Aram Puig Capdevila, Data Science - IT Academy
14 de julio del 2021

Objetivo: ¿Cuándo se prevén más lluvias?

Hacer un análisis y recomendación sobre las precipitaciones de la ciudad de Barcelona basados en sus datos históricos.

Guión:

1. Obtener los datos necesarios y estructurarlos (https://opendata-ajuntament.barcelona.cat/data/ca/dataset/precipitacio-hist-bcn/resource/5da03f48-020e-4f46-9199-a919feac2034?inner_span=True) i https://www.meteo.cat/wpweb/climatologia/serveis-i-dades-climatiques/serie-climatica-historica-de-barcelona/.

1. Guardar los datos de manera estructurada en formato SQL.

1. Realizar un plan sobre (máximo una página) sobre el tipo de análisis a realizar y cómo se presentarán los resultados.

1. Utilizar PANDAS, NUMPY, MATPLOTLIB (u otras librerías) junto a un cuaderno JUPYTER para la presentación del trabajo realizado.

1. Subir el resultado final a un repositorio privado de Github y compartirlo una vez finalizado con el tutor.

Tiempo estimado: 2 días.

Debe teber:

- El plan sobre el análisis a realizar.
- Incluir un análisis supervisado y uno no supervisado.

A considerar:

- Intentar no atascarse en ningún punto. Se evaluará el alcance del proyecto de manera global.
- No se busca la perfección en la ejecución del ejercicio, pero obtener información sobre las limitaciones y conocimientos de los candidatos para ofrecer feedback.

In [1]:
```python
# Librerias básicas
import pandas as pd
import numpy as np
import re
from datetime import datetime
# Librerias visualización
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
import seaborn as sns
from pandas_profiling import ProfileReport

#Otras
import warnings

pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
pd.set_option('display.expand_frame_repr', False)

warnings.filterwarnings('ignore')
```

In [2]:
```python
df = pd.read_csv('precipitacionsbarcelonadesde1786_format_long.csv')
```

In [3]:
```python
df
```

Out[3]:

	Any	Mes	Desc_Mes	Precipitacions
0	1786	1	Gener	32.8
1	1786	2	Febrer	28.4
2	1786	3	Març	84.4
3	1786	4	Abril	42.3
4	1786	5	Maig	8.5
...	...	...	...	...
2815	2020	8	Agost	12.4
2816	2020	9	Setembre	60.2
2817	2020	10	Octubre	23.1
2818	2020	11	Novembre	52.5
2819	2020	12	Desembre	41.5

2820 rows × 4 columns

1. Introducción

Las precipitaciones varían de acuerdo a ciertos ciclos temporales determinados por los movimientos de rotación y traslación terrestres y por la localización astronómica o geográfica del lugar de que se trate. Esos ciclos pueden ser: diarios, mensuales o estacionales o en ciclos anuales, en efecto, siempre hay meses en que las precipitaciones son mayores que en otros.

Para poder evaluar correctamente las características objetivas del clima, en el cual la precipitación, y en especial la lluvia, desempeña un papel muy importante, las precipitaciones mensuales deben haber sido observadas por un período de por lo menos 20 a 30 años, lo que se llama un período de observación largo.

La variación estacional de las precipitaciones, en especial de la lluvia, define el año hidrológico. Este da inicio en el mes siguiente al de menor precipitación media de largo período.

```
Fuentes: 1.
https://es.wikipedia.org/wiki/Precipitaci%C3%B3n_(meteorolog%C3%ADa)#Origen_de_la_precipitaci%C3%B3n
```

1.1 Características del clima de Barcelona y sus precipitaciones

La ciudad de Barcelona posee un clima mediterráneo con influencias marítimas. De acuerdo con los criterios de la clasificación de Köppen-Geiger la ciudad se encuentra próxima a una zona de transición del clima subtropical de veranos secos y calurosos Csa (clima mediterráneo) a un clima subtropical húmedo Cfa.

Una de las singularidades del clima mediterráneo asociadas al ciclo del agua es que la estación más húmeda es el otoño. Esto se debe a la inestabilidad generada por las diferencias de temperatura entre la superficie del mar Mediterráneo y la atmósfera. durante el otoño, las aguas del mar que se han calentado durante todo el verano, entran en contacto con el aire frío. Esta diferencia térmica es la responsable de los chubascos intensos que en pocas horas dejan grandes cantidades de agua.

Los recurrentes episodios de sequía y de lluvias intensas son un hecho natural y típico de nuestra climatología mediterránea. En el futuro, según las proyecciones climáticas se prevé que el efecto que tendrá el cambio climático sobre el ciclo del agua en general, hará más acusada esta problemática de episodios extremos.

La precipitación media anual en Barcelona se sitúa en torno a los 600 mm, con un máximo de precipitaciones de fin de verano y principio de otoño (llegando a superar los 90 mm de media en octubre), que es originado a menudo por el fenómeno conocido como gota fría, que ha llegado a superar en numerosas ocasiones los 100 mm en un día.

La gota fría, DANA (depresión aislada en niveles altos) o baja segregada, es un fenómeno meteorológico anual que suele coincidir con el inicio del otoño y la primavera en el Mediterráneo occidental. Se experimenta particularmente en España y más concretamente a lo largo de la costa este y las islas Baleares, aunque sus efectos pueden sentirse en zonas interiores también. A grandes rasgos, la gota fría es el resultado de un frente de aire polar frío (corriente en chorro) que avanza lentamente sobre Europa occidental a gran altura (normalmente 5-9 km) y que, al chocar con el aire más cálido y húmedo del Mar Mediterráneo, genera fuertes y dañinas tormentas.

Por el contrario, el mínimo se produce al comienzo del verano, llegando a la media algo por encima de los 20 mm en julio. La humedad media anual es alta debido a las condiciones marítimas de la ciudad, situándose entre 69 y 70 % y variando poco a lo largo del año.

```
Fuentes: 1. https://es.wikipedia.org/wiki/Barcelona#Clima
         2. https://www.barcelona.cat/barcelona-pel-
clima/sites/default/files/documents/cap08_cicle_aigua-20180227.pdf
         3. https://es.wikipedia.org/wiki/Gota_fr%C3%ADa
```

1.2 Efectos del cambio climático

El **Grupo Intergubernamental de Expertos sobre Cambio Climático** (más conocido por sus siglas en inglés, **IPCC**) es una entidad científica creada en 1988 por la Organización Meteorológica Mundial (OMM) y el Programa de las Naciones Unidas para el Medio Ambiente (PNUMA). El IPCC concluye que el calentamiento del sistema climático es inequívoco observado cambios en el sistema climático que no tienen precedente, tanto si se comparan con registros históricos observacionales, que datan de mediados del siglo XIX, como si se comparan con registros paleoclimáticos referidos a los últimos milenios. Las observaciones permiten afirmar que la atmósfera y los océanos se han calentado, que la cantidad y extensión de las masas de hielo y nieve han disminuido, que el nivel del mar ha subido y que las concentraciones de gases de efecto invernadero han aumentafo (IPCC 2013).

La temperatura media global muestra un incremento de 0,85ºC (entre 0,65 y 1,06ºC) en el período 1880-2012. Cada una de las últimas tres décadas ha sido más calida que todas las anteriores desde 1850, siendo la primera década del siglo XXI la más cálida de todas. **Las tendencias en periodos cortos (entre 10 y 15 años) están muy afectadas por la variabilidad natural**, tal y como sucede, por ejemplo, en los últimos 15 años, en los que la tasa de calentamiento ha sido inferior a la media registrada desde 1951.

