

Taller_EDA

Roberto Coto

2025-12-10

Contents

3.2 Paso 2: Cargar las Librerías	2
3.3 Paso 3: Importar el Dataset	2
4 Exploración Inicial de los Datos	3
4.1 Ver las Primeras y Ultimas Filas	3
4.2 Estructura del Dataset	4
4.3 Resumen Estadístico Rápido	4
5 Limpieza y Preparación de Datos	5
5.1 Renombrar las Columnas	5
5.1 Renombrar las Columnas	5
6 Estadística Descriptiva: Nota del Examen Final	5
6.1 Medidas de Tendencia Central	5
6.2 Medidas de Dispersión	6
7 Visualización de Datos	7
7.1 Gráfico de Pastel: Categorías de Rendimiento	7
7.3 Diagrama de Cajas: Exámenes Internos 1 y 2	8
7.4 Gráfico de Violín: Nota Final por Nivel de Asistencia	10
8 Análisis de Normalidad: Nota del Examen Final	11
8.1 Histograma de la Nota Final	11
8.2 Gráfico Q-Q (Quantile-Quantile)	12
9.2 Gráfico de Violín	17
10 Análisis de Correlación	19
10.1 Mapa de Calor (Heatmap) de Correlaciones	19
10.2 Gráfico de Dispersión: Asistencia vs Nota Final	22

11 Regresión Lineal Simple	24
11.1 Crear el Modelo de Regresión	24
11.2 Interpretar los Resultados	24
11.3 Visualizar el Modelo de Regresión	25
11.4 Realizar Predicciones	26

3.2 Paso 2: Cargar las Librerías

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
##
```

```
## Adjuntando el paquete: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.4.3
```

```
## corrplot 0.95 loaded
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.4.3
```

```
options(scipen = 999) # evitar notación científica
```

3.3 Paso 3: Importar el Dataset

```
# 1. Importar el dataset
```

```
# NOTA: Ajustar la ruta según la ubicación del archivo
```

```
datos <- read.csv("C:\\Users\\Roberto\\Downloads\\Calificaciones.csv")
```

```
# 2. Verificar que se cargó correctamente
```

```
cat("El dataset tiene", nrow(datos), "filas y", ncol(datos), "columnas\n")
```

```
## El dataset tiene 2000 filas y 7 columnas
```

4 Exploración Inicial de los Datos

4.1 Ver las Primeras y Ultimas Filas

```
head(datos)
```

```
##      Student_ID Attendance.... Internal.Test.1..out.of.40.
## 1      S1000           84                30
## 2      S1001           91                24
## 3      S1002           73                29
## 4      S1003           80                36
## 5      S1004           84                31
## 6      S1005          100                34
##      Internal.Test.2..out.of.40. Assignment.Score..out.of.10. Daily.Study.Hours
## 1                36                7                3
## 2                38                6                3
## 3                26                7                3
## 4                35                7                3
## 5                37                8                3
## 6                34                7                3
##      Final.Exam.Marks..out.of.100.
## 1                72
## 2                56
## 3                56
## 4                74
## 5                66
## 6                79
```

```
tail(datos)
```

```
##      Student_ID Attendance.... Internal.Test.1..out.of.40.
## 1995      S2994           97                40
## 1996      S2995           82                31
## 1997      S2996           78                38
## 1998      S2997           78                30
## 1999      S2998           82                29
## 2000      S2999           97                32
##      Internal.Test.2..out.of.40. Assignment.Score..out.of.10. Daily.Study.Hours
## 1995                38                9                4
## 1996                28                6                2
## 1997                27                7                2
## 1998                33                9                2
## 1999                40                8                3
## 2000                38                7                3
##      Final.Exam.Marks..out.of.100.
## 1995                81
## 1996                52
## 1997                57
## 1998                61
## 1999                59
## 2000                64
```

Número de filas: 6 Número de columnas: 7

4.2 Estructura del Dataset

```
# Ver estructura : tipos de variables
str ( datos )
```

```
## 'data.frame':    2000 obs. of  7 variables:
## $ Student_ID      : chr  "S1000" "S1001" "S1002" "S1003" ...
## $ Attendance....  : int   84  91  73  80  84 100  96  83  91  87 ...
## $ Internal.Test.1..out.of.40. : int   30  24  29  36  31  34  40  39  30  27 ...
## $ Internal.Test.2..out.of.40. : int   36  38  26  35  37  34  36  37  37  37 ...
## $ Assignment.Score..out.of.10. : int    7  6  7  7  8  7  8  7  8  8 ...
## $ Daily.Study.Hours : int    3  3  3  3  3  3  3  2  3  ...
## $ Final.Exam.Marks..out.of.100.: int   72  56  56  74  66  79  83  77  71  61 ...
```

```
# Ver nombres de las columnas
names ( datos )
```

```
## [1] "Student_ID"           "Attendance...."
## [3] "Internal.Test.1..out.of.40." "Internal.Test.2..out.of.40."
## [5] "Assignment.Score..out.of.10." "Daily.Study.Hours"
## [7] "Final.Exam.Marks..out.of.100."
```

1. Variables numéricas: 6
2. Variables de texto: 1

4.3 Resumen Estadístico Rápido

```
summary(datos)
```

```
##   Student_ID      Attendance....   Internal.Test.1..out.of.40.
## Length:2000      Min.    : 52.00   Min.    :18.00
## Class :character  1st Qu.: 80.00   1st Qu.:29.00
## Mode  :character  Median : 85.00   Median :32.00
##                               Mean   : 84.89   Mean   :32.12
##                               3rd Qu.: 90.00   3rd Qu.:35.00
##                               Max.    :100.00   Max.    :40.00
## Internal.Test.2..out.of.40. Assignment.Score..out.of.10. Daily.Study.Hours
## Min.    :16.00      Min.    : 4.000   Min.    :1.000
## 1st Qu.:29.00      1st Qu.: 7.000   1st Qu.:2.000
## Median :33.00      Median : 8.000   Median :3.000
## Mean   :32.46      Mean   : 7.507   Mean   :2.824
## 3rd Qu.:36.00      3rd Qu.: 8.000   3rd Qu.:3.000
## Max.    :40.00      Max.    :10.000   Max.    :5.000
## Final.Exam.Marks..out.of.100.
## Min.    : 25.00
## 1st Qu.: 58.00
## Median : 65.00
## Mean   : 64.86
## 3rd Qu.: 73.00
## Max.    :100.00
```

5 Limpieza y Preparación de Datos

5.1 Renombrar las Columnas

```
names(datos)
```

```
## [1] "Student_ID"           "Attendance..."
## [3] "Internal.Test.1..out.of.40." "Internal.Test.2..out.of.40."
## [5] "Assignment.Score..out.of.10." "Daily.Study.Hours"
## [7] "Final.Exam.Marks..out.of.100."
```

```
colnames(datos) <- c(
  "ID",
  "Asistencia",
  "Examen_Interno_1",
  "Examen_Interno_2",
  "Promedio_Tareas",
  "Horas_Estudio",
  "Nota_Final"
)
names(datos)
```

```
## [1] "ID"           "Asistencia"   "Examen_Interno_1" "Examen_Interno_2"
## [5] "Promedio_Tareas" "Horas_Estudio" "Nota_Final"
```

5.1 Renombrar las Columnas

```
colSums(is.na(datos))
```

```
##           ID           Asistencia Examen_Interno_1 Examen_Interno_2
##           0              0              0              0
## Promedio_Tareas Horas_Estudio      Nota_Final
##           0              0              0
```

```
cat("Total de valores faltantes:", sum(is.na(datos)), "\n")
```

```
## Total de valores faltantes: 0
```

6 Estadística Descriptiva: Nota del Examen Final

6.1 Medidas de Tendencia Central

