

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

**Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю «122 Комп'ютерні науки»**

на тему:

**МАНІПУЛЮВАННЯ ДАНИМИ ЗАСОБАМИ СТАТИСТИЧНОГО
ПАКЕТУ SAS**

Виконала студентка 4 курсу бакалаврату
Рудзей Вікторія Олександрівна


(підпис)

Науковий керівник:
доктор фіз.-мат. наук, професор
Шкільняк Степан Степанович


(підпис)

Засвідчую, що в цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент


(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри теорії і технологій
програмування, протокол № 10

«__» _____ 2021 р.

Завідувач кафедри
проф. М. С. Нікітченко

(підпис)

Київ – 2021

ЗМІСТ

| | |
|---|-----------|
| РЕФЕРАТ | 3 |
| СКРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ | 4 |
| ВСТУП | 5 |
| РОЗДІЛ 1.ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ТЕХНОЛОГІЙ | 8 |
| РОЗДІЛ 2.ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ | 12 |
| 2.1. LSAF | 12 |
| 2.2. Статистичний пакет SAS | 14 |
| РОЗДІЛ 3.РЕАЛІЗАЦІЯ АЛГОРИТМУ ПОРІВНЯННЯ | 18 |
| 3.1. Використання LSAF macro | 18 |
| 3.2. Реалізація алгоритму | 20 |
| РОЗДІЛ 4.ІНСТРУКЦІЯ КОРИСТУВАЧА | 28 |
| 4.1. Запуск програми в LSAF системі | 28 |
| 4.2. Отримані результати | 30 |
| Висновки | 38 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ | 39 |
| ДОДАТКИ | 40 |
| Додаток А. Довідка про впровадження | 40 |
| Додаток Б. Реалізація коду для дизайну файлу | 41 |

РЕФЕРАТ

Обсяг роботи 42 сторінки, 32 ілюстрації, 1 таблиця, 7 джерел посилань.

КЛІНІЧНІ ДОСЛІДЖЕННЯ, АНАЛІЗ ДАНИХ, ПОРІВНЯННЯ ДАТА-СЕТІВ, SAS, LSAF.

Об'єкт дослідження – програми для порівняння дата-сетів, орієнтовані на роботу на фреймворку LSAF із використанням пакету SAS.

Предмет дослідження – різноманітні дата-сети, які містять в собі інформацію про пацієнтів, їх лабораторні дані, історію хвороби та інші.

В роботі використано такі методи дослідження, як порівняння, спостереження та аналіз даних із застосуванням статистичного програмного пакету.

Метою дипломної роботи є розробка оригінального програмного застосунку, який дає змогу автоматизувати порівняння дата-сетів з різних директорій.

Методи розроблення: комп'ютерне моделювання, розробка програмного продукту. Інструменти розроблення: операційна система Windows 10, середовище LSAF; при реалізації використано статистичний пакет SAS.

Результати роботи: наведено огляд та проаналізовано існуючі рішення та технології, які використовуються для автоматизації аналізу даних. Сформульовано задачі, що має вирішувати функціонал в складі системи LSAF. Розроблено оригінальний програмний застосунок із використанням пакету SAS для автоматично порівняння дата-сетів.

За методами розробки та інструментальними засобами робота виконувалася сумісно з пошуком інформації в всесвітній мережі.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

CDISC - Clinical Data Interchange Standards Consortium (CDISC), standards developing organization

COVID-19 – інфекційна хвороба, що переросла у пандемію

CRO – contract research organization

LSAF – інтегроване програмне середовище для програмування та використання SAS забезпечення (Life Science Analytics Framework)

SAS – пакет статистичного програмного забезпечення (Statistical Analysis System)

БД – база даних

ІТ – інформаційні технології

ВСТУП

Оцінка сучасного стану об'єкта розробки. У сучасному світі цінність медицини є дуже високою, при цьому незмірно зростає роль медичних працівників та значення клінічних досліджень загалом. Розпал пандемії корона-вірусу COVID-19 лише посилює цю тенденцію.

Клінічні випробування є ключовим інструментом для впровадження результатів медичних досліджень в практику лікування та подальшого догляду за пацієнтами.

Клінічні дослідження зазвичай проводяться для вже існуючих препаратів з метою їх вдосконалення, визначення нормальних доз для людей різного віку та дослідження побічних ефектів. В процесі таких досліджень часто вдається розробити вдосконалені ліки, кращі за їх попередні аналоги, або ж винайти цілком нові, яких не існувало досі.

Такі дослідження тривають досить довго, іноді понад 5 років, часом не вистачає й 10 років. Це залежить від препарату, який досліджують, та стадії його розробки. Над клінічними дослідженнями зазвичай працює висококваліфікована команда фахівців різних профілів, до якої входять медики, дата-менеджери, програмісти, статистики та інші. При цьому програмістська компонента в клінічних дослідженнях в останній час набуває особливого значення. Тому вкрай важливою є необхідність уникнення помилок в програмах і даних, адже ціна помилки дуже висока – здоров'я, а то й життя людей.

Актуальність роботи та підстави для її виконання. При проведенні досліджень не завжди можна вчасно помітити помилки в даних, адже вони мають великий обсяг та часто оновлюються, іноді щотижня, а то й частіше. Тому для виявлення таких помилок часто створюють спеціальні програми порівняння даних та їх валідації.

Метою розробки запропонованого в роботі програмного застосунку та його інтеграції в організаціях, що використовують фреймворк LSAF, є підвищення ефективності роботи програмістів та статистиків, забезпечення автоматизації перевірки на точність даних, що надходять в систему. В процесі виконання даної кваліфікаційної роботи бакалавра було істотно поглиблено знання пакету SAS, досліджено вже існуючі технології та макроси, розроблені для фреймворку LSAF, запропоновано та реалізовано нові алгоритми для порівняння дата-сетів, написані із застосуванням пакету SAS.

Для досягнення цієї мети поставлені наступні завдання:

- дослідити можливий функціонал пакету SAS та фреймворку LSAF на наявність вбудованих макросів для полегшення роботи;
- здійснити аналіз та поглибити знання макросів для системи LSAF;
- поглибити знання пакету SAS;
- сформулювати алгоритм обробки даних для порівняння;
- розробити алгоритм для перевірки на наявність дата-сетів з іншими назвами в різних директоріях;
- додати можливість видаляти/залишати задані змінні з дата-сетів;
- додати можливість видаляти/залишати пацієнтів з заданими кодовими значеннями з дата-сетів;
- створити кінцевий дата-сет з результатами порівняння та файл у форматі .rtf для зручного читання;
- завершивши попередні етапи, протестувати їх на реальних даних.

Новизна роботи полягає у розробці оригінального програмного застосунку із використанням статистичного пакету SAS для автоматично порівняння дата-сетів з різних директорій.

Можливі сфери застосування роботи – це в першу чергу CRO, які проводять клінічні дослідження, а також програмні системи, які використовують SAS. Прикладом такого CRO є «КОВАНС КЛІНІКАЛ РІСЕРЧ», де було впроваджено запропонований в роботі програмний застосунок для порівняння великої кількості даних (див. додаток А).

РОЗДІЛ 1.

ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ТЕХНОЛОГІЙ

Для огляду існуючих систем, які використовуються для аналізу даних, було досліджено варіанти розробки рішення мовою Python та R. Поставлену в роботі задачу також можна вирішити іншими мовами програмування, але зупинимось саме на цих, адже R дещо схожий на статистичний пакет SAS та подекуди має більший функціонал, а Python був обраний як альтернатива для порівняння.

Розглянемо розробку рішення, використовуючи мову R.

R – це мова програмування і середовище розробки, призначене для аналізу даних та, зокрема, статистичних обчислень, машинного навчання [1]. Завдяки наявності великої кількості безкоштовних бібліотек для різних завдань, в останні роки ця мова стає все більш популярною серед тих, хто займається аналізом даних. В цій галузі середовище R вважається цілком конкурентоздатним з мовою програмування Python.

