3. ANÁLISIS EXPLORATORIO TRAIN

June 8, 2023

1 Análisis explotario del dataset TRAIN

1.0.1 Limpieza de la base

[1]:
```
# Comenzamos importando las librerías necesarias.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
# Almacenamos los datos del dataset TRAIN en un dataframe.
df = pd.read_csv('UH_2023_TRAIN.txt', delimiter='|')
# Visualizamos las primeras instancias.
df.head()
```

[2]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	TIPO	\
0	14	76953	515	4	660	26	2	0	
1	14	84318	515	4	660	26	2	0	
2	14	85579	340	4	520	32	2	0	
3	14	69671	340	4	520	32	2	0	
4	14	14001	852	14	NaN	81	1	0	

	COLOR	SUPERFICIE	PRODUCCION
0	1	0.0	22215.0
1	1	0.0	22215.0
2	1	0.0	20978.0
3	1	0.0	40722.0
4	1	0.0	14126.0

[3]:
```
# Vemos la dimensión del conjunto de datos.
df.shape
```

[3]:
```
(9601, 11)
```

[4]:
```
# Obtenemos información sobre la base de datos.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9601 entries, 0 to 9600
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   CAMPAÑA      9601 non-null   int64
 1   ID_FINCA     9601 non-null   int64
 2   ID_ZONA      9601 non-null   int64
 3   ID_ESTACION  9601 non-null   int64
 4   ALTITUD      9547 non-null   object
 5   VARIEDAD     9601 non-null   int64
 6   MODO         9601 non-null   int64
 7   TIPO         9601 non-null   int64
 8   COLOR        9601 non-null   int64
 9   SUPERFICIE   9601 non-null   float64
 10  PRODUCCION   8526 non-null   float64
dtypes: float64(2), int64(8), object(1)
memory usage: 825.2+ KB
```

[5]:
```
# Vemos que la variable 'ALTITUD' es de tipo "object". En efecto, existen␣
 ↪entradas en las que no nos
# dan una altura fija, sino que nos dicen que la altura está comprendida entre␣
 ↪un rango de valores.
# Una estrategia sería sustituir las entradas tipo rango de valores por la␣
 ↪media entre los extremos
# de dicho intervalo.
```

[6]:
```
# Comprobamos que todos las entradas de la variable altitud son float o str.
altitudes = np.array(df["ALTITUD"].values)
tipos = np.vectorize(lambda x: isinstance(x, (float, str)))(altitudes)

print(f"Número de entradas que no son de tipo float o str: {len(tipos) - np.
 ↪count_nonzero(tipos)}")
```

```
Número de entradas que no son de tipo float o str: 0
```

[7]:
```
# Comprobamos que las entradas tipo float son datos nulos y que las entradas␣
 ↪tipo str tienen len=3
# o len=7.
lista = [x for x in altitudes if isinstance(x, float) or len(str(x)) not in [3,␣
 ↪7]]
lista
```

[7]:
```
[nan,
 nan,
 nan,
 nan,
 nan,
```

nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,
nan,

```
 nan,
 nan]
```

[8]:
```
# Por tanto:
# Nótese que aquellos str que se podrian leer como enteros (por ejemplo␣
 ↪podriamos leer '660'
# como 660.0) tienen longitud 3.
print(altitudes[0])
print(len(altitudes[0]))
# Los nulos son de tipo float.
print(altitudes[4])
print(type(altitudes[4]))
# Aquellos que han sido introducidos como un rango de valores, tienen longitud␣
 ↪mayor que 3, en
# particular, longitud 7.
print(altitudes[12])
print(len(altitudes[12]))
# Nótese que:
print(altitudes[12][0:3])
print(altitudes[12][4:7])
```

```
660
3
nan
<class 'float'>
650-660
7
650
660
```

[9]:
```
# Por tanto, una manera de imputar un valor numérico a las alturas que vienen␣
 ↪dadas en intervalos
# es asignar la media del intervalo:
altitudes = df["ALTITUD"].to_numpy()
valores_altitud = np.zeros_like(altitudes, dtype=np.float64)

for i in range(len(altitudes)):
    if isinstance(altitudes[i], float):
        valores_altitud[i] = altitudes[i]
    elif len(altitudes[i]) > 3:
        valores_altitud[i] = (float(altitudes[i][:3]) + float(altitudes[i][4:
 ↪7])) / 2
    else:
        valores_altitud[i] = float(altitudes[i])

df["ALTITUD"] = valores_altitud
```

[10]:
```
# Comprobamos que ha funcionado. Teniamos que, la entrada 12 de la variable␣
 ↪altitud en el dataset
# original era 650-660 y ahora:
print(altitudes[12])
print(df.ALTITUD[12])
```

```
650-660
655.0
```

[11]:
```
df.info()
# Vemos que ahora todos los valores de la variable altitud son de tipo float␣
 ↪como queríamos.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9601 entries, 0 to 9600
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   CAMPAÑA      9601 non-null   int64
 1   ID_FINCA     9601 non-null   int64
 2   ID_ZONA      9601 non-null   int64
 3   ID_ESTACION  9601 non-null   int64
 4   ALTITUD      9547 non-null   float64
 5   VARIEDAD     9601 non-null   int64
 6   MODO         9601 non-null   int64
 7   TIPO         9601 non-null   int64
 8   COLOR        9601 non-null   int64
 9   SUPERFICIE   9601 non-null   float64
 10  PRODUCCION   8526 non-null   float64
dtypes: float64(3), int64(8)
memory usage: 825.2 KB
```

[12]:
```
# Estudio de duplicados:
duplicate_rows_df = df[df.duplicated()]
print("numero de duplicados: ", duplicate_rows_df.shape)
# No hay datos duplicados.
```

```
numero de duplicados:  (0, 11)
```

[13]:
```
# Estudio de missings:
df.isnull().sum()
```

[13]:
```
CAMPAÑA        0
ID_FINCA       0
ID_ZONA        0
ID_ESTACION    0
ALTITUD       54
VARIEDAD       0
MODO           0
```

```
TIPO              0
COLOR             0
SUPERFICIE        0
PRODUCCION     1075
dtype: int64
```

[14]:
```
# Vemos que existen datos nulos en la variable "PRODUCCION" y "ALTITUD". Los␣
 ↪datos nulos de la
# variable "PRODUCCION" se deben a que obtener dichos datos es justo el␣
 ↪objetivo del problema
# propuesto. Con respecto a los de la variable "ALTITUD", imputamos por la␣
 ↪altitud media de la
# estación correspondiente a cada finca.
group_means = df.groupby('ID_ESTACION')['ALTITUD'].transform('mean')
df['ALTITUD'] = df['ALTITUD'].fillna(group_means)
```

[15]:
```
# El motivo por el cual se imputa utilizando datos de ID_ESTACION en vez de␣
 ↪ID_ZONA es que la
# mayoría de los missing se acumulan en zonas en las cuales todos los valores␣
 ↪de ALTITUD son missing.
# Además ID_ZONA no siempre indica un territorio contenido en una estación sino␣
 ↪que hay zonas
# correspondientes a diferentes estaciones, como podemos comprobar a␣
 ↪continuación.
result = df.groupby('ID_ZONA')['ID_ESTACION'].nunique()
result.sort_values(ascending = False).head(30)
```

[15]:
```
ID_ZONA
559    3
732    3
506    3
464    3
449    3
86     3
272    2
804    2
292    2
585    2
353    2
698    2
379    2
170    2
428    2
441    2
462    2
468    2
511    2
```

```
254    2
474    2
165    2
885    2
919    2
881    2
724    1
700    1
899    1
677    1
672    1
Name: ID_ESTACION, dtype: int64
```

[16]:
```
# Como se nos indica, nuestro objetivo es predecir la producción del año 22␣
 ↪dados los valores
# de unas variables. Así, lo primero que haremos será construir dos dataframes:
# Uno con los datos de 2014-2021 que será el que emplearemos para construir el␣
 ↪modelo.
# Otro con los datos de 2022 que será del que queramos predecir.
```

[17]:
```
#Creamos un dataframe con los datos que vamos a usar para construir el modelo
#Para ello, identificamos las filas en la que la variable producción toma un␣
 ↪valor nulo y las
# eliminamos.Observamos como solo hay valores nulos de producción en la campaña␣
 ↪22:
df[np.isnan(df['PRODUCCION'])].sort_values('CAMPAÑA')
```

[17]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	TIPO	\
8526	22	48626	302	13	600.0	32	2	0	
9234	22	75037	757	12	470.0	15	2	0	
9235	22	3014	462	19	525.0	17	1	0	
9236	22	97661	462	19	525.0	68	1	0	
9237	22	25015	468	7	605.0	32	2	0	
…	…	…	…	…	…	…	…	…	…
8890	22	86267	839	5	610.0	52	2	0	
8891	22	57440	839	5	610.0	32	2	0	
8892	22	79550	839	5	610.0	15	2	0	
8878	22	73893	839	5	610.0	40	2	0	
9600	22	88928	239	6	700.0	52	2	0	

	COLOR	SUPERFICIE	PRODUCCION
8526	1	3.7503	NaN
9234	1	0.5665	NaN
9235	1	0.4181	NaN
9236	0	3.5200	NaN
9237	1	1.5964	NaN
…	…	…	…

```
8890       1       0.4400          NaN
8891       1       3.7200          NaN
8892       1       2.5369          NaN
8878       1       1.1582          NaN
9600       1       1.6099          NaN

[1075 rows x 11 columns]
```

[18]:
```
# Construimos el dataframe que emplearemos para construir el modelo.
df_train=df.loc[~np.isnan(df['PRODUCCION'])]
df_train
```

[18]:
```
      CAMPAÑA  ID_FINCA  ID_ZONA  ID_ESTACION     ALTITUD  VARIEDAD  MODO  \
0          14     76953      515            4  660.000000        26     2
1          14     84318      515            4  660.000000        26     2
2          14     85579      340            4  520.000000        32     2
3          14     69671      340            4  520.000000        32     2
4          14     14001      852           14  659.097938        81     1
…         …         …        …          …          …       …   …
8521       21     37461      239            6  700.000000        52     2
8522       21     58769      239            6  700.000000        32     2
8523       21     58769      239            6  700.000000        59     2
8524       21     88928      239            6  700.000000        40     2
8525       21     88928      239            6  700.000000        52     2

      TIPO  COLOR  SUPERFICIE  PRODUCCION
0        0      1       0.000     22215.0
1        0      1       0.000     22215.0
2        0      1       0.000     20978.0
3        0      1       0.000     40722.0
4        0      1       0.000     14126.0
…     …     …          …          …
8521     0      1       3.680     28160.1
8522     0      1       4.250     41310.0
8523     0      1       4.160     45420.0
8524     0      1       4.750     56140.0
8525     0      1       1.462     13869.9

[8526 rows x 11 columns]
```

[19]:
```
# Vamos ahora a analizar la producción de uva por variedad. ¿Son todas las␣
 ↪variedades igual de
# importantes en cuanto a producción?
prodvariedad=df.groupby(['VARIEDAD','CAMPAÑA'])['PRODUCCION'].sum()
prodvariedad=prodvariedad.unstack().transpose()
prodvariedad.drop(index=22, axis=0,inplace=True)
prodvariedad
```

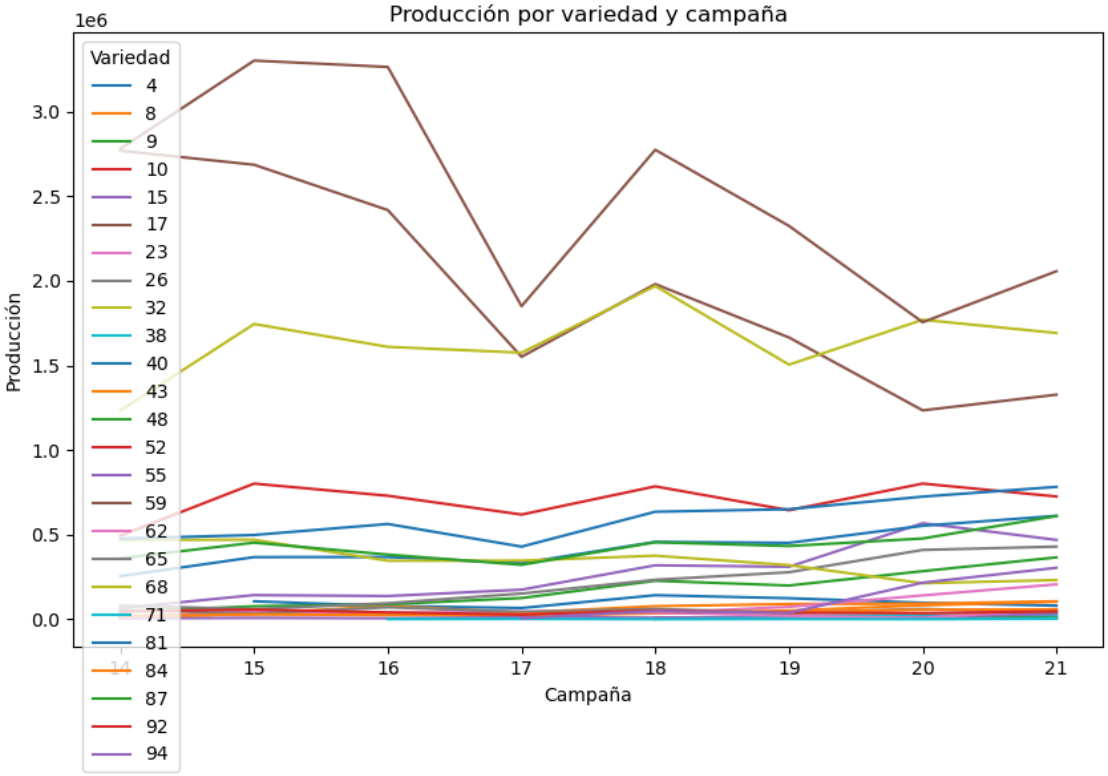
```
#Vemos como hay diferencias notables entre variedades siendo la 17, la 32 y la␣
↪59 las más numerosas.
```

