

# Healthcare

FELIPE NARVAEZ

2023-11-11

```
# Cargar librerías necesarias
```

```
library(ggplot2)
```

```
library(readr)
```

```
# Importar los datos desde el archivo CSV
```

```
healthcare_data <- read_csv("healthcare_dataset.csv")
```

```
## Rows: 10000 Columns: 15
```

```
## _____Column specification _____
```

```
## Delimiter: ","
```

```
## chr (10): Name, Gender, Blood Type, Medical Condition, Doctor, Hospital, In...
```

```
## dbl (3): Age, Billing Amount, Room Number
```

```
## date (2): Date of Admission, Discharge Date
```

```
##
```

```
## 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.
```

```
# Mostrar la especificación de las columnas
```

```
spec(healthcare_data)
```

```
## cols(
```

```
##   Name = col_character(),
```

```
##   Age = col_double(),
```

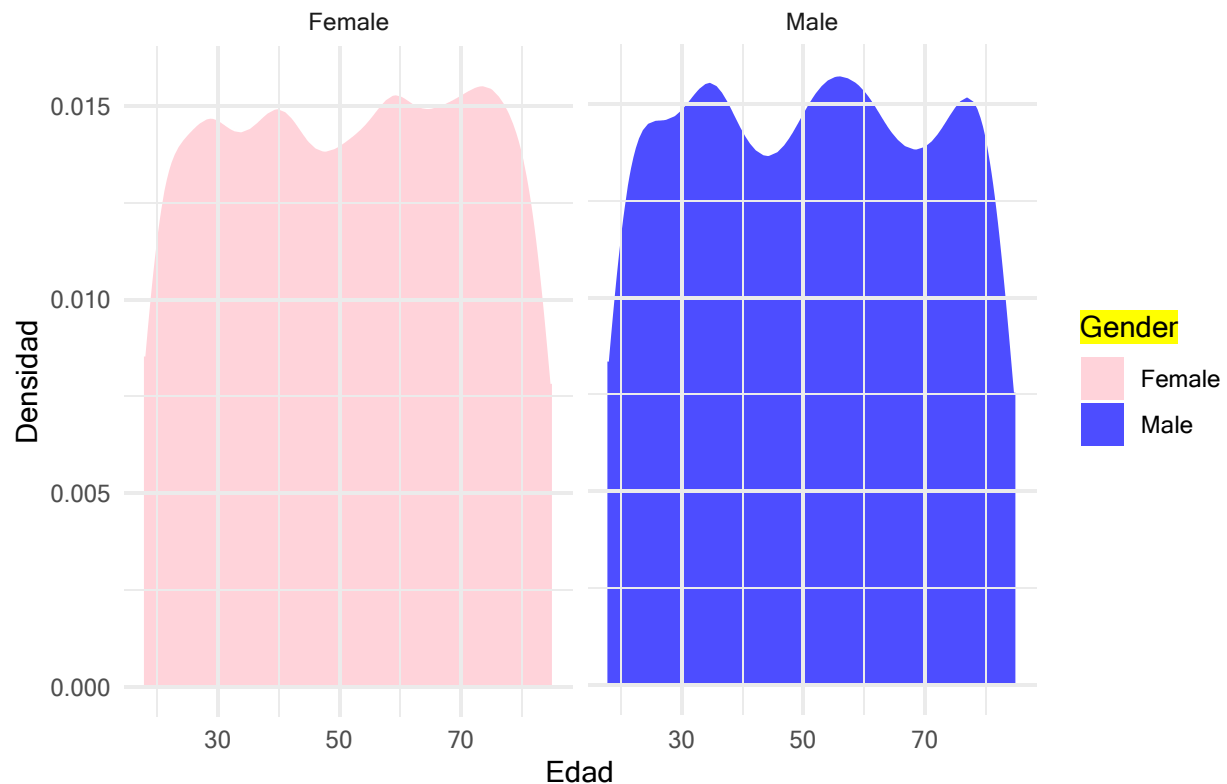
```
##   Gender = col_character(),
```

```
## 'Blood Type ' = col_character(),
## 'Medical Condition ' = col_character(),
## 'Date of Admission ' = col_date(format = ""),
## Doctor = col_character(),
## Hospital = col_character(),
## 'Insurance Provider ' = col_character(),
## 'Billing Amount ' = col_double(),
## 'Room Number ' = col_double(),
## 'Admission Type ' = col_character(),
## 'Discharge Date ' = col_date(format = ""),
## Medication = col_character(),
## 'Test Results ' = col_character()
## )
```

```
# Convertir la columna Age a tipo de dato entero
healthcare_data$Age <- as.integer(healthcare_data$Age)
```

