

Analisi degli effetti del Covid-19 sull'economia mondiale

Data Management

CONTE ENRICO DOCI DAVID FILIP SARA

MATRICOLE: 852679, 799647, 852864

UNIVERSITÀ DI MILANO – ‘BICOCCA’

ANNO ACCADEMICO 2019/2020

Domande di ricerca



Questo studio si propone di raccogliere informazioni di carattere finanziario riguardo le maggiori economie mondiali e confrontarne la dinamica con quella del numero di contagiati di Covid-19 nel periodo di osservazione (Gennaio – Giugno).

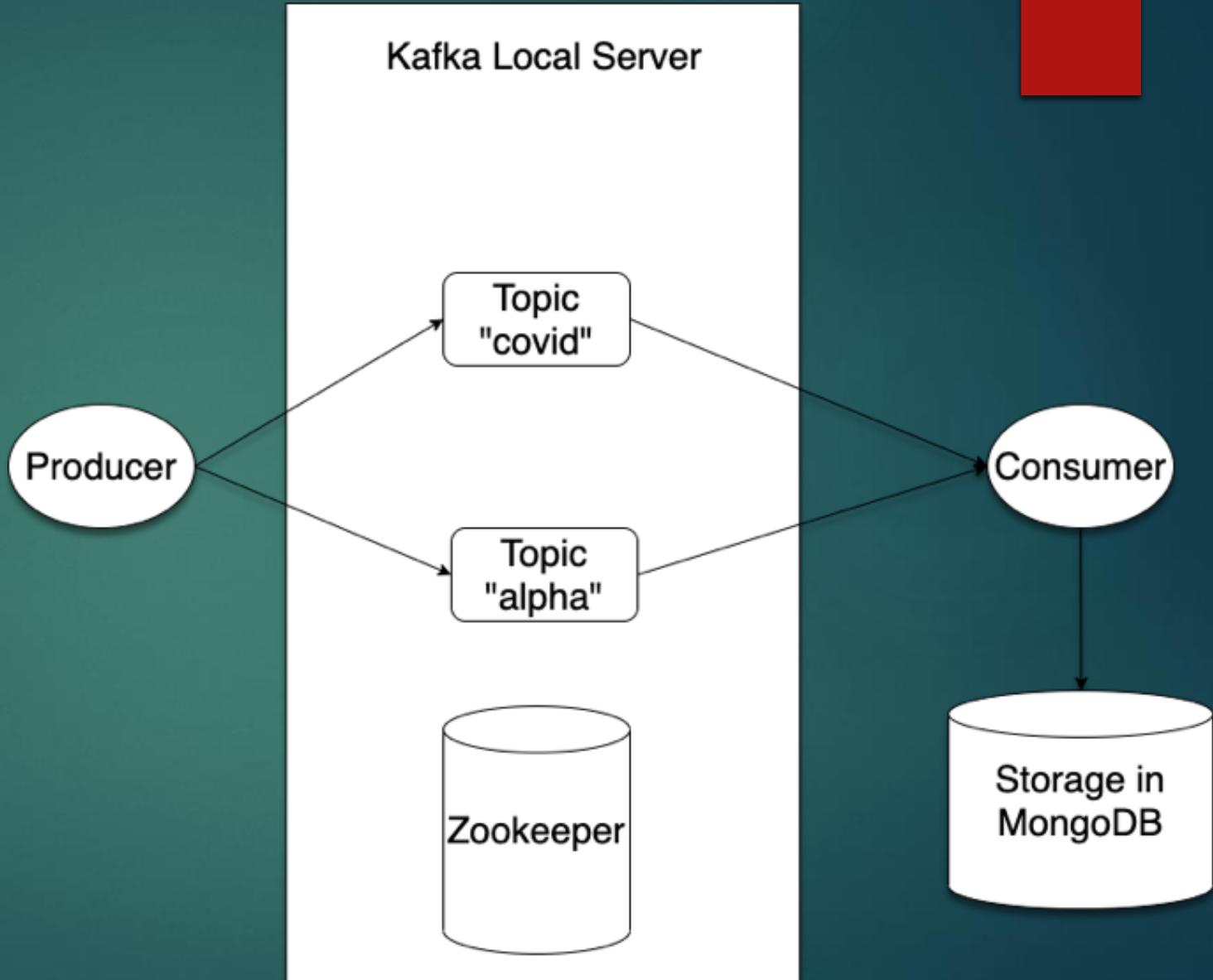


Le nostre domande di ricerca sono:

- 1) Esiste una correlazione tra le performance degli indici di borsa e il numero di contagiati?
- 2) Che comportamento hanno avuto i risparmiatori nei confronti delle Commodities?

