**FINK Jérôme & SEEL Océane**

2015-2016

**Laboratoire E-commerce**

Phase 1 : Data mining et informations statistiques

Sommaire

[1. Code du client Applic\_Data\_Analysis 3](#_Toc435917492)

[1.1 ApplicationDataAnalysis.java 3](#_Toc435917493)

[1.2 Login.java 4](#_Toc435917494)

[1.3 Menu.java 8](#_Toc435917495)

[1.4 GrCouleurRep.java 10](#_Toc435917496)

[1.5 GrCouleurComp.java 11](#_Toc435917497)

[1.6 StatInferTestConf.java 11](#_Toc435917498)

[1.7 StatInferTestHomog.java 11](#_Toc435917499)

[1.8 StatInferTestAnova.java 11](#_Toc435917500)

[1.9 ProtocolePIDEP.java 11](#_Toc435917501)

[1.1 Utility.java 12](#_Toc435917502)

[2. Explications des 3 requêtes d’inférence statistique 14](#_Toc435917503)

[2.1 Test d’hypothèse de conformité 14](#_Toc435917504)

[2.2 Test d’hypothèse d’homogénéité 14](#_Toc435917505)

[2.3 Test d’hypothèse de type ANOVA 14](#_Toc435917506)

# Code du client Applic\_Data\_Analysis

## ApplicationDataAnalysis.java

package application\_data\_analysis;

import java.awt.CardLayout;

import java.net.\*;

public class ApplicationDataAnalysis extends javax.swing.JFrame

{

public static Socket cliSock = null;

public Boolean isConnected = false;

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

ButtonGroup = new javax.swing.ButtonGroup();

Menu = new application\_data\_analysis.Menu();

StatDescrCont = new application\_data\_analysis.StatDescrCont();

GrCouleurRep = new application\_data\_analysis.GrCouleurRep();

GrCouleurComp = new application\_data\_analysis.GrCouleurComp();

StatInferTestConf = new application\_data\_analysis.StatInferTestConf();

StatInferTestHomog = new application\_data\_analysis.StatInferTestHomog();

StatInferTestAnova = new application\_data\_analysis.StatInferTestAnova();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

getContentPane().setLayout(new java.awt.CardLayout());

getContentPane().add(Menu, "Menu");

getContentPane().add(StatDescrCont, "StatDescrCont");

getContentPane().add(GrCouleurRep, "GrCouleurRep");

getContentPane().add(GrCouleurComp, "GrCouleurComp");

getContentPane().add(StatInferTestConf, "StatInferTestConf");

getContentPane().add(StatInferTestHomog, "StatInferTestHomog");

getContentPane().add(StatInferTestAnova, "StatInferTestAnova");

pack();

}// </editor-fold>

public ApplicationDataAnalysis()

{

initComponents();

this.setTitle("Data Analysis");

Utility.InitialisationFlux();

// Lancement du login

(new Login(this, true)).setVisible(true);

if (!isConnected)

this.dispose();

}

public void ChangePanel(String newPanel)

{

CardLayout card = (CardLayout)this.getContentPane().getLayout();

card.show(this.getContentPane(), newPanel);

}

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(ApplicationDataAnalysis.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ApplicationDataAnalysis.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ApplicationDataAnalysis.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(ApplicationDataAnalysis.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new ApplicationDataAnalysis().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.ButtonGroup ButtonGroup;

private application\_data\_analysis.GrCouleurComp GrCouleurComp;

private application\_data\_analysis.GrCouleurRep GrCouleurRep;

private application\_data\_analysis.Menu Menu;

private application\_data\_analysis.StatDescrCont StatDescrCont;

private application\_data\_analysis.StatInferTestAnova StatInferTestAnova;

private application\_data\_analysis.StatInferTestConf StatInferTestConf;

private application\_data\_analysis.StatInferTestHomog StatInferTestHomog;

// End of variables declaration

}

## Login.java

package application\_data\_analysis;

import java.io.\*;

import java.security.\*;

import java.util.Date;

public class Login extends javax.swing.JDialog

{

public Login(java.awt.Frame parent, boolean modal)

{

super(parent, modal);

initComponents();

this.setTitle("Login");

