

## **АНОТАЦІЯ**

Даний програмний продукт призначений для широкого кола користувачів. "CorrOK" представляє собою математичну програму для проведення кореляційного аналізу: на основі вибірок знаходяться коефіцієнти кореляції Pearson product-moment та Kendall's tau-b. В результаті роботи програми виводяться знайдені коефіцієнти кореляції та точкова діаграма, що показує залежність між двома випадковими величинами за Пірсоном.

Програма працює коректно та показала себе надійною в роботі.

Інтерфейс програми виконаний англійською мовою.

## **АННОТАЦИЯ**

Данный программный продукт предназначен для широкого круга пользователей. "CorrOK" представляет собой математическую программу для проведения корреляционного анализа: на основе выборок находятся коэффициенты корреляции Pearson product-moment и Kendall's tau-b. В результате работы программы выводятся найденные коэффициенты корреляции и точечная диаграмма, которая показывает зависимость между двумя случайными величинами по Пирсону.

Программа работает корректно и показала себя надежной в работе.

Интерфейс программы выполнен на английском языке.

## **ABSTRACT**

This software is designed for a wide range of users. "CorrOK" is a program for mathematical correlation analysis, based on samples. It calculates Pearson product-moment and Kendall's tau-b correlation coefficients. As the result, the program displays found correlation coefficients and scatter plot that shows the relationship between two random variables by Pearson.

The program works correctly and has proved reliability in operation.

The program interface is made in English.

## ЗМІСТ

ВСТУП.....	4
ПОСТАНОВКА ЗАДАЧІ .....	5
ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ .....	6
<b>1.1. Вступ .....</b>	<b>6</b>
<b>1.2. Кореляційний аналіз.....</b>	<b>6</b>
<b>1.3. Коефіцієнт кореляції.....</b>	<b>8</b>
<b>1.4. Точкова діаграма (кореляційне поле).....</b>	<b>9</b>
<b>1.5. Висновки до розділу.....</b>	<b>10</b>
ОПИС ПРОГРАМНОГО ПРОДУКТУ .....	11
<b>2.1. Вступ .....</b>	<b>11</b>
<b>2.2. Системні вимоги.....</b>	<b>11</b>
<b>2.3. Формат даних, що завантажуються .....</b>	<b>11</b>
<b>2.4. Головні функціональні частини програми.....</b>	<b>11</b>
<b>2.5. Всі функції програми.....</b>	<b>12</b>
<b>2.6. Алгоритм розв'язку задачі .....</b>	<b>14</b>
<b>2.7 Результати роботи програми .....</b>	<b>15</b>
ВИСНОВКИ.....	19
СПИСОК ЛІТЕРАТУРИ .....	20

## ВСТУП

В сучасному світі однією з актуальних проблем стає проблема аналізу даних. Для коректного аналізу масивів даних необхідно одночасно враховувати величезну кількість факторів, що є непосильною роботою для людини. Саме тому застосування різноманітних програм, призначених для аналізу даних, значно допомагає у вирішенні таких задач.

Дана курсова робота присвячена створенню програми “CorrOK”.

“CorrOK” – програма для проведення кореляційного аналізу: на основі вибірок знаходяться коефіцієнти кореляції Pearson product-moment та Kendall’s tau-b. В результаті роботи програми виводяться знайдені коефіцієнти кореляції та точкова діаграма, що показує залежність між двома випадковими величинами по Пірсону.

Як аналог даної програми можемо розглядати різноманітні статистичні пакети та програми, такі як статистичний пакет Microsoft Excel, ПЗ Stata MP тощо. Функціонал програми не поступається більшості аналогів. Проте, на відміну від аналогів, є інструментом для проведення винятково кореляційного аналізу і є окремим програмним продуктом.

Для реалізації програмного продукту використано операційну систему Windows 7, середовище програмування C++ Builder XE7. Вибір обґрунтовується тим, що дане середовище пропонує широкий спектр зручних інструментів для створення віконних програм для операційних систем з лінійки Windows.

## ПОСТАНОВКА ЗАДАЧІ

Постановка задачі полягає у написанні програми “CorrOK”.

Програма має працювати з файлами Excel, що в свою чергу складаються із стовпців з вибірками, кожен стовпець відповідає конкретній реалізації певної вибірки.

Необхідно реалізувати процедуру пошуку Excel-файлу на жорсткому диску та носіях даних, підключених до ПК.

Головним завданням програми є створення функцій, які на вхід прийматимуть вибірки даних і обчислюватимуть коефіцієнти кореляції для обраних вибірок.

Програма має реалізувати можливість переглядати таблицю із обчисленими характеристиками. Необхідно створити можливість переглядати точкову діаграму залежності між двома ВВ за емпіричними даними (за Пірсоном).

# ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

## 1.1. Вступ

Вивчення реальної дійсності показує, що практично кожне суспільне явище знаходиться в тісному зв'язку і взаємодії з іншими явищами, якими б випадковими вони не здавалися на перший погляд. Так, наприклад, рівень урожайності сільськогосподарських культур залежить від множини природних і економічних факторів, тісно пов'язаних між собою.

І останнім часом задачі пошуку таких знань у великих базах сирих даних стають все більш популярними та актуальними. Одним із популярних методів виявлення цих знань стало знаходження коефіцієнтів кореляції між заданими вибірками для деяких генеральних сукупностей. Суть задачі полягає у перевірці існування залежностей між ВВ, що представлені цими ГС. Адже мета кореляційного аналізу — виявити чи існує істотна залежність однієї змінної від інших.

А коефіцієнт кореляції — показник, який використовують для вимірювання щільності зв'язку між результативними і факторними ознаками у кореляційно-регресійній моделі за лінійної залежності.

