Exploratory Data Analysis - Seoul Bike Sharing Data

Carichiamo i pacchetti necessari:

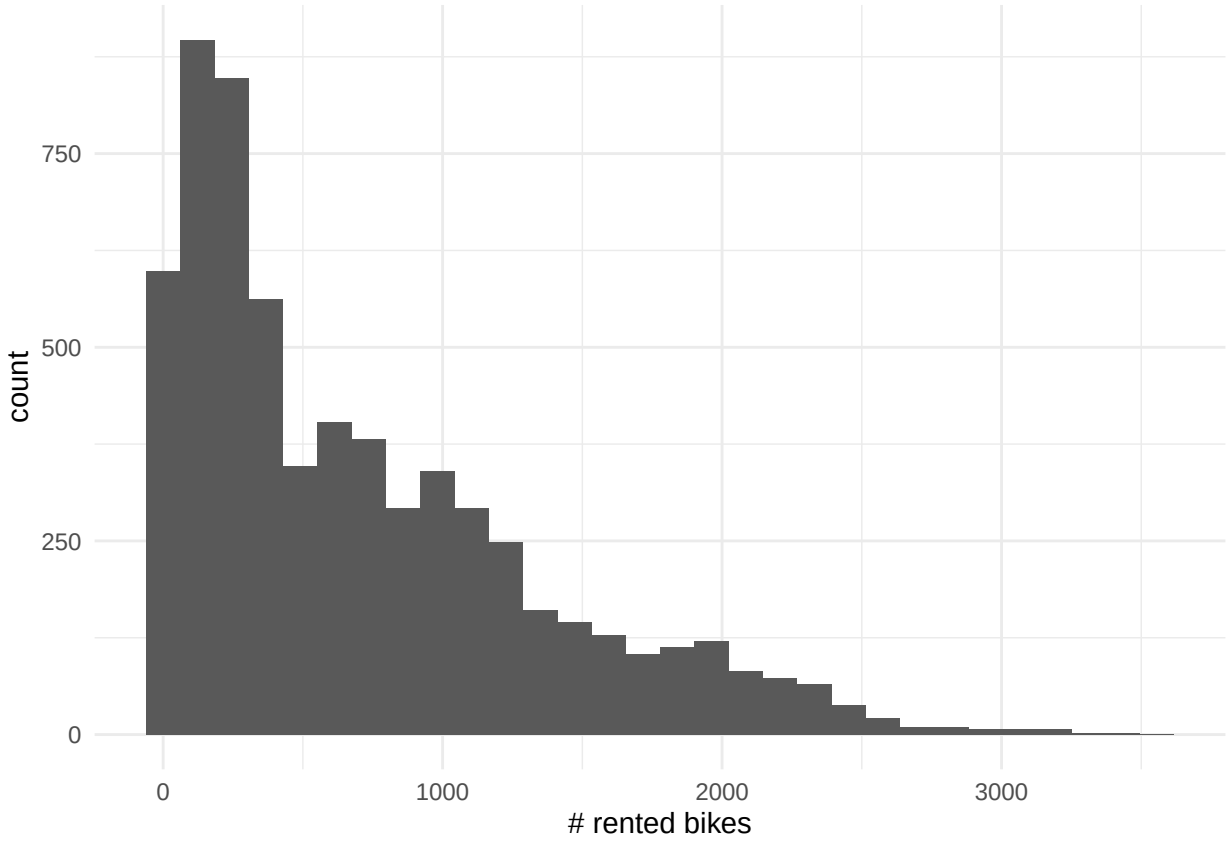
Carichiamo i dati:

Andiamo ad analizzare i dati in modo da identificarne delle proprietà utili alla previsione del numero di bici affittate ogni ora.