ErrorLabel.setVisible(false);

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

LoginLabel = new javax.swing.JLabel();

PwdLabel = new javax.swing.JLabel();

LoginTF = new javax.swing.JTextField();

PwdPF = new javax.swing.JPasswordField();

ConnexionButton = new javax.swing.JButton();

ErrorLabel = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

LoginLabel.setText("Login :");

PwdLabel.setText("Mot de passe :");

ConnexionButton.setText("Connexion");

ConnexionButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

ConnexionButtonActionPerformed(evt);

}

});

ErrorLabel.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

ErrorLabel.setForeground(new java.awt.Color(255, 0, 0));

ErrorLabel.setText("Connexion refusée !");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addComponent(PwdLabel)

.addGap(18, 18, 18)

.addComponent(PwdPF, javax.swing.GroupLayout.PREFERRED\_SIZE, 150, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(LoginLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(LoginTF, javax.swing.GroupLayout.PREFERRED\_SIZE, 150, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(0, 0, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(ErrorLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(ConnexionButton)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(LoginLabel)

.addComponent(LoginTF, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(25, 25, 25)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(PwdLabel)

.addComponent(PwdPF, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 20, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(ConnexionButton)

.addComponent(ErrorLabel))

.addContainerGap())

);

pack();

}// </editor-fold>

private void ConnexionButtonActionPerformed(java.awt.event.ActionEvent evt) {

try

{

// sels

long temps = (new Date()).getTime();

double aleatoire = Math.random();

String Password = new String(PwdPF.getPassword());

// digest

MessageDigest md = MessageDigest.getInstance("SHA-1");

md.update(Password.getBytes());

ByteArrayOutputStream baos = new ByteArrayOutputStream();

DataOutputStream bdos = new DataOutputStream(baos);

bdos.writeLong(temps);

bdos.writeDouble(aleatoire);

md.update(baos.toByteArray());

byte[] pwdDigest = md.digest();

// envoi

Utility.SendMsg(ProtocolePIDEP.LOGIN, "");

Utility.dos.writeUTF(LoginTF.getText());

Utility.dos.writeLong(temps);

Utility.dos.writeDouble(aleatoire);

Utility.dos.writeInt(pwdDigest.length);

Utility.dos.write(pwdDigest);

Utility.dos.flush();

// réponse

String reponse = Utility.ReceiveMsg();

String[] parts = reponse.split("#");

if (parts[0].equals("OUI"))

{

ApplicationDataAnalysis a = (ApplicationDataAnalysis) this.getParent();

a.isConnected = true;

this.dispose();

}

else

ErrorLabel.setVisible(true);

}

catch (NoSuchAlgorithmException ex)

{

System.err.println("Login : NoSuchAlgorithmException : " + ex.getMessage());

}

catch (IOException ex)

{

System.err.println("Login : IOException : " + ex.getMessage());

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the dialog \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

Login dialog = new Login(new javax.swing.JFrame(), true);

dialog.addWindowListener(new java.awt.event.WindowAdapter() {

@Override

public void windowClosing(java.awt.event.WindowEvent e) {

System.exit(0);

}

});

dialog.setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton ConnexionButton;

private javax.swing.JLabel ErrorLabel;

private javax.swing.JLabel LoginLabel;

private javax.swing.JTextField LoginTF;

private javax.swing.JLabel PwdLabel;

private javax.swing.JPasswordField PwdPF;

// End of variables declaration

}

## Menu.java

package application\_data\_analysis;

import javax.swing.SwingUtilities;

public class Menu extends javax.swing.JPanel

{

public Menu()

{

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

TitreLabel = new javax.swing.JLabel();

StatDescrContButton = new javax.swing.JButton();

GrCouleurRepButton = new javax.swing.JButton();

GrCouleurCompButton = new javax.swing.JButton();

StatInferTestConfButton = new javax.swing.JButton();

StatInferTestHomogButton = new javax.swing.JButton();

StatInferTestAnovaButton = new javax.swing.JButton();

QuitterButton = new javax.swing.JButton();

TitreLabel.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

TitreLabel.setForeground(new java.awt.Color(0, 0, 255));

TitreLabel.setText("MENU");

