BUSML 7245: Case Study - BT57.com

Shyni David, Marianne Morgan, Tasneem Motiwala

12 April 2019

```
#Load libraries required for the assignment

library(MASS)
library(tidyverse)
library(dplyr)
library(readr)
library(foreign)
library(car)
library(caret) # for model training and testing
library(pROC) # for ROC curves
library(glmnet)
library(caTools) #for ROC
library(e1071)
library(GGally)
library(stringr)
library(sqldf)
library(nnet) # for multinomial logit
```

Load and process data

```
BTData <- read_csv("BT57_Data_File.csv")

# Create variables Current, Lag1, Lag2 and Lag3 as composites of the different channe
ls, and loyalty as a composite of basic/silver/gold/platinum status
btd_01 <- BTData %>%
        mutate(Current = ifelse(Direct == 1, "direct", ifelse(Disp == 1, "disp", ifel
se(Email == 1, "email", ifelse(Organic == 1, "organic", ifelse(Paid == 1, "paid", "un
known")))))) %>%
        mutate(Lag1 = ifelse(Lag1direct == 1, "direct", ifelse(Lag1disp == 1, "disp",
ifelse(Lag1email == 1, "email", ifelse(Lag1organic == 1, "organic", ifelse(Lag1paid =
= 1, "paid", "unknown")))) ) ) %>%
        mutate(Lag2 = ifelse(Lag2direct == 1, "direct", ifelse(Lag2disp ==1, "disp",
ifelse(Lag2email == 1, "email", ifelse(Lag2organic == 1, "organic", ifelse(Lag2paid =
= 1, "paid", "unknown")))))) %>%
        mutate(Lag3 = ifelse(Lag3direct == 1, "direct", ifelse(Lag3displ ==1, "disp",
ifelse(Lag3email == 1, "email", ifelse(Lag3organic == 1, "organic", ifelse(Lag3paid =
= 1, "paid", "unknown")))))) %>%
        mutate(Loyalty = ifelse(Basic == 1, "01-Basic", ifelse(Silver == 1, "02-Silve
r", ifelse(Gold == 1, "03-Gold", ifelse(Platinum == 1, "04-Platinum", "00-None")))))

# Add Lag_unknown columns for the 3 lags
btd_01 <- btd_01 %>%
```

```
        mutate(Lag1unknown = ifelse(Lag1 == "unknown", 1, 0), Lag2unknown = ifelse(La
g2 == "unknown", 1, 0), Lag3unknown = ifelse(Lag3 == "unknown", 1, 0))

#Check
#x <- sqldf("select Customer_ID, Direct, Disp, Email, Organic, Paid, Current, Lag1, L
ag2, Lag3 from btd_01 where Customer_ID in (107871604, 106257389, 106242216, 10625738
9)")

# Create lag variable restricted to being under 7 days
btd_02 <- btd_01 %>%
        group_by(Customer_ID) %>%
        mutate(Lag1_7day = ifelse((Day - lag(Day) < 7), Lag1, "null")) %>%
        mutate(Lag2_7day = ifelse((Day - lag(Day, 2) < 7), Lag2, "null")) %>%
        mutate(Lag3_7day = ifelse((Day - lag(Day, 3) < 7), Lag3, "null"))  %>%
        ungroup()

#Check
#x1 <- sqldf("select Customer_ID, Day,  Lag1, Lag2, Lag3, Lag1_7day, Lag2_7day, Lag3_
7day from btd_02 where Customer_ID in (107871604, 106257389, 106242216, 106257389)")

# Create lag variable restricted to being under 3 days
btd_03 <- btd_02 %>%
        group_by(Customer_ID) %>%
        mutate(Lag1_3day = ifelse((Day - lag(Day) < 3), Lag1, "null")) %>%
        mutate(Lag2_3day = ifelse((Day - lag(Day,2) < 3), Lag2, "null")) %>%
        mutate(Lag3_3day = ifelse((Day - lag(Day, 3) < 3), Lag3, "null")) %>% ungroup
()

#Check
#x2 <- sqldf("select Customer_ID, Day,  Lag1, Lag2, Lag3, Lag1_3day, Lag2_3day, Lag3_
3day from btd_03 where Customer_ID in (107871604, 106257389, 106242216, 106257389)")

# separate columns for marketing channels: Current
#sqldf("select distinct(current) from btd_03")

btd_04 <- btd_03 %>%
        mutate(Current_unknown = ifelse(Current == "unknown", 1, 0)) %>%
        mutate(Current_direct = ifelse(Current == "direct", 1, 0)) %>%
        mutate(Current_disp = ifelse(Current == "disp", 1, 0)) %>%
        mutate(Current_email = ifelse(Current == "email", 1, 0)) %>%
        mutate(Current_organic = ifelse(Current == "organic", 1, 0)) %>%
        mutate(Current_paid = ifelse(Current == "paid", 1, 0))

#Check
#x3 <-  sqldf("select Customer_ID, Day, Current, Current_unknown, current_direct, cur
rent_disp, current_email, current_organic, current_paid from btd_04 where Customer_ID
in (107871604, 106257389, 106242216, 106257389)")

# separate columns for marketing channels: Lag1-7days
btd_05 <- btd_04  %>%
```

```
        mutate(Lag1_7days_unknown = ifelse(is.na(Lag1_7day) == TRUE, 0, ifelse(Lag1_7
day == "unknown", 1, 0))) %>%
        mutate(Lag1_7days_null = ifelse(is.na(Lag1_7day) == TRUE, 1, 0)) %>%
        mutate(Lag1_7days_direct = ifelse(is.na(Lag1_7day) == TRUE, 0, ifelse(Lag1_7d
ay == "direct", 1, 0))) %>%
        mutate(Lag1_7days_disp = ifelse(is.na(Lag1_7day) == TRUE, 0, ifelse(Lag1_7day
== "disp", 1, 0))) %>%
        mutate(Lag1_7days_email = ifelse(is.na(Lag1_7day) == TRUE, 0, ifelse(Lag1_7da
y == "email", 1, 0))) %>%
        mutate(Lag1_7days_organic = ifelse(is.na(Lag1_7day) == TRUE, 0, ifelse(Lag1_7
day == "organic", 1, 0))) %>%
        mutate(Lag1_7days_paid = ifelse(is.na(Lag1_7day) == TRUE, 0, ifelse(Lag1_7day
== "paid", 1, 0)))

#check
#x4 <-  sqldf("select Customer_ID, Day, Lag1_7day, Lag1_7days_direct, Lag1_7days_disp
, Lag1_7days_email, Lag1_7days_organic, Lag1_7days_paid, Lag1_7days_null, Lag1_7days_
unknown from btd_05 where Customer_ID in (107871604, 106257389, 106242216, 106257389)
")

# separate columns for marketing channels: Lag2-7days
btd_05_01 <- btd_05  %>%
        mutate(Lag2_7days_unknown = ifelse(is.na(Lag2_7day) == TRUE, 0, ifelse(Lag2_7
day == "unknown", 1, 0))) %>%
        mutate(Lag2_7days_null = ifelse(is.na(Lag2_7day) == TRUE, 1, 0)) %>%
        mutate(Lag2_7days_direct = ifelse(is.na(Lag2_7day) == TRUE, 0, ifelse(Lag2_7d
ay == "direct", 1, 0))) %>%
        mutate(Lag2_7days_disp = ifelse(is.na(Lag2_7day) == TRUE, 0, ifelse(Lag2_7day
== "disp", 1, 0))) %>%
        mutate(Lag2_7days_email = ifelse(is.na(Lag2_7day) == TRUE, 0, ifelse(Lag2_7da
y == "email", 1, 0))) %>%
        mutate(Lag2_7days_organic = ifelse(is.na(Lag2_7day) == TRUE, 0, ifelse(Lag2_7
day == "organic", 1, 0))) %>%
        mutate(Lag2_7days_paid = ifelse(is.na(Lag2_7day) == TRUE, 0, ifelse(Lag2_7day
== "paid", 1, 0)))

#check
#x5 <-  sqldf("select Customer_ID, Day, Lag2_7day, Lag2_7days_direct, Lag2_7days_disp
, Lag2_7days_email, Lag2_7days_organic, Lag2_7days_paid, Lag2_7days_null, Lag2_7days_
unknown from btd_05_01 where Customer_ID in (107871604, 106257389, 106242216, 1062573
89)")

# separate columns for marketing channels: Lag3-7days
btd_05_02 <- btd_05_01  %>%
        mutate(Lag3_7days_unknown = ifelse(is.na(Lag3_7day) == TRUE, 0, ifelse(Lag3_7
day == "unknown", 1, 0))) %>%
        mutate(Lag3_7days_null = ifelse(is.na(Lag3_7day) == TRUE, 1, 0)) %>%
        mutate(Lag3_7days_direct = ifelse(is.na(Lag3_7day) == TRUE, 0, ifelse(Lag3_7d
ay == "direct", 1, 0))) %>%
        mutate(Lag3_7days_disp = ifelse(is.na(Lag3_7day) == TRUE, 0, ifelse(Lag3_7day
```

```
== "disp", 1, 0))) %>%
        mutate(Lag3_7days_email = ifelse(is.na(Lag3_7day) == TRUE, 0, ifelse(Lag3_7da
y == "email", 1, 0))) %>%
        mutate(Lag3_7days_organic = ifelse(is.na(Lag3_7day) == TRUE, 0, ifelse(Lag3_7
day == "organic", 1, 0))) %>%
        mutate(Lag3_7days_paid = ifelse(is.na(Lag3_7day) == TRUE, 0, ifelse(Lag3_7day
== "paid", 1, 0)))

#check
#x6 <-  sqldf("select Customer_ID, Day, Lag3_7day, Lag3_7days_direct, Lag3_7days_disp
, Lag3_7days_email, Lag3_7days_organic, Lag3_7days_paid, Lag3_7days_null, Lag3_7days_
unknown from btd_05_02 where Customer_ID in (107871604, 106257389, 106242216, 1062573
89)")

# Separate columns for marketing channels: Lag1-3days
btd_06 <- btd_05_02 %>%
        mutate(Lag1_3days_unknown = ifelse(is.na(Lag1_3day) == TRUE, 0, ifelse(Lag1_3
day == "unknown", 1, 0))) %>%
        mutate(Lag1_3days_null = ifelse(is.na(Lag1_3day) == TRUE, 1, 0)) %>%
        mutate(Lag1_3days_direct = ifelse(is.na(Lag1_3day) == TRUE, 0, ifelse(Lag1_3d
ay == "direct", 1, 0))) %>%
        mutate(Lag1_3days_disp = ifelse(is.na(Lag1_3day) == TRUE, 0, ifelse(Lag1_3day
== "disp", 1, 0))) %>%
        mutate(Lag1_3days_email = ifelse(is.na(Lag1_3day) == TRUE, 0, ifelse(Lag1_3da
y == "email", 1, 0))) %>%
        mutate(Lag1_3days_organic = ifelse(is.na(Lag1_3day) == TRUE, 0, ifelse(Lag1_3
day == "organic", 1, 0))) %>%
        mutate(Lag1_3days_paid = ifelse(is.na(Lag1_3day) == TRUE, 0, ifelse(Lag1_3day
== "paid", 1, 0)))

#check
#x7 <-  sqldf("select Customer_ID, Day, Lag1_3day, Lag1_3days_direct, Lag1_3days_disp
, Lag1_3days_email, Lag1_3days_organic, Lag1_3days_paid, Lag1_3days_null, Lag1_3days_
unknown from btd_06 where Customer_ID in (107871604, 106257389, 106242216, 106257389)
")

# separate columns for marketing channels: Lag2-3days
btd_06_01 <- btd_06 %>%
        mutate(Lag2_3days_unknown = ifelse(is.na(Lag2_3day) == TRUE, 0, ifelse(Lag2_3
day == "unknown", 1, 0))) %>%
        mutate(Lag2_3days_null = ifelse(is.na(Lag2_3day) == TRUE, 1, 0)) %>%
        mutate(Lag2_3days_direct = ifelse(is.na(Lag2_3day) == TRUE, 0, ifelse(Lag2_3d
ay == "direct", 1, 0))) %>%
        mutate(Lag2_3days_disp = ifelse(is.na(Lag2_3day) == TRUE, 0, ifelse(Lag2_3day
== "disp", 1, 0))) %>%
        mutate(Lag2_3days_email = ifelse(is.na(Lag2_3day) == TRUE, 0, ifelse(Lag2_3da
y == "email", 1, 0))) %>%
        mutate(Lag2_3days_organic = ifelse(is.na(Lag2_3day) == TRUE, 0, ifelse(Lag2_3
day == "organic", 1, 0))) %>%
        mutate(Lag2_3days_paid = ifelse(is.na(Lag2_3day) == TRUE, 0, ifelse(Lag2_3day
```

```
== "paid", 1, 0)))

#check
#x8 <-  sqldf("select Customer_ID, Day, Lag2_3day, Lag2_3days_direct, Lag2_3days_disp
, Lag2_3days_email, Lag2_3days_organic, Lag2_3days_paid, Lag2_3days_null, Lag2_3days_
unknown  from btd_06_01 where Customer_ID in (107871604, 106257389, 106242216, 106257
389)")

# separate columns for marketing channels: Lag3-3days
btd_06_02 <- btd_06_01 %>%
        mutate(Lag3_3days_unknown = ifelse(is.na(Lag3_3day) == TRUE, 0, ifelse(Lag3_3
day == "unknown", 1, 0))) %>%
        mutate(Lag3_3days_null = ifelse(is.na(Lag3_3day) == TRUE, 1, 0)) %>%
        mutate(Lag3_3days_direct = ifelse(is.na(Lag3_3day) == TRUE, 0, ifelse(Lag3_3d
ay == "direct", 1, 0))) %>%
        mutate(Lag3_3days_disp = ifelse(is.na(Lag3_3day) == TRUE, 0, ifelse(Lag3_3day
== "disp", 1, 0))) %>%
        mutate(Lag3_3days_email = ifelse(is.na(Lag3_3day) == TRUE, 0, ifelse(Lag3_3da
y == "email", 1, 0))) %>%
        mutate(Lag3_3days_organic = ifelse(is.na(Lag3_3day) == TRUE, 0, ifelse(Lag3_3
day == "organic", 1, 0))) %>%
        mutate(Lag3_3days_paid = ifelse(is.na(Lag3_3day) == TRUE, 0, ifelse(Lag3_3day
== "paid", 1, 0)))

# Some customer IDs dont see to have complete data. Exclude these customers from data
to see how the models perform
# btd_06_02 <- btd_06_02 %>%
#        filter(!(Customer_ID %in% c(111096680, 111155221, 113474511, 113618012, 1136
55275, 113742641, 115569063, 115599079, 115912380, 116506489, 119793828, 119925357)))

# remove observations with all 4 touch-points as unknown
btd_06_02 <- btd_06_02 %>%
        mutate(Combo = paste0(Current,Lag1,Lag2,Lag3)) %>%
        filter(Combo != "unknownunknownunknownunknown")
```

