



# Universidad Autónoma de Baja California



Facultad de Ciencias

Programación para Ciencia de Datos

Proyecto final:

***¿Cómo han cambiado las temperaturas y patrones de lluvia en diferentes regiones de Australia a lo largo del tiempo?***

***“Rain in Australia”***

[Francisco Javier Fonseca Garcia](#)

Matrícula: 358399

Ensenada, Baja California a 17 de Mayo del 2024

## Introducción:

El dataset titulado “Rain in Australia”, fué extraído de Kaggle ([Rain in Australia \(kaggle.com\)](https://www.kaggle.com/dhansh/australia-rain)) y contiene datos de aproximadamente 10 años de las observaciones diarias del clima en distintas locaciones de Australia, el dataset pretende responder a la pregunta de investigación y será analizado y manejado mediante distintas librerías como se muestra en la figura 1, donde también agregamos el dataset a nuestra variable “clima” utilizando pandas e imprimimos el número de filas y columnas para darnos una idea de las dimensiones, donde notamos que tenemos más de 145 mil muestras (filas) y cada dato cuenta con 23 características (columnas).

```
[ ] import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import datetime

#guardamos el dataset en la variable clima
clima=pd.read_csv('./lluvia.csv')

#Mostrar la cantidad de filas y columnas del dataset
clima.shape

(145460, 23)
```

Figura 1. Librerías y agregar el dataset.

Posteriormente imprimimos los encabezados para tener a la mano los nombres de las columnas y darnos una idea de los datos que maneja.

```
#Imprimir encabezados
clima.head()
```

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Cloud9am	Cloud3pm
0	2008-12-01	Albury	13.4	22.9	0.6	NaN	NaN	W	44.0	W	...	71.0	22.0	1007.7	1007.1	8.0	...
1	2008-12-02	Albury	7.4	25.1	0.0	NaN	NaN	WNW	44.0	NNW	...	44.0	25.0	1010.6	1007.8	NaN	...
2	2008-12-03	Albury	12.9	25.7	0.0	NaN	NaN	WSW	46.0	W	...	38.0	30.0	1007.6	1008.7	NaN	...
3	2008-12-04	Albury	9.2	28.0	0.0	NaN	NaN	NE	24.0	SE	...	45.0	16.0	1017.6	1012.8	NaN	...
4	2008-12-05	Albury	17.5	32.3	1.0	NaN	NaN	W	41.0	ENE	...	82.0	33.0	1010.8	1006.0	7.0	...

5 rows x 23 columns

Figura 2. Encabezados del dataset.

Observando las dos columnas vemos que los valores que maneja no son manejables, por lo que procederemos a borrar las dos columnas ya que no pretendemos usarlas, pero antes de eso corroboramos los nombres de las columnas haciendo un listado de ellas y posteriormente ver los tipos de datos que tenemos.

```
# Nombres de las columnas
nombres_columnas = clima.columns.tolist() # Convertimos el índice de columnas a una lista
print("\nNombres de las columnas:", nombres_columnas)

# Tipos de datos de las columnas
tipos_datos = clima.dtypes
print("\nTipos de datos de las columnas:")
print(tipos_datos)
```

Nombres de las columnas: ['Date', 'Location', 'MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine', 'WindGustDir', 'WindGustSpeed', 'WindDir9am', 'WindDir3pm', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am', 'Temp3pm', 'RainToday', 'RainTomorrow']

Tipos de datos de las columnas:

Date	object
Location	object
MinTemp	float64
MaxTemp	float64
Rainfall	float64
Evaporation	float64
Sunshine	float64
WindGustDir	object
WindGustSpeed	float64
WindDir9am	object
WindDir3pm	object
WindSpeed9am	float64
WindSpeed3pm	float64
Humidity9am	float64
Humidity3pm	float64
Pressure9am	float64
Pressure3pm	float64
Cloud9am	float64
Cloud3pm	float64
Temp9am	float64
Temp3pm	float64
RainToday	object
RainTomorrow	object
dtype:	object

Figura 3. Columnas y tipos de datos de cada una.

Por practicidad borraremos algunas columnas adicionales ya como las que manejan las direcciones de 9am y 3pm del viento, ya que no serán necesarias.

```

# Eliminar las columnas que no necesitamos o que no tengan datos utilizables
clima_es = clima.drop(columns=['Evaporation', 'Sunshine', 'WindGustDir', 'WindDir9am', 'WindDir3pm'])

# Verificar que las columnas han sido eliminadas
print(clima_es.head())

```

	Date	Location	MinTemp	MaxTemp	Rainfall	WindGustSpeed	\
0	2008-12-01	Albury	13.4	22.9	0.6	44.0	
1	2008-12-02	Albury	7.4	25.1	0.0	44.0	
2	2008-12-03	Albury	12.9	25.7	0.0	46.0	
3	2008-12-04	Albury	9.2	28.0	0.0	24.0	
4	2008-12-05	Albury	17.5	32.3	1.0	41.0	

	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	\
0	20.0	24.0	71.0	22.0	1007.7	
1	4.0	22.0	44.0	25.0	1010.6	
2	19.0	26.0	38.0	30.0	1007.6	
3	11.0	9.0	45.0	16.0	1017.6	
4	7.0	20.0	82.0	33.0	1010.8	

	Pressure3pm	Cloud9am	Cloud3pm	Temp9am	Temp3pm	RainToday	RainTomorrow
0	1007.1	8.0	NaN	16.9	21.8	No	No
1	1007.8	NaN	NaN	17.2	24.3	No	No
2	1008.7	NaN	2.0	21.0	23.2	No	No
3	1012.8	NaN	NaN	18.1	26.5	No	No
4	1006.0	7.0	8.0	17.8	29.7	No	No

Figura 4. Borrado de columnas.

