Análisis univariante

PID_00268326

Jordi Mas Elias

Tiempo mínimo de dedicación recomendado: 3 horas

Universitat Oberta de Catalunya

Jordi Mas Elias

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por el profesor: Jordi Mas Elias (2019)

Primera edición: septiembre 2019

Avda. Tibidabo, 39-43, 08035 Barcelona
Realización editorial: FUOC

Índice

Introducción

Tal como su nombre indica, el análisis univariante significa el análisis de una sola variable. La naturaleza de la variable que queremos estudiar determinará en buena parte los instrumentos que utilizaremos para el análisis univariante: si es una variable categórica haremos un tipo de tratamiento, mientras que si es una variable numérica haremos otro. Por este motivo será muy importante que tengamos identificada la variable en R con el vector más apropiado.

En muchas ocasiones, el análisis univariante requiere utilizar más de una variable. Por ejemplo, podemos querer observar una variable teniendo en cuenta los valores que toman otras variables del marco de datos. Esto puede llevar a confusión con el trabajo del análisis bivariante, que se ocupa de estudiar la relación entre dos variables. Por eso distinguiremos dos tareas cruciales del análisis de datos: la descripción y la explicación (King y otros, 1994). Este módulo se ocupa de la descripción. Por lo tanto, la utilización de dos variables se orienta a describir una variable y a hacer comparaciones de sus valores entre los diversos subgrupos de otra variable (Babbie, 2013). En ningún momento este proceso se orienta a sugerir si las dos variables en cuestión están o no asociadas entre ellas, que es trabajo del análisis bivariante.

En la primera parte del módulo veremos diferentes representaciones gráficas que podemos utilizar en el análisis univariante. El principal objetivo de una buena visualización es representar de manera clara cómo se distribuyen los valores en una variable. En gran parte, la manera como representamos estos valores estará determinada por si la variable es de tipo categórico o numérico. En la segunda parte del módulo pasaremos de la visualización a la cuantificación. Es decir, buscaremos cómo resumir la distribución de los datos en una variable de manera numérica, mediante uno o pocos números. Miraremos medidas de frecuencia, de centralidad, de dispersión y de localización.

1. Representaciones gráficas

En esta sección combinaremos funciones de los paquetes *dplyr* y *ggplot2* para representar gráficamente la distribución de una variable del marco de datos *gapminder*. Para poder hacer estas representaciones, tendremos que estar suficientemente familiarizados con la gramática de *ggplot2*, puesto que en las páginas siguientes aplicaremos diversas de las geometrías que ofrece el paquete. Según el tipo de variable que queramos representar y la manera como la queramos representar, utilizaremos como principales formas de representación gráfica:

- el histograma,
- el diagrama de líneas,
- el diagrama de barras,
- el diagrama de cajas y el
- diagrama de dispersión.

1.1. Histograma

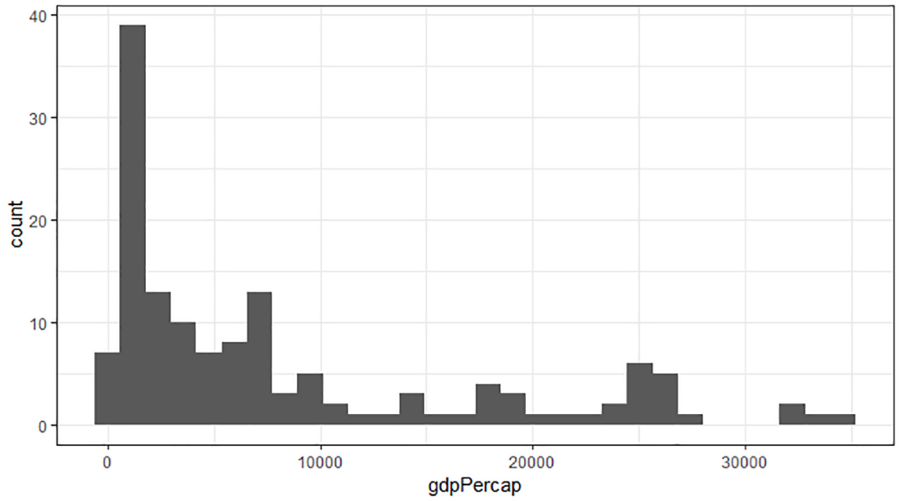
El histograma nos permite visualizar con varias barras verticales la distribución de los valores de una variable numérica. Cada una de las barras representa un intervalo de valores de la variable y la altura de la barra corresponde al número de casos de cada intervalo. Hay que tener en cuenta, pues, que las barras del histograma no representan los valores numéricos originales de la variable, sino una función estadística que separa los datos en diferentes intervalos y las apila por columnas. Con R, representaremos esta figura con la función `geom_histogram()`.

Pérdida de información con el histograma

El histograma nos permite obtener una imagen muy nítida de la forma que tiene una distribución numérica. La contrapartida, sin embargo, es que perdemos información, puesto que los intervalos que creamos homogeneizan el valor de los datos que contienen. Una manera visualmente menos nítida, pero que permite conservar información, es el `geom_dotplot()`, donde cada punto representa una observación. Esta visualización es útil con un número de casos bajo.

En la figura 1 hemos creado un histograma para observar cómo están distribuidos los valores de la variable *gdpPercap* en 1992. Cada barra horizontal representa un intervalo de valores de la variable que indicaremos en el eje de las *x* dentro de los estéticos de *ggplot*. No hace falta que indiquemos el eje vertical de las *y*, puesto que siempre veremos el recuento de casos de cada columna (llamaremos frecuencias a la cantidad de casos). Por defecto, el histograma corta la variable en 30 intervalos de la misma anchura y representa la cantidad de valores que hay en cada intervalo con la estatura de las columnas. En nuestro caso, cada barra representa un intervalo de unos 1.300 dólares. Así, en el primer intervalo, encontraremos el número de países situados entre el intervalo de 0 y 1.300 dólares; en el segundo, los situados entre 1.300 y 2.600 dólares; en el tercero, entre 2.600 y 3.900, y así sucesivamente.

Figura 1. Histograma del PIB per cápita




```
gapminder %>%
  filter(year == 1992) %>%
  ggplot(aes(x = gdpPercap)) +
  geom_histogram() +
  theme_bw()
```

En el histograma que hemos creado vemos cómo el intervalo con más frecuencias es el segundo, probablemente situado entre unos 1.300 y 2.600 dólares per cápita al mes (llamaremos moda a la barra más alta, como veremos más adelante). Si miramos en el eje vertical el recuento de frecuencias, vemos cómo en este intervalo se sitúan cerca de 40 países. Alrededor de los 30.000 dólares, en cambio, no hay ningún caso.[1]

(1)Porcentajes en el eje de las *y*: si en lugar de visualizar el número de frecuencias en el eje vertical preferís visualizar los porcentajes (también denominado densidad), podéis introducir el siguiente estético dentro de la geometría: `geom_histogram(aes(y = ..density..))`.

Cuando creáis un histograma la consola os indica que ha creado 30 intervalos de datos como medida por defecto.[2] Podemos cambiar la anchura de los intervalos que representa cada columna de dos maneras:

(2)Sería lo mismo que introducir el argumento `bins = 30` dentro de la función del histograma. En la consola os tendría que especificar el retorno: *`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.*

1) La primera es indicando la cantidad de intervalos. Si, en lugar de los 30 por defecto queremos observar solo 20, en la función añadiremos `geom_histogram(bins = 20)`. Podéis hacer varias pruebas introduciendo números diferentes para ver cómo cambia la visualización.

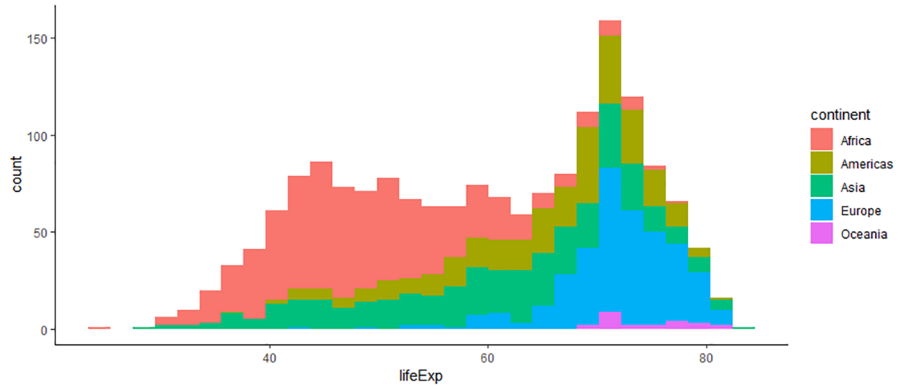
2) La segunda manera de cambiar los intervalos es indicando la anchura. Si queremos, por ejemplo, que cada uno de los intervalos nos represente 3.000 dólares per cápita, usaremos el argumento *binwidth* de la manera siguiente: `geom_histogram(binwidth = 3000)`. También podéis hacer diferentes pruebas para ver cómo cambia la visualización, y también cambiar el color de las barras indicando dentro de la geometría el color con el atributo *fill*, por ejemplo, `geom_histogram(fill = "dark blue")`.

Cómo se calcula el intervalo de cada columna

Para calcular la anchura de las columnas de un histograma R mira el rango de los datos. Es decir, el valor máximo (en este caso unos 40.000 dólares) y el valor mínimo (que se sitúa cerca de 0) y divide el rango en 30 intervalos de la misma longitud. Por ejemplo, el intervalo de todos los valores de la variable PIB per cápita es: `diff(range(gapminder$gdpPercap)) / 30`. Encontraréis más información sobre este proceso si miráis la ficha de los histogramas. Solo tenéis que entrar `?geom_histogram` en la consola y mirar la descripción de los argumentos.

El color también nos puede servir para introducir una nueva variable en el gráfico. Aunque el histograma se utilice para ver la distribución de una variable numérica, también podemos ver a la vez la distribución de estos valores en varias categorías. Observad la figura 2, donde visualizamos la distribución de la esperanza de vida en los diferentes continentes. En el código, hemos llenado las barras del histograma utilizando el argumento `fill = continent` como estético para indicar la variable categórica.

