Examen 1: Filtrado y Manipulación de Datos

Jesus Mudarra Luján

2023-09-05

Pregunta 1

Intenta describir con frases entendibles el conjunto de vuelos retrasados. Intenta dar afirmaciones como por ejemplo:

- Un vuelo tiende a salir unos 5 minutos antes el 21% de las veces y a salir tarde el 30% de las veces restantes:

```
# Cargar librerías
flights %>%
  summarize(cinco_min_antes = mean(dep_delay < -5, na.rm = TRUE),
            cinco_min_tarde = mean(dep_delay > 5, na.rm = TRUE))
```

```
# A tibble: 1 x 2
  cinco_min_antes cinco_min_tarde
            <dbl>           <dbl>
1           0.212           0.303
```

- Los vuelos de la compañía `MQ` nunca llegan más de una hora antes:

```
flights %>%
  filter(carrier=="MQ") %>%
  summarise(min_time = min(arr_delay, na.rm=T))
```

```
# A tibble: 1 x 1
  min_time
     <dbl>
1      -53
```

- El 66% de los vuelos a `HNL` llegan a tiempo y casi el 13% se retrasan más de 20 minutos:

```
flights %>%
  filter(dest == "HNL") %>%
  summarise(a_tiempo = mean(arr_delay <= 0, na.rm = TRUE),
            mas_de_3_horas = mean(arr_delay > 20, na.rm = TRUE))
```

```
# A tibble: 1 x 2
  a_tiempo mas_de_3_horas
     <dbl>          <dbl>
1    0.659          0.130
```

Pregunta 2

Primero guardamos los vuelos no cancelados:

```
not_cancelled <- flights %>%
  filter(!is.na(air_time))
```

Da una versión equivalente a las pipes siguientes sin usar la función `count`:

```
not_cancelled %>% count(dest)
```

Alternativa:

```
not_cancelled %>%
  group_by(dest) %>%
  summarise(n = n())
```

```
# A tibble: 104 x 2
   dest      n
   <chr> <int>
 1 ABQ     254
 2 ACK     264
 3 ALB     418
 4 ANC       8
 5 ATL   16837
 6 AUS    2411
 7 AVL     261
 8 BDL     412
 9 BGR     358
10 BHM     269
# ... with 94 more rows
```

```
not_cancelled %>% count(tailnum, wt = distance)
```

Alternativa:

```
not_cancelled %>%
  group_by(tailnum) %>%
  summarise(total_distance = sum(distance,na.rm=T))
```

```
# A tibble: 4,037 x 2
   tailnum total_distance
   <chr>            <dbl>
 1 D942DN            3418
 2 N0EGMQ          239143
 3 N10156          109664
 4 N102UW           25722
 5 N103US           24619
 6 N104UW           24616
 7 N10575          139903
 8 N105UW           23618
 9 N107US           21677
10 N108UW           32070
# ... with 4,027 more rows
```

Pregunta 3

Para definir un vuelo cancelado hemos usado la función:

`(is.na(dep_delay) | is.na(arr_delay))`

Intenta dar una definición que sea mejor, ya que la nuestra es un poco subóptima. ¿Cuál es la columna más importante?