Summary of data

```
# Frequency of use of each channel at last touchpoint (for Table 2 of paper)
bt57visits <- btd_06_02 %>%
        summarize(Direct = sum(Direct), Disp = sum(Disp), Email = sum(Email), Organic
= sum(Organic), Paid = sum(Paid)) %>%
        gather(key = Channel) %>%
        arrange(-value) %>%
        rename("Channel Visits" = "value")

# Frequency of use of each channel at last touchpoint (for Table 2 of paper) - Purcha
ses only
bt57P <- btd_06_02 %>%
        filter(Purchase == 1) %>%
        summarize(Direct = sum(Direct), Disp = sum(Disp), Email = sum(Email), Organic
= sum(Organic), Paid = sum(Paid)) %>%
        gather(key = Channel) %>%
        arrange(-value) %>%
        rename("Purchases" = "value")

# Calculate conversion rate for each channel based on last touchpoint (Table 2 of pap
er)
bt57sumstats <- bt57visits %>%
        left_join(bt57P) %>%
        mutate(Conversion_Rate = (Purchases / `Channel Visits`) * 100) %>%
        mutate(Conversion_Rate = round(Conversion_Rate, 2)) %>%
        arrange(-Conversion_Rate)

# SUMMARY STATISTICS
bt57sumstats <- as_tibble(bt57sumstats)
bt57sumstats$Conversion_Rate = paste0(bt57sumstats$Conversion_Rate, "%", sep = "")
knitr::kable(bt57sumstats)
```

Channel	Channel Visits	Purchases	Conversion_Rate
Direct	335	71	21.19%
Organic	575	77	13.39%
Paid	289	38	13.15%
Disp	57	7	12.28%
Email	132	8	6.06%

```
# Comparison of Display v/s Paid for ROI
bt57DispPaid <- bt57sumstats %>%
        filter(Channel == "Disp" | Channel == "Paid") %>%
        mutate(Cost = ifelse(Channel == "Disp", 7*`Channel Visits`, 14*`Channel Visit
s`)) %>%
        mutate(Profit = 150*Purchases) %>%
        mutate(ROI = ((Profit-Cost)/Cost)*100) %>%
        mutate(ROI = round(ROI, 2))

# ROI
bt57DispPaid <- as_tibble(bt57DispPaid)
bt57DispPaid$ROI = paste(bt57DispPaid$ROI, "%", sep = "")
knitr::kable(bt57DispPaid)
```

Channel	Channel Visits	Purchases	Conversion_Rate	Cost	Profit	ROI
Paid	289	38	13.15%	4046	5700	40.88%
Disp	57	7	12.28%	399	1050	163.16%

```
# Matrix of visit n v/s visit n-1 (Table 3 of paper) - using unrestricted lag1
bt57visitmatrix <- btd_06_02 %>%
        count(Current, Lag1) %>%
        spread(Lag1, n)

# CONTIGUOUS VISITS FOR THE SAME CUSTOMER -Table 3
knitr::kable(bt57visitmatrix)
```

Current	direct	disp	email	organic	paid	unknown	
direct	112	1	1	14	5	166	36
disp	2	12	2	8	2	20	11
email	5	1	27	9	1	81	8
organic	17	8	8	182	81	245	34
paid	10	1	2	54	57	152	13
unknown	189	34	92	308	143	1306	94

```
# Matrix of visit n v/s visit n-1 (Table 3 of paper) - for Purchases only
bt57visitmatrixP <- btd_06_02 %>%
        filter(Purchase == 1) %>%
        count(Current, Lag1) %>%
        spread(key = Lag1, value = n)

# CONTIGUOS VISITS FOR THE SAME CUSTOMER (PURCHASES ONLY)
knitr::kable(bt57visitmatrixP)
```

Current	direct	disp	email	organic	paid	unknown
direct	59	NA	NA	1	2	9
disp	NA	7	NA	NA	NA	NA
email	NA	NA	5	NA	NA	3
organic	1	NA	NA	74	NA	2
paid	NA	NA	NA	NA	38	NA
unknown	NA	NA	NA	NA	NA	24

```
# Matrix of visit n v/s visit n-1 (Table 3 of paper) - for Non-Purchases only
bt57visitmatrixNP <- btd_06_02 %>%
        filter(Purchase == 0) %>%
        count(Current, Lag1) %>%
        spread(key = Lag1, value = n)

# CONTIGUOS VISITS FOR THE SAME CUSTOMER (NON-PURCHASES ONLY)
knitr::kable(bt57visitmatrixNP)
```

Current	direct	disp	email	organic	paid	unknown	
direct	53	1	1	13	3	157	36
disp	2	5	2	8	2	20	11
email	5	1	22	9	1	78	8
organic	16	8	8	108	81	243	34
paid	10	1	2	54	19	152	13
unknown	189	34	92	308	143	1282	94

```
#### Using lag3Days
# Matrix of visit n v/s visit n-1 (Table 3 of paper) - using unrestricted lag1
bt57visitmatrix3d <- btd_06_02 %>%
        count(Current, Lag1_3day) %>%
        spread(Lag1_3day, n)

# CONTIGUOUS VISITS FOR THE SAME CUSTOMER -Table 3
knitr::kable(bt57visitmatrix3d)
```

Current	direct	disp	email	organic	paid	unknown	
direct	112	1	1	14	5	166	36
disp	2	12	2	8	2	19	12
email	5	1	27	9	1	81	8
organic	17	8	8	182	81	243	36
paid	10	1	2	54	57	151	14
unknown	189	34	92	308	143	1306	94

```
# Matrix of visit n v/s visit n-1 (Table 3 of paper) - for Purchases only
bt57visitmatrixP3d <- btd_06_02 %>%
        filter(Purchase == 1) %>%
        count(Current, Lag1_3day) %>%
        spread(key = Lag1_3day, value = n)

# CONTIGUOS VISITS FOR THE SAME CUSTOMER (PURCHASES ONLY)
knitr::kable(bt57visitmatrixP3d)
```

Current	direct	disp	email	organic	paid	unknown
direct	59	NA	NA	1	2	9
disp	NA	7	NA	NA	NA	NA
email	NA	NA	5	NA	NA	3
organic	1	NA	NA	74	NA	2
paid	NA	NA	NA	NA	38	NA
unknown	NA	NA	NA	NA	NA	24

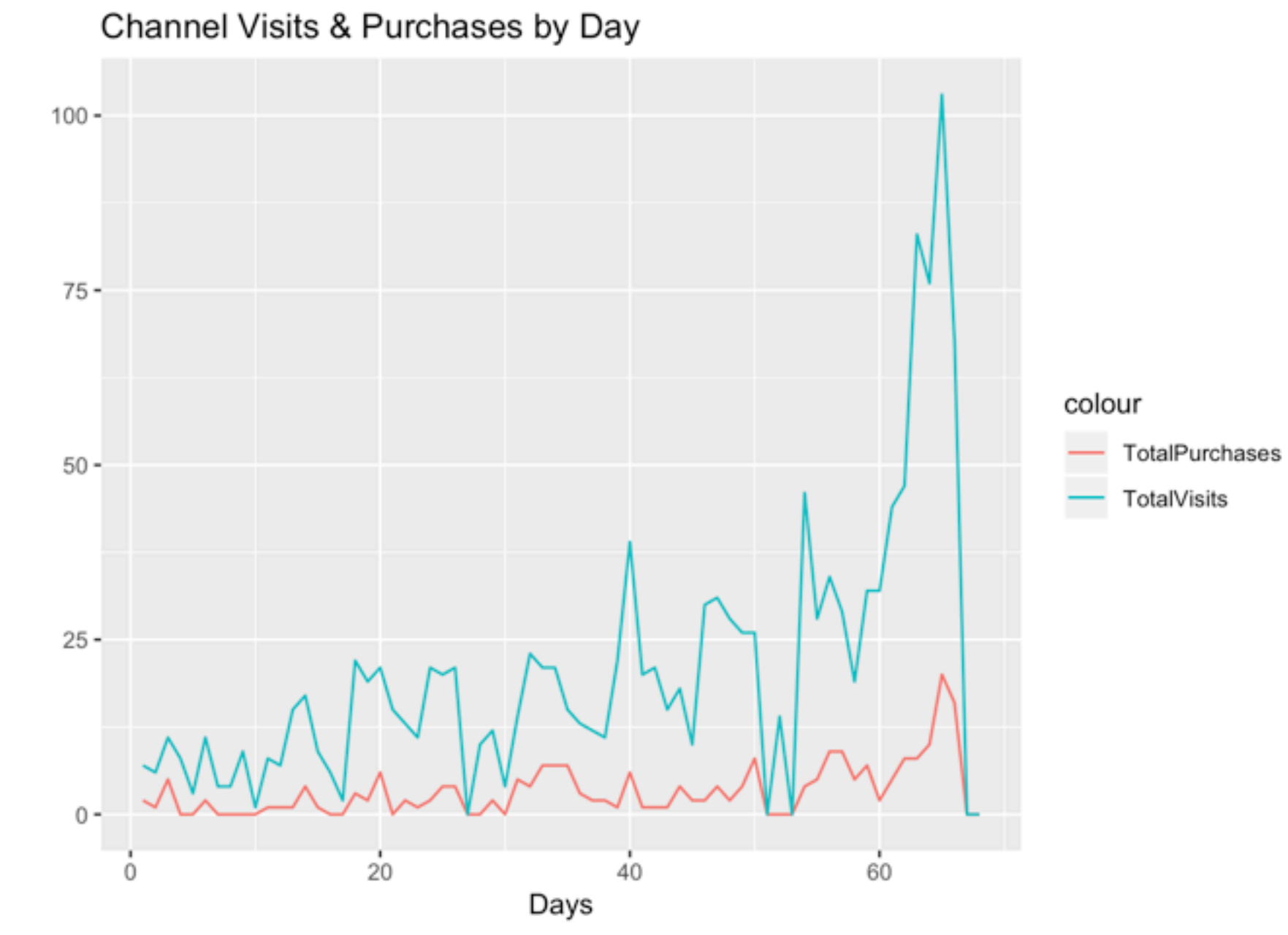
```
# Matrix of visit n v/s visit n-1 (Table 3 of paper) - for Non-Purchases only
bt57visitmatrixNP3d <- btd_06_02 %>%
        filter(Purchase == 0) %>%
        count(Current, Lag1_3day) %>%
        spread(key = Lag1_3day, value = n)

# CONTIGUOS VISITS FOR THE SAME CUSTOMER (NON-PURCHASES ONLY)
knitr::kable(bt57visitmatrixNP3d)
```

Current	direct	disp	email	organic	paid	unknown	
direct	53	1	1	13	3	157	36
disp	2	5	2	8	2	19	12
email	5	1	22	9	1	78	8
organic	16	8	8	108	81	241	36
paid	10	1	2	54	19	151	14
unknown	189	34	92	308	143	1282	94

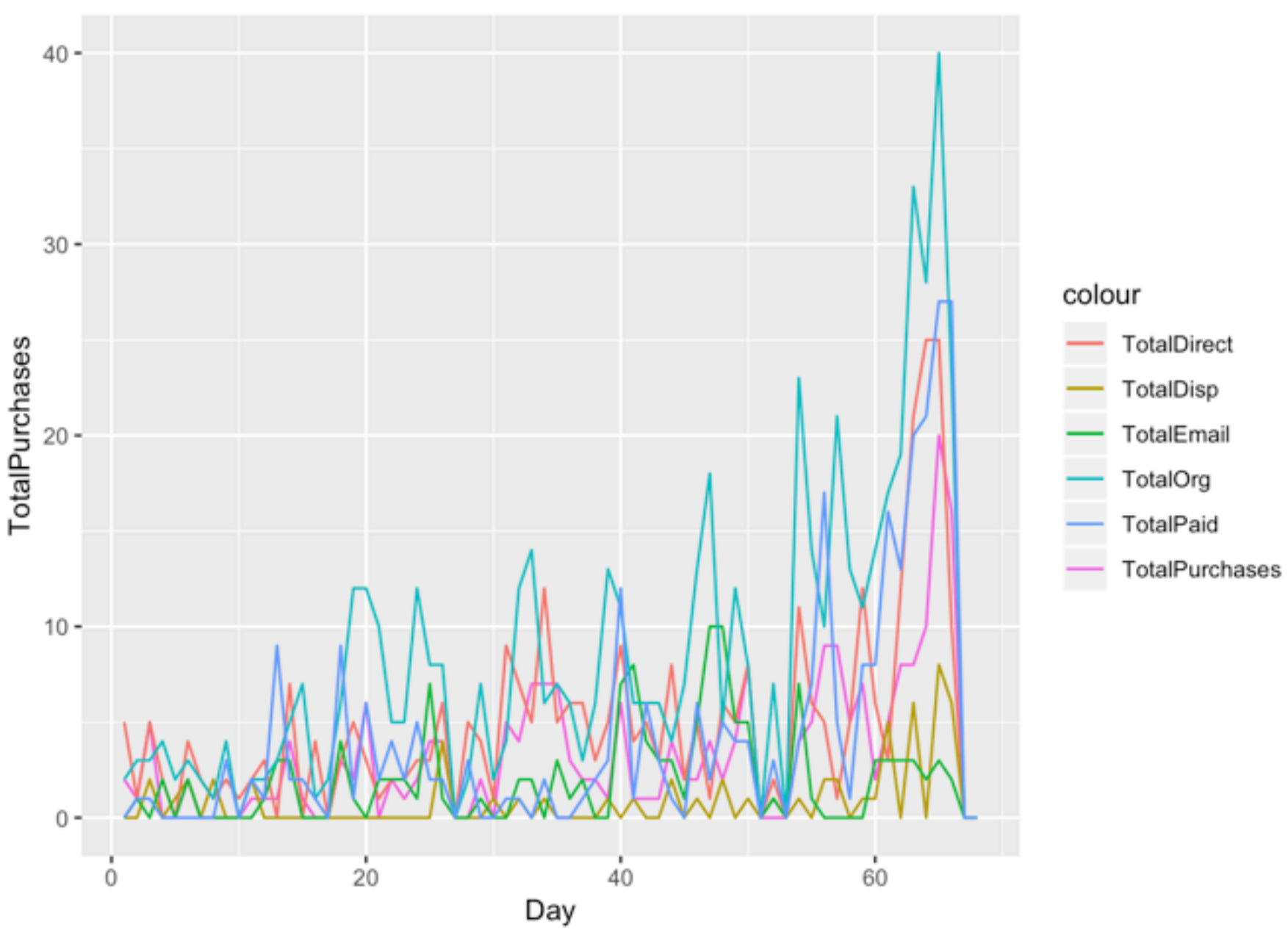
```
# Matrix of visit n v/s visit n-3 (Table 3 of paper)
bt57visitmatrix <- btd_06_02 %>%
        count(Current, Lag3) %>%
        spread(Lag3, n)

# Matrix of visit n v/s visit n-3 (Table 3 of paper) - for Purchases only
bt57visitmatrixP <- btd_06_02 %>%
        filter(Purchase == 1) %>%
        count(Current, Lag3) %>%
        spread(key = Lag3, value = n)

# Matrix of visit n v/s visit n-3 (Table 3 of paper) - for Non-Purchases only
bt57visitmatrixNP <- btd_06_02 %>%
        filter(Purchase == 0) %>%
        count(Current, Lag3) %>%
        spread(key = Lag3, value = n)
```

