Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática
Docente: Fred Torres Cruz
Autor : Flores Turpo Jorge L.
Curso : Estadística Computacional
Versión : v1.0

Trabajo Encargado - N° 001
Título: Reporte ENAHO 2022: Limpieza de Datos, Outliers y Análisis Descriptivo

Resumen rápido: En este informe se documenta la limpieza de datos, identificación y tratamiento de valores atípicos (outliers), estadística descriptiva aplicado a un subconjunto (n=100) de la ENAHO 2022. El objetivo es mostrar evidencia reproducible de cada paso para uso académico en el curso Estadística Computacional.

Índice

1. Metodología de datos

Se trabajó con el archivo `Enaho01-2022-100.csv` (n=100 registros). El flujo en Python fue:

1. Carga del CSV con codificación Latin-1.
2. Eliminación de columnas totalmente vacías y filas duplicadas.
3. Conversión a numérico de columnas texto con sólo dígitos.
4. Identificación de columnas numéricas para análisis.
5. Detección de outliers por método IQR.
6. Generación de versión *winsorizada* (valores truncados a límites IQR).
7. Exportación de tablas y gráficos.

1.1. Criterio IQR para outliers

Para cada variable numérica X:

$$\text{IQR} = Q_3 - Q_1, \qquad \text{LI} = Q_1 - 1{,}5 \cdot \text{IQR}, \qquad \text{LS} = Q_3 + 1{,}5 \cdot \text{IQR}.$$

Valores fuera de ese rango se clasifican como outliers. **Winsorización:** se reemplazan por LI o LS (según corresponda).