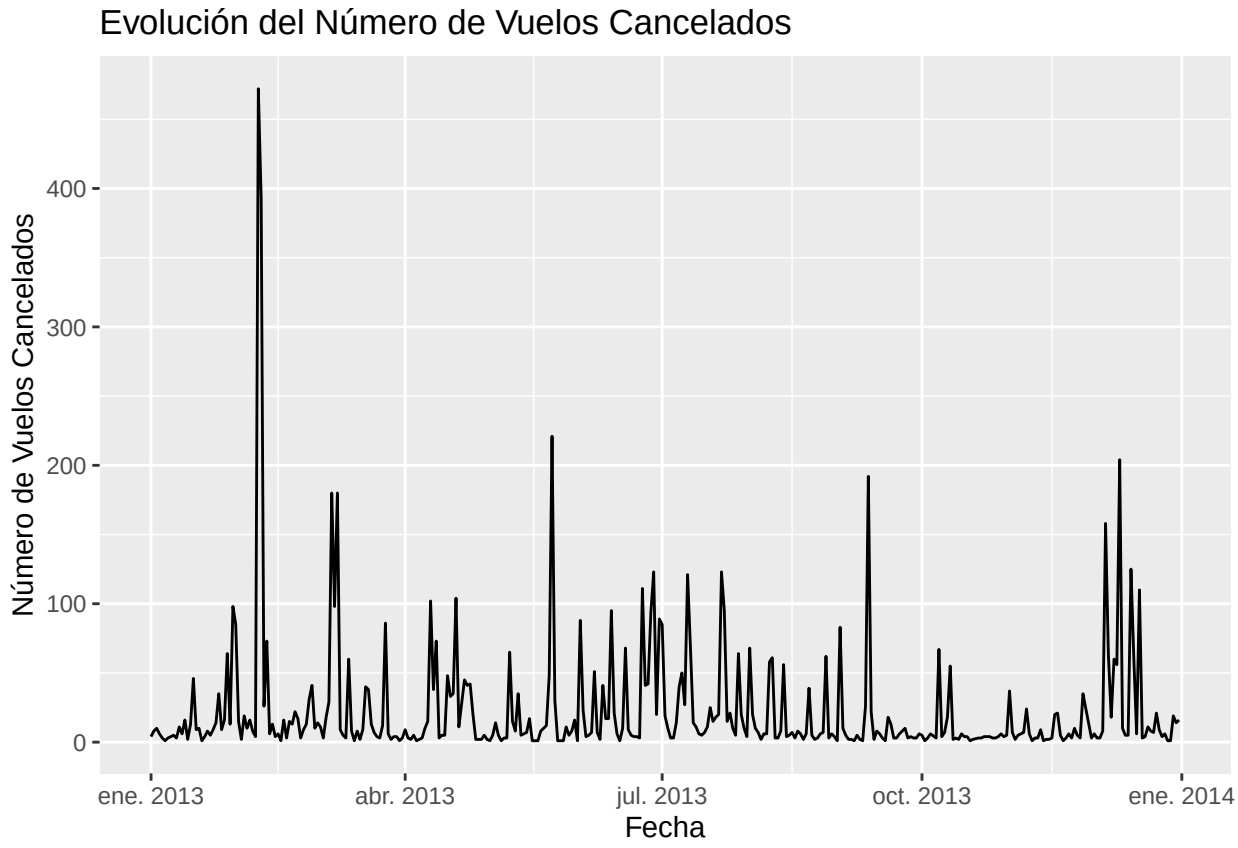
```
cancelled <- flights %>%
  filter(is.na(dep_time))
```

La columna más importante es `dep_time` que nos indica la hora de salida. Si el valor de `dep_time` es `NA`, ya nos indica que el vuelo ha sido cancelado y no es necesario el valor de otras variables.

Pregunta 4

Investiga si existe algún patrón del número de vuelos que se cancelan cada día.

```
flights %>%
  filter(is.na(dep_time)) %>%
  group_by(year, month, day) %>%
  summarise(count = n()) %>%
  ggplot(aes(x = as.Date(paste(year,month,day,sep="-")), y = count)) +
    geom_line() +
    labs(title = "Evolución del Número de Vuelos Cancelados",
         x = "Fecha",
         y = "Número de Vuelos Cancelados")
```