[19]:
```
VARIEDAD        4         8         9        10        15            17  \
CAMPAÑA
14            NaN   10540.0   44560.0       NaN   66260.0  2.768962e+06
15         106930.0   34230.0   77470.0       NaN  143690.0  2.686043e+06
16          79120.0   78900.0   89200.0       NaN  137880.0  2.418128e+06
17          67170.0   40380.0  126070.0       NaN  175650.0  1.551468e+06
18         142960.0   78050.0  228110.0   10490.0  319590.0  1.981065e+06
19         125050.0   90760.0  200180.0    9030.0  310890.0  1.665416e+06
20          98140.0   92800.0  284990.0    7760.0  568260.0  1.234820e+06
21          81360.0  106930.0  366450.0   16890.0  469700.0  1.327676e+06

VARIEDAD        23           26            32       38  …            59  \
CAMPAÑA                                                  …
14            NaN   80930.0000  1.234114e+06      NaN  …  2.780219e+06
15            NaN   60610.0000  1.745344e+06      NaN  …  3.301252e+06
16            NaN   95610.0000  1.610207e+06      NaN  …  3.263780e+06
17            NaN  152700.0000  1.575743e+06      NaN  …  1.850529e+06
18        38000.0  233990.0000  1.969800e+06   2150.0  …  2.774398e+06
19        74010.0  278530.0000  1.505040e+06  21010.0  …  2.324709e+06
20       141800.0  410450.3728  1.770020e+06  23730.0  …  1.755640e+06
21       206960.0  430070.5336  1.692130e+06  20960.0  …  2.057070e+06

VARIEDAD       62       65        68      71           81       84        87  \
CAMPAÑA
14        12560.0  56120.0  469470.0  1240.0  477090.2640  19810.0  360741.6
15        47220.0  33850.0  470580.0     NaN  498701.0700  29440.0  453410.0
16        38910.0  71140.0  346510.0  1970.0  563711.0000  26040.0  382720.0
17        45510.0  44640.0  347230.0  2880.0  429841.0210  32720.0  322440.0
18        39530.0  61570.0  376423.0  1500.0  635950.0000  49410.0  455610.0
19        21260.0  38720.0  320690.0  2290.0  651067.0000  42450.0  433430.0
20         8000.0  61790.0  213750.0  1440.0  725320.4684  56670.0  477950.0
21        60280.0  41840.0  232370.0  3910.0  783002.0000  59270.0  610640.0

VARIEDAD       92        94
CAMPAÑA
14        47210.0       NaN
15        57900.0       NaN
16        41510.0       NaN
17        28360.0    6940.0
18        52380.0   49560.0
19        39120.0   39050.0
20        36520.0  216980.0
21        47390.0  304860.0
```

```
[8 rows x 25 columns]
```

1.0.2 Evolución de la producción por variedad de uva

[20]:
```
#Veamos gráficamente la evolución de la producción por variedad de uva.
prodvariedad.plot(kind='line', figsize=(10, 6))
plt.xlabel('Campaña')
plt.ylabel('Producción')
plt.title('Producción por variedad y campaña')
plt.legend(title='Variedad', loc='upper left')
plt.show()
# Entre las variedades de uva más importantes se observa como las variedades 17 y 59 cuentan
# con una ligera tendencia decreciente hasta equipararse con la variedad 32 que mantiene una
# tendencia constante.
```



[21]:
```
# Vamos a estudiar si la productividad de cada tipo de uva es la misma
# Empezamos trabajando con 2021.
df21=df[df['CAMPAÑA']==21]
df21=df21.groupby(['VARIEDAD']).agg({'SUPERFICIE': 'sum', 'PRODUCCION': 'sum'})
df21['Productividad'] = df21['PRODUCCION']/df21['SUPERFICIE']
```

```
df21.sort_values('PRODUCCION', ascending=False)
# Hay diferencias de productividad considerables entre las variedades 59 y 32,␣
 ↪las dos más producidas.
```

[21]:

VARIEDAD	SUPERFICIE	PRODUCCION	Productividad
59	541.62180	2.057070e+06	3797.983003
32	298.60180	1.692130e+06	5666.844801
17	320.46144	1.327676e+06	4143.012500
81	151.49340	7.830020e+05	5168.555198
52	165.02970	7.260600e+05	4399.571713
40	100.52120	6.117700e+05	6085.979873
87	80.27100	6.106400e+05	7607.230507
15	87.60210	4.697000e+05	5361.743611
26	74.61630	4.300705e+05	5763.761184
9	100.69980	3.664500e+05	3639.034040
94	66.12020	3.048600e+05	4610.693858
68	64.45270	2.323700e+05	3605.279531
23	35.32000	2.069600e+05	5859.569649
8	15.61870	1.069300e+05	6846.280420
43	23.01870	1.044000e+05	4535.442922
4	44.82300	8.136000e+04	1815.139549
62	11.47080	6.028000e+04	5255.082470
84	8.61880	5.927000e+04	6876.827401
92	8.10430	4.739000e+04	5847.513049
65	5.81690	4.184000e+04	7192.834671
55	6.60940	3.332000e+04	5041.304808
38	3.53000	2.096000e+04	5937.677054
10	1.81250	1.689000e+04	9318.620690
48	4.97650	7.640000e+03	1535.215513
71	2.83400	3.910000e+03	1379.675371

[22]:
```
# Ahora trabajamos con el año 2020.
df20=df[df['CAMPAÑA']==20]
df20=df20.groupby(['VARIEDAD']).agg({'SUPERFICIE': 'sum', 'PRODUCCION': 'sum'})
df20['Productividad'] = df20['PRODUCCION']/df20['SUPERFICIE']
df20.sort_values('PRODUCCION', ascending=False)
# De nuevo hay diferencias de productividad significativas entre las variedades␣
 ↪32 y 59.
```

[22]:

VARIEDAD	SUPERFICIE	PRODUCCION	Productividad
32	292.28052	1.770020e+06	6055.895035
59	543.10160	1.755640e+06	3232.618354
17	341.49650	1.234820e+06	3615.908216
52	161.65530	8.023200e+05	4963.153079
81	153.05290	7.253205e+05	4739.018133

```
15          87.01610  5.682600e+05    6530.515617
40         100.39730  5.530700e+05    5508.813484
87          74.32880  4.779500e+05    6430.212784
26          67.39080  4.104504e+05    6090.599500
9           89.46250  2.849900e+05    3185.580551
94          45.34360  2.169800e+05    4785.239813
68          66.55950  2.137500e+05    3211.412345
23          35.90730  1.418000e+05    3949.057712
4           44.07720  9.814000e+04    2226.547966
8           15.61870  9.280000e+04    5941.595651
43          22.00870  8.338000e+04    3788.501820
65           7.12210  6.179000e+04    8675.811909
84           8.61880  5.667000e+04    6575.161275
92           4.36790  3.652000e+04    8360.997276
38           4.22000  2.373000e+04    5623.222749
55           2.36000  1.277000e+04    5411.016949
48           4.97650  1.201000e+04    2413.342711
62          11.47080  8.000000e+03     697.423022
10           1.81250  7.760000e+03    4281.379310
71           2.83400  1.440000e+03     508.115737
```

1.0.3 Evolución de la producción por finca

[23]:
```
# Veamos ahora cuánto produce cada finca.
df_fincas = df_train.groupby(['ID_FINCA', 'CAMPAÑA']).agg({'PRODUCCION': 'sum'})
df_fincas.head(40)
```

[23]:
```
                   PRODUCCION
ID_FINCA CAMPAÑA
200      14          1900.000
         15           778.104
         16          1636.200
         17           829.008
         18           607.212
         19           392.688
         20           545.400
439      14          2215.200
         15          3208.400
         16          6354.400
         21          1901.402
447      14          1824.700
         15          3242.106
         16          2524.284
         17          1336.986
         20          2828.540
         21          2037.340
523      14          2290.400
```

```
          17          3732.000
          18          2836.074
          19          1225.824
          20           947.844
          21           745.122
528       14         22780.000
779       14          2890.000
          15          5190.000
          16          4780.000
          17          3910.000
          18          5337.500
          19          5232.500
797       14         42750.000
          15         53450.000
          16         55930.000
          17         45840.000
          18         39490.000
          19         43360.000
          20         10080.000
          21         21890.000
812       14          4420.000
          15          4150.000
```

[24]:
```
# Veamos cuáles son las fincas con mayores niveles de producción.
df_fincas_total = df_fincas.groupby(['ID_FINCA']).agg({'PRODUCCION': 'sum'})
df_fincas_total = df_fincas_total.sort_values('PRODUCCION', ascending = False)
df_fincas_total
```

[24]:
```
             PRODUCCION
ID_FINCA
49636     2.119196e+06
4024      1.746138e+06
95678     1.706991e+06
48877     1.280985e+06
61177     1.010916e+06
…                  …
41419     1.560000e+02
6532      1.145000e+02
55056     1.129000e+02
27166     1.095150e+02
79118     3.760000e+01

[1193 rows x 1 columns]
```

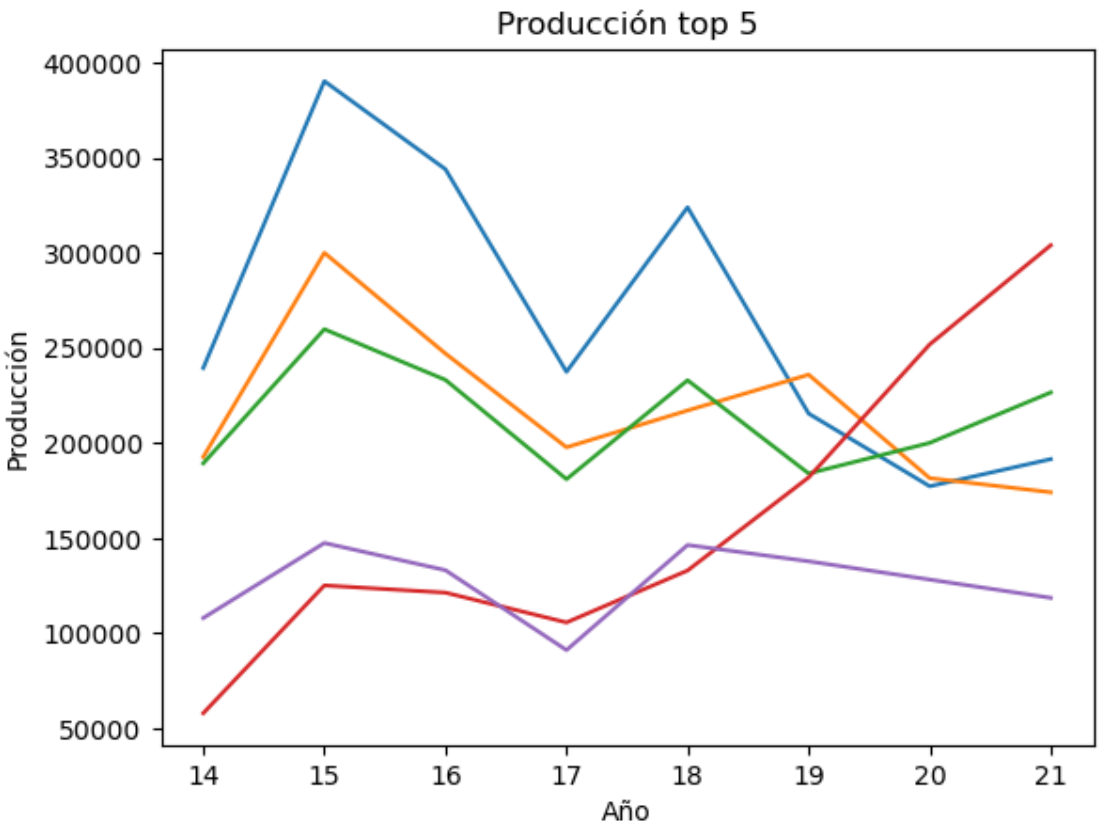
[25]:
```
# Centrémonos en las 5 fincas con mayor producción.
top5_fincas=df_fincas['PRODUCCION'][df_fincas_total.index[:5]]
top5_fincas = top5_fincas.unstack('ID_FINCA')
```

```
top5_fincas
```

[25]:
ID_FINCA	49636	4024	95678	48877	61177
CAMPAÑA					
14	239445.6	192693.1000	189340.000	58014.138	108040.000
15	390200.0	299968.2736	259820.000	125154.539	147460.000
16	343910.0	247038.7540	233182.305	121366.454	133160.000
17	237430.0	197759.8670	180994.758	105696.757	91170.000
18	323910.0	217102.9950	232988.858	132995.121	146406.010
19	215480.0	235914.2020	183985.582	181946.404	137837.820
20	177270.0	181543.5570	200081.277	251842.912	128223.575
21	191550.0	174116.7680	226598.009	303968.571	118618.170

[26]:
```
# Veamos la evolución de la producción de estas 5 fincas.
top5_fincas.plot(legend=None)

plt.title('Producción top 5')
plt.xlabel('Año')
plt.ylabel('Producción')
#La finca que partía en quinta posición cuenta con una tendencia creciente que␣
 ↪la coloca en la
# última campaña como la finca con mayor producción. Las demás fincas no␣
 ↪cuentan con una tendencia
# claramente definida aunque la finca que partía en primera posición tiene una␣
 ↪tendencia ligeramente
# decreciente que la coloca en 2021 en tercera posición.
```

[26]:
```
Text(0, 0.5, 'Producción')
```



[27]:
```
# Parece que puede haber fincas que presentan una crecimiento o decrecimiento␣
 ↪tendencial en la
# producción. Vamos a comprobar si representan una proporción significativa de␣
 ↪la muestra.
# Para ello comenzamos creando el dataframe df_trend, que muestra la producción␣
 ↪por finca a lo
# largo de los años.
df_trend=df_train.groupby(['ID_FINCA', 'CAMPAÑA']).agg({'PRODUCCION':'sum'})
df_trend=df_trend.unstack().transpose()
df_trend
```

[27]:

	ID_FINCA	200	439	447	523	528	779	\
	CAMPAÑA							
PRODUCCION	14	1900.000	2215.200	1824.700	2290.400	22780.0	2890.0	
	15	778.104	3208.400	3242.106	NaN	NaN	5190.0	
	16	1636.200	6354.400	2524.284	NaN	NaN	4780.0	
	17	829.008	NaN	1336.986	3732.000	NaN	3910.0	
	18	607.212	NaN	NaN	2836.074	NaN	5337.5	
	19	392.688	NaN	NaN	1225.824	NaN	5232.5	
	20	545.400	NaN	2828.540	947.844	NaN	NaN	

```
            21            NaN  1901.402  2037.340   745.122      NaN      NaN

ID_FINCA              797     812     917       966    …    98814  \
            CAMPAÑA                                     …
PRODUCCION 14       42750.0  4420.0   698.7       NaN  …  48486.5
            15      53450.0  4150.0   698.7       NaN  …  16100.0
            16      55930.0  5420.0  4569.6    48.015  …  46780.0
            17      45840.0  5260.0  3187.5  2873.140  …   9420.0
            18      39490.0  5780.0     NaN  1122.775  …  50760.0
            19      43360.0  4090.0  3468.0   710.040  …  88860.0
            20      10080.0  3800.0     NaN       NaN  …  13630.0
            21      21890.0  3560.0  2250.0  1101.435  …  38570.0

ID_FINCA                98825     98871     98995   99033    99108   99146  \
            CAMPAÑA
PRODUCCION 14             NaN   895.000  5675.000  2284.2   4520.0     NaN
            15            NaN  1205.000  1880.000     NaN  11900.0  6480.0
            16            NaN  1890.000  3902.858     NaN   7510.0  4080.0
            17            NaN   560.000  3743.395     NaN   5300.0  6060.0
            18            NaN   885.000  4967.232     NaN   5750.0  3700.0
            19            NaN   765.000  3353.477     NaN   3300.0  3380.0
            20            NaN       NaN  3807.528     NaN   6140.0  3300.0
            21      13783.602   676.872  3308.716     NaN   4490.0  4730.0

ID_FINCA              99282   99377      99693
            CAMPAÑA
PRODUCCION 14       6630.663     NaN  16856.590
            15      8000.800  2280.0  14480.844
            16      9230.000  1550.0  15931.125
            17      5840.000     NaN  20130.201
            18      9070.000  2160.0  17597.034
            19      7380.000  1840.0  18405.387
            20      6710.000  2300.0  26876.300
            21      8460.000  2460.0  35418.700

[8 rows x 1193 columns]
```

