ggplot.R

lescobar

2024-12-10

# Ggplot: crear visualizaciones gráficas con los datos   
  
# https://www.econjournals.com/index.php/ijeep/article/view/8037/4503  
  
# Tengo que tener instalado y habilitado tidyverse  
# install.packages("tidyverse")  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

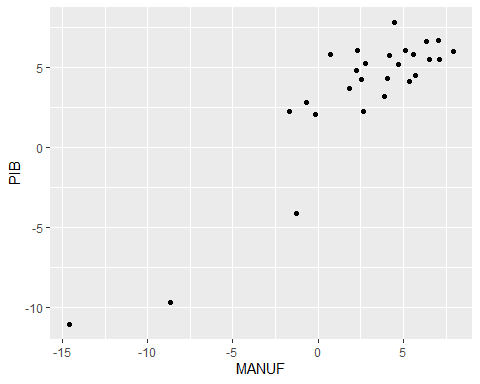
# cargar base de datos  
data <- read\_csv("https://raw.githubusercontent.com/Lufesc/ggplot/refs/heads/main/DatosSectorialesDptos.csv")

## Rows: 27 Columns: 9  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): DEPTO  
## dbl (8): PERIODO, PIB, AGRI, PetrMinas, MANUF, ELEC, CONSTR, COMER  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

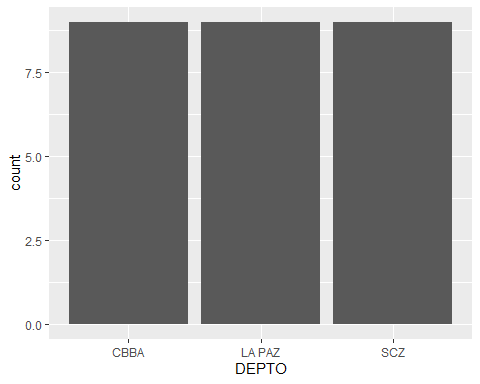
data

## # A tibble: 27 × 9  
## DEPTO PERIODO PIB AGRI PetrMinas MANUF ELEC CONSTR COMER  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 LA PAZ 2015 7.83 2.28 10.3 4.49 5.68 6.97 4.47  
## 2 LA PAZ 2016 5.51 2.39 -0.49 6.55 4.85 6.81 4.82  
## 3 LA PAZ 2017 6.1 3.76 29.5 2.32 3.75 8.16 7.26  
## 4 LA PAZ 2018 4.52 3.74 -0.32 5.69 3.11 3.27 4.08  
## 5 LA PAZ 2019 3.21 4.09 5.36 3.87 3.5 -2.76 3.21  
## 6 LA PAZ 2020 -11.1 3.09 -24.3 -14.6 -4.73 -6.83 -9.08  
## 7 LA PAZ 2021 5.29 -3.35 38.9 2.78 8.24 19.8 8.32  
## 8 LA PAZ 2022 4.8 2.34 7.72 2.24 4.33 4.71 5.48  
## 9 LA PAZ 2023 2.23 -4.84 -10.8 2.64 6.17 1.22 1.45  
## 10 CBBA 2015 6.05 4.37 -6.3 5.1 7.01 4.75 4.57  
## # ℹ 17 more rows

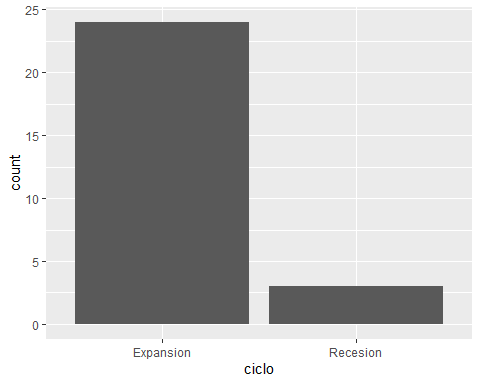
# Exploración inicial gráfico de dispersión  
data %>%  
 ggplot(aes(x=MANUF, y=PIB)) +  
 geom\_point()



