Nombre del alumno: López Jaimes Edgar Felipe
Matricula:202118392
Profesor. Ebner Juárez Elías
Cuestionario Escrito 1er parcial. **Calificación:**
Materia Análisis y Modelado de Datos
Valor total 30%

Instrucciones: contesta correctamente subrayando la respuesta correcta. Debes de entregar escrito a mano correctamente los códigos utilizados, así como compartir en GitHub un repositorio "cuestionario1_nombrealumno" al usuario profebner.

Problema 1: Una empresa de retail ha recopilado datos de ventas de múltiples sucursales, pero presenta valores faltantes, datos duplicados y errores tipográficos. El equipo de análisis de datos necesita limpiar el dataset antes de realizar análisis.

Tareas:

1. Cargar un dataset en R
2. Identificar y manejar valores faltantes
3. Detectar y eliminar valores duplicados
4. Estandarizar formatos de nombres de productos

Codigo

```
# 1. Cargar un dataset

data <- read.csv("data.csv")

# 2. Identificar y manejar valores faltantes
# Identificar valores faltantes
print(summary(data)) # Muestra un resumen de las columnas, incluyendo cuántos NA hay en cada una
print(is.na(data))  # Devuelve una matriz booleana indicando dónde están los NA

# Manejar valores faltantes media
for (col in colnames(data)) {
  if (any(is.na(data[[col]]))) {
    if (is.numeric(data[[col]])) {
      media_col <- mean(data[[col]], na.rm = TRUE)
      data[[col]] <- ifelse(is.na(data[[col]]), media_col, data[[col]])
    } else {
      # Si no es numérico, imputar con la moda
      tabla_freq <- table(data[[col]])
      moda_col <- names(tabla_freq[which.max(tabla_freq)])
      data[[col]] <- ifelse(is.na(data[[col]]), moda_col, data[[col]])
    }
  }
}
```

```
# 3. Detectar y eliminar valores duplicados
print(duplicated(data)) # Devuelve un vector booleano indicando si cada fila es duplicada

# Eliminar valores duplicados
data_sin_duplicados <- distinct(data) # Usando la función distinct() del paquete dplyr

# 4. Estandarizar formatos de nombres de productos
# Estandarizar nombres convertir a minúsculas y eliminar espacios extra)
if ("nombre_producto" %in% colnames(data)) {
  data$nombre_producto <- tolower(trimws(data$nombre_producto))
  # Aquí puedes agregar más estandarizaciones según sea necesario
}
```

Cuestionario de Evaluación

1. ¿Qué función se usa para eliminar valores duplicados en un dataframe en R?

   a) remove_duplicates()
   b) distinct()
   c) filter_duplicates()

2. ¿Cuál es la mejor manera de tratar valores faltantes en una columna numérica?

   a) Eliminarlos directamente siempre
   b) Imputarlos con la media o mediana
   c) Dejar los valores faltantes sin cambios

3. ¿Qué paquete de R facilita la manipulación de datos de manera eficiente?

   a) ggplot2
   b) tidyverse

Problema 2 : Un equipo de marketing necesita analizar datos de interacción en redes sociales, pero los datos están en diferentes formatos y escalas, lo que dificulta el análisis.

Tareas:

Convertir variables categóricas en factores

1. Normalizar valores numéricos
2. Crear nuevas variables derivadas
3. Convertir fechas en formato adecuado

```
1. Convertir variables categóricas en factores
red<-read.csv("C:/Users/DANIELAGUADALUPEAGUI/OneDrive - TECNOLOGICO DE ESTUDIOS SUPERIORES DE IXTAPALUCA/Documentos/TESI/OCTAVO SEMESTRE/ANALISIS Y MODELADO DE DATOS/equipo/redesociales.csv")
str(red)
#aqui pongo codigo para conversion de variables categoricas en factores
red$tipoInteraccion <- as.factor(red$tipoInteraccion)
red$plataforma <- as.factor(red$plataforma)
red$nombreUsuario <- as.factor(red$nombreUsuario)
red$contenido <- as.factor(red$contenido)
red$fecha <- as.Date(red$fecha, format="%d/%m/%Y")
red <- red[, !names(red) %in% "Unnamed: 7"]
str(red)
2. Normalizar valores numéricos
normalizar_minmax <- function(x) {
return((x - min(x, na.rm=TRUE)) / (max(x, na.rm=TRUE) - min(x, na.rm=TRUE)))
}
red$numeroInteracciones <- normalizar_minmax(red$numeroInteracciones)
summary(red$numeroInteracciones)
3. Crear nuevas variables derivadas
red$diaSemana <- weekdays(red$fecha)
red$mes <- format(red$fecha, "%m")
red$anio <- format(red$fecha, "%Y")
red$diasDesdePrimera <- as.numeric(red$fecha - min(red$fecha, na.rm=TRUE))
```

```
if (sum(!is.na(red$numeroInteracciones)) > 0) {
red$numeroInteraccionesNorm <-
normalizar_minmax(red$numeroInteracciones)
}
if (sum(!is.na(red$numeroInteracciones)) > 0) {
red$nivelInteraccion <- ifelse(
red$numeroInteracciones > median(red$numeroInteracciones, na.rm=TRUE),
"Alta", "Baja"
)
red$nivelInteraccion <- as.factor(red$nivelInteraccion)
}
red$longitudContenido <- ifelse(is.na(red$contenido), NA,
nchar(as.character(red$contenido)))
red$palabrasContenido <- ifelse(is.na(red$contenido), NA,
sapply(strsplit(as.character(red$contenido), " "), length))
red <- red %>%
group_by(nombreUsuario) %>%
mutate(frecuenciaInteraccion = sum(!is.na(nombreUsuario))) %>%
ungroup()
4. Convertir fechas en formato adecuado
if (!inherits(red$fecha, "Date")) {
red$fecha <- as.Date(red$fecha, tryFormats = c("%d/%m/%Y", "%Y-%m-%d",
"%
```

Cuestionario de Evaluación

1. ¿Qué función se usa para normalizar datos en R?

   a) normalize()
   b) scale()
   c) rescale()

2. ¿Cuál es la ventaja de convertir variables categóricas en factores en R?

   a) Permite realizar operaciones matemáticas en ellas
   b) Mejora la eficiencia en el procesamiento y análisis
   c) Hace que el dataset ocupe más memoria

3. ¿Qué función permite transformar una columna de texto en una fecha en R?

   a) to_date()
   b) as.Date()
   c) convert_date()

Problema 3: Un analista de datos necesita fusionar dos datasets: uno con información de clientes y otro con sus compras. Es necesario unirlos de manera eficiente.

Tareas:

1. Cargar y explorar los dos datasets en R.
2. Unir los datasets
3. Verificar si hay claves duplicadas o valores faltantes después de la fusión.
4. Realizar una consulta de resumen para verificar la correcta integración.

Código ejemplo:

```
# Cargar las librerías necesarias
library(dplyr)
```

1. Cargar y explorar los datasets

```
clientes <- read.csv("clientes.csv")  # Cargar dataset de clientes
compras <- read.csv("compras.csv")    # Cargar dataset de compras
```

2. Unir los datasets

```
fusionado <- merge(clientes, compras, by = "id_cliente", all = TRUE)
```

3. Verificar claves duplicadas

```
duplicados <- fusionado %>% group_by(id_cliente) %>% filter(n() > 1)
print(duplicados)
# Verificar valores faltantes
faltantes <- colSums(is.na(fusionado))
print(faltantes)
```

4. Consulta de resumen

```
resumen <- fusionado %>%
  group_by(id_cliente) %>%
  summarise(total_compras = n(),
            monto_total = sum(monto, na.rm = TRUE))

print(resumen)
```

Cuestionario de Evaluación

1. ¿Cuál de las siguientes funciones se usa para unir dos datasets en R por una clave común?

    a) merge()
    b) left_join()
    c) concat()

2. ¿Qué función permite identificar si hay valores duplicados en una columna clave?

    a) table()
    b) duplicated()
    c) unique()

3. ¿Qué ocurre si se usa inner_join() en lugar de left_join()?

    a) Se eliminan las filas sin coincidencias en ambas tablas
    b) Se mantienen todas las filas de la tabla izquierda
    c) Se duplican los valores de la clave

Problema 4: Un equipo financiero está analizando transacciones, pero ha detectado valores extremadamente altos o bajos en los datos. Es necesario identificar y manejar los outliers.