[28]:
```
# La idea inicial es llevar a cabo una regresión lineal de la producción a lo
 ↪largo del tiempo.
# La pendiente de la regresión indicará cuánto aumenta la producción por finca
 ↪cada año.
# Comenzaremos trabajando con aquellas fincas que han producido durante todas
 ↪las campañas
#presentes en el dataset por lo que eliminamos las fincas con valores NaN.
df_trend=df_trend.dropna(axis=1)
df_trend
```

[28]:
```
ID_FINCA                797     812     1142       1190       1270     1777   \
          CAMPAÑA
PRODUCCION 14        42750.0  4420.0  2106.5  18222.600   4820.000  514.800
           15        53450.0  4150.0  2128.5  20662.400   9530.000  486.200
           16        55930.0  5420.0  1086.0  27137.000  13305.835  596.200
           17        45840.0  5260.0   648.5  16193.008   7264.790  754.650
           18        39490.0  5780.0    89.5  10150.206  11817.860  880.984
           19        43360.0  4090.0  1047.5  17080.272   4803.320  355.524
           20        10080.0  3800.0   682.0  10382.922   7962.955  304.096
           21        21890.0  3560.0   750.0  11332.728  12391.132  190.060

ID_FINCA                 2083       2360     2486     2821   …   97945  \
          CAMPAÑA                                             …
PRODUCCION 14        10970.0000  27702.000   9060.0  17210.0  …  1155.0
           15        26140.0000  43934.400  11100.0  24020.0  …  1820.0
           16        32270.0000  31010.400   9780.0  21210.0  …  1100.0
           17         7530.0000  21122.450   8210.0  12900.0  …   480.0
           18        49760.0000  33045.240  10560.0  20820.0  …  1145.0
           19        60770.0000  20839.030   6970.0  15600.0  …  2722.5
           20        40250.3728    838.770   8940.0  20480.0  …  1935.0
           21        32440.8096  23654.886  10560.0  10970.0  …  2542.5

ID_FINCA                98118    98140    98265  98571    98814     98995  \
          CAMPAÑA
PRODUCCION 14         3663.3480  27073.2  15990.0   60.0  48486.5  5675.000
           15        23969.8140  41035.4  17280.0  200.0  16100.0  1880.000
           16        18788.1240  36559.0  16080.0  150.0  46780.0  3902.858
           17        17090.6270  18568.5   6860.0  150.0   9420.0  3743.395
           18        22684.2000  33112.0  19270.0  510.0  50760.0  4967.232
           19        11632.1250  18720.5   9660.0  500.0  88860.0  3353.477
           20        12551.1950  31396.0  17160.0  410.0  13630.0  3807.528
           21        18888.7326  56335.0    940.0  250.0  38570.0  3308.716

ID_FINCA              99108     99282      99693
          CAMPAÑA
PRODUCCION 14         4520.0  6630.663  16856.590
           15        11900.0  8000.800  14480.844
           16         7510.0  9230.000  15931.125
           17         5300.0  5840.000  20130.201
           18         5750.0  9070.000  17597.034
           19         3300.0  7380.000  18405.387
           20         6140.0  6710.000  26876.300
           21         4490.0  8460.000  35418.700

[8 rows x 444 columns]
```

[29]:
```
# Guardamos en la variables fincas los identificadores de las fincas.
fincas=df_trend.transpose().index
# Guardamos en x los años correspondientes a las diferentes campañas.
x=list(set(df_train['CAMPAÑA'].values))
x=np.array(x).reshape(-1,1)
```

[30]:
```
from sklearn.linear_model import LinearRegression
```

[31]:
```
model = LinearRegression()
```

[32]:
```
# Llevamos a cabo modelos de regresión lineal para cada finca y mostramos␣
 ↪aquellas cuya producción
# varía tendencialmente, tanto positiva como negativamente, en más de 5000␣
 ↪unidades.
for i in fincas:
    y=df_trend.loc[:, i]
    model.fit(x,y)
    if abs(model.coef_) > 5000:
        print('Finca: ', i, 'Pendiente: ', model.coef_)
# Se observan varias fincas con tendencias de producción notables.
```

```
Finca:  4024 Pendiente:  [-8764.14803571]
Finca:  10376 Pendiente:  [-9838.6872619]
Finca:  14843 Pendiente:  [17172.15954762]
Finca:  45489 Pendiente:  [6408.13508333]
Finca:  48877 Pendiente:  [30525.7275]
Finca:  49541 Pendiente:  [5102.26190476]
Finca:  49636 Pendiente:  [-20222.96666667]
Finca:  52008 Pendiente:  [5558.09685714]
Finca:  58769 Pendiente:  [7228.45238095]
Finca:  68089 Pendiente:  [11520.26640476]
Finca:  72803 Pendiente:  [7259.30019048]
Finca:  80627 Pendiente:  [12265.33883333]
Finca:  83472 Pendiente:  [-5551.66666667]
Finca:  85984 Pendiente:  [-5463.98858333]
Finca:  86582 Pendiente:  [-9482.51785714]
Finca:  95476 Pendiente:  [6099.74264286]
```

[33]:
```
# La finca con mayor pendiente en valor absoluto (20.223) es la 49636, veamos␣
 ↪cómo ha evolucionado
# su producción.
df_trend.loc[:, 49636]
# Se observa como entre 2015 y 2018 la producción de la finca era notablemente␣
 ↪superior a la
# recogida de los años 2019 a 2021. Sin embargo no se observa que la producción␣
 ↪tienda a disminuir
```

```
# 20.000 unidades al año. De hecho observamos como la producción de la campaña␣
 ↪21 supera a la
# producción de la campaña 20. Podríamos decir que se ha producido un cambio de␣
 ↪nivel entre las
# campañas 18 y 19. Por tanto, al menos para esta finca, no parece adecuado␣
 ↪estimar la evolución
# tendencial de la producción en base a la tendencia lineal de la producción␣
 ↪desde 2014. Podría
# ser más efectivo analizar la evolución tendencial de la producción a corto␣
 ↪plazo, teniendo en
# cuenta las 3 o 4 últimas campañas.
```

[33]:
```
                CAMPAÑA
PRODUCCION      14          239445.6
                15          390200.0
                16          343910.0
                17          237430.0
                18          323910.0
                19          215480.0
                20          177270.0
                21          191550.0
Name: 49636, dtype: float64
```

1.0.4 Análisis de superficie y producción por estación y tipo de uva

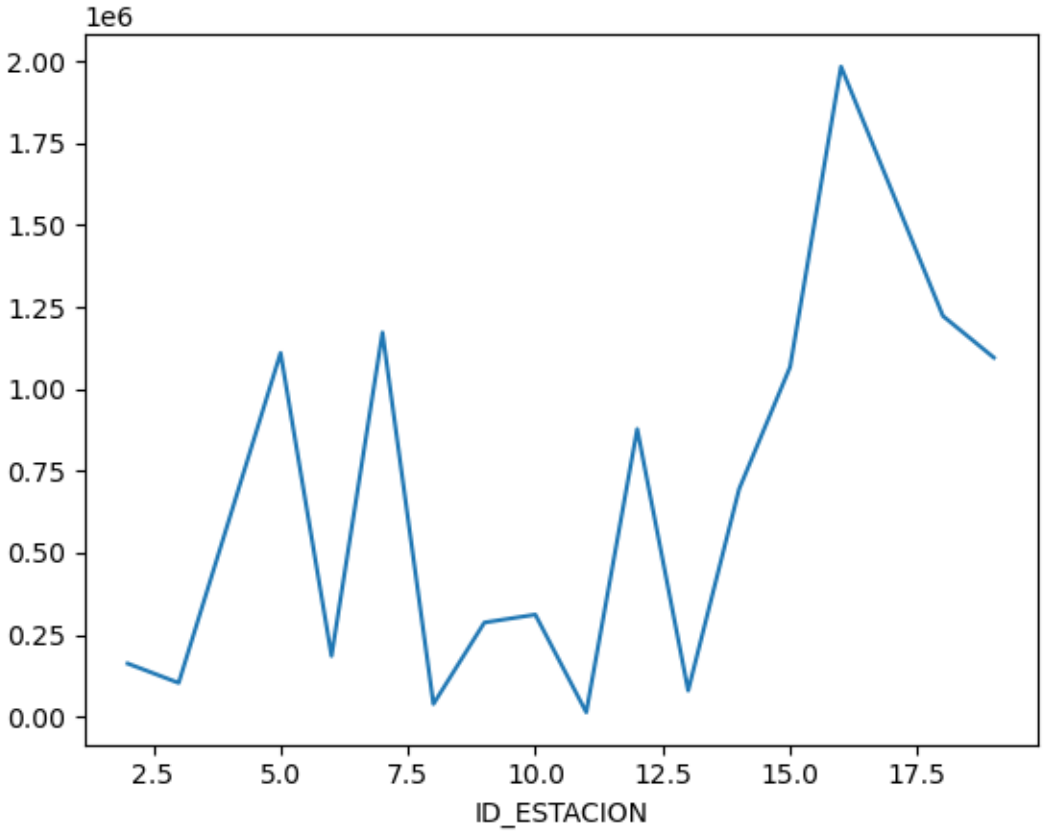
[34]:
```
# Vamos a estudiar la superfecie por finca y por tipo de uva.
# Creamos el dataframe superficies, que contiene las filas con datos sobre␣
 ↪superficie.
superficies=df[df['SUPERFICIE']>0]
# Agrupamos por finca, variedad y campaña y calculamos la superficie y␣
 ↪producción.
superficies=superficies.groupby(['ID_FINCA', 'VARIEDAD', 'CAMPAÑA']).
 ↪agg({'SUPERFICIE': 'sum', 'PRODUCCION': 'sum'})
# Nos quedamos únicamente con las fincas que producen y tienen datos sobre␣
 ↪superficie en la
# campaña 22.
id_fincas_filtered = superficies.loc[(slice(None), slice(None), 22), :].index.
 ↪get_level_values(0).unique()
superficies = superficies.loc[id_fincas_filtered, :]
superficies
# Observamos como solo existen datos disponibles de superficie para las␣
 ↪campañas 20, 21 y 22.
```

[34]:
```
                            SUPERFICIE  PRODUCCION
ID_FINCA VARIEDAD CAMPAÑA
439      9        21           1.0800    1901.402
```

```
                 22          1.0800       0.000
447      40      20          0.4694    2828.540
                 21          0.4694    2037.340
                 22          0.4694       0.000
…                             …          …
99693    81      20          6.3500   26876.300
                 21          6.3500   35418.700
                 22          6.3397       0.000
99793    52      22          0.1326       0.000
         87      22          0.0189       0.000

[2841 rows x 2 columns]
```

[35]:
```
# Estudiamos la producción por estación.
prodest=df[df['CAMPAÑA']==21]
prodest.groupby('ID_ESTACION')['PRODUCCION'].sum().plot()
# Hay distintas estaciones con producción significativa en 2021. Destacan las␣
 ↪estaciones 5, 7,
# 15, 16, 18 y 19.
```

[35]:
```
<AxesSubplot:xlabel='ID_ESTACION'>
```



[36]:
```
# Agrupamos las fincas que abarca cada estación.
estacion = df.groupby('ID_ESTACION')['ID_FINCA'].unique()
# Generamos un dataframe que incluye las filas con superficie positiva␣
 ↪correspondiente a una
# estación dada y calculamos la superficie y producción total agrupando por␣
 ↪finca, variedad y campaña.
# Tomaremos la estación 7 por ser una de las que mayor producción tuvo en 2021.
superficies=df[(df['SUPERFICIE']>0) & (df['ID_FINCA'].isin(estacion[7]))]
superficies=superficies.groupby(['ID_FINCA', 'VARIEDAD', 'CAMPAÑA']).
 ↪agg({'SUPERFICIE': 'sum', 'PRODUCCION': 'sum'})
# Nos quedamos únicamente con los agrupamientos que incluyen datos para la␣
 ↪campaña 22, es decir,
# aquellas fincas y variedad que tienen datos de superficie.
id_fincas_filtered = superficies.loc[(slice(None), slice(None), 22), :].index.
 ↪get_level_values(0).unique()
superficies = superficies.loc[id_fincas_filtered, :]
superficies
```

[36]:

			SUPERFICIE	PRODUCCION
ID_FINCA	VARIEDAD	CAMPAÑA		
439	9	21	1.0800	1901.402
		22	1.0800	0.000
1270	9	20	2.4800	7962.955
		21	2.4800	11788.348
		22	2.4800	0.000
…			…	…
98995	68	21	0.5746	2338.716
		22	0.5746	0.000
99108	52	20	1.6110	6140.000
		21	1.6110	4490.000
		22	1.6110	0.000

[373 rows x 2 columns]

[37]:
```
# Queremos analizar los datos de una estación y una variedad concreta. Ya␣
 ↪habíamos tomado la estación
# 7 y ahora tomaremos la variedad 59.
for finca in set(superficies.index.get_level_values(0)):
    subdf=superficies.loc[(finca,slice(None),slice(None))]
    for variedad in set(subdf.index.get_level_values(0)):
        if variedad == 59: # Se pueden probar 17 32 59, las más cuantiosas
            subbdf=subdf.loc[(variedad,slice(None))]
            if subbdf.index[-1]==22 and len(subbdf.index)>2: # Vamos a tomar␣
 ↪aquellas fincas que
                # tengan datos de superficie en las campañas 20, 21 y 22
```

```
                subbdf['Productividad']=subbdf['PRODUCCION']/
 ↪subbdf['SUPERFICIE'] # Además de la
                # superficie y la producción observamos también la␣
 ↪productividad por unidad de
                #superficie
                print(finca, subbdf, "\n")
# En primer lugar observamos cambios de productividad sospechosos y poco␣
 ↪fiables causados por cambios
# en la superficie. Si nos fijamos en la finca 68.850 su superficie se ha␣
 ↪reducido a la mitad mientras
# que su producción ha crecido un 40%, lo que supone que su productividad se␣
 ↪triplique. En cambio, la
# productividad de las demás fincas no sufre cambios tan bruscos. Por lo tanto␣
 ↪parece que los
# cambios en superficie deben tomarse con cuidado y tal vez no sean␣
 ↪significativos o adecuados para
# predecir, lo comprobaremos a continuación con un modelo de regresión lineal␣
 ↪que mida los cambios de
# producción en función de los cambios de superficie.
# En segundo lugar, al filtrar por estación y variedad el efecto de las␣
 ↪variables meteorológicas
# sobre la productividad debería ser similar para todas las fincas. Vamos a␣
 ↪fijarnos en fincas
# sin cambios de superficie y con producción superior a 5.000 para hacer un␣
 ↪análisis más robusto
# y certero.
# Se observa como algunas fincas (74.473, 80.170, 93.147) sufrieron una␣
 ↪reducción notable en su
# producción en la campaña 21 mientras que otras fincas (32.795, 68.850, 79.
 ↪653) experimentaron
# un fuerte crecimiento. Esto puede deberse a cambios tendenciales en la␣
 ↪producción de cada finca.
# Por tanto, además de analizar la significatividad de los cambios de superficie
# en la predicción de la cantidad producidad, también debemos estudiar si las␣
 ↪tendencias de
# producción y los fenómenos meteorológicos influyen de manera multivariante en␣
 ↪la producción.
```

```
32795           SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20                3.87     26770.0    6917.312661
21                3.84     31890.0    8304.687500
22                3.84         0.0       0.000000

13871           SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20                1.41      3920.0    2780.141844
```

```
21             1.41      3850.0    2730.496454
22             1.41         0.0       0.000000

71788          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           0.2962    2280.912    7700.580689
21           0.2962    2100.840    7092.640108
22           0.2962       0.000       0.000000

51311          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           0.7516    1469.786    1955.542842
21           0.7598     724.660     953.750987
22           0.3799       0.000       0.000000

23668          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20             1.19      9720.0    8168.067227
21             1.19      9360.0    7865.546218
22             1.19         0.0       0.000000

14470          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           0.7824      6990.0    8934.049080
21           0.7824      3710.0    4741.820041
22           0.7824         0.0       0.000000

41607          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           1.6562      2810.0    1696.654993
21           1.6562      3710.0    2240.067625
22           1.6562         0.0       0.000000

38557          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           0.2297       810.0    3526.338703
21           0.2297       830.0    3613.408794
22           0.2297         0.0       0.000000

86229          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           0.1187      916.56    7721.651222
21           0.1187      844.20    7112.047178
22           0.1187        0.00       0.000000

74473          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           1.7953      8180.0    4556.341559
```

```
21           1.7953      5620.0    3130.396034
22           1.7953         0.0       0.000000

93417          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           1.4687     13230.0    9007.966229
21           1.4687     10030.0    6829.168653
22           1.4687         0.0       0.000000

68850          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           4.1452     10770.0    2598.185854
21           2.0800     14730.0    7081.730769
22           2.0800         0.0       0.000000

79653          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           3.1556     11450.0    3628.470022
21           3.1556     15450.0    4896.057802
22           3.1556         0.0       0.000000

80170          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           1.8688      6830.0    3654.751712
21           1.8688      5270.0    2819.991438
22           1.8688         0.0       0.000000

31622          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           0.4247    3622.464    8529.465505
21           0.4247    3336.480    7856.086649
22           0.4247       0.000       0.000000

85407          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           1.1942    4750.214    3977.737397
21           1.1942    1379.020    1154.764696
22           0.5971       0.000       0.000000

57254          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           1.1054      5200.0    4704.179483
21           1.1054      3750.0    3392.437127
22           1.1054         0.0       0.000000

98747          SUPERFICIE  PRODUCCION  Productividad
CAMPAÑA
20           2.1089     10590.0    5021.575229
```

```
21            2.1089      9480.0    4495.234482
22            2.1089         0.0       0.000000
```

