2022

Université Grenoble Alpes

AppStats

MACÉ Gabriel 12010249

VELLETAZ Cédric 11908187

ZARCH Vassili 120107374

1. Introduction

Ce document vise à rendre compte du but et des attendus du projet en définissant tous les aspects de celui-ci.

Il présente les concepts de base liés au projet, le contexte et l'historique dans lequel il s'inscrit, ses contraintes, et son déroulement.

2. Concepts de base

Programmation Orientée Objet - statistiques descriptives/inférentiels - développement - interface

3. Contexte

Notre projet est un projet informatique à but académique. Les enjeux principaux sont de poursuivre l'acquisition de nouvelles compétences quant à la gestion de projet, et de réussir à développer un logiciel. De plus, il existe déjà des interfaces ressemblantes à celle de notre projet, il s'agit donc aussi de se différencier.

4. Description de la demande

4.1.Les objectifs

-Permettre d'entrer des données et d'appliquer des tests statistiques sur celles-ci afin d'en ressortir des informations.

-Permettre de tester les connaissances en statistiques de l'utilisateur

-Permettre à l'utilisateur de comprendre à quoi servent les différentes méthodes statistiques.

-Permettre à l'utilisateur de se corriger et de reconnaître ses erreurs.

- Rendre le logiciel exécutable sur n'importe PC avec le minimum de programme tiers.

4.2.Produit du projet

Le produit de notre projet est un logiciel proposant des applications de méthodes statistiques sur les données entrées par l'utilisateur comportant aussi une partie pédagogique permettant à l'utilisateur de mieux comprendre les statistiques.

4.2.1.Descriptions des risques

Risques portant sur le déploiement logiciel

L'enjeu est de créer un fichier exécutable par l'utilisateur sur les PCs sous Windows, MacOsX et GNU/Linux. Le fichier d'extension .exe est le standard sur windows. Mais sur les machines sous linux, le logiciel de compatibilité wine rend exécutable les fichiers .exe. Si, le but est de développer l'application sur GNU/Linux et MacOS alors développer sur windows n'est pas transversal car la méthode est de créer un package (une forme d'archive). ie, Un fichier archive .deb sous distribution debian, .rpm sous une distro Fedora, .dmg sous MacOS. Le but ici n'est pas de lister les méthodes possibles car par exemple un package Flatpack est indépendant d'une distro. En sachant que pour déployer un package, c'est intégré les différences entre les utilisateurs, les systèmes d'exploitations qui peuvent dépendre entre elles, les gestionnaires de paquets qui ne sont pas connus par les utilisateurs voir tout le monde notamment sur windows et ainsi l'interaction entre chacune des composantes. Alors, Nous excluons le risque de créer un package dépendant d'un système d'exploitation PC en programmant sous java. En effet, l'adage de Sun la société créatrice de java est: "Write once, run anywhere". Un solution est créer un fichier .jar à partir de javafx exécutable possédant une version supérieur ou égal à java 8.
https://www.jetbrains.com/help/idea/packaging-javafx-applications.html#troubleshoot

Risques développement interface graphique:

javafx vs swing
-> le superviseur maîtrise swing mais pas javafx ainsi pas d'aide sur javafx
solution:
encapsuler la partie swing dans javafx
https://openjfx.io/javadoc/19/javafx.swing/javafx/embed/swing/package-summary.html
-> apprendre en autodidacte javafx
-> javafx possède une application pour faire des graphs
https://github.com/HanSolo/charts
-> la facilité de créer un projet de type javafx sur intellij
-> javafx possède son propre WindowBuilder dénommer SceneBuilder
https://gluonhq.com/products/scene-builder/

Risque portant sur la pédagogie

pédagogie statistique vs pédagogie de l'outil logiciel

4.3.Les fonctions du produit

Partie exploitation de données :

-Récupérer des données

Sur un échantillon:

- getMoyenne()
- getVariance()
- getEcarttype()
- getMinimum()
- getMaximum()
- getMediane()
- getQuartiles(int quartile) : Récupérer un quartile
- getTaille() Récupérer la taille de l'échantillon
- getSCT() : Somme des carrés totaux
- Test de normalité
- toString()

- Tester si une vi a un effet sur une vd : Régression linéaire

- getCov() : retourne la covariance des deux échantillons :
- getr()
- getR()
- getBeta1()
- getBeta2()
- getSCE()
- getSCM()
- getF()
- decision()
- toString()

- Tester l'égalité des moyennes : ANOVA

-getSCE()

-getSCM()

-getF() : Test de Fisher

- get $q_{0,95}$ de $F_{a-1,n-a}$ d'une loi de Fisher et compare à F.

-decision() : si $F > q_{0,95}$ => rejette l'égalité des moyennes, sinon non.

-toString()

- test du khideux d’indépendance

- getDifference()
- getDecision()
- toString()

- Tester si l'interaction de plusieurs vi à un effet sur une vd : ANCOVA

- Test de student

-Tables de statistique :

- Loi Normale
- Fractiles de la Loi Normale
- Khideux
- Student
- Loi de Fisher-Snedecor : Probabilité 0.05
- Loi de Fisher-Snedecor : Probabilité 0.025
- Loi de Fisher-Snedecor : Probabilité 0.01

-Représentation graphique:

- Nuage de point et tracé de droite
- Graphe des résidus (régression linéaire)
- Récapitulatifs des moyennes et des variances
- Boxplot (anova)
- Histogramme

Partie pédagogique :

- Test + résultat
- Formules + définition
- Interface graphique conviviale

4.4.Critères d’acceptabilité et de réception

- Les données sont toutes bien récupérées.
- Les résultats retournés sont justes.
- Les exercices d’apprentissages ne comportent pas de fautes.

