

Aeropuertos y Combustibles - Ingresos y Gastos

May 1, 2024

1 Aeropuertos y Combustibles - Ingresos y Gastos

2 Generación de Datos Sintéticos

Creado por:

- V. D. Betancourt

2.1 Introducción

2.1.1 Descripción

Este proyecto permite generar **datos sintéticos** para los **Ingresos y Gastos** correspondientes a **Aeropuertos y Combustibles**.

2.2 Generación de Datos Sintéticos

Los **datos sintéticos** generados constan de **15 variables (columnas)**, que son:

- '**Fecha**': Son fechas parametrizables mensuales, en este caso, generadas para el período: '**2023-01-31**' al '**2024-03-31**'. Cabe señalar que para generar predicciones, el modelo requerirá contar con al menos 2 fechas históricas.
- '**División**': Son 2 categorías: '**Aeropuertos**' y '**Combustibles**'.
- '**Concepto**': Son 2 categorías para cada '**División**': '**Ingresos Cobrados**' y '**Gastos**'.
- '**Subconcepto**': Son categorías que dependen de la '**División**' y el '**Concepto**'.
 - Para Aeropuertos e Ingresos cobrados: '**Arrendamiento**', '**Servicios Aeroportuarios**', '**Tarifa de Uso Aeroportuario (TUA)**'
 - Para Aeropuertos y Gastos: '**Servicios Personales**', '**Materiales**'
 - Para Combustibles e Ingresos Cobrados: '**Almacenamiento**', '**Expendio**'
 - Para Combustibles y Gastos: '**Servicios Personales**', '**Materiales**', '**Servicios Generales**'
- **11 columnas** más correspondientes a los nombres de cada aeropuerto.
 - Se ha creado un catálogo (diccionario) para los nombres de los aeropuertos ('**nombres_aeropuertos**') con la flexibilidad de poder adecuarlo con los nombres reales cuando se tenga dicha información.

```
[15]: import pandas as pd
import numpy as np
from datetime import datetime, timedelta

# Función para generar fechas mensuales
def generar_fechas(inicio, fin):
    return pd.date_range(start=inicio, end=fin, freq='M')

# Crear las fechas para el DataFrame
fechas = generar_fechas('2023-01-31', '2024-03-31')

# Información de las combinaciones División-Concepto-Subconcepto
data_info = [
    ("Aeropuertos", "Ingresos Cobrados", "Arrendamiento"),
    ("Aeropuertos", "Ingresos Cobrados", "Servicios Aeroportuarios"),
    ("Aeropuertos", "Ingresos Cobrados", "Tarifa de Uso Aeroportuario (TUA)"),
    ("Aeropuertos", "Gastos", "Servicios Personales"),
    ("Aeropuertos", "Gastos", "Materiales"),
    ("Aeropuertos", "Gastos", "Servicios Generales"),
    ("Combustibles", "Ingresos Cobrados", "Almacenamiento"),
    ("Combustibles", "Ingresos Cobrados", "Expendio"),
    ("Combustibles", "Gastos", "Servicios Personales"),
    ("Combustibles", "Gastos", "Materiales"),
    ("Combustibles", "Gastos", "Servicios Generales")
]

# Crear el DataFrame
df = pd.DataFrame({
    "Fecha": np.repeat(fechas, len(data_info)),
    "División": np.tile([x[0] for x in data_info], len(fechas)),
    "Concepto": np.tile([x[1] for x in data_info], len(fechas)),
    "Subconcepto": np.tile([x[2] for x in data_info], len(fechas))
})

# Diccionario para nombres de aeropuertos
nombres_aeropuertos = {
    'A1': 'Primer_Aeropuerto',
    'A2': 'Segundo_Aeropuerto',
    'A3': 'Tercer_Aeropuerto',
    'A4': 'Cuarto_Aeropuerto',
    'A5': 'Quinto_Aeropuerto',
    'A6': 'Sexto_Aeropuerto',
    'A7': 'Séptimo_Aeropuerto',
    'A8': 'Octavo_Aeropuerto',
    'A9': 'Noveno_Aeropuerto',
    'A10': 'Décimo_Aeropuerto',
    'A11': 'Undécimo_Aeropuerto'
}
```

```

}

# Generar datos sintéticos y renombrar columnas utilizando una distribución
↳ uniforme
for key, nombre in nombres_aeropuertos.items():
    df[nombre] = np.random.uniform(low=20000, high=2000000, size=len(df))

# Eliminar las columnas de aeropuertos con etiquetas originales
df.drop(columns=[f'A{i}' for i in range(1, 12)], inplace=True, errors='ignore')

# Configuración para que Pandas muestre los números flotantes sin notación
↳ científica
pd.options.display.float_format = '{:.2f}'.format

# Guardar el DataFrame como CSV
df.to_csv('datos_sinteticos_aeropuertos_combustibles.csv', index=False)

df.head(30)

```

```

[15]:

```

	Fecha	División	Concepto	\
0	2023-01-31	Aeropuertos	Ingresos Cobrados	
1	2023-01-31	Aeropuertos	Ingresos Cobrados	
2	2023-01-31	Aeropuertos	Ingresos Cobrados	
3	2023-01-31	Aeropuertos	Gastos	
4	2023-01-31	Aeropuertos	Gastos	
5	2023-01-31	Aeropuertos	Gastos	
6	2023-01-31	Combustibles	Ingresos Cobrados	
7	2023-01-31	Combustibles	Ingresos Cobrados	
8	2023-01-31	Combustibles	Gastos	
9	2023-01-31	Combustibles	Gastos	
10	2023-01-31	Combustibles	Gastos	
11	2023-02-28	Aeropuertos	Ingresos Cobrados	
12	2023-02-28	Aeropuertos	Ingresos Cobrados	
13	2023-02-28	Aeropuertos	Ingresos Cobrados	
14	2023-02-28	Aeropuertos	Gastos	
15	2023-02-28	Aeropuertos	Gastos	
16	2023-02-28	Aeropuertos	Gastos	
17	2023-02-28	Combustibles	Ingresos Cobrados	
18	2023-02-28	Combustibles	Ingresos Cobrados	
19	2023-02-28	Combustibles	Gastos	
20	2023-02-28	Combustibles	Gastos	
21	2023-02-28	Combustibles	Gastos	
22	2023-03-31	Aeropuertos	Ingresos Cobrados	
23	2023-03-31	Aeropuertos	Ingresos Cobrados	
24	2023-03-31	Aeropuertos	Ingresos Cobrados	
25	2023-03-31	Aeropuertos	Gastos	
26	2023-03-31	Aeropuertos	Gastos	

27 2023-03-31 Aeropuertos Gastos
 28 2023-03-31 Combustibles Ingresos Cobrados
 29 2023-03-31 Combustibles Ingresos Cobrados

	Subconcepto	Primer_Aeropuerto	Segundo_Aeropuerto	\
0	Arrendamiento	1503526.86	1952592.18	
1	Servicios Aeroportuarios	391494.18	80115.21	
2	Tarifa de Uso Aeroportuario (TUA)	561625.45	1229867.49	
3	Servicios Personales	1080515.62	948179.56	
4	Materiales	351753.87	1028514.50	
5	Servicios Generales	68380.73	1260306.91	
6	Almacenamiento	945446.73	1939694.96	
7	Expendio	80638.27	1333452.96	
8	Servicios Personales	125770.93	680434.28	
9	Materiales	1175966.07	1265065.88	
10	Servicios Generales	1117400.29	1353264.29	
11	Arrendamiento	1088276.11	1030220.83	
12	Servicios Aeroportuarios	1839613.11	1338855.30	
13	Tarifa de Uso Aeroportuario (TUA)	1344047.73	1261413.86	
14	Servicios Personales	324363.52	71494.49	
15	Materiales	1556785.71	908894.84	
16	Servicios Generales	1273087.75	849115.05	
17	Almacenamiento	662132.50	1760154.67	
18	Expendio	41281.09	1592087.66	
19	Servicios Personales	1537131.84	1495892.27	
20	Materiales	414384.13	891259.62	
21	Servicios Generales	1640658.27	992095.17	
22	Arrendamiento	388004.57	802333.12	
23	Servicios Aeroportuarios	524234.44	1827833.65	
24	Tarifa de Uso Aeroportuario (TUA)	701856.61	22821.10	
25	Servicios Personales	259436.33	553823.82	
26	Materiales	172431.79	388613.35	
27	Servicios Generales	1228081.31	1750943.54	
28	Almacenamiento	1725889.45	1062112.34	
29	Expendio	361040.16	1199907.83	

