

Análisis de encuestas de hogares con R

Módulo 5: Análisis gráfico en R

CEPAL - Unidad de Estadísticas Sociales

Tabla de contenidos I

Graficas en R

Gráficas de variables continuas.

Scaterplot

Diagrama de barras para variables categoricas

Creando mapas

Graficas en R

Introducción

- ▶ Este apartado se centra en mostrar cómo crear gráficos generales en R, lo cual es esencial en el análisis de encuestas para visualizar tendencias y verificar supuestos en el ajuste de modelos estadísticos.
- ▶ Se introduce el paquete ggplot2, una herramienta poderosa y flexible para la creación de gráficos elegantes en R. Fue desarrollado por Hadley Wickham y se basa en el concepto de “Grammar of Graphics.”
- ▶ La carga inicial de librerías y bases de datos es un paso común antes de comenzar cualquier análisis gráfico en R.

Lectura de la base

```
data(BigCity, package = "TeachingSampling")
encuesta <- readRDS("../Data/encuesta.rds")
```

Definir diseño de la muestra con srvyr

Definiendo el diseño muestral, esto se hace de forma análoga a la anterior.

```
library(srvyr)
diseno <- encuesta %>%
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk,
    nest = T
  )
```

Definir nuevas variables

Creando nuevas variables, para ello se hace uso de la función `mutate`.

```
diseno <- diseno %>% mutate(  
  pobreza = ifelse(Poverty != "NotPoor", 1, 0),  
  desempleo = ifelse(Employment == "Unemployed", 1, 0),  
  edad_18 = case_when(  
    Age < 18 ~ "< 18 años",  
    TRUE ~ ">= 18 años"  
  )  
)
```

Sub-grupos

Dividiendo la muestra en sub grupos.

```
sub_Urbano <- diseno %>% filter(Zone == "Urban")  
sub_Rural <- diseno %>% filter(Zone == "Rural")  
sub_Mujer <- diseno %>% filter(Sex == "Female")  
sub_Hombre <- diseno %>% filter(Sex == "Male")
```

Creando tema para las gráficas

Para tener un estilo estándar las gráficas se define el siguiente tema.

```
theme_cepal <- function(...) {  
  theme_light(10) +  
    theme(  
      axis.text.x = element_blank(),  
      axis.ticks.x = element_blank(),  
      axis.text.y = element_blank(),  
      axis.ticks.y = element_blank(),  
      legend.position = "bottom",  
      legend.justification = "left",  
      legend.direction = "horizontal",  
      plot.title = element_text(size = 20, hjust = 0.5),  
      ...  
    )  
}
```

Gráficas de variables continuas.

Histogramas para graficar variables continuas.

- ▶ Un histograma es una representación gráfica de los datos de una variable que utiliza barras, donde la altura de las barras representa la frecuencia de los valores y el ancho de las barras corresponde a la amplitud de los intervalos de clase.
- ▶ Los pasos para realizar el histograma incluyen definir la fuente de información, especificar la variable a graficar (x) y los pesos de muestreo (weight). Luego, se elige el tipo de gráfico, en este caso, un histograma (geom_histogram). Además, se configuran los títulos deseados para el histograma y se aplica el tema de la CEPAL para mejorar su presentación.

Histogramas

Las gráficas son realizadas principalmente con la librería ggplot2y nos apoyamos en la librería patchwork para organizar la visual la visual de las gráficas.

```
library(ggplot2)
library(patchwork)
plot1_Ponde <- ggplot(
  data = encuesta,          # Fuente de datos.
  aes(x = Income, weight = wk) # Parámetros gráficos general.
) +
  geom_histogram(           # Parámetro geométrico.
    aes(y = ..density..)) + # Parámetros del gráfico
  ylab("") +                # Nombre para el eje Y
  ggtitle("Ponderado") +    # Título.
  theme_cepal()             # Aplicando tema
```

Histogramas

De forma análoga se define el gráfico siguiente, note que en este caso se omitió el parámetro `weight`.

```
plot1_SinPonde <-  
  ggplot(encuesta, aes(x = Income)) +  
  geom_histogram(aes(y = ..density..)) +  
  ylab("") +  
  ggtitle("Sin ponderar") +  
  theme_cepal()
```

Histogramas

Para el siguiente dato se tomó la base poblacional (BigCity).

```
plot1_censo <- ggplot(BigCity, aes(x = Income)) +  
  geom_histogram(aes(y = ..density..)) +  
  ylab("") +  
  ggtitle("Poblacional") +  
  theme_cepel() +  
  xlim(0, 2500)  
# Organizando la salida gráfica  
plot1_censo | plot1_Ponde | plot1_SinPonde
```

Histograma

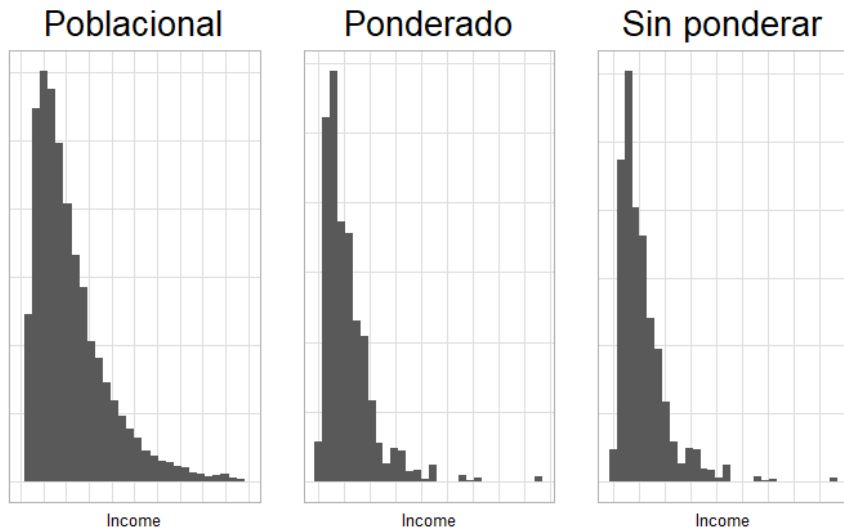


Figura 1: Histograma base para *Income*

Histogramas

Ahora, repetimos la secuencia de gráficos para la variable *Expenditure*

```
plot2_Ponde <- ggplot(  
  data = encuesta,  
  aes(x = Expenditure, weight = wk)  
) +  
  geom_histogram(aes(y = ..density..)) +  
  ylab("") +  
  ggtitle("Ponderado") +  
  theme_cepal()
```

Histogramas

```
plot2_SinPonde <- ggplot(data = encuesta,  
  aes(x = Expenditure)) +  
  geom_histogram(aes(y = ..density..)) +  
  ylab("") +  
  ggtitle("Sin ponderar") +  
  theme_cepal()
```

Histogramas

```
plot2_censo <- ggplot(BigCity, aes(x = Expenditure)) +  
  geom_histogram(aes(y = ..density..)) +  
  ylab("") +  
  ggtitle("Poblacional") +  
  theme_cepal() +  
  xlim(0, 1500)  
  
plot2_censo | plot2_Ponde | plot2_SinPonde
```


Histogramas

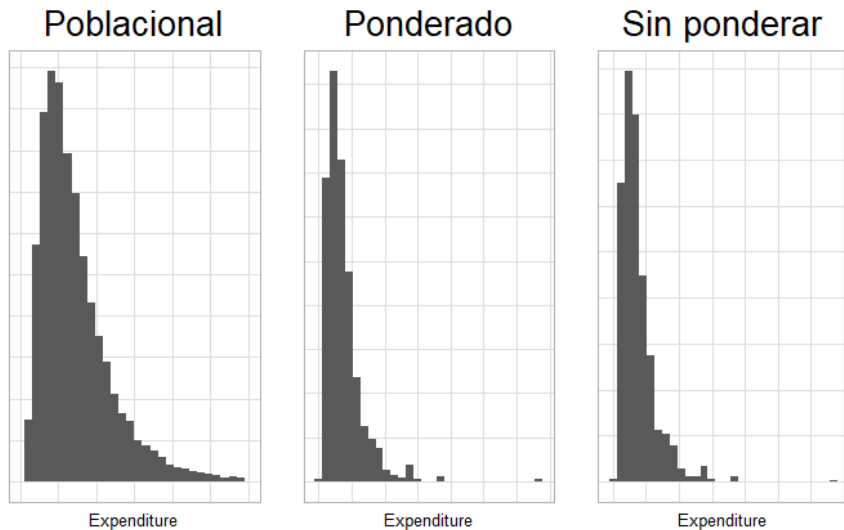


Figura 2: Histograma base para *Expenditure*

Histogramas por sub-grupos

Cuando el interés es realizar comparaciones entre dos o más agrupaciones, es posible hacer uso del parámetro `fill`, el cual “rellena” las barras del histograma con diferentes colores según sea el grupo.

```
plot3_Ponde <- ggplot(  
  encuesta,  
  aes(x = Income, weight = wk)  
) +  
  geom_histogram(  
    aes(y = ..density.., fill = Zone),  
    alpha = 0.5,  
    position = "identity" # Para que las barras no estén apiladas.  
) +  
  ylab("") +  
  ggtitle("Ponderado") +  
  theme_cepal()
```

Histogramas por sub-grupos

La sintaxis es homologa a la anterior, sin embargo, se retiro el parámetro weight.

```
plot3_SinPonde <- ggplot(encuesta, aes(x = Income)) +  
  geom_histogram(aes(y = ..density.., fill = Zone),  
    alpha = 0.5, position = "identity"  
  ) +  
  ggtitle("Sin ponderar") +  
  theme_cepal() +  
  ylab("")
```

Histogramas por sub-grupos

```
plot3_censo <- ggplot(BigCity, aes(x = Income)) +  
  geom_histogram(aes(y = ..density.., fill = Zone),  
    alpha = 0.5, position = "identity"  
  ) +  
  ggtitle("Poblacional") +  
  theme_cepal() +  
  xlim(0, 1500) +  
  ylab("")  
plot3_censo | plot3_Ponde | plot3_SinPonde
```

