SPSS TUTORIALS

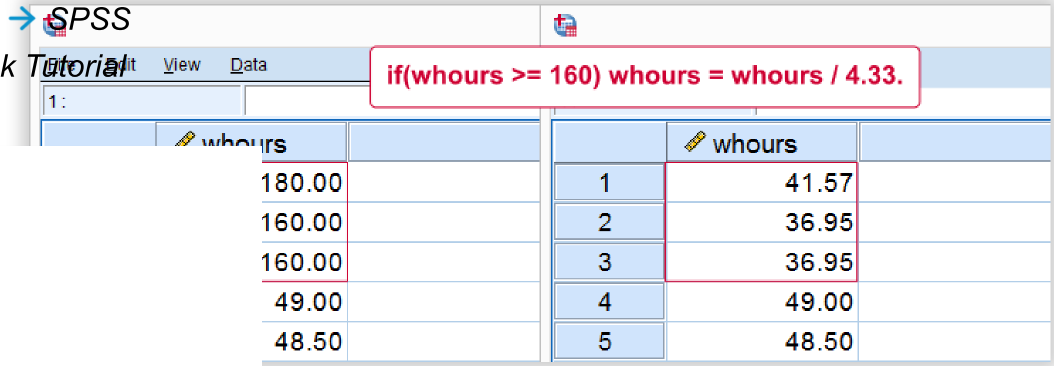
BASICS DATA ANALYSIS T-TEST ANOVA CHI-SQUARE TEST

SPSS IF – A Quick Tutorial

In SPSS, IF computes a new or existing variable for a selection of cases.

For *analyzing* a selection of cases, use FILTER or SELECT IF instead.

You are here: Home → Basics → Essential SPSS Commands → SPSS IF – A Quick Tutorial



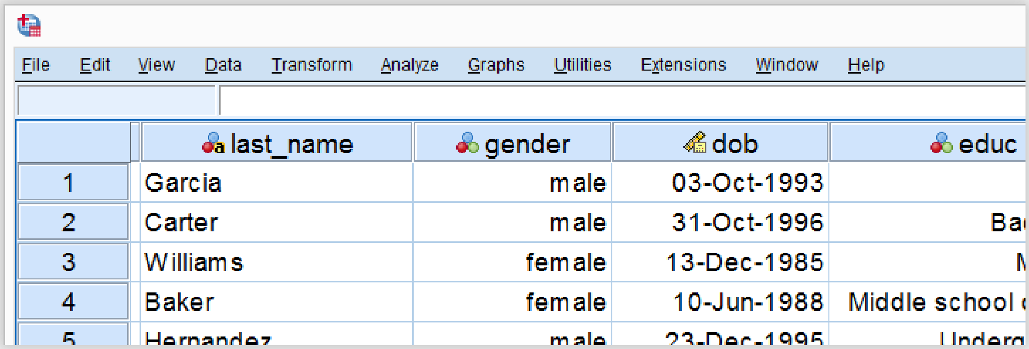
```
if(whours >= 160) whours = whours / 4.33.
```

	whours
	180.00
	160.00
	160.00
	49.00
	48.50

	whours
1	41.57
2	36.95
3	36.95
4	49.00
5	48.50

sed for Examples

All examples use bank.sav, a short survey of bank employees. Part of the data are shown below. For getting the most out of this tutorial, we recommend you download the file and try the examples for yourself.



	last_name	gender	dob	educ
1	Garcia	male	03-Oct-1993	
2	Carter	male	31-Oct-1996	Ba
3	Williams	female	13-Dec-1985	M
4	Baker	female	10-Jun-1988	Middle school
5	Hernandez	male	23-Dec-1995	Underg

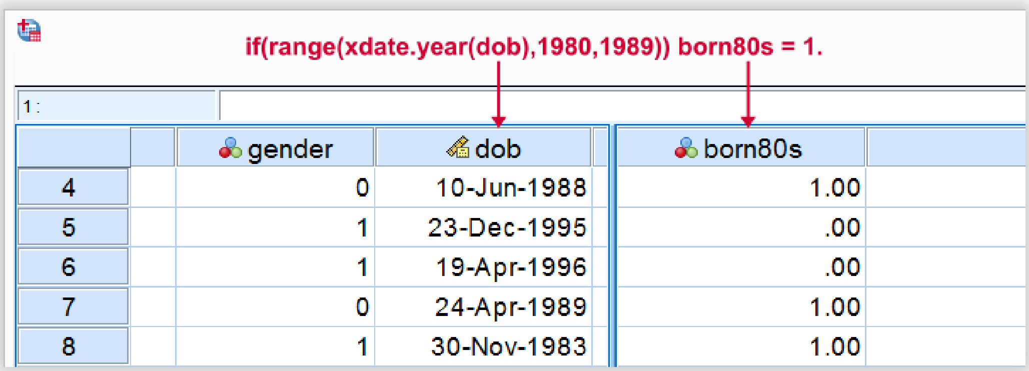
Example 1 - Flag Cases Based on Date Function