	Tercer_Aeropuerto	Cuarto_Aeropuerto	Quinto_Aeropuerto	Sexto_Aeropuerto	\
0	259583.38	710059.62	1989462.97	1282443.15	
1	1169280.12	1866281.52	190737.27	1881531.19	
2	444564.46	1361868.98	1174631.82	327670.45	
3	204986.80	653884.78	659083.32	567937.45	
4	1036656.23	1119003.39	1061807.04	994385.20	
5	175702.16	1329135.21	1169470.84	545189.19	
6	1639587.84	1059348.36	634997.62	1419655.03	
7	569856.23	442343.19	826381.46	1275189.05	
8	765010.42	1045466.74	1769779.60	1910827.14	
9	697259.86	1627290.87	481851.59	508524.60	

10	972648.95	1335765.41	1042155.33	1715989.61
11	1339079.21	1700259.50	1080697.53	847727.89
12	39705.01	1455799.83	1228450.13	811589.50
13	1634148.63	1247536.97	89064.63	1017307.59
14	675646.01	843174.97	113253.03	621117.44
15	1846959.64	1258275.42	488482.19	667054.35
16	330648.06	456594.63	602764.41	751507.91
17	257844.11	957038.50	938138.43	1460131.31
18	866177.91	714392.84	956925.60	1752913.38
19	1798906.75	1498961.40	1672287.13	871315.54
20	49161.09	1866467.51	603747.81	442966.42
21	1281460.17	525281.96	1810374.76	1523445.62
22	810941.76	201109.98	217368.35	1041951.58
23	126525.86	1794413.74	75979.56	919857.43
24	1026401.29	431992.89	817940.70	245278.02
25	1381725.43	1066255.08	1961823.41	1665448.89
26	1589078.82	718978.21	1231892.15	1506607.27
27	1766118.42	1025003.83	1155469.05	1222884.42
28	1963803.03	1920452.65	44115.44	1434053.00
29	1481095.35	886220.36	1047040.41	305986.67

	Séptimo_Aeropuerto	Octavo_Aeropuerto	Noveno_Aeropuerto	\
0	307542.17	1528532.26	1211803.53	
1	1070427.07	36709.48	1096441.54	
2	1485853.61	1553398.86	88384.45	
3	1584968.41	1626445.13	840376.58	
4	591456.87	934640.65	146351.13	
5	1651436.13	1779770.40	218917.09	
6	1156347.84	78072.38	1927320.00	
7	952817.99	1589864.10	84784.14	
8	71457.94	255542.58	1855262.09	
9	577405.11	1320673.40	1158173.96	
10	1077070.45	1097016.98	945623.18	
11	1344505.92	170662.26	226010.42	
12	1959063.73	185871.02	385277.79	
13	1200420.37	1910955.76	1409367.68	
14	936036.31	882630.64	1396714.42	
15	1096237.50	232558.61	1978676.45	
16	151922.92	1591566.57	1145353.48	
17	72237.97	443439.80	219621.46	
18	1749318.20	272598.98	1271329.45	
19	1579639.50	1301140.23	1949366.80	
20	1861695.39	1740978.54	812001.67	
21	84182.00	555718.67	1829295.43	
22	861499.18	622423.66	854971.59	
23	562781.99	72775.26	192251.26	
24	1443980.94	1592690.71	553039.09	

25	1992105.56	1836485.33	1698182.77
26	767559.89	181658.13	519230.66
27	197216.39	140037.29	858950.76
28	1675702.38	1820349.54	594084.20
29	694024.01	1838951.76	1363650.27

	Décimo_Aeropuerto	Undécimo_Aeropuerto
0	1346739.11	1669491.64
1	1059985.10	275591.92
2	476454.95	501530.79
3	1993233.44	1832404.33
4	871509.41	1441273.27
5	1925358.54	249131.18
6	877805.04	107123.19
7	554334.05	833051.71
8	118362.66	858205.37
9	869027.08	1290163.56
10	960917.48	976525.24
11	1286747.24	924888.08
12	1256781.03	397989.25
13	1162923.46	847566.62
14	624588.41	1121481.91
15	1039340.21	135376.19
16	720141.05	923590.15
17	646153.04	719106.21
18	1861227.71	122189.31
19	157247.59	1896094.35
20	1854505.69	1235643.58
21	1282556.74	525984.33
22	1415980.09	850918.97
23	1707398.66	1043876.02
24	548347.74	727561.82
25	1584598.81	1762051.54
26	1657509.89	1299449.87
27	1804302.31	91804.99
28	915603.19	671770.93
29	512532.50	419004.48

```
[16]: # Filas y Columnas
df.shape
```

```
[16]: (165, 15)
```

```
[17]: # Info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 165 entries, 0 to 164

Data columns (total 15 columns):

#	Column	Non-Null Count	Dtype
0	Fecha	165 non-null	datetime64[ns]
1	División	165 non-null	object
2	Concepto	165 non-null	object
3	Subconcepto	165 non-null	object
4	Primer_Aeropuerto	165 non-null	float64
5	Segundo_Aeropuerto	165 non-null	float64
6	Tercer_Aeropuerto	165 non-null	float64
7	Cuarto_Aeropuerto	165 non-null	float64
8	Quinto_Aeropuerto	165 non-null	float64
9	Sexto_Aeropuerto	165 non-null	float64
10	Séptimo_Aeropuerto	165 non-null	float64
11	Octavo_Aeropuerto	165 non-null	float64
12	Noveno_Aeropuerto	165 non-null	float64
13	Décimo_Aeropuerto	165 non-null	float64
14	Undécimo_Aeropuerto	165 non-null	float64

dtypes: datetime64[ns](1), float64(11), object(3)

memory usage: 19.5+ KB

2.3 Carga de Datos

```
[18]: import pandas as pd

# Cargar los datos
df = pd.read_csv('datos_sinteticos_aeropuertos_combustibles.csv')
```

```
[19]: # Asegurar formato en campo Fecha
df['Fecha'] = pd.to_datetime(df['Fecha'])
```

2.4 Exploración de los Datos

2.4.1 Datos Último Trimestre

```
[20]: # Definir el rango de fechas
fecha_inicio = '2024-01-31'
fecha_fin = '2024-03-31'

