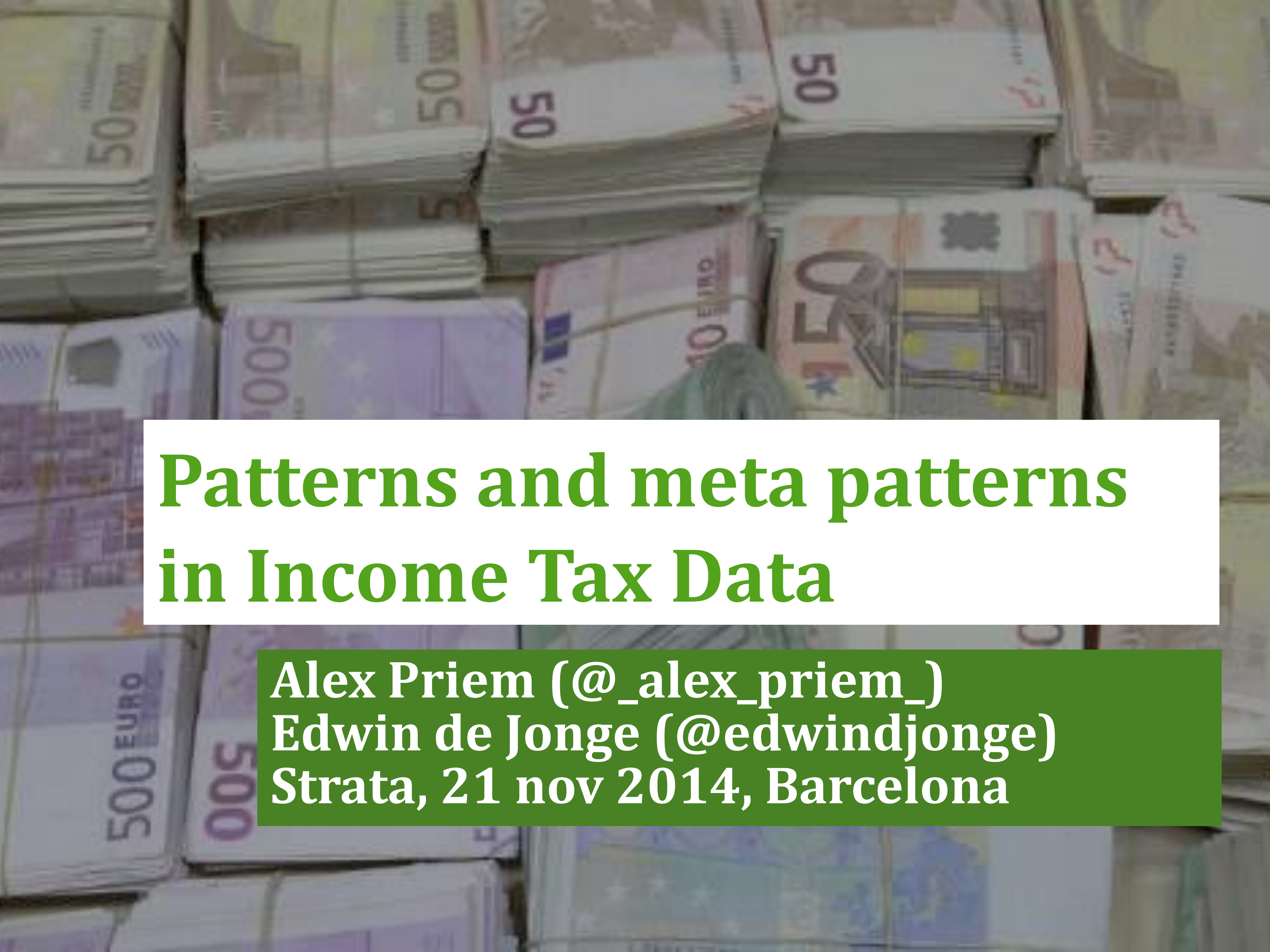
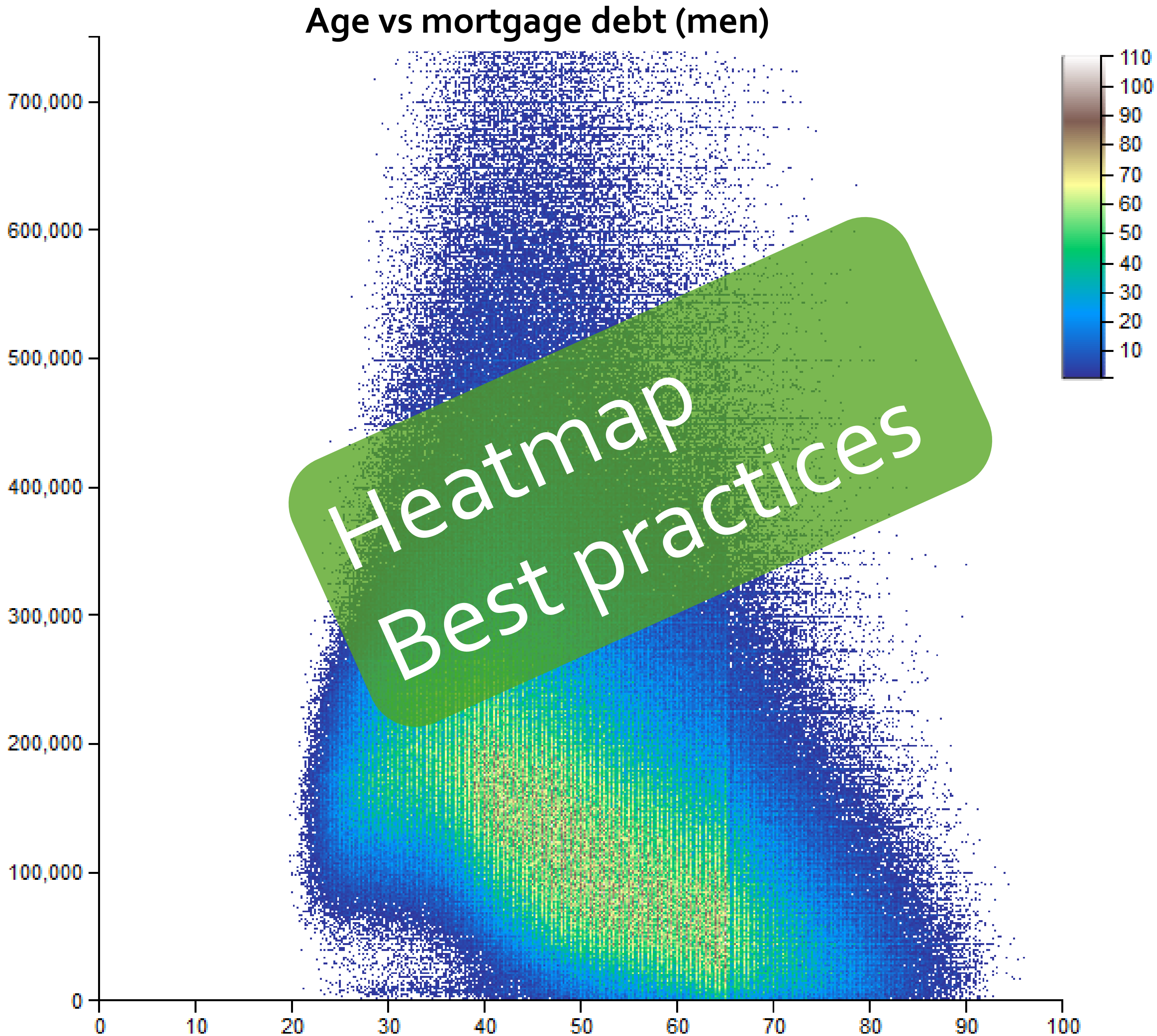


Patterns and meta patterns in Income Tax Data

Alex Priem (@_alex_priem_)
Edwin de Jonge (@edwindjonge)
Strata, 21 nov 2014, Barcelona

Heatmap
Best practices



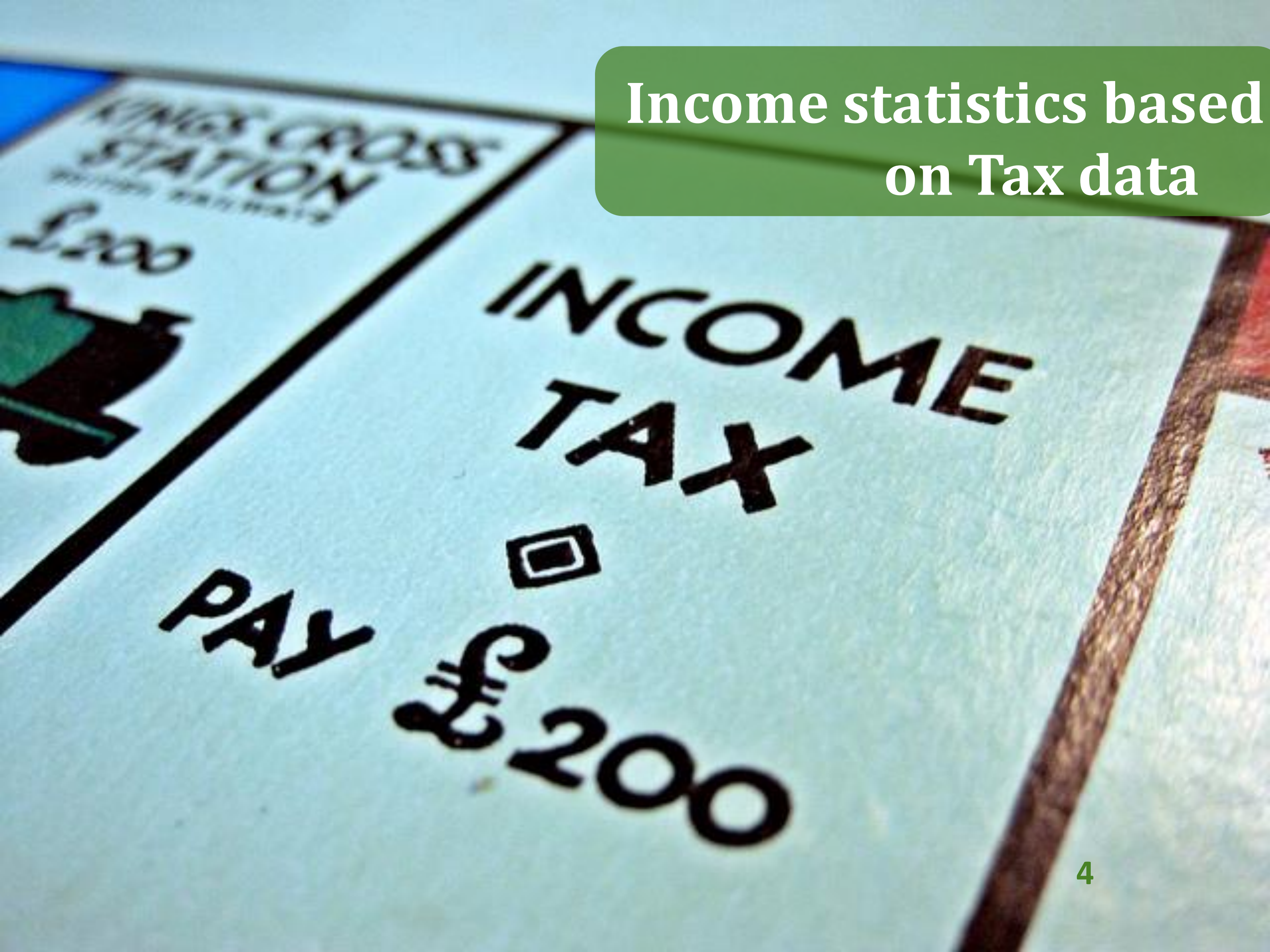
Who are we?

Statistical consultants / Data scientists working @ R&D department of Statistics Netherlands

Statistics Netherlands (SN):

- Government agency
- Produces all official statistics of The Netherlands

Income statistics based on Tax data



Income Tax data

– Contains all income tax records for the Netherlands
– Approx 17M records with 550 variables.
– Used to produce income statistics!

Analysis is not trivial

– Income Tax is complex (at least in the Netherlands)
  - stages of progressive tax
  - Complex Tax deductions (mortgage, flex workers)
  - Complex Tax benefits (child care, social benefits)

Tax data (2)

- 550 variables (for each person in NL):
  - 15 identificators/unique keys
    - Dwelling, person id, etc.
  - 70 categorical
  - 250 numerical variables from the income tax form
  - >200 derived variables (useful for analysis)
    - E.g. expandable income, income of dwelling/household

Income/tax distributions

Income (re)distribution hot topic since Piketty

So how are income/tax/benefits distributed?

- Look at 1D distributions: histograms
- Look at 2D distributions: heatmaps
  - Problem: potentially 0.5 n(n-1) > 100k heatmaps!

  - even more when categorical included

Let look at Patterns.

Heatmap Patterns

– What defines a pattern in heatmap?
  - Peak/Spike? (mode, 0D point)
  - Stripe (1D):
    - Horizontal Line?
    - Vertical Line?
    - Band?
    - Ridge?
  - Blob (2D)
  - Similarity between distributions (2D)

Meta pattern?

Meta patterns constitutes of repeating pattern in:

- different subpopulations
  - E.g. Male/female, Social economic status, Works in branch of Industry
- different pairs of variables
  - Income x age
  - Benefits x age
  - Etc.

So patterns that are generic over different heatmaps.

Looking for patterns

Subpopulations:
– Generate heatmap per category e.g. Age x Gross Income per social economic status
– Automatic cluster heatmaps on distribution simularity

Pairs of variables:
- Generate heatmaps for all pairs
- Prune: remove heatmaps with low support
  1. Use image classification to cluster them
  2. Or Cluster on extracted mode/line (wip)

You will still need to look at the result!

Why Visualization?

Anscombes quartet…

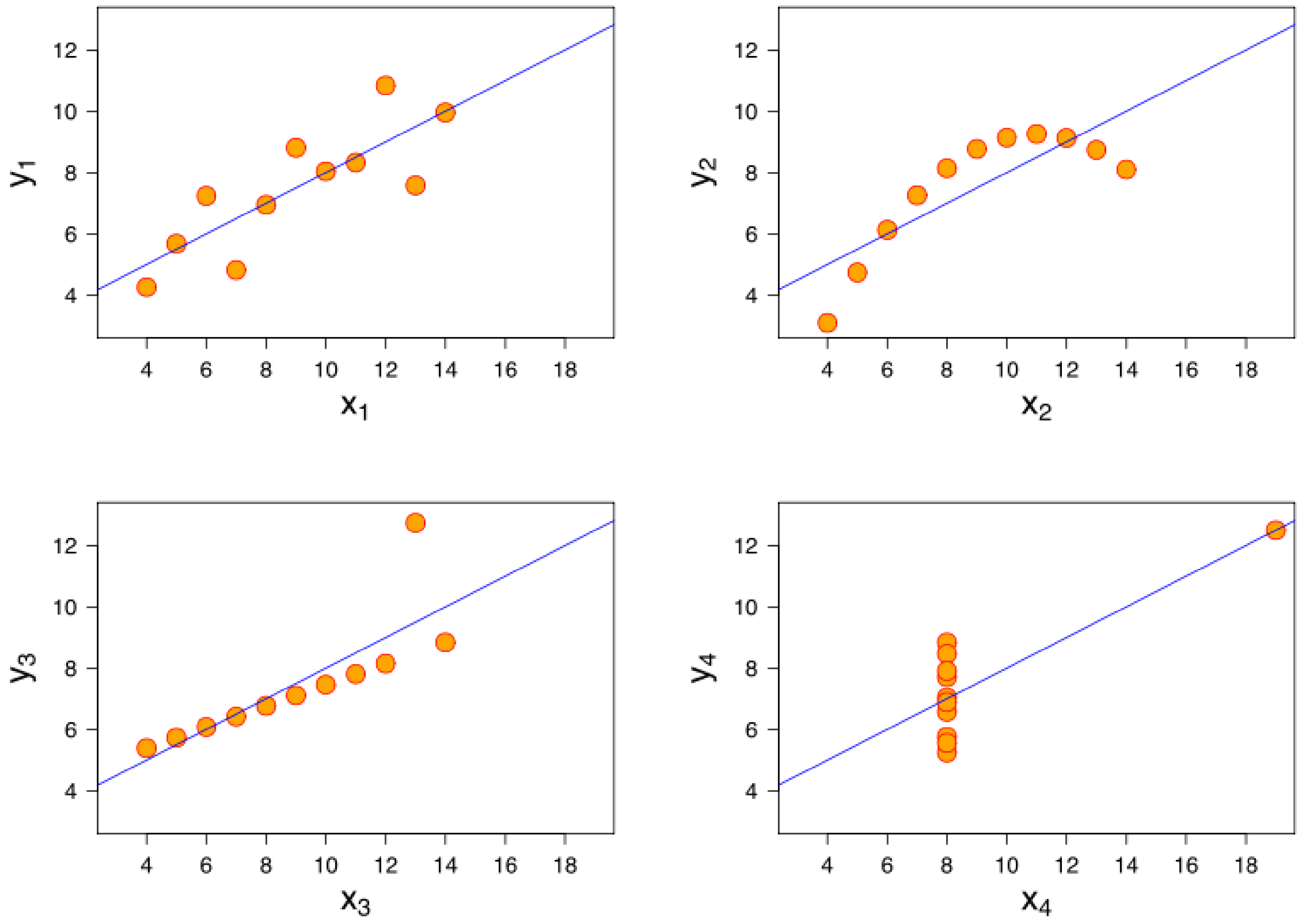
DS1 x	*y*	*DS2 x*	*y*	*DS3 x*	*y*	*DS4 x*	*y*
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.1	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.1	4	5.39	19	12.5
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89

