UNIVERSIDAD
COMPLUTENSE
MADRID

Proyecto de Innovación

Convocatoria 2019/2020

Nº de proyecto:

343

Título del proyecto:

Tutoriales guiados de prácticas en “Estadística: Análisis de Datos e Inferencia” mediante el software libre SAS University Edition

Nombre del responsable del proyecto:

Nirian Martín Apaolaza

Centros:

Facultad de Comercio y Turismo, Facultad de Ciencias Matemáticas

Departamentos:

Departamento de Economía Financiera y Actuarial y Estadística, Departamento de Estadística e Investigación Operativa, Departamento de Álgebra, Geometría y Topología

Contenido

1. Objetivos propuestos en la presentación del proyecto

El objetivo principal de este proyecto era el de la elaboración de material docente para las asignaturas con contenido de Inferencia Estadística y Análisis de Datos, de las Facultades de Ciencias Matemáticas y de Ciencias Económicas y Empresariales; como, por ejemplo, la asignatura de Análisis de Datos que se imparte en la Facultad de Ciencias Matemáticas, en el tercer curso del grado de Matemáticas y Estadística.

En concreto, se pretendía la elaboración de un manual para aprender a utilizar el software estadístico SAS Studio. Si bien SAS es un potente programa de pago, ofrece una versión gratuita (orientada principalmente a los estudiantes) denominada SAS University. SAS University es un software libre o de uso no comercial, que facilita al alumnado la fácil accesibilidad al programa desde cualquier ordenador.

La propuesta consistía en elaborar material docente mediante Jupiter Notebooks, una herramienta que permite incluir código de forma interactiva, resultados del código y elementos de texto descriptivos. Esta herramienta no es exclusiva de SAS University, de hecho, inicialmente estaba pensada para código en R y se ha ido adaptando a muchos otros lenguajes de programación, siendo mucho menos extendido, de momento, su uso para el aprendizaje de SAS.

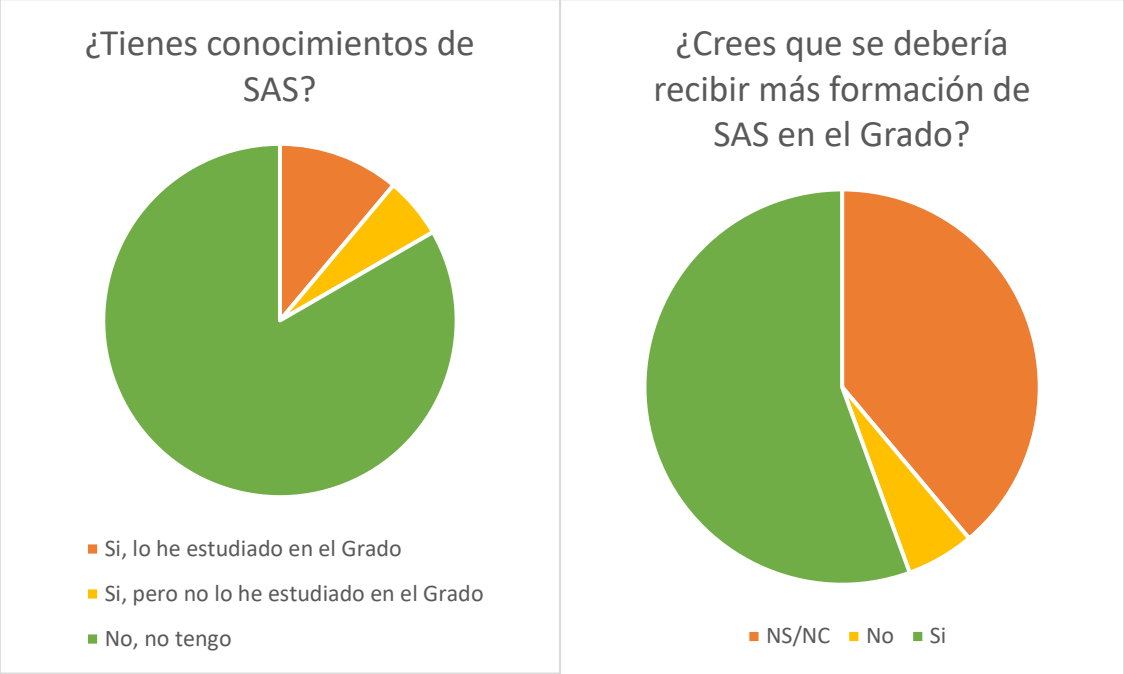
Así pues, los objetivos principales eran que:

1. El alumnado y el profesorado trabajen en un entorno atractivo, como suelen ser los programas a través de paneles como SPSS.
2. El alumnado refuerce sus nuevos conocimientos en Estadística.
3. El alumnado y el profesorado no perciban los comandos del software como una complicación añadida a la propia asignatura.

2. Objetivos alcanzados

Pese a la clara necesidad en las grandes empresas del uso de Software estadístico (que se puede comprobar en la descripción de la mayoría de las prácticas ofertadas) la mayoría de los alumnos de la Facultad de Ciencias Matemáticas carecen de formación en el uso de software estadístico. En particular, no hay ninguna asignatura obligatoria en el grado de Matemáticas cuyo plan de estudios prevea su aprendizaje; y en los grados de Matemáticas-Estadística y doble grado de Economía y Matemáticas-Estadística, el uso de software estadístico no se imparte hasta tercer y cuarto curso, respectivamente.

Se consideró pues, que el primer paso era el detectar de forma detallada las carencias de los alumnos y su conocimiento de SAS, así como la percepción que tenían de éste. Así pues, se realizó una encuesta a los alumnos de la asignatura Análisis de Datos, del tercer curso del grado Matemáticas-Estadística de la Facultad de CC. Matemáticas. La encuesta se realizó por medio del Campus Virtual de la asignatura. Mostramos las respuestas de la encuesta a dos de las preguntas clave.



Efectivamente, como se puede observar en el gráfico de la izquierda, los alumnos encuestados no tienen en general conocimientos sobre SAS. Por otro lado, más de la mitad de los alumnos encuestados consideran que se debería recibir más información en el grado. Además, a la pregunta “En caso de seguir un manual, ¿prefieres que sea

interactivo?", dos de cada tres alumnos contestan afirmativamente. Todas las preguntas y resultados de la encuesta se pueden consultar en la Sección 6.

Parecía claro la necesidad de cubrir esta carencia. Para ello se desarrolló un pequeño manual de análisis de datos con SAS Studio mediante la herramienta Jupiter Lab. Este manual, que se seguirá desarrollando, se centra inicialmente en los siguientes aspectos.

---

1. **Introducción a SAS y Creación de Librerías en SAS**: se aporta una pequeña introducción que pretende no ser demasiado técnica pero sí útil.
2. **Importación de los Datos**: es importante tener en cuenta que para su mayor aplicabilidad los alumnos deben saber trabajar con todo tipo de ficheros de datos, no sólo datos SAS. En este caso, se enseña a importar datos Excel, siendo extensible a cualquier tipo de archivo.
3. **Tratamiento de datos Missing**: se dan algunas directrices para la detección, corrección e imputación de datos missing.
4. **Análisis descriptivo de los datos**: herramientas básicas para el análisis tabular y gráfico de datos univariantes y multivariantes.

Todo esto se ha hecho en base a el conjunto de datos creado por Gregory Lee para su libro "Business Statistics Made Easy in SAS", disponible para usuarios SAS en la plataforma SASSuport y que representa de manera muy realista una compañía ficticia dedicada a la creación y mantenimiento de un software de contabilidad.