# Crear una máscara booleana
mascara = (df['Fecha'] >= fecha_inicio) & (df['Fecha'] <= fecha_fin)

# Aplicar la máscara al DataFrame para obtener las filas en el rango de fechas
df_filtrado = df[mascara]
df_filtrado
```

[20] :

	Fecha	División	Concepto	\
132	2024-01-31	Aeropuertos	Ingresos Cobrados	
133	2024-01-31	Aeropuertos	Ingresos Cobrados	
134	2024-01-31	Aeropuertos	Ingresos Cobrados	
135	2024-01-31	Aeropuertos	Gastos	
136	2024-01-31	Aeropuertos	Gastos	
137	2024-01-31	Aeropuertos	Gastos	
138	2024-01-31	Combustibles	Ingresos Cobrados	
139	2024-01-31	Combustibles	Ingresos Cobrados	
140	2024-01-31	Combustibles	Gastos	
141	2024-01-31	Combustibles	Gastos	
142	2024-01-31	Combustibles	Gastos	
143	2024-02-29	Aeropuertos	Ingresos Cobrados	
144	2024-02-29	Aeropuertos	Ingresos Cobrados	
145	2024-02-29	Aeropuertos	Ingresos Cobrados	
146	2024-02-29	Aeropuertos	Gastos	
147	2024-02-29	Aeropuertos	Gastos	
148	2024-02-29	Aeropuertos	Gastos	
149	2024-02-29	Combustibles	Ingresos Cobrados	
150	2024-02-29	Combustibles	Ingresos Cobrados	
151	2024-02-29	Combustibles	Gastos	
152	2024-02-29	Combustibles	Gastos	
153	2024-02-29	Combustibles	Gastos	
154	2024-03-31	Aeropuertos	Ingresos Cobrados	
155	2024-03-31	Aeropuertos	Ingresos Cobrados	
156	2024-03-31	Aeropuertos	Ingresos Cobrados	
157	2024-03-31	Aeropuertos	Gastos	
158	2024-03-31	Aeropuertos	Gastos	
159	2024-03-31	Aeropuertos	Gastos	
160	2024-03-31	Combustibles	Ingresos Cobrados	
161	2024-03-31	Combustibles	Ingresos Cobrados	
162	2024-03-31	Combustibles	Gastos	
163	2024-03-31	Combustibles	Gastos	
164	2024-03-31	Combustibles	Gastos	

	Subconcepto	Primer_Aeropuerto	Segundo_Aeropuerto	\
132	Arrendamiento	798455.08	432583.10	
133	Servicios Aeroportuarios	1319080.24	1245780.77	
134	Tarifa de Uso Aeroportuario (TUA)	1905113.70	1784548.74	
135	Servicios Personales	305488.71	1901804.93	
136	Materiales	95147.28	1537416.96	
137	Servicios Generales	1674673.58	1290205.26	
138	Almacenamiento	23831.96	892398.11	
139	Expendio	33568.36	934339.02	
140	Servicios Personales	1536905.11	1227545.41	
141	Materiales	1049407.16	1365718.50	
142	Servicios Generales	738141.80	1785166.22	

143	Arrendamiento	694194.56	877866.93
144	Servicios Aeroportuarios	1069106.47	1887458.43
145	Tarifa de Uso Aeroportuario (TUA)	281495.65	1792483.13
146	Servicios Personales	1171847.60	700179.38
147	Materiales	1293075.99	1109957.03
148	Servicios Generales	1452386.99	599585.99
149	Almacenamiento	1812064.05	1880815.43
150	Expendio	1122330.77	1471620.25
151	Servicios Personales	814240.72	158829.52
152	Materiales	696433.42	1293800.15
153	Servicios Generales	1266545.67	1666801.78
154	Arrendamiento	1294531.81	194050.37
155	Servicios Aeroportuarios	27515.69	1302798.68
156	Tarifa de Uso Aeroportuario (TUA)	1535170.22	800392.00
157	Servicios Personales	794116.12	1065608.05
158	Materiales	927076.26	353038.88
159	Servicios Generales	796395.66	1793197.67
160	Almacenamiento	311013.78	1429278.85
161	Expendio	206856.13	264631.92
162	Servicios Personales	1998644.78	1288137.54
163	Materiales	170453.11	1357169.39
164	Servicios Generales	1197198.54	30615.98

	Tercer_Aeropuerto	Cuarto_Aeropuerto	Quinto_Aeropuerto \
132	943707.24	1528556.53	1597195.83
133	228222.46	191996.35	143081.48
134	1688483.94	1987965.63	1946290.60
135	1332746.43	100236.04	1879992.02
136	125276.17	1211700.76	1897822.58
137	1273462.50	566257.66	1823116.10
138	195912.78	1906874.87	864042.25
139	1033561.33	229889.47	595414.27
140	961452.06	1476234.34	1375029.15
141	670995.62	1496339.79	1885902.96
142	1157338.65	713373.37	1538902.69
143	1622780.00	1883592.58	140844.92
144	1315331.03	500427.98	972801.02
145	875885.09	1563593.74	1255981.94
146	227752.30	1056403.70	780355.69
147	1240442.45	1367678.02	1789566.50
148	1642762.43	1117634.60	584991.40
149	524948.71	1935720.08	1791932.27
150	1999976.42	1912981.07	1010298.35
151	1137842.82	947587.83	1201856.54
152	915216.56	866966.85	472430.61
153	1159954.19	1831375.55	1326280.32
154	1784463.79	962642.73	769834.06

155	1523862.14	1815453.03	75057.61
156	1434458.37	614834.14	250902.61
157	1641083.30	1221434.91	78328.50
158	729701.89	1284327.41	394629.58
159	1303748.42	554769.77	510604.51
160	663317.05	1395254.73	1854039.07
161	824302.60	1113481.84	911342.12
162	755271.05	1315853.01	1611293.95
163	1533092.02	639656.80	361784.15
164	1890223.08	526829.69	1633938.40

	Sexto_Aeropuerto	Séptimo_Aeropuerto	Octavo_Aeropuerto \
132	1316482.25	261000.55	951329.41
133	1582510.41	558360.21	1522461.49
134	359068.82	838255.53	381030.63
135	926633.26	820740.67	670214.59
136	360061.57	1319146.41	1519279.34
137	350861.35	516691.17	746921.05
138	1076412.98	772678.60	1635106.70
139	1869145.44	1179955.22	1800466.03
140	1238318.53	1714491.89	709199.89
141	1010726.73	1223124.74	1205830.82
142	1847895.46	905468.11	1271619.45
143	339957.69	841167.55	1938439.49
144	825843.45	1456604.48	1922337.93
145	1577760.80	277654.63	131578.04
146	39739.00	1944923.44	403003.98
147	1983219.53	1908837.24	869465.15
148	630781.23	1524009.69	1706831.48
149	1090640.29	826934.13	501255.94
150	232296.41	121190.48	857954.84
151	1729786.13	1418777.18	766122.41
152	757737.70	912824.86	839952.83
153	1016146.54	1478141.68	883010.98
154	638202.09	249103.19	711768.78
155	1783277.86	1416474.84	700329.42
156	1566412.02	1017717.30	1428312.52
157	1233543.59	1265574.81	557710.29
158	405558.07	187011.68	131757.52
159	621797.72	314479.42	804342.40
160	1790907.36	131443.53	740101.68
161	815307.04	496301.67	833137.78
162	1671144.38	859681.08	328464.97
163	890853.30	758251.24	1192070.68
164	158989.67	1905915.25	891434.49

Noveno_Aeropuerto Décimo_Aeropuerto Undécimo_Aeropuerto

132	86413.89	637562.53	834116.62
133	418003.20	1812588.87	1468001.07
134	1794930.12	25548.28	1278760.59
135	1512795.20	1725132.57	464613.66
136	872508.96	1274413.04	452976.11
137	1434368.58	53305.16	1270236.35
138	1107498.21	508977.16	922473.64
139	1857002.02	257857.04	425927.30
140	165538.90	132266.93	958420.28
141	874382.17	50029.19	349910.64
142	1282465.49	446360.03	1933074.04
143	1211876.41	1728976.73	1269717.56
144	1822536.66	150199.42	1761569.40
145	817181.56	268529.63	1349692.06
146	1105147.94	1160076.66	450912.76
147	408181.47	570070.14	34484.69
148	1616530.11	839439.20	655891.64
149	1227737.75	1784728.62	712178.69
150	1080082.20	396929.52	1937679.89
151	161045.57	1437580.50	1429977.96
152	1919419.84	649093.21	1290643.39
153	1015792.60	1561747.41	842466.40
154	1128413.76	366848.75	1542315.21
155	1010481.18	1615370.32	261960.85
156	458492.15	1821687.32	414964.16
157	1157933.52	925315.22	459967.37
158	1702589.63	582844.73	306828.21
159	1251214.03	835030.14	236868.79
160	1840573.43	1725683.82	1876704.51
161	1927012.06	1073012.07	337828.41
162	494501.84	1823374.97	1590250.09
163	1400448.67	1848396.47	1162338.24
164	454212.12	248044.00	1171041.06

2.4.2 Aeropuertos: Ingresos Cobrados

Gráficos de Barras

```
[28]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Suponiendo que 'df' es tu DataFrame
df['Fecha'] = pd.to_datetime(df['Fecha'])

# Filtrar por fechas, por 'Concepto' y asegurarse de que la 'División' sea
↳ "Aeropuertos"
mask_fecha = (df['Fecha'] >= '2024-01-31') & (df['Fecha'] <= '2024-03-31')
```

```

mask_concepto = df['Concepto'] == 'Ingresos Cobrados'
mask_division = df['División'] == 'Aeropuertos'
df_filtrado = df.loc[mask_fecha & mask_concepto & mask_division]

# Transformar de formato ancho a largo
df_melt = df_filtrado.melt(id_vars=['Fecha', 'División', 'Concepto'],
    ↳ 'Subconcepto'],
                           value_vars=df.columns[4:15],
                           var_name='Aeropuerto', value_name='Ingresos')