Histogramas por sub-grupos

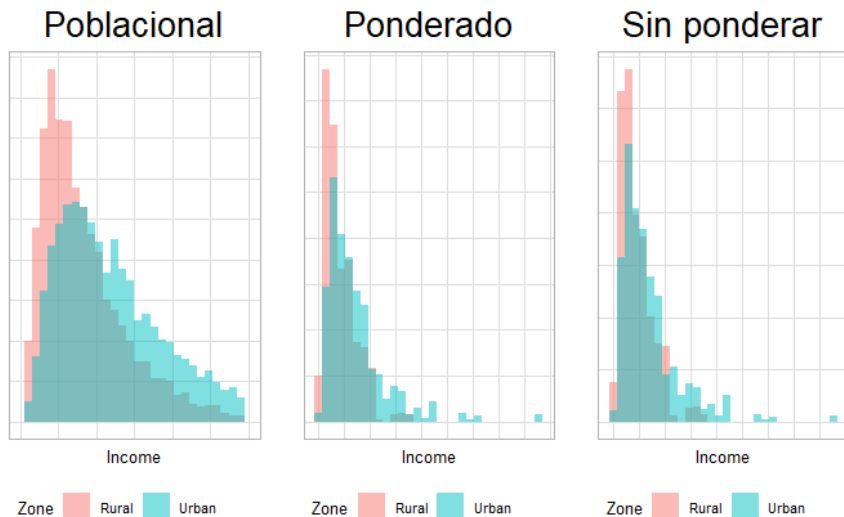


Figura 3: Histograma para *Income*

Histogramas por sub-grupos

Ahora, repetimos la secuencia de gráficos para la variable *Expenditure*

```
plot4_Ponde <- ggplot(  
  encuesta,  
  aes(x = Expenditure, weight = wk)  
) +  
  geom_histogram(aes(y = ..density.., fill = Zone),  
    alpha = 0.5, position = "identity"  
) +  
  ylab("") +  
  ggtitle("Ponderado") +  
  theme_cepal()
```

Histogramas por sub-grupos

```
plot4_SinPonde <- ggplot(  
  encuesta,  
  aes(x = Expenditure)  
) +  
  geom_histogram(aes(y = ..density.., fill = Zone),  
    alpha = 0.5, position = "identity"  
) +  
  ggtitle("Sin ponderar") +  
  theme_cepal() +  
  ylab("")
```

Histogramas por sub-grupos

```
plot4_censo <- ggplot(BigCity, aes(x = Expenditure)) +  
  geom_histogram(aes(y = ..density.., fill = Zone),  
    alpha = 0.5, position = "identity"  
  ) +  
  ggtitle("Poblacional") +  
  theme_cepal() +  
  xlim(0, 1500) +  
  ylab("")  
plot4_censo | plot4_Ponde | plot4_SinPonde
```


Histogramas por sub-grupos

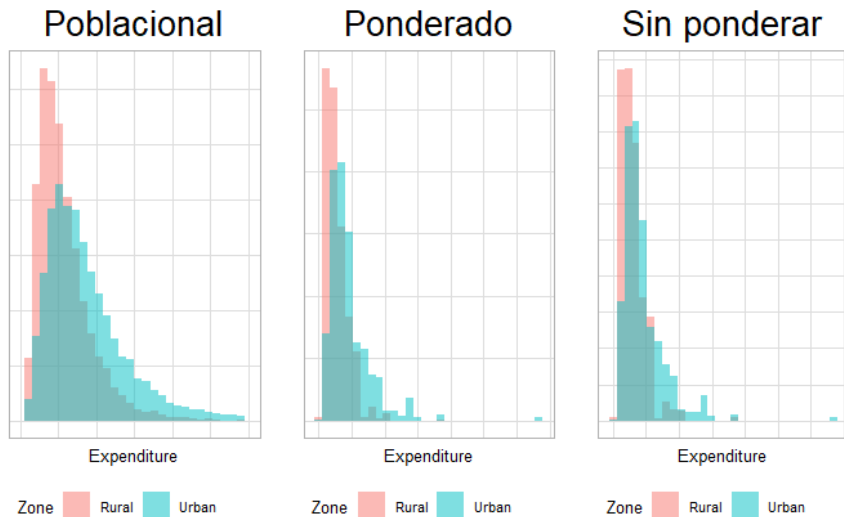


Figura 4: Histograma para *Expenditure*

Histogramas por sub-grupos

Ahora, repetimos la secuencia de gráficos para la variable *Income*, pero haremos el relleno por la variable *sexo*.

```
plot5_Ponde <- ggplot(  
  encuesta,  
  aes(x = Income, weight = wk)  
) +  
  geom_histogram(aes(y = ..density.., fill = Sex),  
    alpha = 0.5, position = "identity"  
) +  
  ylab("") +  
  ggtitle("Ponderado") +  
  theme_cepal()
```

Histogramas por sub-grupos

```
plot5_SinPonde <- ggplot(encuesta, aes(x = Income)) +  
  geom_histogram(aes(y = ..density.., fill = Sex),  
    alpha = 0.5, position = "identity"  
  ) +  
  ggtitle("Sin ponderar") +  
  theme_cepal() +  
  ylab("")
```

Histogramas por sub-grupos

```
plot5_censo <- ggplot(BigCity, aes(x = Income)) +  
  geom_histogram(aes(y = ..density.., fill = Sex),  
    alpha = 0.5, position = "identity"  
  ) +  
  ggtitle("Poblacional") +  
  theme_cepal() +  
  xlim(0, 1500) +  
  ylab("")  
plot5_censo | plot5_Ponde | plot5_SinPonde
```

Histogramas por sub-grupos

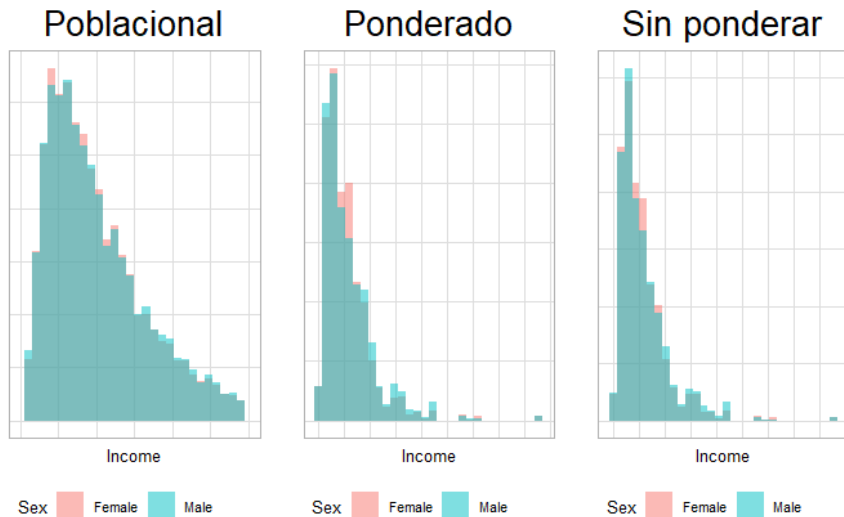


Figura 5: Histograma para *Income*

Histogramas por sub-grupos

Ahora, repetimos la secuencia de gráficos para la variable *Expenditure* y el relleno por la variable *sexo*.

```
plot6_Ponde <- ggplot(  
  encuesta,  
  aes(x = Expenditure, weight = wk)  
) +  
  geom_histogram(aes(y = ..density.., fill = Sex),  
    alpha = 0.5, position = "identity"  
) +  
  ylab("") +  
  ggtitle("Ponderado") +  
  theme_cepal()
```

Histogramas por sub-grupos

```
plot6_SinPonde <- ggplot(encuesta, aes(x = Expenditure)) +  
  geom_histogram(aes(y = ..density.., fill = Sex),  
    alpha = 0.5, position = "identity"  
  ) +  
  ggtitle("Sin ponderar") +  
  theme_cepal() +  
  ylab("")
```

Histogramas por sub-grupos

```
plot6_censo <- ggplot(BigCity, aes(x = Expenditure)) +  
  geom_histogram(aes(y = ..density.., fill = Sex),  
    alpha = 0.5, position = "identity"  
  ) +  
  ggtitle("Poblacional") +  
  theme_cepal() +  
  xlim(0, 1500) +  
  ylab("")  
plot6_censo | plot6_Ponde | plot6_SinPonde
```


Histogramas por sub-grupos

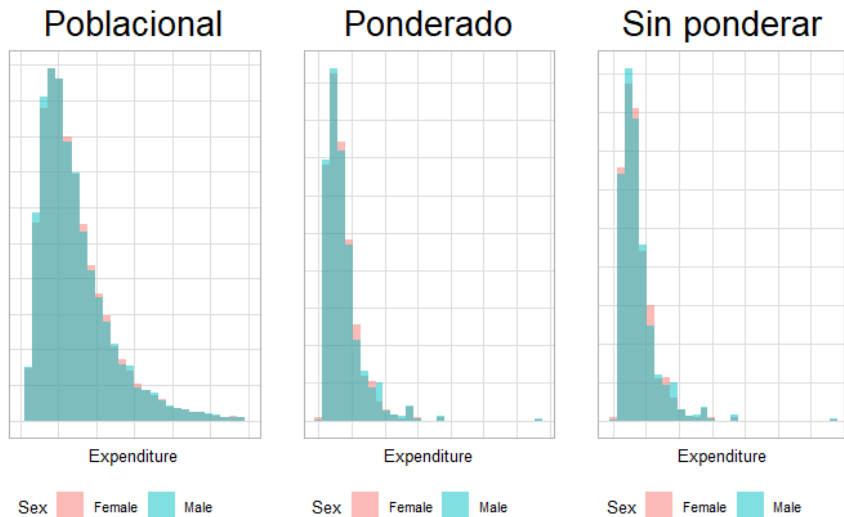


Figura 6: Histograma para *Expenditure*

Agregando densidad

Dadas las cualidades de la librería ggplot2, podemos agregar nuevas capas a la gráfica. Por ejemplo, la densidad con la función `geom_density` e incorporamos el parámetro `alpha` que regula la transparencia del relleno.

```
plot1_Ponde + geom_density(fill = "blue", alpha = 0.3) |  
plot2_Ponde + geom_density(fill = "blue", alpha = 0.3)
```

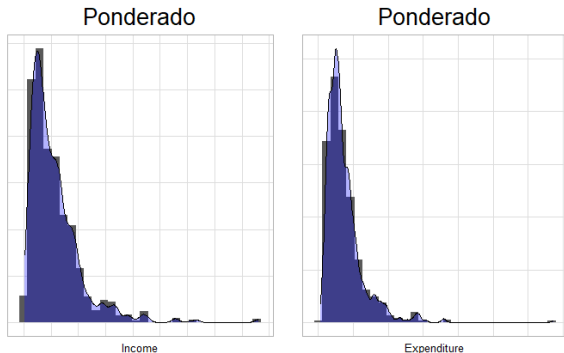


Figura 7: Desnsidad agregada al histograma

Agregando densidad

Al hacer `aes(fill = Zone)` permite que la densidad sea agregada para cada una de las agrupaciones.

```
plot3_Ponde + geom_density(aes(fill = Zone), alpha = 0.3) |  
plot4_Ponde + geom_density(aes(fill = Zone), alpha = 0.3)
```

