

# Proyeccion

Grupo No4

2024-04-20

## Contents

Set working directory	1
R Markdown	30

## Set working directory

```
# Set working directory
setwd("D:/1.Maestria Ciencia Datos/03. INTRODUCCIÓN A LOS MODELOS ESTADÍSTICOS-23MCDAP002-PSMA-00609-19")
# Importing the dataset
data <- read_excel("FINAL TOTAL.xlsx", sheet = "Proyección")
# Drop the last row
data <- data[-nrow(data), ]

#Head
head(data)

## # A tibble: 6 x 16
##   Lote Empresa Fundo Variedad Area Color `Fecha de Siembra` `Plantas por Ha`
##   <chr> <chr> <chr> <chr> <dbl> <chr> <dtm> <dbl>
## 1 CVH-01 Agrico~ Cuat~ Hass      8.05 Negra 2019-10-16 00:00:00      452
## 2 CVH-02 Agrico~ Cuat~ Hass      9.62 Negra 2019-09-30 00:00:00      452
## 3 CVH-03 Agrico~ Cuat~ Hass      9.77 Negra 2019-09-23 00:00:00      452
## 4 CVH-04 Agrico~ Cuat~ Hass      9.58 Negra 2019-10-10 00:00:00      452
## 5 CVH-05 Agrico~ Cuat~ Hass      9.56 Negra 2019-10-28 00:00:00      452
## 6 CVH-06 Agrico~ Cuat~ Hass     10.6 Negra 2019-12-02 00:00:00      452
## # i 8 more variables: `Plantas por Lote` <dbl>, `Frutos por Planta` <dbl>,
## #   `Peso Promedio Fruto` <dbl>, `Kg Bruto Ha` <dbl>, `Kg Bruto Lote` <dbl>,
## #   `Kg Exportado Lote` <dbl>, `Kg Exportado Ha` <dbl>, Estatus <chr>

# Rename specific columns using dplyr's rename() function
data <- data %>%
  rename(
    FSiembra = `Fecha de Siembra`,
    PlantasxHa = `Plantas por Ha`,
    PlantasxLt = `Plantas por Lote`,
    FrutosxPl = `Frutos por Planta`,
    PesoPromFr = `Peso Promedio Fruto`,
    KgBrutoxHa = `Kg Bruto Ha`,
    KgBrutoxLt = `Kg Bruto Lote`,
    KgExpLt = `Kg Exportado Lote`,
    KgExpHa = `Kg Exportado Ha`,
```

```

# Continue renaming as needed
)
#Names
colnames(data)

## [1] "Lote"      "Empresa"   "Fundo"     "Variedad"  "Area"
## [6] "Color"     "FSiembra"  "PlantasxHa" "PlantasxLt" "FrutosxPl"
## [11] "PesoPromFr" "KgBrutoxHa" "KgBrutoxLt" "KgExpxLt"  "KgExpxHa"
## [16] "Estatus"

#STR
str(data)

## tibble [33 x 16] (S3: tbl_df/tbl/data.frame)
## $ Lote      : chr [1:33] "CVH-01" "CVH-02" "CVH-03" "CVH-04" ...
## $ Empresa   : chr [1:33] "Agricola Guili S.A.C" "Agricola Guili S.A.C" "Agricola Guili S.A.C" "Agri
## $ Fundo     : chr [1:33] "Cuatro Vientos" "Cuatro Vientos" "Cuatro Vientos" "Cuatro Vientos" ...
## $ Variedad  : chr [1:33] "Hass" "Hass" "Hass" "Hass" ...
## $ Area      : num [1:33] 8.05 9.62 9.77 9.58 9.56 ...
## $ Color     : chr [1:33] "Negra" "Negra" "Negra" "Negra" ...
## $ FSiembra  : POSIXct[1:33], format: "2019-10-16" "2019-09-30" ...
## $ PlantasxHa: num [1:33] 452 452 452 452 452 452 452 452 452 452 ...
## $ PlantasxLt: num [1:33] 3640 4348 4416 4332 4320 ...
## $ FrutosxPl : num [1:33] 238 229 289 215 106 ...
## $ PesoPromFr: num [1:33] 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.22 ...
## $ KgBrutoxHa: num [1:33] 23632 22726 28698 21366 10588 ...
## $ KgBrutoxLt: num [1:33] 190299 218620 280404 204776 101191 ...
## $ KgExpxLt  : num [1:33] 161754 185827 238343 174059 86012 ...
## $ KgExpxHa  : num [1:33] 20087 19317 24393 18161 9000 ...
## $ Estatus   : chr [1:33] "Cosechando" "Cosechando" "Cosechando" "Cosechando" ...

# Check for missing data
sum(is.na(data))

## [1] 0

colnames(data)

## [1] "Lote"      "Empresa"   "Fundo"     "Variedad"  "Area"
## [6] "Color"     "FSiembra"  "PlantasxHa" "PlantasxLt" "FrutosxPl"
## [11] "PesoPromFr" "KgBrutoxHa" "KgBrutoxLt" "KgExpxLt"  "KgExpxHa"
## [16] "Estatus"

# Basic summary statistics
summary(data)

##      Lote      Empresa      Fundo      Variedad
## Length:33      Length:33      Length:33      Length:33
## Class :character Class :character Class :character Class :character
## Mode  :character Mode  :character Mode  :character Mode  :character
##
##
##      Area      Color      FSiembra
## Min.   : 6.553      Length:33      Min.   :2018-11-05 00:00:00.00
## 1st Qu.: 9.557      Class :character 1st Qu.:2019-10-10 00:00:00.00
## Median : 9.849      Mode  :character  Median :2020-03-12 00:00:00.00

```

```
## Mean :11.515 Mean :2020-03-10 02:10:54.54
## 3rd Qu.:11.697 3rd Qu.:2020-09-18 00:00:00.00
## Max. :26.570 Max. :2021-05-03 00:00:00.00
## PlantasxHa PlantasxLt FrutosxPl PesoPromFr
## Min. :452 Min. : 2962 Min. : 7.166 Min. :0.22
## 1st Qu.:452 1st Qu.: 4320 1st Qu.: 73.438 1st Qu.:0.22
## Median :452 Median : 4452 Median :191.960 Median :0.22
## Mean :452 Mean : 5205 Mean :156.898 Mean :0.22
## 3rd Qu.:452 3rd Qu.: 5287 3rd Qu.:217.078 3rd Qu.:0.22
## Max. :452 Max. :12010 Max. :342.276 Max. :0.22
## KgBrutoxHa KgBrutoxLt KgExpxLt KgExpxHa
## Min. : 712.6 Min. : 6954 Min. : 5911 Min. : 605.7
## 1st Qu.: 7302.7 1st Qu.: 88121 1st Qu.: 74903 1st Qu.: 6207.3
## Median :19088.5 Median :166768 Median :141753 Median :16225.2
## Mean :15601.9 Mean :179851 Mean :152873 Mean :13261.6
## 3rd Qu.:21586.3 3rd Qu.:235544 3rd Qu.:200213 3rd Qu.:18348.3
## Max. :34035.9 Max. :566796 Max. :481776 Max. :28930.5
## Estatus
## Length:33
## Class :character
## Mode :character
##
##
##
numeric_columns <- sapply(data, is.numeric)
data_numeric <- data[, numeric_columns]

print(numeric_columns)

## Lote Empresa Fundo Variedad Area Color FSiembra
## FALSE FALSE FALSE FALSE TRUE FALSE FALSE
## PlantasxHa PlantasxLt FrutosxPl PesoPromFr KgBrutoxHa KgBrutoxLt KgExpxLt
## TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## KgExpxHa Estatus
## TRUE FALSE

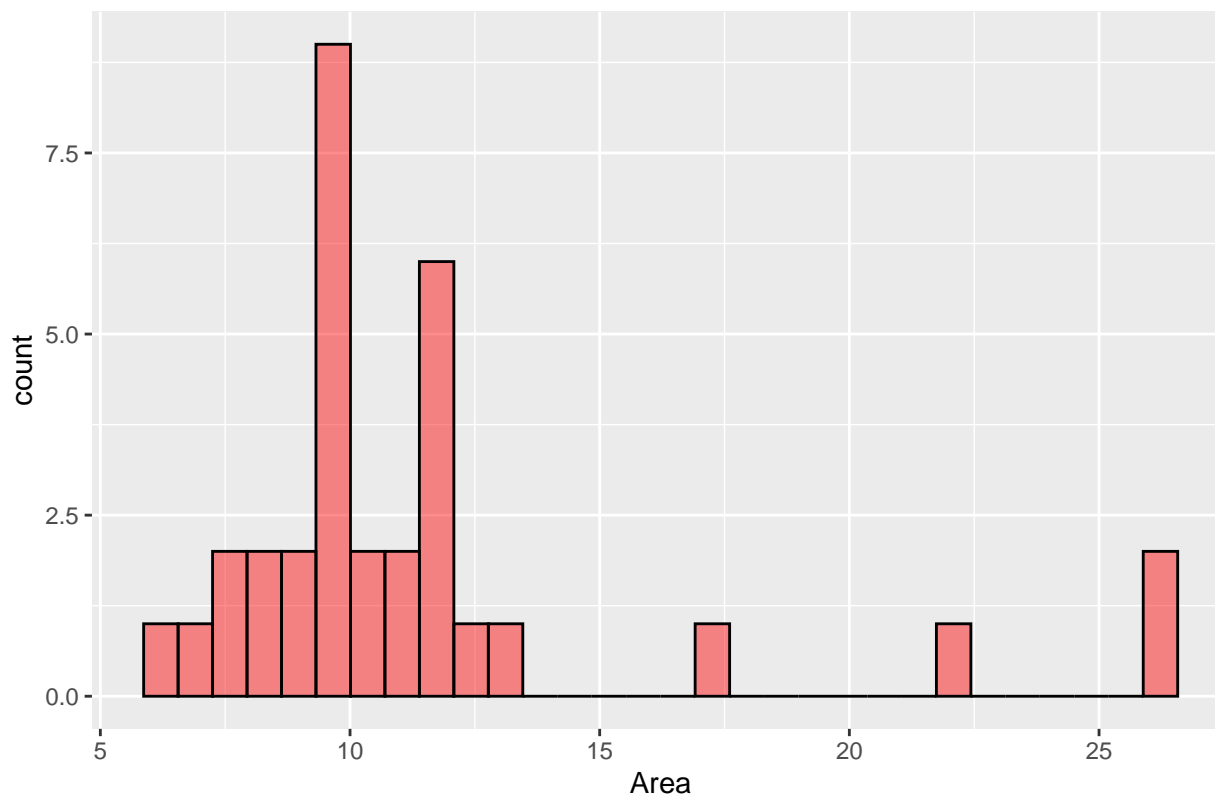
print(data_numeric)

## # A tibble: 33 x 9
## Area PlantasxHa PlantasxLt FrutosxPl PesoPromFr KgBrutoxHa KgBrutoxLt
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 8.05 452 3640. 238. 0.22 23632. 190299.
## 2 9.62 452 4348. 229. 0.22 22726. 218620.
## 3 9.77 452 4416. 289. 0.22 28698. 280404.
## 4 9.58 452 4332. 215. 0.22 21366. 204776.
## 5 9.56 452 4320. 106. 0.22 10588. 101191.
## 6 10.6 452 4787. 200. 0.22 19862. 210345.
## 7 9.16 452 4142. 96.7 0.22 9616. 88121.
## 8 9.57 452 4326. 56.5 0.22 5622. 53811.
## 9 10.2 452 4630. 20.7 0.22 2058. 21084.
## 10 7.27 452 3287. 342. 0.22 34036. 247547.
## # i 23 more rows
## # i 2 more variables: KgExpxLt <dbl>, KgExpxHa <dbl>
```

