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
```

```r
summary(dataset)
```

```
##     instant          dteday                season           yr        
##  Min.   :    1   Min.   :2011-01-01   Min.   :1.000   Min.   :0.0000  
##  1st Qu.: 4346   1st Qu.:2011-07-04   1st Qu.:2.000   1st Qu.:0.0000  
##  Median : 8690   Median :2012-01-02   Median :3.000   Median :1.0000  
##  Mean   : 8690   Mean   :2012-01-02   Mean   :2.502   Mean   :0.5026  
##  3rd Qu.:13034   3rd Qu.:2012-07-02   3rd Qu.:3.000   3rd Qu.:1.0000  
##  Max.   :17379   Max.   :2012-12-31   Max.   :4.000   Max.   :1.0000  
##       mnth             hr           holiday           weekday      
##  Min.   : 1.000   Min.   : 0.00   Min.   :0.00000   Min.   :0.000  
##  1st Qu.: 4.000   1st Qu.: 6.00   1st Qu.:0.00000   1st Qu.:1.000  
##  Median : 7.000   Median :12.00   Median :0.00000   Median :3.000  
##  Mean   : 6.538   Mean   :11.55   Mean   :0.02877   Mean   :3.004  
##  3rd Qu.:10.000   3rd Qu.:18.00   3rd Qu.:0.00000   3rd Qu.:5.000  
##  Max.   :12.000   Max.   :23.00   Max.   :1.00000   Max.   :6.000  
##    workingday       weathersit         temp           atemp      
##  Min.   :0.0000   Min.   :1.000   Min.   :0.020   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:0.340   1st Qu.:0.3333  
##  Median :1.0000   Median :1.000   Median :0.500   Median :0.4848  
##  Mean   :0.6827   Mean   :1.425   Mean   :0.497   Mean   :0.4758  
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:0.660   3rd Qu.:0.6212  
##  Max.   :1.0000   Max.   :4.000   Max.   :1.000   Max.   :1.0000  
##       hum          windspeed          casual         registered    
##  Min.   :0.0000   Min.   :0.0000   Min.   :  0.00   Min.   :  0.0  
##  1st Qu.:0.4800   1st Qu.:0.1045   1st Qu.:  4.00   1st Qu.: 34.0  
##  Median :0.6300   Median :0.1940   Median : 17.00   Median :115.0  
##  Mean   :0.6272   Mean   :0.1901   Mean   : 35.68   Mean   :153.8  
##  3rd Qu.:0.7800   3rd Qu.:0.2537   3rd Qu.: 48.00   3rd Qu.:220.0  
##  Max.   :1.0000   Max.   :0.8507   Max.   :367.00   Max.   :886.0  
##       cnt              dia              mes             año       
##  Min.   :  1.0   Min.   : 1.00   Min.   : 1.000   Min.   :2011  
##  1st Qu.: 40.0   1st Qu.: 8.00   1st Qu.: 4.000   1st Qu.:2011  
```

```
##  Median :142.0   Median :16.00   Median : 7.000   Median :2012
##  Mean   :189.5   Mean   :15.68   Mean   : 6.538   Mean   :2012
##  3rd Qu.:281.0   3rd Qu.:23.00   3rd Qu.:10.000   3rd Qu.:2012
##  Max.   :977.0   Max.   :31.00   Max.   :12.000   Max.   :2012
##       hora
##  Min.   : 0.00
##  1st Qu.: 6.00
##  Median :12.00
##  Mean   :11.55
##  3rd Qu.:18.00
##  Max.   :23.00
```

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.

```
dataset_mes_año <- dataset %>%
                   select(año, mes, cnt) %>%
                   group_by(año, mes) %>%
                   summarize(registros = sum(cnt))
```

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

```
print(dataset_mes_año)
```

```
## # A tibble: 24 x 3
## # Groups:   año [2]
##      año   mes registros
##    <dbl> <dbl>     <int>
##  1  2011     1     38189
##  2  2011     2     48215
##  3  2011     3     64045
##  4  2011     4     94870
##  5  2011     5    135821
##  6  2011     6    143512
##  7  2011     7    141341
##  8  2011     8    136691
##  9  2011     9    127418
## 10  2011    10    123511
## # i 14 more rows
```

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

