**PRÁCTICA FINAL**

**DATA MINING**

# **Índice de Contenidos**

1. Índice de Contenidos · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 1
2. Datos de la Práctica · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 2
3. Objetivo y Fuente de Datos · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 2
4. Análisis de la Variable Objetivo · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 4
5. Análisis del Dataset en Conjunto · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 7
6. Análisis de las Variables Continuas · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 9
7. Análisis de las Variables Discretas · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 10
8. Correlaciones · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 15
9. Modelado · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 17
10. Comparativa de Modelos · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 24
11. Conclusión · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 26
12. Agradecimientos y Comentarios Finales · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · 27

# **Datos de la Práctica**

|  |  |
| --- | --- |
| **Bootcamp IV Big Data & Machine Learning - KeepCoding** | |
| **Asignatura** | Data Mining |
| **Profesora** | Dolores Lorente Muñoz |
| **Alumno** | Francisco Javier Gonzálvez Chico |
| **Título** | Predicción del Precio de la Gasolina 95 en España |

# **Objetivo y Fuente de Datos**

El objetivo de la práctica es abordar un problema de data mining realista: **predecir el precio de la gasolina 95 en España a nivel de provincia desde el punto de vista del consumidor**. Para ello se trabajará con modelización GLM y su comparativa con modelos de regresión y de redes neuronales.

La fuente recoge información actualizada, de forma diaria, de los precios de los combustibles en España –procede del Ministerio de Energía, Turismo y Agenda Digital– y se adjunta en zip.

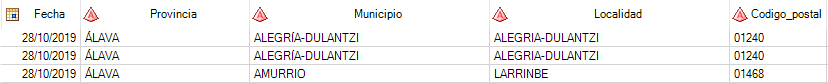
**https://datos.gob.es/es/catalogo/e04990201-precio-de-carburantes-en-las-gasolineras-espanolas**

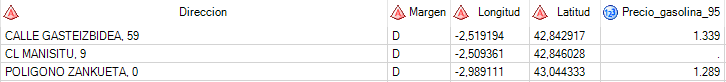
****

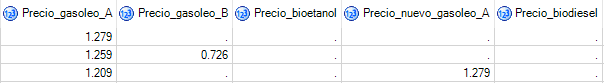
La estructura de registro del dataset es la siguiente:

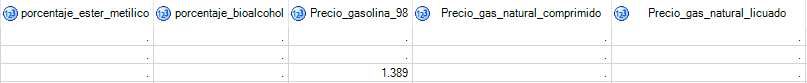
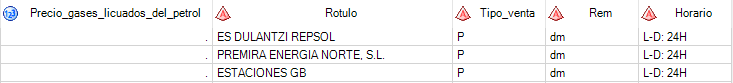
| **VARIABLE** | **TIPO** | **DESCRIPCIÓN** |
| --- | --- | --- |
| fecha | N | Fecha de extracción de la información |
| provincia | α | Provincia donde está ubicada la gasolinera |
| municipio | α | Municipio donde está ubicada la gasolinera |
| localidad | α | Localidad donde está ubicada la gasolinera |
| codigo\_postal | α | Código postal de la ubicación de la gasolinera |
| direccion | α | Dirección de la gasolinera |
| margen | α | **D**: Derecho **|** **I**: Izquierdo **|** **N**: No aplica |
| longitud | α | Coordenada de longitud de la ubicación de la gasolinera |
| latitud | α | Coordenada de latitud de la ubicación de la gasolinera |
| precio\_gasolina\_95 | N | Precio de la gasolina 95 octanos en la gasolinera |
| precio\_gasoleo\_A | N | Precio del gasóleo A en la gasolinera |
| precio\_gasoleo\_B | N | Precio del gasóleo B en la gasolinera |
| precio\_bioetanol | N | Precio del bioetanol en la gasolinera |
| precio\_gasoleo\_A | N | Precio del nuevo gasóleo A en la gasolinera |
| precio\_biodiesel | N | Precio del biodiesel en la gasolinera |
| porcentaje\_ester\_metilico | N | Porcentaje éster metílico en la gasolinera |
| porcentaje\_bioalcohol | N | Porcentaje bioalcohol en la gasolinera |
| precio\_gasolina\_98 | N | Precio de la gasolina 98 octanos en la gasolinera |
| precio\_gas\_natural\_comprimido | N | Precio del gas natural comprimido en la gasolinera |
| precio\_gas\_natural\_licuado | N | Precio del gas natural licuado en la gasolinera |
| precio\_gases\_licuados\_del\_petrol | N | Precio de los gases licuados del petróleo en la gasolinera |
| rotulo | α | Rótulo de la gasolinera |
| tipo\_venta | α | **P**: Venta al público en general  **R**: Venta restringida a socios o cooperativistas |
| rem | α | **OM**: Datos procedentes del operador empresa  **dm**: Datos procedentes del distribuidor minorista |
| horario | α | Horario de atención de la gasolinera |

Presentamos a continuación una imagen que muestra las tres primeras observaciones del dataset cargado en el fichero SAS **gasoline\_in\_spain.sas7bdat**.





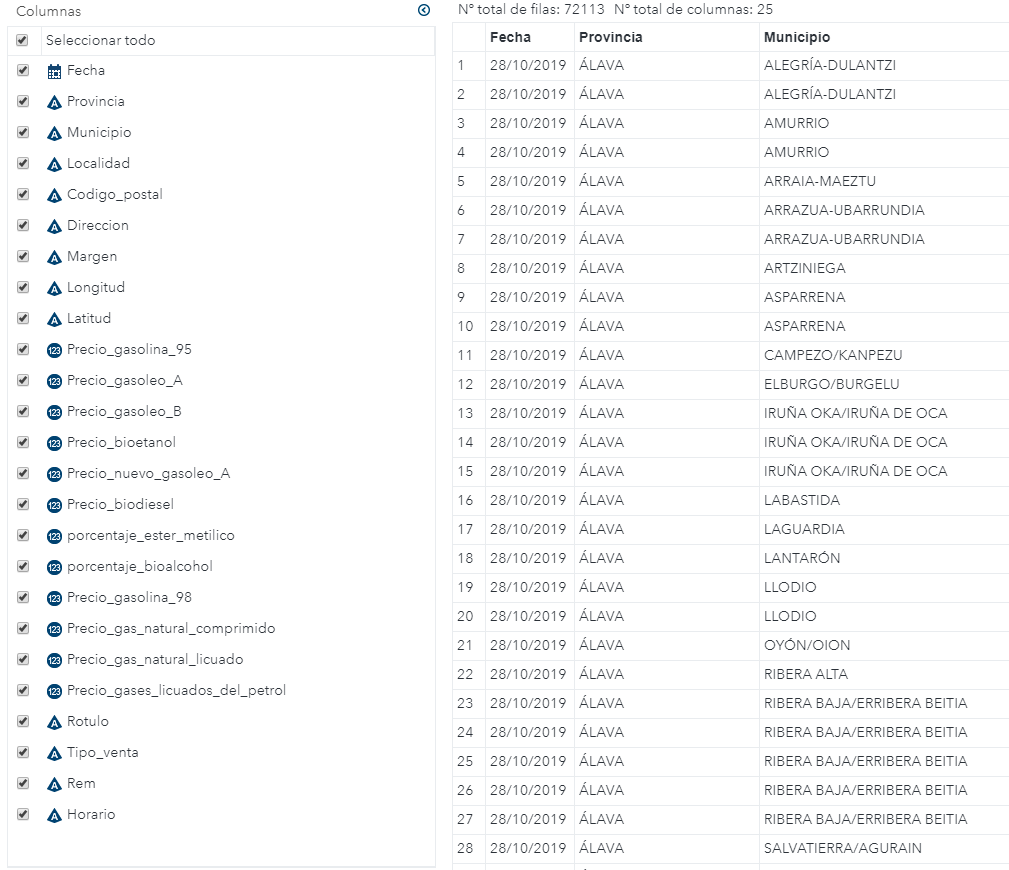




# **Análisis de la Variable Objetivo**

La variable objetivo de nuestro análisis de predicción –nuestro target– será **precio\_gasolina\_95**. Esta columna del dataset es numérica y por tanto continua.

Lo primero que haremos será un análisis estadístico básico para ver que aspecto presentan los valores de nuestra columna objetivo, si existen nulos (missings), que frecuencias abarcan para detectar posibles valores atípicos (outliers) o si siguen alguna distribución.

Para ello, y antes de nada, con un paso **data** cargaremos el fichero en un primer dataset que ubicaremos en la que previamente declaramos como nuestra librería de trabajo, en ella iremos guardando todos los datasets que vayamos generando a medida que avance la realización de la práctica.

libname lib\_work "/home/u44692946/my\_courses/JaviGonzalvez/practice\_DM";

**data** lib\_work.gasolina\_01;

set lib\_work.gasoline\_in\_spain;

**run**;

Observamos un total de 72.113 filas (observaciones) y 25 columnas (variables).

Ejecutamos un procedimiento **means** sobre la variable para obtener unos estadísticos básicos.

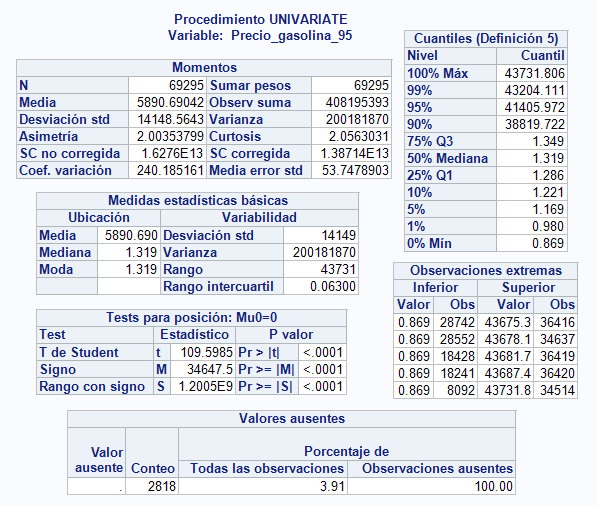
**proc means** data = lib\_work.gasolina\_01;

var precio\_gasolina\_95;

**run**;

Los resultados nos dicen que tenemos 69.295 observaciones en las que nuestra variable objetivo está informada, por tanto deducimos que hay 2.818 registros en los que tiene un valor nulo.

