# Intro Sparklyr

*Favio Vázquez (Total rip-off de Edgar Ruiz)*

*25 de octubre de 2018*

El paquete de R llamado sparklyr facilita el aprendizaje mediante un instalador de Spark que se puede usar dentro de una computadora personal, incluyendo Windows. En este taller los participantes aprenderán a utilizar Spark por medio de R mediante el uso de diferentes técnicas y funciones para:

- Transformar datos
- Crear modelos estadísticos
- Programar canales de datos.

```
install.packages("sparklyr")
install.packages("tidyverse")
install.packages("dbplot")
install.packages("nycflights13")
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------- tidyverse 1.2.1 --
```

```
## ✓ ggplot2 3.0.0     ✓ purrr   0.2.5
## ✓ tibble  1.4.2     ✓ dplyr   0.7.6
## ✓ tidyr   0.8.1     ✓ stringr 1.3.0
## ✓ readr   1.1.1     ✓ forcats 0.3.0
```

```
## -- Conflicts ------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(nycflights13)
library(sparklyr)
```

```
##
## Attaching package: 'sparklyr'
```

```
## The following object is masked from 'package:purrr':
##
##     invoke
```

```
spark_install("2.3.1")
```

```
sc <- spark_connect(master = "local", version = "2.3.1")
```

```
vuelos <- sdf_copy_to(sc, flights)
```

```
vuelos %>%
  tally()
```

```
## # Source: spark<?> [?? x 1]
##         n
## *   <dbl>
## 1 336776.
```

```
vuelos %>%
  group_by(origin) %>%
```

```
  tally()
```

```
## # Source: spark<?> [?? x 2]
##   origin       n
## * <chr>    <dbl>
## 1 JFK    111279.
## 2 LGA    104662.
## 3 EWR    120835.
```

```
vuelos %>%
  group_by(origin) %>%
  summarise(
    promedio_tarde = mean(dep_delay, na.rm = TRUE)
    )
```

```
## # Source: spark<?> [?? x 2]
##   origin promedio_tarde
## * <chr>           <dbl>
## 1 JFK              12.1
## 2 LGA              10.3
## 3 EWR              15.1
```

```
vuelos %>%
  ft_binarizer(
    input_col = "arr_delay",
    output_col = "tarde",
    threshold = 15
  ) %>%
  select(
    arr_delay,
    tarde
  )
```

```
## # Source: spark<?> [?? x 2]
##    arr_delay tarde
##  *     <dbl> <dbl>
##  1       11.    0.
##  2       20.    1.
##  3       33.    1.
##  4      -18.    0.
##  5      -25.    0.
##  6       12.    0.
##  7       19.    1.
##  8      -14.    0.
##  9       -8.    0.
## 10        8.    0.
## # ... with more rows
```

```
vuelos %>%
  mutate(sched_dep_time = as.numeric(sched_dep_time)) %>%
  ft_bucketizer(
    input_col  = "sched_dep_time",
    output_col = "hora",
    splits = c(0, 400, 800, 1200, 1600, 2000, 2400)
  ) %>%
  select(
```

```
    sched_dep_time,
    hora
  )
```

```
## # Source: spark<?> [?? x 2]
##    sched_dep_time  hora
##  *          <dbl> <dbl>
##  1           515.    1.
##  2           529.    1.
##  3           540.    1.
##  4           545.    1.
##  5           600.    1.
##  6           558.    1.
##  7           600.    1.
##  8           600.    1.
##  9           600.    1.
## 10           600.    1.
## # ... with more rows
```

```
vuelos %>%
  mutate(sched_dep_time = as.numeric(sched_dep_time)) %>%
  ft_bucketizer(
    input_col  = "sched_dep_time",
    output_col = "hora",
    splits = c(0, 800, 1200, 1600, 2000, 2400)
  ) %>%
  group_by(hora) %>%
  tally() %>%
  arrange(hora)
```