## 1.2. Кореляційний аналіз

Дослідження і вимірювання взаємозв'язків і взаємозалежностей соціально-економічних явищ є одним з найважливіших завдань статистики. Серед усіх методів вивчення взаємозв'язків в статистиці особливе місце займає метод кореляції, який є логічним продовженням таких методів як зіставлення паралельних рядів, дисперсійний аналіз, аналітичне групування. В поєднанні з цими методами він надає статистичному аналізу завершений характер.

Засновниками теорії кореляції є англійські статистики Карл Пірсон та Френсіс Гальтон.

Кореляційний аналіз дає змогу виміряти ступінь впливу факторних ознак на результативні, встановити єдину міру тісноти зв'язку і роль досліджуваних факторів (фактора) у загальній зміні результативної ознаки. Кореляційний метод дозволяє отримати кількісні характеристики ступеня зв'язку між двома і більшим числом ознак.

Зв'язки між факторами є досить різноманітними. При цьому одні ознаки виступають в ролі факторів, що діють на інші, зумовлюючи їх зміну, інші – в ролі дії цих факторів. Перші з них називають факторними ознаками, другі – результативними.

За допомогою кореляційного аналізу вирішують такі основні завдання:

а) визначення середньої зміни результативної ознаки під впливом одного або кількох факторів;

б) характеристика ступеня залежності результативної ознаки від одного з факторів при фіксованому значенні інших факторів, включених до кореляційної моделі;

в) визначення тісноти зв'язку між результативними і факторними ознаками (як з усіма факторами, так і з кожним фактором окремо при виключенні впливу інших).

Кореляційний зв'язок виражається відповідними математичними рівняннями. За напрямом зв'язок між корелюючими ознаками може бути прямим і оберненим. При прямому зв'язку обидві ознаки змінюються в одному напрямі, тобто із збільшенням факторної ознаки зростає результативна і навпаки. При оберненому зв'язку обидві ознаки змінюються в різних напрямках.

За формою або аналітичним вираженням розрізняють зв'язки прямолінійні (або просто лінійні) і нелінійні (або криволінійні). Якщо зв'язок між ознаками виражається рівнянням прямої лінії, то його називають лінійним зв'язком, якщо ж він виражається рівнянням будь-якої кривої (параболи, гіперболи, показникової, степеневі тощо), то такий зв'язок називають нелінійним або криволінійним.

Залежно від кількості досліджуваних ознак розрізняють парну (просту) і множинну кореляцію. При парній кореляції вивчають зв'язок між двома ознаками (результативною і факторною), при множинній кореляції – зв'язок між трьома і більшим числом ознак (результативною і двома і більшим числом факторів).

Отже, за допомогою методу кореляційного аналізу вирішується два головних завдання:

- а) визначення форми і параметрів рівняння зв'язку;
- б) вимірювання тісноти зв'язку.

Перше завдання вирішується знаходженням рівняння зв'язку і визначенням його параметрів. Друге – за допомогою розрахунку різних показників тісноти зв'язку (напр. коефіцієнта кореляції).

Встановлення причинних залежностей в досліджуваному явищі передує власне кореляційному аналізу. Тому застосуванню методів кореляції повинен передувати глибокий теоретичний аналіз, який охарактеризує основний процес, що протікає в досліджуваному явищі, визначить суттєві зв'язки між окремими його сторонами і характер їх взаємодії.

Попередній аналіз даних створює основу для формулювання конкретного завдання дослідження зв'язків, відбору найважливіших факторів, встановлення можливої форми взаємозв'язку ознак і тим самим приводить до математичної формалізації - до вибору математичного рівняння, яке найбільш повно відтворить існуючі зв'язки. [1] [2] [3] [6]

### **1.3. Коефіцієнт кореляції**

Коефіцієнт кореляції – показник, який використовують для вимірювання щільності зв'язку між результативними і факторними ознаками у кореляційно-регресійній моделі за лінійної залежності. За абсолютною величиною коефіцієнту кореляції коливається в межах від -1 до +1. Чим ближчий цей показник до 0, тим менший зв'язок, чим ближчий він до  $\pm 1$  –

тим зв'язок тісніший. Знак «плюс» при коефіцієнті кореляції означає прямий зв'язок між ознаками  $x$  і  $y$ , знак «мінус» – обернений.

Уперше коефіцієнт кореляції як показник щільності зв'язку використав Пірсон. Спочатку досліджувалася прямолінійна залежність, пов'язана із законом нормального розподілу, відтак виникла потреба в дослідженні й нелінійних залежностей. Для вимірювання щільності зв'язку нелінійних залежностей Пірсон запропонував кореляційне відношення із розробкою методів аналізу взаємозв'язку двох змінних було запропоновано теорію часткових і чистих коефіцієнтів кореляції, а також теорію множинної кореляції.

Коефіцієнт кореляції між досліджуваними ознаками повинен мати високий рівень надійності. Для оцінки надійності коефіцієнту кореляції обчислюють відношення коефіцієнта до його середньої помилки. Якщо воно дорівнює або більше 3, коефіцієнт кореляції вважають достовірним, тобто зв'язок між досліджуваними ознаками доведений. Якщо відношення менше 3, то не можна зробити висновку про достовірність зв'язку між досліджуваними ознаками. Для більшої надійності досліджень слід брати величину відношення коефіцієнту кореляції до його середньої помилки не 3, а 4. Якщо відношення коефіцієнту кореляції до його середньої помилки більше 3, а число спостережень більше 50, то вважають, що розрахований лінійний коефіцієнту кореляції відображає істотний тісний зв'язок. [4] [3]

#### **1.4. Точкова діаграма (кореляційне поле)**

Особливе місце в обґрунтуванні форми зв'язку при проведенні кореляційного аналізу належить графікам, побудованих у системі прямокутних координат на основі емпіричних даних. Графічне зображення фактичних даних дає наочне уявлення про наявність і форму зв'язку між досліджуваними ознаками.

