PCA FEATURES

Manuel Rocamora Valenti

2025-05-22

Contents

Data Preparation

Los datos utilizados han sido previamente transformados, seleccionando únicamente las variables relevantes y revisando su naturaleza (numérica, categórica, etc.).

En esta primera etapa del análisis, se realiza la carga de datos, se declara correctamente el tipo de cada variable y se lleva a cabo la creación de variables dummy, necesarias para la aplicación del análisis de componentes principales (PCA).

```
datos <- read.xlsx("datosLimpio.xlsx")

# Vector con los nombres de las variables a eliminar
vars_a_eliminar <- c(
  "NLR_diff_0_1eval", "PLR_diff_0_1eval", "SII_diff_0_1eval", "Hemoglobin_diff_0_1eval",
  "Neutrophils_diff_0_1eval", "Leukocytes_diff_0_1eval", "Lymphocytes_diff_0_1eval",
  "Protein_diff_0_1eval", "Albumin_diff_0_1eval", "PNI_diff_0_1eval", "NLR_diff_0_1C",
  "Platelets_diff_0_1eval", "SII_diff_0_1C", "PLR_diff_0_1C", "Lymphocytes_diff_0_1C",
  "Neutrophils_diff_0_1C", "PNI_diff_1C_2C", "Hemoglobin_diff_1C_2C",
  "Albumin_diff_2C_1eval", "Platelets_diff_1C_2C", "Leukocytes_diff_0_1C"
)
```

```r
# Eliminar esas columnas del dataframe
datos <- datos[, !(names(datos) %in% vars_a_eliminar)]
```

```r
tipos <- c(
  "categorical", # Sex
  "numerical",   # Age_at_diagnosis
  "categorical", # ECOG
  "numerical",   # BMI
  "numerical",   # %_weight_loss
  "numerical",   # Smoking_exposure
  "categorical", # Diabetes
  "categorical", # Cardiopathy
  "categorical", # Neurodegenerative
  "categorical", # Histology (parece que había un pequeño error en el nombre: Histológicasgy)
  "numerical", # Stage
  "numerical",   # PD_L1
  "categorical", # Statins
  "numerical",   # Total_COreltoeo
  "numerical",   # LDH
  "numerical",   # PCR
  "numerical",   # ALI_pre
  "categorical", # X1º_eval
  "categorical", # Mejor_resp
  rep("numerical", 56)
)

# 2. Validar número de columnas
if(length(tipos) != ncol(datos)) {
  stop("El número de tipos no coincide con el número de columnas en 'datos'")
}

# 3. Crear descripción
descDatos <- data.frame(
  variable = colnames(datos),
  tipo = tipos,
  stringsAsFactors = FALSE
)
rownames(descDatos) <- descDatos$variable

# 4. Aplicar transformación de tipos a 'datos'
for (i in seq_along(tipos)) {
  if (tipos[i] == "categorical") {
    datos[[i]] <- as.factor(datos[[i]])  # o usa as.character() si prefieres
  } else if (tipos[i] == "numerical") {
    datos[[i]] <- as.numeric(datos[[i]])  # por si hay alguna variable mal leída
  }
}
```

Dummy

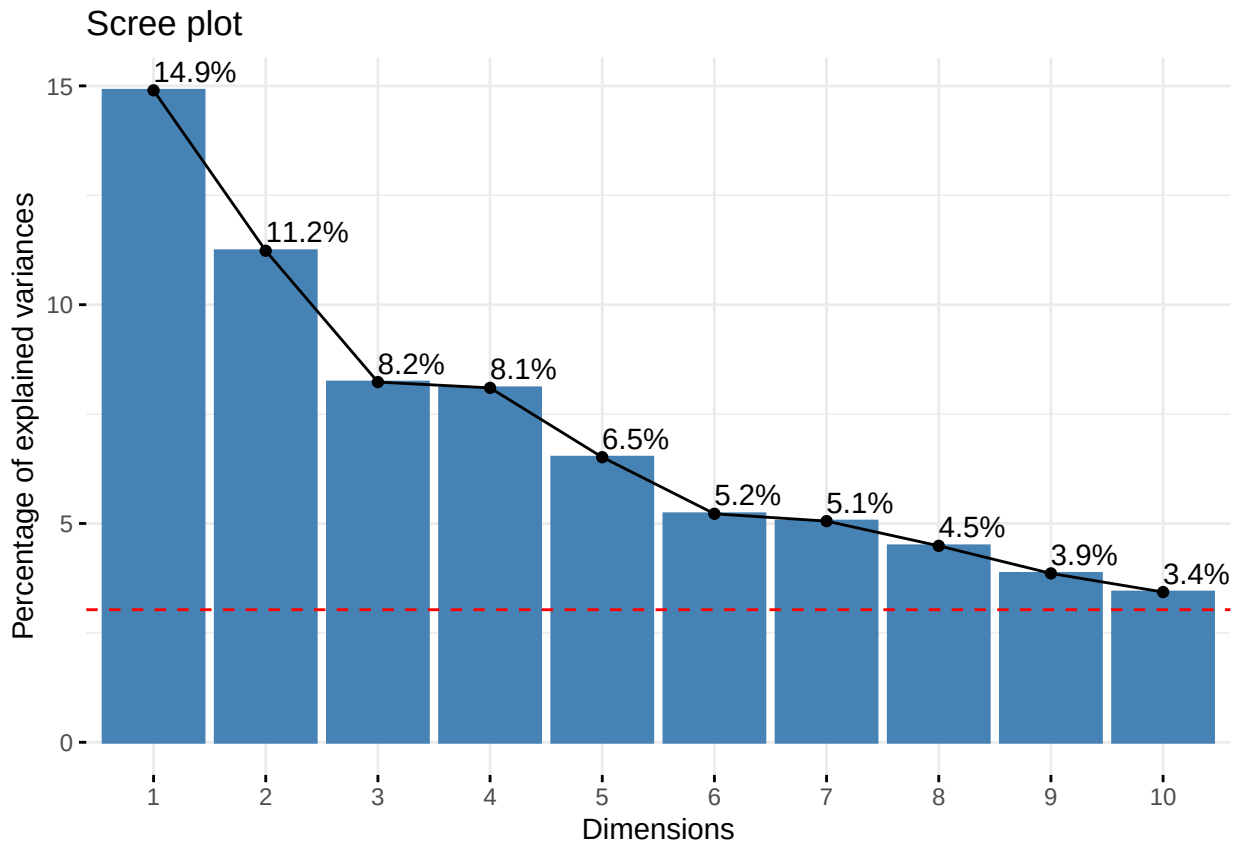
En este punto, se crean las variables dummy para aquellas que son de naturaleza categoricas.

```r
# 1. Lista de variables que TÚ quieres binarizar
mis_vars_a_binarizar <- c(
  "Sex",
  "Diabetes",
  "Cardiopathy",
  "Neurodegenerative_disease",
  "Histology",
  "Statins")

# 2. Verifica que existen y son tipo 'character'
datos[mis_vars_a_binarizar] <- lapply(datos[mis_vars_a_binarizar], as.factor)

#  3. Crear columnas dummy solo de esas, sin tocar el resto
datos_dummy_df <- dummy_cols(
  datos,
  select_columns = mis_vars_a_binarizar,
  remove_selected_columns = TRUE,
  remove_first_dummy = FALSE
)
```

Number PCA

Mediante el teorema del codo vamos a seleccionar el número óptimo de componentes.

```r
variables_excluir <- c("X1ª_eval", "Mejor_resp")

datos_PCA_numerico <- datos_dummy_df[, !(names(datos_dummy_df) %in% variables_excluir)]

# Convertir todo a numérico explícitamente
datos_PCA_numerico[] <- lapply(datos_PCA_numerico, function(x) as.numeric(as.character(x)))

# PCA
res.pca <- PCA(datos_PCA_numerico, scale.unit = TRUE, graph = FALSE, ncp = 10)

# Visualizar eigenvalues
eig.val <- get_eigenvalue(res.pca)
VPmedio <- 100 * (1 / nrow(eig.val))

fviz_eig(res.pca, addlabels = TRUE) +
  geom_hline(yintercept = VPmedio, linetype = 2, color = "red")
```




```
kable(eig.val[1:6,])
```

	eigenvalue	variance.percent	cumulative.variance.percent
Dim.1	12.067696	14.898390	14.89839
Dim.2	9.098331	11.232508	26.13090
Dim.3	6.667539	8.231530	34.36243
Dim.4	6.560376	8.099230	42.46166
Dim.5	5.277047	6.514872	48.97653
Dim.6	4.229797	5.221972	54.19850

```
K = 6
res.pca <- PCA(datos_PCA_numerico, scale.unit = TRUE, graph = FALSE, ncp = K)
```

Como podemos observar, el numero idoneo de componentes es 6.

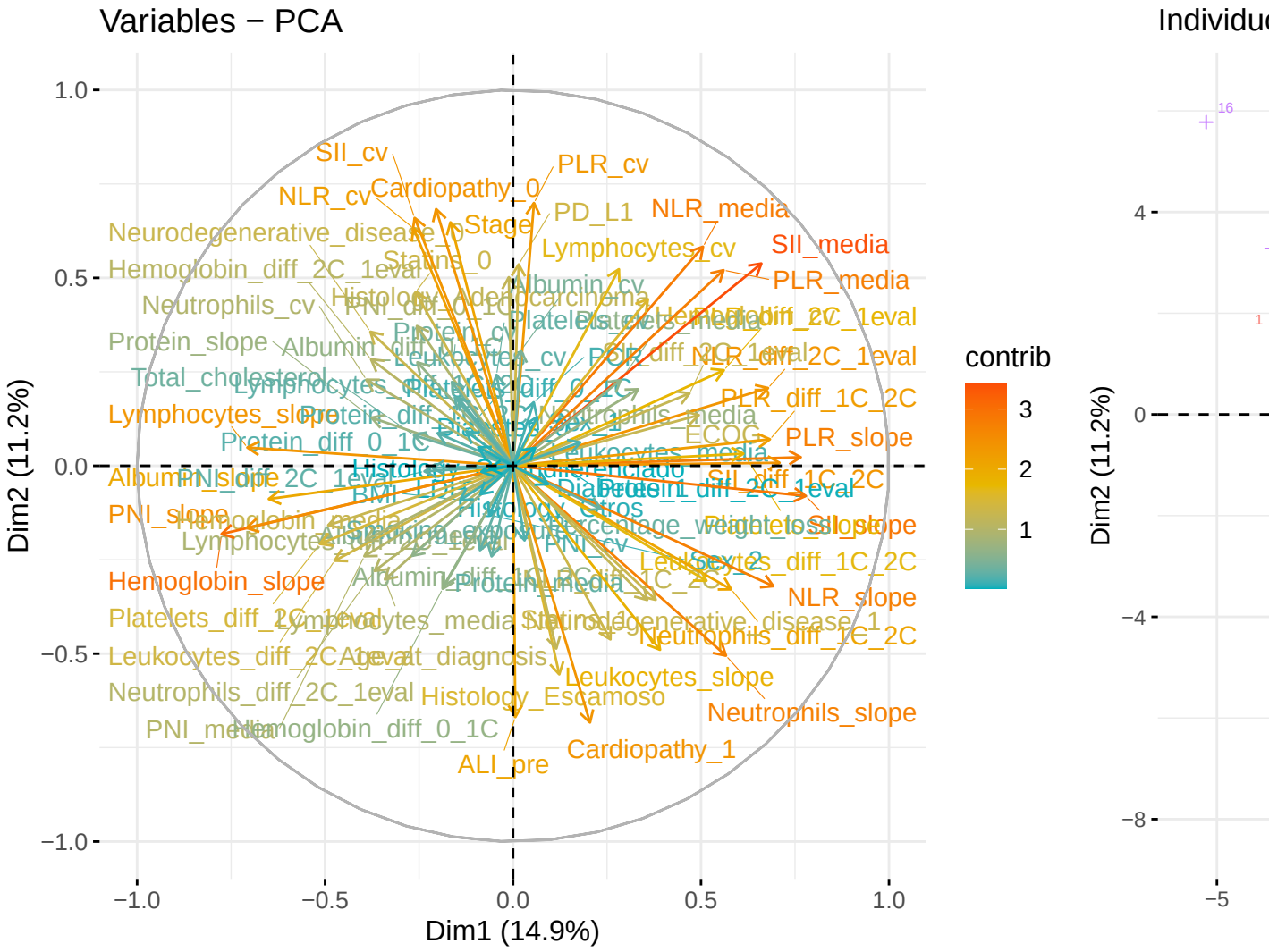
PCA Interpretation

Variables Vs Individuals

X1º_eval

En primer lugar, observamos los espacios de las componentes coloreando por la primera de nuestras variables respuesta.

```r
# Gráfico de variables
grafico_vars <- fviz_pca_var(
  res.pca,
  axes = c(1, 2),
  col.var = "contrib",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  select.var = list(contrib = 100),
  label = "all",
  repel = TRUE
) +
theme(text = element_text(size = 12))  # Aumenta tamaño general

# Definir factor de color (por ejemplo respuesta clínica)
colorear_factor <- datos_dummy_df$'X1ª_eval'

# Gráfico de individuos mejorado
grafico_inds <- fviz_pca_ind(
  res.pca, axes = c(1, 2),
  geom = c("point", "text"),
  repel = TRUE,
  labelsize = 2,
  select.ind = list(cos2 = 100),  # Seleccionar los 30 individuos mejor representados
  habillage = colorear_factor,  # Color según grupo
  addEllipses = FALSE             # Añadir elipses de confianza
) +
  coord_fixed() +
  ggtitle('Individuos mejor representados (Top 30)')

# Combinar ambos gráficos
grafico_vars + grafico_inds
```



La alineación de ciertas variables con los individuos PE (progresión evidente) sugiere que una **evolución positiva (en aumento) de biomarcadores inflamatorios** como el SII, PLR, NLR está estrechamente asociada con un peor pronóstico. Es decir, los pacientes en los que estos indicadores aumentan a lo largo del tiempo tienden a presentar progresión evidente de la enfermedad, lo que podría reflejar una respuesta inflamatoria sistémica desfavorable frente al tratamiento con pembrolizumab.

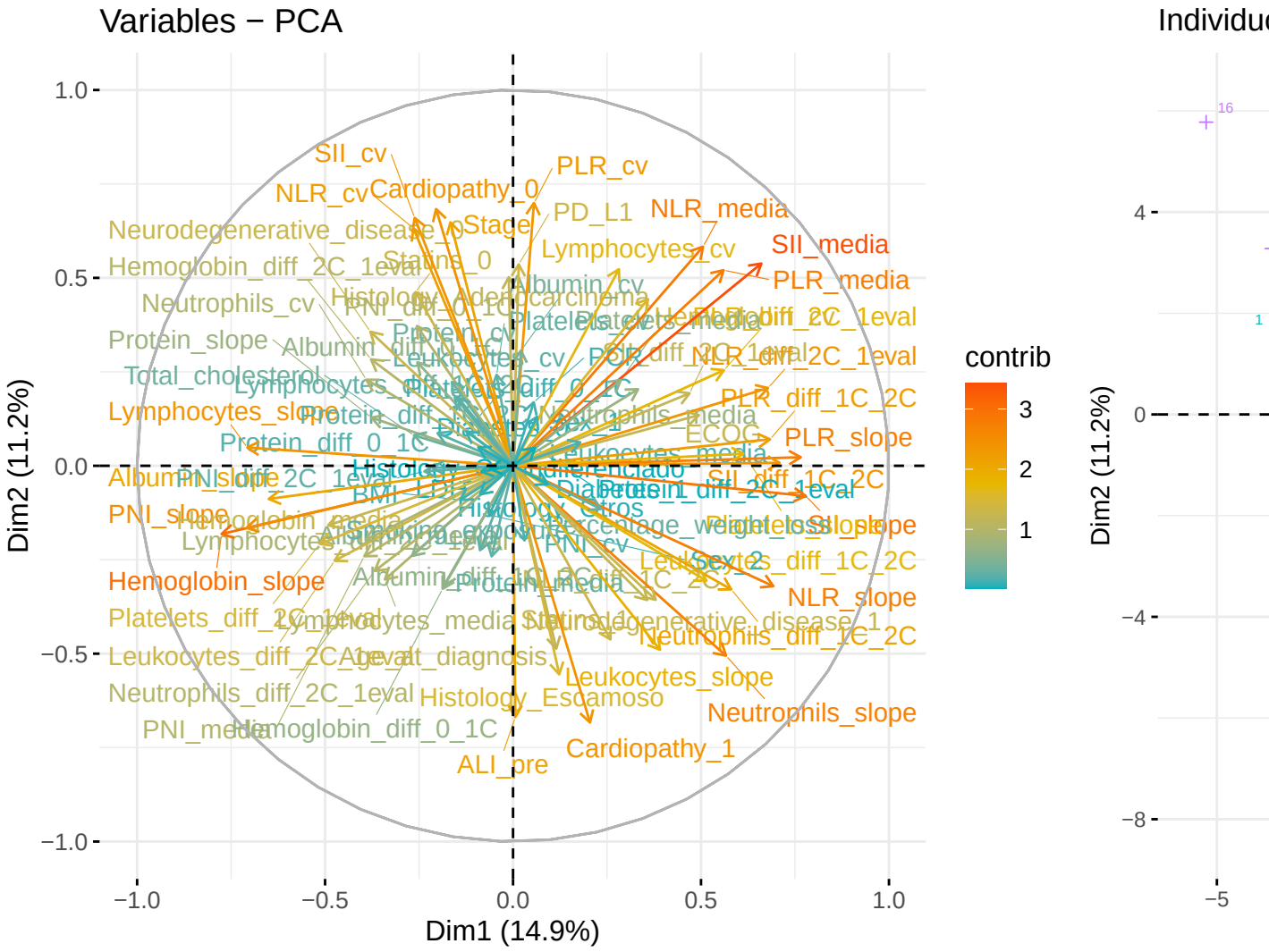
Por otro lado, los pacientes del grupo **RP** (respuesta parcial) se sitúan mayoritariamente alejados de estas variables, en una zona opuesta a su dirección, lo que sugiere que tienen perfiles más estables o incluso mejoran en estos biomarcadores con el tiempo. La dispersión de los individuos del grupo **EE** podría indicar heterogeneidad dentro de esa categoría, o que otras variables no capturadas en estas dos dimensiones son relevantes para explicar sus respuestas.

La evolución temporal de ciertos biomarcadores inflamatorios (SII_slope, PLR_slope, NLR_slope) al parecer no solo tiene un peso importante en la variabilidad total del conjunto de datos, sino que también permite **discriminar grupos clínicos**. En particular, una evolución ascendente de estos índices se asocia con peor evolución clínica. Por tanto, estos indicadores dinámicos podrían emplearse como **potenciales marcadores pronósticos o predictivos** en el seguimiento de pacientes tratados con inmunoterapia.

Mejor_resp

Ahora seguimos los mismos pasos para la segunda varible respuesta.

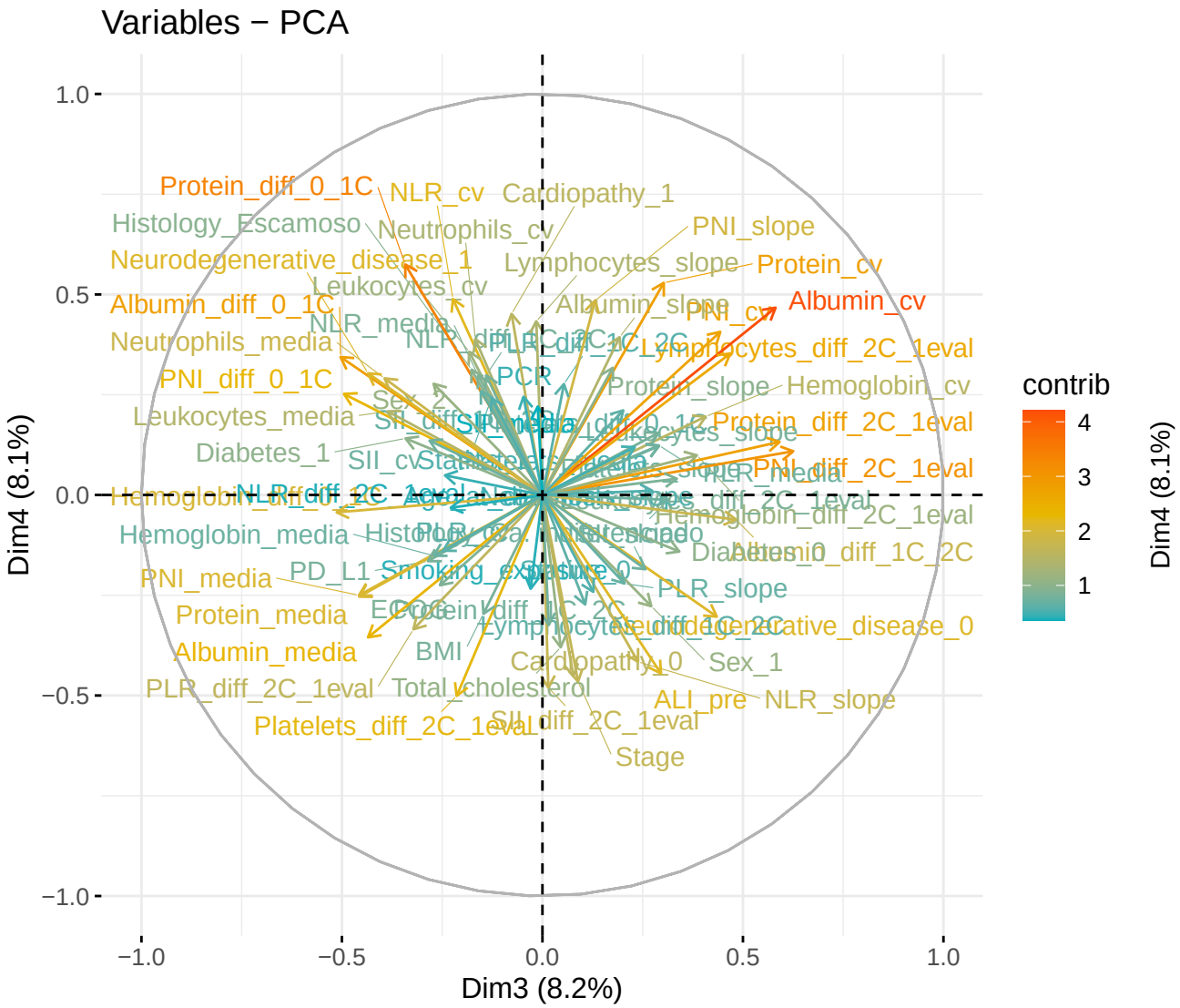
```
# Gráfico de variables
grafico_vars <- fviz_pca_var(
  res.pca,
  axes = c(1, 2),
  col.var = "contrib",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  select.var = list(contrib = 100),
  label = "all",
  repel = TRUE
) +
theme(text = element_text(size = 12))  # Aumenta tamaño general

# Definir factor de color (por ejemplo respuesta clínica)
colorear_factor <- datos_dummy_df$'Mejor_resp'

# Gráfico de individuos mejorado
grafico_inds <- fviz_pca_ind(
  res.pca, axes = c(1, 2),
  geom = c("point", "text"),
  repel = TRUE,
  labelsize = 2,
  select.ind = list(cos2 = 100),  # Seleccionar los 30 individuos mejor representados
  habillage = colorear_factor,  # Color según grupo
  addEllipses = FALSE             # Añadir elipses de confianza
) +
  coord_fixed() +
  ggtitle('Individuos mejor representados (Top 30)')

# Combinar ambos gráficos
grafico_vars + grafico_inds
```



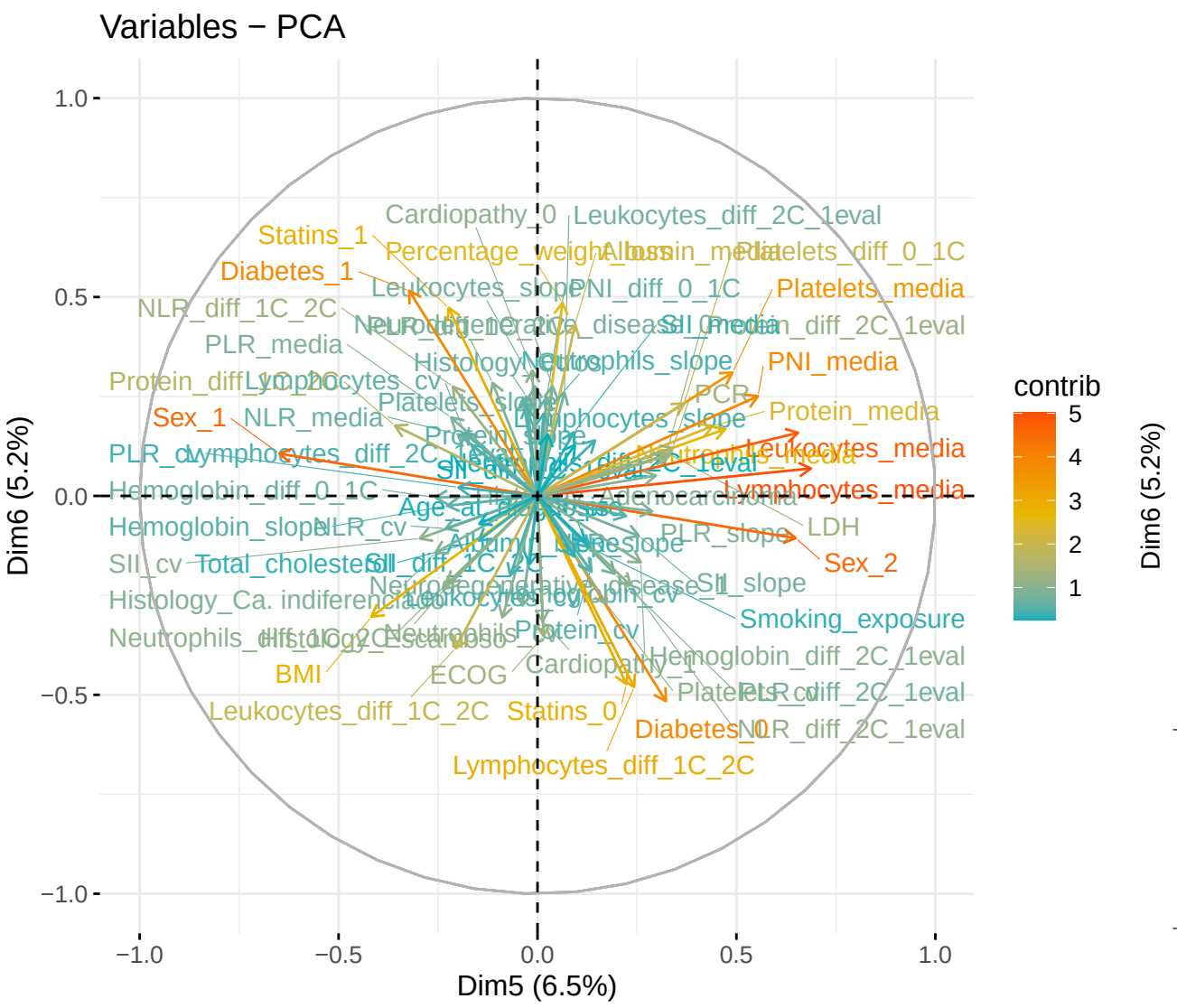
Este gráfico confirma que las **tendencias en los biomarcadores inflamatorios a lo largo del tiempo** son determinantes en la evolución clínica de los pacientes tratados con inmunoterapia. En particular, un aumento sostenido de SII, PLR o NLR se asocia con peor pronóstico (PE), mientras que su estabilidad o descenso se relaciona con mejores respuestas (RC y RP). Esta evidencia respalda el uso de variables dinámicas como herramientas pronósticas en oncología de precisión.

```
library(patchwork)  # asegúrate de tenerlo instalado

# Gráfico de variables
grafico_vars <- fviz_pca_var(
  res.pca,
  axes = c(3, 4),
  col.var = "contrib",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  select.var = list(contrib = 70),
  label = "all",
  repel = TRUE
```

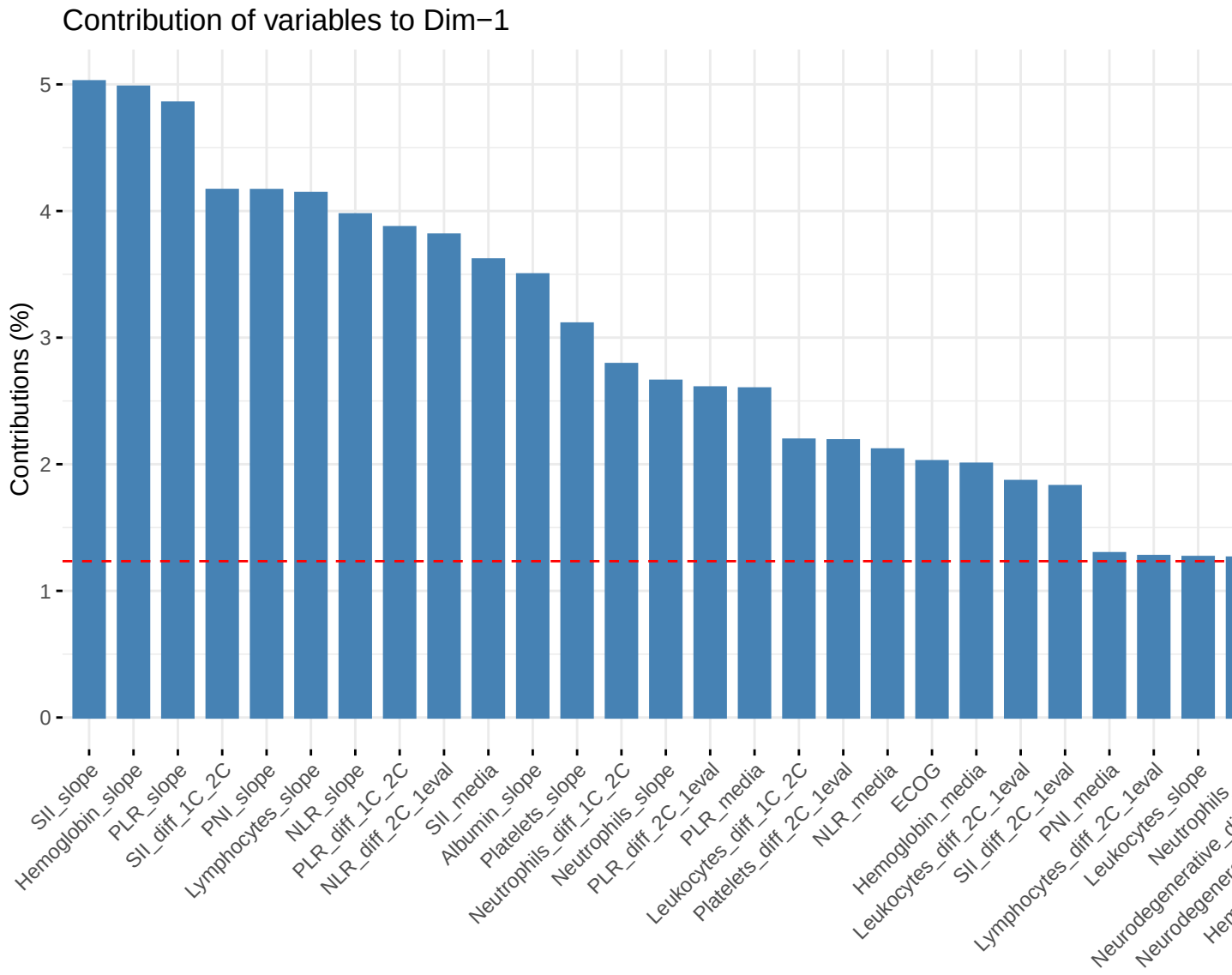
```
) +
theme(text = element_text(size = 12))  # Aumenta tamaño general

# Gráfico de individuos
grafico_inds <- fviz_pca_ind(
  res.pca, axes = c(3, 4), geom = "point",
  habillage = colorear_factor
) +
  coord_fixed() +
  ggtitle('Individuos')

# Combinar ambos gráficos
grafico_vars + grafico_inds  # Esto los pone lado a lado
```



```r
library(patchwork)  # asegúrate de tenerlo instalado

# Gráfico de variables
grafico_vars <- fviz_pca_var(
  res.pca,
  axes = c(5, 6),
  col.var = "contrib",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  select.var = list(contrib = 70),
  label = "all",
  repel = TRUE
) +
theme(text = element_text(size = 12))  # Aumenta tamaño general

# Gráfico de individuos
grafico_inds <- fviz_pca_ind(
  res.pca, axes = c(5, 6),
  geom = "point",
  habillage = colorear_factor
) +
  coord_fixed() +
  ggtitle('Individuos')

# Combinar ambos gráficos
grafico_vars + grafico_inds  # Esto los pone lado a lado
```




```
fviz_contrib(res.pca, choice = "var", axes = 1, top = 50)
```



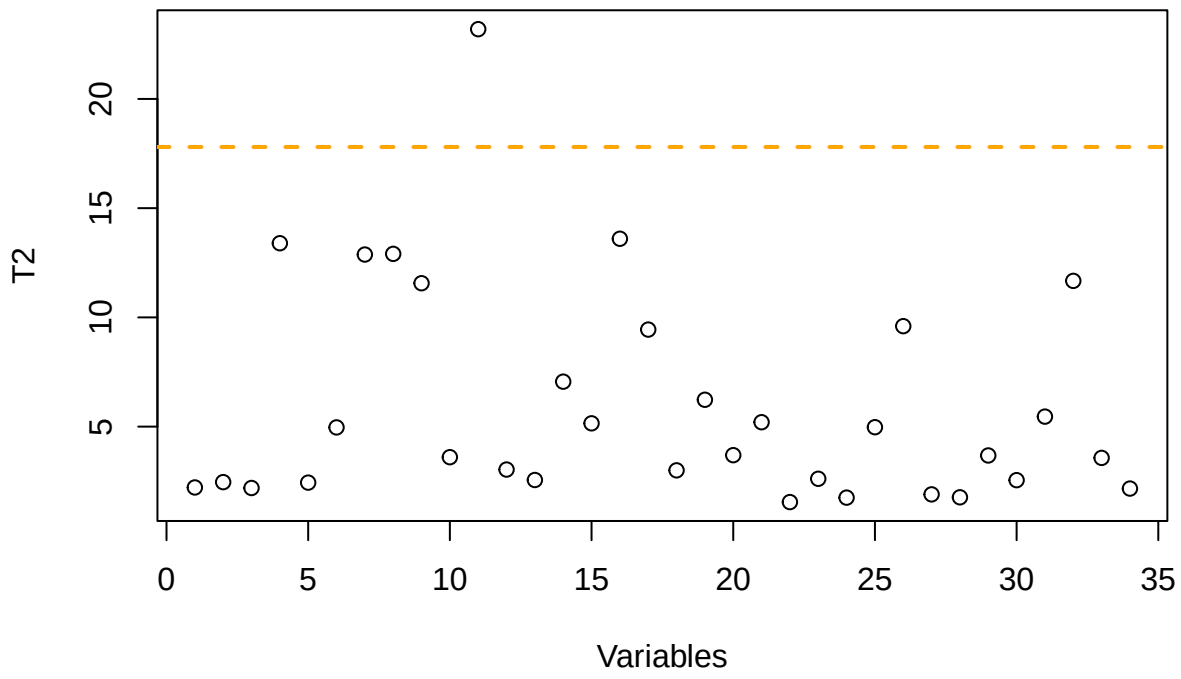

```
#fviz_contrib(res.pca, choice = "var", axes = 2, top = 50)
#fviz_contrib(res.pca, choice = "var", axes = 3, top = 50)
#fviz_contrib(res.pca, choice = "var", axes = 4, top = 50)
#fviz_contrib(res.pca, choice = "var", axes = 5, top = 50)
#fviz_contrib(res.pca, choice = "var", axes = 6, top = 50)
```

El gráfico muestra que las principales variables que contribuyen a la Dimensión 1 del PCA están relacionadas con índices inflamatorios como PLR slope SII slope y NLR slope además de diferencias en biomarcadores como la hemoglobina y albúmina entre distintos momentos de evaluación lo que sugiere que la variabilidad principal de los datos está fuertemente influenciada por cambios en parámetros inmunológicos y nutricionales

Validation

T2-Hotelling

```
# Gráfico T2 Hotelling
misScores = res.pca$ind$coord[,1:K]
miT2 = colSums(t(misScores**2)/eig.val[1:K,1])
I = nrow(datos_dummy_df)
F95 = K*(I**2 - 1)/(I*(I - K)) * qf(0.95, K, I-K)
F99 = K*(I**2 - 1)/(I*(I - K)) * qf(0.99, K, I-K)

plot(1:length(miT2), miT2, type = "p", xlab = "Variables", ylab = "T2")
abline(h = F95, col = "orange", lty = 2, lwd = 2)
abline(h = F99, col = "red3", lty = 2, lwd = 2)
```



Tan solo 1 de las variables sobrepasa el 95%

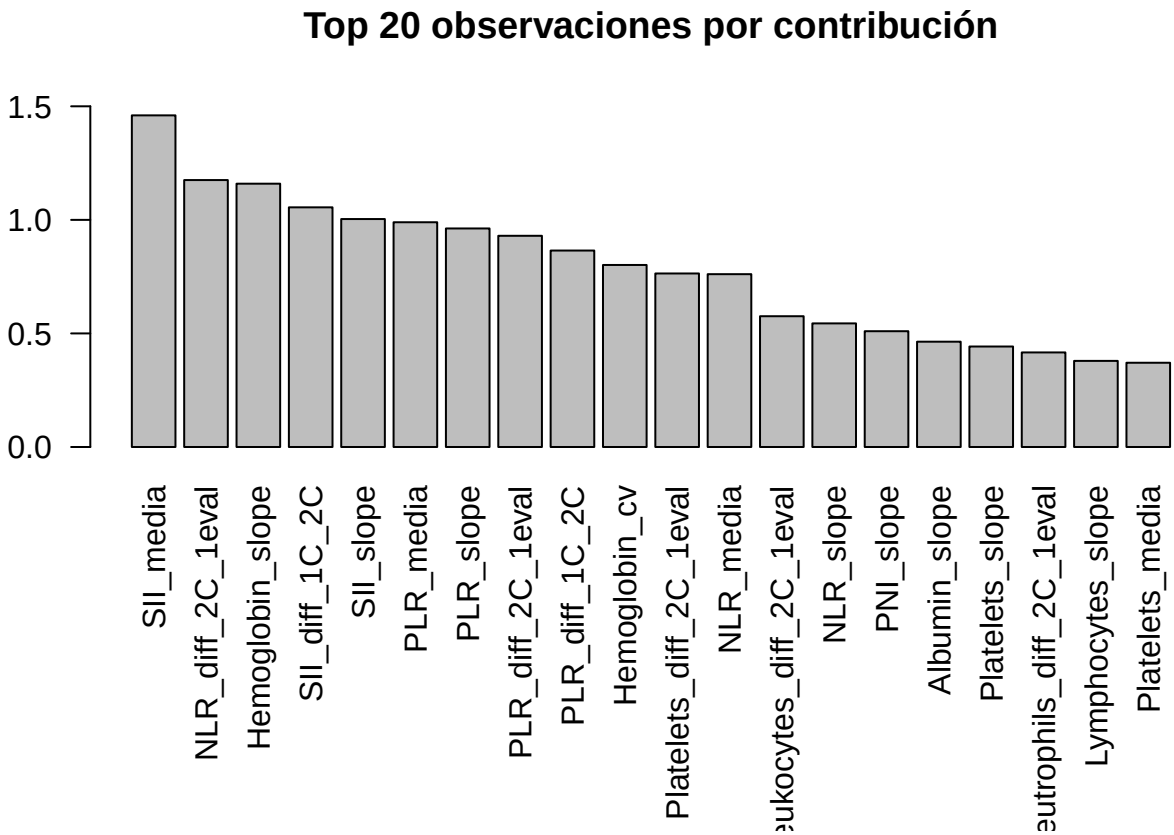
```
anomalas = which(miT2 > F95)
anomalas
```

```
## 11
## 11
```

```r
# Fuction
contribT2 = function (X, scores, loadings, eigenval, observ, cutoff = 2) {
  # X is data matrix and must be centered (or centered and scaled if data were scaled)
  misScoresNorm = t(t(scores**2) / eigenval)
  misContrib = NULL
  for (oo in observ) {
    print(rownames(scores)[oo])
    print(scores[oo,])
    misPCs = which(as.numeric(misScoresNorm[oo,]) > cutoff)
    lacontri = sapply(misPCs, function (cc) (scores[oo,cc]/eigenval[cc])*loadings[,cc]*X[oo,])
    lacontri = rowSums((1*(sign(lacontri) == 1))*lacontri)
    misContrib = cbind(misContrib, lacontri)
  }
  colnames(misContrib) = rownames(misScoresNorm[observ,])
  return(misContrib)
}

#
data_T = datos[,descDatos$tipo == "numerical"]
data_T = data_T[,setdiff(colnames(data_T), c("rating", "weight", "cups"))]
data_T = scale(datos_PCA_numerico, center = TRUE, scale = TRUE)
X = as.matrix(data_T)
# Calculamos los loadings a partir de las coordenadas de las variables
# ya que la librería FactoMineR nos devuelve los loadings ponderados
# por la importancia de cada componente principal.
misLoadings = sweep(res.pca$var$coord, 2, sqrt(res.pca$eig[1:K,1]), FUN="/")
# Calculamos las contribuciones
mycontrisT2 = contribT2(X = X, scores = misScores, loadings = misLoadings,
                        eigenval = eig.val[1:K,1], observ = which.max(miT2),
                        cutoff = 2)
```

```
## [1] "11"
##     Dim.1     Dim.2     Dim.3     Dim.4     Dim.5     Dim.6
## 13.875654  6.408976  1.297930  1.229944  2.931267 -1.609851
```

```r
top_n <- 20

importantes <- order(mycontrisT2[,1], decreasing = TRUE)[1:top_n]

par(mar = c(10,2.3,3,1))
barplot(mycontrisT2[importantes, 1], las = 2,
        main = paste0("Top ", top_n, " observaciones por contribución"),
        ylim = c(0, max(mycontrisT2[importantes,1])*1.1))
```



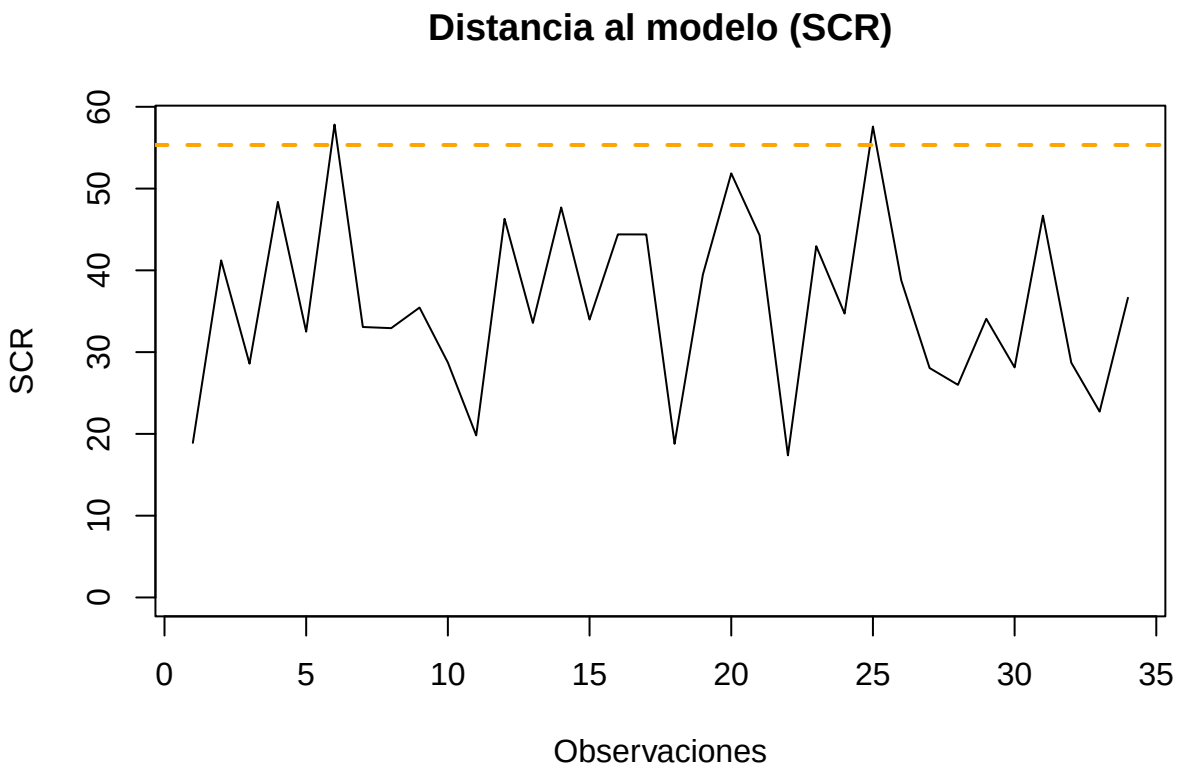
Como podemos observar, a nivel de variables inflamatorias, el paciente 11 se diferencia del resto.

Es decir, sus valores de SII_media, NLR_diff_2C_1eval, etc. . . , son diferentes a la del resto, al tener tan poco pacientes, no podemos permitirnos su eliminación.

SRC

```
# 1. Detectar columnas con NA en X
vars_con_na <- colnames(X)[apply(X, 2, function(x) any(is.na(x)))]

# 2. Filtrar solo variables sin NA
vars_validas <- setdiff(colnames(X), vars_con_na)

# 3. Filtrar X y misLoadings para quedarnos solo con columnas válidas
X_limpio <- X[, vars_validas]
misLoadings_limpio <- misLoadings[vars_validas, ]

# 4. Verificar dimensiones compatibles
if (ncol(X_limpio) == nrow(misLoadings_limpio)) {

  # 5. Recalcular error de reconstrucción y SCR
  myE <- X_limpio - misScores %*% t(misLoadings_limpio)
  mySCR <- rowSums(myE^2)

  # 6. Graficar SCR válidos
```

```r
  idx_validos <- which(is.finite(mySCR))

  if (length(idx_validos) > 0) {
    plot(idx_validos, mySCR[idx_validos], type = "l",
         main = "Distancia al modelo (SCR)",
         ylab = "SCR", xlab = "Observaciones",
         ylim = c(0, max(mySCR, na.rm = TRUE)))

    # Opcional: límites Chi-cuadrado
    g <- var(mySCR, na.rm = TRUE) / (2 * mean(mySCR, na.rm = TRUE))
    h <- (2 * mean(mySCR, na.rm = TRUE)^2) / var(mySCR, na.rm = TRUE)

    chi2lim <- g * qchisq(0.95, df = h)
    chi2lim99 <- g * qchisq(0.99, df = h)

    abline(h = chi2lim, col = "orange", lty = 2, lwd = 2)
    abline(h = chi2lim99, col = "red3", lty = 2, lwd = 2)

  } else {
    message("No hay valores finitos de SCR para graficar.")
  }

} else {
  stop("Las dimensiones de X y loadings no coinciden tras limpiar.")
}
```



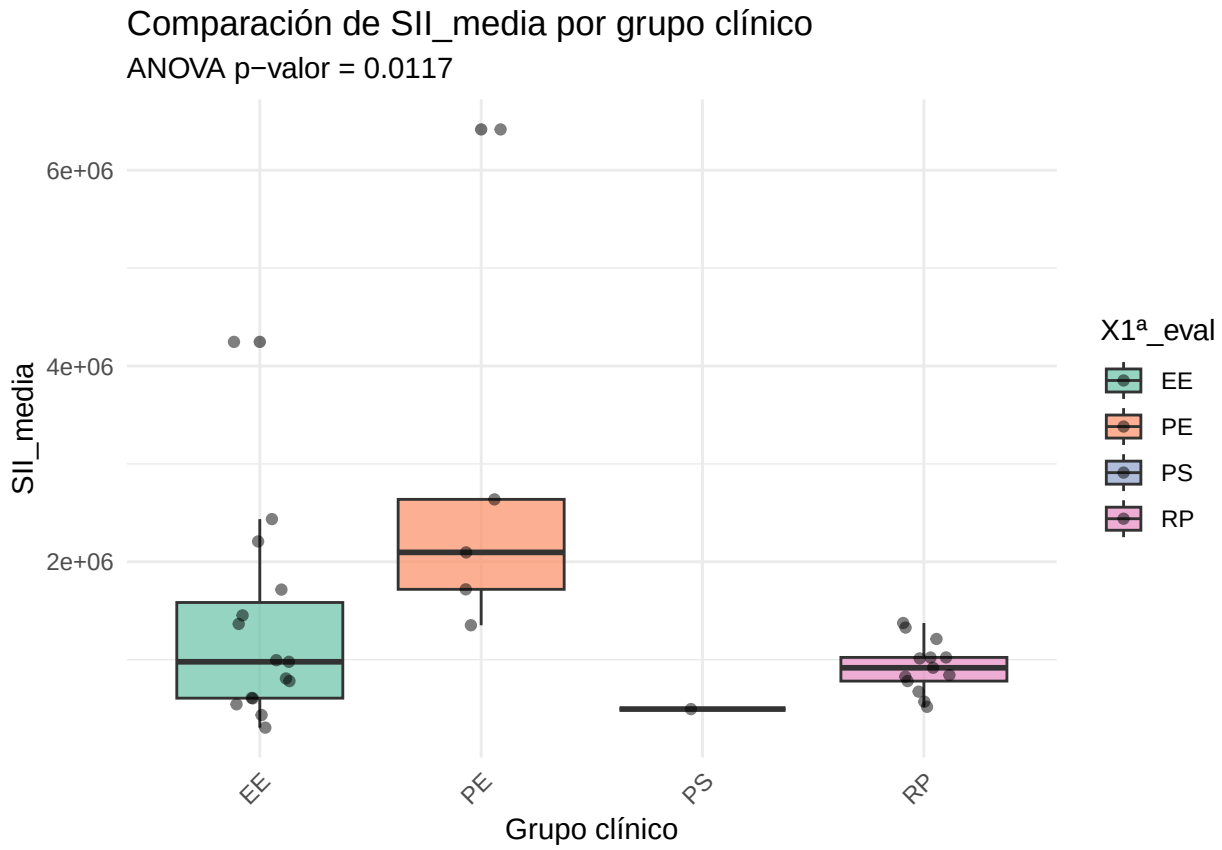
Una vez mas, observamos que tan solo es el paciente 11 el que sobrepasa el nivel del 95%.

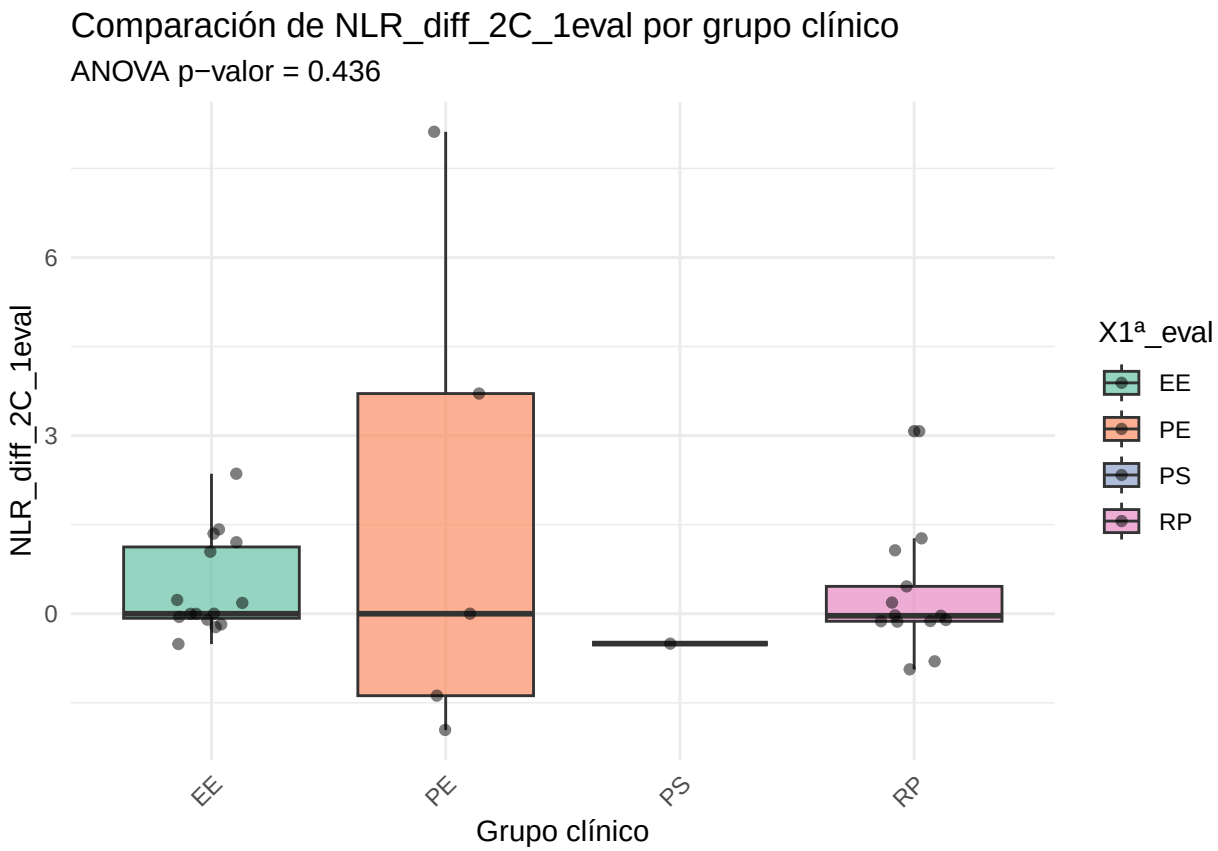
ANOVA

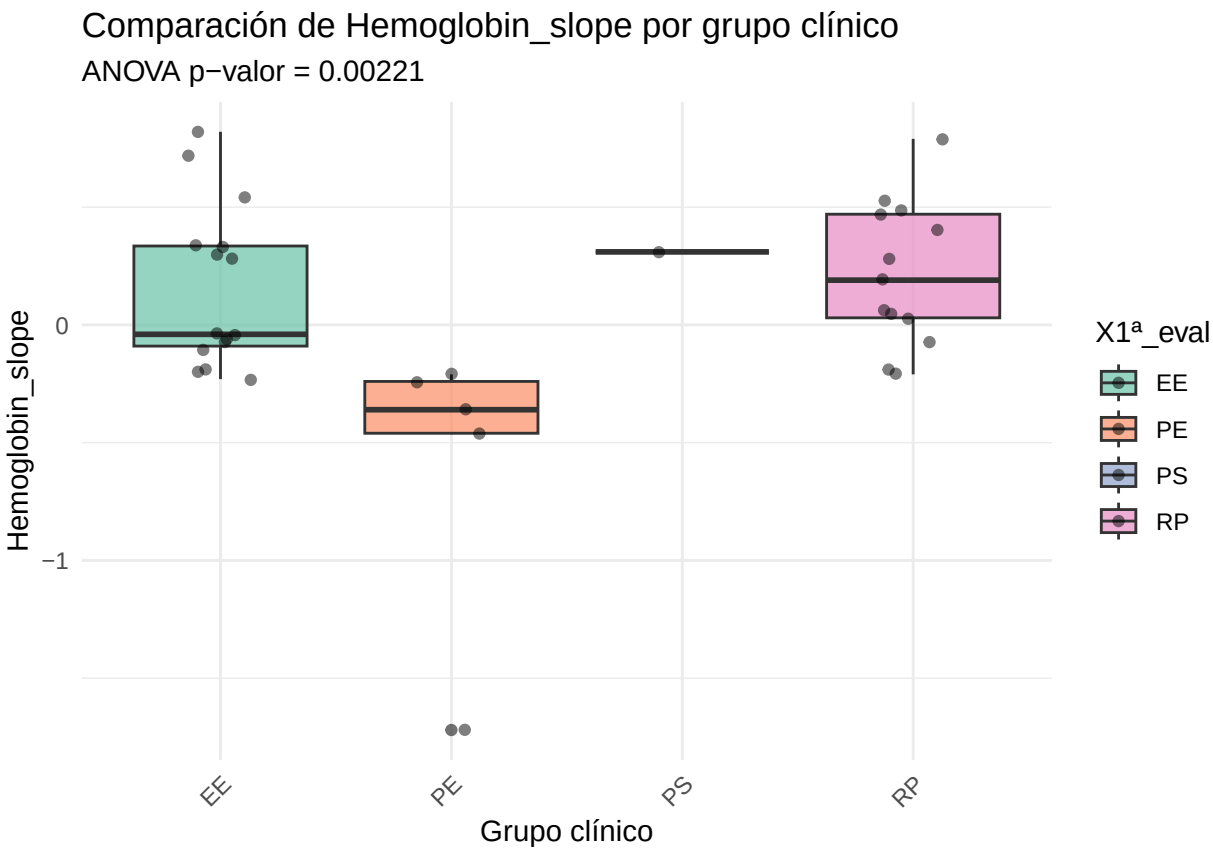
```
vars_dim1 <- c(
  "SII_media", "NLR_diff_2C_1eval", "Hemoglobin_slope", "SII_diff_1C_2C", "SII_slope",
  "PLR_media", "PLR_slope", "PLR_diff_2C_1eval", "PLR_diff_1C_2C", "Hemoglobin_cv",
  "Platelets_diff_2C_1eval", "NLR_media", "Leukocytes_diff_2C_1eval", "NLR_slope",
  "PNI_slope", "Albumin_slope", "Platelets_slope", "Neutrophils_slope",
  "Lymphocytes_slope", "Platelets_media"
)
# Inicializar un data frame vacío para guardar resultados
resultados_anova <- data.frame(
  Variable = character(),
  P_valor = numeric(),
  stringsAsFactors = FALSE
)

# Bucle para calcular ANOVA y guardar p-valores
for (var in vars_dim1) {
  formula_anova <- as.formula(paste(var, "~ X1ª_eval"))
  modelo <- aov(formula_anova, data = datos)
  p_valor <- summary(modelo)[[1]][["Pr(>F)"]][1]

  # Guardar en el dataframe
  resultados_anova <- rbind(
    resultados_anova,
    data.frame(Variable = var, P_valor = p_valor)
  )

  # Mostrar gráfico (opcional)
  p <- ggplot(datos, aes_string(x = "X1ª_eval", y = var, fill = "X1ª_eval")) +
    geom_boxplot(alpha = 0.7) +
    geom_jitter(width = 0.15, alpha = 0.5) +
    labs(
      title = paste("Comparación de", var, "por grupo clínico"),
      subtitle = paste("ANOVA p-valor =", signif(p_valor, 3)),
      x = "Grupo clínico", y = var
    ) +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    scale_fill_brewer(palette = "Set2")

  print(p)
}
```

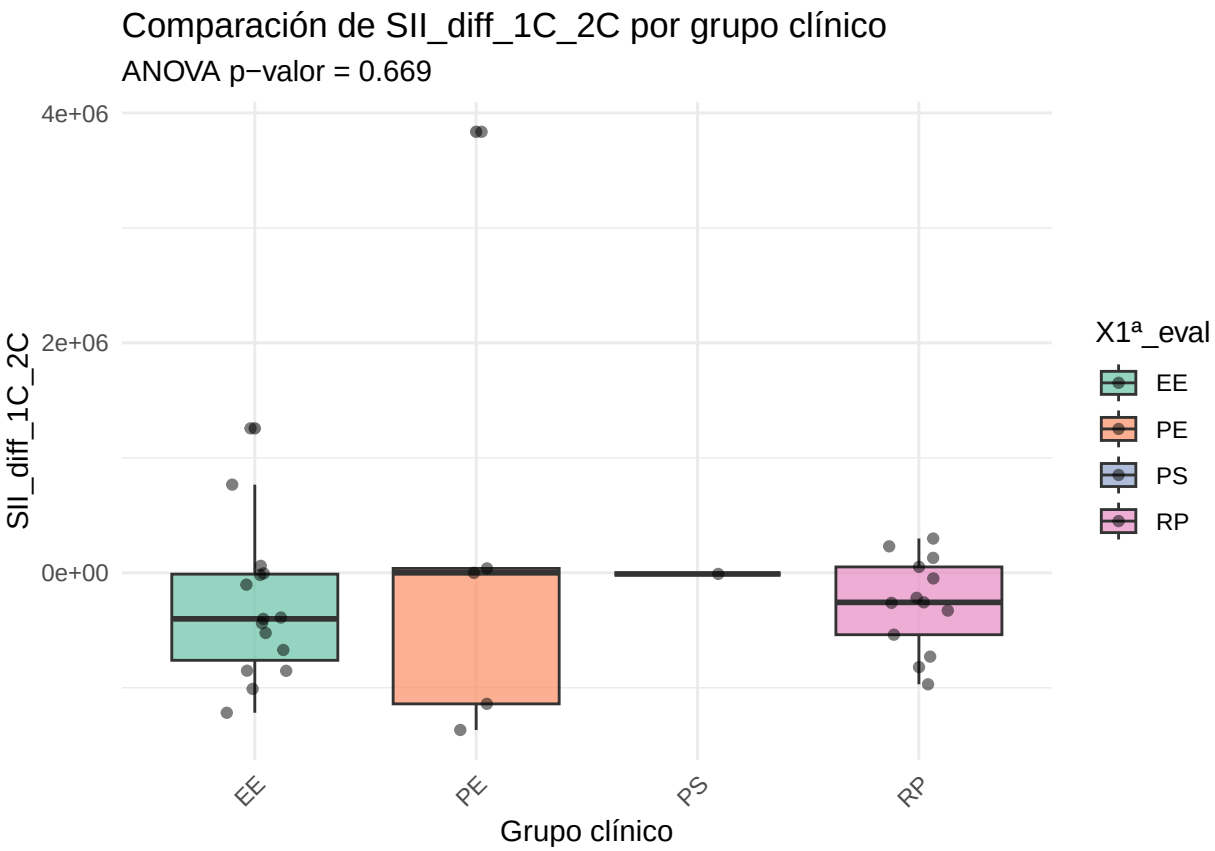
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
```

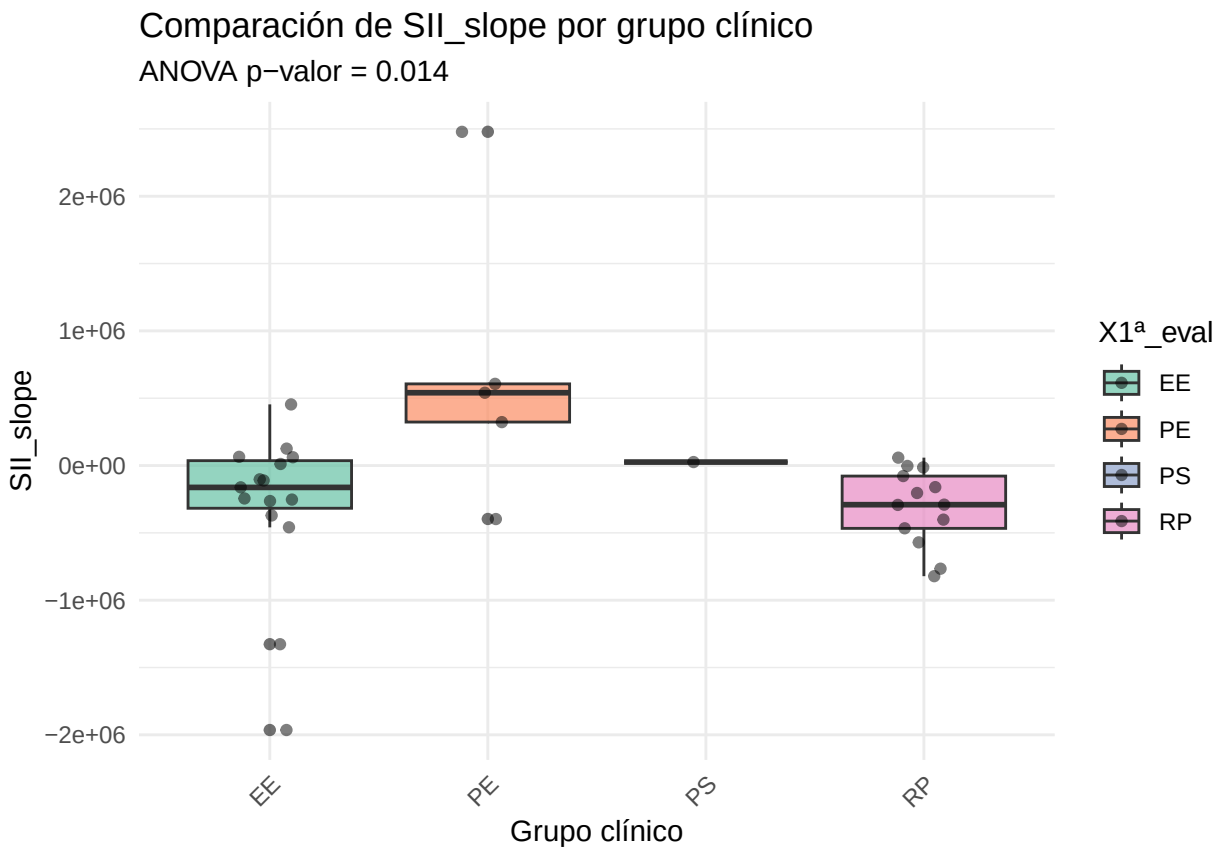
```
## generated.
```

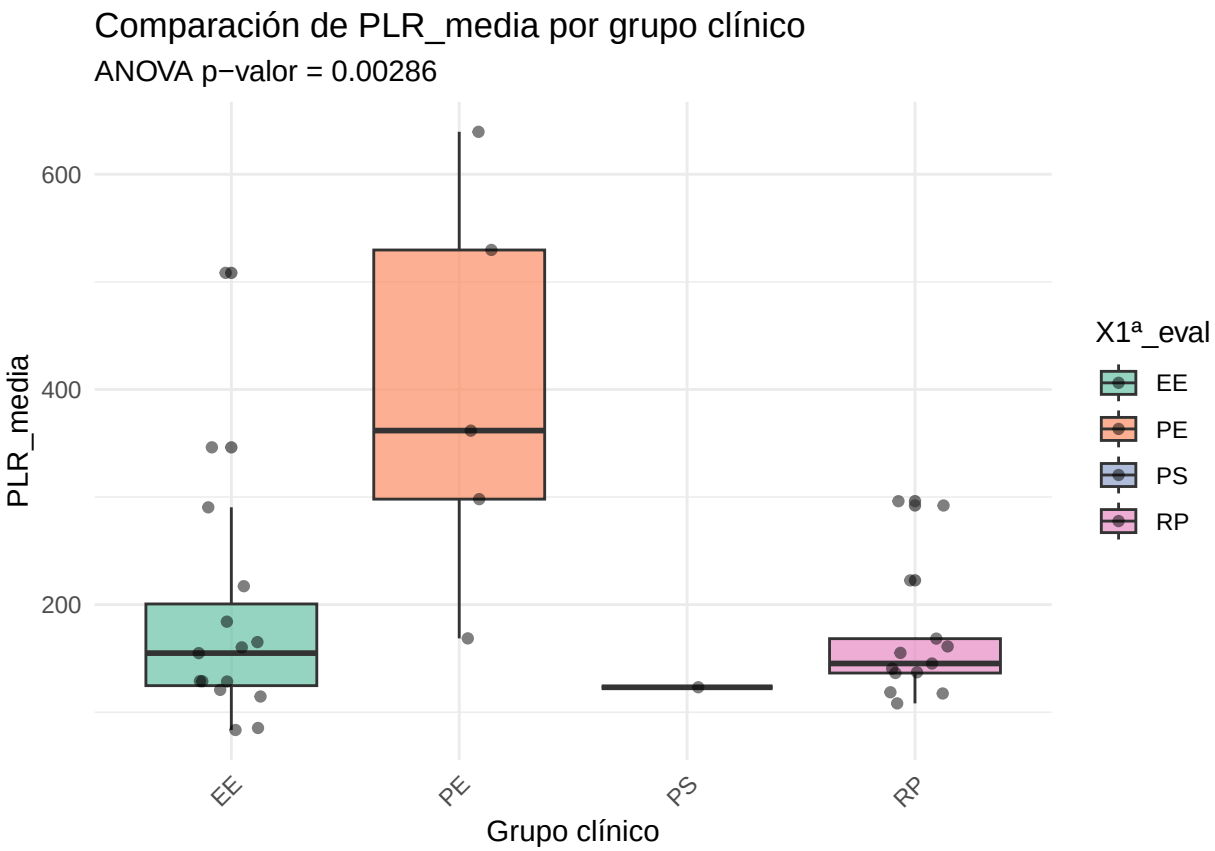

Comparación de SII_media por grupo clínico
ANOVA p−valor = 0.0117
SII_media
6e+06
4e+06
2e+06
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

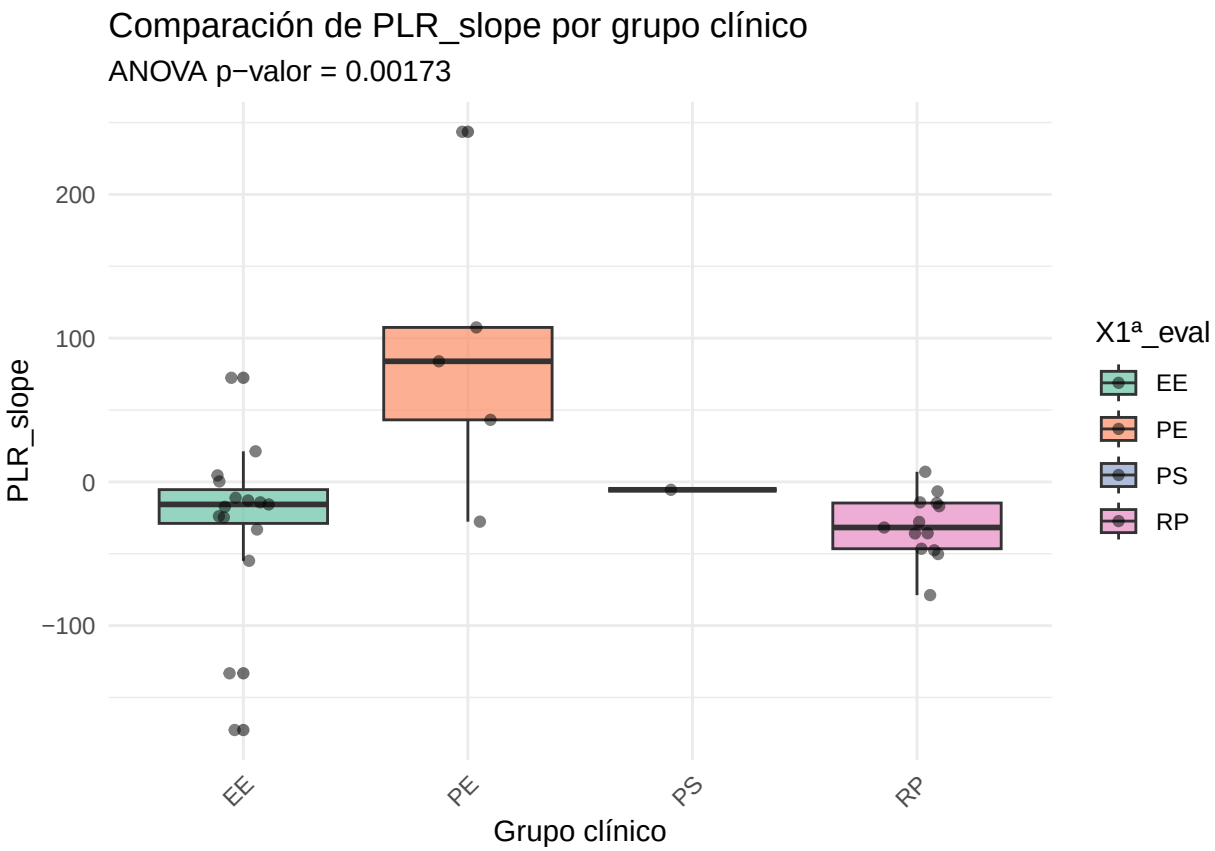

Comparación de NLR_diff_2C_1eval por grupo clínico
ANOVA p−valor = 0.436
NLR_diff_2C_1eval
6
3
0
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

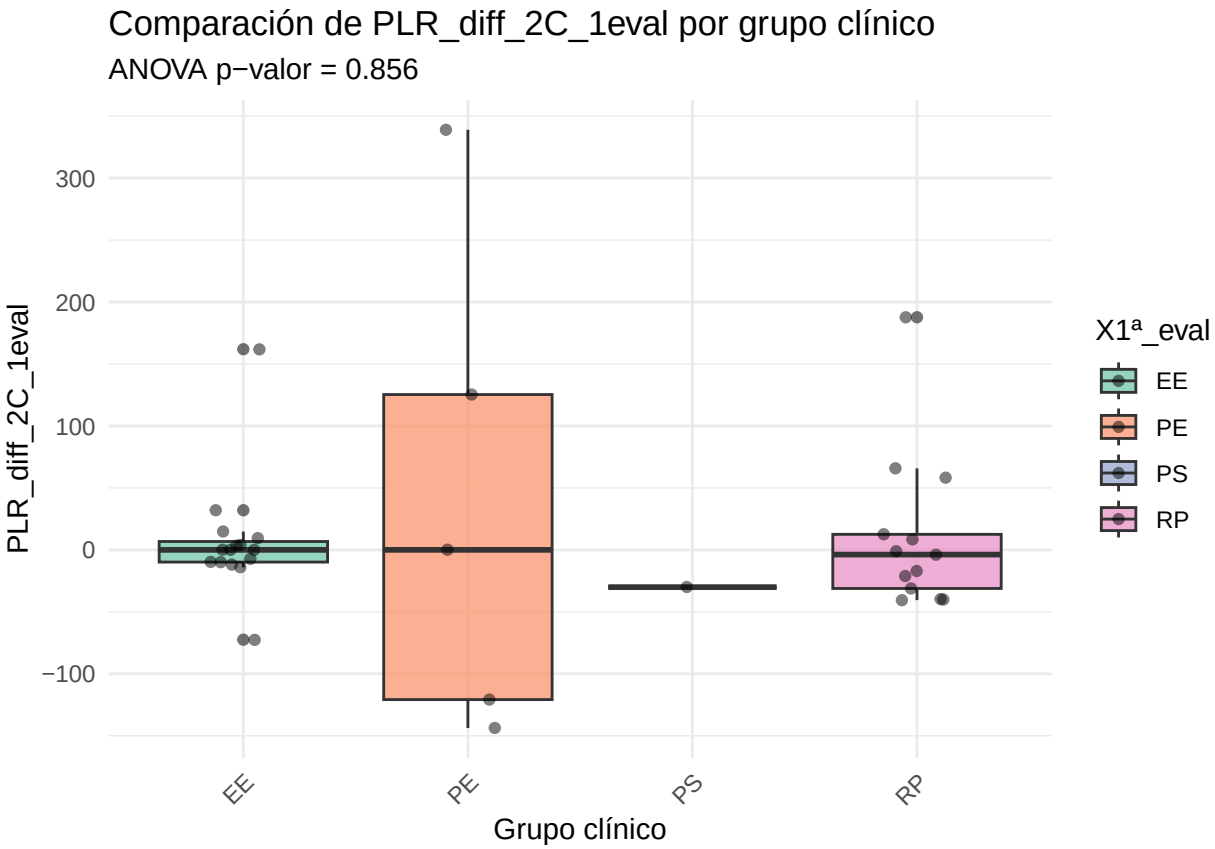

Comparación de Hemoglobin_slope por grupo clínico
ANOVA p−valor = 0.00221
Hemoglobin_slope
0
−1
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

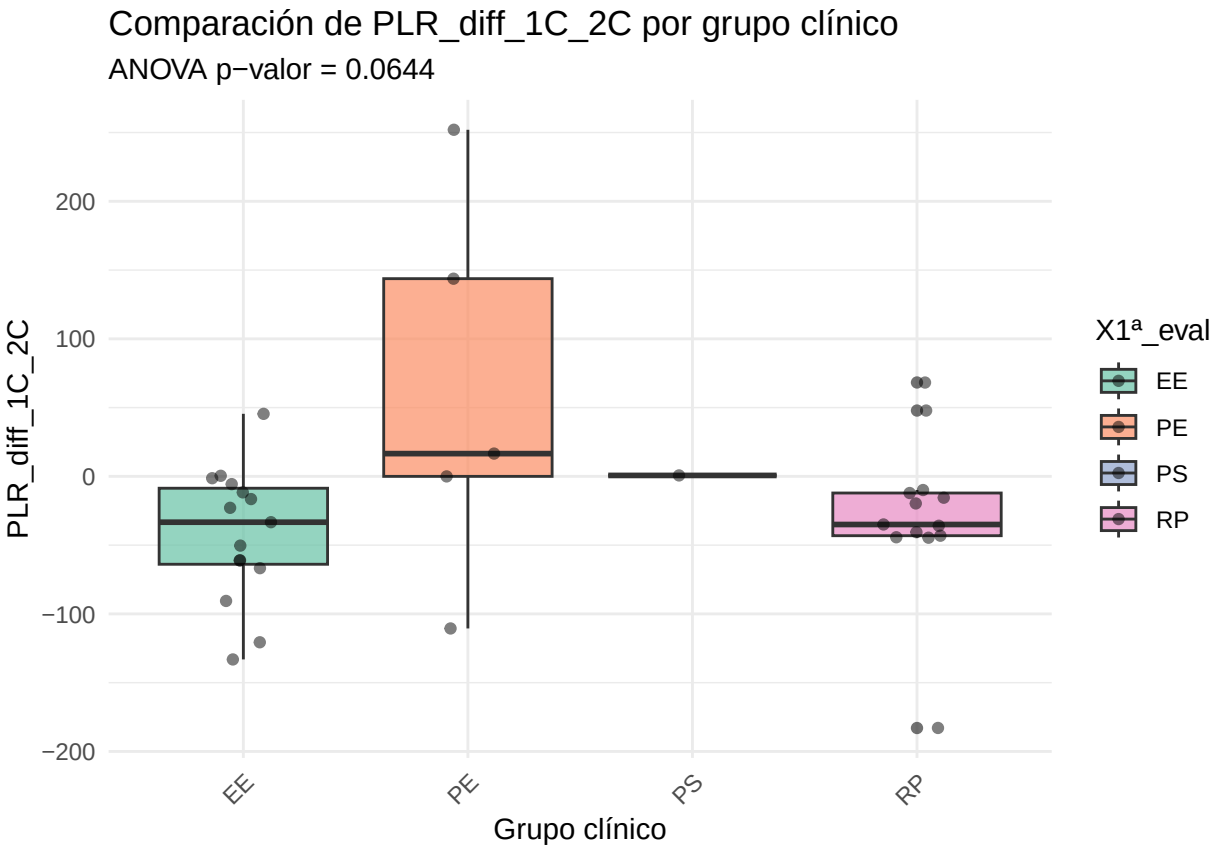

Comparación de SII_diff_1C_2C por grupo clínico
ANOVA p−valor = 0.669
4e+06
2e+06
0e+00
SII_diff_1C_2C
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

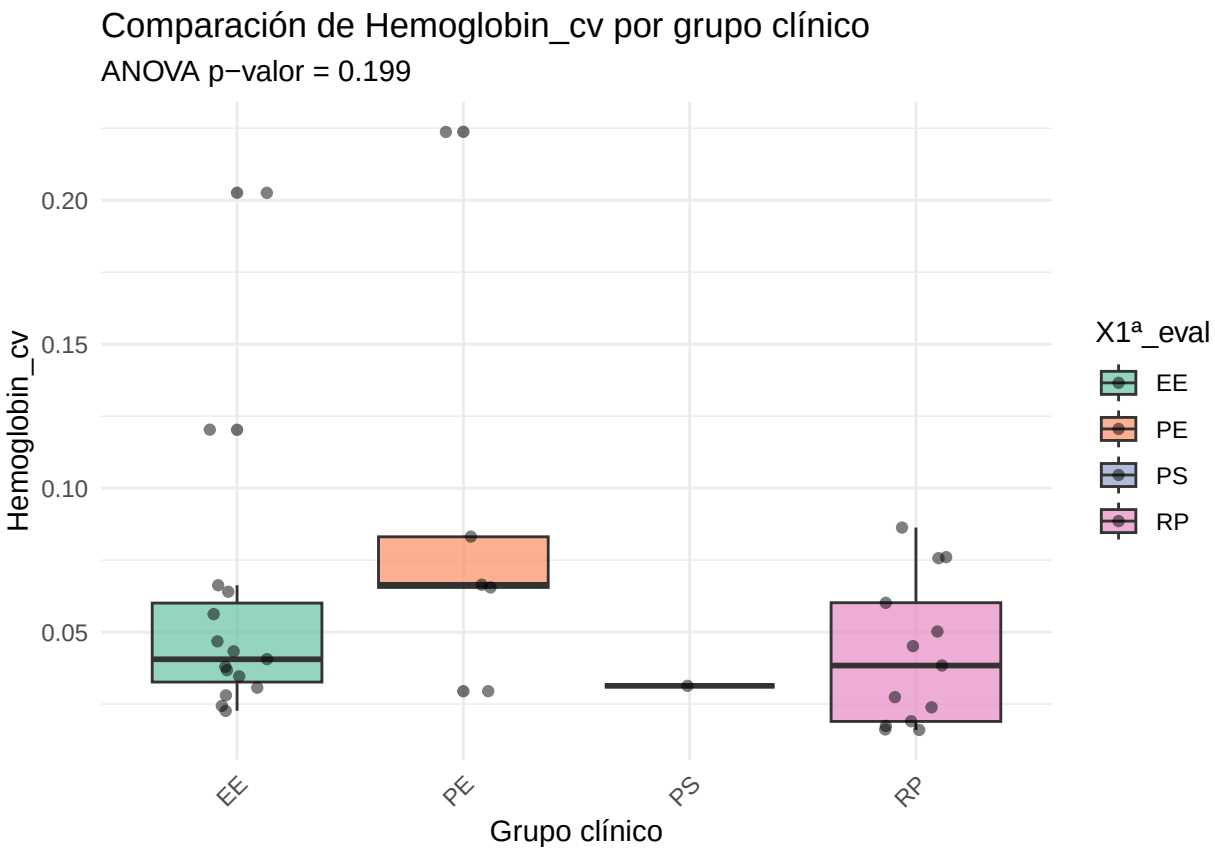

Comparación de SII_slope por grupo clínico
ANOVA p−valor = 0.014
SII_slope
2e+06
1e+06
0e+00
−1e+06
−2e+06
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

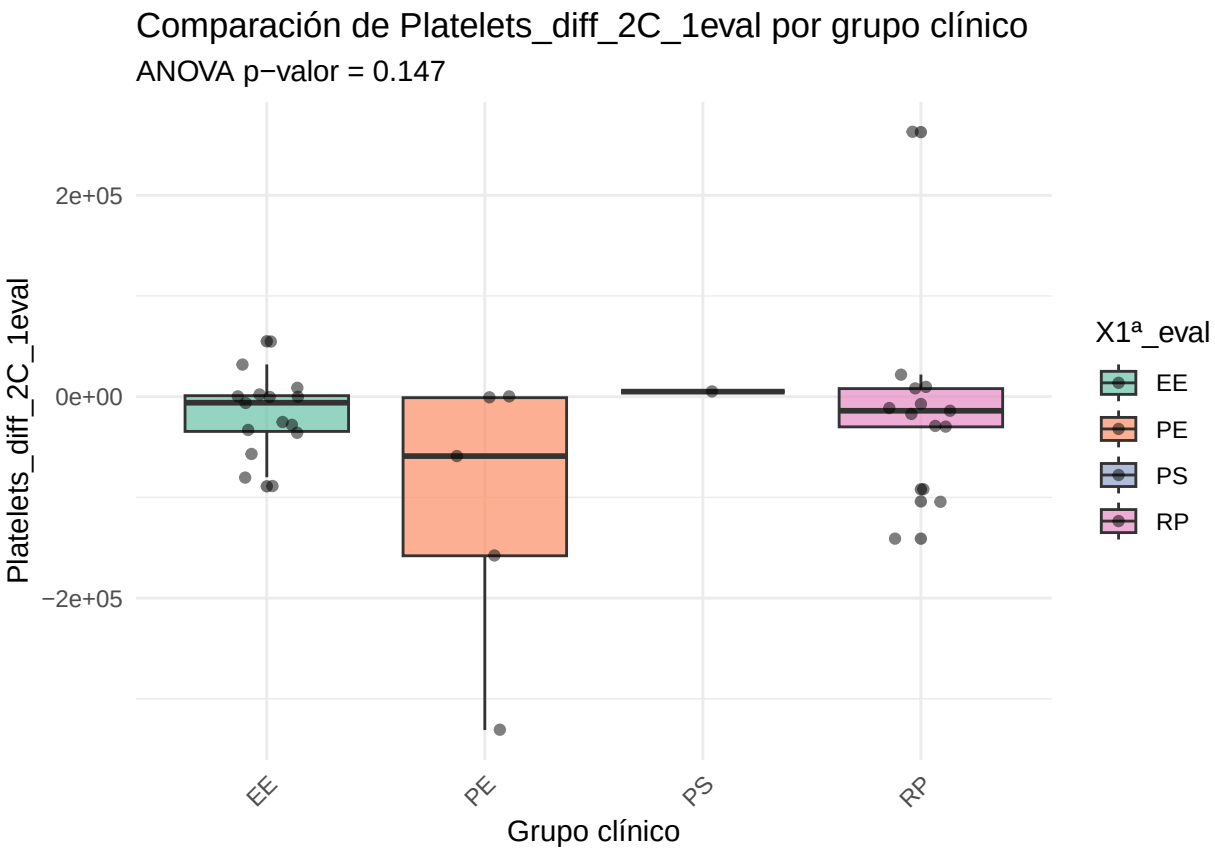

Comparación de PLR_media por grupo clínico
ANOVA p−valor = 0.00286
PLR_media
600
400
200
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

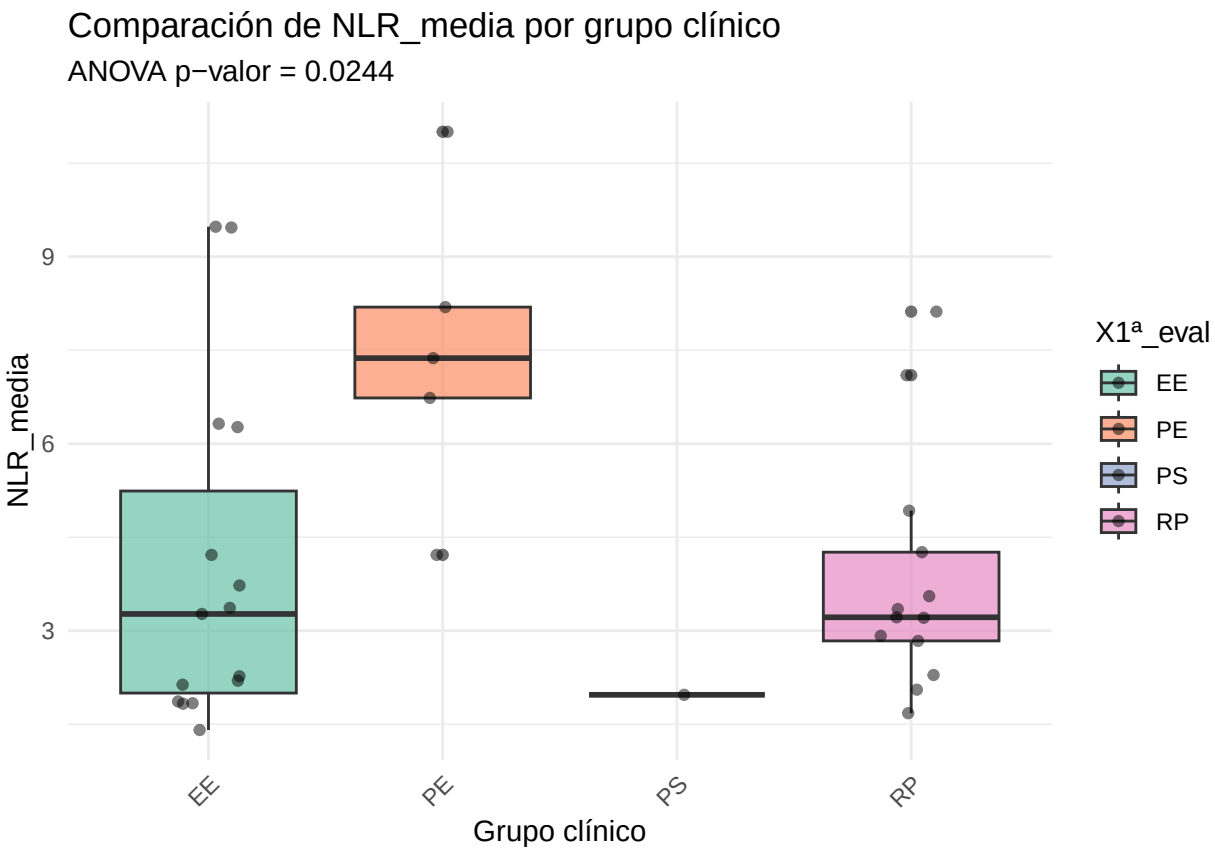

Comparación de PLR_slope por grupo clínico
ANOVA p−valor = 0.00173
PLR_slope
200
100
0
−100
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

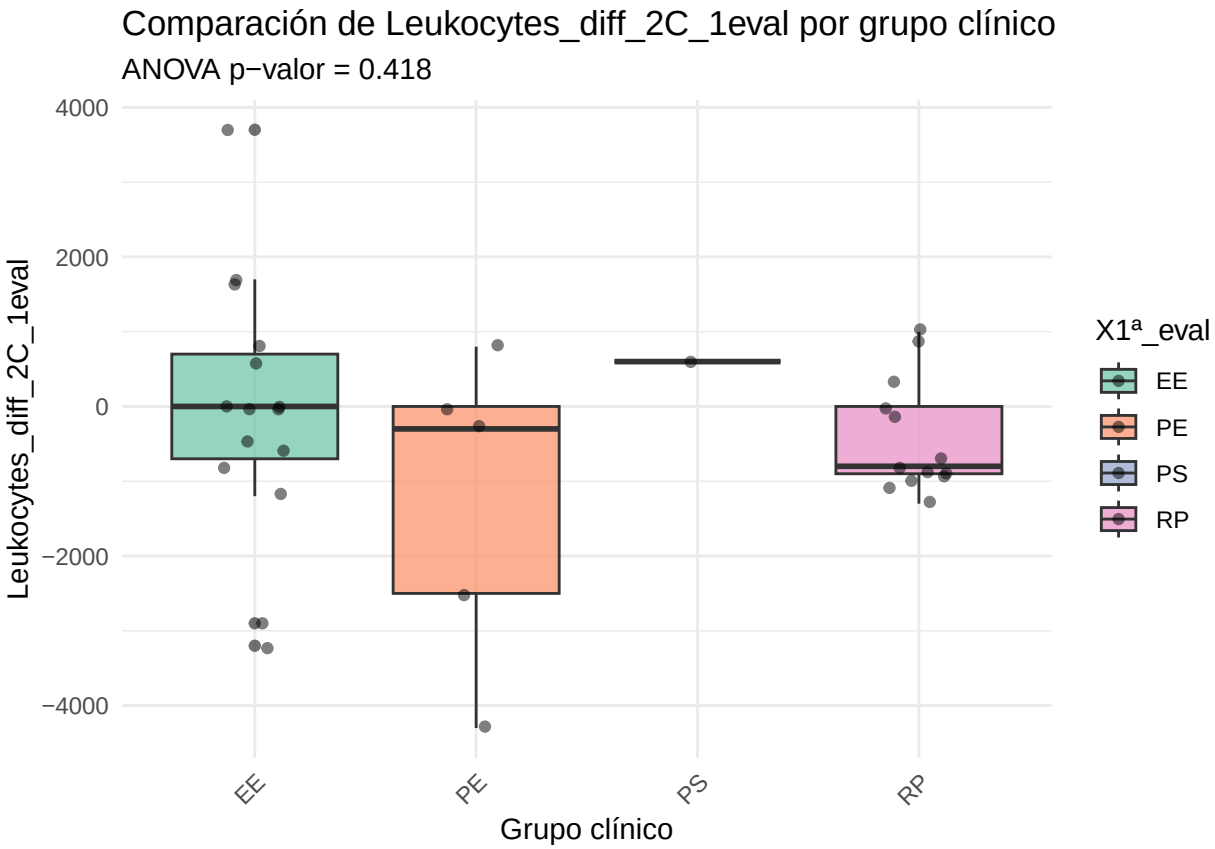

Comparación de PLR_diff_2C_1eval por grupo clínico
ANOVA p−valor = 0.856
PLR_diff_2C_1eval
300
200
100
0
−100
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

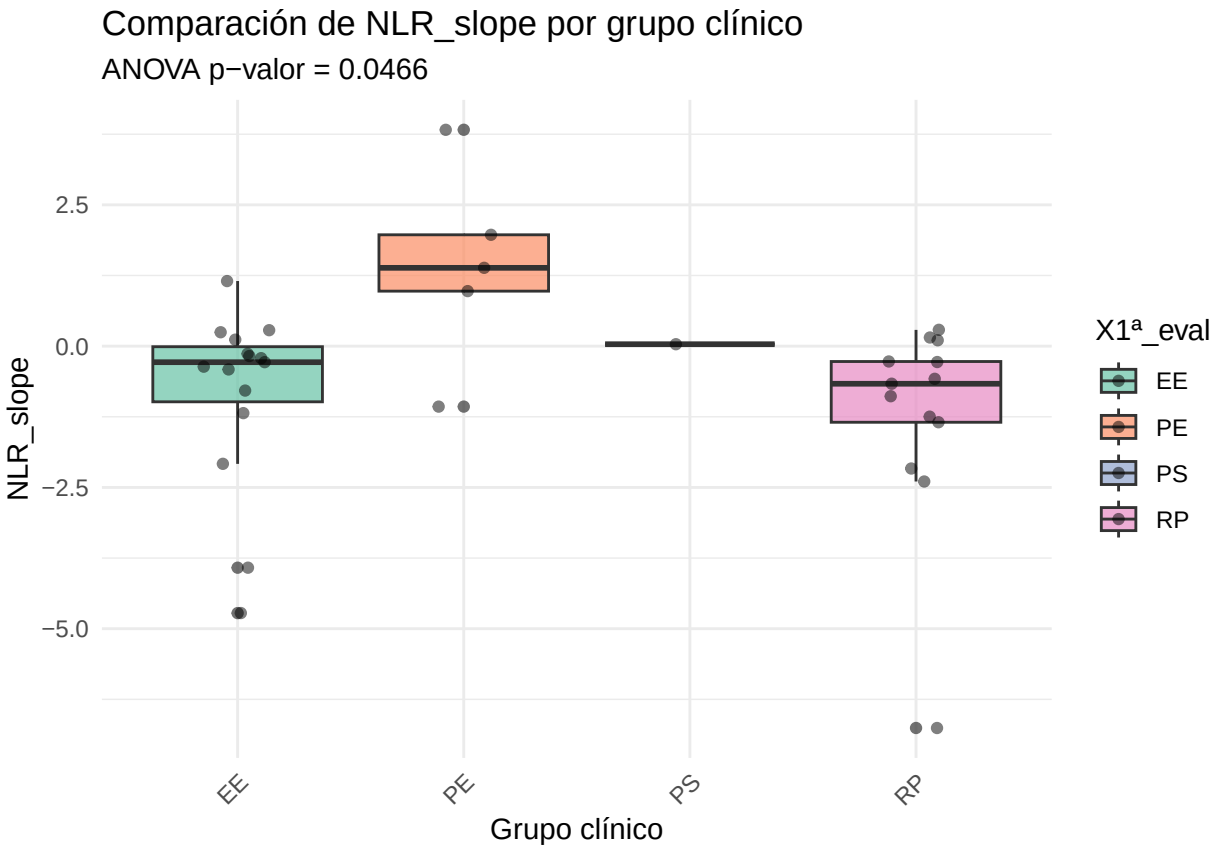

Comparación de PLR_diff_1C_2C por grupo clínico
ANOVA p−valor = 0.0644
PLR_diff_1C_2C
200
100
0
−100
−200
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

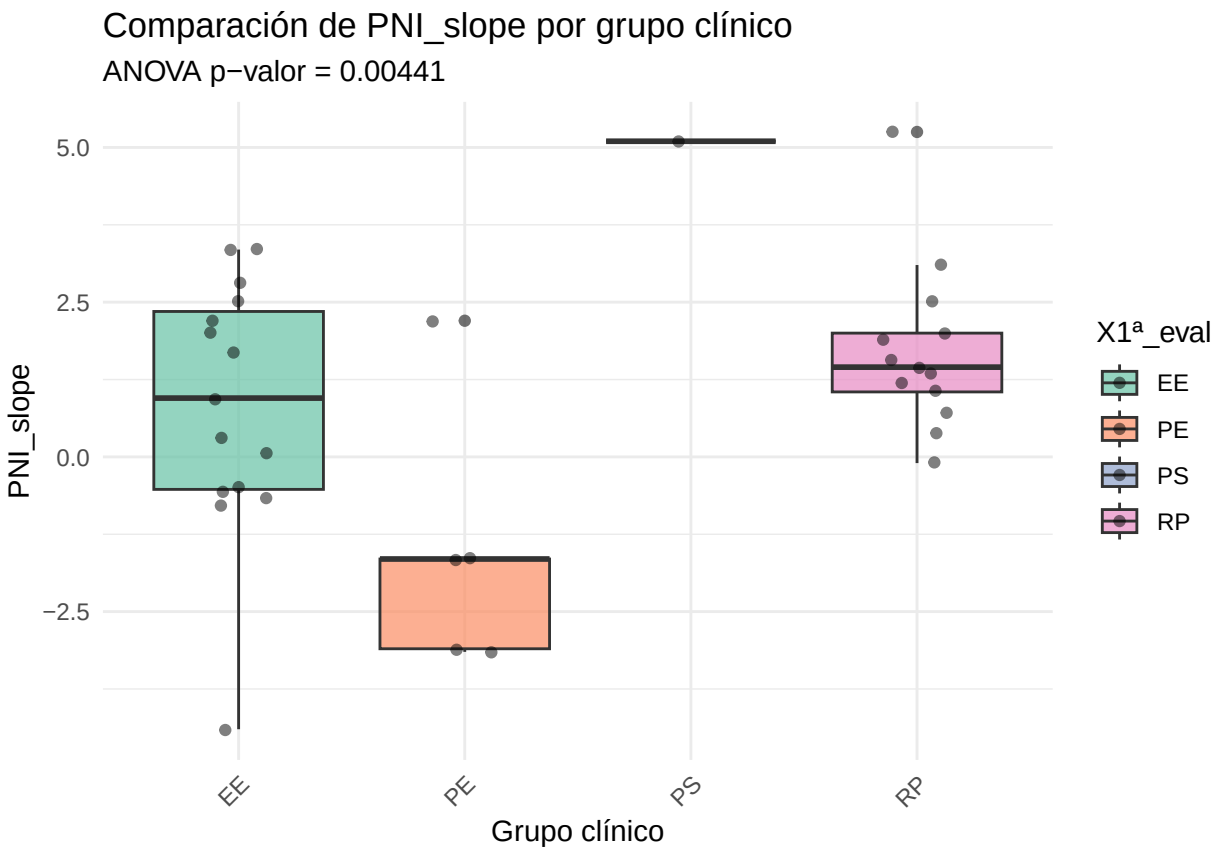

Comparación de Hemoglobin_cv por grupo clínico
ANOVA p−valor = 0.199
Hemoglobin_cv
0.20
0.15
0.10
0.05
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

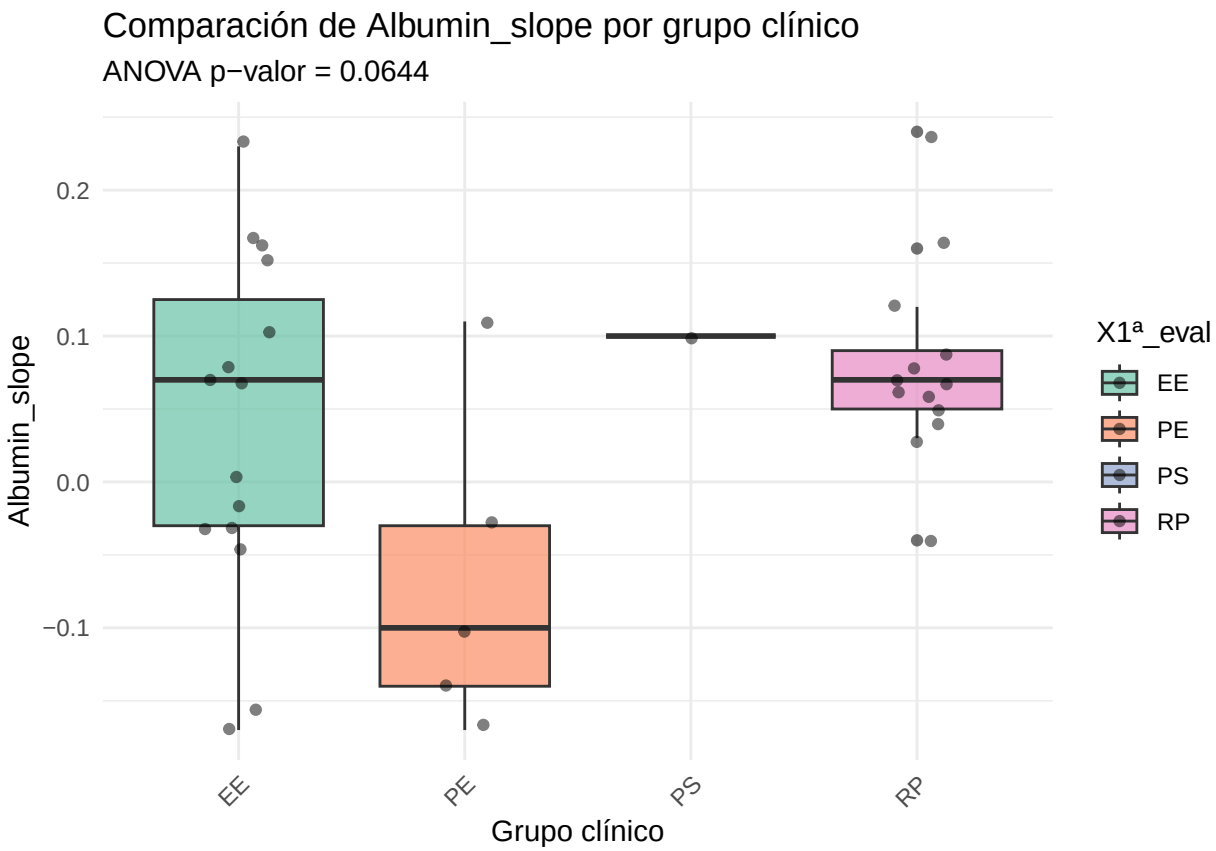

Comparación de Platelets_diff_2C_1eval por grupo clínico
ANOVA p−valor = 0.147
Platelets_diff_2C_1eval
2e+05
0e+00
−2e+05
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

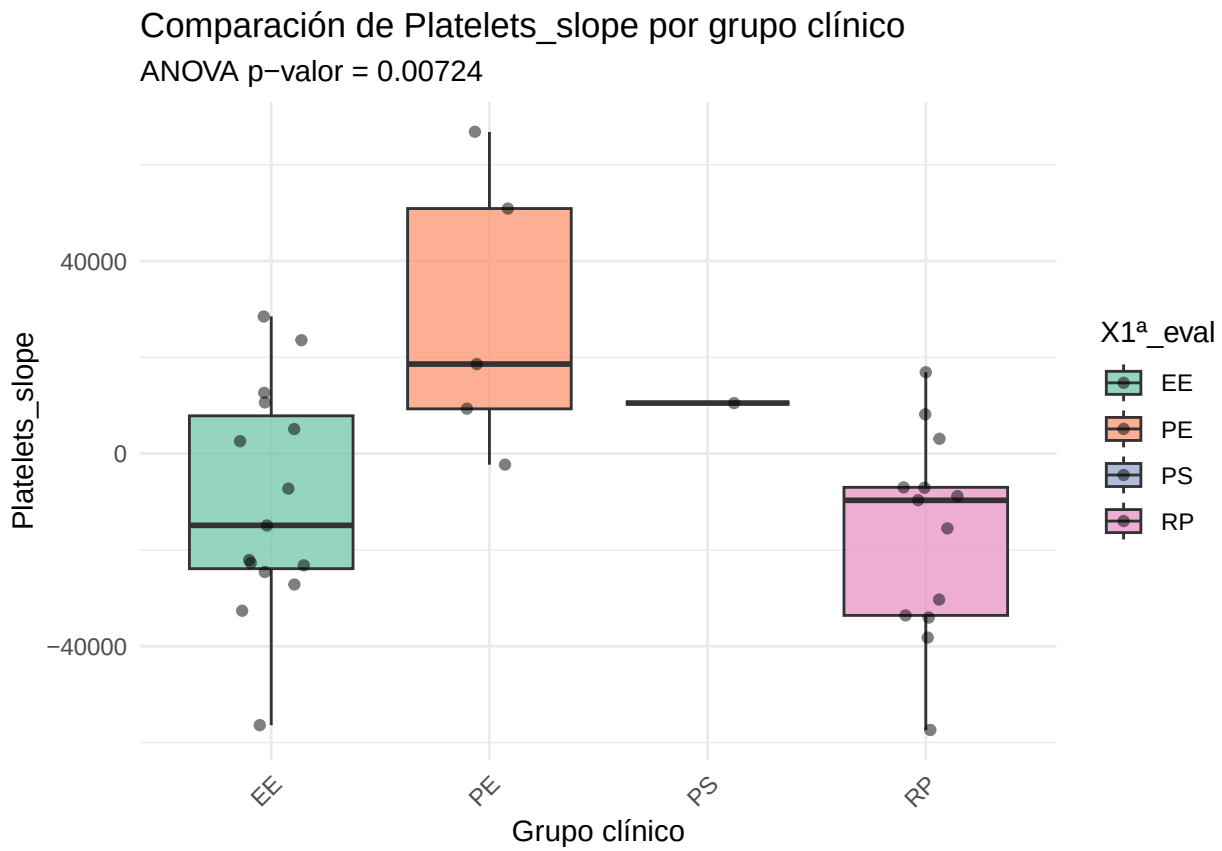

Comparación de NLR_media por grupo clínico
ANOVA p−valor = 0.0244
NLR_media
9
6
3
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

