PCA de datos ventas de tienda de autoservicio

RUBEN PIZARRO GURROLA

2025-08-25

# Objetivo

Aplicar reducción de dimensionalidad mediante modelo de análisis de componentes principales (PCA) en un conjunto de datos ventas de autoservicio.

# Cargar librerías

library(readr) # Para importar y exportar datos  
library(ggplot2) # Para gráficos  
library(dplyr) # Para procesar:filtrar, seleccionar, modificar..  
# library(FactoMineR)  
# library(factoextra)  
library(knitr) # Para generar tablas  
library(kableExtra) # Para modificar tablas HTML  
library(flextable) # Para modificar tablas para word  
   
library(tidyr) # Para formatear y tratar datos  
  
library(gridExtra) # Graficos  
library(factoextra) #Gráficos  
  
  
library(reshape2) # Para gráficos

# Cargar funciones

Se cargan las funciones que se utilizan en el caso

source ("https://raw.githubusercontent.com/rpizarrog/machine\_learning\_r\_python\_casos\_de\_estudio/refs/heads/main/funciones/funciones%20para%20pca.R")

Las funciones

# Funciones para PCA  
# Agosto 2025  
# Rubén Pizarro Gurrola  
# Función para cargar datos de una url local o internet  
f\_cargar\_datos <- function (url){  
 datos <- read.csv(url)  
 return (datos)  
}  
  
# Función para presentar datos a manera de tablas  
# Recibe los datos y el tipo de registros head o tail default son 10  
f\_datos\_tablas <- function(datos, tipo = 'head', decimales = 2) {  
 # Crear tabla con los primeros 10 registros   
 if (tipo == 'head') {  
 tabla <- as.data.frame(head(datos, 10,) %>%  
 round(decimales))  
 # Crear y formatear la flextable  
 flextable(tabla) %>%  
 autofit() %>% # Ajusta automáticamente el ancho de columnas  
 align(align = "center", part = "all") %>% # Centra el contenido  
 fontsize(size = 4, part = "all") %>% # Tamaño de fuente  
 bold(part = "header") %>% # Negrita en el encabezado  
 set\_table\_properties(layout = "autofit") %>% # Ajuste automático para Word  
 theme\_box() %>% # Tema con bordes  
 set\_caption("Primeros registros de ventas") # Título de la tabla  
   
 } else {  
 tabla <- as.data.frame(tail(datos, 10) %>%  
 round(decimales))  
 # Crear y formatear la flextable  
 flextable(tabla) %>%  
 autofit() %>% # Ajusta automáticamente el ancho de columnas  
 align(align = "center", part = "all") %>% # Centra el contenido  
 fontsize(size = 4, part = "all") %>% # Tamaño de fuente  
 bold(part = "header") %>% # Negrita en el encabezado  
 set\_table\_properties(layout = "autofit") %>% # Ajuste automático para Word  
 theme\_box() %>% # Tema con bordes  
 set\_caption("Últimos registros de ventas") # Título de la tabla  
 }  
   
}  
  
  
# Función para analizar cada componente  
f\_analizar\_componente <- function(cargas, componente\_num, n\_vars = 5) {  
 componente <- paste0("PC", componente\_num)  
   
 # Ordenar variables por importancia absoluta  
 vars\_importantes <- cargas %>%  
 select(Variable, all\_of(componente)) %>%  
 mutate(Magnitud = abs(.[[2]])) %>%  
 arrange(desc(Magnitud)) %>%  
 head(n\_vars)  
 rownames(vars\_importantes) <- NULL  
 return(vars\_importantes)  
}  
  
# Función para gráficos de variables importantes de un PCA  
# Recibe las cargas del pca con su varianza explicada de cada # variable de cada componente de las cuales representa las   
# principales variables de cada componente  
# Por default 5 componentes 5 variables   
# aunque puede ser variar  
f\_graf\_cargas\_importantes <- function(cargas, n\_componentes = 5, n\_vars = 5) {  
 # Seleccionar dinámicamente los componentes (PC1, PC2, ..., PCn)  
 componentes <- paste0("PC", 1:n\_componentes)  
   
 cargas\_abs <- cargas %>%  
 select(Variable, all\_of(componentes)) %>%  
 pivot\_longer(  
 cols = -Variable,  
 names\_to = "Componente",  
 values\_to = "Loading"  
 ) %>%  
 group\_by(Componente) %>%  
 mutate(Magnitud = abs(Loading)) %>%  
 slice\_max(order\_by = Magnitud, n = n\_vars, with\_ties = FALSE) %>%  
 ungroup()  
   
 # Gráfico  
 ggplot(cargas\_abs, aes(x = reorder(Variable, Loading), y = Loading, fill = Loading)) +  
 geom\_bar(stat = "identity") +  
 scale\_fill\_gradient2(  
 low = "red", mid = "white", high = "blue",  
 midpoint = 0, name = "Contribución"  
 ) +  
 facet\_wrap(~Componente, scales = "free\_y") +  
 coord\_flip() +  
 labs(  
 title = "Variables más importantes por Componente Principal",  
 subtitle = paste("Top", n\_vars, "variables por cada componente"),  
 x = "Variables",  
 y = "Loading (Contribución)"  
 ) +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5, face = "bold"))  
}  
  
# Función para presentar datos en foramto de tablas  
# Se unen los datos hedad y tail  
# Función que une y visualiza los primeros y últimos registros  
f\_visualizar\_tabla <- function(datos, n\_registros = 5) {  
 # Validar que los datos tengan suficientes registros  
 if (nrow(datos) < (n\_registros \* 2)) {  
 stop("El data.frame es muy pequeño para mostrar head y tail.")  
 }  
   
 # Obtener los primeros y últimos registros  
 head\_df <- head(datos, n\_registros)  
 tail\_df <- tail(datos, n\_registros)  
   
 # Crear una fila de separación visual  
 separador <- tibble(!!!setNames(rep("...", ncol(datos)), names(datos)))  
   
 # Unir los data.frames  
 tabla\_final <- rbind(head\_df, separador, tail\_df)  
   
 # Crear y formatear la flextable  
 flextable(tabla\_final) %>%  
 set\_caption("Primeros y últimos registros del conjunto de datos") %>%  
 autofit() %>%  
 align(align = "center", part = "all") %>%  
 fontsize(size = 8, part = "all") %>%  
 bold(part = "header") %>%  
 set\_table\_properties(layout = "autofit") %>%  
 theme\_box()  
}  
  
# Función para graficar todas las combinaciones de 2 componentes  
f\_componentes\_region <- function(datos, region\_col = "region") {  
   