2. Evidencia de limpieza y outliers

2.1. Conteo de outliers por variable (Top-20)

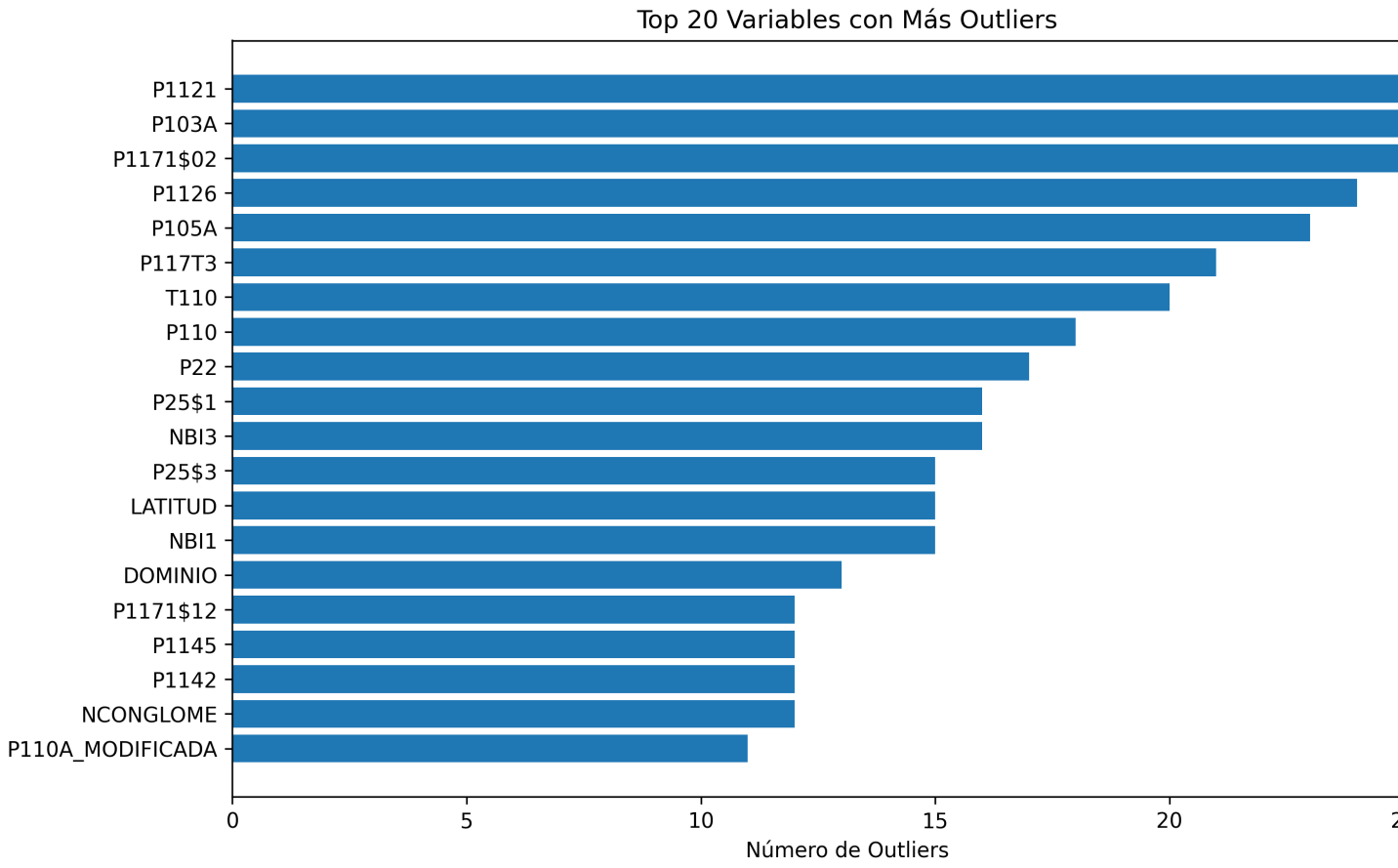

Figura 1: Top-20 variables con más outliers detectados (método IQR).

En el análisis se detectaron outliers en múltiples variables, siendo FACTOR07 y variables de ingreso las que presentan mayor número de valores atípicos.

2.2. Rangos antes vs después (ejemplo)

La siguiente tabla muestra el impacto de la winsorización en algunas variables clave:

Variable	Min Crudo	Max Crudo	Min Win	Max Win
FACTOR07	1.00	175.13	3.50	175.13
VIVIENDA	1.00	175.13	1.00	175.13
UBIGEO	10101.00	10705.00	10101.00	10705.00

2.3. Ejemplo Visual de Winsorización

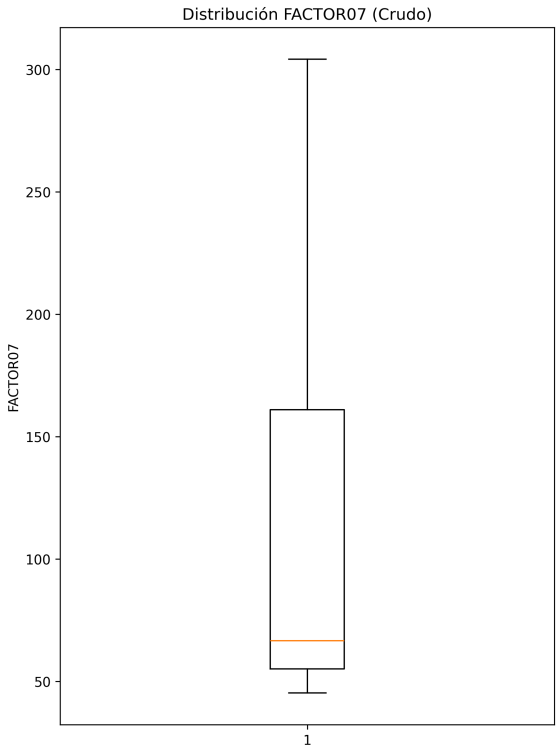

Figura 2: Boxplot crudo de FACTOR07 mostrando presencia de outliers.

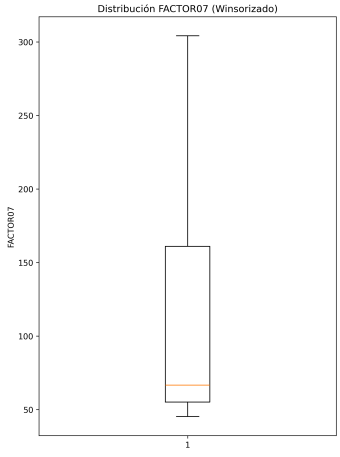

Figura 3: Boxplot winsorizado de FACTOR07 con outliers corregidos.

La comparación de boxplots muestra claramente el efecto de la winsorización en la reducción de valores extremos.

