VisiData Terminal data spreadsheet

START HERE

`vd <file>`
Open CSV, JSON, Excel

U
Undo last change

Arrows / h › j › k › l
Move around

q
Close sheet, quit at last

Shift + F
Frequency table of column

[/]
Sort column up / down

|
Select rows by regex

Ctrl + S
Save sheet to a file

Source file vd data.csv → `load` → **Sheet** rows x columns → `Shift+F, Shift+W, "` → **Derived sheet** freq, pivot, subset → `Ctrl+S` → **Saved file** csv, tsv, json...

CONCEPTS

VisiData is a spreadsheet for the terminal: every view (columns, frequencies, pivots) is a new sheet on a stack.

Commands apply to the current column or row. A g prefix usually means all (gs, gd), z a variant.

Typing a key inside a prompt edits text; Enter accepts, Ctrl+C or Esc cancels.

NAVIGATION

Arrows / h › j › k › l
Move cell by cell

g › h / g › j / g › k / g › l
Jump to left, bottom, top, right edge

PgUp / PgDn Scroll a page

/ / ?
Regex search column, down / up

n / Shift + N
Next / previous match

g › /
Regex search all columns

c Go to column by name

z › r Go to row number

< / >
Previous / next differing value

NAVIGATION (cont.)

{ / }
Previous / next selected row

z › /
Search with a Python expression
Column names work as variables.

ROWS AND SELECTION

s / t / u
Select, toggle, unselect row

g › s / g › t / g › u
Same for all rows

| / \
Select / unselect by regex in column

g › |
Select by regex in any column

z › |
Select by Python expression

,
Select rows equal to this cell

"
Open sheet of selected rows only
Edits flow back to the source sheet; z " makes an independent copy.

g › d Delete selected rows

d Delete current row

a Add an empty row

ROWS AND SELECTION (cont.)

y / p Copy row, paste after

[/]
Sort by column up / down

z › [/ z ›]
Add column to sort criteria

g › [/ g ›]
Sort by all key columns

COLUMNS

/ %
Type column int / float
Needed before numeric sort or sums.

@ / ~
Type column date / string

- Hide column

g › v Unhide all columns

! Make key column (pins left)

H / L
Move column left / right

^ Rename column

=
Add column from Python expression
*Example: price*qty; name=expr names the column.*

: Split column by regex

*
Regex replace into new column

COLUMNS (cont.)

(/)
Expand / contract nested column
For JSON: g (expands every visible column.

Shift + =
Add aggregator (the + key) to column

EDIT AND UNDO

e Edit current cell

g › e
Set column for selected rows

U / R
Undo / redo last change

Ctrl + R
Reload sheet from source

Ctrl + C
Cancel running command

SUMMARIES AND PIVOTS

Shift + F
Frequency table of column
Enter on a row opens the matching source rows.

Shift + I
Describe sheet: stats per column

Shift + W
Pivot table on current column
Mark group-by columns with ! first.

SUMMARIES AND PIVOTS (cont.)

Enter
Open source rows of group or cell

SHEETS, FILES, HELP

q Close current sheet

g › q Quit everything

Ctrl + S
Save sheet; name sets the format
Extension picks the format: .csv, .tsv, .json, .xlsx.

o Open another file

Shift + S
Sheets stack: jump between sheets

g › Shift + S
All sheets of session

Ctrl + ^ Previous sheet

Shift + C
Columns sheet: edit types, widths

&
Join sheets on key columns

Shift + D
Command log of this session

Ctrl + H Help menu

z › Ctrl + H
List all commands and keys

COMMAND LINE

`vd <file> ...`
Open files, last on top

`<cmd> | vd`
Read from stdin

`vd -f <type> <file>`
Force a loader

`vd -b <in> -o <out>`
Convert in batch mode

`vd -p <log.vdj>`
Replay a command log

`vd -d ';' <file>`
Set field delimiter

`vd --skip <n> <file>`
Skip leading lines

`vd .`
Browse a directory

RECIPES

Convert without the UI

```
vd -b data.csv -o data.json
vd -b data.json -o data.xlsx
```

Count values of a column

```
vd data.csv
# move to column, Shift+F
# Enter on a row lists its rows
# Ctrl+S saves the table
```

Explore command output

```
ps aux | vd
# Shift+F on USER, # on a
number
# ] sorts descending
```

Record and replay cleanup

```
# Ctrl+D saves clean.vdj
vd -b -p clean.vdj new.csv -o
out.csv
```