El material docente elaborado se ha realizado en castellano. En las secciones 3, 5 y 6, se detallará con más precisión el trabajo realizado en esta herramienta. Si bien se pretende seguir ampliando el material, entendemos que el material está listo para su uso en cualquiera de las asignaturas con contenido de Inferencia Estadística y/o Análisis de Datos.

Así, los objetivos logrados se podrían resumir en los siguientes puntos

- Detectar las carencias del alumnado en el uso del Software Estadístico SAS.
- Crear material práctico e innovador que facilite el aprendizaje de SAS, creando ejemplos prácticos, mediante JupiterLab, y que se pueda enseñar en las asignaturas de Estadística y Análisis de Datos.

En el Anexo 6 se presenta el manual realizado, el cuál puede encontrarse, junto con el resto de material necesario para su ejecución en el siguiente enlace

https://sites.google.com/view/elenacastillagonzlez/home/research/research-projects

3. Metodología empleada en el proyecto

Para alcanzar los objetivos descritos en las Secciones 1 y 2 se llevaron a cabo las siguientes actuaciones:

1. **Primera Fase**. Detección de las necesidades específicas de aprendizaje en los alumnos. Para ello se llevó a cabo la encuesta comentada en la Sección 6.
2. **Segunda Fase**. Diseño del curso de "JupyterLab". Se tuvieron en cuenta las conclusiones obtenidas en la primera fase, y se tomaron decisiones como:
    - Idioma del curso. Se decidió castellano dado que la docencia en las asignaturas de Estadística de la UCM se imparte principalmente en este idioma.
    - Contenidos a introducir. Como una primera aproximación, se tratan los temas más básicos: lectura de datos, tratamiento de datos missing y análisis descriptivo de los datos.
    - Dificultad y estructura de los ejemplos. Se concluyó que no era necesario para el aprendizaje que los ejemplos tuvieran una dificultad excesivamente elevada. En cambio, era necesario que fuesen muy ilustrativos. Por ello, se eligió un conjunto de datos aplicado a la empresa y bastante realista.
3. **Tercera Fase**. Programación del manual de SAS en "JupyterLab".

Queda pendiente la **Cuarta Fase**, que sería la implementación del curso programado en asignaturas de los diferentes grados donde los miembros del proyecto dan docencia.

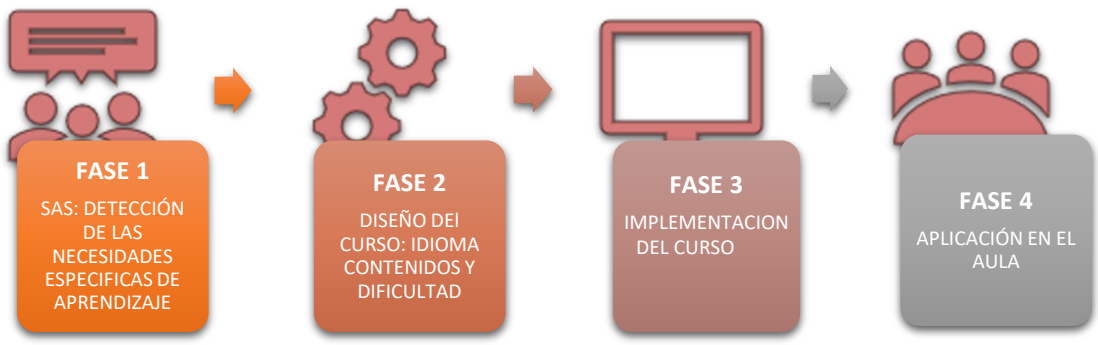


Figura 1: Esquema representativo de las fases seguidas en la realización del proyecto.

Se pretende comenzar al comienzo del curso 2020/2021.Las fases citadas se han desarrollado mediante reuniones de los profesores participantes y trabajo personal de cada uno de los miembros.

4. Recursos humanos

El presente proyecto, propuesto y coordinado por la Facultad de Comercio y Turismo, se trata de un proyecto interfacultativo e interdepartamental. En particular los centros y departamentos implicados son

- Facultad de Ciencias Matemáticas (Departamento de Estadística e Investigación Operativa y Departamento de Álgebra, Geometría y Topología);
- Facultad de Ciencias Económicas y Empresariales (Departamento de Economía Financiera y Actuarial y Estadística).

Descripción de cargo y facultado a la que pertenecen los miembros del proyecto:

1. **Nirian Martín Apaolaza**. Profesora Titular del Departamento de Economía Financiera y Actuarial y Estadística (responsable del proyecto).
2. **Elena María Castilla González**. Contratada predoctoral FPU del Departamento de Estadística e Investigación Operativa.
3. **Pedro José Chocano Feito**. Contratado predoctoral FPI del Departamento de Álgebra, Geometría y Topología.
4. **María Jaenada Malagón**. Estudiante del máster TECI y becaria de colaboración, del Departamento de Estadística e Investigación Operativa.
5. **Leandro Pardo Llorente**. Profesor Catedrático del Departamento de Estadística e Investigación Operativa.

Los miembros PDI del proyecto tienen una amplia experiencia en la docencia de asignaturas con contenidos de Análisis de Datos e Inferencia Estadística en los grados de Matemáticas, Ingeniería Matemática, Matemáticas y Estadística y doble grado de Economía y Matemáticas y Estadística, todos ellos en la Facultad de Ciencias Matemáticas; así como en el grado de Comercio y Turismo.

De hecho, varios de los miembros del equipo de trabajo formaron anteriormente parte de otro Proyecto de Innovación Docente: “Tutorial interactivo de ejemplos básicos y ejercicios de Inferencia Estadística No-paramétrica mediante software libre: implementación mediante R, R-studio y Swirl” (curso 2018/2019).

5. Desarrollo de las actividades

En primer lugar, es necesaria la **instalación de SAS University Edition**. Esto se hará a través de la página web habilitada por SAS.

https://www.sas.com/en_us/software/university-edition/download-software.html

- Elegir sistema operativo del usuario: este software está adaptado para Windows, Mac y Linux. Una vez elegido el sistema operativo correspondiente, SAS nos instará a asegurarnos de que tenemos una de las últimas versiones y un navegador de internet adecuado (los clásicos, Firefox y Google Chrome son compatibles).
- Seguir los siguientes cuatro pasos
    1. Creación de una máquina virtual correctamente configurada: quizás el paso más difícil, es recomendable seguir todas las instrucciones detalladas en la citada página web.
    2. Descarga de SAS University Edition
    3. Configuración del Software en la máquina virtual
    4. Uso del Software

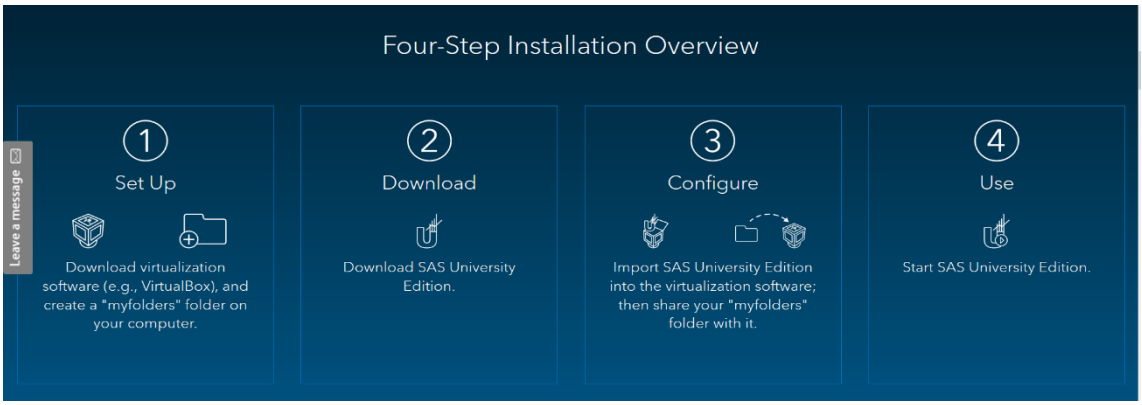


Figura 2: Instalación de SAS University Estudio. Fuente https://www.sas.com/en_us/software/university-edition/download-software.html

Una vez realizado el proceso de instalación, ya se puede empezar a utilizar las herramientas de SAS Studio y Jupyter Lab.