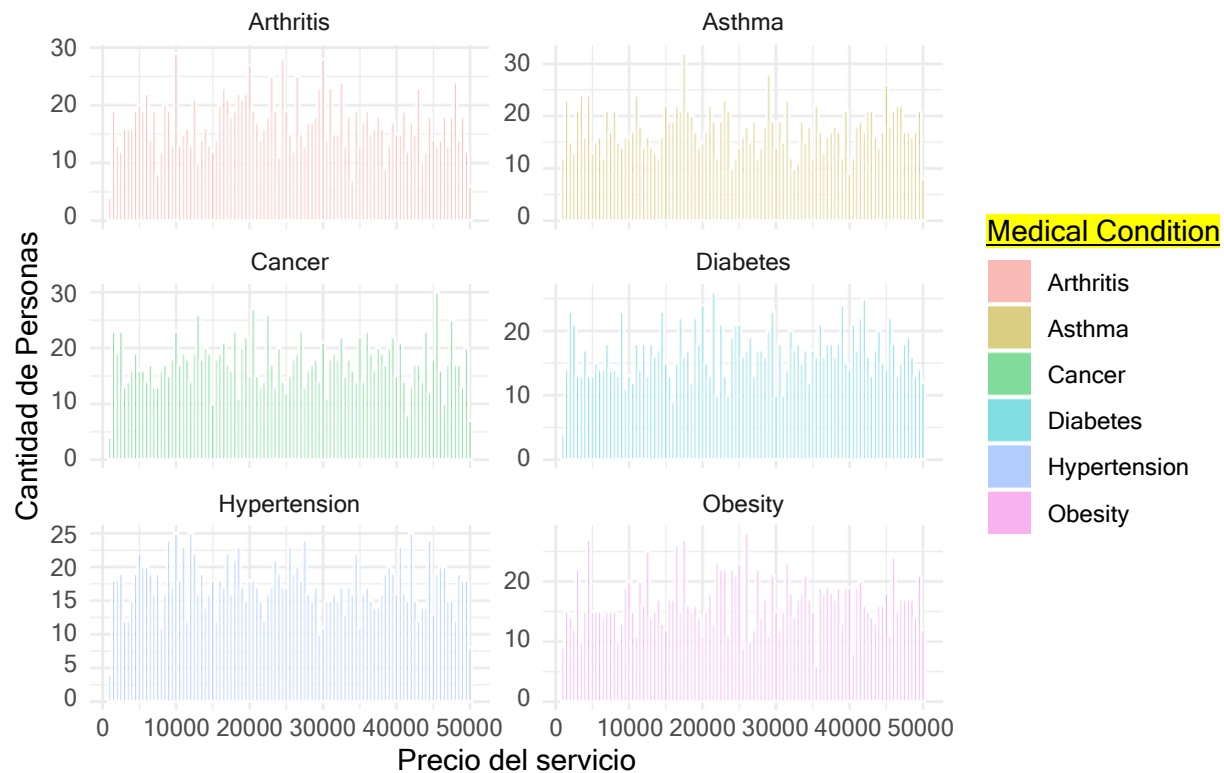
```
# se inicia la creación del gráfico utilizando la función ggplot(). Se especifican las variables x e fi
ggplot(healthcare_data, aes(x = Age, fill = Gender)) +
# geom_density() agrega una capa de densidad al gráfico. Se establece un nivel de transparencia (0.7) p
geom_density(alpha = 0.7, color = "white", position = "identity") +
# labs se utiliza para agregar etiquetas al gráfico, incluyendo el título y etiquetas para los ejes x e
labs(title = "Trazado de Densidad para la Edad por Género",
      x = "Edad",
      y = "Densidad") +
# theme_minimal() establece un tema visual minimalista para el gráfico.
theme_minimal() +
# facet_grid() permite facetar el gráfico por género. scales = "free_y" y space = "free_y" permiten que
facet_grid(. ~ Gender, scales = "free_y", space = "free_y") +
# scale_fill_manual() se utiliza para asignar colores personalizados a las curvas de densidad según la
scale_fill_manual(values = c("pink", "blue"))
```

