

Актуарный семинар
Общество актуариев Казахстана
и Ассоциация профессиональных актуариев

Инструменты Актуария НПФ

Петр Тамбовский

СРО актуариев “Ассоциация профессиональных актуариев”

Алматы-Москва, 18 декабря, 2024 год

Содержание

- Демонстрация программы демографического прогноза
- Архитектуры расчетов:
 - Архитектура расчетов средствами MS-Office (VBA-Excel)
 - Архитектура расчетов средствами MS-Office (VBA-Access)
 - Архитектура расчетов средствами *Python-Excel*
 - Архитектура расчетов средствами *Python-Access*
- Пример web-скрапинга
 - Библиотеки BeautifulSoup, Requests
 - Регулярные выражения (Regex)
- Другие примеры Google Colab
- Примеры расчета денежных потоков

Проблема – увеличение информационных массивов

Массив ИД для АЗ 2010 года до 50 МБ

Массив ИД для АЗ 2024 года более 500 МБ

Расчеты становятся более объемными ввиду увеличения количества участников и застрахованных лиц, усложнения самих расчетов (ежемесячные потоки)

Архитектура расчетов средствами MS-Office (VBA-Excel)

```
Worksheets("ID").Activate
```

‘обращение к листу “ID”

```
i=2
```

```
While Worksheets("ID").Cells(i, 1) <> ""
```

‘пример считывания значения ячейки на листе “ID” в переменную

```
AgeC
```

```
AgeC = Cells(i, 1).Value
```

```
...
```

```
i=i+1
```

```
Wend
```

| Дата рождения |
|---------------|
| 24.06.1941 |
| 16.04.1950 |
| 21.03.1947 |
| 18.08.1947 |
| 24.07.1934 |
| 04.08.1947 |
| 15.07.1945 |
| 03.07.1938 |
| 15.12.1934 |
| 30.03.1935 |

Как сейчас (SSM, VBA) уже сложно

Существенное усложнение алгоритмов затрудняет использование формул в ячейках, а учитывая объем данных в отдельных случаях становится невозможным.

Использование конструкции VBA-Excel ограничено объемом данных.

Архитектура расчетов средствами MS-Office (VBA-Access)

'***** переменные

Dim Tb As Recordset

Dim DBS As Database

'BaseName.mdb - название базы данных

'TableName- имя таблицы Access

Set DBS = DAO.OpenDatabase("BaseName.mdb") SQLr =

"SELECT * FROM " + TableName

Set Tb = DBS.OpenRecordset(SQLr)

'выполняем цикл до последней записи

Do While Not Tb.EOF

'пример считывания значения поля "Дата рождения" из таблицы "TableName" БД ("BaseName.mdb") в переменную AgeC

AgeC = Tb.Fields("Дата рождения")