Figura 2. Histograma con variable categórica




```
gapminder %>%
  ggplot(aes(x = lifeExp,
fill = continent)) +
  geom_histogram(position =
"stack") + theme_classic()
```

En el código, podéis observar cómo hemos añadido un argumento nuevo dentro de la geometría: `position = "stack"`. Si sacamos este argumento veréis que no cambia nada en la visualización. Esto es porque, por defecto, la posición de las columnas es *stack* (apilada). Esto quiere decir que, en cada columna, apilará las categorías una sobre la otra. Podría, alternativamente, poner las categorías una junto a la otra o superponerlas.

En general, `"stack"` será la mejor posición que tendremos para visualizar una variable categórica dentro de un histograma. Alternativamente, podemos probar a visualizar las columnas en otras posiciones. Intentad, por ejemplo, cambiar la posición del anterior histograma a `"fill"`. Esta opción llena la barra hasta la parte superior del gráfico y es útil para ver los porcentajes de cada categoría en cada intervalo. Con la posición `"dodge"` veremos de lado las ba-

[3]En los histogramas, la posición `"dodge"` es solo recomendable cuando tenemos pocos intervalos. Probad, por ejemplo, a combinar *dodge* con un número reducido de intervalos con la función `geom_histogram(position = "dodge", bins = 10)`.

rras separadas por categorías[3], mientras que con "`identity`" las columnas no se apilan sino que conservan su identidad real. Esta última, sin embargo, se puede observar mejor con un diagrama de densidad.

Diagrama de densidad: la alternativa al histograma

Otra opción de análisis univariante para variables numéricas es el diagrama de densidad. Este diagrama resume los datos dibujando un área que nos permite tener una visión general de la forma que tiene la distribución. El diagrama de densidad es útil cuando queremos añadir información adicional al gráfico (porque nos lo deja muy limpio) y también para superponer varias áreas. Probad los códigos siguientes:

```
gapminder %>% ggplot(aes(x = lifeExp)) + geom_density(fill = "orange", bw = 1)
gapminder %>% ggplot(aes(x = lifeExp, fill = continent)) + geom_density(alpha = 0.4)
gapminder %>% filter(year %in% c(1962, 1977, 1992, 2007)) %>% ggplot(aes(x = lifeExp,
fill = factor(year))) + geom_density(alpha = 0.4, bw = 5)
gapminder %>% filter(year %in% c(1962, 1977, 1992, 2007)) %>% ggplot(aes(x = lifeExp,
fill = factor(year))) + geom_density(bw = 5) + facet_wrap(~ year)
```

1) En el primer gráfico hemos indicado la anchura (*bw*, que es una abreviación de *bin-width*).

2) En el segundo gráfico hemos representado varios continentes y hemos dado una transparencia de 0,4 a los diagramas de densidad. Con los mismos datos, podéis intentar hacer un histograma especificando dentro de la geometría `position = identity`.

3) En el tercer gráfico hemos visualizado el PIB per cápita en cuatro años diferentes. Por eso hemos tenido que indicar a R que trate como categórica la variable *año* y la convierta en factor.

4) El último gráfico es parecido al tercero, pero en este caso hemos reducido la anchura de los intervalos y hemos separado los datos por *facets*. Probad de hacer este mismo gráfico con un histograma.

Fijaos en que en el eje vertical siempre vemos la densidad (porcentaje) y que el área que cubre cada distribución está por defecto normalizada, de forma que todas las áreas cubren exactamente la misma superficie.

Hemos visto cómo hay varias maneras de representar un histograma, por ejemplo, modificando la cantidad de intervalos del eje horizontal o añadiendo una variable categórica y cambiando la posición de las barras. No hay una regla concreta para decidir cuál es la mejor visualización, sino que la decisión estará determinada por lo que queramos comunicar, la cantidad de datos disponibles y la distribución de estos. Lo mejor que podemos hacer es probar varias variantes y utilizar el sentido común para decidir cuál es la mejor visualización.

1.2. Diagrama de líneas

El diagrama de líneas se utiliza normalmente para visualizar la tendencia de una variable numérica en el tiempo. En el eje horizontal de las *x*, situaríamos la variable temporal (en el caso de *gapminder* es el año, pero podrían ser meses, días...), mientras que en el eje vertical de las *y*, situaríamos la variable numérica que queremos explicar. Con R, representaremos esta figura geométrica con la función `geom_line()`. En el código siguiente hemos hecho un sumario de la media de la esperanza de vida mundial agrupada por año. A continuación hemos pedido un diagrama de líneas y hemos especificado varios atributos a la geometría:

- el color,
- el tamaño,
- el tipo de línea.

Si imprimimos el código, observaremos que ha habido un aumento importante de la media de la esperanza de vida por país a lo largo del tiempo, puesto que ha pasado de estar por debajo de los 50 años en 1952 a superar los 65 años poco antes del año 2000.

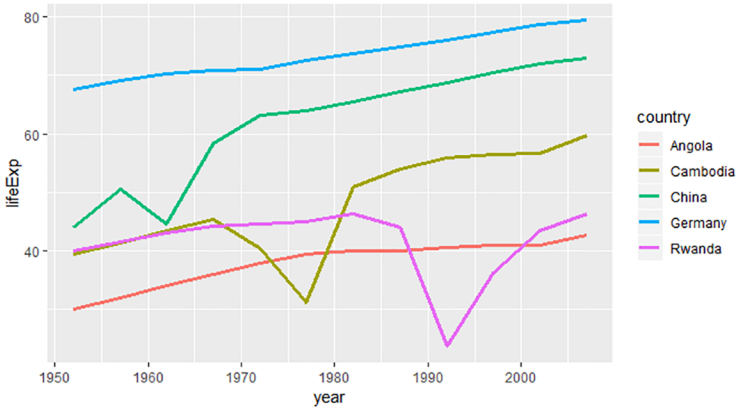
Los tipos de vectores *date*

Los años se almacenan como vectores enteros, pero si necesitamos operar con franjas más cortas de tiempo nos puede interesar convertir el vector *date*, un tipo de vector especial que almacena valores numéricos pero los visualiza como si fuera una fecha. El paquete *lubridate* (https://lubridate.tidyverse.org/) os ayudará a hacer la conversión. Para hacer una prueba, pedidle a R la hora que es con `Sys.time()` y a continuación probad a hacer dos operaciones. Primero, mirad la clase del vector resultante. Segundo, convertid el vector resultante en numérico con `as.numeric(Sys.time())`. Ejecutad esta operación varias veces y descubriréis cómo R almacena realmente este vector.

```
gapminder %>%
  group_by(year) %>%
  summarize(mean_lifeExp = mean(lifeExp)) %>%
  ggplot(aes(x = year, y = mean_lifeExp)) +
  geom_line(col = "dark green", size = 1.2, lty = 5)
```

Hay centenares de combinaciones que podemos realizar con los atributos del diagrama línea. Probad a cambiarlos modificando el color, el tamaño o el tipo de línea dentro de la función de la geometría. También podemos hacer que cada línea represente una variable categórica, mediante el estético de color. En la figura 3 hemos filtrado cinco países de *gapminder* y hemos pedido un diagrama de líneas para ver la evolución temporal de su esperanza de vida. Hemos situado el año en el eje horizontal, la esperanza de vida en el eje vertical y hemos incluido un estético adicional, el color, que representa la variable *country*. Cada color representará la evolución de la esperanza de vida en un país diferente.

Figura 3. Diagrama de línea con variable categórica




```
gapminder %>%
  filter(country %in% c("Cambodia", "Germany", "China", "Angola", "Rwanda")) %>%
  group_by(year, country) %>%
  summarize(mean_lifeExp = mean(lifeExp)) %>%
  ggplot(aes(x = year, y = mean_lifeExp, col = country)) +
  geom_line(size = 1.2)
```

En esta figura podemos observar más detalles que con la que hemos generado con el código anterior. Fijémonos en que, en el caso de China, Camboya y Ruanda, observaremos reducciones repentinas de la esperanza de vida, que marcan periodos históricos relevantes en estos países. El mejor estético para diferenciar diferentes líneas es el color, aunque también podemos utilizar el tipo de línea (*lty*) o una combinación de color y tipo de línea.

El estético *group* para agrupar sin estéticos

También podemos utilizar *group* en los estéticos para construir líneas que representen diferentes valores de una variable categórica. En este caso, sin embargo, no nos mostrará ninguna diferencia visual entre las líneas.

1.3. Diagrama de barras

El diagrama de barras no es un histograma. Para diferenciarlos, diremos que el diagrama de barras toma como punto de partida una variable categórica, mientras que el histograma siempre nos mostrará una variable numérica. Cada una de las columnas del diagrama de barras representa los diferentes valores que toma la variable categórica y la altura de las barras representa unos determinados valores que queremos comparar entre categorías. Hay dos maneras de representar un diagrama de barras.

Diferencia entre diagrama de barras e histograma

Visualmente, fijaos en que hay dos diferencias muy importantes entre el histograma y el diagrama de barras. En primer lugar, las etiquetas del eje horizontal están situadas dentro de las barras del diagrama de barras porque representan la categoría; en cambio, en el histograma están situadas entre las barras porque representan la división entre intervalos. En segundo lugar, en el histograma no hay separación entre barras porque representan la continuidad de la variable. En cambio, como el diagrama de barras representa variables categóricas (no continuas), hay separación entre las barras.

1) La primera es que la altura de las barras represente el número de frecuencias de cada categoría. En este caso, utilizaremos la función geométrica `geom_bar()` para representar el diagrama. Fijaos en el código siguiente. En este caso, solo necesitamos indicar la variable categórica que representaremos en el eje de las *x*. En el eje vertical nos generará automáticamente un recuento de frecuencias en cada categoría.