```
## # A tibble: 1 x 3
## # Groups:   año [1]
##     año   mes registros
##   <dbl> <dbl>     <int>
## 1  2012     9    218573
```

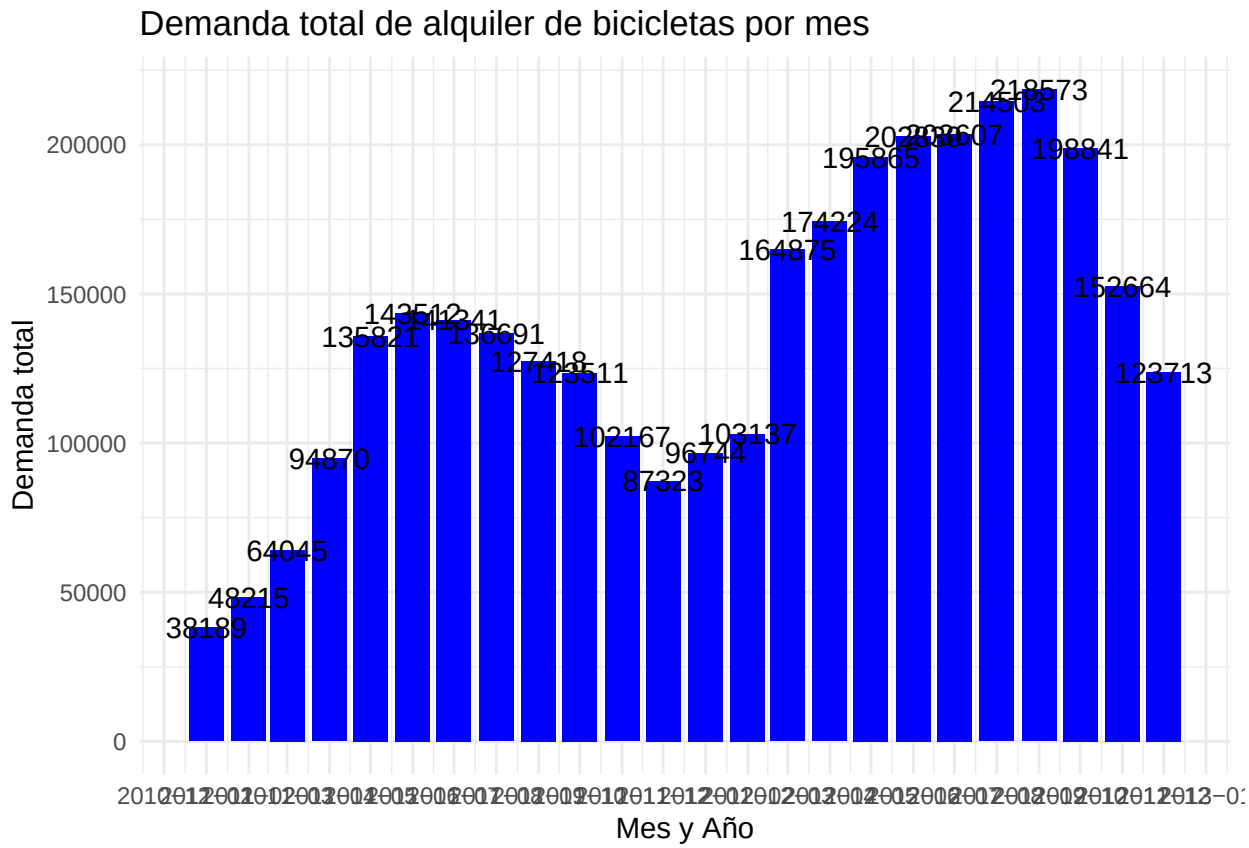
crear una nueva columan de fechas para que los datos salgan ordenados

```
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") +
  geom_text(aes(label = registros), color = "black")+
  scale_x_date(date_breaks = "1 month", date_labels = "%Y-%m") +
  labs(title = "Demanda total de alquiler de bicicletas por mes",
       x = "Mes y Año",
       y = "Demanda total") +
  theme_minimal()
```




```
dataset_mes_año %>% select(año_mes,registros) %>% print()
```

```
## Adding missing grouping variables: `año`
```

```
## # A tibble: 24 x 3
## # Groups:   año [2]
##      año año_mes    registros
##    <dbl> <date>         <int>
##  1  2011 2011-01-01     38189
##  2  2011 2011-02-01     48215
##  3  2011 2011-03-01     64045
##  4  2011 2011-04-01     94870
##  5  2011 2011-05-01    135821
##  6  2011 2011-06-01    143512
##  7  2011 2011-07-01    141341
##  8  2011 2011-08-01    136691
##  9  2011 2011-09-01    127418
## 10  2011 2011-10-01    123511
## # i 14 more rows
```

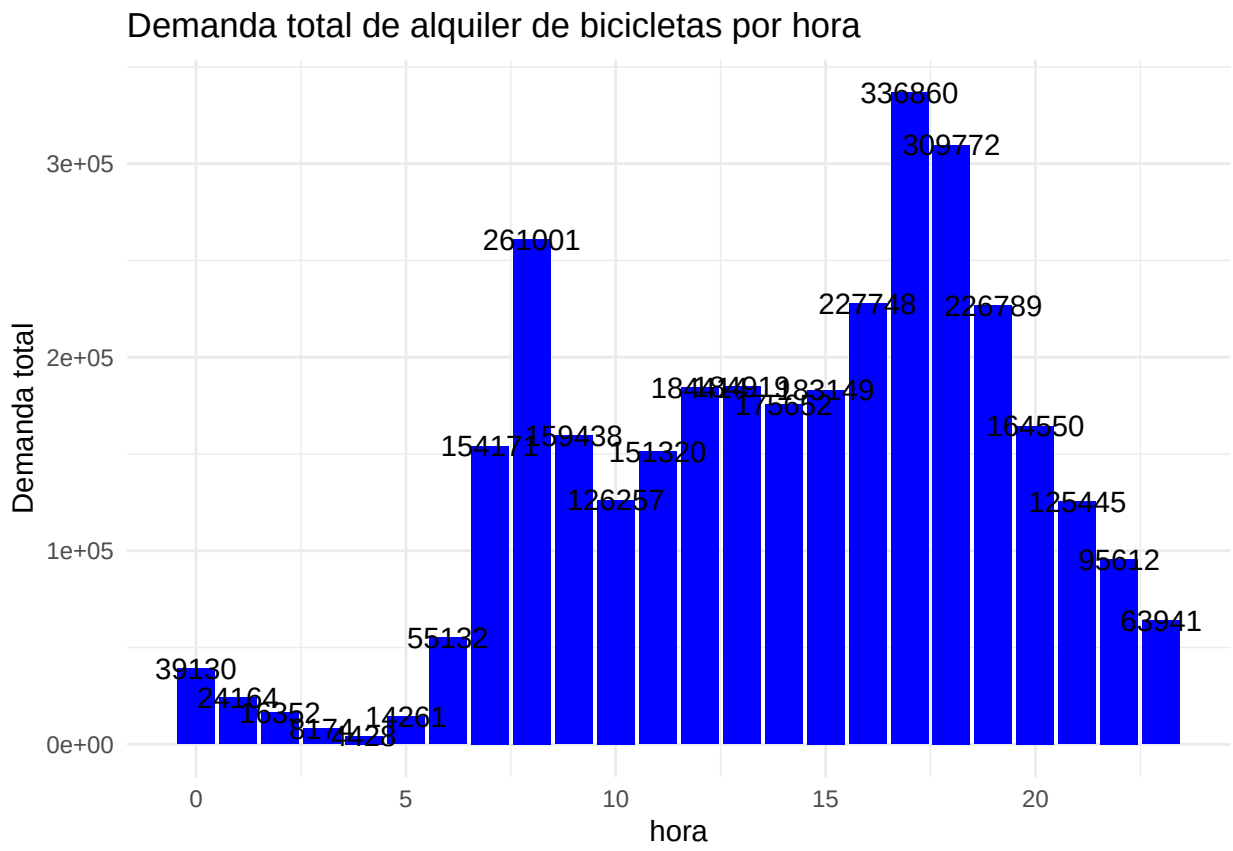
3. ¿Qué rango de hora es la de mayor demanda? Muestre una tabla y una gráfica