Descrizione del dataset

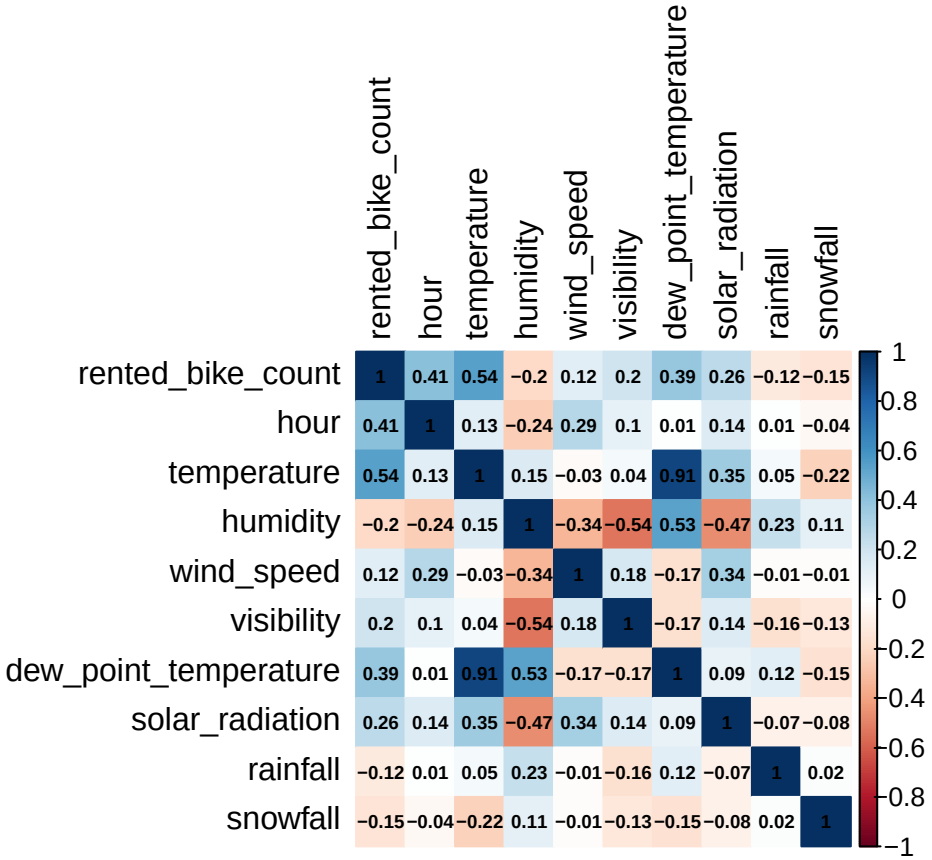
```
bike_train %>%
  ggplot() +
  geom_histogram(aes(rented_bike_count)) +
  xlab("# rented bikes")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Feature selection

```
corrplot(cor(bike_train[, 1:10]), method = "color", type = "full",
         addCoef.col = 1, tl.col = 1, number.cex = 0.6)
```

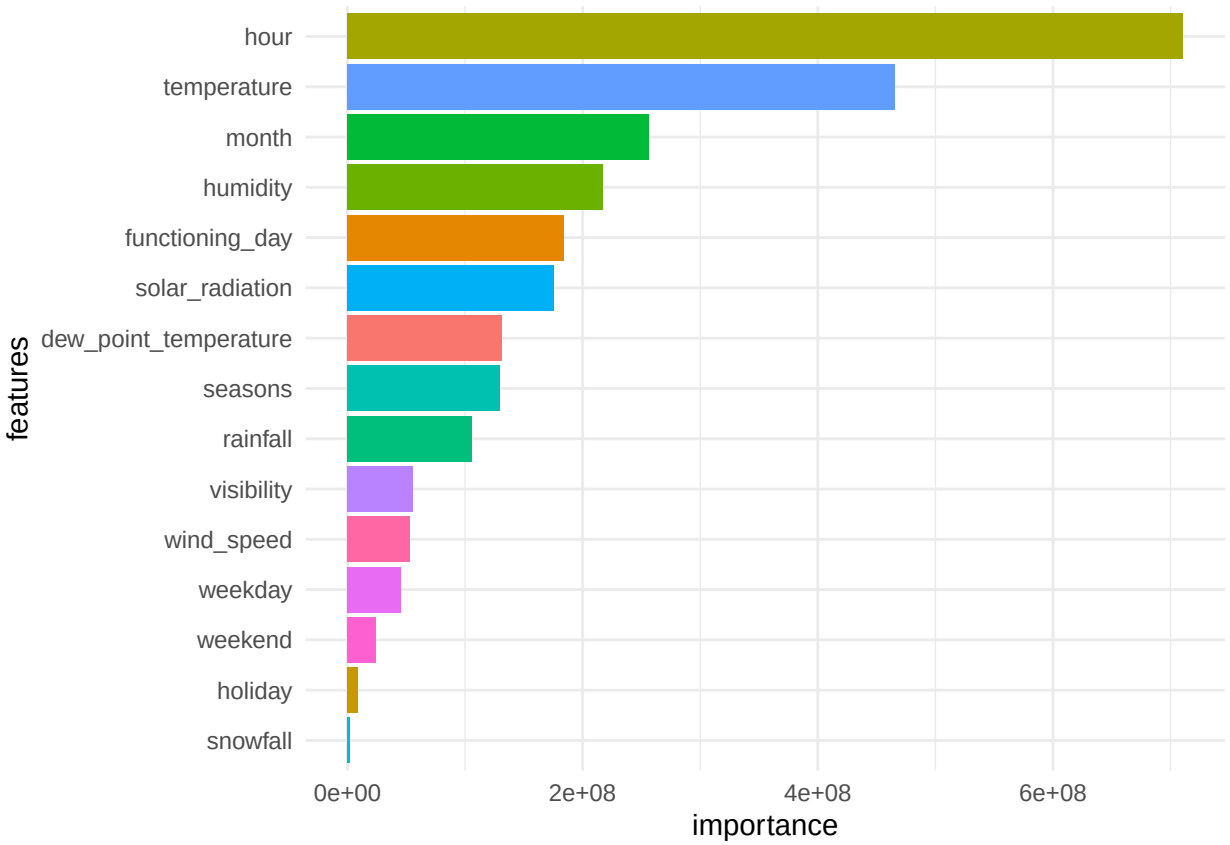


Correlazione lineare