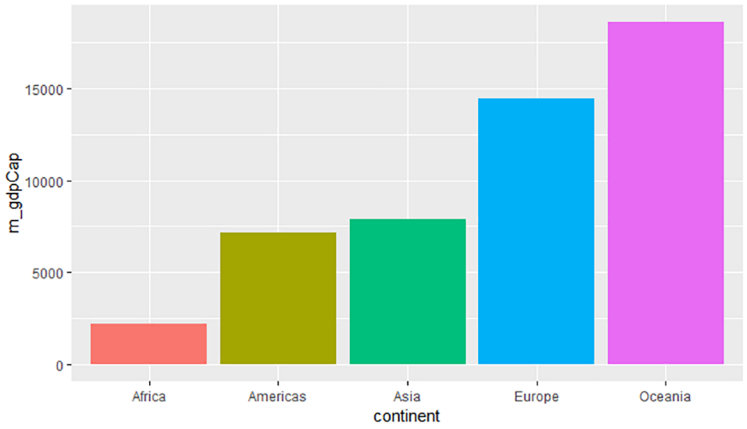
```
ggplot(gapminder, aes(x = continent)) + geom_bar()
```

2) La segunda manera de representar un diagrama de barras es situando una variable categórica en el eje de las *x* pero indicando que la altura de las barras nos muestre el sumario de otra variable, normalmente numérica, que indicaremos en el eje de las *y*. En este caso, utilizaremos la función geométrica `geom_col()` para representar el diagrama. Para poder conseguir esta visualización las funciones de *dplyr* `group_by()` y `summarize()` nos serán de gran ayuda, como vemos en el código asociado a la figura 4. En primer lugar, hemos agrupado los datos por continente y hemos pedido un sumario de la media del PIB per cápita en cada continente. De este modo, en el eje horizontal situaremos la variable categórica continente y en el eje vertical el sumario de la variable numérica agrupada por continente. Para visualizar con más claridad las diferencias entre columnas también hemos introducido el estético *fill* para que nos muestre un color por cada continente. Para evitar redundancias, hemos eliminado la leyenda.

Atributos que diferencian `geom_bar` y `geom_col`.

La única diferencia entre estas dos maneras de representar diagramas de barras es la función estadística que tienen asociada por defecto. La función `geom_bar()` hace un recuento de frecuencias de cada categoría de *x*, puesto que tiene por defecto el atributo `stat = "count"`. La función `geom_col()` nos muestra la identidad de *y*, puesto que tiene por defecto el atributo `stat = "identity"`.

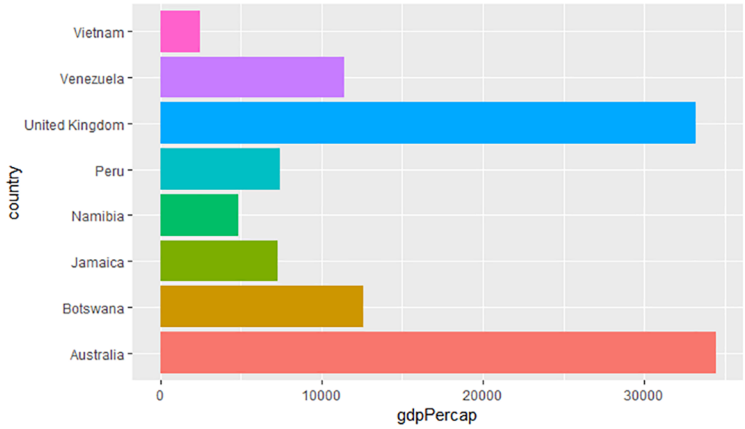
Figura 4. Diagrama de barras




```
gapminder %>%
  group_by(continent) %>%
  summarize(m_gdpCap = mean(gdpPercap)) %>%
  ggplot(aes(x = continent, y = m_gdpCap, fill = continent)) +
  geom_col(show.legend = FALSE)
```

Los diagramas de barras no tienen que ser necesariamente verticales. Nos podemos encontrar con que queramos representar un número relativamente elevado de categorías y no podamos distinguir bien sus nombres en el eje horizontal. También nos podemos encontrar con que tenemos pocas categorías pero los nombres de cada categoría son muy largos. En estos casos, podemos rotar los ejes del gráfico con la opción `coord_flip()`, que intercambia de posición *x* e *y*. En la figura 5 hemos rotado los ejes del diagrama de barras porque teníamos ocho categorías para mostrar.

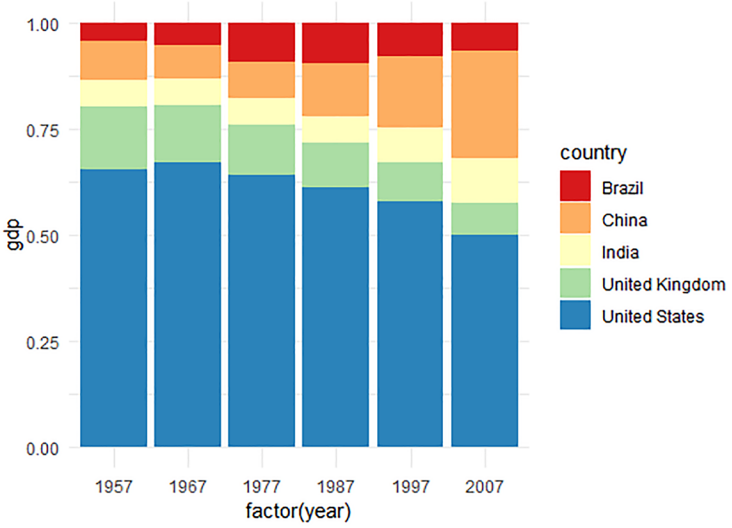
Figura 5. Diagrama de barras horizontales




```
gapminder %>%
  filter(year == 2007, country %in% c("Australia", "Botswana", "Namibia", "Jamaica",
  "Peru", "United Kingdom", "Venezuela", "Vietnam")) %>%
  arrange(desc(country)) %>%
  ggplot(aes(x = country, y = gdpPercap, fill = country)) +
  geom_col(show.legend = FALSE) +
  coord_flip()
```

En los ejemplos que hemos visto hasta ahora, la altura de las barras del diagrama han representado números absolutos. Es decir, las barras nos indicaban un determinado valor de una variable. Otra herramienta que tenemos es la posibilidad de hacer que las columnas nos representen proporciones en lugar de valores. Así, podremos ver la dimensión relativa de cada valor dentro de una categoría concreta. En la figura 6 hemos querido observar la evolución, en términos relativos, del PIB entre grandes potencias mundiales: Brasil, China, India, Reino Unido y Estados Unidos. En este caso, no nos interesa tanto cómo ha crecido el PIB en números absolutos sino cómo ha variado la relación entre ellos. Esta visualización nos la permite el argumento `position = "fill"` dentro de la geometría.

Figura 6. Diagrama de barras con proporciones




```
gapminder %>%
  filter(country %in% c("Brazil", "China", "India", "United Kingdom", "United States"),
  year %in% c(1957, 1967, 1977, 1987, 1997, 2007)) %>%
  group_by(year, country) %>%
  summarize(gdp = gdpPercap * pop) %>%
  ggplot(aes(x = factor(year), y = gdp, fill = country)) +
  geom_col(position = "fill") +
  scale_fill_brewer(palette = 9, type = "div") + theme_minimal()
```

Fijaos en que esta opción nos permite representar tres variables. En el eje horizontal hemos pasado la variable año a factor para que nos la considere categórica; el eje vertical nos muestra el PIB per cápita en términos relativos en vez de términos absolutos, mientras que las barras están fraccionadas por países. Podemos observar que el tamaño relativo del PIB ha disminuido claramente en el caso de los Estados Unidos y el Reino Unido, y ha aumentado en el caso de China y la India.

Evolución de los diagramas de barras

A la hora de representar el resumen de una variable numérica siguiendo los valores de una variable categórica, los diagramas de barras se sustituyen progresivamente por otras representaciones más sofisticadas que pueden representar mejor esta relación. Pensémoslo bien. La barra representa un solo dato, el sumario de una serie de valores de y, pero con la barra no podemos visualizar otra información útil como la distribución de los valores, no sabemos el mínimo ni el máximo, ni tampoco la cantidad de valores. Para hacernos una idea de la evolución de los diagramas de barras, imprimid el código siguiente, que muestra un gráfico que representa exactamente los mismos valores (y mucha más información) que la figura 4.

```
gapminder %>%
  ggplot(aes(x = continent, y = gdpPercap, col = continent)) +
  geom_point(alpha = 0.2, position = position_jitter(width = 0.2),
             show.legend = FALSE) +
  stat_summary(fun.y = mean, geom = "point", col = "black") +
  theme_light()
```

Cuando utilizamos el estético *fill* para representar varios colores dentro de las barras, podemos elegir la posición de las columnas como también hemos podido hacer con el histograma. Tanto `geom_bar()` como `geom_col()` tienen por defecto el argumento `position = "stack"`, que apila las barras una encima de la otra, pero podremos escoger tres maneras más de visualizar la posición de las columnas.

Alternativas para visualizar las columnas

Con `"fill"` nos llena cada columna hasta arriba de todo y nos enseña las proporciones de cada una. La posición `"dodge"` ubica las barras una junto a la otra (va bien cuando tenemos las variables *x* o *fill* con pocas categorías, por ejemplo una variable binaria), mientras que `"identity"` nos muestra la estatura real de cada barra sin apilarlas. Esta última opción se visualiza mejor con transparencias (atributo `alpha`).

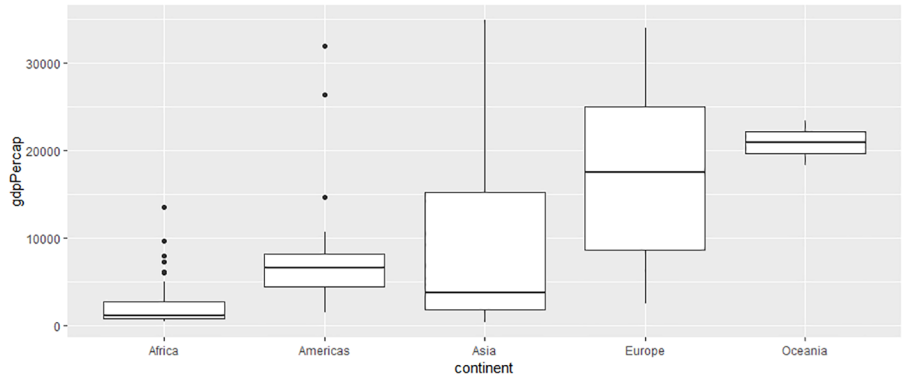
1.4. Diagrama de cajas

El diagrama de cajas nos ayuda a observar y comparar cómo una variable numérica está distribuida según los valores de una variable categórica. A diferencia del diagrama de barras, que solo nos permite visualizar un solo valor por medio de la altura de cada barra, la utilidad del diagrama de cajas es principalmente que nos permite visualizar algunos estadísticos descriptivos, como la mediana de cada distribución o los casos extremos. Con R, representaremos esta figura geométrica con la función `geom_boxplot()`. En el código siguiente hemos pedido un diagrama de cajas para ver cómo estaban distribuidos en 1992 los valores de la variable *gdpPercap* en varios continentes.

```
gapminder %>% filter(year == 1992) %>%
  ggplot(aes(x = continent, y = gdpPercap)) + geom_boxplot()
```

Este código imprime un gráfico que da varios estadísticos descriptivos de la manera como la variable numérica está distribuida según los valores de la variable categórica que hemos representado en el eje de las *x*. Es por eso que el diagrama de cajas es un diagrama bastante sofisticado, que contiene muchos elementos, y requiere práctica para poder ser interpretado adecuadamente. En la figura 7 representamos el diagrama de cajas que genera el código anterior.

