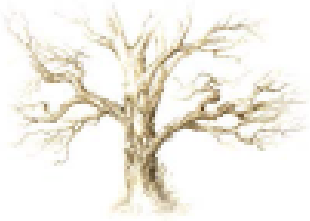


Tatán Rufino

1/11/2019

VIDEOGAMES - Test analítica

Bootstrapping

En primer lugar vamos a cargar los paquetes necesarios

```
library(readr)        #Para leer e importar datos
library(dplyr)        #Análisis exploratorio
library(ggplot2)      #Visualizaciones
library(tidyverse)    #Utilizada para ordenar dataset
library(gsheet)       #Leer google sheets
library(readxl)
library(knitr)        #Varios propósitos
library(DT)           #libreria DataTables
library(caret)        #Variables para Correlation Matrix
library(grid)         #Plotear
library(gridExtra)    #Plotear
library(corrr)        #Var estadisticas
library(ggpubr)
library(qcc)
library(data.table)
library(janitor)
library(arules)
library(arulesViz)
library(lubridate)
library(stats)
library(samplingbook)
library(rmarkdown)
```

Ejercicio 1:

Apertura de archivos

```
setwd("D:/MAIN/CODING/R/WORKS/GENERA/INDIE")

DI <- read_xlsx("Task1-Test_Indie_Analytics-Sessions.xlsx")
kable(head(DI,20))
```

```
list.files()
```

```
##  [1] "Enunciado.pdf"
##  [2] "INDIE - copia.R"
##  [3] "INDIE.html"
##  [4] "INDIE.R"
##  [5] "INDIE.Rmd"
##  [6] "INDIE.tex"
##  [7] "Task1-Test_Indie_Analytics-Sessions.csv"
##  [8] "Task1-Test_Indie_Analytics-Sessions.xlsx"
##  [9] "Task2-Test_Indie_Analytics-IAPs.xlsx"
## [10] "Task3-Test_Indie_Analytics-Missions.xlsx"
```

Análisis exploratorio

ID User	Install Date	Country	ID Session	Session Date	Revenue ($)	Game Time (ms)	Completed Levels
36b29ebb42cf	2019-09-06	DE	b8d2b9ea31f9	2019-09-06	0	1507375	14
3f6273cffacb	2019-09-06	GB	16421fe529fd	2019-09-06	0	173352	0
3f6273cffacb	2019-09-06	GB	4f91b02ce4ce	2019-09-06	0	961532	2
3f6273cffacb	2019-09-06	GB	bb608c22b34f	2019-09-06	0	427253	2
3f6273cffacb	2019-09-06	GB	ad0b3e5680a1	2019-09-06	0	208747	0
3f6273cffacb	2019-09-06	GB	e535fe7774eb	2019-09-06	0	2526796	13
3f6273cffacb	2019-09-06	GB	5cd53317b338	2019-09-07	0	1167251	4
3f6273cffacb	2019-09-06	GB	40ae2e7a1128	2019-09-07	0	876164	2
3f6273cffacb	2019-09-06	GB	a1e012406656	2019-09-09	0	503751	1
3f6273cffacb	2019-09-06	GB	b801771f5349	2019-09-10	0	14286	0
3f6273cffacb	2019-09-06	GB	e331b5f38bbf	2019-09-10	0	269087	0
3f6273cffacb	2019-09-06	GB	eebae0c29580	2019-09-12	0	193870	0

3f6273cffacb	2019-09-06	GB	e2369e0ad4ce	2019-09-13	0	175664	0
698c3c7c654a	2019-09-08	FR	b48d6fb2e20d	2019-09-08	0	2102961	16
e6490c545ac5	2019-09-08	FR	b34104b2f999	2019-09-08	0	1721584	4
e6490c545ac5	2019-09-08	FR	f743a184bf8c	2019-09-09	0	867930	1
3212a4b2d307	2019-09-07	GB	99eb71a43d42	2019-09-07	0	860864	6
3212a4b2d307	2019-09-07	GB	1c97ba53afe1	2019-09-07	0	1181740	2
3212a4b2d307	2019-09-07	GB	917fb5b86a88	2019-09-07	0	452549	3
3212a4b2d307	2019-09-07	GB	eecb68494c89	2019-09-07	0	1789253	6
Cambiamos los n	ombres de las v	ariables p	ara hacer más f	acil su referen	ciación		

```
names(DI)[c(1,2,3,4,5,6,7,8)]=c("user_id","ins_date","country","ses_id","ses_date","revenue","game_time","comp_lvl")

summary(DI)
```

```
##    user_id             ins_date                      country
##  Length:12532       Min.   :2019-09-05 00:00:00   Length:12532
##  Class :character   1st Qu.:2019-09-05 00:00:00   Class :character
##  Mode  :character   Median :2019-09-06 00:00:00   Mode  :character
##                     Mean   :2019-09-06 10:58:45
##                     3rd Qu.:2019-09-08 00:00:00
##                     Max.   :2019-09-08 00:00:00
##     ses_id             ses_date                      revenue
##  Length:12532       Min.   :2019-09-05 00:00:00   Min.   : 0.00000
##  Class :character   1st Qu.:2019-09-07 00:00:00   1st Qu.: 0.00000
##  Mode  :character   Median :2019-09-08 00:00:00   Median : 0.00000
##                     Mean   :2019-09-08 21:44:52   Mean   : 0.02794
##                     3rd Qu.:2019-09-11 00:00:00   3rd Qu.: 0.00000
##                     Max.   :2019-09-15 00:00:00   Max.   :36.70000
##    game_time          comp_lvl
##  Min.   :  10007   Min.   : 0.000
##  1st Qu.: 297122   1st Qu.: 0.000
##  Median : 626269   Median : 2.000
##  Mean   : 809750   Mean   : 3.467
##  3rd Qu.:1072430   3rd Qu.: 4.000
##  Max.   :4022696   Max.   :31.000
```

```
glimpse(DI)
```

```
## Observations: 12,532
## Variables: 8
## $ user_id   <chr> "36b29ebb42cf", "3f6273cffacb", "3f6273cffacb", "3f6...
## $ ins_date  <dttm> 2019-09-06, 2019-09-06, 2019-09-06, 2019-09-06, 201...
## $ country   <chr> "DE", "GB", "GB", "GB", "GB", "GB", "GB", "GB", "GB"...
## $ ses_id    <chr> "b8d2b9ea31f9", "16421fe529fd", "4f91b02ce4ce", "bb6...
## $ ses_date  <dttm> 2019-09-06, 2019-09-06, 2019-09-06, 2019-09-06, 201...
## $ revenue   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
```

```
## $ game_time <dbl> 1507375, 173352, 961532, 427253, 208747, 2526796, 11...
## $ comp_lvl  <dbl> 14, 0, 2, 2, 0, 13, 4, 2, 1, 0, 0, 0, 0, 16, 4, 1, 6...
```

Observamos variables que no tienen la categoría correcta y que nos pueden dar problemas en el análisis

```
DI$ins_date = as.Date(DI$ins_date)
DI$ses_date = as.Date(DI$ses_date)
```

Tenemos una primera visual, y posteriormente procedemos al análisis: Observamos el rango de fechas y el total de paises:

```
min(DI$ses_date)
```

```
## [1] "2019-09-05"
```

```
max(DI$ses_date)
```

```
## [1] "2019-09-15"
```

ses_ini_date = “2019-09-05” ses_end_date = “2019-09-15”

Un total de 11 días de muestreo jugado

```
DI %>% group_by(country) %>% summarise(n=n())
```

country	n
<chr>	<int>
DE	5146
FR	4601
GB	2785

3 rows

Observamos que hay usuarios de 3 países, DE, FR, GB

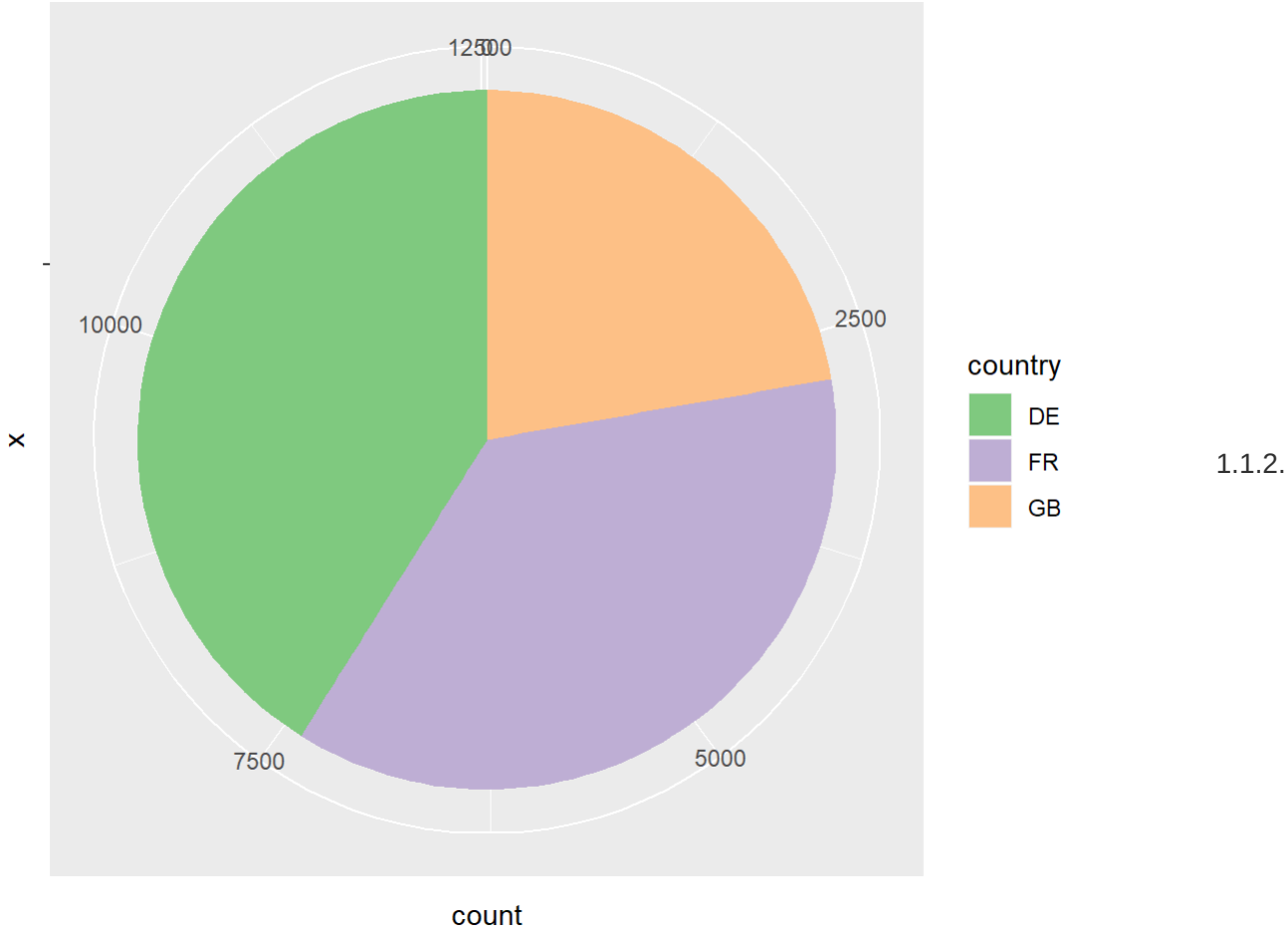
Ejercicio 1.1:

1.1.1. Media Sesiones:

```
E1_ses_avg = DI %>% group_by(ses_date) %>%
  mutate(DE=ifelse(country=="DE",1,0),
         FR=ifelse(country=="FR",1,0),
         GB=ifelse(country=="GB",1,0)) %>%
  summarise(DE=sum(DE),
            FR=sum(FR),
            GB=sum(GB)) %>%
  summarise(DE=mean(DE),
            FR=mean(FR),
            GB=mean(GB)) %>%
  t() %>% as.data.table()
```

```
names(E1_ses_avg)[1] = "ses_avg"
```

```
DI %>% mutate(DAY=day(ses_date)-5) %>%
  ggplot(aes(x = "",fill=country)) +
  geom_bar(width = 1)+
  coord_polar(theta = "y")+
  scale_fill_brewer(palette = "Accent")
```



1.1.2.

Tiempo medio jugado:

```
E1_time_avg = DI %>% group_by(country) %>%
  summarise(time_avg=mean(game_time/(1000*60*60))) ##calculo que lo pasamos a horas
```

1.1.3. Niveles completados:

```
E1_complvl_total = DI %>% group_by(country) %>%
  summarise(complvl_tot = sum(comp_lvl))
```

1.1.4. Niveles completados de media por sesión:

```
E1_complvl_avg = DI %>% group_by(ses_date) %>%
  mutate(DE=ifelse(country=="DE",as.integer(comp_lvl),0),
         FR=ifelse(country=="FR",as.integer(comp_lvl),0),
         GB=ifelse(country=="GB",as.integer(comp_lvl),0)) %>%
  summarise(DE=sum(DE),
            FR=sum(FR),
            GB=sum(GB)) %>%
  summarise(DE=mean(DE),
            FR=mean(FR),
```

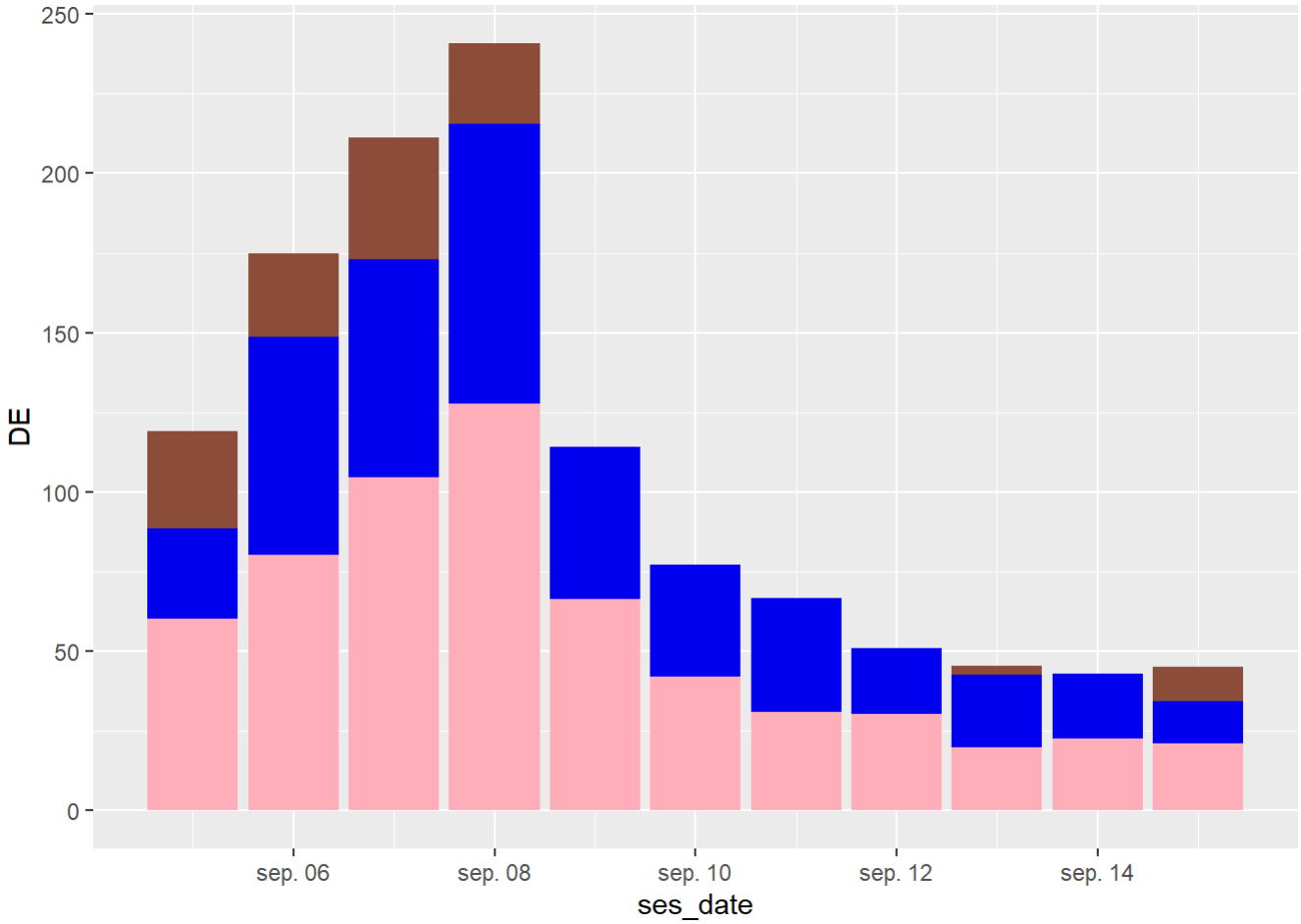
```
            GB=mean(GB)) %>%
  t() %>% as.data.table()
names(E1_complvl_avg)[1] = "complvl_avg"
```

1.1.5. Tiempo jugado por sesión (medio):

```
E1_time_ses_avg = DI %>% group_by(ses_date) %>%
  mutate(DE=ifelse(country=="DE",game_time/(1000*60*60),0),
         FR=ifelse(country=="FR",game_time/(1000*60*60),0),
         GB=ifelse(country=="GB",game_time/(1000*60*60),0)) %>%
  summarise(DE=sum(DE),
            FR=sum(FR),
            GB=sum(GB)) %>%
  summarise(DE=mean(DE),
            FR=mean(FR),
            GB=mean(GB)) %>%
  t() %>% as.data.table()
names(E1_time_ses_avg)[1] = "time_ses_avg"
```

Es interesante, de cualquier modo, observar un histograma con el tiempo jugado por sesión y por país:

```
  DI %>% group_by(ses_date) %>%
  mutate(DE=ifelse(country=="DE",game_time/(1000*60*60),0),
         FR=ifelse(country=="FR",game_time/(1000*60*60),0),
         GB=ifelse(country=="GB",game_time/(1000*60*60),0)) %>%
  summarise(DE=sum(DE),
            FR=sum(FR),
            GB=sum(GB)) %>%
    ggplot(aes(x=ses_date),position="dodge")+
    geom_col(aes(y=DE),fill="salmon4")+
    geom_col(aes(y=FR),fill="blue2")+
    geom_col(aes(y=GB),fill="lightpink1")
```



Vemos que generalmente en Alemania existe una mayor frecuencia de tiempo en juego, seguido de Francia y bastante por debajo Gran Bretaña.