```
dataset_mes_año_hora <- dataset %>%
                  select(hora, cnt) %>%
                  group_by(hora) %>%
                  summarize(demanda = sum(cnt))

print(dataset_mes_año_hora)
```

```
## # A tibble: 24 x 2
##     hora demanda
##    <int>   <int>
##  1     0   39130
##  2     1   24164
##  3     2   16352
##  4     3    8174
##  5     4    4428
##  6     5   14261
##  7     6   55132
##  8     7  154171
##  9     8  261001
## 10     9  159438
## # i 14 more rows
```

```
library(ggplot2)
dataset_mes_año_hora %>%
  ggplot(aes(x = hora, y = demanda)) +
  geom_col(fill = "blue") +
  geom_text(aes(label = demanda), color = "black")+
  labs(title = "Demanda total de alquiler de bicicletas por hora",
       x = "hora",
       y = "Demanda total") +
  theme_minimal()
```




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

```
## # A tibble: 1 x 2
##    hora demanda
##   <int>   <int>
## 1    17  336860
```

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

```
dataset_temporada = dataset %>%
                  select(season, cnt) %>%
                  group_by(season) %>%
                  summarize(demanda = sum(cnt))

print(dataset_temporada)
```

```
## # A tibble: 4 x 2
##   season demanda
##    <int>   <int>
## 1      1  471348
## 2      2  918589
## 3      3 1061129
## 4      4  841613
```

```
dataset_temporada %>% filter(demanda == max(demanda))
```

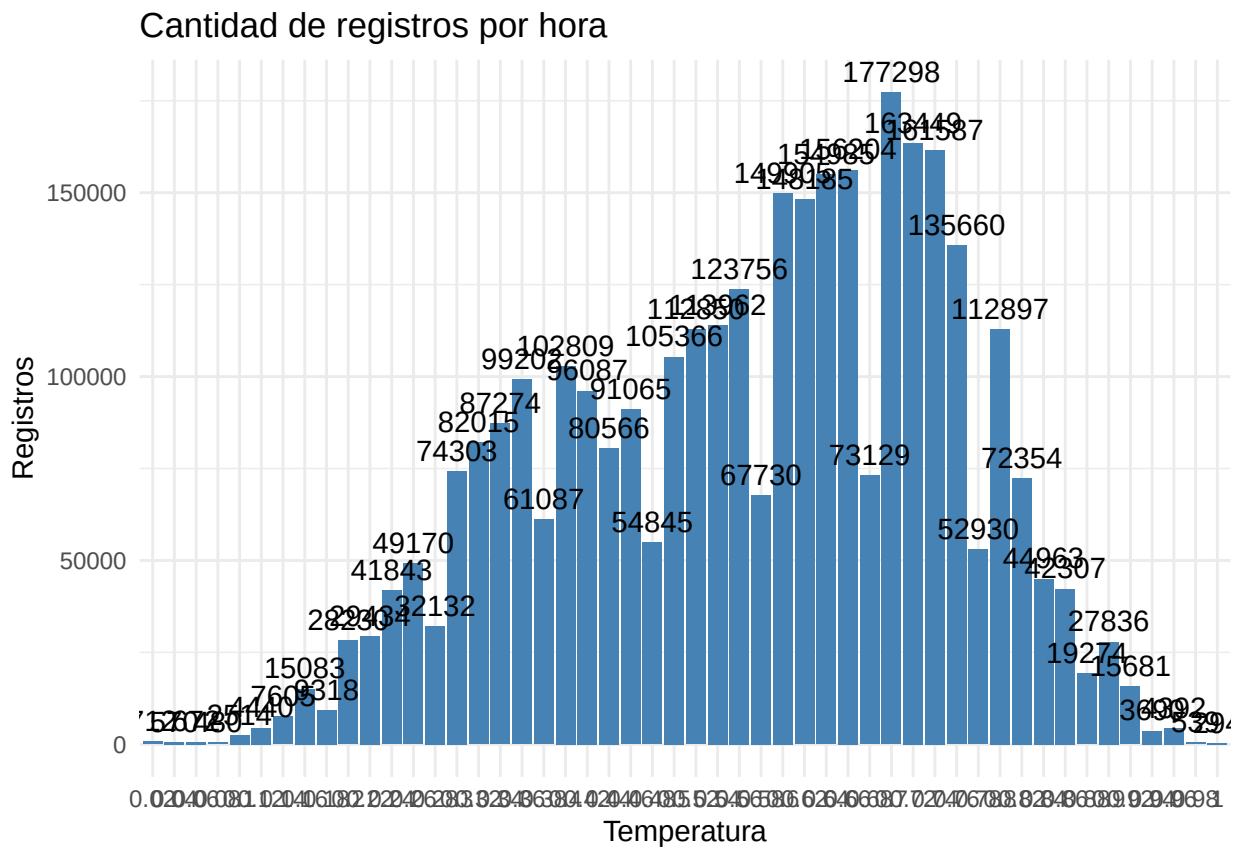
```
## # A tibble: 1 x 2
##   season demanda
##    <int>   <int>
## 1      3 1061129
```

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,cnt) %>%
                  group_by(temp) %>%
                  summarize(demanda = sum(cnt))
```