Iniciando con el análisis, agrupamos las columnas de temperatura min y max por localidad, les sacamos el promedio y el promedio lo ordenamos de menor a mayor para ver qué localidad han tenido las temperaturas en promedio más altas en los últimos 10 años. También sacaremos los estadísticos con la función .describe para temperaturas min y max

```

# Calcular el promedio de MinTemp y MaxTemp por Location
prom_temp_location = clima_es[['Location', 'MinTemp', 'MaxTemp']].groupby('Location').mean()

#Grega una nueva columna donde se hace un promedio de las temperaturas mx y min y las ordena de menor a mayor
prom_temp_location['PromTemp'] = prom_temp_location[['MinTemp', 'MaxTemp']].mean(axis=1)
prom_temp_location = prom_temp_location.sort_values(by='PromTemp')

# Mostrar los resultados y redondear a 2 decimales
print(prom_temp_location.round(2))

```

	Location	MinTemp	MaxTemp	PromTemp
0	MountGinini	3.58	11.66	7.62
1	Ballarat	7.37	18.29	12.83
2	Launceston	7.84	18.95	13.40
3	Hobart	9.08	17.87	13.47
4	Portland	9.59	18.06	13.82
5	Canberra	6.82	20.98	13.90
6	Tuggeranong	7.24	20.77	14.00

Dartmoor	8.62	19.70	14.16
MountGambier	8.83	19.83	14.33
Sale	8.57	20.28	14.42
Bendigo	8.59	21.62	15.11
MelbourneAirport	9.97	20.49	15.23
Watsonia	10.13	20.90	15.51
Nuriootpa	9.36	21.70	15.53
Nhil	9.00	22.41	15.70
Albury	9.53	22.64	16.09
Walpole	11.81	20.46	16.13
Witchcliffe	10.78	21.69	16.23
Melbourne	11.78	20.77	16.27
Waggawagga	9.62	23.03	16.32
Albany	12.95	20.07	16.51
SalmonGums	9.31	24.25	16.78
BadgerysCreek	11.13	24.03	17.58
Adelaide	12.58	22.90	17.74
Mildura	10.73	24.84	17.79
Richmond	11.36	24.46	17.91
Wollongong	14.94	21.47	18.20
Williamstown	12.78	24.00	18.39
Penrith	12.58	24.76	18.67
Newcastle	13.76	24.10	18.93
Sydney	14.87	23.00	18.93
NorahHead	15.36	22.59	18.97
Perth	12.92	25.03	18.98
PerthAirport	12.58	25.53	19.06
SydneyAirport	14.90	23.38	19.14
PearceRAAF	12.30	26.05	19.18
CoffsHarbour	14.44	23.97	19.20
NorfolkIsland	16.87	21.83	19.35

Cobar	13.15	25.87	19.51
Moree	12.91	26.95	19.93
Woomera	13.36	26.60	19.98
AliceSprings	13.14	29.25	21.20
Brisbane	16.42	26.45	21.44
GoldCoast	17.36	25.77	21.57
Uluru	14.47	30.38	22.42
Townsville	20.42	29.37	24.89
Cairns	21.22	29.56	25.39
Katherine	20.55	34.94	27.74
Darwin	23.21	32.54	27.88

```
✓ 0s # estadísticas descriptivas de las temperaturas
      descripcion_temp = clima_df['MaxTemp'].describe()
      print(descripcion_temp)
      descripcion_rain = clima_df['MinTemp'].describe()
      print(descripcion_rain)
```

```
count    144199.000000
mean       23.221348
std         7.119049
min        -4.800000
25%        17.900000
50%        22.600000
75%        28.200000
max        48.100000
Name: MaxTemp, dtype: float64
count    143975.000000
mean       12.194034
std         6.398495
min        -8.500000
25%         7.600000
50%        12.000000
75%        16.900000
max        33.900000
Name: MinTemp, dtype: float64
```

Figura 5. Promedios de temperatura, min, max y estadísticos de ambos.

