INFORME FINAL

ALEJANDRA RENTERÍA TENORIO

ING. LUIS ARMANDO AMAYA

LIMPIEZA DE DATOS

SERVICIO NACIONAL DE APRENDIZAJE

TÉCNICO EN PROGRAMACIÓN DE ANALÍTICA DE DATOS

SANTIAGO DE CALI, OCTUBRE 2020

CONTENIDO

GLOSARIO

Análisis: Examen detallado de una cosa para conocer sus características o cualidades, o su estado, y extraer conclusiones, que se realiza separando o considerando por separado las partes que la constituyen.

Big Data: Es un término evolutivo que describe cualquier cantidad voluminosa de datos estructurados, semiestructurados y no estructurados que tienen el potencial de ser extraídos para obtener información.

Calidad de los Datos: Es la cualidad de un conjunto de información recogida en una base de datos, un sistema de información o un data warehouse que reúne entre sus atributos la exactitud, completitud, integridad, actualización, coherencia, relevancia, accesibilidad y confiabilidad.

Ciencia de Datos: Campo interdisciplinario que utiliza métodos, procesos, algoritmos y sistemas científicos para revelar tendencias y generar información que las empresas pueden utilizar para tomar mejores decisiones y crear productos y servicios más innovadores.

Conocimiento: Hechos o información adquiridos por una persona a través de la experiencia o la educación, la comprensión teórica o práctica de un asunto referente a la realidad.

CRISP-DM: Se trata de un modelo estándar abierto del proceso que describe los enfoques comunes que utilizan los expertos en minería de datos.

CRM: Custormer relationship management, es un término que se refiere a las prácticas, estrategias y sistemas que las empresas utilizan para gestionar y analizar las interacciones con los clientes y los datos que se generan.

Cualitativo: Aquello que está relacionado con la cualidad o con la calidad de algo, es decir, con el modo de ser o con las propiedades de un objeto, un individuo, una entidad o un estado.

Cuantitativo: Variables estadísticas que otorgan, como resultado, un valor numérico.

Dato: Cifra, letra o palabra que se suministra a la computadora como entrada y la máquina almacena en un determinado formato.

Deep Data: Una recolección a gran escala de datos que son también de alta calidad y procesables». En otras palabras: mucha información, rápidamente analizable, de alto valor y con un fin claro y definido de antemano.

Descriptiva: Define algún tema, y consiste en representar con palabras el aspecto o apariencia de una persona, animal, objeto, paisaje, lugar, cosa, situación, etc.

Diagnóstico: Alude al análisis que se realiza para determinar cualquier situación y cuáles son las tendencias. Esta determinación se realiza sobre la base de datos y hechos recogidos y ordenados sistemáticamente.

Eficacia: Equilibrio entre eficacia y eficiencia, es decir, se es efectivo si se es eficaz y eficiente. La eficacia es lograr un resultado o efecto.

Eficiencia: es la capacidad de disponer de alguien o algo para conseguir el cumplimiento adecuado de una función.

Enfoque: Manera de ver las cosas o las ideas y en consecuencia también de tratar los problemas relativos a ellas.

Estadística: Ciencia que utiliza conjuntos de datos numéricos para obtener, a partir de ellos, inferencias basadas en el cálculo de probabilidades.

Etapa: Es un período de tiempo delimitado y contrapuesto siempre con un momento anterior y con otro posterior.

ETL: Extract, Transform and Load es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos.

Información: Conjunto organizado de datos procesados que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje

Informática: Ciencia que administra métodos, técnicas y procesos con el fin de almacenar, procesar y transmitir información y datos en formato digital.

Inteligencia de Negocios: Combinación de tecnología, herramientas y procesos que me permiten transformar mis datos almacenados en información, esta información en conocimiento y este conocimiento dirigido a un plan o una estrategia comercia

Investigación: Es una actividad orientada a la obtención de nuevos conocimientos y su aplicación para la solución a problemas o interrogantes de carácter científico

Jupyter Notebook: Es una organización sin ánimo de lucro creada para "desarrollar software de código abierto, estándares abiertos y servicios para computación interactiva en docenas de lenguajes de programación

KDD: La extracción de conocimiento es la creación de conocimiento a partir de fuentes estructuradas y no estructuradas.

Modelo: Prototipo que sirve de referencia y ejemplo para todos los que diseñan y confeccionan productos de la misma naturaleza.

Negocio: Consiste en un método de formar u obtener dinero a cambio de productos, servicios, o cualquier actividad que se quiera desarrollar.

Predictiva: Agrupa una variedad de técnicas estadísticas de modelización, aprendizaje automático y minería de datos que analiza los datos actuales e históricos reales para hacer predicciones acerca del futuro o acontecimientos no conocidos.

Prescriptiva: Es «lo que debería ser». Se trata de identificar cual es la mejor forma de hacer las cosas. Se establecen leyes, normas, acuerdos psicológicos, etc. de cómo hacer las cosas**.**

Proceso: Conjunto o encadenamiento de fenómenos, asociados al ser humano o a la naturaleza, que se desarrollan en un periodo de tiempo finito o infinito y cuyas fases sucesivas suelen conducir hacia un fin específico.

Productividad: Se define como la cantidad de producción de una unidad de producto o servicio por insumo de cada factor utilizado por unidad de tiempo.

Python: Es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Rendimiento: Hace referencia al resultado deseado efectivamente obtenido por cada unidad que realiza la actividad económica.

Rentabilidad: Relación existente entre los beneficios que proporciona una determinada operación o cosa y la inversión o el esfuerzo que se ha hecho; cuando se trata del rendimiento financiero; se suele expresar en porcentajes.

RStudio: Entorno de desarrollo integrado para el lenguaje de programación R, dedicado a la computación estadística y gráficos.

Síntesis: Exposición breve, escrita u oral, que a modo de resumen contiene un conjunto de ideas fundamentales y relacionadas con un asunto o materia y que estaban dispersas

Sistema Operativo: Es el software principal o conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación de software, ejecutándose en modo privilegiado respecto de los restantes.

Smart Data: Hace referencia a la transformación de largas listas de números y datos en información con valor, con respuesta, útil

Toma de decisiones: Proceso mediante el cual se realiza una elección entre las opciones o formas para resolver diferentes situaciones de la vida en diferentes contextos: a nivel laboral, familiar, personal, sentimental o empresarial (utilizando metodologías cuantitativas que brinda la administración).

Validación: El proceso de revisión que verifica que el sistema de software producido que cumple con las especificaciones y que logra su cometido.

Variable: Característica que puede fluctuar y cuya variación es susceptible a adoptar diferentes valores, los cuales pueden medirse u observarse

INTRODUCCIÓN

El café es uno de los principales productos de exportación del país, y uno por los que se conoce a Colombia internacionalmente. Se cultiva en diferentes regiones a lo largo de todo el territorio, es por su sabor y frescura, resultantes de climas y topografías propias de cada región (Buencafé, 2019). De acuerdo con la Federación Nacional de Cafeteros (FNC) de Colombia, organismo que representa y agrupa a los caficultores de todo el territorio, la cuidadosa selección de los granos de café durante sus etapas de cosecha, y postcosecha (despulpado, lavado, secado y trilla), aseguran una calidad única.

Dada la importancia que este fruto le genera a nuestro país, se vuelve de gran interés para todos los ciudadanos, en especial para los campesinos, que a diario manejan la industria del café.

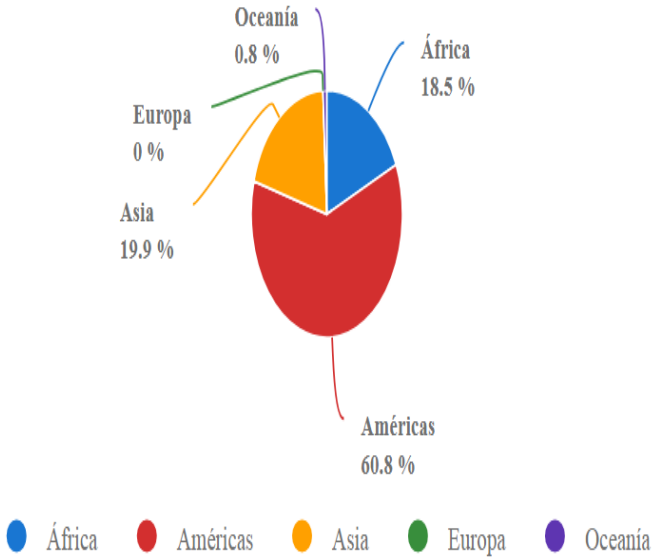
Por tal motivo genera la necesidad de realizar una previa investigación a través de herramientas de amplia especificación como lo son Jupyter Notebook, Pandas Profiling, Excel, entre otras. Las cuales permiten general informes, datos de gran importancia para recrear el modelo predictivo más óptimo.

Este proceso se logra llevar a cabo con la ayuda de algoritmos e instrucciones que especifiquen soluciones puntuales como el mejor rendimiento, su variabilidad, correlación, etc.

CONSULTAS DEL CAFÉ

REPRESENTACION DEMOGRAFICA:

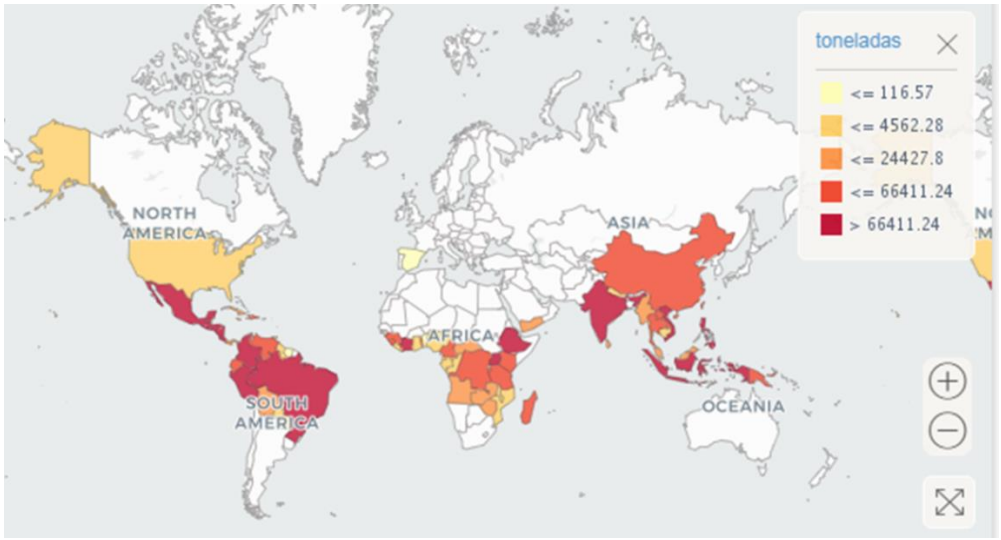
Promedio 1961 – 2018, de producción del café verde por regiones, donde Europa representa el 0%, seguida de Oceanía con un 0.8% en los porcentajes mas bajos



Enlace: http://www.fao.org/faostat/es/#data/qc/visualize

Fuente: Faostat

HISTOGRAMA DE PRODUCCIÓN EN CAFÉ VERDE EN EL MUNDO:

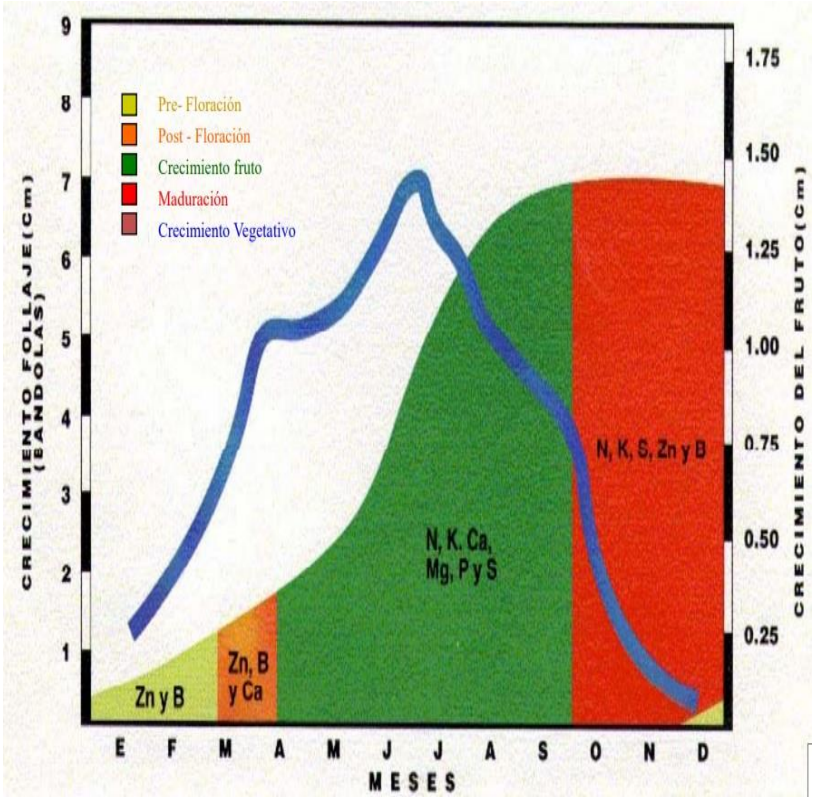


Las denominaciones empleadas en los mapas y la forma en que aparecen presentados los datos implican, por parte de la FAO, en el podemos apreciar que la zona que mas cosecha el café esta ubicada en Suramérica y Centroamérica obteniendo más de 66411.24 toneladas.

Enlace: http://www.fao.org/faostat/es/#data

Fuente: Faostat

CRECIMIENTO DE LA PLANTA DEL CAFÉ:



El mes en el que la planta alcanza su mayor altura es en junio y julio

Enlace: http://sintrainduscafe.org/secciones/en-que-epoca-aplicar-el-fertilizante-en-el-cultivo-del-cafe/

Fuente: Curso fertilización, soy fertilizador

PRODUCCIÓN DE CAFÉ EN LATINOAMÉRICA Y ESPAÑA:

La grafica muestra la cantidad de café producido en Latinoamérica y España durante el año 1980 hasta el 2013, arrojando a Brasil como el numero uno y a España como el menor productor.

Enlace: https://www.youtube.com/watch?v=avhblqstHXw

Fuente: Food and Agriculture Organization of the United Nations (FAO)

BALANCE CAFETERO:

La tabla nos permite mostrar la viabilidad del café, donde en el 2016, 2017 y 2018 con una pérdida del 10%

Enlace: https://cdn.flipsnack.com/widget/v2/widget.html?hash=dpazs597t9

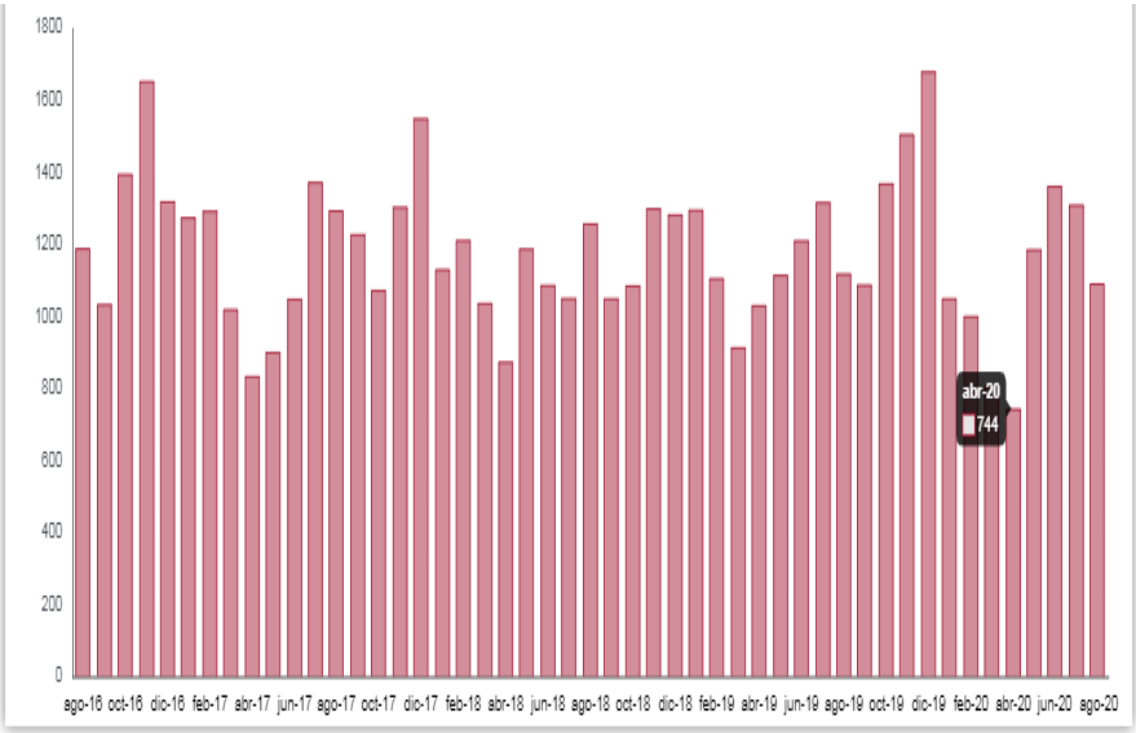
Fuente: Federación nacional de cafeteros Colombianos

BALANCE CAFETERO

	2015	2016	2017	2018	2019
Producción e importaciones	**14,4**	**14,5**	**14,6**	**14,5**	**15,7**
Producción	14,2	14,2	14,2	13,6	14,8
Importaciones	0,2	0,3	0,4	0,9	1,0
Exportaciones y consumo	**14,4**	**14,6**	**14,7**	**14,6**	**15,5**
Exportaciones	12,7	12,9	13,0	12,8	13,7
Consumo interno	1,7	1,7	1,7	1,8	1,8
Balance	**0,0**	**-0,1**	**-0,1**	**-0,1**	**0,2**

Fuente: FNC

PRODUCCION MENSUAL



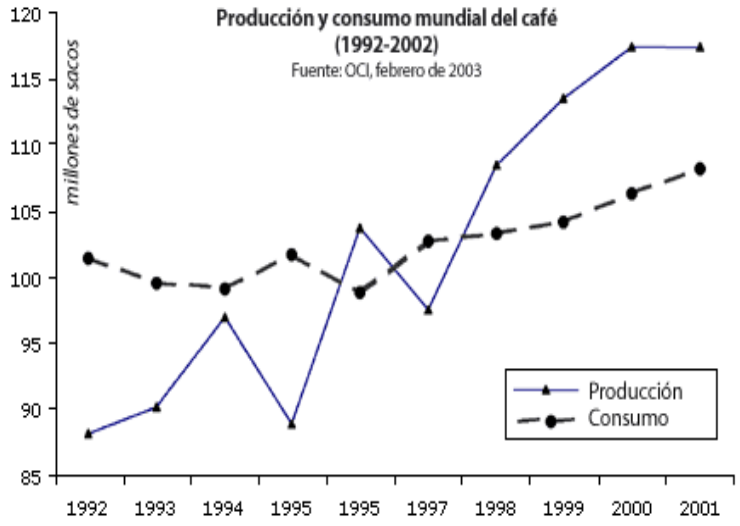
Producción mensual de café verde en miles de sacos de 60 kg.

Se deduce que el mes que menos producción tuvo fue en abril/20con un 744 millones de saco de 60 kg

Enlace: https://federaciondecafeteros.org/wp/estadisticas-cafeteras/

Fuente: Federación nacional de cafeteros Colombianos

PRODUCCION / CONSUMO MUNDIAL DEL CAFE

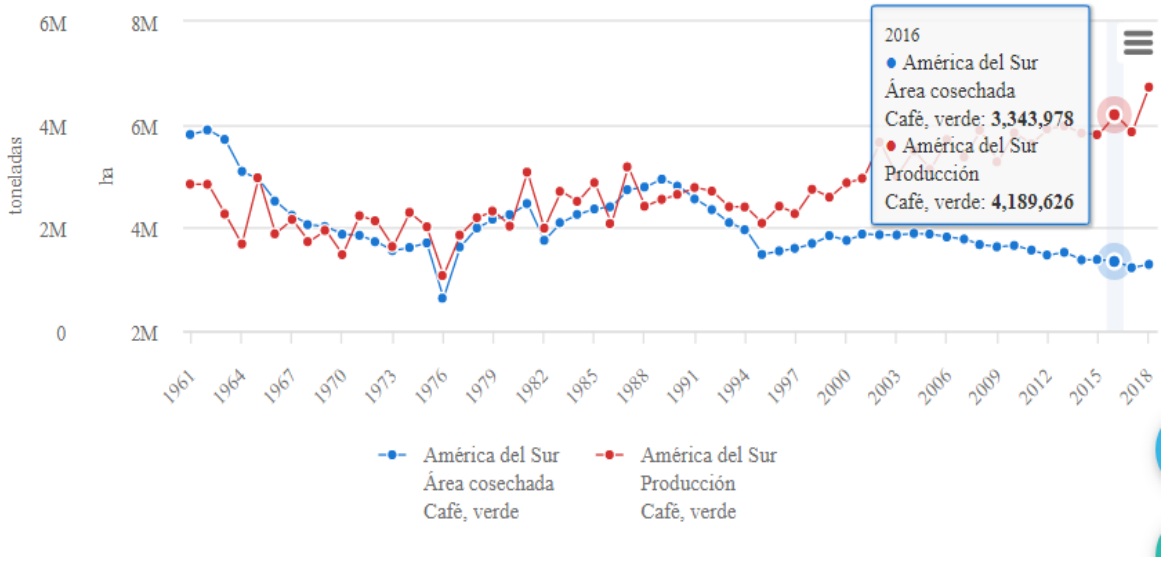


La crisis que atraviesa la industria del café obedece a que la producción es mayor que la demanda. Esta situación ha conducido a una disminución notable de la calidad y a la caída de la producción. Con la mas baja producción fue en 1992 y 1995.

Enlace: http://www.ico.org/projects/goodhygienepractices/cnt/cnt_sp/sec_1/c03.coffeecrisis.html

Fuente: Sección 1

PRODUCCIÓN / RENDIMIENTO



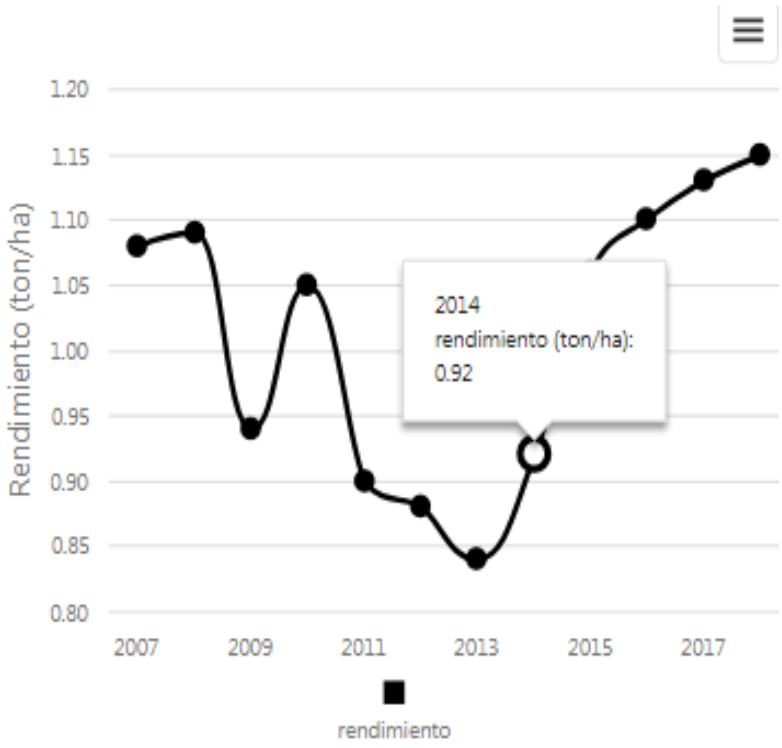
Desde el año 1994 al 2018 se ve que el rendimiento del café no genera mucha dispersión; independiente de la producción la cual asciende.

