Day 2 - Exercises CKM with R (Solutions)

ESTP Course on SDC Methods and Tools for Census 2021

Tobias Enderle

January 28, 2021

Contents

Requirements for the exercises

```
library(data.table)
library(cellKey)
library(ptable)
```

Set the working directory

```
# for example
setwd("C:/.../ESTPcourse/")
```

You will need the following data for the exercise:

```
dat <- fread("test_data_10k.csv.gz")
```

Exercises on the *ptable*-Package

Exercise (1)

To answer the following questions (a) to (e) try to remember which part of `ptab1` could be useful. You could also use a graphic to answer the question.

Question: What will be the noise and the target frequency count after perturbation if you assume ...

(1a) ... a frequency count of 1 and a cell-key of 0.2513548301578?

(1b) ... a frequency count of 1 and a cell-key of 0.97333333?

(1c) ... a frequency count of 970 and a cell-key of 0.70548315646?

(1d) ... a frequency count of 3 and a cell-key of 1.0000000000?

(1e) ... a frequency count of 0 and a cell-key of 0.5012415871?

Hint: Either use the graphical view `plot(object, type='p')` or the ptable itself `object@pTable` to answer the questions.

```
ptab1 <- create_cnt_ptable(D = 2, V = 1.08, js = 1, mono = c(T,T,F,T))
```

Solution

```
ptab1@pTable
```

```
##     i j          p  v    p_int_lb   p_int_ub type
##  1: 0 0 1.00000000  0 0.00000000 1.00000000  all
##  2: 1 0 0.51333333 -1 0.00000000 0.51333333  all
##  3: 1 2 0.46000000  1 0.51333333 0.97333333  all
##  4: 1 3 0.02666667  2 0.97333333 1.00000000  all
##  5: 2 0 0.16560835 -2 0.00000000 0.16560835  all
##  6: 2 2 0.54634992  0 0.16560835 0.71195827  all
##  7: 2 3 0.24486677  1 0.71195827 0.95682504  all
##  8: 2 4 0.04317496  2 0.95682504 1.00000000  all
##  9: 3 2 0.42078468 -1 0.00000000 0.42078468  all
## 10: 3 3 0.27764596  0 0.42078468 0.69843064  all
## 11: 3 4 0.18235404  1 0.69843064 0.88078468  all
## 12: 3 5 0.11921532  2 0.88078468 1.00000000  all
## 13: 4 2 0.07394668 -2 0.00000000 0.07394668  all
## 14: 4 3 0.24421329 -1 0.07394668 0.31815997  all
## 15: 4 4 0.36368006  0 0.31815997 0.68184003  all
## 16: 4 5 0.24421329  1 0.68184003 0.92605332  all
## 17: 4 6 0.07394668  2 0.92605332 1.00000000  all
```

Answers:

(1a) 1-1=0

(1b) 1+2=3

(1c) 970+1=971

(1d) CK is 0.0000: 3-1=2

(1e) zeros won't be changed, positive CK for zero not logical

Exercise (2)

Please design a ptable object 'ptab2' with the following specifications: a maximum noise of D=8, a high variance of V=3 and a probability of 60%, that frequencies won't be changed.

Hint 1: Have a look at the help page '?pt_create_pTable'. There you can find the argument you must apply to set the probability that frequency counts won't be changed.

Hint 2: If you get warnings or the conditions aren't met, you may use the argument 'optim = …'. (Default is 'optim = 1'. An alternative is '4'.)

Useful code:

- `plot(ptab2, type = "t")`
- `ptab2@empResults`

Solution

```
ptab2 <- create_cnt_ptable(D = 8, V = 3, pstay = 0.6, optim=4)
ptab2@empResults
```

```
##    i p_mean p_var p_sum p_stay iter
## 1: 0      0     0     1    1.0    0
## 2: 1      0     3     1    0.6    1
## 3: 2      0     3     1    0.6    1
## 4: 3      0     3     1    0.6    1
## 5: 4      0     3     1    0.6    1
## 6: 5      0     3     1    0.6    1
## 7: 6      0     3     1    0.6    1
## 8: 7      0     3     1    0.6    1
## 9: 8      0     3     1    0.6    1
```

Exercise (3) [Advanced]

Design a further ptable object 'ptab3' with D=8, V=3 but different probabilities for original frequency counts: 50% for small frequency counts and 30% for the last frequency count (the symmetry case).

Remember: The arguments 'D', 'V' and 'js' are scalar input arguments. 'pstay', 'optim' and 'mono' are either scalar or vector input arguments.

Remember: The amount of different frequency counts 'i' in a ptable depends on 'D' (and 'js' which is not used in this exercise). The ptable entries of the last frequency count 'i_max' (symmetry case) will be applied for all frequencies equal or larger than 'i_max' (In the demonstration this morning, 'i_max' was 4. Thus, all frequencies in a table with values larger than 4 will be perturbed the same "way" like a 4).

Hint: Use the result from exercise (2) and extend it.