Se recomienda explicar este proceso en clase, ya que puede resultar un poco complicado para aquellos alumnos con pocos conocimientos informáticos.

6. Anexo

6.1. Encuestas

Respuesta número	Por favor, indica tu sexo	¿Qué software (estadístico) manejas o has usado con más frecuencia? (Selecciona hasta 3)	Cuando trabajas con un nuevo software ¿Consideras útil tener un manual?	En caso de seguir un manual, ¿prefieres que sea interactivo?	¿Es importante que los ejemplos de los manuales sean realistas?	¿Tienes conocimientos de SAS?	En caso afrimativo. ¿Cuál es tu nivel de SAS?	¿Crees que se debería recibir más formación de SAS en el Grado?
Respuesta número: 1	Hombre	R	Si	Si	Me es indiferente	No, no tengo		NS/NC
Respuesta número: 2	Hombre	R Python Matlab	Si	No	Muy importante	No, no tengo		Si
Respuesta número: 3	Hombre	R SPSS Python	Si	Si	Muy importante	No, no tengo	Bajo	NS/NC
Respuesta número: 4	Hombre	R SPSS Python	Si	Si	Muy importante	No, no tengo		Si
Respuesta número: 5	Mujer	R SPSS Python Matlab	Si	No	Muy importante	No, no tengo		Si
Respuesta número: 6	Hombre	R SPSS Python	Si	No	Muy importante	No, no tengo		Si
Respuesta número: 7	Hombre	R Python Matlab	Si	Si	Me es indiferente	No, no tengo		NS/NC
Respuesta número: 8	Mujer	R Excel	Si	Si	Muy importante	Si, lo he estudiado en el Grado	Bajo	Si
Respuesta número: 9	Mujer	SPSS Python Matlab	Si	Si	Muy importante	No, no tengo		Si
Respuesta número: 10	Hombre	R Python Matlab	Si	Si	Muy importante	No, no tengo		NS/NC
Respuesta número: 11	Mujer	R SPSS Python Matlab	Si	No	Muy importante	Si, pero no lo he estudiado en el Grado	Bajo	No
Respuesta número: 12	Hombre	Python	Si	Si	Muy importante	No, no tengo		NS/NC
Respuesta número: 13	Mujer	R Python Excel	Si	No	Me es indiferente	No, no tengo		Si
Respuesta número: 14	Hombre	R SPSS Python	Si	No	Muy importante	Si, lo he estudiado en el Grado	Bajo	Si
Respuesta número: 15	Mujer	Statgraphics R SPSS Python Matlab	Si	Si	Muy importante	No, no tengo		Si
Respuesta número: 16	Mujer	R SPSS Excel	Si	Si	Muy importante	No, no tengo		NS/NC
Respuesta número: 17	Mujer	R SPSS Matlab	Si	Si	Muy importante	No, no tengo		Si
Respuesta número: 18	Hombre	R SPSS Python	No	Si	Me es indiferente	No, no tengo		NS/NC

6.2. Tutoriales guiados

Tutoriales guiados de prácticas en "Estadística, Análisis de Datos e Inferencia" mediante el sofware libre SAS University Edition.

Elena Castilla, Pedro J. Chocano, María Jaenada, Nirian Martín y Leandro Pardo

Introducción

¿Qué es el análisis de datos?

El análisis de datos o "Data Science" es la ciencia que se encarga de examinar un conjunto de datos con el propósito de sacar conclusiones sobre la información que recogen. Las técnicas estadísticas son las herramientas que se utilizan para extraer esta información subyacente en los datos de forma que sea fácilmente interpretable.

Las grandes empresas manejan enormes volúmenes de datos recogidos de distintas fuentes. Analizándolos adecuadamente, consiguen tomar las mejores decisiones para su negocio, y sacar el máximo rendimiento a sus recursos.

Pasos en el Análisis de datos

El análisis de datos se compone de varias etapas. En general, debe seguirse el siguiente esquema:

1. Planteamiento del problema: en esta etapa se define el foco de estudio, la información que pretende obtenerse de los datos.
2. El muestreo o recolección de datos: para poder dar solución al problema planteado, pueden ser necesarios unos datos concretos, que deben recogerse mediante encuestas o experimentos, o nutrirse de datos ya recolectados por la empresa.
3. La limpieza de datos: una vez se tienen las tablas de datos, deben tratarse para asegurarse de que no haya datos faltantes (missings data), datos atípicos debidos a errores en la recolección (outliers), observaciones repetidas
4. Análisis descriptivo: antes de comenzar a extraer la información que se pretende en el problema, es conveniente hacer un análisis de los datos que tenemos para poder entender mejor su naturaleza, por ejemplo, mediante gráficos.
5. El análisis: en esta etapa se aplican las técnicas apropiadas de análisis para obtener la información que necesitamos. Estos análisis se ayudan de la inferencia estadística, que pretende entender las características de un conjunto grande (población) a través de la información de un conjunto pequeño (muestra). Las técnicas más populares son el contraste de hipótesis, predicción, clasificación..etc.
6. Interpretación: por último, deben interpretarse los resultados obtenidos para obtener conclusiones.

Presentación de los datos

En este trabajo se va utilizar un conjunto el conjunto de datos reado por Gregory Lee para su libro "Business Statistics Made Easy in SAS", disponible para usuarios SAS en la plataforma SASSuport.

Los datos correponden a una compañía ficticia, llamada Accu-Phi dedicada a la creación y mantenimiento de un sofware de contabilidad. Según cuenta G.Lee en su libro, la compañía comenzó lanzando un sofware libre que pronto se hizo popular, y posteriormente decidió crear un servicio premium, donde ofrecía nuevas características y soporte. Con el objetivo de plantear la mejor estrategia de marketing para este servicio, la compañía decidió lanzarlo en un territorio 'piloto', y recoger información de sus clientes en este territorio para posteriormente poder analizarlos. Los datos recogidos a cada cliente corresponden a las siguientes variables:

- Licence : tipo de licencia, ('Freeware' o 'Premium').
- Size : es una descripción de la capacidad de facturación del cliente. Esta variable tiene tres niveles: 'Small', 'Medium' y 'Big', en función de unos umbrales
- Trust: nivel de confianza del cliente en el producto. Esta variable se ha medido a través de cuatro preguntas en una encuesta online.
- Customer satisfaction : satisfacción del cliente. Esta variable también se midió a través de una encuesta con cuatro preguntas.
- Enquires : número medio de consultas realizadas por un cliente al mes desde que comenzó a utilizar el producto
- Sales : dinero invertido por el cliente en la empresa. Inicialmente se supone que clientes con tarifa "Premium" o mayor capacidad de facturación serán más propensos a contratar más servicios. Además, en las mismas condiciones del resto de variables, una mayor confianza en la empresa y mayor satisfacción puede producir mayores ventas. Por último, un mayor número de consultas se asocia a un mayor interés por parte del cliente, por tanto, aquellos clientes con medias mensuales

más altas serán más susceptibles de comprar nuevos productos.