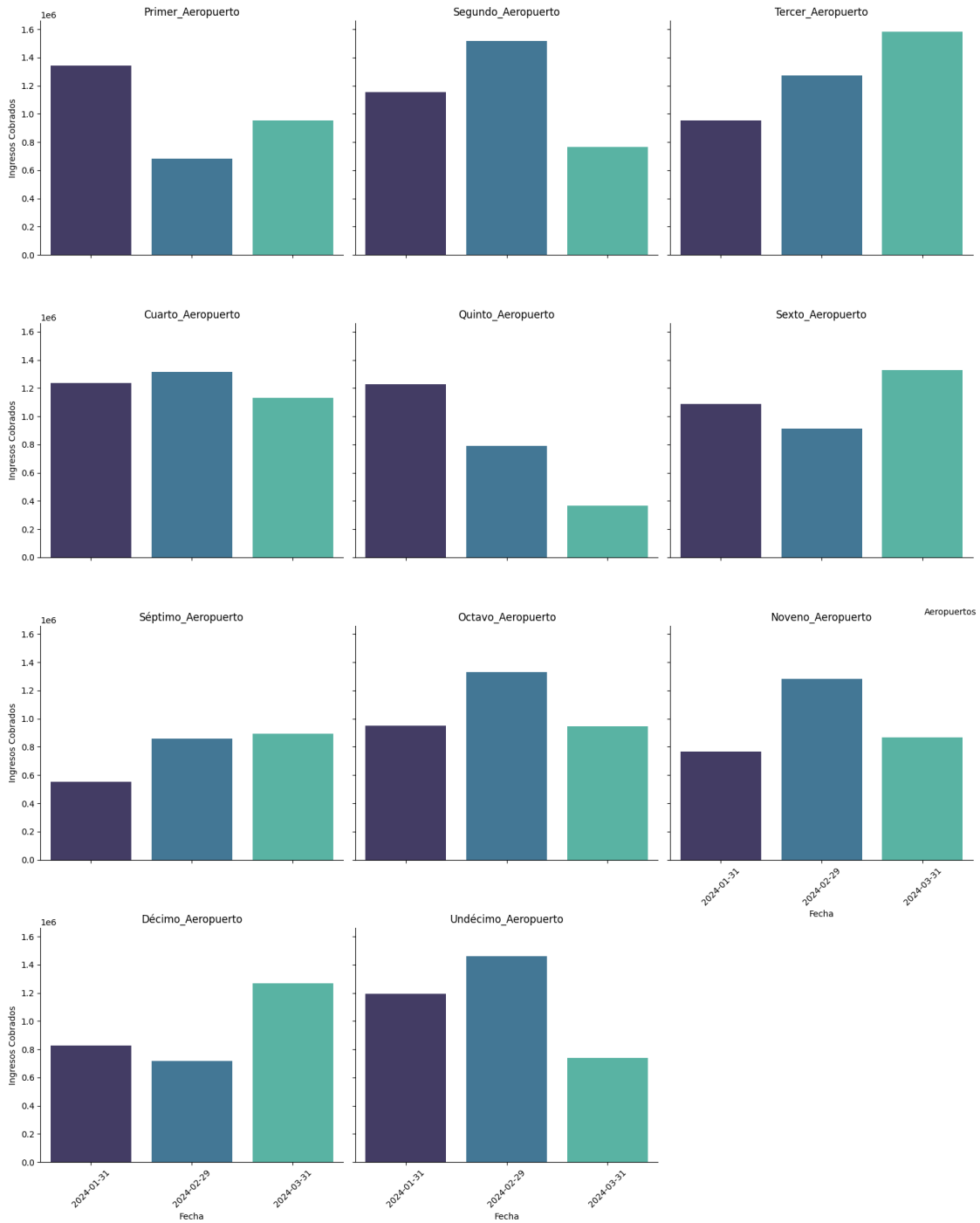
# Crear un FacetGrid para graficar un barplot para cada aeropuerto
g = sns.FacetGrid(df_melt, col='Aeropuerto', col_wrap=3, height=5, aspect=1)
g.map_dataframe(sns.barplot, x='Fecha', y='Ingresos', hue='Fecha',
    ↳ palette='mako', errorbar=None)

# Ajustar títulos y etiquetas
g.set_titles("{col_name}")
g.set_axis_labels("Fecha", "Ingresos Cobrados")
g.add_legend(title="Aeropuertos")

# Ajustar los títulos y etiquetas de los ejes
for ax in g.axes.flat:
    for label in ax.get_xticklabels():
        label.set_rotation(45)
    ax.set_title(ax.get_title(), fontsize=12)

plt.tight_layout()
plt.show()

```



Boxplots y Outliers Se grafican **Boxplots** para la identificación rápida de **Outliers**:

- **map_dataframe**: Se pasa **x='Fecha'** y **y='Ingresos'** como argumentos a la función **plot_box_strip** para especificar qué columnas del DataFrame se usan para los ejes X e Y en cada gráfico.

- **sns.boxplot**: Dibuja los boxplots. El parámetro **showfliers=False** oculta los outliers en el boxplot para evitar la duplicación visual, ya que los mostraremos con **stripplot**.
- **sns.stripplot**: Añade los puntos sobre el boxplot. **jitter=True** añade un pequeño desplazamiento horizontal a los puntos para hacerlos más fáciles de visualizar y evitar que se solapen.
- **plot_box_strip**: Función personalizada que acepta explícitamente argumentos **x** y **y**, que especifican las columnas del DataFrame a utilizar para los ejes del gráfico.

```
[30]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Asegurar que las fechas están en formato datetime
df['Fecha'] = pd.to_datetime(df['Fecha'])

# Filtrar por fechas, 'Concepto' y 'División'
mask_fecha = (df['Fecha'] >= '2024-01-31') & (df['Fecha'] <= '2024-03-31')
mask_concepto = df['Concepto'] == 'Ingresos Cobrados'
mask_division = df['División'] == 'Aeropuertos'
df_filtrado = df.loc[mask_fecha & mask_concepto & mask_division]

# Transformar de formato ancho a largo
df_melt = df_filtrado.melt(id_vars=['Fecha', 'División', 'Concepto'],
                           value_vars=df.columns[4:15],
                           var_name='Aeropuerto', value_name='Ingresos')

# Crear un FacetGrid para graficar un boxplot para cada aeropuerto
g = sns.FacetGrid(df_melt, col='Aeropuerto', col_wrap=3, height=5, aspect=1)

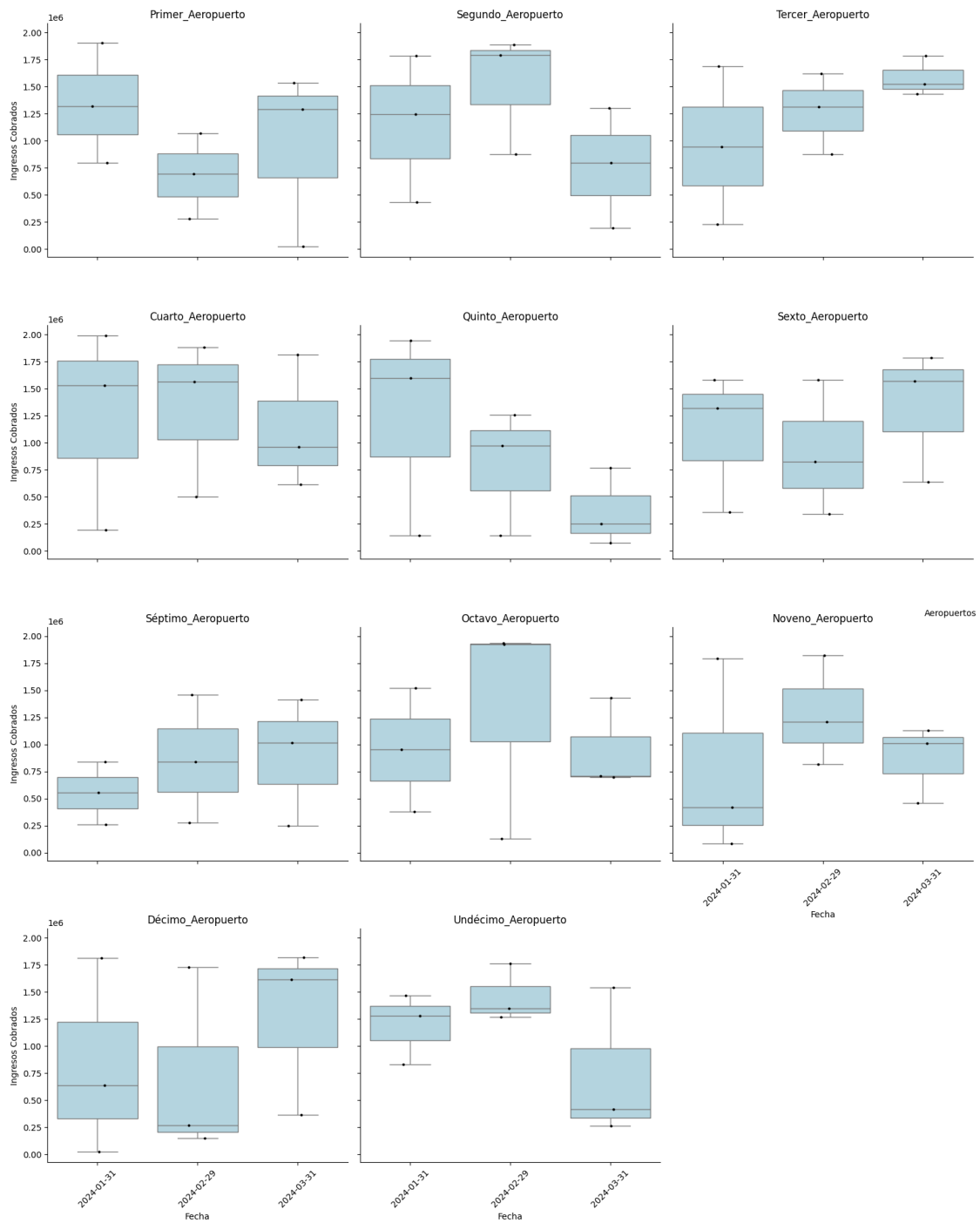
# Función personalizada para dibujar boxplot y stripplot en el mismo eje
def plot_box_strip(data, x, y, **kwargs):
    # Dibuja el boxplot sin puntos outliers visibles
    sns.boxplot(x=x, y=y, data=data, color='lightblue', showfliers=False)
    # Dibuja el stripplot para mostrar todos los puntos incluyendo outliers
    sns.stripplot(x=x, y=y, data=data, color='black', size=3, jitter=True)