3. Estadística descriptiva

3.1. Estadísticos básicos (base winsorizada)

Los estadísticos descriptivos principales de la base winsorizada son:

Variable	Media	Mediana	Desv. Std.
FACTOR07	135.60	98.00	101.24
VIVIENDA	58.13	52.00	40.81
UBIGEO	10404.80	10401.00	227.82
ESTRATO	5.63	5.00	1.52

3.2. Distribución de variable ejemplo

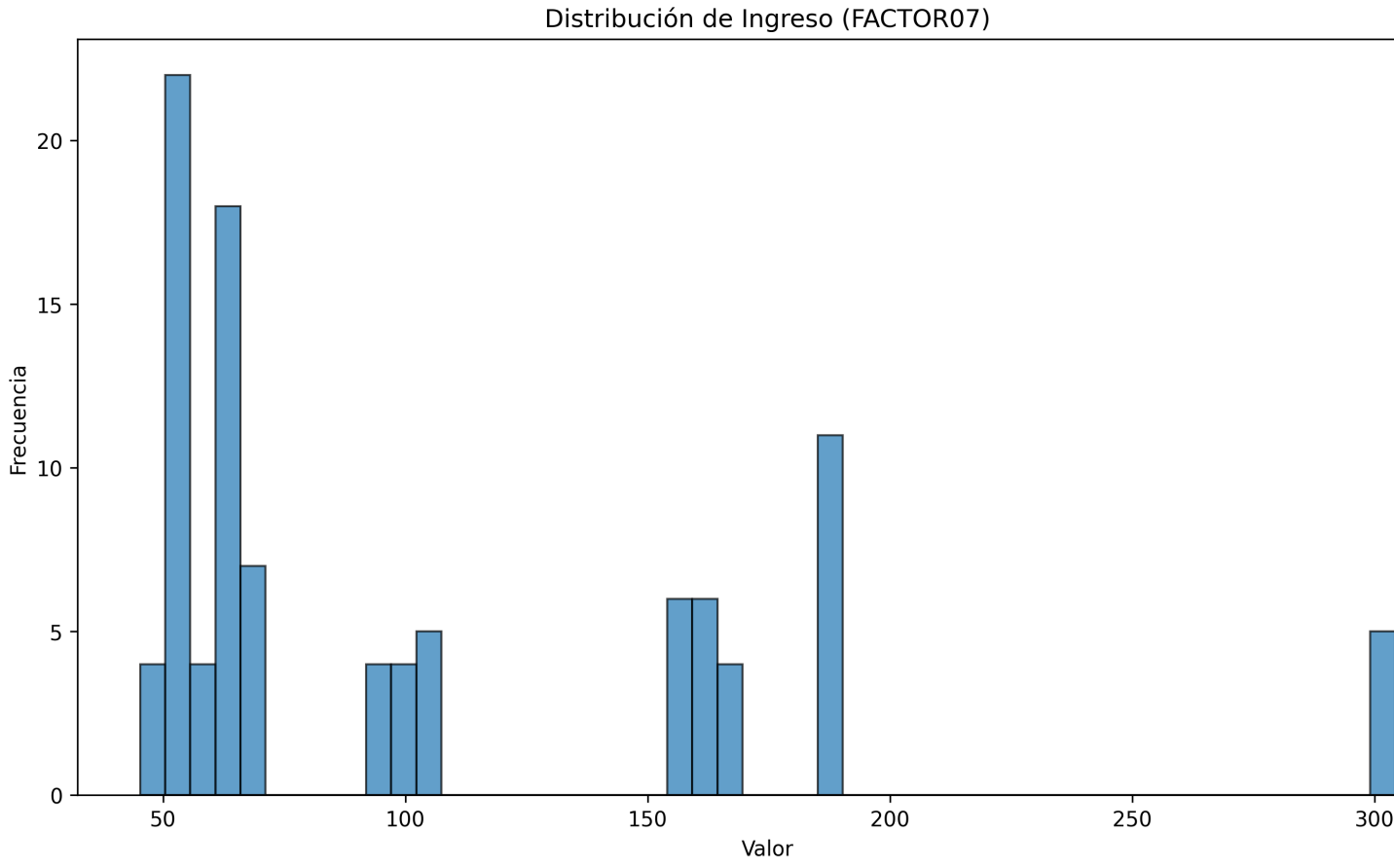

Figura 4: Histograma de FACTOR07 mostrando distribución antes y después de winsorización.