Enlace:

http://www.fao.org/faostat/es/#data/QC/visualize

Fuente: Agronet

RENDIMIENTO DEL CAFÉ VERDE EN COLOMBIA ENTRE EL 2007 Y EL 2008

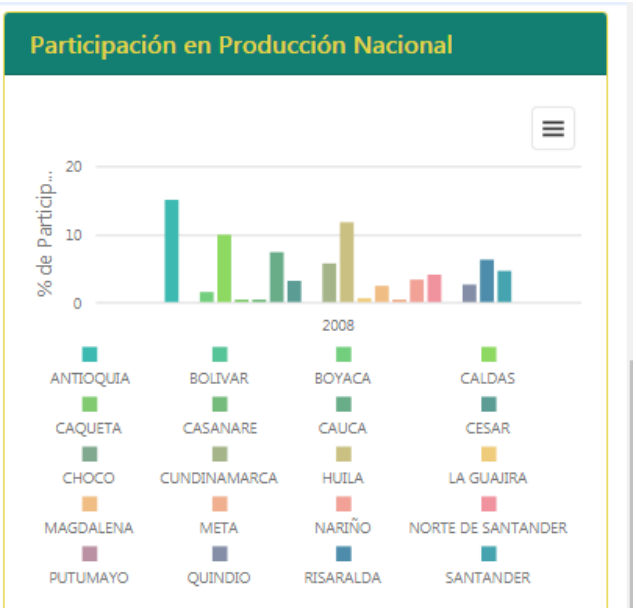


Con respecto al rendimiento del café notamos que después del año 2014 se ha tenido una gran ascendencia de su rendimiento

Enlace:
http://www.fao.org/faostat/es/#data/QC/visualize

Fuente: Agronet

RENDIMIENTO Y PARTICIPACION POR DEPARTAMENTO

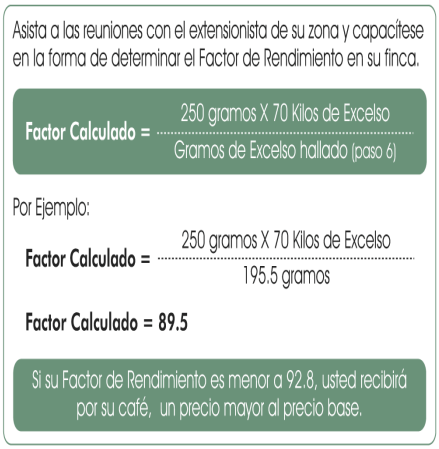


En el área de participación, se podría decir que el departamento Antioquia es el doble del departamento de Casanare

Enlace: http://www.fao.org/faostat/es/#data/QC/visualize

Fuente: Agronet

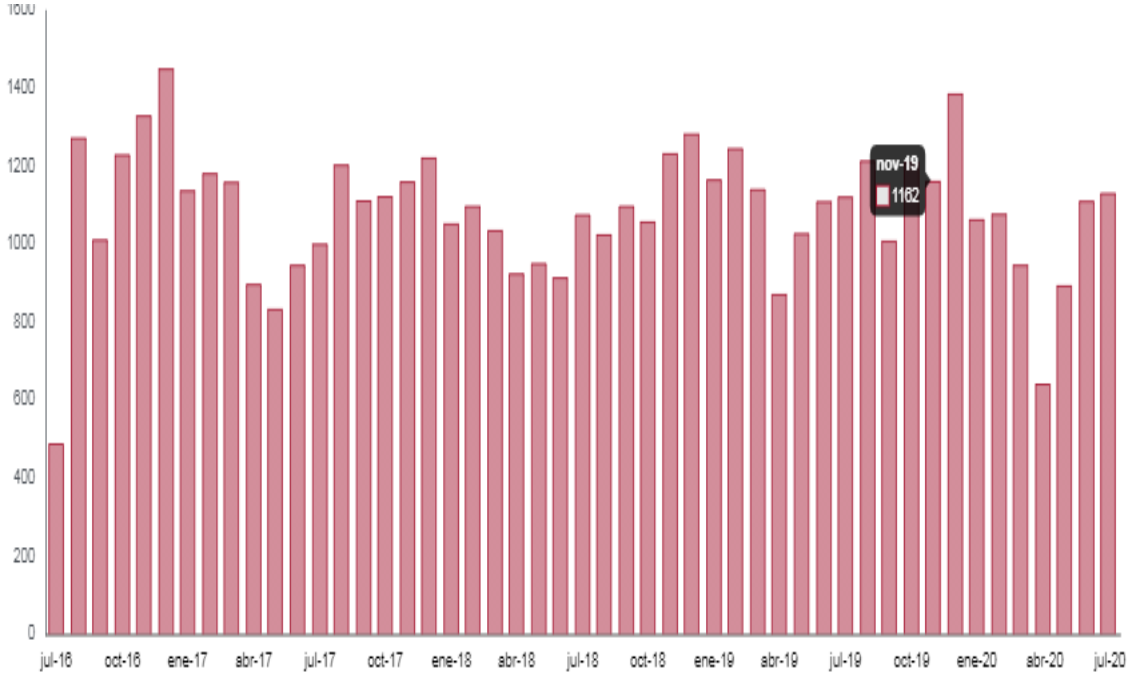
FACTOR DE RENDIMIENTO



Enlace: https://federaciondecafeteros.org/wp/servicios-al-caficultor/aprenda-a-vender-su-cafe/

Fuente: Federación nacional de cafeteros Colombianos

EXPORTACIONES



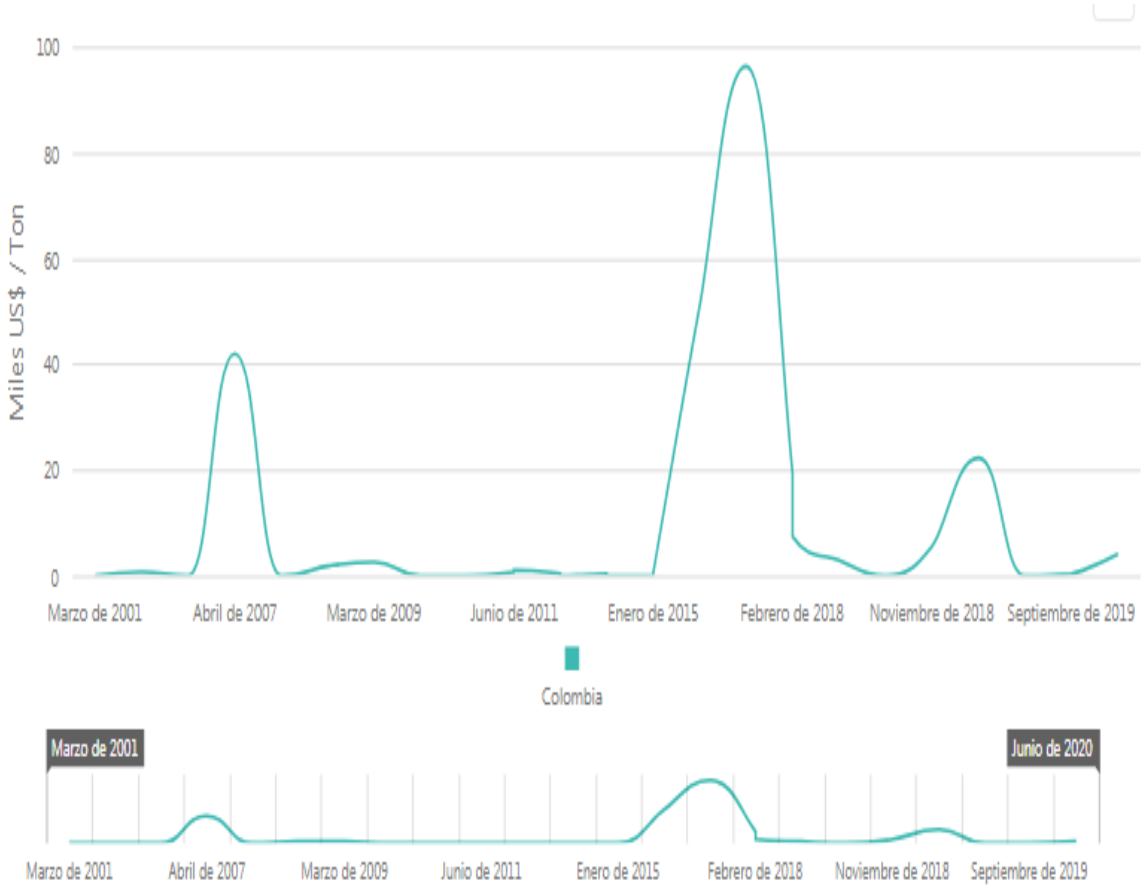
Exportaciones mensuales de café verde equivalente en miles de sacos de 60 kg.

En la grafica se puede observar una similitud en el las mejores exportación en Enero /17 y Enero /20 con aproximadamente unos 1.447 miles de sacos de 60 kg

Enlace: https://federaciondecafeteros.org/wp/estadisticas-cafeteras/

Fuente: Federación nacional de cafeteros Colombianos

PRECIO DE EXPORTACIONES DEL CAFE

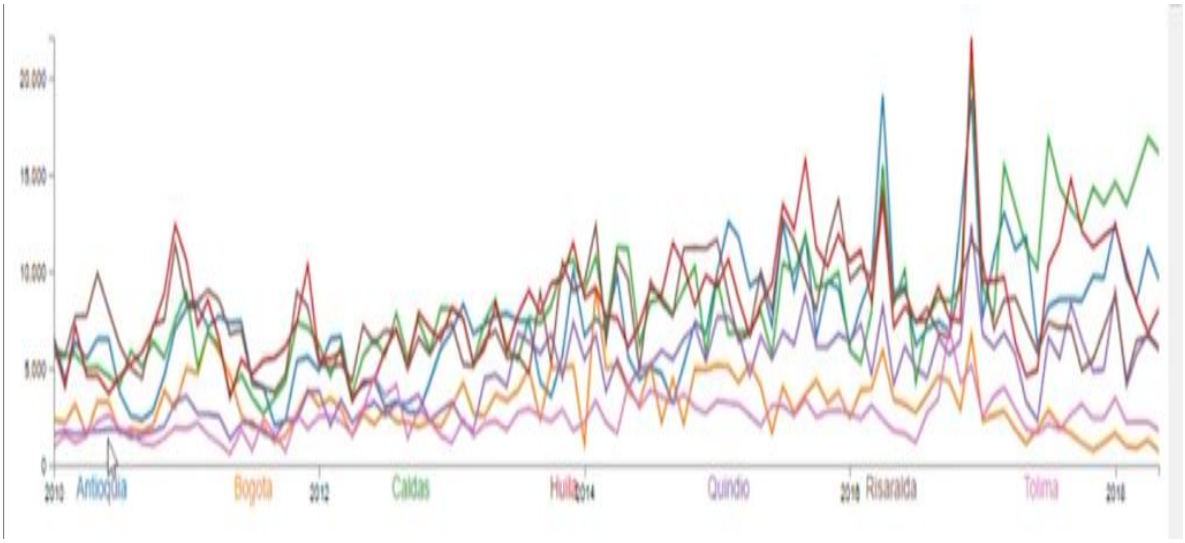


El precio de la exportaciones del café verde tu una gran demanda entre Enero/2015 y Febrero/2018

Enlace: http://www.fao.org/faostat/es/#data/QC/visualize

Fuente: Agronet

EXPORTACIONES DE 5 DEPARTAMENTOS

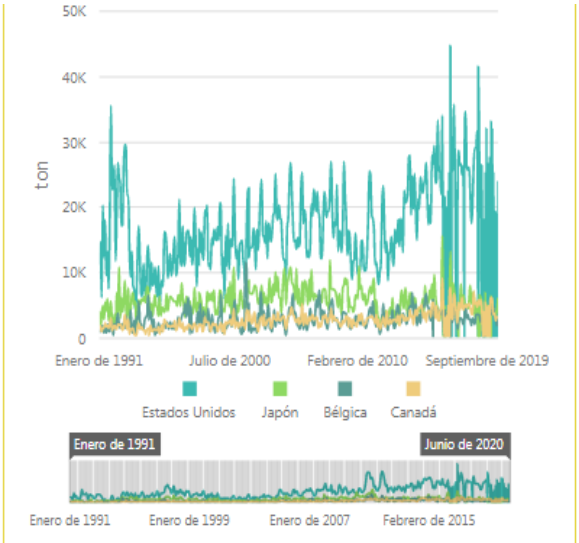


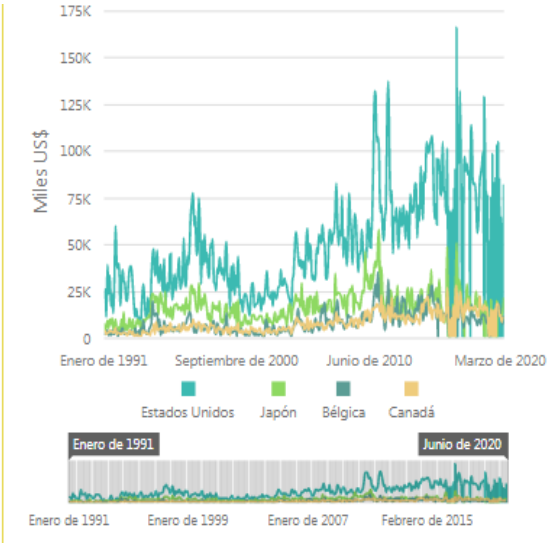
Aquí apreciamos las exportaciones que realizan siete departamentos de Antioquia, Bogotá, Caldas, Huila, Quindío, Risaralda y Tolima con un eje y en toneladas y un eje x de tiempo, en el que Huila lideró en el año 2017.

Enlace: https://www.youtube.com/watch?v=VEnPtvYNaKo&t=590s

Fuente: Canal Harry Cristhian Torres Moreno

DESTINO DE EXPORTACIONES





El menores exportaciones del café colombiano entre 2019 y 2020 se ven en Bélgica y Canadá, y país potencia para exportar es EE. UU.

Enlace: http://www.fao.org/faostat/es/#data/QC/visualize

Fuente: Agronet

VALOR DE COSECHA DEL GRANO DEL CAFÉ EN COLOMBIA DEL 2000-2018

AÑO	VALOR DE LA COSECHA
2000	$2.279.049
2001	$1.959.278
2002	$2.120.915
2003	$2.244.566
2004	$2.668.500
2005	$3.457.525
2006	$3.606.896
2007	$3.818.514
2008	$3.825.079
2009	$3.400.159
2010	$4.365.726
2011	$4.923.317
2012	$3.404.701
2013	$3.375.986
2014	$5.197.328
2015	$6.242.192
2016	$7.109.274
2017	$7.512.632
2018	$6.235.196

Vemos que el mayor valor de cosecha fue en 2017 con 7.512.632

Enlace: https://www.misfinanzasparainvertir.com/la-crisis-del-cafe-impacta-en-colombia/

Fuente: Mis finanzas Davivienda

PRECIO DEL CAFÉ EN EL 2020

Lunes 09 de Marzo del 2020	US$ 1.10	US$ 1.11	US$ 1.13	$ 1,010,000
Domingo 08 de Marzo del 2020	US$ 1.10	US$ 1.11	US$ 1.13	$ 1,010,000
Sábado 07 de Marzo del 2020	US$ 1.10	US$ 1.11	US$ 1.13	$ 1,010,000
Viernes 06 de Marzo del 2020		US$ 1.18	US$ 1.20	$ 1,043,000
Jueves 05 de Marzo del 2020	US$ 1.21	US$ 1.22	US$ 1.24	$ 1,065,000
Miércoles 04 de Marzo del 2020		US$ 1.16	US$ 1.18	$ 1,020,000
Martes 03 de Marzo del 2020	US$ 1.10	US$ 1.11	US$ 1.13	$ 1,014,000
Lunes 02 de Marzo del 2020	US$ 1.08	US$ 1.10	US$ 1.12	$ 997,000
Domingo 01 de Marzo del 2020	US$ 1.08	US$ 1.10	US$ 1.12	$ 997,000
Sábado 29 de Febrero del 2020	US$ 1.08	US$ 1.10	US$ 1.12	$ 997,000
Viernes 28 de Febrero del 2020	US$ 1.09	US$ 1.11	US$ 1.13	$ 990,000
Jueves 27 de Febrero del 2020	US$ 1.07	US$ 1.09	US$ 1.11	$ 968,000
Miércoles 26 de Febrero del 2020	US$ 1.06	US$ 1.07	US$ 1.09	$ 955,000
Martes 25 de Febrero del 2020	US$ 1.09	US$ 1.10	US$ 1.12	$ 985,000
Lunes 24 de Febrero del 2020	US$ 1.04	US$ 1.05	US$ 1.07	$ 907,000
Domingo 23 de Febrero del 2020	US$ 1.04	US$ 1.05	US$ 1.07	$ 907,000
Sábado 22 de Febrero del 2020	US$ 1.04	US$ 1.05	US$ 1.07	$ 907,000
Viernes 21 de Febrero del 2020	US$ 1.07	US$ 1.09	US$ 1.11	$ 930,000
Jueves 20 de Febrero del 2020	US$ 1.06	US$ 1.09	US$ 1.11	$ 930,000
Miércoles 19 de Febrero del 2020	US$ 1.09	US$ 1.11	US$ 1.13	$ 942,000
Martes 18 de Febrero del 2020	US$ 1.09	US$ 1.11	US$ 1.13	$ 942,000
Lunes 17 de Febrero del 2020	US$ 1.04	US$ 1.07	US$ 1.09	$ 890,000
Domingo 16 de Febrero del 2020	US$ 1.04	US$ 1.07	US$ 1.09	$ 890,000
Sábado 15 de Febrero del 2020	US$ 1.04	US$ 1.07	US$ 1.09	$ 890,000
Viernes 14 de Febrero del 2020		US$ 1.03	US$ 1.05	$ 863,000
Jueves 13 de Febrero del 2020	US$ 1.01	US$ 1.03	US$ 1.05	$ 870,000
Miércoles 12 de Febrero del 202	US$ 1.01	US$ 1.03	US$ 1.05	$ 870,000
Martes 04 de Febrero del 2020		US$ 1.05	US$ 1.07	$ 850,000
Lunes 03 de Febrero del 2020	US$ 1.02	US$ 1.04	US$ 1.06	$ 835,000
Domingo 02 de Febrero del 2020	US$ 1.02	US$ 1.04	US$ 1.06	$ 835,000

En lo corrido del años se evidencia que el valor mayor que ha tenido el Café fue el 4 de Mayo con un 1.21 dólares a diferencia del mas bajo el 13 y 12 de Febrero con 1.01 dólares.

Enlace: https://dolar.wilkinsonpc.com.co/cafe.html

Fuente: Dólar web

GRAFICA DE LOS PRECIOS DEL CAFÉ COLOMBIANO

Aquí evidenciamos la grafica de lo visto anteriormente, a diferencia que su rango de selección es del 2006 al 2019, en donde tuvo la cifra mas baja.

Enlace: https://quecafe.info/historia-crisis-cafetera/

Fuente: Quécafe

VIABILIDAD

IMPACTO ECONÓMICO CAFÉ Y COCA 2013

DOLARES USA / TC 2.75

RUBROS/CULTIVOS	CAFÉ	COCA
Hectáreas	425,416	49,800
kilos/ha	612	2,316
Precio/productor (U$)	1.69	4.3
Ingresos/hectárea (U$)	1,032.59	9,958.30
Valor Total Millones U$	439,28	495.95
Resultado	_	81.71

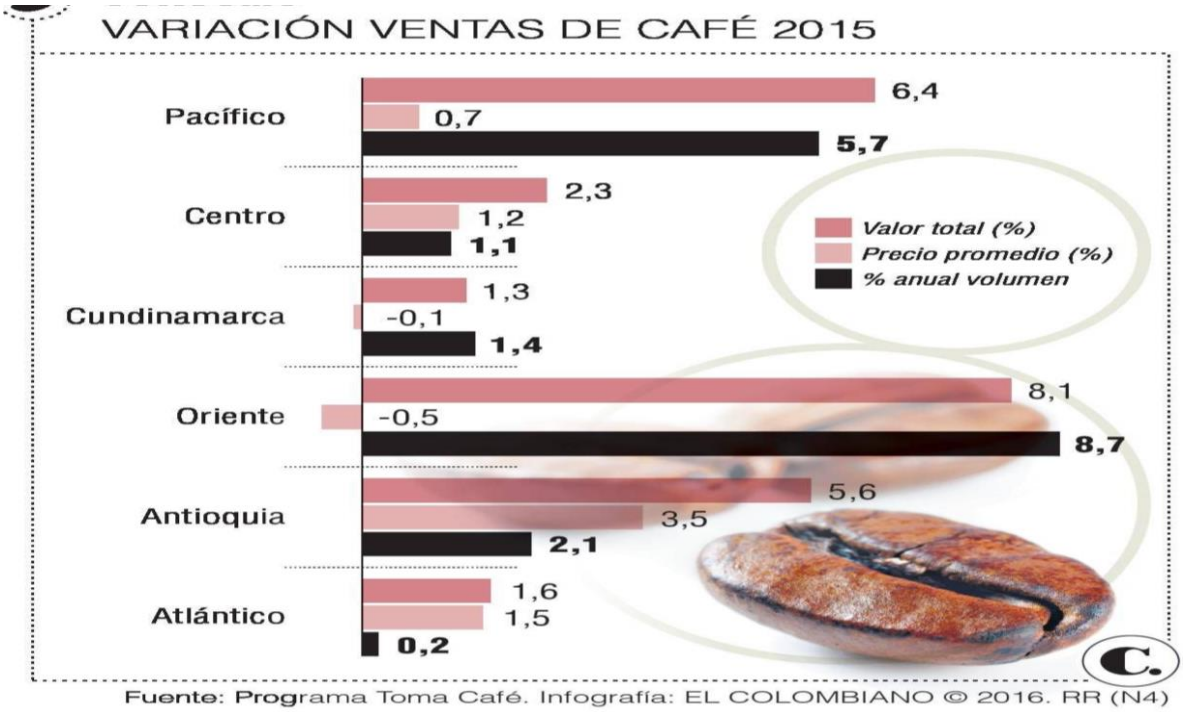
Fuente: UNODC-DEVIDA. Informe Junio 2014 Elaboración: JNC

Esta es una triste realidad que observamos en nuestro país ya que a pesar de que se disponen mas hectáreas y se producen mas kilos de café la coca llegue a poseer mayor ganancias en general sin retribuir todo el proceso e inversiones que pasan los caficultores.

Enlace: https://jhenryhamed07.wixsite.com/sistemaproductivovcc

Fuente: Sistemas productivos

CONSUMO DEL CAFÉ POR ÁREAS



Según la grafica se puede apreciar que los mayores vendedores de café en Colombia esta en el Oriente, a pesar de que son los menores vender a precios promedios

Enlace: https://i.pinimg.com/originals/c8/3d/49/c83d49ba68bdf3b5b4300505ce1e2f84.jpg

Fuente: Programa toma café

ANALISIS DESCRIPTIVO

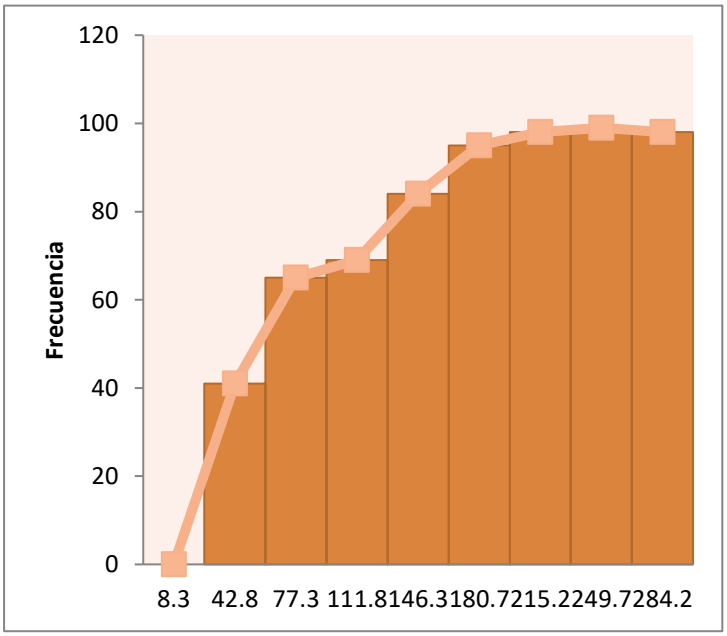
PRECIO EXPORTACION ANUAL DEL CAFÉ

Los datos utilizados para realizar el presente Analisis se centra en lo centavos de dólar por libra de 453.6 gr. Excelso producidos entre 1913 y 2019 en Colombia.

Columna1	
Media	77,88613765
Error típico	6,139421181
Mediana	57,01416667
Moda	15,87
Desviación estándar	63,50666651
Varianza de la muestra	4033,096691
Curtosis	-0,002705977
Coeficiente de asimetría	0,892204052
Rango	275,9327216
Mínimo	8,375833333
Máximo	284,308555
Suma	8333,816728
Cuenta	107

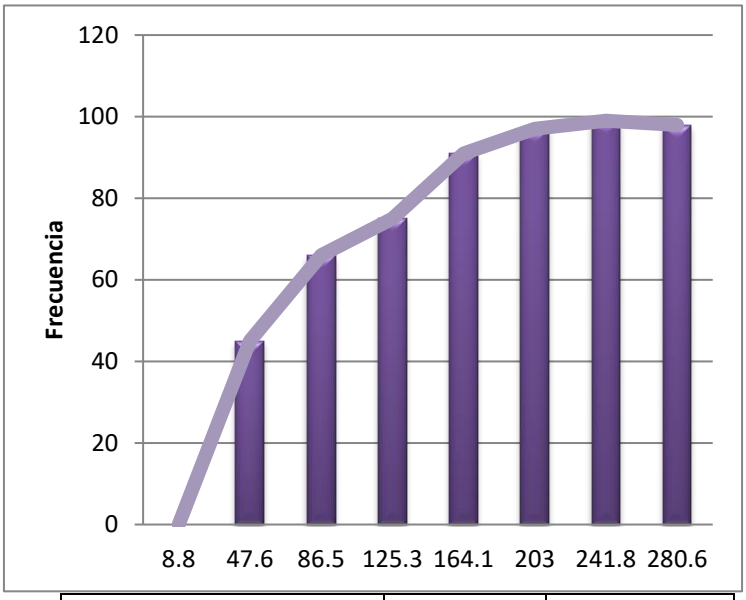
Intervalos		Grupos	Frecuencia
Li	**Ls**		
	8,4	8,3	0
8,4	42,9	42,8	41
42,9	77,4	77,3	65
77,4	111,9	111,8	69
111,9	146,3	146,3	84
146,3	180,8	180,7	95
180,8	215,3	215,2	98
215,3	249,8	249,7	99
249,8	284,3	284,2	98

Numero de clase	7,696966466
Tamaño de clase	34,49159021



PRECIO EXTERNO DURANTE EL AÑO CAFÉ COLOMBIANO

En este apartado se muestran los resultados de la ponderación de los precios centavos de dólar por libra de 453,6 gr. Excelso de los años 1913/14 al 2018/19.