Para este punto se agregó una nueva biblioteca, la cual nos ayuda a tener un mejor control y manejo de las fechas para solucionar un problema a la hora de graficar y poder acotar los tiempos visiblemente mejor, en este caso, para poder tener una vista amplia de los datos completos, se realizó una gráfica donde se ilustran las temperaturas máximas y las mínimas (en °C), usando un gráfico de línea, en otra gráfica por separado se ilustraron las precipitaciones (en mm), ambas en el tiempo de los 10 años, que van desde el 2008 hasta finales del 2017.

```

import matplotlib.dates as mdates

# Cambiar columna 'Date' a formato de fecha
clima_es_df['Date'] = pd.to_datetime(clima_es_df['Date'])

# Filtrar datos para una región específica
region = 'Sydney'
region_df = clima_es_df[clima_es_df['Location'] == region]

# Configurar el estilo de los gráficos
sns.set(style="whitegrid")

# Crear una figura para los gráficos con un tamaño especificado
plt.figure(figsize=(14, 7))

# Tendencia de temperaturas mínimas y máximas
plt.subplot(2, 1, 1) # Crear un subplot para las temperaturas (primero de dos, 2 filas, 1 columna, primer gráfico)
plt.plot(region_df['Date'], region_df['MinTemp'], label='MinTemp') # Graficar la temperatura mínima
plt.plot(region_df['Date'], region_df['MaxTemp'], label='MaxTemp') # Graficar la temperatura máxima
plt.title(f'Tendencias de Temperaturas en {region}') # Título del gráfico
plt.xlabel('Fecha') # Etiqueta del eje x
plt.ylabel('Temperatura (°C)') # Etiqueta del eje y
plt.legend() # Mostrar la leyenda
plt.gca().xaxis.set_major_locator(mdates.YearLocator()) # Configurar las etiquetas del eje x para mostrar solo los años
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y')) # Formatear las etiquetas del eje x para mostrar solo el año
plt.xticks(rotation=45) # Rotar las etiquetas del eje x 45 grados para mejorar la legibilidad

# Tendencia de precipitación
plt.subplot(2, 1, 2) # Crear un subplot para la precipitación (segundo de dos, 2 filas, 1 columna, segundo gráfico)
plt.plot(region_df['Date'], region_df['Rainfall'], label='Rainfall', color='blue') # Graficar la precipitación
plt.title(f'Tendencia de Precipitación en {region}') # Título del gráfico
plt.xlabel('Fecha') # Etiqueta del eje x
plt.ylabel('Precipitación (mm)') # Etiqueta del eje y
plt.legend() # Mostrar la leyenda
plt.gca().xaxis.set_major_locator(mdates.YearLocator()) # Configurar las etiquetas del eje x para mostrar solo los años
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y')) # Formatear las etiquetas del eje x para mostrar solo el año
plt.xticks(rotation=45) # Rotar las etiquetas del eje x 45 grados para mejorar la legibilidad

# Ajustar el layout para que los subplots no se solapen
plt.tight_layout()

# Mostrar el gráfico
plt.show()

```

Al ver los datos de las temperaturas en las gráficas podemos observar de manera más evidente los puntos más altos, como en la temperatura máxima en 2013, donde vemos que ha sido el valor más alto de los 10 años y al ver la gráfica de las precipitaciones, notamos que anterior a eso, tenemos lo que parece una aparente sequía dado que los valores son bajos y prolongados, por lo que podría ser un indicio de esa temperatura tan elevada.

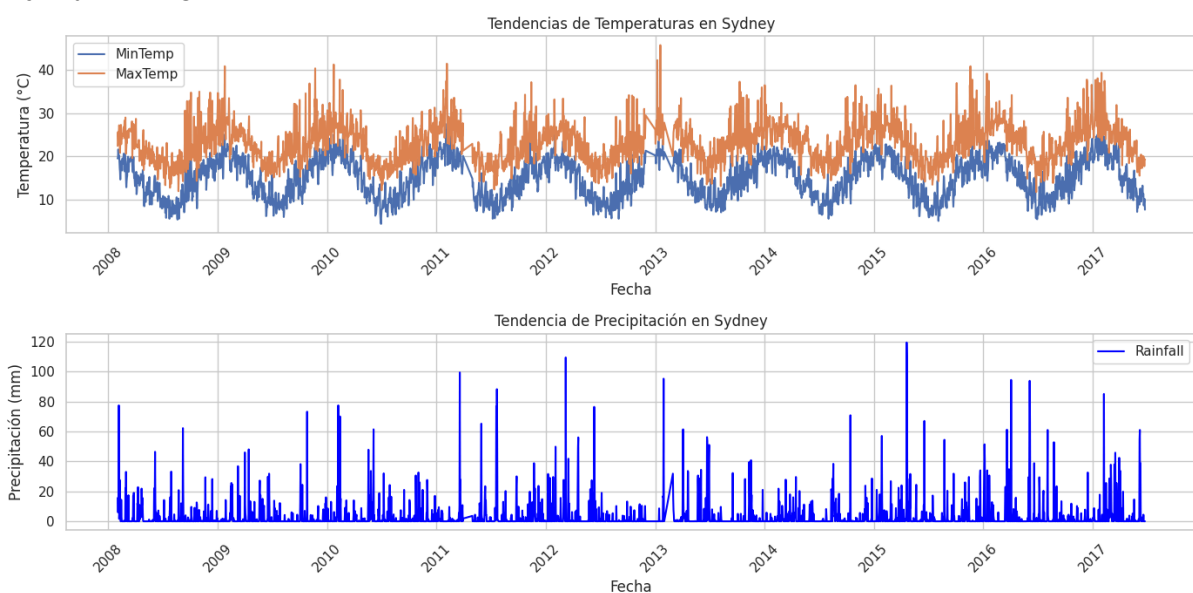


Figura 6. Gráficos de temperatura min, max y precipitaciones de 10 años.

