

R Notebook

Importación de librerías.

Cargamos la base de datos.

```
## [1] "X"           "city"         "area"         "rooms"
## [5] "bathroom"    "parking.spaces" "floor"        "animal"
## [9] "furniture"   "hoa"          "rent.amount"  "property.tax"
## [13] "fire.insurance" "total"
```

Selección de variables.

Resumen rápido del dataframe.

```
summary(df)
```

```
## rent.amount      fire.insurance      city      area
## Length:6080      Length:6080      Min.   :0.0000  Min.   : 10.0
## Class :character  Class :character  1st Qu.:1.0000  1st Qu.: 58.0
## Mode  :character  Mode  :character  Median :1.0000  Median : 100.0
##                                     Mean  :0.8633  Mean  : 151.1
##                                     3rd Qu.:1.0000  3rd Qu.: 200.0
##                                     Max.   :1.0000  Max.   :24606.0
##      rooms      bathroom  parking.spaces  floor
## Min.   : 1.000  Min.   : 1.000  Min.   : 0.000  Length:6080
## 1st Qu.: 2.000  1st Qu.: 1.000  1st Qu.: 1.000  Class :character
## Median : 3.000  Median : 2.000  Median : 1.000  Mode  :character
## Mean   : 2.493  Mean   : 2.342  Mean   : 1.756
## 3rd Qu.: 3.000  3rd Qu.: 3.000  3rd Qu.: 2.000
## Max.   :10.000  Max.   :10.000  Max.   :12.000
##      animal      furniture
## Length:6080      Length:6080
## Class :character  Class :character
## Mode  :character  Mode  :character
##
##
##
```

Las variables que son numéricas las muestra como si fueran tipo carácter. Debido a que cuentan con caracteres especiales. Que impiden que se consideren como valores numéricos. Por lo cual tendremos que limpiar las variables como el **precio de la renta mensual**. **El seguro contra incendios**. Así como la cantidad de **pisos** del departamento.

Eliminación de caracteres especiales.

Tranfromación a variable numérica.

```
unique(df$floor)
```

```
## [1] "-" "10" "3" "12" "2" "16" "6" "4" "1" "7" "13" "9" "14" "5" "8"
## [16] "15" "11" "19" "20" "24" "23" "17" "18" "22" "27" "85" "28" "25" "29" "35"
## [31] "21" "31" "99" "26" "68" "32" "51"
```

Observamos los valores únicos. Marca un carácter del tipo cadena "-". Por lo cual indica que el departamento sólo cuenta con un piso.