La precipitación ha aumentado en las zonas terrestres de latitudes medias del hemisferio norte desde 1950. Se han observado cambios en los episodios extemos desde 1950. El número de días y noches frías ha disminuido y el número de días y noches cálidas ha aumentado a nivel global (IPCC 2013).

[...] Unas emisiones iguales a las tasas actuales o superiores inducirán **cambios en todos los componentes del sistema climático, algunos de ellos sin precedentes en cientos o miles de años. Los cambios tendrán lugar en todas las regiones del globo, incluyendo cambios en la tierra y en el océano, en el ciclo del agua**, en la criosfera, en el nivel del mar, en algunos episodios extremos y en la acidez de los océanos. Muchos de estos cambios persistirán durante muchos siglos. La limitación del cambio climático requerirá reducciones substanciales y sostenidas de las emisiones de CO2 (IPCC 2013).

[...] **En la región Mediterránea** tendrá lugar un incremento de temperatura superior a la media global, más pronunciado en los meses estivales que en los invernales. **Para el escenario RCP 8.5** y para finales del siglo XXI, la región Mediterránea experimentará incrementos medios de temperatura de 3,8ºC y de 6,0ºC en los meses invernales y estivales, respectivamente, y **reducciones medias de precipitación de 12% y 24% en los meses invernales y estivales** , respectivamente. Habrá un aumento de los extremos relacionados con las precipitaciones de origen tormentoso (IPCC 2013).

Los Escenarios RCP (del inglés, Representative Concentration Pathways) son cuatro escenarios de emisiones sobre la evolución estimada de la emisión y concentración de gases de efecto invernadero a la atmósfera durante el siglo XXI, establecidos por el IPCC para la elaboración de su quinto informe de evaluación.

Los cuatro escenarios RCP son: RCP2.6, RCP4.5, RCP6.0 y RCP8.5, cuyo nombre se basa en el posible rango de valores de forzamiento radiativo alcanzado en 2100 (2,6; 4,5 ; 6,0 y 8,5 W / m2, respectivamente). El escenario RCP8.5 sigue el rango más alto de emisiones de gases de efecto invernadero, con concentraciones que crecen rápidamente

```
Fuentes: 1. https://www.miteco.gob.es/es/cambio-climatico/temas/impactos-vulnerabilidad-y-
adaptacion/cap2-tendenciasobservadasyproyeccionesdecambioclimaticosobreespana_tcm30-70204.pdf
```

1.3 Proyecciones de las precipitaciones en Barcelona

El estudio realizado por el Servicio Meteorológico de Cataluña (SMC, 2015) sobre la generación de escenarios climáticos futuros regionalizados para el área metropolitana de Barcelona en una primera fase que incluye el periodo 1971-2050, concluye que para las próximas cuatro décadas, independientemente del escenario de emisiones, se proyecta por toda el área metropolitana **un aumento de temperatura** alrededor de 1ºC (horquilla de 0.8 a 1.1ºC) respecto al valor actual; **situándose 2050 entre 1 y 2ºC por encima de la media del periodo 1971-2000** . este incremento afecta tanto a las temperaturas media como la máxima y la mínima, siendo sensiblemente mayor en los valores mínimos. Estacionalmente, el mayor incremento tendrá lugar en verano.

Respecto a la precipitación, las simulaciones muestran un rango de variación más amplio, con una tendencia menos clara, sobre todo a escala estacional . A grandes rasgos, se espera una ligera disminución que podría alcanzar los 55 mm, pero con una alta variabilidad interanual. Habrán pocos cambios o un aumento moderado en invierno y en verano, y una disminución marcada en primavera. Los extremos de precipitación aumentarán de manera apreciable, esperándose episodios de lluvias intensas.

```
Fuentes: 1. https://www.amb.cat/es/web/ecologia/actualitat/publicacions/detall/-/publicacio/efectes-del-
canvi-climatic-en-el-cicle-de-l-aigua-de-l-area-metropolitana/5773115/11818
```

1.4 Unidades de medida y historial de registros

La unidad de medida que usaremos para las precipitaciones acumuladas mensuales son los milímetros (mm). En el caso del agua, existe una correspondecia entre 1 mm y 1 l/m2, porque un litro en un cubo de un metro de ancho y un metro de largo ocupa en volumen exactamente un milímetro de altura. Un cubo de 1 metro cúbico (un metro de alto, uno de ancho y uno de alto) tiene una capacidad de 1000 litros. Si un metro de altura son 1000 milímetros (mm), entonces 1 mm corresponderá, por tanto, a un litro.

```
1 mm = 1 l/m2
```

Las precipitaciones en forma de nieve también se expresan en cm (centímetros) en lugar de milímetros:

```
1 cm nieve =  10 mm nieve
```

Se considera que, aproximadamente, el espesor de la nieve es 10 veces el del agua. Por tanto, 1 cm de nieve equivaldría aproximadamente a 1 mm de agua, aunque esto puede variar de 0,5 a 2 mm en función de la densidad de la nieve.

```
1 cm nieve ≈ 1 mm lluvia
```

Historial de registros

Tenemos los datos de las precipitaciones acomuladas (mm) en la ciudad de Barcelona para cada mes desde enero de 1786 hasta diciembre de 2020.

In [4]:
```python
df.isna().sum()
```

Out[4]:
```
Any               0
Mes               0
Desc_Mes          0
Precipitacions    0
dtype: int64
```

1.5 Estructuración de los datos

In [5]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2820 entries, 0 to 2819
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Any             2820 non-null   int64
 1   Mes             2820 non-null   int64
 2   Desc_Mes        2820 non-null   object
 3   Precipitacions  2820 non-null   float64
dtypes: float64(1), int64(2), object(1)
memory usage: 88.2+ KB
```

Añadimos la variable día con todos los valores en el día 1, para poder convertir las columnas de Año y Mes a formato 'datetime'. NO significa que los datos fueran recogidos al día uno de cada mes, es sólo para tener esta parte de la fecha. Pasamos las tres variables de fecha a string para juntarlas en una sola columna y a continuación pasarla a datetime. Así podremos trabajar con time series.

In [6]:
```python
a = [1 for x in range(len(df))]
```

In [7]:
```python
df['Dia'] = a
```

In [8]:
```python
df['Dia'] = df['Dia'].astype(str)
```

In [9]:
```python
df['Any'] = df.Any.apply(str)
```

In [10]:
```python
df['Mes'] = df.Mes.apply(str)
```

In [11]:
```python
df['Date'] = df['Any'] + '-' + df['Mes'] + '-' + df['Dia']
```

In [12]:
```python
df['Date']
```

Out[12]:
```
0        1786-1-1
1        1786-2-1
2        1786-3-1
3        1786-4-1
4        1786-5-1
          ...
2815     2020-8-1
2816     2020-9-1
2817    2020-10-1
2818    2020-11-1
2819    2020-12-1
Name: Date, Length: 2820, dtype: object
```

In [13]:
```python
pd.to_datetime(df['Date'])
```

Out[13]:
```
0      1786-01-01
1      1786-02-01
2      1786-03-01
3      1786-04-01
4      1786-05-01
          ...
2815   2020-08-01
2816   2020-09-01
2817   2020-10-01
2818   2020-11-01
2819   2020-12-01
Name: Date, Length: 2820, dtype: datetime64[ns]
```

In [14]:
```python
df.drop(['Dia'], 1, inplace=True)
```

In [15]:
```python
df = df[['Date', 'Any', 'Mes', 'Desc_Mes', 'Precipitacions']]
```

In [16]:
```python
df.set_index('Date', inplace=True)
```

In [17]:
```python
df['Any'] = df['Any'].astype(int)
df['Mes'] = df['Mes'].astype(int)
```

In [18]:
```python
df.to_csv('data_precipitacions.csv')
```

2. Conexión a MySql

Pasamos los datos a MySql, creando la base de datos relacional y las tablas correspondientes, e inyectamos los datos directamente desde el Workbench. Nos conectamos a ella mediante python y accedemos a los datos.

In [19]:
```python
import mysql.connector
from getpass import getpass
from mysql.connector import connect, Error
```

In [20]:
```python
# connection.close()
```

In [23]:
```python
connection = connect(
        host="localhost",
        user=input("Enter username: "),
        password=getpass("Enter password: "),
        database = 'precipitacions'
    )
```

