WEB APPENDIX CHAPTER 5
Mooi, E., Sarstedt, M., & Mooi-Reci, I. (2018). *Market Research. The Process, Data, and Methods using Stata*. Heidelberg: Springer.

Missing Values Analysis and Multiple Imputation in Stata

Missing Value Analysis

We use the Oddjob dataset to illustrate how to run a missing value analysis. First, let's check whether our data contains missing values, and if applicable, identify the underlying missing value pattern using Little's MCAR test. Go to ► Statistics ► Summaries, tables, and tests ► Other tables ► Tabulate missing values. In the dialog box that opens up (Fig. 5.1) tick **Report counts** and leave the box **Variables** empty to summarize all the variables in the dataset and click on **OK**.

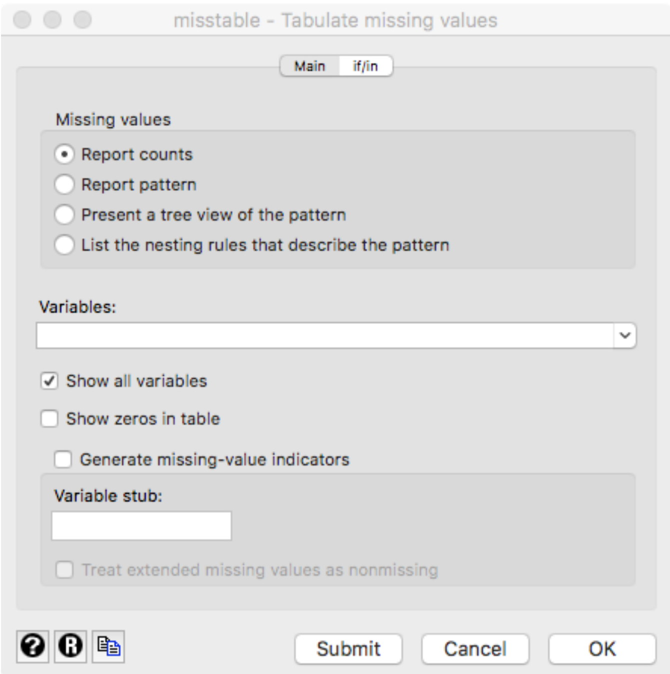

Fig. 5.1 Dialog box missing values table

Stata produces the following output in Table 5.1 solely for variables with missing values. This summarizes the total missing observations (**Obs=.**) and non-missing (**Obs<.**) values for each relevant variable. The rightmost part of the table indicates how the expectation and satisfaction variables are coded. As can be seen, all of the expectation and satisfaction variables, except for *e23* and *s23* have missing values (which explains why these are not listed in the table). While most of these variables have between 20 and 30 missing values, *e3* and *s3* ("... in case something does not work out as planned, Oddjob Airways will find a good solution.") have the most number of missing values (**111**, which corresponds to **10.4%** of the entire data). Alternatively, you can also select the option **Present a tree view of the pattern** (in Fig. 5.1) to view the missing values as a percentage of the total number of observed values. For example, item *e1* counts **27** observations as missing and **1038** as non-missing. Expressed in terms of percentages this means that **2.6%** of the observations are missing (i.e., 27/1038 = 2.6%), and Stata will round up this percentage to **3%**. Note that, this option displays the percentages of maximum 7 variables at a time, meaning that you will need to repeat this command several times to display the patterns of all variables with missing observations. This is important to establish the missing values strategy as discussed in Chap. 5 (section 5.4).

```
misstable summarize
```

