

PEC 1. Las ómicas

Judit García Fernández

Contenido

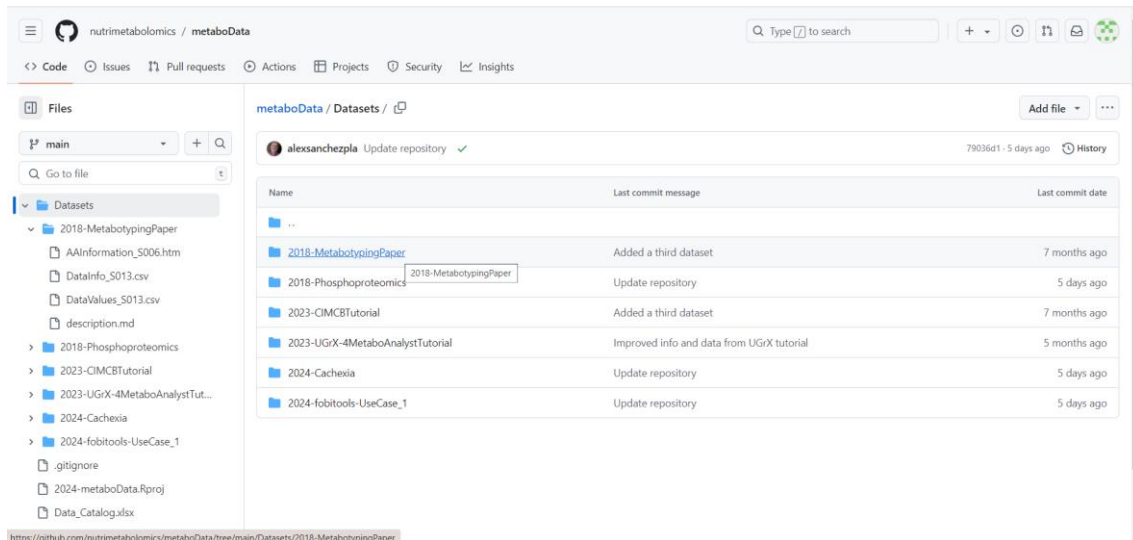
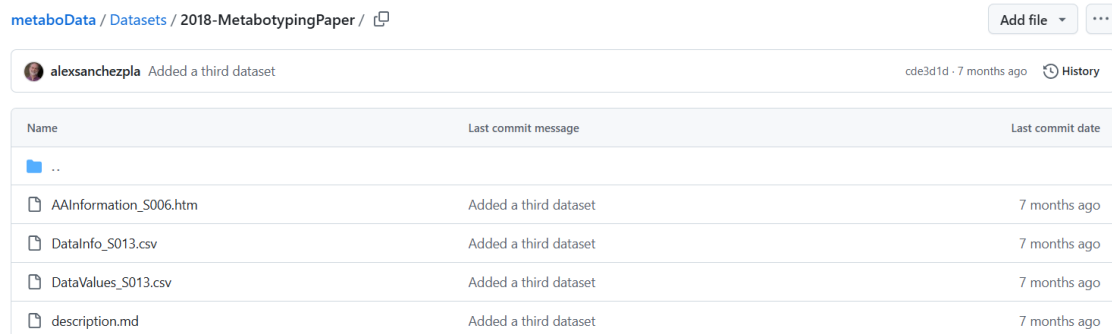
Contenido	1
1.Descarga de datos	1
2.Creación del contenedor.....	2
2.1.Cargamos las bibliotecas	2
2.2.Cargamos los datos y metadatos	3
2.3.Preparamos los datos.....	3
2.4.Creamos y guardamos el contenedor SummarizedExperiment.....	3
3.Exploración de los datos	4
3.1.Descripción general de los datos	4
3.2.Análisis descriptivo	5
3.3.Resumen estadístico	6
4.Reposición de los datos en Github	11

1. Descarga de datos

El inicio de este informe consiste en la búsqueda y selección de un dataset de metabolómica. En este caso, se ha buscado en el repositorio de github <https://github.com/nutrimetabolomics/metaboData/>, en el cual, en la carpeta “Datasets” encontraremos diversos de ellos.

Se selecciona el dataset “2018-MetabotypingPaper”, datos utilizados en el artículo “Metabotipos de respuesta a la cirugía bariátrica independientes de la magnitud de la pérdida de peso” tal y como podemos leer en el archivo description.md de este mismo dataset.

Encontramos dos archivos, DataInfo_S013.csv que contiene los metadatos del dataset (información acerca de los datos, tipos de variables, etc.) y DataValues_S013.csv, el cual contiene los datos propiamente dichos.

2. Creación del contenedor

Para la creación del contenedor de tipo SummarizedExperiment seguiremos los siguientes pasos:

2.1. Cargamos las bibliotecas

Empezamos comprobando si tenemos la biblioteca BiocManager instalada para poder gestionar paquetes de Bioconductor. Si no lo está, se instala automáticamente.

Después, instalamos y cargamos el paquete SummarizedExperiment, utilizado para organizar los datos con los que queremos trabajar de una manera eficiente. El paquete readr es utilizado para leer los archivos, en este caso archivos csv.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
```

```
BiocManager::install("SummarizedExperiment")
```

```
library(SummarizedExperiment)
```

```
library(readr)
```

2.2. Cargamos los datos y metadatos

Para cargar los archivos DataInfo_S013.csv y DataValues_S013.csv utilizamos el comando `read_csv`, renombrándolos como `data_info` y `data_values`.