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

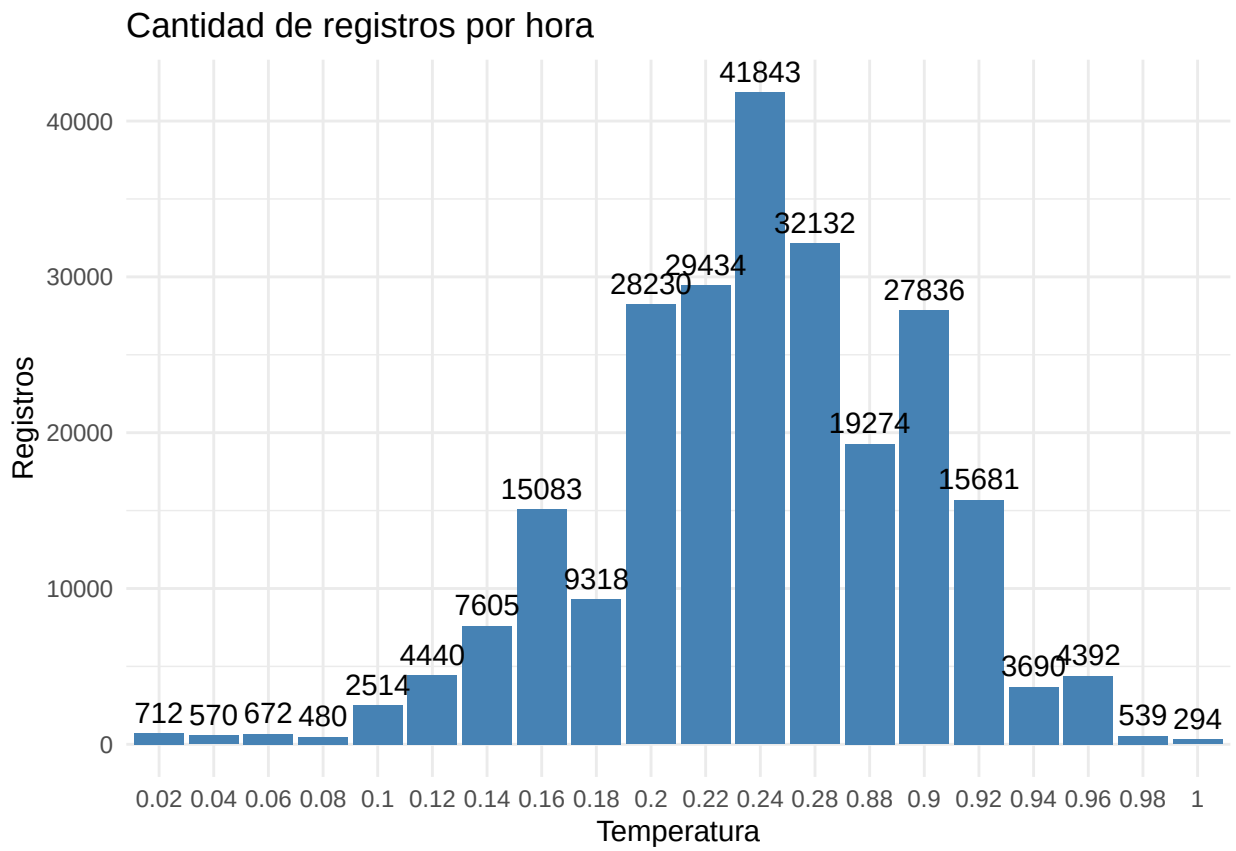


hay un rango de temperaturas a las que disminuye la demanta por lo cual hare un top 20

```
dataset_temperatura %>% arrange(demanda) %>% head(20)
```

```
## # A tibble: 20 x 2
##     temp demanda
##    <dbl>   <int>
##  1  1        294
##  2  0.08     480
##  3  0.98     539
##  4  0.04     570
##  5  0.06     672
##  6  0.02     712
##  7  0.1     2514
##  8  0.94    3690
##  9  0.96    4392
## 10  0.12    4440
## 11  0.14    7605
## 12  0.18    9318
## 13  0.16   15083
## 14  0.92   15681
## 15  0.88   19274
## 16  0.9    27836
## 17  0.2    28230
## 18  0.22   29434
## 19  0.28   32132
## 20  0.24   41843
```

```
dataset_temperatura %>% arrange(demanda) %>% head(20)  %>%
  ggplot(aes(x = as.factor(temp), y = demanda)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = demanda), vjust = -0.5, color = "black") +
  labs(title = "Cantidad de registros por hora",
       x = "Temperatura",
       y = "Registros") +
  theme_minimal()
```