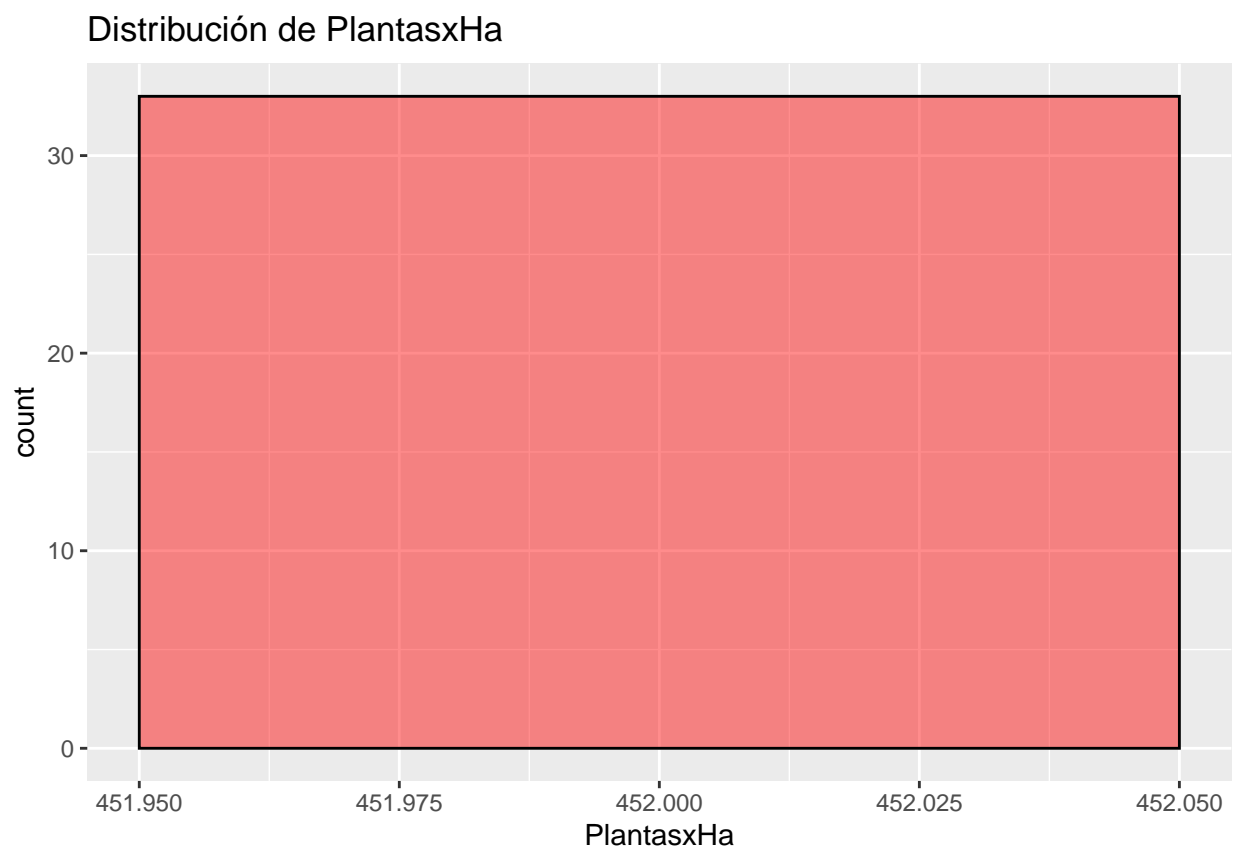
```
# Generating histograms
lapply(names(data_numeric), function(x) {
  ggplot(data, aes_string(x = x)) +
    geom_histogram(alpha=0.45, bins = 30, fill = "red", color = "black") +
    labs(title = paste("Distribución de", x))
})

## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
## [[1]]
```