Trabajar con SAS

Introducción a SAS

SAS es un lenguaje de programación adaptado al mundo empresarial para el tratamiento estadístico de datos. Fue desarrollado en los años sesenta por la empresa SAS Institute, y a día de hoy es uno de las plataformas más utilizadas en el mundo del 'Data Science'.

El lenguaje SAS opera principalmente sobre tablas de datos: puede leerlas, transformarlas, combinarlas, resumirlas, crear informes a partir de ellas, etc. Sus funciones se dividen principalmente en dos procedimientos:

- Pasos DATA : permiten importar datos, crear tablas o hacer operaciones sobre las filas del conjunto de datos.
- Pasos PROC : son funciones para la manipulación de los datos por columnas. Algunas de estas funciones son la media de una variable (columna) o su desviación típica, una gráfica de frecuencias o una visualización de la tabla. Veremos ejemplos más adelante.

Es importante resaltar que después de un paso de cualquier tipo, DATA o PROC, es necesario poner la palabra clave RUN para que se compile el programa. Además, después de cada línea se escribe un punto y coma.

Creación de librerías

SAS trabaja con librerías, un concepto similar al de una carpeta dentro del ordenador, pero en este caso dentro del propio programa. Las tablas de datos se importan a tablas SAS que pueden guardarse en el ordenador de forma permanente, o en una librería temporal llamada 'work' que se reinicia al cerrar la sesión en el programa. Para abrir archivos, es necesarioque éstos estén guardados en una misma carpeta. "Apuntaremos" a esa carpeta con una libreria (en nuestro caso "AccuPhi") para referirnos a la ubicación de la tablas.

Las librerías se crean con el comando de la siguiente celda. El tercer argumento se refiere al directorio a donde quiere apuntar la librería, y debe ser modificado en cada ordenador.

In [2]:

```
LIBNAME AccuPhi '/folders/myshortcuts/SASUniversityEdition/myfolders/Data and code/';
RUN;
```

Out[2]:

```
45   ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
45 ! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
46
47   LIBNAME AccuPhi '/folders/myshortcuts/SASUniversityEdition/myfolders/Data and code/';
NOTE: Libref ACCUPHI was successfully assigned as follows:
      Engine:        V9
      Physical Name: /folders/myshortcuts/SASUniversityEdition/myfolders/Data and code
48   RUN;
49
50   ods html5 (id=saspy_internal) close;ods listing;

51
```

Alternativamente, SAS Studio University Edition permite crear librerías clickando en la parte de izquierda de la pantalla, en el apartado 'Librerías', y posteriormente 'Nueva librería'. Aquí se abrirá una ventana de diálogo donde podemos indicar el nombre 'AccuPhi' y el directorio en el que tenemos guardados los archivos. Debemos tener los archivos guardados en el directorio seleccionado. La instalación de SAS Studio University Edition apunta por defecto a la carpeta compartida con la máquina virtual 'myfolders'.

Importación de datos

SAS permite importar datos desde distintos tipos de archivo y los convierte en tablas SAS para trabajar dentro del programa. Para importar los datos desde un archivo excel, o un archivo csv utilizamos un paso 'PROC' como se muestra en la siguiente celda. Como antes, SAS Studio University Edition ofrece una ventana interactiva en la parte izquierda de la pantalla para crear las tablas SAS de forma más sencilla. Esta opción te ofrece el código necesaario para importar los datos, a falta de rellenar el nombre del archivo.

En PORC IMPORT, la opción DBMS identifica el tipo de data a importar, mientras que con OUT llamaremos al conjunto de datos en formato SAS.

In [3]:

```
PROC IMPORT DATAFILE='/folders/myshortcuts/SASUniversityEdition/myfolders/Data and
code/Data01.xlsx'
    DBMS=XLSX
    OUT=AccuPhi.Data01;
    GETNAMES=YES;
RUN;
```

Out[3]:

```
53   ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
53 ! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
54
55   PROC IMPORT DATAFILE='/folders/myshortcuts/SASUniversityEdition/myfolders/Data and
code/Data01.xlsx'
56       DBMS=XLSX
57       OUT=AccuPhi.Data01;
58       GETNAMES=YES;
59   RUN;
NOTE: One or more variables were converted because the data type is not supported by the V9 engi
ne. For more details, run with
      options MSGLEVEL=I.
NOTE: The import data set has 279 observations and 13 variables.
NOTE: ACCUPHI.DATA01 data set was successfully created.
NOTE: PROCEDURE IMPORT used (Total process time):
      real time           0.23 seconds
      cpu time            0.13 seconds

60
61   ods html5 (id=saspy_internal) close;ods listing;

62
```

Si no queremos guardar los datos de forma permanente en el ordenador, podemos crear la tabla en la librería temporal WORK, y estarán accesibles sólo durante la sesión.

Pasos en el análisis de datos

Planteamiento del problema

El CEO de nuestra empresa nos pide hacer un estudio estadístico con los datos recogidos de forma automática por la empresa. Estos datos corresponden a los clientes de la empresa en el territorio 'piloto', y se corresponden a las variables anteriormente explicadas. El CEO nos encarga que, con esos datos, demos respuesta a una serie de preguntas:

- ¿Cuánto invierte un cliente medio en nuestra empresa?¿Qué porcentaje de clientes compraron la tarifa premium?
- ¿Están los clientes satisfechos con nuestra empresa? ¿Confían en nosotros?
- ¿Existe diferencia del dinero invertido según la capacidad de facturación del cliente, su confianza en la empresa o el número de consultas realizadas?

Para responder a estas preguntas haremos uso de distintas técnicas estadísticas.

Recolección de datos. Creación de variables

Lo primero que hacemos será importar las tablas de datos. SAS acepta la mayoría de formatos comúnmente utilizados para el

Lo primero que haremos será importar las tablas de datos. SAS acepta la mayoría de formatos comúnmente utilizados para el tratamiento de datos, como .xls, xlsx o .csv.

Una vez que tenemos la tabla de datos cargada en la librería, podemos hacer un paso PROC para imprimir las 10 primeras observaciones (o el número que queramos). Así, PROC PRINT nos mostrará la tabla entera pero si al final de la línea ñadimos " (obs=10)" mostraremos sólo las primeras 10 observaciones.

In [4]:

```
PROC PRINT
    DATA = AccuPhi.Data01 (obs=10) ;
RUN;
```

Out[4]:

Scatter Plot of Sales and Trust

Obs	Respondent	License	Size	Trust01	Trust02	Trust03	Trust04	Satisfaction01	Satisfaction02	Satisfaction03	Satisfaction04	Enquiries	Sales
1	1	Freeware	Small	60	60	55	65	6	6	6	5	16	58,346.00
2	2	Premium	Big	100	100	80	100	5	5	55	5	19	144,175.0
3	3	Freeware	Big	70	64	84	83	4	5	5	6	16	88,764.00
4	4	Freeware	Big	67	75	77	70	6	6	6	6	21	81,777.00
5	5	Freeware	Bigg	70	55	55	56	6	5	6	5	12	84,403.00
6	6	Premium	Medium	.	.	.	.	.	.	.	.	14	110,458.0
7	7	Freeware	Small	65	60	72	68	4	4	4	5	12	68,351.00
8	8	Freeware	Big	70	61	50	60	5	5	5	5	21	85,628.00
9	9	Premium	Small	81	95	86	96	7	7	7	3	18	132,240.0
10	10	Premium	Small	70	80	70	65	5	6	5	5	8	66,205.00

Otro paso 'PROC' para imprimir el contenido de la tabla, esto es: nombre del archivo, número de observaciones, número de variables, información sobre las variables como el tipo, el formato, o la longitud de las variables.