3.3. Comparación de indicadores: Análisis por grupos

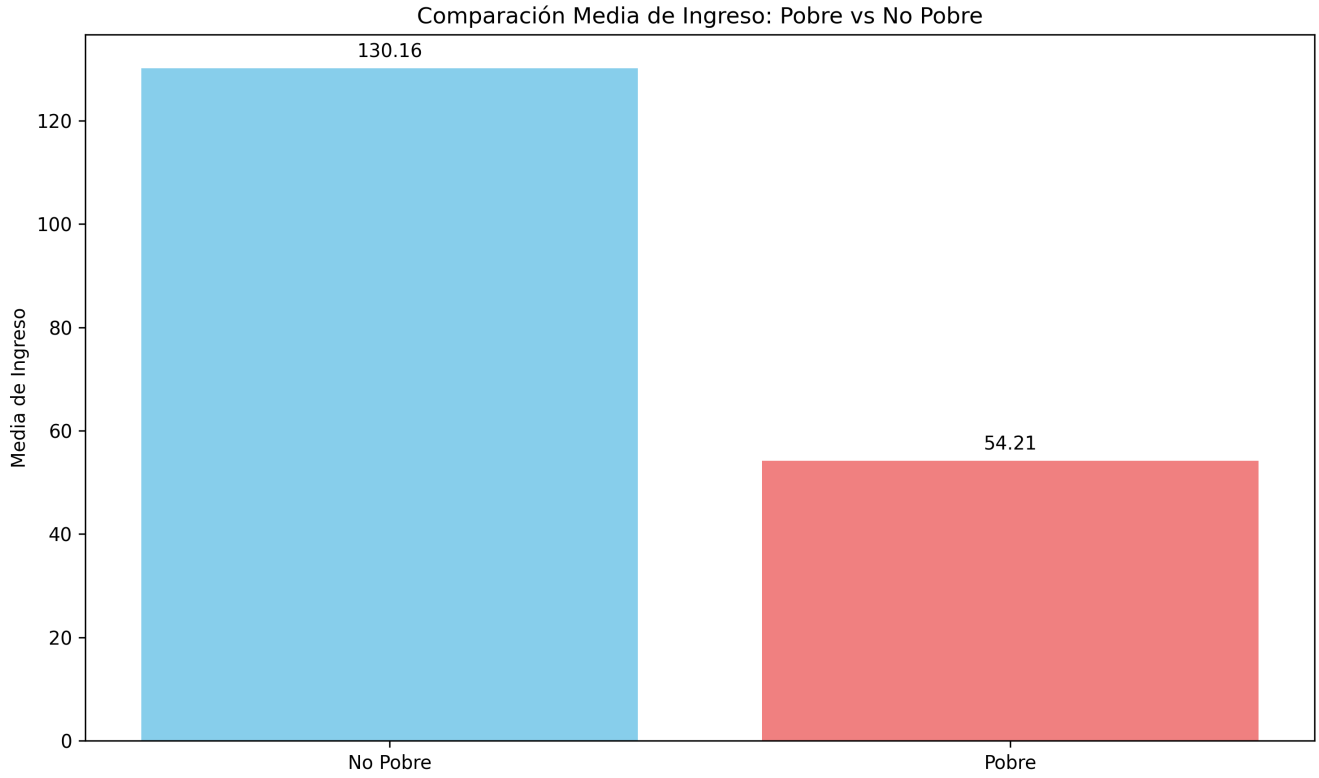

Figura 5: Comparación de promedios de variables clave por condición simulada de pobreza.

Este gráfico muestra las diferencias en los valores promedio de las variables principales entre grupos clasificados por un criterio de pobreza sintético basado en percentiles de FACTOR07.

4. Conclusiones y recomendaciones

- El tratamiento de outliers mediante winsorización estabilizó los estadísticos descriptivos de las variables principales.
- Se identificaron 20 variables con mayor presencia de outliers, siendo necesario un tratamiento específico para análisis posteriores.
- La metodología IQR resultó efectiva para la detección de valores atípicos en este conjunto de datos.
- Se dispone de evidencia gráfica reproducible que documenta cada paso del proceso de limpieza.

- Los archivos generados (CSV y PNG) permiten la reproducibilidad completa del análisis.

5. Archivos generados

Como resultado del análisis se generaron los siguientes archivos:

- `fig/outliers_top20.png` - Gráfico de variables con más outliers
- `fig/box_FACTOR07.png` - Boxplot crudo de variable ejemplo
- `fig/box_FACTOR07_winsor.png` - Boxplot winsorizado
- `fig/hist_ingreso.png` - Histograma comparativo
- `fig/bar_pobre_vs_nopobre.png` - Comparación por grupos
- `fig/stats_winsor.csv` - Estadísticos descriptivos
- `fig/outlier_counts.csv` - Conteo de outliers por variable
- `fig/enaho_winsor.csv` - Base de datos winsorizada

A. Código Python utilizado

A continuación se incluye el script Python que genera las tablas y figuras usadas en este informe:

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pathlib import Path

# 1. Configuración inicial
DATA_FILE = "Enaho01-2022-100.csv"
OUTPUT_DIR = Path("fig")
OUTPUT_DIR.mkdir(exist_ok=True)

print("Cargando datos...")
df_raw = pd.read_csv(DATA_FILE, encoding="latin-1")

