SC1015 Mini-project
Team 3 (SC9)

Lee Zheng Xuan (U2120607F)
Jaren Ng Shing Yu (U2122456F)
Joel Tham Yew Hng (U2121679K)

Our Problem statement

We aim to find out the **correlation** between various **stock prices** and the **Covid-19 situation** in Singapore, allowing us to **predict** future stock prices of selected industries. We hope to use **machine learning models** and **selecting the most suitable model** to predict future industries' stock prices more accurately using Covid-19 as a variable.

Datasets

Our datasets

COVID-19 in Singapore

- Retrieve CSV file from Kaggle that dates from 23 Jan 2020 to 8 Jan 2022

Business days and public holidays

- Retrieve business days using Pandas' library
- Retrieve Singapore's public holiday from Gov.sg

Stock prices in Singapore

- Yfinance API to extract closing stock prices in Singapore's SGX Exchange
- Prices of stocks from various industries dates from 23 Jan 2020 to 8 Jan 2022

COVID-19 Dataset

- Retrieved CSV file from Kaggle
- Location: Singapore
- Data is available from 23 January 2020 to 8 January 2022

	Date	Daily Confirmed	False Positives Found	Cumulative Confirmed	Daily Discharged	Passed but not due to COVID	Cumulative Discharged	Discharged to Isolation	Still Hospitalised	Daily Deaths	...	Cumulative Individuals Vaccinated	Cumulative Individuals Vaccination Completed	Perc population completed at least one dose	Perc population completed vaccination	Sinovac vaccine doses	Cumulative individuals using Sinovac vaccine	Doses of other vaccines recognised by WHO	Cumulative individuals using other vaccines recognised by WHO	Number taken booster shots	Perc population taken booster shots
0	2020-01-23	1	NaN	1	0	0	0	0	1	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2020-01-24	2	NaN	3	0	0	0	0	3	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	2020-01-25	1	NaN	4	0	0	0	0	4	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	2020-01-26	0	NaN	4	0	0	0	0	4	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	2020-01-27	1	NaN	5	0	0	0	0	5	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
712	2022-01-04	842	0.0	281596	271	0	276936	3669	151	3	...	NaN	NaN	88%	87%	NaN	NaN	NaN	NaN	NaN	42%
713	2022-01-05	805	0.0	282401	453	0	277389	4015	155	2	...	NaN	NaN	88%	87%	NaN	NaN	NaN	NaN	NaN	43%
714	2022-01-06	813	0.0	283214	392	0	277781	4449	141	1	...	NaN	NaN	88%	87%	NaN	NaN	NaN	NaN	NaN	44%
715	2022-01-07	777	0.0	283991	338	0	278119	4892	135	2	...	NaN	NaN	89%	87%	NaN	NaN	NaN	NaN	NaN	44%
716	2022-01-08	811	0.0	284802	336	0	278455	5372	130	0	...	NaN	NaN	89%	87%	NaN	NaN	NaN	NaN	NaN	45%

717 rows × 36 columns

We are only concerned about daily confirmed cases as we can use the daily cases to compare with the daily closed of stock prices in Singapore. Thus, we are going to clean up the data to show only the daily confirmed cases.

From the dataset, we have access to different totals for Singapore. Some of these include number of deaths, number of hospitalised, and number of daily confirmed cases.

Industries

We chose the top leaders from each industry to represent the stock dataset.

1. **Pharmaceutical** (Biolidics Limited)
2. **Aviation** (Singapore Airlines)
3. **Transportation** (ComfortDelgo Group)
4. **Telecommunication** (SingTel)
5. **Information technologies** (Creative Technologies)

Stocks Dataset

- Dataset is retrieved from YFinance using API
- According to desired stock, data of closing price is collected individually
- All stock data is updated daily

```
sia = yf.Ticker('C6L.SI')
sia_historical = sia.history(start="2020-01-23", end="2022-01-08", interval="1d")
sia_historical.head()
```

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2020-01-23	8.92	8.94	8.84	8.85	1891800	0	0
2020-01-24	8.84	8.84	8.73	8.82	1471300	0	0
2020-01-28	8.58	8.59	8.48	8.56	3781700	0	0
2020-01-29	8.61	8.63	8.54	8.57	2236300	0	0
2020-01-30	8.57	8.59	8.50	8.54	2097100	0	0

Cleaning of Data

COVID-19 Dataset

- Dataset is partitioned from **1 April 2020** to **8 January 2022** for consistency
- Weekends and Public holidays are removed
  - While data is available on weekends and public holidays, the stock market is closed.
- Used Daily Confirmed data
  - Understanding the data, “Daily Confirmed” shows the number of cases per day
  - The number of cases per day is more relevant for finding the relationship with stock price as stock prices change daily too

Business Days and Public Holidays

- Retrieved Business Days using Pandas' library
- Retrieved Singapore's Public Holidays dataset from Gov.sg

```
publichols = pd.read_csv('PublicHols2020-2022.csv')
publicholsdates = pd.DataFrame(publichols['Date'])
```

```
publicholsdates.Date = pd.to_datetime(publicholsdates.Date)
publicholsdates.head()
```

	Date
0	2020-01-01
1	2020-01-27
2	2020-04-10
3	2020-05-01
4	2020-05-07

```
covid = covid[(covid['Date'] >= '2020-4-01') & (covid['Date'] <= '2022-1-08')]
```

Extracting Covid data from specified dates

```
isBusinessDay = BDay().onOffset
match_series = pd.to_datetime(covid['Date']).map(isBusinessDay)
covid = covid[match_series]
```

Removing Weekends from dataset

```
dates = []
for i in publicholsdates.Date:
    for j in covid.Date:
        if i==j:
            covid = covid[covid.Date != i]
            break
```

Removing Singapore public holidays from dataset

	Date	Daily Confirmed
69	2020-04-01	74
70	2020-04-02	49
71	2020-04-03	65
72	2020-04-04	75
73	2020-04-05	120
...	...	...
712	2022-01-04	842
713	2022-01-05	805
714	2022-01-06	813
715	2022-01-07	777
716	2022-01-08	811

648 rows × 2 columns

Stocks Dataset

- Dataset retrieved from **1 April 2020** to **8 January 2022** to be consistent with Covid Dataset
- Used daily close price
  - See how daily cases affect the price of stocks at the end of the market day

Data Cleaning for Machine Learning

- Removed **70 days** as cases in Singapore mainly started in April
- First case was in January. Daily cases remained under 50 till April.
- Since April, the number of daily cases has been above 100, thus number of cases from January is small and insignificant comparatively.