 # Asegurar que la variable región es factor  
 datos[[region\_col]] <- as.factor(datos[[region\_col]])  
   
 # Lista de combinaciones de pares de componentes  
 pares <- combn(c("PC1", "PC2", "PC3", "PC4"), 2, simplify = FALSE)  
   
 # Generar un gráfico por cada par  
 plots <- lapply(pares, function(par) {  
 ggplot(datos, aes\_string(x = par[1], y = par[2], color = region\_col)) +  
 geom\_point(size = 3, alpha = 0.7) +  
 labs(title = paste("PCA:", par[1], "vs", par[2])) +  
 theme\_minimal()  
 })  
   
 # Organizar todos los gráficos en una cuadrícula  
 grid.arrange(grobs = plots, ncol = 2)  
}  
  
# Función para identificar posibles diferencias entre componentes y variabel categórica  
# datos\_redimensionados debe contener columnas: region, PC1, PC2, PC3, PC4  
f\_diagramas\_caja\_componentes\_region <- function(datos\_redimensionados,  
 region\_col = "region",  
 comp\_cols = c("PC1","PC2","PC3","PC4"),  
 ncol = 2) {  
 # Asegurar factor  
 datos\_redimensionados[[region\_col]] <- as.factor(datos\_redimensionados[[region\_col]])  
   
 # Generar un boxplot por cada componente  
 plots <- lapply(comp\_cols, function(pc) {  
 ggplot(datos\_redimensionados, aes\_string(x = region\_col, y = pc, fill = region\_col)) +  
 geom\_boxplot() +  
 labs(title = paste("Distribución de", pc, "por Región"),  
 x = "Región", y = pc) +  
 theme\_minimal() +  
 theme(legend.position = "none")  
 })  
   
 # Organizar en cuadrícula (2x2 por defecto)  
 grid.arrange(grobs = plots, ncol = ncol)  
}

# Cargar datos

url <- "https://raw.githubusercontent.com/rpizarrog/machine\_learning\_r\_python\_casos\_de\_estudio/refs/heads/main/datos/dataset\_ventas.csv"  
datos\_originales <- f\_cargar\_datos(url)

# Explorar datos

## Descripción de los datos

summary(datos\_originales)

## ventas\_totales clientes ticket\_promedio productos\_vendidos  
## Min. : 351.7 Min. :-0.9713 Min. :-573.01 Min. : 1.511   
## 1st Qu.: 874.5 1st Qu.:15.6275 1st Qu.: 43.08 1st Qu.:40.426   
## Median :1004.9 Median :19.9474 Median : 50.21 Median :50.376   
## Mean :1006.4 Mean :19.9368 Mean : 56.11 Mean :50.110   
## 3rd Qu.:1134.7 3rd Qu.:24.1882 3rd Qu.: 61.23 3rd Qu.:59.356   
## Max. :1785.2 Max. :43.0104 Max. : 784.97 Max. :97.393   
## ventas\_online ventas\_tienda gastos\_operativos publicidad   
## Min. : 67.03 Min. : 183.2 Min. : 65.01 Min. : 29.60   
## 1st Qu.:342.52 1st Qu.: 514.2 1st Qu.:258.09 1st Qu.: 85.93   
## Median :403.88 Median : 600.3 Median :301.13 Median :100.25   
## Mean :404.29 Mean : 604.8 Mean :301.81 Mean :100.54   
## 3rd Qu.:468.28 3rd Qu.: 692.1 3rd Qu.:347.54 3rd Qu.:115.18   
## Max. :743.51 Max. :1069.7 Max. :575.38 Max. :177.03   
## inventario personal descuentos competencia\_precios  
## Min. :147.2 Min. :-0.6571 Min. :14.23 Min. : 65.64   
## 1st Qu.:431.3 1st Qu.: 8.0905 1st Qu.:43.11 1st Qu.: 92.54   
## Median :502.3 Median :10.0123 Median :50.30 Median : 99.82   
## Mean :503.7 Mean :10.0036 Mean :50.26 Mean : 99.78   
## 3rd Qu.:575.4 3rd Qu.:11.8954 3rd Qu.:57.57 3rd Qu.:106.75   
## Max. :896.7 Max. :19.7823 Max. :87.37 Max. :137.28   
## satisfaccion\_clientes dias\_festivos region   
## Min. :2.394 Min. :0.000 Length:3000   
## 1st Qu.:3.657 1st Qu.:0.000 Class :character   
## Median :3.992 Median :1.000 Mode :character   
## Mean :3.995 Mean :0.506   
## 3rd Qu.:4.326 3rd Qu.:1.000   
## Max. :5.640 Max. :1.000

str(datos\_originales)

## 'data.frame': 3000 obs. of 15 variables:  
## $ ventas\_totales : num 1099 972 1130 1305 953 ...  
## $ clientes : num 12.4 15.1 20.5 35.5 21.8 ...  
## $ ticket\_promedio : num 86.1 62.9 53.2 35.6 43.2 ...  
## $ productos\_vendidos : num 62.6 59.4 61.5 45.8 46.1 ...  
## $ ventas\_online : num 498 295 435 520 382 ...  
## $ ventas\_tienda : num 652 582 681 830 535 ...  
## $ gastos\_operativos : num 339 268 360 374 308 ...  
## $ publicidad : num 135 97 109.8 125.2 99.1 ...  
## $ inventario : num 523 458 622 727 439 ...  
## $ personal : num 9.43 12.05 12.86 13 9.82 ...  
## $ descuentos : num 45.1 43.3 53.5 66 52.8 ...  
## $ competencia\_precios : num 94.8 107 96.2 96.7 93.6 ...  
## $ satisfaccion\_clientes: num 4.28 3.92 4.2 4.18 3.75 ...  
## $ dias\_festivos : int 1 1 1 0 0 0 1 0 0 0 ...  
## $ region : chr "Oeste" "Norte" "Este" "Sur" ...

## Primeros registros

Utilizando la función previamente preparada *f\_datos\_tablas()* se construye una tabla con los registros indicados, los primeros *‘head’* o los últimos *‘tail’* del conjunto de datos. Se muestran únicamente los datos numéricos a dos posiciones decimales.

