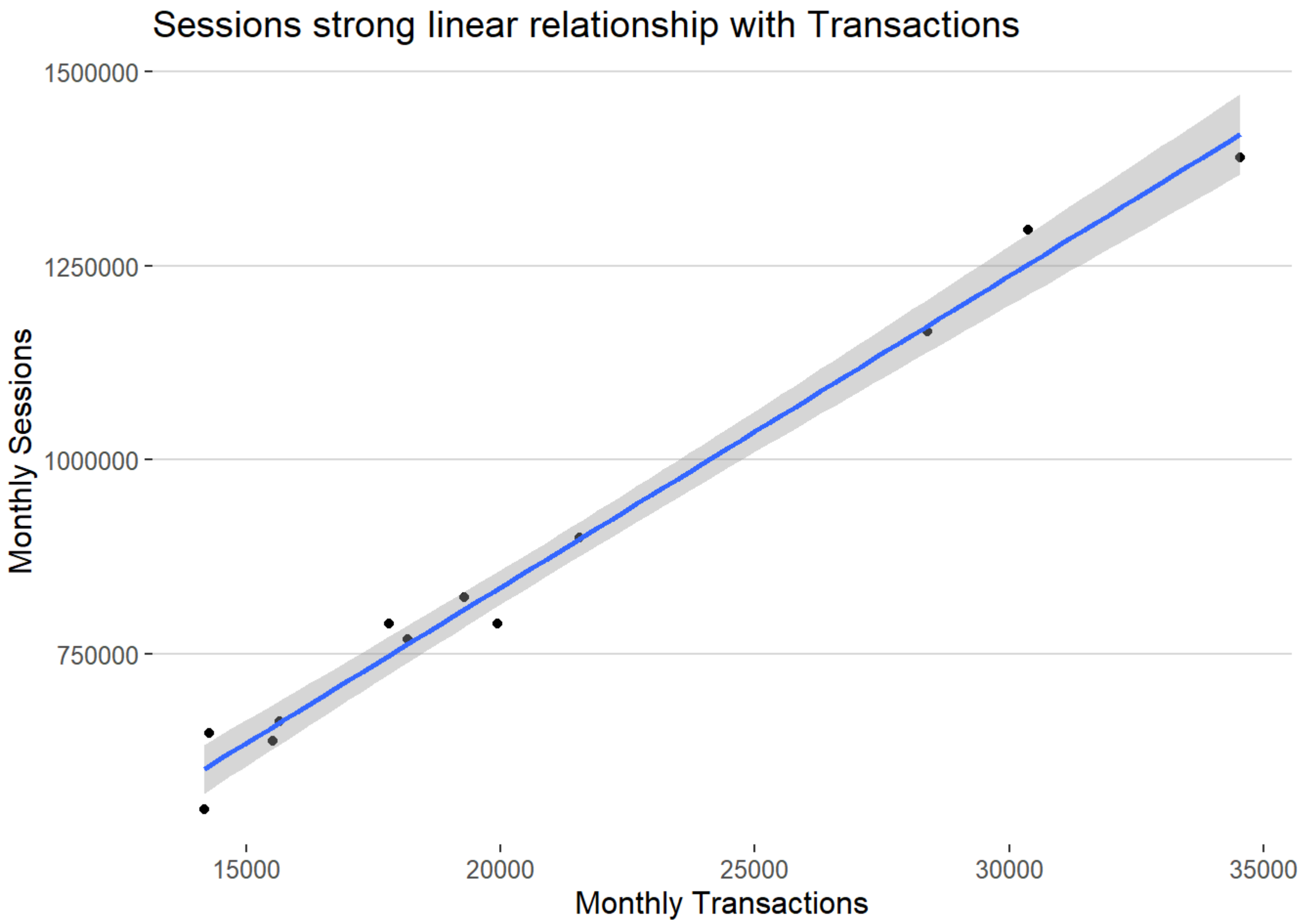
```
##
## Call:
## lm(formula = transactions ~ addsToCart, data = MonthJoin)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -7130  -4797  -1059   2213  12492
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.495e+04  9.494e+03   2.628   0.0253 *
## addsToCart  -2.686e-02  6.017e-02  -0.446   0.6648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6990 on 10 degrees of freedom
## Multiple R-squared:  0.01954,	Adjusted R-squared:  -0.07851
## F-statistic: 0.1993 on 1 and 10 DF,  p-value: 0.6648
```

```
#linear model of transactions by sessions
model2=lm(transactions~sessions, data=MonthJoin)
summary(model2)
```

```
##
## Call:
## lm(formula = transactions ~ sessions, data = MonthJoin)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1130.62  -542.39   -58.65   524.93  1179.84
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.412e+02  8.497e+02  -0.637    0.538
## sessions     2.458e-02  9.374e-04  26.226  1.5e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 845.1 on 10 degrees of freedom
## Multiple R-squared:  0.9857, Adjusted R-squared:  0.9842
## F-statistic: 687.8 on 1 and 10 DF,  p-value: 1.497e-10
```

```
#scatter chart of transactions by sessions
ggplot(data=MonthJoin, aes(x=transactions, y=sessions))+
  geom_point()+
  theme_hc()+
  geom_smooth(method="lm")+
  labs(title = "Sessions strong linear relationship with Transactions", x="Monthly Transactions"
, y="Monthly Sessions")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



#Month over Month Didn’t end up using this data in xlsx file, output didn’t look great but was useful to look at MonthOver data.

