

## ✓ Análisis Descriptivo y Visualización del Sector Industrial de Tierra del Fuego

### Objetivo

Este primer análisis tiene como objetivo proporcionar una comprensión completa de los datos relacionados con la industria en Tierra del Fuego a través de estadísticas descriptivas y visualizaciones. Esto servirá como una base sólida para los modelos de aprendizaje automático que se desarrollarán posteriormente.

### ✓ Importar Librerías y Cargar Datos

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Configuración de estilo para los gráficos
sns.set(style="whitegrid")

# Cargar los datasets usando los enlaces directos a los archivos CSV con codificación 'latin1' y el delimitador correcto
personal_activo_df = pd.read_csv('https://raw.githubusercontent.com/monopologit/Aprendizaje2024/master/data/raw/Personal_Activo_Industria')
establecimientos_df = pd.read_csv('https://raw.githubusercontent.com/monopologit/Aprendizaje2024/master/data/raw/Establecimientos_industria')
produccion_confeccionista_df = pd.read_csv('https://raw.githubusercontent.com/monopologit/Aprendizaje2024/master/data/raw/Produccion_Industrial_Confeccionista')
produccion_electronica_df = pd.read_csv('https://raw.githubusercontent.com/monopologit/Aprendizaje2024/master/data/raw/Produccion_Industrial_Electronica')
produccion_pesquera_df = pd.read_csv('https://raw.githubusercontent.com/monopologit/Aprendizaje2024/master/data/raw/Produccion_Industrial_Pesquera')
produccion_textil_df = pd.read_csv('https://raw.githubusercontent.com/monopologit/Aprendizaje2024/master/data/raw/Produccion_Industrial_Textil')
produccion_plastica_df = pd.read_csv('https://raw.githubusercontent.com/monopologit/Aprendizaje2024/master/data/raw/Produccion_Industrial_Plastica')
produccion_otros_df = pd.read_csv('https://raw.githubusercontent.com/monopologit/Aprendizaje2024/master/data/raw/Produccion_Industrial_Otros')
```

### ✓ Limpiar y Preprocesar los Datos

En esta sección, eliminaremos espacios en blanco en los nombres de las columnas y convertiremos los nombres de los meses en español a números. También manejaremos valores faltantes reemplazándolos con ceros.

```
# Limpiar los nombres de las columnas eliminando espacios en blanco extra
for df in [personal_activo_df, establecimientos_df, produccion_confeccionista_df, produccion_electronica_df,
          produccion_pesquera_df, produccion_textil_df, produccion_plastica_df, produccion_otros_df]:
    df.columns = df.columns.str.strip()

# Diccionario para convertir los nombres de los meses en español a números
meses = {
    'Enero': '01', 'enero': '01', 'Febrero': '02', 'febrero': '02', 'Marzo': '03', 'marzo': '03',
    'Abril': '04', 'abril': '04', 'Mayo': '05', 'mayo': '05', 'Junio': '06', 'junio': '06',
    'Julio': '07', 'julio': '07', 'Agosto': '08', 'agosto': '08', 'Septiembre': '09', 'septiembre': '09',
    'Octubre': '10', 'octubre': '10', 'Noviembre': '11', 'noviembre': '11', 'Diciembre': '12', 'diciembre': '12'
}

# Función para reemplazar los nombres de los meses en español por números y manejar NaN
def convertir_fecha(row):
    if pd.isna(row['Año']) or pd.isna(row['Mes']):
        return pd.NaT
    año = str(int(row['Año']))
    mes = str(row['Mes']).strip()
    mes = meses.get(mes, '01') # Asignar '01' como valor predeterminado si el mes no se encuentra en el diccionario
    return pd.to_datetime(f'{año}-{mes}', format='%Y-%m')

# Aplicar la función para crear la columna 'Fecha' en cada dataset
for df in [personal_activo_df, establecimientos_df, produccion_confeccionista_df, produccion_electronica_df,
          produccion_pesquera_df, produccion_textil_df, produccion_plastica_df, produccion_otros_df]:
    df['Fecha'] = df.apply(convertir_fecha, axis=1)

# Establecer la columna 'Fecha' como el índice para todos los datasets
for df in [personal_activo_df, establecimientos_df, produccion_confeccionista_df, produccion_electronica_df,
          produccion_pesquera_df, produccion_textil_df, produccion_plastica_df, produccion_otros_df]:
    df.set_index('Fecha', inplace=True)