Pero lo que más llama la atención son los valores del intervalo de oscilación **Mínimo-Máximo**. Parece evidentemente exagerado que un litro de gasolina de 95 octanos –ya sea en España o en cualquier sitio del mundo– cueste 0.869€ en una gasolinera y 43.731,81€ en otra. Esto, sumado a que la **Media** y la **Desviación Típica** sean tan altas, nos delata que existen valores atípicos o incongruentes en la muestra.

Ejecutaremos a continuación un procedimiento **univariate** para obtener unos estadísticos más detallados y así detectar esas incongruencias u outliers.

**proc univariate** data = lib\_work.gasolina\_01;

var precio\_gasolina\_95;

**run**;

Con los resultados obtenidos, efectivamente confirmamos los 2.818 valores ausentes y a la vista de la distribución de cuantiles también confirmamos la presencia de –como mínimo– un 10% de valores atípicos en la parte superior de la distribución de la muestra.

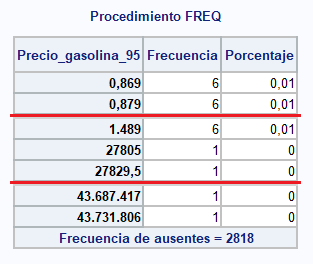
Tanto los missings, como los outliers, nos desvirtúan la distribución correcta de los precios, por tanto tomamos la decisión de eliminarlos.

Para saber con exactitud cuál es el valor umbral, desde el que debemos descartar los valores outliers, ejecutamos un procedimiento **freq**.

**proc freq** data = lib\_work.gasolina\_01;

tables precio\_gasolina\_95;

**run**;

Exportando los resultados a una hoja de cálculo y ordenándolos ascendentemente por nuestra columna en estudio, podemos visionar tres tramos separando sus valores mínimos, sus máximos y la zona central dónde comienzan los valores atípicos.

Como vemos, tras el valor 1,489€ el siguiente salta a 27.805€, por tanto podemos cribar los precios desde ese umbral.

Ejecutamos un paso **data** para el borrado de missings y outliers sobre nuestra variable objetivo y, acto seguido, lanzamos de nuevo un procedimiento **univariate** con gráficos sobre esta primera criba de la muestra para ver cómo nos ha quedado.

**data** lib\_work.gasolina\_02;

set lib\_work.gasolina\_01;

if precio\_gasolina\_95 = **.** or precio\_gasolina\_95 > **1.5** then delete;

**run**;

**proc univariate** data = lib\_work.gasolina\_02;

var precio\_gasolina\_95;

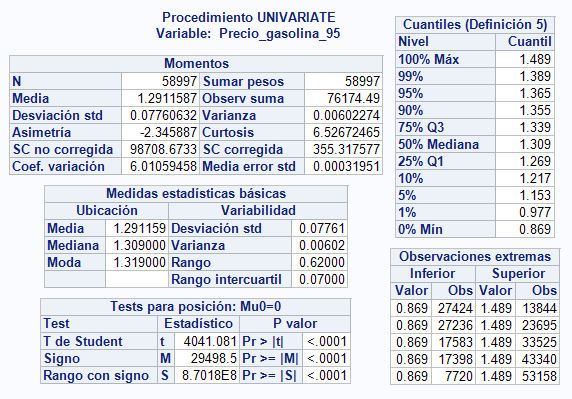
qqplot precio\_gasolina\_95

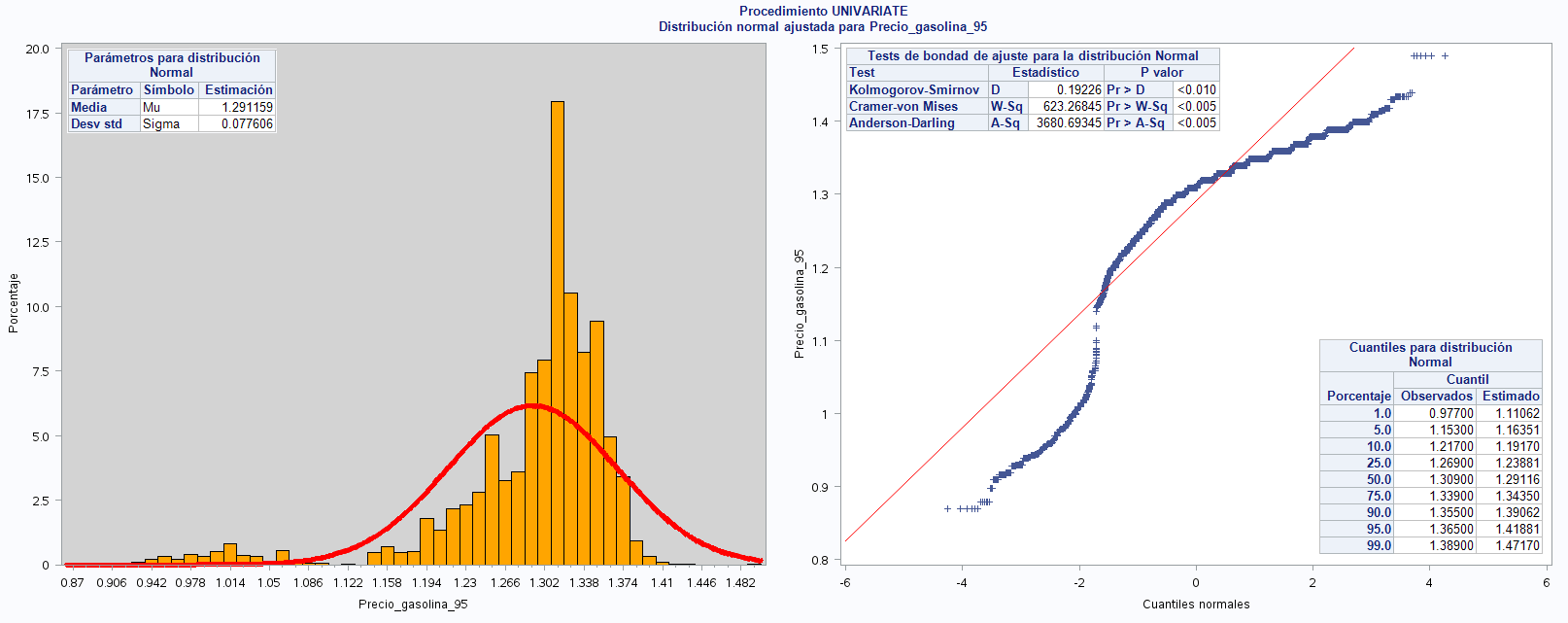
/ normal (mu = est sigma = est color = red l = 1);

histogram

/ normal (color = darkblue w = 4) cfill = orange cframe = lightgrey;

**run**;



Los nuevos resultados ya parecen mucho más correctos. Contamos con 58.997 observaciones informadas, con **Media**, **Mediana** y **Moda** parecidas y **Desviación Típica** y **Varianza** muy bajas.

Las gráficas nos muestran una distribución que parece asemejar cierta normalidad, pero observando sus números, los valores de p-value inferiores a 0,05 lo niegan.

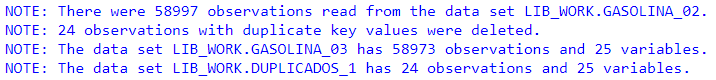
Lo anterior nos parece perfectamente adecuado y concluimos de esta forma el análisis y el data cleansing –o data cooking– de nuestra variable objetivo.

# **Análisis del Dataset en Conjunto**

Nuestro dataset en análisis contiene un conjunto de observaciones que representan gasolineras, dentro del territorio nacional español, con sus características más relevantes en determinados momentos del tiempo (fechas).

Con diferentes fechas tenemos para cada gasolinera, su ubicación geográfica, los precios de los carburantes a la venta, su rótulo e identificación como operador mayorista o distribuidor minorista, su horario y si trabaja a nivel de público en general o sólo para socios cooperativistas.

En definitiva, nuestro dataset ha de tener una clave primaria relacionada con cada gasolinera y evidentemente con la fecha a la que nos proporciona sus datos, según la cual no deberíamos tener observaciones duplicadas a este respecto.

Lo comprobamos con un paso **sort** en el que estableceremos como claves para la ordenación: la fecha de los datos y las coordenadas de geolocalización –longitud y latitud– que nos proporcionan la unicidad de la gasolinera mediante su ubicación geográfica.

**proc sort** nodupkey data = lib\_work.gasolina\_02

out = lib\_work.gasolina\_03

dupout = lib\_work.duplicados\_1;

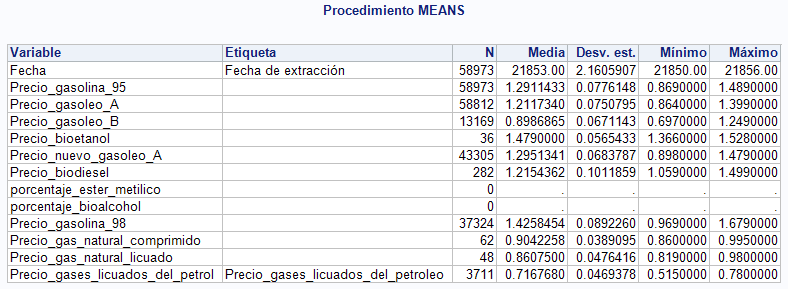
by fecha longitud latitud;

**run**;

Los resultados dicen que se han detectado 24 observaciones con clave duplicada que han sido eliminadas de nuestro dataset de salida (y almacenadas en el dataset duplicados). Por tanto, nuestro trabajo continuo con un dataset que contiene en este momento 58.973 observaciones y 25 variables que vamos a clasificar como categóricas (discretas o dimensiones) y numéricas (continuas o medidas).

|  |  |
| --- | --- |
| **VARIABLES CATEGÓRICAS** | **VARIABLES NUMÉRICAS** |
| fecha | precio\_gasolina\_95 |
| provincia | precio\_gasoleo\_A |
| municipio | precio\_gasoleo\_B |
| localidad | precio\_bioetanol |
| codigo\_postal | precio\_gasoleo\_A |
| direccion | precio\_biodiesel |
| margen | porcentaje\_ester\_metilico |
| longitud | porcentaje\_bioalcohol |
| latitud | precio\_gasolina\_98 |
| rotulo | precio\_gas\_natural\_comprimido |
| tipo\_venta | precio\_gas\_natural\_licuado |
| rem | precio\_gases\_licuados\_del\_petrol |
| horario |  |

# **Análisis de las Variables Continuas**

Para el análisis de las variables numéricas vamos a utilizar un procedimiento **means** sobre ellas.