Згідно з правилами математики при побудові графіка на осі абсцис відкладають значення факторної ознаки, а на осі ординат - значення



результативної ознаки. Відклавши на перетині відповідних значень двох ознак точки, одержимо точковий графік, який називають кореляційним полем. За характером розміщення точок на кореляційному полі роблять висновок про напрям і форму зв'язку. Якщо точки концентруються навколо уявної осі напрямленої зліва, знизу, направо, вгору, то зв'язок прямий, якщо к навпаки зліва, зверху, направо, вниз - зв'язок обернений. Якщо точки розкидані по всьому полю, то це свідчить про те, що зв'язок між ознаками відсутній або дуже слабкий. Характер розміщення точок на кореляційному полі вказує також і на наявність прямолінійного або криволінійного зв'язку між досліджуваними ознаками.

За допомогою графіка добирають відповідне математичне рівняння для кількісної оцінки зв'язку між результативною і факторною ознаками. [1] [2]

### **1.5. Висновки до розділу**

Для виявлення зв'язку між випадковими величинами користуються методами кореляційного аналізу. Це є справді зручним способом для отримання такого виду даних.

В роботі було виконано одне з основних завдань кореляційного аналізу – знаходження коефіцієнтів кореляції, причому задля пошуку не лише лінійного зв'язку між ВВ. Також побудовано точкову діаграму (кореляційне поле) для парної кореляції.

## **ОПИС ПРОГРАМНОГО ПРОДУКТУ**

### **2.1. Вступ**

Виявлення зв'язку між ВВ вручну є досить складним і важким процесом, особливо при значних розмірах вибірок даних. Проте, маючи сучасні обчислювані можливості, необхідність у ручному обчисленні відпала сама по собі. Для автоматизації виявлення зв'язку між ВВ, що було розглянуто в розділі 1, за допомогою Embarcadero RAD Studio XE 7 було створено відповідний програмний продукт.

### **2.2. Системні вимоги**

Для встановлення і нормального функціонування розробленого програмного продукту необхідно виконання наступних мінімальних системних вимог:

- ОС Microsoft Windows XP Pro, Home
- Процесор IBM, Pentium (1000 MHz)
- ОП 2 GB RAM
- SVGA (1024 x 768)
- 50 MB вільного місця на жорсткому диску

### **2.3. Формат даних, що завантажуються**

Коректним форматом даних для завантаження в даному програмному продукті є файли формату \*.xls та \*.xlsx. Формат даних є обмеженим, так як це лише перша версія продукту. В наступних версіях планується розширити ці функціональні можливості(наприклад, підтримка завантаження різних типів файлів).

### **2.4. Головні функціональні частини програми**

- MainUnit.cpp – основна частина програми, з якою пов'язані всі інші частини.
- CourseWork.h – містить всі підключені до основної програми модулі, що допомагають оперувати дизайном розробки програми.

## 2.5. Всі функції програми

Назва	Призначення
void __fastcall TMainForm::btnAboutClick(TObject *Sender)	Відобразити форму з інформацією про програмний продукт
void __fastcall TMainForm::btnExitClick(TObject *Sender)	Вихід з програми
void __fastcall TMainForm::btnHelpClick(TObject *Sender)	Відобразити форму з допоміжною інформацією для коректної роботи з програмним продуктом
void __fastcall TMainForm::btnShowExcelClick(TObject *Sender)	Запустити процес завантаження попередньо обраного Excel файлу
void __fastcall TMainForm::btnBrowseClick(TObject *Sender)	Пошук потрібного Excel файлу через провідник Windows
void __fastcall TMainForm::btnSelectClick(TObject *Sender)	Відобразити форму для вибору необхідних даних для аналізу із завантажених попередньо з Excel файлу
void __fastcall TMainForm::btnMethodPearsonClick(TObject *Sender)	Обчислити значення кореляції Пірсона для обраних даних та відобразити в окремій формі
void __fastcall TMainForm::btnMethodKendallClick(TObject *Sender)	Обчислити значення кореляції Кендалла для обраних даних та відобразити в окремій формі
void __fastcall TExcelForm::ExcelInit(System::WideString	Підключитися до обраного попередньо Excel файлу

File)	
__fastcall TExcelForm::TExcelForm(TComponent* Owner)	Завантаження раніше підключеного Excel файлу
void __fastcall TExcelForm::btnExcelClick(TObject *Sender)	Закриття форми завантаження Excel файлу
float kendall_func( float *arr1, float *arr2, int len )	Підрахунок коефіцієнту кореляції Кендалла
__fastcall TKendallForm::TKendallForm(TComponent* Owner)	Ініціалізація форми з таблицею з коефіцієнтом Кендалла, обчисленим для обраних 2-х реалізацій вибірок
double cov_func(int i, int j, int n)	Підрахунок коваріації
double Pearson (int i, int j, int n)	Обчислення коефіцієнта кореляції Пірсона
__fastcall TPearsonForm::TPearsonForm(TComponent* Owner)	Ініціалізація форми з таблицею з коефіцієнтом кореляції Пірсона, обчисленим для обраних реалізацій вибірок та точковою діаграмою у випадку вибору 2 реалізацій вибірок
void __fastcall TSelectForm::btnSelectSelectClick(TObject *Sender)	Вибір даних для аналізу серед завантажених раніше даних з Excel файлу

**Табл. 2.5.1**

Також в програмі є глобальні змінні, які оголошені для всіх функцій:  
System::WideString FileAddress;  
Variant App, Sh;  
bool checkbool;  
int iRows, iColumns;

