

**14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and
Knowledge Management (IC3K 2022)**

24 - 26 October, 2022

Tutorial

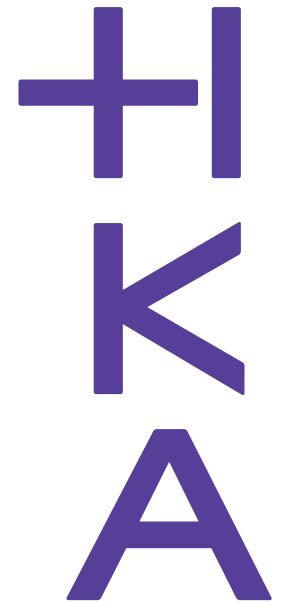
Knowledge Discovery and Information Retrieval using the Shell

Andreas Schmidt

**Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Germany**

**Department of Informatics and
Business Information Systems
University of Applied Sciences Karlsruhe
Germany**

Resources available

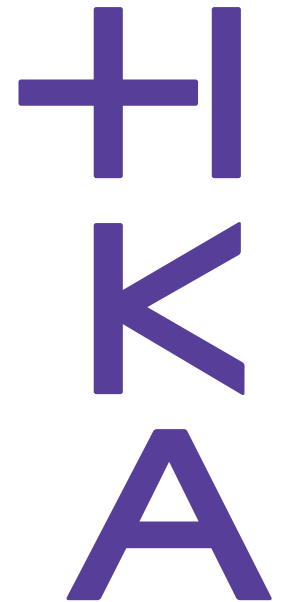


<https://www.smiffy.de/KDIR-2022/>¹

- Slideset
- 3 Exercises
- Command refcard
- Many examples
- Example datasets
- Further resources

1. all materials copyright 2017, 2018, 2019,2020, 2021, 2022 by andreas schmidt

Outlook



- Introduction
- Functionality Overview
- Filter & Pipes Architecture
- Command Overview Part I
- Exercise I: Start solving a criminal case using the shell
- Command overview Part II
- Exercise II: Solve the criminal case from Exercise I
- sed & awk
- Summary & Outlook

Coreutils

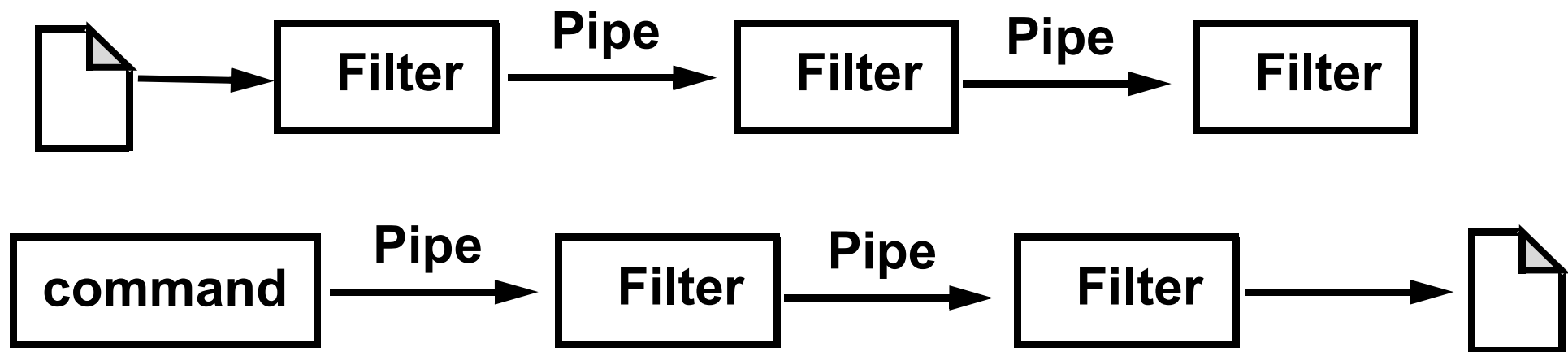


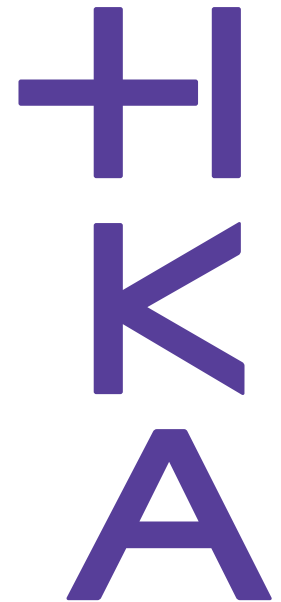
- The GNU Core Utilities are the basic file, shell and text manipulation utilities of the GNU operating system. These are the core utilities which are expected to exist on every operating system. [1]
- These utilities are typically used in a Filter & Pipes Architectural pattern, where the output of the first utility program acts as input for the next utility program (and so on ...)

[1] <https://www.gnu.org/software/coreutils/>

Filter and Pipes Architecture

- Architectural Pattern: Filter and Pipes (Douglas McIlroy, 1973)
- Data exchange between processes
- Loose coupling
- POSIX Standard
- Filter represent data-sources and data-sinks





Why Should I use these Tools (Coreutils)?

With R and Python, there exist great tools that can perform the same job (and much more)

- It's already on your computer and nothing needs to be installed¹
- You don't need to learn a programming language
- You don't need an editor, compiler or interpreter
- Low main memory footprint
- You got first results after 20 sec.
- Intuitive iterative development cycle (add filter by filter) ... like lego blocks
- It makes fun !!!!

1. If you have a Linux or Mac-computer. Windows users have to install *cygwin* or the *Windows Subsystem for Linux (WSL)* to use these tools.

Overview Functionality of the Core Utils



- **Selection/extraction** from unstructured File(s) - search for strings/regexs

your spout, O whale! the mad fiend himself in other thing for the whale-spout, as the event White Whale now reveal his vicinity; but by furthest depths, the Sperm Whale thus booms immeasurable bravadoes the White Whale to them he would take the whale head-and-head such a course excludes the coming onset from boats were plain as the ship's three masts to But at last in his untraceable evolutions, the for a moment the whale drew aside a little, as the sea; and was all fast again. That instant, t from the sea, the White Whale dashed his br

The first uprising momentum of the **whale**—modifying its direction as he mine one jot more me, than this dead one that's lost. Nor white **whale**, iron, men, the white whale's—no, no, no,—blistered fool! this hand did matter of the **whale**, be the front of thy face to me as the palm of this When dusk descended, the whale was still in sight to leeward. "D'ye see him?" cried Ahab; but the whale was not yet in sight. palms. Leeward! the white **whale** goes that way; look to windward, then; a good eye upon the **whale**, the while I'm gone. We'll talk to-morrow, nay, to-night, when the white whale lies down there, tied by head and their bites. It is a thing not uncommonly happening to the **whale**-boats had been observed by the Pequod since the White Whale had been first there!—keep thy keenest eye upon the boats:—mark well the whale!—Ho! downward pointed arm, Ahab knew that the **whale** had sounded; but marble trunk of the **whale**. **whale** swimming out from them, turned, and showed one entire flank as he which, during the past night, the **whale** had reeled the involutions of whale? gone down again?" **Whale**'s way now began to abate, as it seemed, from the boat so rapidly nearing him once more; though indeed the **whale**'s last start had not muttered—"whether these sharks swim to feast on the **whale** or on the White **Whale**'s flank, he seemed strangely oblivious of its advance—as the whale sometimes will—and Ahab was fairly within the smoky mountain mist, which, thrown off from the **whale**'s spout, curled hated **whale**. As both steel and curse sank to the socket, as if sucked instantaneous swiftness, the White **Whale** darted through the weltering Hearing the tremendous rush of the sea-crashing boat, the whale wheeled "The **whale!** The ship!" cried the cringing oarsmen.

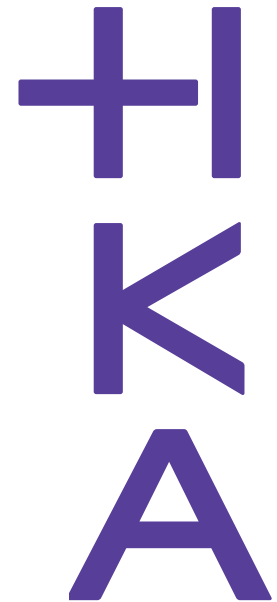
tools: **grep, sed, awk**

possible output:

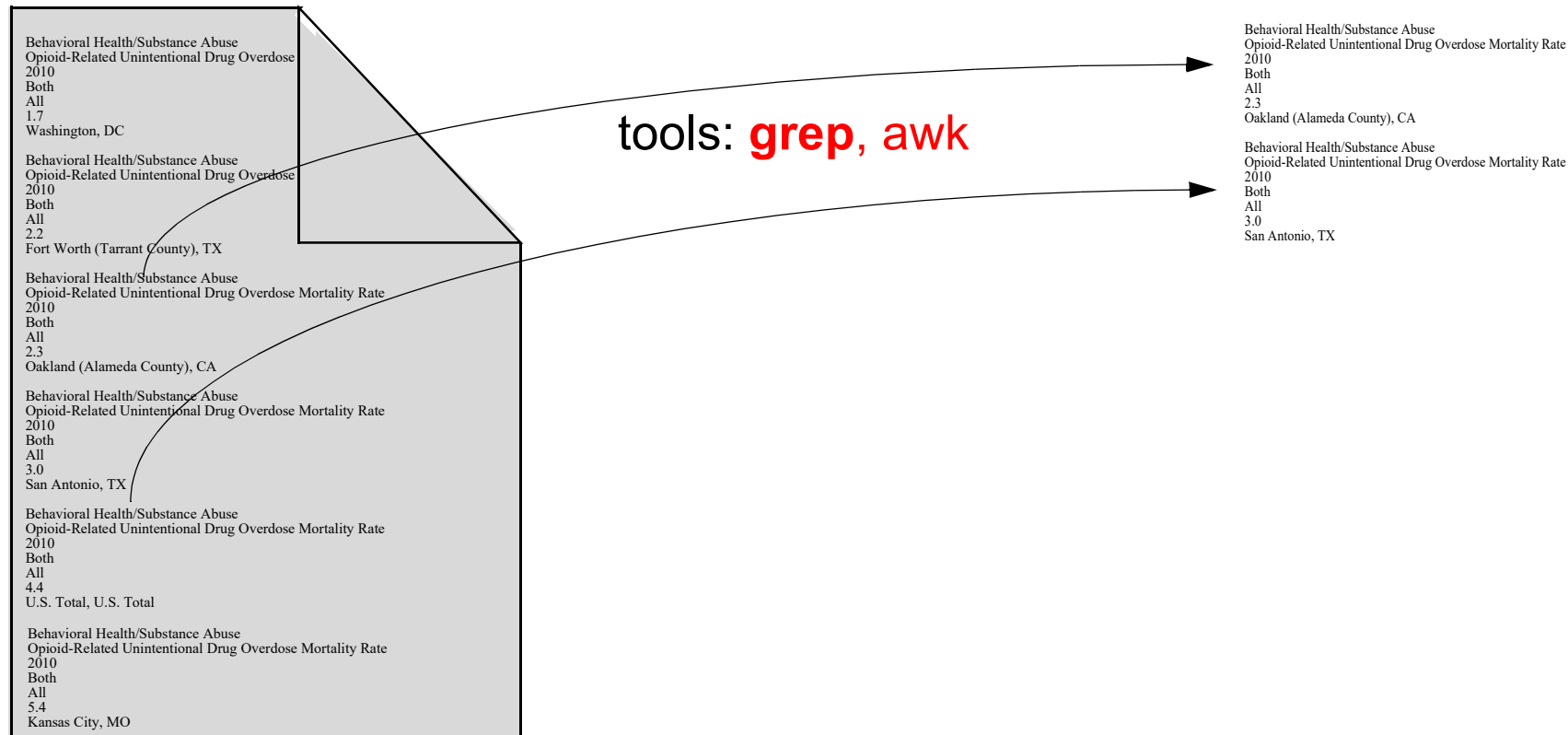
- lines (not) containing pattern
- only pattern (typically with regexps)
- filename(s) (not) containing pattern
- filename(s) & line (not) containing pattern
- additional line numbers
- with additional context lines

The first uprising momentum of the **whale**—modifying its direction as he mine one jot more me, than this dead one that's lost. Nor white **whale**, matter of the **whale**, be the front of thy face to me as the palm of this When dusk descended, the **whale** was still in sight to leeward. "D'ye see him?" cried Ahab; but the whale was not yet in sight. a good eye upon the **whale**, the while I'm gone. We'll talk to-morrow, their bites. It is a thing not uncommonly happening to the **whale**-boats downward pointed arm, Ahab knew that the **whale** had sounded; but marble trunk of the **whale**. **whale** swimming out from them, turned, and showed one entire flank as he which, during the past night, the **whale** had reeled the involutions of **Whale**'s way now began to abate, as it seemed, from the boat so rapidly nearing him once more; though indeed the **whale**'s last start had not muttered—"whether these sharks swim to feast on the **whale** or on the White **Whale**'s flank, he seemed strangely oblivious of its smoky mountain mist, which, thrown off from the **whale**'s spout, curled hated **whale**. As both steel and curse sank to the socket, as if sucked instantaneous swiftness, the White **Whale** darted through the weltering "The **whale!** The ship!" cried the cringing oarsmen.

