2016-
2017

Technilogie de l’e-commerce et mobiles

JÉRÉMY BASTIN & FLORENT CARDOEN

Code exercices élémentaires

Connexion à R avec Rserve (MainFrame.java)

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
        if(connexionR == null)
            CreateRConnection();
        else
        {
            if (connexionR.isConnected()){
                RStatus.setText("Déconnecté");
                connexionR.close();
                jButton1.setText("Se connecter");

            }
            else{
                CreateRConnection();
            }
        }

    }
    private void CreateRConnection()
    {
        try {
            connexionR = new RConnection();
            exercice1Panel1.setConnexionR(connexionR);
            exercice2Panel1.setConnexionR(connexionR);
            exercice3Panel1.setConnexionR(connexionR);
            RStatus.setText("Connecté");
            jButton1.setText("Se déconnecter");
        } catch (RserveException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }
```

Exercice 1 (Exercice1Panel.java)

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
        /* Test les variances */
        /* Test les moyennes */
        REXP response;
        Double b1, b2, e, pvalue;
        Boolean varequal;
        Double ecart = (Double) ecartSpinner.getValue();
        Double moye = (Double) moyenneSpinner.getValue();
        if(connexionR == null || !connexionR.isConnected() ||
filePath.isEmpty())
            return;

        try {
```

```
            connexionR.eval("data <- read.table('"+ filePath + "',
sep=\";\", header=FALSE)$V1");
            e = connexionR.eval("(length(data)*(mean(data)^2))/("+
ecart +"^2)").asDouble();
            b2 = connexionR.eval("qchisq(0.95, length(data)-
1)").asDouble();
            b1 = connexionR.eval("qchisq(0.05, length(data)-
1)").asDouble();

            if(e < b2 && e > b1)
            {
                varequal = true;
                varLabel.setText("Pas de différence significative
pour l'écart type " + ecart);
            }
            else
            {
                varequal = false;
                varLabel.setText("Différence significative pour
l'écart type " + ecart);
            }
            pvalue = connexionR.eval("t.test(data, mu="+moye+",
var.equal = "+ varequal.toString().toUpperCase()
+")$p.value").asDouble();
            pValueLabel.setText(pvalue.toString());
            if(pvalue > 0.05)
                conclusionLabel.setText("Il n'y a pas de différence
significative pour la moyenne");
            else
                conclusionLabel.setText("Il y a une différence
significative pour la moyenne.");

        } catch (RserveException | REXPMismatchException ex) {
            JOptionPane.showMessageDialog(this, ex.getMessage(),
"Erreur", JOptionPane.ERROR_MESSAGE);

Logger.getLogger(Exercice1Panel.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
```

Exercice 2 (Exercice2Panel.java)

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
        REXP response;
        Double b1, b2, e, pvalue;
        Boolean varequal;

        if(connexionR == null || !connexionR.isConnected() ||
filePath.isEmpty())
            return;

        try {
            connexionR.eval("data <- read.csv('"+filePath+"', sep =
';', dec = '.', header = T)");
```

```
            connexionR.eval("attach(data)");
            pvalue = connexionR.eval("var.test(Longueur~Heavea, data
= data)$p.value").asDouble();
            if(pvalue < 0.05){
                varequal = false;
                varLabel.setText("Différence significative de la
variance. Pvalue = " + pvalue.toString());
            }
            else{
                varequal = true;
                varLabel.setText("Pas différence significative de la
variance. Pvalue = " + pvalue.toString());
            }
            pvalue = connexionR.eval("t.test(Longueur~Heavea, data =
data, alternative = \"two.sided\", var.eque=" +
varequal.toString().toUpperCase()+  ")$p.value").asDouble();
            pvalueLabel.setText(pvalue.toString());
            if(pvalue < 0.05)
            {
                conclusionLabel.setText("Il y a une différence
significative des moyennes.");
            }
            else
            {
                conclusionLabel.setText("Il n'y a pas de différence
significative des moyennes.");

            }

        }  catch (RserveException | REXPMismatchException ex) {
            JOptionPane.showMessageDialog(this, ex.getMessage(),
"Erreur", JOptionPane.ERROR_MESSAGE);

