SPSS Learning Module  
Creating and recoding variables

This module shows how to create and recode variables.  In SPSS you can create new variables with **compute** and you can modify the values of an existing variable with **recode**.

1. Computing new variables

Let's use the [auto](http://www.ats.ucla.edu/stat/spss/modules/auto.sav) data for our examples. In this section we will see how to create new variables with **compute**.

**get file 'c:\auto.sav'.**

The variable **length** contains the length of the car in inches. Below we see summary statistics for **length**.

**descriptives variables = length.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Descriptive Statistics** | | | | | |
|  | **N** | **Minimum** | **Maximum** | **Mean** | **Std. Deviation** |
| **Length (in.)** | 74 | 142 | 233 | 187.93 | 22.266 |
| **Valid N (listwise)** | 74 |  |  |  |  |

Let's use the **compute** command to make a new variable that has the length in feet instead of inches, called **lenft**.

**compute lenft = length / 12.**

**execute.**

**descriptive variables=length lenft.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Descriptive Statistics** | | | | | |
|  | **N** | **Minimum** | **Maximum** | **Mean** | **Std. Deviation** |
| **Length (in.)** | 74 | 142 | 233 | 187.93 | 22.266 |
| **LENFT** | 74 | 11.83 | 19.42 | 15.6610 | 1.85553 |
| **Valid N (listwise)** | 74 |  |  |  |  |

Suppose we wanted to make a variable called **length2** which has **length** squared.

**compute length2 = length\*\*2.**

**execute.**

**descriptive variables = length2.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Descriptive Statistics** | | | | | |
|  | **N** | **Minimum** | **Maximum** | **Mean** | **Std. Deviation** |
| **LENGTH2** | 74 | 20164.00 | 54289.00 | 35807.6892 | 8364.04524 |
| **Valid N (listwise)** | 74 |  |  |  |  |

Or we might want to make **loglen** which is the natural log of **length**.  Note that you can shorten the command **descriptive** to just **desc**, and you can shorten **variables** to **var**.

**compute loglen = ln(length).**

**execute.**

**desc var = loglen.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Descriptive Statistics** | | | | | |
|  | **N** | **Minimum** | **Maximum** | **Mean** | **Std. Deviation** |
| **LOGLEN** | 74 | 4.96 | 5.45 | 5.2290 | .12014 |
| **Valid N (listwise)** | 74 |  |  |  |  |

Let's get the mean and standard deviation of **length** and we can make Z-scores of **length**.  In SPSS there are two ways to get the z-scores, and we will show you both ways.  The first way is to use the **save** subcommand after the **descriptive** command.  This will save the z-scores into the data file.  The other way to obtain z-scores is to make them manually, and the code necessary to do that is shown below.  When making z-scores manually, you do not need to use the **save** subcommand with the **descriptive** command.

**desc variables = length**

**/save.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Descriptive Statistics** | | | | | |
|  | **N** | **Minimum** | **Maximum** | **Mean** | **Std. Deviation** |
| **Length (in.)** | 74 | 142 | 233 | 187.93 | 22.266 |
| **Valid N (listwise)** | 74 |  |  |  |  |

The mean is 187.93 and the standard deviation is 22.27, so **zlength** can be computed as shown below.

**compute zlen = (length - 187.93) / 22.27.**

**execute.**

**desc variables = zlen.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Descriptive Statistics** | | | | | |
|  | **N** | **Minimum** | **Maximum** | **Mean** | **Std. Deviation** |
| **ZLEN** | 74 | -2.06 | 2.02 | .0001 | .99984 |
| **Valid N (listwise)** | 74 |  |  |  |  |

With **compute**   
you can use + - for addition and subtraction  
you can use \* / for multiplication and division  
you can use \*\* for exponents (e.g., length\*\*2)  
you can use ( ) for controlling order of operations.

2. Recoding new variables

Suppose that we wanted to break **mpg** down into three categories.  Let's look at a table of **mpg** to see where we might draw the lines for such categories.

