SP Test

Part 2

Alejandro Jiménez Rico

Contents

Step 1

Please calculate the 2nd day, 7th day and 10th day retention (which is the percentage of people that downloaded the game on day 1 who played again on day 2, 7 and 10). Please calculate the retention in two ways, using the user id (user_id) and then the device id (client_mobile_device_aid). (Please note that you only have the device id from the 14th of April).

```
library(data.table)
library(tidyverse)
library(lubridate)
library(viridis)

percent <- function(x, digits = 2, format = "f", ...) {
  paste0(formatC(100 * x, format = format, digits = digits, ...), "%")
}

raw.user <- fread('data/sh_user.csv')
raw.session <- fread('data/sh_session.csv')
raw.battles <- fread('data/sh_battles.csv')
```

```
glimpse(raw.user)
```

```
## Observations: 17,988
## Variables: 54
## $ V1                                <chr> "1", "2", "3", "4", "5...
## $ user_id                           <dbl> 3.042986e+18, 1.234560...
## $ last_session_id                   <dbl> 3.042986e+18, 1.234560...
## $ date_register                     <chr> "2014-11-26", "2014-11...
## $ date_register_android             <chr> "2014-11-26", "2014-11...
## $ date_last_logged                  <chr> NA, "2014-11-26", "201...
## $ platform_last_logged              <chr> NA, "android", "androi...
## $ date_last_logged_android          <chr> NA, "2014-11-26", "201...
## $ register_platform                 <chr> "android", "android", ...
## $ register_ip_ip                    <chr> "62.97.102.90", "84.12...
## $ register_ip_country               <chr> "ES", "ES", "ES", "ES"...
## $ register_ip_lat                   <dbl> 41.38880, 40.41650, 41...
## $ register_ip_lon                   <dbl> 2.15899, -3.70256, 2.1...
## $ register_ip_timezone              <chr> "+02:00", "+02:00", "+...
## $ game_current_level                <int> NA, 1, 1, 1, 1, 1, 1, ...
## $ game_current_cash                 <int> NA, 5, 5, 5, 5, 5, 5, ...
## $ game_current_gold                 <int> NA, 3000, 3000, 3000, ...
## $ game_current_food                 <int> NA, 140, 140, 140, 140...
## $ game_current_xp                   <S3: integer64> NA, 0, 0, 0,...
## $ game_current_stamina              <int> NA, 4, 4, 4, 4, 4, 4, ...
```

```
## $ revenues_dollars_net                    <dbl> NA, NA, NA, NA, NA, NA...
## $ revenues_dollars_gross                  <dbl> NA, NA, NA, NA, NA, NA...
## $ revenues_num_transactions               <int> NA, NA, NA, NA, NA, NA...
## $ revenues_date_first_transaction         <chr> NA, NA, NA, NA, NA, NA...
## $ revenues_date_first_transaction_android <chr> NA, NA, NA, NA, NA, NA...
## $ revenues_1d                             <dbl> NA, NA, NA, NA, NA, NA...
## $ revenues_2d                             <dbl> NA, NA, NA, NA, NA, NA...
## $ revenues_3d                             <dbl> NA, NA, NA, NA, NA, NA...
## $ revenues_1w                             <dbl> NA, NA, NA, NA, NA, NA...
## $ revenues_2w                             <dbl> NA, NA, NA, NA, NA, NA...
## $ revenues_4w                             <dbl> NA, NA, NA, NA, NA, NA...
## $ revenues_8w                             <dbl> NA, NA, NA, NA, NA, NA...
## $ revenues_12w                            <dbl> NA, NA, NA, NA, NA, NA...
## $ revenues_16w                            <dbl> NA, NA, NA, NA, NA, NA...
## $ funnel_last_update                      <chr> NA, NA, "2014-11-27", ...
## $ funnel_current                          <chr> NA, NA, "201", NA, "20...
## $ funnel_5min                             <chr> NA, NA, "201", NA, "20...
## $ funnel_1hour                            <chr> NA, NA, "201", NA, "20...
## $ funnel_1d                               <chr> NA, NA, "201", NA, "20...
## $ user_category                           <chr> "player", "player", "p...
## $ register_ip_region                      <chr> "CATALONIA", "MADRID",...
## $ register_ip_city                        <chr> "BARCELONA", "MADRID",...
## $ revenues_date_first_transaction_woe     <chr> NA, NA, NA, NA, NA, NA...
## $ revenues_incremental_fields_last_update <chr> NA, NA, NA, NA, NA, NA...
## $ register_os                             <chr> NA, NA, NA, NA, NA, NA...
## $ register_device                         <chr> NA, NA, NA, NA, NA, NA...
## $ register_version                        <chr> "0", "0", "0", "0", "0...
## $ register_mobile_uid                     <chr> NA, NA, NA, NA, NA, NA...
## $ register_mobile_device_aid              <lgl> NA, NA, NA, NA, NA, NA...
## $ client_mobile_device_aid                <chr> NA, NA, NA, NA, NA, NA...
## $ last_global_device_id                   <chr> NA, NA, NA, NA, NA, NA...
## $ register_device_android                 <chr> NA, NA, NA, NA, NA, NA...
## $ register_source                         <chr> NA, NA, NA, NA, NA, NA...
## $ register_source_android                 <chr> NA, NA, NA, NA, NA, NA...
```