```
##### Random forest #####
bike_rf1 <- readRDS("models/bike_rf1.rda")
```

Random forest Importance plot:

```
vimp_plot(bike_rf1) +
  xlab("features")
```



Random forest 2 Random forest applicata sui dati con i le variabili categoriche trasformate in dummy:

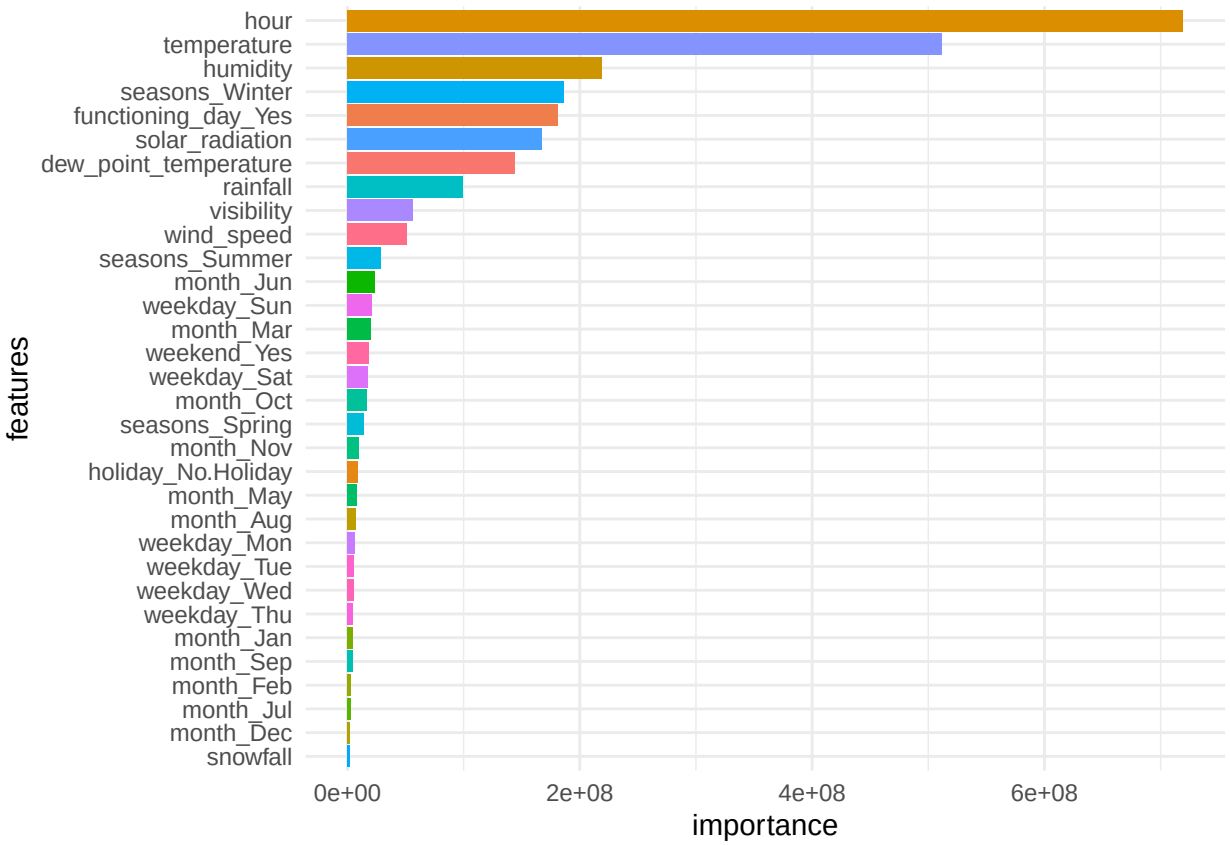
```
bike_train_dummy <- read_csv("data/bike_train_dummy.csv")
```

