

Minería de datos: PRA1 - Selección y preparación de un juego de datos

Autor: Tim Thorp

Diciembre 2023

Contents

Objetivo analítico	2
Juego de datos	2
Limpieza	5
Análisis exploratorio	7
Análisis temporal	7
Análisis por aerolínea	9
Análisis por aeropuerto	11
Análisis por hora programada de salida y llegada (parte 1)	15
Análisis por duración programada (parte 1)	17
Análisis por distancia	19
Discretización de atributos	20
Discretización de horas programadas de salida y llegada	20
Análisis por hora programada de salida y llegada (parte 2)	21
Discretización de duración del vuelo	23
Análisis por duración programada (parte 2)	24
Transformación de la hora del día	25
Análisis de correlaciones	26
Análisis SVD	28
Análisis de los componentes	31
Interpretación de resultados	33

Conclusiones	33
Resumen de hallazgos	33
Implicaciones para el modelado	33
Bibliografía	34

Objetivo analítico

En este estudio sobre retrasos de vuelos en Estados Unidos, abordamos un problema de clasificación con el objetivo de predecir si un vuelo se retrasará más de 15 minutos, considerándolo como una variable binaria. Utilizamos variables como los aeropuertos de origen y destino, la aerolínea, el día de la semana, la hora programada de salida y la distancia del vuelo.

Hemos seleccionado la precisión como nuestra métrica principal, con un objetivo de alcanzar al menos un 80% de precisión. Este análisis busca no solo predecir retrasos, sino también identificar los factores más influyentes y analizar patrones temporales, así como diferencias entre aerolíneas y rutas. La metodología incluye el uso de modelos de clasificación, como la regresión logística y los árboles de decisión, evaluando su rendimiento mediante la división de los datos en conjuntos de entrenamiento y prueba, y mediante la validación cruzada.

Este proyecto no solo ayudará a las aerolíneas y a los aeropuertos en su planificación operativa, sino que también mejorará la experiencia del cliente al proporcionar información sobre los retrasos en los vuelos. La primera fase del estudio se centra en un análisis exploratorio del conjunto de datos, seguido de tareas de limpieza y acondicionamiento. Además, aplicaremos métodos de discretización y realizaremos un estudio utilizando SVD para la reducción de la dimensionalidad, preparando así los datos para el modelado en la siguiente fase del estudio.

Juego de datos

Hemos seleccionado nuestro conjunto de datos desde el *Bureau of Transportation Statistics* del Departamento de Transporte de Estados Unidos. La página web ofrece más de 100 variables, pero nos centramos en las más relevantes para nuestro estudio.

Debido a limitaciones de hardware, solo analizaremos los datos del último mes disponible, septiembre de 2023, que contiene 569,338 observaciones (vuelos). Este enfoque limita la generalización del estudio, pero es necesario para manejar el volumen de datos con nuestros recursos actuales.

El conjunto de datos seleccionado incluye 9 variables numéricas, 5 categóricas y 3 binarias. Las variables son:

- **DAY_OF_WEEK:** Día de la semana en que se realiza el vuelo, en formato numérico (1 = lunes, 7 = domingo). Se tratará como categórica.

- **FL_DATE**: Fecha del vuelo.
- **OP_UNIQUE_CARRIER**: Código de identificación único para la aerolínea operadora.
- **ORIGIN**: Código del aeropuerto de origen.
- **DEST**: Código del aeropuerto de destino.
- **CRS_DEP_TIME**: Hora de salida programada (formato hhmm).
- **DEP_TIME**: Hora de salida real (formato hhmm).
- **DEP_DELAY_NEW**: Retraso en la salida (en minutos).
- **CRS_ARR_TIME**: Hora de llegada programada (formato hhmm).
- **ARR_TIME**: Hora de llegada real (formato hhmm).
- **ARR_DELAY_NEW**: Retraso en la llegada (en minutos).
- **ARR_DEL15**: Indicador de si el vuelo llegó con más de 15 minutos de retraso (1 = sí, 0 = no).
- **CANCELLED**: Indicador de si el vuelo fue cancelado (1 = sí, 0 = no).
- **DIVERTED**: Indicador de si el vuelo fue desviado (1 = sí, 0 = no).
- **CRS_ELAPSED_TIME**: Tiempo total estimado del vuelo (en minutos).
- **ACTUAL_ELAPSED_TIME**: Tiempo total real del vuelo (en minutos).
- **DISTANCE**: Distancia del vuelo en millas.

Estas variables nos permiten aplicar algoritmos supervisados, no supervisados y reglas de asociación para un análisis detallado de los retrasos en los vuelos.

Los vuelos cancelados (**CANCELLED**) y desviados (**DIVERTED**) están fuera del ámbito de este estudio, pero utilizaremos estas variables para filtrar los datos.

Para empezar, cargamos el fichero de datos.

```
flightData <- read.csv('../data/flightData.csv', row.names=NULL, stringsAsFactors=TRUE)
```

Primero, verificamos la estructura del conjunto de datos principal. Observamos el número de columnas y algunos ejemplos del contenido de las filas.

```
str(flightData)
```

```
## 'data.frame':    569338 obs. of  17 variables:
## $ DAY_OF_WEEK      : int   1 1 1 1 1 1 1 1 1 1 ...
## $ FL_DATE          : Factor w/ 30 levels "9/1/2023 12:00:00 AM",...: 25 25 25 25 25 25 25 25 25 25 ...
## $ OP_UNIQUE_CARRIER : Factor w/ 15 levels "9E","AA","AS",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ ORIGIN           : Factor w/ 340 levels "ABE","ABI","ABQ",...: 1 1 1 5 5 12 12 13 15 15 ...
## $ DEST             : Factor w/ 340 levels "ABE","ABI","ABQ",...: 21 21 21 21 21 21 21 21 99 190 ..
## $ CRS_DEP_TIME      : int   654 1242 1727 735 1405 610 1620 1015 1245 706 ...
## $ DEP_TIME          : int   649 1237 1717 725 1400 609 1636 1010 1237 701 ...
## $ DEP_DELAY_NEW     : num   0 0 0 0 0 0 16 0 0 0 ...
## $ CRS_ARR_TIME       : int   900 1450 1938 848 1512 859 1906 1123 1428 803 ...
## $ ARR_TIME          : int   842 1431 1905 820 1454 904 1913 1100 1411 759 ...
## $ ARR_DELAY_NEW     : num   0 0 0 0 0 5 7 0 0 0 ...
## $ ARR_DEL15         : num   0 0 0 0 0 0 0 0 0 0 ...
## $ CANCELLED         : num   0 0 0 0 0 0 0 0 0 0 ...
## $ DIVERTED          : num   0 0 0 0 0 0 0 0 0 0 ...
## $ CRS_ELAPSED_TIME   : num   126 128 131 73 67 109 106 68 103 57 ...
## $ ACTUAL_ELAPSED_TIME: num   113 114 108 55 54 115 97 50 94 58 ...
## $ DISTANCE          : num   692 692 692 145 145 500 500 143 489 136 ...
```

Vemos que contamos con **17 variables** y **569.338** registros.

La variable **DAY_OF_WEEK** está actualmente clasificada como de tipo entero. A pesar de su naturaleza numérica, sería más adecuado considerarla como categórica (factor) en el contexto de este estudio.

La variable FL_DATE consta de 30 niveles, lo que coincide con el número de días en septiembre. El formato de la fecha es MM/DD/YYYY.

Hay 15 aerolíneas distintas (OP_UNIQUE_CARRIER) y 340 aeropuertos (ORIGIN y DEST).

A continuación, miramos el resumen de los datos.

```
summary(flightData)
```

```
## DAY_OF_WEEK FL_DATE OP_UNIQUE_CARRIER
## Min. :1.000 9/21/2023 12:00:00 AM: 20358 WN :117870
## 1st Qu.:2.000 9/28/2023 12:00:00 AM: 20341 DL : 81701
## Median :4.000 9/22/2023 12:00:00 AM: 20268 AA : 76972
## Mean :4.066 9/18/2023 12:00:00 AM: 20264 UA : 62591
## 3rd Qu.:6.000 9/25/2023 12:00:00 AM: 20246 OO : 59245
## Max. :7.000 9/29/2023 12:00:00 AM: 20242 YX : 24839
## (Other) :447619 (Other):146120
## ORIGIN DEST CRS_DEP_TIME DEP_TIME
## ATL : 28344 ATL : 28359 Min. : 4 Min. : 1
## DEN : 24738 DEN : 24727 1st Qu.: 910 1st Qu.: 911
## DFW : 24533 DFW : 24531 Median :1319 Median :1320
## ORD : 22384 ORD : 22368 Mean :1328 Mean :1329
## CLT : 16175 CLT : 16168 3rd Qu.:1734 3rd Qu.:1741
## LAX : 16152 LAX : 16152 Max. :2359 Max. :2400
## (Other):437012 (Other):437033 NA's :6617
## DEP_DELAY_NEW CRS_ARR_TIME ARR_TIME ARR_DELAY_NEW
## Min. : 0.00 Min. : 1 Min. : 1 Min. : 0.0
## 1st Qu.: 0.00 1st Qu.:1105 1st Qu.:1047 1st Qu.: 0.0
## Median : 0.00 Median :1516 Median :1501 Median : 0.0
## Mean : 14.19 Mean :1492 Mean :1463 Mean : 14.1
## 3rd Qu.: 7.00 3rd Qu.:1920 3rd Qu.:1915 3rd Qu.: 7.0
## Max. :2360.00 Max. :2359 Max. :2400 Max. :2367.0
## NA's :6618 NA's :7145 NA's :8451
## ARR_DEL15 CANCELLED DIVERTED CRS_ELAPSED_TIME
## Min. :0.000 Min. :0.0000 Min. :0.000000 Min. : 24
## 1st Qu.:0.000 1st Qu.:0.0000 1st Qu.:0.000000 1st Qu.: 91
## Median :0.000 Median :0.0000 Median :0.000000 Median :126
## Mean :0.187 Mean :0.0124 Mean :0.002447 Mean :144
## 3rd Qu.:0.000 3rd Qu.:0.0000 3rd Qu.:0.000000 3rd Qu.:174
## Max. :1.000 Max. :1.0000 Max. :1.000000 Max. :670
## NA's :8451
## ACTUAL_ELAPSED_TIME DISTANCE
## Min. : 17.0 Min. : 21.0
## 1st Qu.: 86.0 1st Qu.: 393.0
## Median :121.0 Median : 668.0
## Mean :138.6 Mean : 825.5
## 3rd Qu.:169.0 3rd Qu.:1056.0
## Max. :723.0 Max. :5095.0
## NA's :8451
```

Todos los valores numéricos están dentro del rango esperado, pero existen numerosos valores NA en campos como DEP_TIME, DEP_DELAY_NEW, ARR_TIME, ARR_DELAY_NEW, ARR_DEL15 y ACTUAL_ELAPSED_TIME. Estos podrían corresponder a vuelos cancelados (CANCELLED) o desviados (DIVERTED).

Dado que la media de ARR_DEL15 es igual a 0.187, sabemos que el 18.7% de los vuelos en nuestro dataset se retrasaron 15 minutos o más.

Limpieza

El siguiente paso será la limpieza de datos. Primero cambiamos la variable `DAY_OF_WEEK` a tipo factor.

```
flightData$DAY_OF_WEEK <- factor(flightData$DAY_OF_WEEK, levels = c(1, 2, 3, 4, 5, 6, 7),
                                labels = c("lunes", "martes", "miércoles", "jueves",
                                           "viernes", "sábado", "domingo"))

summary(flightData$DAY_OF_WEEK)
```

```
##      lunes      martes miércoles      jueves      viernes      sábado      domingo
##      80423      73166      73919      80529      100284      84214      76803
```

El viernes registró la mayor cantidad de vuelos en nuestro conjunto de datos, en parte debido a que septiembre de 2023 contó con 5 viernes y 5 sábados, mientras que los otros días de la semana solo tuvieron 4.

Cambiamos el formato de `FL_DATE` a tipo `Date` (YYYY-MM-DD) y quitamos la parte de hora:

```
flightData$FL_DATE <- as.Date(flightData$FL_DATE, format = "%m/%d/%Y %I:%M:%S %p")
summary(flightData$FL_DATE)
```

```
##      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
## "2023-09-01" "2023-09-08" "2023-09-16" "2023-09-15" "2023-09-23" "2023-09-30"
```

Confirmamos que todos los vuelos están comprendidos entre el 1 y 30 de septiembre.

Añadimos los nombres de las aerolíneas para facilitar la interpretación de `OP_UNIQUE_CARRIER`.

```
carrier_names <- c("9E" = "Endeavor Air",
                  "AA" = "American Airlines",
                  "AS" = "Alaska Airlines",
                  "B6" = "JetBlue Airways",
                  "DL" = "Delta Air Lines",
                  "F9" = "Frontier Airlines",
                  "G4" = "Allegiant Air",
                  "HA" = "Hawaiian Airlines",
                  "MQ" = "Envoy Air",
                  "NK" = "Spirit Airlines",
                  "OH" = "PSA Airlines",
                  "OO" = "SkyWest Airlines",
                  "UA" = "United Airlines",
                  "WN" = "Southwest Airlines",
                  "YX" = "Republic Airways")

levels(flightData$OP_UNIQUE_CARRIER) <- carrier_names
summary(flightData$OP_UNIQUE_CARRIER)
```

```
##      Endeavor Air  American Airlines  Alaska Airlines  JetBlue Airways
##      17319          76972            21426            21412
##      Delta Air Lines  Frontier Airlines  Allegiant Air  Hawaiian Airlines
##      81701          16353            6892            6718
##      Envoy Air      Spirit Airlines    PSA Airlines   SkyWest Airlines
##      18789          21036            16175            59245
##      United Airlines  Southwest Airlines  Republic Airways
##      62591          117870          24839
```

Borramos los vuelos desviados y cancelados del dataset.

```
# Borramos las filas con vuelos desviados o cancelados
flightData <- flightData[!(flightData$CANCELLED == 1 | flightData$DIVERTED == 1), ]

# Borramos las columnas vacías
flightData$CANCELLED <- NULL
flightData$DIVERTED <- NULL

# Mostramos el dataset actualizado
str(flightData)
```

```
## 'data.frame': 560887 obs. of 15 variables:
## $ DAY_OF_WEEK : Factor w/ 7 levels "lunes","martes",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ FL_DATE : Date, format: "2023-09-04" "2023-09-04" ...
## $ OP_UNIQUE_CARRIER : Factor w/ 15 levels "Endeavor Air",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ ORIGIN : Factor w/ 340 levels "ABE","ABI","ABQ",...: 1 1 1 5 5 12 12 13 15 15 ...
## $ DEST : Factor w/ 340 levels "ABE","ABI","ABQ",...: 21 21 21 21 21 21 21 21 99 190 ..
## $ CRS_DEP_TIME : int 654 1242 1727 735 1405 610 1620 1015 1245 706 ...
## $ DEP_TIME : int 649 1237 1717 725 1400 609 1636 1010 1237 701 ...
## $ DEP_DELAY_NEW : num 0 0 0 0 0 0 16 0 0 0 ...
## $ CRS_ARR_TIME : int 900 1450 1938 848 1512 859 1906 1123 1428 803 ...
## $ ARR_TIME : int 842 1431 1905 820 1454 904 1913 1100 1411 759 ...
## $ ARR_DELAY_NEW : num 0 0 0 0 0 5 7 0 0 0 ...
## $ ARR_DEL15 : num 0 0 0 0 0 0 0 0 0 0 ...
## $ CRS_ELAPSED_TIME : num 126 128 131 73 67 109 106 68 103 57 ...
## $ ACTUAL_ELAPSED_TIME: num 113 114 108 55 54 115 97 50 94 58 ...
## $ DISTANCE : num 692 692 692 145 145 500 500 143 489 136 ...
```

El dataset se ha reducido a **15 variables** y **560.887** registros.