```
media <- mean(datos$Nota_Final)
mediana <- median(datos$Nota_Final)
```

```

tabla_freq <- table(round(datos$Nota_Final, 0))
moda <- as.numeric(names(tabla_freq)[which.max(tabla_freq)])

cat("Media:", round(media, 2), "\n")

```

```
## Media: 64.86
```

```
cat("Mediana:", round(mediana, 2), "\n")
```

```
## Mediana: 65
```

```
cat("Moda:", moda, "\n")
```

```
## Moda: 66
```

6.2 Medidas de Dispersión

```

varianza <- var(datos$Nota_Final)
desv_est <- sd(datos$Nota_Final)
cuartiles <- quantile(datos$Nota_Final, c(0.25, 0.50, 0.75))
rango <- max(datos$Nota_Final) - min(datos$Nota_Final)

cat("Varianza:", round(varianza, 2), "\n")

```

```
## Varianza: 128.62
```

```
cat("Desviación Estándar:", round(desv_est, 2), "\n")
```

```
## Desviación Estándar: 11.34
```

```
cat("Q1:", round(cuartiles[1], 2), "\n")
```

```
## Q1: 58
```

```
cat("Q2:", round(cuartiles[2], 2), "\n")
```

```
## Q2: 65
```

```
cat("Q3:", round(cuartiles[3], 2), "\n")
```

```
## Q3: 73
```

```
cat("Rango:", round(rango, 2), "\n")
```

```
## Rango: 75
```

7 Visualización de Datos

7.1 Gráfico de Pastel: Categorías de Rendimiento

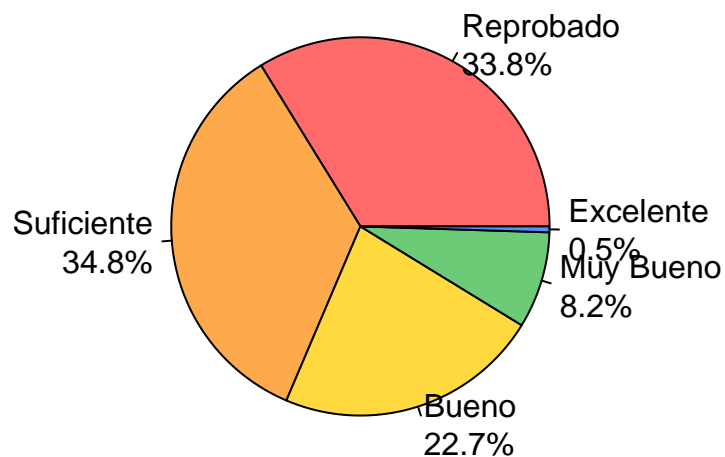
```
datos$Rendimiento <- cut(
  datos$Nota_Final,
  breaks = c(0,60,70,80,90,100),
  labels = c("Reprobado","Suficiente","Bueno","Muy Bueno","Excelente"),
  include.lowest = TRUE
)

frecuencias <- table(datos$Rendimiento)
porcentajes <- round(prop.table(frecuencias)*100,1)
etiquetas <- paste(names(frecuencias), "\n", porcentajes, "%", sep="")

colores <- c("#FF6B6B", "#FFA94D", "#FFD93D", "#6BCB77", "#4D96FF")

pie(
  frecuencias,
  labels = etiquetas,
  col = colores,
  main = "Distribución del Rendimiento Académico"
)
```

Distribución del Rendimiento Académico



7.2 Diagrama de Barras Horizontales: Nivel de Asistencia

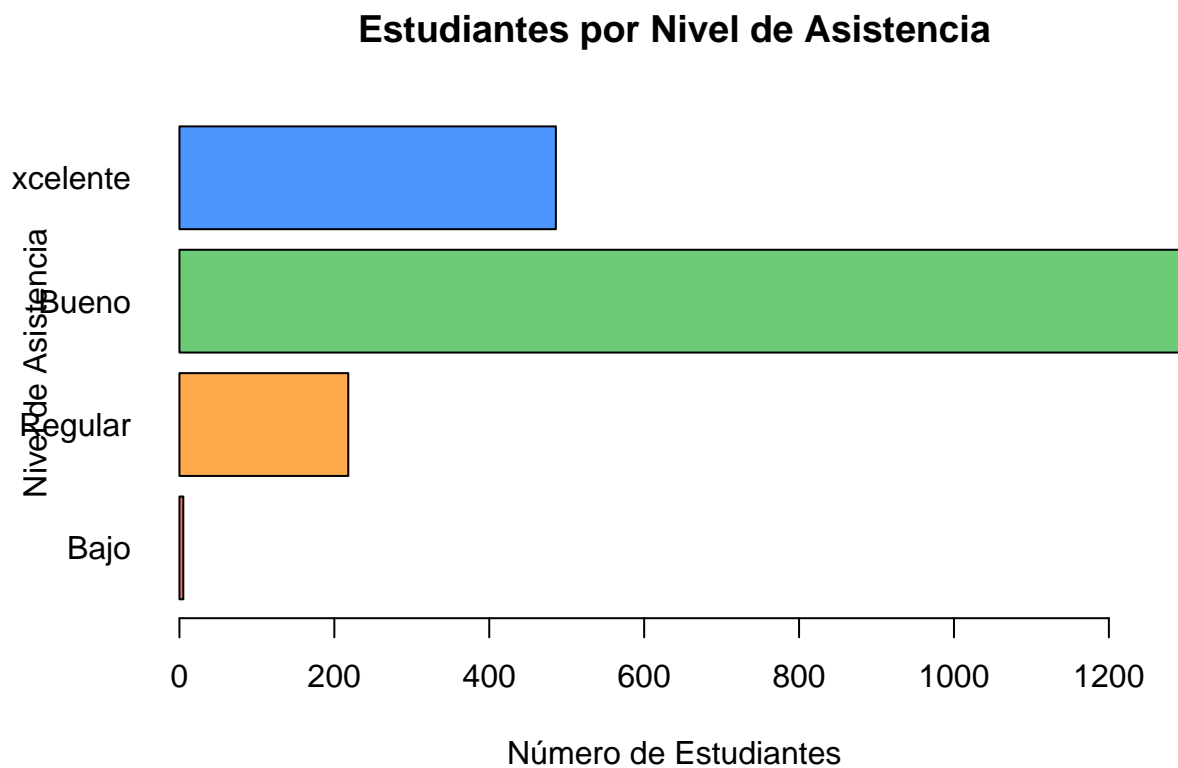
```

datos$Nivel_Asistencia <- cut(
  datos$Asistencia,
  breaks = c(0,60,75,90,100),
  labels = c("Bajo","Regular","Bueno","Excelente"),
  include.lowest = TRUE
)

freq_asistencia <- table(datos$Nivel_Asistencia)

barplot(
  freq_asistencia,
  horiz = TRUE,
  col = c("#FF6B6B", "#FFA94D", "#6BCB77", "#4D96FF"),
  main = "Estudiantes por Nivel de Asistencia",
  xlab = "Número de Estudiantes",
  ylab = "Nivel de Asistencia",
  las = 1
)

```



7.3 Diagrama de Cajas: Exámenes Internos 1 y 2

