CAPITULO 5
PRE-PROCESAMIENTO DE DATOS

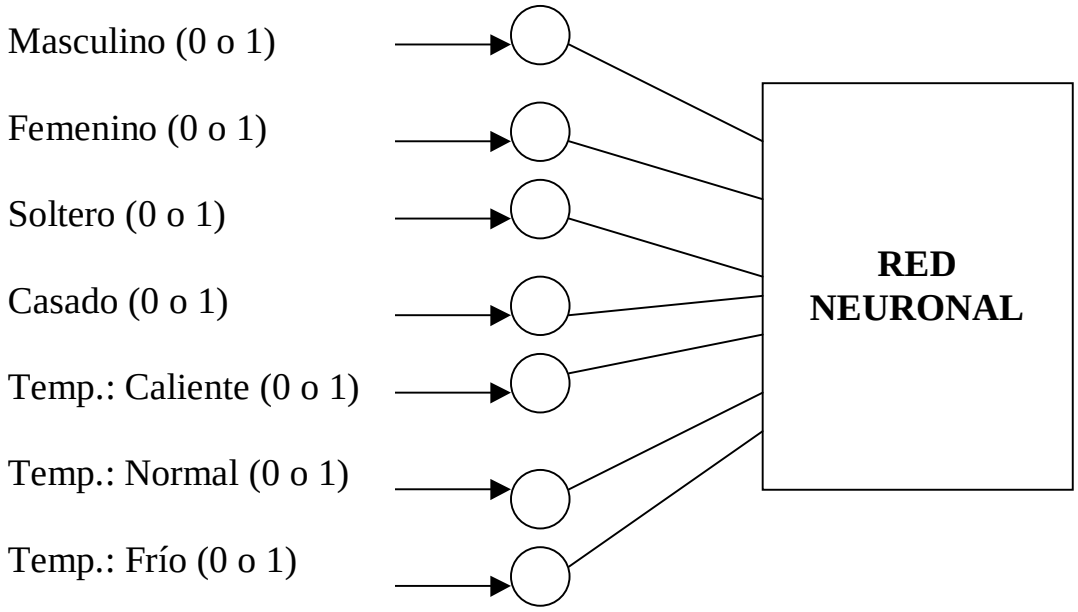
Consiste en la preparación previa de los datos para ser usados por la construcción, entrenamiento y prueba de un modelo de red neuronal.

Alguno de los problemas más comunes en la preparación de los datos es la mezcla de variables continuas y discretas.

a) Debe hacerse una definición previa para el conjunto de variables.

Binarias → sexo, estado civil, temperatura.

La preparación de los datos de entrada para entrenamiento y pronóstico, contemplaría la disposición de la siguiente manera:



Errores comunes:

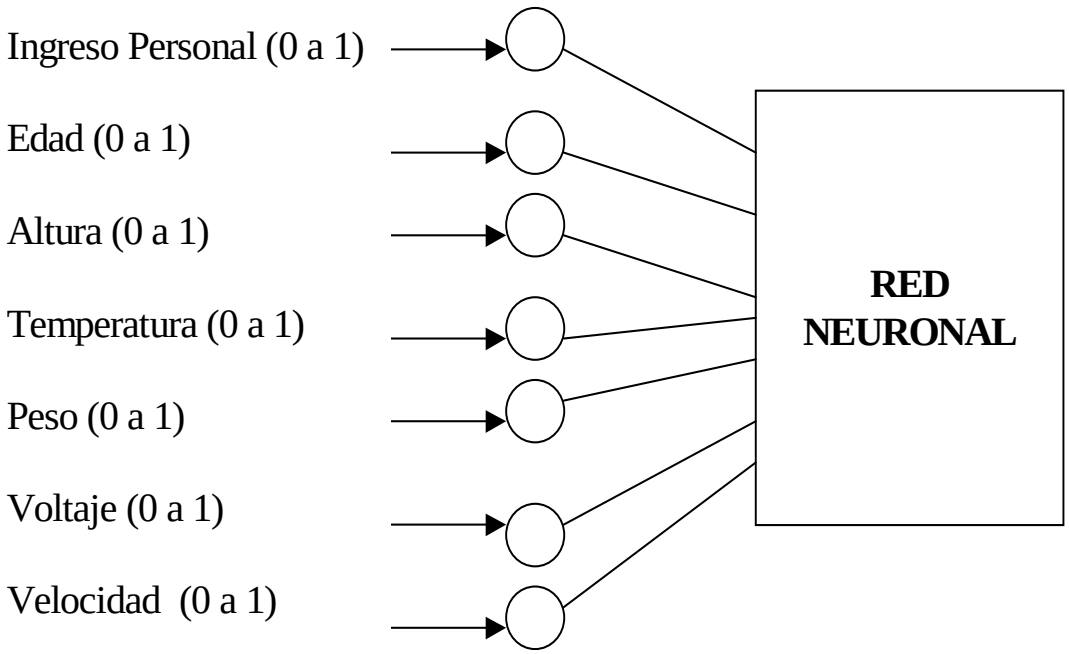
Uso de valores continuos para conceptos simbólicos (animal)
Los meses del año representados con valores numéricos del 1 al 12.

b) Otro ejemplo de mezcla de datos podría ser la definición de variables atributos a través de variables con valores continuos.