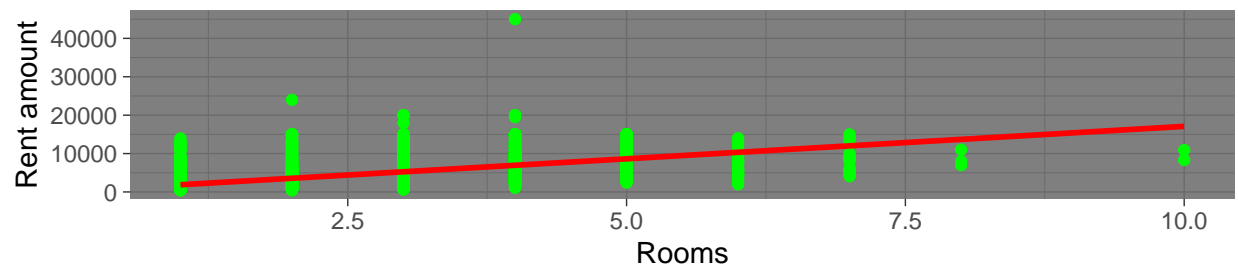
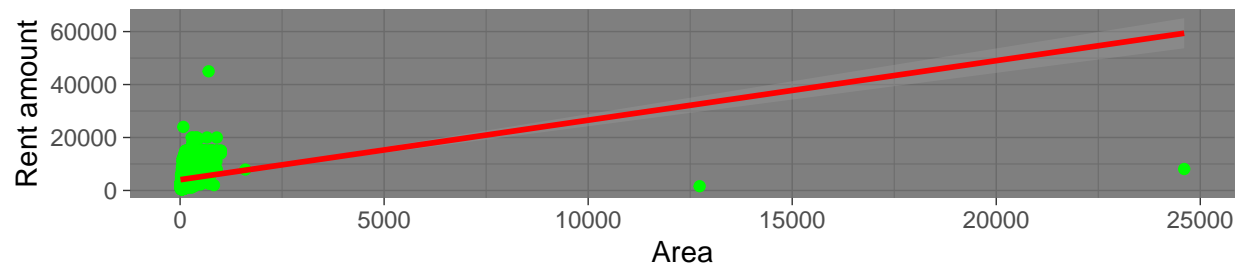
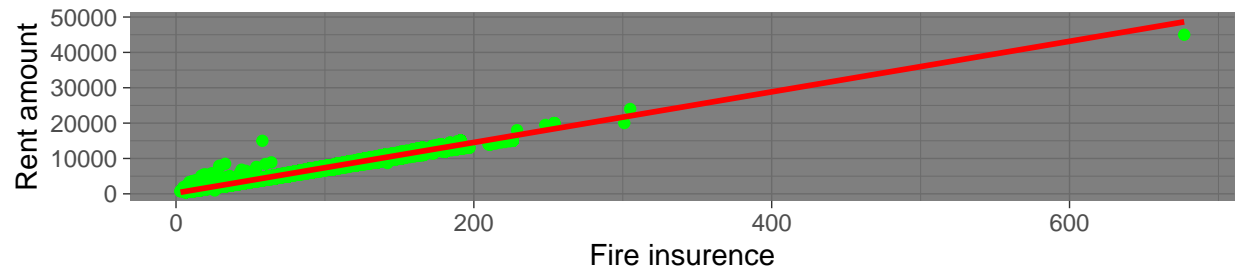
```
##   rent.amount   fire.insurance      city      area
##   Min.      : 420   Min.      : 3.0   Min.      :0.0000   Min.      : 10.0
##   1st Qu.: 1800   1st Qu.: 23.0   1st Qu.:1.0000   1st Qu.: 58.0
##   Median : 3111   Median : 41.0   Median :1.0000   Median : 100.0
##   Mean   : 4396   Mean   : 58.2   Mean   :0.8633   Mean   : 151.1
##   3rd Qu.: 5952   3rd Qu.: 77.0   3rd Qu.:1.0000   3rd Qu.: 200.0
##   Max.    :45000   Max.    :677.0   Max.    :1.0000   Max.    :24606.0
##   rooms      bathroom      parking.spaces      floor
##   Min.      : 1.000   Min.      : 1.000   Min.      : 0.000   Min.      : 0.000
##   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 0.000
##   Median : 3.000   Median : 2.000   Median : 1.000   Median : 4.000
##   Mean   : 2.493   Mean   : 2.342   Mean   : 1.756   Mean   : 5.672
##   3rd Qu.: 3.000   3rd Qu.: 3.000   3rd Qu.: 2.000   3rd Qu.: 9.000
##   Max.    :10.000   Max.    :10.000   Max.    :12.000   Max.    :99.000
```

Observamos que exitosamente realizamos la transformación numérica.

Limpieza de datos.

```
grid.arrange(
  scatter_plot(df$fire.insurance,"Fire insurence"),
  scatter_plot(df$area,"Area"),
  scatter_plot(df$rooms,"Rooms")
)
```

```
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
```