Anscombe's quartet

Property	Value
Mean of x1, x2, x3, x4	All equal: 9
Variance of x1, x2, x3, x4	All equal: 11
Mean of y1, y2, y3, y4	All equal: 7.50
Variance of y1, y2, y3, y4	All equal: 4.1
Correlation for ds1, ds2, ds3, ds4	All equal 0.816
Linear regression for ds1, ds2, ds3, ds4	*All equal:* $y = 3.00 + 0.500x$

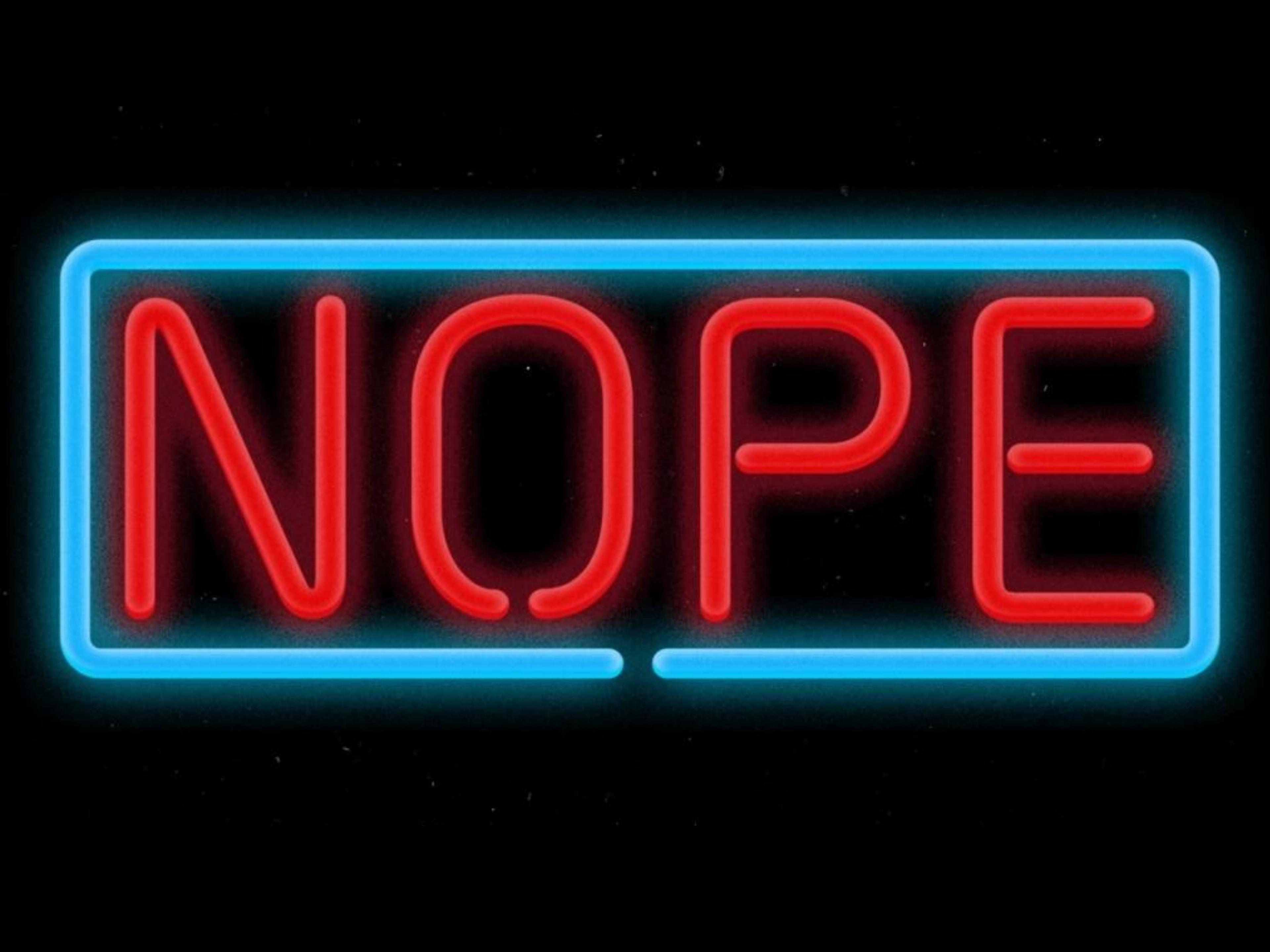
Looks the same, right?

Lets plot!



12
10
8
6
4
y_1
4 6 8 10 12 14 16 18
x_1
12
10
8
6
4
y_2
4 6 8 10 12 14 16 18
x_2
12
10
8
6
4
y_3
4 6 8 10 12 14 16 18
x_3
12
10
8
6
4
y_4
4 6 8 10 12 14 16 18
x_4

So clustering (machine learning) different?


NOPE

Visualization helps to …

– Test your (hidden model) assumptions!

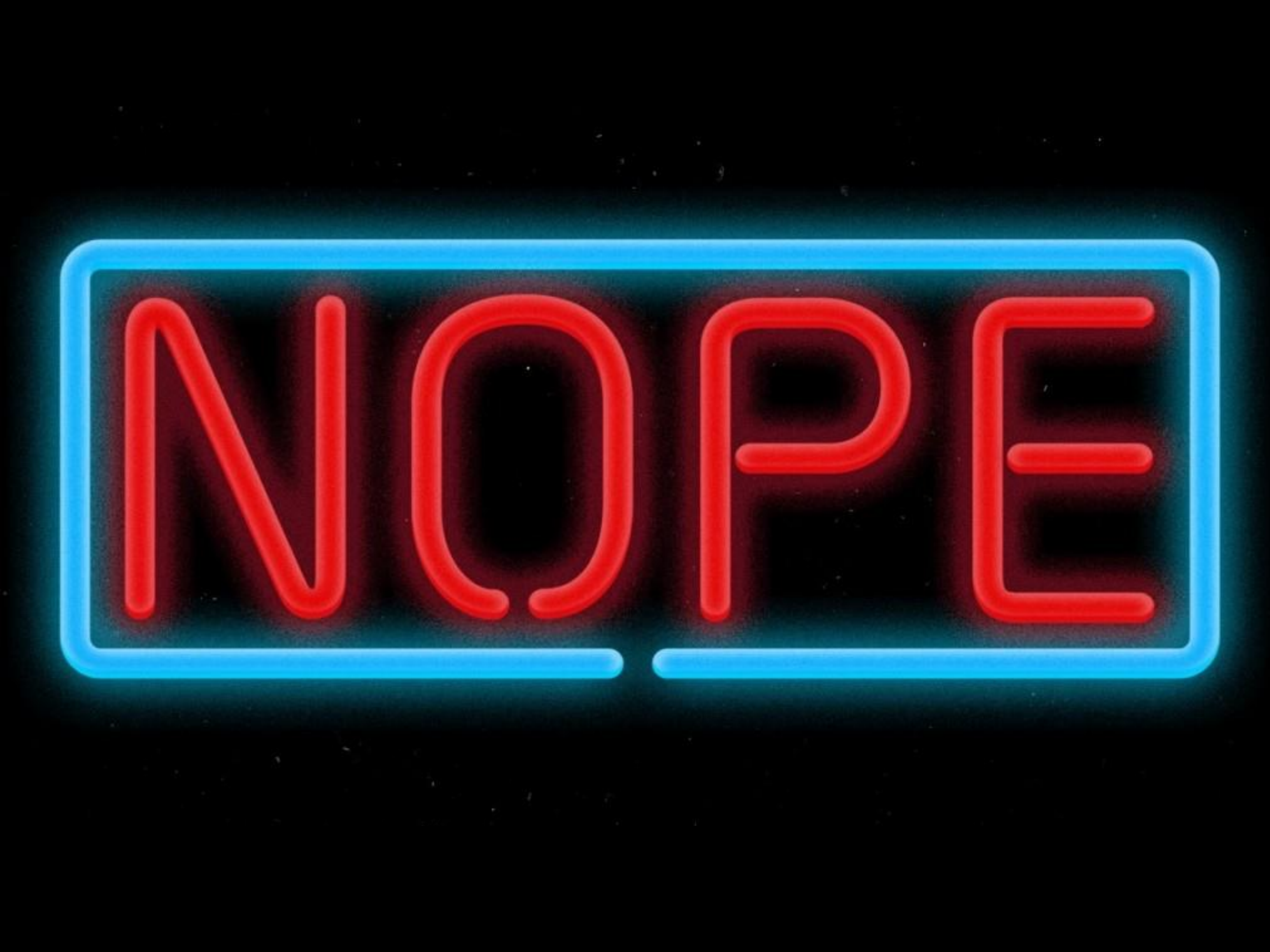
– To find structure in data, e.g.
"How is my data distributed?"

– Visually explore patterns:
  - Are there clusters?
  - Are there outliers?

Heatmap recipe

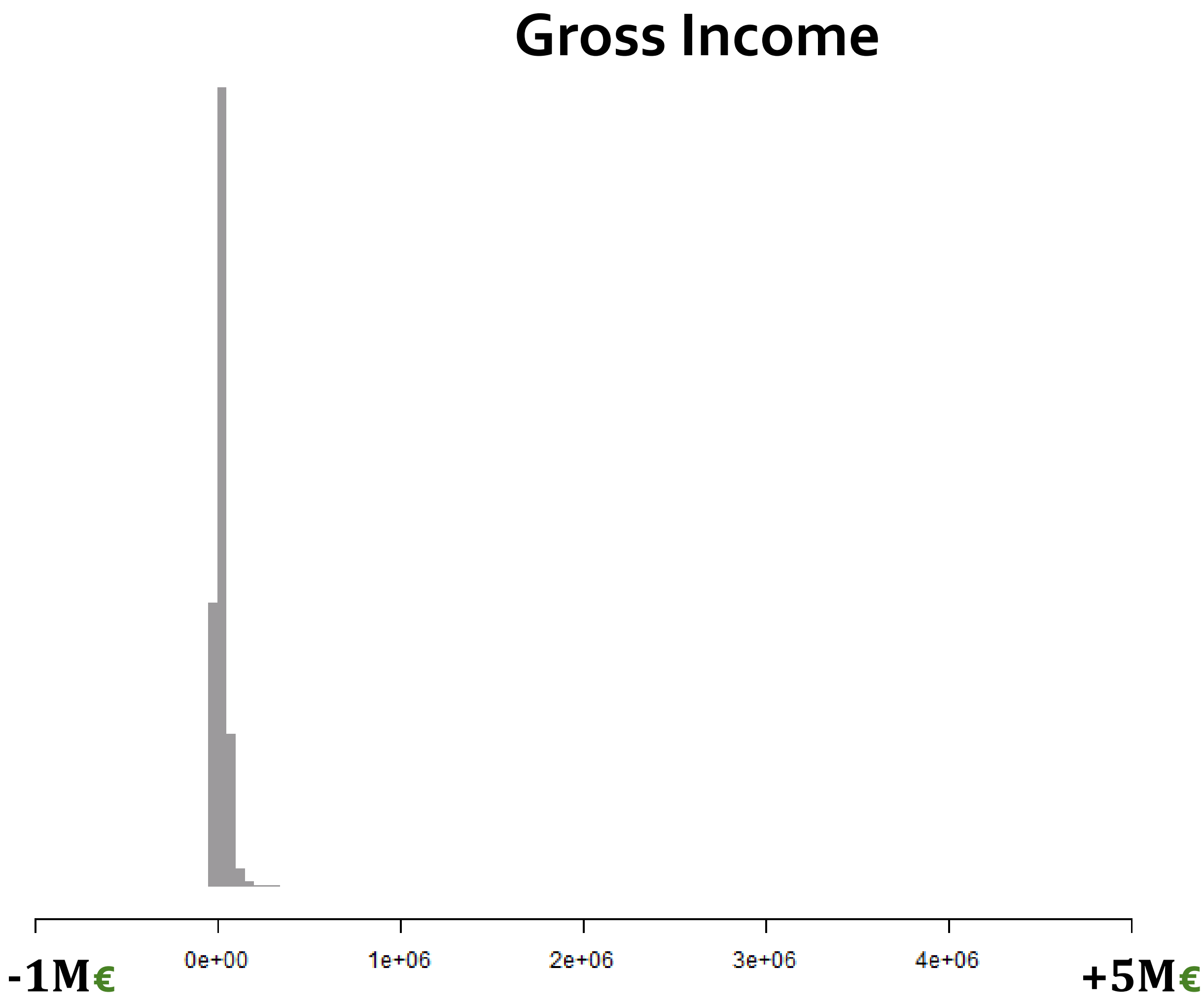
1. Take two numerical variables *x* and *y*
2. Determine range $r_x = [\min(x), \max(x)]$
3. Chop r_x in n_x equal pieces
4. Repeat for *y*
5. We now have $n_x \, . \, n_y$ bins
6. Count # records in each bin
7. Assign colors to counts
8. Plot matrix
9. Enjoy!

Easy as pie?

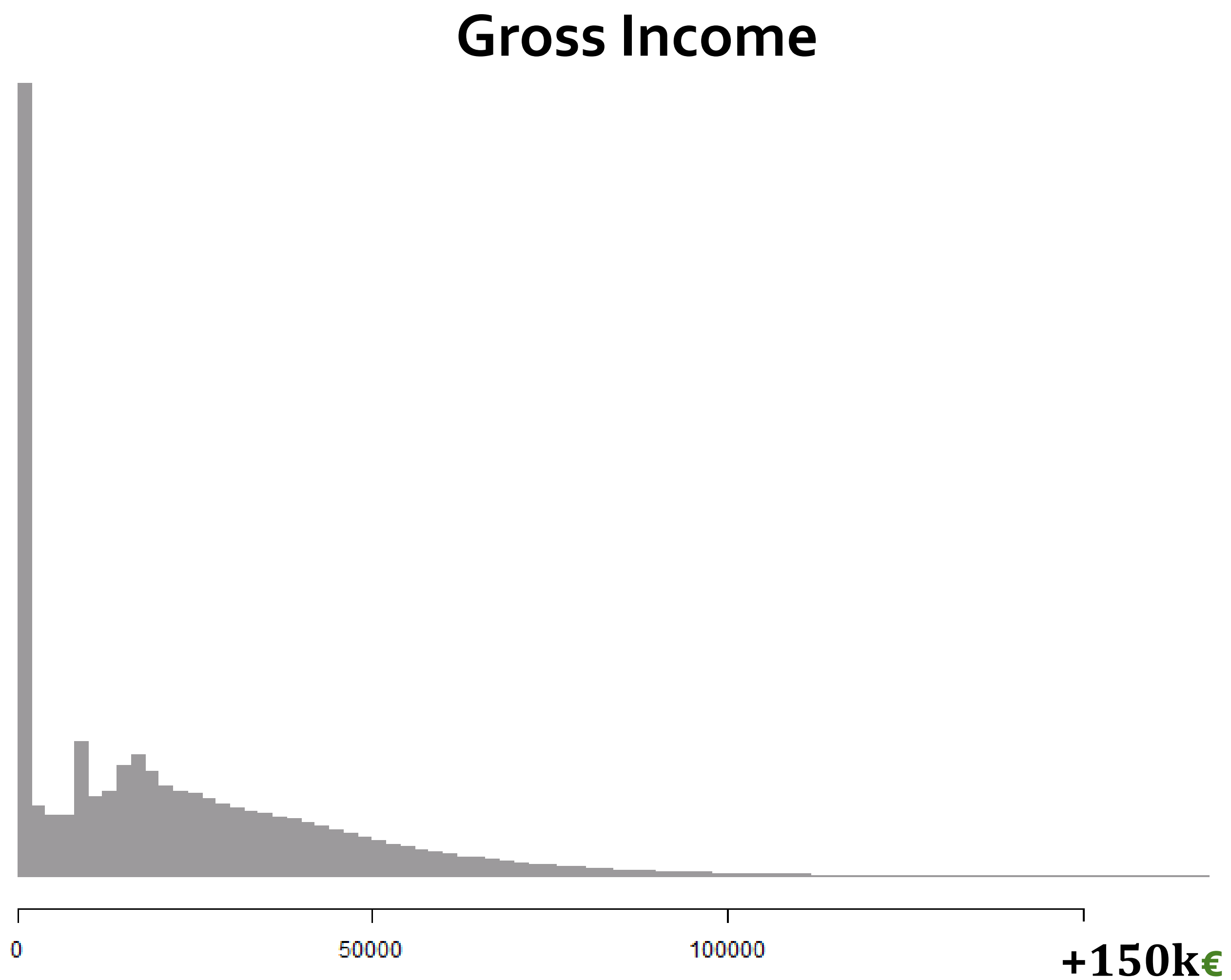

NOPE

1. Take two numerical variables *x* and *y*
2. **Determine range $\mathbf{r_x = [min(x)\,,max(x)]}$**
3. Chop r_x in n_x equal pieces
4. Repeat for *y*
5. We now have $n_x\,.\,n_y$ bins
6. Count # records in each bin
7. Assign colors to counts
8. Plot matrix
9. Enjoy!

Range: Outliers? (1D)



Range: outliers removed (1% removed)



Range: outliers…

Does your data contain outliers?

- If so: most pixels are empty
- Most cases: outliers have low mass and are barely visible

Truncate range: in x or y direction: e.g. 99% quantile

- Interactively: allow for *zoom* and *pan.*

Range: data skewed?

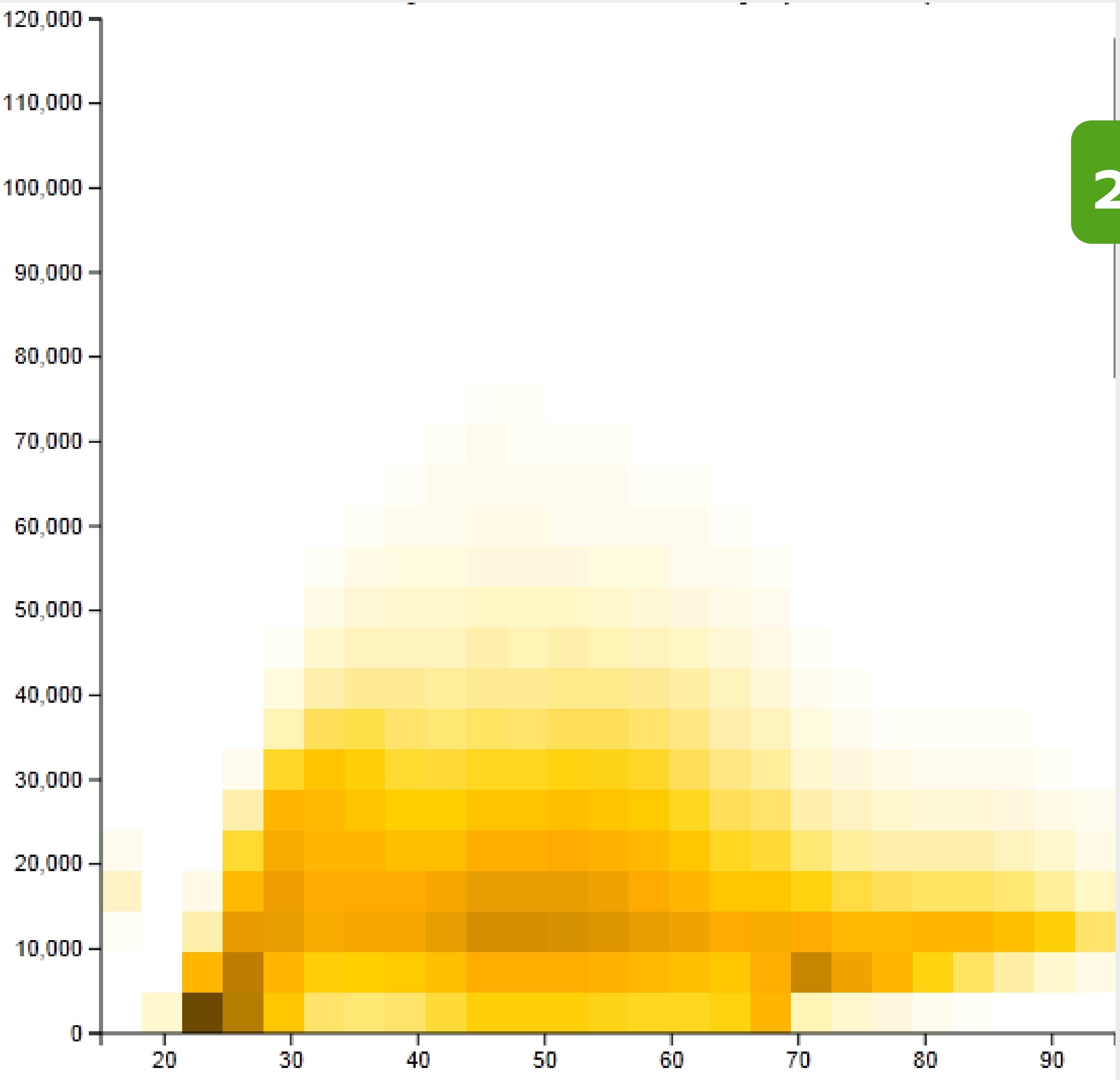
– Many variables are not normal distributed:
  - Power law: $\boldsymbol{x}^{\alpha}$
  - Exponential: e^{ax+b}

So rescale x or y or both

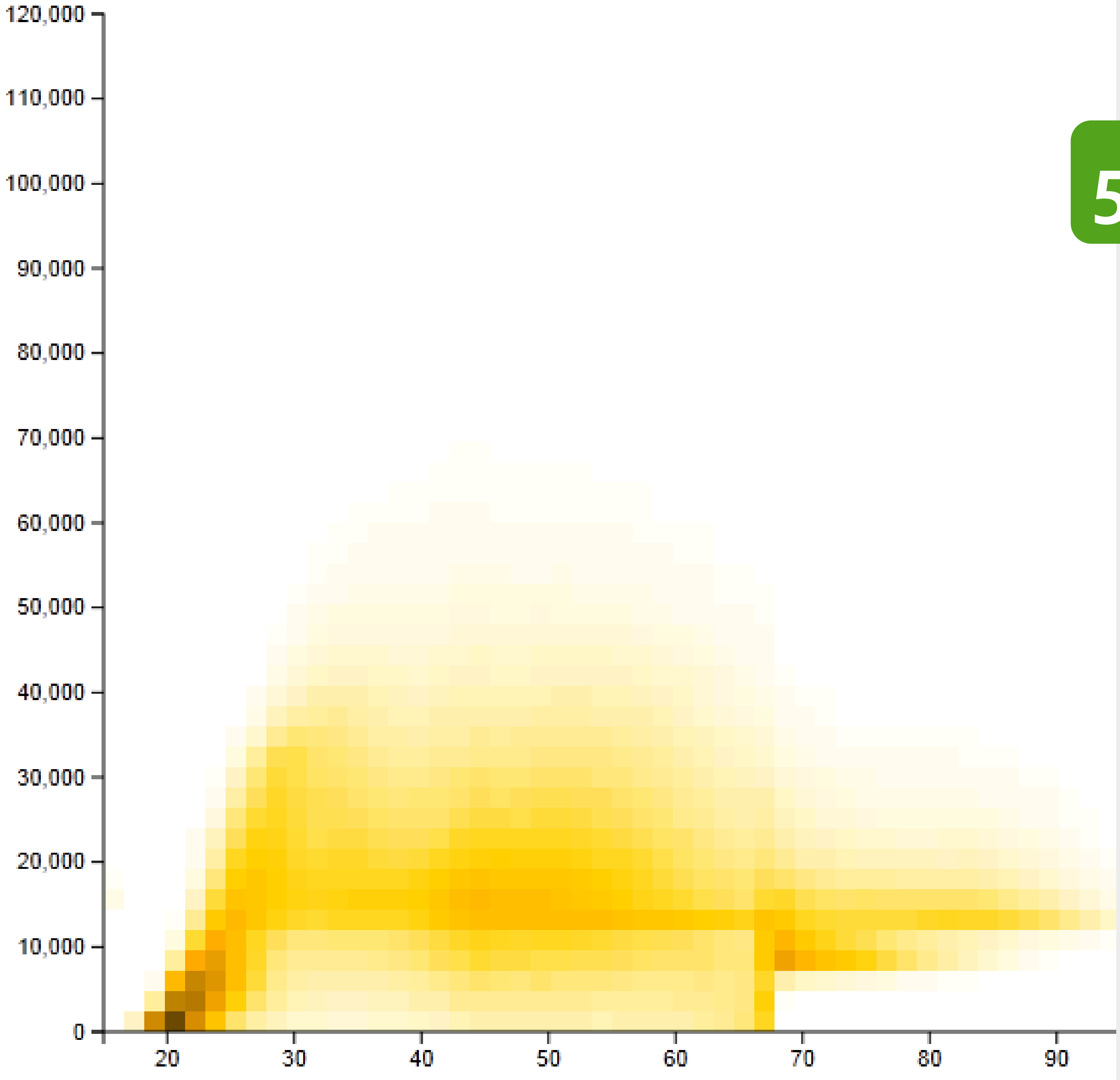
1. Take two numerical variables *x* and *y*
2. Determine range $\mathrm{r_x} = [\min(\mathrm{x}), \max(\mathrm{x})]$
3. **Chop $\boldsymbol{r_x}$ in $\boldsymbol{n_x}$ equal pieces**
4. Repeat for *y*
5. We now have $n_x \,.\, n_y$ bins
6. Count # records in each bin
7. Assign colors to counts
8. Plot matrix
9. Enjoy!

Chop: AKA “Binning”

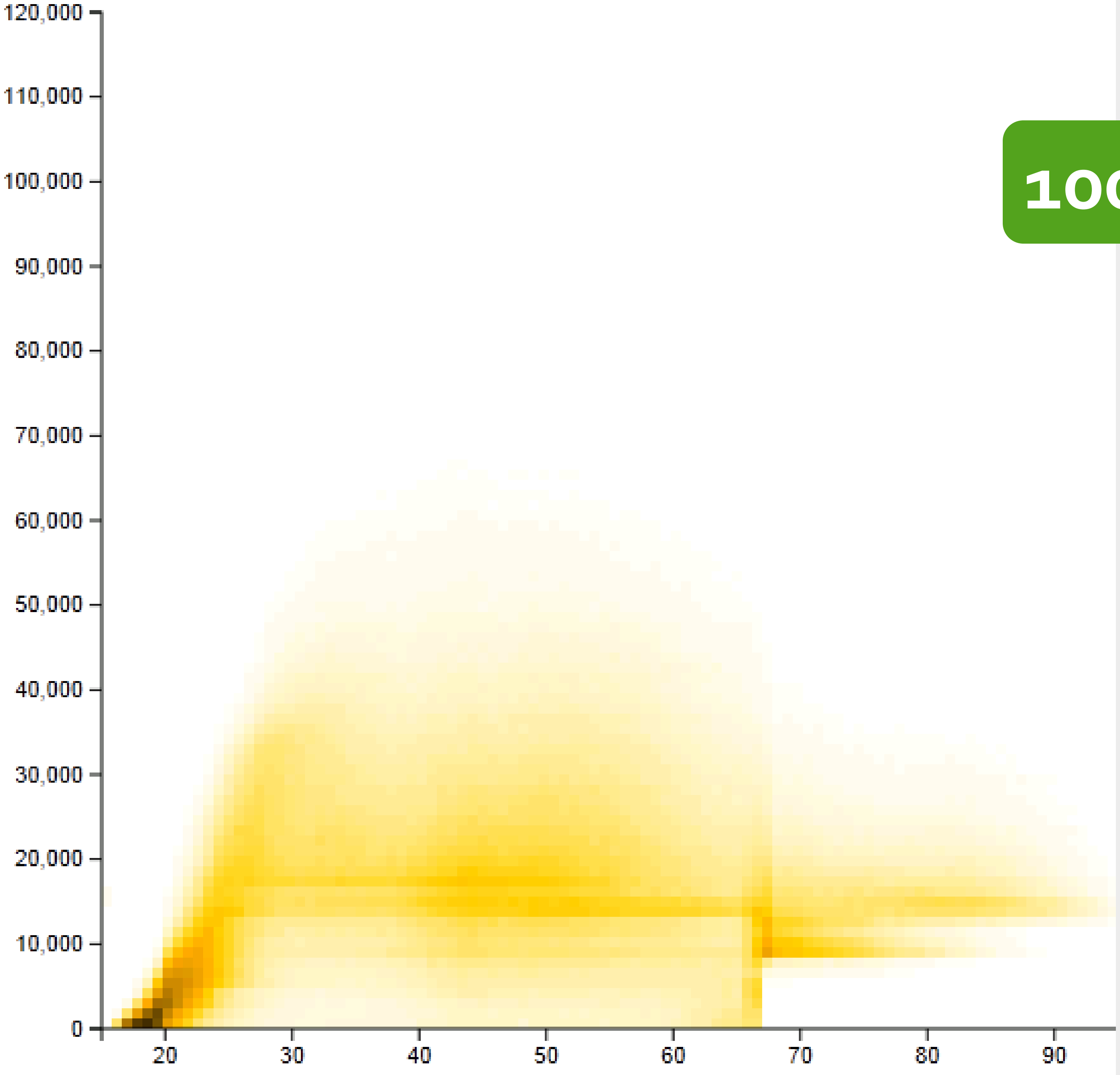
Resolution matters


120,000
110,000
100,000
90,000
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
20
30
40
50
60
70
80
90