Tareas:

1. Identificar outliers mediante diagramas de caja
2. Usar el rango intercuartil para determinar límites de outliers.
3. Manejar los valores atípicos mediante eliminación o transformación
4. Comparar estadísticas antes y después del tratamiento.

```
Instalar y cargar paquetes necesarios
install.packages("ggplot2")  # Instala ggplot2 si no lo tienes
library(ggplot2)  # Carga el paquete
Crear un conjunto de datos de prueba
# Generar datos de transacciones con valores atípicos
set.seed(123)  # Para reproducibilidad
transacciones <- data.frame(
  Monto = c(rnorm(50, mean = 1000, sd = 200),  # 50 valores normales
            5000, 5500, 6000,  # Valores atípicos altos
            200, 150)  # Valores atípicos bajos
)
Identificar outliers con un diagrama de caja
# Crear diagrama de caja
ggplot(transacciones, aes(y = Monto)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Diagrama de Caja de Transacciones",
       y = "Monto de Transacción") +
  theme_minimal()

Usar el Rango Intercuartil (IQR) para detectar outliers
# Calcular cuartiles
Q1 <- quantile(transacciones$Monto, 0.25)
Q3 <- quantile(transacciones$Monto, 0.75)
IQR <- Q3 - Q1  # Rango intercuartil

# Definir límites de outliers
limite_inferior <- Q1 - 1.5 * IQR
limite_superior <- Q3 + 1.5 * IQR

# Identificar outliers
outliers <- transacciones$Monto[transacciones$Monto < limite_inferior |
transacciones$Monto > limite_superior]
print(outliers)  # Muestra los valores atípicos detectados
Manejar outliers (eliminación o transformación)
Eliminar valores atípicos
transacciones_filtradas <- transacciones[transacciones$Monto >= limite_inferior &
transacciones$Monto <= limite_superior, ]
Transformación (Reemplazar con la mediana)
mediana <- median(transacciones$Monto)
transacciones$Monto[transacciones$Monto < limite_inferior | transacciones$Monto >
limite_superior] <- mediana
Comparar estadísticas antes y después
Antes:
```

summary(transacciones$Monto)
Después de eliminación o transformación:
summary(transacciones_filtradas$Monto) # Si eliminaste outliers

Cuestionario de Evaluación

1. ¿Cuál es una forma común de identificar outliers en un dataset?

    a) Usar un histograma
    b) Aplicar la técnica del rango intercuartil (IQR)
    c) Convertir los valores en ceros
2. ¿Qué gráfico es más adecuado para visualizar outliers?

    a) Diagrama de caja
    b) Gráfico de dispersión
    c) Gráfico de barras
3. ¿Cuál es una estrategia válida para manejar outliers en un dataset?

    a) Eliminarlos sin análisis previo
    b) Sustituirlos por la media o mediana
    c) Ignorarlos completamente

Problema 5: Se ha recopilado información de una encuesta con respuestas en formato de texto, pero se necesita transformar las variables categóricas en valores numéricos para análisis estadístico.

Tareas

1. Convertir variables cualitativas en numéricas
2. Aplicar codificación
3. Comparar cómo los modelos de machine learning reaccionan a diferentes codificaciones.

Código

```
# Aplicar codificación
# Codificación One-Hot (usando model.matrix)
encuesta$id <- 1:nrow(encuesta)  # Aseguramos tener un identificador único
formula_one_hot <- as.formula(paste("~",
paste(colnames(encuesta)[sapply(encuesta, is.factor)], collapse = "+"), "- 1"))
one_hot_encoded <- model.matrix(formula_one_hot, data = encuesta)
one_hot_encoded_df <- as.data.frame(one_hot_encoded)
encuesta_codificada_one_hot <- merge(encuesta, one_hot_encoded_df, by.x =
"id", by.y = "row.names")
encuesta_codificada_one_hot$id <- NULL

# Comparar cómo los modelos de machine learning reaccionan a diferentes
codificaciones.

# Crear un modelo con datos codificados Label Encoding
if (!is.null(encuesta_codificada_label$variable_numerica)){
  formula_label <- as.formula(paste("variable_numerica ~ ",
paste(colnames(encuesta_codificada_label)[sapply(encuesta_codificada_label,
is.numeric) & !names(encuesta_codificada_label) %in%
c("variable_numerica","id")], collapse = " + ")))
  modelo_label <- lm(formula_label, data = encuesta_codificada_label)
  print("Resumen del modelo con Label Encoding:")
  print(summary(modelo_label))
}
```

Cuestionario de Evaluación

1. ¿Por qué es importante codificar variables categóricas en modelos predictivos?

   a) Porque los modelos solo aceptan datos numéricos
   b) Porque mejora la visualización de datos
   c) No es importante codificarlas

2. ¿Qué técnica de codificación de variables categóricas crea múltiples columnas binarias?

   a) One-hot encoding
   b) Label encoding
   c) Scaling

3. ¿Qué función en R se usa para transformar variables categóricas en factores numéricos?

   a) factorize()
   b) as.factor()
   c) convert()

Problema 6: Un hospital ha recolectado datos de pacientes, pero algunas variables como presión arterial y nivel de glucosa tienen valores faltantes. El equipo de análisis necesita decidir cómo tratarlos antes de realizar estudios estadísticos.