Tb.MoveNext

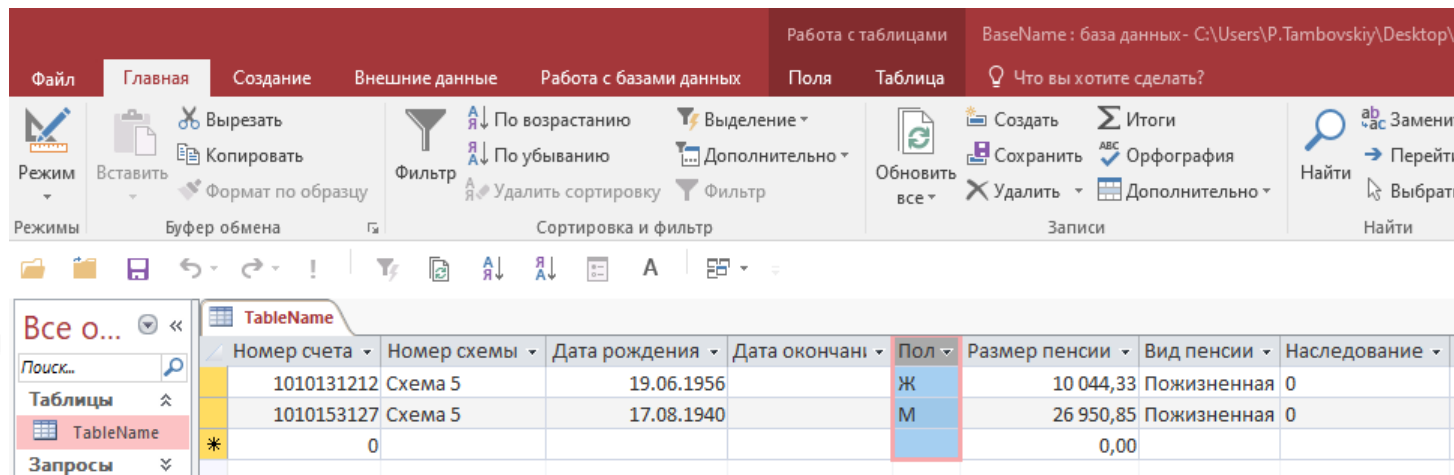
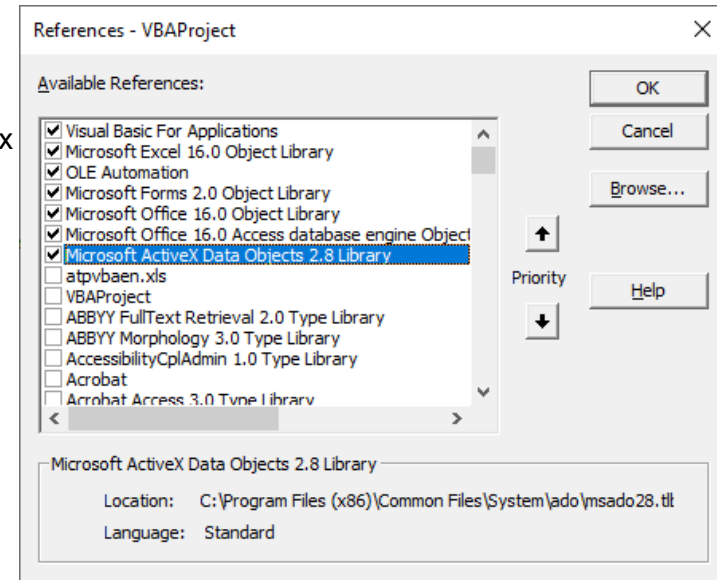
Loop

Tb.Close

Set Tb = Nothing

DBS.Close

Set DBS = Nothing



ADOdb - это программная библиотека, обеспечивающая прикладной интерфейс доступа к базам данных основанная на некоторых концепциях Microsoft ActiveX Data Objects.

Библиотека позволяет работать со следующими СУБД:

Access

ActiveX Data Objects

DB2

Microsoft SQL Server

MySQL

Oracle Database

PostgreSQL

любые СУБД, поддерживающие ODBC

ADOdb использует SQL.

Создание электронных библиотек (КХД) и рост информационных объемов, приводит к снижению производительности клиент серверной архитектуры (VBA-Access). На практике наблюдаются ограничения, при превышении объема информации 2 Гб.

Какие возможные пути решения?

Свободное программное обеспечение

Возможно следующее решение:

MySQL - свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle.

MAMP - это набор решений, состоящий из **бесплатного, открытого и коммерческого программного обеспечения**. Система управления БД – MySQL, MariaDB.

MariaDB - это разработанная сообществом, коммерчески поддерживаемая ветвь системы управления реляционными базами данных MySQL (СУБД), предназначенная для того, чтобы **оставаться свободным программным обеспечением с открытым исходным кодом**.

