Visual analytics of hotel bookings data

Julià Minguillón (Xavier de Moner)

2024-08-12

NOTA: este tutorial usa R + RStudio + ciertas librerías (packages) de R para mostrar el uso de visualizaciones de datos para inspeccionar y analizar un conjunto de datos. Os recomendamos explorar los siguientes enlaces:

1. RStudio: https://posit.co/downloads/ (https://posit.co/downloads/)
2. ggplot2: https://ggplot2.tidyverse.org/ (https://ggplot2.tidyverse.org/)
3. extensiones: https://exts.ggplot2.tidyverse.org/gallery/ (https://exts.ggplot2.tidyverse.org/gallery/)

Cargar packages necesarios

```
library("ggmosaic")
```

```
## Warning: package 'ggmosaic' was built under R version 4.2.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library("ggplot2")
library("fitdistrplus")
```

```
## Loading required package: MASS
```

```
## Loading required package: survival
```

```
## Warning: package 'survival' was built under R version 4.2.3
```

```
library("MASS")
library("survival")
library("ggstatsplot")
```

```
## You can cite this package as:
##      Patil, I. (2021). Visualizations with statistical details: The 'ggstatsplot' approac
h.
##      Journal of Open Source Software, 6(61), 3167, doi:10.21105/joss.03167
```

```
library("tidyverse")
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'tibble' was built under R version 4.2.3
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'purrr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'stringr' was built under R version 4.2.3
```

```
## Warning: package 'forcats' was built under R version 4.2.3
```

```
## Warning: package 'lubridate' was built under R version 4.2.3
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ lubridate 1.9.3     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ dplyr::select() masks MASS::select()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

Data loading and dimensions (N x M)

Leemos el fichero de datos en formato CSV, tiene 119,390 filas y 32 columnas:

```
x=read.csv("hotel_bookings.csv", stringsAsFactors = T)
dim(x)
```

```
## [1] 119390     32
```

Data cleansing

Primero inspeccionaremos los datos usando la función summary() incluída en R. La explicación de cada variable se puede encontrar en el artículo en el cual se describe este conjunto de datos de forma detallada, aunque los nombres de las variables son casi auto-explicativos:

```
##          hotel        is_canceled       lead_time   arrival_date_year
##  City Hotel  :79330   Min.   :0.0000   Min.   :  0   Min.   :2015
##  Resort Hotel:40060   1st Qu.:0.0000   1st Qu.: 18   1st Qu.:2016
##                       Median :0.0000   Median : 69   Median :2016
##                       Mean   :0.3704   Mean   :104   Mean   :2016
##                       3rd Qu.:1.0000   3rd Qu.:160   3rd Qu.:2017
##                       Max.   :1.0000   Max.   :737   Max.   :2017
##
##  arrival_date_month arrival_date_week_number arrival_date_day_of_month
##  August :13877      Min.   : 1.00            Min.   : 1.0
##  July   :12661      1st Qu.:16.00            1st Qu.: 8.0
##  May    :11791      Median :28.00            Median :16.0
##  October:11160      Mean   :27.17            Mean   :15.8
##  April  :11089      3rd Qu.:38.00            3rd Qu.:23.0
##  June   :10939      Max.   :53.00            Max.   :31.0
##  (Other):47873
##  stays_in_weekend_nights stays_in_week_nights     adults
##  Min.   : 0.0000         Min.   : 0.0         Min.   : 0.000
##  1st Qu.: 0.0000         1st Qu.: 1.0         1st Qu.: 2.000
##  Median : 1.0000         Median : 2.0         Median : 2.000
##  Mean   : 0.9276         Mean   : 2.5         Mean   : 1.856
##  3rd Qu.: 2.0000         3rd Qu.: 3.0         3rd Qu.: 2.000
##  Max.   :19.0000         Max.   :50.0         Max.   :55.000
##
##     children           babies                  meal          country
##  Min.   : 0.0000   Min.   : 0.000000   BB       :92310   PRT    :48590
##  1st Qu.: 0.0000   1st Qu.: 0.000000   FB       :  798   GBR    :12129
##  Median : 0.0000   Median : 0.000000   HB       :14463   FRA    :10415
##  Mean   : 0.1039   Mean   : 0.007949   SC       :10650   ESP    : 8568
##  3rd Qu.: 0.0000   3rd Qu.: 0.000000   Undefined: 1169   DEU    : 7287
##  Max.   :10.0000   Max.   :10.000000                     ITA    : 3766
##  NA's   :4                                               (Other):28635
##        market_segment  distribution_channel is_repeated_guest
##  Online TA    :56477   Corporate: 6677      Min.   :0.00000
##  Offline TA/TO:24219   Direct   :14645      1st Qu.:0.00000
##  Groups       :19811   GDS      :  193      Median :0.00000
##  Direct       :12606   TA/TO    :97870      Mean   :0.03191
##  Corporate    : 5295   Undefined:    5      3rd Qu.:0.00000
##  Complementary:  743                        Max.   :1.00000
##  (Other)      :  239
##  previous_cancellations previous_bookings_not_canceled reserved_room_type
##  Min.   : 0.00000       Min.   : 0.0000               A      :85994
##  1st Qu.: 0.00000       1st Qu.: 0.0000               D      :19201
##  Median : 0.00000       Median : 0.0000               E      : 6535
##  Mean   : 0.08712       Mean   : 0.1371               F      : 2897
##  3rd Qu.: 0.00000       3rd Qu.: 0.0000               G      : 2094
##  Max.   :26.00000       Max.   :72.0000               B      : 1118
##                                                       (Other): 1551
##  assigned_room_type booking_changes       deposit_type        agent
##  A      :74053      Min.   : 0.0000   No Deposit:104641   9      :31961
##  D      :25322      1st Qu.: 0.0000   Non Refund: 14587   NULL   :16340
##  E      : 7806      Median : 0.0000   Refundable:   162   240    :13922
##  F      : 3751      Mean   : 0.2211                       1      : 7191
##  G      : 2553      3rd Qu.: 0.0000                       14     : 3640
##  C      : 2375      Max.   :21.0000                       7      : 3539
```

```
##  (Other): 3530                                          (Other):42797
##     company       days_in_waiting_list         customer_type
##  NULL   :112593   Min.   :  0.000      Contract       : 4076
##  40     :   927   1st Qu.:  0.000      Group          :  577
##  223    :   784   Median :  0.000      Transient      :89613
##  67     :   267   Mean   :  2.321      Transient-Party:25124
##  45     :   250   3rd Qu.:  0.000
##  153    :   215   Max.   :391.000
##  (Other):  4354
##       adr          required_car_parking_spaces total_of_special_requests
##  Min.   :  -6.38   Min.   :0.00000             Min.   :0.0000
##  1st Qu.:  69.29   1st Qu.:0.00000             1st Qu.:0.0000
##  Median :  94.58   Median :0.00000             Median :0.0000
##  Mean   : 101.83   Mean   :0.06252             Mean   :0.5714
##  3rd Qu.: 126.00   3rd Qu.:0.00000             3rd Qu.:1.0000
##  Max.   :5400.00   Max.   :8.00000             Max.   :5.0000
##
##  reservation_status reservation_status_date
##  Canceled :43017    2015-10-21:  1461
##  Check-Out:75166    2015-07-06:   805
##  No-Show  : 1207    2016-11-25:   790
##                     2015-01-01:   763
##                     2016-01-18:   625
##                     2015-07-02:   469
##                     (Other)   :114477
```

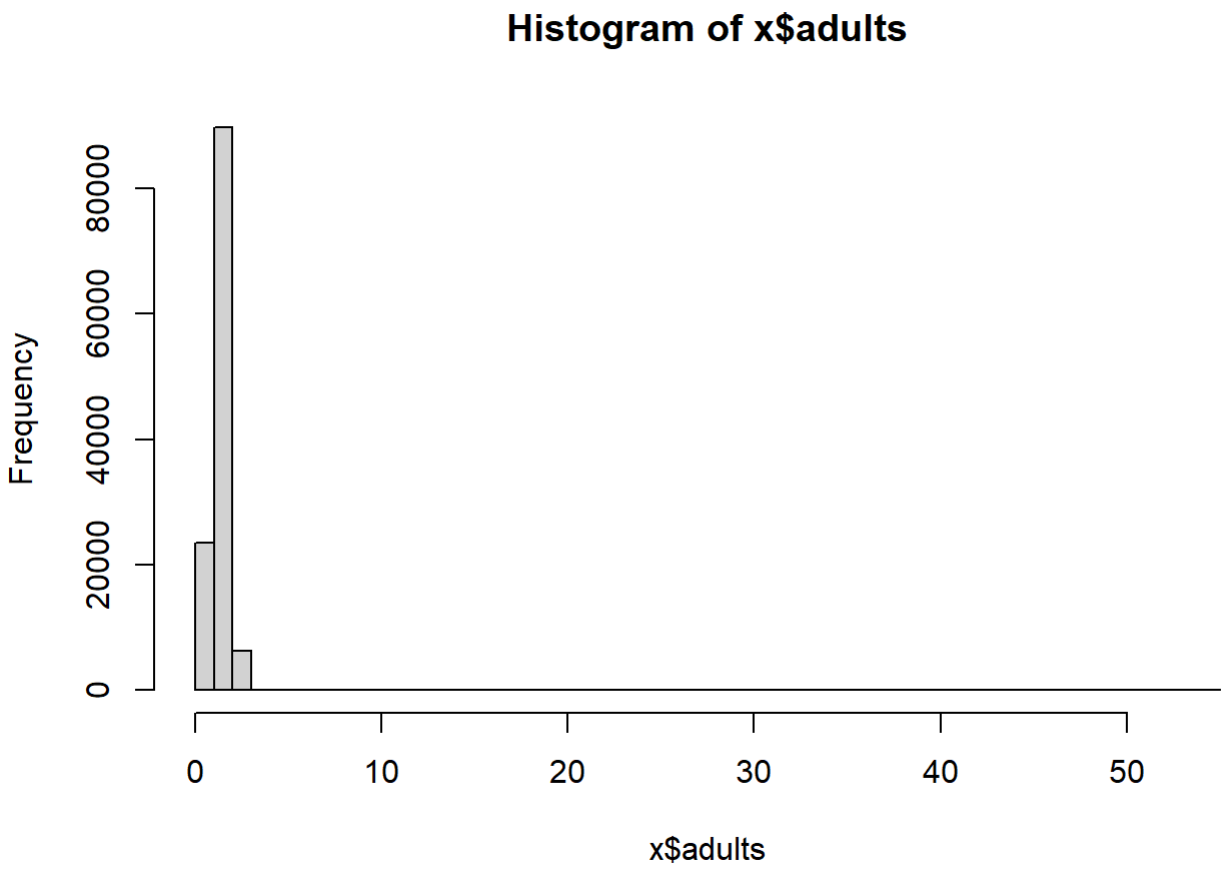
Variables numéricas

Podemos observar algunos valores extraños para algunas variables, por ejemplo:

1. Un máximo de 55 en adults
2. Un máximo de 10 en children (incluyendo valores perdidos)
3. Un máximo de 10 en babies
4. Valores negativos en el coste promedio por día (adr) o muy elevados

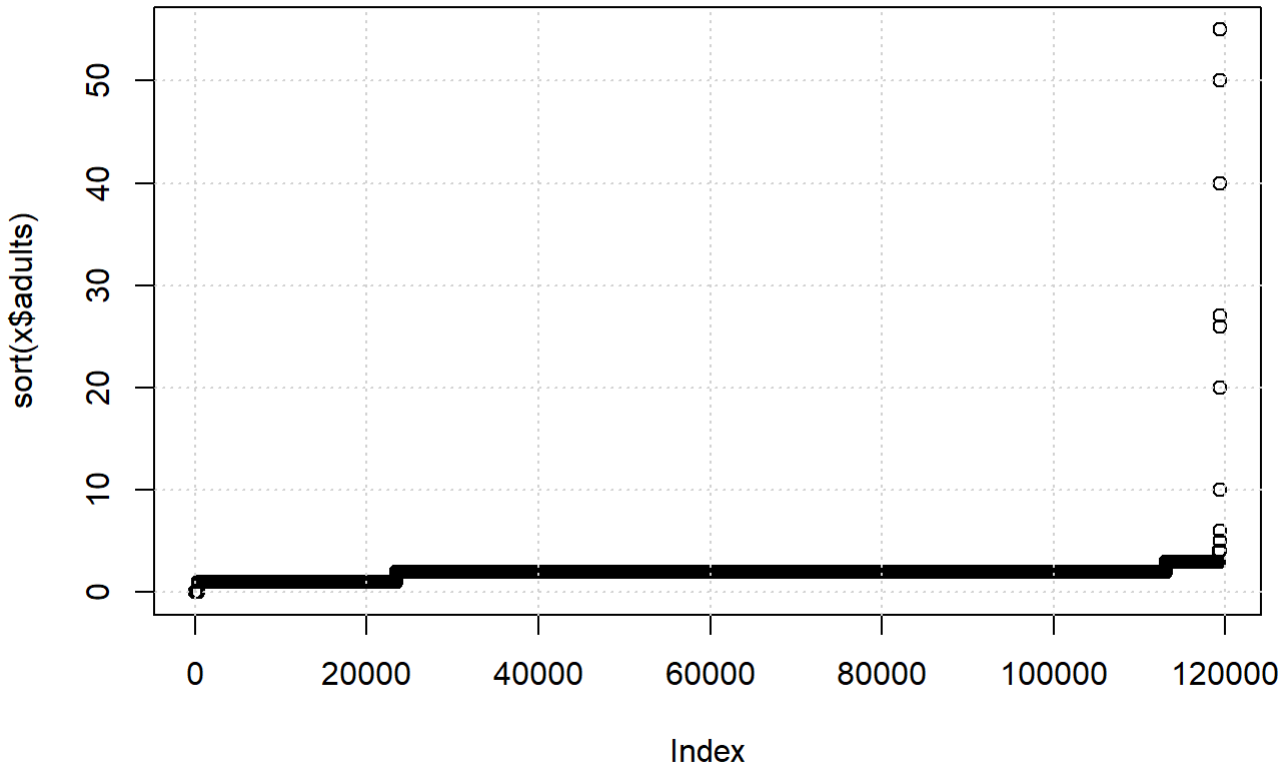
Vamos a visualizar el histograma de la variable adults, indicando al menos 55 intervalos en el histograma, usando la función hist() de R:

```
hist(x$adults,breaks=55)
```



Se puede ver que el histograma no muestra ninguna barra alrededor del 55, dado que se trata de un conjunto muy grande y seguramente se tratará solamente de un caso o pocos. En estos casos, para analizar valores extremos de una variable se pueden pintar los valores de la variable en cuestión de la siguiente manera, ordenando los datos (si son numéricos como en este caso):

```
plot(sort(x$adults))
grid()
```



La variable Index es la posición del elemento una vez ordenado, pero nos interesa más el eje Y, ya que podemos ver que hay unos pocos elementos con valores de 10 o superior. Como se trata de una variable entera pero con un conjunto limitado de valores posibles podemos usar table() para verlos:

```
table(x$adults)
```

```
##
##     0     1     2     3     4     5     6    10    20    26    27    40    50
##   403 23027 89680  6202    62     2     1     1     2     5     2     1     1
##    55
##     1
```

Como se puede ver, hay un caso de una reserva con 10 adultos, dos con 20 adultos, etc., hasta una de 55 adultos! Sin entrar en más consideraciones, eliminaremos todas las filas con reservas de 10 adultos o más:

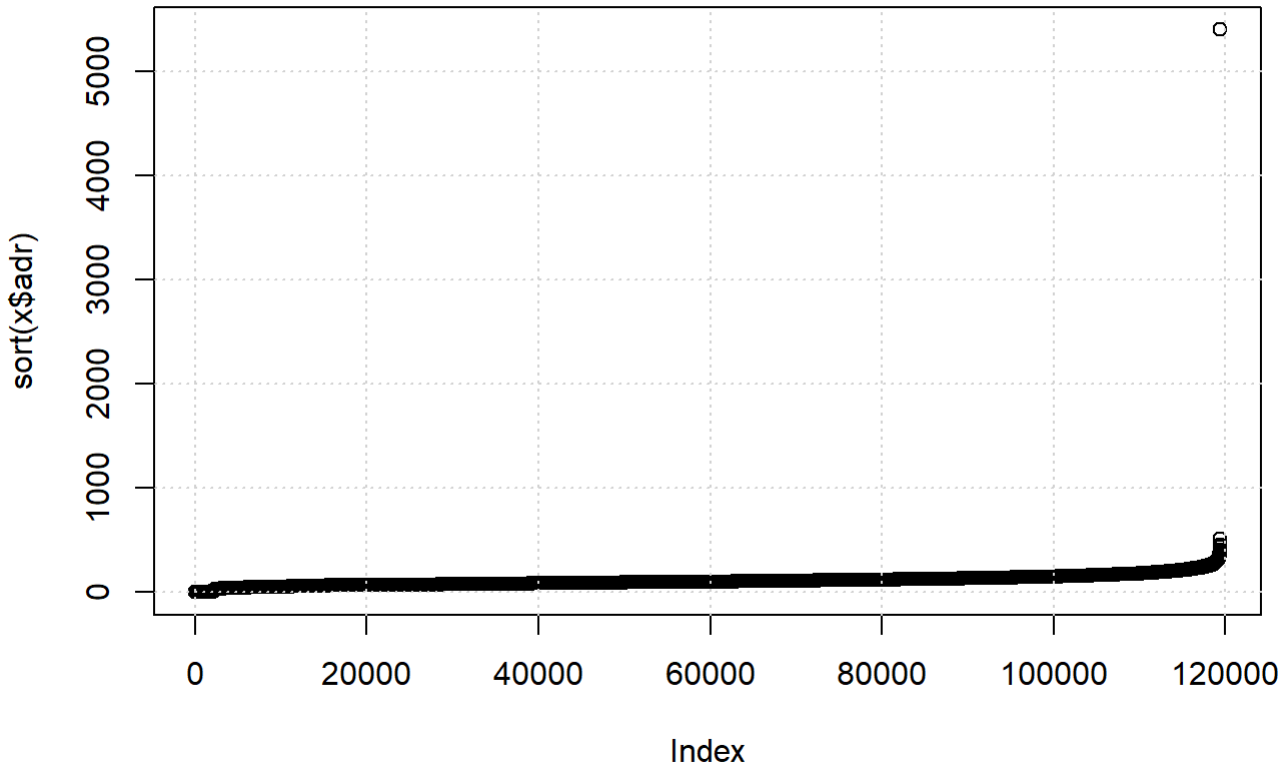
```
x=x[x$adults<10,]
```

EJERCICIO: hacer lo mismo con las variables children y babies

```
# Treiem els outliers de children i babies
x <- x[x$children < 10 & x$babies < 10, ]
```

El histograma de la variable adr (gasto medio por dí) presenta el mismo problema que el caso de la variable adults, así que directamente haremos un gráfico con los valores ordenados:

```
plot(sort(x$adr))
grid()
```



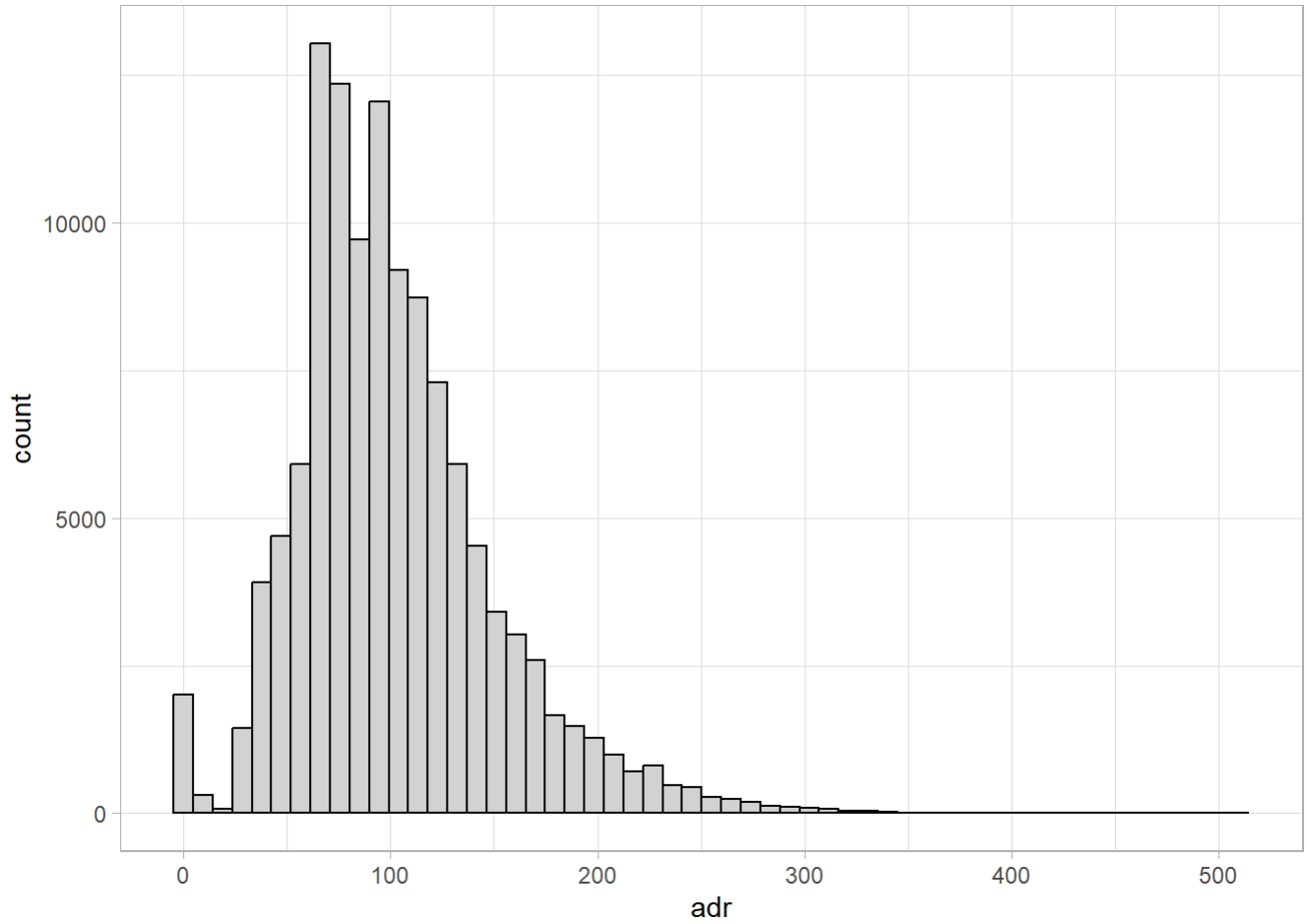
En este caso se ve que hay solamente un valor muy por encima del resto, lo consideramos un outlier y lo eliminamos, así como los valores negativos que no tienen una explicación clara, aunque dejamos los valores 0:

```
x=x[x$adr>=0 & x$adr<1000,]
```

El histograma ahora sí que nos aporta información relevante. Lo dibujamos usando el package ggplot2 que ofrece muchas más opciones que hist():