				Obs<.		
Variable	Obs=.	Obs>.	Obs<.	Unique values	Min	Max
e1	27		1,038	78	1	100
e2	25		1,040	80	1	100
e3	111		954	84	1	100
e4	30		1,035	81	1	100
e5	24		1,041	85	1	100
e6	24		1,041	86	1	100
e7	17		1,048	85	1	100
e8	31		1,034	83	1	100
e9	29		1,036	71	1	100
e10	40		1,025	75	1	100
e11	20		1,045	78	1	100
e12	66		999	86	1	100
e13	34		1,031	80	1	100
e14	89		976	80	1	100
e15	29		1,036	89	1	100
e16	38		1,027	95	1	100
e17	24		1,041	81	1	100
e18	31		1,034	82	1	100
e19	52		1,013	87	1	100
e20	35		1,030	79	1	100
e21	37		1,028	80	1	100
e22	53		1,012	92	1	100
s1	27		1,038	98	1	100
s2	25		1,040	99	1	100
s3	111		954	98	1	100
s4	30		1,035	100	1	100
s5	24		1,041	98	1	100
s6	24		1,041	99	1	100
s7	17		1,048	100	1	100
s8	31		1,034	96	1	100
s9	29		1,036	76	1	100
s10	40		1,025	88	1	100
s11	20		1,045	93	1	100
s12	66		999	75	1	100
s13	34		1,031	92	1	100
s14	89		976	99	1	100
s15	29		1,036	99	1	100
s16	38		1,027	96	1	100
s17	24		1,041	97	1	100
s18	31		1,034	99	1	100
s19	52		1,013	99	1	100
s20	35		1,030	97	1	100
s21	37		1,028	96	1	100
s22	53		1,012	97	1	100

Table 5.1 Univariate statistics table

Next, let's determine whether missing values are MCAR by running Little's MCAR test. The MCAR test in Stata is a user-written package and needs to be installed first. As indicated in Chap. 5, to install this test, type directly `help mcartest` in Stata's command window. Stata will open a new window that invites you to download the user-written program onto your computer. Once the program has been installed you can carry out the test by specifying the relevant variables after the command `mcartest`. Results from Stata's output in Table 5.2, reveal that missing values are <u>not</u> MCAR because the *p*-value is < 0.05

(**Prob > chi-square** = **0.0000**).

```
mcartest e1-e22 s1-s22
note: 2 observations omitted from EM estimation because of all imputation variables
missing

       Little's MCAR test

       Number of obs       = 1063
       Chi-square distance = 4839.1940
       Degrees of freedom  = 4168
       Prob > chi-square   = 0.0000
```

Table 5.2 Output Little's MCAR test in Stata

Following the procedure outlined in Fig. 5.2 in Chap. 5, we need to carry out further tests to establish whether the missingness in variables *e1* to *e22* and *s1* to *s22* is related to another variable in the dataset. While we could principally test all variables included in our dataset, we focus on the respondents' gender. To disclose for potential relationships, we run a series of χ^2*-tests* by contrasting whether or not an observation is missing with the respondent's gender. Before proceeding with this step, we need to create a dummy variable for the missing observations in each of the variables *e1* to *e22* and *s1* to *s22*. This can be easily done by typing:

```
misstable summarize e1-e22 s1-s22, generate(miss)
```

This procedure will create automatically 44 dummy variables with the prefix *miss*. For example, missing observations for variable *e1* will be labelled as *misse1* where 1 refers to 27 missing observations as 0 to the non-missing observations, and so on. These 44 new dummy variables will appear at the bottom of your variable list.

Next, we perform a χ^2-test on respondent's gender and the 44 missing dummy variables separately. To run the χ^2-test, go to ► Statistics ► Summaries, tables, and tests ► Frequency tables ► Two-way table with measures of association. In the dialog box that opens (see Fig. 5.2), move *gender* into the **Row variable:** box and the first dummy variable *misse1* into the **Column variable:** box. Next, move on **Test Statistics**, and check the box **Pearson's chi-squared**. Initiate the analysis by clicking on **OK**.

