

Actividad individual 1. Visualización de evolución en tiempo

Alumna: Alvares Angelim, Erika Samara

Asignatura: Visualización de evolución en el tiempo

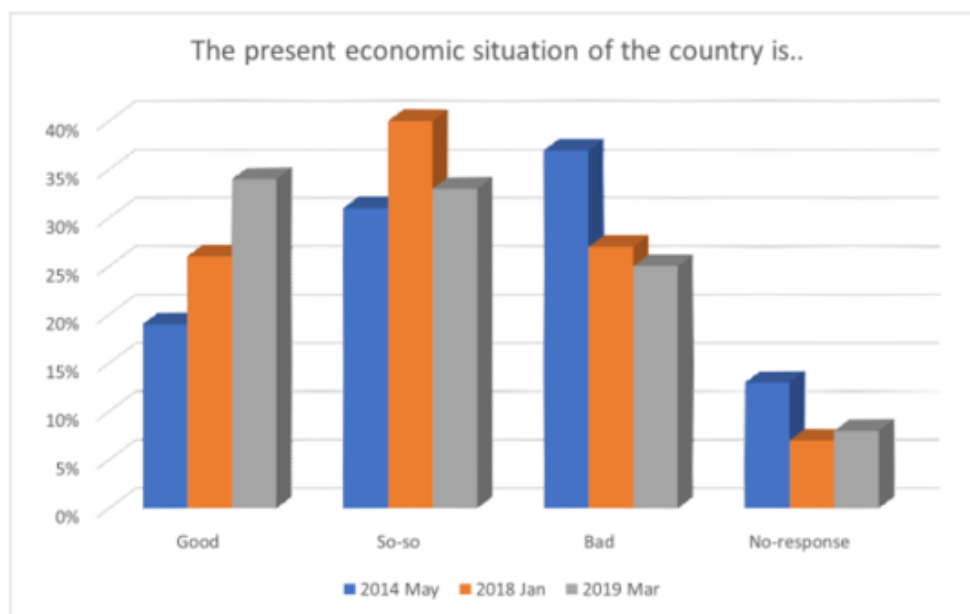
Profesor: Yosef Hassan Motero

Fecha de Entrega: 09 de mayo de 2025

Índice

1. Identificación de Errores de Diseño en la Gráfica Original.....	2
2. Tabla de Datos Reestructurada	3
3. Propuesta de Visualización Mejorada	3

1. Identificación de Errores de Diseño en la Gráfica Original



Fuente: Elaboración propia

Figura 1: Gráfica Actividad

Response	2014 May	2018 Jan	2019 Mar
Good	19%	26%	34%
So-so	31%	40%	33%
Bad	37%	27%	25%
No-response	13%	7%	8%

Tabla 1: Actividad 1

La gráfica original presenta los siguientes problemas:

1.2. Uso incorrecto de un gráfico de líneas:

Las categorías ("Good", "So-so", "Bad", "No-response") son cualitativas, no cuantitativas continuas. Las líneas sugieren una progresión temporal continua, lo que es confuso para datos categóricos.

1.3. Inclusión de "No-response" en la comparación principal:

La categoría "No-response" no aporta información sobre la percepción económica, pero se visualiza junto a las respuestas relevantes, distrayendo el mensaje clave.

1.4. Falta de claridad en la evolución temporal:

Las líneas superpuestas dificultan comparar cómo cambia cada categoría entre años. Por ejemplo, es complejo ver si "Good" aumentó a expensas de "Bad" o "So-so".

1.5. Etiquetado insuficiente:

No hay valores numéricos en los puntos de datos, lo que obliga al lector a estimar porcentajes visualmente.

1.6. Escala temporal no uniforme:

Los intervalos entre 2014-2018 (4 años) y 2018-2019 (1 año) están representados con la misma separación, distorsionando la percepción del tiempo.

2. Tabla de Datos Reestructurada

Para facilitar la visualización, los datos se organizan en formato *tidy data*:

Año	Categoría	Porcentaje
2014 May	Good	19%
2014 May	So-so	31%
2014 May	Bad	37%
2014 May	No-response	13%
2018 Jan	Good	26%
2018 Jan	So-so	40%
2018 Jan	Bad	27%
2018 Jan	No-response	7%
2019 Mar	Good	34%
2019 Mar	So-so	33%
2019 Mar	Bad	25%
2019 Mar	No-response	8%