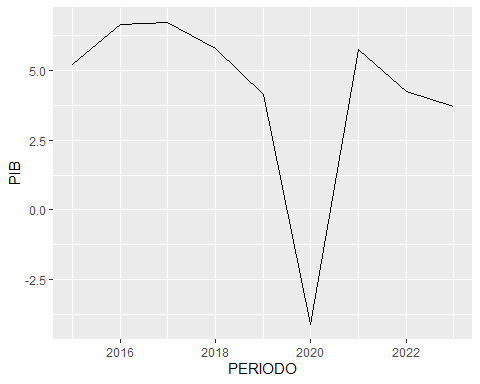
# Gráfico de una variable  
# Gráfico de barras  
data %>%  
 ggplot(aes(x=DEPTO)) +  
 geom\_bar()



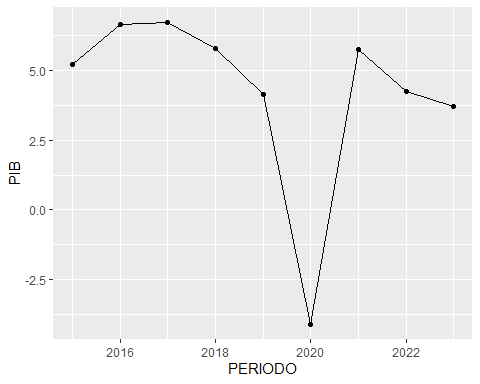
data <- data %>%   
 mutate(ciclo = if\_else(PIB <= 2.0,   
 "Recesion",   
 "Expansion"))   
  
data %>%  
 ggplot(aes(x=ciclo)) +   
 geom\_bar()



# Gráfico de líneas y puntos  
  
data %>%  
 filter(`DEPTO` == "SCZ") %>%  
 ggplot(aes(x = PERIODO, y = PIB))+  
 geom\_line()

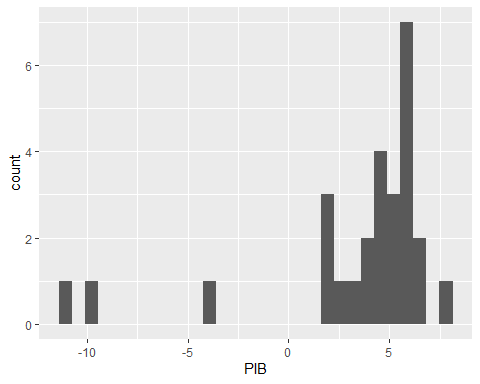


data %>%  
 filter(`DEPTO` == "SCZ") %>%  
 ggplot(aes(x = PERIODO, y = PIB))+  
 geom\_line() +  
 geom\_point()

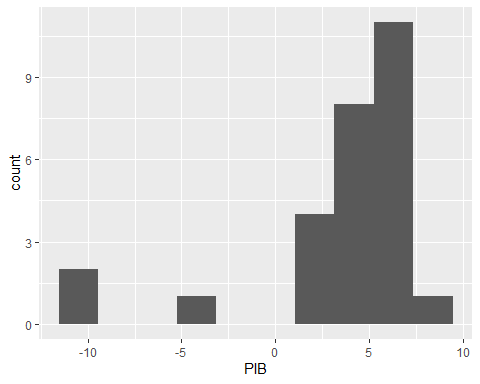


# Histograma  
data %>%  
 ggplot(aes(x = PIB)) +  
 geom\_histogram()

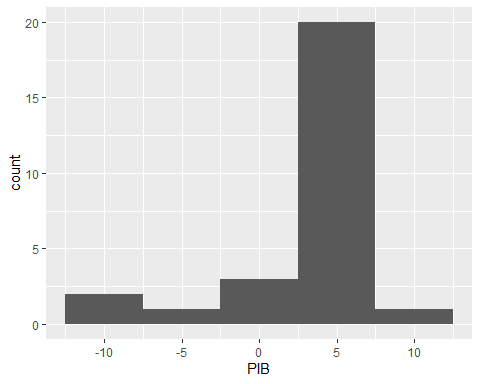
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



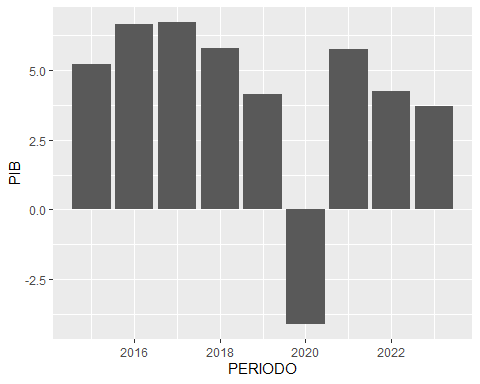
# Personalización del histograma  
# Cambiar el número de bins o el ancho de las barras:  
  
data %>% ggplot(aes(x = PIB)) + geom\_histogram(bins = 10)



