

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

 д.т.н., проф. Віталій МОКІН

2024 року

Бакалаврська дипломна робота на тему:

**«СИСТЕМНИЙ АНАЛІЗ ГЕОПРОСТОРОВИХ ЗВ'ЯЗКІВ МАСИВІВ ВОД БА-
СЕЙНУ РІЧКИ ПІВДЕННИЙ БУГ»**

Виконав: студент 4 курсу, групи СА-206
спеціальності 124 – «Системний аналіз»

 Вадим КАРАВАЄВ

Керівник: к.т.н., доц. каф. САІТ

 Євгеній КРИЖАНОВСЬКИЙ
«03» 06 2024 р.

Рецензент: к.т.н., доц. каф. КН

 Володимир ОЗЕРАНСЬКИЙ
«11» 06 2024 р.

Допущено до захисту

Завідувач кафедри САІТ

 д.т.н., проф. Віталій МОКІН
«04» 06 2024 р.

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність 124 – «Системний аналіз»
Освітньо-професійна програма – Системний аналіз

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

Мокін д.т.н., проф. Віталій МОКІН

“05” 03 2024 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Караваєву Вадиму Олександровичу

1. Тема роботи: «Системний аналіз геопросторових зв'язків масивів вод басейну річки Південний Буг»

керівник роботи: Крижановський Є. М., к.т.н., доц. каф. САІТ

затверджені наказом ВНТУ від «11» 03 2024 року № 80

2. Термін подання студентом роботи 31.05.

3. Вихідні дані до роботи:

1) Карта річки Південний Буг.

4. Зміст текстової частини:

1) Загальна характеристика об'єкту досліджень;

2) Ідентифікація геопросторових зв'язків масивів вод;

3) Аналіз і візуалізація результатів визначення геопросторових зв'язків масивів вод.

5. Перелік ілюстративного матеріалу:

1) Візуальне структурне представлення запропонованого підходу;

2) Блок-схема поетапного вирішення поставленого завдання;

3) Виділенні масиви річок у новому шарі catchment_Intersect;

4) Графік розподілу даних за геоприв'язкою та типом населеного пункту;

5) Тематична карта населених пунктів.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 11.03

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	11.03	31.03	вип
2	Порівняльний аналіз проблематики	01.04.	15.04	вип
3	Вибір оптимальних інформаційних технологій	16.04	30.04	вип
4	Здійснення системного аналізу геопросторових зв'язків масивів вод басейну річки Південний Буг	01.05	15.05	вип
5	Оформлення матеріалів до захисту БДР	16.05	31.05	вип

Студент

Вадим

Вадим КАРАВАЄВ

Керівник роботи

Євгеній

Євгеній КРИЖАНОВСЬКИЙ

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 71 сторінок формату А4, на яких є 37 рисунка, список використаних джерел містить 20 найменувань.

Метою роботи є здійснення системного аналізу геопросторових зв'язків масивів вод басейну річки Південний Буг.

В роботі розглянуто системний підхід до аналізу геопросторових зв'язків масивів вод басейну річки Південний Буг. Дослідження спрямоване на виявлення зв'язків між перетинами водозбірних площ масивів вод та іншими географічними об'єктами (населеними пунктами, річками, водосховищами) використовуючи ГІС та методи системного аналізу даних. Були ідентифіковані водозбірні площі в ArcGIS, здійснено оверлейний аналіз даних в ГІС, а його результати оброблені у GoogleColab за допомогою Python для створення таблиць та графіків. Отримані результати використані для створення тематичних карт, які сприяють кращому візуальному представленню просторових зв'язків водних масивів.

Ключові слова: інформаційна система, аналіз даних, системний аналіз, геоінформаційні системи, водні ресурси, Південний Буг.

ABSTRACT

The bachelor's thesis consists of 71 A4 pages, which contain 37 figures, the list of sources used contains 20 titles.

A systematic approach to analyzing the geospatial relations of water bodies in the Southern Bug River basin is considered. The study is aimed at identifying the connections between the intersections of the catchment areas of water bodies and other geographical objects (settlements, rivers, reservoirs) using GIS and methods of systematic data analysis. The watersheds were identified in ArcGIS, overlay analysis of the data was performed in GIS, and the results were processed in GoogleColab using Python to create tables and graphs. The results were used to create thematic maps that contribute to a better visual representation of the spatial relationships of water bodies.

Keywords: information system, data analysis, system analysis, geographic information systems, water resources, Southern Bug.

ЗМІСТ

ВСТУП.....	3
1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ’ЄКТУ ДОСЛІДЖЕНЬ	4
1.1 Опис об’єкта досліджень	4
1.2 Вибір оптимальних інформаційних технологій	9
1.3 Висновки.....	18
2. ІДЕНТИФІКАЦІЯ ГЕОПРОСТОРОВИХ ЗВ’ЯЗКІВ МАСИВІВ ВОД	19
2.1 Використання середовища ArcGIS для створення водозбірних площ	19
2.2 Інтеграція даних між ArcGIS та Python для проведення аналізу.....	24
2.3 Аугментація даних в середовищі Google Colab.....	26
2.4 Висновки.....	32
3. АНАЛІЗ І ВІЗУАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ВИЗНАЧЕННЯ ГЕОПРОСТОРОВИХ ЗВ’ЯЗКІВ МАСИВІВ ВОД	34
3.1 Створення графіків в середовищі Google Colab для візуалізації даних.	34
3.2 Створення тематичної карти засобами ArcGIS та Python.....	42
3.3 Висновки.....	50
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
Додаток А (обов’язковий). Технічне завдання.....	55
Додаток Б (обов’язковий). Протокол перевірки БДР на наявність текстових запозичень	57
Додаток В (довідниковий). Фрагмент лістингу програми	58
Додаток Г (обов’язковий). Ілюстративна частина	67

ВСТУП

Актуальність теми. Існуючі методи управління водними ресурсами часто базуються на фрагментованих даних і не враховують комплексного впливу факторів. Системний підхід з використанням ГІС та методів системного аналізу дозволяє об'єднувати та аналізувати різноманітні просторові дані для створення точніших і ефективніших результатів. Вивчення геопросторових зв'язків водних масивів допоможе розробити стратегії збору інформації з відкритих джерел, виявляти основні проблеми та підвищити ефективність заходів для мінімізації негативного антропогенного впливу.

Мета і задачі дослідження. Метою роботи є здійснення системного аналізу геопросторових зв'язків масивів вод басейну річки Південний Буг.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

1. Створити водозбірні площі у середовищі ГІС, що дозволить точно визначити межі басейнів річки Південний Буг та їхні характеристики;
2. Провести системний аналіз даних з використанням сучасної мови програмування та середовища розробки для обробки отриманих геопросторових даних;
3. Створити таблиці та графіки для подальшого аналізу результатів визначення геопросторових зв'язків водних масивів;
4. Створити тематичні карти просторових зв'язків та їх характеру.

Об'єктом дослідження є процес здійснення системного аналізу геопросторових зв'язків масивів вод басейну річки Південний Буг.

Предметом дослідження є методи і програмні засоби здійснення системного аналізу геопросторових зв'язків масивів вод басейну річки Південний Буг.

Публікації. За результатами даної роботи була зроблена доповідь на тему «Системний аналіз геопросторових зв'язків масивів вод басейну річки Південний Буг» у «Молодь в науці: дослідження, проблеми, перспективи (Вінниця, 2023-2024 рр.)» з публікацією тез [1], а також опубліковано одну фахову статтю «Метод аугментації текстів про стан масивів вод на основі інтелектуальної прив'язки до багатозв'язних геоінформаційних систем іменованих сутностей» у Віснику Вінницького політехнічного інституту (2023) [2].

1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ'ЄКТУ ДОСЛІДЖЕНЬ

1.1 Опис об'єкта досліджень

Об'єктом дослідження є басейн річки Південний Буг, який включає різноманітні водні масиви та водозбірні площі (рис. 1.1). Річка Південний Буг є однією з найважливіших водних артерій України, протікаючи через декілька областей та забезпечуючи водними ресурсами численні населені пункти, сільськогосподарські угіддя та промислові підприємства. Враховуючи змінність кліматичних умов та антропогенний вплив, необхідно проводити регулярний моніторинг та аналіз стану водних ресурсів [3].

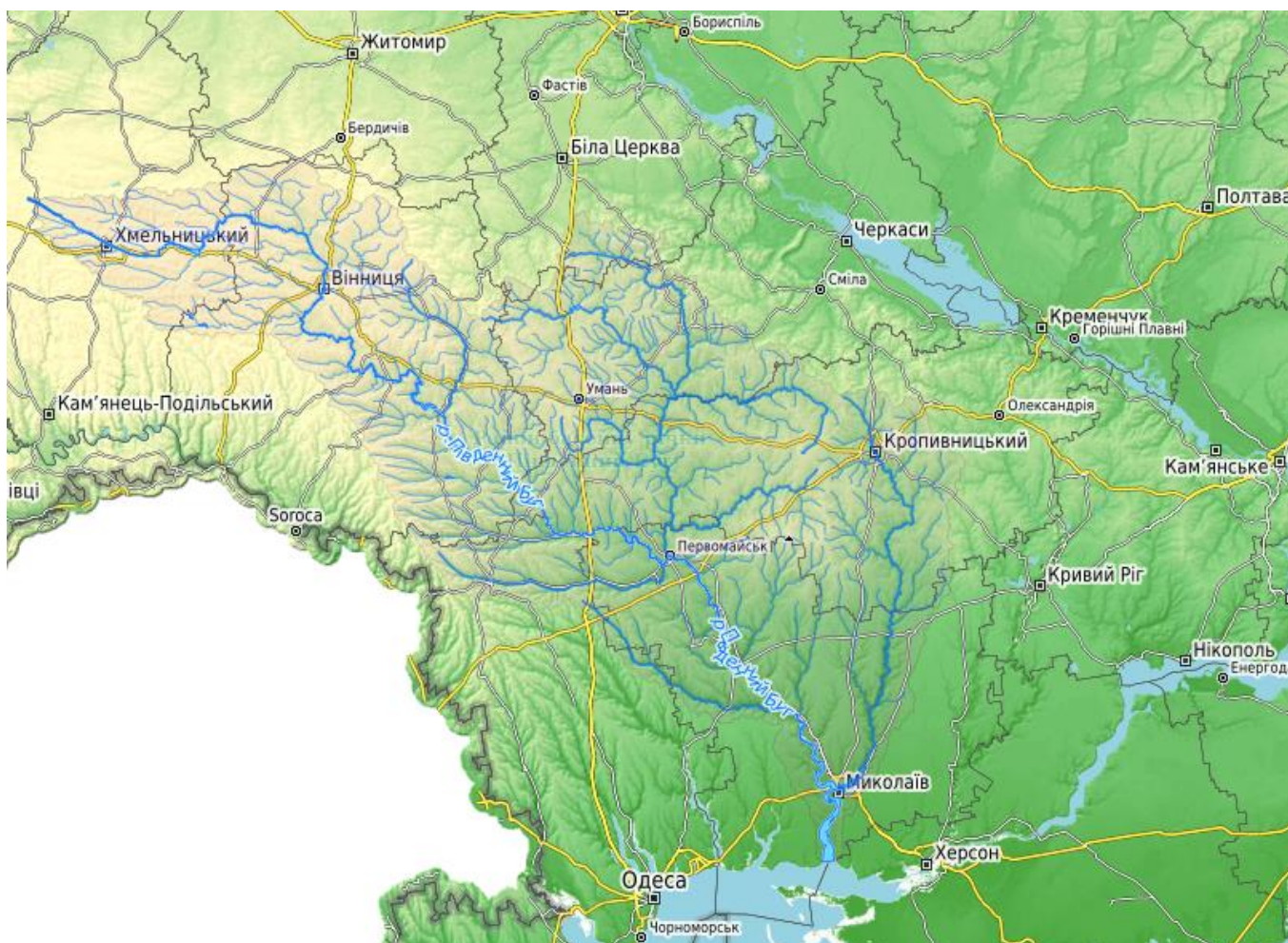


Рисунок 1.1 – Річкова мережа Південного Бугу

Світовий досвід довів, що найкращим способом представлення, зберігання та обробки інформації, яка має просторову складову (географічну прив'язку), є

геоінформаційні системи (ГІС). З точки зору призначення, ГІС — це інформаційна система, яка забезпечує збирання, зберігання, обробку, доступ, відображення та поширення просторово-орієнтованих даних (просторових даних). З точки зору програмно-інформаційної реалізації, ГІС — це сукупність електронних карт з умовними позначеннями об'єктів на них, баз даних з інформацією про ці об'єкти та програмного забезпечення для зручної роботи з картами і базами як з єдиним цілим.

Точне визначення меж басейнів річок є критично важливим для ефективного управління водними ресурсами з кількох екологічних причин. По-перше, чітке визначення меж водозбірних площ дозволяє точно оцінити обсяги доступних водних ресурсів у кожному басейні. Це є основою для планування водозабору та задоволення потреб населення, промисловості та сільського господарства. Відсутність точних меж може призвести до нерівномірного розподілу ресурсів і дефіциту води в окремих регіонах, що негативно впливає на забезпечення базових потреб людини і економічну діяльність.

По-друге, контроль якості води є неможливим без точного визначення меж басейнів. Знаючи межі, можна ідентифікувати джерела забруднення та спрямовувати зусилля на поліпшення якості води. Точні межі дозволяють розробити ефективні заходи для запобігання забрудненню, здійснювати моніторинг та контроль за діяльністю, яка може впливати на якість водних ресурсів. Це сприяє не лише покращенню здоров'я населення, але й збереженню екосистем, які залежать від чистої води.

По-третє, захист від повеней та інших природних катастроф також залежить від розуміння меж водозбірних площ. Це знання є важливим для прогнозування і управління ризиками повеней. Воно дозволяє розробляти ефективні заходи для захисту територій від затоплень, планувати будівництво захисних споруд та інфраструктури, що зменшує негативні наслідки природних катастроф. Без такого знання важко передбачити й мінімізувати шкоду, спричинену природними явищами.

Четвертим важливим аспектом є збереження екологічного балансу в регіоні. Визначення меж басейнів сприяє збереженню екологічного балансу, допомагає управляти водними ресурсами з урахуванням потреб екосистем, підтримувати біорізноманіття і забезпечувати природні процеси, такі як поповнення ґрунтових вод

та підтримка здорових водних екосистем. Це допомагає зберегти природу і створити умови для стійкого розвитку регіону.

П'ятий аспект стосується ефективного планування і управління територіями. Чітке розмежування басейнів сприяє кращому територіальному плануванню та управлінню земельними ресурсами. Це дозволяє інтегрувати водні ресурси в загальні плани розвитку регіону, враховуючи потреби різних секторів економіки та забезпечуючи гармонійний розвиток територій. Відсутність точних даних може призвести до хаотичного планування і неефективного використання ресурсів.

Шостий і, можливо, один з найважливіших аспектів — інформування політики та прийняття рішень. Точні дані про межі водозбірних площ є необхідними для розробки політики управління водними ресурсами. Вони надають науково обґрунтовану основу для прийняття рішень на державному і місцевому рівнях, допомагають визначати пріоритети і направляти інвестиції в інфраструктуру та заходи з управління водними ресурсами. Без такої інформації рішення можуть бути неефективними і навіть шкідливими для регіону.

Таким чином, точне визначення меж басейнів є основою для сталого та ефективного управління водними ресурсами. Це забезпечує їхню якість, кількість та захист від природних загроз, що в кінцевому результаті сприяє добробуту населення і збереженню навколишнього середовища. Геоінформаційні системи відіграють ключову роль у цьому процесі, забезпечуючи зручний і ефективний інструмент для роботи з просторовими даними.

У статті [4], підготовленій за участю авторів цього дослідження, обговорюється питання збору, систематизації та класифікації даних про стан водних масивів у річкових басейнах. Ці дані необхідні для розробки, впровадження та оцінки управлінських заходів, спрямованих на покращення або стабілізацію екологічного стану вод згідно з Водною Рамковою Директивою ЄС [5] та Водним Кодексом України [6].

Відповідно до цієї директиви та українського законодавства створюються плани управління річковими басейнами, які включають оцінку стану вод у всіх водних масивах та розробку і затвердження відповідних планів заходів для кожного з них, з подальшим регулярним моніторингом їх виконання [7]. Такий підхід забезпечує системний аналіз і стратегічне управління водними ресурсами,

дозволяючи адаптувати заходи до конкретних умов кожного басейну та забезпечуючи сталість екосистем.

Згідно з Водною Рамковою Директивою ЄС, одним з ключових аспектів є досягнення "доброго стану" всіх водних масивів, що включає як хімічні, так і екологічні параметри. Для цього необхідно мати детальну і точну інформацію про якість вод, гідрологічні характеристики та біологічні параметри водних екосистем. Водний Кодекс України гармонізовано з цими вимогами, що дозволяє інтегрувати національні стратегії управління водними ресурсами з європейськими стандартами.

Процес збору даних включає використання сучасних геоінформаційних систем (ГІС), автоматизованих систем моніторингу та лабораторних аналізів. ГІС-технології дозволяють інтегрувати дані з різних джерел, візуалізувати їх на карті та аналізувати просторові взаємозв'язки. Це забезпечує більш ефективне прийняття рішень щодо управління водними ресурсами та реалізації заходів з охорони довкілля.

Збір даних починається з використання ГІС, які можуть об'єднувати інформацію з різних джерел, таких як супутникові знімки, дані дистанційного зондування, топографічні карти та результати польових досліджень. Сучасні ГІС-технології дозволяють не тільки зберігати та обробляти велику кількість даних, але й виконувати складні просторові аналізи, які допомагають у визначенні тенденцій і взаємозв'язків у навколишньому середовищі.

Автоматизовані системи моніторингу, такі як сенсори, встановлені в різних точках водозбірного басейну, дозволяють отримувати дані в режимі реального часу. Ці системи можуть вимірювати різні параметри води, такі як температура, рівень рН, концентрація розчиненого кисню, наявність забруднюючих речовин і багато іншого. Дані, зібрані за допомогою автоматизованих систем, надходять до центральної бази даних, де вони обробляються та аналізуються для виявлення можливих загроз та аномалій.

Лабораторні дозволяють проводити детальний хімічний аналіз проб води, взятих у різних точках басейну. Це включає визначення концентрації важких металів, органічних сполук та інших потенційно шкідливих речовин, які можуть негативно впливати на екосистему.

Планування та впровадження заходів з управління водними ресурсами включає кілька етапів. Спочатку проводиться детальний аналіз поточного стану водних масивів, включаючи гідрологічні дослідження, хімічний аналіз води та оцінку біологічних параметрів. Потім розробляються плани заходів, які можуть включати будівництво очисних споруд, реставрацію природних водотоків, заходи з контролю забруднень тощо. Виконання цих заходів регулярно моніториться і оцінюється для забезпечення їх ефективності та адаптації до змін у стані водних масивів.

Детальний аналіз поточного стану водних масивів включає проведення гідрологічних досліджень, що допомагають зрозуміти обсяги та напрямки руху водних потоків, а також визначити критичні зони, що потребують особливої уваги. Хімічний аналіз води дозволяє визначити рівень забруднення та виявити джерела потенційних загроз, таких як промислові відходи або сільськогосподарські добрива. Оцінка біологічних параметрів включає вивчення біорізноманіття, стану флори та фауни у водних масивах, що є важливим індикатором загального екологічного стану.

На основі результатів цього аналізу розробляються конкретні плани заходів, які можуть включати будівництво очисних споруд, реставрацію природних водотоків, заходи з контролю забруднень та інші інфраструктурні проекти. Важливою складовою цих планів є також заходи з екологічної освіти та інформування громадськості, які сприяють підвищенню екологічної свідомості та активній участі громадян у заходах з охорони водного середовища.

Виконання цих заходів регулярно моніториться і оцінюється для забезпечення їх ефективності та адаптації до змін у стані водних масивів. Це включає постійний моніторинг якості води, аналіз результатів виконаних заходів та їх коригування у разі необхідності. Такий підхід дозволяє забезпечити сталий розвиток та збереження водних ресурсів для майбутніх поколінь.

Також важливою складовою є залучення громадськості та зацікавлених сторін до процесу планування та управління водними ресурсами. Відкритість і прозорість інформації про стан водних ресурсів і плани заходів сприяють підвищенню екологічної свідомості населення та активній участі громадян у заходах з охорони водного середовища.

Таким чином, системний аналіз геопросторових зв'язків масивів вод у басейні річки Південний Буг дозволяє ефективно управляти водними ресурсами, забезпечуючи їхню якість і кількість, а також захист від природних загроз, що в кінцевому результаті сприяє добробуту населення і збереженню навколишнього середовища.

1.2 Вибір оптимальних інформаційних технологій

Для вибору оптимальних інформаційних технологій, що використовуються для створення та аналізу геопросторових даних, необхідно розглянути кілька варіантів. Зокрема, слід проаналізувати можливості різних геоінформаційних систем (ГІС) та інструментів, які забезпечують роботу з просторовими даними.

Одним із ключових аспектів при виборі ГІС є їх здатність інтегрувати та обробляти великі обсяги геопросторових даних. Необхідно оцінити, наскільки ефективно кожна система справляється із завданнями імпорту, експорту та зберігання даних. Також важливо врахувати можливості системи щодо підтримки різних форматів даних, таких як растрові, векторні та табличні дані.

Візуалізація є важливим етапом аналізу, оскільки дозволяє представити результати у зручному та наочному форматі. Варто перевірити, чи надає система інструменти для створення карт, графіків, діаграм та інших візуальних представлень даних.

Слід розглянути питання підтримки та оновлення програмного забезпечення. Важливо, щоб обрана ГІС мала активну спільноту користувачів та регулярні оновлення, які забезпечують додавання нових функцій та виправлення помилок.

Основними завданнями дослідження є:

1. Створення водозбірних площ у середовищі ГІС, що дозволяє точно визначити межі басейнів річки Південний Буг та їхні характеристики;
2. Проведення системного аналізу даних за допомогою сучасної мови програмування для обробки отриманих геопросторових даних;
3. Створення таблиць та графіків для візуалізації та подальшого аналізу результатів визначення геопросторових зв'язків масивів вод;

4. Розробка тематичних карт, які відображатимуть просторові зв'язки та їх характер;
5. Навести приклади ефективності управління водними ресурсами через точний аналіз та візуалізацію даних, що сприяє прийняттю обґрунтованих рішень щодо використання і збереження водних ресурсів (рис. 1.2).

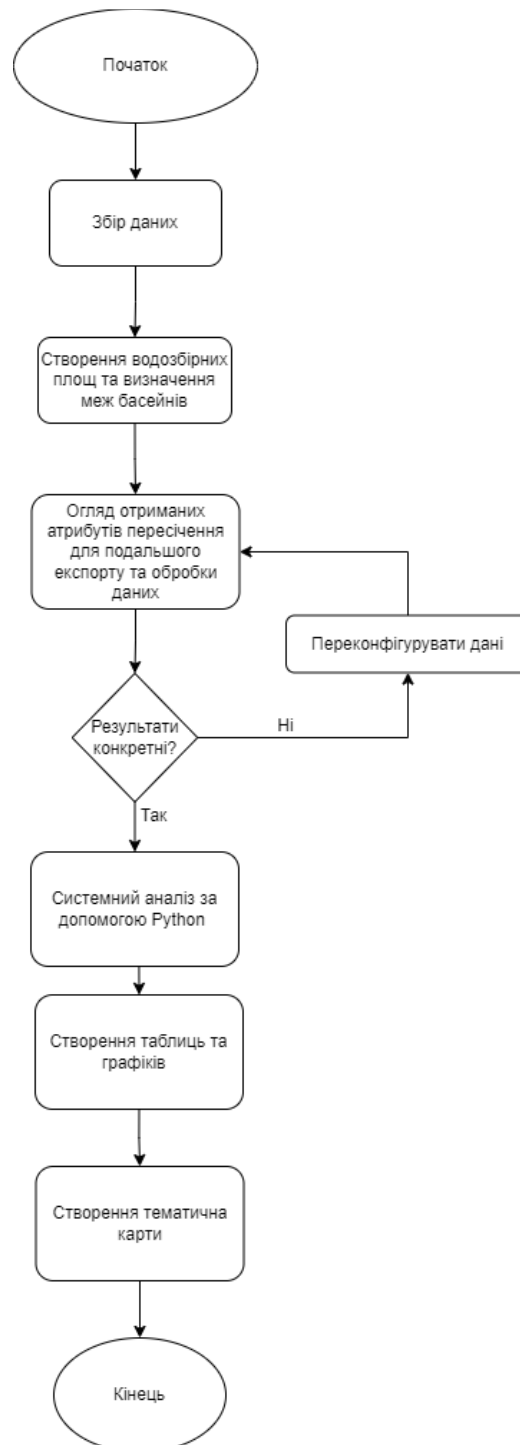


Рисунок 1.2 – Блок-схема поетапного вирішення поставленого завдання

Дослідження геопросторових зв'язків та водних масивів допоможе розробити стратегії збору інформації з відкритих джерел про конкретні водні об'єкти. Обробка зібраних даних дозволить виявити основні проблеми, що підвищить ефективність заходів, спрямованих на мінімізацію негативного антропогенного впливу.