Для того, щоб порівняти два дата-сети в мові R, потрібно використати функцію `comparedf(df1, df2)` [2], в результаті чого ми отримали такий вивід (див. рис.1).

```
Compare Object

Function Call:
comparedf(x = df1, y = df2)

Shared: 3 non-by variables and 3 observations.
Not shared: 2 variables and 0 observations.

Differences found in 2/3 variables compared.
0 variables compared have non-identical attributes.
```

Рисунок 1. Приклад результату порівняння 1 мовою R

Для більш детального опису потрібно використати функцію *summary()* (див. рис.2).

| Summary of overall comparison | |
|---|-------|
| statistic | value |
| Number of by-variables | 0 |
| Number of non-by variables in common | 3 |
| Number of variables compared | 3 |
| Number of variables in x but not y | 1 |
| Number of variables in y but not x | 1 |
| Number of variables compared with some values unequal | 2 |
| Number of variables compared with all values equal | 1 |
| Number of observations in common | 3 |
| Number of observations in x but not y | 0 |
| Number of observations in y but not x | 0 |
| Number of observations with some compared variables unequal | 2 |
| Number of observations with all compared variables equal | 1 |
| Number of values unequal | 4 |

| Differences detected | | | | | | |
|----------------------|-------|---------------|----------|----------|-------|-------|
| var.x | var.y | ..row.names.. | values.x | values.y | row.x | row.y |
| id | id | 1 | person1 | person3 | 1 | 1 |
| id | id | 3 | person3 | person1 | 3 | 3 |
| a | a | 1 | a | c | 1 | 1 |
| a | a | 3 | c | a | 3 | 3 |

Non-identical attributes

No non-identical attributes

Рисунок 2. Приклад результату порівняння 2 мовою R

Розглянемо розробку рішення, використовуючи Python.

Порівняти два дата-сети в мові Python можливо із використанням бібліотеки *datacompu* [3]. Результат порівняння наведений нижче (див. рис.3).

```

Column Summary
-----

Number of columns in common: 18
Number of columns in Original but not in New: 0
Number of columns in New but not in Original: 0

Row Summary
-----

Matched on: policyid
Any duplicates on match values: No
Absolute Tolerance: 0
Relative Tolerance: 0
Number of rows in common: 36,634
Number of rows in Original but not in New: 0
Number of rows in New but not in Original: 0

Number of rows with some compared columns unequal: 2
Number of rows with all compared columns equal: 36,632

Column Comparison
-----

Number of columns compared with some values unequal: 2
Number of columns compared with all values equal: 16
Total number of values which compare unequal: 2

```

Рисунок 3. Приклад результату порівняння мовою Python

Повного адекватного рішення для вирішення поставленої задачі на мові R та Python не було знайдено, хоча, вважаю, це можна реалізувати складнішими шляхами, використовуючи цикли для зчитування назв дата-сетів тощо.

Принциповим недоліком R та Python є те, що ці середовища не інтегровані в систему LSAF, а їх використання суперечить правилам компанії, що стосуються захисту даних. Також важливим фактором є валідованість мови для подачі даних в регулюючі органи, такі як, FDA, PMDA. Саме через те, що статистичні команди та їх результати на мовах R та Python не завалідовані, ми використали статистичний пакет SAS.

Нижче наведена таблиця порівняння можливостей для SAS, R та Python (див. табл.1).

| № | Можливості | SAS | Python | R |
|---|--|-----|--------------|--------------|
| 1 | Безкоштовний ресурс, можливо завантажити та встановити | Ні | Так | Так |
| 2 | Валідований пакет | Так | Ні | Ні |
| 3 | Використання в LSAF | Так | Впровадження | Впровадження |

Таблиця 1. Порівняння SAS, R та Python

РОЗДІЛ 2.

ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

2.1. LSAF

Фреймворк SAS Life Science Analytics (LSAF) – це інтегроване програмне середовище. На сьогоднішній день LSAF є структурою, яка забезпечує належний баланс між ефективністю роботи користувачів та дотриманням належних нормативних вимог: підтримки загального руху даних, аналізу даних та звітності, включаючи стандарти CDISC, управління інформацією та аналітикою.

LSAF має легкий та зрозумілий інтерфейс. Крім програм з розширенням .sas, у LSAF необхідно використовувати програми з розширенням .job для виконання програм у глобальному репозиторії, це потрібно для того, щоб зберігати вихідні дані та журнали виконання до сховища.

Нижче наведена схема того, як проходить робочий процес в системі LSAF (див. рис.4):

1. Отримуємо дані від CRO або ж клінічних сайтів, де проходять медичні дослідження;
2. Розміщуємо дані у відповідні папки;
3. Запускаємо код для всіх необхідних програм або ж створюємо нові, якщо є потреба в тому:
 - 3.1 Якщо вдало запустились програми:
 - 3.1.1 Перевірка на рівні програми;
 - 3.1.2 Надання даних до статистики;
 - 3.2 Якщо програма невдало запустилась:
 - 3.2.1 Виправляємо код, якщо проблема в ньому;
 - 3.2.2 Знаходимо помилки в даних та надсилаємо запит до клінічних сайтів.

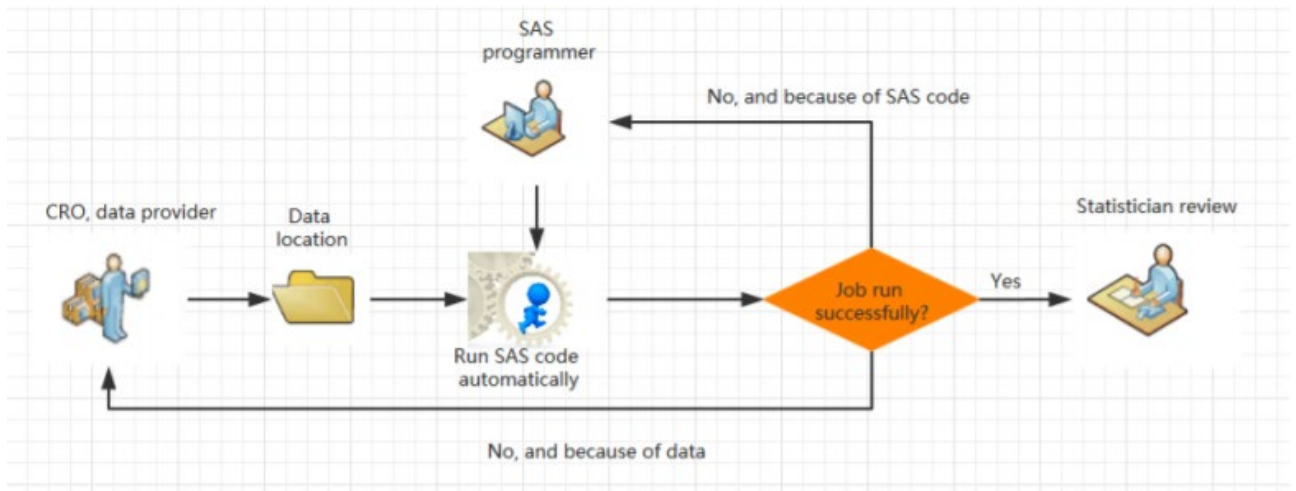


Рисунок 4. Схема роботи в LSAF

За допомогою LSAF [4] середовище SAS забезпечує основу для створення єдиного інтегрованого джерела клінічних даних та метаданих з різних джерел та відділів, де дослідники в клініках можуть миттєво запитувати, отримувати та аналізувати дані.

Наведемо основні переваги при роботі в системі LSAF:

1. Об'єднання клінічної інформації в єдиному глобальному сховищі.
2. Перегляд конкретних версій набору даних, що використовуються, створених файлів журналів, запущених програм SAS та отриманих результатів.
3. Швидке відтворення результатів.
4. Відстеження походження даних від вихідних даних.
5. Швидка відповідь на регуляторні запити.
6. Можливість легко визначити, які аудиторські зміни були внесені, коли і ким були внесені, що зберігається у сховищі.
7. Інтеграція із середовищами програмування та виконання програм SAS.
8. Повний контроль за виконанням завдань SAS.

9. Можливість легкого завантаження та використання стандартних аналітичних програм.
10. Можливість використання електронних підписів, встановлення версій та привілеїв на основі ролей.
11. Контроль усієї інформації через захищені логіни, аудиторські стежки, керування версіями на основі ролей, привілеїв та політики.
12. Повне відображення джерела даних, маніпуляцій із даними та кінцевого пункту призначення даних.
13. Автоматизоване завантаження інформації для клінічних даних за графіком з необхідною частотою.

2.2. Статистичний пакет SAS

SAS (" Statistical Analysis System") – це програмний пакет, розроблений Інститутом SAS. Він успішно використовується для статистичного аналізу та візуалізації даних. Як і кожне адекватне програмне забезпечення, SAS має власну мову, яка може керувати програмою під час її виконання.

