



# BDAS DATAMINING CASE STUDY

Peng Jiang

Pjia958

Github repository page: <https://github.com/pjia958/BDAS>

## **1. Business and Situation Understanding**

The world is facing a rapidly changing nowadays, sustainable development in all sections of countries is the common goal of public persuasion. According to the sustainable development goals (SDGs) and their 169 targets released by the United Nation, they contain a range of aspects in human life including social, economic, and environmental categories. However, achieving these goals require people to take action on the exploration and development of innovation affirmatively (Cancino et al., 2009). There is no doubt that innovation plays a key role in building a country's economic competitiveness. Solow (1956) pointed out that there are positive links between economic growth and technology development. Romer (1990) also illustrated their relationship in a further step which told that technological innovation itself is the outcome of market stimulation and it made a contribution in turn on the accumulation of public wealth. In this context, SDGs set goals 8.2 specifically which emphasize the development of innovation and achieving higher levels of economic productivity and attain sustainable economic growth consequently.

### **1.1 Identify the objectives of the business/situation**

Regarding the domestic contribution of innovation development, industries and companies are the essential components in making progress. According to Govindarajan and Trimble (2012), local companies' innovation outcomes can benefit society effectively and contribute to them globally. So, it's meaningful and purposeful to lead to the insight of the companies' innovation development, in another world, research and development (R&D). To understand and discover how and where a company should focus on innovation, the study is commissioned with the following objectives:

- Understand and discover the relation between industrial input on R&D and its influence by the case study.
- Evaluate a company's R&D strategy and provide effective intervention suggestions.

The tentative criteria to access if the study outcomes are successful will be:

- The relation can be discovered with meaningful and generalized interpretation.
- Provide constructive suggestions on how to achieve sustained development for companies that can be measured or assessed effectively.

## 1.2 Assess the situation

### 1.2.1 Requirements

Regarding the business target, it is necessary to take a case study on companies' and industries' economic performance. To solve data mining tasks, multiple areas of knowledge are commonly involved (Vadim, 2018). By using Big Data Analytics Solution (BDAS), one needs to have related knowledge structure and the usage of specific tools. Benefit from a wide range of external, open-source code library including pySpark lib, Python programming is an appropriate method as it is flexible and powerful on addressing data mining problems. Besides, there are also other tools to analyze and visualize the data. Researchers should have the capability to understand the case thoroughly and harness the tools properly as well.

### 1.2.2 Data resources

Olafsson et al. (2008) suggested that it's vital to identify the data source during the process of data mining. Although the data resources are abundant, the data should be gathered by the following three criteria. Firstly, As the data source should be credible, only by so the analyzing process and result can be convincing. Then, the data must have typical features so that it can be a good case for addressing the problem. Lastly, it has to be legally to gain and use the data. To sum up, data released by official departments or organizations of companies' R&D development is proper to choose.

### 1.2.3 Assumptions and constraints

According to Sculley and Pasanek (2008), the underlying assumptions of the data mining process can be also conducted here in which the data selected is from fixed distribution and well represented, the definition of possible outcomes of the research are also restricted by theory or model chosen.

### 1.2.4 Risks and contingencies

As the whole procedure of data mining is done within a perspective of academic research, there's no significant risk of financial or Scheduling. However, because the quality of data is various, several methods or algorithms may conduct, and the results may not idealistic and notable as expected.

### 1.3 Determine data mining goals

With the understanding of current study target and situation, the business object can be translated into data analysis goals. The initial goals to be completed of data mining objects are:

- Use historical statistic data of companies and industries performance, mainly measured by sales revenue and other attributes, to discover the specific patterns with industrial input on the R&D department (prediction analysis).
- Assess a company's R&D strategy based on the pattern discovered and predict the prospect of its development in the future (prediction evaluation).

The criteria to access if the data mining process's outcomes are successful will be:

- At least a pattern of the data on companies' R&D investment and sales revenue influenced by a is discovered.
- The model can predict a company's future statues by acknowledging the information on its R&D input and measure the error effectively.

Besides, there are also technical terms to ensure data mining process on correct track. These are also going to be conducted including:

- Description on model choosing and assessment. Accuracy and the performance of the model will be included.
- The specific coefficients and will show after the conduction of the data with the statistical outcomes
- Optimization of the data analysis results.