## 2.6. Алгоритм розв'язку задачі

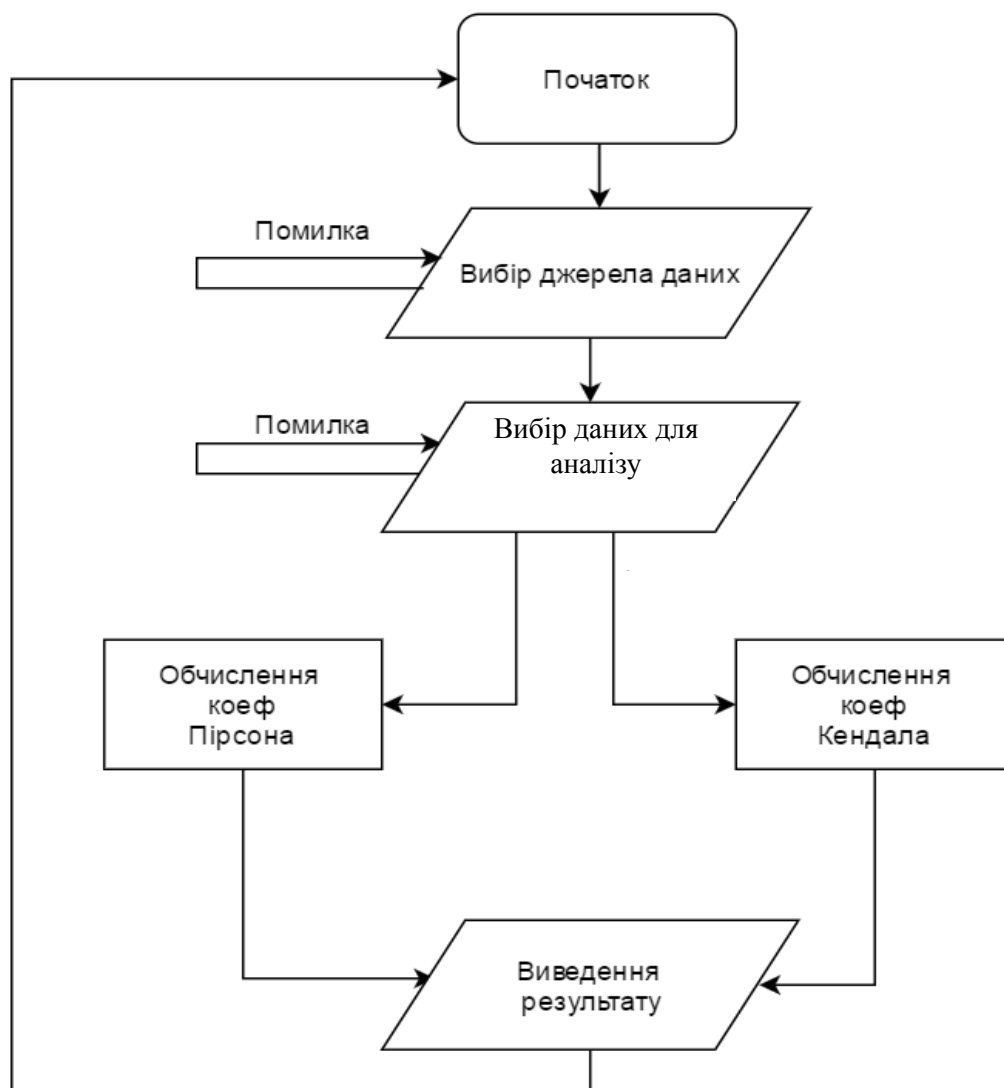
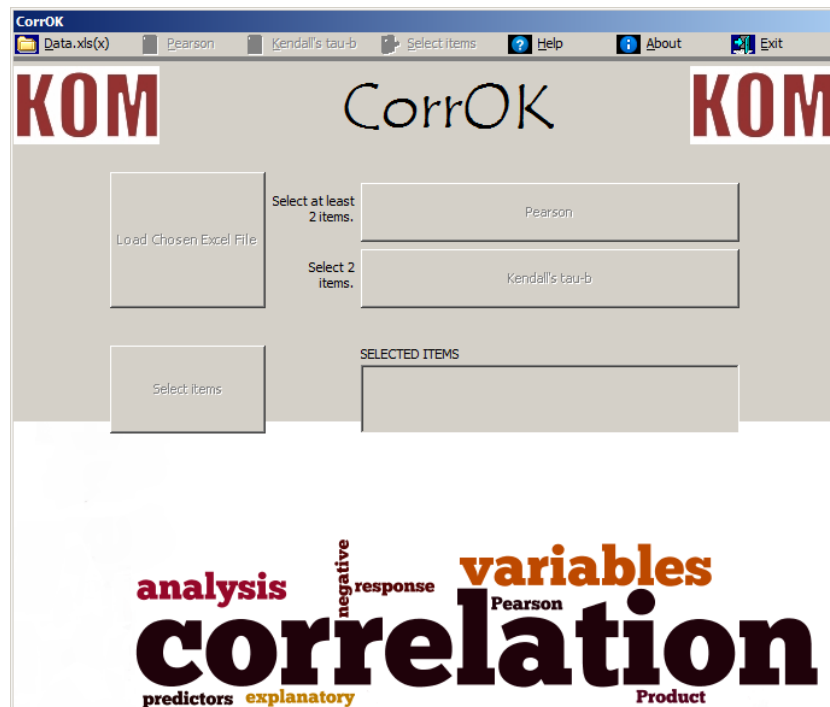