[38]:
```
# Vamos también a comprobar si hay diferencias de productividad por modo.
modo=df_train[df_train['CAMPAÑA']==20]
modo=modo[modo['SUPERFICIE']>0]
modo['Productividad 20']=modo['PRODUCCION']/modo['SUPERFICIE']
modo.groupby('MODO').agg({'Productividad 20': 'mean'})
```

[38]:
```
      Productividad 20
MODO
1          3167.210307
2          5668.534080
```

[39]:
```
modo=df_train[df_train['CAMPAÑA']==21]
modo=modo[modo['SUPERFICIE']>0]
modo['Productividad 21']=modo['PRODUCCION']/modo['SUPERFICIE']
modo.groupby('MODO').agg({'Productividad 21': 'mean'})
#Parece que modo 2 es más productivo tanto para año 20 como 21.
#Productividad de modo 1 crece en 20-21 y modo 2 decrece en 20-21.
```

[39]:
```
      Productividad 21
MODO
1          3616.033406
2          5293.063796
```

1.1 Tratamiento Alternativo Superfice

Fincas con un solo registro por año y mismo valor de superficie

[40]:
```
df_filtered = df[df['CAMPAÑA'].isin([20, 21, 22])]
df_grouped = df_filtered.groupby(['ID_FINCA', 'CAMPAÑA']).size().
 ↪reset_index(name='count')

# Filtrar las fincas que tienen un solo registro por campaña 20, 21 y 22, y␣
 ↪mismo valor en SUPERFICIE en esas campañas
valid_id_fincas = df_grouped.groupby('ID_FINCA').filter(lambda x: (x['count'].
 ↪nunique() == 1) and (x['CAMPAÑA'].nunique() == 3) and (x['CAMPAÑA'].
 ↪isin([20, 21, 22])).all() and (x[x['CAMPAÑA'] == 20]['CAMPAÑA'].count() ==␣
 ↪1) and (x[x['CAMPAÑA'] == 21]['CAMPAÑA'].count() == 1) and (x[x['CAMPAÑA']␣
 ↪== 22]['CAMPAÑA'].count() == 1))

# Obtener la lista de ID_FINCA que cumplen con todas las condiciones
valid_id_fincas = valid_id_fincas['ID_FINCA'].unique().tolist()
valid_id_fincas
len(valid_id_fincas)
```

[40]: 588

[41]:
```
df[df["ID_FINCA"]==1142]
```

[41]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	TIPO	\
55	14	1142	464	14	655.0	59	1	0	
1201	15	1142	464	14	655.0	59	1	0	
2317	16	1142	464	14	655.0	59	1	0	
3397	17	1142	464	14	655.0	59	1	0	
4415	18	1142	464	14	655.0	59	1	0	
5470	19	1142	464	14	655.0	59	1	0	
6519	20	1142	464	14	655.0	59	1	0	
7533	21	1142	464	14	655.0	59	1	0	
8603	22	1142	464	14	655.0	59	1	0	

	COLOR	SUPERFICIE	PRODUCCION
55	1	0.00	2106.5
1201	1	0.00	2128.5
2317	1	0.00	1086.0
3397	1	0.00	648.5
4415	1	0.00	89.5
5470	1	0.00	1047.5
6519	1	0.85	682.0
7533	1	0.85	750.0
8603	1	0.85	NaN

[42]:
```
#Para este tipo de fincas tiene sentido imputar por el único valor de superficie registrado
# Crear una copia del DataFrame df para realizar las modificaciones
df_modified = df.copy()

# Recorrer la lista de ID_FINCA válidos
for id_finca in valid_id_fincas:
    # Obtener el valor de SUPERFICIE en la campaña 20 para el ID_FINCA actual
    superficie_20 = df_modified[(df_modified['CAMPAÑA'] == 20) & (df_modified['ID_FINCA'] == id_finca)]['SUPERFICIE'].values[0]

    # Actualizar el valor de SUPERFICIE en las campañas 14, 15, 16, 17, 18, 19 para el ID_FINCA actual
    df_modified.loc[(df_modified['ID_FINCA'] == id_finca) & (df_modified['CAMPAÑA'].isin([14, 15, 16, 17, 18, 19])), 'SUPERFICIE'] = superficie_20

# Verificar los cambios realizados
df_modified[df_modified['ID_FINCA']==1142]
```

[42]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	TIPO	\
55	14	1142	464	14	655.0	59	1	0	
1201	15	1142	464	14	655.0	59	1	0	
2317	16	1142	464	14	655.0	59	1	0	
3397	17	1142	464	14	655.0	59	1	0	
4415	18	1142	464	14	655.0	59	1	0	
5470	19	1142	464	14	655.0	59	1	0	
6519	20	1142	464	14	655.0	59	1	0	
7533	21	1142	464	14	655.0	59	1	0	
8603	22	1142	464	14	655.0	59	1	0	

	COLOR	SUPERFICIE	PRODUCCION
55	1	0.85	2106.5
1201	1	0.85	2128.5
2317	1	0.85	1086.0
3397	1	0.85	648.5
4415	1	0.85	89.5
5470	1	0.85	1047.5
6519	1	0.85	682.0
7533	1	0.85	750.0
8603	1	0.85	NaN

[43]:
```
df=df_modified
```

Fincas con más de un registro en las campañas 20, 21 y 22 que tienen el mismo valor de SUPERFICIE para cada VARIEDAD en las campañas 20 y 21

[44]:
```
# Filtrar las campañas 20, 21 y 22
df_filtered = df[df['CAMPAÑA'].isin([20, 21])]

# Obtener las fincas con más de un registro en las campañas 20, 21 y 22
valid_id_fincas = df_filtered.groupby('ID_FINCA').filter(lambda x: x['CAMPAÑA'].
 ↪nunique() > 1)

# Filtrar las fincas que tienen el mismo valor de SUPERFICIE para cada VARIEDAD␣
 ↪en las campañas 20 y 21
valid_id_fincas = valid_id_fincas.groupby(['ID_FINCA', 'VARIEDAD']).
 ↪filter(lambda x: x['SUPERFICIE'].nunique() == 1 and set(x['CAMPAÑA']) ==␣
 ↪{20, 21})

# Obtener la lista de ID_FINCA que cumplen con todas las condiciones
valid_id_fincas = valid_id_fincas['ID_FINCA'].unique().tolist()
len(valid_id_fincas)
```

[44]: 670

[45]:
```
df[df['ID_FINCA']== 17321]
```

[45]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	TIPO	\
605	14	17321	449	15	625.0	9	1	0	
1759	15	17321	449	15	625.0	9	1	0	
1760	15	17321	449	15	625.0	59	1	0	
2862	16	17321	449	15	625.0	9	1	0	
2863	16	17321	449	15	625.0	59	1	0	
3913	17	17321	449	15	625.0	9	1	0	
3914	17	17321	449	15	625.0	59	1	0	
4985	18	17321	449	15	625.0	9	1	0	
4986	18	17321	449	15	625.0	59	1	0	
6054	19	17321	449	15	625.0	9	1	0	
6055	19	17321	449	15	625.0	59	1	0	
7084	20	17321	449	15	625.0	9	1	0	
7085	20	17321	449	15	625.0	59	1	0	
8106	21	17321	449	15	625.0	9	1	0	
8107	21	17321	449	15	625.0	59	1	0	
9161	22	17321	449	15	625.0	9	1	0	

	COLOR	SUPERFICIE	PRODUCCION
605	0	0.0000	530.0
1759	0	0.0000	2400.0
1760	1	0.0000	898.5
2862	0	0.0000	2460.0
2863	1	0.0000	763.5
3913	0	0.0000	2890.0
3914	1	0.0000	573.0
4985	0	0.0000	4750.0
4986	1	0.0000	1036.5
6054	0	0.0000	2080.0
6055	1	0.0000	532.5
7084	0	0.6409	2390.0
7085	1	0.1620	382.5
8106	0	0.6409	2520.0
8107	1	0.1620	444.0
9161	0	0.6409	NaN

[46]:
```
#Para estas fincas tiene sentido imputar por el valor correspondiente de la␣
 ↪superficie para cada variedad
# Filtrar las filas correspondientes a los ID_FINCA en la lista valid_id_fincas
df_filtered = df[df['ID_FINCA'].isin(valid_id_fincas)]

# Filtrar las campañas 14-19 y 20
df_filtered = df_filtered[df_filtered['CAMPAÑA'].isin(range(14, 20)) |␣
 ↪df_filtered['CAMPAÑA'].eq(20)]

# Crear un diccionario para almacenar los valores de SUPERFICIE del año 20 por␣
 ↪VARIEDAD de UVA
```

```
superficie_dict = df_filtered[df_filtered['CAMPAÑA'].eq(20)].
 ↪groupby(['ID_FINCA', 'VARIEDAD'])['SUPERFICIE'].first().to_dict()

# Imputar los valores de SUPERFICIE de los años 14-19 con los valores del año␣
 ↪20 según la VARIEDAD de UVA
df['SUPERFICIE'] = df.apply(lambda row: superficie_dict.get((row['ID_FINCA'],␣
 ↪row['VARIEDAD']), row['SUPERFICIE']) if row['ID_FINCA'] in valid_id_fincas␣
 ↪and row['CAMPAÑA'] in range(14, 19+1) else row['SUPERFICIE'], axis=1)

df[df['ID_FINCA']==17321 ]
```

[46]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	TIPO	\
605	14	17321	449	15	625.0	9	1	0	
1759	15	17321	449	15	625.0	9	1	0	
1760	15	17321	449	15	625.0	59	1	0	
2862	16	17321	449	15	625.0	9	1	0	
2863	16	17321	449	15	625.0	59	1	0	
3913	17	17321	449	15	625.0	9	1	0	
3914	17	17321	449	15	625.0	59	1	0	
4985	18	17321	449	15	625.0	9	1	0	
4986	18	17321	449	15	625.0	59	1	0	
6054	19	17321	449	15	625.0	9	1	0	
6055	19	17321	449	15	625.0	59	1	0	
7084	20	17321	449	15	625.0	9	1	0	
7085	20	17321	449	15	625.0	59	1	0	
8106	21	17321	449	15	625.0	9	1	0	
8107	21	17321	449	15	625.0	59	1	0	
9161	22	17321	449	15	625.0	9	1	0	

	COLOR	SUPERFICIE	PRODUCCION
605	0	0.6409	530.0
1759	0	0.6409	2400.0
1760	1	0.1620	898.5
2862	0	0.6409	2460.0
2863	1	0.1620	763.5
3913	0	0.6409	2890.0
3914	1	0.1620	573.0
4985	0	0.6409	4750.0
4986	1	0.1620	1036.5
6054	0	0.6409	2080.0
6055	1	0.1620	532.5
7084	0	0.6409	2390.0
7085	1	0.1620	382.5
8106	0	0.6409	2520.0
8107	1	0.1620	444.0
9161	0	0.6409	NaN

[47]:
```
# Reemplazar los valores 0 de SUPERFICIE por NaN
df['SUPERFICIE'] = df['SUPERFICIE'].replace(0, np.nan)
```

[48]:
```
df.isnull().sum()
```

[48]:
```
CAMPAÑA           0
ID_FINCA          0
ID_ZONA           0
ID_ESTACION       0
ALTITUD           0
VARIEDAD          0
MODO              0
TIPO              0
COLOR             0
SUPERFICIE     2046
PRODUCCION     1075
dtype: int64
```

[49]:
```
id_fincas_nulas = df[df['SUPERFICIE'].isnull()]['ID_FINCA'].unique().tolist()
len(id_fincas_nulas)
```

[49]:
```
498
```

Fincas que tengan la(s) misma(s) variedad de uva todos los años

[50]:
```
df[df['ID_FINCA']==16488]
```

[50]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	TIPO	COLOR	SUPERFICIE	PRODUCCION
1434	15	16488	700	16	605.0	32	2	0	1	NaN	3010.0
2540	16	16488	700	16	605.0	32	2	0	1	NaN	11100.0
3616	17	16488	700	16	605.0	32	2	0	1	NaN	7190.0
4651	18	16488	700	16	605.0	32	2	0	1	NaN	13610.0
5709	19	16488	700	16	605.0	32	2	0	1	NaN	16910.0
7777	21	16488	700	16	605.0	32	2	0	1	2.7417	18440.0
8849	22	16488	700	16	605.0	32	2	0	1	2.7417	NaN

[51]:
```
df[df['ID_FINCA']==2916]
```

[51]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	TIPO	\
297	14	2916	677	5	610.0	17	1	0	
298	14	2916	677	5	610.0	32	2	0	
299	14	2916	677	5	610.0	59	2	0	
1442	15	2916	677	5	610.0	32	2	0	
1443	15	2916	677	5	610.0	17	2	0	
1444	15	2916	677	5	610.0	59	2	0	
2551	16	2916	677	5	610.0	32	2	0	
2552	16	2916	677	5	610.0	17	2	0	
2553	16	2916	677	5	610.0	59	2	0	
3624	17	2916	677	5	610.0	17	1	0	
3625	17	2916	677	5	610.0	17	2	0	
3626	17	2916	677	5	610.0	59	2	0	
4662	18	2916	677	5	610.0	17	1	0	
4663	18	2916	677	5	610.0	17	2	0	
4664	18	2916	677	5	610.0	59	2	0	
5718	19	2916	677	5	610.0	17	1	0	
5719	19	2916	677	5	610.0	17	2	0	
5720	19	2916	677	5	610.0	59	2	0	
6784	20	2916	677	5	610.0	17	1	0	
6785	20	2916	677	5	610.0	32	2	0	
6786	20	2916	677	5	610.0	17	2	0	
7788	21	2916	677	5	610.0	17	1	0	
7789	21	2916	677	5	610.0	32	2	0	
7790	21	2916	677	5	610.0	17	2	0	
7791	21	2916	677	5	610.0	59	2	0	
8860	22	2916	677	5	610.0	17	1	0	
8861	22	2916	677	5	610.0	32	2	0	
8862	22	2916	677	5	610.0	59	2	0	