StatDescrContButton.setText("Statistiques constructives descriptives : moyenne, mode, médiane, écart-type");

StatDescrContButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

StatDescrContButtonActionPerformed(evt);

}

});

GrCouleurRepButton.setText("Répartition du nombre de containers par destination (diagramme sectoriel)");

GrCouleurRepButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

GrCouleurRepButtonActionPerformed(evt);

}

});

GrCouleurCompButton.setText("Répartition du nombre de containers par destination par trimestre (histogramme)");

GrCouleurCompButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

GrCouleurCompButtonActionPerformed(evt);

}

});

StatInferTestConfButton.setText("Test d'hypothèse de conformité");

StatInferTestConfButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

StatInferTestConfButtonActionPerformed(evt);

}

});

StatInferTestHomogButton.setText("Test d'hypothèse d'homogénéité");

StatInferTestHomogButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

StatInferTestHomogButtonActionPerformed(evt);

}

});

StatInferTestAnovaButton.setText("Test d'hypothèse de type ANOVA");

StatInferTestAnovaButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

StatInferTestAnovaButtonActionPerformed(evt);

}

});

QuitterButton.setText("Quitter");

QuitterButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

QuitterButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(TitreLabel)

.addGap(251, 251, 251))

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(StatInferTestAnovaButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(QuitterButton))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(StatDescrContButton)

.addComponent(GrCouleurRepButton)

.addComponent(GrCouleurCompButton)

.addComponent(StatInferTestConfButton)

.addComponent(StatInferTestHomogButton))

.addGap(0, 111, Short.MAX\_VALUE)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(TitreLabel)

.addGap(18, 18, 18)

.addComponent(StatDescrContButton)

.addGap(18, 18, 18)

.addComponent(GrCouleurRepButton)

.addGap(18, 18, 18)

.addComponent(GrCouleurCompButton)

.addGap(18, 18, 18)

.addComponent(StatInferTestConfButton)

.addGap(18, 18, 18)

.addComponent(StatInferTestHomogButton)

.addGap(18, 18, 18)

.addComponent(StatInferTestAnovaButton)

.addContainerGap(21, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(QuitterButton)

.addContainerGap())

);

}// </editor-fold>

private void StatDescrContButtonActionPerformed(java.awt.event.ActionEvent evt) {

ApplicationDataAnalysis app = (ApplicationDataAnalysis)SwingUtilities.getWindowAncestor(this);

app.ChangePanel("StatDescrCont");

}

private void GrCouleurRepButtonActionPerformed(java.awt.event.ActionEvent evt) {

ApplicationDataAnalysis app = (ApplicationDataAnalysis)SwingUtilities.getWindowAncestor(this);

app.ChangePanel("GrCouleurRep");

}

private void GrCouleurCompButtonActionPerformed(java.awt.event.ActionEvent evt) {

ApplicationDataAnalysis app = (ApplicationDataAnalysis)SwingUtilities.getWindowAncestor(this);

app.ChangePanel("GrCouleurComp");

}

private void StatInferTestConfButtonActionPerformed(java.awt.event.ActionEvent evt) {

ApplicationDataAnalysis app = (ApplicationDataAnalysis)SwingUtilities.getWindowAncestor(this);

app.ChangePanel("StatInferTestConf");

}

private void StatInferTestHomogButtonActionPerformed(java.awt.event.ActionEvent evt) {

ApplicationDataAnalysis app = (ApplicationDataAnalysis)SwingUtilities.getWindowAncestor(this);

app.ChangePanel("StatInferTestHomog");

}

private void StatInferTestAnovaButtonActionPerformed(java.awt.event.ActionEvent evt) {

ApplicationDataAnalysis app = (ApplicationDataAnalysis)SwingUtilities.getWindowAncestor(this);

app.ChangePanel("StatInferTestAnova");

}

private void QuitterButtonActionPerformed(java.awt.event.ActionEvent evt) {

Utility.SendMsg(ProtocolePIDEP.LOGOUT, null);

ApplicationDataAnalysis app = (ApplicationDataAnalysis)SwingUtilities.getWindowAncestor(this);

app.dispose();

