Corso di Ingegneria Informatica A.A. 2022/2023

PROGETTO CONTABILITÁ

Nome dello studente:	*Ugo Galliano*
Matricola	*724HHHINGINFOR*
Data Appello	*Luglio 2023*
Esame	*Introduzione Big Data*

Indice

Introduzione

In questo documento viene presentato il processo di svolgimento di un esercizio di contabilità proposto nel corso "Introduzione ai Big Data". L'obiettivo dell'esercizio è sviluppare un'implementazione in linguaggio R e un'implementazione utilizzando il paradigma mapReduce con l'aiuto di un Tool fornito dal Professore Daniele Pirrone.

Si suppone di avere un file in formato CSV chiamato "Ordini", in cui ogni record contiene tre campi:

- TipoDocumento
- Data (nel formato AAAAMMGG)
- Costo

Sono stati richiesti i seguenti compiti:

1) Calcolare la media delle vendite per ogni mese di ogni anno.
2) Calcolare la varianza delle vendite per ogni mese di ogni anno.
3) Identificare il mese di ogni anno con il valore di vendita più alto.
4) Identificare il mese di ogni anno con il valore di vendita più basso.

Successivamente, è stato scritto del codice in linguaggio R per visualizzare graficamente i risultati ottenuti confrontandoli con quelli ottenuti utilizzando il Tool.

Svolgimento

Dopo l'analisi del file "ordini.csv", sono state individuate in ogni record del file tre valori: TipoDocumento, Data, Costo.

- **TipoDocumento** è un dato categorico che rappresenta categorie distinte e non numeriche.
  - FATTURA
  - RICEVUTA
  - NOTA.DI.CREDITO
  - DDT
  - INVENTARIO
  - BUONO.PRELIEVO
  - OFFERTA
  - PREVENTIVO
- **Data** nel formato AAAAMMGG.
- **Costo** un tipo di dato numerico con cifre decimali.

L'obiettivo dell'esercizio è lavorare con il file "ordini.csv", che contiene dati relativi alla vendita di una determinata ditta. Per focalizzarsi sulla vendita, sono stati presi in considerazione solo i dati riguardanti fatture e ricevute, mentre gli altri tipi di ordine sono stati trascurati poiché non rilevanti per la vendita.
Per ottenere i risultati richiesti, è stato utilizzato il Tool di mapReduce. Questo strumento è stato impiegato per analizzare i dati e ricavare le informazioni necessarie dai job specificati nell'esercizio.
I risultati ottenuti utilizzando il Tool di mapReduce sono stati poi confrontati con quelli calcolati in precedenza tramite Rstudio.

Rstudio è un ambiente di sviluppo integrato (IDE) per il linguaggio di programmazione R. È spesso utilizzato per l'analisi dei dati e la generazione di statistiche. Nell'esercizio, Rstudio è stato impiegato per ottenere gli stessi dati richiesti precedentemente tramite il Tool di mapReduce.
Il confronto tra i risultati ottenuti dai due metodi è importante per verificare la correttezza delle analisi e la consistenza dei dati.

Utilizzo Tool

- Calcolare la media di vendite per ogni mese di ogni anno
- Calcolare la varianza di vendite per ogni mese di ogni anno

MAP

La funzione di mapping considera ogni singolo record e determina una chiave in base al tipo di documento. Se il tipo di documento è "FATTURA" o "RICEVUTA", viene assegnata una chiave specifica basata sull'anno e il mese del record. Altrimenti, se il tipo di documento non è di interesse per il calcolo, viene assegnata la chiave "Non calcolare".

In entrambi i casi, la stessa funzione di mapping viene utilizzata sia per il calcolo della media che per il calcolo della varianza. Questo significa che i record vengono trattati in modo simile, con l'unica differenza nella creazione della chiave in base al tipo di documento.

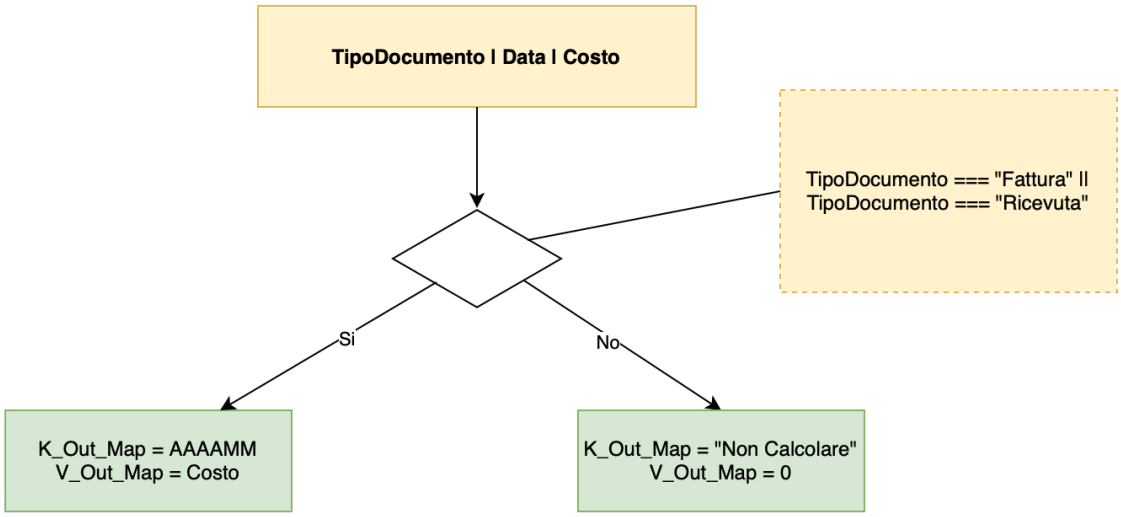

Figura 1 - Schema funzione mapping Media-Varianza

REDUCE MEDIA

All'interno della funzione "Reduce", viene calcolata la somma delle vendite per il mese specifico. Durante questo calcolo, viene incrementato un contatore per tenere traccia dei record validi, ovvero dei mesi in cui sono state effettuate vendite.
Successivamente, per calcolare la media delle vendite, il risultato della somma viene diviso per il valore del contatore. Questo permette di ottenere la media delle vendite per il mese specifico, tenendo conto solo dei mesi in cui sono presenti dati validi.

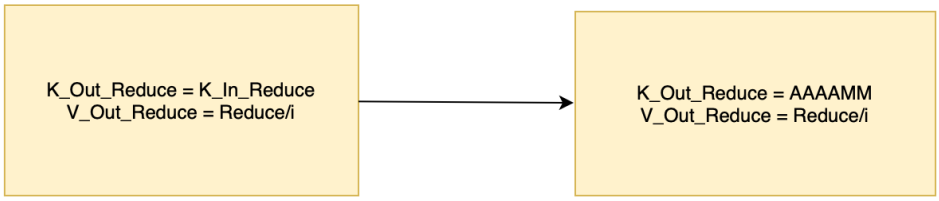

Figura 2 - Schema funzione reduce Media

REDUCE VARIANZA

A partire dal calcolo della media effettuato precedentemente, per il calcolo della varianza si parte dal valore della media per poi procedere con la somma degli scarti quadratici e dividere il risultato per il numero degli elementi.