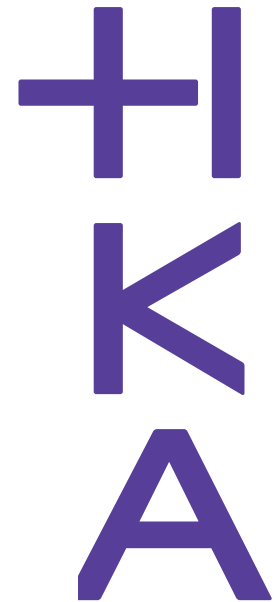
Overview Functionality of the Core Utils



- **Selection/extraction** from structured File(s)



Overview Functionality of the Core Utils



- Projection from structured Files (extract columns)

Taldyqorghan	KAZ	Taldyqorghan	116100		
Benin City	WAN	Nigeria	203000	NULL	NULL
Kabul	AFG	Afghanistan	892000	69.2	
La Paz	HCA	La Paz	NULL	NULL	NULL
Basel	CH	BS	172768	NULL	NULL
Chandler	USA	Arizona	142918	NULL	NULL
Caernarfon	GB	Gwynedd	NULL	NULL	NULL
Simla	IND	Himachal Pradesh	NULL	NULL	NULL
Jundiai	BR	Sao Paulo	293237	NULL	NULL
Buzau	RO	Buzau	145000	NULL	NULL
Chihuahua	MEX	Chihuahua	516153		
Huzhou	TJ	Zhejiang	218071	NULL	NULL
Keckemet	H	Bacs Kiskun	105000		
Helena	USA	Montana	23938	-112	46.6
Kediri	RI	Indonesia	249807	NULL	NULL
Kingston upon Hull	GB	Humberside	269100	NULL	NULL
Grenoble	F	Rhone Alpes	150758	NULL	NULL
Trowbridge	GB	Wiltshire	NULL	NULL	NULL
Usulután	ES	El Salvador	NULL	NULL	NULL
Salgotarjan	H	Nograd	NULL	NULL	NULL
Kislovodsk	R	Stavropolsky kray	120000	NULL	NULL
Swale	GB	Kent	117200	NULL	NULL
Chilung	RC	Taiwan	370049	NULL	NULL
Zhenjiang	TJ	Jiangsu	368316	NULL	NULL
Guarenas	YV	Miranda	134158	NULL	NULL
Moers	D	Nordrhein Westfalen	107011	NULL	NULL
Stroud	GB	Gloucestershire	105400	NULL	NULL
Gorzow Wielkopolski	PL	Gorzowski	123000	NULL	NULL
Urumqi	TJ	Xinjiang Uygur	1160000	88	44
Gelsenkirchen	D	Nordrhein Westfalen	293542	NULL	NULL
Cordoba	E	Andalusia	315948	NULL	NULL
Barrancabermeja	CO	Santander del Sur	180653	NULL	NULL
Trenton	USA	New Jersey	92124	-74.7667	40.2167
Thai Nguyen	VN	Vietnam	171815	NULL	NULL
Novocheboksarsk	R	Chuvash Republic	123000	NULL	NULL
Marghilon	UZB	Farghona	129000	NULL	NULL
Durres	AL	Albania	60000	19.3	41.2
George Town	MAL	Pulau Pinang	219376	NULL	NULL
Odense	DK	Denmark	136803	10.2	55.3
Inglewood	USA	California	111040	NULL	NULL

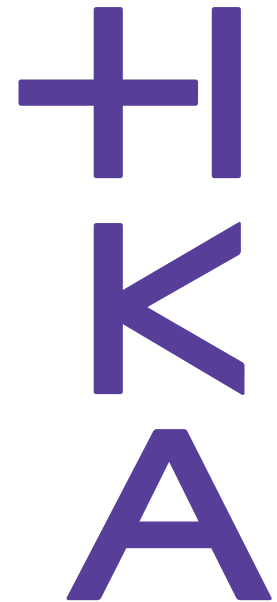
tools: **cut**, **awk**

Taldyqorghan	116100
Benin City	203000
Kabul	892000
La Paz	NULL
Basel	172768
Chandler	142918
Caernarfon	NULL
Simla	NULL
Jundiai	293237
Buzau	145000
Chihuahua	516153
Huzhou	218071
Keckemet	105000
Helena	23938
Kediri	249807
Kingston upon Hull	269100
Grenoble	150758
Trowbridge	NULL
Usulután	NULL
Salgotarjan	NULL
Kislovodsk	120000
Swale	117200
Chilung	370049
Zhenjiang	368316
Guarenas	134158
Moers	107011
Stroud	105400
Gorzow Wielkopolski	123000
Urumqi	1160000
Gelsenkirchen	293542
Cordoba	315948
Barrancabermeja	180653
Trenton	92124
Thai Nguyen	171815
Novocheboksarsk	123000
Marghilon	129000
Durres	60000
George Town	219376
Odense	136803
Inglewood	111040

Specification of ...

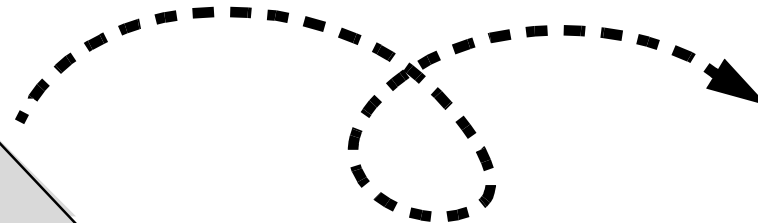
- Column separator
- desired column numbers

Overview Functionality of the Core Utils



- **Sorting of files**

```
New York USA New York 7380
Los Angeles USA California 35536
Chicago USA Illinois 2721547 -87.6
Houston USA Texas 1744058 -95.35 29
Philadelphia USA Pennsylvania 14780
San Diego USA California 117112
Phoenix USA Arizona 1159014 -112.2 33
San Antonio USA Texas 1067816 -98
Dallas USA Texas 1053292 -96.85 32.8
Detroit USA Michigan 1000272 -83.0
San Jose USA California 838744
Indianapolis USA Indiana 746737 -86.
San Francisco USA California 73531
Jacksonville USA Florida 679792 -81.7
Baltimore USA Maryland 675401 -76.4167 39.3333
Columbus USA Ohio 657053 -83 39.5
El Paso USA Texas 599865 NULL NULL
Memphis USA Tennessee 596725 -90 35.05
Milwaukee USA Wisconsin 590503 -87.9 42.95
Boston USA Massachusetts 558394 -71.0333 42.3667
Washington USA Distr. Columbia 543213 -77 38.5
Austin USA Texas 541278 -97.7 30.3
Seattle USA Washington 524704 NULL NULL
Nashville Davidson USA Tennessee 511263 NULL NULL
Cleveland USA Ohio 498246 NULL NULL
Denver USA Colorado 497840 -104.867 39.75
Portland USA Oregon 480824 -122.6 45.6
Fort Worth USA Texas 479716 -97.1 32.4
New Orleans USA Louisiana 476625 -90.25 29.9833
Oklahoma City USA Oklahoma 469852 -97.3 35.3
Nashville USA Tennessee 455657 -86.4 36.1
Tucson USA Arizona 449002 NULL NULL
Charlotte USA North Carolina 441297 -80.9333 35.2167
Kansas City USA Missouri 441259 -94.5833 39.1167
Virginia Beach USA Virginia 430385 NULL NULL
Honolulu USA Hawaii 423475 -157.917 21.3333
Long Beach USA California 421904 -118.15 33.8167
Albuquerque USA New Mexico 419681 NULL NULL
Atlanta USA Georgia 401907 -84.4 33.4
Fresno USA California 396011 NULL NULL
```

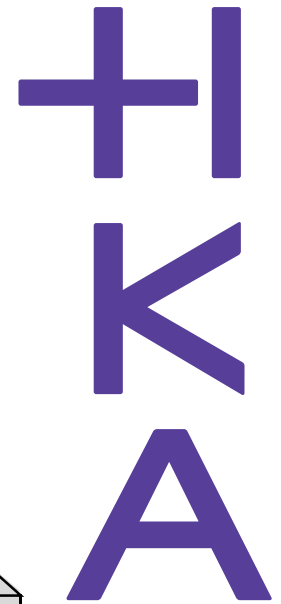


tools: **sort**

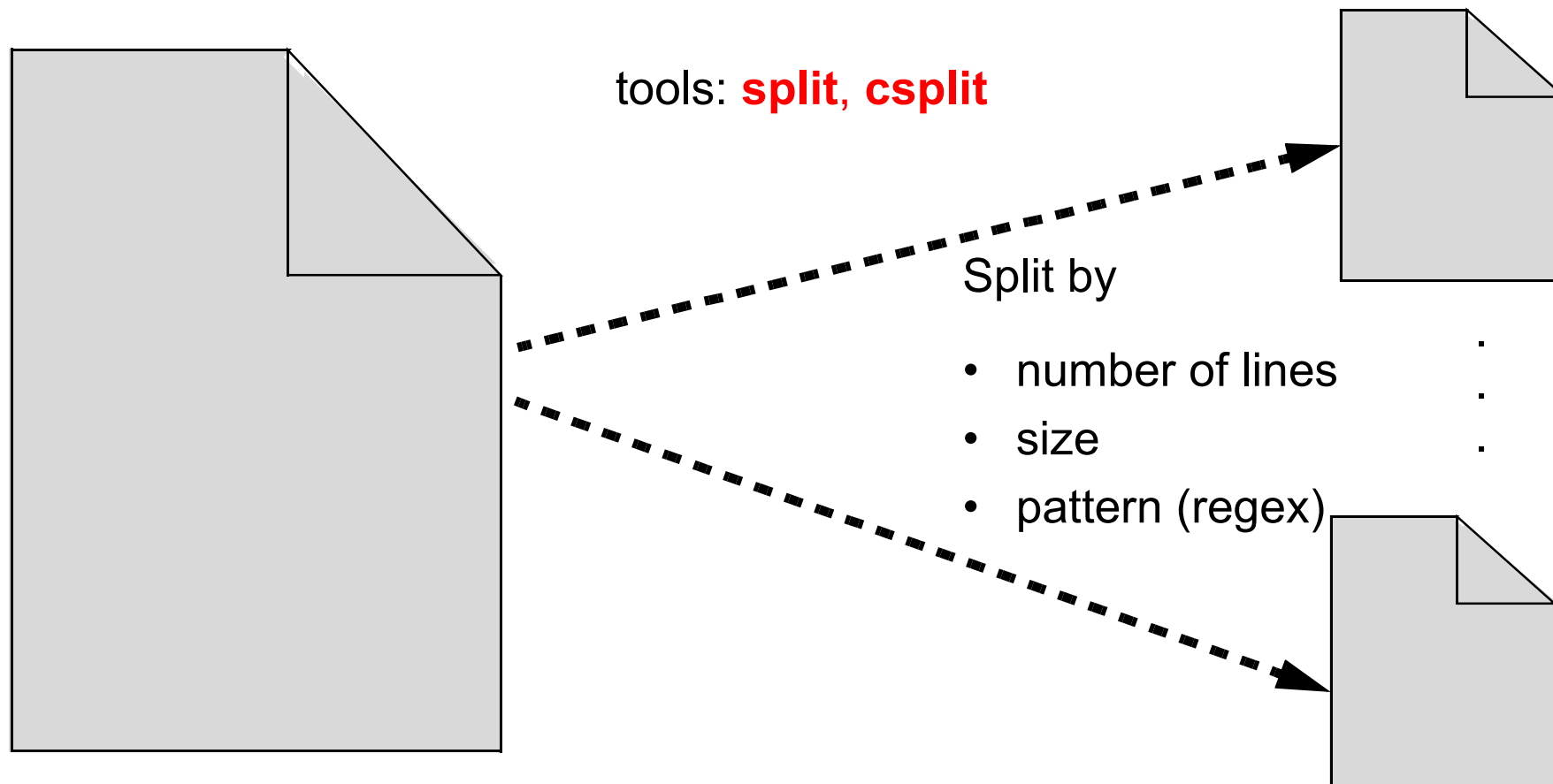
- line-by-line sorting
- Multiple criteria sort
- alphabetic, numeric, random sort
- ascending/descending

```
Phoenix USA Arizona 1159014 -112.2 33.2
Tucson USA Arizona 449002 NULL NULL
Los Angeles USA California 3553638 -118 34
San Diego USA California 1171121 -117.167 32.7333
San Jose USA California 838744 -121.933 37.3667
San Francisco USA California 735315 -122.383 37.6167
Long Beach USA California 421904 -118.15 33.8167
Fresno USA California 396011 NULL NULL
Denver USA Colorado 497840 -104.867 39.75
Washington USA Distr. Columbia 543213 -77 38.5
Jacksonville USA Florida 679792 -81.7 30.5
Atlanta USA Georgia 401907 -84.4 33.4
Honolulu USA Hawaii 423475 -157.917 21.3333
Chicago USA Illinois 2721547 -87.6333 41.8833
Indianapolis USA Indiana 746737 -86.2833 39.7333
New Orleans USA Louisiana 476625 -90.25 29.9833
Baltimore USA Maryland 675401 -76.4167 39.3333
Boston USA Massachusetts 558394 -71.0333 42.3667
Detroit USA Michigan 1000272 -83.0167 42.4167
Kansas City USA Missouri 441259 -94.5833 39.1167
Albuquerque USA New Mexico 419681 NULL NULL
New York USA New York 7380906 -74 40.4
Charlotte USA North Carolina 441297 -80.9333 35.2167
Columbus USA Ohio 657053 -83 39.5
Cleveland USA Ohio 498246 NULL NULL
Oklahoma City USA Oklahoma 469852 -97.3 35.3
Portland USA Oregon 480824 -122.6 45.6
Philadelphia USA Pennsylvania 1478002 -75.25 39.8833
Memphis USA Tennessee 596725 -90 35.05
Nashville Davidson USA Tennessee 511263 NULL NULL
Nashville USA Tennessee 455657 -86.4 36.1
Houston USA Texas 1744058 -95.35 29.9667
San Antonio USA Texas 1067816 -98.4 29.3
Dallas USA Texas 1053292 -96.85 32.85
El Paso USA Texas 599865 NULL NULL
Austin USA Texas 541278 -97.7 30.3
Fort Worth USA Texas 479716 -97.1 32.4
Virginia Beach USA Virginia 430385 NULL NULL
Seattle USA Washington 524704 NULL NULL
Milwaukee USA Wisconsin 590503 -87.9 42.95
```