MySQL Workbench — инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД для системы баз данных MySQL

Архитектура расчетов средствами MS-Office (VBA-MySQL)

***** переменные

```
Dim strSql As String
```

```
Dim rs As Object
```

```
Dim cn As Object
```

```
Set cn = New ADODB.Connection
```

```
cn.Open "DSN=datasql;Uid=datasql;Pwd=datasql"
```

‘DSN – имя сервера базы данных, Uid – имя пользователя, Pwd – пароль

```
strSql = "Select * from TableName" ‘TableName - имя таблицы в базе MySQL
```

```
Set rs = New ADODB.Recordset
```

```
rs.Open strSql, cn
```

```
Do Until rs.EOF
```

‘пример считывания значения поля “Age” из таблицы “TableName” в переменную AgeC

```
AgeC = rs!Age
```

```
.....
```

```
rs.MoveNext
```

```
Loop
```

```
Set rs = Nothing
```

```
cn.Close
```

```
Set cn = Nothing
```

Архитектура расчетов средствами (Python-Excel)

```
import openpyxl
```

```
workbook = openpyxl.load_workbook('C:/Data/Data.xlsx')
```

```
worksheet=workbook["in"]
```

```
n=worksheet.max_row
```

```
j=2
```

```
while j<=n:
```

```
    k="A"+str(j)
```

#пример считывания значения из ячейки A2 листа "in", файла "Data.xlsx" в переменную AgeC

```
    AgeC=worksheet[k].value
```

```
    j=j+1
```

Архитектура расчетов средствами (Python-Access)

```
import pyodbc
```

```
con_string = 'DRIVER={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=C:/ BaseName.mdb;'
conn = pyodbc.connect(con_string)
```

```
cur = conn.cursor()
```

```
cur.execute('SELECT * FROM TableName')
for row in cur.fetchall():
```

#пример считывания значения 1-го столбца из таблицы "TableName" в переменную AgeC

```
AgeC=row[1]
```

```
.....
cur.close()
```

Что такое веб-скрапинг?

Web scraping — это процесс извлечения информации из Интернета.

Когда говорим «извлечение информации» подразумевается именно автоматизация этого процесса.

Зачем это нужно?

Собирать данные с различных площадок для анализа и дальнейшего взаимодействия с ними!

HTTP — это протокол клиент-серверного взаимодействия, позволяющий получать различные ресурсы, например, HTML-документы (веб-страница). Протокол HTTP лежит в основе обмена данными в интернете.

Когда мы открываем интернет-страницу – то, что мы видим это заслуга браузера.

HTML — стандартизированный язык разметки документов в интернете. Большинство веб-страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами; полученный в результате интерпретации текст отображается на экране монитора компьютера или мобильного устройства.

HTML представляет собой основу внутренней структуры сайта, его базовый каркас. С его помощью браузер выстраивает интернет-страницу в виде, который понятен для людей.