Para tener una vista más concisa sobre el punto mencionado anteriormente de la sequía, podremos graficar ahora por el promedio anual para poder observar la tendencia que lleva en estos años

```
# Asegúrate de que la columna 'Date' esté en formato de fecha
clima_es_df['Date'] = pd.to_datetime(clima_es_df['Date'])

# Agregar una columna de año
clima_es_df['Year'] = clima_es_df['Date'].dt.year

# Calcular los promedios anuales
annual_avg = clima_es_df.groupby(['Location', 'Year'])[['MinTemp', 'MaxTemp', 'Rainfall']].mean().reset_index()

# Filtrar datos para una región específica (por ejemplo, "Sydney")
region = 'Sydney'
region_annual_avg = annual_avg[annual_avg['Location'] == region]

# Configurar el estilo de los gráficos
sns.set(style="whitegrid")

# Graficar las tendencias de temperaturas y precipitación
plt.figure(figsize=(14, 7))

# Tendencia de temperaturas mínimas y máximas
plt.subplot(2, 1, 1)
plt.plot(region_annual_avg['Year'], region_annual_avg['MinTemp'], marker='o', label='MinTemp')
plt.plot(region_annual_avg['Year'], region_annual_avg['MaxTemp'], marker='o', label='MaxTemp')
plt.title(f'Tendencias de Temperaturas Anuales en {region}')
plt.xlabel('Año')
plt.ylabel('Temperatura (°C)')
plt.xticks(region_annual_avg['Year'], rotation=45)
plt.legend()

# Tendencia de precipitación
plt.subplot(2, 1, 2)
plt.plot(region_annual_avg['Year'], region_annual_avg['Rainfall'], marker='o', label='Rainfall', color='blue')
plt.title(f'Tendencia de Precipitación Anual en {region}')
plt.xlabel('Año')
plt.ylabel('Precipitación (mm)')
plt.xticks(region_annual_avg['Year'], rotation=45)
plt.legend()

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

Aquí notamos una gran disminución de las precipitaciones en el año de entre el 2013 y 2014, por lo que podemos confirmar parcialmente el punto anterior, de igual manera podemos ver como las temperaturas máximas han ido incrementando periódicamente a lo largo de los años notando que del 2009 al 2012 se mantiene pero incremente unos grados, manteniéndose por un par de años nuevamente hasta volver a aumentar a partir del 2016.



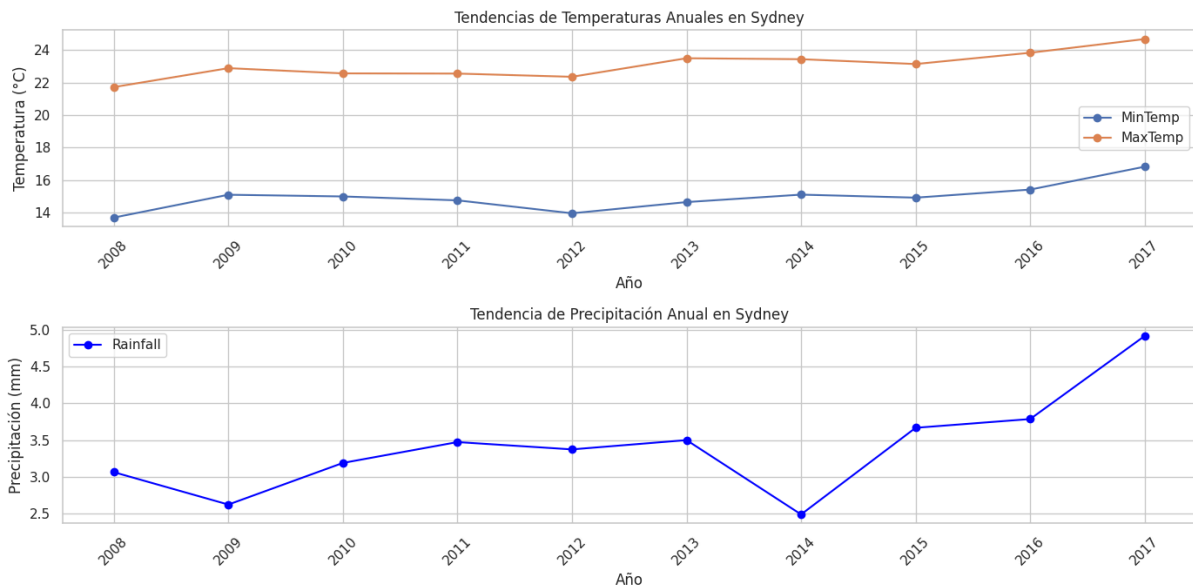


Figura 7. Gráficos de temperatura min, max y precipitaciones por promedio anual.

Para descripciones posteriores podemos graficar la tendencia de temperaturas y precipitaciones por estado y año, lo cual nos podrá servir como una especie de biblioteca, donde podemos observar tendencias de cada estado y de forma visual seleccionar lo que nos interese para poder hacer algún análisis

