

Creating a Macro

1st option

```
data houseprice;
```

```
INPUT type$ price tax;
```

```
DATALINES;
```

```
Single 300000 0.20
```

```
Single 250000 0.25
```

```
Duplex 175000 0.15
```

```
;
```

```
RUN;
```

%MACRO somestats; //%Macro allows you to create your macro and assign it a name. somestats is the macro name.

proc means; //We are telling SAS to run the proc (procedure) called 'means'.

var price tax; //The 'var' statement allows us to tell SAS what variables we want the proc means done on.

```
RUN;
```

%MEND; //Tells SAS that we are finished making our Macro.

data housepricetwo; //We are creating a new data name..because we want a new output.

set houseprice; //We are telling SAS to read our original data set. We need to tell it that because that is where the information is taken from.

if type='Single'; //The 'if' statement allows us to only run data on a specific variables type. Instead of showing us all types of homes, we only get single homes.

```
RUN;
```

%somestats; //We are invoking our macro variable. By invoking, we are putting it into action. It's running proc means on variables price and tax.

2nd option (positional parameters)

%Macro newstats (PROG, VARS); //We open the process of creating our macro with %MACRO and we assign it a name 'newstats'. In parentheses, separated by a comma, we have the two parameters. (PROG, VARS). You can name the parameters anything. We use parameters to allow us greater flexibility.

proc &PROG; //We want to do a procedure (proc)...but instead of specifying a **particular** procedure as we did in **option 1**, we instead call our parameter using & and the parameter name 'PROG'.

var &vars; //Here we are using the 'var' statement (letting SAS know the variables we want to do procedures on)..but instead of declaring a **particular variable**, we just call our parameter (like above).

RUN;

%MEND; //Ends our macro.

%newstats (MEANS, price tax); //We are invoking our macro (&newstats). Then in parentheses, separated by a comma, we have our two parameters. Since our parameters are part of our macro, by calling our macro, we get to specify type of procedure and type of variable we want to evaluate. If we didn't have the parameters, we would not have the flexibility of choosing procedure and variables.

%newstats (Univariate, price); //Same as above.