МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В.Н.Каразіна
Факультет математики і інформатики
Кафедра теоретичної та прикладної інформатики

Кваліфікаційна робота

магістр

на тему Порівняння мов програмування SAS та R в області клінічних досліджень

Виконав: студент 2 курсу, групи МФ-61
спеціальність 122 «Комп'ютерні науки»
освітньо-наукова програма
«Інформатика»

Юскевич В.А.
(прізвище та ініціали)
Керівник Узлов Д.Ю.
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)

Харків – 2024 року

Зміст

1 ВСТУП

Кожен із нас хоч раз в житті стикався з необхідністю приймати лікарські засоби, будь то сироп від кашлю, таблетки від головного болю або звичайний вітамінний комплекс. Але як часто ви замислювались над тим, який шлях лікарській засіб проходить з моменту його винаходження до моменту появи на полицях аптек? Скільки часу це займає і взагалі як це все відбувається? Відповіддю на всі ці питання є два слова «клінічні дослідження». Клінічне дослідження, яке може бути будь-яким випробуванням, проводиться на людях з метою виявлення або перевірки фармакологічних та/або інших фармакодинамічних властивостей досліджуваного препарату чи препаратів. Його основною метою є оцінка впливу цих препаратів на клінічні симптоми захворювання, виявлення можливих побічних реакцій та дослідження процесів абсорбції, розподілу, метаболізму та виведення. Проведення таких досліджень спрямоване на підтвердження безпеки та/або ефективності досліджуваних препаратів. Результати будь якого клінічного дослідження – це в першу чергу величезний об’єм даних, які треба обробити швидко, ефективно та безпечно. На сьогодні існує два основних інструменти для обробки, аналізу та візуалізації даних клінічних досліджень – це SAS та R.

Метою цієї роботи є глибоке порівняння двох основних мов програмування у сфері клінічних досліджень – SAS і R, з ціллю з'ясування їхнього функціоналу, ефективності та придатності для використання в практичних сценаріях клінічних досліджень. Основними аспектами порівняння будуть аналіз методів обробки даних, статистичних методів, можливостей візуалізації даних, продуктивності та доступності програмних інструментів, вартості використання та спільноти користувачів.

Для виконання цієї роботи було поставлено наступні задачі:

1. Проаналізувати доступність та вартість використання SAS і R у контексті клінічних досліджень.

2. Розглянути функціонал інструментів для статистичного аналізу в обох мовах.
3. Розглянути рекомендації регуляторних органів до обох мов
4. Порівняти можливості SAS і R щодо обробки даних клінічних досліджень.
5. Порівняти можливості SAS і R щодо аналізу даних клінічних досліджень.
6. Порівняти можливості SAS і R щодо візуалізації даних у SAS і R в контексті клінічних досліджень.
7. Сформулювати рекомендації щодо вибору між SAS і R для конкретних завдань у клінічних дослідженнях.

Актуальність теми

Тема порівняння мов програмування SAS і R у сфері клінічних досліджень є актуальною та важливою для розвитку наукової спільноти в галузі клінічної медицини та статистики. На основі аналізу вітчизняної та зарубіжної наукової літератури можна виокремити кілька ключових аспектів, що підтверджують актуальність даної теми:

1. Значення у розвитку наукової галузі: Сфера клінічних досліджень постійно розвивається, аналіз та обробка даних стають все більш складними та об'ємними. Використання відповідних програмних інструментів для аналізу та візуалізації даних є ключовим елементом в цьому процесі.
2. Практично розв'язані завдання: Дослідники та фахівці зі сфери клінічних досліджень щодня стикаються з питанням вибору мови програмування для обробки та аналізу даних. Порівняння SAS і R надасть їм можливість краще розуміти переваги та недоліки кожної з цих мов і вибирати оптимальний інструмент для своїх потреб.
3. Актуальні проблеми у сфері: Враховуючи зростання обсягу даних, складність аналізу та необхідність дотримання стандартів регулюючих

органів, клінічні дослідження стикаються з численними викликами. Використання ефективних та надійних інструментів програмування для обробки цих даних стає критичним аспектом роботи у цій галузі.

4. Аналіз наукової літератури: Попередні дослідження в цій області вказують на значний інтерес до порівняння SAS і R. Проте більшість з них надає лише загальні висновки, тому детальний аналіз цих даних з точки зору практичних застосувань та актуальних проблем є важливим кроком у подальшому розвитку галузі.

Отже, порівняння мов програмування SAS і R у сфері клінічних досліджень відповідає потребам сучасної наукової спільноти та вирішує актуальні проблеми, що існують у цій галузі. Він надає важливий внесок у покращення якості та ефективності клінічних досліджень, що впливає на здоров'я та добробут пацієнтів.

Стислий огляд відомих результатів в області дослідження

Ефективність та швидкодія**:** Дослідження, проведене Джоном Смітом та співавторами у 2020 році, показало, що «швидкість обробки даних у SAS може бути вищою порівняно з R у деяких конкретних випадках» [1]. Однак, детальніший аналіз [2] показав, що «R демонструє велику ефективність в обробці великих обсягів даних, особливо за допомогою векторизації та оптимізації коду». Більш того, дослідження Бена Кларка та його колег у 2019 році вказує на те, що «швидкодія R може бути значно покращена за допомогою використання паралельних обчислень та оптимізації алгоритмів» [3].

Крім того, робота Емілі Міллер та співавторів у 2021 році привернула увагу до того, що «SAS може ефективно використовуватись в умовах обмеженої доступності ресурсів, так як він оптимізован для виконання на високопродуктивних серверах» [4]. Однак, вони також відзначили, що «з розвитком технологій обробки даних із збільшенням кількості паралельних обчислень, R демонструє конкурентоспроможність у боротьбі з великими обсягами інформації».

Функціонал для статистичного аналізу**:** Робота Кетрін Лі та її колег у 2019 році підкреслює, що «SAS має багато вбудованих процедур для статистичного аналізу даних у клінічних дослідженнях, що дозволяє зручно та ефективно проводити аналіз» [5]. Однак, дослідження Суман Рао та інших у 2021 році вказує на те, що «R може бути більш гнучким у застосуванні новітніх статистичних методів через широкий вибір пакетів та розвиту відкритого співтовариства» [6]. Крім того, дослідження Марії Гарсіа та співавторів у 2020 році показало, що «R дозволяє швидко реалізувати нові методи та алгоритми за допомогою активного співтовариства та відкритих джерел» [7]. Нижче наведено порівняльну таблицю вже відомих результатів в області дослідження.

Критерій	SAS	R
Навчання та підтримка	Офіційна документація та навчальні матеріали, спеціально орієнтовані на клінічні дослідження, а також комерційна підтримка	Академічні курси та онлайн-ресурси, спільнота користувачів для обміну досвідом та допомоги
Платформа	Доступний на різних операційних системах: Windows, Linux, Unix	Сумісний з більшістю операційних систем: Windows, Linux, MacOS
Масштабованість	Підтримує роботу з великими обсягами даних та масштабується для використання великими командами	Підходить як для невеликих, так і для великих дослідницьких проектів

Критерій	SAS	R
Швидкодія	Зазвичай пропонує гарну швидкодію завдяки компіляції виконання та ефективному використанню пам'яті	При правильному використанні може бути швидким, але може виникнути проблеми з швидкодією для обробки великих обсягів даних
Безпека	Має функції для забезпечення безпеки даних, такі як шифрування та контроль доступу	Безпека даних може залежати від встановленого середовища та дотримання відповідних практик безпеки
Поріг входження	Для початківців може бути складним через високий поріг входження та необхідність професійної підготовки	Зазвичай має більш низький поріг входження, адже доступний безкоштовно та має широку спільноту користувачів, яка надає підтримку та допомогу

Таблиця 1.1 Порівняльна таблиця SAS та R за вже відомими результатами дослідження

Методи дослідження

Основним методом дослідження є порівняння цих двох мов програмування в середовищі розробки SAS Base 9.4 та R 4.3.3. Існують багато критеріїв, за якими порівнюють мови програмування, наприклад: гнучкість, безпека, швидкодія та інші. Мною було обрано сім основних критеріїв порівняння, це – доступність, вартість, регуляторна підтримка, функціонал інструментів, гнучкість, можливості статистичного аналізу, візуалізація. Для

порівняння за першими трьома критеріями було проведено літературний огляд. Для порівняння за рештою критеріїв було отримано тестові дані клінічних випробувань та відтворено шлях, який проходять такі дані в реальній практиці, розбитий на три основних етапи: обробка даних, аналіз даних та візуалізація отриманих результатів.

Робота складається зі вступу, трьох глав, загальних висновків, списку використаної літератури (27), додатків (3). Зміст роботи висвітлено на 42 сторінках основного тексту і містить 2 таблиці і 13 рисунків.

2 ЛІТЕРАТУРНИЙ ОГЛЯД

2.1 Історія SAS та R. Аналіз доступності та вартості

SAS

Мова програмування SAS (Statistical Analysis System) виникла в середині 1960-х років у США на базі дослідницьких проектів університету Стенфорда. Початково SAS розроблялася для здійснення аналізу даних з метою поліпшення ефективності управлінських рішень в галузі бізнесу та фінансів. Більш детально це описано в [8].

Протягом наступних десятиліть SAS стала однією з провідних мов програмування для аналізу даних у різних галузях, включаючи науку про здоров'я. Застосування SAS у клінічних дослідженнях почало набувати великого значення з появою програмних засобів, спеціально розроблених для аналізу медичних даних, як зазначено в [9].

Система SAS включає широкий набір функцій та процедур, що сприяє проведенню аналізу ефективності лікування, порівнянню різних методів терапії, оцінці статистичної значущості результатів та побудові прогнозів. Крім того, SAS використовується для створення стандартів даних у клінічних дослідженнях, що забезпечує консистентність та якість даних, отриманих у різних медичних центрах та лабораторіях.

Платформа SAS доступна для різних операційних систем, таких як Windows, macOS та Linux. Однак доступність може бути обмеженою через ліцензійні обмеження та вимоги до обладнання. SAS пропонує різні види ліцензій, включаючи підписку, одноразову покупку та модель сплачення залежно від використання. Вартість ліцензії може бути значною і зазвичай залежить від обсягу функціональності та підтримки, яку компанія бажає отримати. Ціни на програмне забезпечення SAS надаються по запиту через їхніх представників або на їхньому веб-сайті. SAS також пропонує навчальні

курси, сертифікаційні програми та технічну підтримку для користувачів. Проте ці послуги можуть бути додатковими витратами.