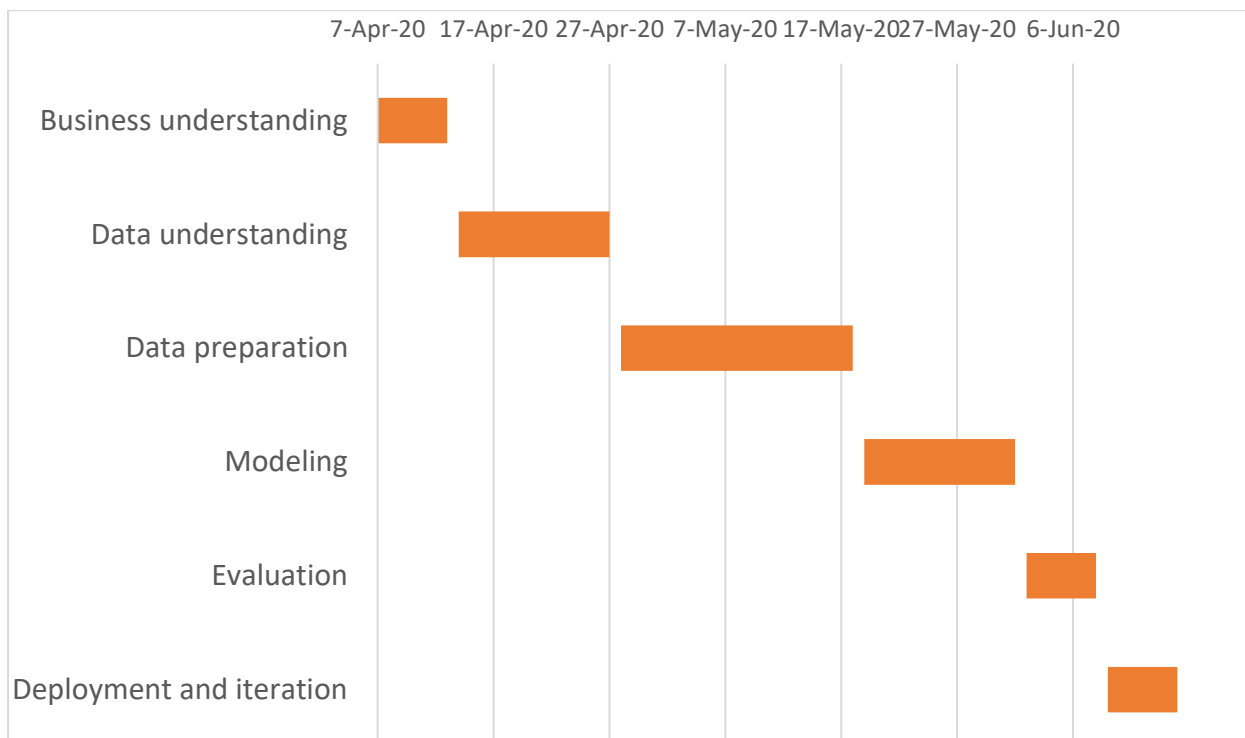
### 1.4 Produce a project plan

The project plan can be edited in the Gantt chart after listing in the table.

Phase	Time	Risks
Business understanding	1 week	Extra time of data set filtering

Data understanding	2 weeks	Data problems, technologies choosing issues
Data preparation	3 weeks	Data handling problems
Modeling	2 weeks	Technology and modeling choosing issues
Evaluation	1 week	Result may be ineffective
Deployment and iteration	1 week	Result may be ineffective

By setting each tasks' starting week and duration, a day-to-day plan chart can be drawn. On the table above, all phases are included during the analysis. In our case, the orange bars mean the task under operating. The first step is business understanding, background research should proceed. Then the data understanding process will begin. Researchers need to collect and build a structure of both the data and the data mining process. Afterward, the data preparation will start. Data modeling work will begin one week later as the data needs to be modified meanwhile modeling. The last step is evaluation and interpreting, necessary iteration can be also conducted.



## 2. Data understanding

Before going into the second phase, necessary environment setup work is needed. As for the Big Data Analytics Solutions, as known as BDAS, the main idea is to operate big data analytics tasks on the cloud. The reason is that the cloud server has a high capacity of computing, it can handle the data mining tasks in a short period of response time. In this study, the cloud server selected is Elastic Compute Cloud (EC2) in Amazon Web Services (AWS). The steps below show the process of building an EC2 instance in AWS and connect to it.

Firstly, an EC2 instance should be created. In AWS console, the specific steps of creating an EC2 instance can be checked and decided. When creating it, two settings are compulsory to be made, which are the Amazon Machine Image (AMI) and instance type.

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 1: Choose an Amazon Machine Image (AMI) Cancel and Exit

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start (0)

My AMIs (0)

AWS Marketplace (13)

Community AMIs (1)

Operating system

☐ Amazon Linux

☐ Cent OS


☐ Debian


☐ Fedora


☐ Gentoo


☐ openSUSE


☐ Other Linux

















722-Image - ami-06b331a0389419d94

722-Image

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

64-bit (x86)

Select

The following results for "722-image" were found in other catalogs:

13 results in AWS Marketplace

AWS Marketplace provides partnered Software that is pre-configured to run on AWS

## Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

In this iteration report, the settings are chosen as above. However, in practical scenario of data mining, all configurations should be selected according to the business object and task object, financial restriction will be also considered.

Then, a key pair will be generated. This is used as an authentication key for logging into the cloud server. After this step, the instance can be created successfully.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 7: Review Instance Launch

Please review your Instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**Improve your instances**  
Your instances may be improved by using the latest AMIs. You can also open a support case to get help with security groups.

**AMI Details**  
722-Image - a  
722-Image  
Root Device Type: e

**Instance Type**  
Instance Type ECU  
t2.micro Vari

**Select an existing key pair or create a new key pair**

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

**Key pair name**  
pjia958

Download Key Pair

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

Known IP addresses  
(80) for web servers. [Edit](#)

[Edit AMI](#)

[Edit instance type](#)

Network Performance  
Low to Moderate

Cancel Previous Launch

New EC2 Experience [Learn more](#)

EC2 Dashboard **New**

Events **New**

Tags

Reports

Limits

INSTANCES

**Instances**

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts **New**

Scheduled Instances

Capacity Reservations

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

**Launch Instance** **Connect** **Actions**

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
	i-00f43d00735a7ea56	t2.micro	us-east-1e	running	Initializing	None

Instance: **i-00f43d00735a7ea56** Public DNS: ec2-35-174-184-81.compute-1.amazonaws.com

**Description** **Status Checks** **Monitoring** **Tags**

Instance ID	Public DNS (IPv4)
i-00f43d00735a7ea56	ec2-35-174-184-81.compute-1.amazonaws.com

Instance state	IPv4 Public IP
running	35.174.184.81

As can be seen from the screenshot above, the EC2 instance is created and in a running state, local computer can connect the cloud server by using the key pair above, this process is done in PuTTY. In Mac OS, the installation of PuTTY and key generation is different with these in Window system. After searching for the correct method, it can finally connect to the cloud server.

New EC2 Experience [Learn more](#)

EC2 Dashboard **New**

Events **New**

Tags

Reports

Limits

INSTANCES

**Instances**

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts **New**

Scheduled Instances

Capacity Reservations

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

**Launch Instance** **Connect** **Actions**

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
	i-00f43d00735a7ea56	t2.micro	us-east-1e	running	Initializing	None

Instance: **i-00f43d00735a7ea56** Public DNS: ec2-35-174-184-81.compute-1.amazonaws.com

**Description** **Status Checks** **Monitoring** **Tags**

Instance ID	Public DNS (IPv4)
i-00f43d00735a7ea56	ec2-35-174-184-81.compute-1.amazonaws.com

