Revista digital
Matemática, Educación e Internet
https://revistas.tec.ac.cr/index.php/matematica
Vol 24, No 2. Marzo, 2024 — Agosto, 2024

ISSN 1659 -0643
DOI: 10.18845/rdmei.v24i2.6914

Introducción a la visualización de datos con ggplot2

Introduction to data visualization with ggplot2

Katalina Oviedo Rodríguez[1]
katalina.oviedo.rodriguez@una.ac.cr
Universidad Nacional
Heredia, Costa Rica

Byron Jiménez Oviedo[2]
byron.jimenez.oviedo@una.ac.cr
Universidad Nacional
Heredia, Costa Rica

Eduardo Aguilar Fernández[3]
eduardo.aguilar.fernandez@una.cr
Universidad Nacional
Heredia, Costa Rica

Recibido: 2 mayo de 2023 Aceptado: 20 Agosto 2023

Resumen: En este documento se presenta una recopilación de los principales códigos del paquete `ggplot2` de `R` para representar información de forma gráfica. Los gráficos que aquí se exponen son los que usualmente se estudian en cursos básicos de estadística descriptiva en la educación superior. El objetivo principal es que esta recopilación sirva de guía para que tanto estudiantes como profesores puedan consultar de una manera más sintetizada algunos de los códigos y ambientes que existen para visualizar información de forma gráfica en `ggplot2`. Las bases de datos utilizadas son de libre acceso y se pueden descargar en `R`. En este documento se incluyen los enlaces a las mismas, lo que permite que el lector pueda replicar los códigos con estas bases para una mejor comprensión. Se asume que el lector posee un conocimiento básico del lenguaje `R` y de estadística descriptiva.

Palabras Clave: Visualización, datos, gráficos estadísticos, ggplot2, R.

Abstract: This document presents a compilation of the main codes of the `ggplot2` package of `R` to present information graphically. The graphs presented here are those that are usually studied in basic descriptive statistics courses in higher education. The main objective is that this collection serves as a guide so that both students and teachers can consult in a more synthesized way some of the codes and environments that exist to visualize information graphically in `ggplot2`. The databases used are freely accessible and can be downloaded at `R`. This document includes links to them, which allows the reader to replicate the codes with these databases for a better understanding. It is assumed that the reader has a basic understanding of `R` language and descriptive statistics.

Keywords: Visualization, data, statistical graphs, ggplot2, R.

[1]Katalina Oviedo Rodríguez, académica e investigadora de la Universidad Nacional de Costa Rica. Dirección postal: Mercedes Norte, Heredia, Costa Rica. Código postal: 40102. Correo electrónico: katalina.oviedo.rodriguez@una.ac.cr.

[2]Byron Jiménez Oviedo, académico e investigador de la Universidad Nacional de Costa Rica. Dirección postal: Mercedes Norte, Heredia, Costa Rica. Código postal: 40102. Correo electrónico: byron.jimenez.oviedo@una.ac.cr.

[3]Eduardo Aguilar Fernández, académico e investigador de la Universidad Nacional de Costa Rica. Dirección postal: Barva, Heredia. Código postal: 40201. Correo electrónico: eduardo.aguilar.fernandez@una.ac.cr.

1. Introducción

La visualización de datos y su análisis se consideran procesos de gran importancia en la actualidad, siendo los gráficos una de las representaciones más utilizadas para presentar información estadística. Casanova (2017) señala que la carga heurística que incorporan los gráficos alimenta la conjetura de quien realiza los análisis, lo que facilita la interpretación, la formulación de hipótesis, las conjeturas, las explicaciones, etc. Una buena presentación gráfica es muy valorada (Sevilla, 2005; Wickham, 2016; Wilkinson, 2012), por tal razón es que el profesional actual debe tener las herramientas mínimas para poder desplegar tal análisis de una manera elegante, de fácil interpretación y precisa.

En la actualidad, `R` es uno de los entornos y lenguajes de programación más utilizados para análisis estadístico, minería de datos, modelización estadística y presentación gráfica de la información (Dobrow, 2016; Kabacoff, 2022; Monahan, 2011; Wickham, 2016; Wickham & Grolemund, 2016). Además, es un entorno abierto, libre, gratuito que ha tenido gran aprobación en la academia, en la investigación científica y en el sector empresarial. Uno de los compiladores que se pueden utilizar es `RStudio`, el cual es utilizado en este documento.

Existen distintos programas y lenguajes que permiten realizar análisis estadístico de datos tales como R, Python, Excel, Power BI, SPSS, PSPP, entre otros, pero no es el objetivo de este documento hacer una comparación de estos; sin embargo, es muy deseable que el profesional actual tenga la capacidad de comparar, discernir y seleccionar entre estos según sus necesidades.

Este documento tiene como objetivo principal brindar un acercamiento a la representación gráfica de información utilizando el paquete `ggplot 2` en `R`, a partir de los gráficos que generalmente se presentan en un curso universitario de estadística descriptiva, entre ellos: gráficos de barras, circular, de líneas, de cajas y bigotes, histogramas, polígonos de frecuencias y ojivas. Se seleccionaron estos tipos de gráficos con el fin de que el estudiantado universitario que da sus primeros pasos en el ambiente `R` tenga un acercamiento amigable y útil a `ggplot2` y para que pueda aprovecharlo en los cursos o ambientes que lo demanden. Las bases de datos utilizadas contienen datos reales y son de libre acceso.

El documento inicia con una introducción sobre el paquete`ggplot 2`, seguido, se describen las bases de datos utilizadas, luego, se brindan ejemplos de algunos elementos gráficos y finalmente se presenta una descripción detallada de la construcción según cada tipo de gráfico.

2. Paquete ggplot2

El paquete `ggplot2` fue creado por Hadley Wickham y fue desarrollado utilizando la gramática de gráficos de Wilkinson (Wickham, 2016). La gramática de gráficos dicta que un gráfico estadístico es una forma de estructurar los datos usando atributos estéticos como color, tamaño, forma (llamados aesthetic attributes) y objetos geométricos como puntos, líneas o barras. Además de transformaciones estadísticas, escalas, coordenadas y el faceting o facetado, lo cual permite visualizar una cierta característica o variable de los datos en subgráficos independientes (Wilkinson, 2012; Wickham, 2016; Teutonico, 2015).

Para poder utilizar `ggplot2` es necesario realizar su instalación y contar con una `versión` reciente de `R`, por ejemplo 3.2.0 en adelante. La versión se puede verificar utilizando el comando *version* en la consola. Luego, una vez que se tenga la versión adecuada, se puede instalar `ggplot2` utilizando el comando usual para la instalación de paquetes en `R` el cual es:

```
install.packages("ggplot2")
library(ggplot2)
```

2.1. Bases de datos empleadas

Una base de datos o dataframe es una estructura que sirve para organizar datos. Es como una matriz con la diferencia de que puede tener columnas de diferentes tipos, de modo que en la primera columna se encuentra cada unidad estadística que conforma la población o muestra con la cual se trabaja.

Con el fin de facilitar el tratamiento de los gráficos para el lector, se utilizan bases de datos de libre acceso que se pueden obtener del paquete `Biostatistics` por medio del enlace https://cran.r-project.org/web/packages/Biostatistics/index.html, esto pues se quiere enfatizar en la representación gráfica de los datos y que el lector pueda replicar de una manera rápida y sencilla los códigos para la construcción de los diferentes gráficos.

El paquete `Biostatistics` proporciona diferentes bases de datos, en este documento se utiliza la base de datos llamada `worldbank` que presenta una variedad de medidas geográficas, económicas, ambientales y sociales para 186 países desde el año 2014, recopiladas a partir de datos publicados por World Bank, los cuales se pueden obtener y verificar en la dirección https://www.worldbank.org/en/home. La descripción de las variables que presenta esta base de datos se muestra en la tabla 1.