```
#Cargamos Los datos
data_values <- read_csv("C:/Users/judit/OneDrive/Escritorio/MÁSTER/3er SEMESTRE/ADO - Análisis de datos ómicos/Reto 1 - Las ómicas/PAC 1/DataValues_S013.csv")

#Cargamos Los metadatos
data_info <- read_csv("C:/Users/judit/OneDrive/Escritorio/MÁSTER/3er SEMESTRE/ADO - Análisis de datos ómicos/Reto 1 - Las ómicas/PAC 1/DataInfo_S013.csv")
```

2.3. Preparamos los datos

Convertimos los datos de expresión en una matriz asignando los nombres de las filas de la matriz a partir de la primera columna de `data_values`:

```
rownames(data_values) <- data_values[[1]]

data_values <- as.matrix(data_values[, -1])
```

Convertimos los metadatos en un dataframe de Bioconductor, también utilizando su primera columna como identificador de filas, eliminándola posteriormente para que solo nos quede en el dataframe los datos descriptivos:

```
rownames(data_info) <- data_info[[1]]

data_info <- data_info[, -1]

data_info <- as(data_info, "DataFrame")
```

2.4. Creamos y guardamos el contenedor SummarizedExperiment

El siguiente paso es crear el contenedor del tipo `SummarizedExperiment` con los datos `data_info` y `data_values`.

```
se <- SummarizedExperiment(assays = list(counts=data_values),
colData = data_info)
```

Para comprobar los datos dentro del objeto `SummarizedExperiment`, mostramos las 6 primeras filas de los metadatos (6 primeras muestras):

```
se

## class: SummarizedExperiment
## dim: 39 695
## metadata(0):
## assays(1): counts
```

```
## rownames: NULL
## rowData names(0):
## colnames(695): SUBJECTS SURGERY ... SM.C24.0_T5 SM.C24.1_T5
## colData names(3): VarName varTpe Description
```

```
head(colData(se))
```

```
## DataFrame with 6 rows and 3 columns
##           VarName      varTpe Description
##           <character> <character> <character>
## SUBJECTS    SUBJECTS      integer  dataDesc
## SURGERY      SURGERY      character dataDesc
## AGE          AGE          integer  dataDesc
## GENDER       GENDER       character dataDesc
## Group        Group        integer  dataDesc
## MEDDM_T0     MEDDM_T0     integer  dataDesc
```

O las primeras filas de los datos de expresión (las primeras muestras):

```
head(assay(se, "counts"), 1)
```

Finalmente, guardamos el contenedor SummarizedExperiment en un archivo .Rda, para poder ser cargado más tarde en R para realizar un análisis de los datos.

```
save(se, file="S013_SummarizedExperiment.Rda")
```

3. Exploración de los datos

3.1. Descripción general de los datos

Esta sección proporciona una visión general del dataset, incluyendo su estructura y las principales variables.

```
# Dimensiones del dataset
dim(se)

## [1] 39 695

# Nombres de variables
colnames(colData(se))

## [1] "VarName"      "varTpe"      "Description"

# Crear un subset con variables de interés
selected_vars <- c("AGE", "GENDER", "PESO_T0", "bmi_T0", "GLU_T0",
  "HOMA_T0", "TG_T0", "COL_T0")
data_subset <- assay(se, "counts")[, selected_vars]
head(data_subset)
```

```
##      AGE  GENDER PESO_T0 bmi_T0 GLU_T0 HOMA_T0 TG_T0 COL_T0
## [1,] "27"  "F"    "151"  "62.9" " 85"  " 2.40" "147" "256"
## [2,] "19"  "F"    "139"  "47.0" " 78"  " 2.32" "150" "180"
## [3,] "42"  "F"    " 84"  "29.8" " 75"  " 1.56" " 45" "211"
## [4,] "37"  "F"    "136"  "53.1" " 71"  " 2.25" "109" "205"
## [5,] "42"  "F"    "121"  "46.6" " 82"  " 1.22" " 30" "102"
## [6,] "24"  "F"    "148"  "48.8" " 71"  " 1.73" " 61" "121"
```

3.2. Análisis descriptivo

En esta sección, se realiza un análisis descriptivo de las variables seleccionadas.

```
# Seleccionar variables de interés y convertirlas a formato numérico (excepto GENDER)
selected_vars <- c("AGE", "PESO_T0", "bmi_T0", "GLU_T0", "HOMA_T0", "TG_T0", "COL_T0")
```

```
# Convertir `assay(se, "counts")` a data.frame y seleccionar columnas
data_subset <- as.data.frame(assay(se, "counts")[, selected_vars])
```

```
# Convertir cada columna de interés a numérica
data_subset <- as.data.frame(lapply(data_subset, function(x) as.numeric(as.character(x)))))
```

```
# Agregar GENDER por separado como una columna de tipo character
data_subset$GENDER <- as.character(assay(se, "counts")[, "GENDER"])
```