Пакет SAS [5] розроблявся в Університеті штату Північна Кароліна з 1966 по 1976 рік, коли був заснований Інститут SAS. Далі SAS був модифікований і розширений у 1980-х та 1990-х роках з додаванням нових статистичних процедур, додаткових компонентів та введенням JMP.

Програмний пакет SAS включає точні методи, високопродуктивні засоби статистичного моделювання для завдань на великих даних та сучасні методи аналізу даних із відсутніми значеннями (неповною інформацією). Система статистичного аналізу розроблена для роботи з даними, отриманими з численних різноманітних джерел. Дані збирають разом і використовують для проведення певного статистичного аналізу для отримання очікуваного результату.

SAS має широкий спектр застосувань завдяки потужному функціоналу, який він пропонує. Нижче наведено кілька варіантів того, що можна ефективно реалізувати за допомогою SAS:

1. Створення графічного звіту. Програма статистичного аналізу дозволяє використовувати необроблені дані для формування графічного звіту (див. рис.5). Завдяки своїй особливості графічний звіт є простим у використанні навіть для людей без спеціальної освіти.
2. Статистичний аналіз. Статистичні дані, отримані в результаті обробки вхідних даних, далі можуть бути використані для статистичного аналізу.
3. Поліпшення якості. Найефективніший спосіб поліпшення якості – це порівняння старих результатів з останніми результатами, які були створені після внесення змін для покращення системи.
4. Виконання досліджень. Дослідження тут полягає в розгляді та аналізі даних перед прийняттям остаточного рішення. Інформація, отримана за допомогою програмного забезпечення SAS, допомагає у прийнятті належних рішень щодо результатів дослідження.
5. Розробка додатка. Розробники можуть писати персоналізовані коди для проектування генерації звітів відповідно до вимог.

Код, створений за допомогою SAS, є задокументованим та перевіраним на відповідність вимогам корпоративного та державного контролю.

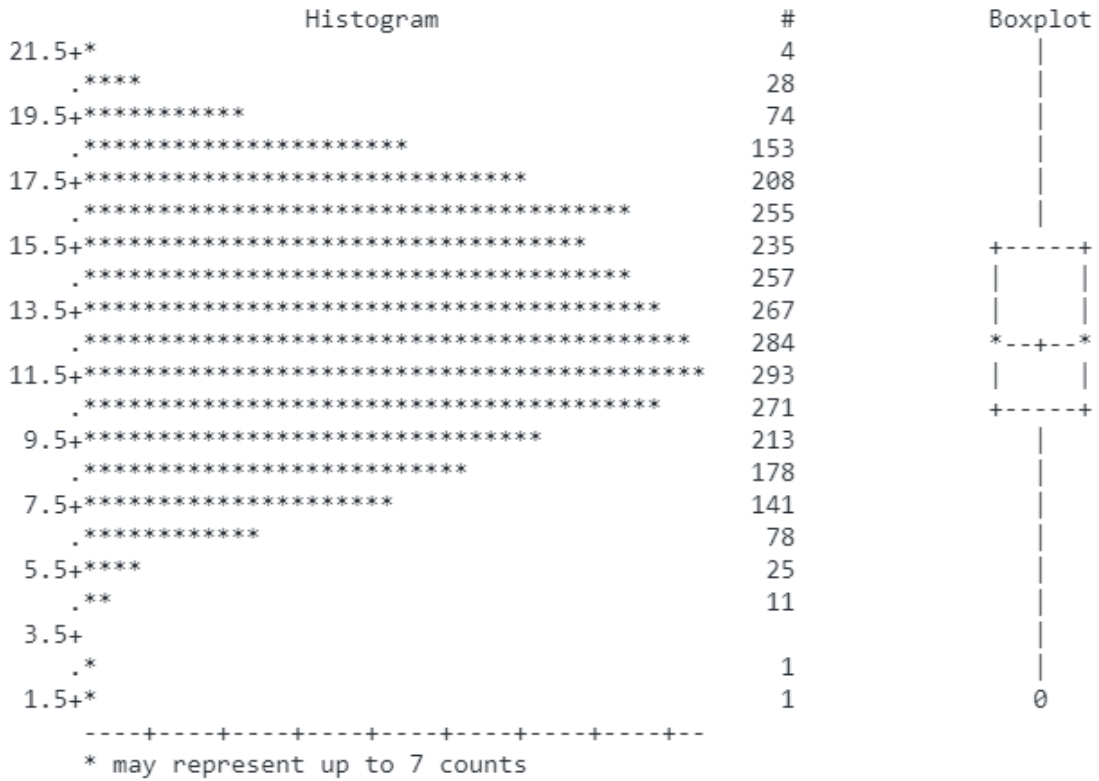


Рисунок 5. Приклад графічного звіту

Для кращого розуміння того, як пишуться коди мовою SAS [6], нижче наведено простий приклад виконання програми (див. рис.6).

```

3
4  data tutorial;
5      x=1;
6      y=10;
7  run;
8
9  proc print data=tutorial;
10 run;
11

```

Рисунок 6. Проста програма на мові SAS

Слово «DATA» є ключовим словом, а «TUTORIAL» – це ім'я файлу, який збереже цю програму. Кожне твердження (оператор), написане мовою SAS, має закінчуватися крапкою з комою.

X та Y використовуються для зберігання призначеного значення, яке буде зберігатися в рядку.

«PROC» – це ключове слово, яке використовується в першому рядку після позначки рядка.

Результат виконання програми дає візуальне подання даних, які включені за допомогою програми (див .рис.7).

Вихідні дані тут мають 2 рядки та 3 стовпці.

Стовпець Obs показує номер, X та Y - значення, присвоєні йому.

```
The SAS System
```

```
Obs      x      y
1         1     10
```

Рисунок 7. Результат виконання програми

РОЗДІЛ 3.

РЕАЛІЗАЦІЯ АЛГОРИТМУ ПОРІВНЯННЯ

1.1. Використання LSAF macro

Статистичний пакет SAS має потужний засіб програмування під назвою Macros, який дозволяє нам уникати повторюваних розділів коду та в разі потреби дає змогу використовувати їх знову і знову. Це також допомагає створювати динамічні змінні в коді, які можуть приймати різні значення для різних екземплярів запуску одного і того ж коду.

SAS має багато операторів Macro, які вбудовані в мову програмування. Вони використовуються іншими програмами SAS без явного їх оголошення. Як приклади можна навести завершення роботи програми, коли виконується якась умова, або фіксація значення часу виконання змінної в листі виконання програми.

Фреймворк LSAF також має власні вбудовані Macro, які можна використовувати всередині системи. Це дає змогу істотно зберегти час, зменшити та спростити код.

До прикладу, в пропонованій в даній роботі програмі було використано *%lsaf_getchildren* Macro [7], який повертає набір даних SAS, що містить метадані об'єктів у контейнері SAS.

Макрос *%lsaf_getchildren* має такі параметри:

- *lsaf_path=* – обов'язковий параметр – шлях до об'єкта-контейнера в SAS Life Science Analytics Framework,
- *sas_dsname=work.lsafGetChildren* – необов'язковий параметр – ім'я вихідного набору даних SAS, який буде містити метадані для об'єктів, позначених як *libref.dataset*,
- *lsaf_recursive=0* – необов'язковий параметр – рівень рекурсії, коли шлях є контейнером.