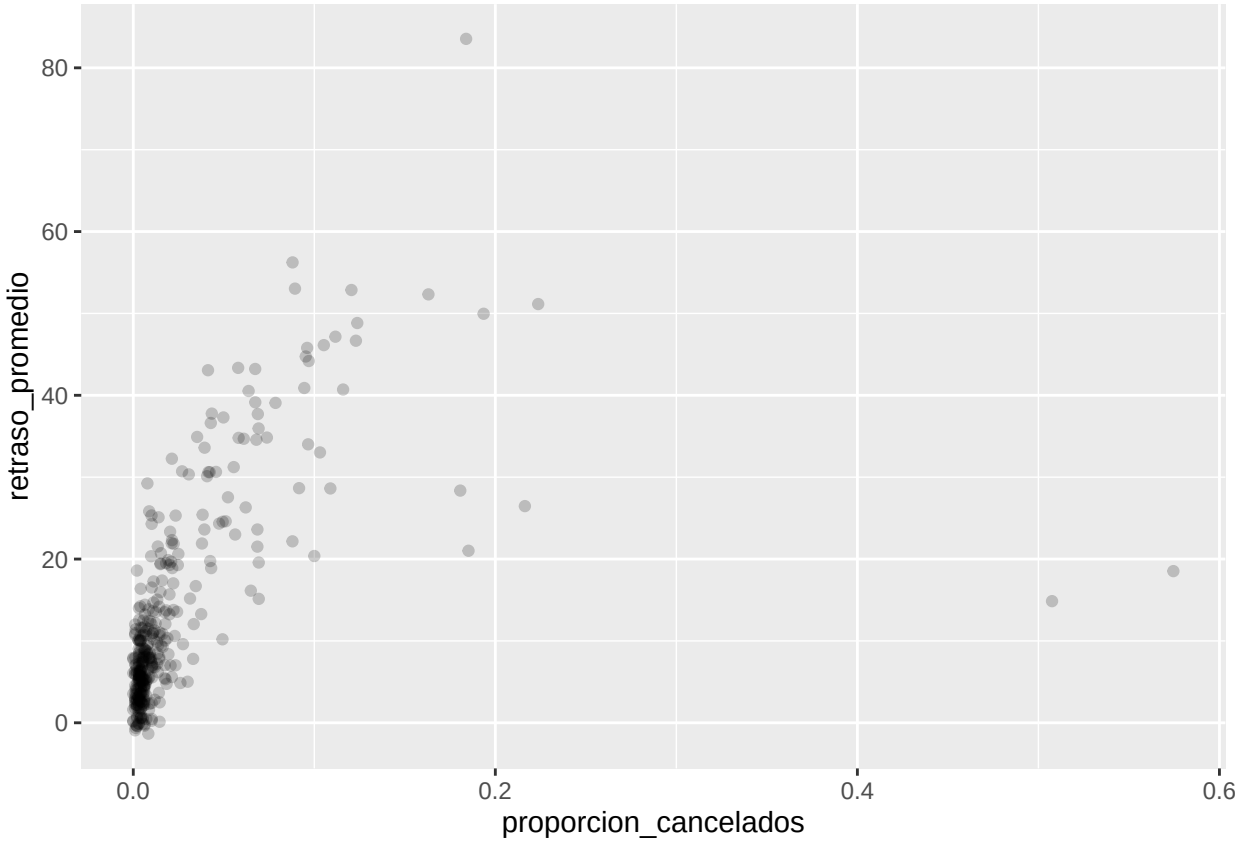


Los retrasos se incrementan en los meses de las vacaciones de verano y festivos por Navidad y principios de año.

Investiga si la proporción de vuelos cancelados está relacionada con el retraso promedio por día en los vuelos.

```
#proporción de vuelos cancelados vs retraso promedio por día
flights %>%
  group_by(year,month,day) %>%
  summarize(proporcion_cancelados = mean(is.na(dep_time)),
            retraso_promedio = mean(dep_delay, na.rm=T)) %>%
  ggplot(mapping = aes(x = proporcion_cancelados, y = retraso_promedio)) +
```

```
  geom_point(alpha = 0.2)
```



No se observa ningún tipo de relación entre la proporción de vuelos cancelados y el retraso promedio cuando agrupamos por días.

Investiga si la proporción de vuelos cancelados está relacionada con el retraso promedio por aeropuerto en los vuelos.

```
#proporción de vuelos cancelados vs retraso promedio por aeropuerto
flights %>%
  group_by(origin) %>%
  summarize(proporcion_cancelados = mean(is.na(dep_time)),
            retraso_promedio = mean(dep_delay, na.rm=T))
```

```
# A tibble: 3 x 3
  origin proporcion_cancelados retraso_promedio
  <chr>                  <dbl>            <dbl>
1 EWR                   0.0268             15.1
2 JFK                   0.0167             12.1
3 LGA                   0.0301             10.3
```

No existe ninguna relación entre la proporción de vuelos cancelados y el retraso promedio cuando agrupamos por aeropuertos de salida.

¿Qué compañía aérea sufre los peores retrasos?

```
flights %>%
  group_by(carrier) %>%
  summarize(retraso = mean(dep_delay, na.rm=T)) %>%
```

```
  arrange(desc(retraso))
```

```
# A tibble: 16 x 2
   carrier retraso
   <chr>     <dbl>
 1 F9        20.2
 2 EV        20.0
 3 YV        19.0
 4 FL        18.7
 5 WN        17.7
 6 9E        16.7
 7 B6        13.0
 8 VX        12.9
 9 OO        12.6
10 UA        12.1
11 MQ        10.6
12 DL         9.26
13 AA         8.59
14 AS         5.80
15 HA         4.90
16 US         3.78
```