```

# Crear diagrama de cajas para ambos exámenes internos
boxplot(

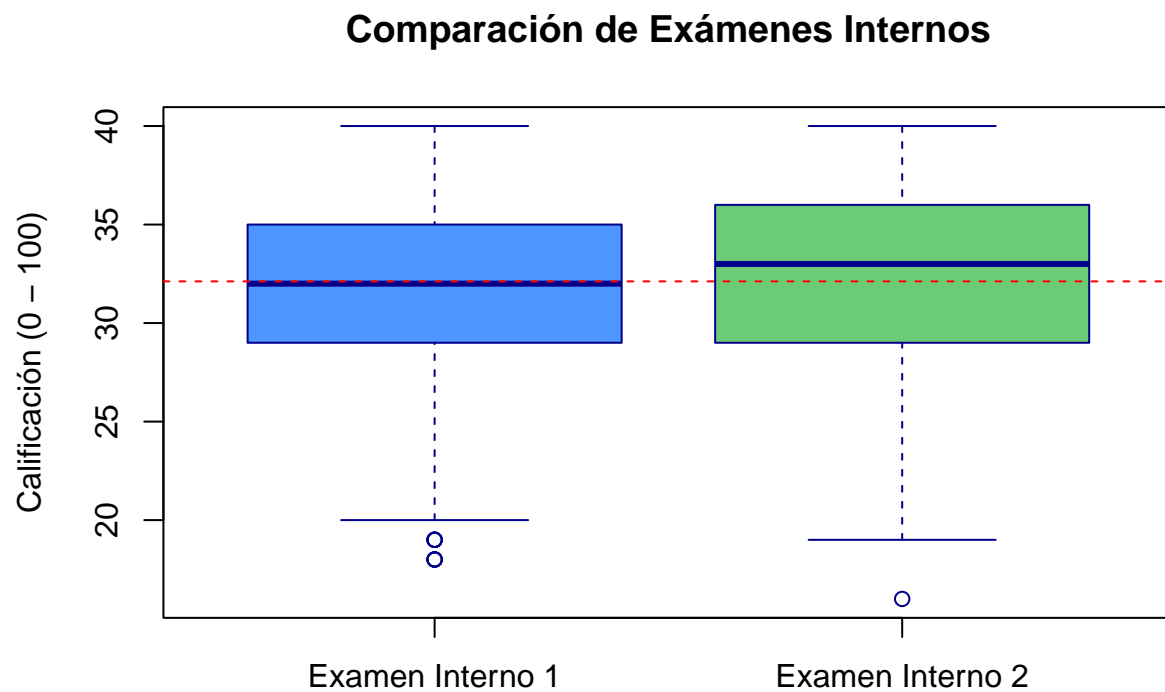
```

```

datos$Examen_Interno_1,
datos$Examen_Interno_2,
names = c("Examen Interno 1", "Examen Interno 2"),
col = c("#4D96FF", "#6BCB77"),
main = "Comparación de Exámenes Internos",
ylab = "Calificación (0 - 100)",
border = "darkblue"
)

# Agregar línea de la media del Examen Interno 1
abline(h = mean(datos$Examen_Interno_1), col = "red", lty = 2)

```



```

# Estadísticas del Examen Interno 1
cat("=== EXAMEN INTERNO 1 ===\n")

## === EXAMEN INTERNO 1 ===

cat("Media:", round(mean(datos$Examen_Interno_1), 2), "\n")

## Media: 32.12

cat("Mediana:", round(median(datos$Examen_Interno_1), 2), "\n")

## Mediana: 32

```

```
cat("Desv. Est.:", round(sd(datos$Examen_Interno_1), 2), "\n\n")
```

```
## Desv. Est.: 4.56
```

```
# Estadísticas del Examen Interno 2
```

```
cat("=== EXAMEN INTERNO 2 ===\n")
```

```
## === EXAMEN INTERNO 2 ===
```

```
cat("Media:", round(mean(datos$Examen_Interno_2), 2), "\n")
```

```
## Media: 32.46
```

```
cat("Mediana:", round(median(datos$Examen_Interno_2), 2), "\n")
```

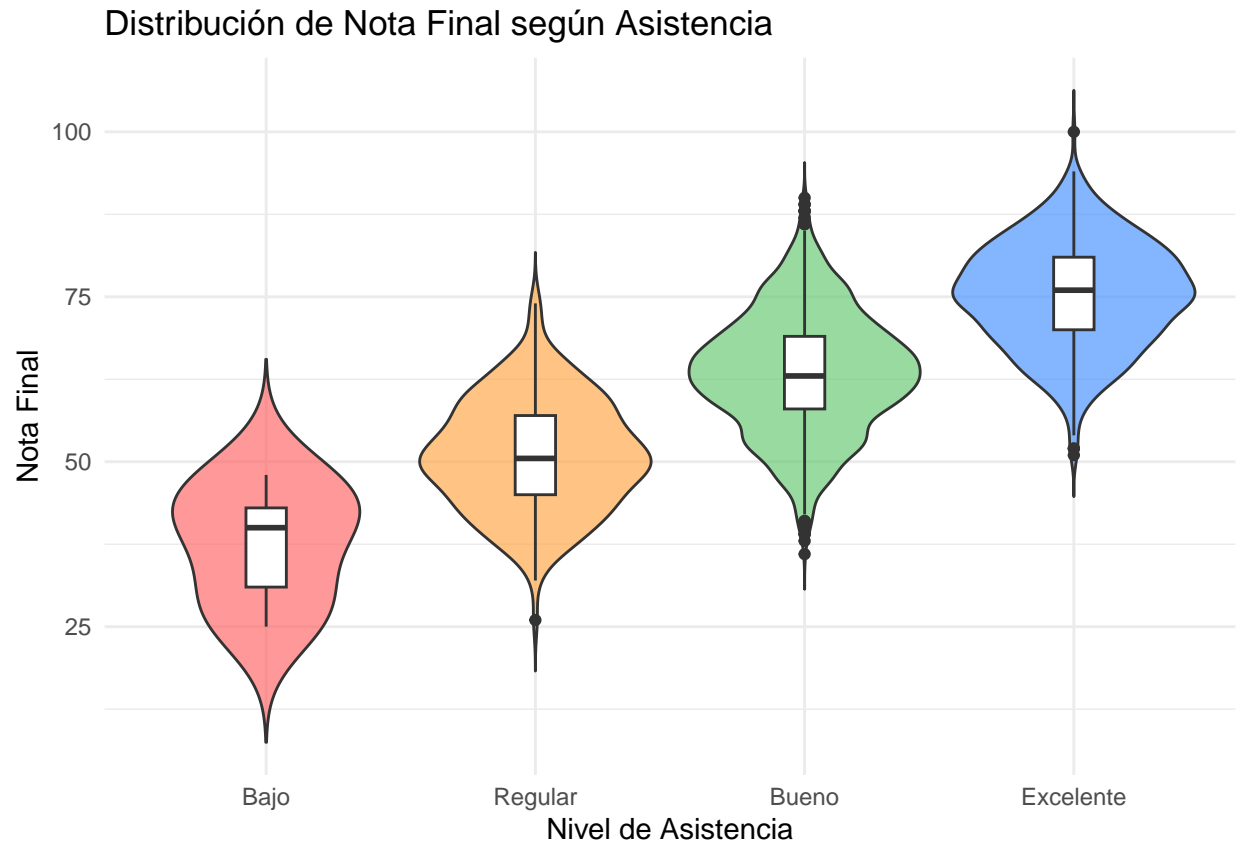
```
## Mediana: 33
```

```
cat("Desv. Est.:", round(sd(datos$Examen_Interno_2), 2), "\n")
```

```
## Desv. Est.: 4.52
```

7.4 Gráfico de Violín: Nota Final por Nivel de Asistencia

```
ggplot(datos, aes(x = Nivel_Asistencia, y = Nota_Final, fill = Nivel_Asistencia)) +  
  geom_violin(trim = FALSE, alpha = 0.7) +  
  geom_boxplot(width = 0.15, fill = "white") +  
  scale_fill_manual(values = c("#FF6B6B", "#FFA94D", "#6BCB77", "#4D96FF")) +  
  labs(  
    title = "Distribución de Nota Final según Asistencia",  
    x = "Nivel de Asistencia",  
    y = "Nota Final"  
  ) +  
  theme_minimal() +  
  theme(legend.position = "none")
```



8 Análisis de Normalidad: Nota del Examen Final

8.1 Histograma de la Nota Final

```
# Crear histograma
hist(
  datos$Nota_Final,
  breaks = 20,
  col = "#4D96FF",
  border = "white",
  main = "Histograma de Nota Final",
  xlab = "Nota del Examen Final",
  ylab = "Frecuencia",
  freq = FALSE # usar densidad en lugar de frecuencia
)

# Agregar curva normal teórica
curve(
  dnorm(
    x,
    mean = mean(datos$Nota_Final),
    sd = sd(datos$Nota_Final)
  ),