Obtenemos finalmente la tabla total con todas las variables obtenidas:

```
  E1.1 = cbind(E1_time_avg[,1],E1_ses_avg,E1_time_avg[,2],E1_time_ses_avg,E1_complvl_total[
,2],E1_complvl_avg)
```

```
    `%DE-GB` = ((18057-9194)/9194)*100
```

La “ganadora” en general en relación al consumo de tiempo, niveles completados… etc, es sin duda Alemania, que en estadísticas como el incremento en niveles completados respecto a GB alcanza casi el 100%, sin embargo aún no se ha analizado la cantidad de dinero gastado por usuario

Tomamos nota de algunos totales:

```
Tot_usr = as.integer(DI %>% distinct(user_id, .keep_all = TRUE)
                         %>% summarise(n=n()))
```

```
Tot_Pay = as.integer(DI %>% distinct(user_id, .keep_all = TRUE) %>%
                       filter(revenue>0) %>%
                       summarise(n=n()))
```

```
Tot_Rev = as.integer(DI %>%  summarise (Tot_Rev=sum(revenue)))
```

```
`%Tot_pay-Tot_usr`=(Tot_Pay/Tot_usr)*100
```

1.1.6. %pagadores:

```
E1_Pay_share = DI %>% group_by(country) %>%
  mutate(Payers=ifelse(revenue>0,1,0)) %>%
  summarise(Tot_gam=n(),`%Paying_share`=(sum(Payers)/n())*100,Tot_Rev=sum(revenue),Max_rev=
max(revenue))
```

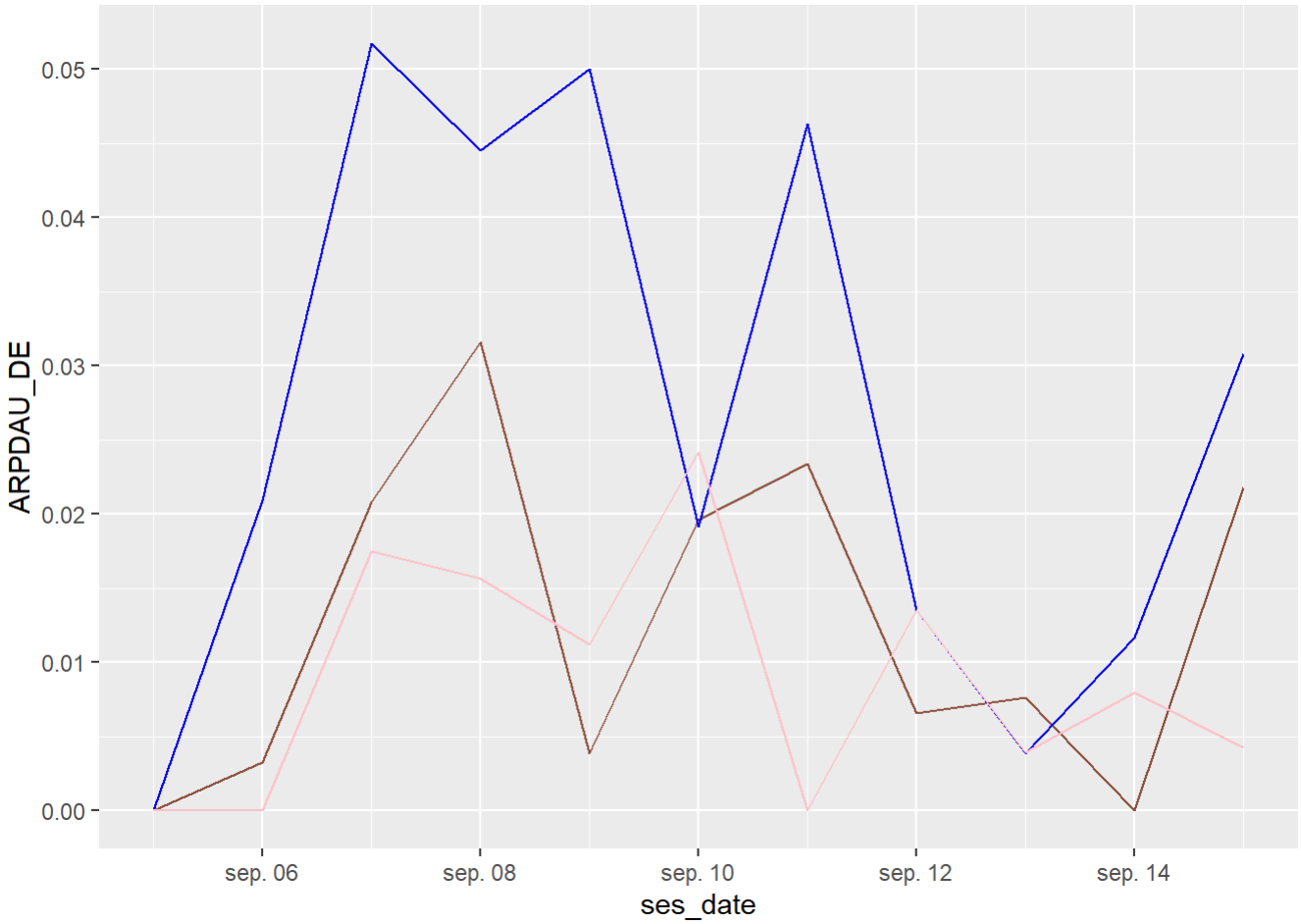
Lo primero a comentar es que el porcentaje de pagadores *por usuario* es bajo en general (0,32%) pero, *es eso una cifra baja realmente?*; como todo es relativo, obviamente no podemos responder a esa pregunta. En cambio cuando dividimos por paises decubrimos algo curioso y es que Alemania, que era la ganadora en cantidad de usuarios, partidas completadas y juego en general es la que posee el % de usuarios (los que hemos llamado) cuando observamos el porcentaje de pagadores más bajo, si bien no el que acumula menos revenues, ya que esa estadística se la lleva Gran bretaña. Francia es aquí quien más ingresos produce por compras in-app, no estando, en cambio, muy alejado de alemania en frecuencia de juego

1.1.7. ARPDAU (Average Revenue Per Daily Active User)

La fórmula se calcula de la siguiente manera: daily income / daily active users, de manera que:

```
E1_ARPDAU = DI %>% group_by(ses_date) %>%
  mutate(rev_DE=ifelse(country=="DE",revenue,0),
         rev_FR=ifelse(country=="FR",revenue,0),
         rev_GB=ifelse(country=="GB",revenue,0)) %>%
  summarise(ARPDAU_DE=sum(rev_DE)/n_distinct(user_id),
            ARPDAU_FR=sum(rev_FR)/n_distinct(user_id),
            ARPDAU_GB=sum(rev_GB)/n_distinct(user_id))
```

```
E1_ARPDAU %>% ggplot(aes(x=ses_date))+
    geom_line(aes(y=ARPDAU_DE), colour="salmon4")+
    geom_line(aes(y=ARPDAU_FR), colour="blue2")+
    geom_line(aes(y=ARPDAU_GB), colour="pink")
```



Y aquí, la ganadora en más de un 100% de la que se encuentra por detrás suya es Francia (de Alemania)

1.1.8. ARPU (Average Revenue Per User). Es como la ARPDAU, pero aplicada a un período

La fórmula se calcula por períodos, en nuestro caso en los 11 días comprendidos en el análisis: total revenue / average suscribers, y además asumimos que la media de suscritos en ese período es el n1 total de jugadores de manera que:

```
E1_ARPU = DI %>% group_by(country) %>%
  summarise(ARPU=sum(revenue)/n())
```

No debe confundirse con la media de la ARPDAU (ya que estaríamos incurriendo en un error de proporciones):

```
   E1_ARPDAU_avg = E1_ARPDAU %>%
  summarise(ARPDAU_DE_avg=mean(ARPDAU_DE),
            ARPDAU_FR_avg=mean(ARPDAU_FR),
            ARPDAU_GB_avg=mean(ARPDAU_GB))
```

1.1.8. ARPPU (Average Revenue Per Paying user). Es la ARPU cambiando el denominador por los usuarios de pago,

```
   E1_ARPPU = DI %>% group_by(country) %>%
     mutate(Payers=ifelse(revenue>0,1,0)) %>%
     summarise(ARPPU=sum(revenue)/sum(Payers))
```

```
   E1.2 = cbind(E1_ARPU,ARPPU=E1_ARPPU$ARPPU,Pay_Share=E1_Pay_share$`%Paying_share`)
```

Nótese que ARPU es equivalente a ARPPU * Paying Share, y efectivamente este hecho puede comprobarse multiplicando el Pay_share por el ARPPU, habiendo sido calculado por separado:

```
E1.2 %>% mutate(ARPU_indirect = ARPPU*Pay_Share/100)
```

country <chr>	ARPU <dbl>	ARPPU <dbl>	Pay_Share <dbl>	ARPU_indirect <dbl>
DE	0.01762923	3.2400	0.5441119	0.01762923
FR	0.04255379	5.5940	0.7607042	0.04255379
GB	0.02285458	3.1825	0.7181329	0.02285458

3 rows

Así pues, la tabla completa de KPIs según país sería la siguiente:

```
E1 = cbind(E1.1,E1.2[,-1])
```

Ejercicio 1.2.:

```
class(E1_ARPDAU)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
summary(as.data.table(E1_ARPDAU))
```

```
##     ses_date            ARPDAU_DE          ARPDAU_FR
##  Min.   :2019-09-05   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:2019-09-07   1st Qu.:0.003557   1st Qu.:0.01260
##  Median :2019-09-10   Median :0.007668   Median :0.02095
##  Mean   :2019-09-10   Mean   :0.012604   Mean   :0.02660
##  3rd Qu.:2019-09-12   3rd Qu.:0.021337   3rd Qu.:0.04542
##  Max.   :2019-09-15   Max.   :0.031592   Max.   :0.05172
##    ARPDAU_GB
##  Min.   :0.000000
##  1st Qu.:0.001965
##  Median :0.007974
##  Mean   :0.008933
##  3rd Qu.:0.014558
##  Max.   :0.024221
```

```
E2 = DI %>% group_by(ses_date) %>%
  mutate(DE=ifelse(country=="DE",1,0),
         FR=ifelse(country=="FR",1,0),
         GB=ifelse(country=="GB",1,0)) %>%
  summarise(DAU_DE=sum(DE),
            DAU_FR=sum(FR),
            DAU_GB=sum(GB))
```

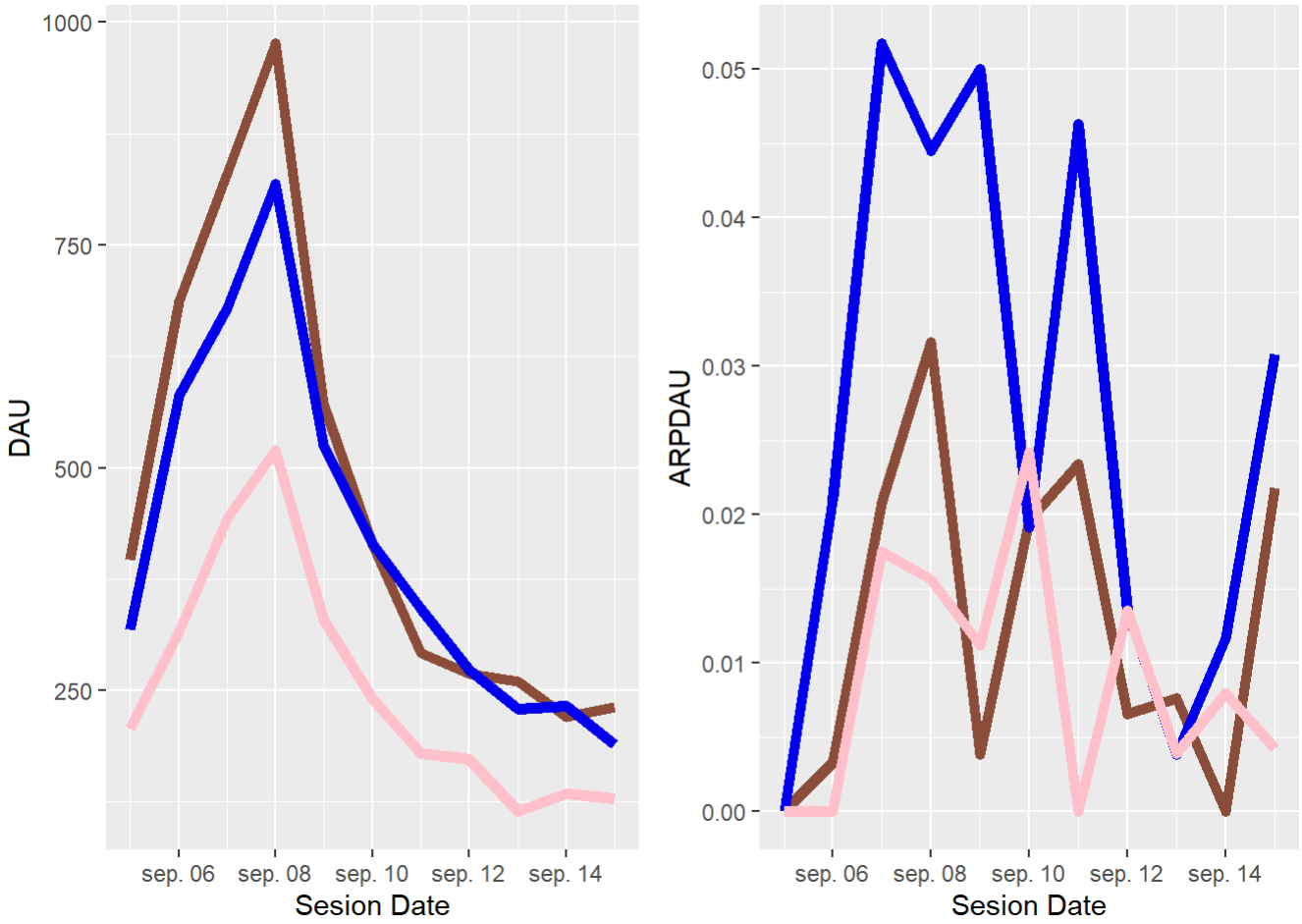
Una de las métricas más interesantes es la DAU, de aquí podemos extraer algunas conclusiones interesantes cuando valoramos el tipo de distribución que sigue:

```
E2_DAU = E2 %>% ggplot(aes(x=ses_date))+
```

```
  geom_line(aes(y=DAU_DE),color="salmon4", size=2)+
  geom_line(aes(y=DAU_FR),color="blue2", size=2)+
  geom_line(aes(y=DAU_GB),color="pink", size=2)+
  xlab("Sesion Date") +
  ylab("DAU")

E2_ARPDAU = E1_ARPDAU %>% ggplot(aes(x=ses_date))+
  geom_line(aes(y=ARPDAU_DE), colour="salmon4", size=2)+
  geom_line(aes(y=ARPDAU_FR), colour="blue2", size=2)+
  geom_line(aes(y=ARPDAU_GB), colour="pink", size=2)+
  xlab("Sesion Date") +
  ylab("ARPDAU")

grid.arrange(E2_DAU,E2_ARPDAU, nrow=1)
```



Y precisamente es interesante por el fenómeno que podemos observar en dicha evolución. Durante la primera mitad del ejercicio era plausible pensar que en alemania se estaban cumpliendo las mejores previsiones en las métricas en relación a la actividad de juego, ahora resumidas y condensadas en la métrica 'DAU', de la cual puede contemplarse una evolución bastante uniforme relativa a las tendencias en los 3 países, pero con un claro *"ganador"*= Alemania. Sin embargo más adelante hemos afinado la búsqueda teniendo en cuenta los ingresos por IAPs y aquí hemos comprobado un *cambio en las expectativas de éxito* relativas a los países, donde Francia ha demostrado poseer los mejores resultados relativos a las métricas principales usadas por los grandes proveedores de KPIs de videojuegos de móviles como Delta DNA (ARPDAU, ARPU, ARPPU…), tal como se ha reflejado en la segunda parte del ejercicio, abanderado por la Paying Share o % pagador.

Es muy interesante y digna de estudio (pero no vamos a entrar en profundidad aquí) el desequilibrio que se observa en las tendencias de la ARPDAU en la evolución de los diferentes países, sufrida principalmente por GB, de la cual hemos visto, *frente a todo pronóstico*, como ha resultado adelantar a Alemania en la tasa de pagadores, pese a tener menor contribución absoluta neta (revenues) que ésta.

```
E1
```

country <chr>	ses_avg <dbl>	time_avg <dbl>	time_ses_avg <dbl>	complvl_tot <dbl>
DE	467.8182	0.2250834	105.29809	18057
FR	418.2727	0.2292536	95.89054	16203
GB	253.1818	0.2175065	55.06868	9194

3 rows | 1-5 of 9 columns

Según observamos en la consulta 1: - Tiempo jugado por sesión (time_ses_avg) –> DE > FR >> GB - Media de Sesiones (ses_avg) –> DE > FR >> GB

Y sin embargo,

- ARPU –> FR >> GB > DE
- Pay_Share –> FR > GB >> DE

En definitiva, son interesantes las *aparentes* contradicciones.

Otras métricas interesantes serían estudiar las New Users, o los llamados K-Factor (que miden la incidencia por cantidad de invitaciones lanzadas por clientes), pero faltan datos de la BBDD extraída de los proveedores para sacar conclusiones. Y en caso de poseer estos datos se podría dar un paso más y obtener la métrica de densidad avanzada [Nº invitaciones enviadas / DAU], algo que podría arrojar mucha luz en la evolución de los datos, incluso de aquellos que presenten aparentes contradicciones, como hemos visto al principio de la exposición de este apartado.