```
Enter username: root
Enter password: ········
```

In [24]:
```python
cursor = connection.cursor()
```

Comprobamos que funcionen la connexión y los datos cargados a MySQL. Los insertamos en un dataframe:

In [25]:
```python
select_clients_query = "SELECT * FROM hist_bcn"

cursor.execute(select_clients_query)
result = cursor.fetchall()
```

In [26]:
```python
result[:5]
```

Out[26]:
```
[(1, '1786', '1', 'Gener', '32.8'),
 (2, '1786', '2', 'Febrer', '28.4'),
 (3, '1786', '3', 'MarÃ§', '84.4'),
 (4, '1786', '4', 'Abril', '42.3'),
 (5, '1786', '5', 'Maig', '8.5')]
```

In [28]:
```python
columns = 'ind', 'Any', 'Mes', 'Mes_desc', 'Precipitacions'
```

In [29]:
```python
sql_df = pd.DataFrame(result, columns=columns)
```

In [30]:
```python
sql_df
```

Out[30]:

	ind	Any	Mes	Mes_desc	Precipitacions
0	1	1786	1	Gener	32.8
1	2	1786	2	Febrer	28.4
2	3	1786	3	MarÃ§	84.4
3	4	1786	4	Abril	42.3
4	5	1786	5	Maig	8.5
...	...	...	...	...	...
2815	2816	2020	8	Agost	12.4
2816	2817	2020	9	Setembre	60.2
2817	2818	2020	10	Octubre	23.1
2818	2819	2020	11	Novembre	52.5
2819	2820	2020	12	Desembre	41.5

2820 rows × 5 columns

3. Exploración

In [31]:
```python
profile = ProfileReport(df, title="Pandas Profiling Report", explorative=True)
```

In [28]:
```python
profile
```

Pandas Profiling Report   Overview   Variables   Interactions   Correlations   Missing values   Sample

Overview

Overview   Warnings 6   Reproduction

Dataset statistics

Number of variables	5
Number of observations	2820
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	710.2 KiB
Average record size in memory	257.9 B

Variable types

Categorical	2
Numeric	3

Variables

Date
Categorical

Distinct	2820
Distinct	100.0%

1857-8-1	1
1799-5-1	1
1945-10-1	1

Out[28]:

3.1 Outliers

In [32]:
```python
from scipy import stats
from scipy.stats import iqr
```

In [33]:
```python
df.drop('Desc_Mes', 1, inplace=True)
```

In [34]:
```python
df.skew()
```

Out[34]:
```
Any               0.000000
Mes               0.000000
Precipitacions    1.871138
dtype: float64
```

In [35]:
```python
z = np.abs(stats.zscore(df))
print(z)
```

```
[[1.72469599 1.59325501 0.36676043]
 [1.72469599 1.30357228 0.46388918]
 [1.72469599 1.01388955 0.77229486]
 ...
 [1.72469599 1.01388955 0.58088516]
 [1.72469599 1.30357228 0.06811145]
 [1.72469599 1.59325501 0.17471041]]
```

In [36]:
```python
outliers = np.where(z>3)
```

In [37]:
```python
outliers
```

Out[37]:
```
(array([  10,   20,   50,   57,   94,  296,  692,  704,  764,  808,  860,
         920,  969, 1041, 1076, 1340, 1344, 1391, 1461, 1545, 1593, 1630,
        1713, 1875, 1895, 1897, 1989, 2084, 2157, 2231, 2287, 2296, 2316,
        2325, 2374, 2421, 2492, 2504, 2710, 2793, 2794, 2811], dtype=int64),
 array([2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
       dtype=int64))
```

In [38]:
```python
outliers[0]
```

Out[38]:
```
array([  10,   20,   50,   57,   94,  296,  692,  704,  764,  808,  860,
        920,  969, 1041, 1076, 1340, 1344, 1391, 1461, 1545, 1593, 1630,
       1713, 1875, 1895, 1897, 1989, 2084, 2157, 2231, 2287, 2296, 2316,
       2325, 2374, 2421, 2492, 2504, 2710, 2793, 2794, 2811], dtype=int64)
```

In [39]:
```python
len(outliers[0])
```

Out[39]:
```
42
```

De momento sólo los identificamos, aún no decidimos que hacer con ellos

3.2 Estudio de 1786 a 1990

In [40]:
```python
df = pd.read_csv('data_precipitacions.csv')
```

In [41]:
```python
df.set_index('Date', inplace=True)
```

In [42]:
```python
mean_ppl = df['Precipitacions'].mean()
```

In [43]:
```python
mean_ppl
```

Out[43]:
```
49.41450354609924
```

In [44]:
```python
df[:1990-1-1]['Precipitacions'].mean()
```

Out[44]:
```
48.201710261569495
```

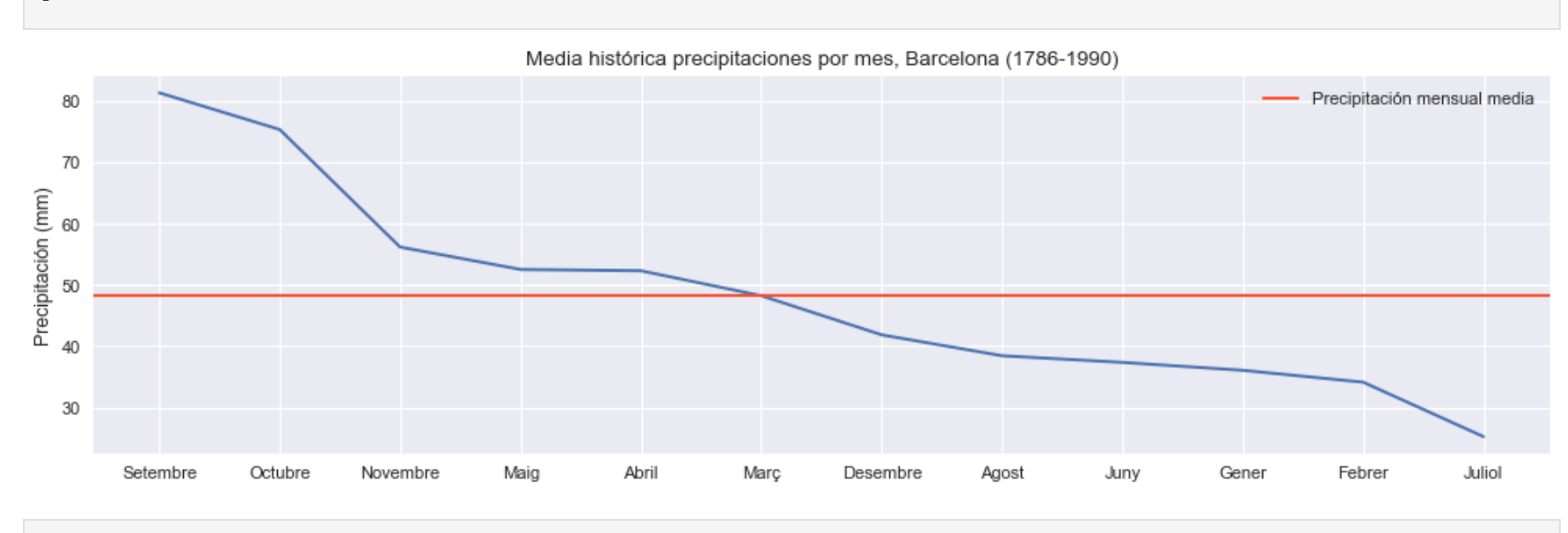
In [45]:
```python
x = round(df.groupby(by='Mes')['Precipitacions'].mean().sum(),2)
print('La media histórica (1786-2020) anual de precipitaciones (mm) es igual a:', x)
```