Tareas

1. Cargar el dataset en R usando read.csv().
2. Identificar los valores faltantes con is.na() y summary().
3. Aplicar distintas estrategias para manejarlos: eliminación (na.omit()), imputación con la media (tidyverse::replace_na()), o interpolación.
4. Comparar los efectos de cada estrategia en el dataset final.

Problema 6

```
install.packages("tidyverse")
install.packages("VIM")
install.packages("naniar")

library(tidyverse)
library(VIM)
library(naniar)


file.choose()
hospital=read.csv("C:\\Users\\edgar\\OneDrive\\Escritorio\\Custionario\\LOPEZJAIME
SEDGARFELIPE\\hospital.csv")
View(hospital)

# Verificar cuántos valores faltantes hay en todo el dataset
sum(is.na(hospital))  # Total de valores faltantes en el dataset

# Verificar cuántos valores faltantes hay en cada columna
colSums(is.na(hospital))  # Cantidad de valores faltantes por columna

# Resumen estadístico del dataset, incluyendo los valores faltantes
summary(hospital)

# Eliminar registros con valores faltantes
hospital_limpio_naomit <- na.omit(hospital)

# Ver el resultado
View(hospital_limpio_naomit)


# Imputación con la media para las columnas de interés
hospital_imputado_media <- hospital %>%
  mutate(
    Presion_arterial = replace_na(Presion_arterial, mean(Presion_arterial, na.rm =
TRUE)),
    Glucosa = replace_na(Glucosa, mean(Glucosa, na.rm = TRUE))
  )
```

```
# Ver los datos después de imputar con la media
cat("Número de registros después de la imputación: ",
nrow(hospital_imputado_media), "\n")
View(hospital_imputado_media)

} else {
  cat("Las columnas 'Presion_arterial' y 'Glucosa' no están presentes en el dataset.\n")
}
```

Cuestionario de Evaluación

1. ¿Qué función en R permite identificar valores faltantes en un dataframe?

   a) missing_values()
   b) is.na()
   c) find_NA()

2. ¿Cuál es una estrategia válida para manejar valores faltantes en una columna numérica?

   a) Eliminarlos sin analizar su impacto
   b) Imputarlos con la media o la mediana
   c) Dejar los valores sin cambios y proceder con el análisis
   d)

3. ¿Cuál es una posible desventaja de eliminar todas las filas con valores faltantes?

   a) Puede reducir la cantidad de datos y afectar la representatividad
   b) No hay ninguna desventaja
   c) Mejora la calidad de los datos siempre

Problema 7: Una empresa de inversiones necesita comparar el desempeño financiero de diversas empresas, pero los datos están en distintas escalas. Se requiere normalizar y estandarizar los datos para hacer comparaciones justas.

Tareas

1. Cargar el dataset de indicadores financieros.
2. Aplicar estandarización utilizando scale().
3. Aplicar normalización con la fórmula (x - min(x)) / (max(x) - min(x)).
4. Evaluar las diferencias entre ambas transformaciones y decidir cuál es más adecuada.

1. Cargar el dataset de indicadores financieros.

```
df <- read.csv("C:/Users/DANIELAGUADALUPEAGUI/OneDrive - TECNOLOGICO
DE
ESTUDIOS SUPERIORES DE IXTAPALUCA/Documentos/TESI/OCTAVO
SEMESTRE/ANALISIS Y MODELADO DE DATOS/equipo/empresa.csv")
```

2. Aplicar estandarización utilizando scale().

```
columnas_numericas <- datos[, c("Ingresos", "utilidadNeta", "margenNeto",
"ROE",
"Liquidez", "Endeudamiento", "PERatio")]
datos_estandarizados <- as.data.frame(scale(columnas_numericas))
head(datos_estandarizados)
```

3. Aplicar normalización con la fórmula (x - min(x)) / (max(x) - min(x)).

```
normalizar <- function(x) {
(x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE))
}
datos_normalizados <- as.data.frame(lapply(columnas_numericas, normalizar))
head(datos_normalizados)
```

