

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторної роботи
з дисципліни «Методи цифрового аналізу даних»
на тему «Підготовка даних для аналізу»
для студентів спеціальності
F7 (123) Комп'ютерна інженерія
всіх форм навчання

Методичні вказівки до виконання лабораторної роботи з дисципліни «Методи цифрового аналізу даних» на тему «Підготовка даних для аналізу» для студентів спеціальності F7 (123) Комп'ютерна інженерія всіх форм навчання / Укл. Т.В. Голуб, Т.С. Дьячук - Запоріжжя: НУ «Запорізька політехніка», 2025.- 45 с.

Укладачі: Т.В. Голуб, доцент
Т.С. Дьячук, старший викладач

Рецензент: С.С. Грушко, доцент, к.т.н.

Відповідальний
за випуск: Т.В. Голуб, доцент

Затверджено
на засіданні кафедри КСМ
Протокол № 2 від 29.08.2025

Рекомендовано до видання
на засіданні НМК факультету КНТ
Протокол 2 від 10.09.2025

ЗМІСТ

Вступ.....	4
Лабораторна робота: Підготовка даних для аналізу.....	5
1 Теоретичні відомості	5
1.1 Середовища для роботи	5
1.2 Необхідні бібліотеки для аналізу даних	5
1.3 Етапи процесу аналізу даних.....	9
1.4 Підготовка даних для подальшого аналізу	10
2 Підготовка даних до аналізу	12
2.1 Завантаження даних	12
2.2 Обробка невизначених даних	16
2.3 Виявлення проблемних місць і постановка завдання.....	18
2.4 Робота зі стовпцями.....	19
2.5 Повторна перевірка на невизначеність даних	27
2.6 Виявлення повторів	28
2.7 Заміна назв стовпців.....	29
3 Лабораторне завдання.....	31
4 Зміст звіту	32
5 Контрольні питання	33
Перелік джерел посилань	34
Додаток А Опис Datasets.....	35

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій і зростання обсягів цифрових даних ефективний аналіз інформації стає ключовим чинником успішного прийняття рішень у багатьох сферах — від бізнесу та економіки до медицини й науки. Водночас, якість аналітичних висновків значною мірою залежить від якості вхідних даних. Саме тому етап підготовки даних є критично важливою складовою процесу цифрового аналізу.

Підготовка даних включає низку процедур, зокрема очищення, нормалізацію, обробку пропущених значень, кодування категоріальних змінних, редукцію розмірності та інші методи, що забезпечують готовність даних до подальшого статистичного або машинного аналізу. Належне опрацювання даних дозволяє зменшити похибки, покращити точність моделей та пришвидшити процес обробки інформації.

Метою цієї лабораторної роботи є набуття практичних навичок з підготовки цифрових даних до аналізу, освоєння базових технік передобробки даних із використанням сучасних інструментів та програмних засобів.

У ході виконання завдань студенти навчатимуться:

- виявляти та усувати проблеми у сирих даних;
- застосовувати основні методи трансформації та нормалізації даних;
- підготувати датасети до подальшого використання в аналітичних або машинних моделях.

Лабораторна робота орієнтована на формування практичних компетентностей, необхідних для ефективної роботи з реальними наборами даних у рамках цифрового аналізу.

ЛАБОРАТОРНА РОБОТА: ПІДГОТОВКА ДАНИХ ДЛЯ АНАЛІЗУ

Мета роботи: отримати навички очищення, нормалізації та стандартизації даних (підготовки даних).

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Середовища для роботи

Одним із найбільш розповсюджених інструментів для аналізу даних є використання бібліотек мови програмування Python.

Таким чином, для проведення такого типу аналізу можна скористатися наступними середовищами:

- скачати та встановити середовище мови Python з підключенням необхідних бібліотек (перелік бібліотек надаватиметься далі);
- скористатися онлайн-середовищем Google Colaboratory.

Для можливості автономної роботи варто встановити стаціонарне **програмне забезпечення мови Python**, яке можна скачати з офіційного сайту за посиланням [1]: <https://www.python.org/> (Для спрощення подальшого підключення бібліотек шлях до папки установки вказати типу: C:\Python\Python312).

При можливості роботи з постійним підключенням до інтернету можна скористатися **середовищем Google Colaboratory**, розробленим компанією Google та призначеним для спеціалістів по обробці даних і дослідженнях в області штучного інтелекту. Середовище дозволяє писати та виконувати код мови Python безпосередньо в браузері. Посилання на середовище: <https://colab.research.google.com/>.

1.2 Необхідні бібліотеки для аналізу даних

До найбільш часто використовуваних бібліотек при аналізі даних та презентації отриманих висновків відносяться:

- NumPy;

- Pandas;
- Matplotlib;
- IPython;
- SciPy.

При роботі на персональному комп'ютері даний перелік бібліотек встановлюється у відповідності з рекомендаціями. При використанні онлайн середовища Google Colaboratory доступ до бібліотек є автоматичним і не вимагає додаткових маніпуляцій.

1.2.1 Бібліотека NumPy

NumPy, скорочення від «Numerical Python», - основний пакет для виконання наукових розрахунків на Python. Він надає наступні особливості:

- можливість швидкого та ефективного створення об'єктів багатовимірних масиву ndarray;
- наявні функції для виконання обчислень над елементами одного масиву або математичних операцій з кількома масивами;
- наявні засоби для читання і запису на диски наборів даних, представлених у вигляді масивів;
- передбачені функції для виконання операцій лінійної алгебри, перетворення Фур'є і генерування випадкових чисел;
- наявні засоби для інтеграції з кодом, написаним на C, C++.

Крім швидких засобів роботи з масивами, однією з основних цілей NumPy в частині аналізу даних є організація контейнера для передачі даних між алгоритмами. Як засіб зберігання і маніпуляції даними масиви NumPy значно ефективніші за вбудовані в Python структури даних.

Для установки пакету необхідно виконати наступне: відкрити консоль (*Win+R -> cmd*) та набрати команду: *pip install numpy*.

1.2.2 Бібліотека Pandas

Бібліотека pandas надає структури даних і функції, спрямовані на спрощення та прискорення роботи зі структурованими даними. Основний об'єкт Pandas – DataFrame (рис. 1.1) - двовимірна таблиця, у якій рядки і стовпці мають мітки.

Бібліотека Pandas поєднує високу продуктивність засобів роботи з масивами, притаманну NumPy, з гнучкими можливостями маніпулювання даними, властивими електронним таблицям і реляційним базам даних. Вона надає розвинені засоби індексування, що дають змогу легко змінювати форму наборів даних, формувати поздовжні і поперечні зрізи, виконувати агрегування і вибирати підмножини.

```
>>> frame
   total_bill  tip  sex  smoker  day  time  size
1    16.99    1.01 Female    No   Sun  Dinner    2
2    10.34    1.66  Male    No   Sun  Dinner    3
3    21.01    3.5  Male    No   Sun  Dinner    3
4    23.68    3.31  Male    No   Sun  Dinner    2
5    24.59    3.61 Female    No   Sun  Dinner    4
6    25.29    4.71  Male    No   Sun  Dinner    4
7     8.77     2  Male    No   Sun  Dinner    2
8    26.88    3.12  Male    No   Sun  Dinner    4
9    15.04    1.96  Male    No   Sun  Dinner    2
10   14.78    3.23  Male    No   Sun  Dinner    2
```

Рисунок 1.1 – Приклад DataFrame в Pandas

Для розробки фінансових додатків Pandas пропонує широкий набір спеціалізованих високопродуктивних засобів аналізу часових рядів.

Сама назва Pandas утворена від:

- Panel Data (панельні дані) – термін використовується в економетриці для позначення багатовимірних структурованих наборів даних;
- Python Data Analysis – термін характеризує процес аналізу даних за допомогою Python.

Для установки пакету необхідно виконати наступне: відкрити консоль (*Win+R -> cmd*) та набрати команду: *pip install pandas*.

При встановленні бібліотеки Pandas автоматично завантажується і бібліотека NumPy.

1.2.3 Бібліотека Matplotlib

Бібліотека Matplotlib – один із найпопулярніших у Python інструментів для створення графіків та інших способів візуалізації двовимірних даних. Інтегрована з IPython, що дає змогу організувати зручне інтерактивне оточення для візуалізації та дослідження даних. Побудовані за допомогою даної бібліотеки графіки є інтерактивними – можна збільшити масштаб певної ділянки графіка і виконувати панорамування за допомогою панелі інструментів у вікні графіка.

Для установки пакету необхідно виконати наступне: відкрити консоль (*Win+R -> cmd*) та набрати команду: *pip install matplotlib*.

1.2.4 Бібліотека Seaborn

Seaborn – це бібліотека для створення статистичних графіків на Python. Вона ґрунтується на matplotlib і тісно взаємодіє зі структурами даних pandas.

Архітектура Seaborn дозволяє швидко вивчити та зрозуміти дані. Seaborn захоплює цілі фрейми даних або масиви, в яких містяться всі дані, і виконує всі внутрішні функції, потрібні для семантичного дослідження та статистичної агрегації для перетворення даних на інформативні графіки.

Вона абстрагує складність, дозволяючи проєктувати графіки відповідно до висунутих потреб.

Для установки пакету: відкрити консоль (*Win+R -> cmd*) та набрати команду: *pip install seaborn*.

Під час встановлення Seaborn бібліотека встановить всі залежності, включаючи Matplotlib, pandas, numpy та scipy.

1.2.5 Бібліотека IPython

IPython - компонент стандартного набору інструментів наукових розрахунків на Python, який пов'язує всі спеціалізовані інструменти. Він забезпечує надійне високопродуктивне середовище для інтерактивних і дослідницьких розрахунків. Це оболонка Python з додатковими можливостями, що має на меті прискорити написання, тестування та налагодження коду на Python. Особливо вона корисна для роботи з даними та їх візуалізації за допомогою matplotlib.

Для установки пакету: відкрити консоль (*Win+R -> cmd*) та набрати команду: *pip install ipython*.

1.2.6 Бібліотека SciPy

SciPy – зібрання пакетів, призначених для розв’язання різних стандартних обчислювальних задач. До таких пакетів відносяться:

- *scipy.integrate*: підпрограми чисельного інтегрування та розв’язання диференціальних рівнянь;
- *scipy.linalg*: підпрограми лінійної алгебри та розкладання матриць, що доповнюють ті, що включені в *numpy.linalg*;
- *scipy.optimize*: алгоритми оптимізації функцій (знаходження мінімумів) і пошуку коренів;
- *scipy.signal*: засоби обробки сигналів;
- *scipy.sparse*: алгоритми роботи з розрідженими матрицями та розв’язання розріджених систем лінійних рівнянь;
- *scipy.stats*: стандартні безперервні та дискретні розподіли ймовірностей (функції щільності ймовірності, формування вибірки, функції безперервного розподілу ймовірності), різноманітні статистичні критерії та додаткові описові статистики;
- *scipy.weave*: засіб для вбудовування коду на C++ з метою прискорення обчислень з масивами.

Для установки пакету: відкрити консоль (*Win+R -> cmd*) та набрати команду: *pip install scipy*.

1.3 Етапи процесу аналізу даних

Для того, щоб отримати результати щодо аналізу даних, необхідно виконати наступні основні етапи:

- взаємодія із зовнішнім світом – читання і запис даних, що зберігаються у файлах різних форматів і в базах даних;
- підготовка – очищення, переформатування, комбінування, нормалізація, зміна форми, формування поздовжніх і поперечних зрізів, перетворення даних для аналізу;

- перетворення – застосування математичних і статистичних операцій до груп наборів даних для породження нових наборів: наприклад, агрегування великої таблиці за груповими змінними;
- моделювання та розрахунок – з'єднання даних зі статистичними моделями, алгоритмами машинного навчання та іншими обчислювальними засобами;
- презентація – створення інтерактивних або статичних графічних уявлень або текстових зведень.

1.4 Підготовка даних для подальшого аналізу

Після завантаження набору даних, які можуть бути отримані з різних джерел, необхідно їх підготувати для подальшого аналізу. Цей процес отримав назву Data Wrangling.

Data Wrangling - (англ. «конфлікт даних») – це процес очищення та об'єднання «сирих» і складних наборів даних для легкого доступу та аналізу.

Оскільки кількість даних і джерел даних швидко зростають і розширюються, стає все більш важливою задачею питання організації великих обсягів доступних даних для аналізу. Цей процес зазвичай включає в себе ручне перетворення та відображення даних з одного в інший формат, щоб забезпечити більш зручне використання та організацію даних.

На практиці є три загальні завдання, пов'язані з процесом Data Wrangling:

- Data cleaning (очистка даних) – видалення неінформативних та помилкових елементів даних;
- Data transformation (перетворення даних) – приведення даних до зручних форматів в залежності від інформації, що там зберігається;
- Data enrichment (доповнення даних) – доповнення вхідних даних елементами, яких не вистачає, з урахуванням логіки інформації, що аналізується.+

До обробки та підготовки даних також відносяться наступні дії:

- стандартизація даних;
- нормалізація даних;
- бінінг;
- створення індикаторної змінної.

Стандартизація даних – дані зазвичай збираються від різних агентств у різних форматах. Стандартизація — це процес перетворення даних у загальний формат, що дозволяє досліднику проводити змістовне порівняння. Стандартизація даних також є терміном для певного типу нормалізації даних, яка досягається відніманням від поточного значення середнього і діленням на стандартне відхилення.

Нормалізація даних – це процес перетворення значень кількох змінних у подібний діапазон. Типові нормалізації включають:

- масштабування змінної таким чином, щоб середнє значення змінної становило 0;
- масштабування змінної, щоб дисперсія становила 1;
- масштабування змінної, щоб значення були в діапазоні від 0 до 1;

Бінінг – це процес перетворення неперервних числових змінних у дискретні категоріальні «біни» для згрупованого аналізу. Наприклад: `df["horsepower"]=df["horsepower"].astype(int, copy=True)` – перетворення всіх значень стовпця в тип `int`;

Індикаторна змінна (або фіктивна змінна) – це числова змінна, яка використовується для позначення категорій. Їх називають *dummies*, тому що самі числа не мають внутрішнього значення. Індикаторні змінні використовуються при роботі з категоріальними змінними для регресійного аналізу в наступних модулях.

Дані, що підлягають аналізу, вимагають різного переліку дій попередньої обробки. Цей процес може включати як всі перераховані кроки, так і будь-яку їх комбінацію в довільній кількості в залежності від структури даних в рамках структурованого набору даних (датасету).

Датасет – (від англ. *dataset*) структурований набір даних, який використовується для аналізу, машинного навчання, побудови графіків і візуалізацій та ін. Завантажений датасет називається **DataFrame**.

2 ПІДГОТОВКА ДАНИХ ДО АНАЛІЗУ

Перед початком роботи з даними для аналізу необхідно підготувати середовище: при автономній роботі на ПК встановити всі необхідні бібліотеки; при роботі в онлайн-середовищі – підготувати відповідну вкладку.

2.1 Завантаження даних

Для завантаження бази даних можна скористатися командою бібліотеки Pandas: `.read_csv`. Для цього спочатку завантажуються вказана бібліотека. Також для зручності подальших маніпуляцій зразу під'єднується бібліотека NumPy. Лістинг для завантаження даних наступний:

```
#завантаження бібліотеки Pandas з псевдонімом pd та
#бібліотеки NumPy з псевдонімом np
import pandas as pd
import numpy as np

# задається ім'я файлу з повним шляхом до нього
filename = "D:\MCAD\Projects\datasets\FileName.csv"

df = pd.read_csv(filename)
```

Команда `.read_csv` має параметр `names`, який є необов'язковим. Формат даного параметру – кортеж, який містить назв стовпців таблиці. За замовчуванням береться перший рядок таблиці, що завантажуються. Прототип виклику функції представлений наступною командою:

```
pd.read_csv(filename, names = headers)
```

Таким чином, параметр `names` можна використовувати для визначення нових назв стовпців при завантаженні датасету.

За замовчуванням роздільником для стовпців в csv-файлі вважається кома. Якщо в якості роздільника використано інший знак (наприклад, крапка з комою «;»), команда зчитування доповнюється наступним аргументом, наведеним на лістингу:

```
df = pd.read_csv(filename, sep=';')
```

Для перевірки коректності зчитування файлу виведемо перші 10 рядків завантаженої таблиці. Для цього скористаємося командою:

```
print(df.head(10))
```

В результаті отримаємо перелік рядків таблиці, приклад якої наведений на рисунку 1.2.

	Customer Number	Customer Name	2016	2017	Percent Growth	Jan Units	Month	Day	Year	Active
0	10002.0	Quest Industries	\$125,000.00	\$162500.00	30.00%	500	1	10	2015	Y
1	552278.0	Smith Plumbing	\$920,000.00	\$101,2000.00	10.00%	700	6	15	2014	Y
2	23477.0	ACME Industrial	\$50,000.00	\$62500.00	25.00%	125	3	29	2016	Y
3	24900.0	Brekke LTD	\$350,000.00	\$490000.00	4.00%	75	10	27	2015	Y
4	651029.0	Harbor Co	\$15,000.00	\$12750.00	-15.00%	Closed	2	2	2014	N
5	215460.0	Kudos Concepts	\$125,000.00	\$162500.00	30.00%	500	9	17	2012	Y
6	458214.0	ITI Fiserv	\$920,000.00	\$101,2000.00	?	?	3	28	2013	Y
7	265482.0	Big Pond	\$50,000.00	\$62500.00	25.00%	125	8	18	2014	Y
8	256327.0	IntelliMax Solutions	\$350,000.00	\$490000.00	4.00%	?	2	15	2016	?
9	785462.0	LEK communications	\$15,000.00	\$12750.00	-15.00%	Closed	7	4	2013	N

Рисунок 1.2 – Приклад завантаженого DataFrame

Із опису до вхідних даних відомо, що стовпці надають наступну інформацію:

- Customer Number – номер клієнта в базі даних;
- Customer Name – ім'я клієнта;
- 2016 та 2017 – прибутки клієнта в 2016 та 2017 роках відповідно;

- Percent Growth – відсоток зростання прибутку клієнта в 2017 р. відносно 2016-го;

- Jan Units – сума замовлених послуг клієнта;

- Month, Day, Year – дата реєстрації клієнта;

- Active – активність клієнта.

Для оцінки якості завантажених даних можна вивести загальну інформацію про них командою:

```
print(df.info())
```

В результаті отримуємо наступні дані, наведені на рисунку 1.3.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18 entries, 0 to 17
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Customer Number       18 non-null    float64
1   Customer Name         18 non-null    object
2   2016                  18 non-null    object
3   2017                  18 non-null    object
4   Percent Growth        18 non-null    object
5   Jan Units             18 non-null    object
6   Month                 18 non-null    int64
7   Day                   18 non-null    int64
8   Year                  18 non-null    int64
9   Active                18 non-null    object
dtypes: float64(1), int64(3), object(6)
memory usage: 1.5+ KB
```

Рисунок 1.3 – Інформація про DataFrame

В даному інформаційному блоці зазначається, що загальний розмір таблиці – 18 рядків з нумерацією від 0 до 17. Таблиця містить 9 стовпців зі своїми назвами. В кожній комірці таблиці існує якийсь значення, тобто таблиця не містить порожніх комірок. В стовпці за номером 0 інформація зберігається в форматі з плаваючою крапкою (float64), в стовпцях 6, 7, 8 зберігаються дані цілочисельного типу

(int64), в інших стовпцях – як об’єкти (найбільш близький аналог – рядок str). Загальний розмір файлу – 1,5 KB.

Проведемо аналіз отриманих даних на основі інформації, наведеної на рисунках 1.2-1.3:

- вхідні дані містять невизначені параметри, в даному випадку позначені «?»;
- вхідні дані завантажені в незручних форматах, що значно ускладнить роботу з ними;
- необхідно перевірити вхідні дані на наявність повторів.

Примітка:

У випадку роботи з неповними даними можливе отримання інформаційного повідомлення вигляду, наведеному на рисунку 1.4.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 65079 entries, 0 to 65078
Data columns (total 7 columns):
  userID      65079 non-null object
Track        63848 non-null object
artist 1      57876 non-null object
genre        63881 non-null object
City 2        65079 non-null object
time         65079 non-null object
Day          65079 non-null object
dtypes: object(7)
memory usage: 3.5+ MB
```

- 1) некоректна назва стовпчиків (червоний маркер);
 - 2) дані є неповними: існують порожні комірки (синій маркер)
- Рисунок 1.4 – Інформація про DataFrame з неповними даними

З тексту видно, що є наступні неточності у вхідних даних:

- в назвах стовпців є зайві символи (пробіли), які варто прибрати (1 – червоний маркер);
- в таблиці є комірки, які не містять жодної інформації, тобто вони порожні (2 – синій маркер). Це видно з того, що загальна кількість

рядків в датасеті прописана в `RangeIndex`, а кількість заповнених комірок для певного стовпчика – безпосередньо у відповідному рядку.

2.2 Обробка невизначених даних

За результатами аналізу вхідного тексту видно, що замість невизначених комірок наведено певний символ «?». Він не дає сприйняти дану комірку як порожню. Тому варто його замінити на відсутність даних `NaN`. Приклад для реалізації цього процесу має наступний вигляд:

```
df.replace("?", np.nan, inplace = True)
```

2.2.1 Виявлення невизначених даних

Для перевірки наявності відсутніх даних можна скористатися командами:

- `.isnull()` – повертає `True`, якщо в комірці стоїть `NaN`;
- `.notnull()` – повертає `True`, якщо в комірці записані дані.

Створимо змінну, яка буде містити інформацію про наявні порожні комірки. Для цього можна скористатися командою:

```
missing_data = df.isnull()
```

Для підрахунку кількості порожніх комірок, тобто пропущених значень, в таблиці можна скористатися циклом `for`: "`True`" представляє відсутнє значення, а "`False`" означає, що значення присутнє в наборі даних. У тілі циклу `for` метод `".value_counts()"` підраховує кількість існуючих значень (в даному випадку кількість "`True`"). Програмна реалізація цієї задачі наведена на лістингу:

```
for column in missing_data.columns.values.tolist():
    print (missing_data[column].value_counts())
    print("")
```

В результаті отримуємо інформацію про DataFrame, наведену на рисунку 1.4.

Видно, що відсутні дані спостерігаються в стовпцях Percent Growth, Jan Units, Active – кількість True відмінна від 0. В інших стовпцях всі комірки заповнені (по 18 значень з 18 існуючих).

```
Customer Number
False      18
Name: count, dtype: int64

Customer Name
False      18
Name: count, dtype: int64

2016
False      18
Name: count, dtype: int64

2017
False      18
Name: count, dtype: int64

Percent Growth
False      16
True         2
Name: count, dtype: int64

Jan Units
False      15
True         3
Name: count, dtype: int64

Month
False      18
Name: count, dtype: int64

Day
False      18
Name: count, dtype: int64

Year
False      18
Name: count, dtype: int64

Active
False      16
True         2
Name: count, dtype: int64
```

Рисунок 1.4 – Перевірка на наявність та відсутність даних в DataFrame

2.2.2 Дії з невизначеними даними.

Відсутні, або невизначені, дані можна обробити наступним чином:

- а) видалити дані:
 - 1) видалення всього рядка;
 - 2) видалення всієї колонки;
- б) замінити дані:
 - 1) заміна відсутніх даних на середнє значення;
 - 2) заміна відсутніх даних за частотою;
 - 3) заміна відсутніх даних за допомогою інших функцій.

Цілі стовпці слід вилучати, лише якщо більшість записів у стовпці порожні. В іншому випадку доцільно провести заповнення комірок за деякою логікою.

2.3 Виявлення проблемних місць і постановка завдання

За результатами проведеного аналізу визначаємо перелік необхідних дій над вхідними даними:

- змінити типи даних на зручні;
- змінити невизначені дані;
- перевірити на наявність повторів.

Необхідні дії по стовпцям:

- а) стовпець 'Customer Number':
 - 1) змінити тип даних на int;
- б) стовпці '2016' and '2017':
 - 1) очистити від зайвих символів;
 - 2) змінити тип даних на float;
- в) стовпець 'Percent Growth':
 - 1) змінити тип даних на float;
 - 2) доповнити невисначаючі дані - частка збільшення обороту значення в стовпці 2017 відносно значення стовпця 2016;
- г) стовпець 'Jan Units':
 - 1) змінити тип даних на float;

- 2) доповнити невивантаючі дані - середнє значення по стовпця для відсутніх даних, 0 - для тексту;
- д) стовпці 'Month', 'Day', 'Year':
 - 1) об'єднати вміст на змінити тип даних на `datetime`;
- е) стовпець 'Active':
 - 1) змінити тип даних на `bool`;
 - 2) замінити дані: якщо Y, то 1, інакше – 0.

2.4 Робота зі стовпцями

Виконаємо заплановані перетворення за кожним стовпцем.

2.4.1 Стовпець 'Customer Number'

В даному стовпці необхідно змінити формат даних з `float` на `int`.

Для перетворення типу даних в `pandas` існує 3 основних способи:

- використати метод `astype()`, щоб примусово задати новий тип;
- створити спеціалізовану користувацьку функцію для перетворення даних;
- використати функції `to_numeric()` або `to_datetime()`.

В даному випадку, зважаючи на дані, що зберігаються в стовпці, доцільно використати перший спосіб. В цьому випадку блок коду матиме наступний вигляд, наведений на лістингу 2.1.

Лістинг 2.1 – Зміна форматів даних в стовпці

```
df["Customer Number"] = df['Customer
    Number'].astype('int')
print(df.dtypes)    #перевірка типів даних в
    стовпцях
```

В результаті отримуємо інформаційне повідомлення, наведене на рисунку 1.5.

Перетворення типу даних першого стовпця пройшло успішно.

	o
Customer Number	int64
Customer Name	object
2016	object
2017	object
Percent Growth	object
Jan Units	object
Month	int64
Day	int64
Year	int64
Active	object

Рисунок 1.5 – Інформація про поточні типи даних в DataFrame

2.4.2 Стовпці '2016' та '2017'

Дані, що зберігаються в зазначених стовпцях, окрім допустимих крапок, в якості роздільника цілої та дробової частин, містять коми та спеціальний символ долара, що унеможливило пряме приведення типів. Тому в даному випадку доцільно розробити користувацьку функцію, яка буде очищувати дані та приводити їх до типу float. Код функції наведений на лістингу 2.2.

Лістинг 2.2 – Функція для очищення та конвертації даних

```
def convert_currency(val):
    # Перетворення числового значення рядка
    # в число з плаваючою крапкою:
    # - видаляє $;
    # - видаляє кому;
    # - перетворює в число з плаваючою крапкою.
    new_val = val.replace(',', ' ').replace('$', ' ')
    return float(new_val)

# застосування розробленої функції
df['2016'] = df['2016'].apply(convert_currency)
df['2017'] = df['2017'].apply(convert_currency)
```

```
# перевіряємо результат
print(df.dtypes)
print(df.head(10))
```

В результаті виконання даного програмного блоку отримано результати, наведені на рисунку 1.6.

a)		b)										
	0	Customer Number	Customer Name	2016	2017	Percent Growth	Jan Units	Month	Day	Year	Active	
Customer Number	int64	0	10002	Quest Industries	125000.0	162500.0	30.00%	500	1	10	2015	Y
Customer Name	object	1	552278	Smith Plumbing	920000.0	1012000.0	10.00%	700	6	15	2014	Y
2016	float64	2	23477	ACME Industrial	50000.0	62500.0	25.00%	125	3	29	2016	Y
2017	float64	3	24900	Brekke LTD	350000.0	490000.0	4.00%	75	10	27	2015	Y
Percent Growth	object	4	651029	Harbor Co	15000.0	12750.0	-15.00%	Closed	2	2	2014	N
Jan Units	object	5	215460	Kudos Concepts	125000.0	162500.0	30.00%	500	9	17	2012	Y
Month	int64	6	458214	ITI Fiserv	920000.0	1012000.0	NaN	NaN	3	28	2013	Y
Day	int64	7	265482	Big Pond	50000.0	62500.0	25.00%	125	8	18	2014	Y
Year	int64	8	256327	IntelliMax Solutions	350000.0	490000.0	4.00%	NaN	2	15	2016	NaN
Active	object	9	785462	LEK communications	15000.0	12750.0	-15.00%	Closed	7	4	2013	N

a)

b)

а) типи даних в стовпцях; б) вміст стовпців

Рисунок 1.6 – DataFrame після перетворення стовпців '2016' та '2017'

2.4.3 Стовпець 'Percent Growth'

В даному стовпці існує зайвий символ «%», а також є порожні дані, позначені як NaN. При безпосередній спробі привести дані даного стовпця до типу 'float' викидається помилка: саме через наявність NaN. Комірку з NaN не можна просто замінити на якесь число або текст з тієї ж причини. Тому в даному випадку вирішено замінити такі дані на числове значення, яке однозначно не використовується і повернути для комірки тип str (інший варіант – використати для досягнення даної мети команду .replace()). Приклад даного процесу наведений на лістингу 2.3.

Лістинг 2.3 – Приклад роботи з невизначеними даними

```
# Заміна комірок з NaN на значення 0
# та повернення до типу str
# для одноманітності даних в стовці
df["Percent Growth"] = df["Percent
    Growth"].fillna(0).astype('str')
```

Далі приводимо дані стовпця до нового типу з паралельним очищенням від зайвих символів і нормалізацією результату (ділимо відповідь на 100). Для цього використовуємо лямбда-функцію (лістинг 2.4).

Лістинг 2.4 – Приклад очищення та конвертації даних

```
# Приведення до типу float
df['Percent Growth'] = df['Percent
    Growth'].apply(lambda x: x.replace('%',
    '')).astype('float')/100
print(df.dtypes)
```

Після приведення даних до нового типу пропущені комірки можна доповнити значеннями. В даному випадку значення визначається як різниця між 1 та відношенням прибутку за 2016 рік до прибутку за 2017 рік (лістинг 2.5).

Лістинг 2.5 – Приклад заповнення пропущених даних

```
# частка збільшення обороту значення
# в стовпці 2017 відносно значення стовпця 2016
df['Percent Growth'] = df.loc[df['Percent Growth']
    == 0.0, 'Percent Growth'] = 1-
    df['2016']/df['2017']
```

В результаті отримусмо таблицю даних, наведену на рисунку 1.7.

	0		Customer Number	Customer Name	2016	2017	Percent Growth	Jan Units	Month	Day	Year	Active
Customer Number	int64	0	10002	Quest Industries	125000.0	162500.0	0.230769	500	1	10	2015	Y
Customer Name	object	1	552278	Smith Plumbing	920000.0	1012000.0	0.090909	700	6	15	2014	Y
2016	float64	2	23477	ACME Industrial	50000.0	62500.0	0.200000	125	3	29	2016	Y
2017	float64	3	24900	Brekke LTD	350000.0	490000.0	0.285714	75	10	27	2015	Y
Percent Growth	float64	4	651029	Harbor Co	15000.0	12750.0	-0.176471	Closed	2	2	2014	N
Jan Units	object	5	215460	Kudos Concepts	125000.0	162500.0	0.230769	500	9	17	2012	Y
Month	int64	6	458214	ITI Fiserv	920000.0	1012000.0	0.090909	NaN	3	28	2013	Y
Day	int64	7	265482	Big Pond	50000.0	62500.0	0.200000	125	8	18	2014	Y
Year	int64	8	256327	IntelliMax Solutions	350000.0	490000.0	0.285714	NaN	2	15	2016	NaN
Active	object	9	785462	LEK communications	15000.0	12750.0	-0.176471	Closed	7	4	2013	N

а)

б)

а) типи даних в стовпцях; б) вміст стовпців

Рисунок 1.7 – DataFrame після перетворення стовпця 'Percent Growth'

Приведення і доповнення даних пройшло успішно.

2.4.4 Стовпець 'Jan Units'

Інформація, що надається в цьому стовпці, також неоднорідна: в ньому містяться числа, рядок, а також є невизначена комірка. Тому послідовність дій буде аналогічна діям з попереднім стовпцем (лістинг 2.6).

Лістинг 2.6 – Приклад заміни невизначених даних

```
# замінимо значення комірок, які необхідно
# доповнити, заздалегідь унікальним значенням
df["Jan Units"].replace(np.nan, '-1', inplace =
    True)
```

Перетворення змісту стовпця на числа з паралельною заміною текстового рядка на нуль (0) виконується за допомогою команди `.to_numeric()` (лістинг 2.7).

Лістинг 2.7 – Приклад перетворення рядка на числа

```
# заміна значень (рядка) на числа,
# якщо значення не є числом – на Nan
#.fillna(0) – заміна Nan на 0
df["Jan Units"] = pd.to_numeric(df['Jan Units'],
                                errors='coerce')#.fillna(0)
print(df.head(10))
```

Заповнити пропущені дані вирішено середнім значенням по стовпцю. Для цього спочатку знаходимо це значення, а потім підставляємо в необхідні комірки (лістинг 2.8).

Лістинг 2.8 – Заповнення пропущених значень середнім

```
#знаходимо середнє значення по стовпцю
avg_norm_loss = df['Jan
                    Units'].astype('float').mean(axis=0)
print("Average of Jan Units:", avg_norm_loss)
# Підставляємо середнє значення в комірки з "-1"
df.loc[df["Jan Units"] == -1, "Jan Units"] =
    avg_norm_loss
```

З метою, щоб в даних не було порожніх комірок і для спрощення подальшого аналізу, у відповідності зі змістом вхідних даних, прийнято рішення реалізувати заміну вмісту комірок, в яких були рядки “Closed” (замінені зараз на Nan), на значення «0» (лістинг 2.9)

Лістинг 2.9 – Заміна невизначених даних на «0»

```
# замінюємо nan на 0
df["Jan Units"].replace(np.nan, 0, inplace=True)
print(df.head(10))
```

В результаті отримуємо вигляд таблиці, наведений на рисунку 1.8.

	Customer Number	Customer Name	2016	2017	Percent Growth	Jan Units	Month	Day	Year	Active
0	10002	Quest Industries	125000.0	162500.0	0.230769	500.0	1	10	2015	Y
1	552278	Smith Plumbing	920000.0	1012000.0	0.090909	700.0	6	15	2014	Y
2	23477	ACME Industrial	50000.0	62500.0	0.200000	125.0	3	29	2016	Y
3	24900	Brekke LTD	350000.0	490000.0	0.285714	75.0	10	27	2015	Y
4	651029	Harbor Co	15000.0	12750.0	-0.176471	0.0	2	2	2014	N
5	215460	Kudos Concepts	125000.0	162500.0	0.230769	500.0	9	17	2012	Y
6	458214	ITI Fiserv	920000.0	1012000.0	0.090909	269.8	3	28	2013	Y
7	265482	Big Pond	50000.0	62500.0	0.200000	125.0	8	18	2014	Y
8	256327	IntelliMax Solutions	350000.0	490000.0	0.285714	269.8	2	15	2016	NaN
9	785462	LEK communications	15000.0	12750.0	-0.176471	0.0	7	4	2013	N

Рисунок 1.8 – DataFrame після перетворення стовпця 'Jan Units'

2.4.5 Стовпці 'Month', 'Day', 'Year'

Дані, наведені в цих стовпцях, представляють собою єдину інформацію. На зараз змінні для їх збереження мають тип *int* і розміщені в трьох окремих стовпчиках. Тому доцільно їх поєднати в одну спеціалізованого типу *datetime*, призначеному для зберігання дати та часу. Отримані дані додаються останнім додатковим стовпцем в загальну таблицю (лістинг 2.10)

Лістинг 2.10 – Об'єднання стовпців з датами

```
# Створення нового стовпця з датою типу datetime64
# і додавання його в кінець таблиці
df["Start_Date"] = pd.to_datetime(df[['Month',
    'Day', 'Year']])
print(df.head(10))
```

В результаті отримуємо вигляд таблиці, наведений на рисунку 1.9.

	Customer Number	Customer Name	2016	2017	Percent Growth	Jan Units	Month	Day	Year	Active	Start_Date
0	10002	Quest Industries	125000.0	162500.0	0.230769	500.0	1	10	2015	True	2015-01-10
1	552278	Smith Plumbing	920000.0	1012000.0	0.090909	700.0	6	15	2014	True	2014-06-15
2	23477	ACME Industrial	50000.0	62500.0	0.200000	125.0	3	29	2016	True	2016-03-29
3	24900	Brekke LTD	350000.0	490000.0	0.285714	75.0	10	27	2015	True	2015-10-27
4	651029	Harbor Co	15000.0	12750.0	-0.176471	0.0	2	2	2014	False	2014-02-02
5	215460	Kudos Concepts	125000.0	162500.0	0.230769	500.0	9	17	2012	True	2012-09-17
6	458214	ITI Fiserv	920000.0	1012000.0	0.090909	269.8	3	28	2013	True	2013-03-28
7	265482	Big Pond	50000.0	62500.0	0.200000	125.0	8	18	2014	True	2014-08-18
8	256327	IntelliMax Solutions	350000.0	490000.0	0.285714	269.8	2	15	2016	False	2016-02-15
9	785462	LEK communications	15000.0	12750.0	-0.176471	0.0	7	4	2013	False	2013-07-04

Рисунок 1.9 – DataFrame після перетворення стовпців 'Month', 'Day', 'Year'

Таким чином, в датасеті з'явився додатковий стовпчик з повною датою події, що дозволяє в подальшому виключити стовпці 'Month', 'Day', 'Year' з аналізу і при цьому спростити алгоритми обробки інформації, яка там міститься, за допомогою стандартних вбудованих інструментів.

2.4.6 Стовпець 'Active'

Інформація в даному стовпці має тип object і містить значення «Y», «N», а також порожню комірку. Доцільно ці дані представити в булевому типі з одночасною заміною Y на 1, N на 0, а також порожню комірку заповнити значенням 0. Для цього скористаємося механізмом `.map()` (лістинг 2.11).

Лістинг 2.11 – Перетворення даних в булевий формат

```
Active_map = {'Y': 1, 'N': 0, np.nan: 0}
df['Active'] =
    df['Active'].map(Active_map).astype('int')
print(df.head(10))
```

В результаті отримуємо (рис.1.10).

	Customer Number	Customer Name	2016	2017	Percent Growth	Jan Units	Month	Day	Year	Active
0	10002	Quest Industries	125000.0	162500.0	0.230769	500.0	1	10	2015	True
1	552278	Smith Plumbing	920000.0	1012000.0	0.090909	700.0	6	15	2014	True
2	23477	ACME Industrial	50000.0	62500.0	0.200000	125.0	3	29	2016	True
3	24900	Brekke LTD	350000.0	490000.0	0.285714	75.0	10	27	2015	True
4	651029	Harbor Co	15000.0	12750.0	-0.176471	0.0	2	2	2014	False
5	215460	Kudos Concepts	125000.0	162500.0	0.230769	500.0	9	17	2012	True
6	458214	ITI Fiserv	920000.0	1012000.0	0.090909	269.8	3	28	2013	True
7	265482	Big Pond	50000.0	62500.0	0.200000	125.0	8	18	2014	True
8	256327	IntelliMax Solutions	350000.0	490000.0	0.285714	269.8	2	15	2016	False
9	785462	LEK communications	15000.0	12750.0	-0.176471	0.0	7	4	2013	False

Рисунок 1.10 – DataFrame після перетворення стовпця 'Active'

В результаті проведених перетворень дані в DataFrame представлені в зручному для подальшої роботи вигляді.

2.5 Повторна перевірка на невизначеність даних

Після перетворення даних проведемо повторну перевірку на наявність незаповнених комірок (лістинг 2.12).

Лістинг 2.12 – Перевірка на наявність незаповнених комірок

```
#Перевіряємо кількість невизначених даних
missing_data = df.isnull()
missing_data.head(10)

for column in missing_data.columns.values.tolist():
    #print(column)
    print (missing_data[column].value_counts())
    print("")
```

Отримуємо інформацію, наведену на рисунку 1.11.

```

Customer Number
False      18
Name: count, dtype: int64

Customer Name
False      18
Name: count, dtype: int64

2016
False      18
Name: count, dtype: int64

2017
False      18
Name: count, dtype: int64

Percent Growth
False      18
Name: count, dtype: int64

Jan Units
False      18
Name: count, dtype: int64

Month
False      18
Name: count, dtype: int64

Day
False      18
Name: count, dtype: int64

Year
False      18
Name: count, dtype: int64

Active
False      18
Name: count, dtype: int64

```

Рисунок 1.11 – Перевірка на наявність та відсутність даних в DataFrame після підготовки даних

Бачимо, що рядків з True в жодному із описів стовпців немає, тобто всі комірки таблиці є заповненими.

2.6 Виявлення повторів

Далі проведемо перевірку таблиці на наявність повторів рядків за допомогою функції `.duplicated()`. Для зручності зразу підраховується загальна кількість таких повторів за допомогою команди:

```

# Перевірка на наявність повторів
print(df.duplicated().sum())

```

Видалення дублікатів відбувається за допомогою функції `df.drop_duplicates()` зі скиданням старих індексів після видалення рядків

Остаточна таблиця буде мати наступний вигляд (рис. 1.12).

	Customer Number	Customer Name	2016y	2017y	Percent Growth	Jan Units	Month	Day	Year	Active	Start_Date
0	10002	Quest Industries	125000.0	162500.0	0.230769	500.0	1	10	2015	True	2015-01-10
1	552278	Smith Plumbing	920000.0	1012000.0	0.090909	700.0	6	15	2014	True	2014-06-15
2	23477	ACME Industrial	50000.0	62500.0	0.200000	125.0	3	29	2016	True	2016-03-29
3	24900	Brekke LTD	350000.0	490000.0	0.285714	75.0	10	27	2015	True	2015-10-27
4	651029	Harbor Co	15000.0	12750.0	-0.176471	0.0	2	2	2014	False	2014-02-02
5	215460	Kudos Concepts	125000.0	162500.0	0.230769	500.0	9	17	2012	True	2012-09-17
6	458214	ITI Fiserv	920000.0	1012000.0	0.090909	269.8	3	28	2013	True	2013-03-28
7	265482	Big Pond	50000.0	62500.0	0.200000	125.0	8	18	2014	True	2014-08-18
8	256327	IntelliMax Solutions	350000.0	490000.0	0.285714	269.8	2	15	2016	False	2016-02-15
9	785462	LEK communications	15000.0	12750.0	-0.176471	0.0	7	4	2013	False	2013-07-04

Рисунок 1.12 – Результуючий вигляд DataFrame після підготовки даних

Отримані дані готові для подальшого аналізу.

3 ЛАБОРАТОРНЕ ЗАВДАННЯ

Виконати підготовку вхідних даних з наборів датасетів у відповідності з варіантом, наведеним в таблиці 3.1, або обрати будь-який довільний датасет по узгодженню з викладачем.

Для підготовки даних необхідно виконати їх перевірку та обробку з урахуванням їх характеристик та призначення наведеної там інформації. До основних критеріїв, які необхідно розглянути, відносяться:

- наявність невизначених або некоректних значень;
- виявлення відсутніх (пропущених) даних;
- відповідність типів даних наведених там інформації;
- виявлення дубльованих записів.

При необхідності перелік критеріїв розширюється з урахуванням особливостей датасету, що аналізується.

Опис вхідних даних датасетів наведений в Додатку А.

Таблиця 3.1 – Завдання за варіантами

Вар.	Назва dataset	Назва файлу
1	Economics Journal Subscription Data	Journals.csv
2	Low birth weight data	lbw.csv
3	Labour Training Evaluation Data	nsw74psidA.csv
4	Economic Journals Data Set	Journals(1).csv
5	Neurocognitive Measures in Psychiatric Groups	NeuroCog.csv
6	Risk Factors Associated with Low Infant Birth Weight	Birthwt.csv
7	Measurements of Forensic Glass Fragments	fgl.csv
8	First Year GPA for College Students	FirstYearGPA.csv
9	NCCTG Lung Cancer Data	cancer.csv
10	Affairs	Affairs.csv

4 ЗМІСТ ЗВІТУ

Звіт з лабораторної роботи може бути представлений у вигляді документу, підготовленому за допомогою текстового редактору, або документу з розширенням .ipynb (файл / завантажити / завантажити ipynb для середовища Google Colaboratory).

Звіт має містити:

- назва роботи;
- виконавець роботи;
- мета роботи;
- хід роботи;
- результати перевірки та обробки вхідних даних;
- висновки;
- відповіді на контрольні питання.

5 КОНТРОЛЬНІ ПИТАННЯ

1. Що таке стандартизація?
2. Що таке нормалізація?
3. Якими способами можна позбутись відсутності даних?
4. Що таке бінінг?
5. Як нормалізувати дані?

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Офіційний сайт Python. URL: <https://www.python.org/>
2. Google Colaboratory. URL: <https://colab.research.google.com/>
3. Pandas API reference. URL:
<https://pandas.pydata.org/docs/reference/index.html>
4. Free datasets. URL:
<https://vincentarelbundock.github.io/Rdatasets/datasets.html>

ДОДАТОК А ОПИС DATASETS

A.1 Economics Journal Subscription Data

Description

Subscriptions to economics journals at US libraries, for the year 2000.

Usage – Journals.csv

Format

A data frame containing 180 observations on 10 variables.

title - Journal title.

publisher - factor with publisher name.

society - factor. Is the journal published by a scholarly society?

price - Library subscription price.

pages - Number of pages.

charpp - Characters per page.

citations - Total number of citations.

foundingyear - Year journal was founded.

subs - Number of library subscriptions.

field - factor with field description.

Details - Data on 180 economic journals, collected in particular for analyzing journal pricing. See also <https://econ.ucsb.edu/~tedb/Journals/jpricing.html> for general information on this topic as well as a more up-to-date version of the data set. This version is taken from Stock and Watson (2007).

The data as obtained from the online complements for Stock and Watson (2007) contained two journals with title “World Development”. One of these (observation 80) seemed to be an error and was changed to “The World Economy”.

Source - Online complements to Stock and Watson (2007).

References

Bergstrom, T. (2001). Free Labor for Costly Journals? *Journal of Economic Perspectives*, 15, 183–198.

Stock, J.H. and Watson, M.W. (2007). Introduction to Econometrics, 2nd ed. Boston: Addison Wesley.

A.2 Low birth weight data

Description - The data come to us from Hosmer and Lemeshow (2000). Called the low birth weight (lbw) data, the response is a binary variable, low, which indicates whether the birth weight of a baby is under 2500g (low=1), or over (low=0).

Usage – lbw.csv

Format - A data frame with 189 observations on the following 10 variables.

low - 1=low birthweight baby; 0=normal weight

smoke - 1=history of mother smoking; 0=mother nonsmoker

race - categorical 1-3: 1=white; 2=black; 3=other

age - age of mother: 14-45

lwt - weight (lbs) at last menstrual period: 80-250 lbs

ptl - number of false of premature labors: 0-3

ht - 1=history of hypertension; 0=no hypertension

ui - 1=uterine irritability; 0 no irritability

ftv - number of physician visits in 1st trimester: 0-6

bwt - birth weight in grams: 709 - 4990 gr

Details

lbw is saved as a data frame. Count models can use ftv as a response variable, or convert it to grouped format

Source

Hosmer, D and S. Lemeshow (2000), Applied Logistic Regression, Wiley

References

Hilbe, Joseph M (2007, 2011), Negative Binomial Regression, Cambridge University Press
Hilbe, Joseph M (2009), Logistic Regression Models, Chapman & Hall/CRC

A.3 Labour Training Evaluation Data

Description

This nsw74demo data frame, with 445 rows and 10 columns, is the subset of the nswdemo dataset for which 1974 earnings are available. Data are for the male experimental control and treatment groups, in an investigation of the effect of training on changes, between 1974-1975 and 1978, in the earnings of individuals who had experienced employment difficulties.

Likewise, nsw74psid1 (2675 rows) is the subset of the nswpsid1 data, and nsw74psid3 (313 rows) is the subset of the nswpsid3 data, for which 1974 income is available. NB, also, the nsw74psidA data set.

Usage - nsw74psidA.csv

Format

Columns are:

trt - a numeric vector identifying the study in which the subjects were enrolled (0 = PSID, 1 = NSW).

age - age (in years).

educ - years of education.

black - (0 = not black, 1 = black).

hisp - (0 = not hispanic, 1 = hispanic).

marr - (0 = not married, 1 = married).

nodeg - (0 = completed high school, 1 = dropout).

re74 - real earnings in 1974.

re75 - real earnings in 1975.

re78 - real earnings in 1978.

Details

The nsw74psidA data set (252 rows) was obtained from nsw74psid1 using:

```
here <- age <= 40 & re74 <= 5000 & re75 <= 5000 & re78 < 30000
nsw74psidA <- nsw74psid1[here, ]
```

References

Dehejia, R.H. and Wahba, S. 1999. Causal effects in non-experimental studies: re-evaluating the evaluation of training programs. *Journal of the American Statistical Association* 94: 1053-1062.

Lalonde, R. 1986. Evaluating the economic evaluations of training programs. *American Economic Review* 76: 604-620.

A.4 Economic Journals Data Set

Description - a cross-section from 2000

number of observations : 180

observation : goods

Usage - Journals(1).csv

Format

A dataframe containing :

title - journal title

pub - publisher

society - scholarly society ?

libprice - library subscription price

pages - number of pages

charpp - characters per page

citestot - total number of citations

date1 - year journal was founded

oclc - number of library subscriptions

field - field description

Source

Professor Theodore Bergstrom of the Department of Economics at the University of California, San Diego.

References

Stock, James H. and Mark W. Watson (2003) *Introduction to Econometrics*, Addison-Wesley Educational Publishers, chapter 6.

A.5 Neurocognitive Measures in Psychiatric Groups

Description - The primary purpose of the study (Hartman, 2016, Heinrichs et al. (2015)) was to evaluate patterns and levels of performance

on neurocognitive measures among individuals with schizophrenia and schizoaffective disorder using a well-validated, comprehensive neurocognitive battery specifically designed for individuals with psychosis (Heinrichs et al. (2008))

Usage - NeuroCog.csv

Format - A data frame with 242 observations on the following 10 variables.

Dx - Diagnostic group, a factor with levels Schizophrenia Schizoaffective Control

Speed - Speed of processing domain T score, a numeric vector

Attention - Attention/Vigilance Domain T score, a numeric vector

Memory - Working memory a numeric vector

Verbal - Verbal Learning Domain T score, a numeric vector

Visual - Visual Learning Domain T score, a numeric vector

ProbSolv - Reasoning/Problem Solving Domain T score, a numeric vector

SocialCog - Social Cognition Domain T score, a numeric vector

Age - Subject age, a numeric vector

Sex - Subject gender, a factor with levels Female Male

Details

The main interest was in determining how well these measures distinguished among all groups and whether there were variables that distinguished between the schizophrenia and schizoaffective groups.

Neurocognitive function was assessed using the MATRICS Consensus Cognitive Battery (MCCB; Nuechterlein et al., 2008). The MCCB consists of 10 individually administered tests that measure cognitive performance in seven domains: speed of processing, attention/vigilance, working memory, verbal learning, visual learning, reasoning and problem solving, and social cognition.

The clinical sample comprised 116 male and female patients who met the following criteria: 1) a diagnosis of schizophrenia (n = 70) or schizoaffective disorder (n = 46) confirmed by the Structured Clinical Interview for DSM-IV-TR Axis I Disorders; 2) outpatient status; 3) a history

free of developmental or learning disability; 4) age 18-65; 5) a history free of neurological or endocrine disorder; and 6) no concurrent DSM-IV-TR diagnosis of substance use disorder.

Non-psychiatric control participants (n = 146) were screened for medical and psychiatric illness and history of substance abuse. Patients were recruited from three outpatient clinics in Hamilton, Ontario, Canada. Control participants were recruited through local newspaper and online classified advertisements for paid research participation.