Acquisizione e streaming dei dati

- ▶ Per rispondere alle nostre domande abbiamo implementato un'architettura di streaming Lambda. I dati vengono richiesti ad API pubbliche e importati in un database locale di MongoDB attraverso la mediazione degli applicativi Kafka.



~ — jupyter-notebook > python ...ain /usr/local/etc/kafka/zookeeper.properties ...a.Kafka /usr/local/etc/kafka/server.properties ...oker-list localhost:9092 --topic alphavantage ...:9092 --topic alphavantage --from-beginning ~ — mongo

```
Last login: Tue Aug 18 11:38:39 on ttys001
[base] enricoconte@MBP-di-Enrico ~ % kafka-server-start /usr/local/etc/kafka/server.properties
[2020-08-18 16:28:50,584] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2020-08-18 16:28:51,068] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation (org.apache.zookeeper.common.X509Util)
[2020-08-18 16:28:51,107] INFO Registered signal handlers for TERM, INT, HUP (org.apache.kafka.common.utils.LoggingSignalHandler)
[2020-08-18 16:28:51,111] INFO Starting (kafka.server.KafkaServer)
[2020-08-18 16:28:51,112] INFO Connecting to zookeeper on localhost:2181 (kafka.server.KafkaServer)
[2020-08-18 16:28:51,130] INFO [ZooKeeperClient Kafka server] Initializing a new session to localhost:2181. (kafka.zookeeper.ZooKeeperClient)
[2020-08-18 16:28:51,136] INFO Client environment:zookeeper.version=3.5.7-f0fd52973d373ffd9c86b81d99842dc2c7f660e, built on 02/10/2020 11:30 GMT (org.apache.zookeeper.ZooKeeper)
[2020-08-18 16:28:51,137] INFO Client environment:host.name=mbp-di-enrico.station (org.apache.zookeeper.ZooKeeper)
[2020-08-18 16:28:51,137] INFO Client environment:java.version=1.8.0_252 (org.apache.zookeeper.ZooKeeper)
[2020-08-18 16:28:51,137] INFO Client environment:java.vendor=AdoptOpenJDK (org.apache.zookeeper.ZooKeeper)
[2020-08-18 16:28:51,137] INFO Client environment:java.home=/Library/Java/JavaVirtualMachines/adoptopenjdk-8.jdk/Contents/Home/jre (org.apache.zookeeper.ZooKeeper)
[2020-08-18 16:28:51,137] INFO Client environment:java.class.path=/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/activation-1.1.1.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/aopalliance-repackaged-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/commons-lang3-3.8.1.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/connect-api-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/connect-basic-auth-extension-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/connect-file-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/connect-json-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/connect-mirror-client-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/connect-runtime-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/connect-transforms-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/hk2-locator-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/hk2-utils-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jackson-core-2.10.2.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jackson-dataformat-csv-2.10.2.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jackson-dataformat-xml-2.10.2.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jackson-databind-2.10.2.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jackson-dataformat-xml-2.10.2.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jackson-jaxrs-base-2.10.2.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jackson-jaxrs-json-provider-2.10.2.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jackson-module-paranamer-2.10.2.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jackson-module-servlet-2.10.2.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jakarta.activation-api-1.2.1.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jakarta.annotation-api-1.3.4.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jakarta.inject-api-2.3.2.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/javassist-3.22.0-CR2.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/javax.servlet-api-3.1.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jersey-common-2.28.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jersey-container-servlet-2.28.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jersey-media-jaxb-2.28.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jetty-client-9.4.24.v20191120.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jetty-continuation-9.4.24.v20191120.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jetty-security-9.4.24.v20191120.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jetty-servlets-9.4.24.v20191120.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jetty-servlet-9.4.24.v20191120.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/kafka-clients-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/kafka-streams-examples-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/kafka-streams-test-utils-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/kafka-tools-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/ksqldb-client-3.26.GA.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/ksqldb-server-2.5.0/libexec/bin/../libs/jersey-client-2.28.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jersey-common-2.28.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jersey-container-servlet-2.28.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jersey-media-jaxb-2.28.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jetty-client-9.4.24.v20191120.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jetty-continuation-9.4.24.v20191120.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jetty-security-9.4.24.v20191120.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jetty-servlets-9.4.24.v20191120.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jetty-servlet-9.4.24.v20191120.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/jopt-simple-5.0.4.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/kafka-clients-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/kafka-log4j-appender-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/kafka-streams-test-utils-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/../libs/kafka-tools-2.5.0.jar:/usr/local/Cellar/kafka/2.5.0/libexec/bin/..
```

Il primo passo è stata l'inizializzazione degli strumenti Kafka. Innanzitutto abbiamo avviato un server locale di Kafka, un Kafka Producer e una Consumer, così da poter automatizzare il processo di raccolta.