Библиотеки Beautiful Soup и Requests



Библиотеки Beautiful Soup и Requests

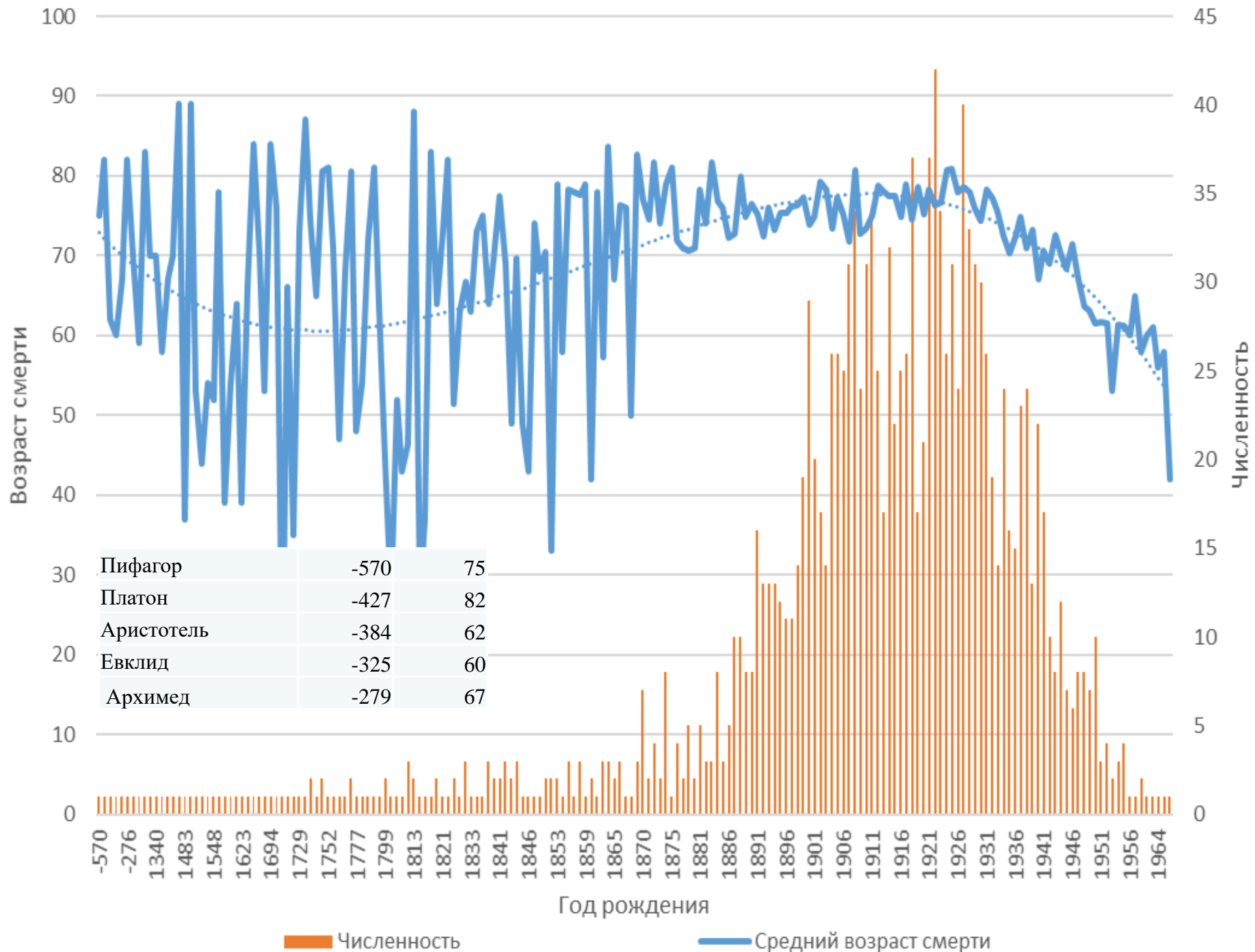
Beautiful Soup — это библиотека для извлечения данных из HTML и XML. С ее помощью можно извлечь из сложной структуры разметки нужную информацию.

Requests — это библиотека, которую можно использовать для отправки всех видов HTTP-запросов. У нее много функций, начиная от передачи параметров в URL-адресах до отправки пользовательских заголовков и проверки SSL.

Документация:

<https://requests.readthedocs.io/en/master/user/quickstart/>
<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Пример



Регулярные выражения

Регулярные выражения (Regular Expressions, regex) – это строки, содержащие совокупность *обычных символов* и *специальных метасимволов (служебные символы)*, которые описывают определенный шаблон.

Эти шаблоны, к примеру, можно использовать для того, чтобы:

- находить и заменять что-то в текстовых данных;
- валидировать строковые поля.

Говоря простым языком, регулярное выражение – это последовательность символов, используемая для поиска и замены текстовой информации в строке или файле. Их поддерживает множество языков программирования общего назначения:

Python, Perl, R и другие.

| | |
|--------|---|
| * | любое количество вхождений предыдущего символа (группы символов), от 0 до бесконечности |
| + | количество вхождений предыдущего символа (группы символов) от 1 до бесконечности |
| ? | 0 или 1 вхождение предыдущего символа (группы символов) |
| {n} | точное количество вхождений – n раз предыдущего символа (группы символов) |
| {n, m} | количество вхождений не менее n и не более m раз предыдущего символа (группы символов) |

Примеры метасимволов:

\d - любая цифра. Аналогично [0-9]

\D - все, кроме цифры. Аналогично [^0-9]