Рис. 2.6.1

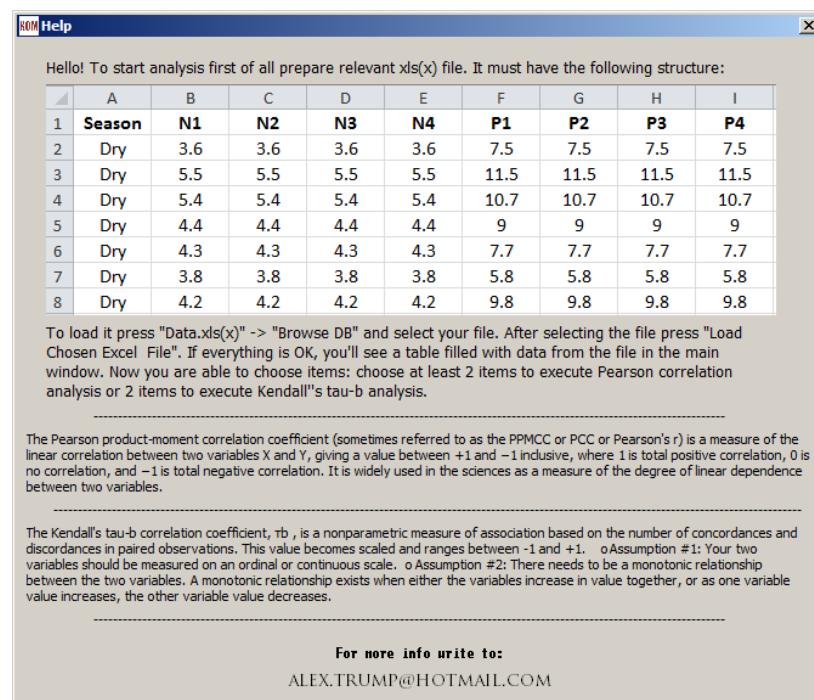
## 2.7 Результати роботи програми

При запуску програми опиняємося в головному вікні програми.



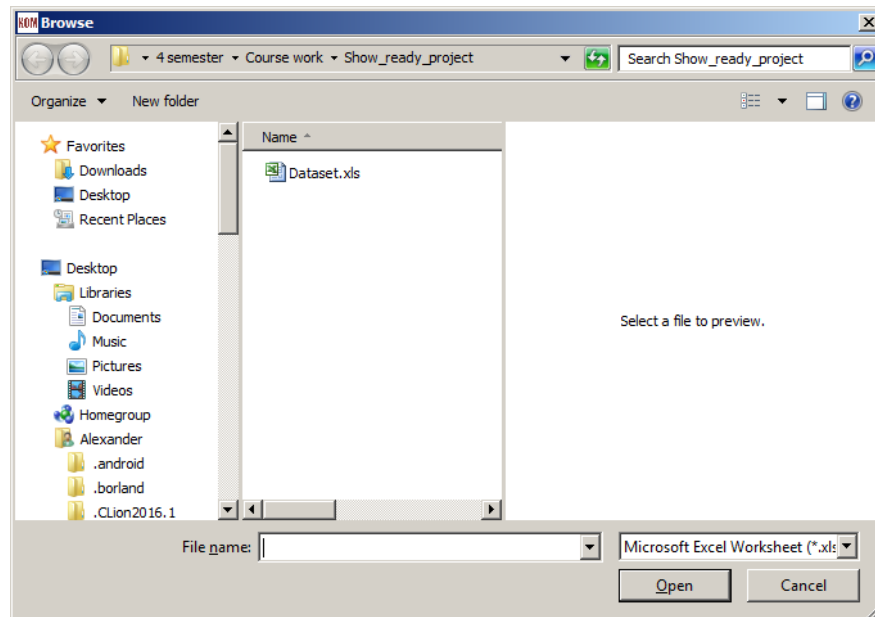
### 2.7.1 Головне вікно

Для подальшої коректної роботи користувачу необхідно ознайомитися з теоретичними викладками та правилами користування програмою. Для цього в меню тиснемо Help.



### 2.7.2 Help

Якщо натиснути на пункт меню Data.xls(x) -> Browse DB, то буде виконано пошук через провідник Windows Excel файлу з необхідними даними.



### 2.7.3 Пошук файлу

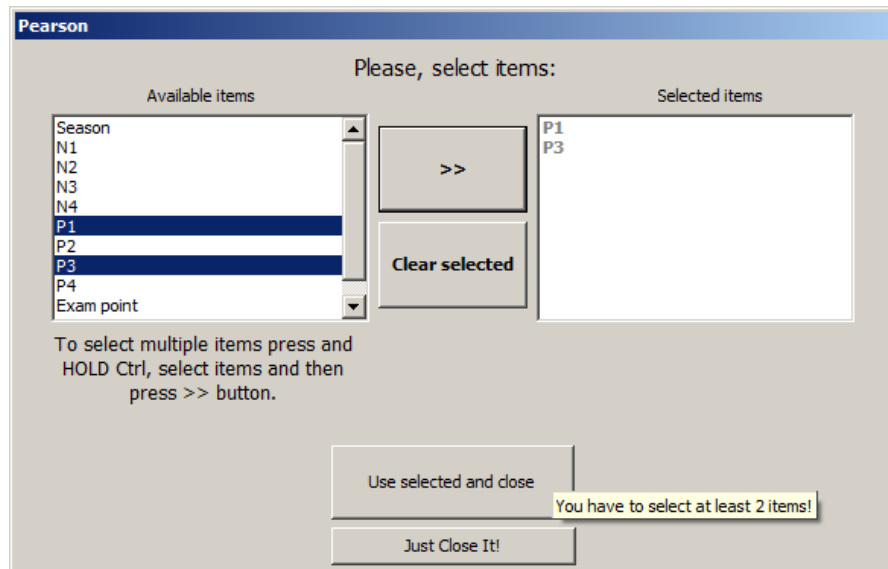
Після натискання Load Chosen Excel File та успішного завантаження файлу в головному вікні програми з'являється наступна таблиця.