Overview Functionality of the Core Utils

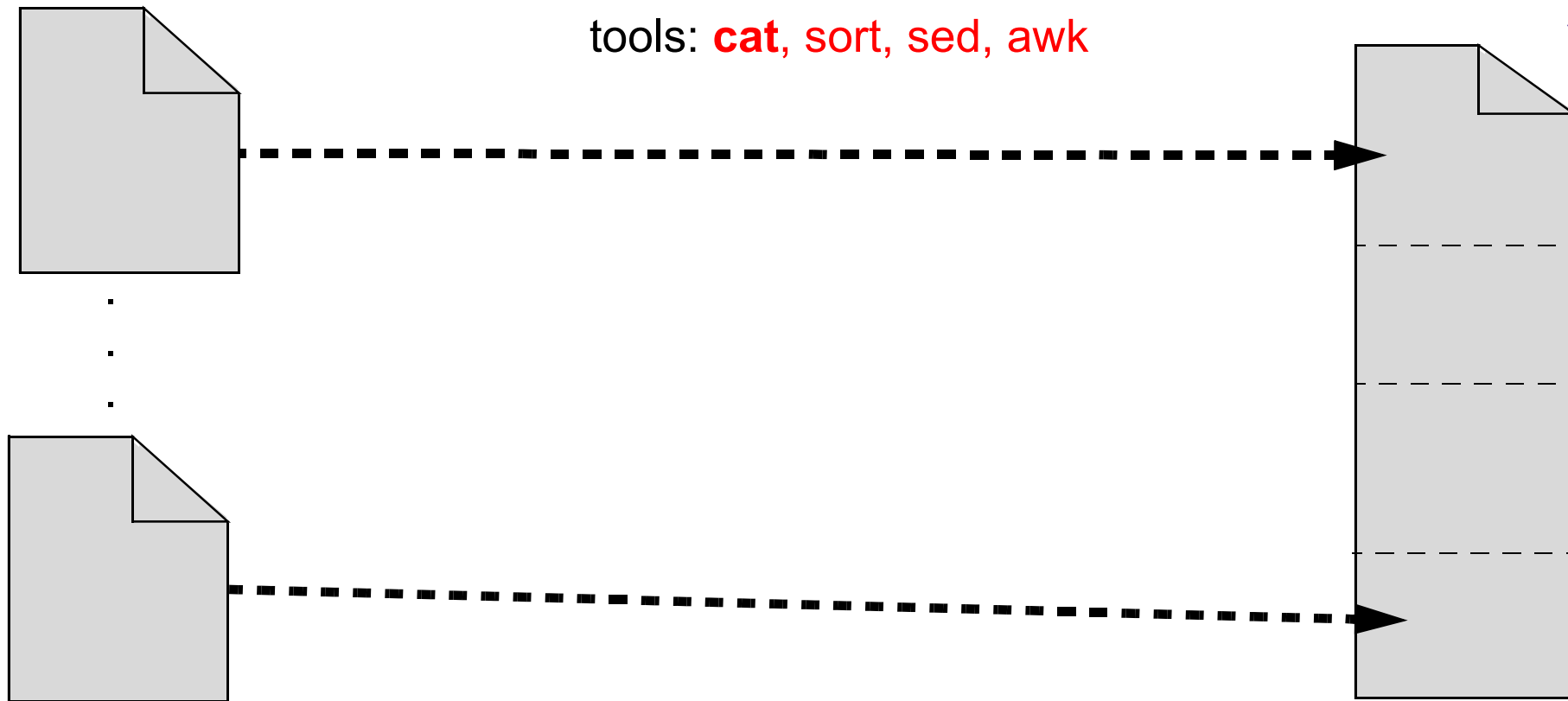


- **Splitting Files**



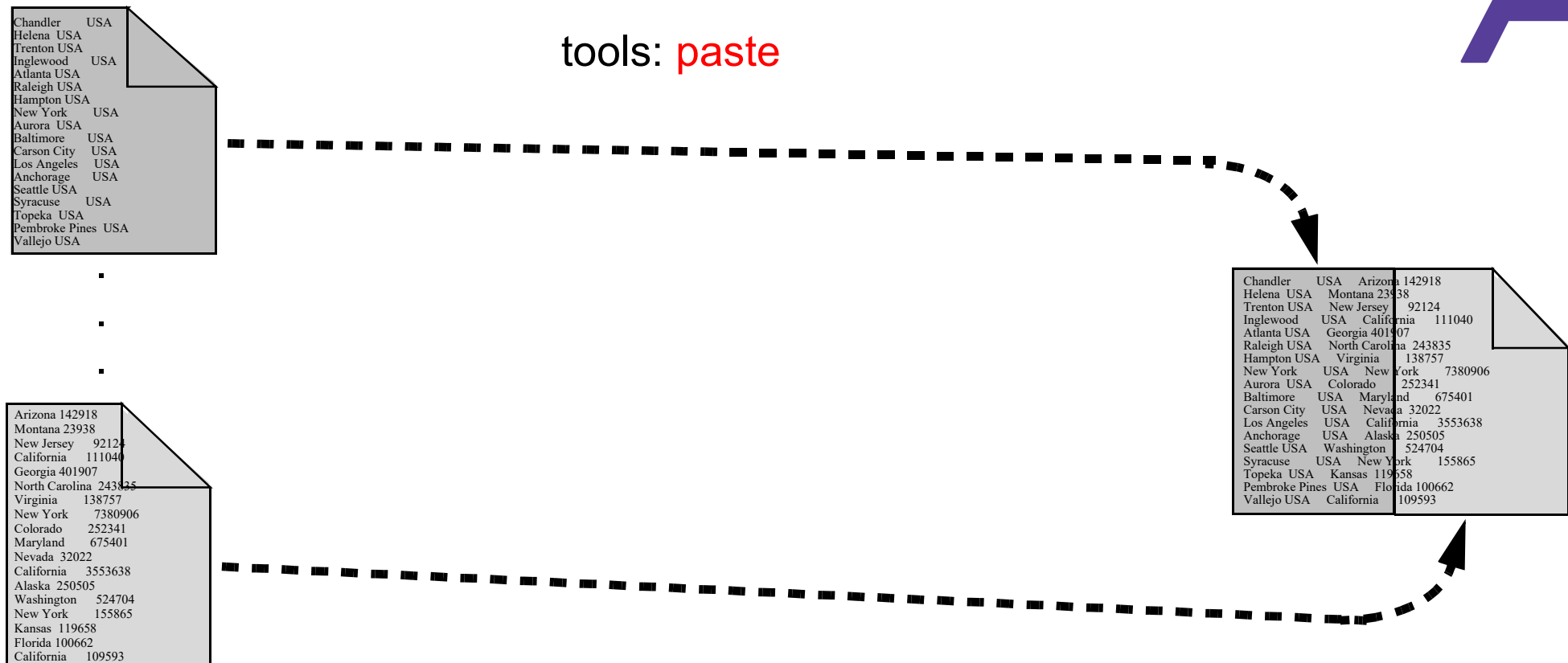
Overview Functionality of the Core Utils

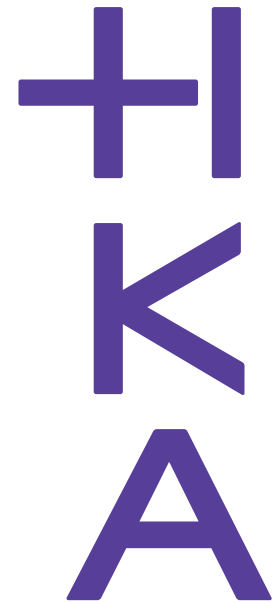
- Merging Files (1) - line-by-line



Overview Functionality of the Core Utils

- Merging Files (2) - column-by-column





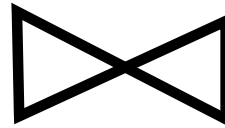
Overview Functionality of the Core Utils

- **Merging Files (3) - by matching column**

Eisenstadt	A	E
Klagenfurt	A	C
St. Polten	A	L
Salzburg	A	S
Graz	A	Styria
Innsbruck	A	
Linz	A	Upper Austria 203000
Vienna	A	Vienna 1583000 16.36
Bregenz	A	Vorarlberg NULL
Kabul	AFG	Afghanistan 892000
Saint Johns	AG	Antigua and Bar
Tirane	AL	Albania 192000 10.7
Korce	AL	Albania 52000 20.5
Elbasan	AL	Albania 53000 20.1
Vlore	AL	Albania 56000 19.3
Durres	AL	Albania 60000 19.3
Shkoder	AL	Albania 62000 19.2
Andorra la Vella	AND	Andorra

tools: **join**

Eisenstadt	10102	Austria
Klagenfurt	87321	Austria
St. Polten	51102	Austria
Salzburg	144000	Austria
Graz	238000	Austria A
Innsbruck	118000	Austria A
Linz	203000	Austria A
Vienna	1583000	Austria A
Bregenz	NULL	Austria A
Kabul	892000	Afghanistan AFG
Saint Johns	36000	Antigua and Barbuda AG
Tirane	192000	Albania AL
Korce	52000	Albania AL
Elbasan	53000	Albania AL
Vlore	56000	Albania AL
Durres	60000	Albania AL
Shkoder	62000	Albania AL

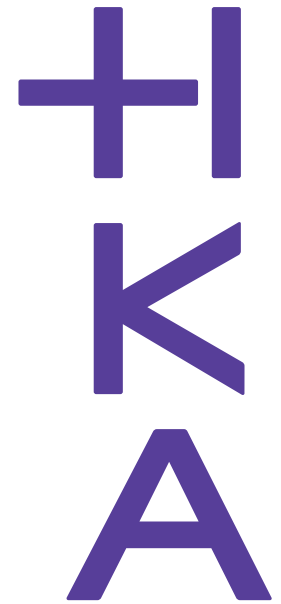


Specification of ...

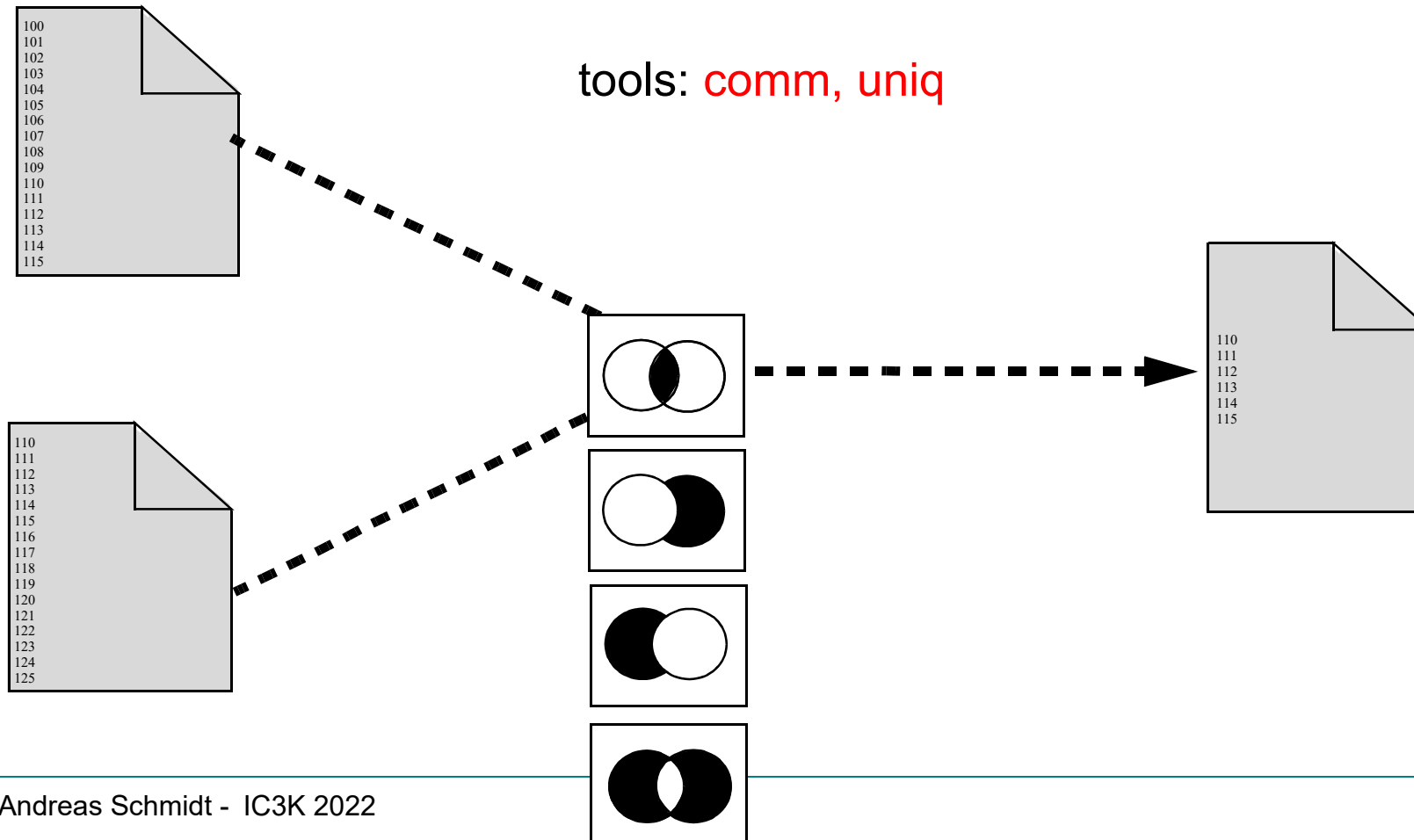
- Join columns (equi join)
- output columns
- support for outer join