# Mapear la función personalizada al FacetGrid
g.map_dataframe(plot_box_strip, x='Fecha', y='Ingresos')

# Ajustar títulos y etiquetas
g.set_titles("{col_name}")
g.set_axis_labels("Fecha", "Ingresos Cobrados")
g.add_legend(title="Aeropuertos")
```

```
# Ajustar los títulos y etiquetas de los ejes
for ax in g.axes.flat:
    for label in ax.get_xticklabels():
        label.set_rotation(45)
    ax.set_title(ax.get_title(), fontsize=12)

plt.tight_layout()
plt.show()
```



2.4.3 Aeropuertos: Gastos

Gráficos de Barras

```
[33]: import pandas as pd
import seaborn as sns
```



```

import matplotlib.pyplot as plt

# Suponiendo que 'df' es tu DataFrame
df['Fecha'] = pd.to_datetime(df['Fecha'])

# Filtrar por fechas, por 'Concepto' y asegurarse de que la 'División' sea
↳ "Aeropuertos"
mask_fecha = (df['Fecha'] >= '2024-01-31') & (df['Fecha'] <= '2024-03-31')
mask_concepto = df['Concepto'] == 'Gastos'
mask_division = df['División'] == 'Aeropuertos'
df_filtrado = df.loc[mask_fecha & mask_concepto & mask_division]

# Transformar de formato ancho a largo
df_melt = df_filtrado.melt(id_vars=['Fecha', 'División', 'Concepto'],
↳ 'Subconcepto'],
                           value_vars=df.columns[4:15],
                           var_name='Aeropuerto', value_name='Gastos')

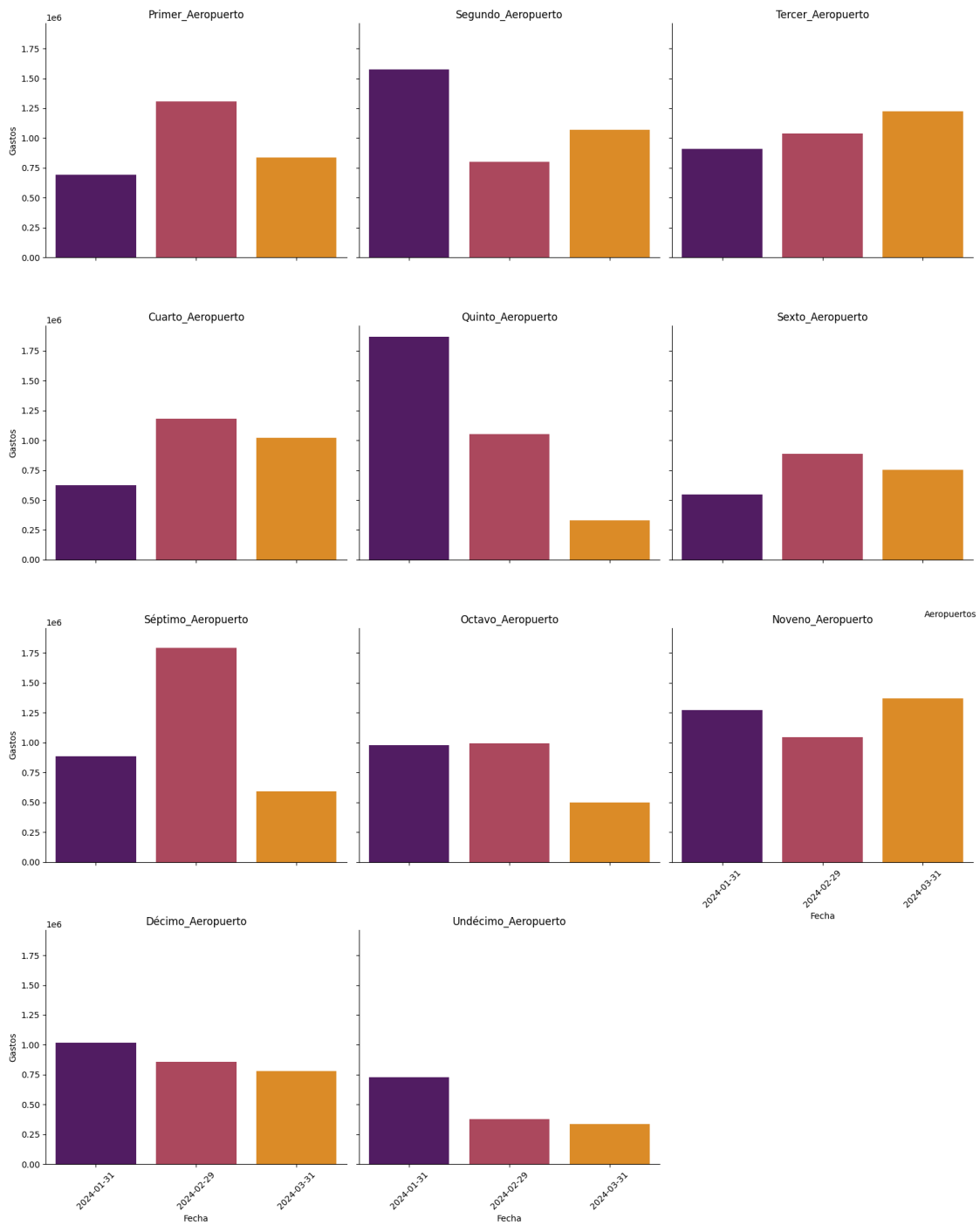
# Crear un FacetGrid para graficar un barplot para cada aeropuerto
g = sns.FacetGrid(df_melt, col='Aeropuerto', col_wrap=3, height=5, aspect=1)
g.map_dataframe(sns.barplot, x='Fecha', y='Gastos', hue='Fecha',
↳ palette='inferno', errorbar=None)

# Ajustar títulos y etiquetas
g.set_titles("{col_name}")
g.set_axis_labels("Fecha", "Gastos")
g.add_legend(title="Aeropuertos")

# Ajustar los títulos y etiquetas de los ejes
for ax in g.axes.flat:
    for label in ax.get_xticklabels():
        label.set_rotation(45)
    ax.set_title(ax.get_title(), fontsize=12)

plt.tight_layout()
plt.show()

```



Boxplots y Outliers

```
[35]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```

# Asegurar que las fechas están en formato datetime
df['Fecha'] = pd.to_datetime(df['Fecha'])

# Filtrar por fechas, 'Concepto' y 'División'
mask_fecha = (df['Fecha'] >= '2024-01-31') & (df['Fecha'] <= '2024-03-31')
mask_concepto = df['Concepto'] == 'Gastos'
mask_division = df['División'] == 'Aeropuertos'
df_filtrado = df.loc[mask_fecha & mask_concepto & mask_division]

# Transformar de formato ancho a largo
df_melt = df_filtrado.melt(id_vars=['Fecha', 'División', 'Concepto'],
                           value_vars=df.columns[4:15],
                           var_name='Aeropuerto', value_name='Gastos')

# Crear un FacetGrid para graficar un boxplot para cada aeropuerto
g = sns.FacetGrid(df_melt, col='Aeropuerto', col_wrap=3, height=5, aspect=1)

# Función personalizada para dibujar boxplot y stripplot en el mismo eje
def plot_box_strip(data, x, y, **kwargs):
    # Dibuja el boxplot sin puntos outliers visibles
    sns.boxplot(x=x, y=y, data=data, color='#efe05f', showfliers=False)
    # Dibuja el stripplot para mostrar todos los puntos incluyendo outliers
    sns.stripplot(x=x, y=y, data=data, color='black', size=3, jitter=True)