```
## Rows: 6307 Columns: 33
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (33): hour, temperature, humidity, wind_speed, visibility, dew_point_tem...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
bike_dummy_rf <- readRDS("models/bike_dummy_rf.rda")
```

Importance plot:

```
vimp_plot(bike_dummy_rf) +
  xlab("features")
```

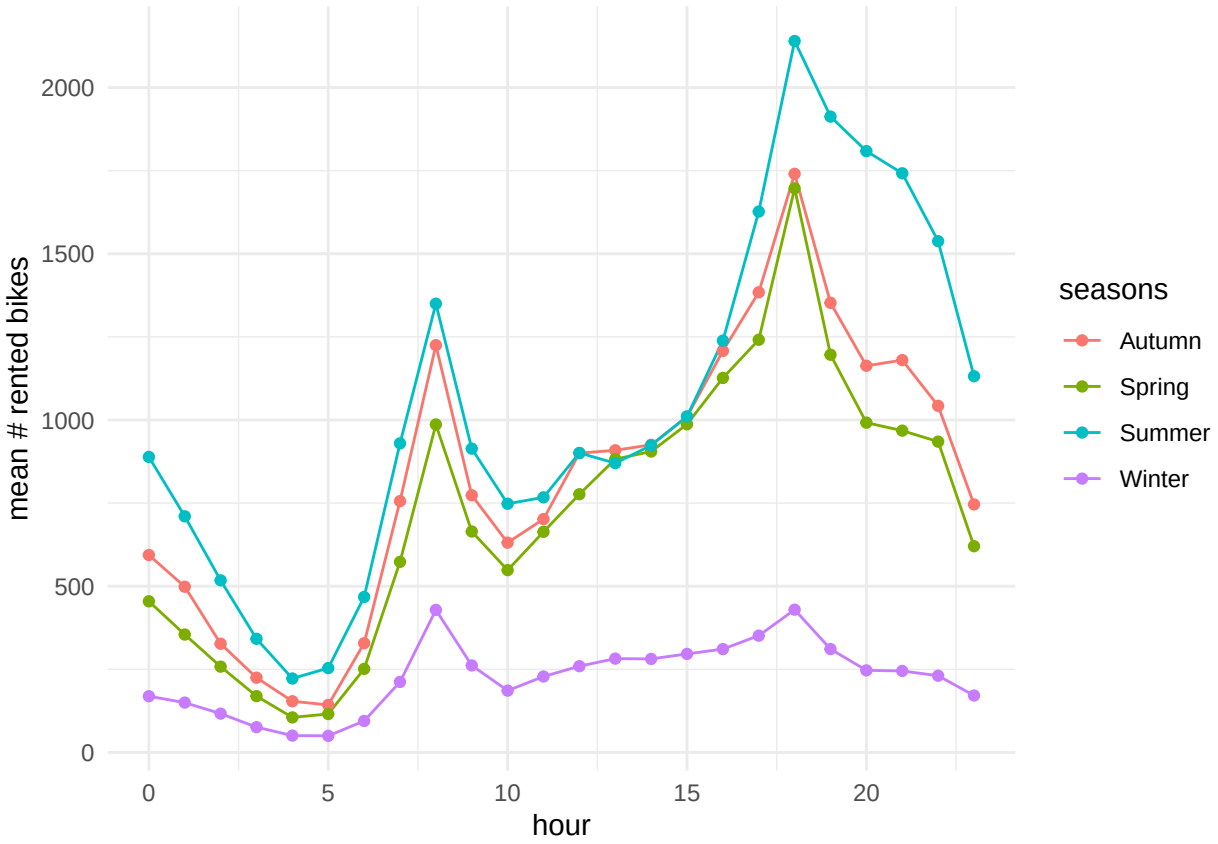


Facciamo i grafici in base all'importanza delle variabili trovata dalla random forest