```
# Agregar columnas de año y mes para análisis temporal
# Estas nuevas columnas permiten análisis y agrupaciones por año y mes.
clima_es_df['Year'] = clima_es_df['Date'].dt.year # Extraer el año de la columna 'Date'
clima_es_df['Month'] = clima_es_df['Date'].dt.month # Extraer el mes de la columna 'Date'

# Calcular las medias mensuales y anuales para cada región
# Agrupa los datos por ubicación, año y mes, y luego calcula la media de 'MinTemp', 'MaxTemp' y 'Rainfall'.
monthly_avg = clima_es_df.groupby(['Location', 'Year', 'Month'])[['MinTemp', 'MaxTemp', 'Rainfall']].mean().reset_index()
# Agrupa los datos por ubicación y año, y luego calcula la media de 'MinTemp', 'MaxTemp' y 'Rainfall'.
annual_avg = clima_es_df.groupby(['Location', 'Year'])[['MinTemp', 'MaxTemp', 'Rainfall']].mean().reset_index()

# Graficar tendencias anuales para todas las regiones
# Crear una nueva figura para las gráficas de tendencias anuales
plt.figure(figsize=(14, 7))

# Tendencias anuales de temperatura
# Utiliza seaborn para crear un gráfico de líneas que muestra las tendencias de la temperatura máxima anual para cada región.
sns.lineplot(x='Year', y='MaxTemp', hue='Location', data=annual_avg)
plt.title('Tendencia Anual de Temperaturas Máximas en Diferentes Regiones') # Título del gráfico
plt.xlabel('Año') # Etiqueta del eje x
plt.ylabel('Temperatura Máxima (°C)') # Etiqueta del eje y
# Configura la leyenda para mostrar fuera del gráfico, a la derecha.
plt.legend(title='Location', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show() # Muestra el gráfico