Al borrar los vuelos cancelados y desviados del dataset, las columnas DEP_TIME, DEP_DELAY_NEW, ARR_TIME, ARR_DELAY_NEW, ARR_DEL15 y ACTUAL_ELAPSED_TIME ya no presentan valores NA:

```
colSums(is.na(flightData))
```

```
##      DAY_OF_WEEK      FL_DATE  OP_UNIQUE_CARRIER      ORIGIN
##      0              0              0              0
##      DEST      CRS_DEP_TIME      DEP_TIME      DEP_DELAY_NEW
##      0              0              0              0
##      CRS_ARR_TIME      ARR_TIME      ARR_DELAY_NEW      ARR_DEL15
##      0              0              0              0
##      CRS_ELAPSED_TIME  ACTUAL_ELAPSED_TIME      DISTANCE
##      0              0              0
```

Y tampoco hay cadenas vacías:

```
colSums(flightData=="")
```

##	DAY_OF_WEEK	FL_DATE	OP_UNIQUE_CARRIER	ORIGIN
##	0	NA	0	0
##	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY_NEW
##	0	0	0	0
##	CRS_ARR_TIME	ARR_TIME	ARR_DELAY_NEW	ARR_DEL15
##	0	0	0	0
##	CRS_ELAPSED_TIME	ACTUAL_ELAPSED_TIME	DISTANCE	
##	0	0	0	

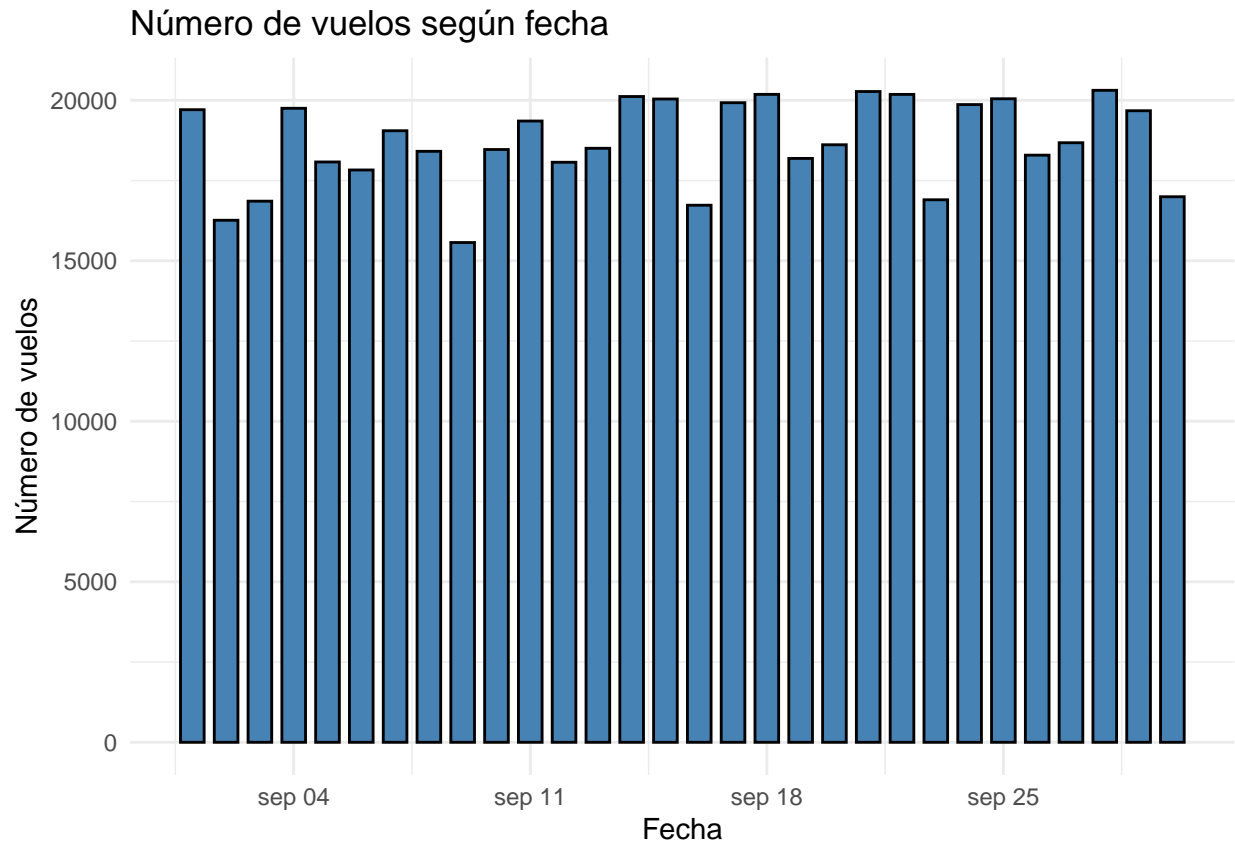
Análisis exploratorio

Vamos a crear gráficos y describir los valores para obtener una visión general de estos atributos y para realizar una primera aproximación a los datos.

Análisis temporal

```
library(ggplot2)

ggplot(flightData, aes(x = FL_DATE)) +
  geom_bar(fill = "steelblue", color = "black", width = 0.7) +
  labs(x = "Fecha", y = "Número de vuelos",
       title = "Número de vuelos según fecha") +
  theme_minimal()
```

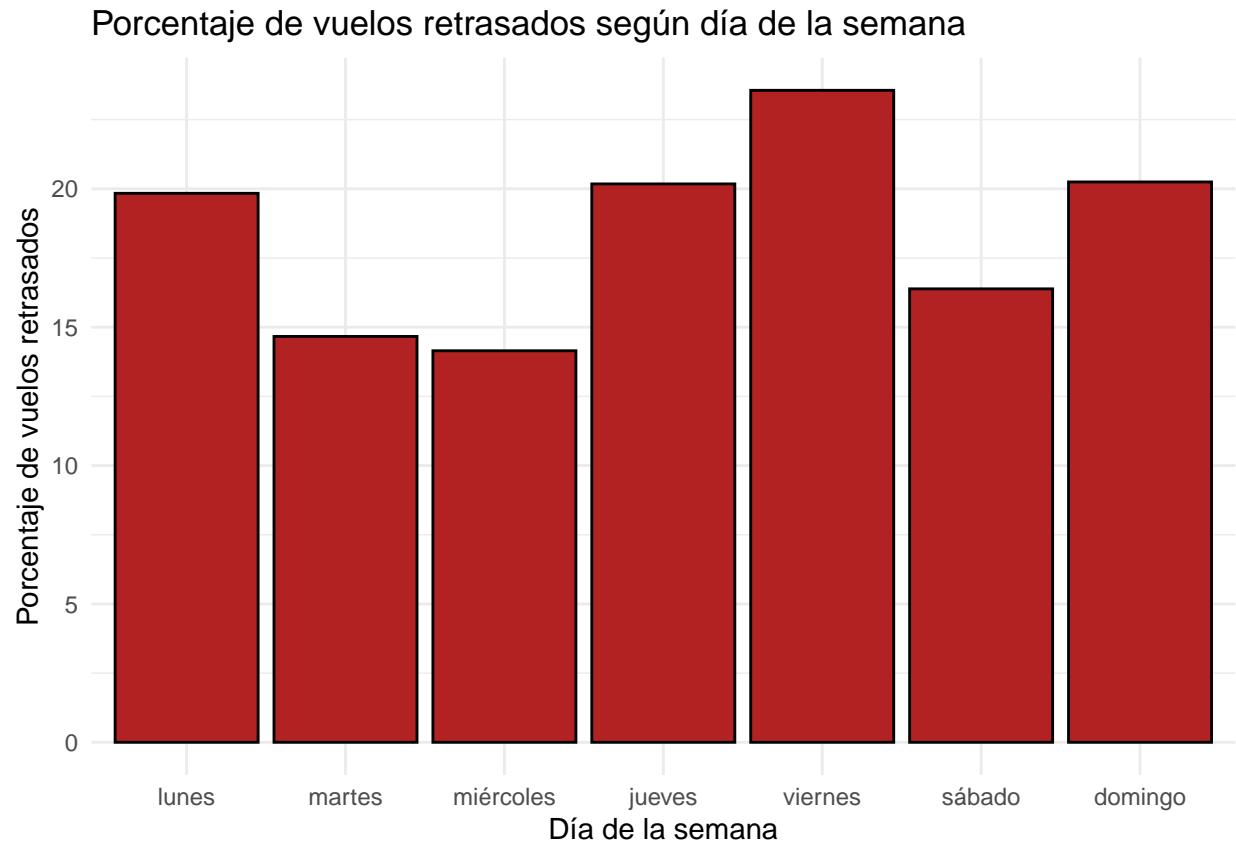


El gráfico revela una variación en el número de vuelos a lo largo del mes. Notamos una disminución en la cantidad de vuelos en ciertas fechas: los días 2, 9, 16, 23 y 30 de septiembre. Todas estas fechas corresponden a sábados.

```
suppressPackageStartupMessages(library(dplyr))

percentage_delays <- flightData %>%
  group_by(DAY_OF_WEEK) %>%
  summarise(Percentage_Delayed = mean(ARR_DEL15, na.rm = TRUE) * 100)

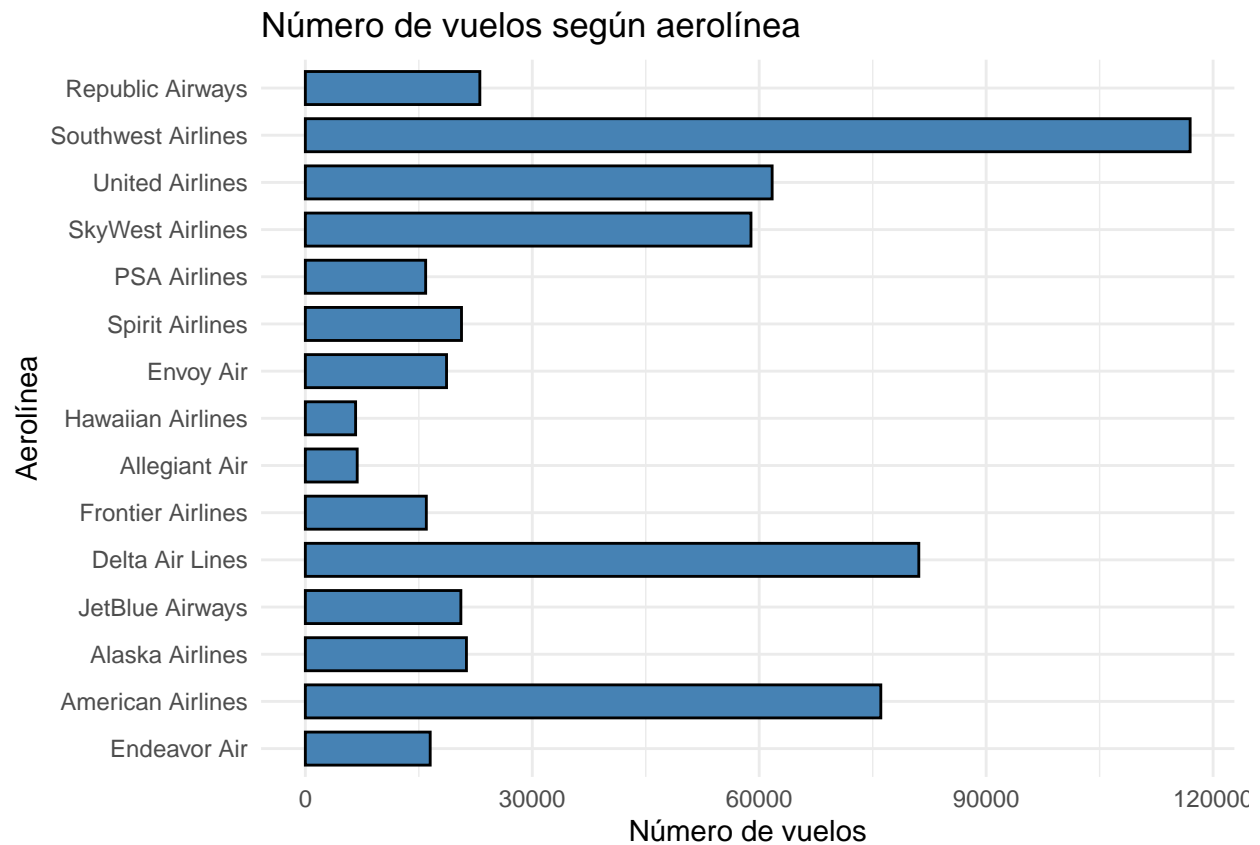
ggplot(percentage_delays, aes(x = DAY_OF_WEEK, y = Percentage_Delayed)) +
  geom_bar(stat = "identity", fill = "firebrick", color = "black") +
  labs(x = "Día de la semana", y = "Porcentaje de vuelos retrasados",
       title = "Porcentaje de vuelos retrasados según día de la semana") +
  theme_minimal()
```

Observamos que los miércoles presentan el menor porcentaje de retrasos (14.7 %), mientras que los viernes muestran el mayor (23.6 %).