	COLOR	SUPERFICIE	PRODUCCION
297	1	3.800	12558.0
298	1	2.330	11700.0
299	1	NaN	4500.0
1442	1	2.330	22400.0
1443	1	3.800	12529.0
1444	1	NaN	12350.0
2551	1	2.330	7820.0
2552	1	3.800	21794.0
2553	1	NaN	8280.0
3624	1	3.800	3094.0
3625	1	3.800	7684.0
3626	1	NaN	7400.0
4662	1	3.800	12631.0
4663	1	3.800	10463.5
4664	1	NaN	8830.0
5718	1	3.800	8916.5

```
5719      1       3.800      6171.0
5720      1         NaN      5360.0
6784      1       3.800      1921.0
6785      1       2.330      1280.0
6786      1       3.800      7225.0
7788      1       3.800      2490.5
7789      1       2.330      8240.0
7790      1       3.800      6766.0
7791      1       1.260      5300.0
8860      1       3.728         NaN
8861      1       2.260         NaN
8862      1       1.260         NaN
```

[52]:
```
# Crear una lista de ID_FINCA que cumplan la condición de VARIEDAD en los años␣
 ↪14-19 y 20, 21 o 22
id_fincas_coincidentes = []
for id_finca in id_fincas_nulas:
    variedades_14_19 = df[(df['ID_FINCA'] == id_finca) & (df['CAMPAÑA'].
 ↪isin(range(14, 20)))]['VARIEDAD'].unique()
    variedades_20_21_22 = df[(df['ID_FINCA'] == id_finca) & (df['CAMPAÑA'].
 ↪isin([20, 21, 22]))]['VARIEDAD'].unique()
    if set(variedades_14_19) == set(variedades_20_21_22):
        id_fincas_coincidentes.append(id_finca)
```

[53]:
```
#Veo que no hay diferencias significativas entre las superficies de cada fincas␣
 ↪entre los distintos años para cada variedad
```

[54]:
```
def dif_max(df, id_fincas_coincidentes):
    max_diferencias_fincas = []

    for id_finca in id_fincas_coincidentes:
        df_selected = df[df['ID_FINCA'] == id_finca]
        variedades = df_selected['VARIEDAD'].unique()
        for variedad in variedades:
            superficies = df_selected[(df_selected['VARIEDAD'] == variedad) &␣
 ↪df_selected['CAMPAÑA'].isin([20, 21, 22])]['SUPERFICIE'].dropna()
            if len(superficies) >= 2:
                diferencia = superficies.max() - superficies.min()
                max_diferencias_fincas.append([diferencia, id_finca])

    return max_diferencias_fincas
dif_max(df, id_fincas_coincidentes)
```

[54]:
```
[[0.0, 70378],
 [0.0, 92531],
 [0.0, 92099],
 [0.0, 89768],
```

```
[0.0, 31153],
[0.0, 74003],
[0.13730000000000064, 19605],
[0.07199999999999962, 2916],
[0.07000000000000028, 2916],
[0.0, 2916],
[0.0, 93538],
[0.0, 93538],
[0.0, 76588],
[0.0, 76588],
[0.0, 28362],
[0.19999999999999996, 50441],
[0.0, 70407],
[0.0, 70407],
[0.0, 52182],
[0.0, 70112],
[0.0, 33084],
[0.0, 47298],
[0.0, 21119],
[0.0, 81717],
[0.0, 47874],
[0.0, 47874],
[0.0040999999999999925, 51311],
[0.0, 63510],
[0.0, 917],
[0.0, 26730],
[0.0, 58948],
[0.0, 76361],
[0.0, 48720],
[0.0, 73540],
[0.00990000000000002, 5548],
[0.0, 72387],
[0.0, 21481],
[0.0, 98871],
[0.0, 32986],
[0.0, 74812],
[0.0, 96004],
[0.0, 51677],
[0.0, 47053],
[0.0, 22092],
[0.6080000000000001, 26979],
[0.0, 62869],
[0.2699999999999996, 34244],
[0.2149000000000001, 78863],
[0.3386, 12622],
[0.0, 16488],
[0.0, 77844],
```

```
 [0.0, 93970],
 [0.0, 84689],
 [0.0, 84689],
 [0.00039999999999995595, 66451],
 [0.0, 90597],
 [0.028600000000000014, 64738],
 [0.0, 55618],
 [0.0, 33486],
 [0.0, 33486],
 [0.005599999999999994, 90076],
 [0.04720000000000013, 63312],
 [0.06330000000000013, 4990],
 [0.0, 25627],
 [0.25459999999999994, 98646],
 [0.0, 43806]]
```

[55]:
```
#Como no hay diferencias grandes de superficie, tendremos que para cada␣
 ↪variedad, la superficie será similar a lo largo de estos
#años. Imputamos por la media de superficie en cada variedad.
def impute(df, id_fincas_coincidentes):
    for id_finca in id_fincas_coincidentes:
        df_selected = df[df['ID_FINCA'] == id_finca]
        variedades = df_selected['VARIEDAD'].unique()
        for variedad in variedades:
            media_superficie = df_selected[df_selected['VARIEDAD'] ==␣
 ↪variedad]['SUPERFICIE'].mean()
            df.loc[(df['ID_FINCA'] == id_finca) & (df['VARIEDAD'] == variedad)␣
 ↪& (df['SUPERFICIE'].isnull()), 'SUPERFICIE'] = media_superficie

    return df
df=impute(df, id_fincas_coincidentes)
```

[56]:
```
df[df['ID_FINCA']==16488]
```

[56]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	TIPO	\
1434	15	16488	700	16	605.0	32	2	0	
2540	16	16488	700	16	605.0	32	2	0	
3616	17	16488	700	16	605.0	32	2	0	
4651	18	16488	700	16	605.0	32	2	0	
5709	19	16488	700	16	605.0	32	2	0	
7777	21	16488	700	16	605.0	32	2	0	
8849	22	16488	700	16	605.0	32	2	0	

	COLOR	SUPERFICIE	PRODUCCION
1434	1	2.7417	3010.0
2540	1	2.7417	11100.0
3616	1	2.7417	7190.0

```
4651       1       2.7417     13610.0
5709       1       2.7417     16910.0
7777       1       2.7417     18440.0
8849       1       2.7417         NaN
```

[57]:
```
df[df['ID_FINCA']==2916]
```

[57]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	TIPO	\
297	14	2916	677	5	610.0	17	1	0	
298	14	2916	677	5	610.0	32	2	0	
299	14	2916	677	5	610.0	59	2	0	
1442	15	2916	677	5	610.0	32	2	0	
1443	15	2916	677	5	610.0	17	2	0	
1444	15	2916	677	5	610.0	59	2	0	
2551	16	2916	677	5	610.0	32	2	0	
2552	16	2916	677	5	610.0	17	2	0	
2553	16	2916	677	5	610.0	59	2	0	
3624	17	2916	677	5	610.0	17	1	0	
3625	17	2916	677	5	610.0	17	2	0	
3626	17	2916	677	5	610.0	59	2	0	
4662	18	2916	677	5	610.0	17	1	0	
4663	18	2916	677	5	610.0	17	2	0	
4664	18	2916	677	5	610.0	59	2	0	
5718	19	2916	677	5	610.0	17	1	0	
5719	19	2916	677	5	610.0	17	2	0	
5720	19	2916	677	5	610.0	59	2	0	
6784	20	2916	677	5	610.0	17	1	0	
6785	20	2916	677	5	610.0	32	2	0	
6786	20	2916	677	5	610.0	17	2	0	
7788	21	2916	677	5	610.0	17	1	0	
7789	21	2916	677	5	610.0	32	2	0	
7790	21	2916	677	5	610.0	17	2	0	
7791	21	2916	677	5	610.0	59	2	0	
8860	22	2916	677	5	610.0	17	1	0	
8861	22	2916	677	5	610.0	32	2	0	
8862	22	2916	677	5	610.0	59	2	0	

	COLOR	SUPERFICIE	PRODUCCION
297	1	3.800	12558.0
298	1	2.330	11700.0
299	1	1.260	4500.0
1442	1	2.330	22400.0
1443	1	3.800	12529.0
1444	1	1.260	12350.0
2551	1	2.330	7820.0
2552	1	3.800	21794.0
2553	1	1.260	8280.0

```
3624      1       3.800      3094.0
3625      1       3.800      7684.0
3626      1       1.260      7400.0
4662      1       3.800     12631.0
4663      1       3.800     10463.5
4664      1       1.260      8830.0
5718      1       3.800      8916.5
5719      1       3.800      6171.0
5720      1       1.260      5360.0
6784      1       3.800      1921.0
6785      1       2.330      1280.0
6786      1       3.800      7225.0
7788      1       3.800      2490.5
7789      1       2.330      8240.0
7790      1       3.800      6766.0
7791      1       1.260      5300.0
8860      1       3.728         NaN
8861      1       2.260         NaN
8862      1       1.260         NaN
```

[58]:
```
df.isnull().sum()
```

[58]:
```
CAMPAÑA           0
ID_FINCA          0
ID_ZONA           0
ID_ESTACION       0
ALTITUD           0
VARIEDAD          0
MODO              0
TIPO              0
COLOR             0
SUPERFICIE     1409
PRODUCCION     1075
dtype: int64
```

[59]:
```
#Para el resto de superficies, necesitamos un modelo de predicción
```

[60]:
```
# Eliminar los registros donde CAMPAÑA=22 (porque quiero usar PRODUCCION como␣
 ↪variable independiente) y crear df1421
df1421 = df[df['CAMPAÑA'] != 22]
```

[61]:
```
# Conjunto de entrenamiento
df_entrenamiento = df1421[df1421['SUPERFICIE'].notnull()]
```

[62]:
```
# Conjunto de prueba
df_prueba = df1421[df1421['SUPERFICIE'].isnull()]
```

[63]:
```
from sklearn.model_selection import train_test_split

# Variables independientes (todas excepto SUPERFICIE)
X = df_entrenamiento.drop('SUPERFICIE', axis=1).values
X_nulo=df_prueba.drop('SUPERFICIE', axis=1).values

# Variable dependiente (SUPERFICIE)
y = df_entrenamiento['SUPERFICIE'].values.reshape(-1, 1)

# División de los datos en conjunto de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[64]:
```
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.metrics import r2_score
```

[65]:
```
from sklearn.ensemble import GradientBoostingRegressor
```

[66]:
```
# Creamos el modelo de Gradient Boosting
gbr = GradientBoostingRegressor(n_estimators=100, learning_rate=0.15,␣
 ↪max_depth=6, random_state=42)

# Ajustamos el modelo con los datos de entrenamiento
gbr.fit(X_train, y_train)

y_pred_test=gbr.predict(X_test)
```

[67]:
```
# Vemos como ha sido el entrenamiento
y_pred_train=gbr.predict(X_train)
rmse_train = mean_squared_error(y_train, y_pred_train, squared=False)
mae_train = mean_absolute_error(y_train, y_pred_train)
r2 = r2_score(y_train, y_pred_train)
print(f'RMSE en train: {rmse_train:.5f}')
print(f'MAE en train: {mae_train:.5f}')
print("R2 score:", r2)
```

```
RMSE en train: 0.57973
MAE en train: 0.38433
R2 score: 0.9548997113545848
```

[68]:
```
# En test
rmse = mean_squared_error(y_test, y_pred_test, squared=False)
print('RMSE:', rmse)
mae=mean_absolute_error(y_test, y_pred_test)
print('MAE:', mae)
r2 = r2_score(y_test, y_pred_test)
print("R2 score:", r2)
```

```
RMSE: 1.0780501087357484
MAE: 0.604035067669054
R2 score: 0.8360060188195588
```

[69]:
```
gbr.fit(X, y)
```

[69]:
```
GradientBoostingRegressor(learning_rate=0.15, max_depth=6, random_state=42)
```

[70]:
```
y_pred= gbr.predict(X_nulo)
```

[71]:
```
# Obtener los índices de los registros con valores nulos en SUPERFICIE y␣
↪CAMPAÑA distinta de 22
indices_nulos = (df['SUPERFICIE'].isnull()) & (df['CAMPAÑA'] != 22)