## Trazado de Densidad para la Edad por Género



```
# Crear un histograma para la columna "Billing Amount" con segmentación por "Medical Condition"
ggplot(healthcare_data, aes(x = `Billing Amount`, fill = `Medical Condition`)) +
# geom_histogram() agrega una capa de histograma al gráfico. Se especifica el ancho de los intervalos (
geom_histogram(binwidth = 500, color = "white", alpha = 0.7, position = "identity") +
labs(title = "Histograma de Precio del servicio segun la condicion medica",
      x = "Precio del servicio",
      y = "Cantidad de Personas") +
theme_minimal() +
#facet_wrap() permite facetar el gráfico por "Medical Condition". scales = "free_y" y ncol = 2 permit
facet_wrap(~`Medical Condition`, scales = "free_y", ncol = 2)
```

## Histograma de Precio del servicio segun la condicion medica



*# Filtrar las primeras 200 filas (personas) del conjunto de datos*

```
subset_data <- healthcare_data[1:200, ]
```

*# Crear un diagrama de caja con puntos para las columnas "Age" y "Medical Condition"*

```
ggplot(subset_data, aes(x = `Medical Condition`, y = Age, fill = `Medical Condition`)) +
```

*# geom\_boxplot() agrega una capa de diagrama de caja al gráfico. Se establece un nivel de transparencia*

```
geom_boxplot(alpha = 0.7) +
```

*# geom\_jitter() agrega una capa de puntos (jitter) al gráfico para mostrar la dispersión de datos. Se u*

```
geom_jitter(position = position_jitter(width = 0.2, height = 0), alpha = 0.5, color = "black") +
```

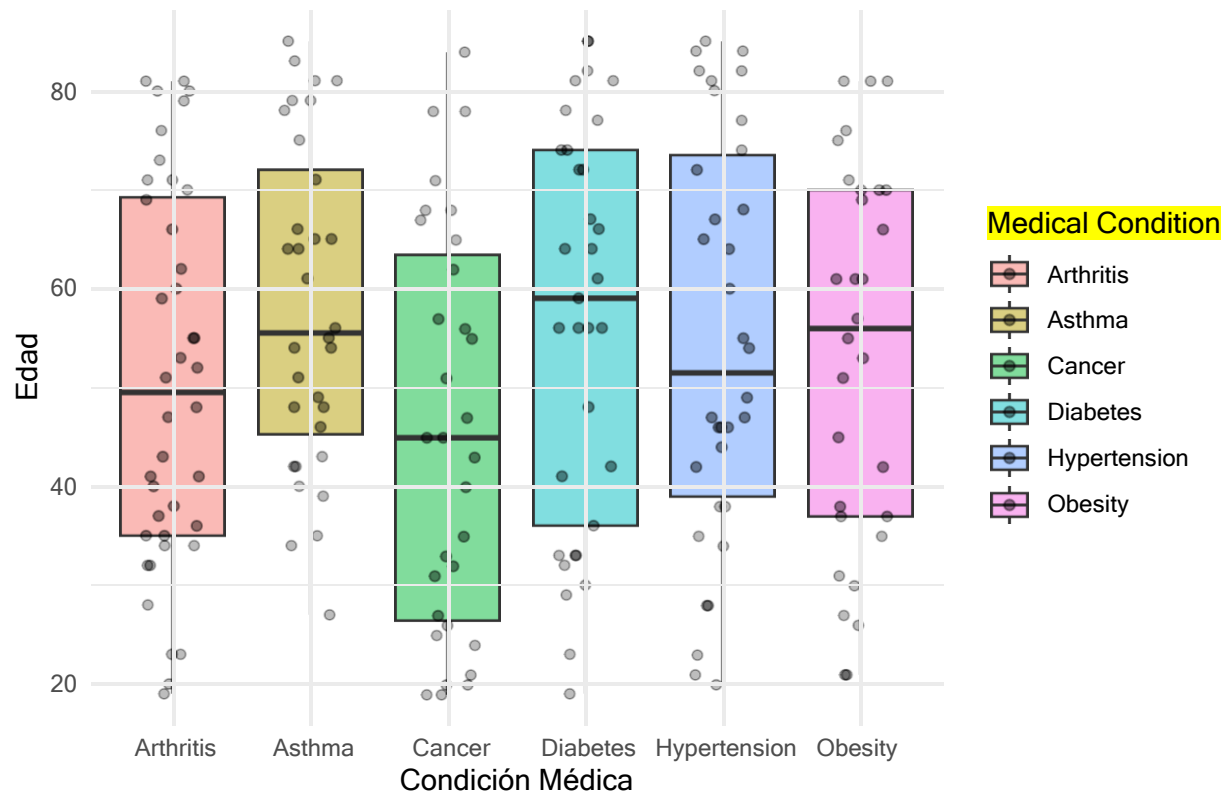
```
labs(title = "Diagrama de Caja con Puntos de Edad por Condición Médica",
```

```
  x = "Condición Médica",
```

```
  y = "Edad") +
```

```
theme_minimal()
```