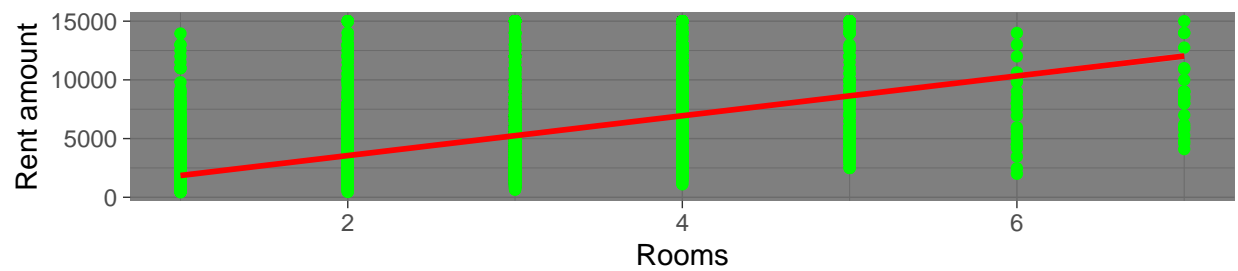
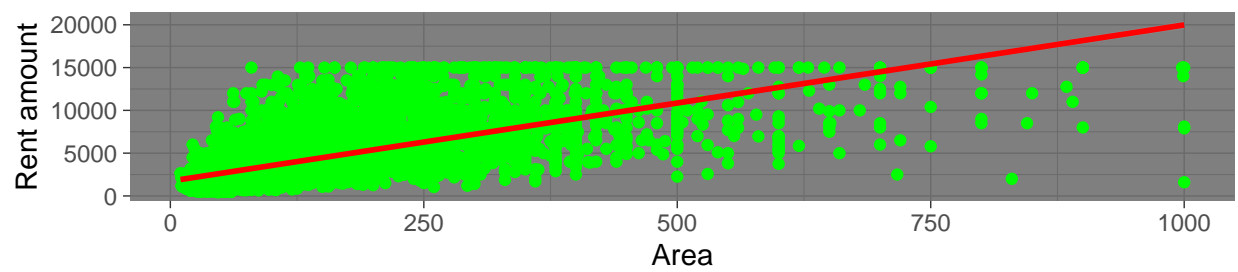
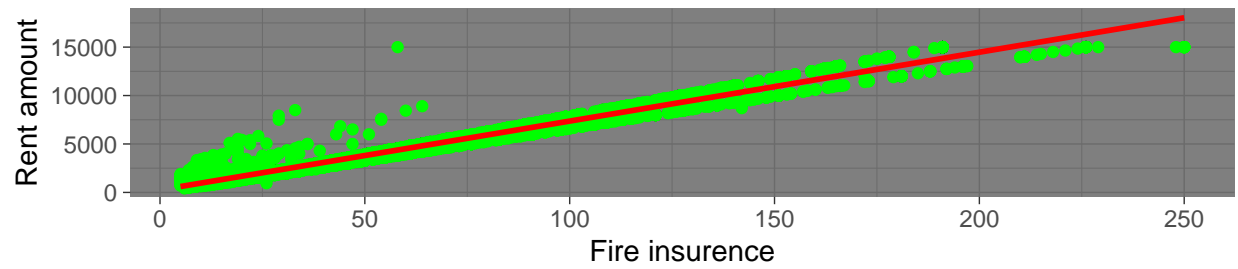


```
df<-df %>%
  mutate(rent.amount=ifelse(rent.amount>15000,15000,rent.amount)) %>%
  mutate(area=ifelse(area>1000,1000,area)) %>%
  mutate(fire.insurance=ifelse(fire.insurance>250,250,fire.insurance)) %>%
  mutate(fire.insurance=ifelse(fire.insurance<5,5,fire.insurance)) %>%
  mutate(rooms=ifelse(rooms>7,7,rooms))
```

Reemplazamos los valores atípicos por valores normales.