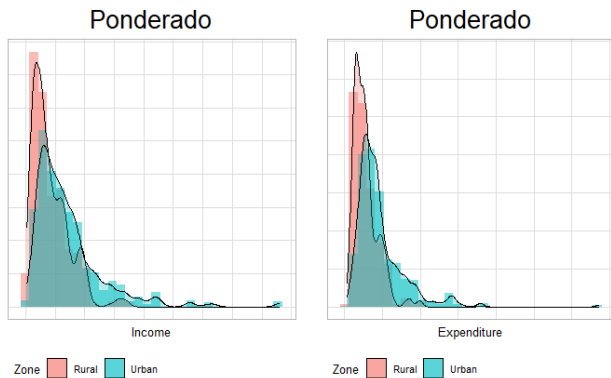


Figura 8: Desnsidad para agregada al histograma por zona

Agregando densidad

En está oportunidad se agrega la densidad por sexo

```
plot5_Ponde + geom_density(aes(fill = Sex), alpha = 0.3) |  
plot6_Ponde + geom_density(aes(fill = Sex), alpha = 0.3)
```

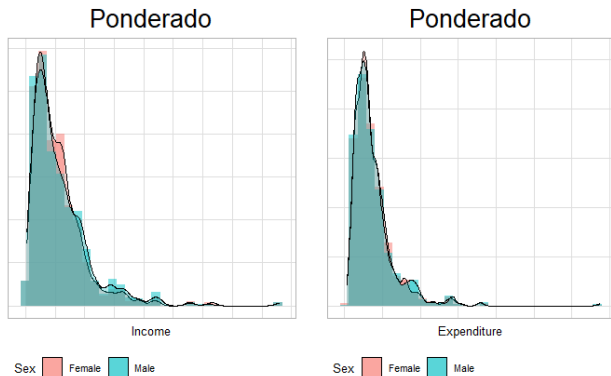


Figura 9: Desnsidad agregada al histograma por sexo

Boxplot

- ▶ El boxplot es un gráfico ampliamente utilizado en estadísticas que fue presentado por John Tukey en 1977. Proporciona una representación resumida de un conjunto de datos utilizando cinco números clave.
- ▶ El boxplot consiste en un rectángulo llamado “caja” y dos segmentos llamados “bigotes”. Este gráfico muestra información sobre la relación entre los cuartiles (Q_1 , Q_2 o mediana y Q_3) y los valores mínimo y máximo del conjunto de datos, la presencia de valores atípicos y la simetría de la distribución.
- ▶ Para crear boxplots en R utilizando ggplot2, se emplea la función `geom_boxplot`.

Boxplot

Otro gráfico que podemos hacer son los diagramas de caja, para esto deben emplear la función `geom_boxplot`.

```
plot7_Ponde <- ggplot( encuesta, aes(x = Income, weight = wk)) +  
  geom_boxplot() + ggtitle("Ponderado") +  
  coord_flip() + theme_cepal()
```

```
plot8_Ponde <- ggplot( encuesta,  
  aes(x = Expenditure, weight = wk)  
) + geom_boxplot() + ggtitle("Ponderado") + coord_flip() +  
  theme_cepal()
```

Boxplot

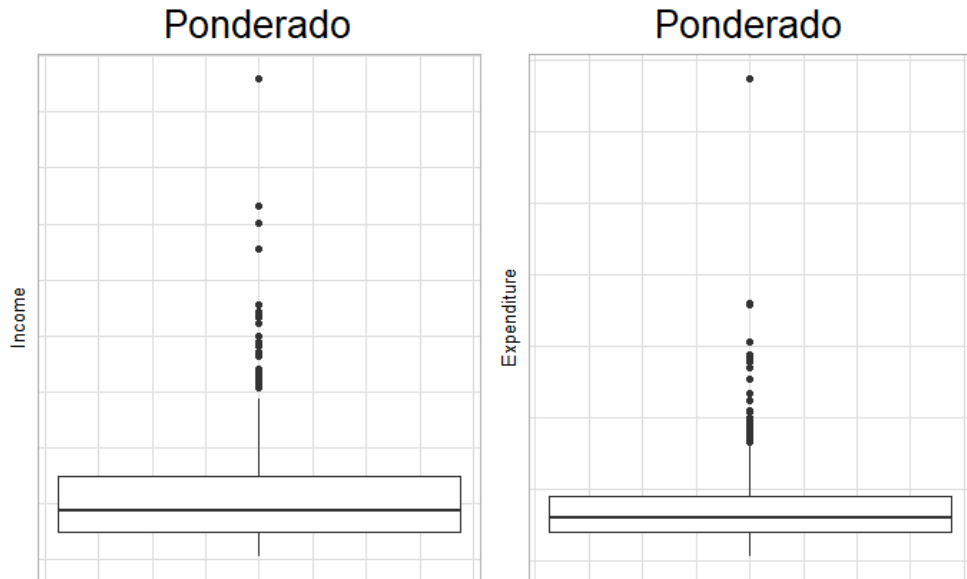


Figura 10: Boxplot para el ingreso y gasto

Boxplot

Estos diagramas también permiten la comparación entre dos o más niveles de agrupamiento.

```
plot9_Ponde <- ggplot(encuesta, aes(x = Income, weight = wk)) +  
  geom_boxplot(aes(fill = Zone)) + ggtitle("Ponderado") +  
  coord_flip() + theme_cepal()
```

```
plot10_Ponde <- ggplot( encuesta,  
  aes(x = Expenditure, weight = wk) ) +  
  geom_boxplot(aes(fill = Zone)) + ggtitle("Ponderado") +  
  coord_flip() + theme_cepal()
```


Boxplot

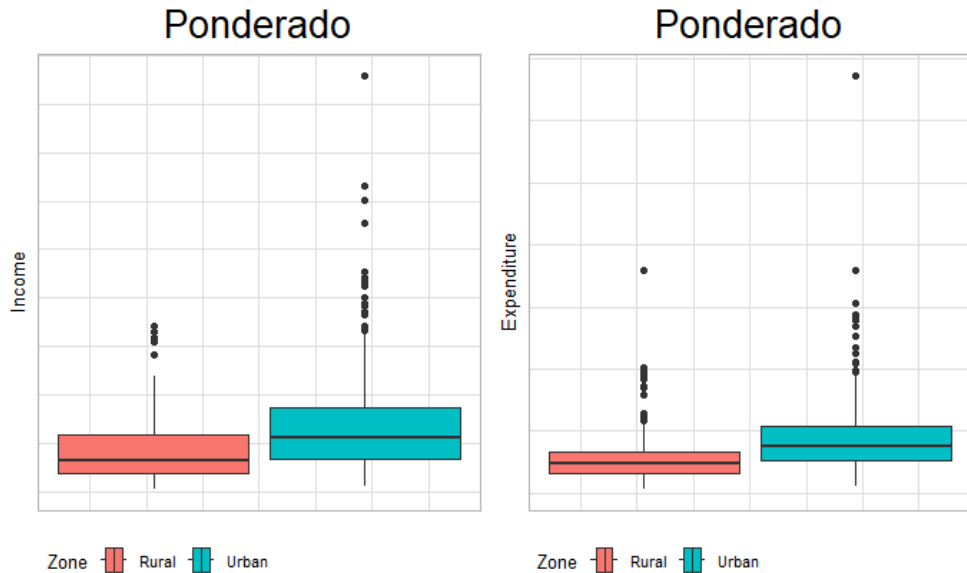


Figura 11: Boxplot para el ingreso y gasto por zona

Boxplot

Ahora, si desean personalizar los colores del relleno debe hacer uso de la función `scale_fill_manual`.

```
colorZona <- c(Urban = "#48C9B0", Rural = "#117864")  
plot9_Ponde + scale_fill_manual(values = colorZona) |  
  plot10_Ponde + scale_fill_manual(values = colorZona)
```

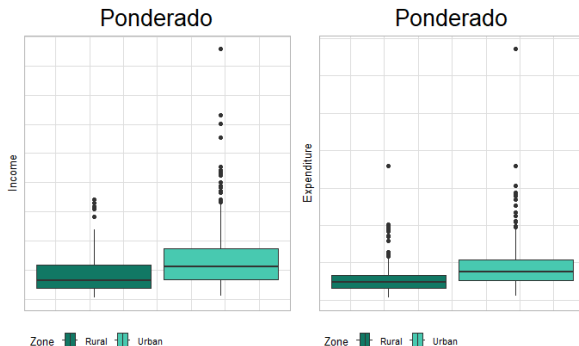


Figura 12: Boxplot para el ingreso y gasto por zona

Boxplot

Comparando los ingresos y gastos por sexo.

```
plot11_Ponde <- ggplot( encuesta,  
  aes(x = Income, weight = wk)  
) + geom_boxplot(aes(fill = Sex)) +  
  ggtitle("Ponderado") + coord_flip() +  
  theme_cepal()
```

```
plot12_Ponde <- ggplot( encuesta,  
  aes(x = Expenditure, weight = wk)  
) + geom_boxplot(aes(fill = Sex)) +  
  ggtitle("Ponderado") + coord_flip() +  
  theme_cepal()
```

Boxplot

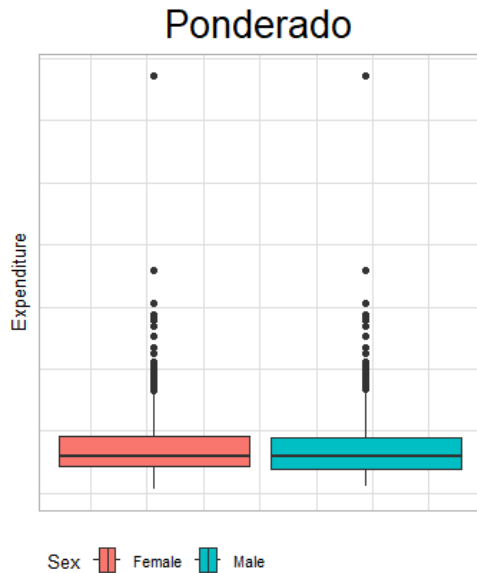
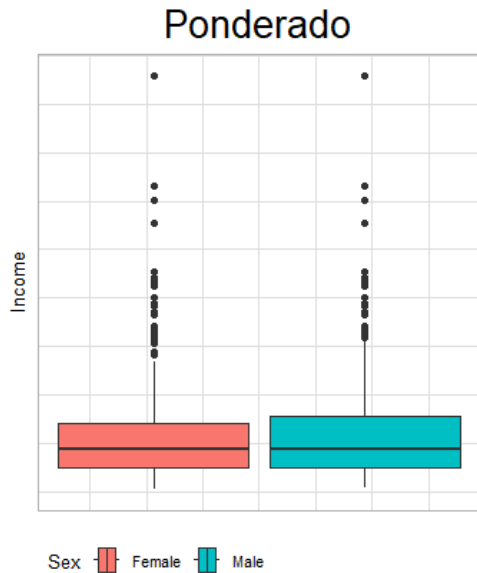