Graph

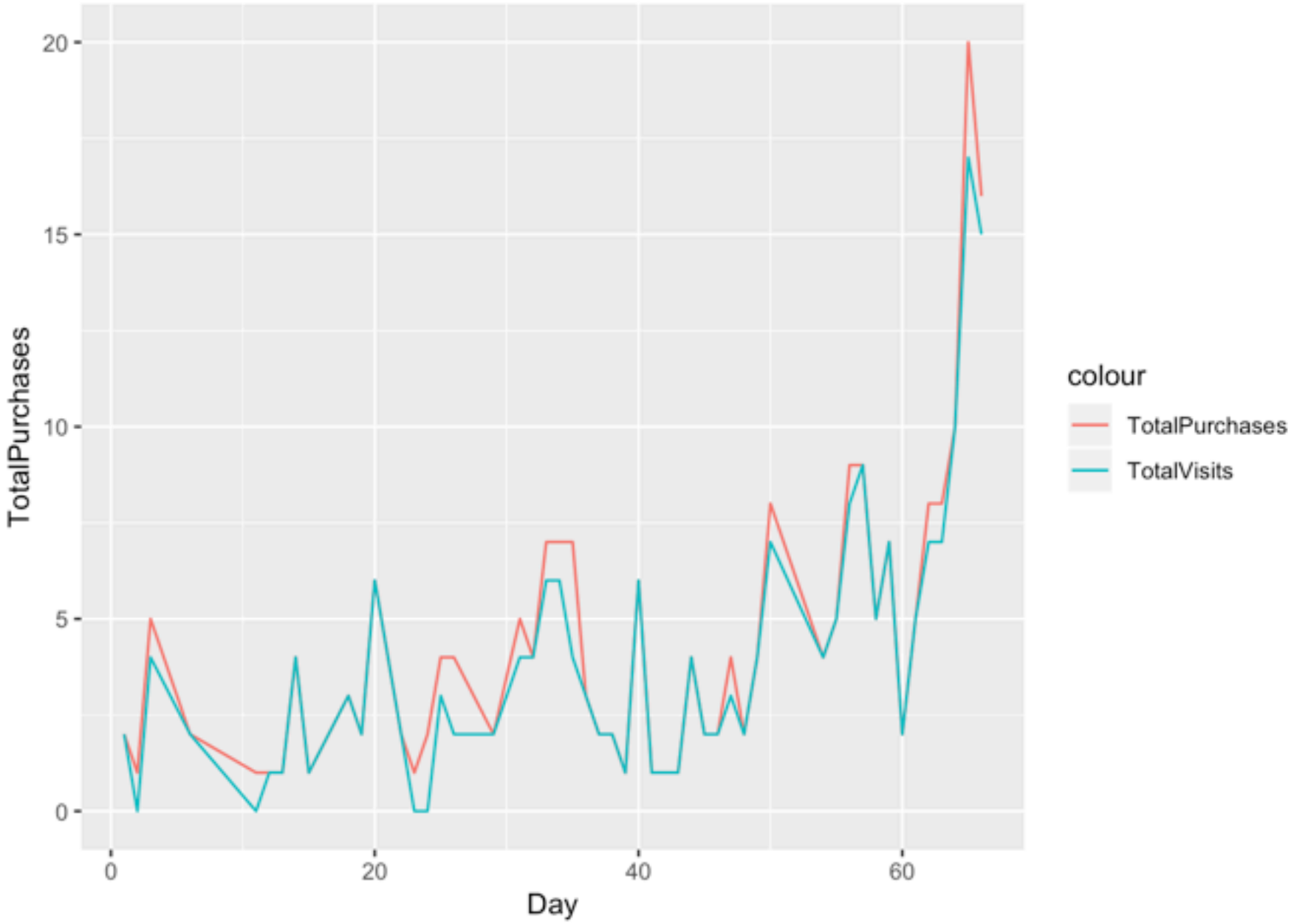
```
bt57graph <- btd_06_02 %>%
        mutate(Visits = Direct + Disp + Email + Organic + Paid) %>%
        group_by(Day) %>%
        summarize(TotalPurchases = sum(Purchase), TotalVisits = sum(Visits), TotalDir
ect = sum(Direct), TotalDisp = sum(Disp), TotalEmail = sum(Email), TotalOrg = sum(Org
anic), TotalPaid = sum(Paid))

ggplot(bt57graph, aes(Day)) +
        geom_line(aes(y = TotalPurchases, color = "TotalPurchases")) +
        geom_line(aes(y = TotalVisits, color = "TotalVisits")) +
        labs(title = "Channel Visits & Purchases by Day", x = "Days", y = "")
```



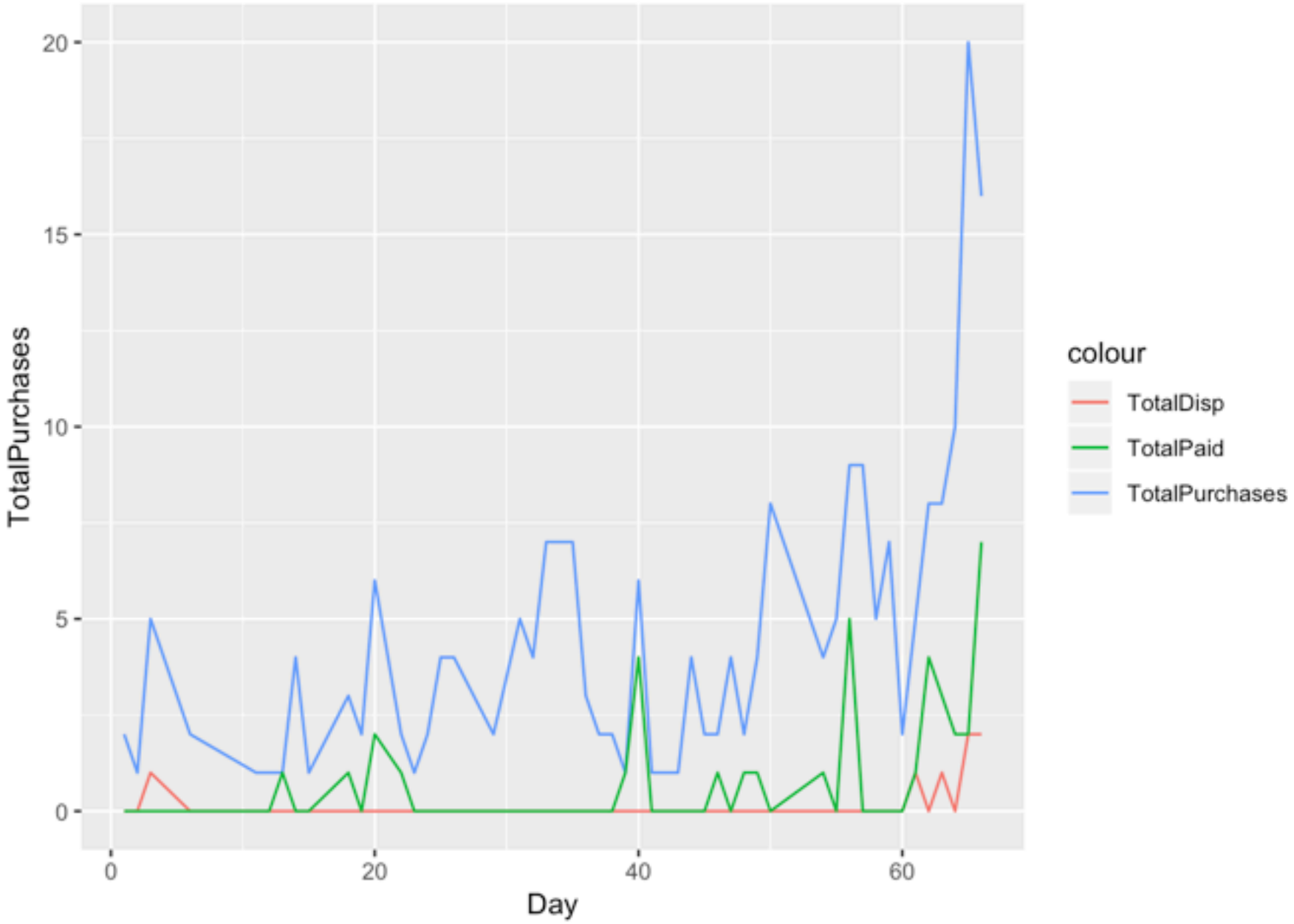

```
ggplot(bt57graph, aes(Day)) +
        geom_line(aes(y = TotalPurchases, color = "TotalPurchases")) +
        geom_line(aes(y = TotalDirect, color = "TotalDirect")) +
        geom_line(aes(y = TotalDisp, color = "TotalDisp")) +
        geom_line(aes(y = TotalEmail, color = "TotalEmail")) +
        geom_line(aes(y = TotalOrg, color = "TotalOrg")) +
        geom_line(aes(y = TotalPaid, color = "TotalPaid"))
```




```
# for purchases only
bt57graphP <- btd_06_02 %>%
        filter(Purchase == 1) %>%
        mutate(Visits = Direct + Disp + Email + Organic + Paid) %>%
        group_by(Day) %>%
        summarize(TotalPurchases = sum(Purchase), TotalVisits = sum(Visits), TotalDir
ect = sum(Direct), TotalDisp = sum(Disp), TotalEmail = sum(Email), TotalOrg = sum(Org
anic), TotalPaid = sum(Paid))

ggplot(bt57graphP, aes(Day)) +
        geom_line(aes(y = TotalPurchases, color = "TotalPurchases")) +
        geom_line(aes(y = TotalVisits, color = "TotalVisits"))
```




```
ggplot(bt57graphP, aes(Day)) +
        geom_line(aes(y = TotalPurchases, color = "TotalPurchases")) +
        #geom_line(aes(y = TotalDirect, color = "TotalDirect")) +
        geom_line(aes(y = TotalDisp, color = "TotalDisp")) +
        #geom_line(aes(y = TotalEmail, color = "TotalEmail")) +
        #geom_line(aes(y = TotalOrg, color = "TotalOrg")) +
        geom_line(aes(y = TotalPaid, color = "TotalPaid"))
```



Prep data and create train/test datasets to run and validate models

```
bt57 <- btd_06_02

# Convert response variable to factor
bt57$Purchase = as.factor(bt57$Purchase) # coded as 1=good_risk, 2=bad_risk
table(bt57$Purchase)
```

```
##
##    0    1
## 3329  225
```

```
#str(bt57)

#The values for the class/response variable must be factors and must be valid names
bt57$Purchase <-as.factor(ifelse(bt57$Purchase==1, "Purchase", "No_Purchase")) #recod
e response variable
table(bt57$Purchase)
```

```
##
## No_Purchase    Purchase
##        3329         225
```

```
#As coded, the default reference/first level is the "lowest" alphabetical value; here it is "Bad". Caret's default "Positive" class is the first level of the outcome variable in binary classification. Make "Good" the first level
bt57$Purchase <- relevel(bt57$Purchase, "Purchase")
table(bt57$Purchase)
```

```
##
##    Purchase No_Purchase
##         225        3329
```

```
set.seed(5678)
trainIndex <- createDataPartition(bt57$Purchase, p = .75,
                                  list = FALSE,
                                  times = 1)
#head(trainIndex)

bt57train <- bt57[ trainIndex,]
bt57test  <- bt57[-trainIndex,]
```