Successivamente abbiamo inizializzato un'istanza di Kafka Zookeeper, che semplifica lo scambio di informazioni tra le parti coinvolte.

- ▶ Dopodiché, abbiamo connesso il Consumer Kafka a MongoDB tramite la libreria python PyMongo, che permette di copiare il contenuto di un particolare topic Kafka all'interno di una Collection di MongoDB. L'operazione di storage è stata svolta con frequenza giornaliera, in quanto i dati vengono aggiornati giornalmente dalle rispettive fonti.
- ▶ In particolare, abbiamo creato due diverse Collection all'interno di MongoDB: una per i dati relativi al contagio chiamata 'covid', importati dall'API messa a disposizione dalla John Hopkins University, e l'altra chiamata come il sito che fornisce l'API per accedere ai dati finanziari, ovvero 'alphavantage'.

```
#import dei dati su mongo
from pymongo import MongoClient
import pymongo
client = MongoClient()
db = client.dataman
alphavantage = db.alphavantage

records_ftse = data_alpha_ftse.to_dict(orient = 'records')
result = db.alphavantage.insert_many(records_ftse)

records_dax = data_alpha_dax.to_dict(orient = 'records')
result = db.alphavantage.insert_many(records_dax)

records_nikkei = data_alpha_nikkei.to_dict(orient = 'records')
result = db.alphavantage.insert_many(records_nikkei)

records_nya = data_alpha_nya.to_dict(orient = 'records')
result = db.alphavantage.insert_many(records_nya)

records_bsesn = data_alpha_bsesn.to_dict(orient = 'records')
result = db.alphavantage.insert_many(records_bsesn)

records_axjo = data_alpha_axjo.to_dict(orient = 'records')
result = db.alphavantage.insert_many(records_axjo)

records_gsptse = data_alpha_gsptse.to_dict(orient = 'records')
result = db.alphavantage.insert_many(records_gsptse)

records_ibex = data_alpha_ibex.to_dict(orient = 'records')
result = db.alphavantage.insert_many(records_ibex)

records_cac = data_alpha_cac.to_dict(orient = 'records')
result = db.alphavantage.insert_many(records_cac)

records_merv = data_alpha_merv.to_dict(orient = 'records')
result = db.alphavantage.insert_many(records_merv)

print('Data Correctly sent to MongoDB')
print('Entering resting phase...')
time.sleep(20)
print('Repeating Loop...')
```

```

28146304, "Cum_Rate_BRENT" : -50.62765248200978 }
{ "_id" : ObjectId("5f54f147bd22e4dae26918f9"), "Date" : ISODate("2020-05-06T00:00:00Z"), "Close_WTI" : 23.989999771118164, "Rate_WTI" : -2.3208457134318894, "Close_BRENT" : 31, "Rate_BRENT" : -4.143476024321114, "Cum_Rate_WTI" : -460.11226852806226, "Cum_Rate_BRENT" : -54.771128506330896 }
{ "_id" : ObjectId("5f54f147bd22e4dae26918fa"), "Date" : ISODate("2020-05-07T00:00:00Z"), "Close_WTI" : 23.549999237060547, "Rate_WTI" : -1.8340997843082054, "Close_BRENT" : 29.479999542236328, "Rate_BRENT" : -4.9032272831086186, "Cum_Rate_WTI" : -461.94636831237045, "Cum_Rate_BRENT" : -59.67435578943952 }
{ "_id" : ObjectId("5f54f147bd22e4dae26918fb"), "Date" : ISODate("2020-05-08T00:00:00Z"), "Close_WTI" : 24.739999771118164, "Rate_WTI" : 5.053080987726393, "Close_BRENT" : 30.850000381469727, "Rate_BRENT" : 4.647221372139381, "Cum_Rate_WTI" : -456.89328732464406, "Cum_Rate_BRENT" : -55.02713441730013 }
{ "_id" : ObjectId("5f54f147bd22e4dae26918fc"), "Date" : ISODate("2020-05-10T00:00:00Z"), "Close_WTI" : 24.67000076293945, "Rate_WTI" : -0.2829413721577209, "Close_BRENT" : 30.700000762939453, "Rate_BRENT" : -0.4862224203419193, "Cum_Rate_WTI" : -457.1762286968018, "Cum_Rate_BRENT" : -55.51355683764205 }
{ "_id" : ObjectId("5f54f147bd22e4dae26918fd"), "Date" : ISODate("2020-05-11T00:00:00Z"), "Close_WTI" : 24.139999389648438, "Rate_WTI" : -2.1483611066333133, "Close_BRENT" : 30.1299991607666, "Rate_BRENT" : -1.8566826970927905, "Cum_Rate_WTI" : -459.3245898034351, "Cum_Rate_BRENT" : -57.37003953473484 }
{ "_id" : ObjectId("5f54f147bd22e4dae26918ff"), "Date" : ISODate("2020-05-12T00:00:00Z"), "Close_WTI" : 25.780000686645508, "Rate_WTI" : 6.793708941435704, "Close_BRENT" : 29.149999618530273, "Rate_BRENT" : -3.252570758489838, "Cum_Rate_WTI" : -452.53088086199944, "Cum_Rate_BRENT" : -60.62261029322468 }