```
# Verificar la estructura del nuevo data frame
str(data_subset)
```

```
## 'data.frame':    39 obs. of  8 variables:
## $ AGE      : num  27 19 42 37 42 24 33 55 40 47 ...
## $ PESO_T0 : num  151 139 84 136 121 148 109 109 114 120 ...
## $ bmi_T0  : num  62.9 47 29.8 53.1 46.6 48.8 43.7 41.8 44 40.6 ...
## $ GLU_T0  : num  85 78 75 71 82 71 80 90 92 84 ...
## $ HOMA_T0 : num  2.4 2.32 1.56 2.25 1.22 1.73 1.82 0.76 1.23 1.45 ...
## $ TG_T0   : num  147 150 45 109 30 61 75 53 164 145 ...
## $ COL_T0  : num  256 180 211 205 102 121 192 181 154 220 ...
## $ GENDER  : chr   "F" "F" "F" "F" ...
```

3.3. Resumen estadístico

```
summary(data_subset)
```

```
##          AGE          PESO_T0          bmi_T0          GLU_T0
## Min.      :19.00    Min.      : 84.0    Min.      :29.80    Min.      : 71.
## 0
## 1st Qu.:35.00    1st Qu.:119.5    1st Qu.:44.40    1st Qu.: 91.
## 0
## Median :41.00    Median :135.0    Median :48.80    Median :103.
## 0
## Mean    :40.79    Mean     :140.0    Mean     :50.52    Mean     :106.
## 5
## 3rd Qu.:46.00    3rd Qu.:155.0    3rd Qu.:55.35    3rd Qu.:109.
## 5
## Max.     :59.00    Max.      :200.0    Max.      :68.60    Max.      :263.
## 0
##      HOMA_T0      TG_T0      COL_T0      GENDER
## Min.      : 0.760    Min.      : 30.0    Min.      :102.0    Length:39
## 1st Qu.: 2.435    1st Qu.: 84.0    1st Qu.:180.5    Class :char
## Median : 4.210    Median :118.0    Median :207.0    Mode  :char
## Mean     : 4.890    Mean     :129.1    Mean     :206.6
## 3rd Qu.: 5.720    3rd Qu.:151.0    3rd Qu.:236.0
## Max.     :13.600    Max.      :268.0    Max.      :295.0
```

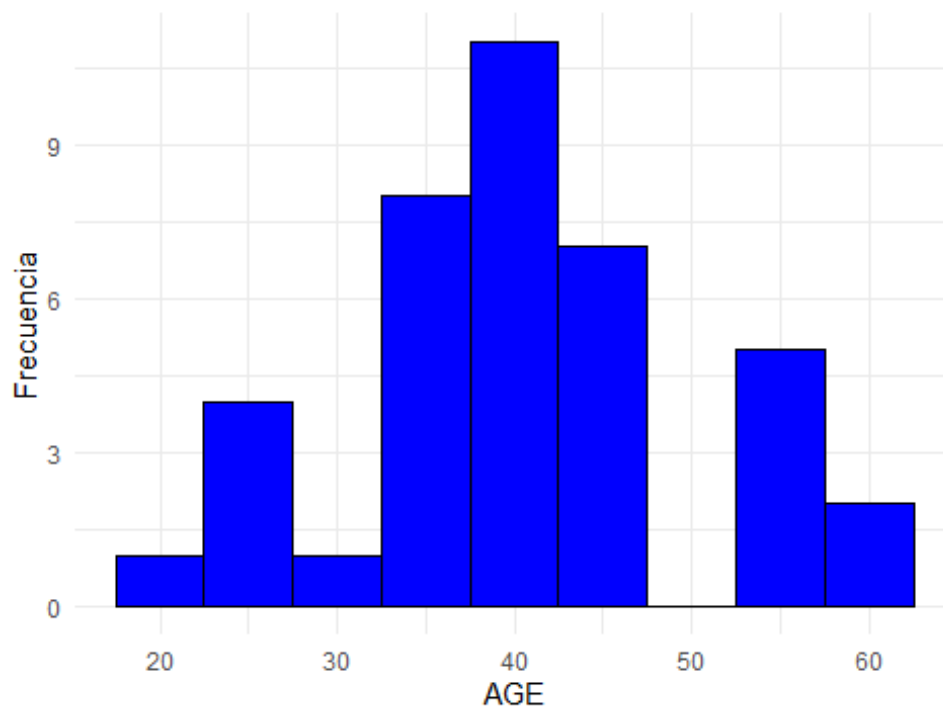
```
#Histogramas
```