Por ejemplo, supongamos que todas las variables han sido estandarizadas.

Continuos → ingreso, edad, altura, temperatura, peso, voltaje, velocidad.

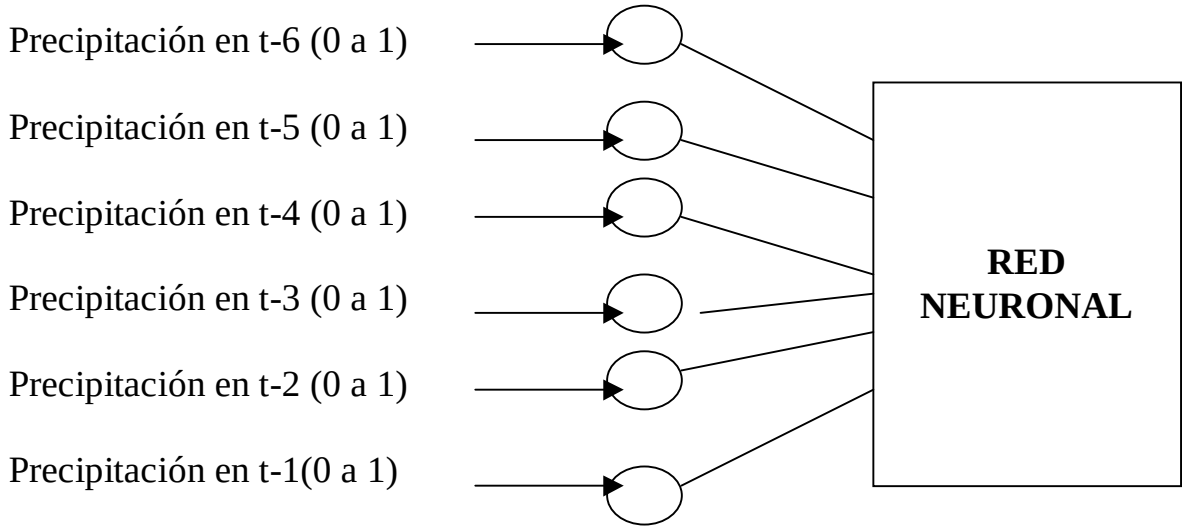
La preparación de los datos de entrada para entrenamiento y pronóstico, contemplaría la disposición de la siguiente manera:



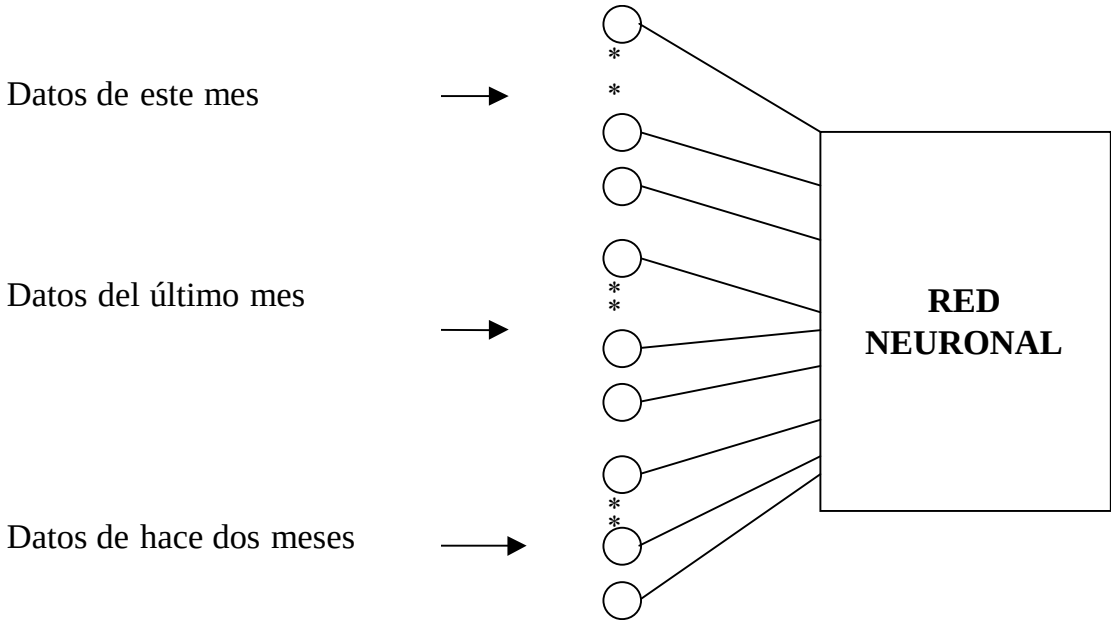
Errores comunes:

Mezclar escalas. (toneladas con kgs, años con meses)
Variables con altas variaciones (máximos y mínimos).

c) Usar como patrones de entrada los diferentes períodos que puedan afectar la salida. Un caso concreto sería las series temporales.