4. Evaluar las diferencias entre ambas transformaciones y decidir cuál es más adecuada.

```
library(ggplot2)
library(tidyr)
resumen_estandarizado <- summary(datos_estandarizados)
resumen_normalizado <- summary(datos_normalizados)
cat("Estadísticas descriptivas de datos estandarizados:\n")
print(resumen_estandarizado)
cat("\nEstadísticas descriptivas de datos normalizados:\n")
print(resumen_normalizado)
df_comparacion <- data.frame(
Original = columnas_numericas$Ingresos,
```

```
Estandarizado = datos_estandarizados$Ingresos,
Normalizado = datos_normalizados$Ingresos
)
df_largo <- pivot_longer(df_comparacion, cols = everything(), names_to =
"Método", values_to = "Valores")
ggplot(df_largo, aes(x = Valores, fill = Método)) +
geom_density(alpha = 0.5) +
labs(title = "Distribución: Estandarización vs. Normalización", x = "Valores", y =
"Densidad") +
theme_minimal()
```

Nota: se utilizo para esta parte la importacion de la librería ggplot y tidyr, pero se ponen al principio del código

Cuestionario de Evaluación

1. ¿Cuál es la diferencia entre estandarización y normalización?

   a) La estandarización ajusta los valores a una media de 0 y desviación estándar de 1, mientras que la normalización los escala entre 0 y 1
   b) No hay diferencia entre ambas técnicas
   c) La normalización siempre da mejores resultados

2. ¿Qué función de R permite estandarizar datos?

   a) normalize()
   b) scale()
   c) standardize()

3. ¿En qué caso es más útil la normalización en lugar de la estandarización?

   a) Cuando los datos tienen distribuciones con valores extremos
   b) Cuando se requiere comparar datos en diferentes escalas
   c) Cuando se trabaja con variables categóricas

Problema 8: Una empresa de comercio electrónico tiene un dataset con información de clientes y otro con el historial de compras. Se necesita fusionar ambas bases para

Tareas

1. Cargar los dos datasets en R.
2. Fusionar los datos usando left_join() de dplyr.
3. Detectar y manejar duplicados con distinct().
4. Verificar si hay inconsistencias después de la integración.

Codigo ejemplo:

```
# Cargar las librerías necesarias
library(dplyr)
```

1. Cargar y explorar los datasets

```
clientes <- read.csv("clientes.csv")  # Cargar dataset de clientes
compras <- read.csv("compras.csv")    # Cargar dataset de compras
```

2. Fusionar los datos usando left_join()

```
fusionado <- left_join(clientes, compras, by = "id_cliente")
```

3. Detectar y manejar duplicados con distinct()

```
fusionado <- fusionado %>% distinct()
```

4. Verificar inconsistencias después de la integración

```
# Verificar claves duplicadas
duplicados <- fusionado %>% group_by(id_cliente) %>% filter(n() > 1)
print(duplicados)

# Verificar valores faltantes
faltantes <- colSums(is.na(fusionado))
print(faltantes)

# Consulta de resumen
resumen <- fusionado %>%
  group_by(id_cliente) %>%
  summarise(total_compras = n(),
            monto_total = sum(monto, na.rm = TRUE))

print(resumen)
```

Cuestionario de Evaluación

1. ¿Qué función en R se usa para unir datasets por una columna común?

   a) merge()

   b) left_join()

   c) combine()

2. ¿Qué ocurre si se usa inner_join() en lugar de left_join()?

   a) Se eliminan las filas sin coincidencias en ambas tablas
   b) Se mantienen todas las filas de la tabla izquierda
   c) Se duplican las filas sin coincidencias

3. ¿Cómo se identifican valores duplicados en R?

   a) duplicated()
   b) unique()
   c) filter_duplicates()

Problema G: Un equipo de calidad de una fábrica detectó que ciertos valores de producción están fuera de lo esperado. Se necesita identificar y decidir qué hacer con estos valores atípicos.