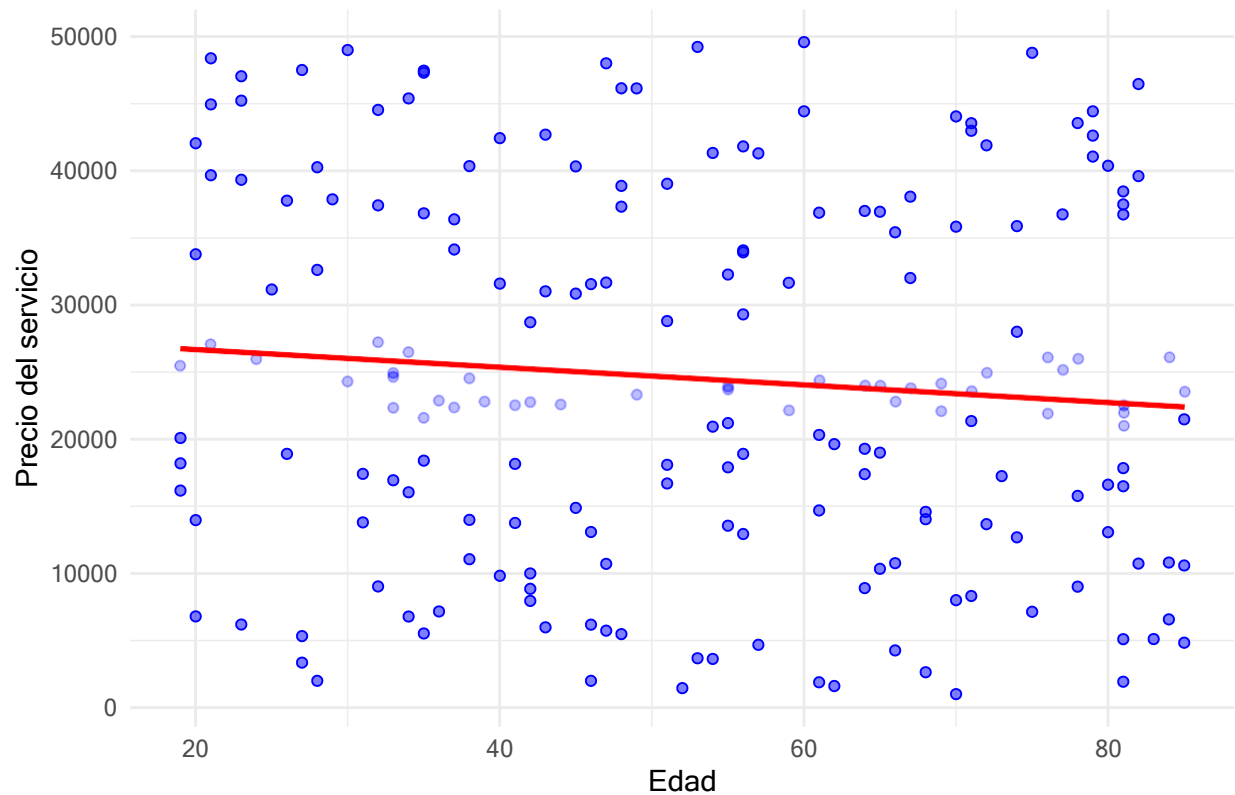
## Diagrama de Caja con Puntos de Edad por Condición Médica



```
# Filtrar las primeras 200 filas (personas) del conjunto de datos
subset_data <- healthcare_data[1:200, ]
```