```
glimpse(raw.session)
```

```
## Observations: 148,724
## Variables: 40
## $ V1                          <chr> "1", "2", "3", "4", "5", "6", "7"...
## $ user_id                     <dbl> 3.043009e+18, 3.043009e+18, 3.043...
## $ session_id                  <dbl> 3.043009e+18, 3.043009e+18, 3.043...
## $ datetime                    <chr> "2014-11-26", "2014-11-26", "2014...
## $ platform                    <chr> "android", "android", "android", ...
## $ version                     <chr> "0", "0", "0", "0", "0", "0", "0"...
## $ sample_ratio                <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ ip_ip                       <chr> "62.97.102.90", "62.97.102.90", "...
## $ ip_country                  <chr> "ES", "ES", "ES", "ES", "ES", "ES...
## $ ip_region                   <chr> "CATALONIA", "CATALONIA", "CATALO...
## $ ip_city                     <chr> "BARCELONA", "BARCELONA", "BARCEL...
## $ ip_lat                      <dbl> 41.3888, 41.3888, 41.3888, 41.388...
## $ ip_lon                      <dbl> 2.15899, 2.15899, 2.15899, 2.1589...
## $ ip_timezone                 <chr> "+02:00", "+02:00", "+02:00", "+0...
## $ client_mobile_os            <chr> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ client_mobile_device        <chr> NA, NA, NA, NA, NA, NA, NA, NA, N...
```

```
## $ client_mobile_language       <chr> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ client_mobile_uid            <chr> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ client_mobile_device_aid     <chr> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_basic_level             <int> 1, 2, 3, 7, 7, 1, 1, 1, 1, 1, 1, ...
## $ game_basic_cash              <int> 5, 10005, 9926, 9371, 9371, 5, 5,...
## $ game_resources_gold          <dbl> 3000, 1002750, 998400, 605526, 60...
## $ game_resources_food          <int> 140, 1000010, 1000070, 1000120, 1...
## $ game_resources_xp            <S3: integer64> 0, 275, 1075, 8675, 867...
## $ game_resources_stamina       <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, ...
## $ game_resources_de            <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_ev            <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_fe            <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_friend_points <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_le            <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_ne            <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_rde           <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_re            <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_rfe           <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_rle           <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_rne           <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_rwe           <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_ssre          <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_sre           <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ game_resources_we            <int> NA, NA, NA, NA, NA, NA, NA, NA, N...
```

```
glimpse(raw.battles)
```

```
## Observations: 397,339
## Variables: 12
## $ V1                           <chr> "1", "2", "3", "4", "5", "6", "7"...
## $ datetime                     <chr> "2015-04-07", "2015-04-19", "2015...
## $ game_basic_cash              <int> 12, 12, 12, 13, 13, 13, 12, 29, 1...
## $ game_basic_level             <int> 1, 1, 1, 2, 2, 2, 1, 3, 1, 2, 3, ...
## $ platform                     <chr> "android", "android", "android", ...
## $ sample_ratio                 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ session_id                   <int> 61949138, 2192785, 89103400, 8910...
## $ single_player_status_node_id <int> 1, 1, 1, 2, 1, 2, 1, 6, 1, 3, 5, ...
## $ single_player_status_tries   <lgl> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ single_player_status_type    <chr> "pve", "pve", "pve", "pve", "pve"...
## $ user_id                      <dbl> 2.000000e+00, 1.940917e+14, 6.785...
## $ version                      <chr> "9999999999", "1504131647", "1504...
```