La compañía aérea que sufre más retrasos es `Envoy Air (MQ)`

Pregunta 5

Difícil: Intenta desentrañar los efectos que producen los retrasos por culpa de malos aeropuertos vs malas compañías aéreas. Por ejemplo, intenta usar:

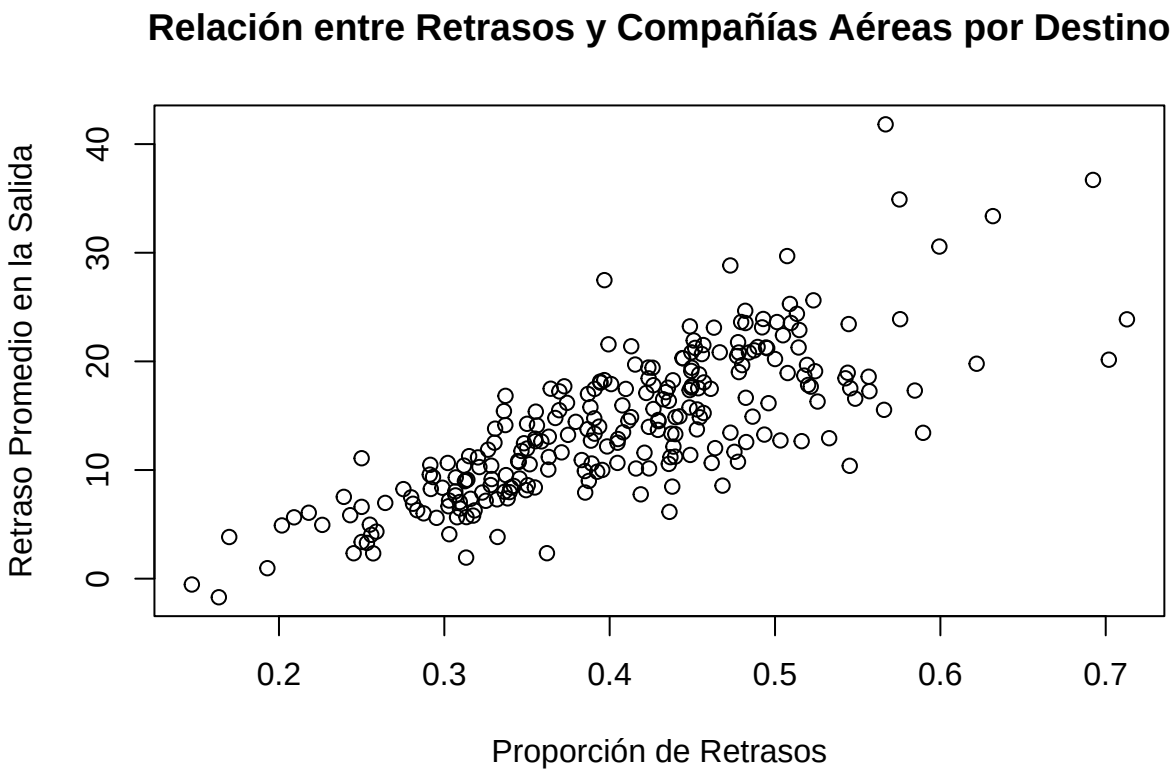
```
vuelos_por_comp_dest <- flights %>%
  group_by(carrier, dest) %>%
  summarise(num_vuelos = n())

retrasos_por_comp_dest <- flights %>%
  group_by(carrier, dest) %>%
  summarise(mean_dep_delay = mean(dep_delay, na.rm = TRUE),
            prop_retrasos = mean(dep_delay > 0, na.rm = TRUE))

resumen_final <- left_join(vuelos_por_comp_dest, retrasos_por_comp_dest, by = c("carrier","dest"))

umbral_min_vuelos <- 50  # Por ejemplo, compañías con al menos 50 vuelos
resumen_filtrado <- resumen_final %>%
  filter(num_vuelos >= umbral_min_vuelos)

plot(resumen_filtrado$prop_retrasos, resumen_filtrado$mean_dep_delay,
     xlab = "Proporción de Retrasos", ylab = "Retraso Promedio en la Salida",
     main = "Relación entre Retrasos y Compañías Aéreas por Destino")
```



Se ve una clara tendencia ascendente en el retraso promedio en la salida al aumentar la proporción de retrasos.

Pregunta 6

¿Qué hace el parámetro `sort` como argumento de `count()`? ¿Cuándo puede sernos útil?