Solution

```
ptab3 <- create_cnt_ptable(D = 8, V = 3, pstay = c(0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.3), optim=4)
```

Exercises on the *cellKey*-Package

Exercise (4)

Please rerun the perturbation from the demonstration lesson and answer some questions.

```
# record keys
dat$rkey <- ck_generate_rkeys(dat = dat, seed = 123)


# dimensions and hierarchy
d_sex <-
  hier_create(
    nodes = c("1","2"),
    root = "Total"
  );

coc.m_cat <- unique(as.character(dat$COC.M))

d_coc.m <-
  hier_compute(
    inp = coc.m_cat, # inp = c("1","21","221", ...)
    dim_spec = c(1,1,1),
    root = "Total",
    method = "len"
  );


# define the table
tab <- ck_setup(
  x = dat,
  rkey = "rkey",
  dims = list(SEX = d_sex, COC.M = d_coc.m)
  )

# prepare and perturb the table
ptab_input <- ck_params_cnts(ptab = ptab1)
tab$params_cnts_set(val = ptab_input, v = "total")
```

```
## --> setting perturbation parameters for variable "total"
```

```
tab$perturb(v = "total")
```

```
## Count variable "total" was perturbed.
```

(4a) What is the maximum relative absolute distance between original and perturbed values? Give an interpretation of the value and search the table cell (give the defintion of the table cell)

Solution

```
tab$summary()
```

```
## +---------------------------------------------+
## |Utility measures for perturbed count variables|
## +---------------------------------------------+
## -- Distribution statistics of perturbations -----------------------------------
##    countvar Min Q10 Q20  Q30 Q40   Mean Median Q60 Q70 Q80 Q90 Q95  Q99 Max
## 1:    total  -2  -1  -1 -0.4   0 -0.061      0   0 0.4   1   1   1 1.68   2
##
## -- Distance-based measures -----------------------------------------------------
## v Variable: "total"
##
##       what    d1    d2    d3
##  1:    Min 0.000 0.000 0.000
##  2:    Q10 0.000 0.000 0.000
##  3:    Q20 0.000 0.000 0.000
##  4:    Q30 0.000 0.000 0.000
##  5:    Q40 0.800 0.000 0.003
##  6:   Mean 0.727 0.030 0.044
##  7: Median 1.000 0.000 0.007
##  8:    Q60 1.000 0.001 0.015
##  9:    Q70 1.000 0.002 0.024
## 10:    Q80 1.000 0.029 0.084
## 11:    Q90 1.800 0.048 0.154
## 12:    Q95 2.000 0.126 0.201
## 13:    Q99 2.000 0.404 0.284
## 14:    Max 2.000 0.500 0.318
##
## +-------------------------------------------------+
## |Utility measures for perturbed numerical variables|
## +-------------------------------------------------+
## x no numerical variables have been perturbed
```

Answer: The maximum relative absolute distance is 0.5. That means, the maximum relative error in the table is 50%. It is the table cell (SEX=1; COC.M=226) which hast been changed from 2 to 3: |(3-2)| / 2 = 0.5.

(4b) There is exactly one table cell, that has been changed by +2. Which one (give the definition of the table cell)?

Solution

```
tab$mod_cnts()
```

```
##       SEX COC.M row_nr pert       ckey countvar
##  1: Total Total     15    0 0.43562778    total
##  2: Total     1     15    0 0.55152974    total
##  3: Total     2     16    1 0.88409804    total
##  4: Total    21     13   -2 0.04609333    total
##  5: Total    22     16    1 0.83800471    total
##  6: Total   221     16    1 0.86374233    total
##  7: Total   222     14   -1 0.10036905    total
```

```
##  8: Total   223     15    0 0.39855888    total
##  9: Total   224     15    0 0.53048343    total
## 10: Total   225     15    0 0.48767352    total
## 11: Total   226     15    0 0.45717750    total
## 12:     1 Total     16    1 0.74500811    total
## 13:     1     1     14   -1 0.19558527    total
## 14:     1     2     15    0 0.54942284    total
## 15:     1    21     15    0 0.62389932    total
## 16:     1    22     16    1 0.92552352    total
## 17:     1   221     15    0 0.55461157    total
## 18:     1   222     13   -2 0.05017674    total
## 19:     1   223     14   -1 0.29954800    total
## 20:     1   224     17    2 0.98893485    total
## 21:     1   225     14   -1 0.31754903    total
## 22:     1   226      7    1 0.71470333    total
## 23:     2 Total     16    1 0.69061967    total
## 24:     2     1     15    0 0.35594447    total
## 25:     2     2     15    0 0.33467520    total
## 26:     2    21     15    0 0.42219401    total
## 27:     2    22     16    1 0.91248119    total
## 28:     2   221     14   -1 0.30913076    total
## 29:     2   222     13   -2 0.05019231    total
## 30:     2   223     14   -1 0.09901088    total
## 31:     2   224     15    0 0.54154858    total
## 32:     2   225     14   -1 0.17012449    total
## 33:     2   226     16    1 0.74247417    total
##       SEX COC.M row_nr pert       ckey countvar
```

Answer: SEX=1 and COC.M=224

Exercise (5)

Now, extend the two-dimensional table by a geographical variable.

(5a) Create the variable hierarchy/dimension for NUTS3. NUTS3 has 3 levels each of length 1. Please use 'hier_compute(...)' (similar to the hierarchy of the variable COC.M).

Solution:

```
nuts3 <- unique(as.character(dat$NUTS3))

d_nuts3 <-
  hier_compute(
    inp = nuts3,
    dim_spec = c(1,1,1),
    root = "Total",
    method = "len"
  );

hier_display(d_nuts3)
```

```
## Total
```

```
## +-1
## | +-11
## | | +-111
## | | +-112
## | | \-113
## | +-12
## | | +-121
## | | +-122
## | | +-123
## | | +-124
## | | +-125
## | | +-126
## | | \-127
## | \-13
## |   \-130
## +-2
## | +-21
## | | +-211
## | | +-212
## | | \-213
## | \-22
## |   +-221
## |   +-222
## |   +-223
## |   +-224
## |   +-225
## |   \-226
## \-3
##   +-31
##   | +-311
##   | +-312
##   | +-313
##   | +-314
##   | \-315
##   +-32
##   | +-321
##   | +-322
##   | \-323
##   +-33
##   | +-331
##   | +-332
##   | +-333
##   | +-334
##   | \-335
##   \-34
##     +-341
##     \-342
```

(5b) Update the following setup using the hierarchy of NUTS3 you created in (5a) and assign it to the object 'tab5'.

```
tab5 <- ck_setup(
  x = dat,
```

```
  rkey = "rkey",
  dims = list(SEX = d_sex, COC.M = d_coc.m)
)
```

Remark: The list for the argument 'dims = ...' must be entered case-sensitively!

