# Macro-selection and micro-editing: a prototype

Wim Hacking, Roger Lemmens

Statistics Netherlands

# Problems with data-editing (read: challenges)

- Less funds supplied
- More electronic input
- More output demanded

# More efficient data editing

1. Electronic questionnaires with checks / feedback
2. Automatic corrections / imputations
3. Meso analysis for
   - manual edits only when having effect at the aggregate level
   - Verifying results step 1

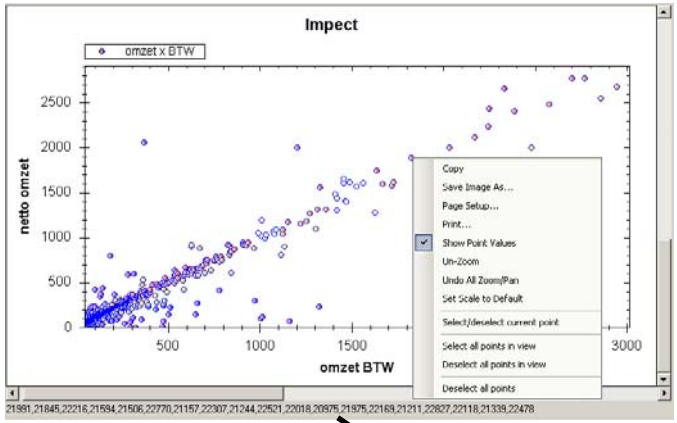
   → MacroView

mAcrodata (aggregated)

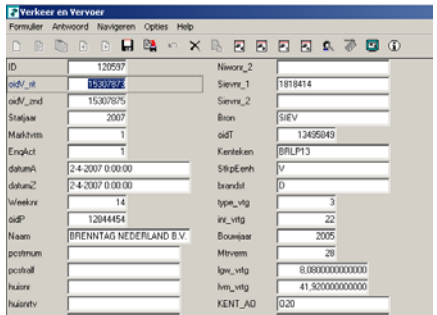
mIcrodata

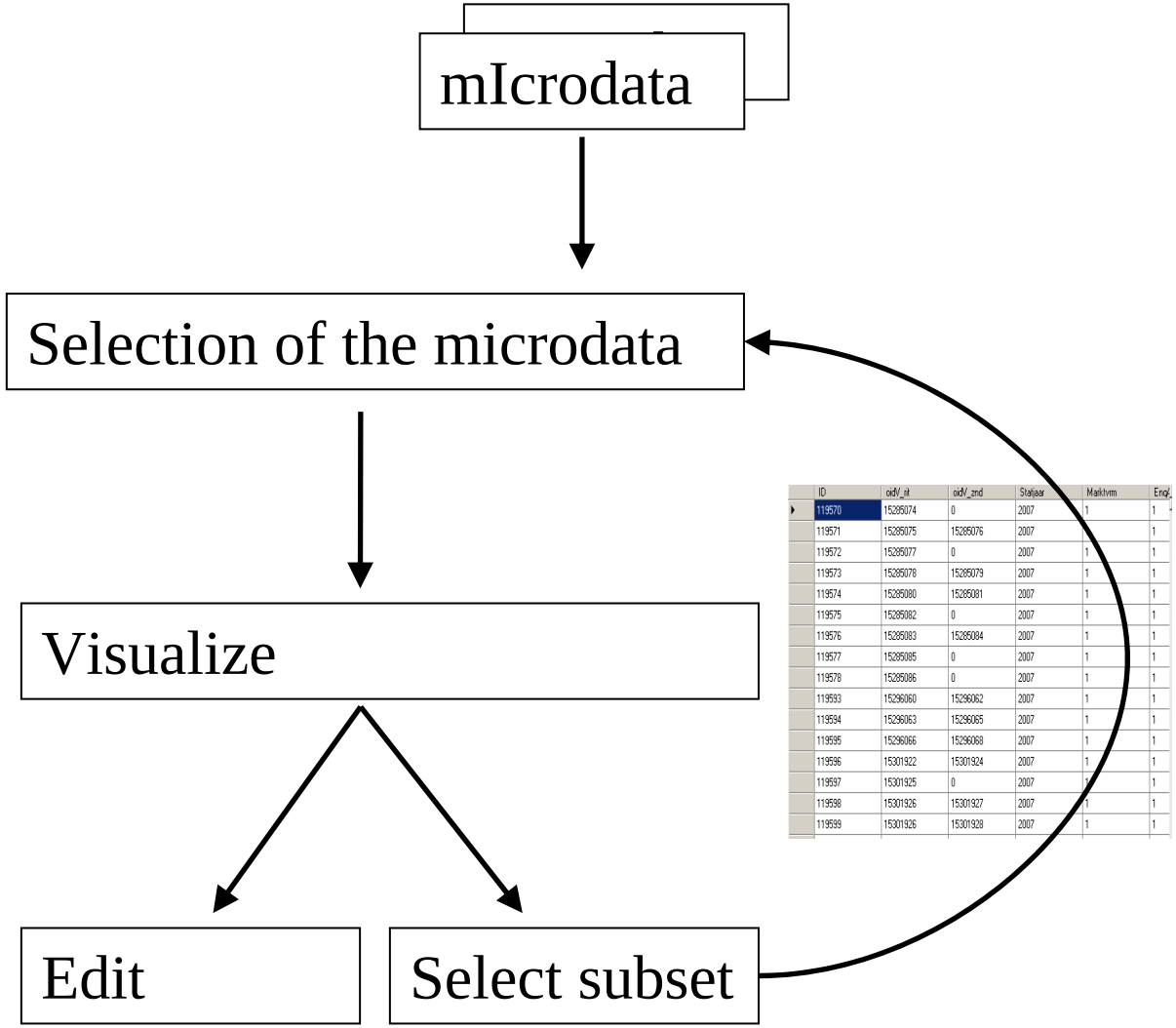
aggregate

Combined aggregate + plausibility functions




| | ID | oidV_rit | oidV_znd | Statjaar | Marktvrm | Enq |
|---|---|---|---|---|---|---|
| ▶ | 119570 | 15285074 | 0 | 2007 | 1 | 1 |
| | 119571 | 15285075 | 15285076 | 2007 | 1 | 1 |
| | 119572 | 15285077 | 0 | 2007 | 1 | 1 |
| | 119573 | 15285078 | 15285079 | 2007 | 1 | 1 |
| | 119574 | 15285080 | 15285081 | 2007 | 1 | 1 |
| | 119575 | 15285082 | 0 | 2007 | 1 | 1 |
| | 119576 | 15285083 | 15285084 | 2007 | 1 | 1 |
| | 119577 | 15285085 | 0 | 2007 | 1 | 1 |
| | 119578 | 15285086 | 0 | 2007 | 1 | 1 |
| | 119593 | 15296060 | 15296062 | 2007 | 1 | 1 |
| | 119594 | 15296063 | 15296065 | 2007 | 1 | 1 |
| | 119595 | 15296066 | 15296068 | 2007 | 1 | 1 |
| | 119596 | 15301922 | 15301924 | 2007 | 1 | 1 |
| | 119597 | 15301925 | 0 | 2007 | 1 | 1 |
| | 119598 | 15301926 | 15301927 | 2007 | 1 | 1 |
| | 119599 | 15301926 | 15301928 | 2007 | 1 | 1 |






| | ID | oidV_nil | oidV_znd | Statjaar | Markt\vrm | Enq |
|---|---|---|---|---|---|---|
| ▶ | 119570 | 15285074 | 0 | 2007 | 1 | 1 |
| | 119571 | 15285075 | 15285076 | 2007 | | 1 |
| | 119572 | 15285077 | 0 | 2007 | 1 | 1 |
| | 119573 | 15285078 | 15285079 | 2007 | 1 | 1 |
| | 119574 | 15285080 | 15285081 | 2007 | 1 | 1 |
| | 119575 | 15285082 | 0 | 2007 | 1 | 1 |
| | 119576 | 15285083 | 15285084 | 2007 | 1 | 1 |
| | 119577 | 15285085 | 0 | 2007 | 1 | 1 |
| | 119578 | 15285086 | 0 | 2007 | 1 | 1 |
| | 119593 | 15296060 | 15296062 | 2007 | 1 | 1 |
| | 119594 | 15296063 | 15296065 | 2007 | 1 | 1 |
| | 119595 | 15296066 | 15296068 | 2007 | 1 | 1 |
| | 119596 | 15301322 | 15301324 | 2007 | 1 | 1 |
| | 119597 | 15301325 | 0 | 2007 | 1 | 1 |
| | 119598 | 15301326 | 15301327 | 2007 | 1 | 1 |
| | 119599 | 15301326 | 15301328 | 2007 | 1 | 1 |

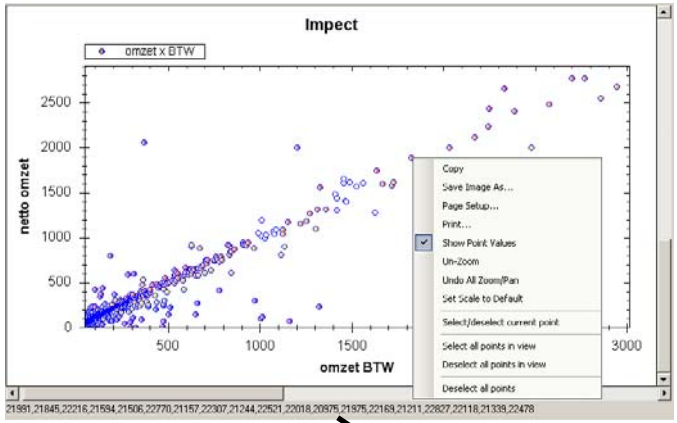
# → MacroView POC

- Text-based (for now)
- Must be flexible
  - For prototyping
  - Final version must be usable for different statistics
- Input from & application to:
  - Transport
  - Production

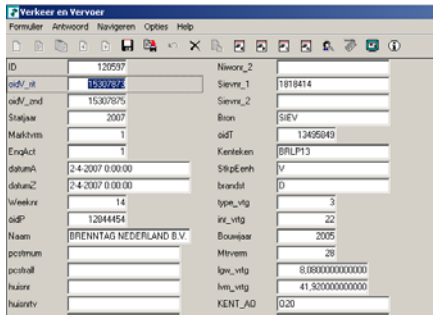
mAcrodata (aggregated)

mIcrodata

Combined aggregate + plausibility functions




| | ID | oidV_rit | oidV_znd | Statjaar | Marktvrm | Enq |
|---|---|---|---|---|---|---|
| ▶ | 119570 | 15285074 | 0 | 2007 | 1 | 1 |
| | 119571 | 15285075 | 15285076 | 2007 | 1 | 1 |
| | 119572 | 15285077 | 0 | 2007 | 1 | 1 |
| | 119573 | 15285078 | 15285079 | 2007 | 1 | 1 |
| | 119574 | 15285080 | 15285081 | 2007 | 1 | 1 |
| | 119575 | 15285082 | 0 | 2007 | 1 | 1 |
| | 119576 | 15285083 | 15285084 | 2007 | 1 | 1 |
| | 119577 | 15285085 | 0 | 2007 | 1 | 1 |
| | 119578 | 15285086 | 0 | 2007 | 1 | 1 |
| | 119593 | 15296060 | 15296062 | 2007 | 1 | 1 |
| | 119594 | 15296063 | 15296065 | 2007 | 1 | 1 |
| | 119595 | 15296066 | 15296068 | 2007 | 1 | 1 |
| | 119596 | 15301922 | 15301924 | 2007 | 1 | 1 |
| | 119597 | 15301925 | 0 | 2007 | 1 | 1 |
| | 119598 | 15301926 | 15301927 | 2007 | 1 | 1 |
| | 119599 | 15301926 | 15301928 | 2007 | 1 | 1 |



