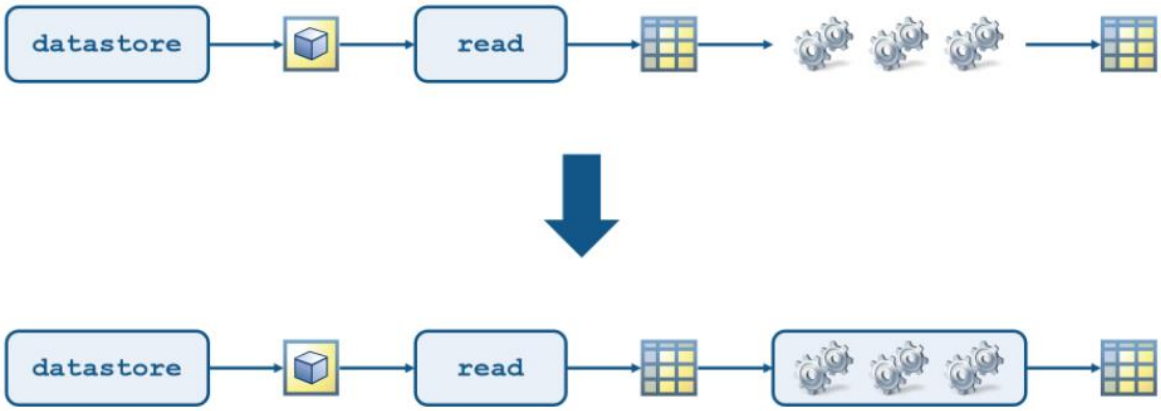
AGREGAR UNA TRANSFORMACIÓN DE DATOS

Funciones de Preprocesamiento Personalizadas

Por lo general, le interesará aplicar una serie de operaciones de preprocesamiento a cada muestra de los datos sin procesar. El primer paso para automatizar este procedimiento es crear una función personalizada que aplique sus operaciones específicas de preprocesamiento.



Actividad 1

Puede agregar una función personalizada **al final** del script o como un script **independiente**. Para el preprocesamiento de datos, la función debe tomar los datos devueltos del almacén de datos como entrada. Debería devolver los datos transformados como salida.

```
function dato_salida = nombre_función(dato_entrada)
    % hacer algo con datos entrada
    dato_salida = ...
end
```

Tarea: Cree un almacén de datos, importe los datos y visualícelos:

```
ds_letras = datastore("*_M_*.txt");
letra = read(ds_letras);
letra = escala(letra);
plot(letra.X,letra.Y)
axis equal
plot(letra.Time,letra.Y)
ylabel("Posición Vertical")
xlabel("Tiempo")
```

Cree una función llamada `escala` que realice las siguientes operaciones:

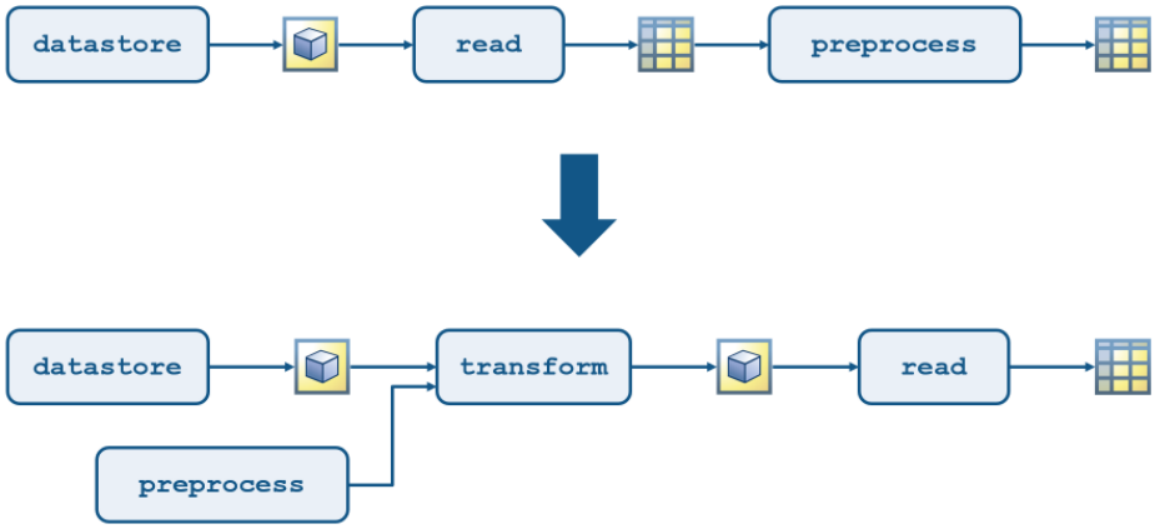
```
letra.Time = (letra.Time - letra.Time(1))/1000;
letra.X = 1.5* letra.X;
```

Debido a que estos comandos modifican la variable `letra` directamente, su función debe utilizar `letra` como variable de entrada y de salida.

Observe que la tercera línea del script inicial llama a la función `escala`. El script no se ejecutará hasta que se haya creado esta función.