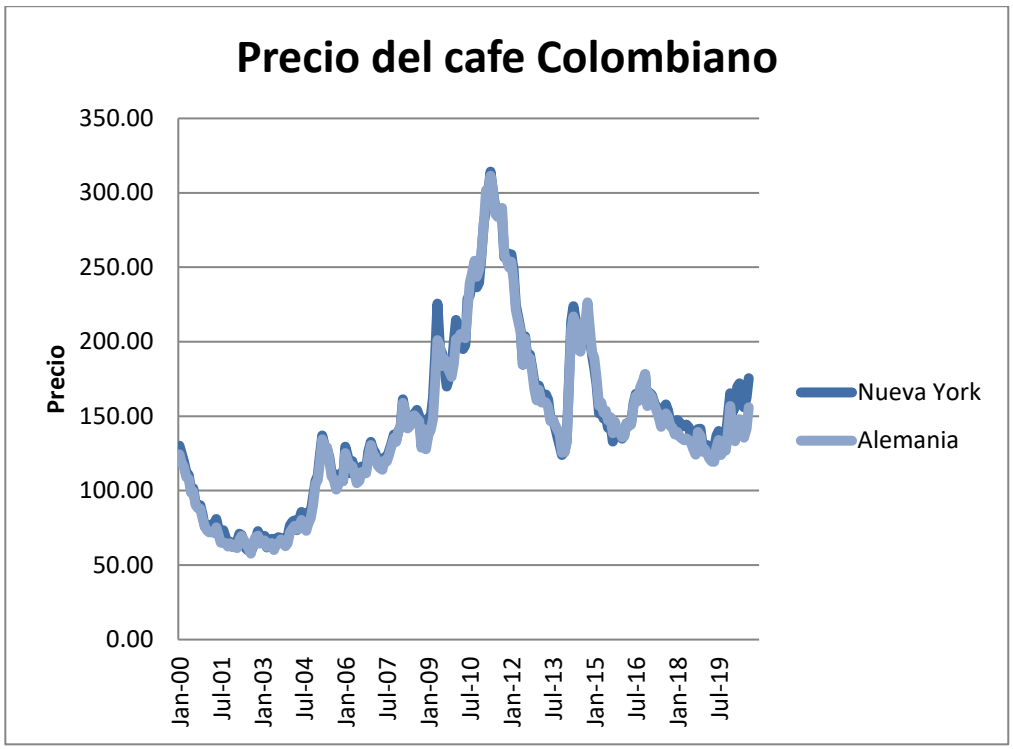
Intervalos		Grupos	Frecuencia
Li	Ls		
	8,9	8,8	0
8,9	47,7	47,6	45
47,7	86,6	86,5	66
86,6	125,4	125,3	75
125,4	164,2	164,1	91
164,2	203,1	203	97
203,1	241,9	241,8	99
241,9	280,7	280,6	98

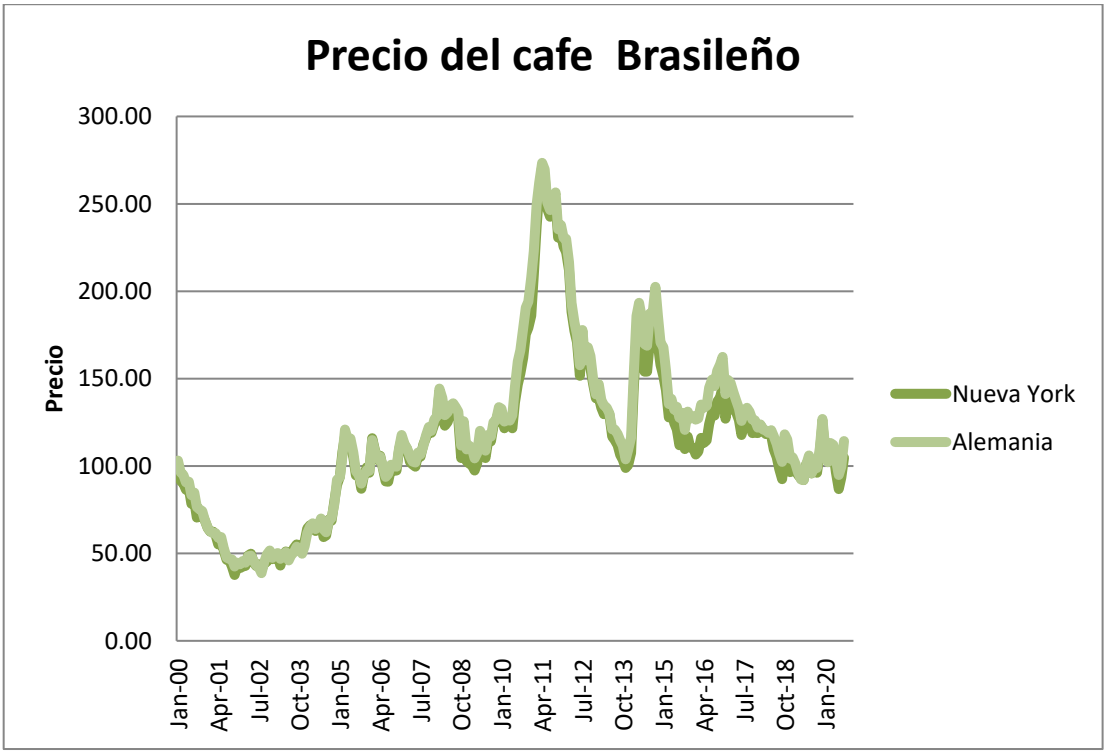
Numero de clase	6,68350936
Tamaño de clase	38,8334524

Columna1	
Media	78,0864614
Error típico	6,14360961
Mediana	57,635
Moda	15,87
Desviación estándar	63,2523323
Varianza de la muestra	4000,85754
Curtosis	-0,06343932
Coeficiente de asimetría	0,8734292
Rango	271,834167
Mínimo	8,90583333
Máximo	280,74
Suma	8277,16491
Cuenta	106

PRECIO INDICATIVO DEL CAFÉ ARABICO EN COLOMBIA Y BRASIL

Vemos una comparación entre los dos mejores países caficultores del mundo hasta el año 2020 vendidos en Estados Unidos y Alemania en centavos de dólar por libra.





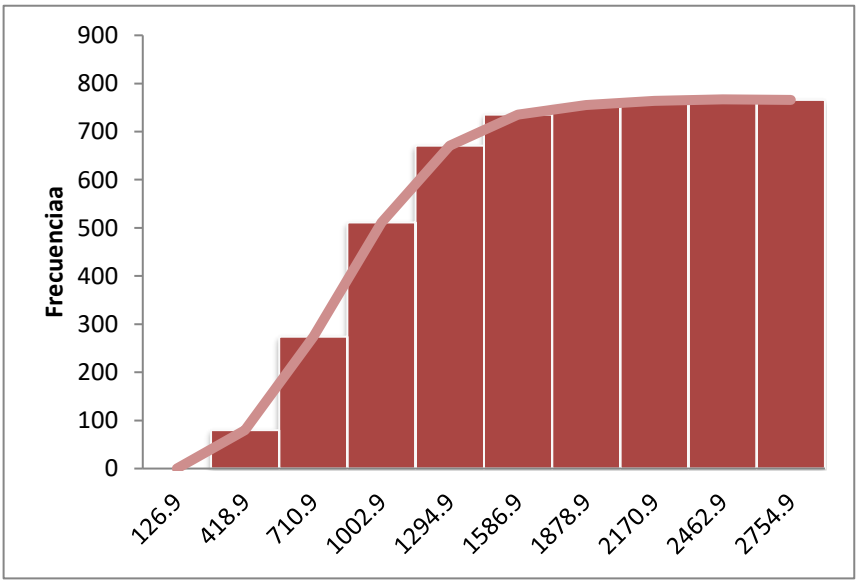
PRODUCCIÓN REGISTRADA MENSUAL DEL CAFE

Este muestra las estadísticas de los miles de sacos que contienen 60 Kg de café verde equivalente durante los últimos 64 años en Colombia.

Columna1	
Media	875,591495
Error típico	13,7358744
Mediana	821,5
Moda	644
Desviación estándar	382,637184
Varianza de la muestra	146411,214
Curtosis	1,25212465
Coeficiente de asimetría	0,85832437
Rango	2628
Mínimo	127
Máximo	2755
Suma	679459
Cuenta	776

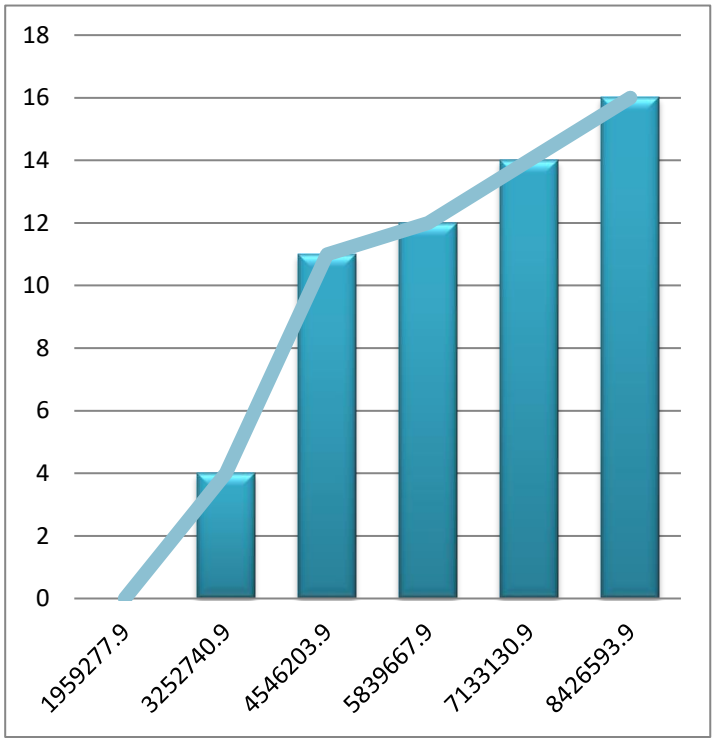
Intervalos		**Grupos**	**Frecuencia**
Li	**Ls**		
	127	126,9	0
127	419	418,9	80
419	711	710,9	274
711	1003	1002,9	511
1003	1295	1294,9	671
1295	1587	1586,9	735
1587	1879	1878,9	755
1879	2171	2170,9	764
2171	2463	2462,9	767
2463	2755	2754,9	766

Numero de clase	9,53654368
Tamaño de clase	292



VALOR DE LA COSECHA REGISTRADA –ANUAL

Evidenciamos los millones de pesos registrados por cada cosecha durante el año cafetero con relación a los últimos 19 años de cosechas.



Intervalos		Grupos	Frecuencia
Li	Ls		
	1.959.278	1959277,9	0
1.959.278	3.252.741	3252740,9	4
3.252.741	4.546.204	4546203,9	11
4.546.204	5.839.668	5839667,9	12
5.839.668	7.133.131	7133130,9	14
7.133.131	8.426.594	8426593,9	16

Numero de clase	4,3
Tamaño de clase	1.293.463

Columna1	
Media	4248304,848
Error típico	399696,1017
Mediana	3712705,211
Moda	#N/A
Desviación estándar	1787495,307
Varianza de la muestra	3,19514E+12
Curtosis	-0,861513046
Coeficiente de asimetría	0,586969758
Rango	5553353,68
Mínimo	1959278
Máximo	7512631,68
Suma	84966096,97
Cuenta	20

ANALISIS DATRAFRAME CAFÉ EN COLOMBIA

In [91]:
```
import pandas as  pd
```

In [92]:
```
# recuerda que se importa la libreria pandas y se le llama pd
```

In [93]:
```
pd.read_csv("PRODUCCIOn.csv")
```

Out [93]:

	Anio	Departamento	Producto	Área (ha)	Producción (Ton)	Rendimiento (Ha/ton)	Producción Nacional (ton)	Área Nacional (ha)
0	2007	ANTIOQUIA	CAFE	112,343.60	120,500.80	1.07	14.54	14.66
1	2007	BOLIVAR	CAFE	502.00	446.00	0.89	0.05	0.07
2	2007	BOYACA	CAFE	11,374.50	9,683.10	0.85	1.17	1.48
3	2007	CALDAS	CAFE	78,393.65	92,815.00	1.18	11.20	10.23
4	2007	CAQUETA	CAFE	2,295.00	2,134.00	0.93	0.26	0.30
...	...	...	...	...	...	...	...	...
261	2018	QUINDIO	CAFE	16,374.73	17,739.03	1.08	2.07	2.21
262	2018	RISARALDA	CAFE	35,874.73	45,918.75	1.28	5.37	4.83
263	2018	SANTANDER	CAFE	42,269.07	55,918.71	1.32	6.53	5.69
264	2018	TOLIMA	CAFE	97,304.04	97,451.31	1.00	11.39	13.11
265	2018	VALLE DEL CAUCA	CAFE	48,305.31	49,667.88	1.03	5.80	6.51

266 rows × 8 columns

In [94]:
```
Produccion_df=pd.read_csv("PRODUCCION.csv")
```

In [95]:
```
Produccion_df
```

Out [95]:

	Anio	Departamento	Producto	Area (ha)	Producción (ton)	Rendimiento (Ha/ton)	Producción Nacional (ton)	Área Nacional (ha)
0	2007	ANTIOQUIA	CAFE	112,343.60	120,500.80	1.07	14.54	14.66
1	2007	BOLIVAR	CAFE	502.00	446.00	0.89	0.05	0.07
2	2007	BOYACA	CAFE	11,374.50	9,683.10	0.85	1.17	1.48
3	2007	CALDAS	CAFE	78,393.65	92,815.00	1.18	11.20	10.23
4	2007	CAQUETA	CAFE	2,295.00	2,134.00	0.93	0.26	0.30
...	...	...	...	...	...	...	...	...
261	2018	QUINDIO	CAFE	16,374.73	17,739.03	1.08	2.07	2.21
262	2018	RISARALDA	CAFE	35,874.73	45,918.75	1.28	5.37	4.83
263	2018	SANTANDER	CAFE	42,269.07	55,918.71	1.32	6.53	5.69
264	2018	TOLIMA	CAFE	97,304.04	97,451.31	1.00	11.39	13.11
265	2018	VALLE DEL CAUCA	CAFE	48,305.31	49,667.88	1.03	5.80	6.51

266 rows × 8 columns

In [96]:
```
type(Produccion_df)
```

Out[96]: pandas.core.frame.DataFrame

In [97]:
```
Produccion_df.dtypes
```

Out[97]:
```
Anio                          int64
Departamento                 object
Producto                     object
Area (ha)                    object
Produccion (ton)             object
Rendimiento (ha/ton)        float64
Produccion Nacional (ton)   float64
Area Nacional (ha)          float64
dtype: object
```

In [98]:
```
Produccion_df.info()
```

```
<class 'pandas.core.frame.DataFrame'> RangeIndex: 266
entries, 0 to 265
Data columns (total 8 columns):
#                 Column                      Non-Null  Count   Dtype
---                                           ----------------------  -----------
0       Anio                        266 non-null    int64
1       Departamento                266 non-null    object
2       Producto                    266 non-null    object
3       Area  (ha)                  266 non-null    object
4       Produccion  (ton)           266 non-null    object
5       Rendimiento  (ha/ton)       266 non-null    float64
6       Produccion Nacional  (ton)  266 non-null    float64
7       Area Nacional  (ha)         266 non-null    float64
dtypes:  float64(3),  int64(1),  object(4) memory usage:
16.8+ KB
```

In [99]:
```
pd.unique(Produccion_df['Anio'])
```

Out[99]: array([2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017,
2018], dtype=int64)

In [100]:
```
pd.unique(Produccion_df['Departamento'])
```

Out[100]: array(['ANTIOQUIA', 'BOLIVAR', 'BOYACA', 'CALDAS', 'CAQUETA', 'CASANARE', 'CAUCA',
'CESAR', 'CHOCO', 'CUNDINAMARCA', 'HUILA', 'LA GUAJIRA', 'MAGDALENA', 'META', 'NARIÑO',
'NORTE DE SANTANDER', 'PUTUMAYO',
'QUINDIO', 'RISARALDA', 'SANTANDER', 'TOLIMA', 'VALLE DEL CAUCA',
'ARAUCA', 'GUAVIARE'], dtype=object)

In [101]:
```
pd.unique(Produccion_df['Producto'])
```

Out[101]: array(['CAFE'], dtype=object)

In [102]:
```
pd.unique(Produccion_df['Area (ha)'])
```

Out[102]: array(['112,343.60', '502.00', '11,374.50', '78,393.65', '2,295.00',
'2,605.00', '53,471.00', '23,172.00', '290.00', '43,017.30',
'89,661.56', '4,785.00', '17,506.00', '2,048.00', '24,458.50',
'30,171.84', '35.00', '19,904.00', '47,689.25', '34,406.67',
'91,679.10', '76,667.80', '114,694.00', '572.00', '10,778.50',
'74,897.00', '2,735.00', '2,149.00', '56,208.00', '23,198.00',
'90.00', '43,633.35', '89,131.20', '4,553.00', '17,521.00',
'2,146.00', '25,582.00', '31.00', '19,571.00', '47,227.00',
'34,169.37', '86,829.20', '72,419.00', '112,420.20', '770.00',
'10,672.50', '73,083.00', '2,332.00', '1,904.00', '57,860.00',
'23,420.00', '70.00', '43,475.84', '86,726.78', '4,488.00',
'17,036.00', '2,216.00', '26,467.20', '33,552.58', '23.00',
'19,052.00', '45,428.00', '37,985.90', '88,667.00', '67,001.30',
'111,602.71', '0.00', '850.00', '9,427.00', '72,240.58',
'2,536.00', '2,198.00', '55,162.00', '22,489.50', '157.50',
'44,264.16', '87,139.53', '4,207.00', '17,000.00', '2,326.00',
'23,504.05', '30,731.96', '24.00', '18,159.00', '47,308.00',
'39,000.64', '84,658.70', '69,332.10', '106,419.57', '10.00',
'8,441.74', '66,331.61', '2,810.00', '2,081.50', '54,246.42',
'22,350.00', '37,478.87', '78,792.21', '4,100.00', '16,577.00',
'2,578.00', '24,263.80', '21,520.45', '40.00', '20,139.30',
'44,733.64', '37,282.04', '93,145.35', '68,038.40', '112,221.14',
'870.00', '6,698.20', '54,871.88', '2,882.50', '2,322.00',
'56,825.00', '22,911.00', '37,175.06', '79,809.34', '5,143.00',
'17,686.00', '2,783.00', '27,806.40', '19,339.31', '42.00',
'21,109.83', '45,588.03', '33,947.15', '90,904.48', '69,456.71',
'109,755.50', '659.04', '9,289.05', '60,264.29', '2,905.84',
'2,232.94', '74,105.64', '25,106.39', '125.01', '36,189.18',
'118,200.88', '5,750.70', '17,016.72', '2,483.43', '32,136.51',
'25,332.45', '24.27', '21,203.03', '39,615.60', '38,613.68',
'97,308.81', '53,481.02', '110,115.86', '936.34', '9,834.39',
'59,757.18', '3,074.92', '2,599.43', '77,068.46', '26,138.58',
'136.88', '33,623.54', '128,273.15', '6,078.64', '18,533.11',
'2,739.71', '33,608.32', '23,724.20', '101.16', '21,462.81',
'40,154.46', '40,733.20', '100,832.91', '56,035.94', '109,649.61',
'1,065.07', '10,461.85', '58,376.40', '3,410.56', '2,752.31',
'77,405.83', '25,948.50', '137.47', '34,101.49', '130,452.40',
'5,631.53', '17,996.31', '2,922.21', '33,490.93', '22,940.64',
'128.65', '21,491.21', '41,732.03', '42,679.11', '103,368.73',
'54,938.79', '105,666.60', '1,065.97', '10,181.80', '56,022.04',
'3,392.22', '2,671.04', '78,421.95', '25,530.59', '134.96',
'33,214.17', '126,052.15', '5,531.20', '17,745.80', '2,924.89',
'32,750.16', '21,520.64', '20,041.70', '40,472.26', '41,387.79',
'100,328.77', '52,648.25', '99,311.53', '1,137.42', '9,598.33',
'51,854.59', '3,408.69', '2,436.63', '80,289.56', '25,158.80',
'125.67', '30,894.16', '122,575.76', '5,340.80', '18,129.50',
'2,926.85', '33,639.55', '21,409.77', '209.29', '17,699.67',
'37,334.16', '42,327.26', '96,018.89', '51,470.86', '98,038.15',
'1,182.13', '9,653.45', '50,762.22', '3,485.24', '2,360.55',
'82,085.54', '23,915.45', '140.33', '29,085.24', '122,002.46',
'4,810.97', '17,414.32', '2,761.01', '33,465.54', '20,873.04',
'209.93', '16,374.73', '35,874.73', '42,269.07', '97,304.04',
'48,305.31'], dtype=object)

In [103]:
```
pd.unique(Produccion_df['Produccion (ton)'])
```

Out[103]:
```
array(['120,500.80', '446.00', '9,683.10', '92,815.00',  '2,134.00',
       '2,048.40',  '51,348.00',  '13,278.50',  '205.90',  '33,729.14',
'129,052.51', '2,958.70', '14,005.00', '1,617.20', '31,770.05',
'13,593.24', '34.00', '25,426.00', '72,842.55', '29,469.52',
'112,322.38', '69,618.24', '113,505.20', '711.00',  '9,547.30',
'86,884.00',  '2,469.00',  '1,388.13',  '48,073.00',  '13,841.45',
'68.00', '78,254.77', '131,316.47', '2,328.90', '14,017.00',
'1,656.96', '31,262.50', '13,593.25', '35.60', '23,669.00',
'60,079.00', '29,016.75', '101,201.88', '65,666.43', '103,703.00',
'292.60', '8,567.97', '81,668.22', '2,332.00', '2,079.70',
'47,221.00', '12,770.00', '78.75', '37,118.07', '104,609.42',
'2,340.40', '13,412.80', '1,672.60', '27,487.71', '10,221.69',
'26.70', '21,985.00', '53,648.00', '26,311.61', '88,633.10',
'62,711.08', '121,253.38', '0.00', '510.00', '7,083.07',
'95,957.90', '2,902.50', '2,564.86', '45,113.00', '13,276.08',
'98.00', '37,214.80', '104,336.56', '2,393.00', '13,600.00',
'2,221.90', '24,594.10', '22,111.65', '21,065.00', '72,091.00',
'27,094.16', '94,230.20', '69,496.65', '115,267.98',  '12.00',
'5,643.39',  '78,805.87',  '2,528.40',  '2,023.50', '41,645.39',
'11,035.85', '32,780.35', '85,150.66', '1,933.00', '13,301.60',
'2,533.75', '24,073.95', '12,332.00', '45.80', '20,814.11',
'49,042.31', '22,089.82', '53,288.42', '65,475.63',  '91,621.30',
'652.50', '4,981.59', '54,115.96', '2,446.38', '1,718.25',
'50,588.14', '19,994.35', '140.00', '30,786.41', '85,212.64',
'3,434.30', '14,096.05', '2,133.10', '28,077.94', '12,214.54',
'48.40', '18,030.13', '36,989.43', '23,271.89', '85,027.49',
'61,190.55', '102,403.24', '395.07', '5,591.05', '58,634.19',
'2,188.92', '1,338.56', '56,303.92', '15,050.27', '105.93',
'24,993.74', '115,874.98', '3,447.31', '10,200.84', '1,650.41',
'28,606.96', '15,185.79', '16.87', '20,599.27', '39,073.92',
'30,227.02', '77,215.36', '42,948.40', '111,452.91',  '606.93',
'6,364.41',  '62,869.38',  '2,503.81',  '1,688.60', '63,365.76',
'16,935.63', '125.42', '25,118.55', '135,971.20',  '3,923.80',
'12,012.98', '1,950.84', '32,321.56', '15,108.55', '76.04',
'22,518.42', '42,719.53', '34,512.79', '86,453.62',  '49,799.28',
'120,365.77', '1,089.74', '9,501.54', '67,231.37', '3,749.27',
'2,626.73', '83,626.44', '22,240.81', '158.20', '31,165.15',
'145,168.10', '4,317.50', '16,691.31', '3,206.35', '36,607.56',
'20,267.64', '124.67', '24,694.56', '47,215.69', '47,304.16',
'105,563.88', '57,583.56', '119,970.68', '1,128.32',  '9,583.80',
'66,661.14', '3,861.63', '2,638.88', '87,642.49', '22,649.03',
'160.62', '31,413.34', '145,154.42', '4,387.19', '17,031.09',
'3,322.42', '37,020.90', '19,590.10', '23,791.30',  '47,357.02',
'47,512.36', '105,976.19', '57,067.08', '140,398.62',  '748.97',
'7,638.99',  '68,668.20',  '5,108.33',  '1,747.51', '97,922.49',
'16,628.14', '158.85', '33,943.39', '133,787.95',  '3,516.80',
'11,937.90', '4,013.11', '35,004.18', '23,409.44', '282.18',
'18,792.05', '46,779.71', '54,908.68', '94,556.71',  '51,687.80',
'141,898.91', '734.91', '7,780.34', '68,670.96', '5,280.40',
'1,629.25', '102,147.00', '14,943.62', '181.42', '32,580.24',
'136,161.86', '2,990.91', '10,826.24', '3,877.62', '35,679.42',
'23,471.69', '289.50', '17,739.03', '45,918.75', '55,918.71',
'97,451.31', '49,667.88'], dtype=object)
```