Figura 13: Boxplot para el ingreso y gasto por sexo

Boxplot

Definiendo el color del relleno para hombres y mujeres.

```
colorSex <- c(Male = "#5DADE2", Female = "#2874A6")  
plot11_Ponde + scale_fill_manual(values = colorSex) |  
  plot12_Ponde + scale_fill_manual(values = colorSex)
```

Boxplot

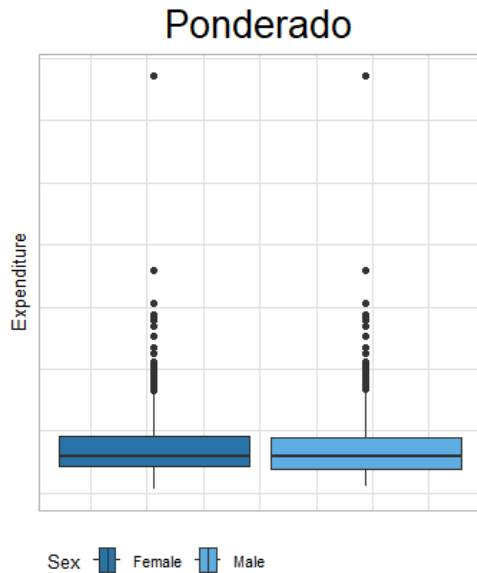
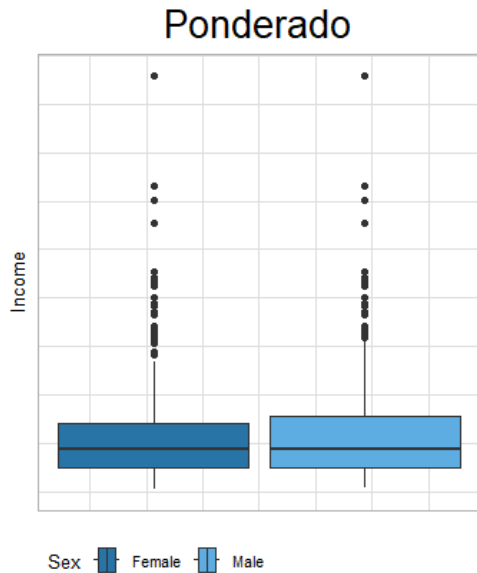


Figura 14: Boxplot para el ingreso y gasto por sexo

Boxplot

Realizando la comparación para más de dos categorías.

```
plot13_Ponde <- ggplot( encuesta,  
  aes(x = Income, weight = wk)) +  
  geom_boxplot(aes(fill = Region)) +  
  ggtitle("Ponderado") + coord_flip() +  
  theme_cepal()
```

```
plot14_Ponde <- ggplot( encuesta,  
  aes(x = Expenditure, weight = wk)) +  
  geom_boxplot(aes(fill = Region)) +  
  ggtitle("Ponderado") + coord_flip() +  
  theme_cepal()
```

Boxplot

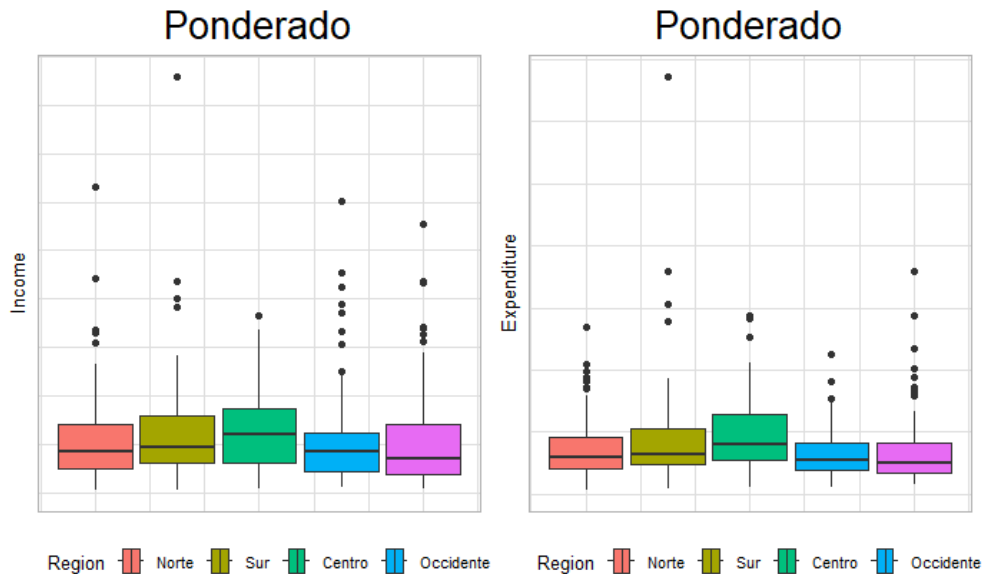
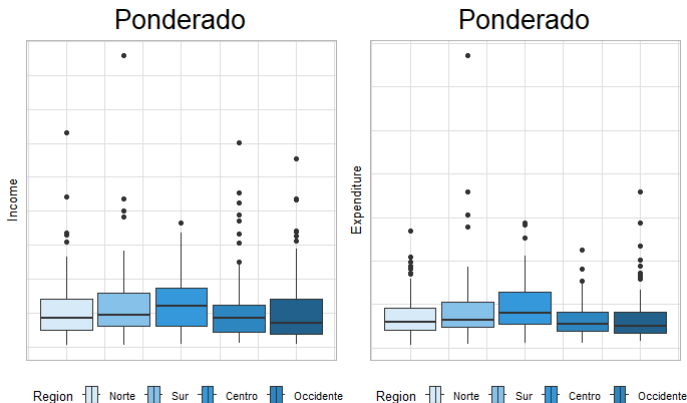


Figura 15: Boxplot para el ingreso y gasto por región

Boxplot

Personalizando los coles cuando hay más de dos categorías.

```
colorRegion <- c(  
  Norte = "#D6EAF8", Sur = "#85C1E9",  
  Centro = "#3498DB", Occidente = "#2E86C1", Oriente = "#21618C"  
)  
p <- plot13_Ponde + scale_fill_manual(values = colorRegion) |  
plot14_Ponde + scale_fill_manual(values = colorRegion)
```



Boxplot

La función `geom_boxplot` permite realizar comparaciones con más de dos variables al tiempo. A continuación se compara los ingresos por sexo en las diferentes zonas.

```
plot15_Ponde <-  
  ggplot(  
    encuesta,  
    aes(x = Income, y = Zone, weight = wk)  
  ) +  
  geom_boxplot(aes(fill = Sex)) +  
  ggtitle("Ponderado") +  
  scale_fill_manual(values = colorSex) +  
  coord_flip()
```

Boxplot

De forma análoga podemos realizar la comparación de los gastos por sexo en las diferentes zonas.

```
plot16_Ponde <-  
  ggplot(  
    encuesta,  
    aes(x = Expenditure, y = Zone, weight = wk)  
  ) +  
  geom_boxplot(aes(fill = Sex)) +  
  ggtitle("Ponderado") +  
  scale_fill_manual(values = colorSex) +  
  coord_flip()
```

Boxplot

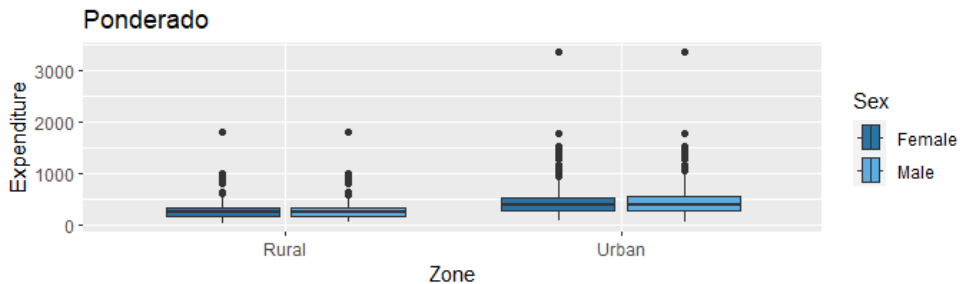
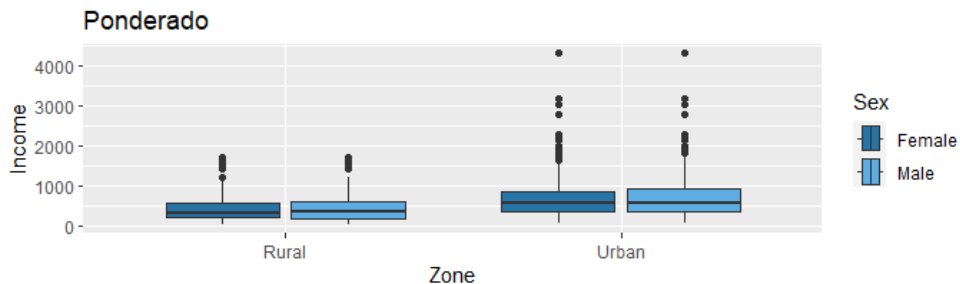


Figura 17: Boxplot para el ingreso y gasto por sexo Y zona

Boxplot

Podemos extender las comparaciones variables que tienen más de dos categorías.

```
plot17_Ponde <-  
  ggplot(  
    encuesta,  
    aes(x = Income, y = Region, weight = wk)  
  ) +  
  geom_boxplot(aes(fill = Sex)) +  
  ggtitle("Ponderado") +  
  scale_fill_manual(values = colorSex) +  
  coord_flip()
```

Boxplot

```
plot18_Ponde <-  
  ggplot(  
    encuesta,  
    aes(  
      x = Expenditure,  
      y = Region, weight = wk  
    )  
  ) +  
  geom_boxplot(aes(fill = Sex)) +  
  ggtitle("Ponderado") +  
  scale_fill_manual(values = colorSex) +  
  coord_flip()  
  
plot17_Ponde / plot18_Ponde
```