Almacenes de datos transformados

Actualmente, todavía debe llamar a la función manualmente. Para automatizar la importación y el preprocesamiento de los datos, le interesa que su almacén de datos aplique esta función siempre que se lean los datos. Esto se puede hacer con un *almacén de datos transformado*. La función `transform` toma un almacén de datos y una función como entradas. Devuelve un nuevo almacén de datos como salida. Este almacén de datos transformado aplica la función proporcionada siempre que importa datos.



Actividad 2

Para utilizar una función como entrada de otra función, cree un *identificador de función* agregando el símbolo `@` al principio del nombre de la función.

```
transform(almacén,@mifunción)
```

Un identificador de función es una referencia a una función. Sin el símbolo `@`, MATLAB interpretará el nombre de la función como una llamada a esa función.

Tarea: Utilice la función `transform` para crear un almacén de datos transformado llamado `ds_trans`. Este almacén de datos debe aplicar la función `escala` a los datos a los que hace referencia `ds_letras`.

Actividad 3

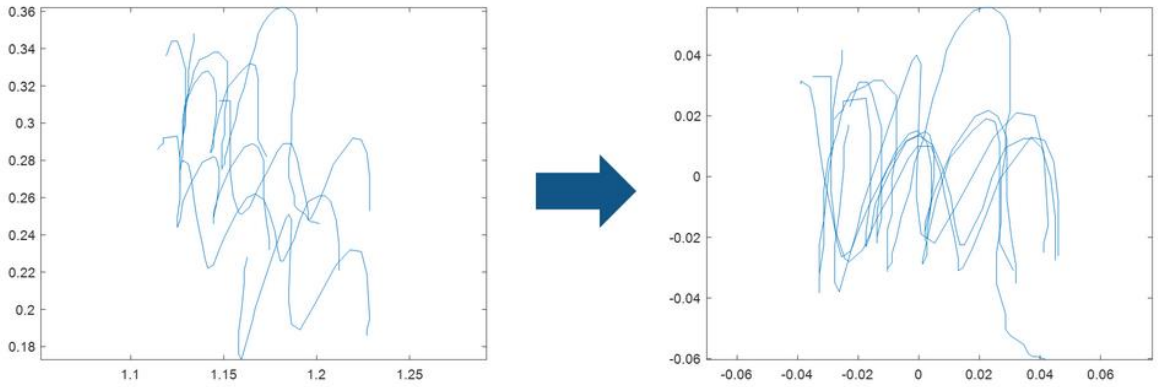
La función `escala` se debe aplicar ahora automáticamente cada vez que se lean datos del almacén de datos `ds_trans`.

Tarea: Utilice la función `readall` para importar todos los datos. Compruebe que la función de preprocesamiento se haya aplicado a todos los archivos mediante la representación de la variable `Y` como función de `Time` y visualizando todas las letras (`X` vs `Y`).

Normalizar Datos

La ubicación de una letra no es importante para clasificarla. Lo que importa es la forma. Un paso de preprocesamiento común para muchos problemas de machine learning es normalizar los datos.

Las normalizaciones habituales incluyen el desplazamiento en la media (de modo que la media de los datos desplazados sea 0) o el desplazamiento y escalado de los datos en un rango fijo (como [-1, 1]). En el caso de las letras escritas a mano, el desplazamiento de los datos de x e y para tener una media de 0 garantizará que todas las letras estén centradas alrededor del mismo punto.



Actividad 4

Tarea: Modifique la función `escala` para restar la posición media de ambos componentes:

```
letra.X = letra.X - mean(letra.X);
letra.Y = letra.Y - mean(letra.Y);
```

Tenga en cuenta que esto introducirá un problema que hará que la gráfica aparezca en blanco. Arreglará esto en la próxima tarea.

Actividad 5

Cualquier cálculo (incluido el uso predeterminado de funciones como `mean`) que incluya `NaN` resultará en `NaN`. Esto es importante en machine learning, donde a menudo se tienen valores ausentes en los datos. En los datos de escritura manual, se produce `NaN` cuando el escritor levantó el lápiz de la tableta.

Puede usar la opción `"omitnan"` para que las funciones estadísticas como `mean` ignoren los valores ausentes.

```
mean(x,"omitnan")
```

Tarea: Agregue la opción `"omitnan"` a las llamadas a `mean` en la función `escala`.

Utilice la función `readall` para importar los datos de todos los archivos a una tabla llamada `letra`. Visualice los datos mediante la representación de `Y` frente a `X`.

Tarea adicional

Intente visualizar las letras (`Y` frente a `X`) para asegurarse de que están centradas en el origen. Compruebe que el código funcione para la letra V sin ninguna modificación.

Tenga en cuenta que el tamaño de las letras no debería importar para su clasificación. Diferentes personas tienen diferentes tamaños de escritura. ¿Puede agregar más preprocesamiento para que todas las letras tengan un mismo tamaño (sin cambiar su forma)?

Archivos requeridos:

`user001_M_1.txt`

`user001_V_1.txt`

`user002_M_1.txt`

`user002_V_1.txt`

`user003_M_1.txt`

`user003_M_2.txt`

`user003_V_1.txt`

`user003_V_1.txt`

`user004_M_1.txt`

`user004_V_1.txt`

`user005_M_1.txt`

`user005_V_1.txt`