**proc means** data = lib\_work.gasolina\_03;

var \_mumeric\_;

**run**;

El procedimiento ha tomado la variable **Fecha** como numérica (ya que las variables de tipo temporal pueden clasificarse de esta forma), pero nosotros la analizaremos más adelante junto a las variables discretas. Asimismo, nos ha proporcionado los estadísticos básicos –que ya conocíamos– para nuestra variable objetivo. Ambas variables son las únicas informadas en la totalidad de las 58.973 observaciones de la muestra.

El resto de variables tienen registros con valores nulos y atendiendo a este detalle nos llama la atención que dos de ellas (**porcentaje\_ester\_metilico** y **porcentaje\_bioalcohol**) están totalmente desinformadas, por tanto las eliminaremos de nuestro estudio. Tomamos la misma decisión para otras cuatro variables, cuyo número de ocurrencias informadas consideramos también demasiado bajo (estando por debajo del 5% del número de muestras total) como para ser de alguna manera condicionantes en el modelo (**Precio\_bioetanol**, **Precio\_biodiesel**, **Precio\_gas\_natural\_comprimido** y **Precio\_gas\_natural\_licuado**).

 En las variables que quedan no observamos a priori excentricidades en sus valores de **Media** ni de **Desviación** **Típica**, por lo que presumimos que no existen outliers.

**data** lib\_work.gasolina\_04;

set lib\_work.gasolina\_03;

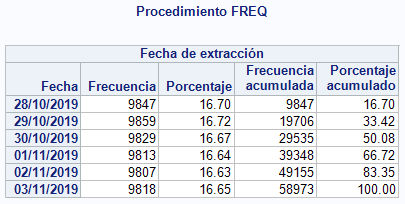
drop porcentaje\_ester\_metilico porcentaje\_bioalcohol

precio\_bioetanol precio\_biodiesel

precio\_gas\_natural\_comprimido precio\_gas\_natural\_licuado;

**run**;

# **Análisis de las Variables Discretas**

La primera variable que queremos observar es la fecha de los datos. Ya hemos visto en el análisis anterior el resultado del procedimiento **means** que nos ha asegurado que está informada en la totalidad de observaciones de la muestra. Vamos a realizar un análisis de frecuencias sobre ella.

**proc freq** data = lib\_work.gasolina\_04;

tables fecha;

**run**;

Podemos ver que tenemos 6 fechas distintas casi correlativas, desde el 28 de octubre al 3 de noviembre de 2019 –faltando el 31 de octubre– con una distribución regular de observaciones oscilando alrededor del 16,67% en cada fecha. Esto nos basta para presumir que la variable está correctamente informada para poder agrupar la información.

A continuación veremos un grupo de variables que ubican geográficamente cada gasolinera en España: **Provincia**, **Municipio**, **Localidad**, **Codigo\_postal**, **Direccion**, **Margen**, **Longitud** y **Latitud**. El principal cometido de las variables discretas en un análisis –además del de identificar las medidas de cada observación– es poder agrupar o clusterizar las variables numéricas. Según lo anterior, quitaremos de nuestro modelo las columnas **Longitud** y **Latitud** ya que identifican de forma puntual cada gasolinera geolocalizándola.

Descartaremos también las variables **Municipio** y **Localidad**, ya que el enunciado de la práctica nos pide una predicción a nivel de provincia, por tanto no necesitamos su granularidad.

De igual modo eliminaremos también la columna Codigo\_postal, ya que este concepto es en base una identificación numérica que condensa la provincia, el municipio y la localidad de cada zona de nuestro país. Como hemos dicho, a nivel geográfico –político– solamente necesitamos la variable **Provincia**.

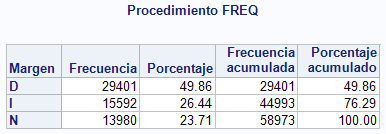
**data** lib\_work.gasolina\_05;

set lib\_work.gasolina\_04;

drop codigo\_postal longitud latitud municipio localidad;

**run**;

La columna **Margen** indica si la gasolinera está en el sentido creciente del kilometraje de una carretera (D) o decreciente (I), correspondiendo el valor ‘no aplica’ (N) si la gasolinera está ubicada en una zona urbana. Veamos con un procedimiento **freq** qué frecuencias nos ofrece.



**proc freq** data = lib\_work.gasolina\_05;

tables margen;

**run**;

La variable está informada en la totalidad de las observaciones de la muestra y parece ser que aproximadamente la mitad de las gasolineras se ubican en el sentido creciente del kilometraje en las carreteras, mientras que algo más de un cuarto lo hacen en el decreciente, quedando el resto ubicadas en zonas urbanas.

Esta variable puede ser interesante para nuestro modelo ya que es posible que el precio del carburante se pudiera ver encarecido o abaratado en función de que los consumidores reposten alejándose, llegando o en el interior de las zonas urbanas más relevantes.

La variable **Direccion**, al igual que las coordenadas de geolocalización, nos proporciona una ubicación puntual y también la eliminaremos de nuestro análisis, pero antes podemos extraer o construir, desde su contenido, otra variable agrupadora. Crearemos una columna **Sitio** para albergar si la gasolinera está en una carretera, un aeropuerto, una avenida, una calle, etc.

**data** lib\_work.gasolina\_06;

set lib\_work.gasolina\_05;

format sitio $20.;

if find(upcase(direccion), "ROPUERTO") then sitio = "AEROPUERTO";

else if find(upcase(direccion), "CENTRO COMERCIAL")

or find(upcase(direccion), "C.C.")

or find(upcase(direccion), "C. C.") then sitio = "CENTRO COMERCIAL";

else if find(upcase(direccion), "POLIGONO")

or find(upcase(direccion), "P.I.")

or find(upcase(direccion), "P. I.")

or find(upcase(direccion), "P.IND")

or find(upcase(direccion), "PG") then sitio = "POLIGONO INDUSTRIAL";

else if find(upcase(direccion), "AUTO")

or find(upcase(direccion), "AUTV")

or find(upcase(direccion), "AU ")

or find(upcase(direccion), "AT ")

or find(upcase(direccion), "A- ")

or find(upcase(direccion), "AP- ") then sitio = "AUTOPISTA-AUTOVIA";

else if find(upcase(direccion), "CARRETERA")

or find(upcase(direccion), "CR ")

or find(upcase(direccion), "CRT")

or find(upcase(direccion), "CTR")

or find(upcase(direccion), "KM") then sitio = "CARRETERA";

else if find(upcase(direccion), "PLAZA")

or find(upcase(direccion), "PZ")

or find(upcase(direccion), "GLORIETA")

or find(upcase(direccion), "ROTONDA") then sitio = "PLAZA";

else if find(upcase(direccion), "AVENIDA")

or find(upcase(direccion), "AVINGUDA")

or find(upcase(direccion), "AV ")

or find(upcase(direccion), "AVGD")

or find(upcase(direccion), "AVD") then sitio = "AVENIDA";

else sitio = "CALLE";

drop Direccion;

**run**;

La asignación anterior se realiza, tras una observación detenida de los valores de la variable **Direccion**, eligiendo ocho categorías representativas, mayoritarias y significativas de la muestra y asignando los valores no correspondientes con ninguna de las igualdades propuestas a la categoría quizá más genérica –“CALLE”–, que abarcará por tanto clasificaciones como “PASEO”, “TRAVESIA”, “RONDA”, “CARRERA”, ”RUA”, “CALLE”, etc.

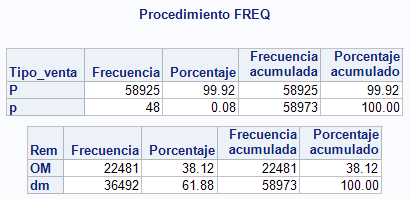
Veamos con un procedimiento **freq** como ha quedado nuestra nueva variable y aprovecharemos para analizar la variable **Provincia**.

**proc freq** data = lib\_work.gasolina\_06;

tables sitio provincia;

**run**;

 Se adjunta solamente el cuadro de resultados de la variable **Sitio** para presentar su aspecto, tras confirmar que la variable **Provincia** contiene las 52 provincias del territorio español y no presenta valores nulos en la muestra.

Veamos a continuación qué nos ofrecen las variables **Tipo\_venta** y **Rem**, ambas de naturaleza dicotómica, la primera indica si la gasolinera vende al público general o restringe la venta a socios cooperativistas, mientras la segunda marca si sus datos los proporciona un operador mayorista o la propia gasolinera.

**proc freq** data = lib\_work.gasolina\_06;

tables tipo\_venta rem;

**run**;

Ninguna de las dos presenta valores nulos en la muestra y mientras que la variable **Rem** parece casi balanceada al 60/40 “dm”/“OM”, observamos que **Tipo\_venta** tiene un único valor “P” o “p” (público general), esta unicidad no nos aporta nada y por ende saldrá descartada de nuestro estudio.