# Crear otra figura para las gráficas de tendencias anuales de precipitación
plt.figure(figsize=(14, 7))

# Tendencias anuales de precipitación
# Utiliza seaborn para crear un gráfico de líneas que muestra las tendencias de la precipitación anual para cada región.
sns.lineplot(x='Year', y='Rainfall', hue='Location', data=annual_avg)
plt.title('Tendencia Anual de Precipitación en Diferentes Regiones') # Título del gráfico
plt.xlabel('Año') # Etiqueta del eje x
plt.ylabel('Precipitación Anual (mm)') # Etiqueta del eje y
# Configura la leyenda para mostrar fuera del gráfico, a la derecha.
plt.legend(title='Location', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show() # Muestra el gráfico
```

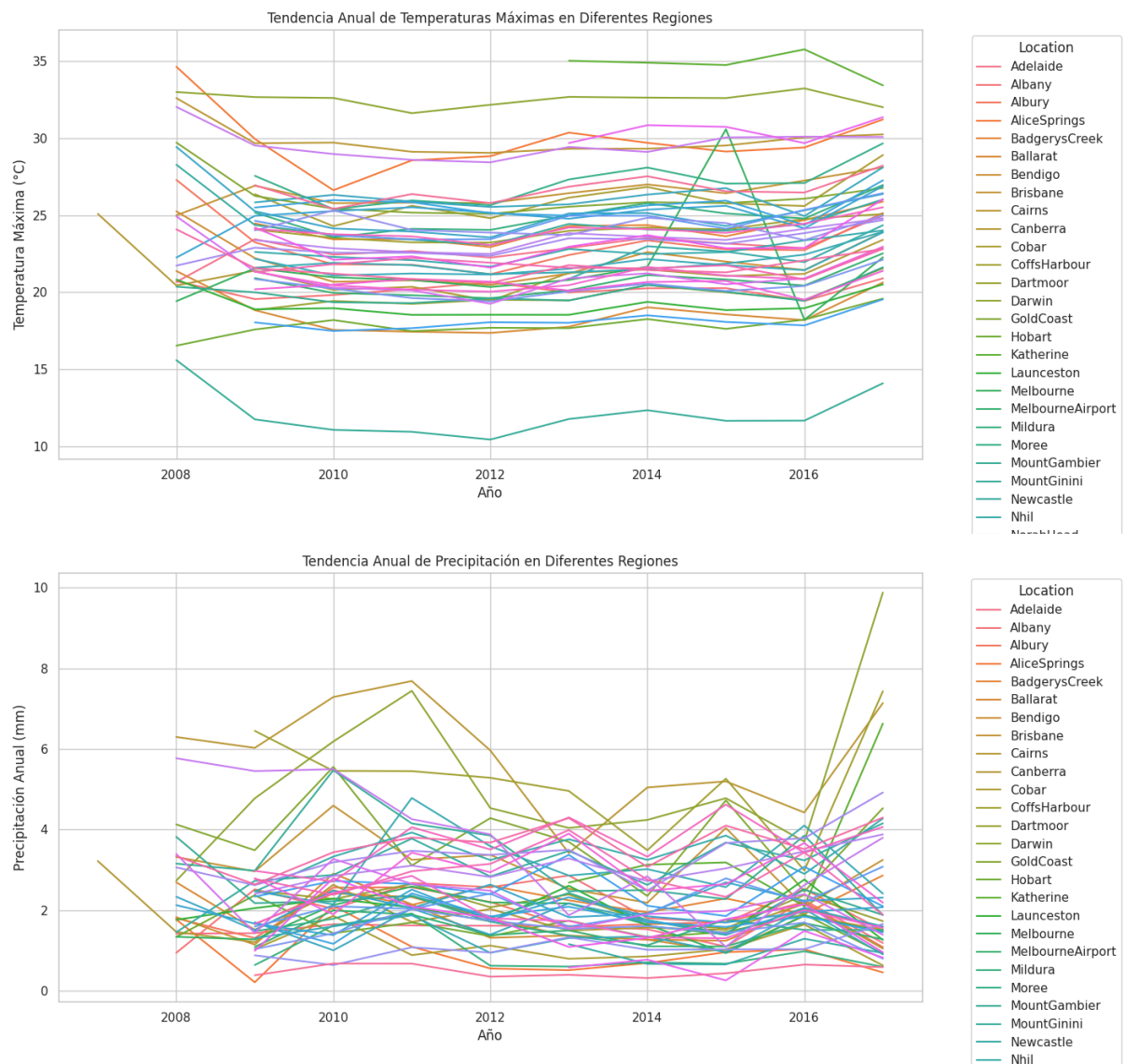


Figura 8. Gráficos de temperatura, max y precipitaciones por tendencia anual por estado.

De igual manera podemos hacerlo por mes para poder analizar la variabilidad con respecto a los meses en cualquier estado, esto con temperaturas máximas y precipitaciones

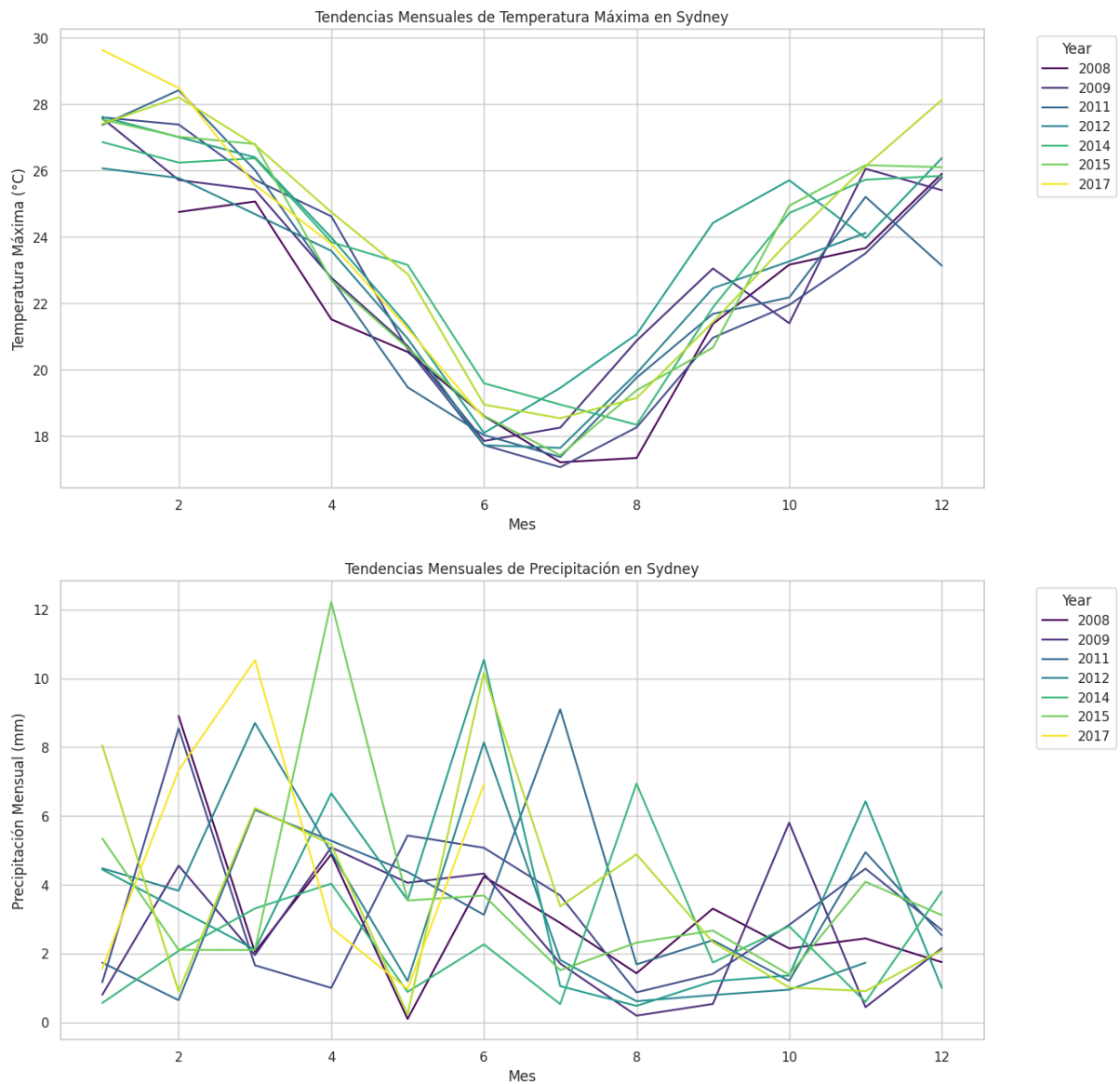


Figura 9. Gráficos de temperatura, max y precipitaciones por tendencia mensual en Sydney.

Ahora identificamos los valores máximos de temperatura que tuvo cada región por cada año y haremos lo mismo con las precipitaciones