```
La media histórica (1786-2020) anual de precipitaciones (mm) es igual a: 592.97
```

In [46]:
```python
hist_prep = round(df[:1990-1-1].groupby(by='Mes')['Precipitacions'].mean().sort_values(ascending=False),2)
hist_prep
```

Out[46]:
```
Mes
9     81.31
10    75.32
11    56.15
5     52.50
4     52.29
3     48.22
12    41.84
8     38.43
6     37.34
1     36.05
2     34.11
7     25.23
Name: Precipitacions, dtype: float64
```

In [47]:
```python
plt.style.use('seaborn')
figure(figsize=(15, 4), dpi=80)

plt.plot(df[:1990-1-1].groupby(by='Desc_Mes')['Precipitacions'].mean().sort_values(ascending=False))
plt.axhline(df[:1990-1-1]['Precipitacions'].mean(), color ='#fc4f30', label='Precipitación mensual media')

plt.title('Media histórica precipitaciones por mes, Barcelona (1786-1990)')
# plt.xlabel('Mes (numérico)')
plt.ylabel('Precipitación (mm)')

plt.legend()

plt.show()
```



In [48]:
```python
df['Any'] = df['Any'].astype(int)
```

3.3 Estudio de 1990 a 2020

Creamos un df derivado del primero que incluya sólo los datos de los últimos 30 años (1990-2020) para comparar las medias hitóricas con las 'actuales'.

In [49]:
```python
df30 = df[df['Any'] > 1989]
```

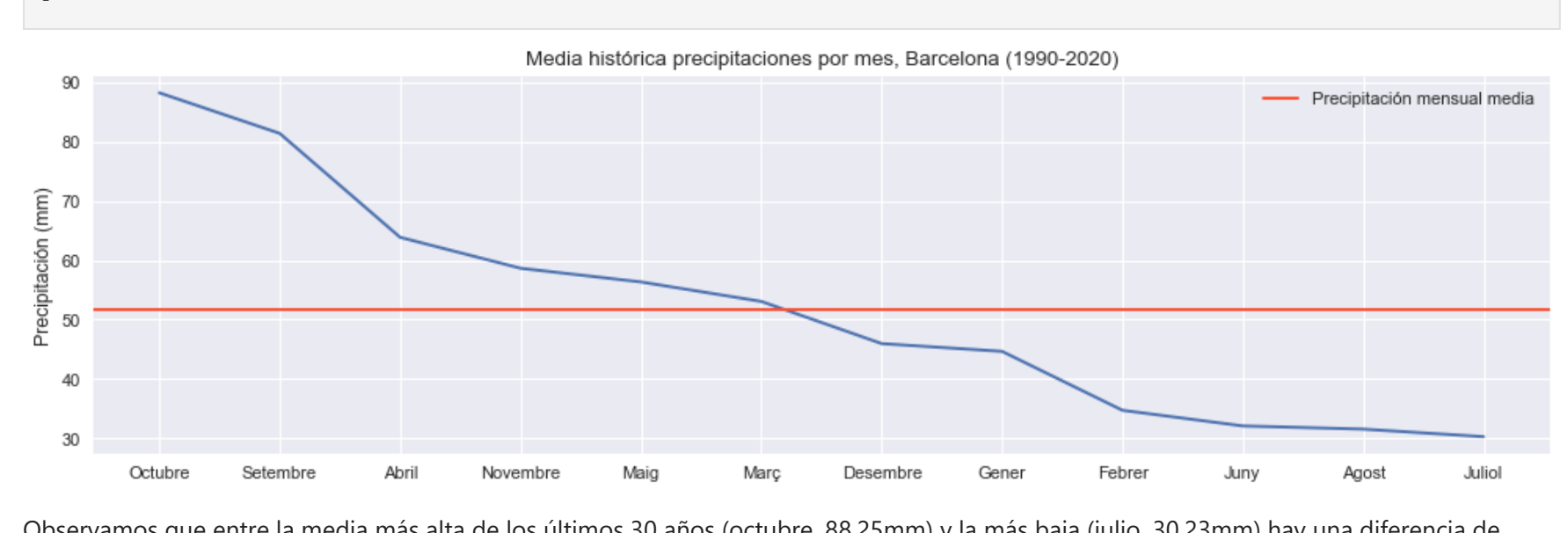
In [50]:
```python
x = round(df30.groupby(by='Mes')['Precipitacions'].mean().sum(),2)
print('La media histórica (1990-2020) anual de precipitaciones (mm) es igual a:', x)
```

```
La media histórica (1990-2020) anual de precipitaciones (mm) es igual a: 620.6
```

In [51]:
```python
prep90 = round(df30.groupby(by='Mes')['Precipitacions'].mean().sort_values(ascending=False),2)
prep90
```

Out[51]:
```
Mes
10    88.25
9     81.40
4     63.87
11    58.65
5     56.34
3     53.05
12    45.94
1     44.63
2     34.68
6     32.05
8     31.51
7     30.23
Name: Precipitacions, dtype: float64
```

In [52]:
```python
plt.style.use('seaborn')
figure(figsize=(15, 4), dpi=80)

plt.plot(df30.groupby(by='Desc_Mes')['Precipitacions'].mean().sort_values(ascending=False))

plt.title('Media histórica precipitaciones por mes, Barcelona (1990-2020)')

plt.axhline(df30['Precipitacions'].mean(), color ='#fc4f30', label='Precipitación mensual media')

plt.ylabel('Precipitación (mm)')

plt.legend()