\w - любая буква, цифра и символ подчеркивания

\W - все, кроме букв, цифр и символа подчеркивания

\s - любой пробельный символ, включая сам пробел: [\t\n\r\f\v]

\S - все, кроме пробельных символов

Регулярные выражения. Примеры

- `re.findall(r'id\d+', msg)` - находит идентификаторы участников в тексте `msg`;
- `re.findall(r'<(\w+)', text_html)` - находит список всех HTML-тегов в тексте `text_html`;
- `re.findall(r'\".*?\"', text_references)` - возвращает список ссылок (текст между "");
- `re.findall(r'>[^\w.?! ,]*<', text_references)` - возвращает список содержимого тегов;

- `from bs4 import BeautifulSoup`
`soup = BeautifulSoup(html_doc, 'html.parser')`
`soup.find_all('a')` – находит все теги `<a>`;

- `for tag in soup.find_all(re.compile("^b")):`
`print(tag.name)` - находит все теги, начинающие с `b`

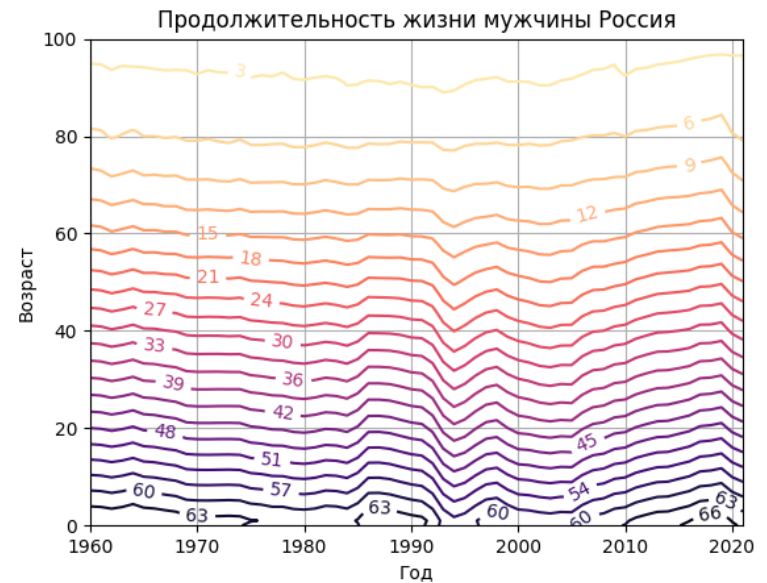
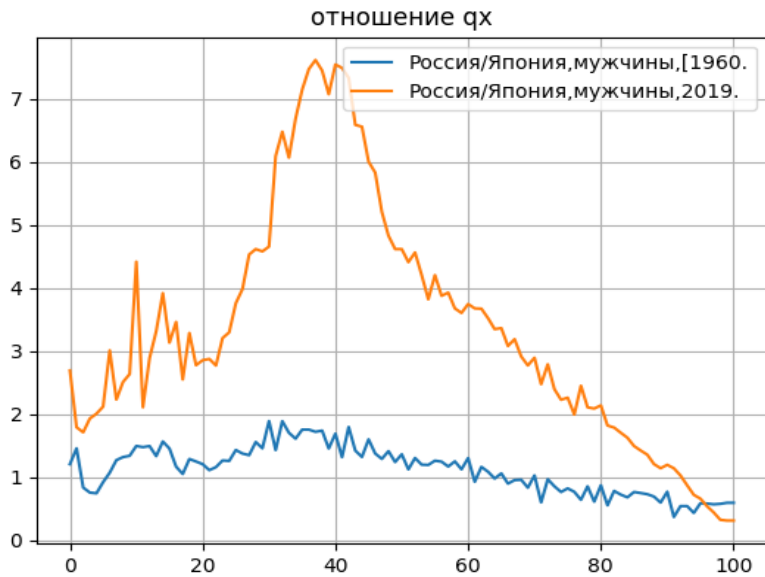
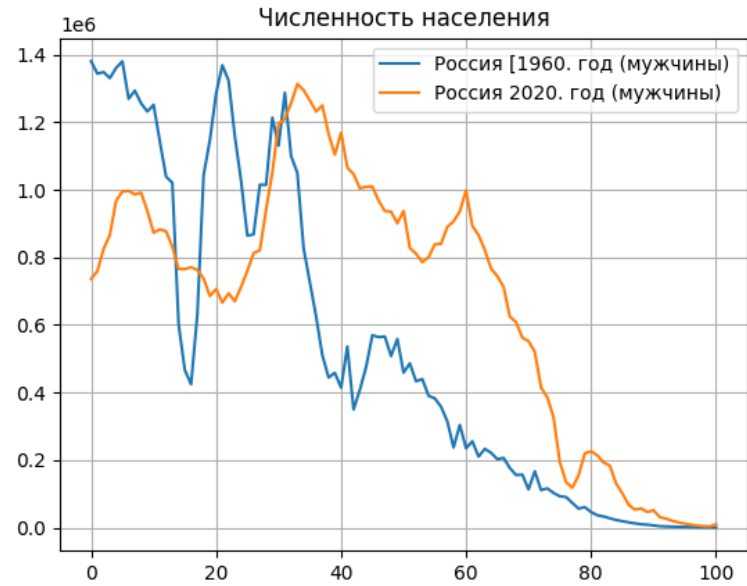
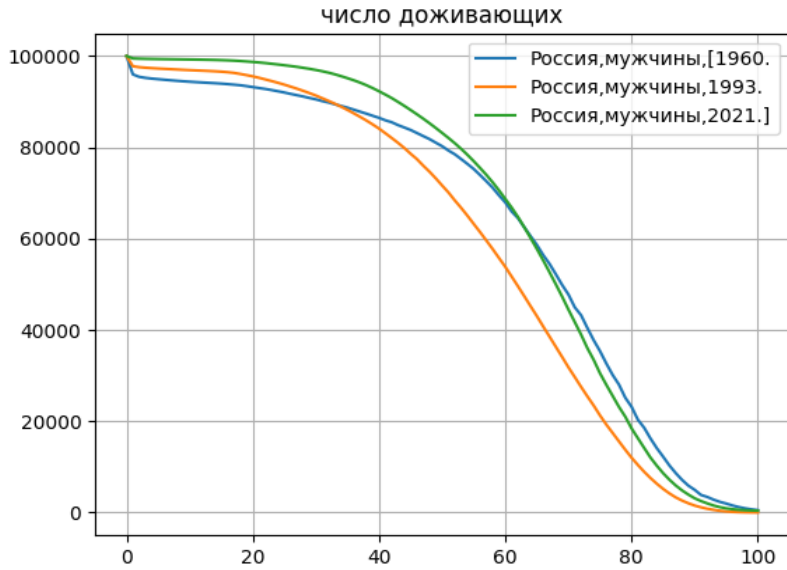
Полезные ссылки:

<https://docs.python.org/3/library/re.html> - документация по регулярным выражениям

<https://regex101.com/> – тестер регулярных выражений

<https://tproger.ru/translations/regular-expression-python/> - хорошая статья по Regex

Data set's visualization



Другие примеры, использование Google Colab

11:58 colab python - x

https://colab.research.google.com colab.research.google.com

Копия_блокнота__Untitled25_ip demography.ipynb

Файл Изменить Вид Вставка Среда

+ Код + Текст

```
from bs4 import BeautifulSoup
import requests as req
AppendList = []
resp = req.get('https://ruspanteon.ru/baburin-')
soup = BeautifulSoup(resp.text,

for tag in soup.find_all('h6'):
    a=tag.a.attrs
    AppendList.append(a['href'])

print(AppendList)
```

['https://ruspanteon.ru/baburin-']

```
[ ] from bs4 import BeautifulSoup
import requests as req
#spisok=['http://ruspanteon.ru/a
TotalList=[]
for item in AppendList:
    resp = req.get(item)
    soup = BeautifulSoup(resp.te
    FIO=(soup.h1.text)
    Years=(soup.strong.text)
    s='.'.join((FIO,Years))
    TotalList.append(s)
#print(s)
```

Файлы

- drive
- sample_data
- MT_morg.xlsx

Текст за умолчан

```
#@titl
variab
from cl
from t
import
from t
import
import
import
import
import
import
global
global
global
```

T0=196
Tp=62
t=Tp
nc=101
ncont=
dqxbg=

NumC=1:
file='|
#file=
#file=

wookbo

#pd.re

MT_morg.xlsx

Диск Доступно: 85.16 GB.