Build classification models - CURRENT channels separated

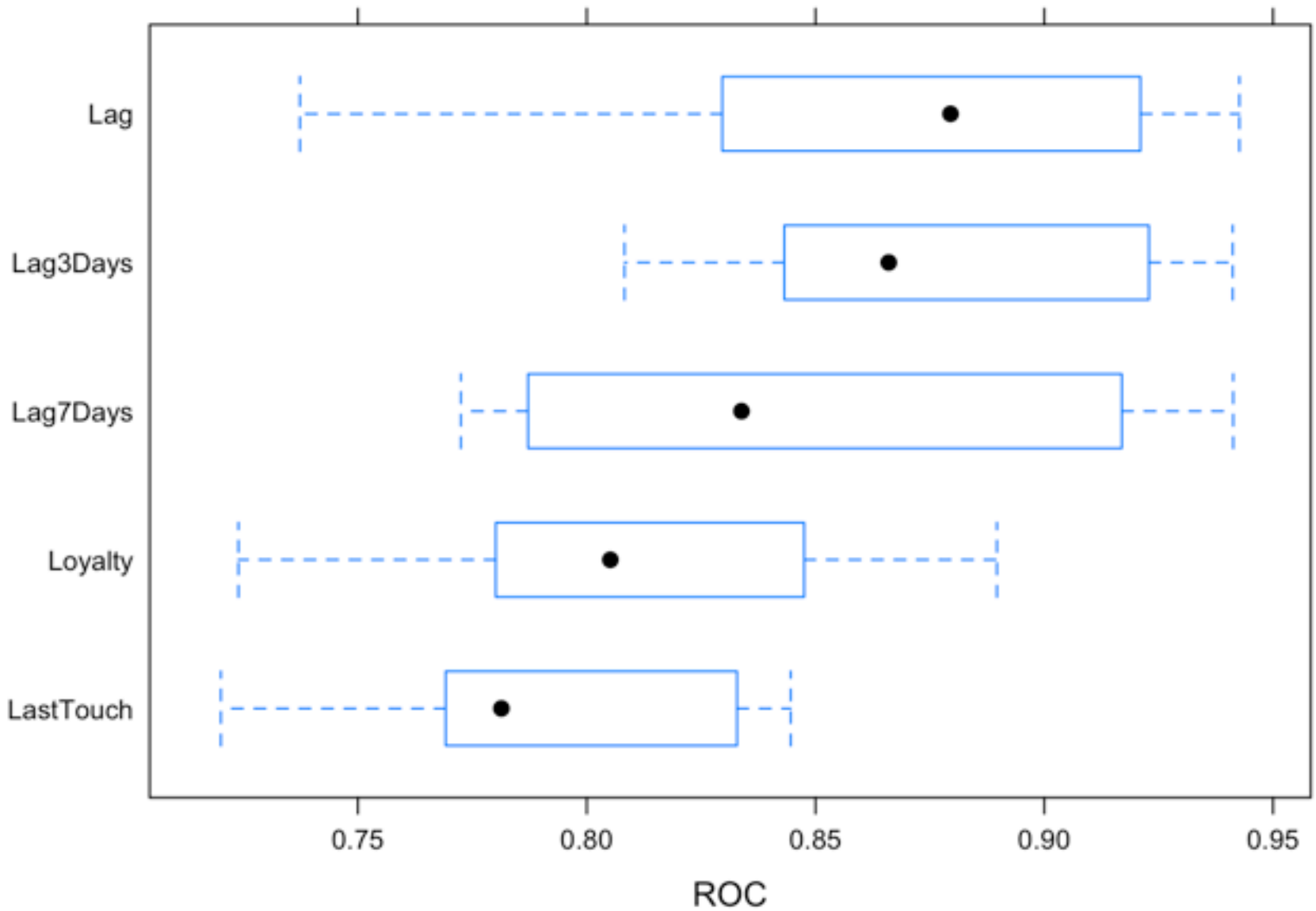
```
# Set parameters for testing models
# Use ROC as the metric to compare models
cmetric <- "ROC"
# Note: Due to imbalanced output, accuracy is not an ideal metric for comparing models

# Model1 - only current touch point
set.seed(1234)
model1 <- train(Purchase ~ Current_unknown + Current_direct + Current_disp + Current_email + Current_organic + Current_paid, data = bt57train, method = "glmnet", family = "binomial" , metric=cmetric, trControl = trainControl("cv", number = 10, summaryFunction=twoClassSummary, classProbs = TRUE), tuneLength = 10)

# Make predictions on Test data
probsTestmodel1 <- predict(model1, type="prob", newdata=bt57test)
predsmodel1 <-probsTestmodel1[,1]
LastTouch <-probsTestmodel1[,1] # repeat above to give proper name for chart

# Model2 - current touch point and loyalty
set.seed(1234)
model2 <- train(Purchase ~ Loyalty + Current_unknown + Current_direct + Current_disp + Current_email + Current_organic + Current_paid, data = bt57train, method = "glmnet"
```

```
, family = "binomial" , metric=cmetric, trControl = trainControl("cv", number = 10, s
ummaryFunction=twoClassSummary, classProbs = TRUE), tuneLength = 10)

# Make predictions on Test data
probsTestmodel2 <- predict(model2, type="prob", newdata=bt57test)
predsmodel2 <-probsTestmodel2[,1]
Loyalty <-probsTestmodel2[,1]

# Model5 - loyalty, current touch point, all 3 lags (3 days) - separated by channel -
excluded null from lag1 and Lag2 as they do not have any nulls
set.seed(1234)
model5_1 <- train(Purchase ~ Loyalty + Current_unknown + Current_direct + Current_dis
p + Current_email + Current_organic + Current_paid + Lag1_3days_direct + Lag1_3days_d
isp + Lag1_3days_email + Lag1_3days_organic + Lag1_3days_paid + Lag1_3days_unknown +
Lag2_3days_direct + Lag2_3days_disp + Lag2_3days_email + Lag2_3days_organic + Lag2_3d
ays_paid + Lag2_3days_unknown + Lag3_3days_direct + Lag3_3days_disp + Lag3_3days_emai
l + Lag3_3days_organic + Lag3_3days_paid + Lag3_3days_unknown + Lag3_3days_null, data
= bt57train, method = "glmnet", na.action = na.omit, family = "binomial" , metric=cme
tric, trControl = trainControl("cv", number = 10, summaryFunction=twoClassSummary, cl
assProbs = TRUE), tuneLength = 10)

# Make predictions on Test data
probsTestmodel5_1 <- predict(model5_1, type="prob", newdata=bt57test)
predsmodel5_1 <-probsTestmodel5_1[,1]
Lag3Days <-probsTestmodel5_1[,1]

# Model8 - loyalty, current touch point, all 3 lags (7 days) - separated by channel -
Excluded null from all 3 lags as they do not have nulls
set.seed(1234)
model8_1 <- train(Purchase ~ Loyalty + Current_unknown + Current_direct + Current_dis
p + Current_email + Current_organic + Current_paid + Lag1_7days_direct + Lag1_7days_d
isp + Lag1_7days_email + Lag1_7days_organic + Lag1_7days_paid + Lag1_7days_unknown +
Lag2_7days_direct + Lag2_7days_disp + Lag2_7days_email + Lag2_7days_organic + Lag2_7d
ays_paid + Lag2_7days_unknown + Lag3_7days_direct + Lag3_7days_disp + Lag3_7days_emai
l + Lag3_7days_organic + Lag3_7days_paid + Lag3_7days_unknown, data = bt57train, meth
od = "glmnet", na.action = na.omit, family = "binomial" , metric=cmetric, trControl =
trainControl("cv", number = 10, summaryFunction=twoClassSummary, classProbs = TRUE),
tuneLength = 10)

# Make predictions on Test data
probsTestmodel8_1 <- predict(model8_1, type="prob", newdata=bt57test)
predsmodel8_1 <-probsTestmodel8_1[,1]
Lag7Days <-probsTestmodel8_1[,1]


# ModelX - loyalty, current touch point, all 3 lags (unfiltered) - separated by chann
el - Excluded null from all 3 lags as they donot have any nulls.
set.seed(1234)
modelX_1 <- train(Purchase ~ Loyalty + Current_unknown + Current_direct + Current_dis
p + Current_email + Current_organic + Current_paid + Lag1direct + Lag1disp + Lag1emai
```

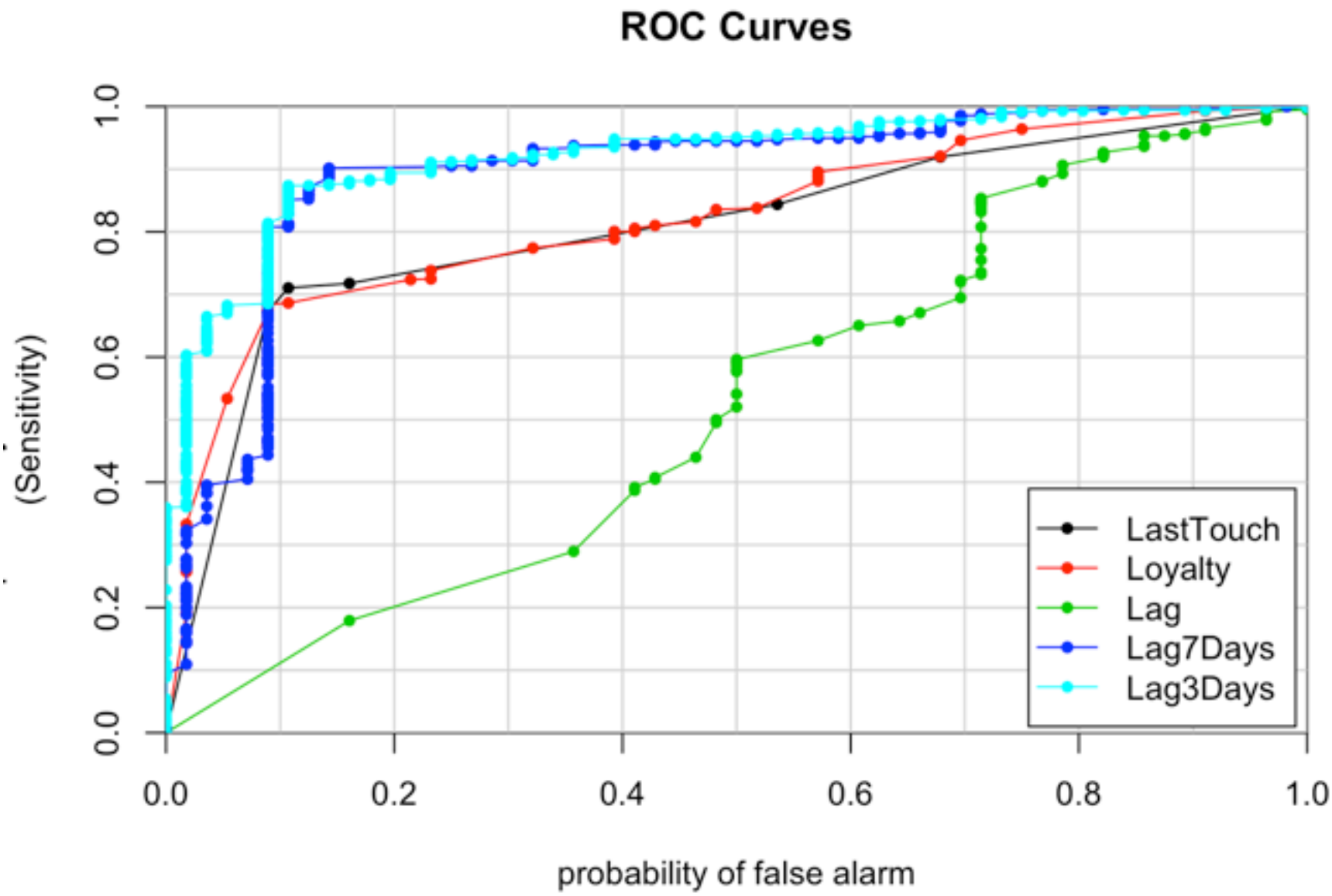
```
l + Lag1organic + Lag1paid + Lag1unknown + Lag2direct + Lag2disp + Lag2email + Lag2or
ganic + Lag2paid + Lag2unknown + Lag3direct + Lag3displ + Lag3email + Lag3organic + L
ag3paid + Lag3unknown, data = bt57train, method = "glmnet", na.action = na.omit, fami
ly = "binomial" , metric=cmetric, trControl = trainControl("cv", number = 10, summary
Function=twoClassSummary, classProbs = TRUE), tuneLength = 10)

# Make predictions on Test data
probsTestmodelX_1 <- predict(modelX_1, type="prob", newdata=bt57test)
predsmodelX_1 <-probsTestmodelX_1[,1]
Lag <-probsTestmodelX_1[,1]



########### Compare models

set.seed(1234)
models <- list(LastTouch = model1, Loyalty = model2, Lag = modelX_1, Lag7Days = model
8_1, Lag3Days = model5_1)
results <- resamples(models)
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: LastTouch, Loyalty, Lag, Lag7Days, Lag3Days
## Number of resamples: 10
##
## ROC
##                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LastTouch 0.7200567 0.7692941 0.7813494 0.7872124 0.8276751 0.8445882    0
## Loyalty   0.7238750 0.7819118 0.8051531 0.8077394 0.8459412 0.8896471    0
## Lag       0.7373206 0.8311147 0.8794999 0.8671790 0.9174370 0.9426511    0
## Lag7Days  0.7725000 0.7926765 0.8338318 0.8473805 0.9063529 0.9412941    0
## Lag3Days  0.8082500 0.8432353 0.8660000 0.8734791 0.9175230 0.9411765    0
##
## Sens
##                Min.    1st Qu.    Median      Mean   3rd Qu.      Max.
## LastTouch 0.00000000 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000
## Loyalty   0.00000000 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000
## Lag       0.00000000 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000
## Lag7Days  0.00000000 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000
## Lag3Days  0.05882353 0.07628676 0.1470588 0.1709559 0.2352941 0.4117647
##           NA's
## LastTouch    0
## Loyalty      0
## Lag          0
## Lag7Days     0
## Lag3Days     0
##
## Spec
##                Min. 1st Qu. Median      Mean  3rd Qu. Max. NA's
## LastTouch 1.0000000   1.000  1.000 1.0000000 1.000000    1    0
## Loyalty   1.0000000   1.000  1.000 1.0000000 1.000000    1    0
## Lag       1.0000000   1.000  1.000 1.0000000 1.000000    1    0
## Lag7Days  1.0000000   1.000  1.000 1.0000000 1.000000    1    0
## Lag3Days  0.9799197   0.988  0.992 0.9915904 0.995996    1    0
```

```
bwplot(results, metric="ROC") # boxplots to compare ROCs from 10 resamples
```




```
# Compare ROC curves
colAUC(cbind(LastTouch, Loyalty, Lag, Lag7Days, Lag3Days), bt57test$Purchase, plotROC
=TRUE)
```




```
##                            LastTouch   Loyalty       Lag  Lag7Days  Lag3Days
## Purchase vs. No_Purchase 0.8078211 0.8224588 0.5241458 0.8940591 0.9191492
```