R

У книзі [10] можна дізнатись, що мова програмування R має цікаву історію, що почалася у 1990-х роках. R розроблена групою статистиків і аналітиків, які працювали у Новозеландському інституті досліджень науково-прикладної статистики (NZISS), який пізніше став відомий як Університет Акленда. Перші версії R були створені як програмне забезпечення для статистичних обчислень та візуалізації даних.

R базується на мові програмування S, яка також виникла у Новозеландському інституті досліджень науково-прикладної статистики. Однак, пізніше R стала незалежною мовою з відкритим вихідним кодом та широким спектром функцій, що дозволило їй стати популярним інструментом у багатьох галузях, включаючи аналіз даних у клінічних дослідженнях, як зазанчаєє автор [11].

Проаналізувавши [12] можна сказати, що, у сфері клінічних досліджень R займає важливе місце. Вона використовується для аналізу клінічних даних, оцінки ефективності лікування, порівняння різних методів терапії, моделювання медичних показників та інших аспектів досліджень у галузі медицини.

Завдяки своїй потужності та гнучкості, R дозволяє дослідникам та медичним спеціалістам ефективно аналізувати великі обсяги даних, проводити складні статистичні обчислення та візуалізувати результати досліджень.

R підтримується на різних операційних системах, а також має широке співтовариство користувачів та розробників, що дозволяє швидко отримати доступ до різних розширень та пакетів для аналізу даних. R є вільним програмним забезпеченням з відкритим вихідним кодом, що означає, що вона безкоштовно доступна для використання. Однак, при використанні в

комерційних проектах можуть виникати витрати на підтримку, навчання персоналу та розробку.

2.2 Функціонал інструментів для статистичного аналізу

SAS

PROC (Procedure): PROC є основним засобом для виконання статистичного аналізу в SAS. Він використовується для виконання різноманітних операцій аналізу даних, включаючи описову статистику, регресійний аналіз, аналіз дисперсії, класифікацію, кластерний аналіз і багато іншого. Наприклад, PROC MEANS використовується для розрахунку основних статистичних характеристик, таких як середнє, медіана, стандартне відхилення, максимум, мінімум тощо.

DATA Step: DATA Step - це основний інструмент для маніпулювання даними в SAS. Він дозволяє виконувати операції зчитування, збереження, трансформації та обробки даних. Наприклад, з допомогою DATA Step можна створити нову змінну в наборі даних, обчислити значення за певною формулою або фільтрувати дані за певними умовами.

Функції: SAS має вбудований набір функцій для виконання різноманітних операцій з даними, включаючи арифметичні, логічні, рядкові та статистичні функції. Наприклад, функція SUM використовується для підсумовування значень змінної.

Формати та макроси: SAS має потужну систему форматів, яка дозволяє змінювати вигляд та інтерпретацію значень змінних. Макроси - це інструмент для автоматизації та повторного використання коду в SAS. Вони дозволяють створювати параметризовані блоки коду, які можуть бути використані повторно в різних частинах програми.

Бібліотеки даних: Бібліотеки даних SAS використовуються для збереження та організації даних. Вони можуть містити один або кілька наборів даних, які можуть бути легко доступні для аналізу та обробки.

Це лише декілька основних аспектів функціоналу SAS для статистичного аналізу.

R

Функції: R надає велику кількість вбудованих та зовнішніх функцій для виконання різноманітних операцій з даними та статистичного аналізу. Наприклад, функція mean() обчислює середнє значення вектору, функція lm() використовується для побудови моделей лінійної регресії.

Пакети: R використовує пакети для організації функцій та додаткового функціоналу. Існують тисячі пакетів, які покривають різні аспекти аналізу даних, візуалізації, машинного навчання тощо. Наприклад, пакет dplyr надає набір функцій для маніпулювання даними, а ggplot2 дозволяє створювати візуалізації.

Візуалізація: R є потужним інструментом для створення візуалізацій даних. Пакети, такі як ggplot2, lattice, plotly, дозволяють будувати різноманітні типи графіків. Наприклад, за допомогою ggplot2 можна легко побудувати стовпчаті діаграми, графіки розсіювання, лінійні графіки тощо.

Статистичні функції: У R доступний широкий набір статистичних функцій для проведення різноманітних аналізів даних, включаючи описову статистику, тестування гіпотез, аналіз варіації тощо. Наприклад, функція t.test() використовується для проведення t-тесту для порівняння середніх значень двох груп.

Моделювання: R підтримує різноманітні методи моделювання, включаючи лінійну регресію, логістичну регресію, дерева класифікації, методи машинного навчання тощо. Наприклад, пакет caret надає інтерфейс для навчання моделей машинного навчання.

2.3 Регуляторна підтримка

SAS

Широко використовується у сфері клінічних досліджень через свою високу репутацію як надійний інструмент для обробки та аналізу даних. Багато фармацевтичних компаній та медичних установ залучають SAS у свої дослідження через його широкий функціонал та відповідність стандартам регуляторних органів. SAS надає спеціальні пакети та процедури для аналізу клінічних даних, які допомагають виконати потрібні аналізи та підготувати дані для подання до регуляторних органів, таких як FDA (Адміністрація з контролю за харчовими продуктами та ліками у США) та EMA (Європейське агентство з лікарських засобів). Наприклад FDA випускає ряд документів «Guidance for Industry» [24], які містять рекомендації щодо виконання клінічних випробувань та подання результатів для оцінки. Багато з цих документів включають поради щодо аналізу даних та використання спеціалізованих програмних засобів, таких як SAS, приміром «Надання регуляторних заявок у електронному форматі» [25].

R

Насправді, R не має офіційної регуляторної підтримки, такої як SAS. Однак все більше використовується в академічних та дослідницьких галузях для аналізу клінічних даних через свою безкоштовність, гнучкість та широкий вибір пакетів для статистичного аналізу. В багатьох випадках, коли дослідження виконуються за науковими цілями або в невеликому масштабі, використання R може бути прийнятним. Але для досліджень, які потребують подання результатів до регуляторних органів, може вимагатися додаткова валідація та документація для забезпечення відповідності вимогам.

Отже, хоча SAS має більшу регуляторну підтримку і широке використання у клінічних дослідженнях, використання R може бути прийнятним у деяких випадках, особливо у дослідницьких проектах. Так,

наприклад 27 вересня 2023 року робоча група з подачі R повідомила у себе на сайті [26], що успішно завершила другий етап пілотного проекту на основі R shiny та отримала лист-відповідь від FDA CDER [27].

3 РОЗРОБКА ПРОГРАМ

3.1 Порівняння можливостей SAS і R щодо обробки даних клінічних досліджень.

Для порівняння можливостей SAS і R щодо обробки та аналізу даних клінічних досліджень, було використано тестові дані SDTM (Study Data Tabulation Model), а саме доменів AE (Adverse Events) в якому зберігається інформація щодо всіх небажаних сторонніх ефектів, які виникли після прийому препарату, як вказано в [13] (див. розділ 6.2.1) та SUPPAE (Supplemental Qualifiers for AE), в якому зберігається додаткова інформація, яка не відповідає стандартам CDISC (Clinical Data Interchange Standards Consortium) [13] (див. розділ 8.4.1) а також тестові дані ADaM (Analysis Data Model) – домени ADSL (Subject-Level Analysis Dataset), який містить 1 запис на суб’єкта, незалежно від типу дизайну клінічного випробування. ADSL — «Набір даних аналізу предметного рівня». [14] (див. розділ 2.3.1) та ADTTE (ADaM Time-to-Event) – набір даних для аналізу часу до настання події. [15].

На першому етапі порівняння було обрано п’ять базових маніпуляцій, які частіше за все використовуються при обробці даних клінічних досліджень, а саме: завантаження, сортування, транспонування, об’єднання та вивантаження. Кожен з вище зазначених наборів даних містить певну кількість записів (строк) та змінних (колонок): AE містить 258290 записів та 21 змінну, SUPPAE містить 506172 записів та 8 змінних, ADSL містить 12345 записів та 15 змінних, ADTTE містить 12345 записів та 10 змінних.

SAS

Для завантаження тестових даних, було використано процедуру IMPORT, яка, як зазначено в [16] (див. розділ Proc Import Statement) «Імпортує зовнішній файл даних до набору даних SAS». Так як тестові дані зберігаються в файлах із розширенням .csv, було використано підхід, який запропоновано в

[16] (див. розділ Example 4: Importing a Comma-Delimited File with a CSV Extension). DATAFILE зчитує файл із зазначеного шляху, OUT створює вихідний набір даних, REPLACE перезаписує набір даних, якщо він вже був створений раніше, GUESSINGROWS визначає кількість рядків файлу, які необхідно просканувати для визначення відповідного типу даних та довжини для змінних. Нижче наведено програмний код, за допомогою якого було імпортовано тестові дані у середу розробки:

```
proc import datafile='C:\Users\VYuskevych\UP\2023\Project\Input\csv\ae.csv'
    out=ae
    dbms=csv
    replace;
    guessingrows=1000;
run;

proc import
datafile='C:\Users\VYuskevych\UP\2023\Project\Input\csv\suppae.csv'
    out=suppae
    dbms=csv
    replace;
    guessingrows=1000;
run;

proc import datafile='C:\Users\VYuskevych\UP\2023\Project\Input\csv\adsl.csv'
    out=adsl
    dbms=csv
    replace;
    guessingrows=1000;
run;
```

Для сортування наборів даних прийнято використовувати процедуру SORT, яка, як зазначено в [16] (див. розділ Proc Sort Statement) «Упорядковує спостереження набору даних SAS за значеннями однієї чи кількох символьних чи числових змінних». В команда BY передають змінні, за якими необхідно відсортувати набір даних. Нижче наведено програмний код, за допомогою якого було відсортовано набори тестових даних:

```
proc sort data=suppae;
      by USUBJID AESEQ;
run;

proc sort data=ae;
      by USUBJID AESEQ;
run;
```