plt.show()
```



Observamos que entre la media más alta de los últimos 30 años (octubre, 88.25mm) y la más baja (julio, 30.23mm) hay una diferencia de 58.02mm. En octubre hay tres veces las precipitaciones de julio, aproximadamente. Le siguen septiembre, con 81.40mm y abril, ya con cierta diferencia, 63.87mm.

A simple vista podemos dividir las temperaturas medias de los último treinta años en tres grupos:

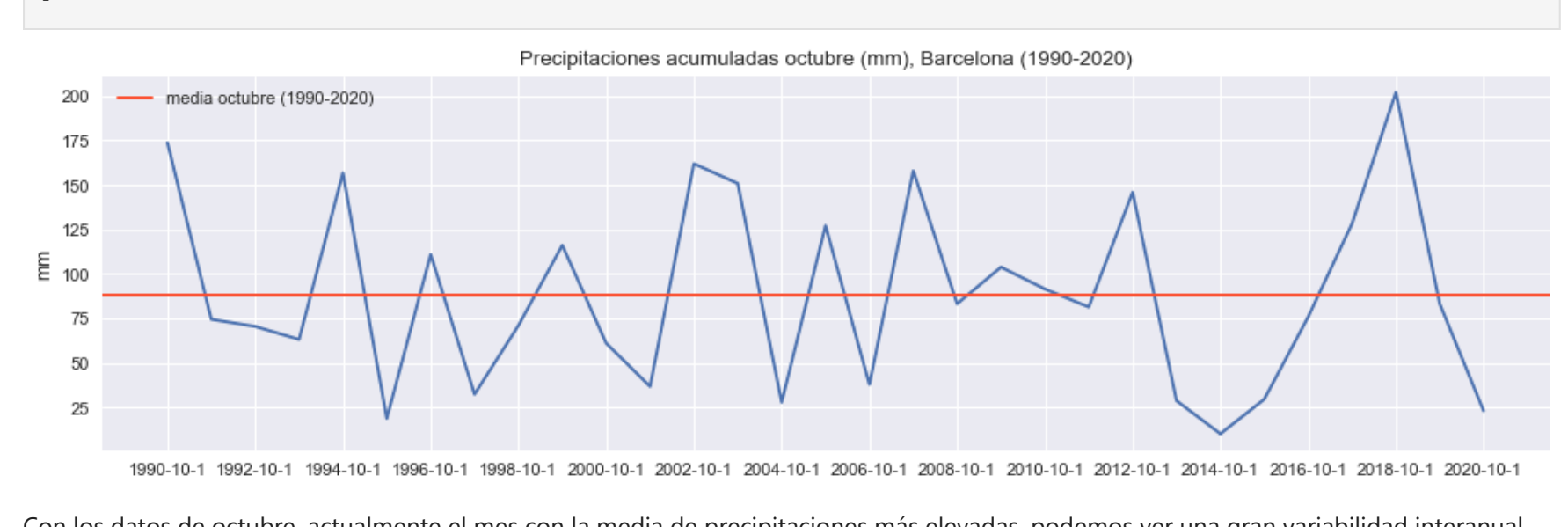
```
                                      1990-2020   1786-1990

Grupo 1, máximas precipitaciones:

                       - Octubre:      88.25       77.80 (2)
                       - Septiembre:   81.40       78.65 (1)

Grupo 2, precipitaciones medianas:

                       - Abril:        63.87       52.30 (5)
                       - Noviembre:    58.65       58.32 (3)
                       - Mayo:         56.34       53.68 (4)
                       - Marzo:        53.05       48.32 (6)
                       - Diciembre:    45.94       42.98 (7)
                       - Enero:        44.63       37.23 (10)

Grupo 3, precipitacioines bajas:

                       - Febrero:      34.68       34.86 (11)
                       - Junio:        32.05       38.28 (9)
                       - Agosto:       31.51       40.51 (8)
                       - Julio:        30.23       25.73 (12)
```

3.3.1 Precipitaciones octubre 1990-2020

In [53]:
```python
df30['Precipitacions'][df['Mes']==10].mean()
```

Out[53]:
```
88.25161290322579
```

In [54]:
```python
plt.style.use('seaborn')
figure(figsize=(15, 4), dpi=80)

plt.plot(df30['Precipitacions'][df['Mes']==10])
plt.axhline(df30['Precipitacions'][df['Mes']==10].mean(), color ='#fc4f30', label='media octubre (1990-2020)')

plt.title('Precipitaciones acumuladas octubre (mm), Barcelona (1990-2020)')
plt.ylabel('mm');

plt.legend()

plt.xticks(np.arange(0, 31, 2))

plt.show()
```

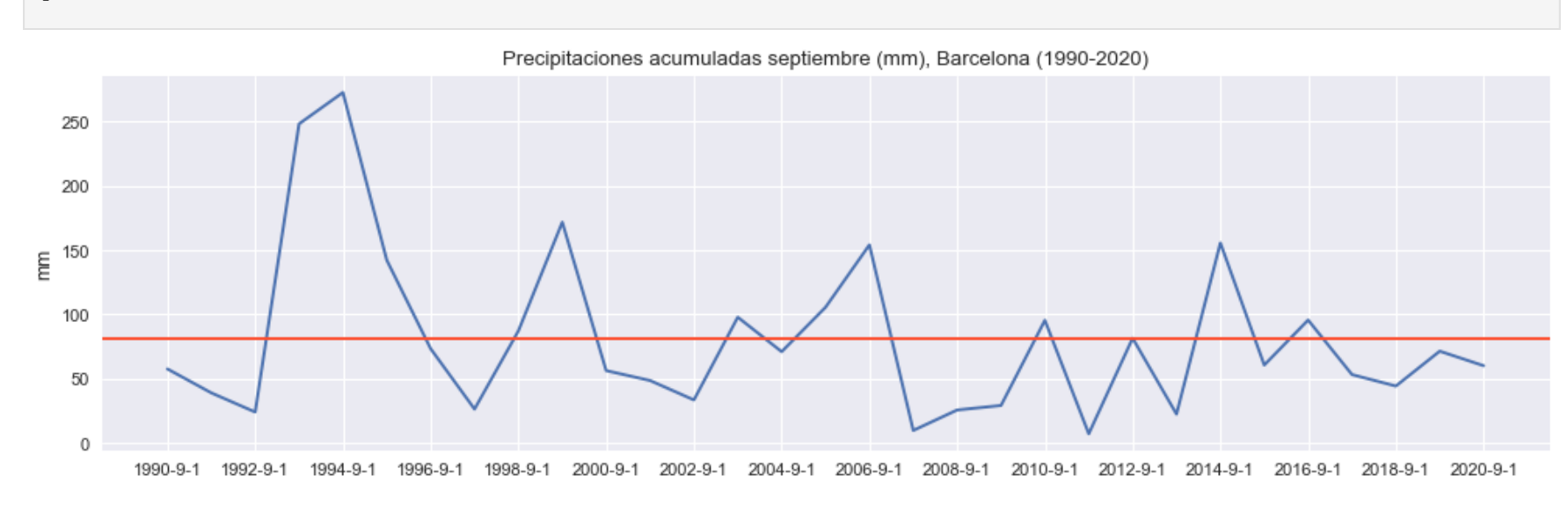


Con los datos de octubre, actualmente el mes con la media de precipitaciones más elevadas, podemos ver una gran variabilidad interanual, habiendo caído el año pasado (2020) tan solo 23.1 mm acumulados y el 2018, 201.9mm.

A parte de su alta variabilidad, podemos apreciar cierto con un año de grandes lluvias y después uno o dos años con pocas precipitacions. No es exacto, pero si hay una especie de variabilidad interanual cíclia. Como veremos a continuación, este patrón aparece con mayor o menor variabilidad en el resto de meses lluviosos. Viendo las sumas anuales, el fenómeno es el mismo, por este motivo hablamos de años lluvios o de sequía.

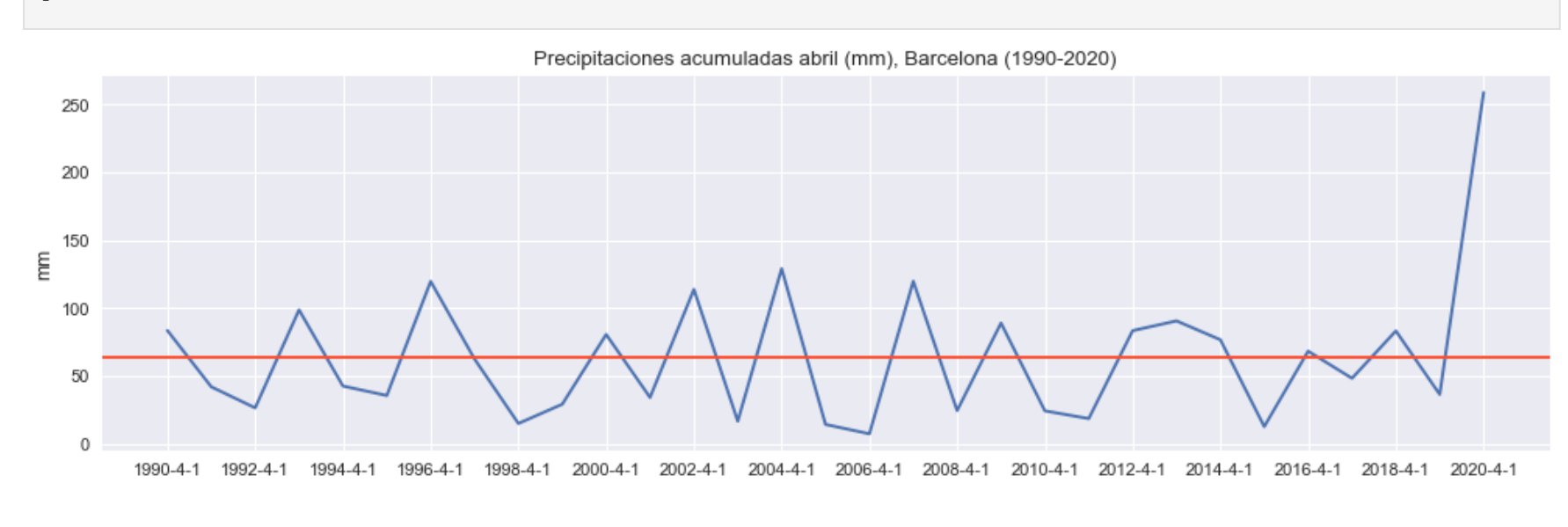
3.3.2 Precipitaciones septiembre 1990-2020

In [55]:
```python
plt.style.use('seaborn')
figure(figsize=(15, 4), dpi=80)