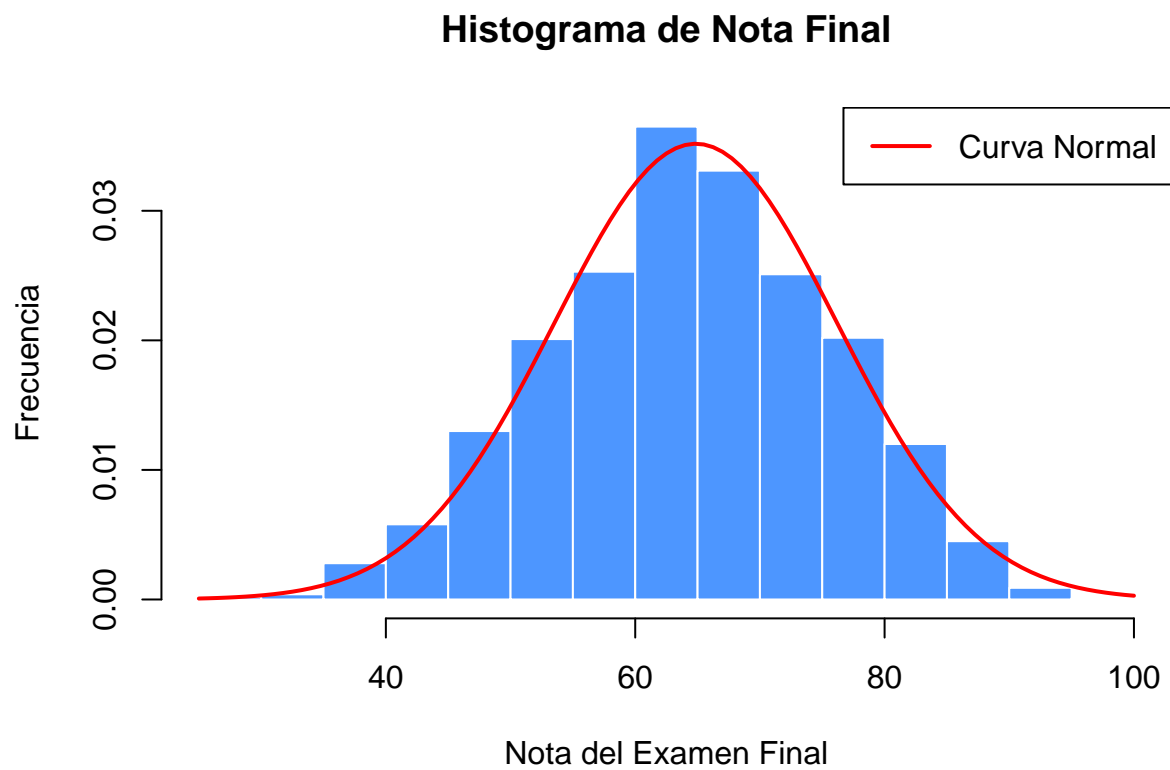
```

```

    add = TRUE,
    col = "red",
    lwd = 2
)

# Agregar leyenda
legend(
  "topright",
  legend = "Curva Normal",
  col = "red",
  lwd = 2
)

```



8.2 Gráfico Q-Q (Quantile-Quantile)

```

# Crear gráfico Q-Q
qqnorm(
  datos$Nota_Final,
  main = "Gráfico Q-Q: Nota Final",
  xlab = "Cuantiles Teóricos",
  ylab = "Cuantiles de los Datos",
  col = "#4D96FF",
  pch = 19,
)

```

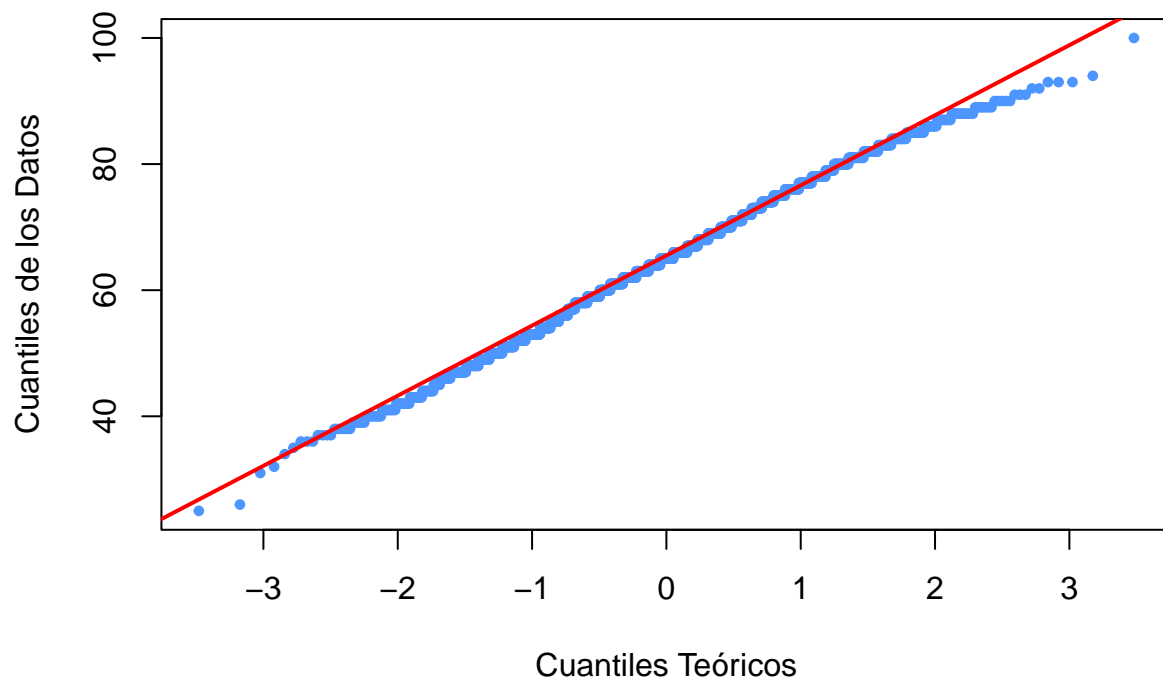
```

    cex = 0.6
  )

  # Agregar línea de referencia
  qqline(
    datos$Nota_Final,
    col = "red",
    lwd = 2
  )

```

Gráfico Q-Q: Nota Final



8.3 Pruebas Estadísticas de Normalidad

```

# Prueba de Shapiro-Wilk

# NOTA: Funciona mejor con menos de 5000 observaciones
shapiro_test <- shapiro.test(datos$Nota_Final)

cat("=== PRUEBA DE SHAPIRO-WILK ===\n")

```

=== PRUEBA DE SHAPIRO-WILK ===

```

cat("Estadístico W:", round(shapiro_test$statistic, 4), "\n")

```

Estadístico W: 0.9966

```
cat("Valor p:", shapiro_test$p.value, "\n")
```

```
## Valor p: 0.0002190907
```

```
# Interpretación
if (shapiro_test$p.value > 0.05) {
  cat("Conclusión: Los datos SON NORMALES (p > 0.05)\n\n")
} else {
  cat("Conclusión: Los datos NO SON NORMALES (p <= 0.05)\n\n")
}
```

```
## Conclusión: Los datos NO SON NORMALES (p <= 0.05)
```

```
# Prueba de Kolmogorov-Smirnov
```

```
# Estandarizar datos
datos_std <- (datos$Nota_Final - mean(datos$Nota_Final)) / sd(datos$Nota_Final)

ks_test <- ks.test(datos_std, "pnorm")
```

```
## Warning in ks.test.default(datos_std, "pnorm"): ties should not be present for
## the one-sample Kolmogorov-Smirnov test
```

```
cat("=== PRUEBA DE KOLMOGOROV-SMIRNOV ===\n")
```

```
## === PRUEBA DE KOLMOGOROV-SMIRNOV ===
```

```
cat("Estadístico D:", round(ks_test$statistic, 4), "\n")
```

```
## Estadístico D: 0.029
```

```
cat("Valor p:", ks_test$p.value, "\n")
```

```
## Valor p: 0.06953239
```

```
# Interpretación
if (ks_test$p.value > 0.05) {
  cat("Conclusión: Los datos SON NORMALES (p > 0.05)\n")
} else {
  cat("Conclusión: Los datos NO SON NORMALES (p <= 0.05)\n")
}
```