Для транспонування наборів даних використовується процедура TRANSPOSE. Ця процедура «Створює вихідний набір даних, переструктуруючи значення в наборі даних SAS, транспонуючи обрані змінні в спостереження» [16] (див. розділ Proc Transpose Statement). В команду ID передається змінна, значення якої будуть іменами нового (транспонованого) набору даних. В команду IDLABEL передається змінна, значення якої будуть лейблами нового (транспонованого) набору даних. В команду VAR передаються всі змінні, які необхідно транспонувати. Нижче наведено програмний код, за допомогою якого було транспоновано набір даних SUPPAE, для подальшого об’єднання із набором даних AE:

```
proc transpose data=suppae out=suppae_tr;
      by USUBJID AESEQ;
      id QNAM;
      idlabel QLABEL;
      var QVAL;
run;
```

Для об’єднання двох наборів даних AE та SUPPAE було використано оператор MERGE. «Оператор MERGE об'єднує спостереження з двох або більше наборів даних SAS» [16] (див. розділ Merge Statement). Після об'єднання AE та SUPPAE було отримано набір даних AE_SUPPAE, який був потім об’єднаний із набором даних ADSL для створення нового набору даних ADAE. У обох випадках використовуються оператори IN, BY та IF. Оператор IN «Створює логічну змінну, яка вказує, чи сприяв набір даних поточному спостереженню» [16] (див. розділ IN= Data Set Option). Оператор BY, який «Контролює операції SET, MERGE, MODIFY або UPDATE в операторі DATA та налаштовує спеціальні змінні групування» [16] (див. розділ BY Statement), тобто створює ключ. Оператор IF, за допомогою якого у першому випадку, ми залишаємо всі записи з набору даних AE та записи, що перетинаються за ключем, з набору даних SUPPAE, у другому випадку залишаємо тільки ті записи з обох наборів даних AE_SUPPAE та ADSL, що перетинаються за

ключем. Нижче наведено програмний код, який було використано для отримання AE_SUPPAE та ADAE:

```
data ae_suppae;
      merge ae(in=a)
               suppae_tr(in=b);
      by USUBJID AESEQ;
      if a;
run;

data adae;
      merge ae_suppae(in=a)
            adsl(in=b);
      by USUBJID;
      if a and b;
run;
```

Останньою базовою маніпуляцією було вивантаження набору даних ADAE у зовнішній файл із розширенням .csv, для проведення подальшого дослідження. Для вивантаження набору даних ADAE було використано процедуру EXPORT, яка є дуже схожою за своїм синтаксисом із процедурою IMPORT. Так як було обрано вивантажити набір даних ADAE у зовнішній файл із розширенням .csv, було використано підхід, який запропоновано в [16] (див. розділ Example 2: Exporting a Subset of Observations to a CSV File). Нижче наведено програмний код, за допомогою якого було експортовано дані у зовнішній файл:

```
proc export data=adae
    outfile='C:\Users\VYuskevych\UP\2023\Project\SAS\Output\adae.csv'
    dbms=csv
    replace;
run;
```

R

Всі ті самі базові маніпуляції було проведено за допомогою мови програмування R. Для завантаження тестових даних було використано функцію READ, яка, як зазначено в [17] (див. розділ read.table: Data Input) «Читає файл у форматі таблиці та створює з нього фрейм даних, де кожен випадок відповідає рядку, а змінні відповідають полям у файлі». Так як тестові

дані зберігаються у файлах з розширенням .csv, було використано варіант функції read.csv. Нижче наведено програмний код, за допомогою якого були імпортовані тестові дані:

```
    ae <- read.csv(«C:/Users/VYuskevych/UP/2023/Project/Input/csv/ae.csv»)

    suppae <-
read.csv(«C:/Users/VYuskevych/UP/2023/Project/Input/csv/suppae.csv»)

    adsl <-
read.csv(«C:/Users/VYuskevych/UP/2023/Project/Input/csv/adsl.csv»)
```

Для сортування було обрано використати функцію ORDER, яка «Повертає перестановку, яка перегруповує перший аргумент у зростаючому або спадному порядку, розриваючи зв'язки за допомогою подальших аргументів.» [17] (див. розділ order: Ordering Permutation). Нижче наведено програмний код за допомогою якого було відсортовано набори тестових даних:

```
suppae <- suppae[order(suppae$USUBJID, suppae$AESEQ), ]

ae <- ae[order(suppae$USUBJID, suppae$AESEQ), ]
```

Для транспонування було використано функцію RESHAPE та підхід запропонований в [17] (див. розділ reshape: Reshape Grouped Data). «Ця функція перетворює формат фрейму даних з «широкого», де повторні вимірювання знаходяться в окремих стовпцях одного запису, у «довгий» формат, де повторні вимірювання розміщені в окремих записах» [17] (див. розділ reshape: Reshape Grouped Data). В IDVAR передаються змінні, за якими треба виконати транспонування, в TIMEVAR передається змінна, значення якої будуть іменами нового (транспонованого) набору даних. Через специфіку функції RESHAPE потрібно було провести додаткові маніпуляції з даними, а саме: залишити тільки необхідні змінні, перейменувати змінні у відповідності до стандартів CDISC [14], за допомогою функції COLNAMES [17] (див. розділ colnames: Row and Column Names). Нижче наведено програмний код, за

допомогою якого було транспоновано набір даних SUPPAE, для подальшого об’єднання із набором даних AE:

```
suppae_tr <- reshape(suppae, idvar = c(«USUBJID», «AESEQ»), timevar = «QNAM», direction = «wide»)

suppae_tr <- suppae_tr[, c(«USUBJID», «AESEQ», «QVAL.TRTEMFL», «QVAL.AESEVCD»)]
colnames(suppae_tr)[colnames(suppae_tr) == «QVAL.TRTEMFL»] <- «TRTEMFL»

colnames(suppae_tr)[colnames(suppae_tr) == «QVAL.AESEVCD»] <- «AESEVCD»
```

Для об’єднання двох наборів даних AE та SUPPAE було використано функцію MERGE. Ця функція «Об'єднує два фрейми даних за спільними стовпцями або назвами рядків» [17] (див. розділ merge: Merge Two Data Frames). Так само як було зазначено вище, ми створюємо набір даних AE_SUPPAE, якій об’єднуємо з ADSL для отримання набору даних ADAE. У першому випадку ми об’єднуємо два набори даних за ключем та залишаємо всі записи з AE, а у другому випадку також за ключем, але залишаємо тільки ті записи, які пересікаються в обох наборах даних. Обидва підходи можна знайти в [17] (див. розділ merge: Merge Two Data Frames). Нижче наведено програмний код:

```
ae_suppae <- merge(ae, suppae_tr, by = c(«USUBJID», «AESEQ»))

adae <- merge(ae_suppae, adsl, by = «USUBJID»)
```

Остання базова маніпуляція, як і у прикладі з SAS – це вивантаження набору даних ADAE у зовнішній файл із розширенням .csv, для проведення подальшого дослідження. Для цього було використано функцію WRITE. Ця функція «зберігає зведені характеристики розділених рівнів селекції в CSV-файли на диск для подальшого аналізу або обробки іншим програмним забезпеченням, або просто для збереження (резервного копіювання) результатів» [17] (див. розділ write.csv: write.csv.R). Ця функція, як і у випадку з SAS дуже схожа за своїм синтаксисом на функцію READ, яку ми вже

використовували. Нижче наведено програмний код, за допомогою якого було експортовано дані у зовнішній файл:

```
write.csv(adae, file =
«C:/Users/VYuskevych/UP/2023/Project/R/Output/adae.csv», row.names = FALSE)
```

3.2 Порівняння можливостей SAS і R щодо аналізу даних клінічних досліджень.

Для наступного етапу порівняння цих двох мов програмування було проведено аналіз даних отриманих на минулому етапі (набір даних ADAE), а саме проаналізувати частоту появи небажаних сторонніх ефектів для кожного класу системи органів всередині кожної групи пацієнтів, тих хто приймає ліки, тих хто приймає плацебо, та загальна кількість, незалежно від препарату, але тільки ті, хто прийняв хоча б одну дозу препарату. Після цього було знайдено тривалість небажаних сторонніх ефектів особливого інтересу, яка зберігається в змінній AEDUR. У нашому випадку таким ефектом є будь який головний біль. Записи з цим ефектом позначені «Y» в змінній AESI.

SAS

Для початку були завантажені зовнішні набори даних ADAE та ADTTE, які зберігаються у файлах із розширенням .csv, для цього як і на попередньому етапі було використано процедуру IMPORT, в якій за допомогою опції WHERE для набору даних ADAE було відібрано тільки тих пацієнтів, які прийняли хоча б одну дозу препарату. Такі записи позначені «Y» в змінній SAFFL. Потім дані було підготовано для аналізу за допомогою базових маніпуляцій, які вже використовувались на минулому етапі. За допомогою об’єднання двох однакових наборів даних ADAE було отримано додаткові записи на кожного суб’єкта без прив’язки до препарату та відсортовано за змінними, по яким буде будуватись аналіз. Нижче наведено програмний код попередньої обробки даних:

```
proc import
datafile='C:\Users\VYuskevych\UP\2023\Project\Input\csv\adtte.csv'
    out=adtte
    dbms=csv
    replace;
      guessingrows=1000;
run;

proc import
datafile='C:\Users\VYuskevych\UP\2023\Project\SAS\Output\adae.csv'
```

```
    out=adae(where=(SAFFL = «Y»))
    dbms=csv
    replace;
    guessingrows=1000;
run;

data adae;
    set adae(in=a)
        adae(in=b);
    if b then TRT01A = «ALL»;
run;

proc sort data=adae;
    by TRT01A;
run;
```

Після попередньої обробки даних було проведено аналіз частоти появи небажаних сторонніх ефектів для кожного класу системи органів всередині кожної групи пацієнтів. Для цього використовується процедура FREQ, яка «генерує одно-, дво-, або n-вимірні таблиці частот та контингентності (хрест-таблиці)». Для двовимірних таблиць, PROC FREQ обчислює тести та міри асоціації. Для n-вимірних таблиць, PROC FREQ забезпечує стратифікований аналіз, обчислюючи статистику в межах страт та між стратами» [16] (див. розділ The FREQ Procedure). За замовчення результат цієї процедуру виводиться в HTML форматі, тому за допомогою команди NOPRINT було відключено автоматичний вивід результатів. В команду TABLES передаються змінні, за якими треба створити таблиці частот. За допомогою команди DROP з набору даних було виключено інформацію про відсотки, так як в цьому випадку вона нам не потрібна для аналізу. В результаті було отримано, що найчастішим небажаним стороннім ефектом в усіх групах пацієнтів є здуття живота в системі класу органів «розладів шлунково-кишкового тракту». Для тих хто приймав ліки його частота становить 8652, для тих хто приймав плацебо – 3693 та для всіх груп – 12345 відповідно. На малюнку 3.2.1 відображено частину отриманих результатів. Нижче наведено програмний код, за допомогою якого було розраховано частоти виникнення небажаних сторонніх ефектів:

```
proc freq data=adae;
```

```
    by TRT01A;
    tables AESOC*AETERM/
    out = ae_freq;
run;
```