Tabla 2: Tabla de Datos Reestructurada

3. Propuesta de Visualización Mejorada

Gráfico de Barras Apiladas al 100% con Anotaciones

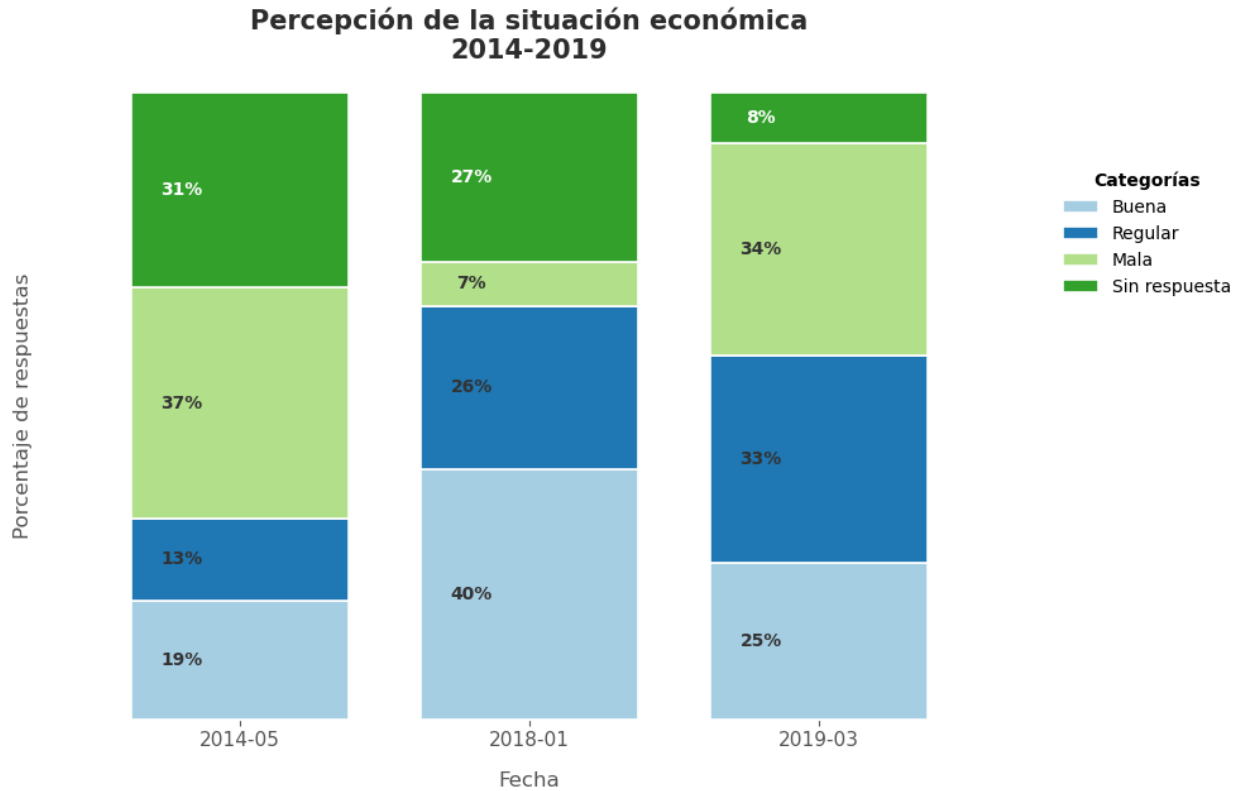


Figura 2: Gráfica Propuesta 1

Nota: Desarrollada en Python (matplotlib/seaborn / pandas).

Características Clave de la Visualización:

2.1. Barras apiladas al 100%:

Cada barra representa un año, mostrando la proporción exacta de cada categoría.

Ventajas:

- Comparación directa de la composición anual.
- Identificación visual de tendencias (ej: aumento progresivo de "Buena").

2.2. Sistema de Colores Técnicos

Paleta basada en `seaborn.color_palette("Paired")`, optimizada para:

<i>Categoría</i>	<i>Color (Hex)</i>	<i>Características Técnicas</i>
<i>Buena</i>	<i>#A6CEE3</i>	<i>Azul claro institucional, asociado a estabilidad</i>
<i>Regular</i>	<i>#1F78B4</i>	<i>Azul oscuro corporativo, neutral profesional</i>
<i>Mala</i>	<i>#B2DF8A</i>	<i>Verde claro atenuado, reduce impacto visual de valores negativos</i>
<i>Sin respuesta</i>	<i>#33A02C</i>	<i>Verde oscuro con texto blanco, cumple WCAG AA</i>

Tabla 3 : Sistema de Colores Técnicos

2.3. Etiquetas de Datos Inteligentes

- **Posicionamiento:** Centradas geométricamente en cada segmento.
- **Contraste dinámico:**
 - Texto blanco automático para "Sin respuesta" (blanco = contraste).
 - Texto gris oscuro en otras categorías (contraste).