# Imputar los valores nulos con y_pred
df.loc[indices_nulos, 'SUPERFICIE'] = np.squeeze(y_pred)
```

[72]:
```
df.isnull().sum()
```

[72]:
```
CAMPAÑA           0
ID_FINCA          0
ID_ZONA           0
ID_ESTACION       0
ALTITUD           0
VARIEDAD          0
MODO              0
TIPO              0
COLOR             0
SUPERFICIE        9
PRODUCCION     1075
dtype: int64
```

[73]:
```
# Crear df22 con los registros donde CAMPAÑA=22
df22 = df[df['CAMPAÑA'] == 22]
df22.isnull().sum()
```

[73]:
```
CAMPAÑA           0
ID_FINCA          0
ID_ZONA           0
ID_ESTACION       0
ALTITUD           0
VARIEDAD          0
MODO              0
TIPO              0
COLOR             0
SUPERFICIE        9
PRODUCCION     1075
dtype: int64
```

[74]:
```
#Los valores nulos de superficie corresponden a 2022. Las fincas son:
id_fincas_nulas = df[df['SUPERFICIE'].isnull()]['ID_FINCA'].unique().tolist()
id_fincas_nulas
```

[74]:
```
[4024, 19067, 86582, 77984, 13054, 16692, 78846, 59317]
```

[75]:
```
df[df['ID_FINCA']==4024]
```

[75]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	TIPO	\
15	14	4024	919	14	655.0	32	2	0	
16	14	4024	919	14	655.0	87	2	0	
17	14	4024	919	14	655.0	17	2	0	
18	14	4024	919	14	655.0	52	2	0	
19	14	4024	919	14	655.0	59	2	0	
20	14	4024	919	14	655.0	81	2	0	
1161	15	4024	919	14	655.0	32	2	0	
1162	15	4024	919	14	655.0	87	2	0	
1163	15	4024	919	14	655.0	40	2	0	
1164	15	4024	919	14	655.0	17	2	0	
1165	15	4024	919	14	655.0	52	2	0	
1166	15	4024	919	14	655.0	59	2	0	
1167	15	4024	919	14	655.0	81	2	0	
2272	16	4024	919	14	655.0	32	2	0	
2273	16	4024	919	14	655.0	87	2	0	
2274	16	4024	919	14	655.0	40	2	0	
2275	16	4024	919	14	655.0	17	2	0	
2276	16	4024	919	14	655.0	52	2	0	
2277	16	4024	919	14	655.0	59	2	0	
2278	16	4024	919	14	655.0	81	2	0	
3349	17	4024	919	14	655.0	32	2	0	
3350	17	4024	919	14	655.0	87	2	0	
3351	17	4024	919	14	655.0	40	2	0	
3352	17	4024	919	14	655.0	17	2	0	
3353	17	4024	919	14	655.0	52	2	0	
3354	17	4024	919	14	655.0	59	2	0	
3355	17	4024	919	14	655.0	81	2	0	
4367	18	4024	919	14	655.0	32	2	0	
4368	18	4024	919	14	655.0	87	2	0	
4369	18	4024	919	14	655.0	40	2	0	
4370	18	4024	919	14	655.0	17	2	0	
4371	18	4024	919	14	655.0	52	2	0	
4372	18	4024	919	14	655.0	59	2	0	
4373	18	4024	919	14	655.0	81	2	0	
5427	19	4024	919	14	655.0	32	2	0	
5428	19	4024	919	14	655.0	87	2	0	
5429	19	4024	919	14	655.0	40	2	0	
5430	19	4024	919	14	655.0	52	2	0	

5431	19	4024	919	14	655.0	59	2	0
5432	19	4024	919	14	655.0	81	2	0
6482	20	4024	919	14	655.0	32	2	0
6483	20	4024	919	14	655.0	87	2	0
6484	20	4024	919	14	655.0	40	2	0
6485	20	4024	919	14	655.0	52	2	0
6486	20	4024	919	14	655.0	59	2	0
6487	20	4024	919	14	655.0	81	2	0
7489	21	4024	919	14	655.0	32	2	0
7490	21	4024	919	14	655.0	87	2	0
7491	21	4024	919	14	655.0	40	2	0
7492	21	4024	919	14	655.0	52	2	0
7493	21	4024	919	14	655.0	59	2	0
7494	21	4024	919	14	655.0	81	2	0
8531	22	4024	919	14	655.0	15	2	0
8532	22	4024	919	14	655.0	32	2	0
8533	22	4024	919	14	655.0	87	2	0
8534	22	4024	919	14	655.0	40	2	0
8535	22	4024	919	14	655.0	52	2	0
8536	22	4024	919	14	655.0	59	2	0
8537	22	4024	919	14	655.0	81	2	0

	COLOR	SUPERFICIE	PRODUCCION
15	1	11.320000	54893.9000
16	0	5.705300	13440.0000
17	1	9.517653	33878.0000
18	1	9.267700	34170.0000
19	1	7.066931	13731.2000
20	1	13.860000	42580.0000
1161	1	11.320000	85731.5310
1162	0	5.705300	49460.0000
1163	1	7.535000	15932.1730
1164	1	9.208107	28770.3156
1165	1	9.267700	48260.0000
1166	1	8.807946	22631.0280
1167	1	13.860000	49183.2260
2272	1	11.320000	48709.3130
2273	0	5.705300	40080.0000
2274	1	7.535000	9575.5660
2275	1	9.409458	35418.5520
2276	1	9.267700	40750.0000
2277	1	7.506912	12380.1930
2278	1	13.860000	60125.1300
3349	1	11.320000	54111.6840
3350	0	5.705300	31710.0000
3351	1	7.535000	10742.6180
3352	1	8.025292	19339.5120

```
3353       1    9.267700  45590.0000
3354       1    7.071915  12785.2330
3355       1   13.860000  23480.8200
4367       1   11.320000  46441.5180
4368       0    5.705300  46380.0000
4369       1    7.535000   8365.4670
4370       1    7.789503  18669.5520
4371       1    9.267700  30180.0000
4372       1    7.487950   8411.8440
4373       1   13.860000  58654.6140
5427       1   11.320000  67691.7900
5428       0    5.705300  67250.0000
5429       1    7.535000   5289.9980
5430       1    9.267700  28940.0000
5431       1    7.221771  15961.3860
5432       1   13.860000  50781.0280
6482       1   11.320000  64104.2230
6483       0    5.705300  45400.0000
6484       1    7.535000  11352.1680
6485       1    9.267700  28960.0000
6486       1    4.990574   4346.7060
6487       1   13.860000  27380.4600
7489       1   11.320000  48718.2080
7490       0    5.705300  57400.0000
7491       1    7.535000  17533.4790
7492       1    9.267700  20710.0000
7493       1    7.677752   8045.2190
7494       1   13.860000  21709.8620
8531       1    0.662200         NaN
8532       1   21.110000         NaN
8533       0   11.923800         NaN
8534       1   18.174600         NaN
8535       1    8.072600         NaN
8536       1         NaN         NaN
8537       1   13.860000         NaN
```

[76]:
```
#Imputamos por la media de superficie en cada variedad
df=impute(df, id_fincas_nulas)
```

[77]:
```
df[df['ID_FINCA']==4024]
```

[77]:
```
      CAMPAÑA  ID_FINCA  ID_ZONA  ID_ESTACION  ALTITUD  VARIEDAD  MODO  TIPO  \
15         14      4024      919           14    655.0        32     2     0
16         14      4024      919           14    655.0        87     2     0
17         14      4024      919           14    655.0        17     2     0
18         14      4024      919           14    655.0        52     2     0
19         14      4024      919           14    655.0        59     2     0
```

20	14	4024	919	14	655.0	81	2	0
1161	15	4024	919	14	655.0	32	2	0
1162	15	4024	919	14	655.0	87	2	0
1163	15	4024	919	14	655.0	40	2	0
1164	15	4024	919	14	655.0	17	2	0
1165	15	4024	919	14	655.0	52	2	0
1166	15	4024	919	14	655.0	59	2	0
1167	15	4024	919	14	655.0	81	2	0
2272	16	4024	919	14	655.0	32	2	0
2273	16	4024	919	14	655.0	87	2	0
2274	16	4024	919	14	655.0	40	2	0
2275	16	4024	919	14	655.0	17	2	0
2276	16	4024	919	14	655.0	52	2	0
2277	16	4024	919	14	655.0	59	2	0
2278	16	4024	919	14	655.0	81	2	0
3349	17	4024	919	14	655.0	32	2	0
3350	17	4024	919	14	655.0	87	2	0
3351	17	4024	919	14	655.0	40	2	0
3352	17	4024	919	14	655.0	17	2	0
3353	17	4024	919	14	655.0	52	2	0
3354	17	4024	919	14	655.0	59	2	0
3355	17	4024	919	14	655.0	81	2	0
4367	18	4024	919	14	655.0	32	2	0
4368	18	4024	919	14	655.0	87	2	0
4369	18	4024	919	14	655.0	40	2	0
4370	18	4024	919	14	655.0	17	2	0
4371	18	4024	919	14	655.0	52	2	0
4372	18	4024	919	14	655.0	59	2	0
4373	18	4024	919	14	655.0	81	2	0
5427	19	4024	919	14	655.0	32	2	0
5428	19	4024	919	14	655.0	87	2	0
5429	19	4024	919	14	655.0	40	2	0
5430	19	4024	919	14	655.0	52	2	0
5431	19	4024	919	14	655.0	59	2	0
5432	19	4024	919	14	655.0	81	2	0
6482	20	4024	919	14	655.0	32	2	0
6483	20	4024	919	14	655.0	87	2	0
6484	20	4024	919	14	655.0	40	2	0
6485	20	4024	919	14	655.0	52	2	0
6486	20	4024	919	14	655.0	59	2	0
6487	20	4024	919	14	655.0	81	2	0
7489	21	4024	919	14	655.0	32	2	0
7490	21	4024	919	14	655.0	87	2	0
7491	21	4024	919	14	655.0	40	2	0
7492	21	4024	919	14	655.0	52	2	0
7493	21	4024	919	14	655.0	59	2	0
7494	21	4024	919	14	655.0	81	2	0

8531	22	4024	919	14	655.0	15	2	0
8532	22	4024	919	14	655.0	32	2	0
8533	22	4024	919	14	655.0	87	2	0
8534	22	4024	919	14	655.0	40	2	0
8535	22	4024	919	14	655.0	52	2	0
8536	22	4024	919	14	655.0	59	2	0
8537	22	4024	919	14	655.0	81	2	0

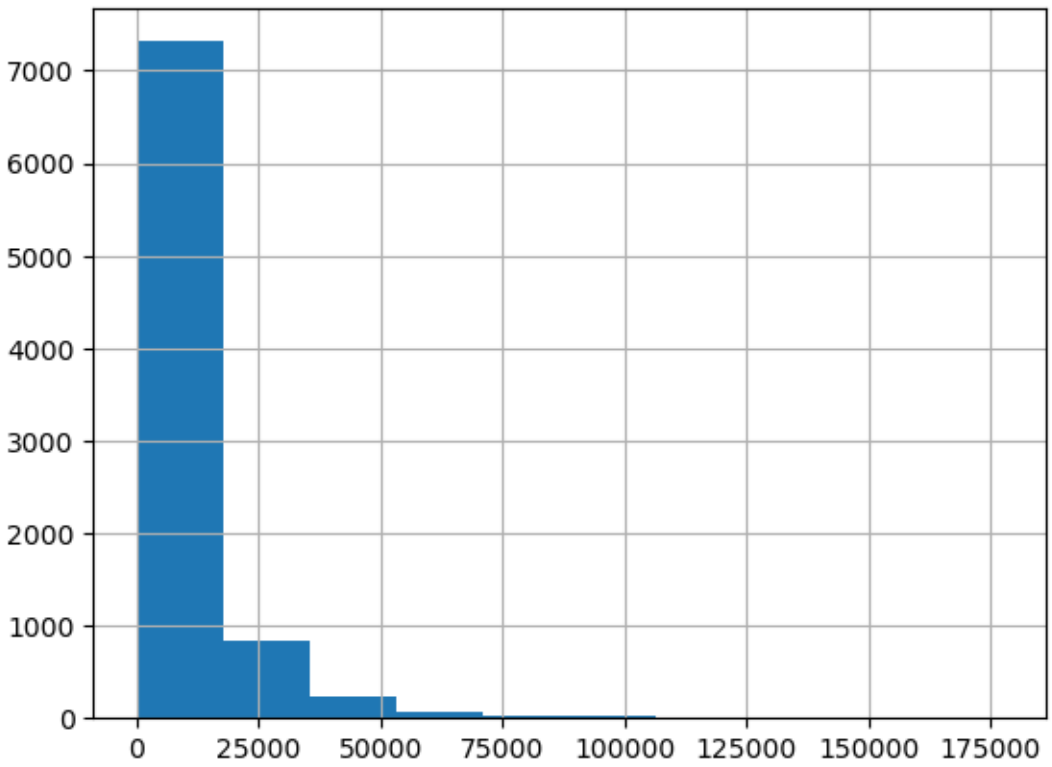
	COLOR	SUPERFICIE	PRODUCCION
15	1	11.320000	54893.9000
16	0	5.705300	13440.0000
17	1	9.517653	33878.0000
18	1	9.267700	34170.0000
19	1	7.066931	13731.2000
20	1	13.860000	42580.0000
1161	1	11.320000	85731.5310
1162	0	5.705300	49460.0000
1163	1	7.535000	15932.1730
1164	1	9.208107	28770.3156
1165	1	9.267700	48260.0000
1166	1	8.807946	22631.0280
1167	1	13.860000	49183.2260
2272	1	11.320000	48709.3130
2273	0	5.705300	40080.0000
2274	1	7.535000	9575.5660
2275	1	9.409458	35418.5520
2276	1	9.267700	40750.0000
2277	1	7.506912	12380.1930
2278	1	13.860000	60125.1300
3349	1	11.320000	54111.6840
3350	0	5.705300	31710.0000
3351	1	7.535000	10742.6180
3352	1	8.025292	19339.5120
3353	1	9.267700	45590.0000
3354	1	7.071915	12785.2330
3355	1	13.860000	23480.8200
4367	1	11.320000	46441.5180
4368	0	5.705300	46380.0000
4369	1	7.535000	8365.4670
4370	1	7.789503	18669.5520
4371	1	9.267700	30180.0000
4372	1	7.487950	8411.8440
4373	1	13.860000	58654.6140
5427	1	11.320000	67691.7900
5428	0	5.705300	67250.0000
5429	1	7.535000	5289.9980
5430	1	9.267700	28940.0000

5431	1	7.221771	15961.3860
5432	1	13.860000	50781.0280
6482	1	11.320000	64104.2230
6483	0	5.705300	45400.0000
6484	1	7.535000	11352.1680
6485	1	9.267700	28960.0000
6486	1	4.990574	4346.7060
6487	1	13.860000	27380.4600
7489	1	11.320000	48718.2080
7490	0	5.705300	57400.0000
7491	1	7.535000	17533.4790
7492	1	9.267700	20710.0000
7493	1	7.677752	8045.2190
7494	1	13.860000	21709.8620
8531	1	0.662200	NaN
8532	1	21.110000	NaN
8533	0	11.923800	NaN
8534	1	18.174600	NaN
8535	1	8.072600	NaN
8536	1	7.228969	NaN
8537	1	13.860000	NaN

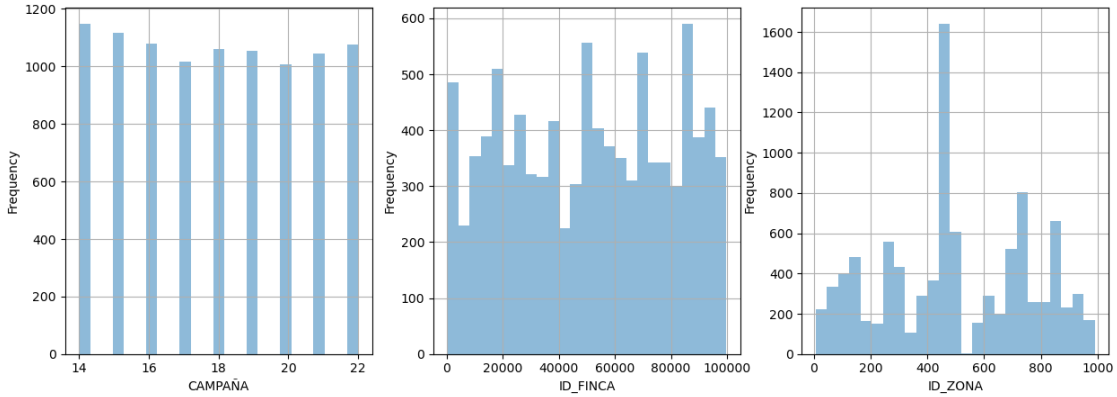
1.1.1 DISTRIBUCIONES DE LAS VARIABLES

[78]:
```
# PRODUCCION
df["PRODUCCION"].hist()
```

[78]: <AxesSubplot:>



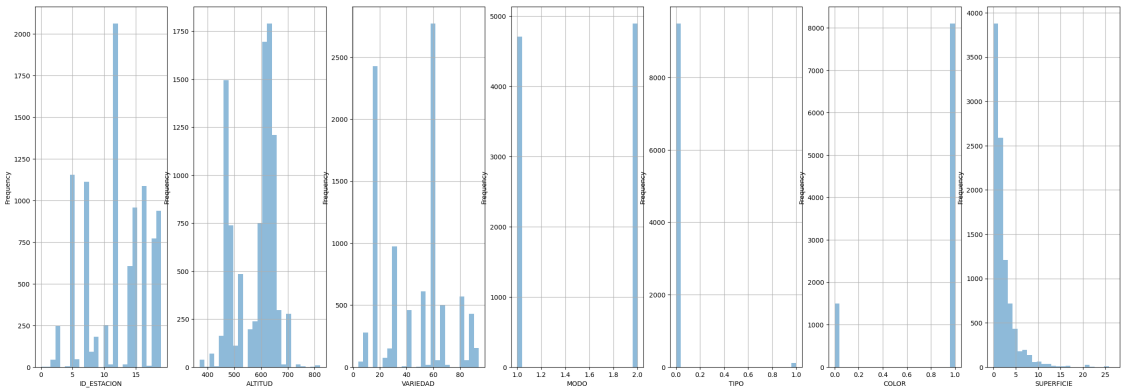
[79]:
```
plt.figure(figsize=(15, 5))