**frequencies variables = mpg.**

|  |  |  |
| --- | --- | --- |
| **Statistics**  Mileage (mpg) | | |
| **N** | **Valid** | 74 |
| **Missing** | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mileage (mpg)** | | | | | |
|  | | **Frequency** | **Percent** | **Valid Percent** | **Cumulative Percent** |
| **Valid** | **12** | 2 | 2.7 | 2.7 | 2.7 |
| **14** | 6 | 8.1 | 8.1 | 10.8 |
| **15** | 2 | 2.7 | 2.7 | 13.5 |
| **16** | 4 | 5.4 | 5.4 | 18.9 |
| **17** | 4 | 5.4 | 5.4 | 24.3 |
| **18** | 9 | 12.2 | 12.2 | 36.5 |
| **19** | 8 | 10.8 | 10.8 | 47.3 |
| **20** | 3 | 4.1 | 4.1 | 51.4 |
| **21** | 5 | 6.8 | 6.8 | 58.1 |
| **22** | 5 | 6.8 | 6.8 | 64.9 |
| **23** | 3 | 4.1 | 4.1 | 68.9 |
| **24** | 4 | 5.4 | 5.4 | 74.3 |
| **25** | 5 | 6.8 | 6.8 | 81.1 |
| **26** | 3 | 4.1 | 4.1 | 85.1 |
| **28** | 3 | 4.1 | 4.1 | 89.2 |
| **29** | 1 | 1.4 | 1.4 | 90.5 |
| **30** | 2 | 2.7 | 2.7 | 93.2 |
| **31** | 1 | 1.4 | 1.4 | 94.6 |
| **34** | 1 | 1.4 | 1.4 | 95.9 |
| **35** | 2 | 2.7 | 2.7 | 98.6 |
| **41** | 1 | 1.4 | 1.4 | 100.0 |
| **Total** | 74 | 100.0 | 100.0 |  |

Let's convert **mpg** into three categories to help make this more readable.  Here we convert **mpg** into three categories using **compute** and **if**.

**compute mpg3 = 1.**

**if (mpg >= 19) & (mpg <= 23) mpg3 = 2.**

**if (mpg >= 24) & (mpg <= 100) mpg3 = 3.**

**execute.**

Now, we could use **mpg3** to show a crosstab of **mpg3** by **foreign** to contrast the mileage of the foreign and domestic cars.

**crosstabs**

**/tables = mpg by mpg3.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Case Processing Summary** | | | | | | |
|  | **Cases** | | | | | |
| **Valid** | | **Missing** | | **Total** | |
| **N** | **Percent** | **N** | **Percent** | **N** | **Percent** |
| **Mileage (mpg) \* MPG3** | 74 | 100.0% | 0 | .0% | 74 | 100.0% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mileage (mpg) \* MPG3 Crosstabulation**  Count | | | | | |
|  | | **MPG3** | | | **Total** |
| **1.00** | **2.00** | **3.00** |
| **Mileage (mpg)** | **12** | 2 |  |  | 2 |
| **14** | 6 |  |  | 6 |
| **15** | 2 |  |  | 2 |
| **16** | 4 |  |  | 4 |
| **17** | 4 |  |  | 4 |
| **18** | 9 |  |  | 9 |
| **19** |  | 8 |  | 8 |
| **20** |  | 3 |  | 3 |
| **21** |  | 5 |  | 5 |
| **22** |  | 5 |  | 5 |
| **23** |  | 3 |  | 3 |
| **24** |  |  | 4 | 4 |
| **25** |  |  | 5 | 5 |
| **26** |  |  | 3 | 3 |
| **28** |  |  | 3 | 3 |
| **29** |  |  | 1 | 1 |
| **30** |  |  | 2 | 2 |
| **31** |  |  | 1 | 1 |
| **34** |  |  | 1 | 1 |
| **35** |  |  | 2 | 2 |
| **41** |  |  | 1 | 1 |
| **Total** | | 27 | 24 | 23 | 74 |