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,cnt) %>%
                  group_by(hum) %>%
                  summarize(demanda = sum(cnt))
```

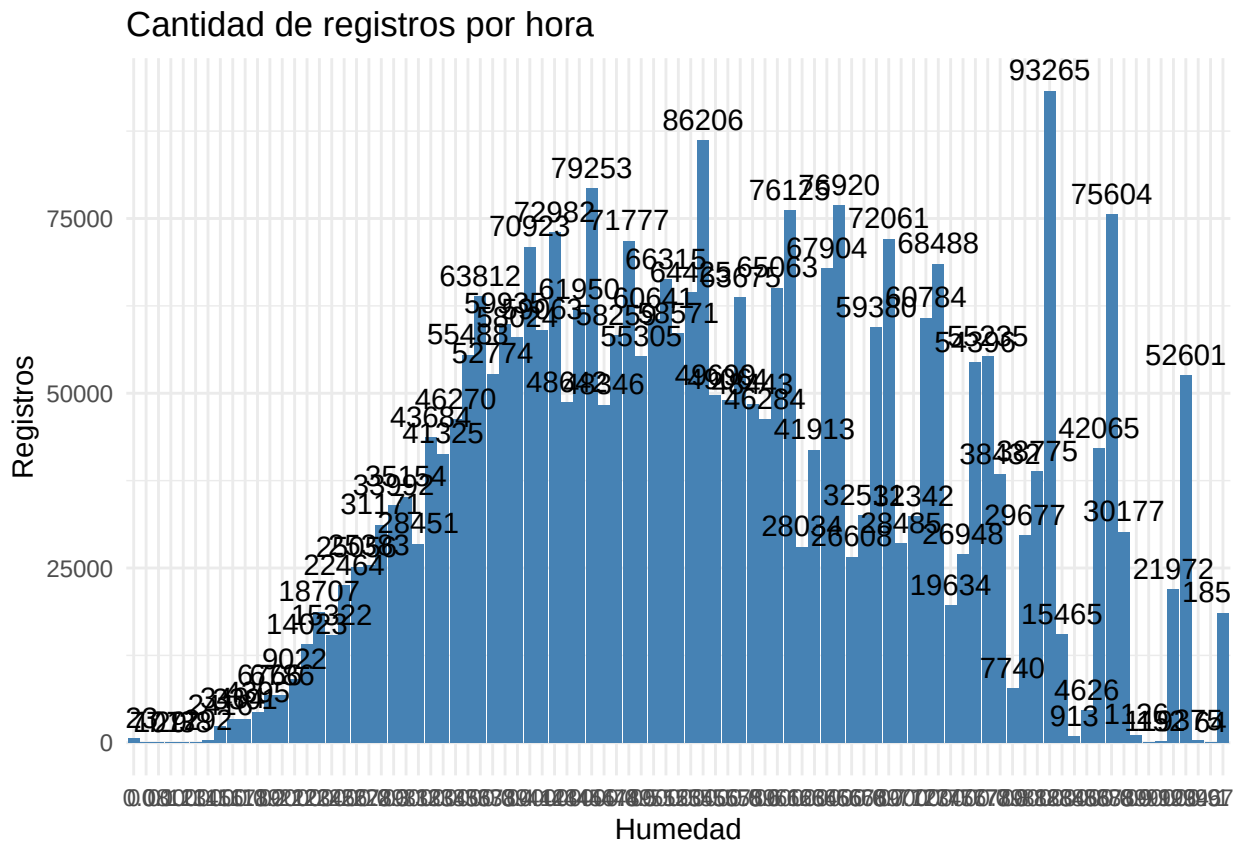
```
dataset_humedad$demanda %>% min()
```