Por último, sólo mencionar que podríamos hacer una estimación de las KPIs de retención a 1, 2,3 y varios días en base a usuarios, dado que poseemos datos de un período de 10 días con sus respectivas ID_session, tal como expuse en el ejercicio anterior que presenté para Genera, que más o menos tendría la siguiente forma:

KPI_ret = DI %>% mutate(DAY_ins=day(ins_date),DAY_ses=day(ses_date)) %>% group_by(user_id)%>% mutate(D1=ifelse(DAY==5,1,0),D2=ifelse(DAY==6,1,0), D3=ifelse(DAY==6,1,0),D4=ifelse(DAY==6,1,0)) %>% summarise(D1=ifelse(sum(D1)>=1,1,0), D2=ifelse(sum(D2)>=1,1,0), D3=ifelse(sum(D3)>=1,1,0), D4=ifelse(sum(D4)>=1,1,0)) %>% mutate(RT1=ifelse((D1+D2+D3+D4)>1,1,0),RT2=ifelse((D1+D2+D3+D4)>2,1,0),RT3=ifelse((D1+D2+D3+D4)>3,1,0)) %>% select(user_id,RT1,RT2,RT3)

etc….

Ejercicio 1.3.:

1.3.1. Media de sesiones:

Haciendo uso de las consultas hechas anteriormente, le aplicamos filtros para día 0 y resto de los días:

```
E3_ses_avg = DI %>% mutate(DAY=day(ses_date)-5) %>%
  group_by(country) %>%
  mutate(DAY_0 = ifelse(DAY==0,1,0),
         DAY_REST=ifelse(DAY>0,1,0)) %>%
  summarise(DAY_0=as.integer(sum(DAY_0)),
            DAY_REST=as.integer(sum(DAY_REST)/10),
            `INCREASE(%)`=((DAY_REST-DAY_0)/DAY_0)*100)
```

Francia ha tenido un incremento en media de sesiones de casi el 35%, pero en general como vemos se ha incrementado en todos los países.

1.3.2. Tiempo jugado por sesión (medio):

```
E3_time_ses_avg = DI %>%
  mutate(DAY=day(ses_date)-5) %>%
  group_by(country) %>%
  mutate(DAY_0=ifelse(DAY==0,game_time/(1000*60*60),0),
         DAY_REST=ifelse(DAY>0,game_time/(1000*60*60),0)) %>%
  summarise(DAY_0=sum(DAY_0),
            DAY_REST=sum(DAY_REST)/10,
            `INCREASE(%)`=((DAY_REST-DAY_0)/DAY_0)*100)
```

Y efectivamente, Francia es el unico que sufre un crecimiento positivo en tiempo de juego. Esto también deja de manifiesto que la evolución de la media de sesiones es independiente a la evolución del tiempo de juego por sesión; es decir, *pese a que aumente la media de sesiones en el tiempo, no implica que ello provoque un aumento en el tiempo medio de dichas sesiones*. En el caso de francia habría que estudiarlo más a fondo como haremos en el análisis exploratorio.

Ejercicio 1.4:

Pese a que se ya ha estado haciendo un análisis e interpretación de datos en el camino, vayamos más a fondo en algunas cuestiones:

4.1. Análisis del incremento y decremento en la evolución media de sesiones y tiempo medio de las mismas:

Tal como hemos visto en el ejercicio 3, vamos a **normalizar** ambas estadísticas para que podamos verlas reflejadas en un mismo gráfico:

```
  row_ses = cbind(E3_ses_avg[,2]/colMeans(E3_ses_avg[,2]),E3_ses_avg[,3]/colMeans(E3_ses_av
g[,3]))
  colnames(row_ses)=c("DAY_ses","DAY_ses")

  row_ses %>% gather(key=DAY_ses)
```

DAY_ses <dbl>	DAY_ses <dbl>
1.2917570	1.2269198
1.0347072	1.1078516
0.6735358	0.6652286

3 rows

```
  row_time = cbind(E3_time_ses_avg[,2]/colMeans(E3_time_ses_avg[,2]),E3_time_ses_avg[,3]/co
lMeans(E3_time_ses_avg[,3]))
  colnames(row_time)=c("DAYtime","DAYtime")
```

Pero hagámoslo con la evolución de días:

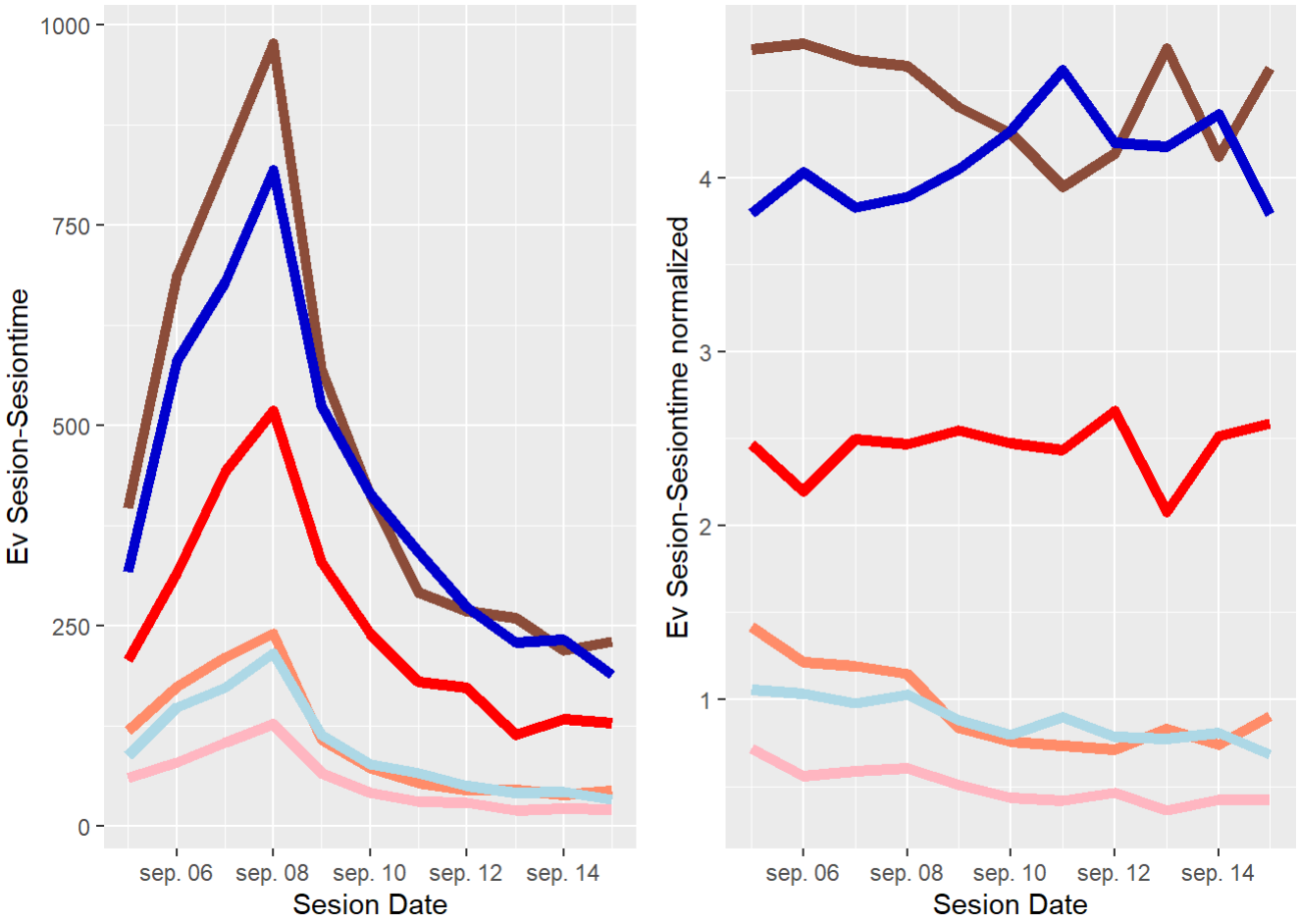
```
    E4.1 = DI %>% group_by(ses_date) %>%
      mutate(DE_ses=ifelse(country=="DE",1,0),
```

```
          FR_ses=ifelse(country=="FR",1,0),
          GB_ses=ifelse(country=="GB",1,0),
          DE_time=ifelse(country=="DE",game_time/(1000*60*60),0),
          FR_time=ifelse(country=="FR",game_time/(1000*60*60),0),
          GB_time=ifelse(country=="GB",game_time/(1000*60*60),0)) %>%
  summarise(DE_ses=sum(DE_ses),
            FR_ses=sum(FR_ses),
            GB_ses=sum(GB_ses),
            DE_time=sum(DE_time),
            FR_time=sum(FR_time),
            GB_time=sum(GB_time)) %>%
  ggplot(aes(x=ses_date),position="dodge")+
  geom_line(aes(y=DE_ses),color="salmon4",size=2)+
  geom_line(aes(y=FR_ses),color="blue3",size=2)+
  geom_line(aes(y=GB_ses),color="red",size=2)+
  geom_line(aes(y=DE_time),color="salmon1",size=2)+
  geom_line(aes(y=FR_time),color="lightblue",size=2)+
  geom_line(aes(y=GB_time),color="lightpink",size=2)+
  xlab("Sesion Date") +
  ylab("Ev Sesion-Sesiontime")
```

O normalizado:

```
E4.2 = DI %>% group_by(ses_date) %>%
  mutate(DE_ses=ifelse(country=="DE",1,0),
         FR_ses=ifelse(country=="FR",1,0),
         GB_ses=ifelse(country=="GB",1,0),
         DE_time=ifelse(country=="DE",game_time/(1000*60*60),0),
         FR_time=ifelse(country=="FR",game_time/(1000*60*60),0),
         GB_time=ifelse(country=="GB",game_time/(1000*60*60),0)) %>%
  summarise(DE_ses=sum(DE_ses/(n()/11)),
            FR_ses=sum(FR_ses/(n()/11)),
            GB_ses=sum(GB_ses/(n()/11)),
            DE_time=sum(DE_time/(n()/11)),
            FR_time=sum(FR_time/(n()/11)),
            GB_time=sum(GB_time/(n()/11))) %>%
  ggplot(aes(x=ses_date),position="dodge")+
  geom_line(aes(y=DE_ses),color="salmon4",size=2)+
  geom_line(aes(y=FR_ses),color="blue3",size=2)+
  geom_line(aes(y=GB_ses),color="red",size=2)+
  geom_line(aes(y=DE_time),color="salmon1",size=2)+
  geom_line(aes(y=FR_time),color="lightblue",size=2)+
  geom_line(aes(y=GB_time),color="lightpink",size=2)+
  xlab("Sesion Date") +
  ylab("Ev Sesion-Sesiontime normalized")
```

```
grid.arrange(E4.1,E4.2, nrow=1)
```

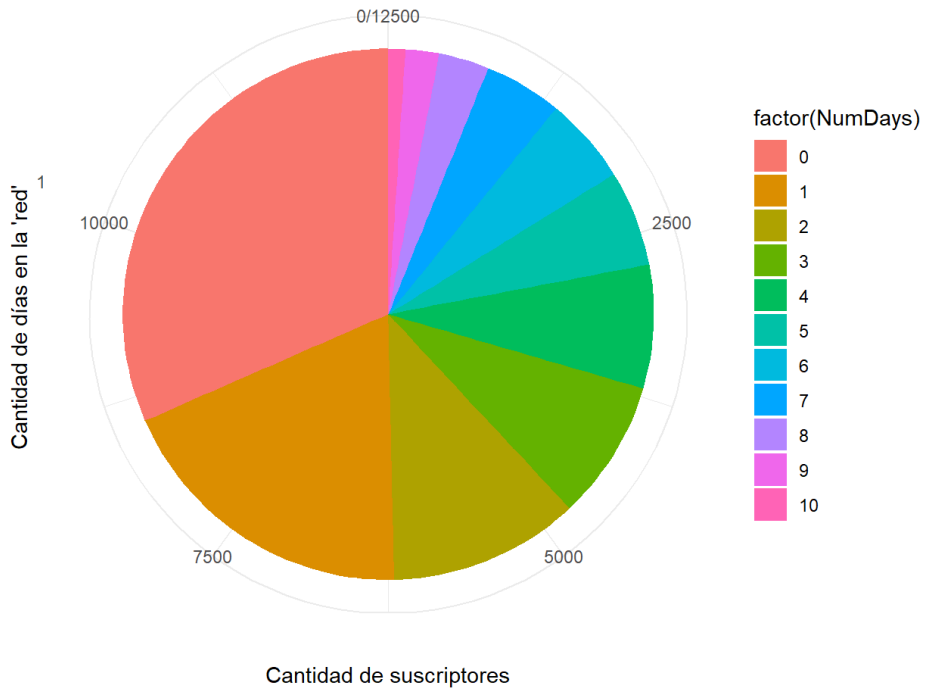


Donde los colores fuertes se refieren a las sesiones medias y los colores suaves a los *tiempos medios por sesión*. Aquí los incrementos que hemos calculado en el E3 no son tan visibles, y eso se debe a que el incremento percibido en lo que hemos considerado "DAY_REST", o días diferentes al Día 0, se concentran principalmente en unos picos percibidos en los días 3-4-5, para después caer por debajo incluso del día 0. Se puede estimar, a partir de estos gráficos como la KPI de retención a días comienza a decaer a partir de los 4-5 días desde la primera sesión. En la gráfica normalizada vemos una mayor estabilidad en los resultados ya que debido a su naturaleza lo que muestra es una tendencia más a largo plazo, y es lógico que no veamos cambios significativos.

Aquí por ejemplo vamos a estudiar la distribución de suscriptores que hay en relación a la cantidad de días (retención) que llevan jugando desde que instalaron el juego:

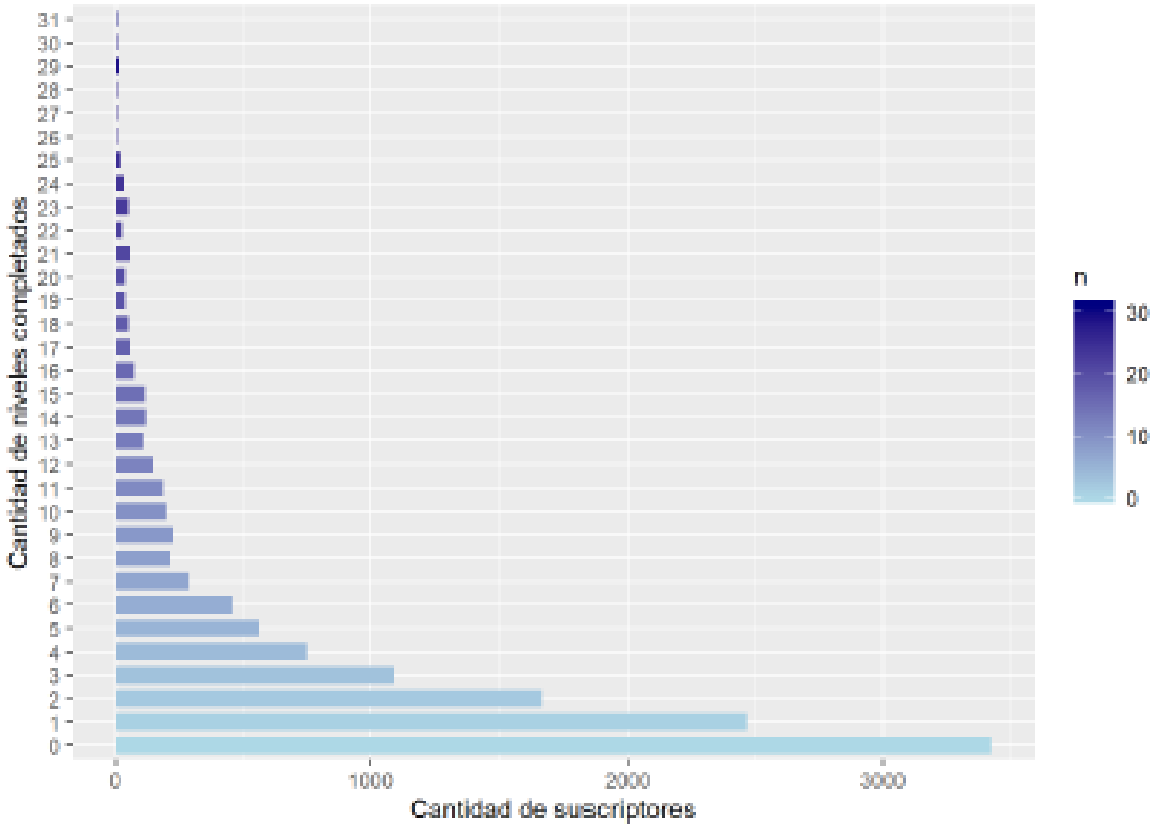
```
E5.2 = DI %>% group_by(ses_id) %>%
  distinct(ses_id, .keep_all = TRUE) %>%
  summarise(first_date=min(ins_date),
              last_date=max(ses_date),
              NumDays=last_date-ins_date)
```

```
  E5.2 %>% ggplot(aes(x = factor(1), fill = factor(NumDays))) +
  geom_bar(width = 1)+
  coord_polar(theta = "y")+
  theme_minimal()+
  xlab("Cantidad de días en la 'red' ")+
  ylab("Cantidad de suscriptores")
```



Aquí podemos observar el número de sesiones abiertas y cantidad de días permanecidos abriendo sesiones por ID de sesión. Todo tiene un aspecto relativamente normal

```
DI %>% group_by(ses_id) %>%
   distinct(ses_id, .keep_all = TRUE) %>%
   summarise(n=sum(comp_lvl)) %>%
   ggplot(aes(x=factor(n),fill=n))+
  scale_fill_gradient(low = "lightblue", high = "navy")+
   geom_bar(width=0.7)+
  coord_flip()+
  xlab("Cantidad de niveles completados")+
  ylab("Cantidad de suscriptores")
```



Sigue una distribución exponencial decreciente de usuarios, sin saltos destacables tampoco, excepto por el hecho de a partir de los 6-7 niveles completados el número de usuarios con probabilidad de éxito disminuye vertiginosamente.

