Container Map Display Swtich Key and Values and Subseting

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

Print Keys and Values

Define container:

```
% Define Container
param_map = containers.Map('KeyType','char', 'ValueType','any');
param_map('share_unbanked_j') = 12;
param_map('equi_r_j') = 2;
param_map('equi_w_j') = 'abc';
param_map('equi_P_j') = 1.2;
```

Print the key and values of the container:

```
param_map_keys = keys(param_map);
param_map_vals = values(param_map);
for i = 1:length(param_map)
    st_key = param_map_keys{i};
    ob_val = param_map_vals{i};
    st_display = strjoin(['pos =' num2str(i) '; key =' string(st_key) '; val =' string(ob_val)]
    disp(st_display);
end
```

```
pos = 1 ; key = equi_P_j ; val = 1.2
pos = 2 ; key = equi_r_j ; val = 2
pos = 3 ; key = equi_w_j ; val = abc
pos = 4 ; key = share_unbanked_j ; val = 12
```

Given Map Switch Keys and Values

Given the container map below, switch so that keys become values and values become keys.

First, this is the map that uses strings as keys and index as values:

```
mp_param_idx = containers.Map('KeyType','char', 'ValueType','any');
mp_param_idx('parm_sk_mean') = 1;
mp_param_idx('parm_sk_sd') = 2;
mp_param_idx('NPquad') = 3;
mp_param_idx('gamma') = 4;
mp_param_idx('HAquad') = 5;
mp_param_idx('theta') = 6;
```

Second, get the keys and the values, convert the values to string:

```
param_map_paramNames = keys(mp_param_idx);
param_map_paramIndex_int = values(mp_param_idx);
% convert cell of int to cell of string
param_map_paramIndex_str = cellfun(@(idx) num2str(idx(:)), param_map_paramIndex, 'uni', 0);
```

Third, generate new Map:

```
mp_idx_params = containers.Map(param_map_paramIndex_str, param_map_paramNames);
param_map_keys = keys(mp_idx_params);
param_map_vals = values(mp_idx_params);
for i = 1:length(mp_idx_params)
    st_key = param_map_keys{i};
    ob_val = param_map_vals{i};
    st_display = strjoin(['pos =' num2str(i) '; key =' string(st_key) '; val =' string(ob_val)]
    disp(st_display);
end
```

```
pos = 1 ; key = 1 ; val = parm_sk_mean
pos = 2 ; key = 2 ; val = parm_sk_sd
pos = 3 ; key = 3 ; val = NPquad
pos = 4 ; key = 4 ; val = gamma
pos = 5 ; key = 5 ; val = HAquad
pos = 6 ; key = 6 ; val = theta
```

Overall, code together shorter:

```
% Single call to convert
mp_idx_params_oneline = containers.Map(...
    cellfun(@(idx) num2str(idx(:)), values(mp_param_idx), 'uni', 0), ...
    keys(mp_param_idx));
% Check equality
disp(['mp_idx_params_oneline==mp_idx_params:' num2str(mp_idx_params_oneline==mp_idx_params) ])
```

```
mp_idx_params_oneline==mp_idx_params:0
```

Select of Subset of Key/Values from a Container Map

There is a larger container map, I want to create a new container map, that keeps a subset of the keys/values of the full container map.

```
% Original Container map
param_map = containers.Map('KeyType','char', 'ValueType','any');
param_map('equi_r_j') = 0.05;
param_map('equi_w_j') = 1.05;
param_map('equi_P_j') = 1;
% To select a subset of keys
ls_st_keys_select = {'equi_w_j', 'equi_P_j'};
% Select
param_map_subset = containers.Map(ls_st_keys_select, values(param_map, ls_st_keys_select));
% display
disp(param_map_subset.keys);
```

```
    {'equi_P_j'}    {'equi_w_j'}
```

```
disp(param_map_subset.values);
```

```
    {[1]}    {[1.0500]}
```