Solution

```
tab5 <- ck_setup(
  x = dat,
  rkey = "rkey",
  dims = list(SEX = d_sex, COC.M = d_coc.m, NUTS3 = d_nuts3)
)
```

(5c) Question: How many cells does the newly generated 3-dimensional table have?

Solution

```
tab5
```

```
## -- Table Information -----------------------------------------------------------
## v 1584 cells in 3 dimensions ("SEX", "COC.M", "NUTS3")
## v weights: no
## -- Tabulated / Perturbed countvars ---------------------------------------------
## [ ] "total"
```

Answer: 1.584

(5d) Apply the perturbation using the ptable 'ptab2'. How many 1's are in original table and how many 1's have been changed by +8 (try to explain)?

Solution

```
ptab_input <- ck_params_cnts(ptab = ptab2)
tab5$params_cnts_set(val = ptab_input, v = "total")
```

```
## --> setting perturbation parameters for variable "total"
```

```
tab5$perturb(v = "total")
```

```
## Count variable "total" was perturbed.
```

```
tab5$freqtab(v = c("total"))[ uwc == 1,]
```

```
##        SEX COC.M NUTS3 vname uwc wc puwc pwc
##    1: Total   221   321 total   1  1    0   0
##    2: Total   222   111 total   1  1    1   1
##    3: Total   222   121 total   1  1    1   1
##    4: Total   222   123 total   1  1    1   1
##    5: Total   222   125 total   1  1    1   1
```

```
## ---
## 133:     2   226   130 total   1  1    0   0
## 134:     2   226    32 total   1  1    1   1
## 135:     2   226   322 total   1  1    1   1
## 136:     2   226   341 total   1  1    1   1
## 137:     2   226   342 total   1  1    1   1
```

```
tab5$freqtab(v = c("total"))[ uwc == 1 & puwc == 9,]
```

```
##      SEX COC.M NUTS3 vname uwc wc puwc pwc
## 1: Total   224   121 total   1  1    9   9
## 2:     1   222   124 total   1  1    9   9
## 3:     1   224   313 total   1  1    9   9
## 4:     2   222   322 total   1  1    9   9
## 5:     2   223   223 total   1  1    9   9
## 6:     2   224   121 total   1  1    9   9
## 7:     2   224   212 total   1  1    9   9
```

Answer: 7 out of 137 1's have been changed by +8 to 9.

(5e) How many (absolute or relative) cells are still original (i.e. remain unchanged) after perturbation?

Solution

```
tab5$measures_cnts(v = "total")$overview
```

```
##     noise  cnt          pct
##  1:    -8    7 0.0044191919
##  2:    -7    3 0.0018939394
##  3:    -6    7 0.0044191919
##  4:    -5   11 0.0069444444
##  5:    -4   25 0.0157828283
##  6:    -3   37 0.0233585859
##  7:    -2   68 0.0429292929
##  8:    -1   79 0.0498737374
##  9:     0 1094 0.6906565657
## 10:     1  100 0.0631313131
## 11:     2   64 0.0404040404
## 12:     3   58 0.0366161616
## 13:     4   23 0.0145202020
## 14:     5    4 0.0025252525
## 15:     6    1 0.0006313131
## 16:     7    2 0.0012626263
## 17:     8    1 0.0006313131
```

Answer: 1094 or 69%

(5f) How large are the three mean distances (utility measures) when you take original zero counts into account?

- d1: absolute distance between original and perturbed values

- d2: relative absolute distance between original and perturbed values
- d3: absolute distance between square-roots of original and perturbed values

```
tab5$freqtab(v = c("total"))[ uwc == 0]
```

```
##        SEX COC.M NUTS3 vname uwc wc puwc pwc
##   1: Total   222   113 total   0  0    0   0
##   2: Total   222   213 total   0  0    0   0
##   3: Total   222   222 total   0  0    0   0
##   4: Total   222   225 total   0  0    0   0
##   5: Total   222   321 total   0  0    0   0
##  ---
## 258:     2   226   331 total   0  0    0   0
## 259:     2   226   332 total   0  0    0   0
## 260:     2   226   333 total   0  0    0   0
## 261:     2   226   334 total   0  0    0   0
## 262:     2   226   335 total   0  0    0   0
```

```
tab5$summary()
```

```
## +-----------------------------------------------+
## |Utility measures for perturbed count variables|
## +-----------------------------------------------+
## -- Distribution statistics of perturbations -----------------------------------
##    countvar Min Q10 Q20 Q30 Q40  Mean Median Q60 Q70 Q80 Q90 Q95 Q99 Max
## 1:    total  -8  -1   0   0   0 0.037      0   0   0   0   1   3   6   8
##
## -- Distance-based measures -----------------------------------------------------
## v Variable: "total"
##
##       what    d1    d2    d3
##  1:    Min 0.000 0.000 0.000
##  2:    Q10 0.000 0.000 0.000
##  3:    Q20 0.000 0.000 0.000
##  4:    Q30 0.000 0.000 0.000
##  5:    Q40 0.000 0.000 0.000
##  6:   Mean 0.835 0.101 0.089
##  7: Median 0.000 0.000 0.000
##  8:    Q60 0.000 0.000 0.000
##  9:    Q70 1.000 0.003 0.038
## 10:    Q80 2.000 0.028 0.108
## 11:    Q90 3.000 0.143 0.283
## 12:    Q95 4.000 0.400 0.449
## 13:    Q99 6.350 1.512 1.051
## 14:    Max 8.000 8.000 2.000
##
## +---------------------------------------------------+
## |Utility measures for perturbed numerical variables|
## +---------------------------------------------------+
## x no numerical variables have been perturbed
```

```
# or
tab5$measures_cnts(v = "total")$measures
```

```
##       what    d1    d2    d3
##  1:    Min 0.000 0.000 0.000
##  2:    Q10 0.000 0.000 0.000
##  3:    Q20 0.000 0.000 0.000
##  4:    Q30 0.000 0.000 0.000
##  5:    Q40 0.000 0.000 0.000
##  6:   Mean 0.835 0.101 0.089
##  7: Median 0.000 0.000 0.000
##  8:    Q60 0.000 0.000 0.000
##  9:    Q70 1.000 0.003 0.038
## 10:    Q80 2.000 0.028 0.108
## 11:    Q90 3.000 0.143 0.283
## 12:    Q95 4.000 0.400 0.449
## 13:    Q99 6.350 1.512 1.051
## 14:    Max 8.000 8.000 2.000
```