Ejercicio 2:

2.3. Análisis exploratorio

```
 DL <- read_xlsx("Task2-Test_Indie_Analytics-IAPs.xlsx")
head(DL,20)
```

ID User <chr>	Install Date <S3: POSIXct>	Country <chr>	Platform <chr>	Purchase Date <S3: POSIXct>
3ffdda7828a5	2019-09-16	GB	ANDROID	2019-09-28
beff2249e4eb	2019-09-08	FR	ANDROID	2019-09-15
f12f2efd9bfc	2019-09-09	GB	ANDROID	2019-09-09
f12f2efd9bfc	2019-09-09	GB	ANDROID	2019-09-09
0730ab83d336	2019-10-10	US	ANDROID	2019-10-20
20448c712e9d	2019-09-06	US	IOS	2019-09-06
20448c712e9d	2019-09-06	US	IOS	2019-09-06
20448c712e9d	2019-09-06	US	IOS	2019-09-06
20448c712e9d	2019-09-06	US	IOS	2019-09-06
20448c712e9d	2019-09-06	US	IOS	2019-09-07

1-10 of 20 rows | 1-5 of 7 columns Previous **1** 2 Next

Cambiamos los nombres de las variables para hacer más facil su referenciación

```
 names(DL)[c(1,2,3,4,5,6,7)]=c("user_id","ins_date","country","platform","purchase_date","I
AP","purchase")
```

Observamos variables que no tienen la categoría correcta y que nos pueden dar problemas en el análisis

```
 Tot_usr2 = as.integer(DL %>% distinct(user_id, .keep_all = TRUE)
                       %>% summarise(n=n()))

 DL$ins_date = as.Date(DL$ins_date)
 DL$purchase_date = as.Date(DL$purchase_date)
```

```
 summary(DL)
```

```
##    user_id             ins_date            country
##  Length:5379        Min.   :2019-08-12   Length:5379
##  Class :character   1st Qu.:2019-09-04   Class :character
##  Mode  :character   Median :2019-09-13   Mode  :character
##                     Mean   :2019-09-15
##                     3rd Qu.:2019-09-25
##                     Max.   :2019-10-23
##    platform         purchase_date            IAP
##  Length:5379        Min.   :2019-08-12   Length:5379
##  Class :character   1st Qu.:2019-09-10   Class :character
##  Mode  :character   Median :2019-09-23   Mode  :character
##                     Mean   :2019-09-23
##                     3rd Qu.:2019-10-08
##                     Max.   :2019-10-24
##     purchase
##  Min.   : 1.00
##  1st Qu.: 1.00
##  Median : 1.00
##  Mean   : 1.29
##  3rd Qu.: 1.00
##  Max.   :20.00
```

```
 glimpse(DL)
```

```
## Observations: 5,379
## Variables: 7
## $ user_id       <chr> "3ffdda7828a5", "beff2249e4eb", "f12f2efd9bfc", ...
## $ ins_date      <date> 2019-09-16, 2019-09-08, 2019-09-09, 2019-09-09,...
## $ country       <chr> "GB", "FR", "GB", "GB", "US", "US", "US", "US", ...
## $ platform      <chr> "ANDROID", "ANDROID", "ANDROID", "ANDROID", "AND...
## $ purchase_date <date> 2019-09-28, 2019-09-15, 2019-09-09, 2019-09-09,...
## $ IAP           <chr> "SpecialPack0", "Pack1", "SpecialPack0", "Pack1"...
## $ purchase      <dbl> 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, ...
```

Tenemos una primera visual, y posteriormente procedemos al análisis:

```
 Tot_Period = as.integer(max(DL$purchase_date) - min(DL$purchase_date))
```

Un total de 73 días de muestreo jugado

```
U1.1 = DL %>% group_by(country) %>%
  summarise(n=n()) %>%
  ggplot(aes(x=country, y=n,fill=n)) +
  geom_bar(stat="identity", width=0.8)+
  scale_fill_gradient(low = "lightblue", high = "navy")+
  xlab("Países")+
  ylab("Cantidad de suscriptores")
```

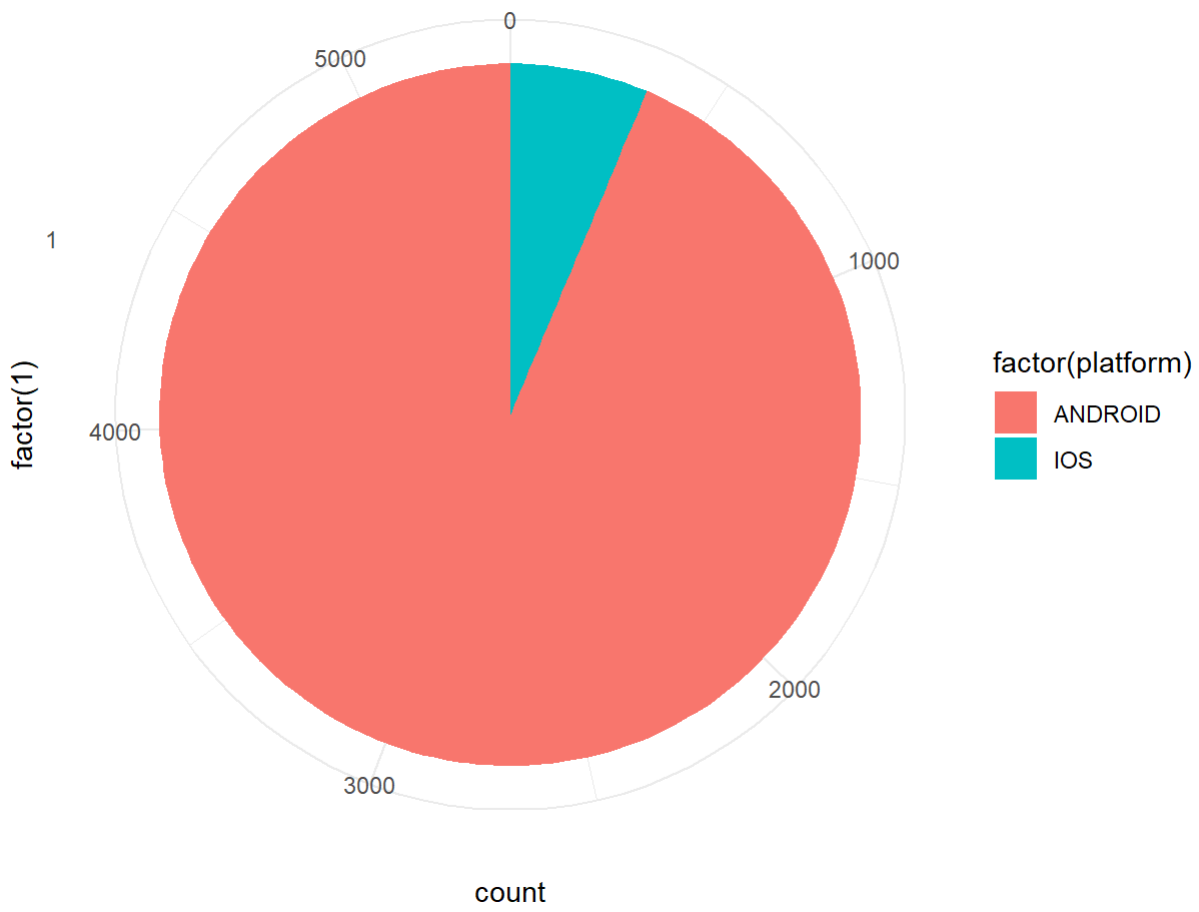
```
DL %>% group_by(country) %>%
  summarise(n=n()) %>% summary()
```

```
##    country                n          
##  Length:10          Min.   :  90.0  
##  Class :character   1st Qu.: 127.8  
##  Mode  :character   Median : 233.0  
##                     Mean   : 537.9  
##                     3rd Qu.: 689.5  
##                     Max.   :2223.0
```

Aquí observamos que tenemos un total de 10 países contemplados en el análisis, donde claramente el Estados Unidos quien lidera con casi 2000 suscriptores registrados en el período contemplado seguido de Alemania con casi 900 y el resto de países que ya se encuentran en el percentil 50% por debajo de 200 suscriptores.

```
U1.2 = DL %>% distinct(user_id, .keep_all = TRUE) %>%
  group_by(platform) %>%
  summarise(n=n_distinct(user_id),`%`=(n_distinct(user_id)/Tot_usr2)*100)
```

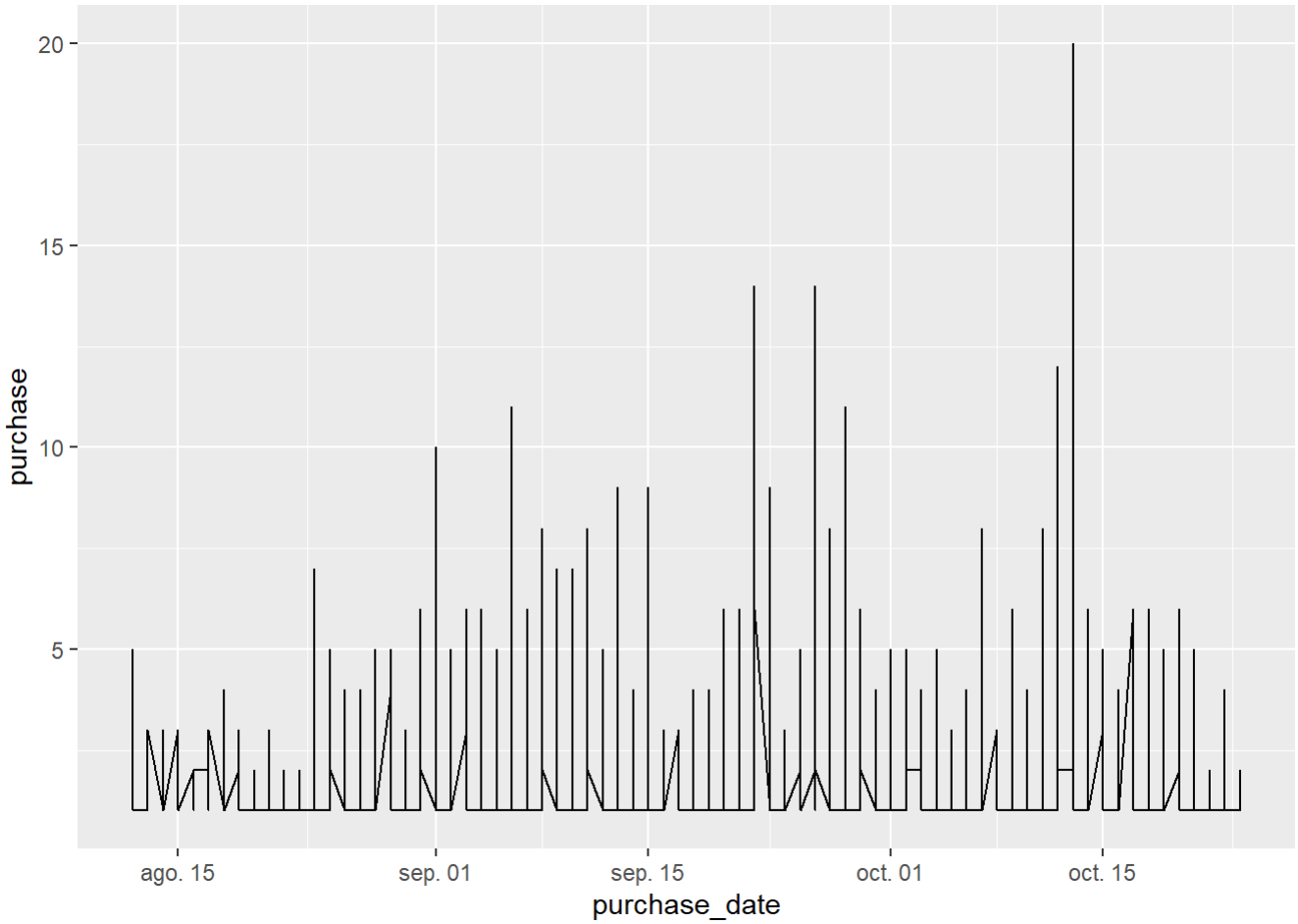
```
DL %>% ggplot(aes(x = factor(1), fill = factor(platform))) +
  geom_bar(width = 1)+
  coord_polar(theta = "y")+
  theme_minimal()
```



Y Android como sistema operativo líder (con una cuota mayor al 95%)

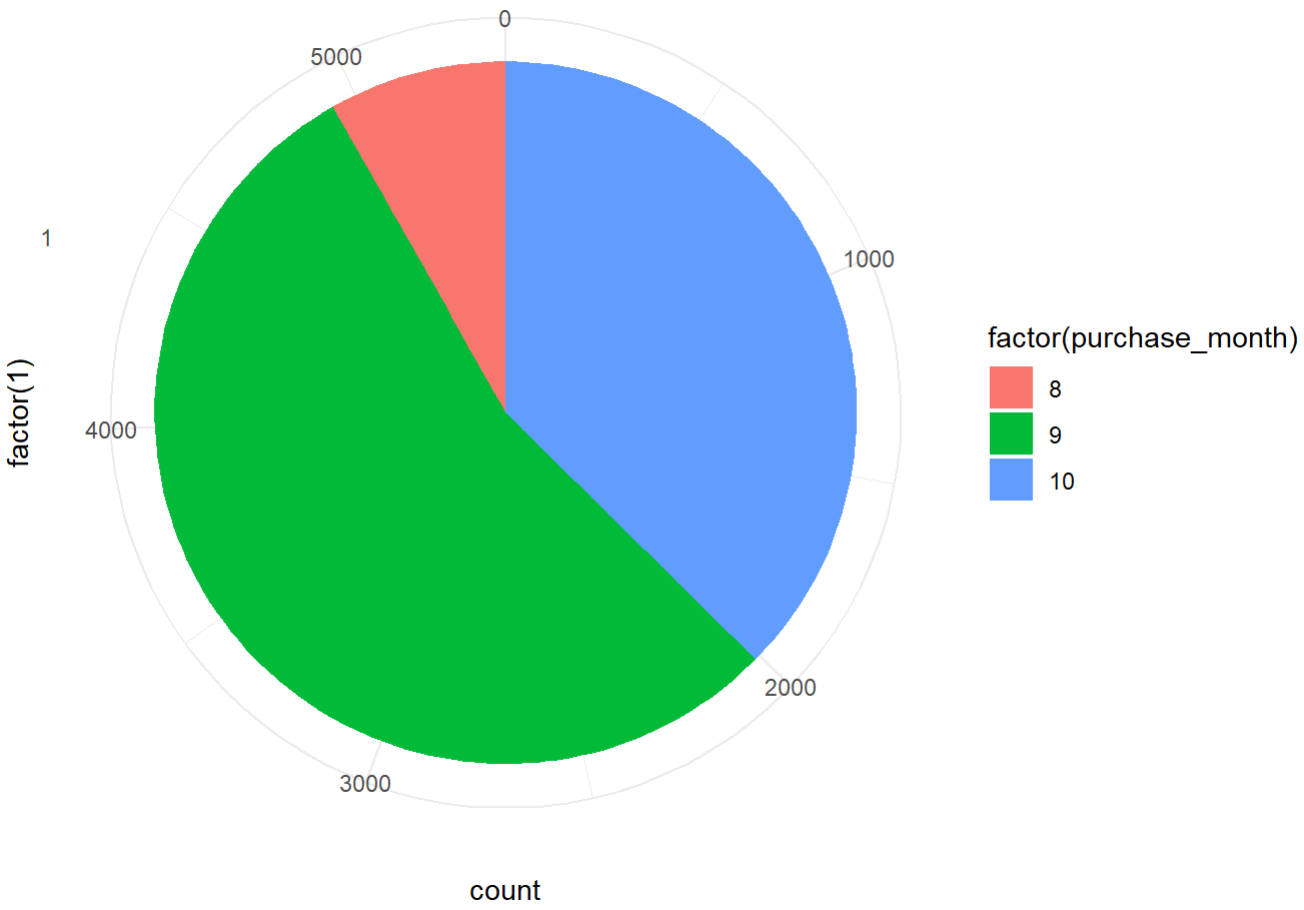
A continuación exploramos la tabla al completo. Haremos un primer chequeo en la relación de la cantidad de comprar s/ día:

```
DL %>% ggplot(aes(x=purchase_date,y=purchase))+
  geom_line()
```



Conforme avanza el año las compras aumentan de máximos, pero desconocemos su densidad:

```
DL %>% mutate(purchase_month=month(purchase_date)) %>%
  group_by(purchase_month) %>%
  ggplot(aes(x = factor(1), fill = factor(purchase_month))) +
  geom_bar(width = 1)+
  coord_polar(theta = "y")+
  theme_minimal()
```



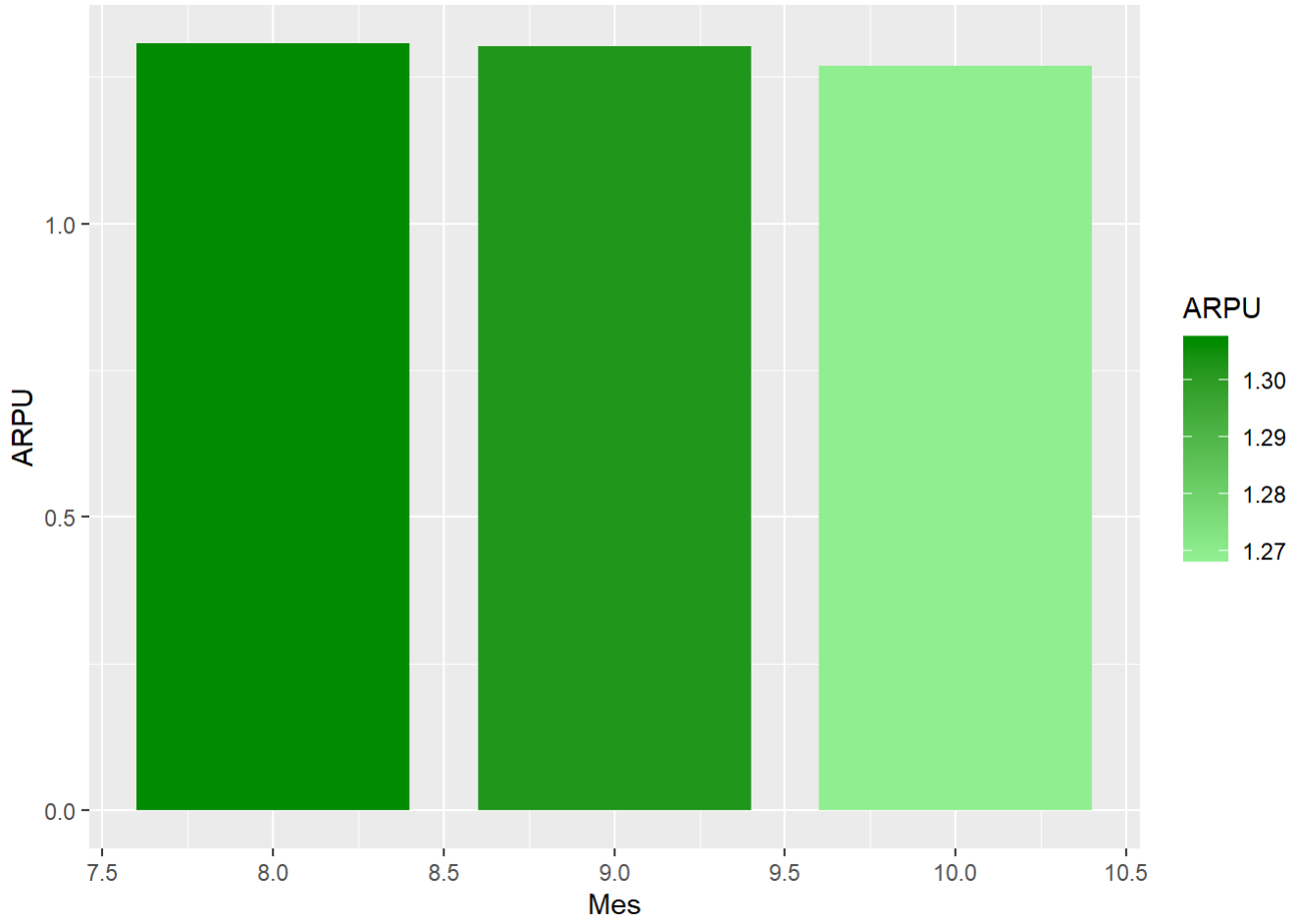
El mes de septiembre aglutina la mayoría absoluta de compras desde que se comenzó el análisis, claro que esto no es fidedigno ya que el mes de agosto comienza a contar casi a mitad de mes, de manera que calculamos la densidad de compras por usuarios al mes, o dicho de otro modo, al ARPU

```
DL %>% mutate(purchase_month=month(purchase_date)) %>%
  group_by(purchase_month) %>%
  summarise(ARPU=sum(purchase)/n())
```

purchase_month <dbl>	ARPU <dbl>
8	1.306818
9	1.302596
10	1.269020

3 rows

```
DL %>% mutate(purchase_month=month(purchase_date)) %>%
  group_by(purchase_month) %>%
  summarise(ARPU=sum(purchase)/n()) %>%
  ggplot(aes(x=purchase_month, y=ARPU,fill=ARPU)) +
  geom_bar(stat="identity", width=0.8)+
  scale_fill_gradient(low = "lightgreen", high = "Green4")+
  xlab("Mes")+
  ylab("ARPU")
```