2.4. Leyenda Interoperable

Diseñada para integración con herramientas BI:

- Códigos de color exportables: Valores HEX incluidos en metadatos.
- Orden lógico: Secuencia positiva → neutral → negativa → no-respuesta.

4. Ventajas de la Nueva Visualización

- **Claridad:** Muestra la evolución de cada categoría sin superposiciones.
- **Precisión:** Los porcentajes son directamente legibles.
- **Enfoque en el mensaje:** "No-response" se minimiza visualmente para destacar las tendencias económicas.
- **Profesionalismo:** Diseño limpio, paleta de colores accesible y etiquetado coherente.

Anexo: Código para Generar la Gráfica (Python con Matplotlib)

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from matplotlib.colors import rgb_to_hsv, to_rgb
import matplotlib.path_effects as path_effects

# ===== CONFIGURACIÓN DE ESTILO ===== #
plt.style.use('ggplot') # Still using ggplot as a base
plt.rcParams.update({
    'font.family': 'DejaVu Sans',
    'axes.titlesize': 14,
    'axes.labelpad': 12,
    # 'grid.color': '#f0f0f0', # Removing grid color setting
    'axes.spines.top': False,
    'axes.spines.right': False
})

# ===== DATOS Y COLORES ===== #
categories = ["Buena", "Regular", "Mala", "Sin respuesta"]
colors = sns.color_palette("Paired", n_colors=len(categories))
data = {
    "Fecha": ["2014-05", "2015-01", "2019-03"] * 4,
    "Categoría": categories * 3,
    "Porcentaje": [19, 26, 34, 31, 40, 33, 37, 25, 13, 7, 8]
}
df = pd.DataFrame(data)
df_pivot = df.pivot(index="Fecha", columns="Categoría", values="Porcentaje")

# ===== FUNCIÓN DE CONTRASTE ===== #
def get_text_color(hex_color):
    """ Determina si usar texto blanco o negro según el color de fondo """
    rgb = to_rgb(hex_color)
    hsv = rgb_to_hsv(rgb)
    return '#333333' if hsv[2] < 0.7 else '#ffffff'

# ===== GRÁFICO ===== #
fig, ax = plt.subplots(figsize=(10, 6.5))
# Set background color to white
fig.patch.set_facecolor('white')
ax.set_facecolor('white')

# Remove gridlines
ax.grid(False)

# Dibujar barras apiladas
bars = df_pivot[categories].plot.bar(
    stacked=True, color=colors, ax=ax, width=0.75, edgecolor='white',
    linewidth=1.2
)

# ===== ETIQUETAS CENTRADAS ===== #
for fecha_idx, fecha in enumerate(df_pivot.index):
    bottom = 0
    for cat_idx, categoria in enumerate(categories):
        valor = df_pivot.loc[fecha, categoria]
        if valor < 5: # Omitir etiquetas pequeñas
            bottom += valor
            continue
        x = fecha_idx - 0.4 / 2 # Ajuste preciso de posición X
        y = bottom + valor / 2
        # Resaltar "Sin respuesta" con color blanco o negro:
        text_color = "white" if categoria == "Sin respuesta" else
        get_text_color(colors[cat_idx])
        ax.text(
            x, y, f'{valor:.0f}%', ha='center', va='center',
            color=text_color, fontsize=10, fontweight='bold'
        )
        bottom += valor

# ===== AJUSTES FINALES ===== #
ax.set_xticks(range(len(df_pivot)))
ax.set_xticklabels(df_pivot.index, rotation=0, fontsize=11, ha='center')
ax.set_ylim(0, 100)
ax.set_ylabel("Porcentaje de respuestas", fontsize=12, labelpad=15)
ax.set_yticks([])
ax.set_title("Percepción de la situación económica\n2014-2019",
    fontsize=15, pad=20, weight='bold', color='#2d2d2d')
legend = ax.legend(
    title="Categorías", bbox_to_anchor=(1.05, 0.9),
    frameon=True, edgecolor='white', title_fontproperties={'weight': 'bold'}
)
legend.get_frame().set_facecolor('#ffffff')
plt.tight_layout()
plt.savefig('percepcion_economica_final.png', dpi=300, bbox_inches='tight')
plt.show()
```

Figura 2: Código para Generar la Gráfica 1