Season	N1	N2	N3	N4	P1	P2	P3
Dry	3.6	3.6	3.6	3.6	7.5	7.5	7.5
Dry	5.5	5.5	5.5	5.5	11.5	11.5	11.5
Dry	5.4	5.4	5.4	5.4	10.7	10.7	10.7
Dry	4.4	4.4	4.4	4.4	9	9	9
Dry	4.3	4.3	4.3	4.3	7.7	7.7	7.7
Dry	3.8	3.8	3.8	3.8	5.8	5.8	5.8

predictors explanatory Product

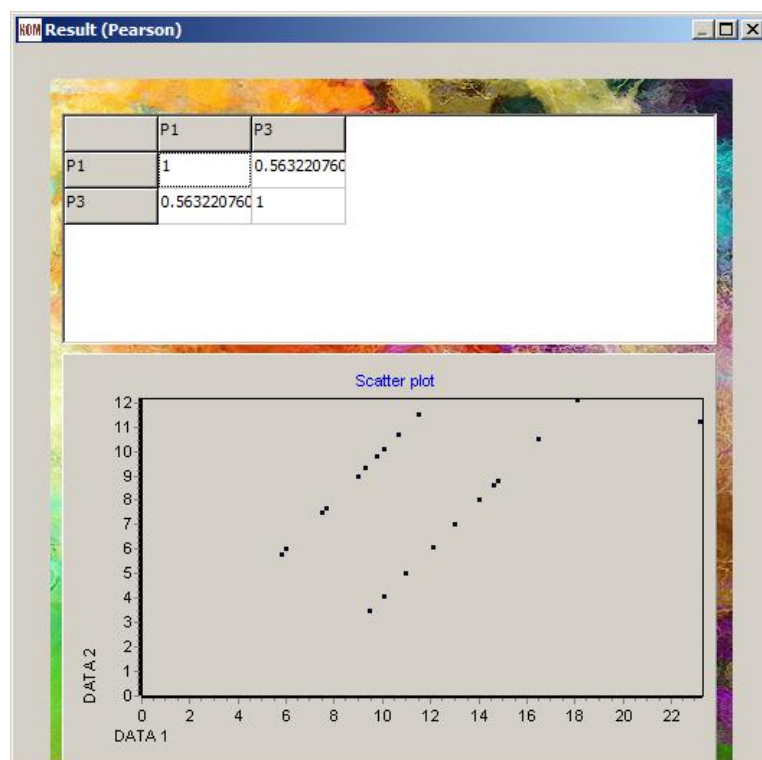
### 2.7.4 Головне вікно

Далі обираємо дані для аналізу. Для цього тиснемо Select items та обираємо більше 1 вибірки.



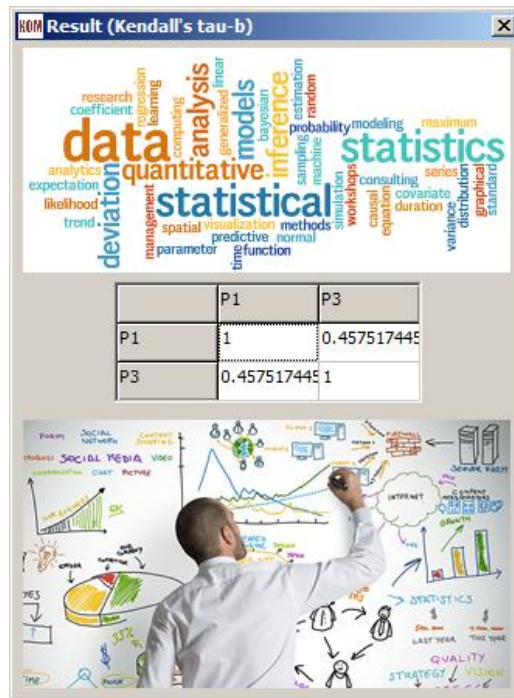
### 2.7.5 Вибір даних для аналізу

Після вибору даних можемо обчислювати коефіцієнти кореляції Пірсона та Кендалла.



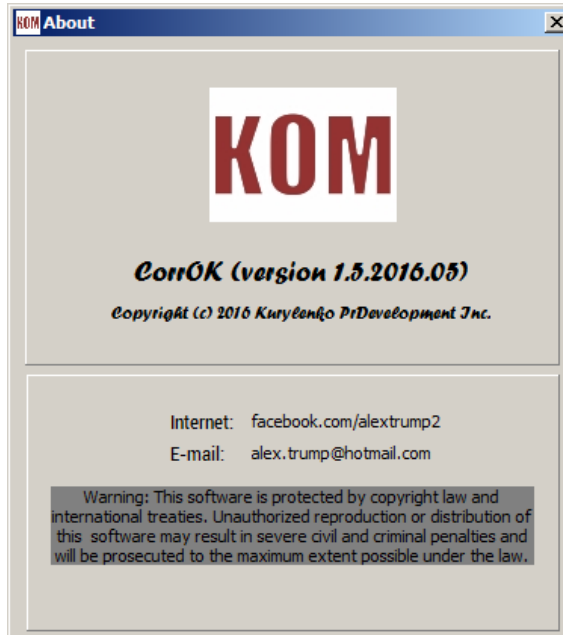
### 2.7.6 Коефіцієнт Пірсона





### 2.7.7 Коефіцієнт Кендалла

Якщо користувача зацікавила інформація про програмний продукт, він може отримати її, натиснувши на About.



### 2.7.8 Інформація про програмний продукт

Для виходу з програми потрібно натиснути Exit.

## ВИСНОВКИ

В даному курсовому проекті проводилося дослідження кореляційного аналізу. Було проведено аналіз теоретичних основ поняття кореляційного аналізу та ознайомлення алгоритмами обчислення коефіцієнтів кореляції.

В результаті написання курсової роботи була створена програма, з допомогою якої можна обчислювати коефіцієнт кореляції Пірсона (Pearson product-moment) та коефіцієнт кореляції Кендалла (Kendall's tau-b). Програма дозволяє користувачу аналізувати дані, завантажені з Excel файлу.