# Specification of microdata

```
DATAMODEL MyData

        primary
                ID
        weight
                TheWeight

        FIELDS

            ID                  :integer
            InterviewId         :integer
            TransportedWeight   :real , ifmissing(0.0)
            TransportType       :integer, ifmissing(0)
            Distance            :real
            TheWeight           :real
            Quarter             :integer
            NstrCode            :integer
        RULES

        IF Type = ‘X’ THEN
            Distance < 2000 “Distance for type X should not exceed 20
        ENDIF
ENDMODEL
```

# Specification of macrodata

```
DATAMODEL MyAggregateData

      AGGREGATEBY
              AggDef =   Quarter *  NstrCode

      FILTERBY
              Filter1 =  'Profit > 1000'

      FIELDS

              Quarter: integer
              NSTRCode: integer

              Average_Distance :real
              Sum_Weights :real
              …
ENDMODEL
```

| Quarter | NSTR | Average_ Distance | Sum_ Weights |
|---|---|---|---|
| 1 | 1 | | |
| 1 | 2 | | |
| … | … | | |

# Specification of macrodata

```
DATAMODEL MyAggregateData

        AGGREGATEBY
                AggDef =   Quarter *  NstrCode
                AggDef2 =    Quarter

        FILTERBY
                Filter1 =  'Profit > 1000'
                Filter2 =  'Profit <= 1000'

        FIELDS

                Quarter: integer
                NSTRCode: integer

                Average_Distance :real
                Sum_Weights :real
                …
ENDMODEL
```

| Quarter | Average_ Distance | Sum_ Weights |
|---|---|---|
| 1 | | |
| 2 | | |
| … | | |

| Quar ter | NSTR | Average_ Distance | Sum_ Weights |
|---|---|---|---|
| 1 | 1 | | |
| 1 | 2 | | |
| … | … | | |

# Specification of aggregate calculation: how to detect anomalies?

1. Distribution properties of the micro data, e.g. its variance.

2. Processing properties
   - % item non-response
   - % previously automatically imputed values in field X

3. Plausibility functions: e.g. the relative change between weighted t-1 and t data:

$$\Delta = \frac{\left| \sum w_{i,t-1} V_{i,t-1} - \sum w_{i,t} V_{i,t} \right|}{w_{i,t-1} V_{i,t-1}}$$

# Comparing aggregate values

| Quarter | Average_ Distance(t-1) | Average_ Distance(t) | Delta | CV_Distance |
|---|---|---|---|---|
| 1 | 2000 | 3500 | 0.75 | 4.5 |
| 2 | 1500 | 1700 | 0.14 | 2.4 |
| 3 | 2300 | 2100 | 0.09 (2,3) | 1.6 (1) |
| 4 | | | | |

# Specification of an aggregate

```
Aggregate MyAgg
INPUT
        CY = MyData
        LY_Agg = MyAggregateData
output
        outputagg = MyAggregatedDataCompared
cells

        Difference := abs(LY_Agg.average_distance -
AVG(CY.Distance))/AVG(LY.Distance) ;

        Med_Dist := Median( CY.Distance  );
Cellcompare
        Aggdef2:
        ratio := average_distance(1) / average_distance(2)
            > 2.0 "There should be more transport in the
                  first quarter!";
EndAggregate
```

$$\frac{\left|\overline{D_{t-1}} - \overline{D_t}\right|}{\overline{D_{t-1}}}$$

| Quarter | Average_ Distance | Sum_ Weights |
|---|---|---|
| 1 | 1.91 | … |
| 2 | 0.89 | … |
| … | … | … |

# Specification of a grid

```
Grid Kerncel_Grid
input = MyAggregateData

rules
        color(average_distance, difference, 0, 30.0, Green, Red)

EndGrid
```

DataCompared

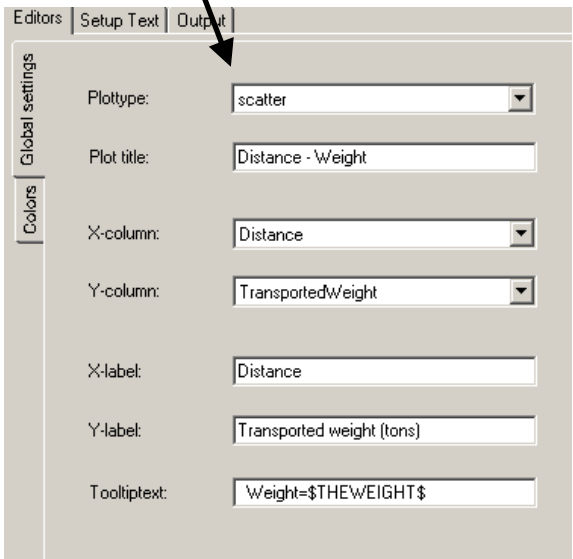
| Nstrcode | 2 | 4 | 3 | 1 |
|---|---|---|---|---|
| 5 | 187,5187464010... | 191,4395028155... | 179,4129263913... | 205,2459405419... |
| 9 | 188,4469635330... | 208,7887405899... | 196,5722122426... | 203,6853819410... |
| 1 | 168,6020864953... | 196,3599714474... | 181,6441893143... | 187,2453554585... |
| 7 | 77,32265762053... | 91,67588217924... | 100,8406390823... | 84,32761936580... |
| 6 | 100,8177237713... | 119,4111116418... | 92,51071640565... | 99,73734611911... |
| --- | --- | --- | --- | --- |
| 3 | 131,9715977787... | 136,3810754699... | 137,2245694952... | 173,7608454718... |
| 4 | 100,6468784322... | 100,0976217977... | 109,7554664419... | 83,76972916386... |
| 8 | 125,3725311203... | 131,9401462608... | 139,0790815251... | 149,4032040016... |
| 0 | 205,7172767994... | 181,8809643032... | 206,0841884122... | 190,3850245341... |
| 2 | 210,2635836591... | 226,3043623321... | 214,0167065868... | 217,6009087066... |

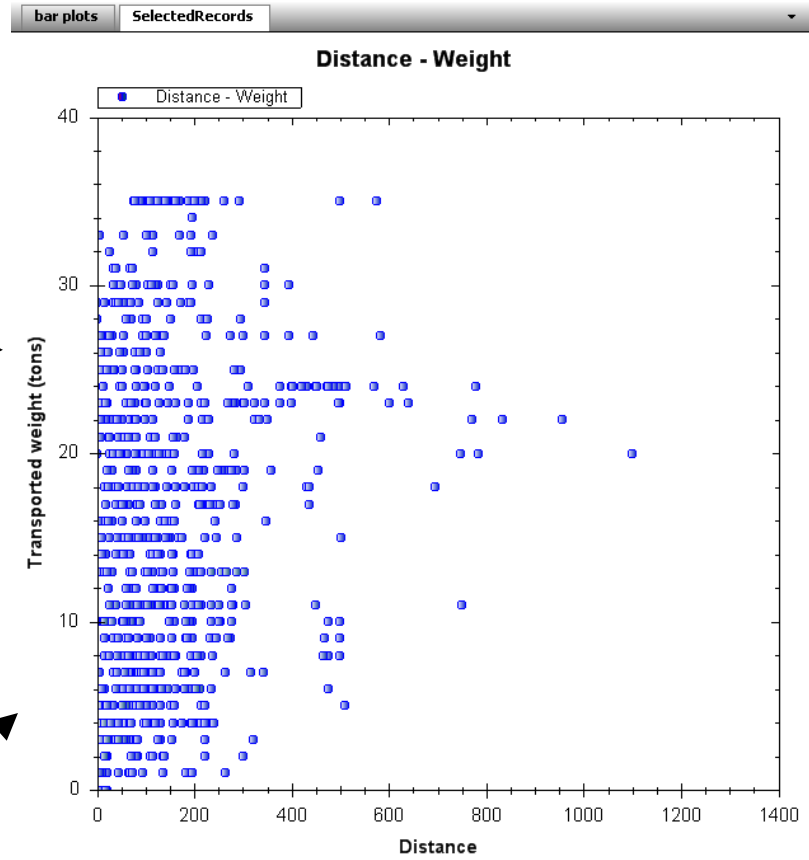
3,360358

**NstrCode * Quarter**

# Specification of a plot

```
Plot MicroPlot
input = MyData

type = scatter

xcolumn = Distance
YColumn = TransportedWeight
ZColumn = TheWeight
mincolorvalue = 1.0
maxcolorvalue = 500.0
color = (Red, Blue, Green)

TooltipText='Weight=$THEWEIGHT$'
endplot
```