```
library(ggplot2)
```

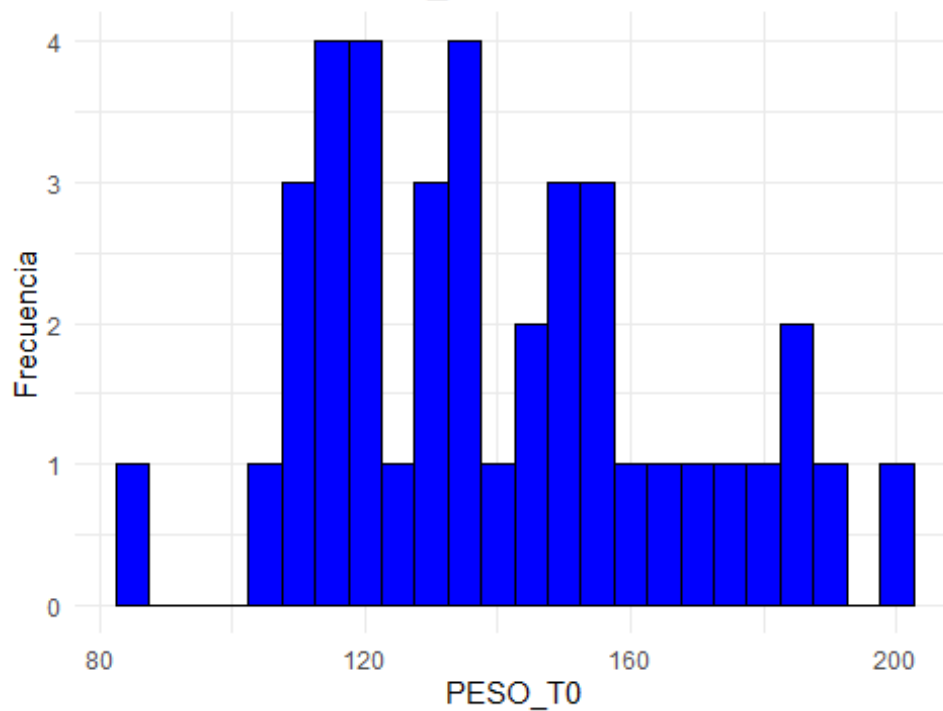
```
for (var in selected_vars) {
  plot <- ggplot(data_subset, aes(x = !!sym(var))) +
    geom_histogram(binwidth = 5, fill = "blue", color = "black")
  +
    labs(title = paste("Distribución de", var), x = var, y = "Fr
ecuencia") +
    theme_minimal()