Logger.getLogger(Exercice1Panel.class.getName()).log(Level.SEVERE,
null, ex);
        }


    }
```

Exercice 3 (Exercice3Panel.java)

```
    private void jButton2ActionPerformed(java.awt.event.ActionEvent
evt) {

        Double pvalue;
        if(connexionR == null || !connexionR.isConnected() ||
filePath.isEmpty())
            return;

        try {
            connexionR.eval("data <- read.csv('"+filePath+"', sep =
';', dec = '.', header = T)");
            connexionR.eval("attach(data)");
```

```
            pvalue = connexionR.eval("summary(aov(nbr ~ populas,
data=data))[[1]][[\"Pr(>F)\"]][[1]]").asDouble();
            pValue.setText(pvalue.toString());

            if(pvalue < 0.05)
            {
                conclusionLabel.setText("Il y a une différence
significative entre les population.");
            }
            else
            {
                conclusionLabel.setText("Il n'y a pas de différence
significative entre les populations.");

            }

        }  catch (RserveException | REXPMismatchException ex) {
            JOptionPane.showMessageDialog(this, ex.getMessage(),
"Erreur", JOptionPane.ERROR_MESSAGE);

Logger.getLogger(Exercice1Panel.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
```

Code du client Application_Data_Analysis

MainFrame.java

```
package client_indep;


import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;


/**
 *
 * @author florentcardoen
 */
public class MainFrame extends javax.swing.JFrame {
    private ClientINDEP client;
    private String login;
    private AuthFrame af;
    /**
```

```
     * Creates new form MainFrame
     */
    public MainFrame(String ip, Integer p) {
        initComponents();
        boolean connect = false;
        do
        {
            try {
                client = new ClientINDEP(ip, p);
                client.connect();
                connect = true;
            } catch (IOException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null,
ex);

                int n = JOptionPane.showConfirmDialog(
                        null,
                        "Impossible de se connecter au
serveur.\nVoulez-vous réessayer?",
                        "Connexion impossible",
                        JOptionPane.YES_NO_OPTION);
                if(n == JOptionPane.NO_OPTION){
                    System.exit(0);
                }
                //System.exit(1);
            }
        }while(connect == false);
        af = new AuthFrame(client, this);
        af.setVisible(true);

        dESCR_CONT1.setClient(client);
        iNFER_TEST_ANOVA1.setClient(client);
        vENTES_COMP1.setClient(client);
        vENTES_REP1.setClient(client);
        iNFER_TEST_CONF1.setClient(client);
    }
    public void setLogin(String log)
    {
```

```
        login = log;
        af.dispose();
        this.setVisible(true);
    }
    public ClientINDEP getClient()
    {
        return client;
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {

             try {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
                } catch (ClassNotFoundException |
InstantiationException | IllegalAccessException |
UnsupportedLookAndFeelException ex) {

Logger.getLogger(AuthFrame.class.getName()).log(Level.SEVERE, null,
ex);
                }
                MainFrame mf = new MainFrame("127.0.0.1", 6001);


            }
        });
    }

    // Variables declaration - do not modify
    private client_indep.DESCR_CONT dESCR_CONT1;
    private client_indep.INFER_TEST_ANOVA iNFER_TEST_ANOVA1;
    private client_indep.INFER_TEST_CONF iNFER_TEST_CONF1;
    private javax.swing.JTabbedPane jTabbedPane2;
```

```
    private client_indep.VENTES_COMP vENTES_COMP1;
    private client_indep.VENTES_REP vENTES_REP1;
    // End of variables declaration
}
```

DESRCONT.java

```
private void genButtonActionPerformed(java.awt.event.ActionEvent
evt) {
        Integer nbr;
        Message response;
        nbr = (Integer) nbrelemSpinner.getValue();

        response = client.Descr_cont(nbr);
        if((Integer)response.getParam("status") == 0)

JOptionPane.showMessageDialog(this,response.getParam("error"),"Erreu
r interne", JOptionPane.ERROR_MESSAGE);
        else
        {