# Crear tabla con los primeros 10 registros (excluyendo región)  
f\_datos\_tablas(datos\_originales[ , !(names(datos\_originales) %in% "region")], 'head')

Primeros registros de ventas

| **ventas\_totales** | **clientes** | **ticket\_promedio** | **productos\_vendidos** | **ventas\_online** | **ventas\_tienda** | **gastos\_operativos** | **publicidad** | **inventario** | **personal** | **descuentos** | **competencia\_precios** | **satisfaccion\_clientes** | **dias\_festivos** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1,099.34 | 12.45 | 86.09 | 62.62 | 498.48 | 652.43 | 339.46 | 135.05 | 522.82 | 9.43 | 45.06 | 94.81 | 4.28 | 1 |
| 972.35 | 15.15 | 62.94 | 59.35 | 294.99 | 581.78 | 268.26 | 97.04 | 457.62 | 12.05 | 43.34 | 107.03 | 3.92 | 1 |
| 1,129.54 | 20.52 | 53.15 | 61.46 | 435.43 | 680.94 | 359.60 | 109.83 | 621.79 | 12.86 | 53.54 | 96.23 | 4.20 | 1 |
| 1,304.61 | 35.53 | 35.62 | 45.81 | 519.76 | 830.11 | 373.67 | 125.24 | 726.88 | 13.00 | 65.98 | 96.73 | 4.18 | 0 |
| 953.17 | 21.85 | 43.20 | 46.10 | 382.06 | 534.54 | 308.44 | 99.08 | 439.43 | 9.82 | 52.78 | 93.57 | 3.75 | 0 |
| 953.17 | 12.39 | 78.63 | 36.10 | 390.29 | 529.59 | 277.73 | 99.18 | 449.83 | 9.51 | 51.13 | 91.71 | 3.82 | 0 |
| 1,315.84 | 28.75 | 45.13 | 63.77 | 556.86 | 851.34 | 343.12 | 141.09 | 703.62 | 14.13 | 59.90 | 103.54 | 4.30 | 1 |
| 1,153.49 | 15.33 | 72.06 | 67.47 | 459.62 | 668.96 | 316.57 | 106.09 | 598.08 | 9.94 | 55.82 | 104.84 | 4.91 | 0 |
| 906.11 | 23.54 | 40.78 | 48.36 | 344.04 | 480.51 | 258.92 | 87.49 | 480.11 | 10.79 | 45.69 | 102.27 | 4.60 | 0 |
| 1,108.51 | 19.81 | 54.27 | 66.91 | 423.03 | 640.85 | 373.60 | 131.16 | 590.12 | 11.79 | 54.11 | 93.84 | 3.44 | 0 |

## Últimos registros

Únicamente los datos numéricos a dos posiciones decimales.

# Crear tabla con los primeros 10 registros (excluyendo región)  
f\_datos\_tablas(datos\_originales[ , !(names(datos\_originales) %in% "region")], 'tail')

Últimos registros de ventas

| **ventas\_totales** | **clientes** | **ticket\_promedio** | **productos\_vendidos** | **ventas\_online** | **ventas\_tienda** | **gastos\_operativos** | **publicidad** | **inventario** | **personal** | **descuentos** | **competencia\_precios** | **satisfaccion\_clientes** | **dias\_festivos** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1,303.28 | 28.75 | 48.53 | 59.94 | 588.87 | 829.08 | 352.21 | 148.84 | 671.51 | 12.60 | 69.64 | 102.78 | 4.40 | 0 |
| 1,120.42 | 23.94 | 48.58 | 41.51 | 493.84 | 723.47 | 323.72 | 117.29 | 533.52 | 14.85 | 57.40 | 107.42 | 2.79 | 0 |
| 1,014.41 | 15.30 | 66.01 | 46.66 | 416.38 | 587.16 | 315.62 | 112.58 | 504.55 | 11.90 | 50.81 | 90.69 | 3.60 | 1 |
| 957.56 | 21.75 | 40.41 | 38.41 | 395.05 | 590.30 | 308.01 | 99.77 | 428.03 | 7.11 | 41.35 | 104.60 | 4.70 | 1 |
| 809.62 | 20.51 | 39.44 | 44.99 | 308.65 | 420.33 | 220.29 | 84.08 | 363.34 | 5.57 | 31.93 | 77.71 | 4.15 | 1 |
| 1,015.50 | 21.17 | 48.18 | 38.96 | 360.72 | 658.55 | 291.43 | 113.20 | 537.25 | 9.00 | 47.60 | 89.51 | 3.62 | 1 |
| 1,051.55 | 26.79 | 41.58 | 51.85 | 482.48 | 591.76 | 359.28 | 111.84 | 530.41 | 13.40 | 54.39 | 95.43 | 3.40 | 1 |
| 751.65 | 8.95 | 87.20 | 45.96 | 366.27 | 490.25 | 260.35 | 77.27 | 344.78 | 6.51 | 34.65 | 103.57 | 4.42 | 1 |
| 1,066.84 | 23.68 | 43.69 | 52.33 | 382.53 | 582.64 | 326.30 | 108.74 | 538.20 | 6.88 | 57.84 | 108.69 | 4.04 | 0 |
| 968.95 | 13.53 | 73.23 | 38.98 | 379.89 | 632.94 | 223.59 | 102.28 | 460.19 | 9.23 | 43.68 | 100.83 | 3.89 | 0 |

# Escalar datos

## Datos numéricos

Seleccionar solo los atributos que son numéricos:

* *‘ventas\_totales, ’clientes’, ‘ticket\_promedio’, ‘productos\_vendidos’, ‘ventas\_online’, ‘ventas\_tienda’, ‘gastos\_operativos’, ‘publicidad’, ‘inventario’, ‘personal’, ‘descuentos’, ‘competencia\_precios’, ‘satisfaccion\_clientes’, ‘dias\_festivos’*
* No incluye el atributo ‘*region*’

datos\_numericos <- datos\_originales %>%  
 select ('ventas\_totales', 'clientes', 'ticket\_promedio', 'productos\_vendidos', 'ventas\_online', 'ventas\_tienda', 'gastos\_operativos', 'publicidad', 'inventario', 'personal', 'descuentos', 'competencia\_precios', 'satisfaccion\_clientes', 'dias\_festivos')

## Escalar datos

A partir de solo los datos numéricos se deben escalar los datos con la función *scale()*; esta función es propia de los paquetes de R.

datos\_escalados <- scale(datos\_numericos)

## Primeros registros datos escalados

Únicamente los datos numéricos escalados a cuatro posiciones decimales.

f\_datos\_tablas(datos\_escalados, 'head',4)