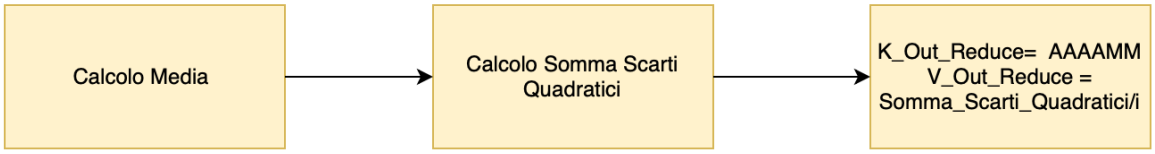

Figura 3 - Schema funzione Reduce Varianza

- Identificare il mese di ogni anno con il valore di vendita più alto.
- Identificare il mese di ogni anno con il valore di vendita più basso.

MAP

Dopo aver calcolato la somma delle vendite di ogni mese per ogni anno, il tool è stato riapplicato per identificare il mese di ogni anno con i valori delle vendite più alte e più basse.
In entrambi i casi, si è assegnato il valore dell'anno individuato alla chiave "K_Out_Map". Per quanto riguarda il valore della chiave "V_Out_Map", si è costruito una stringa che include il mese e la corrispondente somma delle vendite.

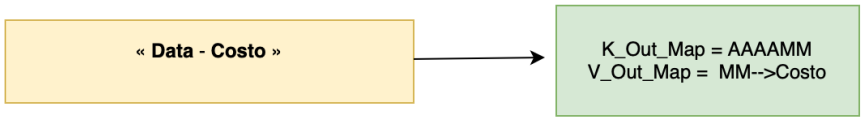

Figura 4 – Schema funzione Mapping Max Vendita / Min Vendita

REDUCE MESE MAGGIORE/MINORE VENDITA

Per individuare il mese con maggiore/minore vendita viene confrontato il valore di ogni item salvando in una variabile il valore di riferimento.
Si procede allo stesso modo anche per il calcolo della minor vendita considerando in questo caso, nel confronto di ogni item, il valore più piccolo.

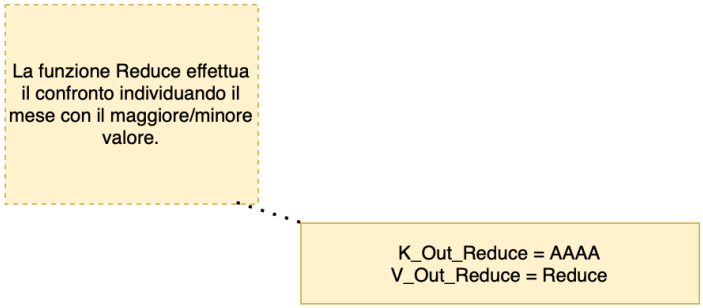

Figura 5 - Schema funzione Reduce Maggiore/Minore Vendita

Risultati Tool (Media e Varianza vendite)

Job Input Split
I record vengono suddivisi utilizzando il carattere di ritorno a capo (\n), ottenendo così vari record separati.

JobMap
Nella fase di mapping del processo se il tipo di ordine non corrisponde a FATTURA O RICEVUTA, il metodo restituisce Kmap = "Non Calcolare" e Vmap = 0. Altrimenti, se il record corrente rappresenta una fattura o una ricevuta, i valori restituiti saranno Kmap = AAAAMM (per raggruppare per anno e mese) e Vmap = costo.

Reduce Media
Nella fase di reduce, inizialmente viene calcolata la somma degli importi. Successivamente, questa somma viene divisa per il numero di elementi presenti riportati da un contatore precedentemente incrementato, ottenendo così la media.

Reduce Varianza
Il reduce per il calcolo della varianza è molto simile a quello per la media. Per calcolare la varianza, è necessario conoscere il valore medio degli elementi. Dopo aver calcolato il valore medio, viene utilizzata una funzione aggiuntiva per calcolare lo scarto quadratico degli elementi. Successivamente, la somma degli scarti quadrati viene divisa per il numero di elementi, che è stato precedentemente calcolato e memorizzato utilizzando un contatore, ottenendo così la varianza delle vendite.

MEDIA

Upload Text File or drop below

```
FATTURA,20160104,139.8
FATTURA,20160104,160.92
FATTURA,20160104,65.03
FATTURA,20160104,535.84
FATTURA,20160104,75.68
```

Script loaded (show/hide) [upload (0)]

Run MapReduce or CTRL+ENTER | Reload Page

Hadoop Map Reduce concept-simualtion 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,	« 201601 -
« FATTURA,20160104,160.92 »	« 201601 - 160.92 »	« 201601 - 108.02 »	108.02, 109.47, 11.13,	179.93080645161288 »
« FATTURA,20160104,65.03 »	« 201601 - 65.03 »	« 201601 - 109.47 »	11.37, 113.02, 13.81,	« 201602 -
« FATTURA,20160104,535.84 »	« 201601 - 535.84 »	« 201601 - 11.13 »	132.2, 139.8, 145.85,	131.92287878787877 »
« FATTURA,20160104,75.68 »	« 201601 - 75.68 »	« 201601 - 11.37 »	15.4, 159.43, 160.65,	« 201603 -
« FATTURA,20160104,332.28 »	« 201601 - 332.28 »	« 201601 - 113.02 »	160.92, 177.71,	222.55799999999996 »
« FATTURA,20160105,343.02 »	« 201601 - 343.02 »	« 201601 - 13.81 »	179.19, 180.28, 19.15,	« 201604 -
« FATTURA,20160104,247.84 »	« 201601 - 247.84 »	« 201601 - 132.2 »	204.55, 22.4, 226.79,	182.44099999999995 »
« FATTURA,20160107,75.85 »	« 201601 - 75.85 »	« 201601 - 139.8 »	235.42, 24.52, 244,	« 201605 -
« FATTURA,20160107,244.39 »	« 201601 - 244.39 »	« 201601 - 145.85 »	244.39, 247.21,	200.4126984126984 »
« FATTURA,20160107,36.16 »	« 201601 - 36.16 »	« 201601 - 15.4 »	247.84, 252.44, 252.7,	« 201606 - 143.58 »
« FATTURA,20160107,34.53 »	« 201601 - 34.53 »	« 201601 - 159.43 »	2537.6, 26.94, 271.83,	« 201607 -
« FATTURA,20160108,51.37 »	« 201601 - 51.37 »	« 201601 - 160.65 »	294.92, 32.37, 332.28,	185.62950819672128 »
« FATTURA,20160108,52.24 »	« 201601 - 52.24 »	« 201601 - 160.92 »	34.53, 343.02, 35.49,	« 201608 -
« FATTURA,20160108,113.02 »	« 201601 - 113.02 »	« 201601 - 177.71 »	36.16, 36.88, 360.17,	66.66857142857143 »