Приклад виклику:

```
%lsaf_getchildren(lsaf_path=, sas_dsname=work.lsafGetChildren,
lsaf_recursive=0);
```

Наведемо приклад коду в програмі (див. рис.8).

```
%lsaf_getchildren(lsaf_path=%cmpres(&directory1), sas_dsname=dsname1 , lsaf_recursive=1);

data dsname1(rename=(name=dsname));
  set dsname1;
  if index(name,"_m")=0 and index(itemType,"sasdataset")>0;
  name=substr(strip(name),1,index(name,".")-1);

run;

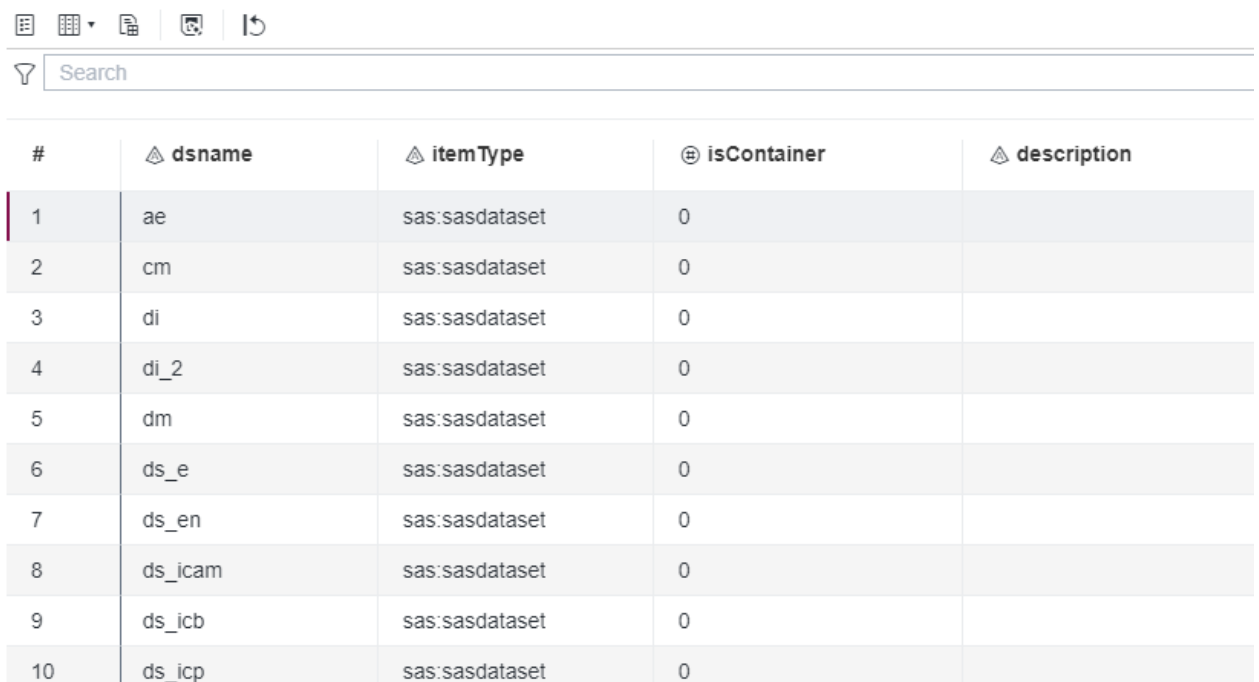
%lsaf_getchildren(lsaf_path=%cmpres(&directory2), sas_dsname=dsname2 , lsaf_recursive=1);

data dsname2(rename=(name=dsname));
  set dsname2;
  if index(name,"_m")=0 and index(itemType,"sasdataset")>0;
  name=substr(strip(name),1,index(name,".")-1);

run;
```

Рисунок 8. Використання LSAF-макросу

В результаті виклику макросу отримаємо дата-сет з списком назв дата-сетів з директорій (див. рис.9).



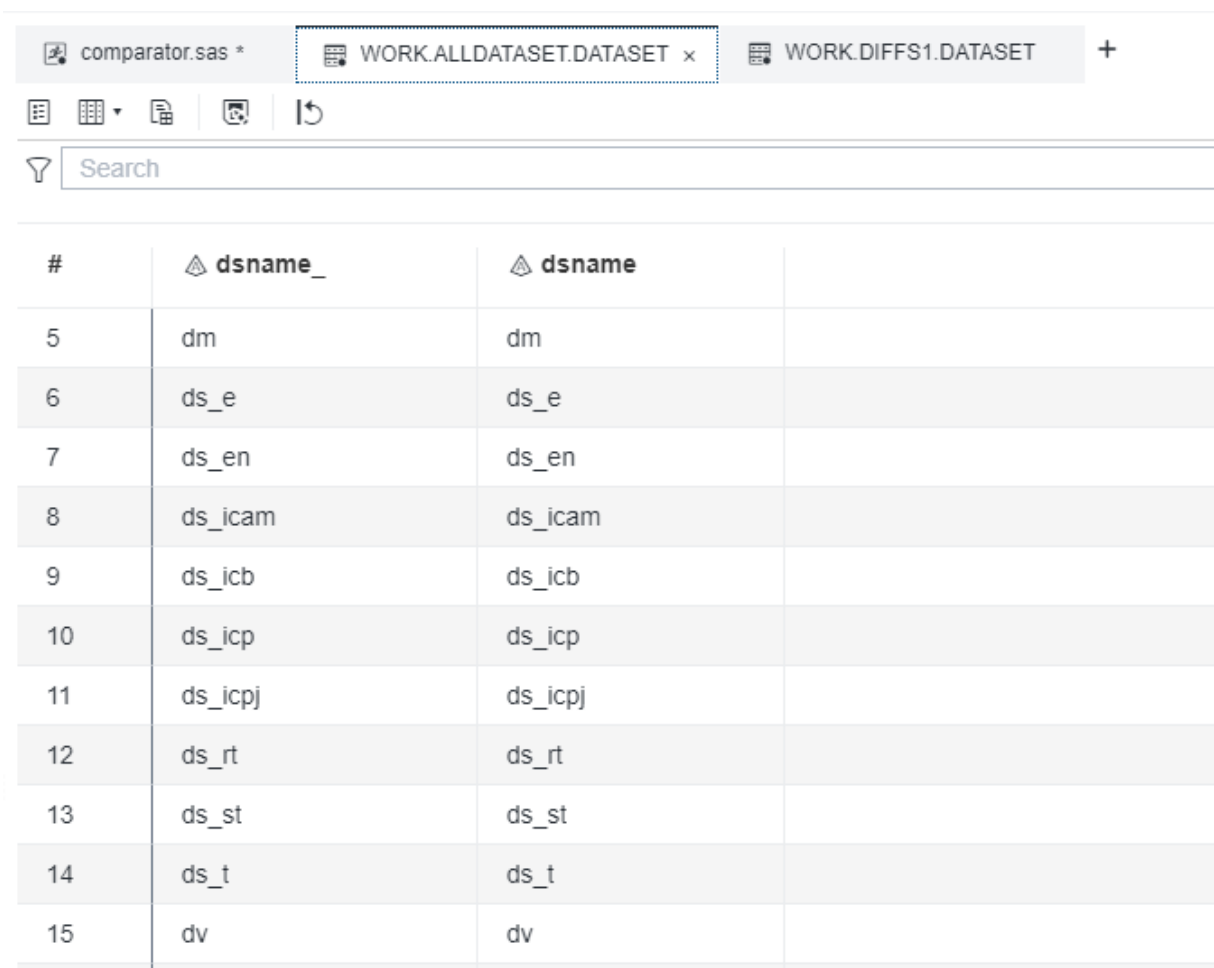
The screenshot shows the SAS interface with a search bar at the top. Below the search bar is a table with the following columns: #, dsname, itemType, isContainer, and description. The table contains 10 rows of data, all with itemType 'sas:sasdataset' and isContainer '0'.

| # | dsname | itemType | isContainer | description |
|----|---------|----------------|-------------|-------------|
| 1 | ae | sas:sasdataset | 0 | |
| 2 | cm | sas:sasdataset | 0 | |
| 3 | di | sas:sasdataset | 0 | |
| 4 | di_2 | sas:sasdataset | 0 | |
| 5 | dm | sas:sasdataset | 0 | |
| 6 | ds_e | sas:sasdataset | 0 | |
| 7 | ds_en | sas:sasdataset | 0 | |
| 8 | ds_icam | sas:sasdataset | 0 | |
| 9 | ds_icb | sas:sasdataset | 0 | |
| 10 | ds_icp | sas:sasdataset | 0 | |

Рисунок 9. Вихідний дата-сет LSAF-макросу

3.2. Реалізація алгоритму

Попередньо використавши LSAF-макросо для пришвидшення роботи програми, ми створюємо загальну БД з назвами дата-сетів для порівняння (див. рис.10).