```
ggplot(data=x, aes(x=adr)) +
  geom_histogram(bins=55, colour="black", fill = "lightgray") +
  theme_light()
```

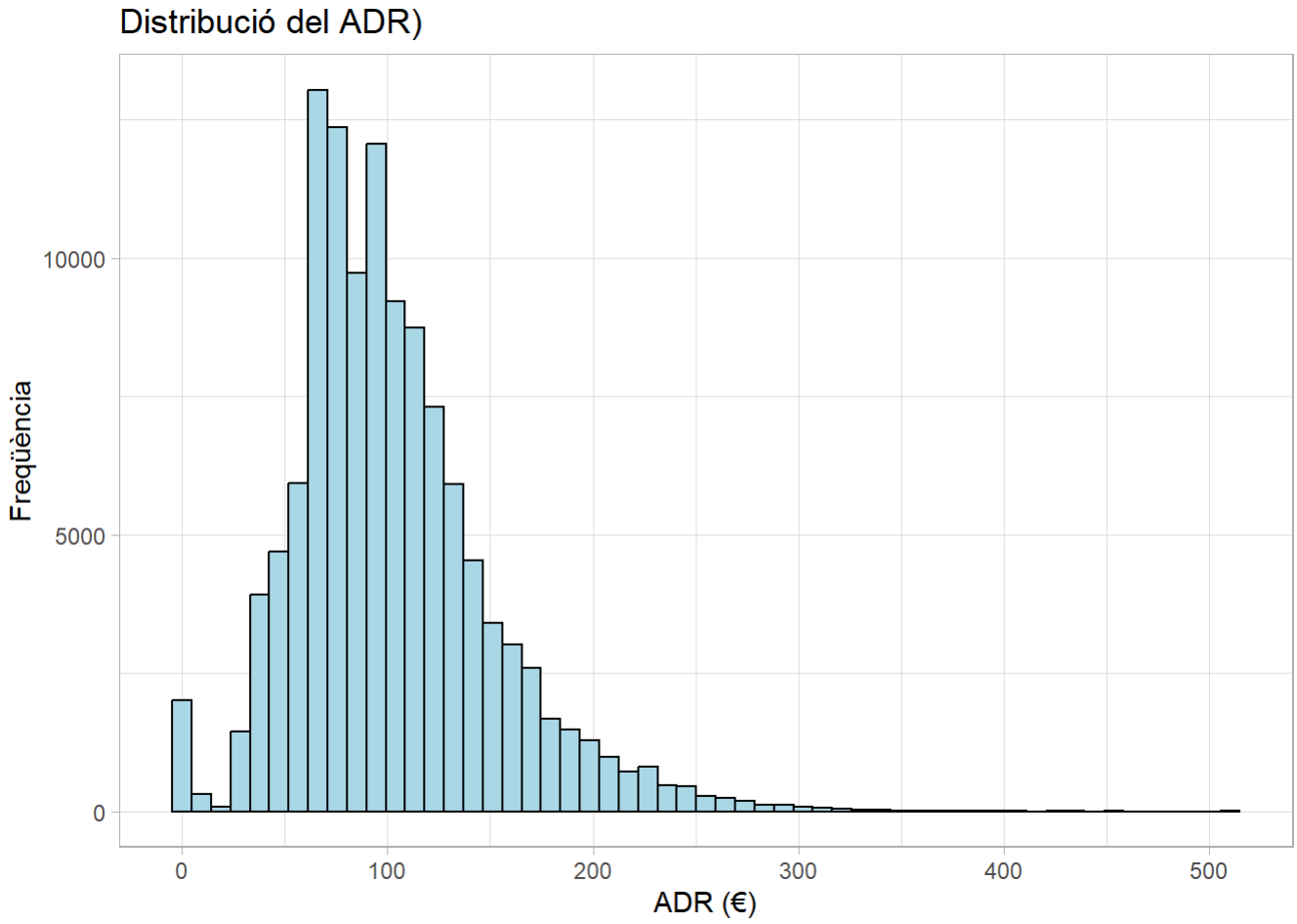
```
## Warning: Removed 4 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



EJERCICIO: retocar el gráfico para que el nombre de los ejes, título, etc. sea el adecuado para una presentación

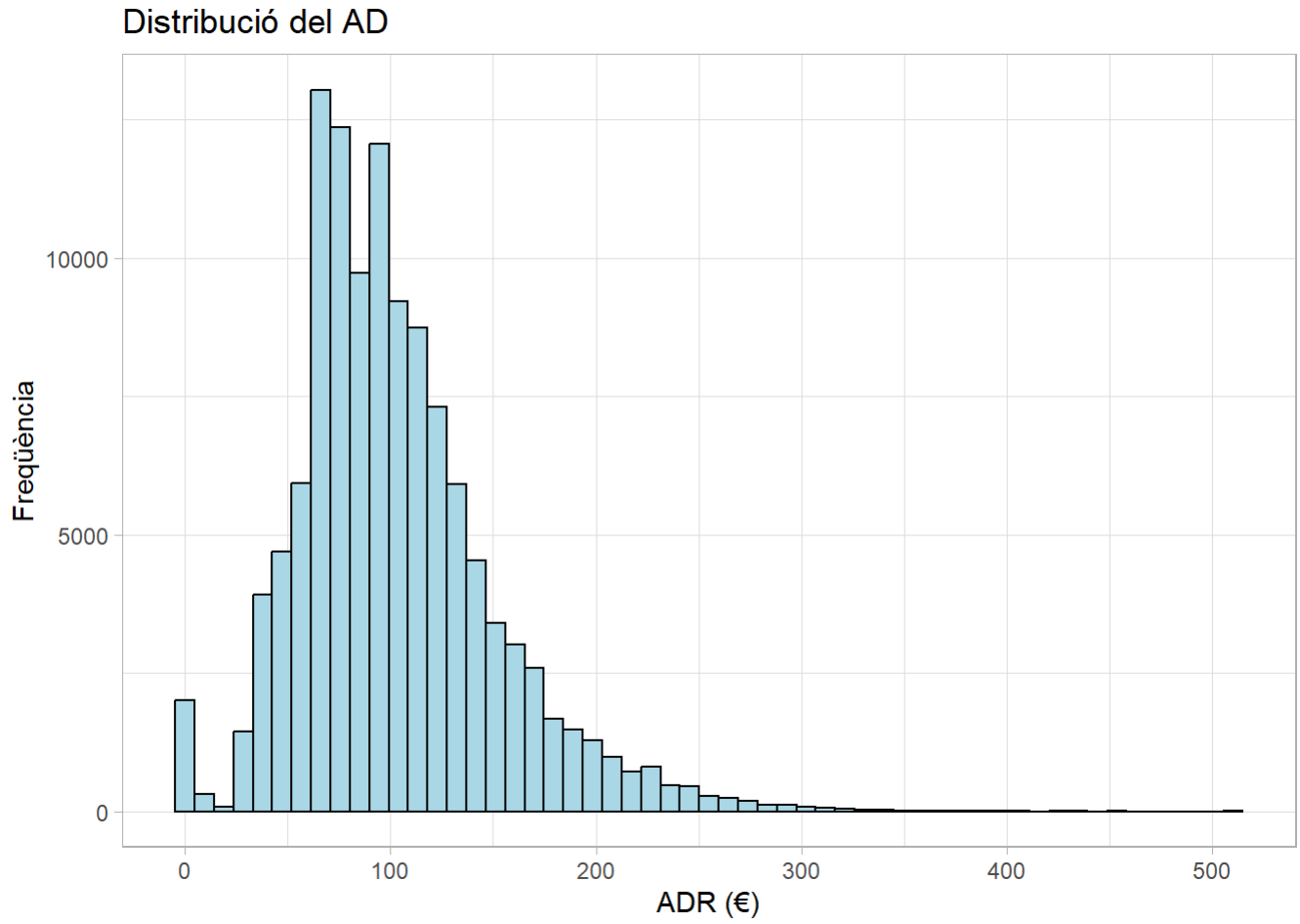
```
# Afegim títols, noms d'eixos
ggplot(data=x, aes(x=adr)) +
  geom_histogram(bins=55, colour="black", fill="lightblue") +
  labs(title="Distribució del ADR)",
       x="ADR (€)", y="Freqüència") +
  theme_light()
```

```
## Warning: Removed 4 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(data=x, aes(x=adr)) +
  geom_histogram(bins=55, colour="black", fill="lightblue") +
  labs(title="Distribució del AD",
       x="ADR (€)", y="Freqüència") +
  theme_light()
```

```
## Warning: Removed 4 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

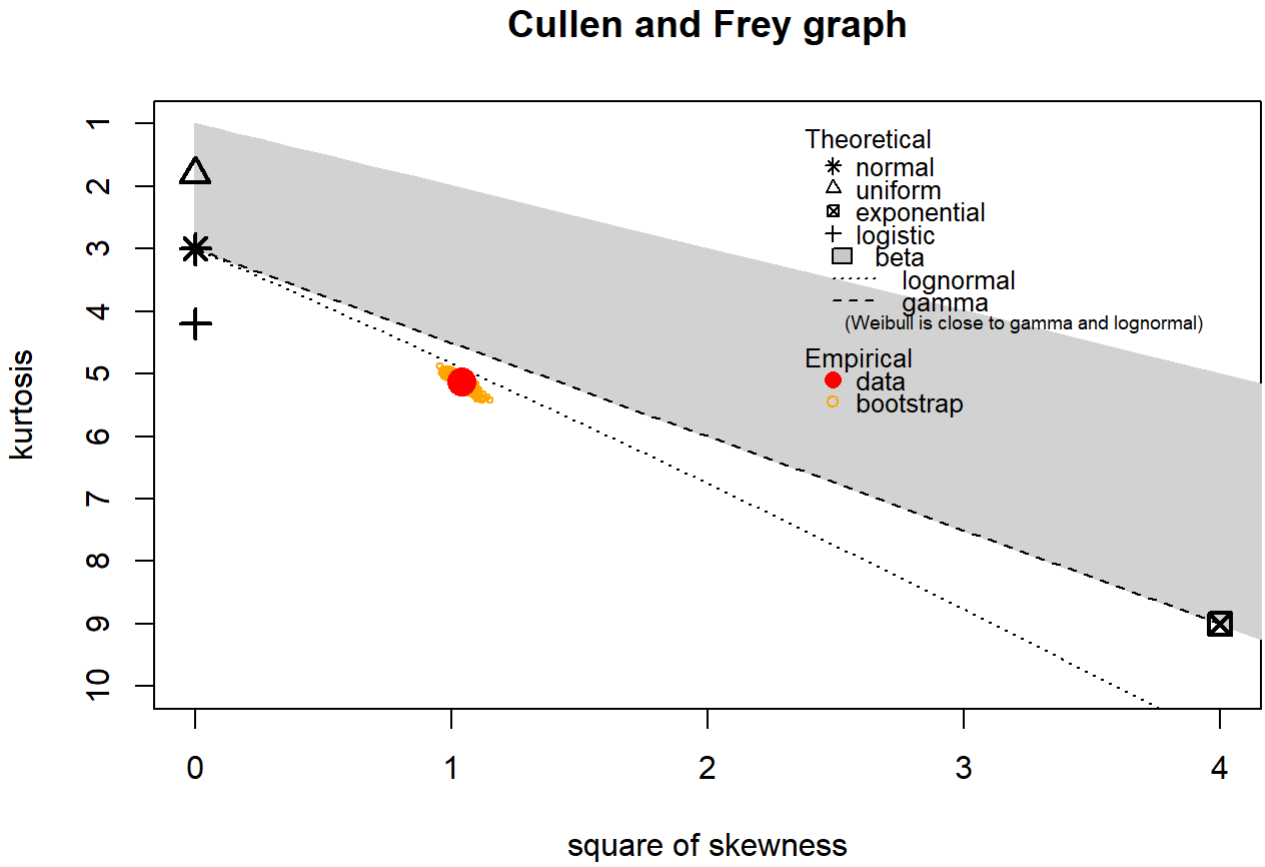


Podemos ver que hay un conjunto de unos 2000 valores 0, los cuales podrían ser analizados de forma separada, por ejemplo. Existen packages de R que nos pueden ayudar a estimar dicha distribución y los parámetros que la determinan de forma visual, como por ejemplo el package fitdistrplus mediante la función descdist():

```
# algun NA quedava
x <- x[!is.na(x$adr) & !is.nan(x$adr) & !is.infinite(x$adr), ]
```

`