Figura 7. Diagrama de cajas



Observemos el diagrama paso a paso. En primer lugar, la línea más gruesa que hay dentro de las cajas es la mediana de la distribución (el número que separa la distribución ordenada en dos partes iguales). Vemos que el único continente con una mediana superior a los 20.000 dólares es Oceanía. Esto significa que si ordenamos por PIB per cápita todos los países de Oceanía, de más grande a más pequeño, el que está en mitad de la distribución supera ligeramente

[4]Cabe decir que Oceanía solo tiene dos países en la muestra, Australia y Nueva Zelanda. Por lo tanto, la cifra que vemos es la que está entremedias del PIB per cápita de cada país.

los 20.000 dólares.[4] Europa se sitúa aproximadamente en los 17.000, mientras que el resto de continentes no superan los 10.000 dólares de mediana. Las partes superior e inferior de cada caja representan el 75 y el 25 por ciento de la distribución. Es decir, si tuviéramos 100 países en cada continente, la parte superior de la caja nos indicaría el 25.º país más rico y la parte inferior el 75.º Las cajas de África, América y Oceanía son muy pequeñas, lo cual significa que hay poca diferencia de riqueza entre el país que hace el 25 y el país que hace el 75 por ciento de la distribución ordenada: en África casi todos son muy pobres, en América casi todos son más bien pobres, mientras que en Oceanía todos son bastante ricos. En cambio, las cajas de Asia y Europa son mucho más largas, lo cual significa que tanto en Asia como Europa la distribución es más dispersa y hay tanto países ricos como países pobres.

Fuera de las cajas, encontramos dos figuras geométricas: líneas y puntos. Las líneas que salen de las cajas hacia arriba y hacia abajo nos indican el intervalo donde se encuentran los países por debajo del 25 por ciento y por encima del 75 por ciento. Cuando hay casos extremos, que son en este caso países muy alejados del resto de la distribución, se representan con puntos.

Líneas y casos extremos

En el caso de *ggplot2*, las líneas hacen como mucho 1,5 del tamaño de la caja. A partir de esta distancia, la longitud de la línea se establece buscando el último valor por debajo de 1,5. Los puntos extremos son todos aquellos valores que superan la longitud de la línea.

Fijaos en que en este diagrama de cajas hemos representado exactamente los mismos datos que en el histograma de la figura 1 y parecidos a los datos de la figura 4. Todos tienen sus ventajas e inconvenientes, y para seleccionar el diagrama más adecuado tenemos que pensar cuál es nuestro propósito y cómo se visualizarán mejor los datos. El histograma y el diagrama de densidad nos permiten dar un vistazo general a la forma que tiene la distribución de una variable. Si la distribución es bimodal, cosa que significa que tenemos dos intervalos con concentraciones altas de valores, estos gráficos nos permitirán distinguir con claridad estas dos puntas. En cambio, esta observación no la podremos hacer con un diagrama de barras, ni con el diagrama de cajas.

La ventaja de los diagramas de cajas y de barras es que podemos ver la distribución en orientación horizontal o vertical añadiendo la capa de coordenadas `coord_flip()`. El diagrama de cajas permite hacer más comparaciones, puesto que describe varias propiedades de los datos, como los valores que se encuentran en el 25, el 50 y el 75 por ciento de la distribución, la amplitud de la distribución y también los casos extremos.

1.5. Diagrama de dispersión

El diagrama de dispersión es una figura de dos dimensiones que representa con puntos la relación entre dos variables numéricas. Con R, representaremos esta figura geométrica con la función `geom_point()`. Más allá de lo que ya hayamos podido ver hasta ahora sobre el diagrama de dispersión, en este apartado destacaremos la manera en que podemos solucionar gráficamente uno de los problemas habituales de este tipo de gráfico: la sobrerrepresentación. Este fenómeno es habitual en variables ordinales y en variables numéricas discretas, cuando los puntos representan una cantidad de valores limitada y se

mueven en un rango más bien pequeño. Esto hace que los puntos se puedan solapar entre ellos. La sobrerrepresentación también es habitual en el caso de variables nominales cuando tenemos una gran cantidad de puntos para representar. Visualmente, los puntos se solapan y no podemos distinguir bien los unos de los otros.

Por poner un ejemplo, pensemos en las variables *año* y *esperanza de vida*. Son dos variables numéricas que tendrían que poder ser representadas con un diagrama de dispersión. Sin embargo, observad qué pasa en la figura 8 cuando intentamos representar estas dos variables del marco de datos *gapminder*. Es evidente que tenemos un problema importante de sobrerrepresentación. La variable *año* admite pocos valores y está tipificada como vector entero, mientras que la variable *esperanza de vida* puede tener poca variación entre sus valores. Con todo, los puntos se solapan.

Figura 8. Diagrama de dispersión con sobrerrepresentación

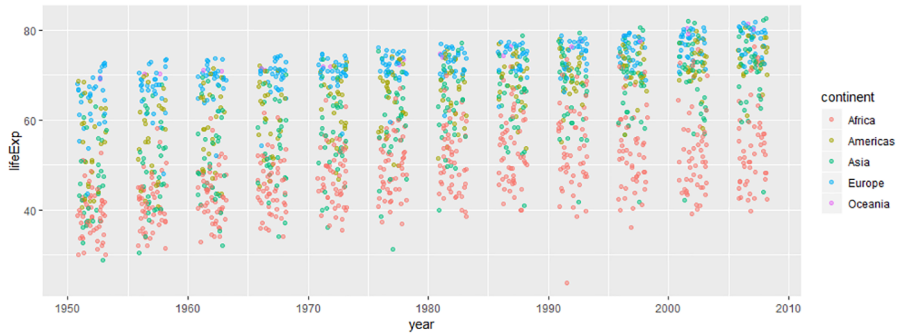
Si nos fijamos en los colores que representan los continentes, sospechamos que los países africanos tienen menos esperanza de vida y los europeos la tienen más elevada, pero nos gustaría ver los datos de manera más nítida para llegar a conclusiones más concretas. Una buena solución es sustituir `geom_point()` por `geom_jitter()`[5], como vemos en el código siguiente, que añadirá un «ruido» o movimiento aleatorio a la posición de los puntos.

[5]Esta función hace lo mismo que `geom_point(position = "jitter")`. Por defecto, la posición de `geom_point()` es `"identity"`.

```
gapminder %>% ggplot(aes(x = year, y = lifeExp, col = continent)) + geom_jitter(width = 1.2,
height = 0.1, alpha = 0.5)
```

El *jittering* separa los puntos entre sí mediante la introducción de un movimiento aleatorio horizontal y vertical que nos permite observar mejor el diagrama de dispersión. Si no añadimos nada dentro el paréntesis de `geom_jitter()` tendremos la misma aleatoriedad en vertical que en horizontal. Sin embargo, hemos querido especificar con el argumento `width` que el movimiento aleatorio en horizontal sea elevado, y en cambio con el argumento `height` hemos ordenado que el movimiento aleatorio en vertical sea mínimo. En la figura 9 vemos el resultado.

Figura 9. Diagrama de dispersión con *jittering*



Otro método que hemos usado para reducir la sobrerrepresentación es el argumento `alpha`, que introduce transparencia en los puntos.

1.6. Retoques finales

Hay miles de retoques que se pueden hacer en un gráfico mediante *ggplot2* para mejorar la visualización. Aquí os explicaremos dos retoques:

- poner títulos al gráfico y a los ejes,
- eliminar casos extremos.

Para cambiar los títulos lo podemos hacer de tres maneras:

1) La primera consiste en especificarlo en las escalas de *x* y de *y*, como vemos en la primera línea del código siguiente. En el ejemplo, si trabajamos con una variable numérica en el eje de las *x*, especificaremos el argumento *name* en la función `scale_x_continuous()`.[6]

2) Si no necesitamos modificar ninguna escala, nos será más fácil y rápido introducir argumentos dentro de la función `labs()` indicando el título general con `title`, el eje horizontal con *x* y el eje vertical con *y*. Si tenemos una leyenda que muestra los valores de un estético adicional solo habrá que introducir el nombre del estético (*size*, *color*, *fill*...) seguido del título.

3) Los ejes horizontal y vertical, como también el título general del gráfico, se pueden introducir por medio de funciones separadas: `ggtitle()` permite introducir el título, `xlab()` permite introducir el título del eje horizontal e `ylab()` el título del eje vertical.[7]

[6] Siempre pondremos *scale*, el nombre del estético y si la variable es *continuous* o *discrete*. Esto también implica estéticos como el color. Por ejemplo, en el caso de un estético categórico de color introduciremos `scale_color_discrete()`.

[7] Si pedís ayuda de la función `theme()` obtendréis mucha información sobre todo lo que podemos llegar a hacer con los títulos. También podéis consultar más información en este enlace (http://www.sthda.com/english/wiki/ggplot2-title-main-axis-and-legend-titles).

```
scale_x_continuous(name = "Título eje x")
labs(title = "Título", x = "Título eje x", y = "Título eje y")
ggtitle("Título") + xlab("Título eje x")+ ylab("Título eje y")
```

El segundo retoque final para generar una mejor visualización del gráfico es la eliminación de casos extremos (*outliers*). Estos casos deforman los datos e impiden observar con claridad el centro de la distribución. Por ejemplo, si queremos mirar la relación entre el PIB per cápita y la esperanza de vida en 1952, hay un caso extremo que nos deforma completamente el gráfico.

```
gapminder %>% filter(year == 1952) %>% ggplot(aes(x = gdpPercap, y = lifeExp)) + geom_point()
```

Vemos que solo tenemos un único caso superior a los 20.000 dólares per cápita y que se aleja mucho del resto de la distribución, puesto que supera un PIB per cápita de 100.000 dólares. Para eliminar este caso extremo, podemos reducir los límites del eje de las *x* simplemente introduciendo la función `xlim(c(0, 20000))`. Otra opción es introducir un nuevo argumento dentro de la función `filter()` para indicar que filtre los casos donde `gdpPercap < 25000`. Estas dos opciones eliminarán el caso extremo y reproducirán el resto de datos.

