Laboratorios #2 – Dplyr y ggplot

Para esta y la siguiente parte deberá subir su solución en un archivo .Rmd colocando la pregunta completa como Rmarkdown y la respuesta que considere adecuada.

Los sistemas de renta de bicycletas se basan en kioskos que son puestos en diferentes áreas de una ciudad. En estos kioskos las personas pueden suscribirse, rentar y devolver las bicicletas. Esto permite que el usuario rente un bicicleta y la pueda devolver en otro lado. Actualmente hay mas de 500 de estos proyectos alrededor del mundo.

Estos kioskos se vuelven sensores del flujo de personas dentro de ciudades.

Su tarea es contestar las preguntas de este documento, basadas en la data que se presenta en el siguiente link.

- Variables
  - **datetime**: hourly date + timestamp
  - **season**: 1 = spring, 2 = summer, 3 = fall, 4 = winter
  - **holiday**: whether the day is considered a holiday
  - **workingday**: whether the day is neither a weekend nor holiday
  - **weather**:
    - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
    - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
    - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
    - 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
  - **temp**: temperature in Celsius
  - **atemp**: "feels like" temperature in Celsius
  - **humidity**: relative humidity
  - **windspeed**: wind speed
  - **casual**: number of non-registered user rentals initiated
  - **registered**: number of registered user rentals initiated
  - **count**: number of total rentals

```
dataset = read.csv("dataset.csv")

head(dataset)
```

```
##   instant     dteday season yr mnth hr holiday weekday workingday weathersit
## 1       1 2011-01-01      1  0    1  0       0       6          0          1
## 2       2 2011-01-01      1  0    1  1       0       6          0          1
## 3       3 2011-01-01      1  0    1  2       0       6          0          1
## 4       4 2011-01-01      1  0    1  3       0       6          0          1
## 5       5 2011-01-01      1  0    1  4       0       6          0          1
## 6       6 2011-01-01      1  0    1  5       0       6          0          2
##   temp  atemp  hum windspeed casual registered cnt
## 1 0.24 0.2879 0.81    0.0000      3         13  16
```

```
## 2 0.22 0.2727 0.80    0.0000      8         32  40
## 3 0.22 0.2727 0.80    0.0000      5         27  32
## 4 0.24 0.2879 0.75    0.0000      3         10  13
## 5 0.24 0.2879 0.75    0.0000      0          1   1
## 6 0.24 0.2576 0.75    0.0896      0          1   1
```

1. Cree un conjunto de columnas nuevas: día, mes, año, hora y minutos a partir de la comlumna datetime, para esto investigue como puede "desarmar" la variable datetime utilizando lubridate y mutate.

```r
# bibliotecas necesarias
library(lubridate)
```

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

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(dplyr)
```

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

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# convertir la columma "dteday" a un formato de fecha y hora
dataset$dteday <- ymd(dataset$dteday)

# extraer las partes de la fecha y la hora
dataset <- dataset %>%
  mutate(
    dia = day(dteday),
    mes = month(dteday),
    año = year(dteday),
  )
```

```r
dataset$hora =   dataset$hr

# Verificar el resultado
head(dataset)
```

```
##   instant     dteday season yr mnth hr holiday weekday workingday weathersit
## 1       1 2011-01-01      1  0    1  0       0       6          0          1
## 2       2 2011-01-01      1  0    1  1       0       6          0          1
## 3       3 2011-01-01      1  0    1  2       0       6          0          1
## 4       4 2011-01-01      1  0    1  3       0       6          0          1
## 5       5 2011-01-01      1  0    1  4       0       6          0          1
## 6       6 2011-01-01      1  0    1  5       0       6          0          2
##   temp  atemp  hum windspeed casual registered cnt dia mes  año hora
## 1 0.24 0.2879 0.81    0.0000      3         13  16   1   1 2011    0
## 2 0.22 0.2727 0.80    0.0000      8         32  40   1   1 2011    1
## 3 0.22 0.2727 0.80    0.0000      5         27  32   1   1 2011    2
## 4 0.24 0.2879 0.75    0.0000      3         10  13   1   1 2011    3
## 5 0.24 0.2879 0.75    0.0000      0          1   1   1   1 2011    4
## 6 0.24 0.2576 0.75    0.0896      0          1   1   1   1 2011    5
```

