Proyecto_AN

Jhoan Rodriguez

2025-10-31

Libreria utilizadas

```
library(formattable)
library(dplyr)
library(tidyverse)
library(readr)
library(ggplot2)
library(scales)
library(knitr)
library(kableExtra)
library(cluster)
library(factoextra)
library(caret)
library(randomForest)
library(fastDummies)
library(DescTools)
```

1. Introducción

- Breve presentación del proyecto.
- Contextualización del problema y su relevancia.

2. Justificación

- Importancia del estudio de datos en el problema abordado.
- Valor agregado del análisis realizado.

3. Objetivos

- General: Enunciar el objetivo principal del proyecto.
- Específicos: Al menos tres objetivos que detallen las metas técnicas del análisis.

4. Fases del Proceso KDD.

4.1 Dominio del problema

- Describir el contexto del fenómeno o situación a analizar.
- Formular preguntas de investigación o hipótesis que orienten la minería de datos.
- Identificar la relevancia del problema y su impacto en la toma de decisiones.

4.2 Selección de Datos

Importacion de datos

```
Data <- read.csv(
  "urban_pluvial_flood_risk_dataset.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)
```

Selección de variables relevantes

Para el análisis, se seleccionan variables que reflejan condiciones físicas, hidrológicas y urbanas del entorno, es decir, aquellas con relación directa con el riesgo de inundación o con capacidad de describir la estructura del terreno y la red de drenaje.

```
# Selección de variables relevantes
DataSeleccion <- Data %>%
  select(
    elevation_m,
    drainage_density_km_per_km2,
    storm_drain_proximity_m,
    historical_rainfall_intensity_mm_hr,
    return_period_years,
    land_use,
    soil_group,
    storm_drain_type
  )
```

Justificación de la selección:

- elevation_m: La altitud define la capacidad de escurrimiento del agua.
- drainage_density_km_per_km2: Representa la eficiencia de drenaje urbano.
- storm_drain_proximity_m: Influye directamente en la probabilidad de acumulación de agua.
- historical_rainfall_intensity_mm_hr: Determina la presión pluvial histórica en la zona.
- return_period_years: Indica la frecuencia esperada de eventos extremos.
- land_use, soil_group, storm_drain_type: Variables categóricas que afectan la infiltración, escorrentía y drenaje.

Limpieza de datos y manejo de valores faltante

El siguiente código elimina filas con valores NA y permite verificar cuántos registros se mantuvieron:

Eliminación de filas con NA

```
# Eliminación de registros (filas) con más del 30% de valores NA
DataLimpia <- DataSeleccion %>%
  filter(if_all(everything(), ~ !is.na(.)))

# Mostrar cantidad de filas antes y después
nrow(DataSeleccion)
```

```
## [1] 2963
```

```
nrow(DataLimpia)
```

```
## [1] 2332
```

Motivos de eliminación:

- segment_id: Identificador único, no aporta información para el análisis.
- city_name, admin_ward, catchment_id: Identificadores geográficos que no reflejan condiciones físicas o hidrológicas.
- latitude, longitude: Variables espaciales que requieren proyección o normalización especial.
- dem_source, rainfall_source: Describen la procedencia de los datos, no influyen directamente en los fenómenos analizados.
- risk_labels: Etiqueta de riesgo, reservada solo para validación, no debe participar en el entrenamiento del modelo.

4.3 Limpieza de Datos

Errores e inconsistencias

Se revisó la existencia de valores duplicados o inconsistencias tipográficas en campos categóricos.

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

El resultado es que no existen registros duplicados en el dataset.