Y no hacer la preparación de la manera que se muestra abajo, ya que sería interminable la preparación de las entradas y por supuesto, la construcción de los modelos de redes neuronales.



d) Un excesivo número de entradas requiere demasiados casos para entrenamiento, y esto puede conducir a:

1. Arquitectura de redes complejas. Es decir demasiadas entradas con complejas estructuras de datos que implican un gran número de nodos de entrada.
2. Alto consumo de tiempo computacional.
3. Esfuerzo humano excesivo conducente a múltiple pruebas por ensayo y error. Esto hace difícil la interpretación de los resultados.

e) Estado actual en la construcción de las redes neuronales

1. Ensayo y Error. Se ajustan los datos de entrada a los resultados deseados. Muy común en los modelos físicos que requieren precisión.

2. Adaptación de la arquitectura de la red. La selección del número de capas ocultas y el número de neuronas ocultas sin seguir ningún tipo de criterio.

3. Adaptación a los objetivos. Forzar el modelo a los resultados deseados.

En los algoritmos de entrenamiento **supervisado** puede reducirse este problema mediante **preprocesamiento.**

PORQUÉ HACER PREPROCESAMIENTO

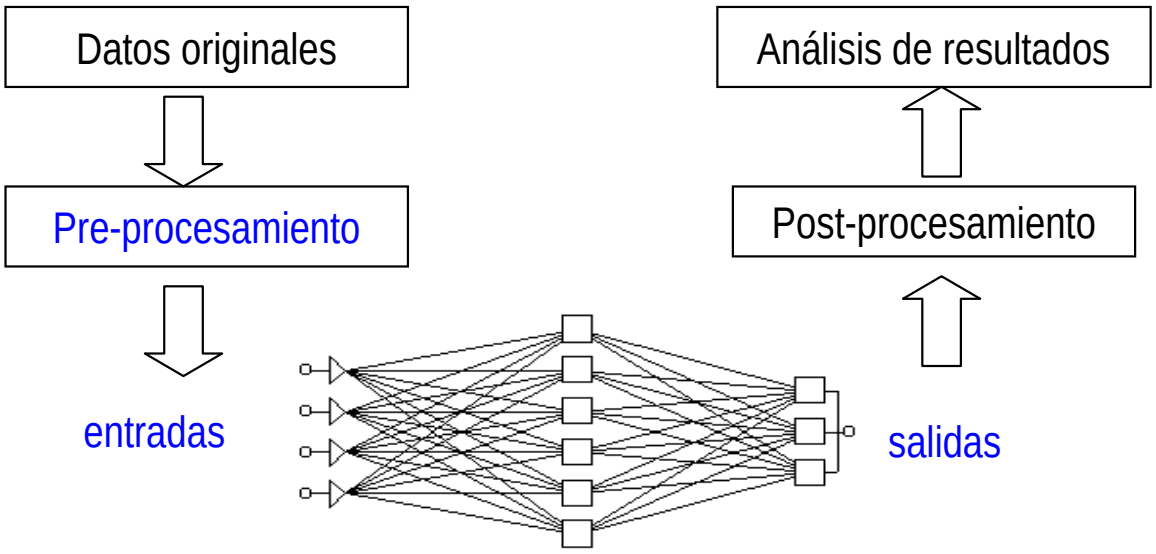
Basados en las premisas señaladas en la página anterior, podríamos indicar que las razones fundamentales para hacer el pre-procesamiento son las siguientes.

1. Relación incremental de las horas/hombre en el diseño y construcción de redes neuronales.

2. Carácter de independencia de los datos con la construcción de la red.

3. Un piso estadístico representativo al proceso heurístico de construcción de la red.

De este modo, en un proceso normal, como producto del pre-procesamiento, existiría una transformación adicional. Ella es la transformación de los resultados conseguidos a través de la aplicación del modelo construido en valores que puedan ser interpretados físicamente. Esto es post-procesamiento.



Hay que tener presente que en toda red neuronal:

Las entradas numéricas producen salidas numéricas.

Las entradas pueden estar en cualquier rango numérico.

La salida es producida en un rango estrictamente limitado.

Las funciones de activación son sensibles a estar en rangos limitados (ejm.: sigmode)

El rango limitado de respuesta de la red y la información en forma numérica implica que la solución neuronal requiera de un preprocesamiento y un postprocesamiento. (Bishop, 1995).

Métodos más comunes de hacer pre-procesamiento:

Tal como se mostró a través de ejemplos en las páginas anteriores, los métodos más comunes de preparación de datos son:

Escalamiento: transformación del contenido de las variables a un rango 0,1.

Análisis de variables nominales: transformación de un valor categórico a un valor numérico.