	TRT01A	AESOC	AETERM	Frequency Count
1	ALL	BLOOD AND LYMPHATIC SYSTEM DISORDERS	LEUKOCYTOSIS	1985
2	ALL	BLOOD AND LYMPHATIC SYSTEM DISORDERS	NEUTROPENIA	1402
3	ALL	GASTROINTESTINAL DISORDERS	ABDOMINAL GAS	12345
4	ALL	GASTROINTESTINAL DISORDERS	ABDOMINAL PAIN	12069
5	ALL	GASTROINTESTINAL DISORDERS	CONSTIPATION	10923
6	ALL	GASTROINTESTINAL DISORDERS	INTERMITTENT INCREASED FLATULENCE	6652
7	ALL	GASTROINTESTINAL DISORDERS	INTERMITTENT TOOTHACHE, LEFT UPPER	5177
8	ALL	GASTROINTESTINAL DISORDERS	LEFT LOWER MOLAR TOOTHACHE	2530
9	ALL	GASTROINTESTINAL DISORDERS	NAUSEA	1695
10	ALL	GENERAL DISORDERS AND ADMINISTRATION SITE CONDITIONS	INCREASED ENERGY	8430

Мал. 3.2.1 Частоти появи небажаних сторонніх ефектів отримані процедурою FREQ

Наступним кроком було проведено аналіз тривалості небажаних сторонніх ефектів за допомогою процедури MEANS. Ця процедура «Обчислює дескриптивну статистику для змінних.» [16] (див. розділ MEANS Procedure). За допомогою команди DROP з набору даних було виключено змінні _type_ та _freq_, так як вони нам не потрібні для аналізу. В результаті було отримано такі дані: кількість головних болей, які тривали хоча б один день (N_), середню тривалість (MEAN_), середнє квадратичне відхилення (STD_), медіану (MEDIAN_), мінімальну (MIN_) та максимальну (MAX_) тривалість. Отримані результати потрібні для демонстрації та візуалізації даних, тому вони були вивантажені в зовнішній файл із розширенням .csv за допомогою вже знайомої нам процедури EXPORT. На малюнку 3.2.2 відображено результат дескриптивної статистики. Нижче наведено програмний код:

```
proc means data=ADAE(where=(not missing(AEDUR) and AESI = «Y»)) noprint;
```

```
    by TRT01A AETERM;
    var AEDUR;
    output out=disc_stat(drop=_type_ _freq_)
                n = N_
                mean= MEAN_
                std=STD_
                median=MEDIAN_
                min=MIN_
                max=MAX_ ;
run;

proc export data=disc_stat
    outfile='C:\Users\VYuskevych\UP\2023\Project\SAS\Output\Disc_stat.csv'
    dbms=csv
    replace;
run;
```

	TRT01A	AETERM	N_	MEAN_	STD_	MEDIAN_	MIN_	MAX_
1	ALL	HEADACHE	9295	6.8505648198	4.0697522029	7	1	18
2	ALL	INTERMITTENT HEADACHE	7272	6.8223322332	4.0665697899	6	1	18
3	DRUG	HEADACHE	6546	7.2083715246	4.2756757112	7	1	18
4	DRUG	INTERMITTENT HEADACHE	5129	7.1762526808	4.2659873562	7	1	18
5	PLACEBO	HEADACHE	2749	5.9985449254	3.382999398	6	1	13
6	PLACEBO	INTERMITTENT HEADACHE	2143	5.9752683154	3.3983635061	6	1	13

Малюнок 3.2.2 Результат дескриптивної статистики отриманий процедурою MEANS

Останнім кроком був прорахований час від дня першого прийому препарату до появи першого головного болю та отримані дані для будування графіку аналізу виживаності методом Каплан Майер, який дозволяє оцінити ймовірність настання події, в нашому випадку – це небажаний сторонній ефект особливого інтересу, а саме головний біль. Для цього використовувався набір даних ADTTE, завантажений на першому кроці другого етапу. Для пацієнтів, які приймали плацебо було отримано результати, що у 25% відсотків досліджуваних головний біль виникав до 27 дня від початку прийому препарату, більш детально на малюнку 3.2.3. Також було оцінено 95% довірчі інтервали для 25% вони складають (25;28) днів.

Summary Statistics for Time Variable AVAL

Quartile Estimates				
Percent	Point Estimate	95% Confidence Interval		
		Transform	[Lower	Upper)
75	97.000	LOGLOG	95.000	112.000
50	56.000	LOGLOG	53.000	58.000
25	27.000	LOGLOG	25.000	28.000

Мал.3.2.3 Результати аналізу виживаності для пацієнтів які приймали плацебо

Для пацієнтів, які приймали ліки було розраховано, що у 25% відсотків досліджуваних головний біль виникав до 28 дня від початку прийому препарату, більш детально на малюнку 3.2.4. Також було оцінено 95% довірчі інтервали для 25% вони складають також складають (27;29) днів.

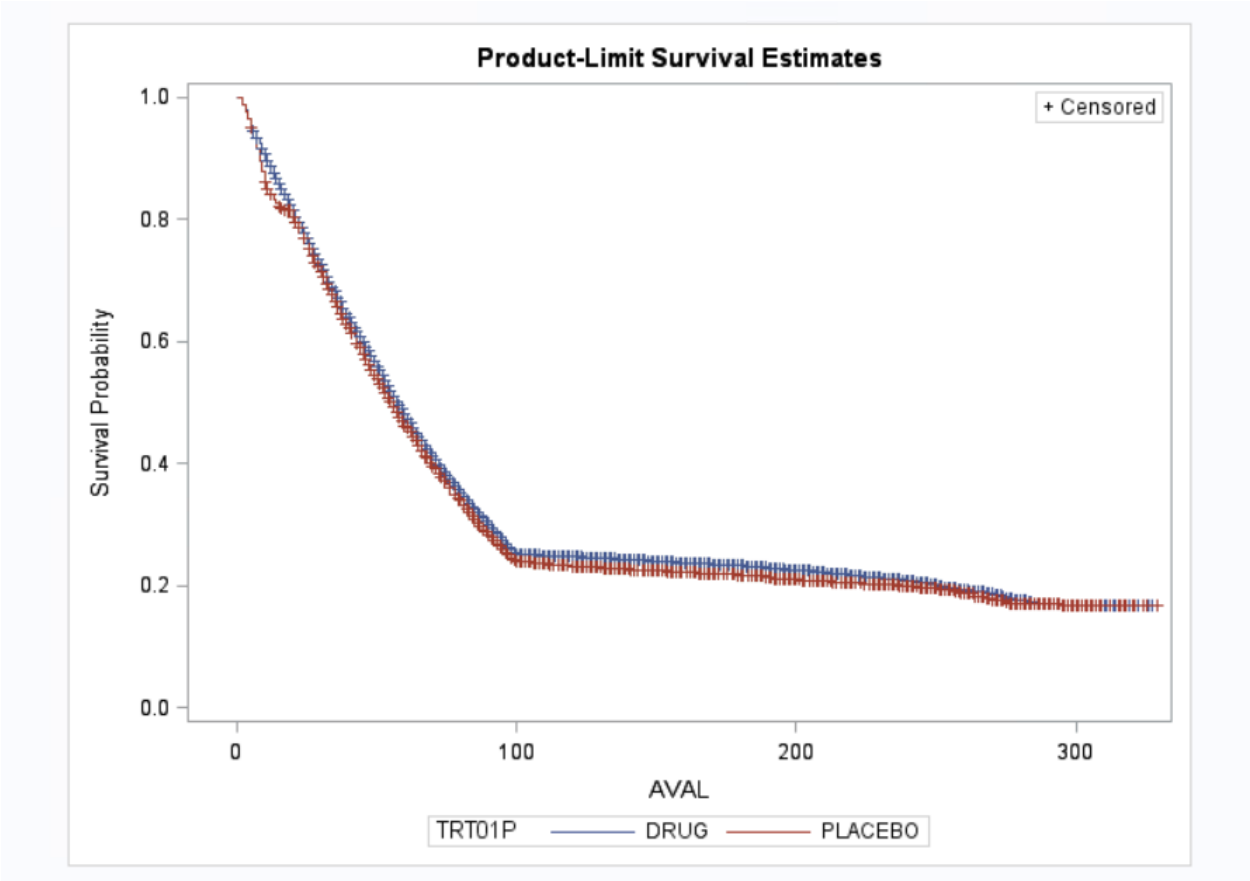
Summary Statistics for Time Variable AVAL

Quartile Estimates				
Percent	Point Estimate	95% Confidence Interval		
		Transform	[Lower	Upper)
75	108.000	LOGLOG	98.000	149.000
50	58.000	LOGLOG	56.000	59.000
25	28.000	LOGLOG	27.000	29.000

Мал.3.2.4 Результати аналізу виживаності для пацієнтів які приймали ліки

Для розрахунку було використано процедуру LIFETEST, яка частіше за все використовується для «непараметричних оцінок функції виживаності для вибірки часу виживання» [16] (див. розділ LIFETEST Procedure). Було використано підхід запропонований в [18], де method = km визначає, що використовується метод Каплан Майера для оцінки виживаності, outsurv = dsplot створює вихідний набір даних dsplot, який містить оцінки виживаності для кожного часового інтервалу. atrisk і plots = survival вказують на створення графіків виживаності, а також кількості осіб, що перебувають у ризику. Фінальним результатом є графік (див. малюнок 3.2.5), який нам знадобиться для наступного етапу дослідження. Нижче наведено програмний код:

```
proc lifetest data=adtte
      method=km outsurv=dsplot atrisk plots=survival;
      time AVAL*CNSR(1);
      strata TRT01P;
run;
```