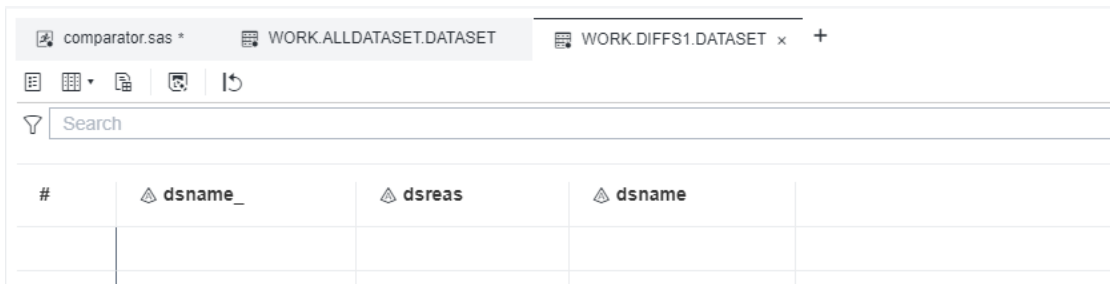


The screenshot shows the SAS interface with a table containing dataset names. The table has three columns: '#', 'dsname_', and 'dsname'. The rows list various dataset names such as 'dm', 'ds_e', 'ds_en', 'ds_icam', 'ds_icb', 'ds_icp', 'ds_icpj', 'ds_rt', 'ds_st', 'ds_t', and 'dv'.

| # | dsname_ | dsname |
|----|---------|---------|
| 5 | dm | dm |
| 6 | ds_e | ds_e |
| 7 | ds_en | ds_en |
| 8 | ds_icam | ds_icam |
| 9 | ds_icb | ds_icb |
| 10 | ds_icp | ds_icp |
| 11 | ds_icpj | ds_icpj |
| 12 | ds_rt | ds_rt |
| 13 | ds_st | ds_st |
| 14 | ds_t | ds_t |
| 15 | dv | dv |

Рисунок 10. БД з назвами дата-сетів

Також створюються БД diffs1 та БД diffs2. Наприклад, в БД diffs1 показані дата-сети, які існують в першій директорії та не існують в другій директорії; якщо такі БД порожні, то це свідчить про те, що ні в одній директорії немає дата-сетів з іншими назвами (див. рис.11).



| # | dsname_ | dsreas | dsname |
|---|---------|--------|--------|
| | | | |

Рисунок 11. БД DIFFS1

Для зручності використання всі назви дата-сетів будуть знаходитись в макро-змінній розділені крапкою, що дозволить звертатись до назви дата-сету в макро-циклі.

Було використано команду *proc sql* для реалізації цього етапу, реалізація відповідного коду наведена нижче (див. рис.12).

```
proc sql noprint;
  select dsname into: dslist separated by '.'
  from alldataset;

  select dsname into: dsname from alldataset;
  select count(dsname) into: dscount
  from alldataset;
quit;

%put Data Set List ===>>> &dslist.;
%put Data Set Count ===>>>&dscount.;
```

Рисунок 12. Приклад коду PROC SQL

Результатом виконання коду є створений список дата-сетів та кількість дата-сетів; так як макро-змінні зберігаються в пам'яті, то побачити їх можна в журналі виконання (LOG) (див. рис.13).

```
MLOGIC(COMPARE_): %PUT Data Set List ===>>> &dslist.
Data Set List ===>>>
ae.cm.di.di_2.dm.ds_e.ds_en.ds_icam.ds_icb.ds_icp.ds_icpj.ds_rt.ds_st.ds_t.dv.eg_log_1.eg_log_2.en.er_tma.er.
lb_1.lb_2.lb_3.lb_4.lb_5.lb_acm.lb_ad13.lb_avanz_hemolas.lb_avanzc5fr1210.lb_cd.lb_ch.lb_coag.lb_hm.lb_ig_1.
a.mh.mh_ho.pe.pe_s.pl.qs_ahus.qs_facitp.qs_resut.rt.sv.tp.vs_1.vs_2.vs_3
MLOGIC(COMPARE_): %PUT Data Set Count ===>>>&dscount.
Data Set Count ===>>>      72
```

Рисунок 13. Приклад виконання коду PROC SQL

Після створення макро-змінної зі списком дата-сетів, всі подальші етапи будуть опрацьовуватись в циклі від першої до останньої назви дата-сету та для двох директорій окремо, адже нам потрібно буде створити дата-сети з різними назвами для порівняння, хоча й різниця в назвах буде незначною.

Приклад початку циклу наведено нижче (див. рис.14).

```
%macro dataloop(dir=, und=);
  %do i = 1 %to &dscount.;

    data _null_;
      call symput('ds',scan("&dslist.",&i.,'.'));
    run;

    %put DATASET====>>>> &ds.;

    ** Selecting which variable will be used in conditions*****

    proc contents data=&dir.&ds. noprint out=v_&ds (keep=NAME);
  run;
```

Рисунок 14. Приклад коду для початку виконання циклу

Наступним етапом буде створення БД для кожного дата-сету зі списком змінних. Це робимо для того, щоб визначити, які змінні наявні, далі записати їх в макро-змінні та використати їх для наступних умов:

- виключити з алгоритму дата-сети які не мають змінних з кодovими номерами (див. рис.15) ;
- видалити/залишити пацієнтів з заданими кодovими номерами (див.рис.16);
- видалити задані змінні які є незначними для порівняння (див. рис.17).

```

proc sql noprint;
select NAME into: drop_s separated by ' '
from ab_&ds
;
quit;
%put DROPPED====>>>> &drop_s.;

data r_&ds;
set r_&ds;

if NAME in ("USUBJID") then flag=2;
else flag=1;

if substr(NAME,1,2) in ("TA","TI","TA","TS","TV") then delete;

f=0;
run;

```

Рисунок 15. Виключення дата-сетів

```

** Subsetting conditions for each dataset *****;

data &ds.&und;
set &dir..&ds.;

%if %length(&subject)>0 %then %do;

%if &subjname=USUBJID %then %do;
%if &subjfl=Y %then %do;
if substr(STRIIP(&subjname),length(STRIIP(&subjname))-7,8) in (&subjects_2);
%end;
%else %do;
if substr(STRIIP(&subjname),length(STRIIP(&subjname))-7,8) in (&subjects_2) then delete;
%end;
%end;

%else %do;
%if &subjfl=Y %then %do;
if STRIP(&subjname) in (&subjects_1);
%end;
%else %do;
if STRIP(&subjname) in (&subjects_1) then delete;
%end;
%end;

%end;

%if %length(&drop_s)>0 %then %do;
drop &drop_s;
%end;

format _all_;
informat _all_;
run;

```

Рисунок 16. Алгоритм видалення/залишку пацієнтів

```

/* Part for dropping variables */

proc sql feedback;
create table a_&ds as
select NAME from v_&ds where (upcase(NAME) in (%upcase(&drop_vars_)));
quit;

PROC TRANSPOSE DATA=a_&ds OUT=ab_&ds ;
var Name;
run;

data ab_&ds;
length NAME $2000 coll $32;
set ab_&ds;
if missing(coll) then coll=" ";
NAME=strip(catx(' ',of col:));

run;

proc sql noprint;
select NAME into: drop_s separated by ' '
from ab_&ds
;
quit;
%put DROPPED====>>>> &drop_s.;

```

Рисунок 17. Алгоритм видалення змінних

Після того, як ми сформуваємо дата-сети для порівняння, виконуємо перший етап:

- порівняти дата-сети, в результаті чого для кожного з них буде створено БД SUM з деталями порівняння (див. рис.18);
- отримати БД SUM (див. рис.19) порівняння та отримати значення з кількістю невідповідностей по кожному дата-сету;
- залишити для порівняння тільки ті, в яких є невідповідності, тобто число нерівних значень більше, ніж 0.

| # | type | batch |
|----|------|---|
| 1 | d | |
| 2 | d | |
| 3 | h | Observation Summary |
| 4 | h | |
| 5 | d | |
| 6 | d | Number of Observations in Common: 0. |
| 7 | d | Total Number of Observations Read from WORK.VS_3: 0. |
| 8 | d | Total Number of Observations Read from WORK.VS_3_: 0. |
| 9 | d | |
| 10 | d | Number of Observations with Some Compared Variables Unequal: 0. |
| 11 | d | Number of Observations with All Compared Variables Equal: 0. |

Рисунок 18. Приклад отриманої БД SUM

```

%macro loop;
  %do i = 1 %to &dscount.;

    data _null_;
      call symput('ds',scan("&dslist.",&i.,'.'));
    run;

    %put DATASET====>>>> &ds.;

    ods output "Summary"=sum;
    ods trace on;
    proc compare base=&ds. compare=&ds._ criterion = 0.00001 ;
      title1 "Proc Compare of Data Sets &ds.";

      format _all_ ;
      informat _all_;
    run;
    ods trace off;
    ods output close;
  
```

Рисунок 19. Приклад коду для отримання БД SUM

Наступними кроками є створення фінального дата-сету та виведення результатів в .rtf файл для комфортного читання та перегляду, передачі репорту до дата-сайтів та спонсора.