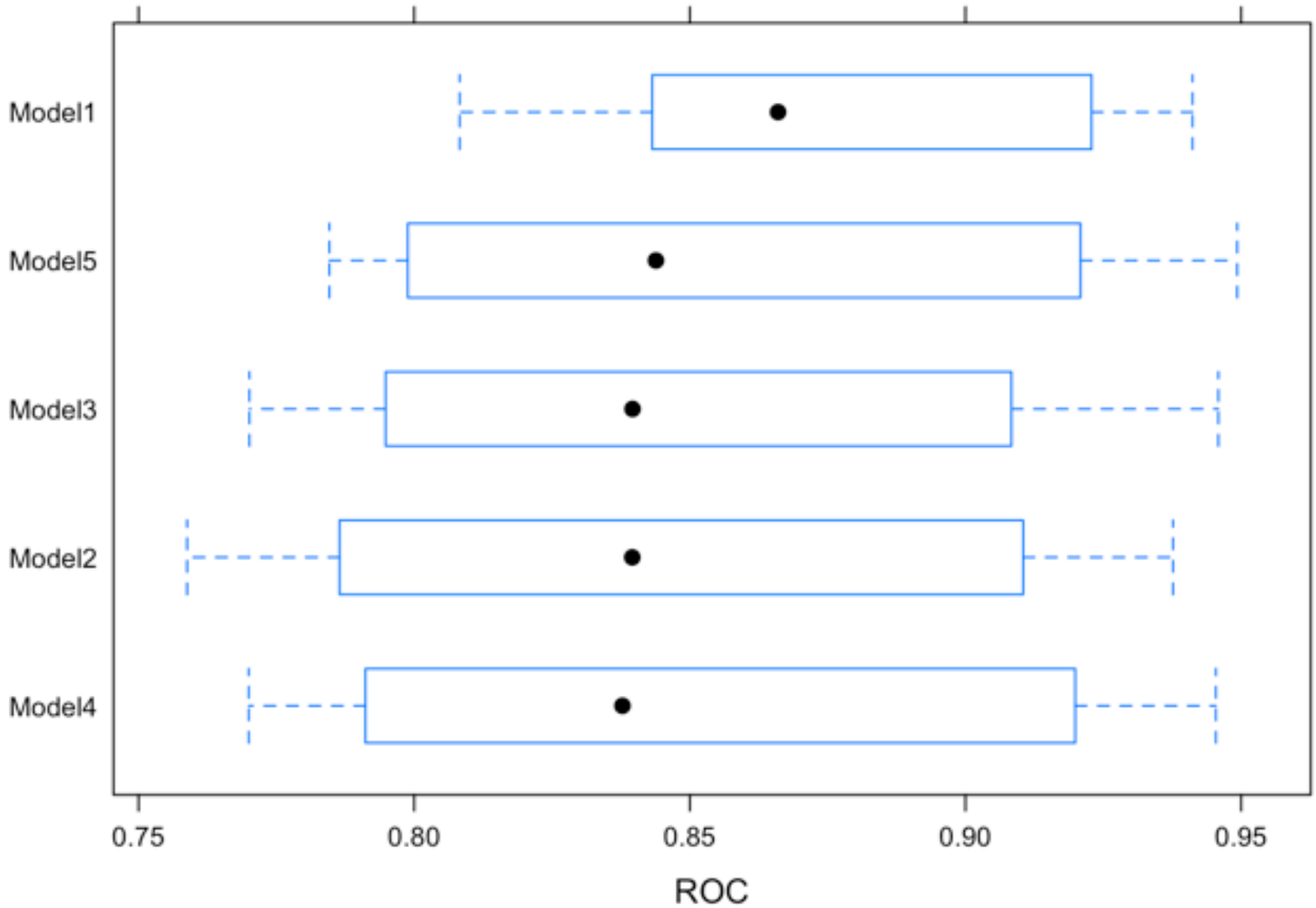
Fine tune the best performing model

```
# Model5 - loyalty, current touch point, all 3 lags (3 days) - separated by channel - remove unknown
set.seed(1234)
model5_2 <- train(Purchase ~ Loyalty + Current_direct + Current_disp + Current_email + Current_organic + Current_paid + Lag1_3days_direct + Lag1_3days_disp + Lag1_3days_email + Lag1_3days_organic + Lag1_3days_paid + Lag2_3days_direct + Lag2_3days_disp + Lag2_3days_email + Lag2_3days_organic + Lag2_3days_paid + Lag3_3days_direct + Lag3_3days_disp + Lag3_3days_email + Lag3_3days_organic + Lag3_3days_paid + Lag3_3days_null, data = bt57train, method = "glmnet", na.action = na.omit, family = "binomial" , metric=cmetric, trControl = trainControl("cv", number = 10, summaryFunction=twoClassSummary, classProbs = TRUE), tuneLength = 10)

# Make predictions on Test data
probsTestmodel5_2 <- predict(model5_2, type="prob", newdata=bt57test)
predsmodel5_2 <-probsTestmodel5_2[,1]

# Model5 - loyalty, current touch point, first 2 lags (3 days) - separated by channel
```

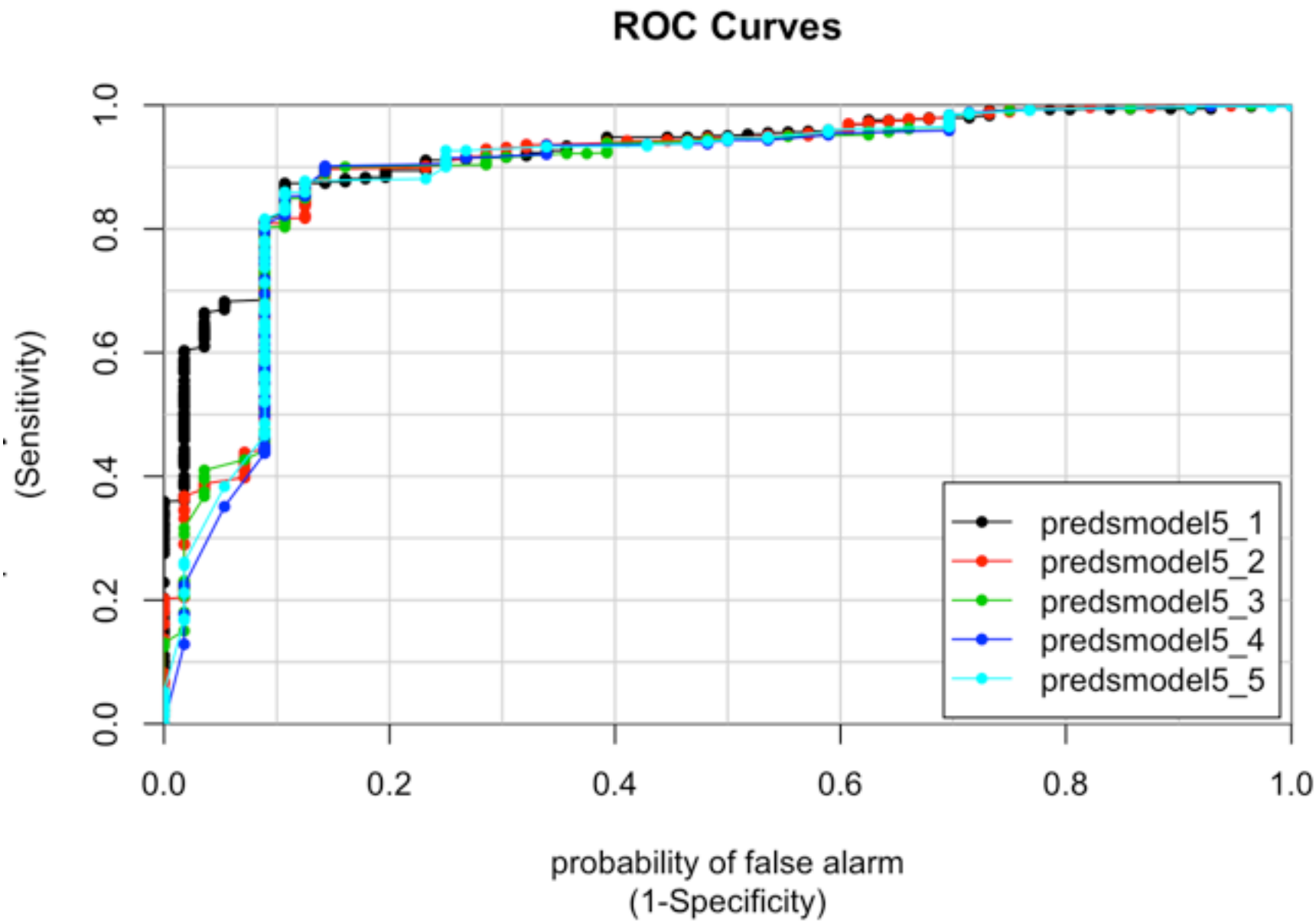
```
- remove unknown
set.seed(1234)
model5_3 <- train(Purchase ~ Loyalty + Current_direct + Current_disp + Current_email
+ Current_organic + Current_paid + Lag1_3days_direct + Lag1_3days_disp + Lag1_3days_e
mail + Lag1_3days_organic + Lag1_3days_paid + Lag2_3days_direct + Lag2_3days_disp + L
ag2_3days_email + Lag2_3days_organic + Lag2_3days_paid, data = bt57train, method = "g
lmnet", na.action = na.omit, family = "binomial" , metric=cmetric, trControl = trainC
ontrol("cv", number = 10, summaryFunction=twoClassSummary, classProbs = TRUE), tuneLe
ngth = 10)

# Make predictions on Test data
probsTestmodel5_3 <- predict(model5_3, type="prob", newdata=bt57test)
predsmodel5_3 <-probsTestmodel5_3[,1]

# Model5 - loyalty, current touch point, first lag (3 days) - separated by channel -
remove unknown and null
set.seed(1234)
model5_4 <- train(Purchase ~ Loyalty + Current_direct + Current_disp + Current_email
+ Current_organic + Current_paid + Lag1_3days_direct + Lag1_3days_disp + Lag1_3days_e
mail + Lag1_3days_organic + Lag1_3days_paid, data = bt57train, method = "glmnet", na.
action = na.omit, family = "binomial" , metric=cmetric, trControl = trainControl("cv"
, number = 10, summaryFunction=twoClassSummary, classProbs = TRUE), tuneLength = 10)

# Make predictions on Test data
probsTestmodel5_4 <- predict(model5_4, type="prob", newdata=bt57test)
predsmodel5_4 <-probsTestmodel5_4[,1]

# Model5 - loyalty, current touch point, first lag (3 days) - separated by channel -
KEEP unknown and null
set.seed(1234)
model5_5 <- train(Purchase ~ Loyalty + Current_unknown + Current_direct + Current_dis
p + Current_email + Current_organic + Current_paid + Lag1_3days_direct + Lag1_3days_d
isp + Lag1_3days_email + Lag1_3days_organic + Lag1_3days_paid + Lag1_3days_unknown, d
ata = bt57train, method = "glmnet", na.action = na.omit, family = "binomial" , metric
=cmetric, trControl = trainControl("cv", number = 10, summaryFunction=twoClassSummary
, classProbs = TRUE), tuneLength = 10)

# Make predictions on Test data
probsTestmodel5_5 <- predict(model5_5, type="prob", newdata=bt57test)
predsmodel5_5 <-probsTestmodel5_5[,1]

########### Compare models

set.seed(1234)
models <- list(model5_1, model5_2, model5_3, model5_4, model5_5)
results <- resamples(models)
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: Model1, Model2, Model3, Model4, Model5
## Number of resamples: 10
##
## ROC
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## Model1 0.8082500 0.8432353 0.8660000 0.8734791 0.9175230 0.9411765    0
## Model2 0.7587999 0.7902353 0.8395533 0.8437188 0.9005255 0.9376471    0
## Model3 0.7701250 0.7975294 0.8396010 0.8475394 0.8989015 0.9458824    0
## Model4 0.7700000 0.7947853 0.8377746 0.8486901 0.9094950 0.9454118    0
## Model5 0.7846250 0.8040142 0.8438462 0.8549196 0.9107920 0.9492941    0
##
## Sens
##              Min.    1st Qu.    Median       Mean    3rd Qu.      Max.
## Model1 0.05882353 0.07628676 0.1470588 0.17095588 0.23529412 0.4117647
## Model2 0.00000000 0.00000000 0.0000000 0.00000000 0.00000000 0.0000000
## Model3 0.00000000 0.00000000 0.0000000 0.00000000 0.00000000 0.0000000
## Model4 0.00000000 0.00000000 0.0000000 0.01250000 0.00000000 0.1250000
## Model5 0.00000000 0.00000000 0.0000000 0.03014706 0.04411765 0.1250000
##        NA's
## Model1    0
## Model2    0
## Model3    0
## Model4    0
## Model5    0
##
## Spec
##             Min.   1st Qu. Median      Mean  3rd Qu. Max. NA's
## Model1 0.9799197 0.9880000  0.992 0.9915904 0.995996    1    0
## Model2 0.9920000 0.9960000  1.000 0.9980000 1.000000    1    0
## Model3 1.0000000 1.0000000  1.000 1.0000000 1.000000    1    0
## Model4 0.9880000 0.9970000  1.000 0.9979984 1.000000    1    0
## Model5 0.9880000 0.9919759  1.000 0.9959968 1.000000    1    0
```

```
bwplot(results, metric="ROC") # boxplots to compare ROCs from 10 resamples
```




```
# Compare ROC curves
colAUC(cbind(predsmodel5_1, predsmodel5_2, predsmodel5_3, predsmodel5_4, predsmodel5_
5), bt57test$Purchase, plotROC=TRUE)
```




```
##                           predsmodel5_1 predsmodel5_2 predsmodel5_3
## Purchase vs. No_Purchase     0.9191492     0.8983946     0.8939196
##                           predsmodel5_4 predsmodel5_5
## Purchase vs. No_Purchase      0.889187     0.8919986
```

```
# The original model including Loyalty, Lasttouch and the 3 lags (3 days) is still th
e best model
```