Austria	A	Vienna
Afghanistan	AFG	
Antigua and Barbuda	AG	
Albania	AL	Tiran
Andorra	AND	An
Angola	ANG	Luanda Luanda 124
Armenia	ARM	Yerevan Armenia 29
Australia	AUS	Canberra A
Azerbaijan	AZ	Baku Azerbaij
Belgium	B	Brussels Brabant 3
Bangladesh	BD	Dhaka Bangla
Barbados	BDS	Bridgetown
Benin	BEN	Porto-Novo Benin
Burkina Faso	BF	Ouagadougou
Bulgaria	BG	Sofia Bulgaria
Bhutan	BHT	Thimphu Bhutan 470
Burundi	BI	Bujumbura Burund
Bosnia and Herzegovina	BIH	Saraj

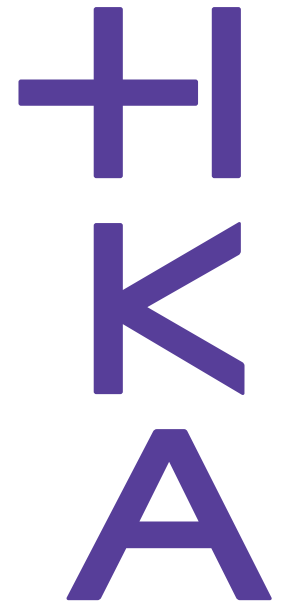
Overview Functionality of the Core Utils



- **Set based operations: Intersect, Minus, Duplicate detection/elimination**

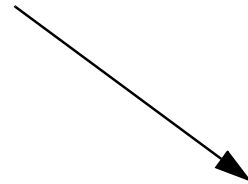


Overview Functionality of the Core Utils



- Script based modification of Files (programatic editing)

Adding lines

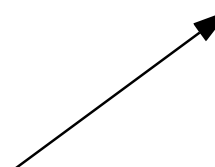


```
Chandler USA
Helena USA
Trenton USA
Inglewood USA
Atlanta USA
Raleigh USA
Hampton USA
New York USA
Aurora USA
Baltimore USA
Carson City USA
Los Angeles USA
Anchorage USA
Seattle USA
Syracuse USA
Topeka USA
Pembroke Pines USA
Vallejo USA
```

Modifying lines

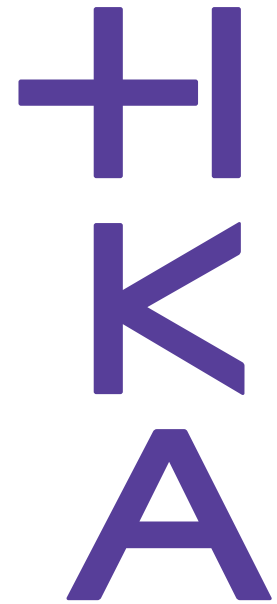


Deleting lines



tools: **sed**, **awk**

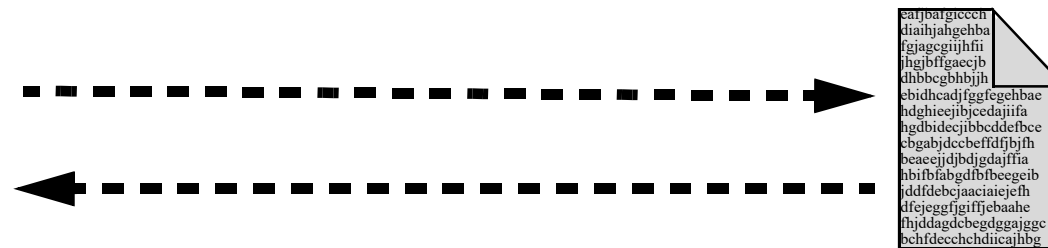
Overview Functionality of the Core Utils



- File compression

```
New York USA New York 7380
Los Angeles USA California 35536
Chicago USA Illinois 2721547 -87.6
Houston USA Texas 1744058 -95.35 29
Philadelphia USA Pennsylvania 14780
San Diego USA California 117112
Phoenix USA Arizona 1159014 -112.2 3
San Antonio USA Texas 1067816 -98
Dallas USA Texas 1053292 -96.85 32.8
Detroit USA Michigan 1000272 -83.0
San Jose USA California 838744
Indianapolis USA Indiana 746737 -86
San Francisco USA California 73531
Jacksonville USA Florida 679792 -81.7
Baltimore USA Maryland 675401 -76.4167 39.3333
Columbus USA Ohio 657053 -83 39.5
El Paso USA Texas 599865 NULL NULL
Memphis USA Tennessee 596725 -90 35.05
Milwaukee USA Wisconsin 590503 -87.9 42.95
Boston USA Massachusetts 558394 -71.0333 42.3667
Washington USA Distr. Columbia 543213 -77 38.5
Austin USA Texas 541278 -97.7 30.3
Seattle USA Washington 524704 NULL NULL
Nashville Davidson USA Tennessee 511263 NULL NULL
Cleveland USA Ohio 498246 NULL NULL
Denver USA Colorado 497840 -104.867 39.75
Portland USA Oregon 480824 -122.6 45.6
Fort Worth USA Texas 479716 -97.1 32.4
New Orleans USA Louisiana 476625 -90.25 29.9833
Oklahoma City USA Oklahoma 469852 -97.3 35.3
Nashville USA Tennessee 455657 -86.4 36.1
Tucson USA Arizona 449002 NULL NULL
Charlotte USA North Carolina 441297 -80.9333 35.2167
Kansas City USA Missouri 441259 -94.5833 39.1167
Virginia Beach USA Virginia 430385 NULL NULL
Honolulu USA Hawaii 423475 -157.917 21.3333
Long Beach USA California 421904 -118.15 33.8167
Albuquerque USA New Mexico 419681 NULL NULL
Atlanta USA Georgia 401907 -84.4 33.4
Fresno USA California 396011 NULL NULL
```

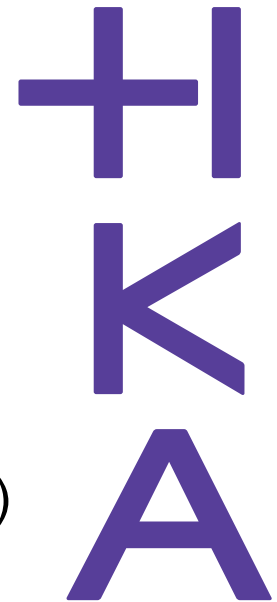
tools: **gzip, gunzip, bzip2, bunzip2**



Operations on compressed data:

- search
- list/concatenate
- inspect interactively

Overview Functionality of the Core Utils



- Character based **transformations**
 - Translate single characters or ranges of characters (i.e. A-Z -> a-z)
 - Delete specified characters
 - Squeeze repeated occurrences of specified characters

tools: **tr**, **sed**

- **Transformations** based on regular expressions

- example (date transformation):

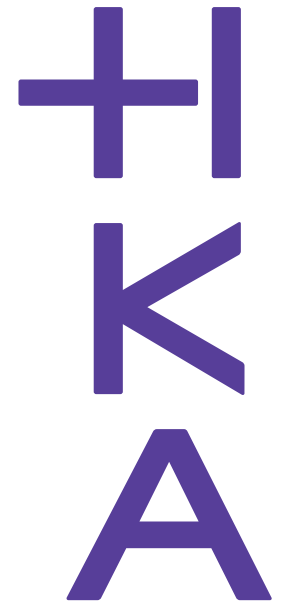
31.9.2019 -> 2019-9-31

```
s#\b([0-9]{1,2})\.[0-9]{1,2}\.([0-9]{4})\b#\3-\2-\1#
```

matching pattern

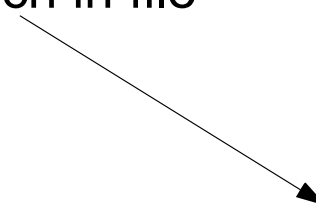
replace-
ment

Overview Functionality of the Core Utils

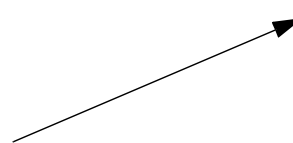


- **File Inspection**

Navigate/search in file

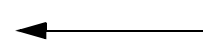


View first/last *n* lines



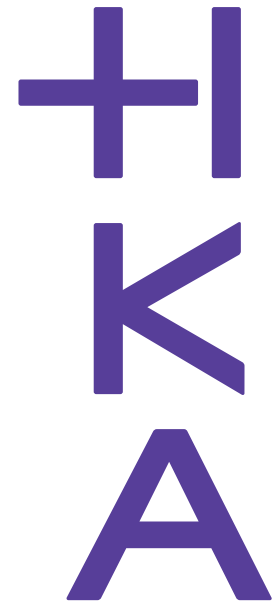
```
New York USA New York 7380
Los Angeles USA California 35536
Chicago USA Illinois 2721547 87.6
Houston USA Texas 1744058 -95.35 29
Philadelphia USA Pennsylvania 14780
San Diego USA California 117112
Phoenix USA Arizona 1159014 112.2 33
San Antonio USA Texas 1067816 -98
Dallas USA Texas 1053292 -96.85 32.8
Detroit USA Michigan 1000272 -83.0
San Jose USA California 838744
Indianapolis USA Indiana 746737 -86
San Francisco USA California 73531
Jacksonville USA Florida 679792 -81.7
Baltimore USA Maryland 675401 -76.4167 39.3333
Columbus USA Ohio 657053 -83 39.5
El Paso USA Texas 599865 NULL NULL
Memphis USA Tennessee 596725 -90 35.05
Milwaukee USA Wisconsin 590503 -87.9 42.95
Boston USA Massachusetts 558394 -71.0333 42.3667
Washington USA Distr. Columbia 543213 -77 38.5
Austin USA Texas 541278 -97.7 30.3
Seattle USA Washington 524704 NULL NULL
Nashville Davidson USA Tennessee 511263 NULL NULL
Cleveland USA Ohio 498246 NULL NULL
Denver USA Colorado 497840 -104.867 39.75
Portland USA Oregon 480824 -122.6 45.6
Fort Worth USA Texas 479716 -97.1 32.4
New Orleans USA Louisiana 476625 -90.25 29.9833
Oklahoma City USA Oklahoma 469852 -97.3 35.3
Nashville USA Tennessee 455657 -86.4 36.1
Tucson USA Arizona 449902 NULL NULL
Charlotte USA North Carolina 441297 -80.9333 35.2167
Kansas City USA Missouri 441259 -94.5833 39.1167
Virginia Beach USA Virginia 430385 NULL NULL
Honolulu USA Hawaii 423475 -157.917 21.3333
Long Beach USA California 421904 -118.15 33.8167
Albuquerque USA New Mexico 419681 NULL NULL
Atlanta USA Georgia 401907 -84.4 33.4
Fresno USA California 396011 NULL NULL
```

Show content



tools: **less**, **head**, **tail**, **cat**, **sed**

Overview Functionality of the Core Utils



- **Counting** lines, words, bytes

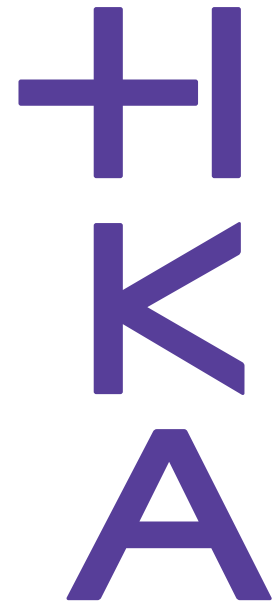
```
New York USA New York 7380
Los Angeles USA California 35536
Chicago USA Illinois 2721547 876
Houston USA Texas 1744058 -95.35 29
Philadelphia USA Pennsylvania 14786
San Diego USA California 117112
Phoenix USA Arizona 1159014 -112.2 33
San Antonio USA Texas 1067816 -98
Dallas USA Texas 1053292 -96.85 32.8
Detroit USA Michigan 1000272 -83.0
San Jose USA California 838744
Indianapolis USA Indiana 746737 -86
San Francisco USA California 73531
Jacksonville USA Florida 679792 -81.1
Baltimore USA Maryland 675401 -76.4167 39.3333
Columbus USA Ohio 657053 -83 39.5
El Paso USA Texas 599865 NULL NULL
Memphis USA Tennessee 596725 -90 35.05
Milwaukee USA Wisconsin 590503 -87.9 42.95
Boston USA Massachusetts 558394 -71.0333 42.3667
Washington USA Distr. Columbia 543213 -77 38.5
Austin USA Texas 541278 -97.7 30.3
Seattle USA Washington 524704 NULL NULL
Nashville Davidson USA Tennessee 511263 NULL NULL
Cleveland USA Ohio 498246 NULL NULL
Denver USA Colorado 497840 -104.867 39.75
Portland USA Oregon 480824 -122.6 45.6
Fort Worth USA Texas 479716 -97.1 32.4
New Orleans USA Louisiana 476625 -90.25 29.9833
Oklahoma City USA Oklahoma 469852 -97.3 35.3
Nashville USA Tennessee 455657 -86.4 36.1
Tucson USA Arizona 449002 NULL NULL
Charlotte USA North Carolina 441297 -80.9333 35.2167
Kansas City USA Missouri 441259 -94.5833 39.1167
Virginia Beach USA Virginia 430385 NULL NULL
Honolulu USA Hawaii 423475 -157.917 21.3333
Long Beach USA California 421904 -118.15 33.8167
Albuquerque USA New Mexico 419681 NULL NULL
Atlanta USA Georgia 401907 -84.4 33.4
Fresno USA California 396011 NULL NULL
```