Source

Hartman, L. I. (2016). Schizophrenia and Schizoaffective Disorder: One Condition or Two? Unpublished PhD dissertation, York University.

Heinrichs, R.W., Pinnock, F., Muharib, E., Hartman, L.I., Goldberg, J.O., & McDermid Vaz, S. (2015). Neurocognitive normality in schizophrenia revisited. *Schizophrenia Research: Cognition*, 2 (4), 227-232. doi: 10.1016/j.scog.2015.09.001

References

Heinrichs, R. W., Ammari, N., McDermid Vaz, S. & Miles, A. (2008). Are schizophrenia and schizoaffective disorder neuropsychologically distinguishable? *Schizophrenia Research*, 99, 149-154.

Nuechterlein K.H., Green M.F., Kern R.S., Baade L.E., Barch D., Cohen J., Essock S., Fenton W.S., Frese F.J., Gold J.M., Goldberg T., Heaton R., Keefe R.S.E., Kraemer H., Mesholam-Gately R., Seidman L.J., Stover E., Weinberger D.R., Young A.S., Zalcman S., Marder S.R. (2008) The MATRICS Consensus Cognitive Battery, Part 1: Test selection, reliability, and validity. *American Journal of Psychiatry*, 165 (2), 203-213. <https://pubmed.ncbi.nlm.nih.gov/18172019/>.

A.6 Risk Factors Associated with Low Infant Birth Weight

Description - The birthwt data frame has 189 rows and 10 columns. The data were collected at Baystate Medical Center, Springfield, Mass during 1986.

Usage – birthwt.csv

Format - This data frame contains the following columns:

low - indicator of birth weight less than 2.5 kg.

age – mother's age in years.

lwt – mother's weight in pounds at last menstrual period.

race – mother's race (1 = white, 2 = black, 3 = other).

smoke - smoking status during pregnancy.

ptl - number of previous premature labours.

ht - history of hypertension.

ui - presence of uterine irritability.

ftv - number of physician visits during the first trimester.

bwt - birth weight in grams.

Source

Hosmer, D.W. and Lemeshow, S. (1989) Applied Logistic Regression.
New York: Wiley

References

Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics
with S. Fourth edition. Springer.

A.7 Measurements of Forensic Glass Fragments

Description

The fgl data frame has 214 rows and 10 columns. It was collected by
B. German on fragments of glass collected in forensic work.

Usage – fgl.csv

Format - This data frame contains the following columns:

RI - refractive index; more precisely the refractive index is 1.518xxxx.

The next 8 measurements are percentages by weight of oxides.

Na - sodium.

Mg - manganese.

Al - aluminium.

Si - silicon.

K - potassium.

Ca - calcium.

Ba - barium.

Fe - iron.

type - The fragments were originally classed into seven types, one of which was absent in this dataset. The categories which occur are window float glass (WinF: 70), window non-float glass (WinNF: 76), vehicle window glass (Veh: 17), containers (Con: 13), tableware (Tabl: 9) and vehicle headlamps (Head: 29).

References

Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

A.8 First Year GPA for College Students

Description - Predicting first-year college GPA

Usage – FirstYearGPA.csv

Format - A data frame with 219 observations on the following 10 variables.

GPA First-year college GPA on a 0.0 to 4.0 scale

HSGPA High school GPA on a 0.0 to 4.0 scale

SATV Verbal/critical reading SAT score

SATM Math SAT score

Male 1= male, 0= female

HU Number of credit hours earned in humanities courses in high school

SS Number of credit hours earned in social science courses in high school

FirstGen 1= student is the first in her or his family to attend college, 0=otherwise

White 1= white students, 0= others

CollegeBound 1=attended a high school where $\geq 50\%$ students intended to go on to college, 0=otherwise

Details

The data in FirstYearGPA contains information from a sample of 219 first year students at a midwestern college that might be used to build a model to predict their first year GPA.

Source

A sample from a larger set of data collected in 1996 by a professor at this college.

A.9 NCCTG Lung Cancer Data

Description - Survival in patients with advanced lung cancer from the North Central Cancer Treatment Group. Performance scores rate how well the patient can perform usual daily activities.

Usage - cancer.csv

data(cancer, package="survival")

Format

inst: Institution code

time: Survival time in days

status: censoring status 1=censored, 2=dead

age: Age in years

sex: Male=1 Female=2

ph.ecog: ECOG performance score as rated by the physician.
0=asymptomatic, 1= symptomatic but completely ambulatory, 2= in bed
<50% of the day, 3= in bed > 50% of the day but not bedbound, 4 = bedbound
ph.karno:Karnofsky performance score (bad=0-good=100) rated by
physician

pat.karno: Karnofsky performance score as rated by patient

meal.cal: Calories consumed at meals

wt.loss: Weight loss in last six months (pounds)

Note

The use of 1/2 for alive/dead instead of the usual 0/1 is a historical footnote. For data contained on punch cards, IBM 360 Fortran treated blank as a zero, which led to a policy within the section of Biostatistics to never use "0" as a data value since one could not distinguish it from a missing value. The policy became a habit, as is often the case; and the 1/2 coding endured long beyond the demise of punch cards and Fortran.

Source

Terry Therneau

References

Loprinzi CL. Laurie JA. Wieand HS. Krook JE. Novotny PJ. Kugler JW. Bartel J. Law M. Bateman M. Klatt NE. et al. Prospective evaluation of prognostic variables from patient-completed questionnaires. North Central Cancer Treatment Group. Journal of Clinical Oncology. 12(3):601-7, 1994.

A.10 Affairs

Fair's Extramarital Affairs Data

Description - Infidelity data, known as Fair's Affairs. Cross-section data from a survey conducted by Psychology Today in 1969.

Usage - "Affairs"

Format

A data frame containing 601 observations on 9 variables.

affairs - numeric. How often engaged in extramarital sexual intercourse during the past year? 0 = none, 1 = once, 2 = twice, 3 = 3 times, 7 = 4–10 times, 12 = monthly, 12 = weekly, 12 = daily.

gender - factor indicating gender.

age - numeric variable coding age in years: 17.5 = under 20, 22 = 20–24, 27 = 25–29, 32 = 30–34, 37 = 35–39, 42 = 40–44, 47 = 45–49, 52 = 50–54, 57 = 55 or over.

yearsmarried - numeric variable coding number of years married: 0.125 = 3 months or less, 0.417 = 4–6 months, 0.75 = 6 months–1 year, 1.5 = 1–2 years, 4 = 3–5 years, 7 = 6–8 years, 10 = 9–11 years, 15 = 12 or more years.

children - factor. Are there children in the marriage?

religiousness - numeric variable coding religiousness: 1 = anti, 2 = not at all, 3 = slightly, 4 = somewhat, 5 = very.

education - numeric variable coding level of education: 9 = grade school, 12 = high school graduate, 14 = some college, 16 = college graduate, 17 = some graduate work, 18 = master's degree, 20 = Ph.D., M.D., or other advanced degree.

occupation - numeric variable coding occupation according to Hollingshead classification (reverse numbering).

rating - numeric variable coding self rating of marriage: 1 = very unhappy, 2 = somewhat unhappy, 3 = average, 4 = happier than average, 5 = very happy.

Source. Online complements to Greene (2003). Table F22.2.

<https://pages.stern.nyu.edu/~wgreene/Text/tables/tablelist5.htm>

References

Greene, W.H. (2003). *Econometric Analysis*, 5th edition. Upper Saddle River, NJ: Prentice Hall.

Fair, R.C. (1978). A Theory of Extramarital Affairs. *Journal of Political Economy*, 86, 45–61.