

LUCRAREA INDIVIDUALĂ

Tema: Program de statistică spreadsheet pentru consolă

A elaborat: Perju Daniela

Grupa: W-2141

Cuprins

1. Introducere
2. Specificațiile
 - 2.1 Posibilitățile multi-dimensionale
 - 2.2 Motoare și module
 - 2.3 Schema interacțiunii motoarelor cu modulele principale
3. Roadmap-ul
 - 3.1 Faze de evoluție
 - 3.2 Etape de implementare
 - 3.3 Schema de implementare
 - 3.4 Graficul de implementare
4. Unitatea de implementare
 - 4.1 Echipa de proiect
 - 4.2 Instituții pilot de testare
 - 4.3 Tehnologiile aferente
 - 4.4 Mentenanța sistemului software
5. Spațiul aplicației
 - 5.1 Concepte specifice
 - 5.2 Arhitectura proceselor
 - 5.3 Domeniul Informatic utilizat
6. Obiective și activități
 - 6.1 Obiective specifice
 - 6.2 Activități specifice
 - 6.3 Riscuri specifice

Introducere

Un **program de statistică spreadsheet pentru consolă** reprezintă o soluție accesibilă și eficientă pentru analiza datelor, oferind utilizatorilor funcționalități esențiale într-un mediu minimalist, bazat pe text. Acest tip de aplicație elimină necesitatea interfețelor grafice complexe, concentrându-se pe prelucrarea rapidă a datelor și ușurința în utilizare.

Programul permite importul datelor din fișiere spreadsheet și oferă o gamă largă de calcule statistice fundamentale, cum ar fi media, mediana, abaterea standard sau varianța. Rezultatele pot fi sintetizate sub formă de tabele rezumative care facilitează interpretarea informațiilor esențiale. Utilizatorii au la dispoziție comenzi simple pentru manipularea datelor, ceea ce face programul potrivit atât pentru începători, cât și pentru utilizatori avansați.

Datorită designului său minimalist, programul este compatibil cu sisteme cu resurse limitate, fiind ideal pentru medii educaționale, cercetare sau situații în care software-ul sofisticat nu este accesibil. Prin această abordare, aplicația democratizează accesul la uneltele statistice, sprijinind utilizatorii să înțeleagă și să interpreteze datele într-un mod eficient și practic.

Specificațiile

2.1 Posibilitățile Multi-dimensionale

Un program de statistică pentru consolă oferă capacitatea de a manipula și analiza seturi de date multi-dimensionale, adică date care au mai multe variabile și relații complexe între acestea. În acest context, programul va permite utilizatorilor să:

- **Gestionarea datelor în mai multe dimensiuni:** Utilizatorii vor putea importa și manipula fișiere ce conțin date structurate pe mai multe coloane (dimensiuni), cum ar fi fișiere CSV sau Excel cu mai multe variabile (ex: vânzări, date demografice, date geografice etc.).
- **Analize statistice multidimensionale:** Programul va oferi funcționalități care permit realizarea de analize statistice complexe, precum regresia multiplă, analiza componentelor principale (PCA) sau analiza de corelație între mai multe variabile simultan.
- **Vizualizare multidimensională:** Deși programul se bazează pe consolă, va permite afișarea tabelelor care rezumă relațiile dintre dimensiuni și va oferi, acolo unde este posibil, opțiuni pentru exportul datelor într-un format vizual pentru analize mai detaliate.

2.2 Motoare și Module

Programul va funcționa pe baza unor **motoare de procesare** și **module de calcul** interconectate, care vor gestiona funcționalitățile specifice ale aplicației. Aceste motoare și module vor fi concepute astfel încât să îndeplinească anumite roluri și să asigure flexibilitate și extensibilitate în utilizare:

- **Motorul de procesare a datelor:** Responsabil pentru importul, curățarea și organizarea datelor. Acest motor va putea să prelucreze fișiere de dimensiuni mari și să le convertească în formate ușor de manipulat. De asemenea, va permite utilizatorilor să filtreze sau să transforme datele înainte de a le analiza.
- **Motorul de calcul statistic:** Va efectua calcule statistice fundamentale (media, mediana, varianța) și analize mai complexe (regresie, corelație, analize multidimensionale). Acesta va fi capabil să calculeze indicatorii statistici pentru seturi de date mari și să aplice formule avansate.
- **Modulele de interfață utilizator:** Aceste module vor asigura interacțiunea utilizatorului cu programul. Deși programul se va baza pe interfața de consolă, aceste module vor gestiona comenzile introduse de utilizator, vor afișa rezultatele și vor oferi opțiuni de personalizare a sesiunii de lucru.