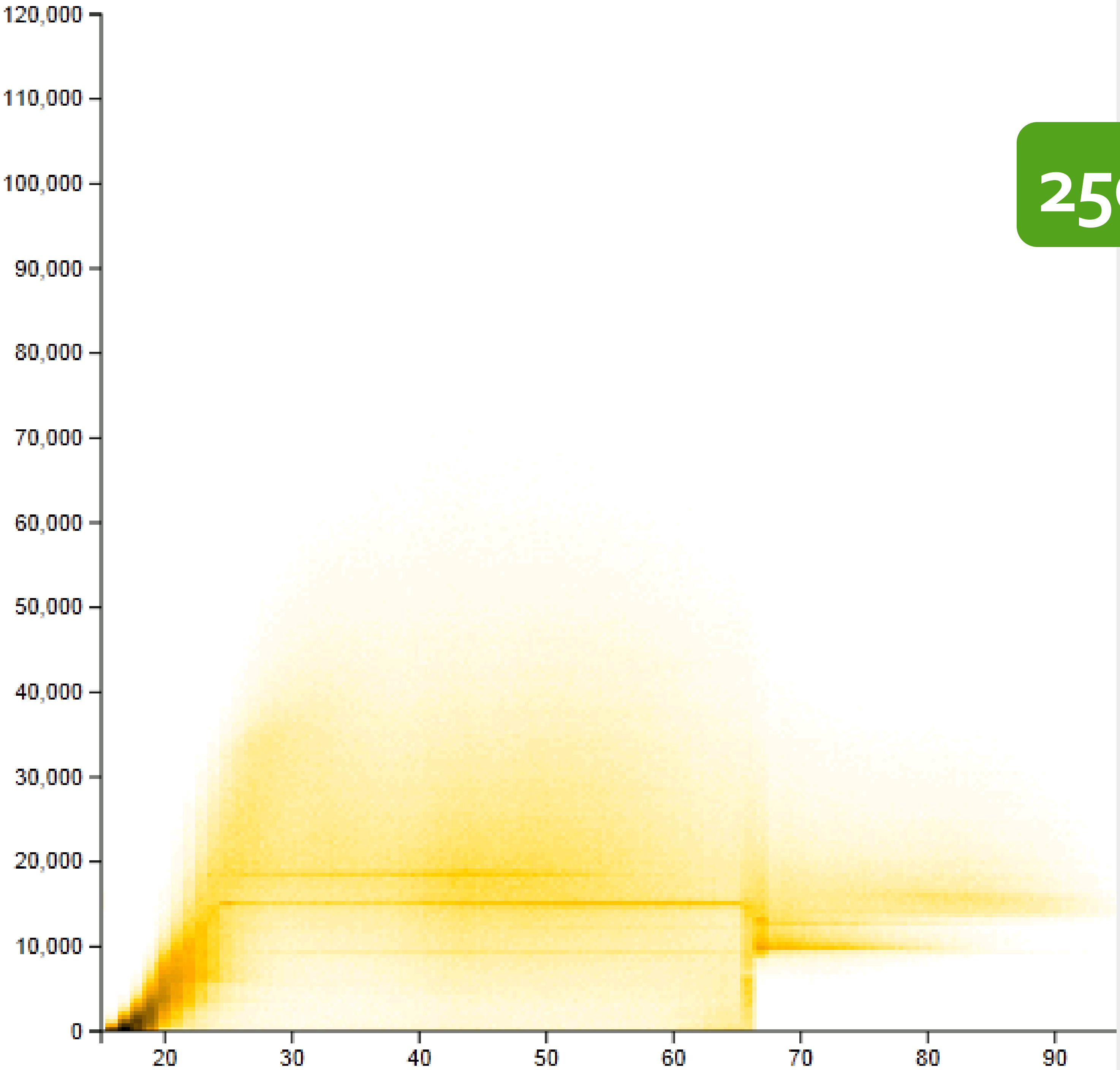
25 X 25


120,000
110,000
100,000
90,000
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
20
30
40
50
60
70
80
90

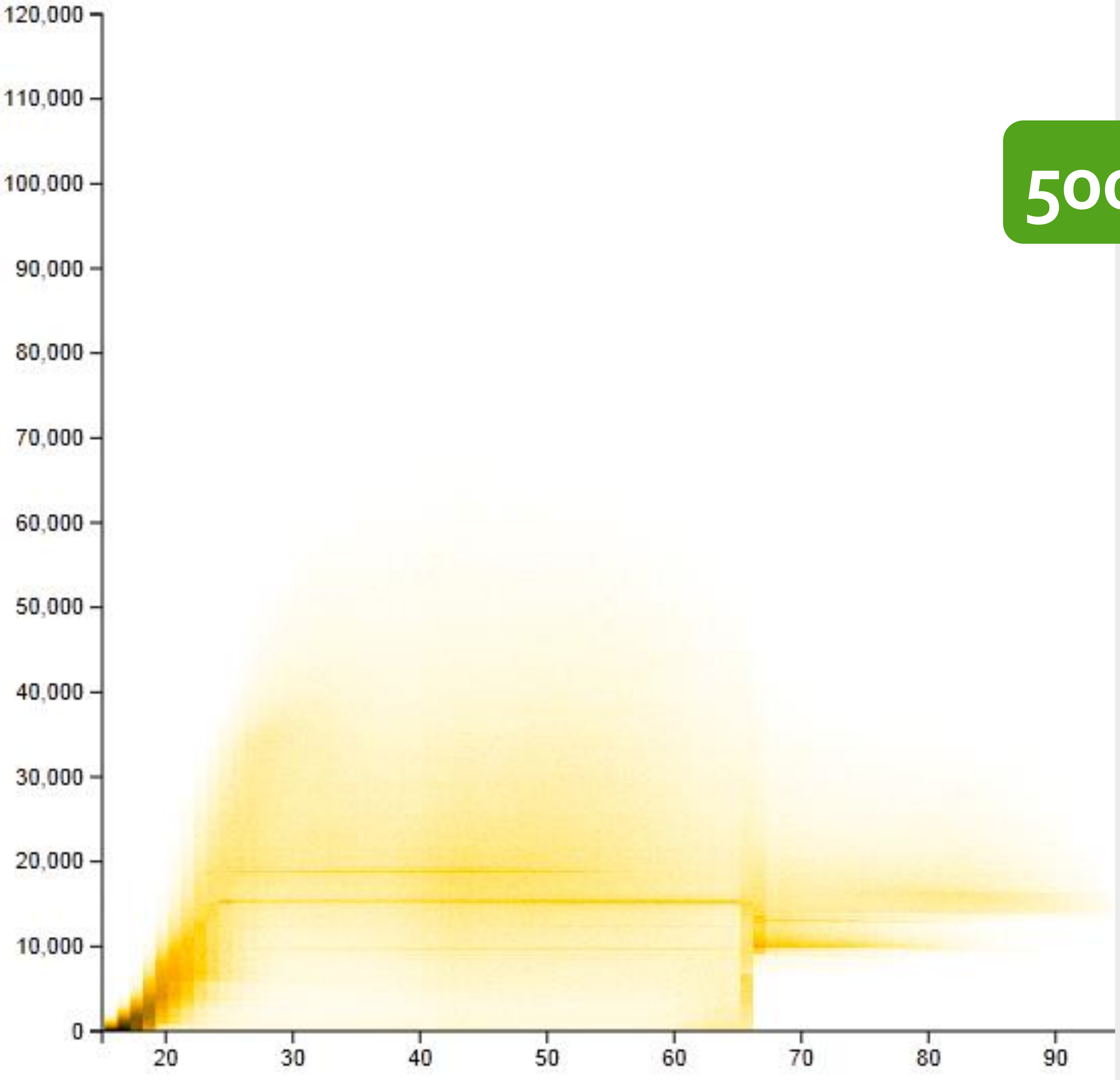
50 X 50


120,000
110,000
100,000
90,000
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
20
30
40
50
60
70
80
90

100 X 100


120,000
110,000
100,000
90,000
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
20
30
40
50
60
70
80
90

250 X 250


120,000
110,000
100,000
90,000
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
20
30
40
50
60
70
80
90

500 X 500

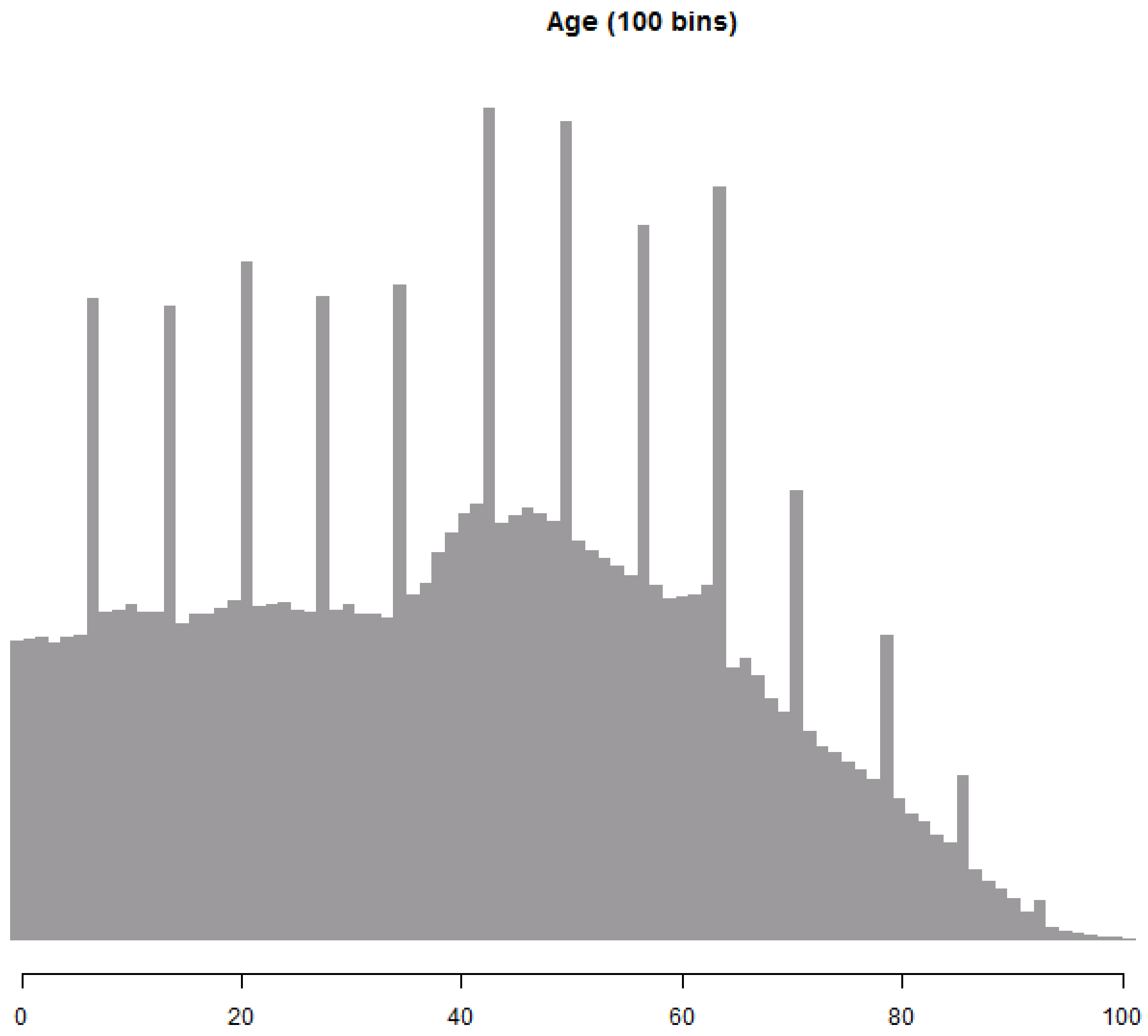
Chop: Too small / Too big

If #bins too small:
- patterns are hidden

If #bins too large:
- heatmap is noisy (signal vs noise)

Optimal nr bins depends on data.
(kernel based approx), but always play with bin size / resolution!

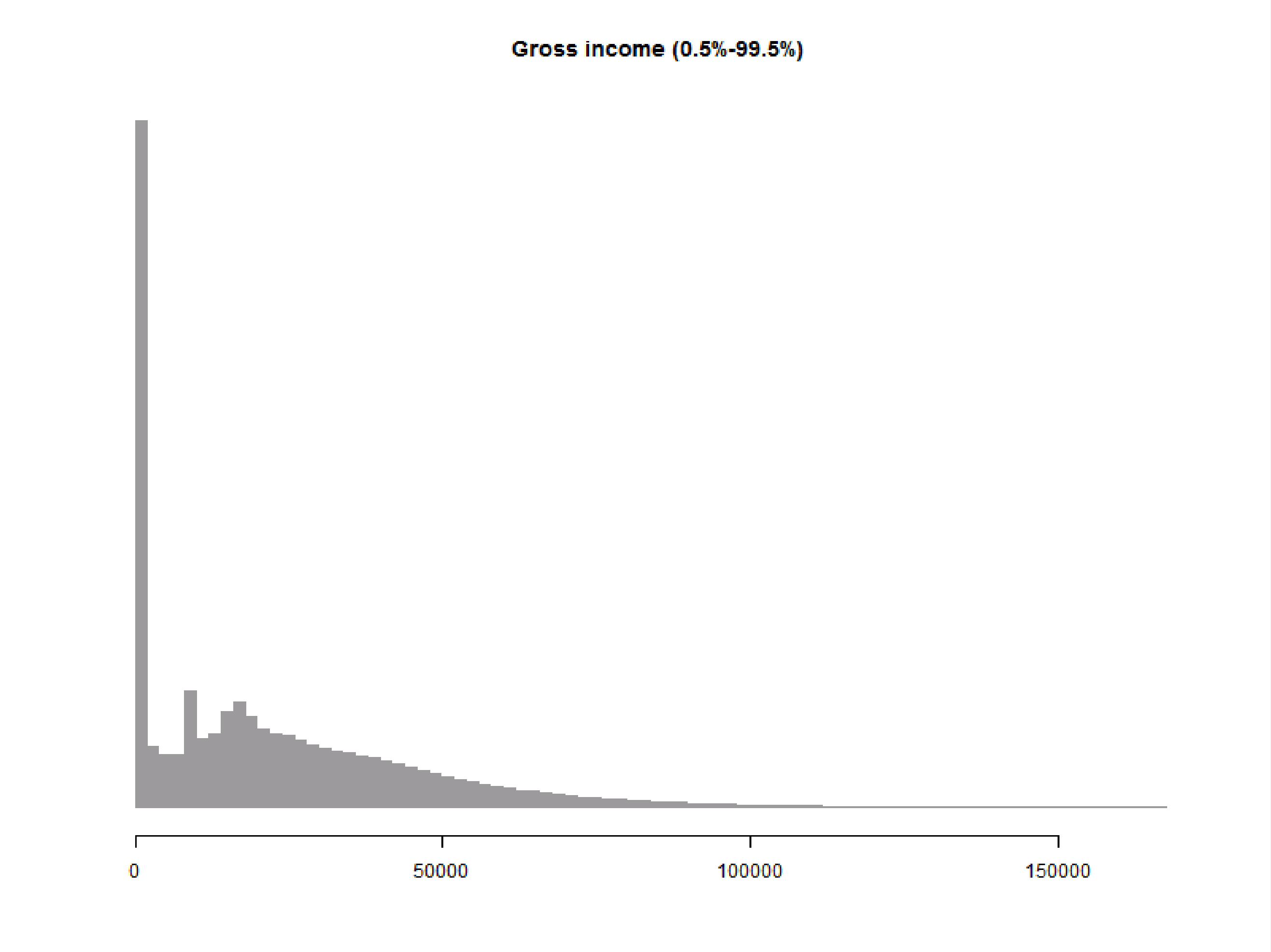
Chop: integers…



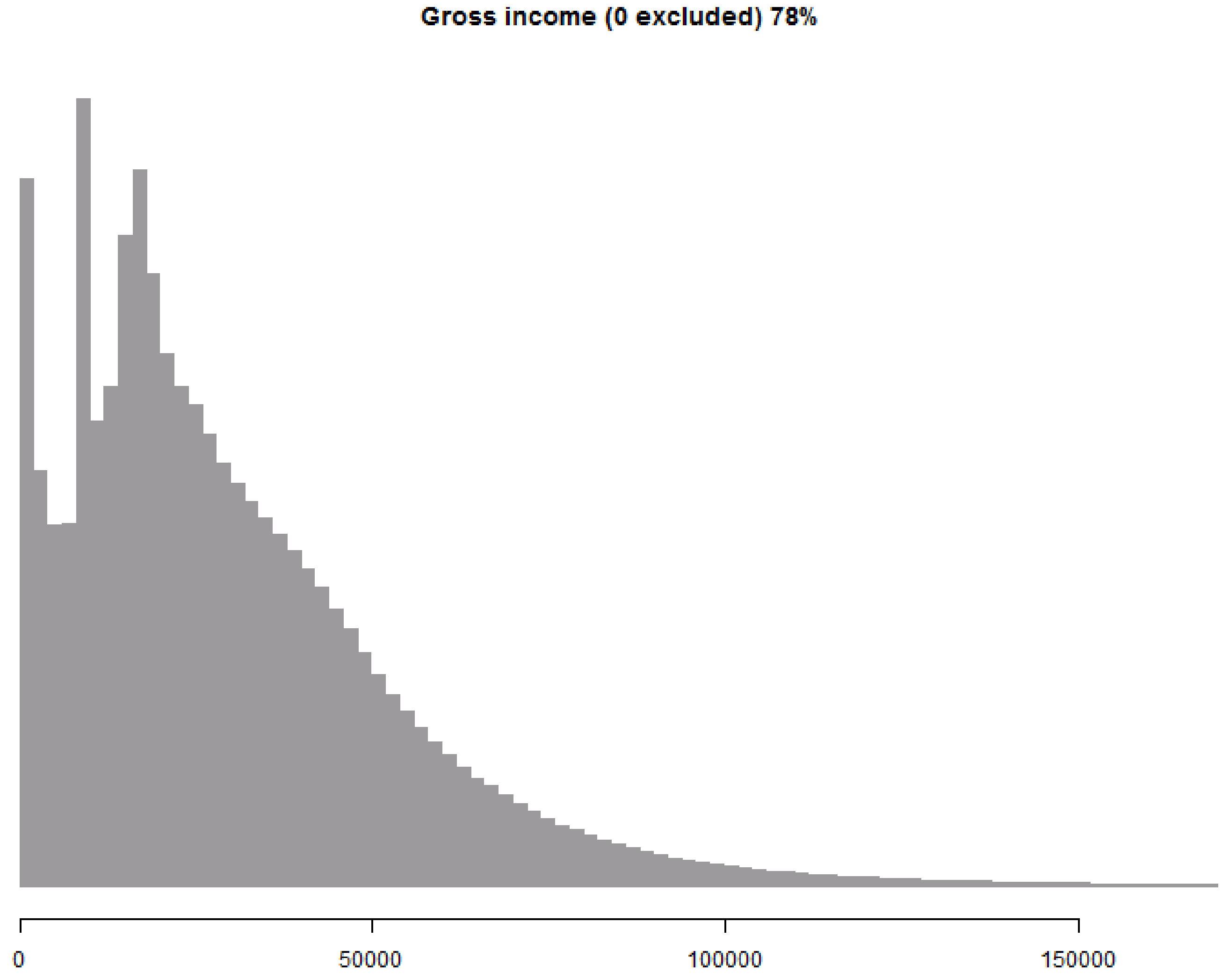
1. Take two numerical variables *x* and *y*
2. Determine range $\mathrm{r_x} = [\min(\mathrm{x}), \max(\mathrm{x})]$
3. Chop r_x in n_x equal pieces
4. Repeat for *y*
5. We now have $n_x \,.\, n_y$ bins
6. **Count # records in each bin**
7. Assign colors to counts
8. Plot matrix
9. Enjoy!