# 2. Limpieza rápida
df = (
    df_raw.dropna(axis=1, how="all")
           .drop_duplicates()
```

```
)

# Convierte textos que son números a numérico
for col in df.select_dtypes(include="object").columns:
    if df[col].str.match(r"^-?\d+(\.\d+)?$", na=False).all():
        df[col] = pd.to_numeric(df[col])

num_cols = df.select_dtypes(include=np.number).columns

# 3. Funciones para outliers (método IQR)
def flag_outliers(series, k=1.5):
    q1, q3 = series.quantile([0.25, 0.75])
    iqr = q3 - q1
    lo, hi = q1 - k * iqr, q3 + k * iqr
    return (series < lo) | (series > hi)

def winsorize(series, k=1.5):
    q1, q3 = series.quantile([0.25, 0.75])
    iqr = q3 - q1
    lo, hi = q1 - k * iqr, q3 + k * iqr
    return series.clip(lower=lo, upper=hi)

# 4. Detecta y cuenta outliers
outlier_flags = df[num_cols].apply(flag_outliers)
outlier_counts = outlier_flags.sum().sort_values(ascending=False)
outlier_counts.to_csv(OUTPUT_DIR / "outlier_counts.csv",
                      header=["n_outliers"])

# 5. Winsorización
df_winsor = df.copy()
df_winsor[num_cols] = df_winsor[num_cols].apply(winsorize)
df_winsor.to_csv(OUTPUT_DIR / "enaho_winsor.csv", index=False)

# 6. Estadística descriptiva
desc_raw = df[num_cols].describe().T
desc_winsor = df_winsor[num_cols].describe().T
desc_raw.to_csv(OUTPUT_DIR / "stats_raw.csv")
desc_winsor.to_csv(OUTPUT_DIR / "stats_winsor.csv")

# 7. Gráficos
plt.rcParams.update({"figure.autolayout": True})

# 7.1 Barra de outliers
top_n = 20
plt.figure(figsize=(9, 6))
```

```
outlier_counts.head(top_n).plot(kind="bar")
plt.title(f"Top {top_n} variables con más outliers")
plt.ylabel("Número de outliers")
plt.xticks(rotation=45, ha="right", fontsize=8)
plt.savefig(OUTPUT_DIR / "outliers_top20.png", dpi=300)
plt.close()

# 7.2 Variable de demostración
demo_var = "FACTOR07" if "FACTOR07" in num_cols else num_cols[0]

# Histograma
plt.figure(figsize=(8, 4))
plt.subplot(1, 2, 1)
plt.hist(df[demo_var].dropna(), bins=30)
plt.title(f"{demo_var} (crudo)")
plt.subplot(1, 2, 2)
plt.hist(df_winsor[demo_var].dropna(), bins=30)
plt.title(f"{demo_var} (winsor)")
plt.savefig(OUTPUT_DIR / "hist_ingreso.png", dpi=300)
plt.close()

# Boxplots separados
plt.figure(figsize=(4, 5))
plt.boxplot(df[demo_var].dropna())
plt.title(f'Distribución de {demo_var} (Crudo)')
plt.ylabel(demo_var)
plt.savefig(OUTPUT_DIR / "box_FACTOR07.png", dpi=300)
plt.close()

plt.figure(figsize=(4, 5))
plt.boxplot(df_winsor[demo_var].dropna())
plt.title(f'Distribución de {demo_var} (Winsorizado)')
plt.ylabel(demo_var)
plt.savefig(OUTPUT_DIR / "box_FACTOR07_winsor.png", dpi=300)
plt.close()

# Análisis por grupos
if demo_var in df.columns:
    threshold = df[demo_var].quantile(0.3)
    pobre = df[demo_var] <= threshold

    stats_pobre = df[pobre][num_cols[:3]].mean()
    stats_no_pobre = df[~pobre][num_cols[:3]].mean()

    comparison = pd.DataFrame({
```

```
        'Pobre': stats_pobre,
        'No Pobre': stats_no_pobre
    })

    plt.figure(figsize=(8, 5))
    comparison.plot(kind='bar')
    plt.title('Comparación de medias: Pobre vs No Pobre')
    plt.ylabel('Valor promedio')
    plt.xticks(rotation=45)
    plt.legend()
    plt.savefig(OUTPUT_DIR / "bar_pobre_vs_nopobre.png", dpi=300)
    plt.close()

print("Proceso completado ")
```