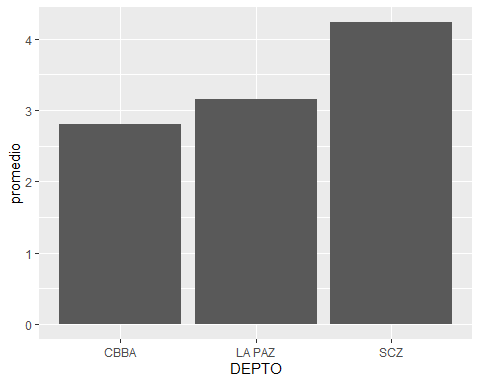
data %>% ggplot(aes(x = PIB)) + geom\_histogram(binwidth = 5) # especificar el ancho de los bins



# Gráfico de columnas  
  
data %>%  
 filter(`DEPTO` == "SCZ") %>%  
 ggplot(aes(x = PERIODO, y = PIB))+  
 geom\_col()

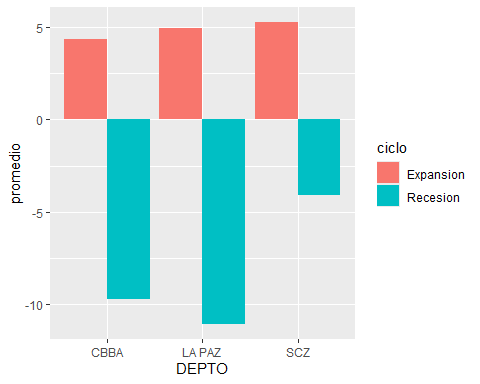


data %>%   
 group\_by(DEPTO) %>%   
 summarise(promedio = mean(PIB)) %>%   
 ggplot(aes(x = DEPTO, y = promedio)) +   
 geom\_col()



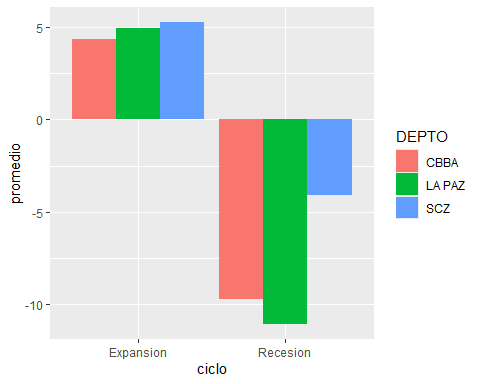
# Gráfico de barras apiladas y proporciones  
  
data %>%   
 group\_by(DEPTO, ciclo) %>%   
 summarise(promedio = mean(PIB)) %>%   
 ggplot(aes(x = DEPTO, y = promedio, fill = ciclo)) +   
 geom\_col(position = "dodge")

## `summarise()` has grouped output by 'DEPTO'. You can override using the  
## `.groups` argument.



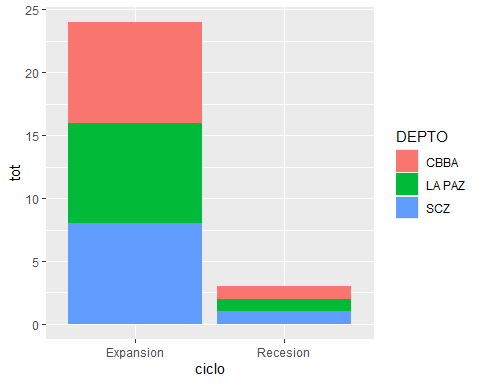
data %>%   
 group\_by(DEPTO, ciclo) %>%   
 summarise(promedio = mean(PIB)) %>%   
 ggplot(aes(x = ciclo, y = promedio, fill = DEPTO)) +   
 geom\_col(position = "dodge")