Count: zero counts

Not every variable is relevant for each person!

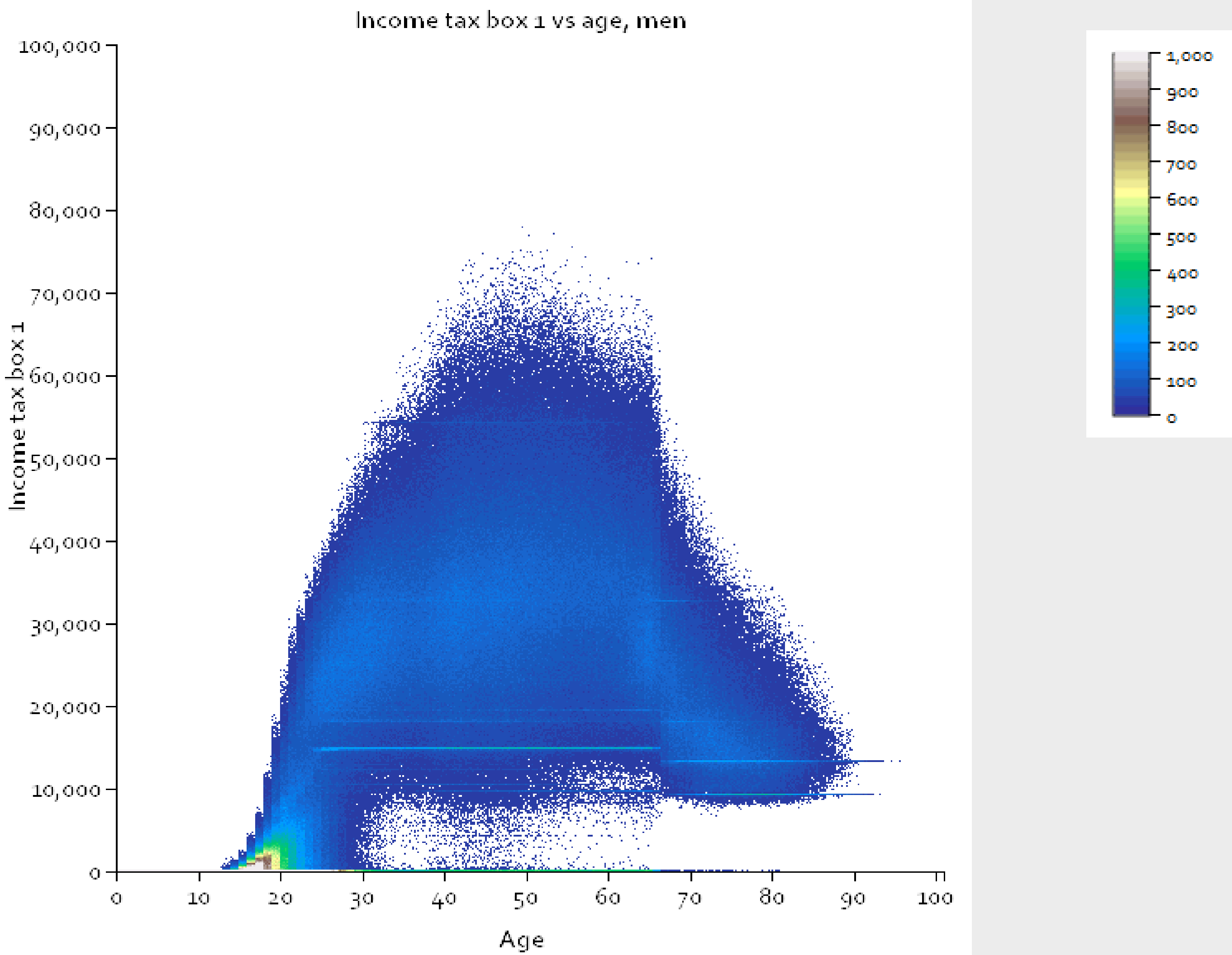


Count: exclude zero values



Assign colors!

1. Take two numerical variables *x* and *y*
2. Determine range $\mathrm{r_x} = [\min(\mathrm{x}), \max(\mathrm{x})]$
3. Chop r_x in n_x equal pieces
4. Repeat for *y*
5. We now have $n_x \,.\, n_y$ bins
6. Count # records in each bin
7. **Assign colors to counts**
8. Plot matrix
9. Enjoy!


Income tax box 1 vs age, men
Income tax box 1
Age
100,000
90,000
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
0
10
20
30
40
50
60
70
80
90
100
1,000
900
800
700
600
500
400
300
200
100
0

Colors: scales

– Color ‘intensity’ implies value
– Percieved response depends on ‘color’ and ‘color lightness’ (compare #00ff00 with #0000ff)
– Different models for color response:
  - RGB (models computer monitor)
  - HSV
  - HCL
  - CIELAB (models human eye)
– Gradient generator: http://davidjohnstone.net/pages/lch-lab-colour-gradient-picker

Colors

– Color has two functions in heatmap:
  - Show ‘counts’ in your data
  - Show ‘patterns’

At least, use a perceptually uniform gradient

- Libs: chroma.js, colorbrewer (R)

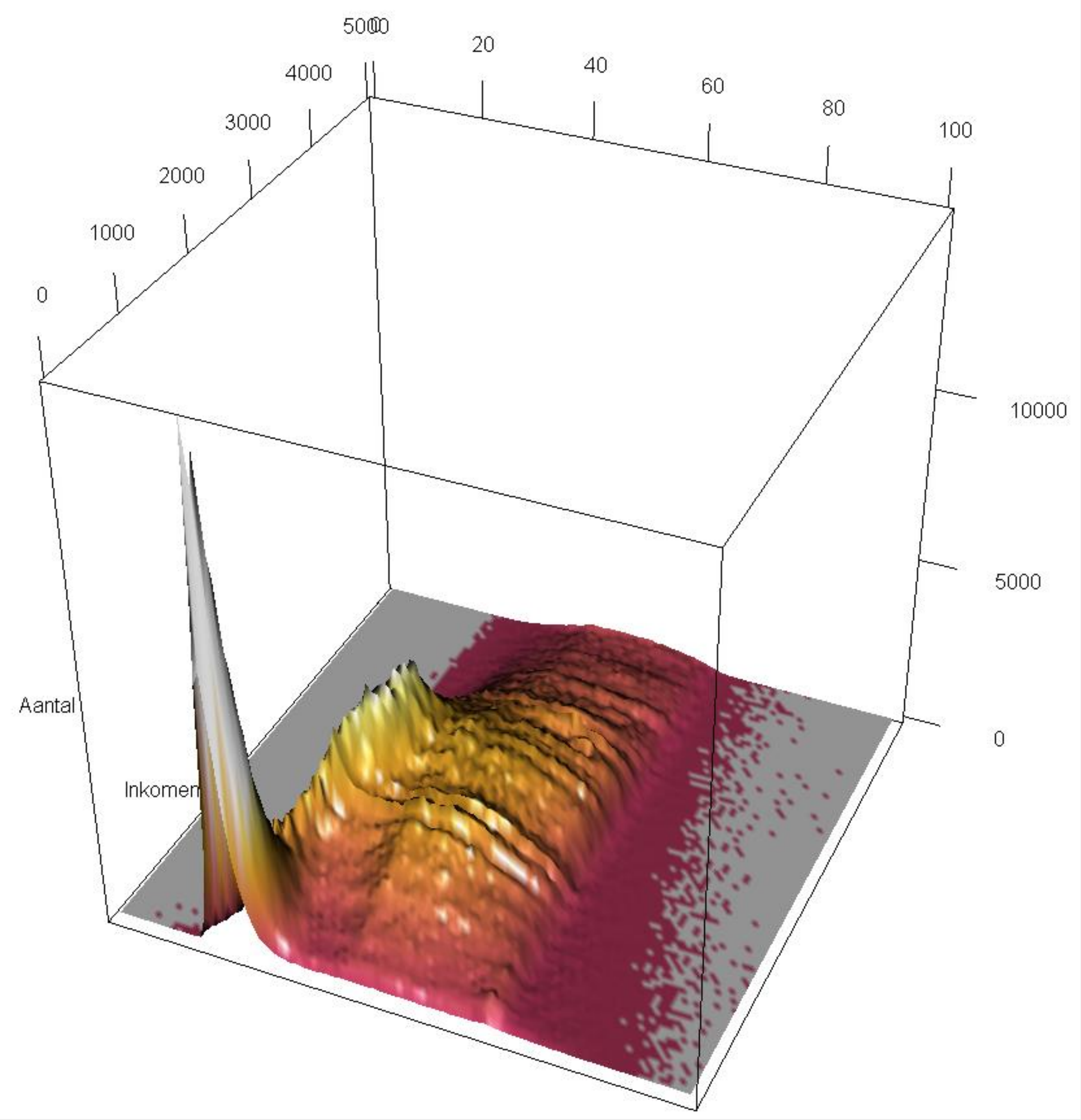
…but patterns need distinct colors

Color scales

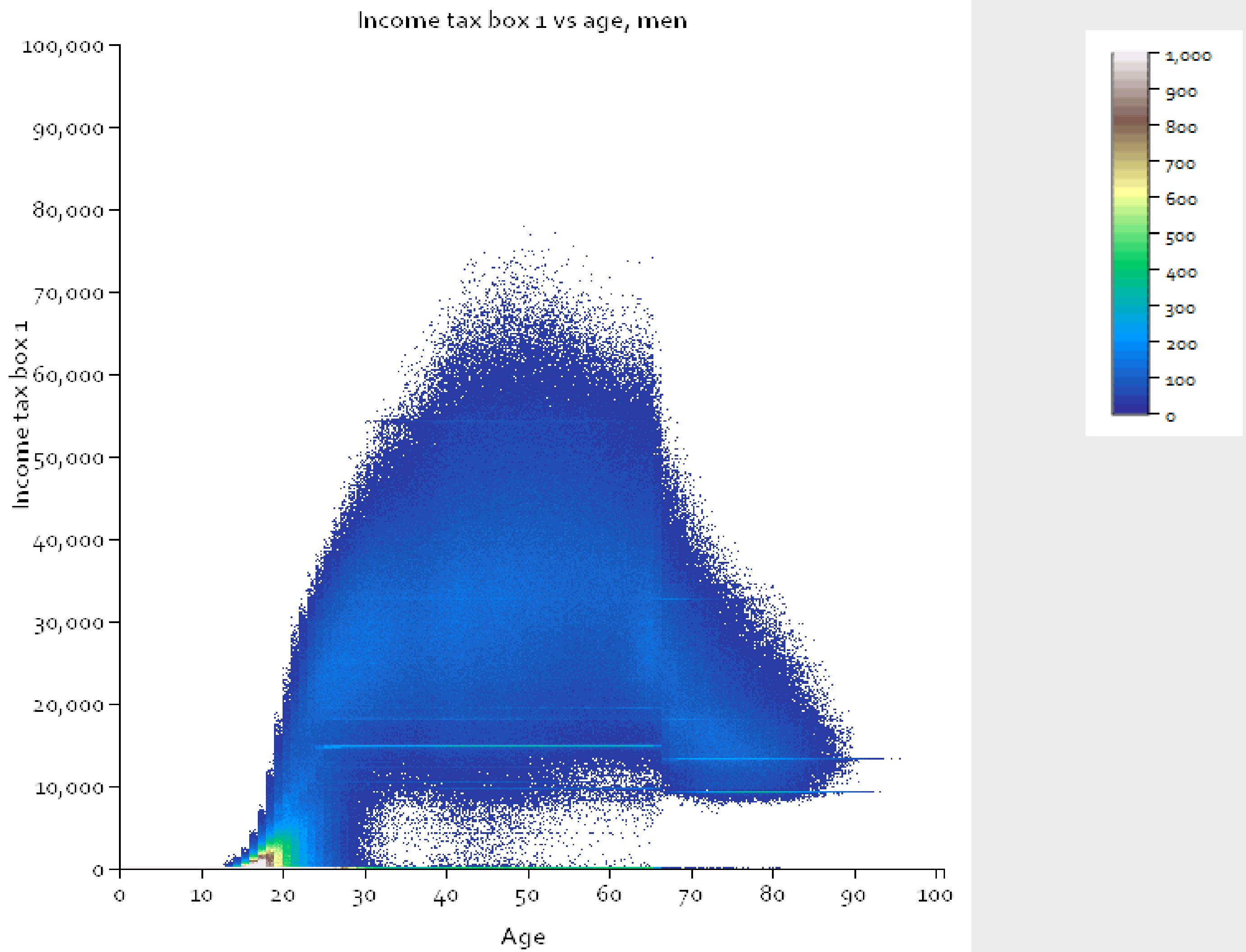
- Range of color scale depends on distribution of data.
- Often have multiple populations/distributions in data
- Severe spikes/stripes drown the smaller distributions:
  - We suggest log scale
  - Sometimes log scale is not enough

- In practice, linear scale with low maximum cut-off works well
- Effect is best understood in 3D (!).