**data** lib\_work.gasolina\_07;

set lib\_work.gasolina\_06;

drop tipo\_venta;

**run**;

Es el turno de la variable **Rotulo** que contiene, en la mayoría de los casos, el rubro del operador mayorista de la gasolinera. El tratamiento que haremos será similar al que implementamos con la columna **Direccion**. Esta columna, como un todo, no nos aporta una posible agrupación, pero si extraemos de ella la marca de la **Empresa** tendremos otra posible variable para clusterización.

**data** lib\_work.gasolina\_08;

set lib\_work.gasolina\_07;

format empresa $20.;

if find(upcase(rotulo), "REPSOL") then empresa = "REPSOL";

else if find(upcase(rotulo), "BP")

or find(upcase(rotulo), "B.P.") then empresa = "BP";

else if find(upcase(rotulo), "CAMPSA") then empresa = "CAMPSA";

else if find(upcase(rotulo), "CEPSA") then empresa = "CEPSA";

else if find(upcase(rotulo), "GALP") then empresa = "GALP";

else if find(upcase(rotulo), "SHELL") then empresa = "SHELL";

else if find(upcase(rotulo), "PETRONOR") then empresa = "PETRONOR";

else if find(upcase(rotulo), "AVIA") then empresa = "AVIA";

else if find(upcase(rotulo), "AGR")

or find(upcase(rotulo), "COO") then empresa = "COOPERATIVAS";

else if find(upcase(rotulo), "ALCAMPO")

or find(upcase(rotulo), "ALIMER")

or find(upcase(rotulo), "AIREMAR")

or find(upcase(rotulo), "BONAREA")

or find(upcase(rotulo), "CARREFOUR")

or find(upcase(rotulo), "CONFORT AUTO")

or find(upcase(rotulo), "LECLERC")

or find(upcase(rotulo), "LOW COST")

or find(upcase(rotulo), "EROSKI")

or find(upcase(rotulo), "SIMPLY")

or find(upcase(rotulo), "SUPECO") then empresa = "SUPERMERCADOS";

else if rem = "OM" then empresa = "OTROS MAYORISTAS";

else empresa = "MINORISTAS";

drop rotulo;

**run**;

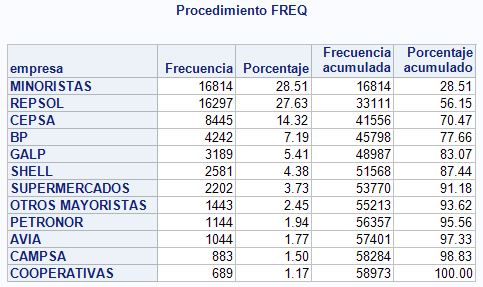
La asignación anterior se realiza eligiendo los ocho mayoristas con más presencia en la muestra, agrupando las gasolineras de cooperativas, agrupando las gasolineras de supermercados y agrupando el resto de las observaciones cuyo **Rem** es igual a “OM” (Operador Mayorista) como “OTROS MAYORISTAS”. Los valores de **Rotulo** que se corresponden con gasolineras no adscritas a mayoristas o resultan inválidos –por no ser lo suficientemente identificativos–, tendrán un valor de **Rem** igual a “dm” (distribuidor minorista) se concentrarán bajo el valor “MINORISTAS”.

**proc freq** data = lib\_work.gasolina\_08 order = freq;

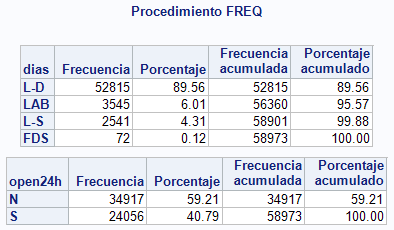
tables empresa;

**run**;

Con el análisis de frecuencias de la variable **Empresa** confirmamos que la columna queda sin valores nulos en la muestra y destacamos que aunque el grupo de “MINORISTAS” sea el más frecuente (con 17.591 registros) agrupa todas las observaciones con porcentaje inferior al 1%.



Nos queda analizar la última variable categórica: **Horario**. Y con ella realizaremos de nuevo un tratamiento de extracción de valores que puedan servirnos como agentes agrupadores. Los valores de esta columna combinan los días de la semana y, en ellos, las franjas horarias en las que una gasolinera está abierta al público. Atendiendo a esto obtendremos dos columnas a partir de esta variable: **Dias** y **Open24h**.

 En la primera reflejaremos aproximadamente el rango de días que la gasolinera abre al público, si detectamos que en el horario aparecen los días lunes (L) y domingo (D) asumiremos que la gasolinera abre todos los días de la semana, si aparecen el lunes (L) y el sábado (S) diremos que la gasolinera sólo cierra los domingos, si aparece el sábado (S) o el domingo (D) que la gasolinera abre los fines de semana y en cualquier otro caso que sólo abre días laborables. La segunda –de naturaleza dicotómica (“S”/“N”)– indicará si la gasolinera permanece abierta las 24 horas o no.

**data** lib\_work.gasolina\_09;

set lib\_work.gasolina\_08;

format dias $3. open24h $1.;

if find(upcase(horario), "L")

and find(upcase(horario), "D") then dias = "L-D";

else if find(upcase(horario), "L")

and find(upcase(horario), "S") then dias = "L-S";

else if find(upcase(horario), "S")

or find(upcase(horario), "D") then dias = "FDS";

else dias = "LAB";

if find(upcase(horario), "24H") then open24h = "S";

else open24h = "N";

drop horario;

**run**;

**proc freq** data = lib\_work.gasolina\_9 order = freq;

tables dias open24h;

**run**;

Con el análisis de frecuencias de ambas variables confirmamos que ninguna de ellas tiene valores nulos, observando que prácticamente el 90% de las gasolineras abren todos los días de la semana y que el horario de apertura 24 horas supera por poco el 40% de la totalidad de nuestra muestra.

# **Correlaciones**

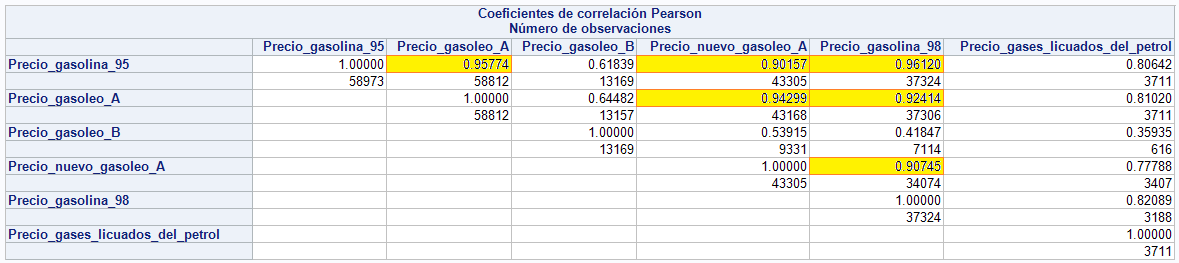
Tras el análisis del conjunto de datos inicial y la creación de nuevas variables a partir de columnas finalmente desechadas, el dataset definitivo cuenta con 58.973 registros u observaciones y 14 campos o variables –8 categóricas y 6 numéricas–.

Es el momento de ver si nuestras variables numéricas están relacionadas, para ello realizaremos un análisis de sus posibles correlaciones mediante un procedimiento **corr**.

**proc corr** data = lib\_work.gasolina\_09 noprob;

var precio\_:;