Мал.3.2.5 Графік виживаності Каплан Майера, сгенерований процедурою LIFETEST

R

Як і у випадку з SAS, перед початком аналізу треба було завантажити тестові дані із зовнішнього файлів з розширенням .csv, ADAE та ADTTE, та провести попередню обробку цих даних для підготовки їх до аналізу. Як і на минулому етапі для завантаження зовнішніх даних було використано функцію READ.CSV. Далі було відібрано з набору даних ADAE було відібрано тільки тих пацієнтів, які прийняли хоча б одну дозу препарату, за тією ж умовою, що і у випадку з SAS. Наступним кроком було об’єднано два однакових набори даних ADAE для отримання додаткових записів для кожного суб’єкта незалежно від препарату, за допомогою функції RBIND. Ця функція «об'єднує два або більше наборів даних за рядками» [17] (див. розділ rbind: Union two or more SparkDataFrames). Нижче наведено програмний код:

```
adtte <- read.csv('C:/Users/VYuskevych/UP/2023/Project/Input/csv/adtte.csv')

adae_a <- read.csv('C:/Users/VYuskevych/UP/2023/Project/R/Output/adae.csv')
adae_a <- adae_a[adae_a$SAFFL == «Y», ]

adae_b <- adae_a
adae_b$TRT01A <- «ALL»
adae <- rbind(adae_a, adae_b)
```

Далі було виконано аналіз частот небажаних сторонніх ефектів. Для початку цього аналізу треба додатково завантажити пакет DPLYR за допомогою функції INSTAL.PACKAGES та підключити цей пакет до бібліотеки використовуючи функцію LIBRARY, для цього було використано підхід запропонований в [17] (див. розділ DPLYR). Після підготовки пакету було виконано фільтрацію даних в ADAE, відфільтровуючи всі рядки, де змінна TRT01A має пропущені значення (NA) за допомогою функції FILTER [17] (див. розділ filter: Return rows with matching conditions), згруповано дані за змінною TRT01A завдяки функції GROUP_BY [17] (див. розділ group_by:

Group by one or more variables) та підраховано кількість спостережень для кожного AETERM у кожній групі функцією COUNT [17] (див. розділ count: Count the number of occurrences). Результат повністю співпадає з результатом SAS. Частина результату відображена на малюнку 3.2.6. Нижче наведено програмний код:

```
instal.packages(dplyr)
library(dplyr)

ae_freq <- adae %>%
filter(!is.na(TRT01A)) %>%
group_by(TRT01A) %>%
count(AESOC, AETERM, name = «COUNT») %>%
ungroup()
```

```
# A tibble: 117 × 4
   TRT01A AESOC                                        AETERM                              COUNT
   <chr>  <chr>                                        <chr>                               <int>
 1 ALL    BLOOD AND LYMPHATIC SYSTEM DISORDERS         LEUKOCYTOSIS                         1985
 2 ALL    BLOOD AND LYMPHATIC SYSTEM DISORDERS         NEUTROPENIA                          1402
 3 ALL    GASTROINTESTINAL DISORDERS                   ABDOMINAL GAS                       12345
 4 ALL    GASTROINTESTINAL DISORDERS                   ABDOMINAL PAIN                      12069
 5 ALL    GASTROINTESTINAL DISORDERS                   CONSTIPATION                        10923
 6 ALL    GASTROINTESTINAL DISORDERS                   INTERMITTENT INCREASED FLATULENCE    6652
 7 ALL    GASTROINTESTINAL DISORDERS                   INTERMITTENT TOOTHACHE, LEFT UPPER   5177
 8 ALL    GASTROINTESTINAL DISORDERS                   LEFT LOWER MOLAR TOOTHACHE           2530
 9 ALL    GASTROINTESTINAL DISORDERS                   NAUSEA                               1695
10 ALL    GENERAL DISORDERS AND ADMINISTRATION SITE CONDITIONS INCREASED ENERGY              8430
```

Мал.3.2.6 Частоти появи небажаних сторонніх ефектів отримані функцією COUNT

Наступним кроком було розрахування дескриптивної статистики для аналізу тривалості небажаних сторонніх ефектів особливого інтересу за допомогою функції AGGREGATE, яка «Розбиває дані на підмножини, обчислює статистику для кожної з них і повертає результат у зручній формі» [17] (див. розділ aggregate: Compute Summary Statistics of Data Subsets). В результаті було створено набір даних disc_stat, в якому зберігається інформація щодо кількості головних болей, які тривали хоча б один день (AEDUR.N), середню тривалість (AEDUR.MEAN), середнє квадратичне відхилення (AEDUR.STD), медіану (AEDUR.MEDIAN), мінімальну (AEDUR.MIN) та максимальну (AEDUR.MAX) тривалість. Було використано модернізований підхід, запропонований в [17] (див. розділ aggregate: Compute

Summary Statistics of Data Subsets). Отримані результати також співпадають з результатами SAS та відображені на малюнку 3.2.7. Нижче наведено програмний код:

```
  disc_stat <- adae[!is.na(adae$AEDUR) & adae$AESI == «Y», ]
  disc_stat <- aggregate(AEDUR ~ TRT01A + AETERM, data = disc_stat, FUN =
function(x) c(N = length(x), MEAN = mean(x), STD = sd(x), MEDIAN = median(x),
MIN = min(x), MAX = max(x)))
```

	TRT01A	AETERM	AEDUR.N	AEDUR.MEAN	AEDUR.STD	AEDUR.MEDIAN	AEDUR.MIN	AEDUR.MAX
1	ALL	HEADACHE	9295.000000	6.850565	4.069752	7.000000	1.000000	18.000000
2	DRUG	HEADACHE	6546.000000	7.208372	4.275676	7.000000	1.000000	18.000000
3	PLACEBO	HEADACHE	2749.000000	5.998545	3.382999	6.000000	1.000000	13.000000
4	ALL	INTERMITTENT HEADACHE	7272.000000	6.822332	4.066570	6.000000	1.000000	18.000000
5	DRUG	INTERMITTENT HEADACHE	5129.000000	7.176253	4.265987	7.000000	1.000000	18.000000
6	PLACEBO	INTERMITTENT HEADACHE	2143.000000	5.975268	3.398364	6.000000	1.000000	13.000000

Мал.3.2.7 Результат дескриптивної статистики отриманий процедурою MEANS

Останній крок це розрахування часу від дня першого прийому препарату до появи першого головного болю та отримання дані для будування графіку аналізу виживаності методом Каплан Майера. Для цього, як і у випадку з SAS використовувався набір даних ADTTE. Для максимальної консистентності було обрано комбінацію підходу запропонованому в [19] та підходу [20].

Спочатку треба завантажити пакети та підключити бібліотеки ggplot2 та visR, аналогічно тому, як це було зроблено на першому кроці другого етапу. Після цього створюється рядок з назвою набору даних за допомогою функції paste0(), далі зберігаються оригінальні опції R за допомогою options(), потім встановлюються глобальні налаштування форматування змінних (кількість десяткових знаків після коми) та графічного оформлення графіків, далі встановлюються глобальні налаштування для відображення таблиць, включаючи кількість рядків на сторінці та інші параметри, наприкінці підготовки даних та форматів відновлюються оригінальні опції R, які були збережені раніше.

За допомогою функції visR::estimate_KM(survfit_object) було обчислено аналіз виживаності методом Каплан Майєра. Далі функція

visR::get_quantile(survfit_object) обчислює квантилі з об'єкта survfit_object, який містить результати аналізу виживання, а функція visR::get_risktable(survfit_object) створює ризикову таблицю. Потім було створено графік виживаності з довірчими інтервалами та ризиковою таблицею. Оператор %>% використовується для подання результату попереднього виразу як аргументу наступного виразу. Таким чином, об'єкт survfit_object, що містить результати обчислення функції виживання, передається функції visr() для створення графіка виживання, після чого за допомогою функцій add_CI() та add_CNSR() до графіка додаються довірчі інтервали та цензуровані значення. Результати обчислення квантилів на мові програмування R повністю співпадають з результатами отриманими за допомогою SAS та наведені на малюнку 3.2.8.

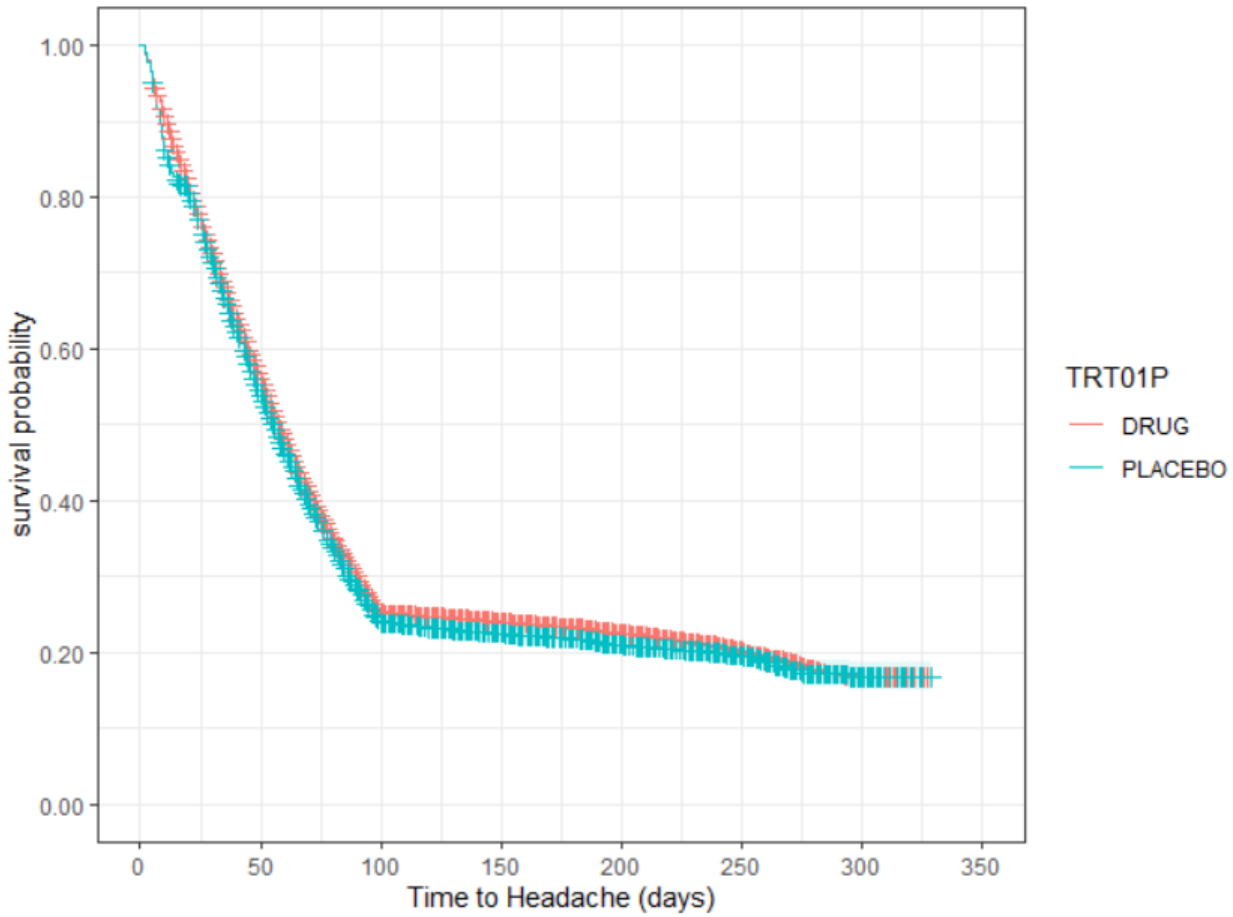
```
>     visR::get_quantile(survfit_object)
          strata quantity 25 50  75
3    TRT01P=DRUG    lower 27 56  98
1    TRT01P=DRUG quantile 28 58 108
5    TRT01P=DRUG    upper 29 59 149
4 TRT01P=PLACEBO    lower 25 53  95
2 TRT01P=PLACEBO quantile 27 56  97
6 TRT01P=PLACEBO    upper 28 58 112
```