Model Fit

```
# Compare the fit of the first 4 models

# Set probability threshold for classifcation
threshold <- 0.3

probsmodel1 <- predict(model1, type="prob", newdata=bt57)
predmodel1 <- factor(ifelse(probsmodel1[, "Purchase"] > threshold, 1, 0))

probsmodel2 <- predict(model2, type="prob", newdata=bt57)
predmodel2 <- factor(ifelse(probsmodel2[, "Purchase"] > threshold, 1, 0))

probsmodel5_1 <- predict(model5_1, type="prob", newdata=bt57)
```

```
predmodel5_1 <- factor(ifelse(probsmodel5_1[, "Purchase"] > threshold, 1, 0))

#probsmodel5_2 <- predict(model5_2, type="prob", newdata=bt57)
#predmodel5_2 <- factor(ifelse(probsmodel5_2[, "Purchase"] > threshold, 1, 0))

probsmodel8_1 <- predict(model8_1, type="prob", newdata=bt57)
predmodel8_1 <- factor(ifelse(probsmodel8_1[, "Purchase"] > threshold, 1, 0))

# add the predictions to the data
btd_06_02_preds <- btd_06_02 %>%
        mutate(Model1 = predmodel1, Model2 = predmodel2, Model5_1 = predmodel5_1, Mod
el8_1 = predmodel8_1) %>%
        select(ID, Purchase, Direct, Disp, Email, Organic, Paid, Model1, Model2, Mode
l5_1, Model8_1)

# Frequency of use of each channel at last touchpoint - observed
bt57visits_pred <- btd_06_02_preds %>%
        summarize(Direct = sum(Direct), Disp = sum(Disp), Email = sum(Email), Organic
= sum(Organic), Paid = sum(Paid)) %>%
        gather(key = Channel) %>%
        #arrange(-value) %>%
        rename("Channel Visits" = "value")

# Frequency of use of each channel at last touchpoint (Purchases only) - observed
bt57P_pred <- btd_06_02_preds %>%
        filter(Purchase == 1) %>%
        summarize(Direct = sum(Direct), Disp = sum(Disp), Email = sum(Email), Organic
= sum(Organic), Paid = sum(Paid)) %>%
        gather(key = Channel) %>%
        #arrange(-value) %>%
        rename("Observed" = "value")

# Frequency of use of each channel at last touchpoint (Purchases only) - Model1
bt57M1 <- btd_06_02_preds %>%
        filter(Model1 == 1) %>%
        summarize(Direct = sum(Direct), Disp = sum(Disp), Email = sum(Email), Organic
= sum(Organic), Paid = sum(Paid)) %>%
        gather(key = Channel) %>%
        #arrange(-value) %>%
        rename("Model1" = "value")

# Frequency of use of each channel at last touchpoint (Purchases only) - Model2
bt57M2 <- btd_06_02_preds %>%
        filter(Model2 == 1) %>%
        summarize(Direct = sum(Direct), Disp = sum(Disp), Email = sum(Email), Organic
= sum(Organic), Paid = sum(Paid)) %>%
        gather(key = Channel) %>%
        #arrange(-value) %>%
        rename("Model2" = "value")
```

```
# Frequency of use of each channel at last touchpoint (Purchases only) - Model5_1
bt57M5_1 <- btd_06_02_preds %>%
        filter(Model5_1 == 1) %>%
        summarize(Direct = sum(Direct), Disp = sum(Disp), Email = sum(Email), Organic
= sum(Organic), Paid = sum(Paid)) %>%
        gather(key = Channel) %>%
        #arrange(-value) %>%
        rename("Model5_1" = "value")

# Frequency of use of each channel at last touchpoint (Purchases only) - Model8_1
bt57M8_1 <- btd_06_02_preds %>%
        filter(Model8_1 == 1) %>%
        summarize(Direct = sum(Direct), Disp = sum(Disp), Email = sum(Email), Organic
= sum(Organic), Paid = sum(Paid)) %>%
        gather(key = Channel) %>%
        #arrange(-value) %>%
        rename("Model8_1" = "value")


# Calculate Mean Absolute Percent Error
bt57sumstats_pred <- bt57visits_pred %>%
        left_join(bt57P_pred) %>%
        left_join(bt57M1) %>%
        left_join(bt57M2) %>%
        left_join(bt57M5_1) %>%
        left_join(bt57M8_1) %>%
        mutate(MAPE_5_1 = ((abs(Model5_1 - Observed)) / Observed) * 100, MAPE_8_1 = (
(abs(Model8_1 - Observed)) / Observed) * 100)


# SUMMARY STATISTICS
knitr::kable(bt57sumstats_pred)
```

Channel	Channel Visits	Observed	Model1	Model2	Model5_1	Model8_1	MAPE_5_1	MAPE_8_1
Direct	335	71	0	0	123	107	73.239437	50.70423
Disp	57	7	0	0	3	0	57.142857	100.00000
Email	132	8	0	0	2	0	75.000000	100.00000
Organic	575	77	0	0	74	2	3.896104	97.40260
Paid	289	38	0	0	45	21	18.421053	44.73684

Model coefficients

```
# Run model 1 (LastTouch) with complete data
out_model1_1 <- glm(Purchase ~ Current_unknown + Current_direct + Current_disp + Curr
ent_email + Current_organic + Current_paid, btd_06_02,  family = "binomial")
summary(out_model1_1)
```

```
##
## Call:
## glm(formula = Purchase ~ Current_unknown + Current_direct + Current_disp +
##     Current_email + Current_organic + Current_paid, family = "binomial",
##     data = btd_06_02)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6902  -0.5310  -0.1493  -0.1493   3.0009
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.88787    0.17407 -10.846  < 2e-16 ***
## Current_unknown -2.60357    0.26913  -9.674  < 2e-16 ***
## Current_direct   0.57460    0.21948   2.618  0.00885 **
## Current_disp    -0.07825    0.43950  -0.178  0.85869
## Current_email   -0.85297    0.40418  -2.110  0.03483 *
## Current_organic  0.02107    0.21283   0.099  0.92113
## Current_paid          NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1677.3  on 3553  degrees of freedom
## Residual deviance: 1390.5  on 3548  degrees of freedom
## AIC: 1402.5
##
## Number of Fisher Scoring iterations: 7
```

```
# Run model 2 (LastTouch + Loyalty) with complete data
out_model2_1 <- glm(Purchase ~ Loyalty + Current_unknown + Current_direct + Current_d
isp + Current_email + Current_organic + Current_paid, btd_06_02,  family = "binomial"
)
summary(out_model2_1)
```

```
##
## Call:
## glm(formula = Purchase ~ Loyalty + Current_unknown + Current_direct +
##     Current_disp + Current_email + Current_organic + Current_paid,
##     family = "binomial", data = btd_06_02)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7937  -0.4261  -0.1603  -0.1246   3.1654
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -2.353790   0.221472 -10.628  < 2e-16 ***
## Loyalty01-Basic     0.829832   0.214262   3.873 0.000108 ***
## Loyalty02-Silver    0.655634   0.212044   3.092 0.001988 **
## Loyalty03-Gold      0.148927   0.338946   0.439 0.660383
## Loyalty04-Platinum  0.581329   0.265021   2.194 0.028270 *
## Current_unknown    -2.649480   0.271882  -9.745  < 2e-16 ***
## Current_direct      0.530343   0.227442   2.332 0.019713 *
## Current_disp       -0.172952   0.442744  -0.391 0.696067
## Current_email      -0.994288   0.409947  -2.425 0.015291 *
## Current_organic    -0.009339   0.215623  -0.043 0.965453
## Current_paid              NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1677.3  on 3553  degrees of freedom
## Residual deviance: 1371.4  on 3544  degrees of freedom
## AIC: 1391.4
##
## Number of Fisher Scoring iterations: 7
```

```
# Run model 5 with complete data
out_model5_1 <- glm(Purchase ~ Loyalty + Current_unknown + Current_direct + Current_d
isp + Current_email + Current_organic + Current_paid + Lag1_3days_direct + Lag1_3days
_disp + Lag1_3days_email + Lag1_3days_organic + Lag1_3days_paid + Lag1_3days_unknown
+ Lag2_3days_direct + Lag2_3days_disp + Lag2_3days_email + Lag2_3days_organic + Lag2_
3days_paid + Lag2_3days_unknown + Lag3_3days_direct + Lag3_3days_disp + Lag3_3days_em
ail + Lag3_3days_organic + Lag3_3days_paid + Lag3_3days_null + Lag3_3days_unknown, bt
d_06_02,  family = "binomial")
summary(out_model5_1)
```

```
##
## Call:
## glm(formula = Purchase ~ Loyalty + Current_unknown + Current_direct +
##     Current_disp + Current_email + Current_organic + Current_paid +
```

```
##     Lag1_3days_direct + Lag1_3days_disp + Lag1_3days_email +
##     Lag1_3days_organic + Lag1_3days_paid + Lag1_3days_unknown +
##     Lag2_3days_direct + Lag2_3days_disp + Lag2_3days_email +
##     Lag2_3days_organic + Lag2_3days_paid + Lag2_3days_unknown +
##     Lag3_3days_direct + Lag3_3days_disp + Lag3_3days_email +
##     Lag3_3days_organic + Lag3_3days_paid + Lag3_3days_null +
##     Lag3_3days_unknown, family = "binomial", data = btd_06_02)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4940  -0.2528  -0.1268  -0.0495   3.2243
##
## Coefficients: (1 not defined because of singularities)
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -21.57317  419.94474  -0.051 0.959030
## Loyalty01-Basic      0.87703    0.23091   3.798 0.000146 ***
## Loyalty02-Silver     0.81581    0.23701   3.442 0.000577 ***
## Loyalty03-Gold       0.30559    0.36916   0.828 0.407779
## Loyalty04-Platinum   0.64426    0.29542   2.181 0.029199 *
## Current_unknown     -2.33955    0.29811  -7.848 4.23e-15 ***
## Current_direct       1.10462    0.31606   3.495 0.000474 ***
## Current_disp         0.14203    0.58515   0.243 0.808216
## Current_email       -0.37894    0.49290  -0.769 0.442016
## Current_organic     -0.13762    0.25117  -0.548 0.583741
## Current_paid              NA         NA      NA       NA
## Lag1_3days_direct   17.22592  419.94447   0.041 0.967280
## Lag1_3days_disp     17.47376  419.94473   0.042 0.966810
## Lag1_3days_email    16.66551  419.94479   0.040 0.968344
## Lag1_3days_organic  17.44210  419.94449   0.042 0.966870
## Lag1_3days_paid     17.40132  419.94452   0.041 0.966947
## Lag1_3days_unknown  16.28264  419.94450   0.039 0.969071
## Lag2_3days_direct   -0.10124    0.45220  -0.224 0.822854
## Lag2_3days_disp     -1.46993    1.12871  -1.302 0.192810
## Lag2_3days_email     0.03766    0.73309   0.051 0.959029
## Lag2_3days_organic   0.10915    0.45569   0.240 0.810703
## Lag2_3days_paid     -0.16558    0.49073  -0.337 0.735811
## Lag2_3days_unknown   0.23580    0.44040   0.535 0.592357
## Lag3_3days_direct    2.53865    0.43736   5.804 6.46e-09 ***
## Lag3_3days_disp      1.73162    1.17724   1.471 0.141312
## Lag3_3days_email     1.78588    0.85252   2.095 0.036188 *
## Lag3_3days_organic   2.71230    0.38806   6.989 2.76e-12 ***
## Lag3_3days_paid      2.33164    0.45172   5.162 2.45e-07 ***
## Lag3_3days_null      2.66138    0.45094   5.902 3.60e-09 ***
## Lag3_3days_unknown   2.63278    0.33924   7.761 8.44e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1677.3  on 3553  degrees of freedom
```

```
## Residual deviance: 1078.6  on 3525  degrees of freedom
## AIC: 1136.6
##
## Number of Fisher Scoring iterations: 17
```

```
# without the lag_null & unknown
out_model5_1_X <- glm(Purchase ~ Loyalty + Current_direct + Current_disp + Current_em
ail + Current_organic + Current_paid + Lag1_3days_direct + Lag1_3days_disp + Lag1_3da
ys_email + Lag1_3days_organic + Lag1_3days_paid + Lag2_3days_direct + Lag2_3days_disp
+ Lag2_3days_email + Lag2_3days_organic + Lag2_3days_paid + Lag3_3days_direct + Lag3_
3days_disp + Lag3_3days_email + Lag3_3days_organic + Lag3_3days_paid, btd_06_02,  fam
ily = "binomial")
summary(out_model5_1_X)
```

```
##
## Call:
## glm(formula = Purchase ~ Loyalty + Current_direct + Current_disp +
##     Current_email + Current_organic + Current_paid + Lag1_3days_direct +
##     Lag1_3days_disp + Lag1_3days_email + Lag1_3days_organic +
##     Lag1_3days_paid + Lag2_3days_direct + Lag2_3days_disp + Lag2_3days_email +
##     Lag2_3days_organic + Lag2_3days_paid + Lag3_3days_direct +
##     Lag3_3days_disp + Lag3_3days_email + Lag3_3days_organic +
##     Lag3_3days_paid, family = "binomial", data = btd_06_02)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3526  -0.2774  -0.1619  -0.0822   3.5897
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -6.441535   0.317081 -20.315  < 2e-16 ***
## Loyalty01-Basic     0.849626   0.226967   3.743 0.000182 ***
## Loyalty02-Silver    0.780350   0.229320   3.403 0.000667 ***
## Loyalty03-Gold      0.198327   0.358638   0.553 0.580264
## Loyalty04-Platinum  0.462446   0.284429   1.626 0.103976
## Current_direct      3.331485   0.281406  11.839  < 2e-16 ***
## Current_disp        2.437835   0.527697   4.620 3.84e-06 ***
## Current_email       2.101737   0.464440   4.525 6.03e-06 ***
## Current_organic     2.353231   0.253871   9.269  < 2e-16 ***
## Current_paid        2.642237   0.295692   8.936  < 2e-16 ***
## Lag1_3days_direct   2.109837   0.253652   8.318  < 2e-16 ***
## Lag1_3days_disp     2.482410   0.528962   4.693 2.69e-06 ***
## Lag1_3days_email    1.254310   0.549054   2.284 0.022343 *
## Lag1_3days_organic  2.113941   0.241598   8.750  < 2e-16 ***
## Lag1_3days_paid     2.222395   0.277736   8.002 1.23e-15 ***
## Lag2_3days_direct   0.160541   0.276685   0.580 0.561759
## Lag2_3days_disp    -1.305183   1.102996  -1.183 0.236687
## Lag2_3days_email    0.358954   0.611810   0.587 0.557399
## Lag2_3days_organic  0.234558   0.232463   1.009 0.312969
```

```
## Lag2_3days_paid    -0.049677   0.297883  -0.167 0.867555
## Lag3_3days_direct   0.462225   0.330133   1.400 0.161479
## Lag3_3days_disp    -0.504443   1.142934  -0.441 0.658954
## Lag3_3days_email   -0.358837   0.816148  -0.440 0.660175
## Lag3_3days_organic  0.554283   0.251987   2.200 0.027832 *
## Lag3_3days_paid     0.004824   0.336581   0.014 0.988564
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1677.3  on 3553  degrees of freedom
## Residual deviance: 1179.5  on 3529  degrees of freedom
## AIC: 1229.5
##
## Number of Fisher Scoring iterations: 7
```

```
# Run model 5 with complete data - only 1st lag
out_model5_1_L1 <- glm(Purchase ~ Loyalty + Current_unknown + Current_direct + Curren
t_disp + Current_email + Current_organic + Current_paid + Lag1_3days_direct + Lag1_3d
ays_disp + Lag1_3days_email + Lag1_3days_organic + Lag1_3days_paid + Lag1_3days_unkno
wn, btd_06_02,  family = "binomial")
summary(out_model5_1_L1)
```

```
##
## Call:
## glm(formula = Purchase ~ Loyalty + Current_unknown + Current_direct +
##     Current_disp + Current_email + Current_organic + Current_paid +
##     Lag1_3days_direct + Lag1_3days_disp + Lag1_3days_email +
##     Lag1_3days_organic + Lag1_3days_paid + Lag1_3days_unknown,
##     family = "binomial", data = btd_06_02)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1910  -0.2709  -0.1668  -0.0709   3.5214
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -18.6108   420.9334  -0.044 0.964735
## Loyalty01-Basic      0.8154     0.2249   3.625 0.000288 ***
## Loyalty02-Silver     0.7895     0.2265   3.486 0.000491 ***
## Loyalty03-Gold       0.2142     0.3553   0.603 0.546696
## Loyalty04-Platinum   0.5338     0.2795   1.910 0.056123 .
## Current_unknown     -2.5809     0.2865  -9.008  < 2e-16 ***
## Current_direct       0.7372     0.2826   2.608 0.009102 **
## Current_disp        -0.1661     0.5198  -0.320 0.749274
## Current_email       -0.5029     0.4549  -1.105 0.268949
## Current_organic     -0.1913     0.2320  -0.824 0.409688
## Current_paid            NA         NA      NA       NA
## Lag1_3days_direct   16.9962   420.9334   0.040 0.967792
## Lag1_3days_disp     17.1379   420.9336   0.041 0.967524
## Lag1_3days_email    16.1081   420.9337   0.038 0.969474
## Lag1_3days_organic  17.0896   420.9334   0.041 0.967615
## Lag1_3days_paid     17.0902   420.9334   0.041 0.967614
## Lag1_3days_unknown  14.9937   420.9334   0.036 0.971585
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1677.3  on 3553  degrees of freedom
## Residual deviance: 1180.5  on 3538  degrees of freedom
## AIC: 1212.5
##
## Number of Fisher Scoring iterations: 17
```

```
# Run model 5 with complete data - only 3rd lag
out_model5_1_L3 <- glm(Purchase ~ Loyalty + Current_unknown + Current_direct + Curren
t_disp + Current_email + Current_organic + Current_paid + Lag3_3days_direct + Lag3_3d
ays_disp + Lag3_3days_email + Lag3_3days_organic + Lag3_3days_paid + Lag3_3days_null
+ Lag3_3days_unknown + Lag3_3days_null, btd_06_02,  family = "binomial")
summary(out_model5_1_L3)
```

```
##
## Call:
## glm(formula = Purchase ~ Loyalty + Current_unknown + Current_direct +
##     Current_disp + Current_email + Current_organic + Current_paid +
##     Lag3_3days_direct + Lag3_3days_disp + Lag3_3days_email +
##     Lag3_3days_organic + Lag3_3days_paid + Lag3_3days_null +
##     Lag3_3days_unknown + Lag3_3days_null, family = "binomial",
##     data = btd_06_02)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2420  -0.2690  -0.1574  -0.0569   3.1896
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -4.82465    0.36934 -13.063  < 2e-16 ***
## Loyalty01-Basic     0.83403    0.22444   3.716 0.000202 ***
## Loyalty02-Silver    0.76788    0.22662   3.388 0.000703 ***
## Loyalty03-Gold      0.34769    0.35739   0.973 0.330625
## Loyalty04-Platinum  0.62932    0.28428   2.214 0.026848 *
## Current_unknown    -2.36939    0.28242  -8.389  < 2e-16 ***
## Current_direct      1.01591    0.25984   3.910 9.24e-05 ***
## Current_disp        0.25063    0.47593   0.527 0.598459
## Current_email      -0.56353    0.43462  -1.297 0.194773
## Current_organic     0.06417    0.22800   0.281 0.778382
## Current_paid             NA         NA      NA       NA
## Lag3_3days_direct   2.80858    0.39422   7.124 1.05e-12 ***
## Lag3_3days_disp     1.80412    1.11246   1.622 0.104860
## Lag3_3days_email    1.98406    0.79889   2.484 0.013009 *
## Lag3_3days_organic  3.09725    0.35433   8.741  < 2e-16 ***
## Lag3_3days_paid     2.78732    0.40754   6.839 7.96e-12 ***
## Lag3_3days_null     2.11349    0.33512   6.307 2.85e-10 ***
## Lag3_3days_unknown  3.12525    0.30891  10.117  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1677.3  on 3553  degrees of freedom
## Residual deviance: 1168.6  on 3537  degrees of freedom
## AIC: 1202.6
##
## Number of Fisher Scoring iterations: 7
```

```
# Test correlation between current and lag1
#bt57cor <- bt57 %>%
#        select(Purchase, Current_direct, Current_disp, Current_email, Current_organi
c, Current_paid, Lag1_3days_direct, Lag1_3days_disp, Lag1_3days_email, Lag1_3days_org
anic, Lag1_3days_paid)

#ggpairs(bt57cor)


# Odds Ratio
knitr::kable(exp(coef(out_model5_1)))
```