Tabla 1: Descripción de las variables de la base de datos worldbank. Fuente: Elaboración propia

Climate_region	Factor con niveles Templado o Polar Tropical
Income_binary	Factor con niveles alta baja
Country_Name	Nombre del país
Country_Code	Código de tres letras del país
Region	Región geográfica
Income_group	Divide a los países en uno de cuatro grupos de ingresos
Population	Tamaño de la población
Land_area	Área del país en km^2
Forest_area	Superficie boscosa como porcentaje de la superficie terrestre
Precipitation	Precipitación anual en mm
Population_density	Personas por km^2
Capital_lat	Latitud de la capital
GNP_per_Cap	Producto Nacional Bruto per cápita en $
Population_growth	Crecimiento anual de la población en porcentaje
Cereal_yield	Rendimiento de cereales en Kg por Ha
Female_life_expectancy	Esperanza de vida media de las mujeres en años
Under_5_mortality	Muertes de niños menores de 5 años por 100000
Renewables	Consumo de energía renovable (porcentaje del consumo total de energía final)
CO2	Producción de CO_2 en toneladas per cápita
PM25	Contaminación del aire PM2.5, exposición anual media (microgramos por m^3)
Women_in_parliament	Porcentaje de escaños ocupados por mujeres en los parlamentos nacionales
GINI_index	Índice Gini de desigualdad de la riqueza
Govt_spend_education	Gasto público en educación, total (porcentaje del PIB)
Secondary_school_enrolment	Matricula escolar, secundaria (porcentaje neto)
School_gender_parity	Índice de paridad de género para la matrícula escolar

Para poder instalar y ver una parte esta base de datos se utiliza el siguiente código.

```
install.packages("Biostatistics")
library("Biostatistics")
head(worldbank)
```

Para evitar algunos inconvenientes con los datos, ya que esto podría distorsionar el sentido de este documento, se puede omitir algunas de las unidades estadísticas que tiene datos faltantes (NAs) en algunas de las variables que se van a utilizar. Para esto se utiliza lo siguiente.

```
faltantes = is.na(worldbank$Forest_area) |
            is.na(worldbank$Precipitation)
worldbank = subset(worldbank, subset = !faltantes)
```

2.2. Un acercamiento a través de ejemplos

En esta sección es presenta cada parte de la construcción de un gráfico estadístico básico usando `ggplot2`. Más precisamente, se presentan las tres componentes o capas claves de un gráfico en `ggplot2`, a saber:

1. Los datos (data): en este caso se trata de la base de datos utilizada.
2. La estética (aesthetics, aes): relaciona las variables deseadas de la base de datos y la forma de percibirla.
3. La geometría (geom): esta capa indica como se quiere representar cada observación.

Por ejemplo, usando la base `worldbank` se pueden visualizar las variables `Forest_area` y `Precipitation` por medio de un diagrama de dispersión (ver Figura 1).

```
ggplot(data = worldbank, aes(x = Forest_area, y = Precipitation)) +
geom_point()
```

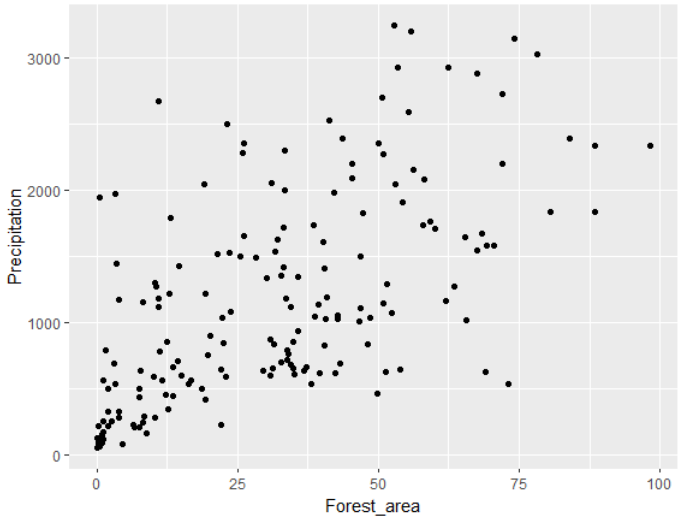

Figura 1: Gráfico de dispersión para las variables `Forest_area` y `Precipitacion`

Hasta este momento se tiene la estructura ggplot (datos, aes (x = , y =)), luego, para insertar una nueva capa se utiliza el símbolo "+ ". En el ejemplo anterior se insertó la capa de la geometría, específicamente, se usó puntos (`geom_point`) para representar el cruce de los valores de las dos variables seleccionadas. Por ejemplo, si se quiere usar la geometría de histograma para la variable `Forest_area`, la cual tiene una naturaleza continua, se utiliza la `geom_histogram` y se escribe la siguiente instrucción, la cual permite generar la Figura 2.

```
ggplot(data = worldbank, aes(x = Forest_area)) +
geom_histogram()
```

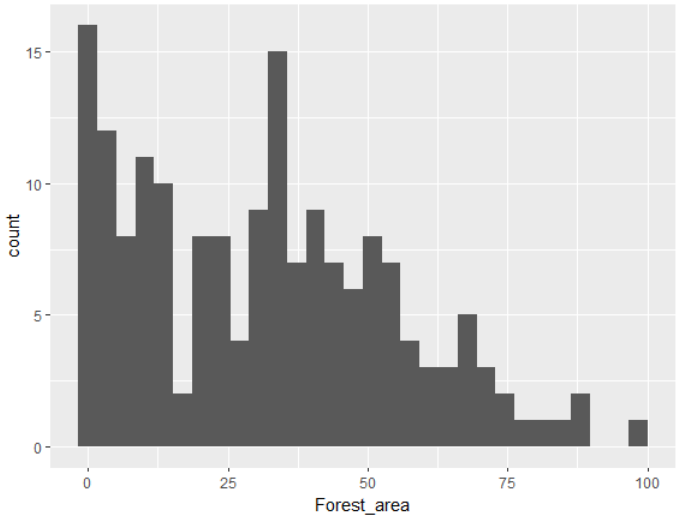

Figura 2: Histograma para la variable `Forest_area`

Existen muchas geometrías que se pueden invocar, como líneas, barras, cajas o densidad. Algunas de estas se explicarán más adelante.

Por otro lado, `aesthetics` tiene varios atributos que pueden ser de gran utilidad, por ejemplo, el color, la forma y el tamaño. En el caso del ejemplo de la Figura 1 en el cual se graficaron los valores de las variables `Forest_area` y `Precipitacion` con puntos, se puede agregar color para cada punto según el clima de la región (ver Figura 3). Una instrucción que permite asignar el color a cada punto es la siguiente:

```
ggplot(data = worldbank, aes(x=Forest_area,y=Precipitation, colour =
    ↪ Climate_region)) +
geom_point()
```

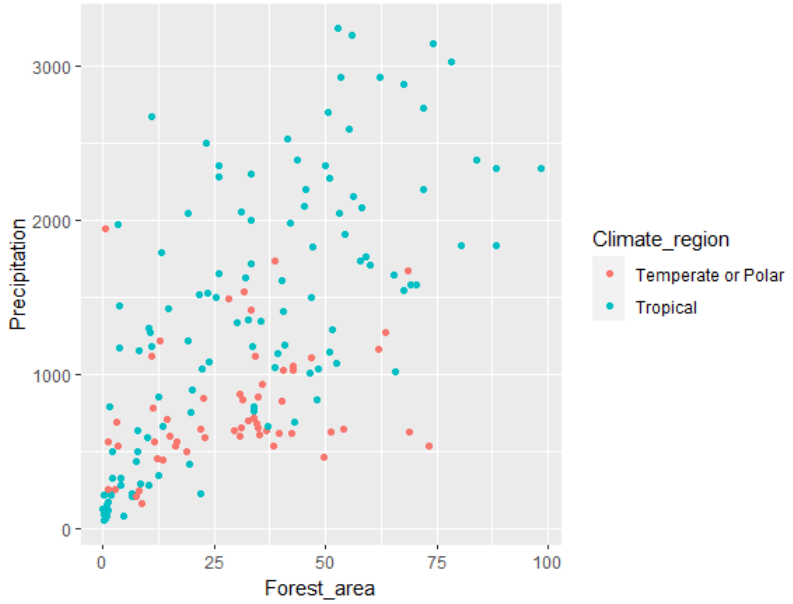

Figura 3: Gráfico de dispersión para `Forest_area` y `Precipitacion`, clasificada (color) por clima de la región

Un efecto muy similar se puede obtener utilizando la forma (shape) o el tamaño (size), tal y como se muestra en la Figura 4.

```
ggplot(data = worldbank, aes(x = Forest_area, y = Precipitation, shape =
    ↪ Climate_region)) +
geom_point()
```

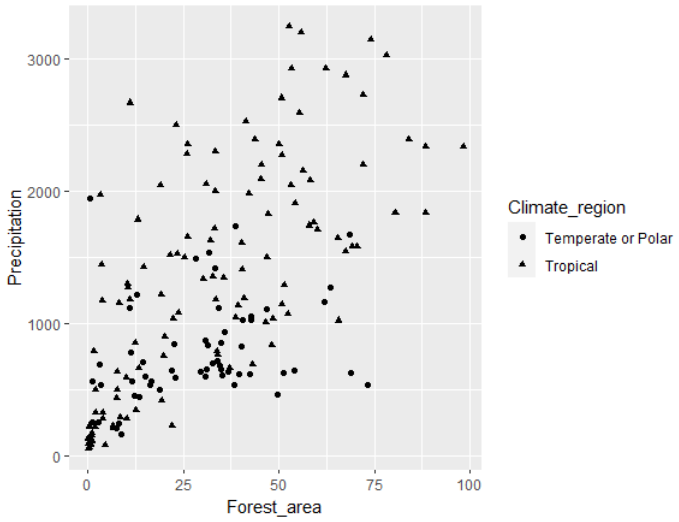

Figura 4: Gráfico de dispersión para `Forest_area` y `Precipitacion`, clasificada (forma) por clima de la región