Мал.3.2.8 Результати аналізу виживаності отримані за допомогою R

Після виконання цієї програми ми отримуємо графік (див. малюнок 3.2.9), який буде використовуватись на наступному етапі. Нижче наведено програмний код аналізу виживаності методом Каплан Майера:

```
# Packages
library(ggplot2)
library(visR)

  # Save original options()
  old <- options()

  # Global formatting options
  options(digits = 3)

  # Global ggplot settings
```

```
  theme_set(theme_bw())

  # Global table settings
  options(DT.options = list(pageLength = 10,
                            language = list(search = 'Filter:'),
                            scrollX = TRUE))

  # Restore original options()
  options(old)

  # Estimate a survival object
  survfit_object <-  adtte %>%
    visR::estimate_KM(data = ., strata = «TRT01P»)
    visR::get_quantile(survfit_object)
    visR::get_risktable(survfit_object)

  # Create and display a Kaplan-Meier from the survival object and add a
risktable
  visr(survfit_object) %>%
    visR::add_CI() %>%
    visR::add_CNSR()
```



Мал.3.2.9 Графік виживаності Каплан Майера, сгенерований пакетом visR

3.3 Порівняння особливостей візуалізації даних у SAS і R в контексті клінічних досліджень.

Фінальним етапом обробки даних клінічних досліджень, зазвичай, є створення TLF (Table, Listing, Figures), тобто формування звіту шляхом приведення результатів цих досліджень до певного вигляду та відправка їх до регуляторних органів країни. Формат та тип звіту залежить від замовника, але частіше за все це rtf для таблиць та лістингів, і pdf для фігур. Саме тому було вирішено порівняти можливості візуалізації даних, отриманих на минулому етапі, а саме створення таблиці з дескриптивною статистикою тривалості головного болю, з виводом в rtf файл, а також вивід графіку виживаності методом Каплан Майера в pdf файл.

SAS

Для створення звіту частіше за все використовується процедура REPORT, яка «поєднує можливості процедур PRINT, MEANS і TABULATE з можливостями кроку даних в одному інструменті для написання звітів» [16] (див. розділ REPORT Procedure). Ця процедура є дуже гнучкою та має майже безліч додаткових опцій та можливостей, тому було вирішено скористатись вже готовим підходом, запропонованим в [21], який був модифікований для нашого випадку, наприклад зміна тайтлів та футноутів, назва змінних та їх кількість, та інше. Так як SAS, при виводі звіту в rtf, автоматично записує тайтл таблиці в хедер документу, використовується команда bodytitle, яка дозволяє винести його за межі хедеру. Але особливістю цієї команди є створення футноуту на останній сторінці звіту [21], незалежно від того чи був він заданий в коді, тому, щоб подавити цю особливість було задано футноут із порожнім значенням за допомогою команди footnote. Результат процедури репорт відображено на малюнку 3.3.1. Нижче наведено програмний код:

```
ods rtf file = «C:\Users\VYuskevych\UP\2023\Project\SAS\Output\t_aedur.rtf»
bodytitle;
options nodate nonumber;

proc report data=final nowd
```

```
      style(report)={just=center outputwidth=7 in}
      style(lines)=header{background=white asis=on font_size=12pt
font_face=«TimesRoman»
            font_weight=bold just=left}
      style(header)=header{background=white font_size=12pt
font_face=«TimesRoman» frame=box
            font_weight=bold}
      style(column)=header{background=white font_size=10pt
font_face=«TimesRoman»
            font_weight=medium};

      columns ORD1 ORD2 AETERM STAT ALL DRUG PLACEBO;
      define ORD1 / order noprint;
      define ORD2 / order noprint;
      define AETERM / «Prefered Term» width=15 style(column)={just=left}
order order=data;
      define STAT / «Statistics» width=11 style(column)={just=center};
      define ALL / «All Subjects» width=11 style(column)={just=center};
      define DRUG / «Drug» width=11 style(column)={just=center};
      define PLACEBO / «Placebo» width=11 style(column)={just=center};

      title1 color=black font=«Times New Roman» height=14pt bold «Duration of
Adverse Events of Special Interest» ;
      footnote1 « «;

run;

ods rtf close;
```

Duration of Adverse Events of Special Interest

Prefered Term	Statistics	All Subjects	Drug	Placebo
HEADACHE	N	9295	6546	2749
HEADACHE	mean (std)	6.85 (4.07)	7.21 (4.276)	6 (3.383)
HEADACHE	median	7	7	6
HEADACHE	min, max	1, 18	1, 18	1, 13
INTERMITTENT HEADACHE	N	7272	5129	2143
INTERMITTENT HEADACHE	mean (std)	6.82 (4.067)	7.18 (4.266)	5.98 (3.398)
INTERMITTENT HEADACHE	median	6	7	6
INTERMITTENT HEADACHE	min, max	1, 18	1, 18	1, 13

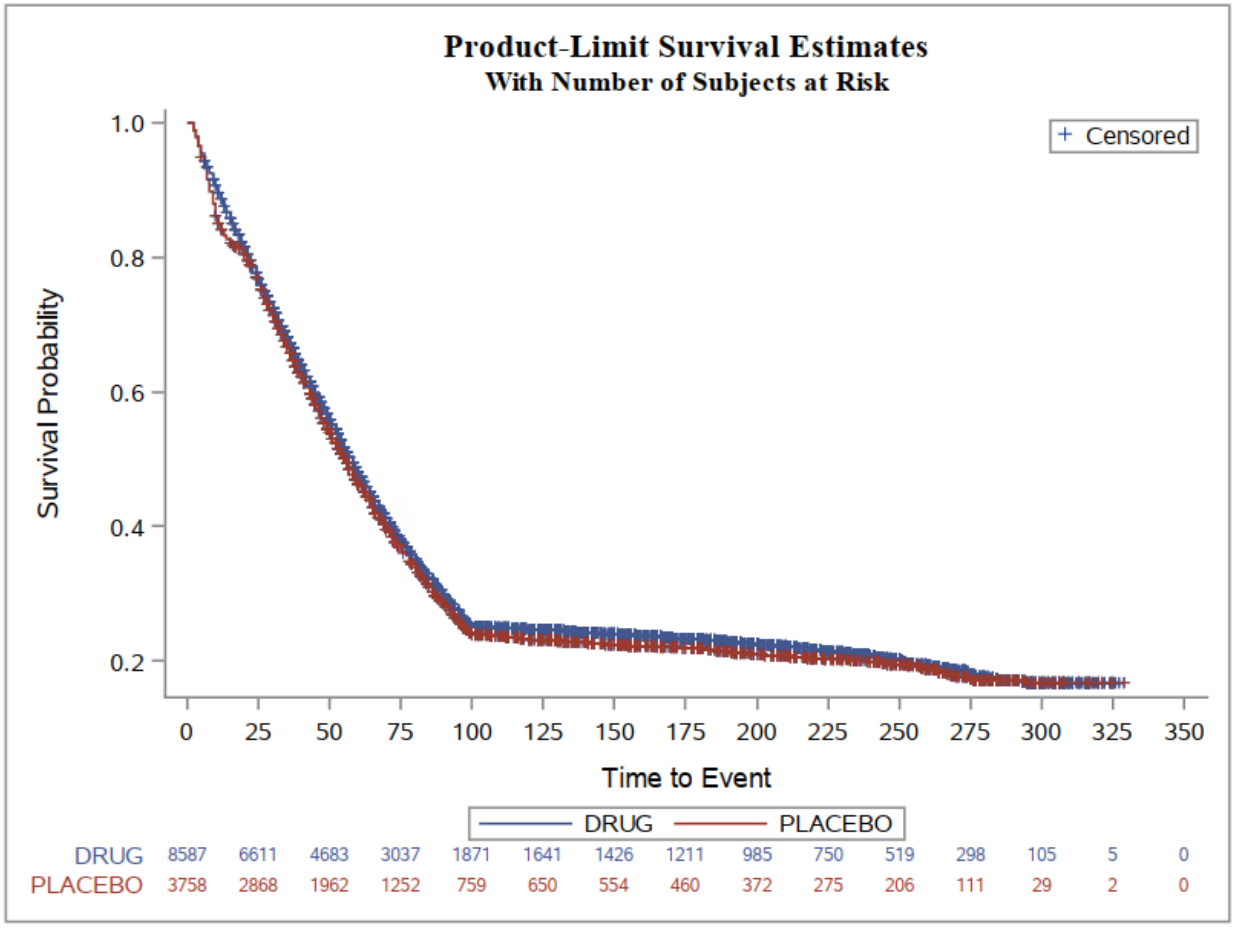
Мал.3.3.1 Звіт тривалості головної болі, створений за допомогою SAS Proc Report

Останнім кроком є вивід графіку виживаності Каплан Майера в pdf файл для демонстрації можливостей візуалізації графіків. Процедура LIFETEST за замовченням генерує вивід графіка в HTML з базовим зовнішнім виглядом, який існує тільки протягом сесії, тому для виводу цього графіку в будь-який

зовнішній файл, або зміни його візуалу, треба використовувати додаткові процедури та команди. Однією з таких процедур є SGPLOT, яка «створює одну або декілька діаграм і накладає їх на один набір вісей» [16] (див. розділ SGPLOT Procedure). Для виводу графіка в pdf файл використовується команда ODS PDF, яка «відкриває, керує або закриває напрямок PDF, який виробляє вихідні дані у форматі PDF» [16] (див. розділ ODS PDF Statement).