Figura 6 - Risultati Media Tool Map Reduce

VARIANZA

Run MapReduce or CTRL+ENTER | Reload Page

Hadoop Map Reduce concept-simualtion 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,	« 201601 -
« FATTURA,20160104,160.92 »	« 201601 - 160.92 »	« 201601 - 108.02 »	108.02, 109.47, 11.13,	109182.80908805925 »
« FATTURA,20160104,65.03 »	« 201601 - 65.03 »	« 201601 - 109.47 »	11.37, 113.02, 13.81,	« 201602 -
« FATTURA,20160104,535.84 »	« 201601 - 535.84 »	« 201601 - 11.13 »	132.2, 139.8, 145.85,	37769.967093227744 »
« FATTURA,20160104,75.68 »	« 201601 - 75.68 »	« 201601 - 11.37 »	15.4, 159.43, 160.65,	« 201603 -
« FATTURA,20160104,332.28 »	« 201601 - 332.28 »	« 201601 - 113.02 »	160.92, 177.71,	290493.27602690906 »
« FATTURA,20160105,343.02 »	« 201601 - 343.02 »	« 201601 - 13.81 »	179.19, 180.28, 19.15,	« 201604 -
« FATTURA,20160104,247.84 »	« 201601 - 247.84 »	« 201601 - 132.2 »	204.55, 22.4, 226.79,	188377.84004566658 »
« FATTURA,20160107,75.85 »	« 201601 - 75.85 »	« 201601 - 139.8 »	235.42, 24.52, 244,	« 201605 -
« FATTURA,20160107,244.39 »	« 201601 - 244.39 »	« 201601 - 145.85 »	244.39, 247.21,	193220.81406732177 »
« FATTURA,20160107,36.16 »	« 201601 - 36.16 »	« 201601 - 15.4 »	247.84, 252.44, 252.7,	« 201606 -
« FATTURA,20160107,34.53 »	« 201601 - 34.53 »	« 201601 - 159.43 »	2537.6, 26.94, 271.83,	68699.70694246574 »
« FATTURA,20160108,51.37 »	« 201601 - 51.37 »	« 201601 - 160.65 »	294.92, 32.37, 332.28,	« 201607 -
« FATTURA,20160108,52.24 »	« 201601 - 52.24 »	« 201601 - 160.92 »	34.53, 343.02, 35.49,	158191.32087352863 »
« FATTURA,20160108,113.02 »	« 201601 - 113.02 »	« 201601 - 177.71 »	36.16, 36.88, 360.17,	« 201608 -
« FATTURA,20160108,97.17 »	« 201601 - 97.17 »	« 201601 - 179.19 »	37.05, 42.5, 503.81,	5458.796831292518 »
« FATTURA,20160108,252.7 »	« 201601 - 252.7 »	« 201601 - 180.28 »	51.37, 52.24, 53.94,	« 201609 -
« FATTURA,20160109,55.66 »	« 201601 - 55.66 »	« 201601 - 19.15 »	535.84, 55.66, 57.44,	34899.239073361074 »
« FATTURA,20160111,106.6 »	« 201601 - 106.6 »	« 201601 - 204.55 »	59.62, 635.4, 65.03,	« 201610 -
« FATTURA,20160111,96.94 »	« 201601 - 96.94 »	« 201601 - 22.4 »	68.99, 7.32, 7.38, 7.88,	150478.95836662196 »
« FATTURA,20160111,109.47 »	« 201601 - 109.47 »	« 201601 - 226.79 »	75.68, 75.85, 80.15,	« 201611 -
« FATTURA,20160112,24.52 »	« 201601 - 24.52 »	« 201601 - 235.42 »	96.94, 97.17 »	383866.6588618788 »
« FATTURA,20160112,59.62 »	« 201601 - 59.62 »	« 201601 - 24.52 »	« 201602 - 104.86,	« 201612 -
« FATTURA,20160112,42.5 »	« 201601 - 42.5 »	« 201601 - 244 »	11.36, 112.98, 12.49,	251288.91641831142 »
« FATTURA,20160112,159.43 »	« 201601 - 159.43 »	« 201601 - 244.39 »	122.31, 14.38, 14.47,	« 201701 -
« FATTURA,20160112,204.55 »	« 201601 - 204.55 »	« 201601 - 247.21 »	14.96, 142.06, 143.81,	178609.9715327846 »
« FATTURA,20160113,57.44 »	« 201601 - 57.44 »	« 201601 - 247.84 »	148.53, 155.51,	« 201702 -
« FATTURA,20160113,132.2 »	« 201601 - 132.2 »	« 201601 - 252.44 »	160.47, 178.91, 18.17,	42604.881243834694 »

Figura 7 - Risultati Varianza Tool Map Reduce

Risultati Tool (Mese all'anno con max e min vendite)

Dopo aver calcolato la somma delle vendite di ogni mese per ogni anno, il tool è stato riapplicato per identificare il mese di ogni anno con i valori delle vendite più alte e più basse.

La somma delle vendite di ogni mese per ogni anno, ha restituito dei record in cui la chiave è "AAAAMM" e il valore è proprio la somma delle vendite del mese.
Successivamente il tool è stato rieseguito in modo da cambiare la chiave "AAAAMM" in "AAAA" ed il valore "MM ---> IMPORTO".

CALCOLA SOMMA DEL MESE PER OGNI ANNO

Job Input Split
Sono stati restituiti i vari record separati dal carattere ritorno a capo (/n).

JobMap
Nella fase di map del processo se il tipo di ordine non corrisponde a FATTURA O RICEVUTA, il metodo restituisce Kmap = "Non Calcolare" e Vmap = 0. Altrimenti, se il record corrente rappresenta una fattura o una ricevuta, i valori restituiti saranno Kmap = AAAAMM (per raggruppare per anno e mese) e Vmap = costo.

Reduce Somma
La funzione di reduce è stata in questo caso una somma degli importi.
Da questo primo algoritmo quindi sono stati ottenuti risultati di questo tipo:

<<AAAAMM,IMPORTO>>
<<AAAAMM,IMPORTO>>

SOMMA

Upload Text File or drop below

```
FATTURA,20160104,139.8
FATTURA,20160104,160.92
FATTURA,20160104,65.03
FATTURA,20160104,535.84
FATTURA,20160104,75.68
```

Script loaded (show/hide) [upload (0)]

Run MapReduce or CTRL+ENTER | Reload Page

Hadoop Map Reduce concept-simualtion 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,	« 201601 - 11155.71 »
« FATTURA,20160104,160.92 »	« 201601 - 160.92 »	« 201601 - 108.02 »	108.02, 109.47, 11.13,	« 201602 - 8706.91 »
« FATTURA,20160104,65.03 »	« 201601 - 65.03 »	« 201601 - 109.47 »	11.37, 113.02, 13.81,	« 201603 -
« FATTURA,20160104,535.84 »	« 201601 - 535.84 »	« 201601 - 11.13 »	132.2, 139.8, 145.85,	12240.689999999999 »
« FATTURA,20160104,75.68 »	« 201601 - 75.68 »	« 201601 - 11.37 »	15.4, 159.43, 160.65,	« 201604 -
« FATTURA,20160104,332.28 »	« 201601 - 332.28 »	« 201601 - 113.02 »	160.92, 177.71,	10946.459999999997 »
« FATTURA,20160105,343.02 »	« 201601 - 343.02 »	« 201601 - 13.81 »	179.19, 180.28, 19.15,	« 201605 - 12626 »
« FATTURA,20160104,247.84 »	« 201601 - 247.84 »	« 201601 - 132.2 »	204.55, 22.4, 226.79,	« 201606 - 10481.34 »
« FATTURA,20160107,75.85 »	« 201601 - 75.85 »	« 201601 - 139.8 »	235.42, 24.52, 244,	« 201607 -
« FATTURA,20160107,244.39 »	« 201601 - 244.39 »	« 201601 - 145.85 »	244.39, 247.21,	11323.399999999998 »
« FATTURA,20160107,36.16 »	« 201601 - 36.16 »	« 201601 - 15.4 »	247.84, 252.44, 252.7,	« 201608 - 1400.04 »
« FATTURA,20160107,34.53 »	« 201601 - 34.53 »	« 201601 - 159.43 »	2537.6, 26.94, 271.83,	« 201609 -
« FATTURA,20160108,51.37 »	« 201601 - 51.37 »	« 201601 - 160.65 »	294.92, 32.37, 332.28,	7398.759999999998 »
« FATTURA,20160108,52.24 »	« 201601 - 52.24 »	« 201601 - 160.92 »	34.53, 343.02, 35.49,	« 201610 -
« FATTURA,20160108,113.02 »	« 201601 - 113.02 »	« 201601 - 177.71 »	36.16, 36.88, 360.17,	10640.960000000001 »

Figura 8 - Risultati Somma Tool Map Reduce

CALCOLA MESE DI OGNI ANNO CON MAX/MIN VENDITE

Il risultato ottenuto è stato poi analizzato una seconda volta.

Job Input Split
Sono stati restituiti i vari record separati dal carattere ritorno a capo (/n).

JobMap
Per prima cosa, è stato necessario pulire i record in analisi, poiché erano formattati con parentesi angolate. È stato eseguito un processo di pulizia utilizzando la funzione "**replace**" per trasformare il formato da "<<AAAAMM,IMPORTO>>" a "AAAA,IMPORTO".

Successivamente, l'obiettivo era raggruppare i dati per anno, quindi la chiave "K_Out_Map" è stata impostata come "AAAA".

Nel caso del "V_OUT_MAP", invece, è stato necessario concatenare il mese con l'importo, poiché l'esercizio richiedeva di individuare il mese con le vendite maggiori e minori.
L'importo è un valore che verrà utilizzato successivamente nella fase di riduzione dei dati. Di conseguenza, la chiave "V_OUT_MAP" è stata impostata come "MM" --> IMPORTO.

Job Reduce
All'interno della funzione di riduzione (Reduce), l'accumulatore è stato utilizzato per memorizzare il valore più alto o più basso.
Nel processo di riduzione, le stringhe vengono separate per ottenere il mese e l'importo come due variabili distinte. Successivamente, gli importi vengono confrontati: se l'importo dell'elemento corrente è inferiore (per calcolare il mese con le vendite minori) o superiore (per calcolare il mese con le vendite maggiori) rispetto a quello presente nell'accumulator, l'elemento restituito diventa l'item corrente. In caso contrario, l'elemento restituito sarà quello precedentemente presente nell'accumulator.

MESE DI OGNI ANNO CON MAX VENDITA PER ANNO

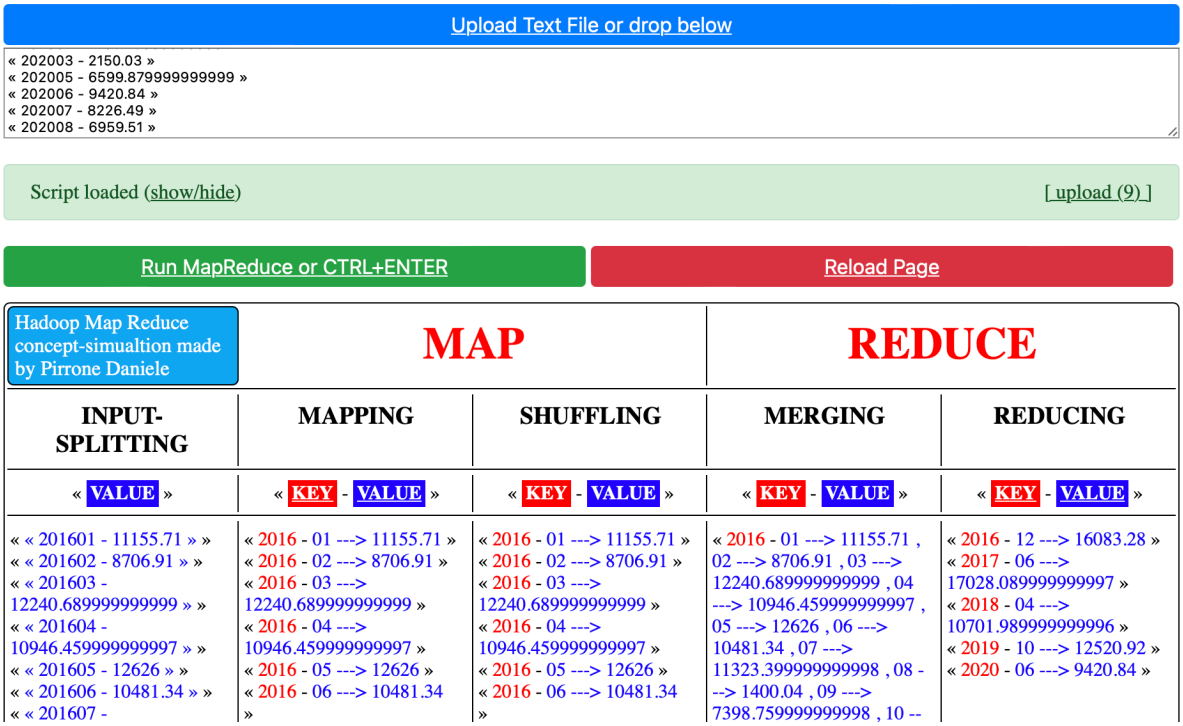

Figura 9 - Risultati mese con max vendite per anno

MESE DI OGNI ANNO CON MIN VENDITA PER ANNO

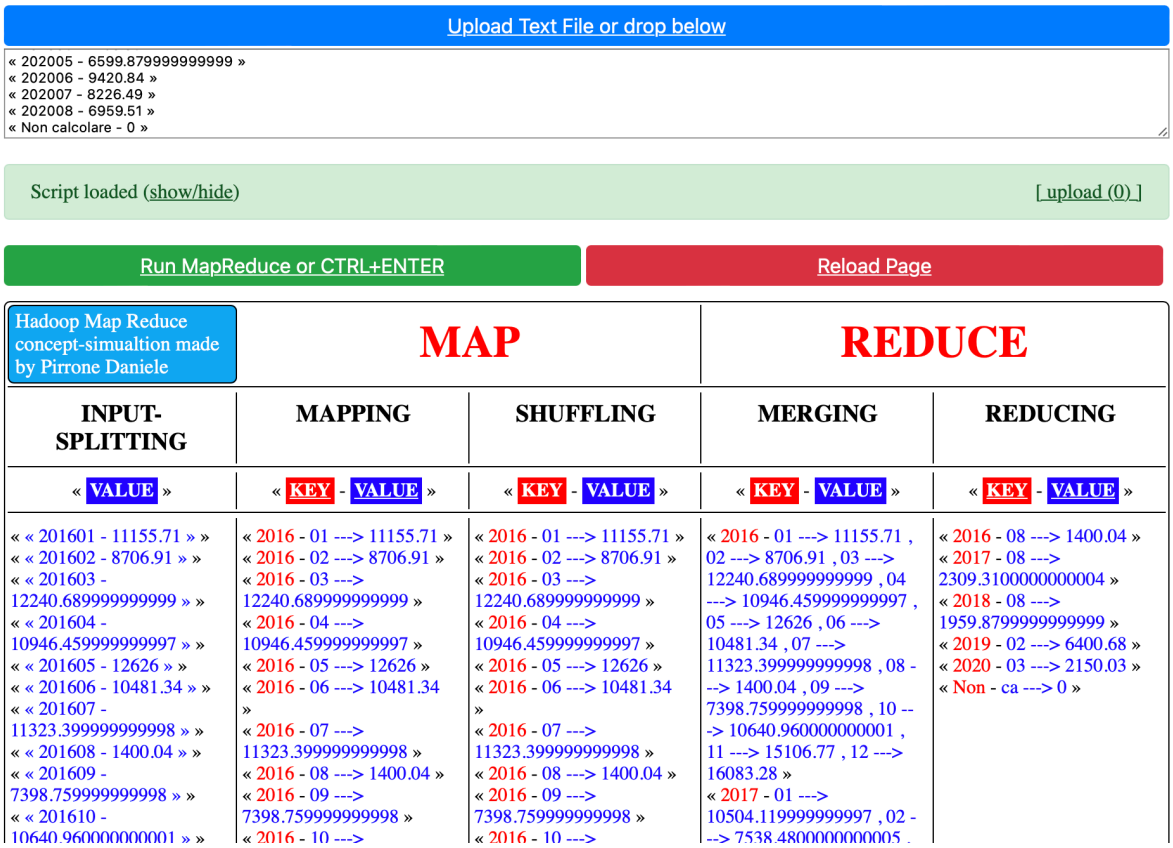

Figura 10 - Risultati mese con min vendite per anno

Utilizzo R Studio

Per l'analisi dei dati con R il processo seguito è il seguente :

- Lettura del file Ordini.csv.
- Stampa delle statistiche relative al file letto.
- Vengono inseriti i nomi alle diverse colonne del dataframe.
- Modifica del dataframe in input con filtro solo dei recordo dove il TIPODOCUMENTO è FATTURA o RICEVUTA.
- Conversione della colonna DATA nel formato corretto.
- Viene aggiunta la colonna ANNOMESE al dataset.
- Viene calcolata la media di ogni mese per ogni anno.
- Stampa del grafico relativo alla media di ogni mese per ogni anno.
- Viene calcolata la varianza di ogni mese per ogni anno.
- Stampa del grafico relativo alla varianza di ogni mese per ogni anno.
- Calcolo della somma di vendite di ogni mese per ogni anno.
- Calcolo dei mesi di ogni anno con il valore di vendita più alto.
- Stampa del grafico relativo ai mesi di ogni anno con il valore di vendita più alto.
- Calcolo dei mesi di ogni anno con il valore di vendita più basso.
- Stampa del grafico relativo ai mesi di ogni anno con il valore di vendita più basso.

Per quanto riguarda i valori medi ed i dati relativi alle vendite con importi maggiori o minori i risultati sono stati praticamente identici tra il Tool e R (se non per qualche arrotondamento).

Tuttavia, i dati relativi alla varianza, calcolati inizialmente con il Tool e successivamente con R, sono risultati diversi. Questa discrepanza è dovuta alla differenza nell'applicazione della funzione di varianza.

In R, per il calcolo della varianza, si divide la somma dei quadrati delle differenze per il numero di elementi meno uno. Nell'implementazione con il Tool, invece, la somma dei quadrati delle differenze è stata divisa per il numero di elementi.

Per ottenere risultati consistenti è possibile modificare il calcolo della varianza nel Tool (andando a dividere il numero di elementi meno uno) in modo che corrisponda alla formula utilizzata in R.

	ANNOMESE	MEDIA
1	2016-01	180.58869
2	2016-02	131.92288
3	2016-03	222.55800
4	2016-04	182.44100
5	2016-05	200.41270
6	2016-06	143.58000
7	2016-07	185.62951
8	2016-08	66.66857
9	2016-09	119.33484
10	2016-10	174.44197
11	2016-11	256.04695
12	2016-12	240.04896
13	2017-01	194.52074
14	2017-02	137.06327
15	2017-03	174.72517
16	2017-04	274.46844
17	2017-05	157.31725
18	2017-06	370.17587
19	2017-07	274.78467

Figura 11 - Media Risultati di R

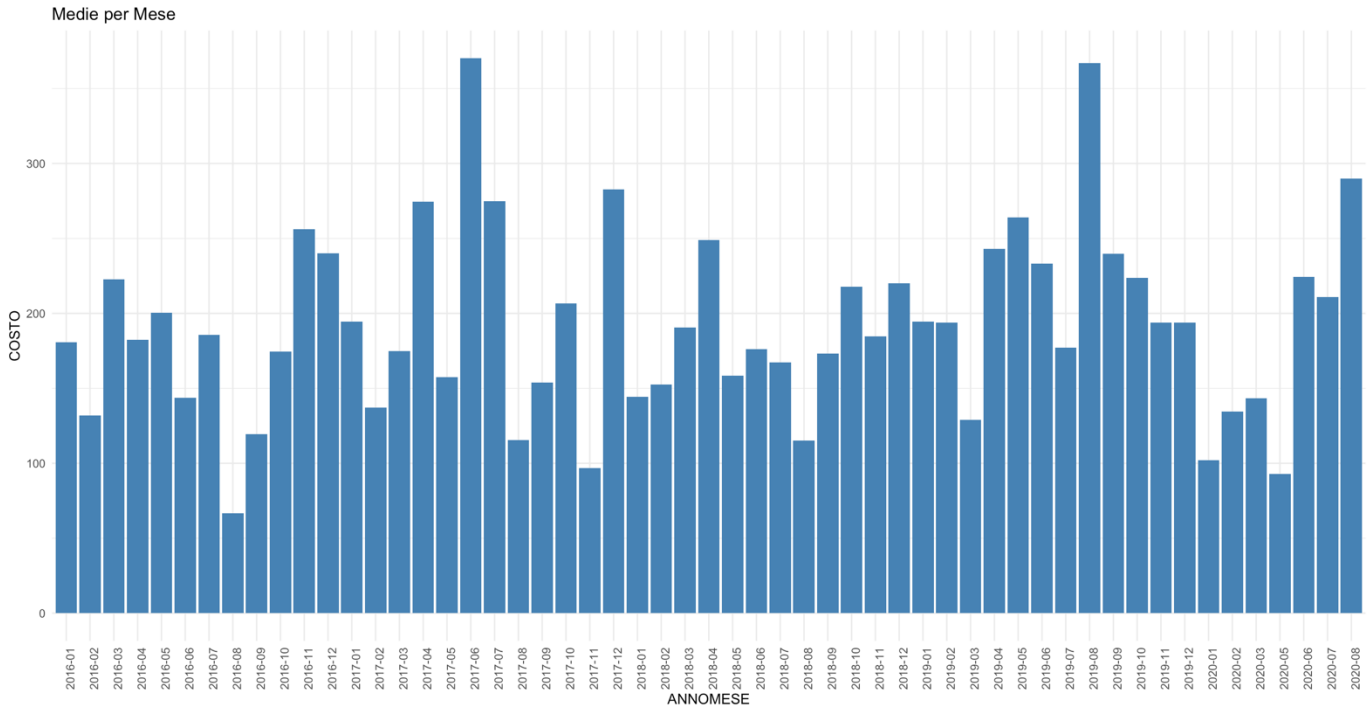

Figura 12 - Grafico medie per mese

	ANNOMESE	VARIANZA
1	2016-01	112794.955
2	2016-02	38351.044
3	2016-03	295872.781
4	2016-04	191570.685
5	2016-05	196337.279
6	2016-06	69653.870
7	2016-07	160827.843
8	2016-08	5731.737
9	2016-09	35471.358
10	2016-10	152986.941
11	2016-11	390485.050
12	2016-12	255096.324
13	2017-01	181979.971

Figura 13 – Varianza Risultati in R

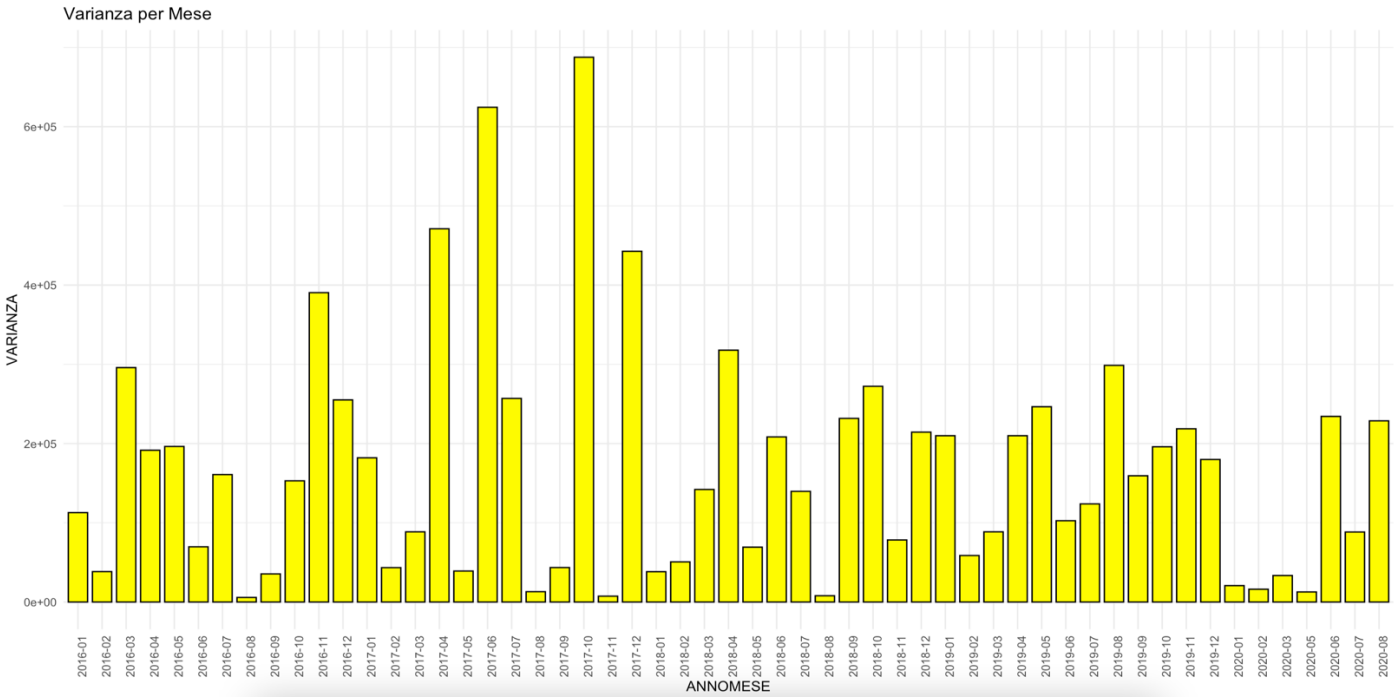

Figura 14 - Grafico Varianze per mese

	ANNOMESE	SOMMA
1	2016-12	16083.28
2	2017-06	17028.09
3	2018-04	10701.99
4	2019-10	12520.92
5	2020-06	9420.84

Figura 15 - Max vendite mese per anno

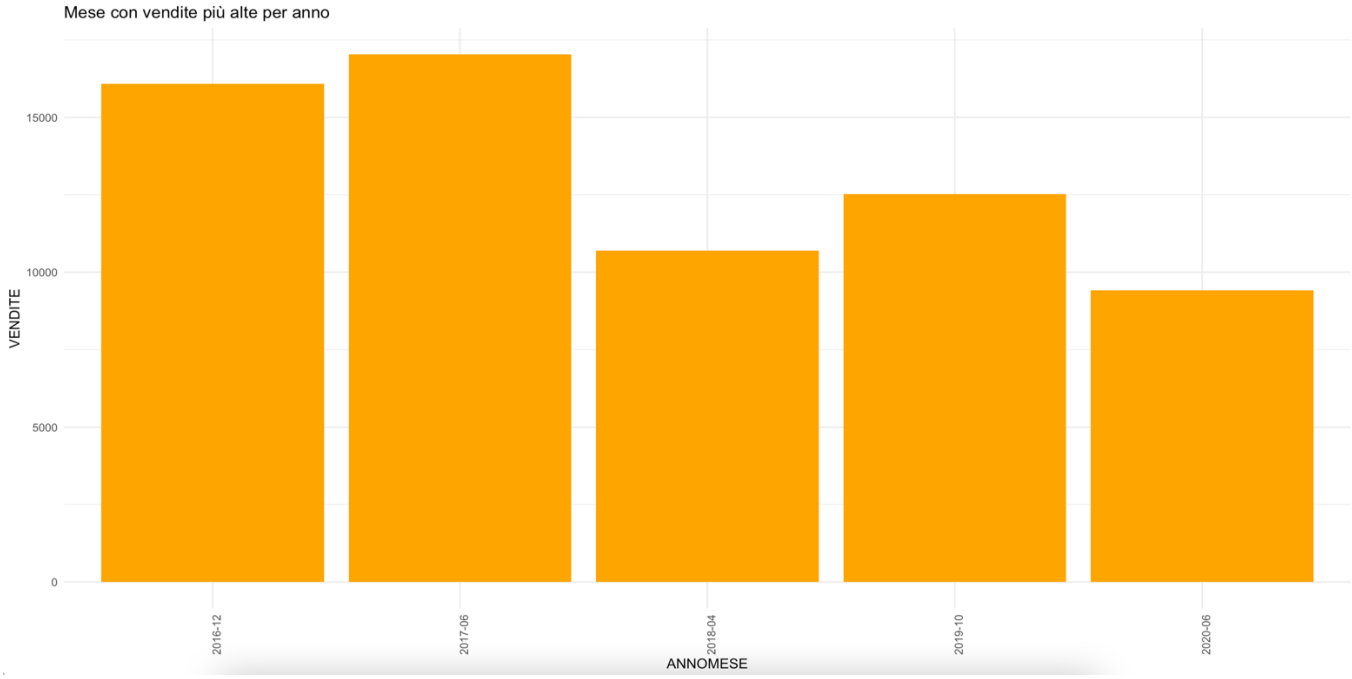

Figura 16 - Grafico max vendite mese per anno

	ANNOMESE	SOMMA
1	2016-08	1400.04
2	2017-08	2309.31
3	2018-08	1959.88
4	2019-02	6400.68
5	2020-03	2150.03

Figura 17 - min vendite mese per anno

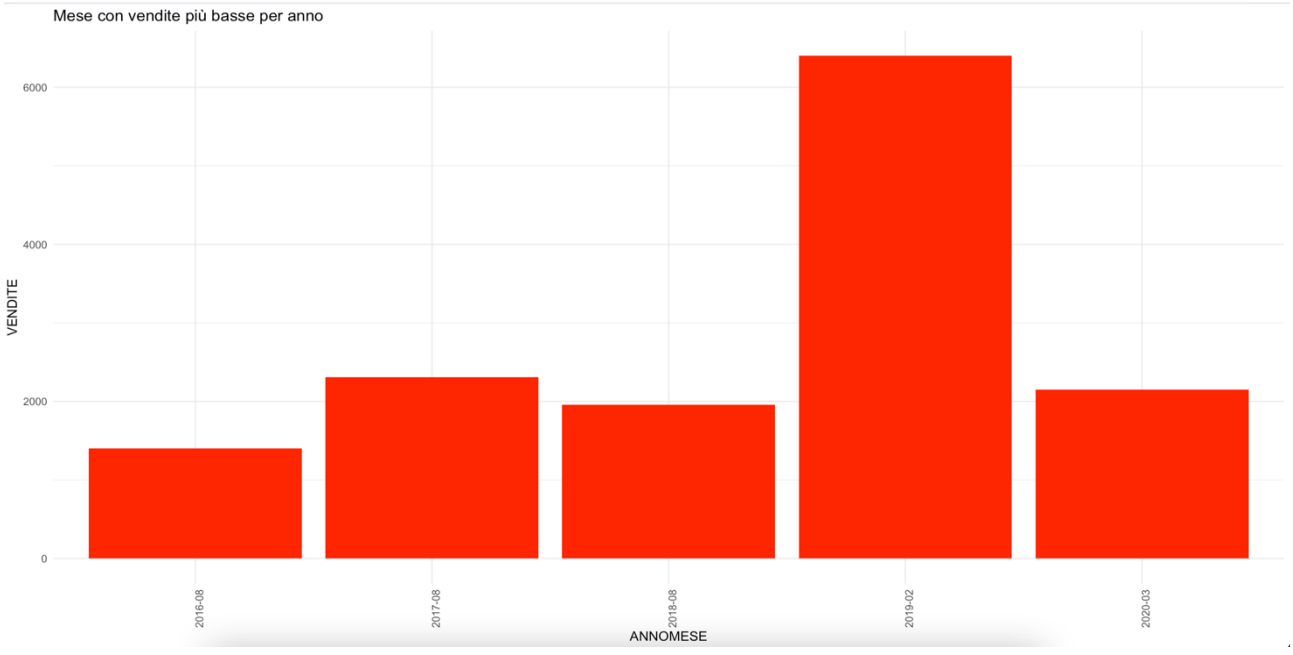

Figura 18 - Grafico min vendite mese per anno

Conclusioni

R è un linguaggio molto potente e semplice da utilizzare per l'estrazione di informazioni statistiche. Grazie a R, è stato possibile ottenere una vasta gamma di dati e visualizzarli in modo intuitivo tramite poche righe di codice. La sua vasta gamma di pacchetti e funzionalità statistiche ha reso il processo di analisi dei dati molto efficiente.
L'utilizzo del Tool fornito dal professore Daniele Pirrone ha permesso l'applicazione dell'algoritmo di MAP-REDUCE a piccoli data frame. Grazie al Tool, è stato possibile comprendere e applicare le funzioni di mapping e reducing, portando a termine con successo l'esercizio.
L'interfaccia intuitiva e le funzionalità fornite dal Tool hanno semplificato notevolmente il processo di elaborazione dei dati e l'ottenimento dei risultati desiderati.

Appendice

Tool

Calcolo Media

```
var S = ["|", ","];
function jobInputSplit(input_str) {
  return input_str.split("\n")
}

function jobMap(V_In_Map) {
  return V_In_Map.map(function(item) {
    var tipoOrdine = item.split(",")[0]
    var data = item.split(",")[1]
    var costo = item.split(",")[2]
    if (tipoOrdine === "FATTURA" || tipoOrdine === "RICEVUTA") {
      K_Out_Map = data.slice(0, 6)
      V_Out_Map = costo
    } else {
      K_Out_Map = "Non calcolare"
      V_Out_Map = 0
    }
    return keyVal(K_Out_Map, V_Out_Map)
  });
}

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]);
    let i = 1
    var Reduce = V_In_Reduce.reduce(function(accumulator, item) {
      if (K_In_Reduce !== "Non calcolare") {
        i++
        return parseFloat(accumulator) + parseFloat(item)
      } else {
        return parseFloat(accumulator) + parseFloat(item)
      }
    });
    K_Out_Reduce = K_In_Reduce
    V_Out_Reduce = Reduce / i
    return keyVal(K_Out_Reduce, V_Out_Reduce)
  });
}
```

Calcolo Varianza

```
var S = ["|", ","];

function jobInputSplit(input_str) {
  return input_str.split("\n")
}

function jobMap(V_In_Map) {
  return V_In_Map.map(function(item) {
    var tipoOrdine = item.split(",")[0]
    var data = item.split(",")[1]
    var costo = item.split(",")[2]
    if (tipoOrdine === "FATTURA" || tipoOrdine === "RICEVUTA") {
      K_Out_Map = data.slice(0, 6)
      V_Out_Map = costo
    } else {
      K_Out_Map = "Non calcolare"
      V_Out_Map = 0
    }
    return keyVal(K_Out_Map, V_Out_Map)
  });
}

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]);
    let i = 1
    var Reduce = V_In_Reduce.reduce(function(accumulator, item) {
      if (K_In_Reduce !== "Non calcolare") {
        i++
        return parseFloat(accumulator) + parseFloat(item)
      } else {
        return parseFloat(accumulator) + parseFloat(item)
      }
    });
    var media = parseFloat(Reduce) / parseFloat(i)
    var Somma_Scarti_Quadratici = V_In_Reduce.reduce(function(accumulator, item)
{
      return parseFloat(accumulator) + Math.pow(parseFloat(media) -
parseFloat(item), 2)
    }, 0)
    K_Out_Reduce = K_In_Reduce
    V_Out_Reduce = parseFloat(Somma_Scarti_Quadratici) / i
    return keyVal(K_Out_Reduce, V_Out_Reduce)
  });
}
```