```
bike_train %>%
  group_by(hour, seasons) %>%
  summarize(
    mean_rented_bike_count = mean(rented_bike_count)
  ) %>%
  ggplot(aes(hour, mean_rented_bike_count, group = seasons, color = seasons)) +
  geom_line() +
  geom_point() +
  ylab("mean # rented bikes")
```

Plot 1 - media rented_bike_count raggruppato per hour, season

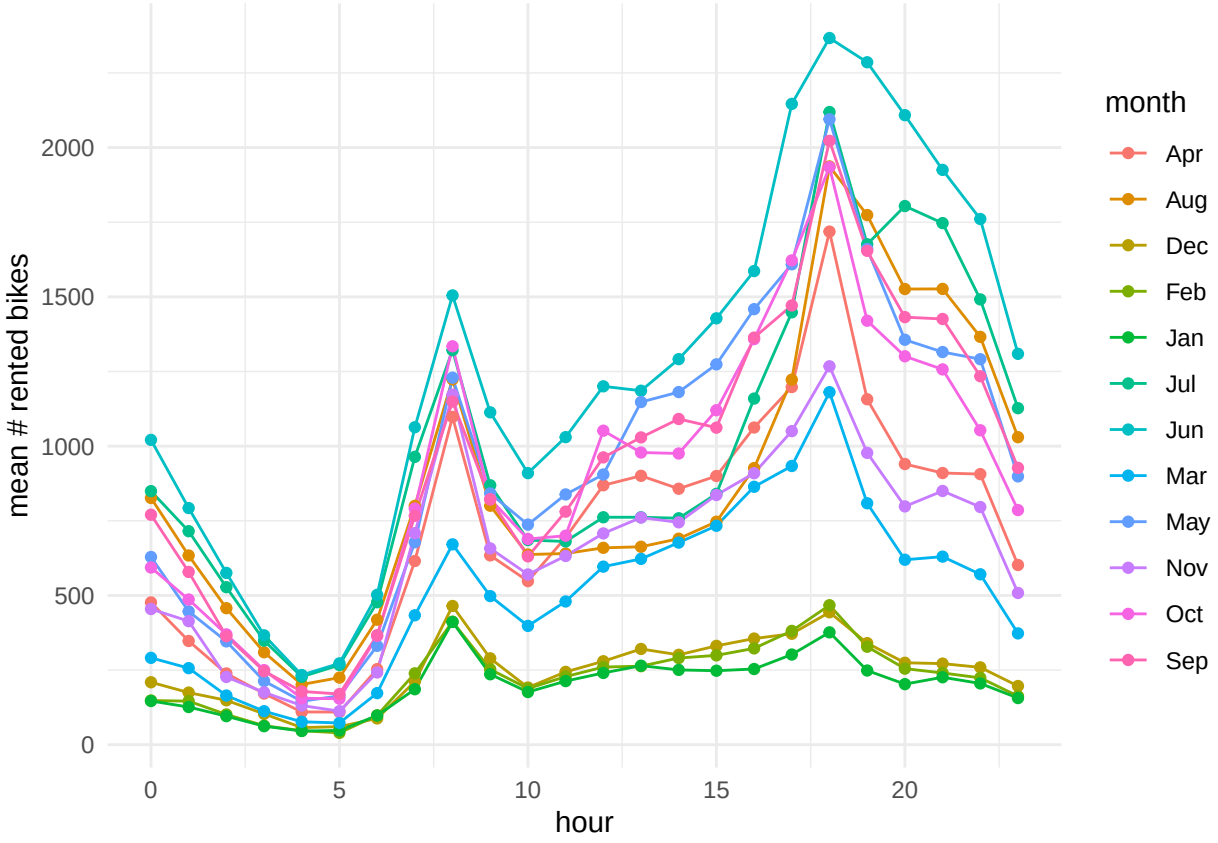
```
## `summarise()` has grouped output by 'hour'. You can override using the
## `.groups` argument.
```




