

Работа с таможенными данными

- Материалы и инструменты
- Метаданные
- Пытаемся получить данные
- Изготавливаем CSV-выборку
- Рассматриваем CSV-выборку в LibreOffice

Формат данных

- Данные предоставляются в формате DBF (1979 год, Ashton-Tate под CP/M, <https://ru.wikipedia.org/wiki/DBF>)
- Внутри dbf-файлов кодировка cp866
- Нам нужны данные в CSV в кодировке Unicode
- Полный набор данных занимает 2 Gb
- Для плохо влезающих в оперативную память данных придуман термин **BIG DATA**

LibreOffice

- Офисные программы не рассчитаны на **Bigdata**
- 10 тысяч строк удобны для работы,
но уже 100 тысяч — тяжело обрабатывать,
миллион — практически невозможно
- Данные таможни — это около
13 миллионов строк за неполных четыре года
- Чтобы с ними работать, нужна выборка

Выбор инструмента

- Нам нужно получить .csv файлы которые можно импортировать в офисный пакет
- Мы это сделаем в командной строке Ubuntu
- Его рекомендует Microsoft:
<https://msdn.microsoft.com/en-us/commandline/wsl/about>
<https://xakep.ru/2017/02/03/wsl-in-colors/>
- В OSX нужен Homebrew <https://brew.sh/> и установка дополнительных программ руками

Математическое обеспечение

- **curl** – загрузчик URL
- **bzip2** – архиватор
- **libdbd-xbase-perl** – конвертер .dbf в .csv
- **dbfinfo** из библиотеки **shapelib**
- **iconv** – перекодировщик (есть в системе)
- Устанавливаем программы при помощи **sudo**:
**sudo apt install -y **
curl libdbd-xbase-perl curl bzip2 shapelib

Получение метаданных

- URL: <http://stat.custom.ru>
«Выгрузка данных» → «Параметры»
 - «Справочник по ТНВЭД»
 - «Справочник по странам»
 - «Справочник по единицам измерения»
 - «Справочник субъектов РФ в федеральном округе»
 - «Справочник федеральных округов»
- Вводим код с картинки (она же САРТСНА)
- Загружаем метаданные

Метаданные в формате DBF

Получаем следующие файлы в формате DBF:

- THBED.dbf Справочник по ТНВЭД
- СТРАНА.dbf Справочник по странам
- ED.dbf Единицы измерения
- SUBRF.dbf Субъекты РФ
- FO.dbf Федеральные округа

Метаданные: dbfinfo

Info for CTPAHA.dbf 2 Columns, 254 Records in file

KOD string (3,0)

NAME string (35,0)

Info for ED.dbf 3 Columns, 30 Records in file

KOD string (3,0)

NAME string (40,0)

SHORT_NAME string (20,0)

Info for FO.dbf 2 Columns, 10 Records in file

OKATO_1 string (2,0)

OKATO_1_N string (255,0)

Info for SUBRF.dbf 2 Columns, 92 Records in file

OKATO_2 string (5,0)

OKATO_2_N string (255,0)

Info for THBED.dbf 2 Columns, 30736 Records in file

KOD string (10,0)

SIMPLE_NAM string (255,0)

Получаем данные ТСВТ

- Период «все года» или «год» не даёт месяца, скачиваем поквартально
- На 2017-10-10 последние данные — за июль, видимо июль неполный
- Нет параллельной загрузки
- Нет фиксированных URL
- Всё это занимает очень много времени

Данные: dbfinfo TCBT.dbf

Info for TCBT.dbf

10 Columns, 3531862 Records in file

NAPR	string	(2,0)
PERIOD	string	(4,0)
STRANA	string	(3,0)
TNVED	string	(10,0)
EDIZM	string	(20,0)
STOIM	float	(22,0)
NETTO	float	(22,0)
KOL	float	(22,0)
REGION	string	(255,0)
REGION_S	string	(255,0)

Получение ТСВТ по месяцам

- URL: <http://stat.custom.ru>
- «Выгрузка данных» →
- Вид выгрузки «Данные ТСВТ» →
- Уровень ТНВЭД «10 знаков» →
- Период «квартал (разбивка по месяцам)» →
- Федеральный округ «все» →
- Субъект РФ в федеральном округе «все» →
- Для получения данных введите код с картинки

Обработка данных

- Не смогли выкачать данные? Возьмите готовые!
<http://black.pu.ru/ТСВТ/>
- у нас есть много DBF-файлов, обрабатываем их скриптом на shell, который:
 - bunzip2 разжимает .bz2 → .dbf
 - dbf_dump преобразует .dbf → .csv
 - iconv перекодировывает cp866 → unicode

Команды shell

- # распаковать файлы *.dbf.bz2

```
for FILE in *.dbf.bz2; do
```

```
    bunzip2 $FILE
```

```
done
```

```
# преобразовать *.dbf в *.csv
```

```
for DBF in *.dbf; do
```

```
    CSV=`echo $DBF | sed "s/.dbf/.csv/"`
```

```
    dbf_dump $DBF | \
```

```
    iconv -f cp866 -t utf8 > $CSV
```

```
done
```

Получение TCBT.csv.bz2

- У нас есть данные в формате CSV за 2014-2017 годы.
- Получаем единый TCBT.csv:
`cat 201*.csv > TCBT.csv`
- Размером 2.3Gb. Сожмём:
`bzip2 -9 TCBT.csv`
- Получили файл `TCBT.csv.bz2` размером 161мегабайт. Он есть на нашем зеркале.

Создание выборки

- Для создания выборки используется `grep`
<https://ru.wikipedia.org/wiki/Grep>
- `grep` написал Кен Томпсон в 1974 году
- Умеет быстро печатать строки с образцом в очень большом текстовом файле данных
`grep образец данные.csv > выборка.csv`
- Можно указать несколько файлов данных
- `grep` умеет работать через трубу (pipeline, оно же – `|`)

grep метаданных

- В массиве данных всё написано БОЛЬШИМИ русскими буквами, не путайте „С“ латинскую и „С“ кириллическую!
- Ищем сыр:
grep СЫР THBED.csv
- Рассматриваем результат, там много разного
- Обнаруживаем товарную группу 0406
- Творог это тоже сыр, ищем:
grep ТВОРОГ THBED.csv

Выборка данных grep

- Используем распакованный файл
`grep ":0406" TCBT.csv | grep "^ИМ:01/" > jan.csv`
Первый grep отдаёт в «трубу» сыр, второй выбирает январские сыры: 1107 сыров.
- Используем запакованный файл
`bzcat TCBT.csv.bz2 | grep ":0406" | \`
`grep "^ИМ:03/" > mar.csv`
Первый grep, пропускает сыры, второй мартовские.
- Проблема: кроме сыров могут пройти любые данные, начинающиеся на 406. Это допустимо при дальнейшей обработке в LibreOffice.

Выборка данных awk

- awk — продвинутая программа обработки текстов, перебирает строку за строкой:
`bzcat TCBT.csv.bz2 | awk -F: \`
`'$1~/ИМ/ && $2~/^03/ && $4 ~ /^0406/ {print}'`
- ключ -F разделитель полей. Если условие для строки истинно, то выполняем **print**.
- Условие: поле 1 содержит ИМ, поле 2 начинается на 03, а поле 4 на 0406.
- Вывод команды посылаем в > `file.csv`

Импорт данных в LibreOffice

- По сути очень просто
- Тонкости: цифровые поля импортируем как стандарт США (с точкой)
- Формат времени вида ММ/YYYY прописываем после импорта
- Перед импортом.csv с миллионом записей сохраняемся