Generating datatables from a directory tree

`gen_group_command.py -command datatable ...`

These details can be found from : **`gen_group_command.py -help`**

- or see: **`gen_group_command.py -hweb`**
- example section F : dataTable creation

Assume this study has 10 subjects, each with 2x2x3 betas, and the betas are from a design of (column headers and possible values):

- `visit :` **`before  after`**
- `color :` **`red     green`**
- `task  :` **`T1      T2      T3`**

Enter the sample working directory and view the sample commands:

```
cd AFNI_data6/GroupAna_cases/gen_datatables
afni_open -e ggc_dt_commands.tcsh
tcsh ggc_dt_commands.tcsh
```

Example 1. each subject has one dataset with 12 volumes

(the tcsh script generates such a data tree)

```
gen_group_command.py                          \
   -command datatable                         \
   -dsets all_results/sub*.nii.gz             \
   -dt_factor_list visit before after         \
   -dt_factor_list color red green            \
   -dt_factor_list task  T1 T2 T3             \
   -subs_betas B_R_T1 B_R_T2 B_R_T3           \
               B_G_T1 B_G_T2 B_G_T3           \
               A_R_T1 A_R_T2 A_R_T3           \
               A_G_T1 A_G_T2 A_G_T3           \
   -write_script dt_1.txt
```

Example 2. like 1, but each volume is in a separate dataset

(the tcsh script generates such a data tree)

```
gen_group_command.py                              \
    -command datatable                            \
    -dt_factor_list visit before after            \
    -dt_factor_list color red green               \
    -dt_factor_list task  T1 T2 T3                \
    -dsets all_results/data.B_R_T1/sub*.gz        \
    -dsets all_results/data.B_R_T2/sub*.gz        \
    -dsets all_results/data.B_R_T3/sub*.gz        \
    -dsets all_results/data.B_G_T1/sub*.gz        \
    -dsets all_results/data.B_G_T2/sub*.gz        \
    -dsets all_results/data.B_G_T3/sub*.gz        \
    -dsets all_results/data.A_R_T1/sub*.gz        \
    -dsets all_results/data.A_R_T2/sub*.gz        \
    -dsets all_results/data.A_R_T3/sub*.gz        \
    -dsets all_results/data.A_G_T1/sub*.gz        \
    -dsets all_results/data.A_G_T2/sub*.gz        \
    -dsets all_results/data.A_G_T3/sub*.gz        \
    -write_script dt_2.txt
```

Example 3. like 2, but include a table with subject attributes

(the tcsh script generates such a table)

- goal: combine an existing table with lines for the data volumes
- each subject line is duplicated for every data volume line
- the tcsh script generates **`subject_attrs.tsv`**

Subj	Group	ValA	ValB
sub-0044	G_0044	VA_0044	VB_0044
sub-0046	G_0046	VA_0046	VB_0046
sub-0049	G_0049	VA_0049	VB_0049
sub-0053	G_0053	VA_0053	VB_0053
sub-0060	G_0060	VA_0060	VB_0060
sub-0061	G_0061	VA_0061	VB_0061
sub-0064	G_0064	VA_0064	VB_0064
sub-0073	G_0073	VA_0073	VB_0073
sub-0075	G_0075	VA_0075	VB_0075
sub-0076	G_0076	VA_0076	VB_0076

For example, imagine:

```
Group : horse, rhino
ValA  : age
ValB  : grumpiness rating
```
(beware correlation!)

Example 3. like 2, but include the exiting table via **-dt_tsv**

```
gen_group_command.py                                \
   -command datatable                               \
   -dt_tsv subject_attrs.tsv                        \
   -dt_factor_list visit before after               \
   -dt_factor_list color red green                  \
   -dt_factor_list task  T1 T2 T3                   \
   -dsets all_results/data.B_R_T1/sub*.gz           \
   -dsets all_results/data.B_R_T2/sub*.gz           \
   -dsets all_results/data.B_R_T3/sub*.gz           \
   -dsets all_results/data.B_G_T1/sub*.gz           \
   -dsets all_results/data.B_G_T2/sub*.gz           \
   -dsets all_results/data.B_G_T3/sub*.gz           \
   -dsets all_results/data.A_R_T1/sub*.gz           \
   -dsets all_results/data.A_R_T2/sub*.gz           \
   -dsets all_results/data.A_R_T3/sub*.gz           \
   -dsets all_results/data.A_G_T1/sub*.gz           \
   -dsets all_results/data.A_G_T2/sub*.gz           \
   -dsets all_results/data.A_G_T3/sub*.gz           \
   -write_script dt_3.txt
```