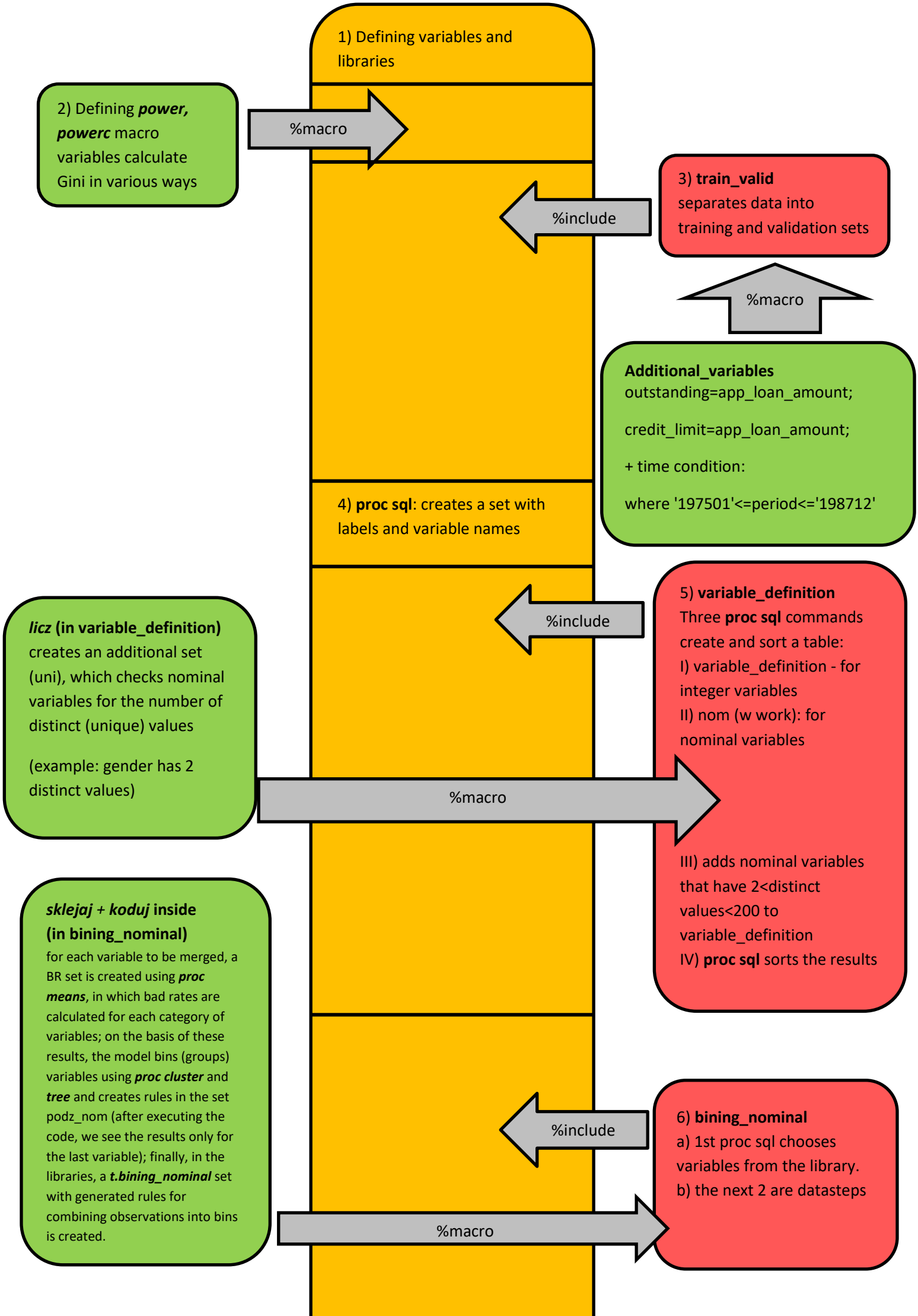
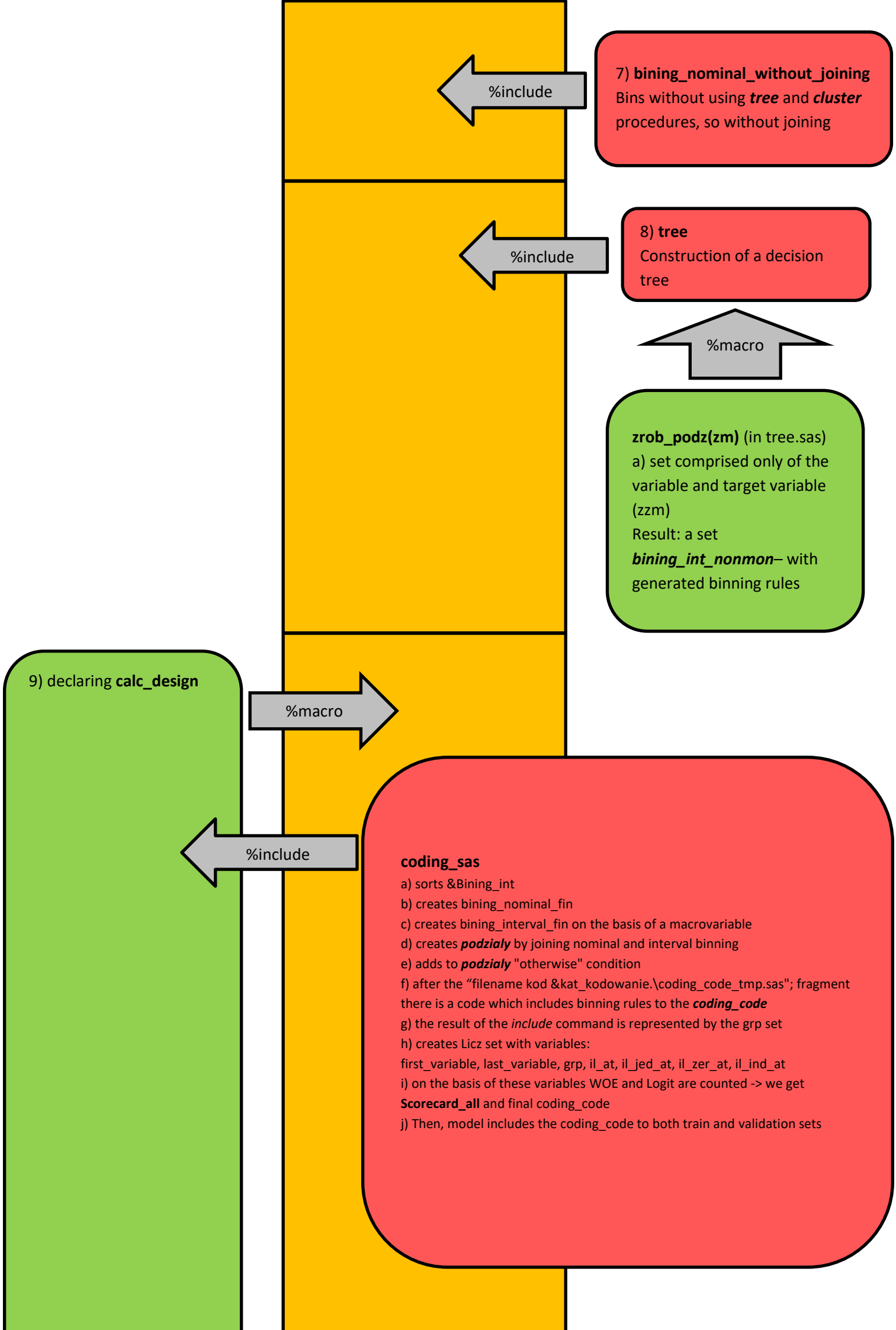
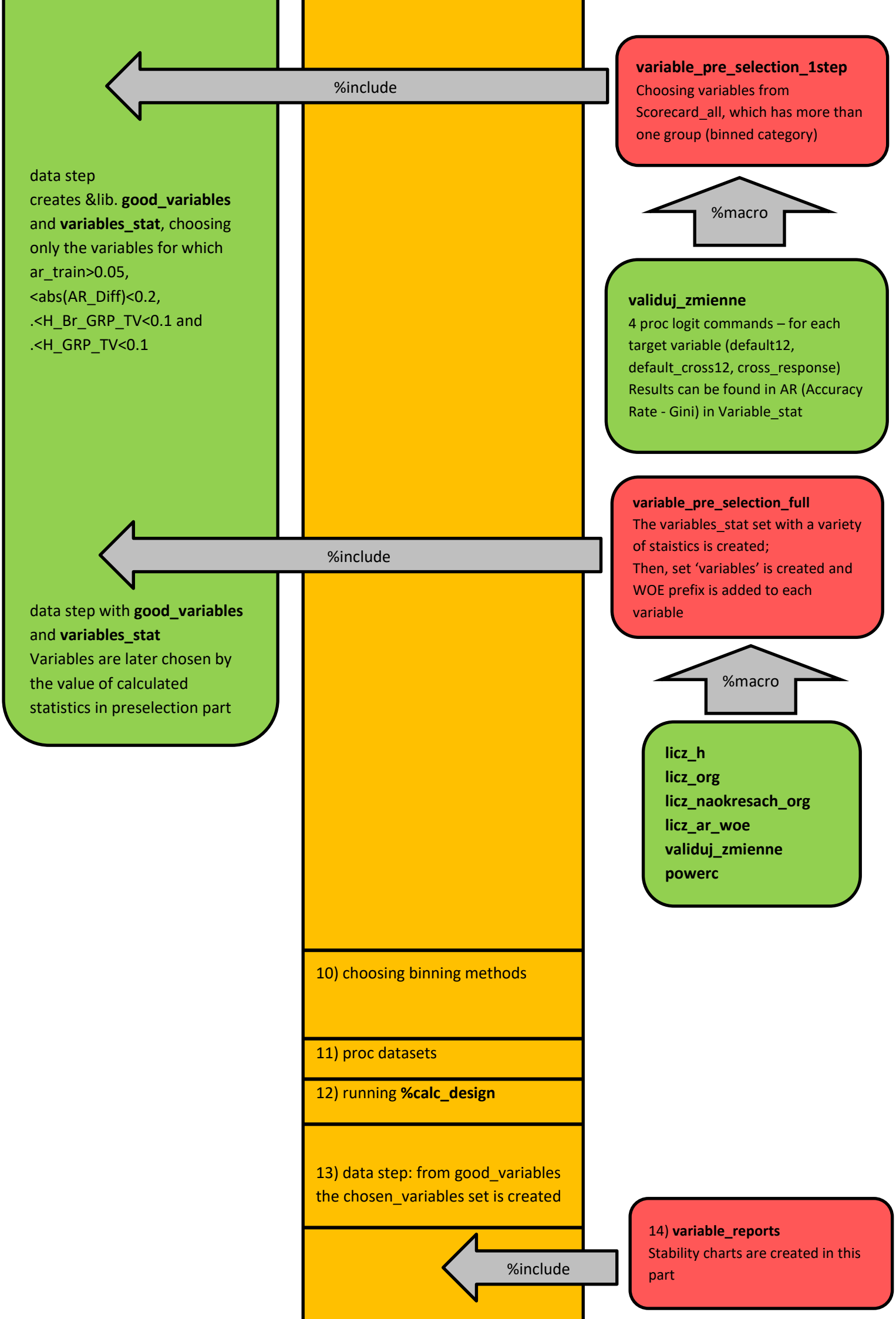
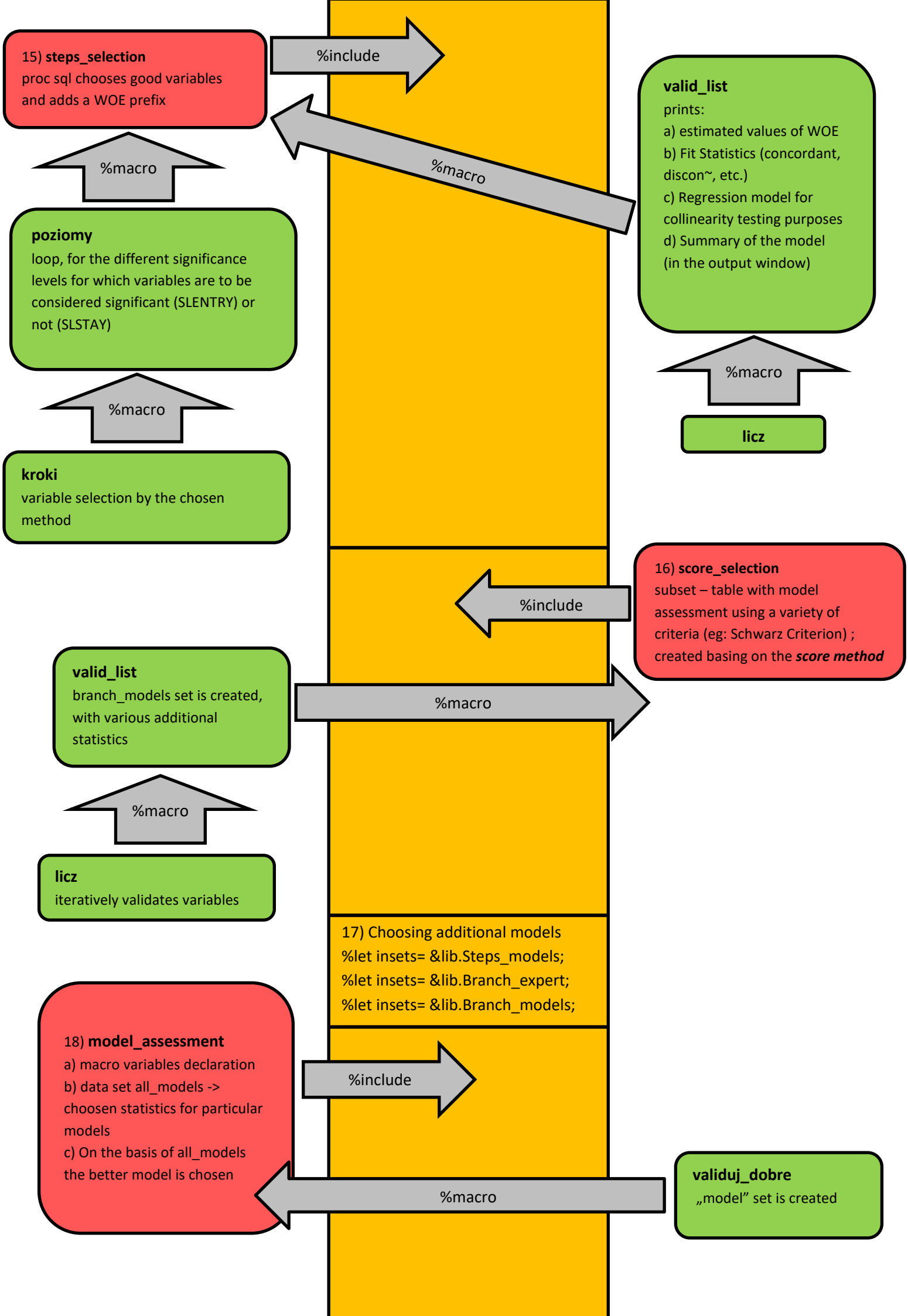
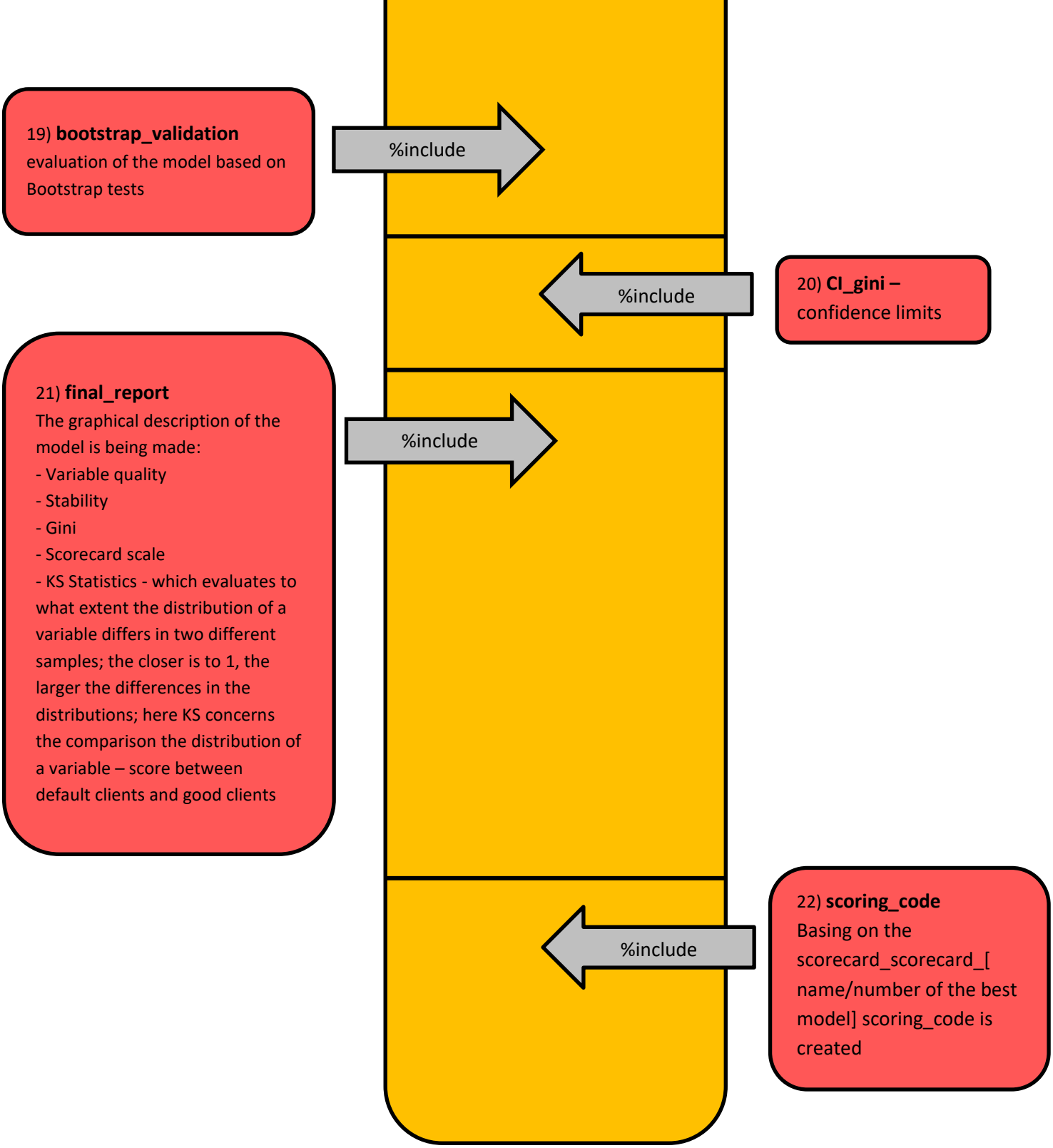

1) Defining variables and libraries
2) Defining **power, powerc** macro variables calculate Gini in various ways
%macro
3) **train_valid** separates data into training and validation sets
%include
%macro
Additional_variables
outstanding=app_loan_amount;
credit_limit=app_loan_amount;
+ time condition:
where '197501'<=period<='198712'
4) **proc sql**: creates a set with labels and variable names
%include
5) **variable_definition**
Three **proc sql** commands create and sort a table:
I) variable_definition - for integer variables
II) nom (w work): for nominal variables
***licz* (in variable_definition)** creates an additional set (uni), which checks nominal variables for the number of distinct (unique) values
(example: gender has 2 distinct values)
%macro
III) adds nominal variables that have 2<distinct values<200 to variable_definition
IV) **proc sql** sorts the results
***sklejaj* + *koduj* inside (in bining_nominal)**
for each variable to be merged, a BR set is created using ***proc means***, in which bad rates are calculated for each category of variables; on the basis of these results, the model bins (groups) variables using ***proc cluster*** and ***tree*** and creates rules in the set podz_nom (after executing the code, we see the results only for the last variable); finally, in the libraries, a ***t.bining_nominal*** set with generated rules for combining observations into bins is created.