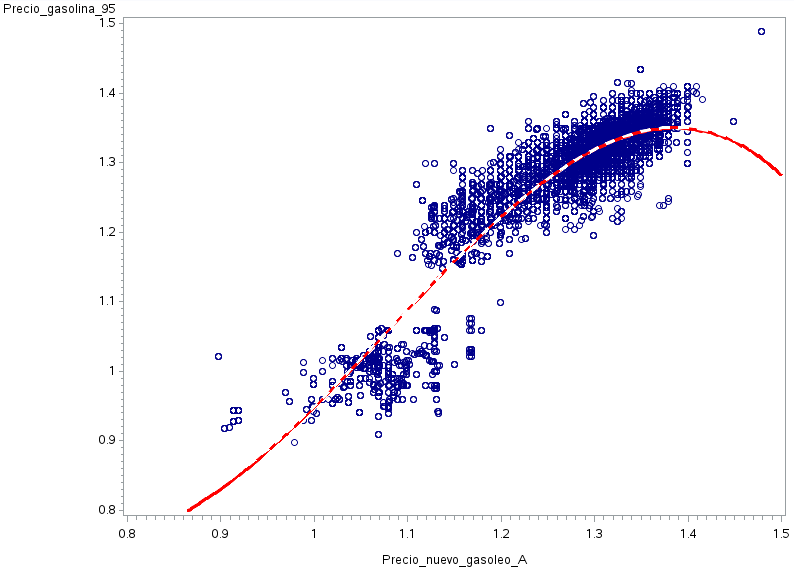
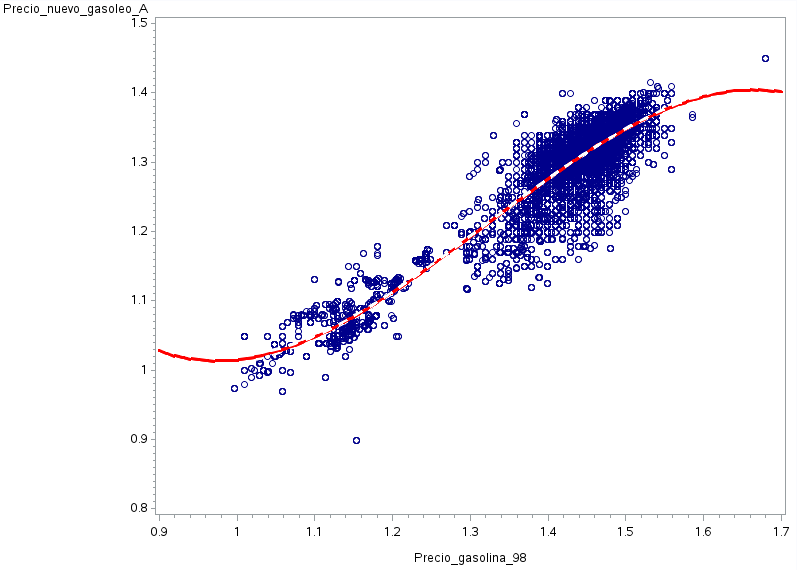
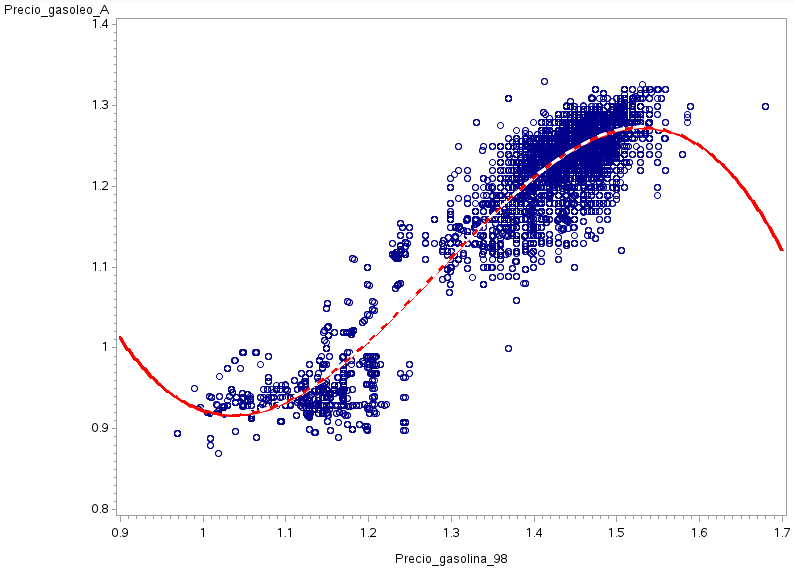
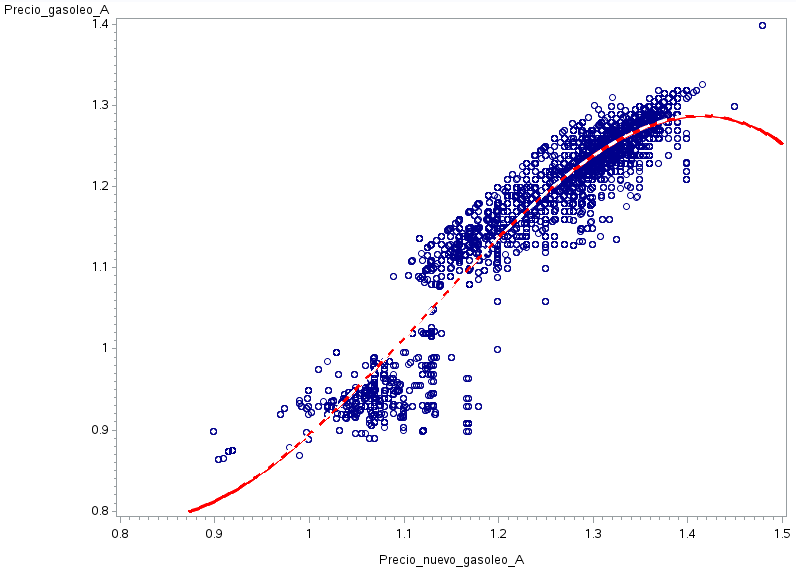
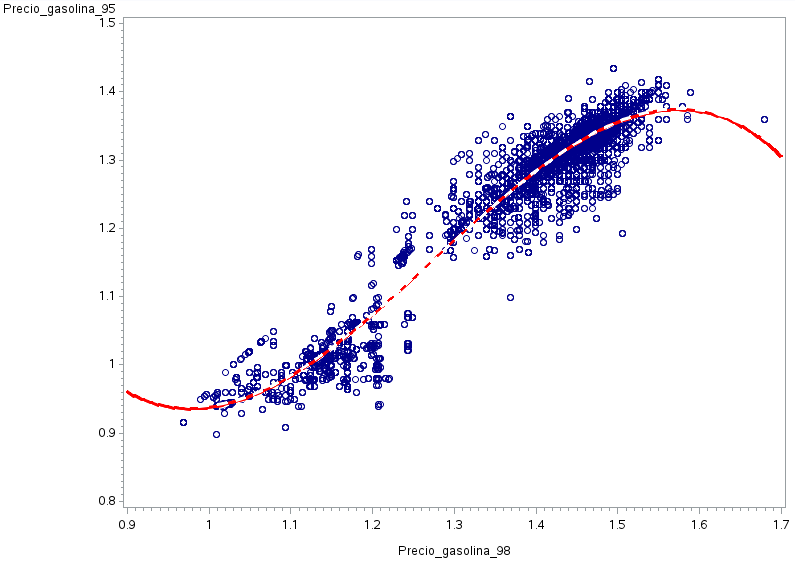
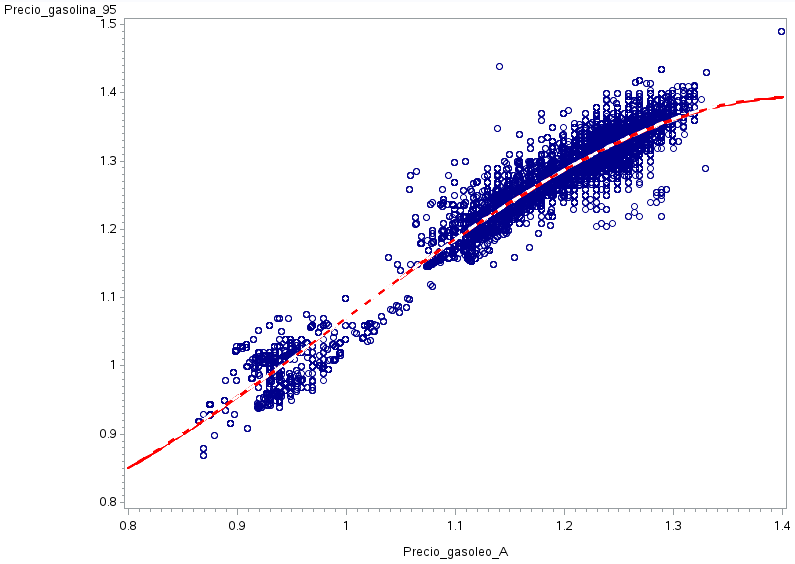
**run**;

La matriz de resultados se presenta únicamente con la parte superior de la diagonal principal informada y con los coeficientes de correlación superiores al 0,85 resaltados para su mejor visualización.

Como puede observarse, aparecen correlaciones fuertes entre varias de nuestras variables, ordenadas de mayor a menor:

* Un 96,12% de correlación entre **Precio\_gasolina\_95** y **Precio\_gasolina\_98**.
* Un 95,77% de correlación entre **Precio\_gasolina\_95** y **Precio\_gasoleo\_A**.
* Un 94,3% de correlación entre **Precio\_gasoleo\_A** y **Precio\_nuevo\_gasoleo\_A**.
* Un 92,41% de correlación entre **Precio\_gasoleo\_A** y **Precio\_ gasolina\_98**.
* Un 90,74% de correlación entre **Precio\_nuevo\_gasoleo\_A** y **Precio\_ gasolina\_98**.
* Un 90,16% de correlación entre **Precio\_ gasolina\_95** y **Precio\_nuevo\_gasoleo\_A**.

Pintemos estas correlaciones para visualizar de mejor manera su interdependencia mediante un procedimiento **gplot**.



symbol1 interpol = rcclm95

value = circle

cv = darkblue

ci = red

width = **2**

co = white;

**proc gplot** data = lib\_work.gasolina\_09;

plot precio\_gasolina\_95\*precio\_gasolina\_98 / regeqn;

plot precio\_gasolina\_95\*precio\_gasoleo\_A / regeqn;

plot precio\_gasoleo\_A\*precio\_nuevo\_gasoleo\_A / regeqn;

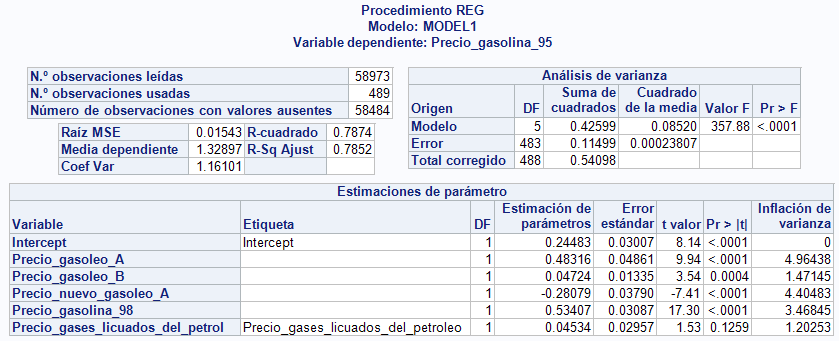
plot precio\_gasoleo\_A\*precio\_gasolina\_98 / regeqn;

plot precio\_nuevo\_gasoleo\_A\*precio\_gasolina\_98 / regeqn;

plot precio\_gasolina\_95\*precio\_nuevo\_gasoleo\_A / regeqn;

**run**;

Las gráficas presentan todas unas regresiones aparentemente rectilíneas en las zonas con mayor número de elementos agrupados, curvándose en los extremos de los ejes debido a la mínima población de esas áreas.

Comprobemos un modelo de regresión para nuestra variable objetivo dependiendo del resto de variables numéricas en el que pediremos el factor de inflación de la varianza (VIF).