Primeros registros de ventas escalados

| **ventas\_totales** | **clientes** | **ticket\_promedio** | **productos\_vendidos** | **ventas\_online** | **ventas\_tienda** | **gastos\_operativos** | **publicidad** | **inventario** | **personal** | **descuentos** | **competencia\_precios** | **satisfaccion\_clientes** | **dias\_festivos** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.4709 | -1.1748 | 0.8849 | 0.8931 | 1.0171 | 0.3710 | 0.5667 | 1.5848 | 0.1793 | -0.2050 | -0.4727 | -0.4870 | 0.5940 | 0.9879 |
| -0.1725 | -0.7517 | 0.2017 | 0.6597 | -1.1803 | -0.1787 | -0.5049 | -0.1610 | -0.4312 | 0.7267 | -0.6294 | 0.7094 | -0.1539 | 0.9879 |
| 0.6239 | 0.0919 | -0.0871 | 0.8105 | 0.3362 | 0.5927 | 0.8699 | 0.4266 | 1.1062 | 1.0140 | 0.2986 | -0.3476 | 0.4275 | 0.9879 |
| 1.5110 | 2.4463 | -0.6046 | -0.3072 | 1.2469 | 1.7533 | 1.0816 | 1.1343 | 2.0904 | 1.0645 | 1.4301 | -0.2991 | 0.3732 | -1.0119 |
| -0.2697 | 0.2995 | -0.3808 | -0.2859 | -0.2400 | -0.5462 | 0.0998 | -0.0671 | -0.6016 | -0.0650 | 0.2292 | -0.6084 | -0.5122 | -1.0119 |
| -0.2697 | -1.1845 | 0.6648 | -0.9999 | -0.1512 | -0.5848 | -0.3624 | -0.0623 | -0.5042 | -0.1734 | 0.0791 | -0.7899 | -0.3542 | -1.0119 |
| 1.5679 | 1.3821 | -0.3239 | 0.9748 | 1.6475 | 1.9185 | 0.6218 | 1.8625 | 1.8726 | 1.4651 | 0.8767 | 0.3678 | 0.6284 | 0.9879 |
| 0.7453 | -0.7222 | 0.4708 | 1.2395 | 0.5975 | 0.4996 | 0.2222 | 0.2548 | 0.8842 | -0.0240 | 0.5055 | 0.4946 | 1.8749 | -1.0119 |
| -0.5082 | 0.5646 | -0.4523 | -0.1251 | -0.6506 | -0.9666 | -0.6455 | -0.5996 | -0.2206 | 0.2777 | -0.4160 | 0.2435 | 1.2524 | -1.0119 |
| 0.5174 | -0.0192 | -0.0542 | 1.1990 | 0.2023 | 0.2809 | 1.0805 | 1.4062 | 0.8096 | 0.6347 | 0.3507 | -0.5817 | -1.1500 | -1.0119 |

## Últimos registros datos escalados

Únicamente los datos numéricos escalados a cuatro posiciones decimales.

f\_datos\_tablas(datos\_escalados, 'tail',4)

Últimos registros de ventas escalados

| **ventas\_totales** | **clientes** | **ticket\_promedio** | **productos\_vendidos** | **ventas\_online** | **ventas\_tienda** | **gastos\_operativos** | **publicidad** | **inventario** | **personal** | **descuentos** | **competencia\_precios** | **satisfaccion\_clientes** | **dias\_festivos** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.5042 | 1.3833 | -0.2237 | 0.7016 | 1.9931 | 1.7453 | 0.7586 | 2.2184 | 1.5718 | 0.9198 | 1.7628 | 0.2933 | 0.8317 | -1.0119 |
| 0.5777 | 0.6281 | -0.2220 | -0.6138 | 0.9670 | 0.9237 | 0.3298 | 0.7691 | 0.2796 | 1.7211 | 0.6492 | 0.7472 | -2.4840 | -1.0119 |
| 0.0406 | -0.7279 | 0.2922 | -0.2463 | 0.1305 | -0.1368 | 0.2079 | 0.5531 | 0.0082 | 0.6720 | 0.0503 | -0.8896 | -0.8193 | 0.9879 |
| -0.2475 | 0.2837 | -0.4633 | -0.8353 | -0.0998 | -0.1124 | 0.0933 | -0.0352 | -0.7084 | -1.0262 | -0.8106 | 0.4715 | 1.4606 | 0.9879 |
| -0.9971 | 0.0899 | -0.4920 | -0.3651 | -1.0328 | -1.4348 | -1.2269 | -0.7562 | -1.3142 | -1.5719 | -1.6676 | -2.1600 | 0.3165 | 0.9879 |
| 0.0461 | 0.1930 | -0.2340 | -0.7957 | -0.4706 | 0.4185 | -0.1563 | 0.5813 | 0.3145 | -0.3560 | -0.2418 | -1.0054 | -0.7690 | 0.9879 |
| 0.2288 | 1.0758 | -0.4287 | 0.1242 | 0.8443 | -0.1010 | 0.8650 | 0.5191 | 0.2504 | 1.2069 | 0.3754 | -0.4263 | -1.2250 | 0.9879 |
| -1.2908 | -1.7242 | 0.9177 | -0.2960 | -0.4106 | -0.8908 | -0.6239 | -1.0690 | -1.4880 | -1.2383 | -1.4199 | 0.3709 | 0.8749 | 0.9879 |
| 0.3062 | 0.5867 | -0.3665 | 0.1588 | -0.2350 | -0.1721 | 0.3687 | 0.3766 | 0.3234 | -1.1071 | 0.6899 | 0.8719 | 0.0913 | -1.0119 |
| -0.1898 | -1.0054 | 0.5054 | -0.7941 | -0.2635 | 0.2193 | -1.1773 | 0.0798 | -0.4072 | -0.2739 | -0.5989 | 0.1023 | -0.2137 | -1.0119 |

# Modelo PCA

Con la función *prcomp()* de la librería base de R, específicamente al paquete *stat*, se construye el modelo PCA con los datos escalados.

# Aplicar PCA  
pca\_resultado <- prcomp(datos\_escalados, center = TRUE, scale. = TRUE)

## Autovalores

Con la variable *pca\_resultado* se visualizan en modo consola los autovalores.