**crosstabs**

**/tables = mpg3 by foreign**

**/cells = count column.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Case Processing Summary** | | | | | | |
|  | **Cases** | | | | | |
| **Valid** | | **Missing** | | **Total** | |
| **N** | **Percent** | **N** | **Percent** | **N** | **Percent** |
| **MPG3 \* Car type** | 74 | 100.0% | 0 | .0% | 74 | 100.0% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MPG3 \* Car type Crosstabulation** | | | | | |
|  | | | **Car type** | | **Total** |
| **Domestic** | **Foreign** |
| **MPG3** | **1.00** | **Count** | 22 | 5 | 27 |
| **% within Car type** | 42.3% | 22.7% | 36.5% |
| **2.00** | **Count** | 19 | 5 | 24 |
| **% within Car type** | 36.5% | 22.7% | 32.4% |
| **3.00** | **Count** | 11 | 12 | 23 |
| **% within Car type** | 21.2% | 54.5% | 31.1% |
| **Total** | | **Count** | 52 | 22 | 74 |
| **% within Car type** | 100.0% | 100.0% | 100.0% |

The crosstab above shows that 21% of the domestic cars fall into the **high**category, while 55% of the foreign cars fit into this category.

3. Recoding variables using recode

There is an easier way to recode **mpg** to three categories using **recode**.  Using this method, we do not need to make a copy of **mpg** or use the **compute**command.  We simply use the **recode** command with the **into** option with the name of the new variable into which we want to recode **mpg**.  In this case, we will recode **mpg** into **mpg3a** using three categories: lo-18 into 1, 12-23 into 2, and 24-hi into 3.  Note the **lo** and **hi** are SPSS  keywords that can be used when we do not know the lowest or the highest values of the variable.

**recode mpg (lo thru 18=1) (19 thru 23=2) (24 thru hi=3) into mpg3a.**

**execute.**

Let's double check to see that this worked correctly.  We see that it worked perfectly.

**crosstabs**

**/tables = mpg by mpg3a.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Case Processing Summary** | | | | | | |
|  | **Cases** | | | | | |
| **Valid** | | **Missing** | | **Total** | |
| **N** | **Percent** | **N** | **Percent** | **N** | **Percent** |
| **Mileage (mpg) \* MPG3A** | 74 | 100.0% | 0 | .0% | 74 | 100.0% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mileage (mpg) \* MPG3A Crosstabulation**  Count | | | | | |
|  | | **MPG3A** | | | **Total** |
| **1.00** | **2.00** | **3.00** |
| **Mileage (mpg)** | **12** | 2 |  |  | 2 |
| **14** | 6 |  |  | 6 |
| **15** | 2 |  |  | 2 |
| **16** | 4 |  |  | 4 |
| **17** | 4 |  |  | 4 |
| **18** | 9 |  |  | 9 |
| **19** |  | 8 |  | 8 |
| **20** |  | 3 |  | 3 |
| **21** |  | 5 |  | 5 |
| **22** |  | 5 |  | 5 |
| **23** |  | 3 |  | 3 |
| **24** |  |  | 4 | 4 |
| **25** |  |  | 5 | 5 |
| **26** |  |  | 3 | 3 |
| **28** |  |  | 3 | 3 |
| **29** |  |  | 1 | 1 |
| **30** |  |  | 2 | 2 |
| **31** |  |  | 1 | 1 |
| **34** |  |  | 1 | 1 |
| **35** |  |  | 2 | 2 |
| **41** |  |  | 1 | 1 |
| **Total** | | 27 | 24 | 23 | 74 |

4. Recodes with if

Let's create a variable called **mpgfd** that assesses the mileage of the cars with respect to their origin.  This variable, **mpgfd**, will have two values:  
  
0 if below the median mpg for its group (foreign/domestic)  
1 if at/above the median mpg for its group (foreign/domestic).