  print(plot)
}
```

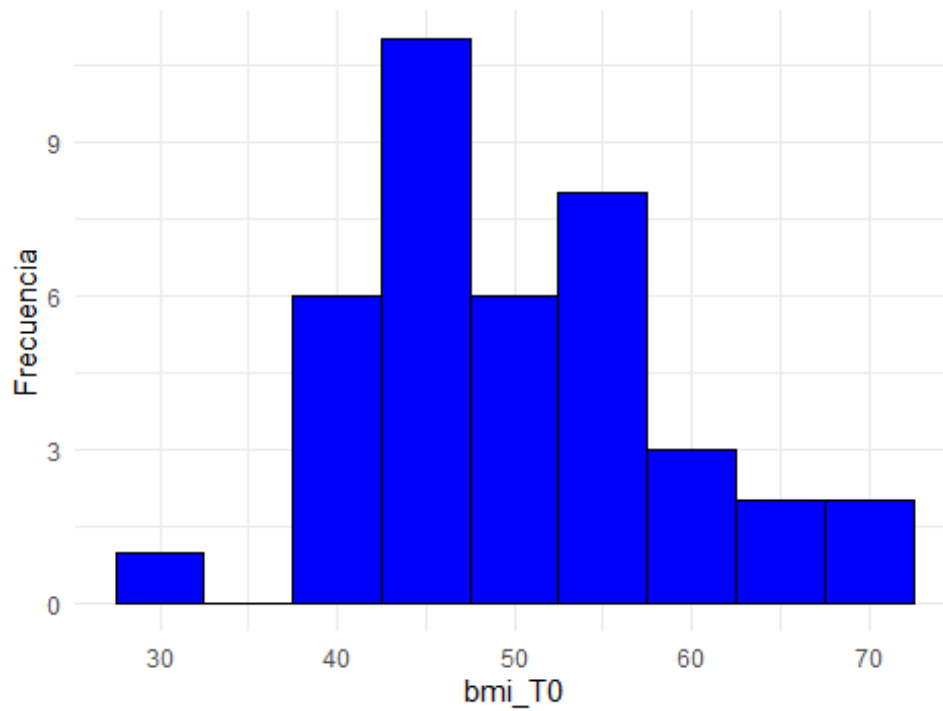
Distribución de AGE



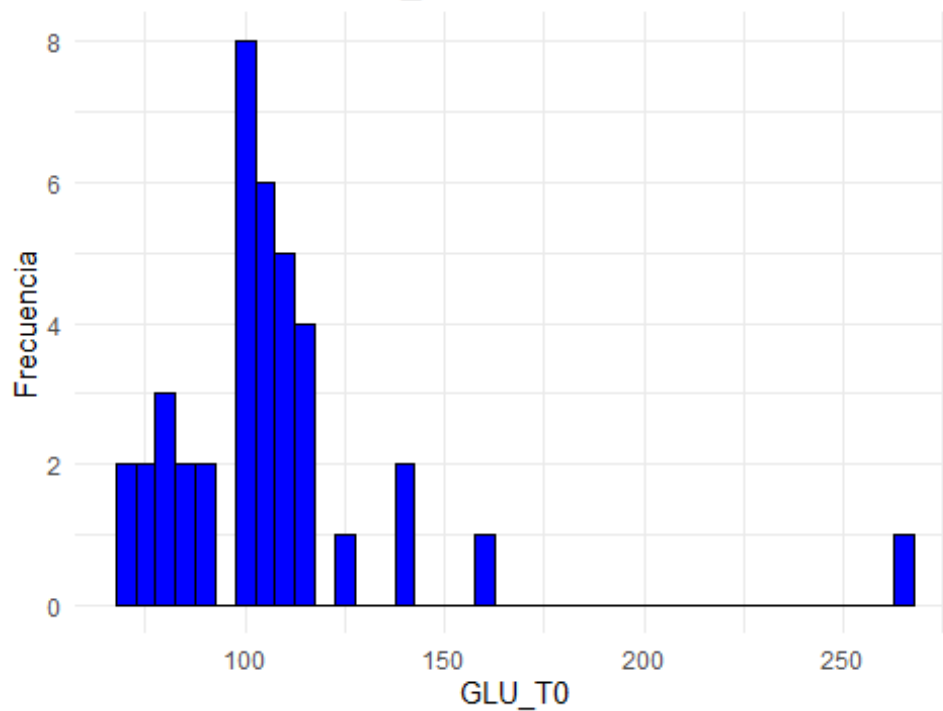
Distribución de PESO_T0