Let's flag all respondents born during the 80's. The syntax below first computes our flag variable -born80s- as a column of zeroes. We then set it to one if the year -extracted from the date of birth- is in the RANGE 1980 through 1989.

```
*Create new variable holding only zeroes.
compute born80s = 0.

*Set value to 1 if respondent born between 1980 and 1
if(range(xdate.year(dob),1980,1989)) born80s = 1.
execute.
```

```
*Optionally: add value labels.
add value labels born80s 0 'Not born during 80s' 1 'B
```

Result



if(range(xdate.year(dob),1980,1989)) born80s = 1.

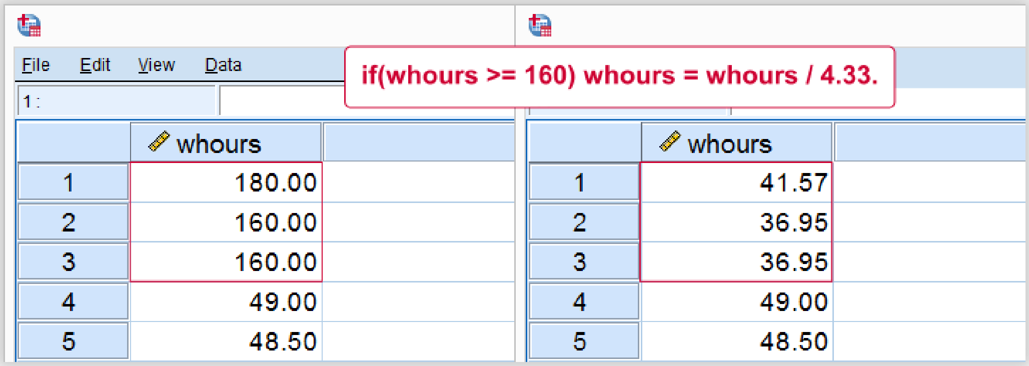
	gender	dob	born80s
4	0	10-Jun-1988	1.00
5	1	23-Dec-1995	.00
6	1	19-Apr-1996	.00
7	0	24-Apr-1989	1.00
8	1	30-Nov-1983	1.00

Example 2 - Replace Range of Values by Function

Next, if we'd run a histogram on weekly working hours -whours- we'd see values of 160 hours and over. However, weeks only hold (24 * 7 =) 168 hours. Even Kim Jong Un wouldn't claim he works 160 hours per week! We assume these respondents filled out their *monthly* -rather than weekly- working hours. On average, months hold (52 / 12 =) 4.33 weeks. So we'll divide weekly hours by 4.33 but only for cases scoring 160 or over.

```
*Sort cases descendingly on weekly hours.
sort cases by whours (d).

*Divide 160 or more hours by 4.33 (average weeks per
if(whours >= 160) whours = whours / 4.33.
execute.
```

Result



if(whours >= 160) whours = whours / 4.33.

	whours
1	180.00
2	160.00
3	160.00
4	49.00
5	48.50

	whours
1	41.57
2	36.95
3	36.95
4	49.00
5	48.50

Note

We could have done this correction with RECODE as well:

```
RECODE whours (160 = 36.95)(180 = 41.57).
```

Note, however, that RECODE becomes tedious insofar as we must correct more distinct values. It works reasonably for this variable but IF works great for *all* variables.

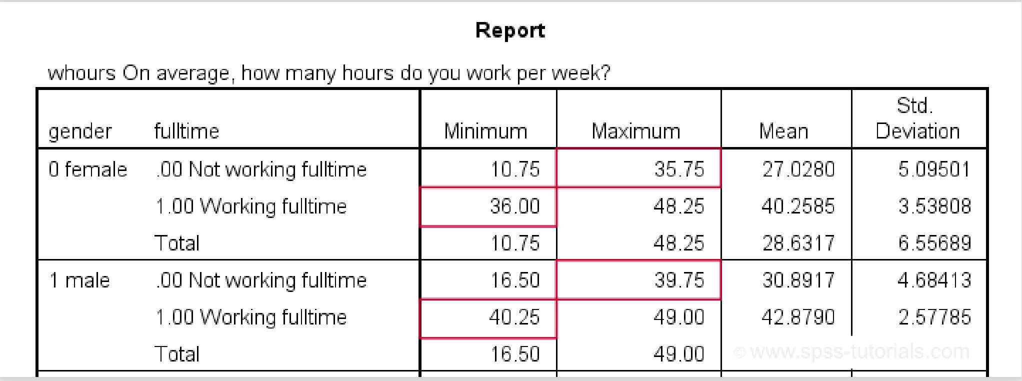
Example 3 - Compute Variable Differently Based on Ger