# Extraer resultados  
autovalores <- pca\_resultado$sdev^2  
print(autovalores)

## [1] 7.42110793 1.27474996 1.04645356 0.98436857 0.96978228 0.49542700  
## [7] 0.46975812 0.31575139 0.26851005 0.20804863 0.20593132 0.16896056  
## [13] 0.14311535 0.02803527

## Autovectores

Con la variable *pca\_resultado* se visualizan en modo consola los autovectores.

autovectores <- pca\_resultado$rotation  
print(autovectores)

## PC1 PC2 PC3 PC4  
## ventas\_totales -0.360775617 -0.056466422 0.001622064 -0.0034627157  
## clientes -0.244711452 0.522593620 0.006620625 -0.0341189329  
## ticket\_promedio 0.056986962 -0.828779865 0.027137190 0.0477447258  
## productos\_vendidos -0.274838625 -0.086202980 -0.002144472 -0.0091578291  
## ventas\_online -0.315276136 -0.070075637 -0.025471537 0.0107478403  
## ventas\_tienda -0.338859295 -0.061648039 0.005649093 -0.0017858404  
## gastos\_operativos -0.331136276 -0.057908348 0.012710047 0.0080631040  
## publicidad -0.330606674 -0.056816785 -0.001051735 0.0004921992  
## inventario -0.341440790 -0.049034215 0.002803893 -0.0062924082  
## personal -0.270462501 -0.048271939 -0.013175319 0.0131928109  
## descuentos -0.328964678 -0.062917722 0.020856973 0.0042096973  
## competencia\_precios -0.004556283 0.068975459 0.502584606 0.8090895065  
## satisfaccion\_clientes 0.003043775 -0.025610892 0.636242837 -0.1150889567  
## dias\_festivos -0.004152720 -0.001062185 -0.583400762 0.5728646075  
## PC5 PC6 PC7 PC8  
## ventas\_totales 0.002376491 0.034171450 0.0771815460 -0.005303971  
## clientes -0.015075564 -0.004560904 0.0610261545 0.769067940  
## ticket\_promedio 0.038270637 -0.017540814 0.0782087473 0.524100986  
## productos\_vendidos 0.002363323 0.332225865 -0.8892866340 0.005718688  
## ventas\_online 0.009473637 0.097636702 0.2417450105 -0.348540217  
## ventas\_tienda 0.003597128 0.046849242 0.1148522789 0.030304050  
## gastos\_operativos -0.024435623 0.083189944 0.1132914873 -0.050915646  
## publicidad 0.013561088 0.070412547 0.1548210474 -0.060696968  
## inventario 0.007796924 0.063082150 0.1262868598 0.010191412  
## personal 0.001797282 -0.923779517 -0.2495683791 -0.057809301  
## descuentos -0.002011045 0.082642511 0.0811128130 -0.028522621  
## competencia\_precios 0.295916654 0.006147023 -0.0110760881 -0.010575688  
## satisfaccion\_clientes -0.761257112 -0.009924098 -0.0005822101 -0.014196924  
## dias\_festivos -0.574687774 0.014474901 -0.0004733418 0.023323043  
## PC9 PC10 PC11 PC12  
## ventas\_totales -0.068979626 -0.020451048 -0.005591312 0.082105784  
## clientes 0.240064543 0.048922342 0.004615671 -0.093289090  
## ticket\_promedio 0.142449618 0.023252309 0.001580372 -0.056768219  
## productos\_vendidos 0.110720480 0.001655023 -0.039135494 -0.010766358  
## ventas\_online 0.806749292 0.198570440 0.050296773 -0.076528061  
## ventas\_tienda -0.110076035 -0.030552634 -0.062504383 0.737584405  
## gastos\_operativos -0.103364567 -0.698372655 0.459902578 -0.332194286  
## publicidad -0.195861088 -0.079492998 -0.803365097 -0.374821771  
## inventario -0.147007142 -0.065864065 0.027471887 0.308567987  
## personal 0.041586877 0.005664159 0.002669918 -0.033258940  
## descuentos -0.410961802 0.676468706 0.365518010 -0.289866701  
## competencia\_precios 0.006355954 -0.004628021 -0.004374192 0.006833660  
## satisfaccion\_clientes 0.024381673 0.011664610 -0.024725013 0.007437929  
## dias\_festivos -0.013708378 0.010010756 -0.008672737 0.006360213  
## PC13 PC14  
## ventas\_totales 0.053920237 0.919033715  
## clientes -0.017827139 -0.036057889  
## ticket\_promedio 0.001162291 -0.015186177  
## productos\_vendidos 0.004731474 -0.042066949  
## ventas\_online -0.011611477 -0.081158262  
## ventas\_tienda -0.516526744 -0.192961128  
## gastos\_operativos -0.165206542 -0.127509119  
## publicidad -0.081119206 -0.132387144  
## inventario 0.825921740 -0.238346782  
## personal 0.001238938 -0.047934624  
## descuentos -0.116927869 -0.123880786  
## competencia\_precios 0.004723802 0.003945926  
## satisfaccion\_clientes 0.008552774 0.001142044  
## dias\_festivos 0.005194852 0.000882965

## Varianza explicada y acumulada

Se calcula la varianza explicada y la varianza acumulada a partir de los autovalores y autovectores.

varianza\_explicada <- autovalores / sum(autovalores)  
print("Varianza explicada por componente:")

## [1] "Varianza explicada por componente:"

print(varianza\_explicada)

## [1] 0.530079138 0.091053569 0.074746683 0.070312041 0.069270163 0.035387643  
## [7] 0.033554152 0.022553671 0.019179289 0.014860617 0.014709380 0.012068612  
## [13] 0.010222525 0.002002519

varianza\_acumulada <- cumsum(varianza\_explicada)  
cat("\n") # Salto de linea en consola

print("Varianza acumulada de componentes:")

## [1] "Varianza acumulada de componentes:"

print(varianza\_acumulada)

## [1] 0.5300791 0.6211327 0.6958794 0.7661914 0.8354616 0.8708492 0.9044034  
## [8] 0.9269571 0.9461363 0.9609970 0.9757063 0.9877750 0.9979975 1.0000000

## Resultados

Se construye una tabla de la varianza explicada y acumulada.

# Crear dataframe con los resultados  
resultados\_pca <- data.frame(  
 Componente = paste0("PC", 1:length(autovalores)),  
 Autovalor = autovalores,  
 Varianza\_Individual = varianza\_explicada \* 100,  
 Varianza\_Acumulada = varianza\_acumulada \* 100  
)  
  
 # resultados\_pca  
  
 flextable(resultados\_pca) %>%  
 autofit() %>% # Ajusta automáticamente el ancho de columnas  
 align(align = "center", part = "all") %>% # Centra el contenido  
 fontsize(size = 9, part = "all") %>% # Tamaño de fuente  
 bold(part = "header") %>% # Negrita en el encabezado  
 set\_table\_properties(layout = "autofit") %>% # Ajuste automático para Word  
 theme\_box() %>% # Tema con bordes  
 set\_caption("Varianza explicada y acumulada por componente") # Título de la tabla