Instance state	IPv4 Public IP
running	35.174.184.81

ubuntu@ip-172-31-48-192: ~

```

V3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (.ssl,c:645)
[I 10:05:36.680 NotebookApp] Adapting to protocol v5.1 for kernel 6d5c1153-b725-42c5-9211-34a739e72559
[I 10:07:35.670 NotebookApp] Saving file at /Untitled.ipynb
[I 10:22:03.858 NotebookApp] Starting buffering for 6d5c1153-b725-42c5-9211-34a739e72559:ea4e20ae49fa432fb0cbe2dde90d8d0
[W 10:22:05.433 NotebookApp] SSL Error on 9 ('116.12.62.212', 53785): [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (.ssl,c:645)
[I 10:22:05.065 NotebookApp] Adapting to protocol v5.1 for kernel 6d5c1153-b725-42c5-9211-34a739e72559
[I 10:22:06.054 NotebookApp] Restoring connection for 6d5c1153-b725-42c5-9211-34a739e72559:ea4e20ae49fa432fb0cbe2dde90d8d0
[W 10:24:15.915 NotebookApp] SSL Error on 10 ('116.12.62.212', 53800): [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (.ssl,c:645)
[W 10:24:15.922 NotebookApp] SSL Error on 11 ('116.12.62.212', 53801): [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (.ssl,c:645)
[I 10:24:32.153 NotebookApp] Saving file at /Untitled.ipynb
[I 10:32:43.879 NotebookApp] Starting buffering for 6d5c1153-b725-42c5-9211-34a739e72559:ea4e20ae49fa432fb0cbe2dde90d8d0

```

Joseph@Josephs-Air ~ % putty

```

(putty:14246): Gtk-WARNING **: 21:54:57.908: Attempting to store
/Users/Joesph/.local/share/recently-used.xbel', but failed: Fai
?/Users/Joesph/.local/share/recently-used.xbel,3CUFW0?: No su
ry

(putty:14246): Gtk-WARNING **: 21:54:57.908: Attempting to set
f '/Users/Joesph/.local/share/recently-used.xbel', but failed: Fai
?/Users/Joesph/.local/share/recently-used.xbel,NLFW0?: No su
ry

(putty:14246): Gtk-WARNING **: 21:55:08.661: Attempting to stor
/Users/Joesph/.local/share/recently-used.xbel', but failed: Fai
?/Users/Joesph/.local/share/recently-used.xbel,NLFW0?: No su
ry

(putty:14246): Gtk-WARNING **: 21:55:08.661: Attempting to set
f '/Users/Joesph/.local/share/recently-used.xbel', but failed: Fai
irectory

```



The image shows two screenshots. The top screenshot is from the AWS Management Console, displaying a table of EC2 instances. The bottom screenshot shows a Jupyter Notebook interface running on an EC2 instance, with a code cell executed.

**AWS Management Console Screenshot:**

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm
	i-00f43d00735a7ea56	t2.micro	us-east-1e	running	2/2 checks ...	Non

**Jupyter Notebook Screenshot:**

Instance: i-00f43d00735a7ea56 Public DNS: ec2-35-174-184-81.compute-1.amazonaws.com

Public DNS (IPv4): ec2-35-174-184-81.compute-1.amazonaws.com  
IPv4 Public IP: 35.174.184.81

Instance state: running

ConnSuccess 35.174.184.81:8888/notebooks/ConnSuccess.ipynb

jupyter ConnSuccess Last Checkpoint: 1 小时前 (unsaved changes)

```
In [3]: print('hello EC2')
hello EC2
```

Finally, EC2 instances is connected. On the cloud server, a Jupyter notebook is running. It can be used to run python code and conduct the further data mining tasks. Besides, the Spark, a unified and high-effective data analytics engine will be used in this study. Lastly, data understanding phase can start.

## 2.1 Collect initial data

Concerning the understanding of business data mining object and the situation analyzed, the data need to reflect the companies and industrial's performance before and after different degrees of R&D intervention, so that the data resources are constrained on national and organizational websites. However, it's not easy to find a suitable data set as it can be too large or digress and fuzzy. After spending time and effort on browsing and evaluating a range of data sets, the raw data set was collected on an official open data website of the United

Kingdom, 'data.gov.uk'. By collecting data on this site, the credit, accuracy, legality, and integrity of the data can be ensured.

## 2.2 Describe the data

The data set is named “Global companies by R&D investment by sector 2010”, recorded detailed information of top-1000 global companies and their economic performance by R&D investment. Formatted in comma-separated values (CSV), the raw data set contains 38 columns and 1050 rows.

The fields of the datasets involve the multi-dimension information of the company and different kinds of data types including string, Boolean, continuous number and categorical variables. The first part is the brief information of a company including its location, industry description, whether it follows International Financial Reporting Standards (IFRS) or listed in Financial Times Global 500. Then comes the companies' financial record, including their investment in R&D, operation profit, employee status, and sales data. The dataset also brings in a short period of historical data of R&D input. There is little overlap in the coding schemes for the various data sources because the data sources contain different attributes with details. In the field of IFRS, asterisks are used to indicate that the company doesn't follow IFRS.

Overall, it's a high-quality dataset with valid data. One of the superficial features of the data set is that it contains detailed data both in numerical and categorical type, listed in time series. It can be beneficial to forwarding data mining and knowledge digging (Freitas, 1996).

## 2.3 Explore the data

Since the raw data set that is a data set is edited with multiple table headers, the CSV file cannot be recognized and imported by open-source data analyzing tools. So, initial data preprocess work is implemented on this step. By combining the headers, the data can be successfully imported.

RANKING OF TOP 1,000 GLOBAL COMPANIES BY R&D INVESTMENT WITHIN INDUSTRY													
										2009 R&D investment			
										Growth		operating	
										1yr	4yr	profit	sales
										%	%	%	%
Company	Country	Normali	Foreign	Not IFRS	Industry	ICB code	Industry	Listing status	◆M				
1 Aerospace & defence (33)									12918.19	-0.7	4.5	60.4	4.1
EADS, T	The Netherlands				1	2713	Aerospa	FT Globi	2557.09	4	8		6.7
Boeing,	USA			*	1	2713	Aerospa	FT Globi	2148.74	1	16	165.7	5.1
Finmecc	Italy				1	2717	Defence	Listed	1711.24	9	7	166.9	11.7

Raw data set header

Compar	Country	Normali	Foreign	Not IFRS	Industry	ICB code	Industry	Listing s	2009 R&D	1 yr growth	4 yr growth	R&D inv	R&D inv
1 Aerospace & defence (33)									12918.19	-0.7	4.5	60.4	4.1
EADS, T	The Netherlands				1	2713	Aerospa	FT Glob	2557.09	4	8		6.7
Boeing,	USA			*	1	2713	Aerospa	FT Glob	2148.74	1	16	165.7	5.1
Finmecc	Italy				1	2717	Defence	Listed	1711.24	9	7	166.9	11.7
United	USA			*	1	2713	Aerospa	FT Glob	964.77	-12	-2	24.1	3
Thales,	France				1	2717	Defence	Listed	589.69	17	22		5.2