Análisis por aerolínea

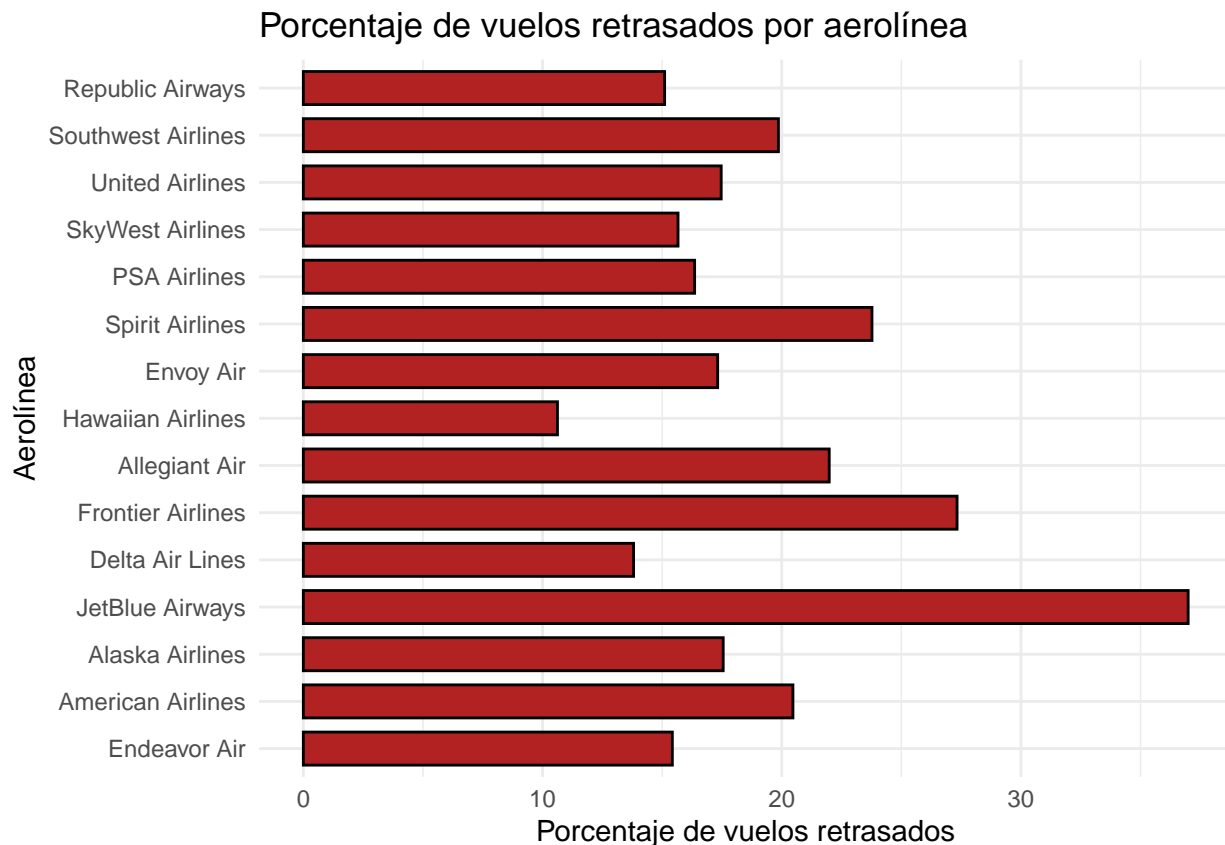
```
ggplot(flightData, aes(y = OP_UNIQUE_CARRIER)) +  
  geom_bar(fill = "steelblue", color = "black", width = 0.7) +  
  theme_minimal() +  
  labs(x = "Número de vuelos", y = "Aerolínea",  
       title = "Número de vuelos según aerolínea")
```



En cuanto al volumen de vuelos, Southwest, Delta y American son las aerolíneas con mayor cantidad.

```
percentage_delays <- flightData %>%
  group_by(OP_UNIQUE_CARRIER) %>%
  summarise(Percentage_Delayed = mean(ARR_DEL15, na.rm = TRUE) * 100)

ggplot(permission_delays, aes(x = Percentage_Delayed, y = OP_UNIQUE_CARRIER)) +
  geom_bar(stat = "identity", fill = "firebrick", color = "black", width = 0.7) +
  labs(x = "Porcentaje de vuelos retrasados", y = "Aerolínea",
       title = "Porcentaje de vuelos retrasados por aerolínea") +
  theme_minimal()
```



El porcentaje de vuelos retrasados varía significativamente entre las aerolíneas. La aerolínea con el mayor porcentaje de retrasos es JetBlue con 37.0 %, mientras que Hawaiian Airlines muestra la menor incidencia con solo el 10.6 %.

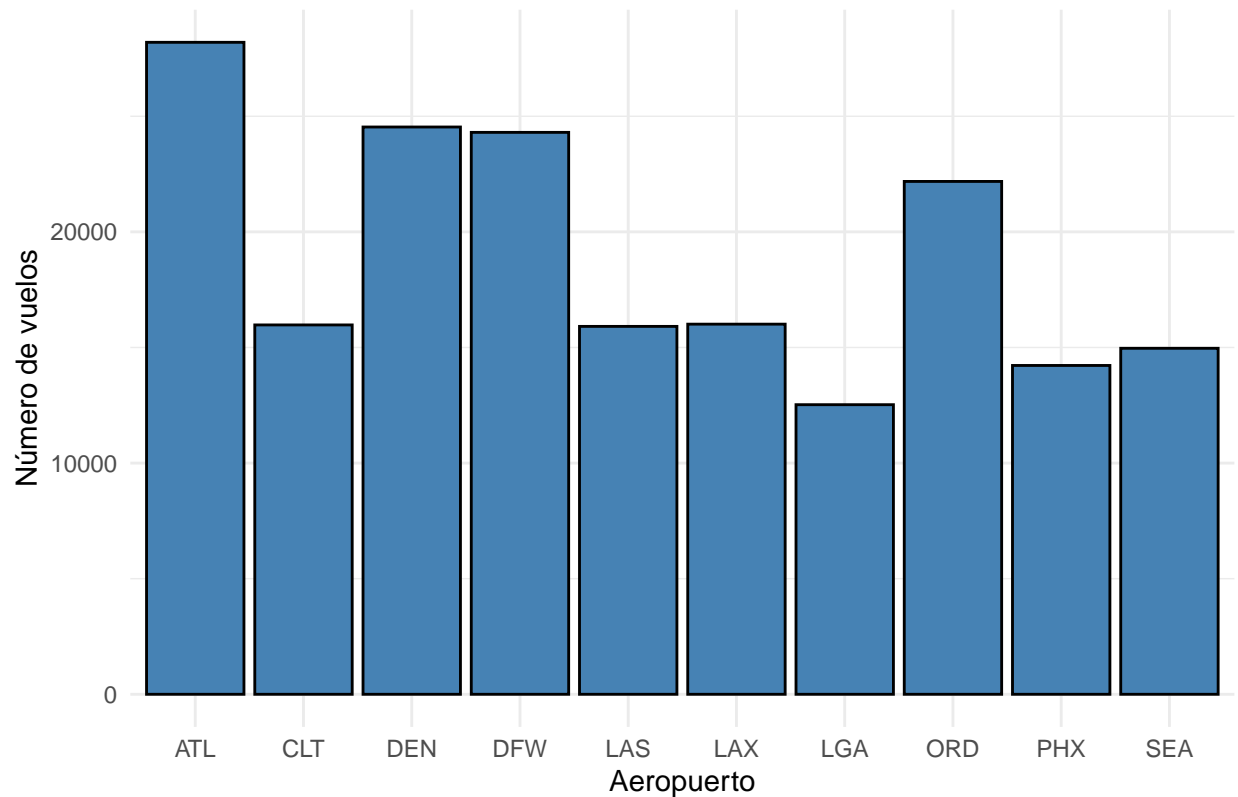
Análisis por aeropuerto

Dado que nuestro dataset cuenta con 340 aeropuertos distintos, este análisis se enfoca en los diez aeropuertos con el mayor número de vuelos.

```
top_airports <- flightData %>%
  group_by(ORIGIN) %>%
  summarise(FlightCount = n()) %>%
  arrange(desc(FlightCount)) %>%
  top_n(10, FlightCount)

ggplot(top_airports, aes(x = ORIGIN, y = FlightCount)) +
  geom_bar(stat = "identity", fill = "steelblue", color = "black") +
  theme_minimal() +
  labs(x = "Aeropuerto", y = "Número de vuelos",
       title = "Los 10 aeropuertos de salida con mayor número de vuelos")
```

Los 10 aeropuertos de salida con mayor número de vuelos

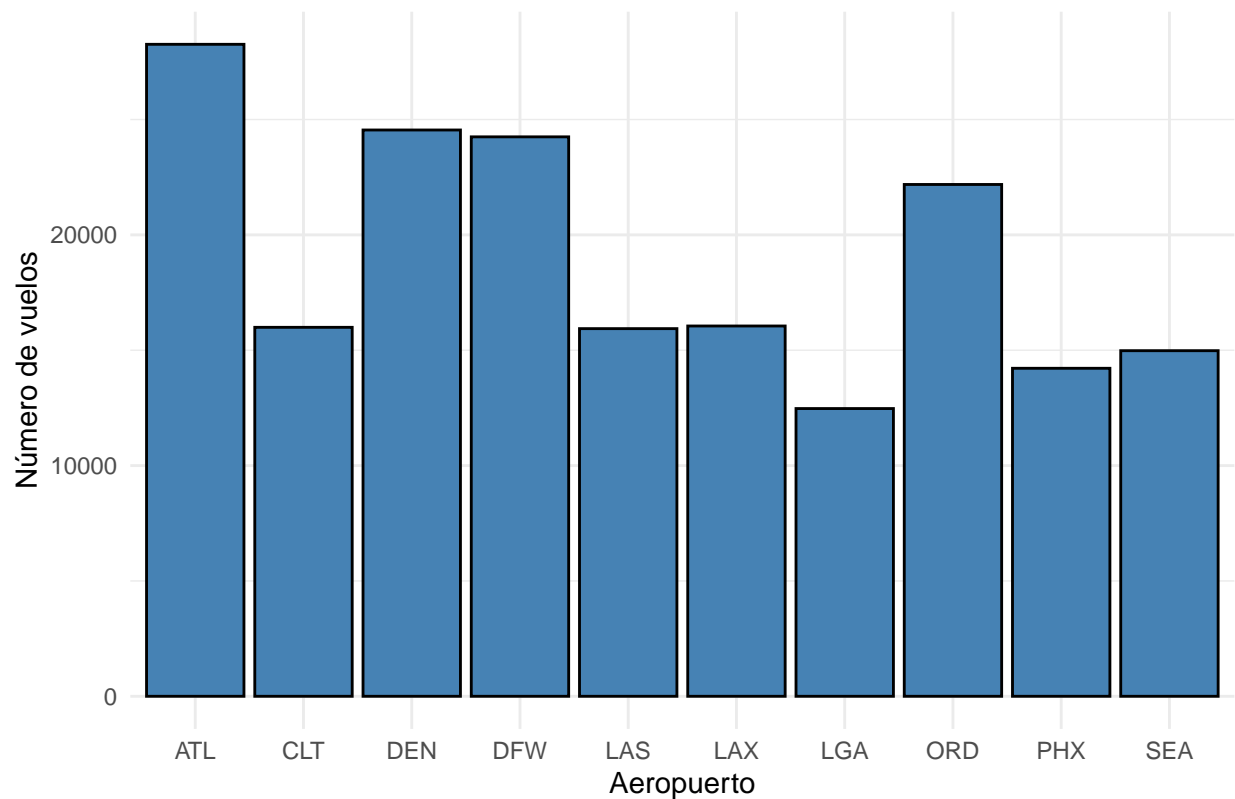


Los aeropuertos de Atlanta (ATL), Denver (DEN) y Dallas Fort-Worth (DFW) son los más concurridos en términos de volumen de vuelos.

```
top_airports <- flightData %>%
  group_by(DEST) %>%
  summarise(FlightCount = n()) %>%
  arrange(desc(FlightCount)) %>%
  top_n(10, FlightCount)

ggplot(top_airports, aes(x = DEST, y = FlightCount)) +
  geom_bar(stat = "identity", fill = "steelblue", color = "black") +
  theme_minimal() +
  labs(x = "Aeropuerto", y = "Número de vuelos",
       title = "Los 10 aeropuertos de llegada con mayor número de vuelos")
```

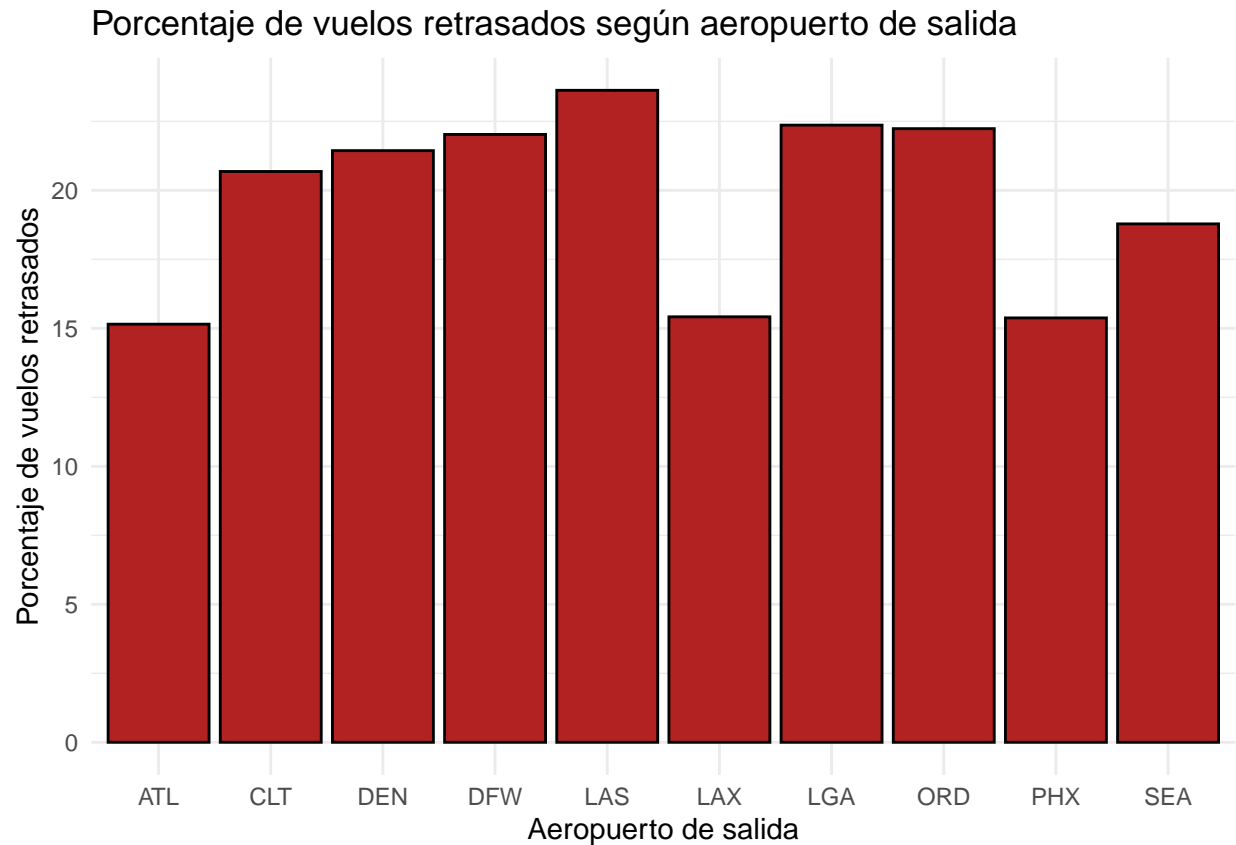
Los 10 aeropuertos de llegada con mayor número de vuelos