plt.plot(df30['Precipitacions'][df['Mes']==9])
plt.axhline(df30['Precipitacions'][df['Mes']==9].mean(), color ='#fc4f30', label='media septiembre (1990-2020)'

plt.title('Precipitaciones acumuladas septiembre (mm), Barcelona (1990-2020)')
plt.ylabel('mm');

plt.xticks(np.arange(0, 31, 2))

plt.show()
```



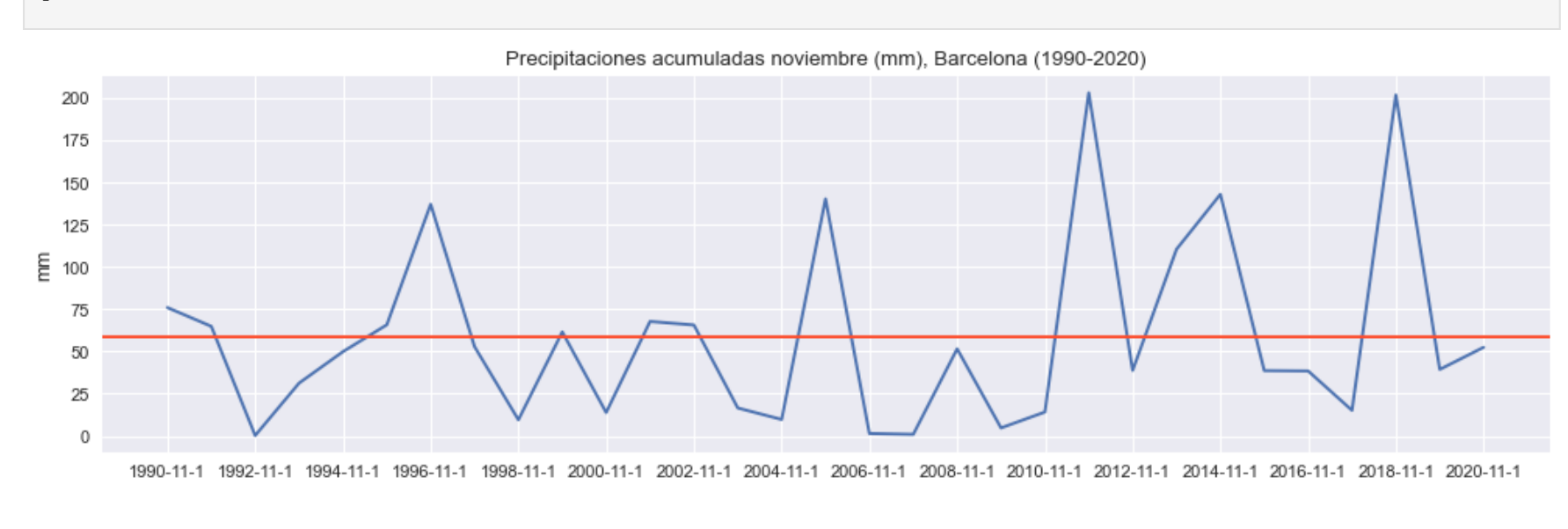
3.3.3 Precipitaciones abril 1990-2020

In [56]:
```python
plt.style.use('seaborn')
figure(figsize=(15, 4), dpi=80)

plt.plot(df30['Precipitacions'][df['Mes']==4])
plt.axhline(df30['Precipitacions'][df['Mes']==4].mean(), color ='#fc4f30', label='media abril (1990-2020)')

plt.title('Precipitaciones acumuladas abril (mm), Barcelona (1990-2020)')
plt.ylabel('mm');

plt.xticks(np.arange(0, 31, 2))

plt.show()
```



3.3.4 Precipitaciones noviembre 1990-2020

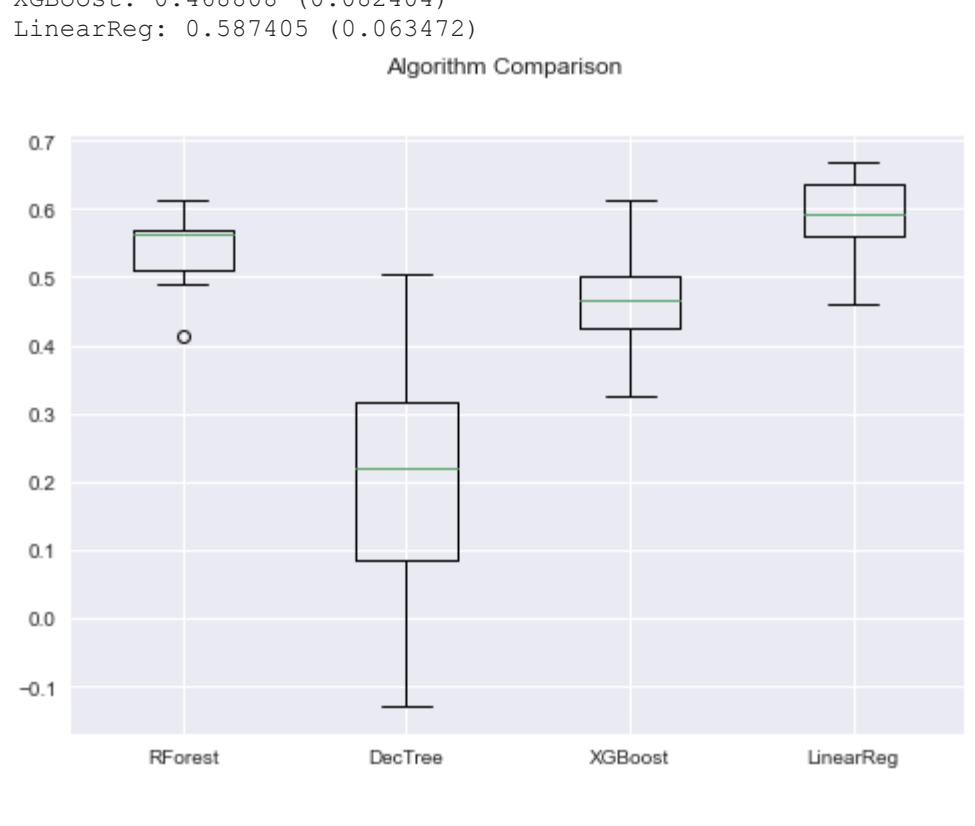
In [57]:
```python
plt.style.use('seaborn')
figure(figsize=(15, 4), dpi=80)