{ "_id" : ObjectId("5f54f147bd22e4dae26918ff"), "Date" : ISODate("2020-05-13T00:00:00Z"), "Close_WTI" : 25.290000915527344, "Rate_WTI" : -1.9006972772192072, "Close_BRENT" : 29.489999771118164, "Rate_BRENT" : 1.1663813277436785, "Cum_Rate_WTI" : -454.3157813921863, "Cum_Rate_BRENT" : -59.456228965481 }
{ "_id" : ObjectId("5f54f147bd22e4dae26918fd"), "Date" : ISODate("2020-05-14T00:00:00Z"), "Close_WTI" : 27.559999465942383, "Rate_WTI" : 8.97587373759771, "Close_BRENT" : 31.510000228881836, "Rate_BRENT" : 6.849781191731356, "Cum_Rate_WTI" : -445.4557044016209, "Cum_Rate_BRENT" : -52.60644773734965 }
{ "_id" : ObjectId("5f54f147bd22e4dae2691901"), "Date" : ISODate("2020-05-15T00:00:00Z"), "Close_WTI" : 29.43000030517578, "Rate_WTI" : 6.785199112736833, "Close_BRENT" : 32.86000061035156, "Rate_BRENT" : 4.28435535278837, "Cum_Rate_WTI" : -438.6705052888404, "Cum_Rate_BRENT" : -48.32209242096128 }
{ "_id" : ObjectId("5f54f147bd22e4dae2691902"), "Date" : ISODate("2020-05-17T00:00:00Z"), "Close_WTI" : 30.260000228881836, "Rate_WTI" : 2.2000001220703125, "Close_BRENT" : 33.439998626708984, "Rate_BRENT" : 1.7650578380534503, "Cum_Rate_WTI" : -435.850254132628, "Cum_Rate_BRENT" : -46.55703458290783 }
Type "it" for more
] > it
{ "_id" : ObjectId("5f54f147bd22e4dae2691903"), "Date" : ISODate("2020-05-18T00:00:00Z"), "Close_WTI" : 31.81999969482422, "Rate_WTI" : 5.155318751298065, "Close_BRENT" : 35.68000030517578, "Rate_BRENT" : 6.698569887732223, "Cum_Rate_WTI" : -430.69493538196474, "Cum_Rate_BRENT" : -39.85846469517561 }
{ "_id" : ObjectId("5f54f147bd22e4dae2691904"), "Date" : ISODate("2020-05-19T00:00:00Z"), "Close_WTI" : 32.5, "Rate_WTI" : 2.1370217212364992, "Close_BRENT" : 34.470001220703125, "Rate_BRENT" : -3.391253010435463, "Cum_Rate_WTI" : -428.5579136607282, "Cum_Rate_BRENT" : -43.24971705611076 }
{ "_id" : ObjectId("5f54f147bd22e4dae2691905"), "Date" : ISODate("2020-05-20T00:00:00Z"), "Close_WTI" : 33.4900016784668, "Rate_WTI" : 3.046159010667067, "Close_BRENT" : 35.86000061035156, "Rate_BRENT" : 4.032490108568914, "Cum_Rate_WTI" : -425.51175465006116, "Cum_Rate_BRENT" : -39.217227557904216 }
{ "_id" : ObjectId("5f54f147bd22e4dae2691906"), "Date" : ISODate("2020-05-21T00:00:00Z"), "Close_WTI" : 33.91999816894531, "Rate_WTI" : 1.2839548191333543, "Close_BRENT" : 36.150001525878906, "Rate_BRENT" : 0.8087030412476635, "Cum_Rate_WTI" : -424.22779983092784, "Cum_Rate_BRENT" : -38.4085245557945 }
{ "_id" : ObjectId("5f54f147bd22e4dae2691907"), "Date" : ISODate("2020-05-22T00:00:00Z"), "Close_WTI" : 33.25, "Rate_WTI" : -1.975230501613205, "Close_BRENT" : 34.759991515321, "Rate_BRENT" : -1.1938657760067906, "Cum_Rate_WTI" : -426.20303038845395 }
{ "_id" : ObjectId("5f54f147bd22e4dae2691908"), "Date" : ISODate("2020-05-24T00:00:00Z"), "Close_WTI" : 32.93000030517578, "Rate_WTI" : -0.96240509715695, "Close_BRENT" : 34.759991515321, "Rate_BRENT" : -1.1938657760067906, "Cum_Rate_WTI" : -427.16543548566966, "Cum_Rate_BRENT" : -42.285657772353744 }
{ "_id" : ObjectId("5f54f147bd22e4dae2691909"), "Date" : ISODate("2020-05-26T00:00:00Z"), "Close_WTI" : 34.349998474121094, "Rate_WTI" : 4.312171745477099, "Close_BRENT" : 35.779998779296875, "Rate_BRENT" : 2.9344088234083707, "Cum_Rate_WTI" : -422.8537401926, "Cum_Rate_BRENT" : -39.35124894945376 }
{ "_id" : ObjectId("5f54f147bd22e4dae269190a"), "Date" : ISODate("2020-05-27T00:00:00Z"), "Close_WTI" : 32.810001373291016, "Rate_WTI" : -4.48, "Close_BRENT" : 34.759991515321, "Rate_BRENT" : 0.707856869313, "Cum_Rate_WTI" : -427.3365160523879, "Cum_Rate_BRENT" : -46.03095680581469 }
{ "_id" : ObjectId("5f54f147bd22e4dae269190b"), "Date" : ISODate("2020-05-28T00:00:00Z"), "Close_WTI" : 33.709999084472656, "Rate_WTI" : 2.743250177205, "Close_BRENT" : 34.759991515321, "Rate_BRENT" : 2.702005760775834, "Cum_Rate_WTI" : -424.59345700848645, "Cum_Rate_BRENT" : -41.328951045039055 }
{ "_id" : ObjectId("5f54f147bd22e4dae269190c"), "Date" : ISODate("2020-05-29T00:00:00Z"), "Close_WTI" : 35.4900016784668, "Rate_WTI" : 5.2803400840614, "Close_BRENT" : 35.33000183105479, "Rate_BRENT" : 1.9583029368181037, "Cum_Rate_WTI" : -419.3131169207451, "Cum_Rate_BRENT" : -40.27059075220955 }
