Análisis de datos de empleados de archivo CSV

En esta tarea, trabajará con un archivo CSV que contiene información sobre los empleados

Contenido del archivo: El archivo CSV contiene datos sobre los empleados, incluidas sus edades, salarios y fechas de contratación.

Problemas de calidad: El archivo tiene varios problemas que afectan a su calidad y dificultan su análisis.

Resumen de la tarea

Su tarea es abordar los problemas de calidad en el archivo CSV proporcionado y preparar los datos para el análisis. A continuación, te explicamos cómo puedes abordar la tarea:

Pasos para completar la tarea

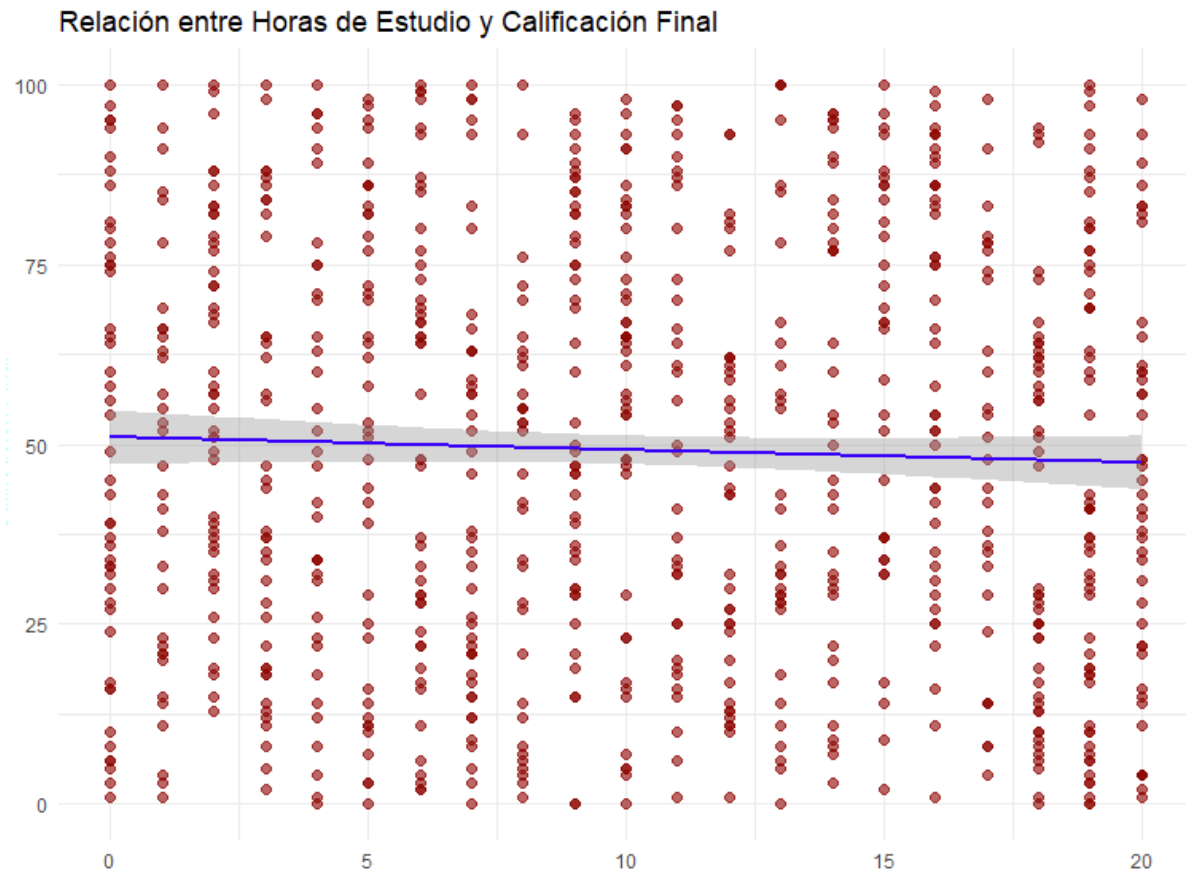
Identifique los problemas de calidad: Comience por examinar cuidadosamente el archivo CSV para identificar los problemas específicos que afectan su calidad. Busque inconsistencias, datos faltantes o cualquier otra anomalía que pueda afectar el análisis.