Distribución de Area

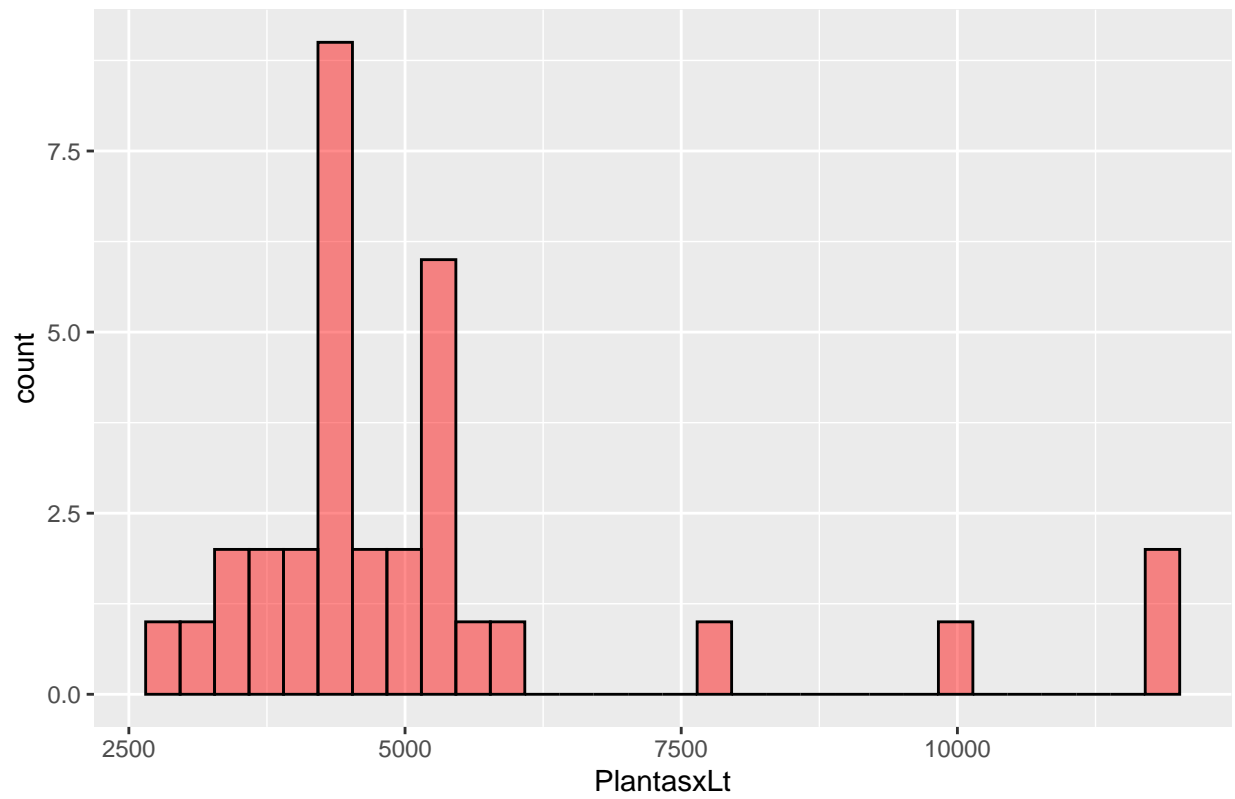


```
##
## [[2]]
```

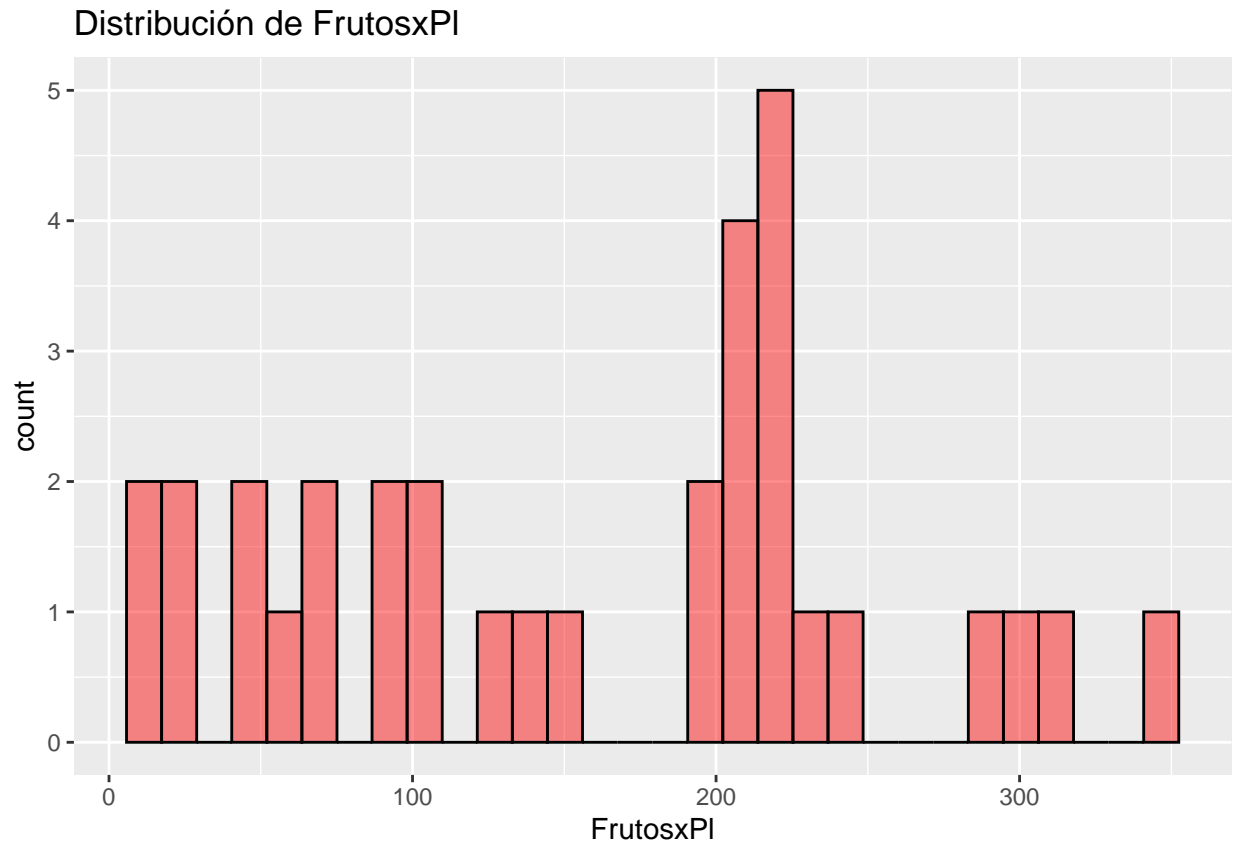


```
##  
## [[3]]
```

Distribución de PlantasxLt

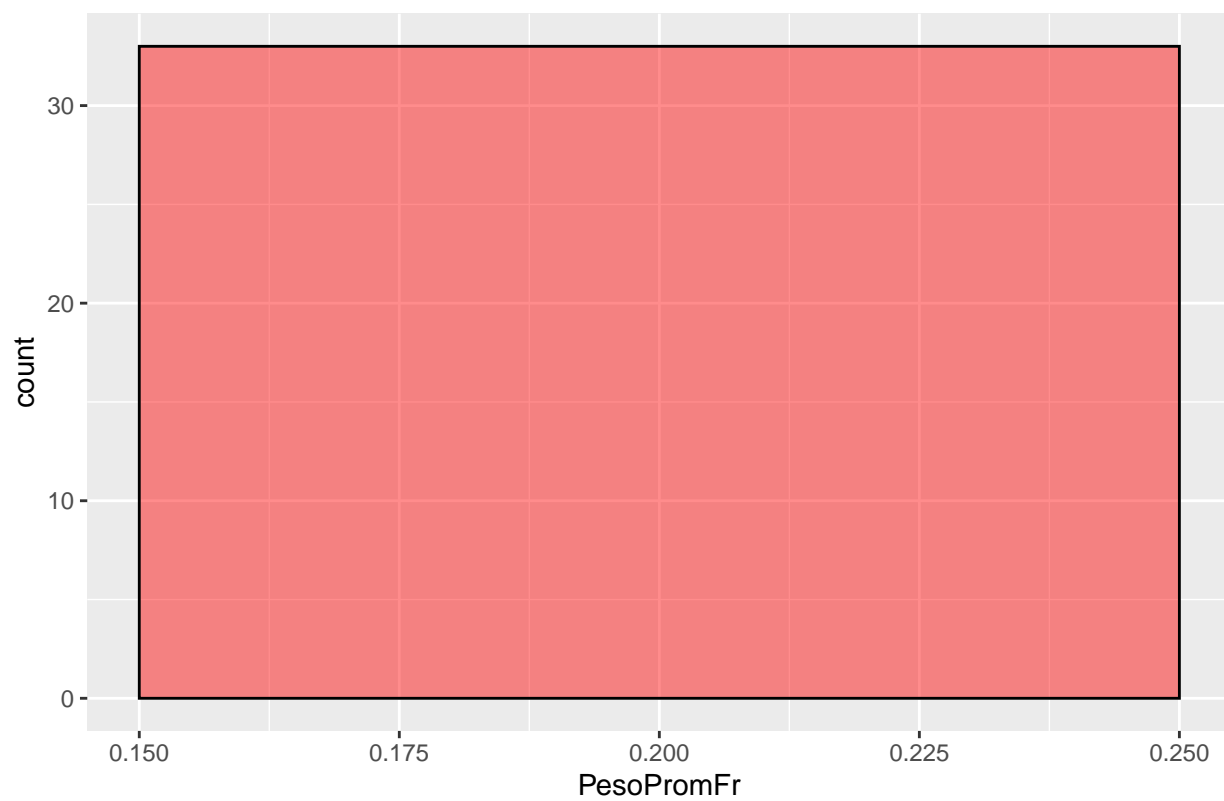


```
##  
## [[4]]
```



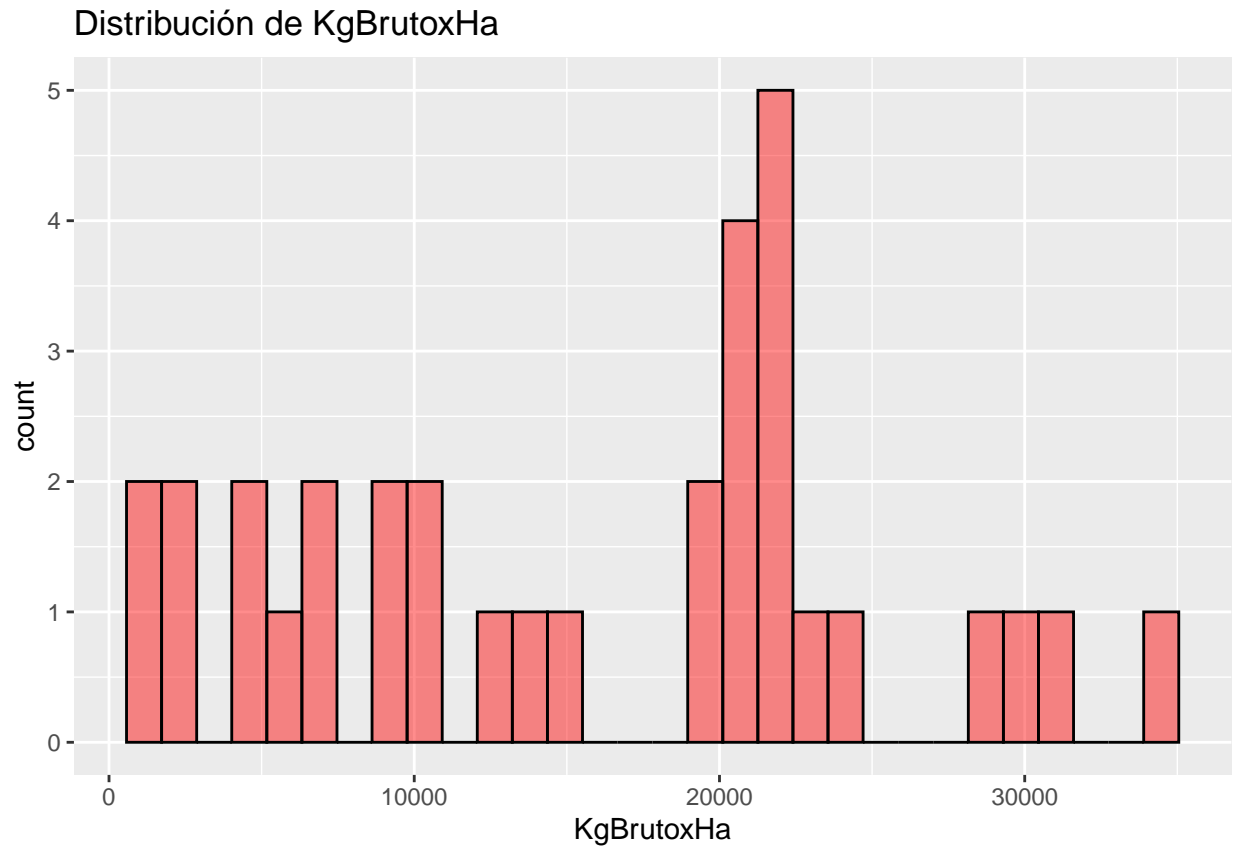
```
##  
## [[5]]
```

Distribución de PesoPromFr

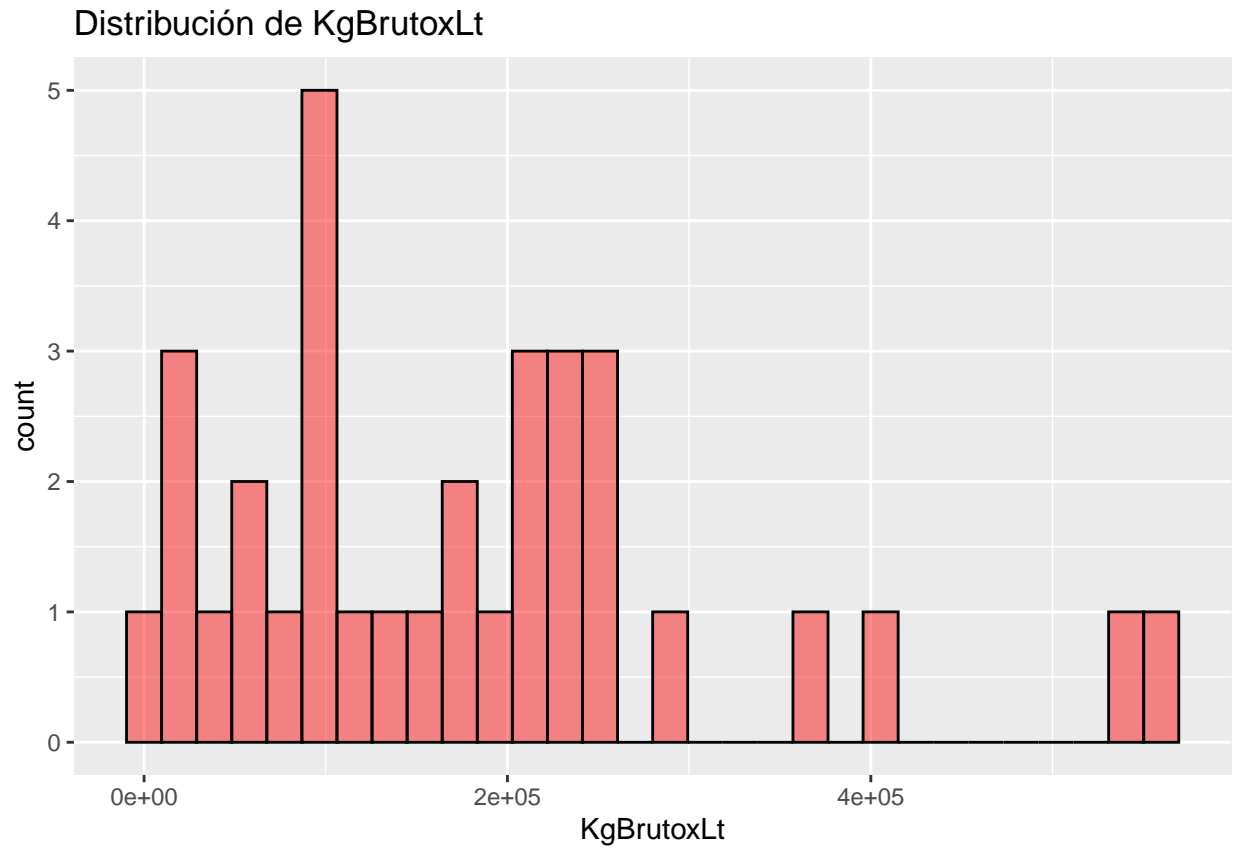


```
##  
## [[6]]
```

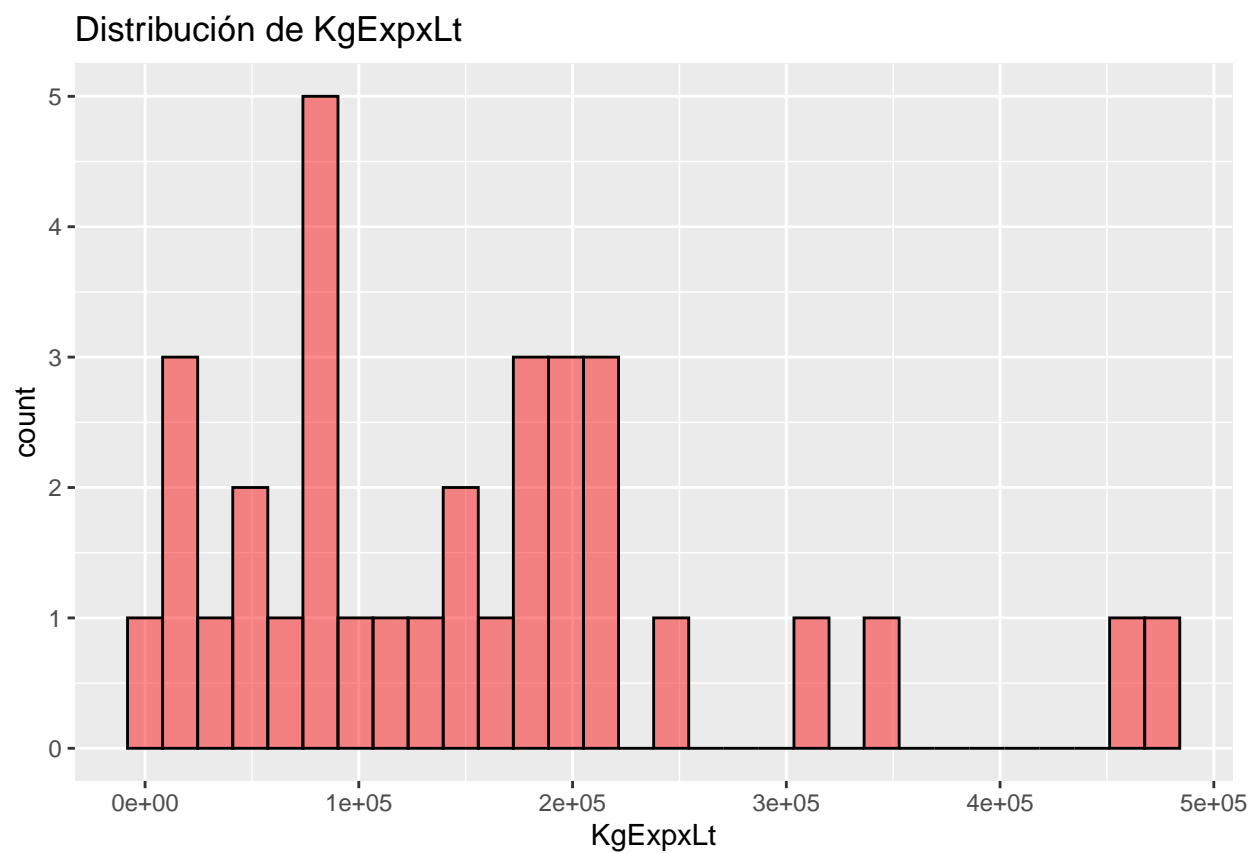




```
##  
## [[7]]
```

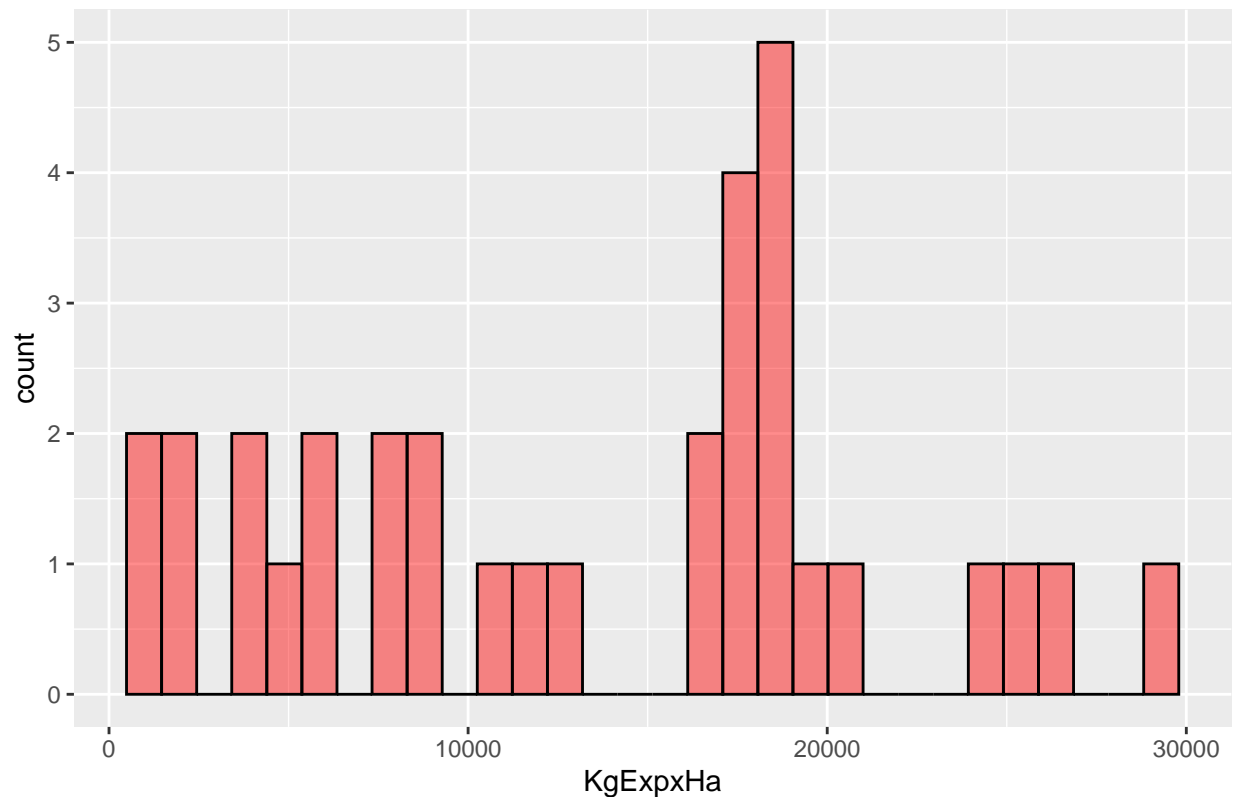


```
##  
## [[8]]
```



```
##  
## [[9]]
```

## Distribución de KgExpxHa



```
names(data)
```

```
## [1] "Lote"      "Empresa"   "Fundo"     "Variedad"  "Area"
## [6] "Color"     "FSiembra"  "PlantasxHa" "PlantasxLt" "FrutosxPl"
## [11] "PesoPromFr" "KgBrutoxHa" "KgBrutoxLt" "KgExpxLt"  "KgExpxHa"
## [16] "Estatus"
```

```
names(data_numeric)
```

```
## [1] "Area"      "PlantasxHa" "PlantasxLt" "FrutosxPl"  "PesoPromFr"
## [6] "KgBrutoxHa" "KgBrutoxLt" "KgExpxLt"  "KgExpxHa"
```

```
#####
```

```
# Configurar tamaño del gráfico
```

```
options(repr.plot.width=12, repr.plot.height=6)
```

```