2. ¿Qué mes es el que tiene la mayor demanda? Muestre una tabla y una gráfica

Asumiendo que cada linea representa que un cliente haya pedido una bicicleta debo contar las filas para saber cuantas bicicletas se han pedido.

```r
dataset_mes_año = dataset %>%
                  select(año,mes) %>%
                  group_by(año,mes) %>%
                  summarize(registros = n())
```

```
## `summarise()` has grouped output by 'año'. You can override using the `.groups`
## argument.
```

```r
print(dataset_mes_año)
```

```
## # A tibble: 24 x 3
## # Groups:   año [2]
##      año   mes registros
##    <dbl> <dbl>     <int>
##  1  2011     1       688
##  2  2011     2       649
##  3  2011     3       730
##  4  2011     4       719
##  5  2011     5       744
##  6  2011     6       720
##  7  2011     7       744
##  8  2011     8       731
##  9  2011     9       717
## 10  2011    10       743
## # i 14 more rows
```

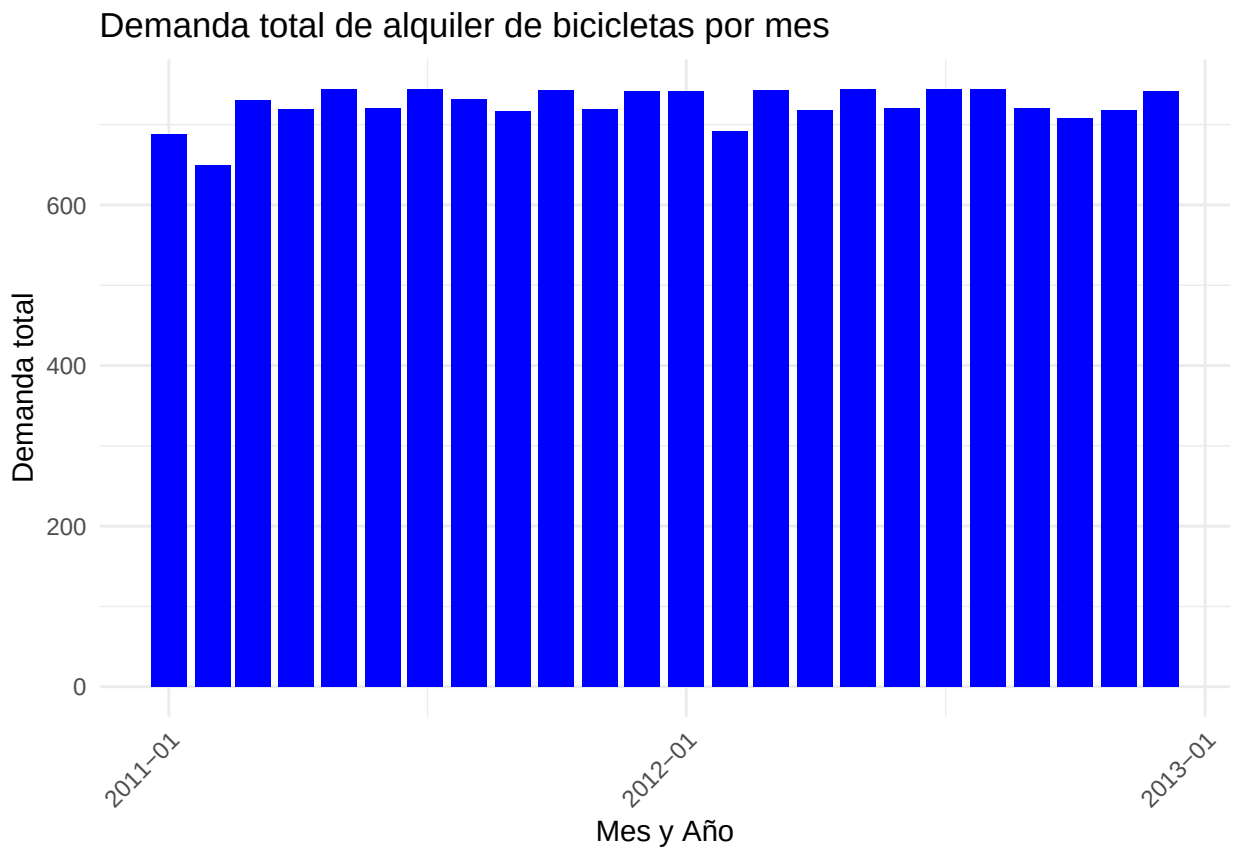
crear una nueva columan de fechas para que los datos salgan ordenados

```
dataset_mes_año <- dataset_mes_año %>%
  mutate(año_mes = as.Date(paste(año, mes, "01", sep = "-"), format = "%Y-%m-%d"))
```

```
library(ggplot2)
```