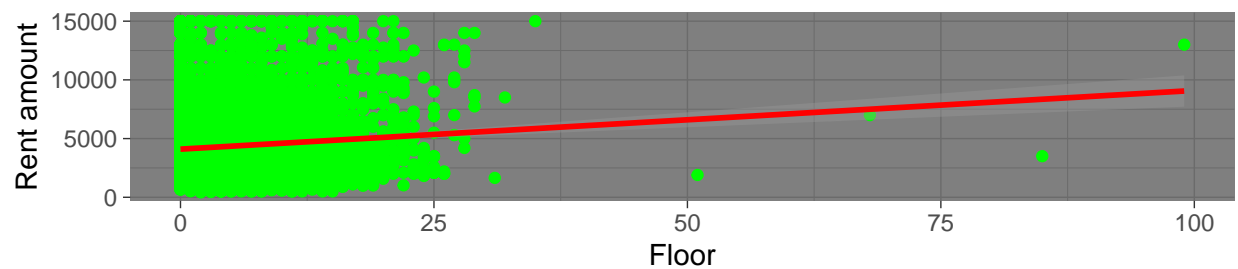
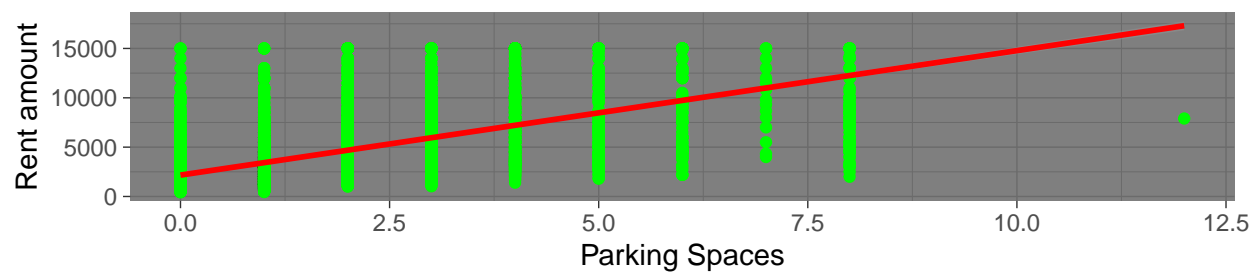
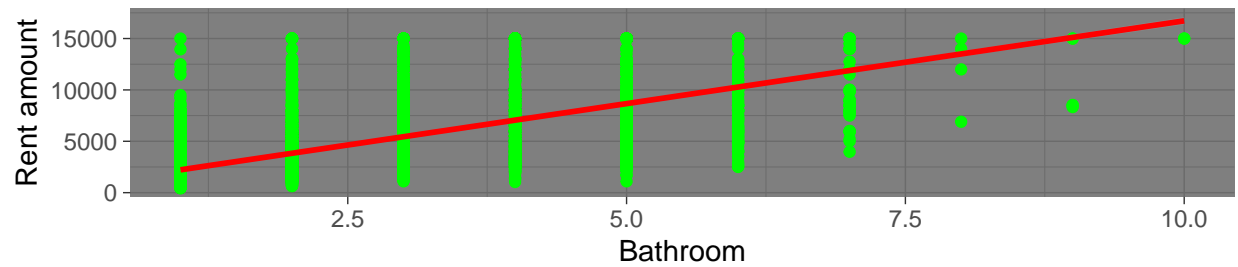
```
grid.arrange(
  scatter_plot(df$fire.insurance,"Fire insurence"),
  scatter_plot(df$area,"Area"),
  scatter_plot(df$rooms,"Rooms")
)
```

```
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
```



```
grid.arrange(scatter_plot(df$bathroom,"Bathroom"),
              scatter_plot(df$parking.spaces,"Parking Spaces"),
              scatter_plot(df$floor,"Floor"))
```

```
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
```