# Crear el boxplot
```

```
boxplot <- ggplot(data, aes(x = Fundo, y = KgExpxHa, fill = Fundo)) +  
  geom_boxplot() +
```

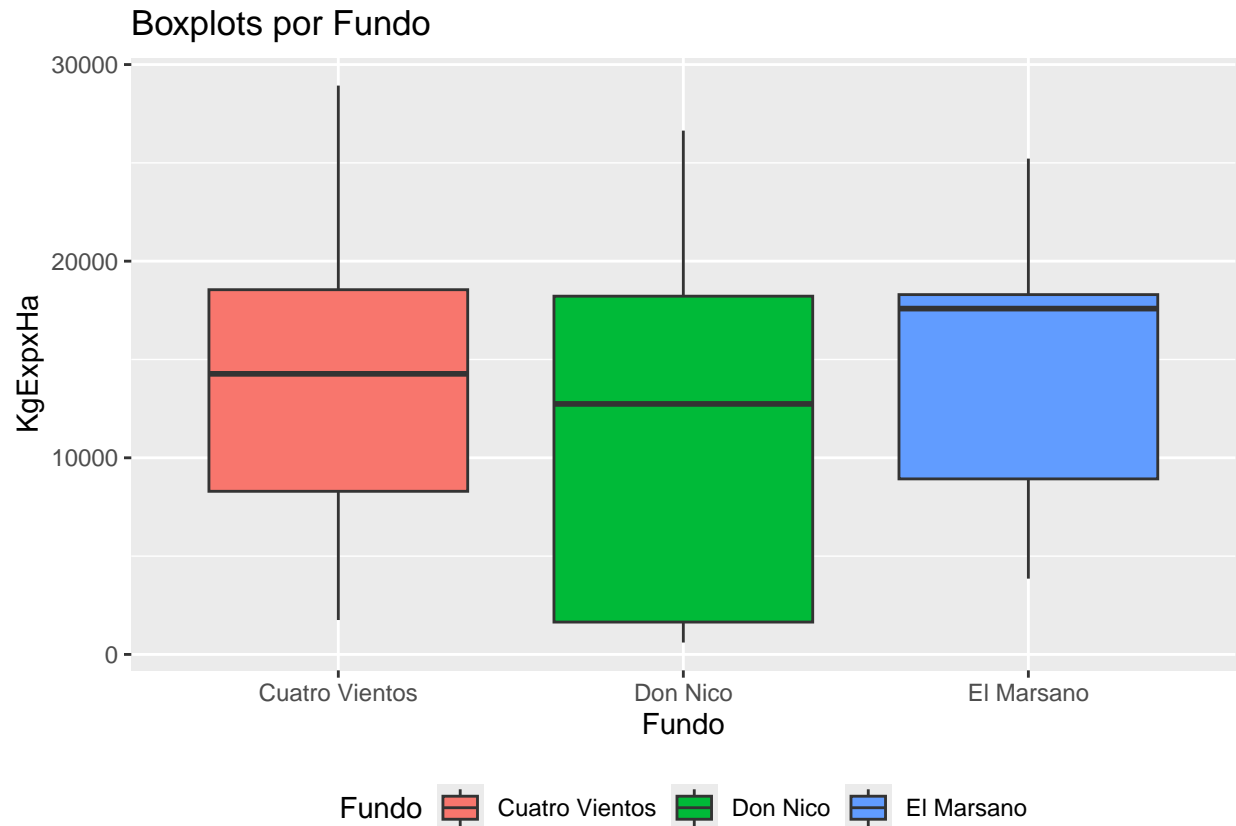
```
  labs(title = "Boxplots por Fundo")
```

```
# Establecer la ubicación de la leyenda
```

```
boxplot <- boxplot + theme(legend.position = "bottom")
```

```
# Mostrar el boxplot
```

```
print(boxplot)
```



```
#####
names(data)
```

```
## [1] "Lote"      "Empresa"   "Fundo"     "Variedad"  "Area"
## [6] "Color"     "FSiembra"  "PlantasxHa" "PlantasxLt" "FrutosxPl"
## [11] "PesoPromFr" "KgBrutoxHa" "KgBrutoxLt" "KgExpxLt"  "KgExpxHa"
## [16] "Estatus"
```

```
# Crear los boxplots
```

```
boxplot <- ggplot(data, aes(x = Area, y = KgExpxHa, fill = Fundo)) +
```

```
  geom_boxplot() +
```

```
  labs(title = "Boxplots por Fundo") +
```

```
  facet_wrap(~Variedad) # Cambia "Otra_Variable" por el nombre de la variable que deseas usar para dividir
```

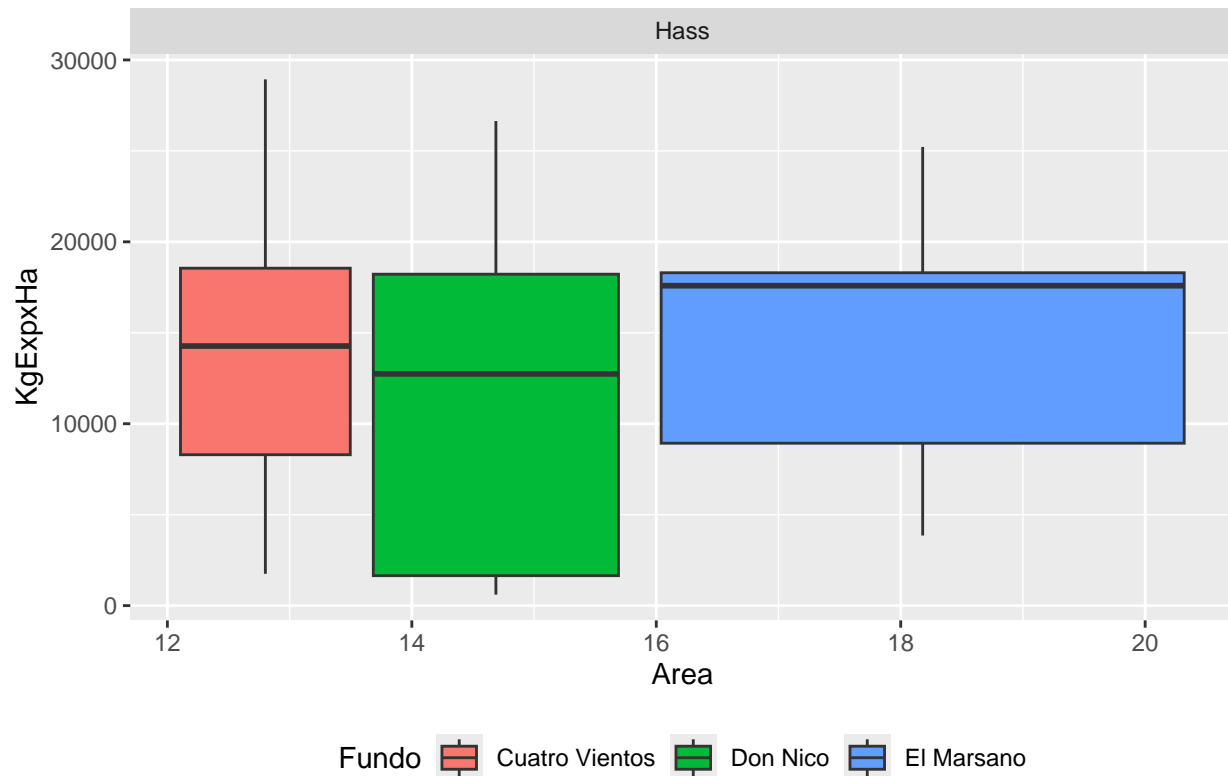
```
# Establecer la ubicación de la leyenda
```

```
boxplot <- boxplot + theme(legend.position = "bottom")
```

```
# Mostrar los boxplots
```

```
print(boxplot)
```