```
## [1] 17
```

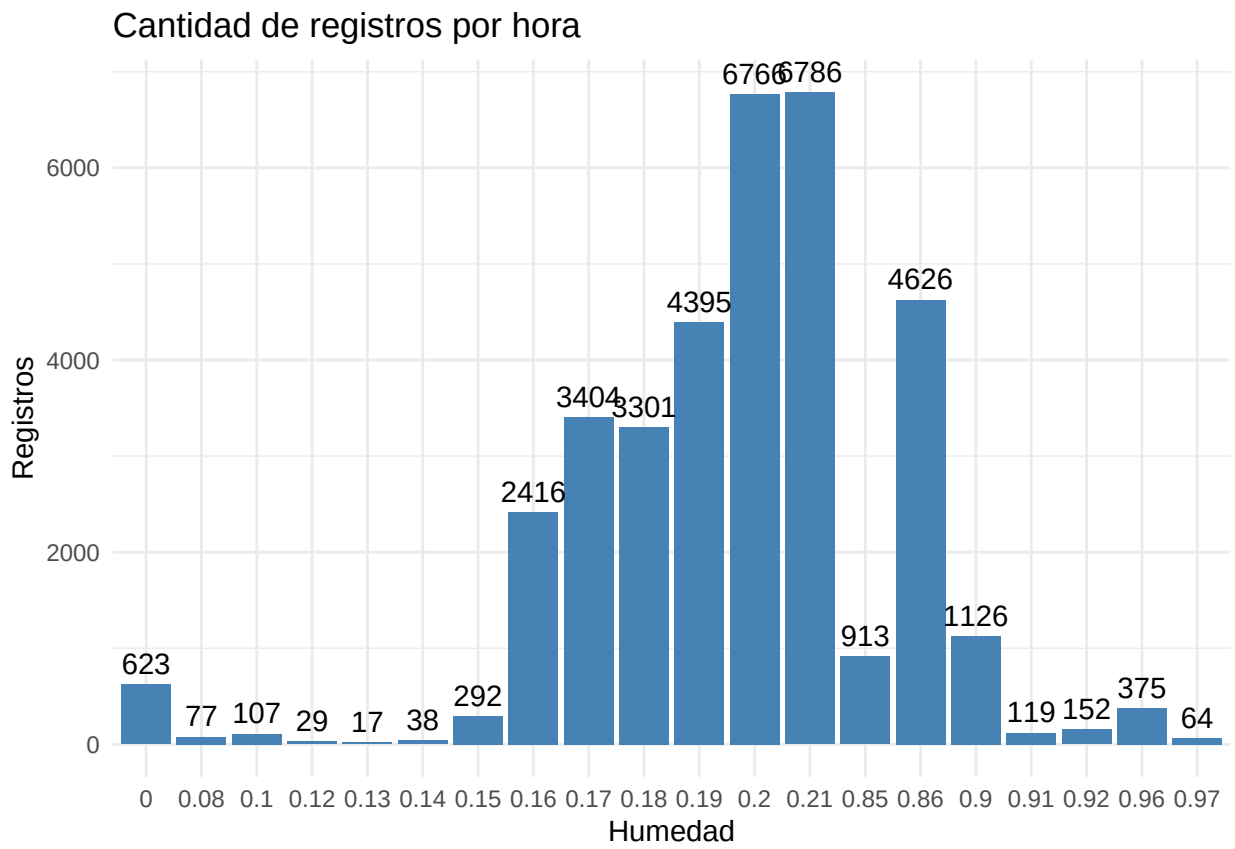
```
dataset_humedad %>% filter(demanda == min(demanda))
```

```
## # A tibble: 1 x 2
##     hum demanda
##   <dbl>   <int>
## 1  0.13      17
```

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




```
dataset_humedad %>% arrange(demanda) %>% head(20)  %>%
  ggplot(aes(x = as.factor(hum), y = demanda)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = demanda), vjust = -0.5, color = "black") +
  labs(title = "Cantidad de registros por hora",
       x = "Humedad",
       y = "Registros") +
  theme_minimal()
```



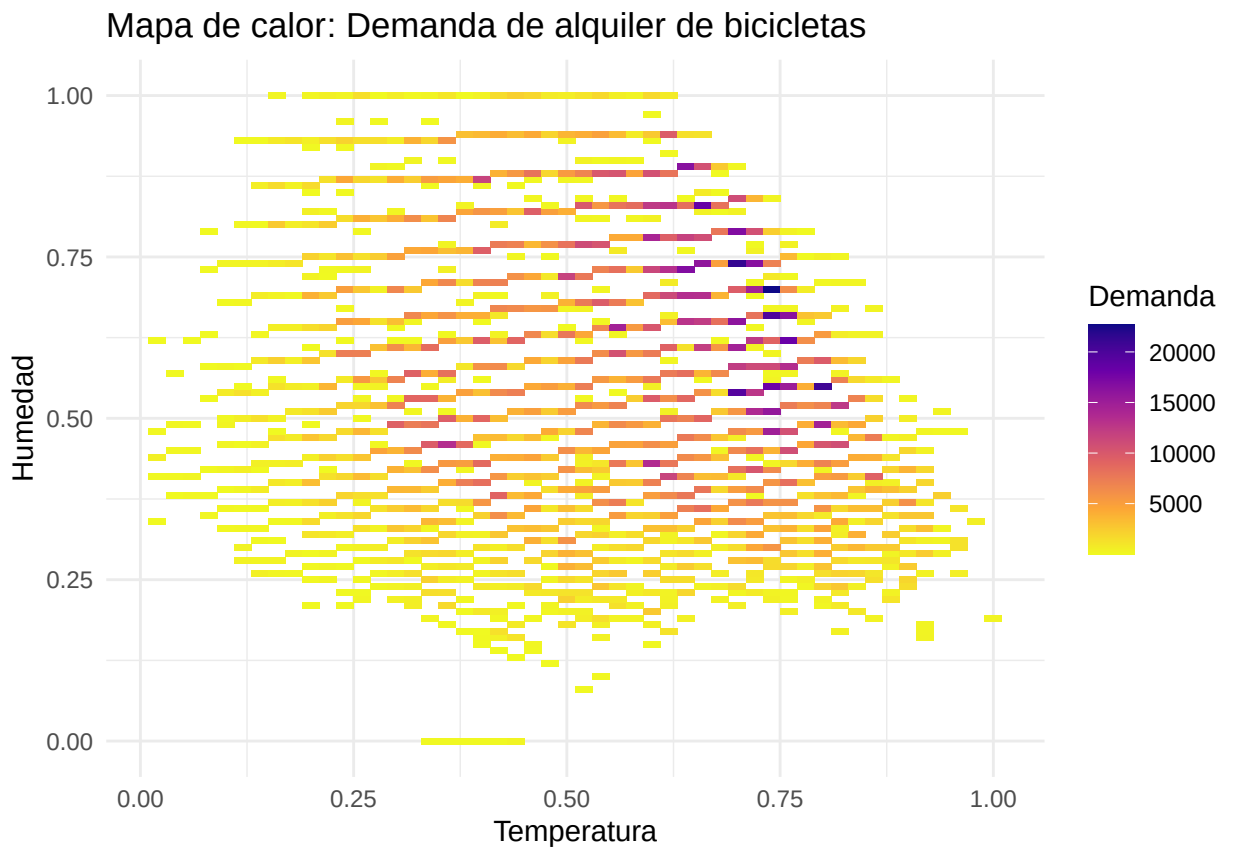
Disminuye a humedades muy bajas y algunas muy altas

7. ¿Que condiciones climáticas serian ideales para nuestra demanda? (considere una función de densidad bivariable para la temperatura y la humedad)