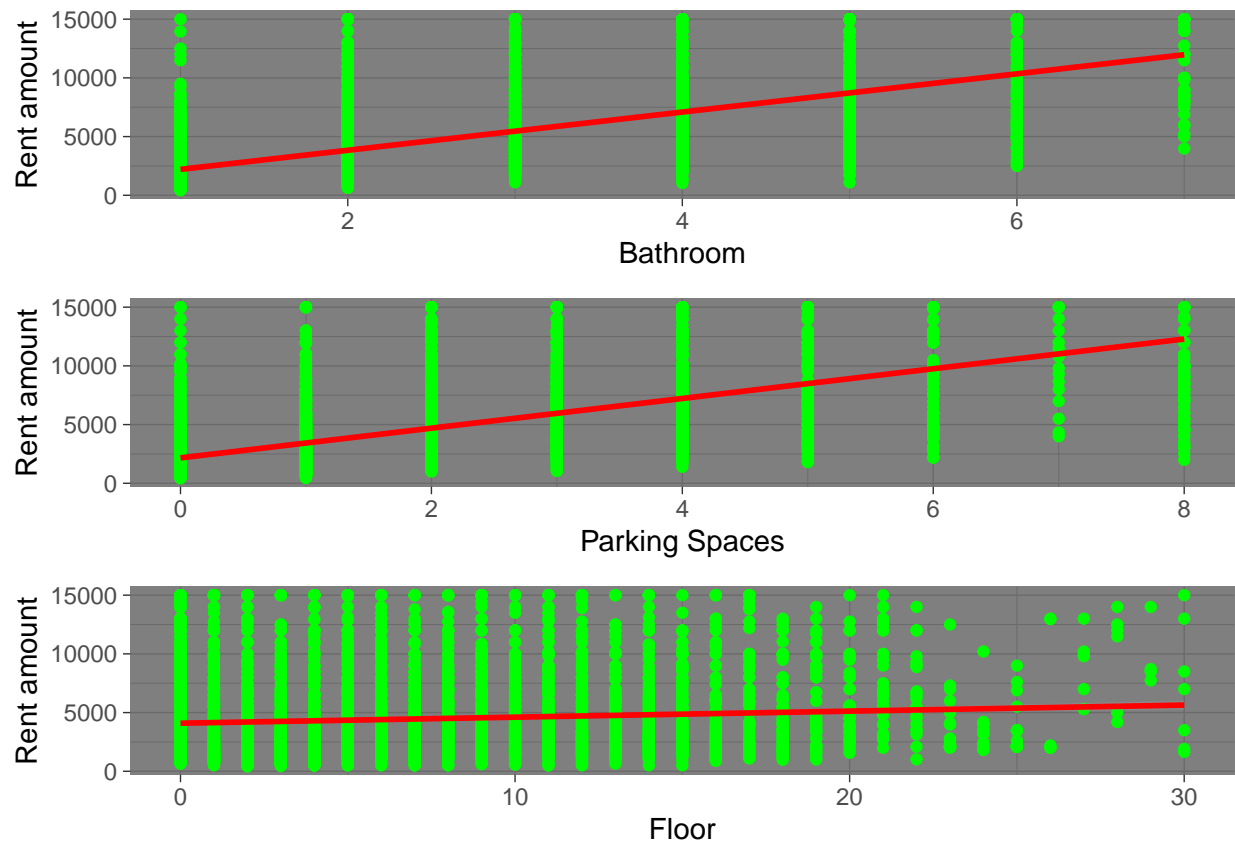


```
df<-df %>%
  mutate(bathroom=ifelse(bathroom>7,7,bathroom)) %>%
  mutate(parking.spaces=ifelse(parking.spaces>10,8,parking.spaces)) %>%
  mutate(floor=ifelse(floor>30,30,floor))
```

Reemplazamos los valores que se salen de lo normal. Por valores normales.

```
grid.arrange(scatter_plot(df$bathroom,"Bathroom"),
             scatter_plot(df$parking.spaces,"Parking Spaces"),
             scatter_plot(df$floor,"Floor"))
```

```
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
```