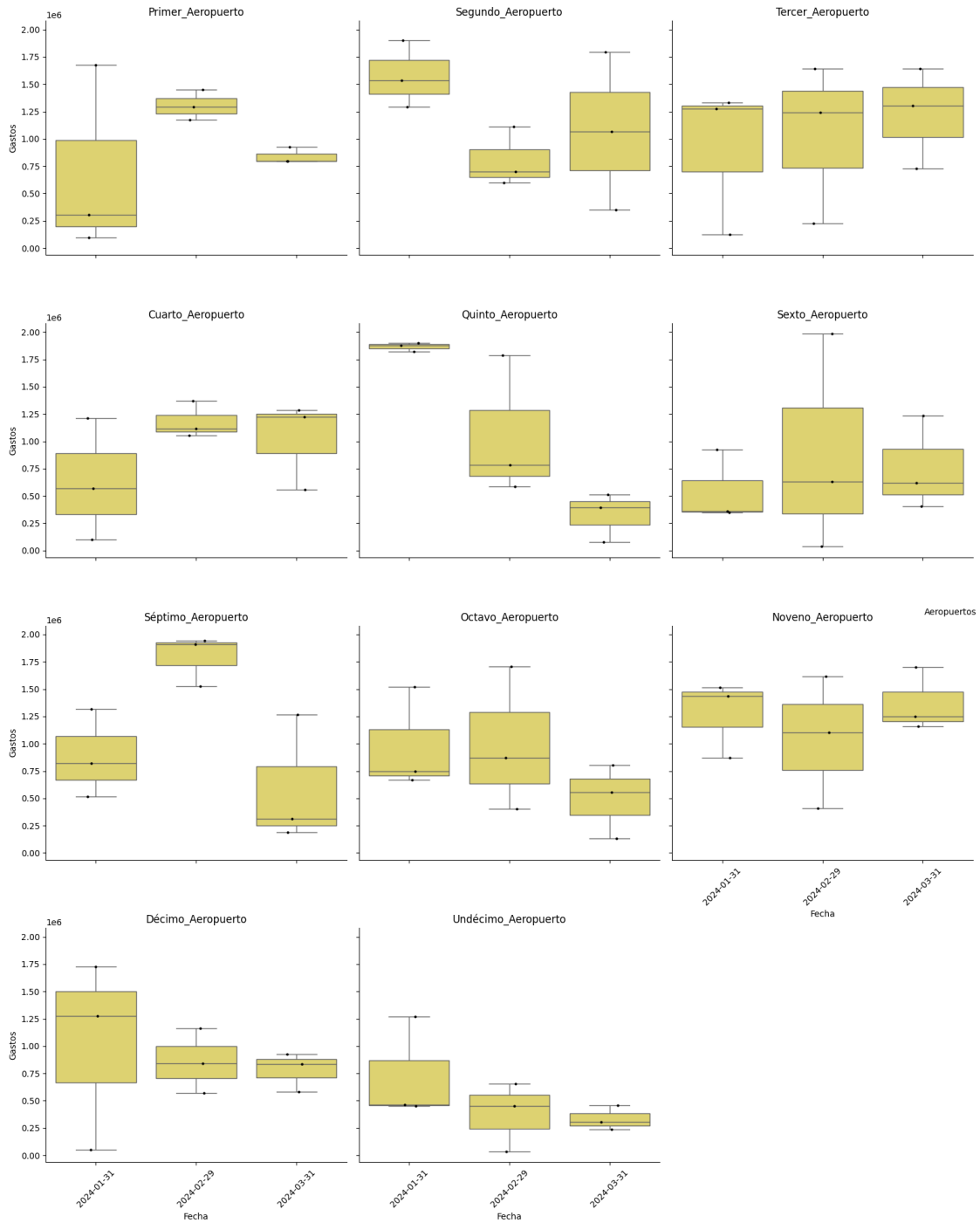
# Mapear la función personalizada al FacetGrid
g.map_dataframe(plot_box_strip, x='Fecha', y='Gastos')

# Ajustar títulos y etiquetas
g.set_titles("{col_name}")
g.set_axis_labels("Fecha", "Gastos")
g.add_legend(title="Aeropuertos")

# Ajustar los títulos y etiquetas de los ejes
for ax in g.axes.flat:
    for label in ax.get_xticklabels():
        label.set_rotation(45)
    ax.set_title(ax.get_title(), fontsize=12)

plt.tight_layout()
plt.show()

```



2.4.4 Combustibles: Ingresos Cobrados

Gráficos de Barras

```
[38]: import pandas as pd
import seaborn as sns
```

```

import matplotlib.pyplot as plt

# Suponiendo que 'df' es tu DataFrame
df['Fecha'] = pd.to_datetime(df['Fecha'])

# Filtrar por fechas, por 'Concepto' y asegurarse de que la 'División' sea
↳ "Combustibles"
mask_fecha = (df['Fecha'] >= '2024-01-31') & (df['Fecha'] <= '2024-03-31')
mask_concepto = df['Concepto'] == 'Ingresos Cobrados'
mask_division = df['División'] == 'Combustibles'
df_filtrado = df.loc[mask_fecha & mask_concepto & mask_division]

# Transformar de formato ancho a largo
df_melt = df_filtrado.melt(id_vars=['Fecha', 'División', 'Concepto'],
↳ 'Subconcepto'],
                           value_vars=df.columns[4:15],
                           var_name='Combustibles', value_name='Ingresos')

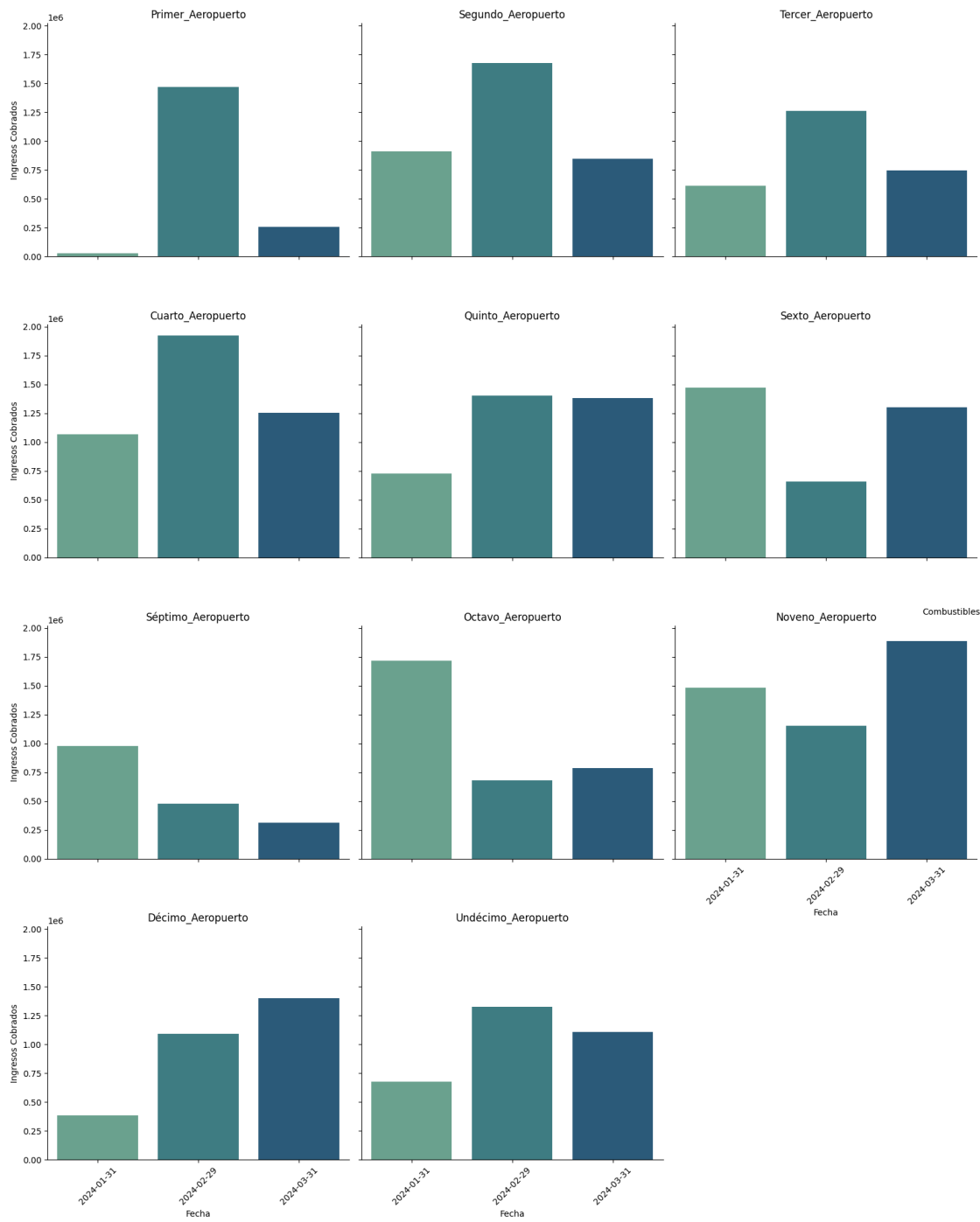
# Crear un FacetGrid para graficar un barplot para cada aeropuerto
g = sns.FacetGrid(df_melt, col='Combustibles', col_wrap=3, height=5, aspect=1)
g.map_dataframe(sns.barplot, x='Fecha', y='Ingresos', hue='Fecha',
↳ palette='crest', errorbar=None)

# Ajustar títulos y etiquetas
g.set_titles("{col_name}")
g.set_axis_labels("Fecha", "Ingresos Cobrados")
g.add_legend(title="Combustibles")