Для створення результуючого файлу було використано процедуру *proc report*, що дозволяє створювати файли з різним розширенням та дизайном. Приклад коду для використаного дизайну наведено у Додатку Б.

Нижче наведено частину коду, використаного для створення результуючого файлу (див. рис.20).

```
ods tagsets.rtf style=Custom startpage=no nogtitle nogfootnote;

proc report data=final(keep=dsname compres);
  column dsname compres;
  define dsname/display style=[cellwidth=20% just=1] 'Dataset';
  define compres/display style=[cellwidth=40% just=1] 'Mismatches';

  compute before _page_ / style=Header{fontweight=bold fontsize=12pt};
  line "Datasets that were compared" color=blue;
  line "" ;
  endcomp;

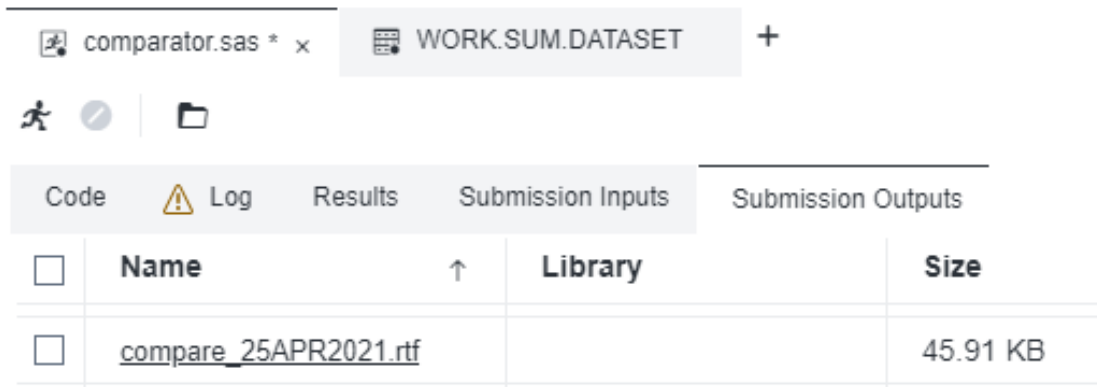
run;
title;
ods tagsets.rtf style=Custom startpage=yes nogtitle nogfootnote ;

%if &cn>0 %then %do;
proc sql noprint;
  select distinct dsname into:src1-:src%left(&cn) from final
  where statno >0;
quit;
  %do i=1 %to &cn;

    %gcompare(base=&&src&i, comp=&&src&i.&&und., out=all_comp);
  options ORIENTATION=PORTRAIT linesize=95 pagesize=120;
```

Рисунок 20. Приклад коду для створення файлу

Виконавши усі попередні етапи, ми отримаємо файл з .rtf розширенням з детальним порівнянням, який далі можемо завантажити та переглянути (див. рис.21).



The screenshot shows the SAS interface with a window titled 'comparator.sas * x' and a sub-window 'WORK.SUM.DATASET'. Below the window title bar, there are icons for a person, a refresh symbol, and a folder. A horizontal menu bar contains 'Code', 'Log' (with a warning icon), 'Results', 'Submission Inputs', and 'Submission Outputs'. Below this menu is a table with the following structure:

| <input type="checkbox"/> | Name | ↑ | Library | Size |
|--------------------------|------------------------------|---|---------|----------|
| <input type="checkbox"/> | <u>compare_25APR2021.rtf</u> | | | 45.91 KB |

Рисунок 21. Виведення та створення результуючого файлу

РОЗДІЛ 4.

ІНСТРУКЦІЯ КОРИСТУВАЧА

В цьому розділі опишемо функціонування розробленого програмного застосунку у вигляді інструкції користувача.

4.1. Запуск програми в LSAF системі

Так як LSAF – це платне середовище для використання SAS, то спочатку потрібно увійти в систему, ввести логін та пароль. В системі LSAF є Repository (репозиторій для фінальних даних) та Workspace (для розробки, тестування та програмування програм).

Спочатку створюємо програму з розширенням `.sas`, нехай це буде `comparator.sas`, також, потрібно створити `.job` файл, де ми можемо задавати загальні параметри, шляхи для вхідних та вихідних даних.

Щоб запустити програму, потрібно натиснути на іконку та кнопку «Run job» (див.рис.22).

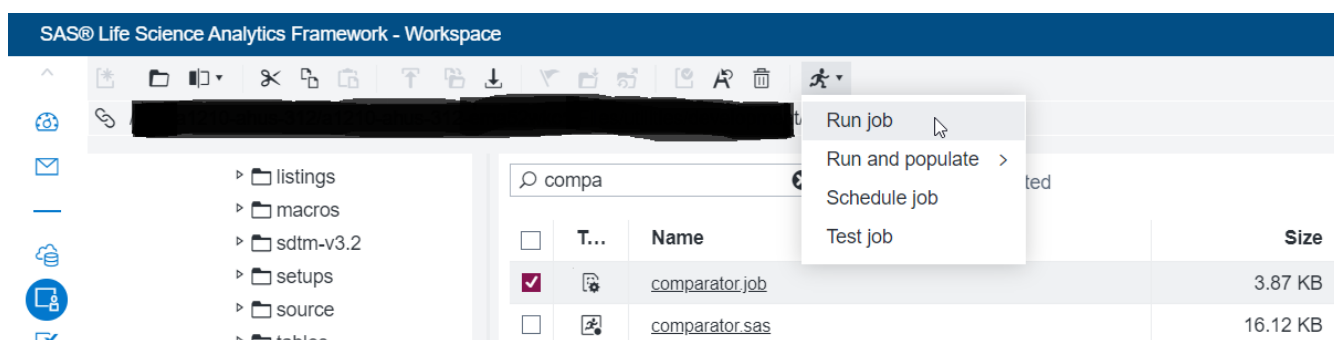


Рисунок 22. Запуск програми в LSAF

Далі відкривається вікно з `.job` файлом, який має декілька параметрів для запуску програми (див. рис.23), а саме:

1. Saveinlog – флаг-змiна для означення того чи буде результат програми зберiгатись у .log файл.
2. Зазвичай завдяки .log файлу можливо побачити, якi було допущено помилки при запуску програми. Наприклад, якщо ввести невірну адресу для першої чи другої директорії, якої не існує або ж до якої немає доступу, отримаємо помилку та повідомлення (див. рис.24), після чого програма буде зупинена, вихідний файл не буде створено.
3. Name – назва нашої програми або ж ім'я, яке ми хочемо використовувати для створення .log та .lst файлів;
4. First directory – перша директорія для порівняння;
5. Second Directory - друга директорія для порівняння;
6. Subject flag to keep – флаг, який визначає зберігати чи видаляти суб'єктів, Y – зберігати, N - видалити;
7. Subjects to keep/delete – записуємо номер суб'єктів яких потрібно зберегти чи видалити, розділяючи пробілом ;
8. Variables to drop – задаємо назви змінних, які потрібно видалити. Зазвичай, це змінні дати створення чи зміни запису, які не мають значення для порівняння.

Run Job

Name: comparator.job

Documentation:

| Label | Value |
|-------------------------|--|
| saveindlog | <input type="text" value="Y"/> |
| name | <input type="text" value="comparator"/> |
| first directory | <input type="text" value="directory1"/> |
| second directory | <input type="text" value="directory2"/> |
| subject flag to keep | <input type="text" value="Y"/> |
| subjects to keep/delete | <input type="text" value="001 002 003"/> |
| variables to drop | <input type="text" value="recorddate recordposition m"/> |
| | |
| | |
| | |

Рисунок 23. Файл .job для програми

| Program Name | Log Line No. | Discrepancy | Log Message |
|--------------|--------------|-------------|---|
| comparator | 990 | Error | ERROR: Check if DIR1 or DIR2 exists or its spelling |

Рисунок 24. Файл .rtf, приклад виконання програми

4.2. Отримані результати

Результатами виконання програми є .rtf файл (див. рис.25) з загальним порівнянням усіх дата-сетів. На початку документу ми показуємо усю інформацію про проект, шляхи до директорій, номери пацієнтів, яких ми хотіли б виключити чи включити в аналіз та порівняння, перелік змінних, які потрібно видалити тощо. Якщо ж ми не вказуємо чи не використовуємо один із параметрів, тоді відповідні записи не будуть зображені в репорті.