El parámetro `sort` en la función `count()` controla si los resultados se deben ordenar o no en orden descendente cuando se realiza el conteo.

Vuelve a la lista de funciones útiles para filtrar y mutar y describe cómo cada operación cambia cuando la juntamos con un `group_by`.

No entiendo esta pregunta.

Pregunta 7

Vamos a por los peores aviones. Investiga el top 10 de qué aviones (número de cola y compañía) llegaron más tarde a su destino.

```
top_tarde <- flights %>%
  group_by(tailnum, carrier) %>%
  summarise(mean = mean(arr_delay, na.rm=T)) %>%
  arrange(desc(mean))
head(top_tarde, 10)
```

```
# A tibble: 10 x 3
# Groups:   tailnum [10]
   tailnum carrier  mean
   <chr>   <chr>   <dbl>
```

```
 1 N844MH  DL       320
 2 N911DA  DL       294
 3 N922EV  EV       276
 4 N587NW  DL       264
 5 N851NW  DL       219
 6 N928DN  DL       201
 7 N7715E  WN       188
 8 N654UA  UA       185
 9 N665MQ  MQ       175.
10 N427SW  OO       157
```

Pregunta 8

Queremos saber qué hora del día nos conviene volar si queremos evitar los retrasos en la salida.

```
flights %>%
  group_by(hour) %>%
  summarize(retraso_promedio = mean(dep_delay, na.rm=TRUE)) %>%
  arrange(retraso_promedio)
```

```
# A tibble: 20 x 2
    hour retraso_promedio
   <dbl>            <dbl>
 1     5            0.688
 2     6            1.64
 3     7            1.91
 4     8            4.13
 5     9            4.58
 6    10            6.50
 7    11            7.19
 8    12            8.61
 9    13           11.4
10    14           13.8
11    23           14.0
12    15           16.9
13    16           18.8
14    22           18.8
15    17           21.1
16    18           21.1
17    21           24.2
18    20           24.3
19    19           24.8
20     1          NaN
```

A las 5 de la mañana es la hora del día que conviene volar ya que es cuando menos retrasos se han producido en la salida.

Difícil: Queremos saber qué día de la semana nos conviene volar si queremos evitar los retrasos en la salida.

```
flights %>%
  mutate(day_of_week = weekdays(as.Date(paste(year,month,day,sep="-")))) %>%
  group_by(day_of_week) %>%
  summarize(retraso_promedio = mean(dep_delay, na.rm=TRUE)) %>%
  arrange(retraso_promedio)
```

```
# A tibble: 7 x 2
  day_of_week retraso_promedio
```

```
  <chr>                  <dbl>
1 sábado                  7.65
2 martes                 10.6
3 domingo                11.6
4 miércoles              11.8
5 viernes                14.7
6 lunes                  14.8
7 jueves                 16.1
```

El día de la semana que más conviene volar es el sábado.

Pregunta 9

Para cada destino, calcula el total de minutos de retraso acumulado.

```
flights %>%
  group_by(dest) %>%
  summarise(total_min_delay_cum = sum(dep_delay, na.rm=T)) %>%
  arrange(desc(total_min_delay_cum))
```

```
# A tibble: 105 x 2
   dest  total_min_delay_cum
   <chr>               <dbl>
 1 ORD                225840
 2 ATL                211391
 3 SFO                170221
 4 MCO                157661
 5 FLL                151933
 6 LAX                151136
 7 BOS                131387
 8 CLT                126335
 9 DEN                109140
10 DTW                107019
# ... with 95 more rows
```

Para cada uno de ellos, calcula la proporción del total de retraso para dicho destino.

Pregunta 10

Los retrasos suelen estar correlacionados con el tiempo. Aunque el problema que ha causado el primer retraso de un avión se resuelva, el resto de vuelos se retrasan para que salgan primero los aviones que debían haber partido antes. Intenta usar la función `lag()` y explora cómo el retraso de un avión se relaciona con el retraso del avión inmediatamente anterior o posterior.

```
flights_with_delays <- flights %>%
  arrange(year, month, day, dep_time) %>%
  mutate(previous_dep_delay = lag(dep_delay, order_by = dep_time),
         next_dep_delay = lead(dep_delay, order_by = dep_time))
```

```
correlation_previous <- cor(flights_with_delays$dep_delay,
                            flights_with_delays$previous_dep_delay, use = "complete.obs")
correlation_previous
```

```
[1] 0.254277
```

```
correlation_next <- cor(flights_with_delays$dep_delay,
                        flights_with_delays$next_dep_delay, use = "complete.obs")
```

```
correlation_next
```

```
[1] 0.254277
```