Answer: without zeros: 0.835 0.101 0.089

Exercise (6)

(6a) Produce the table from exercise 5 again and assign it to the object 'tab6'.

```
tab6 <- ck_setup(...)
```

Hint: Don't copy the object like: tab6 <- tab5 (!! doesn't work)

Remark: If you try to perturb a table and receive the message `--> Variable "total" was already perturbed: parameters are not updated.` then you have to rerun the 'ck_setup(..)' step. you can't perturb the object twice.

Solution

```
tab6 <- ck_setup(
  x = dat,
  rkey = "rkey",
  dims = list(SEX = d_sex, COC.M = d_coc.m, NUTS3 = d_nuts3)
)
```

(6b) Perturb 'tab6' by the following ptable.

```
ptab6 <- create_cnt_ptable(D = 8, V = 2, pstay = 0.6, optim=4)
```

Solution

```
tab6$params_cnts_set(val = ck_params_cnts(ptab = ptab6), v = "total")
```

```
## --> setting perturbation parameters for variable "total"
```

```r
tab6$perturb(v = "total")
```

```
## Count variable "total" was perturbed.
```

(6c) Compare the measure "relative absolute distance" between the two different perturbations in (5) and (6). Which perturbation comes along with a lower information loss?

Solution

```r
tab5$measures_cnts(v = "total")$measures
```

```
##       what    d1    d2    d3
##  1:    Min 0.000 0.000 0.000
##  2:    Q10 0.000 0.000 0.000
##  3:    Q20 0.000 0.000 0.000
##  4:    Q30 0.000 0.000 0.000
##  5:    Q40 0.000 0.000 0.000
##  6:   Mean 0.835 0.101 0.089
##  7: Median 0.000 0.000 0.000
##  8:    Q60 0.000 0.000 0.000
##  9:    Q70 1.000 0.003 0.038
## 10:    Q80 2.000 0.028 0.108
## 11:    Q90 3.000 0.143 0.283
## 12:    Q95 4.000 0.400 0.449
## 13:    Q99 6.350 1.512 1.051
## 14:    Max 8.000 8.000 2.000
```

```r
tab6$measures_cnts(v = "total")$measures
```

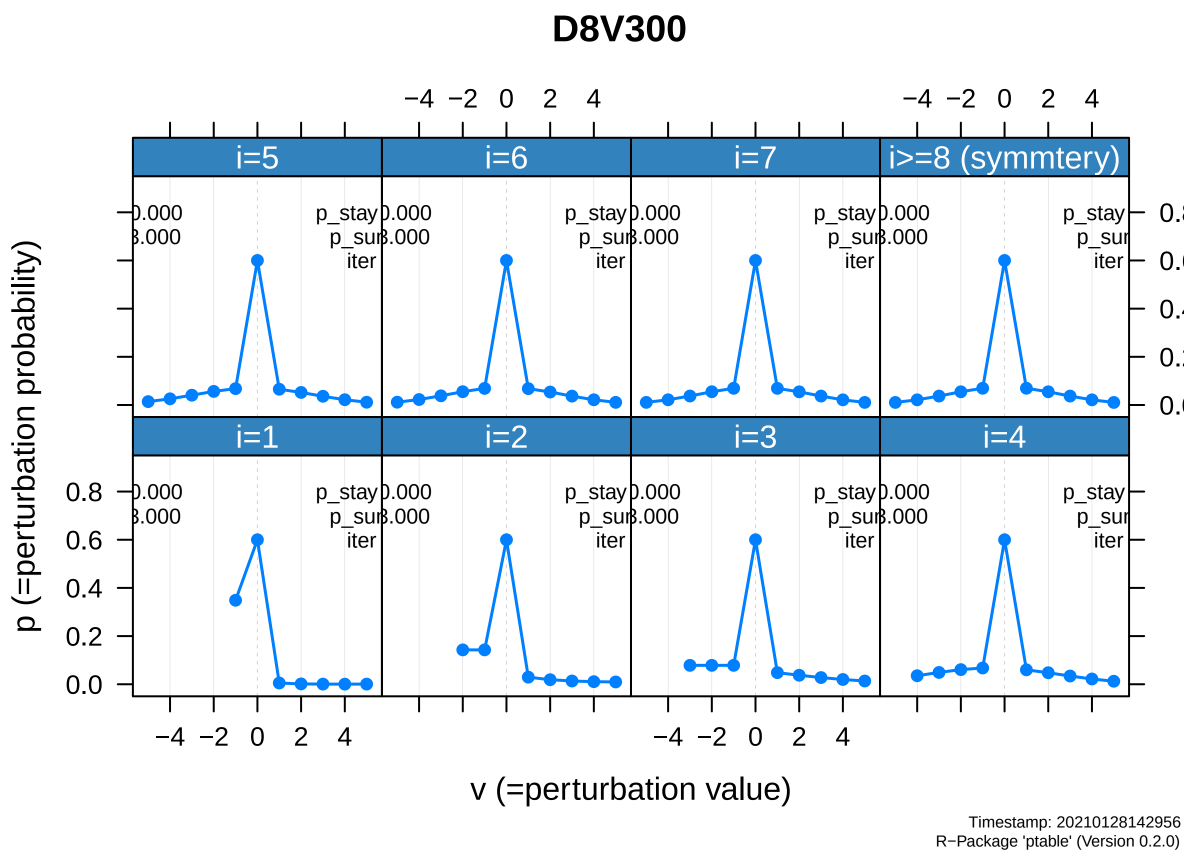
```
##       what    d1    d2    d3
##  1:    Min 0.000 0.000 0.000
##  2:    Q10 0.000 0.000 0.000
##  3:    Q20 0.000 0.000 0.000
##  4:    Q30 0.000 0.000 0.000
##  5:    Q40 0.000 0.000 0.000
##  6:   Mean 0.708 0.093 0.079
##  7: Median 0.000 0.000 0.000
##  8:    Q60 0.000 0.000 0.000
##  9:    Q70 1.000 0.004 0.036
## 10:    Q80 2.000 0.027 0.097
## 11:    Q90 2.000 0.136 0.252
## 12:    Q95 3.000 0.333 0.414
## 13:    Q99 5.000 1.550 1.000
## 14:    Max 8.000 8.000 2.000
```