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

```
dataset_mes_año %>%
  ggplot(aes(x = año_mes, y = registros)) +
  geom_col(fill = "blue") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y-%m") +
  labs(title = "Demanda total de alquiler de bicicletas por mes",
       x = "Mes y Año",
       y = "Demanda total") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Buscando el maximo de cada columna

```
dataset_mes_año$registros %>% max()
```

```
## [1] 744
```

Estos serian los meses en donde hay un maximo de consumidores del servicio

```
dataset_mes_año %>% filter(registros == (dataset_mes_año$registros %>% max()))
```

```
## # A tibble: 5 x 4
## # Groups:   año [2]
##     año   mes registros año_mes
##   <dbl> <dbl>     <int> <date>
## 1  2011     5       744 2011-05-01
## 2  2011     7       744 2011-07-01
## 3  2012     5       744 2012-05-01
## 4  2012     7       744 2012-07-01
## 5  2012     8       744 2012-08-01
```

3. ¿Qué rango de hora es la de mayor demanda? Muestre una tabla y una gráfica

```
dataset_mes_año_hora = dataset %>%
                  select(hora) %>%
                  group_by(hora) %>%
                  summarize(registros = n())
```

```
dataset_mes_año_hora$registros %>% max()
```

```
## [1] 730
```

```
dataset_mes_año_hora %>% filter(registros == (dataset_mes_año_hora$registros %>% max()))
```

```
## # A tibble: 2 x 2
##    hora registros
##   <int>     <int>
## 1    16       730
## 2    17       730
```

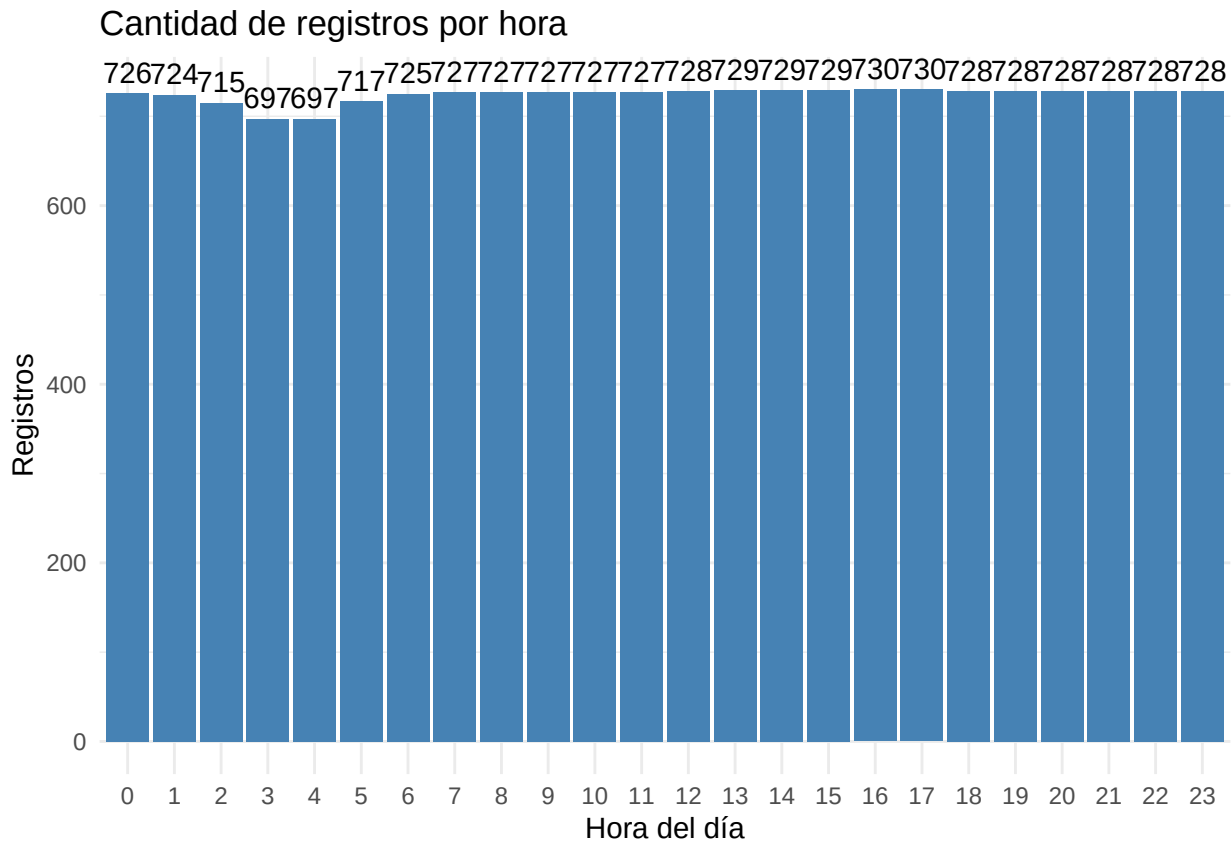
```
dataset_mes_año_hora
```

```
## # A tibble: 24 x 2
##     hora registros
##    <int>     <int>
##  1     0       726
##  2     1       724
##  3     2       715
##  4     3       697
##  5     4       697
##  6     5       717
##  7     6       725
##  8     7       727
##  9     8       727
## 10     9       727
## # i 14 more rows
```

```r
dataset_mes_año_hora %>% filter(registros == (dataset_mes_año_hora$registros %>% max())) %>% select(hora
```

```
## # A tibble: 2 x 2
##    hora registros
##   <int>     <int>
## 1    16       730
## 2    17       730
```

```r
dataset_mes_año_hora %>%
  ggplot(aes(x = as.factor(hora), y = registros)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = registros), vjust = -0.5, color = "black") +
  labs(title = "Cantidad de registros por hora",
       x = "Hora del día",
       y = "Registros") +
  theme_minimal()
```



si sacara el promedio por mes solo lo dividiria entre 24 porque son dos años pero tendria la misma forma la grafica.

4. ¿Qué temporada es la mas alta? Muestre una tabla.

```
dataset_temporada = dataset %>%
                  select(season) %>%
                  group_by(season) %>%
                  summarize(registros = n())
```

```
dataset_temporada$registros %>% max()
```

```
## [1] 4496
```

```
dataset_temporada
```

```
## # A tibble: 4 x 2
##   season registros
##    <int>     <int>
## 1      1      4242
## 2      2      4409
## 3      3      4496
## 4      4      4232
```

En la temporada 3 es la mas alta

5. ¿A que temperatura disminuye la demanda? Muestre una gráfica para analizar y dar su respuesta.

Agrupando una suma por temperatura