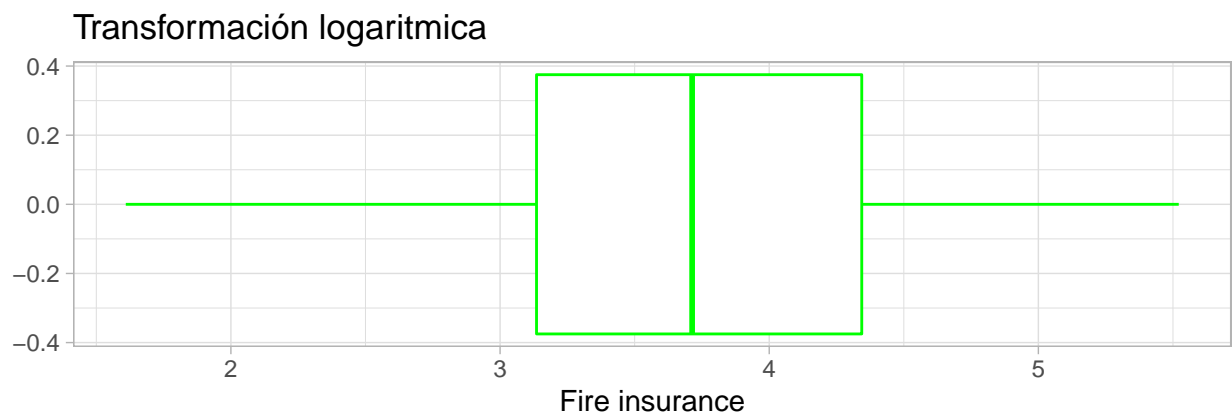
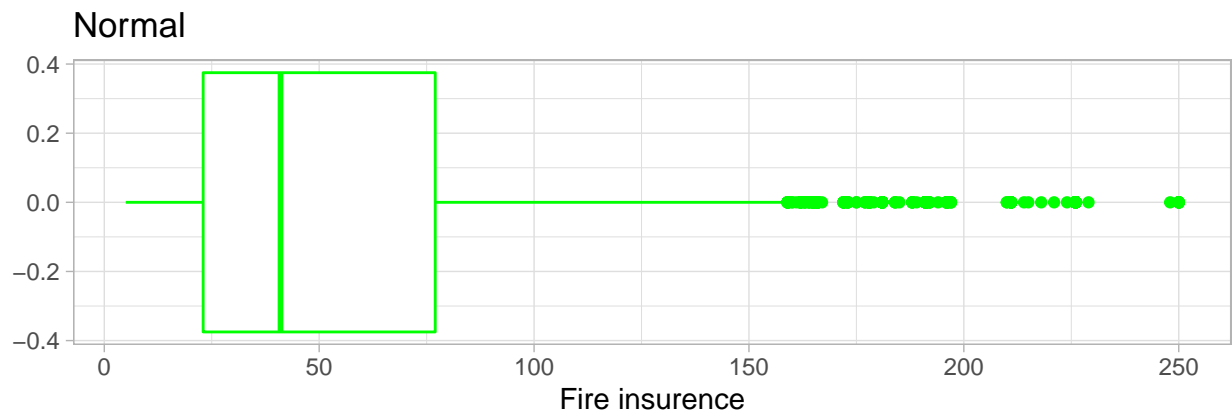


```
box_plot<-function(x_var,x_name,title){
  ggplot(df,aes(x=x_var)) +
  geom_boxplot(color="green") +
  theme_light() +
  xlab(x_name)+
  labs(title = title)
}
```

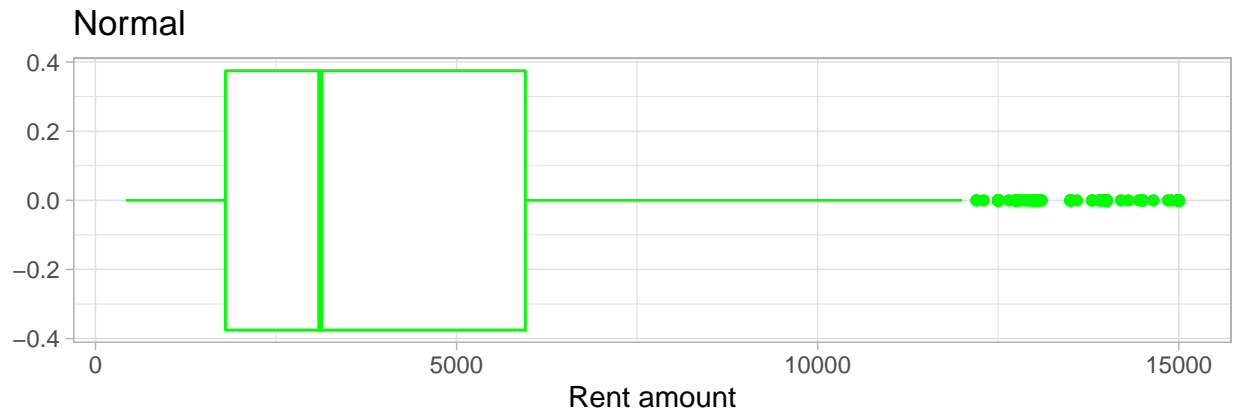
Transformación logarítmica.

Realizamos dicha transformación para poder mejorar la distribución de los datos.