Como era de esperar, ATL, DEN y DFW también son los aeropuertos con más vuelos de llegada.

```
percentage_delays <- flightData %>%
  group_by(ORIGIN) %>%
  summarise(FlightCount = n(),
            Percentage_Delayed = mean(ARR_DEL15, na.rm = TRUE) * 100) %>%
  arrange(desc(FlightCount)) %>%
  top_n(10, FlightCount)

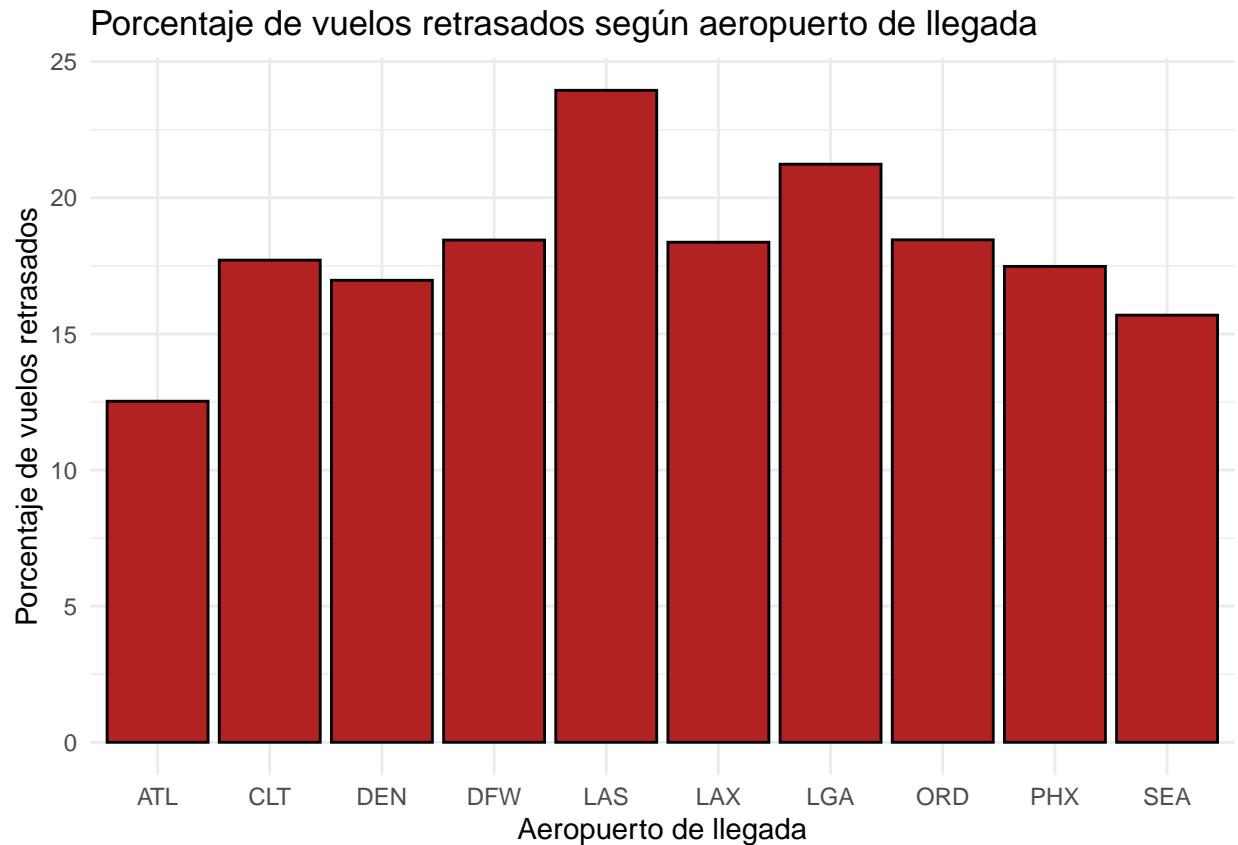
ggplot(permission_delays, aes(x = ORIGIN, y = Percentage_Delayed)) +
  geom_bar(stat = "identity", fill = "firebrick", color = "black") +
  labs(x = "Aeropuerto de salida", y = "Porcentaje de vuelos retrasados",
       title = "Porcentaje de vuelos retrasados según aeropuerto de salida") +
  theme_minimal()
```



Entre los diez aeropuertos con más vuelos de salida, Atlanta (ATL), Los Ángeles (LAX) y Phoenix (PHX) tienen los menores porcentajes de retrasos, mientras que Las Vegas (LAS) registra el mayor porcentaje.

```
percentage_delays <- flightData %>%
  group_by(DEST) %>%
  summarise(FlightCount = n(),
            Percentage_Delayed = mean(ARR_DEL15, na.rm = TRUE) * 100) %>%
  arrange(desc(FlightCount)) %>%
  top_n(10, FlightCount)

ggplot(permission_delays, aes(x = DEST, y = Percentage_Delayed)) +
  geom_bar(stat = "identity", fill = "firebrick", color = "black") +
  labs(x = "Aeropuerto de llegada", y = "Porcentaje de vuelos retrasados",
       title = "Porcentaje de vuelos retrasados según aeropuerto de llegada") +
  theme_minimal()
```

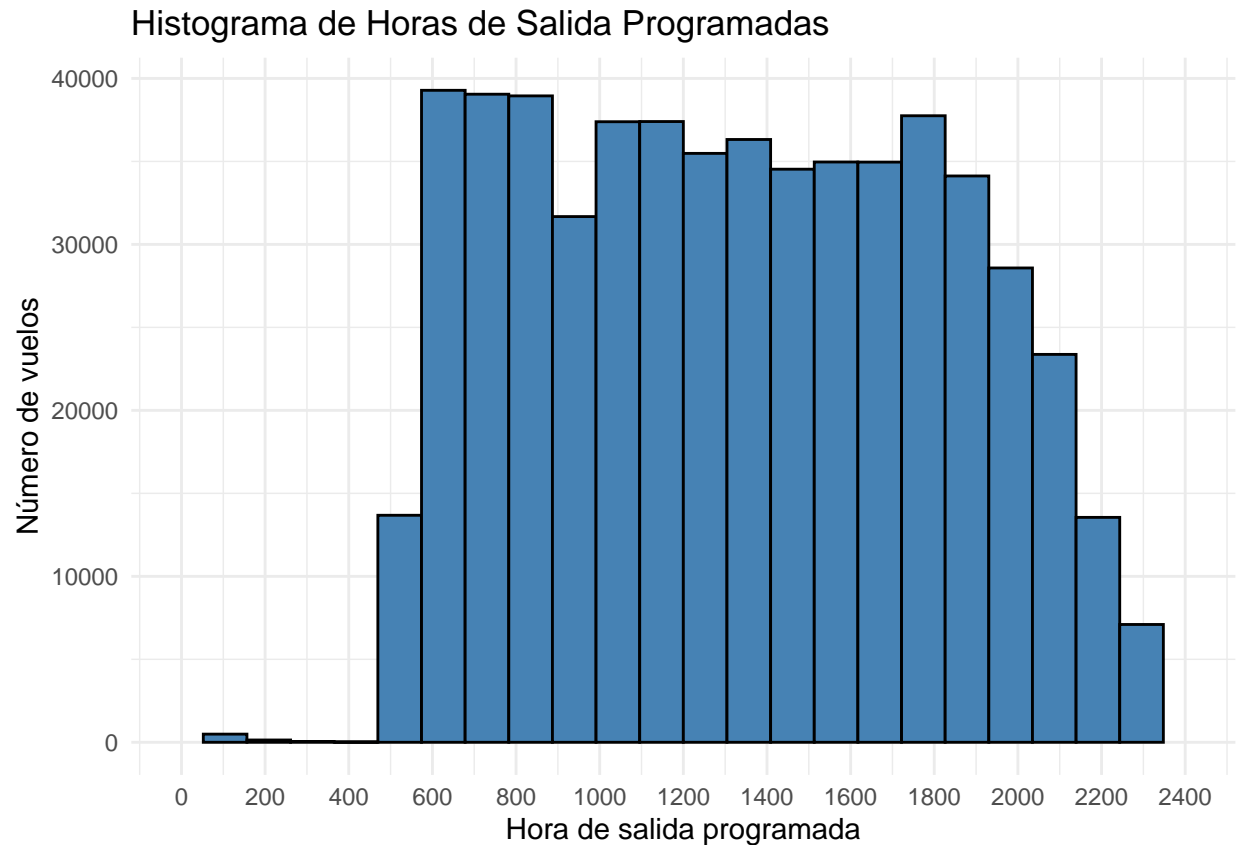


En el lado de los aeropuertos de llegada, observamos algunas similitudes: Los Ángeles (LAX) sigue presentando el mayor porcentaje de retrasos, y Atlanta (ATL) sigue destacando por su menor tasa de retrasos. Sin embargo, los aeropuertos LAX y PHX pierden su posición entre los aeropuertos más eficientes.

Análisis por hora programada de salida y llegada (parte 1)

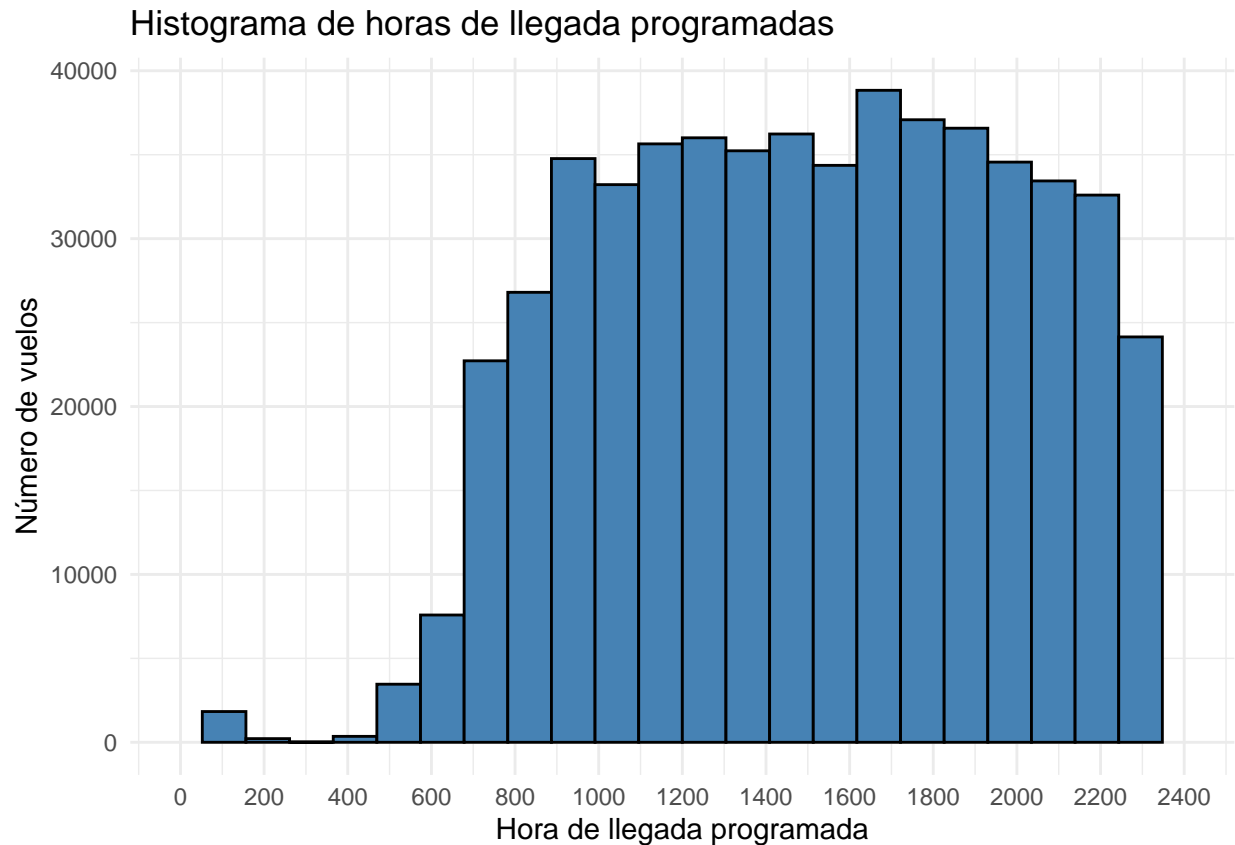
Pasamos a mirar las variables continuas.

```
ggplot(flightData, aes(x = CRS_DEP_TIME)) +  
  geom_histogram(bins = 24, fill = "steelblue", color = "black") +  
  scale_x_continuous(breaks = seq(0, 2400, by = 200), limits = c(0, 2400)) +  
  labs(x = "Hora de salida programada",  
       y = "Número de vuelos",  
       title = "Histograma de Horas de Salida Programadas") +  
  theme_minimal()
```



El análisis de las horas de salida programadas revela que los vuelos son mucho menos frecuentes durante la madrugada (00:00–05:00).

```
ggplot(flightData, aes(x = CRS_ARR_TIME)) +  
  geom_histogram(bins = 24, fill = "steelblue", color = "black") +  
  scale_x_continuous(breaks = seq(0, 2400, by = 200), limits = c(0, 2400)) +  
  labs(x = "Hora de llegada programada",  
       y = "Número de vuelos",  
       title = "Histograma de horas de llegada programadas") +  
  theme_minimal()
```

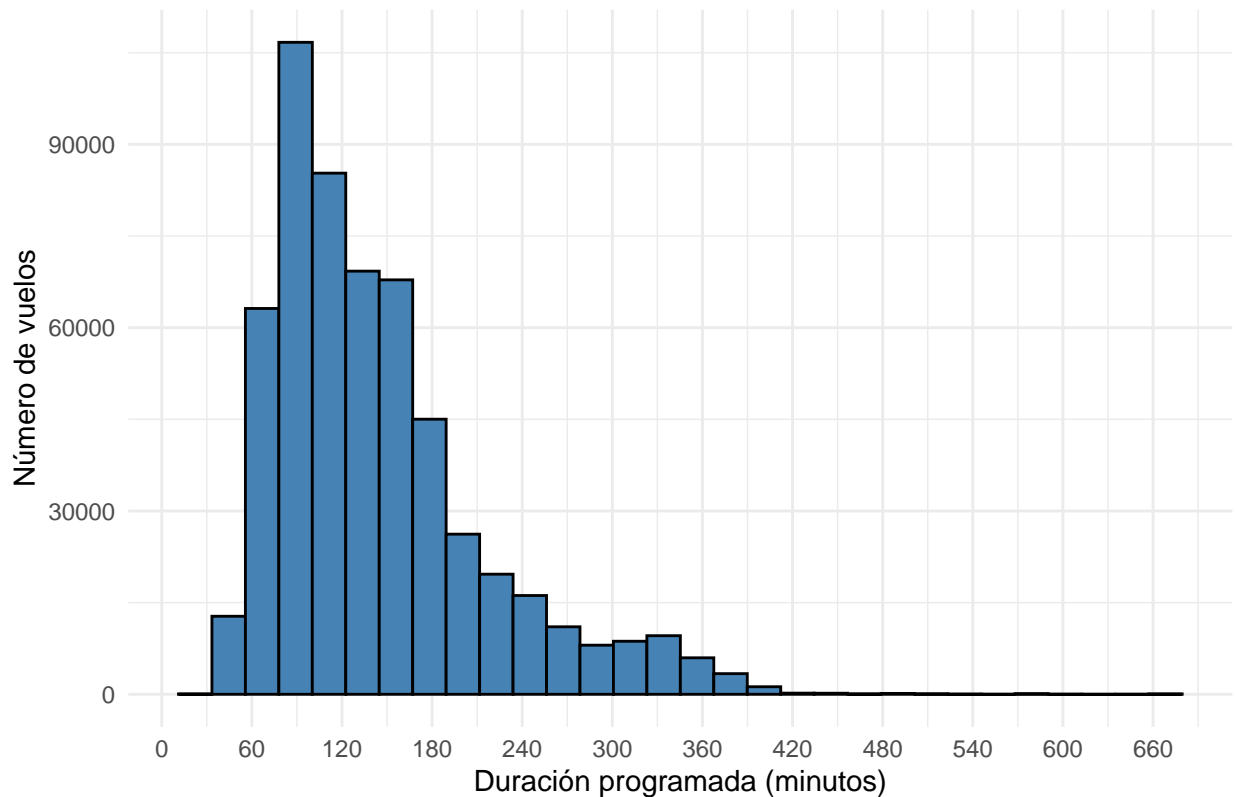



Similar al patrón de salidas, los vuelos de llegada también son menos frecuentes en horas tempranas, pero con una distribución que tiende a centrarse más tarde en el día.