## Boxplots por Fundo



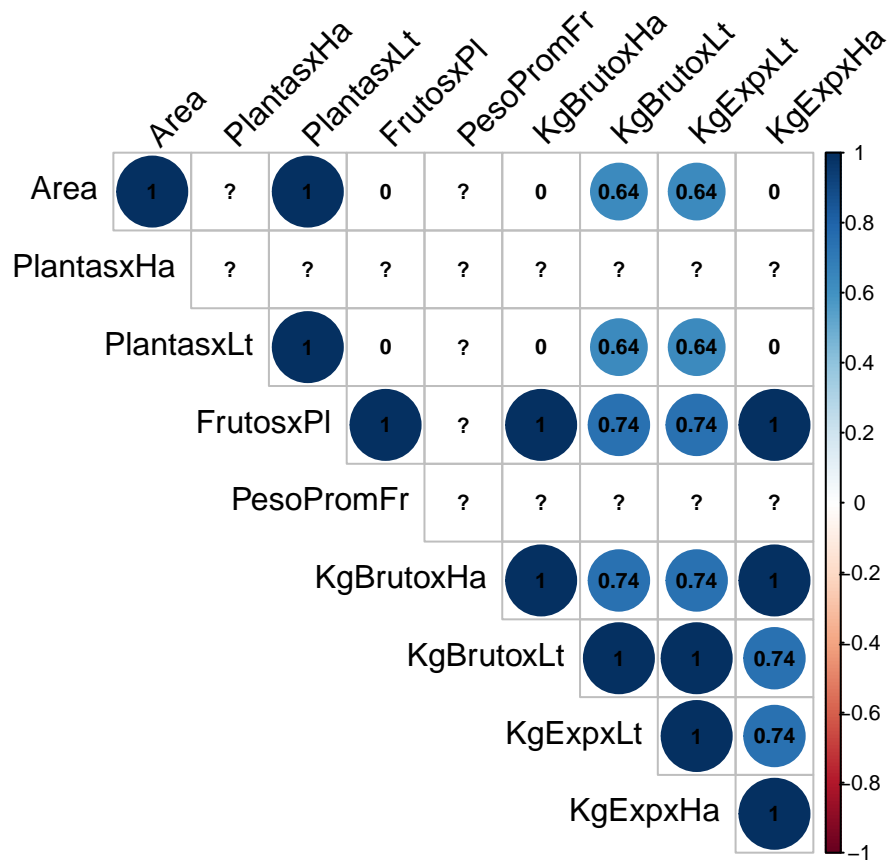
```
# Correlation matrix (if applicable)
# Install the 'corrplot' package if not already installed
if (!require("corrplot")) install.packages("corrplot")

## Loading required package: corrplot
## corrplot 0.92 loaded

library(corrplot)
correlation_matrix <- cor(data_numeric, use = "pairwise.complete.obs")

## Warning in cor(data_numeric, use = "pairwise.complete.obs"): the standard
## deviation is zero

corrplot(correlation_matrix, method = "circle", type = "upper", #order = "hclust",
         tl.col = "black", # text label color
         tl.srt = 45,      # text label rotation in degrees
         addCoef.col = "black", # color of the correlation coefficients
         number.cex = 0.7,  # size of the correlation coefficients
         cl.cex = 0.7,      # size of the color legend text
         cl.ratio = 0.1     # ratio of the color legend size
)
```



```
colnames(data_numeric)
```

```
## [1] "Area"      "PlantasxHa" "PlantasxLt" "FrutosxPl"  "PesoPromFr"
## [6] "KgBrutoxHa" "KgBrutoxLt" "KgExpxLt"   "KgExpxHa"
```

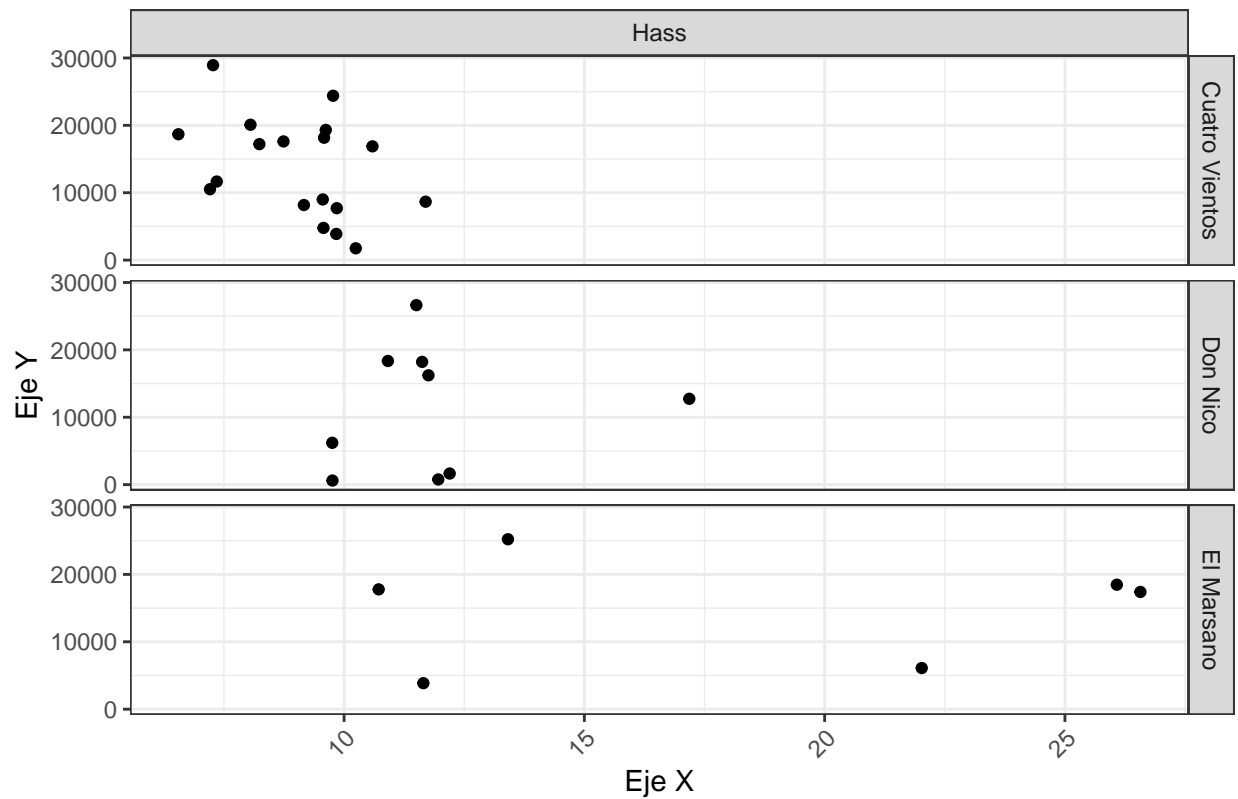
```
colnames(data)
```

```
## [1] "Lote"      "Empresa"    "Fundo"      "Variedad"   "Area"
## [6] "Color"     "FSiembra"   "PlantasxHa" "PlantasxLt" "FrutosxPl"
## [11] "PesoPromFr" "KgBrutoxHa" "KgBrutoxLt" "KgExpxLt"   "KgExpxHa"
## [16] "Estatus"
```

```
###
```

```
ggplot(data, aes(x = Area, y = KgExpxHa)) +
  geom_point() +
  facet_grid(Fundo ~ Variedad) + # Creates a grid of plots by Variedad and Color
  labs(title = "Exploration of Kg Exportado Lote by Variedad and Color",
        x = "Eje X",
        y = "Eje Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

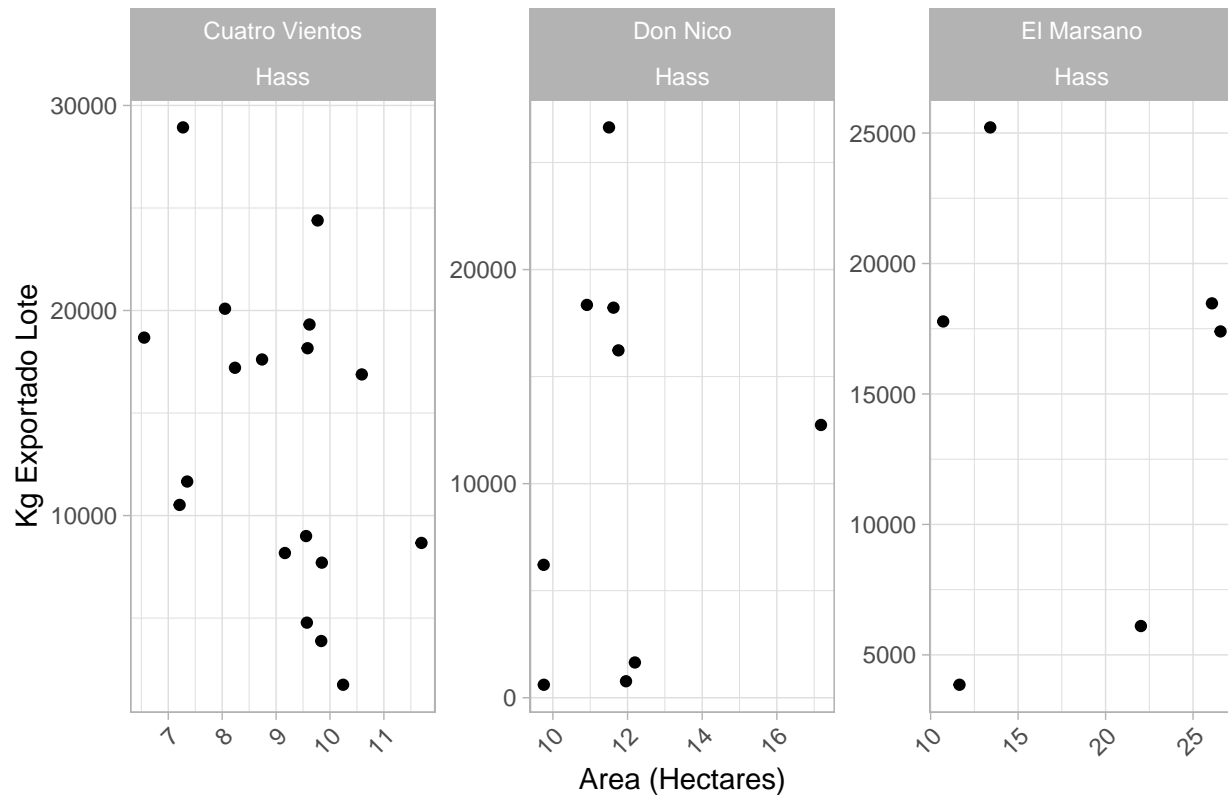
## Exploration of Kg Exportado Lote by Variedad and Color



```
# Using facet_wrap to explore the same data but with a single wrapping variable
ggplot(data, aes(x = Area, y = KgExpHa)) +
  geom_point() +
  facet_wrap(~ Fundo + Variedad, scales = "free") +
  labs(title = "Exploration of Kg Exportado Lote by Variedad and Color",
        x = "Area (Hectares)",
        y = "Kg Exportado Lote") +
  theme_light() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