Ahora sí, comprobamos que efectivamente en agosto es cuando hay mayor densidad de compras, quizás tenga que ver con el hecho de que la gente tiene más tiempo libre de vacaciones y puede jugar más, y poco a poco decrece Exploramos también la cantidad de IAPs y su frecuencia de compra:

```
U3.1 =DL %>% group_by(IAP) %>%
  summarise(quantity=n(),
            gain=sum(purchase),
            effect=sum(purchase)/n()) %>%
  arrange(effect)
```

```
summary(U3.1)
```

```
##      IAP                quantity          gain           effect
##  Length:13          Min.   :   1.0   Min.   :   1.0   Min.   :1.000
##  Class :character   1st Qu.:  23.0   1st Qu.:  25.0   1st Qu.:1.087
##  Mode  :character   Median : 260.0   Median : 348.0   Median :1.138
##                     Mean   : 413.8   Mean   : 533.9   Mean   :1.210
##                     3rd Qu.: 464.0   3rd Qu.: 509.0   3rd Qu.:1.364
##                     Max.   :1881.0   Max.   :2810.0   Max.   :1.500
```

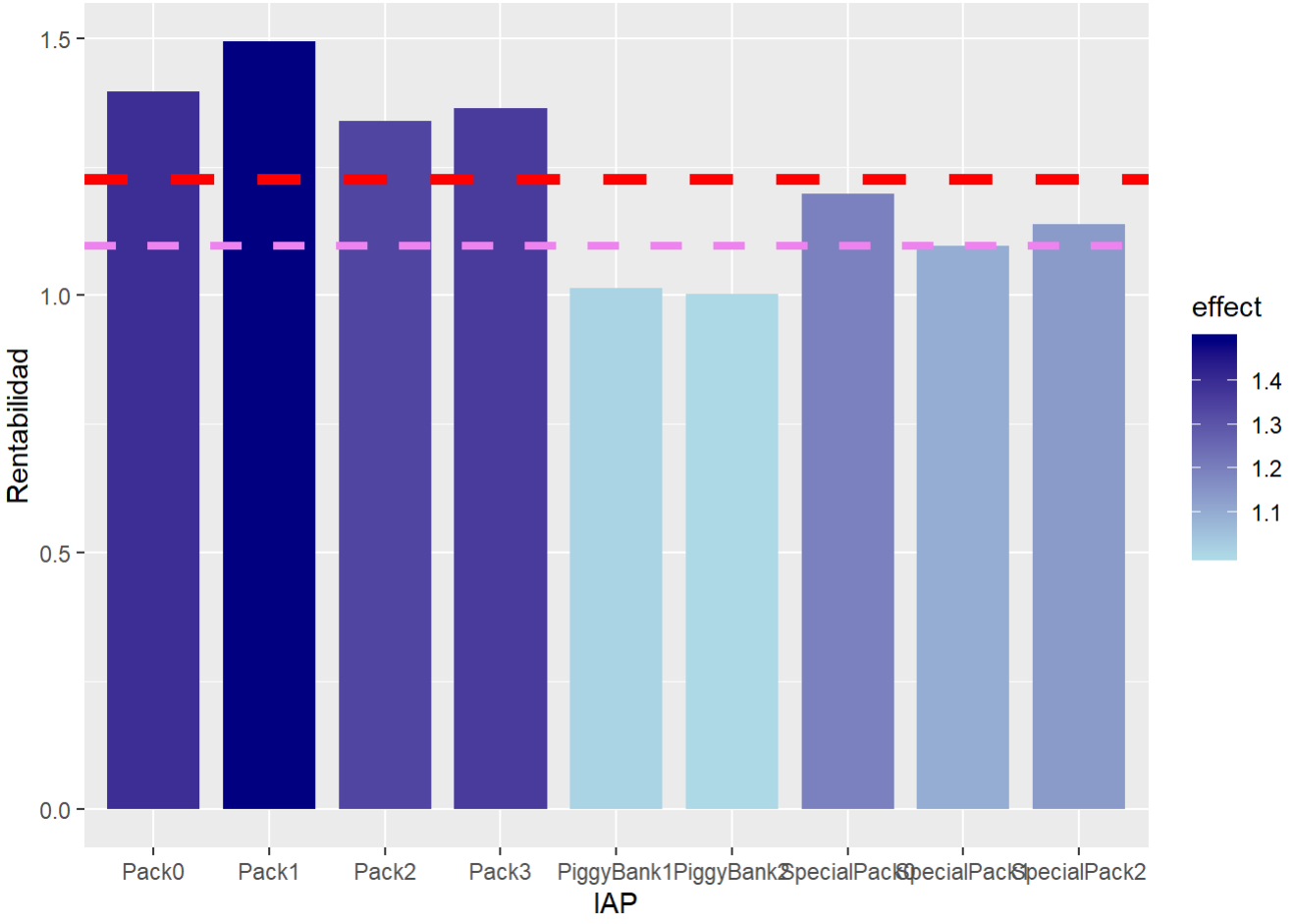
Ahí tenemos un orden de la rentabilidad económica de los diferentes IAPs, en los que el pack5 parece que es el que tiene mayor rentabilidad pero no es representativo, ya que se ha consumido muy poco, de manera que hacemos limpieza de los IAP que consideremos despreciables, basándonos en la cantidad correspondiente al primer cuartil, 23 unidades, y queda lo siguiente:

```
U3.1 = U3.1 %>% filter(quantity>23)
```

```
summary(U3.1)
```

```
##      IAP                quantity          gain            effect
##  Length:9           Min.   : 107.0   Min.   : 132.0   Min.   :1.002
##  Class :character   1st Qu.: 260.0   1st Qu.: 348.0   1st Qu.:1.097
##  Mode  :character   Median : 459.0   Median : 460.0   Median :1.198
##                     Mean   : 593.7   Mean   : 766.8   Mean   :1.227
##                     3rd Qu.: 499.0   3rd Qu.: 697.0   3rd Qu.:1.364
##                     Max.   :1881.0   Max.   :2810.0   Max.   :1.494
```

```
U3.1 %>% ggplot(aes(x=IAP, y=effect,fill=effect)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "lightblue", high = "navy") +
  xlab("IAP") +
  ylab("Rentabilidad") +
  geom_hline(yintercept=mean(U3.1$effect), linetype="dashed", color="red", size=2)+
geom_hline(yintercept=quantile(U3.1$effect,probs = 0.25), linetype="dashed", color="violet"
, size=1.5)
```



Donde el Pack1 gana con creces, teniendo el récord en ganancias y en rentabilidad, así como cantidad consumida. En base a lo observado aquí, y con tal de simplificar un poco el análisis posterior, vamos a agrupar los IAP en 3 grupos: - Grupo 1 (Correspondiente a los más rentables): Pack1, 0, 3 y 2 - Grupo 2 (Correspondiente a los medios (se encuentran entre media y quartil 25%)): SpecialPacks - Grupo 3 (Correspondiente a los menos rentables): PiggyBanks

2.1.1. Métricas por país

```
U2.1.1 = DL %>% group_by(country) %>%
```

```
  mutate(AND=ifelse(platform=="ANDROID",1,0),
         IOS=ifelse(platform=="IOS",1,0),
         G1=ifelse(IAP=="Pack0" | IAP=="Pack1" | IAP=="Pack2" | IAP=="Pack3",1,0),
         G2=ifelse(IAP=="SpecialPack0" | IAP=="SpecialPack1" | IAP=="SpecialPack2" ,1,0),
         G3=ifelse(IAP=="Pack4" | IAP=="Pack5" | IAP=="Pack6" | IAP=="SpecialPack3" | IAP==
"PiggyBank1" | IAP=="PiggyBank2" ,1,0)) %>%
  summarise(pref_plat=ifelse(sum(AND)>sum(IOS),"ANDROID","IOS"),
            tot_purch=sum(purchase),
            freq_purch_sub=sum(purchase)/n(), ###frecuencia por suscripción o jugada
            freq_purch=sum(purchase)/n_distinct(user_id), ##Frecuencia por cantidad de usua
rios totales
            G1_rate=sum(G1)/n(),
            G2_rate=sum(G2)/n(),
            G3_rate=sum(G3)/n()) %>%
  arrange(freq_purch)
```

Esta tabla completa arroja una series de datos muy interesantes. En orden descendente (de mayor a menor, valoramos las siguientes métricas) - tot_purch: US >> DE > FR >> rest - freq_purch: FR > SE > CA - G1_rate: GB > FR > SE - G2_rate: DK > CA > CH - G3_rate: CH > US > DK

```
U2.1.1.totpurch = U2.1.1 %>% ggplot(aes(x=country, y=tot_purch,fill=tot_purch)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "lightgreen", high = "green4") +
  xlab("País") +
  ylab("Total de compras")
```

```
U2.1.1.freq_purch = U2.1.1 %>% ggplot(aes(x=country, y=freq_purch,fill=freq_purch)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "lightblue", high = "navy") +
  xlab("País") +
  ylab("Frecuencia de compras")
```

```
U2.1.1.freq_purch_sub = U2.1.1 %>% ggplot(aes(x=country, y=freq_purch_sub,fill=freq_purch_s
ub)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "plum", high = "purple4") +
  xlab("País") +
  ylab("Frecuencia de compras por suscriptor o jugada")
```

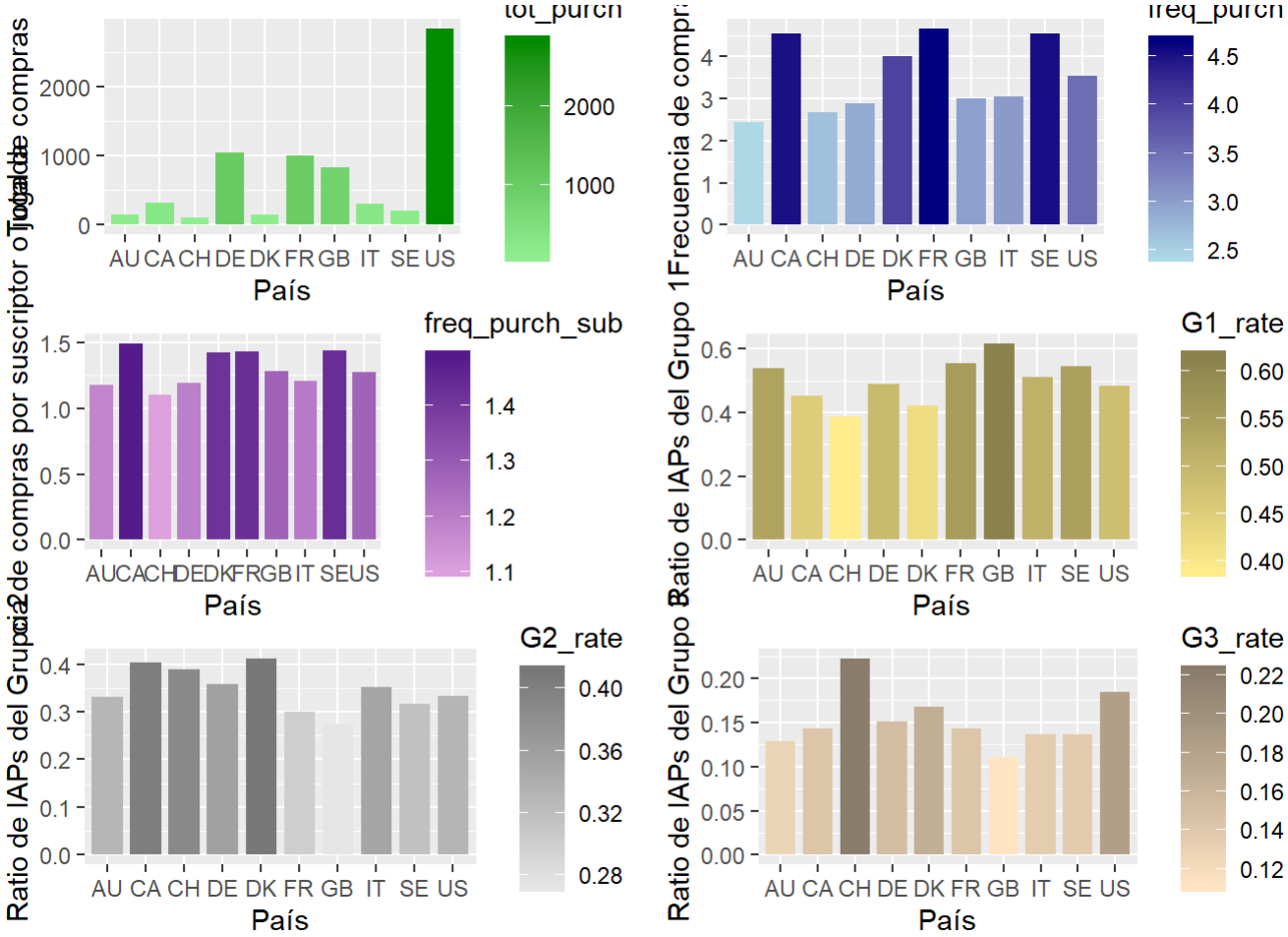
```
U2.1.1.G1_rate = U2.1.1 %>% ggplot(aes(x=country, y=G1_rate,fill=G1_rate)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "lightgoldenrod1", high = "lightgoldenrod4") +
  xlab("País") +
  ylab("Ratio de IAPs del Grupo 1")
```

```
U2.1.1.G2_rate = U2.1.1 %>% ggplot(aes(x=country, y=G2_rate,fill=G2_rate)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "gray90", high = "grey47") +
  xlab("País") +
  ylab("Ratio de IAPs del Grupo 2")
```

```
U2.1.1.G3_rate = U2.1.1 %>% ggplot(aes(x=country, y=G3_rate,fill=G3_rate)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "bisque1", high = "bisque4") +
```

```
  xlab("País") +
  ylab("Ratio de IAPs del Grupo 3")
```

```
grid.arrange(U2.1.1.totpurch,U2.1.1.freq_purch,U2.1.1.freq_purch_sub,U2.1.1.G1_rate,U2.1.1.
G2_rate,U2.1.1.G3_rate, nrow=2)
```



Tras la exposición de datos y gráficos, sumariamos lo siguiente:
- En el total de compras US gana muy por encima de la media, pero este dato generalmente no es un buen indicador ya que las estádisticas nunca se miden en términos absolutos, por el simple hecho de que US tiene mayor población objetiva que el resto de países que se analizan. Dicho esto, es un país objetivo importante
- Según la frecuencia de compras, que probablemente es la métrica más importante, Francia está por encima de los demás, seguido de cerca de Suecia y finalmente seguido de Canadá. Parece que los países Franco-parlantes son un público objetivo.
- Según los ratios de IAPs del grupo 1, que son los importantes y en los que apuntaremos comentarios, decir que Gran Bretaña se encuentra a la cabeza, seguido, de nuevo, de Francia y Suecia.

Conclusiones:
- Francia es el activo más importante como país, y donde, suponemos, se encuentra el público objetivo más importante a salvaguardar.
- Suecia tiene un potencial similar, pese a que no reporta grandes beneficios en términos absolutos. - Estados unidos y Alemania, como ya vimos también en el ejercicio anterior, son activos muy importantes no por el factor *calidad*, si no más bien por el de cantidad. Merecen una atención especial igualmente.
-Finalmente, GB parece ser uno de los activos crecientes con más potencial. Arrojando los mejores resultados en el consumo de IAPs de mayor rentabilidad (grupo 1), también tiene valores altos en casi todos los ratios, podría situarse a la cabeza en los próximos años si se sigue trabajando en las necesidades objetivas del público británico.

2.1.2. Métricas por plataforma

Sobre las métricas por plataforma no haré gran hincapié ya que, como hemos visto, el uso de los videojuegos en plataformas IOS no supera apenas el 5% de los usuarios totales, lo que convierte el análisis de cualquier métrica relativamente *despreciable*, estadísticamente hablando.