## Data set header after initial preprocessing

After initial preprocessing, it is allowed to get an insight into the data. Before moving onto the specific criteria, initial exploration work can be operated.

```
In [1]: import findspark
findspark.init('/home/ubuntu/spark-2.1.1-bin-hadoop2.7')
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('basics').getOrCreate()
```

```
In [2]: df = spark.read.format("csv").option("header", "true").load("./2010_RD_Score")
```

```
In [3]: df.show()
```

```
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
|          Company|Company ID|  Country|  Country GDP $|Not IFRS|In
dustry group|ICB code|Industry description|Listing status|2009 R&D inve
stment|1 Yr R&D investment growth%|4 Yr R&D investment growth%|R&D inve
stment / operation profit%|R&D investment / sales%|R&D + capex / sales%
|Operating profit|If made profit|1 Yr operating profit%|4 Yr operating
profit%|Operating profit / sales%|Sales|1 Yr sales growth%|4 Yr sales g
rowth%|Employees|1 Yr employees growth%|4 Yr employees growth%|R&D / em
ployees |1 Yr R&D / employees growth%|4 Yr R&D / employees growth%|Sale
s outside region%|Market cap|Market cap change%|If market cap increase|
2009-2010-2000|2009-2010-2000|2009-2010-2000|2009-2010-2000|2009-2010-2000|
```

```
In [10]: df.count()
```

```
Out[10]: 1000
```

It doesn't show well by directly using the show function as there are too many fields. So other functions can be tried. However, by using count function the quantity of data set can be shown.

```
In [6]: df.describe()
```

```
Out[6]: DataFrame[summary: string, Company: string, Company ID: string, Country:
string, Country GDP $: string, Not IFRS: string, Industry group: string,
ICB code: string, Industry description: string, Listing status: string, 2
009 R&D investment: string, 1 Yr R&D investment growth%: string, 4 Yr R&D
investment growth%: string, R&D investment / operation profit%: string, R
&D investment / sales%: string, R&D + capex / sales%: string, Operating p
rofit: string, If made profit: string, 1 Yr operating profit%: string, 4
Yr operating profit%: string, Operating profit / sales%: string, Sales: s
tring, 1 Yr sales growth%: string, 4 Yr sales growth%: string, Employees:
string, 1 Yr employees growth%: string, 4 Yr employees growth%: string, R
&D / employees : string, 1 Yr R&D / employees growth%: string, 4 Yr R&D /
employees growth%: string, Sales outside region%: string, Market cap: str
ing, Market cap change%: string, If market cap increase: string, R&D spen
d 2008: string, R&D spend 2007: string, R&D spend 2006: string, R&D spend
2005: string]
```

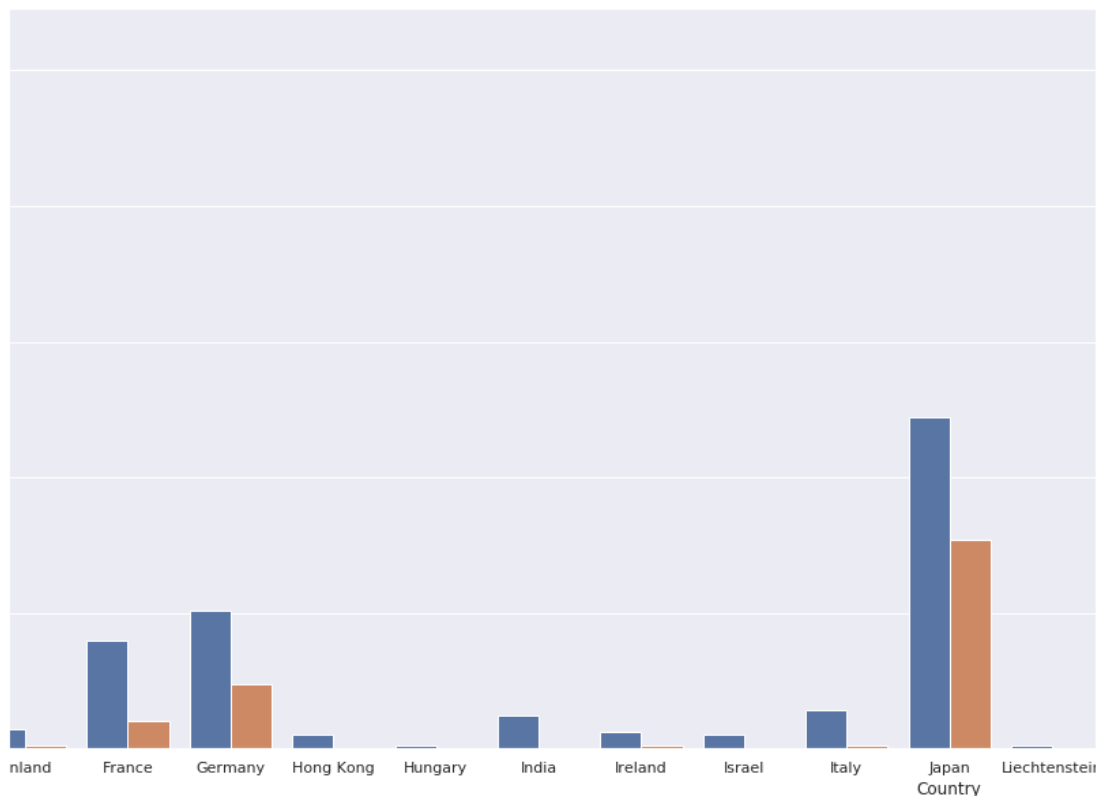
By using describe function, all the fields and their data type can be shown.

```
In [9]: df.printSchema()
```

```
root
|-- Company: string (nullable = true)
|-- Company ID: string (nullable = true)
|-- Country: string (nullable = true)
|-- Country GDP $: string (nullable = true)
|-- Not IFRS: string (nullable = true)
|-- Industry group: string (nullable = true)
|-- ICB code: string (nullable = true)
|-- Industry description: string (nullable = true)
|-- Listing status: string (nullable = true)
|-- 2009 R&D investment: string (nullable = true)
|-- 1 Yr R&D investment growth%: string (nullable = true)
|-- 4 Yr R&D investment growth%: string (nullable = true)
|-- R&D investment / operation profit%: string (nullable = true)
|-- R&D investment / sales%: string (nullable = true)
|-- R&D + capex / sales%: string (nullable = true)
|-- Operating profit: string (nullable = true)
|-- If made profit: string (nullable = true)
|-- 1 Yr operating profit%: string (nullable = true)
|-- 4 Yr operating profit%: string (nullable = true)
|-- Operating profit / sales%: string (nullable = true)
|-- Sales: string (nullable = true)
|-- 1 Yr sales growth%: string (nullable = true)
|-- 4 Yr sales growth%: string (nullable = true)
|-- Employees: string (nullable = true)
|-- 1 Yr employees growth%: string (nullable = true)
|-- 4 Yr employees growth%: string (nullable = true)
|-- R&D / employees : string (nullable = true)
|-- 1 Yr R&D / employees growth%: string (nullable = true)
|-- 4 Yr R&D / employees growth%: string (nullable = true)
|-- Sales outside region%: string (nullable = true)
|-- Market cap: string (nullable = true)
|-- Market cap change%: string (nullable = true)
```