In [5]:

```
PROC CONTENTS
    DATA= AccuPhi.Data01;
RUN;
```

Out[5]:

Scatter Plot of Sales and Trust

The CONTENTS Procedure

Data Set Name	ACCUPHI.DATA01	**Observations**	279
Member Type	DATA	**Variables**	13
Engine	V9	**Indexes**	0
Created	03/10/2020 14:08:27	**Observation Length**	104
Last Modified	03/10/2020 14:08:27	**Deleted Observations**	0
Protection		**Compressed**	NO
Data Set Type		**Sorted**	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	629
Obs in First Data Page	279
Number of Data Set Repairs	0
Filename	/folders/myshortcuts/SASUniversityEdition/myfolders/Data and code/data01.sas7bdat
Release Created	9.0401M6
Host Created	Linux
Inode Number	22
Access Permission	rwxrwx---
Owner Name	root
File Size	128KB
File Size (bytes)	131072

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
12	Enquiries	Num	8	BEST.		Enquiries
2	License	Char	8	$8.	$8.	License
1	Respondent	Num	8	BEST.		Respondent
13	Sales	Num	8	COMMA15.2		Sales
8	Satisfaction01	Num	8	BEST.		Satisfaction01
9	Satisfaction02	Num	8	BEST.		Satisfaction02
10	Satisfaction03	Num	8	BEST.		Satisfaction03
11	Satisfaction04	Num	8	BEST.		Satisfaction04
3	Size	Char	6	$6.	$6.	Size
4	Trust01	Num	8	BEST.		Trust01
5	Trust02	Num	8	BEST.		Trust02
6	Trust03	Num	8	BEST.		Trust03
7	Trust04	Num	8	BEST.		Trust04

En una tabla SAS, se pueden introducir nuevas variables, que pueden ser de utilidad en el análisis. Por ejemplo, en nuestro caso tenemos 4 variables distintas que miden el nivel de confianza en la empresa según las respuestas de los clientes en una encuesta. Podríamos estar interesados en resumir esta información en una sóla variable a través de la media aritmética de las variables "Trust01", "Trust02", "Trust03", y "Trust04". La nueva variable recibe el nombre de "Trust".

Tomamos la precaución de escribir en una tabla temporal en la libreria temporal Work para no alterar la tabla original.

In [6]:

```
DATA work.Data01_n;
    SET AccuPhi.Data01;
    Trust = MEAN(Trust01,Trust02,Trust03,Trust04 );
RUN;
```

Out[6]:

```
82   ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
82 ! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
83
84   DATA work.Data01_n;
85       SET AccuPhi.Data01;
86       Trust = MEAN(Trust01,Trust02,Trust03,Trust04 );
87   RUN;
NOTE: Missing values were generated as a result of performing an operation on missing values.
      Each place is given by: (Number of times) at (Line):(Column).
      7 at 86:13
NOTE: There were 279 observations read from the data set ACCUPHI.DATA01.
NOTE: The data set WORK.DATA01_N has 279 observations and 14 variables.
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.01 seconds

88
89   ods html5 (id=saspy_internal) close;ods listing;

90
```

Imaginemos ahora que pretendemos resumir, como lo hicimos con la variable "Trust", la información que recogen las variables "Satisfaction" en una sola variable. Sin embargo, supongamos en este caso que la variable "Satisfaction01" es el doble de importante que la demás. Para ello, debemos crear La nueva variable "Satisfaction" mediante una media ponderada de las variables "Satisfaction01", "Satisfaction03", "Satisfaction03" y "Satisfaction04".

In [7]:

```
DATA work.Data01_n;
    SET work.Data01_n;
    Satisfaction = (2*Satisfaction01+Satisfaction02+Satisfaction03+Satisfaction04 )/5;
RUN;
```

Out[7]:

```
92   ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
92 ! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
93
94   DATA work.Data01_n;
95       SET work.Data01_n;
96       Satisfaction = (2*Satisfaction01+Satisfaction02+Satisfaction03+Satisfaction04 )/5;
97   RUN;
NOTE: Missing values were generated as a result of performing an operation on missing values.
      Each place is given by: (Number of times) at (Line):(Column).
      25 at 96:22    6 at 96:37    2 at 96:52
NOTE: There were 279 observations read from the data set WORK.DATA01_N.
NOTE: The data set WORK.DATA01_N has 279 observations and 15 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds

98
99   ods html5 (id=saspy_internal) close;ods listing;

100
```

Ahora, podemos eliminar las variables que ya no necesitamos, es decir: "Trust01", "Trust02", "Trust03", "Trust04", "Satisfaction01", "Satisfaction02", "Satisfaction03" y "Satisfaction04".

In [8]:

```
DATA work.Data01_n;
    SET work.Data01_n;
    DROP  Trust01 Trust02 Trust03 Trust04 Satisfaction01 Satisfaction02 Satisfaction03
Satisfaction04;
RUN;
```

Out[8]:

```
102  ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
102! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
103
104  DATA work.Data01_n;
105      SET work.Data01_n;
106      DROP  Trust01 Trust02 Trust03 Trust04 Satisfaction01 Satisfaction02 Satisfaction03
Satisfaction04;
107  RUN;
NOTE: There were 279 observations read from the data set WORK.DATA01_N.
NOTE: The data set WORK.DATA01_N has 279 observations and 7 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.02 seconds

108
109  ods html5 (id=saspy_internal) close;ods listing;

110
```

Mostramos con un PROC PRINT el resultado de la tabla después de la modificaciones hechas. Observamos que tenemos las dos nuevas variables creadas a partir de las antiguas, que ya no figuran.

In [9]:

```
PROC PRINT
   DATA = work.Data01_n (OBS=10);
RUN;
```

Out[9]:

Scatter Plot of Sales and Trust

Obs	Respondent	License	Size	Enquiries	Sales	Trust	Satisfaction
1	1	Freeware	Small	16	58,346.00	60.00	5.8
2	2	Premium	Big	19	144,175.00	95.00	15.0
3	3	Freeware	Big	16	88,764.00	75.25	4.8
4	4	Freeware	Big	21	81,777.00	72.25	6.0
5	5	Freeware	Bigg	12	84,403.00	59.00	5.6
6	6	Premium	Medium	14	110,458.00	.	.

Obs	Respondent	License	Size	Enquiries	Sales	Trust	Satisfaction
7	7	Freeware	Small	12	68,351.00	66.25	4.2
8	8	Freeware	Big	21	85,628.00	60.25	5.0
9	9	Premium	Small	18	132,240.00	89.50	6.2
10	10	Premium	Small	8	66,205.00	71.25	5.2

También podemos ordenar la tabla según alguna variable. Por ejemplo, podemos ordenarlo según el número de consultas realizadas, y en caso de empate, por el número de cliente.

In [10]:

```
PROC SORT DATA=work.Data01_n;
    BY Enquiries Respondent;
RUN;
```

Out[10]:

```
121  ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
121! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
122
123  PROC SORT DATA=work.Data01_n;
124      BY Enquiries Respondent;
125  RUN;
NOTE: There were 279 observations read from the data set WORK.DATA01_N.
NOTE: The data set WORK.DATA01_N has 279 observations and 7 variables.
NOTE: PROCEDURE SORT used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds

126
127  ods html5 (id=saspy_internal) close;ods listing;

128
```

Ahora, podemos hacer un PROC print donde comprobamos que el conjunto de datos está ordenado como queríamos.