DIR1 library: directory1
 DIR2 library: directory2

Datasets that are not in DIR2 and were not compared

| Dataset | Reason |
|--------------|--|
| <u>adnor</u> | Compared <u>dataset</u> is not in DIR2 |

□

Datasets that were compared

| Dataset | Mismatches |
|------------------|------------------------------------|
| <u>lb_allph</u> | Dataset in DIR1 and DIR2 are empty |
| <u>lb_alxbrn</u> | Dataset in DIR1 and DIR2 are empty |
| <u>lb_eurfn</u> | Dataset in DIR1 and DIR2 are empty |
| <u>lb_secug</u> | Dataset in DIR1 and DIR2 are empty |
| <u>pc_bio</u> | Dataset in DIR1 and DIR2 are empty |
| <u>pc_bio_2</u> | Dataset in DIR1 and DIR2 are empty |
| <u>pc_bio_3</u> | Dataset in DIR1 and DIR2 are empty |
| <u>pc_bio_4</u> | Dataset in DIR1 and DIR2 are empty |
| <u>pc_bio_5</u> | Dataset in DIR1 and DIR2 are empty |
| <u>pc_bio_6</u> | Dataset in DIR1 and DIR2 are empty |
| <u>pc_bio_7</u> | Dataset in DIR1 and DIR2 are empty |

Рисунок 25. Файл .rtf, приклад 1

Наведено приклад виконання програми та створений нею документ, з якого видно, що програма порівняння була виконана для директорії1 та директорії2.

Далі маємо для кожного випадку порівняння різні розділи:

1. Дата-сети, яких не існує в директорії1 та не порівнювались;
2. Дата-сети, яких не існує в директорії2 та не порівнювались;
3. Дата-сети, які порівнювались:

- a. Прошли порівняння та не мають невідповідності (*No mismatches*);

b. Мають невідповідності:

- i. Якщо дата-сет з однієї із директорій пустий, отримаємо повідомлення про те, в якій саме директорії дата-сет порожній; надалі цей дата-сет не буде включений в процес аналізу (див. рис.26);
- ii. Якщо дата-сети непорожні, то далі ми отримаємо повідомлення про те, яку кількість невідповідностей вони мають.

Наприклад, *Number of mismatches found: 20* (див. рис.27).

| Datasets that were compared | | |
|-----------------------------|------------------------------------|------|
| Dataset | Mismatches | |
| ae | Number of mismatches found: | 447 |
| cm | Number of mismatches found: | 2164 |
| di | Number of mismatches found: | 58 |
| di_2 | Number of mismatches found: | 43 |
| dm | Number of mismatches found: | 34 |
| ds_e | Number of mismatches found: | 1 |
| ds_en | Number of mismatches found: | 34 |
| ds_icam | Number of mismatches found: | 25 |
| ds_icb | Number of mismatches found: | 31 |
| ds_icp | Number of mismatches found: | 29 |
| ds_icpi | Number of mismatches found: | 5 |
| ds_rt | Dataset in DIR1 and DIR2 are empty | |
| ds_st | Number of mismatches found: | 5 |
| ds_t | Number of mismatches found: | 31 |
| dv | Number of mismatches found: | 43 |

Рисунок 26. Файл .rtf, приклад 2

| Datasets that were compared | |
|------------------------------------|--------------------------------|
| Dataset | Mismatches |
| ae | No mismatches |
| ce | No mismatches |
| cm | Number of mismatches found: 4 |
| dm | No mismatches |
| ds | No mismatches |
| dv | No mismatches |
| ec | No mismatches |
| eg | No mismatches |
| ex | No mismatches |
| ho | No mismatches |
| ie | No mismatches |
| is | No mismatches |
| lb | Number of mismatches found: 20 |
| mh | No mismatches |
| pc | No mismatches |
| pe | No mismatches |
| pr | No mismatches |
| qs | No mismatches |
| se | No mismatches |
| suppae | No mismatches |
| suppce | No mismatches |
| suppcm | No mismatches |
| suppdm | No mismatches |
| suppds | No mismatches |
| suppdv | No mismatches |
| suppec | No mismatches |
| suppeg | No mismatches |
| suppex | No mismatches |
| suppho | No mismatches |
| suppie | No mismatches |
| suppis | No mismatches |
| supplb | Number of mismatches found: 9 |
| suppmh | No mismatches |

Рисунок 27. Файл .rtf, приклад 3

Наведемо приклад заголовку, який ми отримаємо, якщо вказати відповідні параметри (видалити пацієнтів з вказаними номерами та видалити змінні) (див. рис.28). Таким чином, ми спрощуємо порівняння, так як не буде зображено невідповідностей у вказаних змінах, адже вони не мають істотного значення для порівняння.

Subjects that were deleted from datasets: 001 002 003
 Variables that were dropped from datasets: ~~recorddate recordposition mincreated maxupdated savets~~

| Datasets that were compared | |
|-----------------------------|-----------------------------------|
| Dataset | Mismatches |
| incl_1 | Number of mismatches found: 8 |
| incl_j | Number of mismatches found: 3 |
| incl_j1 | Number of mismatches found: 1 |
| lab | Number of mismatches found: 1558 |
| lb_1 | Number of mismatches found: 89 |
| lb_2 | Number of mismatches found: 462 |
| lb_3 | Number of mismatches found: 134 |
| lb_4 | Number of mismatches found: 346 |
| lb_5 | Number of mismatches found: 1 |
| lb_acm | Number of mismatches found: 12453 |
| lb_ad13 | Number of mismatches found: 31 |
| lb_avanz_hemolas | Number of mismatches found: 631 |
| lb_avanz5fr1210 | Number of mismatches found: 458 |
| lb_cd | Number of mismatches found: 31 |
| lb_ch | Number of mismatches found: 22 |
| lb_coag | Number of mismatches found: 22 |
| lb_hm | Number of mismatches found: 26 |
| lb_ig_1 | Number of mismatches found: 152 |
| lb_ig_2 | Number of mismatches found: 5 |
| lb_pk_1 | Number of mismatches found: 432 |
| lb_pk_2 | Number of mismatches found: 211 |
| lb_ppd_ada | No mismatches |
| lb_ppd_c5total | No mismatches |
| lb_ppd_pk1210 | No mismatches |
| lb_prgs | Number of mismatches found: 25 |
| lb_pregu | Number of mismatches found: 118 |
| lb_st | Number of mismatches found: 117 |
| lb_ua | Number of mismatches found: 22 |
| mh | Number of mismatches found: 307 |
| mh_ho | Number of mismatches found: 56 |

(Continued)

Рисунок 28. Файл .rtf, приклад видалення змінних

Після загальної перевірки на невідповідності далі в репорті ми отримаємо повне порівняння для кожного дата-сету, який мав невідповідності.

Наприклад, маємо порівняння дата-сету DS_E (див. рис.29). З репорту можна побачити, що кількість записів однакова, але є невідповідність в атрибутах для змінних, тобто інша довжина чи змінений заголовок (Folder, StudyEnvSireNumber мають різну довжину); також, була видалена одна змінна з дата-сету з першої директорії.

The COMPARE Procedure
 Comparison of WORK.DS_E with WORK.DS_E_
 (Method=RELATIVE(2.22E-09), Criterion=0.00001)

Data Set Summary

| Dataset | Created | Modified | NVar | NObs |
|------------|------------------|-----------------------------|-----------------|-----------------|
| WORK.DS_E | 17NOV20:12:13:40 | 17NOV20:12:13:40 | 49 | 1 |
| WORK.DS_E_ | 17NOV20:12:13:48 | 17NOV20:12:13:48 | 50 | 1 |

Variables Summary

Number of Variables in Common: 49.
 Number of Variables in WORK.DS_E_ but not in WORK.DS_E: 1.
 Number of Variables with Differing Attributes: 2.