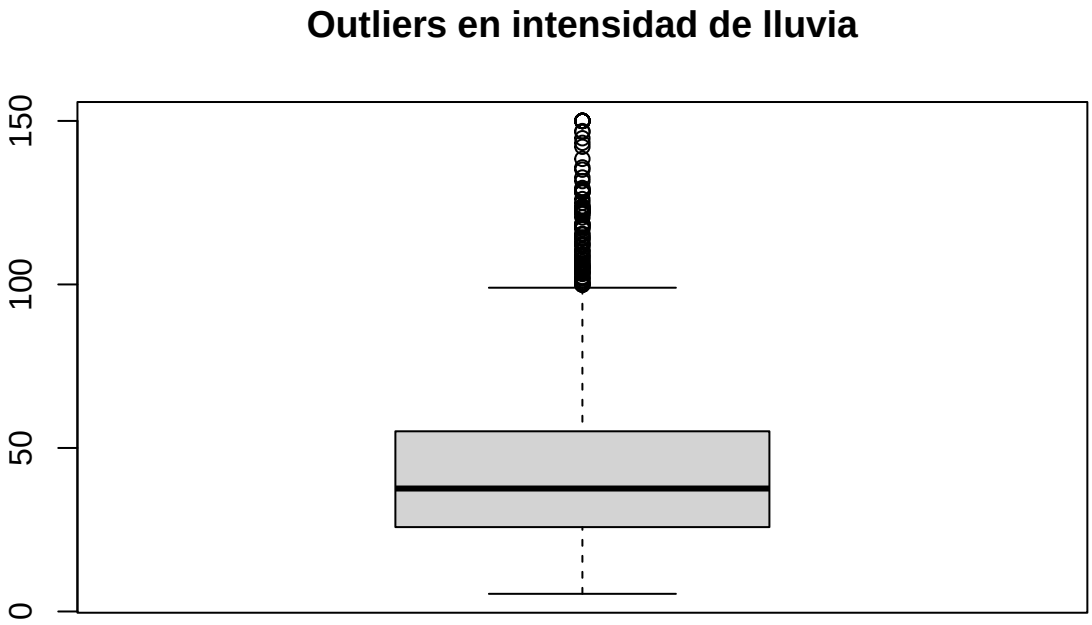
Outliers

Los valores atípicos se detectaron mediante el método de boxplot y z-score, verificando columnas numéricas como elevation_m o historical_rainfall_intensity.

Outliers historical_rainfall_intensity_mm_hr

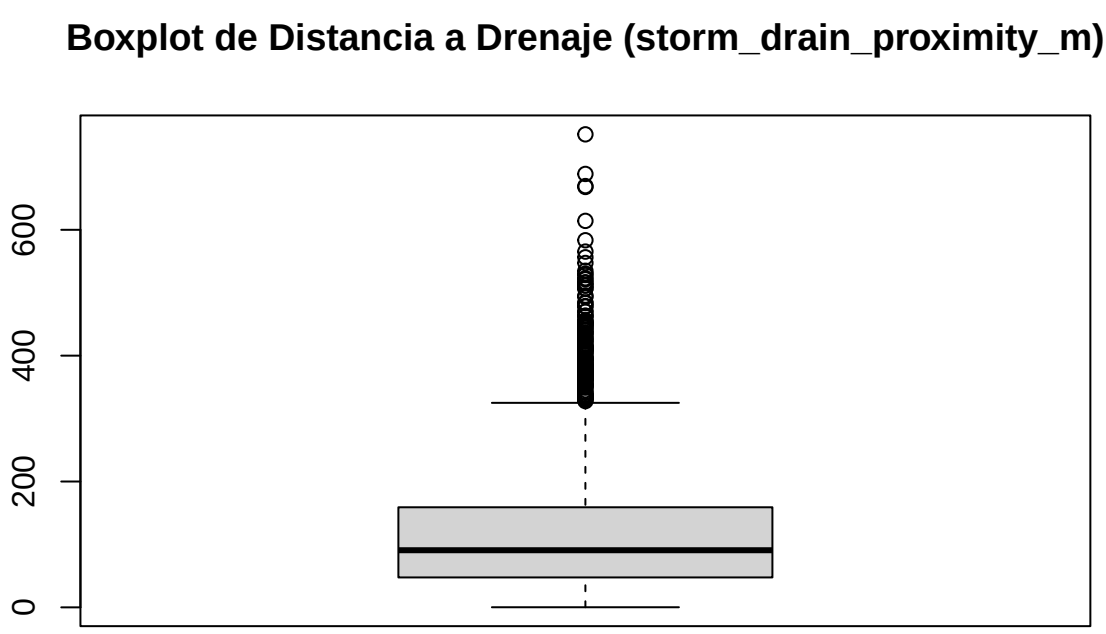
```r
# Detección de outliers por Z-score
z_scores <- scale(DataLimpia[, sapply(DataLimpia, is.numeric)])
outliers <- which(abs(z_scores) > 3, arr.ind = TRUE)

# Visualización de posibles outliers
boxplot(DataLimpia$historical_rainfall_intensity_mm_hr, main="Outliers en intensidad de lluvia")
```