In [104]:
```
pd.unique(Produccion_df['Rendimiento (ha/ton)'])
```

Out[104]:
```
array([1.07, 0.89, 0.85, 1.18, 0.93, 0.79, 0.96, 0.57, 0.71, 0.78, 1.44,
       0.62, 0.8 , 1.3 , 0.45, 0.97, 1.28, 1.53, 0.86, 1.23, 0.91, 0.99,
       1.24, 1.16, 0.9 , 0.65, 0.6 , 0.76, 1.79, 1.47, 0.51, 0.77, 1.22,
       1.15, 1.21, 1.27, 1.17, 0.92, 0.38, 1.12, 1.  , 1.09, 0.82, 0.55,
       1.13, 0.52, 0.75, 1.04, 0.3 , 0.69, 0.94, 0.  , 1.33, 1.14, 0.59,
       0.84, 1.2 , 1.05, 0.72, 1.11, 1.52, 1.08, 0.67, 1.19, 0.49, 0.87,
       0.47, 0.98, 1.03, 1.1 , 0.74, 2.  , 0.83, 1.01, 0.63, 0.81, 0.88,
       0.66, 0.7 , 1.06, 0.64, 1.02, 0.95, 1.41, 1.32, 1.5 , 1.26, 1.37,
       1.35, 1.25, 1.45, 1.29, 1.4 , 1.38])
```

In [105]:
```
pd.unique(Produccion_df['Produccion Nacional (ton)'])
```

Out[105]:
```
array([1.454e+01, 5.000e-02, 1.170e+00, 1.120e+01, 2.600e-01, 2.500e-01,
       6.190e+00, 1.600e+00, 2.000e-02, 4.070e+00, 1.557e+01, 3.600e-01,
       1.690e+00, 2.000e-01, 3.830e+00, 1.640e+00, 0.000e+00, 3.070e+00,
       8.790e+00, 3.560e+00, 1.355e+01, 8.400e+00, 1.370e+01, 9.000e-02,
       1.150e+00, 1.049e+01, 3.000e-01, 1.700e-01, 5.800e+00, 1.670e+00,
       1.000e-02, 9.440e+00, 1.585e+01, 2.800e-01, 3.770e+00, 2.860e+00,
       7.250e+00, 3.500e+00, 1.221e+01, 7.930e+00, 1.463e+01, 4.000e-02,
       1.210e+00, 1.152e+01, 3.300e-01, 2.900e-01, 6.660e+00, 1.800e+00,
       5.240e+00, 1.476e+01, 1.890e+00, 2.400e-01, 3.880e+00, 1.440e+00,
       3.100e+00, 7.570e+00, 3.710e+00, 1.250e+01, 8.850e+00, 1.556e+01,
       7.000e-02, 9.100e-01, 1.231e+01, 3.700e-01, 5.790e+00, 1.700e+00,
       4.780e+00, 1.339e+01, 3.100e-01, 1.750e+00, 3.160e+00, 2.840e+00,
       2.700e+00, 9.250e+00, 3.480e+00, 1.209e+01, 8.920e+00, 1.800e+01,
       8.000e-02, 8.800e-01, 3.900e-01, 3.200e-01, 6.500e+00, 1.720e+00,
       5.120e+00, 1.330e+01, 2.080e+00, 4.000e-01, 3.760e+00, 1.930e+00,
       3.250e+00, 7.660e+00, 3.450e+00, 8.320e+00, 1.022e+01, 1.462e+01,
       1.000e-01, 7.900e-01, 8.630e+00, 2.700e-01, 8.070e+00, 3.190e+00,
       4.910e+00, 1.360e+01, 5.500e-01, 2.250e+00, 3.400e-01, 4.480e+00,
       1.950e+00, 2.880e+00, 5.900e+00, 1.357e+01, 9.760e+00, 1.570e+01,
       6.000e-02, 8.600e-01, 8.990e+00, 2.100e-01, 2.310e+00, 1.777e+01,
       5.300e-01, 1.560e+00, 4.390e+00, 2.330e+00, 5.990e+00, 4.640e+00,
       1.184e+01, 6.590e+00, 1.530e+01, 8.700e-01, 2.300e-01, 8.700e+00,
       1.867e+01, 5.400e-01, 1.650e+00, 4.440e+00, 2.070e+00, 3.090e+00,
       5.860e+00, 4.740e+00, 1.187e+01, 6.840e+00, 1.415e+01, 1.300e-01,
       1.120e+00, 7.900e+00, 4.400e-01, 9.830e+00, 2.620e+00, 3.660e+00,
       1.707e+01, 5.100e-01, 1.960e+00, 3.800e-01, 4.300e+00, 2.380e+00,
       2.900e+00, 5.550e+00, 5.560e+00, 1.241e+01, 6.770e+00, 1.405e+01,
       7.810e+00, 4.500e-01, 1.026e+01, 2.650e+00, 3.680e+00, 1.700e+01,
       1.990e+00, 4.340e+00, 2.290e+00, 2.790e+00, 6.680e+00, 1.649e+01,
       9.000e-01, 8.060e+00, 6.000e-01, 1.150e+01, 3.990e+00, 1.571e+01,
       4.100e-01, 1.400e+00, 4.700e-01, 4.110e+00, 2.750e+00, 3.000e-02,
       2.210e+00, 5.490e+00, 6.450e+00, 1.110e+01, 6.070e+00, 1.658e+01,
       8.020e+00, 6.200e-01, 1.900e-01, 1.194e+01, 3.810e+00, 1.591e+01,
       3.500e-01, 1.260e+00, 4.170e+00, 2.740e+00, 5.370e+00, 6.530e+00,
       1.139e+01])
```

In [106]:
```
pd.unique(Produccion_df['Area Nacional (ha)'])
```

Out[106]:
```
array([1.466e+01, 7.000e-02, 1.480e+00, 1.023e+01, 3.000e-01, 3.400e-01,
       6.980e+00, 3.020e+00, 4.000e-02, 5.610e+00, 1.170e+01, 6.200e-01,
       2.280e+00, 2.700e-01, 3.190e+00, 3.940e+00, 0.000e+00, 2.600e+00,
       6.220e+00, 4.490e+00, 1.196e+01, 1.000e+01, 1.513e+01, 8.000e-02,
       1.420e+00, 9.880e+00, 3.600e-01, 2.800e-01, 7.410e+00, 3.060e+00,
       1.000e-02, 5.750e+00, 1.175e+01, 6.000e-01, 2.310e+00, 3.370e+00,
       3.980e+00, 2.580e+00, 6.230e+00, 4.510e+00, 1.145e+01, 9.550e+00,
       1.490e+01, 1.000e-01, 1.410e+00, 9.680e+00, 3.100e-01, 2.500e-01,
       7.670e+00, 3.100e+00, 5.760e+00, 1.149e+01, 5.900e-01, 2.260e+00,
       2.900e-01, 3.510e+00, 4.450e+00, 2.520e+00, 6.020e+00, 5.030e+00,
       8.880e+00, 1.499e+01, 1.100e-01, 1.270e+00, 9.710e+00, 2.000e-02,
       5.950e+00, 1.171e+01, 5.700e-01, 3.160e+00, 4.130e+00, 2.440e+00,
       6.360e+00, 5.240e+00, 1.137e+01, 9.310e+00, 1.494e+01, 1.200e-01,
       1.180e+00, 3.900e-01, 7.610e+00, 3.140e+00, 5.260e+00, 1.106e+01,
       5.800e-01, 2.330e+00, 3.410e+00, 2.830e+00, 6.280e+00, 5.230e+00,
       1.308e+01, 1.580e+01, 9.400e-01, 7.720e+00, 4.100e-01, 3.300e-01,
       8.000e+00, 3.220e+00, 1.123e+01, 7.200e-01, 2.490e+00, 3.910e+00,
       2.720e+00, 2.970e+00, 6.420e+00, 4.780e+00, 1.280e+01, 9.780e+00,
       1.422e+01, 9.000e-02, 1.200e+00, 7.810e+00, 3.800e-01, 9.600e+00,
       3.250e+00, 4.690e+00, 1.531e+01, 7.500e-01, 2.200e+00, 3.200e-01,
       4.160e+00, 3.280e+00, 2.750e+00, 5.130e+00, 5.000e+00, 1.261e+01,
       6.930e+00, 1.384e+01, 1.240e+00, 7.510e+00, 9.690e+00, 3.290e+00,
       4.230e+00, 1.612e+01, 7.600e-01, 4.220e+00, 2.980e+00, 2.700e+00,
       5.050e+00, 5.120e+00, 1.267e+01, 7.040e+00, 1.369e+01, 1.300e-01,
       1.310e+00, 7.290e+00, 4.300e-01, 9.660e+00, 3.240e+00, 4.260e+00,
       1.628e+01, 7.000e-01, 2.250e+00, 4.180e+00, 2.860e+00, 2.680e+00,
       5.210e+00, 5.330e+00, 1.290e+01, 6.860e+00, 1.359e+01, 1.400e-01,
       7.200e+00, 4.400e-01, 1.008e+01, 4.270e+00, 1.621e+01, 7.100e-01,
       4.210e+00, 2.770e+00, 5.200e+00, 5.320e+00, 6.770e+00, 1.318e+01,
       1.500e-01, 6.880e+00, 4.500e-01, 1.066e+01, 3.340e+00, 4.100e+00,
       1.627e+01, 2.410e+00, 4.470e+00, 2.840e+00, 3.000e-02, 2.350e+00,
       4.960e+00, 5.620e+00, 1.275e+01, 6.830e+00, 1.321e+01, 1.600e-01,
       1.300e+00, 6.840e+00, 4.700e-01, 3.920e+00, 1.643e+01, 6.500e-01,
       3.700e-01, 2.810e+00, 2.210e+00, 4.830e+00, 5.690e+00, 1.311e+01,
       6.510e+00])
```

In [107]:
```
Produccion_df['Anio'].min()
```

Out[107]: 2007

In [108]:
```
Produccion_df['Anio'].max()
```

Out[108]: 2018

In [109]:
```
Produccion_df['Area (ha)'].min()+" Hectarea"
```

Out[109]: '0.00 Hectarea'

In [110]:
```
Produccion_df['Area (ha)'].max()+" Hectarea"
```

Out[110]: '99,311.53 Hectarea'

In [111]:
```
Produccion_df['Rendimiento (ha/ton)'].min()
```

Out[111]: 0.0

In [112]:
```
Produccion_df['Rendimiento (ha/ton)'].max()
```

Out[112]: 2.0

In [113]:
```
Produccion_df['Anio'].isnull()
```

Out[113]:
```
0      False
1      False
2      False
3      False
4      False
       ...
261    False
262    False
263    False
264    False
265    False
Name: Anio, Length: 266, dtype: bool
```

In [114]:
```
Produccion_df['Area (ha)'].isnull()
```

Out[114]:
```
0      False
1      False
2      False
3      False
4      False
       ...
261    False
262    False
263    False
264    False
265    False
Name: Area (ha), Length: 266, dtype: bool
```

In [115]:
```
Produccion_df['Rendimiento (ha/ton)'].isnull()
```

Out[115]:
```
0      False
1      False
2      False
3      False
4      False
       ...
261    False
262    False
263    False
264    False
265    False
Name: Rendimiento (ha/ton), Length: 266, dtype: bool
```

In [116]:
```
Produccion_df['Rendimiento (ha/ton)'].isnull().sum()
```

Out[116]: 0

In [117]:
```
Produccion_df['Area (ha)'].isnull().sum()
```

Out[117]: 0

In [118]:
```
Produccion_grouped_Anio=Produccion_df.groupby("Anio").sum()

Produccion_grouped_Anio
```

Out[118]:

Anio	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
2007	20.91	100.01	100.00
2008	21.62	100.00	99.99
2009	19.39	100.00	99.98
2010	20.84	100.01	100.00
2011	19.65	100.02	100.00
2012	19.75	99.99	100.00
2013	16.71	100.00	99.99
2014	18.09	100.00	100.00
2015	22.54	99.98	100.00
2016	22.34	99.99	100.00
2017	23.50	100.01	100.00
2018	23.75	100.00	100.02

In [119]:
```
Produccion_grouped_Anio_Rendimiento=Produccion_df.groupby("Rendimiento (ha/ton)"

Produccion_grouped_Anio_Rendimiento
```

Out[119]:

	Anio								Produccion Naci		
	count	mean	std	min	25%	50%	75%	max	count	mean	..
Rendimiento (ha/ton)											
0.00	2.0	2011.0	1.414214	2010.0	2010.50	2011.0	2011.50	2012.0	2.0	0.000	..
0.30	1.0	2009.0	NaN	2009.0	2009.00	2009.0	2009.00	2009.0	1.0	1.440	..
0.38	1.0	2009.0	NaN	2009.0	2009.00	2009.0	2009.00	2009.0	1.0	0.040	..
0.45	2.0	2007.5	0.707107	2007.0	2007.25	2007.5	2007.75	2008.0	2.0	1.640	..
0.47	1.0	2011.0	NaN	2011.0	2011.00	2011.0	2011.00	2011.0	1.0	0.300	..
...	...	...	...	...	...	...	...	...	...	...	..
1.50	1.0	2017.0	NaN	2017.0	2017.00	2017.0	2017.00	2017.0	1.0	0.600	..
1.52	2.0	2014.0	5.656854	2010.0	2012.00	2014.0	2016.00	2018.0	2.0	4.935	..
1.53	1.0	2007.0	NaN	2007.0	2007.00	2007.0	2007.00	2007.0	1.0	8.790	..
1.79	1.0	2008.0	NaN	2008.0	2008.00	2008.0	2008.00	2008.0	1.0	9.440	..
2.00	1.0	2012.0	NaN	2012.0	2012.00	2012.0	2012.00	2012.0	1.0	0.020	..

94 rows × 24 columns

In [120]:
```
Produccion_grouped_Departamento=Produccion_df.groupby(["Anio","Departamento"]).s
Produccion_grouped_Departamento
```

Out[120]:

Anio	Departamento	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
2007	**ANTIOQUIA**	1.07	14.54	14.66
	BOLIVAR	0.89	0.05	0.07
	BOYACA	0.85	1.17	1.48
	CALDAS	1.18	11.20	10.23
	CAQUETA	0.93	0.26	0.30
...	**...**	...	...	...
2018	**QUINDIO**	1.08	2.07	2.21
	RISARALDA	1.28	5.37	4.83
	SANTANDER	1.32	6.53	5.69
	TOLIMA	1.00	11.39	13.11
	VALLE DEL CAUCA	1.03	5.80	6.51

266 rows × 3 columns

In [121]:
```
Produccion_grouped_Departamento_Rendimiento=Produccion_df.groupby(["Anio","Depar
Produccion_grouped_Departamento_Rendimiento
```

Out[121]:

Anio	Departamento	Rendimiento (ha/ton)	Producto Area (ha)		Produccion (ton)	Produccion Nacional (ton)	Area Nacional (ha)
2007	**ANTIOQUIA**	**1.07**	CAFE	112,343.60	120,500.80	14.54	14.66
	BOLIVAR	**0.89**	CAFE	502.00	446.00	0.05	0.07
	BOYACA	**0.85**	CAFE	11,374.50	9,683.10	1.17	1.48
	CALDAS	**1.18**	CAFE	78,393.65	92,815.00	11.20	10.23
	CAQUETA	**0.93**	CAFE	2,295.00	2,134.00	0.26	0.30
...	**...**	**...**	...	...	...	...	...
2018	**QUINDIO**	**1.08**	CAFE	16,374.73	17,739.03	2.07	2.21
	RISARALDA	**1.28**	CAFE	35,874.73	45,918.75	5.37	4.83
	SANTANDER	**1.32**	CAFE	42,269.07	55,918.71	6.53	5.69
	TOLIMA	**1.00**	CAFE	97,304.04	97,451.31	11.39	13.11
	VALLE DEL CAUCA	**1.03**	CAFE	48,305.31	49,667.88	5.80	6.51

266 rows × 5 column

In [122]:
```
Produccion_df['Produccion (ton)'].count()
# cuenta el numero de registros en el dataframe para el campo de la Producción
```

Out[122]: 266

In [123]:
```
Produccion_df['Anio'].count()
# cuenta el numero de registros en el dataframe para el campo del año
```

Out[123]: 266

In [124]:
```
Produccion_grouped_Departamento=Produccion_df.groupby(["Anio","Departamento"]).s
Produccion_grouped_Departamento
# Resume los valores maximo de Anio y Departamento
```

Out[124]:
```
Rendimiento (ha/ton)      2.00
Produccion Nacional (ton) 18.67
Area Nacional (ha)        16.43
dtype: float64
```

In [125]:
```
Produccion_grouped_Departamento=Produccion_df.groupby(["Anio","Departamento"]).s
Produccion_grouped_Departamento
# Resume los valores minimos de Anio y Departamento
```

Out[125]:
```
Rendimiento (ha/ton)      0.0
Produccion Nacional (ton) 0.0
Area Nacional (ha)        0.0
dtype: float64
```

In [126]:
```
Produccion_df.groupby('Departamento')['Produccion Nacional (ton)'].sum()
# Agrupa los datos por Departamento y describe la suma de la Produccion Nacional
```

Out[126]: Departamento

ANTIOQUIA	183.32
ARAUCA	0.00
BOLIVAR	1.01
BOYACA	11.89
CALDAS	115.87
CAQUETA	4.83
CASANARE	3.09
CAUCA	99.87
CESAR	25.29
CHOCO	0.21
CUNDINAMARCA	55.98
GUAVIARE	0.00
HUILA	188.60
LA GUAJIRA	4.98
MAGDALENA	21.17
META	3.88
NARIÑO	48.63
NORTE DE SANTANDER	26.00
PUTUMAYO	0.10
QUINDIO	34.08
RISARALDA	80.23
SANTANDER	54.89
TOLIMA	143.26
VALLE DEL CAUCA	92.83

Name: Produccion Nacional (ton), dtype: float64

In [127]:
```
Produccion_grouped_Departamento=Produccion_df.groupby(["Anio","Departamento"]).d
Produccion_grouped_Departamento
```

Out[127]:

		Rendimiento (ha/ton)								Produccion Nacional (ton)			
		count	mean	std	min	25%	50%	75%	max	count	mean	...	75%
Anio	Departamento												
2007	ANTIOQUIA	1.0	1.07	NaN	1.07	1.07	1.07	1.07	1.07	1.0	14.54	...	14.54
	BOLIVAR	1.0	0.89	NaN	0.89	0.89	0.89	0.89	0.89	1.0	0.05	...	0.05
	BOYACA	1.0	0.85	NaN	0.85	0.85	0.85	0.85	0.85	1.0	1.17	...	1.17
	CALDAS	1.0	1.18	NaN	1.18	1.18	1.18	1.18	1.18	1.0	11.20	...	11.20
	CAQUETA	1.0	0.93	NaN	0.93	0.93	0.93	0.93	0.93	1.0	0.26	...	0.26
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2018	QUINDIO	1.0	1.08	NaN	1.08	1.08	1.08	1.08	1.08	1.0	2.07	...	2.07
	RISARALDA	1.0	1.28	NaN	1.28	1.28	1.28	1.28	1.28	1.0	5.37	...	5.37
	SANTANDER	1.0	1.32	NaN	1.32	1.32	1.32	1.32	1.32	1.0	6.53	...	6.53
	TOLIMA	1.0	1.00	NaN	1.00	1.00	1.00	1.00	1.00	1.0	11.39	...	11.39
	VALLE DEL CAUCA	1.0	1.03	NaN	1.03	1.03	1.03	1.03	1.03	1.0	5.80	...	5.80

In [128]:
```
Produccion_df.dropna()
# Elimina los valores faltantes o NaN de cada columna
```

Out[128]:

	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
0	2007	ANTIOQUIA	CAFE	112,343.60	120,500.80	1.07	14.54	14.66
1	2007	BOLIVAR	CAFE	502.00	446.00	0.89	0.05	0.07
2	2007	BOYACA	CAFE	11,374.50	9,683.10	0.85	1.17	1.48
3	2007	CALDAS	CAFE	78,393.65	92,815.00	1.18	11.20	10.23
4	2007	CAQUETA	CAFE	2,295.00	2,134.00	0.93	0.26	0.30
...	...	...	...	...	...	...	...	...
261	2018	QUINDIO	CAFE	16,374.73	17,739.03	1.08	2.07	2.21
262	2018	RISARALDA	CAFE	35,874.73	45,918.75	1.28	5.37	4.83
263	2018	SANTANDER	CAFE	42,269.07	55,918.71	1.32	6.53	5.69
264	2018	TOLIMA	CAFE	97,304.04	97,451.31	1.00	11.39	13.11
265	2018	VALLE DEL CAUCA	CAFE	48,305.31	49,667.88	1.03	5.80	6.51

266 rows × 8 columns

In [129]:
```
Produccion_Departamento=Produccion_df.groupby(["Anio","Departamento","Area Nacio
Produccion_Departamento
# Da una amplia descripcion de los datos numericos de Anio, Departamento y Area
```

Out[129]:

			Rendimiento (ha/ton)								Produccion Nacion		
			count	mean	std	min	25%	50%	75%	max	count	mean	std
Anio	Departamento	Area Nacional (ha)											
2007	ANTIOQUIA	14.66	1.0	1.07	NaN	1.07	1.07	1.07	1.07	1.07	1.0	14.54	Na
	BOLIVAR	0.07	1.0	0.89	NaN	0.89	0.89	0.89	0.89	0.89	1.0	0.05	Na
	BOYACA	1.48	1.0	0.85	NaN	0.85	0.85	0.85	0.85	0.85	1.0	1.17	Na
	CALDAS	10.23	1.0	1.18	NaN	1.18	1.18	1.18	1.18	1.18	1.0	11.20	Na
	CAQUETA	0.30	1.0	0.93	NaN	0.93	0.93	0.93	0.93	0.93	1.0	0.26	Na
...	...	...	...	...	...	...	...	...	...	...	...	...	.
2018	QUINDIO	2.21	1.0	1.08	NaN	1.08	1.08	1.08	1.08	1.08	1.0	2.07	Na
	RISARALDA	4.83	1.0	1.28	NaN	1.28	1.28	1.28	1.28	1.28	1.0	5.37	Na
	SANTANDER	5.69	1.0	1.32	NaN	1.32	1.32	1.32	1.32	1.32	1.0	6.53	Na
	TOLIMA	13.11	1.0	1.00	NaN	1.00	1.00	1.00	1.00	1.00	1.0	11.39	Na
			1.0	1.03	NaN	1.03	1.03	1.03	1.03	1.03	1.0	5.80	Na

266 rows × 16 columns

In [130]:
```
Produccion_Anio=Produccion_df.groupby(["Anio"]).describe()
Produccion_Anio
# Da una amplia descripcion de los datos numericos de Anio
```

Out[130]:

	Rendimiento (ha/ton)								Produccion Nacional (ton)				
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75%	m
Anio													
2007	22.0	0.950455	0.279566	0.45	0.7900	0.900	1.1525	1.53	22.0	4.545909	...	7.8475	1
2008	22.0	0.982727	0.322670	0.45	0.7775	0.905	1.2000	1.79	22.0	4.545455	...	7.7600	1
2009	22.0	0.881364	0.264652	0.30	0.7600	0.930	1.1125	1.21	22.0	4.545455	...	7.3425	1
2010	23.0	0.906087	0.324692	0.00	0.7050	0.960	1.1250	1.52	23.0	4.348261	...	7.3550	1
2011	23.0	0.854348	0.238305	0.47	0.6100	0.900	1.0550	1.20	23.0	4.348696	...	7.0800	1
2012	23.0	0.858696	0.329618	0.00	0.7450	0.830	0.9150	2.00	23.0	4.347391	...	6.9850	1
2013	22.0	0.759545	0.145421	0.60	0.6000	0.755	0.8800	0.99	22.0	4.545455	...	6.4400	1
2014	22.0	0.822273	0.157629	0.64	0.6500	0.815	0.9500	1.06	22.0	4.545455	...	6.5950	1
2015	22.0	1.024545	0.110096	0.77	0.9350	1.065	1.1075	1.15	22.0	4.544545	...	6.4675	1
2016	21.0	1.063810	0.116725	0.79	0.9600	1.120	1.1500	1.19	21.0	4.761429	...	6.6800	1
2017	22.0	1.068182	0.272443	0.66	0.8450	1.090	1.2900	1.50	22.0	4.545909	...	6.3550	1
2018	22.0	1.079545	0.296672	0.62	0.8575	1.120	1.3125	1.52	22.0	4.545455	...	6.3475	1

12 rows × 24 columns

In [131]:
```
Produccion_Anio_Rendimiento=Produccion_df.groupby(["Rendimiento (ha/ton)"]).desc
Produccion_Anio_Rendimiento
# Da una amplia descripcion de los datos numericos de Rendimiento (ha/ton)
```

Out[131]:

	Anio								Produccion Nacion		
	count	mean		std	min	25%	50%	75%	max	count	mean	...
Rendimiento (ha/ton)												
0.00	2.0	2011.0	1.414214	2010.0	2010.50	2011.0	2011.50	2012.0	2.0	0.000	...
0.30	1.0	2009.0	NaN	2009.0	2009.00	2009.0	2009.00	2009.0	1.0	1.440	...
0.38	1.0	2009.0	NaN	2009.0	2009.00	2009.0	2009.00	2009.0	1.0	0.040	...
0.45	2.0	2007.5	0.707107	2007.0	2007.25	2007.5	2007.75	2008.0	2.0	1.640	...
0.47	1.0	2011.0	NaN	2011.0	2011.00	2011.0	2011.00	2011.0	1.0	0.300	...
...	...	...	...	...	...	...	...	...	...	...	...
1.50	1.0	2017.0	NaN	2017.0	2017.00	2017.0	2017.00	2017.0	1.0	0.600	...
1.52	2.0	2014.0	5.656854	2010.0	2012.00	2014.0	2016.00	2018.0	2.0	4.935	...
1.53	1.0	2007.0	NaN	2007.0	2007.00	2007.0	2007.00	2007.0	1.0	8.790	...
1.79	1.0	2008.0	NaN	2008.0	2008.00	2008.0	2008.00	2008.0	1.0	9.440	...
2.00	1.0	2012.0	NaN	2012.0	2012.00	2012.0	2012.00	2012.0	1.0	0.020	...