```
## # Source:     spark<?> [?? x 2]
## # Ordered by: hora
##    hora      n
## * <dbl>  <dbl>
## 1    0. 50726.
## 2    1. 80295.
## 3    2. 83731.
## 4    3. 90652.
## 5    4. 31372.
```

```
muestra_vuelos <-vuelos %>%
  filter(!is.na(arr_delay)) %>%
  mutate(sched_dep_time = as.numeric(sched_dep_time)) %>%
  ft_binarizer(
    input_col = "arr_delay",
    output_col = "tarde",
    threshold = 15
  ) %>%
  ft_bucketizer(
    input_col = "sched_dep_time",
    output_col = "horas",
    splits = c(400, 800, 1200, 1600, 2000, 2400)
  ) %>%
  mutate(dephour = paste0("h", as.integer(horas))) %>%
  sdf_partition(entrenar = 0.01, examinar = 0.09, otros = 0.9)
```

```
muestra_vuelos$entrenar
```

```
## # Source: spark<?> [?? x 22]
##     year month   day dep_time sched_dep_time dep_delay arr_time
##  * <int> <int> <int>    <int>          <dbl>     <dbl>    <int>
##  1  2013     1     1      657           700.       -3.      959
##  2  2013     1     1     1120           944.       96.     1331
##  3  2013     1     1     1317          1325.       -8.     1454
##  4  2013     1     1     1339          1335.        4.     1654
##  5  2013     1     1     1550          1550.        0.     1844
##  6  2013     1     1     1806          1810.       -4.     2002
##  7  2013     1     1     1843          1850.       -7.     2052
##  8  2013     1     1     1915          1920.       -5.     2238
##  9  2013     1     2      712           700.       12.      945
## 10  2013     1     2      739           745.       -6.     1116
## # ... with more rows, and 15 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, tarde <dbl>, horas <dbl>,
## #   dephour <chr>
```

```
modelo <- muestra_vuelos$entrenar %>%
  ml_logistic_regression(tarde ~.)
```

## Visualizaciones

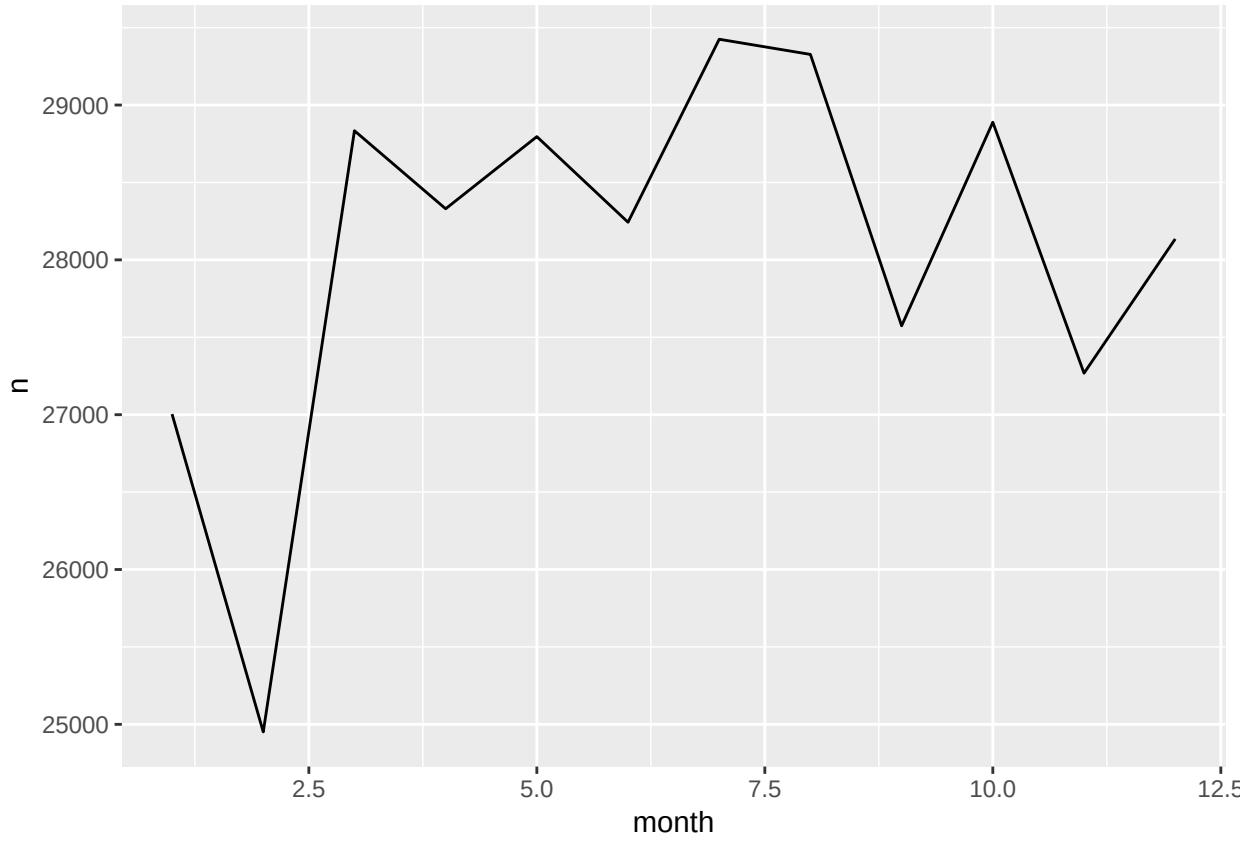
```
per_month <- vuelos %>%
  group_by(month) %>%
  tally() %>%
  collect()

per_month
```