Outliers storm_drain_proximity_m

```r
boxplot(DataLimpia$storm_drain_proximity_m,
        main = "Boxplot de Distancia a Drenaje (storm_drain_proximity_m)")
```



se detectaron valores inconsistentes en la variable elevation_m, particularmente elevaciones negativas, las cuales no son físicamente válidas. Por ello, se eliminaron los registros correspondientes

```
#  Eliminar elevaciones negativas
DataLimpia <- DataLimpia[DataLimpia$elevation_m >= 0, ]
```

Posteriormente, para evitar la distorsión que generan valores extremos en la variable historical_rainfall_intensity_mm_hr, se aplicó el método de Winsorización, ajustando los valores al percentil 1% y 99%.

```
#  Aplicar winsorización en la columna de lluvia histórica en una variable nueva llamara rainfall_winso
DataLimpia$rainfall_winsorizada <- DescTools::Winsorize(
  DataLimpia$historical_rainfall_intensity_mm_hr,
  val = quantile(
    DataLimpia$historical_rainfall_intensity_mm_hr,
    probs = c(0.01, 0.99),
    na.rm = TRUE
  )
)
```

```
#  Aplicar winsorización en la columna de lluvia histórica en una variable nueva llamara rainfall_winso
DataLimpia$storm_drain_proximity_winsorizada <- DescTools::Winsorize(
  DataLimpia$storm_drain_proximity_m,
  val = quantile(
    DataLimpia$storm_drain_proximity_m,
    probs = c(0.01, 0.99),
    na.rm = TRUE
```

```
  )
)
```

tratemiendo de datos:

• Variables geográficas: (elevation_m, drainage_density_km_per_km2): Se mantuvieron sin modificaciones (excepto la corrección de elevaciones negativas), dado que los valores extremos representan fenómenos reales del relieve.

• Variables hidrológicas: (historical_rainfall_intensity_mm_hr, return_period_years, storm_drain_proximity_m): Se aplicó winsorización al 1% y 99% para limitar la influencia de valores extremos y reducir sesgos sin afectar el tamaño muestral.

Justificación:

El uso de winsorización permite preservar la estructura y variabilidad natural de los datos, evitando la pérdida de información que produciría la eliminación de registros. Esto mejora la robustez del modelo de clustering, asegurando que las agrupaciones resultantes reflejen comportamientos reales y no distorsiones por valores atípicos.

4.4 Transformación de Datos. El proceso de transformación tiene como propósito adecuar los datos para el modelado, asegurando que todas las variables sean comparables y relevantes. Se realizaron las siguientes etapas:

Normalización de variables numéricas

Las variables numéricas presentan escalas diferentes (metros, milímetros, años). Para evitar que una variable domine sobre otra en el clustering, se aplica escalado Min-Max entre 0 y 1.

```
DataTransform <- DataLimpia %>%
  mutate(across(c(elevation_m,
                  drainage_density_km_per_km2,
                  storm_drain_proximity_m,
                  historical_rainfall_intensity_mm_hr,
                  return_period_years),
                ~ (.-min(.)) / (max(.)-min(.)),
                .names = "norm_{col}"))
```

Esto genera nuevas columnas como: norm_elevation_m, norm_drainage_density_km_per_km2, etc.

Codificación de variables categóricas

Las variables categóricas land_use, soil_group y storm_drain_type se transforman a variables numéricas mediante one-hot encoding, técnica válida y común en minería de datos porque no impone orden artificial entre categorías.

```
DataTransform <- fastDummies::dummy_cols(DataTransform,
                                         select_columns = c("land_use", "soil_group", "storm_drain_type
                                         remove_first_dummy = TRUE,
                                         remove_selected_columns = TRUE)
```

Justificación:

Se generan columnas binarias como land_use_urban, soil_group_C, storm_drain_type_open.

Creación de variables derivadas.