Podemos querer, sin embargo, conservar este caso extremo y generar un nuevo marco de datos en el cual lo tengamos señalado. En este caso, podemos utilizar una combinación de las funciones `filter()` y `mutate()`, como vemos a continuación. Con la primera fórmula creamos el nuevo marco de datos `gap52` filtrado por 1952 que contiene la variable lógica `extrem`, que es TRUE para los casos con PIB superior per cápita a los 25.000 dólares y FALSE para el resto. A continuación, hacemos dos operaciones:

1) En la primera, filtramos por el caso extremo, que nos devolverá todos los valores que sean TRUE en la columna *extrem*. Aquí podemos comprobar que Kuwait es el caso extremo de nuestros datos.

2) En la segunda operación filtramos excluyendo el caso extremo (con el símbolo `!`) y pedimos un *ggplot2* de los datos filtrados.

```
gap52 <- gapminder %>% filter(year == 1952) %>% mutate(extrem = gdpPercap > 25000)
> gap52 %>% filter(extrem)
# A tibble: 1 x 6
  country continent  year lifeExp    pop gdpPercap
  <fct>   <fct>     <int>   <dbl>  <int>     <dbl>
1 Kuwait  Asia       1952    55.6 160000   108382.
gap52 %>% filter(!extrem) %>% ggplot(aes(x = gdpPercap, y = lifeExp)) + geom_point()
```

Hay otras maneras de eliminar o filtrar casos extremos. La mayoría requerirán la función `filter()` o bien un poco de imaginación con las herramientas de los paquetes de R que ya conocemos.

2. Medidas de análisis univariante

Hasta ahora, hemos visto varias maneras de representar por medio de visualizaciones gráficas la información que contiene una variable. En este apartado buscaremos, en lugar de una visualización, uno o pocos números que nos puedan sintetizar la misma información. En otras palabras, lo que queremos es cuantificar la manera en que están distribuidos los valores en la variable. El ejercicio de encontrar medidas para sintetizar distribuciones por medio de cifras concretas nos permitirá obtener información comparable, de forma que podremos contrastar diferentes variables entre ellas. La manera como sintetizaremos la información, sin embargo, variará según el tipo de variable. En el caso de las variables categóricas, no podemos hacer demasiado más que observar las frecuencias de cada valor. En el caso de las variables numéricas, podemos resumir la información mediante las diversas medidas que indiquen la centralidad y la dispersión de la distribución y la localización de determinados valores. Buena parte de este trabajo la podemos hacer directamente con la función `summary()`, que devolverá las frecuencias de una variable categórica y algunas medidas de centralidad y localización de una variable numérica.

2.1. Tabla de frecuencias

La tabla de frecuencias es una de las modalidades más típicas para representar variables categóricas, como es el caso de la variable *continent* del marco de datos *gapminder*. Normalmente, en la tabla mostramos el número de veces que se repite cada valor categórico en la variable y su porcentaje sobre el total. Las frecuencias las podemos obtener con las funciones `table()` o `summary()`[8] y el porcentaje de frecuencias con la función `prop.table()`.

[8]La distinción principal entre `table()` y `summary()` es que la primera genera una tabla, mientras que la segunda genera un vector entero.

```
table(gapminder$continent)
prop.table(table(gapminder$continent))
```

Si trabajamos con variables categóricas nominales, tendremos bastante con las dos funciones anteriores para visualizar los datos, pero si se trata de variables ordinales también nos puede interesar ver el acumulado para cada categoría. Es decir, querremos observar cómo se acumulan los valores y los porcentajes si los añadimos a los valores o porcentajes de las filas anteriores. En el código siguiente vemos cómo crear una tabla con frecuencias y porcentajes acumulados a partir de la variable `income` del marco de datos `wb`.

Nota

Podéis encontrar el código del marco de datos `wb` en el anexo de este módulo.

```
> data.frame(categories = rev(unique(wb$income)),
             freq = rev(summary(wb$income))) %>%
     mutate(freq_cum = cumsum(freq),
            percent = round(prop.table(freq), 2),
            percent_cum = cumsum(percent))
```

```
      categories freq freq_cum percent percent_cum
1            low    3        3    0.20        0.20
2 lower-middle    8       11    0.53        0.73
3 upper-middle    3       14    0.20        0.93
4           high    1       15    0.07        1.00
```

La segunda columna de la tabla que hemos generado muestra las frecuencias que se repiten para cada categoría y la tercera columna muestra el acumulado de frecuencias, de forma que, por ejemplo, en la fila «lower-middle» vemos el acumulado de «lower-middle» y «low», en la fila «upper-middle» vemos el acumulado de «lower-middle», «low» y «upper-middle», y así sucesivamente. En las últimas dos columnas hacemos el mismo procedimiento: primero, enseñamos el porcentaje de frecuencias sobre el total en cada categoría y a continuación el porcentaje acumulado.

Si queremos hacer una tabla de frecuencias con otra variable, solo habrá que sustituir las dos veces que aparece `wb$income` por la variable categórica que queramos reproducir.

2.2. Medidas de centralidad

Para sintetizar una distribución en un solo valor, una manera es buscar dónde se encuentra ubicado el centro de la distribución. Fijaos en el vector siguiente. El objeto `ev` representa una distribución que nos indica la esperanza de vida, en números absolutos, recogida en once países de una región del mundo. Si queremos transformar estos datos en un solo dato numérico que nos resuma dónde se encuentra el centro de esta distribución, ¿cómo lo haríamos? ¿Qué número creéis que puede resumir mejor esta distribución?

```
ev <- c(65, 67, 68, 73, 74, 77, 80, 80, 80, 82, 83)
```

La respuesta es que dependerá de la pregunta concreta que hagamos, puesto que no hay una sola manera de sintetizar esta distribución para encontrar el centro. Principalmente, hay tres maneras de resumir estos datos:

1) La primera es buscar el valor que se repite más veces. A esto se le llama moda y, en nuestro caso, sería el número 80, que se repite tres veces.

2) La segunda es buscar el valor que se encuentra en el centro de la distribución ordenada. A esto se le llama mediana, que en el vector `ev` es el número 77, puesto que es el valor que queda justo en medio de la distribución si lo ordenamos de más pequeño a más grande.

3) La tercera manera es con la media, que se obtiene sumando todos los valores de la distribución y dividiéndolos por el número de casos.

2.2.1. Media

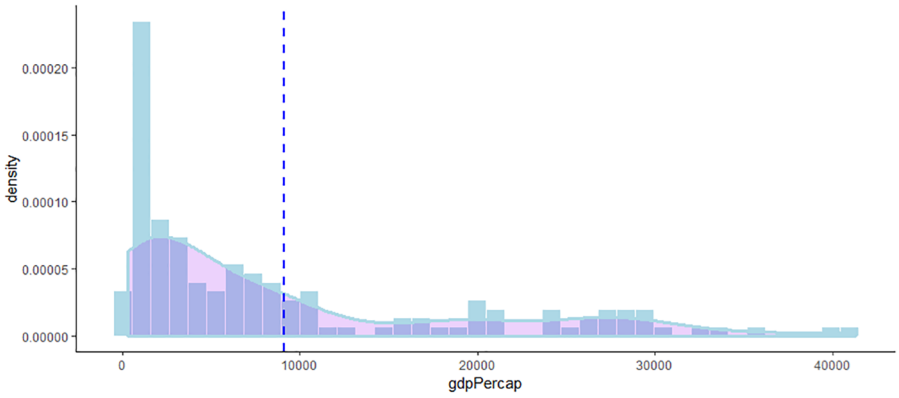
La media es una de las medidas de tendencia central más utilizadas para resumir distribuciones. Como vemos a continuación, para obtener la media tenemos que sumar todos los valores de un vector y dividir el resultado entre el número total de valores, que son 11.

```
media <- sum(c(65, 67, 68, 73, 74, 77, 80, 80, 80, 82, 83)) / 11
```

Lo que hace la media es decirnos cuál sería el valor que obtendríamos si niveláramos el peso de cada valor de la distribución entre sus casos. Una manera muy ilustrativa para entender qué representa la media es mediante la distribución de la riqueza de un país. En cualquier país, la riqueza acostumbra a estar distribuida de manera desigual. Hay individuos que tienen más dinero y hay otros que tienen menos. Lo que nos dice la media es cuánto dinero le tocaría a cada habitante si cogiéramos el dinero de toda la población y lo repartiéramos a partes iguales entre ellos. En todas las distribuciones la media hace exactamente esto: suma todos los valores y los divide entre el número de frecuencias.

Fijémonos en la figura 10, donde hemos superpuesto un diagrama de densidad y un histograma que ilustran cómo estaba distribuido el PIB per cápita en los diferentes países del mundo en 1997. Con línea azul hemos indicado dónde se encuentra la media.

Figura 10. Media del PIB per cápita mundial por países



Si cogemos el dinero de los países más ricos y lo repartimos entre todos los países de forma que el dinero quede distribuido equitativamente, veremos que todos los países tendrían 9.090 dólares por habitante. Podemos pensar en la media como el punto que equilibra los datos. Este valor divide el diagrama de densidad en dos áreas aproximadamente iguales, de forma que el área de densidad de la izquierda es igual al área de densidad de la derecha. La media se calcula con la función `mean()`[9]. En el código siguiente hemos creado el objeto `gap97_mean` donde hemos especificado la media con una línea vertical `geom_vline()`. Imprimid el objeto después de crearlo.