plt.plot(df30['Precipitacions'][df['Mes']==11])
plt.axhline(df30['Precipitacions'][df['Mes']==11].mean(), color ='#fc4f30', label='media noviembre (1990-2020)'

plt.title('Precipitaciones acumuladas noviembre (mm), Barcelona (1990-2020)')
plt.ylabel('mm');

plt.xticks(np.arange(0, 31, 2))

plt.show()
```



3.3.5 Variación interanual

In [117…
```python
models = [('RForest', RandomForestRegressor()),
          ('DecTree', DecisionTreeRegressor()),
          ('XGBoost', xgb.XGBRegressor()),
          ('LinearReg', LinearRegression())]


results = []
names = []

for name, model in models:
    kfold = model_selection.KFold(n_splits=10, random_state=2, shuffle = True)
    cv_results = model_selection.cross_val_score(model, X, y, cv=kfold)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)



fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```

```
RForest: 0.543114 (0.057645)
DecTree: 0.195874 (0.181715)
XGBoost: 0.468808 (0.082404)
LinearReg: 0.587405 (0.063472)
```



5.3 Linear Regression

In [118…
```python
len(df2)
```

Out[118… 852

In [119…
```python
forecast_col ='Precipitacions'
```

In [120…
```python
X = df2.drop([forecast_col],1)
y = df2[forecast_col]
```

In [121…
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

In [122…
```python
steps = [('scaler', StandardScaler()), ('rf', LinearRegression())]
pipeline = Pipeline(steps)
```

In [123…
```python
parameters = {}
```

In [124…
```python
grid = GridSearchCV(pipeline, param_grid=parameters, cv=5)
```

In [125…
```python
grid.fit(X_train, y_train)
print ("score = %3.4f" %(grid.score(X_test,y_test)))
print (grid.best_params_)
```

```
score = 0.6623
{}
```

In [126…
```python
lr_pred = grid.predict(X_test)
```

In [164…
```python
df_lr=pd.DataFrame({'Actual':y_test, 'Predicted':lr_pred})
df_lr[:10]
```

Out[164…

	Actual	Predicted
1990	88.5	32.573583
2041	17.4	139.723966
2261	39.3	43.443573
2683	0.4	147.712848
2201	26.6	39.537782
2701	24.4	34.730412
2282	10.2	45.124738
2063	35.3	42.293652
2266	4.3	36.972625
2519	84.4	27.963137

In [128…
```python
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, lr_pred).round(2))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, lr_pred).round(2))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, lr_pred)).round(2))
```

```
Mean Absolute Error: 23.66
Mean Squared Error: 855.84
Root Mean Squared Error: 29.25
```

5.4 Random Forest Regressor

In [129…
```python
forecast_col ='Precipitacions'
```

In [130…
```python
X = df2.drop([forecast_col],1)
y = df2[forecast_col]
```

In [131…
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

In [132…
```python
steps = [('scaler', StandardScaler()), ('rf', RandomForestRegressor())]
pipeline = Pipeline(steps)
```

In [133…
```python
parameters = {'rf__n_estimators':[100, 500, 700], 'rf__max_features':['auto',
'sqrt', 'log2']}
```

In [134…
```python
grid = GridSearchCV(pipeline, param_grid=parameters, cv=5)
```

In [135…
```python
grid.fit(X_train, y_train)
print ("score = %3.4f" %(grid.score(X_test,y_test)))
print (grid.best_params_)
```

```
score = 0.5571
{'rf__max_features': 'log2', 'rf__n_estimators': 100}
```

In [136…
```python
rf_pred = grid.predict(X_test)
```

In [153…
```python
df_rf=pd.DataFrame({'Actual':y_test, 'Predicted':rf_pred})
df_rf[:10]
```

Out[153…

	Actual	Predicted
1990	88.5	146.037
2041	17.4	58.685
2261	39.3	21.786
2683	0.4	151.571
2201	26.6	32.710
2701	24.4	30.853
2282	10.2	29.998
2063	35.3	121.351
2266	4.3	43.467
2519	84.4	20.891

In [138…
```python
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, rf_pred).round(2))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, rf_pred).round(2))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, rf_pred)).round(2))
```

```
Mean Absolute Error: 23.82
Mean Squared Error: 948.44
Root Mean Squared Error: 30.8
```

5.4 Neural network

Añadimos variables para intentar mejorar el r2 score de la regresión. Añadimos la temperatura máxima y mínima mensual de Barcelona des de 1950. El df que pasaremos al algoritmo será desde el 1950 a 2020 para no tener nulls.

In [139…
```python
forecast_col ='Precipitacions'
```

In [140…
```python
X = df2.drop([forecast_col],1)
y = df2[forecast_col]
```

In [141…
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

In [142…
```python
scaler = StandardScaler()
scaler.fit(X_train)
StandardScaler(copy=True, with_mean=True, with_std=True)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

In [143…
```python
mlp = MLPRegressor(hidden_layer_sizes=(6,6,6),max_iter=500)
```

In [144…
```python
mlp.fit(X_train,y_train)
```

Out[144… MLPRegressor(hidden_layer_sizes=(6, 6, 6), max_iter=500)

In [145…
```python
mlp.score(X_train,y_train)
```

Out[145… 0.6130842436614741

In [146…
```python
mlp_pred = mlp.predict(X_test)
```

In [150…
```python
df_mlp=pd.DataFrame({'Actual':y_test, 'Predicted':mlp_pred})
df_mlp[:10]
```

Out[150…

	Actual	Predicted
1990	88.5	142.947632
2041	17.4	53.033183
2261	39.3	32.491560
2683	0.4	28.965057
2201	26.6	35.724590
2701	24.4	37.210239
2282	10.2	28.454192
2063	35.3	47.709205
2266	4.3	38.692868
2519	84.4	32.948265

In [154…
```python
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, mlp_pred).round(2))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, mlp_pred).round(2))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, mlp_pred)).round(2))
```

```
Mean Absolute Error: 24.57
Mean Squared Error: 922.53
Root Mean Squared Error: 30.37
```

5.4.1 Hyperparameter tunning

In [155…
```python
mlp = MLPRegressor(max_iter=500)
parameter_space = {
    'hidden_layer_sizes': [(50,50,50), (50,100,50), (100,)],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd', 'adam'],
    'alpha': [0.0001, 0.05],
    'learning_rate': ['constant','adaptive'],
}
```

In [156…
```python
clf = GridSearchCV(mlp, parameter_space, n_jobs=-1, cv=3)
clf.fit(X_train, y_train)
```

Out[156…
```
GridSearchCV(cv=3, estimator=MLPRegressor(max_iter=500), n_jobs=-1,
             param_grid={'activation': ['tanh', 'relu'],
                         'alpha': [0.0001, 0.05],
                         'hidden_layer_sizes': [(50, 50, 50), (50, 100, 50),
                                                (100,)],
                         'learning_rate': ['constant', 'adaptive'],
                         'solver': ['sgd', 'adam']})
```

In [157…
```python
# Best parameter set
print('Best parameters found:\n', clf.best_params_)
```