Exploratory Analysis

Correlation Coefficient

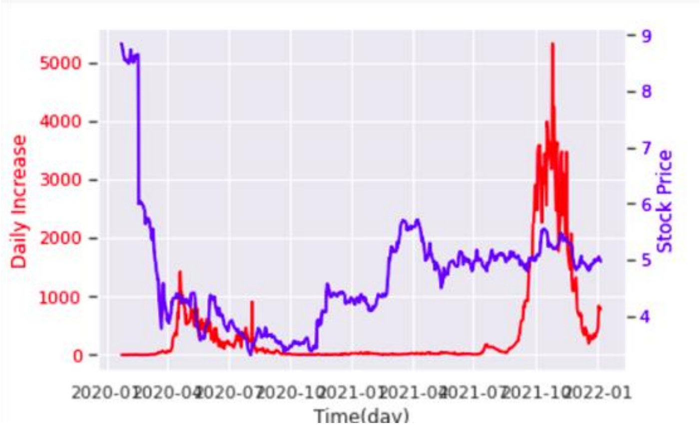
We plotted bi-variate joint plot and heat map for each company in the industry.

```
[ ] # Create a joint dataframe by concatenating the two variables
    covidsia = pd.concat([covid, sia_close], axis = 1).reindex(covid.index)
    covidsia.info
    sb.jointplot(data = covidsia, x = "Close", y = "dailyConfirmed", height = 12)
    # covidsia
```

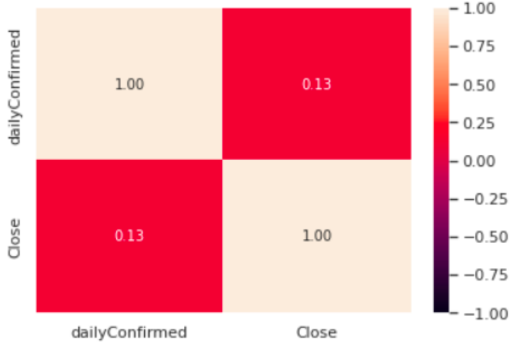
```
[ ] sb.heatmap(covidbiolidics.corr(), vmin = -1, vmax = 1, annot = True, fmt=".2f")
```

- We noticed for each industry, the coefficient varies from high to low.
- We categorize the data into 3 categories:
  - Strong correlation
  - Moderate Correlation
  - Weak/No Correlation

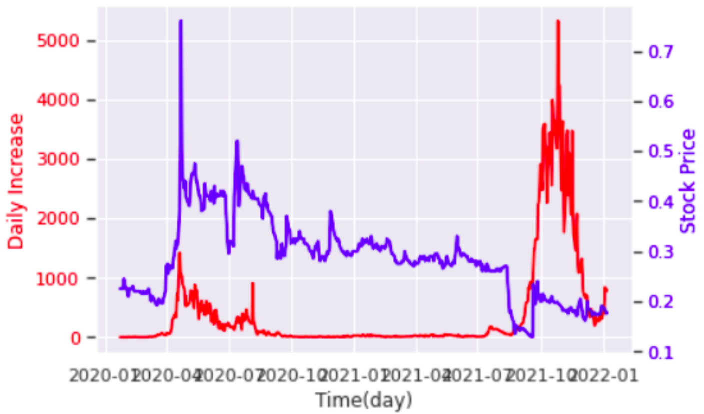
Singapore Airlines (Aviation)




```
[ ]  sb.heatmap(covidsia.corr(), vmin = -1, vmax = 1, annot = True, fmt=".2f")

<matplotlib.axes._subplots.AxesSubplot at 0x7f3381845e90>
```



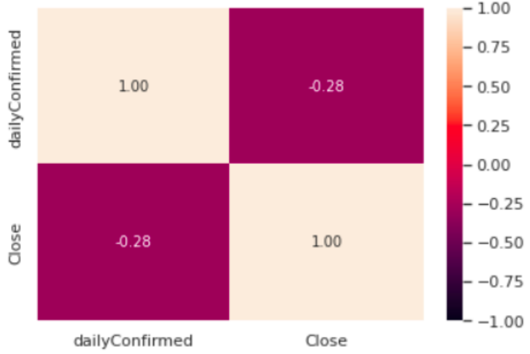
Biolidics Limited (Pharmaceutical)