```

# Función para identificar años con eventos climáticos extremos
def find_extreme_years(df, variable, top_n=5):
    extreme_years = df.groupby(['Location', 'Year'])[variable].mean().reset_index()
    extreme_years = extreme_years.sort_values(by=variable, ascending=False).groupby('Location').head(top_n)
    return extreme_years

# Identificar años con temperaturas máximas más altas
extreme_temp_years = find_extreme_years(clima_es_df, 'MaxTemp')

# Identificar años con precipitación más alta
extreme_rainfall_years = find_extreme_years(clima_es_df, 'Rainfall')

# Mostrar los resultados
print("Años con temperaturas máximas más altas por región:")
print(extreme_temp_years)

print("\nAños con precipitación más alta por región:")
print(extreme_rainfall_years)

```

```

➡ Años con temperaturas máximas más altas por región:
   Location  Year  MaxTemp
160  Katherine 2016  35.769101
157  Katherine 2013  35.031208
158  Katherine 2014  34.902486
159  Katherine 2015  34.752941
30  AliceSprings 2008  34.641935
..      ...      ...      ...
219  MountGinini 2008  15.590323
228  MountGinini 2017  14.090909
225  MountGinini 2014  12.339452
224  MountGinini 2013  11.767164
220  MountGinini 2009  11.742398

[245 rows x 3 columns]

Años con precipitación más alta por región:
   Location  Year  Rainfall
136  Darwin  2017  9.880682
82   Cairns  2011  7.684776
130  Darwin  2011  7.441791
117  CoffsHarbour 2017  7.430682
81   Cairns  2010  7.287123
..      ...      ...      ...
451  Woomera  2016  0.649580
452  Woomera  2017  0.585795
384  Uluru    2013  0.577049
450  Woomera  2015  0.434521
386  Uluru    2015  0.257459

[245 rows x 3 columns]

```

Figura 10. Valores máximo de temperatura y precipitación por año en cada ciudad.

Para tener una vista general de los cambios de temperatura, compararemos por periodos de tiempo, es decir, del 2008 al 2012 será el primer periodo y del 2013 al 2017 el segundo, al tener el promedio de estos periodos podremos compararlos para darnos cuenta que la tendencia indica nuevamente que las temperatura han ido en aumento prácticamente en todos los estados, por lo que podríamos generalizar que la temperatura de Australia ha ido en aumento en la década analizada



```
# Función para comparar periodos específicos
def compare_periods(df, start_year, end_year, variable):
    period_data = df[(df['Year'] >= start_year) & (df['Year'] <= end_year)]
    period_avg = period_data.groupby('Location')[variable].mean().reset_index()
    return period_avg

# Comparar los años del 2008 al 2012 con los 2013 hasta el 2017
period1 = compare_periods(clima_es_df, 2008, 2012, 'MaxTemp')
period2 = compare_periods(clima_es_df, 2013, 2017, 'MaxTemp')

# Merge para comparación
comparison = period1.merge(period2, on='Location', suffixes=('_08-12', '_13-17'))

# Mostrar los resultados
print("Comparación de temperaturas máximas entre los años 2008-2012 y 2013-2017:")
print(comparison)

# Graficar la comparación de temperaturas máximas
comparison.set_index('Location')[['MaxTemp_08-12', 'MaxTemp_13-17']].plot(kind='bar', figsize=(14, 7))

# Personalizar el gráfico
plt.title('Comparación de Temperaturas Máximas Promedio por Período')
plt.xlabel('Región')
plt.ylabel('Temperatura Máxima Promedio (°C)')
plt.xticks(rotation=90)
plt.legend(title='Período')
plt.tight_layout()