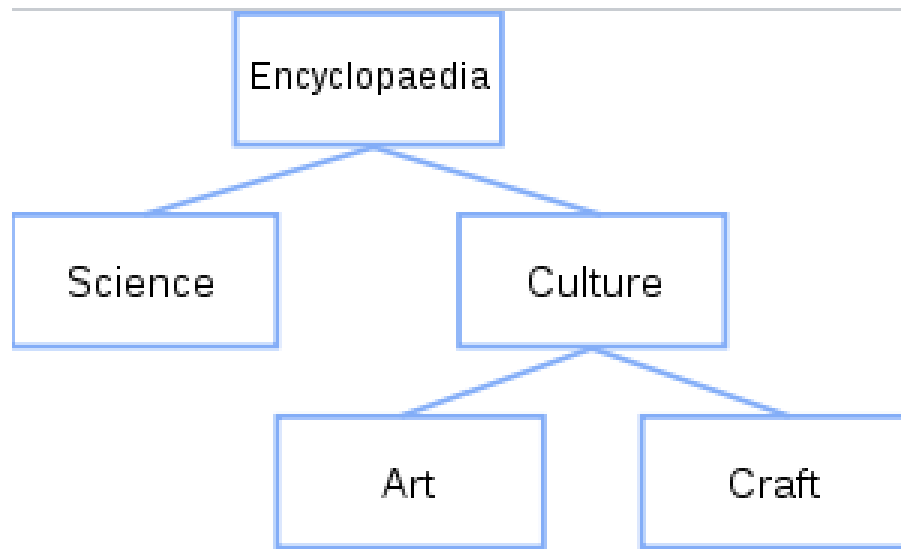
n lines
k words
i bytes

tools: **wc**

Overview Functionality of the Core Utils



- **Search directory tree**



tools: **find, ls**

Search by

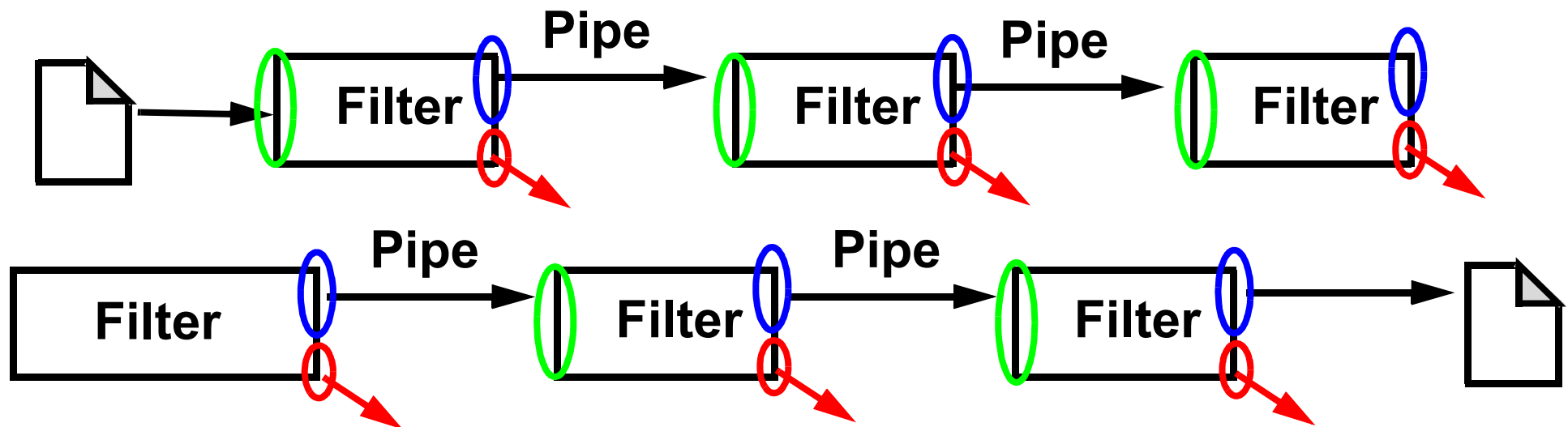
- name pattern
- type
- access date
- user/group
- ...
- a mix of above

an additional action can be performed on the found files (default: print filename)

Filter and Pipes (Combining commands)

Communication between filters via channels

- Standard Input (**STDIN**)
- Standard Output (**STDOUT**)
- Standard Error (**STDERR**)





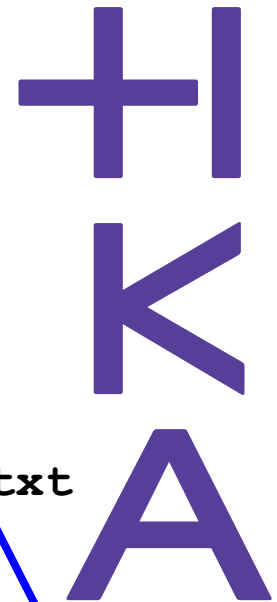
Communication Channels/Redirection

- In-/Output Redirection
 - `|` : Pipe operator: Connect **STDOUT** of a command with **STDIN** of the next command
 - `>` : Redirect Standard Output (into file)
 - `<` : Redirect Standard Input (from file)
 - `2>` : Redirect Standard Error (into file)
 - `>>` : Redirect Standard Output (append into file)

- Example¹:

```
cut -d, -f1 city.csv | sort | uniq -c | \  
sort -nr | awk '$1>1' > result.txt
```

1. <https://www.smiffy.de/KDIR-2022/>



Retrieving the names of cities which have „name siblings“

```
cut -d, -f1 city.csv|sort|uniq -c|sort -nr|awk '$1>1' > res.txt
```

Binjai
Hsinchu
Zhuhai
Jinxi
Reynosa
Livonia
"Hpa an"
Paterson
Kaifeng
Orlando
Brescia
Tepic
...

...

Aachen
Aalborg
Aarau
Aarhus
Aarri
Aba
Abakan
Abancay
Abeokuta
Aberdeen
Aberystwyth

...
1 Leiyang
1 Lekoa
1 Lelystad
1 Lengshuijiang
1 Leninsk
3 Leon
1 Leshan
1 Leszno
1 Leticia
1 Leverkusen
...

3 Trujillo
3 Springfield
3 Merida
3 Leon
3 Kingston
3 Cordoba
3 Alexandria
3 "La Paz"
2 York
2 Yichun
...
1 Zurich

3 Trujillo
3 Springfield
3 Merida
3 Leon
3 Kingston
3 Cordoba
3 Alexandria
3 "La Paz"
2 York
2 Yichun



Another example: Word count

```
grep '[A-Za-z]+' -Eo moby-dick.txt | \  
tr 'A-Z' 'a-z' | \  
sort | uniq -c | sort -nr | \  
less
```

The
Project
Gutenberg
EBook
of
Moby
Dick
or
The
Whale
...

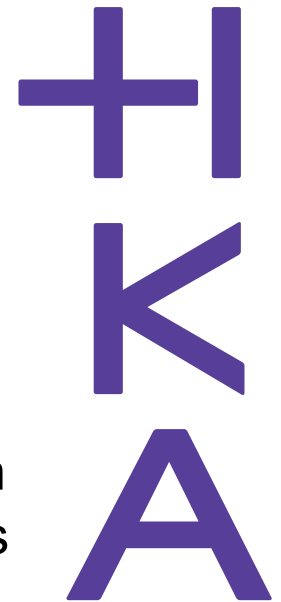
the
project
gutenberg
ebook
of
moby
dick
or
the
whale
...

a
a
a
a
a
a
a
a
a
a
...

4805 a
2 aback
2 abaft
3 abandon
7 abandoned
1 abandonedly
2 abandonment
2 abased

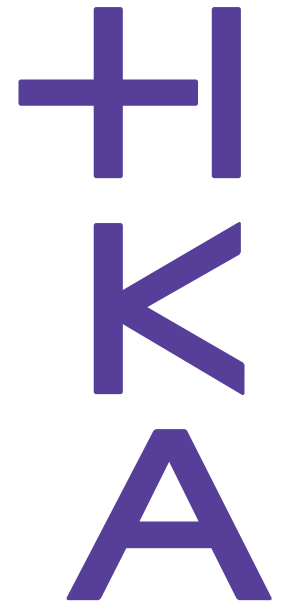
14715 the
6742 of
6517 and
4805 a
4707 to
4241 in
3100 that
2536 it
2532 his

Some Best Practice



- Incremental development of a complex pipe command (filter by filter)
- Typically, the last command is a *less* or *head* command, so that you can see, what's the result, but not get overrun by the large number of results
- While development: If the input is large, start with a *head -n<x>* command, to reduce the data lines to process or simply extract an sample using *awk*^(*)
- Create intermediate result file, so that the entire process chain does not have to be repeatedly run through.

```
(*)  
# 0.01 % extract  
awk 'rand() < 0.0001 {print $0}' very-big-file.csv
```



What happens next ...

- Some general comments about the commands
- Short presentation of the most important commands
- for each command ...
 - mention most important features
 - possible preconditions
 - typically show one or two examples to clarify usage
 - link to further examples
- 2 (3) Exercises: Based on a series of police files, you have to solve a criminal case - thrilling !!!!! ;-)
 - Resources (see <https://www.smiffy.de/KDIR-2022/>):
 - Slideset
 - Command Refcard
 - Command examples



General comment

- Most of the commands accept the input from file or from STDIN. If no (or not enough) input files are given, it is expected that the input comes from STDIN (some commands like *join*, *comm* expect a „-“ character as parameter, if the input comes from STDIN)

```
head -n4 my-file.txt
```

```
cat -n my-file.txt | head -n4
```

- Most of the commands have a lot of options which couldn't be explained in detail. To get an overview of the possibilities of a command, simply type

```
man command
```

- Example:

```
man head
```

```
/cygdrive/c/Users/scan0004/Dropbox/dbkda-2017/tutorial
HEAD(1) User Commands HEAD(1)
NAME
  head - output the first part of files
SYNOPSIS
  head [OPTION]... [FILE]...
DESCRIPTION
  Print the first 10 lines of each FILE to standard output. With more
  than one FILE, precede each with a header giving the file name.

  With no FILE, or when FILE is -, read standard input.

  Mandatory arguments to long options are mandatory for short options
  too.

  -c, --bytes=[-]JNUM
        print the first NUM bytes of each file; with the leading '-',
        print all but the last NUM bytes of each file

  -n, --lines=[-]JNUM
        print the first NUM lines instead of the first 10; with the
        leading '-', print all but the last NUM lines of each file

  -q, --quiet, --silent
        never print headers giving file names

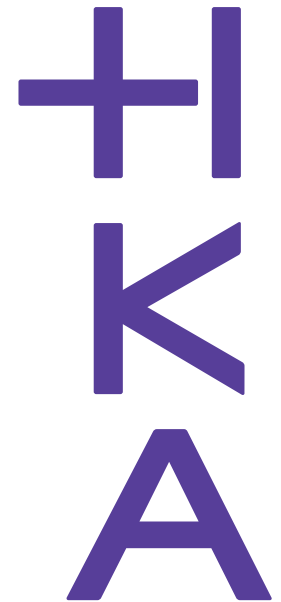
  -v, --verbose
        always print headers giving file names

  -z, --zero-terminated
        line delimiter is NUL, not newline

  --help display this help and exit

  --version
        output version information and exit
Manual page head(1) line 1 (press h for help or q to quit)
```

cat command



- Print content of file to STDOUT

```
cat HelloWorld.java
```

- Concatenate files and writes them via redirection (>) to a file

```
cat german_cities.csv french_cities.csv > cities.csv
```

```
cat *_cities.csv > cities.csv
```

- Add line numbers to each line in file(s)

```
cat -n city.csv
```

- Create a file with input from STDIN:

```
cat > grep-search-words.txt
```

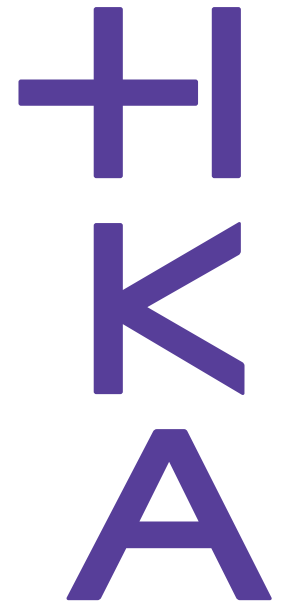
```
Obama
```

```
Climate
```

```
CTRL-D
```

- More example:

```
https://www.smiffy.de/KDIR-2022/command-examples/cat,%20head,%20tail,%20less,%20wc
```



head/tail/wc command

- **head: view first n lines or skip last n lines of a file.**

- View first 5 lines from file:

```
head -n5 city.csv
```

- Print all but the last 20 lines:

```
head -n -20 city.csv
```

to remove trailing line(s)

- **tail: view last n lines or start from line n**

- View last 10 lines of a file

```
tail -n 10 city.csv
```

- **wc: Count the number of lines, words and bytes**

```
wc city.csv
```

less command



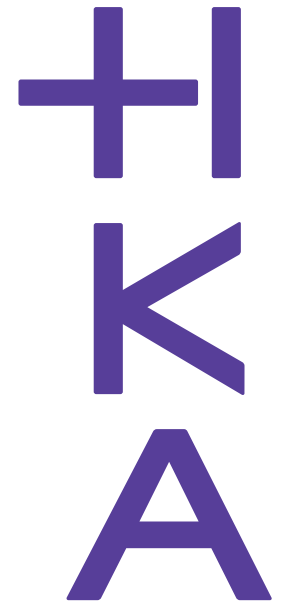
- Page by page scrolling of a file or STDIN (also with search capability)
- Examples:

```
less city.csv  
ls -l | less
```

```
man head      # inspection of man-pages with less !!
```