MÉTODOS DE PREPROCESAMIENTO DE DATOS

Como métodos alternativos y con mayor consistencia de análisis, surgen técnicas emergentes que de algún modo proveen medios para la preparación y escogencia de los datos.
Sin embargo, el piso estadístico como fuente de afirmación teórica para la selección y preparación de los datos, sigue siendo el recurso más idóneo de pre-procesamiento.

1. Data mining

Exploración de datos para la búsqueda de:
- patrones consistentes
- relación sistemática entre variables

No identifica las relaciones específicas entre las variables

El proceso consiste de:
- Exploración
- Construcción del modelo
- Definición de patrones

2. Data warehousing

Organización de datos multivariantes para facilitar futuras recuperaciones de información.

No identifica las relaciones específicas entre las variables.

El proceso consiste de:
- Relaciones de búsquedas exhaustivas entre grandes bases de datos
- Extracción de variables
- Creación de nuevos conjunto de datos

3. Análisis Exploratorio de Datos

Exploración de datos usando una gran variedad de técnicas de análisis multivariante para la búsqueda de patrones sistemáticos.

Identifica las relaciones específicas entre las variables

El proceso consiste de:

Métodos exploratorios de estadística básica
Técnicas exploratorias multivariantes
Aplicación estadística sobre los datos, incluyendo grandes volúmenes de datos.

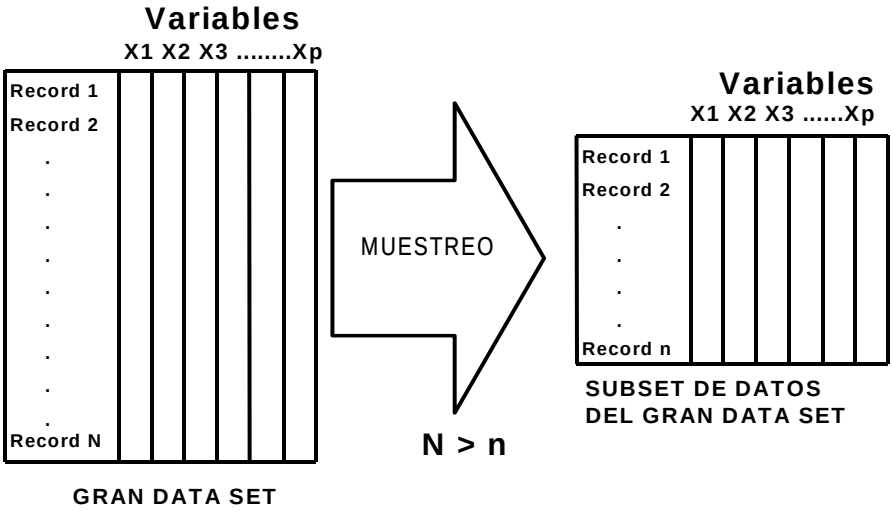
Métodos exploratorios de estadística básica

Revisión de la tendencia mediante la distribución que siguen las variables.
Relación significativa mediante la matriz de correlación entre variables.
……..

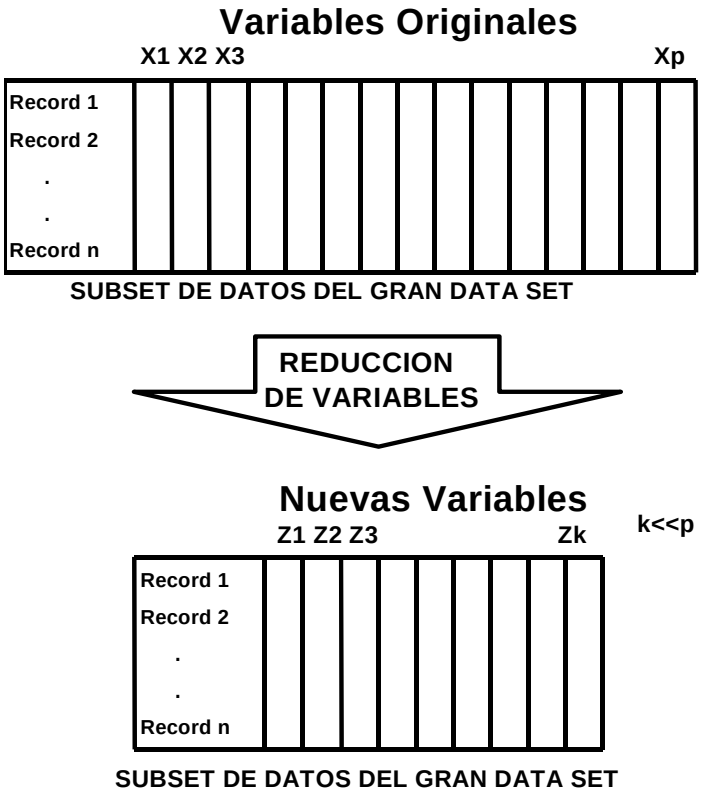
Técnicas exploratorias multivariantes

Muestreo
Análisis Factorial
Análisis de componentes principales
Regresión múltiple
Análisis de series de tiempo
Análisis de conglomerados (cluster)
Correlación canónica

1.- Selección de muestras



2.- Reducción de variables



EJEMPLO DE PREPROCESAMIENTO DE DATOS USANDO MATLAB

Este ejercicio consiste de la utilización del método de componentes principales como herramienta para la reducción de las variables independientes.