Es claro que los diferentes atributos se adecuan mejor según el tipo de variable, por ejemplo, se puede utilizar color y forma para variables categóricas y tamaño para variables continuas (ver Figura 5). En el caso de que se quiera un cambio de color de la geometría usada, basta indicarlo dentro de esta capa.

```
ggplot(data = worldbank, aes(x = Forest_area, y = Precipitation, shape =
    ↪ Climate_region)) +
geom_point(color = "red")
```

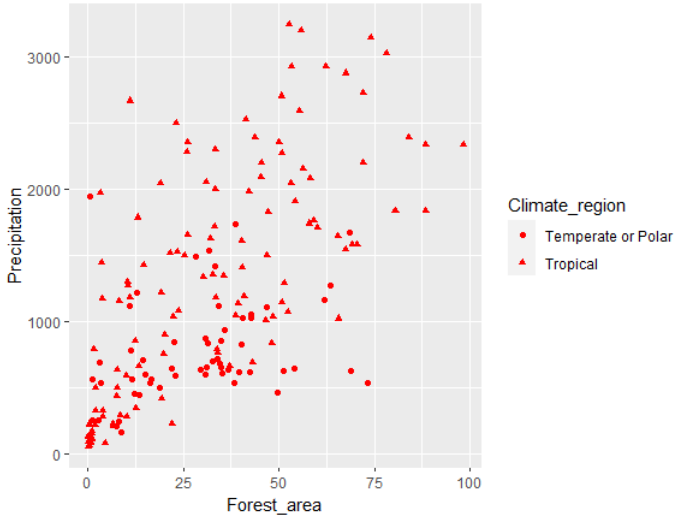

Figura 5: Gráfico de dispersión para `Forest_area` y `Precipitacion`, clasificada por clima de la región

Se puede observar que solo se ha indicado el color en inglés y en comillas, sin usar `aes`.

Otra forma de agregar categorías adicionales en una gráfica es por medio del comando `face_wrap`, el cual permite crear gráficos multi-panel, es decir, tablas de gráficos mostrando un subconjunto diferente de los datos (ver Figura 6). Este se agrega como una nueva capa usando el símbolo + y luego se especifíca la variable deseada antecedida del símbolo ∼.

```
ggplot(data = worldbank, aes(x = Forest_area, y = Precipitation)) +
geom_point() +
facet_wrap(~ Climate_region)
```

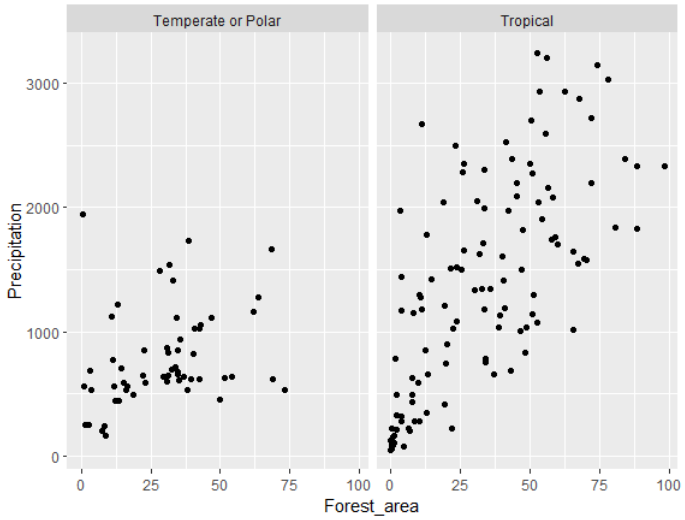

Figura 6: Gráfico de dispersión multi-panel para las variablesForest_area y Precipitacion, clasificada por clima de la región.

En algunas ocasiones se quiere marcar la tendencia que presentan las variables dentro del gráfico de dispersión. Para realizar esto se puede agregar a dicho gráfico una curva estándar con geom_smooth, la cual es una capa de geometría que ajusta una curva a los datos y despliega el error estándar (ver Figura 7). Este comando tiene varios métodos para calcular la curva. Por ejemplo, si se quiere un modelo lineal se utiliza el siguiente comando.

```
ggplot(data = worldbank, aes(x = Forest_area, y = Precipitation)) +
geom_point() +
geom_smooth(method = "lm")
```

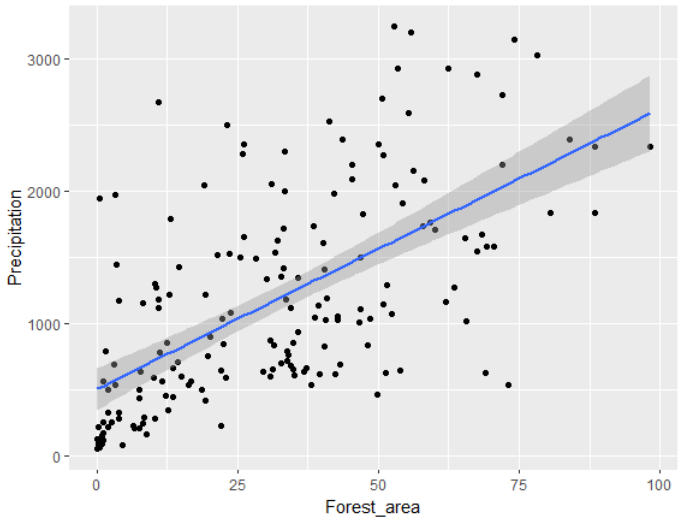

Figura 7: Gráfico de dispersión para las variables Forest_area y Precipitacion, con línea de tendencia.

Ahora, si se quiere la curva que mejor se ajuste con ciertas ondulaciones se utiliza el método loess donde las ondulaciones se pueden controlar con el parámetro span con valores entre 0 (mayor cantidad de ondulaciones) y 1 (menor cantidad de ondulaciones) (ver Figura 8).

```
ggplot(data = worldbank, aes(x = Forest_area, y = Precipitation)) +
geom_point() +
geom_smooth(span = 0.3)
```

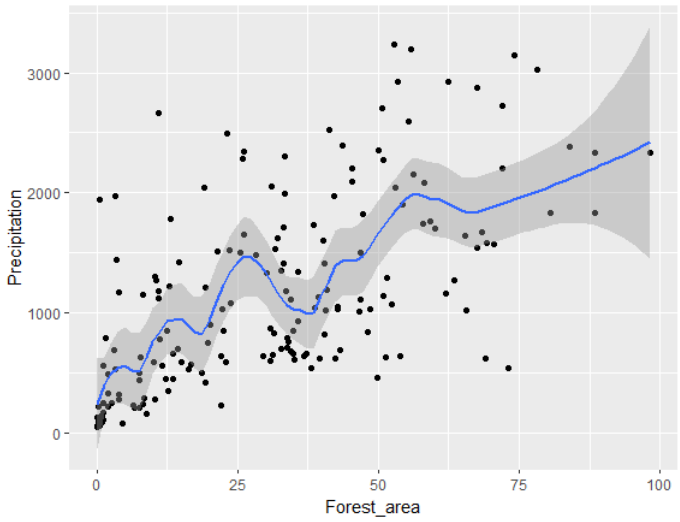

Figura 8: Gráfico de dispersión para las variables `Forest_area` y `Precipitacion`, con curva de tendencia.

Por defecto, `geom_smooth` tiene los párametros `method = "loess"`, `span = 1` (ver Figura 9).

```
ggplot(data = worldbank, aes(x = Forest_area, y = Precipitation))+
geom_point() +
geom_smooth()
```

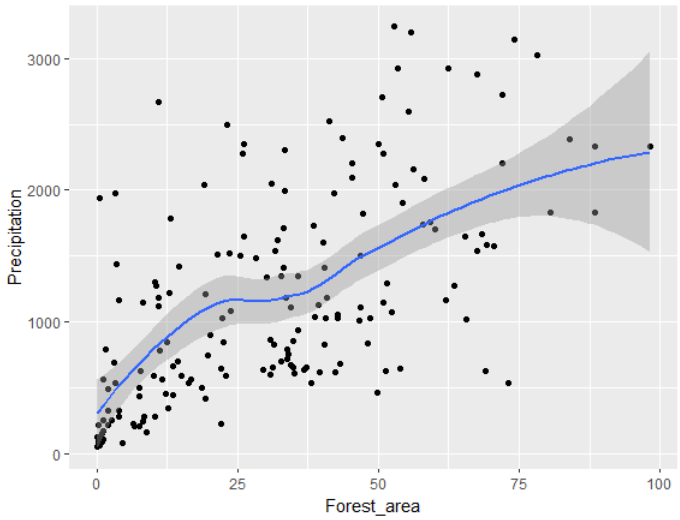

Figura 9: Gráfico de dispersión para las variables `Forest_area` y `Precipitacion`, con curva (por defecto) de tendencia.