94 rows × 24 columns

In [132]:
```
Produccion_df.describe()
# Indica datos estadísticos generales del dataframe produccion
```

Out[132]:

	Anio	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
count	266.000000	266.000000	266.000000	266.000000
mean	2012.469925	0.936429	4.511316	4.511203
std	3.443484	0.267129	4.950568	4.565865
min	2007.000000	0.000000	0.000000	0.000000
25%	2010.000000	0.750000	0.352500	0.390000
50%	2012.000000	0.940000	2.720000	3.120000
75%	2015.000000	1.120000	7.147500	6.875000
max	2018.000000	2.000000	18.670000	16.430000

In [133]:
```
Produccion_df.describe()
Produccion_df.mean()
# Indica el promedio del dataframe produccion para Rendimiento, Produccion
```

Out[133]:
```
Anio                          2012.469925
Rendimiento (ha/ton)             0.936429
Produccion Nacional (ton)        4.511316
Area Nacional (ha)               4.511203
dtype: float64
```

In [136]:
```
Produccion_df["Produccion Nacional (ton)"].describe()
# Indica datos estadísticos generales para la Producción nacional del dataframe
```

Out[136]:
```
count    266.000000
mean       4.511316
std        4.950568
min        0.000000
25%        0.352500
50%        2.720000
75%        7.147500
max       18.670000
Name: Produccion Nacional (ton), dtype: float64
```

In [139]:
```
Produccion_counts=Produccion_df.groupby('Produccion (ton)')['Produccion (ton)'].
# creamos una grafica en barras indicando la cantidad de Embarked por Pclass
```

In [140]:
```
Produccion_df.duplicated().sum()
#Registros que esten duplicados
```

Out[140]: 0

In [141]:
```
Produccion_df.groupby('Rendimiento (ha/ton)')['Rendimiento (ha/ton)'].count()
# Agrupa los datos por Rendimiento (ha/ton) y describe la cantidad de cada uno
```

Out[141]:
```
Rendimiento (ha/ton)
0.00    2
0.30    1
0.38    1
0.45    2
0.47    1
       ..
1.50    1
1.52    2
1.53    1
1.79    1
2.00    1
Name: Rendimiento (ha/ton), Length: 94, dtype: int64
```

In [142]:
```
Produccion_df.groupby('Departamento')['Departamento'].count()['VALLE DEL CAUCA']
# aAgrupa los datos por Departamento y cuenta los Departamento que sean igual a
```

Out[142]: 12

In [143]:
```
Produccion_df.groupby('Produccion  (ton)')['Produccion (ton)'].count()['17,739.03
# Agrupa los datos por Produccion (ton) y cuenta losProduccion (ton) que sean ig
```

Out[143]: 1

In [144]:
```
Produccion_df['Area Nacional (ha)']*5
# Multiplica todos los valores de Area Nacional (ha) por cinco
```

Out[144]:
```
0      73.30
1       0.35
2       7.40
3      51.15
4       1.50
...
261    11.05
262    24.15
263    28.45
264    65.55
265    32.55
Name: Area Nacional (ha), Length: 266, dtype: float64
```

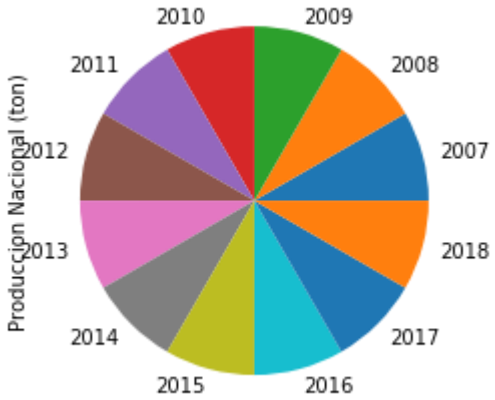
In [145]:
```
Produccion_df.groupby('Anio')['Rendimiento (ha/ton)'].sum()
# Agrupa los datos por Anio y describe la suma del Rendimiento (ha/ton) cada uno
```

Out[145]:
```
Anio
2007  20.91
2008  21.62
2009  19.39
2010  20.84
2011  19.65
2012  19.75
2013  16.71
2014  18.09
2015  22.54
2016  22.34
2017  23.50
2018  23.75
Name: Rendimiento (ha/ton), dtype: float64
```

In [146]:
```
Produccion_df.groupby('Anio')['Produccion  (ton)'].sum()
# Agrupa los datos por Anio y describe la suma de la Produccion (ton) cada uno
```

Out[146]:
```
Anio
2007       120,500.80446.009,683.1092,815.002,134.002,048...
2008       113,505.20711.009,547.3086,884.002,469.001,388...
2009       103,703.00292.608,567.9781,668.222,332.002,079...
2010       121,253.380.00510.007,083.0795,957.902,902.502...
2011       115,267.9812.00510.005,643.3978,805.872,528.40...
2012      91,621.30652.504,981.5954,115.962,446.381,718....
2013       102,403.24395.075,591.0558,634.192,188.921,338...
2014       111,452.91606.936,364.4162,869.382,503.811,688...
2015      120,365.771,089.749,501.5467,231.373,749.272,6...
2016      119,970.681,128.329,583.8066,661.143,861.632,6...
2017       140,398.62748.977,638.9968,668.205,108.331,747...
2018       141,898.91734.917,780.3468,670.965,280.401,629...
Name: Produccion (ton), dtype: object
```

In [147]:
```
total_count = Produccion_df.groupby('Anio')['Produccion Nacional (ton)'].sum()
# creamos una grafica en barras indicando la cantidad de Anio por Produccion Nac

total_count.plot(kind='pie');
# tambien graficuemos en tipo barra
```



In [148]:
```
Produccion_df['Rendimiento (ha/ton)']-0.49
# Disminuye todos los valores del Rendimiento (ha/ton) menos el 49%
```

Out[148]:
```
0      0.58
1      0.40
2      0.36
3      0.69
4      0.44
      ...
261    0.59
262    0.79
263    0.83
264    0.51
265    0.54
Name: Rendimiento (ha/ton), Length: 266, dtype: float64
```

In [149]:
```
Produccion_df['Rendimiento (ha/ton)']+0.63
# Aumenta todos los valores del Rendimiento (ha/ton) más el 63%
```

Out[149]:
```
0      1.70
1      1.52
2      1.48
3      1.81
4      1.56
      ...
261    1.71
262    1.91
263    1.95
264    1.63
265    1.66
Name: Rendimiento (ha/ton), Length: 266, dtype: float64
```

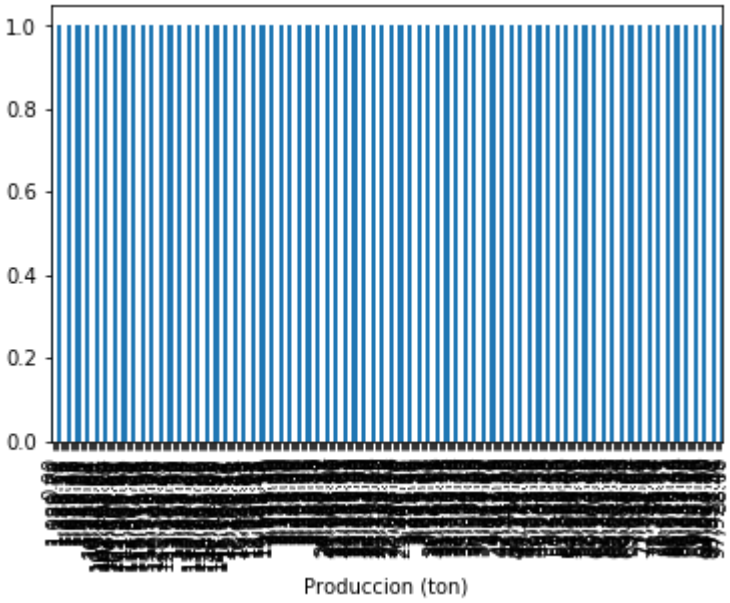
In [150]:
```
Produccion_df[Produccion_df.Departamento=='RISARALDA']
# selecciona de la fila Departamento los valores iguales a RISARALDA
```

Out[150]:

	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
18	2007	RISARALDA	CAFE	47,689.25	72,842.55	1.53	8.79	6.22
40	2008	RISARALDA	CAFE	47,227.00	60,079.00	1.27	7.25	6.23
62	2009	RISARALDA	CAFE	45,428.00	53,648.00	1.18	7.57	6.02
85	2010	RISARALDA	CAFE	47,308.00	72,091.00	1.52	9.25	6.36
108	2011	RISARALDA	CAFE	44,733.64	49,042.31	1.10	7.66	6.28
131	2012	RISARALDA	CAFE	45,588.03	36,989.43	0.81	5.90	6.42
153	2013	RISARALDA	CAFE	39,615.60	39,073.92	0.99	5.99	5.13
175	2014	RISARALDA	CAFE	40,154.46	42,719.53	1.06	5.86	5.05
197	2015	RISARALDA	CAFE	41,732.03	47,215.69	1.13	5.55	5.21
218	2016	RISARALDA	CAFE	40,472.26	47,357.02	1.17	5.55	5.20
240	2017	RISARALDA	CAFE	37,334.16	46,779.71	1.25	5.49	4.96
262	2018	RISARALDA	CAFE	35,874.73	45,918.75	1.28	5.37	4.83

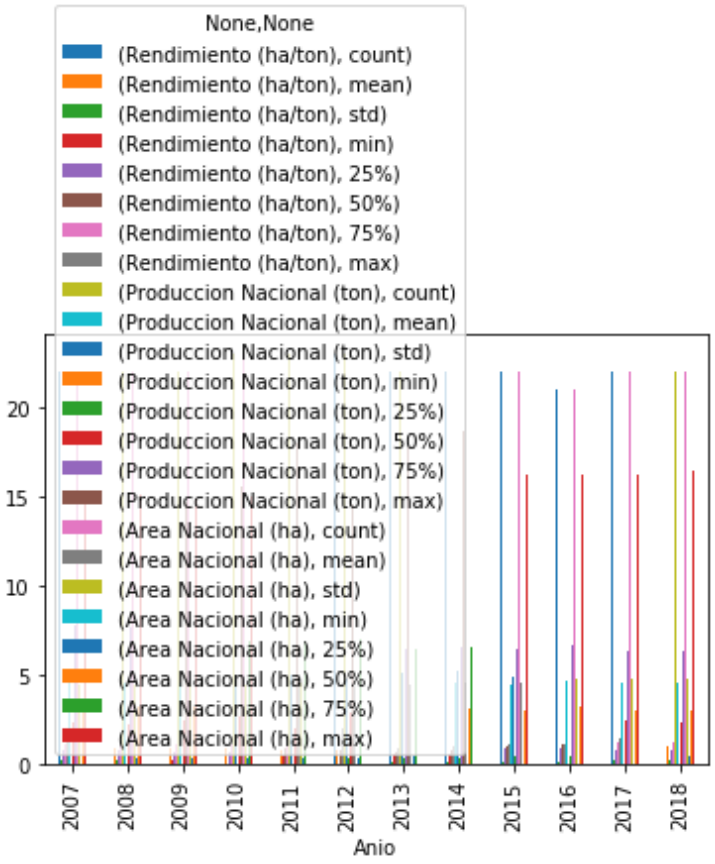
In [151]:
```
import numpy as np

import re

import sys
```

Out[151]: <matplotlib.axes._subplots.AxesSubplot at 0x94b4c08>



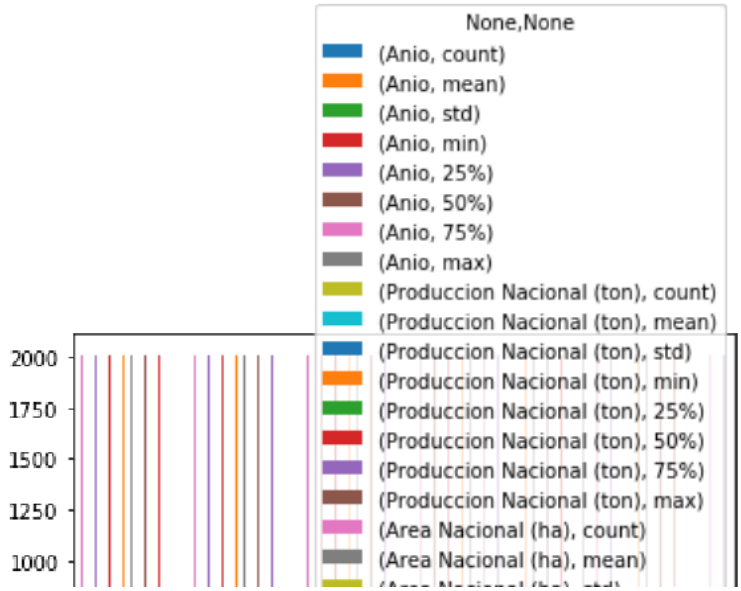
In [152]:
```
# Construcción del gráfico produccion por año tipo lineas

import numpy as np

import re
```

Out[152]: <matplotlib.axes._subplots.AxesSubplot at 0x9b94108>

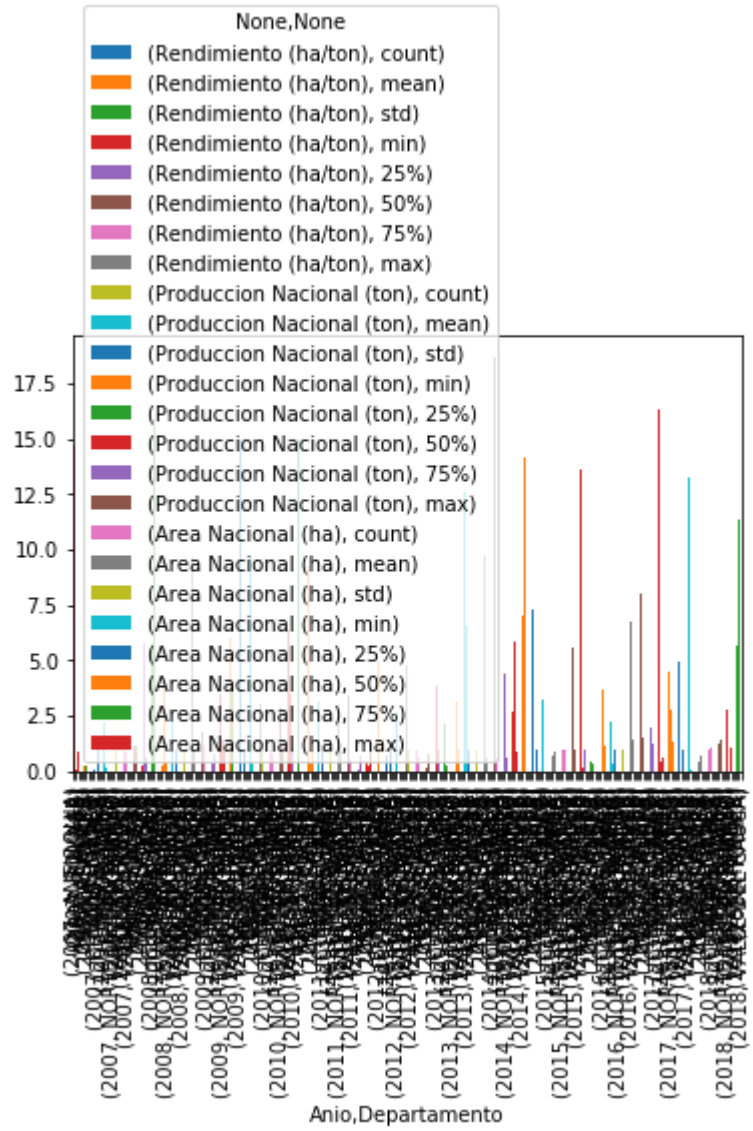


In [153]:
```
# Construcción del gráfico Rendimiento por año tipo lineas

%matplotlib inline
```

Out[153]: <matplotlib.axes._subplots.AxesSubplot at 0xbc031c8>



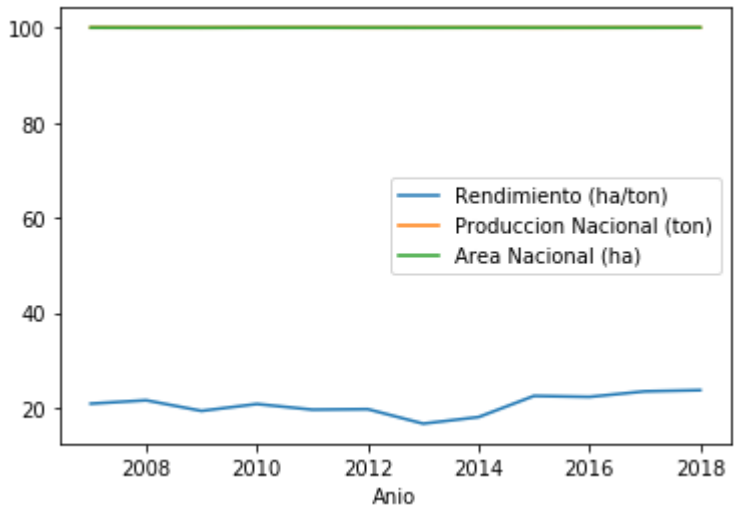
In [154]:
```
# Construcción del gráfico produccion por departamento año tipo lineas

%matplotlib inline
```



Out[154]: <matplotlib.axes._subplots.AxesSubplot at 0xd392508>

In [155]:
```
import numpy as np

import re

import sys
```

Out[155]: <matplotlib.axes._subplots.AxesSubplot at 0x1256dfc8>



In [161]:

```
grouped_data=Produccion_df.groupby("Departamento")
z=grouped_data.describe().mean()
print(z)
# Indica datos estadísticos generales del dataframe de la columna Departamento
```

```
Anio                          count    11.083333
                              mean   2012.382576
                              std       3.479313
                              min    2007.333333
                              25%    2009.854167
                              50%    2012.375000
                              75%    2014.895833
                              max    2017.458333
Rendimiento (ha/ton)          count    11.083333
                              mean      0.889467
                              std       0.216119
                              min       0.620833
                              25%       0.769167
                              50%       0.863750
                              75%       0.986771
                              max       1.235417
Produccion Nacional (ton)     count    11.083333
                              mean      4.166733
                              std       0.719931
                              min       3.261250
                              25%       3.687812
                              50%       4.031667
                              75%       4.614271
                              max       5.387500
Area Nacional (ha)            count    11.083333
                              mean      4.166632
                              std       0.511340
                              min       3.537500
                              25%       3.758229
                              50%       4.136042
                              75%       4.588854
                              max       4.838333
dtype: float64
```

In [163]:
```
departamentos_counts=Produccion_df.groupby("Departamento")["Producto"].count()
print(departamentos_counts)
# Verificar y cuenta cada uno de los Departamentos
```

```
Departamento
ANTIOQUIA                12
ARAUCA                    2
BOLIVAR                  12
BOYACA                   12
CALDAS                   12
CAQUETA                  12
CASANARE                 12
CAUCA                    12
CESAR                    12
CHOCO                    12
CUNDINAMARCA             12
GUAVIARE                  1
HUILA                    12
LA  GUAJIRA              12
MAGDALENA                12
META                     12
NARIÑO                   12
NORTE DE SANTANDER       12
PUTUMAYO                 11
QUINDIO                  12
RISARALDA                12
SANTANDER                12
TOLIMA                   12
VALLE DEL CAUCA          12
Name: Producto, dtype: int64
```

In [165]:
```
Grupos_Departamentos=Produccion_df.groupby("Anio")["Departamento"].count()
print(Grupos_Departamentos)
# Indica la cantidad de Departamentos
```

```
Anio
2007   22
2008   22
2009   22
2010   23
2011   23
2012   23
2013   22
2014   22
2015   22
2016   21
2017   22
2018   22
Name: Departamento, dtype: int64
```

In [167]:

```
Departamento_Meta=Produccion_df.loc[Produccion_df["Departamento"]=="META"]
print(Departamento_Meta)
# Indica los resultados estadísticos por año para el Departamento Meta
```

	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	\
13	2007	META	CAFE	2,048.00	1,617.20	
35	2008	META	CAFE	2,146.00	1,656.96	
57	2009	META	CAFE	2,216.00	1,672.60	
80	2010	META	CAFE	2,326.00	2,221.90	
103	2011	META	CAFE	2,578.00	2,533.75	
126	2012	META	CAFE	2,783.00	2,133.10	
148	2013	META	CAFE	2,483.43	1,650.41	
170	2014	META	CAFE	2,739.71	1,950.84	
192	2015	META	CAFE	2,922.21	3,206.35	
214	2016	META	CAFE	2,924.89	3,322.42	
235	2017	META	CAFE	2,926.85	4,013.11	
257	2018	META	CAFE	2,761.01	3,877.62	

	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
13	0.79	0.20	0.27
35	0.77	0.20	0.28
57	0.75	0.24	0.29
80	0.96	0.29	0.31
103	0.98	0.40	0.36
126	0.77	0.34	0.39
148	0.66	0.25	0.32
170	0.71	0.27	0.34
192	1.10	0.38	0.36
214	1.14	0.39	0.38
235	1.37	0.47	0.39
257	1.40	0.45	0.37

In [168]:

```
Departamento_QUINDIO=Produccion_df.loc[Produccion_df["Departamento"]=="QUINDIO"]
print(Departamento_QUINDIO)
# Indica los resultados estadísticos por año para el Departamento Quindio
```

	Anio	Departamento	Producto	Area (ha)	Produccion (ton) \
17	2007	QUINDIO	CAFE	19,904.00	25,426.00
39	2008	QUINDIO	CAFE	19,571.00	23,669.00
61	2009	QUINDIO	CAFE	19,052.00	21,985.00
84	2010	QUINDIO	CAFE	18,159.00	21,065.00
107	2011	QUINDIO	CAFE	20,139.30	20,814.11
130	2012	QUINDIO	CAFE	21,109.83	18,030.13
152	2013	QUINDIO	CAFE	21,203.03	20,599.27
174	2014	QUINDIO	CAFE	21,462.81	22,518.42
196	2015	QUINDIO	CAFE	21,491.21	24,694.56
217	2016	QUINDIO	CAFE	20,041.70	23,791.30
239	2017	QUINDIO	CAFE	17,699.67	18,792.05
261	2018	QUINDIO	CAFE	16,374.73	17,739.03

	Rendimiento	(ha/ton)	Produccion	Nacional	(ton)	Area	Nacional	(ha)
17		1.28			3.07			2.60
39		1.21			2.86			2.58
61		1.15			3.10			2.52
84		1.16			2.70			2.44
107		1.03			3.25			2.83
130		0.85			2.88			2.97
152		0.97			3.16			2.75
174		1.05			3.09			2.70
196		1.15			2.90			2.68
217		1.19			2.79			2.58
239		1.06			2.21			2.35
261		1.08			2.07			2.21

In [170]:

```
Estadística_Anio2015=Produccion_df.loc[Produccion_df["Anio"]==2015]
print(Estadística_Anio2015)
# Indica los resultados estadísticos por departamento para el año 2015
```