Boxplot

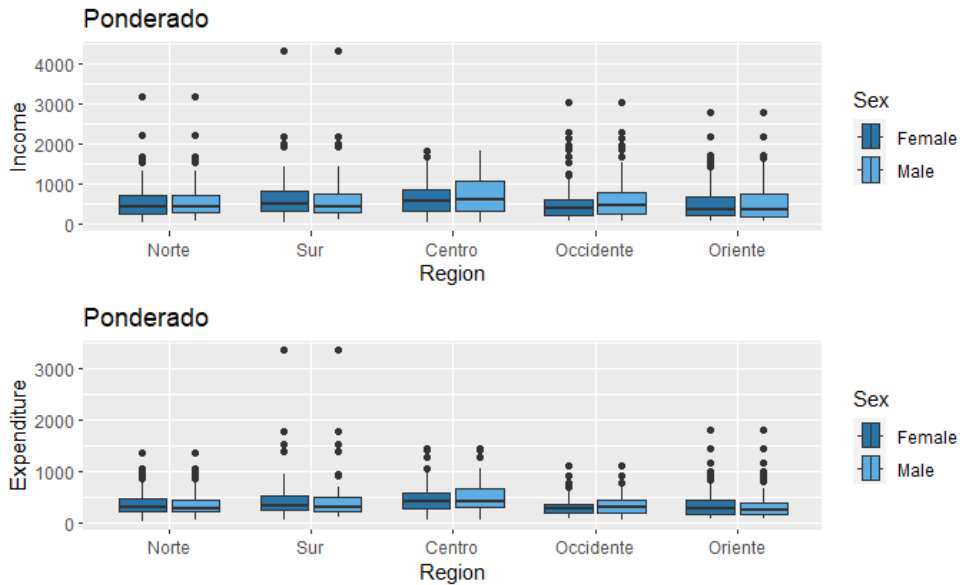


Figura 18: Boxplot para el ingreso y gasto por sexo Y zona

Scaterplot

Introducción

- ▶ Un diagrama de dispersión es una representación gráfica que muestra observaciones como puntos en un plano.
- ▶ Cada punto en el diagrama de dispersión se posiciona de acuerdo a los valores de dos variables.
- ▶ Los puntos pueden tener atributos como tamaño, color y forma, conocidos como estéticas.
- ▶ En R, se utiliza `geom_point` para crear un diagrama de dispersión.
- ▶ Es posible asignar estéticas a variables o establecer valores constantes para ellas en el gráfico.
- ▶ Se pueden utilizar diagramas de dispersión para visualizar la relación entre dos variables, identificar patrones o tendencias, y explorar datos.

Scaterplot

Para ejemplificar el uso de esta función, se graficarán las variables ingresos y gastos como se muestra a continuación:

```
plot19_Ponde <-  
  ggplot(  
    encuesta,  
    aes(  
      y = Income,  
      x = Expenditure,  
      weight = wk  
    )  
  ) +  
  geom_point() +  
  theme_cepal()
```

Scaterplot

Note, que este caso el parámetro `weight` no esta aportando información visual al gráfico.

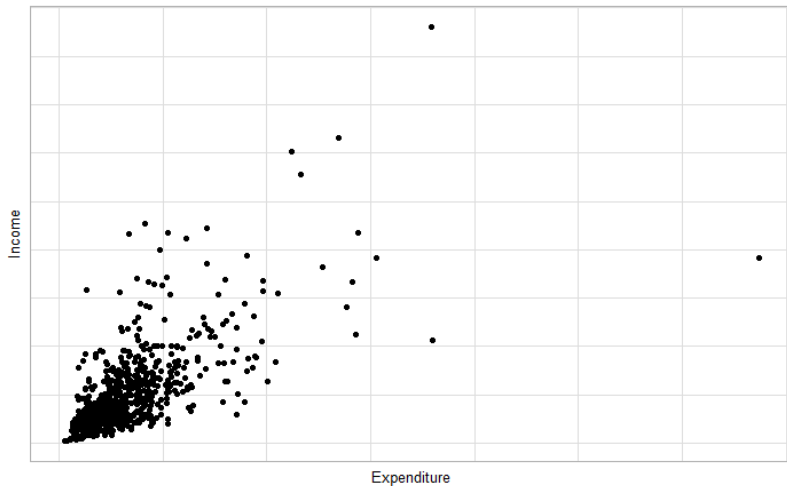


Figura 19: Diagramas de dispersión del ingreso frente al gasto

Scaterplot

El parámetro `weight` lo podemos usar controlar el tamaño de los puntos de esa forma tener un mejor panorama del comportamiento de la muestra.

```
plot20_Ponde <-  
  ggplot(  
    encuesta,  
    aes(y = Income, x = Expenditure)  
  ) +  
  geom_point(aes(size = wk), alpha = 0.3) +  
  theme_cepal()
```

Scaterplot

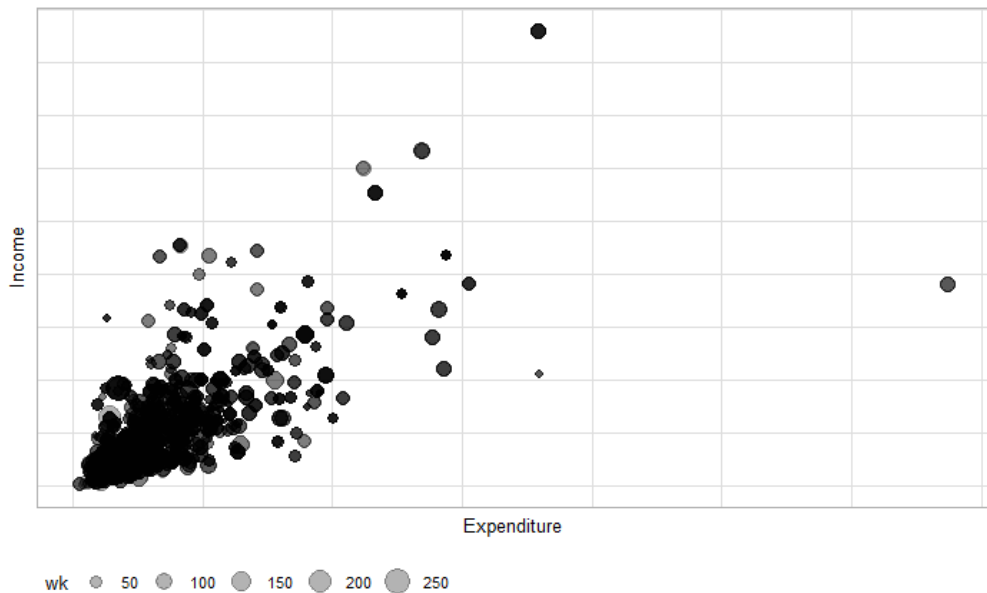


Figura 20: Diagramas de dispersión del ingreso frente al gasto con pesos

Scatterplot

Otra forma de usar la variable `wk`, es asignar la intensidad del color según el valor de la variable.

```
plot21_Ponde <-  
  ggplot(  
    encuesta,  
    aes(y = Income, x = Expenditure)  
  ) +  
  geom_point(aes(col = wk), alpha = 0.3) +  
  theme_cepel()
```

Scaterplot

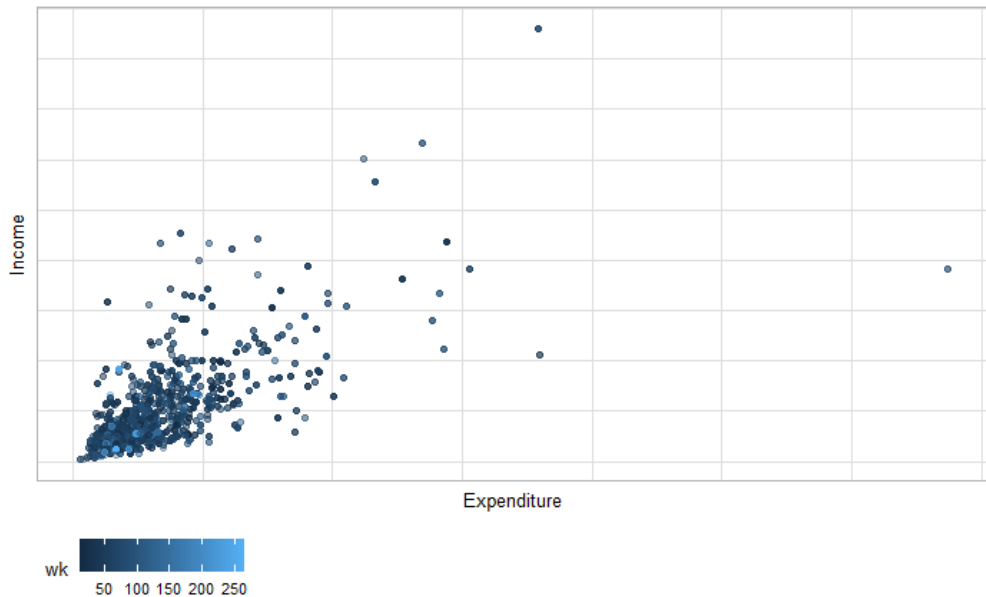


Figura 21: Diagramas de dispersión del ingreso frente al gasto con pesos

Scaterplot

Podemos extender las bondades de los gráfico de ggplot2 para obtener mayor información de las muestra. Por ejemplo, agrupar los datos por Zona, para lograr esto se introduce el parámetro shape.

```
plot22_Ponde <-  
  ggplot(  
    encuesta,  
    aes(  
      y = Income, x = Expenditure,  
      shape = Zone) # Formas por zona  
  ) + geom_point(aes(  
    size = wk, color = Zone  
  ), alpha = 0.3) +  
  labs(size = "Peso") +  
  scale_color_manual(values = colorZona) +  
  theme_cepal()
```


Scaterplot

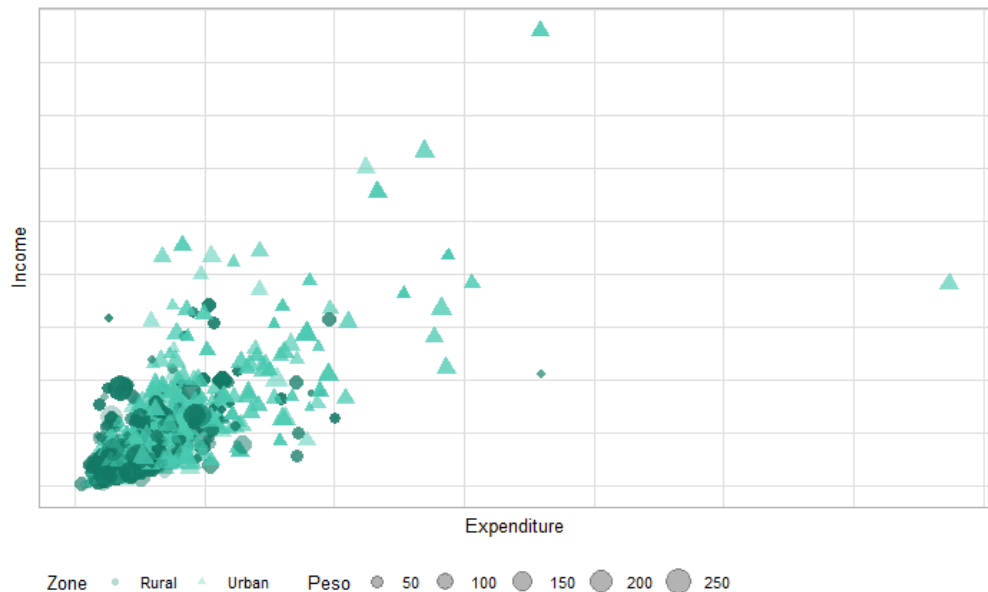


Figura 22: Diagramas de dispersión del ingreso frente al gasto con pesos por zona