Existen otros métodos como `gam` y `rlm`; sin embargo, estos se dejan para que la persona pueda investigar cuando conviene utilizarlos. Por otro lado, si se desea un gráfico con los mínimos detalles y dejando la elección del tipo de gráfico puede utilizarse la graficación rápida usando el comando `qplot` (ver Figura 10) (la letra `q` viene de `quick` que significa `rápido`).

```
qplot(Forest_area, data = worldbank)
```

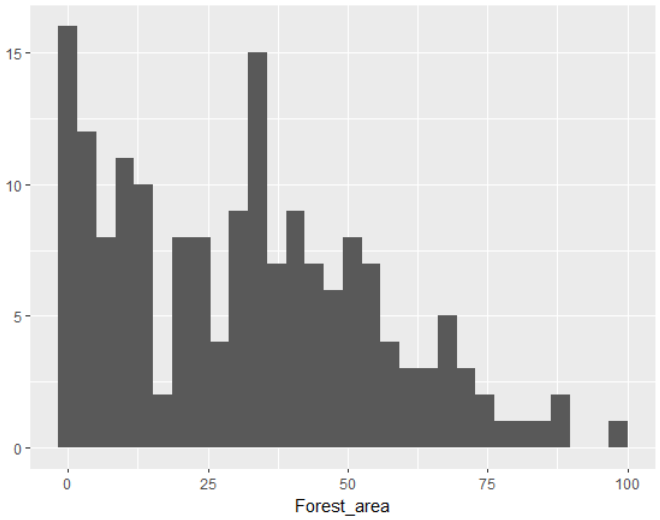

Figura 10: Histograma para la variable `Forest_area`.

3. Gráficos estadísticos

En esta sección se muestra cómo construir algunos de los principales gráficos que se estudian en cursos de estadística básica como el gráfico de barras, de líneas, circular, histogramas ojivas, polígonos de frecuencias y dispersión. Algunos de estos ya se construyeron anteriormente; sin embargo, se presentan a continuación aspectos que permitirán al lector mejorar su presentación.

3.1. Gráfico de Barras

El gráfico de barras se utiliza para presentar frecuencias para una variable discreta o una variable categórica. A continuación se presenta una serie de instrucciones que permiten construir dicho gráfico y a su vez, se aprovecha para introducir nuevos comandos que controlan el aspecto estético por defecto y permiten agregar otros elementos como el título y la fuente.

Para construir el gráfico de barras se utiliza la base de datos `worldbank` y la variable `Income_group` (ver Figura 11).

```
barras <- ggplot(data = worldbank, aes(Income_group)) +
          geom_bar()
print(barras)
```

Observe que se ha guardado la información del gráfico en la variable `barras` a la cual se agregó la geometría requerida. Para poder observar el gráfico, simplemente se usa la función `print` indicando como argumento el nombre del gráfico.

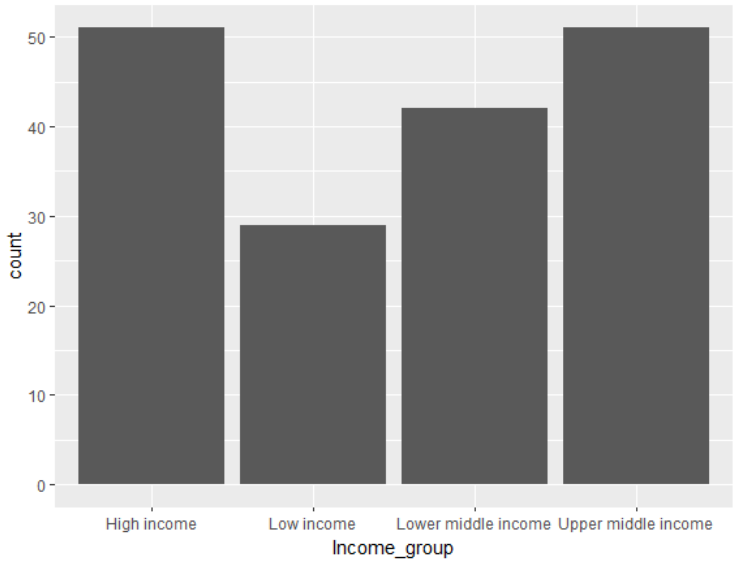

Figura 11: Gráfico de barras para la variable `Income_group`

3.2. Modificando los ejes

Al desplegar el gráfico, `R` coloca por defecto algunas etiquetas a los ejes. Para modificar dichas etiquetas se agregan las capas `xlab` y `ylab`, indicándo entre comillas el nombre que se quiere colocar a cada eje (ver Figura 12).

```
barras +
xlab("Grupos de ingresos") +
ylab("Cantidad")
```

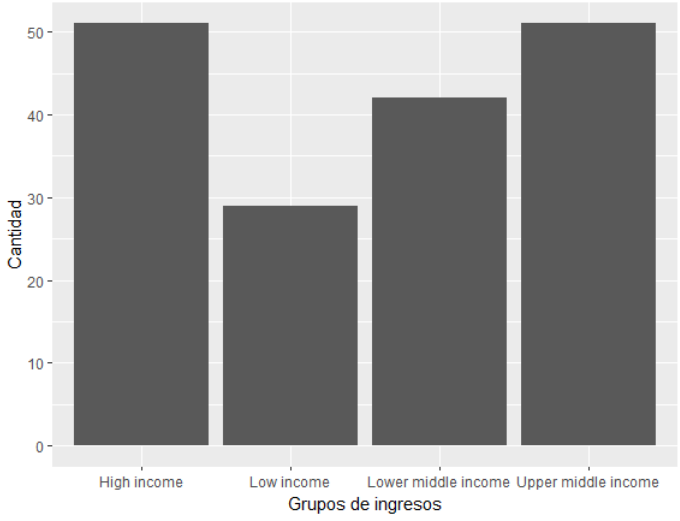

Figura 12: Gráfico de barras para la variable `Income_group`

3.3. Título y fuente

Para colocar un título, un subtítulo (en caso de que sea necesario, por ejemplo una nota introductoria) y la fuente se utiliza el comando `labs` con los argumentos *title*, *subtitle* y *caption*, respectivamente (ver Figura 13).

```
barras +
xlab("Grupos de ingresos") +
ylab("Cantidad") +
labs(title = "Distribución de países según su nivel de ingreso",
   ↪ subtitle = "Año de estudio 2014",
     caption = "Fuente:wolrdbank")
```

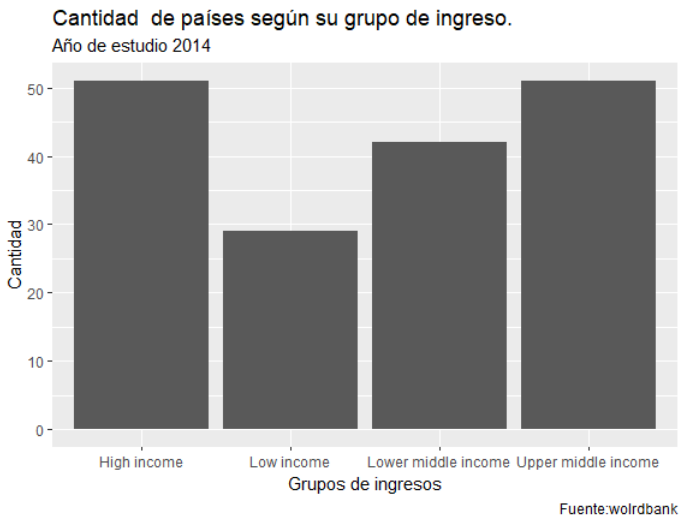

Figura 13: Gráfico de barras para la variable `Income_group`.

Para centrar texto, cambiar el tipo, el color y el tamaño de la fuente se utiliza la capa `theme` y los argumentos *plot.title*, *plot.subtitle* y *plot.caption*, respectivamente, de la siguiente manera (ver Figura 14).

```
barras +
xlab("Grupos de ingresos") +
ylab("Cantidad")+
labs(title = "Distribución de países según su nivel de ingreso",
   ↪ subtitle = "Año de estudio 2014",
     caption = "Fuente:wolrdbank")+
theme(plot.title = element_text(color = "blue", size = 12, face =
   ↪ "bold",hjust = 0.5),
      plot.subtitle = element_text(color = "#F8766D", hjust = 0.5),
      plot.caption = element_text(color = "red", face = "italic", hjust =
         ↪ 0))
```

Entre las opciones que tienen los argumentos de `theme` están:

1. color: acepta el nombre del color en inglés o su código hexadecimal.