**sort cases by foreign.**

**examine variables = mpg by foreign**

**/plot none**

**/compare group**

**/ percentiles (5,10,25,50,75,95) haverage.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Case Processing Summary** | | | | | | |
|  | **Cases** | | | | | |
| **Valid** | | **Missing** | | **Total** | |
| **N** | **Percent** | **N** | **Percent** | **N** | **Percent** |
| **Mileage (mpg)** | 74 | 100.0% | 0 | .0% | 74 | 100.0% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Descriptives** | | | | |
|  | | | **Statistic** | **Std. Error** |
| **Mileage (mpg)** | **Mean** | | 21.30 | .673 |
| **95% Confidence Interval for Mean** | **Lower Bound** | 19.96 |  |
| **Upper Bound** | 22.64 |  |
| **5% Trimmed Mean** | | 20.92 |  |
| **Median** | | 20.00 |  |
| **Variance** | | 33.472 |  |
| **Std. Deviation** | | 5.786 |  |
| **Minimum** | | 12 |  |
| **Maximum** | | 41 |  |
| **Range** | | 29 |  |
| **Interquartile Range** | | 7.25 |  |
| **Skewness** | | .968 | .279 |
| **Kurtosis** | | 1.130 | .552 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Percentiles** | | | | | | | |
|  | | **Percentiles** | | | | | |
| **5** | **10** | **25** | **50** | **75** | **95** |
| **Weighted Average(Definition 1)** | **Mileage (mpg)** | 14.00 | 14.00 | 17.75 | 20.00 | 25.00 | 34.25 |
| **Tukey's Hinges** | **Mileage (mpg)** |  |  | 18.00 | 20.00 | 25.00 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Case Processing Summary** | | | | | | | |
|  | | **Cases** | | | | | |
| **Valid** | | **Missing** | | **Total** | |
|  | **Car type** | **N** | **Percent** | **N** | **Percent** | **N** | **Percent** |
| **Mileage (mpg)** | **Domestic** | 52 | 100.0% | 0 | .0% | 52 | 100.0% |
| **Foreign** | 22 | 100.0% | 0 | .0% | 22 | 100.0% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Descriptives** | | | | | |
|  | **Car type** | | | **Statistic** | **Std. Error** |
| **Mileage (mpg)** | **Domestic** | **Mean** | | 19.83 | .658 |
| **95% Confidence Interval for Mean** | **Lower Bound** | 18.51 |  |
| **Upper Bound** | 21.15 |  |
| **5% Trimmed Mean** | | 19.60 |  |
| **Median** | | 19.00 |  |
| **Variance** | | 22.499 |  |
| **Std. Deviation** | | 4.743 |  |
| **Minimum** | | 12 |  |
| **Maximum** | | 34 |  |
| **Range** | | 22 |  |
| **Interquartile Range** | | 5.75 |  |
| **Skewness** | | .794 | .330 |
| **Kurtosis** | | .612 | .650 |
| **Foreign** | **Mean** | | 24.77 | 1.410 |
| **95% Confidence Interval for Mean** | **Lower Bound** | 21.84 |  |
| **Upper Bound** | 27.70 |  |
| **5% Trimmed Mean** | | 24.48 |  |
| **Median** | | 24.50 |  |
| **Variance** | | 43.708 |  |
| **Std. Deviation** | | 6.611 |  |
| **Minimum** | | 14 |  |
| **Maximum** | | 41 |  |
| **Range** | | 27 |  |
| **Interquartile Range** | | 8.25 |  |
| **Skewness** | | .706 | .491 |
| **Kurtosis** | | .468 | .953 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Percentiles** | | | | | | | | |
|  | | | **Percentiles** | | | | | |
|  | **Car type** |  | **5** | **10** | **25** | **50** | **75** | **95** |
| **Weighted Average(Definition 1)** | **Mileage (mpg)** | **Domestic** | 13.30 | 14.00 | 16.25 | 19.00 | 22.00 | 29.35 |
| **Foreign** | 14.45 | 17.00 | 20.25 | 24.50 | 28.50 | 40.10 |
| **Tukey's Hinges** | **Mileage (mpg)** | **Domestic** |  |  | 16.50 | 19.00 | 22.00 |  |
| **Foreign** |  |  | 21.00 | 24.50 | 28.00 |  |