```
dataset_humedad_temperatura <- dataset %>%
                  select(hum, temp, cnt) %>%
                  group_by(hum, temp) %>%
                  summarize(demanda = sum(cnt))
```

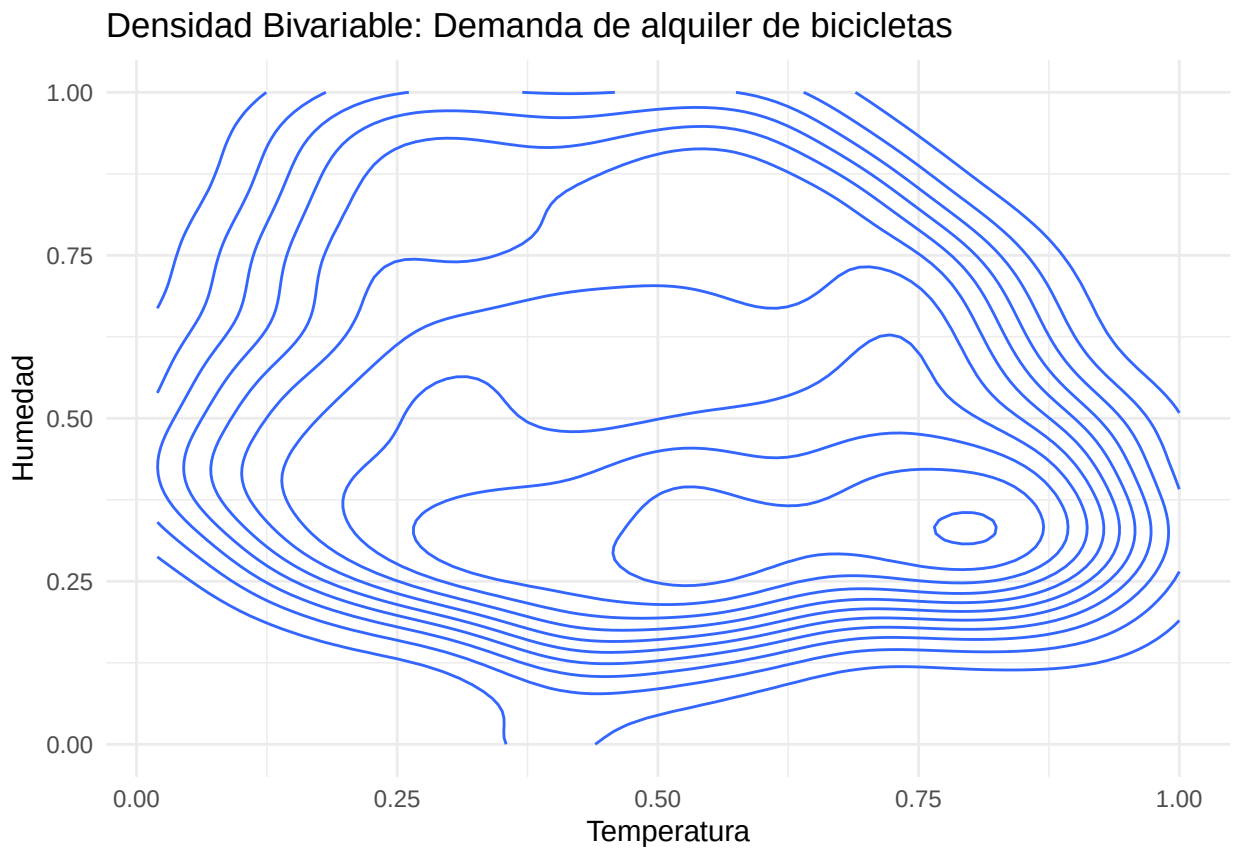
```
## `summarise()` has grouped output by 'hum'. You can override using the `.groups`
## argument.
```

```
dataset_humedad_temperatura %>% ggplot(aes(x = temp, y = hum, fill = demanda)) +
  geom_tile() +
  scale_fill_viridis_c(option = "plasma", direction = -1) +
  labs(title = "Mapa de calor: Demanda de alquiler de bicicletas",
       x = "Temperatura",
       y = "Humedad",
       fill = "Demanda") +
  theme_minimal()
```