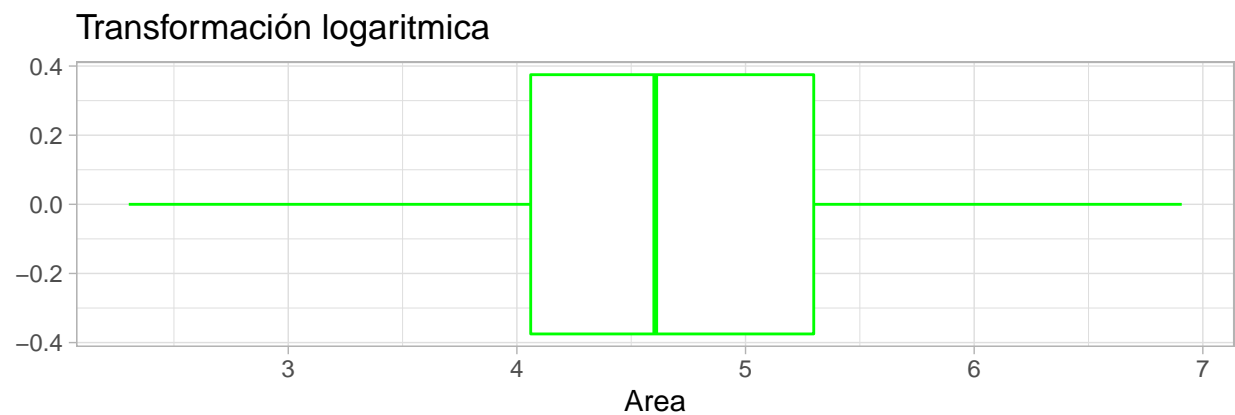
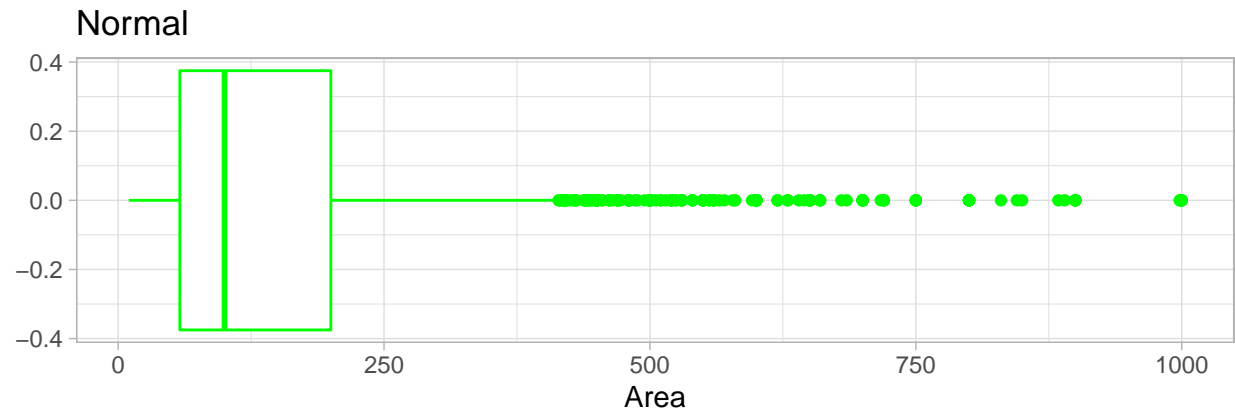
```
grid.arrange(box_plot(df$fire.insurance,"Fire insurance","Normal"),
  box_plot(log(df$fire.insurance),"Fire insurance","Transformación logarítmica"))
```



```
grid.arrange(box_plot(df$rent.amount,"Rent amount","Normal"),
              box_plot(log(df$rent.amount),"Rent amount","Transformación logaritmica"))
```



```
grid.arrange(box_plot(df$area, "Area", "Normal"),  
              box_plot(log(df$area), "Area", "Transformación logarítmica"))
```

```
df<- df %>%
  mutate(rent.amount=log(rent.amount),
         area=log(area),
         fire.insurance=log(fire.insurance))
```

Guardamos la base de datos limpia.

```
write.table(df,file = "houses_to_rent_clear.csv",sep=";",row.names = F)
```