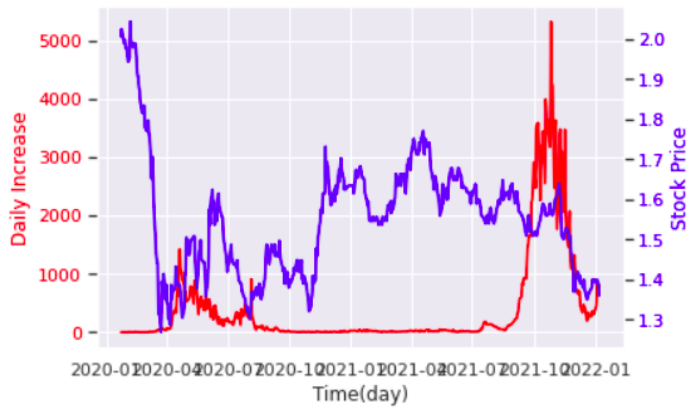
<Figure size 75x75 with 0 Axes>

```
sb.heatmap(covidbiolidics.corr(), vmin = -1, vmax = 1, annot = True, fmt=".2f")
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f337f0aca50>



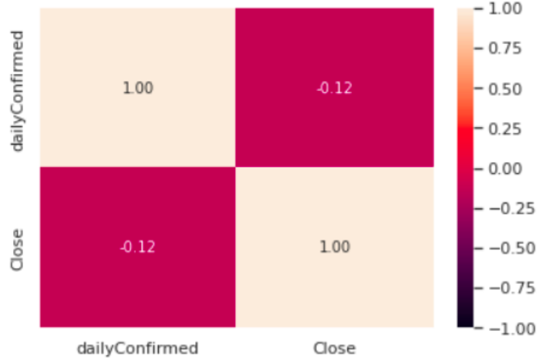
ComfortDelgro Corp (Transportation)



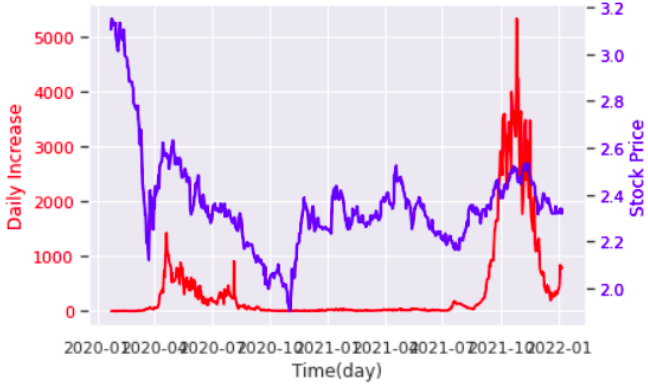

```
<Figure size 75x75 with 0 Axes>
```

```
sb.heatmap(covidcomfortdel.corr(), vmin = -1, vmax = 1, annot = True, fmt=".2f")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f337ea6f310>
```



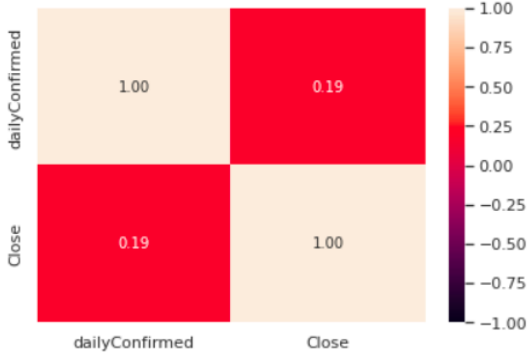
Singtel (Telecommunication)



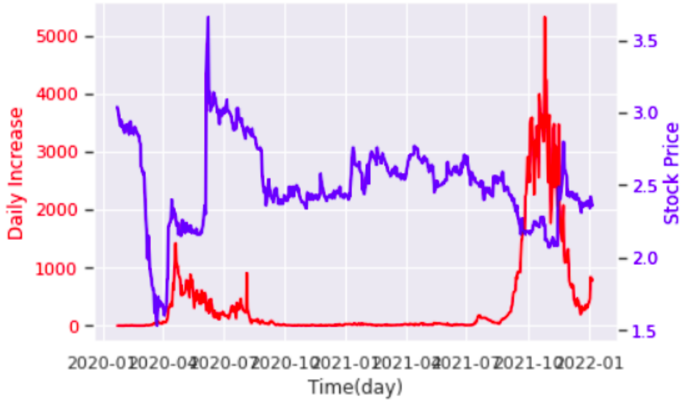

```
<Figure size 75x75 with 0 Axes>
```

```
sb.heatmap(covidsingtel.corr(), vmin = -1, vmax = 1, annot = True, fmt=".2f")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f337e75c710>
```



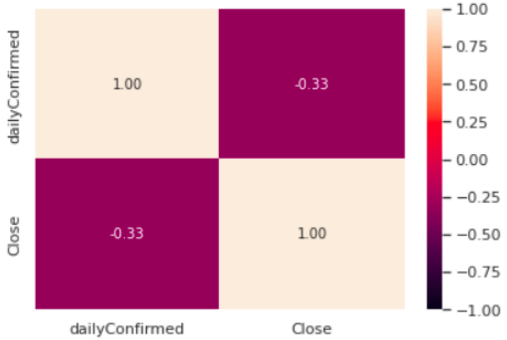
Creative Technologies (Information Technology)



<Figure size 75x75 with 0 Axes>

```
sb.heatmap(covidcreative.corr(), vmin = -1, vmax = 1, annot = True, fmt=".2f")
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f337e210d90>



Machine Learning

Methodology

1. Partitioning the dataset
   - 75% Train Dataset
   - 25% Test Dataset
   - Data is not randomised as it is time series and the order of occurrence of data points matters

```
[ ] covid_ml = pd.DataFrame(covid["dailyConfirmed"])
    sia_ml = pd.DataFrame(sia_close["Close"])

    # Split the Dataset into Train and Test, 75% 25% respectively
    X_train = covid_ml.iloc[124:495,:]
    X_test = covid_ml.iloc[:124,:]
    y_train = sia_ml.iloc[124:495,:]
    y_test = sia_ml.iloc[:124,:]
```

Methodology

2. Train dataset using Linear Regression,

   Hypothesized Linear Model:

   Stock Price = Coefficient x Daily Covid Cases + Intercept (y=mx+c)

```
# Linear Regression using Train Data
linreg = LinearRegression()         # create the linear regression object
linreg.fit(X_train, y_train)        # train the linear regression model
```

Methodology

3. Predict Total Values corresponding to Daily Covid Cases using Linear Regression

```
# Coefficients of the Linear Regression line
print('Intercept of Regression \t: b = ', linreg.intercept_)
print('Coefficients of Regression \t: a = ', linreg.coef_)
print()

y_train_pred = linreg.predict(X_train)
y_test_pred = linreg.predict(X_test)
```

Methodology

4. Checking for goodness of fit:

   - Explained Variance
   - Mean Squared Error

```
# Check the Goodness of Fit (on Test Data)
print("Goodness of Fit of Model \tTest Dataset")
print("Explained Variance (R^2) \t:", linreg.score(X_test, y_test))
print("Mean Squared Error (MSE) \t:", mean_squared_error(y_test, y_test_pred))
print()
```

Machine Learning Findings

Mean Squared Error (MSE)

Through our analysis, the MSE for all the **test data set** in all models are **way higher** than the one in the **train dataset** as shown in the next 5 slides.

