

Análisis Estadístico Computacional

Intervalos de Confianza

Renzo Robiño Tito Chura
Estadística Computacional

11 de noviembre de 2025

Ejercicio 1: Concentración de PM2.5 en Lima

Código R

```
1 library(ggplot2)
2
3 # Datos proporcionados
4 n <- 50
5 media <- 45.3
6 sd <- 8.7
7 z <- 1.96
8 limite_oms <- 15
9
10 # a) Intervalo de confianza al 95% - MOSTRAR RESULTADO
11 error_estandar <- sd / sqrt(n)
12 margen_error <- z * error_estandar
13 ic_inf <- media - margen_error
14 ic_sup <- media + margen_error
15
16 print("a) Intervalo de Confianza 95% para la concentraci n promedio:")
17 print(paste("L mite inferior:", round(ic_inf, 2), " g /m "))
18 print(paste("L mite superior:", round(ic_sup, 2), " g /m "))
19
20 # b) Gr fico de densidad con IC
21 set.seed(123)
22 datos_densidad <- data.frame(concentracion = rnorm(1000, media, sd))
23
24 print("b) Gr fico de densidad con IC:")
25 ggplot(datos_densidad, aes(concentracion)) +
26   geom_density(fill = "skyblue", alpha = 0.7) +
27   geom_vline(xintercept = c(ic_inf, ic_sup), linetype = "dashed", color
28             = "red", size = 1) +
29   geom_vline(xintercept = media, color = "blue", size = 1) +
30   labs(title = "Distribuci n de concentraciones PM2.5 con IC 95%",
31         x = "Concentraci n ( g /m )", y = "Densidad")
32
33 # c) Mapa de calor por distritos
34 distritos <- c("San Juan de Lurigancho", "Ate", "Villa El Salvador", "
35               Los Olivos", "San Mart n de Porres")
36 concentraciones <- c(52.1, 48.3, 41.2, 46.8, 43.9)
```

```

35
36 datos_mapa <- data.frame(
37   distrito = factor(distritos, levels = distritos),
38   concentracion = concentraciones
39 )
40
41 print("c) Mapa de calor por distritos:")
42 ggplot(datos_mapa, aes(x = distrito, y = 1, fill = concentracion)) +
43   geom_tile(color = "white") +
44   geom_text(aes(label = paste0(concentracion, " g / m ")), color =
45             "black", size = 3.5) +
46   scale_fill_gradient2(low = "green", mid = "yellow", high = "red",
47                         midpoint = 45) +
48   theme(axis.text.x = element_text(angle = 45, hjust = 1),
49         axis.title.y = element_blank(),
50         axis.text.y = element_blank()) +
51   labs(title = "Concentraciones de PM2.5 por distrito",
52        x = "Distrito")
53
54 # d) Comparacion con limite OMS usando barras horizontales
55 datos_comparacion <- data.frame(
56   categoria = c("Lima Metropolitana", "Limite OMS"),
57   valor = c(media, limite_oms),
58   inf = c(ic_inf, NA),
59   sup = c(ic_sup, NA)
60 )
61
62 print("d) Comparacion con limite OMS:")
63 ggplot(datos_comparacion, aes(x = valor, y = categoria)) +
64   geom_point(size = 3) +
65   geom_errorbarh(aes(xmin = inf, xmax = sup), height = 0.1, size = 1,
66                  data = subset(datos_comparacion, categoria == "Lima
67                  Metropolitana")) +
68   geom_vline(xintercept = limite_oms, color = "red", linetype = "dashed"
69               , size = 1) +
70   labs(title = "Comparacion con limite OMS de PM2.5",
71        x = "Concentracion ( g / m )", y = "")
72
73 # e) Serie temporal con bandas de confianza
74 set.seed(123)
75 dias <- 1:30
76 concentraciones_diarias <- cumsum(rnorm(30, 0, 2)) + media
77 ic_banda <- 1.96 * sd / sqrt(length(dias))
78
79 datos_serie <- data.frame(
80   dia = dias,
81   concentracion = concentraciones_diarias
82 )
83
84 print("e) Serie temporal con bandas de confianza:")
85 ggplot(datos_serie, aes(x = dia, y = concentracion)) +
86   geom_line(color = "blue", size = 1) +
87   geom_ribbon(aes(ymin = concentracion - ic_banda,
88                   ymax = concentracion + ic_banda),
89               alpha = 0.3, fill = "lightblue") +
90   geom_hline(yintercept = limite_oms, color = "red", linetype = "dashed"
91               , size = 1) +
92   labs(title = "Evolucion diaria de PM2.5 con banda de confianza",
93        x = "Dia", y = "Concentracion ( g / m )")

```

```
x = "D_a", y = "Concentracion ( g / m )")
```

Resultados Ejercicio 1

a) Intervalo de Confianza 95% para la concentración promedio:

- [1] "Límite inferior: 42.89 g/m³"
- [1] "Límite superior: 47.71 g/m³"