## `summarise()` has grouped output by 'DEPTO'. You can override using the  
## `.groups` argument.



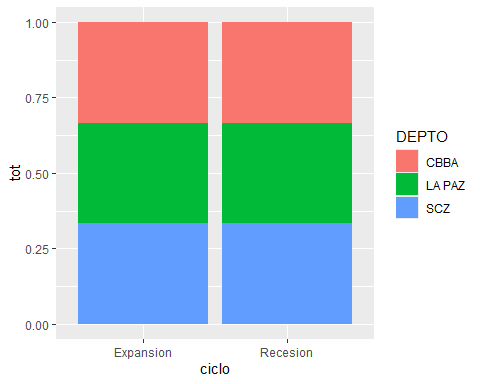
data %>%   
 group\_by(DEPTO, ciclo) %>%   
 summarise(tot = n()) %>%   
 ggplot(aes(x = ciclo, y = tot, fill = DEPTO)) +   
 geom\_col(position = "stack")

## `summarise()` has grouped output by 'DEPTO'. You can override using the  
## `.groups` argument.

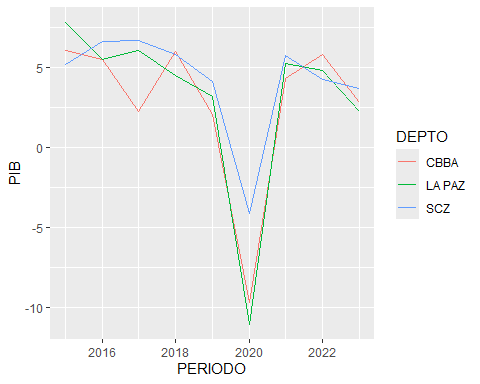


data %>%   
 group\_by(DEPTO, ciclo) %>%   
 summarise(tot = n()) %>%   
 ggplot(aes(x = ciclo, y = tot, fill = DEPTO)) +   
 geom\_col(position = "fill")

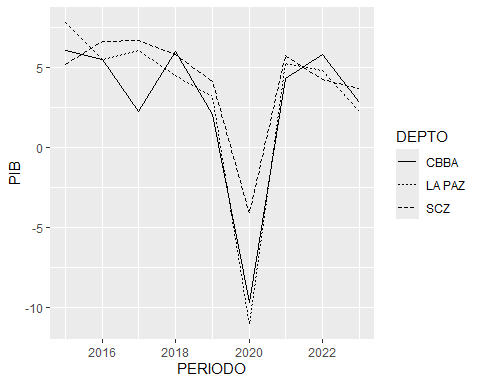
## `summarise()` has grouped output by 'DEPTO'. You can override using the  
## `.groups` argument.



# Gráficos de línea combinando diferentes atributos  
  
data %>%   
 filter(`DEPTO` %in% c("LA PAZ", "CBBA", "SCZ")) %>%   
 ggplot(aes(x = PERIODO, y = PIB, color = `DEPTO`)) +   
 geom\_line()

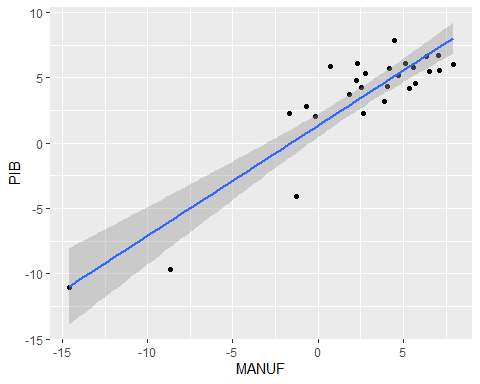


data %>%   
 filter(`DEPTO` %in% c("LA PAZ", "CBBA", "SCZ")) %>%   
 ggplot(aes(x = PERIODO, y = PIB, linetype = `DEPTO`)) +   
 geom\_line()



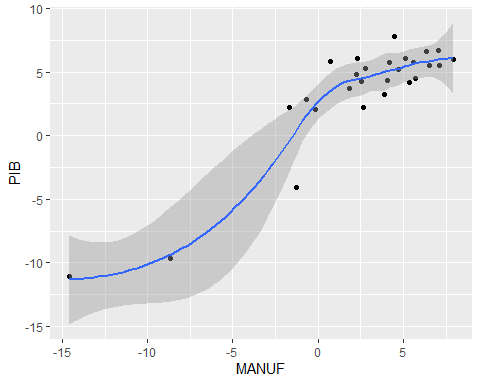
# Gráficos de dispersión  
  
data %>%   
 ggplot(aes(x=MANUF, y=PIB)) +  
 geom\_point() +  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula = 'y ~ x'