plt.subplot(1,3,1)
df['CAMPAÑA'].plot.hist(alpha=0.5, bins=25, grid = True)
plt.xlabel('CAMPAÑA')

plt.subplot(1,3,2)
df['ID_FINCA'].plot.hist(alpha=0.5, bins=25, grid = True)
plt.xlabel('ID_FINCA')

plt.subplot(1,3,3)
df['ID_ZONA'].plot.hist(alpha=0.5, bins=25, grid = True)
plt.xlabel('ID_ZONA')

plt.show()
```



Observaciones: - Contamos con una cantidad similar de datos para cada campaña, lo cual resulta beneficioso debido a que evita inclinar el modelo hacia los acontecimientos de una campaña específica. - Observamos que hay ciertas fincas que tienen un número considerablemente menor de registros en comparación con otras. Es crucial considerar esta disparidad y asegurarnos de que el modelo no se limite únicamente a predecir con precisión los registros pertenecientes a las fincas más numerosas. - Al igual que ocurre con el campo ID_FINCA, hay zonas que están considerablemente más presentes en los datos en comparación con otras. Por lo tanto, es necesario estar atentos para evitar que el modelo priorice acertar en las zonas más comunes en lugar de tener una visión equilibrada y precisa en todas las zonas.

[80]:
```
plt.figure(figsize=(30, 10))

plt.subplot(1,7,1)
df['ID_ESTACION'].plot.hist(alpha=0.5, bins=25, grid = True)
plt.xlabel('ID_ESTACION')

plt.subplot(1,7,2)
df['ALTITUD'].plot.hist(alpha=0.5, bins=25, grid = True)
plt.xlabel('ALTITUD')

plt.subplot(1,7,3)
df['VARIEDAD'].plot.hist(alpha=0.5, bins=25, grid = True)
plt.xlabel('VARIEDAD')

plt.subplot(1,7,4)
df['MODO'].plot.hist(alpha=0.5, bins=25, grid = True)
plt.xlabel('MODO')

plt.subplot(1,7,5)
df['TIPO'].plot.hist(alpha=0.5, bins=25, grid = True)
plt.xlabel('TIPO')

plt.subplot(1,7,6)
```

```
df['COLOR'].plot.hist(alpha=0.5, bins=25, grid = True)
plt.xlabel('COLOR')

plt.subplot(1,7,7)
df['SUPERFICIE'].plot.hist(alpha=0.5, bins=25, grid = True)
plt.xlabel('SUPERFICIE')

plt.show()
```



Observaciones: - Hay muchos más registros en determinadas estaciones. Esto puede motivar la idea de codificar la variable ID_ESTACION y así contrarrestras este sesgo. - Los registros para el modo de cultivo están equilibrados. - Tenemos que tener en cuenta que hay variedades de uva, tipos de cultivo (tipo 1) y colores (color 0) minoritarios.

1.1.2 Análisis multivariante de la producción mediante modelos de regresión

[81]:
```
# Vamos a comenzar a obtener parámetros mediante modelos de regresión lineal␣
 ↪para estimar
# el crecimiento tendencial de la producción para cada finca.
# Creamos el dataframe df_betas, el cual emplearemos para crear una variable␣
 ↪'Betas' que indique
# la tendencia en la producción absoluta de cada finca, variedad y modo.
df_betas=df.groupby(['ID_FINCA', 'VARIEDAD', 'MODO', 'CAMPAÑA']).
 ↪agg({'PRODUCCION':'sum'})
df_betas=df_betas.unstack().transpose()
df_betas
```

[81]:
```
ID_FINCA                200       439                447                    \
VARIEDAD                 59         9        52        17                40
MODO                      1         2         2         1         2         2
           CAMPAÑA
PRODUCCION 14       1900.000       NaN  2215.2  1824.700       NaN       NaN
           15        778.104       NaN  3208.4       NaN  3242.106       NaN
```

```
           16       1636.200       NaN  6354.4   864.108  1660.176      NaN
           17        829.008       NaN     NaN       NaN  1336.986      NaN
           18        607.212       NaN     NaN       NaN       NaN      NaN
           19        392.688       NaN     NaN       NaN       NaN      NaN
           20        545.400       NaN     NaN       NaN       NaN  2828.54
           21            NaN  1901.402     NaN       NaN       NaN  2037.34
           22            NaN     0.000     NaN       NaN       NaN     0.00

ID_FINCA              523             528   702    …   99033    99108  \
VARIEDAD                32      59       59    59  …      81       52
MODO                     2       1        1     2  …       2        2
           CAMPAÑA                                  …
PRODUCCION 14            NaN  2290.4  22780.0   NaN  …  2284.2   4520.0
           15            NaN     NaN      NaN   NaN  …     NaN  11900.0
           16            NaN     NaN      NaN   NaN  …     NaN   7510.0
           17       3732.000     NaN      NaN   NaN  …     NaN   5300.0
           18       2836.074     NaN      NaN   NaN  …     NaN   5750.0
           19       1225.824     NaN      NaN   NaN  …     NaN   3300.0
           20        947.844     NaN      NaN   NaN  …     NaN   6140.0
           21        745.122     NaN      NaN   NaN  …     NaN   4490.0
           22          0.000     NaN      NaN   0.0  …     NaN      0.0

ID_FINCA              99146             99282  99377             99693 99793  \
VARIEDAD                17                59     52                81    52
MODO                     1       2         2      1       2         1     2
           CAMPAÑA
PRODUCCION 14          NaN     NaN  6630.663    NaN     NaN  16856.590   NaN
           15       6480.0     NaN  8000.800    NaN  2280.0  14480.844   NaN
           16       4080.0     NaN  9230.000  560.0   990.0  15931.125   NaN
           17       6060.0     NaN  5840.000    NaN     NaN  20130.201   NaN
           18          NaN  3700.0  9070.000    NaN  2160.0  17597.034   NaN
           19          NaN  3380.0  7380.000    NaN  1840.0  18405.387   NaN
           20          NaN  3300.0  6710.000    NaN  2300.0  26876.300   NaN
           21          NaN  4730.0  8460.000    NaN  2460.0  35418.700   NaN
           22          NaN     0.0     0.000    NaN     0.0      0.000   0.0

ID_FINCA
VARIEDAD             87
MODO                  2
           CAMPAÑA
PRODUCCION 14       NaN
           15       NaN
           16       NaN
           17       NaN
           18       NaN
           19       NaN
           20       NaN
```

```
21       NaN
22       0.0

[9 rows x 1946 columns]
```

[82]:
```
# Definimos la siguiente función que calcula la tendencia a corto plazo de la cantidad que
# aumenta/decrece la producción de una finca y variedad para una campaña dada. Se toman solo
# tendencias claramente definidas, dadas por una regresión lineal con un R^2 superior a 0.95.
# Para dicha regresión se toman las cuatro campañas anteriores a la indicada en la función.
def beta(finca, variedad, modo, campaña, df=df_betas): # la función utiliza de manera predefinida
    # el dataframe df_betas
    lr = LinearRegression()
    if df[(finca, variedad, modo)].iloc[(campaña-18):(campaña-14)].isnull().values.any(): # si hay
        # valores missing no hay producción en alguno de los años y por tanto no hay tendencia
        beta = 0
    else:
        x = np.arange(4).reshape(-1,1)
        y = df[(finca, variedad, modo)].iloc[(campaña-18):(campaña-14)].values
        lr.fit(x,y)
        if lr.score(x,y)>0.95: # Si R^2 menor que cierta cota, consideramos que no hay tendencia
            beta = lr.coef_
        else:
            beta = 0
    return beta
```

[83]:
```
# Añadimos los valores de las betas en una nueva columna del dataframe df.
df['Betas']=0
for fila in range(len(df)): # Se coge longitud df_train para no tocar las filas de la campaña 22
    if df.loc[fila, 'CAMPAÑA'] >= 18:  # Para 2018 hacia atrás no calculamos tendencias por no
        # haber datos suficientes
        df.loc[fila, 'Betas']=beta(df.loc[fila, 'ID_FINCA'], df.loc[fila, 'VARIEDAD'], df.loc[fila, 'MODO'], df.loc[fila, 'CAMPAÑA'])
```

[84]:
```
# Vamos a añadir ahora las variables de superficies. Por si hubieran diferencias significativas
```

```
# entre crecimiento y decrecimiento de superficies tomaremos las valores␣
 ↪positivos y negativos
# por separado.
cambios_superficies=df[df['CAMPAÑA']>19].groupby(['ID_FINCA', 'VARIEDAD',␣
 ↪'CAMPAÑA']).agg({'SUPERFICIE': 'sum', 'PRODUCCION': 'sum'})
cambios_superficies
```

[84]:

ID_FINCA	VARIEDAD	CAMPAÑA	SUPERFICIE	PRODUCCION
200	59	20	0.3700	545.400
439	9	21	1.0800	1901.402
		22	1.0800	0.000
447	40	20	0.4694	2828.540
		21	0.4694	2037.340
…			…	…
99693	81	20	6.3500	26876.300
		21	6.3500	35418.700
		22	6.3397	0.000
99793	52	22	0.1326	0.000
	87	22	0.0189	0.000

[3025 rows x 2 columns]

[85]:
```
# Esta función calcula el crecimiento porcentual de la superficie de una finca␣
 ↪y variedad. Solo
# considera crecimientos positivos, los decrecimientos son considerados en la␣
 ↪función definida a
# continuación.
def crecimiento_superficie(finca, variedad, campaña, df=cambios_superficies):
    actual = df.loc[(finca, variedad, campaña), 'SUPERFICIE']
    try:
        anterior = df.loc[(finca, variedad, campaña - 1), 'SUPERFICIE']
    except KeyError:
        anterior = actual
    try:
        aum = actual / anterior - 1
    except:
        aum = 0
    if aum < 0:
        aum = 0
    return aum
```

[86]:
```
# Esta función calcula el decrecimiento porcentual de la superfice de una finca␣
 ↪y variedad.
def descenso_superficie(finca, variedad, campaña, df=cambios_superficies):
    actual = df.loc[(finca, variedad, campaña), 'SUPERFICIE']
    try:
```

```
        anterior = df.loc[(finca, variedad, campaña - 1), 'SUPERFICIE']
    except KeyError:
        anterior = actual
    try:
        desc = actual / anterior - 1
    except:
        desc = 0
    if desc > 0:
        desc = 0
    return desc
```

[87]:
```
# Añadimos los valores de los cambios de superficie a al dataframe df.
df['Aumento superficie']=0
for fila in range(len(df)):
    if df.loc[fila, 'CAMPAÑA'] >= 21:  #Para 2020 hacia atrás no tenemos datos␣
 ↪de cambios de
        # superficie
        df.loc[fila, 'Aumento superficie']=crecimiento_superficie(df.loc[fila,␣
 ↪'ID_FINCA'], df.loc[fila, 'VARIEDAD'], df.loc[fila, 'CAMPAÑA'])
        df['Descenso superficie']=0
for fila in range(len(df)):
    if df.loc[fila, 'CAMPAÑA'] >= 21:
        df.loc[fila, 'Descenso superficie']=descenso_superficie(df.loc[fila,␣
 ↪'ID_FINCA'], df.loc[fila, 'VARIEDAD'], df.loc[fila, 'CAMPAÑA'])
```

[88]:
```
# Esta función calcula Producción_t / Producción_t-1 para aquellas filas en las␣
 ↪que es posible.
def tasa_prod(finca, variedad, modo, campaña, df=df_betas):
    try:
        prod = df[(finca, variedad, modo)].iloc[(campaña-14)]
        prod_ant = df[(finca, variedad, modo)].iloc[(campaña-15)]
        tasa = prod / prod_ant
    except:
        tasa = np.nan
    return tasa
```

[89]:
```
# Añado a la columna 'Tasa producción' el crecimiento porcentual anual de la␣
 ↪producción para cada
# finca, variedad y modo.
df['Tasa producción']=0
for fila in range(len(df)):
    df.loc[fila, 'Tasa producción'] = tasa_prod(df.loc[fila, 'ID_FINCA'], df.
 ↪loc[fila, 'VARIEDAD'], df.loc[fila, 'MODO'], df.loc[fila, 'CAMPAÑA'])
```

[90]:
```
# La función beta que se definió anteriormente devuelve cuánto crece la␣
 ↪producción tendencial
# absoluta cada año.
```

```
# Con la función definida a continuación vamos a estimar el porcentaje de␣
 ↪crecimiento de producción
# respecto al año anterior con la fórmula beta / E[producción].
def tasa_beta(finca, variedad, modo, campaña, df=df_betas):
    lr = LinearRegression()
    if df[(finca, variedad, modo)].iloc[(campaña-18):(campaña-14)].isnull().
 ↪values.any(): #Si hay
        #valores missing no hay producción y por tanto no hay tendencia
        tasa = 1
    else:
        x = np.arange(4).reshape(-1,1)
        y = df[(finca, variedad, modo)].iloc[(campaña-18):(campaña-14)].values
        lr.fit(x,y)
        if lr.score(x,y)>0.95: # Si R^2 menor que cierta cota, consideramos que␣
 ↪no hay tendencia
            beta = lr.coef_
            tasa = beta / np.mean(df[(finca, variedad, modo)].iloc[(campaña-18):
 ↪(campaña-14)].values)
        else:
            tasa = 1
    return tasa
```

[91]:
```
# Añadimos  a la columna 'Tasa betas' el crecimiento porcentual tendencial de␣
 ↪la producción por
# finca, variedad y modo.
df['Tasa betas'] = 1
for fila in range(len(df)):
    if df.loc[fila, 'CAMPAÑA'] in range(18,22):  #Para 2017 hacia atrás no␣
 ↪calculamos tendencias por no haber datos suficientes
        df.loc[fila, 'Tasa betas']=tasa_beta(df.loc[fila, 'ID_FINCA'], df.
 ↪loc[fila, 'VARIEDAD'], df.loc[fila, 'MODO'], df.loc[fila, 'CAMPAÑA'])
```

[92]:
```
import pickle

# En el jupyter correspondiente al análisis explotario del dataset METEO hemos␣
 ↪calculado variables
# meteorológicas relevantes para la producción. Basándonos en criterios␣
 ↪científicos explicados en
# el pdf utilizaremos como variables las precipitaciones y temperaturas en␣
 ↪enero-marzo, abril-mayo
# junio.
with open('precip_mes_16.pkl', 'rb') as f:
    precip_mes_16 = pickle.load(f)
with open('precip_mes_17.pkl', 'rb') as f:
    precip_mes_17 = pickle.load(f)
with open('precip_mes_18.pkl', 'rb') as f:
```