Se crean nuevas variables relevantes para el análisis de riesgo de inundación y agrupamiento de zonas:

```
DataTransform <- DataTransform %>%
  mutate(
    elevation_rain_ratio = norm_elevation_m / (norm_historical_rainfall_intensity_mm_hr + 0.001),
    drainage_rain_index = norm_drainage_density_km_per_km2 * norm_historical_rainfall_intensity_mm_hr,
    proximity_index = 1 / (norm_storm_drain_proximity_m + 0.01)
  )
```

Justificación:

- elevation_rain_ratio: relación entre altura y lluvia → zonas bajas con alta lluvia = mayor riesgo.
- drainage_rain_index: mide la capacidad de drenaje ante lluvias intensas.
- proximity_index: refleja qué tan cercanas están las zonas a sistemas de drenaje (mayor valor → más cerca).

Comparación antes y después.

```
head(DataLimpia %>% select(elevation_m, drainage_density_km_per_km2))
```

```
##   elevation_m drainage_density_km_per_km2
## 1       30.88                       11.00
## 2       24.28                        7.32
## 3       35.70                        4.50
## 4       15.36                        8.97
## 5       15.80                        8.25
## 6       20.08                        5.88
```

```
head(DataTransform %>% select(norm_elevation_m, norm_drainage_density_km_per_km2, elevation_rain_ratio))
```

```
##   norm_elevation_m norm_drainage_density_km_per_km2 elevation_rain_ratio
## 1       0.11571921                        0.9000000           1.51503887
## 2       0.09097045                        0.5560748           0.18334937
## 3       0.13379331                        0.2925234           1.24458093
## 4       0.05752212                        0.7102804           0.07217431
## 5       0.05917204                        0.6429907           0.25132553
## 6       0.07522124                        0.4214953           0.15708071
```

Clustering (para segmentar zonas por riesgo.

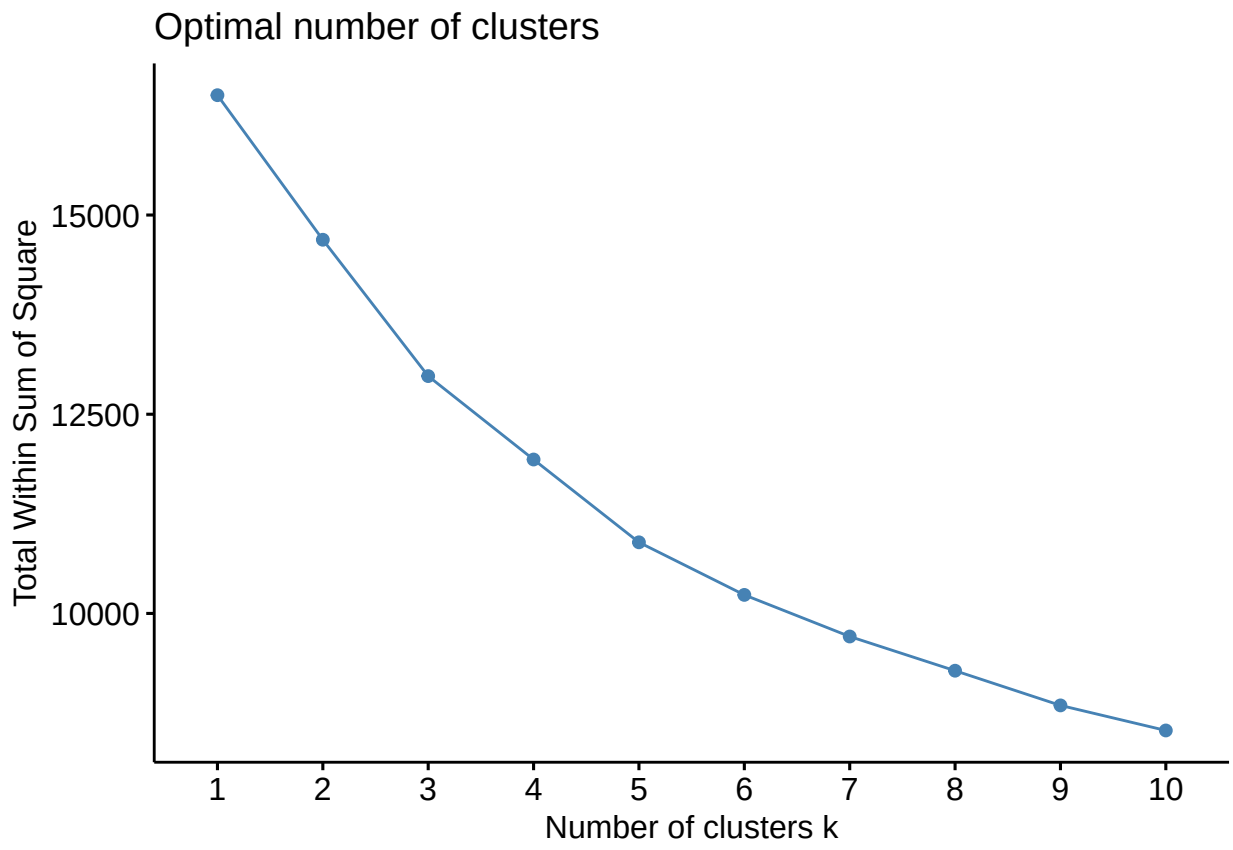
```r
data_cluster <- DataLimpia %>%
  select(elevation_m,
    drainage_density_km_per_km2,
    storm_drain_proximity_m,
    historical_rainfall_intensity_mm_hr,
    return_period_years,
    land_use,
    soil_group,
    storm_drain_type)

# Convertir variables categóricas a numéricas si es necesario
data_cluster <- data_cluster %>%
  mutate(across(where(is.character), as.factor)) %>%
  mutate(across(where(is.factor), as.numeric))

# Escalamiento (muy importante)
data_scaled <- scale(data_cluster)
```