	Anio	Departamento	Producto		Area (ha)	Produccion (ton)	\
179	2015	ANTIOQUIA	CAFE	109,649.61	120,365.77		
180	2015	BOLIVAR	CAFE	1,065.07	1,089.74		
181	2015	BOYACA	CAFE	10,461.85	9,501.54		
182	2015	CALDAS	CAFE	58,376.40	67,231.37		
183	2015	CAQUETA	CAFE	3,410.56	3,749.27		
184	2015	CASANARE	CAFE	2,752.31	2,626.73		
185	2015	CAUCA	CAFE	77,405.83	83,626.44		
186	2015	CESAR	CAFE	25,948.50	22,240.81		
187	2015	CHOCO	CAFE	137.47	158.20		
188	2015	CUNDINAMARCA	CAFE	34,101.49	31,165.15		
189	2015	HUILA	CAFE	130,452.40	145,168.10		
190	2015	LA GUAJIRA	CAFE	5,631.53	4,317.50		
191	2015	MAGDALENA	CAFE	17,996.31	16,691.31		
192	2015	META	CAFE	2,922.21	3,206.35		
193	2015	NARIÑO	CAFE	33,490.93	36,607.56		
194	2015	NORTE DE SANTANDER	CAFE	22,940.64	20,267.64		
195	2015	PUTUMAYO	CAFE	128.65	124.67		
196	2015	QUINDIO	CAFE	21,491.21	24,694.56		
197	2015	RISARALDA	CAFE	41,732.03	47,215.69		
198	2015	SANTANDER	CAFE	42,679.11	47,304.16		
199	2015	TOLIMA	CAFE	103,368.73	105,563.88		
200	2015	VALLE DEL CAUCA	CAFE	54,938.79	57,583.56		

	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
179	1.10	14.15	13.69
180	1.02	0.13	0.13
181	0.91	1.12	1.31
182	1.15	7.90	7.29
183	1.10	0.44	0.43
184	0.95	0.31	0.34
185	1.08	9.83	9.66
186	0.86	2.62	3.24
187	1.15	0.02	0.02
188	0.91	3.66	4.26
189	1.11	17.07	16.28
190	0.77	0.51	0.70
191	0.93	1.96	2.25
192	1.10	0.38	0.36
193	1.09	4.30	4.18
194	0.88	2.38	2.86
195	0.97	0.01	0.02
196	1.15	2.90	2.68
197	1.13	5.55	5.21
198	1.11	5.56	5.33
199	1.02	12.41	12.90
200	1.05	6.77	6.86

In [171]:

```
Estadística_Anio2018=Produccion_df.loc[Produccion_df["Anio"]==2018]
print(Estadística_Anio2018)
# Indica los resultados estadísticos por departamento para el año 2018
```

Anio		Departamento	Producto	Area (ha)	Produccion (ton)	\
244		2018	ANTIOQUIA	CAFE	98,038.15	141,898.91
245		2018	BOLIVAR	CAFE	1,182.13	734.91
246		2018	BOYACA	CAFE	9,653.45	7,780.34
	247	2018	CALDAS	CAFE	50,762.22	68,670.96
248		2018	CAQUETA	CAFE	3,485.24	5,280.40
249		2018	CASANARE	CAFE	2,360.55	1,629.25
	250	2018	CAUCA	CAFE	82,085.54	102,147.00
	251	2018	CESAR	CAFE	23,915.45	14,943.62
252		2018	CHOCO	CAFE	140.33	181.42
253		2018	CUNDINAMARCA	CAFE	29,085.24	32,580.24
	254	2018	HUILA	CAFE	122,002.46	136,161.86
255		2018	LA GUAJIRA	CAFE	4,810.97	2,990.91
256		2018	MAGDALENA	CAFE	17,414.32	10,826.24
	257	2018	META	CAFE	2,761.01	3,877.62
	258	2018	NARIÑO	CAFE	33,465.54	35,679.42
	259	2018	NORTE DE SANTANDER	CAFE	20,873.04	23,471.69
	260	2018	PUTUMAYO	CAFE	209.93	289.50
	261	2018	QUINDIO	CAFE	16,374.73	17,739.03
	262	2018	RISARALDA	CAFE	35,874.73	45,918.75
	263	2018	SANTANDER	CAFE	42,269.07	55,918.71
	264	2018	TOLIMA	CAFE	97,304.04	97,451.31
	265	2018	VALLE DEL CAUCA	CAFE	48,305.31	49,667.88

	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
244	1.45	16.58	13.21
245	0.62	0.09	0.16
246	0.81	0.91	1.30
247	1.35	8.02	6.84
248	1.52	0.62	0.47
249	0.69	0.19	0.32
250	1.24	11.94	11.06
251	0.62	1.75	3.22
252	1.29	0.02	0.02
253	1.12	3.81	3.92
254	1.12	15.91	16.43
255	0.62	0.35	0.65
256	0.62	1.26	2.35
257	1.40	0.45	0.37
258	1.07	4.17	4.51
259	1.12	2.74	2.81
260	1.38	0.03	0.03
261	1.08	2.07	2.21
262	1.28	5.37	4.83
263	1.32	6.53	5.69
264	1.00	11.39	13.11
265	1.03	5.80	6.51

In [173]:
```
Produccion_df[0:10]
#lista los primeros 10 elementos del dataframe
```

Out[173]:

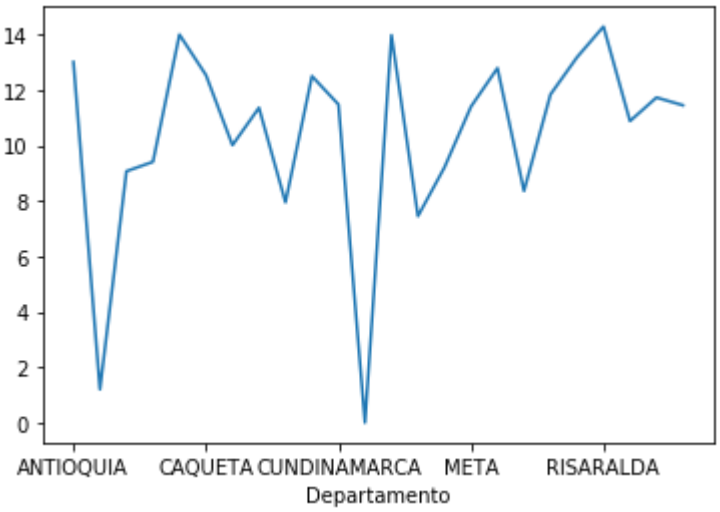
	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
0	2007	ANTIOQUIA	CAFE	112,343.60	120,500.80	1.07	14.54	14.66
1	2007	BOLIVAR	CAFE	502.00	446.00	0.89	0.05	0.07
2	2007	BOYACA	CAFE	11,374.50	9,683.10	0.85	1.17	1.48
3	2007	CALDAS	CAFE	78,393.65	92,815.00	1.18	11.20	10.23
4	2007	CAQUETA	CAFE	2,295.00	2,134.00	0.93	0.26	0.30
5	2007	CASANARE	CAFE	2,605.00	2,048.40	0.79	0.25	0.34
6	2007	CAUCA	CAFE	53,471.00	51,348.00	0.96	6.19	6.98
7	2007	CESAR	CAFE	23,172.00	13,278.50	0.57	1.60	3.02
8	2007	CHOCO	CAFE	290.00	205.90	0.71	0.02	0.04
9	2007	CUNDINAMARCA	CAFE	43,017.30	33,729.14	0.78	4.07	5.61

In [174]:
```
Produccion_df[11:30]
#lista los elementos desde el 11 al 29, no incluye el 30
```

Out[174]:

	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
11	2007	LA GUAJIRA	CAFE	4,785.00	2,958.70	0.62	0.36	0.62
12	2007	MAGDALENA	CAFE	17,506.00	14,005.00	0.80	1.69	2.28
13	2007	META	CAFE	2,048.00	1,617.20	0.79	0.20	0.27
14	2007	NARIÑO	CAFE	24,458.50	31,770.05	1.30	3.83	3.19
15	2007	NORTE DE SANTANDER	CAFE	30,171.84	13,593.24	0.45	1.64	3.94
16	2007	PUTUMAYO	CAFE	35.00	34.00	0.97	0.00	0.00
17	2007	QUINDIO	CAFE	19,904.00	25,426.00	1.28	3.07	2.60
18	2007	RISARALDA	CAFE	47,689.25	72,842.55	1.53	8.79	6.22
19	2007	SANTANDER	CAFE	34,406.67	29,469.52	0.86	3.56	4.49
20	2007	TOLIMA	CAFE	91,679.10	112,322.38	1.23	13.55	11.96
21	2007	VALLE DEL CAUCA	CAFE	76,667.80	69,618.24	0.91	8.40	10.00
22	2008	ANTIOQUIA	CAFE	114,694.00	113,505.20	0.99	13.70	15.13
23	2008	BOLIVAR	CAFE	572.00	711.00	1.24	0.09	0.08
24	2008	BOYACA	CAFE	10,778.50	9,547.30	0.89	1.15	1.42
25	2008	CALDAS	CAFE	74,897.00	86,884.00	1.16	10.49	9.88
26	2008	CAQUETA	CAFE	2,735.00	2,469.00	0.90	0.30	0.36
27	2008	CASANARE	CAFE	2,149.00	1,388.13	0.65	0.17	0.28
28	2008	CAUCA	CAFE	56,208.00	48,073.00	0.86	5.80	7.41
29	2008	CESAR	CAFE	23,198.00	13,841.45	0.60	1.67	3.06

In [182]:
```
total_count = Produccion_df.groupby('Departamento')['Rendimiento (ha/ton)'].sum( # creamos una grafica lineal indicando la cantidad de Departamento por Rendimien
total_count.plot(kind='line');
```



In [184]:
```
Produccion_df.tail()
#Muestra los ultimos valores del dataframe
```

Out[184]:

	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
261	2018	QUINDIO	CAFE	16,374.73	17,739.03	1.08	2.07	2.21
262	2018	RISARALDA	CAFE	35,874.73	45,918.75	1.28	5.37	4.83
263	2018	SANTANDER	CAFE	42,269.07	55,918.71	1.32	6.53	5.69
264	2018	TOLIMA	CAFE	97,304.04	97,451.31	1.00	11.39	13.11
265	2018	VALLE DEL CAUCA	CAFE	48,305.31	49,667.88	1.03	5.80	6.51

PANDAS PROFILING

In [176]:

```
# USO DE PANDAS  PROFILING

# Instructor Ing. Luis Armando Amaya Q.

import pandas as  pd

import numpy as np

from pandas_profiling import ProfileReport

profile=ProfileReport(produccion_df, title='CAFE', html ={'style': {'full_width'
profile
#NOTA IMPORTANTE

# LA DOS SIGUIENTES INSTRUCCIONES, CREAN UN INFORME EN FORMATO HTML

# DEBE BUSCARLO EN SU COMPUTADOR CON EL NOMBRE:---> ANALISIS EXPLORATORIO CADE_P  # LUEGO DE
ENCONTRAR LA CARPETA ---> Producción_Cafe                                    <----------------

# PARA ABRIR EL INFORME DEBE HACER CLIC SOBRE EL ARCHIVO LLAMADO ----------------------------------------->your
```

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

Overview

Dataset statistics

Number of variables	8
Number of observations	266
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	16.8 KiB
Average record size in memory	64.5 B

Variable types

NUM	4
CAT	4

Warnings

Producto has constant value "266"	Constant
Area (ha) has a high cardinality: 261 distinct values	High cardinality
Produccion (ton) has a high cardinality: 262 distinct values	High cardinality
Area Nacional (ha) is highly correlated with Produccion	High correlation

Out[176]:

In [ ]:
```
#NOTA IMPORTANTE

# LA DOS SIGUIENTES INSTRUCCIONES, CREAN UN INFORME EN FORMATO HTML

# DEBE BUSCARLO EN SU COMPUTADOR CON EL NOMBRE:---> ANALISIS EXPLORATORIO CADE_P  # LUEGO DE
ENCONTRAR LA CARPETA ---> Producción_Cafe                                    <----------------

# PARA ABRIR EL INFORME DEBE HACER CLIC SOBRE EL ARCHIVO LLAMADO ----------------------------------------->your

# RECUERDE: -----> LA DOS SIGUIENTES INSTRUCCIONES, CREAN UN INFORME EN FORMATO # TAMBIÉN LE SUBÍ
```

In [ ]:
```
#    En este punto se inicia procedimientos gráficos y estadísticos  particulares

#    Espero que consulte en Internet algunos conceptos si tiene dudas o quiere rec
```

In [160]:
```
import numpy as np      # libreria para calculos

import matplotlib.pyplot as plt

#%matplotlib inline

import seaborn as sns   #Esta libreria permite construir gráficos muy particulare     #si se
requiere se puede Definir un indice para listar la informacion del datafr  # por ejemplo -
-->produccion_df=produccion_df.set_index('Departamento')
```

Out[160]:

Rendimiento (ha/ton)	Area (ha)	Produccion Nacional (ton)	Area Nacional (ha)
1.07	112,343.60	14.54	14.66
0.89	502.00	0.05	0.07
0.85	11,374.50	1.17	1.48
1.18	78,393.65	11.20	10.23
0.93	2,295.00	0.26	0.30

In [163]:
```
# Para estar seguros del dataframe original, se vuelve a leer

produccion_df=pd.read_csv("produccionc.csv")
# Asignación del nombre del Dataframe
```

In [164]:
```
produccion_df.describe()
# Información estadístico del Dataframe para las variables
```

Out[164]:

	Anio	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
count	266.000000	266.000000	266.000000	266.000000
mean	2012.469925	0.936429	4.511316	4.511203
std	3.443484	0.267129	4.950568	4.565865
min	2007.000000	0.000000	0.000000	0.000000
25%	2010.000000	0.750000	0.352500	0.390000
50%	2012.000000	0.940000	2.720000	3.120000
75%	2015.000000	1.120000	7.147500	6.875000
max	2018.000000	2.000000	18.670000	16.430000

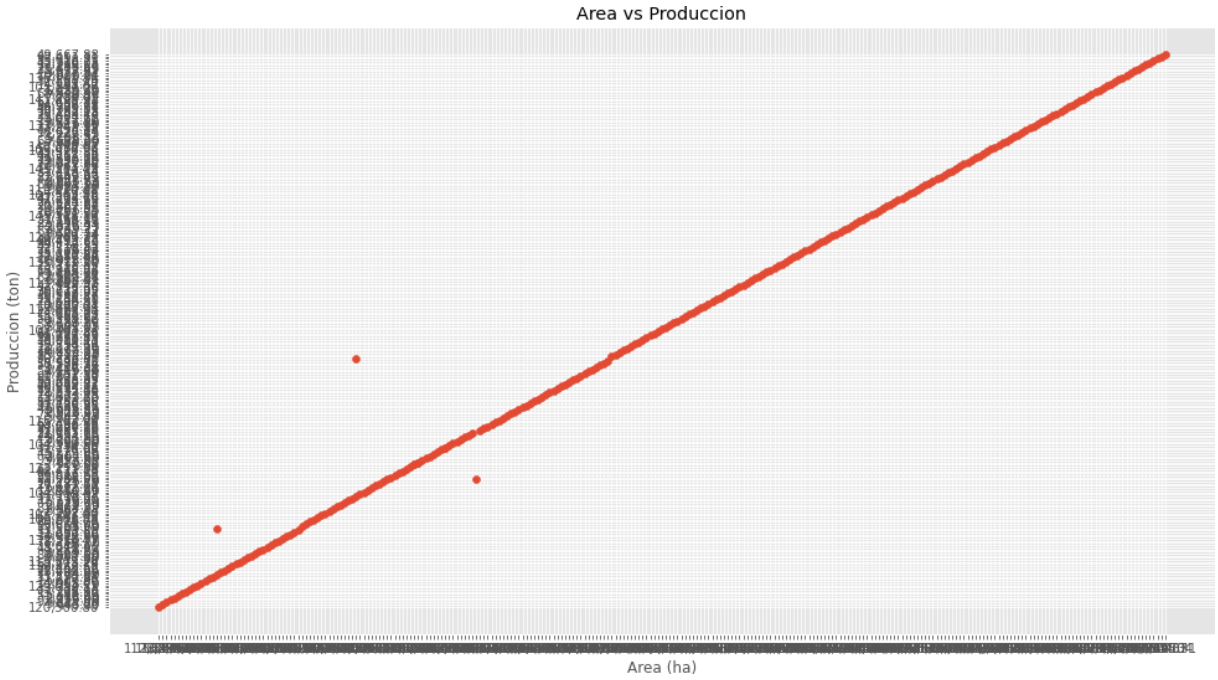
```
# estas instrucciones aun no las voy a emplear, por eso están con el simbolo #
#arreglo=list(produccion_df.columns)
#produccion1_df= produccion_df[arreglo[2:len(arreglo)]]
#produccion1_df
#print(produccion1_df)
#arreglo2.describe()
```

In [165]:
```
# Obtenemos información de los tipos de las variables del Dataframe o DataSet

produccion_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 266 entries, 0 to 265
Data columns (total 8 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   Anio                        266 non-null    int64  
 1   Departamento                266 non-null    object 
 2   Producto                    266 non-null    object 
 3   Area (ha)                   266 non-null    object 
 4   Produccion (ton)            266 non-null    object 
 5   Rendimiento (ha/ton)        266 non-null    float64
 6   Produccion Nacional (ton)   266 non-null    float64
 7   Area Nacional (ha)          266 non-null    float64
dtypes: float64(3), int64(1), object(4)
memory usage: 16.8+ KB
```

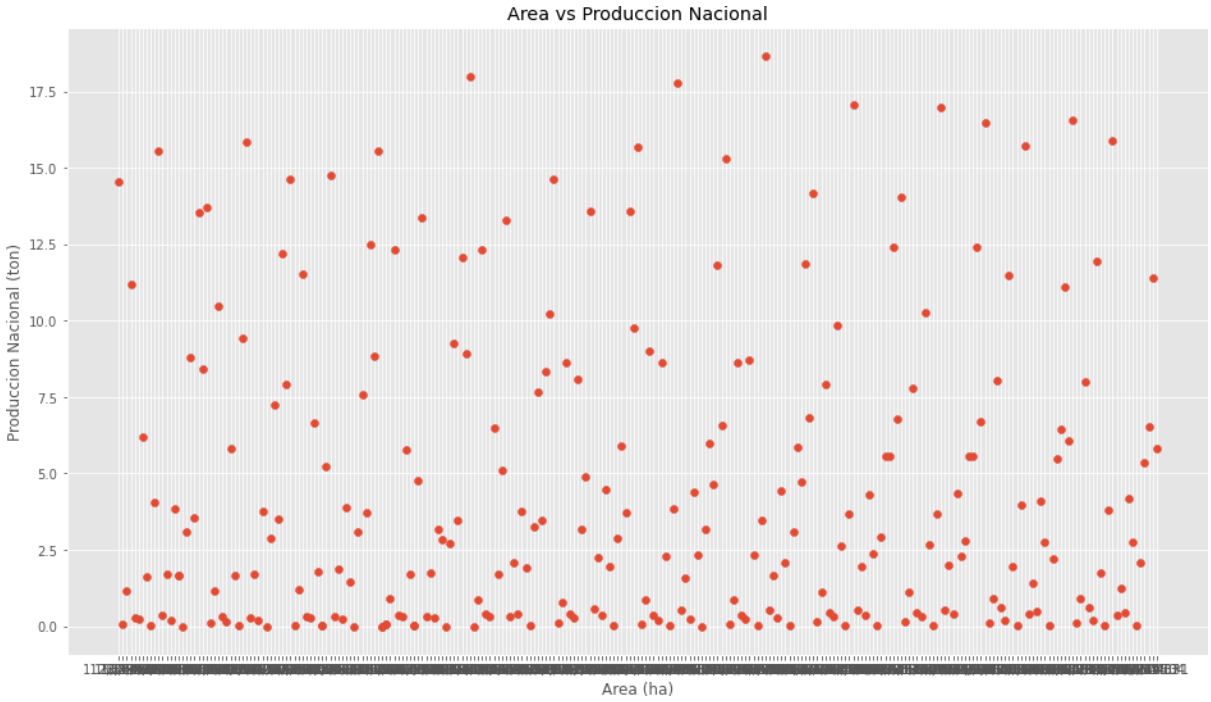
In [47]:
```
# Gráfico del comportamiento del Area versus Produccion.

plt.scatter(produccion_df['Area (ha)'],produccion_df['Produccion (ton)'])
plt.title('Area vs Produccion')
plt.xlabel('Area (ha)')
plt.ylabel("Produccion (ton)")
```

Out[47]: Text(0, 0.5, 'Produccion (ton)')



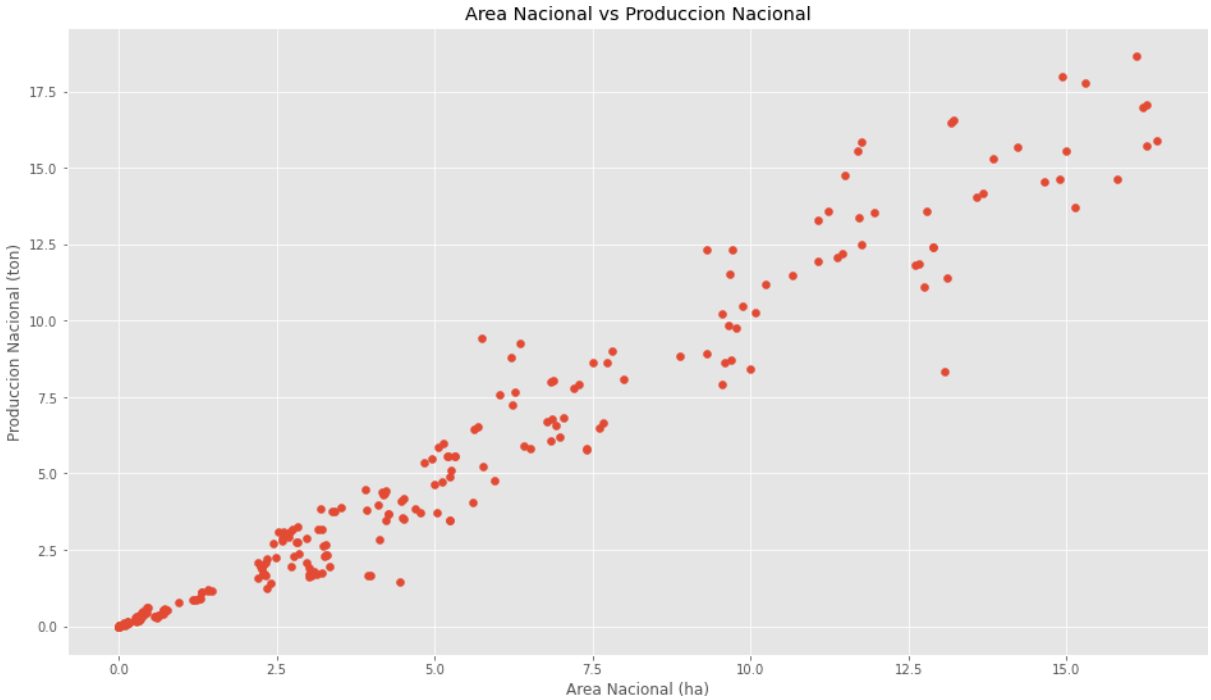
In [40]:
```
# Gráfico del comportamiento del Area versus Produccion Nacional

plt.scatter(produccion_df['Area (ha)'],produccion_df['Produccion Nacional (ton)'
plt.title('Area vs Produccion Nacional')
plt.xlabel('Area (ha)')
plt.ylabel("Produccion Nacional (ton)")
```

Out[40]: Text(0, 0.5, 'Produccion Nacional (ton)')



