

st.module

`st.module`

descriptive statistics of finite populations taken in their entirety, mostly using textbook formulas

Requires `str.lg` `csv.lg` `df.lg` and `rs.lg`

st.data

`st.data`

```
make "a (df.create [num char num num] (list se "id iseq 1 90 se "sex (rs.rep [female male] 45) se "year (rs.pick iseq 1970 1972 90) se "age rs.rand 1 100 90) )
```

```
make "b (df.create [num char num num] (list se "id iseq 1 90 se "sex (rs.rep [female male missing] 30) se "year (rs.pick iseq 1970 1972 90) (se "age rs.rand 1 100 88 "missing "missing)) )
```

```
make "patients (df.create [char char] [[PATIENT_ID 1001 4004 3003 2002 5005] [LASTNAME Hopper Wirth Kemeny Gosling Kurtz]] )
```

```
make "visits (df.create [char char num] [[PATIENT_ID 2002 1001 4004 2002 1001 3003 4004 1001] [VISIT_DATE 2020-09-10 2020-09-17 2020-09-24 2020-10-08 missing 2020-11-12 2020-11-05 2020-11-19] [SCORE 6.8 5.5 8.4 missing 6.6 missing 7.0 5.3]] )
```

```
make "f (df.create [num] [[price 150 250 688 695 795 795 895 895 895 1099 1166 1333 1499 1693 1699 1775 1895]] )
```

st.mergesort

`st.mergesort :list`

Returns a sorted list of numbers

Parameters

list: numerical list, missing values not allowed

Examples

```
show st.mergesort [1 4 2.5 0.5 3]
```

st.freq

`st.freq :df :names [newvar "N"]`

Computes multi-way frequency tables

Parameters

df: dataframe

names: list of grouping variables, missing values allowed

newvar: new variable's name, default "N"

Returns

A dataframe

Examples

```
df.print st.freq :a [sex]
```

```
df.print st.freq :a [sex year]
```

```
df.print st.freq :b [sex]
```

```
df.print st.freq :b [sex year]
```

st.proportions

```
st.proportions :df :names
```

Computes multi-way proportion tables, using up to three variables. New variables' names: "N" and "N1"

Parameters

df: dataframe

names: list of grouping variables, missing values allowed.

Returns

A dataframe

Examples

```
df.print st.proportions :a [sex]
```

```
df.print st.proportions :a [sex year]
```

```
df.print st.proportions :b [sex]
```

```
df.print st.proportions :b [sex year]
```

st.summary

```
st.summary :df :var [names [total]] [types [num]]
```

Splits the dataframe into non-overlapping groups, computes a summary (n sum mean variance sd skew kurt) of a numeric variable for each group omitting missing values and combines the results in a new dataframe. See Xiangrui Meng, Simpler Online Updates for Arbitrary-Order Central Moments, <https://arxiv.org/abs/1510.04923> and John D Cook, Computing skewness and kurtosis in one pass, https://www.johndcook.com/skewness_kurtosis.html

Parameters

df: dataframe

var: numerical variable name, missing values omitted

names: list of grouping variables, missing values allowed, default "total" (no grouping variables)

types: list of grouping variables' types, default "num"

Returns

A dataframe

Notes

n = number of non missing "var" values

sum = sum of non missing "var" values

mean = mean of non missing "var" values

variance = variance of non missing "var" values

sd = standard deviation of non missing "var" values

skew = skewness of non missing "var" values

kurt = kurtosis of non missing "var" values

Examples

```
df.print st.summary :a "age"
```

```
df.print (st.summary :a "age [sex] [char])
```

```
df.print st.summary :b "age"
```

```
df.print (st.summary :b "age [sex] [char])
```

```
df.print (st.summary :a "age [sex year] [char num])
```

```
df.print (st.summary :b "age [sex year] [char num])
```

st.summary.n

```
st.summary.n :df
```

Extracts number of observation "n" from a summary dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

```
make "df st.summary :a "age
```

```
df.print (st.summary.n :df)
```

```
make "df (st.summary :b "age [sex] [char])
```

```
df.print (st.summary.n :df)
```

st.summary.sum

```
st.summary.sum :df
```

Extracts "sum" from a summary dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

```
make "df st.summary :a "age
```

```
df.print (st.summary.sum :df)
```

```
make "df (st.summary :b "age [sex] [char])
```

```
df.print (st.summary.sum :df)
```

st.summary.mean

```
st.summary.mean :df
```

Extracts "mean" from a summary dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

```
make "df st.summary :a "age
```

```
df.print (st.summary.mean :df)
```

```
make "df (st.summary :b "age [sex] [char])
```

```
df.print (st.summary.mean :df)
```

st.summary.variance

```
st.summary.variance :df
```

Extracts "variance" from a summary dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

```
make "df st.summary :a "age
```

```
df.print (st.summary.variance :df)
```

```
make "df (st.summary :b "age [sex] [char])
```

```
df.print (st.summary.variance :df)
```

st.summary.sd

```
st.summary.sd :df
```

Extracts standard deviation "sd" from a summary dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

```
make "df st.summary :a "age
df.print (st.summary.sd :df)

make "df (st.summary :b "age [sex] [char])
df.print (st.summary.sd :df)
```

st.summary.skew

```
st.summary.skew :df
```

Extracts skewness "skew" from a summary dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

```
make "df st.summary :a "age
df.print (st.summary.skew :df)

make "df (st.summary :b "age [sex] [char])
df.print (st.summary.skew :df)
```

st.summary.kurt

```
st.summary.kurt :df
```

Extracts kurtosis "kurt" from a summary dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

```
make "df st.summary :a "age
```

```
df.print (st.summary.kurt :df)
```

```
make "df (st.summary :b "age [sex] [char])
```

```
df.print (st.summary.kurt :df)
```

st.min.max

```
st.min.max :df :var [names [total]] [types [num]] [ISO "false"]
```

Splits the dataframe into non-overlapping groups, computes the minimum and maximum of a numeric variable, that may be a date in ISO format "yyyy-mm-dd", for each group omitting missing values and combines the results in a new dataframe (either (total n min max) or ("grouping variables" n min max)).