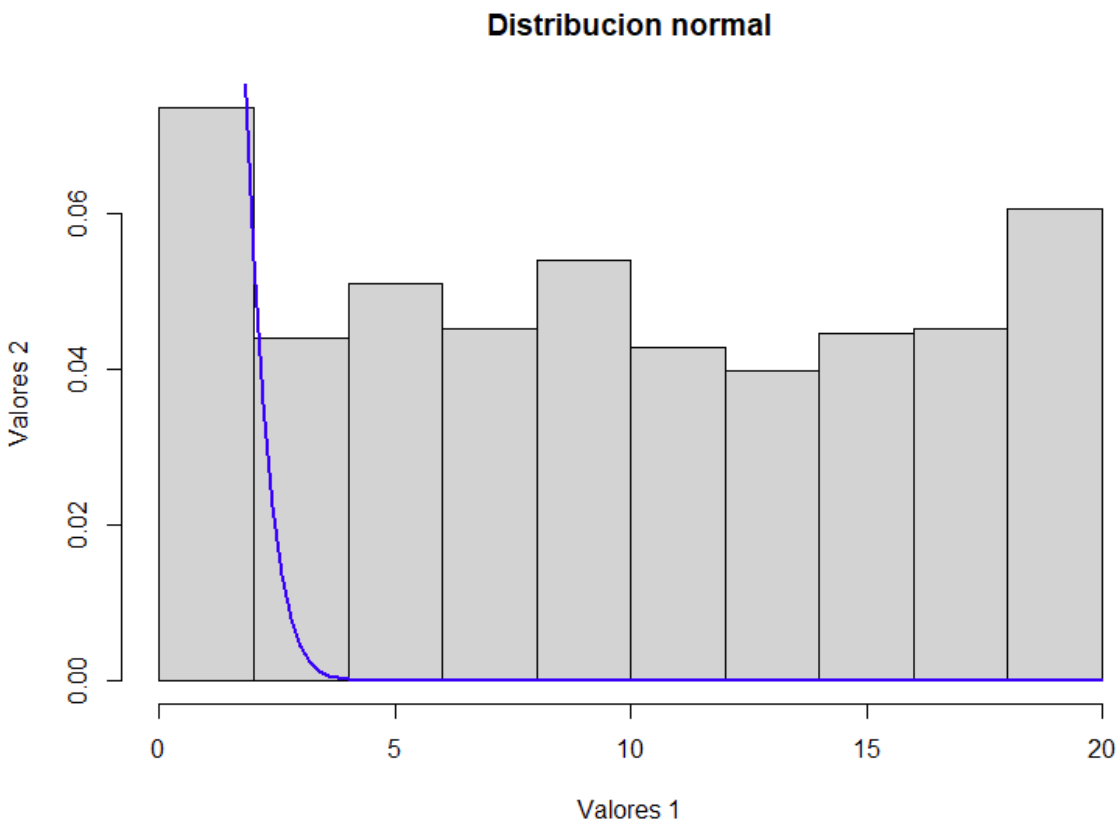
Limpieza de datos: Una vez que haya identificado los problemas, proceda a limpiar los datos. Esto puede implicar abordar los valores faltantes, corregir incoherencias y asegurarse de que los datos tengan el formato correcto para el análisis.