# Ajustar los títulos y etiquetas de los ejes
for ax in g.axes.flat:
    for label in ax.get_xticklabels():
        label.set_rotation(45)
    ax.set_title(ax.get_title(), fontsize=12)

plt.tight_layout()
plt.show()

```



Boxplots y Outliers

```
[41]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```

# Asegurar que las fechas están en formato datetime
df['Fecha'] = pd.to_datetime(df['Fecha'])

# Filtrar por fechas, 'Concepto' y 'División'
mask_fecha = (df['Fecha'] >= '2024-01-31') & (df['Fecha'] <= '2024-03-31')
mask_concepto = df['Concepto'] == 'Ingresos Cobrados'
mask_division = df['División'] == 'Combustibles'
df_filtrado = df.loc[mask_fecha & mask_concepto & mask_division]

# Transformar de formato ancho a largo
df_melt = df_filtrado.melt(id_vars=['Fecha', 'División', 'Concepto'],
    ↪ 'Subconcepto'],
                           value_vars=df.columns[4:15],
                           var_name='Combustibles', value_name='Ingresos')

# Crear un FacetGrid para graficar un boxplot para cada aeropuerto
g = sns.FacetGrid(df_melt, col='Combustibles', col_wrap=3, height=5, aspect=1)

# Función personalizada para dibujar boxplot y stripplot en el mismo eje
def plot_box_strip(data, x, y, **kwargs):
    # Dibuja el boxplot sin puntos outliers visibles
    sns.boxplot(x=x, y=y, data=data, color='#0b65ac', showfliers=False)
    # Dibuja el stripplot para mostrar todos los puntos incluyendo outliers
    sns.stripplot(x=x, y=y, data=data, color='black', size=3, jitter=True)

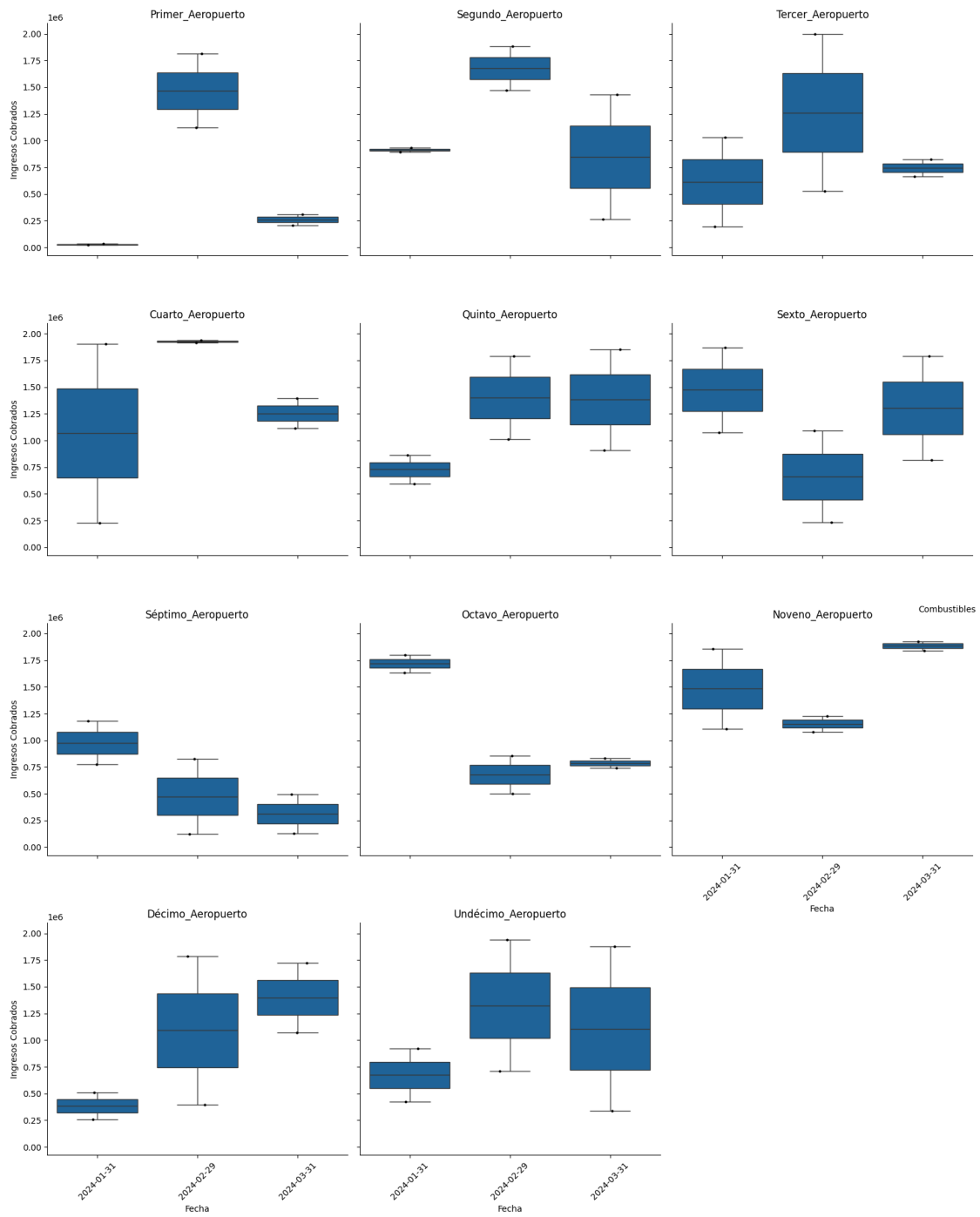
# Mapear la función personalizada al FacetGrid
g.map_dataframe(plot_box_strip, x='Fecha', y='Ingresos')

# Ajustar títulos y etiquetas
g.set_titles("{col_name}")
g.set_axis_labels("Fecha", "Ingresos Cobrados")
g.add_legend(title="Combustibles")

# Ajustar los títulos y etiquetas de los ejes
for ax in g.axes.flat:
    for label in ax.get_xticklabels():
        label.set_rotation(45)
    ax.set_title(ax.get_title(), fontsize=12)

plt.tight_layout()
plt.show()

```



2.4.5 Combustibles: Gastos

Gráficos de Barras

```
[43]: import pandas as pd
import seaborn as sns
```



```

import matplotlib.pyplot as plt

# Suponiendo que 'df' es tu DataFrame
df['Fecha'] = pd.to_datetime(df['Fecha'])

# Filtrar por fechas, por 'Concepto' y asegurarse de que la 'División' sea
↳ "Combustibles"
mask_fecha = (df['Fecha'] >= '2024-01-31') & (df['Fecha'] <= '2024-03-31')
mask_concepto = df['Concepto'] == 'Gastos'
mask_division = df['División'] == 'Combustibles'
df_filtrado = df.loc[mask_fecha & mask_concepto & mask_division]

# Transformar de formato ancho a largo
df_melt = df_filtrado.melt(id_vars=['Fecha', 'División', 'Concepto'],
↳ 'Subconcepto'],
                           value_vars=df.columns[4:15],
                           var_name='Combustibles', value_name='Gastos')