System.out.println((Double)response.getParam("moyenne"));
            DefaultTableModel model = new DefaultTableModel();
            model.addColumn("Fonction");
            model.addColumn("Resultat");
            model.addRow(new Object[]{"Moyenne",
(Double)response.getParam("moyenne")});
            model.addRow(new Object[]{"Médiane",
(Double)response.getParam("mediane")});
            model.addRow(new Object[]{"Ecart Type",
(Double)response.getParam("ecart-type")});
            model.addRow(new Object[]{"Mode",
(String)response.getParam("mode")});

            resultsTable.setModel(model);
        }

    }
```

VENTES_COMP.java

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
       Integer annee, nbr;
        String str;
        Message response;
        HashMap data;

        Calendar cal = Calendar.getInstance();
        cal.setTime((Date)yearSpinner.getValue());
        annee = cal.get(Calendar.YEAR);
        if(dixRadio.isSelected()){
            nbr = 10;
        }
```

```
        else
        {
            nbr = 1;
        }

        response = client.Ventes_comp(annee, nbr);

        if((Integer)response.getParam("status") == 0)

JOptionPane.showMessageDialog(this,response.getParam("error"),"Erreu
r interne", JOptionPane.ERROR_MESSAGE);
        else
        {
            DefaultCategoryDataset ds = new
DefaultCategoryDataset();

            if(dixRadio.isSelected())
            {
                str = "Ventes des "+nbr+" dernières années jusque "
+ annee;
                for (Integer i = (annee-9); i <= annee; i++)
                {

ds.setValue((Double)response.getParam(i.toString()), "montant", i);
                }
            }
            else
            {
                str = "Ventes de l'année " + annee;
                String[] mois = {"Janvier", "Février", "Mars",
"Avril", "Mai", "Juin", "Juillet", "Aout", "Septembre","Octobre",
"Novembre","Décembre"};
                for (Integer i = 1; i <= 12; i++)
                {

ds.setValue((Double)response.getParam(i.toString()), "montant",
mois[i-1]);
                }
            }


            JFreeChart jfc = ChartFactory.createBarChart(str,
"Total", "Montant", ds, PlotOrientation.VERTICAL,true, true, false);

            ChartPanel cp = new ChartPanel(jfc);
            //chartPane = new JPanel();
            chartPane.setLayout(new java.awt.BorderLayout());
            chartPane.add(cp);
            chartPane.setVisible(true);
            chartPane.validate();

//System.out.println((Integer)response.getParam("status"));
        }
    }
```

VENTES_REP.java

```
private void genButtonActionPerformed(java.awt.event.ActionEvent
evt) {
        Integer annee, mois;
        String str;
        Message response;
        HashMap data;
        Calendar cal = Calendar.getInstance();
        cal.setTime((Date)yearSpinner.getValue());
        annee = cal.get(Calendar.YEAR);

        if(monthCheckBox.isSelected())
            mois = monthCombo.getSelectedIndex() + 1;
        else
            mois = 0;

        response = client.Ventes_rep(annee, mois);
        if((Integer)response.getParam("status") == 0)

JOptionPane.showMessageDialog(this,response.getParam("error"),"Erreu
r interne", JOptionPane.ERROR_MESSAGE);
        else
        {
            data = response.getParams();
            Iterator it = data.entrySet().iterator();
            DefaultPieDataset ds = new DefaultPieDataset();
            while (it.hasNext()) {
                Map.Entry pair = (Map.Entry)it.next();
                if(!((String)pair.getKey()).equals("status"))
                    ds.setValue((String)pair.getKey(),
(Double)pair.getValue());

                //System.out.println(pair.getKey() + " = " +
pair.getValue());
                it.remove();
            }
            str = "Représentation des ventes ";
            if(mois == 0)
                str += "de l'année "+ annee;
            else
                str += "du mois "+mois+"/"+annee;

            JFreeChart jfc = ChartFactory.createPieChart (str, ds,
true, true, true);

            ChartPanel cp = new ChartPanel(jfc);
            //chartPane = new JPanel();
            chartPane.setLayout(new java.awt.BorderLayout());
            chartPane.add(cp);
            chartPane.setVisible(true);
            chartPane.validate();

//System.out.println((Integer)response.getParam("status"));
        }
    }
```