Gráficos Ejercicio 1

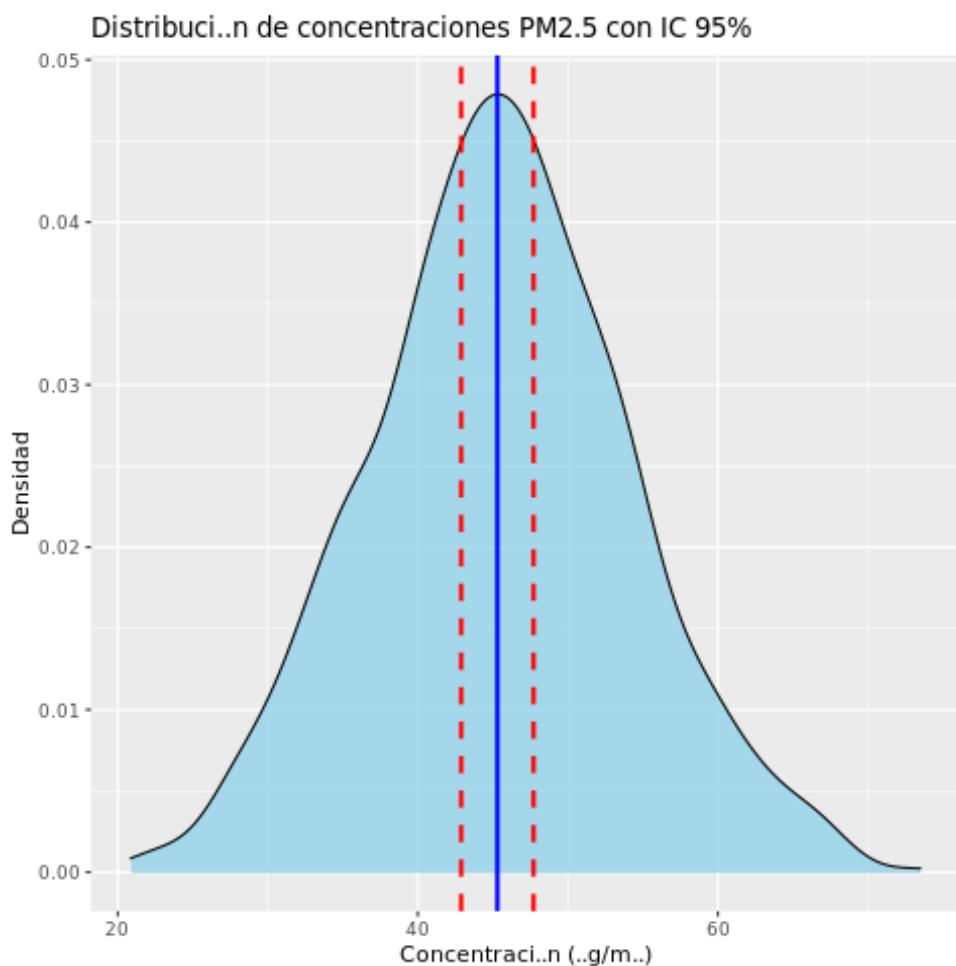


Figura 1: Distribución de concentraciones PM2.5 con IC 95 %

Concentraciones de PM2.5 por distrito

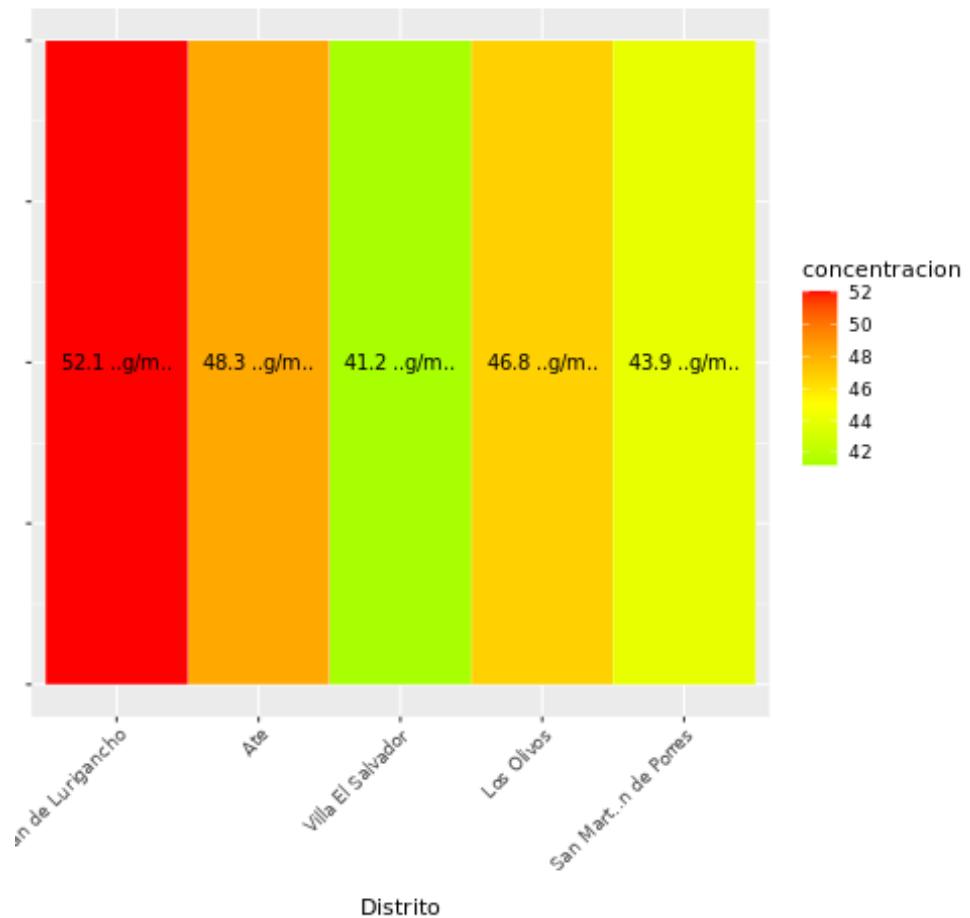


Figura 2: Concentraciones de PM2.5 por distrito

Comparaci..n con l..mite OMS de PM2.5

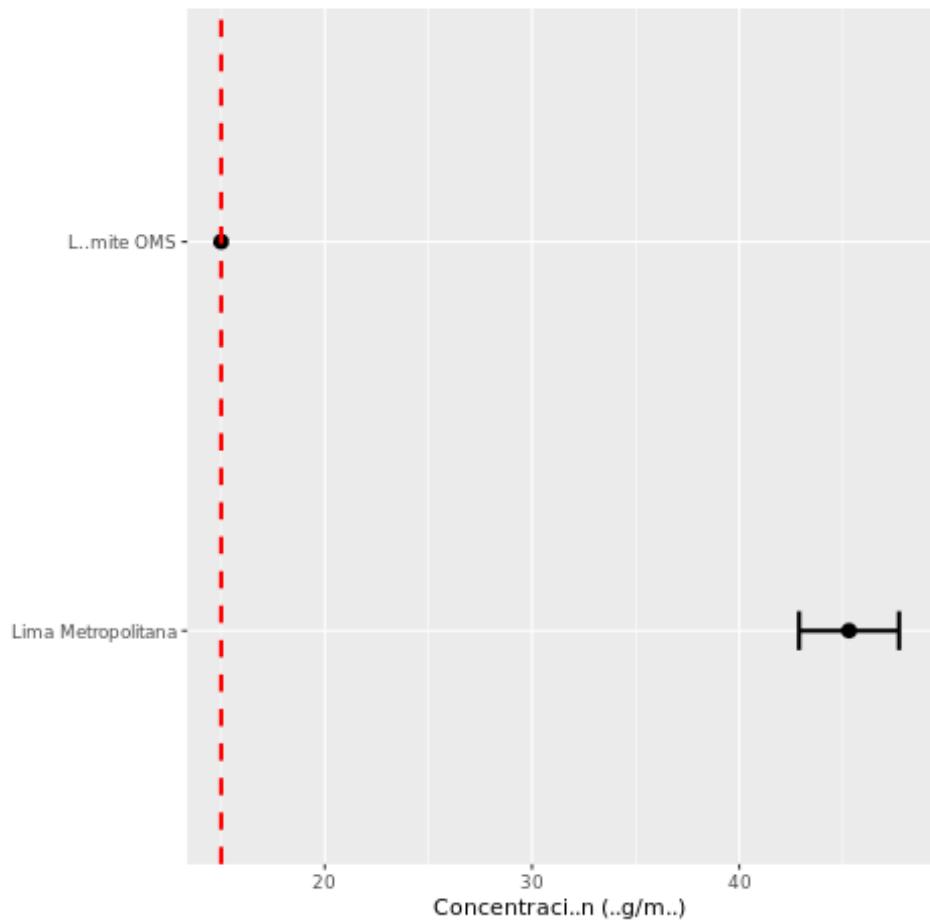


Figura 3: Comparación con límite OMS de PM2.5

Evoluci..n diaria de PM2.5 con banda de confianza

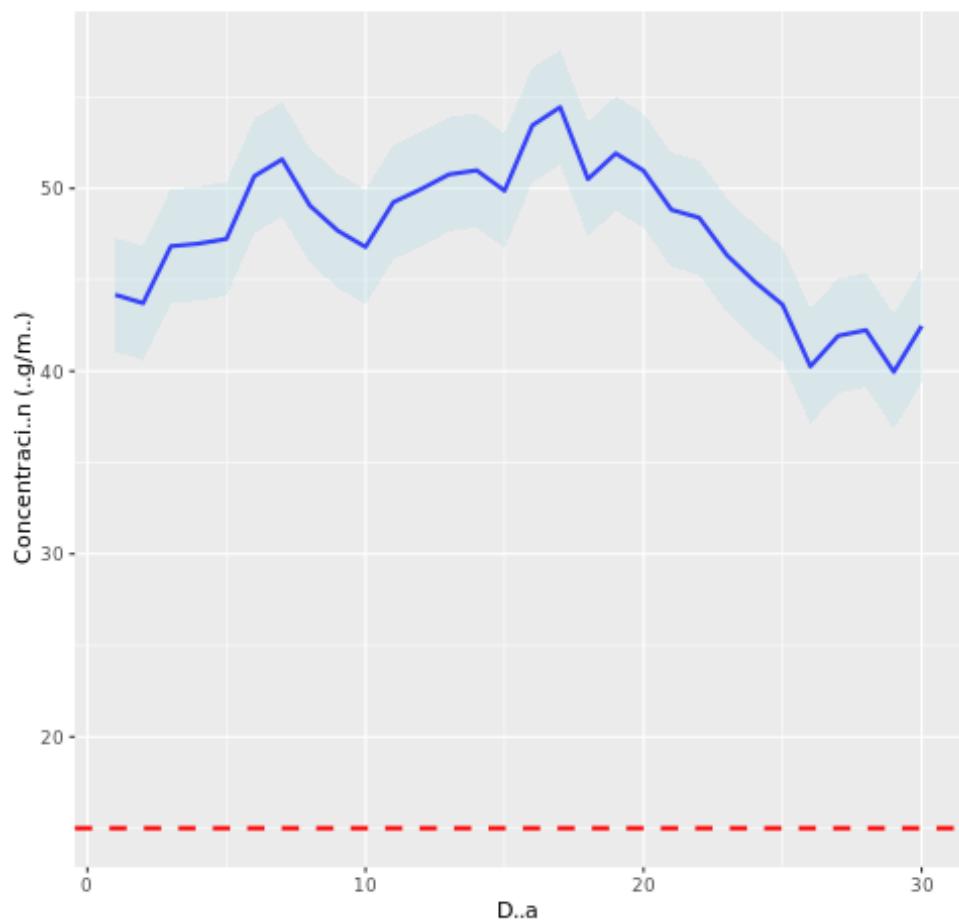


Figura 4: Evolución diaria de PM2.5 con banda de confianza

Ejercicio 2: Machine Learning para Precio de Quinua

Código R

```
1 library(ggplot2)
2
3 # Datos proporcionados
4 precio_media <- 8.50
5 precio_sd <- 1.20
6 error_estandar <- 0.35
7 n_meses <- 36
8 precio_minimo <- 7.00
9
10 # a) Intervalos de predicci n - MOSTRAR RESULTADOS
11 z_valores <- c(1.645, 1.96, 2.576)
12 niveles <- c("90%", "95%", "99%")
13
14 ics_prediccion <- data.frame(
15   Nivel = niveles,
16   Inferior = precio_media - z_valores * error_estandar,
17   Superior = precio_media + z_valores * error_estandar
18 )
19
20 # MOSTRAR intervalos en consola
21 print("a) Intervalos de predicci n para el pr ximo mes:")
22 print(ics_prediccion)
23
24 # b) Serie temporal hist rica
25 set.seed(123)
26 fechas <- seq.Date(from = as.Date("2021-01-01"), by = "month", length =
27   out = n_meses)
28 precios_historicos <- cumsum(rnorm(n_meses, 0, 0.2)) + precio_media
29
30 datos_serie <- data.frame(
31   Fecha = fechas,
32   Precio = precios_historicos
33 )
34
35 print("b) Gr fico de serie temporal:")
36 ggplot(datos_serie, aes(x = Fecha, y = Precio)) +
37   geom_line(color = "blue", size = 1) +
38   geom_point(size = 1) +
39   labs(title = "Serie temporal hist rica de precios de quinua",
40         x = "Fecha", y = "Precio (S/ por kg)")
41
42 # c) Bandas de confianza
43 datos_bandas <- datos_serie
44 datos_bandas$inf_90 <- datos_bandas$Precio - 1.645 * error_estandar
45 datos_bandas$sup_90 <- datos_bandas$Precio + 1.645 * error_estandar
46 datos_bandas$inf_95 <- datos_bandas$Precio - 1.96 * error_estandar
47 datos_bandas$sup_95 <- datos_bandas$Precio + 1.96 * error_estandar
48 datos_bandas$inf_99 <- datos_bandas$Precio - 2.576 * error_estandar
49 datos_bandas$sup_99 <- datos_bandas$Precio + 2.576 * error_estandar
50
51 print("c) Gr fico con bandas de confianza:")
52 ggplot(datos_bandas, aes(x = Fecha, y = Precio)) +
53   geom_line(color = "black", size = 1) +
```

```

53 geom_ribbon(aes(ymin = inf_90, ymax = sup_90), fill = "green", alpha =
54   0.3) +
55 geom_ribbon(aes(ymin = inf_95, ymax = sup_95), fill = "blue", alpha =
56   0.3) +
57 geom_ribbon(aes(ymin = inf_99, ymax = sup_99), fill = "red", alpha =
58   0.3) +
59 labs(title = "Precios históricos con bandas de confianza",
60       x = "Fecha", y = "Precio (S/ por kg)")
61
62 # d) Boxplot por trimestre
63 datos_serie$Trimestre <- factor(rep(1:12, each = 3)[1:n_meses])
64
65 print("d) Boxplot por trimestre:")
66 ggplot(datos_serie, aes(x = Trimestre, y = Precio)) +
67   geom_boxplot(fill = "lightblue", alpha = 0.7) +
68   geom_hline(yintercept = precio_minimo, color = "red", linetype =
69     "dashed", size = 1) +
70   labs(title = "Distribución de precios por trimestre",
71         x = "Trimestre", y = "Precio (S/ por kg)")
72
73 # f) Gráfico de violín para residuales
74 set.seed(123)
75 residuales <- rnorm(100, 0, error_estandar)
76 datos_residuales <- data.frame(Residual = residuales)
77
78 print("f) Gráfico de violín para residuales:")
79 ggplot(datos_residuales, aes(x = "", y = Residual)) +
80   geom_violin(fill = "lightgreen", alpha = 0.7) +
81   geom_boxplot(width = 0.1, fill = "white", alpha = 0.7) +
82   labs(title = "Distribución de residuales del modelo",
83         x = "", y = "Residual")

```

Resultados Ejercicio 2

a) Intervalos de predicción para el próximo mes:

Nivel Inferior Superior

1	90%	7.924	9.076
2	95%	7.814	9.186
3	99%	7.598	9.402

Gráficos Ejercicio 2

Serie temporal hist..rica de precios de quinua

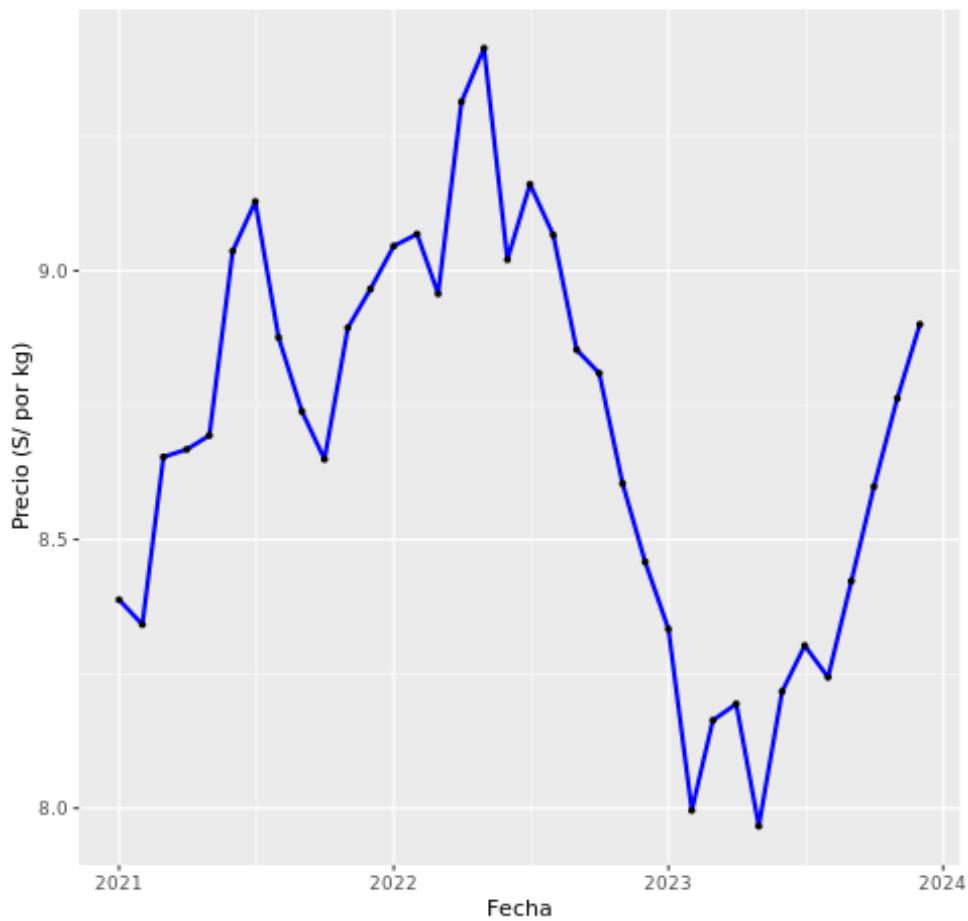


Figura 5: Serie temporal histórica de precios de quinua

Precios hist..ricos con bandas de confianza

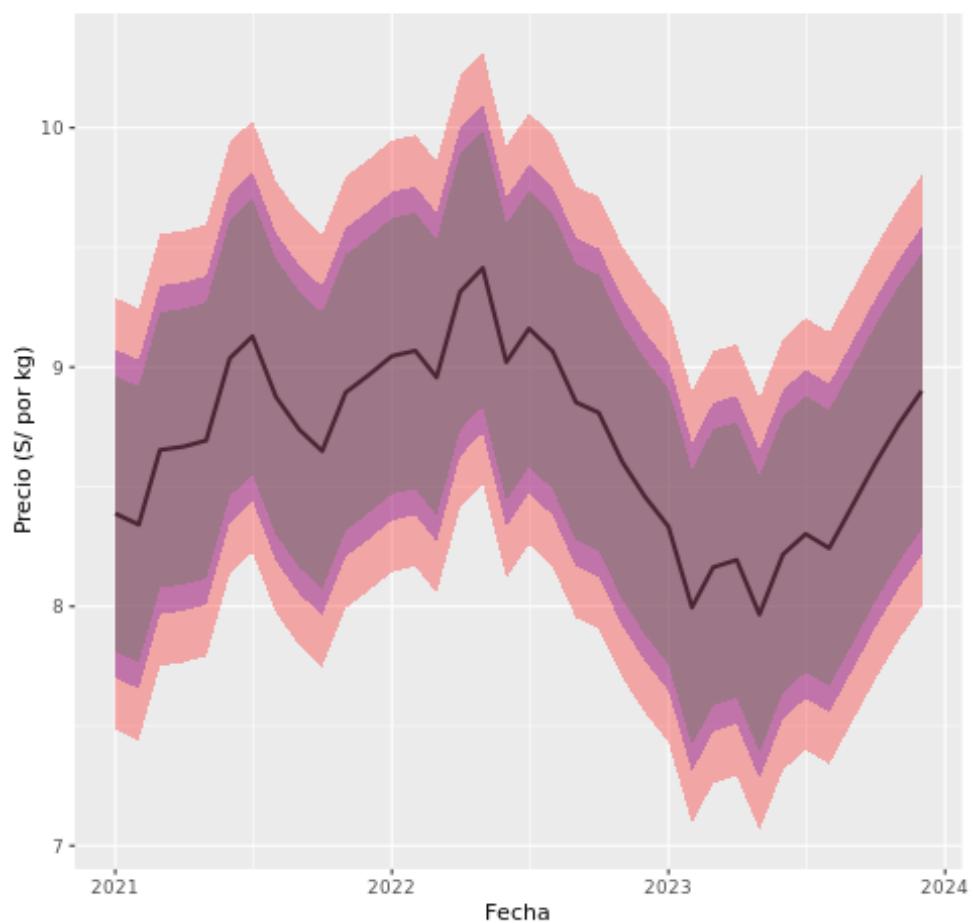


Figura 6: Precios históricos con bandas de confianza

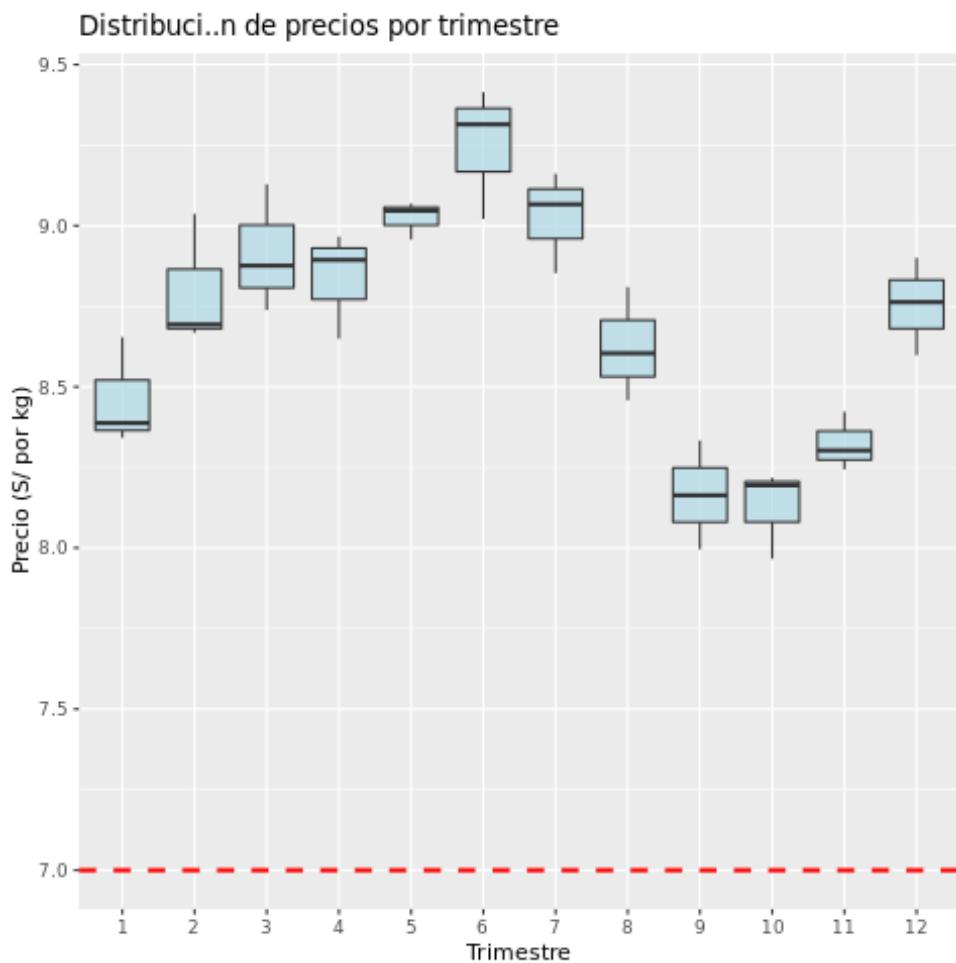


Figura 7: Distribución de precios por trimestre

Distribuci..n de residuales del modelo

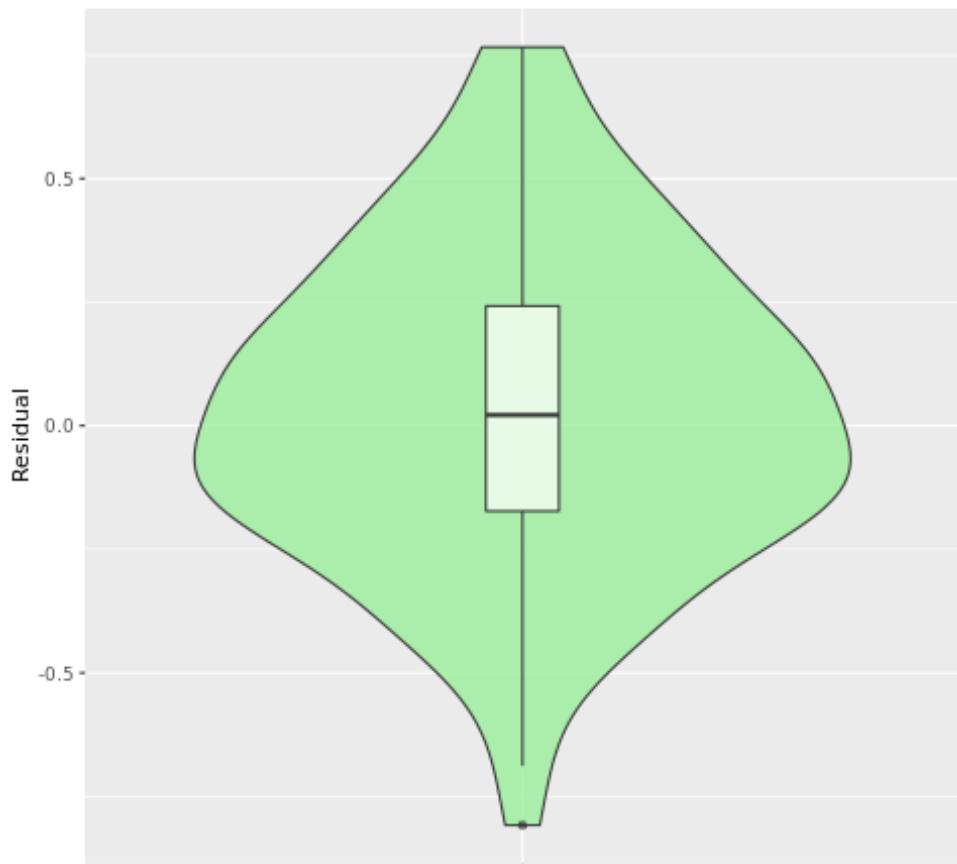


Figura 8: Distribución de residuales del modelo

Ejercicio 3: Efectividad de Campañas de Vacunación

Código R

```
1 library(ggplot2)
2
3 # Datos proporcionados
4 estrategia_A <- list(media = 68.5, sd = 12.3, n = 13)
5 estrategia_B <- list(media = 76.2, sd = 10.8, n = 12)
6
7 # a) IC para cada estrategia - MOSTRAR RESULTADOS
8 calcular_ic_t <- function(media, sd, n, confianza = 0.95) {
9   alpha <- 1 - confianza
10  t_val <- qt(1 - alpha/2, df = n - 1)
11  error <- t_val * sd / sqrt(n)
12  return(c(media - error, media + error))
13}
14
15 ic_A <- calcular_ic_t(estrategia_A$media, estrategia_A$sd, estrategia_A$n)
16 ic_B <- calcular_ic_t(estrategia_B$media, estrategia_B$sd, estrategia_B$n)
17
18 print("a) Intervalos de Confianza 95% para cada estrategia:")
19 print(paste("Estrategia A:", round(ic_A[1], 2), "-", round(ic_A[2], 2),
20           "%"))
21 print(paste("Estrategia B:", round(ic_B[1], 2), "-", round(ic_B[2], 2),
22           "%"))
23
24 # b) Forest plot
25 datos_forest <- data.frame(
26   Estrategia = c("A (Tradicional)", "B (Brigadas m viles)"),
27   Media = c(estrategia_A$media, estrategia_B$media),
28   Inferior = c(ic_A[1], ic_B[1]),
29   Superior = c(ic_A[2], ic_B[2])
30 )
31
32 print("b) Forest plot comparando estrategias:")
33 ggplot(datos_forest, aes(x = Media, y = Estrategia)) +
34   geom_point(size = 3, color = "blue") +
35   geom_errorbarh(aes(xmin = Inferior, xmax = Superior),
36                  height = 0.1, color = "blue", size = 1) +
37   labs(title = "Comparaci n de cobertura de vacunaci n por estrategia",
38        ,
39        x = "Cobertura (%)", y = "Estrategia")
40
41 # c) Gr fico de barras por regi n
42 regiones <- c("Costa", "Sierra", "Selva")
43 cobertura_A <- c(72.1, 65.8, 67.5)
44 cobertura_B <- c(78.3, 74.2, 76.1)
45 sd_A <- c(8.2, 10.5, 9.8)
46 sd_B <- c(7.1, 9.3, 8.6)
47
48 datos_regiones <- data.frame(
49   Region = rep(regiones, 2),
50   Estrategia = rep(c("A", "B"), each = 3),
51   Cobertura = c(cobertura_A, cobertura_B),
52   SD = c(sd_A, sd_B)
```

```

50 )
51
52 print("c) Gráfico de barras por región:")
53 ggplot(datos_regiones, aes(x = Region, y = Cobertura, fill = Estrategia)
54   ) +
55   geom_bar(stat = "identity", position = position_dodge(0.8), alpha =
56   0.7) +
57   geom_errorbar(aes(ymin = Cobertura - SD, ymax = Cobertura + SD),
58   position = position_dodge(0.8), width = 0.2) +
59   scale_fill_manual(values = c("A" = "red", "B" = "blue")) +
60   labs(title = "Cobertura de vacunación por región y estrategia",
61   x = "Región", y = "Cobertura (%)")
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84

```

Resultados Ejercicio 3

a) Intervalos de Confianza 95% para cada estrategia:

[1] "Estrategia A: 61.16 - 75.84 %"

[1] "Estrategia B: 69.56 - 82.84 %"

d) Intervalo de Confianza para la diferencia:

[1] "Diferencia (B - A): 7.7 %"

[1] "IC 95% diferencia: -1.89 - 17.29 %"

Gráficos Ejercicio 3

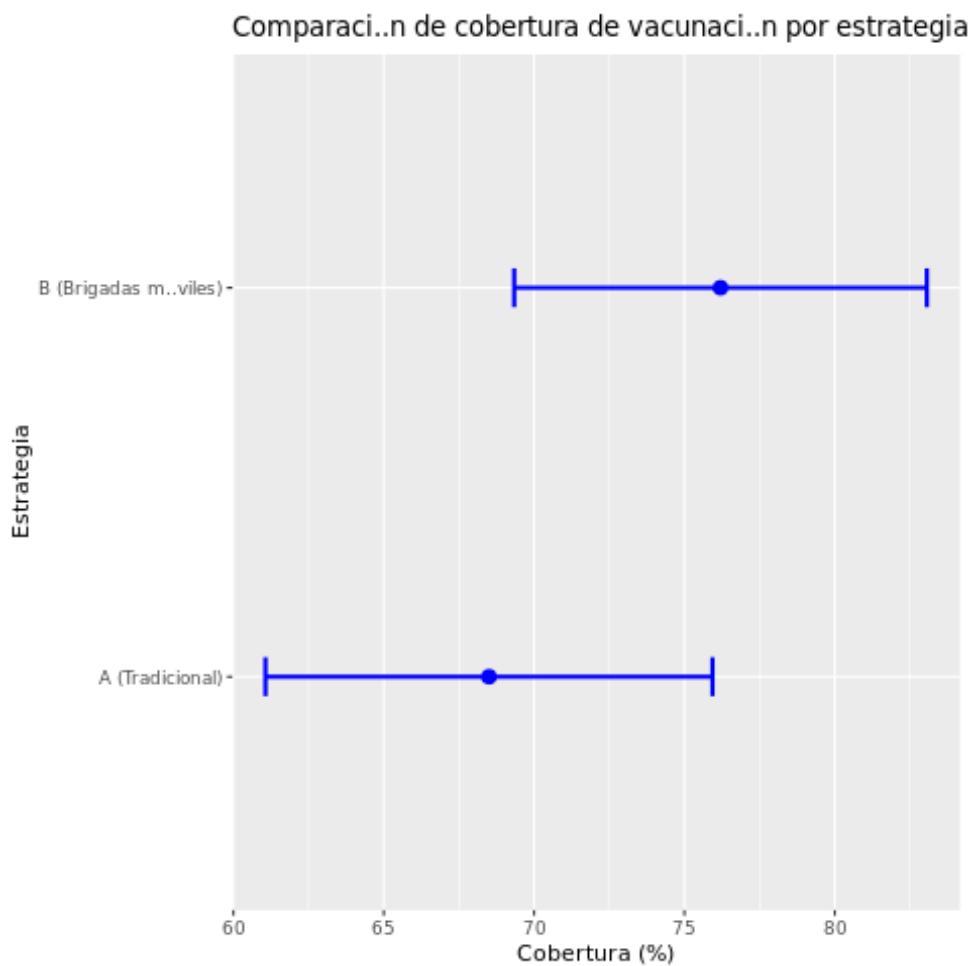


Figura 9: Comparación de cobertura de vacunación por estrategia

Cobertura de vacunaci..n por regi..n y estrategia

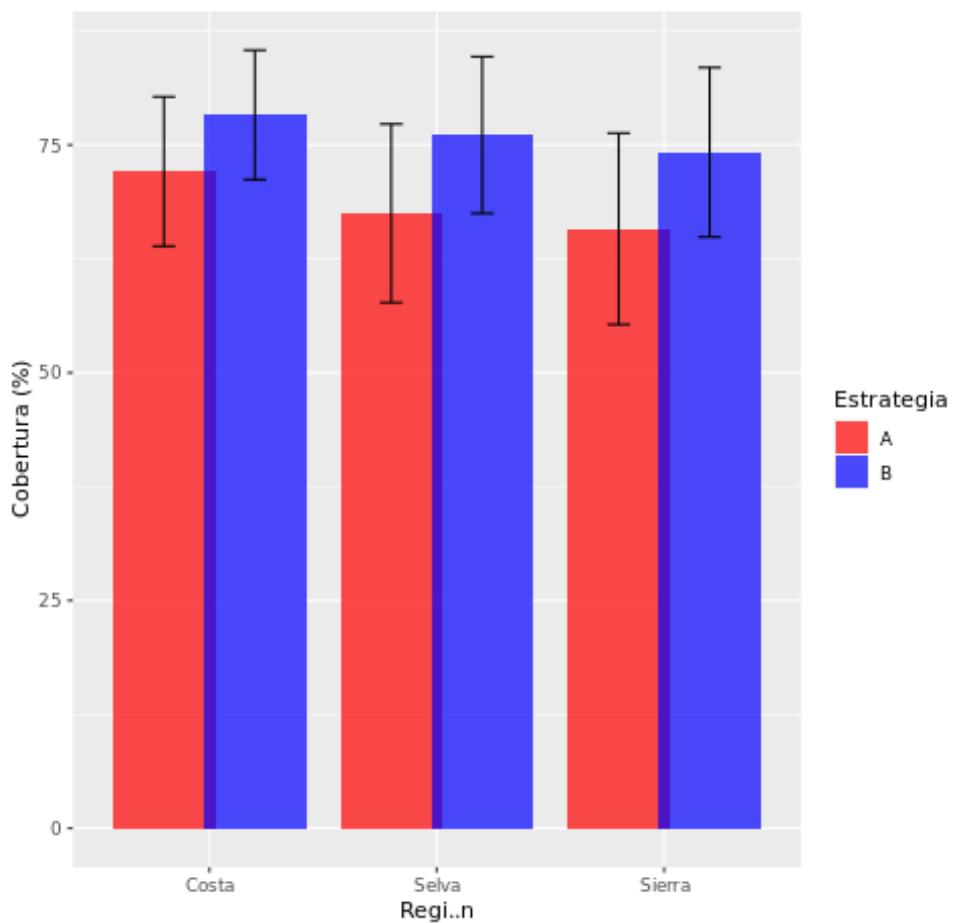


Figura 10: Cobertura de vacunación por región y estrategia

Ejercicio 4: Análisis de Sentimientos en Redes Sociales

Código R

```
1 library(ggplot2)
2
3 # Datos proporcionados
4 n_total <- 2400
5 positivos <- 1680
6 neutrales <- 480
7 negativos <- 240
8
9 p_hat <- positivos / n_total
10
11 # a) IC usando m todo normal y Wilson - MOSTRAR RESULTADOS
12 z <- 1.96
13
14 # M todo normal
15 ic_normal <- p_hat + c(-1, 1) * z * sqrt(p_hat * (1 - p_hat) / n_total)
16
17 # M todo Wilson
18 wilson_denom <- 1 + z^2 / n_total
19 wilson_center <- (p_hat + z^2 / (2 * n_total)) / wilson_denom
20 wilson_margin <- (z * sqrt(p_hat * (1 - p_hat) / n_total + z^2 / (4 * n_
    total^2))) / wilson_denom
21 ic_wilson <- c(wilson_center - wilson_margin, wilson_center + wilson_
    margin)
22
23 print("a) Intervalos de Confianza 95% para sentimiento positivo:")
24 print(paste("M todo Normal:", round(ic_normal[1], 4), "-", round(ic_
    normal[2], 4)))
25 print(paste("M todo Wilson:", round(ic_wilson[1], 4), "-", round(ic_
    wilson[2], 4)))
26
27 # b) Gráfico de barras apiladas semanal
28 set.seed(123)
29 semanas <- 1:12
30
31 # Crear datos semanales
32 datos_semanales <- data.frame(
33   Semana = semanas,
34   Positivo = runif(12, 0.65, 0.75),
35   Neutral = runif(12, 0.15, 0.25),
36   Negativo = runif(12, 0.05, 0.15)
37 )
38
39 # Normalizar para que sumen 1
40 for(i in 1:nrow(datos_semanales)) {
41   total <- sum(datos_semanales[i, c("Positivo", "Neutral", "Negativo")])
42   datos_semanales[i, c("Positivo", "Neutral", "Negativo")] <- datos_
        semanales[i, c("Positivo", "Neutral", "Negativo")] / total
43 }
44
45 # Convertir a formato largo
46 datos_largos <- data.frame(
47   Semana = rep(semanas, 3),
48   Sentimiento = rep(c("Positivo", "Neutral", "Negativo"), each = 12),
```

```

49 Proporcion = c(datos_semanales$Positivo, datos_semanales$Neutral,
50   datos_semanales$Negativo)
51 )
52 print("b) Gráfico de barras apiladas semanal:")
53 ggplot(datos_largos, aes(x = factor(Semana), y = Proporcion, fill =
54   Sentimiento)) +
55   geom_col(position = "stack") +
56   scale_fill_manual(values = c("Positivo" = "green", "Neutral" = "gray",
57     "Negativo" = "red")) +
58   labs(title = "Evolución semanal del sentimiento en publicaciones",
59     x = "Semana", y = "Proporción")
60
61 # c) Evolución del sentimiento positivo con banda de confianza
62 datos_evolucion <- data.frame(
63   Semana = semanas,
64   Positivo = datos_semanales$Positivo,
65   error = sqrt(datos_semanales$Positivo * (1 - datos_semanales$Positivo)
66     / 200),
67   inf = datos_semanales$Positivo - 1.96 * sqrt(datos_semanales$Positivo
68     * (1 - datos_semanales$Positivo) / 200),
69   sup = datos_semanales$Positivo + 1.96 * sqrt(datos_semanales$Positivo
70     * (1 - datos_semanales$Positivo) / 200)
71 )
72
73 print("c) Evolución del sentimiento positivo con IC:")
74 ggplot(datos_evolucion, aes(x = Semana, y = Positivo)) +
75   geom_line(color = "green", size = 1) +
76   geom_point(color = "darkgreen", size = 2) +
77   geom_ribbon(aes(ymin = inf, ymax = sup), alpha = 0.2, fill = "green")
78   +
79   labs(title = "Evolución del sentimiento positivo con IC 95%",
80     x = "Semana", y = "Proporción de sentimiento positivo") +
81   ylim(0, 1)
82
83 # e) Funnel plot para identificar semanas atípicas
84 print("e) Funnel plot para semanas atípicas:")
85 ggplot(datos_evolucion, aes(x = Positivo, y = error)) +
86   geom_point(size = 3, color = "blue") +
87   geom_vline(xintercept = p_hat, linetype = "dashed", color = "red") +
88   geom_line(aes(x = p_hat + 1.96 * error), linetype = "dashed", color =
89     "gray") +
90   geom_line(aes(x = p_hat - 1.96 * error), linetype = "dashed", color =
91     "gray") +
92   labs(title = "Funnel plot para identificar semanas atípicas",
93     x = "Proporción de sentimiento positivo",
94     y = "Error estándar")
95
96 # f) Gráfico de rea apilada normalizada
97 print("f) Gráfico de rea apilada normalizada:")
98 ggplot(datos_largos, aes(x = Semana, y = Proporcion, fill = Sentimiento))
99   +
100   geom_area(position = "fill") +
101   scale_fill_manual(values = c("Positivo" = "green", "Neutral" = "gray",
102     "Negativo" = "red")) +
103   labs(title = "Evolución temporal del sentimiento (100% apilado)",
104     x = "Semana", y = "Proporción")