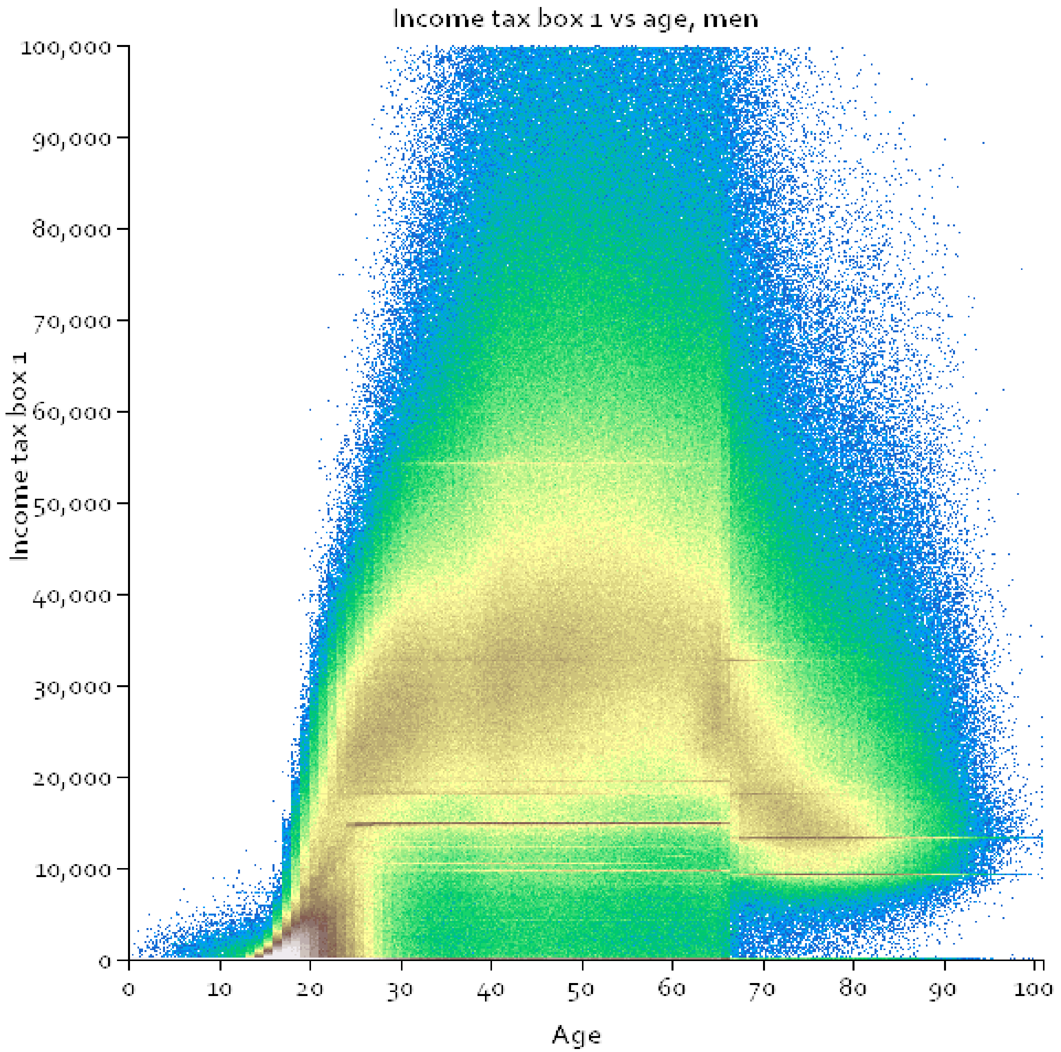
Peaks are best cut-off

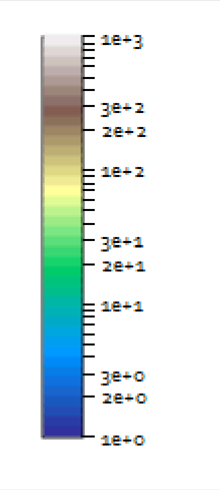


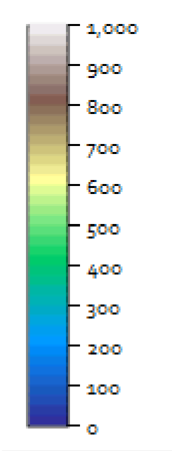
Example: Linear gradient



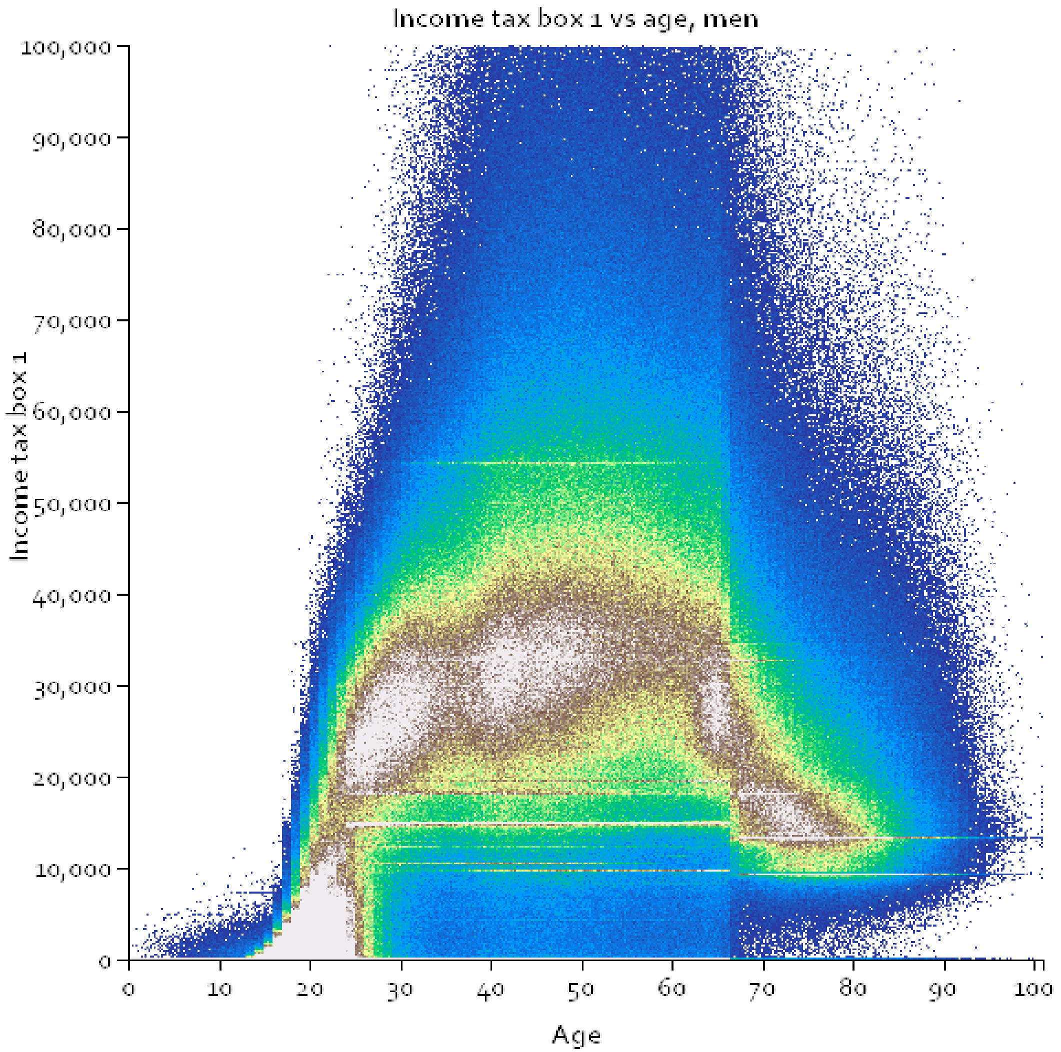
Log-gradient

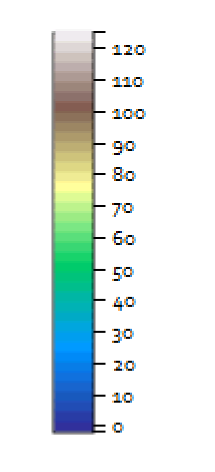




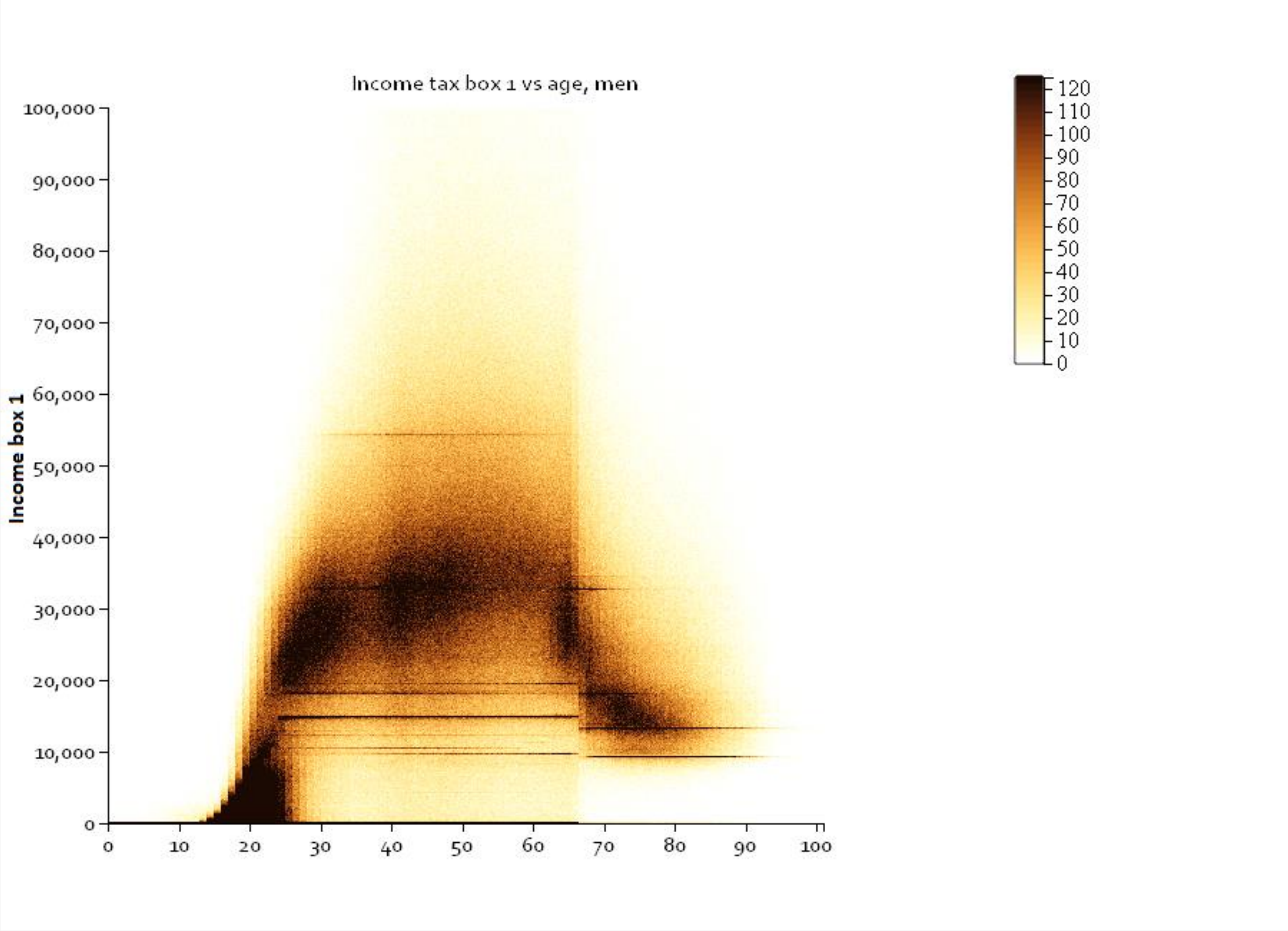


Linear gradient with cut-off

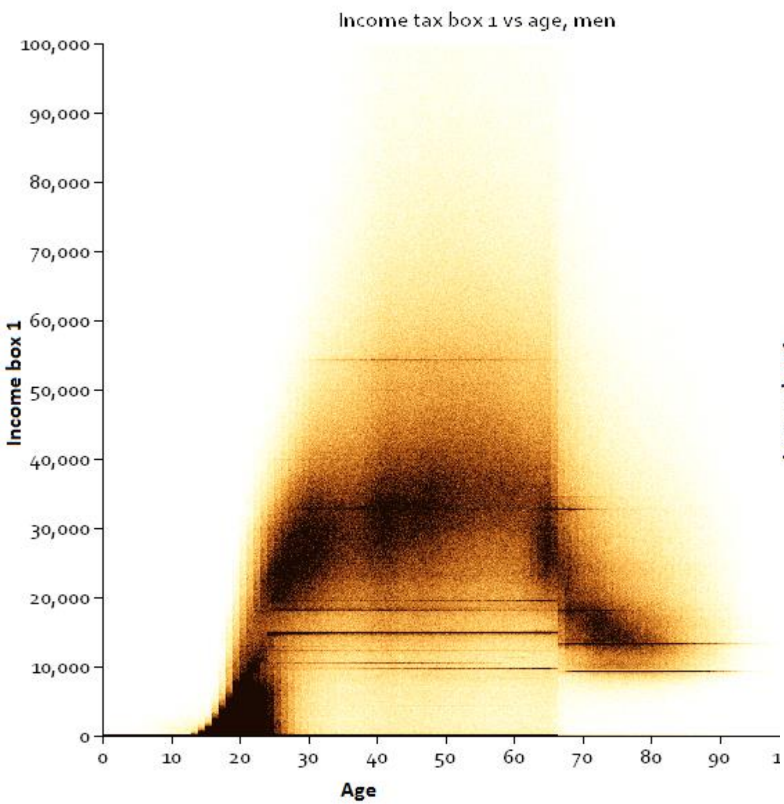


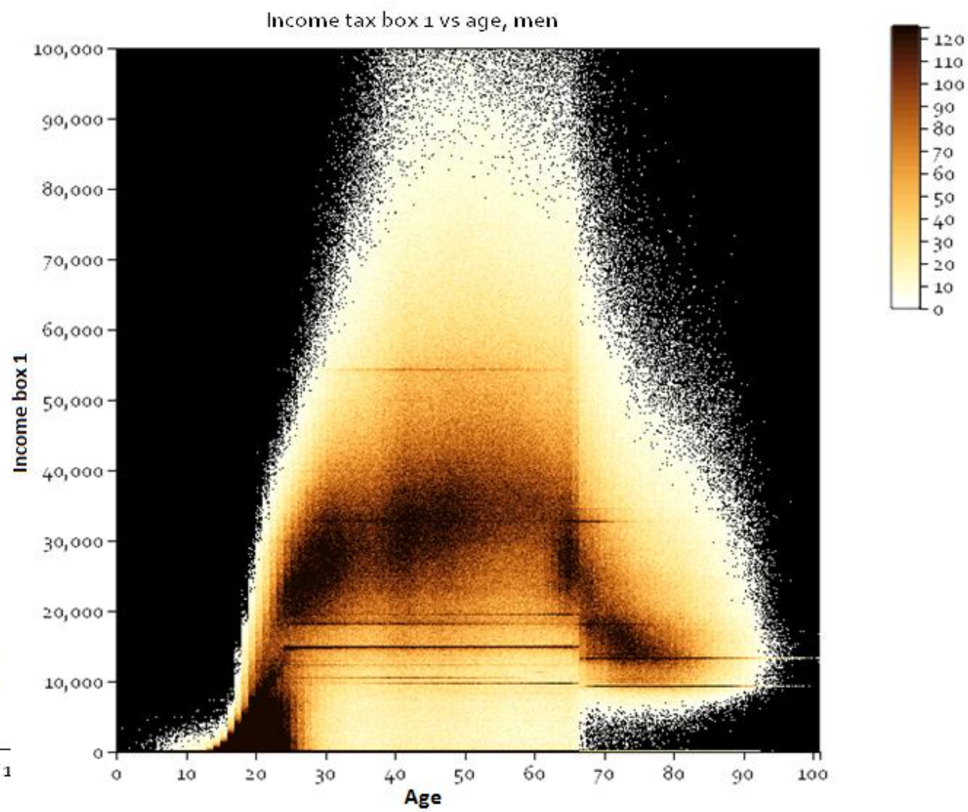


Perceptually uniform gradient

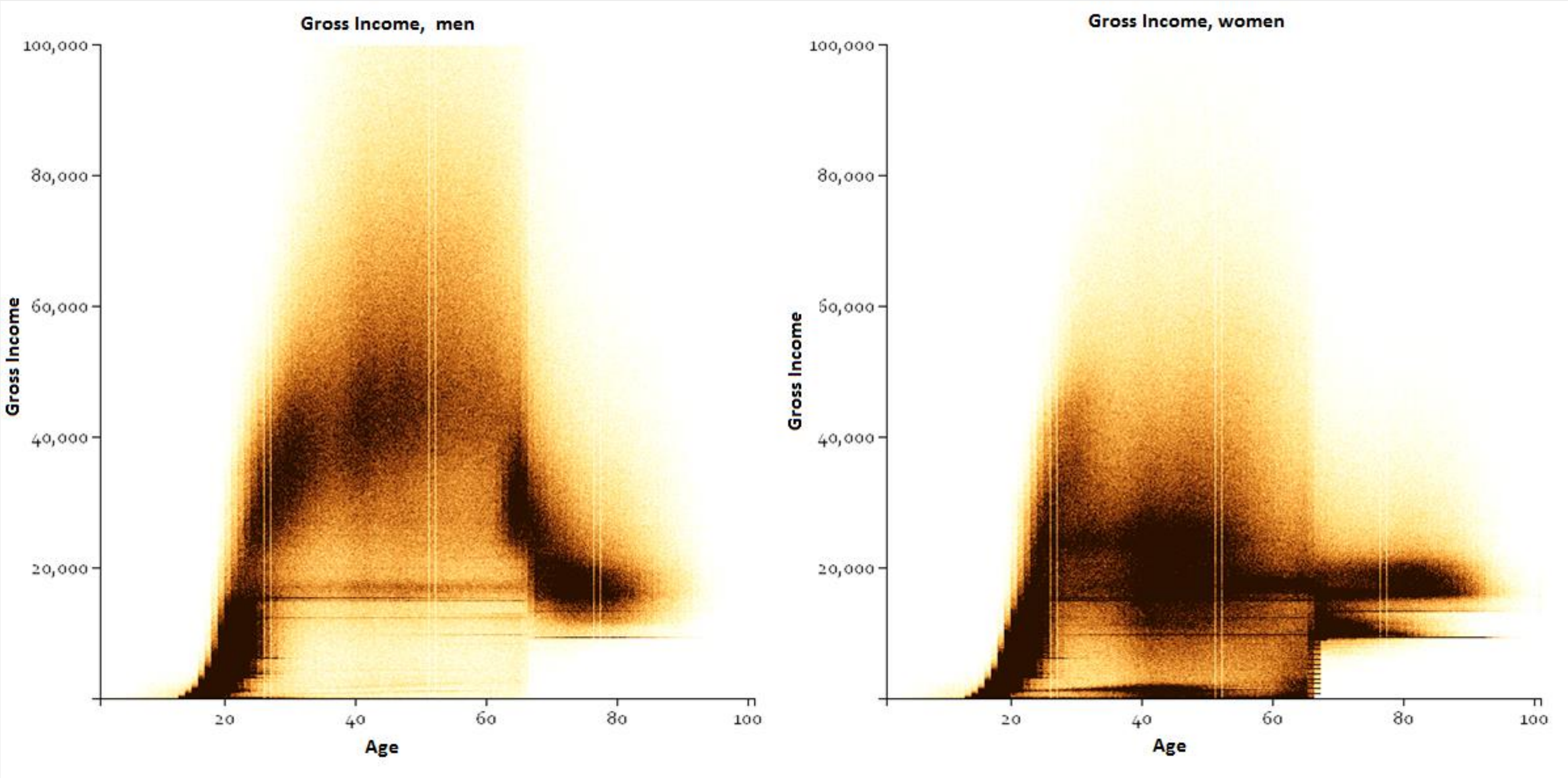


Colors: background/missings matters





Heatmaps side-by-side: gross income, men vs women



men

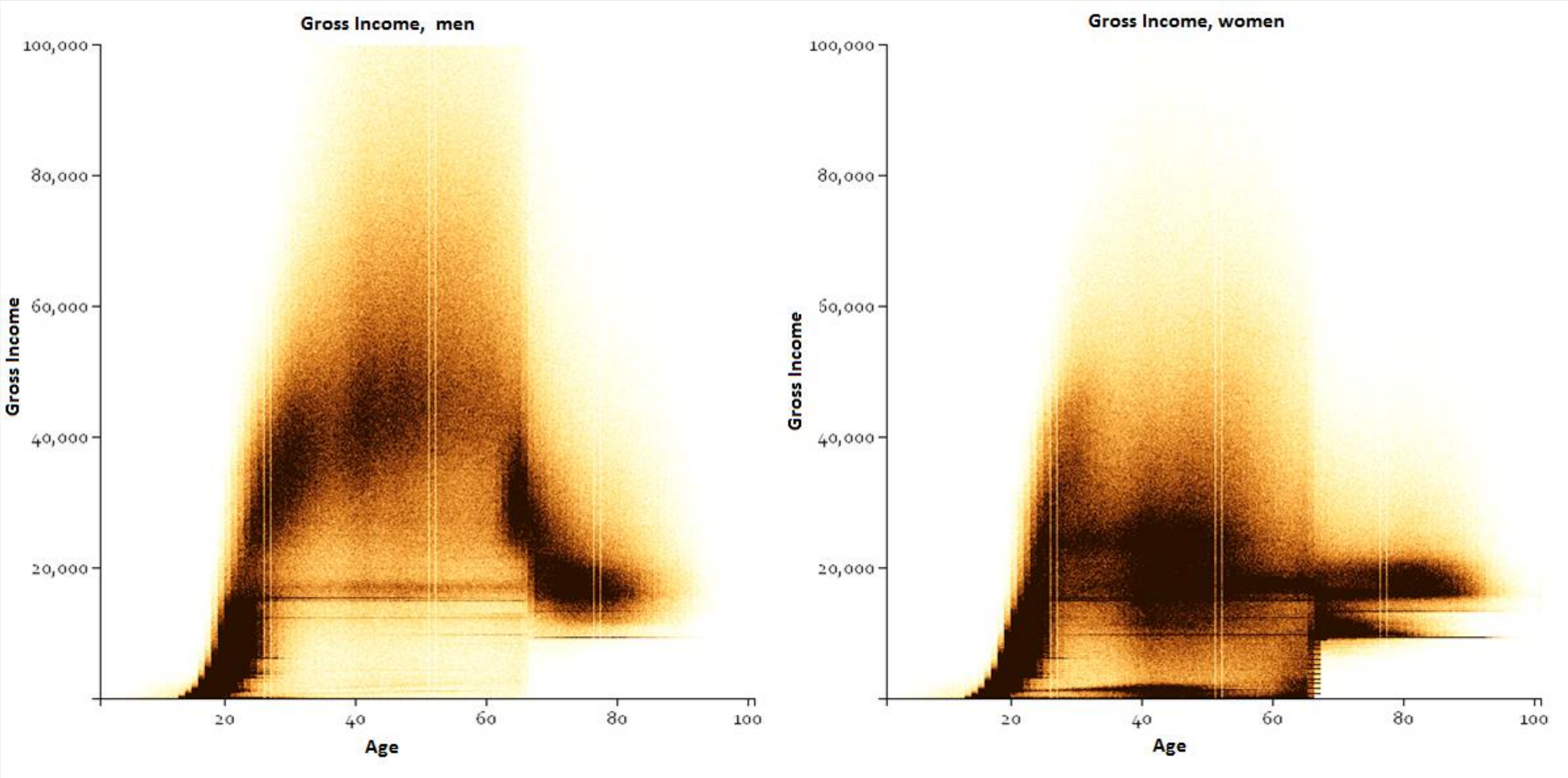
Meta pattern

Meta patterns constitutes of repeating pattern in:

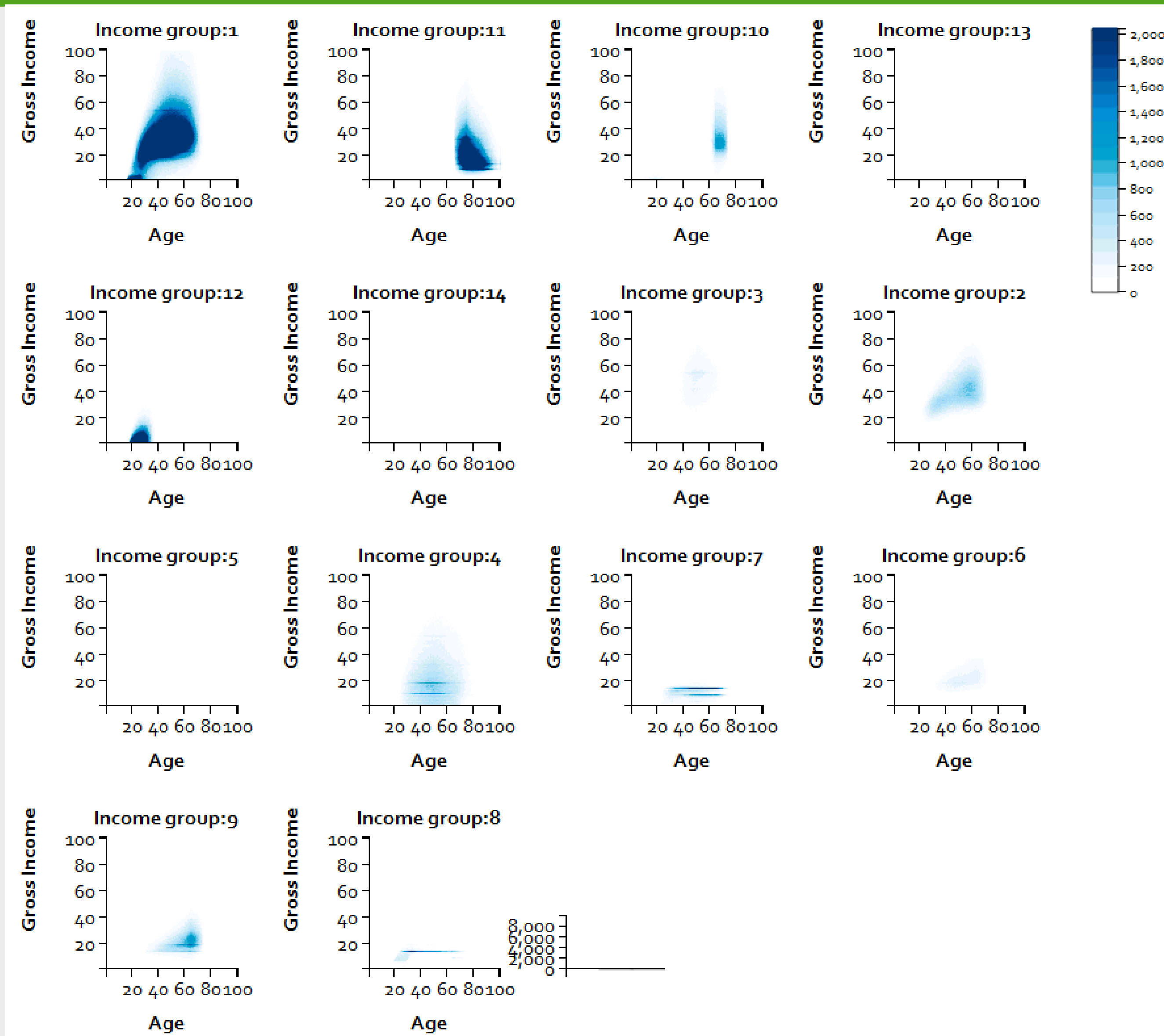
- **different subpopulations**