Першим кроком, за допомогою вже відомої нам процедури LIFETEST, було створено графік виживаності методом Каплан Майера, в який ще додано таблицю кількості осіб, що перебувають у ризику. Результат нової процедури було збережено в набір даних Dsplot. Далі цей набір даних було використано в процедурі SGPLOT методом, запропонованим в [23], який був адаптований під поточні дані. Результат роботи наведено на малюнку 3.3.2. Нижче наведено програмний код:

```
ods trace on / label;
ods output SurvivalPlot = Dsplot;
proc lifetest data=adtte
      method=km atrisk plots=survival(atrisk(maxlen=13 outside)=0 to 350 by
25);
      time AVAL*CNSR(1) ;
      strata TRT01P;
run;

ods pdf file="C:\Users\VYuskevych\UP\2023\Project\SAS\Output\KM_plot.pdf";

title font="Times New Roman" height=12pt 'Product-Limit Survival Estimates';
title2 font="Times New Roman" height=10pt 'With Number of Subjects at Risk';

proc sgplot data=Dsplot noborder;
      xaxis values=(0 to 350 by 25) label="Time to Event";
    step x=time y=survival / group=stratum name='s';
    scatter x=time y=censored / markerattrs=(symbol=plus) name='c';
    scatter x=time y=censored / markerattrs=(symbol=plus) GROUP=stratum;
    xaxistable atrisk / x=tatrisk class=stratum colorgroup=stratum ;
    keylegend 'c' / location=inside position=topright;
    keylegend 's';
run;

ods pdf close;
```



Мал.3.3.2 Графік виживаності методом Каплан Майера, згенерований SAS процедурою SGPLOT

R

На відміну від SAS, в R існує величезна кількість різноманітних пакетів для генерації звітів, наприклад R Markdown, officer, r2rtf та інші, більш детально в [17]. Проаналізувавши можливості існуючих пакетів, їх особливості та синтаксис, було прийнято рішення використовувати саме пакет r2rtf, який, на мою думку, найбільше підходить для поточної задачі. Так само як і на етапі з SAS, було використано поєднання із вже існуючих рішень, запропонованих в [22], які були модернізовані під потреби тестових даних: наприклад розміри шрифтів для тайтла (rtf_title()), хедера (rtf_colheader()), та тексту самої таблиці (rtf_body()). Також було додано функцію rtf_page(), за допомогою якої було встановлено одинарні верхні та нижні межі таблиці, так як за замовченням вони двійні, а також потрібний нам розмір полів. Нижче наведено програмний код:

```
disc_stat %>%
 rtf_page(border_first = «single»,
          border_last = «single»,
          margin = c(1, 1, 1, 1 ,1, 1)) %>%
 rtf_title(title = «Duration of Adverse Events by System Organ Class»,
           text_font_size = 14,
           text_format = «b») %>%
 rtf_colheader(colheader = «Prefered Term | Statistics | All Subjects | Drug
| Placebo»,
               col_rel_width = c(4, 2, 2, 2, 2),
               text_font_size = 12,
               text_format = «b») %>%
 rtf_body(col_rel_width = c(4, 2, 2, 2, 2),
          text_justification = c(«l», rep(«c»,4)),
          text_font_size = 10) %>%
 rtf_encode() %>%
 write_rtf(«C:/Users/VYuskevych/UP/2023/Project/R/Output/t_aedur.rtf»)
```

Як можна помітити, обсяг самого програмного коду набагато менше в порівнянні з SAS, та інтуїтивно простіше та зрозуміліше. Результат генерування звіту наведено на малюнку 3.3.3.

Duration of Adverse Events by System Organ Class

Prefered Term	Statistics	All Subjects	Drug	Placebo
HEADACHE	N	9295	6546	2749
HEADACHE	mean (std)	6.85 (4.07)	7.21 (4.276)	6 (3.383)
HEADACHE	median	7	7	6
HEADACHE	min, max	1, 18	1, 18	1, 13
INTERMITTENT HEADACHE	N	7272	5129	2143
INTERMITTENT HEADACHE	mean (std)	6.82 (4.067)	7.18 (4.266)	5.98 (3.398)
INTERMITTENT HEADACHE	median	6	7	6
INTERMITTENT HEADACHE	min, max	1, 18	1, 18	1, 13

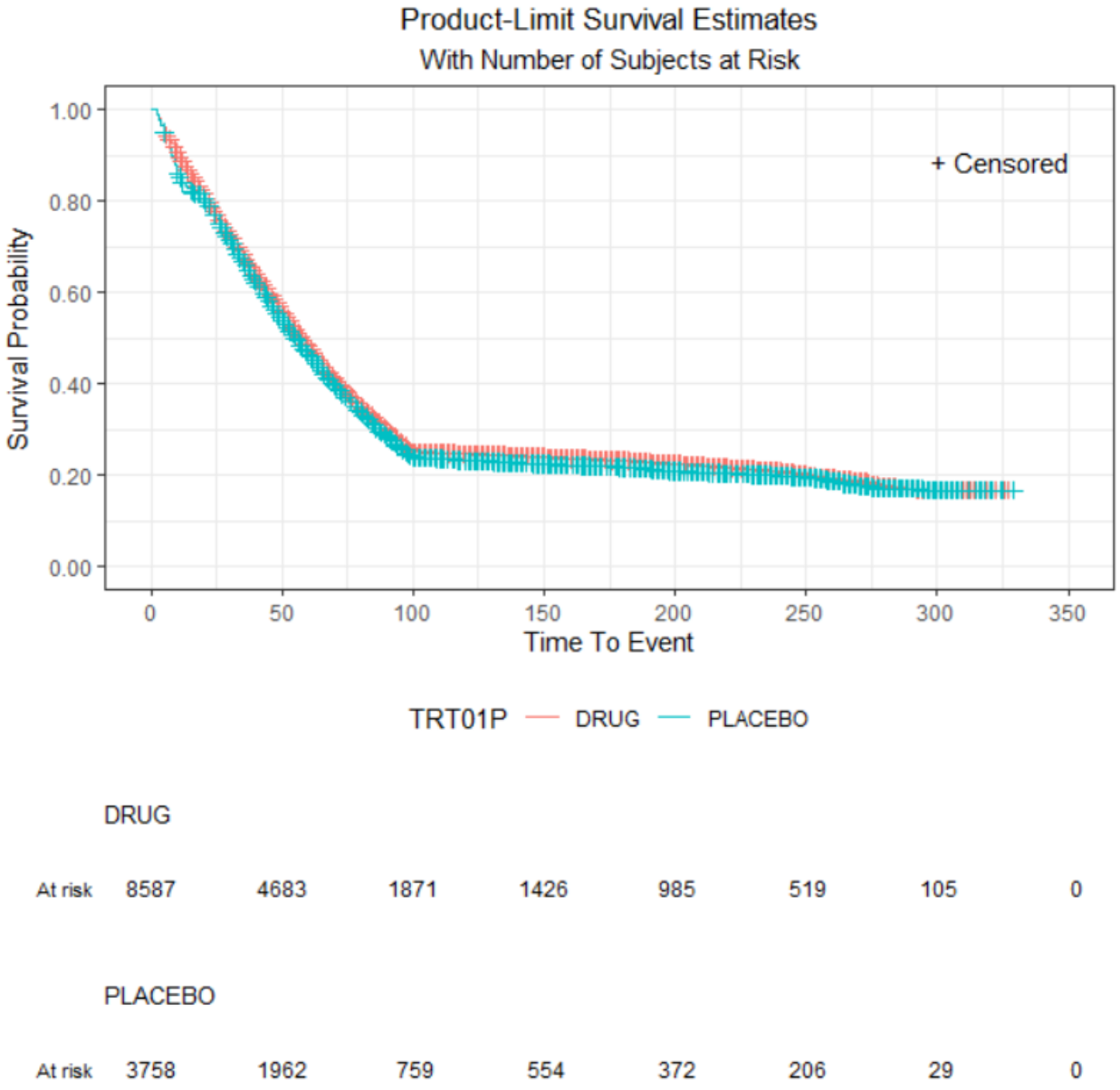
Мал.3.3.3 Звіт тривалості головної болі, створений за допомогою R пакету r2rtf

Останнім кроком є вивід графіку виживаності методом Каплан Майера в pdf файл. Для цього було використано модифікований код з минулого етапу дослідження, а саме: було додано заголовок та підзаголовок (функція ggtitle(title = , subtitle =)), за допомогою функції visr() було додано назви осей та переміщено легенду з правого боку графіка в нижню частину,