Tareas

1. Visualizar los datos con un diagrama de caja usando ggplot2::geom_boxplot().
2. Determinar outliers utilizando el rango intercuartil (IQR).
3. Aplicar estrategias para manejarlos: eliminación, transformación o imputación.
4. Analizar el impacto de cada estrategia en el dataset.

```
Cargar librerías necesarias
install.packages("ggplot2")  # Solo si no lo tienes instalado
library(ggplot2)  # Cargar ggplot2 para gráficos
Crear un dataset simulado de producción
# Fijar semilla para reproducibilidad
set.seed(123)

# Generar datos normales con algunos valores atípicos
produccion <- data.frame(
  Unidades = c(rnorm(50, mean = 500, sd = 50),  # 50 valores dentro del rango
normal
               800, 850, 900,  # Valores atípicos altos
               200, 150)  # Valores atípicos bajos
)
Visualizar los datos con un diagrama de caja
ggplot(produccion, aes(y = Unidades)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "Diagrama de Caja de Producción",
       y = "Unidades Producidas") +
  theme_minimal()
Detectar outliers usando el Rango Intercuartil (IQR)
# Calcular Q1 y Q3
Q1 <- quantile(produccion$Unidades, 0.25)
Q3 <- quantile(produccion$Unidades, 0.75)
IQR <- Q3 - Q1  # Rango intercuartil

# Definir límites
limite_inferior <- Q1 - 1.5 * IQR
limite_superior <- Q3 + 1.5 * IQR

# Filtrar valores atípicos
outliers <- produccion$Unidades[produccion$Unidades < limite_inferior |
produccion$Unidades > limite_superior]
print(outliers)  # Muestra los valores atípicos detectados
Estrategias para manejar outliers
produccion_sin_outliers <- produccion[produccion$Unidades >= limite_inferior &
produccion$Unidades <= limite_superior, ]
```

```
Transformación (Reemplazar con la mediana)
mediana <- median(produccion$Unidades)
produccion$Unidades[produccion$Unidades < limite_inferior | produccion$Unidades >
limite_superior] <- mediana

Imputación (Sustitución por la media)
media <- mean(produccion$Unidades)
produccion$Unidades[produccion$Unidades < limite_inferior | produccion$Unidades >
limite_superior] <- media

summary(produccion$Unidades)  # Si transformaste los valores
```

Cuestionario de Evaluación

1. ¿Cómo se detectan valores atípicos en un conjunto de datos?

   a) Usando diagramas de caja y la técnica del rango intercuartil
   b) Eliminando cualquier dato que parezca extraño
   c) Usando solo la media y la desviación estándar

2. ¿Cuál de los siguientes métodos es adecuado para visualizar outliers?

   a) Gráfico de barras
   b) Diagrama de caja
   c) Histograma

3. ¿Cuál es una estrategia válida para manejar valores atípicos?

   a) Siempre eliminarlos
   b) Analizar su impacto y considerar imputaciones o transformaciones
   c) Ignorarlos y proceder con el análisis

Problema 10: Se han recopilado respuestas de una encuesta donde las variables son de tipo categórico (por ejemplo, satisfacción del cliente: "baja", "media", "alta"). Se requiere convertir estos datos en formato numérico para análisis estadístico.

Tareas

1. Convertir variables categóricas en factores con as.factor().
2. Aplicar codificación one-hot con model.matrix().
3. Evaluar cómo estas transformaciones impactan en modelos de regresión.

Problema 10

```
# Cargar librería necesaria
library(dplyr)

# Generar un dataframe de ejemplo
set.seed(123)
data <- data.frame(
  ID = 1:10,
  Satisfaccion = sample(c("baja", "media", "alta"), 10, replace = TRUE),
  Servicio = sample(c("A", "B", "C"), 10, replace = TRUE)
)

# Convertir variables categóricas en factores
data$Satisfaccion <- as.factor(data$Satisfaccion)
data$Servicio <- as.factor(data$Servicio)

# Aplicar codificación one-hot
one_hot_encoded <- model.matrix(~ Satisfaccion + Servicio - 1, data = data)

# Mostrar el dataframe original y el transformado
print("Data original:")
print(data)

print("Data codificada:")
print(one_hot_encoded)

# Evaluar el impacto en modelos de regresión
# Generamos una variable de respuesta ficticia
data$Score <- rnorm(10, mean = 50, sd = 10)


# Ajustar un modelo de regresión lineal
modelo <- lm(Score ~ ., data = as.data.frame(one_hot_encoded))
summary(modelo)
```

Cuestionario de Evaluación

1. ¿Por qué es importante codificar variables categóricas en modelos predictivos?

    a) Porque los modelos estadísticos requieren datos numéricos
    b) Porque es obligatorio para todas las variables
    c) No es necesario codificarlas

2. ¿Qué técnica de codificación crea múltiples columnas binarias?

    a) One-hot encoding
    b) Label encoding
    c) Scaling

3. ¿Qué función permite convertir una variable categórica en un factor en R?

    a) as.factor()
    b) convert()
    c) factorize()