- different pairs of variables

So patterns that are generic over different heatmaps.

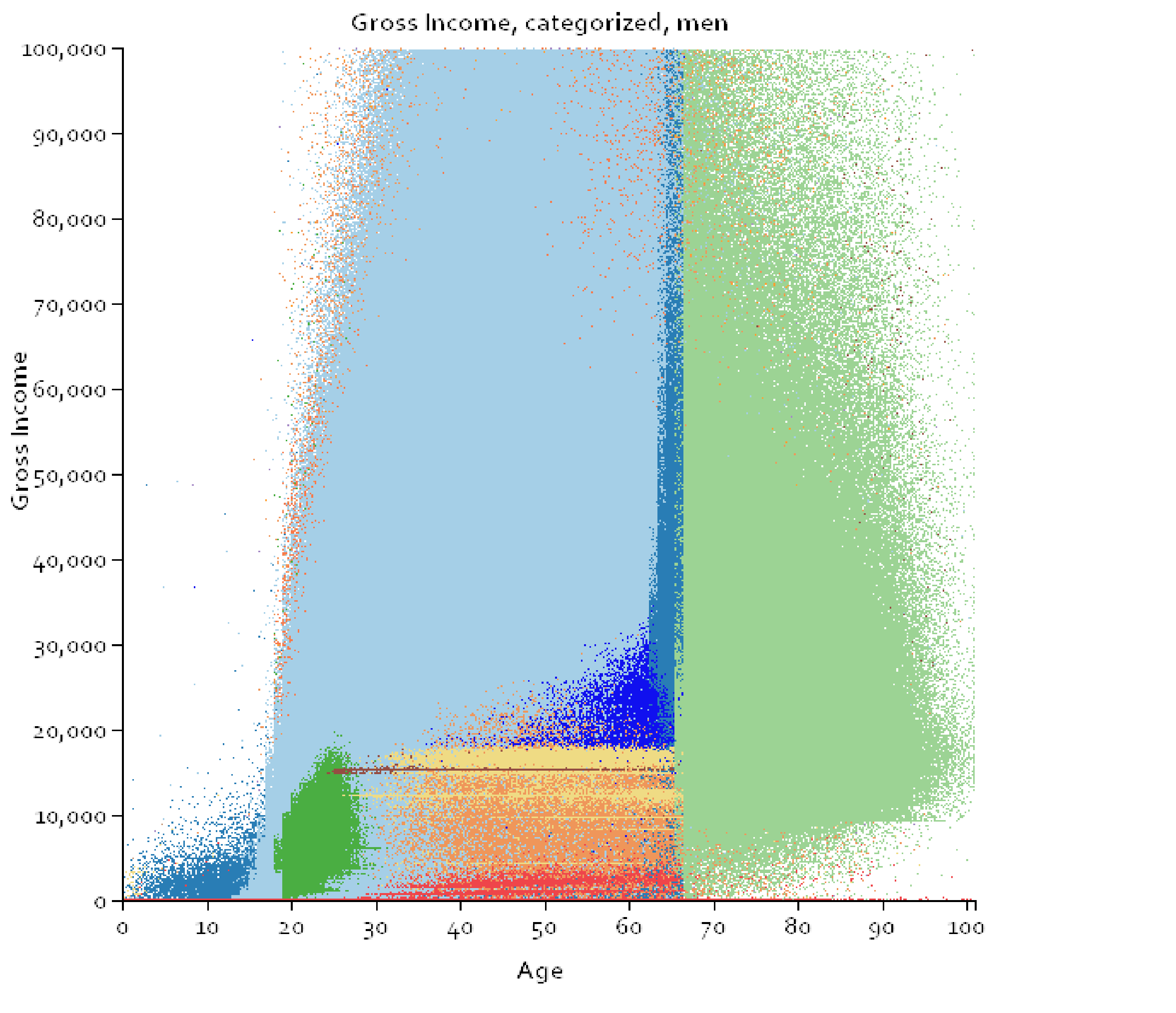
Heatmaps decomposed in subpopulations:



Gross income by socioeconomic status



Gross income, men, categorized by socioeconomic status



Patterns

- Stripes are real, not outliers:
- Corresponds with benefits, tax breaks
- Needs paradigm shift: data is not normally distributed (but we knew that).

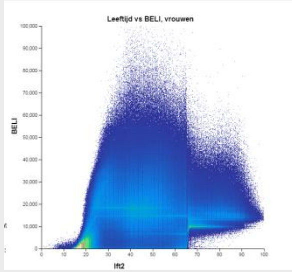
Meta pattern

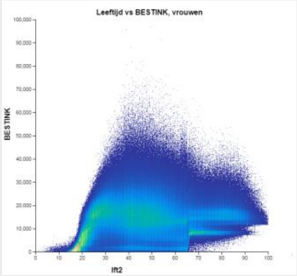
Meta patterns constitutes of repeating pattern in:

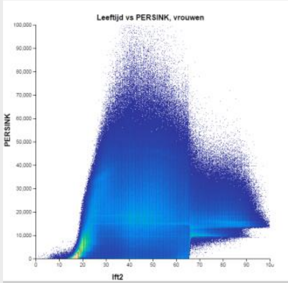
- different subpopulations
- **different pairs of variables**

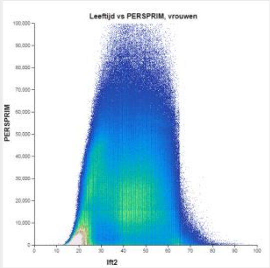
So patterns that are generic over different heatmaps.