2.3 Schema Interacțiunii Motoarelor cu Modulele Principale

Interacțiunea dintre motoare și module va fi reglată printr-o arhitectură clară, care va permite fluxuri de date eficiente și un control precis asupra proceselor de analiză și procesare. Schema generală a interacțiunii va arăta astfel:

Importul și prelucrarea datelor:

- Utilizatorul încarcă fișierele de date prin intermediul **modulelor de interfață utilizator**.
- Motorul de procesare a datelor primește fișierele și le convertește într-un format intern.
- Dacă sunt necesare curățări sau transformări ale datelor, motorul de procesare va aplica procedurile corespunzătoare.

Analiza statistică:

- Datele prelucrate sunt transmise către **motorul de calcul statistic**, care efectuează analizele solicitate.
- Motorul va calcula indicatorii statistici fundamentali și va realiza analize complexe, în funcție de comanda utilizatorului.
- Rezultatele analizei sunt returnate către modulele de interfață utilizator pentru a fi afișate pe consolă.

Exportul rezultatelor:

- Dacă utilizatorul dorește să exporte rezultatele, modulele de interfață utilizator vor permite exportul acestora într-un format compatibil (CSV, Excel).
- Motorul de procesare va formata datele într-un mod adecvat pentru export și va trimite fișierul către sistemul de fișiere.

Această structură modulară asigură o separare clară a funcționalităților, ceea ce permite atât extinderea programului cu noi module, cât și o utilizare eficientă a resurselor de calcul.

Roadmap-ul

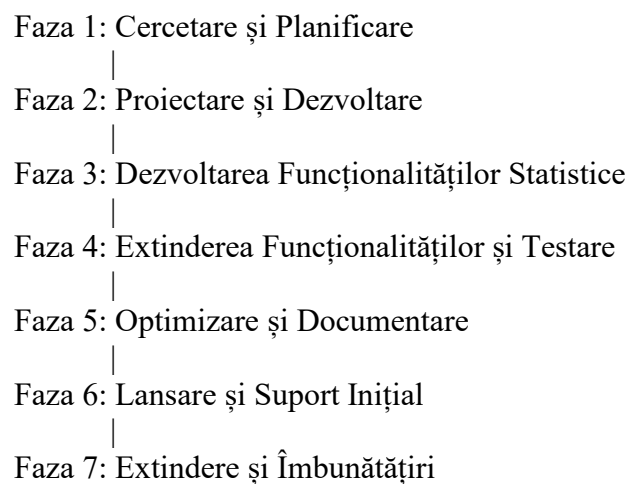
3.1 Faze de Evoluție

Faza	Descriere	Durata estimată
Faza 1: Cercetare și Planificare	Definirea cerințelor și obiectivelor programului. Studii de fezabilitate, analiza cerințelor tehnice și planificarea arhitecturii.	3 luni
Faza 2: Proiectare și Dezvoltare Inițială	Proiectarea interfeței utilizator și arhitecturii tehnice. Dezvoltarea primelor module de import și procesare a datelor.	6 luni
Faza 3: Dezvoltarea Funcționalităților Statistice	Implementarea motoarelor de calcul statistic și a modulelor pentru analize simple (media, mediana, varianța).	6 luni
Faza 4: Extinderea Funcționalităților și Testare	Adăugarea de funcționalități avansate (regresie multiplă, analiza componentelor principale). Testarea și validarea sistemului.	9 luni
Faza 5: Optimizare și Documentare	Îmbunătățirea performanței programului și documentarea completă a codului și a funcționalităților.	6 luni
Faza 6: Lansare și Suport Inițial	Lansarea oficială a programului, oferirea suportului inițial și actualizărilor.	3 luni
Faza 7: Extindere și Îmbunătățiri	Extinderea funcționalităților și integrarea de noi module, precum și actualizarea periodică a programului.	1-2 ani