Let's get the date of registration of every user

```
df.registration <- raw.user[,c("user_id", "date_register")]
```

We should check for duplicates

```
length(df.registration$user_id) - length(unique(df.registration$user_id))
```

```
## [1] 63
```

```
df <- df.registration %>%
    group_by(user_id) %>%
    summarise(N=n())

unique(df$N)
```

```
## [1] 1 2
```

There are 63 people who has been registered twice.

```
users.twice <- df[df[,2] == 2,1]
double.registration <- df.registration[df.registration$user_id %in% as.array(users.twice$user_id)]

double.registration <- double.registration %>%
    group_by(user_id) %>%
    summarise(unique = n_distinct(date_register))

unique(double.registration$unique)
```

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

But those who registered twice, did it both times on the same day. So we can discard duplicated registration records and move on.

```
reg <- df.registration[!duplicated(df.registration[,'user_id']),]
rm(df,df.registration,double.registration,users.twice)
```

```
con <- raw.session[,c("user_id", "datetime")]

df <- merge(reg,con)
df <- df %>%
    mutate(date_register = ymd(date_register)) %>%
    mutate(datetime = ymd(datetime))

df$ret <- df$datetime - df$date_register +1
df <- df %>%
    mutate(ret = as.numeric(ret))
df <- df[,c(1,4)]
df <- df[!duplicated(df[,c('user_id','ret')]),]
```

Now we just plot the results

```
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked _by_ '.GlobalEnv':
##
##     percent

## The following object is masked from 'package:viridis':
##
##     viridis_pal

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
ret <- as.data.frame(table(df$ret))
ret$Freq <- as.numeric(ret$Freq/sum(ret$Freq))
ret$Var1 <- as.numeric(ret$Var1)
```