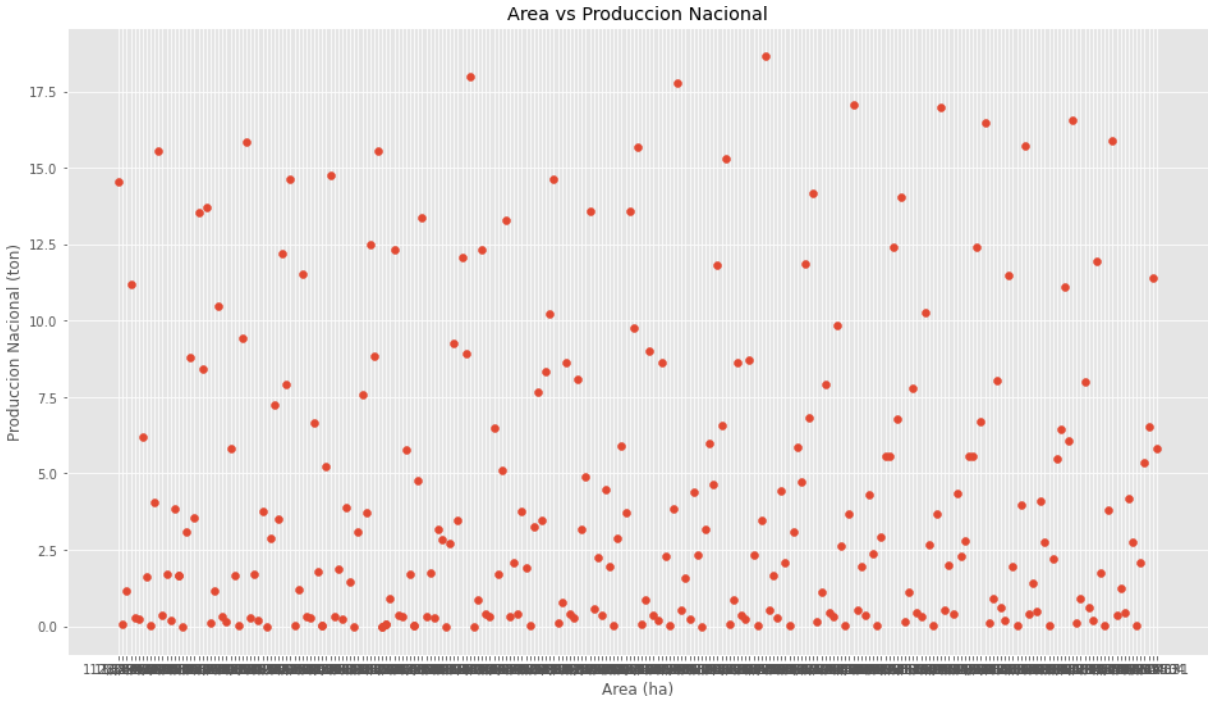
In [49]:
```
# Gráfico del comportamiento del Area Nacional versus Produccion Nacional

plt.scatter(produccion_df['Area Nacional (ha)'],produccion_df['Produccion Nacion
plt.title('Area Nacional vs Produccion  Nacional')
plt.xlabel('Area Nacional (ha)')
plt.ylabel("Produccion Nacional (ton)")
```

Out[49]: Text(0, 0.5, 'Produccion Nacional (ton)')



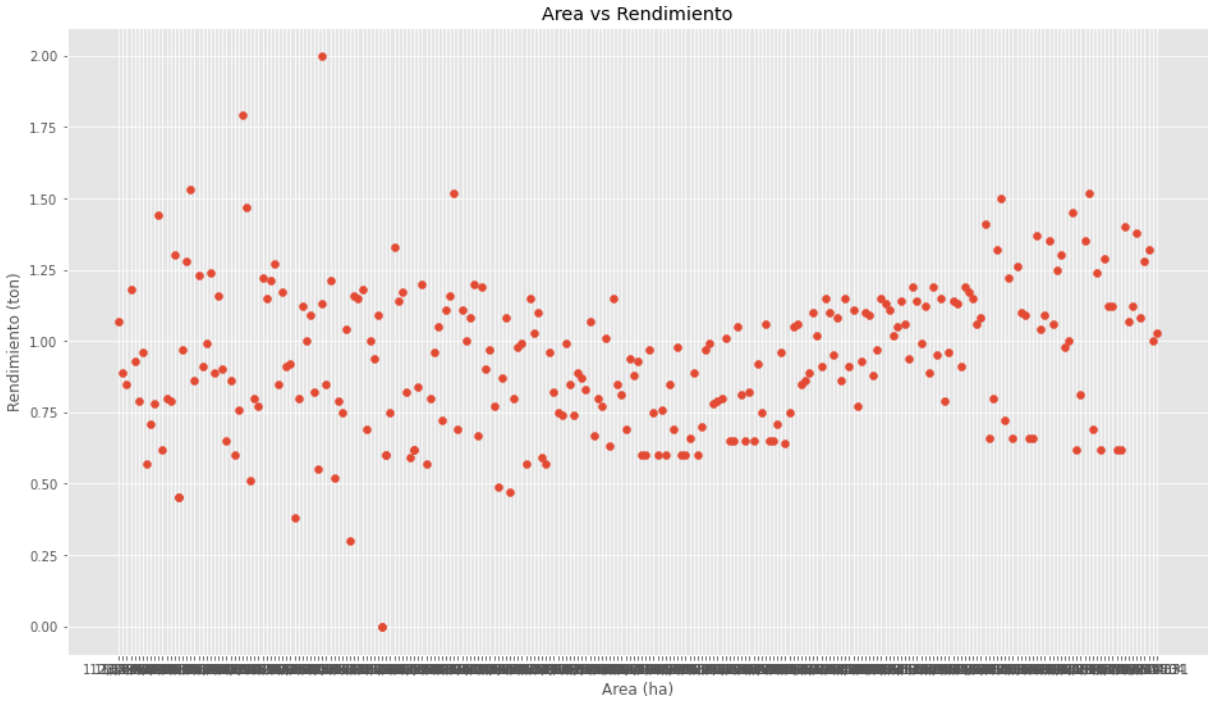
In [41]:
```
# Gráfico del comportamiento del Area versus Produccion Nacional

plt.scatter(produccion_df['Area (ha)'],produccion_df['Produccion Nacional (ton)'
plt.title('Area vs Produccion Nacional')
plt.xlabel('Area (ha)')
plt.ylabel("Produccion Nacional (ton)")
```

Out[41]: Text(0, 0.5, 'Produccion Nacional (ton)')



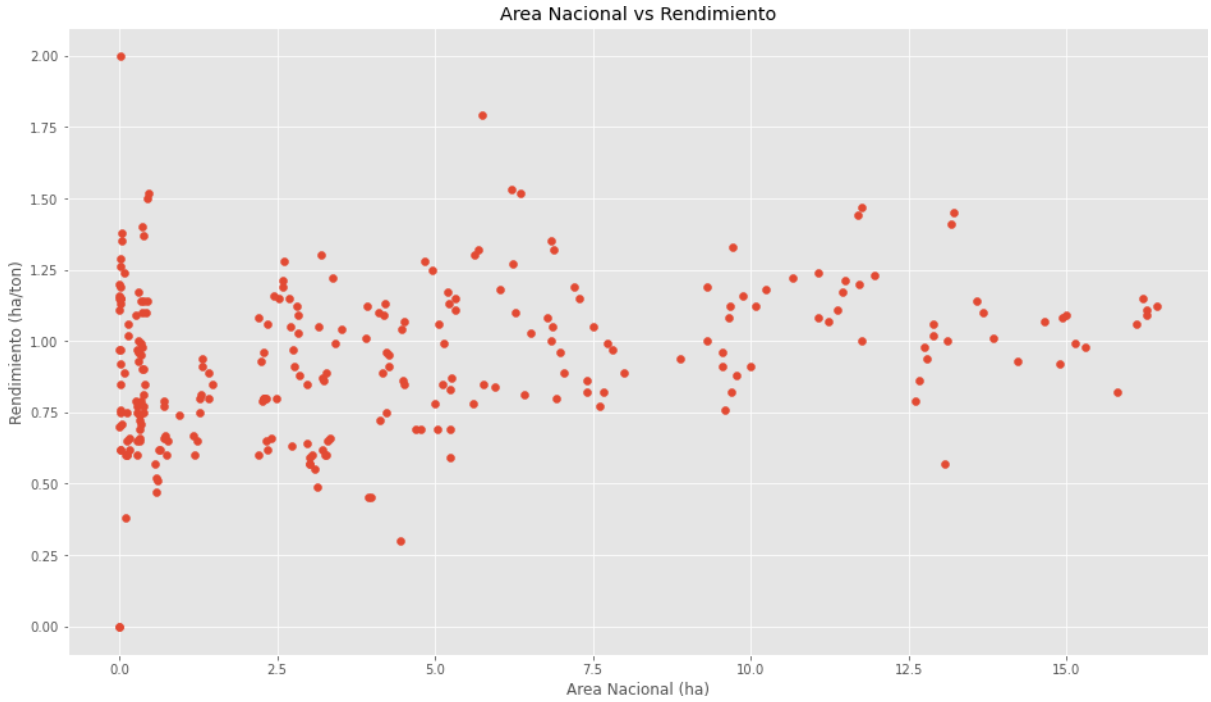
In [43]:
```
# Gráfico del comportamiento del Area versus Rendimiento

plt.scatter(produccion_df['Area (ha)'],produccion_df['Rendimiento (ha/ton)'])
plt.title('Area vs Rendimiento')
plt.xlabel('Area (ha)')
plt.ylabel("Rendimiento (ton)")
```

Out[43]:
```
Text(0, 0.5, 'Rendimiento (ton)')
```



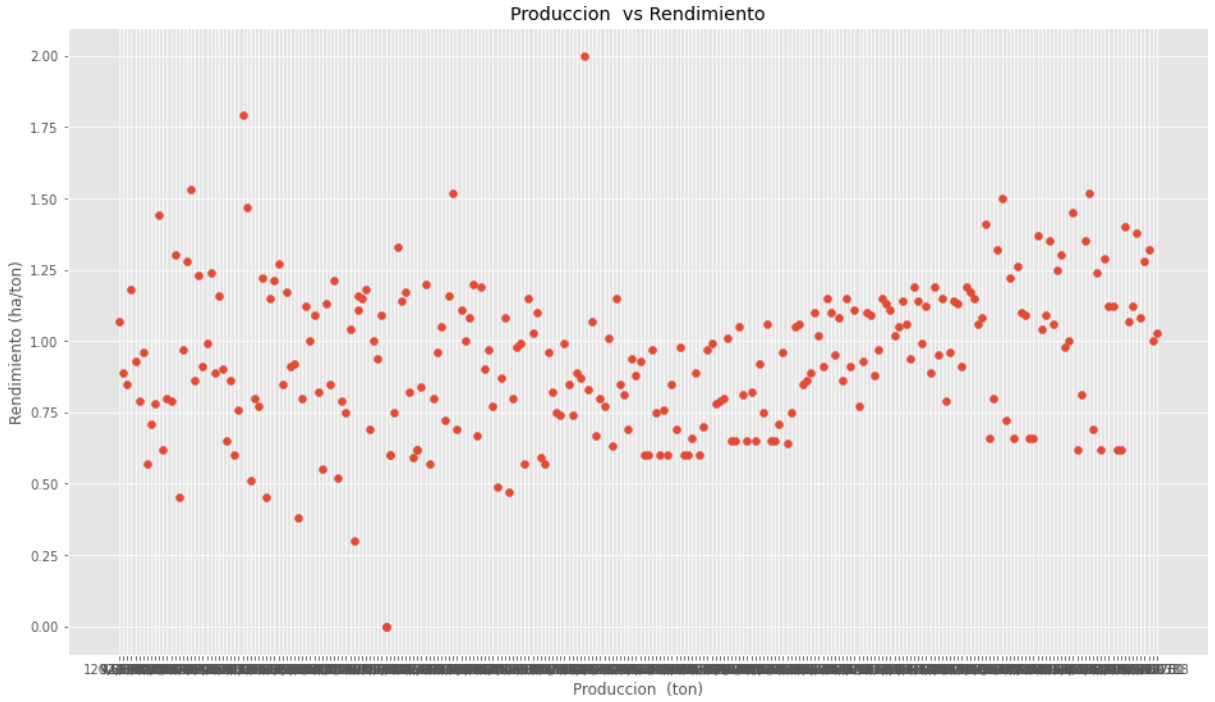
In [44]:
```
# Gráfico del comportamiento del Area Nacional versus Rendimiento

plt.scatter(produccion_df['Area Nacional (ha)'],produccion_df['Rendimiento (ha/t
plt.title('Area Nacional vs Rendimiento')
plt.xlabel('Area Nacional (ha)')
plt.ylabel("Rendimiento (ha/ton)")
```

Out[44]: Text(0, 0.5, 'Rendimiento (ha/ton)')



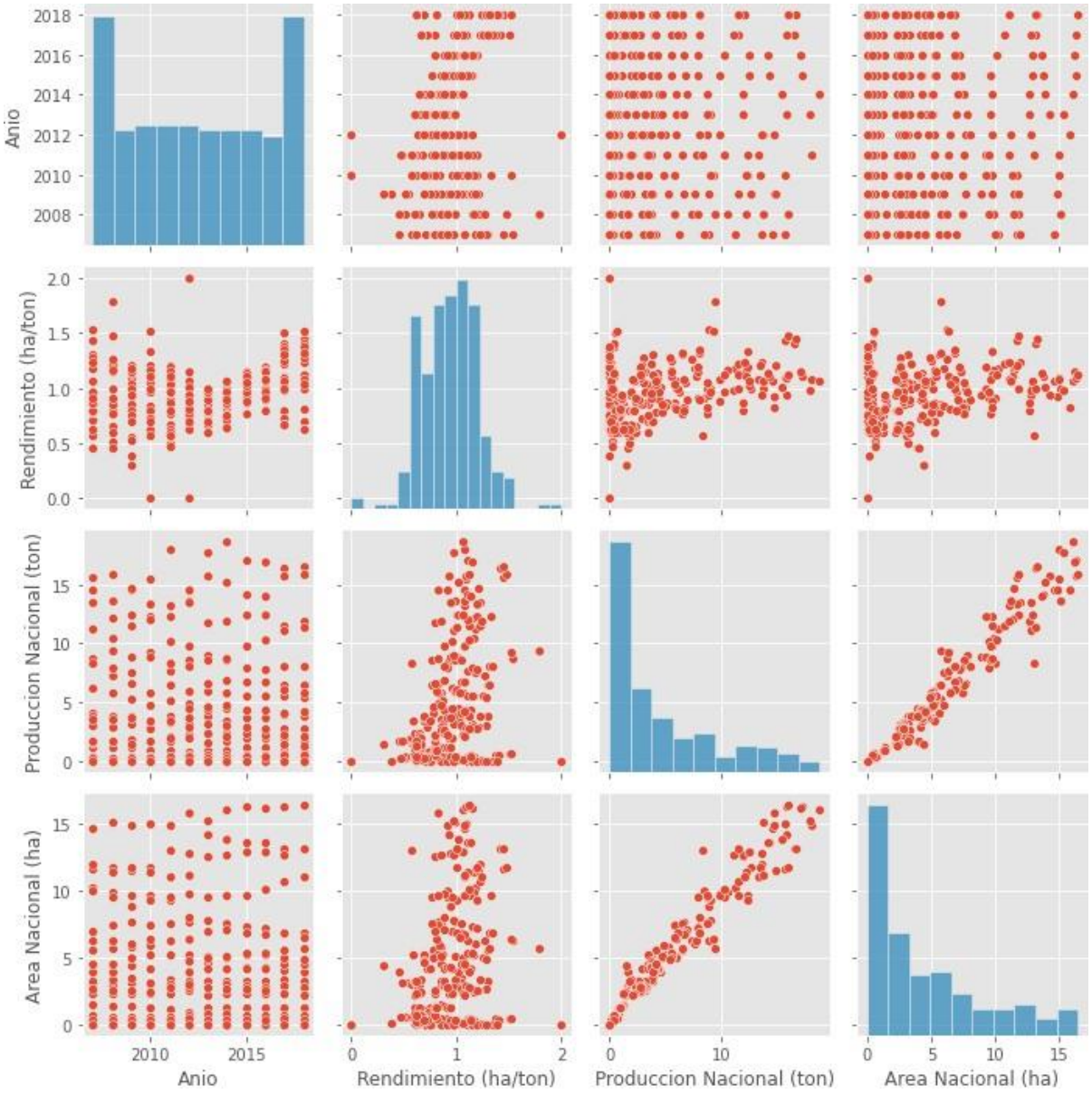
In [45]:
```
# Gráfico del comportamiento de la Produccion versus Rendimiento

plt.scatter(produccion_df['Produccion (ton)'], produccion_df['Rendimiento (ha/to
plt.title('Produccion  vs Rendimiento')
plt.xlabel('Produccion  (ton)')
plt.ylabel("Rendimiento (ha/ton)")
```

Out[45]: Text(0, 0.5, 'Rendimiento (ha/ton)')



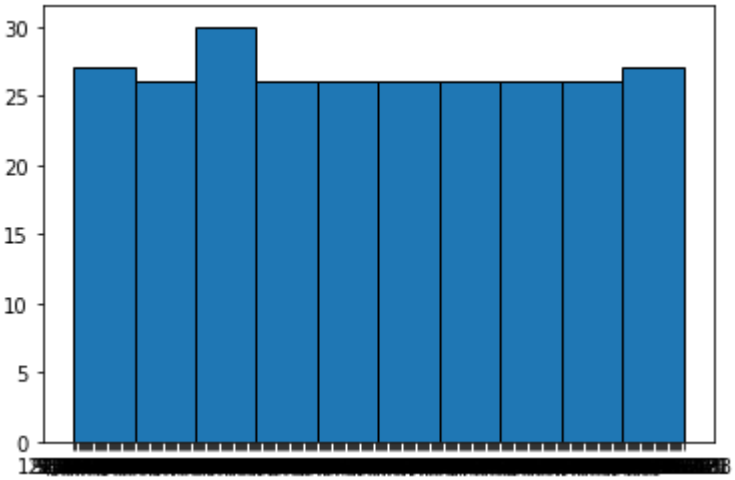
In [42]:
```
import seaborn as sns        #Esta libreria permite construir gráficos muy  particulare

sns.pairplot(produccion_df)
```

Out[42]: <seaborn.axisgrid.PairGrid at 0x1c3e522c7f0>



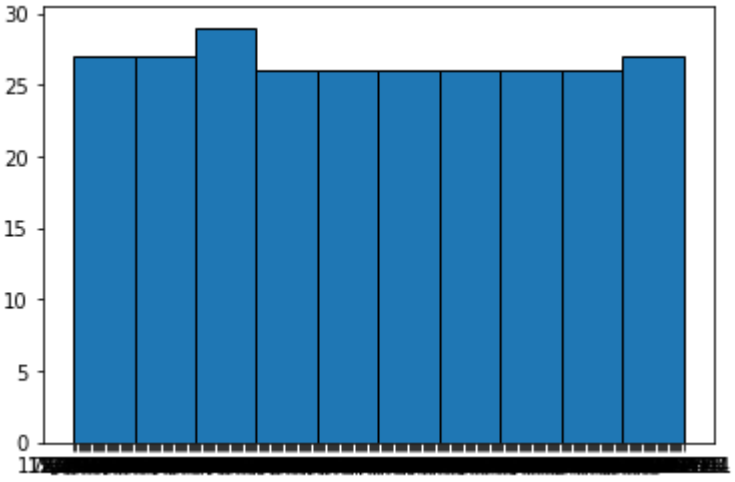
In [170]:
```
# Histograma de la Produccion de Café

plt.hist(produccion_df['Produccion (ton)'], edgecolor='black', linewidth=1)
```

Out[170]:
```
(array([27., 26., 30., 26., 26., 26., 26., 26., 26., 27.]),
 array([  0. ,  26.1,  52.2,  78.3, 104.4, 130.5, 156.6, 182.7, 208.8,
        234.9, 261. ]),
 <a list of 10 Patch objects>)
```



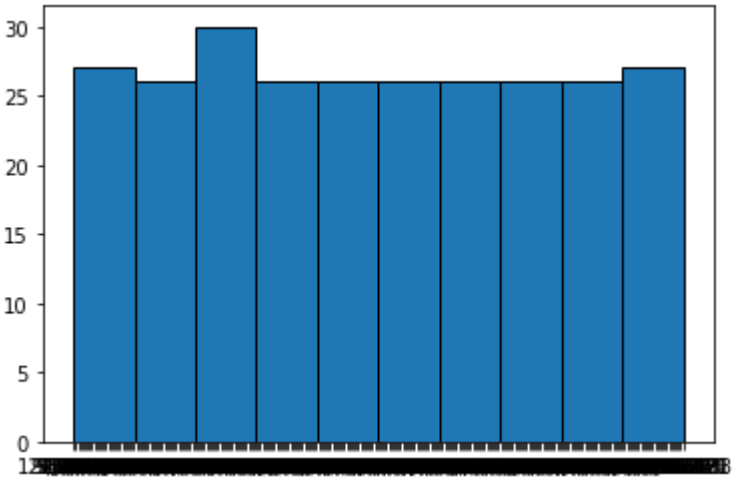
In [126]:
```
# Histograma del Área sembrada

plt.hist(produccion_df['Area (ha)'], edgecolor='black', linewidth=1)
```

Out[126]:
```
(array([27., 27., 29., 26., 26., 26., 26., 26., 26., 27.]),
 array([  0.,  26.,  52.,  78., 104., 130., 156., 182., 208., 234., 260.]),
 <a list of 10 Patch objects>)
```



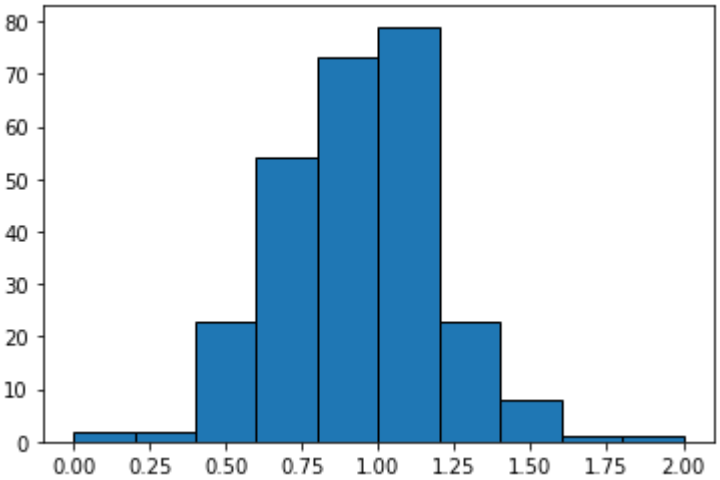
In [127]:
```
# Histograma de la produccion de Café

plt.hist(produccion_df['Produccion (ton)'], edgecolor='black', linewidth=1)
```

Out[127]:
```
(array([27., 26., 30., 26., 26., 26., 26., 26., 26., 27.]),
 array([  0. ,  26.1,  52.2,  78.3, 104.4, 130.5, 156.6, 182.7, 208.8,
        234.9,  261. ]),
 <a list of 10 Patch objects>)
```



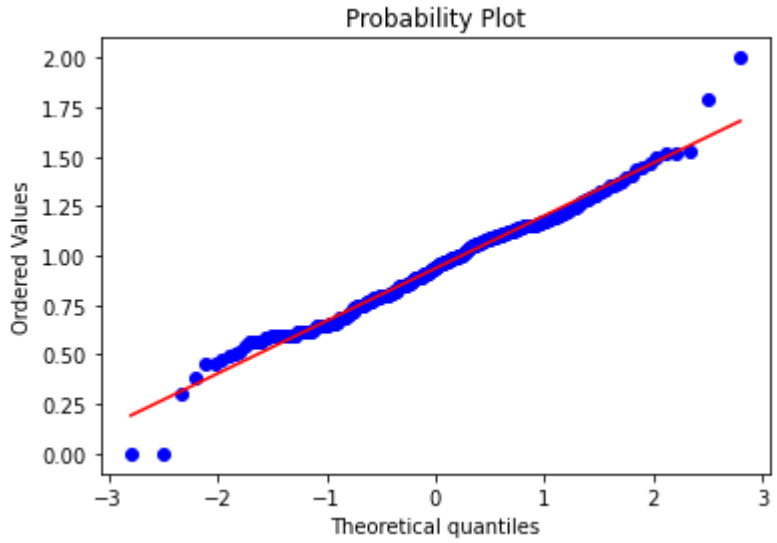
In [171]:
```
# Histograma del rendimiento del Café

plt.hist(produccion_df['Rendimiento (ha/ton)'], edgecolor='black', linewidth=1)
```

Out[171]:
```
(array([ 2.,  2., 23., 54., 73., 79., 23.,  8.,  1.,  1.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <a list of 10 Patch objects>)
```



In [ ]:
```
# el anterior histograma tiene la forma de la campana de Gauss, lo que indica qu
```

In [172]:
```
# Además, para corroborar la anterior distribucion normal, podemos construir # el gráfico QUANTILE-QUANTILE NORMAL

# si los puntos están muy cerca a la linea recta, indica que los valores tienen

import pylab

import scipy.stats as stats #librerias para construir estos tipos de graficos
```



In [187]:
```
# importar la liberia shapiro para realizar el TEST DE SHAPIRO WILK,

# el test de Shapiro Wilk CONFIRMA EFECTIVAMENTE la correlacion entre las variab

from scipy.stats import shapiro

estadistico,p_value =shapiro(produccion_df['Rendimiento (ha/ton)'])
print('Estadística=%.3f, El Valor de: p_value=%.3f' % (estadistico,p_value))
# Si el valor entregado en la variable P_VALUE es MENOR a 0.05, # indica
```

Estadística=0.983, El Valor de: p_value=0.003

In [177]:
```
# valores correlacion de Spearman

import numpy as np

produccion_correlacion_spearman = produccion_df.corr(method='spearman')
produccion_correlacion_spearman
# los valores del COEFICIENTE DE SPEARMAN cercanos a cero o inferiores a (+-)(0.    # indica
que las variables no tienen correlacion
```

Out[177]:

	Anio	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
Anio	1.000000	0.180205	0.037725	0.023246
Rendimiento (ha/ton)	0.180205	1.000000	0.366952	0.264041
Produccion Nacional (ton)	0.037725	0.366952	1.000000	0.986380
Area Nacional (ha)	0.023246	0.264041	0.986380	1.000000