12:00 colab.research.google.com

demography.ipynb

```
#form.mainloop()
Россия
0
Фильтрация данных
0 0
```

Суммарный коэффициент рождаемости

Фильтрация данных

```
0 0
```

Изменение уровня смертности (%)Россия

Фильтрация данных

```
0 1
```

Изменение уровня смертности (%)Россия

Расчет денежных потоков в Access-VBA

Модель денежных потоков НПФ ГАЗФОНД

LAT | Таблицы смертности | Параметры

Дата расчета: _____ Начало расчета: _____ Время расчета (мин.): _____

Страховые и инвестиционные контракты

Выбор схемы:

Выбор базы:

Период:

Индексация: Корректировка:

Сценарий расчета:

Настройки

Запись результатов расчета в базу данных

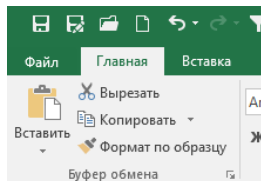
Выбор актуарного базиса (Таблица смертности):

Интерполяция Ix:

Выбор актуарного базиса (Ставки):

Индексация выплат в конце года:

Рост расходов ежемесячно:



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|----|-------------|-------------|------------------------------|--------------------------------------|----------------------------|------------|------------------------------------|----------------------------------|-------------------------|----------------|---------------|--------|---------|----------------------|---------------------|
| | Название | дата списка | название таблицы в MS Access | доходность, начисленная на счета (%) | ставка дисконтирования (%) | индексация | ежегодный темп снижения смертности | операционные расходы (руб в год) | темп роста расходов (%) | доходность (%) | УК (%) ИНВЕСТ | СД (%) | АСВ (%) | Вознаграждение фонду | СР% от ИД (Схема 1) |
| 11 | Схема1_кейс | 31.12.2023 | Схема1_кейс | | | | 0% | 7586 | | | 9.63% | 0.040% | 0.05% | 15% | |
| 12 | Портфель | 31.12.2023 | БД_САС | | | | 0% | 1964 | 5% | | 9.63% | 0.040% | 0.05% | 15% | 5% |
| 13 | 728324 | 31.12.2023 | 728324 | | | | 0% | 1964 | 5% | | 9.63% | 0.040% | 0.05% | 15% | 5% |
| 14 | 119842 | 31.12.2023 | 119842 | | | | 0% | 1964 | 5% | | 9.63% | 0.040% | 0.05% | 15% | 5% |
| 15 | 111583 | 31.12.2023 | 111583 | | | | 0% | 1964 | 5% | | 9.63% | 0.040% | 0.05% | 15% | 5% |
| 16 | Группа МСФО | 31.12.2023 | Группа_МСФО | | | | 0% | 1964 | 5% | | 9.63% | 0.040% | 0.05% | 15% | 5% |
| 17 | 10481 | 31.12.2023 | 10481 | | | | 0% | 1964 | 5% | | 9.63% | 0.040% | 0.05% | 15% | 5% |
| 18 | | | | | | | | | | | | | | | |

для инвест схем, для

Расчет денежных потоков

Модель CF ver 17IFRS_ставки МСФО-перм инфл_Сх2_7_23072024-CP - Excel

Файл Главная Вставка Разметка страницы Формулы Данные Рецензирование Вид Разработчик Настройки SAS Что вы хотите сделать?