```
dataset_humedad_temperatura %>% ggplot( aes(x = temp, y = hum, fill = demanda)) +
  geom_density_2d() +
  labs(title = "Densidad Bivariable: Demanda de alquiler de bicicletas",
       x = "Temperatura",
       y = "Humedad",
       fill = "Demanda") +
  theme_minimal()
```

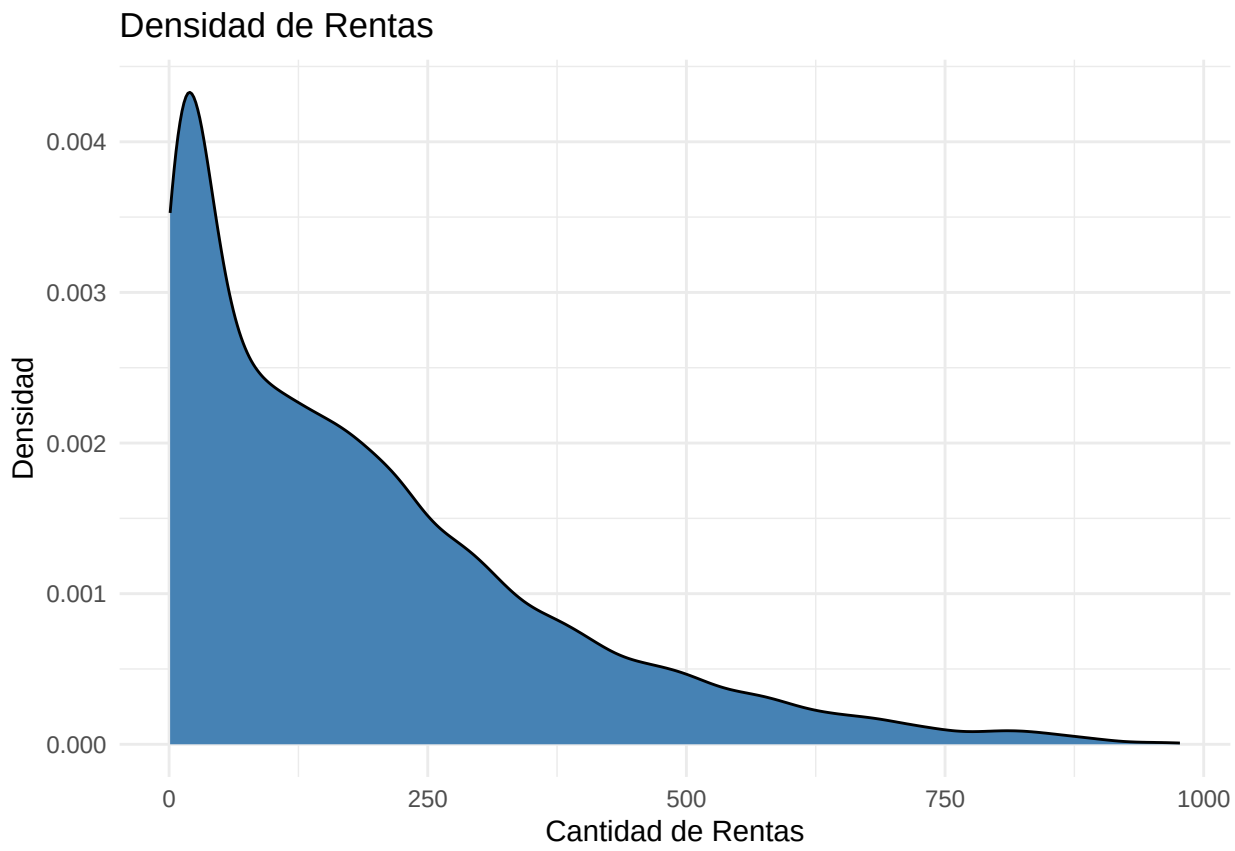
```
## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a ‘group‘ aesthetic or to convert a numerical
##   variable into a factor?
```



con una temperatura arriba del 0.75% y una humedad arriba de 75% en en los dos casos por debajo de 90%

8. Mueste una gráfica de la densidad de rentas.

```
dataset %>% ggplot( aes(x = cnt)) +
  geom_density(fill = "steelblue") +
  labs(title = "Densidad de Rentas",
       x = "Cantidad de Rentas",
       y = "Densidad") +
  theme_minimal()
```



9. ¿En promedio de personas que rentan bicicletas y están registradas?

```
promedio_personas_registradas <- dataset %>%
  filter(registered > 0) %>%
  summarise(promedio = mean(cnt))

promedio_personas_registradas$promedio
```

```
## [1] 189.7231
```

10. Determine la mediana de personas que rentan bicicletas y no están registradas.

```
mediana_personas_no_registradas <- dataset %>%
  filter(registered == 0) %>%
  summarise(mediana = median(cnt))

mediana_personas_no_registradas$mediana
```

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

11. Deterimne la renta total, renta promedio por cada tipo de estación.

No existen tipos de estacion solo dare la total

```
renta_total <- sum(dataset$cnt)
renta_promedio <- mean(dataset$cnt)

print(paste("Renta total:", renta_total))
```

```
## [1] "Renta total: 3292679"
```

```
print(paste("Renta promedio:", renta_promedio))
```

```
## [1] "Renta promedio: 189.463087634501"
```