Un modelo de la forma $y=f(x_i)$, $i=1,N$, podría ser reducido en el número de variables originales x_i. De este modo, se puede tener un nuevo modelo basado en las variables transformadas z, de la forma $y=f(z_i)$, $i=1,n$, siendo $n<<N$.

Sea el conjunto de datos dado por el archivo en Excel 250_x1_x9.xls, una muestra estratificada obtenida después de separar el conjunto total de los datos en dos porciones, una para extraer muestras para entrenamiento y la otra para hacer la verificación del modelo. Esta muestra contiene los valores de *x* y los valores de *y*. Desde este archivo se extraen solamente las variables independientes *x* para ser examinadas mediante el método de reducción. Este archivo es data250.txt (disponible en formato ascii).

Desde Matlab, procederemos a hacer el análisis de componentes principales. A manera de ejemplo, solamente se incluyen el procedimiento general y es comparado con el método numérico de descomposición de valores singulares (SVD), a los fines de conocer el método alternativo para hacer estos cálculos básicos.

```
load a:\data250.txt

X=data250;
%
% Matriz X con solo las variables independientes
%
size(X)
%Tamaño de la matriz de datos X
ans =

   258    10
```

```
X=X(:,2:10);
size(X)
ans =

    258    9
%La matriz que compone las variables independientes es de 258 observaciones y 9
variables independientes.

X(1:1,1:10)
ans =

  Columns 1 through 7

    0.2941    0.3737    0.0377    0.9930    0.9982    0.2746    1.2645

  Columns 8 through 9

    0.0       0.0010

% Esta muestra le será explorada a los fines de determinar si se le puede aplicar una
% reducción de variables.
% Utilizaremos los dos métodos que pueden determinar los componentes
% principales.
%
% Método de la descomposición de los valores singulares.
help SVD
 SVD    Singular value decomposition.
    [U,S,V] = SVD(X) produces a diagonal matrix S, of the same
    dimension as X and with nonnegative diagonal elements in
    decreasing order, and unitary matrices U and V so that
    X = U*S*V'.

    S = SVD(X) returns a vector containing the singular values.

    [U,S,V] = SVD(X,0) produces the "economy size"
    decomposition. If X is m-by-n with m > n, then only the
    first n columns of U are computed and S is n-by-n.

    See also SVDS.
```

% De acuerdo a este método podemos obtener una matriz diagonal
% que incluye los valores propios de la matriz simétrica.
% Para nuestro caso la matriz simétrica es la matriz de correlación.
%
% 1.- Determinamos la matriz de correlación R.
R=corrcoef(X)
R =

Columns 1 through 7

```
    1.0000   -0.0112   -0.8479   -0.0646   -0.1847   -0.8152    0.1269
   -0.0112    1.0000    0.0191    0.0745   -0.0415    0.0381   -0.0478
   -0.8479    0.0191    1.0000    0.0256    0.0839    0.9784   -0.1359
   -0.0646    0.0745    0.0256    1.0000    0.0595   -0.0075    0.1409
   -0.1847   -0.0415    0.0839    0.0595    1.0000    0.0241    0.0925
   -0.8152    0.0381    0.9784   -0.0075    0.0241    1.0000   -0.1672
    0.1269   -0.0478   -0.1359    0.1409    0.0925   -0.1672    1.0000
    0.7016    0.0203   -0.9187   -0.0808   -0.1766   -0.8308    0.0523
   -0.7847   -0.0042    0.9685    0.0592    0.1416    0.9011   -0.0867
```

Columns 8 through 9

```
    0.7016   -0.7847
    0.0203   -0.0042
   -0.9187    0.9685
   -0.0808    0.0592
   -0.1766    0.1416
   -0.8308    0.9011
    0.0523   -0.0867
    1.0000   -0.9867
   -0.9867    1.0000
```

% a simple vista podemos observar en la matriz componentes con valores cercanos a 1.
% Esto pareciera indicar una colinealidad entre variables.
% De acuerdo a los valores singulares, tenemos:
%
[u,lambda,v]=svd(R);
%
% donde:
%
lambda

lambda =

Columns 1 through 7

```
    4.5427         0         0         0         0         0         0
         0    1.2239         0         0         0         0         0
         0         0    1.0644         0         0         0         0
         0         0         0    0.9047         0         0         0
         0         0         0         0    0.8054         0         0
         0         0         0         0         0    0.3364         0
         0         0         0         0         0         0    0.1211
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
```

Columns 8 through 9

```
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
    0.0015         0
         0    0.0000
```