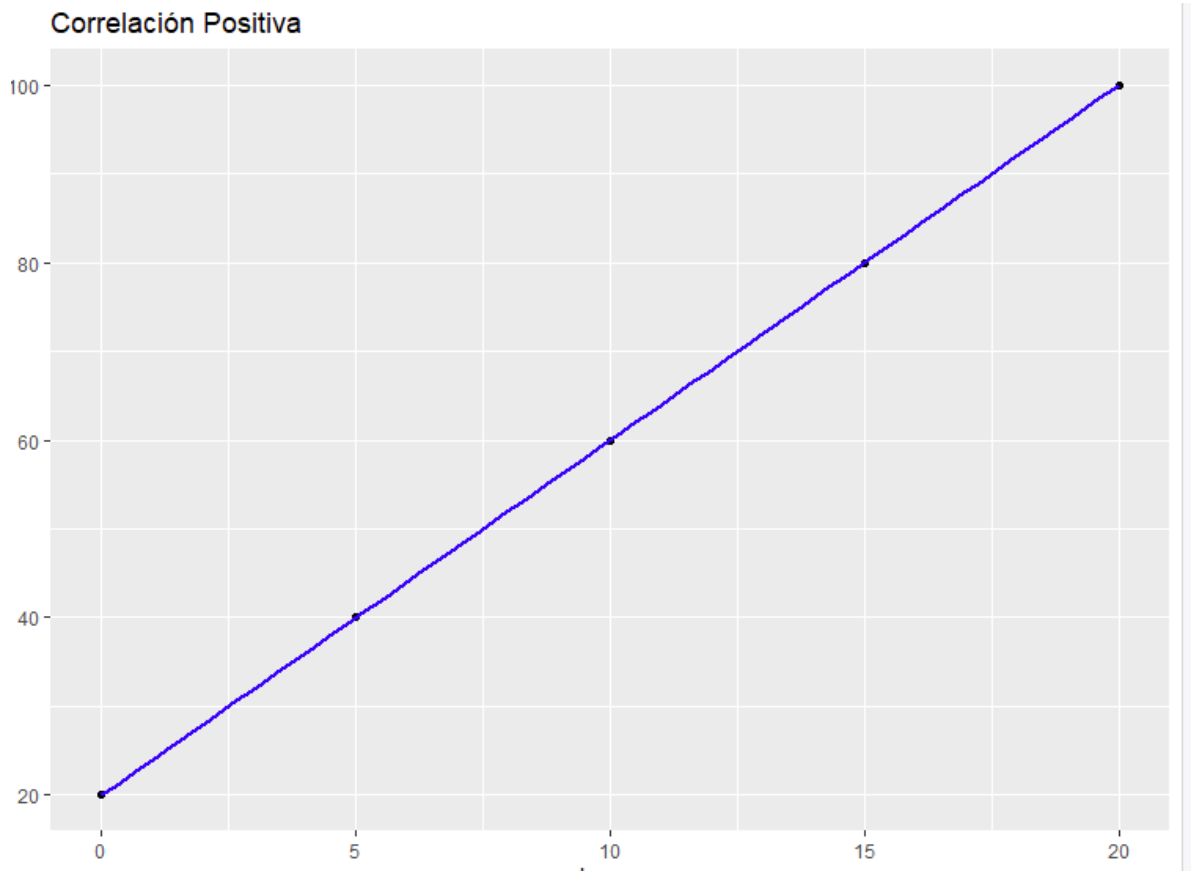
Transformación de datos: Después de limpiar los datos, considere si se necesitan transformaciones adicionales para que los datos sean más adecuados para el análisis. Esto podría incluir la creación de nuevas columnas, la agregación de datos o el cambio de formato de ciertos campos.

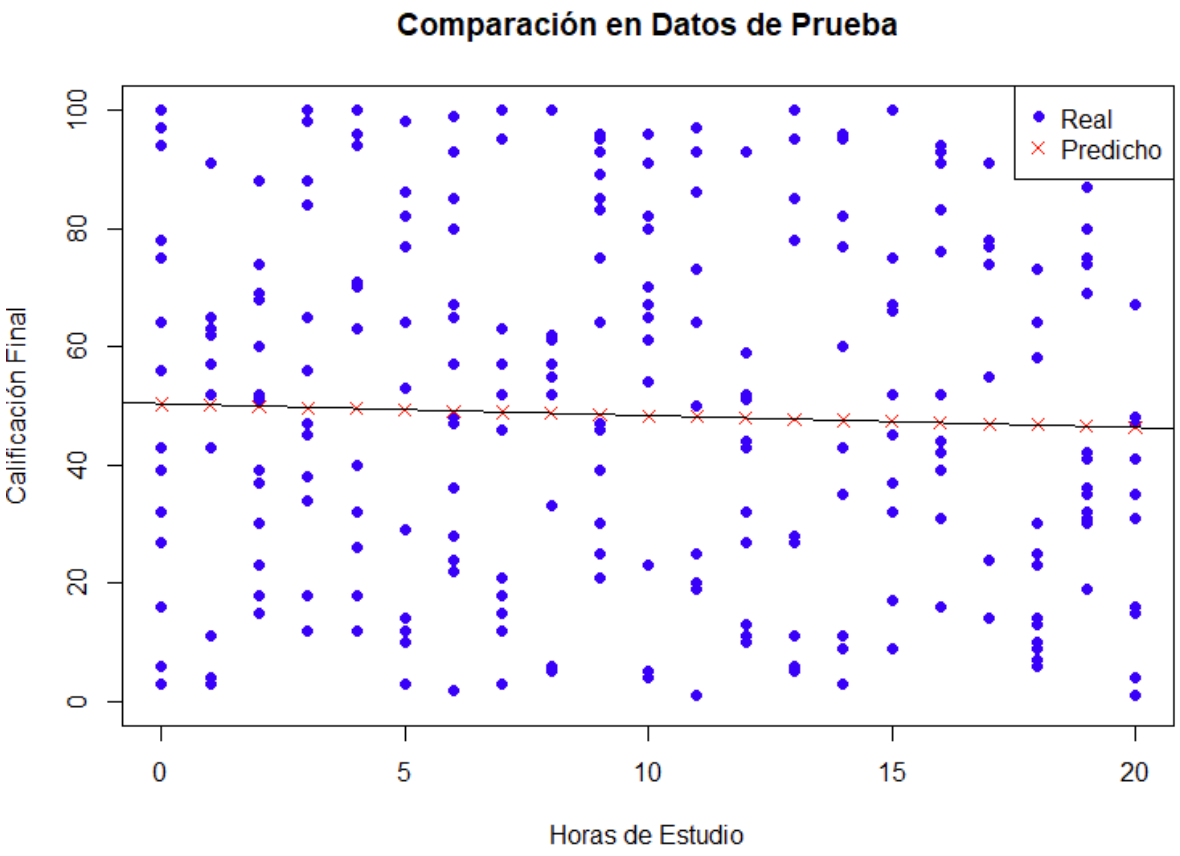
Documentación: A medida que trabajas en el proceso de limpieza y transformación de datos, asegúrate de documentar los cambios que realizas y las razones detrás de ellos. Esta documentación será valiosa para comprender los pasos dados y justificar las decisiones tomadas.