| | A | B | C | D | E | F | G | H | I | J | K | L |
|----|-----------------------------|-------------|-----------------|-------------------|---|---|--|-----------------|----------------------|-------------|------------------|-------------|
| | вариант | Численность | Средний возраст | остатки на счетах | современная стоимость потока взносов, выплат выкупных, наследственных сумм и пенсий | современная стоимость потоков расходов на АСВ и вознаграждение УК, СД | современная стоимость потока операционных расходов | сумма потоков | Вознаграждение фонда | | PM_НПО_страховые | |
| 1 | | | | | | | | | | | | |
| 2 | 31.12.2023 | 126 035 | 25 681,1400 | 243 934 625 | 167 776 446 | 16 685 022 | 1 935 756 | 186 397 224 | 0 | - | 4 381 705,31 | 18 620 778 |
| 3 | Схема 1 | 119361 | | 239 477 724 714 | 163 969 239 933 | 16 392 243 259 | 1 838 046 995 | 182 199 530 186 | 29 341 370 647 | | | |
| 4 | Схема 2 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | | | |
| 5 | Схема 3 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | | | |
| 6 | Схема 4 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | | | |
| 7 | Схема 5 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | | | |
| 8 | Схема 6 | 6 674,00 | | 4 456 900 785 | 3 807 205 773 | 292 778 498 | 97 709 232 | 4 197 693 502 | 377 423 976 | | | |
| 9 | Схема 8 | - | | - | - | - | - | - | - | | | |
| 10 | руб. | 01.01.2024 | 01.02.2024 | 01.03.2024 | 01.04.2024 | 01.05.2024 | 01.06.2024 | 01.07.2024 | 01.08.2024 | 01.09.2024 | 01.10.2024 | 01.11.2024 |
| 11 | сумма всех выплат, в т.ч. | 1366274973 | 1383488110 | 1368087195 | 1351438955 | 1335027173 | 1318111974 | 1301939376 | 1284986520 | 1268388893 | 1252403004 | 1236264024 |
| 12 | выплаты пенсий | 1366274973 | 1350614001 | 1335214154 | 1318632317 | 1302277958 | 1285444542 | 1269214026 | 1252508863 | 1236008731 | 1220099703 | 1204003666 |
| 13 | выплаты наследникам | 0 | 140739,2881 | 278948,594 | 414011,1394 | 546238,5459 | 675425,0524 | 802017,0228 | 925422,7635 | 1045916,886 | 1164035,176 | 1279047,211 |
| 14 | выплаты ритуального пособия | 0 | 32733369,98 | 32594091,91 | 32392626,93 | 32202976,27 | 31992006,82 | 31923333,37 | 31552234,56 | 31334244,25 | 31139265,05 | 30981310,7 |
| 15 | выплаты по расторжению | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | операционные расходы | 20621965,67 | 20451416,24 | 20283556,12 | 20096355,52 | 19911287,92 | 19717167,08 | 19528677,57 | 19333897,28 | 19140290,85 | 18955017,89 | 18765412,23 |
| 17 | УК | 0 | 149097776,1 | 156895170,4 | 162148744,1 | 165624318,8 | 167489800,9 | 168212437,8 | 167939724,6 | 166949479 | 165471282,7 | 163546613,4 |
| 18 | СД | 0 | 8011026,185 | 7949477,934 | 7883907,861 | 7821796,681 | 7758037,639 | 7697844,944 | 7635948,371 | 7574594,667 | 7516611,514 | 7457012,873 |
| 19 | АСВ | 0 | 10013782,73 | 9936847,418 | 9854884,826 | 9777245,851 | 9697547,048 | 9622306,18 | 9544935,464 | 9468243,334 | 9395764,392 | 9321266,092 |
| 20 | Вознаграждение фонду | 0 | 295072593,4 | 310681310,8 | 321211935,2 | 328190907,3 | 331954832,8 | 333434883,2 | 332927156,8 | 330985524,2 | 328067600,6 | 324257867,4 |
| 21 | | 0 | 2,34812E-07 | 3,1711E-07 | -4,60521E-11 | 6,28126E-08 | 1,21005E-07 | 152980,0395 | 12252252,1 | 16234883,02 | 13965410,39 | 7115855,876 |