In [178]:
```
# valores correlacion de Pearson

import numpy as np

produccion_correlacion_pearson = produccion_df.corr(method='pearson')
```

Out[178]:

	Anio	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
Anio	1.000000	0.173474	0.007957	0.008715
Rendimiento (ha/ton)	0.173474	1.000000	0.385570	0.280677
Produccion Nacional (ton)	0.007957	0.385570	1.000000	0.978409
Area Nacional (ha)	0.008715	0.280677	0.978409	1.000000

In [180]:
```
# valores correlacion de Kendall

import numpy as np

produccion_correlacion_kendall = produccion_df.corr(method='kendall')
```

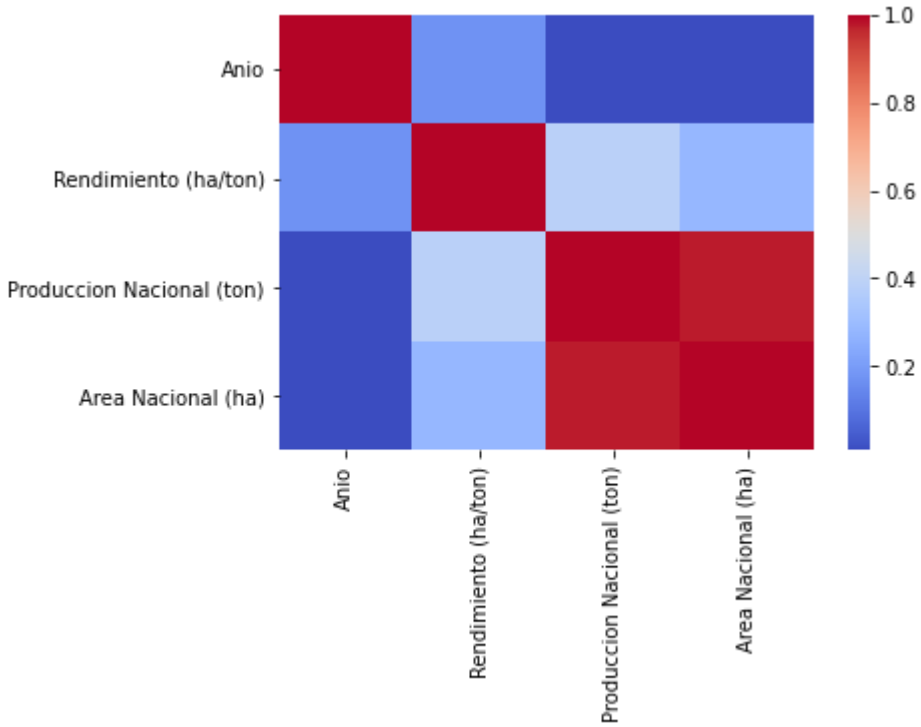
Out[180]:

	Anio	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
Anio	1.000000	0.140836	0.026879	0.016567
Rendimiento (ha/ton)	0.140836	1.000000	0.265165	0.186979
Produccion Nacional (ton)	0.026879	0.265165	1.000000	0.909233
Area Nacional (ha)	0.016567	0.186979	0.909233	1.000000

In [186]:

```
# Generacion de mapa  de calor para observar fácilmente las variables correlacion
# las rojas son correlaciones fuertes positivas y las azules correlaciones  negat
import seaborn as sns
# esta libreria permite crear gráficos  estadísticos
sns.heatmap(produccion_correlacion_pearson,
            xticklabels=produccion_correlacion_pearson.col
            umns,
            yticklabels=produccion_correlacion_pearson.col
            umns, cmap='coolwarm'
           )
```

Out[186]: <matplotlib.axes._subplots.AxesSubplot at 0x1e337877130>



MODELO PREDICTIVO

In [33]:

```
# Imports  necesarios

import numpy as np import
pandas as pd import
seaborn  as sb

import matplotlib.pyplot as plt

%matplotlib inline
from mpl_toolkits.mplot3d import Axes3D

from matplotlib import cm
```

In [35]:
```
produccion_df.shape
#Nos indica un dataframe de 266 registros con 8 variables  o
```

Out[35]: (266, 8)

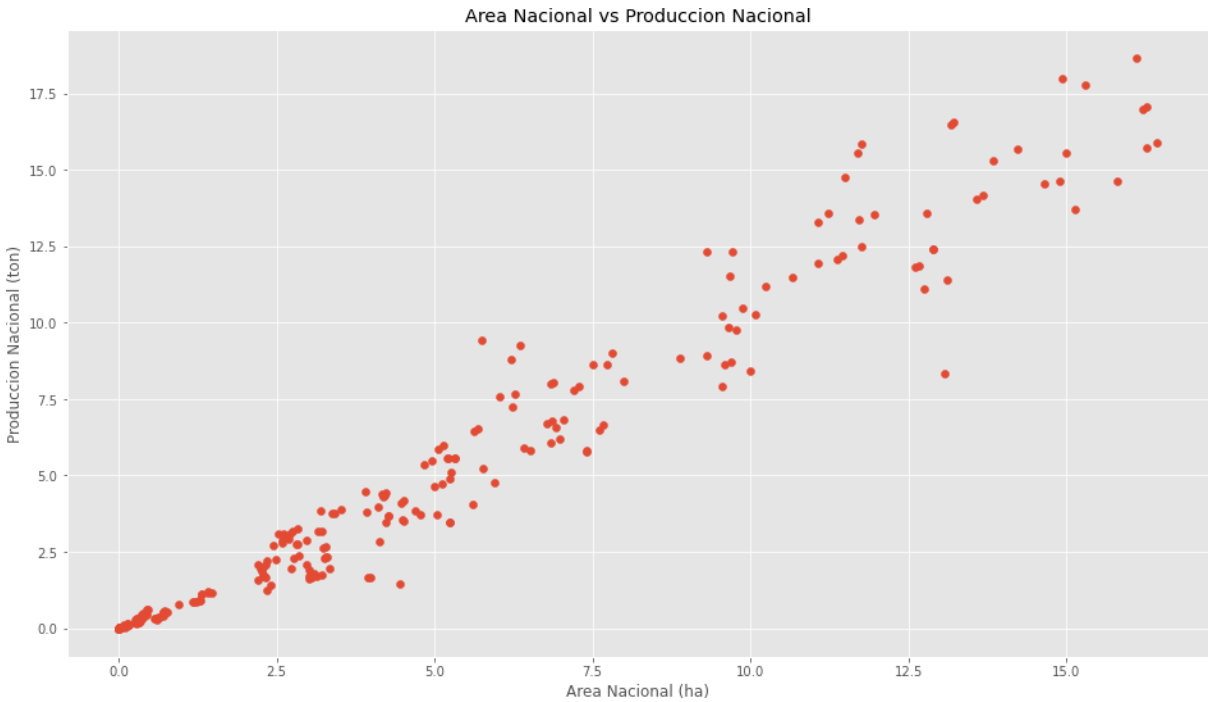
In [36]:
```
produccion_df.describe()
```

Out[36]:

	Anio	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
count	266.000000	266.000000	266.000000	266.000000
mean	2012.469925	0.936429	4.511316	4.511203
std	3.443484	0.267129	4.950568	4.565865
min	2007.000000	0.000000	0.000000	0.000000
25%	2010.000000	0.750000	0.352500	0.390000
50%	2012.000000	0.940000	2.720000	3.120000
75%	2015.000000	1.120000	7.147500	6.875000
max	2018.000000	2.000000	18.670000	16.430000

In [88]:
```
# Gráfico de dispersión del comportamiento del Area Nacional versus Produccion N
plt.scatter(produccion_df['Area Nacional (ha)'],produccion_df['Produccion Nacion
plt.title('Area Nacional vs Produccion Nacional')

plt.xlabel('Area Nacional (ha)')
plt.ylabel("Produccion Nacional (ton)")
```

Out[88]: Text(0, 0.5, 'Produccion Nacional (ton)')



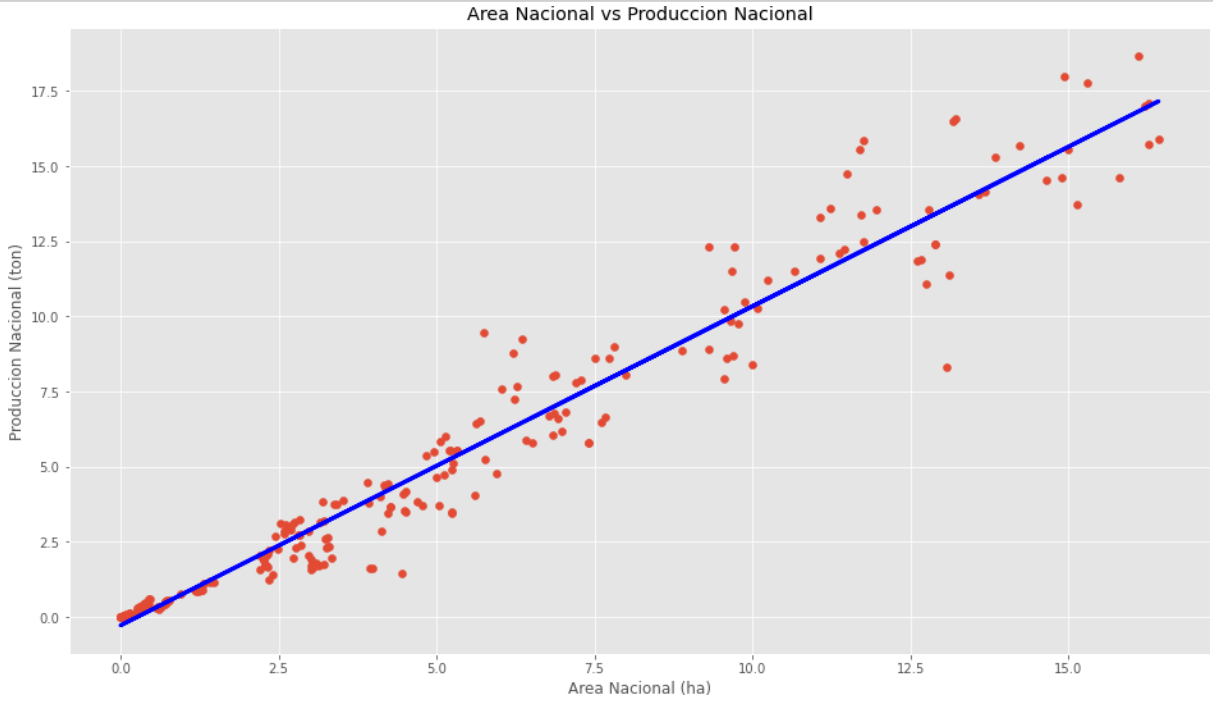
In [89]:
```
# Iniciamos el proceso para determinar el modelo de regresion lineal, de la anal
# Asignamos a nuestra variable de entrada X (En este caso corresponde al Area Na
# Asignamos a la variable dependiente Y (En este caso corresponde a la Produccio dataX
=produccion_df[["Area Nacional (ha)"]]
X_train = np.array(dataX)
y_train = produccion_df['Produccion Nacional (ton)'].values
```

In [90]:
```
# Creamos la función objeto para determinar la Regresión Lineal Y= mX+bo

regr = linear_model.LinearRegression()

# Entrenamos nuestro modelo de regresion lineal, con la siguiente función

regr.fit(X_train, y_train)

# Hacemos las predicciones segun el modelo de regresion  lineal

y_pred = regr.predict(X_train)

# Ahora imprimimos los resultados obtenidos

# Vemos el valor de la pendiente, osea la variable m, el coeficiente de la varia

print('Valor de la tangente (m) o Coefficients:=======> ', regr.coef_)
# Ahora el valor de la constante bo, es decir el valor donde la recta corta el e
```

```
Valor de la tangente (m) o Coefficients:=======>  [1.06084584]
Valor de la constante o Independent term: =====> -0.2743751434833559
Error cuadrado medio o Mean squared error:=====> 1.04
valor de la varianza o Variance score:=========> 0.96
```

In [91]:

```
# Gráfico de dispersion del comportamiento del Area Nacional versus Produccion N
plt.scatter(produccion_df['Area Nacional (ha)'],produccion_df['Produccion Nacion
plt.title('Area Nacional vs Produccion Nacional')

plt.xlabel('Area Nacional (ha)')
plt.ylabel("Produccion Nacional (ton)")

# A continuación se grafica en colo azul, la funcion lineal obtenida a partir de

plt.plot(X_train[:, 0], y_pred, color='blue', linewidth=3)
```



In [92]:

```
# Ahora vamos  a predecir utilizando la función obtenida, la producción nacional   #
Queremos predecir cuántos toneladas de producción nacional de Café vamos a obt #  según
nuestro  modelo,  hacemos:

produccion_obtenida  =  regr.predict([[2]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %prod
```

```
Estimación  de  la  Producción  Nacional  del  Café  en  toneladas===>1.847
```

In [93]:

```
# Ahora vamos  a predecir utilizando la función obtenida, la producción nacional   #
Queremos predecir cuántos toneladas de producción nacional de Café vamos a obt #  según
nuestro  modelo,  hacemos:

produccion_obtenida = regr.predict([[2.5]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %prod
```

```
Estimación  de  la  Producción  Nacional  del  Café  en  toneladas===>2.378
```

In [94]:
```
# Ahora vamos  a predecir utilizando la función obtenida, la producción nacional   #
Queremos predecir cuántos toneladas de producción nacional de Café vamos a obt #  según
nuestro  modelo,  hacemos:

produccion_obtenida  =  regr.predict([[8]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %prod
```

```
Estimación  de  la  Producción  Nacional  del  Café  en  toneladas===>8.212
```

In [95]:
```
# Ahora vamos  a predecir utilizando la función obtenida, la producción nacional   #
Queremos predecir cuántos toneladas de producción nacional de Café vamos a obt #  según
nuestro  modelo,  hacemos:

produccion_obtenida  =  regr.predict([[11]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %prod
```

```
Estimación  de  la  Producción  Nacional  del  Café  en  toneladas===>11.395
```

In [96]:
```
# Ahora vamos  a predecir utilizando la función obtenida, la producción nacional   #
Queremos predecir cuántos toneladas de producción nacional de Café vamos a obt #  según
nuestro  modelo,  hacemos:

produccion_obtenida  =  regr.predict([[15]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %prod
```

```
Estimación  de  la  Producción  Nacional  del  Café  en  toneladas===>15.638
```

In [97]:
```
# Ahora vamos  a predecir utilizando la función obtenida, la producción nacional   #
Queremos predecir cuántos toneladas de producción nacional de Café vamos a obt #  según
nuestro  modelo,  hacemos:

produccion_obtenida  =  regr.predict([[35]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %prod
```

```
Estimación  de  la  Producción  Nacional  del  Café  en  toneladas===>36.855
```

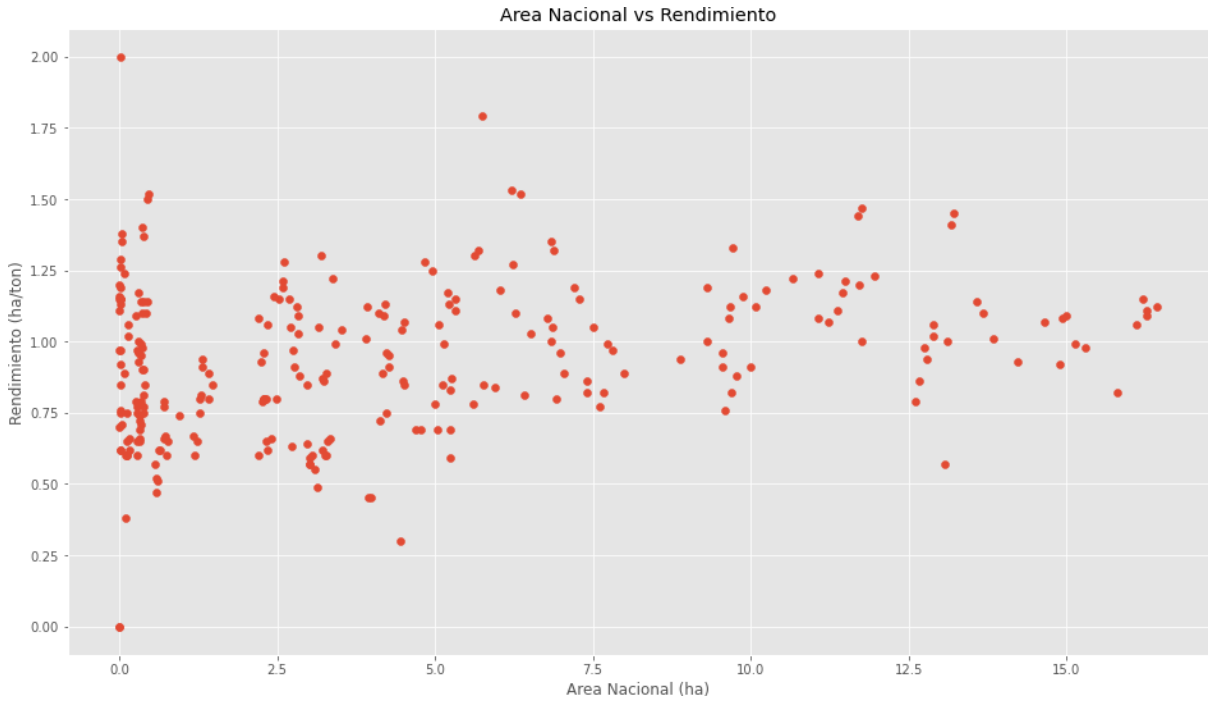
In [98]:
```
# AHORA, VAMOS A CONSTRUIR NUEVOS MODELOS PREDICTIVOS UTILIZANDO EL METODO DE RE
```

In [105]:
```
# Así,de esta manera, ya conociendo el proceso de analitica deterministica, pode  #
Iniciamos el proceso para determinar el modelo de regresion lineal

# Asignamos a nuestra variable de entrada X (En este caso corresponde al Area Na          #
Asignamos a la variable dependiente Y (En este caso corresponde a Rendimiento) dataX
=produccion_df[["Area Nacional (ha)"]]

X_train  =  np.array(dataX)
y_train = produccion_df['Rendimiento (ha/ton)'].values
```

In [106]:

```
# Gráfico de dispersion del comportamiento del Area Nacional versus Produccion N
plt.scatter(produccion_df['Area Nacional (ha)'],produccion_df['Rendimiento (ha/t
plt.title('Area Nacional vs Rendimiento')

plt.xlabel('Area Nacional (ha)')
plt.ylabel("Rendimiento (ha/ton)")
```

Out[106]: Text(0, 0.5, 'Rendimiento (ha/ton)')



In [107]:

```
# Creamos la función objeto para determinar la Regresión Lineal Y= mX+bo

regr = linear_model.LinearRegression()

# Entrenamos nuestro modelo de regresion lineal, con la siguiente función

regr.fit(X_train, y_train)

# Hacemos las predicciones segun el modelo de regresion  lineal

y_pred = regr.predict(X_train)

# Ahora imprimimos los resultados obtenidos

# Vemos el valor de la pendiente, osea la variable m, el coeficiente de la varia

print('Valor de la tangente (m) o Coefficients:=======> ', regr.coef_)
# Ahora el valor de la constante bo, es decir el valor donde la recta corta el e
```

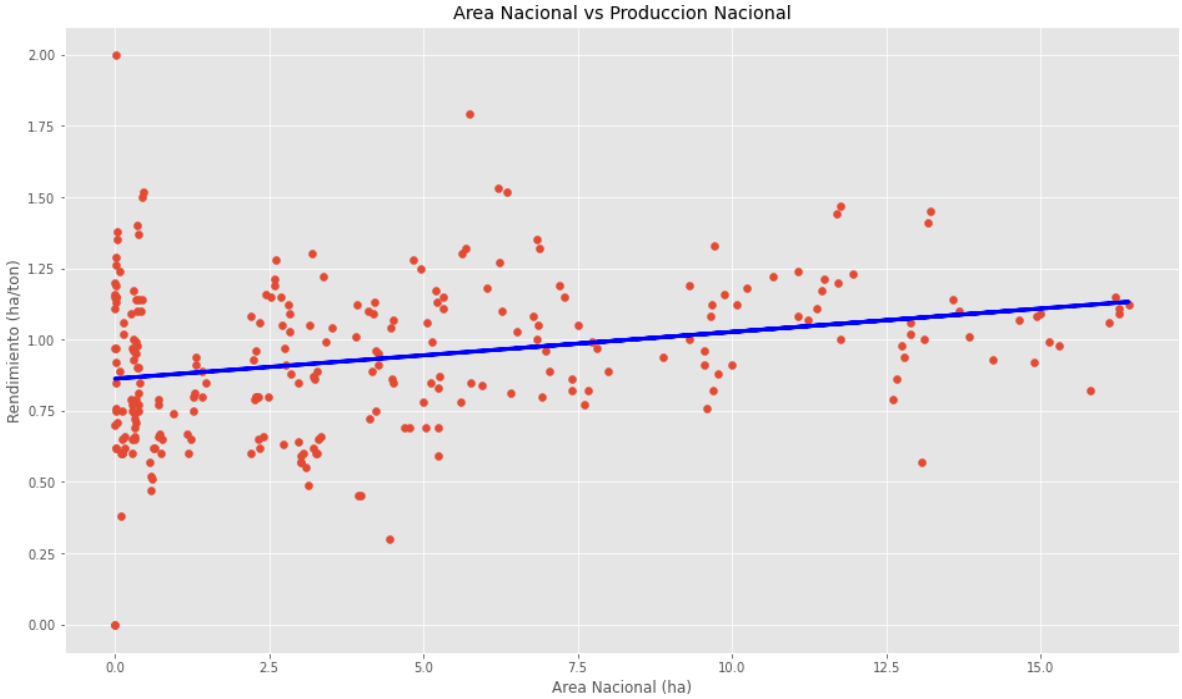
```
Valor de la tangente (m) o Coefficients:=======>  [0.01642121]
Valor de la constante o Independent term: =====> 0.8623491800082033
Error cuadrado medio o Mean squared error:=====>  0.07
valor de la varianza o Variance score:=========> 0.08
```

In [108]:

```
# Gráfico de dispersion del comportamiento del Area Nacional versus Produccion N
plt.scatter(produccion_df['Area Nacional (ha)'],produccion_df['Rendimiento (ha/t
plt.title('Area Nacional vs Produccion Nacional')

plt.xlabel('Area Nacional (ha)')
plt.ylabel("Rendimiento (ha/ton)")

# A continuación se grafica en colo azul, la funcion lineal obtenida a partir de

plt.plot(X_train[: 0], y_pred, color='blue', linewidth=3)
```

In [110]:

```
# Ahora vamos a predecir utilizando la función obtenida, eL RENDIMIENTO (ha/ton)
 # Queremos predecir el rendimiento de la producción nacional de Café vamos  a obt
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[2]])
print('Estimación del rendimiento del Café en (hectareas/toneladas)===>%.3f' %pr
```



Estimación del rendimiento del Café en (hectareas/toneladas)===>0.895

In [111]
:

```
# Ahora vamos a predecir utilizando la función obtenida, eL RENDIMIENTO (ha/ton)
# Queremos predecir el rendimiento de la producción nacional de Café vamos a obt
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[6]])
print('Estimación del rendimiento del Café en
(hectareas/toneladas)===>%.3f' %pr
```

Estimación del rendimiento del Café en (hectareas/toneladas)===>0.961

In [112]:

```
# Ahora vamos a predecir utilizando la función obtenida, eL RENDIMIENTO (ha/ton)
# Queremos predecir el rendimiento de la producción nacional de Café vamos a obt
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[11]])
print('Estimación del rendimiento del Café en
(hectareas/toneladas)===>%.3f' %pr
```

Estimación del rendimiento del Café en (hectareas/toneladas)===>1.043

In [113]:

```
# Ahora vamos a predecir utilizando la función obtenida, eL RENDIMIENTO (ha/ton)
# Queremos predecir el rendimiento de la producción nacional de Café vamos a obt
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[26]])
print('Estimación del rendimiento del Café en (hectareas/toneladas)===>%.3f' %pr
```

Estimación del rendimiento del Café en (hectareas/toneladas)===>1.289

CONCLUSIÓN

Gracias a todos los modelos empleados en el transcurso del proyecto se logra predecir variables importantes en el desarrollo del café a nivel mundial, pero con mayor relevancia en nuestro país, Colombia. Se puede jugar con los datos y moldear sí las soluciones deseadas son o no las indicadas.

A pesar de las grandes rentabilidades que presenta el café Colombiano, se evidencia que por la mala organización hay una disminución en precios, exportaciones, cosechas. Esto se debe a que las plantaciones se enfrentan a plagas, al calentamiento global, el COVID-19, pero más importante a la alta competencia en el mercado, haciendo que sus precios caigan por los pisos durante los últimos años.

BIBLIOGRAFIA

http://www.fao.org/faostat/es/#data

http://sintrainduscafe.org/secciones/en-que-epoca-aplicar-el-fertilizante-en-el-cultivo-del-cafe/

https://federaciondecafeteros.org/wp/estadisticas-cafeteras/

https://cdn.flipsnack.com/widget/v2/widget.html?hash=dpazs597t9

https://www.misfinanzasparainvertir.com/la-crisis-del-cafe-impacta-en-colombia/