By using the print schema function, all the fields' schema can be also gained. Still, matplotlib.pyplot is a good tool of visualization.

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb89d8767b8>
```



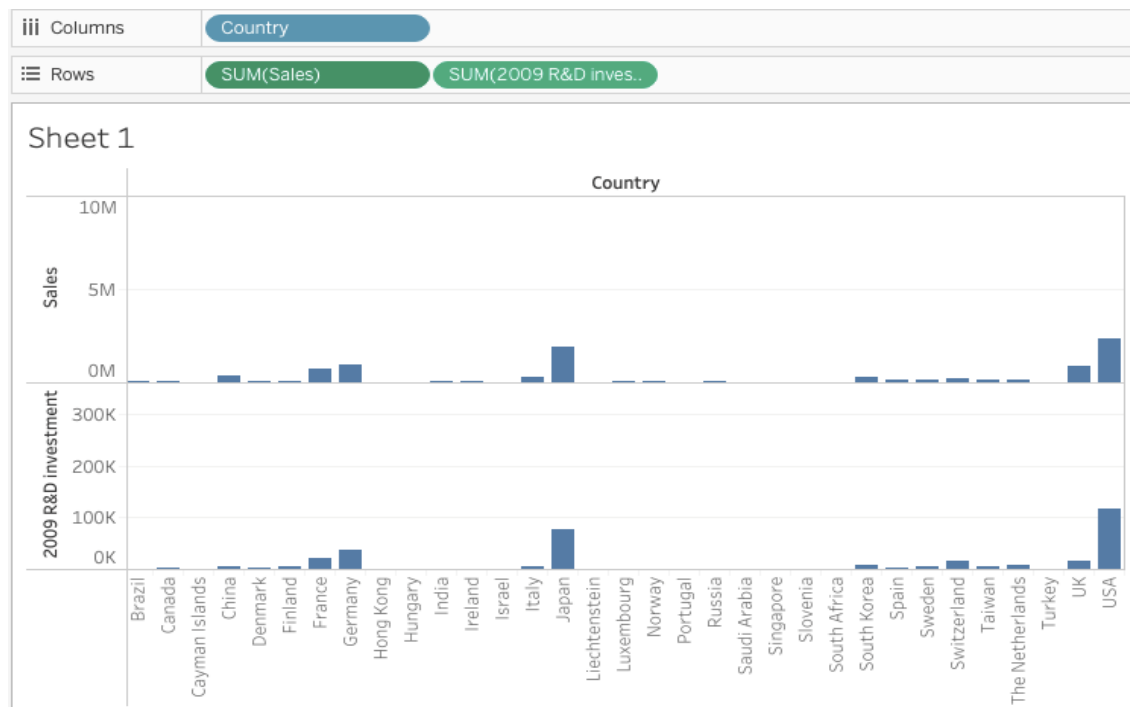
In the process of data exploration, it's essential to answer the following questions:

- What sort of hypotheses have you formed about the data?

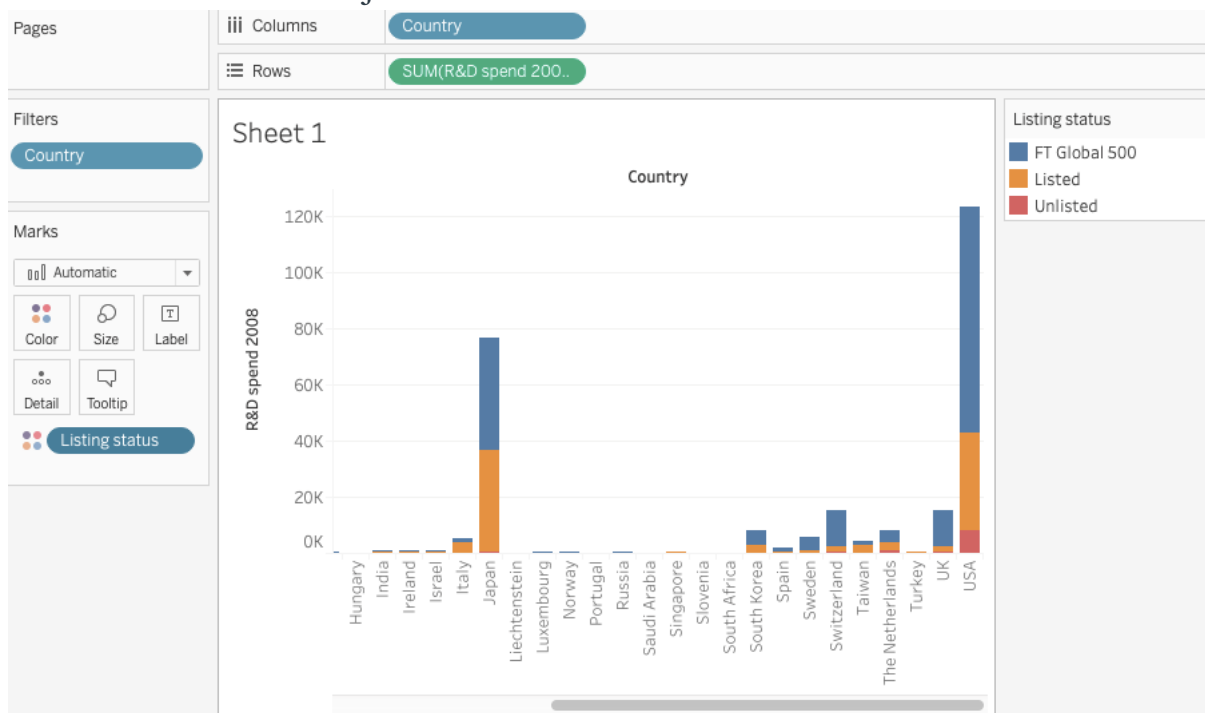
To address the business problem properly, the hypothesis is that there is a significant correlation between investment in R&D with the other one or several factors. To align with the business object, it's requisite to analyze the relationships among different variables.

- Which attributes seem promising for further analysis?

As the pyShark has a good capacity to do the data set computing tasks but limited visualization, other tools can be options. For answering this question, a correlation calculation is preferred. However, the original data types is string, so a further data process may conduct to handle this question. By using the open-source visualization tool, Tableau Desktop, some surface-lying relation can be revealed. It can be illustrated in diagrams. From the chart graph below, it illustrated the relationship between the sum of companies' sales revenue and investment in R&D. They have positive correlation relation by observation. It reflects the data set is qualified to conduct further analysis as well.



Another finding is that companies listed in Financial Times 500 have a higher tendency to invest in R&D investment. It can be treated as a standard to measure if a company is powerful or not. For those companies which are large and facing extremely intense competition, they regard innovation sectors as powerful driven force and spent a lot on enhancing it. This phenomenon can be evidence of business objects.



- Have your explorations revealed new characteristics of the data?

By visualizing the data set, another finding is about the data set quality, in which there are several rows of data that are not effective as shown below.

2010_RD_Scoreboard_Global... Company	2010_RD_Scoreboar... Country	2010_RD_Scoreboard_... Normalised	2010_RD_Scoreboard_Glob... Foreign owned	2010_RD_Scoreboa... Not IFRS
35 Real estate invest...	null	null	null	null
No companies with s...	null	null	null	null
36 Real estate invest...	null	null	null	null
No companies with s...	null	null	null	null
8 Electricity (13)	null	null	null	null
37 Software & compu...	null	null	null	null
6 Chemicals (69)	null	null	null	null

By browsing the description from the data source, the data is categorized by sector so that some rows are occupied for separation and explanation.

- How have these explorations changed your initial hypothesis?

Currently, the data set is considered qualified and seems promising to have convincing results, so the initial hypothesis basically remains the same.

- Can you identify particular subsets of data for later use?

Yes. According to the detailed category of the data set, it is convenient for researchers to select target columns and rows as analyze target. If needed, subsets are also can be identified explicitly.

## 2.4 Verify the data quality

Errors and missing values have been checked in previous steps; some rows need to be removed. By verify the quality, the missing values still exist in multiple columns. Python spark packages provide functions to remove all the missing values and it can be utilized as below.

```
#Data cleaning
df.na.drop()
```



Besides, the field “foreign owned” is also ineffective as the data is illustrated on a global perspective. Because the data set is already been done an initial preprocess, some of the header names exist inconsistencies, these issues need to be fixed in the data preparation process. The open-source tool Tableau provides powerful function, by which researchers can edit the dataset directly in the software.

In a word, in the data understanding process, basic acknowledgement of data set basic information has been gained. By having known the status of data set, the quality of data is on a high level, however the ineffective fields and data types need to be modified in data preparation phase.

### **3. Data preparation**

#### **3.1 Select the data**

Since the data set was collected by reviewing and filtering a large number of data sets, most information contained is effective and related to the data mining objects consequently. However, some of the redundant and meaningless items and attributes will be removed. For instance, the rows which used as separated line and columns “Foreign owned” will be deleted in forwarding steps.

Considering other columns, they contain the information which reflects a company’s scale, location, and economic performance. These factors may have a potential relationship between its R&D development and sales revenue, as a result, they are retained.

By doing as above, criteria below may access this process:

- Is a given attribute relevant to you data mining goals?

Yes, according to the business understanding object and data mining object, the attributes which are directly reflect the companies’ cost on investment and their outcomes can be closely relevant to the data mining goals.

- Does the quality of a particular data set or attribute preclude the validate of your results?

On current stage, the quality has influence on the data mining process and even data mining results. However it is understandable as there are none data processing tasks done. After data pre-processing, the quality can be



optimized significantly and it can be used validated in further data mining tasks.

- Can you salvage such data?

Yes, by data cleaning and construction, data set quality can be improved.

- Are there any constraints on using particular fields?

Considering the data mining object and data set, there are no content which may cause ethics issues, all the fields can be used openly.

### 3.2 Clean the data

To address the previous inconsistent columns names and invalid rows issues, Python is harnessed here. The rows can be filtered and removed accordingly by using the corresponding sentences. Every company should have an Industry Classification Benchmark (ICB) code to indicate the company's sector within the macroeconomy. If a row does not have one, it can be considered invalid. The column "Foreign owned" is judged as an ineffective attribute since the data set is on a global background. The capture below shows the data cleaning process and result.

As for the data content trimming work, the column "Normalized" is removed because of missing value issue.

Then, a

```
In [25]: #Data cleaning
df.na.drop()

Out[25]: DataFrame[Company: string, Company ID: string, Country: string, Country GDP $: string, Not IF
RS: string, Industry group: string, ICB code: string, Industry description: string, Listing s
tatus: string, 2009 R&D investment: string, 1 Yr R&D investment growth%: string, 4 Yr R&D inv
estment growth%: string, R&D investment / operation profit%: string, R&D investment / sales%:
string, R&D + capex / sales%: string, Operating profit: string, If made profit: string, 1 Yr
operating profit%: string, 4 Yr operating profit%: string, Operating profit / sales%: string,
Sales: string, 1 Yr sales growth%: string, 4 Yr sales growth%: string, Employees: string, 1 Y
r employees growth%: string, 4 Yr employees growth%: string, R&D / employees : string, 1 Yr R
&D / employees growth%: string, 4 Yr R&D / employees growth%: string, Sales outside region%:
string, Market cap: string, Market cap change%: string, If market cap increase: string, R&D s
pend 2008: string, R&D spend 2007: string, R&D spend 2006: string, R&D spend 2005: string]

In [27]: df.na.drop().count()

Out[27]: 149
```

### 3.3 Construct the data

Since the data is from official and historical sources, researchers cannot construct new data without convincing resources. Otherwise, it can influence data quality negatively. Regarding the cleaned data set, new attributes can be generated according to the data mining objective.

First, “Company ID” should be considered to derive from the data set as it may bother when using long company names in all data mining occasions. Then, two attributes of profit can be included. The first one is “If made profit”, it can help us to have a more intuitive acknowledgment about the company’s performance. Similarly, according to the data set, a categorical attribute “If market cap increase” can also be constructed to show if a company is “bigger” after certain input on R&D. The capture below shows the columns after construction.