```

Resultados Ejercicio 4

- a) Intervalos de Confianza 95% para sentimiento positivo:
[1] "Método Normal: 0.682 - 0.718"
[1] "Método Wilson: 0.6823 - 0.7176"

Gráficos Ejercicio 4

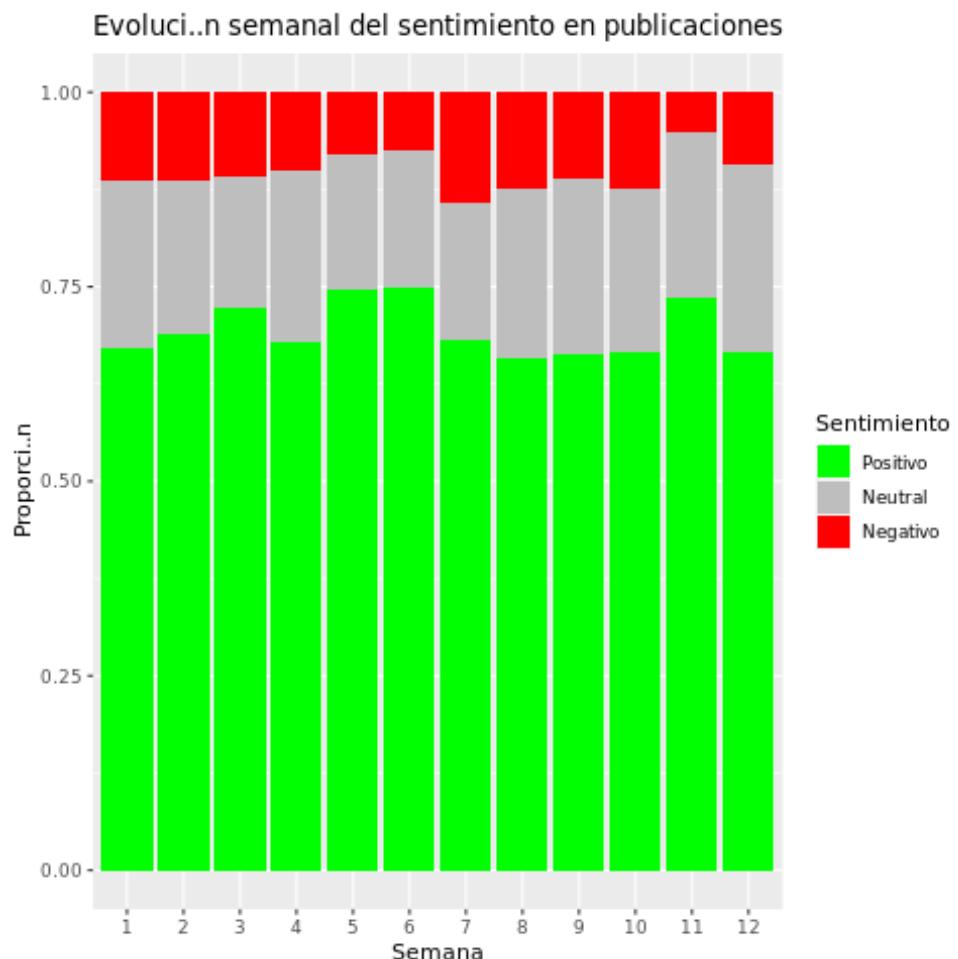


Figura 11: Evolución semanal del sentimiento en publicaciones

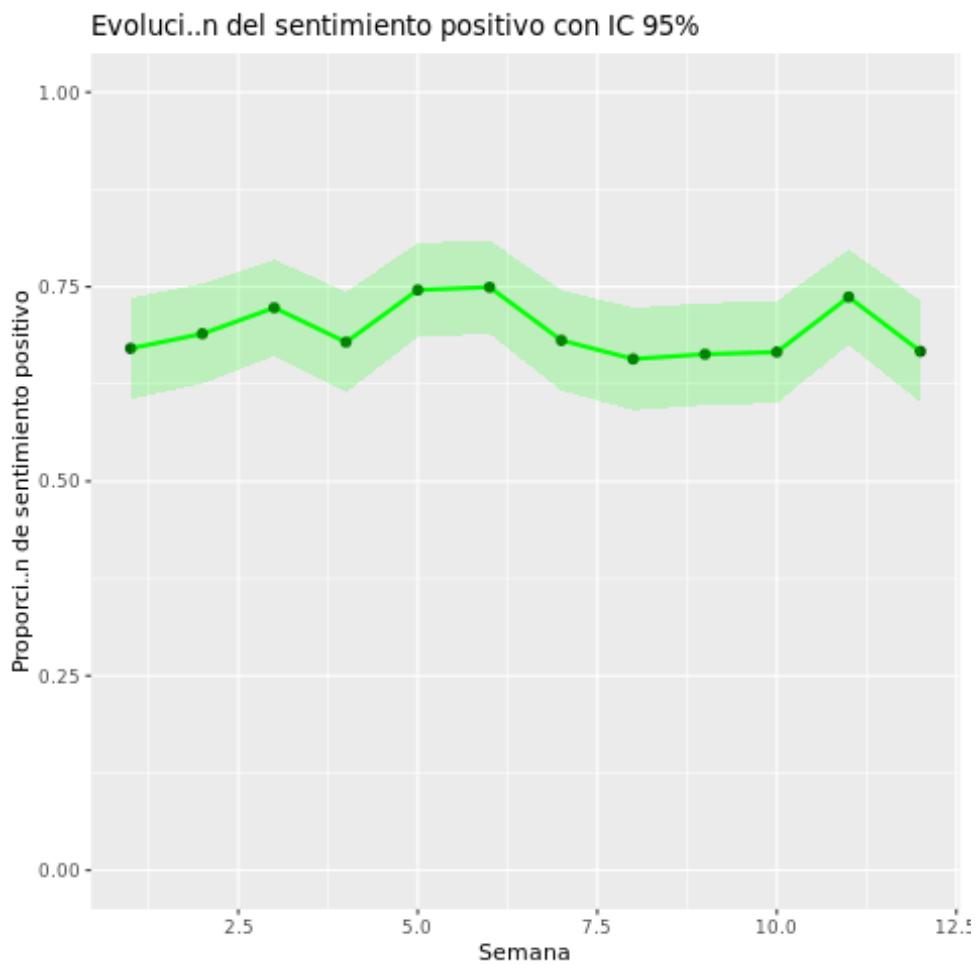


Figura 12: Evolución del sentimiento positivo con IC 95 %

Funnel plot para identificar semanas at..picas

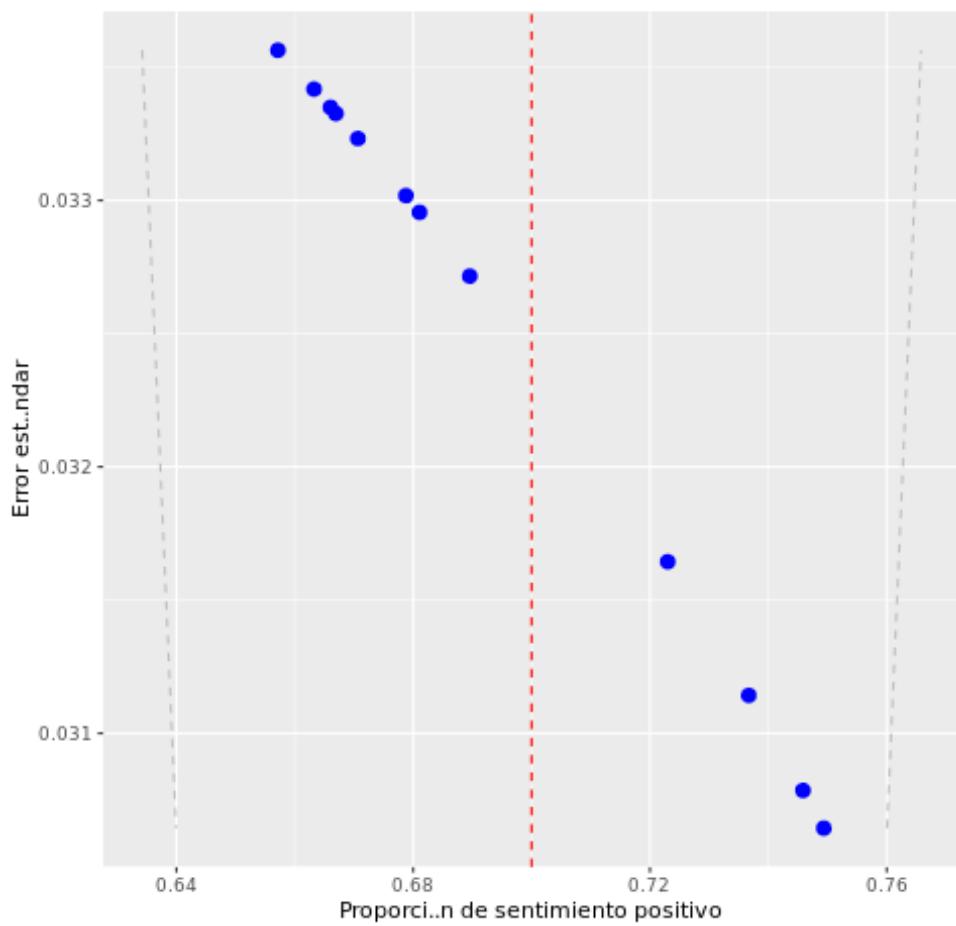


Figura 13: Funnel plot para identificar semanas atípicas

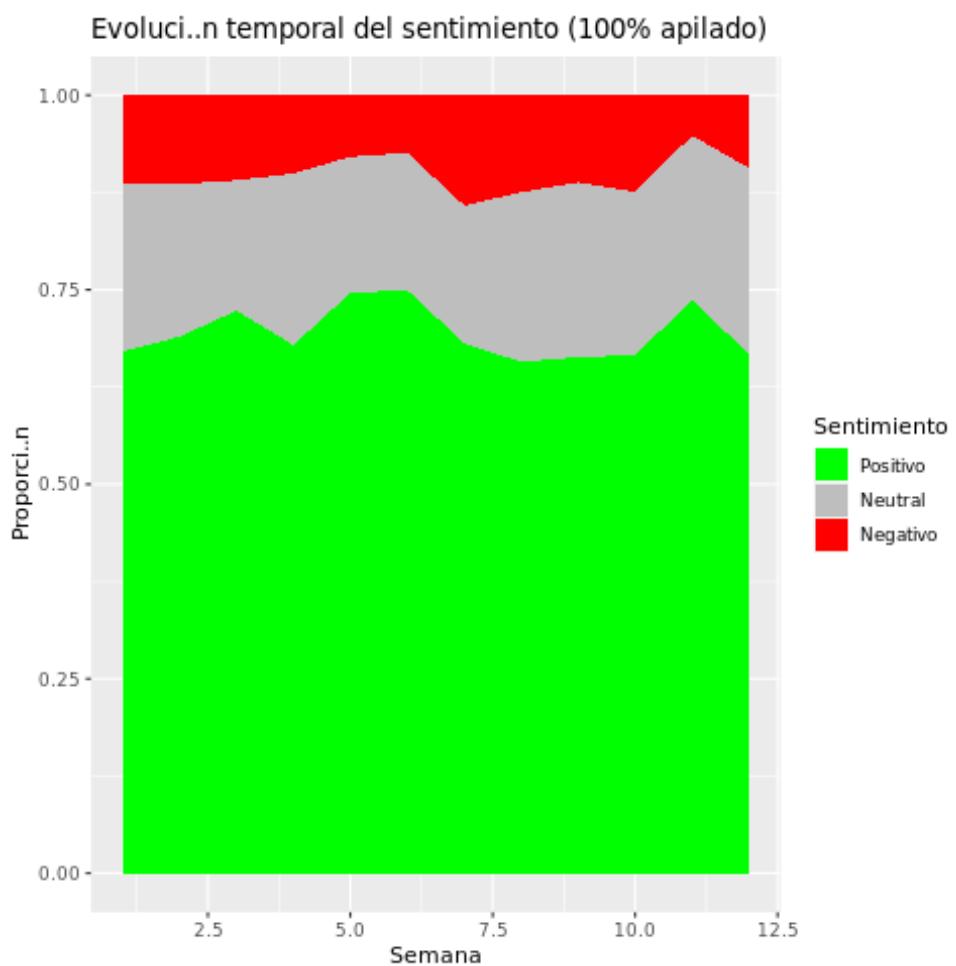


Figura 14: Evolución temporal del sentimiento (100 % apilado)

Ejercicio 5: Modelos Predictivos para Demanda de Energía

Código R

```
1 library(ggplot2)
2
3 # Datos proporcionados
4 modelos <- c("ARIMA", "Random Forest", "XGBoost", "LSTM", "Regresi n
5     Lineal")
6 maes <- c(245, 198, 185, 192, 267)
7 sds <- c(35, 28, 32, 41, 38)
8 n <- 40
9 baseline <- 280
10
11 # a) IC para cada modelo - MOSTRAR RESULTADOS
12 calcular_ic_mae <- function(mae, sd, n, confianza = 0.95) {
13     alpha <- 1 - confianza
14     t_val <- qt(1 - alpha/2, df = n - 1)
15     error <- t_val * sd / sqrt(n)
16     return(c(mae - error, mae + error))
17 }
18
19 ics_modelos <- data.frame(
20     Modelo = modelos,
21     MAE = maes,
22     Inferior = numeric(length(modelos)),
23     Superior = numeric(length(modelos))
24 )
25
26 for(i in 1:length(modelos)) {
27     ic <- calcular_ic_mae(maes[i], sds[i], n)
28     ics_modelos$Inferior[i] <- ic[1]
29     ics_modelos$Superior[i] <- ic[2]
30 }
31
32 print("a) Intervalos de Confianza 95% para MAE de cada modelo:")
33 print(ics_modelos)
34
35 # b) Forest plot horizontal
36 ics_modelos$Modelo <- factor(ics_modelos$Modelo,
37                               levels = ics_modelos$Modelo[order(ics_
38                               modelos$MAE)])
39
40 print("b) Forest plot comparando modelos:")
41 ggplot(ics_modelos, aes(x = MAE, y = Modelo)) +
42     geom_point(size = 3, color = "blue") +
43     geom_errorbarh(aes(xmin = Inferior, xmax = Superior),
44                     height = 0.2, color = "blue", size = 1) +
45     geom_vline(xintercept = baseline, color = "red", linetype = "dashed",
46                size = 1) +
47     labs(title = "Comparaci n de modelos predictivos - MAE con IC 95%",
48          x = "MAE (MW)", y = "Modelo")
49