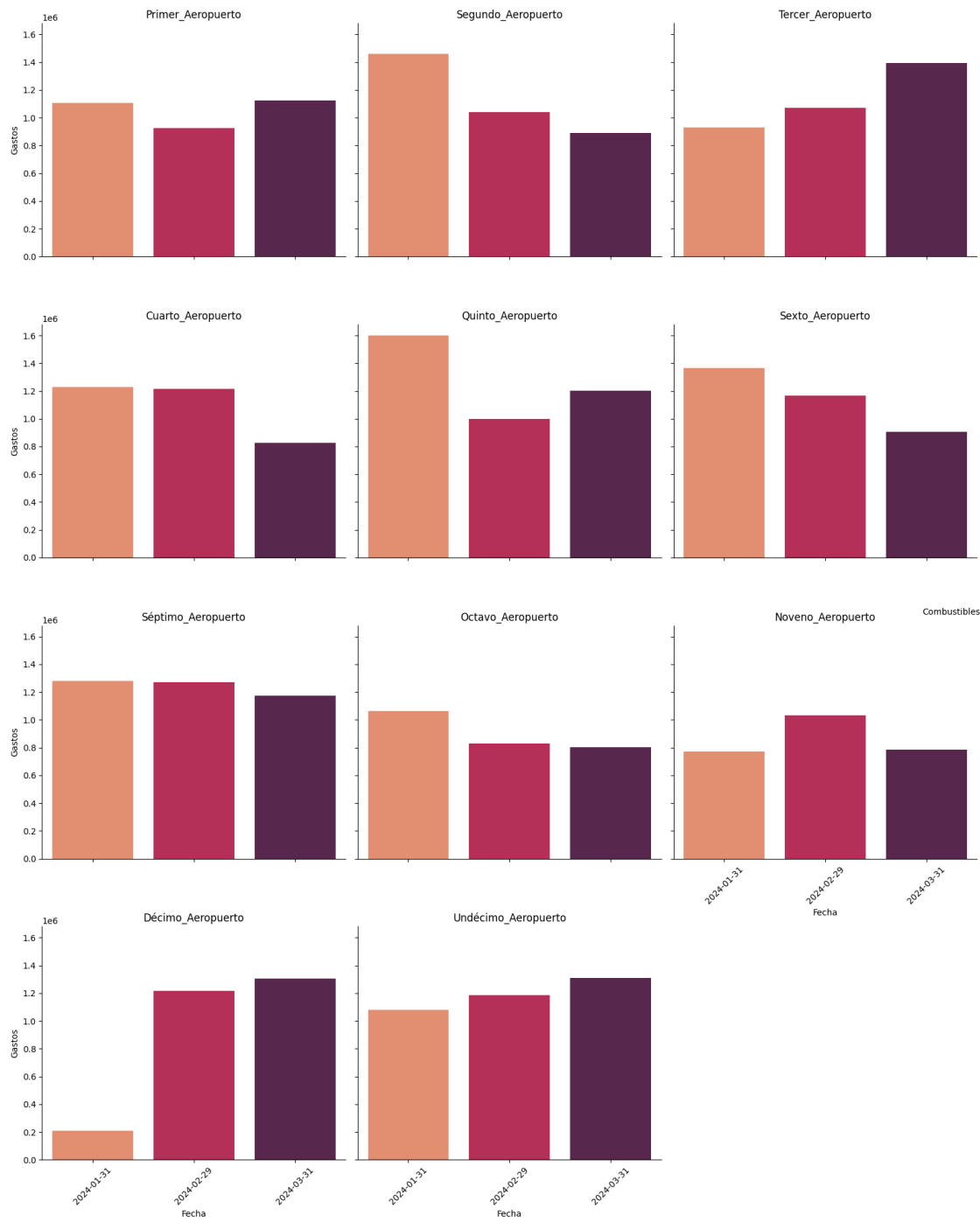
# Crear un FacetGrid para graficar un barplot para cada aeropuerto
g = sns.FacetGrid(df_melt, col='Combustibles', col_wrap=3, height=5, aspect=1)
g.map_dataframe(sns.barplot, x='Fecha', y='Gastos', hue='Fecha',
↳ palette='rocket_r', errorbar=None)

# Ajustar títulos y etiquetas
g.set_titles("{col_name}")
g.set_axis_labels("Fecha", "Gastos")
g.add_legend(title="Combustibles")

# Ajustar los títulos y etiquetas de los ejes
for ax in g.axes.flat:
    for label in ax.get_xticklabels():
        label.set_rotation(45)
    ax.set_title(ax.get_title(), fontsize=12)

plt.tight_layout()
plt.show()

```



Boxplots y Outliers

```
[48]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```

# Asegurar que las fechas están en formato datetime
df['Fecha'] = pd.to_datetime(df['Fecha'])

# Filtrar por fechas, 'Concepto' y 'División'
mask_fecha = (df['Fecha'] >= '2024-01-31') & (df['Fecha'] <= '2024-03-31')
mask_concepto = df['Concepto'] == 'Gastos'
mask_division = df['División'] == 'Combustibles'
df_filtrado = df.loc[mask_fecha & mask_concepto & mask_division]

# Transformar de formato ancho a largo
df_melt = df_filtrado.melt(id_vars=['Fecha', 'División', 'Concepto'],
    ↪ 'Subconcepto'],
                        value_vars=df.columns[4:15],
                        var_name='Combustibles', value_name='Gastos')

# Crear un FacetGrid para graficar un boxplot para cada aeropuerto
g = sns.FacetGrid(df_melt, col='Combustibles', col_wrap=3, height=5, aspect=1)

# Función personalizada para dibujar boxplot y stripplot en el mismo eje
def plot_box_strip(data, x, y, **kwargs):
    # Dibuja el boxplot sin puntos outliers visibles
    sns.boxplot(x=x, y=y, data=data, color='#e2e3a9', showfliers=False)
    # Dibuja el stripplot para mostrar todos los puntos incluyendo outliers
    sns.stripplot(x=x, y=y, data=data, color='black', size=3, jitter=True)

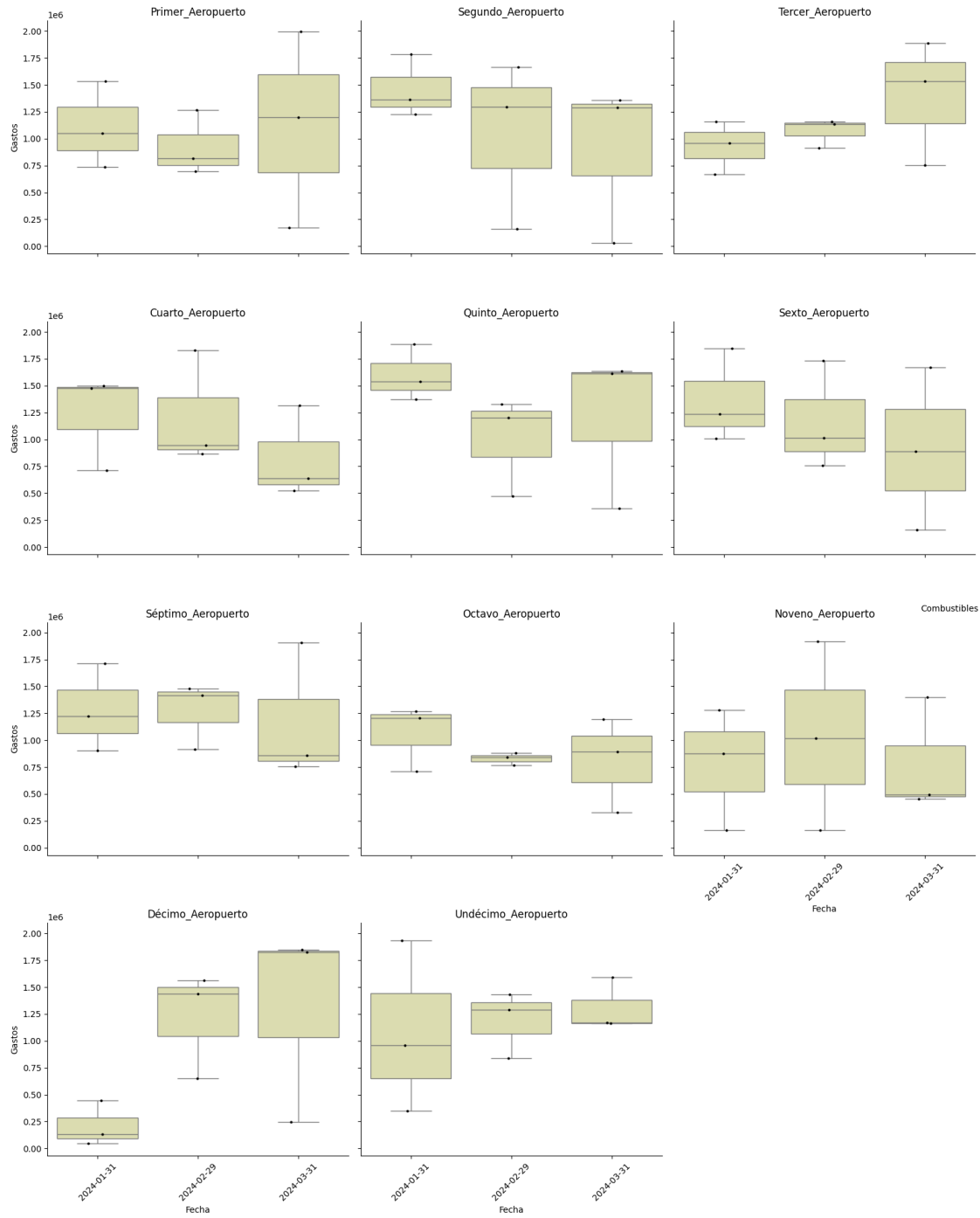
# Mapear la función personalizada al FacetGrid
g.map_dataframe(plot_box_strip, x='Fecha', y='Gastos')

# Ajustar títulos y etiquetas
g.set_titles("{col_name}")
g.set_axis_labels("Fecha", "Gastos")
g.add_legend(title="Combustibles")

# Ajustar los títulos y etiquetas de los ejes
for ax in g.axes.flat:
    for label in ax.get_xticklabels():
        label.set_rotation(45)
    ax.set_title(ax.get_title(), fontsize=12)

plt.tight_layout()
plt.show()

```



2.5 Fin