```
## Conclusión: Los datos SON NORMALES (p > 0.05)
```

```
#9 Visualizaciones Avanzadas ## 9.1 Mapa de Calor (Heatmap) de Correlaciones
```

```
## Seleccionar solo las variables numéricas (sin ID)
```

```
vars_numericas <- datos[, c(
  "Asistencia",
  "Examen_Interno_1",
  "Examen_Interno_2",
  "Promedio_Tareas",
  "Horas_Estudio",
  "Nota_Final"
)]
```

```
# Calcular la matriz de correlación
```

```
matriz_cor <- cor(vars_numericas)
```

```
# Mostrar la matriz redondeada
```

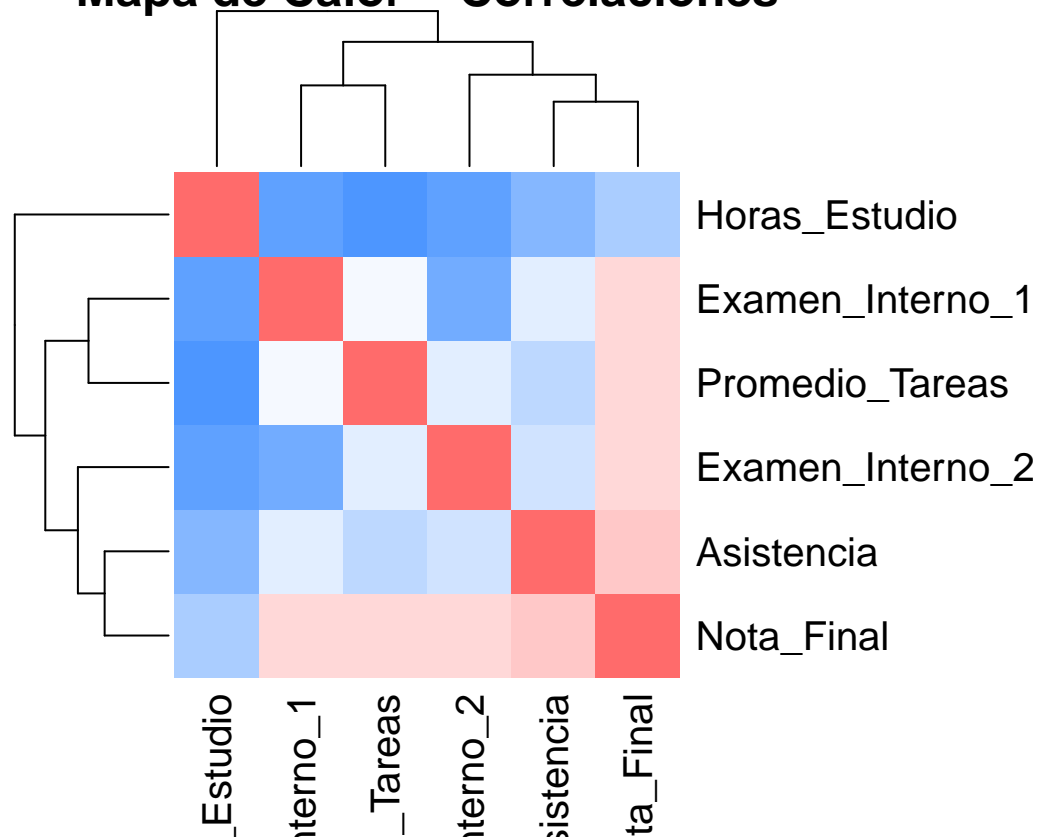
```
round(matriz_cor, 2)
```

```
##           Asistencia Examen_Interno_1 Examen_Interno_2 Promedio_Tareas
## Asistencia           1.00           0.51           0.49           0.43
## Examen_Interno_1      0.51           1.00           0.26           0.57
## Examen_Interno_2      0.49           0.26           1.00           0.54
## Promedio_Tareas       0.43           0.57           0.54           1.00
## Horas_Estudio         0.30           0.21           0.23           0.17
## Nota_Final            0.73           0.69           0.69           0.67
##           Horas_Estudio Nota_Final
## Asistencia           0.30      0.73
## Examen_Interno_1      0.21      0.69
## Examen_Interno_2      0.23      0.69
## Promedio_Tareas       0.17      0.67
## Horas_Estudio         1.00      0.41
## Nota_Final            0.41      1.00
```

```
# Crear mapa de calor básico
```

```
heatmap(
  matriz_cor,
  col = colorRampPalette(c("#4D96FF", "white", "#FF6B6B"))(20),
  symm = TRUE,
  main = "Mapa de Calor - Correlaciones"
)
```

Mapa de Calor – Correlaciones



```
# Cargar reshape2
library(reshape2)

# Convertir matriz a formato largo
matriz_largo <- melt(matriz_cor)
colnames(matriz_largo) <- c("Variable1", "Variable2", "Correlacion")

# Crear mapa de calor con ggplot2
ggplot(matriz_largo, aes(x = Variable1, y = Variable2, fill = Correlacion)) +
  geom_tile(color = "white") +
  geom_text(aes(label = round(Correlacion, 2)), size = 3) +
  scale_fill_gradient2(
    low = "#4D96FF",
    mid = "white",
    high = "#FF6B6B",
    midpoint = 0,
    limit = c(-1, 1)
  ) +
  labs(
    title = "Mapa de Calor - Matriz de Correlaciones",
    x = "",
    y = ""
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