50 # c) L nea vertical roja para baseline (ya incluida en el forest plot)
51 print("c) Baseline incluido en forest plot")
```

```

50 # d) Boxplot comparando distribuciones de error
51 set.seed(123)
52 datos_boxplot <- data.frame(
53   Modelo = rep(modelos, each = 100),
54   Error = c(
55     rnorm(100, maes[1], sds[1]),
56     rnorm(100, maes[2], sds[2]),
57     rnorm(100, maes[3], sds[3]),
58     rnorm(100, maes[4], sds[4]),
59     rnorm(100, maes[5], sds[5])
60   )
61 )
62
63 print("d) Boxplot de distribuciones de error:")
64 ggplot(datos_boxplot, aes(x = Modelo, y = Error)) +
65   geom_boxplot(fill = "lightblue", alpha = 0.7) +
66   geom_hline(yintercept = baseline, color = "red", linetype = "dashed",
67   size = 1) +
68   labs(title = "Distribución de errores por modelo",
69       x = "Modelo", y = "MAE (MW)") +
70   theme(axis.text.x = element_text(angle = 45, hjust = 1))
71
72 # f) Heatmap de los límites del IC
73 datos_heatmap <- data.frame(
74   Modelo = rep(modelos, 2),
75   Limite = rep(c("Inferior", "Superior"), each = 5),
76   Valor = c(ics_modelos$Inferior, ics_modelos$Superior)
77 )
78
79 print("f) Heatmap de los límites del IC:")
80 ggplot(datos_heatmap, aes(x = Limite, y = Modelo, fill = Valor)) +
81   geom_tile(color = "white") +
82   geom_text(aes(label = round(Valor, 1)), color = "black", size = 3) +
83   scale_fill_gradient2(low = "green", high = "red", mid = "yellow",
84   midpoint = 220) +
85   labs(title = "Los límites inferior y superior del IC por modelo",
86       x = "Límite del IC", y = "Modelo")
87
88 # g) Identificación visual de modelos que superan baseline
89 print("g) Modelos que superan consistentemente el baseline (280 MW):")
90 for(i in 1:nrow(ics_modelos)) {
91   if(ics_modelos$Superior[i] < baseline) {
92     print(paste(" ", ics_modelos$Modelo[i], "- MAE máximo:", round(
93       ics_modelos$Superior[i], 1), "MW"))
94   }
95 }

```

Resultados Ejercicio 5

a) Intervalos de Confianza 95% para MAE de cada modelo:

Modelo MAE Inferior Superior

1	ARIMA	245	233.8979	256.1021
2	Random Forest	198	189.3196	206.6804
3	XGBoost	185	175.0816	194.9184
4	LSTM	192	179.2974	204.7026
5	Regresión Lineal	267	255.2221	278.7779

g) Modelos que superan consistentemente el baseline (280 MW):

[1] " Random Forest - MAE máximo: 206.7 MW"

[1] " XGBoost - MAE máximo: 194.9 MW"

[1] " LSTM - MAE máximo: 204.7 MW"