}

// Variables declaration - do not modify

private javax.swing.JButton GrCouleurCompButton;

private javax.swing.JButton GrCouleurRepButton;

private javax.swing.JButton QuitterButton;

private javax.swing.JButton StatDescrContButton;

private javax.swing.JButton StatInferTestAnovaButton;

private javax.swing.JButton StatInferTestConfButton;

private javax.swing.JButton StatInferTestHomogButton;

private javax.swing.JLabel TitreLabel;

// End of variables declaration

}

## GrCouleurRep.java

## GrCouleurComp.java

## StatInferTestConf.java

## StatInferTestHomog.java

## StatInferTestAnova.java

## ProtocolePIDEP.java

package application\_data\_analysis;

public class ProtocolePIDEP

{

public static final int LOGIN = 1;

public static final int GET\_STAT\_DESCR\_CONT = 2;

public static final int GET\_GR\_COULEUR\_REP = 3;

public static final int GET\_GR\_COULEUR\_COMP = 4;

public static final int GET\_STAT\_INFER\_TEST\_CONF = 5;

public static final int GET\_STAT\_INFER\_TEST\_HOMOG = 6;

public static final int GET\_STAT\_INFER\_TEST\_ANOVA = 7;

public static final int LOGOUT = 8;

}

## Utility.java

package application\_data\_analysis;

import java.io.\*;

import java.net.\*;

import java.util.Properties;

public final class Utility

{

private static DataInputStream dis;

public static DataOutputStream dos;

private static String adresse;

private static int port;

public static void InitialisationFlux()

{

FichierProperties();

try

{

ApplicationDataAnalysis.cliSock = new Socket(adresse, port);

dis = new DataInputStream(new BufferedInputStream(ApplicationDataAnalysis.cliSock.getInputStream()));

dos = new DataOutputStream(new BufferedOutputStream(ApplicationDataAnalysis.cliSock.getOutputStream()));

}

catch (IOException e)

{

System.err.println("Utility : Erreur de création de la socket, dis et dos (IO) : " + e);

}

}

private static void FichierProperties()

{

Properties prop = new Properties();

try

{

FileInputStream FIS = new FileInputStream("DataAnalysis.properties");

prop.load(FIS);

}

catch(FileNotFoundException ex)

{

try

{

FileOutputStream FOS = new FileOutputStream("DataAnalysis.properties");

prop.setProperty("Adresse", "192.168.1.4");

prop.setProperty("Port", "31049");

try

{

prop.store(FOS, null);

}

catch (IOException ex1)

{

System.err.println("Utility : Ecriture properties (IO) : " + ex1.getMessage());

System.exit(0);

}

}

catch (FileNotFoundException ex1)

{

System.err.println("Utility : Properties (FileNotFoundException) : " + ex1.getMessage());

System.exit(0);

}

}

catch(IOException ex)

{

System.err.println("Utility : Lecture properties (IO) : " + ex.getMessage());

System.exit(0);

}

adresse = prop.getProperty("Adresse");

port = Integer.parseInt(prop.getProperty("Port"));

}

public static void SendMsg(int requete, String chargeUtile)

{

chargeUtile = requete + "#" + chargeUtile;

int taille = chargeUtile.length();

String message = String.valueOf(taille) + "#" + chargeUtile;

try

{

dos.write(message.getBytes());

dos.flush();

}

catch(IOException e)

{

System.err.println("Utility : Erreur d'envoi de msg (IO) : " + e);

}

}

public static String ReceiveMsg()

{

byte b;

StringBuffer taille = new StringBuffer();

StringBuffer message = new StringBuffer();

try

{

while ((b = dis.readByte()) != (byte)'#')

{

if (b != (byte)'#')

taille.append((char)b);

}

for (int i = 0; i < Integer.parseInt(taille.toString()); i++)

{

b = dis.readByte();

message.append((char)b);

}

}

catch(IOException e)

{

System.err.println("Utility : Erreur de reception de msg (IO) : " + e);

}

return message.toString();

}

}

# Explications des 3 requêtes d’inférence statistique

## Test d’hypothèse de conformité

## Test d’hypothèse d’homogénéité

## Test d’hypothèse de type ANOVA