[9] Alternativamente, también se puede calcular con `sum()` / `length()`, donde primero sumamos los valores de un vector y después dividimos el resultado por el recuento de observaciones.

```
gap97_mean <- gapminder %>% filter(year == 1997) %>% ggplot(aes(x = gdpPercap)) +
```

```
geom_histogram(aes(y=..density..), colour= "white", fill= "lightblue", bins = 40) +
geom_density(col = "lightblue", size = 1.2, fill = "purple", alpha = 0.2) +
geom_vline(aes(xintercept = mean(gdpPercap)), colour = "blue", linetype = "dashed", size = 1) +
theme_classic()
gap97_mean
```

Las dos maneras más habituales de calcular la media es o bien dentro de la función `summarize()` o bien mediante un vector, como se indica en el código siguiente. En primer lugar, hemos creado un vector donde filtramos las observaciones por el año 1992 y, a continuación, hemos pedido la media del vector.

```
gdp_cap1992 <- filter(gapminder, year == 1992)
mean(gdp_cap1992$gdpPercap, trim = 0, na.rm = FALSE)
```

En este código tenemos que especificar dos argumentos adicionales que tiene por defecto la función de la media. Con el argumento *trim* podemos eliminar casos extremos de la distribución. Tendremos que especificar un valor de 0 a 0,5, que indicará el porcentaje de observaciones que eliminará de cada extremo de la distribución. El argumento *na.rm* es extremadamente importante, puesto que elimina los NA de la distribución. El marco de datos *gapminder* no tiene datos perdidos, pero será frecuente encontrárnoslos en la mayoría de marcos de datos. Es por eso que tendremos que cambiar el argumento a TRUE para que nos muestre la media.

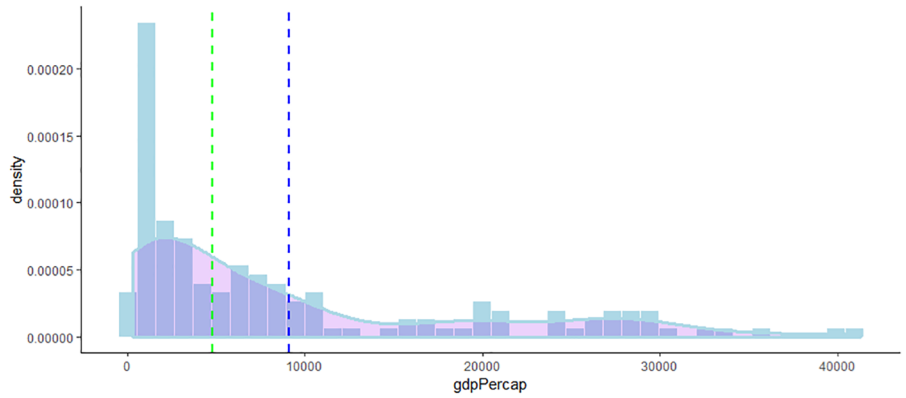
2.2.2. Mediana

La mediana nos indica qué valor tiene la observación que se encuentra exactamente en medio de la distribución. Dicho de otro modo: si tenemos una distribución con 144 países, la mediana nos ordena los valores de más grande a más pequeño, y nos indica el valor del país que hace 72, que es el que se encuentra en medio de la distribución (lo obtenemos dividiendo 144 entre dos). Como vemos en el código siguiente, hemos utilizado el objeto `gap97_mean`, ya creado, que contiene toda la información sobre la geometría anterior, para añadirle una línea nueva con información sobre la mediana.

```
gap97_mean + geom_vline(aes(xintercept = median(gdpPercap)), color="green",
linetype = "dashed", size = 1)
```

En la figura 11 hemos ilustrado el resultado del código donde vemos una nueva línea vertical, de color verde, que representa la mediana.

Figura 11. Mediana y media del PIB per cápita mundial por países



Es interesante ver cómo los dos centros divergen claramente. Si calculamos la mediana, veremos que se encuentra en los 4.782 dólares por habitante, mientras que la media se encontraba en los 9.090 dólares. Como la gran mayoría de observaciones ocurren en los valores más bajos del histograma, es lógico que la mediana se sitúe más sesgada a la izquierda de la distribución.

Otras medidas de localización

La mediana es una medida tanto de centralidad como de localización, puesto que nos indica el valor por debajo del cual tenemos ubicado el 50 por ciento de observaciones de una distribución ordenada. Para referirnos a otras localizaciones de la distribución diferentes a la mediana, utilizaremos términos como:

- los percentiles (si contamos por unidades del 1 al 100),
- los deciles (si contamos de 10 en 10),
- los quintiles (de 20 en 20) y los
- cuartos (de 25 en 25).

Localizar el valor es fácil con la función `quantile()`, donde solo tenemos que indicar el nombre del objeto y la posición del valor que queremos localizar en la escala de 0 a 1. Por ejemplo, si queremos encontrar el percentil 40 (que también sería el cuarto decil o el segundo quintil) marcaremos `quantile(vector, 0.4)`. Podemos indicar tantas localizaciones como queramos si utilizamos el concatenado. Si queremos saber todos los quintiles introduciremos `quantile(vector, c(0, 0.2, 0.4, 0.6, 0.8, 1)`. El cero nos devolverá el valor más bajo y el 1 nos devolverá el valor más alto de la distribución.

2.2.3. Moda

La observación más repetida en una distribución es la moda. Pueden ser ejemplos de moda la religión o el partido político dominante en un país. La moda es más fácil de encontrar en una variable categórica, puesto que será sencillamente la categoría con más frecuencias. Si imprimimos el sumario o pedimos una tabla de la variable continente de *gapminder* encontraremos fácilmente la moda. Vemos claramente que la moda es África, la distribución con más frecuencias: 624.

```
> summary(gapminder$continent)
> table(gapminder$continent)
  Africa Americas     Asia   Europe  Oceania
     624      300      396      360       24
```

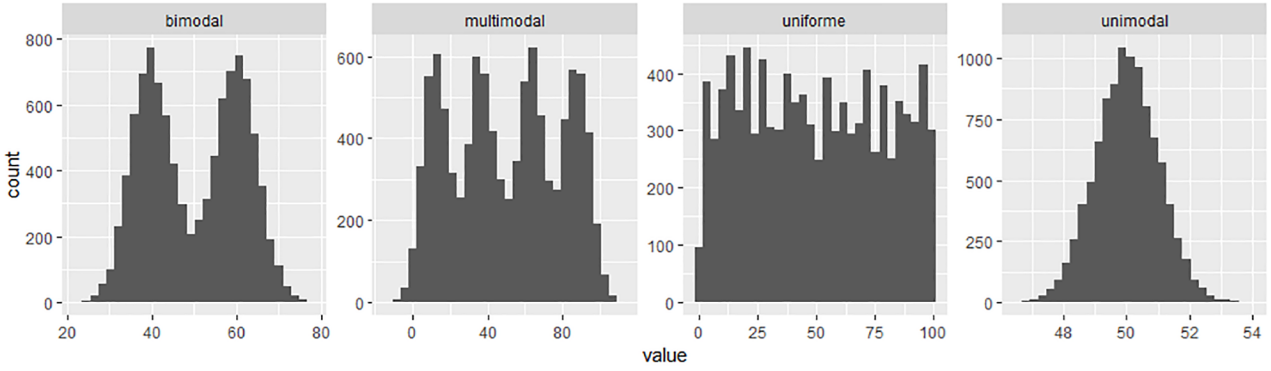
Encontrar la moda en una variable numérica, en cambio, no es un ejercicio tan evidente. Tenemos que pensar que estas variables pueden adoptar un número infinito de valores y, por lo tanto, es poco probable encontrar algún número repetido y, si se da el caso, no nos aportará información más allá de la casualidad de que se repitan. La moda solo puede ser fácil de encontrar en el supuesto de que la variable en cuestión sea discreta, como la edad de una población. En este caso, si tenemos un número suficientemente grande de observaciones en relación con los valores que puede adoptar la variable, podremos obtener la moda sin problemas.

La discrecionalidad de la moda

Este procedimiento tiene un punto de discrecionalidad, puesto que podemos cambiar los intervalos del histograma o la sensibilidad del diagrama de densidad. Esto puede hacer que la moda que establezcamos visualmente varíe en función de los parámetros que establecemos. Haced la prueba: coged el código que hemos utilizado para crear la figura 10 y cambiad los intervalos (*binwidth*) a 20. Veréis cómo, por ejemplo, la moda está más repartida entre dos intervalos.

En variables numéricas continuas, sin embargo, lo tenemos más complicado. Para obtener información sobre nuestro análisis, será más útil, en lugar de encontrar un valor concreto, saber dónde están concentrados los valores de la variable. Este ejercicio lo podemos hacer fácilmente con un histograma o un diagrama de densidad, en el cual podemos visualizar cuál es el punto más alto de la distribución. En la figura 12 podemos observar los diferentes tipos de distribuciones que pueden dar lugar a diferentes tipos de modas.

Figura 12. Tipos de moda



Hablaremos de distribución unimodal cuando solo haya una punta, bimodal cuando haya dos, multimodal cuando haya más de dos puntas y uniforme cuando no se aprecie ninguna punta concreta.

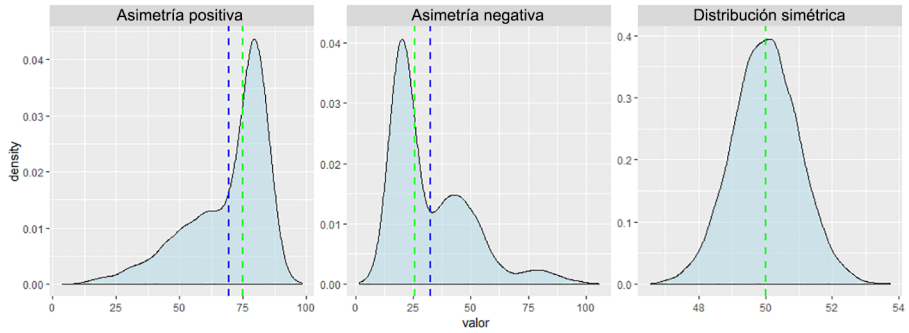
2.2.4. Tipos de transformaciones

A medida que nos vamos familiarizando con diferentes tipos de distribuciones, veremos que podemos intuir la forma que tienen sin visualizarlas. Tendremos bastante con conocer la moda, la media y la mediana. Si la media es inferior a la mediana, seguramente nos encontraremos con un gráfico parecido al diagrama de densidad de la izquierda de la figura 13 y diremos que es una distribución asimétrica positiva.[10] Como vemos, la cola del gráfico está a la izquierda y los valores están sesgados a la derecha de la distribución. Si la media, en cambio, es superior a la mediana, posiblemente nos encontraremos con el gráfico del medio y diremos que es una distribución asimétrica negativa. La cola del gráfico, en este caso, se encuentra a la derecha de la distribución y los valores están sesgados a la izquierda de la distribución. En último lugar,

(10)Con el diagrama de densidad se pierde información en relación con el histograma, puesto que no tenemos la información separada por intervalos. Aun así, es útil cuando queremos añadir información encima del gráfico, puesto que visualmente no quedará sobrecargado.

si la moda, la media y la mediana coinciden más o menos, querrá decir que tenemos una distribución simétrica. En este caso, la cola a un lado y al otro es parecida y no hay sesgo.