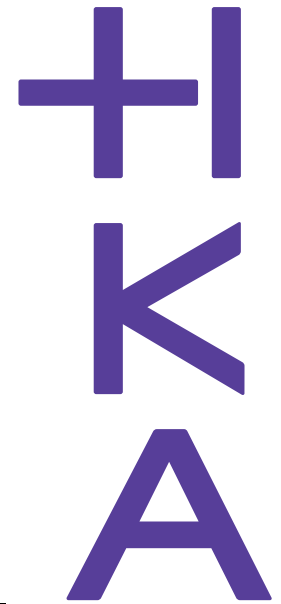
- Commands:
 - `q` : quit less
 - `>` : Goto end of file
 - `<` : Goto begin of file
 - `f`: Scroll forward one page
 - `b`: scroll backwards on page
 - `e, ret, ↓` : scroll forward one line
 - `y, ↑` : scroll backwards one line
 - `nd` : scroll forward n lines (i.e. $20n$)
 - `mb` : scroll backwards m lines
 - `ng`: Goto line $\langle n \rangle$

less commands (2)



- */pattern* : Search forward the next line with *pattern*
- *?pattern* : Search backward the previous line with *pattern*
- *n* : repeat previous search
- *N* : repeat previous search in reverse direction
- *&pattern* : Display only lines containing the *pattern* (type *<ret>* to quit)
- *!command* : executes shell command
- *v* : invokes standard editor for file (at current position, if supported)

type `man less` for complete reference



grep command

- Print lines matching pattern (case sensitive)

```
grep USA city.csv
```

- Print lines containing the regular expression (City starting with 'S', ending with 'g')

```
grep -E 'S[a-z]+g,' city.csv
```

- Print only lines, **not** containing the String NULL

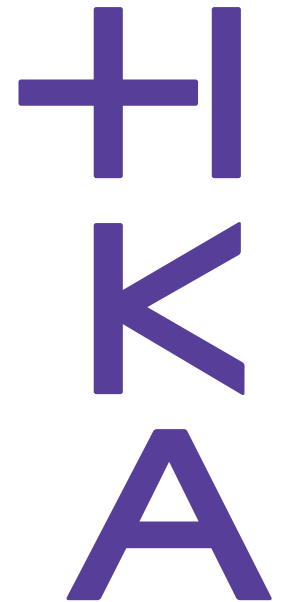
```
grep -v NULL city.csv
```

- Print lines which contain the pattern 'Agassi'

```
grep Agassi bbc sport/tennis/*.txt
```

when multiple files are queried,
the filename is part of the
result (<filename>:<line matching pattern>)

Search

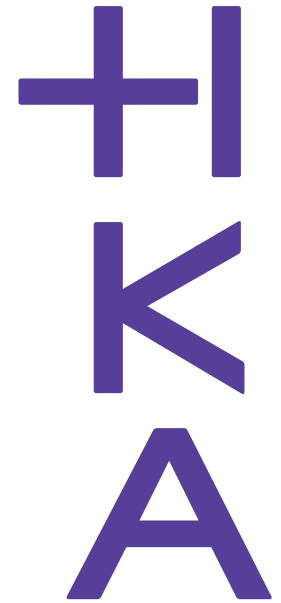


- Print **name of files** which contain the pattern 'Agassi'
`grep -l Agassi bbc sport/tennis/*.txt`
- Print only **matching part** (i.e. 'Salzburg' instead of whole line)
`grep -E -o 'S[a-z]+g' city.csv`
- Look for lines containing words from a file (OR-Semantic)
`grep -f grep-search-words.txt -E newsCorpora.csv`
 - file: grep-search-words.txt
Obama
Climate

More example:

<https://www.smiffy.de/KDIR-2022/command-examples/grep>

File operations



- Print selected parts of lines from each file to standard output.

```
cut -d',' -f1,4 city.csv
```

Column separator

Output columns 1 and 4

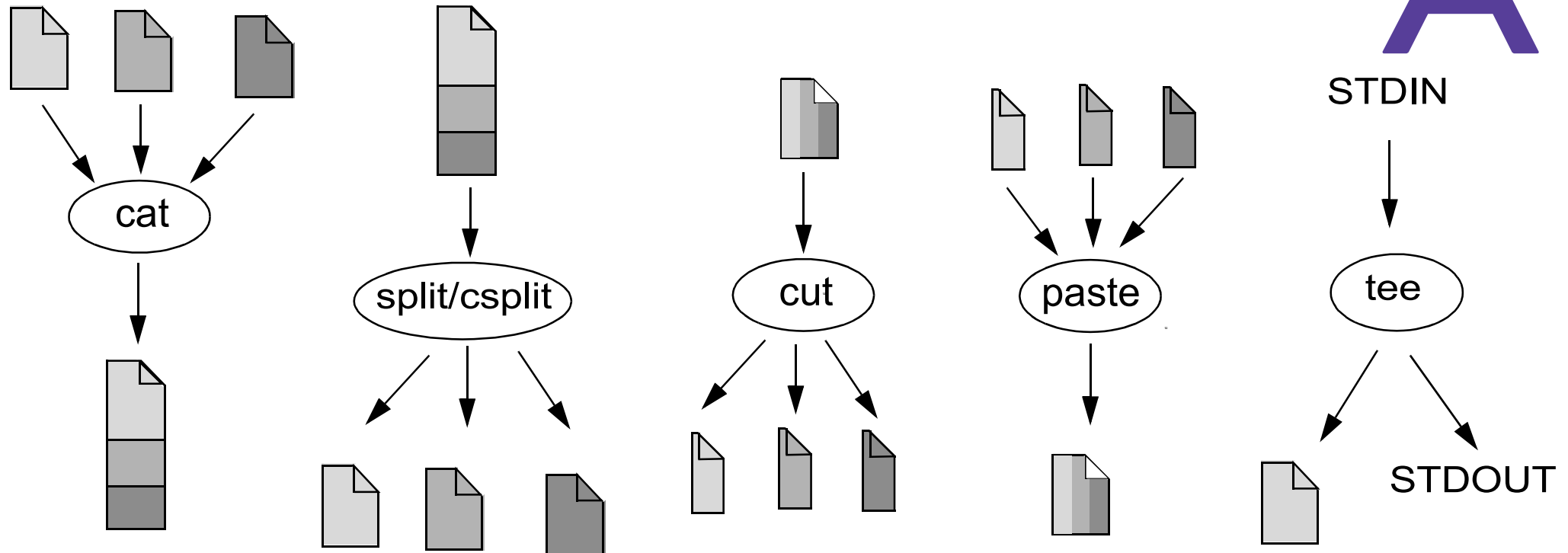
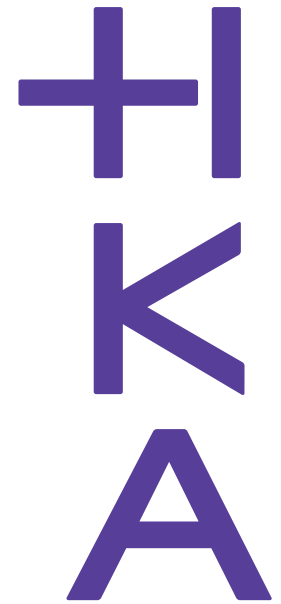
- Output bytes 10 to 20 from each line

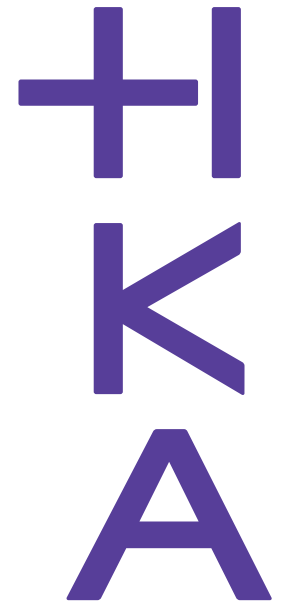
```
cut -b10-20 data.fixed
```

- Output bytes 1-5 and starting from position 20 to the end of line:

```
cut -b1-5,20- data.fixed
```

Summary of Fundamental File Operations



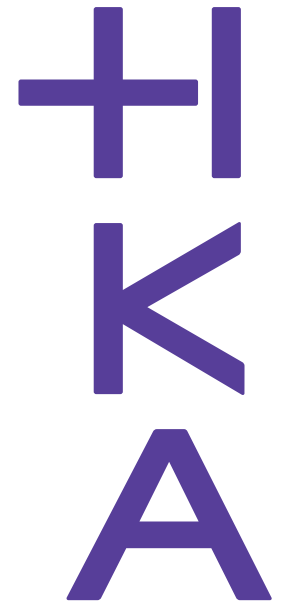


And now you are prepared for a ...

*thrilling exercise** **!!!!!!**

Go to the page www.smiffy.de/KDIR-2022, open the first Exercise and work on the task ...

(*) command line murders by Noah Veltman,
<https://github.com/veltman/clmystery>



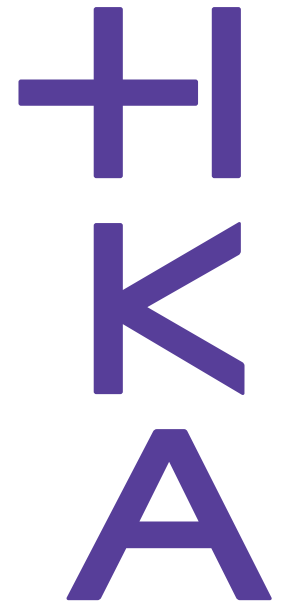
mystery File/Directory Structure

clmystery > mystery

"mystery" durchsuchen

Name	Änderungsdatum	Typ	Größe
interviews	16.04.2021 17:19	Dateiordner	
memberships	16.04.2021 17:19	Dateiordner	
streets	16.04.2021 17:19	Dateiordner	
crimescene	16.04.2021 17:19	Datei	417 KB
people	16.04.2021 17:19	Datei	219 KB
vehicles	16.04.2021 17:19	Datei	486 KB

 **starting point**



- `mystery/people$`

```
head mystery/people
```

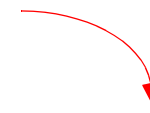
```
*****
```

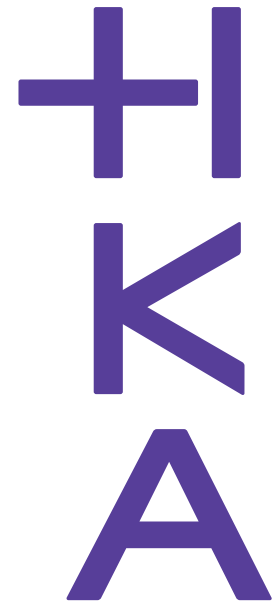
To go to the street someone lives on, use the file for that street name in the 'streets' subdirectory.

To knock on their door and investigate, read the line number they live on from the file. If a line looks like gibberish, you're at the wrong house.

```
*****
```

NAME	GENDER	AGE	ADDRESS
Alicia Fuentes	F	48	Walton Street, line 433
Jo-Ting Losev	F	46	Hemenway Street, line 390
...			
Annabel Fuglsang	M	40	Haley Street, line 176
Diego Michan	M	74	Wyola Place, line 25





- File `mystery/streets/Haley_Street`, lines 174 - 179

```
173 ...  
174 pinto simile fuzing pestering neutralized atriums  
daunted  
175 irradiates liquidates flimflams dispossessed  
176 SEE INTERVIEW #871877  
177 balmy metamorphosis nervier pilfered  
178 proofreaders steeping editorialized solutions
```

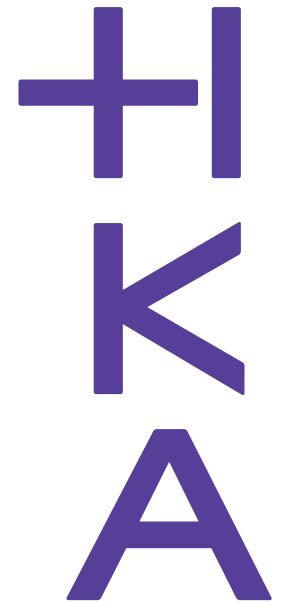
- Interviews:

```
$ ls mystery/interviews/interview-* | head -n5  
mystery/interviews/interview-000296  
mystery/interviews/interview-00448418  
...
```

▶ **mystery/interviews/interview-871877**

```
$ cat mystery/interviews/interview-871877  
Mr. Fuglsang is male and has brown hair ...
```

sort



- Sort lines of text files and/or lines from STDIN
- Write sorted concatenation of all FILE(s) to standard output.
- sorting alphabetic, numeric, ascending, descending, case (in)sensitive
- column(s)/bytes to be sorted can be specified
- Random sort option (-R)
- Remove of identical lines (-u)
- Examples:
 - sort the entries in file alphabetically
`sort member-list.txt`
 - sort the entries in file (Format: <first-name> <last-name>) by **second** column
`sort -t' ' -k2 member-list.txt`

field separator <space>

sort - examples

- sort file by country code, and as a second criteria population (numeric, descending)

```
sort -t, -k2,2 -k4,4nr city.csv
```

field separator: ,

numeric (-n), descending (-r)

second sort criteria from column 4 to column 4

first sort criteria from column 2 to column 2

- More example:

```
https://www.smiffy.de/KDIR-2022/command-examples/sort
```

Compare Operator



- comm - compare two **sorted** files line by line

Barcelona
Bern
Chamonix
Karlsruhe
Pisa
Porto
Rio

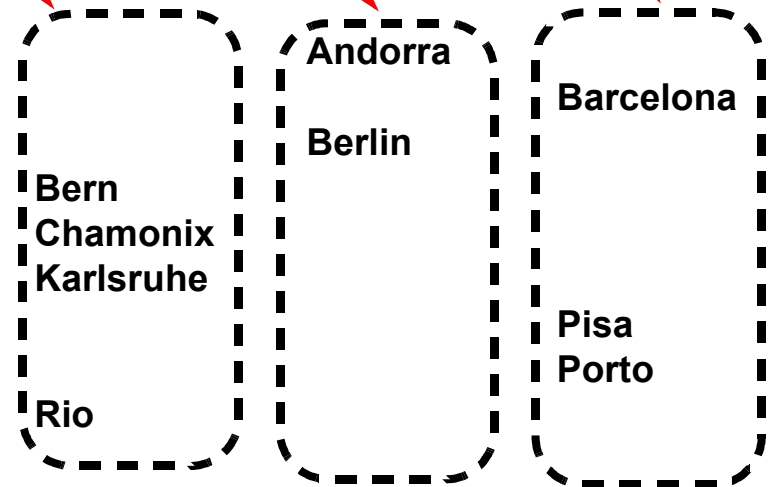
Andorra
Barcelona
Berlin
Pisa
Porto

comm

only in file1

only in file2

in file1
and file2

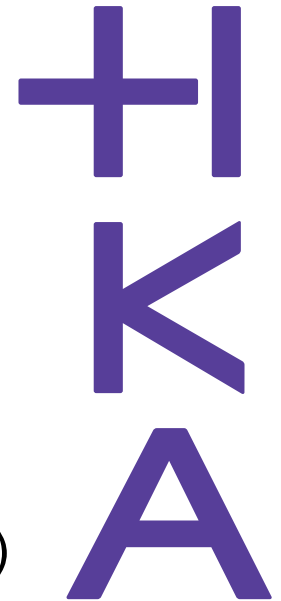


- Options:

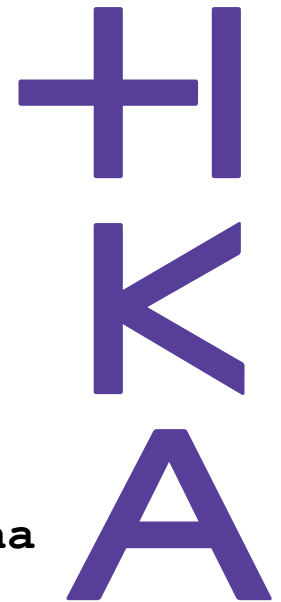
- -1: suppress column 1
- -2: suppress column 2

- -3: suppress column 3
- --total: output a summary

uniq



- report or omit **repeated** lines
- Filter adjacent matching lines from INPUT
- Range of comparison can be specified (first n chars, skip first m chars)
- options:
 - -c: count number of occurrences
 - -d: only print duplicate lines
 - -u: only print unique line
 - -i: ignore case



uniq - example

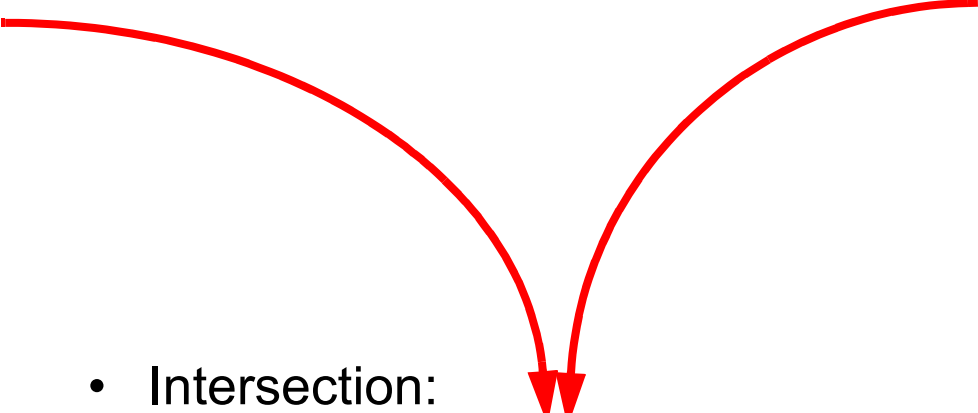
- file1.txt

Barcelona
Bern
Chamonix
Karlsruhe
Pisa
Porto
Rio

- file2.txt

Andorra
Barcelona
Berlin
Pisa
Porto

- Intersection:

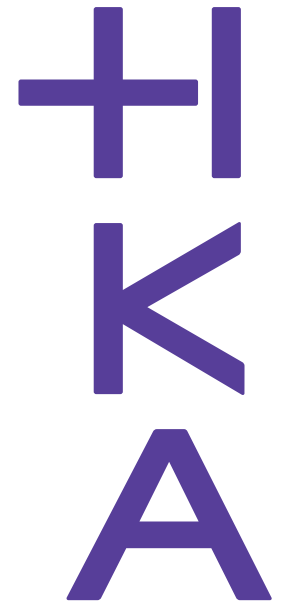


```
$ cat file*.txt | sort | uniq -d  
Barcelona  
Pisa  
Porto
```

- More example:

<https://www.smiffy.de/KDIR-2022/command-examples/comm,%20uniq>

Exercise Part II

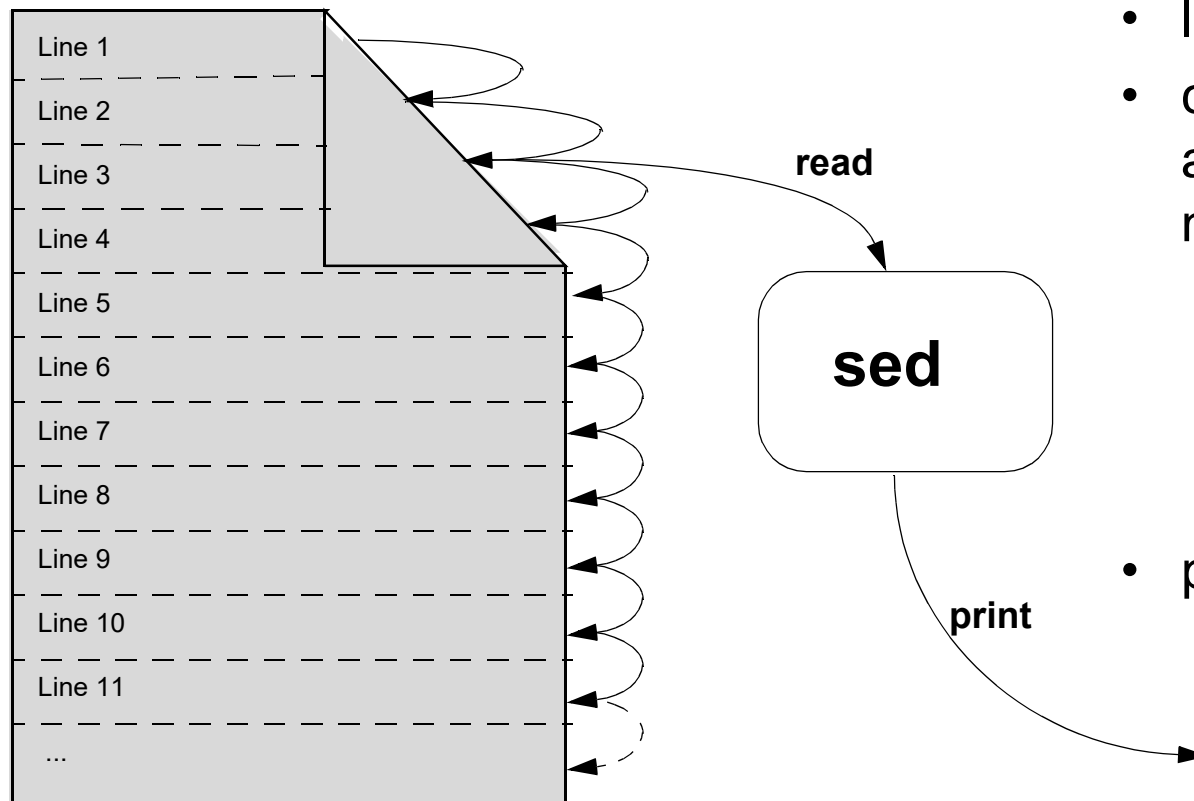


continue solving the case from the first exercise (Exercise II) ...

... still thrilling ;-)

(*) command line murders by Noah Veltman,
<https://github.com/veltman/clmystery>

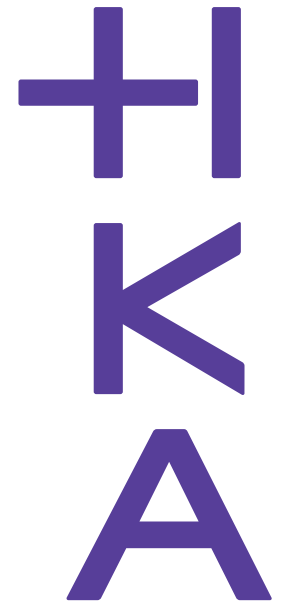
sed Principle



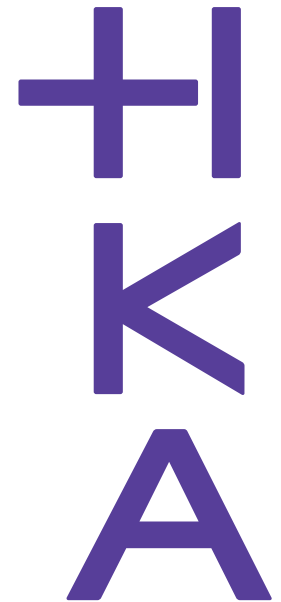
For every line:

- load line in pattern space
- check if optional address/address range matches
 - apply operation (insert, append, delete, substitute, change, print, ...)
on pattern space
- print content of pattern space

String Substitution with sed



- sed - Stream Editor
- non interactiv, controlled by a script
- line oriented text processing
- A loop executes script commands on each matching (by address) input line
- short scripts are typically given as parameter, longer scripts as files (-f option)
- Possible operations: Insert, Substitude, Delete, Append, Change, Print, Delete
- Commands in script can take an optional *address*, specifying the line(s) to be performed.
- *Address* can be a a single line number or a regular expression
- *Address* can be an interval (start, stop)
- Default behavior: printing each processed line to STDOUT (suppress with: -n)



sed commands

s: substitute

- Replace in every line the first occurrences of D with GER

```
sed 's/\bD\b/GER/' city.csv > city2.csv
```

- Replace **all occurrences** of NULL in a line with \N (**Inplace Substitution**)

```
sed -i 's/\bNULL\b/\\N/g' city.cs
```

- Replace „Stuttgart“ with „Stuttgart am Neckar“ (**extended regexp**)

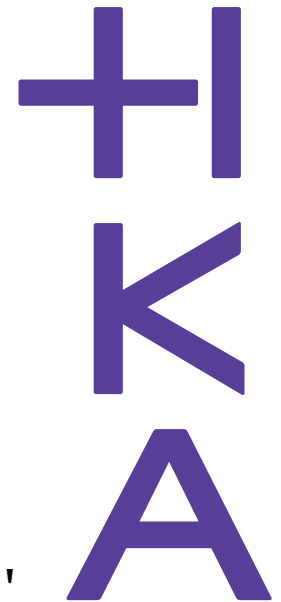
```
sed -E '/^Stuttgart/ s/^(^[,]+)/\1 am Neckar/' city.csv
```

p: print (typically used with default printing behaviour off (-n option))

- print from line 10 to 20 (resp.: 5-10, 23, 56-71)

```
sed -n 10,20p city.csv
```

```
sed -n '5,10p;23p;56,71p' city.csv
```



i: insert

- Insert dataset about Karlsruhe at line 2

```
sed '2i Karlsruhe,D,"Baden Wuerttemberg",312005,49.0,6.8'  
city.csv
```

d: delete

- delete the city Aachen (inplace)

```
sed -i '/^Aachen/ d' city.csv
```

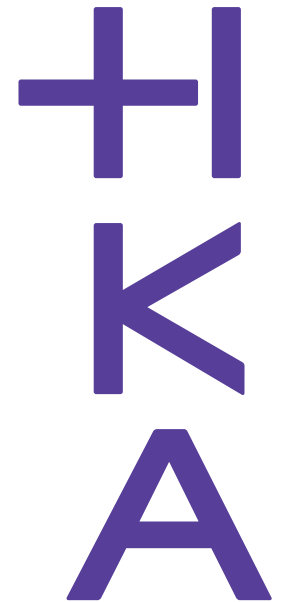
- delete all empty lines

```
sed '/^ *$/d' The-Adventures-of-Tom-Sawyer.txt
```

- delete lines 2-10

```
sed '2,10d' city.csv
```

sed Examples



c: change

- Replace entry of Biel

```
sed '/^Biel,/ c Biel,CH,BE,53308,47.8,7.14' city.csv
```

a: append

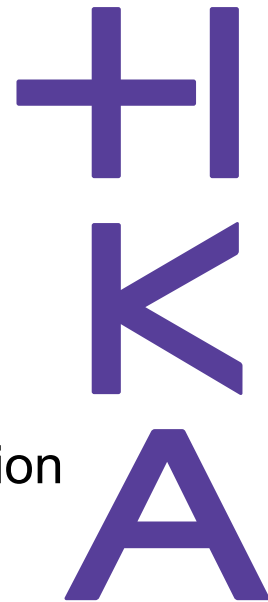
- Underline each CHAPTER

```
sed '/^CHAPTER/ a -----' The-Adventures-of-Tom-Sawyer.txt
```