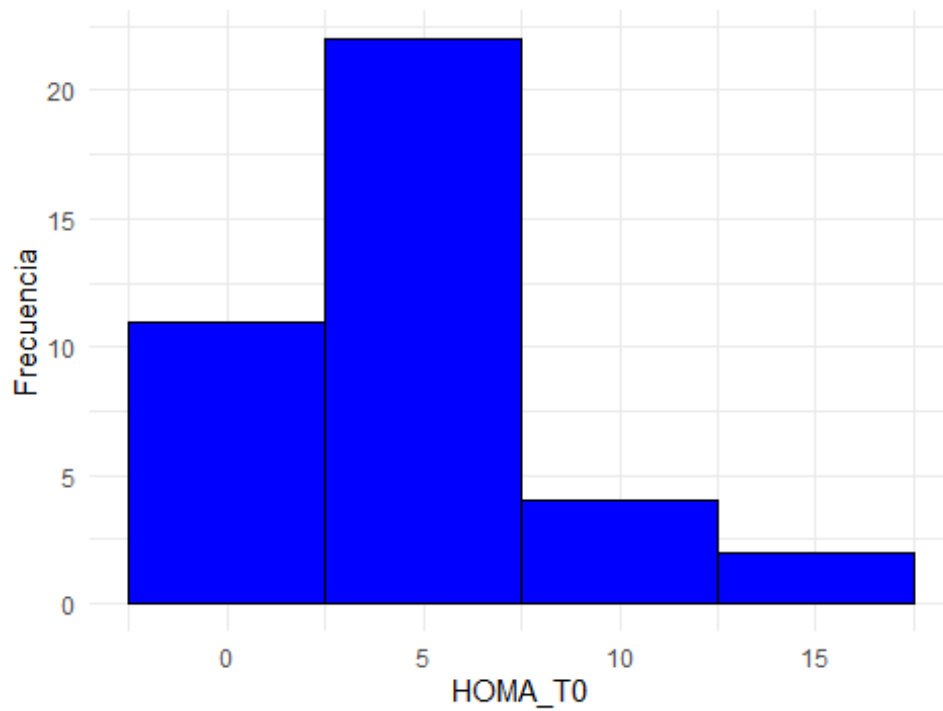
Distribución de bmi_T0



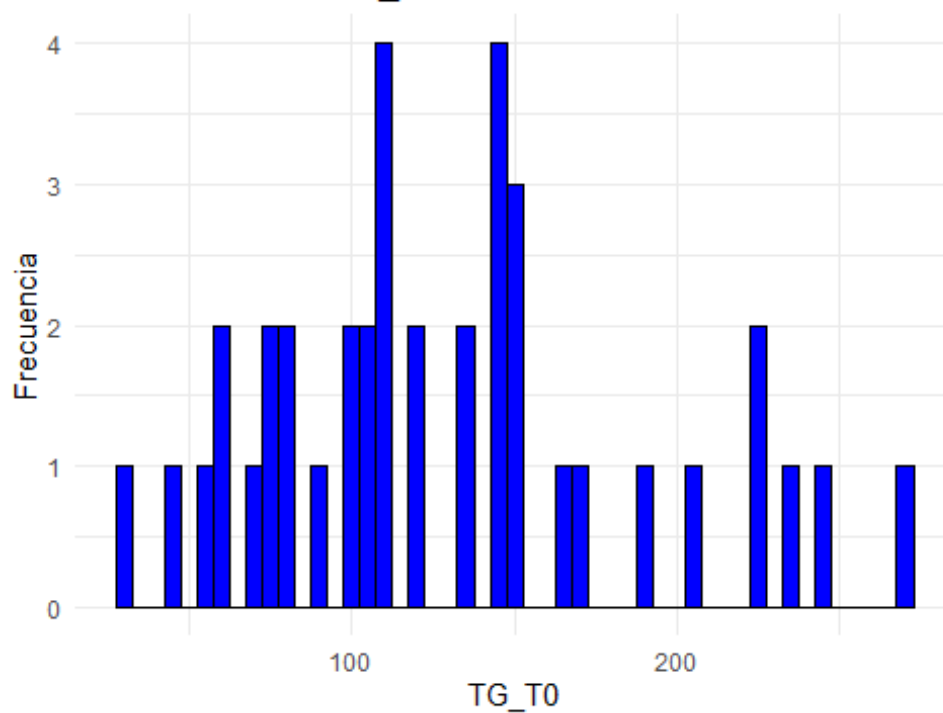
Distribución de GLU_T0

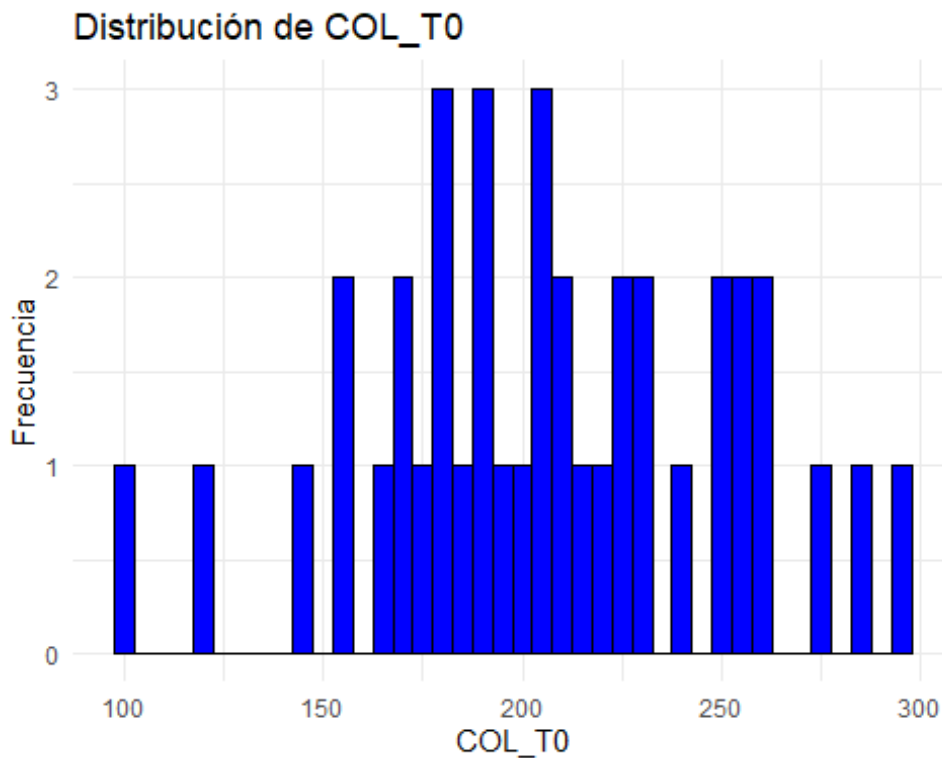