2. hjust (posición): puede colocarse a la izquierda (hjust = 0), centrado (hjust = 0.5) o a la derecha (hjust = 1). O bien, puede ajustarse usando valores entre 0 y 1.

3. face (fuente): se utiliza para cambiar el tipo de letra mediante los parámetros *plain*, *italic*, *bold* o *bold.italic*, los cuales deben escribirse entre comillas.

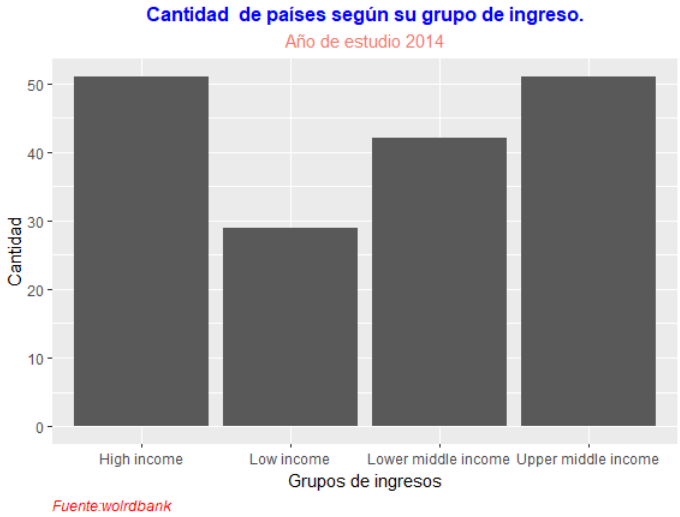

Figura 14: Gráfico de barras con modificaciones de texto para la variable `Income_group`

También puede realizarse la división de texto mediante el uso del comando $\backslash n$. Si en lugar de barras verticales se quieren barras horizontales se utiliza `coord_flip` (ver Figura 15).

```
barras +
xlab("Grupos de ingresos") +
ylab("Cantidad")+
labs(title = "Cantidad de países según su grupo de ingreso.", subtitle =
   ↪ "Año de estudio 2014",
     caption = "Fuente:wolrdbank")+
theme(plot.title = element_text(color = "blue", size = 12, face = "bold",
   ↪ hjust = 0.5),
      plot.caption = element_text(color = "red", face = "italic", hjust =
         ↪ 0)) +
coord_flip()
```

Se ha eliminado el subtítulo para simplificar la exposición.

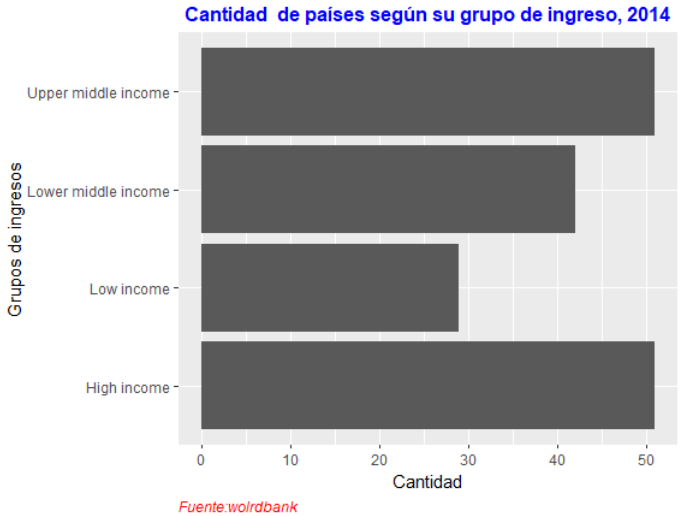

Figura 15: Gráfico de barras horizontales para la variable `Income_group`

Para modificar la estética del gráfico de barras se puede cambiar el tema de colores, el cual es gris por defecto. Esto se puede hacer manualmente o utilizar paletas preestablecidas, las cuales se pueden encontrar por medio del enlace https://r-graph-gallery.com/38-rcolorbrewers-palettes.html. Además, puede modificarse el fondo (backgroud), el contorno del fondo (panel.border) y ordenarse las barras utilizando el paquete `forcats` y las funciones `fct_infreq` (para ordenar los datos) y la función `fct_rev`, la cual invierte el orden en que los datos aparecen (ver Figura 16).

```
library(forcats)

ggplot(worldbank,aes(x=fct_rev(fct_infreq(Income_group)),fill=Income_group))+
geom_bar()+
xlab("Grupos de ingresos")+
ylab("Cantidad")+
labs(title = "Cantidad de países según su grupo de ingreso,
   ↪ 2014",caption = "Fuente:wolrdbank")+
theme( plot.title = element_text(size = 12, face = "bold",hjust = 0.5),
     plot.caption = element_text(face = "italic",hjust = 0),
     panel.background = element_rect(fill = "white"),
     panel.border = element_rect(fill = "transparent",color = 8, size =
        ↪ 2), legend.position = "none")+
coord_flip()+
scale_fill_brewer(palette="Reds")
```

Observe que el argumento `legend.position = "none"` quita las leyendas.

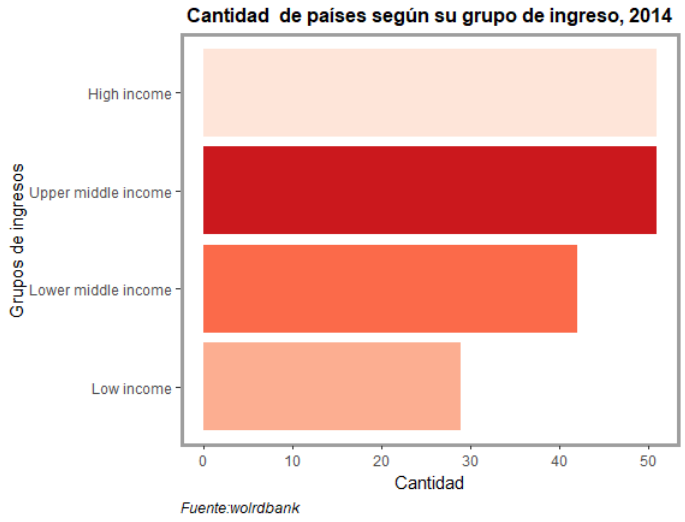

Figura 16: Gráfico de barras completo para la variable `Income_group`.

3.4. Gráfico circular

No existe un comando (`geom`) específico para crear un gráfico circular; sin embargo, para dicha construcción se utiliza el código `geom_bar` y se hace el cambio de coordenadas rectangulares a polares, con el comando `coor_polar`. Para ilustrar dicha construcción se calculan primero las proporciones de la variable de la siguiente manera.

```
library(tidyverse)
ig <- worldbank %>%
      group_by(Income_group) %>%
      summarise(n = n()) %>%
```

```
      mutate(prop = n/sum(n))
```

En el código anterior se ha utilizado la biblioteca `tidyverse`, la cual es un conjunto de paquetes diseñados para ciencias de datos. Luego se crea un dataframe `ig`, filtrando (usando el símbolo %¿ %) la base original `worldbank` como sigue: primero se agrupa por cada valor único de la variable `Income_group`, luego se crea un nuevo dataframe con el conteo de cada valor por medio de `summarise` y, por último, se agrega una nueva columna a la nueva base de datos con `mutate` y esta nueva columna tiene el porcentaje.

```
print(ig)
# A tibble: 4 x 3
```

High income	51	0.295
Low income	29	0.168
Lower middle income	42	0.243
Upper middle income	51	0.295

El lector ya estará acostumbrado a muchos de los comandos que se utilizan, por lo cual se recomienda repasarlos y enfocarse en lo nuevo. Para una construcción básica se tiene el siguiente código (ver Figura 17).

```
ggplot(ig, aes(x = "", y = prop, fill = Income_group)) +
geom_bar(stat = "identity") +
labs(title = "Porcentanjes de países \nsegún su grupo de ingreso, 2014",
   ↪ caption = "Fuente: wolrdbank")+
theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
     plot.caption = element_text(face = "italic", hjust = 0),
     panel.background = element_rect(fill = "white"),
     panel.border = element_rect(fill = "transparent", color = 8, size =
        ↪ 2)) +
xlab(NULL) +
ylab(NULL) +
coord_polar("y") +
scale_fill_brewer(palette="Reds")
```