{ "_id" : ObjectId("5f54f147bd22e4dae269190d"), "Date" : ISODate("2020-06-02T00:00:00Z"), "Close_WTI" : 36.810001373291016, "Rate_WTI" : 3.8651760000000004, "Close_BRENT" : 35.33000183105479, "Rate_BRENT" : 1.22919140515, "Cum_Rate_WTI" : -415.4474205334164, "Cum_Rate_BRENT" : -27.646761566815805 }
Type "it" for more
] > it
{ "_id" : ObjectId("5f54f189647610c91b9e8400"), "Country" : "Italy", "Correlations" : -0.74, "Lat" : 41.87194, "Long" : 12.56738, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8401"), "Country" : "US", "Correlations" : -0.66, "Lat" : 40, "Long" : -100, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8402"), "Country" : "Brazil", "Correlations" : -0.34, "Lat" : -14.235, "Long" : -51.9253, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8403"), "Country" : "Australia", "Correlations" : -0.53, "Lat" : -42.8821, "Long" : 147.3272, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8404"), "Country" : "Australia", "Correlations" : -0.53, "Lat" : -37.8136, "Long" : 149.9631, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8405"), "Country" : "Australia", "Correlations" : -0.53, "Lat" : -35.4735, "Long" : 149.0124, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8406"), "Country" : "Australia", "Correlations" : -0.53, "Lat" : -34.9285, "Long" : 139.6007, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8407"), "Country" : "Australia", "Correlations" : -0.53, "Lat" : -33.8688, "Long" : 151.2093, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8408"), "Country" : "Australia", "Correlations" : -0.53, "Lat" : -31.9505, "Long" : 115.8605, "Date" : "2020-06-02" }
Type "it" for more
] > it
{ "_id" : ObjectId("5f54f189647610c91b9e8409"), "Country" : "Australia", "Correlations" : -0.53, "Lat" : -27.4698, "Long" : 153.0251, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e840a"), "Country" : "Australia", "Correlations" : -0.53, "Lat" : -12.4634, "Long" : 138.8456, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e840b"), "Country" : "China", "Correlations" : -0.038, "Lat" : 19.1959, "Long" : 109.7453, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e840c"), "Country" : "China", "Correlations" : -0.038, "Lat" : 22.1667, "Long" : 113.55, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e840d"), "Country" : "China", "Correlations" : -0.038, "Lat" : 22.3, "Long" : 114.2, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e840e"), "Country" : "China", "Correlations" : -0.038, "Lat" : 23.3417, "Long" : 113.4244, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e840f"), "Country" : "China", "Correlations" : -0.038, "Lat" : 23.8298, "Long" : 108.7881, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8410"), "Country" : "China", "Correlations" : -0.038, "Lat" : 24.974, "Long" : 101.487, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8411"), "Country" : "China", "Correlations" : -0.038, "Lat" : 26.0789, "Long" : 117.9874, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8412"), "Country" : "China", "Correlations" : -0.038, "Lat" : 26.8154, "Long" : 106.8748, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8413"), "Country" : "China", "Correlations" : -0.038, "Lat" : 27.6104, "Long" : 111.7088, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8414"), "Country" : "China", "Correlations" : -0.038, "Lat" : 27.614, "Long" : 115.7221, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8415"), "Country" : "China", "Correlations" : -0.038, "Lat" : 29.1832, "Long" : 120.0934, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8416"), "Country" : "China", "Correlations" : -0.038, "Lat" : 30.0572, "Long" : 107.874, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8417"), "Country" : "China", "Correlations" : -0.038, "Lat" : 30.6171, "Long" : 102.7103, "Date" : "2020-06-02" }
{ "_id" : ObjectId("5f54f189647610c91b9e8418"), "Country" : "China", "Correlations" : -0.038, "Lat" : 30.9756, "Long" : 112.2797, "Date" : "2020-06-02" }