INFER_TEST_CONF.java

```
private void genButtonActionPerformed(java.awt.event.ActionEvent
evt) {
        Integer nbr;
        Integer montant;
        Message response;
        nbr = (Integer) nbrelemSpinner1.getValue();
        montant = (Integer) montantSpinner.getValue();

        response = client.Infer_test_conf(nbr, montant);
        if((Integer)response.getParam("status") == 1)
        {
            statusLabel.setText(String.valueOf(((Double)
response.getParam("pvalue"))));
            conclusionLabel.setText((String)
response.getParam("conclusion"));
        }

    }
```

INFER_TEST_ANOVA

```
    private void genButtonActionPerformed(java.awt.event.ActionEvent
evt) {
        Integer nbr;

        Message response;
        nbr = (Integer) nbrelemSpinner1.getValue();

        response = client.Infer_test_anova(nbr);
        if((Integer)response.getParam("status") == 1)
        {
            statusLabel.setText(String.valueOf(((Double)
response.getParam("pvalue"))));
            conclusionLabel.setText((String)
response.getParam("conclusion"));
        }
        else

JOptionPane.showMessageDialog(this,response.getParam("error"),"Erreu
r interne", JOptionPane.ERROR_MESSAGE);
    }
```

AUTHFRAME.java

```
private void ActionButtonActionPerformed(java.awt.event.ActionEvent
evt) {
        String login = LoginTextField.getText();
        String password = PasswordTextField.getText();
        Message response = new Message();

        response = client.login(login, password);

        int status = (Integer) response.getParam("status");
        if(status == 1)
```

```
        {

ResponseLabel.setText((String)response.getParam("message"));
            parent.setLogin(login);
        }
        else

ResponseLabel.setText((String)response.getParam("error"));

    }
```

CLIENTINDEP.java

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package client_indep;

import server_indep.protocoleINDEP;
import client.Client;
import static java.awt.SystemColor.text;
import java.io.ByteArrayOutputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import server.Message;
import server_dismap.protocoleDISMAP;

/**
 *
 * @author bastin
 */
public class ClientINDEP extends Client implements protocoleINDEP {

    public ClientINDEP(String ip, int p) throws IOException {
        super(ip, p);

        //Demo();
    }
    public Message Descr_cont(Integer NbrElem){
        Message request = new Message();
        Message response = new Message();

        request.setType(GET_STAT_DESCR_CONT);
        request.addParam("nbrelem", NbrElem);
```

```
    sendMessage(request);
    response = receiveMessage();

    return response;

}
public Message Infer_test_conf(Integer nbrElem, Integer montant)
{
    Message request = new Message();
    Message response = new Message();

    request.setType(GET_STAT_INFER_TEST_CONF);
    request.addParam("nbrElem", nbrElem);
    request.addParam("montant", montant);


    sendMessage(request);
    response = receiveMessage();

    return response;
}
public Message Infer_test_anova(Integer nbrElem)
{
    Message request = new Message();
    Message response = new Message();

    request.setType(GET_STAT_INFER_TEST_ANOVA);
    request.addParam("nbrElem", nbrElem);

    sendMessage(request);
    response = receiveMessage();

    return response;
}
public Message Ventes_rep(Integer annee, Integer mois)
{
    Message request = new Message();
    Message response = new Message();

    request.setType(GET_GR_VENTES_REP);
    request.addParam("annee", annee);
    request.addParam("mois", mois);


    sendMessage(request);
    response = receiveMessage();

    return response;
}
public Message Ventes_comp(Integer annee, Integer nbr)
{
    Message request = new Message();
    Message response = new Message();

    request.setType(GET_GR_VENTES_COMP);
    request.addParam("annee", annee);
```

```
        request.addParam("nbr", nbr);


        sendMessage(request);
        response = receiveMessage();

        return response;
    }

    public Message login(String login, String password)
    {

        Message request = new Message();
        Message response = new Message();

        request.setType(REQUEST_LOGIN);
        request.addParam("login", login);
        try {
            MessageDigest md = MessageDigest.getInstance("SHA-256");
            md.update(login.getBytes());
            md.update(password.getBytes());
            long temps = (new Date()).getTime();

            double alea = Math.random();
            ByteArrayOutputStream baos = new
ByteArrayOutputStream();
            DataOutputStream bdos = new DataOutputStream(baos);

            bdos.writeLong(temps); bdos.writeDouble(alea);
            md.update(baos.toByteArray());
            byte[] msgD = md.digest();
            request.addParam("password", msgD);
            request.addParam("time", temps);
            request.addParam("random", alea);