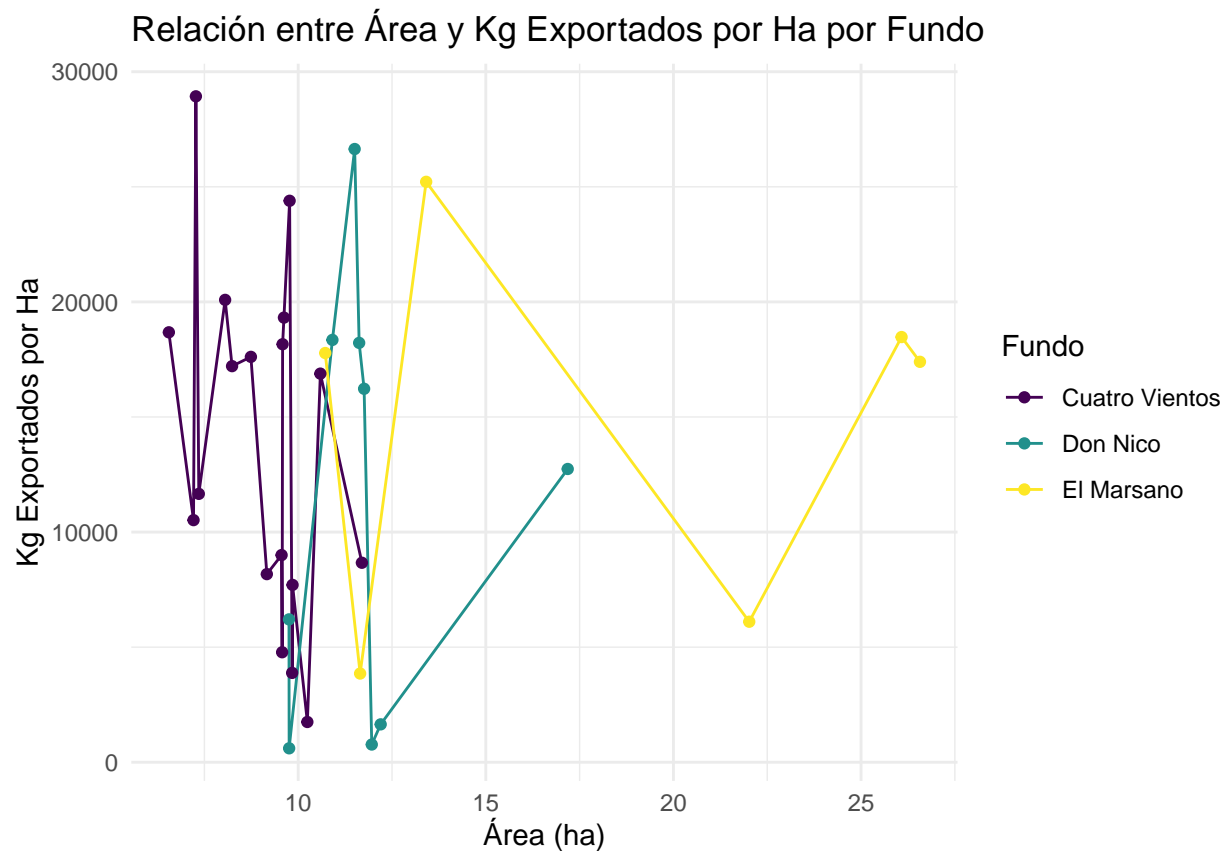


## Exploration of Kg Exportado Lote by Variedad and Color



```
# Ahora exportar después de cambiar el directorio
write.csv(data, "my_data.csv", row.names = FALSE)

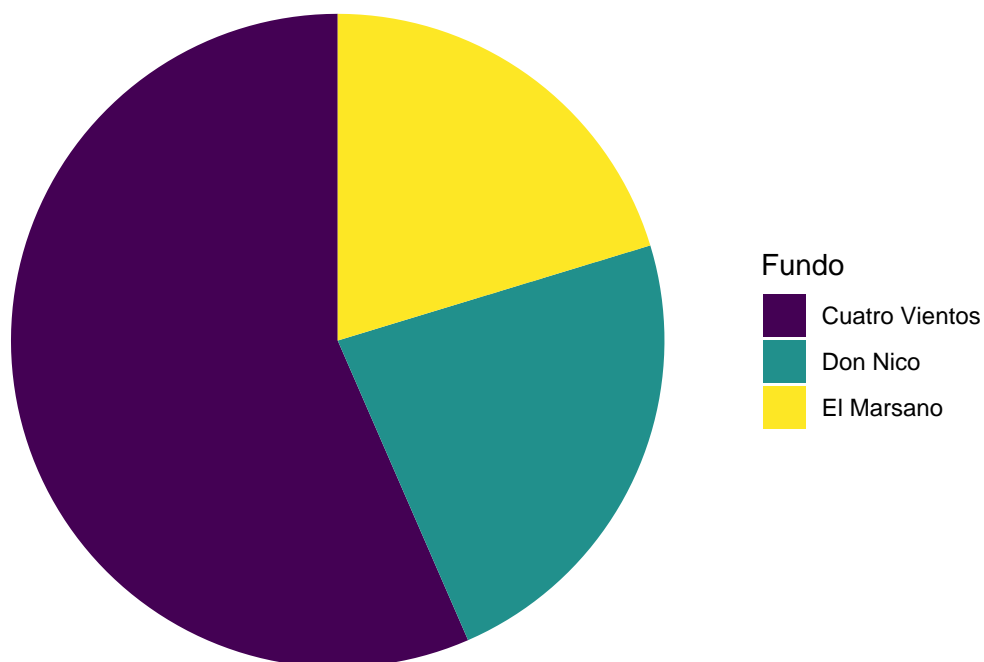
# Líneas
data$Fundo <- as.factor(data$Fundo)
data$Area <- as.numeric(data$Area)
data$KgExpHa <- as.numeric(data$KgExpHa)
# Crear el gráfico de líneas
ggplot(data, aes(x=Area, y=KgExpHa, color=Fundo, group=Fundo)) +
  geom_line() +
  geom_point() + # Agregar puntos para mejor visualización de los datos
  labs(title="Relación entre Área y Kg Exportados por Ha por Fundo",
        x="Área (ha)", y="Kg Exportados por Ha") +
  scale_color_viridis_d() + # Usar una paleta de colores para diferenciar los Fundos
  theme_minimal() # Tema minimalista
```



```
# Sectores
# Preparar los datos sumando o promediando KgExpHa por Fundo
data_summary <- data %>%
  group_by(Fundo) %>%
  summarize(Total_KgExpHa = sum(KgExpHa, na.rm = TRUE)) # Utilizar sum() o mean() según el caso

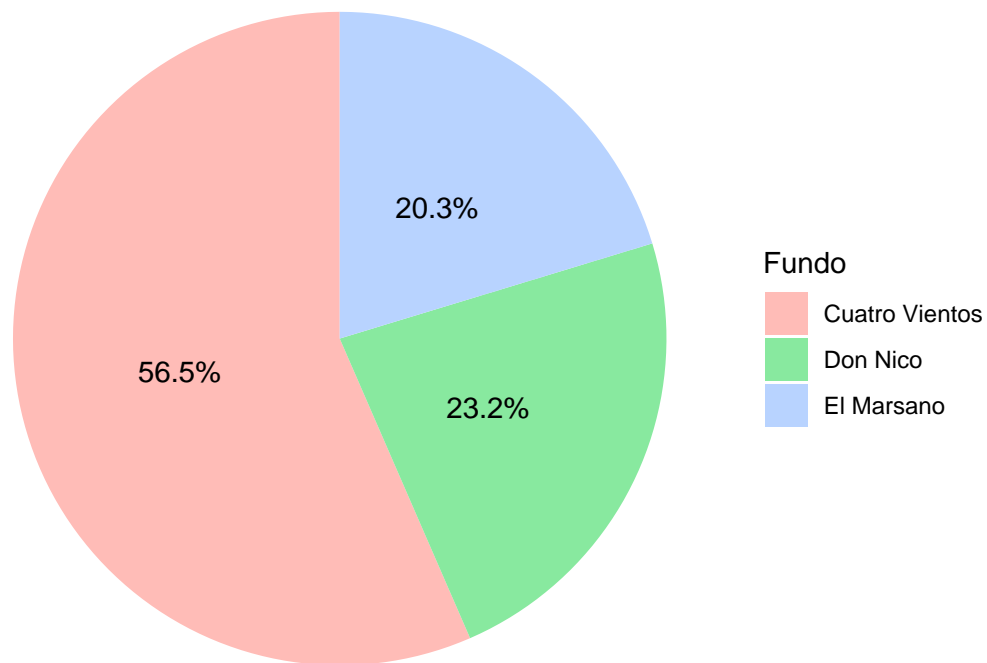
# Crear el gráfico de sectores
ggplot(data_summary, aes(x="", y=Total_KgExpHa, fill=Fundo)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar(theta = "y") + # Convertir el gráfico de barras a un gráfico circular
  theme_void() + # Eliminar elementos del gráfico no necesarios
  labs(title="Distribución de Kg Exportados por Ha por Fundo") +
  scale_fill_viridis_d() # Usar una paleta de colores
```

## Distribución de Kg Exportados por Ha por Fundo



```
#####  
library(scales) # Para el formateo de los porcentajes  
# Preparar los datos sumando KgExpHa por Fundo  
data_summary <- data %>%  
  group_by(Fundo) %>%  
  summarize(Total_KgExpHa = sum(KgExpHa, na.rm = TRUE)) %>%  
  mutate(Percentage = Total_KgExpHa / sum(Total_KgExpHa) * 100) # Calcular el porcentaje  
# Función para generar una paleta de colores pastel  
pastel <- function(colours = 5) {  
  hues = seq(15, 375, length = colours + 1)  
  hcl(h = hues, l = 85, c = 65)[1:colours]  
}  
  
# Crear el gráfico de sectores con porcentajes  
ggplot(data_summary, aes(x="", y=Total_KgExpHa, fill=Fundo)) +  
  geom_bar(width = 1, stat = "identity") +  
  coord_polar(theta = "y") +  
  theme_void() +  
  labs(title="Distribución de Kg Exportados por Ha por Fundo") +  
  scale_fill_manual(values = pastel(colours = length(unique(data$Fundo)))) + # Colores pastel  
  geom_text(aes(label = paste0(round(Percentage, 1), "%")), position = position_stack(vjust = 0.5)) #
```

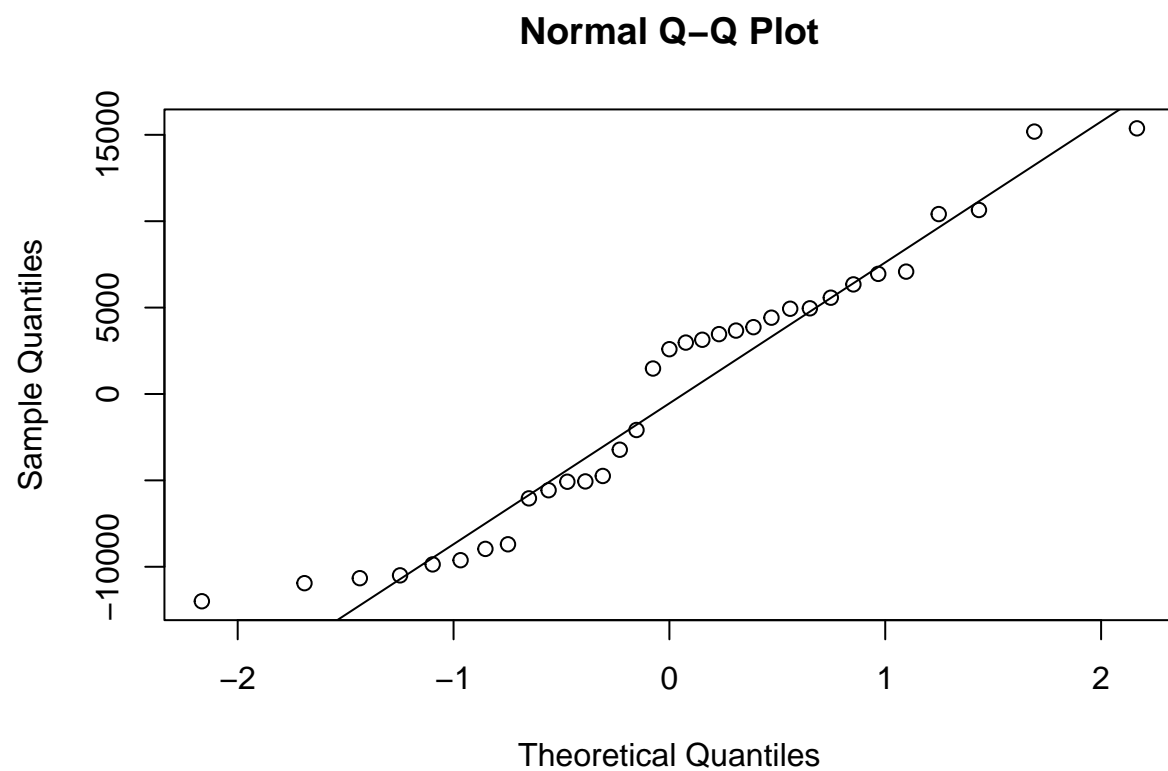
## Distribución de Kg Exportados por Ha por Fundo