# Manejar valores faltantes reemplazando NaN con 0
for df in [personal_activo_df, establecimientos_df, produccion_confeccionista_df, produccion_electronica_df,
          produccion_pesquera_df, produccion_textil_df, produccion_plastica_df, produccion_otros_df]:
    df.fillna(0, inplace=True)

# Crear una columna 'Total' sumando las columnas existentes en personal_activo_df
```

```
personal_activo_df['Total'] = personal_activo_df[['Confeccionistas', 'Electrónicas', 'Plásticas', 'Textiles', 'Pesqueras', 'Otras']].sur
```

## ▼ Estadísticas Descriptivas

A continuación, mostramos las estadísticas descriptivas de cada uno de los datasets para proporcionar una visión general de los datos.

```
print("Estadísticas Descriptivas de Personal Activo en Establecimientos Industriales")
print(personal_activo_df.describe())
print("\nEstadísticas Descriptivas de Establecimientos Industriales")
print(establecimientos_df.describe())
print("\nEstadísticas Descriptivas de Producción Industrial Confeccionista")
print(produccion_confeccionista_df.describe())
print("\nEstadísticas Descriptivas de Producción Industrial Electrónica")
print(produccion_electronica_df.describe())
print("\nEstadísticas Descriptivas de Producción Industrial Pesquera")
print(produccion_pesquera_df.describe())
print("\nEstadísticas Descriptivas de Producción Industrial Textil")
print(produccion_textil_df.describe())
print("\nEstadísticas Descriptivas de Producción Industrial Plástica")
print(produccion_plastica_df.describe())
print("\nEstadísticas Descriptivas de Producción Industrial Otros")
print(produccion_otros_df.describe())
```

### ↩ Estadísticas Descriptivas de Personal Activo en Establecimientos Industriales

	Año	Confeccionistas	Electrónicas	Plásticas	Textiles \
count	288.000000	288.000000	288.000000	288.000000	288.000000
mean	1984.388889	411.135417	6.397694	642.027778	578.840278
std	236.011489	148.665231	3.484517	169.477222	145.714444
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2006.000000	300.000000	3.366000	634.250000	557.750000
50%	2012.000000	393.000000	6.271000	705.500000	623.000000
75%	2018.000000	526.500000	9.171750	741.000000	660.000000
max	2024.000000	767.000000	13.820000	838.000000	776.000000

	Pesqueras	Otras	Unnamed: 8	Unnamed: 9	Total
count	288.000000	288.000000	288.0	288.0	288.000000