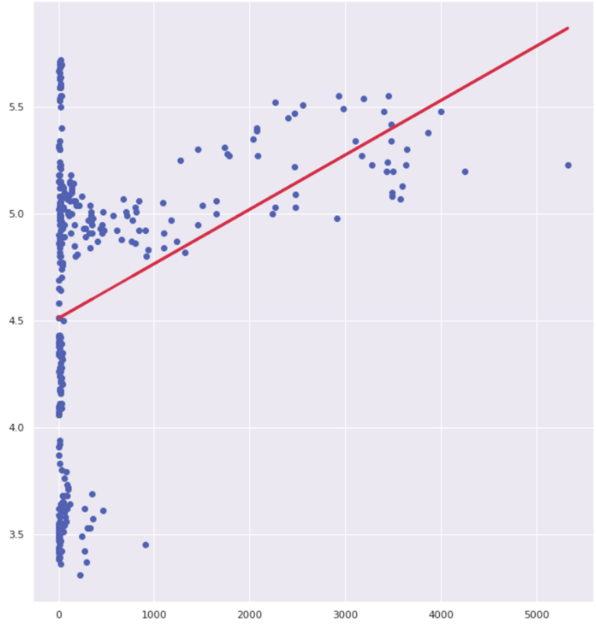
Mean Squared Error (MSE) for SIA

Through our analysis, the MSE for all the **test data set** in all models are **way higher** than the one in the **train dataset** as shown below

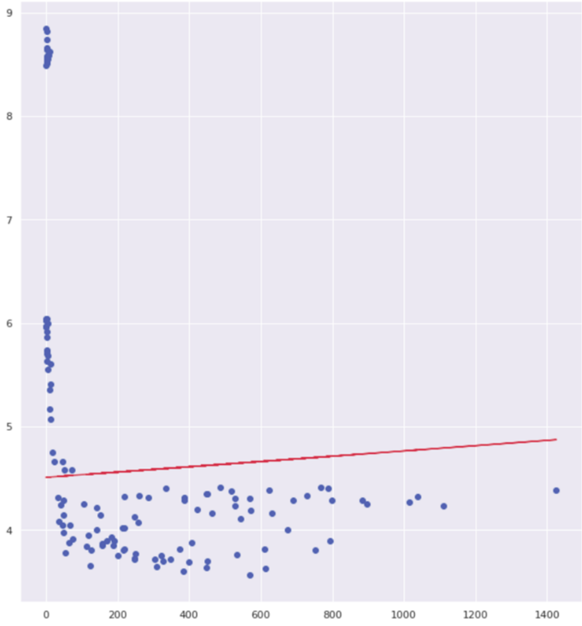
```
# Check the Goodness of Fit (on Train Data)
print("Goodness of Fit of Model \tTrain Dataset")
print("Explained Variance (R^2) \t:", linreg.score(X_train, y_train))
print("Mean Squared Error (MSE) \t:", mean_squared_error(y_train, y_train_pred))
print()

# Check the Goodness of Fit (on Test Data)
print("Goodness of Fit of Model \tTest Dataset")
print("Explained Variance (R^2) \t:", linreg.score(X_test, y_test))
print("Mean Squared Error (MSE) \t:", mean_squared_error(y_test, y_test_pred))
print()

# Plot the Predictions vs the True values
f, axes = plt.subplots(1, 2, figsize=(24, 12))

regline_x = X_train
regline_y = linreg.intercept_ + linreg.coef_ * X_train
axes[0].scatter(X_train, y_train)
axes[0].plot(regline_x, y_train_pred, 'r-', linewidth = 3)

axes[1].scatter(X_test, y_test)
axes[1].plot(X_test, y_test_pred, 'r-')

plt.show()
```

```
Intercept of Regression        : b =  [4.50961208]
Coefficients of Regression     : a =  [[0.00025521]]

Goodness of Fit of Model        Train Dataset
Explained Variance (R^2)        : 0.13416674312561616
Mean Squared Error (MSE)        : 0.3875934660305546

Goodness of Fit of Model        Test Dataset
Explained Variance (R^2)        : -0.09959229910771028
Mean Squared Error (MSE)        : 2.7971092381132534
```

Aviation (Singapore Airlines (SIA))



Train Dataset



Test Dataset

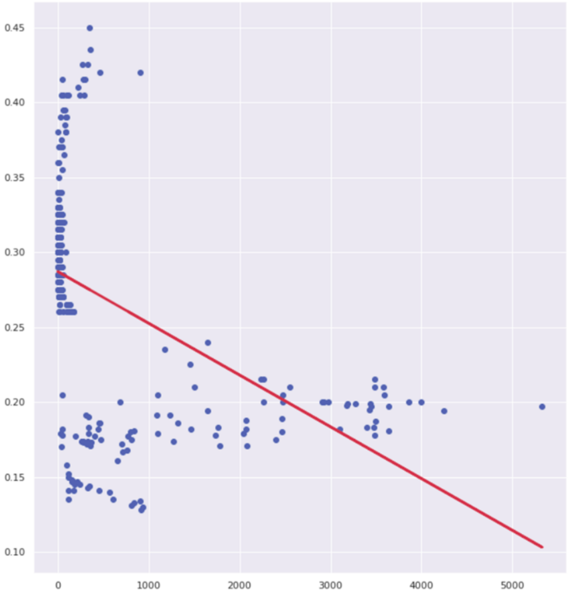
Mean Squared Error (MSE) for Pharmaceutical (Biolidics)

Through our analysis, the MSE for all the **test data set** in all models are **way higher** than the one in the **train dataset** as shown below

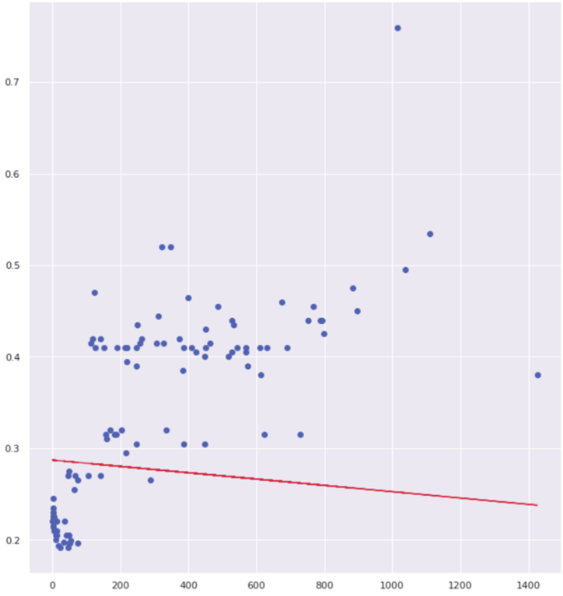
```
# Check the Goodness of Fit (on Train Data)
print("Goodness of Fit of Model \tTrain Dataset")
print("Explained Variance (R^2) \t:", linreg.score(X_train, y_train))
print("Mean Squared Error (MSE) \t:", mean_squared_error(y_train, y_train_pred))
print()

# Check the Goodness of Fit (on Test Data)
print("Goodness of Fit of Model \tTest Dataset")
print("Explained Variance (R^2) \t:", linreg.score(X_test, y_test))
print("Mean Squared Error (MSE) \t:", mean_squared_error(y_test, y_test_pred))
print()

# Plot the Predictions vs the True values
f, axes = plt.subplots(1, 2, figsize=(24, 12))
regline_x = X_train
regline_y = linreg.intercept_ + linreg.coef_ * X_train
axes[0].scatter(X_train, y_train)
axes[0].plot(regline_x, regline_y, 'r-', linewidth = 3)

axes[1].scatter(X_test, y_test)
axes[1].plot(X_test, y_test_pred, 'r-')

plt.show()
```

```
Intercept of Regression          : b =  [0.28714339]
Coefficients of Regression       : a =  [[-3.45554954e-05]]

Goodness of Fit of Model         Train Dataset
Explained Variance (R^2)         : 0.24112466820337564
Mean Squared Error (MSE)         : 0.0034654606789729175

Goodness of Fit of Model         Test Dataset
Explained Variance (R^2)         : -0.3614249856269791
Mean Squared Error (MSE)         : 0.015649046916820039
```

Pharmaceutical (Biolidics)



Train Dataset



Test Dataset

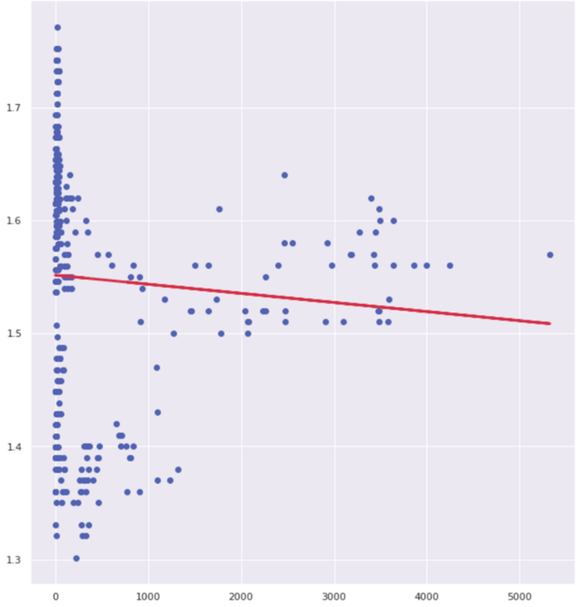
Mean Squared Error (MSE) for Transportation (ComfortDelgro Corp)

Through our analysis, the MSE for all the **test data set** in all models are **way higher** than the one in the **train dataset** as shown below

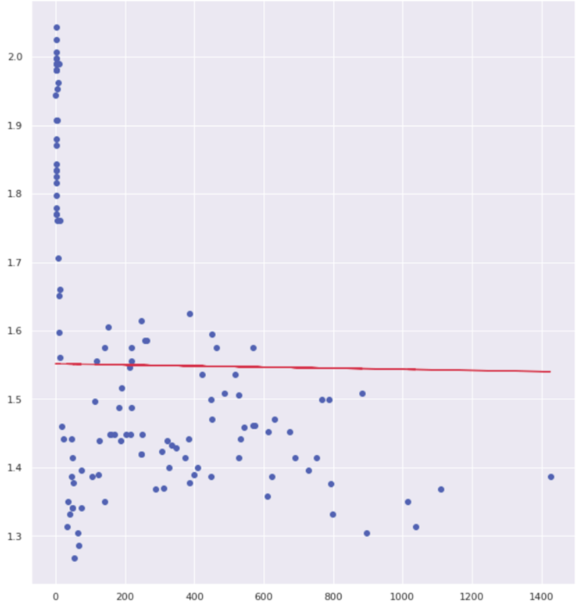
```python
# Check the Goodness of Fit (on Train Data)
print("Goodness of Fit of Model \tTrain Dataset")
print("Explained Variance (R^2) \t:", linreg.score(X_train, y_train))
print("Mean Squared Error (MSE) \t:", mean_squared_error(y_train, y_train_pred))
print()

# Check the Goodness of Fit (on Test Data)
print("Goodness of Fit of Model \tTest Dataset")
print("Explained Variance (R^2) \t:", linreg.score(X_test, y_test))
print("Mean Squared Error (MSE) \t:", mean_squared_error(y_test, y_test_pred))
print()

# Plot the Predictions vs the True values
f, axes = plt.subplots(1, 2, figsize=(24, 12))

regline_x = X_train
regline_y = linreg.intercept_ + linreg.coef_ * X_train
axes[0].scatter(X_train, y_train)
axes[0].plot(regline_x, y_train_pred, 'r-', linewidth = 3)

axes[1].scatter(X_test, y_test)
axes[1].plot(X_test, y_test_pred, 'r-')

plt.show()
```

```
Intercept of Regression         : b =  [1.55154755]
Coefficients of Regression      : a =  [[-8.07173852e-06]]

Goodness of Fit of Model        Train Dataset
Explained Variance (R^2)        : 0.004978881901344101
Mean Squared Error (MSE)        : 0.012006946543906662

Goodness of Fit of Model        Test Dataset
Explained Variance (R^2)        : 0.005852829094018297
Mean Squared Error (MSE)        : 0.04746906208912548
```

Transportation (ComfortDelgro Corp)



Train Dataset



Test Dataset

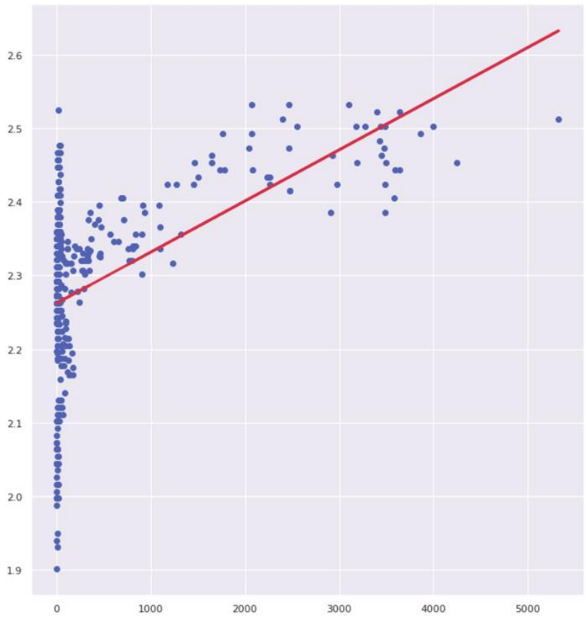
Mean Squared Error (MSE) for Telecommunications (SingTel)

Through our analysis, the MSE for all the **test data set** in all models are **way higher** than the one in the **train dataset** as shown below

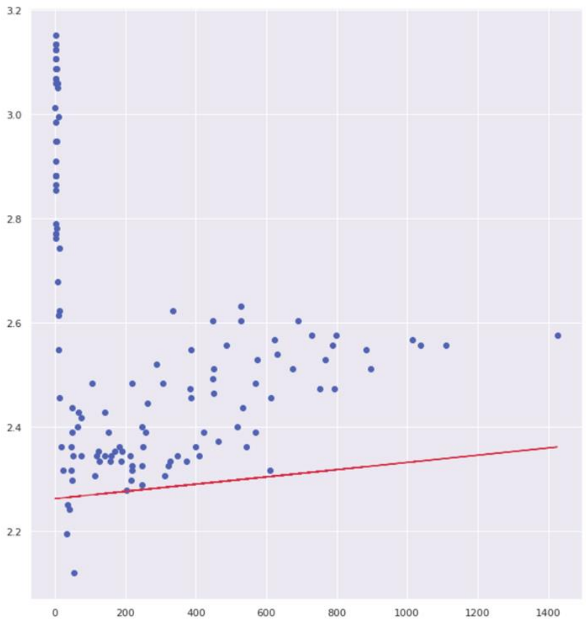
```
# Check the Goodness of Fit (on Train Data)
print("Goodness of Fit of Model \tTrain Dataset")
print("Explained Variance (R^2) \t:", linreg.score(X_train, y_train))
print("Mean Squared Error (MSE) \t:", mean_squared_error(y_train, y_train_pred))
print()

# Check the Goodness of Fit (on Test Data)
print("Goodness of Fit of Model \tTest Dataset")
print("Explained Variance (R^2) \t:", linreg.score(X_test, y_test))
print("Mean Squared Error (MSE) \t:", mean_squared_error(y_test, y_test_pred))
print()

# Plot the Predictions vs the True values
f, axes = plt.subplots(1, 2, figsize=(24, 12))
regline_x = X_train
regline_y = linreg.intercept_ + linreg.coef_ * X_train
axes[0].scatter(X_train, y_train)
axes[0].plot(regline_x, regline_y, 'r-', linewidth = 3)

axes[1].scatter(X_test, y_test)
axes[1].plot(X_test, y_test_pred, 'r-')

plt.show()
```

```
Intercept of Regression         : b =  [2.26200127]
Coefficients of Regression      : a =  [[6.95165228e-05]]

Goodness of Fit of Model        Train Dataset
Explained Variance (R^2)        : 0.2852775572371870 6
Mean Squared Error (MSE)        : 0.011164616967138 68

Goodness of Fit of Model        Test Dataset
Explained Variance (R^2)        : -1.193710516534934 7
Mean Squared Error (MSE)        : 0.1531052743570867
```

Telecommunications (SingTel Telecommunications Ltd)



Train Dataset



Test Dataset

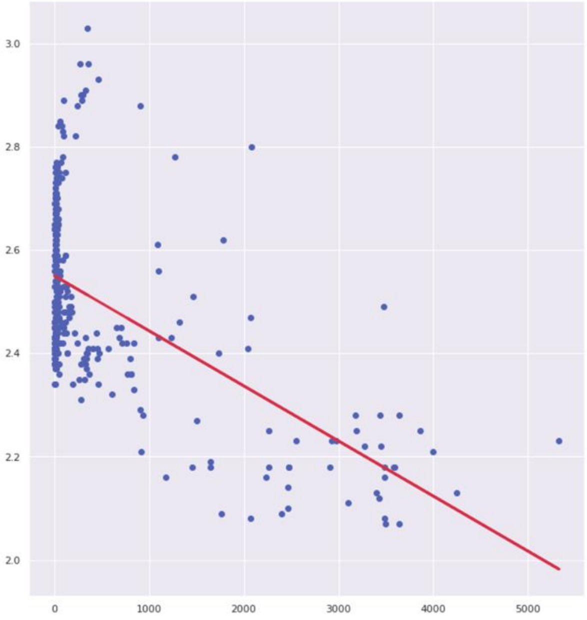
Mean Squared Error (MSE) for Information technologies (Creative)

Through our analysis, the MSE for all the **test data set** in all models are **way higher** than the one in the **train dataset** as shown below.

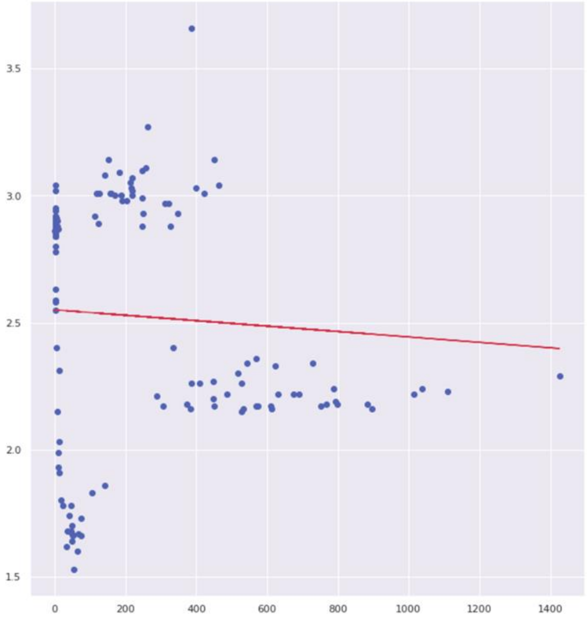
```
# Check the Goodness of Fit (on Train Data)
print("Goodness of Fit of Model \tTrain Dataset")
print("Explained Variance (R^2) \t:", linreg.score(X_train, y_train))
print("Mean Squared Error (MSE) \t:", mean_squared_error(y_train, y_train_pred))
print()

# Check the Goodness of Fit (on Test Data)
print("Goodness of Fit of Model \tTest Dataset")
print("Explained Variance (R^2) \t:", linreg.score(X_test, y_test))
print("Mean Squared Error (MSE) \t:", mean_squared_error(y_test, y_test_pred))
print()

# Plot the Predictions vs the True values
f, axes = plt.subplots(1, 2, figsize=(24, 12))
regline_x = X_train
regline_y = linreg.intercept_ + linreg.coef_ * X_train
axes[0].scatter(X_train, y_train)
axes[0].plot(regline_x, regline_y, 'r-', linewidth = 3)

axes[1].scatter(X_test, y_test)
axes[1].plot(X_test, y_test_pred, 'r-')

plt.show()
```

```
Intercept of Regression          : b =  [2.5503403]
Coefficients of Regression       : a =  [[-0.0001067]]

Goodness of Fit of Model         Train Dataset
Explained Variance (R^2)         : 0.3407663737239871
Mean Squared Error (MSE)         : 0.02030998803007122

Goodness of Fit of Model         Test Dataset
Explained Variance (R^2)         : 0.015786427045632268
Mean Squared Error (MSE)         : 0.23742137007322217
```

Information technology (Creative Technologies)



Train Dataset



Test Dataset

Explained Variance

1. Some stocks have a negative explained variance in the **test dataset**, indicating there is an inverse relation from the **train dataset.** For example, SIA (Aviation), Biolidics (Pharma), SingTel (Telecommunications)

```
Intercept of Regression             : b =  [4.50961208]
Coefficients of Regression          : a =  [[0.00025521]]

Goodness of Fit of Model            Train Dataset
Explained Variance (R^2)            : 0.13416674312561616
Mean Squared Error (MSE)            : 0.3875934660305546

Goodness of Fit of Model            Test Dataset
Explained Variance (R^2)            : -0.09959229910771028
Mean Squared Error (MSE)            : 2.7971092381132534
```

SIA (Aviation)

```
Intercept of Regression             : b =  [0.28714339]
Coefficients of Regression          : a =  [[-3.45554954e-05

Goodness of Fit of Model            Train Dataset
Explained Variance (R^2)            : 0.24112466820337564
Mean Squared Error (MSE)            : 0.0034654606789729175

Goodness of Fit of Model            Test Dataset
Explained Variance (R^2)            : -0.3614249856269791
Mean Squared Error (MSE)            : 0.01564904691682039
```

Biolidics (Pharma)

```
Intercept of Regression             : b =  [2.26200127]
Coefficients of Regression          : a =  [[6.95165228e-05]]

Goodness of Fit of Model            Train Dataset
Explained Variance (R^2)            : 0.28527755723718706
Mean Squared Error (MSE)            : 0.01116461696713868

Goodness of Fit of Model            Test Dataset
Explained Variance (R^2)            : -1.1937105165349347
Mean Squared Error (MSE)            : 0.1531052743570867
```

SingTel (Telecommunications)

Explained Variance

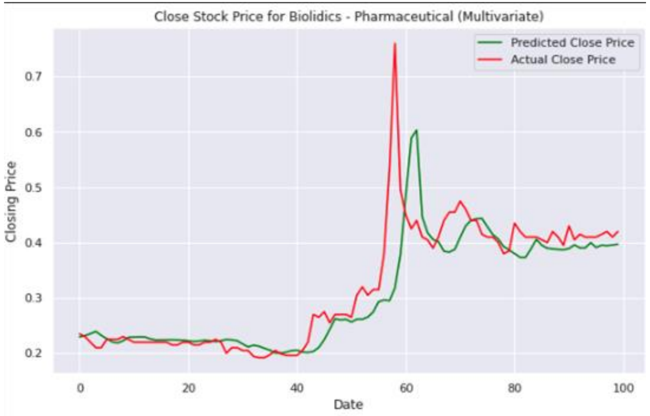
1. As seen for all 3 companies representing 3 industries, there is a low correlation in the train dataset but it has even a weaker correlation in the test dataset. Hence, **linear regression model** is not a good predictor for stock prices using COVID-19 data.
2. Thus, we have decided to use another model to predict future stock prices using COVID-19 Data.
3. We need to consider think of it as a multivariate timeseries problem

Long Short-Term Memory Network Model

- Recurrent Neural Network
- Time-series Model
- Remember **previous sequential** data and use it for processing the current input
- Predict first 100 datapoints