3.2 Etape de Implementare

Etapă	Activități principale	Durata estimată
Etapa 1: Planificare și Analiză	Definirea cerințelor programului, crearea unui plan detaliat de implementare.	1 lună
Etapa 2: Design Arhitectural și Interfața Utilizatorului	Crearea arhitecturii tehnice și a designului interfeței bazate pe consolă.	2 luni
Etapa 3: Dezvoltare Motor de Procesare	Dezvoltarea modului de import al datelor și procesare inițială.	2 luni
Etapa 4: Dezvoltare Funcționalități Statistice	Implementarea calculului statistic de bază (media, varianța, mediana).	2 luni
Etapa 5: Testare și Validare	Testarea funcționalităților și corectarea erorilor.	2 luni
Etapa 6: Optimizare Performanță	Optimizarea programului pentru performanță și scalabilitate.	2 luni
Etapa 7: Documentație și Lansare	Crearea documentației pentru utilizatori și dezvoltatori, lansarea programului.	1 lună
Etapa 8: Suport și Îmbunătățiri Post-Lansare	Suport pentru utilizatori și actualizări periodice.	Continuă

3.3 Schema de Implementare



Schema arată evoluția liniară a dezvoltării programului, începând cu analiza cerințelor, continuând cu designul și dezvoltarea funcționalităților, și culminând cu lansarea oficială și îmbunătățiri ulterioare pe termen lung.

3.4 Graficul de Implementare pe 4, 6, 8 ani

Grafic pe 4 ani

An	Faza 1-2 (Cercetare & Dezvoltare Inițială)	Faza 3-4 (Dezvoltare Statistici & Testare)	Faza 5 (Optimizare & Lansare)
1	✓		
2	✓	✓	
3		✓	✓
4			✓

Grafic pe 6 ani

An	Faza 1-2 (Cercetare & Dezvoltare Inițială)	Faza 3-4 (Dezvoltare Statistici & Testare)	Faza 5 (Optimizare & Lansare)	Faza 6-7 (Extindere & Suport)
1	✓			
2	✓			
3		✓		
4		✓	✓	
5			✓	✓
6				✓

Grafic pe 8 ani

An	Faza 1-2 (Cercetare & Dezvoltare Inițială)	Faza 3-4 (Dezvoltare Statistici & Testare)	Faza 5 (Optimizare & Lansare)	Faza 6-7 (Extindere & Suport)
1	✓			
2	✓			
3		✓		
4		✓	✓	
5			✓	✓
6				✓
7				✓
8				✓

Aceste grafice ilustrează evoluția etapelor de dezvoltare a programului pe termene de 4, 6 și 8 ani, incluzând activitățile de optimizare, lansare și extindere a funcționalităților.

Unitatea de implementare

Pentru a asigura succesul dezvoltării și implementării programului de statistică pentru consolă, este esențial să existe o echipă bine structurată, instituții de testare care să valideze funcționalitățile, tehnologiile adecvate pentru dezvoltare și un plan robust de mentenanță.

4.1 Echipa de Proiect: Specialiști și Roluri

Rol	Număr de specialiști	Responsabilități principale
Manager de Proiect	1	Coordonarea generală a proiectului, gestionarea resurselor și a timeline-ului.
Arhitect Software	1	Proiectarea arhitecturii sistemului și a interacțiunii modulelor.
Dezvoltatori Backend	2	Dezvoltarea motoarelor de procesare a datelor și a funcționalităților statistice.
Dezvoltator de Interfață (CLI)	1	Proiectarea și dezvoltarea interfeței pentru consolă (CLI).
Specialist în Statistică	1	Definirea și validarea funcționalităților statistice.
Specialist QA (Quality Assurance)	1	Testarea funcționalităților și identificarea erorilor.
Tehnician IT	1	Gestionarea infrastructurii hardware și suportul pentru echipa de dezvoltare.

Total: **8 specialiști**, fiecare cu un rol bine definit pentru a asigura o dezvoltare eficientă și o implementare de succes.

4.2 Instituții Pilot de Testare

Pentru a valida funcționalitățile programului, se vor selecta instituții pilot din diverse domenii care se bazează pe analiza statistică. Aceste instituții vor oferi feedback detaliat și vor ajuta la ajustarea programului înainte de lansarea oficială.

Instituție	Motivul selecției
Universități (ex: Facultatea de Matematică și Informatică)	Testarea în medii academice pentru validarea funcționalităților educaționale.
Birouri de Statistică Naționale/Regionale	Validarea programului în utilizări profesionale pentru analiză de date publice.
Companii de consultanță sau cercetare	Testarea aplicației în medii corporate pentru analiza datelor comerciale.
Organizații non-guvernamentale	Utilizarea în proiecte de cercetare cu resurse limitate.