	x
(Intercept)	0.000000e+00
Loyalty01-Basic	2.403759e+00
Loyalty02-Silver	2.260997e+00
Loyalty03-Gold	1.357427e+00
Loyalty04-Platinum	1.904574e+00
Current_unknown	9.637130e-02
Current_direct	3.018091e+00
Current_disp	1.152613e+00
Current_email	6.845854e-01
Current_organic	8.714266e-01
Current_paid	NA
Lag1_3days_direct	3.027778e+07
Lag1_3days_disp	3.879338e+07
Lag1_3days_email	1.728770e+07
Lag1_3days_organic	3.758451e+07
Lag1_3days_paid	3.608256e+07
Lag1_3days_unknown	1.178855e+07
Lag2_3days_direct	9.037197e-01
Lag2_3days_disp	2.299407e-01
Lag2_3days_email	1.038379e+00
Lag2_3days_organic	1.115327e+00

Lag2_3days_paid	8.474052e-01
Lag2_3days_unknown	1.265921e+00
Lag3_3days_direct	1.266256e+01
Lag3_3days_disp	5.649802e+00
Lag3_3days_email	5.964800e+00
Lag3_3days_organic	1.506382e+01
Lag3_3days_paid	1.029476e+01
Lag3_3days_null	1.431606e+01
Lag3_3days_unknown	1.391242e+01

Consideration Stage - using Multinomia Logistic Regression

```
# Re-level the Current variable to use 'unknown' as reference
bt57$Current2 <- relevel(as.factor(bt57$Current), ref = "unknown")

cs <- multinom(Current2 ~ Loyalty, data = bt57)
```

```
## # weights:  36 (25 variable)
## initial  value 6367.913154
## iter  10 value 4385.018720
## iter  20 value 4272.519142
## iter  30 value 4219.164027
## final  value 4218.782442
## converged
```

```
summary(cs)
```

```
## Call:
## multinom(formula = Current2 ~ Loyalty, data = bt57)
##
## Coefficients:
##         (Intercept) Loyalty01-Basic Loyalty02-Silver Loyalty03-Gold
## direct    -2.138367     -0.24701367        0.5111148      0.2374848
## disp      -3.524400      0.53288942       -0.7957478    -14.7038812
## email     -4.468747      1.73305216        2.0942937      2.2543381
## organic   -1.297036      0.13138149       -0.2033640      0.1633261
## paid      -1.670142     -0.04511145       -0.9762556     -0.9239797
##         Loyalty04-Platinum
## direct           0.8122020
## disp            -0.1052471
## email            1.5329144
## organic         -0.1722823
## paid            -0.4544898
##
## Std. Errors:
##         (Intercept) Loyalty01-Basic Loyalty02-Silver Loyalty03-Gold
## direct   0.12459973       0.2008571        0.1565906   2.444006e-01
## disp     0.23911450       0.3176991        0.4119864   1.360930e-07
## email    0.38004608       0.4228328        0.4022240   4.503376e-01
## organic  0.08731833       0.1281714        0.1253966   1.767182e-01
## paid     0.10164843       0.1550880        0.1806702   3.049679e-01
##         Loyalty04-Platinum
## direct           0.1913525
## disp             0.4779226
## email            0.4818805
## organic          0.1768621
## paid             0.2275767
##
## Residual Deviance: 8437.565
## AIC: 8487.565
```

```
# calculate p-value
z <- summary(cs)$coefficients/summary(cs)$standard.errors
p <- (1 - pnorm(abs(z), 0, 1)) * 2
knitr::kable(p)
```

	(Intercept)	Loyalty01-Basic	Loyalty02-Silver	Loyalty03-Gold	Loyalty04-Platinum
direct	0	0.2187728	0.0010984	0.3311982	0.0000219
disp	0	0.0934760	0.0534224	0.0000000	0.8257014
email	0	0.0000416	0.0000002	0.0000006	0.0014671
organic	0	0.3053419	0.1048535	0.3553728	0.3300042
paid	0	0.7711459	0.0000001	0.0024475	0.0458161

```
# calculate odds ratio
knitr::kable(exp(coef(cs)))
```

	(Intercept)	Loyalty01-Basic	Loyalty02-Silver	Loyalty03-Gold	Loyalty04-Platinum
direct	0.1178471	0.7811300	1.6671487	1.2680557	2.2528633
disp	0.0294695	1.7038483	0.4512437	0.0000004	0.9001021
email	0.0114617	5.6578964	8.1197044	9.5289836	4.6316555
organic	0.2733409	1.1404027	0.8159812	1.1774206	0.8417415
paid	0.1882203	0.9558909	0.3767190	0.3969362	0.6347718