```
bike_train %>%
  group_by(hour, month) %>%
  summarize(
    mean_rented_bike_count = mean(rented_bike_count)
  ) %>%
  ggplot(aes(hour, mean_rented_bike_count, group = month, color = month)) +
  geom_line() +
  geom_point() +
  ylab("mean # rented bikes")
```

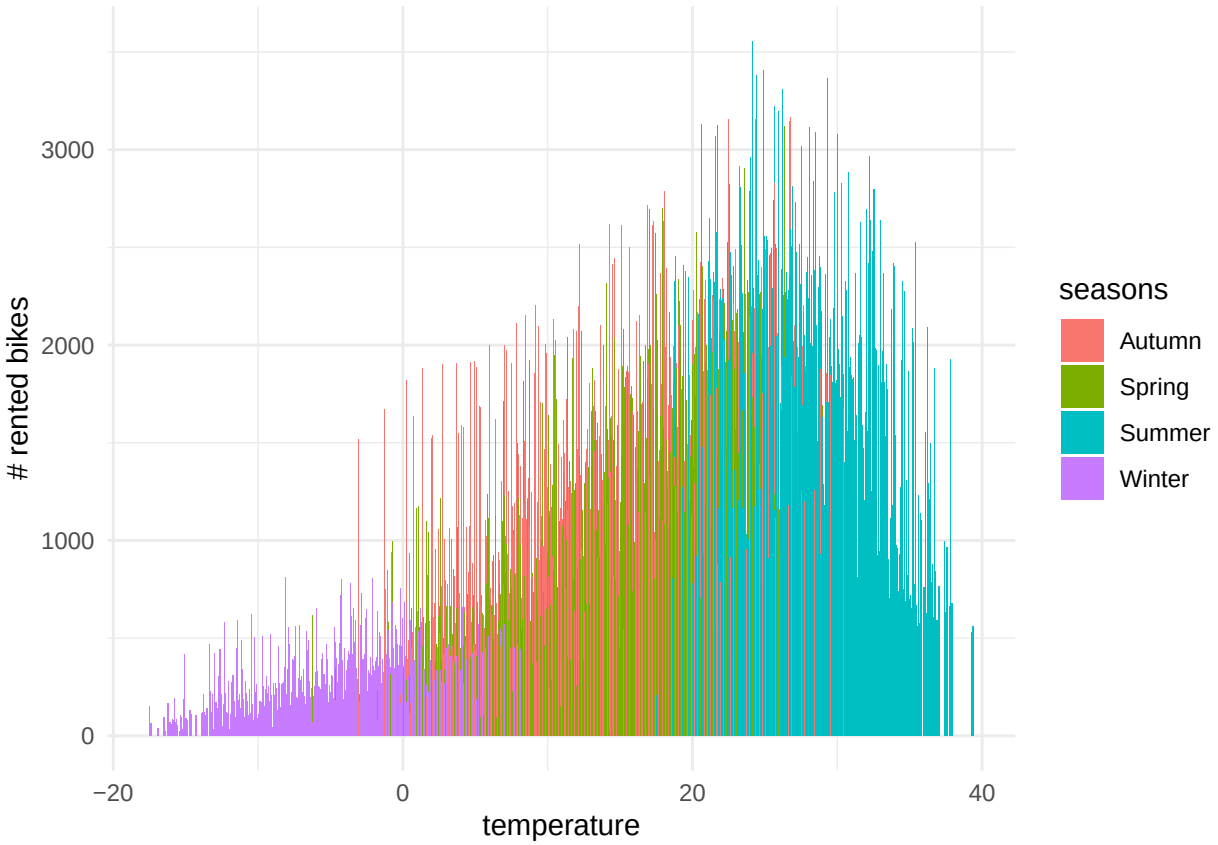
Plot 2 - media rented_bike_count raggruppato per hour, month

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



Plot 3 - distribuzione di rented_bike_count in base alla temperatura Colora per stagione:

```
bike_train %>%
  ggplot() +
  geom_col(aes(temperature, rented_bike_count, fill = seasons),
           position = "dodge2") +
  ylab("# rented bikes")
```


3000
2000
1000
0
rented bikes
−20
0
20
40
temperature
seasons
Autumn
Spring
Summer
Winter