```
    precip_mes_18 = pickle.load(f)
with open('precip_mes_19.pkl', 'rb') as f:
    precip_mes_19 = pickle.load(f)
with open('precip_mes_20.pkl', 'rb') as f:
    precip_mes_20 = pickle.load(f)
with open('precip_mes_21.pkl', 'rb') as f:
    precip_mes_21 = pickle.load(f)
with open('precip_mes_22.pkl', 'rb') as f:
    precip_mes_22 = pickle.load(f)



with open('t_mes_16.pkl', 'rb') as f:
    t_mes_16 = pickle.load(f)
with open('t_mes_17.pkl', 'rb') as f:
    t_mes_17 = pickle.load(f)
with open('t_mes_18.pkl', 'rb') as f:
    t_mes_18 = pickle.load(f)
with open('t_mes_19.pkl', 'rb') as f:
    t_mes_19 = pickle.load(f)
with open('t_mes_20.pkl', 'rb') as f:
    t_mes_20 = pickle.load(f)
with open('t_mes_21.pkl', 'rb') as f:
    t_mes_21 = pickle.load(f)
with open('t_mes_22.pkl', 'rb') as f:
    t_mes_22 = pickle.load(f)
```

[93]:
```
# Agregamos al dataframe df las columnas correspondientes a las variables␣
 ↪meteorológicas mencionadas.
# Estas variables están diferenciadas no solo por campaña sino también por␣
 ↪estación meteorológica
# permitiendo obtener conclusiones más certeras.
meses=['enero', 'febrero', 'marzo', 'abril', 'mayo', 'junio']
for m in range(len(meses)):
    precip_col = f"precip_{meses[m]}"
    df[precip_col] = np.nan


for c in range(16, 23):
    for e in range(20):
        for m in range(6):
            precip_mes = globals()[f"precip_mes_{c}"][e][m]
            mes_index = meses[m]
            precip_col = f"precip_{mes_index}"
            df.loc[(df['CAMPAÑA'] == c) & (df['ID_ESTACION'] == e), precip_col]␣
 ↪= precip_mes
```

```
for m in range(len(meses)):
    temp_col = f"t_mes_{meses[m]}"
    df[temp_col] = np.nan
for c in range(16, 23):
    for e in range(20):
        for m in range(6):
            temp_mes = globals()[f"t_mes_{c}"][e][m]
            mes_index = meses[m]
            temp_col = f"t_mes_{mes_index}"
            df.loc[(df['CAMPAÑA'] == c) & (df['ID_ESTACION'] == e), temp_col] =␣
 ↪temp_mes

df['Precip invierno'] = df['precip_enero'] + df['precip_febrero'] +␣
 ↪df['precip_marzo']
df['Precip primavera'] = df['precip_abril'] + df['precip_mayo']
df['Precip invierno 2'] = df['Precip invierno']*df['Precip invierno']
df['Precip primavera 2'] = df['Precip primavera']*df['Precip primavera']
df['precip_junio 2'] = df['precip_junio']*df['precip_junio']
df['Temp invierno'] = (df['t_mes_enero'] + df['t_mes_febrero'] +␣
 ↪df['t_mes_marzo']) /3
df['Temp primavera'] = (df['t_mes_abril'] + df['t_mes_mayo']) / 2
df['Temp invierno 2'] = df['Temp invierno']*df['Temp invierno']
df['Temp primavera 2'] = df['Temp primavera']*df['Temp primavera']
df['t_mes_junio 2'] = df['t_mes_junio']*df['t_mes_junio']
```

[94]:
```
df
```

[94]:

	CAMPAÑA	ID_FINCA	ID_ZONA	ID_ESTACION	ALTITUD	VARIEDAD	MODO	\
0	14	76953	515	4	660.000000	26	2	
1	14	84318	515	4	660.000000	26	2	
2	14	85579	340	4	520.000000	32	2	
3	14	69671	340	4	520.000000	32	2	
4	14	14001	852	14	659.097938	81	1	
…	…	…	…	…	…	…	…	
9596	22	37461	239	6	700.000000	52	2	
9597	22	58769	239	6	700.000000	32	2	
9598	22	58769	239	6	700.000000	59	2	
9599	22	88928	239	6	700.000000	40	2	
9600	22	88928	239	6	700.000000	52	2	

	TIPO	COLOR	SUPERFICIE	…	Precip invierno	Precip primavera	\
0	0	1	4.567922	…	NaN	NaN	
1	0	1	4.081958	…	NaN	NaN	
2	0	1	4.076241	…	NaN	NaN	
3	0	1	6.362788	…	NaN	NaN	
4	0	1	4.963560	…	NaN	NaN	
…	…	…	…	…	…	…	

```
9596     0      1    3.680000  …            220.0            212.0
9597     0      1    4.250000  …            220.0            212.0
9598     0      1    4.070000  …            220.0            212.0
9599     0      1    4.572700  …            220.0            212.0
9600     0      1    1.609900  …            220.0            212.0

      Precip invierno 2  Precip primavera 2  precip_junio 2  Temp invierno  \
0                   NaN                 NaN             NaN            NaN
1                   NaN                 NaN             NaN            NaN
2                   NaN                 NaN             NaN            NaN
3                   NaN                 NaN             NaN            NaN
4                   NaN                 NaN             NaN            NaN
…                   …                   …               …              …
9596            48400.0             44944.0           225.0        7.87391
9597            48400.0             44944.0           225.0        7.87391
9598            48400.0             44944.0           225.0        7.87391
9599            48400.0             44944.0           225.0        7.87391
9600            48400.0             44944.0           225.0        7.87391

      Temp primavera  Temp invierno 2  Temp primavera 2  t_mes_junio 2
0                NaN              NaN               NaN            NaN
1                NaN              NaN               NaN            NaN
2                NaN              NaN               NaN            NaN
3                NaN              NaN               NaN            NaN
4                NaN              NaN               NaN            NaN
…                …                …                 …              …
9596       14.806071        61.998464         219.219732     591.152606
9597       14.806071        61.998464         219.219732     591.152606
9598       14.806071        61.998464         219.219732     591.152606
9599       14.806071        61.998464         219.219732     591.152606
9600       14.806071        61.998464         219.219732     591.152606

[9601 rows x 38 columns]
```

[95]:
```
import statsmodels.api as sm
from statsmodels.tools import add_constant
```

[96]:
```
# Llevamos a cabo un primer modelo de regresión con las variables de␣
 ↪crecimiento porcentual
# tendencial, cambios de superficie, precipitaciones y temperatura.

df21 = df[(df['CAMPAÑA'] < 22) & (df['CAMPAÑA'] > 15)]
df_regresion = df21[(df21['PRODUCCION']>5000)].dropna(how='any')
x = df_regresion[['Tasa betas', 'Aumento superficie', 'Descenso superficie',␣
 ↪'Precip invierno', 'Precip primavera',
                  'precip_junio', 'Temp invierno', 'Temp primavera',␣
 ↪'t_mes_junio']]
```

```
x = add_constant(x)
y = df_regresion['Tasa producción']
w = df_regresion['PRODUCCION']

model = sm.WLS(y, x, weights=w)
results = model.fit()
print(results.summary())
```

```
                            WLS Regression Results
==============================================================================
Dep. Variable:        Tasa producción   R-squared:                       0.010
Model:                            WLS   Adj. R-squared:                  0.007
Method:                 Least Squares   F-statistic:                     3.066
Date:                Thu, 08 Jun 2023   Prob (F-statistic):            0.00116
Time:                        16:28:17   Log-Likelihood:                -6704.1
No. Observations:                2709   AIC:                         1.343e+04
Df Residuals:                    2699   BIC:                         1.349e+04
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
=======
                         coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const                  4.4473      2.003      2.221      0.026       0.520
8.374
Tasa betas            -1.5228      0.432     -3.528      0.000      -2.369
-0.677
Aumento superficie    -0.0293      0.682     -0.043      0.966      -1.367
1.309
Descenso superficie    0.2820      1.545      0.183      0.855      -2.747
3.311
Precip invierno        0.0009      0.001      0.845      0.398      -0.001
0.003
Precip primavera      -0.0014      0.002     -0.589      0.556      -0.006
0.003
precip_junio           0.0045      0.004      1.225      0.221      -0.003
0.012
Temp invierno         -0.0184      0.146     -0.126      0.900      -0.304
0.268
Temp primavera        -0.1027      0.197     -0.520      0.603      -0.489
0.284
t_mes_junio            0.0032      0.122      0.026      0.979      -0.235
0.241
==============================================================================
Omnibus:                     5234.772   Durbin-Watson:                   1.331
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          7465211.376
```

```
Skew:                          15.056   Prob(JB):                         0.00
Kurtosis:                     258.402   Cond. No.                     7.15e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 7.15e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[97]:
```
# Dotamos al modelo anterior de mayor libertad dotando de no linealidad a las␣
 ↪variables meteorológicas
# bajo la hipótesis de que la producción dependa de manera no lineal de la␣
 ↪cantidad de lluvias y
# temperatura. Esta hipótesis se basa en que una cantidad moderada de lluvias␣
 ↪en algunos periodos
# podría resulta positiva para la producción mientras que una sequía extrema o␣
 ↪una demasiada cantidad
# de lluvia podría afectar negativamente. De igual modo, un frío o calor␣
 ↪extremo podrían resultar
# inconvenientes para la producción frente a temperaturas más moderadas. Los␣
 ↪criterios científicos
# que respaldan estas hipótesis están reflejados en el pdf.
df_regresion = df[(df['PRODUCCION']>5000)].dropna(how='any')
x = df_regresion[['Tasa betas', 'Aumento superficie', 'Descenso superficie',␣
 ↪'Precip invierno',
                  'Precip primavera', 'precip_junio', 'Precip invierno 2',␣
 ↪'Precip primavera 2',
                  'precip_junio 2', 'Temp invierno', 'Temp primavera',␣
 ↪'t_mes_junio',
                  'Temp invierno 2', 'Temp primavera 2', 't_mes_junio 2']]
x = add_constant(x)
y = df_regresion['Tasa producción']
w = df_regresion['PRODUCCION']

model = sm.WLS(y, x, weights=w)
results = model.fit()
print(results.summary())
```

```
                            WLS Regression Results
==============================================================================
Dep. Variable:        Tasa producción   R-squared:                       0.013
Model:                            WLS   Adj. R-squared:                  0.008
Method:                 Least Squares   F-statistic:                     2.408
Date:                Thu, 08 Jun 2023   Prob (F-statistic):            0.00179
Time:                        16:28:17   Log-Likelihood:                -6699.8
No. Observations:                2709   AIC:                         1.343e+04
```

```
Df Residuals:                     2693   BIC:                         1.353e+04
Df Model:                           15
Covariance Type:            nonrobust
==============================================================================
=======
                           coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const                  -44.1453     30.251     -1.459      0.145    -103.463
15.172
Tasa betas              -1.5554      0.432     -3.601      0.000      -2.403
-0.708
Aumento superficie      -0.0716      0.685     -0.105      0.917      -1.414
1.271
Descenso superficie      0.2628      1.547      0.170      0.865      -2.770
3.295
Precip invierno         -0.0019      0.004     -0.470      0.638      -0.010
0.006
Precip primavera        -0.0114      0.010     -1.111      0.267      -0.031
0.009
precip_junio             0.0287      0.012      2.350      0.019       0.005
0.053
Precip invierno 2    1.206e-05   1.65e-05      0.733      0.464   -2.02e-05
4.43e-05
Precip primavera 2    5.63e-05    4.8e-05      1.173      0.241   -3.78e-05
0.000
precip_junio 2          -0.0004      0.000     -2.129      0.033      -0.001
-3.15e-05
Temp invierno           -0.2698      1.480     -0.182      0.855      -3.171
2.632
Temp primavera           1.8553      3.986      0.465      0.642      -5.960
9.671
t_mes_junio              3.3151      3.686      0.900      0.368      -3.912
10.542
Temp invierno 2          0.0152      0.079      0.191      0.848      -0.140
0.171
Temp primavera 2        -0.0707      0.133     -0.532      0.595      -0.331
0.190
t_mes_junio 2           -0.0751      0.083     -0.902      0.367      -0.238
0.088
==============================================================================
Omnibus:                     5229.024   Durbin-Watson:                   1.330
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          7432808.834
Skew:                          15.019   Prob(JB):                         0.00
Kurtosis:                     257.848   Cond. No.                     2.13e+07
==============================================================================
```

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.13e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

1.1.3 Conclusiones

Los modelos de regresión anteriores no aportan conclusiones claras. Los cambios de superficie no son significativos y las precipitaciones en invierno y primavera no son significativas. La lluvia en primavera parece influir negativamente sobre la producción mientras que en junio influye positivamente, lo cual resulta contrario a la intuición científica. La variable 'Tasa betas', que indica la variación tendencial de las fincas tampoco ofrece resultados interpretables pues ante un crecimiento tendencial positivo indica una notable caída esperada de producción. Además, la explicabilidad de la varianza es prácticamente nula con un R^2 de 0.01 y 0.013 respectivamente. Antes esta situación, dar una predicción de los cambios de producción en base a la cantidad de lluvias y precipitaciones no parece un método robusto. Sin embargo, dado que algunas variables son significativas al 10%, las condiciones meteorológicas podrían resultar de utilidad al estimar la producción utilizando otro modelo.

Ya vimos con anterioridad que los cambios de superficie resultaban sospechosos y poco fiables, y con los modelos de regresión que hemos realizado concluimos que no son significativos. Por ello, decidimos desechar las variables de superficie del modelo. En cuanto a las condiciones meteorológicas no queda clara su influencia directa sobre la producción, no tenemos certeza sobre cómo dichas condiciones influyen sobre la producción puesto que el número de campañas disponibles es reducido y existe gran variabilidad en los datos de las fincas. Sin embargo, parece claro que campañas con condiciones meteorológicas similares tendrán una producción esperada muy próxima.

Así, la producción esperada de una finca dependerá de los cambios realizados en la finca (reestructuración de variedades de uva, sustitución de viñas viejas por nuevas, protección de las viñas frente a plagas y condiciones adversas), la tendencia de producción de la finca (aumentos o descensos tendenciales de la producción debidos a contratación de personal, expectativas financieras, condiciones del suelo) y las condiciones meteorológicas (años meteorológicos parecidos ofrecen producciones similares).

Concluimos entonces que la baja capacidad predictiva de los modelos lineales presentados se debe a la omisión de variables relevantes y a la indebida estructuración de variables al tratar con datos de panel. No podemos garantizar por ejemplo que un aumento de precipitaciones en primavera repercuta positivamente en la producción de una finca. Para aquellas fincas situadas en una estación meteorológica que haya sufrido sequías en años anteriores un aumento de precipitaciones se prevé positivo para la futura cosecha mientras que en aquellas fincas situadas en estaciones con precipitaciones abundantes todos los años una mayor cantidad de lluvia no repercutirá positivamente en la producción e incluso puede repercutir negativamente.

El razonamiento teórico y las conclusiones extraídas del análisis realizado indican que, aunque no seamos capaces de dilucidar la implicación directa de las variables meteorológicas sobre la producción, la premisa de que años meteorológicos similares ofrecen, ceteris paribus, capacidades productivas similares es sólida. Por ello, vamos a plantear un modelo de predicción que tenga en cuenta la evolución temporal de la producción, de modo que considere cambios tendenciales y reestructuraciones entre variedades de uva, además de las similitudes meteorológicas entre campañas

para la estación en la que se encuentre cada finca de modo que la predicción para la campaña 22 sea próxima a las campañas con mayor similitud.

[ ]:

[ ]: