Universität Hamburg
Department Informatik
Wissenschaftliches Rechnen
Dr. J. Kunkel, H. Hatzel

Übungsblatt 1 zur Vorlesung
Big Data Analytics
im WiSe 2015/2016
Abgabe: 23.10.2015, 480 P

In diesem Übungsblatt erlernen bzw. wiederholen Sie die Grundlagen von Java, Python und R. Diese drei Programmiersprachen werden wir im weiteren Verlauf des Semesters verwenden.
Alle Aufgaben sind mit einer Punktzahl versehen welche der erwarteten Bearbeitungszeit in Minuten entspricht. 50% der möglichen Punktzahl müssen auf jedem Übungsblatt erzielt werden. Alle Übungsblätter sind in Gruppen von drei Personen zu bearbeiten und abzugeben.
Schicken Sie bis zum Abgabedatum eine Email mit den Namen der Gruppenmitglieder und den abzugebenden Artefakten in einem TAR-Archiv an: `bd-abgabe@wr.informatik.uni-hamburg.de`.
Als Subject nehmen Sie bitte die Nachnamen aller Gruppenmitglieder, bspw. KunkelHatzel. Sobald Ihre Email angekommen ist, wird diese auf die Abgabenmodalitäten (Vorhandensein und Format der notwendigen Artefakte) überprüft. Die Namen der Dateien finden Sie in den Abschnitten Abgabe von jeder Aufgabe. Bitte schickt keine vom Compiler erstellten Dateien mit!
Zur Bewertung der Vorlesung und Übung können Sie uns Feedback schicken:

`http://goo.gl/forms/B01IIDHvW2`

Bitte dokumentieren Sie die benötigte Bearbeitungszeit für die einzelnen Aufgaben. Dies dient nicht nur der Qualitätskontrolle, sondern erlaubt uns später interessante Schlussfolgerungen über die Leistungsfähigkeit der einzelnen Lösungen zu ziehen. Bitte dokumentieren Sie die benötigte Bearbeitungszeit für die einzelnen Aufgaben in `bearbeitungszeit.txt` (Format der Zeilen: `AufgabenNr Zeit`).
Sollten Probleme auftauchen, wenden Sie sich bitte an die Mailingliste der Veranstaltung:

`bd1516@wr.informatik.uni-hamburg.de`

1 Cluster-Kennung (30 P)

Im Folgenden sollen Sie sich auf dem Cluster einloggen und das Navigieren in einer Shell üben.
Bitte melden Sie sich auf dem Cluster an und machen Sie sich ein wenig mit den grundlegenden Linux-Befehlen vertraut. Informationen dazu finden Sie in unserem Wiki in der Sektion Cluster:

`http://wr.informatik.uni-hamburg.de/teaching/ressourcen/start`

Beachten Sie insbesondere den Link „Beginners' Guide".
Die folgenden konkrete Aufgaben haben Sie zu bewältigen:

1. Installieren Sie Putty oder ein vergleichbaren SSH-Client (siehe Beginners' Guide)
2. *Einloggen*

   Loggen Sie sich auf dem Cluster mit ihrem Benutzer und Passwort ein.

   Loggen Sie sich danach mit `ssh` auf dem Knoten `abu1` ein, dort werden Sie typischerweise die für die Vorlesung notwendige Pakete vorfinden.
3. *Bewegen im CLI (Command Line Interface)*

   a) Machen Sie sich mit der Verwendung von Manual-Pages vertraut: `$> man man`

   b) Lassen Sie sich den Namen des aktuellen Arbeitsverzeichnisses anzeigen: `$> man pwd`

   c) Lassen Sie sich den Inhalt Ihres Homeverzeichnisses anzeigen: `$> man ls`

d) Erzeugen Sie ein neues Verzeichnis mit dem Namen `testdir`: `$> man mkdir`

e) Ändern Sie das Arbeitsverzeichnis in das neue Verzeichnis: `$> cd testdir`

f) Lassen Sie sich noch einmal das aktuelle Arbeitsverzeichnis anzeigen.

g) Erzeugen Sie eine leere Datei mit dem Namen `testfile`: `$> man touch`

h) Benennen Sie die neue Datei um in `testfile2`: `$> man mv`

i) Kopieren Sie die umbenannte Datei in `testfile3`: `$> man cp`

j) Löschen Sie die Datei `testfile2`: `$> man rm`

Frage: Warum trägt die Manual-Page zu `cd` den Titel „POSIX Programmer's Manual" wohingegen die anderen Manual-Pages aus dieser Aufgabe den Titel „User Commands" haben? (Tipp: Sehen sie sich die Manual-Page zu `cd` genau an.)

4. *Packen eines Archiv*

a) Erstellen Sie ein Verzeichnis mit dem Namen `testarchiv`.

b) Erzeugen Sie darin eine Datei mit zufälligem Inhalt:
`$> dd if=/dev/urandom of=testarchiv/zufallsdatei bs=1k count=256`

c) Lassen Sie sich die Größe der Datei anzeigen:
`$> ls -lh testarchiv/zufallsdatei`

d) Lassen Sie sich die Größe des Verzeichnisses anzeigen:
`$> ls -ldh testarchiv`

e) Erzeugen Sie ein `tar`-Archiv, welches das Verzeichnis enthält:
`$> tar -cf testarchiv.tar testarchiv`

f) Lassen Sie sich die Größe des Archives `testarchiv.tar` ausgeben.
Was fällt Ihnen auf?

g) Komprimieren Sie das Archiv:
`$> gzip testarchiv.tar`

Das Archiv ist nun erstellt. `gzip` hat das Archiv automatisch in `testarchiv.tar.gz` umbenannt.

h) Lassen Sie sich die Größe des gepackten Archives `testarchiv.tar.gz` ausgeben.

Frage: Es ist möglich, ein gepacktes Archiv (`.tar.gz`) mit einem Aufruf von `tar` zu erzeugen? Wie hätte dieser Aufruf lauten müssen?

i) Lassen Sie sich den Inhalt des gepackten Archives ausgeben.

Abgabe:

`1-cluster-navigation.txt` D
okumentieren Sie Ihre Shell Befehle und die Antworten durch Copy&Paste der Ausgabe.

2 Hello World in Python (Python, R & Java) (150 P)

Diese Aufgabe soll dazu dienen, dass Sie eine Entwicklungsumgebung bereitstellen (bspw. Eclipse für Java, RStudio für R) und erste Schritte mit der Programmsprache unternehmen.

2.1 Python

Wir empfehlen Python mit Hilfe eines Texteditors zu entwickeln. Wenn Sie gerne eine IDE benutzen wollen steht Ihnen das natürlich frei.
Eine Einführung in Python finden sie hier: `https://developers.google.com/edu/python/`. Da wir im Verlauf der Übung für mathematische Berechnungen auch das Python Modul NumPy verwenden werden, können Sie sich auch eine Einführung in NumPy ansehen: `http://wiki.scipy.org/Tentative_NumPy_Tutorial`.

- Schreiben Sie Programm das den Text „Hello World“ ausgibt.
- Bringen Sie das Programm zur Ausführung.
- Wie lässt sich das Programm ohne expliziten Aufruf des Python-Interpreters in der Shell ausführen? (Beantworten Sie diese Frage in einem Kommentar in Ihrem Programm)

2.2 Java

Für die Entwicklung von Java Anwendungen empfehlen wir die Entwicklungsumgebung Eclipse.
Um in Zukunft auf Ihrem PC lokal Java Anwendungen für Hadoop zu entwickeln müssen Sie die entsprechenden .jar Dateien in den Buildpath ihres Projekts einfügen. Rechts klicken Sie dafür in Eclipse auf eines ihrer bereits erstellten Projekte und wählen: `Build Path > Add External Archives` fügen sie hier alle jar Dateien hinzu die sie in `/home/kunkel/bigdata/jars` auf dem Cluster finden (kopieren Sie diese vorher auf ihren lokalen Rechner).
Wir gehen davon aus das Sie schon Erfahrung mit Java haben, sollten trotzdem Probleme auftreten empfehlen wir diese Einleitung in Java: `http://heather.cs.ucdavis.edu/~matloff/Java/JavaIntro.html`
Schreiben Sie Programm das den Text „Hello World” ausgibt. Bringen Sie das Programm zur Ausführung.

2.3 R

Eine Einführung für R finden Sie hier: `http://data.princeton.edu/R/introducingR.pdf` und im Paket „swirl“ (siehe Vorlesungsfolien). Um die neueste Version von R zur Verfügung zu haben verwenden Sie den Befehl:

```
1 module load r
```

Schreiben Sie Programm, das den Text „Hello World” ausgibt. Bringen Sie das Programm zur Ausführung.

Abgabe:

`2-hello.`$(java|py|R)$ Ihre Source-Code Dateien für die entsprechende Sprache

3 CSV Daten Einlesen (Python, R & Java) (180 P)

Ein beliebtes plattformunabhängiges Dateiformat für strukturiert Daten ist CSV. In dieser Aufgabe sollen Sie in jeder der drei Sprachen eine (beliebige) CSV-Datei einlesen und den Durchschnittswert einer (auf Kommandozeile gewählten) Spalte berechnen. D.h. ihr Programm sollte sich wie folgt aufrufen lassen:
`python csv.py [DATEI] [SPALTENNR]` bzw.
`java csv.java [DATEI] [SPALTENNR]` bzw.
`Rscript csv.R [DATEI] [SPALTENNR]`
Zum Testen haben wir eine Messreihe von CSV-Daten auf dem Cluster bereitgestellt: `/home/hatzel/bigdata/pitbull-ddw1024k.csv`

Testen Sie die Geschwindigkeit, hierzu sollte ihr Programm den Mittelwert der Spalte 'Deltatime' berechnen und ausgeben. Nutzen sie das Unix-Programm `time` um zu evaluieren welches der drei Programme diese Daten am schnellsten verarbeitet.

3.1 Hinweise und Codegerüste

Erstellen Sie folgende Struktur bzw. nutzen Sie aufgezeigte Pakete bzw. Hilfsfunktionen.

3.1.1 Java

```java
// Nutzen Sie:
import java.io.BufferedReader;
import java.io.FileReader;

public class MyCSV{
  // Schreiben Sie die Funktion
  public double computeMeanCSV(string dateiname, int spaltenNr){
    ...
  }

  public static void main (String[] args) {
    // Die Argumente sind in args verfügbar
  }
}
```

3.1.2 Python

```python
import sys

# Die Argumente sind im Array sys.argv verfügbar
print(sys.argv)

# Schreiben sie die Funktionen
def computeMeanCSV(dateiname, spaltenNr):
  ...
  return mean
```

3.1.3 R

```r
# Argumente sind in CommandArgs verfügbar
args = commandArgs(trailingOnly = TRUE)

# schreiben Sie die Funktion
computeMeanCSV = function(dateiname, spaltenNr){
  ...
}
```

Abgabe:

`3-csv.`$(java\|py\|R)$	Ihren jeweiligen Source-Code
`3-csv-speed.txt`	Ihre Zeitmessung sowie eine kurze Einschätzung zu den Zeitunterschieden

4 Wortanalyse der Wikipedia (Java & Python) (120 P)

Ihre Programme sollen die Häufigkeit eines gegeben Begriffs (übergeben als Kommandozeilenparameter) in dem von uns gegebenen Ausschnitts der Wikipedia finden (auf dem Cluster zu finden unter `/home/hatzel/bigdata/wiki pedia-text.csv`). Endresultat soll ein Bild sein das in den Pixeln die Häufigkeit des Begriffs in den einzelnen Artikeln zeigt. Dabei soll jeder Pixel des Bildes einem Artikel entsprechen (d.h. Pixel 1 entspricht dem ersten Artikel).

Abbildung 1: Visualisierung der Vorkommnisse des Worts „data” im gegeben Datensatz.

Skalieren sie die Dunkelheit des Pixels linear mit der Anzahl der Nennungen des Worts in einem Artikel, wobei die Farbei Weiß bedeutet, dass dies der Artikel ist in dem das Wort am häufigsten vorkommt und schwarz angibt das das Wort nicht im jeweiligen Artikel vorkommt. Folgen Sie der gleichen Reihenfolge der Pixel wie die Artikel in der gegebene CSV Datei sortiert sind.
Ein Beispiel dafür wie Ihr Bild aussehen soll finden Sie in Abbildung 1. *Hinweis:* Je nach Zoom-Stufe und PDF-Reader kann es vorkommen das die einzelnen Pixel verschwommen wirken. Probieren Sie in diesem Fall einfach andere Zoomstufen oder andere Programme.
Um zu vermeiden, dass die Artikel mit den meisten Wörtern automatisch jene mit den meisten Wortnennungen sind, überlegen Sie sich ein einfaches Verfahren um die Häufigkeiten in den Artikel zu normalisieren.

4.1 Datensatz: wikipedia-text.csv

Die Datei enthält Artikel der Wikipedia mit ihren Titeln und dem Volltext (sowie dem letzten Änderungsdatum). Die Datei wurden basierend auf frei verfügbaren Datenbankdumps der Wikipedia von `https://dumps.wikimedia.org/` erstellt.
Der Header der CSV-Datei ist wie folgt aufgebaut.

```
1 ArtikelID,Titel,Änderungsdatum,RoherText
```

Ein Auszug aus der Datei:

```
"125515","Union Beach, New Jersey","2015-05-07T14:43:48Z","{{Infobox settlement|name                     = Union
    ↪ Beach....}}"
"125516","Upper Freehold Township, New Jersey","2015-04-26T05:16:05Z","{{Infobox settlement|name
    ↪ = Upper Freehold....}}" Township
"125517","Wall Township, New Jersey","2015-05-25T03:23:22Z","{{Infobox settlement|name                      = Wall
    ↪ Township....}}"
"125518","Wanamassa, New Jersey","2015-04-20T05:42:53Z","{{Infobox settlement|official_name           = Wanamassa
    ↪ ....}}"
```

4.2 Hinweise und Codegerüste

4.2.1 Python

```python
# Wir verwenden das Modul pypng https://pypi.python.org/pypi/pypng
import png
import math

# Eine Liste mit den Worthäufigkeiten für jeden Artikel (hier 10 Artikel)
count_data = [1,100,0,0,255,28,0,0,2,10]

# Die Funktion erstellt eine quadratische N*N Matrix von der übergebenen Liste
def make_square(list_, default_value=0):
    width = int(math.ceil(math.sqrt(len(list_))))
    # Daten in Zeilen aufteilen
    result = [list_[x:x + width] for x in range(0,len(list_), width)]
    # Letzte Zeile mit Standardwerten auffüllen
    result[-1] += [default_value for _ in range(len(result[-1]), width)]
    return result

count_data = make_square(count_data)
with open("mypngfile" + ".png", "wb") as picture_file:
    picture = png.Writer(len(count_data), len(count_data), greyscale=True)
    picture.write(picture_file, count_data)
```

4.2.2 Java

```java
import java.awt.Color;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.util.Arrays;
import java.util.List;

# Stellt Methoden zur Verfügung für die Bild Ein-/Ausgabe
import javax.imageio.*;

public class writer {

  public static void main(String[] args) {
    List<Integer> counts = Arrays.asList(1,2,2,255,4,21,3,123,1,55);

    int width = (int) Math.ceil(Math.sqrt(counts.size()));
    int height = (int) Math.sqrt(counts.size());
    BufferedImage image = new BufferedImage(width, height, BufferedImage.TYPE_BYTE_GRAY);
    int i = 0;
    for (int y = 0; y < height; y++){
      for (int x = 0; x < width; x++){
        if (i < counts.size()){
          int c = counts.get(i);
          image.setRGB(x,y, new Color(c, c, c).getRGB());
        } else {
```

```
          image.setRGB(x, y, new Color(0, 0, 0).getRGB());
        }
        i++;
      }
    }
    try {
      File image_file = new File("mypngfile.png");
      ImageIO.write(image, "png", image_file);
    } catch (IOException e) {
      System.out.println("IO error :(");
    }

  }

}
```

4.2.3 Worthäufigkeiten in Python

Diese Teilaufgabe ist nur in Python zu implementieren.
Ermitteln Sie die Häufigkeit aller Wörter in den jeweiligen Artikeln sowie die jeweilige Länge und speichern Sie diese in einer CSV-Datei. Dabei soll jedes mindestens einem Artikel vorkommende Wort in einer Spalte der CSV Datei zu finden sein, d.h. die Zeilen der CSV-Datei sind
Artikelname, Artikellänge, Häufigkeit Wort1, Wort2, ... WortN
Der Header der Datei spezifiziert hierbei die einzelnen Worte.
Beispielheader: Artikelname, Artikellänge, Der, Die, Das, Computer, Auto, ...

Erstellen Sie zusätzlich ein Aggregat der Häufigkeiten über alle Artikel (d.h. die Worthäufigkeiten für jedes einzelne Wort im gesamten Datensatz), dies kann in einem zweiten Schritt und in einer gesonderten CSV Datei passieren.
Verarbeiten Sie hierbei nur ein Subset des zur Verfügung gestellten Datensatzes (1000 Zeilen), nutzen Sie dafür das Programm `head` um die ersten 1000 Zeilen der CSV-Datei zu verarbeiten.
Warum ist eine CSV Datei nur begrenzt geeignet um diese Daten abzuspeichern? (Dokumentieren Sie Ihre Antwort als Kommentar im Source-Code)

Abgabe:

`4-wiki-picture.`$(java\|py)$	Ihren jeweiligen Source-Code
`4-wiki-csv.`(py)	Ihr Source-Code für Aufgabe 4.1.3.
`4-wiki-picture.png`	Ihr erstelltes Bild für die ersten 1000 Zeilen der CSV-Datei.