Análisis por duración programada (parte 1)

```
ggplot(flightData, aes(x = CRS_ELAPSED_TIME)) +  
  geom_histogram(bins = 30, fill = "steelblue", color = "black") +  
  scale_x_continuous(breaks = seq(0, max(flightData$CRS_ELAPSED_TIME, na.rm = TRUE), by = 60)) +  
  labs(x = "Duración programada (minutos)",  
       y = "Número de vuelos",  
       title = "Histograma de duración programada de vuelo") +  
  theme_minimal()
```

Histograma de duración programada de vuelo



La duración programada de la mayoría de los vuelos oscila entre 1 y 7 horas (60 a 420 minutos). Sin embargo, se observan algunos valores atípicos, especialmente vuelos con duraciones superiores a 11 horas (660 minutos).

```
flightData %>%
  arrange(desc(CRS_ELAPSED_TIME)) %>%
  head(10)
```

##	DAY_OF_WEEK	FL_DATE	OP_UNIQUE_CARRIER	ORIGIN	DEST	CRS_DEP_TIME	DEP_TIME
## 1	lunes	2023-09-04	Hawaiian Airlines	BOS	HNL	800	1033
## 2	lunes	2023-09-11	Hawaiian Airlines	BOS	HNL	800	807
## 3	lunes	2023-09-18	Hawaiian Airlines	BOS	HNL	800	759
## 4	lunes	2023-09-25	Hawaiian Airlines	BOS	HNL	800	759
## 5	martes	2023-09-05	Hawaiian Airlines	BOS	HNL	800	803
## 6	martes	2023-09-12	Hawaiian Airlines	BOS	HNL	800	102
## 7	martes	2023-09-19	Hawaiian Airlines	BOS	HNL	800	831
## 8	martes	2023-09-26	Hawaiian Airlines	BOS	HNL	800	827
## 9	jueves	2023-09-07	Hawaiian Airlines	BOS	HNL	800	757
## 10	jueves	2023-09-14	Hawaiian Airlines	BOS	HNL	800	803

##	DEP_DELAY_NEW	CRS_ARR_TIME	ARR_TIME	ARR_DELAY_NEW	ARR_DEL15	CRS_ELAPSED_TIME
## 1	153	1310	1533	143	1	670
## 2	7	1310	1337	27	1	670
## 3	0	1310	1239	0	0	670
## 4	0	1310	1301	0	0	670
## 5	3	1310	1309	0	0	670
## 6	1022	1310	532	982	1	670

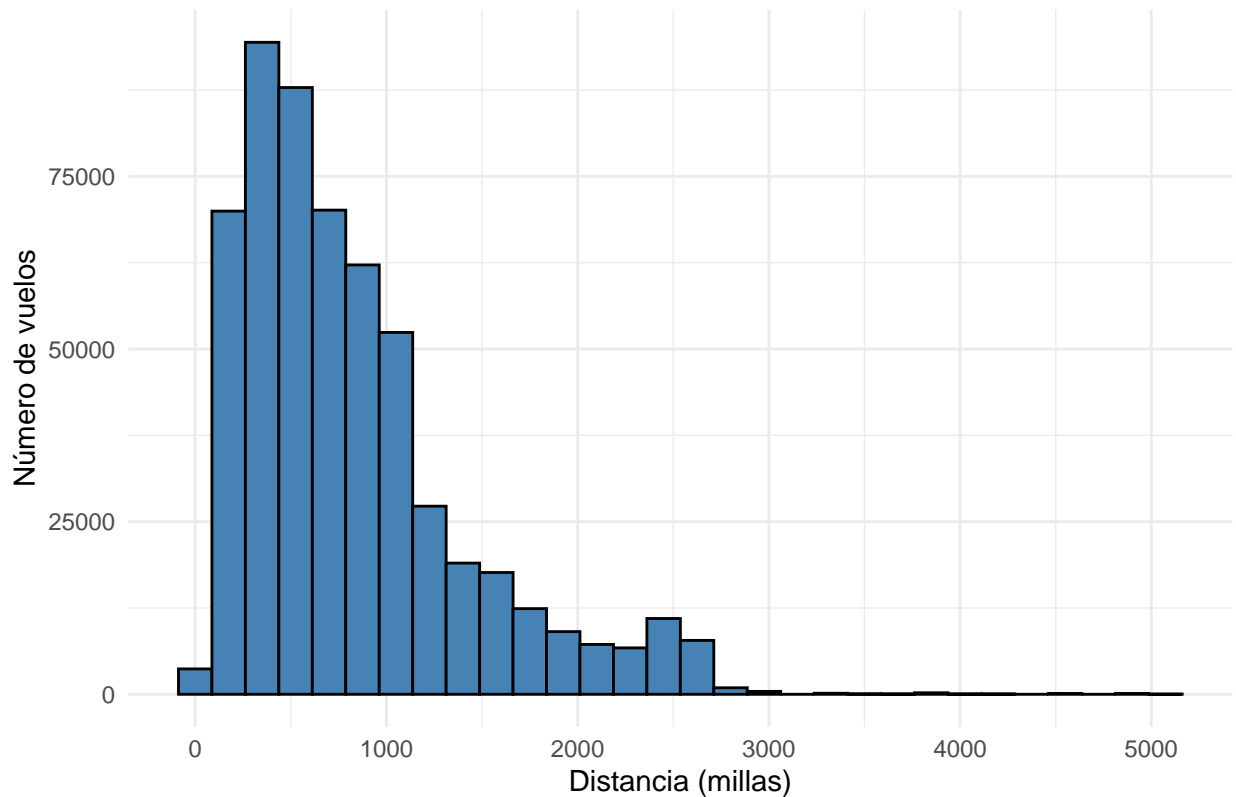
## 7	31	1310	1259	0	0	670
## 8	27	1310	1230	0	0	670
## 9	0	1310	1255	0	0	670
## 10	3	1310	1257	0	0	670
##	ACTUAL_ELAPSED_TIME	DISTANCE				
## 1	660	5095				
## 2	690	5095				
## 3	640	5095				
## 4	662	5095				
## 5	666	5095				
## 6	630	5095				
## 7	628	5095				
## 8	603	5095				
## 9	658	5095				
## 10	654	5095				

Al investigar estos valores atípicos, se descubre que corresponden a vuelos de larga distancia, como los de Boston a Hawaii, justificando así su extensa duración.

Análisis por distancia

```
ggplot(flightData, aes(x = DISTANCE)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "black") +
  labs(x = "Distancia (millas)",
       y = "Número de vuelos",
       title = "Histograma de distancia de vuelo") +
  theme_minimal()
```

Histograma de distancia de vuelo



El análisis de la distancia de los vuelos muestra un patrón similar al de la duración. Los valores extremos, nuevamente, corresponden principalmente a rutas de larga distancia hacia destinos como Hawaii.

Discretización de atributos

Para mejorar la flexibilidad en la fase de modelización y obtener insights más detallados, decidimos discretizar algunas variables.

Discretización de horas programadas de salida y llegada

Convertimos las horas programadas de salida y llegada en categorías discretas. Esto se hace dividiendo el día en intervalos de una hora, como “00:00-00:59”, “01:00-01:59”, etc.

```
breaks <- seq(from = 0, to = 2400, by = 100)

labels <- c("00:00-00:59", "01:00-01:59", "02:00-02:59", "03:00-03:59",
            "04:00-04:59", "05:00-05:59", "06:00-06:59", "07:00-07:59",
            "08:00-08:59", "09:00-09:59", "10:00-10:59", "11:00-11:59",
            "12:00-12:59", "13:00-13:59", "14:00-14:59", "15:00-15:59",
```

```

      "16:00-16:59", "17:00-17:59", "18:00-18:59", "19:00-19:59",
      "20:00-20:59", "21:00-21:59", "22:00-22:59", "23:00-23:59")

flightData$CRS_DEP_TIME_DISCRETE <- cut(flightData$CRS_DEP_TIME,
                                         breaks = breaks,
                                         include.lowest = TRUE,
                                         labels = labels)
flightData$CRS_ARR_TIME_DISCRETE <- cut(flightData$CRS_ARR_TIME,
                                         breaks = breaks,
                                         include.lowest = TRUE,
                                         labels = labels)

levels(flightData$CRS_DEP_TIME_DISCRETE)

## [1] "00:00-00:59" "01:00-01:59" "02:00-02:59" "03:00-03:59" "04:00-04:59"
## [6] "05:00-05:59" "06:00-06:59" "07:00-07:59" "08:00-08:59" "09:00-09:59"
## [11] "10:00-10:59" "11:00-11:59" "12:00-12:59" "13:00-13:59" "14:00-14:59"
## [16] "15:00-15:59" "16:00-16:59" "17:00-17:59" "18:00-18:59" "19:00-19:59"
## [21] "20:00-20:59" "21:00-21:59" "22:00-22:59" "23:00-23:59"

```

```
levels(flightData$CRS_ARR_TIME_DISCRETE)
```

```

## [1] "00:00-00:59" "01:00-01:59" "02:00-02:59" "03:00-03:59" "04:00-04:59"
## [6] "05:00-05:59" "06:00-06:59" "07:00-07:59" "08:00-08:59" "09:00-09:59"
## [11] "10:00-10:59" "11:00-11:59" "12:00-12:59" "13:00-13:59" "14:00-14:59"
## [16] "15:00-15:59" "16:00-16:59" "17:00-17:59" "18:00-18:59" "19:00-19:59"
## [21] "20:00-20:59" "21:00-21:59" "22:00-22:59" "23:00-23:59"

```

Análisis por hora programada de salida y llegada (parte 2)

A continuación, analizamos el porcentaje de vuelos retrasados en cada franja horaria.

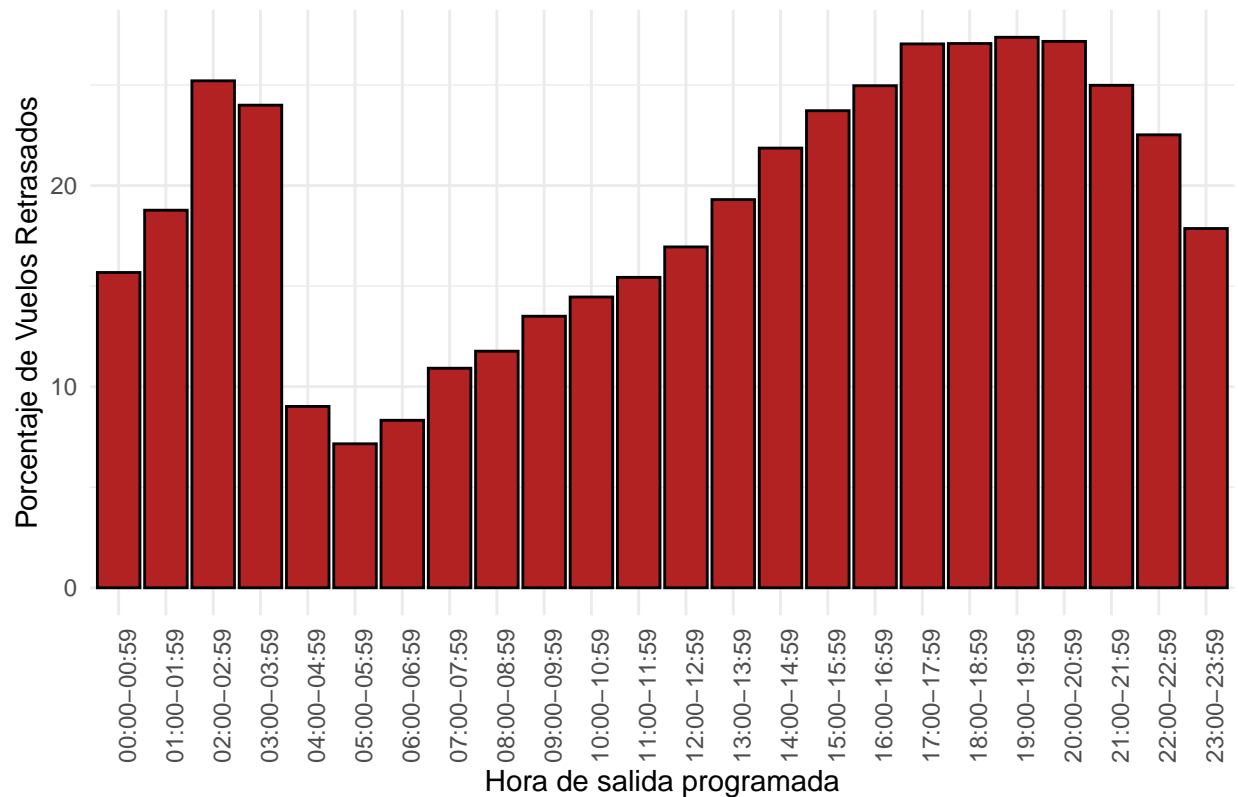
```

percentage_delays <- flightData %>%
  group_by(CRS_DEP_TIME_DISCRETE) %>%
  summarise(Percentage_Delayed = mean(ARR_DEL15) * 100)

ggplot(permission_delays, aes(x = CRS_DEP_TIME_DISCRETE, y = Percentage_Delayed)) +
  geom_bar(stat = "identity", fill = "firebrick", color = "black") +
  labs(x = "Hora de salida programada", y = "Porcentaje de Vuelos Retrasados",
       title = "Porcentaje de vuelos retrasados por hora de salida programada") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90))

```

Porcentaje de vuelos retrasados por hora de salida programada



Al estudiar el porcentaje de vuelos retrasados por cada franja horaria, observamos patrones interesantes:

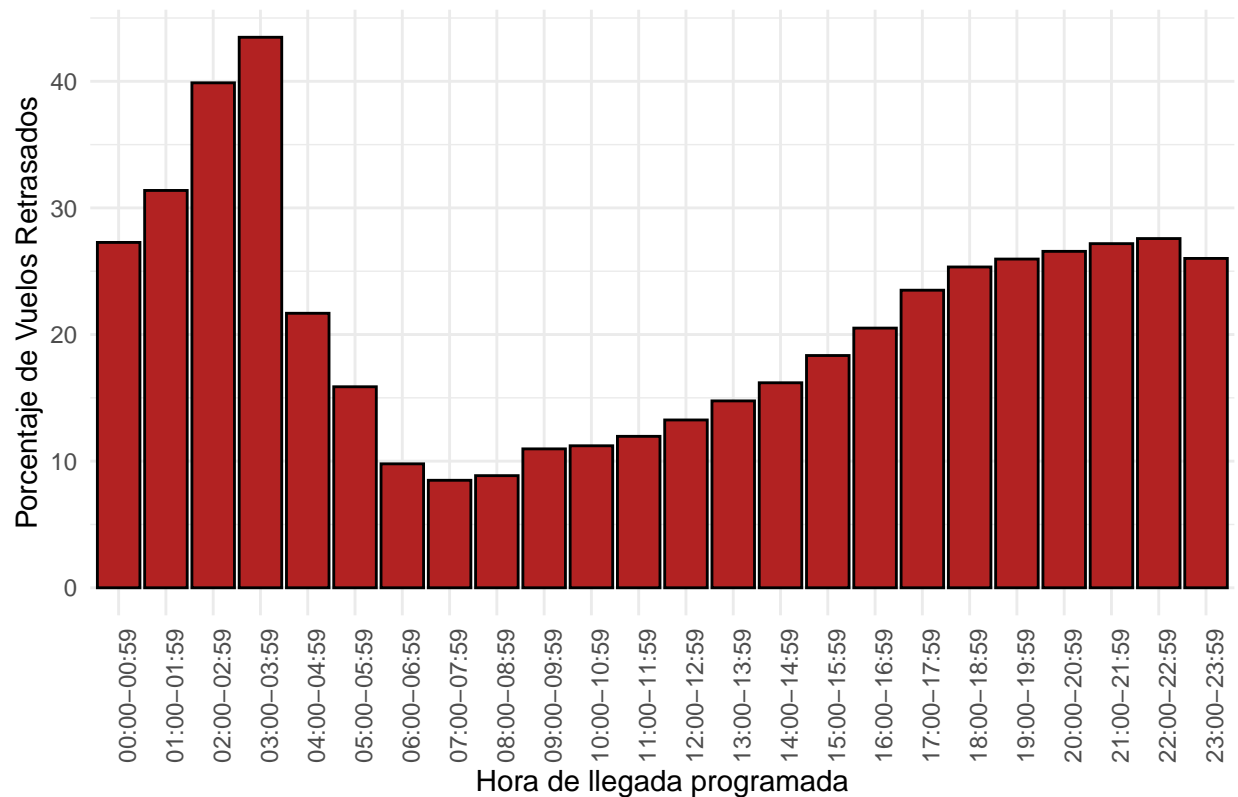
- Los vuelos que salen entre las 05:00 y 05:59 presentan el menor retraso (7.2 %), lo que podría atribuirse a una menor congestión aérea en las primeras horas del día.
- Por el contrario, los vuelos con salida entre las 19:00 y 19:59 registran el mayor porcentaje de retrasos (27.4 %). Esto se podría explicar por el efecto acumulativo de retrasos anteriores durante el día.

Curiosamente, hay un aumento inesperado en los retrasos entre la 01:00 y 03:59, lo que podría deberse a la menor cantidad de vuelos y, por tanto, a una muestra más pequeña que podría distorsionar las estadísticas.

```
percentage_delays <- flightData %>%
  group_by(CRS_ARR_TIME_DISCRETE) %>%
  summarise(Percentage_Delayed = mean(ARR_DEL15) * 100)

ggplot(permission_delays, aes(x = CRS_ARR_TIME_DISCRETE, y = Percentage_Delayed)) +
  geom_bar(stat = "identity", fill = "firebrick", color = "black") +
  labs(x = "Hora de llegada programada", y = "Porcentaje de Vuelos Retrasados",
       title = "Porcentaje de vuelos retrasados por hora de llegada programada") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90))
```

Porcentaje de vuelos retrasados por hora de llegada programada



Los patrones de retraso en las horas de llegada son similares a los de las salidas, con un incremento general a lo largo del día. Los vuelos que llegan entre las 07:00 y 07:59 tienen menos retrasos (8.5 %), mientras que aquellos programados para llegar entre las 03:00 y 03:59 muestran un elevado porcentaje de retrasos (43.5 %), lo que podría estar influenciado por el mismo factor de muestra reducida durante las horas nocturnas.

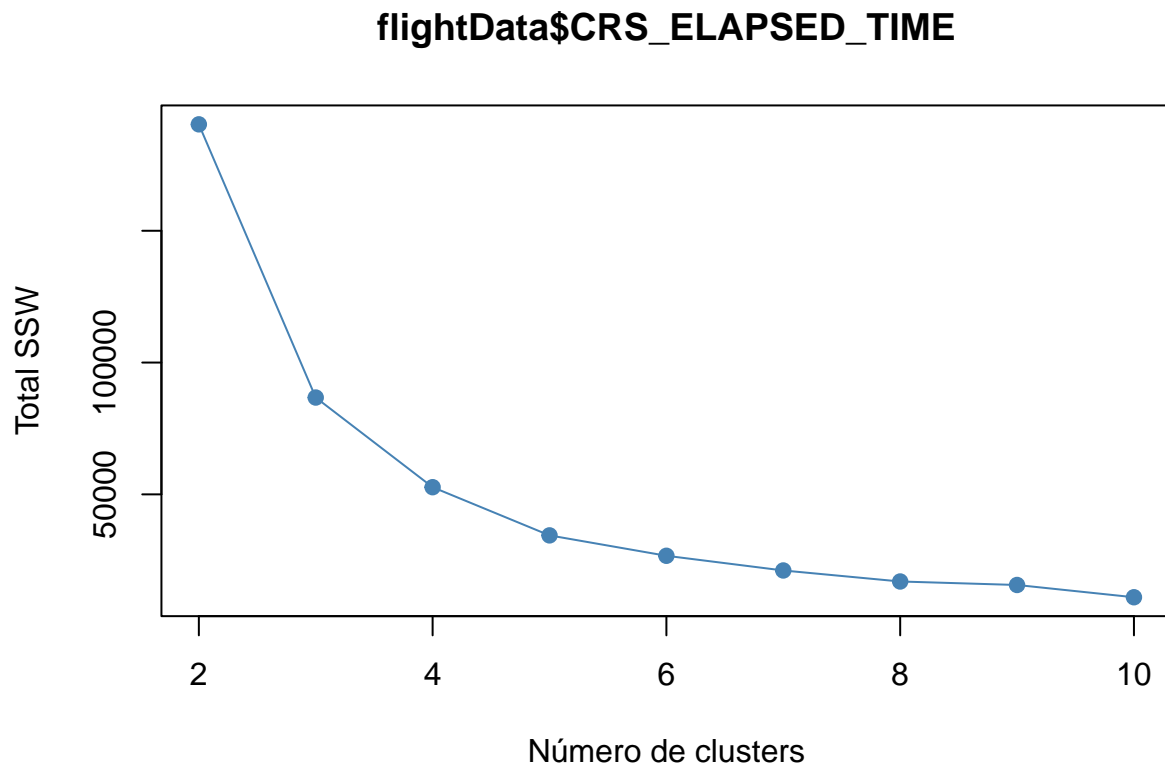
Discretización de duración del vuelo

Para discretizar la variable `CRS_ELAPSED_TIME`, empleamos el método de *k*-means.

```
resultados <- rep(0, 10)

for (i in 2:10) {
  set.seed(123)
  fit <- kmeans(scale(flightData$CRS_ELAPSED_TIME), centers = i)
  resultados[i] <- fit$tot.withinss
}

plot(2:10, resultados[2:10], type = "o", col = "steelblue", pch = 19,
     xlab = "Número de clusters", ylab = "Total SSW", main = "flightData$CRS_ELAPSED_TIME")
```



Tras analizar diferentes números de clústeres, determinamos que el óptimo parece estar entre 4 y 5, ya que es en este punto donde la curva comienza a estabilizarse.

```
suppressPackageStartupMessages(library(arules))

set.seed(123)
flightData$CRS_ELAPSED_TIME_DISCRETE <- discretize(flightData$CRS_ELAPSED_TIME,
                                                    method = "cluster",
                                                    breaks = 5)

summary(flightData$CRS_ELAPSED_TIME_DISCRETE)
```

```
## [24,98.7) [98.7,144) [144,204) [204,289) [289,670]
##      172873      161473      134859      57835      33847
```

k-means ha formado 5 clústeres para la duración programada de los vuelos:

- Vuelo muy cortos: 24-99 minutos
- Vuelos cortos: 99-144 minutos
- Vuelos intermedios: 144-204 minutos
- Vuelos largos: 204-289 minutos
- Vuelos muy largos: 289-670 minutos

Análisis por duración programada (parte 2)

Con las categorías definidas, procedemos a analizar cómo la duración del vuelo afecta el porcentaje de retrasos.

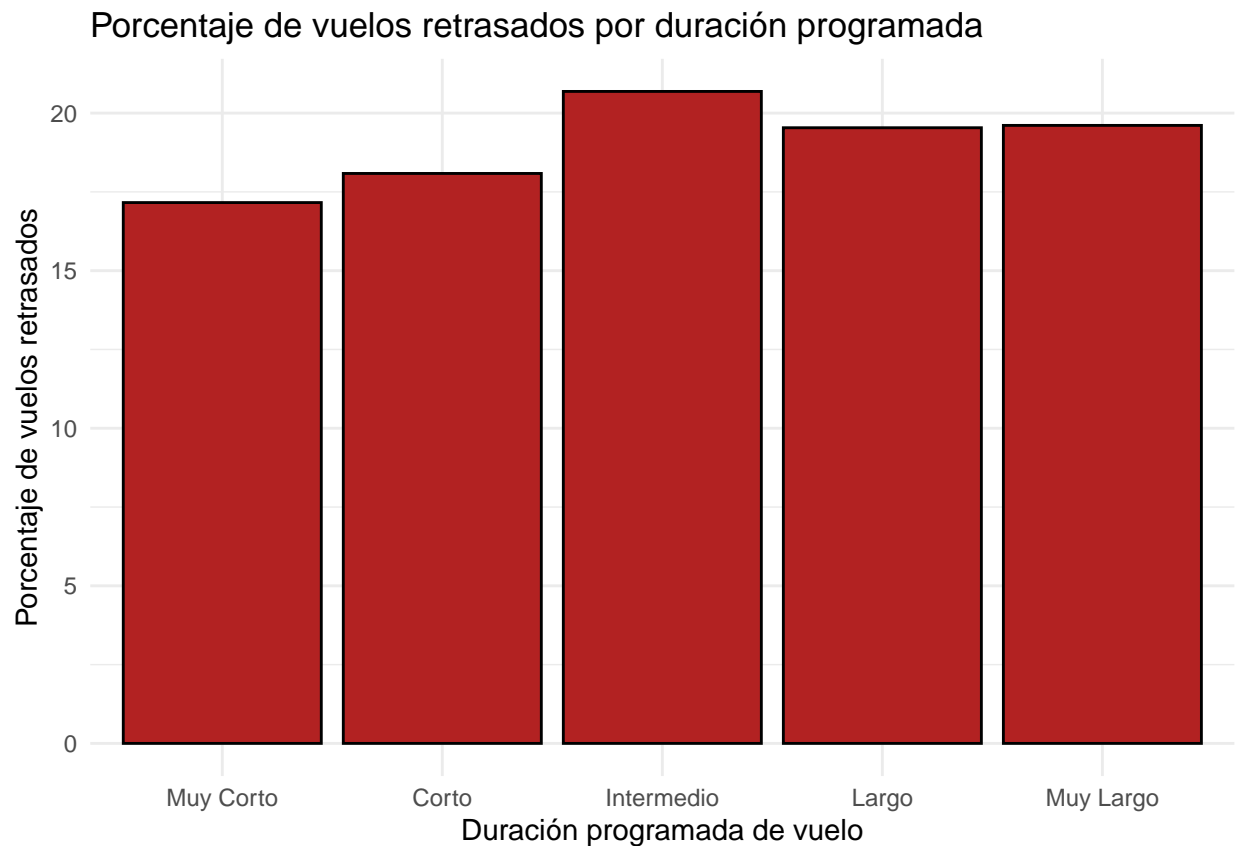

```

levels(flightData$CRS_ELAPSED_TIME_DISCRETE) <- c("Muy Corto", "Corto", "Intermedio", "Largo", "Muy Largo")

percentage_delays <- flightData %>%
  group_by(CRS_ELAPSED_TIME_DISCRETE) %>%
  summarise(Percentage_Delayed = mean(ARR_DEL15) * 100)

ggplot(permission_delays, aes(x = CRS_ELAPSED_TIME_DISCRETE, y = Percentage_Delayed)) +
  geom_bar(stat = "identity", fill = "firebrick", color = "black") +
  labs(x = "Duración programada de vuelo", y = "Porcentaje de vuelos retrasados",
       title = "Porcentaje de vuelos retrasados por duración programada") +
  theme_minimal()

```



Los resultados muestran que hay una variación ligera en los retrasos en función de la duración programada del vuelo. Los vuelos muy cortos y cortos (24-144 minutos) suelen retrasarse menos que los vuelos de mayor duración. Los vuelos de menor duración podrían ser menos susceptibles a factores que causan retrasos, como la congestión en los aeropuertos o las complicaciones logísticas.

Transformación de la hora del día

Para preparar nuestros datos para la fase de modelado, es importante modificar el formato de las columnas que representan la hora del día. Dado que cada hora solo tiene valores de 00 a 59 minutos, dejando un intervalo sin usar entre 60 y 99, transformamos estas horas a un formato continuo en minutos desde medianoche.