In [11]:

```
PROC PRINT
    DATA = work.Data01_n (OBS=10);
RUN;
```

Out[11]:

Scatter Plot of Sales and Trust

Obs	Respondent	License	Size	Enquiries	Sales	Trust	Satisfaction
1	158	Freeware	Small	1	8,118.00	.	.
2	160	Freeware	Small	3	9,728.00	.	.
3	279	Freeware	Small	3	3,177.00	.	.
4	274	Freeware	Small	6	44,790.00	24.25	6.2
5	26	Freeware	Small	7	60,110.00	55.00	5.8
6	66	Premium	Small	7	56,305.00	52.50	6.2
7	68	Freeware	Small	7	28,486.00	18.25	4.0
8	107	Freeware	Small	7	48,079.00	42.75	5.8

Obs	Respondent	License	Size	Enquiries	Sales	Trust	Satisfaction
10	10	Premium	Small	8	66,205.00	71.25	5.2

Limpieza. Tratamiento de 'missing data'.

Antes de comenzar un análisis de datos es importante hacer una depuración en las tablas. Es común cometer errores en la recolección de datos, por ejemplo, en la escritura de datos en las variables (caracteres duplicados, datos en variables que no corresponden, etc). Además, también debe tenerse en cuenta la posibilidad de datos faltantes en algunas variables (no se guardaron bien los datos, no se rellenó cierta casilla en el formulario, etc).

SAS identifica observaciones faltantes en variables numéricas con un punto (·) y en variables de tipo caracter con un espacio en blanco. Otros Sofwares, sin embargo, guardan estos datos con la cadena de caracteres 'NaN', por lo que debe prestarse especial atención en su tratamiento.

Empezamos la depuración de datos mirando las observaciones con errores en la recolección de datos. Identificamos primero observaciones de tipo 'clase', es decir, variables que solo tienen un número fijo de valores posibles, como por ejemplo sexo, puntuaciones de 1-5 o la variable 'Size' que clasifica a los clientes según su capacidad de facturación.

En primer lugar cargamos la tabla original y creamos una tabla auxiliar para trabajar en ella:

In [12]:

```
DATA work.Data01_missing;
    SET AccuPhi.Data01;
RUN;
```

Out[12]:

```
139  ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
139! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
140
141  DATA work.Data01_missing;
142      SET AccuPhi.Data01;
143  RUN;
NOTE: There were 279 observations read from the data set ACCUPHI.DATA01.
NOTE: The data set WORK.DATA01_MISSING has 279 observations and 13 variables.
NOTE: DATA statement used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds

144
145  ods html5 (id=saspy_internal) close;ods listing;

146
```

Para detectar los errores debidos a fallos en la recolección de variables de clase, podemos utilizar el PROC FREQ de SAS, que crea una tabla para cada variable mostrando los valores encontrados y frecuencia en cada una de estas variables. Las observaciones candidatas serán aquellas que presenten un único valor. A continuación escribimos las variables sobre las que queremos hacer el estudio, en nuestro caso "License", "Size", "Satisfaction01", "Satisfaction02", "Satisfaction03" y "Satisfaction04".

In [13]:

```
PROC FREQ
    DATA = work.Data01_missing;
    TABLES License Size  Satisfaction01 Satisfaction02 Satisfaction03 Satisfaction04;
RUN;
```

Out[13]:

Scatter Plot of Sales and Trust

The FREQ Procedure

License

License	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Freeware	145	51.97	145	51.97
Premium	134	48.03	279	100.00

Size

Size	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Big	119	42.65	119	42.65
Bigg	1	0.36	120	43.01
Medium	81	29.03	201	72.04
Small	78	27.96	279	100.00

Satisfaction01

Satisfaction01	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	1	0.39	1	0.39
2	4	1.57	5	1.97
3	8	3.15	13	5.12
4	32	12.60	45	17.72
5	85	33.46	130	51.18
6	87	34.25	217	85.43
7	37	14.57	254	100.00

Frequency Missing = 25

Satisfaction02

Satisfaction02	Frequency	Percent	Cumulative Frequency	Cumulative Percent
2	4	1.53	4	1.53
3	8	3.07	12	4.60
4	31	11.88	43	16.48
5	77	29.50	120	45.98
6	100	38.31	220	84.29
7	41	15.71	261	100.00

Frequency Missing = 18

Satisfaction03

Satisfaction03	Frequency	Percent	Cumulative Frequency	Cumulative Percent
2	4	1.54	4	1.54
3	8	3.08	12	4.62
4	29	11.15	41	15.77
5	68	26.15	109	41.92
6	103	39.62	212	81.54
7	47	18.08	259	99.62
55	1	0.38	260	100.00

Frequency Missing = 19

Satisfaction04

Satisfaction04	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	1	0.39	1	0.39
2	5	1.94	6	2.33
3	18	6.98	24	9.30
4	30	11.63	54	20.93
5	102	39.53	156	60.47
6	94	36.43	250	96.90
7	8	3.10	258	100.00

Frequency Missing = 21

Vemos que en la primera tabla, que corresponde a la variable Size, hay dos valores de la variable con una única observación, "Medium" y "Bigg". Por el conocimiento a priori sobre las clases, podemos concluir que el valor es correcto, y el segundo se debe a un error de escritura en la palabra "Big". Para corregirlo, simplemente cambiamos el valor "Bigg" de la tabla por un "Big", como se muestra en la siguiente celda:

In [14]:

```
DATA work.Data01_missing;
    SET work.Data01_missing;
    IF Size ="Bigg" THEN Size = "Big";
RUN;
```

Out[14]:

```
158  ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
158! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
159
160  DATA work.Data01_missing;
161      SET work.Data01_missing;
162      IF Size ="Bigg" THEN Size = "Big";
```

```
163  RUN;
NOTE: There were 279 observations read from the data set WORK.DATA01_MISSING.
NOTE: The data set WORK.DATA01_MISSING has 279 observations and 13 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds

164
165  ods html5 (id=saspy_internal) close;ods listing;

166
```

Comprobamos que el error se ha corregido imprimiendo de nuevo la tabla de frecuencias para la variable Size.

In [15]:

```
PROC FREQ
    DATA = work.Data01_missing;
    TABLES  Size;
RUN;
```

Out[15]:

Scatter Plot of Sales and Trust

The FREQ Procedure

Size

Size	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Big	120	43.01	120	43.01
Medium	81	29.03	201	72.04
Small	78	27.96	279	100.00

Por otro lado, también podemos ver que la variable Satisfaction04 tiene una observación de 55. Esta variable recoge puntuaciones del 1 al 7, por lo que el valor 55 se corresponde a otro error en la escritura, vamos a suponer, del número 5. Como antes, podemos corregirlo buscando la observación en la tabla.

In [16]:

```
DATA work.Data01_missing;
    SET work.Data01_missing;
    IF Satisfaction03= 55 THEN Satisfaction03 = 5;
RUN;
```

Out[16]:

```
178  ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
178! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
179
180  DATA work.Data01_missing;
181      SET work.Data01_missing;
182      IF Satisfaction03= 55 THEN Satisfaction03 = 5;
183  RUN;
NOTE: There were 279 observations read from the data set WORK.DATA01_MISSING.
NOTE: The data set WORK.DATA01_MISSING has 279 observations and 13 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds

184
```

```
185  ods html5 (id=saspy_internal) close;ods listing;

186
```

Volvemos a comprobar, por ejemplo, con un PROC FREQ que se ha corregido correctamente.