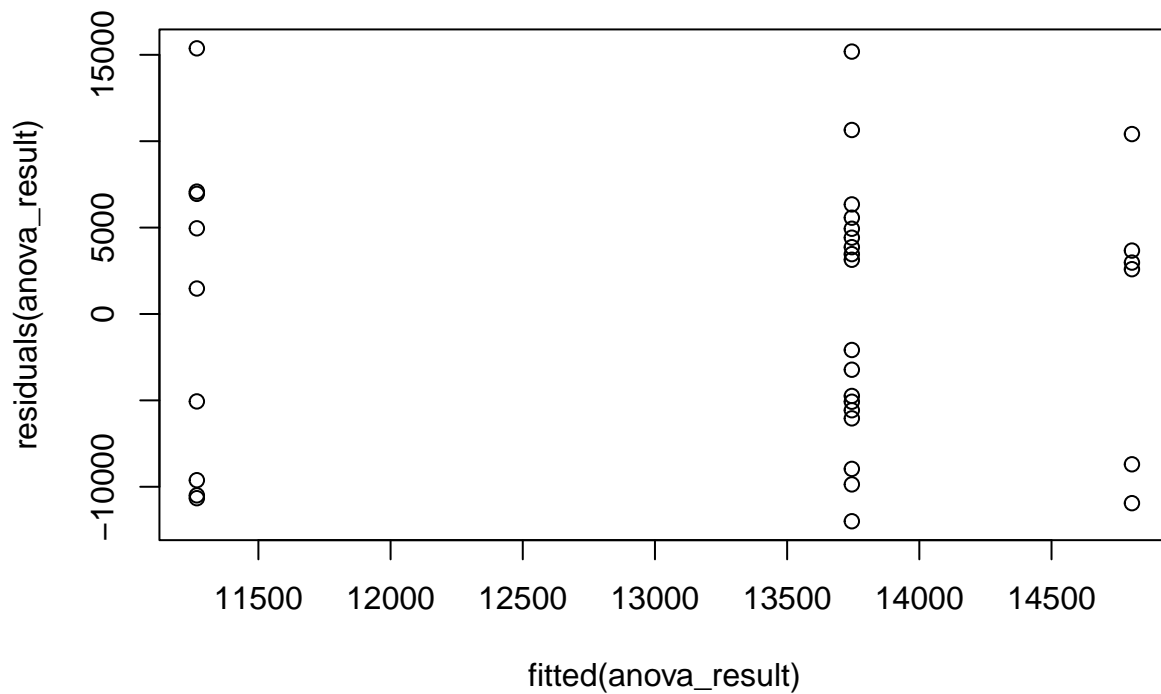
```
### ANOVA ###  
# Ensure that 'Fundo' is a factor and 'KgExpHa' is numeric  
data$Fundo <- as.factor(data$Fundo)  
data$KgExpHa <- as.numeric(data$KgExpHa)  
# ANOVA to compare 'KgExpHa' across different 'Fundo'  
anova_result <- aov(KgExpHa ~ Fundo, data = data)  
# Check the summary of the ANOVA  
summary(anova_result)
```

```
##           Df    Sum Sq Mean Sq F value Pr(>F)  
## Fundo      2 5.428e+07 27140215    0.41  0.667  
## Residuals 30 1.985e+09 66156745
```

```
# Check for assumptions: Normality  
qqnorm(residuals(anova_result))  
qqline(residuals(anova_result))
```



```
# Homogeneity of variances  
plot(residuals(anova_result) ~ fitted(anova_result))
```



```
# If ANOVA is significant, conduct post-hoc tests
```

```
TukeyHSD(anova_result)
```

```
## Tukey multiple comparisons of means
```

```
## 95% family-wise confidence level
```

```
##
```

```
## Fit: aov(formula = KgExpHa ~ Fundo, data = data)
```

```
##
```

```
## $Fundo
```

```
## diff lwr upr p adj
```

```
## Don Nico-Cuatro Vientos -2477.571 -10663.654 5708.512 0.7382554
```

```
## El Marsano-Cuatro Vientos 1059.531 -8392.943 10512.006 0.9588378
```

```
## El Marsano-Don Nico 3537.102 -7031.086 14105.290 0.6906061
```

```
## ANOVA #2
```

```
# Convert 'Fundo' to a factor if necessary
```

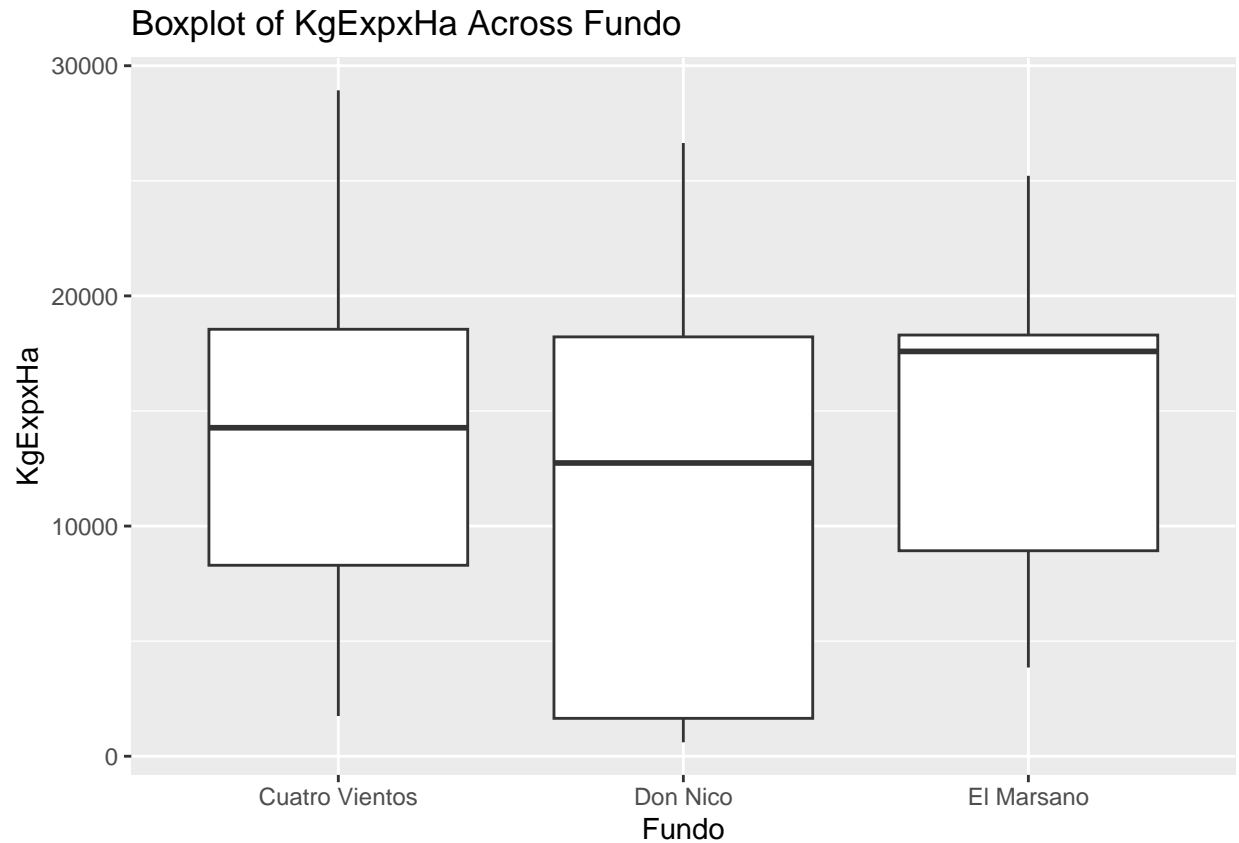
```
data$Fundo <- as.factor(data$Fundo)
```

```
# Boxplot for 'KgExpHa' across different 'Fundo'
```

```
ggplot(data, aes(x=Fundo, y=KgExpHa)) +
```

```
geom_boxplot() +
```