Візуальне структурне представлення запропонованого підходу, створене за допомогою сервісу diagrams.net [8] (рис. 1.3).

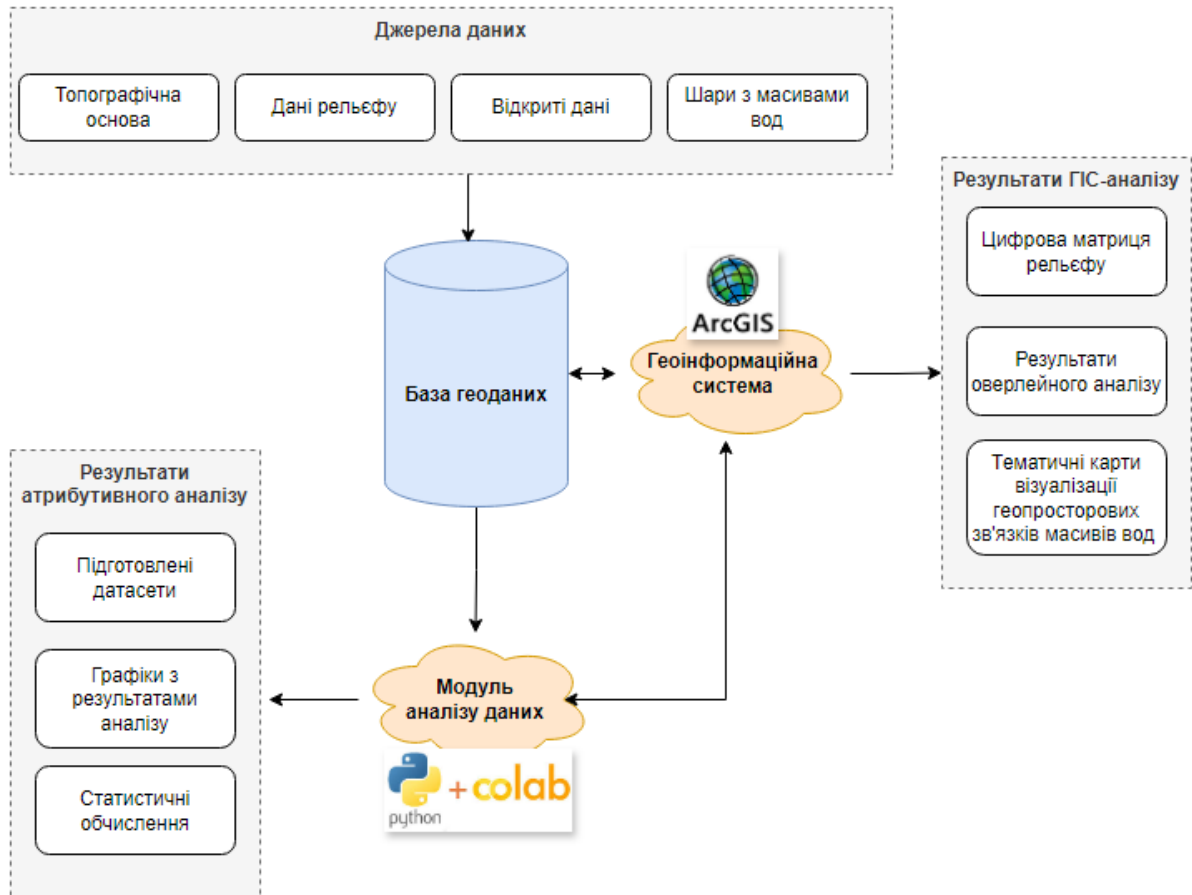


Рисунок 1.3 – Візуальне структурне представлення запропонованого підходу

Значна увага приділяється також використанню мови програмування Python та відповідних бібліотек (Pandas, Seaborn, Matplotlib) для обробки та аналізу даних. Це дозволяє автоматизувати процеси аналізу, підвищити точність та оперативність отримання результатів.

Аналіз просторових Існуючі методи управління водними ресурсами часто базуються на фрагментованих даних і не враховують комплексного впливу різних факторів. Системний підхід, що використовує можливості геоінформаційних систем (ГІС) та системний аналіз, дозволяє об'єднати та аналізувати різноманітні просторові

дані для створення більш точних і ефективних результатів (рис. 1.4). Вивчення геопросторових зв'язків і динаміки водних масивів дозволить розробити стратегії, спрямовані на збереження водних екосистем, підвищення ефективності використання водних ресурсів та мінімізацію негативного впливу людської діяльності зв'язків [9].

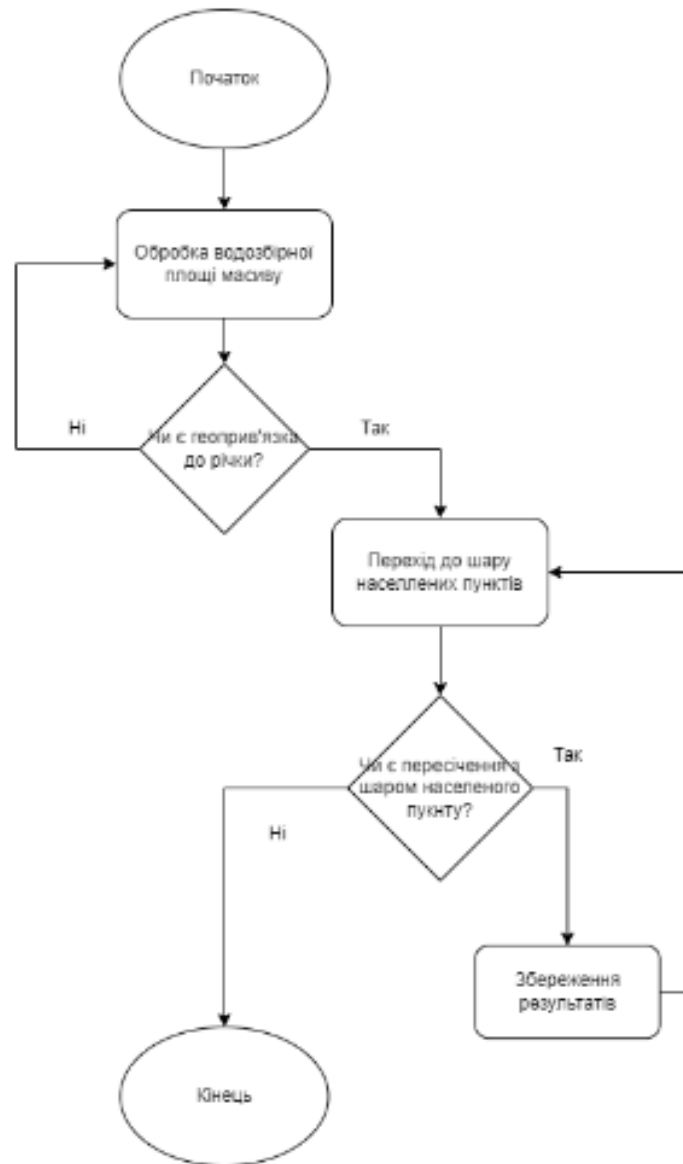


Рисунок 1.4 – Блок-схема обробки даних.

ArcGIS – це потужна геоінформаційна система, розроблена компанією Esri, яка використовується для створення, управління, аналізу та відображення геопросторових даних. Ця система дозволяє користувачам працювати з великими наборами даних та проводити складні просторові аналізи для різних галузей, таких як екологія, урбаністика, інфраструктура, сільське господарство та багато інших.

Основні можливості ArcGIS включають створення та управління геоданими, аналіз просторових даних для виявлення закономірностей та тенденцій, візуалізацію даних за допомогою карт, графіків та інших інструментів, підтримку різних форматів даних та інтеграцію з іншими системами. ArcGIS надає численні методи та технології для визначення водозбірних площ, що робить її важливим інструментом для гідрологічного аналізу та управління водними ресурсами [10].

Після розгляду основних варіантів геоінформаційних систем, обраний варіант для реалізації проекту "Системний аналіз геопросторових зв'язків масивів вод басейну річки Південний Буг" – ArcGIS. Цей вибір обумовлений такими причинами: можливості аналізу та візуалізації, інтеграція з іншими системами, професійна підтримка та документація, створення водозбірних площ. ArcGIS надає потужні інструменти для аналізу та візуалізації геопросторових даних. Це дозволяє проводити детальний аналіз водозбірних площ та інших гідрологічних параметрів, що є критично важливим для ефективного управління водними ресурсами.

У рамках проекту було використано ArcGIS для створення водозбірних площ масивів вод басейну річки Південний Буг. Було проведено аналіз просторових даних, визначено межі басейнів та створено тематичні карти, що відображають різні аспекти водних ресурсів (рис. 1.5).

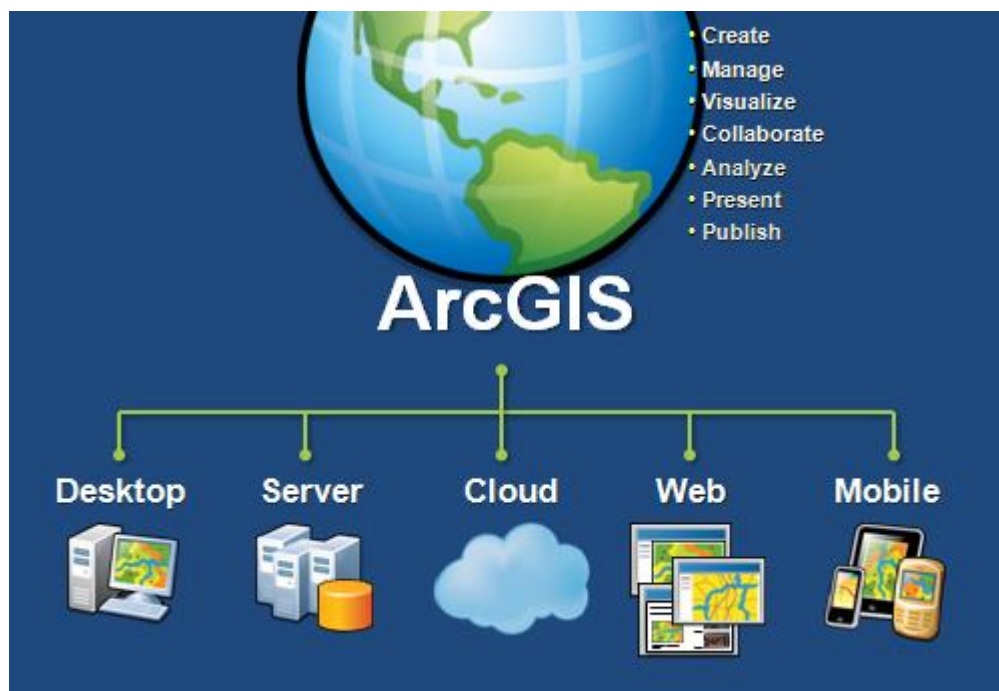


Рисунок 1.5 – Переваги ArcGIS

Додатково, у режимі компонування в ArcGIS була додана легенда, що допомагає користувачам зрозуміти значення кольорів та інших символів на карті, роблячи візуалізацію більш зрозумілою та інформативною.

Таким чином, використання ArcGIS забезпечило високу точність та ефективність у створенні та аналізі геопросторових даних, що сприяло досягненню цілей проекту та забезпечило основу для подальших досліджень та управлінських рішень.

OpenStreetMap (OSM) – це відкритий та спільнотний проект, метою якого є створення безкоштовної, редагованої та доступної для всіх геопросторової бази даних світу [11]. OSM надає детальні карти, які створюються та підтримуються волонтерами з усього світу. Основні переваги OpenStreetMap включають відкритість та доступність даних, можливість безкоштовного використання, змінення та розповсюдження даних, спільнотний підхід до оновлення та покращення карт. Відкритий доступ до геопросторових даних дозволяє використовувати OSM для різних цілей, від особистих проектів до комерційного використання (рис. 1.6).

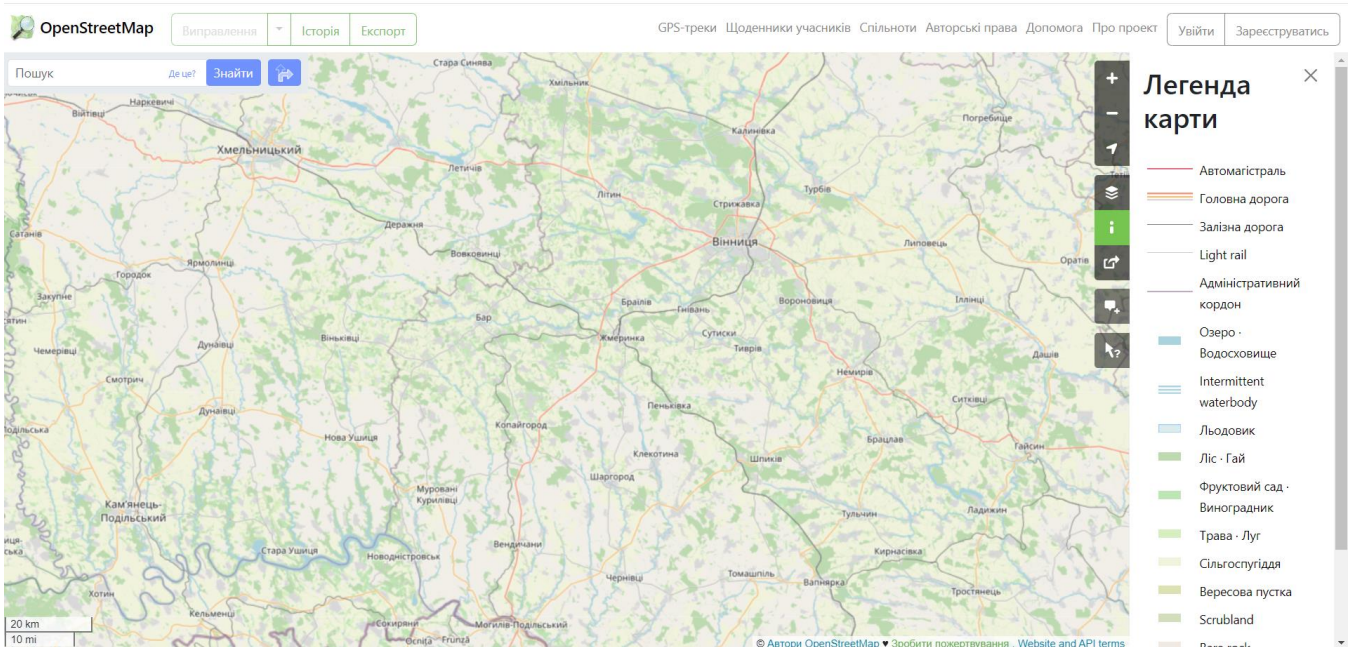


Рисунок 1.6 – Загальний вигляд сервісу OpenStreetMap.

OSM підтримується великою спільнотою волонтерів, які додають нові дані, виправляють помилки та покращують існуючу інформацію. Це дозволяє проекту швидко оновлюватися та залишатися актуальним. Така колаборація сприяє

створенню детальних та точних карт, які відображають реальний стан речей на місцях.

Гнучкість та масштабованість OSM дозволяють використовувати дані для різних додатків та сервісів, таких як навігаційні системи, геоінформаційні системи (ГІС), аналіз просторових даних та інше. Дані OSM можна легко інтегрувати з іншими інструментами та технологіями, що робить їх корисними для широкого кола користувачів.

Для роботи з даними OSM існує безліч інструментів та API, які дозволяють розробникам отримувати доступ до карт, здійснювати запити до бази даних, створювати візуалізації та інтегрувати дані у власні додатки. Це робить OSM потужним інструментом для розробників та аналітиків, які працюють з геопросторовими даними.

Google Colab (або Google Colaboratory) – це безкоштовний онлайн-сервіс від Google, який дозволяє писати та виконувати Python-код у веб-браузері з доступом до потужних обчислювальних ресурсів, таких як GPU та TPU. Colab є ідеальним інструментом для машинного навчання, аналізу даних і наукових досліджень, завдяки своїм численним можливостям та перевагам [12].

Google Colab працює в хмарному середовищі, тому не потребує встановлення додаткового програмного забезпечення, що робить його доступним з будь-якого пристрою, підключеного до Інтернету. Це дозволяє користувачам уникнути складних процедур налаштування та конфігурації, зосереджуючись безпосередньо на розробці та аналізі даних.

Хмарне середовище Colab забезпечує безкоштовний доступ до обчислювальних ресурсів, таких як GPU і TPU, які автоматично виділяються та масштабуються відповідно до потреб користувача, що особливо корисно для завдань глибокого навчання. Це дозволяє швидко та ефективно виконувати обчислення, які б вимагали значного часу та ресурсів на звичайних комп'ютерах.

Colab підтримує формат Jupyter Notebook, дозволяючи створювати інтерактивні ноутбуки, що включають код, текстові блоки, графіки та діаграми. Ця інтерактивність дозволяє легко комбінувати пояснювальні тексти з виконуваним кодом, що сприяє кращому розумінню та документуванню процесу аналізу даних. Крім того, Colab підтримує численні бібліотеки Python для наукових розрахунків,

машинного навчання та візуалізації даних, такі як TensorFlow, Keras, NumPy, Pandas, Matplotlib та інші, що робить його потужним інструментом для різноманітних дослідницьких завдань.

Ще однією важливою перевагою Google Colab є можливість спільної роботи. Кілька користувачів можуть одночасно працювати над проектами в реальному часі, що полегшує співпрацю та обмін ідеями.

Це особливо корисно для команд, які працюють над спільними проектами, оскільки дозволяє синхронізувати зміни та забезпечувати ефективну комунікацію між учасниками. Всі зміни, зроблені в ноутбуках, автоматично зберігаються на Google Drive, забезпечуючи надійне збереження даних та доступ до попередніх версій (рис. 1.7).

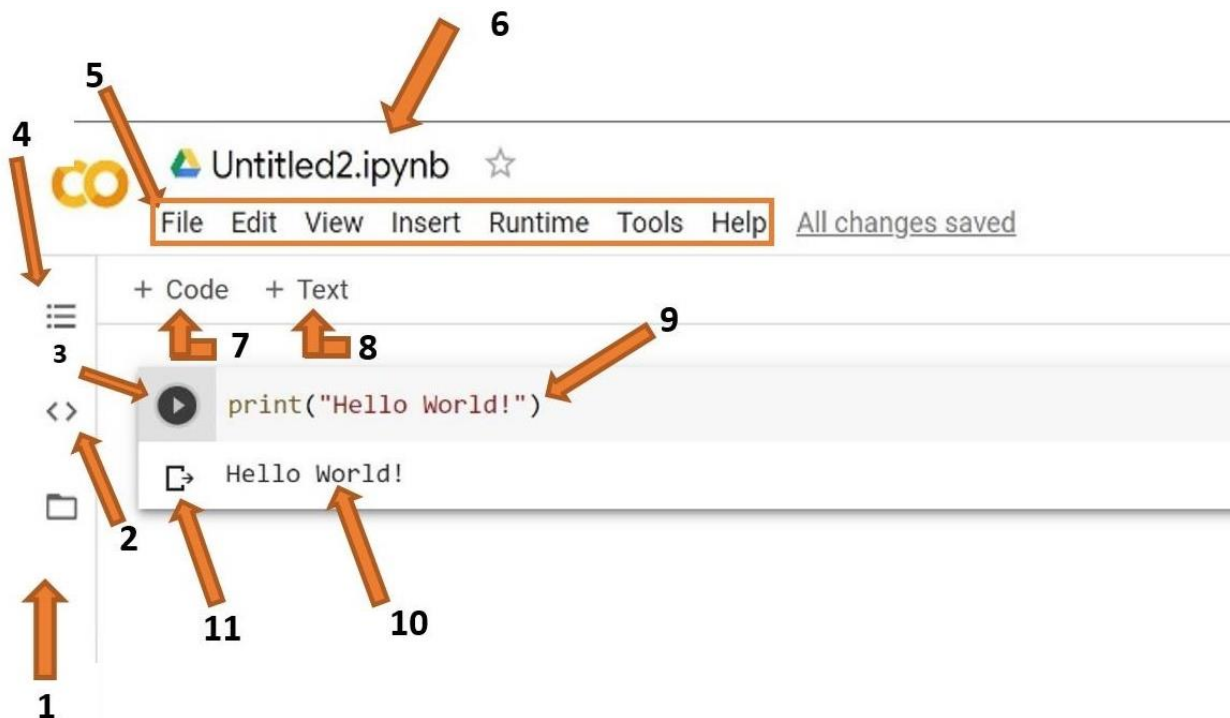


Рисунок 1.7 – Загальний вигляд робочого середовища Google Colab

Google Colab є потужним інструментом, що забезпечує простоту використання, потужні обчислювальні ресурси та підтримку спільної роботи, роблячи його ідеальним вибором для багатьох наукових та інженерних завдань.

Python – це високорівнева мова програмування, яка широко використовується для аналізу даних, машинного навчання, веб-розробки та автоматизації. Завдяки простому синтаксису і великій кількості бібліотек, Python став одним із найпопулярніших інструментів серед дослідників даних та інженерів.

Однією з основних переваг Python є його багатий набір бібліотек, спеціально розроблених для обробки, аналізу та візуалізації даних. Нижче представлено короткий огляд деяких із найважливіших бібліотек, які використовуються для аналізу даних: Pandas, Seaborn та Matplotlib.

Pandas – це потужна бібліотека для маніпуляції та аналізу даних, створена для роботи з табличними даними. Вона забезпечує високу продуктивність і простоту використання для обробки великих наборів даних. Основна структура даних у Pandas, DataFrame, дозволяє зручно працювати з табличними даними, що мають ряди та стовпці. Ця бібліотека надає інструменти для фільтрації, сортування, агрегації, групування та перетворення даних. Вона також підтримує імпорт та експорт даних з різних джерел, таких як CSV, Excel, SQL-бази даних, і забезпечує можливість обробки пропущених значень, дублікати та інші аномалії в даних.

Seaborn – це бібліотека для візуалізації даних, побудована на базі Matplotlib. Вона спрощує створення привабливих і складних статистичних графіків. Seaborn дозволяє легко створювати графіки розподілу, парні графіки, коробкові діаграми, тематичні карти та багато інших типів візуалізацій. Вона також надає набір стилів і тем для покращення вигляду графіків і підтримує роботу з DataFrame з Pandas, що дозволяє легко створювати графіки безпосередньо з табличних даних. Крім того, Seaborn пропонує інструменти для вивчення взаємозв'язків між змінними, такі як регресійні лінії та спільні розподіли.

Matplotlib – це базова бібліотека для створення статичних, анімаційних та інтерактивних візуалізацій у Python. Вона дозволяє створювати лінійні графіки, гістограми, діаграми розсіювання та інші типи візуалізацій. Matplotlib надає високий рівень кастомізації графіків, включаючи налаштування кольорів, шрифтів, міток осей та легенд. Вона також підтримує інтеграцію з іншими бібліотеками для наукових обчислень, такими як Pandas та SciPy, і використовується як основа для інших візуалізаційних бібліотек, таких як Seaborn.

Google Colab інтегрується з цими бібліотеками, дозволяючи користувачам ефективно використовувати їх для аналізу даних та створення візуалізацій у хмарному середовищі. Colab підтримує формат Jupyter Notebook, дозволяючи створювати інтерактивні ноутбуки, що включають код, текстові блоки, графіки та

діаграми. Це інтерактивне середовище також сприяє спільній роботі, дозволяючи кільком користувачам одночасно працювати над проектами в реальному часі.

Colab забезпечує безкоштовний доступ до обчислювальних ресурсів, таких як GPU і TPU, які автоматично виділяються та масштабуються відповідно до потреб користувача, що особливо корисно для завдань глибокого навчання. Всі зміни, зроблені в ноутбуках, автоматично зберігаються на Google Drive, забезпечуючи надійне збереження даних та доступ до попередніх версій.