In [17]:

```
PROC FREQ
    DATA = work.Data01_missing;
    TABLES Satisfaction03;
RUN;
```

Out[17]:

Scatter Plot of Sales and Trust

The FREQ Procedure

Satisfaction03

Satisfaction03	Frequency	Percent	Cumulative Frequency	Cumulative Percent
2	4	1.54	4	1.54
3	8	3.08	12	4.62
4	29	11.15	41	15.77
5	69	26.54	110	42.31
6	103	39.62	213	81.92
7	47	18.08	260	100.00

Frequency Missing = 19

Para terminar, nos preocupamos de los datos "missing". Estos valores pueden darse en variables de tipo continuo o discreto. Para "rellenar" valores faltantes en valores de tipo continuo pueden usarse distintos métodos como la imputación por media o la predicción del valor en función del resto de variables. De la misma forma, para imputar datos discreto podría utilizarse la moda de la variable o una predicción de clase (esto es algo más sofisticado).

En la tabla Data01_missing, encontrábamos datos faltantes en todas las variables Trust y Satisfaction en la cuarta observación. En el caso de las variables Satisfaction, en el PROC FREQ anterior se mostraba a pie de tabla "Frequency Missing = 1" en todas ellas. Como sólo vamos a utilizar la media de estas variables, primero reescribimos la tabla Data01_missing sustituyendo las variables Trust y Satisfaction por su media y media ponderada respectivamente. Esto es equivalente a lo que hicimos en la sección previa, pero agrupándolo en un solo PROC.

In [18]:

```
DATA work.Data01_missing;
    set work.Data01_missing;
    Trust = MEAN(Trust01,Trust02,Trust03,Trust04 );
    Satisfaction = (2*Satisfaction01+Satisfaction02+Satisfaction03+Satisfaction04 )/5;
    DROP Trust01 Trust02 Trust03 Trust04 Satisfaction01 Satisfaction02 Satisfaction03
Satisfaction04;
RUN;
```

Out[18]:

```
198  ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
198! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
```

```
199
200  DATA work.Data01_missing;
201       set work.Data01_missing;
202       Trust = MEAN(Trust01,Trust02,Trust03,Trust04 );
203       Satisfaction = (2*Satisfaction01+Satisfaction02+Satisfaction03+Satisfaction04 )/5;
204       DROP Trust01 Trust02 Trust03 Trust04 Satisfaction01 Satisfaction02 Satisfaction03
Satisfaction04;
205  RUN;
NOTE: Missing values were generated as a result of performing an operation on missing values.
      Each place is given by: (Number of times) at (Line):(Column).
      7 at 202:13    25 at 203:22    6 at 203:37    2 at 203:52
NOTE: There were 279 observations read from the data set WORK.DATA01_MISSING.
NOTE: The data set WORK.DATA01_MISSING has 279 observations and 7 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.01 seconds

206
207  ods html5 (id=saspy_internal) close;ods listing;

208
```

Mediante el PROC PRINT observamos cómo hay valores missing o faltantes.

In [19]:

```
PROC PRINT
    DATA = work.Data01_missing (OBS=14);
RUN;
```

Out[19]:

Scatter Plot of Sales and Trust

Obs	Respondent	License	Size	Enquiries	Sales	Trust	Satisfaction
1	1	Freeware	Small	16	58,346.00	60.00	5.8
2	2	Premium	Big	19	144,175.00	95.00	5.0
3	3	Freeware	Big	16	88,764.00	75.25	4.8
4	4	Freeware	Big	21	81,777.00	72.25	6.0
5	5	Freeware	Big	12	84,403.00	59.00	5.6
6	6	Premium	Medium	14	110,458.00	.	.
7	7	Freeware	Small	12	68,351.00	66.25	4.2
8	8	Freeware	Big	21	85,628.00	60.25	5.0
9	9	Premium	Small	18	132,240.00	89.50	6.2
10	10	Premium	Small	8	66,205.00	71.25	5.2
11	11	Premium	Small	13	66,766.00	49.75	6.0
12	12	Freeware	Small	11	50,382.00	45.00	4.0
13	13	Freeware	Big	20	89,561.00	49.00	.
14	14	Premium	Small	11	74,573.00	78.00	5.6

En estos casos hay dos opciones: eliminar toda la fila que contiene una observación faltante, o imputar estos valores. La imputación más sencilla se hace a través de la media de la variable para todas las observaciones. Se puede utilizar para hacer el paso 'PROC STDIZE'

In [20]:

```
PROC STDIZE
    DATA=Data01_missing
    OUT = Data01_missing
    reponly  /* solo reemplaza */
    method=MEAN;
    VAR Trust Satisfaction;
RUN;
```

Out[20]:

```
219  ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
219! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
220
221  PROC STDIZE
222       DATA=Data01_missing
223       OUT = Data01_missing
224       reponly  /* solo reemplaza */
225       method=MEAN;
226       VAR Trust Satisfaction;
227  RUN;
NOTE: There were 279 observations read from the data set WORK.DATA01_MISSING.
NOTE: The data set WORK.DATA01_MISSING has 279 observations and 7 variables.
NOTE: PROCEDURE STDIZE used (Total process time):
      real time           0.00 seconds
      cpu time            0.01 seconds

228
229  ods html5 (id=saspy_internal) close;ods listing;

230
```

Por último, comprobamos que los datos se han imputado correctamente con un PROC PRINT

In [21]:

```
PROC PRINT
    DATA = Data01_missing (OBS=14);
RUN;
```

Out[21]:

Scatter Plot of Sales and Trust

Obs	Respondent	License	Size	Enquiries	Sales	Trust	Satisfaction
1	1	Freeware	Small	16	58,346.00	60.0000	5.80000
2	2	Premium	Big	19	144,175.00	95.0000	5.00000
3	3	Freeware	Big	16	88,764.00	75.2500	4.80000
4	4	Freeware	Big	21	81,777.00	72.2500	6.00000
5	5	Freeware	Big	12	84,403.00	59.0000	5.60000
6	6	Premium	Medium	14	110,458.00	70.6903	5.37886
7	7	Freeware	Small	12	68,351.00	66.2500	4.20000
8	8	Freeware	Big	21	85,628.00	60.2500	5.00000
9	9	Premium	Small	18	132,240.00	89.5000	6.20000
10	10	Premium	Small	8	66,205.00	71.2500	5.20000
11	11	Premium	Small	13	66,766.00	49.7500	6.00000
12	12	Freeware	Small	11	50,382.00	45.0000	4.00000

```
199
200  DATA work.Data01_missing;
201       set work.Data01_missing;
202       Trust = MEAN(Trust01,Trust02,Trust03,Trust04 );
203       Satisfaction = (2*Satisfaction01+Satisfaction02+Satisfaction03+Satisfaction04 )/5;
204       DROP Trust01 Trust02 Trust03 Trust04 Satisfaction01 Satisfaction02 Satisfaction03
Satisfaction04;
205  RUN;
NOTE: Missing values were generated as a result of performing an operation on missing values.
      Each place is given by: (Number of times) at (Line):(Column).
      7 at 202:13    25 at 203:22    6 at 203:37    2 at 203:52
NOTE: There were 279 observations read from the data set WORK.DATA01_MISSING.
NOTE: The data set WORK.DATA01_MISSING has 279 observations and 7 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.01 seconds

206
207  ods html5 (id=saspy_internal) close;ods listing;

208
```

Mediante el PROC PRINT observamos cómo hay valores missing o faltantes.