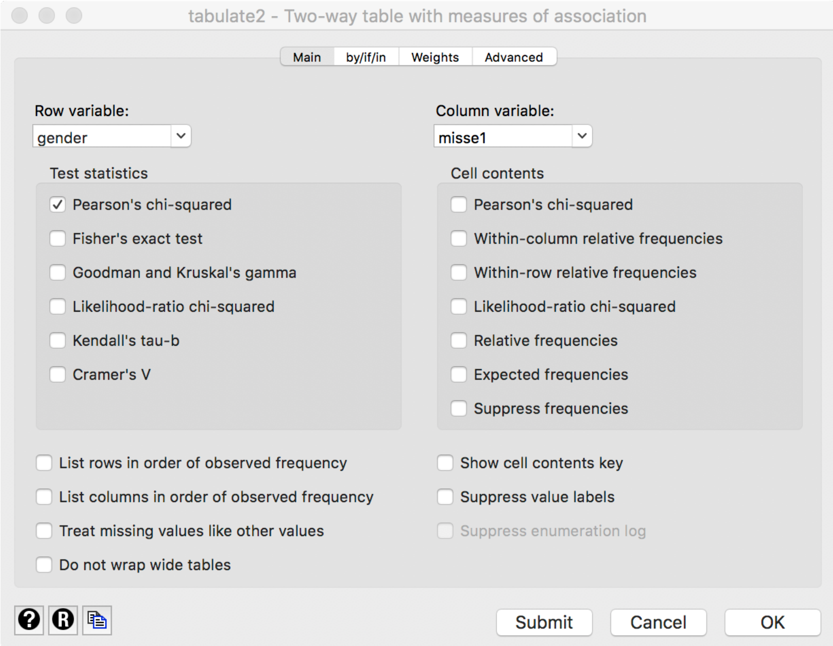

Fig 5.2 Crosstabs dialog box

The *p*-value of **0.627** in Table 5.3 indicates that there is no significant relationship between the respondents' gender and the missingness of observations in *misse1*. We would now have to repeat this test for the remaining 43 variables.

```
tabulate gender misse1, chi2

           |        (e1>=.)
    Gender |         0          1 |     Total
-----------+----------------------+----------
    female |       274          6 |       280
      male |       764         21 |       785
-----------+----------------------+----------
     Total |     1,038         27 |     1,065

          Pearson chi2(1) =   0.2367   Pr = 0.627
```

Table 5.3 χ^2-test output

Results from all separate 44 χ^2-tests (not shown here) yield significant relationships only for two variables: *misse18* and *misss18*. Considering that we carried out 44 tests at a significance level of 5%, we can expect 44·0,05≈2 erroneous rejections of (true) null hypothesis (i.e., type I errors; see Chap. 6). Hence, the two significant results in the χ^2-tests are statistically expected and we can conclude that the data are MNAR—at least with regard to the respondents' *gender*. In principle, we could proceed by testing relationships between variables with missing values and further variables such as *status* or *gender*.

Multiple Imputation

While our prior analyses indicated that the data are MNAR when considering *gender*, we nevertheless proceed by illustrating the use of multiple imputation in Stata. This technique involves the following three major steps. First, to initiate multiple imputation we need to inform Stata that we want to perform multiple imputation. To do so, we need to change the formatting of the data and type by typing:

```
mi set mlong
```

Next, we need to indicate which variables with missing observations we would like to impute. List all variables that you wish to include in your subsequent analysis. For example, if you want to run a regression (see Chap. 7) of *overall_sat* on *s1*, *s2*, *s3*, *s4*, and *s5*, you need to include these six variables in the multiple imputation procedure (Enders 2010). In addition, you should include further variables that potentially explain (or have been shown to explain; see previous step) the missingness in the variables' observations such as the respondents' demographics. In our example, we include *overall_sat*, *s1-s5*, *age*, *gender*, and *status* in the multiple imputation procedure. We therefore type the following:

```
mi register imputed overall_sat s1-s5 age gender status
```

Note that, to reproduce our results, we need to set a seed with a random number otherwise Stata will draw different samples every time it runs the imputation procedure and we will not be able to replicate our results. We randomly choose for the following 5 digits:

```
set seed 34576
```

Next, specify the number of times the missing values should be replaced (i.e., ***m*** = **5**) which will produce 5 datasets. All of these imputed datasets will be combined into one single multiple-imputation dataset which is shown in Table 5.5 below.

```
. mi impute mvn overall_sat s1-s5 age gender status, add(5) rseed (34576)
note: variables overall_sat age gender status contain no soft missing (.) values;
imputing nothing

       Performing EM optimization:
         observed log likelihood = -21411.082 at iteration 9

       Performing MCMC data augmentation ...

       Multivariate imputation                     Imputations =        5
       Multivariate normal regression                    added =        5
       Imputed: m=1 through m=5                        updated =        0

       Prior: uniform                               Iterations =      500
                                                       burn-in =      100
                                                       between =      100

       ------------------------------------------------------------------
                          |               Observations per m
                          |----------------------------------------------
                 Variable |   Complete   Incomplete   Imputed |     Total
       -------------------+-----------------------------------+----------
              overall_sat |       1065            0         0 |      1065
                       s1 |       1038           27        27 |      1065
                       s2 |       1040           25        25 |      1065
                       s3 |        954          111       111 |      1065
                       s4 |       1035           30        30 |      1065
                       s5 |       1041           24        24 |      1065
                      age |       1065            0         0 |      1065
                   gender |       1065            0         0 |      1065
                   status |       1065            0         0 |      1065
       ------------------------------------------------------------------
       (complete + incomplete = total; imputed is the minimum across m
        of the number of filled-in observations.)
```