```
require(fitdistrplus)
descdist(x$adr,boot=1000)
```




```
## summary statistics
## ------
## min:  0   max:  510
## median:  94.62
## mean:  101.801
## estimated sd:  48.14267
## estimated skewness:  1.018978
## estimated kurtosis:  5.133366
```

Como se puede observar, los datos reales (observación, en azul) y los simulados (en amarillo) están cerca de lo que podría ser una distribución lognormal.

De todas formas, con el objetivo de experimentar con un conjunto de datos lo más limpio posible vamos a proceder a:

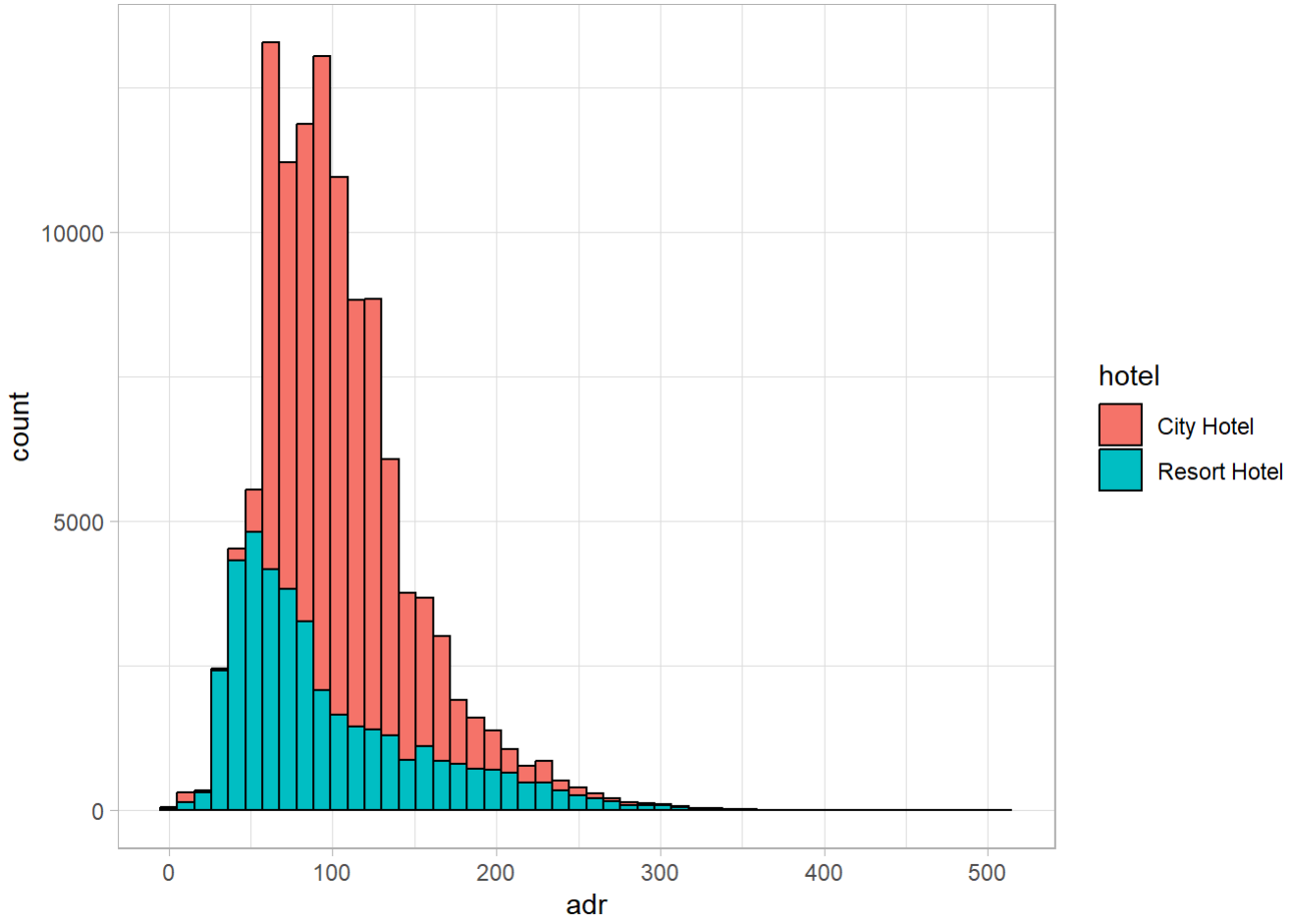
1. eliminar las estancias de 0 días
2. eliminar las estancias a coste 0
3. eliminar las estancias sin personas
4. substituir los NA de la variable children por 0

```
x[is.na(x$children),'children']=0
x=x[x$adr>0 & (x$stays_in_week_nights+x$stays_in_weekend_nights)>0 & (x$adults+x$children+x$babies)>0 & !is.na(x$children),]
```

Variables categóricas

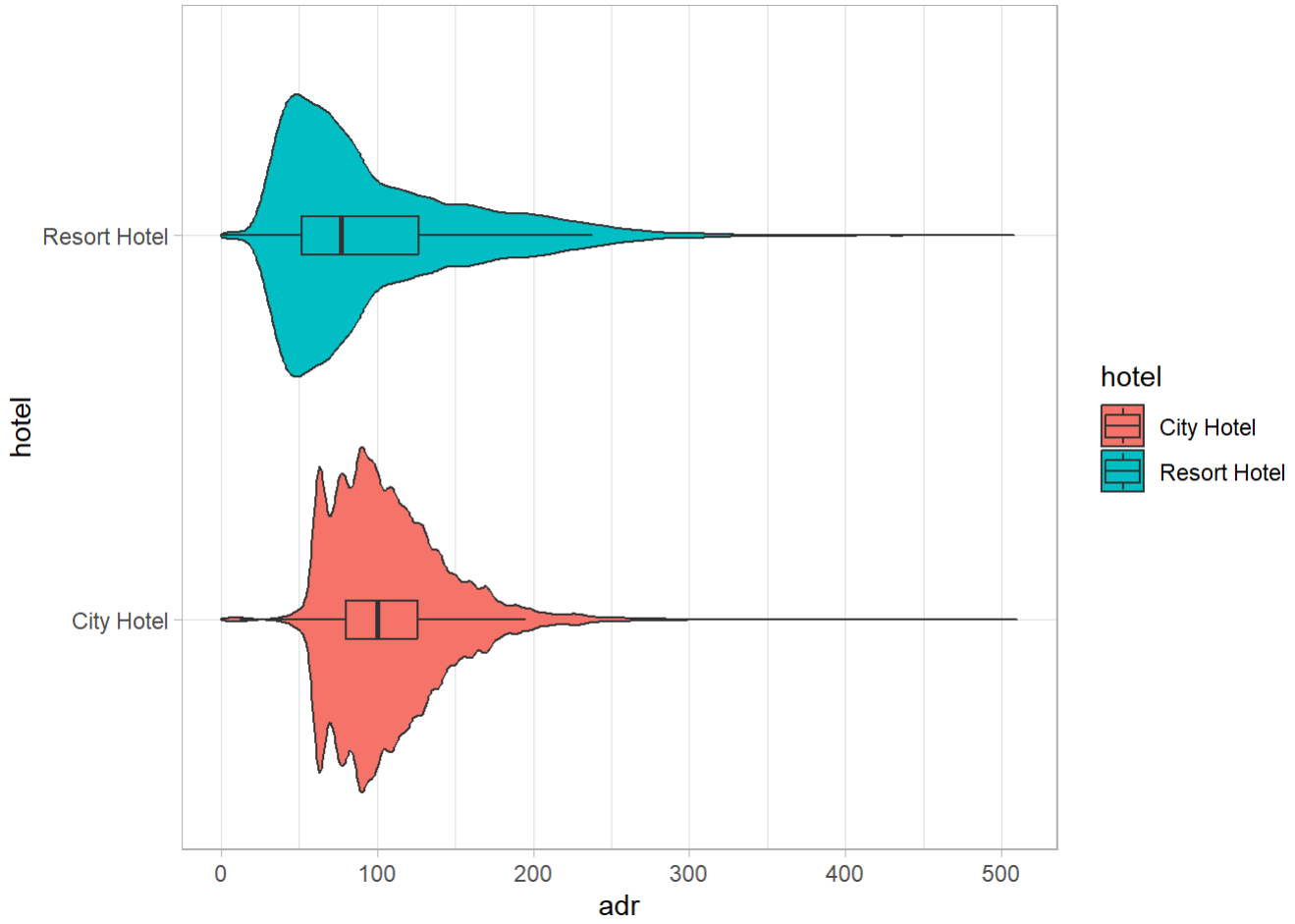
Por lo que respecta a las variables categóricas, la función summary() ya nos da una primera idea de los valores posibles que puede coger cada una. Por ejemplo, en el conjunto original (antes de eliminar outliers) hay 79,330 reservas en un hotel de ciudad (Lisboa) y 40,060 en un resort (el Algarve). Podemos preguntarnos si la distribución del coste es la misma para ambos grupos, ya sea mediante el test estadístico adecuado o simplemente comparando histogramas, en este caso usando el package ggplot2 mucho más potente para crear gráficos de todo tipo:

```
# require(ggplot2)
ggplot(data=x, aes(x=adr, fill=hotel)) +
  geom_histogram(bins=50, colour="black") +
  theme_light()
```



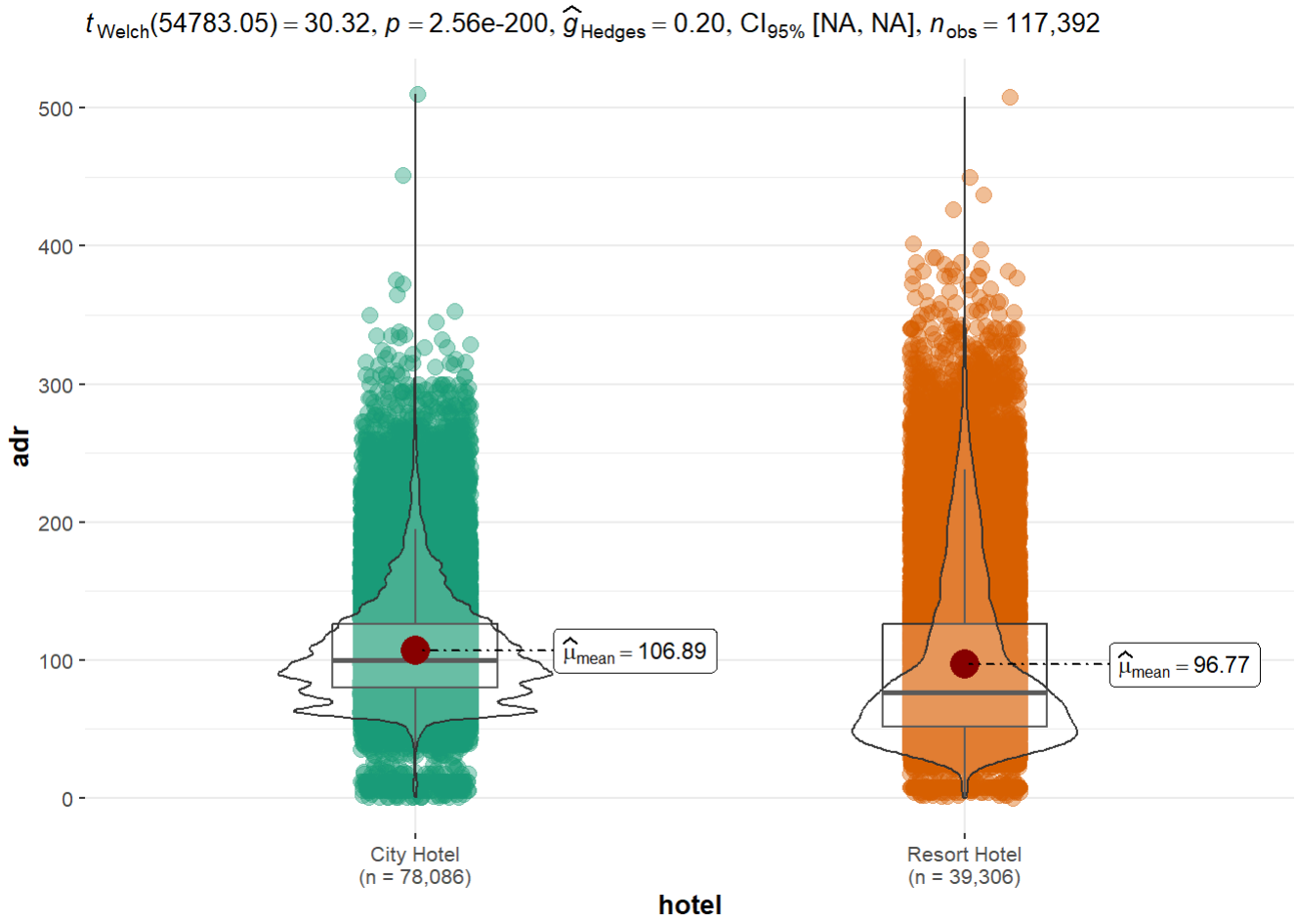
Se puede observar que los precios en Lisboa (City Hotel) más típicos están ligeramente a la derecha de los más típicos en el Algarve (Resort Hotel), aunque en cambio los precios más altos en Lisboa decrecen más rápido que en el Algarve. Con un plot de tipo violin podremos ver más detalle, especialmente si también mostramos los cuartiles típicos de un box-plot:

```
ggplot(data=x, aes(x=hotel, y=adr, fill=hotel)) +
  geom_violin() + geom_boxplot(width=.1, outliers = F) +
  coord_flip() +
  theme_light()
```



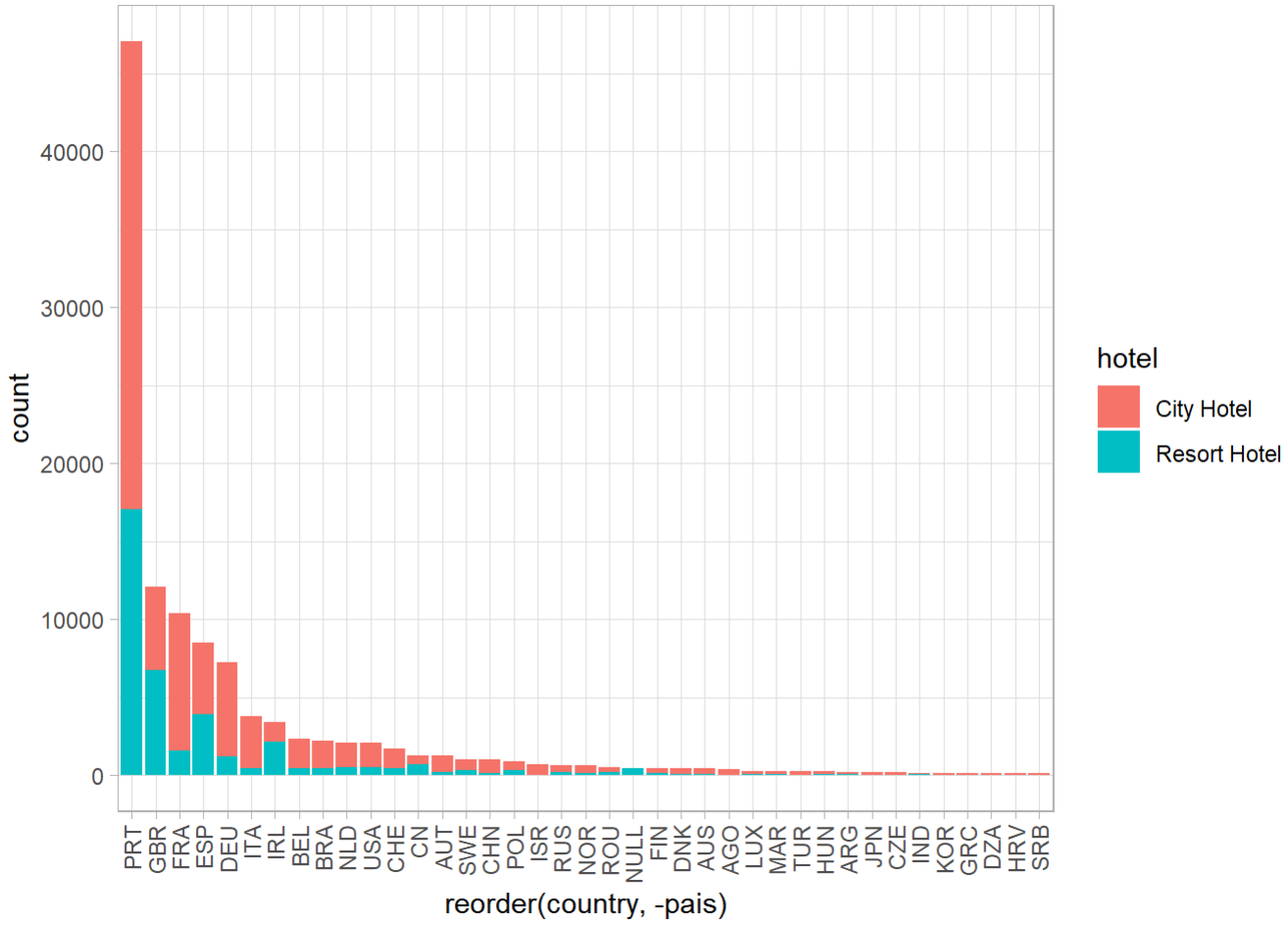
Existe un package de R llamado ggstatsplot que dispone de funciones específicas para cada tipo de gráfico, incluyendo también los tests estadísticos adecuados para establecer si existen diferencias entre grupos:

```
# require(ggstatsplot)
ggbetweenstats(data=x, x=hotel, y=adr)
```



Una variable interesante es la procedencia de los clientes del hotel (country). El problema es que es una variable con muchos valores diferentes (178), por lo que debemos quedarnos con los paises que aportan más turistas, mostrando también si escogen hotel de ciudad o resort:

```
# require(tidyverse)
# paises con al menos 100 reservas
xx = x %>% group_by(country) %>% mutate(pais=n()) %>% filter(pais>=100)
xx$country=factor(xx$country)
ggplot(data=xx, aes(x=reorder(country, -pais))) +
  geom_bar(stat="count", aes(fill=hotel)) +
  theme_light() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



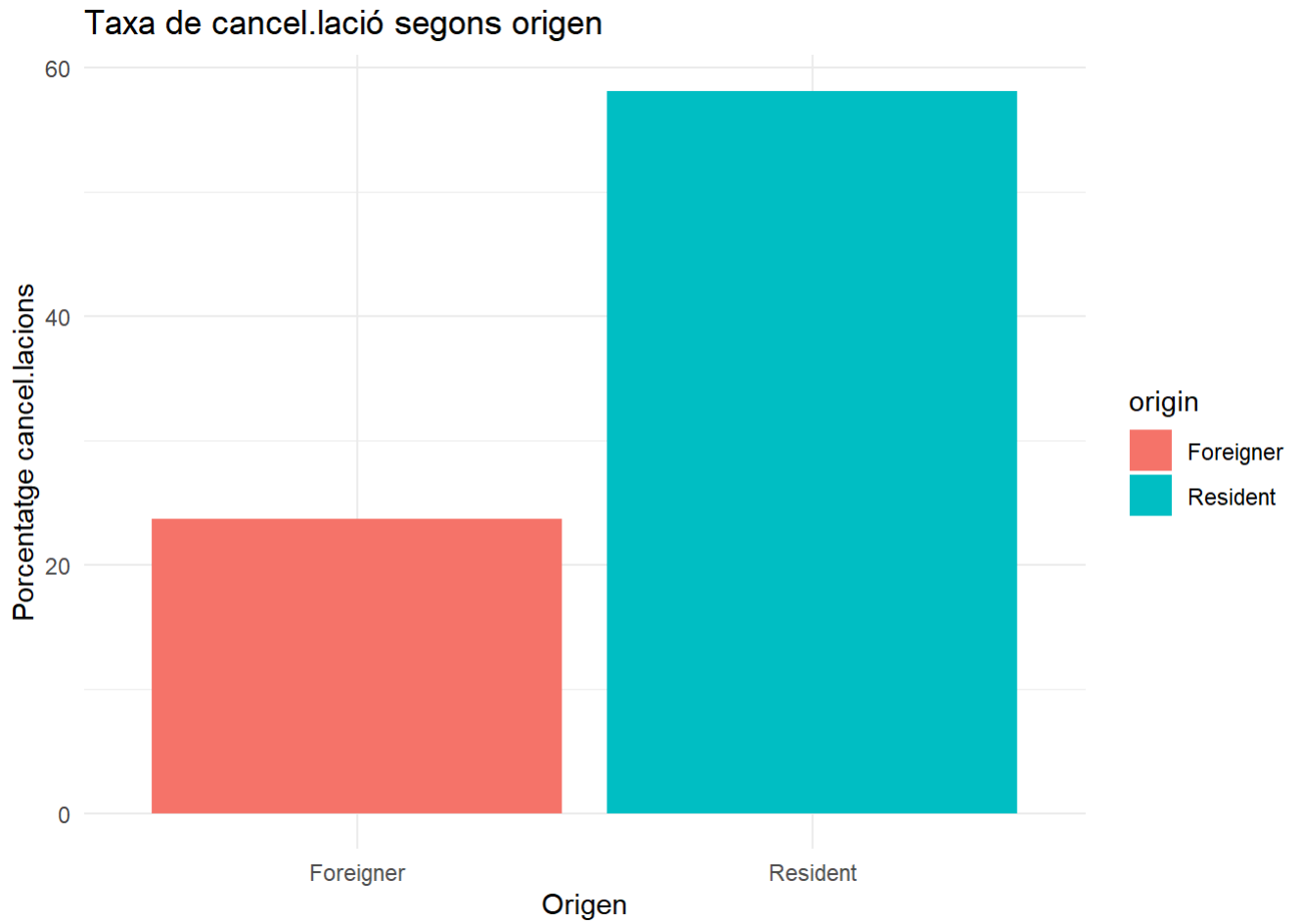
Obviamente, Portugal (PRT) ocupa la primera posición destacada, seguida de paises “cercanos”, como Gran Bretaña, Francia y España. Los visitantes de Gran Bretaña e Irlanda optan más por un resort, mientras que los de Francia, Alemania e Italia principalmente visitan la ciudad de Lisboa.