mean	377.572917	302.104167	0.0	0.0	2318.078250
std	134.980660	171.723533	0.0	0.0	578.354667
min	0.000000	0.000000	0.0	0.0	0.000000
25%	258.250000	122.750000	0.0	0.0	2157.685750
50%	445.500000	391.000000	0.0	0.0	2499.077000
75%	486.000000	446.000000	0.0	0.0	2707.212750
max	557.000000	579.000000	0.0	0.0	3057.065000

### Estadísticas Descriptivas de Establecimientos Industriales

	Año	Total	Confeccionistas	Electrónicas	Plásticas \
count	291.000000	291.000000	291.000000	291.000000	291.000000
mean	1963.931271	49.298969	5.532646	21.924399	7.580756
std	308.935428	9.242889	1.154362	5.089657	1.504994
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2006.000000	47.000000	5.000000	20.000000	7.000000
50%	2012.000000	50.000000	6.000000	22.000000	7.000000
75%	2018.000000	55.000000	6.000000	26.000000	9.000000
max	2024.000000	57.000000	7.000000	28.000000	9.000000

	Textiles	Pesqueras	Otras
count	291.000000	291.000000	291.000000
mean	7.697595	2.707904	3.051546
std	1.291068	0.986411	1.265220
min	0.000000	0.000000	0.000000
25%	8.000000	2.000000	2.000000
50%	8.000000	3.000000	3.000000
75%	8.000000	3.000000	4.000000
max	8.000000	5.000000	6.000000

### Estadísticas Descriptivas de Producción Industrial Confeccionista

	Año	Total (unidades)	Colcha	Cubrecama	Frazada \
count	145.000000	145.000000	145.000000	145.000000	145.000000
mean	1948.744828	316.392731	71.212407	96.110710	65.230634
std	369.569640	124.866559	142.102195	232.371689	186.271861
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2015.000000	249.348000	5.000000	0.000000	0.000000
50%	2018.000000	319.524000	23.355000	0.000000	3.468000
75%	2021.000000	399.137000	57.000000	3.000000	15.000000
max	2024.000000	610.624000	839.000000	975.000000	986.000000

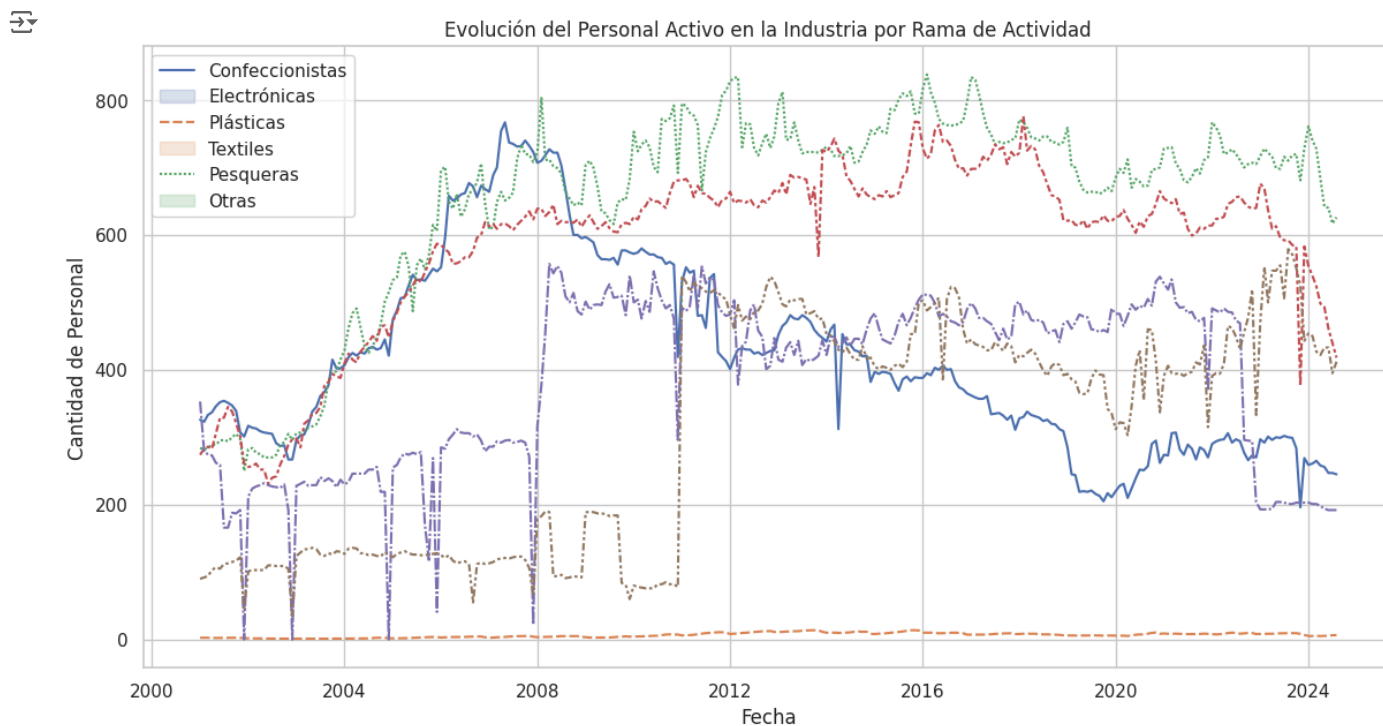
	Funda	Juego de Sabana	Juego de Toalla	Manta	Mantel \
count	145.000000	145.000000	145.000000	145.000000	145.000000
mean	30.734607	197.557807	40.980834	34.334621	51.488600
std	22.563961	84.803412	145.700623	123.659888	147.102543
min	0.000000	0.000000	0.000000	0.000000	0.000000

## Visualización de Datos

En esta sección, graficamos la evolución del personal activo, el número de establecimientos industriales y la producción de productos específicos a lo largo del tiempo.

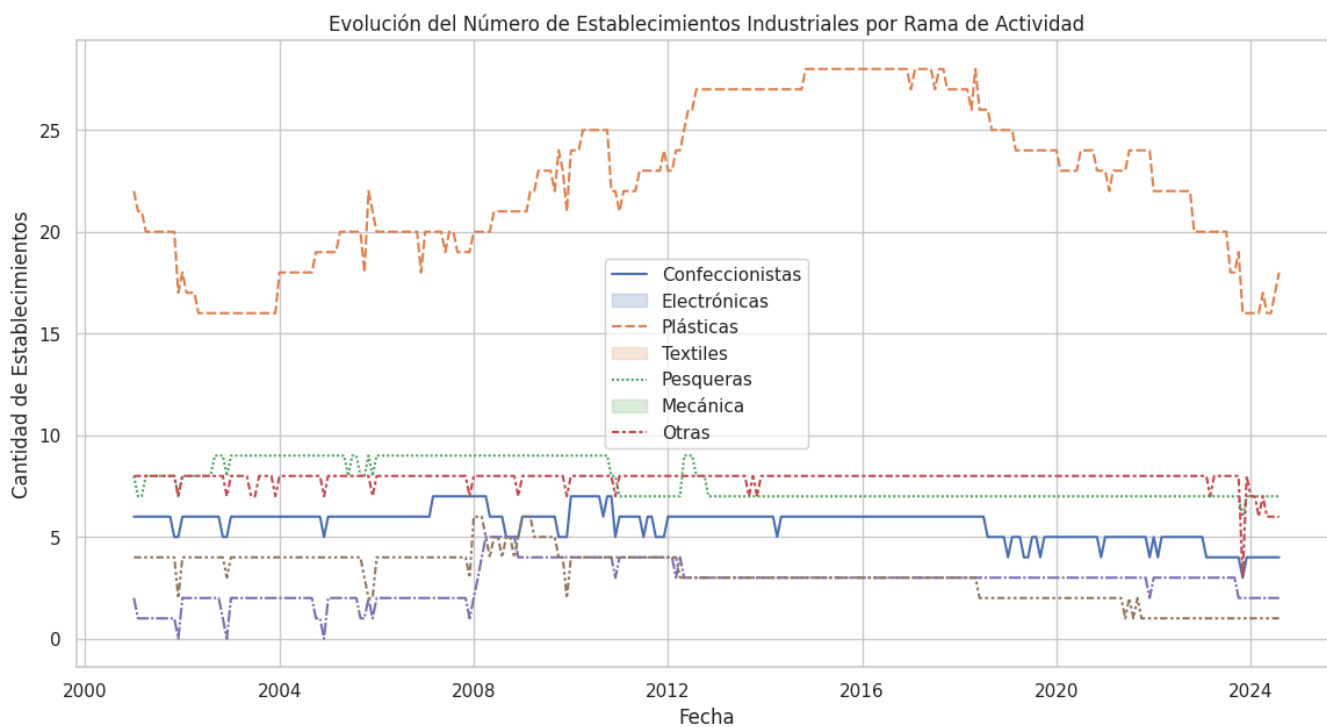
### Evolución del Personal Activo en la Industria por Rama de Actividad