2.2. Ejercicio 2: Métricas por día 0 vs resto

```
summary(DL)
```

```
##    user_id             ins_date            country
##  Length:5379        Min.   :2019-08-12   Length:5379
##  Class :character   1st Qu.:2019-09-04   Class :character
##  Mode  :character   Median :2019-09-13   Mode  :character
##                     Mean   :2019-09-15
##                     3rd Qu.:2019-09-25
##                     Max.   :2019-10-23
##    platform         purchase_date            IAP
##  Length:5379        Min.   :2019-08-12   Length:5379
##  Class :character   1st Qu.:2019-09-10   Class :character
##  Mode  :character   Median :2019-09-23   Mode  :character
##                     Mean   :2019-09-23
##                     3rd Qu.:2019-10-08
##                     Max.   :2019-10-24
##     purchase
##  Min.   : 1.00
##  1st Qu.: 1.00
##  Median : 1.00
##  Mean   : 1.29
##  3rd Qu.: 1.00
##  Max.   :20.00
```

Calculamos el Nº de días totales en los que se hace el análisis

```
  Tot_days = as.integer(max(DL$purchase_date)-min(DL$purchase_date))
```

```
DL2 = DL %>% mutate(DAY=as.integer(purchase_date-min(purchase_date)))
```

```
U3.1 = DL2 %>% group_by(country) %>%
  filter(DAY<2) %>%
  mutate(AND=ifelse(platform=="ANDROID",1,0),
         IOS=ifelse(platform=="IOS",1,0),
         G1=ifelse(IAP=="Pack0" | IAP=="Pack1" | IAP=="Pack2" | IAP=="Pack3",1,0),
         G2=ifelse(IAP=="SpecialPack0" | IAP=="SpecialPack1" | IAP=="SpecialPack2" ,1,0),
         G3=ifelse(IAP=="Pack4" | IAP=="Pack5" | IAP=="Pack6" | IAP=="SpecialPack3" | IAP==
"PiggyBank1" | IAP=="PiggyBank2" ,1,0)) %>%
  summarise(pref_plat=ifelse(sum(AND)>sum(IOS),"ANDROID","IOS"),
            tot_purch=sum(purchase)/(2),
            freq_purch=sum(purchase)/n_distinct(user_id), ##Frecuencia por cantidad de usua
rios totales
            G1_rate=sum(G1)/n(),
            G2_rate=sum(G2)/n(),
            G3_rate=sum(G3)/n()) %>%
  arrange(G3_rate)
```

Escojo como el *día 0* hasta el día 2 del muestreo, ya que el primer día apenas se incorporan países, y es una manera simplificada de hacerlo, pese a que se podría seleccionar dia 0 por cada país:

```
U3.2 = DL2 %>% group_by(country) %>%
  filter(DAY>1) %>%
  mutate(AND=ifelse(platform=="ANDROID",1,0),
```

```
        IOS=ifelse(platform=="IOS",1,0),
        G1=ifelse(IAP=="Pack0" | IAP=="Pack1" | IAP=="Pack2" | IAP=="Pack3",1,0),
        G2=ifelse(IAP=="SpecialPack0" | IAP=="SpecialPack1" | IAP=="SpecialPack2" ,1,0),
        G3=ifelse(IAP=="Pack4" | IAP=="Pack5" | IAP=="Pack6" | IAP=="SpecialPack3" | IAP==
"PiggyBank1" | IAP=="PiggyBank2" ,1,0)) %>%
  summarise(pref_plat=ifelse(sum(AND)>sum(IOS),"ANDROID","IOS"),
            tot_purch=sum(purchase/(Tot_days-2)),
            freq_purch=(sum(purchase)/n_distinct(user_id)), ##Frecuencia por cantidad de us
uarios totales, que tenemos que dividir por 72, para que podamos comparar datos
            G1_rate=sum(G1)/n(),
            G2_rate=sum(G2)/n(),
            G3_rate=sum(G3)/n()) %>%
  arrange(G3_rate)
```

Es difícil valorar los datos, pese a que lo importante aquí son los ratios relativos a la población completa

Vemos métricas del Día 0:

```
U3.1.totpurch = U3.1 %>% ggplot(aes(x=country, y=tot_purch,fill=tot_purch)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "lightgreen", high = "green4") +
  xlab("País") +
  ylab("Total de compras")
```

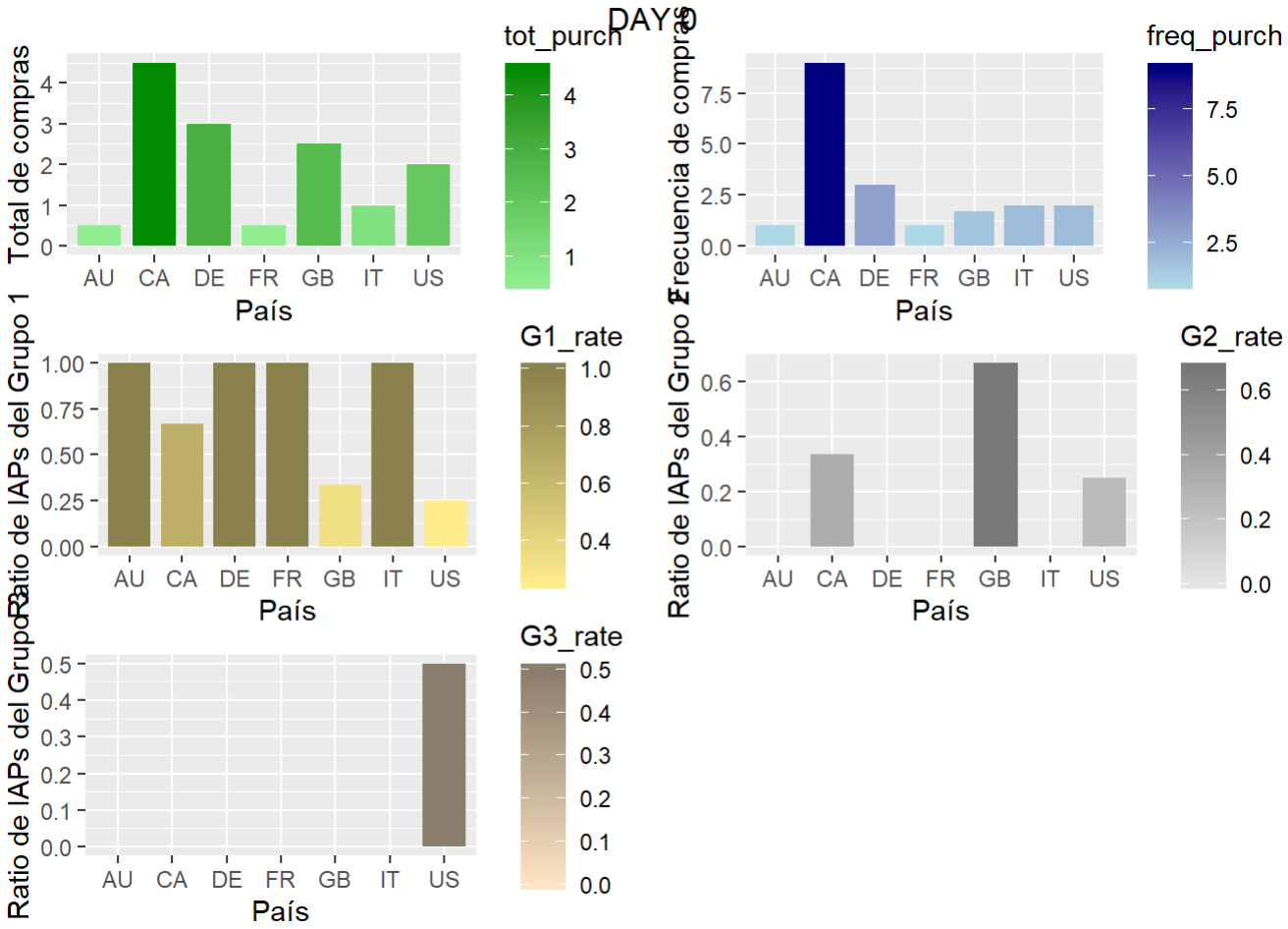
```
U3.1.freq_purch = U3.1 %>% ggplot(aes(x=country, y=freq_purch,fill=freq_purch)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "lightblue", high = "navy") +
  xlab("País") +
  ylab("Frecuencia de compras")
```

```
U3.1.G1_rate = U3.1 %>% ggplot(aes(x=country, y=G1_rate,fill=G1_rate)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "lightgoldenrod1", high = "lightgoldenrod4") +
  xlab("País") +
  ylab("Ratio de IAPs del Grupo 1")
```

```
U3.1.G2_rate = U3.1 %>% ggplot(aes(x=country, y=G2_rate,fill=G2_rate)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "gray90", high = "grey47") +
  xlab("País") +
  ylab("Ratio de IAPs del Grupo 2")
```

```
U3.1.G3_rate = U3.1 %>% ggplot(aes(x=country, y=G3_rate,fill=G3_rate)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "bisque1", high = "bisque4") +
  xlab("País") +
  ylab("Ratio de IAPs del Grupo 3")
```

```
grid.arrange(U3.1.totpurch,U3.1.freq_purch,U3.1.G1_rate,U3.1.G2_rate,U3.1.G3_rate, nrow=2,t
op="DAY 0")
```



Vemos métricas del resto de días:

```
U3.2.totpurch = U3.2 %>% ggplot(aes(x=country, y=tot_purch,fill=tot_purch)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "lightgreen", high = "green4") +
  xlab("País") +
  ylab("Total de compras")
```

```
U3.2.freq_purch = U3.2 %>% ggplot(aes(x=country, y=freq_purch,fill=freq_purch)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "lightblue", high = "navy") +
  xlab("País") +
  ylab("Frecuencia de compras")
```

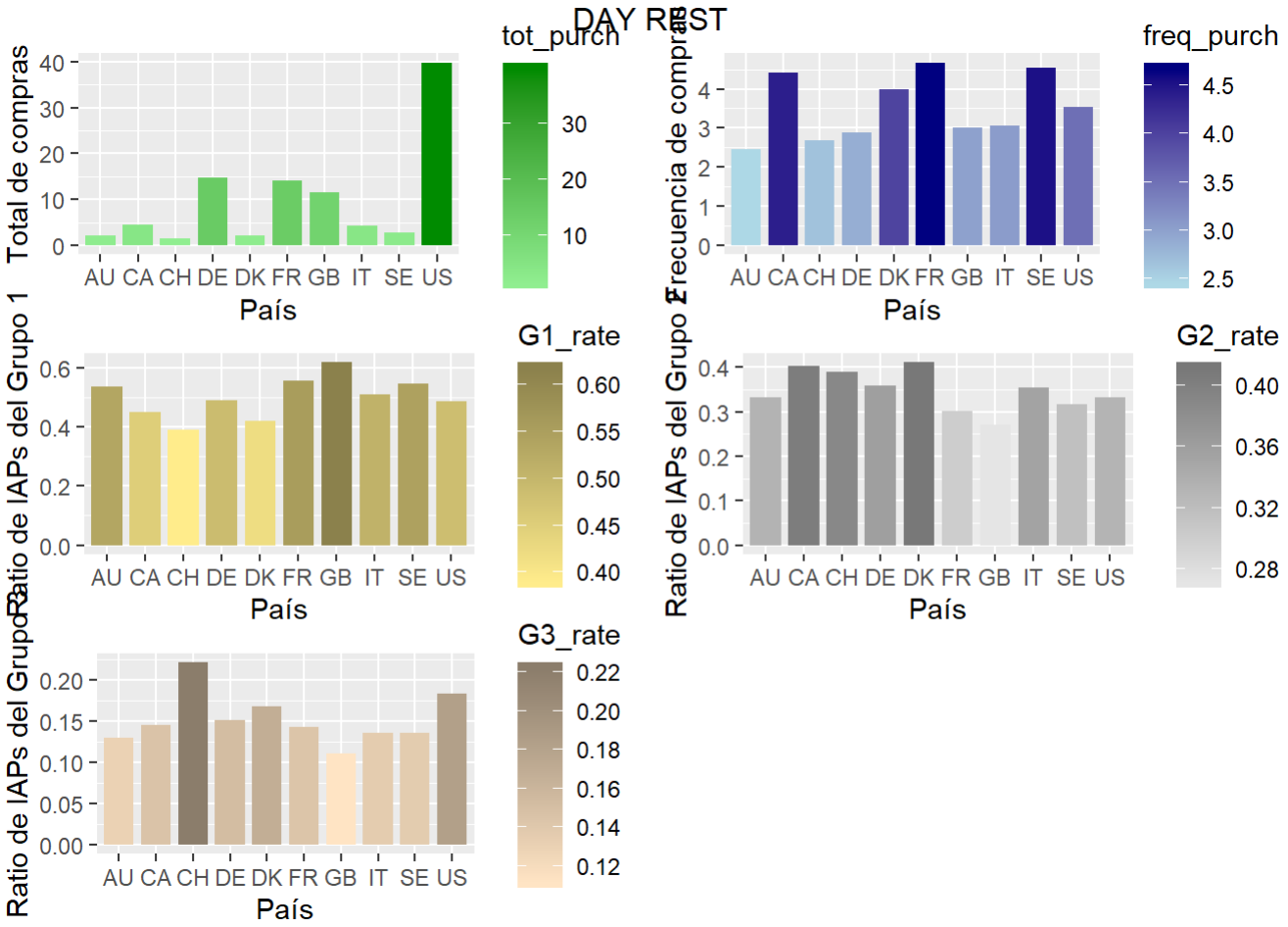
```
U3.2.G1_rate = U3.2 %>% ggplot(aes(x=country, y=G1_rate,fill=G1_rate)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "lightgoldenrod1", high = "lightgoldenrod4") +
  xlab("País") +
  ylab("Ratio de IAPs del Grupo 1")
```

```
U3.2.G2_rate = U3.2 %>% ggplot(aes(x=country, y=G2_rate,fill=G2_rate)) +
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "gray90", high = "grey47") +
  xlab("País") +
  ylab("Ratio de IAPs del Grupo 2")
```

```
U3.2.G3_rate = U3.2 %>% ggplot(aes(x=country, y=G3_rate,fill=G3_rate)) +
```

```
  geom_bar(stat="identity", width=0.8) +
  scale_fill_gradient(low = "bisque1", high = "bisque4") +
  xlab("País") +
  ylab("Ratio de IAPs del Grupo 3")
```

```
grid.arrange(U3.2.totpurch,U3.2.freq_purch,U3.2.G1_rate,U3.2.G2_rate,U3.2.G3_rate, nrow=2,top="DAY REST")
```



Es posible que no se esté profundizando lo suficiente en este apartado. Habría que estudiar por separado el caso de cada país y estudiar las retenciones o métricas a día 0 y resto de días como sí pudimos hacer en mayor profundidad en el ejercicio anterior, pero requeriría muchas horas de programación que para este caso no creo que sean necesarias.

Las conclusiones obtenidas en base a los análisis ejecutados en este ejercicio se encuentran resumidas antes de estos conjuntos de gráficos y espero que hayan contribuido a la mejora de la compresión de los interrogantes que se han planteado.

Ejercicio 3:

Apertura de archivos

```
getwd()
```

```
## [1] "D:/MAIN/CODING/R/WORKS/GENERA/INDIE"
```

```
GT = read.csv("TASK3.csv",sep=";")

lm_eqn <- function(df,y,x){
  m <- lm(y ~ x, df);
  eq <- substitute(italic(y) == a + b %.% italic(x)*","~~italic(r)^2~"="~r2,
                   list(a = format(unname(coef(m)[1]), digits = 2),
                        b = format(unname(coef(m)[2]), digits = 2),
                        r2 = format(summary(m)$r.squared, digits = 3)))
  as.character(as.expression(eq));
}
```

Análisis exploratorio

Premisas:

1. Los datos están extraídos de un juego tipo “saga” en el que no existen los farmeos.
2. La dificultad no es más que la media de intentos por nivel:

- La dificultad se calculará pues, DIF = completed/(fail+completed)

3. El flujo de acciones dentro de una partida debe seguir el esquema:
4. Cada ‘ID Mission’ solo puede contener un missionStarted.
5. No contabilizan partidas con ayuda inicial.