In [19]:

```
PROC PRINT
    DATA = work.Data01_missing (OBS=14);
RUN;
```

Out[19]:

Scatter Plot of Sales and Trust

Obs	Respondent	License	Size	Enquiries	Sales	Trust	Satisfaction
1	1	Freeware	Small	16	58,346.00	60.00	5.8
2	2	Premium	Big	19	144,175.00	95.00	5.0
3	3	Freeware	Big	16	88,764.00	75.25	4.8
4	4	Freeware	Big	21	81,777.00	72.25	6.0
5	5	Freeware	Big	12	84,403.00	59.00	5.6
6	6	Premium	Medium	14	110,458.00	.	.
7	7	Freeware	Small	12	68,351.00	66.25	4.2
8	8	Freeware	Big	21	85,628.00	60.25	5.0
9	9	Premium	Small	18	132,240.00	89.50	6.2
10	10	Premium	Small	8	66,205.00	71.25	5.2
11	11	Premium	Small	13	66,766.00	49.75	6.0
12	12	Freeware	Small	11	50,382.00	45.00	4.0
13	13	Freeware	Big	20	89,561.00	49.00	.
14	14	Premium	Small	11	74,573.00	78.00	5.6

En estos casos hay dos opciones: eliminar toda la fila que contiene una observación faltante, o imputar estos valores. La imputación más sencilla se hace a través de la media de la variable para todas las observaciones. Se puede utilizar para hacer el paso 'PROC STDIZE'

In [20]:

```
PROC STDIZE
    DATA=Data01_missing
    OUT = Data01_missing
    reponly  /* solo reemplaza */
    method=MEAN;
    VAR Trust Satisfaction;
RUN;
```

Out[20]:

```
219  ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline') d
evice=svg style=HTMLBlue; ods
219! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
220
221  PROC STDIZE
222       DATA=Data01_missing
223       OUT = Data01_missing
224       reponly  /* solo reemplaza */
225       method=MEAN;
226       VAR Trust Satisfaction;
227  RUN;
NOTE: There were 279 observations read from the data set WORK.DATA01_MISSING.
NOTE: The data set WORK.DATA01_MISSING has 279 observations and 7 variables.
NOTE: PROCEDURE STDIZE used (Total process time):
      real time           0.00 seconds
      cpu time            0.01 seconds

228
229  ods html5 (id=saspy_internal) close;ods listing;

230
```

Por último, comprobamos que los datos se han imputado correctamente con un PROC PRINT

In [21]:

```
PROC PRINT
    DATA = Data01_missing (OBS=14);
RUN;
```

Out[21]:

Scatter Plot of Sales and Trust

Obs	Respondent	License	Size	Enquiries	Sales	Trust	Satisfaction
1	1	Freeware	Small	16	58,346.00	60.0000	5.80000
2	2	Premium	Big	19	144,175.00	95.0000	5.00000
3	3	Freeware	Big	16	88,764.00	75.2500	4.80000
4	4	Freeware	Big	21	81,777.00	72.2500	6.00000
5	5	Freeware	Big	12	84,403.00	59.0000	5.60000
6	6	Premium	Medium	14	110,458.00	70.6903	5.37886
7	7	Freeware	Small	12	68,351.00	66.2500	4.20000
8	8	Freeware	Big	21	85,628.00	60.2500	5.00000
9	9	Premium	Small	18	132,240.00	89.5000	6.20000
10	10	Premium	Small	8	66,205.00	71.2500	5.20000
11	11	Premium	Small	13	66,766.00	49.7500	6.00000
12	12	Freeware	Small	11	50,382.00	45.0000	4.00000

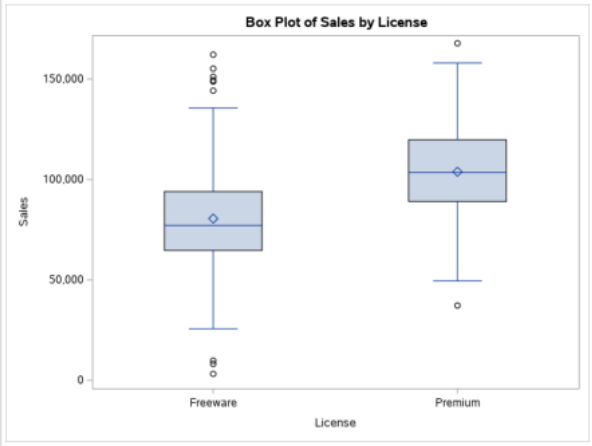
Los clientas con menos capacidad de facturación tienden a hacer menos consultas, aunque no hay realmente una tendencia clara.

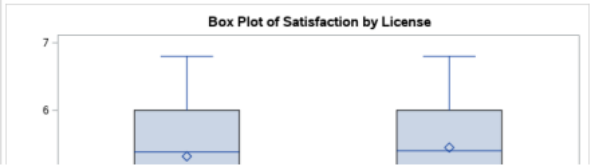
Realizamos un estudio similar con otras variables:

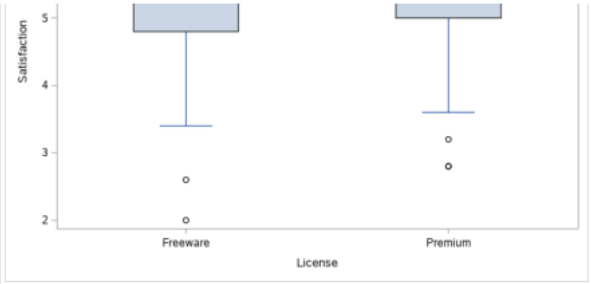
In [39]:

```
PROC SGPLOT DATA=Data01_des;
    TITLE Box Plot of Sales by License;
    vbox Sales / category=License;
RUN;
PROC SGPLOT  DATA=Data01_des;
    TITLE Box Plot of Satisfaction by License;
    vbox Satisfaction / category=License;
RUN;
```

Out[39]:







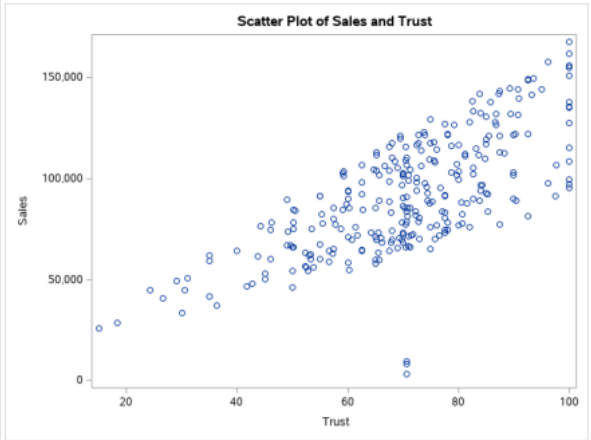
Observamos como la satisfacción de los clientes Premium y Freeware es prácticamente igual, presenciándose algún outlier negativo en ambos casos. Aunque puede resultar extraño que los clientes Premium no estén más satisfechos, hay que tener en cuenta que, seguramente, éstos sean más exigentes que los clientes Freeware.

Por último, queremos relacionar las variables numéricas "Trust" y "Sales" viendo si hay una tendencia clara que los relaciones. Esto lo hacemos con un diagrama de dispersión (PROC SGPLOT, opción scatter):

In [40]:

```
PROC SGPLOT
    DATA=Data01_des;
    TITLE Scatter Plot of Sales and Trust;
    scatter x=Trust y=Sales;
RUN;
```

Out[40]:



Se observa de manera clara, que a mayor confianza en la empresa mayor dinero se invierte. Parece claro, pues, que la empresa deber centrase en mejorar el nivel de confianza de sus clientes para maximizar beneficios. Nótese que hay tres observaciones atípicas, pero dentro de la lógica. Clientes que, pese a confiar en la empresa, han decidido no invertir mucho dinero. Sería más sopreprendente la existencia de clientes que, con un nivel bajo de confianza, invirtieran mucho dinaero en la empresa.