EJERCICIO: existen diferencias entre los habitantes de Portugal (del país) y el resto (“extranjeros”)?

```
# Etiquetem reserves segons país d'origen (Portual o altre)
x$origin <- ifelse(x$country == "PRT", "Resident", "Foreigner")

# Comparem la taxa (percentatge) de cancel.lació
library(dplyr)
cancel_rate <- x %>%
  group_by(origin) %>%
  summarise(
    cancellations = mean(as.numeric(is_canceled)) * 100,
    count = n()
  )
print(cancel_rate)
```

```
## # A tibble: 2 × 3
##   origin    cancellations count
##   <chr>             <dbl> <int>
## 1 Foreigner          23.7 70372
## 2 Resident           58.1 47020
```

```
# Per últim, mostrem els reultats en gràfic de barres
ggplot(cancel_rate, aes(x = origin, y = cancellations, fill = origin)) +
  geom_bar(stat = "identity") +
  labs(title = "Taxa de cancel.lació segons origen",
       x = "Origen", y = "Porcentatge cancel.lacions") +
  theme_minimal()
```

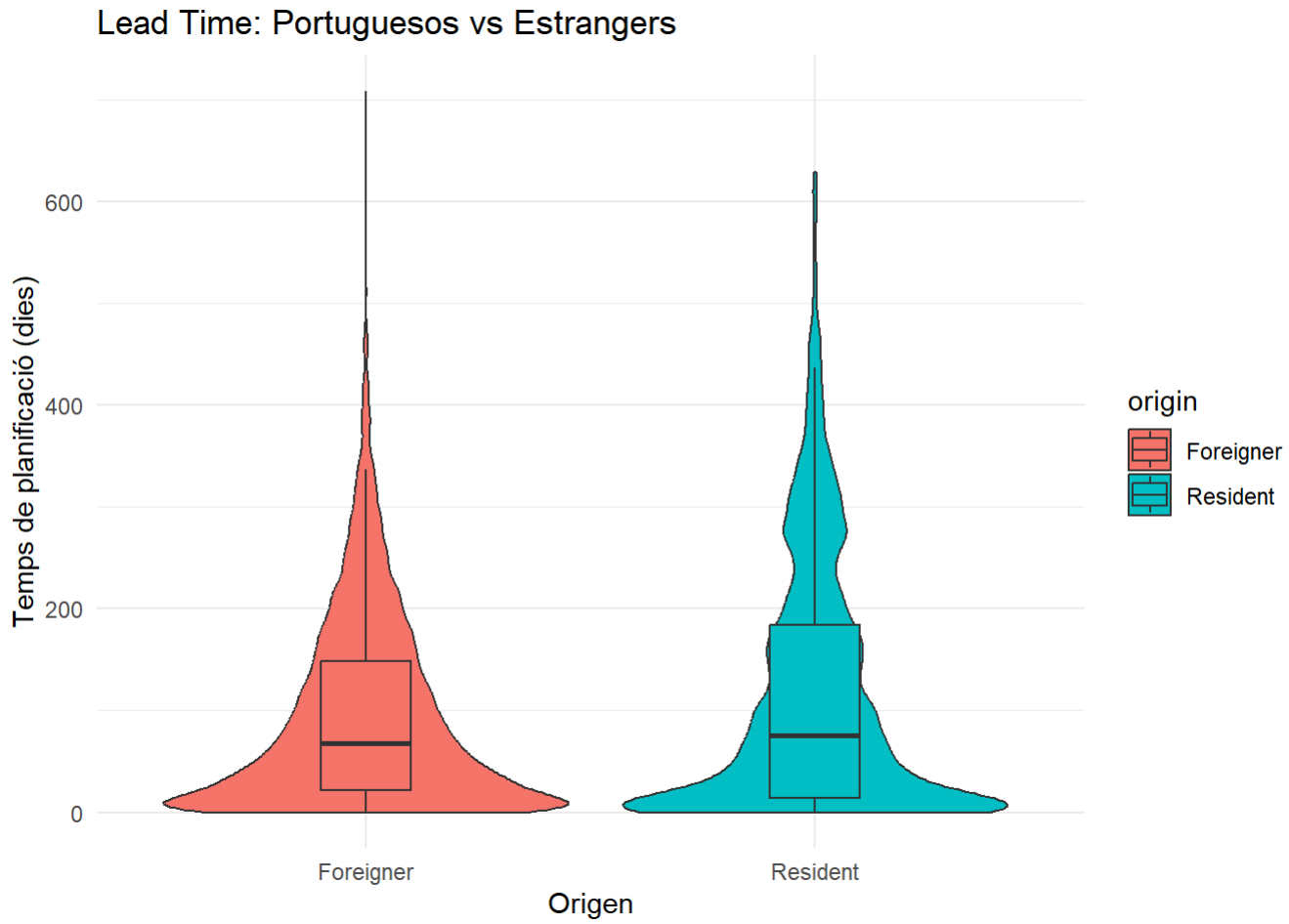


Taxa de cancel.lació: La taxa de cancel.lació és molt més alta que la dels estrangers. La meitat de les reserves fetes pels residents són cancel.lades. Segurament la confiança augmenta la impulsivitat i la flexibilitat. Els estrangers planifiquen millor al ser viatges més llars.

```
# Comparem ADR amb test estadístic
t.test(adr ~ origin, data = x)
```

```
##
##  Welch Two Sample t-test
##
## data:  adr by origin
## t = 51.829, df = 101904, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group Foreigner and group Residen
t is not equal to 0
## 95 percent confidence interval:
##  13.67193 14.74661
## sample estimates:
## mean in group Foreigner  mean in group Resident
##               109.19079                94.98152
```

```
# Creem el gràfic de distribució
ggplot(x, aes(x = origin, y = lead_time, fill = origin)) +
  geom_violin(trim = TRUE) +
  geom_boxplot(width = 0.2, outlier.shape = NA) +
  labs(title = "Lead Time: Portuguesos vs Estrangers",
       x = "Origen", y = "Temps de planificació (dies)") +
  theme_minimal()
```



ADR:

el p-valor del test és 2.2e-16, la qual cosa ens fa apreciar una diferència gran entre els grups. Els estrangers gasten més per nit, segurament perquè estan dispoats a pagar preus més alts o potser perquè trien allotjaments de més qualitat. També pot ser perquè tiren dates de vacances llargues.

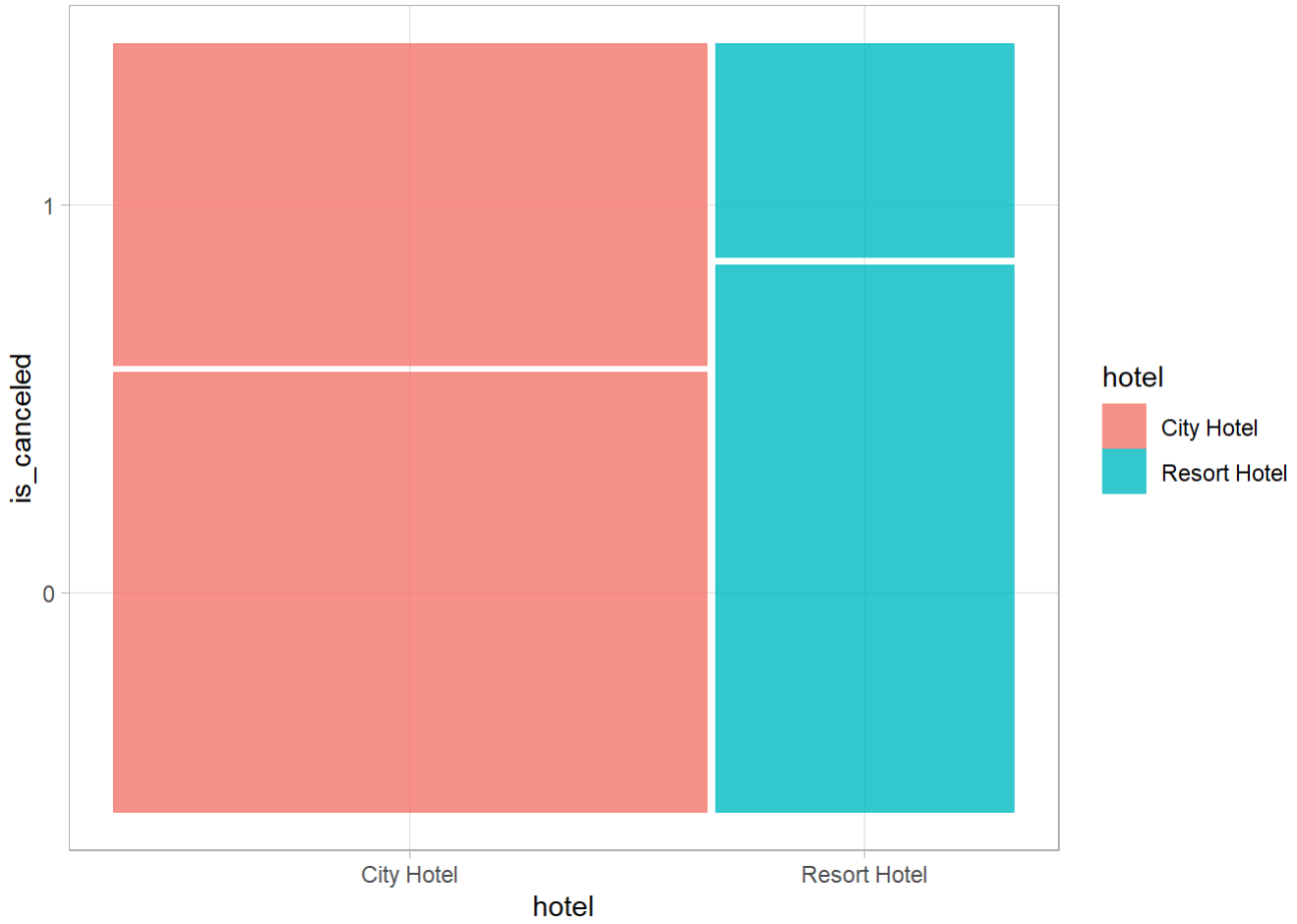
Otra de las variables interesantes es is_canceled que nos indica si una reserva fue cancelada o no (un 37.0% de las veces). Podemos ver la relación entre dos variables categóricas usando un gráfico de mosaico:

```
# require(ggmosaic)
x$is_canceled=as.factor(x$is_canceled)
ggplot(data=x) +
  geom_mosaic(aes(x=product(is_canceled, hotel), fill=hotel)) +
  theme_light()
```

```
## Warning: The `scale_name` argument of `continuous_scale()` is deprecated as of ggplot2
## 3.5.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `trans` argument of `continuous_scale()` is deprecated as of ggplot2 3.5.0.
## ℹ Please use the `transform` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

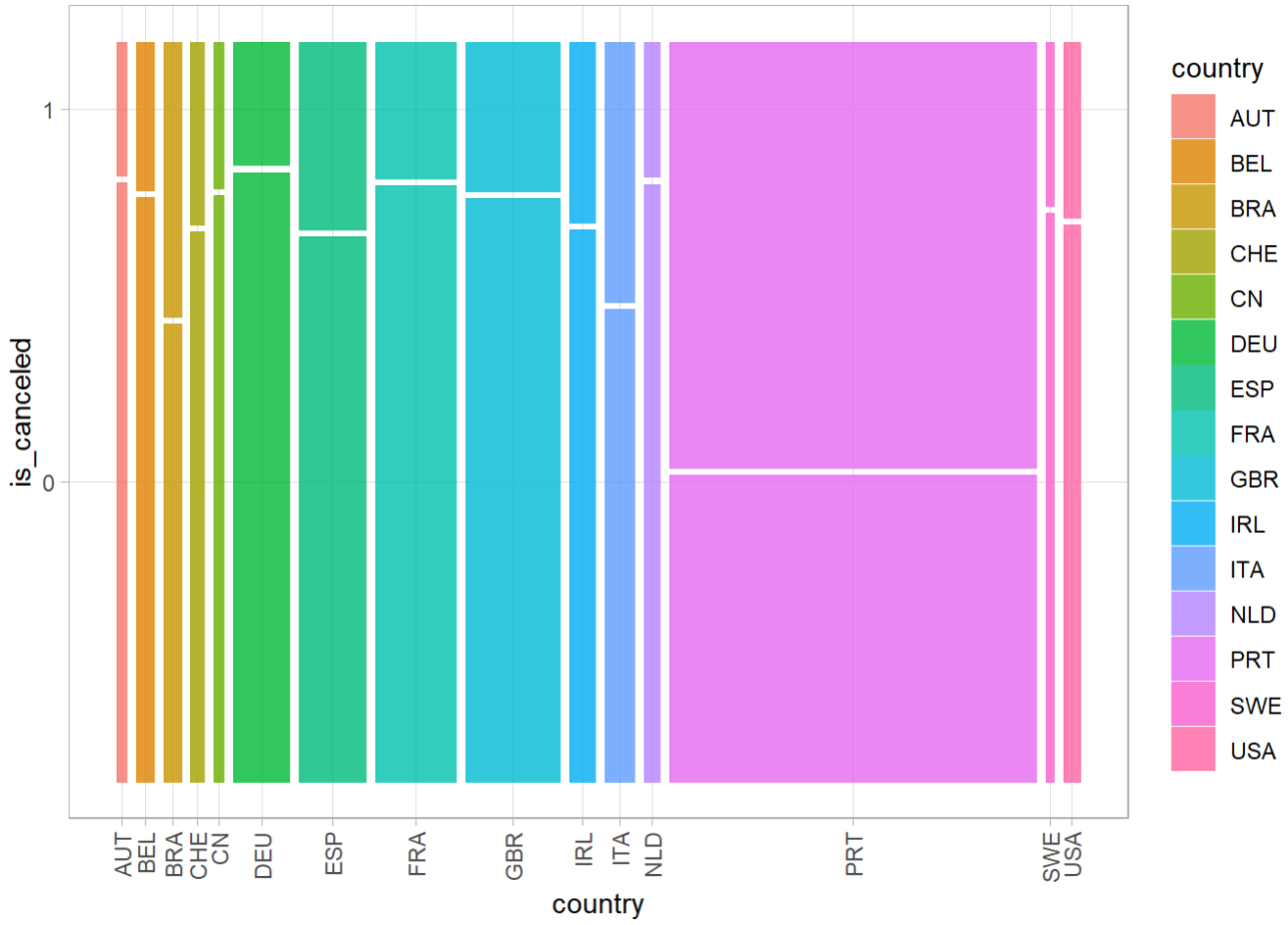
```
## Warning: `unite_()` was deprecated in tidyr 1.2.0.
## ℹ Please use `unite()` instead.
## ℹ The deprecated feature was likely used in the ggmosaic package.
##   Please report the issue at <https://github.com/haleyjeppson/ggmosaic>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Se puede observar que el porcentaje de cancelaciones (1 en el eje Y) en un resort es inferior al de un hotel en la ciudad de Lisboa. En el eje X, los tamaños relativos de cada columna se corresponden también con la proporción de cada tipo de hotel. Es importante no pensar en las etiquetas del eje Y (0 / 1) como la propoción numérica real de cancelación, ya que puede llevar a engaño.