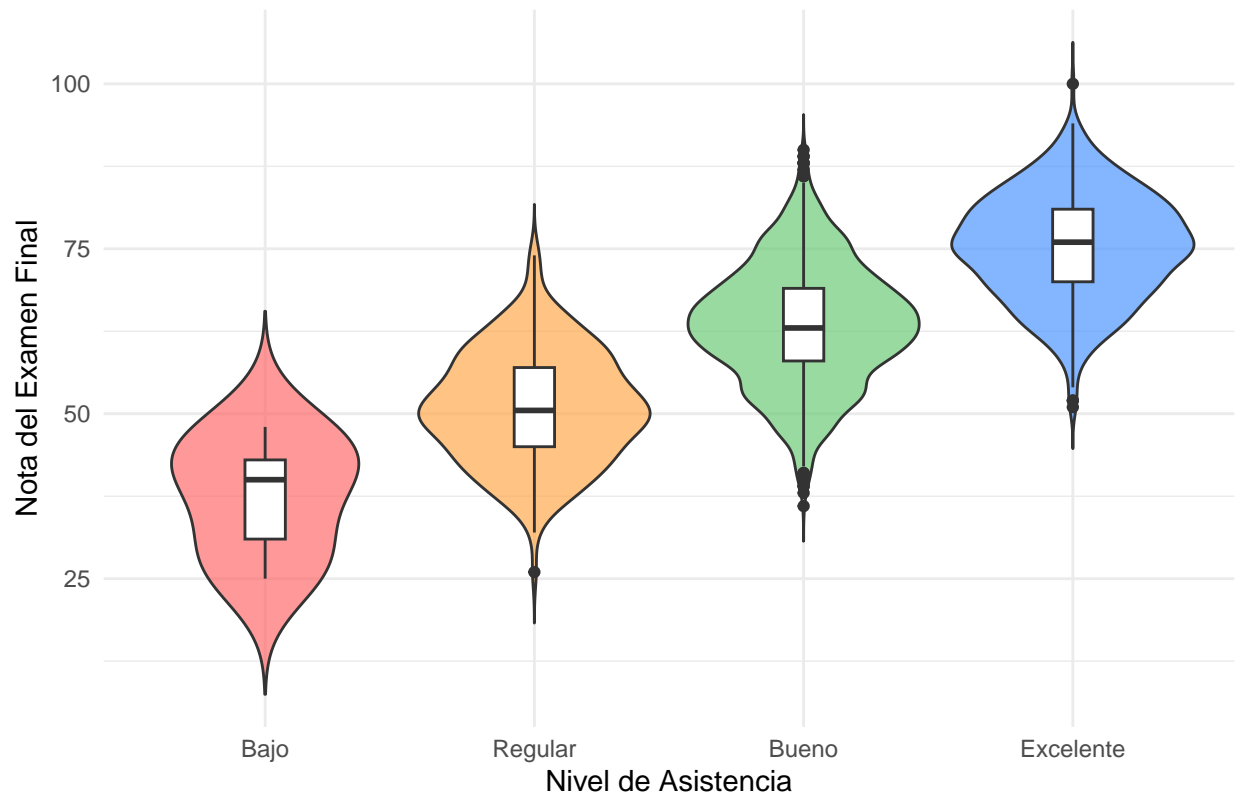
Mapa de Calor – Matriz de Correlaciones



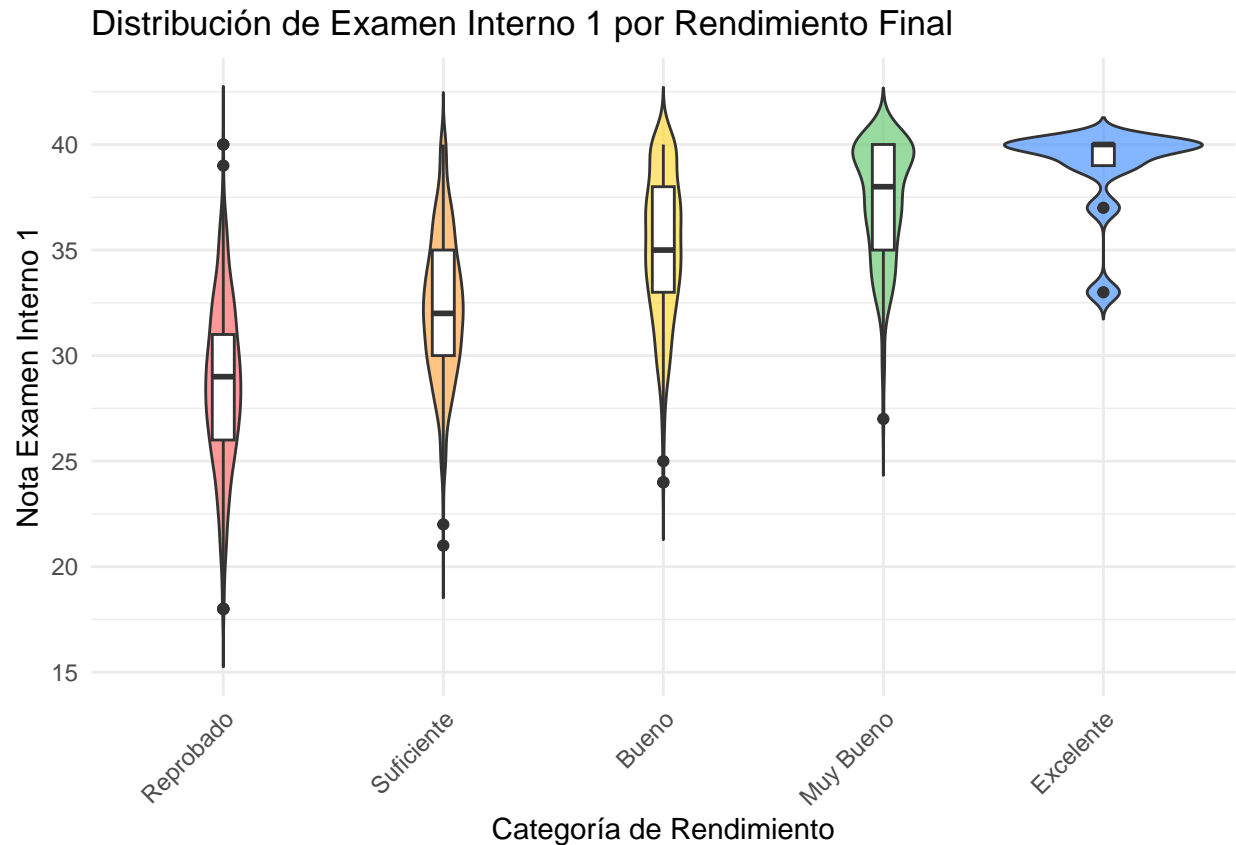
9.2 Gráfico de Violín

```
# Gráfico de violín: Nota Final por Nivel de Asistencia
ggplot(datos, aes(x = Nivel_Asistencia, y = Nota_Final, fill = Nivel_Asistencia)) +
  geom_violin(trim = FALSE, alpha = 0.7) +
  geom_boxplot(width = 0.15, fill = "white") +
  scale_fill_manual(values = c("#FF6B6B", "#FFA94D", "#6BCB77", "#4D96FF")) +
  labs(
    title = "Distribución de Nota Final por Nivel de Asistencia",
    x = "Nivel de Asistencia",
    y = "Nota del Examen Final"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```

Distribución de Nota Final por Nivel de Asistencia



```
# Gráfico de violín por categoría de rendimiento
ggplot(datos, aes(x = Rendimiento, y = Examen_Interno_1, fill = Rendimiento)) +
  geom_violin(trim = FALSE, alpha = 0.7) +
  geom_boxplot(width = 0.1, fill = "white") +
  scale_fill_manual(values = c("#FF6B6B", "#FFA94D", "#FFD93D", "#6BCB77", "#4D96FF")) +
  labs(
    title = "Distribución de Examen Interno 1 por Rendimiento Final",
    x = "Categoría de Rendimiento",
    y = "Nota Examen Interno 1"
  ) +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



10 Análisis de Correlación

10.1 Mapa de Calor (Heatmap) de Correlaciones

```
# Seleccionar solo variables numéricas
vars_numericas <- datos[, c(
  "Asistencia",
  "Examen_Interno_1",
  "Examen_Interno_2",
  "Promedio_Tareas",
  "Horas_Estudio",
  "Nota_Final"
)]
```

```
# Calcular matriz de correlación
matriz_cor <- cor(vars_numericas)
```

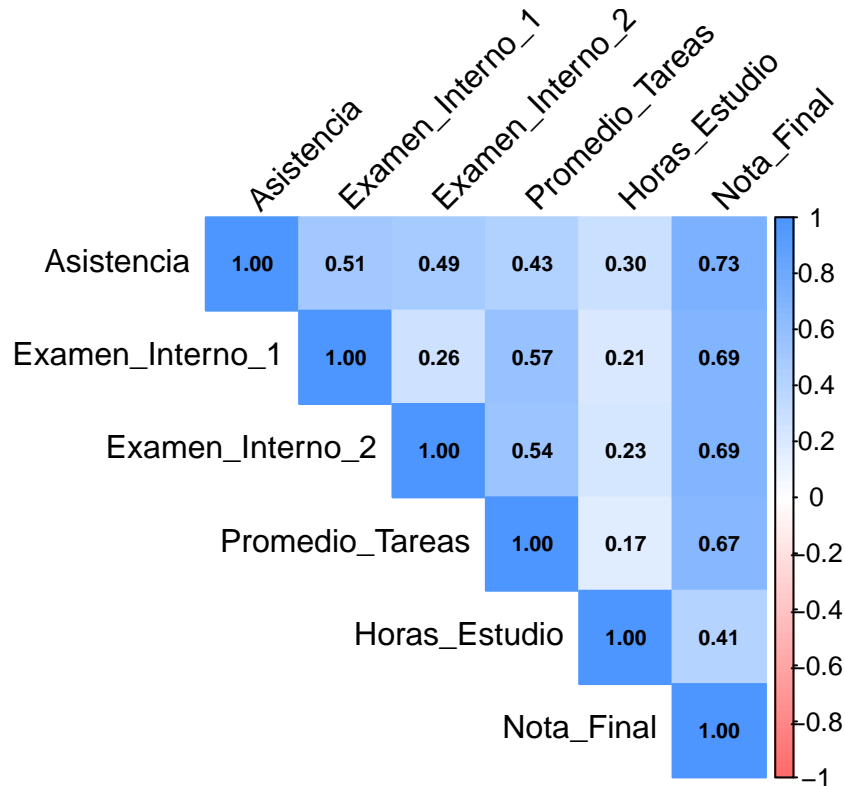
```
# Mostrar matriz redondeada
round(matriz_cor, 2)
```

```
##           Asistencia Examen_Interno_1 Examen_Interno_2 Promedio_Tareas
## Asistencia           1.00           0.51           0.49           0.43
## Examen_Interno_1      0.51           1.00           0.26           0.57
```