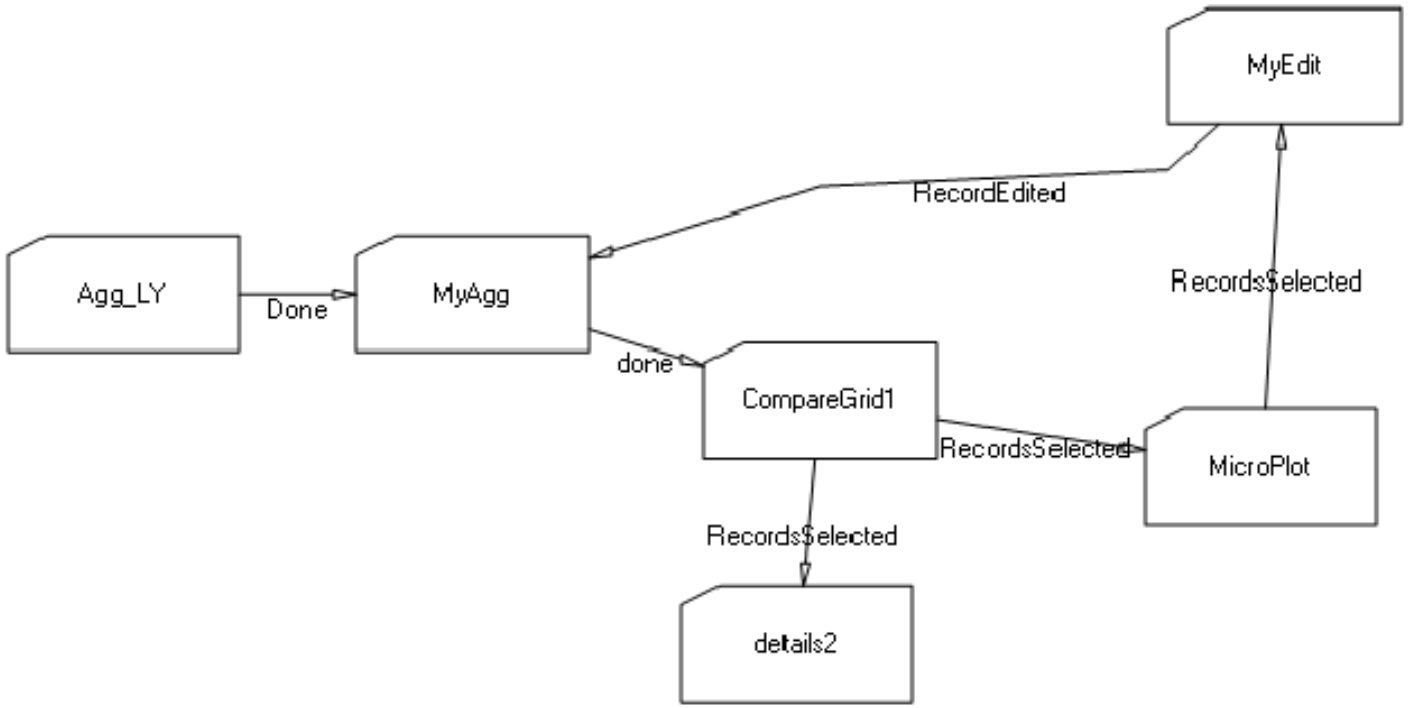


# Specification of the proces

```
Proces

Agg_LY(Data_LY, DataMacro_LY)
Agg_LY.Done -> MyAgg(Data, DataMacro_LY, DataMacro)
MyAgg.done -> comparegrid1(av.outputdata) AT Demo Position Top
Label 'DataCompared'

comparegrid1.RecordsSelected ->
MicroPlot(comparegrid1.SelectedRecords) AT Demo POSITION Tab
Label 'SelectedRecords'

comparegrid1.RecordsSelected ->
Details2(comparegrid1.SelectedRecords) AT Demo POSITION Left
Label 'MicroData'

MicroPlot.RecordsSelected -> MyEdit(MicroPlot.SelectedRecords)
MyEdit.RecordEdited -> MyAgg(Data, DataMacro_LY, DataMacro)

Endproces
```

# Specification of the proces (2)



# Micro-editing

MicroData

| ID | Distance | Quarter | NstrCode |
|---|---|---|---|
| 166593 | 145 | 2 | 1 |
| 166594 | 95 | 2 | 1 |
| 166595 | 120 | 2 | 1 |
| 166653 | 204 | 2 | 1 |
| 166707 | 145 | 2 | 1 |
| 166710 | 145 | 2 | 1 |
| 166713 | 145 | 2 | 1 |

n Vervoer

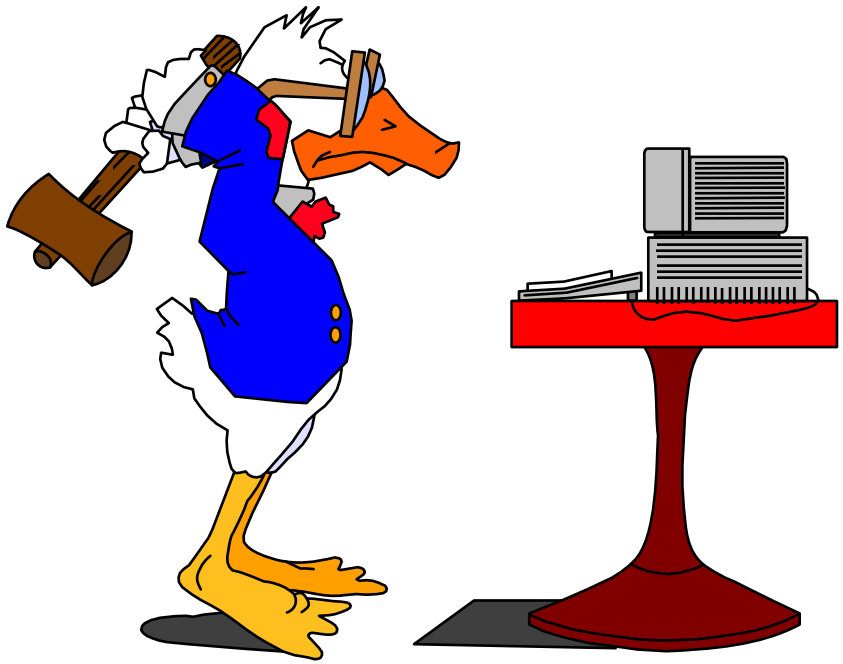
Forms Answer Navigate Options Help

| | |
|---|---|
| ID | 166899 |
| TransportedW | 13,000000000000000 |
| TransportType | 2 |
| Distance | 760,00000000000000 |
| TheWeight | 145,41993495312800 |
| Quarter | 2 |
| NstrCode | 1 |

# MacroView: currently

- Testing with real data
  - Road transport data (~demo)
  - Production statistics
  - Generating new ideas for the POC
- Investigating the possible integration with Blaise
  - Data layer
  - Syntax
- Making the POC more robust
  - Better checking of illegal syntax

# A demo…



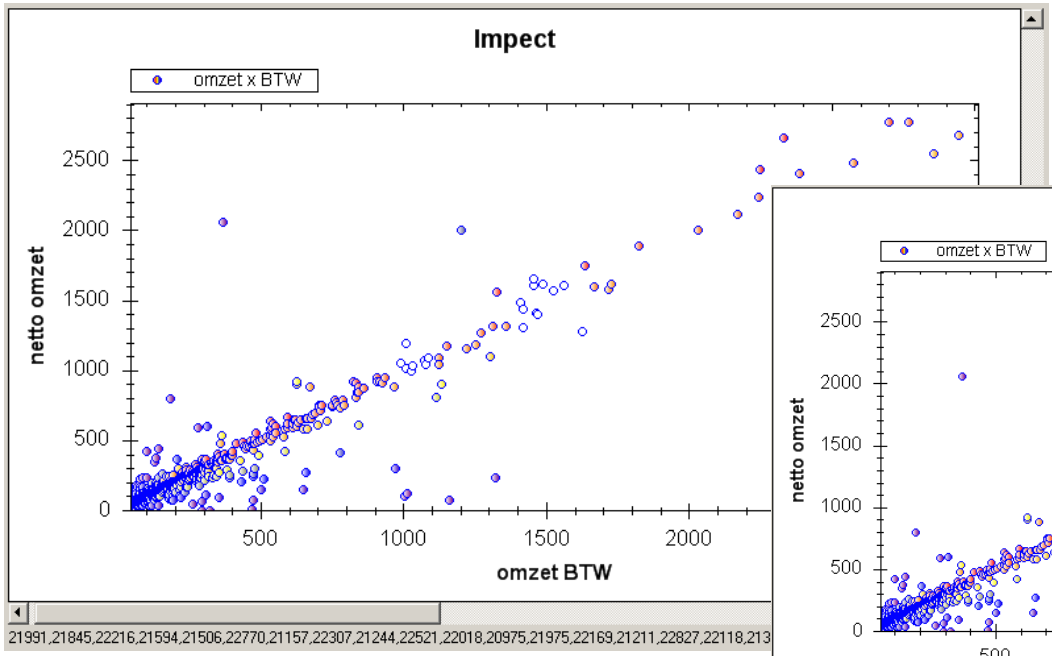
# Questions ?

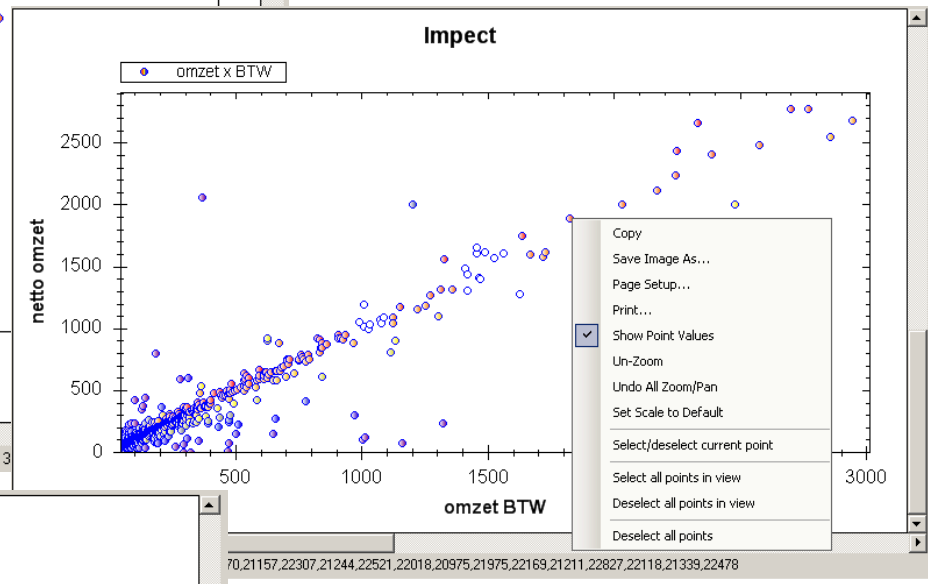
# Grid met aggregaat + aankleuring

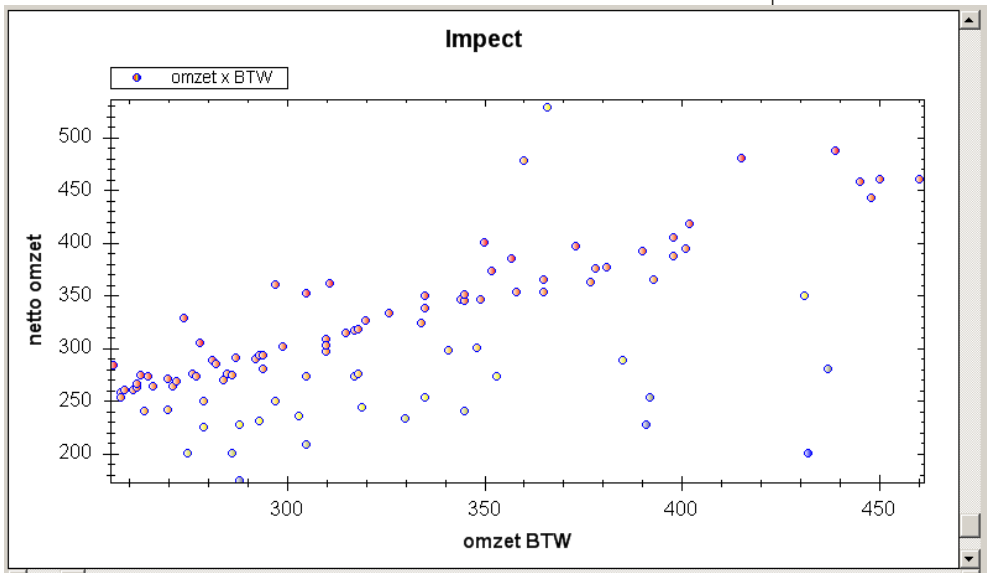
| ID | KERNCEL | btw_totaal | omzet_gemiddeld | plausibiliteitsindex |
|---|---|---|---|---|
| 12806 | 29100 | 59530 | 1175,92 | 1 |
| 12844 | 29300 | 65214 | 1758,9211 | 1 |
| 12852 | 29400 | 9206 | 1118,625 | 1 |
| 12883 | 29500 | 46303 | 1346,4839 | 1 |
| 12884 | 29600 | 364 | 361 | 0 |
| 12892 | 29700 | 11663 | 1029,875 | 1 |
| 12916 | 31100X | 43278 | 1795,1667 | 1 |
| 12948 | 31620X | 62377 | 1874,5625 | 1 |
| 12967 | 32100 | 54011 | 2176,8421 | 1 |
| 12973 | 32300X | 6401 | 1142 | 1 |
| 12975 | 33101 | 531 | 259,5 | 0 |

| | | | iddeld | PLAUSIBILITEITSI |
|---|---|---|---|---|
| 11578 | 15120 | 7767 | 963,125 | 1 |
| 11592 | 15130 | 48936 | 3430,2857 | 1 |
| 11605 | 15200 | 28904 | 2190 | 1 |
| 11611 | 15310 | 7565 | 1224,6667 | 1 |
| 11617 | 15330X | 14821 | 2512,1667 | 1 |
| 11621 | 15400 | 16733 | 4129,25 | 1 |
| 11654 | 15500 | 18498 | 445,7879 | 0 |
| 11667 | 15610 | 16612 | 1163,4615 | 1 |

# Plots met aankleuring/zoom/selectie

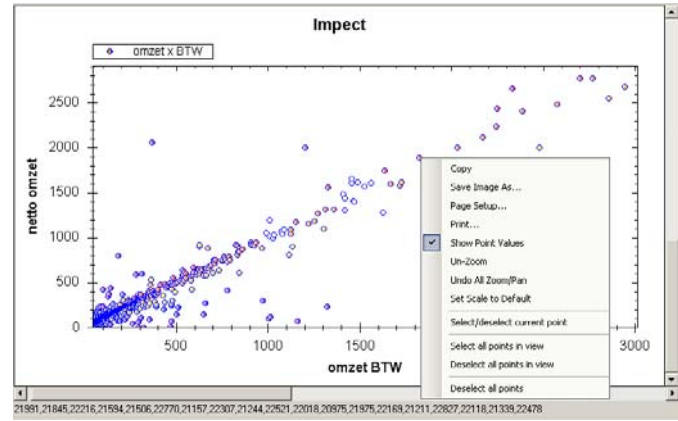


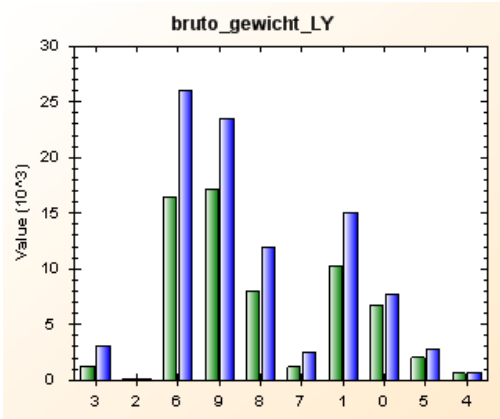



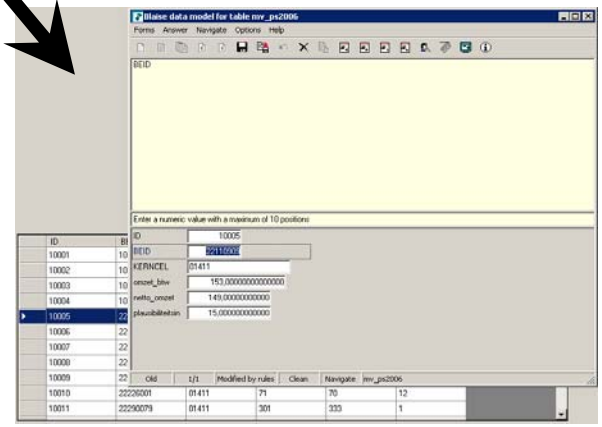



# Idee:

| ID | KERNCEL | btw_totaal | omzet_gemiddeld | plausibiliteitsindex |
|---|---|---|---|---|
| 12806 | 29100 | 59530 | 1175,92 | 1 |
| 12844 | 29300 | 65214 | 1758,9211 | 1 |
| 12852 | 29400 | 9206 | 1118,625 | 1 |
| 12883 | 29500 | 46303 | 1346,4839 | 1 |
| 12884 | 29600 | 364 | 361 | 0 |
| 12892 | 29700 | 11663 | 1029,875 | 1 |
| 12916 | 31100X | 43278 | 1795,1667 | 1 |
| 12948 | 31620X | 62377 | 1874,5625 | 1 |
| 12967 | 32100 | 54011 | 2176,8421 | 1 |
| 12973 | 32300X | 6401 | 1142 | 1 |
| 12975 | 33101 | 531 | 259,5 | |



# Todo/vragen:

Met echte data aan de slag bij V&V en Mesogaafmaken

Vervolg: waar moet de definitieve versie gemaakt worden? Binnen een project of bij Blaise ?

# Select data source(s)

- Select of data sources occurs through so called .boi file(s)
- Presently only one data source can be selected
- Version for 2001: multiple data sources
  - record set data
  - aggregated data: table meta (Cristal)

# Define groups

- Grouping variable types are:
  - enumeration
  - integer/ float
  - classification

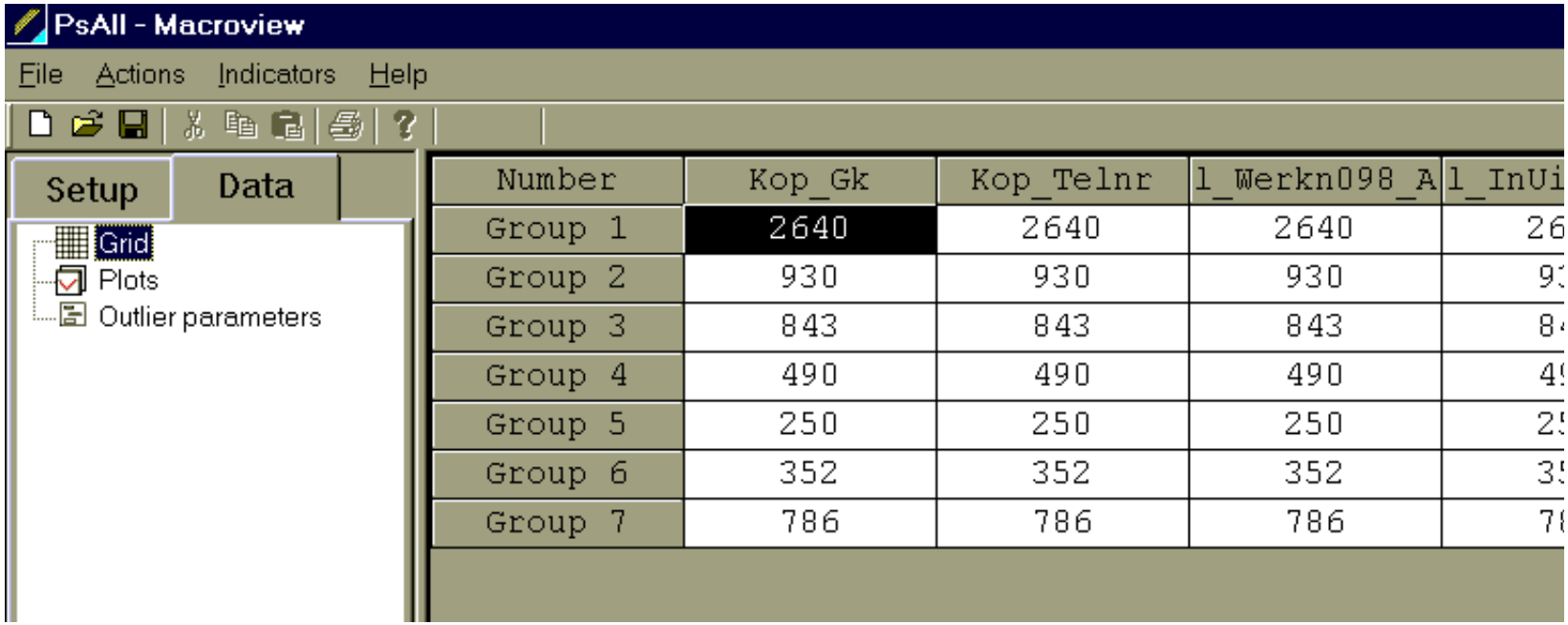
- Group the data, e.g.
  - sex (male, female)
  - turnover (0..100, 101..200, 201..300)

  leads to 6 groups

# Define/select variables to observe

- select variables, e.g.
  - TotalSalaries , NoPersons
- Define derived variables, if needed, e.g.
  - AvgSalary := TotalSalaries / NoPersons
- Generate table grid
- Choose indicators to analyse:
  - Mean, median, MAD, min, max, variance, standard deviation, alpha-trimmed mean, cell-filling, number of records.

# Show aggregates in grid:




| Number | Kop_Gk | Kop_Telnr | l_Werkn098_A | l_InUi |
|---|---|---|---|---|
| Group 1 | 2640 | 2640 | 2640 | 26 |
| Group 2 | 930 | 930 | 930 | 9 |
| Group 3 | 843 | 843 | 843 | 8 |
| Group 4 | 490 | 490 | 490 | 4 |
| Group 5 | 250 | 250 | 250 | 2 |
| Group 6 | 352 | 352 | 352 | 3 |
| Group 7 | 786 | 786 | 786 | 7 |

# Define/select variables to observe

## Alternative aggregate functions

CV(a) = std(a) /mean(a)

| Std | Mean(a) | a | CV(a) |
|---|---|---|---|
| Group 1 | 100.2 | 3.4 | 0.034 |
| Group 2 | 90.4 | 6.2 | 0.069 |

## Define checks on data (later: using checks from Blaise)

- abs(profit97- profit98)*2/(profit97+ profit98) < 0.1 --> percentage
- Filling(NoPersons) < 0.3 (non-response) --> signalling flag

| Mean | profit97 | profit98 | Δ profit | Filling < 0.3 |
|---|---|---|---|---|
| Group 1 | 234 | 250 | 2.3% | |
| Group 2 | 456 | 503 | 14.7% | |

# Macro level: Detection of suspicious cells

Identify suspicious data in cells (outliers), which have a potential impact on survey estimates using:

- Estimates of the distribution (mean,std,…)
- Manual outlier detection using boolean expression variables
- Automatic outlier detection using Kosinsky algorithm

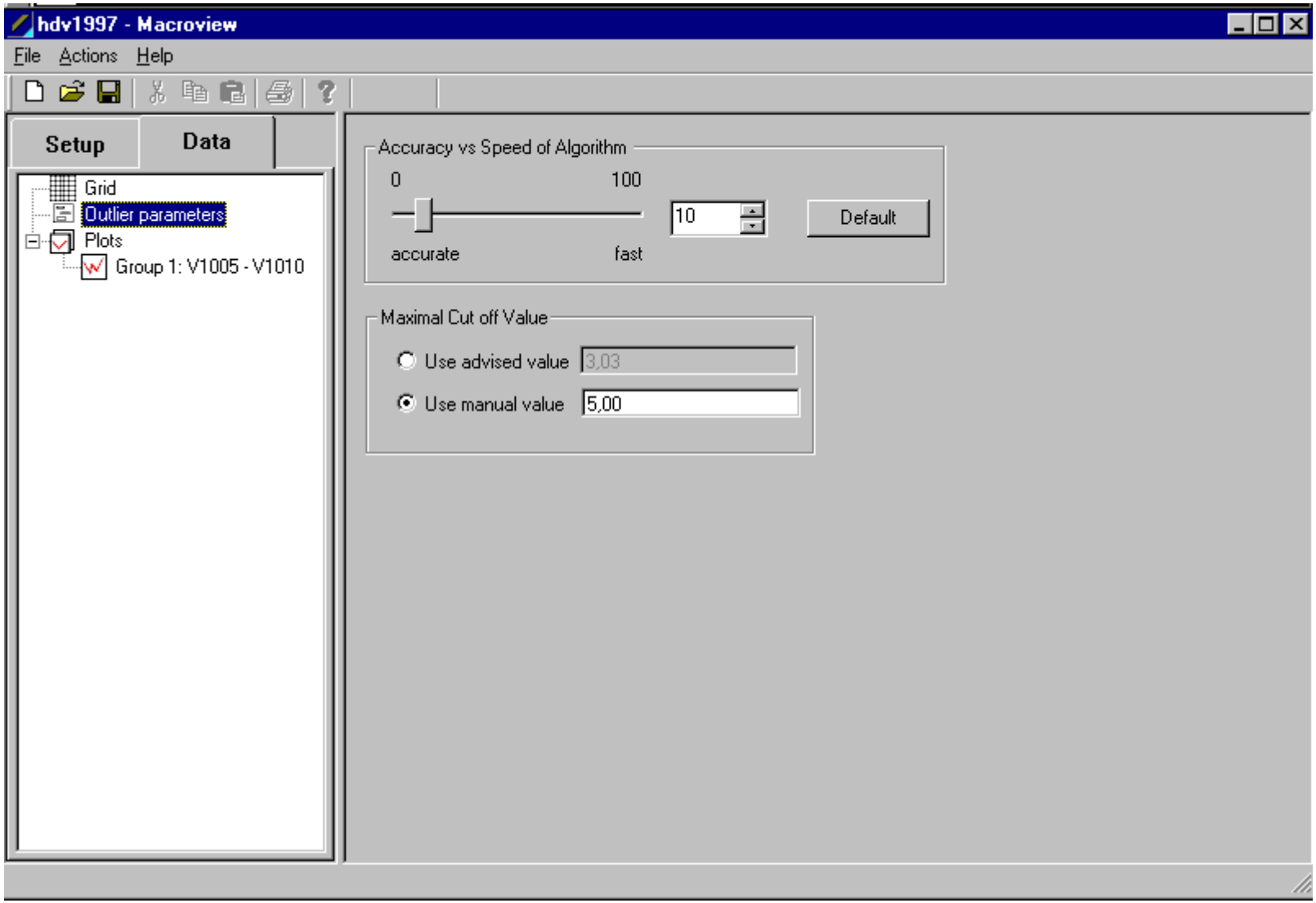
# Outlier detection: Kosinsky

- 1-D: use |x-median(x)|/MAD(x) distance to robustly detect outliers
- N-D: No median defined, use Mahalanobis distance instead:
- $M^2 = \Sigma\ (y_i - \mu)^T\ C^{-1}\ (y_i - \mu)$ ( $(y_i - \mu)/\sigma$ in 1D)

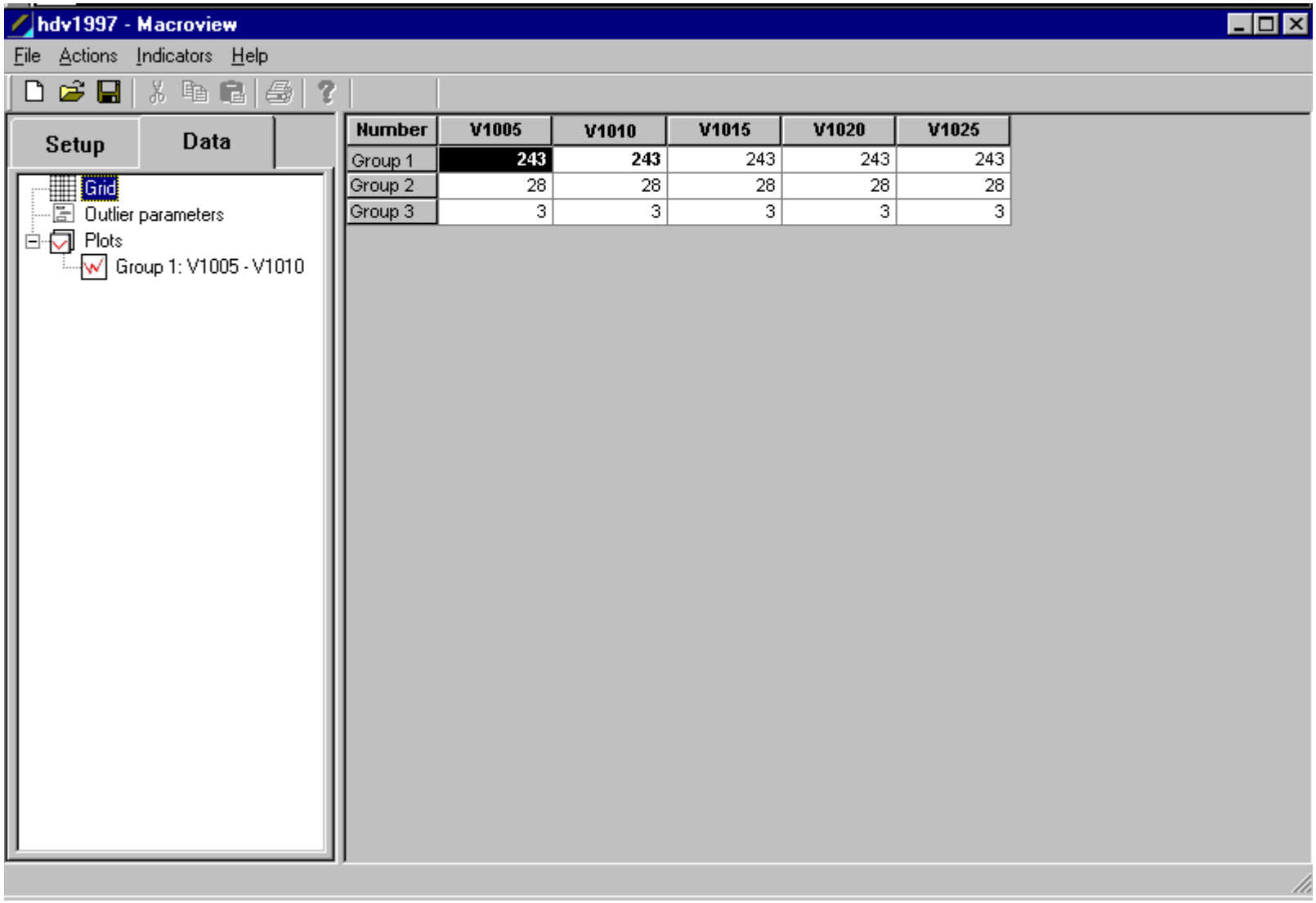
# Kosinski-algorithm

- 1. Start with $n_0$=0.1*n ‘good’ points
- 2. Good points ⇨ $\mu$
- 3. Obtain all Mahalanobis distances
- 4. Take the (1+***f***)*$n_i$ points with the smallest distances if distance < ***cutoff***
- 5. $n_{i+1}$= (1+f)*$n_i$
- 6. Repeat until no more points added.

# Parameters for outlier detection: f and cutoff



# Grid after outlier detection



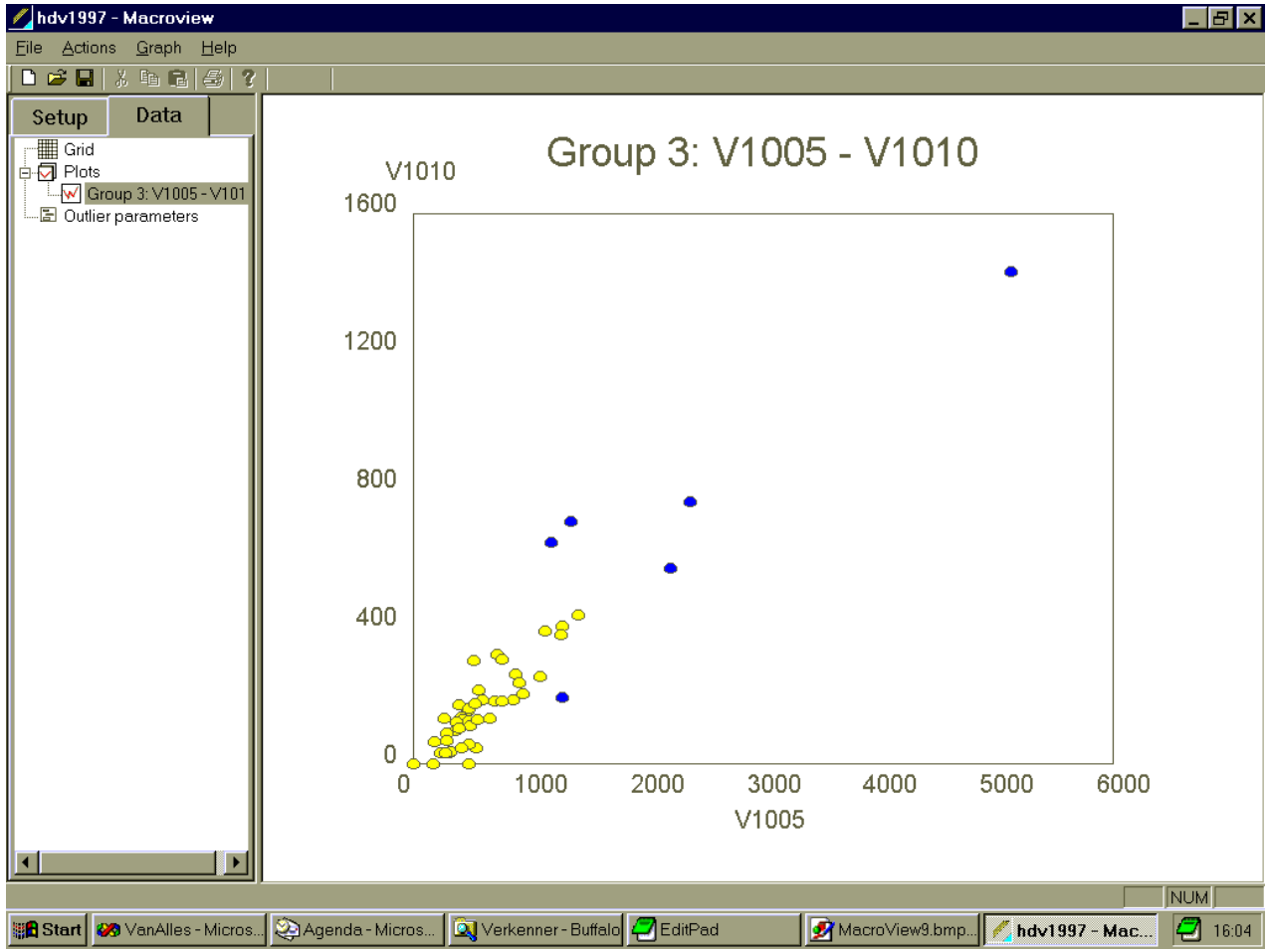

| Number | V1005 | V1010 | V1015 | V1020 | V1025 |
|---|---|---|---|---|---|
| Group 1 | 243 | 243 | 243 | 243 | 243 |
| Group 2 | 28 | 28 | 28 | 28 | 28 |
| Group 3 | 3 | 3 | 3 | 3 | 3 |

# Meso level:
# Detection of points in plots

Identify suspicious data in XY plots (outliers), which have a potential impact on survey estimates using:

- Manual outlier detection using boolean expression variables
- Automatic outlier detection using Kosinsky

# Scatter plot with outliers marked blue:



# Define/select variables to observe

## Alternative aggregate functions

CV(a) = std(a) /mean(a)

| Std | Mean(a) | a | CV(a) |
|---|---|---|---|
| Group 1 | 100.2 | 3.4 | 0.034 |
| Group 2 | 90.4 | 6.2 | 0.069 |

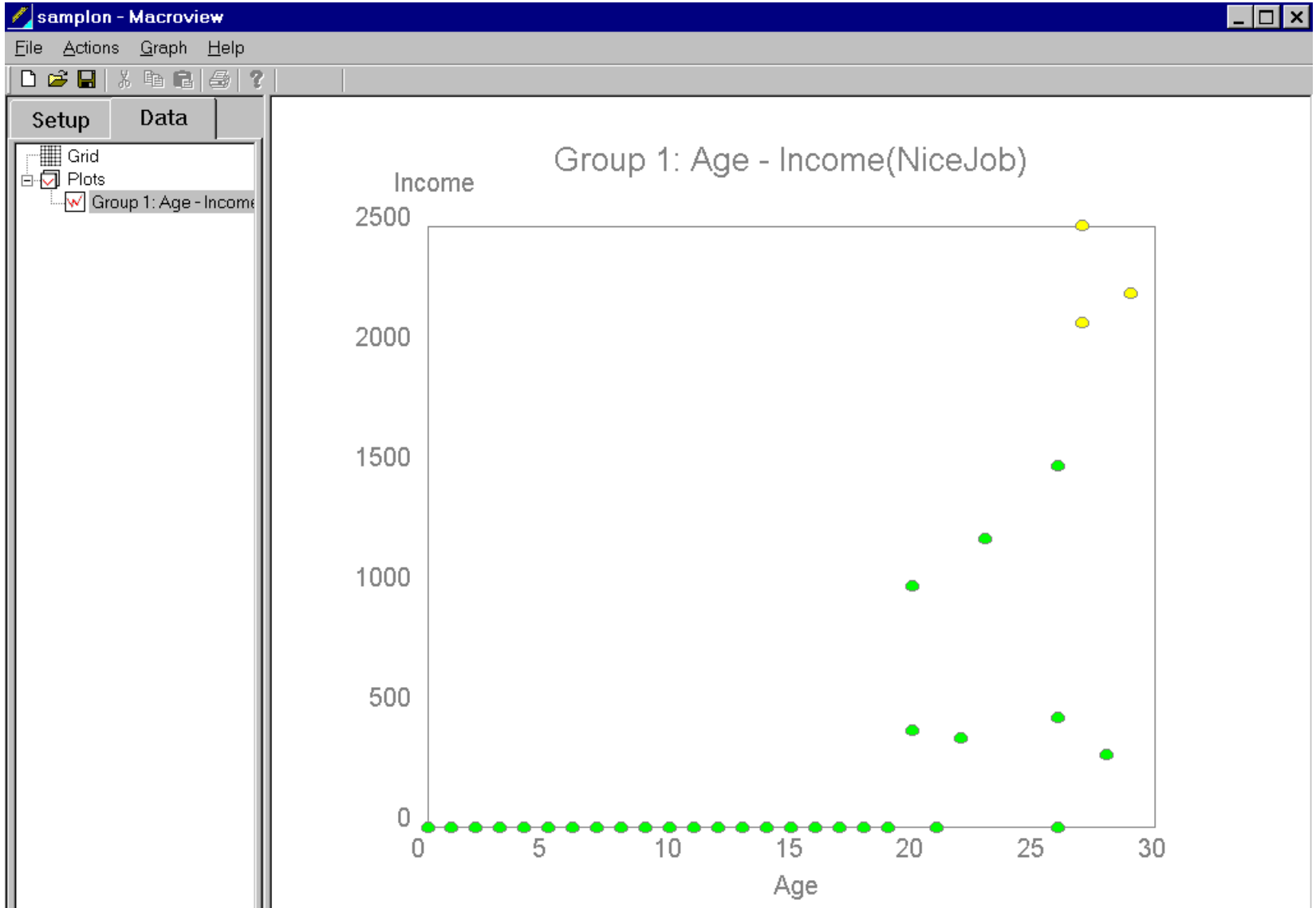
## Define checks on data (later: using checks from Blaise)

- abs(profit97- profit98)*2/(profit97+ profit98) < 0.1 --> percentage
- Filling(NoPersons) < 0.3 (non-response) --> signalling flag

| Mean | profit97 | profit98 | Δ profit | Filling < 0.3 |
|---|---|---|---|---|
| Group 1 | 234 | 250 | 2.3% | |
| Group 2 | 456 | 503 | 14.7% | |

# Scatter plot with condition

(Points not complying to the conditions are green)

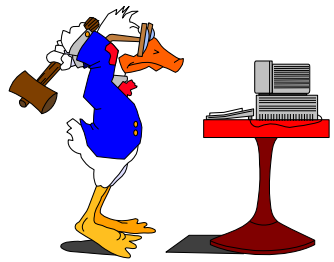


# Micro level: record editing

- Identify suspicious records either automatically or manually
- Sort suspicious records according to some outlier index, e.g. mahalanobis distance
- Edit records  manually; worst cases first

# Micro editing

- Directly: click on suspicious point in xy-plot and edit it
  - now:
    - copy OLEDB --> Blaise
    - call RunDEP(...) in BlTools.dll
    - copy Blaise --> OLEDB
  - later:
    - use DEC with existing IBlaiseDatabase reference

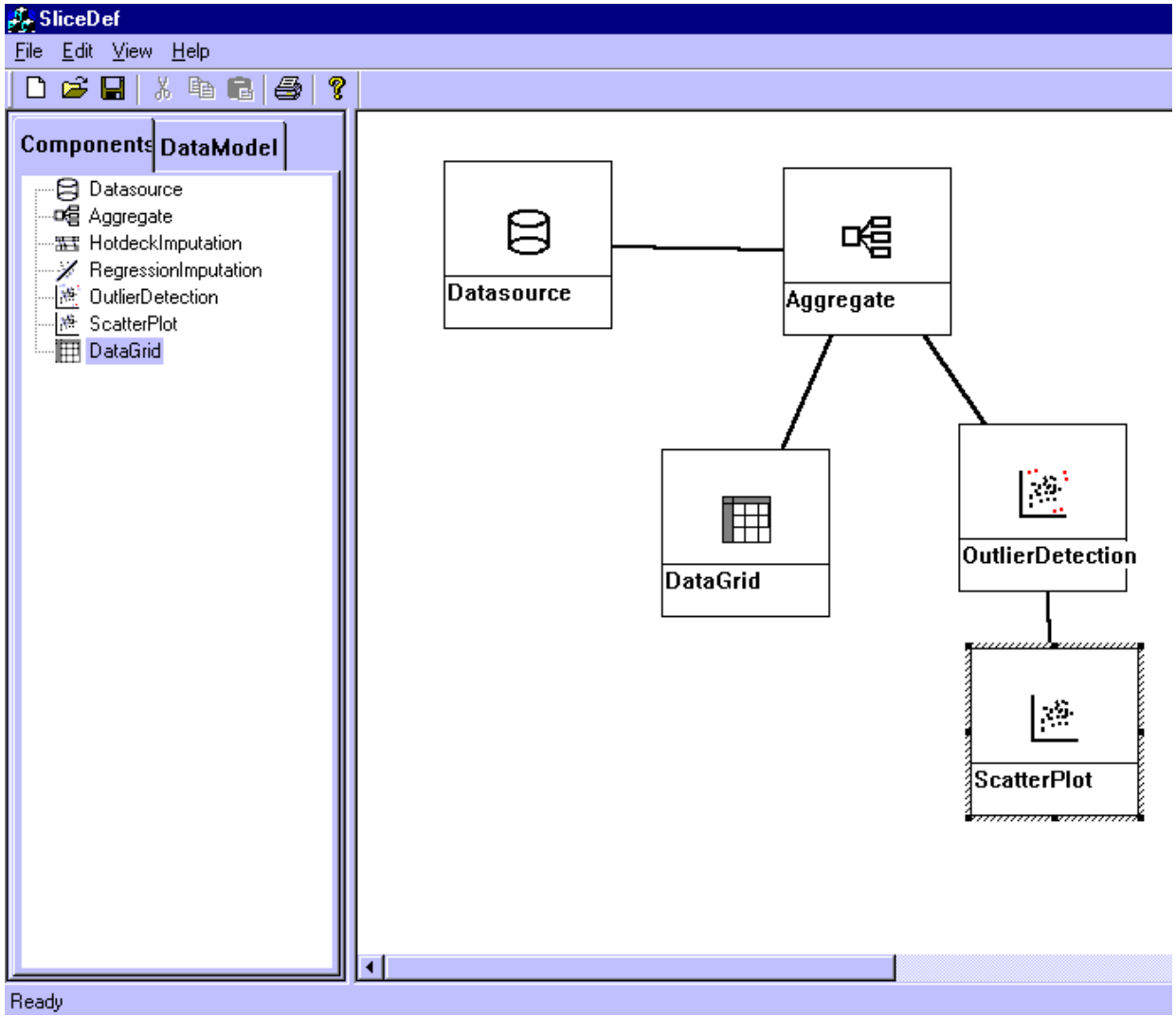


# MacroView and Slice:

## The building blocks of macroview

- Under the hood the macroview application consists of several modules; these modules can and will be used as building blocks in SLICE.