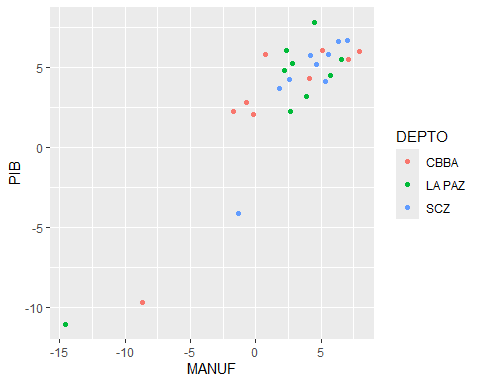


data %>%   
 ggplot(aes(x=MANUF, y=PIB)) +  
 geom\_point() +  
 geom\_smooth()

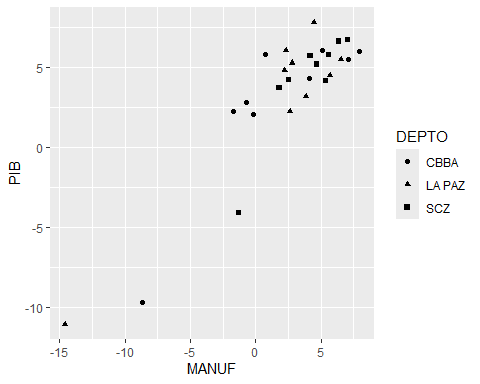
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



data %>%   
 ggplot(aes(x=MANUF, y=PIB, color = DEPTO)) +  
 geom\_point()

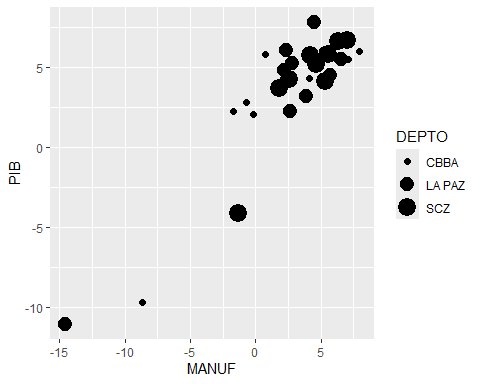


data %>%   
 ggplot(aes(x=MANUF, y=PIB, shape = DEPTO)) +  
 geom\_point()

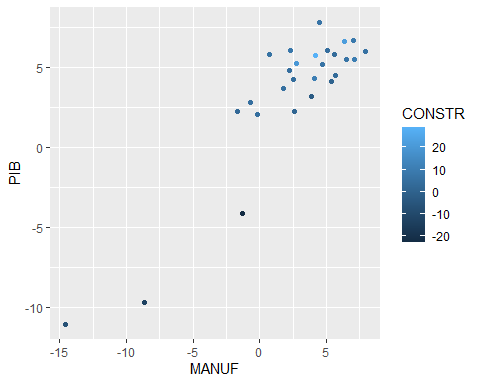


data %>%   
 ggplot(aes(x=MANUF, y=PIB, size = DEPTO)) +  
 geom\_point()

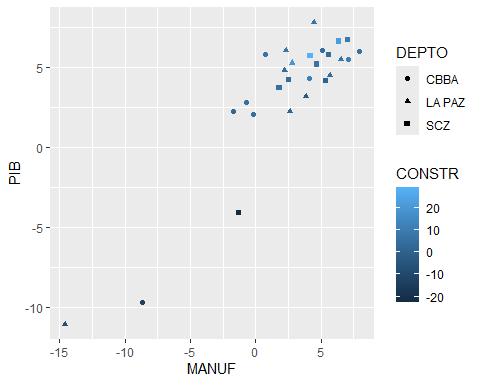
## Warning: Using size for a discrete variable is not advised.