## Examen_Interno_2	0.49	0.26	1.00	0.54
## Promedio_Tareas	0.43	0.57	0.54	1.00
## Horas_Estudio	0.30	0.21	0.23	0.17
## Nota_Final	0.73	0.69	0.69	0.67
##	Horas_Estudio	Nota_Final		
## Asistencia	0.30	0.73		
## Examen_Interno_1	0.21	0.69		
## Examen_Interno_2	0.23	0.69		
## Promedio_Tareas	0.17	0.67		
## Horas_Estudio	1.00	0.41		
## Nota_Final	0.41	1.00		

```
# =====
# Mapa de calor con corrplot
# =====
corrplot(
  matriz_cor,
  method = "color",
  type = "upper",
  addCoef.col = "black",
  number.cex = 0.7,
  tl.col = "black",
  tl.srt = 45,
  col = colorRampPalette(c("#FF6B6B", "white", "#4D96FF"))(100),
  title = "Mapa de Calor - Correlaciones",
  mar = c(0, 0, 2, 0)
)
```

Mapa de Calor – Correlaciones

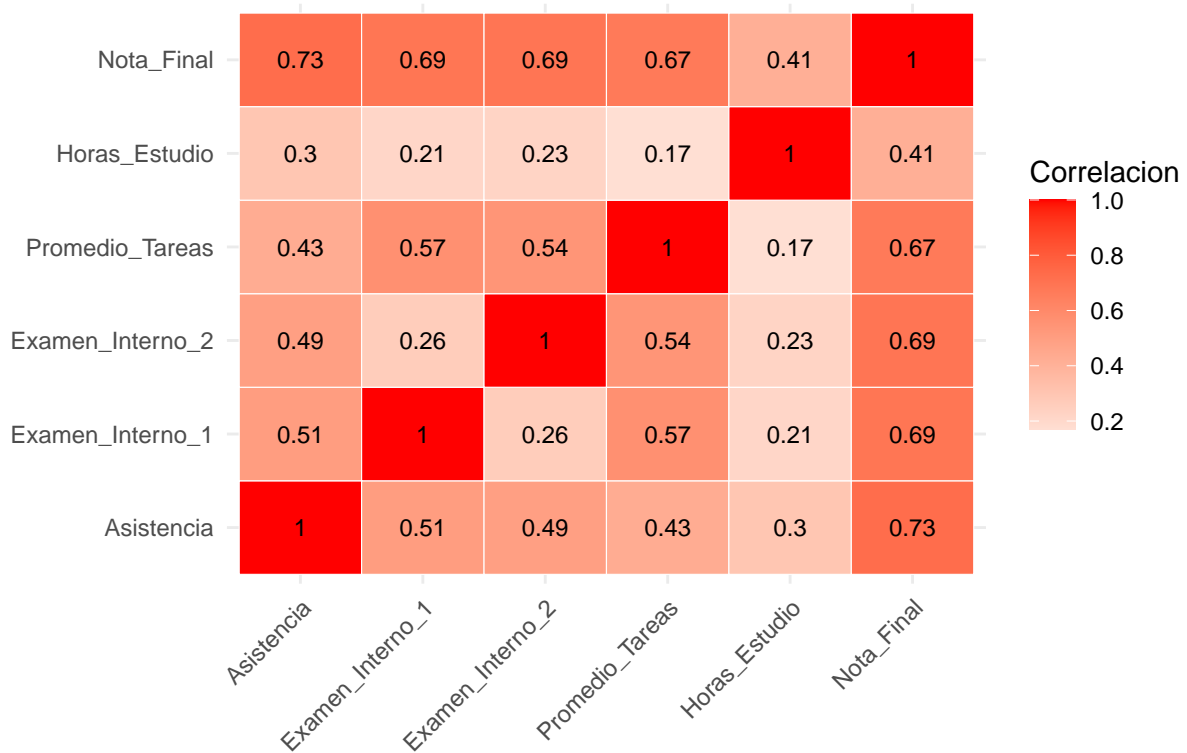


```
# =====
# Mapa de calor con ggplot2
# =====

# Convertir la matriz a formato largo
matriz_largo <- melt(matriz_cor)
colnames(matriz_largo) <- c("Variable1", "Variable2", "Correlacion")

# Crear heatmap con ggplot2
ggplot(matriz_largo, aes(x = Variable1, y = Variable2, fill = Correlacion)) +
  geom_tile(color = "white") +
  geom_text(aes(label = round(Correlacion, 2)), size = 3) +
  scale_fill_gradient2(
    low = "blue",
    mid = "white",
    high = "red",
    midpoint = 0
  ) +
  labs(title = "Matriz de Correlación", x = "", y = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Matriz de Correlación

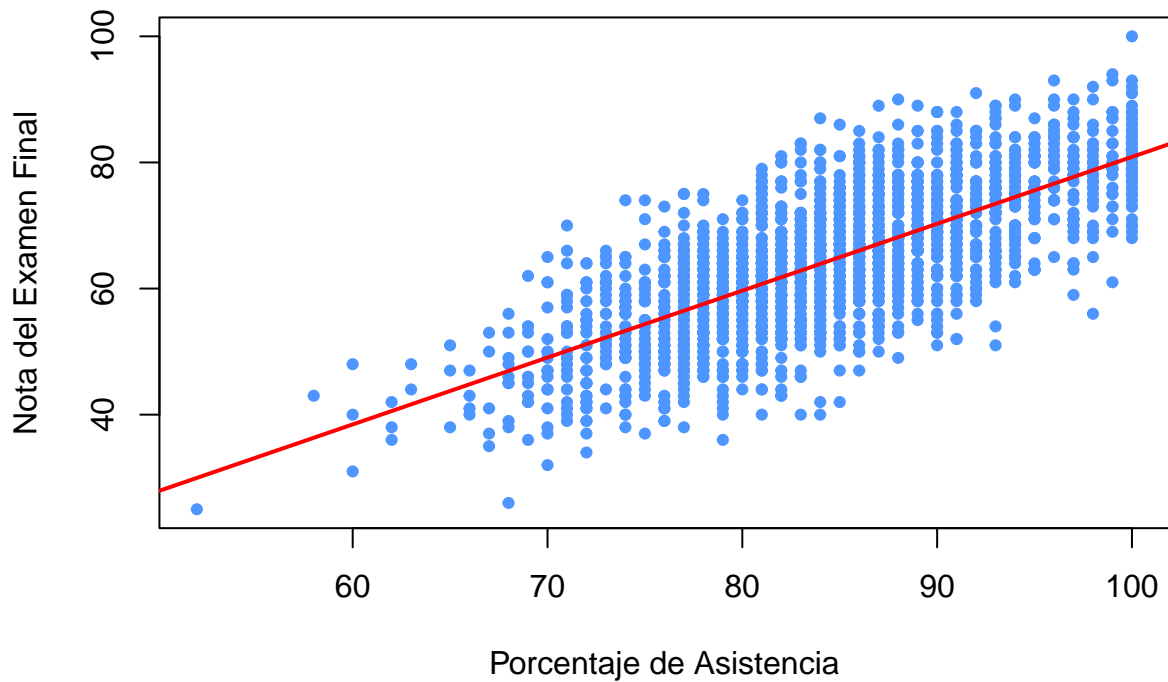


10.2 Gráfico de Dispersión: Asistencia vs Nota Final

