



UTIU
UNIVERSITÀ TELEMATICA
INTERNAZIONALE **UNINETTUNO**

Corso di Ingegneria Informatica
A.A. 2023/2024

**PROGETTO CONTABILITÀ DITTA
STECCAPARAPERTUTTI S.R.L.**

| | |
|----------------------|------------------------------|
| Nome dello studente: | <i>Manuel Frasca</i> |
| Matricola | <i>744HHHINGINFOR</i> |
| Data Appello | <i>Ottobre 2023</i> |
| Esame | <i>Introduzione Big Data</i> |

Indice

| | |
|--|--------|
| Premessa / Introduzione | Pag 3 |
| <ul style="list-style-type: none">• Lettera della Ditta• Ambiente di sviluppo• R• Tool MapReduce | |
| Svolgimento | Pag 9 |
| <ul style="list-style-type: none">• Analisi del problema• Job1• Job2• Job3• Job4 | |
| Conclusioni | Pag 19 |
| <ul style="list-style-type: none">• Risultati ottenuti• Confronto MapReduce con R• Analisi Dati su Previsioni Future | |
| Riferimento bibliografici, Sitografia | Pag 22 |
| Appendice | Pag 23 |
| <ul style="list-style-type: none">• Job1: Codice R, Codice MapReduce e Grafici• Job2: Codice R, Codice MapReduce e Grafici• Job3: Codice R, Codice MapReduce e Grafici• Job4: Codice R, Codice MapReduce e Grafici• Codice R Previsione Futura e Grafici | |

Premessa / Introduzione

Lettera della Ditta

La presente relazione descrive le modalità di svolgimento del progetto per il corso di Introduzione ai Big Data con la seguente traccia:

La ditta STECCAPARAPETUTTI S.R.L. ha un sistema di contabilità che memorizza tutti gli ordini

fatti (per ordini si intendono Fatture, Buoni Prelievo, Offerte...). In previsione dell'anno

successivo, la ditta vorrebbe sapere il suo andamento di vendita.

Scaricato il file Ordini.csv strutturato in questa maniera:

Tipo-documento,data(aaaammgg),costo(€)

FATTURA, 20160104,139.8

FATTURA, 20160104,169.2

FATTURA, 20160104,65.3

.....

Progettare e realizzare per ogni anno i seguenti job:

1. Calcola la media di vendita per ogni mese di ogni anno
2. Calcola la varianza di vendita per ogni mese di ogni anno
3. Identifica il mese di ogni anno con maggiore vendita
4. Identifica il mese di ogni anno con minore vendita

Per ciascun job bisogna illustrare e documentare in un rapporto finale:

- Implementazione R
- Implementazione del paradigma mapReduce di HADOOP.

L'obiettivo è quello di progettare 4 jobs, che elaborando i dati sugli ordini effettuati dal 2016 al 2019, fornisca informazioni sull'andamento di vendita per l'anno successivo.

Ambiente di sviluppo

L'ambiente di sviluppo in termini di Hardware è composto da un PC Portatile ASUS VivoBook 15 con le seguenti specifiche tecniche:

- Intel Core i7-9750H 6 x 2.6 – 4.5 GHz, Coffee Lake-H
- Scheda grafica NVIDIA GeForce GTX 1650 Mobile – 4 GB VRAM, GDDR5
- Memoria RAM 8 GB , DDR4-2666
- Hard Disk 256 GB SSD

In termini di Software si lavora con un Sistema Operativo Ubuntu 22.04.3 LTS a 64 bit.

```
manuel@Manuel-PC: ~$ sudo lshw -short
[sudo] password di manuel:
Percorso H/W    Dispositivo    Classe    Descrizione
=====
/0              system         VivoBook_ASUSLaptop_X571GT_F571G
/0/0            bus            X571GT
/0/0            memory         64KiB BIOS
/0/3a           memory         8GiB Memoria di sistema
/0/3a/0         memory         4GiB SODIMM DDR4 Sincrono 2667 M
/0/3a/1         memory         Project-Id-Version: lshwReport-M
/0/3a/2         memory         4GiB SODIMM DDR4 Sincrono 2667 M
/0/3a/3         memory         Project-Id-Version: lshwReport-M
/0/46           memory         384KiB L1 cache
/0/47           memory         1536KiB L2 cache
/0/48           memory         12MiB L3 cache
/0/49           processor      Intel(R) Core(TM) i7-9750H CPU @
/0/100          bridge         8th Gen Core Processor Host Brld
/0/100/1        bridge         6th-10th Gen Core Processor PCIe
/0/100/1/0      display        TU117M [GeForce GTX 1650 Mobile
/0/100/2        /dev/fb0      display        CoffeeLake-H GT2 [UHD Graphics 6
/0/100/4        generic        Xeon E3-1200 v5/E3-1500 v5/6th G
/0/100/8        generic        Xeon E3-1200 v5/v6 / E3-1500 v5
/0/100/12       generic        Cannon Lake PCH Thermal Controll
/0/100/14       bus            Cannon Lake PCH USB 3.1 xHCI Hos
/0/100/14/0     usb1           bus            xHCI Host Controller
/0/100/14/0/6   mmc0           bus            USB2.0-CRW
/0/100/14/0/7   input9         multimedia     USB2.0 HD UVC WebCam: USB2.0 HD
/0/100/14/0/e   communication   Bluetooth 9460/9560 Jefferson Pe
/0/100/14/1     usb2           bus            xHCI Host Controller
/0/100/14.2     memory         RAM memory
/0/100/14.3     wlo1           network        Cannon Lake PCH CNVL WiFi
/0/100/15       bus            Cannon Lake PCH Serial IO I2C Co
/0/100/16       communication   Cannon Lake PCH HECI Controller
/0/100/17       storage        82801 Mobile SATA Controller [RA
/0/100/1d       bridge         Cannon Lake PCH PCI Express Root
/0/100/1d/0     /dev/nvme0     storage        HFM256GDJTG-8310A
/0/100/1d/0/0   hwmn3          disk           NVMe disk
/0/100/1d/0/2   /dev/ng0n1     disk           NVMe disk
/0/100/1d/0/1   /dev/nvme0n1   disk           256GB NVMe disk
/0/100/1d/0/1/1 /dev/nvme0n1p1 volume         511MiB Windows FAT volume
/0/100/1d/0/1/2 /dev/nvme0n1p2 volume         237GiB EXT4 volume
/0/100/1d.6     bridge         Cannon Lake PCH PCI Express Root
/0/100/1d.6/0   enp3s0         network        RTL8111/8168/8411 PCI Express Gl
/0/100/1e       communication   Cannon Lake PCH Serial IO UART H
/0/100/1e.3     bus            Cannon Lake PCH SPI Host Control
/0/100/1f       bridge         HM470 Chipset LPC/eSPI Controlle
/0/100/1f/0     system         PnP device PNP0c02
/0/100/1f/1     system         PnP device PNP0c02
/0/100/1f/2     system         PnP device PNP0b00
/0/100/1f/3     generic        PnP device INT3f0d
/0/100/1f/4     input          PnP device PNP0303
/0/100/1f/5     system         PnP device PNP0c02
/0/100/1f/6     system         PnP device PNP0c02
/0/100/1f/7     system         PnP device PNP0c02
/0/100/1f/8     system         PnP device PNP0c02
/0/100/1f.3     card0          multimedia     Cannon Lake PCH cAVS
```

R

Per la scrittura del codice in R è stato utilizzato l'IDE Visual Studio Code:

Version: 1.82.2

Commit: abd2f3db4bdb28f9e95536dfa84d8479f1eb312d

Date: 2023-09-14T05:51:20.981Z

Electron: 25.8.1

ElectronBuildId: 23779380

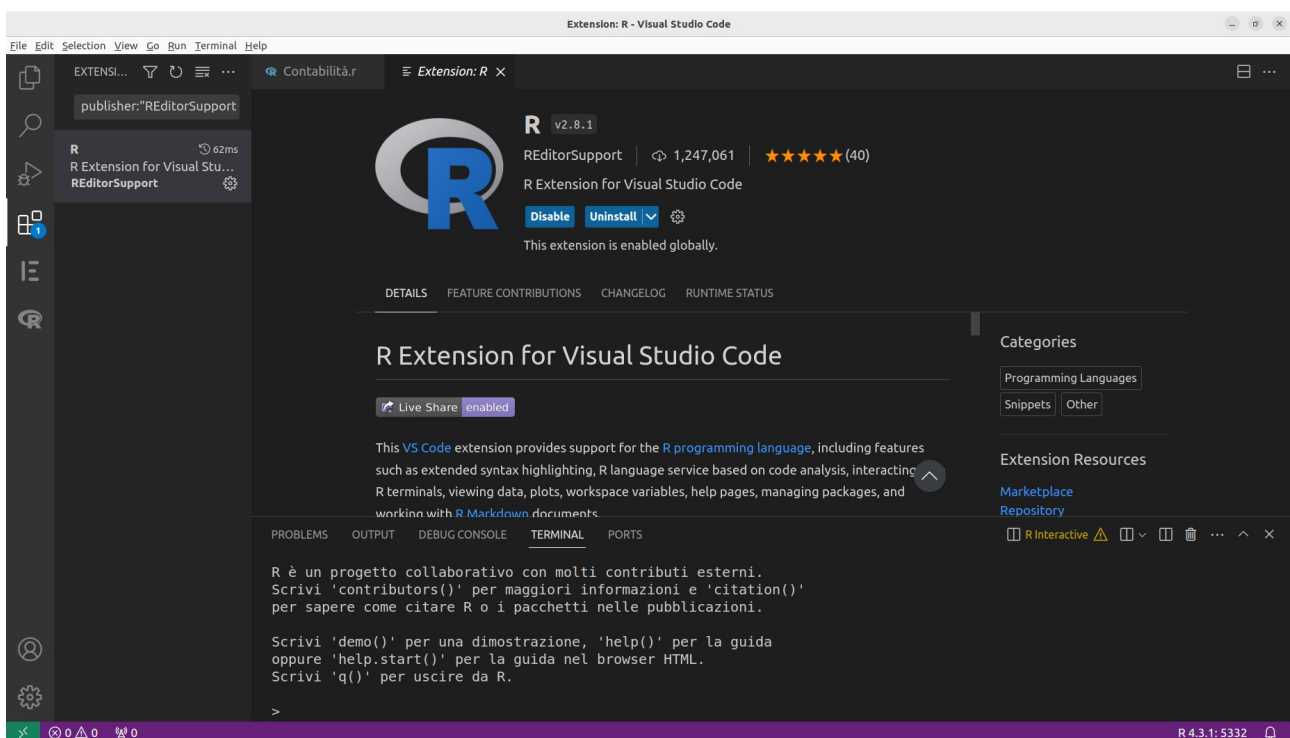
Chromium: 114.0.5735.289

Node.js: 18.15.0

V8: 11.4.183.29-electron.0

OS: Linux x64 6.2.0-32-generic.

Utilizzando l'estensione di REditorSupport è stato possibile programmare in R su Visual Studio Code.



Essendo pratico con il linguaggio SQL ho installato tramite il comando **`install.packages("sqldf")`** la libreria dove è possibile utilizzare R con le Query SQL.

Il tool MapReduce

Il tool è sviluppato in javascript con interfaccia HTML e permette di simulare un programma MapReduce attraverso un generico Browser. Il codice è rilasciato con licenza GNU, quindi è possibile utilizzarlo e modificarlo secondo necessità.

Il tool è stato realizzato per scopi didattici, dunque occorre precisarne le limitazioni:

- simula il paradigma di MapReduce ma in realtà lavora in modalità stand alone non distribuita;

- permette di utilizzare file di input di dimensioni limitate che dipendono dalla macchina su cui viene effettuato il run;

- gestisce in input soltanto file di testo.

Il tool può essere scaricato dalla sezione “Libri e Articoli” del corso di “Introduzione ai Big Data” dell’Università Uninettuno.

Una volta scaricato il file Map-reduce_tool.zip basta decomprimere la cartella e aprire via browser il file MapReduce.html nella folder “Free License Software”.

Upload Text File or drop below

input_str : Insert your dataset or drop here...

LOAD JOB MAP REDUCE

```
// First Element splits Key|Value Second element splits the values of Value if it is an array
var S=["|","|","|"];
```

```
var S = ["|","|","|"];
// => WRITE YOUR CODE HERE <=
```

```
function jobInputSplit ( input_str ) // it takes a text string from document.getElementById('input_text_area').value and it convert in array
```

```
// input_str => document.getElementById('input_text_area').value;
// => WRITE YOUR CODE HERE <=
```

```
function jobMap ( V_In_Map ) //from array return an array [k|v , k1|v1 , ..., kn|vn]
```

```
return V_In_Map.map(function(item){
    // => WRITE YOUR CODE HERE <=
});
```

```
function jobReduce ( K_In_Reduce_V_In_Reduce ) //processing of array [k|v , k1|v1 , ..., kn|vn]
```

```
return K_In_Reduce_V_In_Reduce.map(function (items){
    var K_In_Reduce = items.split(S[0])[0];
    var V_In_Reduce = items.split(S[0])[1].split(S[1]);
    var Reduce = V_In_Reduce.reduce(function (accumulator, item) {
        // => WRITE YOUR CODE HERE <=
    });
    // => WRITE YOUR CODE HERE <=
});
```

Nella text area è possibile inserire il testo di input o semplicemente fare drag and drop di un file per copiarne il contenuto.

Nella sezione function `jobInputSplit (input_str)` utilizziamo la variabile preimpostata “`input_str`” che conterrà il valore della text area descritta al punto precedente. Dunque in questa fase è possibile utilizzare la funzione di `split` per gestire eventuali caratteri di separazione (ad esempio un carattere di invio che separa righe dell’input o il carattere di pipe come nell’esempio). Essa quindi restituirà un array di elementi della sezione di input a cui è stato applicata la funzione di `split`.

Nella sezione function `jobMap (V_In_Map)` scriveremo il codice relativo alla funzione di `map`.

Prende in input l’array generato dalla funzione precedentemente descritta e ritorna una prima lista di coppie key-value.

Nell’ultima sezione function `jobReduce (K_In_Reduce, V_In_Reduce)` scriveremo il codice relativo al job di `reduce`. La funzione prende in input le coppie key-value della fase di `map`, quindi generiamo le nuove key-value facendo gli opportuni calcoli sui valori per applicare la `reduce`.

Tramite il pulsante “LOAD JOB MAP REDUCE” si carica il codice e si avvia il run con il pulsante “Run MapReduce or CTRL+ENTER” (se non sono stati esplicitati errori di sintassi nel codice). Al termine verrà mostrato un nuovo div contenente i risultati di tutte le fasi della simulazione del job MapReduce.

Per ulteriori informazioni è possibile consultare la guida utente presente all’interno della cartella zip e la classe interattiva n. 20.1 del corso.

| Upload Text File or drop below | | | | |
|---|-----------------|-----------------|-----------------|-----------------|
| 1 2 1 3 2 4 5 6 8 9 8 7 4 3 3 8 2 | | | | |
| Script loaded (show/hide) [upload (0)] | | | | |
| Run MapReduce or CTRL+ENTER | | | Reload Page | |
| <div>Hadoop Map Reduce concept-simulation made by Pirrone Daniele</div> | | MAP | REDUCE | |
| INPUT-SPLITTING | MAPPING | SHUFFLING | MERGING | REDUCING |
| « VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » |
| « 1 » | « 1 - 1 » | « 1 - 1 » | « 1 - 1, 1 » | « 1 - 2 » |
| « 2 » | « 2 - 1 » | « 1 - 1 » | « 2 - 1, 1, 1 » | « 2 - 3 » |
| « 1 » | « 1 - 1 » | « 2 - 1 » | « 3 - 1, 1, 1 » | « 3 - 3 » |
| « 3 » | « 3 - 1 » | « 2 - 1 » | « 4 - 1, 1 » | « 4 - 2 » |
| « 2 » | « 2 - 1 » | « 2 - 1 » | « 5 - 1 » | « 5 - 1 » |
| « 4 » | « 4 - 1 » | « 3 - 1 » | « 6 - 1 » | « 6 - 1 » |
| « 5 » | « 5 - 1 » | « 3 - 1 » | « 7 - 1 » | « 7 - 1 » |
| « 6 » | « 6 - 1 » | « 3 - 1 » | « 8 - 1, 1, 1 » | « 8 - 3 » |
| « 8 » | « 8 - 1 » | « 4 - 1 » | « 9 - 1 » | « 9 - 1 » |
| « 9 » | « 9 - 1 » | « 4 - 1 » | | |
| « 8 » | « 8 - 1 » | « 5 - 1 » | | |
| « 7 » | « 7 - 1 » | « 6 - 1 » | | |
| « 4 » | « 4 - 1 » | « 7 - 1 » | | |
| « 3 » | « 3 - 1 » | « 8 - 1 » | | |
| « 3 » | « 3 - 1 » | « 8 - 1 » | | |
| « 8 » | « 8 - 1 » | « 8 - 1 » | | |
| « 2 » | « 2 - 1 » | « 9 - 1 » | | |

E' stato preferito utilizzare questo Tool sviluppato dal tutor del corso piuttosto che utilizzare Hadoop perché non possedendo un Cluster di PC per l'elaborazione dati sarebbe stato inutile andare a configurare ed installare Hadoop per poterlo utilizzare solamente in locale.

Ad ogni modo, per questo tipo di progetto, il Tool e le risorse computazionali a disposizione risultano più che sufficienti.

Svolgimento

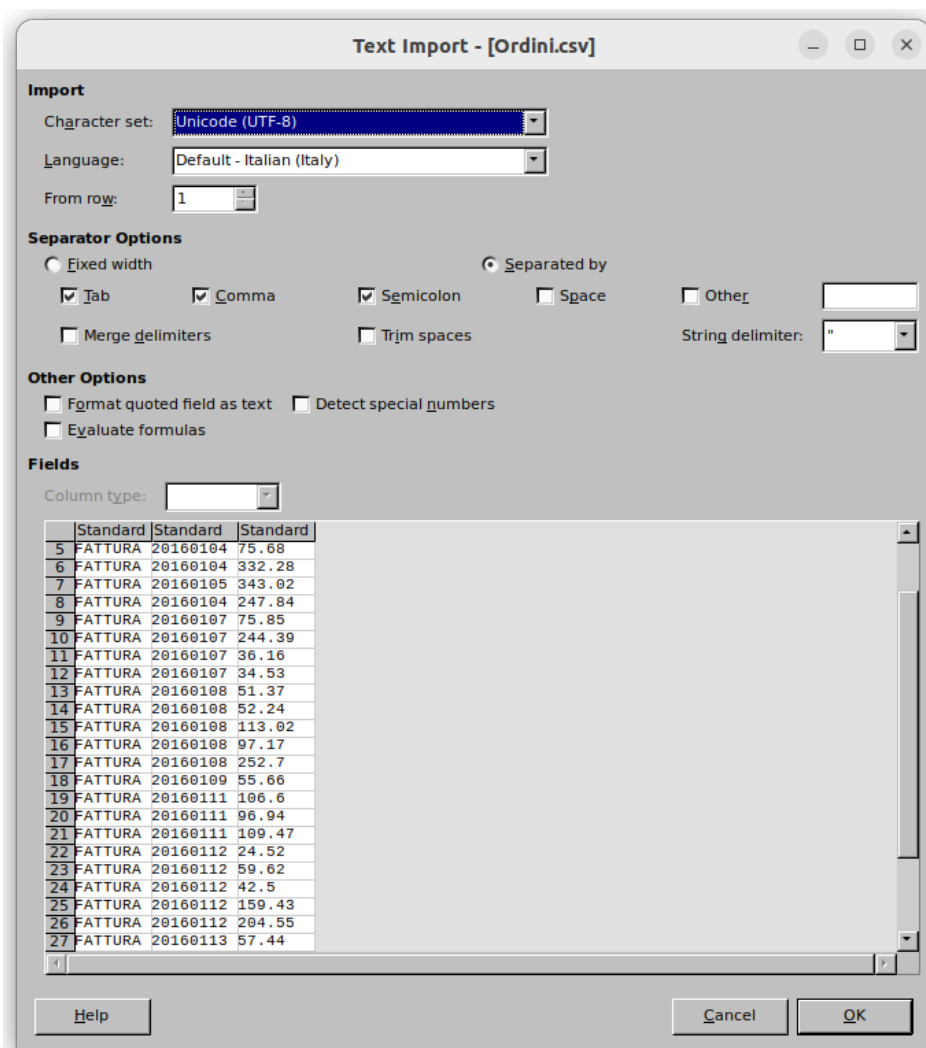
Analisi del Problema

Prima di progettare i jobs richiesti è stata effettuata un'analisi dei dati a disposizione. Il file in input è nel formato csv (comma-separated values) e raccoglie un elenco di ordini incolonnati e distinti da un ritorno a capo.

Ogni riga si riferisce ad un ordine e contiene le seguenti informazioni separate da virgola:

- Tipo-documento, identifica la tipologia di ordine e assume uno dei seguenti valori: Buono prelievo, DDT, Fattura, Inventario, Offerta, Nota di credito, Ricevuta, Preventivo;
- Data, identifica la data in cui è stato effettuato l'ordine espresso nel formato YYYYMMGG;
- Costo, identifica l'importo dell'ordine espresso in valore numerico con doppia cifra decimale.

In seguito un estratto del file Ordini.csv



Essendo richiesti calcoli sulle vendite è stato necessario individuare le tipologie di ordini utili al calcolo delle stesse.

Attraverso la seguente Query in R con l'IDE VSCode è stato possibile individuare tutte le tipologie di dato presenti all'interno della prima colonna, al fine di considerare i dati necessari all'Analisi:

```
install.packages("sqldf")  
library(sqldf)  
file_contabilità <- read.csv("/home/manuel/Scrivania/BigData/HW/Ordini.csv",  
header = FALSE)  
names (file_contabilità) <- c("Tipologia", "Data", "Prezzo")  
risultato = sqldf("  
SELECT  
    Tipologia  
FROM  
    file_contabilità  
GROUP BY  
    Tipologia  
")  
print(risultato)
```

Risultato del print():

```
1 BUONO.PRELIEVO  
2     DDT  
3     FATTURA  
4     INVENTARIO  
5 NOTA.DI.CREDITO  
6     OFFERTA  
7     PREVENTIVO  
8     RICEVUTA
```

Si individuano come tipologie di documento valide a rappresentare una vendita le seguenti: Fattura e Ricevuta.

All'interno è presente anche la tipologia "Nota di Credito".

Una nota di credito deve essere emessa entro un anno dall'operazione alla quale si riferisce, ma questa informazione (il riferimento alla fattura) non è contenuta nel file a disposizione.

Considerando quindi che non è possibile sapere a quale fattura una nota di credito si riferisce e di conseguenza non è possibile scalare l'importo al periodo di competenza, si sceglie di non considerare la tipologia Nota di credito come documento da includere per il computo delle vendite.

Quindi i valori da prendere in considerazione sono FATTURA e RICEVUTA.

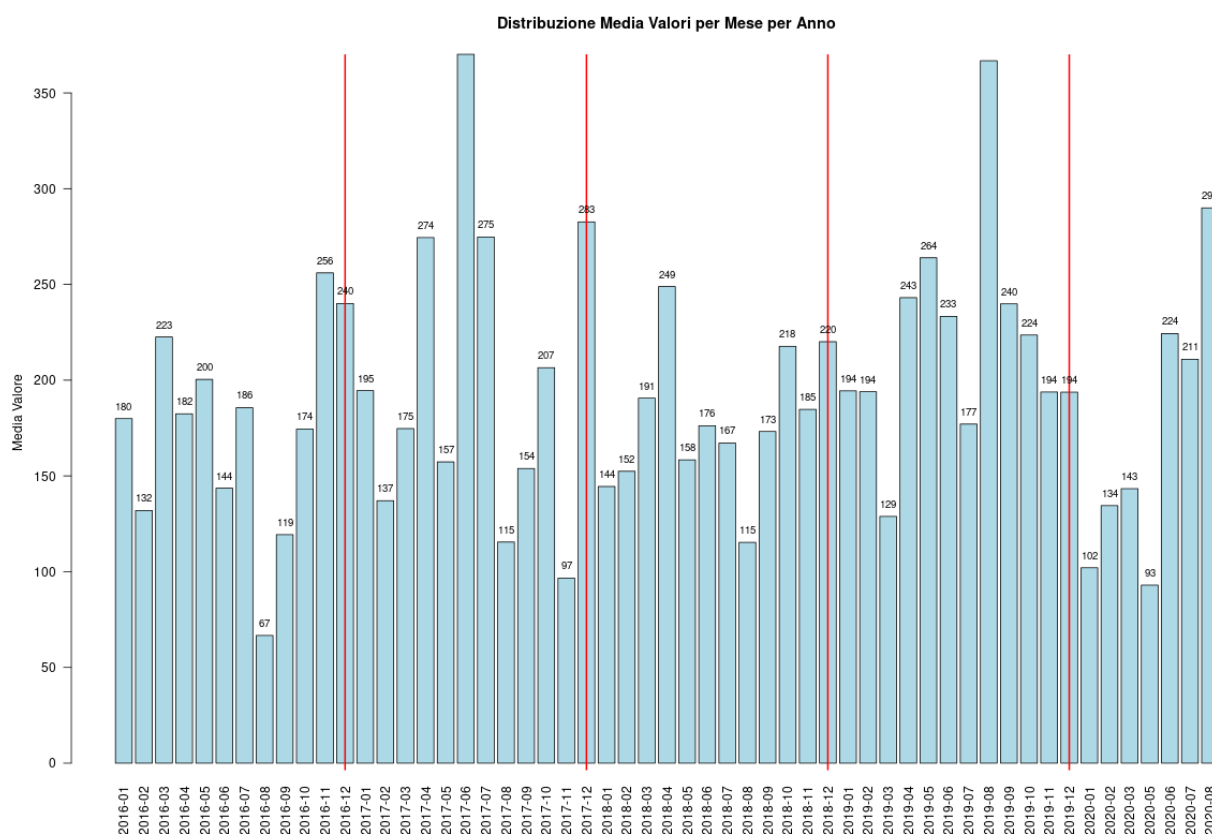
Job 1

Problema: Calcola la media di vendita per ogni mese di ogni anno.

Implementazione in R

1. Avvio libreria SQLite
2. Lettura file .csv da Directory
3. Aggiungo le intestazioni alle colonne nominandole Tipologia, Data e Prezzo
4. Avvio Query SQL
5. Raggruppa per Anno e Mese
6. Seleziona Media su Prezzo con Anno e Mese dal file dove nella colonna nominata Tipologia è presente il valore “FATTURA” o “RICEVUTA”
7. Ordina per Anno crescente e Mese crescente
8. Stampa Output

Il grafico risultante è il seguente:



L'implementazione del codice nel linguaggio R è riportata in Appendice Job1(ogni riga di codice è commentata per spiegarne il funzionamento), oltre al file Job1.csv che è il file prodotto da questo Job.

Implementazione in MapReduce

Una possibile soluzione in Map Reduce prevede queste fasi:

1. Leggo l'input
2. Si separa l'input per righe e filtro per tipo di documento "FATTURA" e "RICEVUTA"
3. Trasformo la data dal formato YYYYMMGG in YYYYMM
4. Imposto la data (YYYYMM) come K_Out_Map e l'importo come V_Out_Map ed eseguo la fase di mapping
5. La fase di shuffling ordina automaticamente l'insieme di coppie key-value
6. In fase di merging verranno uniti automaticamente tutti gli importi sotto la stessa key (YYYYMM)
7. Per ogni key si calcola la media (conoscendo la dimensione dell'array contenente gli importi per ogni key) ed effettuiamo il reducing
8. Visualizzo l'output

Una parte dell'output risultante è il seguente:

Upload Text File or drop below

OFFERTA, 20200810, 72.69
FATTURA, 20200810, 112.51
FATTURA, 20200810, 223.55
BUONO.PRELIEVO, 20200810, 58.56

Script loaded (show/hide)

[upload (1)]

Run MapReduce or CTRL+ENTER

Reload Page

Hadoop Map Reduce concept-simulation made by Pirrone Daniele

MAP

REDUCE

| INPUT-SPLITTING | MAPPING | SHUFFLING | MERGING | REDUCING |
|-----------------------------|---------------------|---------------------|---|---------------------|
| « VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » |
| « FATTURA,20160104,139.8 » | « 201601 - 139.8 » | « 201601 - 106.6 » | « 201601 - 106.6, 108.02, 109.47, 11.13, 11.37, » | « 201601 - 179.93 » |
| » | « 201601 - 160.92 » | « 201601 - 108.02 » | » | « 201602 - 131.92 » |
| « FATTURA,20160104,160.92 » | « 201601 - 65.03 » | « 201601 - 109.47 » | 113.02, 13.81, » | « 201603 - 222.56 » |
| » | « 201601 - 535.84 » | « 201601 - 11.13 » | 132.2, 139.8, » | « 201604 - 182.44 » |
| « FATTURA,20160104,65.03 » | « 201601 - 332.28 » | « 201601 - 113.02 » | 145.85, 15.4, » | « 201606 - 143.58 » |
| » | « 201601 - 343.02 » | « 201601 - 13.81 » | 159.43, 160.65, » | « 201607 - 185.63 » |
| « FATTURA,20160104,535.84 » | « 201601 - 247.84 » | « 201601 - 132.2 » | 160.92, 177.71, » | « 201608 - 66.67 » |
| » | « 201601 - 75.85 » | « 201601 - 139.8 » | 179.19, 180.28, » | « 201609 - 119.33 » |
| « FATTURA,20160104,75.68 » | « 201601 - 244.39 » | « 201601 - 145.85 » | 19.15, 204.55, 22.4, » | « 201610 - 174.44 » |
| » | « 201601 - 36.16 » | « 201601 - 15.4 » | 226.79, 235.42, » | « 201611 - 256.15 » |
| « FATTURA,20160104,75.68 » | « 201601 - 34.53 » | « 201601 - 159.43 » | 24.52, 244, 244.39, » | « 201612 - 250.21 » |
| » | « 201601 - 51.37 » | « 201601 - 160.65 » | 247.21, 247.84, » | « 201701 - 194.52 » |
| « FATTURA,20160104,332.28 » | « 201601 - 52.24 » | « 201601 - 160.92 » | 252.44, 252.7, » | « 201702 - 137.06 » |
| » | « 201601 - 113.02 » | « 201601 - 177.71 » | 2537.6, 26.94, » | « 201703 - 174.73 » |
| « FATTURA,20160105,343.02 » | « 201601 - 97.17 » | « 201601 - 179.19 » | 271.83, 294.92, » | « 201704 - 274.47 » |
| » | « 201601 - 252.7 » | « 201601 - 180.28 » | 32.37, 332.28, » | « 201705 - 157.32 » |
| « FATTURA,20160107,36.16 » | « 201601 - 55.66 » | « 201601 - 19.15 » | 34.53, 343.02, » | « 201706 - 370.18 » |
| » | « 201601 - 106.6 » | « 201601 - 204.55 » | 35.49, 36.16, 36.88, » | « 201707 - 274.78 » |
| « FATTURA,20160104,247.84 » | « 201601 - 96.94 » | « 201601 - 22.4 » | 360.17, 37.05, 42.5, » | « 201708 - 120.59 » |
| » | « 201601 - 109.47 » | « 201601 - 226.79 » | 503.81, 51.37, » | « 201709 - 157.89 » |
| « FATTURA,20160107,75.85 » | « 201601 - 24.52 » | « 201601 - 235.42 » | 52.24, 53.94, » | « 201710 - 206.53 » |
| » | « 201601 - 59.62 » | « 201601 - 24.52 » | 535.84, 55.66, » | « 201711 - 96.63 » |
| « FATTURA,20160107,244.39 » | « 201601 - 42.5 » | « 201601 - 244 » | 57.44, 59.62, 635.4, » | « 201712 - 282.65 » |
| » | « 201601 - 159.43 » | « 201601 - 244.39 » | 65.03, 68.99, 7.32, » | « 201801 - 144.47 » |
| « FATTURA,20160107,36.16 » | « 201601 - 204.55 » | « 201601 - 247.21 » | 7.38, 7.88, 75.68, » | « 201802 - 152.43 » |
| » | « 201601 - 57.44 » | « 201601 - 247.84 » | 75.85, 80.15, 96.94, » | « 201803 - 190.62 » |
| « FATTURA,20160107,36.16 » | « 201601 - 132.2 » | « 201601 - 252.44 » | 97.17 » | « 201804 - 248.88 » |

Script Developed by Daniele Pirrone 2020 and free distributed

L'implementazione del codice con MapReduce è riportata in Appendice Job1.

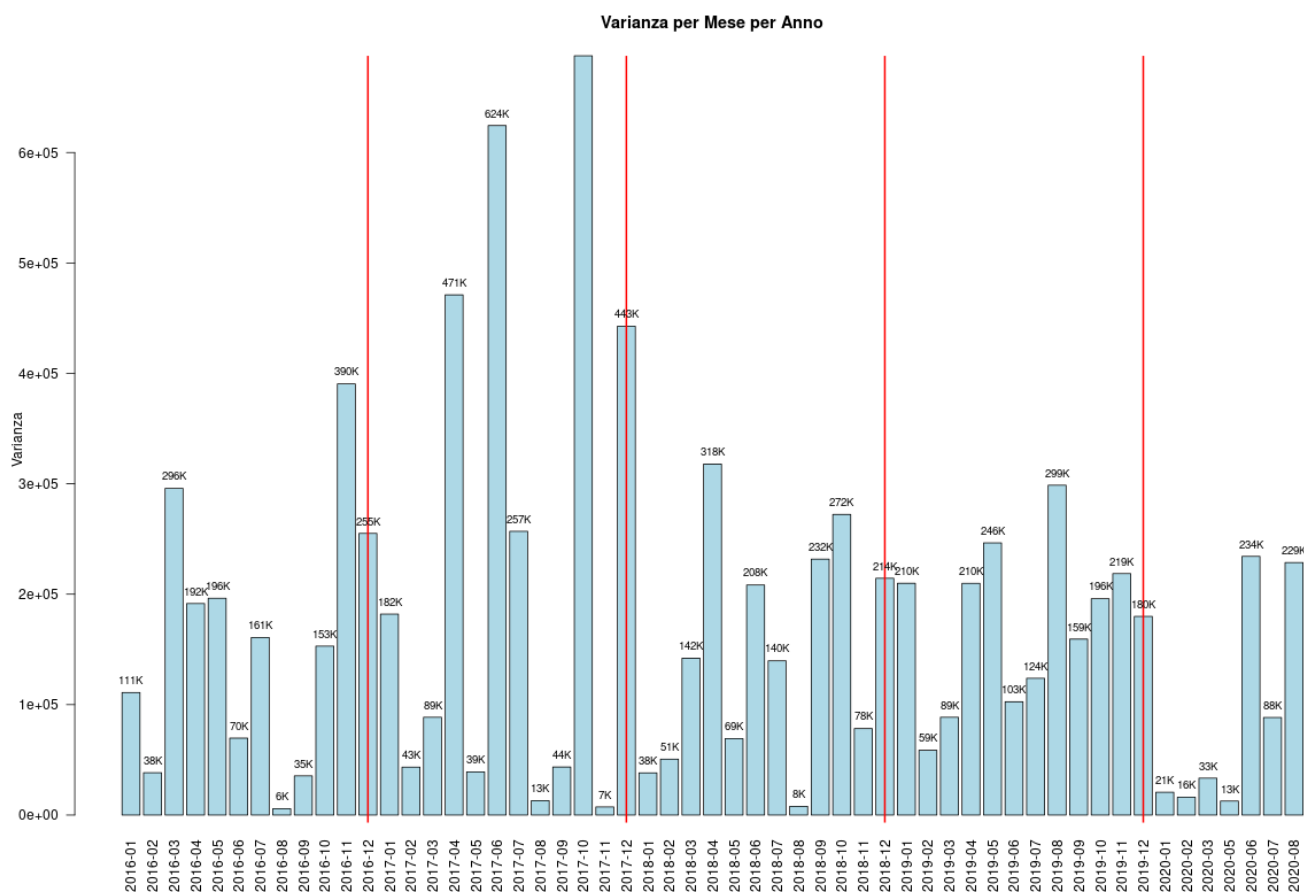
Job 2

Problema: Calcola la varianza di vendita per ogni mese di ogni anno.

Implementazione in R

1. Avvio libreria SQLite
2. Lettura file .csv da Directory
3. Aggiungo le intestazioni alle colonne nominandole Tipologia, Data e Prezzo
4. Avvio Query SQL
5. Raggruppa per Anno e Mese
6. Seleziona Varianza su Prezzo con Anno e Mese dal file dove nella colonna nominata Tipologia è presente il valore “FATTURA” o “RICEVUTA”
7. Ordina per Anno crescente e Mese crescente
8. Stampa Output

Il grafico risultante è il seguente:



L'implementazione del codice nel linguaggio R è riportata in Appendice Job2(ogni riga di codice è commentata per spiegarne il funzionamento), oltre al file Job2.csv che è il file prodotto da questo Job.

Implementazione in MapReduce

Una possibile soluzione in Map Reduce prevede queste fasi:

1. Leggo l'input
2. Si separa l'input per righe e filtro per tipo di documento "FATTURA" e "RICEVUTA"
3. Trasformo la data dal formato YYYYMMGG in YYYYMM
4. Imposto la data (YYYYMM) come K_Out_Map e l'importo come V_Out_Map ed eseguo la fase di mapping
5. La fase di shuffling ordina automaticamente l'insieme di coppie key-value
6. In fase di merging verranno uniti automaticamente tutti gli importi sotto la stessa key (YYYYMM)
7. Per ogni key si calcola la varianza (conoscendo la dimensione dell'array contenente gli importi per ogni key) ed effettuiamo il reducing
8. Visualizzo l'output

Una parte dell'output risultante è il seguente:

| Hadoop Map Reduce concept-simulation made by Pirrone Daniele | MAP | | REDUCE | |
|--|---------------------|---------------------|--|--|
| | INPUT-SPLITTING | SHUFFLING | MERGING | REDUCING |
| | « VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » |
| « FATTURA,20160104,139.8 » | « 201601 - 139.8 » | « 201601 - 106.6 » | « 201601 - 106.6, 108.02, 109.47, 11.13, 11.37, 113.02, 13.81, 132.2, 139.8, 145.85, 15.4, 159.43, 160.65, 160.92, 177.71, 179.19, 180.28, 19.15, 204.55, 22.4, 226.79, 235.42, 24.52, 244, 244.39, 247.21, 247.84, 252.44, 252.7, 2537.6, 26.94, 271.83, 294.92, 32.37, 332.28, 34.53, 343.02, 35.49, 36.16, 36.88, 360.17, 37.05, 42.5, 503.81, 51.37, 52.24, 53.94, 535.84, 55.66, 57.44, 59.62, 635.4, 65.03, 68.99, 7.32, 7.38, 7.88, 75.68, 75.85, 80.15, 96.94, 97.17 » | « 201601 - 110972.69 » « 201602 - 38351.04 » « 201603 - 295872.78 » « 201604 - 191570.68 » « 201605 - 196337.28 » « 201606 - 69653.87 » « 201607 - 160827.84 » « 201608 - 5731.74 » « 201609 - 35471.36 » « 201610 - 152986.94 » « 201611 - 390485.05 » « 201612 - 255096.32 » « 201701 - 181979.97 » « 201702 - 43393.86 » « 201703 - 88570.21 » « 201704 - 471101.40 » « 201705 - 39083.51 » « 201706 - 624458.35 » « 201707 - 256962.89 » « 201708 - 13003.69 » « 201709 - 43505.10 » « 201710 - » |
| « FATTURA,20160104,160.92 » | « 201601 - 160.92 » | « 201601 - 108.02 » | | |
| « FATTURA,20160104,65.03 » | « 201601 - 65.03 » | « 201601 - 109.47 » | | |
| « FATTURA,20160104,535.84 » | « 201601 - 535.84 » | « 201601 - 11.13 » | | |
| « FATTURA,20160104,75.68 » | « 201601 - 75.68 » | « 201601 - 11.37 » | | |
| « FATTURA,20160104,332.28 » | « 201601 - 332.28 » | « 201601 - 113.02 » | | |
| « FATTURA,20160105,343.02 » | « 201601 - 343.02 » | « 201601 - 13.81 » | | |
| « FATTURA,20160105,247.84 » | « 201601 - 247.84 » | « 201601 - 132.2 » | | |
| « FATTURA,20160105,75.85 » | « 201601 - 75.85 » | « 201601 - 139.8 » | | |
| « FATTURA,20160105,244.39 » | « 201601 - 244.39 » | « 201601 - 145.85 » | | |
| « FATTURA,20160105,36.16 » | « 201601 - 36.16 » | « 201601 - 15.4 » | | |
| « FATTURA,20160105,34.53 » | « 201601 - 34.53 » | « 201601 - 159.43 » | | |
| « FATTURA,20160105,51.37 » | « 201601 - 51.37 » | « 201601 - 160.65 » | | |
| « FATTURA,20160105,52.24 » | « 201601 - 52.24 » | « 201601 - 160.92 » | | |
| « FATTURA,20160105,113.02 » | « 201601 - 113.02 » | « 201601 - 177.71 » | | |
| « FATTURA,20160105,97.17 » | « 201601 - 97.17 » | « 201601 - 179.19 » | | |
| « FATTURA,20160105,252.7 » | « 201601 - 252.7 » | « 201601 - 180.28 » | | |
| « FATTURA,20160105,55.66 » | « 201601 - 55.66 » | « 201601 - 19.15 » | | |
| « FATTURA,20160105,106.6 » | « 201601 - 106.6 » | « 201601 - 204.55 » | | |
| « FATTURA,20160105,96.94 » | « 201601 - 96.94 » | « 201601 - 22.4 » | | |
| « FATTURA,20160105,109.47 » | « 201601 - 109.47 » | « 201601 - 226.79 » | | |
| « FATTURA,20160107,75.85 » | « 201601 - 24.52 » | « 201601 - 235.42 » | | |
| « FATTURA,20160107,244.39 » | « 201601 - 59.62 » | « 201601 - 24.52 » | | |
| « FATTURA,20160107,36.16 » | « 201601 - 42.5 » | « 201601 - 244 » | | |
| « FATTURA,20160107,34.53 » | « 201601 - 159.43 » | « 201601 - 244.39 » | | |
| « FATTURA,20160108,51.37 » | « 201601 - 204.55 » | « 201601 - 247.21 » | | |
| « FATTURA,20160108,52.24 » | « 201601 - 57.44 » | « 201601 - 247.84 » | | |
| « FATTURA,20160108,113.02 » | « 201601 - 132.2 » | « 201601 - 252.44 » | | |
| « FATTURA,20160108,97.17 » | « 201601 - 177.71 » | « 201601 - 252.7 » | | |
| « FATTURA,20160108,252.7 » | « 201601 - 235.42 » | « 201601 - 2537.6 » | | |
| « FATTURA,20160109,55.66 » | « 201601 - 19.15 » | « 201601 - 26.94 » | | |
| « FATTURA,20160109,7.38 » | « 201601 - 271.83 » | « 201601 - 271.83 » | | |
| « FATTURA,20160109,13.81 » | « 201601 - 252.44 » | « 201601 - 294.92 » | | |
| | « 201601 - 11.37 » | « 201601 - 32.37 » | | |
| | « 201601 - 226.79 » | « 201601 - 332.28 » | | |
| | « 201601 - 244 » | « 201601 - 34.53 » | | |
| | « 201601 - 32.37 » | « 201601 - 343.02 » | | |
| | « 201601 - 7.32 » | « 201601 - 35.49 » | | |
| | « 201601 - 160.65 » | « 201601 - 36.16 » | | |
| | « 201601 - 26.94 » | « 201601 - 36.88 » | | |
| | « 201601 - 635.4 » | « 201601 - 360.17 » | | |
| | « 201601 - 7.38 » | « 201601 - 37.05 » | | |
| | « 201601 - 13.81 » | « 201601 - 42.5 » | | |

Script Developed by Daniele Pirrone 2020 and free distributed

L'implementazione del codice con MapReduce è riportata in Appendice Job2.

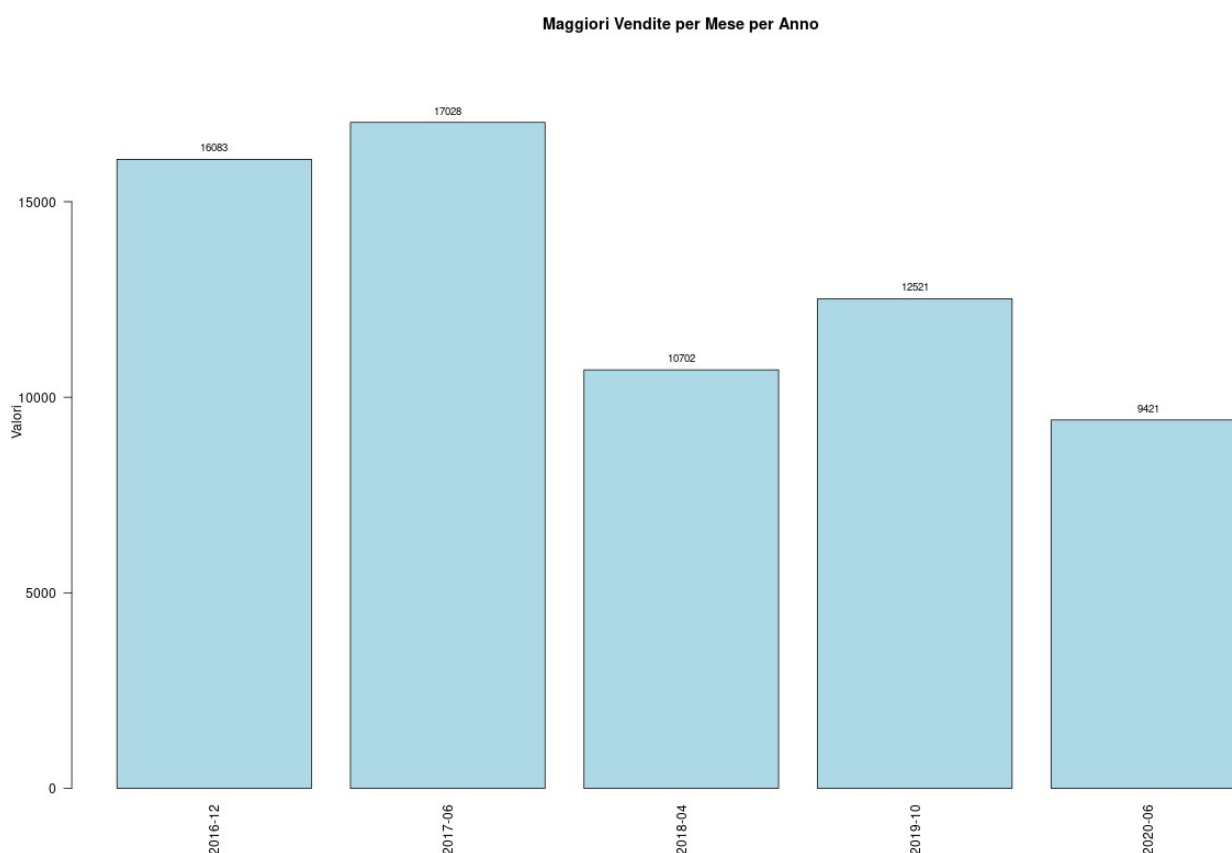
Job 3

Problema: Identifica il mese di ogni anno con maggiore vendita.

Implementazione in R

1. Avvio libreria SQLite
2. Lettura file .csv da Directory
3. Aggiungo le intestazioni alle colonne nominandole Tipologia, Data e Prezzo
4. Avvio Query SQL
5. Crea una tabella in cui viene calcolata la somma Prezzo raggruppando per Anno e Mese dove nella colonna nominata Tipologia è presente il valore “FATTURA” o “RICEVUTA”
6. Seleziona il Prezzo con Anno e Mese dalla tabella creata, sovrapposta (JOIN) alla stessa tabella dove sono selezionate le vendite massime raggruppate per Anno
7. Ordina per Anno crescente
8. Stampa Output

Il grafico risultante è il seguente:



L'implementazione del codice nel linguaggio R è riportata in Appendice Job3(ogni riga di codice è commentata per spiegarne il funzionamento), oltre al file Job3.csv che è il file prodotto da questo Job.

Implementazione in MapReduce

Una possibile soluzione in Map Reduce prevede queste fasi:

1. Leggo l'input
2. Si separa l'input per righe e filtro per tipo di documento "FATTURA" e "RICEVUTA"
3. Trasformo la data dal formato YYYYMMGG in YYYYMM
4. Imposto la data (YYYYMM) come K_Out_Map e l'importo come V_Out_Map ed eseguo la fase di mapping
5. La fase di shuffling ordina automaticamente l'insieme di coppie key-value
6. In fase di merging verranno uniti automaticamente tutti gli importi sotto la stessa key (YYYYMM)
7. Per ogni key si esegue la sommatoria degli importi ottenendo il fatturato totale mensile
8. Esegue una nuova MapReduce
 - a. Imposta l'anno come key e "mese-importo" come value ed esegue il mapping
 - b. Esegue esplicitamente fase di merging per unire sotto lo stesso anno i 12 valori "mese-importo"
 - c. Imposta la fase di reducing in cui per ogni anno ottiene il mese con importo maggiore
9. Visualizzo l'output:

| Hadoop Map Reduce concept-simulation made by Pirrone Daniele | | MAP | | REDUCE |
|--|---------------------|---------------------|--|-----------------|
| INPUT-SPLITTING | MAPPING | SHUFFLING | MERGING | REDUCING |
| « VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » |
| « FATTURA,20160104,139.8 » | « 201601 - 139.8 » | « 201601 - 106.6 » | « 201601 - 106.6, 108.02, 109.47, 11.13, 11.37, 113.02, 13.81, 132.2, 139.8, 145.85, 15.4, 159.43, 160.65, 160.92, 177.71, 179.19, 180.28, 19.15, 204.55, 22.4, 226.79, 235.42, 24.52, 244, 244.39, 247.21, 247.84, 252.44, 252.7, 2537.6, 26.94, 271.83, 294.92, 32.37, 332.28, 34.53, 343.02, 35.49, 36.16, 36.88, 360.17, 37.05, 42.5, 503.81, 51.37, 52.24, 53.94, 535.84, 55.66, 57.44, 59.62, 635.4, 65.03, 68.99, 7.32, 7.38, 7.88, 75.68, 75.85, 80.15, 96.94, 97.17 » | « 2016 - 12 » |
| « FATTURA,20160104,160.92 » | « 201601 - 160.92 » | « 201601 - 108.02 » | « 201601 - 108.02 » | « 2017 - 06 » |
| « FATTURA,20160104,65.03 » | « 201601 - 65.03 » | « 201601 - 109.47 » | « 201601 - 109.47 » | « 2018 - 04 » |
| « FATTURA,20160104,535.84 » | « 201601 - 535.84 » | « 201601 - 11.13 » | « 201601 - 11.13 » | « 2019 - 10 » |
| « FATTURA,20160104,75.68 » | « 201601 - 75.68 » | « 201601 - 11.37 » | « 201601 - 11.37 » | « 2020 - 06 » |
| « FATTURA,20160104,332.28 » | « 201601 - 332.28 » | « 201601 - 113.02 » | « 201601 - 113.02 » | |
| « FATTURA,20160105,343.02 » | « 201601 - 343.02 » | « 201601 - 13.81 » | « 201601 - 13.81 » | |
| « FATTURA,20160104,247.84 » | « 201601 - 247.84 » | « 201601 - 132.2 » | « 201601 - 132.2 » | |
| « FATTURA,20160107,75.85 » | « 201601 - 75.85 » | « 201601 - 139.8 » | « 201601 - 139.8 » | |
| « FATTURA,20160107,244.39 » | « 201601 - 244.39 » | « 201601 - 145.85 » | « 201601 - 145.85 » | |
| « FATTURA,20160107,36.16 » | « 201601 - 36.16 » | « 201601 - 15.4 » | « 201601 - 15.4 » | |
| « FATTURA,20160107,34.53 » | « 201601 - 34.53 » | « 201601 - 159.43 » | « 201601 - 159.43 » | |
| « FATTURA,20160108,51.37 » | « 201601 - 51.37 » | « 201601 - 160.65 » | « 201601 - 160.65 » | |
| « FATTURA,20160108,52.24 » | « 201601 - 52.24 » | « 201601 - 160.92 » | « 201601 - 160.92 » | |
| « FATTURA,20160108,113.02 » | « 201601 - 113.02 » | « 201601 - 177.71 » | « 201601 - 177.71 » | |
| « FATTURA,20160108,97.17 » | « 201601 - 97.17 » | « 201601 - 179.19 » | « 201601 - 179.19 » | |
| « FATTURA,20160108,252.7 » | « 201601 - 252.7 » | « 201601 - 180.28 » | « 201601 - 180.28 » | |
| « FATTURA,20160109,55.66 » | « 201601 - 55.66 » | « 201601 - 19.15 » | « 201601 - 19.15 » | |
| « FATTURA,20160111,106.6 » | « 201601 - 106.6 » | « 201601 - 204.55 » | « 201601 - 204.55 » | |
| « FATTURA,20160111,96.94 » | « 201601 - 96.94 » | « 201601 - 22.4 » | « 201601 - 22.4 » | |
| « FATTURA,20160111,109.47 » | « 201601 - 109.47 » | « 201601 - 226.79 » | « 201601 - 226.79 » | |
| « FATTURA,20160112,24.52 » | « 201601 - 24.52 » | « 201601 - 235.42 » | « 201601 - 235.42 » | |
| « FATTURA,20160112,59.62 » | « 201601 - 59.62 » | « 201601 - 24.52 » | « 201601 - 24.52 » | |
| « FATTURA,20160112,42.5 » | « 201601 - 42.5 » | « 201601 - 244 » | « 201601 - 244 » | |
| « FATTURA,20160112,159.43 » | « 201601 - 159.43 » | « 201601 - 244.39 » | « 201601 - 244.39 » | |
| « FATTURA,20160112,204.55 » | « 201601 - 204.55 » | « 201601 - 247.21 » | « 201601 - 247.21 » | |
| « FATTURA,20160113,57.44 » | « 201601 - 57.44 » | « 201601 - 247.84 » | « 201601 - 247.84 » | |
| « FATTURA,20160113,132.2 » | « 201601 - 132.2 » | « 201601 - 252.44 » | « 201601 - 252.44 » | |
| « FATTURA,20160113,177.71 » | « 201601 - 177.71 » | « 201601 - 252.7 » | « 201601 - 252.7 » | |
| « FATTURA,20160114,235.42 » | « 201601 - 235.42 » | « 201601 - 2537.6 » | « 201601 - 2537.6 » | |

L'implementazione del codice con MapReduce è riportata in Appendice Job3.

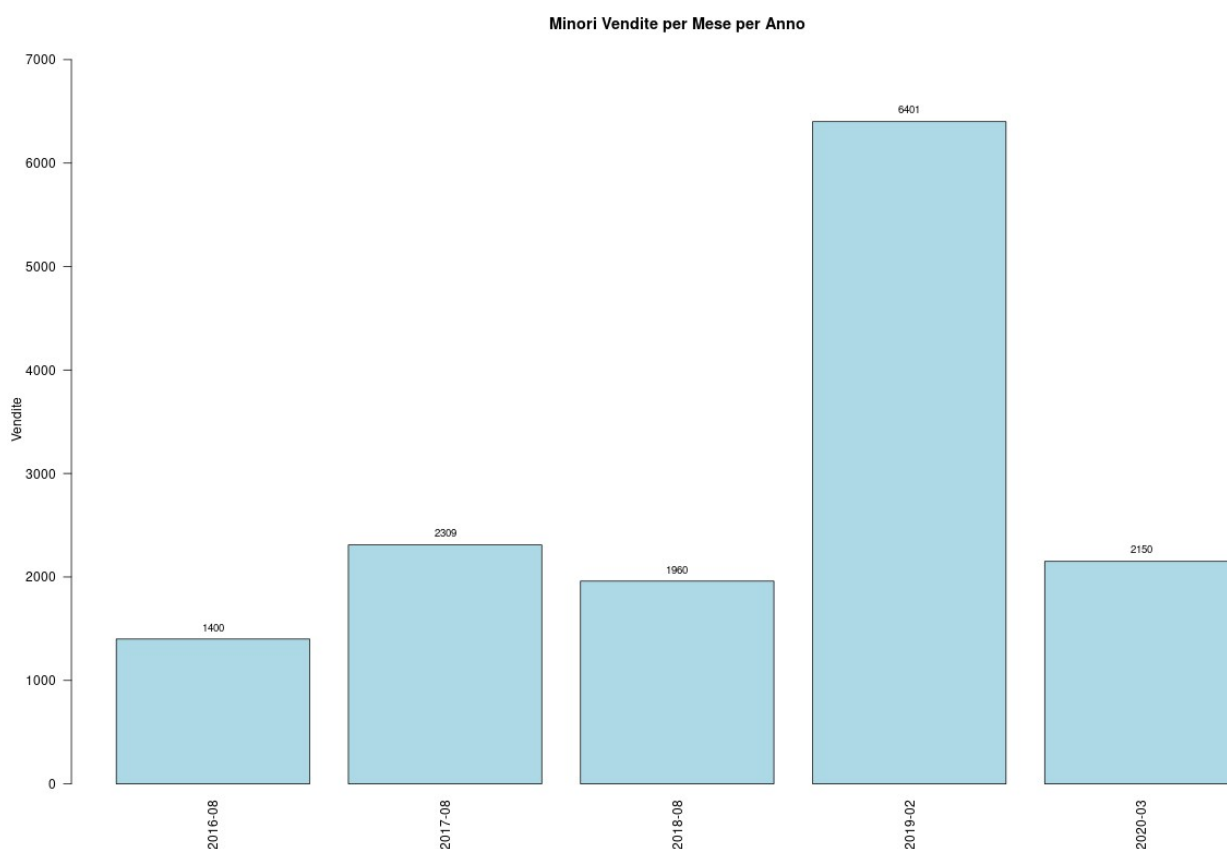
Job 4

Problema: Identifica il mese di ogni anno con minore vendita.

Implementazione in R

1. Avvio libreria SQLite
2. Lettura file .csv da Directory
3. Aggiungo le intestazioni alle colonne nominandole Tipologia, Data e Prezzo
4. Avvio Query SQL
5. Crea una tabella in cui viene calcolata la somma Prezzo raggruppando per Anno e Mese dove nella colonna nominata Tipologia è presente il valore “FATTURA” o “RICEVUTA”
6. Seleziona il Prezzo con Anno e Mese dalla tabella creata, sovrapposta (JOIN) alla stessa tabella dove sono selezionate le vendite minime raggruppate per Anno
7. Ordina per Anno crescente
8. Stampa Output

Il grafico risultante è il seguente:



L'implementazione del codice nel linguaggio R è riportata in Appendice Job4(ogni riga di codice è commentata per spiegarne il funzionamento), oltre al file Job4.csv che è il file prodotto da questo Job.

Implementazione in MapReduce

Una possibile soluzione in Map Reduce prevede queste fasi:

1. Leggo l'input
2. Si separa l'input per righe e filtro per tipo di documento "FATTURA" e "RICEVUTA"
3. Trasformo la data dal formato YYYYMMGG in YYYYMM
4. Imposto la data (YYYYMM) come K_Out_Map e l'importo come V_Out_Map ed eseguo la fase di mapping
5. La fase di shuffling ordina automaticamente l'insieme di coppie key-value
6. In fase di merging verranno uniti automaticamente tutti gli importi sotto la stessa key (YYYYMM)
7. Per ogni key si esegue la sommatoria degli importi ottenendo il fatturato totale mensile
8. Esegue una nuova MapReduce
 - a. Imposta l'anno come key e "mese-importo" come value ed esegue il mapping
 - b. Esegue esplicitamente fase di merging per unire sotto lo stesso anno i 12 valori "mese-importo"
 - c. Imposta la fase di reducing in cui per ogni anno ottiene il mese con importo minore
9. Visualizzo l'output:

| Hadoop Map Reduce concept-simulation made by Pirrone Daniele | | MAP | | | REDUCE | |
|--|--|---------------------|--|---------------------|--|---|
| INPUT-SPLITTING | | MAPPING | | SHUFFLING | MERGING | REDUCING |
| « VALUE » | | « KEY - VALUE » | | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » |
| « FATTURA,20160104,139.8 » | | « 201601 - 139.8 » | | « 201601 - 106.6 » | « 201601 - 106.6, 108.02, 109.47, 11.13, 11.37, 113.02, 13.81, 132.2, 139.8, 145.85, 15.4, 159.43, 160.65, 160.92, 177.71, 179.19, 180.28, 19.15, 204.55, 22.4, 226.79, 235.42, 24.52, 244, 244.39, 247.21, 247.84, 252.44, 252.7, 2537.6, 26.94, 271.83, 294.92, 32.37, 332.28, 34.53, 343.02, 35.49, 36.16, 36.88, 360.17, 37.05, 42.5, 503.81, 51.37, 52.24, 53.94, 535.84, 55.66, 57.44, 59.62, 635.4, 65.03, 68.99, 7.32, 7.38, 7.88, 75.68, 75.85, 80.15, 96.94, 97.17 » | « 2016 - 08 » « 2017 - 08 » « 2018 - 08 » « 2019 - 12 » « 2020 - 03 » |
| « FATTURA,20160104,160.92 » | | « 201601 - 160.92 » | | « 201601 - 108.02 » | | |
| « FATTURA,20160104,65.03 » | | « 201601 - 65.03 » | | « 201601 - 109.47 » | | |
| « FATTURA,20160104,535.84 » | | « 201601 - 535.84 » | | « 201601 - 11.13 » | | |
| « FATTURA,20160104,75.68 » | | « 201601 - 75.68 » | | « 201601 - 11.37 » | | |
| « FATTURA,20160104,332.28 » | | « 201601 - 332.28 » | | « 201601 - 113.02 » | | |
| « FATTURA,20160105,343.02 » | | « 201601 - 343.02 » | | « 201601 - 13.81 » | | |
| « FATTURA,20160104,247.84 » | | « 201601 - 247.84 » | | « 201601 - 132.2 » | | |
| « FATTURA,20160107,75.85 » | | « 201601 - 75.85 » | | « 201601 - 139.8 » | | |
| « FATTURA,20160107,244.39 » | | « 201601 - 244.39 » | | « 201601 - 145.85 » | | |
| « FATTURA,20160107,36.16 » | | « 201601 - 36.16 » | | « 201601 - 15.4 » | | |
| « FATTURA,20160107,34.53 » | | « 201601 - 34.53 » | | « 201601 - 159.43 » | | |
| « FATTURA,20160108,51.37 » | | « 201601 - 51.37 » | | « 201601 - 160.65 » | | |
| « FATTURA,20160108,52.24 » | | « 201601 - 52.24 » | | « 201601 - 160.92 » | | |
| « FATTURA,20160108,113.02 » | | « 201601 - 113.02 » | | « 201601 - 177.71 » | | |
| « FATTURA,20160108,97.17 » | | « 201601 - 97.17 » | | « 201601 - 179.19 » | | |
| « FATTURA,20160108,252.7 » | | « 201601 - 252.7 » | | « 201601 - 180.28 » | | |
| « FATTURA,20160109,55.66 » | | « 201601 - 55.66 » | | « 201601 - 19.15 » | | |
| « FATTURA,20160111,106.6 » | | « 201601 - 106.6 » | | « 201601 - 204.55 » | | |
| « FATTURA,20160111,96.94 » | | « 201601 - 96.94 » | | « 201601 - 22.4 » | | |
| « FATTURA,20160111,109.47 » | | « 201601 - 109.47 » | | « 201601 - 226.79 » | | |
| « FATTURA,20160112,24.52 » | | « 201601 - 24.52 » | | « 201601 - 235.42 » | | |
| « FATTURA,20160112,59.62 » | | « 201601 - 59.62 » | | « 201601 - 24.52 » | | |
| « FATTURA,20160112,42.5 » | | « 201601 - 42.5 » | | « 201601 - 244 » | | |
| « FATTURA,20160112,159.43 » | | « 201601 - 159.43 » | | « 201601 - 244.39 » | | |
| « FATTURA,20160112,204.55 » | | « 201601 - 204.55 » | | « 201601 - 247.21 » | | |
| « FATTURA,20160113,57.44 » | | « 201601 - 57.44 » | | « 201601 - 247.84 » | | |
| « FATTURA,20160113,132.2 » | | « 201601 - 132.2 » | | « 201601 - 252.44 » | | |
| « FATTURA,20160113,177.71 » | | « 201601 - 177.71 » | | « 201601 - 252.7 » | | |
| « FATTURA,20160114,235.42 » | | « 201601 - 235.42 » | | « 201601 - 2537.6 » | | |
| « FATTURA,20160114,235.42 » | | « 201601 - 235.42 » | | « 201601 - 2537.6 » | | |

L'implementazione del codice con MapReduce è riportata in Appendice Job4.

Conclusioni

Risultati Ottenuti

L'obiettivo principale è stato scrivere il codice per i job richiesti, si è dunque calcolato:

- la media di vendita per ogni mese di ogni anno;
- la varianza di vendita per ogni mese di ogni anno;
- il mese di ogni anno con maggiore vendita;
- il mese di ogni anno con minore vendita.

Occorre specificare che i dati a disposizione per l'anno 2020 arrivano fino ad Agosto e di conseguenza incompleti.

Da questi Job si sono potute fare delle statistiche rispetto agli anni passati, vedendo come sia le vendite che il fatturato siano potuti variare in base agli anni di attività della Ditta, grazie alla visualizzazione dei grafici elaborati.

R e Map Reduce a confronto

Confrontare R con Hadoop MapReduce riguarda non solo l'analisi delle prestazioni ma anche le differenze fondamentali nell'applicabilità. Entrambi gli strumenti hanno i propri vantaggi e svantaggi, a seconda del contesto in cui vengono utilizzati.

Applicabilità:

- **R:** È un linguaggio di programmazione per l'analisi statistica e la visualizzazione dei dati. R è ideale per l'analisi di dataset di dimensioni moderate e per prototipazione rapida.
- **MapReduce di Hadoop:** È un framework per il calcolo distribuito e lo storage di dati su un cluster di computer. MapReduce è progettato per elaborare enormi quantità di dati in modo parallelo.

Prestazioni:

- **R:** Tende a essere più lento su grandi dataset perché è progettato per essere eseguito su una singola macchina e ha una capacità di memoria limitata.
- **MapReduce di Hadoop:** È in grado di elaborare petabyte di dati in modo efficiente utilizzando un cluster di computer. Il parallelismo e la distribuzione riducono significativamente il tempo di elaborazione.

Data la dimensione estremamente ridotta (150,9 kB) del file da Analizzare Ordini.csv, il linguaggio R risulta molto più efficiente e funzionale rispetto a MapReduce di Hadoop.

Utilizzando il Tool del prof. Pirrone infatti si nota subito la differenza di velocità di elaborazione, che nel mio caso con le mie specifiche hardware e software risulta essere di pochi secondi, rispetto alla velocità del linguaggio R che è quasi impercettibile.

Diverso sarebbe stato se il file da analizzare Ordini.csv avesse avuto dimensioni dell'ordine di Terabyte (TB) Petabyte (PB). In quel caso R non sarebbe riuscito a produrre i risultati con le giuste performance perché adoperando in locale non avrebbe avuto abbastanza potenza di calcolo e memoria.

MapReduce invece, con un'adeguata configurazione in ambiente Hadoop su un sistema distribuito multinodo, avrebbe avuto dei tempi di calcolo estremamente migliori rispetto alla soluzione sequenziale.

Analisi Dati su Previsioni Future

Dalle analisi dei Job effettuati si può provare ad avere una previsione futura dei dati.

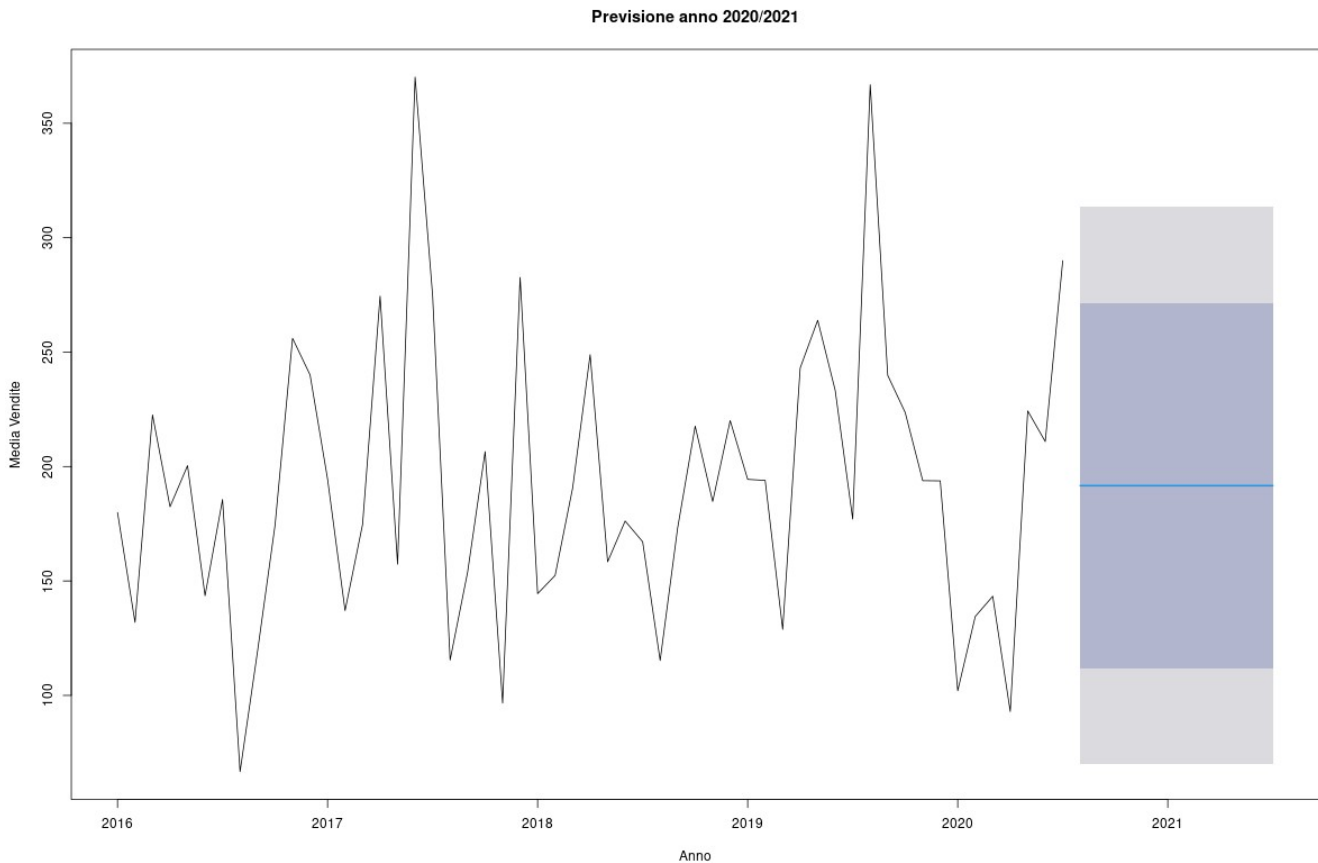
Una prima ipotesi di previsione, è andare a calcolare la media delle vendite relative all'anno 2020/2021 per ogni mese.

Per fare una previsione della media di vendita annuale, è possibile utilizzare un modello di serie storica come ARIMA in R utilizzando come base il file estrapolato dal Job1 (job1.csv), così da avere già la media storica di vendite ordinata dal 01/2016 al 08/2020.

Il modello ARIMA (AutoRegressive Integrated Moving Average) è un modello statistico popolare utilizzato per analizzare e prevedere dati di serie temporali. Il modello ARIMA è in realtà una combinazione di modelli, che include componenti auto-regressivi (AR), componenti di media mobile (MA) e un termine di integrazione (I) che rappresenta il numero di differenze non stagionali necessarie per rendere la serie stazionaria.

Per ottenere il grafico si è creata una serie temporale con frequenza mensile (12 mesi per anno) e utilizzato un modello ARIMA per effettuare le previsioni per i prossimi 12 mesi.

Grazie poi alla libreria Forecast in linguaggio R è stato possibile ottenere il grafico con la previsione dell'andamento rispetto ai mesi successivi fino ad un anno di distanza dagli ultimi dati rilevati.



Il grafico risultante da una previsione ARIMA utilizzando la libreria Forecast in R è un grafico a linee, noto come "line chart" in inglese. In questo tipo di grafico, i dati storici sono rappresentati da linee continue, mentre la previsione futura è rappresentata da una seconda linea continua che rappresenta il valore medio. In aggiunta, è incluso un intervallo di confidenza intorno alla previsione, che è mostrato come un'area blu che tende al bianco man mano che ci si allontana dalla media di previsione.

Il grafico a linee è particolarmente efficace in questo contesto perché permette una facile visualizzazione delle tendenze nel tempo e offre un modo intuitivo di confrontare i dati storici con le previsioni future.

L'implementazione del codice in linguaggio R è riportata in Appendice Previsioni.r.

Riferimento bibliografici, Sitografia

MAP REDUCE TOOL -

<https://www.uninettunouniversity.net/it/cyberspaziomateriale didattico.aspx?faculty=°ree=207&planid=605&courseid=4386&lf=it&lezid=101901&matdidid=208073&type=3&tutorid=152312>

Classe interattiva n. 20.1 -

<https://www.uninettunouniversity.net/it/cyberspaziomateriale didattico.aspx?&faculty=°ree=207&planid=605&courseid=4386&lf=it&lezid=101901&matdidid=207982&type=11&matdididjoinid=207981&tutorid=152312>

<https://www.fattura24.com/dizionario>

Appendice

Job1

Job1.r

```
#Installa il package per utilizzare SQL
install.packages("sqldf")

#Avvia la libreria SQLite
library(sqldf)

#Legge file da directory
file_contabilità <- read.csv("/home/manuel/Scrivania/BigData/HW/Ordini.csv", header=FALSE)

#Rinomina le colonne del file Ordini.csv
names(file_contabilità) <- c("Tipologia", "Data", "Prezzo")

#Avvia Query SQL
risultato <- sqldf("
SELECT
AVG(Prezzo) AS media_valore,
SUBSTR(Data, 1, 4) AS Anno,
SUBSTR(Data, 5, 2) AS Mese,
COUNT(Data) AS Numero_di_Vendite
FROM
file_contabilità
WHERE
Tipologia = 'FATTURA' OR Tipologia = 'RICEVUTA'
GROUP BY
Anno, Mese
ORDER BY
Anno ASC, Mese ASC
")

#Stampa Risultato
print(risultato)

# Combina Anno e Mese per ottenere una rappresentazione nel formato "YYYY-MM"
risultato$Periodo <- paste(risultato$Anno, risultato$Mese, sep="-")

# Converta la colonna Anno in numerico
risultato$Anno <- as.numeric(risultato$Anno)

# Salva i risultati nel file job1.csv
write.csv(risultato, file = "/home/manuel/Scrivania/BigData/HW/HW Mio/job1.csv", row.names
= FALSE)

# Salva il grafico
png(filename="/home/manuel/Scrivania/BigData/HW/HW Mio/job1grafico.png",width=1200,
height=800)
```

```

# Crea un bar plot
grafico = barplot(risultato$media_valore,
names.arg=risultato$Periodo,
main="Distribuzione Media Valori per Mese per Anno",
xlab="",
ylab="Media Valore",
col="lightblue",
border="black",
las=2)

# Determina dove inserire i divisori tra gli anni
divisori = which(diff(risultato$Anno) == 1)

# Aggiungi i divisori al grafico
abline(v=grafico[divisori+0.5], col="red", lwd=2)

# Aggiungi valori al grafico
text(grafico, risultato$media_valore + 0.5, labels=round(risultato$media_valore, 0), pos=3,
cex=0.8)

# Chiudi il dispositivo di output, salvando l'immagine
dev.off()

```

Job1.js

```

function jobInputSplit(input_str){
  return input_str.split('\n').filter(function(ordine){
    return (ordine.split(',')[0] === 'FATTURA' || ordine.split(',')[0] === 'RICEVUTA');
  });
}

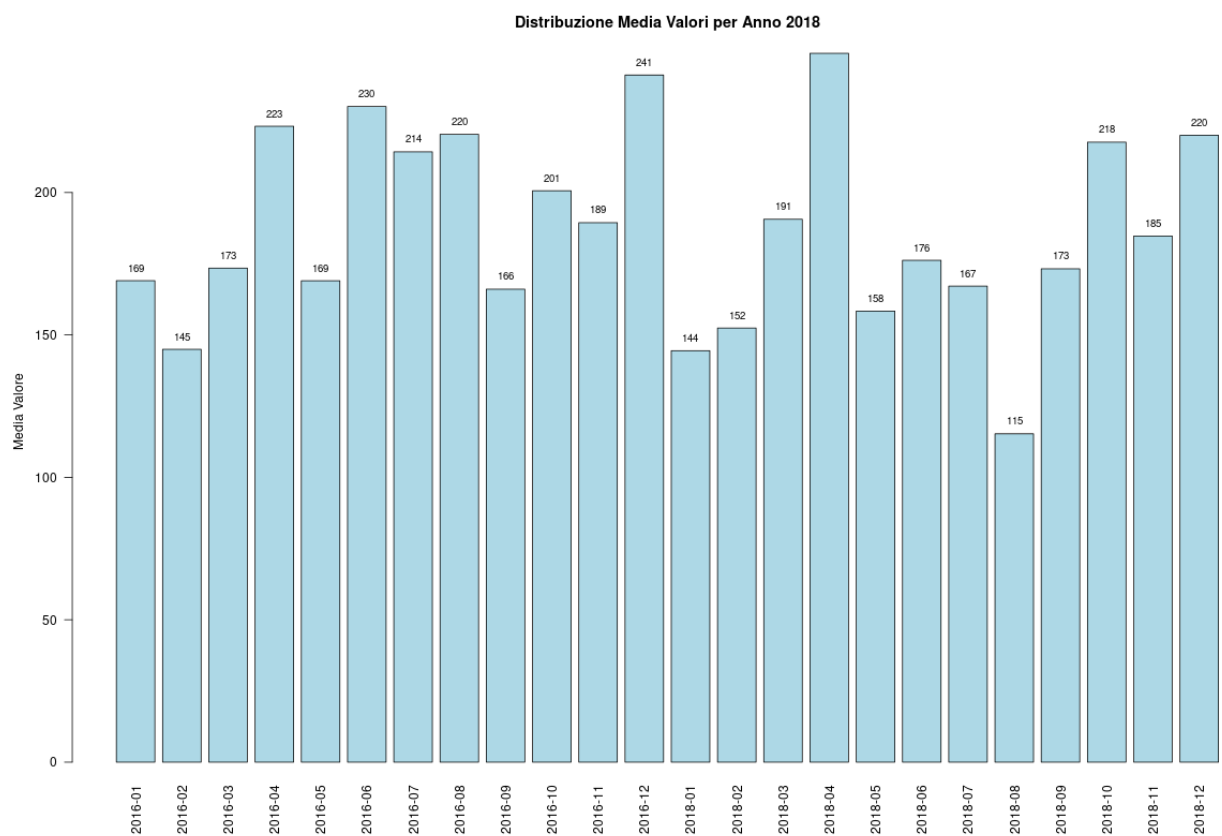
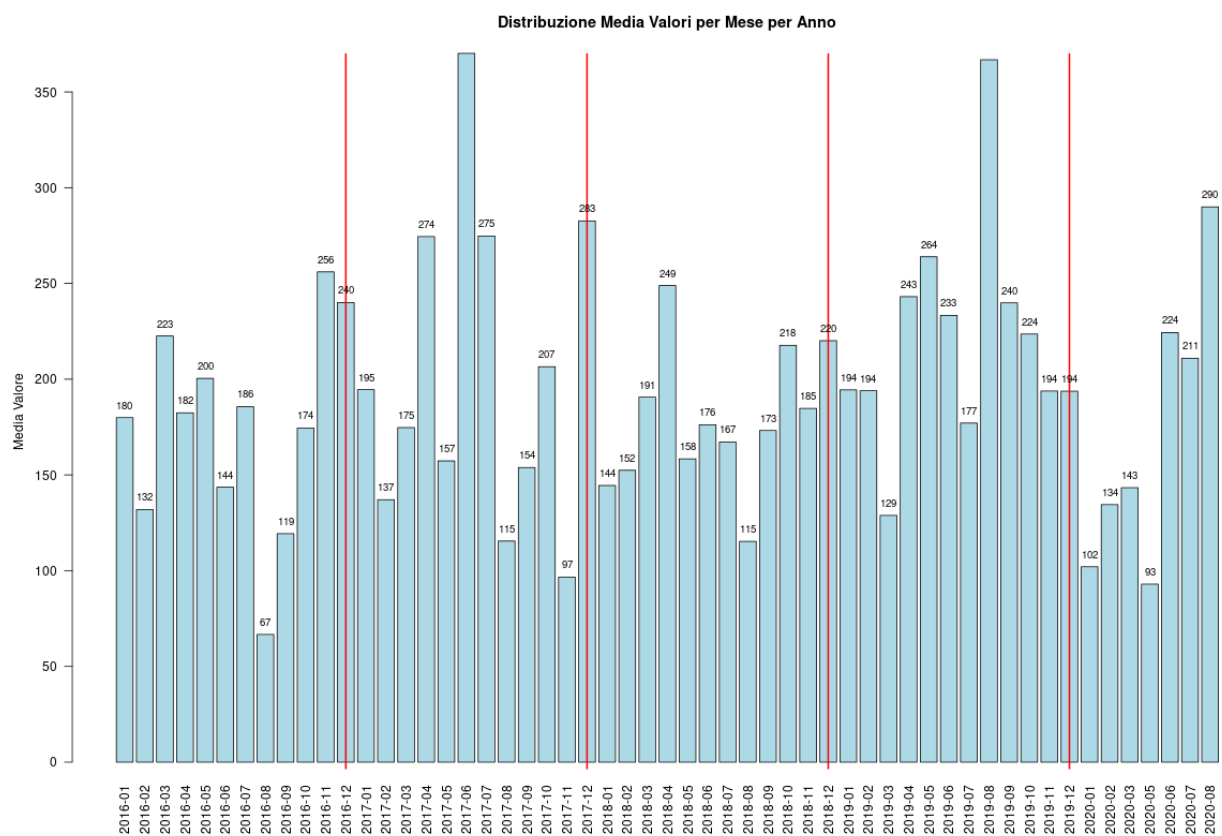
function jobMap(V_In_Map){
  return V_In_Map.map(function(item){
    var vendita = item.split(',');
    return keyVal(vendita[1].substring(0, 6), vendita[2]);
  });
}

function jobReduce(K_In_Reduce_V_In_Reduce){
  return K_In_Reduce_V_In_Reduce.map(function (items){
    var K_In_Reduce = items.split(S[0])[0];
    var V_In_Reduce = items.split(S[0])[1].split(S[1]);
    var Reduce = V_In_Reduce.reduce(function (accumulator, item)
    {
      return parseFloat(accumulator) + parseFloat(item);
    });
    return keyVal(K_In_Reduce,
      parseFloat(Reduce/V_In_Reduce.length).toFixed(2));
  });
}

```


Job1.csv

```
"media_valore","Anno","Mese","Numero_di_Vendite","Periodo"
179.930806451613,2016,"01",62,"2016-01"
131.922878787879,2016,"02",66,"2016-02"
222.558,2016,"03",55,"2016-03"
182.441,2016,"04",60,"2016-04"
200.412698412698,2016,"05",63,"2016-05"
143.58,2016,"06",73,"2016-06"
185.629508196721,2016,"07",61,"2016-07"
66.6685714285714,2016,"08",21,"2016-08"
119.334838709677,2016,"09",62,"2016-09"
174.441967213115,2016,"10",61,"2016-10"
256.046949152542,2016,"11",59,"2016-11"
240.048955223881,2016,"12",67,"2016-12"
194.520740740741,2017,"01",54,"2017-01"
137.063272727273,2017,"02",55,"2017-02"
174.725172413793,2017,"03",58,"2017-03"
274.4684375,2017,"04",32,"2017-04"
157.31725,2017,"05",40,"2017-05"
370.175869565217,2017,"06",46,"2017-06"
274.784666666667,2017,"07",60,"2017-07"
115.4655,2017,"08",20,"2017-08"
153.856271186441,2017,"09",59,"2017-09"
206.527547169811,2017,"10",53,"2017-10"
96.6337777777778,2017,"11",45,"2017-11"
282.651707317073,2017,"12",41,"2017-12"
144.473137254902,2018,"01",51,"2018-01"
152.434615384615,2018,"02",39,"2018-02"
190.615217391304,2018,"03",46,"2018-03"
248.883488372093,2018,"04",43,"2018-04"
158.388775510204,2018,"05",49,"2018-05"
176.204893617021,2018,"06",47,"2018-06"
167.145208333333,2018,"07",48,"2018-07"
115.287058823529,2018,"08",17,"2018-08"
173.219761904762,2018,"09",42,"2018-09"
217.671428571429,2018,"10",35,"2018-10"
184.736538461538,2018,"11",52,"2018-11"
220.090909090909,2018,"12",33,"2018-12"
194.409130434783,2019,"01",46,"2019-01"
193.96,2019,"02",33,"2019-02"
128.864423076923,2019,"03",52,"2019-03"
243.089268292683,2019,"04",41,"2019-04"
263.925121951219,2019,"05",41,"2019-05"
233.303255813953,2019,"06",43,"2019-06"
177.07170212766,2019,"07",47,"2019-07"
366.8272,2019,"08",25,"2019-08"
239.870612244898,2019,"09",49,"2019-09"
223.587857142857,2019,"10",56,"2019-10"
193.836181818182,2019,"11",55,"2019-11"
193.747647058824,2019,"12",34,"2019-12"
102.029791666667,2020,"01",48,"2020-01"
134.476774193548,2020,"02",31,"2020-02"
143.335333333333,2020,"03",15,"2020-03"
92.9560563380282,2020,"05",71,"2020-05"
224.305714285714,2020,"06",42,"2020-06"
210.935641025641,2020,"07",39,"2020-07"
289.979583333333,2020,"08",24,"2020-08"
```



Upload Text File or drop below

OFFERTA, 20200810, 72.69
FATTURA, 20200810, 112.51
FATTURA, 20200810, 223.55
BUONO. PRELIEVO, 20200810, 58.56

Script loaded (show/hide)

[upload (1)]

Run MapReduce or CTRL+ENTER

Reload Page

| | | | | |
|--|---------------------|---------------------|---|---------------------|
| Hadoop Map Reduce concept-simulation made by Pirrone Daniele | MAP | | REDUCE | |
| | INPUT-SPLITTING | MAPPING | SHUFFLING | MERGING |
| | « VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » |
| « FATTURA,20160104,139.8 » | « 201601 - 139.8 » | « 201601 - 106.6 » | « 201601 - 106.6, 108.02, 109.47, 11.13, 11.37, » | « 201601 - 179.93 » |
| « FATTURA,20160104,160.92 » | « 201601 - 160.92 » | « 201601 - 108.02 » | « 201601 - 109.47 » | « 201602 - 131.92 » |
| « FATTURA,20160104,65.03 » | « 201601 - 65.03 » | « 201601 - 109.47 » | « 201601 - 11.13 » | « 201603 - 222.56 » |
| « FATTURA,20160104,160.92 » | « 201601 - 535.84 » | « 201601 - 11.13 » | « 201601 - 113.02, 13.81, » | « 201604 - 182.44 » |
| « FATTURA,20160104,65.03 » | « 201601 - 75.68 » | « 201601 - 11.37 » | « 201601 - 132.2, 139.8, » | « 201605 - 200.41 » |
| « FATTURA,20160104,65.03 » | « 201601 - 332.28 » | « 201601 - 113.02 » | « 201601 - 145.85, 15.4, » | « 201606 - 143.58 » |
| « FATTURA,20160104,535.84 » | « 201601 - 343.02 » | « 201601 - 13.81 » | « 201601 - 159.43, 160.65, » | « 201607 - 185.63 » |
| « FATTURA,20160104,535.84 » | « 201601 - 247.84 » | « 201601 - 132.2 » | « 201601 - 160.92, 177.71, » | « 201608 - 66.67 » |
| « FATTURA,20160104,535.84 » | « 201601 - 75.85 » | « 201601 - 139.8 » | « 201601 - 179.19, 180.28, » | « 201609 - 119.33 » |
| « FATTURA,20160104,75.68 » | « 201601 - 244.39 » | « 201601 - 145.85 » | « 201601 - 19.15, 204.55, 22.4, » | « 201610 - 174.44 » |
| « FATTURA,20160104,75.68 » | « 201601 - 36.16 » | « 201601 - 15.4 » | « 201601 - 226.79, 235.42, » | « 201611 - 256.15 » |
| « FATTURA,20160104,332.28 » | « 201601 - 34.53 » | « 201601 - 159.43 » | « 201601 - 24.52, 244, 244.39, » | « 201612 - 250.21 » |
| « FATTURA,20160104,332.28 » | « 201601 - 51.37 » | « 201601 - 160.65 » | « 201601 - 247.21, 247.84, » | « 201701 - 194.52 » |
| « FATTURA,20160105,343.02 » | « 201601 - 52.24 » | « 201601 - 160.92 » | « 201601 - 252.44, 252.7, » | « 201702 - 137.06 » |
| « FATTURA,20160105,343.02 » | « 201601 - 113.02 » | « 201601 - 177.71 » | « 201601 - 2537.6, 26.94, » | « 201703 - 174.73 » |
| « FATTURA,20160105,343.02 » | « 201601 - 97.17 » | « 201601 - 179.19 » | « 201601 - 271.83, 294.92, » | « 201704 - 274.47 » |
| « FATTURA,20160105,343.02 » | « 201601 - 252.7 » | « 201601 - 180.28 » | « 201601 - 32.37, 332.28, » | « 201705 - 157.32 » |
| « FATTURA,20160105,343.02 » | « 201601 - 55.66 » | « 201601 - 19.15 » | « 201601 - 34.53, 343.02, » | « 201706 - 370.18 » |
| « FATTURA,20160105,343.02 » | « 201601 - 106.6 » | « 201601 - 204.55 » | « 201601 - 35.49, 36.16, 36.88, » | « 201707 - 274.78 » |
| « FATTURA,20160105,343.02 » | « 201601 - 96.94 » | « 201601 - 22.4 » | « 201601 - 360.17, 37.05, 42.5, » | « 201708 - 120.59 » |
| « FATTURA,20160105,343.02 » | « 201601 - 109.47 » | « 201601 - 226.79 » | « 201601 - 503.81, 51.37, » | « 201709 - 157.89 » |
| « FATTURA,20160107,75.85 » | « 201601 - 24.52 » | « 201601 - 235.42 » | « 201601 - 52.24, 53.94, » | « 201710 - 206.53 » |
| « FATTURA,20160107,75.85 » | « 201601 - 59.62 » | « 201601 - 24.52 » | « 201601 - 535.84, 55.66, » | « 201711 - 96.63 » |
| « FATTURA,20160107,75.85 » | « 201601 - 42.5 » | « 201601 - 244 » | « 201601 - 57.44, 59.62, 635.4, » | « 201712 - 282.65 » |
| « FATTURA,20160107,244.39 » | « 201601 - 159.43 » | « 201601 - 244.39 » | « 201601 - 65.03, 68.99, 7.32, » | « 201801 - 144.47 » |
| « FATTURA,20160107,244.39 » | « 201601 - 204.55 » | « 201601 - 247.21 » | « 201601 - 7.38, 7.88, 75.68, » | « 201802 - 152.43 » |
| « FATTURA,20160107,36.16 » | « 201601 - 57.44 » | « 201601 - 247.84 » | « 201601 - 75.85, 80.15, 96.94, » | « 201803 - 190.62 » |
| « FATTURA,20160107,36.16 » | « 201601 - 132.2 » | « 201601 - 252.44 » | « 201601 - 97.17 » | « 201804 - 248.88 » |

Job2

Job2.r

```
#Installa il package per utilizzare SQL
install.packages("sqldf")

#Avvia la libreria SQLite
library(sqldf)

#Legge file da directory
file_contabilità <- read.csv("/home/manuel/Scrivania/BigData/HW/Ordini.csv", header=FALSE)

#Rinomina le colonne del file Ordini.csv
names(file_contabilità) <- c("Tipologia", "Data", "Prezzo")

#Avvia Query SQL
risultato <- sqldf("
SELECT
VARIANCE(Prezzo) AS Varianza,
SUBSTR(Data, 1, 4) AS Anno,
SUBSTR(Data, 5, 2) AS Mese,
COUNT(Data) AS Numero_di_Vendite
FROM
file_contabilità
WHERE
Tipologia = 'FATTURA' OR Tipologia = 'RICEVUTA'
GROUP BY
Anno='2018', Mese
ORDER BY
Anno ASC, Mese ASC
")

#Stampa Risultato
print(risultato)

# Combina Anno e Mese per ottenere una rappresentazione nel formato "YYYY-MM"
risultato$Periodo <- paste(risultato$Anno, risultato$Mese, sep="-")

# Converte la colonna Anno in numerico
risultato$Anno <- as.numeric(risultato$Anno)

# Salva i risultati nel file job1.csv
write.csv(risultato, file = "/home/manuel/Scrivania/BigData/HW/HW Mio/job2(2018).csv",
row.names = FALSE)

# Salva il grafico
```

```
png(filename="/home/manuel/Scrivania/BigData/HW/HW
Mio/job2grafico(2018).png",width=1200, height=800)
```

```
# Crea un bar plot
```

```
grafico = barplot(risultato$Varianza,
names.arg=risultato$Periodo,
main="Varianza per Anno 2018",
xlab="",
ylab="Varianza",
col="lightblue",
border="black",
las=2)
```

```
# Determina dove inserire i divisori tra gli anni
```

```
divisori = which(diff(risultato$Anno) == 1)
```

```
# Aggiungi i divisori al grafico
```

```
abline(v=grafico[divisori+0.5], col="red", lwd=2)
```

```
# Aggiungi valori al grafico
```

```
text(grafico, risultato$Varianza + 0.5, labels=round(risultato$Varianza, 0), pos=3, cex=0.8)
```

```
# Chiudi il dispositivo di output, salvando l'immagine
```

```
dev.off()
```

Job2.js

```
function jobInputSplit(input_str){
```

```
  return input_str.split("\n").filter(function(ordine){
```

```
    return (ordine.split(',')[0] === 'FATTURA' || ordine.split(',')[0] === 'RICEVUTA');
```

```
  });
```

```
}
```

```
function jobMap(V_In_Map){
```

```
  return V_In_Map.map(function(item){
```

```
    var vendita = item.split(',');
```

```
    return keyVal(vendita[1].substring(0, 6), vendita[2]);
```

```
  });
```

```
}
```

```
function jobReduce(K_In_Reduce_V_In_Reduce){
```

```
  return K_In_Reduce_V_In_Reduce.map(function (items){
```

```
    var K_In_Reduce = items.split(S[0])[0];
```

```
    var V_In_Reduce = items.split(S[0])[1].split(S[1]);
```

```
    var num = V_In_Reduce.length;
```

```
    //media
```

```
    var somma = V_In_Reduce.reduce(function (accumulator, item) {
```

```
      return parseFloat(accumulator) + parseFloat(item);
```

```
    });
```

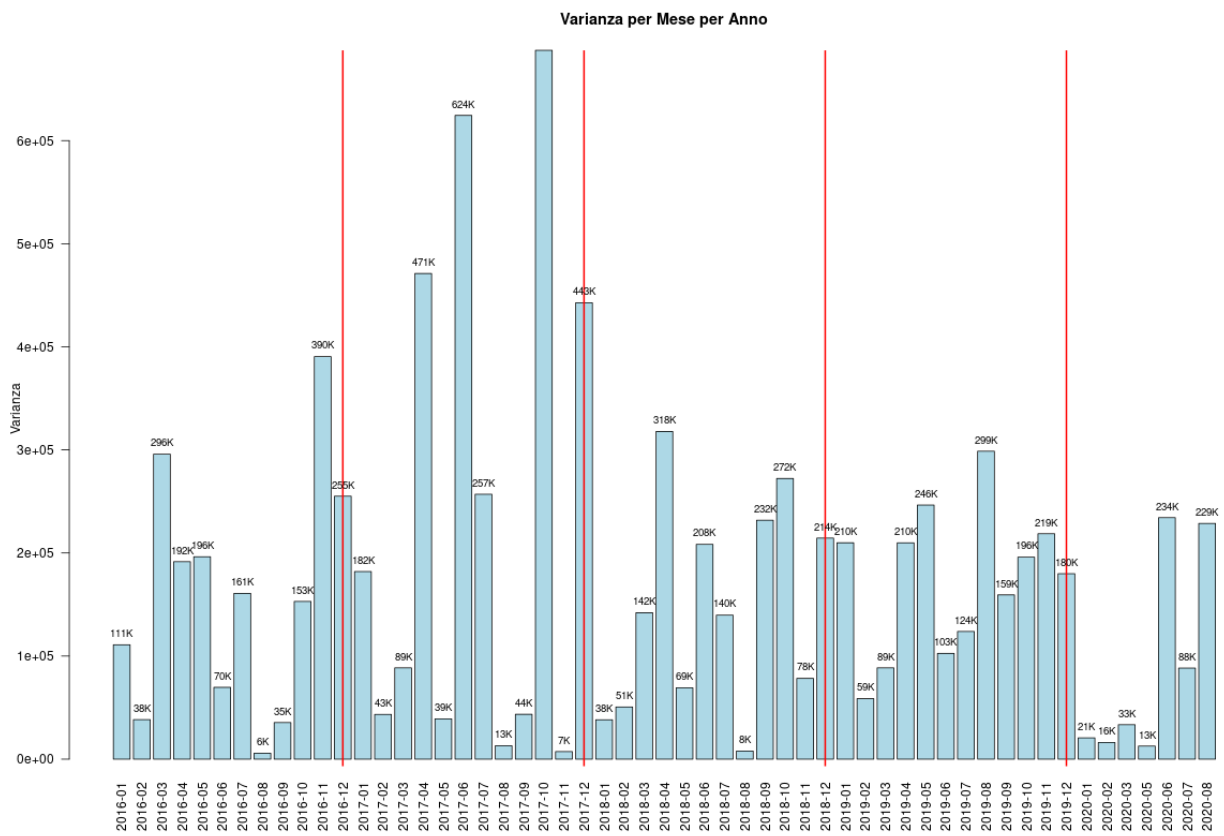
```
    var media = somma/num;
```

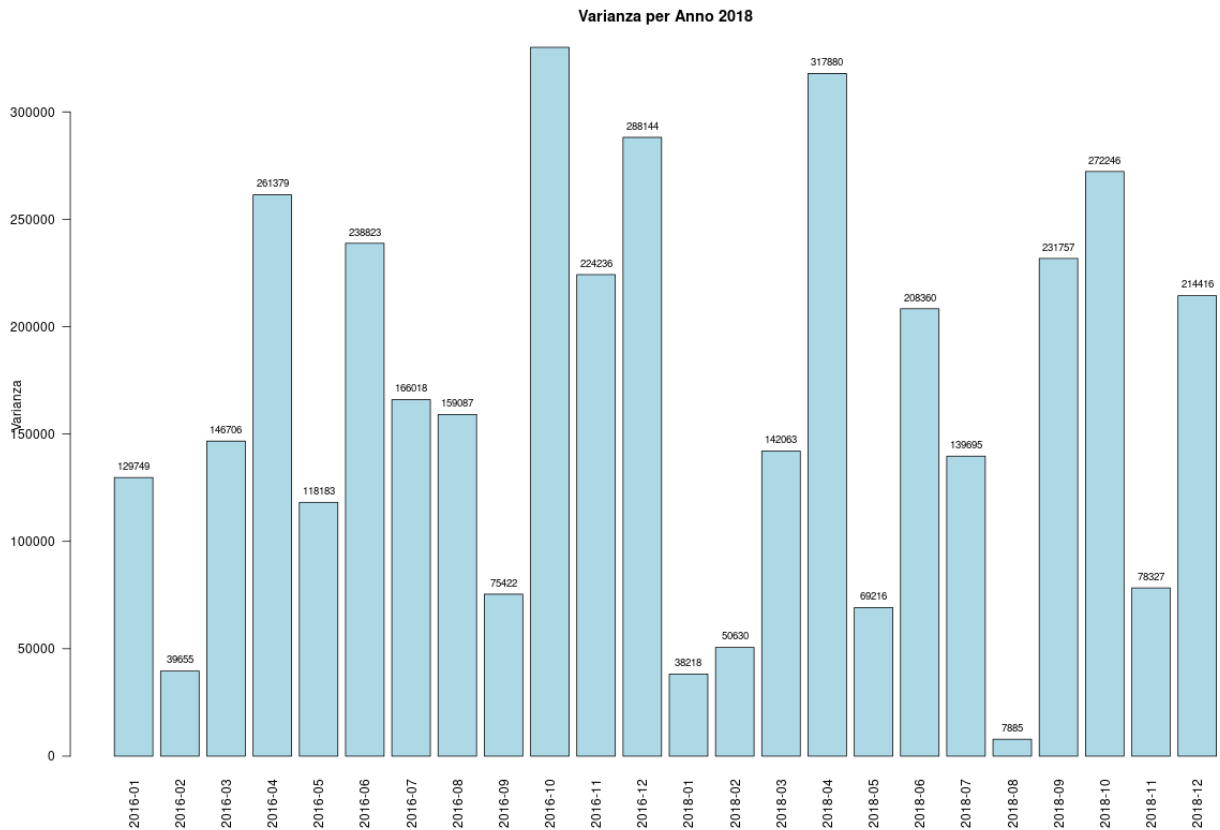
```
//VARIANZA: sommatoria[(x - mean)^2] / (num - 1)
var scarti = V_In_Reduce.map(function(item){
return (Math.pow(parseFloat(item) - media, 2))
});
var somma_scarti = scarti.reduce(function (accumulator, item){
return parseFloat(accumulator) + parseFloat(item);
});
var varianza = somma_scarti / (num-1);
return keyVal(K_In_Reduce, varianza.toFixed(2));
});
}
```

Job2.csv

```
"Varianza", "Anno", "Mese", "Numero_di_Vendite", "Periodo"
110972.691204257, 2016, "01", 62, "2016-01"
38351.0435100466, 2016, "02", 66, "2016-02"
295872.781138519, 2016, "03", 55, "2016-03"
191570.684792203, 2016, "04", 60, "2016-04"
196337.278810343, 2016, "05", 63, "2016-05"
69653.8695388889, 2016, "06", 73, "2016-06"
160827.842888087, 2016, "07", 61, "2016-07"
5731.73667285714, 2016, "08", 21, "2016-08"
35471.3577466949, 2016, "09", 62, "2016-09"
152986.941006066, 2016, "10", 61, "2016-10"
390485.049531911, 2016, "11", 59, "2016-11"
255096.324242831, 2016, "12", 67, "2016-12"
181979.970995667, 2017, "01", 54, "2017-01"
43393.8605261279, 2017, "02", 55, "2017-02"
88570.2054008469, 2017, "03", 58, "2017-03"
471101.399361996, 2017, "04", 32, "2017-04"
39083.5122050641, 2017, "05", 40, "2017-05"
624458.349562561, 2017, "06", 46, "2017-06"
256962.885998192, 2017, "07", 60, "2017-07"
13003.6894155263, 2017, "08", 20, "2017-08"
43505.1029134424, 2017, "09", 59, "2017-09"
687600.467134253, 2017, "10", 53, "2017-10"
7355.77151949495, 2017, "11", 45, "2017-11"
442718.918099512, 2017, "12", 41, "2017-12"
38217.6561699608, 2018, "01", 51, "2018-01"
50629.9956360324, 2018, "02", 39, "2018-02"
142062.611345507, 2018, "03", 46, "2018-03"
317879.844685161, 2018, "04", 43, "2018-04"
69215.7165818027, 2018, "05", 49, "2018-05"
208360.387216836, 2018, "06", 47, "2018-06"
139695.276144637, 2018, "07", 48, "2018-07"
7884.58927205882, 2018, "08", 17, "2018-08"
231756.571460917, 2018, "09", 42, "2018-09"
272246.003412605, 2018, "10", 35, "2018-10"
78327.0505054299, 2018, "11", 52, "2018-11"
214416.386952273, 2018, "12", 33, "2018-12"
209887.779501449, 2019, "01", 46, "2019-01"
58695.5622375, 2019, "02", 33, "2019-02"
88573.3934486803, 2019, "03", 52, "2019-03"
209881.555421951, 2019, "04", 41, "2019-04"
246470.71733561, 2019, "05", 41, "2019-05"
102591.096817719, 2019, "06", 43, "2019-06"
123837.395210083, 2019, "07", 47, "2019-07"
```

298707.347662667, 2019, "08", 25, "2019-08"
 159336.312372534, 2019, "09", 49, "2019-09"
 195923.47288987, 2019, "10", 56, "2019-10"
 218650.987764781, 2019, "11", 55, "2019-11"
 179880.289818538, 2019, "12", 34, "2019-12"
 20652.7359808067, 2020, "01", 48, "2020-01"
 16164.2672292473, 2020, "02", 31, "2020-02"
 33410.441012381, 2020, "03", 15, "2020-03"
 12680.2182185111, 2020, "05", 71, "2020-05"
 234286.611210453, 2020, "06", 42, "2020-06"
 88352.7118989204, 2020, "07", 39, "2020-07"
 228667.813247645, 2020, "08", 24, "2020-08"





| Hadoop Map Reduce concept-simulation made by Pirrone Daniele | | MAP | | REDUCE | |
|--|---------------------|---------------------|--|------------------------|--|
| INPUT-SPLITTING | MAPPING | SHUFFLING | MERGING | REDUCING | |
| « VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » | |
| « FATTURA,20160104,139.8 » | « 201601 - 139.8 » | « 201601 - 106.6 » | « 201601 - 106.6, 108.02, 109.47, 11.13, 11.37, 113.02, 13.81, 132.2, 139.8, 145.85, 15.4, 159.43, 160.65, 160.92, 177.71, 179.19, 180.28, 19.15, 204.55, 22.4, 226.79, 235.42, 24.52, 244, 244.39, 247.21, 247.84, 252.44, 252.7, 2537.6, 26.94, 271.83, 294.92, 32.37, 332.28, 34.53, 343.02, 35.49, 36.16, 36.88, 360.17, 37.05, 42.5, 503.81, 51.37, 52.24, 53.94, 535.84, 55.66, 57.44, 59.62, 635.4, 65.03, 68.99, 7.32, 7.38, 7.88, 75.68, 75.85, 80.15, 96.94, 97.17 » | « 201601 - 110972.69 » | |
| « FATTURA,20160104,160.92 » | « 201601 - 160.92 » | « 201601 - 108.02 » | « 201601 - 109.47 » | « 201602 - 38351.04 » | |
| « FATTURA,20160104,160.92 » | « 201601 - 65.03 » | « 201601 - 109.47 » | « 201601 - 11.13 » | « 201603 - 295872.78 » | |
| « FATTURA,20160104,65.03 » | « 201601 - 535.84 » | « 201601 - 11.13 » | « 201601 - 11.37 » | « 201604 - 191570.68 » | |
| « FATTURA,20160104,65.03 » | « 201601 - 75.68 » | « 201601 - 11.37 » | « 201601 - 132.2 » | « 201605 - 196337.28 » | |
| « FATTURA,20160104,65.03 » | « 201601 - 332.28 » | « 201601 - 113.02 » | « 201601 - 145.85 » | « 201606 - 69653.87 » | |
| « FATTURA,20160104,65.03 » | « 201601 - 343.02 » | « 201601 - 13.81 » | « 201601 - 159.43 » | « 201607 - 160827.84 » | |
| « FATTURA,20160104,65.03 » | « 201601 - 247.84 » | « 201601 - 132.2 » | « 201601 - 160.65 » | « 201608 - 5731.74 » | |
| « FATTURA,20160104,535.84 » | « 201601 - 75.85 » | « 201601 - 139.8 » | « 201601 - 179.19 » | « 201609 - 35471.36 » | |
| « FATTURA,20160104,535.84 » | « 201601 - 244.39 » | « 201601 - 145.85 » | « 201601 - 19.15 » | « 201610 - 152986.94 » | |
| « FATTURA,20160104,75.68 » | « 201601 - 36.16 » | « 201601 - 15.4 » | « 201601 - 226.79 » | « 201611 - 390485.05 » | |
| « FATTURA,20160104,75.68 » | « 201601 - 34.53 » | « 201601 - 159.43 » | « 201601 - 24.52 » | « 201612 - 255096.32 » | |
| « FATTURA,20160104,75.68 » | « 201601 - 51.37 » | « 201601 - 160.65 » | « 201601 - 247.21 » | « 201701 - 181979.97 » | |
| « FATTURA,20160104,332.28 » | « 201601 - 52.24 » | « 201601 - 160.92 » | « 201601 - 252.44 » | « 201702 - 43393.86 » | |
| « FATTURA,20160104,332.28 » | « 201601 - 113.02 » | « 201601 - 177.71 » | « 201601 - 2537.6 » | « 201703 - 88570.21 » | |
| « FATTURA,20160104,332.28 » | « 201601 - 97.17 » | « 201601 - 179.19 » | « 201601 - 271.83 » | « 201704 - 471101.40 » | |
| « FATTURA,20160105,343.02 » | « 201601 - 252.7 » | « 201601 - 180.28 » | « 201601 - 32.37 » | « 201705 - 39083.51 » | |
| « FATTURA,20160105,343.02 » | « 201601 - 55.66 » | « 201601 - 19.15 » | « 201601 - 34.53 » | « 201706 - 624458.35 » | |
| « FATTURA,20160105,343.02 » | « 201601 - 106.6 » | « 201601 - 204.55 » | « 201601 - 35.49 » | « 201707 - 256962.89 » | |
| « FATTURA,20160104,247.84 » | « 201601 - 96.94 » | « 201601 - 22.4 » | « 201601 - 57.44 » | « 201708 - 13003.69 » | |
| « FATTURA,20160104,247.84 » | « 201601 - 109.47 » | « 201601 - 226.79 » | « 201601 - 65.03 » | « 201709 - 43505.10 » | |
| « FATTURA,20160107,75.85 » | « 201601 - 24.52 » | « 201601 - 235.42 » | « 201601 - 7.38 » | « 201710 - 43505.10 » | |
| « FATTURA,20160107,75.85 » | « 201601 - 59.62 » | « 201601 - 24.52 » | « 201601 - 75.85 » | « 201711 - » | |
| « FATTURA,20160107,75.85 » | « 201601 - 42.5 » | « 201601 - 244 » | « 201601 - 97.17 » | « 201712 - » | |
| « FATTURA,20160107,244.39 » | « 201601 - 159.43 » | « 201601 - 244.39 » | « 201601 - 11.36 » | « 201713 - » | |
| « FATTURA,20160107,244.39 » | « 201601 - 204.55 » | « 201601 - 247.21 » | « 201601 - 12.49 » | « 201714 - » | |
| « FATTURA,20160107,36.16 » | « 201601 - 57.44 » | « 201601 - 247.84 » | « 201601 - 14.38 » | « 201715 - » | |
| « FATTURA,20160107,36.16 » | « 201601 - 132.2 » | « 201601 - 252.44 » | « 201601 - 142.06 » | « 201716 - » | |
| « FATTURA,20160107,34.53 » | « 201601 - 177.71 » | « 201601 - 252.7 » | « 201601 - 148.53 » | « 201717 - » | |
| « FATTURA,20160107,34.53 » | « 201601 - 235.42 » | « 201601 - 2537.6 » | « 201601 - 160.47 » | « 201718 - » | |
| « FATTURA,20160108,51.37 » | « 201601 - 19.15 » | « 201601 - 26.94 » | « 201601 - 18.17 » | « 201719 - » | |
| « FATTURA,20160108,51.37 » | « 201601 - 271.83 » | « 201601 - 271.83 » | « 201601 - 188.69 » | « 201720 - » | |
| « FATTURA,20160108,52.24 » | « 201601 - 252.44 » | « 201601 - 294.92 » | « 201601 - 190.15 » | « 201721 - » | |
| « FATTURA,20160108,52.24 » | « 201601 - 11.37 » | « 201601 - 32.37 » | « 201601 - 20.67 » | « 201722 - » | |
| « FATTURA,20160108,113.02 » | « 201601 - 226.79 » | « 201601 - 332.28 » | « 201601 - 219.6 » | « 201723 - » | |
| « FATTURA,20160108,113.02 » | « 201601 - 244 » | « 201601 - 34.53 » | « 201601 - 23.47 » | « 201724 - » | |
| « FATTURA,20160108,113.02 » | « 201601 - 32.37 » | « 201601 - 343.02 » | « 201601 - 242.29 » | « 201725 - » | |
| « FATTURA,20160108,97.17 » | « 201601 - 7.32 » | « 201601 - 35.49 » | « 201601 - 253.03 » | « 201726 - » | |
| « FATTURA,20160108,97.17 » | « 201601 - 160.65 » | « 201601 - 36.16 » | « 201601 - 257.02 » | « 201727 - » | |
| « FATTURA,20160108,252.7 » | « 201601 - 26.94 » | « 201601 - 36.88 » | « 201601 - 257.02 » | « 201728 - » | |
| « FATTURA,20160108,252.7 » | « 201601 - 635.4 » | « 201601 - 360.17 » | « 201601 - 257.02 » | « 201729 - » | |
| « FATTURA,20160109,55.66 » | « 201601 - 7.38 » | « 201601 - 37.05 » | « 201601 - 257.02 » | « 201730 - » | |
| « FATTURA,20160109,55.66 » | « 201601 - 13.81 » | « 201601 - 42.5 » | « 201601 - 257.02 » | « 201731 - » | |

Job3

Job3.r

```
#Installa il package per utilizzare SQL
install.packages("sqldf")

#Avvia la libreria SQLite
library(sqldf)

#Legge file da directory
file_contabilità <- read.csv("/home/manuel/Scrivania/BigData/HW/Ordini.csv", header=FALSE)

#Rinomina le colonne del file Ordini.csv
names(file_contabilità) <- c("Tipologia", "Data", "Prezzo")

#Avvia Query SQL
risultato <- sqldf("
WITH VenditeMensili AS (
SELECT
SUBSTR(Data, 1, 4) AS Anno,
SUBSTR(Data, 5, 2) AS Mese,
SUM(Prezzo) AS TotaleVendite
FROM
file_contabilità
WHERE
Tipologia = 'FATTURA' OR Tipologia = 'RICEVUTA'
GROUP BY
Anno, Mese
)

SELECT
vm.Anno AS Anno,
vm.Mese AS Mese,
vm.TotaleVendite AS TotaleVendite
FROM
VenditeMensili vm
JOIN (
SELECT
Anno,
MAX(TotaleVendite) AS MaxVendite
FROM
VenditeMensili
GROUP BY
Anno
) AS SubQuery
ON
vm.Anno = SubQuery.Anno AND vm.TotaleVendite = SubQuery.MaxVendite
ORDER BY
vm.Anno;
```

```
")
```

```
#Stampa Risultato  
print(risultato)
```

```
# Combina Anno e Mese per ottenere una rappresentazione nel formato "YYYY-MM"  
risultato$Periodo <- paste(risultato$Anno, risultato$Mese, sep="-")
```

```
# Converti la colonna Anno in numerico  
risultato$Anno <- as.numeric(risultato$Anno)
```

```
# Salva i risultati nel file job1.csv  
write.csv(risultato, file = "/home/manuel/Scrivania/BigData/HW/HW Mio/job3.csv", row.names  
= FALSE)
```

```
# Salva il grafico  
png(filename="/home/manuel/Scrivania/BigData/HW/HW Mio/job3grafico.png",width=1200,  
height=800)
```

```
# Calcola il valore massimo più un margine del 10%  
max_val_con_margine = max(risultato$TotaleVendite) * 1.10
```

```
# Crea un bar plot  
grafico = barplot(risultato$TotaleVendite,  
names.arg=risultato$Periodo,  
main="Maggiori Vendite per Mese per Anno",  
xlab="",  
ylab="Valori",  
col="lightblue",  
border="black",  
las=2,  
ylim=c(0, max_val_con_margine) # Imposta i limiti dell'asse y  
)
```

```
# Aggiungi valori al grafico  
text(grafico, risultato$TotaleVendite + 0.5, labels=round(risultato$TotaleVendite, 0), pos=3,  
cex=0.8)
```

```
# Chiudi il dispositivo di output, salvando l'immagine  
dev.off()
```

Job3.js

```
function jobInputSplit(input_str){  
return input_str.split('\n').filter(function(ordine){  
return (ordine.split(',')[0] === 'FATTURA' ||  
ordine.split(',')[0] === 'RICEVUTA'); });  
}
```

```

function jobMap(V_In_Map){
return V_In_Map.map(function(item){
var vendita = item.split(',');
return keyVal(vendita[1].substring(0, 6), vendita[2]);
});
}

function jobReduce(K_In_Reduce_V_In_Reduce){
var fatturato_aaaamm = K_In_Reduce_V_In_Reduce.map(function(items){
var K_In_Reduce = items.split(S[0])[0];
var V_In_Reduce = items.split(S[0])[1].split(S[1]);
var Reduce = V_In_Reduce.reduce(function (accumulator, item)
{
return parseFloat(accumulator) + parseFloat(item);
});
return keyVal(K_In_Reduce, Reduce.toFixed(2));
});
var V_Out_Map2 = merge(
fatturato_aaaamm.map(function(item) {
var arr = item.split(S[0]);
var key = arr[0].substring(0, 4);//anno
var val=arr[0].substring(4,6).concat(S[2]).concat(arr[1].toString());
//meseimporto
return keyVal(key, val);
})
);
var V_Out_Reduce2 = V_Out_Map2.map(function (items){
var K_In_Reduce = items.split(S[0])[0]; //anno
var V_In_Reduce = items.split(S[0])[1].split(S[1]);
//meseimporto
var Reduce = V_In_Reduce.reduce(function (maxItem,
currentItem) {
var maxImporto = 1 * maxItem.split(S[2])[1];
var currentImporto = 1 * currentItem.split(S[2])[1];
return currentImporto > maxImporto ? currentItem : maxItem;
});
return keyVal(K_In_Reduce, Reduce.split(S[2])[0]);
});
return V_Out_Reduce2;
}

```

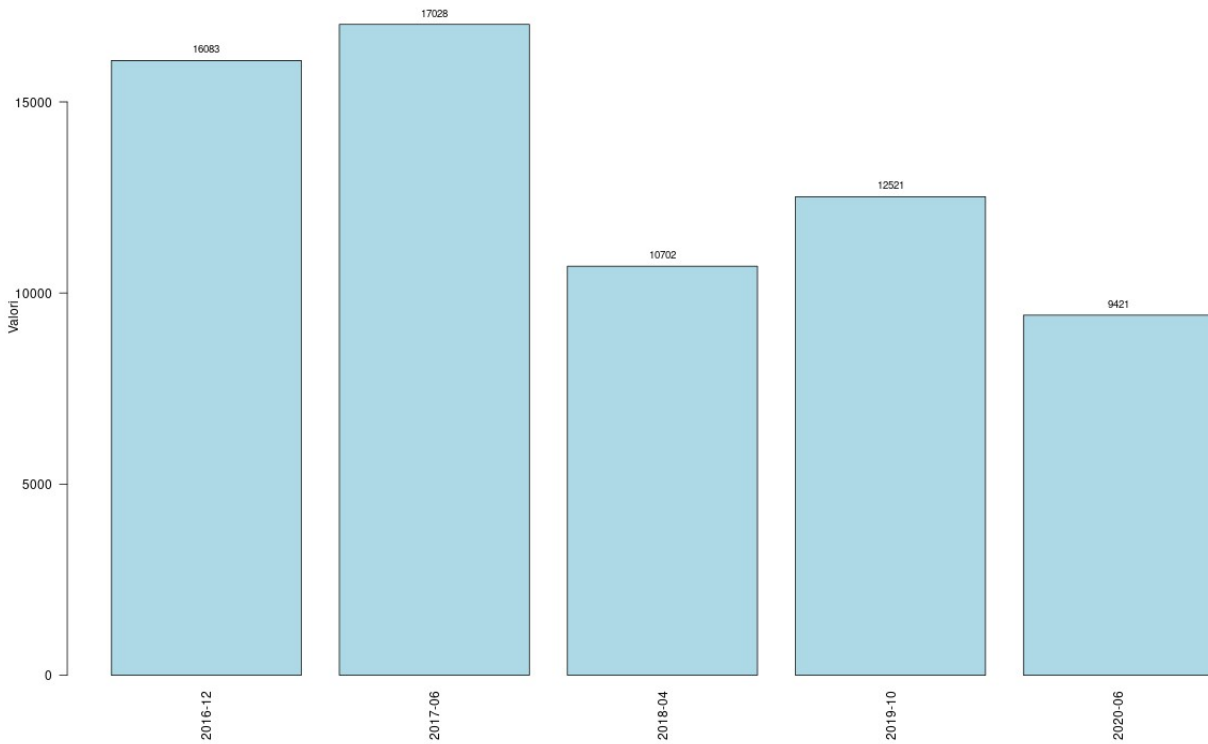
Job3.csv

```

"Anno", "Mese", "TotaleVendite", "Periodo"
2016, "12", 16083.28, "2016-12"
2017, "06", 17028.09, "2017-06"
2018, "04", 10701.99, "2018-04"
2019, "10", 12520.92, "2019-10"
2020, "06", 9420.84, "2020-06"

```

Maggiori Vendite per Mese per Anno



| Hadoop Map Reduce concept-simulation made by Pirrone Daniele | | MAP | | REDUCE | |
|--|---------------------|---------------------|--|-----------------|--|
| INPUT-SPLITTING | MAPPING | SHUFFLING | MERGING | REDUCING | |
| « VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » | |
| « FATTURA_20160104,139.8 » | « 201601 - 139.8 » | « 201601 - 106.6 » | « 201601 - 106.6, 108.02, 109.47, 11.13, 11.37, 113.02, 13.81, 132.2, 139.8, 145.85, 15.4, 159.43, 160.65, 160.92, 177.71, 179.19, 180.28, 19.15, 204.55, 22.4, 226.79, 235.42, 24.52, 244, 244.39, 247.21, 247.84, 252.44, 252.7, 2537.6, 26.94, 271.83, 294.92, 32.37, 332.28, 34.53, 343.02, 35.49, 36.16, 36.88, 360.17, 37.05, 42.5, 503.81, 51.37, 52.24, 53.94, 535.84, 55.66, 57.44, 59.62, 635.4, 65.03, 68.99, 7.32, 7.38, 7.88, 75.68, 75.85, 80.15, 96.94, 97.17 » | « 2016 - 12 » | |
| « FATTURA_20160104,160.92 » | « 201601 - 160.92 » | « 201601 - 108.02 » | « 201601 - 108.02 » | « 2017 - 06 » | |
| « FATTURA_20160104,65.03 » | « 201601 - 65.03 » | « 201601 - 109.47 » | « 201601 - 109.47 » | « 2018 - 04 » | |
| « FATTURA_20160104,535.84 » | « 201601 - 535.84 » | « 201601 - 11.13 » | « 201601 - 11.13 » | « 2019 - 10 » | |
| « FATTURA_20160104,75.68 » | « 201601 - 75.68 » | « 201601 - 11.37 » | « 201601 - 11.37 » | « 2020 - 06 » | |
| « FATTURA_20160104,332.28 » | « 201601 - 332.28 » | « 201601 - 113.02 » | « 201601 - 113.02 » | | |
| « FATTURA_20160105,343.02 » | « 201601 - 343.02 » | « 201601 - 13.81 » | « 201601 - 13.81 » | | |
| « FATTURA_20160104,247.84 » | « 201601 - 247.84 » | « 201601 - 132.2 » | « 201601 - 132.2 » | | |
| « FATTURA_20160107,75.85 » | « 201601 - 75.85 » | « 201601 - 139.8 » | « 201601 - 139.8 » | | |
| « FATTURA_20160107,244.39 » | « 201601 - 244.39 » | « 201601 - 145.85 » | « 201601 - 145.85 » | | |
| « FATTURA_20160107,36.16 » | « 201601 - 36.16 » | « 201601 - 15.4 » | « 201601 - 15.4 » | | |
| « FATTURA_20160107,34.53 » | « 201601 - 34.53 » | « 201601 - 159.43 » | « 201601 - 159.43 » | | |
| « FATTURA_20160108,51.37 » | « 201601 - 51.37 » | « 201601 - 160.65 » | « 201601 - 160.65 » | | |
| « FATTURA_20160108,52.24 » | « 201601 - 52.24 » | « 201601 - 160.92 » | « 201601 - 160.92 » | | |
| « FATTURA_20160108,113.02 » | « 201601 - 113.02 » | « 201601 - 177.71 » | « 201601 - 177.71 » | | |
| « FATTURA_20160108,97.17 » | « 201601 - 97.17 » | « 201601 - 179.19 » | « 201601 - 179.19 » | | |
| « FATTURA_20160108,252.7 » | « 201601 - 252.7 » | « 201601 - 180.28 » | « 201601 - 180.28 » | | |
| « FATTURA_20160109,55.66 » | « 201601 - 55.66 » | « 201601 - 19.15 » | « 201601 - 19.15 » | | |
| « FATTURA_20160111,106.6 » | « 201601 - 106.6 » | « 201601 - 204.55 » | « 201601 - 204.55 » | | |
| « FATTURA_20160111,96.94 » | « 201601 - 96.94 » | « 201601 - 22.4 » | « 201601 - 22.4 » | | |
| « FATTURA_20160111,109.47 » | « 201601 - 109.47 » | « 201601 - 226.79 » | « 201601 - 226.79 » | | |
| « FATTURA_20160112,24.52 » | « 201601 - 24.52 » | « 201601 - 235.42 » | « 201601 - 235.42 » | | |
| « FATTURA_20160112,59.62 » | « 201601 - 59.62 » | « 201601 - 24.52 » | « 201601 - 24.52 » | | |
| « FATTURA_20160112,42.5 » | « 201601 - 42.5 » | « 201601 - 244 » | « 201601 - 244 » | | |
| « FATTURA_20160112,159.43 » | « 201601 - 159.43 » | « 201601 - 244.39 » | « 201601 - 244.39 » | | |
| « FATTURA_20160112,204.55 » | « 201601 - 204.55 » | « 201601 - 247.21 » | « 201601 - 247.21 » | | |
| « FATTURA_20160113,57.44 » | « 201601 - 57.44 » | « 201601 - 247.84 » | « 201601 - 247.84 » | | |
| « FATTURA_20160113,132.2 » | « 201601 - 132.2 » | « 201601 - 252.44 » | « 201601 - 252.44 » | | |
| « FATTURA_20160113,177.71 » | « 201601 - 177.71 » | « 201601 - 252.7 » | « 201601 - 252.7 » | | |
| « FATTURA_20160114,235.42 » | « 201601 - 235.42 » | « 201601 - 2537.6 » | « 201601 - 2537.6 » | | |

Job4

Job4.r

```
#Installa il package per utilizzare SQL
install.packages("sqldf")

#Avvia la libreria SQLite
library(sqldf)

#Legge file da directory
file_contabilità <- read.csv("/home/manuel/Scrivania/BigData/HW/Ordini.csv", header=FALSE)

#Rinomina le colonne del file Ordini.csv
names(file_contabilità) <- c("Tipologia", "Data", "Prezzo")

#Avvia Query SQL
risultato <- sqldf("
WITH VenditeMensili AS (
SELECT
SUBSTR(Data, 1, 4) AS Anno,
SUBSTR(Data, 5, 2) AS Mese,
SUM(Prezzo) AS TotaleVendite
FROM
file_contabilità
WHERE
Tipologia = 'FATTURA' OR Tipologia = 'RICEVUTA'
GROUP BY
Anno, Mese
)

SELECT
vm.Anno AS Anno,
vm.Mese AS Mese,
vm.TotaleVendite AS TotaleVendite
FROM
VenditeMensili vm
JOIN (
SELECT
Anno,
MIN(TotaleVendite) AS MinVendite
FROM
VenditeMensili
GROUP BY
Anno
) AS SubQuery
ON
vm.Anno = SubQuery.Anno AND vm.TotaleVendite = SubQuery.MinVendite
ORDER BY
```

```

vm.Anno;

")

#Stampa Risultato
print(risultato)

# Combina Anno e Mese per ottenere una rappresentazione nel formato "YYYY-MM"
risultato$Periodo <- paste(risultato$Anno, risultato$Mese, sep="-")

# Converti la colonna Anno in numerico
risultato$Anno <- as.numeric(risultato$Anno)

# Salva i risultati nel file job1.csv
write.csv(risultato, file = "/home/manuel/Scrivania/BigData/HW/HW Mio/job4.csv", row.names
= FALSE)

# Salva il grafico
png(filename="/home/manuel/Scrivania/BigData/HW/HW Mio/job4grafico.png",width=1200,
height=800)

# Calcola il valore massimo più un margine del 10%
max_val_con_margine = max(risultato$TotaleVendite) * 1.10

# Crea un bar plot
grafico = barplot(risultato$TotaleVendite,
names.arg=risultato$Periodo,
main="Minori Vendite per Mese per Anno",
xlab="",
ylab="Vendite",
col="lightblue",
border="black",
las=2,
ylim=c(0, max_val_con_margine) # Imposta i limiti dell'asse y
)

# Aggiungi valori al grafico
text(grafico, risultato$TotaleVendite + 0.5, labels=round(risultato$TotaleVendite, 0), pos=3,
cex=0.8)

# Chiudi il dispositivo di output, salvando l'immagine
dev.off()

```

Job4.js

```

function jobInputSplit(input_str){
return input_str.split('\n').filter(function(ordine){
return ordine.split(',')[0] === 'FATTURA';
});
}

```

```

}
function jobMap(V_In_Map){
return V_In_Map.map(function(item){
var vendita = item.split(',');
return keyVal(vendita[1].substring(0, 6), vendita[2]);
});
}

function jobReduce(K_In_Reduce_V_In_Reduce){
var fatturato_aaaamm = K_In_Reduce_V_In_Reduce.map(function
(items){
var K_In_Reduce = items.split(S[0])[0];
var V_In_Reduce = items.split(S[0])[1].split(S[1]);
var Reduce = V_In_Reduce.reduce(function (accumulator, item)
{
return parseFloat(accumulator) + parseFloat(item);
});
return keyVal(K_In_Reduce, Reduce.toFixed(2));
});
var V_Out_Map2 = merge(
fatturato_aaaamm.map(function(item) {
var arr = item.split(S[0]);var key = arr[0].substring(0, 4);
//anno
var val = arr[0].substring(4,6).concat(S[2]).concat(arr[1].toString());//meseimporto
return keyVal(key, val);
})
);
var V_Out_Reduce2 = V_Out_Map2.map(function (items){
var K_In_Reduce = items.split(S[0])[0];//anno
var V_In_Reduce = items.split(S[0])[1].split(S[1]);
//meseimporto
var Reduce = V_In_Reduce.reduce(function (minItem, currentItem) {
var minImporto = 1 * minItem.split(S[2])[1];
var currentImporto = 1 * currentItem.split(S[2])[1];
return currentImporto < minImporto ? currentItem : minItem;
});
return keyVal(K_In_Reduce, Reduce.split(S[2])[0]);
});
return V_Out_Reduce2;
}

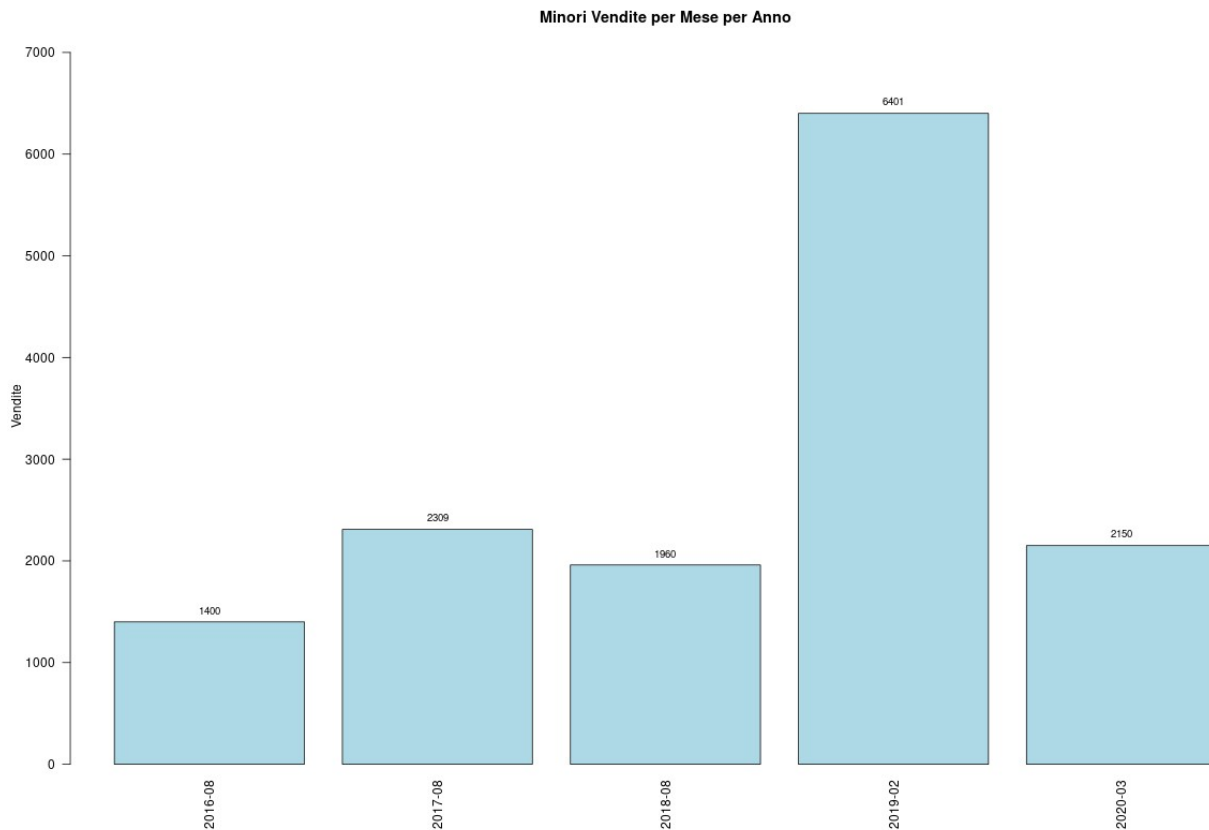
```

Job4.csv

```

"Anno", "Mese", "TotaleVendite", "Periodo"
2016, "08", 1400.04, "2016-08"
2017, "08", 2309.31, "2017-08"
2018, "08", 1959.88, "2018-08"
2019, "02", 6400.68, "2019-02"
2020, "03", 2150.03, "2020-03"

```



Hadoop Map Reduce
concept-simulation made
by Pirrone Daniele

MAP

REDUCE

| INPUT-SPLITTING | MAPPING | SHUFFLING | MERGING | REDUCING |
|---|---|--|---|---|
| « VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » | « KEY - VALUE » |
| « FATTURA,20160104,139.8 » « FATTURA,20160104,160.92 » « FATTURA,20160104,65.03 » « FATTURA,20160104,535.84 » « FATTURA,20160104,75.68 » « FATTURA,20160104,332.28 » « FATTURA,20160105,343.02 » « FATTURA,20160104,247.84 » « FATTURA,20160107,75.85 » « FATTURA,20160107,244.39 » « FATTURA,20160107,36.16 » « FATTURA,20160107,34.53 » « FATTURA,20160108,51.37 » « FATTURA,20160108,52.24 » « FATTURA,20160108,113.02 » « FATTURA,20160108,97.17 » « FATTURA,20160108,252.7 » « FATTURA,20160109,55.66 » « FATTURA,20160111,106.6 » « FATTURA,20160111,96.94 » « FATTURA,20160111,109.47 » « FATTURA,20160112,24.52 » « FATTURA,20160112,59.62 » « FATTURA,20160112,42.5 » « FATTURA,20160112,159.43 » « FATTURA,20160112,204.55 » « FATTURA,20160113,57.44 » « FATTURA,20160113,132.2 » « FATTURA,20160113,177.71 » « FATTURA,20160114,235.42 » | « 201601 - 139.8 » « 201601 - 160.92 » « 201601 - 65.03 » « 201601 - 535.84 » « 201601 - 75.68 » « 201601 - 332.28 » « 201601 - 343.02 » « 201601 - 247.84 » « 201601 - 75.85 » « 201601 - 244.39 » « 201601 - 36.16 » « 201601 - 34.53 » « 201601 - 51.37 » « 201601 - 52.24 » « 201601 - 113.02 » « 201601 - 97.17 » « 201601 - 252.7 » « 201601 - 55.66 » « 201601 - 106.6 » « 201601 - 96.94 » « 201601 - 109.47 » « 201601 - 24.52 » « 201601 - 59.62 » « 201601 - 42.5 » « 201601 - 159.43 » « 201601 - 204.55 » « 201601 - 57.44 » « 201601 - 132.2 » « 201601 - 177.71 » « 201601 - 235.42 » | « 201601 - 106.6 » « 201601 - 108.02 » « 201601 - 109.47 » « 201601 - 11.13 » « 201601 - 11.37 » « 201601 - 113.02 » « 201601 - 13.81 » « 201601 - 132.2 » « 201601 - 139.8 » « 201601 - 145.85 » « 201601 - 15.4 » « 201601 - 159.43 » « 201601 - 160.65 » « 201601 - 160.92 » « 201601 - 177.71 » « 201601 - 179.19 » « 201601 - 180.28 » « 201601 - 19.15 » « 201601 - 204.55 » « 201601 - 22.4 » « 201601 - 226.79 » « 201601 - 235.42 » « 201601 - 24.52 » « 201601 - 244 » « 201601 - 244.39 » « 201601 - 247.21 » « 201601 - 247.84 » « 201601 - 252.44 » « 201601 - 252.7 » « 201601 - 2537.6 » | « 201601 - 106.6, 108.02, 109.47, 11.13, 11.37, 113.02, 13.81, 132.2, 139.8, 145.85, 15.4, 159.43, 160.65, 160.92, 177.71, 179.19, 180.28, 19.15, 204.55, 22.4, 226.79, 235.42, 24.52, 244, 244.39, 247.21, 247.84, 252.44, 252.7, 2537.6, 26.94, 271.83, 294.92, 32.37, 332.28, 34.53, 343.02, 35.49, 36.16, 36.88, 360.17, 37.05, 42.5, 503.81, 51.37, 52.24, 53.94, 535.84, 55.66, 57.44, 59.62, 635.4, 65.03, 68.99, 7.32, 7.38, 7.88, 75.68, 75.85, 80.15, 96.94, 97.17 » « 201602 - 104.86, 11.36, 112.98, 12.49, 122.31, 14.38, 14.47, 14.96, 142.06, 143.81, 148.53, 155.51, 160.47, 178.91, 18.17, 18.71, 180, 188.69, 19.76, 190.15, 194.96, 20.67, 217.99, 219.6, 23.08, 23.18, 23.47, 24.56, 242.29, 25.72, 253.03, 257.02, 261.02, 265.84, 27.67, 30.4, 330.67, 35.9, 36.6, 37.93, 40.26, 40.26, 41.36, 44.73, 45.46, 50.74, 50.95, 54.5, 56.07, 56.18, 59.29, 62.65, 63.9, 7.06, 71.28, 8.64, 81.41, 83, 83.81, 84.56, 88.04, 917.7, 939.4, 97.2, 976, 98.28 » | « 2016 - 08 » « 2017 - 08 » « 2018 - 08 » « 2019 - 12 » « 2020 - 03 » |

Previsioni Future

Previsioni.r

```
# Installa e carica i pacchetti necessari
install.packages("forecast")
install.packages("dplyr")
library(forecast)
library(dplyr)

# Legge il file CSV
dati <- read.csv("/home/manuel/Scrivania/BigData/HW/HW Mio/job1/job1.csv")

# Converte l'anno e il mese in una colonna di date per una serie temporale
dati$data_ts <- as.Date(paste(dati$Anno, dati$Mese, "01", sep="-"))

# Ordina i dati in base alla data
dati <- dati %>% arrange(data_ts)

# Crea la serie temporale
serie_temporale <- ts(dati$media_valore, start=c(min(dati$Anno), min(dati$Mese)),
frequency=12)

# Addestra un modello ARIMA
modello_arima <- auto.arima(serie_temporale)

# Effettua la previsione per i prossimi 12 mesi
previsione <- forecast(modello_arima, h=12)

# Salvare il file di output
png(filename = "/home/manuel/Scrivania/BigData/HW/HW Mio/previsione.png", width = 1200,
height = 800)

# Visualizza la previsione
plot(previsione, main="Previsione anno 2020/2021", xlab="Anno", ylab="Media Vendite")

# Chiudere il dispositivo grafico, salvando il file
dev.off()
```

Previsione anno 2020/2021