Figura 13. Distribuciones simétricas y asimétricas



Las distribuciones de los ingresos en un país acostumbran a tener una asimetría negativa. Esto quiere decir que la mayoría de casos están situados a la izquierda de la distribución y la mediana es inferior a la media. En otras palabras, el individuo situado en mitad de la distribución (la mediana) no tiene tanto dinero como el que le tocaría a cada individuo si repartiéramos todo el dinero a partes iguales entre el conjunto de población (la media).

Sensibilidad de la media

La media es sensible a distribuciones sesgadas y valores extremos, de forma que, en estos casos, puede ser un método más adecuado para medir la centralidad. Una manera de evitar que los valores extremos influyan excesivamente en el cálculo de la media es con el argumento *trim*, que elimina la proporción que indicamos de cada extremo. Por ejemplo, `mean(variable1, trim = 0.01)` eliminará el último 1 por ciento de cada extremo.

A menudo, para visualizar mejor estas distribuciones y hacer pruebas de significación estadística, se acostumbra a construir una versión transformada de la variable. En los casos de distribuciones con asimetría negativa, se aplica la escala logarítmica con las funciones `log()`, `log2()` o `log10()`, mientras que en distribuciones con asimetría positiva se puede aplicar el exponencial con la función `exp()`.

2.3. Medidas de dispersión

Las medidas de dispersión nos indican la separación o dispersión de los valores en una distribución numérica. En general, si los valores están muy concentrados alrededor del centro indicarán un valor bajo, mientras que si están muy separados indicarán un valor alto. Hay tres maneras principales de mirar la dispersión:

- el rango,
- el rango intercuartílico,
- la desviación típica.

2.3.1. Rango

El rango muestra la diferencia entre el valor máximo y el valor mínimo de una distribución. Es una medida de dispersión que no se utiliza mucho porque es muy sensible a los valores extremos, de forma que un solo valor puede tener un gran efecto en el valor que muestre el rango. En el código siguiente indicamos dos maneras de calcular el rango:

1) La primera es utilizar la función `range()` para encontrar los valores máximo y mínimo de una distribución y aplicar `diff()` para ver la diferencia entre estos dos valores.

2) La segunda es restarle el valor mínimo `min()` de la distribución al valor máximo `max()`.

```
> diff(range(gapminder$gdpPercap))
> max(gapminder$gdpPercap) - min(gapminder$gdpPercap)
113282
```

Los valores mínimo y máximo de la distribución también se pueden obtener mediante la función `summary()`.

2.3.2. Rango intercuartílico

A diferencia del rango, el rango intercuartílico (*InterQuartile Range*, *IQR*) es menos sensible a los valores extremos, puesto que mide la diferencia entre el primer y el tercero cuantil de la distribución. En otras palabras, mide la diferencia entre los valores que ocupan los lugares 25 y 75 por ciento del rango en una distribución ordenada. Esta medida de dispersión se puede obtener con la función `IQR()`. Alternativamente, como vemos en el código siguiente, también podemos calcularlo restando el cuantil 25 al cuantil 75 usando la función `quantile()` e indicando, en escala de 0 a 1, la posición correspondiente del primer y el tercer cuantil.

El IQR es la medida de la caja

Fijaos en la figura 7 de este módulo. Lo que hace el IQR es medir de forma cuantitativa el tamaño de las cajas de un diagrama de cajas.

```
> IQR(gapminder$gdpPercap)
> quantile(gapminder$gdpPercap, 0.75) - quantile(gapminder$gdpPercap, 0.25)
8123.402
```

El primer y tercer cuantiles de la distribución también se pueden obtener mediante la función `summary()`. Los encontraremos con el nombre de *1st Qu.* y *3rd Qu.*

2.3.3. Desviación típica

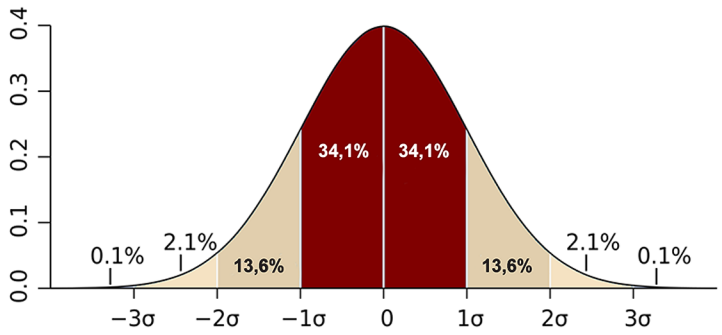
La desviación típica es una medida de dispersión bastante más compleja en comparación con las que hemos visto anteriormente, y es por eso que le tendremos que dedicar más espacio y atención. Esta medida indica la dispersión de los valores respecto de la media. Para entender su significado numérico tendremos que tener presentes tres cuestiones importantes:

1) En primer lugar, un valor alto de desviación típica querrá decir que, de media, los valores están situados lejos de la media, mientras que un valor bajo de desviación típica querrá decir que los valores están concentrados cerca de la media.

2) En segundo lugar, la desviación típica está medida con las mismas unidades que los valores originales. Esto quiere decir que si medimos la esperanza de vida en años, donde los valores oscilan entre 50 y 80 años, tendremos desviaciones típicas más bajas que si medimos esperanzas de vida en meses, donde tendremos valores más elevados. Si medimos el PIB per cápita obtendremos seguramente desviaciones típicas más altas, puesto que los valores de esta variable se mueven entre varios miles. Por lo tanto, la desviación típica nos será útil para comparar distribuciones, siempre que estén medidas en las mismas unidades.

3) En tercer y último lugar, la desviación típica nos indica con bastante precisión qué porcentaje de valores están situados alrededor de la media. En una distribución simétrica normal, como en la figura 14, una distribución típica nos indicará que el 68,2 por ciento de los valores están situados a la izquierda y a la derecha de la media. Dos desviaciones típicas nos indicarán que un 95,4 por ciento de los valores están situados a izquierda y derecha de la media y tres desviaciones típicas nos indicarán que el 99,6 de los valores están situados a izquierda y derecha de la media.

Figura 14. Distribución normal con desviaciones típicas



Font: M. W. Toews, CC BY 2.5. (https://en.wikipedia.org/wiki/Standard_deviation#/media/File:Standard_deviation_diagram.svg)

Para ver el concepto de la desviación típica de una manera más práctica, en el código siguiente hemos creado los vectores `dev_ex1` y `dev_ex2`, formados por nueve valores cada uno. Con la función `mean()` veremos que la media de

los dos vectores es la misma: 5,11. Esto significa que si vamos sacando peso a los valores numéricos más altos y damos este peso a los valores numéricos más bajos, de forma que al final nos quede este peso repartido de manera equitativa entre todos los valores, la cifra que obtendremos será 5,11.

```
dev_ex1 <- c(4, 6, 5, 8, 5, 3, 5, 6, 4)
dev_ex2 <- c(3, 9, 4, 8, 6, 1, 5, 8, 2)
```

Sabemos que tanto en un vector como en el otro la media es la misma, pero ¿hasta qué punto es adecuado el número 5,11 para representarnos los valores de cada vector? Será más adecuado si todos los valores son cercanos a 5. En cambio, 5,11 será una medida menos fiel de representar los valores del vector si estos son valores alejados de 5, como 1, 2 o 9.

La desviación típica nos ayuda a hacernos una idea de si los valores de un vector están cerca o lejos de la media. Para calcularla hemos reproducido los pasos para el vector `dev_ex1` en el código siguiente. En primer lugar, calculamos la diferencia de cada uno de los vectores respecto de la media. Para los valores que sean inferiores a la media, esta diferencia nos saldrá negativa, mientras que para los valores superiores a la media la diferencia será positiva. Para hacer que todos los números sean positivos, elevaremos todos los valores al cuadrado y seguidamente los sumaremos. A continuación dividimos esta suma por el número de observaciones menos uno para evitar que el valor obtenido sea más grande a medida que tengamos más observaciones. El valor que obtenemos hasta aquí se denomina *varianza*. La desviación típica se obtiene de la raíz cuadrada de la varianza.[11]

[11]Hemos puesto las fórmulas del cuadro siguiente entre paréntesis para poder ver impreso en la consola cada uno de los pasos para obtener la desviación típica.

```
(diff_mediana <- dev_ex1 - mean(dev_ex1))
(diff_cuadrado <- (diff_mediana)^2)
(suma_diff_cuadrado <- sum(diff_cuadrado))
(varianza <- suma_diff_cuadrado/(length(dev_ex1) - 1))
(desviacion_tipica <- sqrt(varianza))
```

La desviación típica del vector `dev_ex1` es de 1,45, mientras que si cambiamos los códigos y buscamos la desviación típica del vector `dev_ex2` veremos que es de 2,84. Por suerte, para saber la desviación típica no habrá que hacer estas operaciones complicadas, sino que la función `sd()` nos la calculará directamente.

Y ahora que tenemos la desviación típica de estos vectores, ¿cómo podemos interpretarlos? Principalmente, lo que podemos decir es que en el primer vector encontraremos aproximadamente un 68,2 por ciento de los valores entre 3,66 y 6,56, mientras que en el segundo vector encontraremos el 68,2 por ciento de los valores entre 2,27 y 7,95.[12] Así, pues, si miramos las desviaciones típicas de uno y otro vector, podemos interpretar que los valores del segundo

[12]Estos números los obtenemos restando una desviación típica a la mediana (5,11 – 1,45 = 3,66, y 5,11 – 2,84 = 2,27) y sumando una desviación típica a la mediana (5,11 +1,45 = 6,56, y 5,11 + 2,84 = 7,95). La estimación asume que se trata de una distribución simétrica normal con un número elevado de casos.

vector están más dispersos en relación con la mediana que en el primero. Si añadimos o restamos dos desviaciones típicas a la media podremos saber también dónde estarán situados el 95,4 y el 99,6 por ciento de los valores.

La desviación típica es mucho más sensible a los valores extremos que el IQR, pero lo es menos que el rango.