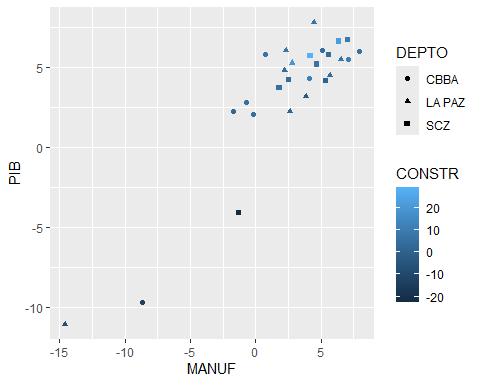
data %>%   
 ggplot(aes(x=MANUF, y=PIB, color = CONSTR)) +  
 geom\_point()



data %>%   
 ggplot(aes(x=MANUF, y=PIB, shape = DEPTO, color = CONSTR)) +  
 geom\_point()



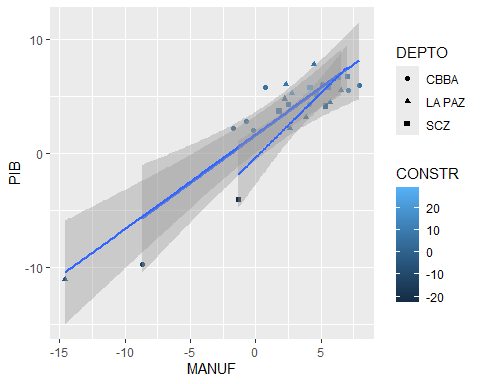
data %>%   
 ggplot(aes(x=MANUF, y=PIB, shape = DEPTO, color = CONSTR)) +  
 geom\_point()



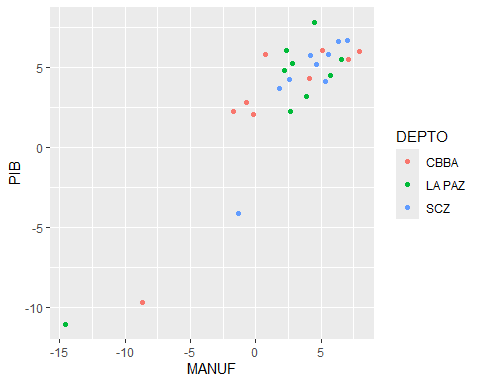
data %>%   
 ggplot(aes(x=MANUF, y=PIB, shape = DEPTO, color = CONSTR)) +  
 geom\_point() +  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula = 'y ~ x'

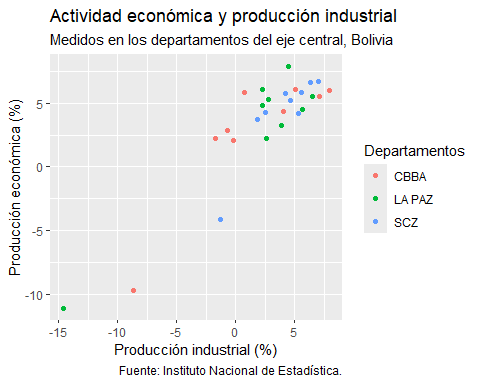
## Warning: The following aesthetics were dropped during statistical transformation:  
## colour.  
## ℹ This can happen when ggplot fails to infer the correct grouping structure in  
## the data.  
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
## variable into a factor?



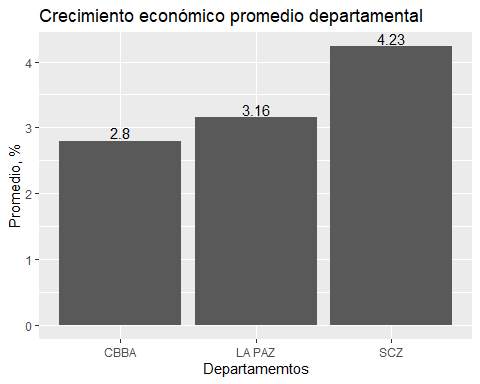
## Manipulación de gráficos  
# Rotulación: es preciso rotular claramente cada elemento de un gráfico.  
  
data %>%   
 ggplot(aes(x=MANUF, y=PIB, color = DEPTO)) +  
 geom\_point()



# Una versión rotulada del mismo gráfico  
  
data %>%   
 ggplot(aes(x=MANUF, y=PIB, color = DEPTO)) +  
 geom\_point() +  
 labs(title = "Actividad económica y producción industrial",  
 subtitle = "Medidos en los departamentos del eje central, Bolivia",  
 caption = "Fuente: Instituto Nacional de Estadística.",  
 x = "Producción industrial (%)",  
 y = "Producción económica (%)",  
 color = "Departamentos")