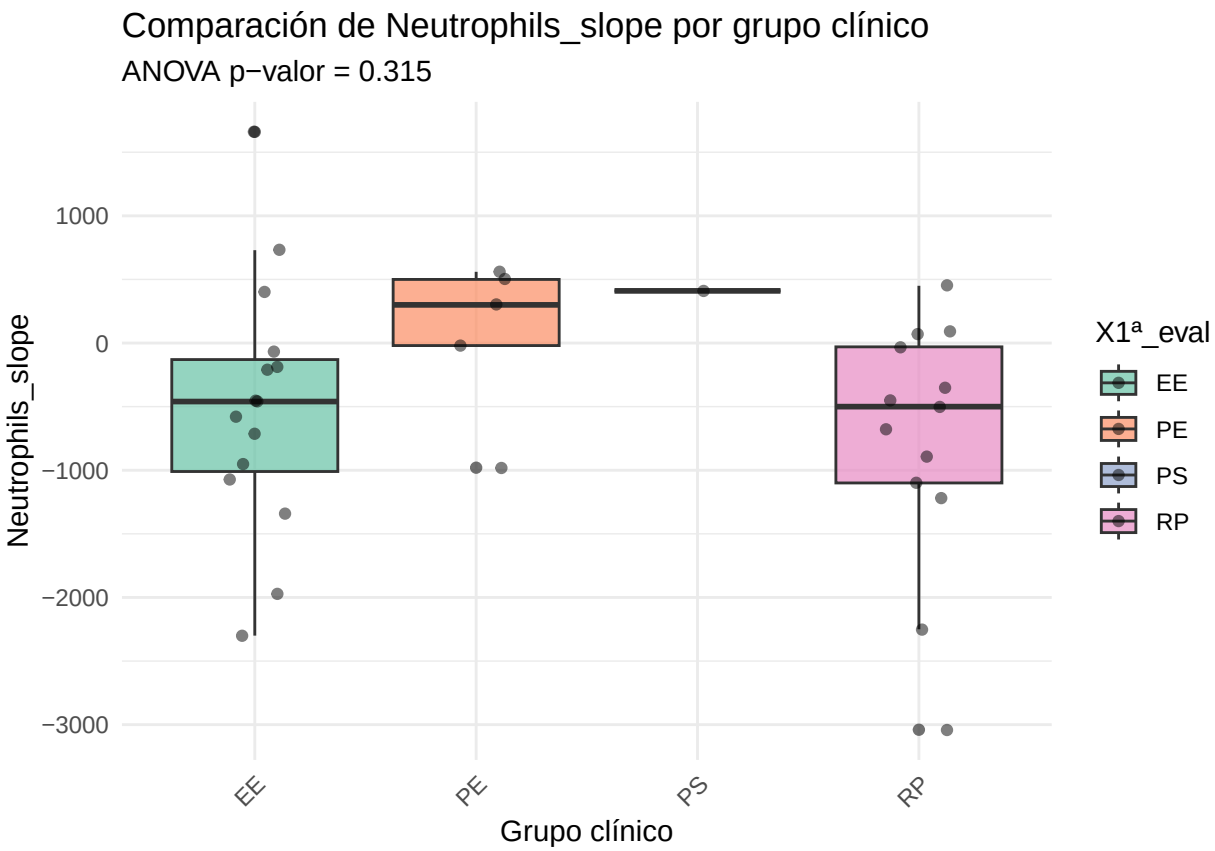

Comparación de Leukocytes_diff_2C_1eval por grupo clínico
ANOVA p−valor = 0.418
Leukocytes_diff_2C_1eval
4000
2000
0
−2000
−4000
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

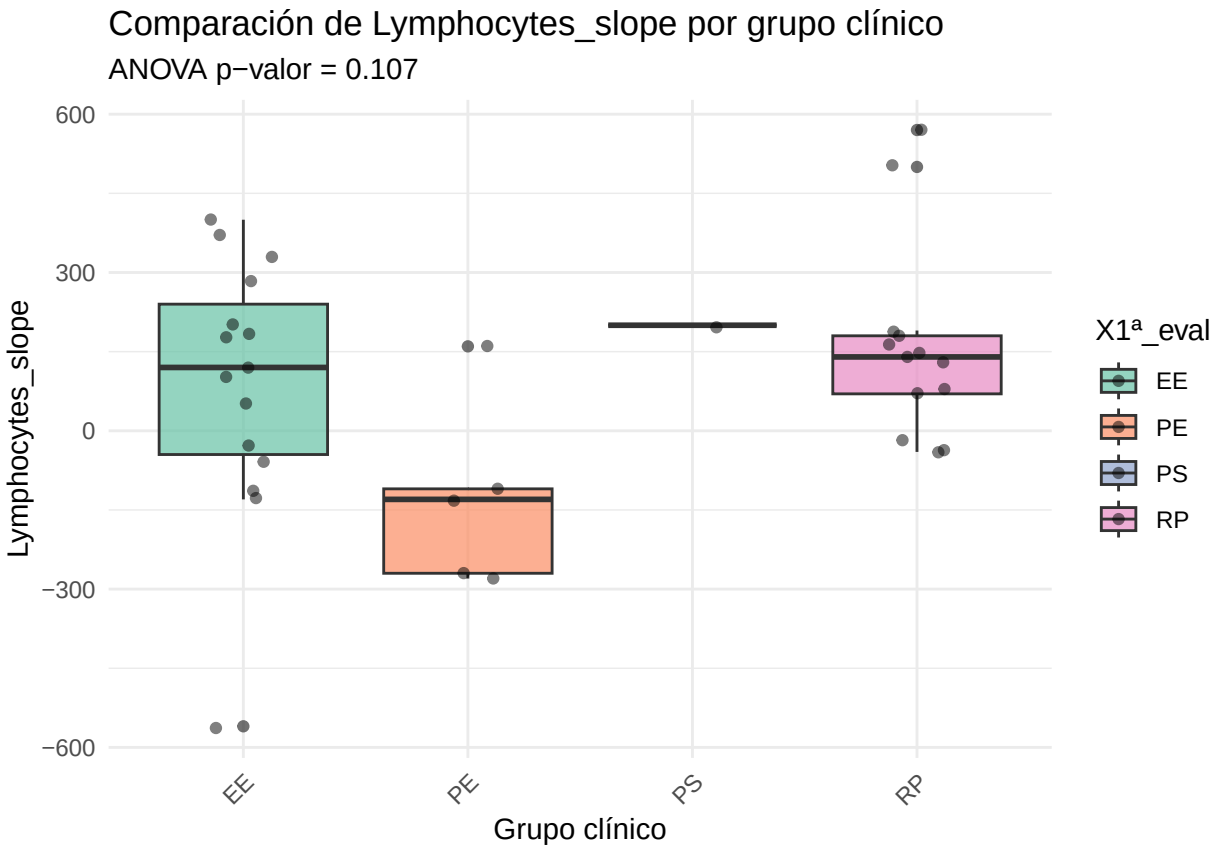

Comparación de NLR_slope por grupo clínico
ANOVA p−valor = 0.0466
NLR_slope
2.5
0.0
−2.5
−5.0
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP

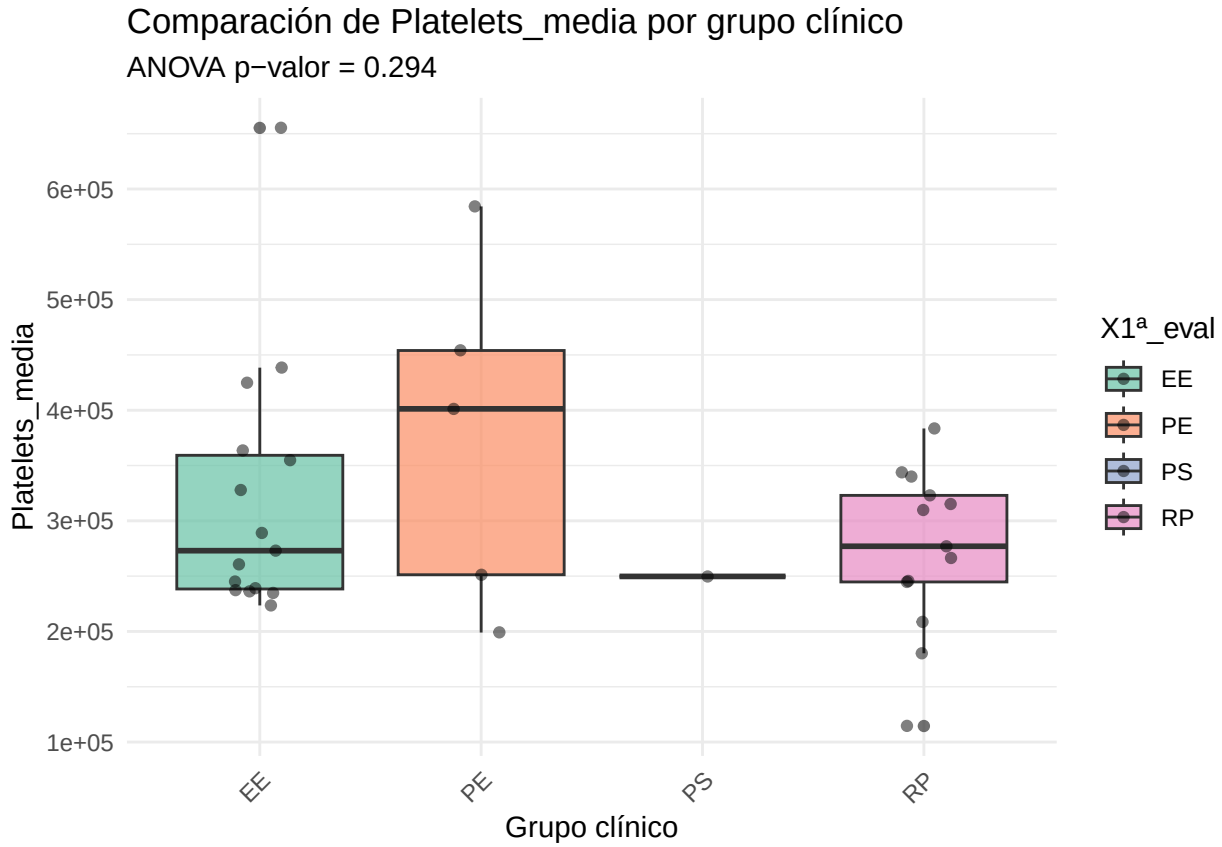

Comparación de PNI_slope por grupo clínico
ANOVA p−valor = 0.00441
PNI_slope
5.0
2.5
0.0
−2.5
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP


Comparación de Albumin_slope por grupo clínico
ANOVA p−valor = 0.0644
Albumin_slope
0.2
0.1
0.0
−0.1
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP


Comparación de Platelets_slope por grupo clínico
ANOVA p−valor = 0.00724
Platelets_slope
40000
0
−40000
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP


Comparación de Neutrophils_slope por grupo clínico
ANOVA p−valor = 0.315
Neutrophils_slope
1000
0
−1000
−2000
−3000
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP


Comparación de Lymphocytes_slope por grupo clínico
ANOVA p−valor = 0.107
Lymphocytes_slope
600
300
0
−300
−600
EE
PE
PS
RP
Grupo clínico
X1ª_eval
EE
PE
PS
RP




