# Select Subset of Rows and Columns

**back to Fan's Intro Math for Econ, Matlab Examples, or MEconTools Repositories**

## Generate a Table

```
close all;
% Generate Table 1
ar_fl_abc1 = [0.4 0.1 0.25 0.3 0.4 1 1.1];
ar_fl_abc2 = [0.4 0.1 0.2 0.3 0.4  2 2.2];
number1 = '123';
number2 = '456';
mt_data_a = [ar_fl_abc1' ar_fl_abc2'];
tb_test_a = array2table(mt_data_a);
cl_col_names_a = {['col' num2str(number1)], ['col' num2str(number2)]};
cl_row_names_a = strcat('rowA=', string((1:size(mt_data_a,1))));
tb_test_a.Properties.VariableNames = cl_col_names_a;
tb_test_a.Properties.RowNames = cl_row_names_a;
% a and b must have the same row names
cl_st_varrownames = tb_test_a.Properties.RowNames;
tb_test_a = addvars(tb_test_a, cl_st_varrownames, 'Before', 1);
% a and b must have the same row names
st_varrownames = string(cl_st_varrownames);
tb_test_a = addvars(tb_test_a, st_varrownames, 'Before', 1);
tb_test_a = addvars(tb_test_a, ["a", "b", "cc", "aa", "b", "z", "zz"]', 'Before', 1);
disp(tb_test_a);
```

| | Var1 | st_varrownames | cl_st_varrownames | col123 | col456 |
|---|---|---|---|---|---|
| rowA=1 | "a" | "rowA=1" | {'rowA=1'} | 0.4 | 0.4 |
| rowA=2 | "b" | "rowA=2" | {'rowA=2'} | 0.1 | 0.1 |
| rowA=3 | "cc" | "rowA=3" | {'rowA=3'} | 0.25 | 0.2 |
| rowA=4 | "aa" | "rowA=4" | {'rowA=4'} | 0.3 | 0.3 |
| rowA=5 | "b" | "rowA=5" | {'rowA=5'} | 0.4 | 0.4 |
| rowA=6 | "z" | "rowA=6" | {'rowA=6'} | 1 | 2 |
| rowA=7 | "zz" | "rowA=7" | {'rowA=7'} | 1.1 | 2.2 |

## Select Rows if ColX is Equal to Something

Select a subset of rows based on the variable value in one column

```
% select the rows where Var1="b"
disp(tb_test_a(strcmp(tb_test_a.Var1, "b"),:));
```

| | Var1 | st_varrownames | cl_st_varrownames | col123 | col456 |
|---|---|---|---|---|---|
| rowA=2 | "b" | "rowA=2" | {'rowA=2'} | 0.1 | 0.1 |
| rowA=5 | "b" | "rowA=5" | {'rowA=5'} | 0.4 | 0.4 |

```
% select the rows where col123=0.4
disp(tb_test_a(tb_test_a.col123==0.4,:));
```

| | Var1 | st_varrownames | cl_st_varrownames | col123 | col456 |
|---|---|---|---|---|---|

```
      rowA=1    "a"        "rowA=1"          {'rowA=1'}        0.4       0.4
      rowA=5    "b"        "rowA=5"          {'rowA=5'}        0.4       0.4
```

## Select Rows if ColX is Equal to Something or Something else

Select if the value in Var1 is either the string a or the string b, below, specify these explicitly

```
% select the rows where Var1="b" or Var1="a"
disp(tb_test_a(strcmp(tb_test_a.Var1, "b") | strcmp(tb_test_a.Var1, "a"),:));
```

```
              Var1    st_varrownames    cl_st_varrownames    col123    col456
              ____    ______________    _________________    ______    ______

      rowA=1    "a"        "rowA=1"          {'rowA=1'}        0.4       0.4
      rowA=2    "b"        "rowA=2"          {'rowA=2'}        0.1       0.1
      rowA=5    "b"        "rowA=5"          {'rowA=5'}        0.4       0.4
```

Alternatively, use matches, to find if the variable is equal to either a or b, the list of potential match is a string array.

```
% Using matches
ar_st_potential_matches = ["a", "b"];
disp(tb_test_a(matches(tb_test_a.Var1, ar_st_potential_matches),:));
```

```
              Var1    st_varrownames    cl_st_varrownames    col123    col456
              ____    ______________    _________________    ______    ______

      rowA=1    "a"        "rowA=1"          {'rowA=1'}        0.4       0.4
      rowA=2    "b"        "rowA=2"          {'rowA=2'}        0.1       0.1
      rowA=5    "b"        "rowA=5"          {'rowA=5'}        0.4       0.4
```

Now match over any a to z letters, picking up any letters a to z if they appear in column Var1.

```
% Using matches
ar_st_match_atz = string(('a':'z')');
disp(tb_test_a(matches(tb_test_a.Var1, ar_st_match_atz),:));
```

```
              Var1    st_varrownames    cl_st_varrownames    col123    col456
              ____    ______________    _________________    ______    ______

      rowA=1    "a"        "rowA=1"          {'rowA=1'}        0.4       0.4
      rowA=2    "b"        "rowA=2"          {'rowA=2'}        0.1       0.1
      rowA=5    "b"        "rowA=5"          {'rowA=5'}        0.4       0.4
      rowA=6    "z"        "rowA=6"          {'rowA=6'}          1         2
```

## Read in a Table from an Excel File

There are estimates stored in a table. Each row is a different estimation result, with a different set of estimates, for each row some fixed (not-estimated) parameter might vary. Each column represents a different parameter, or the parameter's state (initial value, estimated value, standard error, etc).