**proc reg** data = lib\_work.gasolina\_09;

model precio\_gasolina\_95 = precio\_gasoleo\_A precio\_gasoleo\_B

precio\_nuevo\_gasoleo\_A precio\_gasolina\_98

precio\_gases\_licuados\_del\_petrol / vif;

**run**;

Los resultados nos dicen que no hay ninguna variable con un factor de inflación de varianza superior a 5, por tanto podemos considerar correcta la estimación y también que no existe ningún alto grado de multicolinealidad entre nuestra variable objetivo y el resto, aunque sí están correlacionadas en mayor o menor medida (ya que todos los VIF son superiores a 1).

# **Modelado**

Llegados a este punto, terminados los análisis de todas las variables de la muestra, debemos atender al enunciado de la práctica que literalmente dice: “***predecir el precio de la gasolina 95 en España a nivel de provincia desde el punto de vista del consumidor***”.

Pensemos que como consumidores, podremos saber siempre donde se ubican las gasolineras, si pertenecen o están adscritas a alguna empresa mayorista o sus días y horarios de apertura y atención al público, pero nunca conoceremos a futuro ninguno de sus precios, ni el de nuestro objetivo, ni el del resto de carburantes.

Es por esto que caeríamos en un error si incluyéramos en nuestro análisis de modelado estas variables con las que no podremos contar, independientemente de que los análisis previos hayan demostrado que existe una alta correlación e interdependencia entre todas ellas.

Es el momento de descartarlas de la muestra ya que no deben ser partes implicadas en nuestro modelo final.



**data** lib\_work.gasolina\_10;

set lib\_work.gasolina\_09;

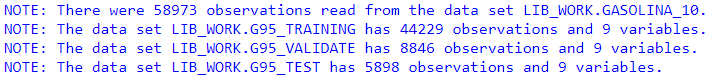
drop precio\_gasoleo\_A precio\_gasoleo\_B precio\_nuevo\_gasoleo\_A

precio\_gasolina\_98 precio\_gases\_licuados\_del\_petrol;

**run**;

Como estamos ante el reto de predecir los valores de una variable de tipo continuo, aplicaremos un primer modelado mediante el algoritmo de un modelo lineal general (GLM).

Particionaremos nuestra muestra –de 58.973 registros– de la siguiente forma:

* Para el entrenamiento del modelo: el 75% de la muestra = 44.229 observaciones.
* Para la validación del modelo: el 15% de la muestra = 8.846 observaciones.
* Para el testeo del modelo: el 10% de la muestra = 5.898 observaciones.

**data** lib\_work.g95\_training lib\_work.g95\_validate lib\_work.g95\_test;

set lib\_work.gasolina\_10;

if \_N\_ <= **44229** then output lib\_work.g95\_training;

else if \_N\_ <= **53075** then output lib\_work.g95\_validate;

else output lib\_work.g95\_test;

**run**;

Para intentar aproximarnos al mejor modelo y poder conocer si nos irá mejor incluyendo sólo variables independientes o incluyendo también interacciones entre ellas, vamos a utilizar el procedimiento de modelado **GLMselect** para generar una gran cantidad de modelos. Una vez los tengamos, seremos capaces de comparar sus frecuencias y coeficientes de error y, en base a estos parámetros, elegiremos el que nos resulte más conveniente.

Vamos a generar en primer lugar, con el algoritmo de modelado **GLMselect**, 250 modelos con método paso a paso (stepwise) con validación normal, sin interdependencias o interacciones entre sus variables, utilizando la partición de datos que hemos establecido para entrenamiento.

Después repetiremos el mismo proceso pero utilizando la validación cruzada.

Tendremos 1.000 modelos repitiendo ambos procesos anteriores pero añadiendo las siguientes interacciones –que consideramos pueden ser las más relevantes– entre las variables para ver si los coeficientes y frecuencias mejoran: **fecha\*provincia fecha\*sitio fecha\*empresa fecha\*dias fecha\*open24h margen\*sitio sitio\*empresa sitio\*dias sitio\*open24h provincia\*empresa provincia\*sitio provincia\*margen empresa\*dias empresa\*open24h dias\*open24h**

Volveremos a generar otros 1.000 modelos utilizando el método hacia atrás (backward), por sus buenos tiempos de ejecución, con lo que conseguiremos doblar el número de muestras.

Para ello escribiremos un procedimiento **macro**, que modificaremos mínimamente para cambiar el método de análisis y de validación y añadir las interacciones, para que se nos generen, en un fichero de salida, los datos de cada ejecución de 250 modelos.

%let lib01='/home/u44692946/my\_courses/JaviGonzalvez/practice\_DM/macroGLMs01.txt';

**%macro** ***macro\_GLMs\_01***;

%do semilla = **12345** %to **12594**;

ods graphics on;

ods output SelectionSummary = modelos;

ods output SelectedEffects = efectos;

ods output Glmselect.SelectedModel.FitStatistics = ajuste;

proc glmselect

data=lib\_work.g95\_training plots = all seed = &semilla.;

partition fraction (validate = **0.4**);

class fecha provincia margen rem sitio empresa dias open24h;

model precio\_gasolina\_95 = fecha provincia margen rem sitio empresa dias

open24h

/ selection = stepwise (select = aic choose = validate)

details = all

stats = all;

run;

ods graphics off;

ods html close;

data union;

i = **12**;

set efectos;

set ajuste point = i;

run;

data \_null\_;

semilla = &semilla.;

file &lib01. mod;

set union;

put effects @**300** nvalue1 @**291** semilla;

run;

%end;

proc sql;

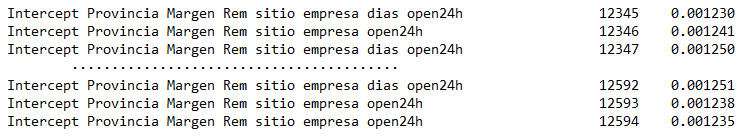
drop table modelos, efectos, ajuste, union;

quit;

**%mend**;

***%macro\_GLMs\_01***;



El resultado de la ejecución de este código son 250 líneas de texto –cada una correspondiente a un modelo–, en las que se ven las variables significativas, su semilla y su coeficiente de error.

El código para la consecución de los primeros 250 modelos, con método de análisis stepwise, validación normal y sin interacciones entre las variables, lo repetiremos, tras su primera ejecución, otras 11 veces, modificando o añadiendo cada vez algunas cosas en las líneas numeradas que se ven sombreadas.

Para la consecución de los siguientes 250 modelos, con método de análisis stepwise, validación normal y con interacciones entre las variables, modificaremos las siguientes líneas del código:

%let lib02='/home/u44692946/my\_courses/JaviGonzalvez/practice\_DM/macroGLMs02.txt';



**%macro** ***macro\_GLMs\_02***;

%do semilla = **12595** %to **12844**;



model precio\_gasolina\_95 = fecha provincia margen rem sitio empresa dias open24h

fecha\*provincia fecha\*rem fecha\*sitio fecha\*empresa fecha\*open24h provincia\*margen

provincia\*sitio margen\*sitio sitio\*empresa sitio\*dias sitio\*open24h empresa\*dias

provincia\*empresa empresa\*open24h dias\*open24h



file &lib02. mod;



***%macro\_GLMs\_02***;

En los siguientes 250 modelos con método de análisis stepwise, cambiamos a validación cruzada y quitamos las interacciones entre las variables. Modificaremos las siguientes líneas del código:

%let lib03='/home/u44692946/my\_courses/JaviGonzalvez/practice\_DM/macroGLMs03.txt';



**%macro** ***macro\_GLMs\_03***;

%do semilla = **12845** %to **13094**;



model precio\_gasolina\_95 = fecha provincia margen rem sitio empresa dias open24h



/ selection = stepwise (select = aic choose = cv)



file &lib03. mod;



***%macro\_GLMs\_03***;

Para los últimos 250 modelos con método de análisis stepwise, seguimos con validación cruzada y ponemos las interacciones entre las variables. Modificaremos las siguientes líneas del código:

%let lib04='/home/u44692946/my\_courses/JaviGonzalvez/practice\_DM/macroGLMs04.txt';



**%macro** ***macro\_GLMs\_04***;

%do semilla = **13095** %to **13344**;



model precio\_gasolina\_95 = fecha provincia margen rem sitio empresa dias open24h

fecha\*provincia fecha\*rem fecha\*sitio fecha\*empresa fecha\*open24h provincia\*margen

provincia\*sitio margen\*sitio sitio\*empresa sitio\*dias sitio\*open24h empresa\*dias

provincia\*empresa empresa\*open24h dias\*open24h



file &lib04. mod;

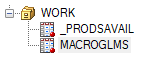


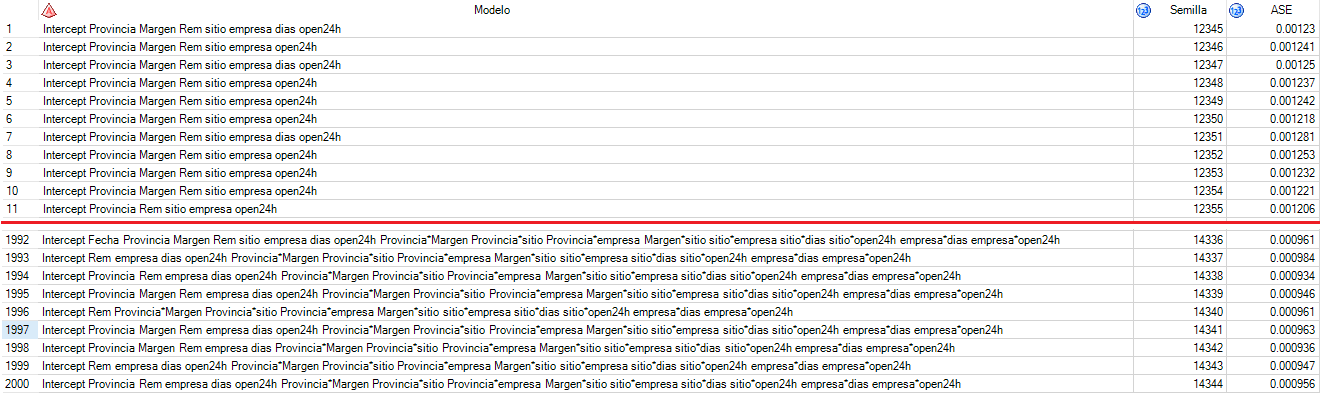
***%macro\_GLMs\_04***;

Los cuatro pasos anteriores se repetirán en sendas macros, en las que seguiremos secuenciando la numeración de la semilla y generando finalmente otros cuatro ficheros de salida correlativos –obteniendo 1.000 modelos adicionales–, cambiando al método de análisis backward, utilizando en dos de ellos validación normal y en los otros dos validación cruzada e intercalando la inclusión o no de las interacciones de las variables.