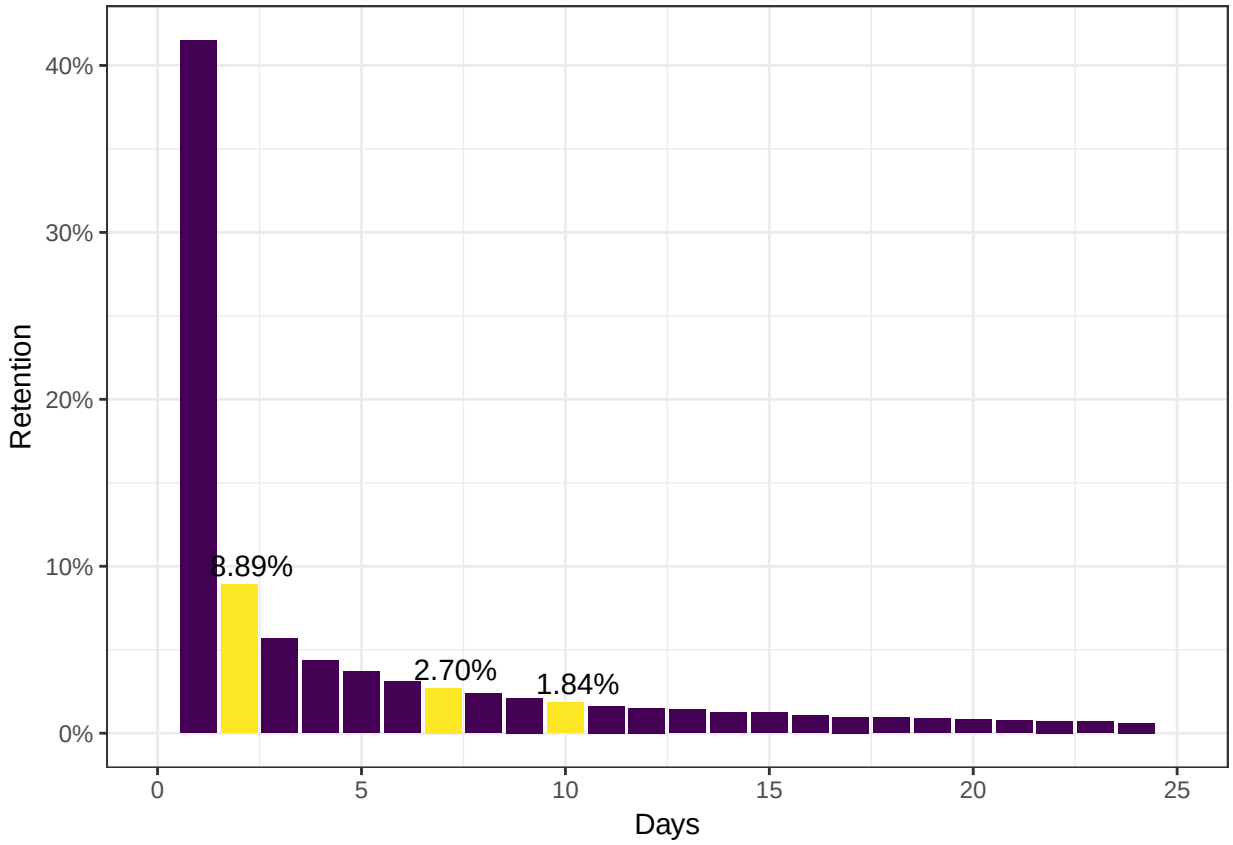
```r
ret$colour <- ifelse(ret$Var1 == 2 | ret$Var1 == 7 | ret$Var1 == 10, "yellow", "blue")

gg <- ggplot(ret,  aes(x=Var1, y=Freq, fill=colour)) +
    geom_bar(stat="identity") +
    scale_y_continuous(labels = scales::percent_format()) +
    scale_x_continuous(limits=c(0,25)) +
    scale_fill_viridis(discrete = TRUE) +
    geom_text(aes(label=ifelse(Var1 == 2 | Var1 == 7 | Var1 == 10, percent(round(Freq, digits= 4)), ""))
    theme_bw() +
    labs(x = "Days", y = "Retention") +
    theme(legend.position = "NONE")

gg
```

```
## Warning: Removed 61 rows containing missing values (position_stack).
```

```
## Warning: Removed 61 rows containing missing values (geom_text).
```



Step 2

Please explain if the results are the same using the two different ids and if not why not.

I am not entirely confident if I correctly unerstood the question. But at this point we should notice that it makes no sense whatsoever to group the users using their `device_id` because we have data only from one day. This implies that we can not know anything about those users at any other day.

Step 3

Please explain how much of the variation in the 2nd day retention can be explained by whether

or not the user completed the tutorial on the 1st day and whether it is a significant variable in understanding 2nd day retention. (You know if someone has complete the tutorial from the funnel steps in the user file. We are interested in the variable called funnel_1d and anyone with a funnel step higher than 2116 has completed the tutorial). You have to use a logistic regression.

```
df.tutorial <- raw.user[,c("user_id", "funnel_1d")]
df.tutorial$tut <- ifelse(df.tutorial$funnel_1d < 2116, 1, 0)
df.tutorial$tut[is.na(df.tutorial$tut)] <- 0


pass.tutorial <- df.tutorial[,c(1,3)]
pass.tutorial$user_id <- as.factor(pass.tutorial$user_id)

ret2f <- df %>%
    mutate(ret2 = ifelse(ret == 2, 1, 0)) %>%
    group_by(user_id,ret2 )

ret2f$user_id <- as.factor(ret2f$user_id)

ret2 <- ret2f[,c(1,3)] %>%
    group_by(user_id) %>%
    summarise(ret2=sum(ret2))

tut.ret <- merge(pass.tutorial, ret2)
tut.ret$tut <- as.factor(tut.ret$tut)
tut.ret$ret2 <- as.factor(tut.ret$ret2)
```

Now we have the data ready for testing and doing regressions.