Varianza explicada y acumulada por componente

| **Componente** | **Autovalor** | **Varianza\_Individual** | **Varianza\_Acumulada** |
| --- | --- | --- | --- |
| PC1 | 7.42110793 | 53.0079138 | 53.00791 |
| PC2 | 1.27474996 | 9.1053569 | 62.11327 |
| PC3 | 1.04645356 | 7.4746683 | 69.58794 |
| PC4 | 0.98436857 | 7.0312041 | 76.61914 |
| PC5 | 0.96978228 | 6.9270163 | 83.54616 |
| PC6 | 0.49542700 | 3.5387643 | 87.08492 |
| PC7 | 0.46975812 | 3.3554152 | 90.44034 |
| PC8 | 0.31575139 | 2.2553671 | 92.69571 |
| PC9 | 0.26851005 | 1.9179289 | 94.61363 |
| PC10 | 0.20804863 | 1.4860617 | 96.09970 |
| PC11 | 0.20593132 | 1.4709380 | 97.57063 |
| PC12 | 0.16896056 | 1.2068612 | 98.77750 |
| PC13 | 0.14311535 | 1.0222525 | 99.79975 |
| PC14 | 0.02803527 | 0.2002519 | 100.00000 |

## Variables importanes de cada componente

Con la función precodificada *f\_analizar\_componente()* se enlistan las cinco principales variables que tienen carga absoluta sobre el componente específico, se identifican solo los primeros cinco componentes por la varianza acumulada del *83%*.

# ANÁLISIS DE VARIABLES POR COMPONENTE  
  
  
  
# Analizar los primeros 5 componentes  
print("==================================================")

## [1] "=================================================="

print("ASOCIACIÓN DE VARIABLES POR COMPONENTE PRINCIPAL")

## [1] "ASOCIACIÓN DE VARIABLES POR COMPONENTE PRINCIPAL"

print("==================================================")

## [1] "=================================================="

# Obtener loadings (autovectores)  
cargas <- as.data.frame(pca\_resultado$rotation)  
cargas$Variable <- rownames(cargas)  
  
for(i in 1:5) {  
 print(paste0("COMPONENTE PC", i, ":"))  
 print("----------------------------------------")  
   
 vars\_componente <- f\_analizar\_componente(cargas, i, 5)  
 print(vars\_componente)  
}

## [1] "COMPONENTE PC1:"  
## [1] "----------------------------------------"  
## Variable PC1 Magnitud  
## 1 ventas\_totales -0.3607756 0.3607756  
## 2 inventario -0.3414408 0.3414408  
## 3 ventas\_tienda -0.3388593 0.3388593  
## 4 gastos\_operativos -0.3311363 0.3311363  
## 5 publicidad -0.3306067 0.3306067  
## [1] "COMPONENTE PC2:"  
## [1] "----------------------------------------"  
## Variable PC2 Magnitud  
## 1 ticket\_promedio -0.82877986 0.82877986  
## 2 clientes 0.52259362 0.52259362  
## 3 productos\_vendidos -0.08620298 0.08620298  
## 4 ventas\_online -0.07007564 0.07007564  
## 5 competencia\_precios 0.06897546 0.06897546  
## [1] "COMPONENTE PC3:"  
## [1] "----------------------------------------"  
## Variable PC3 Magnitud  
## 1 satisfaccion\_clientes 0.63624284 0.63624284  
## 2 dias\_festivos -0.58340076 0.58340076  
## 3 competencia\_precios 0.50258461 0.50258461  
## 4 ticket\_promedio 0.02713719 0.02713719  
## 5 ventas\_online -0.02547154 0.02547154  
## [1] "COMPONENTE PC4:"  
## [1] "----------------------------------------"  
## Variable PC4 Magnitud  
## 1 competencia\_precios 0.80908951 0.80908951  
## 2 dias\_festivos 0.57286461 0.57286461  
## 3 satisfaccion\_clientes -0.11508896 0.11508896  
## 4 ticket\_promedio 0.04774473 0.04774473  
## 5 clientes -0.03411893 0.03411893  
## [1] "COMPONENTE PC5:"  
## [1] "----------------------------------------"  
## Variable PC5 Magnitud  
## 1 satisfaccion\_clientes -0.76125711 0.76125711  
## 2 dias\_festivos -0.57468777 0.57468777  
## 3 competencia\_precios 0.29591665 0.29591665  
## 4 ticket\_promedio 0.03827064 0.03827064  
## 5 gastos\_operativos -0.02443562 0.02443562

## Gráfico de variables y componentes

El gráfico siguiente se interpreta de la sigueinte manera: los cuadros reflejan las variables que mas se asocian a cada componente, por ejemplo:

Para el componente *1*, las variables *ventas\_totales*, *inventario*, *ventas\_tienda*, *gastos\_operativos* y *publicidad* son las que mayor relación tienen con ese componente con valores negativos pero en magnitud absoluta alrededor de *0.3* todas ellas.