Clustering (K-Means) + Validación

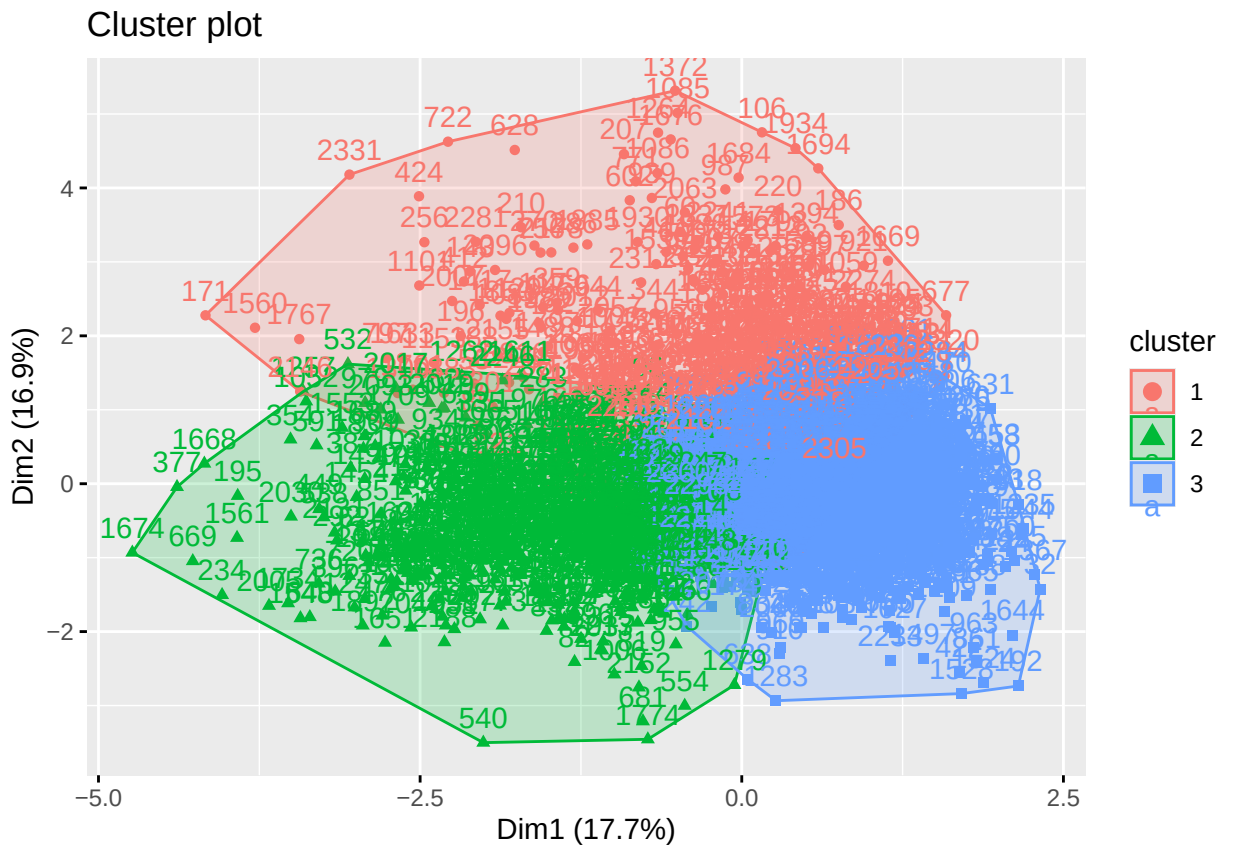
```r
# 2.1 Determinar número óptimo de clusters con el método del codo
fviz_nbclust(data_scaled, kmeans, method = "wss")
```



```r
# 2.2 Probar K-Means con k = 3 (o el valor que el gráfico indique)
set.seed(123)
kmeans_model <- kmeans(data_scaled, centers = 3, nstart = 25)

# 2.3 Añadir los clusters al dataset original
DataLimpia$cluster_risk <- as.factor(kmeans_model$cluster)

# 2.4 Visualizar los clusters
fviz_cluster(kmeans_model, data = data_scaled)
```