# Mostrar el gráfico
plt.show()
```

Comparación de temperaturas máximas entre los años 2008-2012 y 2013-2017:

	Location	MaxTemp_08-12	MaxTemp_13-17
0	Adelaide	22.476879	23.314686
1	Albany	20.053394	20.087155
2	Albury	22.139424	23.085181
3	AliceSprings	28.619078	29.808836
4	BadgerysCreek	23.588109	24.421717
5	Ballarat	17.890356	18.640485
6	Bendigo	21.143566	22.047788
7	Brisbane	25.962555	26.929036
8	Cairns	29.473305	29.634929
9	Canberra	20.373738	21.481417
10	Cobar	25.271571	26.391843
11	CoffsHarbour	23.526896	24.350440
12	Dartmoor	19.271803	20.051150
13	Darwin	32.370455	32.708955
14	GoldCoast	25.592232	25.922818
15	Hobart	17.601326	18.126182
16	Launceston	18.784955	19.106231
17	Melbourne	20.755303	20.789096
18	MelbourneAirport	20.088000	20.843257
19	Mildura	24.234357	25.368738
20	Moree	26.155857	27.642014
21	MountGambier	19.581901	20.049439
22	MountGinini	11.153841	12.101493
23	Newcastle	23.761890	24.408430
24	NorahHead	22.173077	22.947995
25	NorfolkIsland	21.614857	22.017786
26	Nuriootpa	21.412213	21.956841
27	PearceRAAF	25.902097	26.179751
28	Penrith	24.357102	25.121420
29	Perth	24.879659	25.187508
30	PerthAirport	25.618929	25.458670
31	Portland	17.811079	18.267185
32	Richmond	23.816643	25.022194
33	Sale	20.052966	20.481666

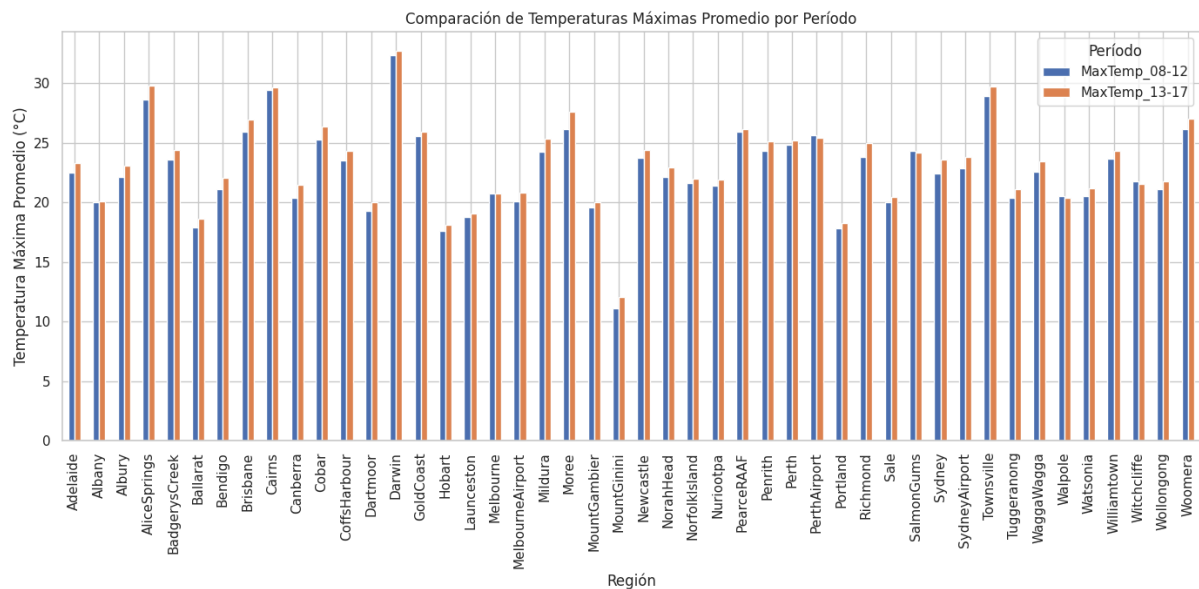


Figura 11. Comparación de temperatura de dos periodos de tiempo.

Finalmente, podemos concluir que con base en los datos analizados, los patrones o la tendencia de la temperatura nos indica un aumento en gran parte de australia, en promedio

de 2°C aproximadamente, esto se puede deber a diversos factores como podría ser el cambio climático, o en algunos casos especiales, los datos podrían representar eventos catastróficos como incendios forestales, los cuales podrían tener como factor fundamental las sequías presentadas en un lapso de tiempo, de igual manera podemos enlazar las temperaturas promedio con factores geográficos, dado que es lógico que las localidades de lado del polo sur sean menos calientes.

En cuestiones de precipitaciones, solo con la gráfica de la tendencia anual, podemos observar que hay fechas donde no llueve mucho, pero en los últimos años después de una sequía, las precipitaciones aumentaron considerablemente, nuevamente esto podría estar directamente relacionado con el cambio climático, ya que las características que este genera podrían ocasionar un ambiente idóneo para las lluvias.