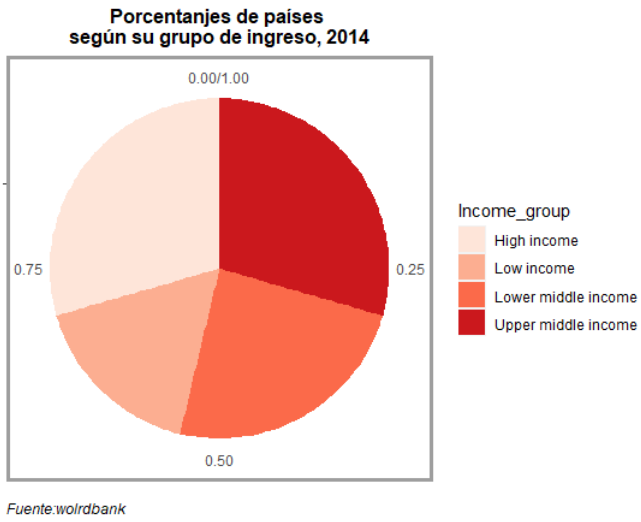

Figura 17: Gráfico circular para la variable `Income_group`

Antes de mejorar la estética del gráfico circular de la Figura 17, es importante mencionar que en el código aparece la expresión `stat = ''identity''` en el `geom_bar`, esto se debe a que `geom_bar`

tiene por defecto `stat = ''count''`, el cual cuenta los casos que aparecen en cada posición del eje x, razón por la que hasta el momento solo se ha colocado una variable con `geom_bar`. En el caso de que se quiera dar los valores de y, se debe indicar esto por medio de `stat = ''identity''`. Además, en la estética (`aes`) se ha escrito `x =''''` para indicar que solo se tendrá una columna y se llenará con los datos de los ingresos por región. Lo anterior también es válido si se quiere un gráfico de barras, para el cual se tiene los valores de x y y.

Ahora, para quitar el fondo gris y las etiquetas que se obtienen por defecto se puede utilizar `theme_void`. Además, para colocar los porcentajes se usa `geom_text` (ver Figura 18).

```
ggplot(ig, aes(x = "", y = prop, fill = Income_group)) +
geom_bar(stat = "identity") +
labs(title = "Porcentanjes de países \nsegún su grupo de ingreso, 2014",
   ↪ caption = "Fuente: wolrdbank") +
theme_void() +
theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
      plot.caption = element_text(face = "italic", hjust = 0),
      panel.background = element_rect(fill = "white"),
      panel.border = element_rect(fill = "transparent", color = 8, size =
         ↪ 2))+
xlab(NULL) +
ylab(NULL) +
coord_polar("y") +
scale_fill_brewer(palette = "Reds") +
geom_text(aes(label = paste0(round(prop*100), "%")), position =
   ↪ position_stack(vjust = 0.5))
```

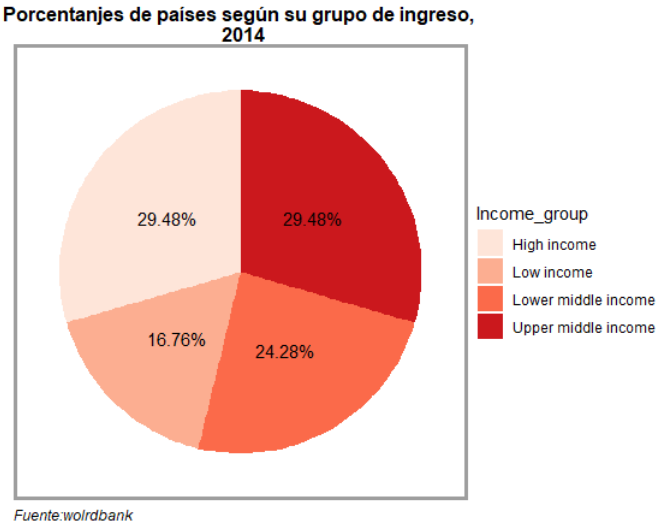

Figura 18: Gráfico circular para la variable `Income_group`.

Observe que el `theme_void` se colocó antes de `theme` y que se ha realizado una pequeña operación para calcular los porcentajes.

3.5. Gráfico de líneas

El lector ya puede imaginar cómo es la estructura del código de un gráfico de líneas, para el cual basta con usar `geom_line`. Se crea un dataframe simple que corresponde al producto interno bruto en millones de euros de Costa Rica y España, desde el año 1999 al año 2021. Estos datos fueron tomados de `https://datosmacro.expansion.com/pib`.

```
Años=c(2021, 2020, 2019, 2018, 2017, 2016, 2015, 2014, 2013, 2012, 2011,
    ↪ 2010, 2009, 2008, 2007, 2006, 2005, 2004, 2003, 2002, 2001, 2000,
    ↪ 1999)

PIB_anual_CR=c(54428,   54451, 57531, 52833, 53589, 53178, 50866, 39145,
    ↪ 38361, 36738, 30726, 28381, 22068, 20945, 19615, 18092, 16106,
    ↪ 14972, 15278, 17551, 17841, 16251, 13366)

PIB_CR_Esp= data.frame( "Años" =Años, "PIB_anual_CR" = PIB_anual_CR)
```

Una vez creado el data_frame, se puede realizar el gráfico de líneas (ver Figura 19) de la siguiente manera.

```
ggplot(data = PIB_CR_Esp, aes(x = Años, y = PIB_anual_CR)) +
geom_line() +
geom_point() +
xlab("Años") +
ylab("PIB Anual") +
labs(title = "PIB anual de Costa Rica, 1999-2021", caption = "Fuente:
    ↪ Datos.macro.com")+
theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
      plot.caption = element_text(face = "italic", hjust = 0),
      panel.background = element_rect(fill = "white"),
      panel.border = element_rect(fill = "transparent", color = 8, size =
          ↪ 2),
      legend.position = "none")
```

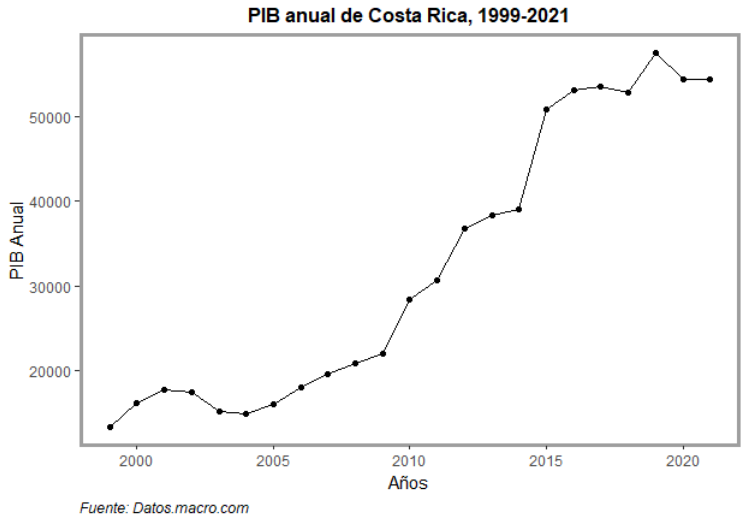

Figura 19: Gráfico de líneas para la variable `PIB Anual`.

También se puede dar énfasis añadiendo la geometría de puntos. En el siguiente código se pueden ver cambios de color, tamaño, tema y etiquetas (ver Figura 20).

```
ggplot(data = PIB_CR_Esp, aes(x = Años, y = PIB_anual_CR)) +
geom_line(color = "red", size = 0.7) +
geom_point(color = "blue") +
xlab("Años") + ylab("PIB Anual") +
labs(title = "PIB anual de Costa Rica, 1999-2021", caption = "Fuente:
    ↪ Datos.macro.com") +
theme_light() +
```

```
theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
      plot.caption = element_text(face = "italic", hjust = 0),
      panel.background = element_rect(fill = "white"),
      panel.border = element_rect(fill = "transparent", color = 8, size =
         ↪ 2),
      legend.position = "none")
```

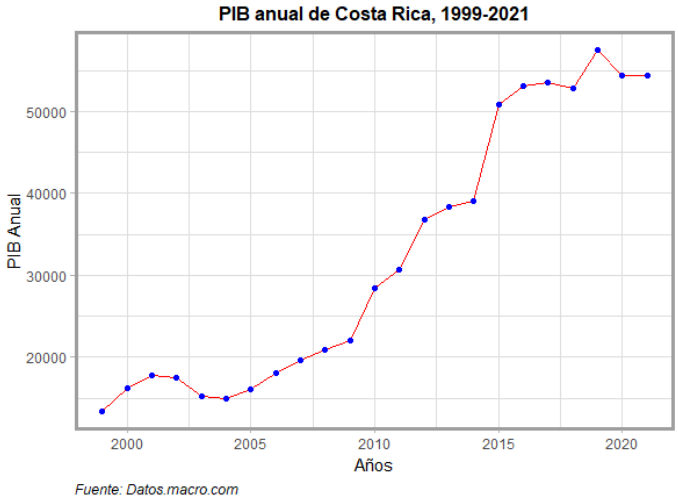

Figura 20: Gráfico de líneas con color para la variable `PIB Anual`.