```
plt.figure(figsize=(14, 7))
sns.lineplot(data=personal_activo_df[['Confeccionistas', 'Electrónicas', 'Plásticas', 'Textiles', 'Pesqueras', 'Otras']])
plt.title('Evolución del Personal Activo en la Industria por Rama de Actividad')
plt.xlabel('Fecha')
plt.ylabel('Cantidad de Personal')
plt.legend(labels=['Confeccionistas', 'Electrónicas', 'Plásticas', 'Textiles', 'Pesqueras', 'Otras'])
plt.show()
```



### Evolución del Número de Establecimientos Industriales por Rama de Actividad

```
plt.figure(figsize=(14, 7))
sns.lineplot(data=establecimientos_df[['Confeccionistas', 'Electrónicas', 'Plásticas', 'Textiles', 'Pesqueras', 'Mecánica', 'Otras']])
plt.title('Evolución del Número de Establecimientos Industriales por Rama de Actividad')
plt.xlabel('Fecha')
plt.ylabel('Cantidad de Establecimientos')
plt.legend(labels=['Confeccionistas', 'Electrónicas', 'Plásticas', 'Textiles', 'Pesqueras', 'Mecánica', 'Otras'])
plt.show()
```



## ✓ Producción de Televisores en el Sector Electrónico a lo Largo del Tiempo

Se pueden mostrar muchos ejemplos pero por ser parte del sector industrial de las electrónicas, se elije este ejemplo.

```
plt.figure(figsize=(14, 7))
sns.lineplot(data=produccion_electronica_df['Televisor'])
plt.title('Producción de Televisores en el Sector Electrónico a lo Largo del Tiempo')
plt.xlabel('Fecha')
plt.ylabel('Cantidad Producida')
plt.show()
```