Parameters

df: dataframe

var: numerical variable name, missing values omitted

names: list of grouping variables, missing values allowed, default "total" (no grouping variables)

types: list of grouping variables' types, default "num"

ISO: "var" is an ISO date? default "false"

Returns

A dataframe

Examples

```
df.print st.min.max :a "age
```

```
df.print (st.min.max :a "age [sex] [char])
```

```
df.print st.min.max :b "age
```

```
df.print (st.min.max :b "age [sex] [char])
```

```
df.print (st.min.max :a "age [sex year] [char num])
```

```
df.print (st.min.max :b "age [sex year] [char num])
```

```
df.print (st.min.max :visits "VISIT_DATE [total] [num] "true)
```

```
df.print (st.min.max :visits "VISIT_DATE [PATIENT_ID] [char] "true)
```

st.min

```
st.min :df
```

Extracts "min" from a min.max dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

```
make "df st.min.max :a "age
```

```
df.print (st.min :df)
```

```
make "df (st.min.max :a "age [sex] [char])
```

```
df.print (st.min :df)
```

```
make "df st.min.max :b "age
```

```
df.print (st.min :df)
```

```
make "df (st.min.max :b "age [sex] [char])
```

```
df.print (st.min :df)
```

st.max

```
st.max :df
```

Extracts "max" from a min.max dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

```
make "df st.min.max :a "age
```

```
df.print (st.max :df)
```

```
make "df (st.min.max :a "age [sex] [char])
```

```
df.print (st.max :df)
```

```
make "df st.min.max :b "age
```

```
df.print (st.max :df)
```

```
make "df (st.min.max :b "age [sex] [char])
```

```
df.print (st.max :df)
```

st.fivenum

```
st.fivenum :df :var [names [total]] [types [num]]
```

Splits the dataframe into non-overlapping groups, computes Tukey's "fivenum" summary of a numeric variable for each group omitting missing values and combines the results in a new dataframe (either (total n min Q1 Q2 Q3 max) or ("grouping variables" n min Q1 Q2 Q3 max)). See Tukey, John W (1977). Exploratory Data Analysis. Addison-Wesley.

Parameters

df: dataframe

var: numerical variable name, missing values omitted

names: list of grouping variables, missing values allowed, default "total" (no grouping variables)

types: list of grouping variables' types, default "num"

Returns

A dataframe

Notes

n = number of non missing "var" values

min = minimum value

Q1 = first quartile

Q2 = second quartile (median)

Q3 = third quartile

max = maximum value

Examples

```
df.print st.fivenum :a "age
```

```
df.print (st.fivenum :a "age [sex] [char])
```

```
df.print st.fivenum :b "age
```

```
df.print (st.fivenum :b "age [sex] [char])
```

```
df.print st.fivenum :f "price
```

```
df.print (st.fivenum :a "age [sex year] [char num])
```

```
df.print (st.fivenum :b "age [sex year] [char num])
```

st.fivenum.Q1

```
st.fivenum.Q1 :df
```

Extracts "Q1" from a fivenum dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

```
make "df st.fivenum :a "age
```

```
df.print (st.fivenum.Q1 :df)
```

```
make "df (st.fivenum :a "age [sex] [char])
```

```
df.print (st.fivenum.Q1 :df)
```

st.fivenum.Q2

```
st.fivenum.Q2 :df
```

Extracts "Q2" (the median) from a fivenum dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

Examples

```
make "df st.fivenum :a "age
```

```
df.print (st.fivenum.Q2 :df)
```

```
make "df (st.fivenum :a "age [sex] [char])
```

```
df.print (st.fivenum.Q2 :df)
```

st.fivenum.Q3

```
st.fivenum.Q3 :df
```

Extracts "Q3" (the median) from a fivenum dataframe

Parameters

df: dataframe

Returns

A dataframe

Examples

Examples

```
make "df st.fivenum :a "age
```

```
df.print (st.fivenum.Q3 :df)
```

```
make "df (st.fivenum :a "age [sex] [char])
```

```
df.print (st.fivenum.Q3 :df)
```

st.fivenum.IQR

```
st.fivenum.IQR :df
```

Computes "IQR" (interquartile range) from a fivenum dataframe and lower and upper bounds for moderate and extreme outliers. See Tukey, John W (1977). Exploratory Data Analysis. Addison-Wesley.

Parameters

df: dataframe

Returns

A dataframe

Notes

IQR = interquartile range, $Q3 - Q1$

lbound = lower bound, $Q1 - 1.5 \cdot IQR$

elbound = extreme lower bound, $Q1 - 3 \cdot IQR$

ubound = upper bound, $Q3 + 1.5 \cdot IQR$

eubound = extreme upper bound, $Q3 + 3 \cdot IQR$

Examples

```
make "df st.fivenum :a "age
```

```
df.print (st.fivenum.IQR :df)
```

```
make "df (st.fivenum :a "age [sex] [char])
```

```
df.print (st.fivenum.IQR :df)
```