Таким чином, Google Colab у поєднанні з потужними бібліотеками Python, такими як Pandas, Seaborn та Matplotlib, забезпечує ефективний інструментарій для обробки, аналізу та візуалізації даних, що робить його незамінним для дослідників та розробників у багатьох галузях.

1.3 Висновки

В ході аналізу розглянуто основні аспекти басейну річки Південний Буг, включаючи його географічні, гідрологічні та екологічні характеристики. Детальний опис басейну річки заклав основу для подальшого дослідження.

Проаналізовано сучасні методи та програмні технології для аналізу геопросторових даних. Використання ArcGIS для створення водозбірних площ та Python у Google Colab для системного аналізу даних виявилось оптимальним вибором.

Загалом, перший розділ роботи забезпечує міцну базу для подальших досліджень, зосереджуючись на важливих аспектах управління водними ресурсами.

2 ІДЕНТИФІКАЦІЯ ГЕОПРОСТОРОВИХ ЗВ'ЯЗКІВ МАСИВІВ ВОД

2.1 Використання середовища ArcGIS для створення водозбірних площ

Для створення водозбірних площ у ArcGIS спочатку необхідно зібрати і підготувати демографічні та гідрологічні дані, зокрема дані про рельєф місцевості (Digital Elevation Model, DEM), гідрологічні об'єкти (річки, озера), кліматичні умови та інші релевантні дані. Ці дані повинні пройти попередню обробку, яка включає очищення, формування єдиного формату, перевірку на відповідність та вирішення проблем з відсутніми або некоректними даними. Першим етапом процесу є підготовка вихідних даних. Це включає збір і валідацію цифрових моделей рельєфу (ЦМР), гідрографічних мереж, а також інших геопросторових даних, таких як типи ґрунтів і використання земель. Ці дані імпортуються в ArcGIS для подальшої обробки [13].

Для здійснення системного аналізу геопросторових зв'язків масивів вод басейну річки Південний Буг використовується програмне середовище ArcGIS, що дозволяє створювати, аналізувати та візуалізувати геопросторові дані. Використання ArcGIS допомагає виявити просторові зв'язки та закономірності, які впливають на формування та розподіл водних ресурсів у басейні.

Наступним кроком є використання інструментів ArcGIS для визначення водозбірних площ. Одним із ключових інструментів є функція "Hydrology" у розділі "Spatial Analyst Tools". Цей інструмент включає кілька основних операцій:

1. Fill (Заповнення): Ця операція вирівнює западини в ЦМР, що дозволяє воді текти без перешкод;
2. Flow Direction (Напрямок потоку): Визначає напрямок потоку води для кожної клітинки на основі ЦМР;
3. Flow Accumulation (Накопичення потоку): Обчислює накопичення потоку води, показуючи, скільки води тече через кожну клітинку;
4. Stream Definition (Визначення річок): Визначає річкові потоки на основі значень накопичення потоку;
5. Watershed (Водозбір): Визначає межі водозбірних площ на основі даних про напрямок потоку та річкові мережі.

Використовуючи ці інструменти, ArcGIS автоматично визначає межі водозбірних площ для кожної річки у масиві. Результати обробки представлені у вигляді полігональних шарів, що відображають водозбірні площі [14]. Ці полігони можуть бути додатково аналізовані та інтегровані з іншими геопросторовими даними (рис. 2.1).

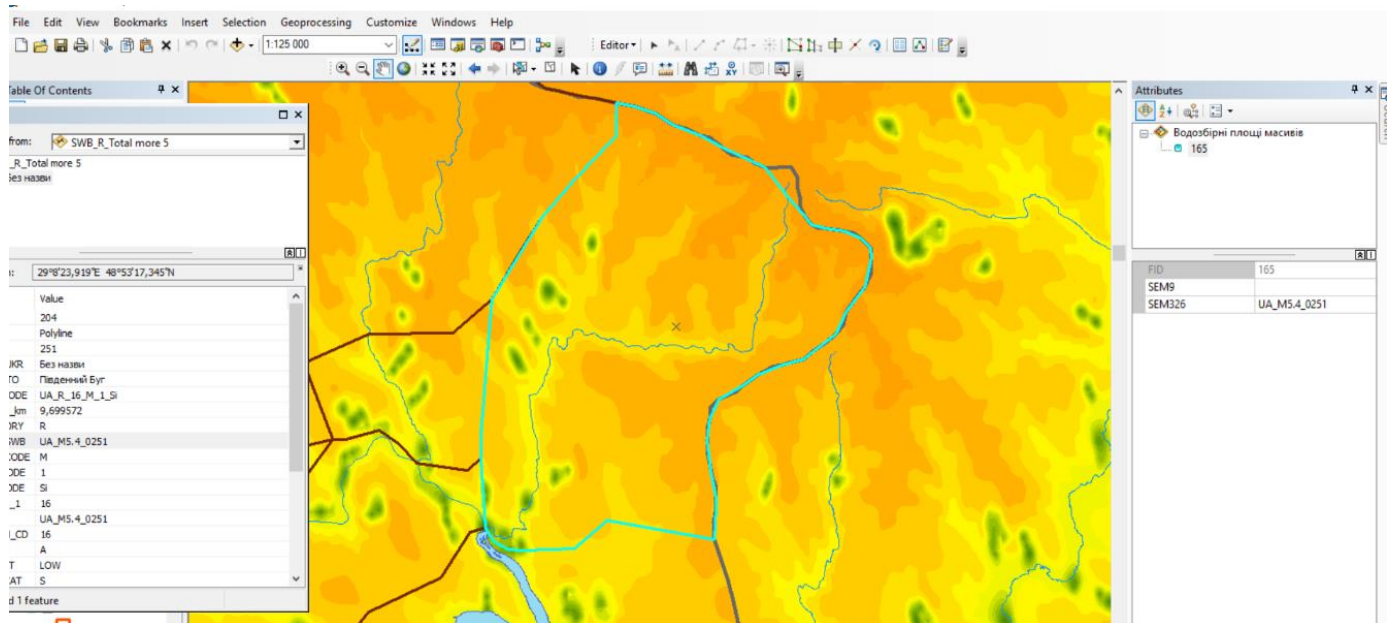


Рисунок 2.1 – Приклад створення водозбірної площі

Наступним кроком є заповнення карти річок Південного Бугу водозбірними площами (рис. 2.2). Цей процес починається з детального обмальовування річки по рельєфу, що дозволяє визначити точні межі водозбірних площ. Важливо враховувати всі особливості рельєфу, щоб правильно відобразити напрямки стоку води та зону впливу кожної ділянки річки [15].

Для досягнення цієї мети потрібно ретельно проаналізувати топографічні дані, враховуючи висоту, схили, і особливості ландшафту. Це дозволить визначити природні вододіли, які обмежують водозбірні площі. Особлива увага приділяється визначенню точок злиття річок і струмків, які суттєво впливають на формування водозбірних басейнів.

Крім того, врахування напрямку стоку води є критичним, оскільки він впливає на розподіл водних ресурсів і потенційний вплив на навколишні території. Правильне відображення зони впливу кожної ділянки річки забезпечує точність даних, які будуть використовуватися для подальшого аналізу та моделювання [16].

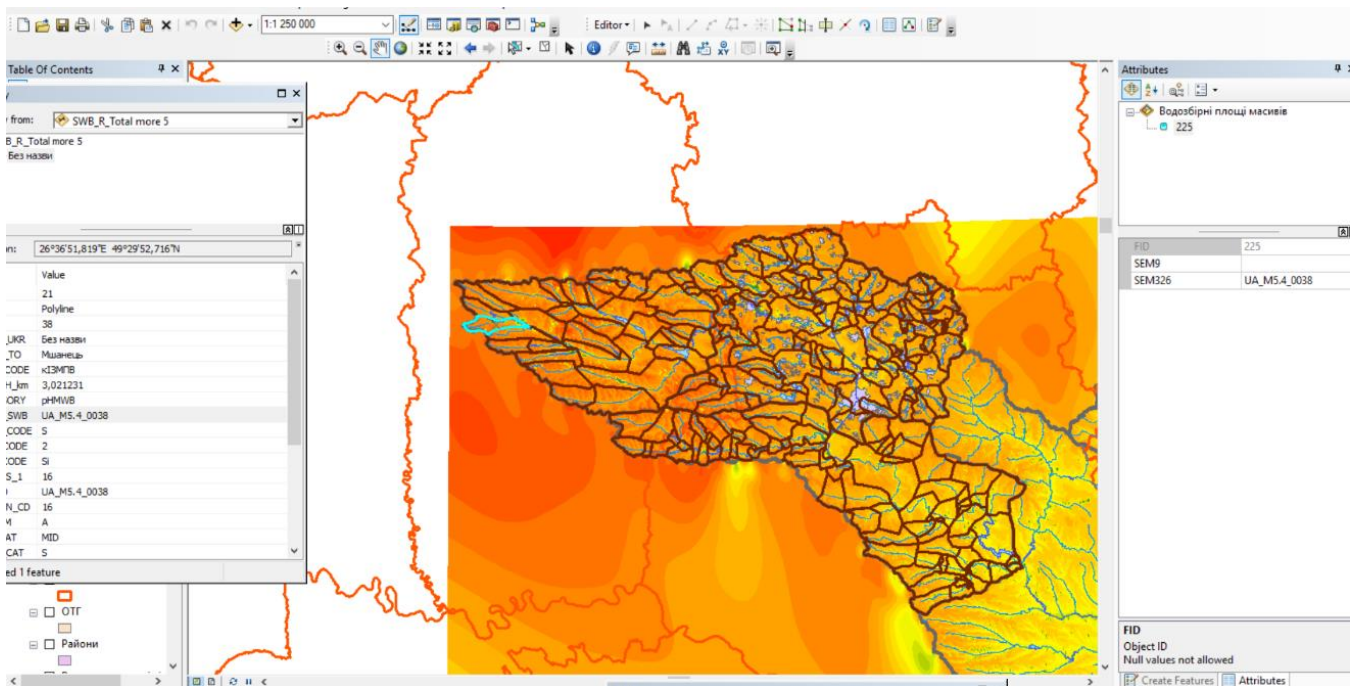


Рисунок 2.2 – Завершення заповнення карти водозбірними площами

Після цього проводиться аналіз пересічення, який включає обчислення напрямків потоків води за допомогою інструменту Intersect (рис. 2.3).

T-SBB - ArcMap

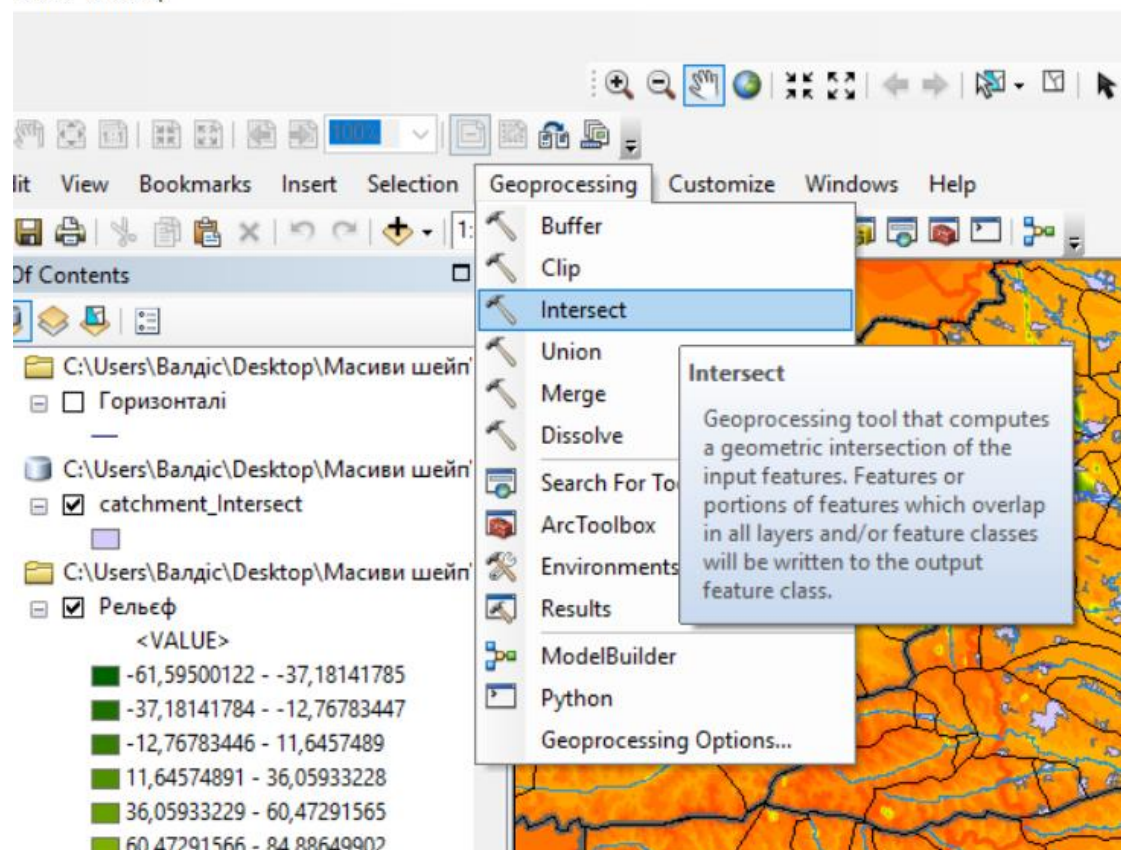


Рисунок 2.3 – Інструмент Intersect

Цей інструмент визначає місця перетину річок з іншими географічними об'єктами, такими як населені пункти, дороги або інфраструктурні об'єкти. Аналіз пересічення дозволяє зрозуміти, як річки взаємодіють з різними елементами ландшафту і де можливі проблемні зони, такі як ризик підтоплення або забруднення водних ресурсів (рис. 2.4).

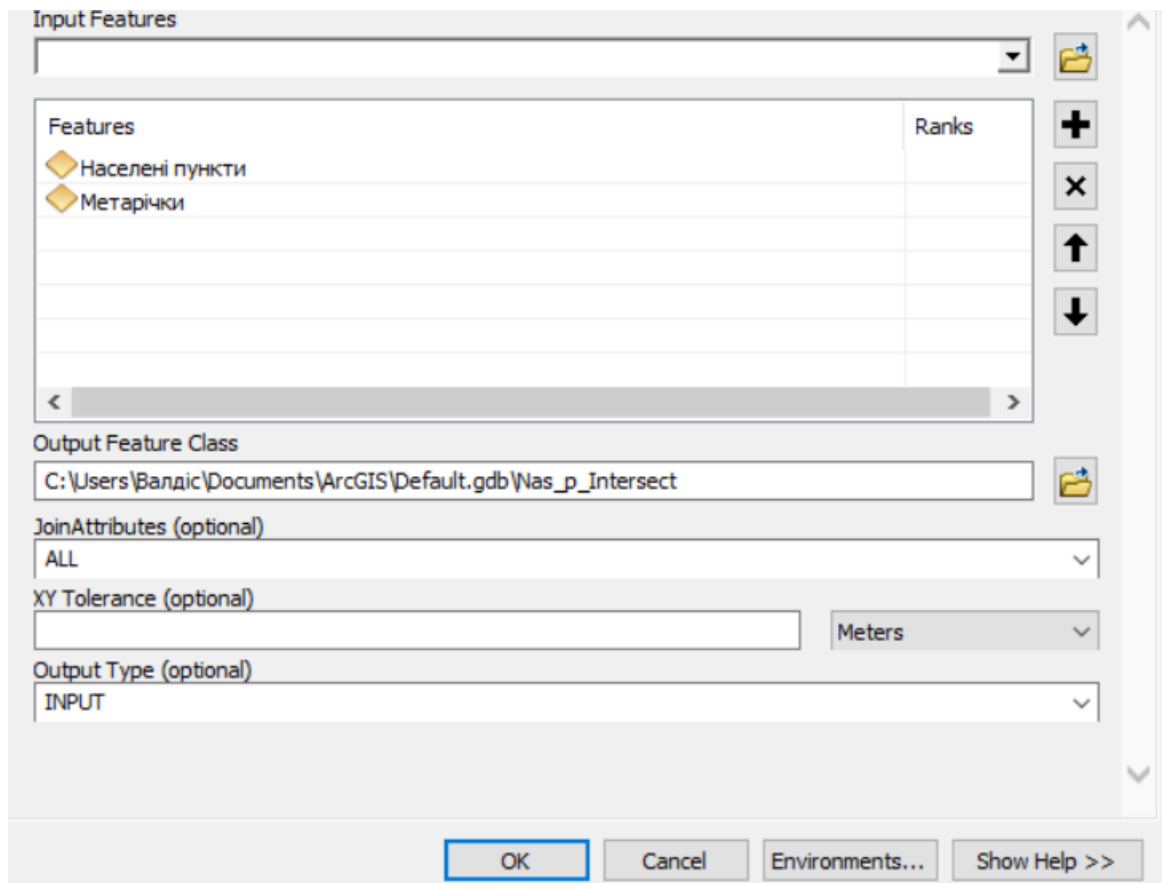


Рисунок 2.4 – Застосування інструменту Intersect

Завдяки Intersect, можна також визначити напрямки течії води на основі моделі рельєфу. Це включає обчислення, у яких напрямках тече вода, враховуючи природні особливості рельєфу, такі як схили, долини та інші топографічні елементи [17].

Такий підхід дозволяє отримати більш повну картину гідрологічних процесів у басейні річки Південний Буг та краще зрозуміти, як вода розподіляється по території (рис. 2.5).

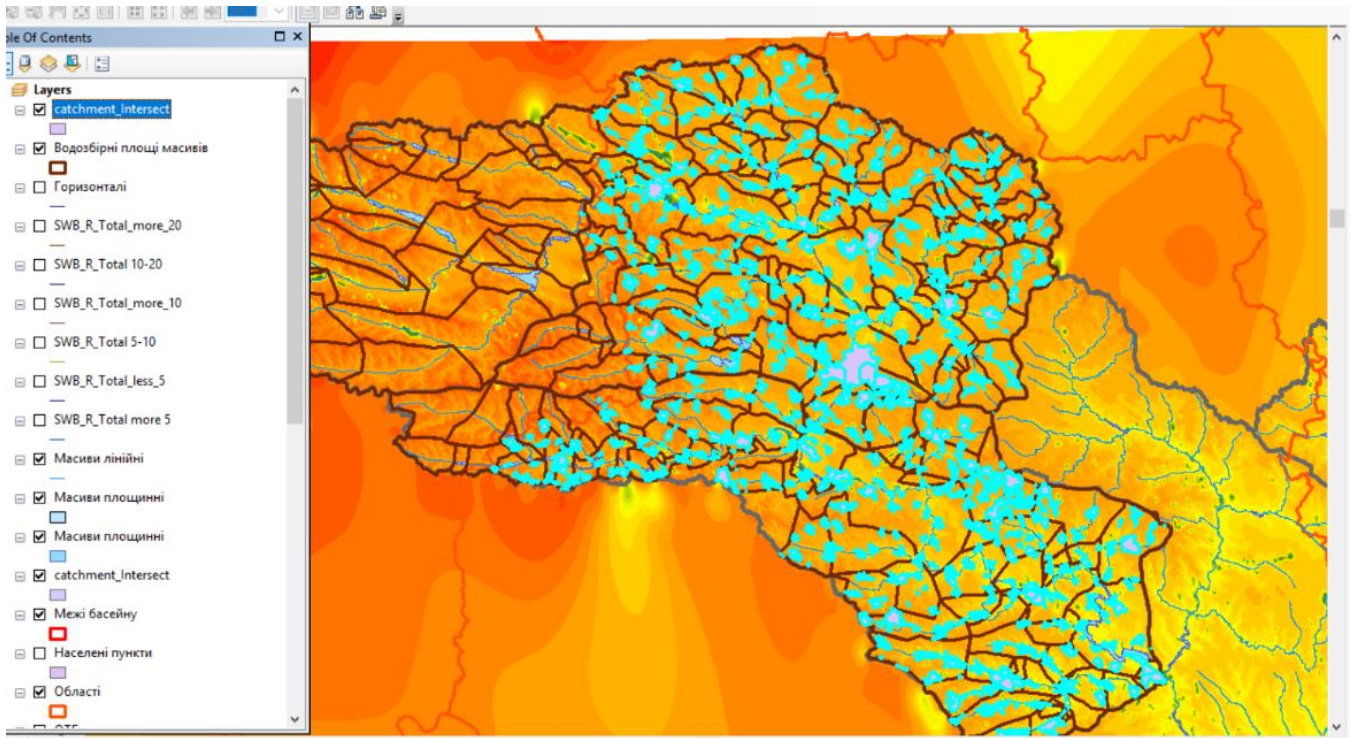


Рисунок 2.5 – Виділенні масиви річок у новому шарі catchment_Intersect

За бажанням, можна додатково включити в аналіз інші шари даних, наприклад, землекористування, екологічні зони або інші природні та антропогенні фактори. Це дозволяє створити багат шарову модель, яка відображає всі ключові аспекти водного режиму басейну та допомагає приймати більш обґрунтовані рішення щодо управління водними ресурсами.

Як результат ми отримуємо таблицю атрибутів з шару «catchment_Intersect» з якою будемо працювати далі для більш глибокого аналізу (рис. 2.6).

FID	Shape	FID_catchment	SEM9	SEM126	FID_Метарічки	SEM9	SEM15	SEM150	SEM152	SEM157	SEM112	SEM4	SEM7	SEM11	SEM20	SEM34	SEM35	SEM36
1	Polyline	155	UA_M5.4_0254	48	Без назви (с. Зарічне)	МЕНЕЕ 20 М		0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
2	Polyline	155	UA_M5.4_0254	48	Без назви (с. Зарічне)	МЕНЕЕ 20 М		0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
3	Polyline	169	UA_M5.4_0272	49	Самець (р. Сільниця)	МЕНЕЕ 20 М	277	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
4	Polyline	169	UA_M5.4_0272	49	Самець (р. Сільниця)	МЕНЕЕ 20 М	277	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
5	Polyline	173	UA_M5.4_0273	49	Самець (р. Сільниця)	МЕНЕЕ 20 М	277	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
6	Polyline	174	UA_M5.4_0274	49	Самець (р. Сільниця)	МЕНЕЕ 20 М	277	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
7	Polyline	179	UA_M5.4_0256	50	Тупчинка	МЕНЕЕ 20 М	280	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
8	Polyline	179	UA_M5.4_0256	50	Тупчинка	МЕНЕЕ 20 М	280	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
9	Polyline	181	UA_M5.4_0263	50	Тупчинка	МЕНЕЕ 20 М	280	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
10	Polyline	181	UA_M5.4_0263	50	Тупчинка	МЕНЕЕ 20 М	280	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
11	Polyline	181	UA_M5.4_0263	50	Тупчинка	МЕНЕЕ 20 М	280	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
12	Polyline	181	UA_M5.4_0263	50	Тупчинка	МЕНЕЕ 20 М	280	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
13	Polyline	177	UA_M5.4_0259	51	Кістява	МЕНЕЕ 20 М	279	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
14	Polyline	182	UA_M5.4_0269	51	Кістява	МЕНЕЕ 20 М	279	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
15	Polyline	182	UA_M5.4_0269	51	Кістява	МЕНЕЕ 20 М	279	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
16	Polyline	185	UA_M5.4_0270	51	Кістява	МЕНЕЕ 20 М	279	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
17	Polyline	185	UA_M5.4_0270	51	Кістява	МЕНЕЕ 20 М	279	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
18	Polyline	185	UA_M5.4_0270	51	Кістява	МЕНЕЕ 20 М	279	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
19	Polyline	175	UA_M5.4_0260	52	Самотня	МЕНЕЕ 20 М		0		0		0	0	0	1	0		ПОСТОЯННІ
20	Polyline	155	UA_M5.4_0254	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
21	Polyline	155	UA_M5.4_0254	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
22	Polyline	155	UA_M5.4_0254	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
23	Polyline	171	UA_M5.4_0262	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
24	Polyline	174	UA_M5.4_0274	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
25	Polyline	175	UA_M5.4_0260	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
26	Polyline	175	UA_M5.4_0260	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
27	Polyline	176	UA_M5.4_0255	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
28	Polyline	176	UA_M5.4_0255	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
29	Polyline	176	UA_M5.4_0255	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
30	Polyline	177	UA_M5.4_0259	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
31	Polyline	179	UA_M5.4_0256	117	Сільниця	МЕНЕЕ 20 М	276	0		0	31410000L	0	0	0	1	0		ПОСТОЯННІ
32	Polyline	18	UA_M5.4_0126	149	Радуха	МЕНЕЕ 20 М	326	0		0		0	0	0	1	0		ПОСТОЯННІ
33	Polyline	118	UA_M5.4_0214	151	Розсохувата	МЕНЕЕ 20 М	300	0		0		0	0	0	1	0		ПОСТОЯННІ
34	Polyline	119	UA_M5.4_0213	151	Розсохувата	МЕНЕЕ 20 М	300	0		0		0	0	0	1	0		ПОСТОЯННІ

Рисунок 2.6 – Таблиця атрибутів шару catchment_Intersect

Використовуючи інструменти ArcGIS, створюються карти та схеми, що відображають межі водозбірних площ, гідрологічні об'єкти та інші важливі характеристики. Також можна створювати тематичні карти, які візуалізують різні параметри, такі як інтенсивність опадів, об'єми води, розподіл забруднень тощо.

ArcGIS дозволяє інтегрувати водозбірні площі з іншими геопросторовими даними, такими як земельні ресурси, інфраструктура, екологічні зони, що сприяє проведенню комплексного аналізу.

Система також підтримує інтеграцію з іншими інформаційними системами та базами даних, що дозволяє проводити більш складні аналізи та приймати обґрунтовані рішення.

Перевагами використання ArcGIS є висока точність і надійність при визначенні меж водозбірних площ, гнучкість та масштабованість для обробки великих обсягів даних, потужні інструменти для візуалізації та інтерпретації даних, а також можливість інтеграції з Python для автоматизації процесів та проведення складних аналізів.

Таким чином, використання ArcGIS для створення водозбірних площ є важливим інструментом для ефективного управління водними ресурсами, надаючи можливість проводити точний аналіз та приймати обґрунтовані рішення.

2.2 Інтеграція даних між ArcGIS та Python для проведення аналізу

Для подальшої роботи з даними необхідно перенести їх з середовища ArcGIS в Google Colab, використовуючи відповідні інструменти ArcGIS. Цей процес дозволяє використовувати Python для більш детального аналізу та обробки даних, що надає більшу гнучкість та можливості для візуалізації та моделювання.

Перший крок – це експорт даних з ArcGIS у формат, який можна легко імпортувати в Google Colab. Для цього відкриваємо середовище ArcGIS і переходимо до панелі з назвою ArcToolbox, яка містить широкий набір інструментів для обробки та перетворення даних. У цій панелі обираємо Conversion Tools, що містить інструменти для конвертації різних типів файлів.

Далі переходимо шляхом Conversion Tools => Excel => Table to Excel. Цей інструмент дозволяє конвертувати таблиці з ArcGIS у формат Excel, який зручно використовувати для подальшої роботи в інших середовищах. Після вибору цього інструменту відкриється нове вікно (рис. 2.7), де потрібно налаштувати параметри експорту [18].

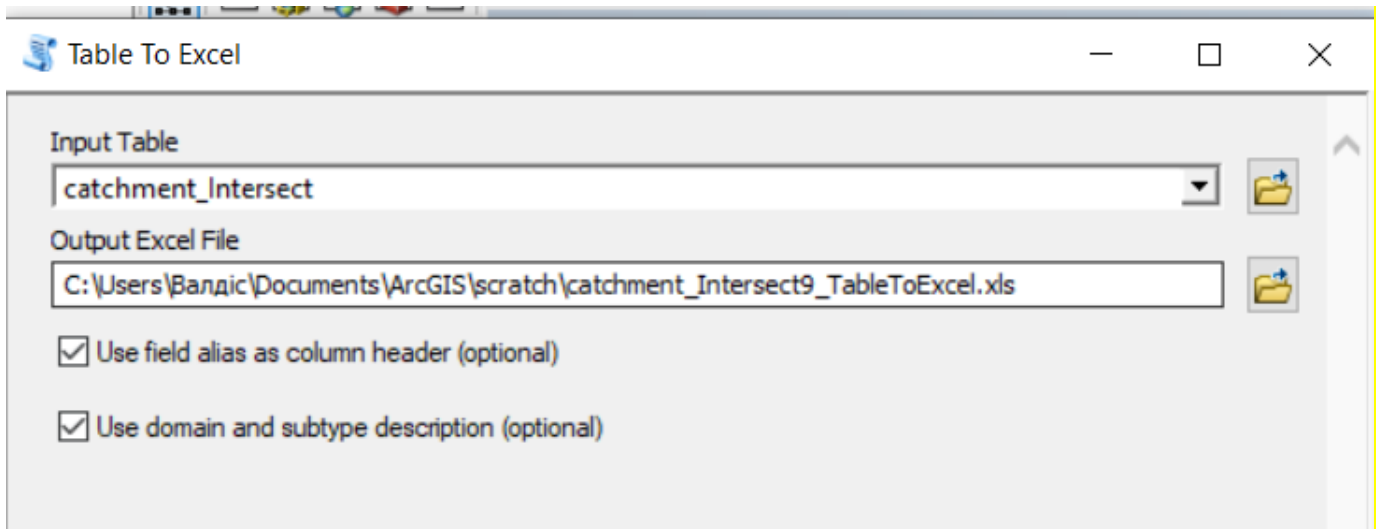


Рисунок 2.7 – Експорт даних в Excel шару catchment_Intersect

У новому вікні обираємо таблицю, яку бажаємо експортувати, та вказуємо шлях для збереження Excel-файлу на комп'ютері. Необхідно перевірити, щоб всі дані були коректно відображені в таблиці та відповідали очікуваному формату. Після налаштування параметрів натискаємо кнопку "ОК" для початку процесу конвертації.

Після завершення експорту, Excel-файл можна завантажити в Google Colab для подальшого аналізу. Для цього відкриваємо Google Colab і створюємо новий проект або відкриваємо існуючий. Використовуючи стандартні бібліотеки Python, такі як Pandas, завантажуюмо Excel-файл у середовище Colab і продовжуємо роботу з даними (рис. 2.8).

Цей процес забезпечує інтерактивну та ефективну роботу з даними у Google Colab, що є особливо корисним для дослідників і аналітиків, які працюють з великими наборами даних та потребують швидкого доступу до обчислювальних ресурсів. Завдяки інтеграції з Google Drive та можливості ділитися проектами з

іншими користувачами, Google Colab також сприяє спільній роботі та обміну знаннями між членами команди.

Google Colab

Unlitled1.ipymb

☆

🔧

Файл

Изменить

Вид

Вставка

Среда выполнения

Инструменты

Справка

Последнее сохранение: 17 мая

+

Код

+

Текст

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

Рисунок 2.8 – Перенесення даних між ArcGIS та Excel

Завдяки цьому можна проводити різноманітні операції аналізу, візуалізації та моделювання, використовуючи інструменти Python, що значно розширює можливості обробки геопросторових даних, отриманих з ArcGIS.

2.3 Аугментація даних в середовищі Google Colab

Для початку потрібно завантажити бібліотеку Pandas, яка є однією з найпотужніших і найпоширеніших бібліотек для роботи з даними в Python. Pandas надає інструменти для ефективного зчитування, обробки, аналізу та зберігання даних у зручному форматі.

Завдяки цій бібліотеці ми можемо легко маніпулювати нашими даними, виконувати різні операції та аналізи, а також підготувати дані для подальшої візуалізації чи моделювання.

Щоб завантажити бібліотеку Pandas і почати працювати з нашим отриманим датасетом, необхідно виконати наступні кроки (рис. 2.9).

```
[ ] import pandas as pd
```

```
[ ] df = pd.read_csv('catchment_Intersect9_TableToExcel.csv')
df
```

Рисунок 2.9 – Підключення Pandas для початку роботи

Після завантаження бібліотеки Pandas та імпортування даних у DataFrame, наступним кроком є проведення розвідувального аналізу стовпців таблиці. Цей етап є критично важливим для розуміння структури та змісту даних, що містяться в датасеті.

Розвідувальний аналіз стовпців таблиці дозволяє отримати загальне уявлення про кожен стовпець у датасеті. Цей процес включає отримання інформації про тип даних, які зберігаються у кожному стовпці, кількість ненульових значень та загальну кількість значень. Виконання цього аналізу допомагає виявити наявність пропущених значень, визначити, які стовпці містять числові дані, які – текстові або категоріальні, а також оцінити, чи потребують дані додаткової обробки.

Отримана інформація дозволяє:

1. Оцінити якість даних визначити наявність пропущених або некоректних значень, які можуть вимагати очищення або заповнення;
2. Розуміти структуру даних з'ясувати, які стовпці можна використовувати для подальшого аналізу та побудови моделей, а які можуть бути допоміжними або взагалі виключені з аналізу;
3. Планувати подальшу обробку визначити, які стовпці потребують перетворення типів даних, нормалізації або інших видів попередньої обробки.

Отримання детальної інформації про стовпці допомагає ефективно підготувати дані для подальшого аналізу та забезпечує, що всі важливі аспекти даних будуть враховані. Це є фундаментом для створення нового, очищеного та структурованого датасету, готового до подальших етапів аналізу або моделювання. (рис. 2.10).

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 335 entries, 0 to 334
Data columns (total 80 columns):
#   Column                Non-Null Count  Dtype
---  -
0   FID                    335 non-null    int64
1   FID_catchment          335 non-null    int64
2   SEM9                   335 non-null    object
3   SEM326                 335 non-null    object
4   FID_Метарічки         335 non-null    int64
5   SEM9.1                 335 non-null    object
6   SEM15                  335 non-null    object
7   SEM150                 335 non-null    object
8   SEM152                 335 non-null    int64
9   SEM327                 335 non-null    object
10  SEM157                 335 non-null    int64
11  SEM112                 335 non-null    object
12  SEM4                   335 non-null    int64
13  SEM7                   335 non-null    int64
14  SEM11                  335 non-null    int64
15  SEM20                  335 non-null    int64
16  SEM34                  335 non-null    int64
17  SEM35                  335 non-null    object
18  SEM36                  335 non-null    object
19  SEM340                 335 non-null    int64
20  SEM_32801              335 non-null    int64
21  GROUPSLAVE             335 non-null    int64
22  SEM_32804              335 non-null    object
23  SEM116                 335 non-null    object
24  SEM117                 335 non-null    object
25  SEM118                 335 non-null    object
26  DATECREATE             0 non-null      float64
27  TIMECREATE             335 non-null    object
28  AUTHORNAME             335 non-null    object
29  SEM32850               0 non-null      float64
30  SEM32851               335 non-null    object
31  SEM32852               335 non-null    object
32  DATECORR               0 non-null      float64
33  TIMECORR               335 non-null    object
34  AUTHRCORR              335 non-null    object
35  SEM32809               335 non-null    object
36  VIEWSCALE              335 non-null    int64
37  OBJCOLOR               335 non-null    object
38  FONTHEIGHT             335 non-null    int64
39  FONTNAME               335 non-null    object
40  LINETHICK              335 non-null    int64
41  THICKENDL              335 non-null    int64
42  COLWEIGHT              335 non-null    int64
43  SEM_32856              335 non-null    object
44  BackColor              335 non-null    object
45  ObjInfo                335 non-null    object
46  SEM_32858              335 non-null    object
47  SEM_32203              0 non-null      float64
48  SEM_32204              335 non-null    object
49  DateFrom               0 non-null      float64
50  ObjectCode             335 non-null    int64
51  ObjectKey              335 non-null    object
52  ObjectID               335 non-null    int64
```

Рисунок 2.10 – Детальна інформація про стовпці в датасеті

Після отримання загальної інформації про структуру та вміст таблиці, перехід до наступного етапу – аугментація даних – стає важливим кроком у дослідженні. Цей процес включає об'єднання та трансформацію існуючих стовпців для створення нових, що дозволяє отримати додаткову інформацію та покращити аналіз [19].

Для початку аугментації даних ми створимо нову копію датасету (рис. 2.11), щоб зберегти вихідні дані та внести зміни лише в копію. Новий датасет буде містити лише необхідні нам стовпці, які включають:

1. SEM326: цей стовпець містить код геоприв'язки в середовищі ArcGIS. Він дозволяє точно визначити географічні координати або розташування об'єкта на мапі;
2. SEM9.1: це назва річки, яка може бути важливою для аналізу водних об'єктів та їх впливу на досліджувану територію;
3. codename: тип населеного пункту, що може вказувати на характеристики та особливості місцевості;
4. nameua_new: назва населеного пункту, яка ідентифікує конкретний населений пункт;
5. adminua_2: район, в якому знаходиться населений пункт, що може вказувати на адміністративну підпорядкованість;
6. codename_3: територіальна громада, яка може бути важливою для вивчення територіальної організації та взаємодії між населеними пунктами.

Ці стовпці мають значну інформаційну цінність і можуть бути використані для подальшого аналізу та моделювання в рамках дослідження.



```
columns_to_copy = ["SEM326", "SEM9.1", "adminua_1", "codename", "nameua_new", "adminua_2", "codename_3", "adminua_3"]
new_df = df[columns_to_copy].copy()

new_df['codename'] = new_df['codename'].replace(
    ['селище', 'село', 'місто', 'селища міського типу'],
    ['с-ще', 'с', 'м', 'сmt']
)
```

Рисунок 2.11 – Фрагмент коду копіювання та заміни даних

Для створення стовпця code залишимо стовпець SEM326, що містять відповідні дані, а для geo_names потрібно застосувати певні трансформації з існуючих стовпців.

Стовпець codename для більш візуального представлення необхідно зробити заміну даних, використовуючи функцію replace, що зображено на фрагменті коду (рис. 2.11). Кожен тип населеного пункту ('с-ще', 'с', 'м', 'сmt') був окремо позначений у датасеті. Це може включати зміну форматування або об'єднання даних для

покращення читабельності та візуального сприйняття. Наприклад, можна замінити кодові назви на більш інформативні або зрозумілі тексти.

Після завершення аналізу даних, наступним кроком є об'єднання стовпців `codename`, `nameua_new`, `adminua_3` та `codename_3`. Це може бути зроблено шляхом застосування функції `apply`, яка дозволяє об'єднати значення цих чотирьох стовпців у новий стовець. Отримання такого об'єднаного стовпця спростить аналіз даних, оскільки всі необхідні дані будуть представлені в одному місці, що полегшить сприйняття та розуміння.

Додавання артикля «р» (річка) до стовпця `SEM9.1` можна здійснити за допомогою простої функції `apply`. Це додасть необхідний префікс до кожного значення у стовпці, що допоможе у легкому розрізненні даних про річки у таблиці.

Після об'єднання стовпців та додавання артикля, старі стовпці (`codename`, `nameua_new`, `adminua_3`, `codename_3`) можна видалити, оскільки вони більше не мають потреби. Це можна зробити за допомогою функції `drop`. Після цих операцій отримаємо таблицю, яка буде зручною для подальшого аналізу та обробки даних (рис. 2.12).

```
# Об'єднання стовпців
new_df['codenames'] = new_df.apply(lambda row: row['codename'] + ' ' + row['nameua_new'], axis=1)
# Об'єднання стовпців
new_df['Miskrada'] = new_df.apply(lambda row: row['adminua_3'] + ' ' + row['codename_3'], axis=1)
# Видалення оригіналів
new_df.drop(['adminua_3', 'codename_3'], axis=1, inplace=True)
new_df.drop(['codename', 'nameua_new'], axis=1, inplace=True)

new_df['SEM9.1'] = new_df['SEM9.1'].apply(lambda x: 'р. ' + x)
```

	SEM326	SEM9.1	adminua_1	adminua_2	codenames	Miskrada
0	UA_M5.4_0254	р. Без назви (с. Зарічне)	Вінницька	Тульчинський район	с. Гуральня	Тульчинська міська територіальна громада
1	UA_M5.4_0254	р. Без назви (с. Зарічне)	Вінницька	Тульчинський район	с. Шура-Копіївська	Тульчинська міська територіальна громада
2	UA_M5.4_0272	р. Самець (р. Сільниця)	Вінницька	Тульчинський район	с. Кришинці	Тульчинська міська територіальна громада
3	UA_M5.4_0272	р. Самець (р. Сільниця)	Вінницька	Тульчинський район	с. Тарасівка	Тульчинська міська територіальна громада
4	UA_M5.4_0273	р. Самець (р. Сільниця)	Вінницька	Тульчинський район	с. Холодівка	Тульчинська міська територіальна громада

Рисунок 2.12 – Об'єднання та видалення стовпців

Для подальшого аналізу необхідно провести групування даних за стовпцем `SEM326`. Під час цього процесу ми використовуємо функцію `join` для об'єднання значень стовпця `codenames` у один масив для кожної групи. Це дозволяє нам згрупувати всі відповідні дані разом у відповідних групах.

Для виконання цього завдання використовуємо метод `groupby` разом із функцією `agg`, яка дозволяє використовувати `join` для об'єднання значень в один масив. Цей процес допомагає створити новий стовпець, де дані будуть згруповані разом за значенням стовпця `SEM326`.

Після завершення групування та застосування функції `join` ми перевіряємо результат, щоб переконатися, що операція була виконана правильно. Це дозволяє нам впевнитися, що дані коректно згруповані та об'єднані відповідно до наших очікувань і вимог. Такий підхід до обробки даних дозволяє забезпечити точність та надійність результатів аналізу (рис. 2.13).



```
grouped_df = new_df.groupby(['SEM326']).agg({
    'codenames': lambda x: ', '.join(sorted(set(x))),
    'SEM9.1': 'first',          # Перший елемент у групі
    'adminua_2': 'first',       # Перший елемент у групі
    'Miskrada': 'first'         # Перший елемент у групі
}).reset_index()
grouped_df
```

	SEM326	codenames	SEM9.1	adminua_2	Miskrada
0	UA_M5.4_0010	с. Березна, с. Чудинівці	р. Пожарка	Хмільницький район	Хмільницька міська територіальна громада
1	UA_M5.4_0011	м. Хмільник, с. Вербівка, с. Голодьки, с. Кожу...	р. Струмок Лелія	Хмільницький район	Хмільницька міська територіальна громада
2	UA_M5.4_0013	СМТ. Стрижавка, м. Вінниця, с. Гушинці, с. Кал...	р. Бобрка	Хмільницький район	Іванівська сільська територіальна громада
3	UA_M5.4_0014	СМТ. Сутиски, с. Лани, с. Лука-Мелешківська, с...	р. Студениця	Вінницький район	Лука-Мелешківська сільська територіальна громада
4	UA_M5.4_0015	СМТ. Сутиски, СМТ. Тиврів, с. Кліщів, с. Потуш...	р. Краснянка	Тульчинський район	Шпиківська селищна територіальна громада
...	...	...	...	...	...
135	UA_M5.4_0269	с. Клебань, с. Федьківка	р. Кільтява	Тульчинський район	Тульчинська міська територіальна громада
136	UA_M5.4_0270	с. Ганнопіль, с. Пестеля, с. Суворовське	р. Кільтява	Тульчинський район	Тульчинська міська територіальна громада
137	UA_M5.4_0272	с. Кришинці, с. Тарасівка	р. Самець (р.Сільниця)	Тульчинський район	Тульчинська міська територіальна громада
138	UA_M5.4_0273	с. Холодівка	р. Самець (р.Сільниця)	Тульчинський район	Тульчинська міська територіальна громада
139	UA_M5.4_0274	с. Клебань	р. Самець (р.Сільниця)	Тульчинський район	Тульчинська міська територіальна громада

140 rows x 5 columns

Рисунок 2.13 – Групування даних

Завершальним етапом процесу аугментації даних є об'єднання всіх необхідних стовпців для створення нових стовпців `code` та `geo_names`, а також видалення оригінальних стовпців. Ця операція дозволить створити фінальний датасет, який буде містити всю необхідну інформацію в компактному та зручному вигляді (рис. 2.14).

Створення нових стовпців `code` та `geo_names` полягає в об'єднанні відповідних стовпців даних у єдину інформаційну одиницю. Наприклад, створення стовпця `code` може включати у себе поєднання кодів геоприв'язки та типу населеного пункту, тоді як створення стовпця `geo_names` може включати назви населених пунктів та їх

географічну прив'язку. Це допомагає зберігати зв'язані дані разом та полегшує їх подальший аналіз.

Після створення нових стовпців необхідно видалити оригінальні стовпці, які були використані для їх створення. Це дозволяє зменшити розмір датасету і полегшує подальшу обробку та аналіз даних.

Завершивши ці кроки, ми отримаємо фінальний датасет, який буде містити всю необхідну інформацію в компактному та зручному форматі для подальшого використання. Такий датасет буде ідеальним інструментом для проведення подальших досліджень та аналізу даних.

```
grouped_df['geo_names'] = grouped_df.apply(lambda row: row['SEM9.1'] + ', ' + row['codenames']
                                           + ', ' + row['adminua_2'] + ', ' + row['Miskrada'], axis=1)
grouped_df.drop(['SEM9.1', 'codenames', 'adminua_2', 'Miskrada'], axis=1, inplace=True)
grouped_df
```

	SEM326	geo_names
0	UA_M5.4_0010	р. Пожарка, с. Березна, с. Чудинівці, Хмільниц...
1	UA_M5.4_0011	р. Струмок Лелія, м. Хмільник, с. Вербівка, с....
2	UA_M5.4_0013	р. Бобрка, СМТ. Стрижавка, м. Вінниця, с. Гущи...
3	UA_M5.4_0014	р. Студениця, СМТ. Сутиски, с. Лани, с. Лука-М...
4	UA_M5.4_0015	р. Краснянка, СМТ. Сутиски, СМТ. Тиврів, с. Кл...
...	...	...
135	UA_M5.4_0269	р. Кільтява, с. Клебань, с. Федьківка, Тульчин...
136	UA_M5.4_0270	р. Кільтява, с. Ганнопіль, с. Пестеля, с. Суво...
137	UA_M5.4_0272	р. Сامةць (р.Сільниця), с. Кришинці, с. Тарасі...
138	UA_M5.4_0273	р. Сامةць (р.Сільниця), с. Холодівка, Тульчинс...
139	UA_M5.4_0274	р. Сامةць (р.Сільниця), с. Клебань, Тульчинськ...

140 rows × 2 columns

Рисунок 2.14 – Фінальний датасет

Таким чином, завершивши аугментацію даних, ми отримуємо новий, більш інформативний датасет, готовий для подальшого аналізу та використання у вирішенні поставлених задач. Зберігаємо за допомогою метода `to_csv` для подальшої можливої роботи в майбутньому.

2.4 Висновки

У другому розділі розглянуто методологію та інструменти для аналізу даних з метою розв'язання поставленої задачі.

Використання середовища ArcGIS дозволило створити водозбірні площі, що забезпечило точне визначення меж басейнів та їхніх характеристик. Цей інструмент виявився ефективним для картографування та візуалізації геопросторових даних.

Інтеграція даних між ArcGIS та Python забезпечила можливість проведення комплексного аналізу. Дані, отримані в ArcGIS, були оброблені за допомогою Python, що дозволило ефективно аналізувати взаємозв'язки між гідрологічними та екологічними параметрами.

Аугментація даних в середовищі Google Colab надала додаткові можливості для обробки великих обсягів інформації. Це середовище дозволило застосувати складні алгоритми та методи машинного навчання для поглибленого аналізу геопросторових даних.

Загалом, другий розділ демонструє ефективне використання сучасних інструментів та методів для аналізу даних, що є ключовим для розв'язання поставленої задачі.

3. АНАЛІЗ І ВІЗУАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ВИЗНАЧЕННЯ ГЕОПРОСТОРОВИХ ЗВ'ЯЗКІВ МАСИВІВ ВОД

3.1 Створення графіків в середовищі Google Colab для візуалізації даних

Для досягнення більш виразного та візуального уявлення про результати нашого дослідження, ми використали середовище Google Colab для створення графіків. Ці графіки дозволили нам не лише представити дані у зрозумілій формі, але й сприяли подальшому аналізу водозбірних площ та їх взаємодії з населеними пунктами, що перетинаються з ними.

Використання візуалізацій дозволило краще розуміти просторові зв'язки між водозбірними площами та населеними пунктами, а також виявити вплив цих водозбірних областей на різні аспекти життя та розвитку населення. Такий підхід сприяє більш глибокому аналізу та врахуванню факторів, що впливають на розвиток регіонів [20].

Використовуючи Python та його потужні бібліотеки для візуалізації, такі як Matplotlib, Pandas і Seaborn, було створено різноманітні графіки для ілюстрації даних.

Для початку було завантажено датасет та необхідні бібліотеки для подальшої роботи з даними (рис. 3.1).

```
[ ] import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
#Завантаження та огляд даних
df = pd.read_csv('catchment_Intersect9_TableToExcel.csv')
df
```

	FID	FID_catchment	SEM9	SEM326	FID_Метарічки	SEM9.1	SEM15	SEM150	SEM152	SEM327	...	type_1	codename_1	adminua_1	adminru_1	adminen_1	koatuu_1	nameua_ne
0	1	155	UA_M5.4_0254	48	Без назви (с. Зарічне)	МЕНЕЕ 20 М			0		...	Області, автономії та ради	Межі областей та автономні	Вінницька	Винницкая	Vinnytska	500000000	Гуральн
1	2	155	UA_M5.4_0254	48	Без назви (с. Зарічне)	МЕНЕЕ 20 М			0		...	Області, автономії та ради	Межі областей та автономні	Вінницька	Винницкая	Vinnytska	500000000	Шура Копітвськ
2	3	169	UA_M5.4_0272	49	Самець (р.Сільниця)	МЕНЕЕ 20 М	277		0		...	Області, автономії та ради	Межі областей та автономні	Вінницька	Винницкая	Vinnytska	500000000	Кришчин
3	4	169	UA_M5.4_0272	49	Самець (р.Сільниця)	МЕНЕЕ 20 М	277		0		...	Області, автономії та ради	Межі областей та автономні	Вінницька	Винницкая	Vinnytska	500000000	Тарасівк
					Самець	МЕНЕЕ						Області,	Межі					

Рисунок 3.1 – Завантаження датасету

Для підрахунку кількості перетинів річок з різними типами населених пунктів (codename) та відображення цих даних на графіку типу barplot, ми можемо використати бібліотеки Python, такі як Pandas, Matplotlib або Seaborn (рис. 3.2).

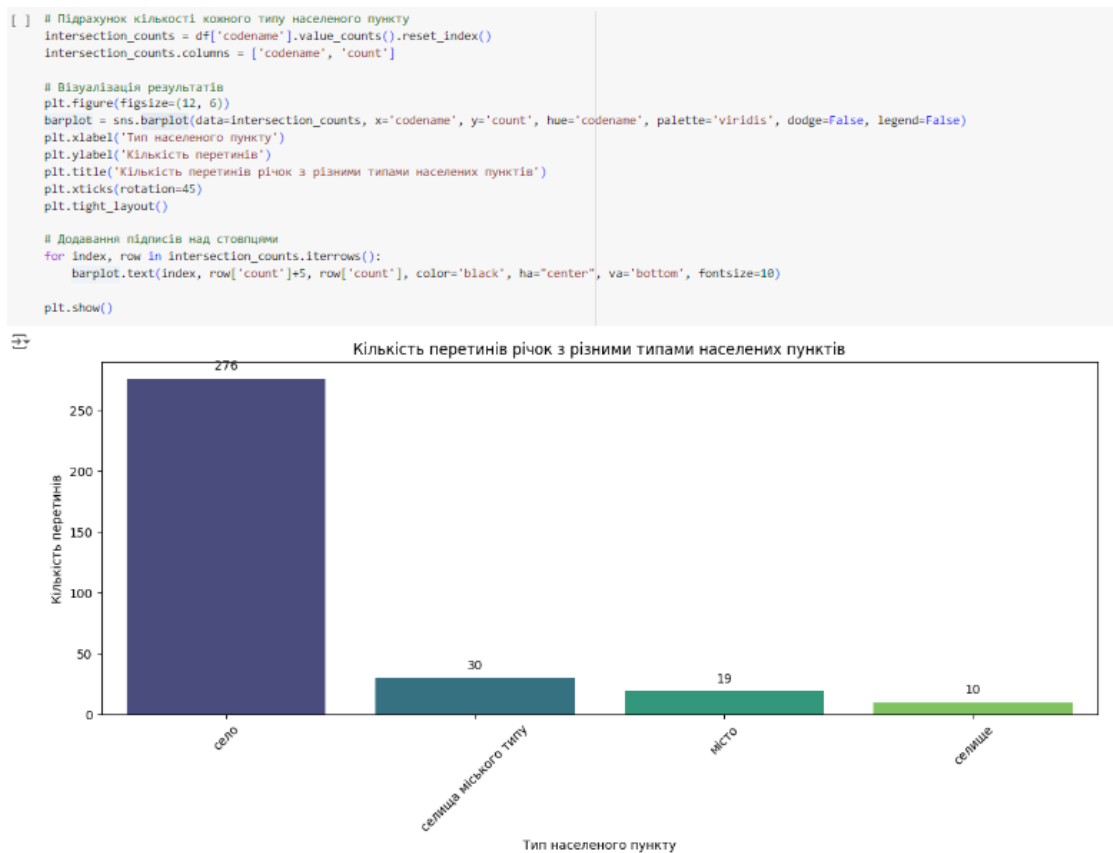


Рисунок 3.2 – Графік за кількістю населених пунктів за типами

Стовпець "population" представляє числові дані, які мають значний інформаційний потенціал, що розкриває широкі можливості для глибшого аналізу демографічних характеристик вивчених територій. Існують різні способи використання цих даних, які дозволяють розуміти розподіл населення за типами населених пунктів, такими як села, містечка, міста та селища міського типу.

Побудова різноманітних графіків, що візуалізують цей розподіл, допомагає наочно представити, як населення розподіляється між цими категоріями, виявити закономірності та аномалії. За допомогою даних зі стовпця "population" можна обчислити загальне населення для кожного типу населеного пункту, що дозволяє з'ясувати, які типи населених пунктів найбільше впливають на загальну демографічну картину регіону.

Такий аналіз дозволяє виявити потенційні проблеми, такі як перенаселеність або недовикористання певних територій, що може служити основою для прийняття рішень щодо розвитку регіональної політики. Важливою частиною такого аналізу є планування стратегій для сталого розвитку регіонів, що базується на збалансованому розподілі ресурсів та покращенні умов життя населення.

Отже, стовпець "population" не лише набір чисел, а важливе джерело інформації, яке сприяє більш ефективному управлінню територіями та ресурсами, а також сприяє розвитку стратегій для сталого розвитку регіонів (рис. 3.3).



Рисунок 3.3 – Графік за кількістю населених пунктів за типами

На графіку чітко видно, що населення розподіляється нерівномірно між різними типами населених пунктів. Зокрема, селища (с-ще) мають найменшу кількість населення. У той час як найбільша кількість населення проживає у містах, що складає 2 849 360 осіб. Це демонструє значну концентрацію населення у міських районах порівняно з селищами та іншими типами населених пунктів.

Для розподілу за семантикою SEM326, яка містить код геоприв'язки в середовищі ArcGIS, ми можемо використати графік для візуалізації кількості записів для кожного коду.

Це допоможе нам зрозуміти розподіл даних за географічними областями або регіонами (рис. 3.4).

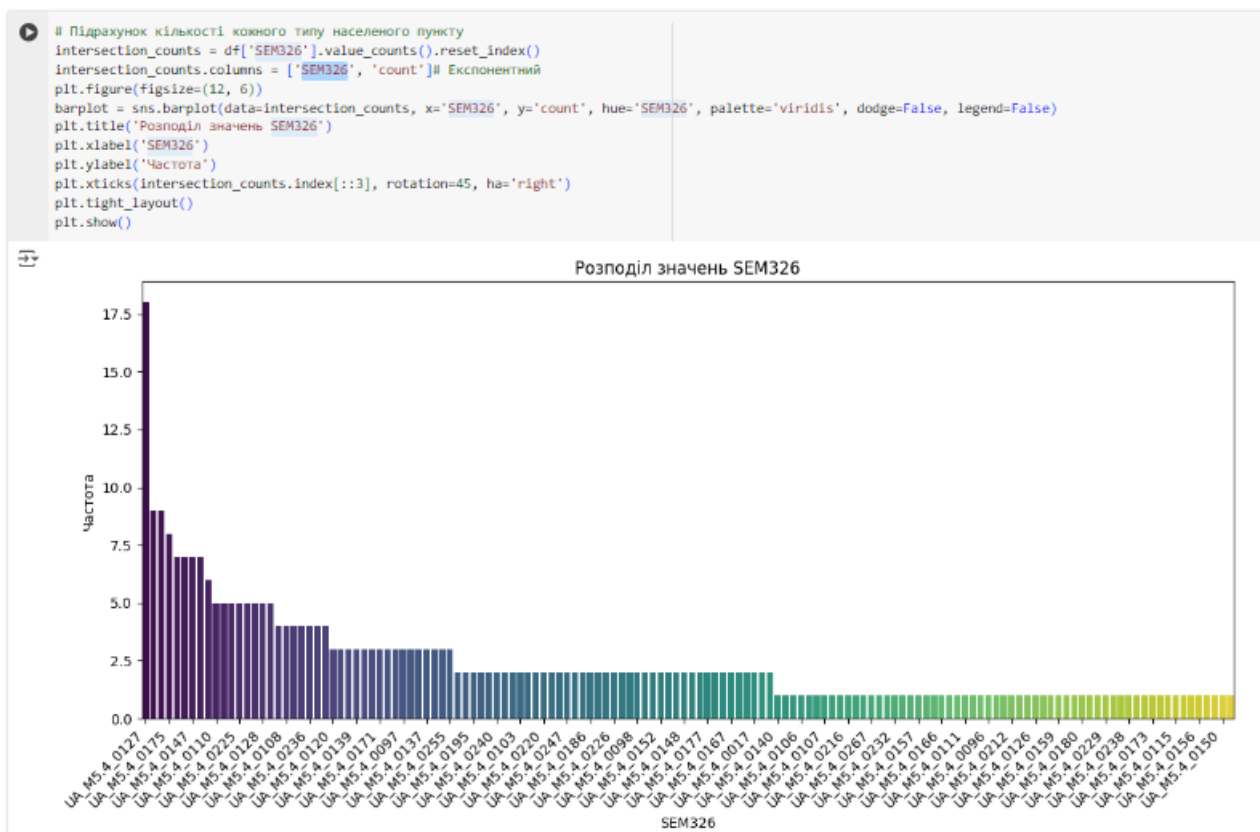


Рисунок 3.4 – Графік розподілу даних за географічними областями або регіонами

Тепер побудуємо графік розподілу за кодом геоприв'язки та типом населеного пункту, які включають смт (селище міського типу). Цей графік допоможе нам візуалізувати, як розподіляється географічна прив'язка для різних типів населених пунктів і виявити можливі взаємозв'язки між ними.

Тепер перейдемо до побудови графіка розподілу за кодом геоприв'язки та типом населеного пункту, включаючи смт (селище міського типу). Для цього ми скопіюємо потрібні стовпці SEM326 та type, що містить тип населеного пункту, використовуючи функцію сору(). Потім згрупуємо дані за кодом геоприв'язки та типом населеного пункту за допомогою функції groupby, і виконаємо підрахунок кількості записів за кожною групою з допомогою функції count(). Нарешті, побудуємо графік типу bar для візуалізації цього розподілу.

Цей аналіз допоможе нам краще зрозуміти, як географічна прив'язка різних типів населених пунктів розподілена у вибірці та виявити можливі залежності між ними (рис. 3.5).

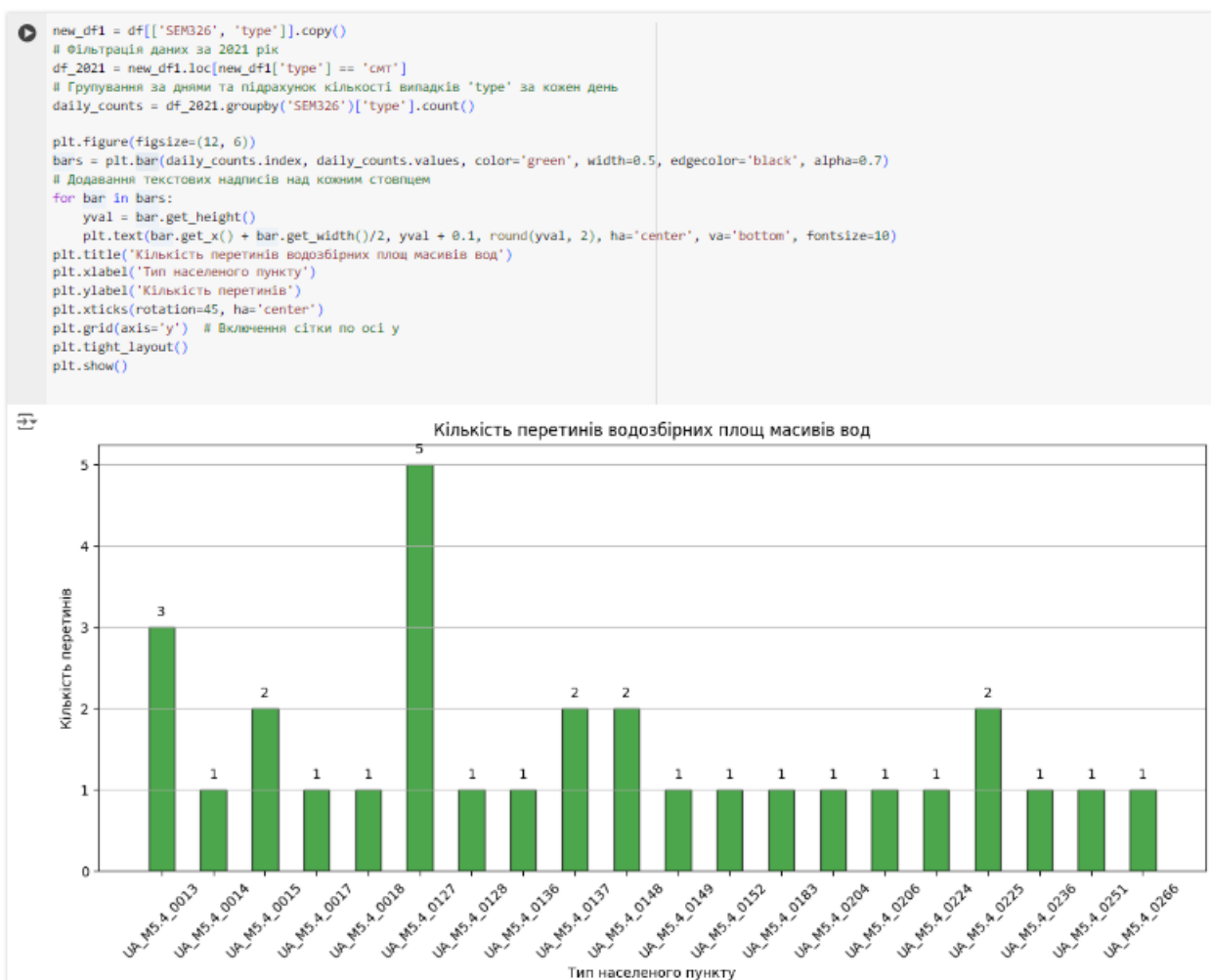


Рисунок 3.5 – Графік розподілу даних геоприв'язки по СМТ

Графік розподілу кількості перетинів водозбірних площ масивів вод в залежності від SEM326 геоприв'язки за типом населеного пункту дозволить нам

візуально порівняти, як часто водозбірні площі перетинаються з різними типами населених пунктів.

Це допоможе нам зрозуміти, як розподіляється водний ресурс в залежності від географічної локації та типу населеного пункту, а також можливих впливів цих факторів на водозбір та його управління.

Для цього використаємо графік `histplot`, що дозволить нам побудувати гістограму розподілу кількості перетинів водозбірних площ масивів вод за типом населеного пункту в залежності від SEM326 геоприв'язки. Цей тип графіку дозволяє швидко візуалізувати розподіл даних і виявити можливі тенденції та взаємозв'язки (рис. 3.6).

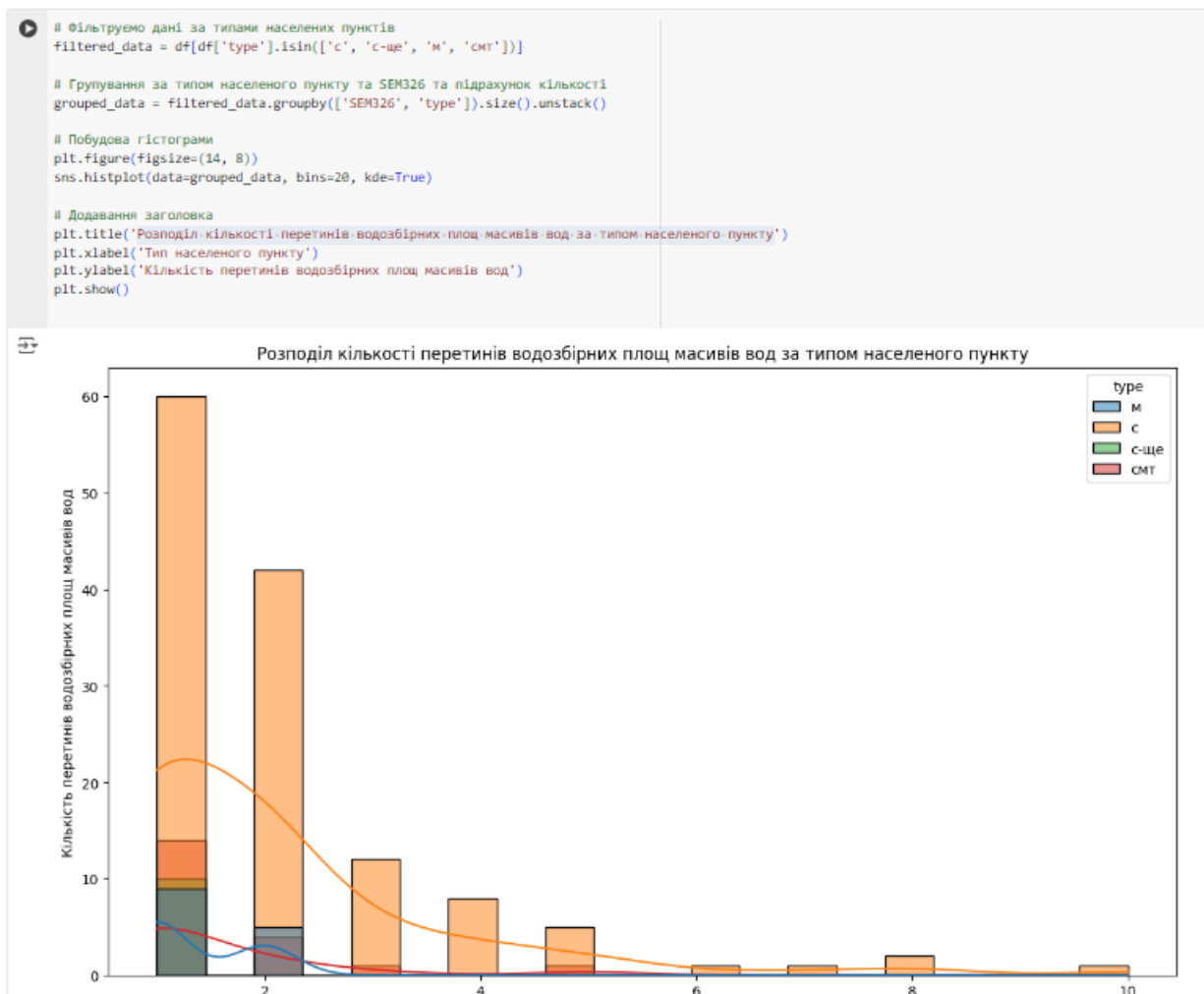


Рисунок 3.6 – Графік розподілу даних за геоприв'язкою та типом населеного пункту

Тепер ми побудуємо графік повного розподілу за геоприв'язкою та типом населеного пункту з підрахованою кількістю населених пунктів, які будуть

позначені над стовпцями, та використаємо різні кольори для візуального виділення. Це допоможе нам краще розглянути розподіл даних і виявити можливі взаємозв'язки між географічною прив'язкою та типом населеного пункту.

Для цього скористаємося функцією `isin()`, яка дозволить нам розділити дані зі стовпця на чотири різні групи: 'с', 'с-ще', 'м' та 'сmt'. Потім згрупуємо дані за геоприв'язкою SEM326 та типом населеного пункту `type` та підрахуємо кількість населених пунктів у кожній групі за допомогою функції `size()` (рис. 3.7).

```
# Фільтруємо дані за типами населених пунктів
filtered_data = df[df['type'].isin(['с', 'с-ще', 'м', 'сmt'])]

# Групування за типом населеного пункту та SEM326 та підрахунок кількості
grouped_data = filtered_data.groupby(['SEM326', 'type']).size().unstack(fill_value=0)

# Побудова стовпчастої діаграми
fig, ax = plt.subplots(figsize=(14, 8))

# Побудова стовпчиків з різними кольорами для кожного типу населеного пункту
colors = ['skyblue', 'lightgreen', 'salmon', 'orange']
bars = grouped_data.plot(kind='bar', stacked=True, ax=ax, color=colors, edgecolor='black')

# Додавання текстових надписів над кожним стовпчиком
for container in ax.containers:
    ax.bar_label(container)

# Додавання заголовка та підписів до осей
plt.title('Розподіл кількості перетинів водозбірних площ масивів вод за типом населеного пункту')
plt.xlabel('SEM326')
plt.ylabel('Кількість перетинів водозбірних площ масивів вод')

# Додавання легенди
plt.legend(title='Тип населеного пункту', labels=['с', 'с-ще', 'м', 'сmt'])

# Встановлення міток осі X
xticks = range(len(grouped_data.index))
ax.set_xticks(xticks)
ax.set_xticklabels(grouped_data.index, rotation=45, ha='right')

# Вибірково відображення міток осі X
for label in ax.get_xticklabels()[::2]:
    label.set_visible(True)
for label in ax.get_xticklabels()[1::2]:
    label.set_visible(False)

# Відображення графіку
plt.tight_layout()
plt.show()
```

Рисунок 3.7 – Фрагмент коду графіку

Щоб відобразити кількість населених пунктів у надписах, ми використовуємо цикл за допомогою функції `for` та методу `bar_label`. Цей підхід дозволяє динамічно додавати надписи з кількістю населених пунктів на кожен стовпець графіка, що покращує візуальне сприйняття та розуміння даних.

Далі, в параметри легенди функції "legend" було додано назву графіку та легенди, використовуючи аргумент labels, які відповідають кожній з чотирьох категорій: 'с', 'с-ще', 'м', 'сmt'. Це дозволяє відображати пояснення до кожного типу населеного пункту на графіку і полегшує розуміння даних.

Як результат, ми отримаємо графік з розподілом кількості населених пунктів за геоприв'язкою та типом населеного пункту. На графіку будуть представлені чотири категорії: 'с', 'с-ще', 'м' та 'сmt', кожна з яких буде мати своїм кольором і відповідним підписом у легенді. Крім того, над кожним стовпчиком буде вказана кількість населених пунктів у відповідній категорії (рис. 3.8).

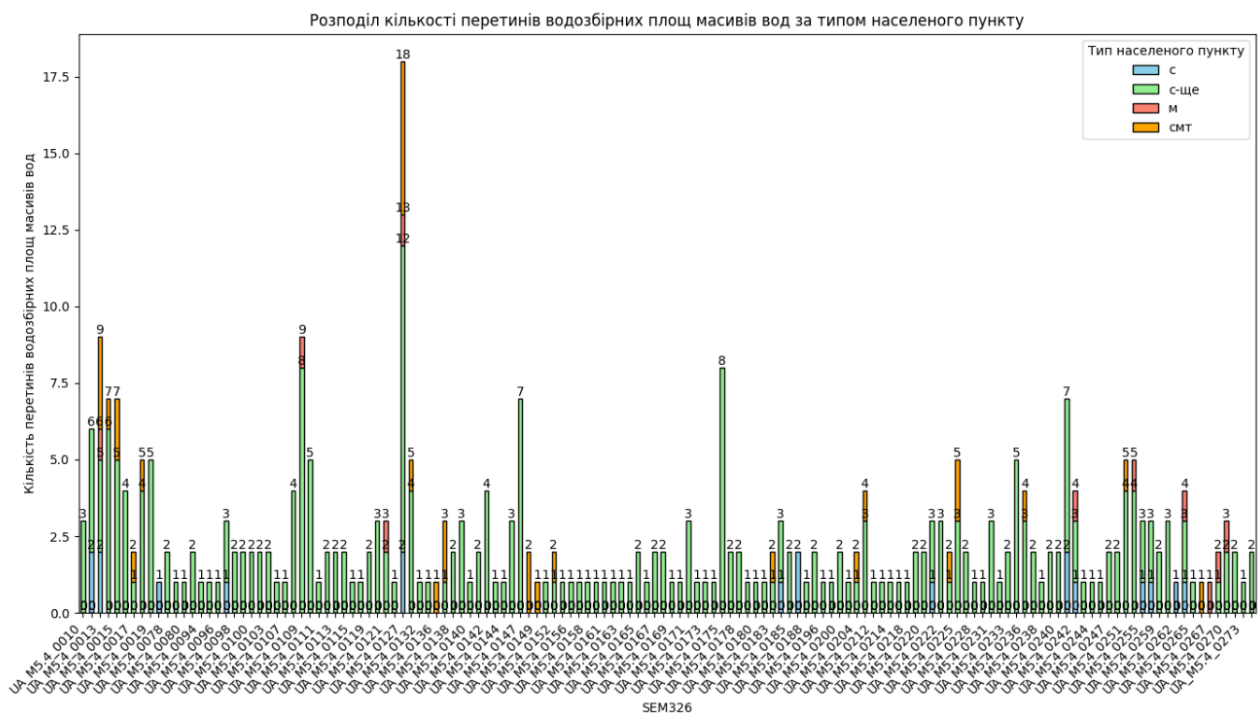


Рисунок 3.8 – Графік розподілу кількості перетинів водозбірних площ масивів вод за типом населеного пункту

За допомогою графіків ми отримали уявлення про розподіл кількості населених пунктів за географічною прив'язкою та типом населеного пункту. Було видно, що наявність населених пунктів різних типів варіюється залежно від геоприв'язки, що вказує на різноманітність географічних умов та характеристик територій.

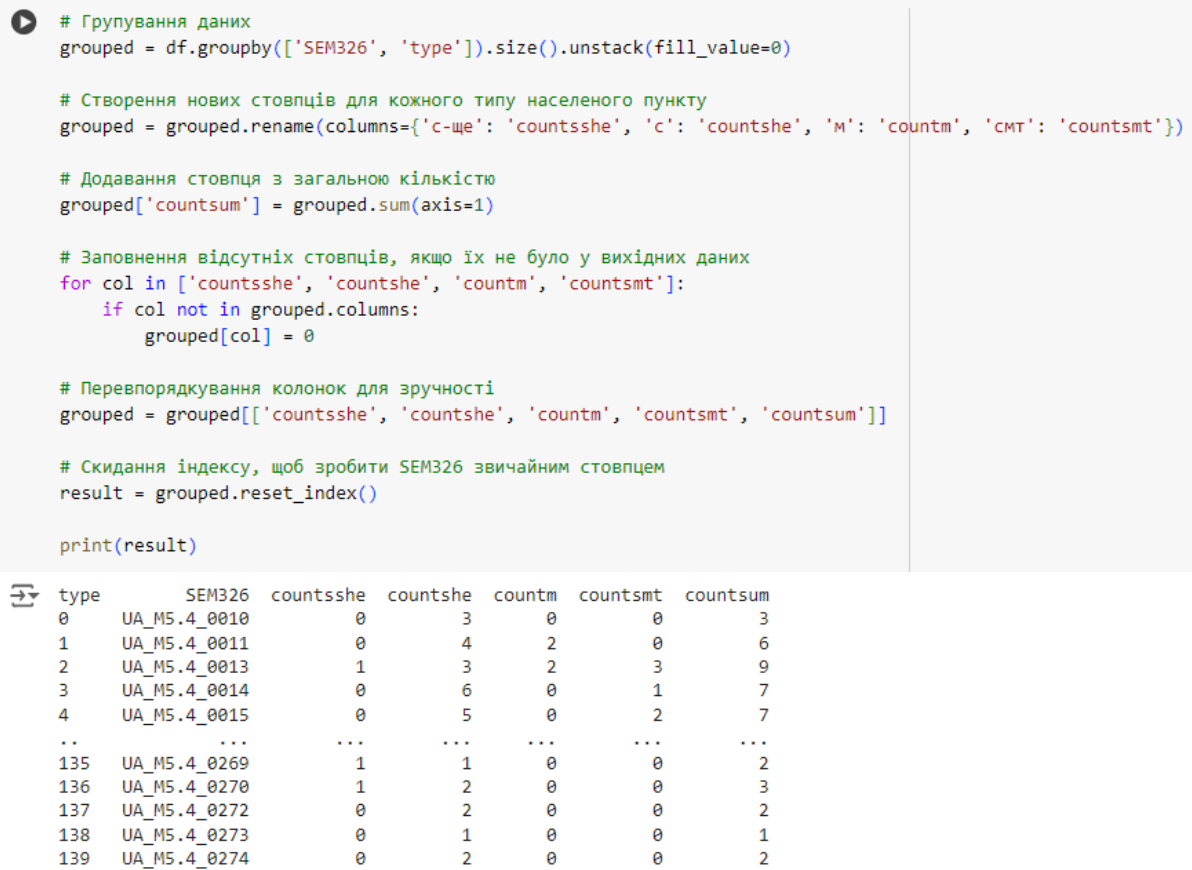
Крім того, графіки також дозволили нам виявити можливі залежності та тенденції в розподілі населених пунктів, такі як кількість населених пунктів у різних географічних областях або співвідношення між різними типами населених пунктів.

У загальному, візуалізація даних за допомогою графіків сприяє кращому розумінню розподілу та характеристик населених пунктів, що може бути корисним для подальшого аналізу та прийняття рішень в галузі географії, управління територіальними ресурсами, та іншим суміжним сферам.

3.2 Створення тематичної карти засобами ArcGIS та Python

На основі таблиці водозбірних площ ми створимо ще одну таблицю, яка буде включати в себе загальну кількість перетинів з населеними пунктами, а також окремо за типами населених пунктів: сільські ради (с-ще), села (с), міста (м) та селища міського типу (сmt).

Для цього ми скористаємося методом групування даних за типом населеного пункту і обчислимо кількість перетинів для кожного типу. Результатом буде нова таблиця, яка буде включати загальну кількість перетинів та розподіл їх за типами населених пунктів (рис. 3.9).



```

# Групування даних
grouped = df.groupby(['SEM326', 'type']).size().unstack(fill_value=0)

# Створення нових стовпців для кожного типу населеного пункту
grouped = grouped.rename(columns={'с-ще': 'countsshe', 'с': 'countshe', 'м': 'countm', 'сmt': 'countsm'})

# Додавання стовпця з загальною кількістю
grouped['countsum'] = grouped.sum(axis=1)

# Заповнення відсутніх стовпців, якщо їх не було у вихідних даних
for col in ['countsshe', 'countshe', 'countm', 'countsm']:
    if col not in grouped.columns:
        grouped[col] = 0

# Перевпорядкування колонок для зручності
grouped = grouped[['countsshe', 'countshe', 'countm', 'countsm', 'countsum']]

# Скидання індексу, щоб зробити SEM326 звичайним стовпцем
result = grouped.reset_index()

print(result)

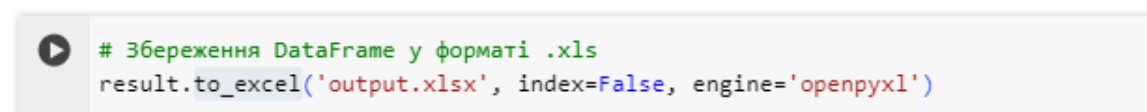
```

	type	SEM326	countsshe	countshe	countm	countsm	countsum
0	UA_MS.4_0010		0	3	0	0	3
1	UA_MS.4_0011		0	4	2	0	6
2	UA_MS.4_0013		1	3	2	3	9
3	UA_MS.4_0014		0	6	0	1	7
4	UA_MS.4_0015		0	5	0	2	7
...	...		...	...	...	...	...
135	UA_MS.4_0269		1	1	0	0	2
136	UA_MS.4_0270		1	2	0	0	3
137	UA_MS.4_0272		0	2	0	0	2
138	UA_MS.4_0273		0	1	0	0	1
139	UA_MS.4_0274		0	2	0	0	2

Рисунок 3.9 – Фрагмент коду та таблиці.

Після створення нової таблиці з розподілом перетинів за типами населених пунктів ми можемо зберегти її у відповідному форматі за допомогою функції `to_excel()`.

Ця функція дозволяє зберегти дані у форматі Excel, що часто використовується для подальшого аналізу чи обробки в інших програмах. Вона приймає шлях до файлу, в який ми хочемо зберегти дані, та інші необхідні параметри, такі як назва аркуша та формат даних. Після виконання цієї операції ми матимемо збережену таблицю з розподілом перетинів за типами населених пунктів у форматі Excel (рис. 3.10).



```

# Збереження DataFrame у форматі .xls
result.to_excel('output.xlsx', index=False, engine='openpyxl')

```

Рисунок 3.10 – Фрагмент коду зберігання таблиці

Далі перейшовши в середовище ArcGIS необхідно виконати команду Join на водозбірних площах масивів (рис. 3.11).

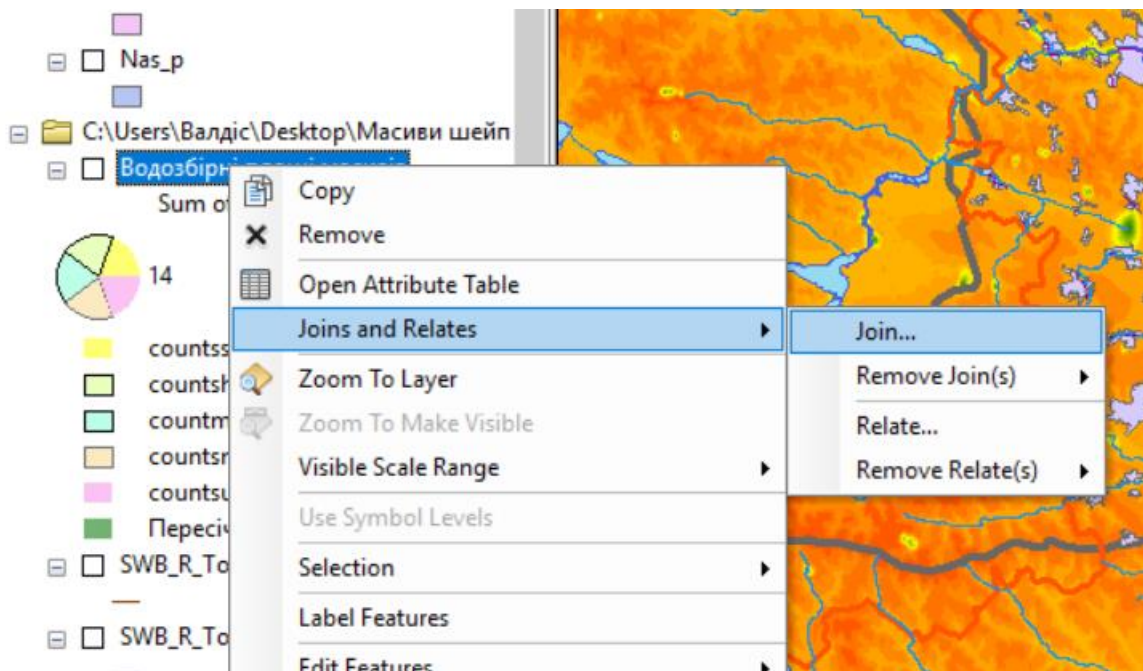


Рисунок 3.11 – Під'єднання таблиці в ArcGIS

В вікні яке з'явиться необхідно вибрати шлях до файлу таблиці яку ми зробили і далі зв'язати поля SEM326 і code щоб програма розуміла, що це геоприв'язка. Після цього програма зможе правильно ідентифікувати та інтерпретувати ці дані як геоприв'язку при подальшому використанні або аналізі (рис. 3.12).

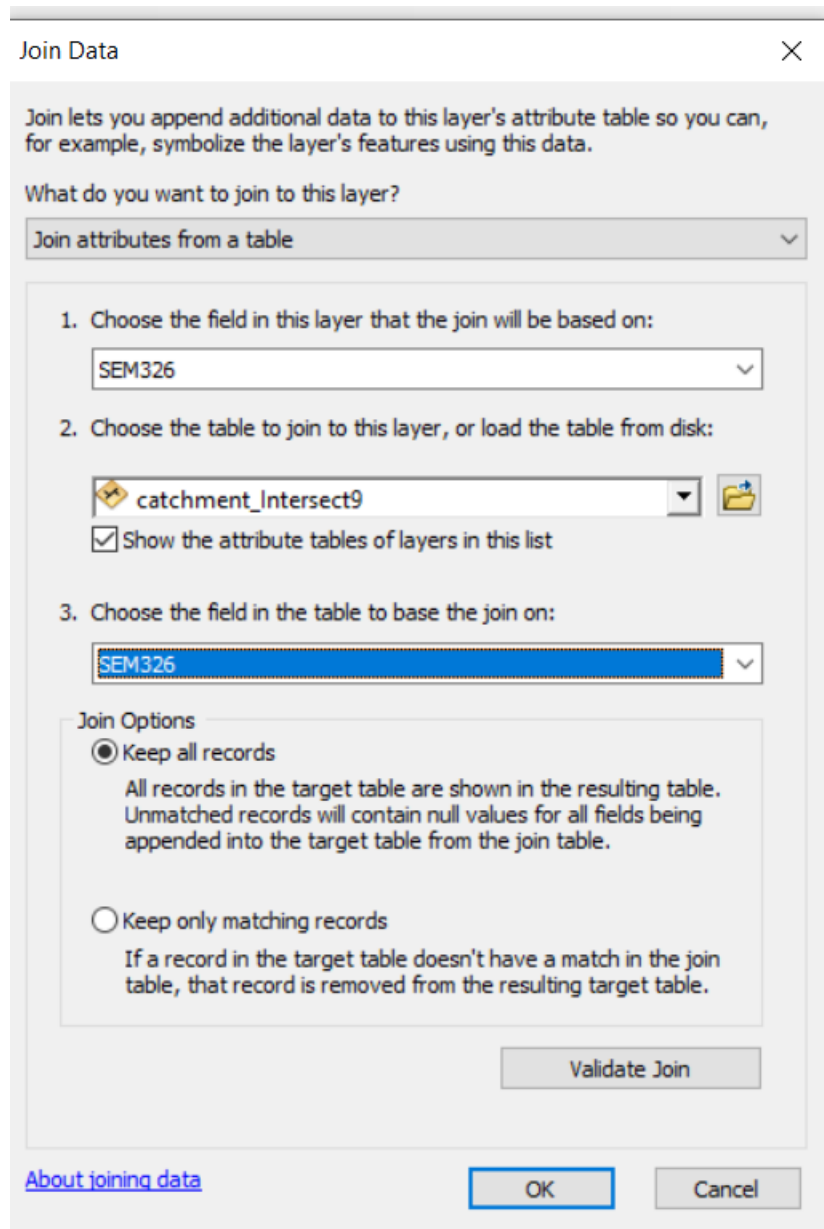


Рисунок 3.12 – Вікно імпорту даних

Після завершення всіх вищезазначених дій ми маємо можливість створити тематичну карту, що створюється наступним чином. Потрібно перейти в меню "properties" і вибрати вкладку "Symbology". Це дозволяє налаштувати символіку та візуальні ефекти для наших даних. У вкладці "Symbology" ми можемо вибрати тип символу, кольори, прозорість, масштаб та інші параметри, які хочемо використовувати для відображення нашої тематичної карти. Після вибору необхідних налаштувань ми можемо побачити попередній перегляд карти і, якщо все виглядає правильно, застосувати налаштування та зберегти тематичну карту для подальшого використання або аналізу (рис. 3.13).

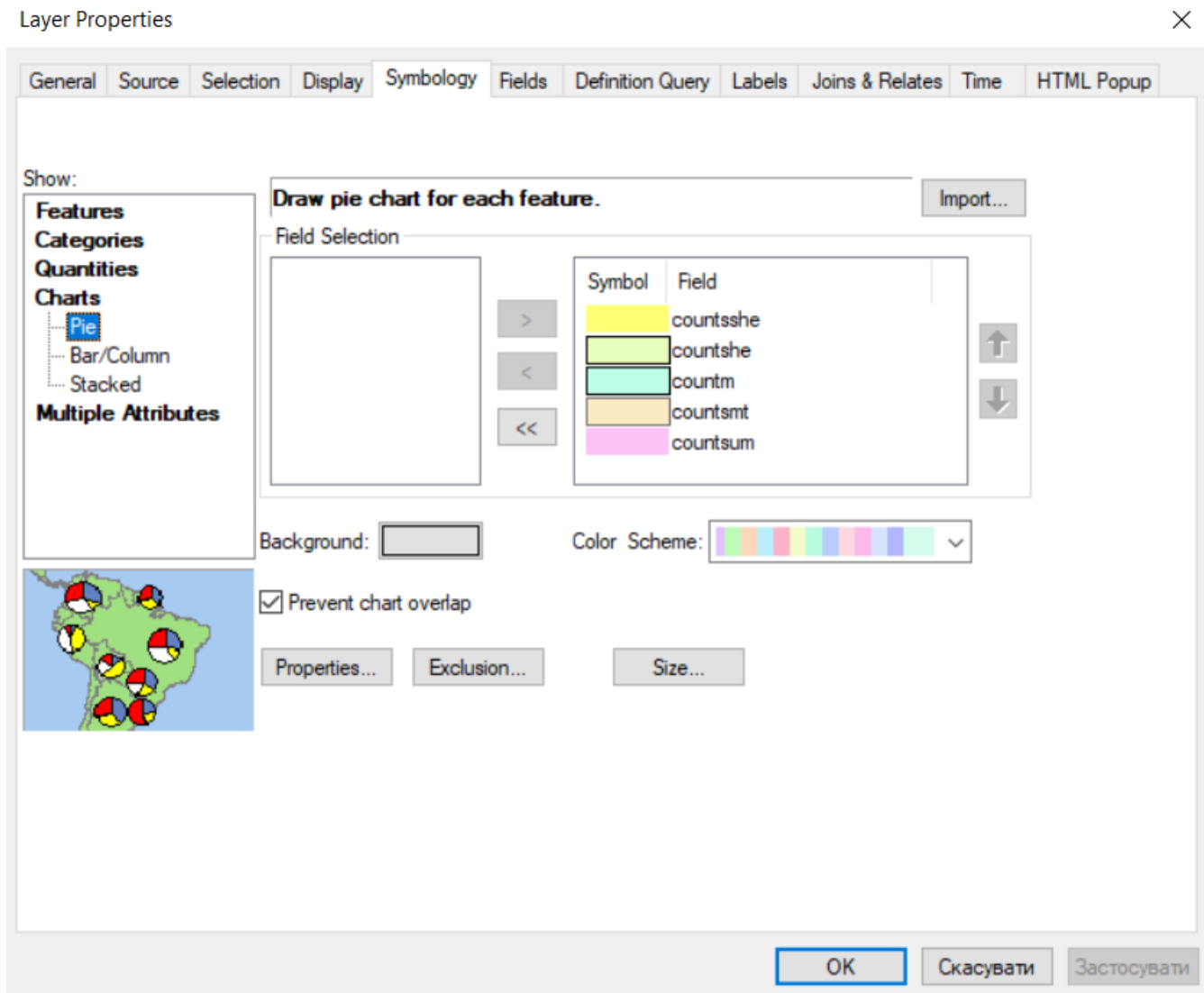


Рисунок 3.13 – Вікно налаштування тематичної карти

Побудуємо кругові діаграми в вкладці Pie, обираючи всі стовпці, які мають числові значення, а потім натискаємо "ОК". Це дозволить візуалізувати розподіл числових значень з кожного стовпця у вигляді кругових діаграм.

Таким чином, ми зможемо швидко та ефективно проаналізувати розподіл числових даних за допомогою кругових діаграм (рис. 3.14).

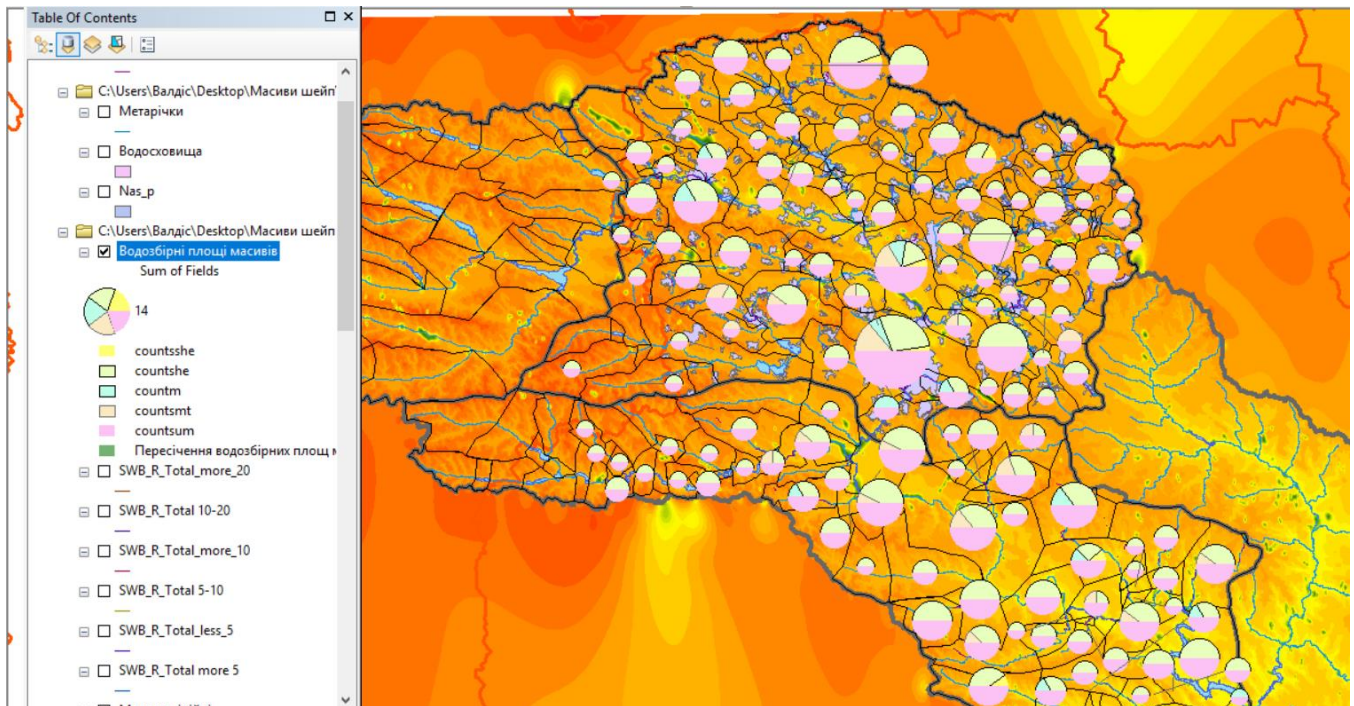


Рисунок 3.14 – Тематична карта населених пунктів

Як результат, отримуємо тематичну карту з круговими діаграмами, які відображають кількість перетинів річок з населеними пунктами за кольором. Рожевий колір відображає загальну кількість перетинів, яка займає половину круга. Жовтий колір відповідає селищам, світло-зелений - селам, світло-оранжевий - селищам міського типу, а блакитний - містам.

Аналогічним методом створюємо тематичну карту кількості населення, де колір відображає чисельність населення в різних населених пунктах. Для цього використовуємо стовпець «population», який містить інформацію про кількість мешканців у кожному населеному пункті. Залежно від величини цього стовпця, кожен пункт буде зафарбований у відповідний колір на карті, що дозволить візуально оцінити розподіл населення.

Для покращення інтерпретації даних і зручності аналізу, у режимі компоновки в ArcGIS було додано легенду, яка пояснює значення кольорових індикаторів на карті. Процес додавання легенди виконувався наступним чином:

1. Перехід до режиму компоновки. На верхній панелі інструментів обрано «Вид» (View) і далі «Компонування» (Layout View);
2. Вибір інструменту додавання легенди. На панелі інструментів знайдено і вибрано «Додати легенду» (Add Legend), що відкрило майстер створення легенди;

3. Налаштування легенди. У майстрі створення легенди вибрано шари, які повинні бути включені в легенду, зокрема ті, що містять інформацію про чисельність населення;

4. Форматування легенди. Вибрано стиль і формат легенди відповідно до потреб. Змінено шрифти, кольори, рамки та інші параметри, щоб легенда була чіткою та інформативною;

5. Розміщення легенди. Після налаштування легенду розміщено на карті в зручному місці, з можливістю переміщувати та змінювати розмір легенди для кращого відображення;

6. Завершення компоновки. Після додавання легенди перевірено, як вона виглядає на карті, і внесено необхідні корективи (рис. 3.15).

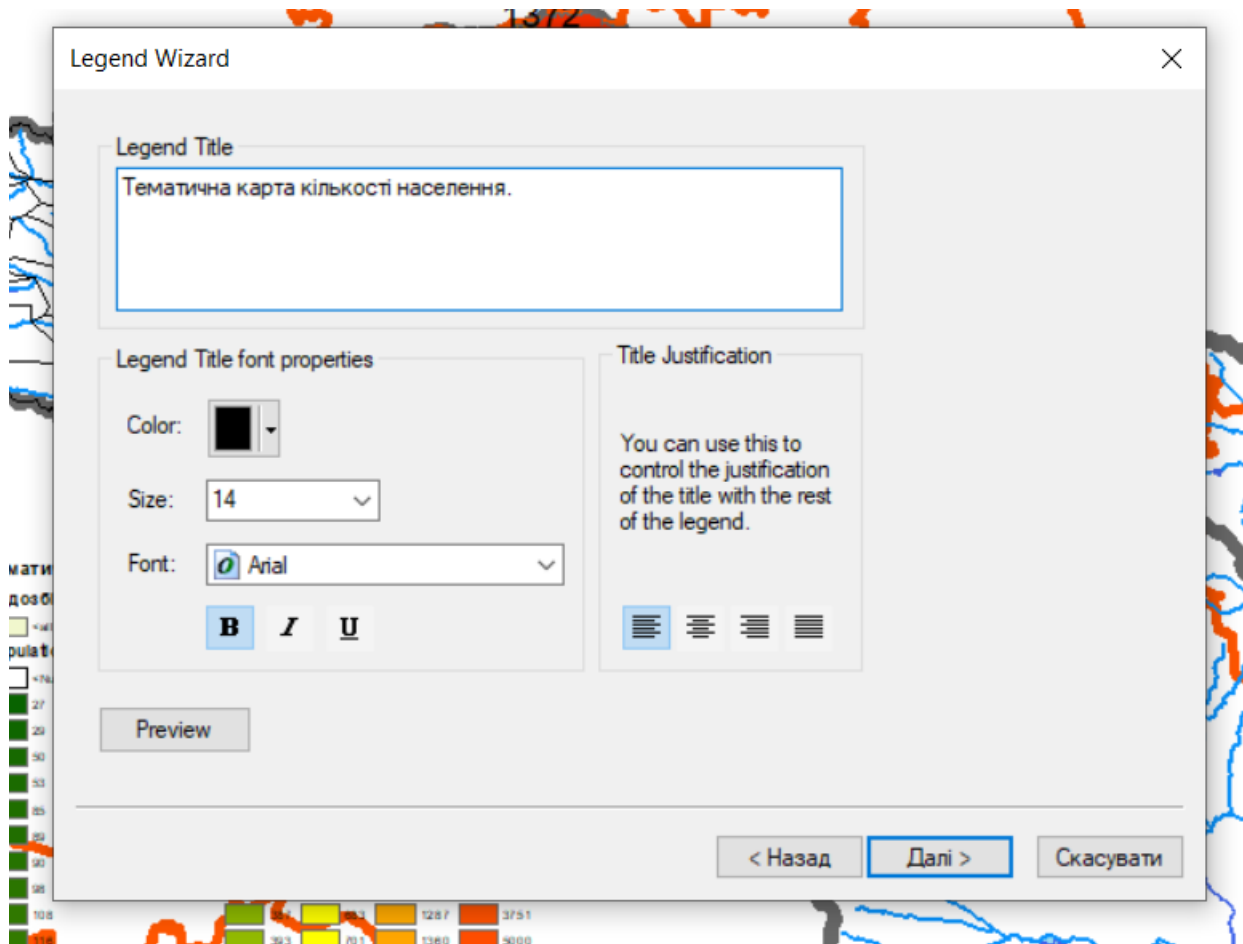


Рисунок 3.15 – Створення легенди карти

Це допомогло зробити карту більш інформативною та зручною для аналізу, оскільки користувачі можуть легко розуміти значення кольорових індикаторів, які відображають чисельність населення в різних населених пунктах (рис. 3.16).

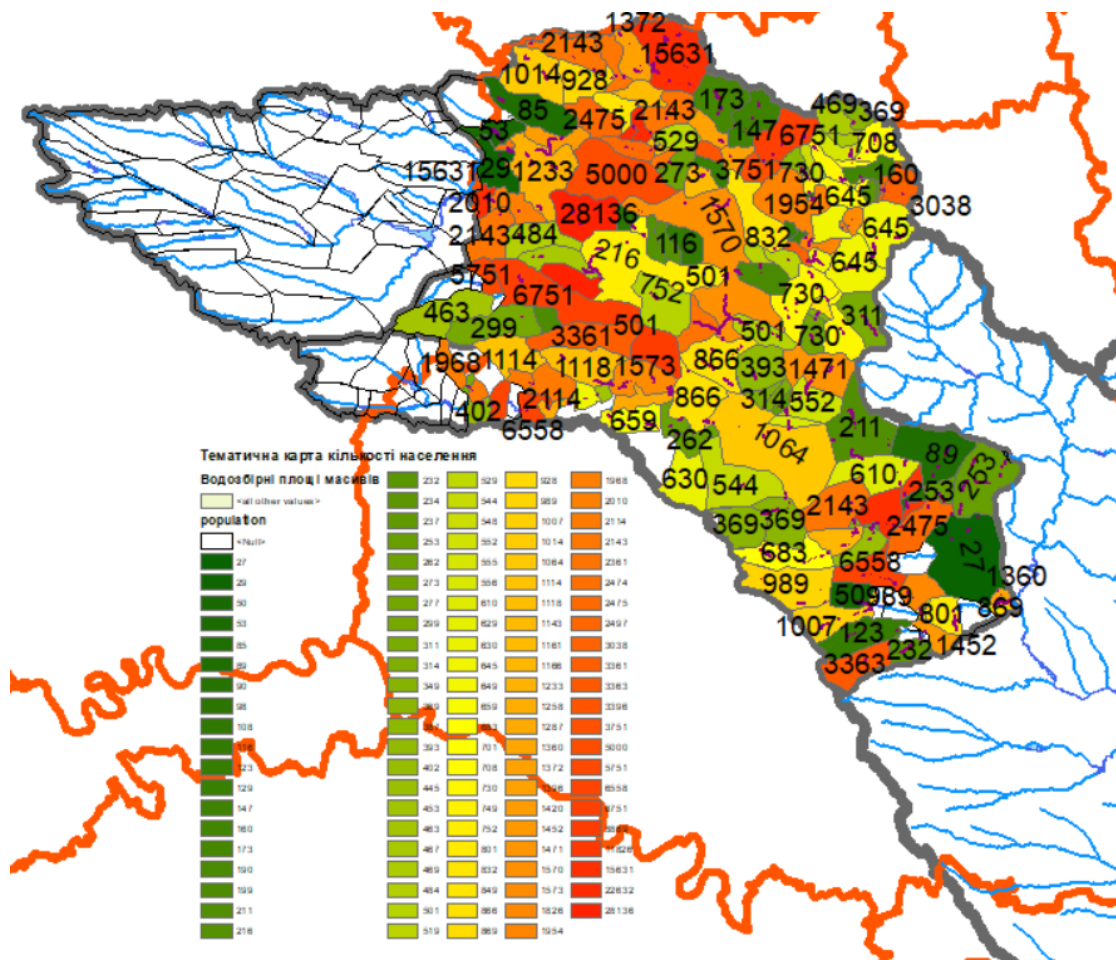


Рисунок 3.16 – Тематична карта кількості населення

Початково, населені пункти з найменшою кількістю мешканців будуть зафарбовані в салатовий колір. Цей колір обрано для позначення меншої щільності населення, що дає змогу швидко ідентифікувати області з низькою чисельністю мешканців. Чим більше населення в конкретному пункті, тим інтенсивніше змінюється колір до оранжевого. Оранжевий колір символізує найбільшу щільність населення, дозволяючи легко виділити області з високою чисельністю мешканців.

Такий метод візуалізації дозволяє оперативно оцінити розподіл перетинів річок з населеними пунктами за їхнім типом, використовуючи кольорове кодування. Графічне представлення даних у вигляді тематичної карти та кругових діаграм надає зручний спосіб аналізу, де різні кольори відображають різні категорії населених пунктів. Це дозволяє швидко виявити патерни та тенденції у взаємодії річок із селищами, сілами та містами, що може бути корисним при плануванні і розробці стратегій управління водними ресурсами та інфраструктурою населених пунктів.

3.3 Висновки

У третьому розділі представлені результати досліджень та аналіз водозбірних площ із відповідною візуалізацією.

Створення графіків у середовищі Google Colab забезпечило ефективну візуалізацію даних. За допомогою Python було побудовано різноманітні графіки, що дозволило наочно представити зміни у водному режимі та взаємозв'язки між гідрологічними та екологічними параметрами.

Тематична карта, створена засобами ArcGIS та Python, відобразила просторові зв'язки та структуру водних ресурсів басейну річки Південний Буг. Ця карта дозволила краще зрозуміти геопросторові закономірності та виявити проблемні зони, що потребують додаткової уваги.

Загалом, третій розділ демонструє використання сучасних інструментів для аналізу та візуалізації водозбірних площ, що сприяє глибшому розумінню досліджуваних об'єктів і прийняттю обґрунтованих рішень щодо управління водними ресурсами.

ВИСНОВКИ

В ході виконання бакалаврської дипломної роботи проведено комплексний системний аналіз геопросторових зв'язків водних масивів басейну річки Південний Буг, що дозволило визначити суть проблеми та довести актуальність дослідження для ефективного управління водними ресурсами.

У середовищі ArcGIS були створені водозбірні площі, що дозволило точно визначити межі басейнів та їхні характеристики, що стало основою для подальшого аналізу. За допомогою Python у середовищі Google Colab було проведено системний аналіз геопросторових даних, що дозволило обробити великі обсяги інформації та розробити алгоритми для аналізу взаємозв'язків між гідрологічними та екологічними параметрами.

Таблиці та графіки, створені для візуалізації характеристик водозбірних площ, допомогли наочно представити зміни у водному режимі та виявити ключові закономірності. Тематичні карти, створені на основі проведеного аналізу, відобразили просторові зв'язки та структуру водних ресурсів басейну, що сприяло глибшому розумінню гідрологічної та екологічної ситуації в регіоні.

Загалом, виконані завдання дозволили комплексно дослідити водні ресурси басейну річки Південний Буг, використовуючи сучасні геоінформаційні технології та інструменти. Отримані результати можуть бути корисними для прийняття рішень щодо управління водними ресурсами, охорони довкілля та планування стійкого розвитку регіону.

За результатами даної роботи була зроблена доповідь на тему «Системний аналіз геопросторових зв'язків масивів вод басейну річки Південний Буг» у «Молодь в науці: дослідження, проблеми, перспективи (Вінниця, 2023-2024 рр.)» з публікацією тез, а також опубліковано одну фахову статтю «Метод аугментації текстів про стан масивів вод на основі інтелектуальної прив'язки до багатозв'язних геоінформаційних систем іменованих сутностей» у Віснику Вінницького політехнічного інституту (2023).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Караваєв В.О., Крижановський Є.М., Бойко О.Р. Системний аналіз геопросторових зв'язків масивів вод басейну річки Південний Буг. *Міжнародна науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи (Вінниця, 2023-2024 рр.)»*. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/view/21654>
2. Мокін В. Б., Бондалєтов К. О., Крижановський Є. М., Караваєв В. О. Метод аугментації текстів про стан масивів вод на основі інтелектуальної прив'язки до багатозв'язних геоінформаційних систем іменованих сутностей. *Вісник ВПІ*. 2023. № 3. С. 55–65. DOI: <https://doi.org/10.31649/1997-9266-2023-168-3-55-65>
3. Система прийняття управлінських рішень керівниками водогосподарських організацій для басейну річки Південний Буг з використанням геоінформаційних технологій: методичний посібник. В. Б. Мокін та ін. ; за ред. В.Б. Мокіна Вінниця: УНІВЕРСУМ-Вінниця, 2021. 244 с. URL: <https://ir.lib.vntu.edu.ua/handle/123456789/7328?locale-attribute=en>
4. Мокін В. Б., Гораш М. А., Крижановський Є. М., Вуж Т. Є. Інформаційна інтелектуальна технологія автоматизованої геоприв'язки екологічної текстової природно-мовної інформації, *Наукові праці ВНТУ*, № 4, 2020. [Електронний ресурс]. URL: <https://praci.vntu.edu.ua/index.php/praci/article/view/624>
5. Directive 2000/60/ec of the European Parliament and of the Council. EUR-Lex –to European Union Law. Access: 07.06.2023. URL: https://eur-lex.europa.eu/resource.html?uri=cellar:5c835afb-2ec6-4577-bdf8-756d3d694eeb.0004.02/DOC_1&format=PDF
6. Верховна Рада України, Водний кодекс України, Кодекс України від 06.06.1995 р. № 213/95-ВР, станом на 19 серп. 2022 р. URL: <https://zakon.rada.gov.ua/laws/show/213/95-вр#Text>
7. Кабінет міністрів України, Постанова від 18.05.2017 р. № 336, Про затвердження Порядку розроблення плану управління річковим басейном URL: <https://www.kmu.gov.ua/npas/249999756> . Дата звернення 04.06.2023.

8. Draw.io – середовище для створення діаграм та схем URL:
<https://app.diagrams.net/>
9. Pre-modeling by Arc Hydro Tools. *International Journal for Engineering Modelling*, Vol. 37 No. 1, 2024. DOI:
<https://doi.org/10.31534/engmod.2024.1.ri.04a>
10. Maribeth H. P. Mastering ArcGIS Pro Maribeth., МН Price 2023. 25 с.
[Електронний ресурс] URL: <https://lazytrader.org/wp-content/uploads/2023/07/9781264091201-p.pdf>
11. OpenStreetMap (OSM) сервіс для завантаження карт URL:
<https://openstreetmap.org>
12. Google Colab безкоштовний хмарний сервіс для виконання Python-коду URL:
<https://colab.research.google.com/>
13. Introduction to ArcGIS Pro Energy Reports. 2023. [Електронний ресурс] URL:
https://link.springer.com/chapter/10.1007/978-3-031-42227-0_1
14. Tutorial: Geodatabase Workbook ArcGIS, 2022. 49 р. [Електронний ресурс]
URL: https://downloads.esri.com/Support/documentation/ao_/GeodatabaseWorkbook_Tutorial.pdf
15. How Watershed Works ArcGIS Pro Documentation 2022. [Електронний ресурс]
URL: <https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/how-watershed-works.htm>
16. How to Create a Watershed Model Using the Hydrology Toolset on Esri Support. 2021. [Електронний ресурс] URL: <https://support.esri.com/en-us/knowledge-base/how-to-create-a-watershed-model-using-the-hydrology-too-000012346>
17. Pierce. L., Overlay Operations in ArcGIS Pro. 2021. [Електронний ресурс] URL:
<https://storymaps.arcgis.com/stories/094b457f95a24d84818e3f9081ea52bf>
18. Schilling J., Generate_SWMM_inp: An open-source QGIS plugin to import and export model input files for SWMM, 2022 [Електронний ресурс] URL:
<https://www.mdpi.com/2073-4441/14/14/2262>
19. Vitalii Mokin, and Kostiantyn Bondaletov, “SpaCy for Ukrainian text similarity,” *Kaggle: Your Machine Learning and Data Science Community*. [Електронний ресурс] URL: <https://www.kaggle.com/code/bondaletov/spacy-for-ukrainian-textsimilarity>

20. Мокін В. Б., Гребінь В. В., Крижановський Є. М. Методи оцінювання та засоби автоматизації розрахунку складових водогосподарського балансу районів річкових басейнів України. Вінниця : ВНТУ, 2023. – 168 с. URL: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/38140/134496.pdf?sequence=2&isAllowed=y>

Додаток А
(обов'язковий)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

« ____ » _____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на бакалаврську дипломну роботу

СИСТЕМНИЙ АНАЛІЗ ГЕОПРОСТОРОВИХ ЗВ'ЯЗКІВ МАСИВІВ ВОД
БАСЕЙНУ РІЧКИ ПІВДЕННИЙ БУТ

08-34.БДР.005.00.000 ТЗ

Керівник: к.т.н., доц.

_____ Євгеній КРИЖАНОВСЬКИЙ

« __ » _____ 2024 р.

Розробив студент гр. СА-206

_____ Вадим КАРАВАЄВ

« __ » _____ 2024 р.

1. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____2024р., та індивідуальне завдання на БДР, затверджене протоколом №__ засідання кафедри САІТ від «__» _____ 2024р.

2. Джерела розробки:

1) Мокін В. Б., Бондалетов К. О., Крижановський Є. М., Караваєв В. О. Метод аугментації текстів про стан масивів вод на основі інтелектуальної прив'язки до багатозв'язних геоінформаційних систем іменованих сутностей. Вісник ВПІ. 2023. № 3. С. 55–65. DOI: <https://doi.org/10.31649/1997-9266-2023-168-3-55-65>

2) Pre-modeling by Arc Hydro Tools. International Journal for Engineering Modelling, Vol. 37 No. 1, 2024. DOI: <https://doi.org/10.31534/engmod.2024.1.ri.04a>

3) OpenStreetMap (OSM) – сервіс для завантаження карт. URL: <https://openstreetmap.org>

3. Мета і призначення роботи.

Метою дослідження є здійснення системного аналізу геопросторових зв'язків масивів вод басейну річки Південний Буг.

4. Вихідні дані для проведення робіт:

Карта річки Південний Буг з OpenStreetMap, таблиця атрибутів у середовищі ArcGIS як основний датасет для аналізу. Похідний датасет, який містить кількість перетинів населених пунктів, імпортовано в ArcGIS для створення тематичної карти.

5. Методи дослідження:

Використання ГІС та системний аналіз за допомогою мови програмування Python.

6. Етапи роботи і терміни їх виконання:

- | | |
|--|---------------|
| а) Аналіз предметної області | _____ – _____ |
| б) Порівняльний аналіз проблематики | _____ – _____ |
| в) Вибір оптимальних інформаційних технологій | _____ – _____ |
| г) Здійснення системного аналізу геопросторових зв'язків масивів вод басейну річки Південний Буг | _____ – _____ |
| д) Оформлення матеріалів до захисту БДР | _____ – _____ |

7. Очікувані результати та порядок реалізації:

Виконаний системний аналіз геопросторових зв'язків водних масивів басейну річки Південний Буг.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання бакалаврських кваліфікаційних робіт для студентів спеціальностей: 124 «Системний аналіз», 126 «Інформаційні системи та технології» (освітня програма «Прикладні інформаційні технології»)).

9. Порядок приймання роботи

Публічний захист	«__» _____ 2024 р.
Початок розробки	«__» _____ 2024 р.
Граничні терміни виконання БДР	«__» _____ 2024 р.

Розробив студент групи СА-206 _____ Вадим КАРАВАЄВ

Додаток Б
(обов'язковий)

Протокол перевірки бакалаврської дипломної роботи на наявність текстових за-
позичень

Назва роботи: «Системний аналіз геопросторових зв'язків масивів вод басейну річ-
ки Південний Буг»

Тип роботи: бакалаврська дипломна робота

Підрозділ: кафедра САІТ

Показники звіту подібності Unicheck

Оригінальність 96,14 % Схожість 3,86 %

Аналіз звіту подібності (відмітити потрібне)

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

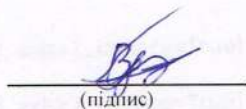
Особа, відповідальна за перевірку


(підпис)

Сергій ЖУКОВ

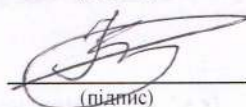
Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи


(підпис)

Вадим КАРАВАЄВ

Керівник роботи


(підпис)

Євгеній КРИЖАНОВСЬКИЙ

Додаток В
(довідниковий)
Фрагмент лістингу програми

Код з ноутбука catchment_Intersect9_TableToExcel_joint.ipynb

```
import pandas as pd

df = pd.read_csv('catchment_Intersect9_TableToExcel.csv')

#####

df

#####

df.info()

#####

columns_to_copy = ["SEM326", "SEM9.1", "adminua_1", "codename", "nameua_new", "adminua_2", "codename_3",
"adminua_3"]

new_df = df[columns_to_copy].copy()

new_df['codename'] = new_df['codename'].replace(
    ['селище', 'село', 'місто', 'селища міського типу'],
    ['с-ще', 'с', 'м', 'смт']
)

# Об'єднання стовпців
new_df['codenames'] = new_df.apply(lambda row: row['codename'] + '. ' + row['nameua_new'], axis=1)

# Об'єднання стовпців
new_df['Miskrada'] = new_df.apply(lambda row: row['adminua_3'] + ' ' + row['codename_3'], axis=1)

# Видалення оригіналів
new_df.drop(['adminua_3', 'codename_3'], axis=1, inplace=True)
new_df.drop(['codename', 'nameua_new'], axis=1, inplace=True)

new_df['SEM9.1'] = new_df['SEM9.1'].apply(lambda x: 'п. ' + x)

#####

grouped_df = new_df.groupby(['SEM326']).agg({
    'codenames': lambda x: ', '.join(sorted(set(x))),
    'SEM9.1': 'first',      # Перший елемент у групі
    'adminua_2': 'first',   # Перший елемент у групі
```

```

'Miskrada': 'first'      # Перший елемент у групі
}).reset_index()

grouped_df

////////////////////////////////////

grouped_df['geo_names'] = grouped_df.apply(lambda row: row['SEM9.1'] + ', ' + row['codenames']

                                     + ', ' + row['adminua_2'] + ', ' + row['Miskrada'], axis=1)

grouped_df.drop(['SEM9.1', 'codenames', 'adminua_2', 'Miskrada'], axis=1, inplace=True)

grouped_df

////////////////////////////////////

df.to_csv('TableToExcelExport.csv', index=False, encoding='utf-8-sig')

```

Код з ноутбука catchment_Intersect9_TableToExcel_schedule.ipynb

```

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

////////////////////////////////////

#Завантаження та огляд даних

df = pd.read_csv('catchment_Intersect9_TableToExcel.csv')

df

////////////////////////////////////

#Інформація про структуру датасету

df.info()

////////////////////////////////////

#Видалення стовпців з нульовими даними

df.drop(['DATECREATE', 'TIMECREATE', 'AUTHORNAME', 'SEM32850', 'SEM32851',

        'SEM32852', 'DATECORR', 'TIMECORR', 'AUTHORCORR', 'SEM32809', 'VIEWSCALE',

        'OBJCOLOR', 'FONTHEIGHT', 'FONTNAME', 'LINETHICK', 'THICKENDL', 'COLWEIGHT',

        'SEM_32856', 'BackColor', 'ObjInfo', 'SEM_32858', 'SEM_32203', 'SEM_32204',

        'DateFrom', 'SEM9', 'SEM327', 'SEM35', 'SEM152', 'SEM157', 'SEM4', 'SEM7',

        'SEM11', 'SEM20', 'SEM34', 'ObjectCode', 'koatuu_1'], axis=1, inplace=True)

////////////////////////////////////

df

////////////////////////////////////

# Вибір лише числових стовпців

numeric_df = df.select_dtypes(include=[float, int])

# Створення матриці кореляції

```

```

corr = numeric_df.corr()

# Побудова матриці кореляції за допомогою seaborn
plt.figure(figsize=(10, 8))
heatmap = sns.heatmap(corr, annot=True, cmap='coolwarm', vmin=-1, vmax=1, cbar_kws={'label': 'Кореляція'})

# Додавання заголовку
plt.title('Матриця кореляції', fontsize=16)

# Показ графіка
plt.show()

////////////////////////////////////

sns.pairplot(df, hue='SEM326')

////////////////////////////////////

# Підрахунок кількості кожного типу населеного пункту
intersection_counts = df['codename'].value_counts().reset_index()
intersection_counts.columns = ['codename', 'count']

# Візуалізація результатів
plt.figure(figsize=(12, 6))

barplot = sns.barplot(data=intersection_counts, x='codename', y='count', hue='codename', palette='viridis', dodge=False,
legend=False)

plt.xlabel('Тип населеного пункту')
plt.ylabel('Кількість перетинів')
plt.title('Кількість перетинів річок з різними типами населених пунктів')
plt.xticks(rotation=45)
plt.tight_layout()

# Додавання підписів над стовпцями
for index, row in intersection_counts.iterrows():
    barplot.text(index, row['count']+5, row['count'], color='black', ha="center", va='bottom', fontsize=10)

plt.show()

////////////////////////////////////

# Групування даних за типом населеного пункту та обчислення сумарного населення
grouped_data = df.groupby('type')['population'].sum().reset_index()

```

```
# Кольори для кожного типу населеного пункту
```

```
colors = {
    'с': 'skyblue',
    'с-ще': 'lightgreen',
    'м': 'coral',
    'сmt': 'gold'
}

bar_colors = [colors[typ] for typ in grouped_data['type']]
```

```
# Побудова графіку
```

```
plt.figure(figsize=(12, 6))

bars = plt.bar(grouped_data['type'], grouped_data['population'], color=bar_colors, edgecolor='black')
```

```
# Додавання підписів над стовпчиками
```

```
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, yval + 50, int(yval), ha='center', va='bottom', fontsize=10)
```

```
# Додавання заголовка та підписів до осей
```

```
plt.title('Населення за типами населених пунктів')
plt.xlabel('Тип населеного пункту')
plt.ylabel('Населення')
```

```
# Легенда
```

```
legend_labels = [plt.Line2D([0], [0], color=colors[typ], lw=4) for typ in colors]
plt.legend(legend_labels, colors.keys())
```

```
# Відображення графіку
```

```
plt.tight_layout()
plt.show()

////////////////////////////////////
```

```
# Підрахунок кількості кожного типу населеного пункту
```

```
intersection_counts = df['SEM326'].value_counts().reset_index()
intersection_counts.columns = ['SEM326', 'count']# Експонентний

plt.figure(figsize=(12, 6))

barplot = sns.barplot(data=intersection_counts, x='SEM326', y='count', hue='SEM326', palette='viridis', dodge=False,
legend=False)
```

```

plt.title('Розподіл значень SEM326')
plt.xlabel('SEM326')
plt.ylabel('Частота')
plt.xticks(intersection_counts.index[::3], rotation=45, ha='right')
plt.tight_layout()
plt.show()

////////////////////////////////////

new_df1 = df[['SEM326', 'type']].copy()

# Фільтрація даних
df_2021 = new_df1.loc[new_df1['type'] == 'смт']

# Групування за днями та підрахунок кількості випадків 'type' за кожен день
daily_counts = df_2021.groupby('SEM326')['type'].count()

plt.figure(figsize=(12, 6))

bars = plt.bar(daily_counts.index, daily_counts.values, color='green', width=0.5, edgecolor='black', alpha=0.7)

# Додавання текстових надписів над кожним стовпцем
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval + 0.1, round(yval, 2), ha='center', va='bottom', fontsize=10)

plt.title('Кількість перетинів водозбірних площ масивів вод')
plt.xlabel('Тип населеного пункту')
plt.ylabel('Кількість перетинів')
plt.xticks(rotation=45, ha='center')
plt.grid(axis='y') # Включення сітки по осі y
plt.tight_layout()
plt.show()

////////////////////////////////////

# Фільтруємо дані за типами населених пунктів
filtered_data = df[df['type'].isin(['с', 'с-ще', 'м', 'смт'])]

# Групування за типом населеного пункту та SEM326 та підрахунок кількості
grouped_data = filtered_data.groupby(['SEM326', 'type']).size().unstack()

# Побудова гістограми
plt.figure(figsize=(14, 8))
sns.histplot(data=grouped_data, bins=20, kde=True)

```



```

# Додавання заголовка
plt.title('Розподіл кількості перетинів водозбірних площ масивів вод за типом населеного пункту')
plt.xlabel('Тип населеного пункту')
plt.ylabel('Кількість перетинів водозбірних площ масивів вод')
plt.show()

////////////////////////////////////

# Фільтруємо дані за типами населених пунктів
filtered_data = df[df['type'].isin(['с', 'с-ще', 'м', 'смт'])]

# Групування за типом населеного пункту та SEM326 та підрахунок кількості
grouped_data = filtered_data.groupby(['SEM326', 'type']).size().unstack(fill_value=0)

# Побудова стовпчастої діаграми
fig, ax = plt.subplots(figsize=(14, 8))

# Побудова стовпчиків з різними кольорами для кожного типу населеного пункту
colors = ['skyblue', 'lightgreen', 'salmon', 'orange']
bars = grouped_data.plot(kind='bar', stacked=True, ax=ax, color=colors, edgecolor='black')

# Додавання текстових надписів над кожним стовпчиком
for container in ax.containers:
    ax.bar_label(container)

# Додавання заголовка та підписів до осей
plt.title('Розподіл кількості перетинів водозбірних площ масивів вод за типом населеного пункту')
plt.xlabel('SEM326')
plt.ylabel('Кількість перетинів водозбірних площ масивів вод')

# Додавання легенди
plt.legend(title='Тип населеного пункту', labels=['с', 'с-ще', 'м', 'смт'])

# Встановлення міток осі X
xticks = range(len(grouped_data.index))
ax.set_xticks(xticks)
ax.set_xticklabels(grouped_data.index, rotation=45, ha='right')

```

```

# Вибірково відображення міток осі X
for label in ax.get_xticklabels()[::2]:
    label.set_visible(True)
for label in ax.get_xticklabels()[1::2]:
    label.set_visible(False)

# Відображення графіку
plt.tight_layout()
plt.show()

////////////////////////////////////

# Групування за днями та обчислення щоденного ROMI
daily_romi = df.groupby(df['SEM326'])['type'].count()

# Побудова графіку
plt.figure(figsize=(12, 6))

# plt.plot(daily_romi.index, daily_romi.values, marker='o', linestyle='-')
plt.hist(daily_romi, bins=20, color='skyblue', edgecolor='black')
plt.title('Кількість перетинів річок з різними типами населених пунктів')
plt.xlabel('Кількість перетинів')
plt.ylabel('Тип населеного пункту')
plt.xticks( rotation=45, ha='right') # Відображаємо кожен 8-й день
plt.grid(True)

# Щоб підписи не накладалися
plt.tight_layout()
plt.show()

#, markerfacecolor='w', marker='o'

////////////////////////////////////

new_df1 = df[['SEM326', 'type']].copy()

# Фільтруємо дані за типами населених пунктів
filtered_data = new_df1[new_df1['type'].isin(['c', 'с-ще', 'м', 'смт'])]

# Групування за типом населеного пункту та SEM326 та підрахунок кількості
grouped_data = filtered_data.groupby(['SEM326', 'type']).size().unstack(fill_value=0)

# Фільтрація даних
df_2021 = new_df1.loc[new_df1['type'] == 'м']

# Групування за SEM326 та type та підрахунок кількості

```

```
#daily_counts = new_df1.groupby(['SEM326', 'type']).size()
```

```
# Побудова графіку
```

```
plt.figure(figsize=(12, 6))
```

```
plt.hist(grouped_data.values, edgecolor='black')
```

```
plt.title('Кількість перетинів водозбірних площ масивів вод')
```

```
plt.xlabel('Кількість перетинів')
```

```
plt.ylabel('Частота')
```

```
plt.grid(axis='y') # Включення сітки по осі y
```

```
# Додавання легенди
```

```
plt.legend(title='Тип населеного пункту', labels=['с', 'с-ще', 'м', 'сmt'])
```

```
plt.tight_layout()
```

```
plt.show()
```

Код з ноутбука catchment_Intersect9_TableToExcel_thematic_map.ipynb

```
////////////////////////////////////
```

```
import pandas as pd
```

```
////////////////////////////////////
```

```
df = pd.read_csv('catchment_Intersect9_TableToExcel.csv')
```

```
df
```

```
////////////////////////////////////
```

```
# Групування даних
```

```
grouped = df.groupby(['SEM326', 'type']).size().unstack(fill_value=0)
```

```
# Створення нових стовпців для кожного типу населеного пункту
```

```
grouped = grouped.rename(columns={'с-ще': 'countsshe', 'с': 'countshe', 'м': 'countm', 'сmt': 'countsm'})
```

```
# Додавання стовпця з загальною кількістю
```

```
grouped['countsum'] = grouped.sum(axis=1)
```

```
# Заповнення відсутніх стовпців, якщо їх не було у вихідних даних
```

```
for col in ['countsshe', 'countshe', 'countm', 'countsm']:
```

```
    if col not in grouped.columns:
```

```
        grouped[col] = 0
```

```
# Перевпорядкування колонок для зручності
```

```
grouped = grouped[['countsshe', 'countshe', 'countm', 'countsm', 'countsum']]
```

```
# Скидання індексу, щоб зробити SEM326 звичайним стовпцем
```

```
result = grouped.reset_index()
```

```
print(result)
```

```
////////////////////////////////////
```

```
# Збереження DataFrame у форматі .xls
```

```
result.to_excel('output.xlsx', index=False, engine='openpyxl')
```

Додаток Г
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

**СИСТЕМНИЙ АНАЛІЗ ГЕОПРОСТОРОВИХ ЗВ'ЯЗКІВ МАСИВІВ ВОД
БАСЕЙНУ РІЧКИ ПІВДЕННИЙ БУГ**

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2024 р.

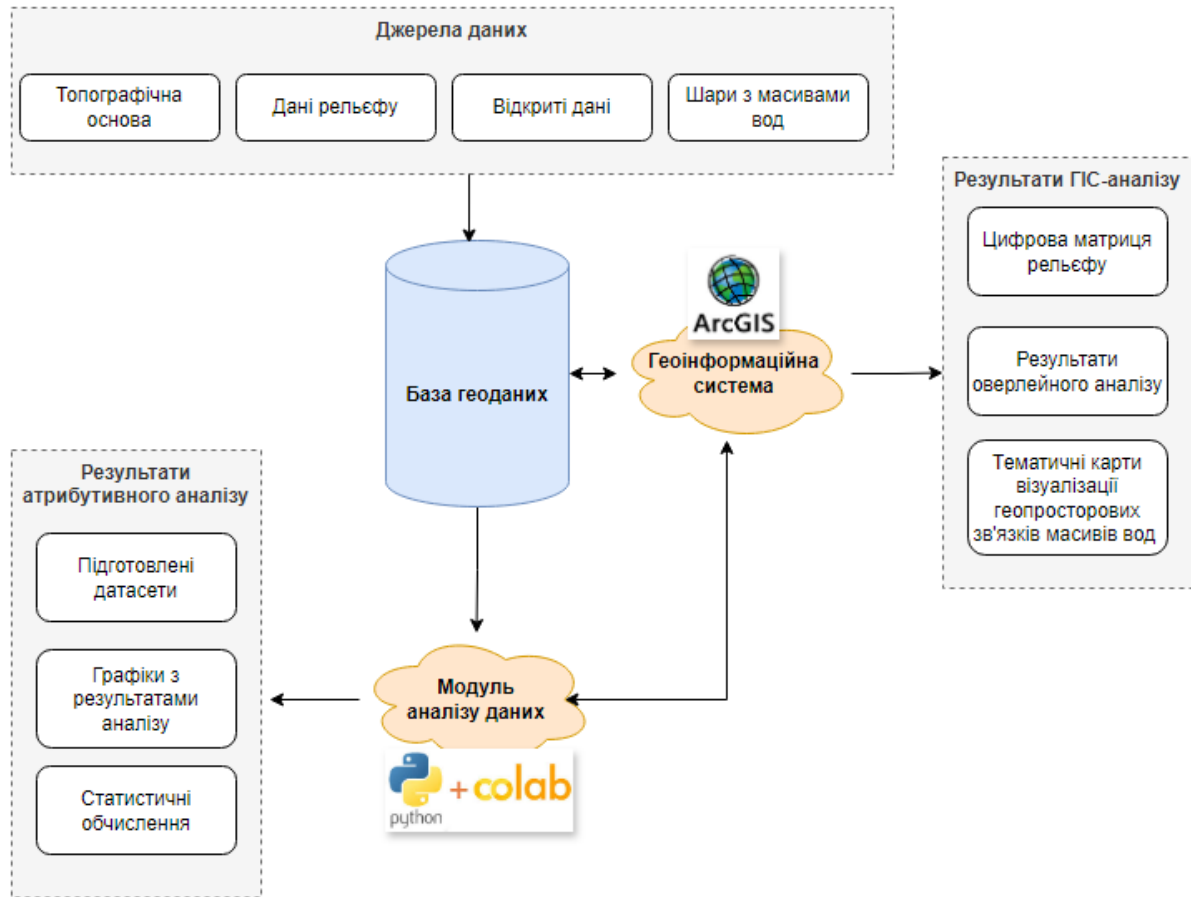


Рисунок Г.1 - Візуальне структурне представлення запропонованого підходу

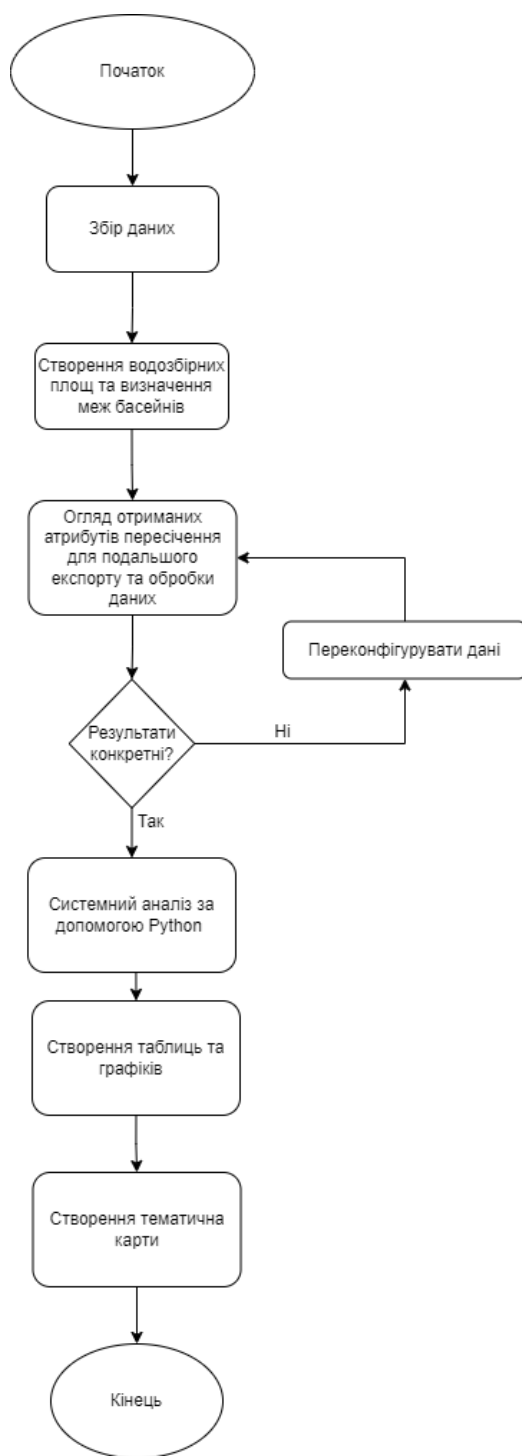


Рисунок Г.2 – Блок-схема поетапного вирішення поставленого завдання

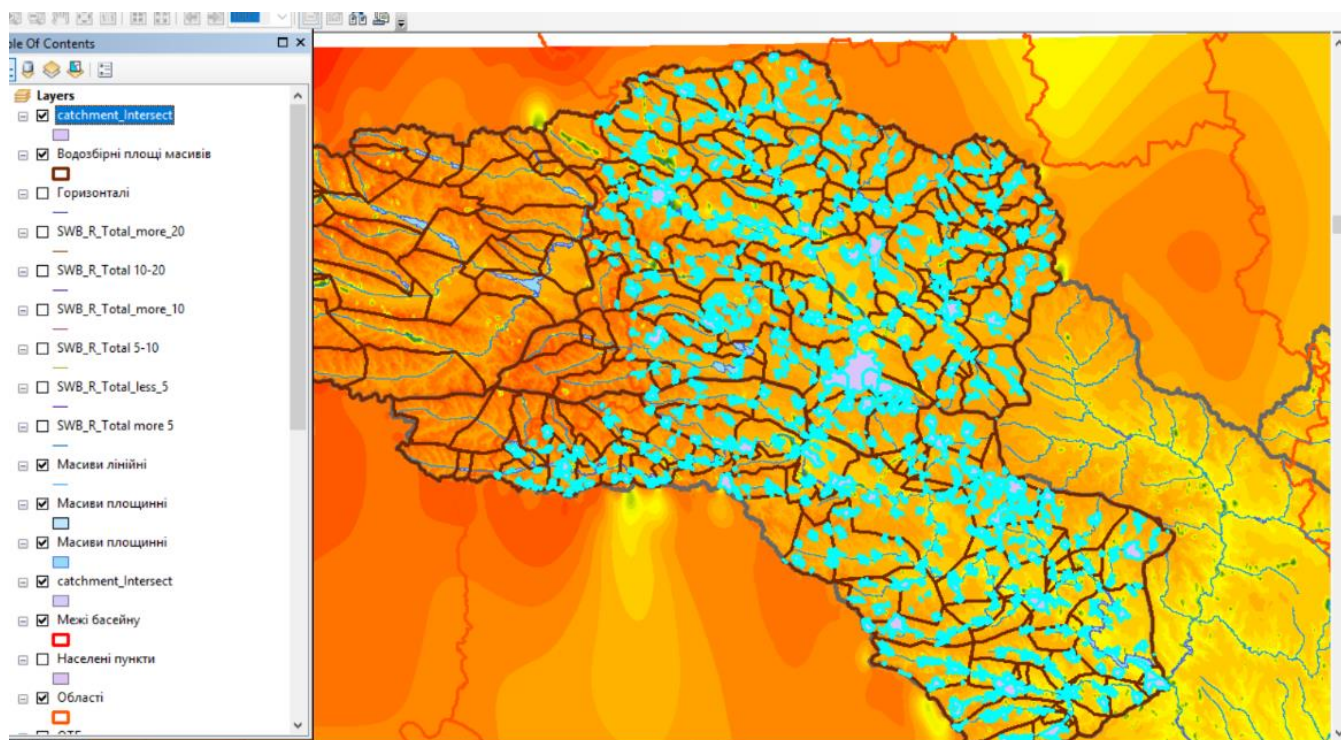


Рисунок Г.3 – Виділенні масиви річок у новому шарі catchment_Intersect

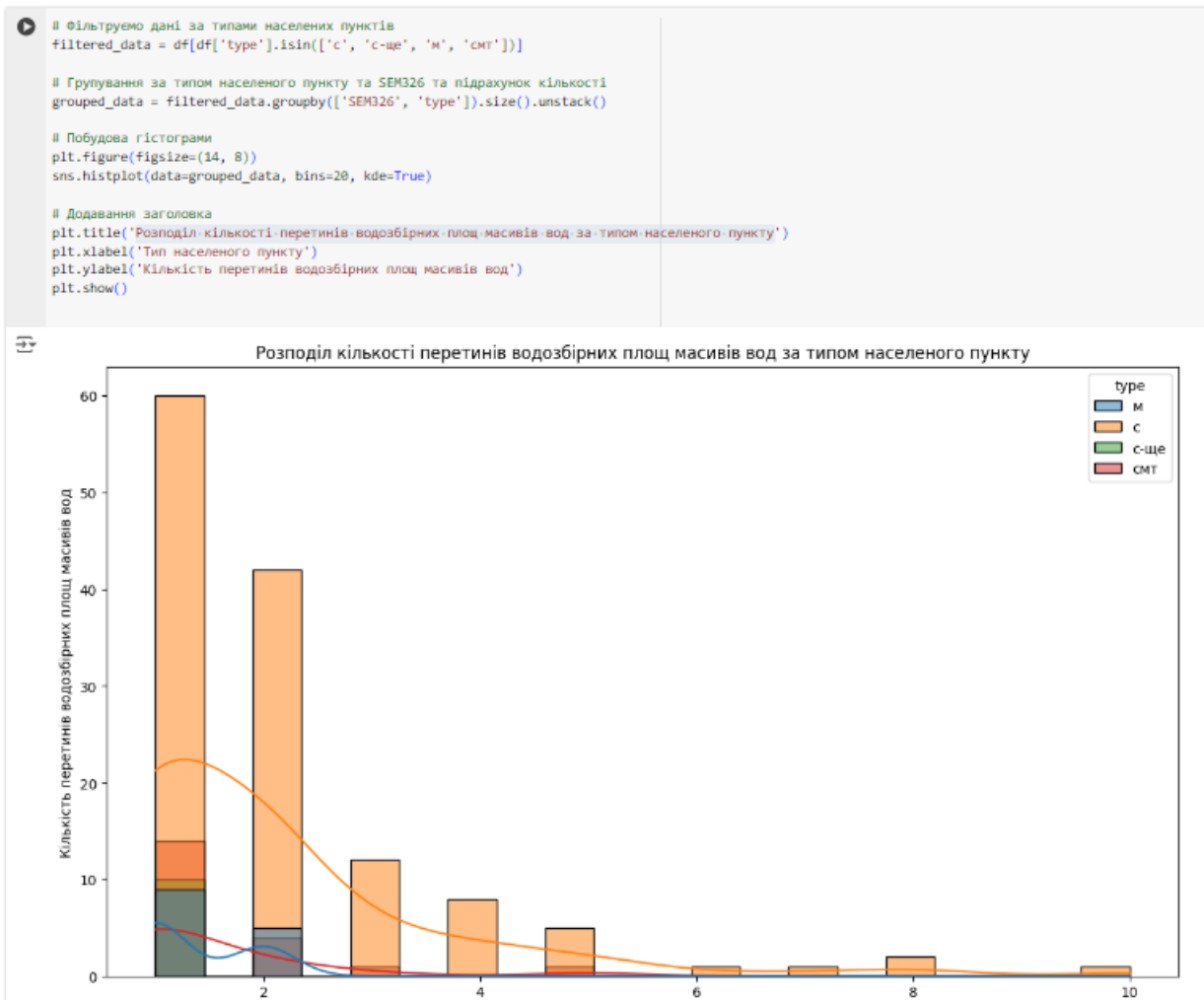


Рисунок Г.4 – Графік розподілу даних за геоприв’язкою та типом населеного пункту

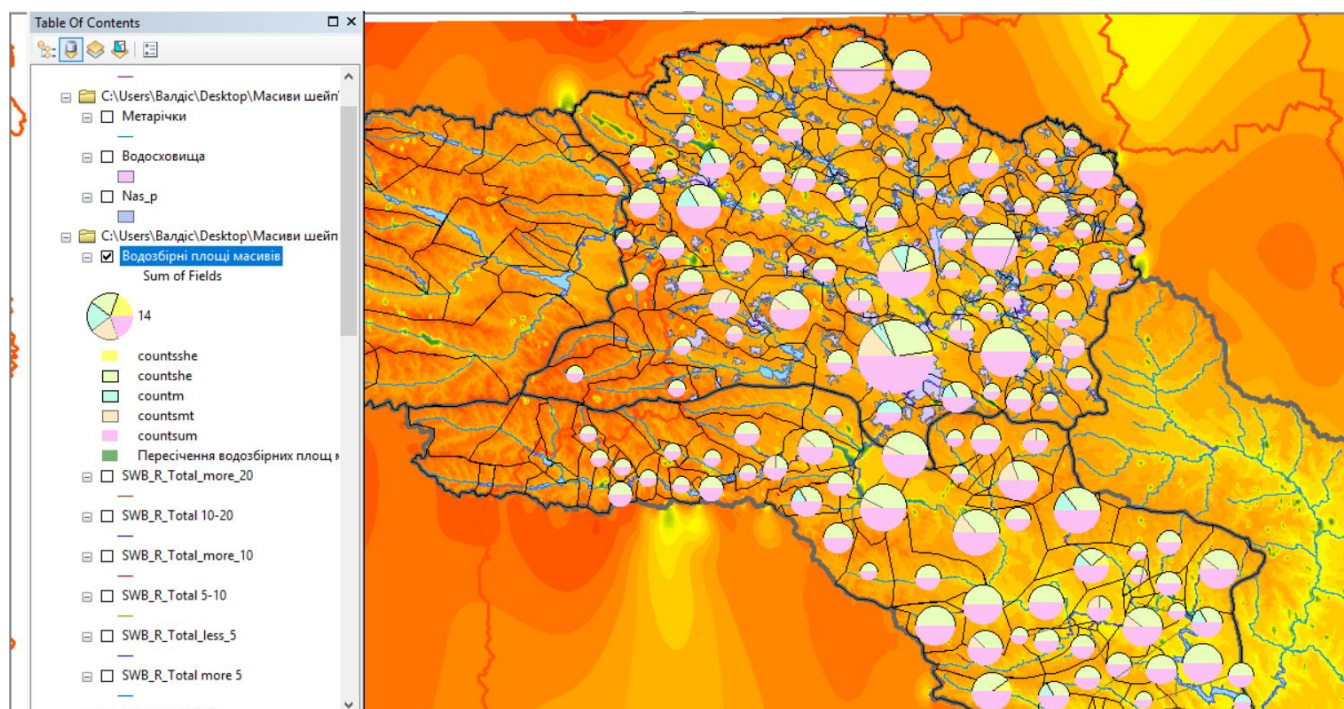


Рисунок Г.5 – Тематична карта населених пунктів