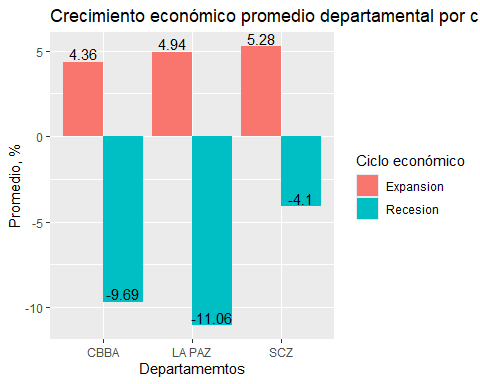


data %>%   
 group\_by(DEPTO) %>%   
 summarise(promedio = mean(PIB)) %>%   
 ggplot(aes(x = DEPTO, y = promedio)) +   
 geom\_col() +  
 labs(title = "Crecimiento económico promedio departamental",  
 x = "Departamemtos",  
 y = "Promedio, %") +   
 geom\_text(aes(label=round(promedio, 2)), size=4, color="black", vjust = -0.2)



data %>%   
 group\_by(DEPTO, ciclo) %>%   
 summarise(promedio = mean(PIB)) %>%   
 ggplot(aes(x = DEPTO, y = promedio, fill = ciclo)) +   
 geom\_col(position = "dodge") +  
 labs(title = "Crecimiento económico promedio departamental por ciclo",  
 x = "Departamemtos",  
 y = "Promedio, %",  
 fill = "Ciclo económico") +   
 geom\_text(aes(label=round(promedio, 2)), size=4, color="black", vjust = -0.2, position = position\_dodge(0.9))

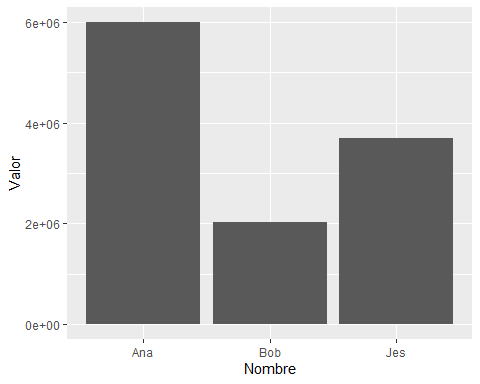
## `summarise()` has grouped output by 'DEPTO'. You can override using the  
## `.groups` argument.



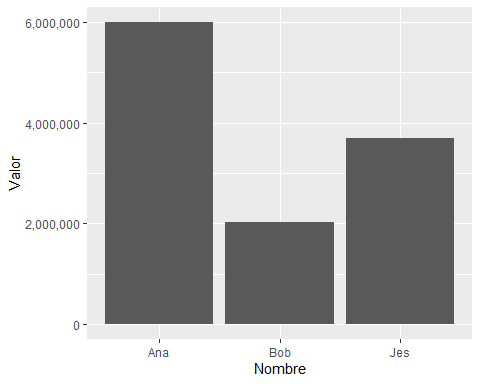
# Se puede realizar más ajustes: https://tidyverse.github.io/ggplot2-docs/reference/geom\_text.html  
  
# Cambiar el formato de números en ejes  
  
datos\_grandes <- tibble(Nombre = c("Bob", "Ana", "Jes"),  
 Valor = c(2023889, 5998300, 3700112))  
  
datos\_grandes

## # A tibble: 3 × 2  
## Nombre Valor  
## <chr> <dbl>  
## 1 Bob 2023889  
## 2 Ana 5998300  
## 3 Jes 3700112

datos\_grandes %>%   
 ggplot(aes(x = Nombre, y = Valor)) +  
 geom\_col()



datos\_grandes %>%   
 ggplot(aes(x = Nombre, y = Valor)) +  
 geom\_col() +  
 scale\_y\_continuous(labels = scales::label\_comma())



datos\_grandes %>%  
 mutate(Valor = Valor/1000000) %>%   
 ggplot(aes(x = Nombre, y = Valor)) +  
 geom\_col() +  
 labs(y = "Valor (millones)")