5. Contraintes

5.1. Contraintes de coûts

Aucun budget n’est alloué à ce projet.

0 euros = 0 dollars = 0 francs = 0 livre-sterling = 0 francs CFA

5.2. Contraintes de délais

Pour le nouvel an.

5.3. Contraintes matérielles

(Spécifier le matériel nécessaire au bon fonctionnement du produit)

Un ordinateur comportant java.

5.4. Autres contraintes

Spécifier les éventuelles contraintes à prendre en compte dans le cadre du projet (normes techniques, clauses juridiques, etc.)

6. Déroulement du projet

6.1. Planification

Représenter les grandes phases du projet et les étapes principales

A. Définition de la structure du code et des différentes fonctions
B. Création de la partie exploitation des données :
    a. Développement des fonctions
    b. Création de l'interface
C. Création de la partie pédagogique :
    a. Développement des fonctions
    b. Création de l'interface

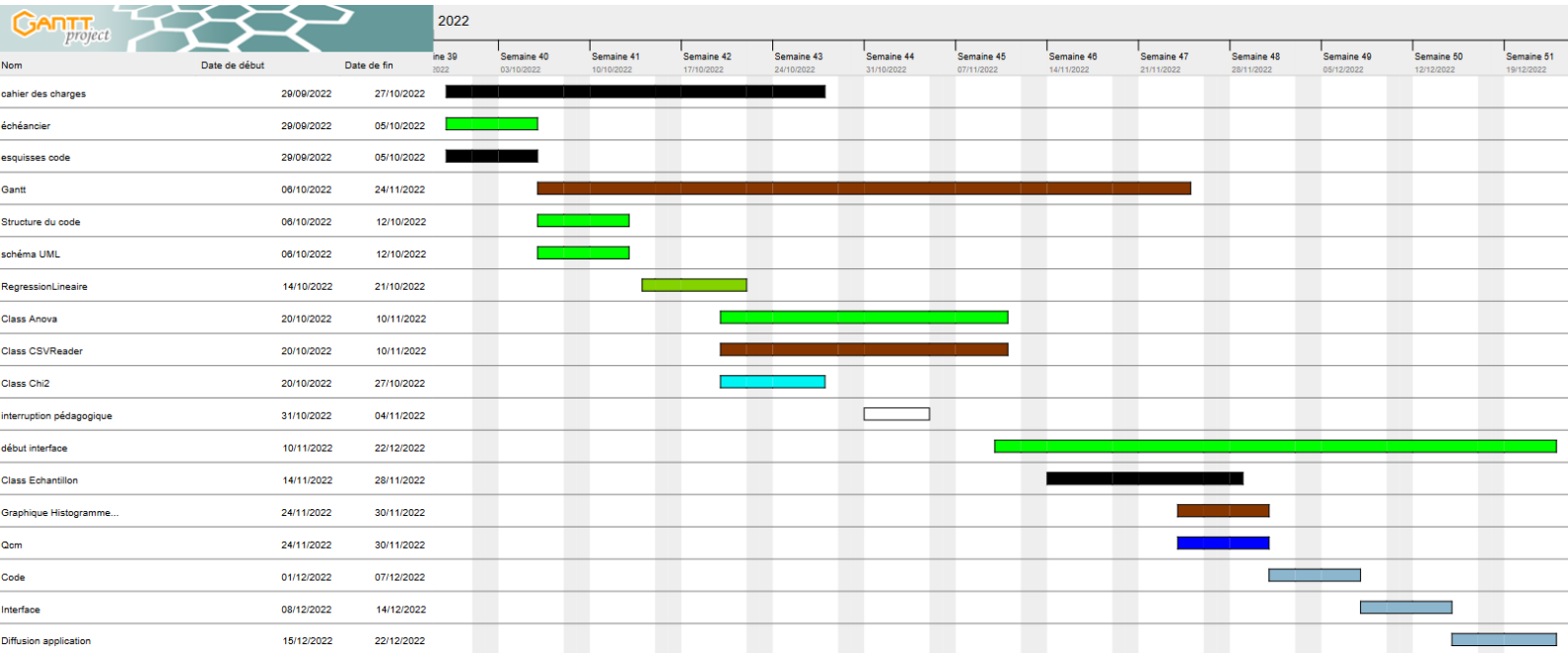

Diagramme Prévisionnel du déroulement du projet

6.2. Ressources

Lister les ressources humaines et matérielles que le client peut mettre à la disposition du prestataire

7. Annexes

Lister et joindre au cahier des charges les éventuels documents que le client peut mettre à disposition

8. Glossaire

Définit l'ensemble des termes spécialisés du document

9. Références

Indique les références bibliographiques vers d'autres documents apportant des informations complémentaires

Math – The Commons Math User Guide - Statistics. (s. d.). Consulté le 29 septembre 2022, à l'adresse https://commons.apache.org/proper/commons-math/userguide/stat.html

https://github.com/kermitt/ANOVA ANOVA réaliser en java

10. Index

Liste les mots-clés du document et où les trouver dans celui-ci