Scaterplot

De forma similar podemos obtener el resultado por sexo.

```
plot23_Ponde <-  
  ggplot( encuesta, aes(  
    y = Income,  
    x = Expenditure,  
    shape = Sex ) ) +  
  geom_point(aes( size = wk,  
    color = Sex), alpha = 0.3 ) +  
  labs(size = "Peso") +  
  scale_color_manual(values = colorSex) +  
  theme_cepal()
```

Scaterplot

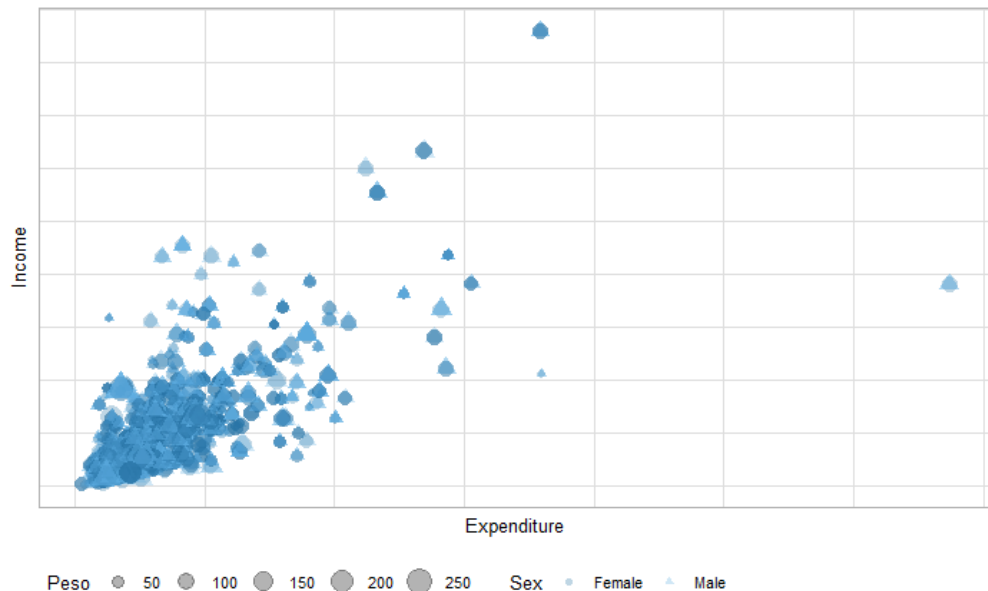


Figura 23: Diagramas de dispersión del ingreso frente al gasto con pesos por sexo

Scaterplot

Un resultado equivalente se obtiene por región.

```
plot24_Ponde <-  
  ggplot(encuesta,  
    aes(y = Income, x = Expenditure, shape = Region ) ) +  
  geom_point(aes( size = wk, color = Region),  
    alpha = 0.3 ) +  labs(size = "Peso") +  
  scale_color_manual(values = colorRegion) +  
  theme_cepal()
```

Scaterplot

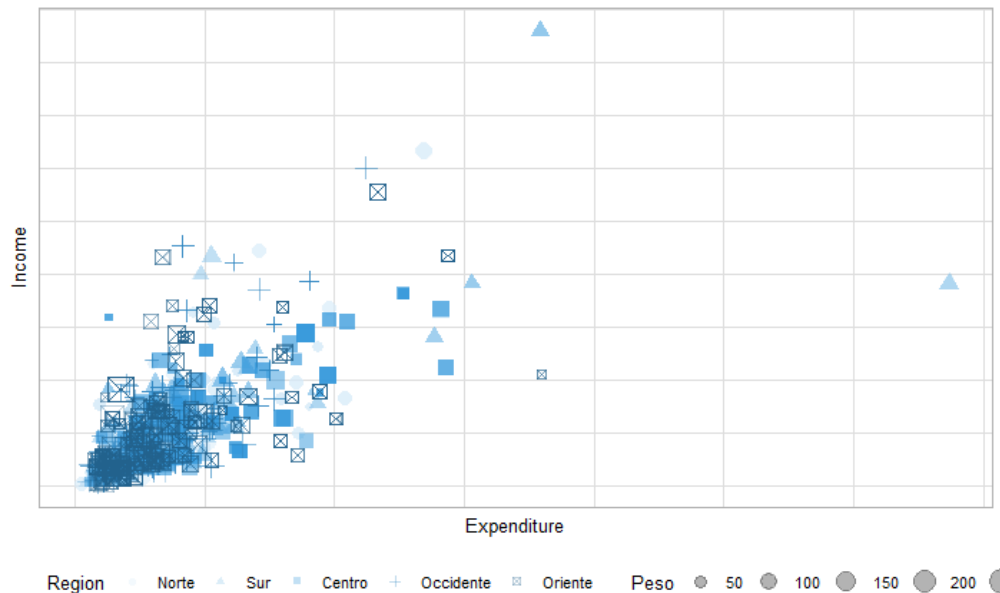


Figura 24: Diagramas de dispersión del ingreso frente al gasto con pesos por sexo

Diagrama de barras para variables categoricas

Diagrama de barras

Para realizar estos gráfico un primer paso es realizar las estimaciones puntuales.

```
(tamano_zona <- diseno %>%  
  group_by(Zone) %>%  
  summarise(  
    Nd = survey_total(vartype = c("se", "ci"))  
  ))
```

Zone	Nd	Nd_se	Nd_low	Nd_upp
Rural	72102	3062	66039	78165
Urban	78164	2847	72526	83802

Diagrama de barras

```
plot25_Ponde <- ggplot(  
  data = tamaño_zona,           # Fuente de los datos  
  aes(x = Zone,                 # Valores en el eje x  
      y = Nd,                   # Altura de la barras  
      ymax = Nd_upp,           # Limite superior del IC  
      ymin = Nd_low,           # Limite inferior del IC  
      fill = Zone               # Color del relleno  
) + geom_bar( stat = "identity", # Valor incluido en la tabla  
              position = "dodge") +  
  geom_errorbar(                # Gráfica del IC.  
    position = position_dodge(width = 0.9),  
    width = 0.3  
) + theme_bw()
```


Diagrama de barras

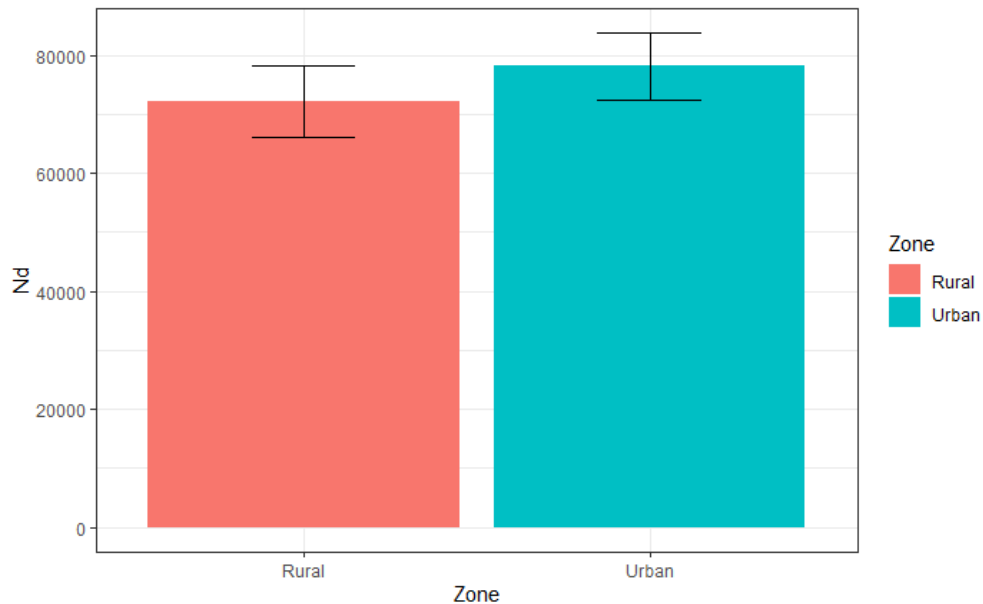


Figura 25: Diagramas de barras total de personas estimado por zona

Diagrama de barras

Como se ha visto en los gráficos anteriores podemos extender a muchas categorías.

```
(tamano_pobreza <- disenio %>%  
  group_by(Poverty) %>%  
  summarise(  
    Nd = survey_total(vartype = c("se", "ci"))  
  ))
```

Poverty	Nd	Nd_se	Nd_low	Nd_upp
NotPoor	91398	4395	82696	100101
Extreme	21519	4949	11719	31319
Relative	37349	3695	30032	44666

Diagrama de barras

El gráfico se obtiene con una sintaxis homologa a la anterior.

```
plot26_Ponde <- ggplot(  
  data = tamano_pobreza,  
  aes( x = Poverty, y = Nd,  
        ymax = Nd_upp, ymin = Nd_low,  
        fill = Poverty ) ) +  
  geom_bar(stat = "identity", position = "dodge") +  
  geom_errorbar(  
    position = position_dodge(width = 0.9),  
    width = 0.3  
  ) + theme_bw()
```

Diagrama de barras

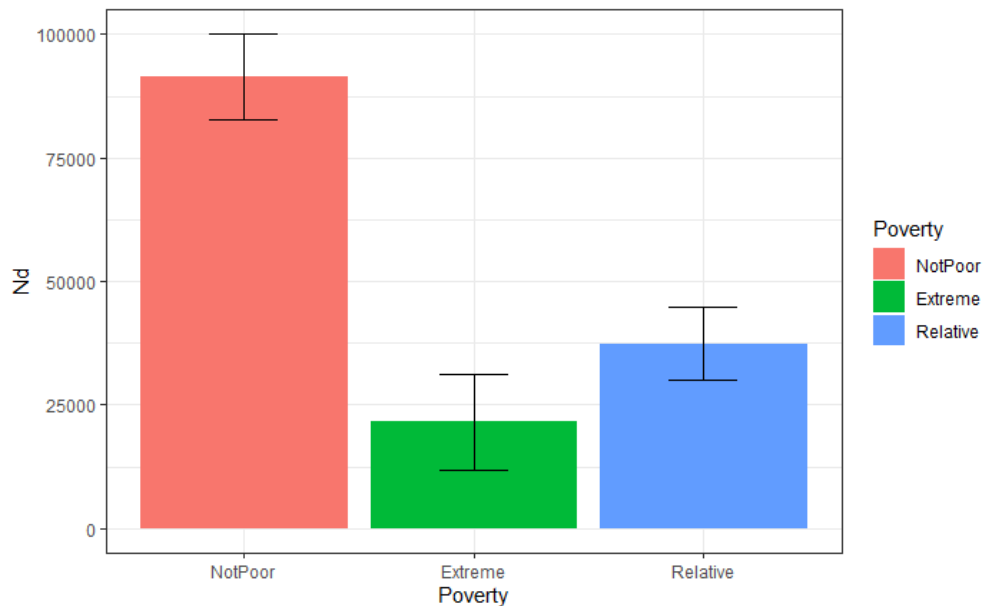


Figura 26: Diagramas de barras del total de personas condición de pobreza estimado