```

MongoDB

Ecco come si presentano i dati nelle Collections finali di MongoDB: la sintassi utilizzata è quella del dizionario {chiave : valore} tipica del formato JSON.

Pre- Processing

- ▶ All'interno degli script python utilizzati per l'import abbiamo anche effettuato il pre-processing dei dati.
- ▶ In particolare, una volta eliminati i valori NaN ,ove presenti, e gli attributi a noi inutili, abbiamo calcolato, a partire dai prezzi di chiusura giornalieri, i rendimenti cumulativi degli indici di borsa selezionati e la performance cumulativa delle Commodities scelte, ovvero Oro, Argento, petrolio Brent e petrolio WTI.
- ▶ Abbiamo infine calcolato il numero complessivo di contagiati da Covid-19 per ognuno dei 17 paesi inclusi nello studio. I suddetti paesi sono: Italia, Spagna, Francia, Germania, Regno Unito, USA, Canada, Turchia, Arabia Saudita, Russia, India, Cina, Giappone, Corea del Sud, Australia, Brasile e Argentina.

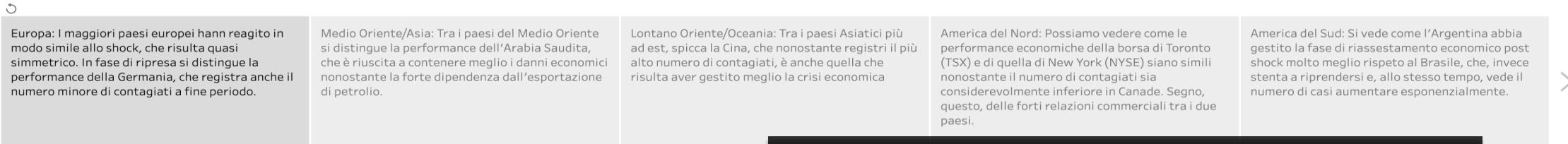


Per rendere le misure confrontabili abbiamo normalizzato tutti i rendimenti rispetto al valore registrato nel primo giorno in cui nessuna delle rilevazioni risultasse mancante, ovvero il 22 Gennaio.