4.3 Tehnologiile Aferente

Tehnologiile utilizate în dezvoltarea programului vor fi selectate pentru a asigura performanță, scalabilitate și compatibilitate cu diferite platforme.

Tehnologie	Utilizare
Python	Limbaj de programare principal, datorită bibliotecilor pentru statistici și analiză.
Pandas, NumPy, SciPy	Biblioteci pentru procesarea și analiza datelor statistice.
SQLite	Stocarea locală a datelor pentru sesiuni complexe de analiză.
Text User Interface (TUI)	Crearea interfeței bazate pe consolă, folosind biblioteci precum rich.
Git/GitHub	Gestionarea versiunilor codului sursă și colaborarea între echipă.
Docker	Asigurarea unui mediu standardizat pentru testare și distribuire.
Unit Testing Framework (ex: pytest)	Testarea automată a funcționalităților.

4.4 Mentenanța Sistemului Software

Pentru a menține programul actualizat și funcțional, mentenanța va fi împărțită în mai multe componente:

Actualizări periodice:

1. Corectarea erorilor raportate de utilizatori.
2. Adăugarea de funcționalități noi pe baza feedbackului.
3. Optimizarea performanței pentru a gestiona seturi de date mai mari sau mai complexe.

Suport tehnic:

1. Crearea unei echipe dedicate de suport tehnic pentru a răspunde problemelor utilizatorilor.
2. Furnizarea unei documentații detaliate pentru instalare, utilizare și depanare.

Monitorizare și raportare:

1. Implementarea unui sistem de raportare a erorilor (ex: log-uri automate).
2. Analiza feedback-ului pentru îmbunătățiri continue.

Compatibilitate pe termen lung:

1. Asigurarea compatibilității programului cu versiunile noi ale sistemelor de operare.
2. Testarea periodică pe diferite platforme pentru a preveni problemele de interoperabilitate.

Un plan de mentenanță bine structurat va asigura longevitatea programului, menținându-l util și eficient pe termen lung.

Spațiul aplicației

5.1 Concepte Specifice

Programul este construit pe baza următoarelor concepte specifice, care definesc obiectivele și modul său de operare:

- **Prelucrarea datelor în loturi (Batch Processing):** Datele sunt procesate în unități definite (ex: rânduri sau tabele întregi) pentru a maximiza eficiența și a minimiza consumul de resurse.
- **Statistici descriptive și inferențiale:** Programul oferă funcții pentru analiza statistică de bază (media, mediana, varianța) și analize complexe (regresii, teste de ipoteză).
- **Interfață bazată pe consolă (CLI):** Se prioritizează simplitatea și accesibilitatea prin utilizarea unei interfețe textuale care elimină dependența de interfețe grafice.
- **Compatibilitate multiplatformă:** Programul este proiectat să funcționeze pe diverse sisteme de operare (Windows, Linux, macOS) fără adaptări majore.
- **Extensibilitate modulară:** Arhitectura modulară permite adăugarea facilă a noilor funcționalități, cum ar fi suport pentru analize mai avansate sau formate de fișiere suplimentare.

5.2 Arhitectura Proceselor

Arhitectura proceselor din cadrul aplicației este proiectată pentru a asigura un flux logic și eficient al operațiunilor, având la bază următoarele etape:

Importul datelor:

- Datele sunt încărcate din fișiere CSV, Excel sau alte formate compatibile.
- Motorul de procesare a datelor verifică integritatea și curăță eventualele erori.

Prelucrarea datelor:

- Datele sunt organizate în structuri interne optimizate pentru analiză.
- Utilizatorul poate aplica filtre sau transformări (ex: eliminarea valorilor lipsă).

Analiza statistică:

- Motorul statistic procesează comenzile utilizatorului și returnează indicatorii ceruți.
- Analizele complexe (ex: regresie multiplă) sunt efectuate folosind algoritmi eficienți.

Prezentarea rezultatelor:

- Rezultatele sunt afișate sub formă de tabele rezumative sau text simplu.
- Opțional, datele pot fi exportate în formate compatibile pentru utilizări ulterioare.

Salvarea și exportul:

- Programul permite salvarea setărilor și a datelor analizate într-un format local.
- Exportul rezultatelor poate include formate precum CSV, JSON sau PDF.