Incremental value

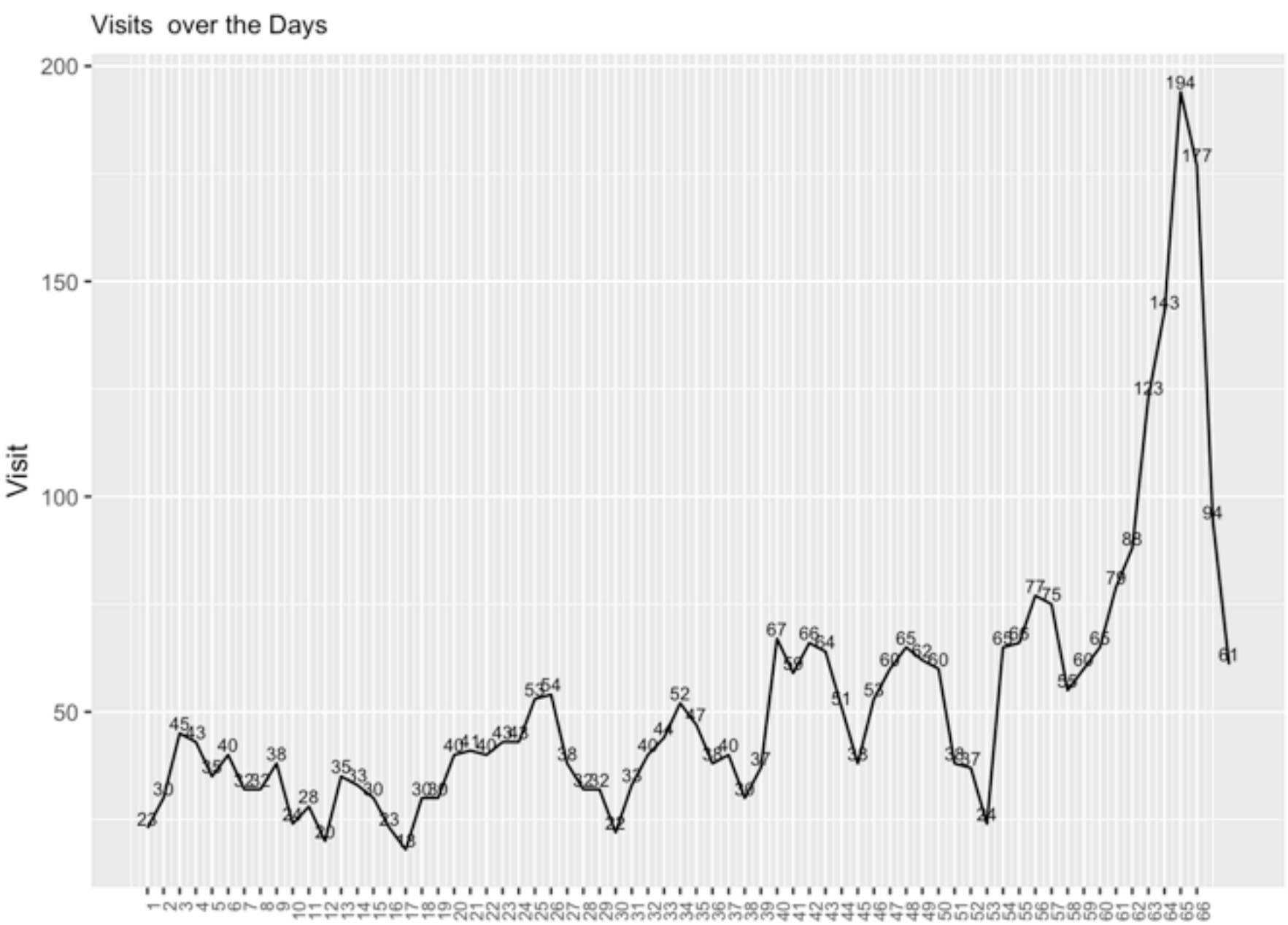
```
# Set probability threshold for classifcation
threshold <- 0.3

# Overall purchases - this is the same as probsmodel5_1
overallprobs <- predict(model5_1, type="prob", newdata=bt57)
overall <- ifelse(overallprobs[, "Purchase"] > threshold, 1, 0)

# Export the above to calculate cost/profit
bt57_prob <- bt57 %>%
        mutate(model5_1_0.3 = overall)
write_excel_csv(bt57_prob, "bt57_prob_0.3.csv")

# Change data to remove a single channel
# Direct
bt57direct <- bt57 %>%
        mutate(Current_direct = 0, Lag1_3days_direct = 0, Lag2_3days_direct = 0, Lag3
_3days_direct = 0)

directprobs <- predict(model5_1, type="prob", newdata=bt57direct)
direct <- ifelse(directprobs[, "Purchase"] > threshold, 1, 0)

# Display
bt57display <- bt57 %>%
        mutate(Current_disp = 0, Lag1_3days_disp = 0, Lag2_3days_disp = 0, Lag3_3days
_disp = 0)

displayprobs <- predict(model5_1, type="prob", newdata=bt57display)
display <- ifelse(displayprobs[, "Purchase"] > threshold, 1, 0)

# Email
bt57email <- bt57 %>%
        mutate(Current_email = 0, Lag1_3days_email = 0, Lag2_3days_email = 0, Lag3_3d
ays_email = 0)
```

```
emailprobs <- predict(model5_1, type="prob", newdata=bt57email)
email <- ifelse(emailprobs[, "Purchase"] > threshold, 1, 0)

# Organic
bt57organic <- bt57 %>%
        mutate(Current_organic = 0, Lag1_3days_organic = 0, Lag2_3days_organic = 0, L
ag3_3days_organic = 0)

organicprobs <- predict(model5_1, type="prob", newdata=bt57organic)
organic <- ifelse(organicprobs[, "Purchase"] > threshold, 1, 0)

# Paid
bt57paid <- bt57 %>%
        mutate(Current_paid = 0, Lag1_3days_paid = 0, Lag2_3days_paid = 0, Lag3_3days
_paid = 0)

paidprobs <- predict(model5_1, type="prob", newdata=bt57paid)
paid <- ifelse(paidprobs[, "Purchase"] > threshold, 1, 0)


# add the predictions to the data
bt57_channeluse <- bt57 %>%
        mutate(OverallPred = overall, DirectPred = direct, DisplayPred = display, Ema
ilPred = email, OrganicPred = organic, PaidPred = paid) %>%
        select(Purchase, OverallPred, DirectPred, DisplayPred, EmailPred, OrganicPred
, PaidPred) %>%
        mutate(Purchase = ifelse(Purchase == "Purchase", 1, 0)) %>%
        summarize(Observed = sum(Purchase), Overall = sum(OverallPred), NoDirect = su
m(DirectPred), NoDisplay = sum(DisplayPred), NoEmail = sum(EmailPred), NoOrganic = su
m(OrganicPred), NoPaid = sum(PaidPred))

knitr::kable(bt57_channeluse)
```

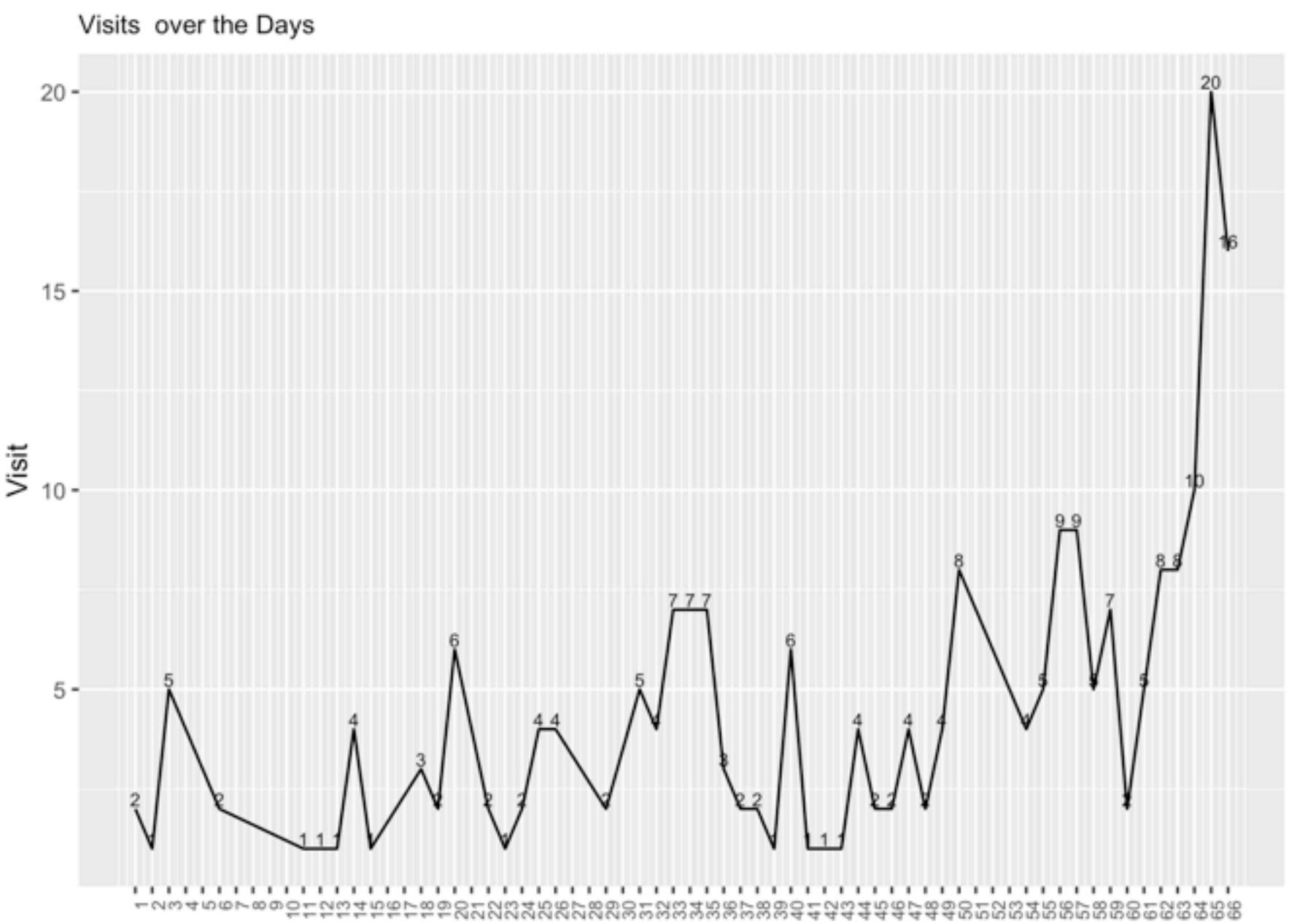
Observed	Overall	NoDirect	NoDisplay	NoEmail	NoOrganic	NoPaid
225	247	122	245	250	154	193

Summary Graphs

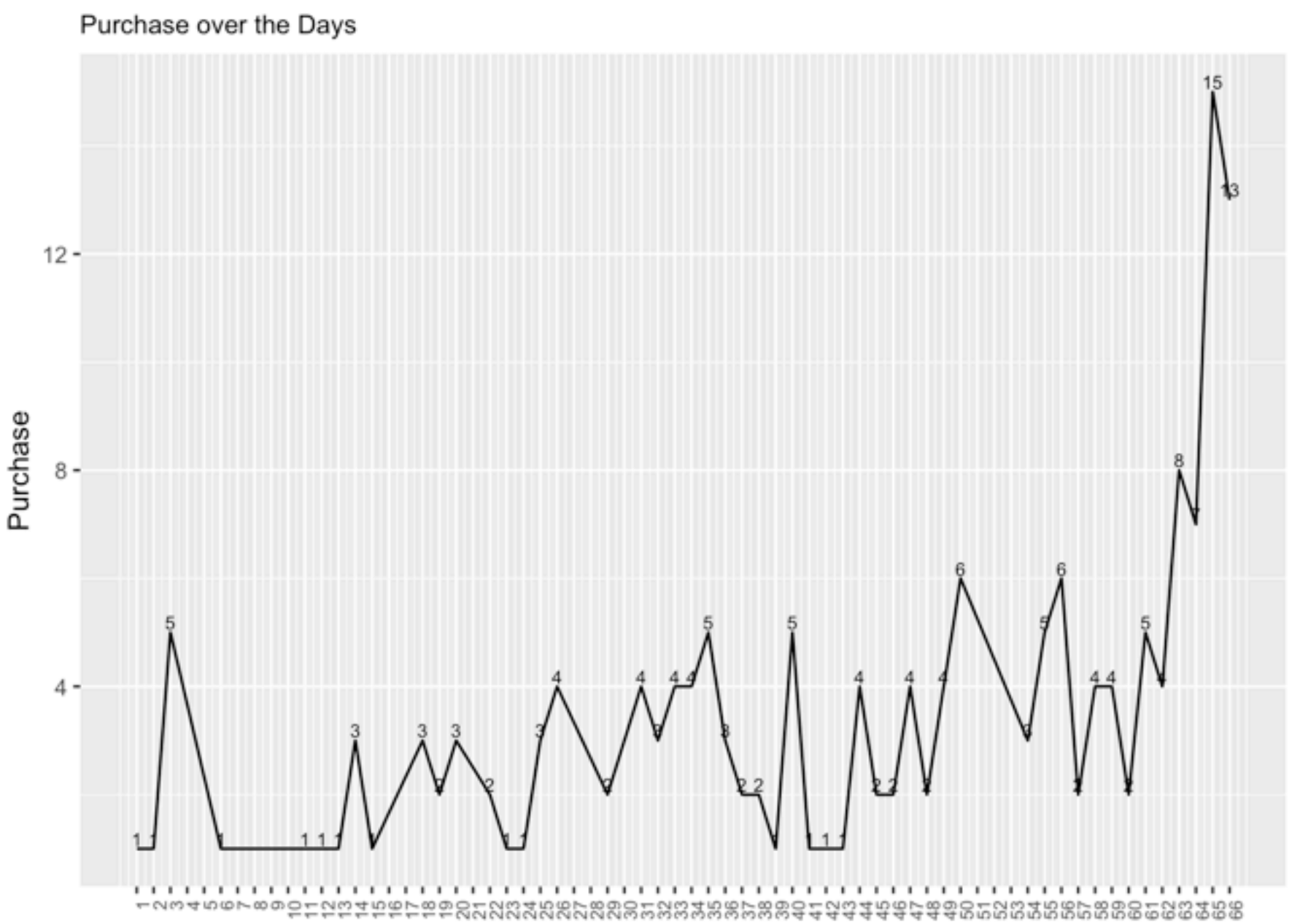
```
# Graph of Visits by Day
SST_bothpurchasestatus <- btd_06_02  %>%
        group_by(Day) %>%
        count(Customer_ID) %>%
        select(Day, Customer_ID, n)

SST_bothpurch <- btd_06_02 %>%
        group_by(Day) %>%
        count(Customer_ID) %>%
        select(Day, n)

SS <- aggregate(SST_bothpurch$n, by=list(DD = SST_bothpurch$Day), FUN = sum)
colnames(SS) <- c("Day", "Visit")

SST <- btd_06_02 %>% filter(Purchase == 1) %>% group_by(Day ) %>%   count(Customer_ID
) %>%  select(Day, n)
SSTT <- aggregate(SST$n, by=list(DD = SST$Day), FUN = sum)
colnames(SSTT) <- c("Day" , "Visit")

SS %>%
        ggplot(aes(x=Day, y=Visit)) +
        geom_line() +
        labs(title = "Visits  over the Days") +
        scale_x_continuous(breaks = round(seq(min(SSTT$Day), max(SSTT$Day), by = 1),
1)) +
        geom_text(aes(label=Visit), size = 2.5, position=position_dodge(width=0.2), v
just=-0.25) +
        theme(plot.title = element_text(size =10),axis.text.x = element_text(size =7,
angle = 90, hjust =1),axis.title.x=element_blank())
```




```
# Graph of Visits by day - but restricted to purchases
SSTT %>% ggplot(aes(x=Day, y= Visit)) + geom_line() +  labs(title = "Visits  over the
Days")  + scale_x_continuous(breaks = round(seq(min(SSTT$Day), max(SSTT$Day), by = 1)
, 1)) + geom_text(aes(label=Visit), size = 2.5, position=position_dodge(width=0.2), v
just=-0.25) + theme(plot.title = element_text(size =10),axis.text.x = element_text(si
ze =7,angle = 90, hjust =1),axis.title.x=element_blank())
```




```
# Graph of Purchases by day
SST_P <- SST %>% mutate(PC = 1) %>% select(-n)
SSTT_01 <- aggregate(SST_P$PC, by=list(DD = SST_P$Day), FUN = sum)
colnames(SSTT_01) <- c("Day" , "Purchase")

SSTT_01  %>% ggplot(aes(x=Day, y= Purchase)) + geom_line() +  labs(title = "Purchase
over the Days")  +scale_x_continuous(breaks = round(seq(min(SSTT_01$Day), max(SSTT_01
$Day), by = 1), 1)) + geom_text(aes(label=Purchase), size = 2.5, position=position_do
dge(width=0.2), vjust=-0.25) + theme(plot.title = element_text(size =10),axis.text.x
= element_text(size =7,angle = 90, hjust =1),axis.title.x=element_blank())
```


Purchase over the Days
Purchase
12
8
4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66