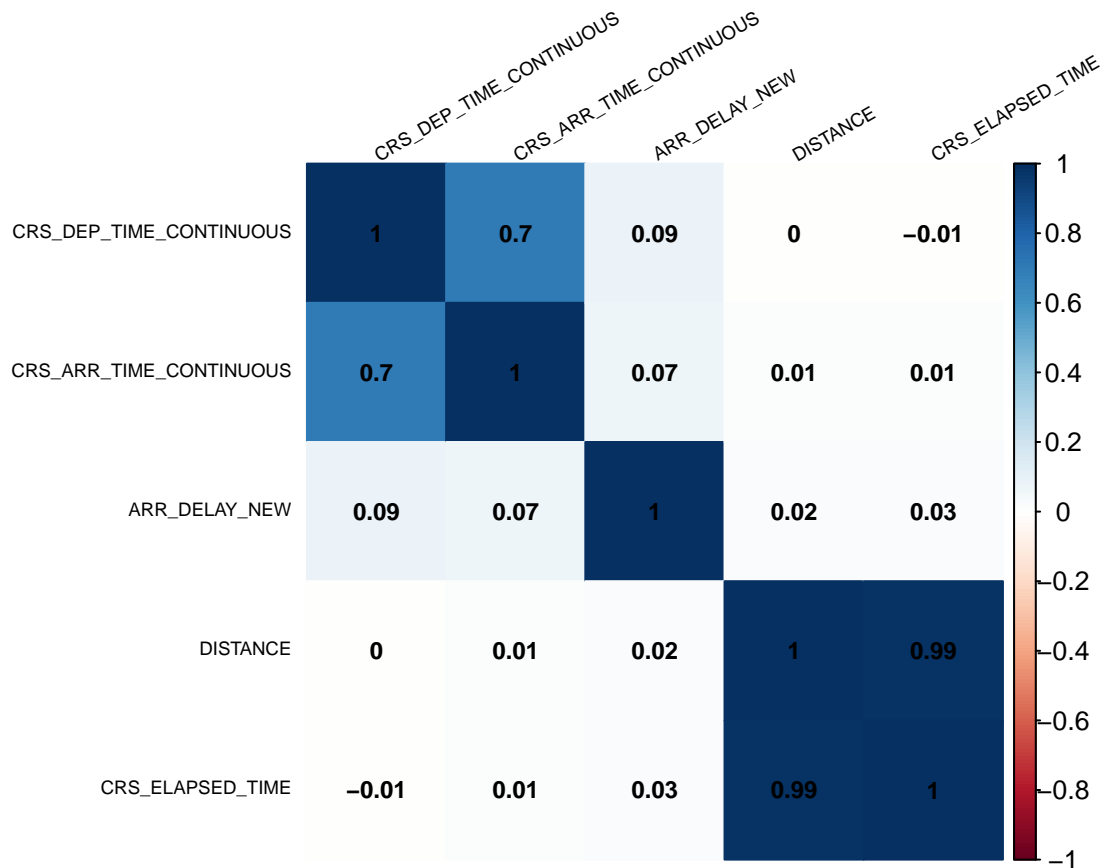
```
convert_time_to_minutes <- function(time) {  
  hours <- time %/% 100  
  minutes <- time %% 100  
  return(hours * 60 + minutes)  
}  
  
flightData$CRS_DEP_TIME_CONTINUOUS <- convert_time_to_minutes(flightData$CRS_DEP_TIME)  
flightData$CRS_ARR_TIME_CONTINUOUS <- convert_time_to_minutes(flightData$CRS_ARR_TIME)  
  
head(flightData[,c("CRS_DEP_TIME_CONTINUOUS", "CRS_ARR_TIME_CONTINUOUS")])
```

```
##   CRS_DEP_TIME_CONTINUOUS CRS_ARR_TIME_CONTINUOUS  
## 1                      414                      540  
## 2                      762                      890  
## 3                     1047                     1178  
## 4                      455                      528  
## 5                      845                      912  
## 6                      370                      539
```

Análisis de correlaciones

Aunque nuestra variable dependiente (ARR_DEL15) es binaria, exploramos cómo las variables numéricas están relacionadas con la duración del retraso (ARR_DELAY_NEW).

```
library(corrplot)  
  
n = c("ARR_DELAY_NEW", "CRS_ELAPSED_TIME", "DISTANCE",  
      "CRS_DEP_TIME_CONTINUOUS", "CRS_ARR_TIME_CONTINUOUS")  
  
factores = flightData %>% select(all_of(n))  
res <- cor(factores)  
corrplot(res, method="color", tl.col="black", tl.srt=30, order="AOE",  
          tl.cex=0.6, number.cex=0.75, sig.level=0.01, addCoef.col="black")
```



Nuestro análisis revela que ninguna de las variables numéricas muestra una correlación lineal fuerte (superior a 0.09) con `ARR_DELAY_NEW`. Aunque esto sugiere la ausencia de una relación lineal directa, no descarta la posibilidad de relaciones no lineales. Anteriormente observamos que el porcentaje de vuelos retrasados varía según la hora de salida y llegada, tendiendo a disminuir por la mañana y aumentar por la tarde.

Por otro lado, encontramos una correlación casi perfecta (0.99) entre `DISTANCE` y `CRS_ELAPSED_TIME`. Esto indica una redundancia en la información proporcionada por estas dos variables. Para evitar la inclusión de datos redundantes y mejorar la precisión de nuestro modelo, optamos por eliminar la variable `DISTANCE` del conjunto de datos.

```
flightData$DISTANCE <- NULL
str(flightData)
```

```
## 'data.frame': 560887 obs. of 19 variables:
## $ DAY_OF_WEEK : Factor w/ 7 levels "lunes","martes",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ FL_DATE : Date, format: "2023-09-04" "2023-09-04" ...
## $ OP_UNIQUE_CARRIER : Factor w/ 15 levels "Endeavor Air",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ ORIGIN : Factor w/ 340 levels "ABE","ABI","ABQ",...: 1 1 1 5 5 12 12 13 15 15 ...
## $ DEST : Factor w/ 340 levels "ABE","ABI","ABQ",...: 21 21 21 21 21 21 21 21 21 99 ...
## $ CRS_DEP_TIME : int 654 1242 1727 735 1405 610 1620 1015 1245 706 ...
## $ DEP_TIME : int 649 1237 1717 725 1400 609 1636 1010 1237 701 ...
## $ DEP_DELAY_NEW : num 0 0 0 0 0 0 16 0 0 0 ...
## $ CRS_ARR_TIME : int 900 1450 1938 848 1512 859 1906 1123 1428 803 ...
## $ ARR_TIME : int 842 1431 1905 820 1454 904 1913 1100 1411 759 ...
## $ ARR_DELAY_NEW : num 0 0 0 0 0 5 7 0 0 0 ...
## $ ARR_DEL15 : num 0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ CRS_ELAPSED_TIME      : num  126 128 131 73 67 109 106 68 103 57 ...
## $ ACTUAL_ELAPSED_TIME   : num   113 114 108 55 54 115 97 50 94 58 ...
## $ CRS_DEP_TIME_DISCRETE : Factor w/ 24 levels "00:00-00:59",...: 7 13 18 8 15 7 17 11 13 8 ...
## $ CRS_ARR_TIME_DISCRETE : Factor w/ 24 levels "00:00-00:59",...: 9 15 20 9 16 9 20 12 15 9 ...
## $ CRS_ELAPSED_TIME_DISCRETE: Factor w/ 5 levels "Muy Corto","Corto",...: 2 2 2 1 1 2 2 1 2 1 ...
##   ..- attr(*, "discretized:breaks")= num [1:6] 24 98.7 143.9 203.6 288.6 ...
##   ..- attr(*, "discretized:method")= chr "cluster"
## $ CRS_DEP_TIME_CONTINUOUS : num   414 762 1047 455 845 ...
## $ CRS_ARR_TIME_CONTINUOUS : num   540 890 1178 528 912 ...
```

Análisis SVD

En nuestro análisis SVD, las limitaciones de hardware nos llevan a seleccionar cuidadosamente nuestras variables. Para evitar la sobrecarga de 680 variables dummy derivadas de 340 aeropuertos de salida y llegada, optamos por incluir solo un subconjunto de variables.

Incorporamos las tres variables numéricas:

1. **CRS_ELAPSED_TIME**: Duración programada del vuelo.
2. **CRS_DEP_TIME_CONTINUOUS**: Hora de salida, convertida a minutos desde medianoche.
3. **CRS_ARR_TIME_CONTINUOUS**: Hora de llegada en el mismo formato.

En cuanto a las categóricas, seleccionamos solo las aerolíneas (**OP_UNIQUE_CARRIER**) y los días de la semana (**DAY_OF_WEEK**), resultando en 15 dummies para las aerolíneas y 7 para los días de la semana.

El conjunto final para SVD incluye estas 25 variables (numéricas y categóricas). Utilizamos **fastDummies** para las variables dummy y **scale** para las numéricas. Combinamos estos datos y aplicamos SVD, buscando descubrir patrones y relaciones significativas en los datos de vuelo.

```
suppressPackageStartupMessages(library(fastDummies))

# Transformamos las columnas categóricas en variables dummy y eliminamos las columnas originales
dummy_data <- flightData %>%
  select(DAY_OF_WEEK, OP_UNIQUE_CARRIER) %>%
  dummy_cols(remove_selected_columns = TRUE)

# Normalizamos las variables numéricas
numeric_data <- flightData %>%
  select(CRS_ELAPSED_TIME, CRS_DEP_TIME_CONTINUOUS, CRS_ARR_TIME_CONTINUOUS) %>%
  scale()

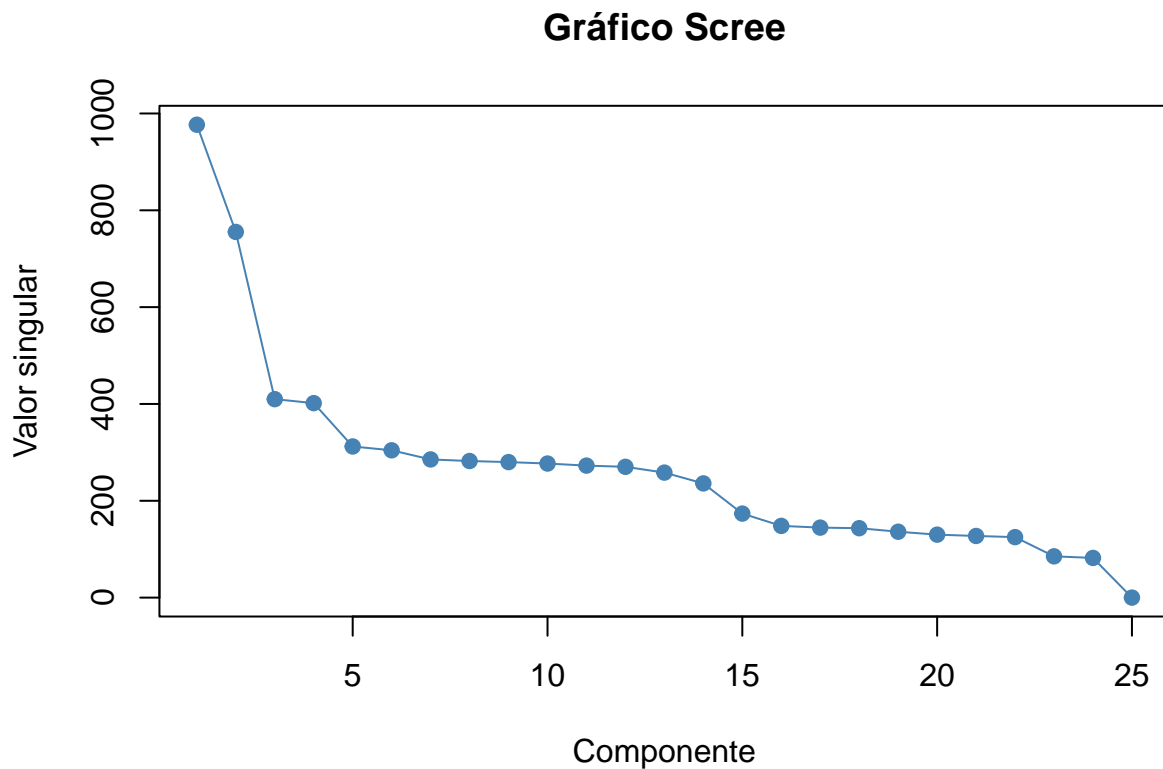
# Unimos los datos numéricos y dummy en un solo conjunto de datos
Z <- cbind(numeric_data, dummy_data)

# Aplicamos SVD al conjunto de datos combinado
svd_results <- svd(Z)

# Obtenemos los valores singulares del resultado de SVD
```

```
singular_values <- svd_results$d

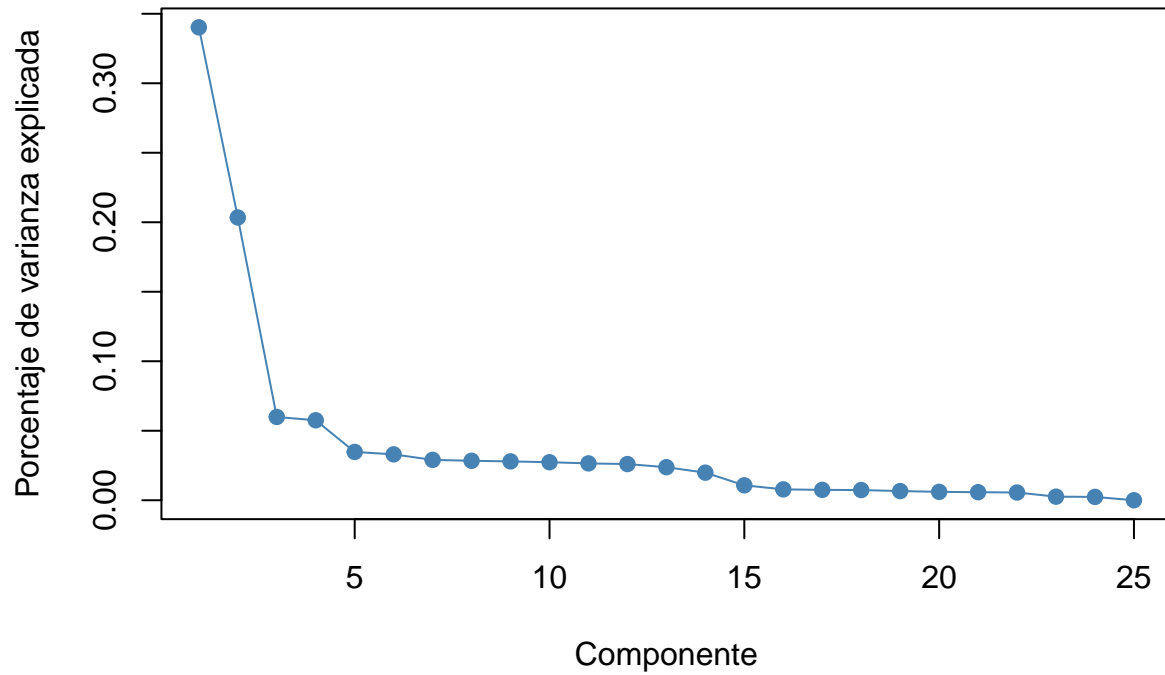
# Mostramos los valores singulares para identificar componentes importantes
plot(singular_values, type = "o", col = "steelblue", pch = 19,
     xlab = "Componente",
     ylab = "Valor singular",
     main = "Gráfico Scree")
```



```
# Determinamos la varianza explicada por cada componente
total_variance <- sum(singular_values^2)
explained_variance <- singular_values^2 / total_variance

# Visualizamos el porcentaje de varianza explicada por cada componente
plot(explained_variance, type = "o", col = "steelblue", pch = 19,
     xlab = "Componente",
     ylab = "Porcentaje de varianza explicada",
     main = "Gráfico de Varianza Explicada")
```