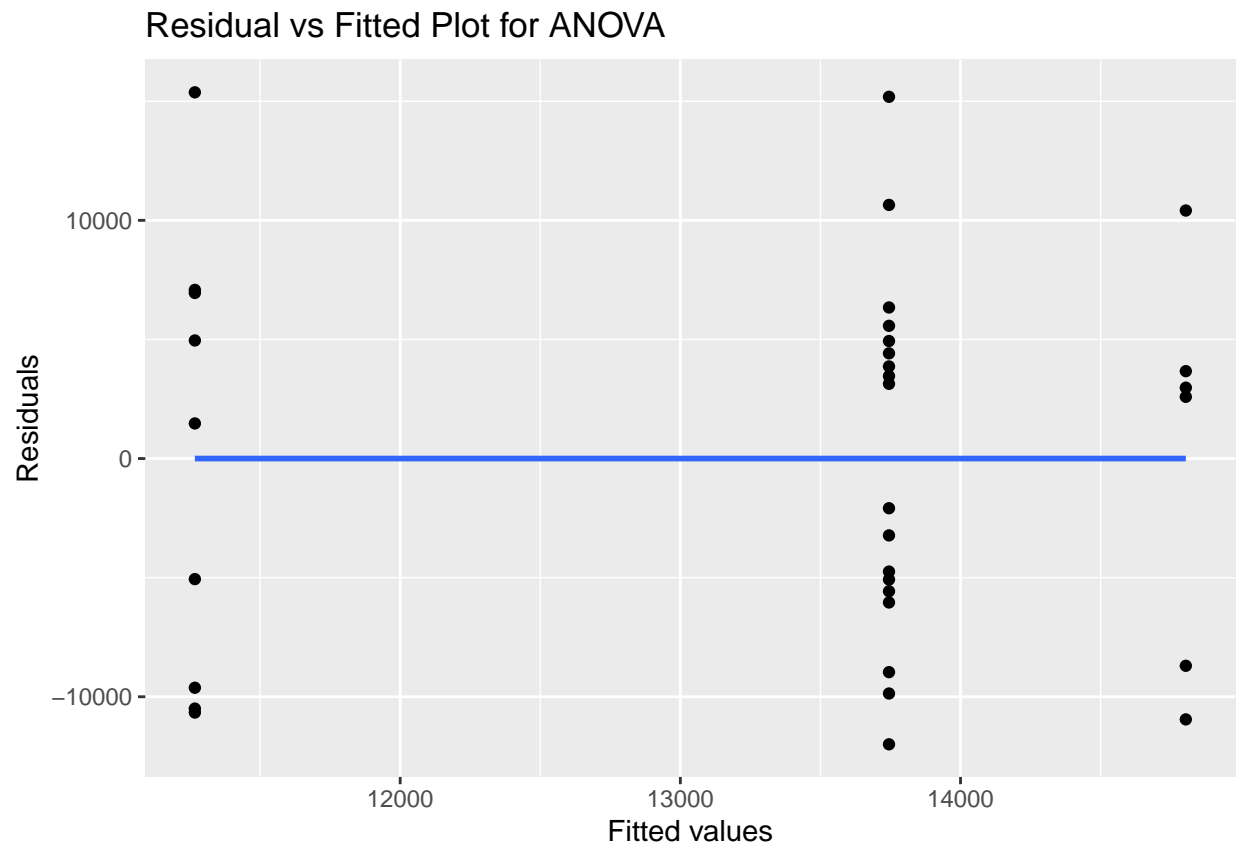
```
labs(title="Boxplot of KgExpHa Across Fundo", x="Fundo", y="KgExpHa")
```



```
# Performing the ANOVA
anova_result <- aov(KgExpHa ~ Fundo, data = data)

# Residual plot
res_data <- data.frame(residuals= residuals(anova_result), fitted=fitted(anova_result))
ggplot(res_data, aes(x=fitted, y=residuals)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE) +
  labs(title="Residual vs Fitted Plot for ANOVA", x="Fitted values", y="Residuals")

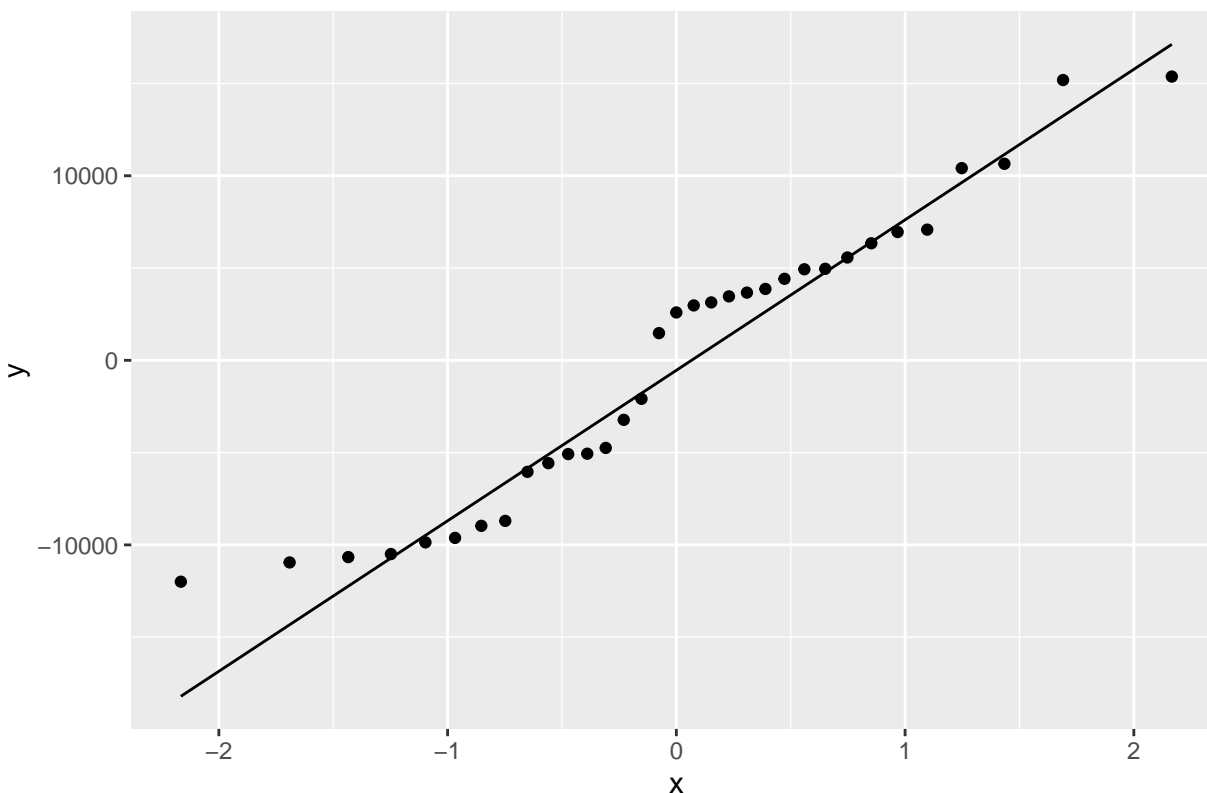
## `geom_smooth()` using formula = 'y ~ x'
```



```
# QQ plot of residuals
ggplot(res_data, aes(sample=residuals)) +
  geom_qq() +
  geom_qq_line() +
  labs(title="QQ Plot of Residuals")
```



QQ Plot of Residuals



```
#####
```

```
# Create a pairwise scatter plot
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2
```

```
# Increase the cardinality threshold
ggpairs(data, cardinality_threshold = 35)
```

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

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

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

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

```
## Warning in cor(x, y): the standard deviation is zero
```

```
## Warning in cor(x, y): the standard deviation is zero
```

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

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

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

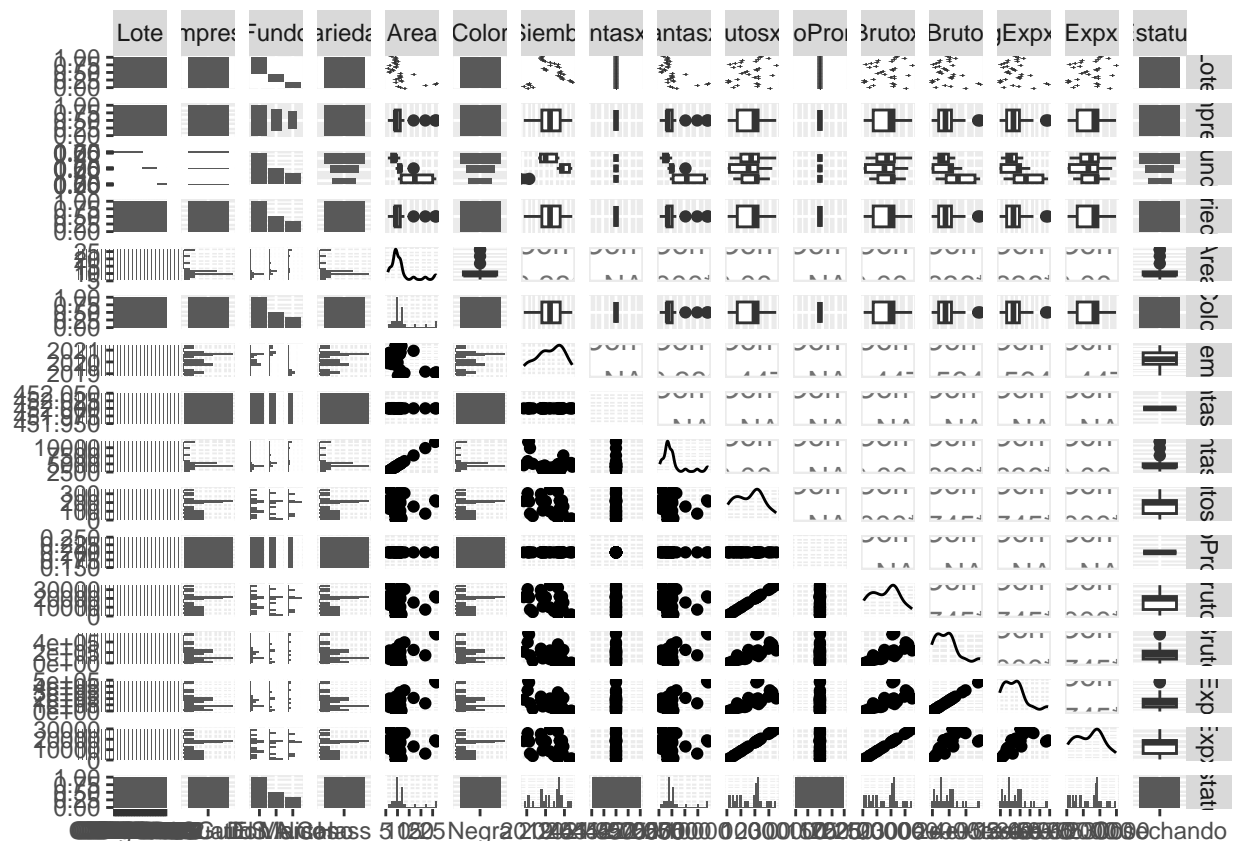
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

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

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



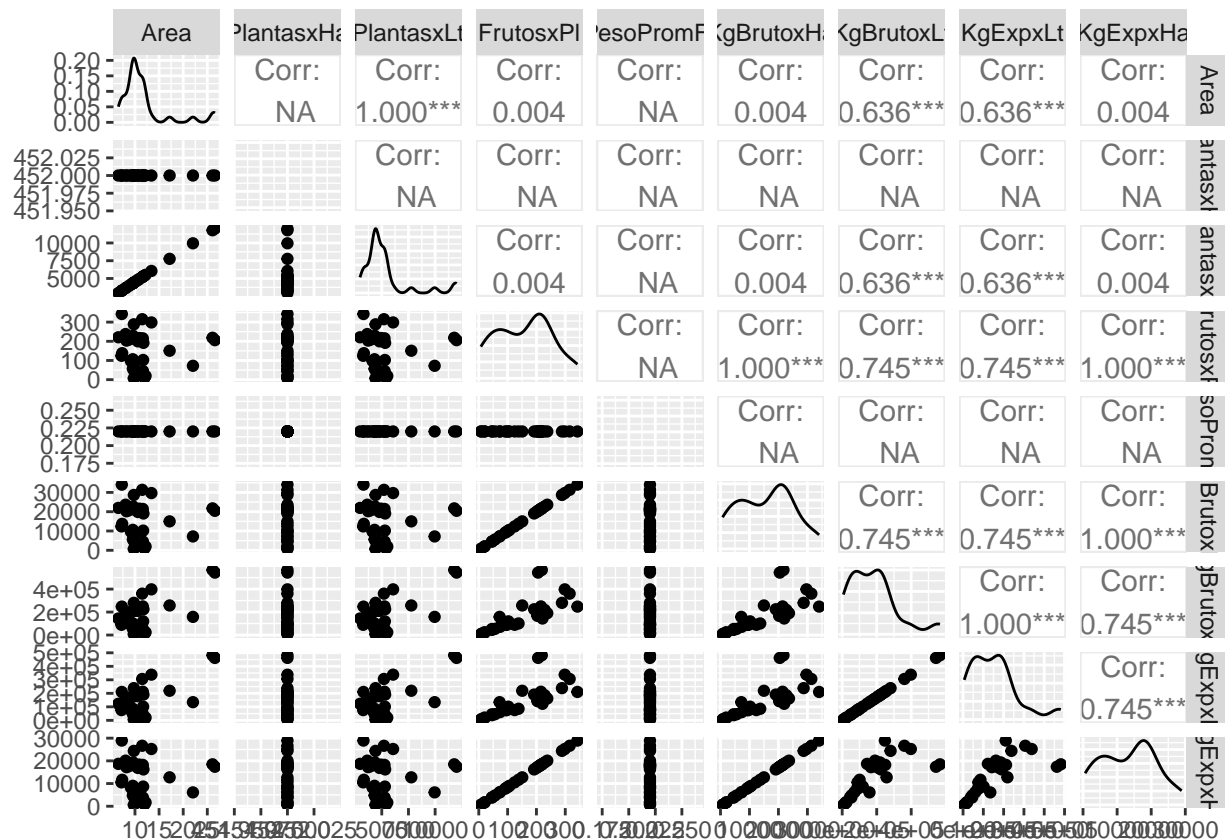
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using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



```
ggpairs(data_numeric, cardinality_threshold = 35)
```

```
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
```

```
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
```



```
p <- ggpairs(data, cardinality_threshold = 35)
# Save the plot with ggsave
ggsave("ggpairs_plot.png", plot = p, width = 30, height = 30, dpi = 600)
```

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

```
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
```

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

```
## Warning in cor(x, y): the standard deviation is zero
```



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

#### Notas:

1. **Packages:** Ensure you have the necessary R packages (`readxl`, `ggplot2`, `dplyr`) installed. If not, you can install them using `install.packages()`.

This markdown script provides a structure for conducting an EDA with visualizations, and you can further expand upon this as needed.

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.