3.6. Gráfico de cajas y bigotes

Los gráficos de cajas y bigotes son una manera de mostrar la distribución de una variable continua a través de los cuartiles. Este tipo de gráfico puede dar información sobre la simetría, la variabilidad (cuán estrechos están los datos agrupados), los valores atípicos y la comparación entre magnitudes. Para crear un gráfico de cajas y bigotes (ver Figura 21) se utiliza la geometría `geom_boxplot` tal y como se muestra a continuación.

```
ggplot(worldbank, aes(Precipitation, Climate_region)) +
geom_boxplot(color = "#5499c7", fill = "#D6eaf8", outlier.color =
   ↪ "#f5b7b1")
```

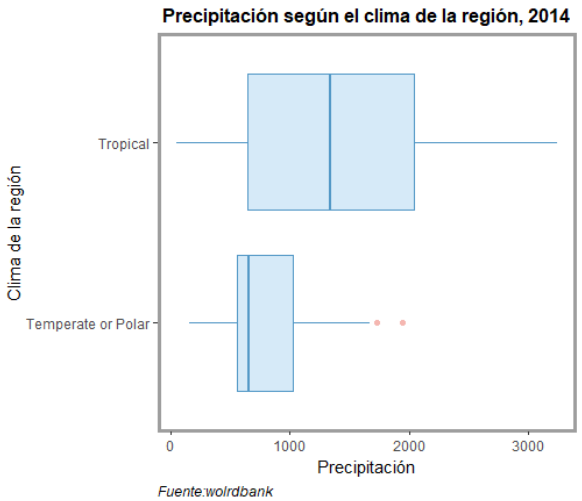

Figura 21: Gráfico de cajas para las variables `Precipitation` y `Climate_region`

Se puede utilizar `geom_jitter` para agregar una pequeña variación aleatoria a la posición de los puntos (ancho `width` y alto `height`) con el fin de darse una idea de su distribución (ver Figura 22).

```
ggplot(worldbank, aes(Precipitation, Climate_region)) +
geom_boxplot(color = "#5499c7", fill = "#D6eaf8", outlier.shape = NA) +
geom_jitter(width = 0.1, color = "#5499c7")
```

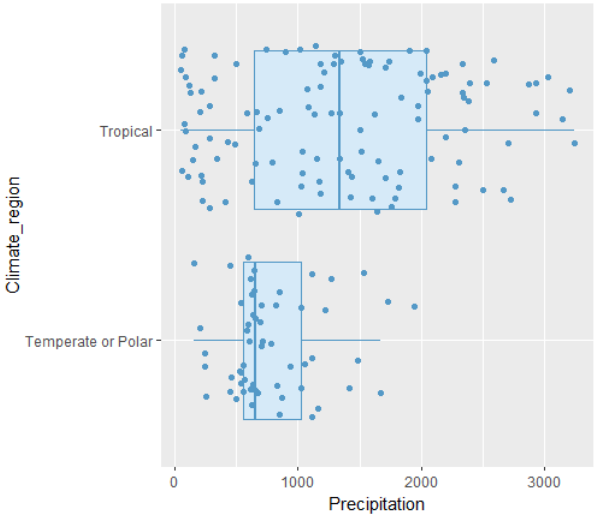

Figura 22: Gráfico de cajas con distribución de puntos para las variables `Precipitation` y `Climate_region`

3.7. Histogramas, polígonos de frecuencia y ojivas

Con el histograma y el polígono de frecuencias se puede visualizar la distribución de una variable continua contando el número de observaciones que se encuentran en las clases previamente determinadas en el eje x (`bins`). La diferencia radica en que el histograma utiliza barras y el polígono de frecuencia utiliza líneas.

Tal y como se expuso en la Sección 2.2, para obtener un histograma (ver Figura 23) puede escribirse la siguiente instrucción:

```
ggplot(data = worldbank, aes(x = Forest_area)) +
geom_histogram(bins = 10, color = "darkblue", fill = "lightblue")
```

Si se quiere modificar la estética (ver Figura 24) puede considerarse la misma estructura utilizada en algunos de los apartados anteriores.

```
ggplot(data = worldbank, aes(x=Forest_area)) +
  geom_histogram(bins = 10, color = "darkblue", fill = "lightblue") +
  xlab("Area de bosque") +
  ylab("Cantidad") +
  labs(title = "Area de bosque, 2014", caption = "Fuente:wolrdbank") +
  theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
        plot.caption = element_text(face = "italic",hjust = 0),
        panel.background = element_rect(fill = "white"),
        panel.border = element_blank(),
        legend.position = "none",
        axis.line = element_line(color = 'darkblue'))
```

Observe que en el código anterior se ha quitado el borde y luego se le colocado solo el eje x y el eje y.

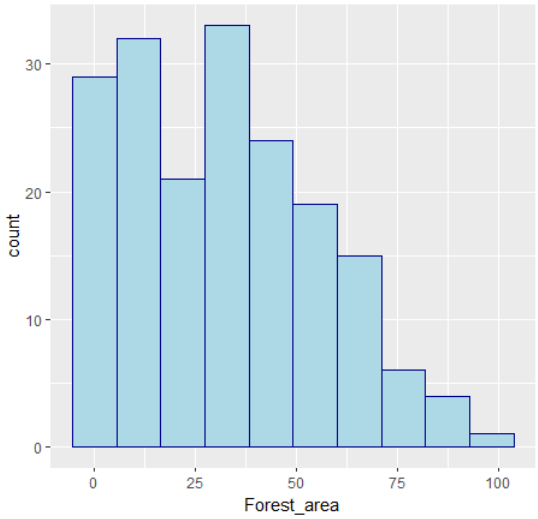

Figura 23: Histograma para la variable `Forest_area`

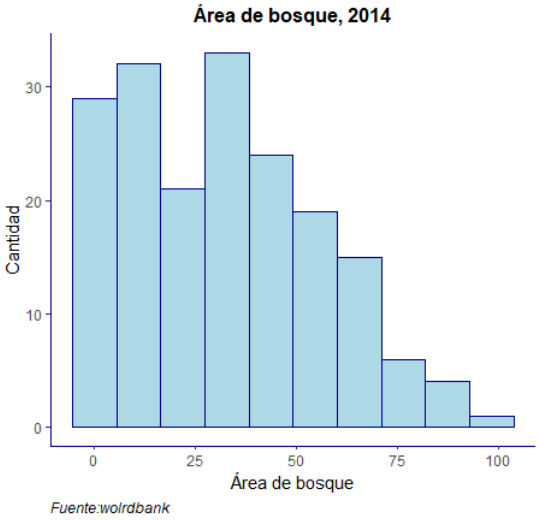

Figura 24: Histograma con cambios de estética para la variable `Forest_area`

Se puede extraer la información del histograma utilizando el comando `ggplot_build`.

```
infoHist <- ggplot_build(H)
$data
$data[[1]]
   y count x xmin xmax density ncount ndensity flipped_aes
1 29 29 0.00000 -5.461437 5.461437 0.0144292321 0.87878788 0.87878788
   ↪ FALSE
2 32 32 10.92287 5.461437 16.384312 0.0159219113 0.96969697 0.96969697
   ↪ FALSE
3 21 21 21.84575 16.384312 27.307187 0.0104487543 0.63636364 0.63636364
   ↪ FALSE
4 33 33 32.76862 27.307187 38.230062 0.0164194710 1.00000000 1.00000000
   ↪ FALSE
5 24 24 43.69150 38.230062 49.152937 0.0119414335 0.72727273 0.72727273
   ↪ FALSE
6 19 19 54.61437 49.152937 60.075811 0.0094536348 0.57575758 0.57575758
   ↪ FALSE
```

```
7 15 15 65.53725 60.075811 70.998686 0.0074633959 0.45454545 0.45454545
   ↪ FALSE
8 6 6 76.46012 70.998686 81.921561 0.0029853584 0.18181818 0.18181818
   ↪ FALSE
9 4 4 87.38300 81.921561 92.844436 0.0019902389 0.12121212 0.12121212
   ↪ FALSE
10 1 1 98.30587 92.844436 103.767310 0.0004975597 0.03030303 0.03030303
   ↪ FALSE
  PANEL group ymin ymax colour fill linewidth linetype alpha
1 1 -1 0 29 darkblue lightblue 0.5 1 NA
2 1 -1 0 32 darkblue lightblue 0.5 1 NA
3 1 -1 0 21 darkblue lightblue 0.5 1 NA
4 1 -1 0 33 darkblue lightblue 0.5 1 NA
5 1 -1 0 24 darkblue lightblue 0.5 1 NA
6 1 -1 0 19 darkblue lightblue 0.5 1 NA
7 1 -1 0 15 darkblue lightblue 0.5 1 NA
8 1 -1 0 6 darkblue lightblue 0.5 1 NA
9 1 -1 0 4 darkblue lightblue 0.5 1 NA
10 1 -1 0 1 darkblue lightblue 0.5 1 NA
```