Pregunta 11

Vamos a por los destinos esta vez. Localiza vuelos que llegaron "demasiado rápido" a sus destinos. Seguramente, el becario se equivocó al introducir el tiempo de vuelo y se trate de un error en los datos. Calcula para ello el cociente entre el tiempo en el aire de cada vuelo relativo al tiempo de vuelo del avión que tardó menos en llegar a dicho destino. ¿Qué vuelos fueron los que más se retrasaron en el aire?

```
flights %>%
  group_by(dest) %>%
  mutate(tiempo_aire_relativo = air_time/min(air_time, na.rm=T)) %>%
  arrange(desc(tiempo_aire_relativo)) %>%
  select(tailnum, tiempo_aire_relativo)
```

```
# A tibble: 336,776 x 3
# Groups:   dest [105]
   dest  tailnum tiempo_aire_relativo
   <chr> <chr>                 <dbl>
 1 BOS   N37255                 5.33
 2 BOS   N967UW                 5.10
 3 BOS   N284JB                 4.71
 4 BOS   N3FKAA                 4.57
 5 BOS   N76522                 4.38
 6 BOS   N346NB                 4.33
 7 BOS   N913XJ                 4.10
 8 BOS   N3DRAA                 4.10
 9 DCA   N745VJ                 4.09
10 ACK   N328JB                 4.03
# ... with 336,766 more rows
```

Pregunta 12

Encuentra todos los destinos a los que vuelan dos o más compañías y para cada uno de ellos, crea un ranking de las mejores compañías para volar a cada destino (utiliza el criterio que consideres más conveniente como probabilidad de retraso, velocidad o tiempo de vuelo, número de vuelos al año...)

```
destinos_multiples <- flights %>%
  group_by(dest) %>%
  summarize(num_comp = n_distinct(carrier)) %>%
  filter(num_comp >= 2)
head(destinos_multiples)
```

```
# A tibble: 6 x 2
  dest  num_comp
  <chr>    <int>
1 ATL          7
2 AUS          6
3 AVL          2
4 BDL          2
5 BGR          2
6 BNA          5
```

```r
probabilidad_retraso <- flights %>%
  group_by(carrier, dest) %>%
  summarize(probabilidad_retraso = mean(dep_delay > 0, na.rm = TRUE))

destinos_con_multiples_comp <- destinos_multiples %>%
  left_join(probabilidad_retraso, by = "dest")

ranking_comp_por_destino <- destinos_con_multiples_comp %>%
  arrange(dest, probabilidad_retraso) %>%
  group_by(dest) %>%
  mutate(ranking = row_number()) %>%
  filter(ranking <= 3)

head(ranking_comp_por_destino) # Seleccionar las 3 mejores compañías por destino
```

```
# A tibble: 6 x 5
# Groups:   dest [2]
  dest  num_comp carrier probabilidad_retraso ranking
  <chr>    <int> <chr>                  <dbl>   <int>
1 ATL          7 9E                     0.193       1
2 ATL          7 MQ                     0.293       2
3 ATL          7 DL                     0.312       3
4 AUS          6 WN                     0.332       1
5 AUS          6 DL                     0.345       2
6 AUS          6 B6                     0.442       3
```

Finalmente, para cada avión (basándonos en el número de cola) cuenta el número de vuelos que hace antes de sufrir su primer retraso de más de una hora. Valora entonces la fiabilidad del avión o de la compañía aérea asociada al mismo.

```r
flights %>%
  arrange(tailnum, year, month, day, dep_time) %>%
  group_by(tailnum) %>%
  mutate(primer_retraso = cumsum(dep_delay > 60) == 0) %>%
  filter(primer_retraso) %>%
  summarize(num_vuelos_sin_retraso = n()) %>%
  arrange(desc(num_vuelos_sin_retraso))
```

```
# A tibble: 3,755 x 2
   tailnum num_vuelos_sin_retraso
   <chr>                    <int>
 1 N954UW                     206
 2 N952UW                     163
 3 N957UW                     142
 4 N5FAAA                     117
 5 N38727                      99
 6 N3742C                      98
 7 N5EWAA                      98
 8 N705TW                      97
 9 N765US                      97
10 N635JB                      94
# ... with 3,745 more rows
```