Table 5.5 Multiple imputation output

For each variable with missing values, Table 5.5 shows the number of missing values in the original dataset and the total number of imputed values, which is simply *m* times the number of missing values. The procedure doesn't look as if it has done much for us, but in fact it has created five datasets containing imputed values that Stata saves in its memory. You can now use this imputed dataset to estimate various models, including ANOVA analysis (will be discussed in Chap. 6) or regression models (will be discussed in Chap. 7). Note that to run an regression model using this imputed dataset you will need to add the following rule (`mi estimate, dots`) before the command.

When initiating an analysis, Stata now produces an output for the pooled dataset with 5 imputations (where *Imputation*=5). Many procedures additionally support pooling of results from the analysis or multiply imputed datasets. For example, running a regression of *overall_sat* on *s1*-*s5*, will produce the outputs in Tables 5.6.

```
mi estimate, dots: reg overall_sat s1-s5

Imputations (5):
  ..... done

Multiple-imputation estimates                   Imputations       =          5
Linear regression                               Number of obs     =      1,065
                                                Average RVI       =     0.0466
                                                Largest FMI       =     0.1484
                                                Complete DF       =       1059
DF adjustment:   Small sample                   DF:     min       =     166.92
                                                        avg       =     721.66
                                                        max       =   1,050.85
Model F test:       Equal FMI                   F(   5,  867.3)   =      76.58
Within VCE type:          OLS                   Prob > F          =     0.0000

------------------------------------------------------------------------------
 overall_sat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          s1 |   .0065047   .0026442     2.46   0.014     .0013162    .0116933
          s2 |  -.0011543   .0032049    -0.36   0.719    -.0074678    .0051593
          s3 |   .0106959   .0027458     3.90   0.000     .0052748    .0161169
          s4 |   .0008919   .0026399     0.34   0.736    -.0042897    .0060734
          s5 |   .0249557   .0023864    10.46   0.000      .020273    .0296383
       _cons |   1.915659   .1292857    14.82   0.000     1.661958     2.16936
------------------------------------------------------------------------------
```

Table 5.6 Regression coefficients table using the imputed dataset

As you can see, the **Coefficients** output in Table 5.6 also shows the unstandardized coefficients along with their significances for the pooled data at the bottom of the output. As you can see, the differences in results between the original data in Table 5.7 and the pooled data in Table 5.6 are rather marginal. Even for *s3*, which had the most missing values, the unstandardized coefficient differs only at the third decimal place with no change in its significance. In the context of this regression analysis, these results suggest that we could likewise use the original data using listwise deletion.

```
reg overall_sat s1-s5

      Source |       SS           df       MS      Number of obs   =     1,565
-------------+----------------------------------   F(5, 1559)      =    132.69
       Model |  1298.39455         5  259.678911   Prob > F        =    0.0000
    Residual |  3051.09426     1,559   1.9570842   R-squared       =    0.2985
-------------+----------------------------------   Adj R-squared   =    0.2963
       Total |  4349.48882     1,564  2.78100308   Root MSE        =     1.399

------------------------------------------------------------------------------
 overall_sat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
```

s1	.0056173	.0021508	2.61	0.009	.0013986	.009836
s2	-.0014933	.0025584	-0.58	0.560	-.0065115	.0035249
s3	.0119562	.0021307	5.61	0.000	.007777	.0161355
s4	.0004428	.0021459	0.21	0.837	-.0037664	.004652
s5	.027616	.0019521	14.15	0.000	.023787	.031445
_cons	1.920074	.1091477	17.59	0.000	1.705982	2.134166

Table 5.7 Regression coefficients table using the original dataset (without imputation)