Esta información puede ser útil para análisis o para ser utilizada en otros gráficos. Por ejemplo, se puede generar un polígono de frecuencias (ver Figura 25) de la siguiente manera.

```
ggplot(data = worldbank, aes(x = Forest_area)) +
  geom_histogram(bins = 10, color = "darkblue", fill = "lightblue") +
  xlab("Area de bosque") +
  ylab("Cantidad") +
  labs(title = "Area de bosque, 2014", caption = "Fuente: wolrdbank")+
  theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
        plot.caption = element_text(face = "italic", hjust = 0),
        panel.background = element_rect(fill = "white"),
        panel.border = element_blank(),
        legend.position = "none",
        axis.line = element_line(color = 'darkblue'))
```

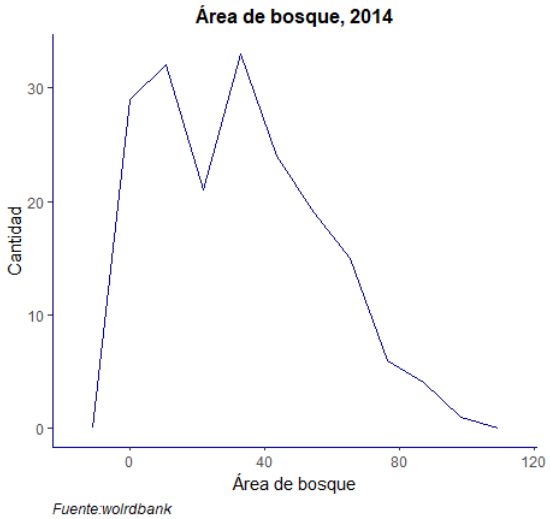

Figura 25: Polígono de frecuencias para la variable `Forest_area`

Por otro lado, se puede utilizar `infoHist` en la geometría `geom_point` para colocar los puntos de interés (ver Figura 26). Es decir, de `infoHist$data[[1]]` se selecciona x y y.

```
ggplot() +
  geom_freqpoly(data= worldbank,aes(x=Forest_area), bins = 10,color=
     ↪ "darkblue") +
  geom_point(aes(infoHist$data[[1]]$x,infoHist$data[[1]]$y),color="red") +
  xlab("Area de bosque") +
  ylab("Cantidad") +
  labs(title = "Area de bosque, 2014", caption = "Fuente: wolrdbank")+
  theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
        plot.caption = element_text(face = "italic",hjust = 0),
        panel.background = element_rect(fill = "white"),
        panel.border = element_blank(),
        legend.position = "none",
        axis.line = element_line(color = 'darkblue'))
```

Observe que se han utilizado dos bases de datos, de las cuales se especifican en cada geometría.

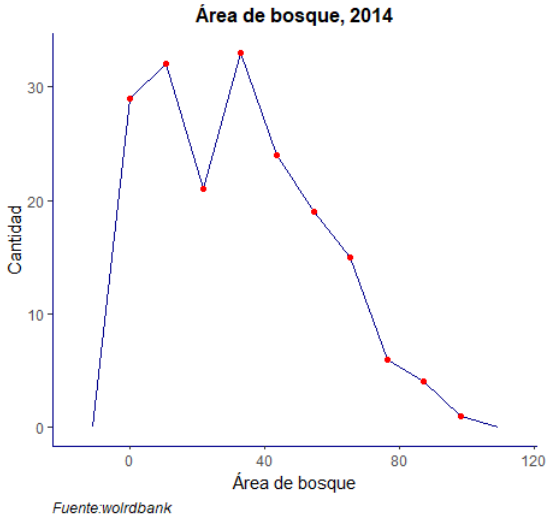

Figura 26: Polígono de frecuencias para la variable `Forest_area`

Si se consideran las frecuencias acumuladas y se utiliza el polígono de frecuencias para obtener su gráfica, entonces se tendrá la gráfica conocida como `ojiva`, también llamada polígono de frecuencias acumuladas (nombre más sugestivo) (ver Figura 27). Para acumular el conteo que realiza un histograma o un polígono de frecuencias se utiliza la función `cumsum`.

```
ggplot() +
  geom_freqpoly(data = worldbank, aes(x = Forest_area, y =
     ↪ cumsum(..count..)), bins = 10,
                                              color = "darkblue") +
  geom_point(aes(infoHist$data[[1]]$x, cumsum(infoHist$data[[1]]$y)),
     ↪ color = "red") +
  xlab("Area de bosque") +
  ylab("Cantidad") +
  labs(title = "Area acumulada de bosque, 2014", caption = "Fuente:
     ↪ wolrdbank")+
  theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
        plot.caption = element_text(face = "italic",hjust = 0),
        panel.background = element_rect(fill = "white"),
        panel.border = element_blank(),
        legend.position = "none",
        axis.line = element_line(color = 'darkblue'))
```

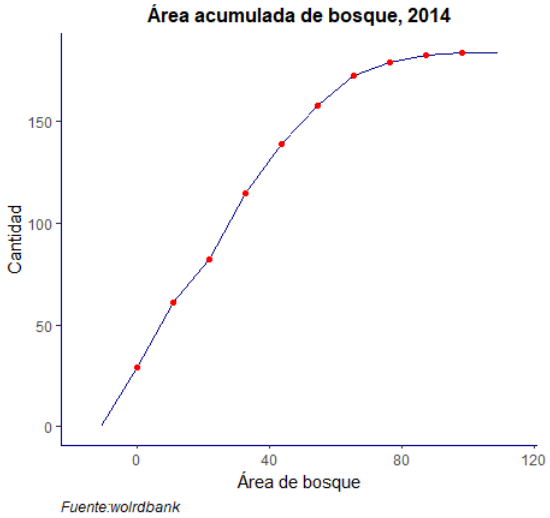

Figura 27: Polígono de frecuencias acumuladas para la variable `Forest_area`

4. Conclusiones

Actualmente la visualización de datos es una necesidad en muchas áreas, en este sentido el manejo adecuado de un software, por ejemplo `R`, es indispensable para presentar información estadística.

Se considera la necesidad de que este tipo de software sea implementado en cursos de estadística de educación superior, ya que es necesario que el estudiantado desarrolle las habilidades de graficación y otras a través del uso y manipulación del software, siendo el paquete `ggplot2` una herramienta muy valiosa, pues permite realizar gráficos de una manera sencilla y con una alta calidad.

Además, la formación en el manejo adecuado del lenguaje `R` le permitirá al estudiantado desenvolverse en diferentes áreas en las cuales deba trabajar con datos estadísticos y visualización de información, por ejemplo, la minería de datos o data mining, la cual está en auge.

En este documento presentaron algunos de los gráficos básicos que se estudian en un curso introductorio de estadística descriptiva y algunos de los códigos que ofrece el paquete `ggplot2` de `R` para su representación. El objetivo principal es que lo expuesto en este documento pueda ser utilizado en el estudio de la presentación de información por medio de gráficas, contenido que se estudia en cursos de estadística básica. El fin de esto es que tanto profesores como estudiantes puedan consultar de una manera más sintetizada esta información.

Es importante acotar que `ggplot2` ofrece gran variedad de códigos, los cuales pueden consultarse en la documentación de este paquete y en diferentes fuentes de internet.

5. Bibliografía

[1] Dobrow, R. P. (2016). Introduction to stochastic processes with R. John Wiley & Sons.

[2] Kabacoff, R. (2022). R in Action: Data Analysis and Graphics with R and Tidyverse. Simon and Schuster.

[3] Monahan, J. F. (2011). Numerical methods of statistics. Cambridge University Press.

[4] Sevilla Moróder, J. (2005). Gramática de las gráficas: pistas para mejorar las representaciones de datos. Universidad Pública de Navarra/Nafarroako Unibertsitate Publikoa.

[5] Teutonico, D. (2015). ggplot2 Essentials. Packt Publishing Ltd.

[6] Wickham, H. (2016). Data analysis. ggplot2: elegant graphics for data analysis, 189-201.

[7] Wickham, H., & Grolemund, G. (2016). R for data science: import, tidy, transform, visualize, and model data. .ºReilly Media, Inc.".

[8] Wilkinson, L. (2012). The grammar of graphics (pp. 375-414). Springer Berlin Heidelberg.