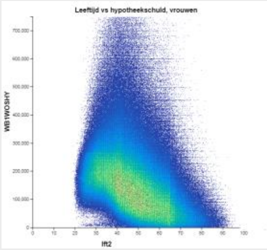
Image classification of heatmaps

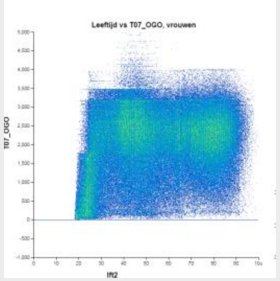


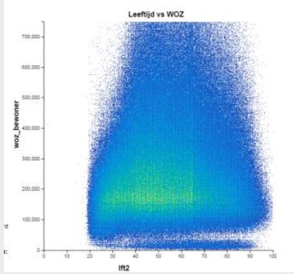


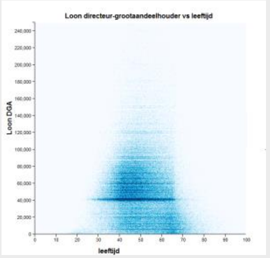


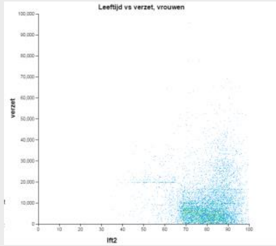


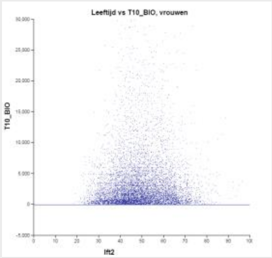


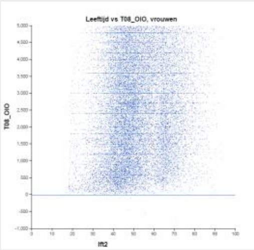




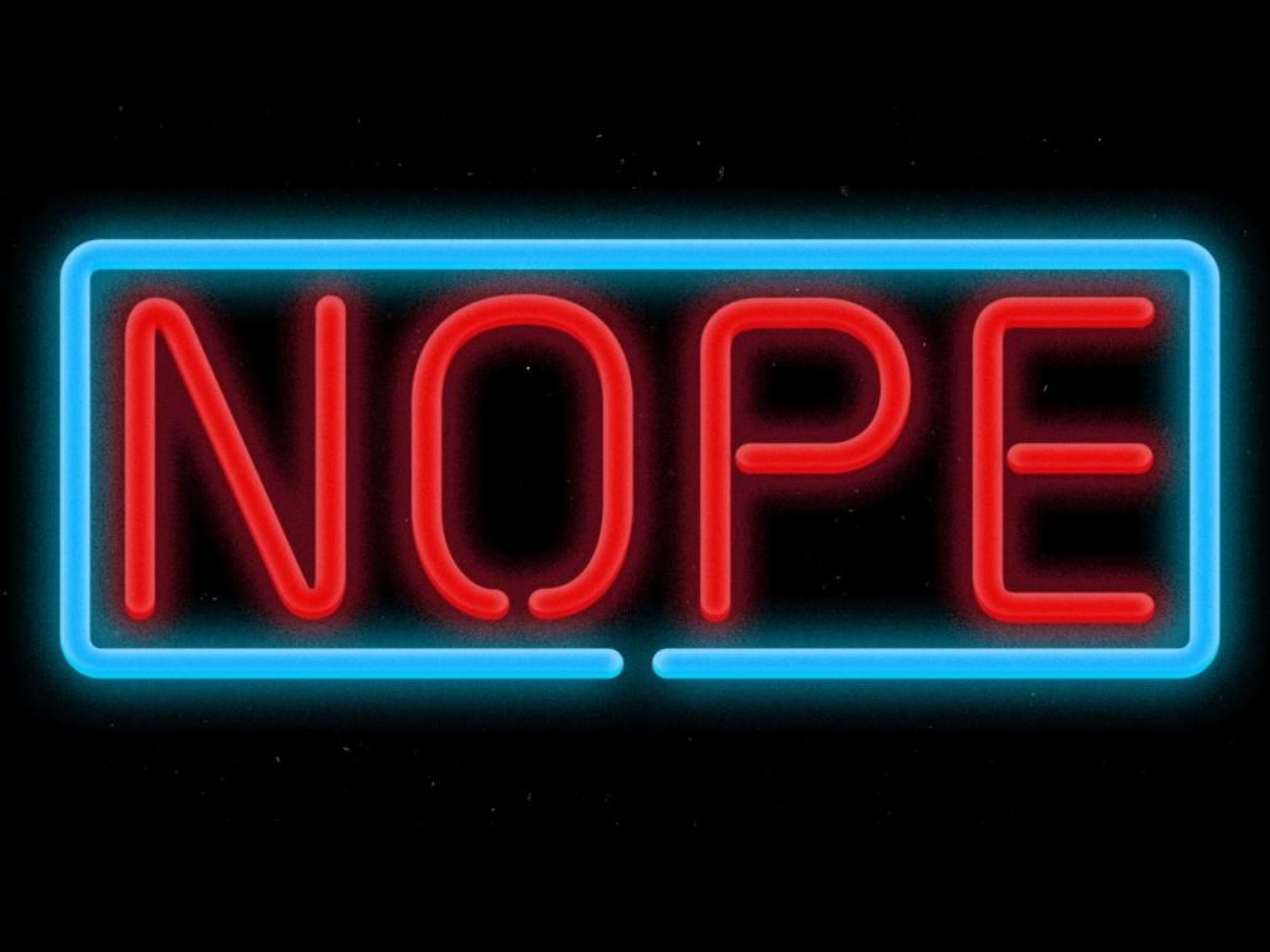




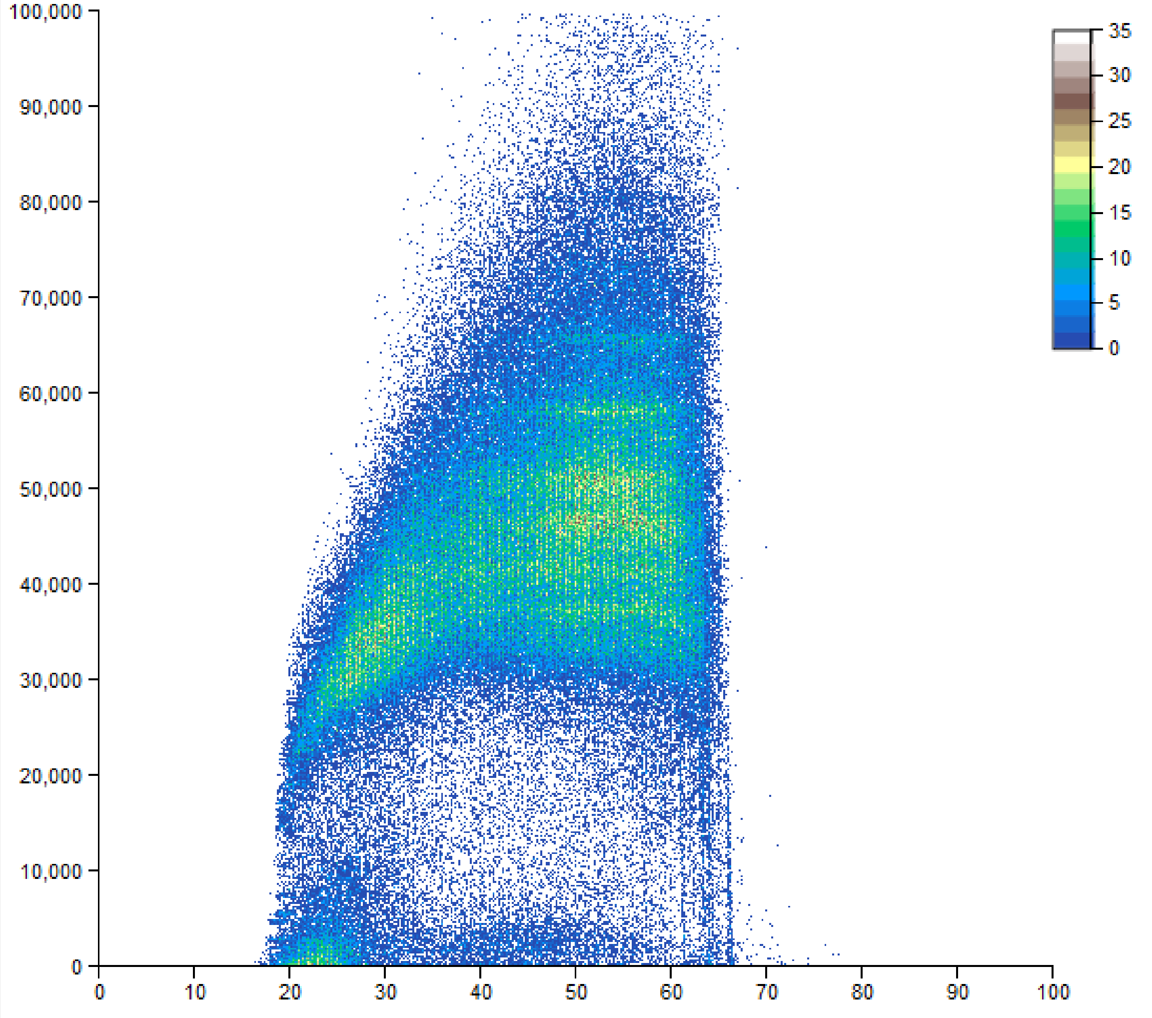




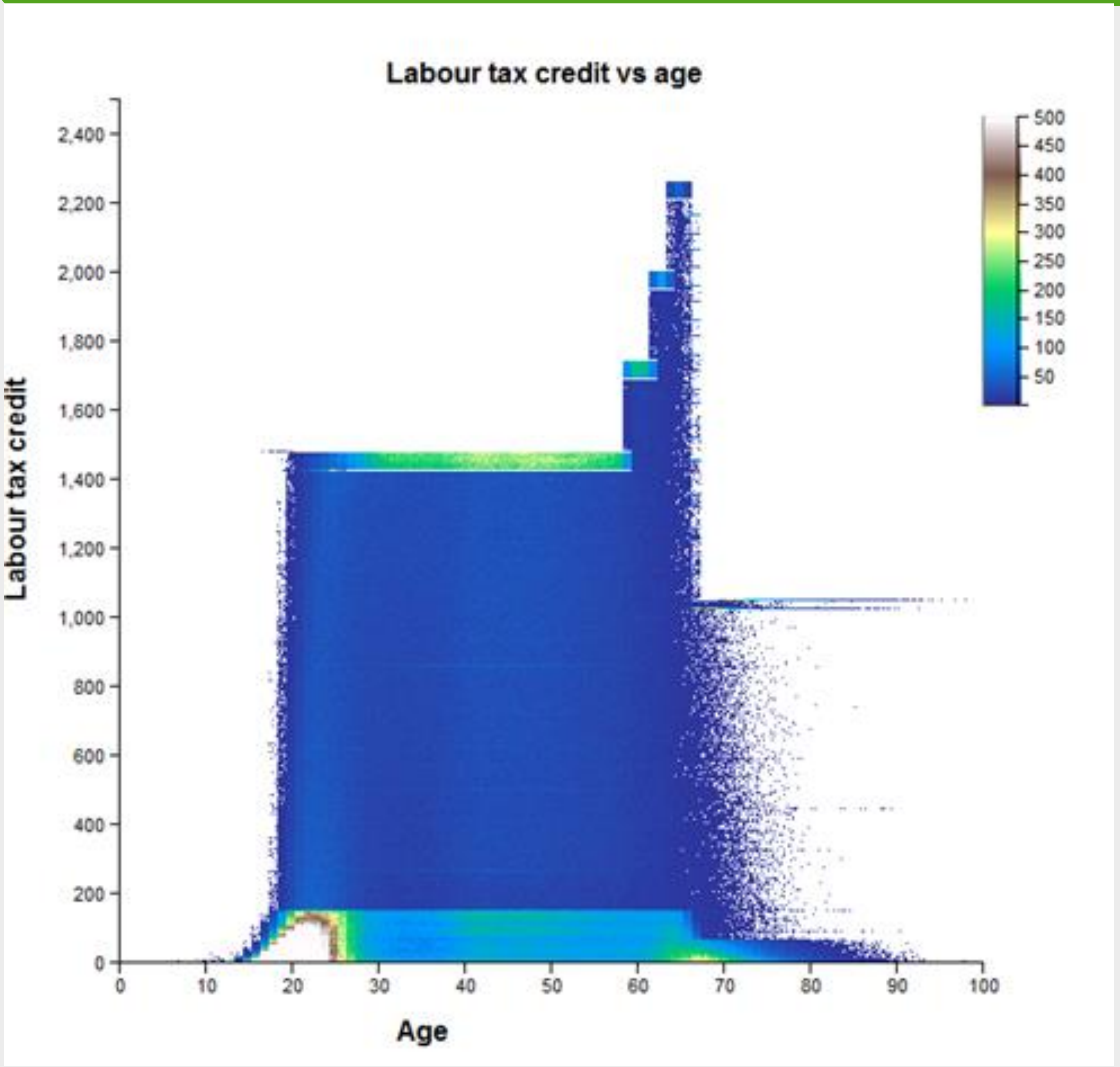
No Domain knowledge required?


NOPE

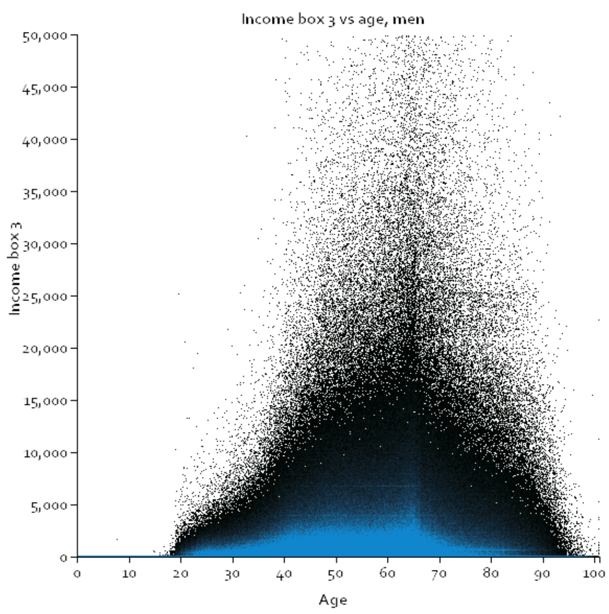
Salary pay structure

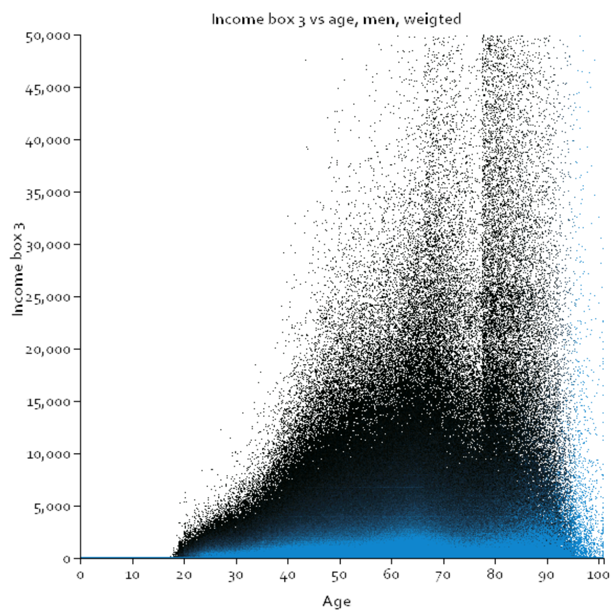


Domain knowledge, take II



Pattern removal: Effect of weighting





Summary

Heatmaps:

– ideal tool for analyzing big datasets

– Be aware of perceptual and data biases!

Questions?

Thank you for your attention!

More info?

ah.priem@cbs.nl / @_alex_priem

e.dejonge@cbs.nl / @edwindjonge

Heatmapping code available at https://github.com/alexpriem/heatmapr