% y los componentes principales quedan definidos por los autovectores v.
% Es decir,
v

v =

Columns 1 through 7

```
    0.4099   -0.0231    0.0029   -0.0638    0.0429   -0.7847    0.4412
   -0.0061   -0.0748    0.8255    0.4151    0.3717   -0.0327   -0.0374
   -0.4657   -0.0444    0.0129   -0.0592    0.0467   -0.0017    0.2365
   -0.0262    0.5561    0.4731   -0.2810   -0.6186   -0.0003    0.0676
   -0.0743    0.5044   -0.2952    0.7853   -0.1093   -0.0377    0.1501
   -0.4467   -0.1113    0.0358   -0.0753    0.0564    0.1203    0.7284
    0.0685    0.6384   -0.0698   -0.3383    0.6755    0.1033    0.0391
    0.4399   -0.0847    0.0325    0.0428   -0.0416    0.5173    0.3781
   -0.4594    0.0343   -0.0148   -0.0496    0.0449   -0.2983   -0.2141
```

Columns 8 through 9

```
   -0.1171    0.0360
    0.0028    0.0006
   -0.4486    0.7198
   -0.0068   -0.0006
   -0.0164    0.0003
    0.3606   -0.3208
   -0.0010    0.0000
   -0.5856   -0.2029
   -0.5583   -0.5801
```

% Siendo estos vectores ortonormales. Por ejemplo, el producto de los dos

```
% primeros autovectores debería ser cero.

v(1:1,:)*v(2:2,:)'

ans =
 -7.7195e-017

%Estos autovectores definen el nuevo espacio dimensional ortonormal donde
% las variables originales podrían quedar definidas mediante una rotación
% a estos nuevos ejes.
%
% Se puede observar que la traza tanto de la matriz de correlación R como
% de los autovalores es la misma.
%
trace(R)

ans =
    10

trace(lambda)

ans =
   10.0000

% Además los autovalores están ordenados en orden decreciente y ellos
% representan en su sumatoria, la varianza total. Veamos:

lambda=lambda*[1 1 1 1 1 1 1 1 1]';

lambda'

ans =
  Columns 1 through 7

    4.5427    1.2239    1.0644    0.9047    0.8054    0.3364    0.1211

  Columns 8 through 9

    0.15      0.0000
```

% por otro lado, utilizando los componentes principales, nosotros
% podemos estimar los autovalores, autovectores y el porcentaje
% explicado por cada uno de las nuevas variables cuando ellas son
% rotadas. El comando en Matlab tiene la siguiente sintaxis:

```
help pcacov

  PCACOV  Principal Component Analysis using the covariance matrix.

  [PC, LATENT, EXPLAINED] = PCACOV(X) takes a the covariance matrix,
    X, and returns the principal components in PC, the eigenvalues of
    the covariance matrix of X in LATENT, and the percentage of the
    total variance in the observations explained by each eigenvector
    in EXPLAINED.
```

% De acuerdo a esta función comando de Matlab y utilizando la matriz de
% correlación R como matriz de entrada, tenemos
%

```
 [cp,lambda,explicacion]=pcacov(R);
```

%donde:
% cp: matriz de componentes principales,
% lambda: matriz diagonal con los autovalores de R,
% explicacion: porcentaje explicado de la varianza por cada uno de los
% componentes.
% Los componentes principales (autovectores) son:

```
cp

cp =

  Columns 1 through 7

    0.4099   -0.0231    0.0029   -0.0638    0.0429   -0.7847    0.4412
   -0.0061   -0.0748    0.8255    0.4151    0.3717   -0.0327   -0.0374
   -0.4657   -0.0444    0.0129   -0.0592    0.0467   -0.0017    0.2365
   -0.0262    0.5561    0.4731   -0.2810   -0.6186   -0.0003    0.0676
   -0.0743    0.5044   -0.2952    0.7853   -0.1093   -0.0377    0.1501
   -0.4467   -0.1113    0.0358   -0.0753    0.0564    0.1203    0.7284
    0.0685    0.6384   -0.0698   -0.3383    0.6755    0.1033    0.0391
    0.4399   -0.0847    0.0325    0.0428   -0.0416    0.5173    0.3781
   -0.4594    0.0343   -0.0148   -0.0496    0.0449   -0.2983   -0.2141
```

Columns 8 through 9

```
   -0.1171    0.0360
    0.0028    0.0006
   -0.4486    0.7198
   -0.0068   -0.0006
   -0.0164    0.0003
    0.3606   -0.3208
   -0.0010    0.0000
   -0.5856   -0.2029
   -0.5583   -0.5801
```

% Los resultados en cp y v resultan los mismos por ambos métodos.