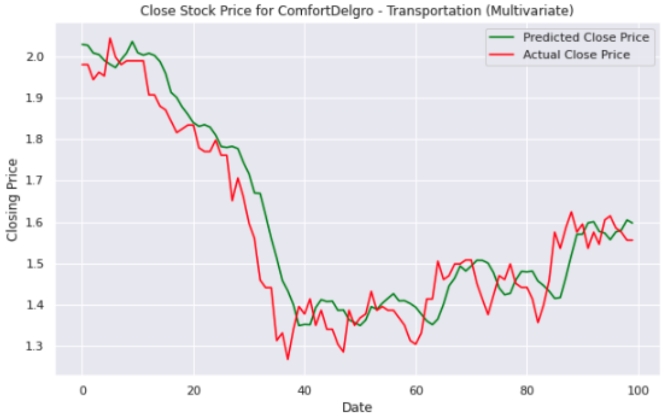

```
Explained Variance (R^2)          : 0.9485801907990711
Mean Squared Error (MSE)          : 0.0007395508

Goodness of Fit of Model          Test Dataset
Explained Variance (R^2)          : 0.9118968735085866
Mean Squared Error (MSE)          : 0.006080222
```

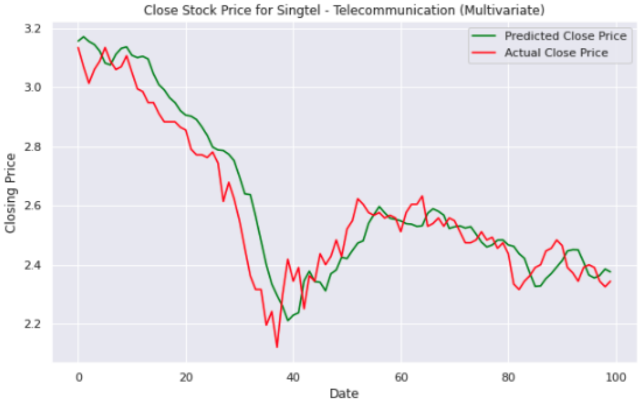
Aviation (SIA)




```
Goodness of Fit of Model          Train Dataset
Explained Variance (R^2)          : 0.9249127283462957
Mean Squared Error (MSE)          : 0.0008138199

Goodness of Fit of Model          Test Dataset
Explained Variance (R^2)          : 0.6319612188736942
Mean Squared Error (MSE)          : 0.010606961
```

Pharmaceutical (Biolidics)






```
Goodness of Fit of Model        Train Dataset
Explained Variance (R^2)        : 0.89603997658061
Mean Squared Error (MSE)        : 0.0020355128

Goodness of Fit of Model        Test Dataset
Explained Variance (R^2)        : 0.8846303992980342
Mean Squared Error (MSE)        : 0.008154833
```

```
Goodness of Fit of Model        Train Dataset
Explained Variance (R^2)        : 0.878366475012218
Mean Squared Error (MSE)        : 0.0012276274

Goodness of Fit of Model        Test Dataset
Explained Variance (R^2)        : 0.8813825906618244
Mean Squared Error (MSE)        : 0.004603002
```

Transportation (ComfortDelgro)

Telecommunications (SingTel)


```
Goodness of Fit of Model        Train Dataset
Explained Variance (R^2)        : 0.7462196755714823
Mean Squared Error (MSE)        : 0.0016062096

Goodness of Fit of Model        Test Dataset
Explained Variance (R^2)        : 0.6779904354693729
Mean Squared Error (MSE)        : 0.017096665
```

Conclusion

- None of the stocks selected worked with Covid-19 for the **linear regression model** since the explained variance is close to zero or negative. The MSE is also too large for test datasets
- Linear Regression Model might not be optimal for time-series dataset
- Thus, to predict these stocks, we need to consider other facts and think of it as a multivariate problem
- We can also look for other stocks to find a relation between them with Covid-19 cases
- Lastly, we can also look for other Machine Learning models like LSTM to predict stock prices as seen previously as it is suitable for time series datasets.

Learning Points

Learning Points

- An API allows the user to collect real time data that is updated regularly, depending on the data provider
- How to extract data using API to retrieve real time data
- How using pd.to_datetime().date() allows appropriate date format to be used

Learning Points

- Understanding the data we are extracting from e.g. Stock Market closes on weekends and public holidays. Thus, removal of corresponding data for proper alignment is necessary
- Plotting of 2 times series data with different units on the same graph
- Learning new machine learning model like LSTM which uses RNN

Possible Improvements

Possible Improvements

- Include covid cases for weekends and holidays
    - These cases may increase and affect stock prices
- Explore other factors that can cause stocks prices to change
    - For example, political unrest, improvement of technology, speculation, brand reputation, etc.

Thank You!

The team

Lee Zheng Xuan

- Finding Datasets and researching on the suitable API for project e.g. Yfinance
- Constructed codes to clean datasets
- Organisation of codes
- Constructed Exploratory Analysis such as Correlations and Biplots with team
- Analysis of data output to determine the relevance to the project
- Help evaluate and directed the flow of reasoning of our project
- Constructed both Linear Regression and LSTM models
- Worked with teammates to come out with conclusion and analysis
- Designed and created PowerPoint and formulate explanations for script

Jaren Ng Shing Yu

- Problem statement formulation
- Built skeleton and data extraction codes for retrieving data in csv and Yfinance, and code for all the other sections
- Breaking down the project into different sections and writing code e.g. (Problem statement, collection of dataset, cleaning of dataset, EDA, Machine Learning etc.)
- Organisation of codes
- Analysis of data output to determine the relevance to the project
- Constructed Exploratory Analysis like Correlations and Biplots with team
- Help construct the Linear Regression and LSTM models
- Evaluating the Machine Learning Regression model and suggesting new models to fit time series
- Working together with team to come out conclusion and analysis for our project
- Designed and created PowerPoint and formulate explanations for script

Joel Tham

- Researched the appropriate datasets to use
- Researched RNN/LSTM ML models
- Constructed draft prototype of LSTM model and which helped us finalise our LSTM model
- Evaluate the methodology to make sure it made sense
- Ensured that our datasets was cleaned and explored properly
- General error checking
- Vetted PowerPoint slides, script, and code to ensure coherence with course materials, and in general
- Organising and labelling of our various code segments
- Coordinated with the team to finalise our conclusion and analysis for our project