%include
6) **bining_nominal**
a) 1st proc sql chooses variables from the library.
b) the next 2 are datasteps
%macro


%include
7) **bining_nominal_without_joining**
Bins without using ***tree*** and ***cluster*** procedures, so without joining
%include
8) **tree**
Construction of a decision tree
%macro
zrob_podz(zm) (in tree.sas)
a) set comprised only of the variable and target variable (zzm)
Result: a set ***bining_int_nonmon***– with generated binning rules
9) declaring **calc_design**
%macro
%include
coding_sas
a) sorts &Bining_int
b) creates bining_nominal_fin
c) creates bining_interval_fin on the basis of a macrovariable
d) creates ***podzialy*** by joining nominal and interval binning
e) adds to ***podzialy*** "otherwise" condition
f) after the “filename kod &kat_kodowanie.\coding_code_tmp.sas"; fragment there is a code which includes binning rules to the ***coding_code***
g) the result of the *include* command is represented by the grp set
h) creates Licz set with variables:
first_variable, last_variable, grp, il_at, il_jed_at, il_zer_at, il_ind_at
i) on the basis of these variables WOE and Logit are counted -> we get **Scorecard_all** and final coding_code
j) Then, model includes the coding_code to both train and validation sets


%include
variable_pre_selection_1step
Choosing variables from Scorecard_all, which has more than one group (binned category)
%macro
validuj_zmienne
4 proc logit commands – for each target variable (default12, default_cross12, cross_response)
Results can be found in AR (Accuracy Rate - Gini) in Variable_stat
data step
creates &lib. good_variables and variables_stat, choosing only the variables for which
ar_train>0.05,
<abs(AR_Diff)<0.2,
.<H_Br_GRP_TV<0.1 and
.<H_GRP_TV<0.1
%include
variable_pre_selection_full
The variables_stat set with a variety of staistics is created;
Then, set 'variables' is created and WOE prefix is added to each variable
%macro
licz_h
licz_org
licz_naokresach_org
licz_ar_woe
validuj_zmienne
powerc
data step with good_variables and variables_stat
Variables are later chosen by the value of calculated statistics in preselection part
10) choosing binning methods
11) proc datasets
12) running %calc_design
13) data step: from good_variables the chosen_variables set is created
%include
14) variable_reports
Stability charts are created in this part


15) steps_selection
proc sql chooses good variables and adds a WOE prefix
%include
%macro
%macro
valid_list
prints:
a) estimated values of WOE
b) Fit Statistics (concordant, discon~, etc.)
c) Regression model for collinearity testing purposes
d) Summary of the model (in the output window)
%macro
licz
poziomy
loop, for the different significance levels for which variables are to be considered significant (SLENTRY) or not (SLSTAY)
%macro
kroki
variable selection by the chosen method
16) score_selection
subset – table with model assessment using a variety of criteria (eg: Schwarz Criterion) ; created basing on the score method
%include
valid_list
branch_models set is created, with various additional statistics
%macro
%macro
licz
iteratively validates variables
17) Choosing additional models
%let insets= &lib.Steps_models;
%let insets= &lib.Branch_expert;
%let insets= &lib.Branch_models;
18) model_assessment
a) macro variables declaration
b) data set all_models -> choosen statistics for particular models
c) On the basis of all_models the better model is chosen
%include
%macro
validuj_dobre
„model” set is created


19) **bootstrap_validation**
evaluation of the model based on Bootstrap tests
%include
%include
20) **CI_gini –**
confidence limits
21) **final_report**
The graphical description of the model is being made:
- Variable quality
- Stability
- Gini
- Scorecard scale
- KS Statistics - which evaluates to what extent the distribution of a variable differs in two different samples; the closer is to 1, the larger the differences in the distributions; here KS concerns the comparison the distribution of a variable – score between default clients and good clients
%include
%include
22) **scoring_code**
Basing on the scorecard_scorecard_[name/number of the best model] scoring_code is created