```
dataset_temperatura = dataset %>%
                  select(temp) %>%
                  group_by(temp) %>%
                  summarize(registros = n())
```

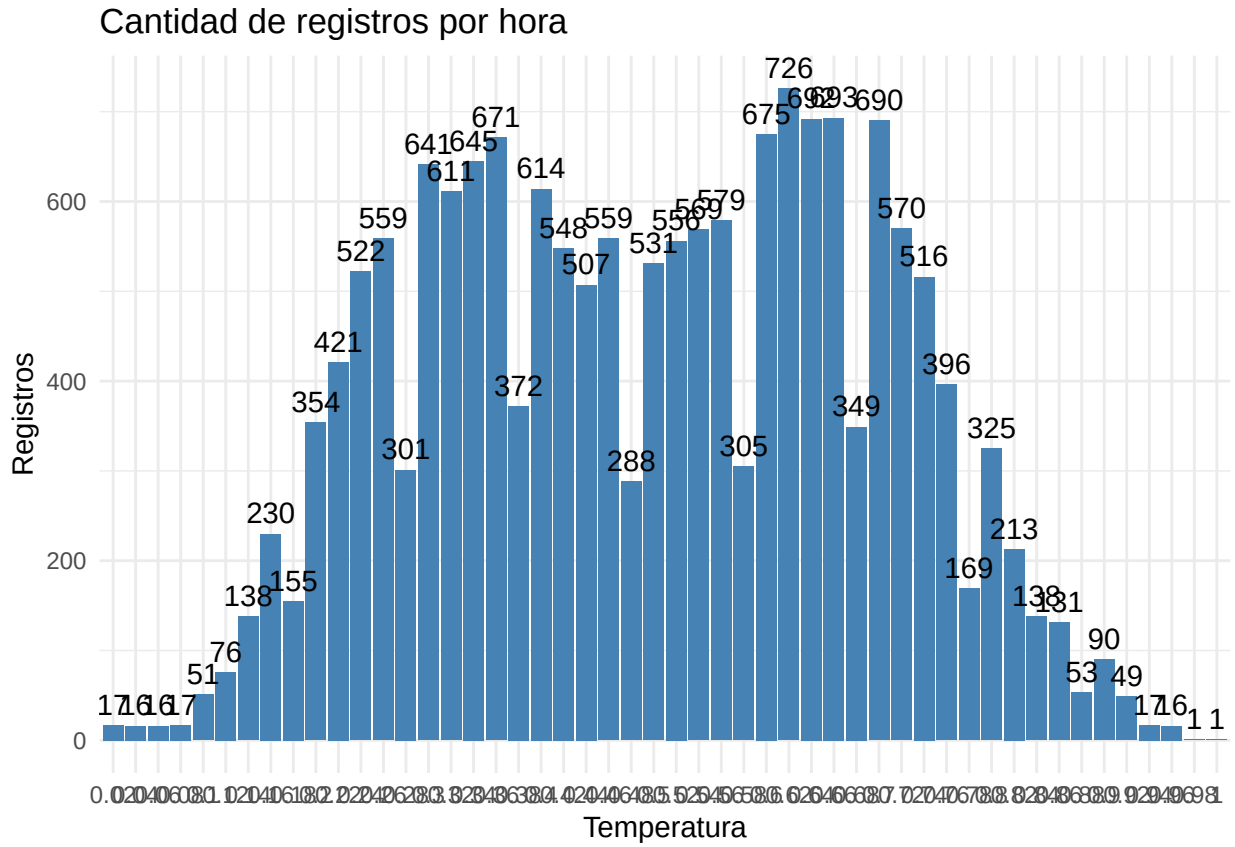
```
dataset_temperatura$registros %>% max()
```

```
## [1] 726
```

```
dataset_temperatura %>% filter(registros == (dataset_temperatura$registros %>% min()))
```

```
## # A tibble: 2 x 2
##    temp registros
##   <dbl>     <int>
## 1  0.98         1
## 2  1            1
```

```
dataset_temperatura %>%
  ggplot(aes(x = as.factor(temp), y = registros)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = registros), vjust = -0.5, color = "black") +
  labs(title = "Cantidad de registros por hora",
       x = "Temperatura",
       y = "Registros") +
  theme_minimal()
```




```
head(dataset_temperatura %>% arrange(registros))
```

```
## # A tibble: 6 x 2
##    temp registros
##   <dbl>     <int>
## 1  0.98         1
## 2  1            1
## 3  0.04        16
## 4  0.06        16
## 5  0.96        16
## 6  0.02        17
```

La demanda disminuye en temperaturas muy bajas o muy altas

6. ¿A que humedad disminuye la demanda? Muestre una gráfica para analizar y dar su respuesta.