Una vez generados los 2.000 modelos, tal y como se ha descrito, me parece de rigor comentar que las ejecuciones iterativas del procedimiento **GLMselect** tardan alrededor de pocos minutos cuando no se utilizan las interacciones de las variables. En cambio, cuando se incluyen dichas interacciones el tiempo de ejecución se alarga –a veces– hasta una hora de proceso. Se ha comprobado también que la validación normal cumple con menores tiempos de ejecución que la validación cruzada. En cuanto a los métodos, stepwise y backward tienen tiempos de proceso bastante análogos, mientras que el método forward se vuelve demasiado costoso e inasumible, en consecuencia se decide no utilizarlo, entendiendo que con los 2.000 modelos generados contamos con suficientes para proceder a analizar y encontrar el más adecuado.

Tenemos por tanto ocho ficheros –con 250 registros cada uno– que preferimos unir y ajustar, manualmente en local, para después importarlos desde SAS Guide ayudándonos del asistente.



Una vez creada la tabla MACROGLMS, tras importarla sobre la librería WORK, tiene el siguiente aspecto.

La cargamos de forma definitiva en nuestra librería de trabajo con un paso **data**.

**data** lib\_work.modelos\_glmselect;

set macroglms;

**run**;

Y una vez tenemos nuestros modelos cargados, conformamos a partir de ellos una nueva tabla en la que calcularemos una columna para la frecuencia de cada **Modelo**, dos columnas con los valores mínimo y máximo del **ASE** (el error de predicción) y dos columnas más con la media y la desviación típica de dicho error de predicción. Ordenaremos sus filas por la frecuencia del Modelo para observar cuales son los que se han obtenido más veces en las macros iterativas.

**proc sql**;

create table lib\_work.modelos

as select Modelo, count(modelo) as Count\_Modelo, min(Ase) as Min\_Ase,

max(Ase) as Max\_Ase, avg(Ase) as AVG\_Ase, STD(Ase) as STD\_Ase

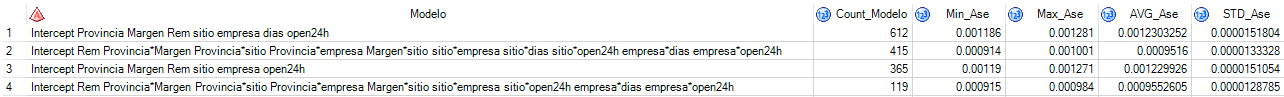
from lib\_work.modelos\_glmselect

group by Modelo

order by Count\_Modelo

desc;

**quit**;

Observando las cuatro primeras filas tenemos los modelos que tienen un frecuencia superior a 100 veces (un 5% de la muestra). Y de entre ellos los dos más interesantes son los situados en las dos primeras filas.

El primero –M1– ha aparecido 612 veces (un 30,6%), es un modelo sin interacciones entre sus variables [**Intercept Provincia Margen Rem sitio empresa dias open24h**] y con un margen de error predictivo bastante bueno, entre 0.001186 y 0.001281.

El segundo –M2– ha aparecido 415 veces (un 20,75%), es un modelo con interacciones entre sus variables [**Intercept Rem Provincia\*Margen Provincia\*sitio Provincia\*empresa Margen\*sitio sitio\*empresa sitio\*dias sitio\*open24h empresa\*dias empresa\*open24h**] y con un margen de error predictivo, que mejora al anterior, entre 0.000914 y 0.001001.

En este momento debemos decidir cual nos parece el mejor de ambos. Veamos pros y contras:

* M1 ha aparecido casi un 10% –200 veces– más que M2
* M1 no tiene interacciones entre sus siete variables, por tanto es un modelo bastante estable que tendrá unos tiempos de ejecución como mínimo aceptables.
* M2 tiene nueve interacciones entre variables lo cual penaliza muchísimo sus tiempos de ejecución.
* M2 mejora 2 diezmilésimas, tanto el mínimo como el máximo, del error predictivo de M1.

En base a lo anterior, nos parece que el mejor de ambos modelos es M1, porque ha sido el iterado más veces, por ser más estable y ligero en ejecución y porque la mejora del margen de error que ofrece M2 no nos parece motivo suficiente como para seleccionarlo frente al resto de comparativas.

Nos quedamos por tanto con el primero de los modelos: M1.

**Precio\_gasolina\_95 = *Intercept* + provincia + margen + rem + sitio + empresa + dias +**

**open24h + *error\_predictivo\_aleatorio***

Veamos la resolución de nuestro recién elegido mejor modelo lineal general aplicando el procedimiento **glm**.

**proc glm** data = lib\_work.g95\_training;

class provincia margen rem sitio empresa dias open24h;

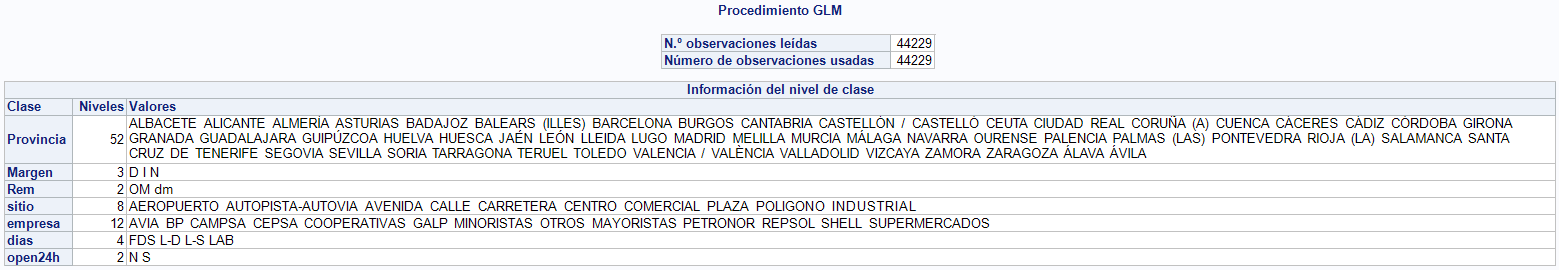
model precio\_gasolina\_95 = provincia margen rem sitio empresa

dias open24h

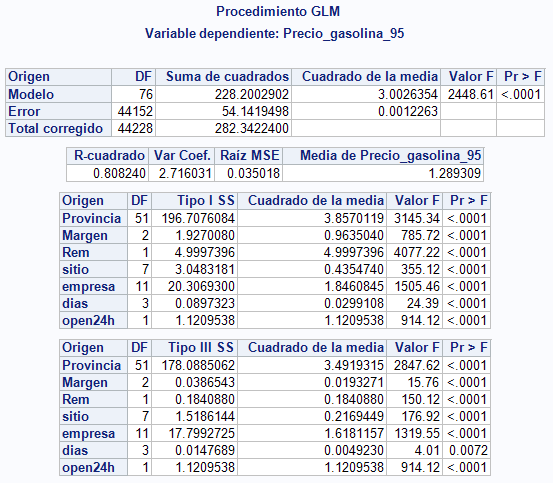
/ solution e;

**run**;

La solución nos presenta primeramente recuento de las observaciones utilizadas y una síntesis de las variables del modelo, en la que podemos apreciar el número y el detalle de las categorías englobadas dentro de cada una de ellas



A continuación podemos ver los coeficientes totales del modelo.