(6d) [Advanced] Compare the distributions of the two ptables `ptab2` (which was used to perturb `tab5`) and `ptab6` (which was used to perturb `tab6`) and try to explain the result in (6c).

Solution

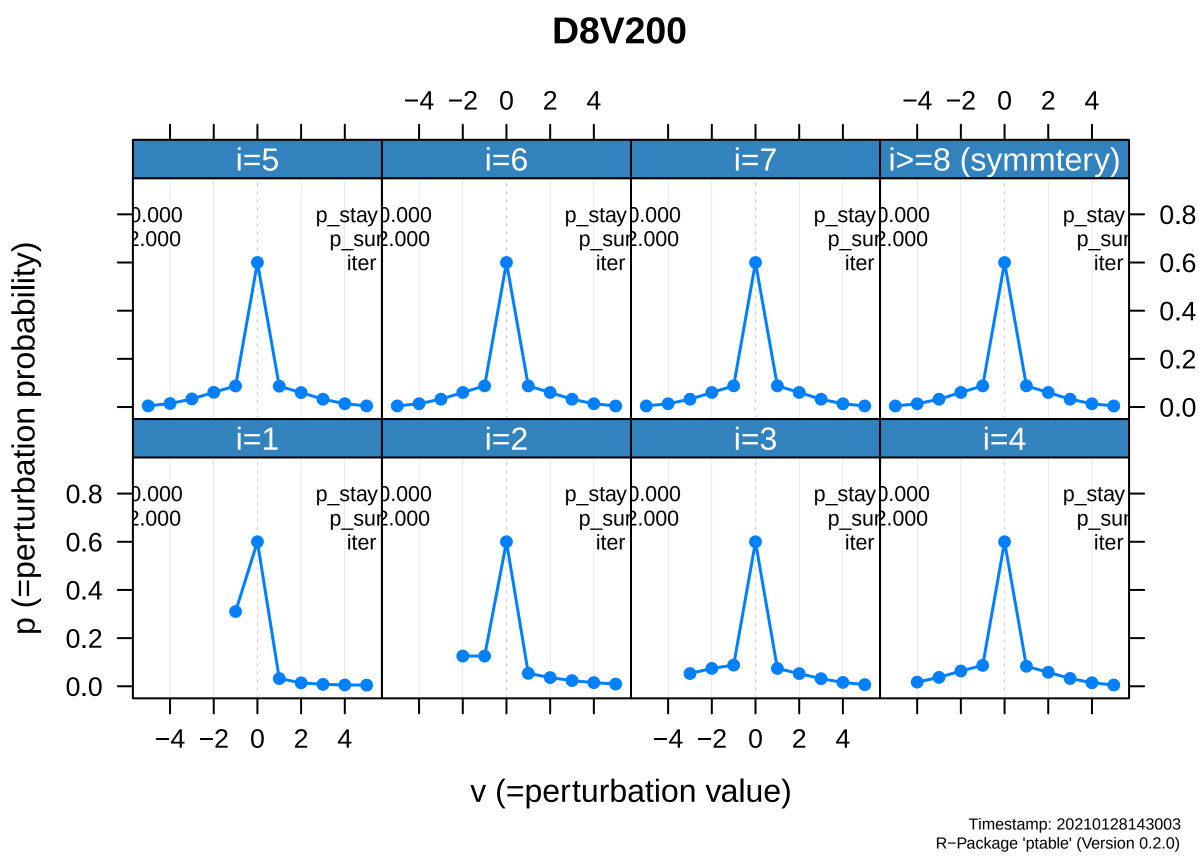
```r
plot(ptab2, type="d")
```

```
## Distribution of Perturbation Values
```




```r
plot(ptab6, type="d")
```

```
## Distribution of Perturbation Values
```



Answer: The distributions are almost identical. However, the variance that differs is the main reason.

Exercise (7) [Advanced]

(7a) Use `dat` and create a household data set (with `HID`, `LAU2`, size of the household `Size` and mean age using `AGE.H`). Then, assign a record key to each household.