% Del mismo modo, los autovalores son:
lambda'

ans =
Columns 1 through 7

```
    4.5427    1.2239    1.0644    0.9047    0.8054    0.3364    0.1211
```

Columns 8 through 9

```
    0.15      0.0000
```

% el porcentaje explicado por cada una de las variables al ser transformadas
% es:
explicacion'

ans =
Columns 1 through 7

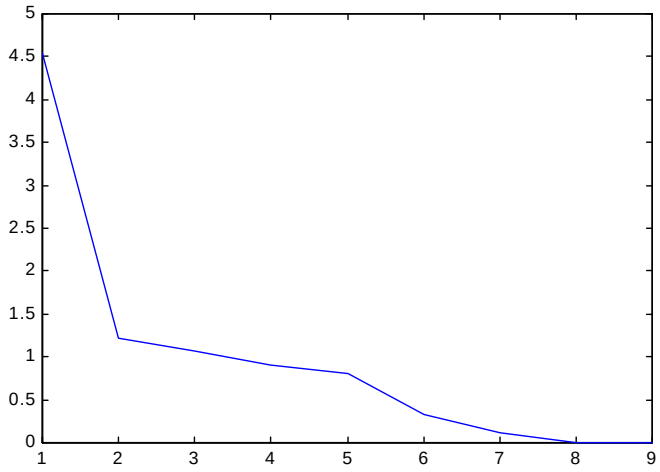
```
   50.4745   13.5986   11.8265   10.0519    8.9484    3.7375    1.3460
```

Columns 8 through 9

```
    0.166     0.0001
```

% De acuerdo a lo anterior, las tres primeras variables están
% en capacidad de explicar el 75.89% de la varianza total, de acuerdo
% al criterio de Kayser (lambda >1). Sin embargo, bajo el criterio de
% Jolliffe (lambda > 0.7, la capacidad de explicar la varianza total
% está en el orden del 94.89%. Estas referencias deben tenerse presente
% en el momento de decidir que tan eficiente se desea mejorar el nivel de
% predicción del modelo con las variables transformadas. Es decir, a
% este punto, hemos elegido los primeros k_1 componentes principales. Los

% restantes k2 componenetes serían determinados evaluando la correlación
% de la variable dependiente con las variables transformadas z. Esto será
% hecho más adelante.
% Si aprovechamos y comparamos estos valores gráficamente mediante
% el scree plot, podemos también observar que los dos primeros
% componentes serían retenidos.
% Veamos:
plot(lambda)



% Estandarización de la matriz de datos a los fines de poder rotarlos.
% Se estandariza variable por variable

```
sx1=(X(:,1:1)-ones(258,1).*mean(X(:,1:1)))./sqrt(var(X(:,1:1)));
sx2=(X(:,2:2)-ones(258,1).*mean(X(:,2:2)))./sqrt(var(X(:,2:2)));
sx3=(X(:,3:3)-ones(258,1).*mean(X(:,3:3)))./sqrt(var(X(:,3:3)));
sx4=(X(:,4:4)-ones(258,1).*mean(X(:,4:4)))./sqrt(var(X(:,4:4)));
sx5=(X(:,5:5)-ones(258,1).*mean(X(:,5:5)))./sqrt(var(X(:,5:5)));
sx6=(X(:,6:6)-ones(258,1).*mean(X(:,6:6)))./sqrt(var(X(:,6:6)));
sx7=(X(:,7:7)-ones(258,1).*mean(X(:,7:7)))./sqrt(var(X(:,7:7)));
sx8=(X(:,8:8)-ones(258,1).*mean(X(:,8:8)))./sqrt(var(X(:,8:8)));
sx9=(X(:,9:9)-ones(258,1).*mean(X(:,9:9)))./sqrt(var(X(:,9:9)));
SX=[sx1 sx2 sx3 sx4 sx5 sx6 sx7 sx8 sx9];
```

% Cálculo de las variables transformadas, mediante la rotación de sus
% ejes a los nuevos definidos por los componenetes principales cp o v.
%
%

```
Z1=SX*cp;
Z2=SX*v;
% Comprobación de que tanto cp como v son equivalentes
Z1-Z2;

who
Your variables are:
R                cp               sx3              sx9
SX               data250          sx4              u
X                explicacion      sx5              v
Z1               lambda           sx6
Z2               sx1              sx7
ans              sx2              sx8

% En este momento debemos revisar si los últimos k2 componentes se pueden
% incorporar revisando las correlación entre la variable dependiente Y y
% las nuevas variables en Z.
% Para ello utilizamos la fuente original de los datos contenidos en la
% hoja excel, debidamente transformados a formato ascii. Este archivo es
% xsyy_250.txt.

load a:XsyY_250.txt
who

Your variables are:
R                cp               sx3              sx9
SX               data250          sx4              u
X                explicacion      sx5              v
Z1               lambda           sx6              xsyy_250
Z2               sx1              sx7
ans              sx2              sx8

Y=xsyy_250(:,11:11);
size(Y)
ans =
   258     1

% Y es entonces correlacionado con Z.
% trasladamos las nuevas variables a un archivo ascii para ser leído
% desde excel.
```

%
save a:Z1.txt Z1 -ascii

Desde Excel, Se prepara una nueva hoja de datos compuesta por este archivo y los valores de Y en la última columna del archivo Zs_y_250.xls.
Haciendo uso de las herramientas de Excel, Análisis de datos, calculamos el coeficiente de correlación entre las variables Z(s) y Y.