Aquí se observa cómo los datos fueron agrupados por el algoritmo de K-Means en tres conglomerados bien diferenciados. Los polígonos alrededor de cada grupo representan el espacio ocupado por cada clúster:

Se evidencia una separación clara entre los tres grupos, lo que indica que las variables seleccionadas (elevación, tipo de suelo, densidad de drenaje, proximidad a drenajes, entre otras) aportaron información suficiente para segmentar zonas con características de riesgo similares.

El clúster identificado como riesgo Alto se concentra hacia los valores negativos de Dim1 y Dim2, mientras que el riesgo Bajo tiende a ubicarse hacia valores positivos, confirmando diferencias significativas en las características geográficas e hidrológicas de cada grupo.

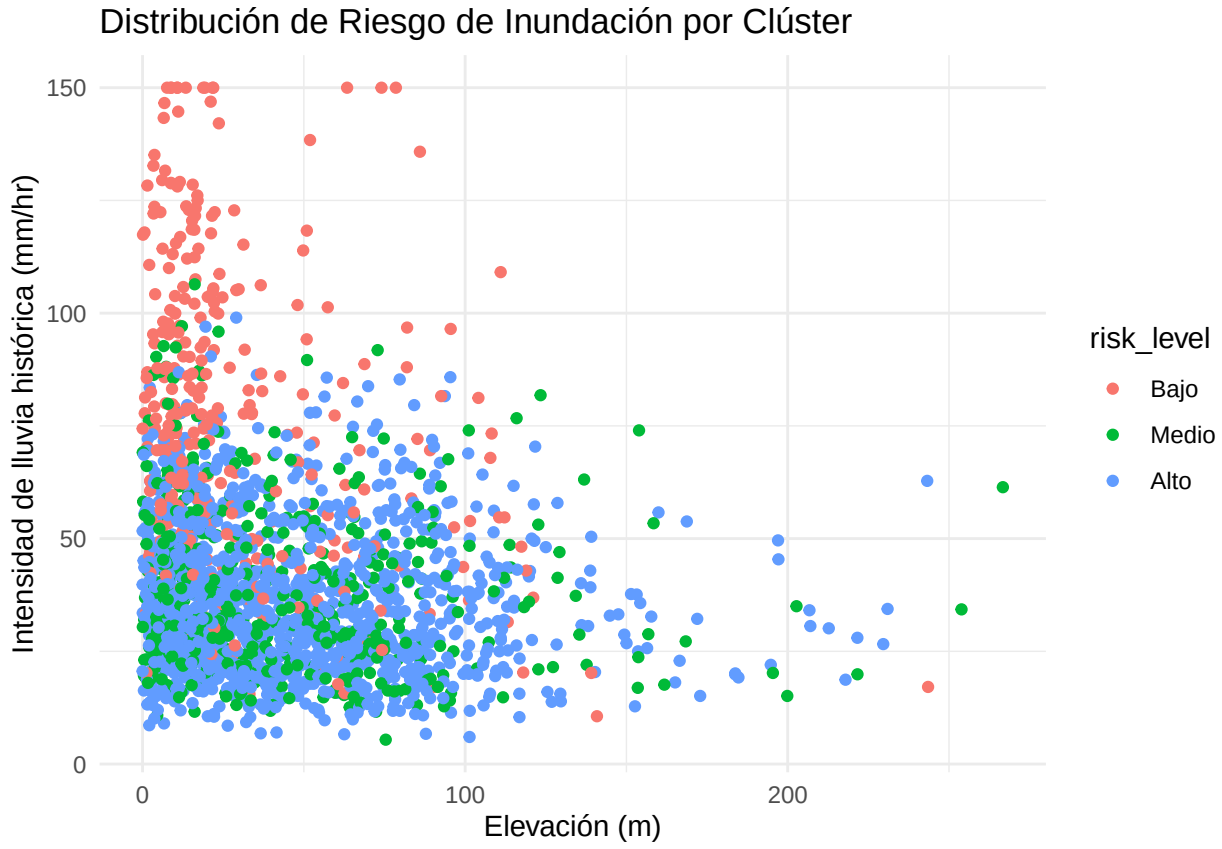
La distribución compacta de cada clúster refuerza la consistencia del modelo y respalda la fiabilidad de la clasificación realizada.

3. Convertir Clustering en Etiquetas de Riesgo

```
# Crear una columma final de riesgo (Bajo, Medio, Alto)
DataLimpia$risk_level <- recode(DataLimpia$cluster_risk,
                            "1" = "Bajo",
                            "2" = "Medio",
                            "3" = "Alto")
DataLimpia$risk_level <- factor(DataLimpia$risk_level, levels = c("Bajo", "Medio", "Alto"))
```

```
library(ggplot2)

ggplot(DataLimpia, aes(x = elevation_m, y = historical_rainfall_intensity_mm_hr, color = risk_level)) +
  geom_point() +
  labs(title = "Distribución de Riesgo de Inundación por Clúster",
       x = "Elevación (m)",
       y = "Intensidad de lluvia histórica (mm/hr)") +