Solution

```
dat <- fread("test_data_10k.csv.gz")
dat$rkey <- ck_generate_rkeys(dat = dat, seed = 123)
```

```
# compute mean age of the household
dat[, hh_mean_age:=mean(AGE.H), by=HID]

# compute cell-key for households (i.e. aggregate record keys of the household members)
dat[, hh_ckey:=sum(rkey), by=HID]

# remove integer before the decimal points (i.e. modulo operation)
dat[, hh_ckey := hh_ckey %% 1]


# household data set
hh_dat <- unique(dat, by = "HID")

## select variables
```

```r
hh_dat <- hh_dat[,  .(LAU2, Size, hh_mean_age, hh_ckey)]

# result
hh_dat
```

```
##          LAU2 Size hh_mean_age    hh_ckey
##     1: 121025    4    31.00000 0.36787698
##     2: 312074    1    49.00000 0.94046728
##     3: 223030    1    37.00000 0.04555650
##     4: 314011    4    31.25000 0.42857428
##     5: 130015    3    78.00000 0.08773815
##    ---
##  9996: 130013    2    38.50000 0.37418980
##  9997: 130010   60    26.95000 0.32836301
##  9998: 332033   20    74.85000 0.21361981
##  9999: 323015   70    74.77143 0.95991730
## 10000: 334009  100    75.81000 0.48365275
```

Important: The household *cell-key* could also be interpreted as the *record-key* of the household. That is, if you are going to produce a household table and perturb it, the cell-key could be interpreted as record-key. Therefore:

```r
setnames(hh_dat, "hh_ckey", "hh_rkey")
hh_dat
```

```
##          LAU2 Size hh_mean_age    hh_rkey
##     1: 121025    4    31.00000 0.36787698
##     2: 312074    1    49.00000 0.94046728
##     3: 223030    1    37.00000 0.04555650
##     4: 314011    4    31.25000 0.42857428
##     5: 130015    3    78.00000 0.08773815
##    ---
##  9996: 130013    2    38.50000 0.37418980
##  9997: 130010   60    26.95000 0.32836301
##  9998: 332033   20    74.85000 0.21361981
##  9999: 323015   70    74.77143 0.95991730
## 10000: 334009  100    75.81000 0.48365275
```

(7b) How many households do we have? What will be the cell-key for this total number (don't use the cellKey-package; compute it manually) and what would be the noise if we use `ptab1` (manual lookup using the graph or look into the ptable)?

```r
ptab1 <- create_cnt_ptable(D = 2, V = 1.08, js = 1, mono = c(T,T,F,T))
```

Solution

```r
nrow(hh_dat)
```

```
## [1] 10000
```

```
sum(hh_dat$hh_rkey) %% 1
```

```
## [1] 0.4356278
```

```
ptab1@pTable
```

```
##     i j          p  v   p_int_lb   p_int_ub type
##  1: 0 0 1.00000000  0 0.00000000 1.00000000  all
##  2: 1 0 0.51333333 -1 0.00000000 0.51333333  all
##  3: 1 2 0.46000000  1 0.51333333 0.97333333  all
##  4: 1 3 0.02666667  2 0.97333333 1.00000000  all
##  5: 2 0 0.16560835 -2 0.00000000 0.16560835  all
##  6: 2 2 0.54634992  0 0.16560835 0.71195827  all
##  7: 2 3 0.24486677  1 0.71195827 0.95682504  all
##  8: 2 4 0.04317496  2 0.95682504 1.00000000  all
##  9: 3 2 0.42078468 -1 0.00000000 0.42078468  all
## 10: 3 3 0.27764596  0 0.42078468 0.69843064  all
## 11: 3 4 0.18235404  1 0.69843064 0.88078468  all
## 12: 3 5 0.11921532  2 0.88078468 1.00000000  all
## 13: 4 2 0.07394668 -2 0.00000000 0.07394668  all
## 14: 4 3 0.24421329 -1 0.07394668 0.31815997  all
## 15: 4 4 0.36368006  0 0.31815997 0.68184003  all
## 16: 4 5 0.24421329  1 0.68184003 0.92605332  all
## 17: 4 6 0.07394668  2 0.92605332 1.00000000  all
```

Answer: The total frequency count 10000 has the cell-key 0.4356278 and will be perturbed by 0.

Exercise (8) [Advanced]

Create a one-dimensional table with the hierarchical variable NUTS3 and apply a filter.

(8a) Create a table object 'tab8' with a filter (argument 'countvars = ...'). The filter shall only count females (variable 'sex == 2'). (i.e. call the filter). Use the help page '?cellkey_pkg' to define the argument 'countvars'.

```
tab8 <-
```

Solution

```
dat[, female := ifelse(SEX == 2, 1, 0)]

tab8 <- ck_setup(
  x = dat,
  rkey = "rkey",
  dims = list(NUTS3 = d_nuts3),
  countvars = "female"
)
```

(8b) Perturb the table using the new countvar and ptable 'ptab1'.