We'll now flag cases who work fulltime. However, "fulltime" means 40 hours for male employees and 36 hours for female employees. So we need to use different formulas based on gender. The IF command below does just that.

```
*Compute fulltime holding only zeroes.
compute fulltime = 0.

*Set fulltime to 1 if whours >= 36 for females or who
if(gender = 0 & whours >= 36) fulltime = 1.
if(gender = 1 & whours >= 40) fulltime = 1.

*Optionally, add value labels.
add value labels fulltime 0 'Not working fulltime' 1

*Quick check.
means whours by gender by fulltime
/cells min max mean stddev.
```

Result

Our syntax ends with a MEANS table showing minima, maxima, means and standard deviations per gender per group. This table -shown below- is a nice way to check the results.



Report

whours On average, how many hours do you work per week?

gender	fulltime	Minimum	Maximum	Mean	Std. Deviation
0 female	.00 Not working fulltime	10.75	35.75	27.0280	5.09501
	1.00 Working fulltime	36.00	48.25	40.2585	3.53808
	Total	10.75	48.25	28.6317	6.55689
1 male	.00 Not working fulltime	16.50	39.75	30.8917	4.68413
	1.00 Working fulltime	40.25	49.00	42.8790	2.57785
	Total	16.50	49.00		

The **maximum** for females *not* working fulltime is below 36. The **minimum** for females working fulltime is 36. And so on.

SPSS IF Versus DO IF

Some SPSS users may be familiar with DO IF. The main differences between DO IF and IF are that

- IF is a single line command while DO IF requires at least 3 lines: DO IF, some transformation(s) and END IF.
- IF is a conditional COMPUTE command whereas DO IF can affect other transformations -such as RECODE or COUNT- as well.
- If cases meet more than 1 condition, the *first* condition prevails when using DO IF - ELSE IF. If you use multiple IF commands instead, the *last* condition met by each case takes effect. The syntax below sketches this idea.

DO IF - ELSE IF Versus Multiple IF Commands

```
*DO IF: respondents meeting both conditions get resul
do if(condition_1).
result_1.
else if(condition_2). /*excludes cases meeting condit
result_2.
end if.

*IF: respondents meeting both conditions get result_2
if(condition_1) result_1.
if(condition_2) result_2. /*includes cases meeting cc
```

SPSS IF Versus RECODE

In many cases, RECODE is an easier alternative for IF. However, RECODE has more limitations too.
First off, RECODE only replaces (ranges of) constants -such as 0, 99 or system missing values- by other constants. So something like

```
recode overall (sysmis = q1).
```

is **not possible** -q1 is a variable, not a constant- but

```
if(sysmis(overall)) overall = q1.
```

works fine. You can't RECODE a function -mean, sum or whatever- into anything nor recode anything into a function. You'll need IF for doing so.

Second, RECODE can only set values based on a single variable. This is the reason why

you can't recode 2 variables into one

but you can use an IF condition involving multiple variables:

```
if(gender = 0 & whours >= 36) fulltime = 1.
```

is perfectly possible.

You can get around this limitation by combining RECODE with DO IF, however. Like so, our last example shows a different route to flag fulltime working males and females using different criteria.

Example 4 - Compute Variable Differently Based on Gender

```
*Recode whours into fulltime for everyone.
recode whours (40 thru hi = 1)(else = 0) into fulltim

*Apply different recode for female respondents.
do if(gender = 0).
recode whours (36 thru hi = 1)(else = 0) into fulltim
end if.

*Optionally, add value labels.
add value labels fulltime2 0 'Not working fulltime' 1

*Quick check.
means whours by gender by fulltime2
/cells min max mean stddev.
```

Final Notes

This tutorial presented a brief discussion of the IF command with a couple of examples. I hope you found them helpful. If I missed anything essential, please throw me a comment below.

Thanks for reading!

Let me know what you think!

Your name*

Your email address*

Your website

Your comment*

Done!

**Required field. Your comment will show up after approval from a moderator.*

This tutorial has 42 comments

By Moheeb Jouda on November 5th, 2019

Thank you Ruben for this information!

By Ruben Geert van den Berg on October 25th, 2019

Hi Moheeb!

A nice way to do this is using the ANY function.

For example:

Expand comment | all comments

1 … 9

Get In Touch!

Ruben Geert van den Berg

in LinkedIn

f Facebook

SPSS Help (Netherlands)

Sigma Plus Statistiek

www.sigma-plus-statistiek.nl

info@sigma-plus-statistiek.nl

SPSS Help (International)

SPSS tutorials

www.spss-tutorials.com

info@spss-tutorials.com