```
#Adif takes the actual difference between x and its lag
Adif<-function(x){
  y=(x-lag(x))
}

#Rdif takes relative difference between x and its lag
Rdif<-function(x){
  y=(x-lag(x))/lag(x)
}

#creating month over month data for each month
MonthOver<- MonthJoin%>%
  #Calculating actual and relative difference
  mutate(Rdif_sessions=Rdif(sessions), Adif_sessions=Adif(sessions),
         Rdif_transactions=Rdif(transactions), Adif_transactions=Adif(transactions),
         Rdif_QTY=Rdif(QTY), Adif_QTY=Adif(QTY),
         Rdif_ECR=Rdif(ECR), Adif_ECR=Adif(ECR),
         Rdif_addsToCart=Rdif(addsToCart), Adif_addsToCart=Adif(addsToCart)
  )

#Taking last two months of MonthJoin dataset to later calculate Actual and
#Relative Difference as equation in xlsx
recMonth<-
  MonthJoin%>%
  filter(YearMonth>'2013-04-01')
```

#Writing data to xlsx

```
#creating workbook
wb<- createWorkbook()

#bold style to be applied to headers
bold <- createStyle(textDecoration = "Bold", halign = "center", valign = "center", wrapText = TR
UE)

#adding month by device data
addWorksheet(wb, "Month by Device")
writeDataTable(wb, "Month by Device", groupedCounts, headerStyle = bold)

#adding month over month data
addWorksheet(wb, "Month Over Month")
writeData(wb, "Month Over Month", recMonth, headerStyle = bold)

#adding Absolute Difference and Relative Difference headers
writeData(wb,"Month Over Month", x=c("Absolute Difference", "Relative Difference"), startCol = 1
, startRow = 4)
#adding bold style
addStyle(wb,"Month Over Month", bold, col = 1, row = 4:5)

#writing absolute and relative difference formulas for each row
writeFormula(wb, "Month Over Month", x=c("B3-B2", "(B3-B2)/B2"),startCol = 2, startRow = 4)
writeFormula(wb, "Month Over Month", x=c("C3-C2", "(C3-C2)/C2"),startCol = 3, startRow = 4)
writeFormula(wb, "Month Over Month", x=c("D3-D2", "(D3-D2)/D2"),startCol = 4, startRow = 4)
writeFormula(wb, "Month Over Month", x=c("E3-E2", "(E3-E2)/E2"),startCol = 5, startRow = 4)
writeFormula(wb, "Month Over Month", x=c("F3-F2", "(F3-F2)/F2"),startCol = 6, startRow = 4)
writeFormula(wb, "Month Over Month", x=c("G3-G2", "(G3-G2)/G2"),startCol = 7, startRow = 4)
writeFormula(wb, "Month Over Month", x=c("H3-H2", "(H3-H2)/H2"),startCol = 8, startRow = 4)
writeFormula(wb, "Month Over Month", x=c("I3-I2", "(I3-I2)/I2"),startCol = 9, startRow = 4)

#creating style to format numbers as percentages
pct <- createStyle(numFmt="0%")

#adding style to relative differences
addStyle(wb, "Month Over Month",pct,col= 2:9, row = 5)

#creating a positive style (green) and negative style (red)
negStyle <- createStyle(fontColour = "#9C0006", bgFill = "#FFC7CE")
posStyle <- createStyle(fontColour = "#006100", bgFill = "#C6EFCE")

#conditionally formatting positive difference to be green and negative to be red
conditionalFormatting(wb, "Month Over Month",
                      cols = 2:9,
                      rows = 4:5, rule = "<0", style = negStyle
)
conditionalFormatting(wb, "Month Over Month",
                      cols = 2:9,
                      rows = 4:5, rule = ">0", style = posStyle
)

#adding device percentages sheet to workbook
```

```
addWorksheet(wb, "Device Percentages")
writeData(wb, "Device Percentages", groupedPerc, headerStyle = bold)

#saving workbook
saveWorkbook(wb, file="websale.xlsx", overwrite=TRUE)
```