The first question to be answered now is whether or not these two variables we are considering are correlated. Our variables are cathegorical. This means that they do not express a magnitude by a factor that labels a category. Two cathegorical variables can be correlated in the same sense as two numerical variables; but the statistical test performed for each case is quite different. To measure the dependence between cathegorical variables there is a widely used test, named *Chi-Squared Test*. Using this test, we can decide whether or not two cathegorical variables are correlated.

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
tb1 <- table(tut.ret$tut, tut.ret$ret2)
chisq.test(tb1)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tb1
## X-squared = 1265.9, df = 1, p-value < 2.2e-16
```

The result of this test is plainly clear. This result states that these variables are not independent. Thus, we conclude that there is a relationship between having finished the tutorial and the 2nd day retention.

Now, we can check further the relationship between these variables using a *Logistic Regression*. The point of using this regression is that is common to predict categorical outcomes using categorical variables. We can perform a Logistic Regression trying to predict whether or not some player have been retented the 2nd day since its regristration and determine if the `tutorial` variable (which is categorical) is useful for this prediction.

```
model <- glm(tut.ret$ret2 ~ tut.ret$tut, family = binomial)
summary(model)
```

```
##
## Call:
## glm(formula = tut.ret$ret2 ~ tut.ret$tut, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8316  -0.8316  -0.3533  -0.3533   2.3685
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.88413    0.02007  -44.05   <2e-16 ***
## tut.ret$tut1 -1.85841    0.05789  -32.10   <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: 18715  on 17962  degrees of freedom
## Residual deviance: 17223  on 17961  degrees of freedom
## AIC: 17227
##
## Number of Fisher Scoring iterations: 5
```

Considering the pvalue obtained, the variable that indicates whether or not a player was completed the tutorial is relevant to determine the 2nd day retention of that player.

Now that we know that the variable *is* relevant, we could ask *how much* relevant it is. Or, in other words, how much variation of the prediction on the 2nd day retention is caused by this variable. In Logistic Regression, technically we can not talk about any measure of *How much variation can be explained by this variable*. If we were talking about linear regression we could simply compute the R^2 and interpret its result as the percentage of the variation that could be explained with the variable; but there is no such thing in Logistic Regression. Notwithstanding, there are some so-called pseudo-R^2 as `Cos & Snell` and `Nagelkerke`.

Now we are going to apply the latter, since it is an adjustment of the former.

```
library(rms)
```

```
## Loading required package: Hmisc
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:dplyr':
##
```

```
##     combine, src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units

## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##     backsolve
```

```
x <- as.numeric(tut.ret$tut)
y <- as.numeric(tut.ret$ret2)
model2 <- lrm(y ~ x)
print(model2)
```

```
## Logistic Regression Model
##
##  lrm(formula = y ~ x)
##
##                        Model Likelihood     Discrimination    Rank Discrim.
##                           Ratio Test           Indexes           Indexes
##  Obs          17963    LR chi2    1492.62    R2       0.123    C       0.652
##   1           14095    d.f.             1    g        0.825    Dxy     0.304
##   2            3868    Pr(> chi2) <0.0001    gr       2.281    gamma   0.730
##  max |deriv| 2e-08                          gp       0.103    tau-a   0.103
##                                             Brier    0.157
##
##            Coef    S.E.   Wald Z Pr(>|Z|)
##  Intercept  0.9743 0.0675  14.43 <0.0001
##  x         -1.8584 0.0579 -32.10 <0.0001
##
```

The output of this model leads to a Nagelkerke's $R^2 = 0.123$ which seems pretty low. Could we say that a 12.30% of the variance of the 2nd day retention can be explain from the variable that states whether or not a user has finished his tutorial? Technically, we should not (see [1]), but it is the closest interpretation we can conclude.

[1]: Applied Logistic Regression