The estimatetion results file is stored in: M4Econ\table_exa\excel_exa.xlsx. We want to load in this file. Directory is one root up and one root down. The file has multiple sheets, automatically loads in the first sheet. And print table variables names, column names.

```
srn_excel_exa = 'C:\Users\fan\M4Econ\table\_exa\excel_exa.xlsx';
```

```
tb_read = readtable(srn_excel_exa);
disp((tb_read.Properties.VariableNames)');
```

```
    {'estimodelctr'       }
    {'Var1'               }
    {'FVAL'               }
    {'EXITFLAG'           }
    {'esti_iterations'    }
    {'esti_funccount'     }
    {'mean_h_sd'          }
    {'NPquad_esti'        }
    {'NPquad_se'          }
    {'NPquad_actl'        }
    {'gamma_esti'         }
    {'gamma_se'           }
    {'gamma_actl'         }
    {'lambda_esti'        }
    {'lambda_se'          }
    {'lambda_actl'        }
    {'msrErrProtSD_esti'  }
    {'msrErrProtSD_se'    }
    {'msrErrProtSD_actl'  }
    {'freePriceFrac_esti' }
    {'freePriceFrac_se'   }
    {'freePriceFrac_actl' }
    {'h_exoshk_sd_esti'   }
    {'h_exoshk_sd_se'     }
    {'h_exoshk_sd_actl'   }
    {'h_endoshk_sd_esti'  }
    {'h_endoshk_sd_se'    }
    {'h_endoshk_sd_actl'  }
    {'parm_sk_mean_init'  }
    {'parm_sk_sd_init'    }
    {'NPquad_init'        }
    {'gamma_init'         }
    {'HAquad_init'        }
    {'theta_init'         }
    {'lambda_init'        }
    {'msrErrProtSD_init'  }
    {'logProt_init'       }
    {'freePriceFrac_init' }
    {'h_exoshk_sd_init'   }
    {'h_endoshk_sd_init'  }
    {'prod_hgt0_coef_init'}
    {'prod_prot_coef_init'}
    {'prod_cons_coef_init'}
    {'prod_male_coef_init'}
    {'prod_wgt0_coef_init'}
    {'endoshkCount'       }
    {'guasshermite'       }
    {'len_curEstiParam'   }
    {'fixedVarIndex'      }
    {'esti_method'        }
    {'esti_option_type'   }
    {'subset_iter_rounds' }
    {'lambda_frac_disc'   }
```

## Select Table Columns based on Column Name Strings

Given the table that we loaded in above, select only the columns that start with some string like "gamma", or columns that end with certain strings, like "_esti".

The startsWith, contains, and endsWith are string functions that generate logical arrays based on which elements of the tring array satisfies the criteria. So this is not a table function, it is a string function.

```
ar_st_col_names = tb_read.Properties.VariableNames;
ar_st_col_names_prod = ar_st_col_names(startsWith(ar_st_col_names, 'prod_'));
ar_st_col_names_esti = ar_st_col_names(endsWith(ar_st_col_names, '_esti'));
ar_st_col_names_sd = ar_st_col_names(contains(ar_st_col_names, '_sd_'));
disp(ar_st_col_names_prod');
```

```
    {'prod_hgt0_coef_init'}
    {'prod_prot_coef_init'}
    {'prod_cons_coef_init'}
    {'prod_male_coef_init'}
    {'prod_wgt0_coef_init'}
```

```
disp(ar_st_col_names_esti');
```

```
    {'NPquad_esti'       }
    {'gamma_esti'        }
    {'lambda_esti'       }
    {'msrErrProtSD_esti' }
    {'freePriceFrac_esti'}
    {'h_exoshk_sd_esti'  }
    {'h_endoshk_sd_esti' }
```

```
disp(ar_st_col_names_sd');
```

```
    {'h_exoshk_sd_esti' }
    {'h_exoshk_sd_se'   }
    {'h_exoshk_sd_actl' }
    {'h_endoshk_sd_esti'}
    {'h_endoshk_sd_se'  }
    {'h_endoshk_sd_actl'}
    {'parm_sk_sd_init'  }
    {'h_exoshk_sd_init' }
    {'h_endoshk_sd_init'}
```

We can select columns that contain the string *sd* as well as *actl* in them, by considering joint conditions.

```
ar_it_select = contains(ar_st_col_names, '_sd_').*endsWith(ar_st_col_names, '_actl');
ar_st_col_names_selected = ar_st_col_names(ar_it_select==1);
disp(ar_st_col_names_selected');
```

```
    {'h_exoshk_sd_actl' }
    {'h_endoshk_sd_actl'}
```

```
% show values from selected columns
disp(tb_read(1:10, ar_st_col_names_selected));
```

| h_exoshk_sd_actl | h_endoshk_sd_actl |
|---|---|
| 0.042154 | 0.012103 |
| 0.042514 | 0.011849 |
| 0.042571 | 0.011352 |
| 0.04263 | 0.010598 |
| 0.042662 | 0.0089398 |
| 0.042664 | 0.0088495 |
| 0.042638 | 0.0078152 |
| 0.042689 | 0.0027549 |

```
0.042692            0.0024652
0.042625             0.002309
```