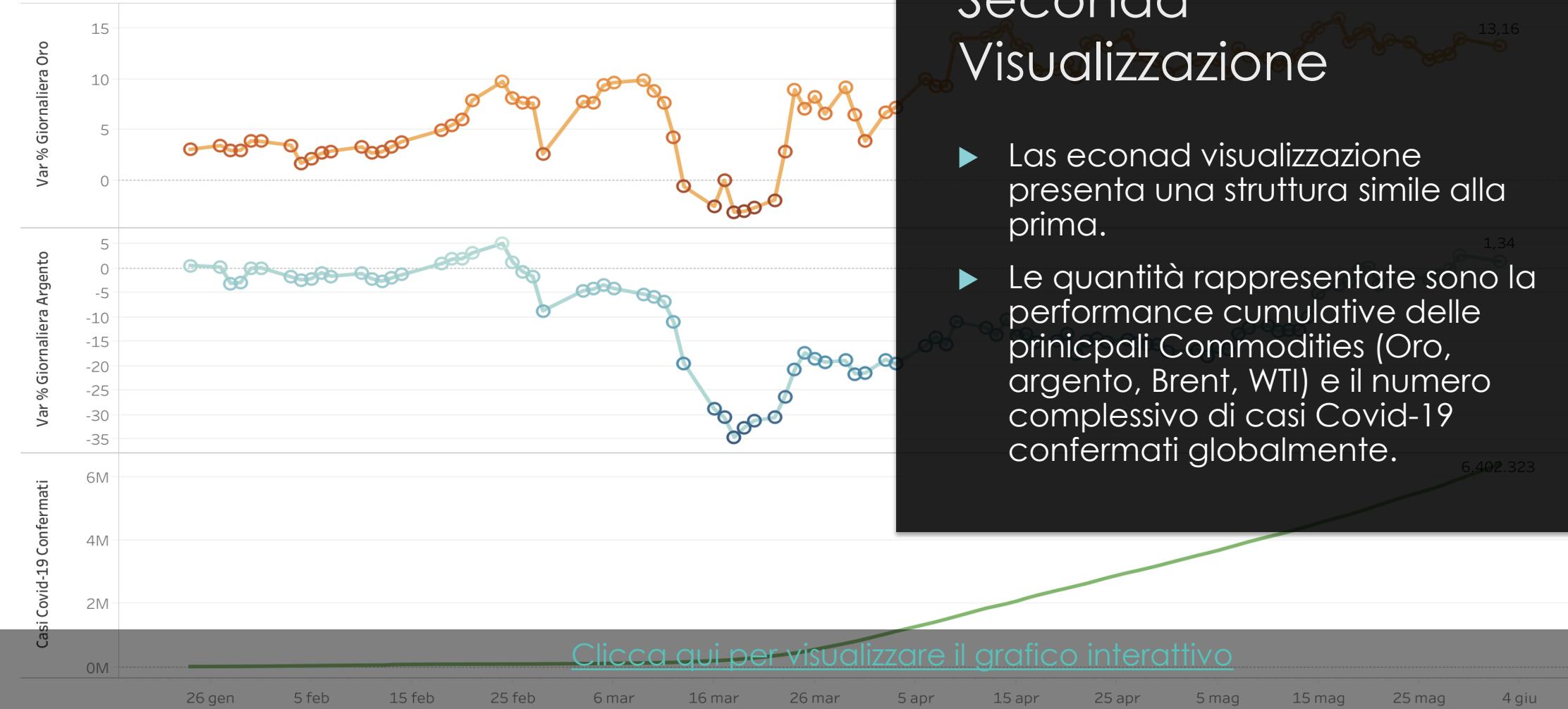
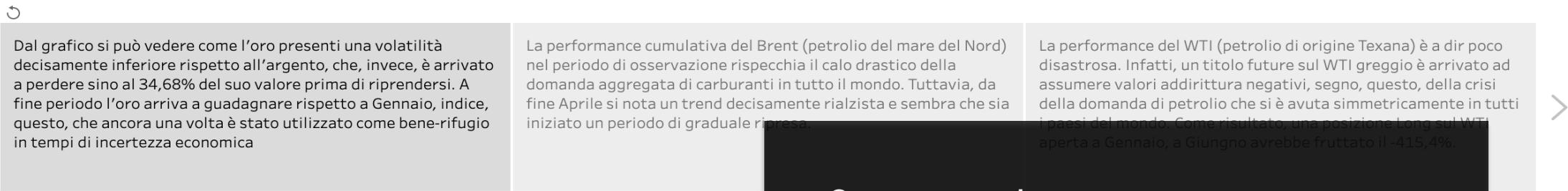


Infine, abbiamo importato i dataset finali sul software Tableau Desktop per realizzare le visualizzazioni interattive.