```
## # A tibble: 12 x 2
##    month     n
##    <int> <dbl>
##  1    12 28135.
##  2     6 28243.
##  3     9 27574.
##  4    10 28889.
##  5     2 24951.
##  6     4 28330.
##  7     5 28796.
##  8     1 27004.
##  9    11 27268.
## 10     3 28834.
## 11     7 29425.
## 12     8 29327.
```

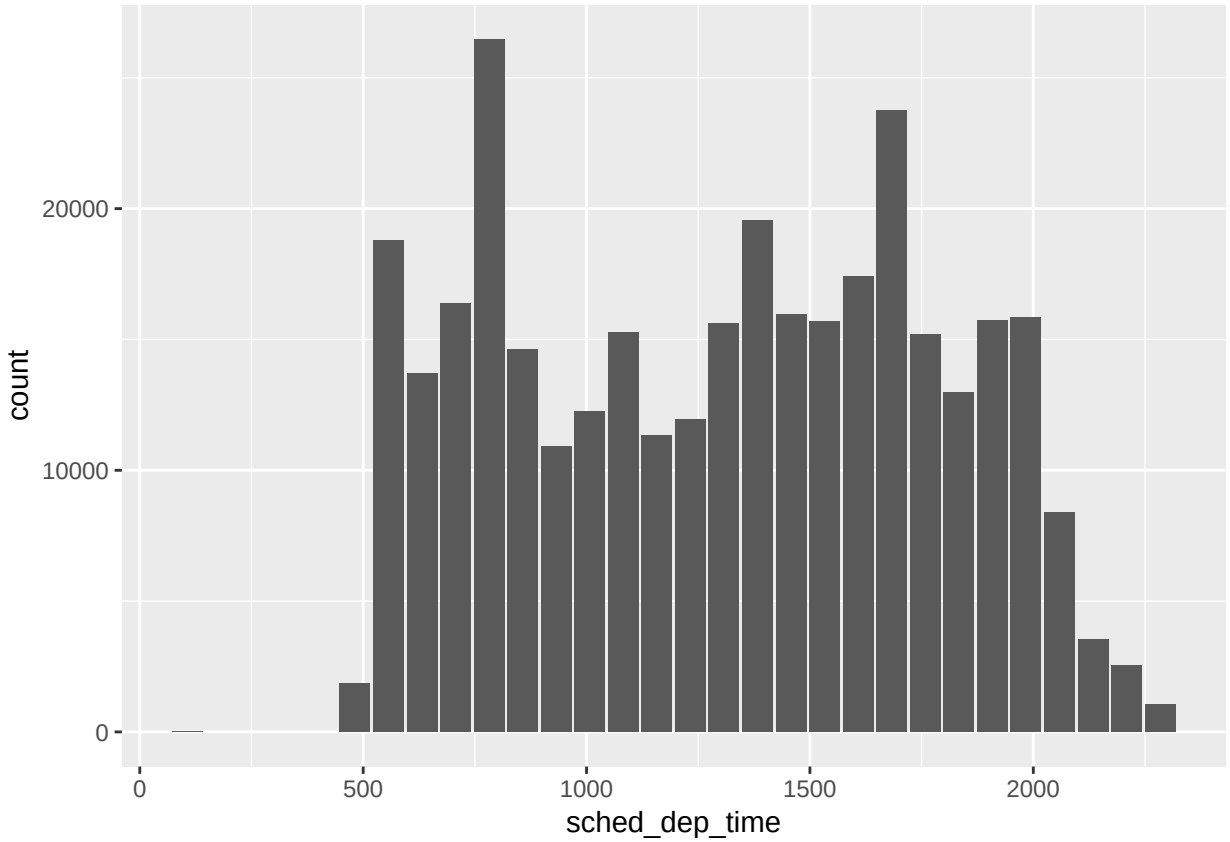
```
library(ggplot2)

ggplot(per_month) +
  geom_line(aes(month, n))
```

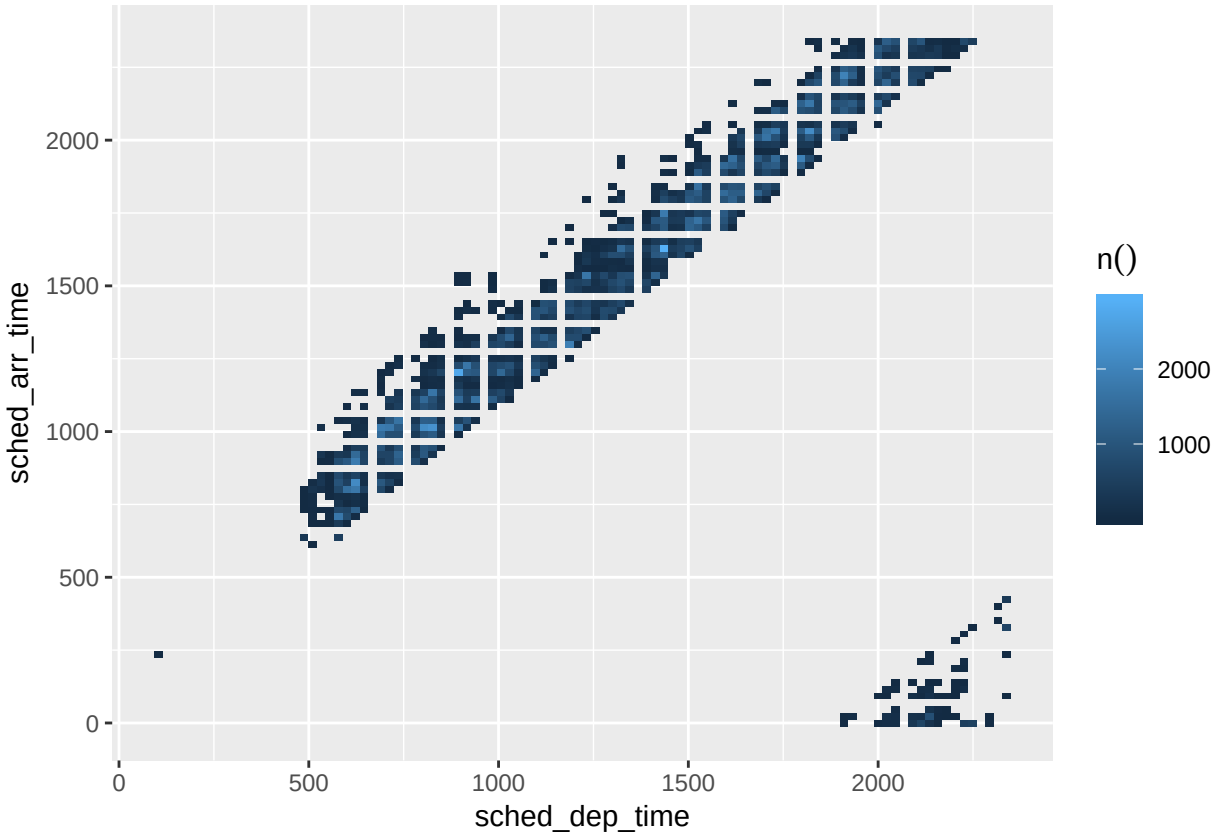



```
library(dbplot)
```

```
vuelos %>%
  dbplot_histogram(sched_dep_time)
```

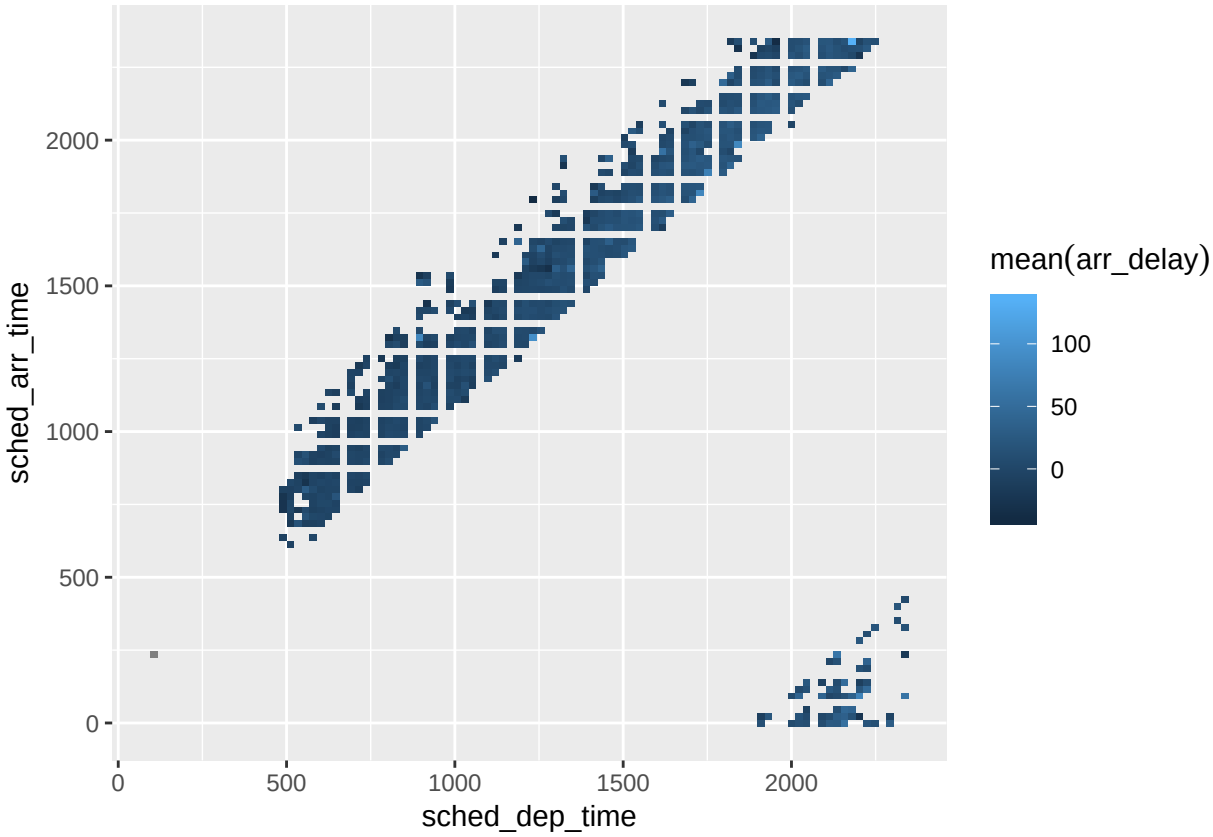



```
vuelos %>%
  dbplot_raster(sched_dep_time, sched_arr_time)
```

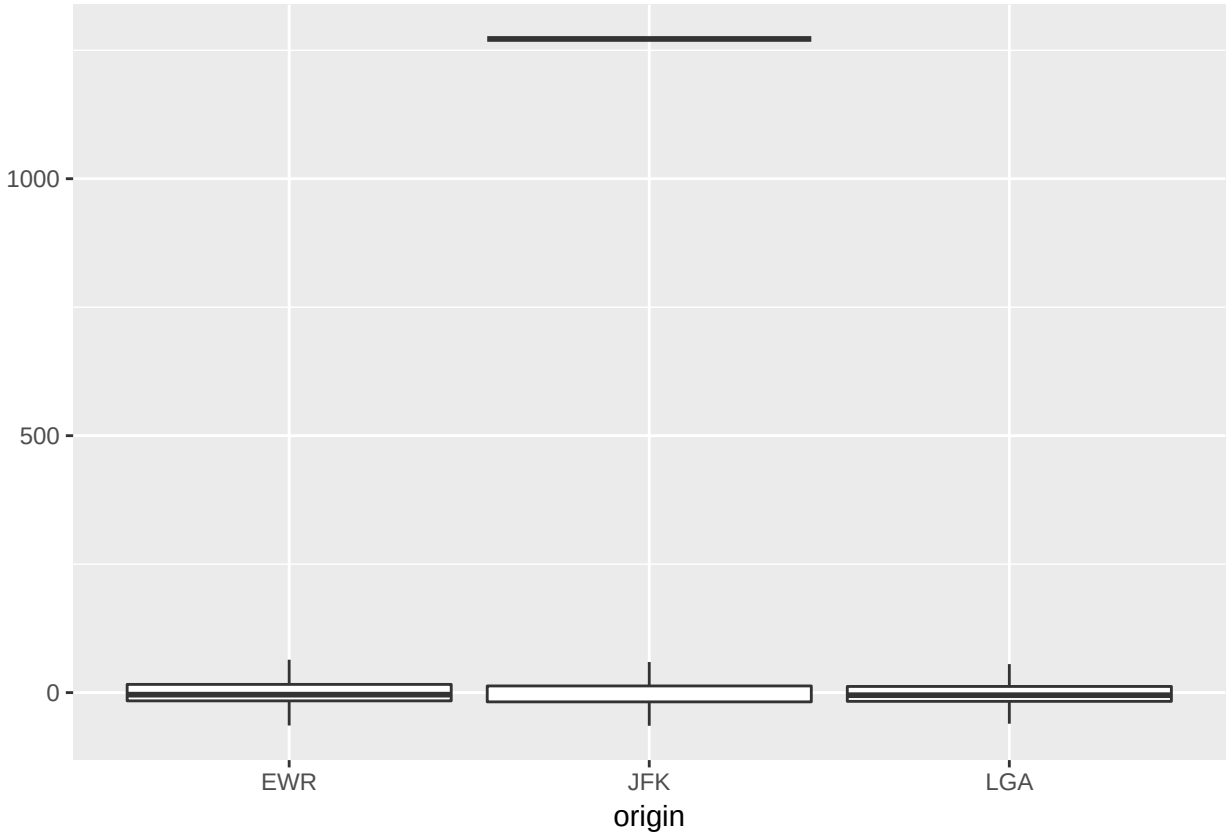



```
vuelos %>%
  dbplot_raster(sched_dep_time, sched_arr_time, mean(arr_delay))
```

```
## Warning: Missing values are always removed in SQL.
## Use `AVG(x, na.rm = TRUE)` to silence this warning
```




```
vuelos %>%
  dbplot_boxplot(origin, arr_delay)
```




## Pipelines (Tuberias)

```
entrenar <- muestra_vuelos$entrenar %>%
  mutate(
    arr_delay = ifelse(arr_delay == "NaN", 0, arr_delay)
  ) %>%
  select(
    month,
    sched_dep_time,
    arr_delay,
    distance
  ) %>%
  mutate_all(as.numeric)
```

```
tuberia_vuelos <- ml_pipeline(sc) %>%
  ft_dplyr_transformer(
    tbl = entrenar
  ) %>%
  ft_binarizer(
    input_col = "arr_delay",
    output_col = "tarde",
    threshold = 15
  ) %>%
  ft_bucketizer(
```

```
    input_col = "sched_dep_time",
    output_col = "horas",
    splits = c(400, 800, 1200, 1600, 2000, 2400)
  )  %>%
  ft_r_formula(tarde ~ horas + distance + arr_delay) %>%
  ml_logistic_regression()

tuberia_vuelos
```

```
## Pipeline (Estimator) with 5 stages
## <pipeline_8e5627878dd9>
##   Stages
##   |--1 SQLTransformer (Transformer)
##   |    <dplyr_transformer_8e563471b5d5>
##   |     (Parameters -- Column Names)
##   |--2 Binarizer (Transformer)
##   |    <binarizer_8e561558cdd9>
##   |     (Parameters -- Column Names)
##   |      input_col: arr_delay
##   |      output_col: tarde
##   |--3 Bucketizer (Transformer)
##   |    <bucketizer_8e5679326c81>
##   |     (Parameters -- Column Names)
##   |      input_col: sched_dep_time
##   |      output_col: horas
##   |--4 RFormula (Estimator)
##   |    <r_formula_8e565d71c750>
##   |     (Parameters -- Column Names)
##   |      features_col: features
##   |      label_col: label
##   |     (Parameters)
##   |      force_index_label: FALSE
##   |      formula: tarde ~ horas + distance + arr_delay
##   |      handle_invalid: error
##   |      stringIndexerOrderType: frequencyDesc
##   |--5 LogisticRegression (Estimator)
##   |    <logistic_regression_8e5658d4851d>
##   |     (Parameters -- Column Names)
##   |      features_col: features
##   |      label_col: label
##   |      prediction_col: prediction
##   |      probability_col: probability
##   |      raw_prediction_col: rawPrediction
##   |     (Parameters)
##   |      aggregation_depth: 2
##   |      elastic_net_param: 0
##   |      family: auto
##   |      fit_intercept: TRUE
##   |      max_iter: 100
##   |      reg_param: 0
##   |      standardization: TRUE
##   |      threshold: 0.5
##   |      tol: 1e-06
```

```
modelo_nuevo <- ml_fit(
  tuberia_vuelos,
  muestra_vuelos$entrenar
  )

modelo_nuevo
```

```
## PipelineModel (Transformer) with 5 stages
## <pipeline_8e5627878dd9>
##   Stages
##   |--1 SQLTransformer (Transformer)
##   |    <dplyr_transformer_8e563471b5d5>
##   |     (Parameters -- Column Names)
##   |--2 Binarizer (Transformer)
##   |    <binarizer_8e561558cdd9>
##   |     (Parameters -- Column Names)
##   |      input_col: arr_delay
##   |      output_col: tarde
##   |--3 Bucketizer (Transformer)
##   |    <bucketizer_8e5679326c81>
##   |     (Parameters -- Column Names)
##   |      input_col: sched_dep_time
##   |      output_col: horas
##   |--4 RFormulaModel (Transformer)
##   |    <r_formula_8e565d71c750>
##   |     (Parameters -- Column Names)
##   |      features_col: features
##   |      label_col: label
##   |     (Transformer Info)
##   |      formula:  chr "tarde ~ horas + distance + arr_delay"
##   |--5 LogisticRegressionModel (Transformer)
##   |    <logistic_regression_8e5658d4851d>
##   |     (Parameters -- Column Names)
##   |      features_col: features
##   |      label_col: label
##   |      prediction_col: prediction
##   |      probability_col: probability
##   |      raw_prediction_col: rawPrediction
##   |     (Transformer Info)
##   |      coefficient_matrix:  num [1, 1:3] -0.11352 -0.000256 27.801079
##   |      coefficients:  num [1:3] -0.11352 -0.000256 27.801079
##   |      intercept:  num -430
##   |      intercept_vector:  num -430
##   |      num_classes:  int 2
##   |      num_features:  int 3
##   |      threshold:  num 0.5
```

```
predicciones <- ml_transform(
  x = modelo_nuevo,
  dataset = muestra_vuelos$examinar
)

predicciones
```

```
## # Source: spark<?> [?? x 11]
##    month sched_dep_time arr_delay distance tarde horas features  label
##  * <dbl>          <dbl>     <dbl>    <dbl> <dbl> <dbl> <list>    <dbl>
##  1    1.           545.      -18.    1576.    0.    0. <dbl [3]>    0.
##  2    1.           600.      -14.    2565.    0.    0. <dbl [3]>    0.
##  3    1.           600.        7.    2475.    0.    0. <dbl [3]>    0.
##  4    1.           607.      -17.    1085.    0.    0. <dbl [3]>    0.
##  5    1.           647.        5.     301.    0.    0. <dbl [3]>    0.
##  6    1.           700.      -23.     997.    0.    0. <dbl [3]>    0.
##  7    1.           705.       27.    1147.    1.    0. <dbl [3]>    1.
##  8    1.           720.       10.     738.    0.    0. <dbl [3]>    0.
##  9    1.           805.      -19.     200.    0.    1. <dbl [3]>    0.
## 10    1.           810.       11.    1029.    0.    1. <dbl [3]>    0.
## # ... with more rows, and 3 more variables: rawPrediction <list>,
## #   probability <list>, prediction <dbl>
```

```
predicciones%>%
  group_by(tarde, prediction) %>%
  tally()
```

```
## # Source: spark<?> [?? x 3]
## # Groups: tarde
##   tarde prediction      n
## * <dbl>      <dbl>  <dbl>
## 1    0.         0. 22618.
## 2    1.         1.  7076.
```

```
ml_save(tuberia_vuelos, "tuberia", overwrite = TRUE)
```

```
## Model successfully saved.
```

```
dir("tuberia")
```

```
## [1] "metadata" "stages"
```

```
spark_disconnect(sc)
```