- He observado que el valor del Booster se conserva desde el inicio hasta el final de la mission, lo que quiere decir, que si filtramos quitando todo Booster, estaríamos quitando las partidas con ayudas iniciales, de forma sencilla y directa

6. No contabilizan partidas tras uso de continues, solo hasta el momento de usarlo.

- Esta condición está directamente ligada a la condición 4), el razonamiento a seguir es sencillo:
  - la ID de misión es clave para nuestra limpieza y filtraje. Cada ID_mision solo puede contener un mission Started
  - Y una sola acción posterior: mCompleted, mFailed, mAbandoned(está no influirá en la dificultad), y como lo único que puede ocurrir dentro de la mission_id para que haya *más de dos filas* es el uso de continues, y éstos no se van a contar, pues sencillamente filtramos por grupo manteniendo sólo las 2 primeras filas

```
names(GT) = c("user_id","action","level_id","mission_id","B1","B2","B3","leftover")

summary(GT)
```

```
##          user_id                   action         level_id
##  49989f01f069:  376   missionAbandoned:  103   Min.   :40.0
##  e74b14927d1e:  327   missionCompleted: 6512   1st Qu.:44.0
##  7b521e79e19d:  325   missionFailed   :13422   Median :49.0
##  14ca17d9503f:  302   missionStarted  :20319   Mean   :51.2
##  e6b5582aa050:  301   useContinue     :  607   3rd Qu.:58.0
##  156b7f92ff90:  296                            Max.   :70.0
##  (Other)     :39036
##         mission_id          B1               B2               B3
##  1ec6a2f64de6:    6   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  20e4c403102c:    6   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  6194bc45854e:    6   Median :0.0000   Median :0.0000   Median :0.0000
##  675266cb1956:    6   Mean   :0.0201   Mean   :0.0326   Mean   :0.0238
##  78aaead721c6:    6   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  848b8c19e5a4:    6   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  (Other)     :40927   NA's   :607      NA's   :607      NA's   :607
##     leftover
##  Min.   : 0.00
##  1st Qu.: 1.00
##  Median : 4.00
##  Mean   : 4.87
##  3rd Qu.: 7.00
##  Max.   :31.00
##  NA's   :34451
```

```
n_mission_id = as.integer(GT %>% distinct(mission_id, .keep_all = FALSE) %>% summarise(n=n(
)))
#20320
m_useContinue = as.integer(GT %>% filter(action=="useContinue") %>% distinct(mission_id, .k
eep_all = FALSE)  %>% summarise(n=n()))
#596
```

Matriz que me indica la cantidad de completados o fallos tras el uso de continue. Esto nos ayudará a cotejar el resultado cuando vayamos a hacer el cálculo total

```
m_excess = GT %>% group_by(mission_id) %>%
```

```
  mutate(m_cont=ifelse(action=="useContinue",1,0),
         check=sum(m_cont)) %>%
  filter(check==1) %>% ##de esta forma filtramos los mission_id en los que se haya usado
  group_by(mission_id) %>%
  mutate(m_start=ifelse(action=="missionStarted",1,0),
      m_comp=ifelse(action=="missionCompleted",1,0),
      m_fail=ifelse(action=="missionFailed",1,0)) %>%
  summarise(ST=sum(m_start),
            COM=sum(m_comp),
            FAI=sum(m_fail)) %>%
  mutate(comp_excess=ifelse(COM-ST==0,1,0),
         fail_excess=ifelse(FAI-ST==1,1,0)) %>%
  summarise(comp_excess=sum(comp_excess),fail_excess=sum(fail_excess))

# comp_excess fail_excess
# 431           145
```

#Ejercicio 1: Limpieza de tabla: ***

```
DT_cleaned = GT %>% replace_na(list(B1=0,B2=0,B3=0)) %>% ##Cambiamos los NA (en el futuro e
sto puede ser útil)
            filter_at(vars(c(B1,B2,B3)),all_vars(.==0)) %>%  ###Cumplimos con la condicion
5, se eliminan 3232 registros
            group_by(mission_id) %>%
            mutate(start_counter=ifelse(action=="missionStarted",1,0), ###Cumplimos con la
condicion 4. Se eliminan 5 registros (es posible que provengan de una errata del proveedor?
)
                   n_starts=sum(start_counter)) %>%
            filter(n_starts == 1) %>%
            select(-c(start_counter,n_starts)) %>%
            filter(row_number()==1 | row_number()==2)  ##Cumplimos con la condición 6, basa
ndonos en la condición 3
```

#Ejercicio 2: Distribución de Dificultad por nivel: ***

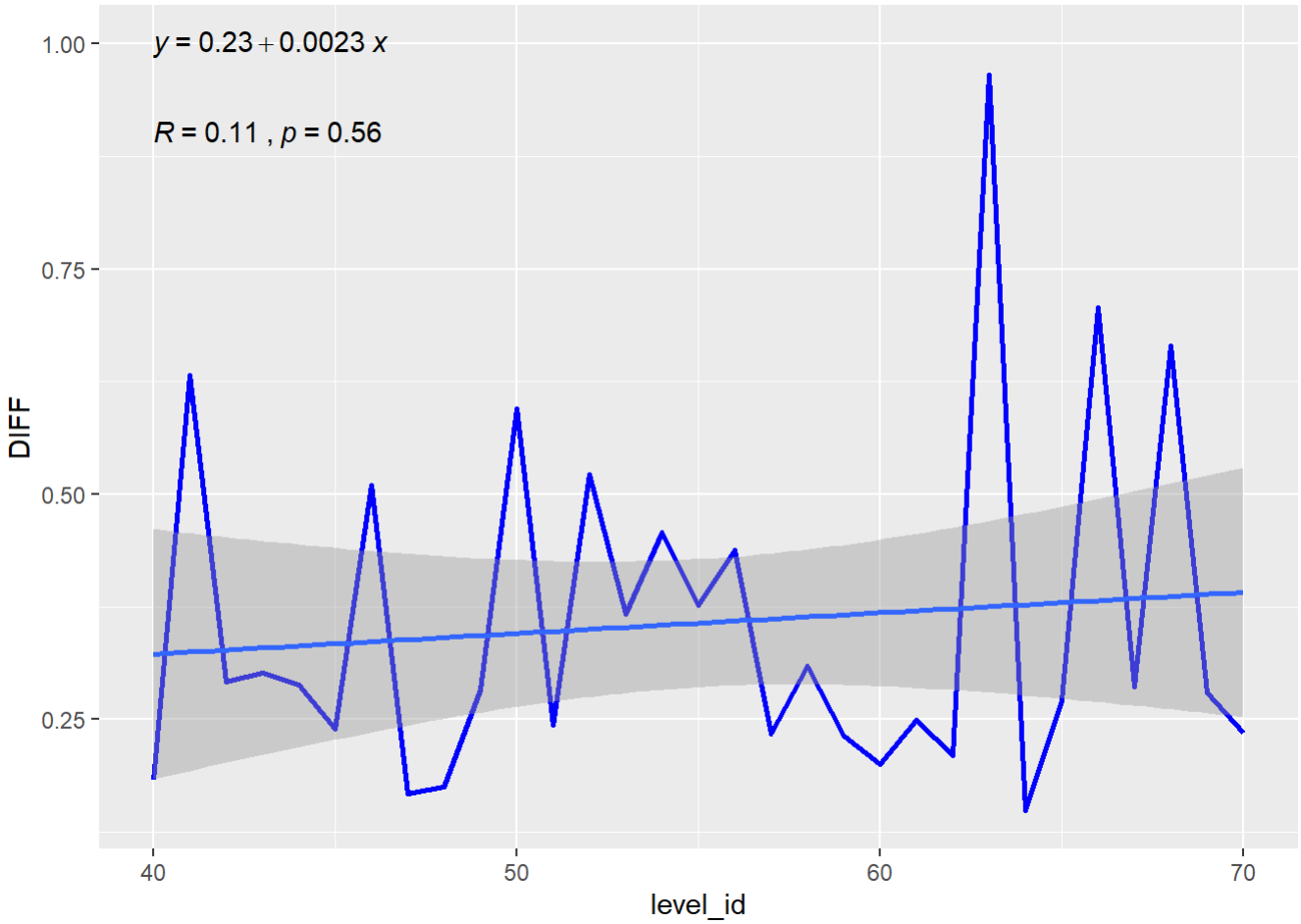
```
Difficulty_wo_boosters = DT_cleaned %>%
  group_by(level_id) %>%
  mutate(fail=ifelse(action=="missionFailed",1,0),
         comp=ifelse(action=="missionCompleted",1,0)) %>%
  summarise(DIFF = sum(comp)/(sum(comp)+sum(fail)))
```

Observamos la evolución de la dificultad según nivel:

```
Difficulty_wo_boosters %>% ggplot(aes(x=level_id,y=DIFF))+
  geom_line(size=1,color="blue",
            add = "reg.line")+

  stat_cor(label.y = 0.9) +
  stat_regline_equation(label.y = 1)+
  geom_smooth(method='lm', formula= y~x)
```

```
## Warning: Ignoring unknown parameters: add
```



Puede constatarse a priori que no sigue una evolución con un sentido claro, excepto que aumenta de media por nivel.

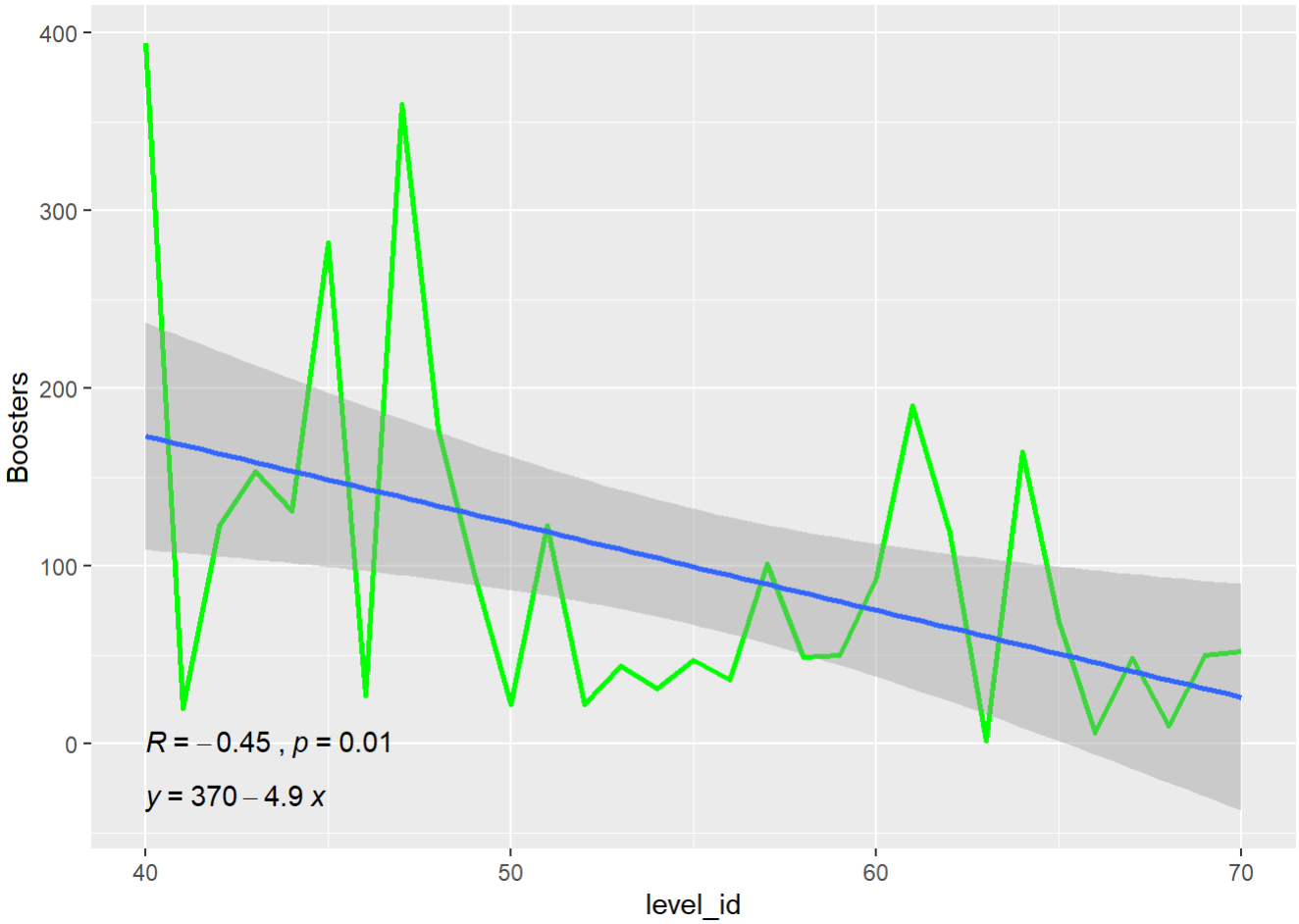
#Ejercicio 3: Uso de ayudas iniciales. Comparativas de partidas con y sin ayuda inicial ***

Visualizamos del uso de boosters:

```
Boosters = GT %>% group_by(level_id) %>%
  replace_na(list(B1=0,B2=0,B3=0)) %>%
  summarise(Boosters=sum(B1)+sum(B2)+sum(B3))
```

```
Boosters %>% ggplot(aes(x=level_id,y=Boosters))+
  geom_line(size=1,color="green",
            add = "reg.line")+
  stat_cor(label.y = 0) +
  stat_regline_equation(label.y = -30)+
  geom_smooth(method='lm', formula= y~x)
```

```
## Warning: Ignoring unknown parameters: add
```

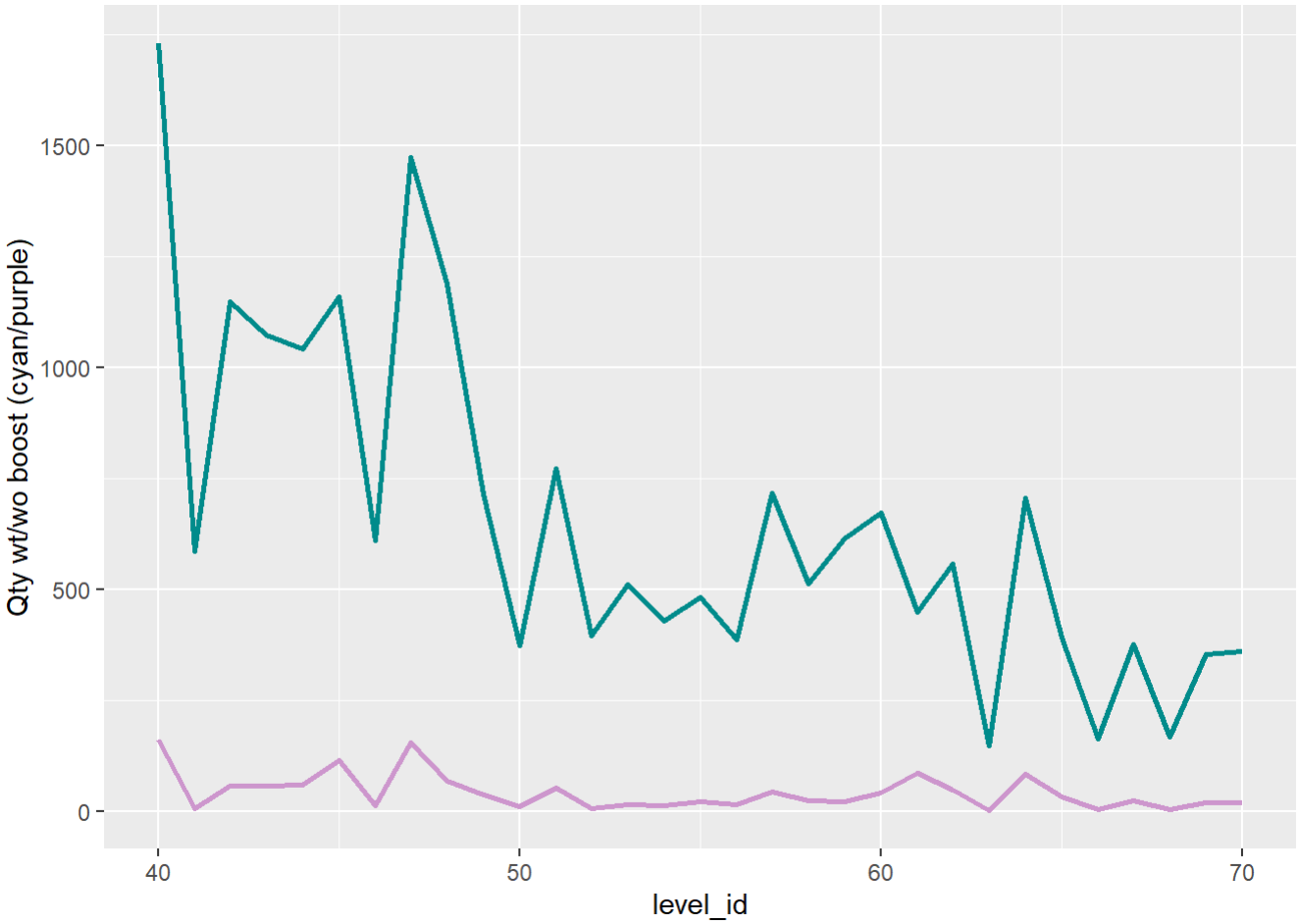


Muchas ayudas al principios, moderadas posteriormente, y en sentido descendente conforme se avanza de nivel

Visualizamos la cantidad de partidas por nivel jugadas con boosters y sin ellos:

```
Comparative_wtwo_Boosters = GT %>%
  replace_na(list(B1=0,B2=0,B3=0)) %>%
  distinct(mission_id, .keep_all = TRUE) %>%
  group_by(level_id) %>%
  mutate(wo_booster=ifelse(B1==0 | B2==0 | B3==0 ,1,0),
         wt_booster=ifelse(B1>0 | B2>0 | B3>0 ,1,0)) %>%
  summarise(wt_booster=sum(wt_booster),
            wo_booster=sum(wo_booster))
```

```
Comparative_wtwo_Boosters %>% ggplot(aes(x=level_id))+
  geom_line(aes(y=wt_booster),size=1,color="plum3")+
  geom_line(aes(y=wo_booster),size=1,color="cyan4")+
  ylab("Qty wt/wo boost (cyan/purple)")
```