Preparación del análisis de datos: Por último, prepare los datos limpios y transformados para el análisis. Esto puede implicar exportar los datos a un nuevo archivo CSV o prepararlos para un análisis posterior utilizando herramientas como Python, R o Excel.


Relación entre Horas de Estudio y Calificación Final
100
75
50
25
0
0
5
10
15
20


Distribucion normal
Valores 2
0.00
0.02
0.04
0.06
0
5
10
15
20
Valores 1


Correlación Positiva
100
80
60
40
20
0
5
10
15
20




```r
data = read.csv('C:\\Users\\Analista de Datos\\Desktop\\Daniel del Valle\\Rproyect\\Nueva carpeta\\Datos.csv')

#Analizar estructura
head(data)
summary(data)
str(data)
data$Horas.de.estudio[is.na(data$Horas.de.estudio)] <- mean(data$Horas.de.estudio, na.rm = TRUE)

# visualizar la relación con un gráfico de dispersión
library(ggplot2)
ggplot(data, aes(x = Horas.de.estudio, y = Calificacion.final)) +
  geom_point(alpha = 0.6, size = 2, color = "darkred") +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Relación entre Horas de Estudio y Calificación Final",
       x = "Horas de Estudio",
       y = "Calificación Final") +
  theme_minimal()  # Tema más limpio y simple

#crear un histograma para visualizar la distribucion
hist(data$Horas.de.estudio, probability = TRUE, main = "Distribucion normal",xlab="Valores 1",ylab = "Valores 2")

#añadir la curva de densidad normal
curve(dnorm(x,mean=0,sd=1),add=TRUE,col="blue",lwd=2)
```

```r
# Calcular el coeficiente de correlación de Pearson
correlation = cor(data$Horas.de.estudio, data$Calificacion.final)
print(paste("Coeficiente de correlación de Pearson:", correlation))
```

```r
#correlacion positiva
# Correlación positiva (altura y peso)
horas = c(0, 5, 10, 15, 20)
calificacion = c(20, 40, 60, 80, 100)

# Calcular la correlación
cor(calificacion, horas)

# Gráfico de dispersión
ggplot(data.frame(horas, calificacion), aes(x = horas, y = calificacion )) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Correlación Positiva")
```

```r
# Gráfico comparativo en conjunto de prueba
plot(datos_prueba$Horas.de.estudio, datos_prueba$Calificacion.final,
     main = "Comparación en Datos de Prueba",
     xlab = "Horas de Estudio", ylab = "Calificación Final", pch = 19, col = "blue")
points(datos_prueba$Horas.de.estudio, predicciones_prueba, col = "red", pch = 4)
abline(modelo_entrenamiento, col = "black")
legend("topright", legend = c("Real", "Predicho"), col = c("blue", "red"), pch = c(19, 4))
```