використанням функції add_annotation() додано анотацію для цензурованих суб'єктів. На відміну від SAS, не виникає потреби попередньо зберігати результати в додатковий набір даних, що значно зменшує час виконання. На малюнку 3.3.4 зображено результат створення графіка виживаності. Готовий графік було збережено в файл під назвою KM_plot.pdf за допомогою функції ggsave(), яка «є зручним інструментом для збереження графіка» [16] (див. розділ ggsave: Save a ggplot with sensible defaults). Нижче наведено програмний код:

```
# Figure to pdf
library(ggplot2)
library(visR)

 # Save original options()
  old <- options()

  # Global formatting options
  options(digits = 3)

  # Global ggplot settings
  theme_set(theme_bw())

  # Global table settings
  options(DT.options = list(pageLength = 14,
                            language = list(search = 'Filter:'),
                            scrollX = TRUE))

  # Restore original options()
  options(old)

# Estimate a survival object
survfit_object <-  adtte %>%
  visR::estimate_KM(data = ., strata = "TRT01P")

# Create and display a Kaplan-Meier from the survival object and add a
risktable
plot <- (visr(survfit_object,
              legend_position = "bottom",
              x_label = "Time To Event",
              y_label = "Survival Probability") +
              ggtitle("Product-Limit Survival Estimates", subtitle = "With
Number of Subjects at Risk") +
              theme(plot.title = element_text(hjust = 0.5),
                    plot.subtitle = element_text(hjust = 0.5))) %>%
  visR::add_CI() %>%
  visR::add_CNSR() %>%
  visR::add_risktable() %>%
  visR::add_annotation(label = "+ Censored",
                       xmin = 0.8,
                       xmax = 1.0,
                       ymin = 0.8,
                       ymax = 0.9)
```

```
# Save the plot as a PDF file
ggsave("C:/Users/VYuskevych/UP/2023/Project/R/Output/KM_plot.pdf", plot,
scale = 1)
```



Мал. 3.3.4 Графік виживаності методом Каплан Майера, згенерований R функцією visr

4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

В ході аналізу результатів дослідження, для зручності, було створено порівняльну таблицю за сімома критеріями, зазначеними у вступній частині. Порівняльна таблиця наведена нижче:

Критерій	SAS	R
Доступність	Може бути обмежений через високі вимоги до ліцензування та вартість	Легко доступний для завантаження та використання через відкриті джерела та безкоштовний характер
Вартість	Платний; вимагає придбання ліцензій	Безкоштовний; ліцензія не потрібна
Регуляторна підтримка	Широко використовується фармацевтичними компаніями та медичними установами для відповідності регуляторним вимогам	Зазвичай використовується в академічних та дослідницьких галузях, потенційно вимагає додаткової валідації для регуляторної підтримки
Функціонал інструментів	Широкий набір вбудованих процедур, функцій та команд	Має вбудовані функції та пакети, а також велику кількість додаткових пакетів та розширень

Критерій	SAS	R
Гнучкість	Стабільний та зручний в роботі з великими обсягами даних, але не має гнучкості; вимагає глибоких знань синтаксису; зменшення обсягу коду значно впливає на читабельність	Має гнучкість та можливість розробки власних алгоритмів та функцій, через що не вимагає глибокого знання синтаксису; код компактний та зрозумілий інтуїтивно
Можливості статистичного аналізу	Має розширений набір статистичних процедур та методів, спеціально призначених для аналізу клінічних даних	Вбудованих пакетів та функцій недостатньо для проведення навіть базового статистичного аналізу, але має багато додаткових пакетів та функцій для різноманітних статистичних аналізів, таких як аналіз виживання, лінійна та логістична регресія, мета-аналіз тощо;
Візуалізація	Спеціалізовані інструменти для візуалізації даних, але є обмеження вбудованими процедурами; вимагає окремих знань та певний час на навчання	Широкий вибір пакетів для створення різноманітних графіків та візуалізації даних, які вимагають додаткових знань, але потребують значно менше часу на навчання на відміну від SAS

Таблиця 4.1 Аналіз результатів дослідження

5 ВИСНОВОК

В ході дослідження було проведено глибоке порівняння мов програмування SAS та R в сфері клінічних випробувань. SAS є стабільним та надійним інструментом для обробки, аналізу, та візуалізації даних клінічних випробувань, який спокійно справляється з великим обсягом даних. Результат процедур статистичного аналізу не потребує додаткової валідації та приймається регуляторними органами по всьому світу. Але за цю надійність треба платити, бо SAS є комерційним програмним забезпеченням, ціни на ліцензію якого залежать від багатьох факторів і обговорюються індивідуально з кожним замовником. SAS потребує глибоких знань синтаксису, професійної підготовки та сертифікації користувачів. R є безкоштовним та легко доступним програмним забезпеченням, яке також гарно справляється з обробкою великих наборів даних, але для статистичного аналізу потребує завантаження додаткових пакетів, які на відміну від SAS не є повністю ратифікованими на момент дослідження, та потребують додаткової валідації. Синтаксис R є зрозумілим на інтуїтивному рівні, тому базове розуміння логіки кодування та гнучкість значно полегшують навчальний процес, та зменшують час витрачений на нього.

Проаналізувавши результати дослідження можна прийти до висновку, що R переважує за п’ятьма з семи критеріїв: доступність, вартість, функціонал інструментів, гнучкість, візуалізація. Однак поки він не отримає значної регуляторної підтримки, SAS буде залишатись монополістом у сфері клінічних випробувань. Але світ не стоїть на місці і постійно розвивається і з кожним днем, все більше клінічних досліджень проводять на R, тому у найближчому майбутньому R може і не буде популярнішим за SAS, але точно буде на рівні.

6 СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Smith, J. et al. (2020). «Comparative Analysis of SAS and R for Clinical Trial Data Analysis.» *Journal of Clinical Research*, 15(2), 45-58.
2. Johnson, A. (2021). «Efficiency Analysis of R Programming Language in Clinical Research.» Clinical Statistics Review, 25(4), 112-125.
3. Clark, B. et al. (2019). «Enhancing Speed and Efficiency of R for Clinical Trial Data Analysis.» Journal of Biomedical Informatics, 18(3), 245-260.
4. Miller, E. et al. (2021). «Exploring the Role of SAS in Handling Big Data in Clinical Trials.» Health Data Analytics Journal, 7(1), 78-93.
5. Lee, K. et al. (2019). «Statistical Analysis with SAS in Clinical Trials: A Comprehensive Review.» Clinical Trials Journal, 10(3), 78-92.
6. Rao, S. et al. (2021). «Flexibility of R in Implementing Advanced Statistical Methods in Clinical Research.» Statistical Methodology Review, 28(2), 210-225.
7. Garcia, M. et al. (2020). «Advantages of Using R for Implementation of Novel Statistical Methods in Clinical Research.» Journal of Biostatistics, 14(4), 315-330.
8. Goodnight, J. H. (2008). «SAS System for Statistical Analysis.» John Wiley & Sons.
9. Machanic, L. J., Smith, K. A., Smith, K. A., & Machanic, L. J. (2008). «Clinical Data Management.» CRC Press.
10. Ihaka, R., & Gentleman, R. (1996). «R: A Language for Data Analysis and Graphics.» Journal of Computational and Graphical Statistics, 5(3), 299-314.
11. Becker, R. A., Chambers, J. M., & Wilks, A. R. (1988). «The New S Language: A Programming Environment for Data Analysis and Graphics.» Wadsworth & Brooks/Cole.
12. Kuhn, M., & Johnson, K. (2013). «Applied Predictive Modeling.» Springer.

13. CDISC Submission Data Standards Team. Study Data Tabulation Model Implementation Guide: Human Clinical Trials Version 3.3 (Final). Clinical Data Interchange Standards Consortium, Inc., 2018. 426 p.
14. CDISC Analysis Data Model Team. Analysis Data Model Implementation Guide Version 1.3 (Final). Clinical Data Interchange Standards Consortium, Inc., 2021. 88 p.
15. CDISC Analysis Data Model (ADaM) Team. The ADaM Basic Data Structure for Time-to-Event Analyses Version 1.0. Clinical Data Interchange Standards Consortium, Inc., May 8, 2012. 31 p.
16. SAS Help Center. *SAS Help Center*. URL: https://documentation.sas.com/doc/ru/pgmsascdc/9.4_3.5/pgmsaswlcm/home.htm (date of access: 26.03.2024).
17. Home - RDocumentation. *Home - RDocumentation.* URL: https://www.rdocumentation.org/ (date of access: 26.03.2024).
18. Kuhfeld W. F., So Y., SAS Institute Inc. Creating and customizing the kaplan-meier survival plot in proc lifetest in the SAS/STAT® 13.1 release. P. 7. URL: https://www.lexjansen.com/pharmasug/2014/SP/PharmaSUG-2014-SP14-SAS.pdf (date of access: 15.04.2024).
19. Survival Analysis with visR using CDISC ADaM Time-To-Event Analysis Dataset (ADTTE). The Comprehensive R Archive Network. URL: https://cran.r-project.org/web/packages/visR/vignettes/CDISC_ADaM.html (date of access: 15.04.2024).
20. R: wrapper around quantile methods. search.r-project.org. URL: https://search.r-project.org/CRAN/refmans/visR/html/get_quantile.html (date of access: 15.04.2024).
21. Creating Word Tables using PROC REPORT and ODS RTF. URL: https://www.lexjansen.com/pharmasug/2004/TechnicalTechniques/TT02.pdf (date of access: 17.04.2024).
22. R2rtf – an R package to produce rich text format (RTF) tables and figures. PharmaSUG. URL:

https://www.pharmasug.org/proceedings/2020/DV/PharmaSUG-2020-DV-198.pdf (date of access: 19.04.2024).

23. Survival plot with a twist using SGPLOT procedure. *Graphically Speaking*. URL: https://blogs.sas.com/content/graphicallyspeaking/2018/02/19/survival-plot-twist-using-sgplot-procedure/ (date of access: 22.04.2024).
24. Search for FDA Guidance Documents. U.S. Food and Drug Administration. URL: https://www.fda.gov/regulatory-information/search-fda-guidance-documents (date of access: 26.04.2024).
25. Providing Regulatory Submissions In Electronic Format – Standardized Study Data. Official edition. 2021. 15 p. URL: https://www.fda.gov/media/82716/download (date of access: 26.04.2024).
26. Shiny App Successfully Reviewed by FDA CDER Staff (Pilot 2 Announcement 2) - R Consortium. R Consortium. URL: https://www.r-consortium.org/announcement/2023/10/05/shiny-app-successfully-reviewed-by-fda-cder-staff-pilot-2-announcement-2 (date of access: 26.04.2024).
27. Statistical review and evaluation. URL: https://github.com/RConsortium/submissions-wg/blob/0f1dc5c30985d413f75d196c2b6caa96231b26ee/_Documents/Summary_R_Pilot2_Submission%2027SEP2023.pdf (date of access: 26.04.2024).

ВІДГУК НАУКОВОГО КЕРІВНИКА