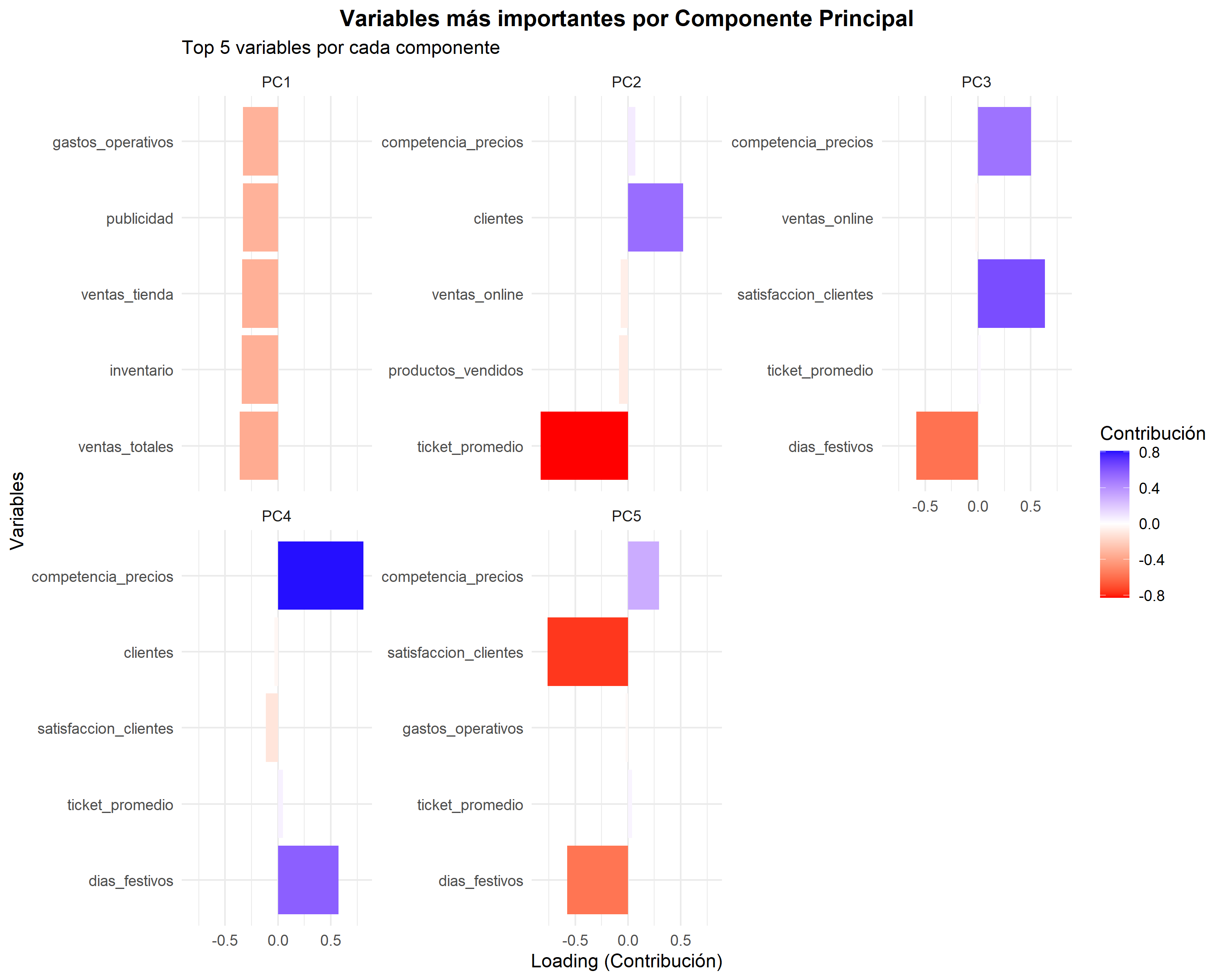
Para el componente *2* las variables *ticket\_promedio*  con valor de *-0.82* y *clientes*  con *0.52* son las que mas se asocian a dicho componente.

El componente 3, las variables que más se relacionan son *satisfaccion\_clientes* con valor aproximado de *0.60* y *dias\_festivos* alrededor de *-0.58*.

El componente *4*, se relaciona de manera importante con las variables *competencia\_precios* *0.80* y *dias\_festivos* con *0.57*.

El componente *5*, se asocia a las variables de manera importante con *satisfaccion\_clientes* *-0.76* y *dias\_festivos* con *-0.57*.

# Gráfico de variables importantes para los primeros 5 componentes  
  
  
f\_graf\_cargas\_importantes(cargas, 5, 5)



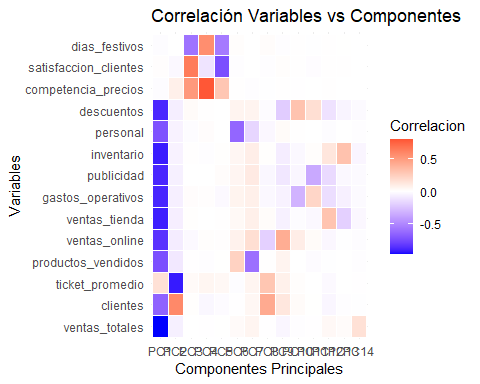
## Correlación de variables y componentes

Aquí se observa la correlación de los autovectores con cada componente de manera visual a través de mapa de calor.

# Calcular correlación entre variables originales y componentes  
correlaciones <- sweep(pca\_resultado$rotation, 2, pca\_resultado$sdev, FUN = "\*")  
  
# Ver la matriz de correlaciones  
print(correlaciones)

## PC1 PC2 PC3 PC4  
## ventas\_totales -0.982814492 -0.063753315 0.001659312 -0.0034355455  
## clientes -0.666635854 0.590033415 0.006772655 -0.0338512187  
## ticket\_promedio 0.155242233 -0.935732460 0.027760345 0.0473700968  
## productos\_vendidos -0.748707427 -0.097327324 -0.002193716 -0.0090859722  
## ventas\_online -0.858866122 -0.079118776 -0.026056444 0.0106635074  
## ventas\_tienda -0.923110682 -0.069603611 0.005778814 -0.0017718278  
## gastos\_operativos -0.902071858 -0.065381319 0.013001910 0.0079998369  
## publicidad -0.900629131 -0.064148892 -0.001075886 0.0004883372  
## inventario -0.930143118 -0.055361995 0.002868280 -0.0062430349  
## personal -0.736786116 -0.054501348 -0.013477866 0.0130892935  
## descuentos -0.896156053 -0.071037144 0.021335914 0.0041766659  
## competencia\_precios -0.012412095 0.077876622 0.514125521 0.8027409858  
## satisfaccion\_clientes 0.008291765 -0.028915933 0.650852963 -0.1141859112  
## dias\_festivos -0.011312719 -0.001199258 -0.596797469 0.5683696254  
## PC5 PC6 PC7 PC8  
## ventas\_totales 0.002340309 0.024052113 0.0528993850 -0.002980395  
## clientes -0.014846042 -0.003210264 0.0418266569 0.432152811  
## ticket\_promedio 0.037687976 -0.012346378 0.0536034175 0.294501568  
## productos\_vendidos 0.002327342 0.233842410 -0.6095073045 0.003213432  
## ventas\_online 0.009329404 0.068723131 0.1656893786 -0.195850882  
## ventas\_tienda 0.003542362 0.032975577 0.0787184922 0.017028379  
## gastos\_operativos -0.024063597 0.058554553 0.0776487427 -0.028610398  
## publicidad 0.013354624 0.049560981 0.1061126477 -0.034106694  
## inventario 0.007678218 0.044401365 0.0865556285 0.005726734  
## personal 0.001769919 -0.650216768 -0.1710514296 -0.032484063  
## descuentos -0.001980427 0.058169233 0.0555938323 -0.016027363  
## competencia\_precios 0.291411395 0.004326679 -0.0075914293 -0.005942665  
## satisfaccion\_clientes -0.749667159 -0.006985233 -0.0003990404 -0.007977501  
## dias\_festivos -0.565938294 0.010188387 -0.0003244233 0.013105629  
## PC9 PC10 PC11 PC12  
## ventas\_totales -0.035743832 -0.0093282031 -0.0025373181 0.033749429  
## clientes 0.124396538 0.0223146291 0.0020945756 -0.038346306  
## ticket\_promedio 0.073814480 0.0106059241 0.0007171673 -0.023334470  
## productos\_vendidos 0.057373089 0.0007548948 -0.0177595524 -0.004425491  
## ventas\_online 0.418040990 0.0905726408 0.0228245024 -0.031456716  
## ventas\_tienda -0.057039151 -0.0139357738 -0.0283642737 0.303182688  
## gastos\_operativos -0.053561405 -0.3185441677 0.2087022066 -0.136547839  
## publicidad -0.101491212 -0.0362586232 -0.3645643154 -0.154069787  
## inventario -0.076176095 -0.0300421466 0.0124666479 0.126836293  
## personal 0.021549470 0.0025835560 0.0012115996 -0.013671025  
## descuentos -0.212952004 0.3085532625 0.1658708145 -0.119149165  
## competencia\_precios 0.003293525 -0.0021109489 -0.0019849932 0.002808963  
## satisfaccion\_clientes 0.012634084 0.0053205023 -0.0112201259 0.003057347  
## dias\_festivos -0.007103401 0.0045661409 -0.0039356581 0.002614354  
## PC13 PC14  
## ventas\_totales 0.0203983436 0.1538805768  
## clientes -0.0067441119 -0.0060374376  
## ticket\_promedio 0.0004397014 -0.0025427334  
## productos\_vendidos 0.0017899444 -0.0070435787  
## ventas\_online -0.0043926902 -0.0135889250  
## ventas\_tienda -0.1954051132 -0.0323089014  
## gastos\_operativos -0.0624986090 -0.0213497900  
## publicidad -0.0306878740 -0.0221665536  
## inventario 0.3124510643 -0.0399081554  
## personal 0.0004686976 -0.0080260468  
## descuentos -0.0442345024 -0.0207422716  
## competencia\_precios 0.0017870420 0.0006606954  
## satisfaccion\_clientes 0.0032355648 0.0001912208  
## dias\_festivos 0.0019652430 0.0001478413