En el caso de cancelación por país para los países con más turistas:

```
# paises con al menos 1000 reservas
xx = x %>% group_by(country) %>% mutate(pais=n()) %>% filter(pais>=1000)
xx$country=factor(xx$country)
ggplot(data=xx) +
  geom_mosaic(aes(x=product(is_canceled, country), fill=country)) +
  theme_light() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



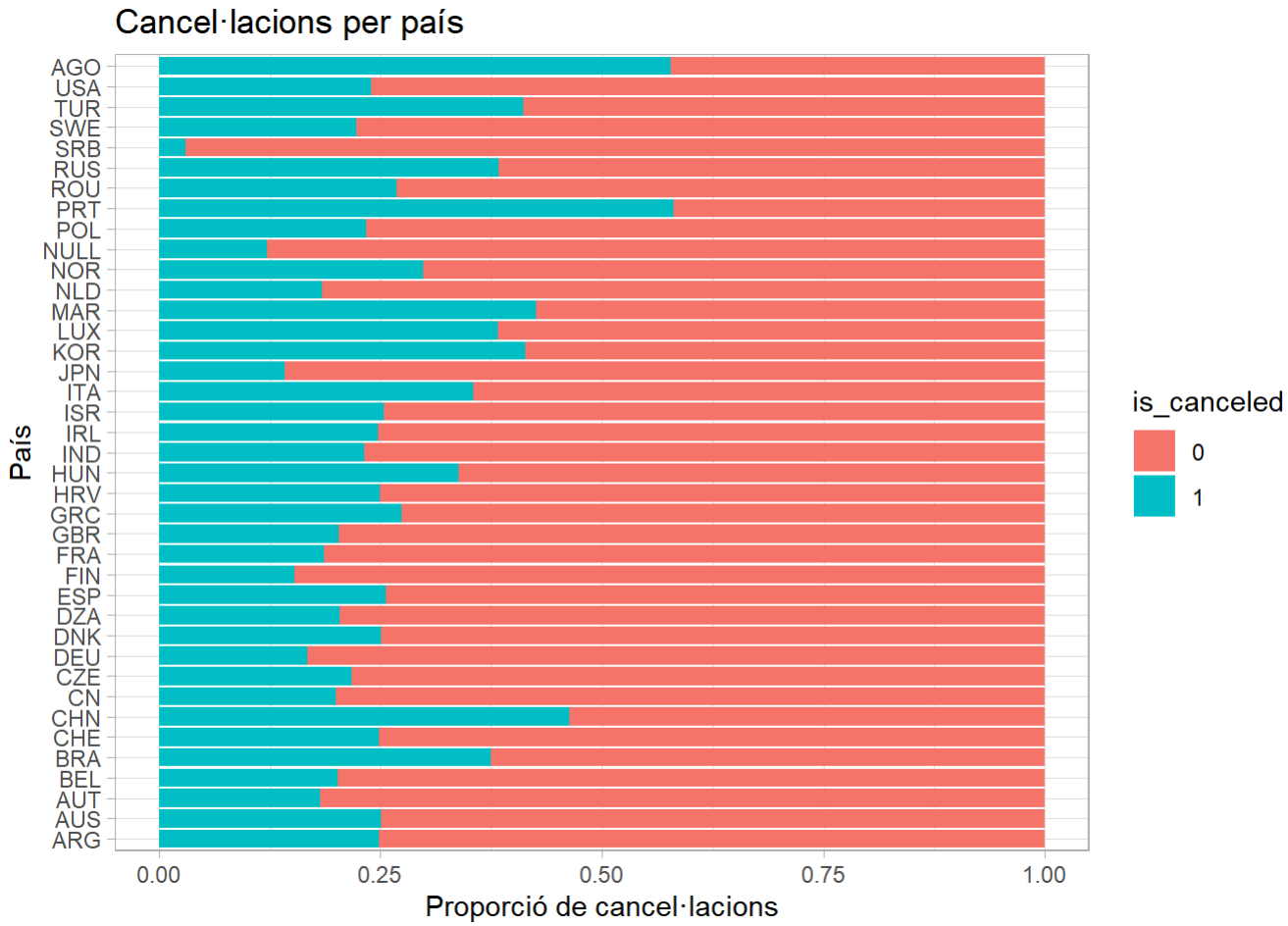
Se puede comprobar que la tasa de cancelaciones es mucho mayor para los turistas locales (de Portugal, PRT), mientras que es mucho más baja para el resto de paises. No obstante, este gráfico no es de lectura fácil, en este caso no hay ningún orden ni de los paises ni del porcentaje de cancelaciones.

EJERCICIO: mejorar el gráfico anterior para hacerlo más inteligible, y plantearse si es posible visualizar las relaciones entre tres o más variables de tipo categórico.

```
# Simplifiquem i millorem  gràfic de mosaics per fer-lo més intel.ligible.
library(forcats)
xx <- x %>%
  group_by(country) %>%
  filter(n() >= 100) %>%
  mutate(country = fct_reorder(country, is_canceled, .fun = mean))
```

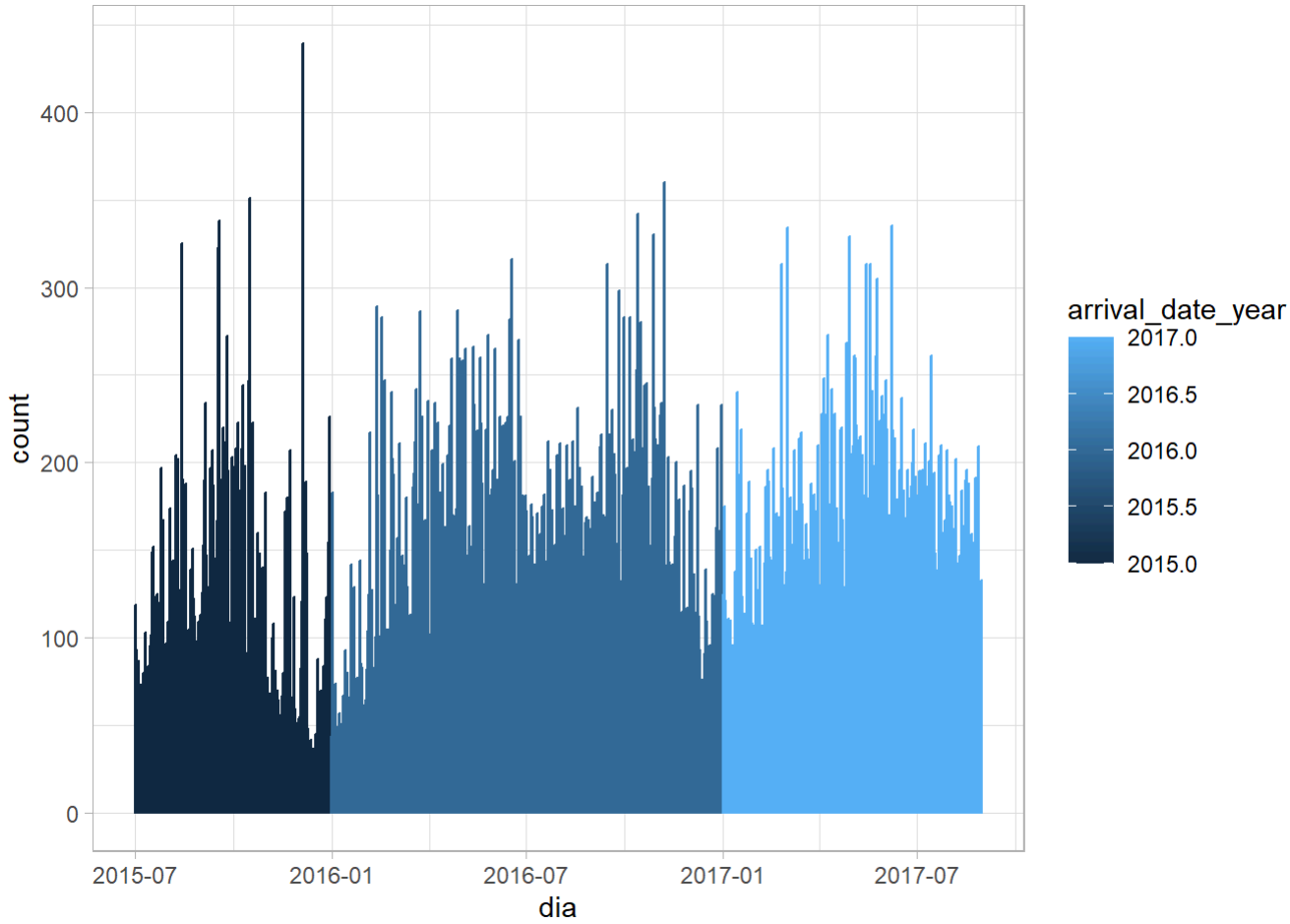
```
## Warning: There were 39 warnings in `mutate()`.
## The first warning was:
## ℹ In argument: `country = fct_reorder(country, is_canceled, .fun = mean)`.
## ℹ In group 1: `country = AGO`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## ℹ Run `dplyr::last_dplyr_warnings()` to see the 38 remaining warnings.
```

```
ggplot(data=xx, aes(x=country, fill=is_canceled)) +
  geom_bar(position="fill") +
  coord_flip() +
  labs(title="Cancel·lacions per país",
       x="País", y="Proporció de cancel·lacions") +
  theme_light()
```



Finalmente, vamos a analizar el comportamiento de las reservas con respecto a la fecha de entrada. Primero, usando el package lubridate de R (una maravilla para manipular datos de tipo fecha y hora) crearemos una variable dia para saber qué día de la semana fue la llegada al hotel, y analizaremos cuantas reservas hubo cada día:

```
# require(lubridate)
x$dia=as_date(paste0(x$arrival_date_year,'-',x$arrival_date_month,'-',x$arrival_date_day_of_m
onth))
ggplot(data=x,aes(x=dia,group=arrival_date_year,color=arrival_date_year)) +
  geom_bar() +
  theme_light()
```

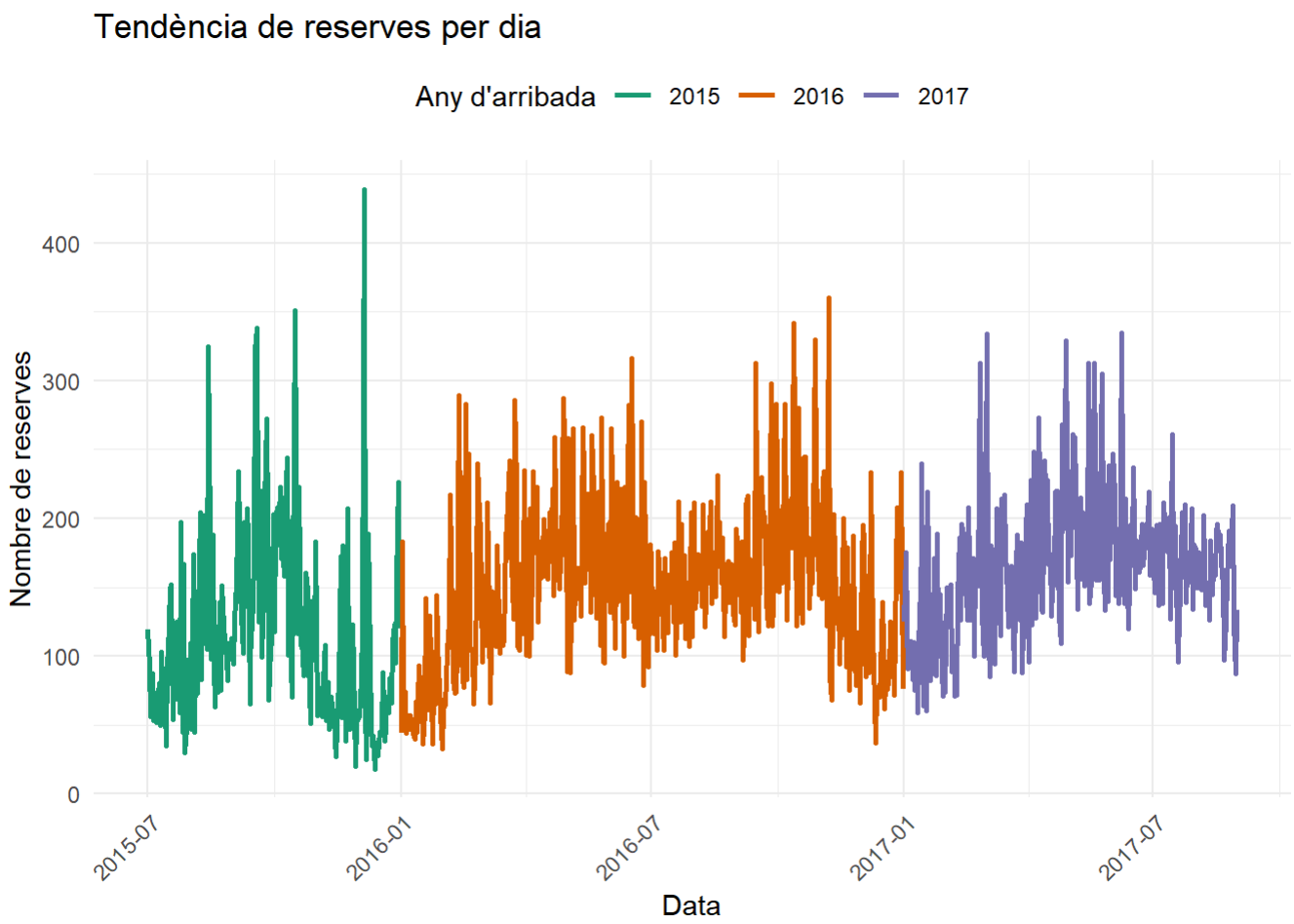


Tal y como describe el artículo, los datos comprenden desde el 1 de Julio de 2015 hasta el 31 de agosto de 2017. Se pueden observar algunos picos que podrían ser interesantes.

EJERCICIO: mejorar y desdoblar el gráfico anterior por tipo de hotel o por país de orígen.

```
# Llibreries necessàries
library(lubridate)
library(ggplot2)
library(dplyr)

# 1. Crear la variable 'dia' assegurant-nos que és una data vàlida
x$dia <- as_date(paste0(x$arrival_date_year, "-", x$arrival_date_month, "-", x$arrival_date_d
ay_of_month))

# 2. Filtrar files amb valors vàlids
x <- x %>% filter(!is.na(dia))

# 3. Crear el gràfic per mostrar la tendència de les reserves per dia
ggplot(data = x, aes(x = dia, color = as.factor(arrival_date_year))) +
  geom_line(stat = "count", size = 1) +  # Utilitzar línies per mostrar la tendència
  scale_color_brewer(palette = "Dark2", name = "Any d'arribada") +  # Millorar els colors
  labs(
    title = "Tendència de reserves per dia",
    x = "Data",
    y = "Nombre de reserves"
  ) +
  theme_minimal() +
  theme(
    legend.position = "top",
    axis.text.x = element_text(angle = 45, hjust = 1)  # Rotar etiquetes de l'eix X
  )
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Con el día calculado, junto con las variables stays_in_week/weekend_nights podemos tratar de categorizar manualmente el tipo de viaje, de acuerdo a los siguientes criterios (arbitrarios, claramente mejorables):

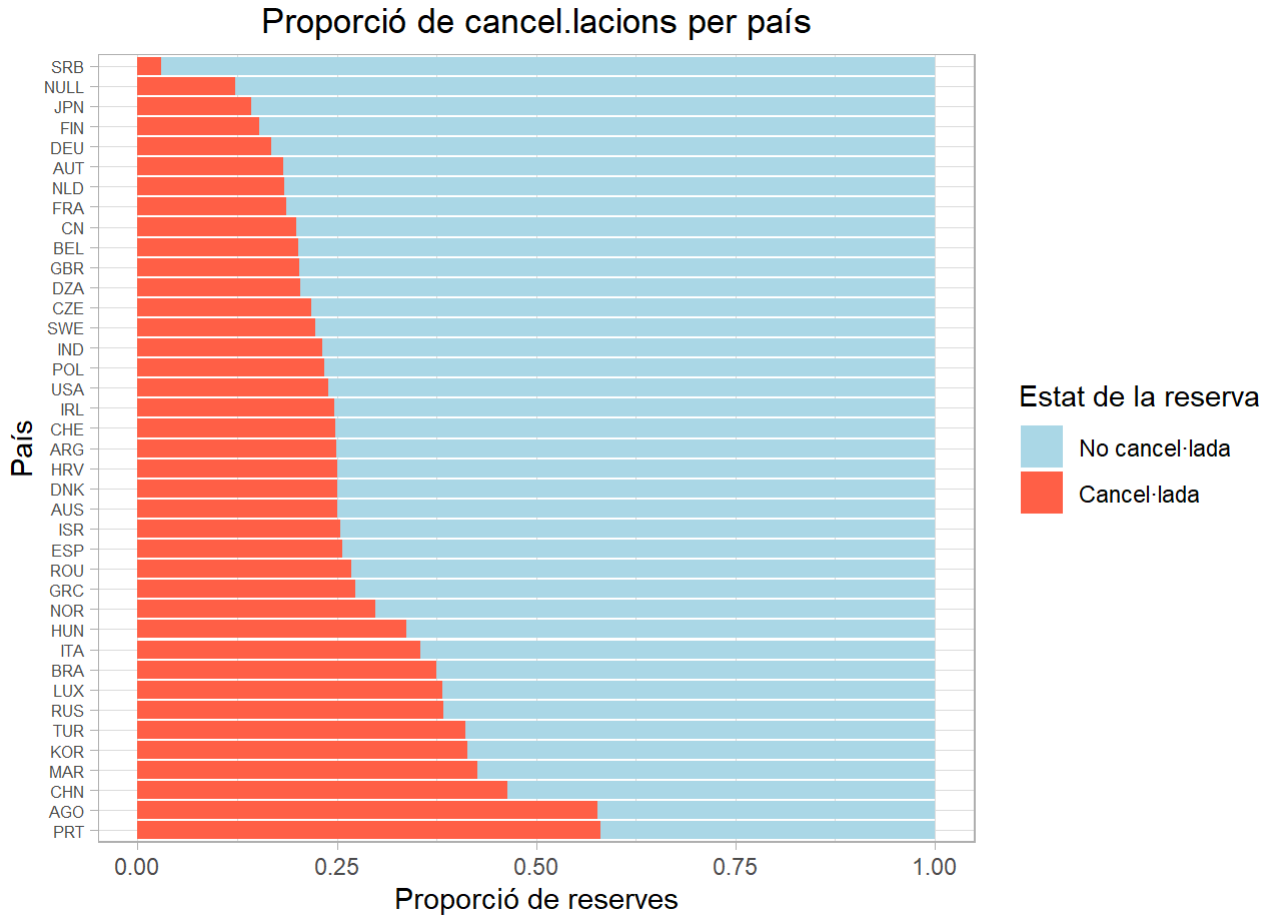
1. si stays_in_weekend_nights es cero => viaje de trabajo
2. si stays_in_week_nights es cero o uno y en este caso la entrada es en viernes => fin de semana
3. si stays_in_week_nights es cinco y stays_in_weekend_nights es tres (es decir, de sábado a sábado o de domingo a domingo o de sábado a domingo) => paquete semanal de vacaciones
4. si stays_in_weekend_nights es uno o dos y stays_in_week_days es cinco o menos => trabajo + descanso
5. el resto => vacaciones

Una manera de refinar esta clasificación sería mirar la cantidad de adultos, niños y bebés para decidir si se trata de una persona viajando por trabajo o bien una familia.

```
# require(lubridate)
x$tipo=ifelse(x$stays_in_weekend_nights==0, "work",
       ifelse(x$stays_in_week_nights==0, "weekend",
       ifelse(x$stays_in_week_nights==1 & wday(x$dia)==6, "weekend",
       ifelse(x$stays_in_week_nights==5 & (x$stays_in_weekend_nights==3 | x$stays_in_weekend_
nights==4), "package",
       ifelse(x$stays_in_week_nights<=5 & x$stays_in_weekend_nights<3, "work+rest",
       "rest")))))
```

Las posibilidades son infinitas: se puede enriquecer el dataset con datos de tipo geográfico (la distancia entre paises), demográficos, económicos (renta per capita), etc. Debéis explorar este dataset y en este proceso de exploración decidir qué historia queréis explicar sobre el mismo.

```
# Anàlisi de cancel·lacions per país

# Llibreries
library(dplyr)
library(ggplot2)
library(forcats)

# 1. Convertim is_canceled a factor (x poder agrupar)
x$is_canceled <- as.factor(x$is_canceled)


# 2. Filtrem x països amb  100 reserves mínim i reordenar x proporció cancel·lacions
xx <- x %>%
  group_by(country) %>%
  filter(n() >= 100) %>%  # països >=100 reserves
  summarise(
    cancel_rate = mean(as.numeric(as.character(is_canceled))),  # Proporció cancel.lacions
    total_reserves = n()
  ) %>%
  arrange(desc(cancel_rate))  # Ordenem països segons  proporció x fer-lo més impactant i ent
endidoor



# 3. Reordenem països x cancel_rate
x$country <- factor(x$country, levels = xx$country)


# 4. Creem gràfic de barres apilat (% x país)
plot <- ggplot(data = x %>% filter(country %in% xx$country), aes(x = country, fill = is_cance
led)) +
  geom_bar(position = "fill") +  # Gràfic apilat
  coord_flip() +  # Girem x facilitar la lectura
  scale_fill_manual(
    values = c("0" = "lightblue", "1" = "tomato"),
    labels = c("No cancel·lada", "Cancel·lada")
  ) +  # Colors cotnrastats
  labs(
    title = "Proporció de cancel.lacions per país",
    x = "País",
    y = "Proporció de reserves",
    fill = "Estat de la reserva"
  ) +
  theme_light() +
  theme(
    axis.text.y = element_text(size = 6),  # Mida text països
    plot.margin = margin(10, 20, 10, 10)  # Mida marges
  ) +
  theme(plot.title = element_text(hjust = 0.5))  # Centrem títol

# 5. Mostrem gràfic
print(plot)
```




```
# 6. Guardem gràfic
ggsave("cancelacions_per_pais.png", plot = plot, height = 15, width = 8)
```