5.3 Domeniul Informatic Utilizat

Aplicația se încadrează în domeniul **informaticii aplicate în statistici și analiză de date**, folosind un set de tehnologii și paradigme pentru a facilita prelucrarea eficientă a datelor.

Aspect	Descriere
Tip de aplicație	Software de analiză statistică (local, bazat pe consolă).
Tehnologii utilizate	Python (Pandas, NumPy, SciPy), SQLite pentru stocarea datelor temporare.
Paradigmă de programare	Programare procedurală și modulară pentru flexibilitate și extensibilitate.
Metodologii de dezvoltare	Agile, pentru a permite adaptarea rapidă la cerințele utilizatorilor.
Medii de execuție	Sisteme de operare multiplatformă (Windows, Linux, macOS).
Infrastructură hardware	Consum redus de resurse, compatibil cu sisteme entry-level.

Acest domeniu se intersectează cu știința datelor, statistica și informatică de procesare, sprijinind utilizatorii în gestionarea eficientă a seturilor de date și extragerea informațiilor relevante.

Obiective și activități

6.1 Obiective Specifice

1. Dezvoltarea unui software eficient pentru analiza statistică:

Asigurarea funcționalităților esențiale pentru prelucrarea datelor și calculul statistic.

2. Crearea unei interfețe prietenoase și intuitive pentru utilizatorii non-tehnici:

Oferirea unui design bazat pe consolă ușor de învățat și utilizat.

3. Asigurarea compatibilității multiplatformă:

Funcționarea fără probleme pe Windows, Linux și macOS.

4. Optimizarea performanței pentru gestionarea seturilor mari de date:

Implementarea algoritmilor eficienți pentru calculul statistic.

5. Extensibilitatea modulară a aplicației:

Posibilitatea de a adăuga noi funcționalități sau module fără a afecta structura existentă.

6. Testarea în medii diverse pentru validarea calității și acurateței:

Obținerea unui feedback consistent din partea instituțiilor pilot și ajustarea aplicației.

6.2 Activități Specifice

Activitate	Descriere	Durata estimată
1. Analiza cerințelor utilizatorilor	Realizarea interviurilor și analizelor pentru a înțelege nevoile utilizatorilor finali.	1 lună
2. Proiectarea arhitecturii aplicației	Crearea unui design modular și flexibil al programului.	2 luni
3. Dezvoltarea interfeței bazate pe consolă (CLI)	Implementarea funcționalităților de bază pentru interacțiunea utilizatorului cu aplicația.	3 luni
4. Implementarea funcționalităților statistice	Dezvoltarea și integrarea algoritmilor pentru analize statistice (descriere, inferență).	6 luni
5. Testarea aplicației în instituții pilot	Validarea aplicației în medii academice, guvernamentale și corporative.	2 luni
6. Optimizarea performanței aplicației	Corectarea erorilor și creșterea eficienței algoritmilor.	2 luni
7. Crearea documentației pentru utilizatori	Elaborarea unui ghid de utilizare detaliat și accesibil.	1 lună
8. Lansarea aplicației	Pregătirea și publicarea oficială a aplicației.	1 lună
9. Mentenanță și actualizări	Monitorizarea performanței aplicației și implementarea îmbunătățirilor periodice.	Continuă

6.3 Riscuri Specifice

Risc	Descriere	Măsuri de reducere
Risc tehnologic	Posibilitatea ca tehnologiile selectate să devină depășite sau incompatibile.	Alegerea tehnologiilor open-source bine susținute și actualizarea periodică.
Risc de performanță	Algoritmii statistici pot fi ineficienți pentru seturi de date foarte mari.	Testarea și optimizarea riguroasă a algoritmilor.
Risc de utilizator	Interfața CLI poate fi percepută ca dificilă de utilizatorii non-tehnici.	Crearea unui ghid detaliat și asigurarea unei structuri intuitive a interfeței.
Risc de implementare	Întârzieri în dezvoltare din cauza unei planificări insuficiente.	Urmărirea unui roadmap detaliat și utilizarea metodologiilor Agile.
Risc de acceptare	Feedback negativ din partea instituțiilor pilot sau utilizatorilor finali.	Integrarea unui proces activ de colectare a feedback-ului și ajustare rapidă.
Risc de securitate	Posibile vulnerabilități în manipularea datelor sau stocarea locală.	Implementarea măsurilor de criptare și testarea riguros a securității.