Gráficos Ejercicio 5

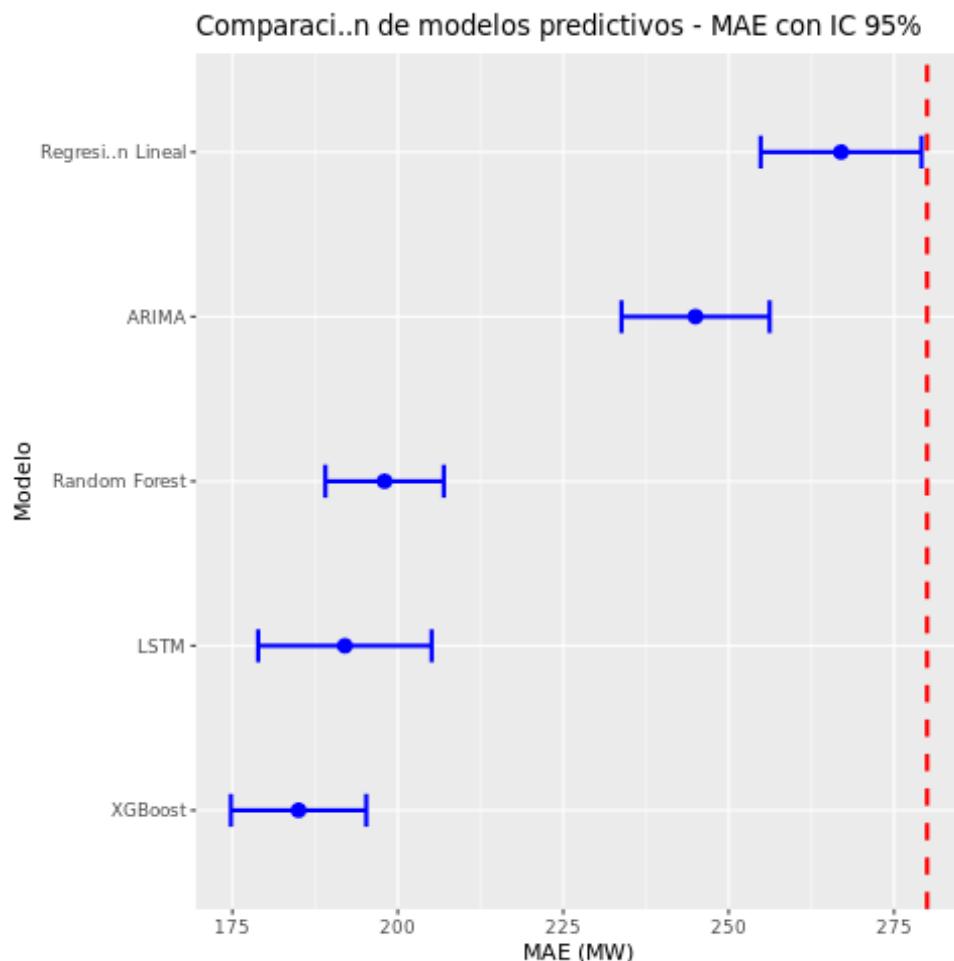


Figura 15: Comparación de modelos predictivos - MAE con IC 95 %

Distribuci..n de errores por modelo

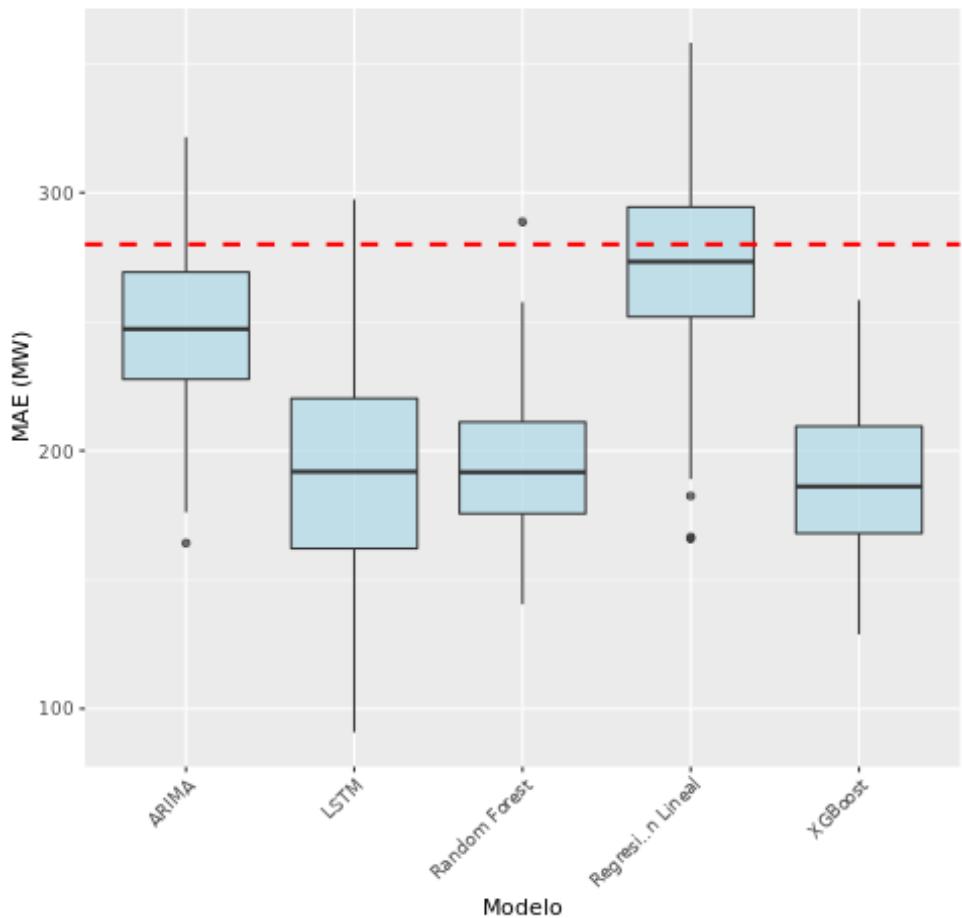


Figura 16: Distribución de errores por modelo

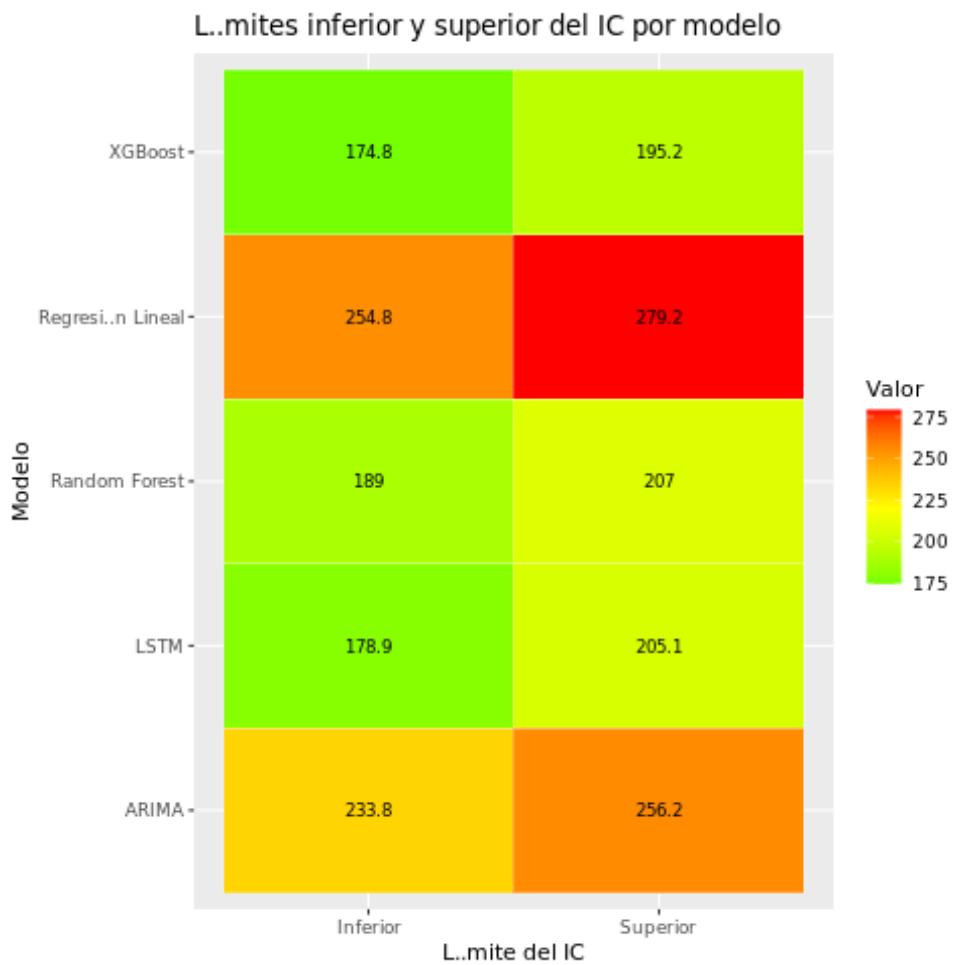


Figura 17: Límites inferior y superior del IC por modelo

Ejercicio 6: Impacto de Tutoría Virtual en Universidades

Código R

```
1 library(ggplot2)
2
3 # Datos proporcionados
4 control <- list(media = 13.2, sd = 2.1, n = 450)
5 experimental <- list(media = 14.5, sd = 1.8, n = 430)
6
7 # a) IC al 99% para cada grupo - MOSTRAR RESULTADOS
8 calcular_ic_grupo <- function(media, sd, n, confianza = 0.99) {
9   alpha <- 1 - confianza
10  z <- qnorm(1 - alpha/2) # Para muestras grandes usamos normal
11  error <- z * sd / sqrt(n)
12  return(c(media - error, media + error))
13}
14
15 ic_control <- calcular_ic_grupo(control$media, control$sd, control$n)
16 ic_experimental <- calcular_ic_grupo(experimental$media, experimental$sd,
17 , experimental$n)
18
19 print("a) Intervalos de Confianza 99%:")
20 print(paste("Grupo Control:", round(ic_control[1], 2), "-", round(ic_
21 control[2], 2)))
22 print(paste("Grupo Experimental:", round(ic_experimental[1], 2), "-",
23 round(ic_experimental[2], 2)))
24
25 # b) Gráfico de violín comparando distribuciones
26 set.seed(123)
27 datos_violin <- data.frame(
28   Grupo = rep(c("Control", "Experimental"), c(control$n, experimental$n)),
29   Puntaje = c(rnorm(control$n, control$media, control$sd),
30               rnorm(experimental$n, experimental$media, experimental$sd))
31 )
32
33 print("b) Gráfico de violín comparando distribuciones:")
34 ggplot(datos_violin, aes(x = Grupo, y = Puntaje, fill = Grupo)) +
35   geom_violin(alpha = 0.7, trim = FALSE) +
36   geom_boxplot(width = 0.1, fill = "white", alpha = 0.7) +
37   scale_fill_manual(values = c("Control" = "red", "Experimental" = "blue"))
38   labs(title = "Distribución de promedios ponderados por grupo",
39         x = "Grupo", y = "Promedio Ponderado")
40
41 # c) Forest plot por área académica
42 areas <- c("Ingenierías", "Ciencias de la Salud", "Ciencias Sociales",
43           "Humanidades")
44 diferencias <- c(1.1, 1.5, 0.9, 1.3)
45 sd_areas <- c(0.3, 0.4, 0.35, 0.38)
46
47 datos_areas <- data.frame(
48   Area = areas,
49   Diferencia = diferencias,
```

```

46 Inferior = diferencias - 2.576 * sd_areas,
47 Superior = diferencias + 2.576 * sd_areas
48 )
49
50 print("c) Forest plot por rea acad mica:")
51 ggplot(datos_areas, aes(x = Diferencia, y = Area)) +
52   geom_point(size = 3, color = "purple") +
53   geom_errorbarh(aes(xmin = Inferior, xmax = Superior),
54                 height = 0.1, color = "purple", size = 1) +
55   geom_vline(xintercept = 0, linetype = "dashed", color = "gray") +
56   labs(title = "Diferencia de mejora por rea acad mica (IC 99%)",
57         x = "Diferencia de promedio (Experimental - Control)",
58         y = "rea Acad mica")
59
60 # d) IC para diferencia global - MOSTRAR RESULTADO
61 calcular_ic_diferencia_global <- function(media1, sd1, n1, media2, sd2,
62   n2, confianza = 0.99) {
63   z <- qnorm(1 - (1 - confianza)/2)
64   diferencia <- media2 - media1
65   error <- z * sqrt(sd1^2/n1 + sd2^2/n2)
66   return(c(diferencia - error, diferencia + error))
67 }
68
69 ic_diferencia_global <- calcular_ic_diferencia_global(
70   control$media, control$sd, control$n,
71   experimental$media, experimental$sd, experimental$n
72 )
73
74 print("d) Diferencia global entre grupos:")
75 print(paste("Diferencia media:", round(experimental$media - control$media, 2)))
76 print(paste("IC 99% diferencia:", round(ic_diferencia_global[1], 2), "-",
77             round(ic_diferencia_global[2], 2)))
78
79 # e) Gr fico de densidad superpuesta
80 print("e) Gr fico de densidad superpuesta:")
81 ggplot(datos_violin, aes(x = Puntaje, fill = Grupo)) +
82   geom_density(alpha = 0.5) +
83   scale_fill_manual(values = c("Control" = "red", "Experimental" = "blue"))
84   labs(title = "Distribuciones de densidad superpuestas",
85         x = "Promedio Ponderado", y = "Densidad")
86
87 # g) Scatter plot bivariado
88 set.seed(123)
89 universidades <- 30
90 promedio_inicial <- runif(universidades, 11, 14)
91 mejora <- runif(universidades, 0.5, 2.0)
92
93 datos_scatter <- data.frame(
94   Universidad = paste0("U", 1:universidades),
95   Promedio_Inicial = promedio_inicial,
96   Mejora = mejora
97 )
98
99 print("g) Scatter plot bivariado:")
100 ggplot(datos_scatter, aes(x = Promedio_Inicial, y = Mejora)) +
101   geom_point(size = 3, alpha = 0.7, color = "blue") +

```

```

100     geom_smooth(method = "lm", color = "red", se = TRUE) +
101   labs(title = "Relaci n entre promedio inicial y mejora",
102       x = "Promedio Inicial", y = "Mejora (puntos)")

```

Resultados Ejercicio 6

- a) Intervalos de Confianza 99%:
- [1] "Grupo Control: 12.94 - 13.46"
- [1] "Grupo Experimental: 14.28 - 14.72"
- d) Diferencia global entre grupos:
- [1] "Diferencia media: 1.3"
- [1] "IC 99% diferencia: 1.02 - 1.58"