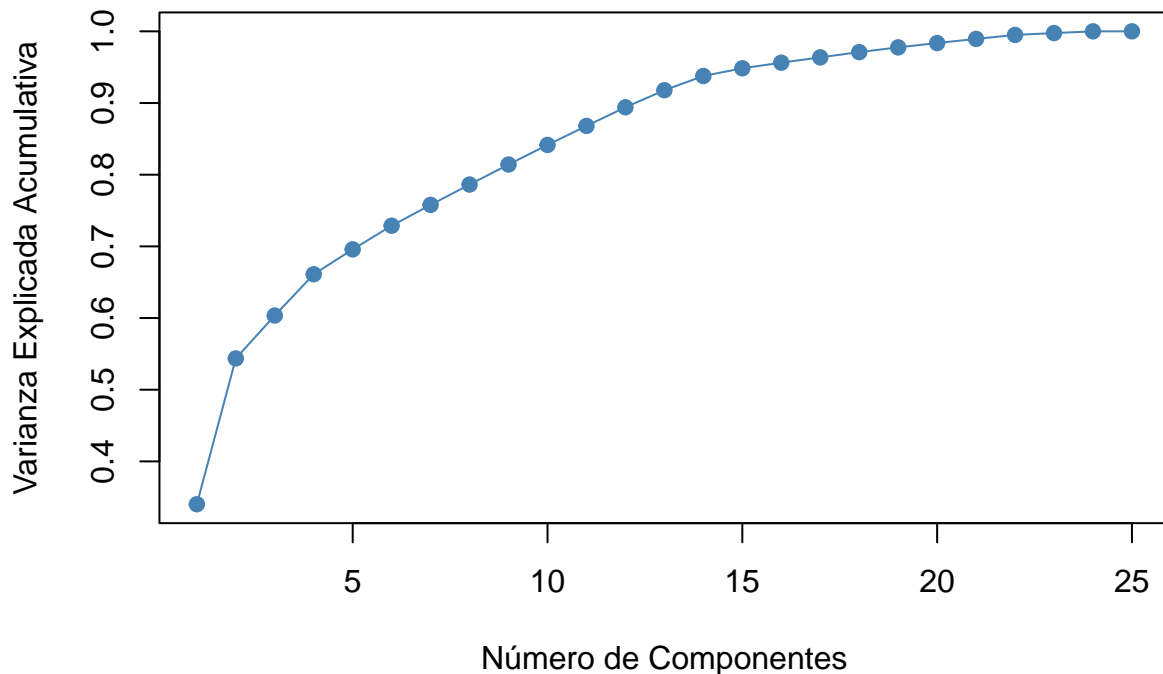
Gráfico de Varianza Explicada



```
# Calculamos la varianza acumulada
cum_explained_variance <- cumsum(singular_values^2) / total_variance

# Mostramos la varianza explicada acumulativa
plot(cum_explained_variance, type = "o", col = "steelblue", pch = 19,
     xlab = "Número de Componentes",
     ylab = "Varianza Explicada Acumulativa",
     main = "Gráfico de Varianza Explicada Acumulativa")
```

Gráfico de Varianza Explicada Acumulativa



En el análisis del gráfico Scree, identificamos un “codo” evidente después de solo 2 componentes, que explican el 54.4% de la varianza total. Sin embargo, para capturar el 90% de la varianza, necesitamos incluir hasta 13 componentes, los cuales explican juntos el 91.8% de la varianza. Esto indica que, utilizando poco más de la mitad de las variables originales, podemos representar más del 90% de la información contenida en los datos.

Análisis de los componentes

La matriz V es una de las tres matrices resultantes de la descomposición SVD. Cada columna de esta matriz representa un componente singular y sus elementos muestran cómo cada variable original contribuye a ese componente. En otras palabras, nos ayuda a comprender la importancia relativa de nuestras variables originales en cada uno de los componentes singulares identificados.

Mostramos los primeros 3 componentes de la matriz para analizar y entender mejor su contribución individual a nuestro conjunto de datos:

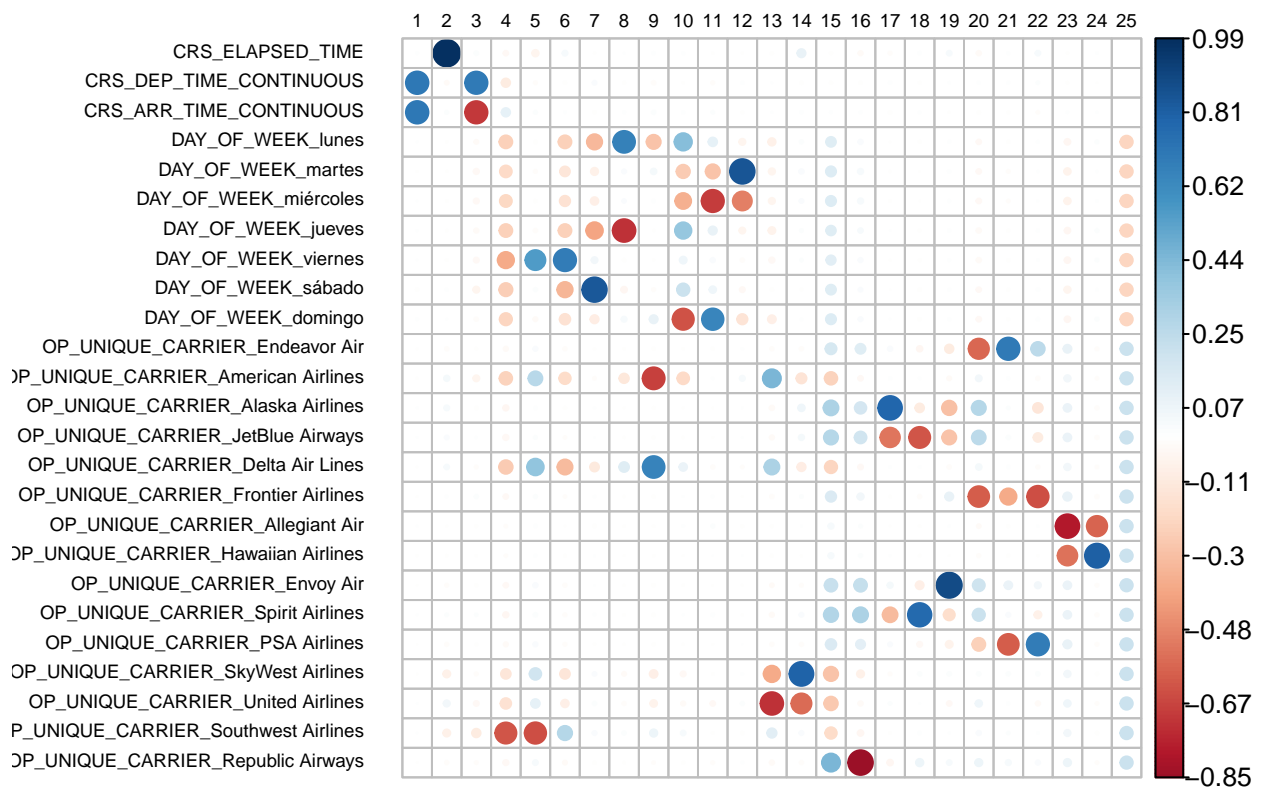
```
V_matrix <- svd_results$v
rownames(V_matrix) <- colnames(Z)
V_matrix[,1:3]
```

##	[,1]	[,2]	[,3]
## CRS_ELAPSED_TIME	4.408865e-03	0.9904341623	0.019563729
## CRS_DEP_TIME_CONTINUOUS	7.070417e-01	-0.0199489208	0.698324100
## CRS_ARR_TIME_CONTINUOUS	7.070765e-01	0.0137610881	-0.698807490
## DAY_OF_WEEK_lunes	1.407418e-03	-0.0026117424	-0.022209914

```
## DAY_OF_WEEK_martes -1.020687e-03 -0.0019490563 -0.039413284
## DAY_OF_WEEK_miércoles 9.283873e-04 -0.0029047553 -0.036470128
## DAY_OF_WEEK_jueves 1.590550e-03 -0.0031473120 -0.026577855
## DAY_OF_WEEK_viernes 7.221105e-04 -0.0034934085 -0.034952167
## DAY_OF_WEEK_sábado -8.569530e-03 0.0071238869 -0.052742301
## DAY_OF_WEEK_domingo 4.870448e-03 0.0010076946 -0.017174330
## OP_UNIQUE_CARRIER_Endavor Air -5.466181e-04 -0.0173703988 -0.004039178
## OP_UNIQUE_CARRIER_American Airlines -1.251267e-03 0.0411476111 -0.051368873
## OP_UNIQUE_CARRIER_Alaska Airlines 1.965524e-03 0.0355451874 -0.009757398
## OP_UNIQUE_CARRIER_JetBlue Airways 5.775564e-04 0.0222105360 0.002500524
## OP_UNIQUE_CARRIER_Delta Air Lines -4.758009e-04 0.0318305895 -0.013160560
## OP_UNIQUE_CARRIER_Frontier Airlines -6.406396e-04 0.0056435151 0.017352744
## OP_UNIQUE_CARRIER_Allegiant Air 8.022591e-04 -0.0007363409 -0.005085856
## OP_UNIQUE_CARRIER_Hawaiian Airlines -1.414108e-03 -0.0028548265 -0.001892994
## OP_UNIQUE_CARRIER_Envoy Air -3.098291e-04 -0.0142784786 -0.007784804
## OP_UNIQUE_CARRIER_Spirit Airlines 9.898533e-04 0.0110818038 0.010013040
## OP_UNIQUE_CARRIER_PSA Airlines -1.779205e-07 -0.0158793424 -0.004462580
## OP_UNIQUE_CARRIER_SkyWest Airlines -2.457338e-04 -0.0694699233 -0.014391186
## OP_UNIQUE_CARRIER_United Airlines 6.524204e-04 0.0543075222 -0.033967331
## OP_UNIQUE_CARRIER_Southwest Airlines 3.839197e-04 -0.0700581207 -0.101495483
## OP_UNIQUE_CARRIER_Republic Airways -5.586614e-04 -0.0170940267 -0.012000044
```

Dada la gran cantidad de variables, un gráfico facilitará la interpretación de estas contribuciones:

```
corrplot(V_matrix, is.cor=FALSE, tl.cex=0.6, tl.srt=0, tl.offset=1, tl.col="black")
```



Interpretación de resultados

- **Componente 1:** Las variables `CRS_DEP_TIME_CONTINUOUS` y `ARR_TIME_CONTINUOUS` dominan este componente, ambos en la misma dirección. Esto sugiere que este componente refleja una relación directa entre las horas de salida y llegada.
- **Componente 2:** Se caracteriza por una influencia predominante de `CRS_ELAPSED_TIME`, lo que sugiere que este componente se enfoca principalmente en la duración del vuelo.
- **Componente 3:** En este caso, `CRS_DEP_TIME_CONTINUOUS` y `ARR_TIME_CONTINUOUS` también son significativos, pero con signos opuestos. Esto puede indicar que el componente está capturando situaciones en las que hay una diferencia significativa entre las horas de salida y llegada, como en los vuelos que salen por la noche (22h00) y llegan por la madrugada del día siguiente (00h30).
- **Del componente 4 en adelante:** Vemos una mayor contribución de variables como los días de la semana y las aerolíneas.

Algunas variables dummy, correspondientes a *Allegiant Air* y *Hawaiian Airlines*, no tienen representación directa hasta el componente 23. Aunque representan una proporción pequeña del total, es posible que nuestro modelo sea sesgado hacia las aerolíneas con mayor número de vuelos.

Conclusiones

Resumen de hallazgos

Nuestro análisis exploratorio ha revelado patrones temporales claros en los retrasos de vuelos. Los vuelos que salen en las primeras horas de la mañana (05:00-05:59) suelen tener menos retrasos, mientras que aquellos en la tarde y noche muestran un aumento significativo. Esta tendencia podría deberse a la acumulación de retrasos a lo largo del día.

Identificamos diferencias notables en los porcentajes de retrasos entre aerolíneas y rutas. Aerolíneas como JetBlue mostraron mayores índices de retraso, mientras que Hawaiian Airlines registró los menores. Los aeropuertos con mayor tráfico, como Atlanta, Denver y Dallas Fort-Worth, también mostraron patrones distintos en términos de retrasos.

Hemos encontrado una correlación casi perfecta entre la distancia y el tiempo estimado del vuelo, lo que nos llevó a eliminar la variable `DISTANCE` para evitar redundancias en nuestro modelo.

Los resultados sugieren que los vuelos más cortos suelen tener menos retrasos en comparación con los vuelos más largos. Esto indica que los vuelos de menor duración podrían ser menos susceptibles a factores que causan retrasos.

Implicaciones para el modelado

El análisis SVD ha reducido la dimensionalidad de nuestro conjunto de datos, permitiéndonos representar la mayoría de la varianza con menos componentes. Esto nos ayudará a construir modelos de clasificación más eficientes y precisos.

La transformación de las horas de vuelo a un formato continuo en minutos desde medianoche facilitará el modelado y permitirá un análisis más preciso de la influencia temporal en los retrasos.

Dado que algunas variables dummy, como las correspondientes a ciertas aerolíneas, no aparecieron en los componentes principales hasta muy tarde en el análisis SVD, debemos ser conscientes de posibles sesgos en nuestro modelo hacia aerolíneas con mayor número de vuelos.

Bibliografía

Bureau of Transportation Statistics. (2023). *TranStats*. Recuperado de https://www.transtats.bts.gov/DL_SelectFields.aspx?gnoyr_VQ=FGJ&QO_fu146_anzr=b0-gvzr

GeeksforGeeks. (2023). *Dummy Variables in R Programming*. Recuperado de <https://www.geeksforgeeks.org/dummy-variables-in-r-programming/>

Gironés Roig, J. (2023). *Gestión de características*. Universitat Oberta de Catalunya.

Gironés Roig, J. (2023). *Modelos no supervisados*. Universitat Oberta de Catalunya.

Kaplan, J. & Schlegel, B. (2023). *fastDummies: Fast Creation of Dummy (Binary) Columns and Rows from Categorical Variables* (Versión 1.7.1). Recuperado de <https://jacobkap.github.io/fastDummies/>

Montoliu Colás, R. (2023). *Evaluación de modelos*. Universitat Oberta de Catalunya.

Montoliu Colás, R. (2023). *Preprocesado de datos*. Universitat Oberta de Catalunya.

Peng, R. D. (2023). *Dimension Reduction*. En *Exploratory Data Analysis*. Recuperado de <https://bookdown.org/rdpeng/exdata/dimension-reduction.html>

R Core Team. (2023). *Documentación sobre SVD*. Recuperado de <https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/svd>

```
write.csv(flightData, '../data/flightData_clean.csv', row.names = FALSE)
```