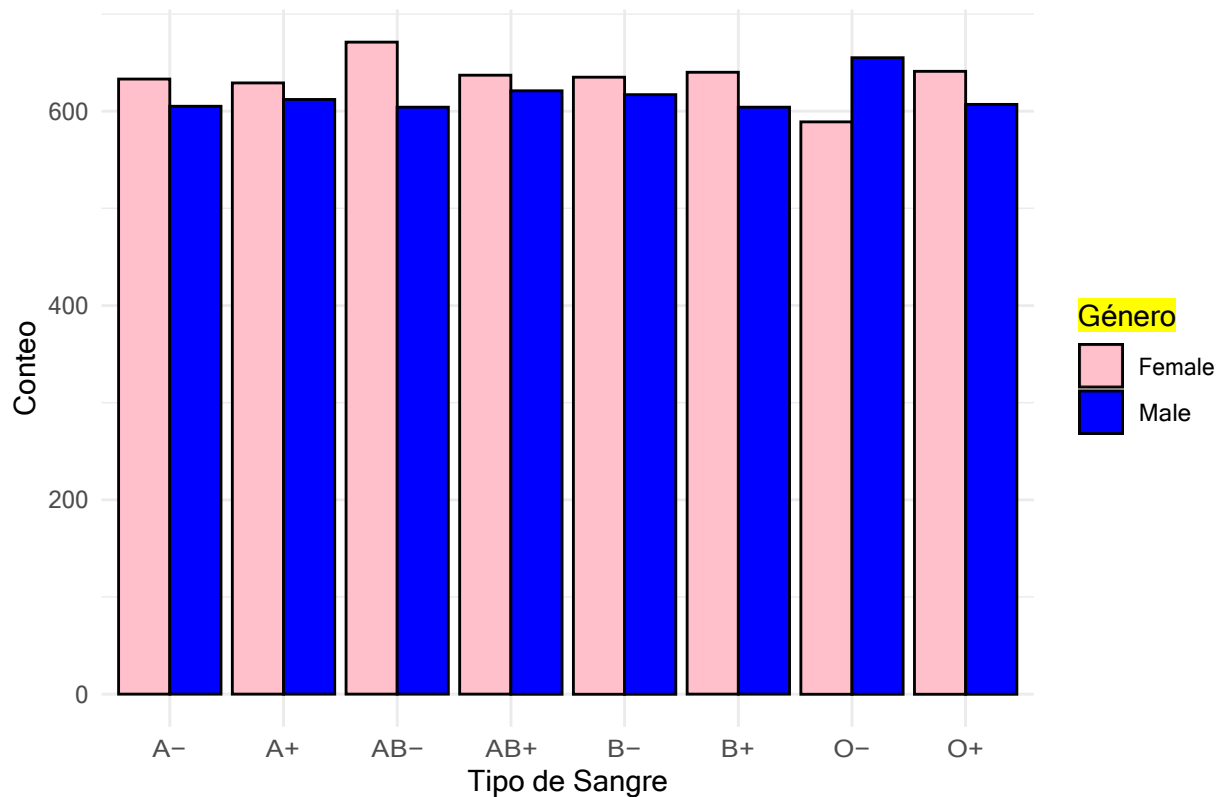
```
# Crear un diagrama de dispersión con tendencia lineal para las columnas "Billing Amount" y "Age"
ggplot(subset_data, aes(x = Age, y = `Billing Amount`)) +
# geom_point() agrega una capa de puntos al gráfico para mostrar la relación entre "Billing Amount" y "
geom_point(alpha = 0.5, color = "blue") +
# geom_smooth() agrega una capa de tendencia lineal al gráfico. Se utiliza el método de regresión lineal
geom_smooth(method = "lm", se = FALSE, color = "red", formula = y ~ x) +
labs(title = "Diagrama de Dispersión con Tendencia Lineal",
      x = "Edad",
      y = "Precio del servicio") +
theme_minimal()
```

### Diagrama de Dispersión con Tendencia Lineal



```
# Diagrama de barras para el tipo de sangre segun el Género
ggplot(healthcare_data, aes(x = `Blood Type`, fill = `Gender`)) +
  #geom_bar agrega barras al gráfico. Se utiliza position = "dodge" para que las barras de diferentes gé
  geom_bar(position = "dodge", color = "black", stat = "count") +
  labs(title = "Distribución de Género y Tipo de Sangre",
        x = "Tipo de Sangre",
        y = "Conteo",
        fill = "Género") +
  scale_fill_manual(values = c("pink", "blue")) + # Asignar colores personalizados
  theme_minimal()
```

## Distribución de Género y Tipo de Sangre



*# Filtrar las primeras 100 filas (personas) del conjunto de datos*

```
subset_data <- healthcare_data[1:100, ]
```

*# Crear un diagrama de dispersión conectado para las columnas "Age" y "Billing Amount"*

```
ggplot(subset_data, aes(x = Age, y = `Billing Amount`)) +
```

*#geom\_line() agrega una línea que conecta los puntos en el gráfico. Se ha establecido el color de la*

```
geom_line(color = "blue") +
```

*# geom\_point agrega puntos al gráfico. Se ha establecido la forma (shape) del punto como 22 (un tipo*

```
geom_point(shape = 22, fill = "red", size = 2) +
```

```
labs(title = "Diagrama de Líneas Conectado de Edad y Precio del Servicio",
```

```
  x = "Edad",
```

```
  y = "Precio del Servicio") +
```

```
theme_minimal()
```

Diagrama de Líneas Conectado de Edad y Precio del Servicio