Ejercicio: ejemplo de sensibilidad a los valores extremos

Para probar la sensibilidad de las medidas sintéticas de dispersión ante casos extremos, fijaos en este vector: `c(1,2,3,4,5,6,7,8,9,10,20)`. Tenemos un caso extremo, el número 20. Probad a calcular el rango `diff(range())`, el rango intercuartílico `IQR()` y la desviación típica `sd()`. Después, sustituid el número 20 por el 11 y volved a hacer los mismos cálculos. ¿Cuál es la medida de dispersión más sensible a los casos extremos? ¿Y cuál es la menos sensible?

Resumen

En este módulo hemos aprendido diferentes técnicas de visualización y cuantificación en el análisis de una sola variable. En la primera sección, no solamente hemos visto cómo analizar visualmente una variable, sino también cómo cruzarla con una segunda variable para poder observar con más matices sus valores. Esto nos sirve también para repasar algunos procedimientos del paquete *dplyr* que nos ayudan a transformar variables. Tenemos que recordar, principalmente, que visualizaremos las variables numéricas con un histograma y las numéricas a lo largo del tiempo con un diagrama de líneas. Con el diagrama de cajas también podemos visualizar una variable numérica, y nos es especialmente útil cuando la cruzamos con una variable categórica, puesto que podemos comparar más fácilmente algunos de sus parámetros estadísticos. Utilizaremos el diagrama de dispersión para analizar dos variables numéricas, mientras que reservaremos el diagrama de cajas para las variables categóricas.

La segunda parte de este módulo la hemos dedicado a las técnicas para cuantificar variables. Los estadísticos descriptivos que podemos sacar variarán según si la variable que queremos analizar es numérica o categórica. La función `summary()` ilustra de una manera muy clara las posibilidades principales que nos da cada tipo de variable. Es importante retener que el análisis visual y el análisis cuantitativo son en parte sustitutivos y en parte complementarios. Son sustitutivos porque con la visualización de una variable nos podemos hacer la idea de las medidas cuantitativas asociadas a esta variable. En otras palabras, visualizar una distribución ya nos permite intuir cómo estarán situadas la media, la mediana, la moda o la desviación típica. Y viceversa, si sabemos las medidas de centralidad y dispersión de una distribución podemos hacernos una idea de qué forma visual tendrá. Son complementarias porque si miramos una variable por los dos lados, la parte visual y la parte cuantitativa, podremos ver más matices y nos ayudará a hacer un análisis univariante más detallado.

Ejercicios de autoevaluación

Para un mejor el aprendizaje, intentad hacer mentalmente el máximo de ejercicios posible, sin utilizar R.

1. Cambia la función para que nos muestre las proporciones en cada intervalo.

```
geom_histogram()
```

2. Tenemos una distribución con valores de 10 a 100. ¿Qué argumento indicaremos para tener un histograma formado por 15 intervalos?

```
geom_histogram()
```

3. Visualiza el diagrama de densidad siguiente con una transparencia del 50 por ciento.

```
geom_density()
```

4. Genera un diagrama de línea con todas las variables del marco de datos siguiente.

```
md$exports, md$year, md$country
```

5. Cambia los estéticos de la geometría de forma que el color de la línea y el tipo de línea muestre la variable `country`.

```
geom_line()
```

6. Indica la geometría correcta para visualizar el recuento de observaciones.

```
geom_bar(), geom_col()
```

7. Elimina la leyenda del gráfico con un argumento dentro de la geometría.

```
geom_col()
```

8. Pide un diagrama de cajas con las variables siguientes.

```
md$categorica, md$numerica
```

9. Introduce ruido horizontal en el diagrama de dispersión siguiente.

```
ggplot(md, aes(x, y)) + geom_point()
```

10. Sin utilizar las funciones de *dplyr*, crea el vector lógico `extrem` a partir de la variable siguiente para detectar casos superiores al valor 5.

```
militarydata$gasto_pib
```

11. Pide una tabla de frecuencias con porcentajes de la variable siguiente.

```
tradedata$countries
```

12. Genera una tabla de frecuencias de clase vector para la variable siguiente.

```
tradedata$countries
```

13. ¿Cuál es la moda del vector siguiente?

```
c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10)
```

14. ¿Cuál es la mediana del vector siguiente?

```
c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10)
```

15. Simplifica las funciones siguientes para calcular la media de este objeto.

```
sum(distribution) / length(distribution)
```

16. Calcula el cuantil 66 del vector siguiente.

```
militarydata$gasto_pib
```

17. Pide el valor mínimo del vector siguiente.

```
militarydata$gasto_pib
```

18. Pregunta a R si la IQR del vector siguiente es inferior al rango.

```
md$vector
```

19. ¿Cuál es el resultado obvio del ejercicio 18?

```
TRUE o FALSE
```

20. Calcula la desviación típica del vector siguiente.

```
militarydata$vector
```

Solucionario

1. `geom_histogram(position = "fill")`

2. `geom_histogram(bins = 15)` o `geom_histogram(binwidth = 6)`

3. `geom_density(alpha = 0.5)`

4. `ggplot(md, aes(x = year, y = exports, col = country)) + geom_line()`

5. `geom_line(aes(col = country, lty = country)`

6. `geom_bar()`

7. `geom_col(show.legend = FALSE)`

8. `ggplot(md, aes(x = categorica, y = numerica)) + geom_boxplot()`

9. `ggplot(md, aes(x, y)) + geom_jitter(width = 1.2, height = 0)`

10. `extrem <- militarydata$gasto_pib > 5`

11. `prop.table(table(tradedata$countries))`

12. `summary(tradedata$countries )`

13. `10`

14. `6`

15. `mean(distribution)`

16. `quantile(militarydata$gasto_pib, 0.66)`

17. `min(militarydata$gasto_pib)`

18. `IQR(md$vector) < diff(range(md$vector))`

19. `TRUE`

20. `sd(militarydata$vector)`

Glosario

`geom_bar()` Introduce la geometría de un diagrama de barras.

`geom_boxplot()` Introduce la geometría de un diagrama de cajas.

`geom_col()` Introduce la geometría de un diagrama de barras.

`geom_density()` Introduce la geometría de un diagrama de densidad.

`geom_histogram()` Introduce la geometría de un histograma.

`geom_hline()` Introduce la geometría de una línea horizontal.

`geom_jitter()` Introduce ruido en un diagrama de dispersión.

`geom_line()` Introduce la geometría de un diagrama de línea.

`geom_point()` Introduce la geometría de un diagrama de dispersión.

`geom_vline()` Introduce la geometría de una línea vertical.

`ggtitle()` Permite poner un título en el gráfico.

`IRQ()` Muestra el rango intercuartílico de una distribución.

`labs()` Permite poner etiquetas en el gráfico.

`max()` Muestra el valor más alto de una distribución numérica.

`mean()` Devuelve la suma de todos los valores de una distribución dividido por el número de valores.

`median()` Devuelve el valor central de una distribución ordenada.

`min()` Muestra el valor más bajo de una distribución numérica.

`prop.table()` Permite obtener los porcentajes de una tabla de frecuencias.

`quantile()` Muestra el valor del cuantil que especificamos de una distribución.

`range()` Devuelve el valor mínimo y máximo de una distribución.

`sd()` Muestra la desviación típica de una distribución.

`table()` Muestra la tabla de frecuencias de una variable categórica.

`xlab()` Permite introducir el título de eje de las x.

`ylab()` Permite introducir el título de eje de las y.

Bibliografía

Babbie, E. R. (2013). *The practice of social research*. Wadsworth: Cengage Learning.

Brancati, D. (2018). *Social Scientific Research*. Londres: Sage Publications Ltd.

Halperin, S.; Heath, O. (2016). *Political Research: Methods and Practical Skills*. Oxford: Oxford University Press.

King, G.; Keohane, R. O.; Verba, S. (1994). *Designing Social Inquiry: Scientific Inference in Qualitative Research*. Princeton: Princeton University Press.

Anexo del módulo

Código Apartado 2.1. Tabla de frecuencias

```
wb <- data.frame(country = c("Antigua and Barbuda", "Belice", "Costa Rica", "Dominica",
"Dominican Republic", "El Salvador", "Guyana", "Guatemala", "Haiti", "Honduras", "Jamaica",
"Nicaragua", "Panama", "Surinam", "Trinidad and Tobago"), income = factor(c("high", "upper-middle",
"upper-middle", "upper-middle", "upper-middle", "lower-middle", "upper-middle", "upper-middle",
"low", "lower-middle", "upper-middle", "lower-middle", "high", "upper-middle", "high")),
stringsAsFactors = FALSE)
```

Código Figura 12

```
moda <- data.frame(
  unimodal = rnorm(10000, 50), #unimodal
  bimodal = c(rnorm(5000, 60, 5), rnorm(5000, 40, 5)), #bimodal
  multimodal = c(rnorm(2500, 12, 7), rnorm(2500, 37, 7),
                 rnorm(2500, 63, 7), rnorm(2500, 88, 7)),#multimodal
  uniforme = rep(sample(1:100, 5000, replace = T), 2))
moda_ty <- gather(moda, tipo)
ggplot(moda_ty, aes(x = value)) +
  geom_histogram() +
  facet_grid(. ~ tipo, scale = "free")
```

Código Figura 13

```
simetria <- data.frame(simetrica = rnorm(10000, 50),
              dre = c(rnorm(5500, 80, 5), rnorm(2500, 65, 9),
                      rnorm(1500, 50, 9), rnorm(500, 30, 9)),
              esq = c(rnorm(5500, 20, 5), rnorm(2500, 40, 9),
                      rnorm(1500, 50, 9), rnorm(500, 80, 9)))
distr_ty <- gather(simetria, tipo, valor)
mm <- distr_ty %>%
  group_by(tipo) %>%
  summarize(grp.mean = mean(valor),
            grp.median = median(valor))
ggplot(distr_ty, aes(x = valor)) +
  geom_density(fill = "lightblue", alpha = 0.5) +
  facet_wrap(. ~ tipo, scale = "free",
             labeller = as_labeller(c(dre = "Asimetría positiva",
                                      esq = "Asimetría negativa",
                                      simetrica = "Distribución simétrica"))) +
  geom_vline(data = mm, aes(xintercept=grp.mean),
           color="blue", linetype="dashed",
```

```
            size=1) +
  geom_vline(data = mm, aes(xintercept=grp.median),
             color="green", linetype="dashed",
             size=1)
```