Distribución de HOMA_T0

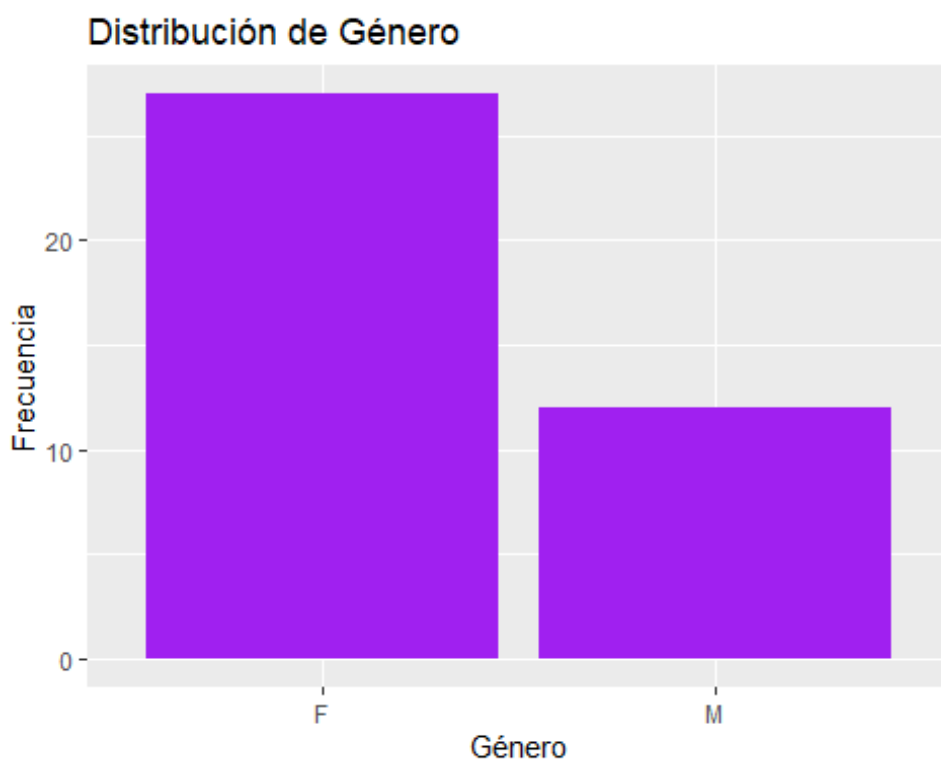


Distribución de TG_T0



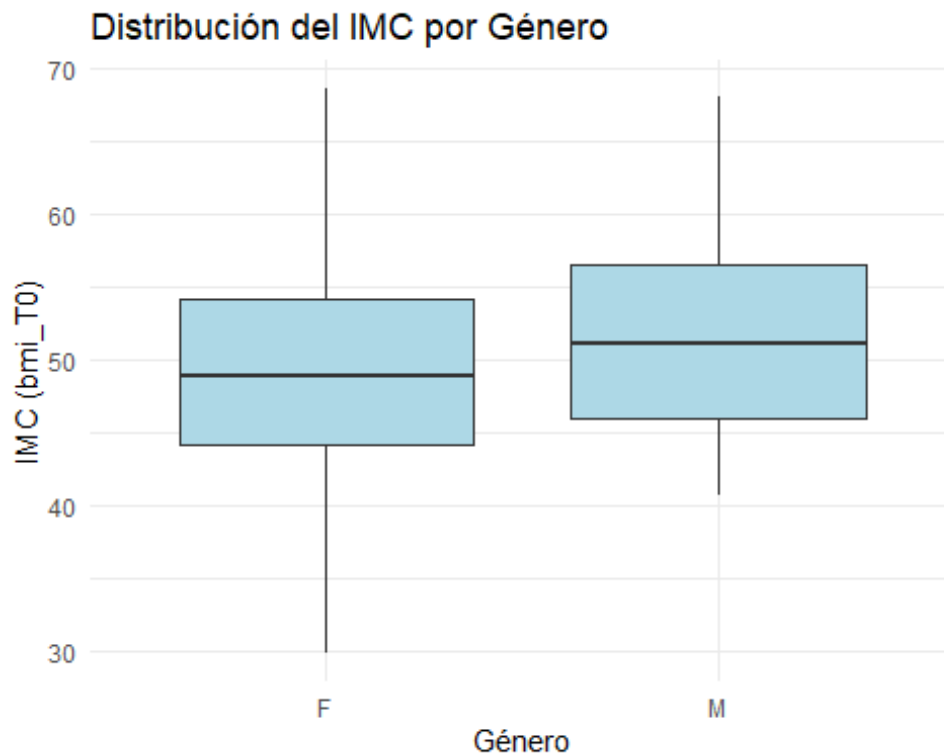