```
# Ordenar los resultados por p-valor ascendente
resultados_anova <- resultados_anova[order(resultados_anova$P_valor), ]
print(resultados_anova)
```

```
##                    Variable     P_valor
## 7                 PLR_slope 0.001729654
## 3           Hemoglobin_slope 0.002212932
## 6                 PLR_media 0.002860611
## 15                PNI_slope 0.004405763
## 17          Platelets_slope 0.007239990
## 1                 SII_media 0.011728533
## 5                 SII_slope 0.014021189
## 12                NLR_media 0.024372500
## 14                NLR_slope 0.046601965
## 16            Albumin_slope 0.064391448
## 9             PLR_diff_1C_2C 0.064447319
## 19        Lymphocytes_slope 0.106613191
## 11  Platelets_diff_2C_1eval 0.147045346
## 10            Hemoglobin_cv 0.199323473
## 20          Platelets_media 0.294211704
## 18        Neutrophils_slope 0.314587683
## 13 Leukocytes_diff_2C_1eval 0.417783605
## 2         NLR_diff_2C_1eval 0.436099239
## 4            SII_diff_1C_2C 0.668567085
## 8         PLR_diff_2C_1eval 0.856144402
```

Los resultados del ANOVA aplicados a las pendientes de evolución de biomarcadores (*_slope) muestran diferencias estadísticamente significativas entre los grupos clínicos en varias variables clave.

En particular, SII_slope, PLR_slope, NLR_slope y Hemoglobin_slope presentan p-valores < 0.05, indicando que su evolución en el tiempo varía significativamente entre los grupos. En todos estos casos, el grupo PE (progresión evidente) muestra valores claramente positivos, reflejando un aumento progresivo de los índices inflamatorios y una caída de la hemoglobina, lo que sugiere un perfil clínico desfavorable.

En contraste, los grupos RP y EE tienden a presentar pendientes negativas o estables, reflejando mejor evolución. La variable Lymphocytes_slope, aunque muestra diferencias visuales coherentes con ese patrón (descenso en PE y aumento en RP/EE), no alcanza significación estadística ($p = 0.107$), posiblemente debido a mayor variabilidad o menor tamaño muestral.

Estos resultados refuerzan la utilidad de los biomarcadores dinámicos como predictores del curso clínico en pacientes tratados con inmunoterapia.

```
library(writexl)
```

```
## Warning: package 'writexl' was built under R version 4.3.3
```

```
# Exportar el dataframe a un archivo Excel
write_xlsx(datos_dummy_df, path = "datos_PCA.xlsx")
```