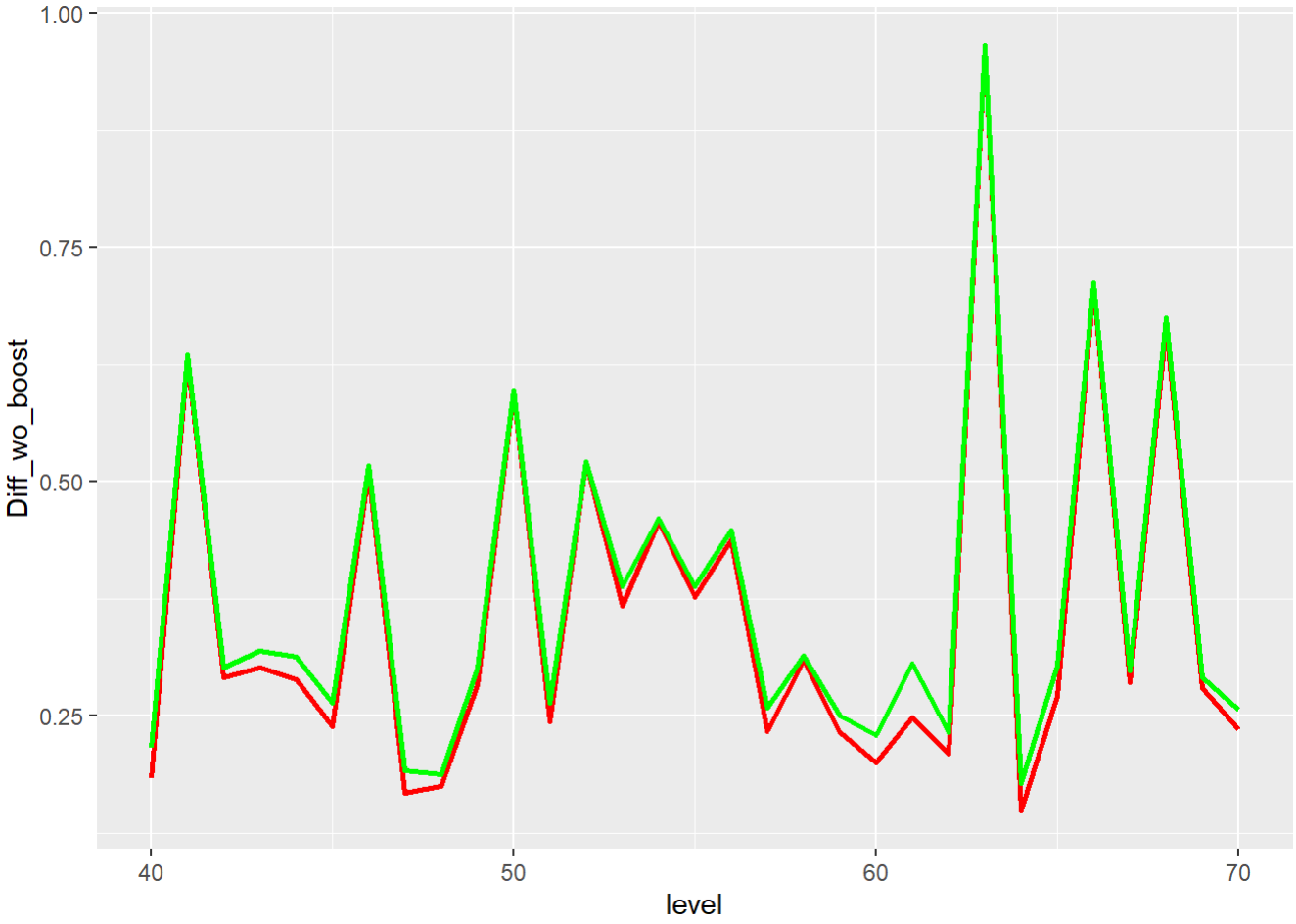


Sólo en los ultimos niveles parecen acercarse el número de usos de éstos y con bastante cercanía

```
Difficulty_wt_boosters = GT %>% replace_na(list(B1=0,B2=0,B3=0)) %>%
  group_by(mission_id) %>%
  mutate(start_counter=ifelse(action=="missionStarted",1,0),
         n_starts=sum(start_counter)) %>%
  filter(n_starts == 1) %>%
  select(-c(start_counter,n_starts)) %>%
  filter(row_number()==1 | row_number()==2) %>%
  group_by(level_id) %>%
  mutate(fail=ifelse(action=="missionFailed",1,0),
         comp=ifelse(action=="missionCompleted",1,0)) %>%
  summarise(DIFF = sum(comp)/(sum(comp)+sum(fail)))

Diff_comparative=cbind(Difficulty_wo_boosters,Difficulty_wt_boosters[-1])
names(Diff_comparative)=c("level","Diff_wo_boost","Diff_wt_boost")
```

```
Diff_comparative %>% ggplot(aes(x=level))+
  geom_line(aes(y=Diff_wo_boost),size=1,color="red")+
  geom_line(aes(y=Diff_wt_boost),size=1,color="green")
```



Y lo curioso es que, la dificultad aumenta ligeramente si se cuentan la resolución de partidas con el uso de Boosters

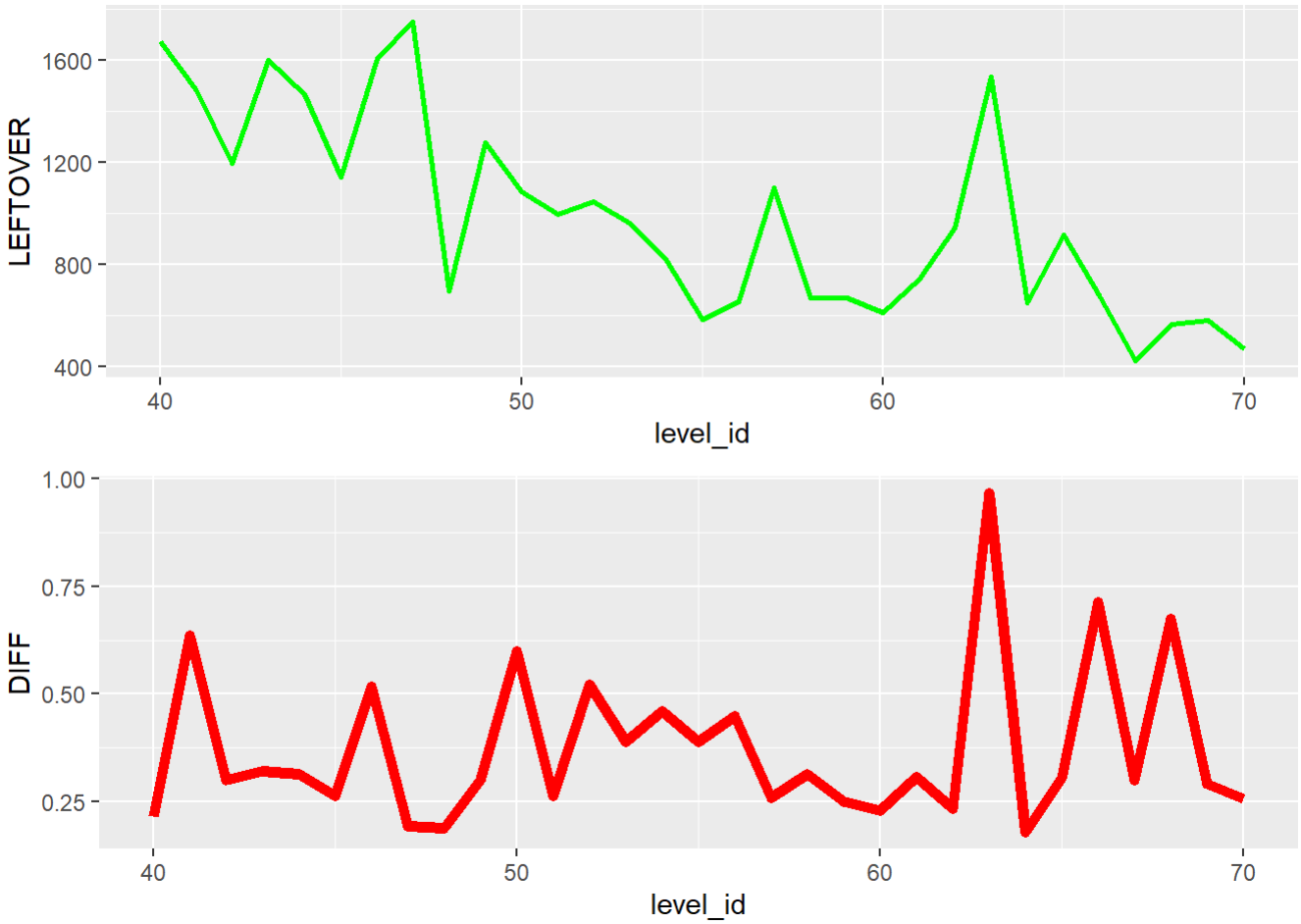
#Ejercicio 4: Comparativa Dificultad vs movimientos sobrantes: ***

Básicamente nos valdremos de la ecuación anterior pero añadiéndole una columna con el recuento de movimientos sobrantes

```
Diff_vs_leftover = GT %>% replace_na(list(B1=0,B2=0,B3=0,leftover=0)) %>%
  group_by(mission_id) %>%
  mutate(start_counter=ifelse(action=="missionStarted",1,0),
         n_starts=sum(start_counter)) %>%
  filter(n_starts == 1) %>%
  select(-c(start_counter,n_starts)) %>%
  filter(row_number()==1 | row_number()==2)%>%
  group_by(level_id) %>%
  mutate(fail=ifelse(action=="missionFailed",1,0),
         comp=ifelse(action=="missionCompleted",1,0)) %>%
  summarise(DIFF = sum(comp)/(sum(comp)+sum(fail)),LEFTOVER=sum(leftover))
```

```
E4.1 = Diff_vs_leftover %>% ggplot(aes(x=level_id))+
  geom_line(aes(y=DIFF),size=2,color="red")
E4.2 = Diff_vs_leftover %>% ggplot(aes(x=level_id))+
  geom_line(aes(y=LEFTOVER),size=1,color="green")

grid.arrange(E4.2,E4.1,nrow=2)
```

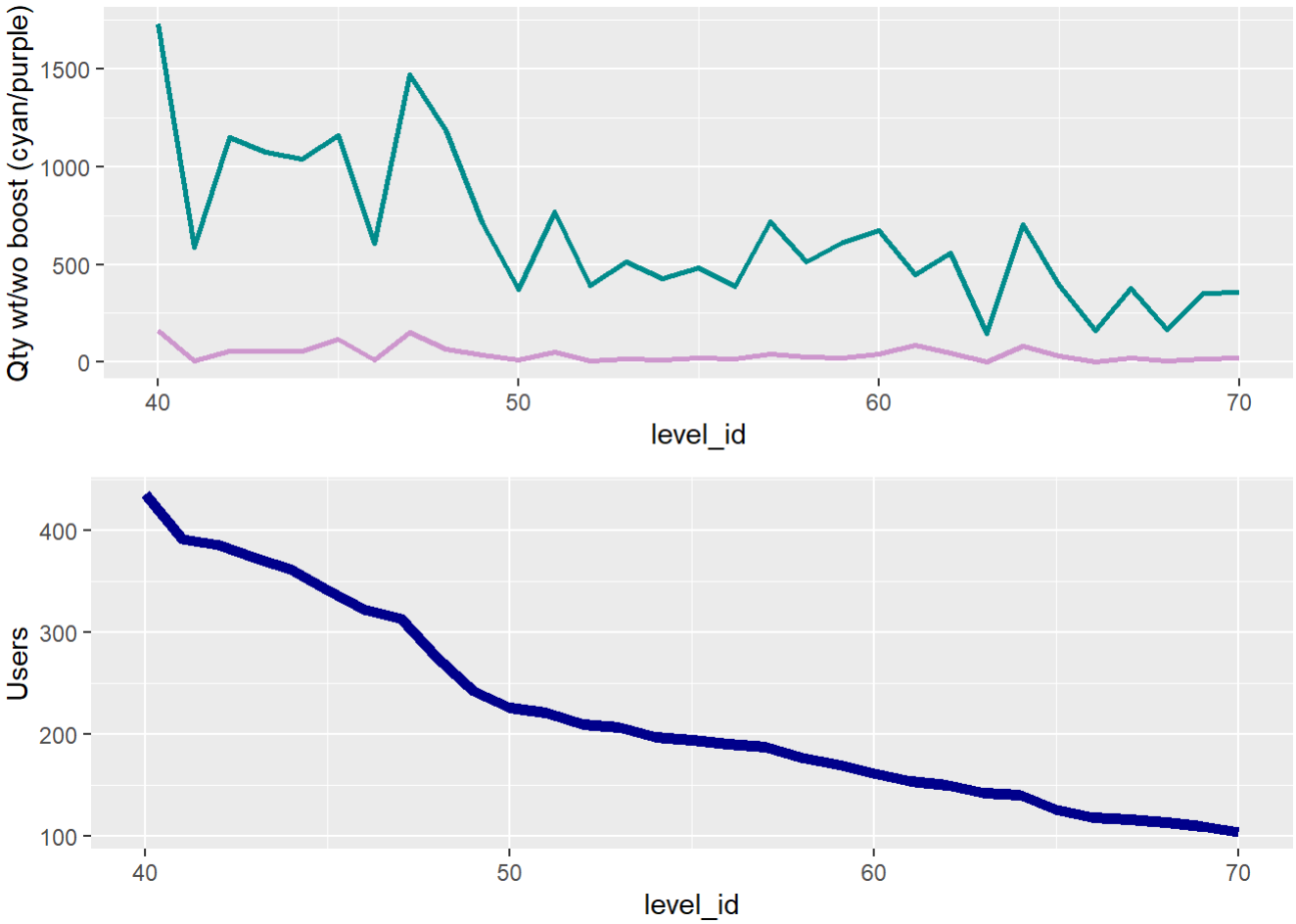


#Ejercicio 5: Conclusiones: ***

- En el gráfico correspondiente a la dificultad vs leftover podemos observar como parece existir una tendencia muy similar en su evolución conforme se aumenta de nivel. Se puede ver claramente como en los picos más significativos (en torno a los niveles 47(positivo), 48.5(negativo), a lo largo de los niveles 50-60 (negativo) y después un gran pico en torno al nivel 63-64).
- Parece que la mayor carga de movimientos sobrantes se da en los niveles de mayor dificultad o previos a éstos, pero en general sigue la misma tendencia que la dificultad, es decir, a mayor dificultad, mayor cantidad de movimientos sobrantes e ídem para la baja dificultad.
- En el caso de uso de Boosters, su uso cae en picado como vemos en la comparativa del gráfico de abajo. Pese a que la tendencia de este es descendente, no es tan pronunciada como lo es la caida de usuarios conforme se avanza. A su lado, parece una tendencia de uso casi uniforme.

```
E5.1 = Comparative_wtwo_Boosters %>% ggplot(aes(x=level_id))+
  geom_line(aes(y=wt_booster),size=1,color="plum3")+
  geom_line(aes(y=wo_booster),size=1,color="cyan4")+
  ylab("Qty wt/wo boost (cyan/purple)")
E5.2 = GT %>% group_by(level_id) %>%
  distinct(user_id) %>%
  summarise(Users=n()) %>%
  ggplot(aes(x=level_id))+
  geom_line(aes(y=Users),size=2,color="darkblue")

grid.arrange(E5.1,E5.2,nrow=2)
```

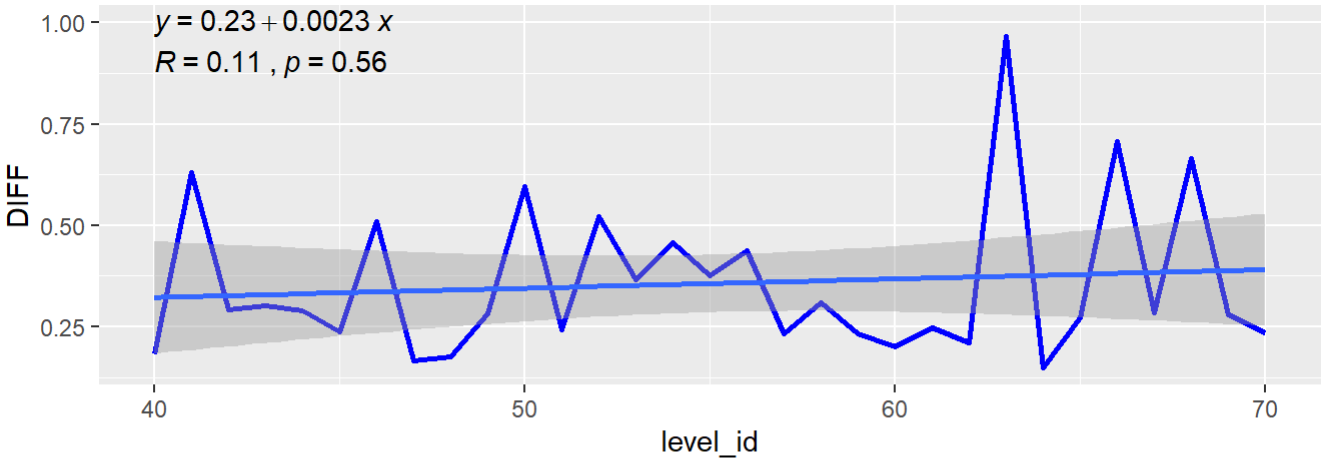


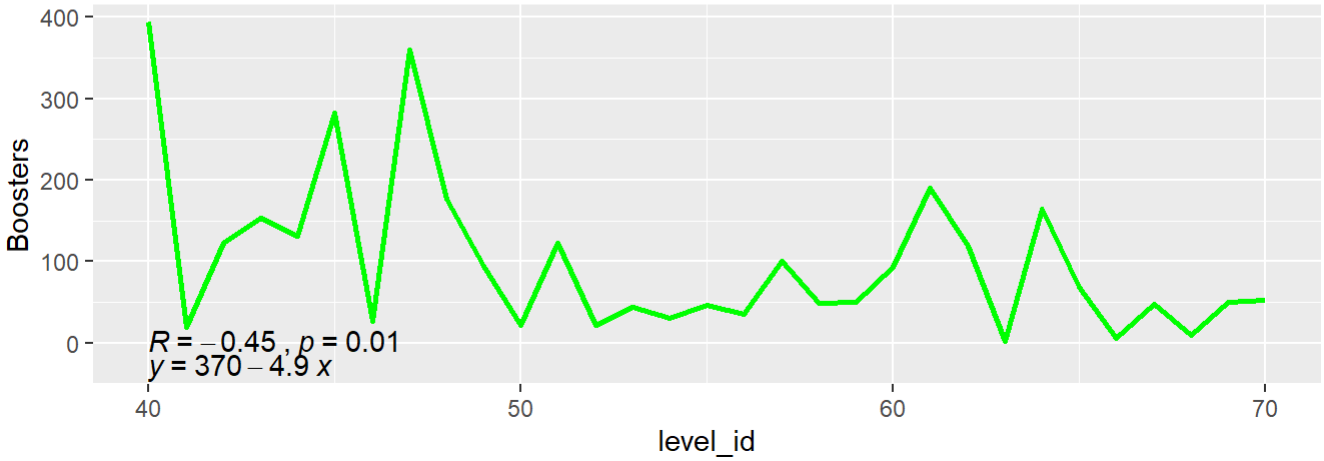
- Sin embargo ahondando un poco más se observa como la zona de niveles de mayor estabilidad evolutiva (~50-60) la caida de usuarios es algo más suave. Se va adquiriendo una mayor resiliencia conforme se sube de nivel.
- Respecto a la dificultad, además de lo observado, hacer hincapié en los niveles donde la tendencia se desregulariza: Niveles 46-50-52-63-66-68. Más abajo podriamos valorar su comportamiento respecto al uso de Boosters:

```
  E5.3 = Difficulty_wo_boosters %>% ggplot(aes(x=level_id,y=DIFF))+
  geom_line(size=1,color="blue")+
  stat_cor(label.y = 0.9) +
  stat_regline_equation(label.y = 1)+
  geom_smooth(method='lm', formula= y~x)

E5.4 = Boosters %>% ggplot(aes(x=level_id,y=Boosters))+
  geom_line(size=1,color="green")+
  stat_cor(label.y = 0) +
  stat_regline_equation(label.y = -30)

grid.arrange(E5.3,E5.4,nrow=2)
```





- Se contempla, finalmente, como el uso de Boosters es *inversamente proporcional* a la dificultad del evento, lo que arroja coherencia a unos datos hasta ahora difíciles de encajar. A mayor uso de boosters > mayor éxito > mayor ratio de de missionCompleted > menor dificultad