```
Best parameters found:
 {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'adaptive', 'solver': 'sgd'}
```

In [158…
```python
# All results
means = clf.cv_results_['mean_test_score']
stds = clf.cv_results_['std_test_score']
for mean, std, params in zip(means, stds, clf.cv_results_['params']):
    print("%0.3f (+/-%0.03f) for %r" % (mean, std * 2, params))
```

```
0.595 (+/-0.111) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'constant', 'solver': 'sgd'}
0.158 (+/-0.066) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'constant', 'solver': 'adam'}
0.573 (+/-0.155) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.178 (+/-0.049) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'adaptive', 'solver': 'adam'}
0.595 (+/-0.107) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'constant', 'solver': 'sgd'}
0.185 (+/-0.030) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'constant', 'solver': 'adam'}
0.583 (+/-0.107) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.160 (+/-0.057) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'adaptive', 'solver': 'adam'}
0.578 (+/-0.116) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'sgd'}
0.562 (+/-0.100) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'adam'}
0.576 (+/-0.127) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.563 (+/-0.105) for {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'adaptive', 'solver': 'adam'}
0.592 (+/-0.134) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'constant', 'solver': 'sgd'}
0.156 (+/-0.066) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'constant', 'solver': 'adam'}
0.601 (+/-0.075) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.159 (+/-0.035) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'adaptive', 'solver': 'adam'}
0.601 (+/-0.098) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'constant', 'solver': 'sgd'}
0.174 (+/-0.044) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'constant', 'solver': 'adam'}
0.420 (+/-0.599) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.178 (+/-0.047) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'adaptive', 'solver': 'adam'}
0.578 (+/-0.127) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'sgd'}
0.562 (+/-0.102) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'adam'}
0.573 (+/-0.129) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (100,), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.563 (+/-0.105) for {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (100,), 'learning_rate': 'adaptive', 'solver': 'adam'}
0.209 (+/-1.039) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'constant', 'solver': 'sgd'}
0.584 (+/-0.107) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'constant', 'solver': 'adam'}
0.581 (+/-0.106) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.583 (+/-0.119) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'adaptive', 'solver': 'adam'}
0.571 (+/-0.113) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'constant', 'solver': 'sgd'}
0.592 (+/-0.103) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'constant', 'solver': 'adam'}
0.561 (+/-0.101) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.592 (+/-0.112) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'adaptive', 'solver': 'adam'}
0.598 (+/-0.108) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'sgd'}
0.592 (+/-0.105) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'adam'}
0.611 (+/-0.102) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.590 (+/-0.107) for {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'adaptive', 'solver': 'adam'}
0.553 (+/-0.162) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'constant', 'solver': 'sgd'}
0.585 (+/-0.094) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'constant', 'solver': 'adam'}
0.575 (+/-0.134) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.592 (+/-0.098) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'adaptive', 'solver': 'adam'}
0.558 (+/-0.112) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'constant', 'solver': 'sgd'}
0.582 (+/-0.113) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'constant', 'solver': 'adam'}
0.559 (+/-0.103) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.586 (+/-0.096) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'adaptive', 'solver': 'adam'}
0.589 (+/-0.104) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'sgd'}
0.590 (+/-0.105) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'adam'}
0.600 (+/-0.116) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (100,), 'learning_rate': 'adaptive', 'solver': 'sgd'}
0.590 (+/-0.102) for {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (100,), 'learning_rate': 'adaptive', 'solver': 'adam'}
```

Los mejores resultados del hyperparameter tunning no son diferentes de los que ya teníamos.

5.5 Scores

In [159…
```python
columns = ['model', 'R2_score', 'MAE', 'MSE', 'RMSE']
```

In [160…
```python
data = {'model':['LinearRegression', 'RandomForestRegressor', 'MLPRegressor'],
        'R2_score':[0.6187, 0.6115, 0.6286],
        'MAE':[26.05,24.69,23.08],
        'MSE':[1262.32,1153.99,908.52],
        'RMSE':[35.53,33.97,30.14]}
```

In [161…
```python
results = pd.DataFrame(data)
```

In [162…
```python
results.set_index('model', inplace=True)
```

In [163…
```python
results
```

Out[163…

	R2_score	MAE	MSE	RMSE
model				
LinearRegression	0.6187	26.05	1262.32	35.53
RandomForestRegressor	0.6115	24.69	1153.99	33.97
MLPRegressor	0.6286	23.08	908.52	30.14

6. Conclusiones

El primer hecho para evaluar las estimaciones de precipitaciones en un futuro es el estudio de los ciclos pasados. Para ello hemos investigado sobre el clima de Barcelona, sus variaciones históricas y las influencias - más recientes - del cambio climático. Para aproximarnos al futuro hemos estudiado los datos dividiéndolos en dos grupos: de 1786-1990 y 1990-2020.

Tanto en el primer período largo de tiempo cómo en el segundo, y considerando las características del clima de Barcelona, los meses más lluviosos han sido y serán octubre, septiembre, noviembre y abril

Creemos que el ejercicio requiere una hipótesis más definida y un período de predicción determinado. Aún para el caso y sin hipótesis definida, trataremos de determinar cuándo se prevén más lluvias durante los próximos 3 años. Dado que el ejercicio de regresión lineal no ha sido un gran éxito no podemos apoyarnos en sus datos para las predicciones. No obstante, gracias a otros trabajos parecidos, a la información de expertos y nuestra própia exploración trataremos de llegar a algunas conclusiones:

```
1. Tomando como referentes de clima actual y próximo los datos de los último teinta años, podemos
esperar una                  regularidad en los meses con más precipitaciones acumuladas, siendo estos
en orden descendiente: octubre,                   septiembre, abril y noviembre; pero, a su vez, una alta
irregularidad de año a año, acentuada por los efectos del          cambio climático. Del mismo modo
podemos esperar mayores descargas puntuales, tempestas y lluvias intensas.


2. El análisis de los clusters del modelo no-supervisado nos ha acercado a los datos y nos ha dado otro
punto de              referncia para comprender su distribución. Como extra, hemos podido usar las
agrupaciones para mejorar las scores de        los modelos supervisados.

3. El modelo supervisado deja mucho que desear con el MAE más bajo de 23.08, un margen de error
inaceptable ya que hay        meses los cuales, su media histórica de precipitaciones es inferior a esta
métrica. Aún así, estamos satisfechos por        el hecho de haber incrementado en gran medida el R2
score, que inicialmente era de -0.02, hasta 0.628.
```

Para incrementar las predicciones de precipitaciones recomendamos seguir el artículo *"Rainfall forecasting model using machine learning methods: Case study Terengganu, Malaysia"*, donde llegano a conseguir un R2 de 0.999 en la predicción de precipitaciones con modelos supervisados de regresión lineal.

Nos habría gustado profundizar en el estudio de los outliers, así como en el de períodos más reducidos de tiempo, cada 3 o cinco años y el estudio de meses por separado. Hubiéramos probado otros algoritmos de clusterización y habríamos trabajado más el feature engineering: obtención de nuevas variables y datos para aumentar el R2 score.

```
Fuentes: 1. https://www.sciencedirect.com/science/article/pii/S2090447920302069
```