В процесі створення роботи використовувались знання, отримані з дисципліни «Теорія ймовірностей і математична статистика», а також певна кількість методів алгоритмізації та програмування. Використовувалася достатньо велика кількість додаткової літератури.

Дана програма не є досконалою і потребує подальшого допрацювання, проте наразі є досить ефективним інструментом для обчислення вищевказаних коефіцієнтів кореляції. В подальшому варто додати можливість обчислення інших коефіцієнтів кореляції та побудову довірчих інтервалів.

## СПИСОК ЛІТЕРАТУРИ

1. Мармоза А.Т. Теорія статистики: Навчальний посібник. - К: Ельга, Ніка - Центр, 2003. - 392 с.
2. <http://pidruchniki.com/>
3. Елисеева И.И., Юзбашев М.М. Общая теория статистики: Учебник / Под ред. И.И. Елисеевой / - М.: Финансы и статистика, 2000. - 281 с.
4. Економічна енциклопедія: У трьох томах. Т. 1. / Редкол.: ...С. В. Мочерний (відп. ред.) та ін. – К.: Видавничий центр “Академія”, 2000. – 864 с.
5. А.Я.Архангельский, М.А.Тагин «Программирование в С++ Builder», седьмое издание. Издательство: Бином-Пресс, 2010, 896 с. ISBN 5-9518-0166-4
6. Гаральд Крамер «Математические методы статистики», издание второе, стереотипное.  
Издательство: Мир, 1975, Москва

## Додаток А. Лістинг програми

### MainUnit.cpp

```
#include <vcl.h>
#pragma hdrstop

#include "MainUnit.h"
#include "AboutUnit.h"
#include "HelpUnit.h"
#include "ExcelUnit.h"
#include "SelectUnit.h"
#include "PearsonUnit.h"
#include "KendallUnit.h"

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TMainForm *MainForm;
extern System::WideString FileAddress;
Variant App;
bool checkbool;
//-----
__fastcall TMainForm::TMainForm(TComponent* Owner)
: TForm(Owner)
{
    this->btnShowExcel->Enabled=false;
    this->btnMethodPearson->Enabled=false;
    this->btnMenuPearson->Enabled=false;
    this->btnMethodKendall->Enabled=false;
    this->btnMenuKendall->Enabled=false;
    this->btnMenuSelect->Enabled=false;
    this->btnSelect->Enabled=false;
}
//-----

void __fastcall TMainForm::btnAboutClick(TObject *Sender)
{
    TAboutForm *AboutForm = new TAboutForm(this);
    AboutForm->ShowModal();
    delete AboutForm;
}
//-----

void __fastcall TMainForm::btnExitClick(TObject *Sender)
{
    try{
        App.OleProcedure("Quit");
    }
    catch(...){
    }
    MainForm->Close();
}
//-----

void __fastcall TMainForm::btnHelpClick(TObject *Sender)
{
    THelpForm *HelpForm = new THelpForm(this);
    HelpForm->ShowModal();
    delete HelpForm;
}
//-----

void __fastcall TMainForm::btnShowExcelClick(TObject *Sender)
{
    TExcelForm *ExcelForm = new TExcelForm(this);
    ExcelForm->ShowModal();
    delete ExcelForm;
    this->btnMenuSelect->Enabled=true;
    this->btnSelect->Enabled=true;
}
//-----

void __fastcall TMainForm::btnBrowseClick(TObject *Sender)
{
    TOpenDialog* openDialog = new TOpenDialog(this);
    openDialog->Title="Browse";
    openDialog->InitialDir = GetCurrentDir();
    openDialog->Filter =
        "Microsoft Excel Worksheet|*.xls|Microsoft
        Excel Worksheet|*.xlsx";
    openDialog->FilterIndex = 1;
    if (openDialog->Execute()){
        ShowMessage("File : "+openDialog-
        >FileName);
        FileAddress=openDialog->FileName;
        btnShowExcel->Enabled=true;
    }
    else Application->MessageBox(L"Open file was
    cancelled", L"Cancelled", MB_OK+MB_ICONERROR);
    delete openDialog;
}
//-----

void __fastcall TMainForm::btnSelectClick(TObject *Sender)
{
    TSelectForm *SelectForm = new TSelectForm(this);
    SelectForm->ShowModal();
    delete SelectForm;
}
//-----

void __fastcall TMainForm::btnMenuSelectClick(TObject *Sender)
{
    TSelectForm *SelectForm = new TSelectForm(this);
    SelectForm->ShowModal();
    delete SelectForm;
}
//-----

void __fastcall TMainForm::btnMethodPearsonClick(TObject
*Sender)
{
    TPearsonForm *PearsonForm = new
    TPearsonForm(this);
    PearsonForm->ShowModal();
    delete PearsonForm;
}
//-----

void __fastcall TMainForm::btnMenuPearsonClick(TObject
*Sender)
{
    TPearsonForm *PearsonForm = new
    TPearsonForm(this);
    PearsonForm->ShowModal();
    delete PearsonForm;
}
//-----

void __fastcall TMainForm::btnMethodKendallClick(TObject
*Sender)
{
    TKendallForm *KendallForm = new TKendallForm(this);
    KendallForm->ShowModal();
    delete KendallForm;
}
//-----

void __fastcall TMainForm::btnMenuKendallClick(TObject *Sender)
{
    TKendallForm *KendallForm = new TKendallForm(this);
    KendallForm->ShowModal();
}
```

```

        delete KendallForm;
    }
    //-----