Diagrama de barras

De forma similar a los gráficos de Caja es posible realizar comparaciones entre más dos variables.

```
tamano_ocupacion_pobreza <- disenio %>%  
  group_by(desempleo, Poverty) %>%  
  summarise(  
    Nd = survey_total(vartype = c("se", "ci"))  
  ) %>%  as.data.frame() %>%  
  mutate(  
    desempleo = ifelse(is.na(desempleo),  
                        "Ninos",desempleo))
```

Diagrama de barras

El gráfico para la tabla anterior queda de la siguiente forma.

```
plot27_Ponde <-  
  ggplot(  
    data = tamano_ocupacion_pobreza,  
    aes( x = Poverty, y = Nd,  
          ymax = Nd_upp, ymin = Nd_low,  
          fill = as.factor(desempleo) ) ) +  
  geom_bar(stat = "identity", position = "dodge") +  
  geom_errorbar( position = position_dodge(width = 0.9),  
                 width = 0.3 ) + theme_bw() + labs(fill = "Desempleo")  
plot27_Ponde
```

Diagrama de barras

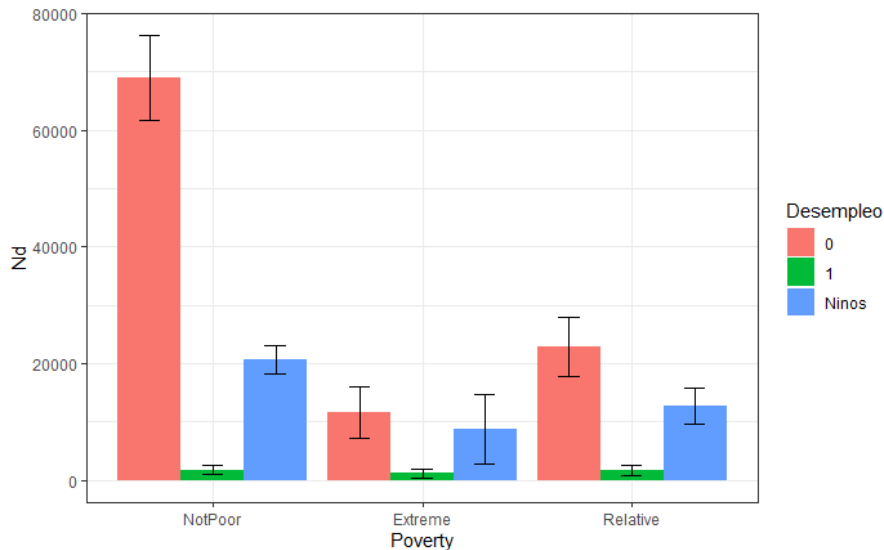


Figura 27: Diagramas de barras del total de personas condición de pobreza y estado de ocupación estimado

Diagrama de barras

En estos gráficos podemos presentar proporciones por variables.

```
(prop_ZonaH_Pobreza <- sub_Hombre %>%  
  group_by(Zone, Poverty) %>%  
  summarise(  
    prop = survey_prop(  
      vartype = c("se", "ci")  
    )  
  ) %>%  
  data.frame())
```


Diagrama de barras

Después de tener la tabla con los valores a presentar el gráfico se realiza con la siguiente sintaxis.

```
plot28_Ponde <- ggplot(  
  data = prop_ZonaH_Pobreza,  
  aes(  
    x = Poverty, y = prop,  
    ymax = prop_upp, ymin = prop_low,  
    fill = Zone  
  )) +  
  geom_bar(stat = "identity", position = "dodge") +  
  geom_errorbar(  
    position = position_dodge(width = 0.9),  
    width = 0.3  
  ) + scale_fill_manual(values = colorZona) +  
  theme_bw()
```

Diagrama de barras

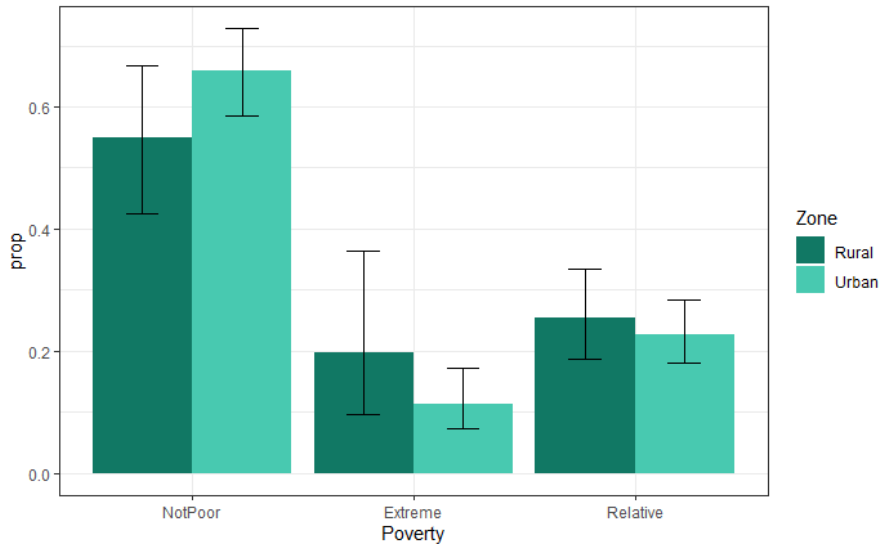


Figura 28: Diagramas de barras del total de personas condición de pobreza y zona estimado

Diagrama de barras

Proporción de hombres en condición de pobreza por región

```
prop_RegionH_Pobreza <- sub_Hombre %>%  
  group_by(Region, pobreza) %>%  
  summarise(  
    prop = survey_prop(vartype = c("se", "ci"))  
  ) %>%  
  data.frame()
```

Diagrama de barras

```
plot29_Ponde <- ggplot(  
  data = prop_RegionH_Pobreza,  
  aes(  
    x = Region, y = prop,  
    ymax = prop_upp, ymin = prop_low,  
    fill = as.factor(pobreza) ) ) +  
geom_bar(stat = "identity", position = "dodge") +  
geom_errorbar(  
  position = position_dodge(width = 0.9),  
  width = 0.3  
) + labs(fill = "Pobreza") +  
theme_bw()
```

Diagrama de barras

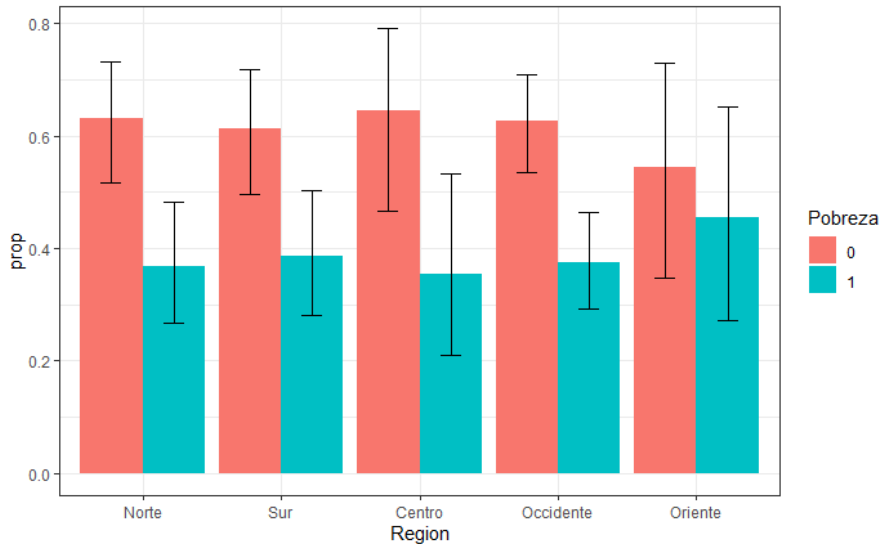


Figura 29: Diagramas de barras del total de personas condición de pobreza y región estimado

Creando mapas

Introducción

- ▶ Los mapas son herramientas poderosas para visualizar datos, especialmente para indicadores sociales y demográficos.
- ▶ Para crear mapas en R, se requiere información geoespacial que contenga coordenadas o delimitaciones geográficas.
- ▶ Sitios web como <http://www.diva-gis.org/gdata> ofrecen bases de datos gratuitas con vectores geográficos.
- ▶ Estos conjuntos de datos contienen observaciones de longitud y latitud que permiten representar puntos y polígonos en un mapa.
- ▶ En R, existen varias bibliotecas para crear mapas, incluyendo `tmap` y `ggplot2`.

Mapas con tmap.

Para realizar el mapa hay que contar con el archivo de *shepefile*

```
library(sf)
library(tmap)
shapeBigCity <- read_sf("/Data/shapeBigCity/BigCity.shp")
```

La forma más simple de crear el mapa es con la sintaxis.

```
tm_shape(shapeBigCity) + # shepefile
  tm_polygons(col = "Region") # Variable de interés.
```


Mapas con tmap.