  theme_minimal()
```



Se evidencia la distribución del riesgo de inundación según los clústeres obtenidos, con base en dos variables ambientales clave: elevación (m) y la intensidad histórica de lluvia (mm/hr). Se observa que:

Las áreas con menor elevación y mayor intensidad de lluvia histórica presentan mayor concentración de puntos correspondientes al clúster de riesgo Alto, indicando mayor susceptibilidad a inundaciones.

El clúster de riesgo Medio se ubica en una zona intermedia tanto en elevación como en precipitación.

El clúster de riesgo Bajo se encuentra mayormente distribuido en zonas con elevaciones más altas y menor intensidad de lluvia, lo cual sugiere condiciones más seguras frente a posibles inundaciones.

4.5 Minería de Datos

- Seleccionar y justificar el algoritmo o técnica empleada (clasificación, regresión, clustering, etc.).
- Describir la división de datos (entrenamiento y prueba).
- Presentar las métricas de evaluación (Accuracy, F1-Score, MAE, etc.).
- Incluir visualizaciones que respalden los resultados del modelo.

4.6 Interpretación y Evaluación

- Analizar críticamente los resultados obtenidos.
- Validar el conocimiento descubierto frente a las hipótesis y objetivos planteados.
- Evaluar el valor del conocimiento extraído para el contexto del problema.

5. Conclusiones

- Sintetizar los principales hallazgos.
- Reflexionar sobre el proceso completo y sus limitaciones.
- Proponer posibles trabajos futuros o mejoras.

6. Anexos

- Gráficos, tablas, fragmentos de código, resultados adicionales que complementen el análisis.