```
# Gráfico de barras para variable categórica (ej. GENDER)
ggplot(data_subset, aes(x = GENDER)) +
  geom_bar(fill = "purple") +
  labs(title = "Distribución de Género", x = "Género", y = "Frecuencia")
```



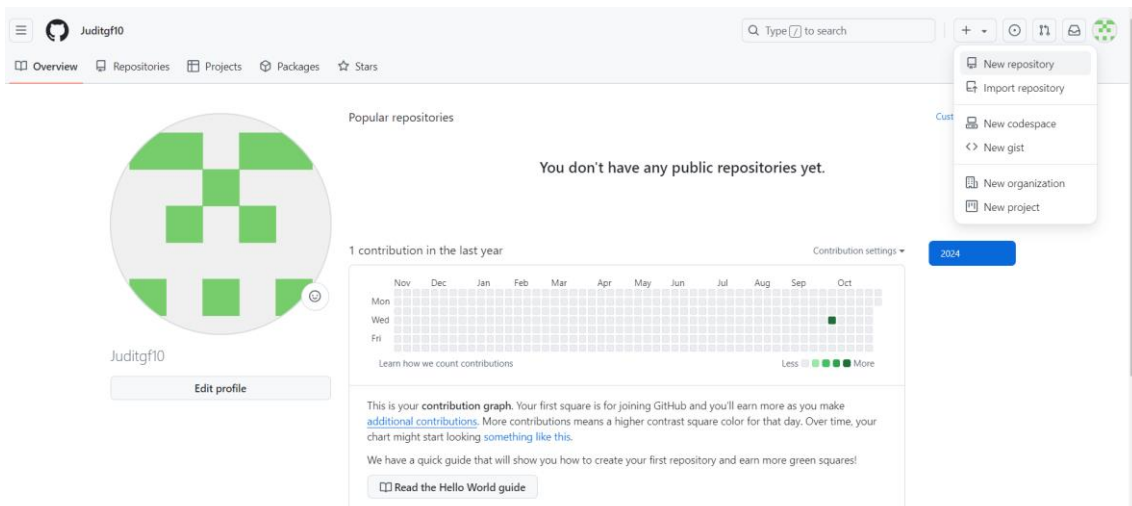
```
# Comparar BMI entre géneros
ggplot(data_subset, aes(x = GENDER, y = bmi_T0)) +
```

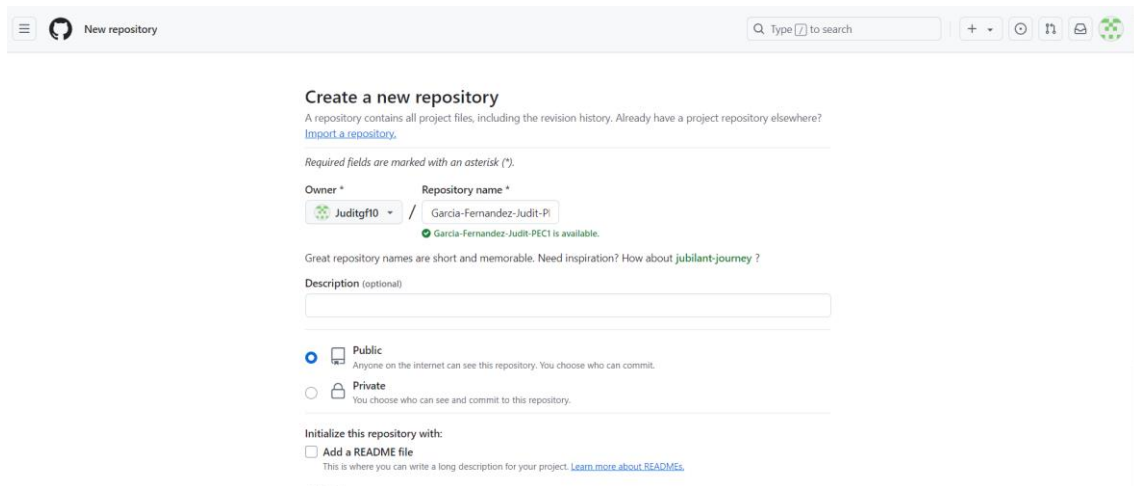
```
geom_boxplot(fill = "lightblue") +  
labs(title = "Distribución del IMC por Género", x = "Género",  
y = "IMC (bmi_T0)") +  
theme_minimal()
```



4. Reposición de los datos en Github

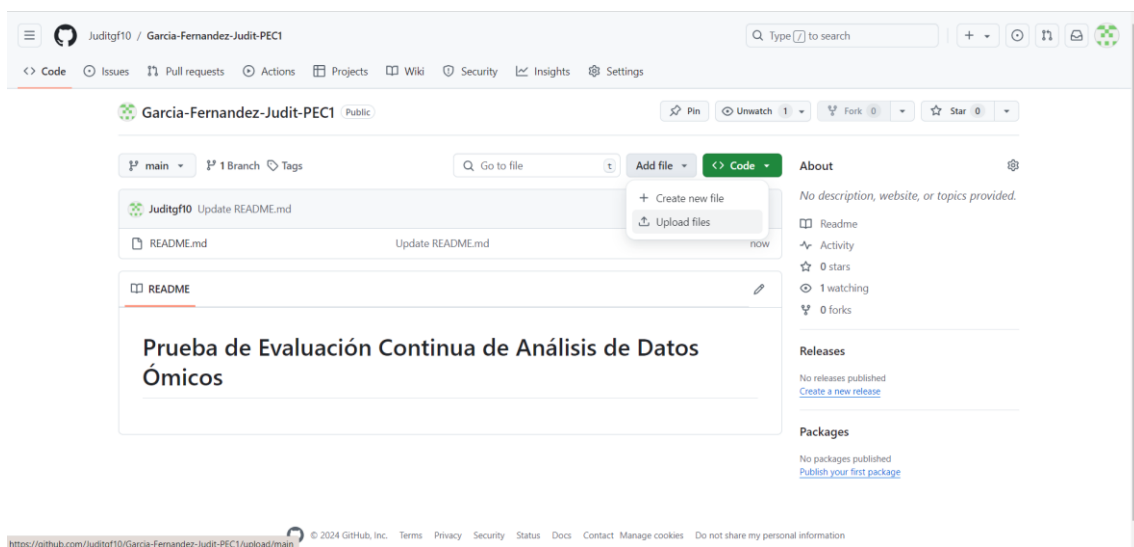
Para crear el repositorio de datos en Github, seleccionaremos “New repository”, le asignaremos un nombre y lo marcaremos como repositorio público, además de añadir el archivo README:





The screenshot shows the 'Create a new repository' page on GitHub. At the top, there's a search bar and navigation icons. The main heading is 'Create a new repository', followed by a subtext: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. Below this, a note states 'Required fields are marked with an asterisk (*)'. The form has two main sections: 'Owner' and 'Repository name'. The 'Owner' dropdown is set to 'Juditgf10'. The 'Repository name' dropdown is set to 'Garcia-Fernandez-Judit-PI', with a green checkmark indicating it's available. A hint says 'Great repository names are short and memorable. Need inspiration? How about [jubilant-journey](#) ?'. There's a 'Description (optional)' text area. Below that, there are two radio buttons for visibility: 'Public' (selected) and 'Private'. The 'Public' option has a subtext: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option has a subtext: 'You choose who can see and commit to this repository.' At the bottom, there's a section 'Initialize this repository with:' with a checkbox 'Add a README file' (unchecked). A subtext says 'This is where you can write a long description for your project. [Learn more about READMEs.](#)'.

Para añadir los archivos que se corresponden con esta PEC, en “Add file” > “Upload files”:



The screenshot shows the GitHub repository page for 'Garcia-Fernandez-Judit-PEC1'. The repository is public. The main content area shows the 'main' branch with a file 'README.md' that was updated 'now'. A dropdown menu is open over the 'Add file' button, showing options: '+ Create new file' and 'Upload files'. The 'About' section on the right shows 'No description, website, or topics provided.' and lists repository statistics: 0 stars, 1 watching, 0 forks. The 'Releases' section shows 'No releases published' with a link to 'Create a new release'. The 'Packages' section shows 'No packages published' with a link to 'Publish your first package'.

La URL para acceder a este repositorio Github es: <https://github.com/Juditgf10/Garcia-Fernandez-Judit-PEC1>