Solution

```
tab8$params_cnts_set(val = ck_params_cnts(ptab = ptab1), v = "female")
```

```
## --> setting perturbation parameters for variable "female"
```

```
tab8$perturb(v = "female")
```

```
## Count variable "female" was perturbed.
```

```
tab8$freqtab(v = c("female"))
```

```
##     NUTS3  vname   uwc    wc  puwc   pwc
##  1: Total female 15271 15271 15272 15272
##  2:     1 female  6815  6815  6816  6816
##  3:    11 female   501   501   502   502
##  4:   111 female    66    66    65    65
##  5:   112 female   297   297   297   297
##  6:   113 female   138   138   136   136
##  7:    12 female  2780  2780  2781  2781
##  8:   121 female   373   373   374   374
##  9:   122 female   430   430   430   430
## 10:   123 female   259   259   259   259
## 11:   124 female   284   284   284   284
## 12:   125 female   219   219   218   218
## 13:   126 female   610   610   610   610
## 14:   127 female   605   605   604   604
## 15:    13 female  3534  3534  3533  3533
## 16:   130 female  3534  3534  3533  3533
## 17:     2 female  2857  2857  2855  2855
## 18:    21 female   881   881   882   882
## 19:   211 female   485   485   485   485
## 20:   212 female   175   175   173   173
## 21:   213 female   221   221   219   219
## 22:    22 female  1976  1976  1975  1975
## 23:   221 female   896   896   897   897
## 24:   222 female   133   133   132   132
## 25:   223 female   268   268   267   267
## 26:   224 female   316   316   315   315
## 27:   225 female   236   236   236   236
## 28:   226 female   127   127   126   126
## 29:     3 female  5599  5599  5601  5601
## 30:    31 female  2450  2450  2450  2450
## 31:   311 female   380   380   380   380
## 32:   312 female  1131  1131  1131  1131
## 33:   313 female   296   296   296   296
## 34:   314 female   246   246   245   245
## 35:   315 female   397   397   397   397
## 36:    32 female  1054  1054  1055  1055
## 37:   321 female    30    30    31    31
```

```
## 38:   322 female   293   293   293   293
## 39:   323 female   731   731   731   731
## 40:    33 female  1398  1398  1399  1399
## 41:   331 female    56    56    57    57
## 42:   332 female   583   583   583   583
## 43:   333 female    63    63    63    63
## 44:   334 female   259   259   260   260
## 45:   335 female   437   437   437   437
## 46:    34 female   697   697   695   695
## 47:   341 female   171   171   171   171
## 48:   342 female   526   526   526   526
##      NUTS3  vname   uwc    wc  puwc   pwc
```

Exercise (9) [Advanced]

(9a) Compare the distributions of the two following ptables.

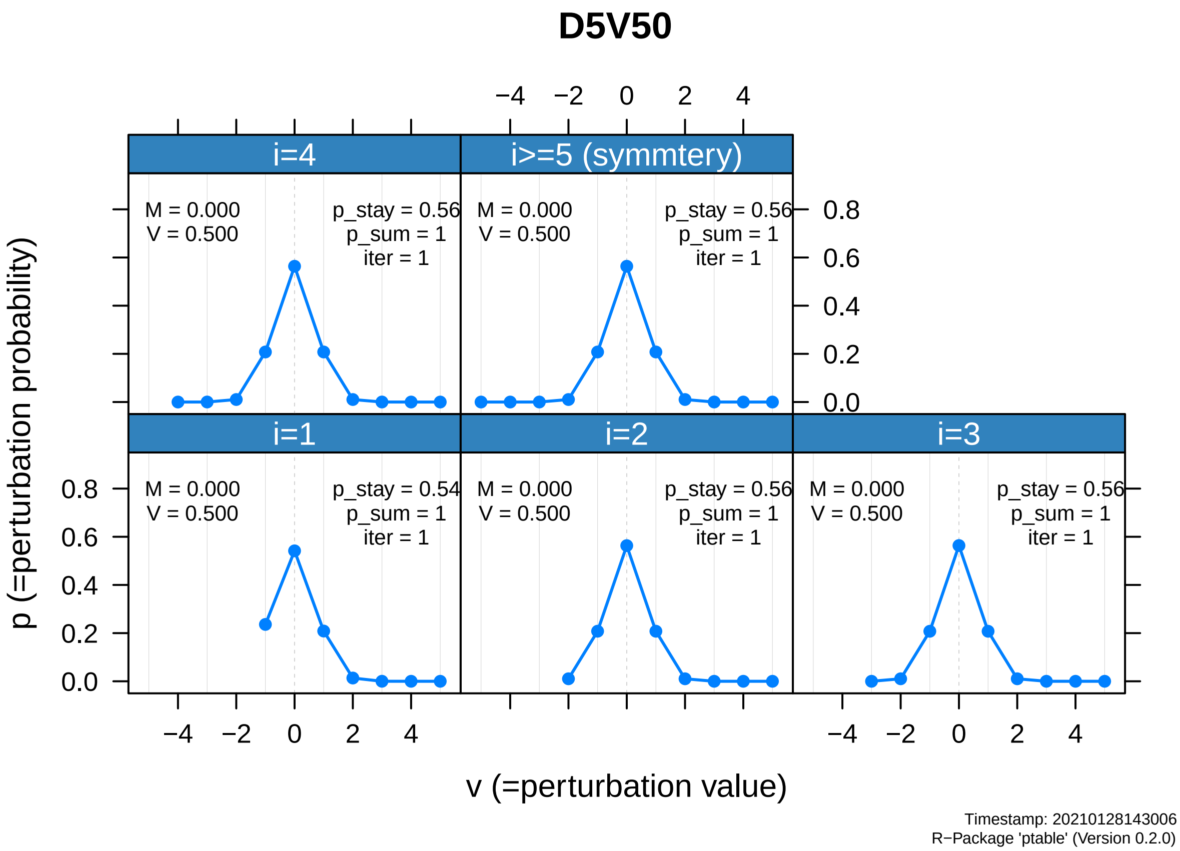
```
ptab91 <- create_cnt_ptable(D = 5, V = 0.5, optim=4)
ptab92 <- create_cnt_ptable(D = 5, V = 2, optim=4)
```

What is the main difference between the distributions (look at the graph)?