```
# Crear gráfico de dispersión
plot(
  datos$Asistencia, datos$Nota_Final,
  main = "Relación entre Asistencia y Nota Final",
  xlab = "Porcentaje de Asistencia",
  ylab = "Nota del Examen Final",
  col = "#4D96FF",
  pch = 19,
  cex = 0.7
)

# Agregar línea de tendencia
abline(
  lm(Nota_Final ~ Asistencia, data = datos),
  col = "red",
  lwd = 2
)
```

Relación entre Asistencia y Nota Final



```
## 10.3 Cálculo del Coeficiente de Correlación
```

```
# Calcular correlación de Pearson
correlacion <- cor(datos$Asistencia, datos$Nota_Final)

cat("Coeficiente de correlación (r):", round(correlacion, 4), "\n")
```

```
## Coeficiente de correlación (r): 0.7256
```

```
# Prueba de significancia
prueba_cor <- cor.test(datos$Asistencia, datos$Nota_Final)

cat("Valor p:", prueba_cor$p.value, "\n")
```

```
## Valor p: 0
```

```
# Interpretación
if (prueba_cor$p.value < 0.05) {
  cat("La correlación ES estadísticamente significativa\n")
} else {
  cat("La correlación NO es significativa\n")
}
```

```
## La correlación ES estadísticamente significativa
```

11 Regresión Lineal Simple

11.1 Crear el Modelo de Regresión

```
# Crear el modelo de regresión
modelo <- lm(Nota_Final ~ Asistencia, data = datos)

# Ver resumen del modelo
summary(modelo)

##
## Call:
## lm(formula = Nota_Final ~ Asistencia, data = datos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.9094  -5.4021   0.1196   5.4230  23.0906
##
## Coefficients:
##              Estimate Std. Error t value      Pr(>|t|)
## (Intercept) -25.1883     1.9181  -13.13 <0.0000000000000002 ***
## Asistencia   1.0607     0.0225   47.14 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.806 on 1998 degrees of freedom
## Multiple R-squared:  0.5266, Adjusted R-squared:  0.5263
## F-statistic: 2222 on 1 and 1998 DF,  p-value: < 0.00000000000000022
```

11.2 Interpretar los Resultados

```
# Obtener coeficientes
intercepto <- coef(modelo)[1]
pendiente <- coef(modelo)[2]

cat("=== ECUACIÓN DEL MODELO ===\n")

## === ECUACIÓN DEL MODELO ===

cat(
  "Nota_Final =",
  round(intercepto, 2), "+",
  round(pendiente, 2), "* Asistencia\n\n"
)

## Nota_Final = -25.19 + 1.06 * Asistencia
```

```
# Coeficiente de determinación R2
```

```
r_cuadrado <- summary(modelo)$r.squared
```

```
cat("R cuadrado (R2):", round(r_cuadrado, 4), "\n")
```

```
## R cuadrado (R2): 0.5266
```

```
cat(  
  "Esto significa que la Asistencia explica el",  
  round(r_cuadrado * 100, 1),  
  "% de la variabilidad en la Nota Final\n"  
)
```

```
## Esto significa que la Asistencia explica el 52.7 % de la variabilidad en la Nota Final
```

11.3 Visualizar el Modelo de Regresión

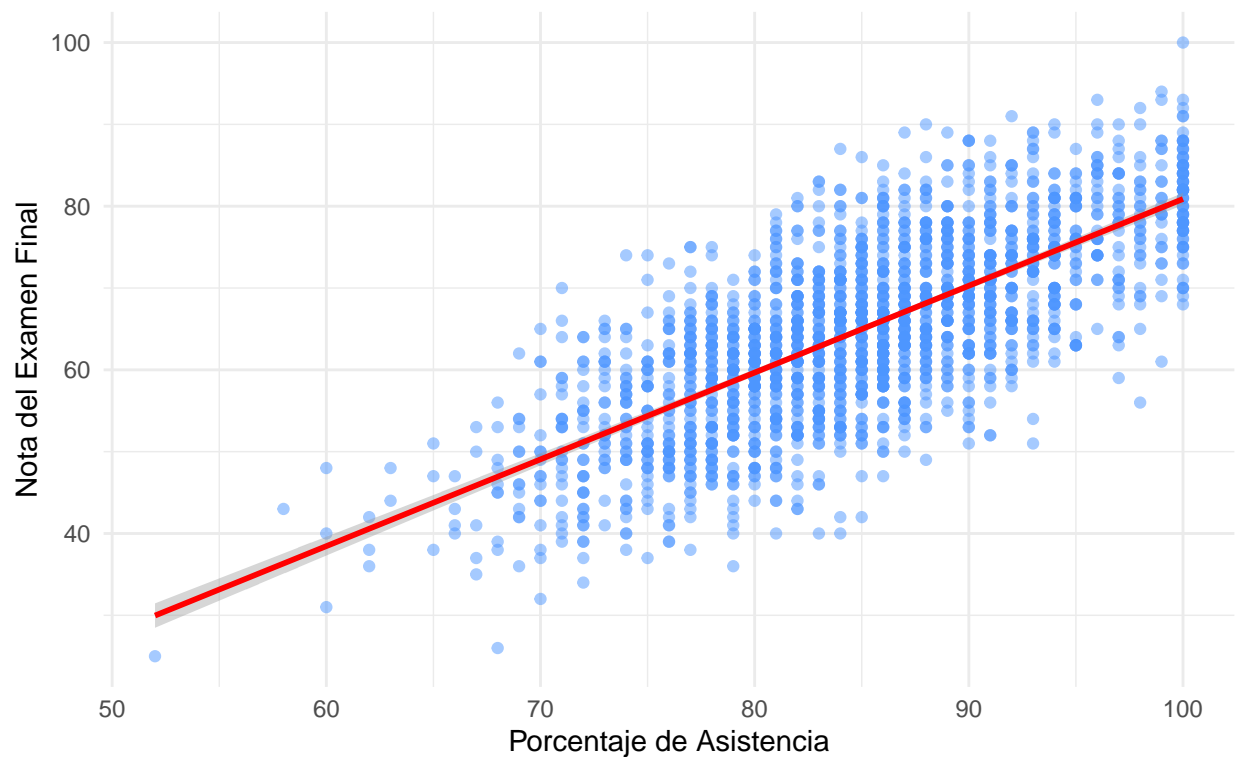
```
# Gráfico con ggplot2
```

```
ggplot(datos, aes(x = Asistencia, y = Nota_Final)) +  
  geom_point(color = "#4D96FF", alpha = 0.5) +  
  geom_smooth(method = "lm", color = "red", se = TRUE) +  
  labs(  
    title = "Regresión Lineal: Nota Final vs Asistencia",  
    subtitle = paste("R2 =", round(r_cuadrado, 4)),  
    x = "Porcentaje de Asistencia",  
    y = "Nota del Examen Final"  
  ) +  
  theme_minimal()
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

Regresión Lineal: Nota Final vs Asistencia

$R^2 = 0.5266$



11.4 Realizar Predicciones

```
# Crear datos para predicción
nuevos_datos <- data.frame(
  Asistencia = c(60, 70, 80, 90, 100)
)

# Realizar predicciones
predicciones <- predict(modelo, newdata = nuevos_datos)

# Mostrar resultados
cat("=== PREDICCIONES ===\n")
```

```
## === PREDICCIONES ===
```

```
for (i in 1:5) {
  cat(
    "Si la asistencia es", nuevos_datos$Asistencia[i],
    "% -> Nota predicha:",
    round(predicciones[i], 1), "\n"
  )
}
```

```
## Si la asistencia es 60 % -> Nota predicha: 38.5
## Si la asistencia es 70 % -> Nota predicha: 49.1
## Si la asistencia es 80 % -> Nota predicha: 59.7
## Si la asistencia es 90 % -> Nota predicha: 70.3
## Si la asistencia es 100 % -> Nota predicha: 80.9
```