	z1	*z2*	*z3*	*z4*	*z5*	*z6*	*z7*	*z8*	*z9*	*y*
z1	1									
z2	-5.8652E-10	1								
z3	-6.6705E-10	-7.0874E-10	1							
z4	-6.835E-10	6.7833E-10	1.7804E-09	1						
z5	8.3844E-10	-8.9969E-10	1.5748E-09	-1.435E-09	1					
z6	3.7894E-10	-1.2338E-09	-1.0961E-09	3.506E-12	9.1571E-10	1				
z7	7.1656E-10	-1.5066E-09	-1.7512E-10	-5.1295E-10	-4.9184E-11	1.3506E-09	1			
z8	9.8102E-10	2.4257E-10	-1.0774E-09	-1.0164E-09	2.4151E-09	9.3456E-12	7.3366E-10	1		
z9	9.533E-10	-2.2547E-09	-5.5642E-10	-2.1003E-09	-1.1666E-09	-1.4544E-09	1.0328E-08	3.493E-10	1	
y	-0.35907712	-0.17094046	0.24284207	-0.00511319	0.09067255	0.69279374	0.48477097	-0.2438865	-0.01328031	1

Podemos observar en la última fila, que a pesar de seleccionar los cinco primeros componentes. El componente seis y el siete tienen una buena correlación con Y. De ahí que estos dos últimos componentes, mencionados como los k_2 componentes, pueden ser incorporados como parte de los componentes definitivamente seleccionados en el proceso de reducción. Este proceso de incorporación de los últimos componentes ajustaría los componentes totales seleccionados.
Por tanto, $k=k_1+k_2=7$

Entonces la matriz definitiva de variables reducidas quedaría compuesta por los componentes z1,z2,z3,z4,z5,z6,z7.

Var_reducidas= Z1(:,1:7);

Finalmente, Estas variables reducidas más la variable Y forman el nuevo conjunto de datos para entrenamiento del modelo de redes neuronales.
En efecto, el conjunto final de datos para entrenamiento sería una mtriz compuesta por estas variables reducidas y la variable Y (variable respuesta o variable dependiente).

Var_reducidas= Z1(:,1:7);

```
» Datos = [Var_reducidas,Y];
» size(Datos)
ans =

   258     8
```

GLOSARIO DE ALGUNOS TÉRMINOS EQUIVALENTES, COMUNMENTE UTILIZADOS EN REDES NEURONALES Y EN ESTADISTICA[1]

Redes Neuronales	Estadística
Generalizando a partir de los datos	Inferencia Estadística
El conjunto de todos los casos posibles para lograr la generalización. Dominio	Población
Una función de los valores en una población. Por ejemplo, la media o un peso sináptico globalmente óptimo	Parámetro
Una función de los valores en una muestra. Pro ejemplo, la media o el peso sináptico en la fase de aprendizaje	Estadístico
Neurona, nodo, unidad computacional, unidad de procesamiento, unidad	Un elemento sencillo de computación lineal o no lineal que acepta una o más entradas, son manipulados matemáticamente mediante una función y podría dirigir el resultado a una o más neuronas
Red neuronal	Una clase de modelos flexibles de regresión no lineal o discriminante, modelos de reducción de datos, sistemas dinámicos no lineales diseñados frecuentemente con un gran número de neuronas interconectadas y organizadas en capas.
Métodos Estadísticos	Regresión Lineal, Análisis Discriminante, Búsquedas aleatorias
Arquitectura	Modelo
Construcción	Modelado
Entrenamiento, Aprendizaje, Adaptación	Estimación, Ajuste del Modelo, Optimización
Clasificación	Análisis Discriminante

1 *Extraído de Warren S. Sarle saswss@unx.sas.com Apr 29, 1996*

Redes Neuronales	Estadística
Mapping, Aproximación de Funciones	Regresión
Bias	Intercepto
La diferencia entre el valor esperado de un estadístico y su correspondiente valor real (parámetro)	Sesgo
Las reglas Delta, Adaline, Widrow-Hoff, Mínimos cuadrados	Algoritmos iterativos supervisados mediante una convergencia del error para entrenar un perceprón lineal
La regla Delta Generalizada	Algoritmos iterativos supervisados mediante una convergencia del error para entrenar un perceprón no lineal
Retropropagación (Backpropagation)	Teorema basado en la estimación para un perceptrón multicapa mediante varios algoritmos, tal como la regla Delta Generalizada
Red neuronal probabilística	Análisis Discriminante de Kernel
General Regression Neural Network	Regresión Kernel
Ciclo (Epoch)	Iteración
Entrenamiento continuo, incremental, en línea, instantáneo	Los estimados actualizan iterativamente una observación a la vez, vía ecuaciones en diferencia. Similar a una aproximación estocástica
Entrenamiento en lote, fuera de línea	Los estimados se actualizan iterativamente después de cada pasada completa sobre todos los datos como en la mayoría de los algoritmos de regresión no lineal