```

### PearsonUnit.cpp

```

#include <vcl.h>
#pragma hdrstop

#include "PearsonUnit.h"
#include "MainUnit.h"
#include "ExcelUnit.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TPearsonForm *PearsonForm;
bool metod;

//-----
double cov_func(int i, int j, int n){
    double xm = 0;
    double ym = 0;
    for (int count = 0; count < n; count++) {
        xm += StrToFloat(MainForm->StringGridExcel->Cells[j][count+1]);
        ym += StrToFloat(MainForm->StringGridExcel->Cells[j][count+1]);
    }
    xm=xm/n;
    ym=ym/n;

    double cov = 0;
    for (int count = 0; count < n; count++) {
        cov += (StrToFloat(MainForm->StringGridExcel->Cells[j][count+1]) -
xm)*(StrToFloat(MainForm->StringGridExcel->Cells[j][count+1]) - ym);
    }

    return cov/(n-1);
}
//-----

double Pearson (int i, int j, int n){
    double s=cov_func(i, j, n)*cov_func(j, j, n);
    return cov_func(i, j, n)/sqrt(s);
}
//-----

__fastcall TPearsonForm::TPearsonForm(TComponent* Owner)
: TForm(Owner)
{
    this->ChartPearson->ShowHint=true;
    int count = MainForm->ListBoxMain->Items->Count;
    this->StringGridResult->ColCount = count+1;
    this->StringGridResult->RowCount = count+1;
    for (int j = 0; j < count; j++) {
        this->StringGridResult->Cells[j+1][0] =
MainForm->ListBoxMain->Items->Strings[j];
        this->StringGridResult->Cells[0][j+1] =
MainForm->ListBoxMain->Items->Strings[j];
    }
    try{
        for (int i = 0; i < MainForm->ListBoxNumbers->Items->Count; i++) {
            for (int j = 0; j < i+1; j++){
                double res =
Pearson(StrToInt(MainForm->ListBoxNumbers->Items->Strings[i]),
StrToInt(MainForm->ListBoxNumbers->Items->Strings[j]),
MainForm->StringGridExcel->RowCount - 1);
                this->StringGridResult->Cells[i+1][j+1]= FloatToStr(res);
            }
        }
    }
}

```

```

        this->StringGridResult->Cells[j+1][i+1]= FloatToStr(res);
    }
}
if (count == 2) {
    int c1 = StrToInt(MainForm->ListBoxNumbers->Items->Strings[1]);
    int c2 = StrToInt(MainForm->ListBoxNumbers->Items->Strings[0]);
    Series1->Clear();
    for (int i = 0; i < MainForm->StringGridExcel->RowCount-1 ; i++) {
        double y_series =
StrToFloat(MainForm->StringGridExcel->Cells[c1][i+1]);
        double x_series =
StrToFloat(MainForm->StringGridExcel->Cells[c2][i+1]);
        this->Series1->AddXY( y_series,
x_series, "",clBlue);
    }
    ChartPearson->Refresh();
    this->ChartPearson->ShowHint=false;
}
}
catch(...){
    Application->MessageBox(L"Try another
data!!",
    L"Error",MB_OK+MB_ICONERROR);
}
}
//-----

```

### KendallUnit.cpp

```

#include <vcl.h>
#pragma hdrstop

#include "KendallUnit.h"
#include "MainUnit.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TKendallForm *KendallForm;
//-----

float kendall_func( float *arr1, float *arr2, int len )
{
    int m1 = 0, m2 = 0, s = 0, nPair , i, j ;
    float cor ;

    for(i = 0; i < len; i++) {
        for(j = i + 1; j < len; j++) {
            if(arr2[i] > arr2[j]) {
                if (arr1[i] > arr1[j]) {
                    s++;
                } else if(arr1[i] < arr1[j]) {
                    s--;
                } else {
                    m1++;
                }
            } else if(arr2[i] < arr2[j]) {
                if (arr1[i] > arr1[j]) {
                    s--;
                } else if(arr1[i] < arr1[j]) {
                    s++;
                } else {
                    m1++;
                }
            } else {
                m2++;
            }
        }
    }

    if(arr1[i] == arr1[j]) {
        m1++;
    }
}

```

```

    }
    }
}

nPair = len * (len - 1) / 2;

if( m1 < nPair && m2 < nPair )
    cor = s / ( sqrtf((float)(nPair-m1)) * sqrtf((float)(nPair-m2)) );
else
    cor = 0.0f;

return cor;
}
//-----

__fastcall TKendallForm::TKendallForm(TComponent* Owner)
: TForm(Owner)
{
    try{
        for (int j = 0; j < 2; j++) {
            this->StringGridKendall-
>Cells[j+1][0] = MainForm->ListBoxMain->Items->Strings[j];
            this->StringGridKendall-
>Cells[0][j+1] = MainForm->ListBoxMain->Items->Strings[j];
        }
        int t = MainForm->StringGridExcel-
>RowCount - 1;

        float *arr1, *arr2;
        arr1 = (float*) calloc (t,sizeof(float));
        arr2 = (float*) calloc (t,sizeof(float));
        int j1 = StrToInt(MainForm-
>ListBoxNumbers->Items->Strings[0]);
        int j2 = StrToInt(MainForm-
>ListBoxNumbers->Items->Strings[1]);
        bool WAT = true;;
        try{
            for (int i = 1; i < t+1; i++) {
                arr1[i-1] =
StrToFloat(MainForm->StringGridExcel->Cells[j1][i]);
                arr2[i-1] =
StrToFloat(MainForm->StringGridExcel->Cells[j2][i]);
            }
        }
        catch(...){
            Application->MessageBox(L"Try
another data!!",

            L"Error",MB_OK+MB_ICONERROR);
            WAT = false;
        }
        free(arr1);
        free(arr2);
        if (WAT == true) {
            float cor = kendall_func(arr1,
arr2, t);

            this->StringGridKendall-
>Cells[2][1] = cor;

            this->StringGridKendall-
>Cells[1][2] = cor;

            this->StringGridKendall-
>Cells[1][1] = 1;

            this->StringGridKendall-
>Cells[2][2] = 1;
        }
    }
    catch(...){
    }
}

```