Gráficos Ejercicio 6

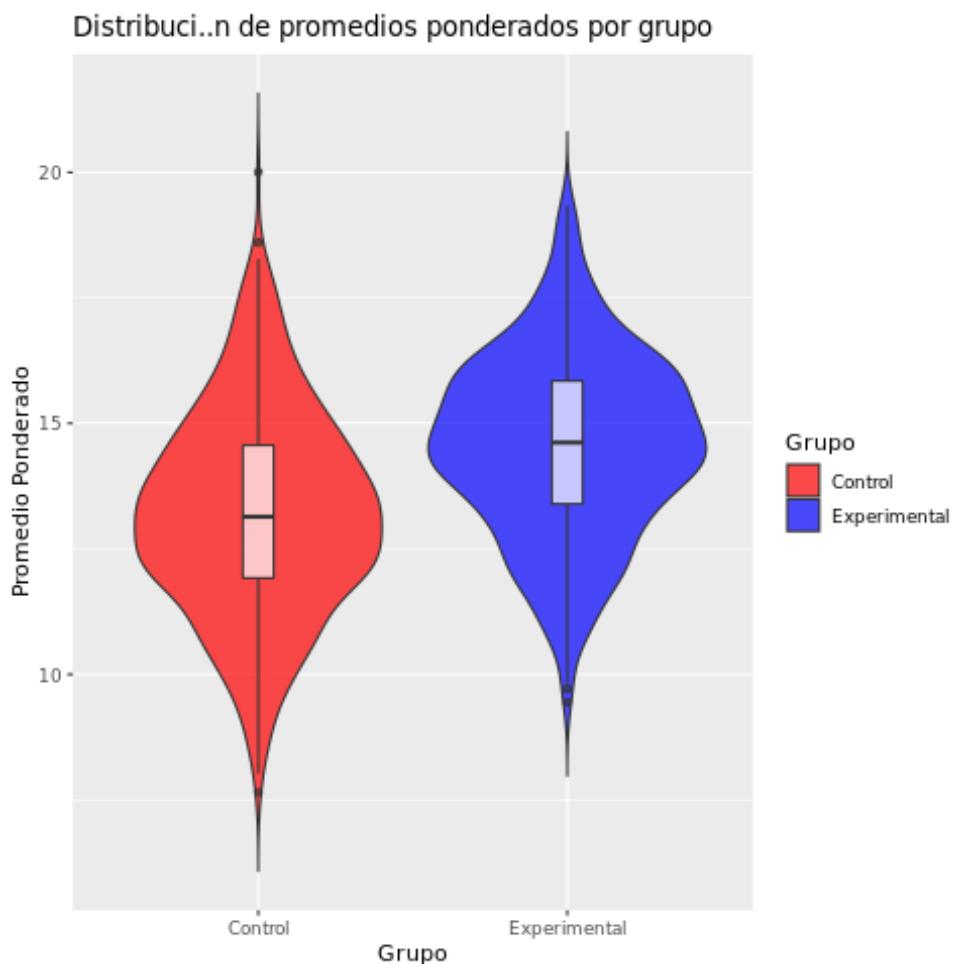


Figura 18: Distribución de promedios ponderados por grupo

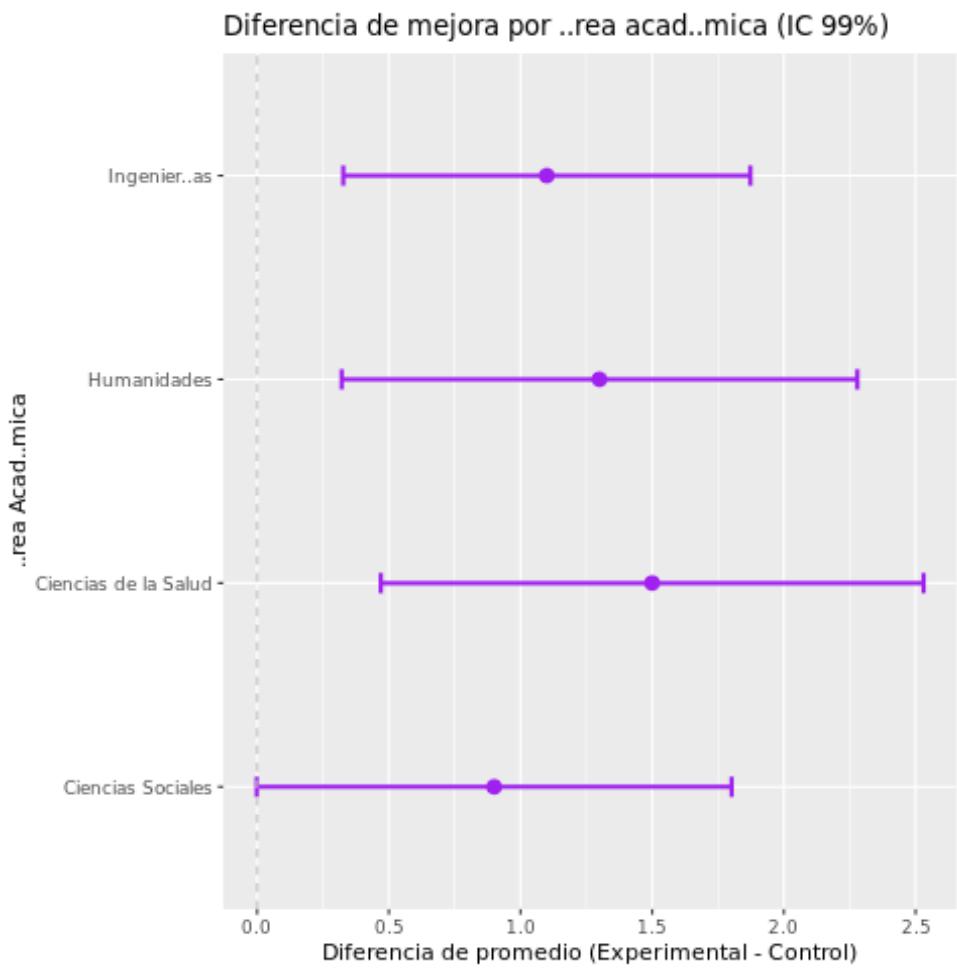


Figura 19: Diferencia de mejora por área académica (IC 99 %)

Distribuciones de densidad superpuestas

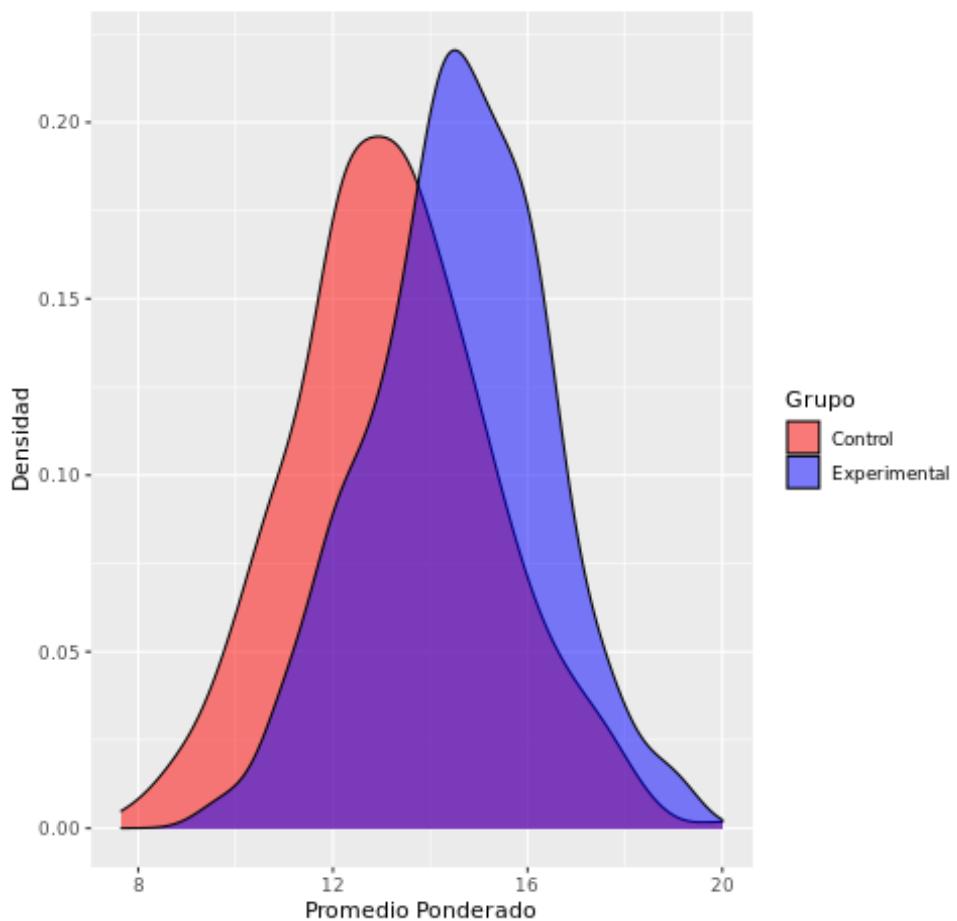


Figura 20: Distribuciones de densidad superpuestas

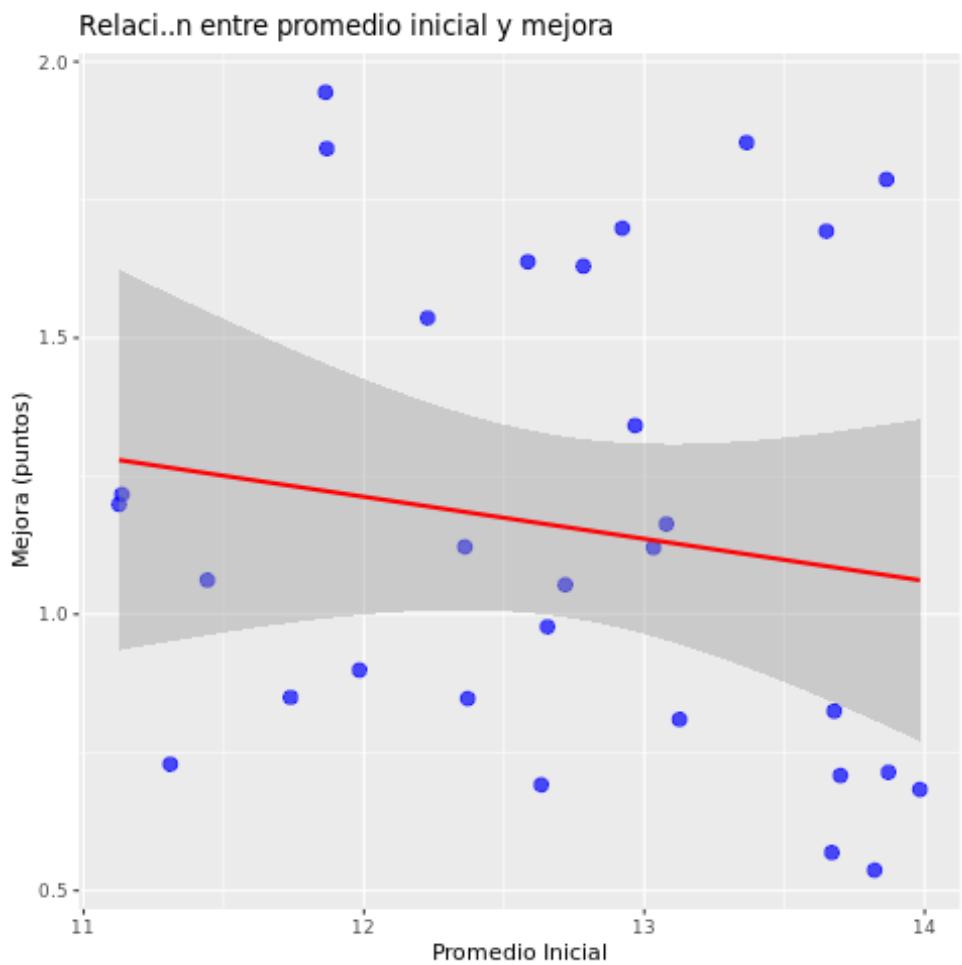


Figura 21: Relación entre promedio inicial y mejora