Listing of Common Variables with Differing Attributes

| Variable | Dataset | Type | Length | Label |
|-------------------------------|------------|------|--------|-------------------------------|
| Folder | WORK.DS_E | Char | 50 | Folder OID |
| | WORK.DS_E_ | Char | 150 | Folder OID |
| StudyEnvSiteNumber | WORK.DS_E | Char | 20 | StudyEnvSiteNumber |
| | WORK.DS_E_ | Char | 60 | StudyEnvSiteNumber |

Observation Summary

| Observation | Base | Compare |
|----------------------|------|---------|
| First Obs | 1 | 1 |
| First Unequal | 1 | 1 |
| Last Unequal | 1 | 1 |
| Last Obs | 1 | 1 |

Number of Observations in Common: 1.
 Total Number of Observations Read from WORK.DS_E: 1.
 Total Number of Observations Read from WORK.DS_E_: 1.

Number of Observations with Some Compared Variables Unequal: 1.
 Number of Observations with All Compared Variables Equal: 0.

Рисунок 29. Файл .rtf, приклад звіту порівняння

Нижче наведемо приклад порівняння значень в змінних з дата-сету DS_E (див. рис.30). Тут Folderid SaveTS – це «робочі» змінні, які зазвичай ми видаляємо при порівнянні, адже вони не мають жодного впливу на самі значення.

| Internal id for the folder | | | | |
|----------------------------|------------------|---------------------|--------|----------|
| Obs | Base folderid | Compare folderid | Diff. | % Diff |
| 1 | 265125 | 206337 | -58788 | -22.1737 |

| Timestamp of last save in clinical views | | | | |
|--|----------------|-------------------|-----------|---------|
| Obs | Base SaveTS | Compare SaveTS | Diff. | % Diff |
| 1 | 1.91911E9 | 1.89116E9 | -27952616 | -1.4565 |

Рисунок 30. Файл .rtf, приклад порівняння змінних

З порівняння змінної FLAG_STD (див. рис.31) можна зробити висновок, що значення істотно змінилися або ж були додані нові записи чи було змінено сортування. На такі невідповідності потрібно звертати увагу, далі ще раз перевірити їх власноруч та надсилати запит до сайту.

| FLAG Coded Value | | |
|------------------|------------------------|---------------------------|
| Obs | Base Value FLAG_STD | Compare Value FLAG_STD |
| 10 | Y | NA |
| 11 | N | NA |
| 14 | N | Y |
| 15 | Y | N |
| 16 | Y | N |
| 24 | Y | N |

Рисунок 31. Файл .rtf, приклад порівняння змінної FLAG_STD

Розглянемо на завершення приклад, коли зміни в даних є істотними та важливими, тобто коли з'являються чи зникають певні показники або ж коли змінюються показники лабораторних даних (див. рис.32).

```

The COMPARE Procedure
Comparison of WORK.LB_COAG with WORK.LB_COAG_
(Method=RELATIVE(2.22E-09), Criterion=0.00001)

Data Set Summary

Dataset              Created              Modified              NVar  NObs
-----              -
WORK.LB_COAG         17NOV20:12:13:44   17NOV20:12:13:44   62    22
WORK.LB_COAG_       17NOV20:12:13:52   17NOV20:12:13:52   63    22

```

```

|| Coagulation - Prothrombin Time(RAW)
|| Base Value          Compare Value
Obs || LBPT_RAW          LBPT_RAW
-----+-----+
||
3 || ND

```

```

|| Coagulation - Activated PT Time (%) (RAW)
|| Base Value          Compare Value
Obs || LBAPTTQ_RAW        LBAPTTQ_RAW
-----+-----+
||
8 || ND

```

```

|| Coagulation - Prothrombin Time (%) (RAW)
|| Base Value          Compare Value
Obs || LBPTQ_RAW          LBPTQ_RAW
-----+-----+
||
8 || ND

```

```

|| Result or Finding in Original Units
|| Base Value          Compare Value
Obs || LBORRES            LBORRES
-----+-----+
||
134 || 1                50
135 || 50               1
457 || 150            1
458 || 1              150
1151 || 90.0          22.0
1152 || 22.0         90.0
3346 || 20            1
3347 || 1            20
4141 || 2            40
4142 || 40           2
6738 || 4.5          4.4
6739 || 4.4          4.5
7502 || 20.0         70.0
..

```

Рисунок 32. Файл .rtf, приклад порівняння LB дата-сету

Висновки

Основним результатом даної кваліфікаційної роботи бакалавра є оригінальний програмний застосунок для автоматизації порівняння дата-сетів з різних директорій, розроблений із використанням фреймворку LSAF та пакету SAS. Впровадження такого застосунку дало змогу підвищити ефективність роботи програмістів та статистиків, автоматизувати перевірку на точність даних, що надходять в систему.

В процесі виконання даної роботи було поглиблено знання пакету SAS, досліджено вже існуючі технології та макроси, розроблені для фреймворку LSAF, розроблено нові алгоритми для порівняння дата-сетів, написані із застосуванням пакету SAS.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Освоюємо R [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://uk.wikibooks.org/wiki/>.
2. Ethan Heinzen, Ryan Lennon, Andrew Hanson The comparedf function [Електронний ресурс] / Ethan Heinzen, Ryan Lennon, Andrew Hanson – Режим доступу до ресурсу: <https://cran.r-project.org/web/packages/arsenal/vignettes/comparedf.html>.
3. Costas Andreou How to Quickly Compare Data Sets [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://towardsdatascience.com/how-to-quickly-compare-data-sets>.
4. What is LSAF [Електронний ресурс] – Режим доступу до ресурсу: <https://ocs-lifesciences.com/what-is-lsaf/>
5. SAS/STAT Software [Електронний ресурс] – Режим доступу до ресурсу: https://www.sas.com/en_us/software/stat.html
6. Statistical Analysis Software (SAS) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.statisticssolutions.com/statistical-analysis-software-sas/>
7. Support / Samples & SAS notes [Електронний ресурс] – Режим доступу до ресурсу: <https://support.sas.com/kb/60/264.html>

ДОДАТКИ

Додаток А. Довідка про впровадження

ДОВІДКА

Видана студентці факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка Рудзей Вікторії Олександрівні в тому, що результати виконання її кваліфікаційної роботи бакалавра впроваджені на підприємстві «КОВАНС КЛІНІКАЛ РІСЕРЧ».

Зокрема, на підприємстві використовується запропонована методика обробки та порівняння даних, використовуючи статистичний пакет SAS.

Планується подальше виконання робіт щодо розширення функціоналу програми та покращення засобів зображення аналітичних результатів.

Директор

23 квітня 2021 р.



О. В. Рябцев

COVANCE CLINICAL RESEARCH UKRAINE LLC
6 Oleny Telihy str., Building 6,
Kyiv, 04112, Ukraine
■ +380 44 594 64 44

COVANCE.
by labcorp

Додаток Б. Реалізація коду для дизайну файлу

```
proc template;
  define style Styles.Custom;
  parent = Styles.RTF;
  STYLE SystemTitle /
  FONT_FACE = "Comic Sans MS"
  FONT_SIZE = 3
  FONT_WEIGHT = medium
  FONT_STYLE = roman
  FOREGROUND = midnightblue
  BACKGROUND = white
  CELLSPACING = 0
  CELLPADDING = 0;
  STYLE SystemFooter /
  FONT_FACE = "Comic Sans MS"
  FONT_SIZE = 3
  FONT_WEIGHT = bold
  FONT_STYLE = italic
  FOREGROUND = midnightblue
  BACKGROUND = white
  CELLSPACING = 0
  CELLPADDING = 0;
  STYLE Header /
  FONT_FACE = "Times New Roman"
  FONT_SIZE = 4
  FONT_WEIGHT = medium
  FONT_STYLE = roman
  FOREGROUND = midnightblue
  BACKGROUND = white
  CELLSPACING = 0
  CELLPADDING = 0;
  STYLE Data /
  FONT_FACE = "Times New Roman"
  FONT_SIZE = 3
  FONT_WEIGHT = medium
  FONT_STYLE = roman
  FOREGROUND = black
  BACKGROUND = white
```

```
CELLSPACING = 0
CELLPADDING = 0;
STYLE Table /
FOREGROUND = black
BACKGROUND = white
CELLSPACING = 0
CELLPADDING = 0
FRAME = HSIDES
RULES = NONE;
STYLE Body /
FOREGROUND = black
BACKGROUND = white
TOPMARGIN=0.1in
BOTTOMMARGIN=0.1in
LEFTMARGIN = 0.1in
RIGHTMARGIN = 0.1in;
STYLE SysTitleAndFooterContainer /
CELLSPACING=0;

end;
run;
```