We see that the median is 19.00 for the domestic (foreign=0) cars and 24.50 for the foreign (foreign=1) cars.  The **compute** and **recode** commands below recode **mpg** into **mpgfd** based on the median for the domestic cars and the median for the foreign cars.  In this example, we show how to create a new variable with all missing values, which can then be recoded.  In SPSS, to create a new variable with all missing values, you use the **compute** command and set the new variable equal to **$sysmis**.  The SPSS system variable **$sysmis** creates system missing values.  We also use the **do if** command, which is useful when you want to recode a variable based on different values of another variable.  Remember that you will need to use an **end if** command at the end of your do-loop.

**compute mpgfd = $sysmis.**

**do if foreign = 0.**

**recode mpg (lo thru 18=0) (19 thru hi=1) into mpgfd.**

**end if.**

**do if foreign = 1.**

**recode mpg (lo thru 24=0) (25 thru hi=1) into mpgfd.**

**end if.**

**execute.**

We can check the new variables using the command below.  The recoded variable **mpgfd** looks correct.

**crosstabs**

**/tables = mpg by mpgfd.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Case Processing Summary** | | | | | | |
|  | **Cases** | | | | | |
| **Valid** | | **Missing** | | **Total** | |
| **N** | **Percent** | **N** | **Percent** | **N** | **Percent** |
| **Mileage (mpg) \* MPGFD** | 74 | 100.0% | 0 | .0% | 74 | 100.0% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mileage (mpg) \* MPGFD Crosstabulation**  Count | | | | |
|  | | **MPGFD** | | **Total** |
| **.00** | **1.00** |
| **Mileage (mpg)** | **12** | 2 |  | 2 |
| **14** | 6 |  | 6 |
| **15** | 2 |  | 2 |
| **16** | 4 |  | 4 |
| **17** | 4 |  | 4 |
| **18** | 9 |  | 9 |
| **19** |  | 8 | 8 |
| **20** |  | 3 | 3 |
| **21** | 2 | 3 | 5 |
| **22** |  | 5 | 5 |
| **23** | 3 |  | 3 |
| **24** | 1 | 3 | 4 |
| **25** |  | 5 | 5 |
| **26** |  | 3 | 3 |
| **28** |  | 3 | 3 |
| **29** |  | 1 | 1 |
| **30** |  | 2 | 2 |
| **31** |  | 1 | 1 |
| **34** |  | 1 | 1 |
| **35** |  | 2 | 2 |
| **41** |  | 1 | 1 |
| **Total** | | 33 | 41 | 74 |

Summary

Create a new variable **len\_ft** which is **length** divided by 12.

**compute len\_ft = length / 12.**

Recode**mpg** into **mpg3**, having three categories, 1 2 3, using **compute** and**if**.

**compute mpg3 = 1.**

**if (mpg >= 19) & (mpg <= 23) mpg3 = 2.**

**if (mpg >= 24) & (mpg <= 100) mpg3 = 3.**

**execute.**

Recode **mpg** into **mpg3a**, having three categories using **recode**.

**recode**  **mpg (lo thru 18=1) (19 thru 23=2) (24 thru hi=3) into mpg3a.**

**execute.**

Recode**mpg** into **mpgfd**, having two categories, but using different cutoffs for foreign and domestic cars.

**compute mpgfd = $sysmis.**

**do if foreign = 0 .**

**recode mpg (lo thru 18=0) (19 thru hi=1) into mpgfd.**

**end if.**

**do if foreign = 1.**

**recode mpg (lo thru 24=0) (25 thru hi=1) into mpgfd.**

**end if.**

**execute.**