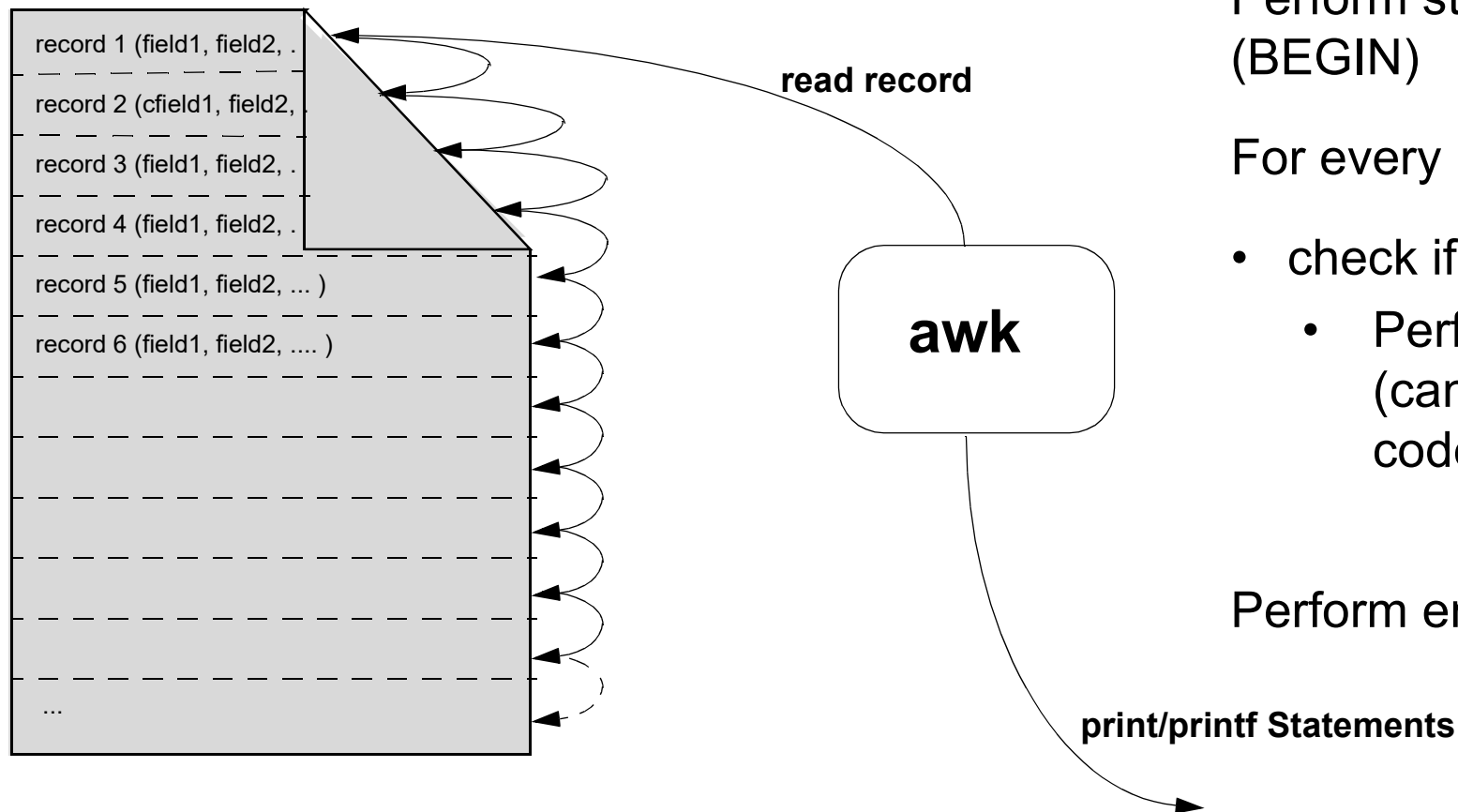
r: read file

- insert the content from file city-D.csv starting at line 3

```
sed '3 r city-D.csv' city.csv'
```



awk Principle

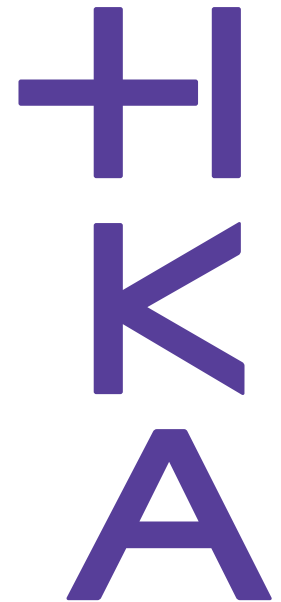


Perform start action
 (BEGIN)

For every record:

- check if condition holds
 - Perform action
 (can be any arbitrary
 code)

Perform end action (END)



awk

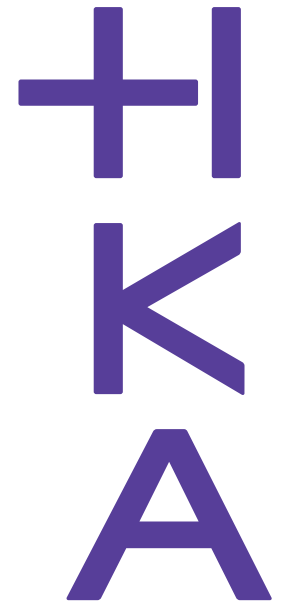
- like sed, but with powerful programming language
- filter and report writer
- flexible record definition (i.e. line with columns, record with fields, ...)
- full programming language, support for associative arrays
- structure: one or multiple *pattern* { *action* } blocks
- special BEGIN, END pattern match **before** the first record is read and **after** the last record is read
- Access to column values via \$1, \$2, ... variables (\$0: whole record)
- Examples:

```
awk -F, ' $3=="Bayern" && $4 < 1000000 { print $1, $4 }' city.csv
```

pattern

action

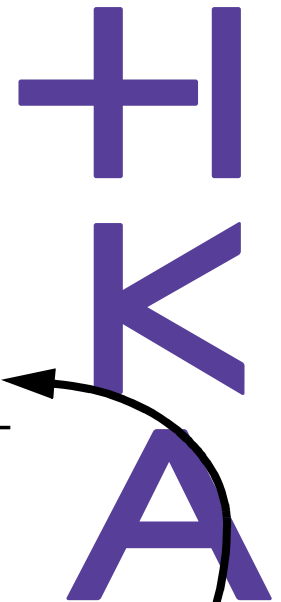
awk



- predefined variables
 - NF: number of fields
 - NR: number of records
 - FS: field separator (default: " ", same as -F from command line)
 - RS: record separator (default: \n)
 - ORS: output record separator
 - OFS: output field separator
 - FPAT: Field pattern (alternative way to specify a field instead of use of FS)
 - FILENAME: contains the file that is actually read

- More example:

`https://www.smiffy.de/KDIR-2022/command-examples/awk`



awk example: multi-line input

Andreas Schmidt
KIT
Germany

Manolo Diaz
IARIA
USA

Fritz Laux
University Reutlingen
Germany

Andreas Schmidt, KIT, Germany
Manolo Diaz, IARIA, USA
Fritz Laux, University Reutlingen, Ger-
many

Input

Output

cat **adres.txt** | awk -f **rec2csv.awk**

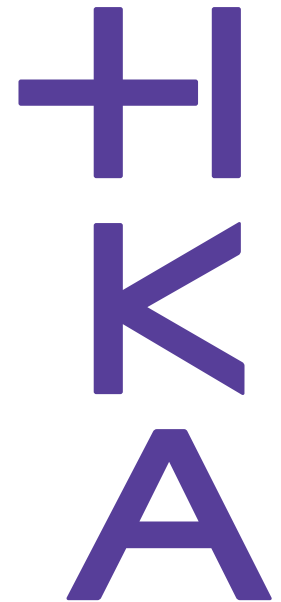
set input and
output separators

```
BEGIN {  
  FS="\n"  
  RS="\n\n"  
  OFS=","  
  ORS="\n"  
}
```

give awk a hint that
anything has changed

```
{  
  $1=$1  
  print $0  
}
```

print whole record



awk

- Calculating average population

```
awk -F, -f average.awk city.csv
```

```
# script: average.awk
```

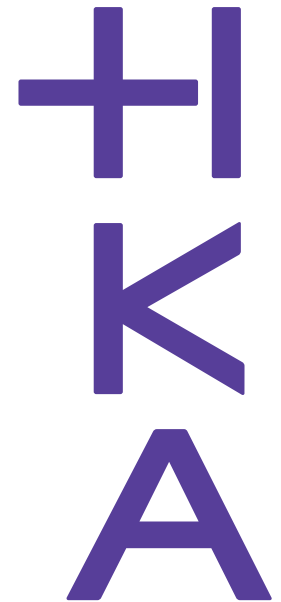
```
BEGIN { sum = 0  
        num = 0  
}
```

```
$4 != "NULL" {  
    sum += $4  
    num++  
}
```

```
END { print "Average population: "sum/num }
```

optional, because 0
is default value

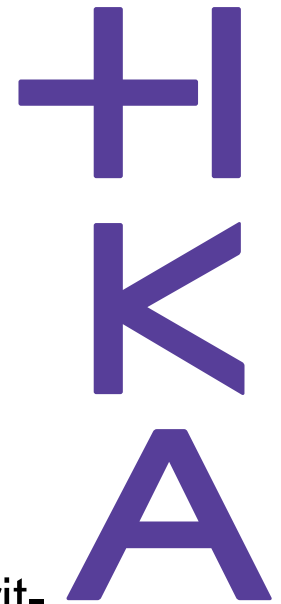
pattern



And again its time for crime* ...

As homework: You have solved the case, but there is room for improvement for future cases. So continue with Exercise III ...

(*) command line murders by Noah Veltman,
<https://github.com/veltman/clmystery>



Commands not Covered (not complete)

- **xargs**: build and execute command lines from standard input

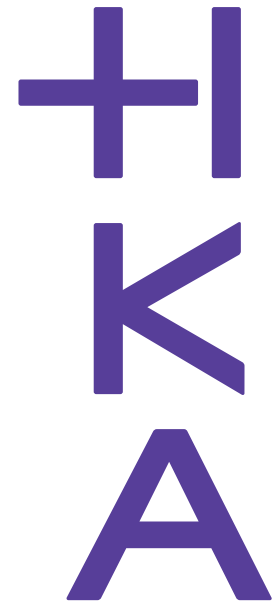
```
grep -l Agassi bbc sport/tennis/*.txt | \  
xargs grep -l Federer
```
- **tr**: translate, squeeze, and/or delete characters from standard input, writing to standard output.

```
tr 'A-Z' 'a-z' < moby-dick.txt
```
- **paste**: merge lines of files

```
paste -d', ' col1.txt col2.txt col3.txt > col_1-3.txt
```
- **find**: search for files in a directory hierarchy

```
find ./misc -name \*.txt -print
```
- **join**: join lines of two files on a common field

join Example



- city.csv

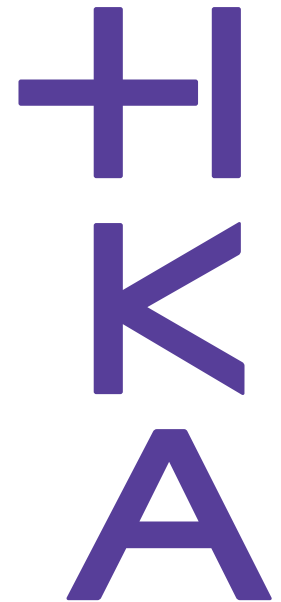
```
Aachen,D,"Nordrhein Westfalen",247113,NULL,NULL ...  
Aalborg,DK,Denmark,113865,10,57  
Aarau,CH,AG,NULL,NULL,NULL  
Aarhus,DK,Denmark,194345,10.1,56.1  
Aarri,WAN,Nigeria,111000,NULL,NULL  
...
```

- country.csv

```
Germany,D,Berlin,Berlin,356910,83536115  
Djibouti,DJI,Djibouti,Djibouti,22000,427642  
Denmark,DK,Copenhagen,Denmark,43070,5249632  
Algeria,DZ,Algiers,Algeria,2381740,29183032
```

```
sort -k2 -t, city.csv | join -t, -12 -22 - country.csv \  
-o1.1,2.1,1.3,1.4
```

```
Aachen,Germany,"Nordrhein Westfalen",247113  
Aalborg,Denmark,Denmark,113865  
Aarau,Switzerland,AG,NULL  
Aarhus,Denmark,Denmark,194345  
Aarri,Nigeria,Nigeria,111000  
Aba,Nigeria,Nigeria,264000  
Abakan,Russia,"Rep. of Khakassiya",161000
```



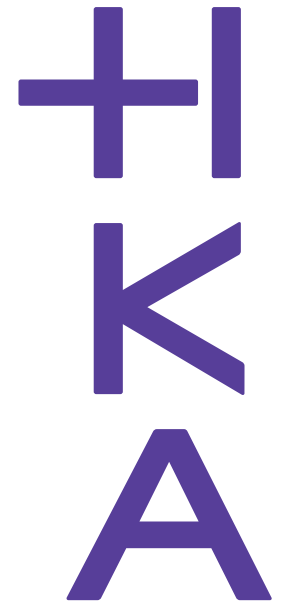
Commands not Covered (not complete)

- **split**: split file by number of rows/bytes/records
- **csplit**: split file at given patterns (include/exclude semantics)

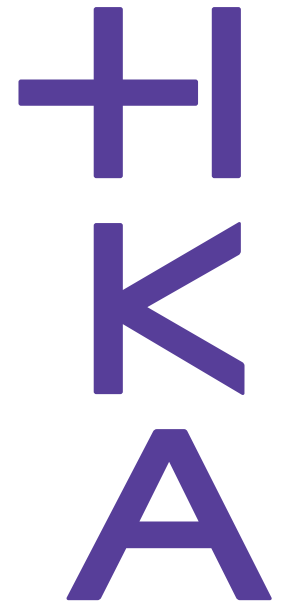
Examples:

<https://www.smiffy.de/KDIR-2022/command-examples/split>

Summary & Outlook



- Summary
 - Powerful filter and pipes architecture
 - Allows easy incremental development
 - Suitable for structured and unstructured data, ETL process
- Outlook
 - Utility make to handle dependencies between files
 - bash control flow elements like conditional execution, loops
 - bash functions
 - Seamless visualization using gnuplot
 - Easily extensible with own filters in any language



Further readings

- <http://www.theunixschool.com/p/awk-sed.html>
- Dale Dougherty, Arnold Robbins sed & awk, 2nd Edition UNIX Power Tools. O'Reilly, 2nd Edition 1997
- Arnold Robbins. Sed and Awk: Pocket Reference, 2nd Edition Paperback – June , O'Reilly, 2002
- Ramesh Natarajan. sed and awk 101 hacks. <http://www.thegeekstuff.com/sed-awk-101-hacks-ebook/>
- gnuplot homepage: <http://www.gnuplot.info/>