# Slice: a decomposed macroview as an example



# Slice

- Overall framework for editing/imputation and display modules, based on COM technology

- *Data source independency*: slice record sets
- *Small building blocks* with just 1 function: (COM) idea, LEGO bricks: flexibility

- Data exchange between blocks based on SLICE records, data description uses Blaise meta
- Usable in many environments because of COM: VB, VC++, Delphi, Manipula,...

- Data from / to: Blaise, OLEDB databases, ...

- Currently available: CherryPi and Imputation

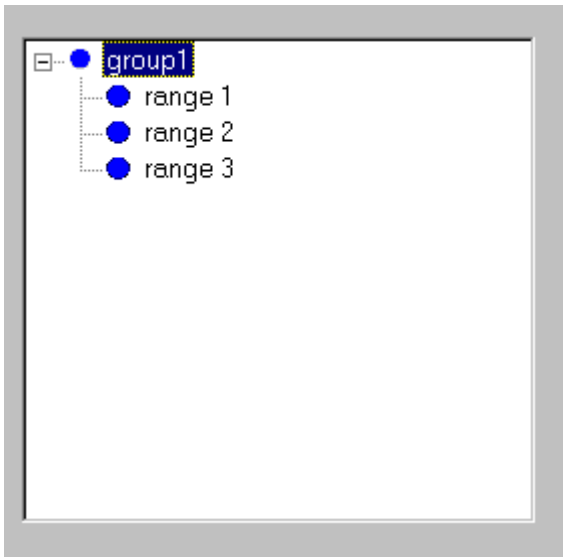
# Future of macroview (1):

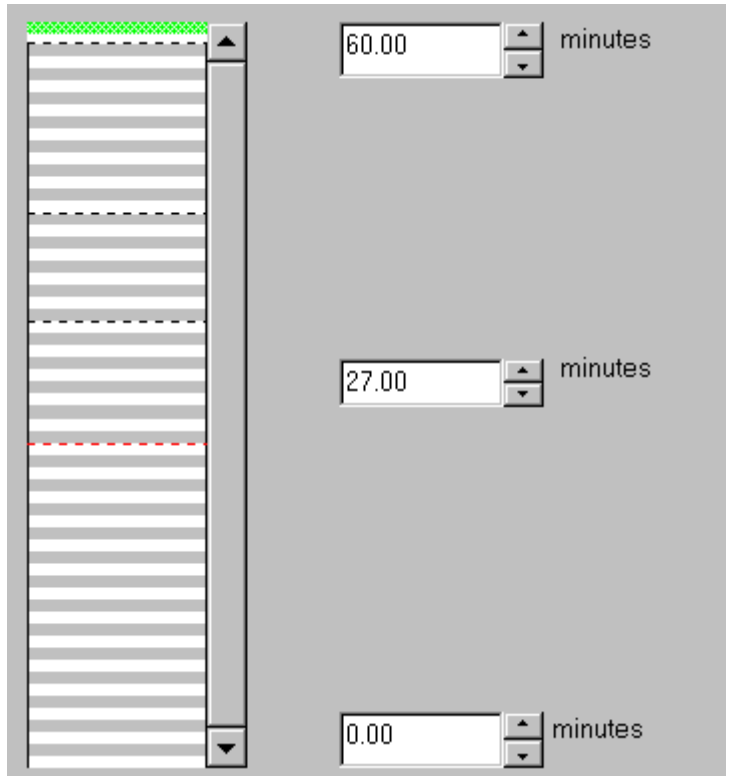
- new aggregate functions for demography changes:
  - import/export(data source1, data source 2)

| Mean | Year98 | Year99 | Import(..) |
|---|---|---|---|
| Group 1 | 100.2 | 140.2 | 4 |
| Group 2 | 90.4 | 88.3 | 0 |

# Future of macroview (2):

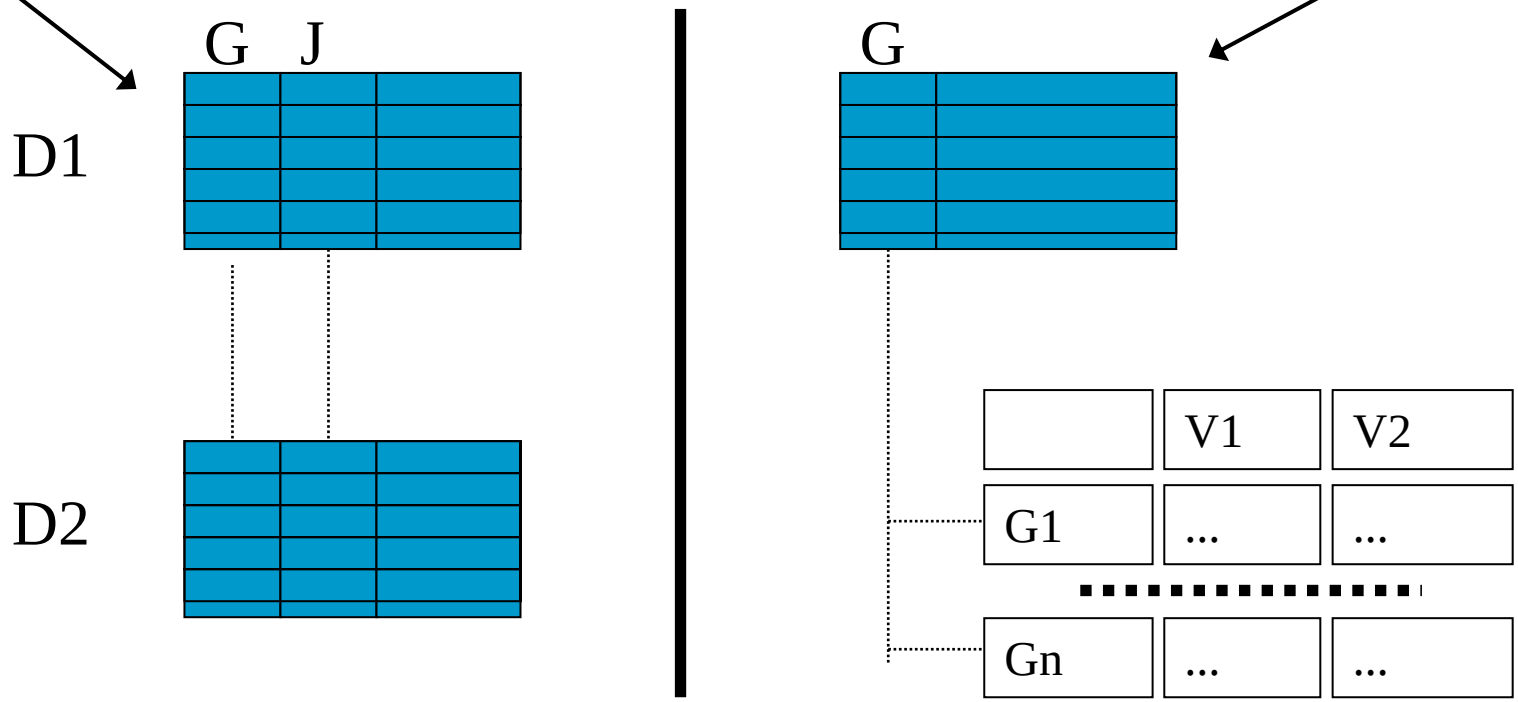
- one more intuitive dialog for the definition van integer/float (time/date) variable types
- Manipulation of defined ranges



# Future of macroview (3):

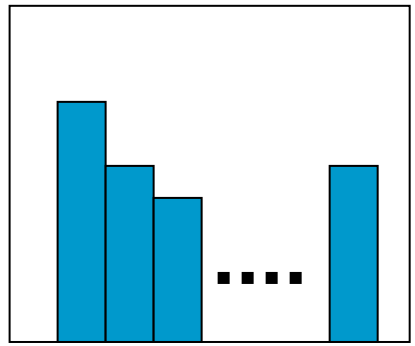
- More data sources :
  - all data sources (Di) must contain the grouping fields (G)
  - other data sources with detail records: join fields J) must be supplied
  - in case of t-1, t-2, ...: often just aggregates; table meta: Cristal



# Future of macroview (4):

- plots of data at aggregated level, e.g. histogram of average profit for some or all groups

| Mean | Year98 | Year99 | Import(..) |
|---|---|---|---|
| Group 1 | 100.2 | 140.2 | 4 |
| ▪▪▪▪ | ▪▪▪▪ | ▪▪▪▪ | ▪▪▪▪ |
| Group n | 90.4 | 88.3 | 0 |



- more than one aggregate function active in grid

| Mean | Std | | |
|---|---|---|---|
| | Year98 | Year98 | Year99 |
| Group 1 | 10.2 | 100.2 | 140.2 |
| ▪▪▪▪ | ▪▪▪▪ | ▪▪▪▪ | ▪▪▪▪ |
| Group n | 9.4 | 90.4 | 88.3 |

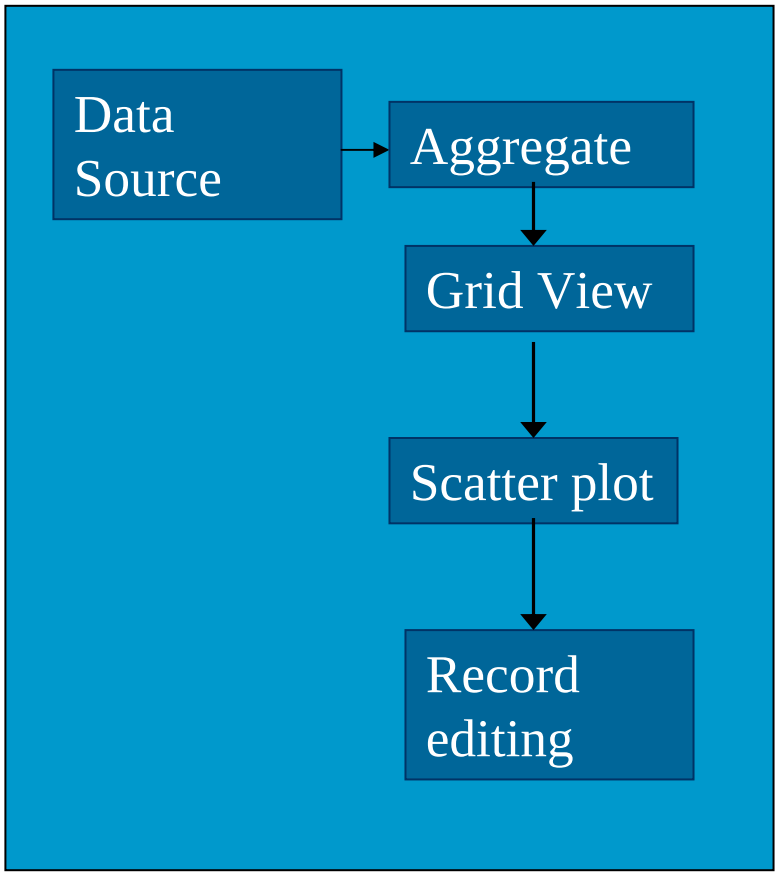
# Future of macroview (5):