Performance Economica e Contagio



Performance Cumulativa Commodities



Seconda Visualizzazione

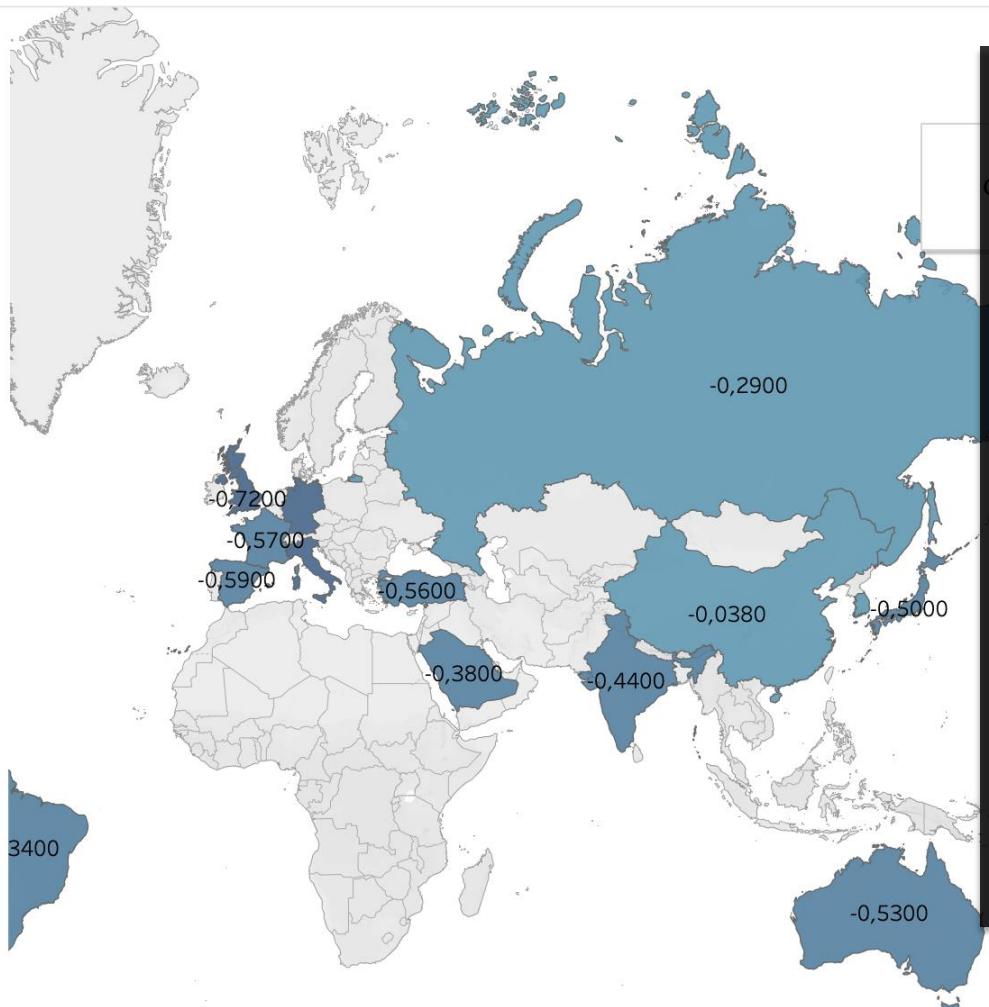
- ▶ La seconda visualizzazione presenta una struttura simile alla prima.
- ▶ Le quantità rappresentate sono la performance cumulativa delle principali Commodities (Oro, argento, Brent, WTI) e il numero complessivo di casi Covid-19 confermati globalmente.



- ▶ Per le visualizzazioni sopra esposte abbiamo utilizzato lo strumento ‘Storia’ di Tableau. Ovvero, scorrendo le didascalie sopra la grafico, il grafico stesso si aggiorna e cambiano i dati mostrati. Nel primo caso abbiamo diviso i paesi in base all’area geografica. Nel secondo caso abbiamo presentato prima la performance di Oro e Argento e dopo quella di Brent e WTI
- ▶ Nella prima visualizzazione è possibile, tramite un menù a tendina, selezionare uno dei paesi rappresentati per vederne i dati più nel dettaglio
- ▶ Nella seconda visualizzazione abbiamo implementato la possibilità di animare la serie storica per apprezzarne l’evoluzione nel tempo.

Mappa Coroletica dell'intensità della correlazione tra Performance economica e numero di Contagiati

Intensità della correla...
-1,000 1,000



Terza

Questi valori rappresentano rispettivamente, correlazione perfettamente negativa o positiva tra i fenomeni in esame. Il valore 0 indica indipendenza.

Visualizzazione

- ▶ La terza visualizzazione è una mappa coroletica.
- ▶ L'intensità del colore rappresenta l'intensità della correlazione lineare tra la performance dell'indice di borsa del paese e il numero di casi confermati dello stesso.

[Clicca qui per visualizzare il grafico interattivo](#)