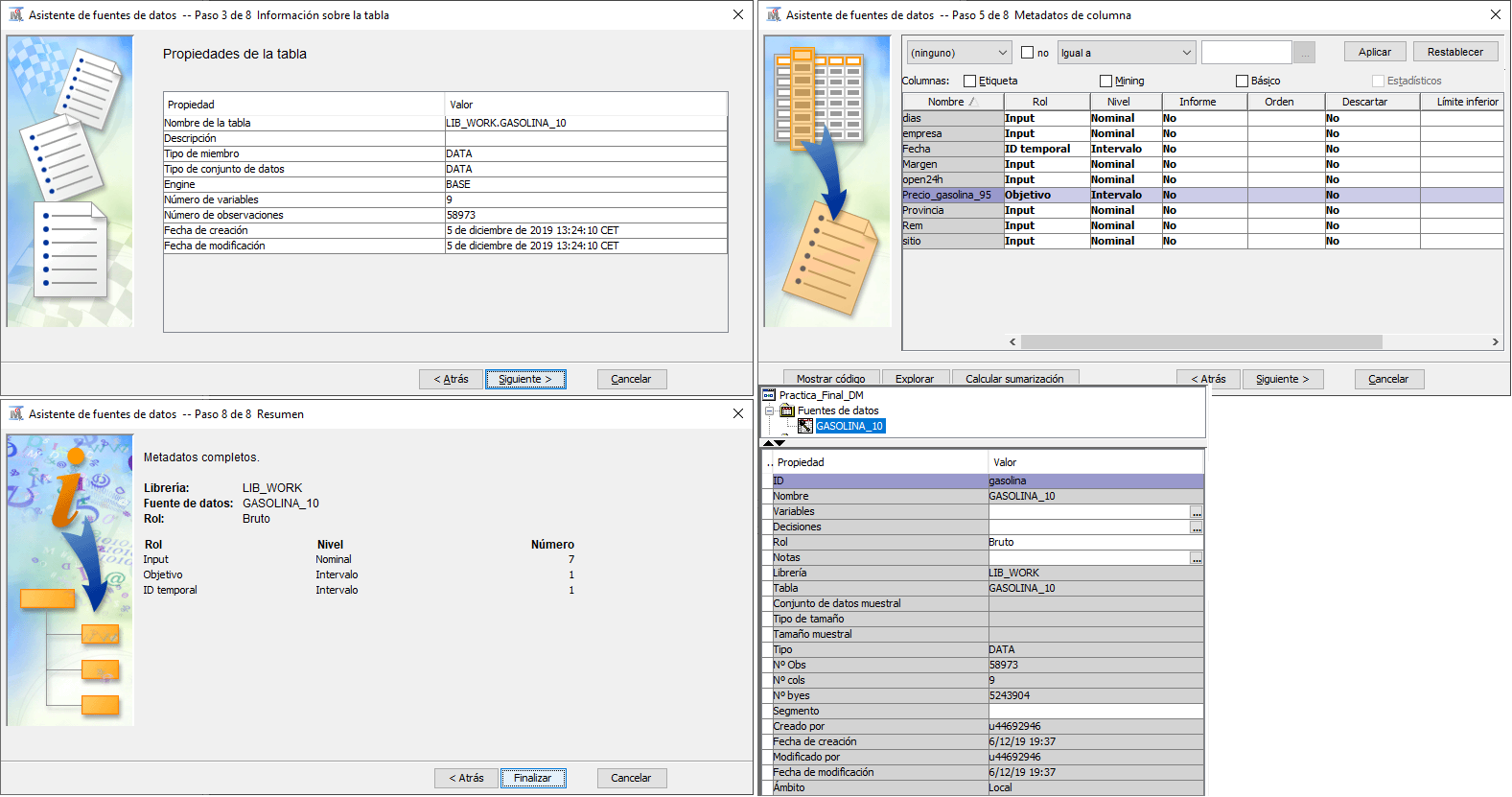


Apreciamos un R-cuadrado o bondad de ajuste del modelo que podemos considerar decente, por superar –por poco, eso sí– el 0,8 y unos p-value todos inferiores a 0,001 (salvo en la variable **días**, aunque también es un valor perfectamente válido por ser menor a 0,05) que nos dicen que nuestros predictores categóricamente son adecuados. A lo que añadimos que es un modelo estable, con tiempos de ejecución presumiblemente bastante aceptables y con un margen de error predictivo muy bajo –del orden de una milésima–.

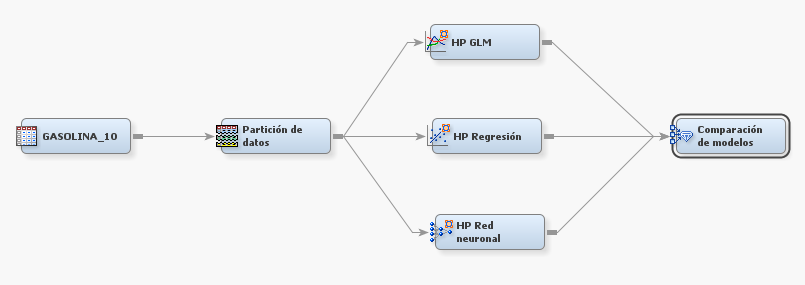
Debemos reseñar que a cada una de las categorías –de cada una de las variables del modelo– se le asocia un coeficiente de significación o estimación para la predicción que buscamos.

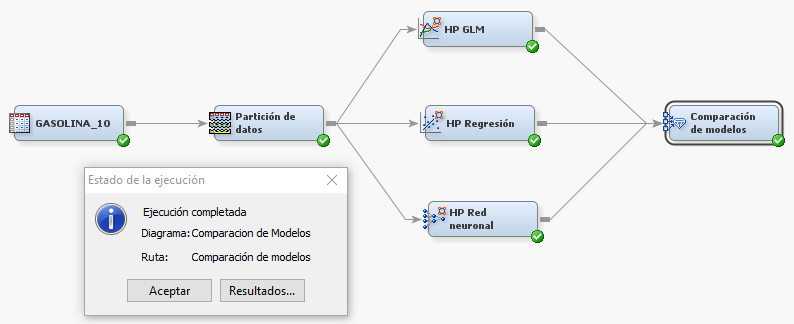
# **Comparativa de Modelos**

Para realizar una comparativa de modelos utilizaremos SAS Miner. Compararemos nuestro modelo GLM con un modelo de regresión y con una red neuronal, utilizando la versión High Performance (HP) para obtener modelos de alto rendimiento.

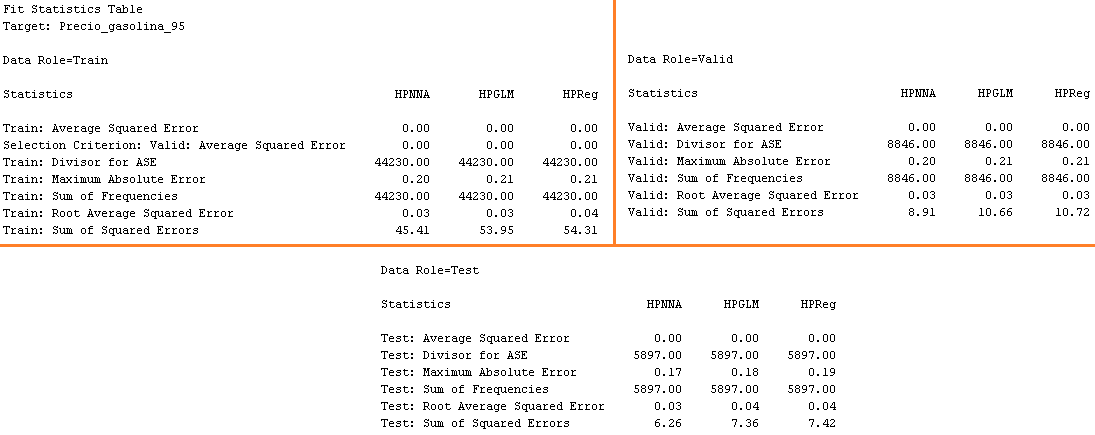
Como pasos previos, creamos un nuevo proyecto en SAS Miner “Practica\_Final\_DM” y ayudándonos de los asistentes, creamos una nueva librería [LIB\_WORK] apuntando a la carpeta en la que hemos ido almacenando todas las tablas SAS que han ido resultando de nuestro trabajo previo *[/home/u44692946/my\_courses/JaviGonzalvez/practice\_DM*] y marcamos una fuente de datos, apuntando a tabla SAS que contiene la muestra con 58.973 registros [gasolina\_10] en la que señalamos nuestra variable objetivo [precio\_gasolina\_95].

Creamos un primer diagrama en el proyecto al que arrastramos nuestra fuente de datos y un elemento de particionado, para poder introducir las mismas particiones que hicimos en el párrafo de modelado.

A la salida del particionado de datos colocamos los tres modelos HP a comparar: un GLM, un modelo de regresión y una red neuronal y unimos sus tres salidas en un comparador de modelos.

Situándonos en el último nodo “Comparación de modelos” y pulsando el botón derecho del ratón seleccionamos ejecutar.

Pulsamos el botón “Resultados” y pasamos a ver que hemos obtenido.

Presentamos las tablas de estadísticos para el Entrenamiento, Validación y Test de los modelos comparados. En ellas se puede observar que la red neuronal es la que tiene mejor (más baja) suma de cuadrados de error. Vemos también que el GLM mejora al modelo de regresión.

De la misma forma se observa que la red neuronal baja una centésima el máximo error absoluto del GLM y que este iguala al modelo de regresión en este aspecto, salvo en Test donde le rebaja una centésima.

En los modelos comparados, por lo que vemos, se otorga una ligerísima ventaja a la red neuronal frente al modelo GLM, quedando el modelo de regresión como el peor de la terna.

Tras analizar la comparativa podemos decir que nuestro modelo GLM elegido es bueno y sería implementable, sobre todo porque aunque el modelo de la red neuronal presente esa ligera ventaja predictiva, de seguro tendrá tiempos de ejecución más largos (peores o más pesados) y carecerá de versión explicativa.

# **Conclusión**

Finalmente recomendamos la elección del modelo GLM por su estabilidad, buenos tiempos de ejecución y más que aceptables márgenes de error predictivo

**Precio\_gasolina\_95 = *Intercept* + provincia + margen + rem + sitio + empresa + dias +**

**open24h + *error\_predictivo\_aleatorio***

Con los estimadores obtenidos –para sus siete variables, con sus categorías–, como el mejor modelo para predecir el precio de la gasolina 95 en España.

# **Agradecimientos y Comentarios Finales**

Como nota final de este trabajo me gustaría dar las gracias a nuestra profesora, **Dolores Lorente**, por su buen trabajo y gestión, por su gran aporte de documentación y por su generosa ayuda y la perfecta atención a la hora de corregir nuestros errores y despejar nuestras dudas (durante las clases y en Slack).

Sobre esta –mi práctica final– quiero decir que, atendiendo explícitamente al enunciado, sé que podría haber obviado algunas partes, como el análisis particular de las variables continuas (salvo la objetivo) o el de sus correlaciones, dado que los precios de los carburantes no iban a entrar en la parte final de modelado. Pero decidí realizar y plasmar esos ejercicios principalmente por el interés de practicar y también siendo sincero para poder presentar un trabajo más completo.

Comentar que para mí, la asignatura de DataMining, ha sido realmente difícil. La he encontrado muy densa y profunda y creo que requiere de unos conocimientos de estadística más que básicos y sobre todo bien asentados. La curva de aprendizaje, para poder llegar a manejar con soltura los procedimientos de análisis y modelado de SAS es bastante pronunciada, bastante más alta que la de la media –entiendo que, como en todo, la experiencia es un grado y aquí particularmente pienso que cuenta muchísimo– sobre todo a la hora interpretar y decidir sobre los análisis y modelos estadísticos que se realizan.

Sin más, mi agradecimiento por las cosas que he aprendido, que espero se vean reflejadas en esta práctica.

**F. Javier Gonzálvez**