```
dataset_humedad = dataset %>%
                  select(hum) %>%
                  group_by(hum) %>%
                  summarize(registros = n())
```

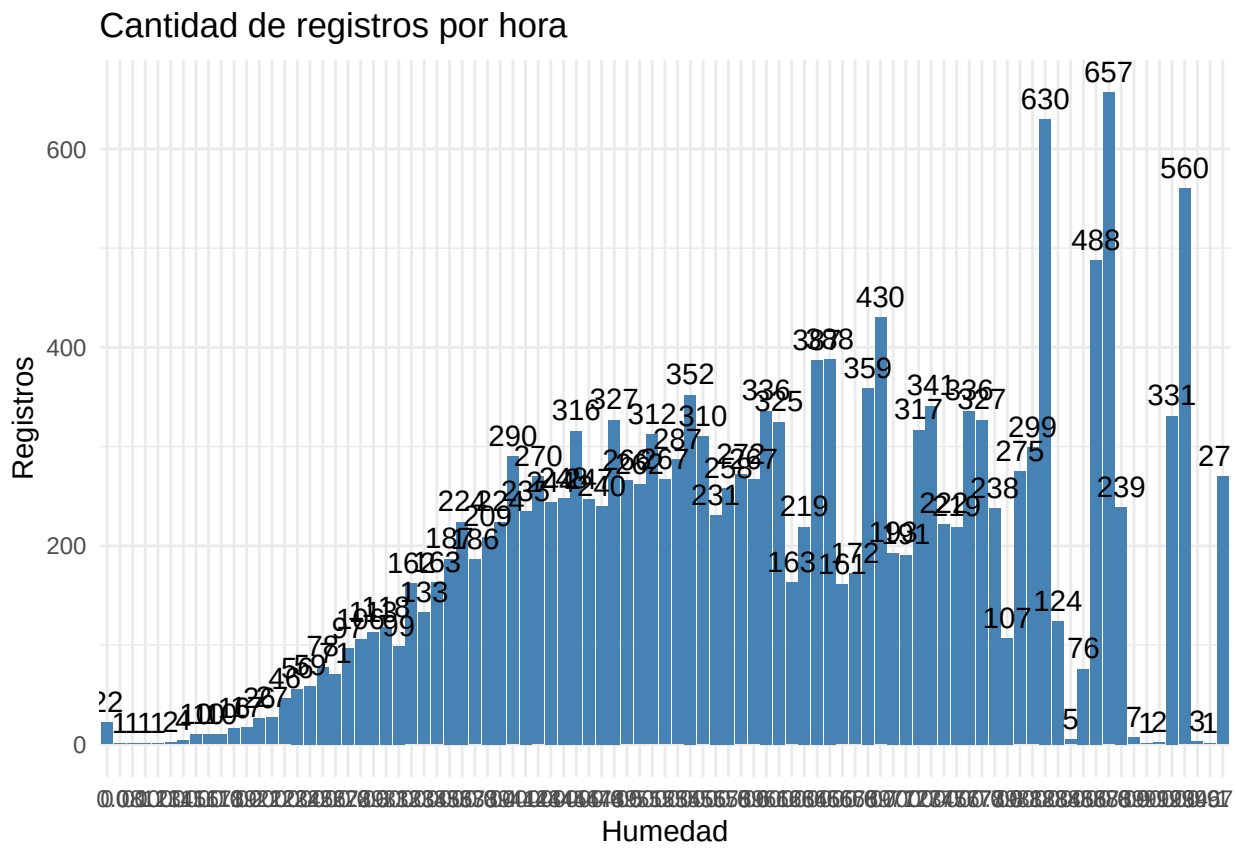
```
dataset_humedad$registros %>% min()
```

```
## [1] 1
```

```
dataset_humedad %>% filter(registros == (dataset_humedad$registros %>% min()))
```

```
## # A tibble: 6 x 2
##     hum registros
##   <dbl>     <int>
## 1  0.08         1
## 2  0.1          1
## 3  0.12         1
## 4  0.13         1
## 5  0.91         1
## 6  0.97         1
```

```
dataset_humedad %>%
  ggplot(aes(x = as.factor(hum), y = registros)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = registros), vjust = -0.5, color = "black") +
  labs(title = "Cantidad de registros por hora",
       x = "Humedad",
       y = "Registros") +
  theme_minimal()
```



```
head(dataset_humedad %>% arrange(registros))
```

```
## # A tibble: 6 x 2
##     hum registros
##   <dbl>     <int>
## 1  0.08         1
## 2  0.1          1
## 3  0.12         1
## 4  0.13         1
## 5  0.91         1
## 6  0.97         1
```