Calcolo Somma

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

function jobInputSplit(input_str){
 // input_str => document.getElementById('input_text_area').value;
    return input_str.split("\n")
}

function jobMap(V_In_Map){
 return V_In_Map.map(function(item){
 var tipoOrdine = item.split(",")[0]
    var data = item.split(",")[1]
    var costo = item.split(",")[2]
    if (tipoOrdine === "FATTURA" || tipoOrdine === "RICEVUTA") {
      K_Out_Map = data.slice(0, 6)
      V_Out_Map = costo
    } else {
      K_Out_Map = "Non calcolare"
      V_Out_Map = 0
    }
    return keyVal(K_Out_Map, V_Out_Map)
});
 }

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]);
    let i = 1
    var Reduce = V_In_Reduce.reduce(function(accumulator, item) {
      if (K_In_Reduce !== "Non calcolare") {
        i++
        return parseFloat(accumulator) + parseFloat(item)
      } else {
        return parseFloat(accumulator) + parseFloat(item)
      }
    });

    K_Out_Reduce = K_In_Reduce

    V_Out_Reduce = Reduce

    return keyVal(K_Out_Reduce, V_Out_Reduce)

});

}
```

Calcolo mese con valore di vendita più alto per ogni anno

```
var S = ["|", ","];

function jobInputSplit(input_str) {
    // input_str => document.getElementById('input_text_area').value;
    return input_str.split("\n")
}

function jobMap(V_In_Map) {
    return V_In_Map.map(function(item) {
        var nuovaStringa = item.replace(/«|»/g, '').split("-")
        const data = nuovaStringa[0].trim()
        const mese = data.slice(4, 6)
        const somma = nuovaStringa[1]
        K_Out_Map = data.slice(0, 4)
        V_Out_Map = mese + " --->" + somma
        console.log(K_Out_Map)
        console.log(V_Out_Map)
        return keyVal(K_Out_Map, V_Out_Map)
    });
}

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) {
            if (parseFloat(accumulator.split(" --->")[1]) <
parseFloat(item.split(" --->")[1])) {
                return item
            } else {
                return accumulator
            }
        });
        K_Out_Reduce = K_In_Reduce
        V_Out_Reduce = Reduce
        return keyVal(K_Out_Reduce, V_Out_Reduce)
    });
}
```

Calcolo mese con valore di vendita più basso per ogni anno

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

function jobInputSplit(input_str){
 // input_str => document.getElementById('input_text_area').value;
return input_str.split("\n")


}

function jobMap(V_In_Map){
 return V_In_Map.map(function(item){
    var nuovaStringa = item.replace(/«|»/g, '').split("-")
        const data = nuovaStringa[0].trim()
        const mese = data.slice(4, 6)
        const somma = nuovaStringa[1]
        K_Out_Map = data.slice(0, 4)
        V_Out_Map = mese + " --->" + somma
        console.log(K_Out_Map)
        console.log(V_Out_Map)
        return keyVal(K_Out_Map, V_Out_Map)

});

 }

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) {
            if (parseFloat(accumulator.split(" --->")[1]) >
parseFloat(item.split(" --->")[1])) {
                return item
            } else {
                return accumulator
            }
        });
        K_Out_Reduce = K_In_Reduce
        V_Out_Reduce = Reduce
        return keyVal(K_Out_Reduce, V_Out_Reduce)

});
}
```

R

```
# Leggi il file CSV utilizzando la funzione read.csv()
1. dati <- read.csv(“Ordini.csv”)

#Stampa un riassunto statistico del dataframe.
2. summary(dati)

#Aggiungi l’header al dataframe
3. colnames(dati) <- c("TIPODOCUMENTO","DATA","COSTO")

#  Filtra il dataframe per TIPODOCUMENTO uguale a "FATTURA" o "RICEVUTA"
4. dati_filtrati <- dati[dati$TIPODOCUMENTO %in% c("FATTURA", "RICEVUTA"), ]

# Converti la colonna "DATA" nel formato corretto
5. dati_filtrati $DATA <- ymd(dati_filtrati $DATA)

# Aggiungi la colonna "ANNOMESE" al dataset
6. dati_filtrati <- dati_filtrati %>% mutate(ANNOMESE = format(DATA, "%Y-%m"))

# Calcola la media per ogni mese di ogni anno
7. media_per_mese <- dati_filtrati %>%     group_by(ANNOMESE) %>%
   summarise(MEDIA = mean(COSTO))

#Grafico per visualizzare le medie di ogni mese per ogni anno.
8. grafico <- ggplot(media_per_mese, aes(x = ANNOMESE, y = MEDIA)) +
9.      geom_bar(stat = "identity", fill = "steelblue", position = position_dodge(width = 0.8)) +
10.    labs(x = "ANNOMESE", y = "COSTO") +
11.    ggtitle("Medie per Mese") +
12.    theme_minimal() +
13.    theme(axis.text.x = element_text(angle = 90, hjust = 1),
14.          axis.text = element_text(margin = margin(t = 5, unit = "pt")))

#Calcolo varianza per ogni mese di ogni anno.
15. varianza_per_mese <- dati_filtrati %>%     group_by(ANNOMESE) %>%
    summarise(VARIANZA = var(COSTO))

#Grafico per visualizzare le varianze di ogni mese per ogni anno.
16. graficoVarianza <- ggplot(varianza_per_mese, aes(x = ANNOMESE, y = VARIANZA))
17.     geom_bar(stat = "identity", fill = "yellow", color = "black", width = 0.8) +
18.     labs(x = "ANNOMESE", y = "VARIANZA") +
19.     ggtitle("Varianza per Mese") +
20.     theme_minimal() +
21.     theme(axis.text.x = element_text(angle = 90, hjust = 1),
22.           axis.text = element_text(margin = margin(t = 5, unit = "pt")))
```

```
#Viene calcolato la somma delle vendite di ogni mese per ogni anno.
somma_per_mese <- dati_filtrati %>%
  group_by(ANNOMESE) %>%
  summarise(SOMMA = sum(COSTO))


# Trova il mese con le vendite più alte per ogni anno
mese_max <- somma_per_mese %>%
group_by(substring(ANNOMESE, 1, 4)) %>%
top_n(1, SOMMA)

# Trova il mese con le vendite più basse per ogni anno
mese_min <- somma_per_mese %>%
group_by(substring(ANNOMESE, 1, 4)) %>%
top_n(-1, SOMMA)

# Grafico max vendite mese di ogni anno
grafico_max <- ggplot(mese_max, aes(x = ANNOMESE, y = SOMMA))
  geom_bar(stat = "identity", fill = "orange", position = position_dodge(width = 0.8))
  labs(x = "ANNOMESE", y = "VENDITE")
  ggtitle("Mese con vendite più alte per anno")
  theme_minimal()
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
  axis.text = element_text(margin = margin(t = 5, unit = "pt")))


#Grafico min vendite mese di ogni anno
grafico_min <- ggplot(mese_min, aes(x = ANNOMESE, y = SOMMA))
  geom_bar(stat = "identity", fill = "orange", position = position_dodge(width = 0.8))
  labs(x = "ANNOMESE", y = "VENDITE")
  ggtitle("Mese con vendite più basse per anno")
  theme_minimal()
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
  axis.text = element_text(margin = margin(t = 5, unit = "pt")))
```