        } catch (NoSuchAlgorithmException ex) {

Logger.getLogger(ClientINDEP.class.getName()).log(Level.SEVERE,
null, ex);
            return null;
        } catch (IOException ex) {

Logger.getLogger(ClientINDEP.class.getName()).log(Level.SEVERE,
null, ex);
            return null;
        }

        request.addParam("login", login);


        sendMessage(request);
        response = receiveMessage();

        return response;

    }
```

```
    public Message receiveMessage()
    {
        Message msg = null;

        try {
            msg = (Message) ois.readObject();
        } catch (IOException ex) {

Logger.getLogger(ClientINDEP.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (ClassNotFoundException ex) {

Logger.getLogger(ClientINDEP.class.getName()).log(Level.SEVERE,
null, ex);
        }

        return msg;
    }

    public void sendMessage(Message msg)
    {
        try {
            oos.writeObject(msg);
            oos.flush();
        } catch (IOException ex) {

Logger.getLogger(ClientINDEP.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }

    public static void main(String[] args) {
        try {
            new ClientINDEP("127.0.0.1", 6001);
        } catch (IOException ex) {

Logger.getLogger(ClientINDEP.class.getName()).log(Level.SEVERE,
null, ex);
            System.exit(1);
        }
    }
}
```

Explications des deux requêtes d'inférence statistique

Test d'hypothèse de conformité

Pour débuter, nous avons une première hypothèse, appelée hypothèse nulle qui est dans ce cas-ci, le chiffre d'affaires moyen mensuelle d'un vendeur « free » de notre magasin. Nous allons chercher dans la base de données un échantillon aléatoire et représentatif composé d'un *x* chiffres d'affaires mensuelles de *y* vendeurs. *x* et *y* seront défini dans l'interface graphique.

Nous allons créer une chaine de caractères qui permettra de créer le tableau avec tous les éléments de l'échantillon. Cette chaine de caractères sera envoyée à R via Rserve.

```
t.test(frameValeur, mu=MoyenneSupposée)
```

Par défaut, cette fonction fait un test bilatéral. Le but est de savoir si la moyenne supposée de l'hypothèses diffère ou pas de la valeur de l'échantillon. Nous allons prendre un seuil de risque de 5%.

La fonction t.test() de R renvoie une série de valeur dont la p-value. Celui-ci correspond au pourcentage de risque de rejeter à tort l'hypothèse nulle.

Si celle-ci est inférieure au pourcentage de risque que l'on tolère (0.05 dans notre cas), cela signifie qu'il y a peu de chance de se tromper en rejetant l'hypothèse nulle et donc on la rejette.

Par contre, si la p-value obtenue est supérieur ou égale à 0.05, alors nous ne prenons pas le risque d'avoir tort en rejetant l'hypothèse nulle. Donc, nous acceptons l'hypothèse nulle.

Test d'hypothèse de type ANOVA

Nous avons une hypothèse nulle qui dans ce cas-ci, suppose que les montant des achats est le même en fonction des zones de livraisons (villes). Nous allons chercher dans la base de données x échantillons de chaque zone. Pour pouvoir passer les données à R, il faut les formater d'une certaine façon :

1. Ville1 : 100
2. Ville2 : 150
3. Ville1 : 70
4. Ville1 : 80
5. Ville3 : 200

```
aov(y ~ A, data=frameValeur)
```

Dès que les données sont insérées correctement, Nous pouvons appeler la fonction aov(). Celle-ci renvoie aussi une p-value qui représente également le risque de rejet à tord de l'hypothèse nulle.

Si la p-value est petite nous allons donc rejeter l'hypothèse nulle c'est-à-dire que les zones n'ont pas les mêmes montants lors de leurs achats.

Tandis que si la p-value est plus grande que le risque que l'on accepte prendre, alors nous ne rejetons pas l'hypothèse nulle. Cela signifie que les zones ont les mêmes montant d'achats.

Schéma général de l'application Android