

OperationsExplorer

Daten, Technologien, Tests

Meik Bittkowski

Hinweis

- Wir sind noch in der **geschlossenen Beta-Phase**
- **Veröffentlichung** des OperationsExplorers im **Frühjahr 2014**
- Dies ist eine also **Vorschau** auf ein Tool, das sich noch in **Entwicklung** befindet

Übersicht

- Über das Projekt
- Daten, Daten, Daten
- Architektur des OperationsExplorer
- Vorbereitung der Daten
- Von den Daten zur Karte: ein Schnelldurchlauf
- Testen

Über das Projekt

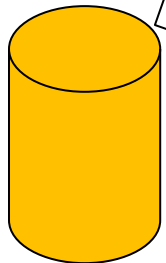
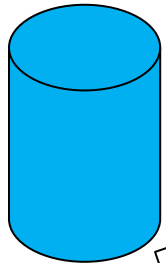
- **Wer erkrankt wo? – Werkzeug für Daten-Journalisten in der Medizin**
- **Gemeinsames Projekt** von Volker Stollorz und der *Scientific Databases and Visualization* Gruppe am Heidelberger Institut für Theoretische Studien (HITS)
- Idee zur Zusammenarbeit entstand während Volkers Aufenthalt am HITS als ***Journalist in Residence***
- Gefördert im Programm “Neue Wege im Wissenschaftsjournalismus” der **Robert Bosch Stiftung**

Daten, Daten, Daten

- **ICD-10** (4-Steller) Diagnosen nach Wohnort der Patienten (Kreis), Geschlecht und Altersgruppe
 - 2009 bis 2011
 - Umfang: je ca. 6 Mio. Zeilen
- **OPS** (4-Steller) Schlüssel nach Wohnort der Patienten (Kreis), Geschlecht und Altersgruppe
 - 2011 (2009 und 2010 werden noch erworben)
 - Umfang: je ca. 5,6 Mio. Zeilen
- **Bevölkerungsdaten** für die Kreise nach Alter
 - 2009 bis 2011
 - Umfang: je ca. 42 000 Zeilen
- **Taxonomien** für ICD-10 bzw. für OPS
 - Umfang: ca. 380 000 bzw. ca. 180 000 Zeilen XML
- **Verwaltungskarte** von Deutschland, Bundesländer und Kreise

Architektur des OperationsExplorer

Datenbank: Metadaten, Bevölkerungsdaten



Server / Backend

Grails
Groovy
SolrJ
Apache Pig
Spock

Suchanfragen
Datenverwaltung

Browserfenster / Frontend

HTML/CSS
Bootstrap
Javascript
D3.js
TopoJSON
Geostats.js
AngularJS

Darstellung
Interaktion

Suchserver:

Fallzahlen, Taxonomien

Vorbereitung der Daten

- **Taxonomien**

- Relevante Informationen (Codes + Label) aus den XML-Dateien **extrahieren**
- Für die Suche **indexieren**
- Metadaten anlegen

- **Bevölkerungsdaten**

- Datenbankeinträge erzeugen
- Metadaten anlegen
- Cache erneuern

- **Karte**

- **Vereinfachung der Geometrie**, so dass die Karte kleiner wird und schneller im Browser dargestellt werden kann. (Von ca. **6MB auf 418KB**)
- Verknüpfung der Geometrie mit den Kreisschlüsseln

Vorbereitung der Daten

- OPS / ICD10 Fallzahlen
 - In ein definiertes Zwischenformat überführen
 - Automatisch **Daten aggregieren** (Geschlecht “beide”)
 - Altersgruppen auf die in der Standardpopulation benutzten **Altersgruppen abbilden**
 - Berücksichtigung von **Kreisreformen**
 - Fallzahlen in **Standardpopulation** berechnen
 - Daten für die Suche **indexieren**
 - Jedes Datum wird als eine Art “**Minidokument**” betrachtet, das **effizient verschlagwortet** wird
 - Ermöglicht **sehr schnelle Suche** nach allen “Minidokumenten” für bspw. eine Diagnose, eine Altersgruppe und ein Geschlecht
 - Metadaten anlegen

Vorbereitung der Daten

- Administrationsinterface:

OperationExplorer Administration

ICD-10/OPS Fallzahlen

Bevölkerungsdaten

Taxonomien

ICD-10/OPS Fallzahlen

Vorhandene Datensätze

Bezeichner	Datentyp	Jahr	Dateiname	Vorverarbeitet	Indexiert	Bevölkerungsdaten
icd10 fallzahlen für 2009	ICD-10	2009	icdhd4_AG_GE_Kreis_2009.csv	✓	✓	✓
icd10 fallzahlen für 2010	ICD-10	2010	icd10hd4_AG_GE_Kreis_2010.csv	✓	✓	✓
icd10 fallzahlen für 2011	ICD-10	2011	icd10hd4_AG_GE_Kreis_2011.csv	✓	✓	✓
ops fallzahlen für 2011	OPS	2011	Kreis_AG_GE_OPS4_2011.csv	✓	✓	✓

Neuen Datensatz erstellen

Bezeichner

Datentyp

Jahr

Format

Daten Keine Datei ausgewählt.

Bereit für die Daten aus 2012!

Von den Daten zur Karte: ein Schnelldurchlauf

Verfügbare Datensätze

ICD-10

ICD-10 Code auswählen:

Code / Beschreibung
k70
K70-K77 (Krankheiten der Leber)
K70 (Alkoholische Leberkrankheit)
K70.0 (Alkoholische Fettleber)
K70.1 (Alkoholische Hepatitis)
K70.2 (Alkoholische Fibrose und Sklerose der Leber)

- “k70” wird an den **Server** geschickt
- Suche in den indexierten **Taxonomiedaten**
- Die **zehn besten Vorschläge** werden zurück an den **Browser** gesendet

Von den Daten zur Karte: ein Schnelldurchlauf

Verfügbare Datensätze

ICD-10

ICD-10 Code(s) auswählen:

× K70 (Alkoholische Leberkrankheit)

Altersgruppe auswählen*:

Alle

= ≤ ≥

Geschlecht auswählen:

Frauen und Männer

Erhebungsjahr auswählen:

2009

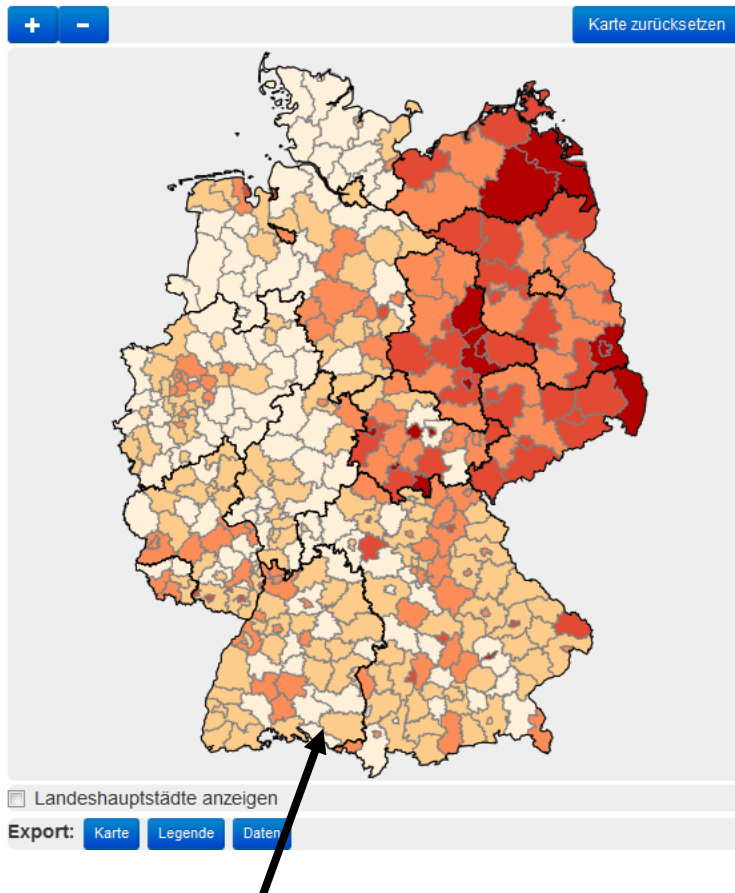
Intervall auswählen

- “K70” ausgewählt
- **Erweiterung** des Codes zu “K70*” (**Wildcard** für beliebige Zeichen)
- **Suchanfrage** mit den ausgewählten **Parametern** wird an den **Server** geschickt

Von den Daten zur Karte: ein Schnelldurchlauf

- Auf dem **Server**:
 - **Suchanfrage** an den **Suchserver** mit den “Minidokumenten”
 - Liefert (**altersstandardisierte**) **Anzahl der Fälle** für das ausgewählte Geschlecht und die ausgewählte Altersgruppe im ausgewählten Jahr
 - **Suchanfrage** an die **Datenbank** mit den Bevölkerungsdaten
 - **Berechnung der Bevölkerung** für das ausgewählte Geschlecht und die ausgewählte Altersgruppe im ausgewählten Jahr
 - **Antwort an den Browser**: Zuordnung von Einwohnerzahlen, Fallzahlen und (altersstandardisierten) Fallzahlen je 100 000 Einwohner zu jedem Kreis über den Kreisschlüssel

Von den Daten zur Karte: ein Schnelldurchlauf



Jeder Kreis ist ein **SVG-Linienzug**, der über den Kreisschlüssel angesprochen werden kann.

- Im Browser:
 - Berechnung der Klassen für die Einfärbung (Jenks-Caspall-Algorithmus)
 - Jede Klasse bekommt über eine Palette von Farben (<http://colorbrewer2.org/>) eine Farbe zugewiesen
 - Alle Kreise werden neu eingefärbt

Testen

- Arbeitet der OperationsExplorer auch **korrekt**?
- **Wann** arbeitet der OperationsExplorer **korrekt**?
- **Manuell Testfälle** für einzelne Codes und Kreise ausarbeiten (so wie man es halt in **Excel** machen würde...)
- **Automatische Tests** für diese Testfälle zum OperationsExplorer hinzufügen

Testen: Ein Testfall in Excel

	A	B	C	D	E	F	G	H	I	J	K	L
1	Kreis	Kreisbezeichner	Altersgruppe	Total	M	W	Standard Pop	Standard Pop. / Total	Standard Pop. / M	Standard Pop. / W	K703 Total	K703 in Total Standard Pop.
62	8221	Heidelberg, Kreisfreie Stadt	50 bis unter 51 Jahre	2119	1021	1098		0	0	0		0
63	8221	Heidelberg, Kreisfreie Stadt	51 bis unter 52 Jahre	1981	1013	968		0	0	0		0
64	8221	Heidelberg, Kreisfreie Stadt	52 bis unter 53 Jahre	1930	936	994		0	0	0		0
65	8221	Heidelberg, Kreisfreie Stadt	53 bis unter 54 Jahre	1843	938	905		0	0	0		0
66	8221	Heidelberg, Kreisfreie Stadt	54 bis unter 55 Jahre	1817	958	859		0	0	0		0
67		AG 11		9690	4866	4824	7000	0.722394221	1.438553226	1.451077944	10	7.223942208
68	8221	Heidelberg, Kreisfreie Stadt	55 bis unter 56 Jahre	1763	867	896		0	0	0		0
69	8221	Heidelberg, Kreisfreie Stadt	56 bis unter 57 Jahre	1700	838	862		0	0	0		0
70	8221	Heidelberg, Kreisfreie Stadt	57 bis unter 58 Jahre	1580	754	826		0	0	0		0
71	8221	Heidelberg, Kreisfreie Stadt	58 bis unter 59 Jahre	1539	705	834		0	0	0		0
72	8221	Heidelberg, Kreisfreie Stadt	59 bis unter 60 Jahre	1594	797	797		0	0	0		0
73		AG 12		8176	3961	4215	6000	0.733855186	1.514768998	1.423487544	5	3.66927593
74	8221	Heidelberg, Kreisfreie Stadt	60 bis unter 61 Jahre	1553	767	786		0	0	0		0
75	8221	Heidelberg, Kreisfreie Stadt	61 bis unter 62 Jahre	1424	696	728		0	0	0		0
76	8221	Heidelberg, Kreisfreie Stadt	62 bis unter 63 Jahre	1372	664	708		0	0	0		0
77	8221	Heidelberg, Kreisfreie Stadt	63 bis unter 64 Jahre	1186	582	604		0	0	0		0
78	8221	Heidelberg, Kreisfreie Stadt	64 bis unter 65 Jahre	1003	491	512		0	0	0		0
79		AG 13		6538	3200	3338	5000	0.764759865	1.5625	1.497902936	5	3.823799327
80	8221	Heidelberg, Kreisfreie Stadt	65 bis unter 66 Jahre	1428	700	728		0	0	0		0
81	8221	Heidelberg, Kreisfreie Stadt	66 bis unter 67 Jahre	1418	693	725		0	0	0		0
82	8221	Heidelberg, Kreisfreie Stadt	67 bis unter 68 Jahre	1257	590	667		0	0	0		0
83	8221	Heidelberg, Kreisfreie Stadt	68 bis unter 69 Jahre	1460	696	764		0	0	0		0
84	8221	Heidelberg, Kreisfreie Stadt	69 bis unter 70 Jahre	1521	713	808		0	0	0		0
85		AG 14		7084	3392	3692	4000	0.564652739	1.179245283	1.083423619	2	1.129305477
86	8221	Heidelberg, Kreisfreie Stadt	70 bis unter 71 Jahre	1451	684	767		0	0	0		0
87	8221	Heidelberg, Kreisfreie Stadt	71 bis unter 72 Jahre	1355	634	721		0	0	0		0
88	8221	Heidelberg, Kreisfreie Stadt	72 bis unter 73 Jahre	1155	540	615		0	0	0		0
89	8221	Heidelberg, Kreisfreie Stadt	73 bis unter 74 Jahre	1066	467	599		0	0	0		0
90	8221	Heidelberg, Kreisfreie Stadt	74 bis unter 75 Jahre	1154	493	661		0	0	0		0
91		AG 15		6181	2818	3363	3000	0.485358356	1.064584812	0.892060666	5	2.426791781
92	8221	Heidelberg, Kreisfreie Stadt	75 bis unter 80 Jahre	4058	1701	2357		0	0	0		0
93		AG 16		4058	1701	2357	2000	0.492853622	1.175778954	0.848536275	2	0.985707245
94	8221	Heidelberg, Kreisfreie Stadt	80 bis unter 85 Jahre	3497	1266	2231		0	0	0		0
95		AG 17		3497	1266	2231	1000	0.285959394	0.789889415	0.448229494	1	0.285959394
96	8221	Heidelberg, Kreisfreie Stadt	85 Jahre und mehr	3500	911	2589		0	0	0		0
97		AG 18		3500	911	2589	1000	0.285714286	1.097694841	0.386249517		0
98												
99	8221	Heidelberg, Kreisfreie Sta	Insgesamt	146466	69258	77208	100000				41	26.52299694

Datenquelle: Statistisches Bundesamt, Fallbezogene Krankenhausstatistik (DRG Statistik 2009), Bevölkerungsstand 2009



Testen: Ein Testfall in Grails/Spock

```
def "search for K70.3 incidents, 2009, both genders, all age groups in heidelberg"(){
  when:
  def response = informationCenterService.getIncidentsForQuery(year, yearTo, searchYearRange, maennlich, altersStufe,
    altersStufenOperator, opSchluessel)

  then:
  response.incidents['08221'].anzahlOps as Integer == 1 + 1 + 1 + 3 + 3 + 6 + 2 + 7 + 1 + 1 + 5 + 1 + 2 + 2 + 4 + 1 // = 41, data from csv
  df.format(response.incidents['08221'].opPer100k as Double) == df.format(response.incidents['08221'].opPerStandardPop as Double)
  df.format(response.incidents['08221'].opPerStandardPop as Double) == df.format(26.5229969448) // data from spreadsheet

  where:
  year = 2009
  yearTo = 2009
  searchYearRange = "false"
  maennlich = "both"
  altersStufe = 22
  altersStufenOperator = "equals"
  opSchluessel = "K703"
}
```


Testen

- Zur Zeit **43 Tests**, die alle **erfolgreich** laufen
- Wem das nicht reicht: wir werden die Software als **Open Source** veröffentlichen. Dann kann jeder, der will, **weitere Tests** schreiben! Wir werden das auch tun.



Vielen Dank!

DATENQUELLE: STATISTISCHES BUNDESAMT,
FALLBEZOGENE KRANKENHAUSSTATISTIK (DRG STATISTIK 2009-2011),
BEVÖLKERUNGSSTAND 2009-2011
KARTENMATERIAL: © GEOBASIS-DE / [BKG](#) 2013

GEFÖRDERT VON DER **ROBERT BOSCH STIFTUNG**