# Convertir matriz a data.frame largo  
df\_cor <- melt(correlaciones, varnames = c("Variable", "Componente"), value.name = "Correlacion")  
  
# Graficar heatmap  
ggplot(df\_cor, aes(x = Componente, y = Variable, fill = Correlacion)) +  
 geom\_tile(color = "white") +  
 scale\_fill\_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +  
 theme\_minimal() +  
 labs(title = "Correlación Variables vs Componentes",  
 x = "Componentes Principales", y = "Variables")



## Datos reducidos

Con el siguiente código se generan los datos redimensionados a partir de la variable ya conocida *pca\_resultado* con cuatro componentes.

Al conjunto de datos *datos\_redimensionados* con cuatro dimension, se le agrega el atributo de region de los datos originales.

Usando función *flextable(head(datos\_redimensionados))* lo que modifica a los primeros y últimos registros es precisamente *head()* y *tail()*.

# Tomar los datos redimensionados y dejar 4 componentes  
datos\_redimensionados <- as.data.frame(pca\_resultado$x[, 1:4])  
names(datos\_redimensionados) <- c("PC1", "PC2", "PC3", "PC4")  
  
# (Opcional) Adjuntar identificador de fila y/o columnas no numéricas útiles (p. ej. region)  
datos\_redimensionados <- round(datos\_redimensionados, 4)  
if ("region" %in% names(datos\_originales)) {  
 datos\_redimensionados <- bind\_cols(datos\_originales %>% select(region), datos\_redimensionados)  
}  
  
# Primeros y Últimos registros  
  
# Mandar llamar la función para visualziar los primeros y últimos registros  
f\_visualizar\_tabla(datos\_redimensionados)

Primeros y últimos registros del conjunto de datos

| **region** | **PC1** | **PC2** | **PC3** | **PC4** |
| --- | --- | --- | --- | --- |
| Oeste | -1.0858 | -1.6869 | -0.453 | 0.1859 |
| Norte | 0.8794 | -0.3975 | -0.32 | 1.1806 |
| Este | -1.9626 | -0.2775 | -0.4802 | 0.2361 |
| Sur | -4.2796 | 1.1251 | 0.6927 | -0.9529 |
| Este | 0.4838 | 0.551 | -0.0412 | -1.0348 |
| ... | ... | ... | ... | ... |
| Oeste | 0.0741 | 0.3102 | -1.5619 | -0.1841 |
| Norte | -1.6357 | 0.6681 | -1.5934 | 0.3353 |
| Este | 3.3196 | -1.1584 | 0.1599 | 0.8538 |
| Sur | -0.4702 | 0.6167 | 1.1199 | 0.0624 |
| Norte | 1.347 | -0.7121 | 0.4971 | -0.4222 |

# Interpretación

Para dar un agregado a la interpretación del modelo *PCA* que se le da usando *Python*, con los datos redimensionados en *R*, se generan dos gráficos tratando de encontrar algunos patrones con relación a cada componente y la variable categórica llamada *region*.

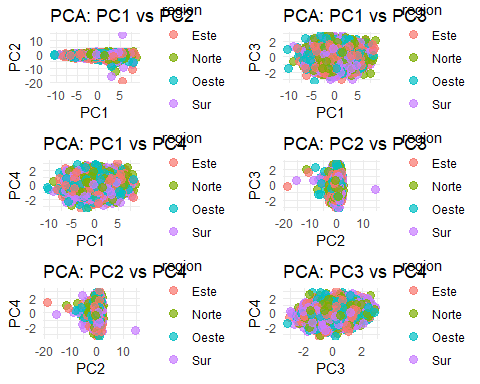
## Visualizar componentes por región

Con los datos redimensionados y con el atributo *region* se generan las seis posibles combinaciones para comparar si existen grupos significativos de acuerdo con pares de componentes.

En la gráfica se observa que no hay patrones importantes en cada componente de acuerdo a la región.

Se utiliza la función *f\_componentes\_region()* para visualizar las comparaciones:

# Uso de la función  
f\_componentes\_region(datos\_redimensionados)



## Diferencias sigificativas

Al observar visualmente los diagramas de cajas no se identifica que existan diferencias o patrones importantes entre las regiones y los cuatro componentes; sin embargo para lograr significancias estadísticamente hablando habrá que hacer pruebas tal vez comi prueba *ANOVA* o prueba de *Kruskall Wallis* dependiente de si los datos provienen de una distribución normal o no.

Esto último de pruebas de diferencias estadísticas, queda fuera de este libro pero se deja a manera de interpretación para una importante discusión y conclusión.

# Usando la función  
f\_diagramas\_caja\_componentes\_region(datos\_redimensionados)