- allow classification variable as a grouping variable (SBI,..)
- box-whisker, …, plots
- Integration with Blaise:
  - Macroview as a whole part of the Blaise shell
  - Aggregation and robust outlier detection (Kosinsky) as slice modules; available via Manipula

# Slice in more detail

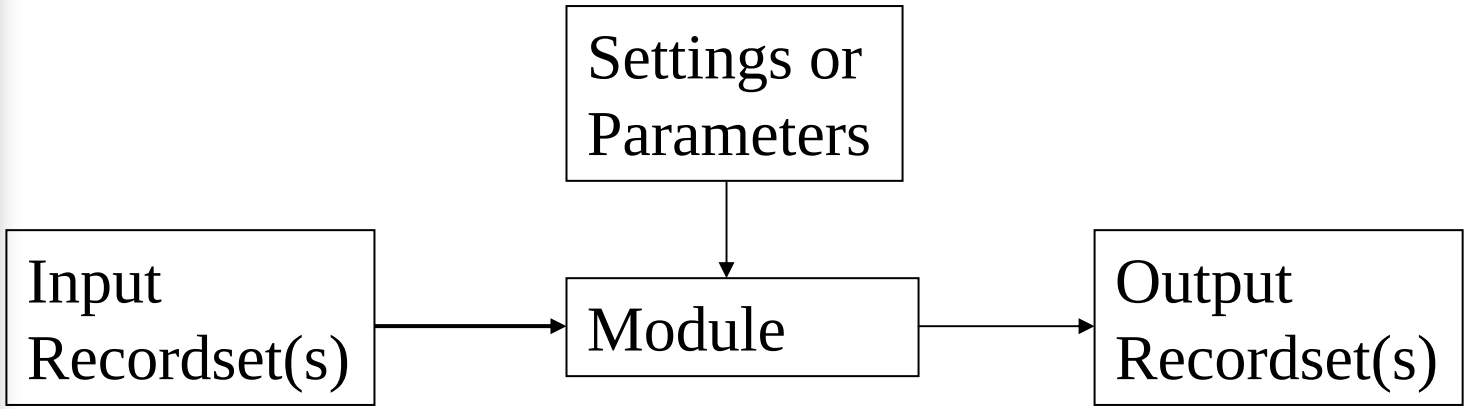
# MacroView and Slice:

## The building blocks of macroview



# Slice modules:

- Recieves and sends so-called Slice records
- Has no knowledge of the origin or destination of the records
- Has (in principle) one function
- Most modules will be non-interactive



# Define/select variables to observe

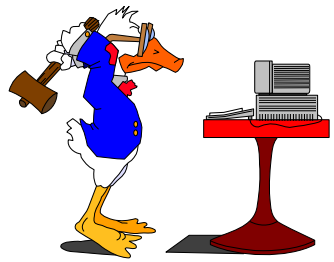
- Derived variables can also be conditions, e.g.
  - Filling(NoPersons) > 0.3  (non-response) --> signalling flag
  - abs(profit97- profit98)*2/(profit97+ profit98) < 0.1 --> percentage
  - alternative aggregate function: std(a)/mean(a) or just mean(a)
  - .

# Storage

- Storage is done using the IRegister component in BlRegA.dll
- Undo/Redo ??

# Blaise OLEDB Information (.boi) file

- Reference to Blaise meta file (.bmi) file
- Reference to an OLEDB provider and table, e.g.
  - provider = C:\MyDatabase.mdb (Access) , table = Customer
- Mapping between OLEDB fields and Blaise fields:
  - Customer.Name (String) <--> Name (STRING[20])
  - Customer.Type (Integer) <--> Type (AgeClass : (Rich (1), Medium (2), Poor (3) ) )



# SiAggregate

Datasource
(.boi)

Income, Age,...

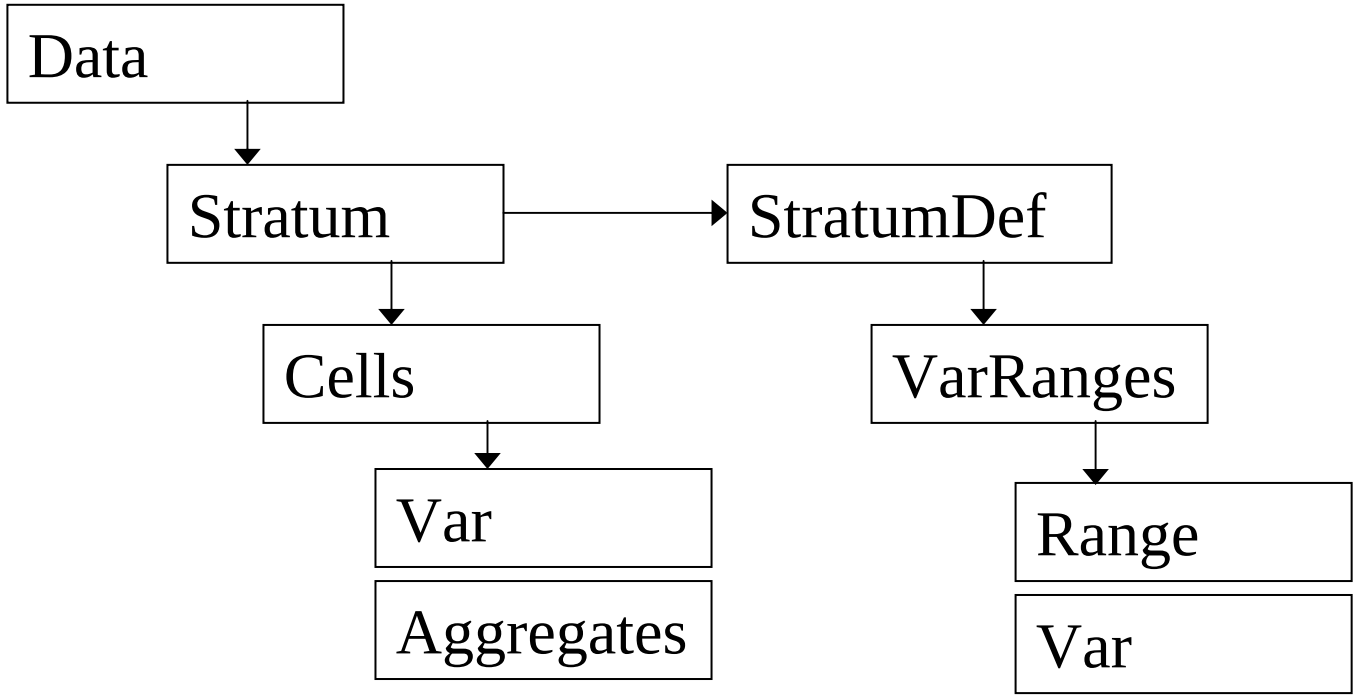
Grouping:
- GetQuery
-

Integer Grouping

Enum Grouping

Classification Grouping

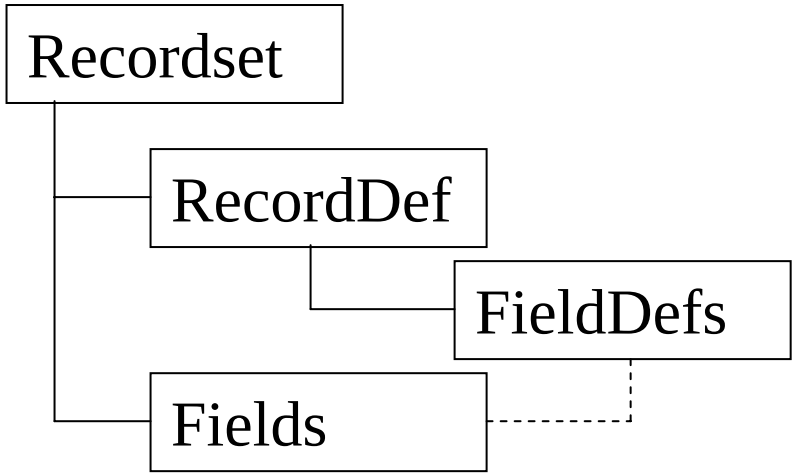
# SiAggregate



# Slice records:

- Database-independent and simple recordset
- Records can come from/go to any OLEDB or Blaise source
- Will handle the distribution to multiple modules
- Will take care of buffering of records, if necessary

*The internal structure of a slice record*



# Using the slice modules:

- Both the modules and the Slice recordset can be used from any programming language that supports the use of COM
- Modules under construction:
  - CherryPi: automatic error detection
  - SiImputeXXX: several imputation modules

# Near future:

- Com object for the Slice records
- Modules: CherryPi and SiRegressionImputation

# Future:

- More modules (Aggregation, more imputation methods, ...)
- Visual prototyping tool

# Visual inspection methods:

- Show Mahalanobis distances for selected recordset
- Show outliers (based on Kosinski algorithm) in datagrid or in scatterplot

- Mark outliers by hand in a scatterplot
- Clicking outlier point(s) allows editing of the record (using the Blaise DEP)

- Visual presentation
  - Flagged aggregates
  - List of sorted records
  - Scatter plot
  - Histogram

# An example: the CherriPi module:

- Input: Recordset and a set of Edit Rules
- Output: Copy of input recordsset with some fields marked ‘erroneous’
- Based on edit rules, e.g.

$$\text{profit} + \text{costs} = \text{turnover}$$

$$\text{costs} - 0.5 * \text{turnover} > 0$$

- Automatic localization of erroneous records
- Localize faulty item(s) in erroneous record: e.g. algorithm as implemented by Cherry Pi (based on the Fellegi-Holt paradigm)

# Multiple datasources (e.g.: t, t-1)

| | Situation 1 | Situation 2 | Situation 3 |
|---|---|---|---|
| Coupled | fully | partially | not |
| Grouping | yes | yes | yes |
| Aggregated | no | no | yes |
| | | | |
| Remark | | | Cristal |

# Cristal:

- *Dimensions* = Variable
- *Observation Item* = Variable + aggregation method
- *Hierarchy* = Variable +range :
  - *Classification item* = range
- *Data point* = 1 cell in grid
  = classification item x observation item

| | | | |
|---|---|---|---|
| cccc | cccc | cccc | cccc |
| cccc | cccc | cccc | cccc |
| cccc | cccc | cccc | cccc |