El mapa resultante es:

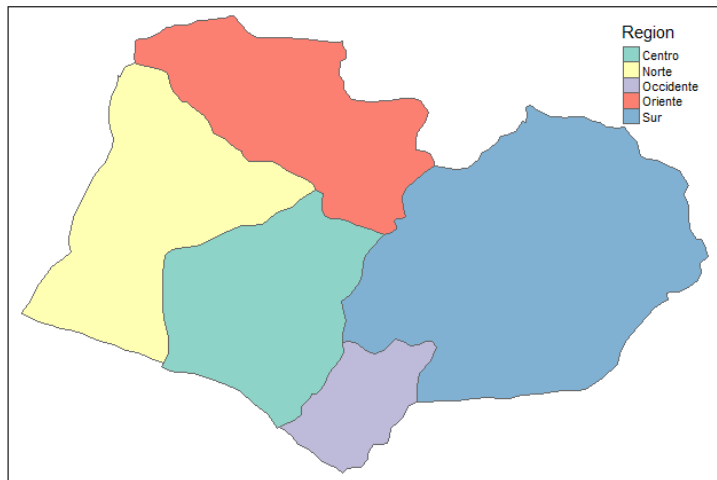


Figura 30: Mapa por región

Mapas con tmap.

```
brks <- c(0, .2, .4, .6, 0.8, 1)
shape_temp <- tm_shape(
  shapeBigCity %>%      # shapefile
  left_join(            # Agregando una variable
    prop_RegionH_Pobreza %>%
      filter(pobreza == 1), # Filtrando el nivel de interés.
    by = "Region"
  ))

shape_temp + tm_polygons(
  "prop",              # Nombre de la columna
  breaks = brks,       # Puntos de corte
  title = "pobreza",   # Titulo del labels.
  palette = "YlOrRd"    # Paleta y dirección de colores
)
```

Mapas con tmap.

El mapa resultante es:

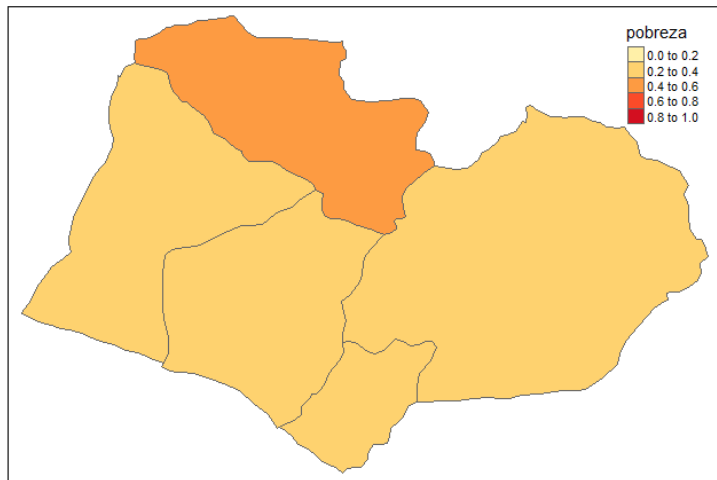


Figura 31: Mapa por región de la pobreza

Estimación del ingreso medio por región.

```
(prom_region <- svyby(~Income, ~Region, diseno,  
  svymean,  
  na.rm = T, covmat = TRUE,  
  vartype = c("cv")  
))
```

	Region	Income	cv
Norte	Norte	552.4	0.1002
Sur	Sur	625.8	0.0997
Centro	Centro	650.8	0.0945
Occidente	Occidente	517.0	0.0894
Oriente	Oriente	541.8	0.1323

Mapas con tmap.

```
brks <- c(0, 0.1, 1)
shape_temp <- tm_shape(
  shapeBigCity %>%
    left_join(
      prom_region,
      by = "Region"
    )
)

shape_temp + tm_polygons(
  "cv",
  breaks = brks,
  title = "cv",
  palette = c("#FFFFFF", "#000000"),
) + tm_layout(asp = 0)
```

Mapas con tmap.

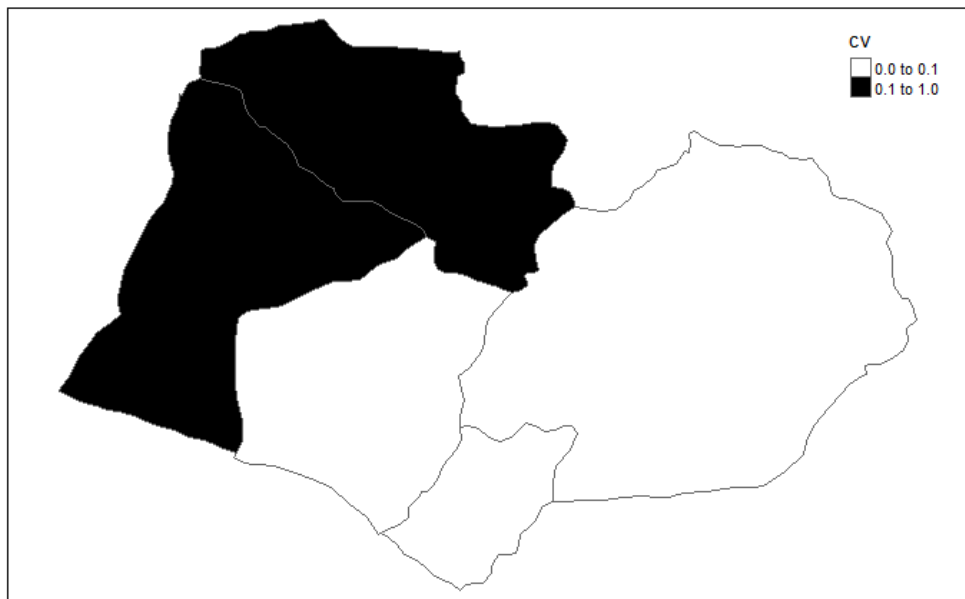


Figura 32: Mapa por región del Coeficiente de variación del ingreso medio

Estimación de la pobreza por región.

Estimación de la pobreza por región y sexo

```
prom_region_Sex <- diseno %>%  
  group_by(Region, Zone, Sex, pobreza) %>%  
  summarise(prop = survey_mean(vartype = "cv")) %>%  
  filter(pobreza == 1, Zone == "Rural", Sex == "Female")  
data.frame(prom_region_Sex)
```

Region	Zone	Sex	pobreza	prop	prop_cv
Norte	Rural	Female	1	0.3138	0.2296
Sur	Rural	Female	1	0.3027	0.1423
Centro	Rural	Female	1	0.4244	0.4633
Occidente	Rural	Female	1	0.5076	0.2024
Oriente	Rural	Female	1	0.5477	0.2297

Mapas con tmap.

```
shape_temp <- tm_shape(  
  shapeBigCity %>%  
    left_join(  
      prom_region_Sex,  
      by = "Region"  
    )  
)  
  
shape_temp + tm_polygons(  
  "prop",  
  title = "Pobreza",  
) + tm_layout(asp = 0)
```


Mapas con tmap.

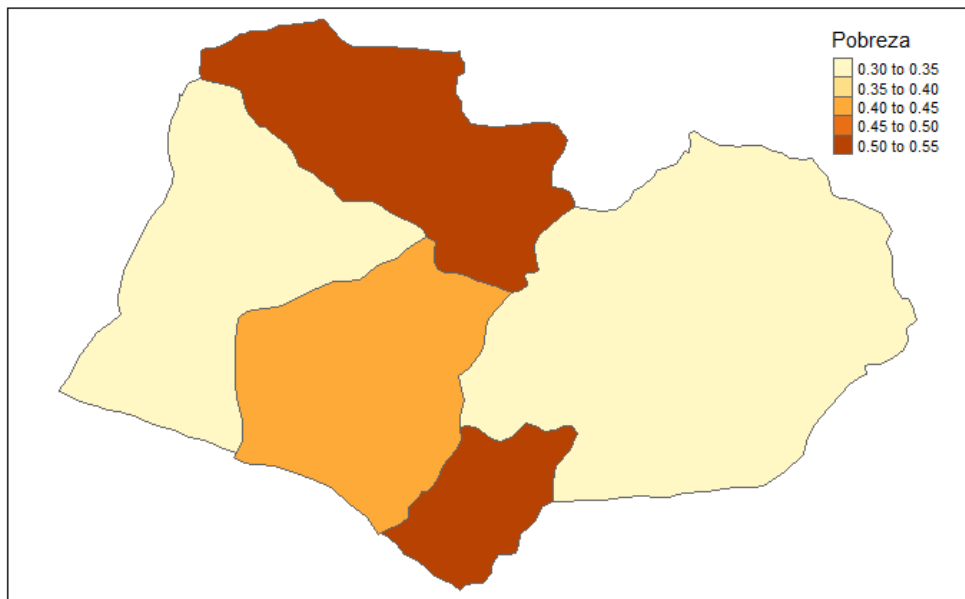


Figura 33: Mapa por región de la pobreza monetaria

Mapas con tmap.

```
shape_temp + tm_polygons(  
  "prop_cv",  
  title = "cv",  
  palette = c("#FFFFFF", "#000000"),  
  breaks = c(0, 0.2, 1)  
) + tm_layout(asp = 0)
```

Mapas con tmap.



Figura 34: Mapa por región del CV para la estimación de la pobreza monetaria

Mapas con ggplot

```
library(biscale)
library(cowplot)
temp_shape <- shapeBigCity %>%
  left_join(
    prom_region_Sex,
    by = "Region"
  )
k <- 3
datos.RM.bi <- bi_class(temp_shape,
  y = prop, x = prop_cv, dim = k,
  style = "fisher"
)
```

Mapas con ggplot

```
map.RM <- ggplot() +  
  geom_sf(  
    data = datos.RM.bi,  
    aes(fill = bi_class, geometry = geometry),  
    colour = "white", size = 0.1  
  ) +  
  bi_scale_fill(pal = "GrPink", dim = k) +  
  bi_theme() +  
  theme(legend.position = "none")
```

Mapas con ggplot

```
# Crear la leyenda para el mapa
legend1 <- bi_legend(
  pal = "GrPink", dim = k,
  xlab = "Coeficiente de variación",
  ylab = "Pobreza", size = 8
)

mapa1 <- ggdraw() +
  draw_plot(map.RM, 0, 0, 1, scale = 0.7) +
  draw_plot(legend1, 0.75, 0.4, 0.2, 0.2, scale = 1) +
  draw_text("Estimaciones directas de la pobreza en la mujer rural",
    vjust = -13, size = 18
  )

mapa1
```

mapas con ggplot

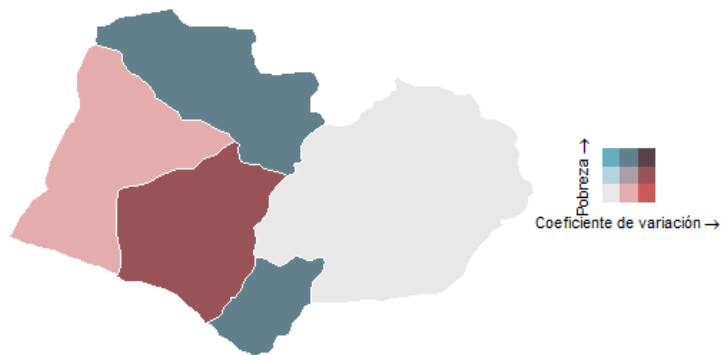


Figura 35: Mapa por región del CV para la estimación de la pobreza monetaria

¡Gracias!

Email: andres.gutierrez@cepal.org