Solution

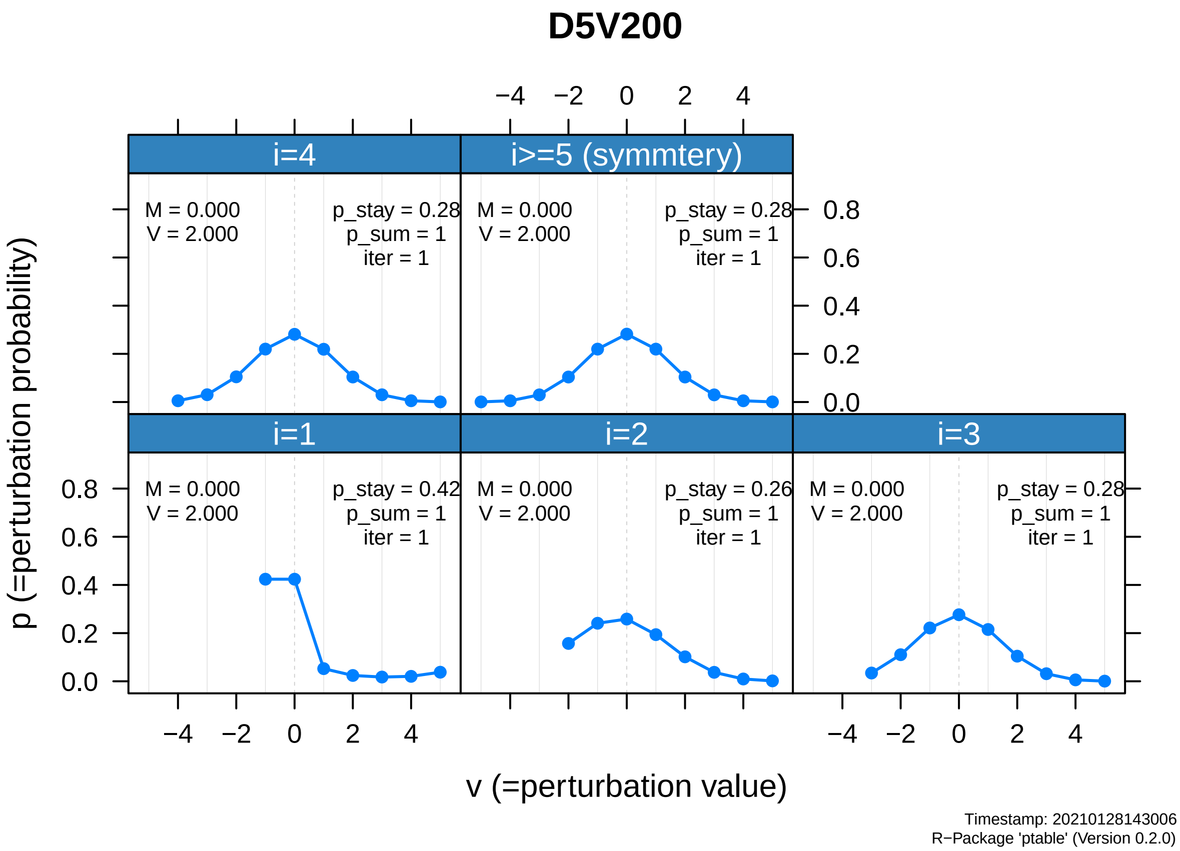
```
plot(ptab91, type="d")
```

```
## Distribution of Perturbation Values
```




```
plot(ptab92, type="d")
```

```
## Distribution of Perturbation Values
```



Answer: Leptocurtic curve (`ptab91`) with high probabilitiy for noise 0 versus normal curtosis (`ptab92`).

(9b) What would you expect: Which ptable has a lower loss of information? Perturb the table you have designed in (8) twice and perturb the tables with the two ptables.

```
tab91 <- ...
tab92 <- ...

...
```

Solution

```
tab91 <- ck_setup(
  x = dat,
  rkey = "rkey",
  dims = list(NUTS3 = d_nuts3)
)
tab92 <- ck_setup(
  x = dat,
  rkey = "rkey",
  dims = list(NUTS3 = d_nuts3)
)
tab91$params_cnts_set(val = ck_params_cnts(ptab = ptab91), v = "total")
```

```
## --> setting perturbation parameters for variable "total"
```

```r
tab92$params_cnts_set(val = ck_params_cnts(ptab = ptab92), v = "total")
```

```
## --> setting perturbation parameters for variable "total"
```

```r
tab91$perturb(v = "total")
```

```
## Count variable "total" was perturbed.
```

```r
tab92$perturb(v = "total")
```

```
## Count variable "total" was perturbed.
```

```r
tab91$measures_cnts(v = "total")$measures
```

```
##       what    d1    d2    d3
##  1:    Min 0.000 0.000 0.000
##  2:    Q10 0.000 0.000 0.000
##  3:    Q20 0.000 0.000 0.000
##  4:    Q30 0.000 0.000 0.000
##  5:    Q40 0.000 0.000 0.000
##  6:   Mean 0.396 0.002 0.010
##  7: Median 0.000 0.000 0.000
##  8:    Q60 0.000 0.000 0.000
##  9:    Q70 1.000 0.001 0.013
## 10:    Q80 1.000 0.002 0.020
## 11:    Q90 1.000 0.003 0.026
## 12:    Q95 1.000 0.006 0.039
## 13:    Q99 1.530 0.024 0.093
## 14:    Max 2.000 0.037 0.135
```

```r
tab92$measures_cnts(v = "total")$measures
```

```
##       what    d1    d2    d3
##  1:    Min 0.000 0.000 0.000
##  2:    Q10 0.000 0.000 0.000
##  3:    Q20 0.000 0.000 0.000
##  4:    Q30 0.000 0.000 0.000
##  5:    Q40 0.000 0.000 0.000
##  6:   Mean 0.875 0.003 0.021
##  7: Median 1.000 0.000 0.011
##  8:    Q60 1.000 0.001 0.019
##  9:    Q70 1.000 0.002 0.024
## 10:    Q80 2.000 0.002 0.031
## 11:    Q90 2.000 0.005 0.045
## 12:    Q95 2.000 0.012 0.077
## 13:    Q99 3.000 0.037 0.150
## 14:    Max 3.000 0.056 0.201
```

Answer: `ptab91` has a lower variance and, hence, a lower loss of information.