```
print(TargetData.shape)
```

```
(1000, 37)
```

```
print(TargetData.columns)
```

```
Index(['Company', 'Company ID', 'Country', 'Normalised', 'Not IFR S',
      'Industry group', 'ICB code', 'Industry description', 'Listi
ng status',
      '2009 R&D investment', '1 Yr R&D investment growth%',
      '4 Yr R&D investment growth%', 'R&D investment / operation p
rofit%',
      'R&D investment / sales%', 'R&D + capex / sales%', 'Operatin
g profit',
      'If made profit', '1 Yr operating profit%', '4 Yr operating
profit%',
      'Operating profit / sales%', 'Sales', '1 Yr sales growth%',
      '4 Yr sales growth%', 'Employees', '1 Yr employees growth%',
      '4 Yr employees growth%', 'R&D / employees ',
      '1 Yr R&D / employees growth%', '4 Yr R&D / employees growt
h%',
      'Sales outside region%', 'Market cap', 'Market cap change%',
      'If market cap increase', 'R&D spend 2008', 'R&D spend 200
7',
      'R&D spend 2006', 'R&D spend 2005'],
      dtype='object')
```

### 3.4 Integrate various data sources

To address the business object better, it can be informative to append extend background data about the companies. A countries' GDP can be a criterion to reflect a company's competitive environment. As a result, the country's GDP data is integrated into the current data set. The data source of the countries' GDP is from the world bank data (<https://data.worldbank.org/>).

After integration, a new data set is generated for operations in the following steps. Compare with the raw data set, new fields and data has been added.

```
df.printSchema()
```

```
root
|-- Company: string (nullable = true)
|-- Company ID: string (nullable = true)
|-- Country: string (nullable = true)
|-- Country GDP $: string (nullable = true)
|-- Not IFRS: string (nullable = true)
|-- Industry group: string (nullable = true)
|-- ICB code: string (nullable = true)
|-- Industry description: string (nullable = true)
|-- Listing status: string (nullable = true)
|-- 2009 R&D investment: string (nullable = true)
|-- 1 Yr R&D investment growth%: string (nullable = true)
|-- 4 Yr R&D investment growth%: string (nullable = true)
|-- R&D investment / operation profit%: string (nullable = true)
|-- R&D investment / sales%: string (nullable = true)
|-- R&D + capex / sales%: string (nullable = true)
|-- Operating profit: string (nullable = true)
|-- If made profit: string (nullable = true)
|-- 1 Yr operating profit%: string (nullable = true)
|-- 4 Yr operating profit%: string (nullable = true)
|-- Operating profit / sales%: string (nullable = true)
|-- Sales: string (nullable = true)
|-- 1 Yr sales growth%: string (nullable = true)
|-- 4 Yr sales growth%: string (nullable = true)
|-- Employees: string (nullable = true)
|-- 1 Yr employees growth%: string (nullable = true)
|-- 4 Yr employees growth%: string (nullable = true)
|-- R&D / employees : string (nullable = true)
|-- 1 Yr R&D / employees growth%: string (nullable = true)
|-- 4 Yr R&D / employees growth%: string (nullable = true)
|-- Sales outside region%: string (nullable = true)
|-- Market cap: string (nullable = true)
|-- Market cap change%: string (nullable = true)
|-- If market cap increase: string (nullable = true)
```

### 3.5 Format the data as required

After data integration, the data set is well prepared. However, since the last step is finished on Numbers, a Mac OS spread application, the format of the data set is now “.paper”. So, it needs to be reformatted into CSV files form third-party software. After the operation, the data set is successfully converted and accessible from open-source software including Tableau and Python.

In pySpark, the data types in data frame can be converted by the function selectExpr. One of the fields converting result can be shown below.

```
df2 = df.selectExpr("cast(Sales as float) Sales")
df2.printSchema()
```

```
root
|-- Sales: float (nullable = true)
```

While in spark, the name of the field must to be converted into a single string, otherwise it may not be recognized by the library and fail to convert.

Rename process (notice: not all data are selected, only fields which are related to the datamining goals are invloved):

```
df2 = df.withColumnRenamed("Country GDP $", "GDP").withColumnRenamed("2009 R&D investment", "2009_")
df2.printSchema()
```

```
root
|-- Company: string (nullable = true)
|-- Company ID: string (nullable = true)
|-- Country: string (nullable = true)
|-- GDP: string (nullable = true)
|-- Not IFRS: string (nullable = true)
|-- Industry group: string (nullable = true)
|-- ICB code: string (nullable = true)
|-- Industry_description: string (nullable = true)
|-- Listing status: string (nullable = true)
|-- 2009_R&D_investment: string (nullable = true)
|-- 1_Yr_R&D_investment_growth%: string (nullable = true)
|-- 4_Yr_R&D_investment_growth%: string (nullable = true)
|-- R&D_investment / operation profit%: string (nullable = true)
|-- R&D_investment / sales%: string (nullable = true)
|-- R&D + capex / sales%: string (nullable = true)
|-- Operating_profit: string (nullable = true)
|-- If made profit: string (nullable = true)
|-- 1_Yr_operating_profit%: string (nullable = true)
|-- 4_Yr_operating_profit%: string (nullable = true)
|-- Operating_profit / sales%: string (nullable = true)
|-- Sales: string (nullable = true)
|-- 1_Yr_sales_growth%: string (nullable = true)
|-- 4_Yr_sales_growth%: string (nullable = true)
|-- Employees: string (nullable = true)
|-- 1_Yr_employees_growth%: string (nullable = true)
|-- 4_Yr_employees_growth%: string (nullable = true)
|-- R&D / employees : string (nullable = true)
|-- 1_Yr_R&D / employees growth%: string (nullable = true)
|-- 4_Yr_R&D / employees growth%: string (nullable = true)
```

Data reformatting, string types are converted into numeric double type:

```

df3 = df2.selectExpr("Company",
                    "cast(Sales as double) Sales",
                    "Country",
                    "cast(Operating_profit as double) Operating_profit",
                    "cast(Employees as double) Employees",
                    "cast(Market_cap as double) Market_cap")
df3.printSchema()

root
|-- Company: string (nullable = true)
|-- Sales: double (nullable = true)
|-- Country: string (nullable = true)
|-- Operating_profit: double (nullable = true)
|-- Employees: double (nullable = true)
|-- Market_cap: double (nullable = true)

```

Although the data set can be imported into data analyzing tools, the data type is still needed to check again. In Python, it shows that some of the columns which contain numeric data but have data type “Object”, this can cause an exception when conducting algorithm. Other tools can be used to conduct data formatting here and the results can be shown in the following screenshot.

```

print(TargetData.dtypes)

```

Company	object
Company ID	int64
Country	object
Country GDP \$	float64
Not IFRS	object
Industry group	int64
ICB code	int64
Industry description	object
Listing status	object
2009 R&D investment	float64
1 Yr R&D investment growth%	float64
4 Yr R&D investment growth%	float64
R&D investment / operation profit%	float64
R&D investment / sales%	float64
R&D + capex / sales%	float64
Operating profit	object
If made profit	object
1 Yr operating profit%	float64
4 Yr operating profit%	float64
Operating profit / sales%	float64
Sales	object
1 Yr sales growth%	float64
4 Yr sales growth%	float64
Employees	float64
1 Yr employees growth%	float64
4 Yr employees growth%	float64
R&D / employees	float64
1 Yr R&D / employees growth%	float64
4 Yr R&D / employees growth%	float64
Sales outside region%	float64
Market cap	object
Market cap change%	float64
If market cap increase	object
R&D spend 2008	float64
R&D spend 2007	float64

## Before the reformatting

```
[32]: TargetData["2009 R&D investment"] = TargetData["2009 R&D investment"].astype("float")
      TargetData["Operating profit"] = TargetData["Operating profit"].astype("float")
      TargetData["Sales"] = TargetData["Sales"].astype("float")
      TargetData["Market cap"] = TargetData["Market cap"].astype("float")

      print(TargetData.dtypes)
```

Company	object
Company ID	int64
Country	object
Country GDP \$	float64
Not IFRS	object
Industry group	int64
ICB code	int64
Industry description	object
Listing status	object
2009 R&D investment	float64
1 Yr R&D investment growth%	float64
4 Yr R&D investment growth%	float64
R&D investment / operation profit%	float64
R&D investment / sales%	float64
R&D + capex / sales%	float64
Operating profit	float64
If made profit	object
1 Yr operating profit%	float64
4 Yr operating profit%	float64
Operating profit / sales%	float64
Sales	float64
1 Yr sales growth%	float64
4 Yr sales growth%	float64
Employees	float64
1 Yr employees growth%	float64
4 Yr employees growth%	float64
R&D / employees	float64
1 Yr R&D / employees growth%	float64
4 Yr R&D / employees growth%	float64
Sales outside region%	float64
Market cap	float64

## After the reformatting

So far, the data set is cleaned, constructed, and integrated effectively, it is ready for modeling.

## 4. Data transformation

### 4.1 Reduce the data

According to a further experimental data mining process, it is found out that the description information in the initial data mining process is redundant, and still exist missing and error data. As a result, the data need to reduction horizontally and vertically to make the data set simple and more precise to address the data mining object. For example, “Company description” and “Not IFRS” are not considered as effective variables in the current stage of the data mining process.

Also, according to the logical process, fields with less relationship with the data mining object can be reduced. When tried to reduce the data, an error occurred:



```

-----
Py4JJavaError                                Traceback (most recent call last)
~/spark-2.1.1-bin-hadoop2.7/python/pyspark/sql/utils.py in deco(*a, **kw)
    62         try:
--> 63             return f(*a, **kw)
    64         except py4j.protocol.Py4JJavaError as e:

~/spark-2.1.1-bin-hadoop2.7/python/lib/py4j-0.10.4-src.zip/py4j/protocol.py in get_return_val
ue(answer, gateway_client, target_id, name)
    318         "An error occurred while calling {0}{1}{2}.\n".
--> 319         format(target_id, ".", name), value)
    320     else:

Py4JJavaError: An error occurred while calling o252.selectExpr.
: org.apache.spark.sql.catalyst.parser.ParseException:
mismatched input '&' expecting <EOF>(line 1, pos 40)

== SQL ==
cast(2009_R&D_investment as long) 2009_R&D_investment
-----^^^

197) at org.apache.spark.sql.catalyst.parser.ParseException.withCommand(ParseDriver.scala:

```

After trying to find out the reasons, I noticed that a kind of specific exceptions when using pySpark, which is caused by the field name. By inspecting the error, the spark lib is compiled by Java, and some special characters such as “&” and “%” are not permitted. These special symbols may cause runtime errors of data mining. So, in the later datamining tasks, pySpark is used as a tool but also other data mining tool will be considered to handle the data set. The reduced data set:

```

In [61]: df3 = df2.selectExpr("Company",
                             "cast(Sales as double) Sales",
                             "Country",
                             "cast(Operating_profit as double) Operating_profit",
                             "cast(2009_RD_investment as double) 2009_RD_investment",
                             "cast(Employees as double) Employees",
                             "cast(Market_cap as double) Market_cap")

df3.printSchema()

root
|-- Company: string (nullable = true)
|-- Sales: double (nullable = true)
|-- Country: string (nullable = true)
|-- Operating_profit: double (nullable = true)
|-- 2009_RD_investment: double (nullable = true)
|-- Employees: double (nullable = true)
|-- Market_cap: double (nullable = true)

```

As can be seen, the field name of “2009 R&D investment” is converted to “2009\_RD\_investment”.

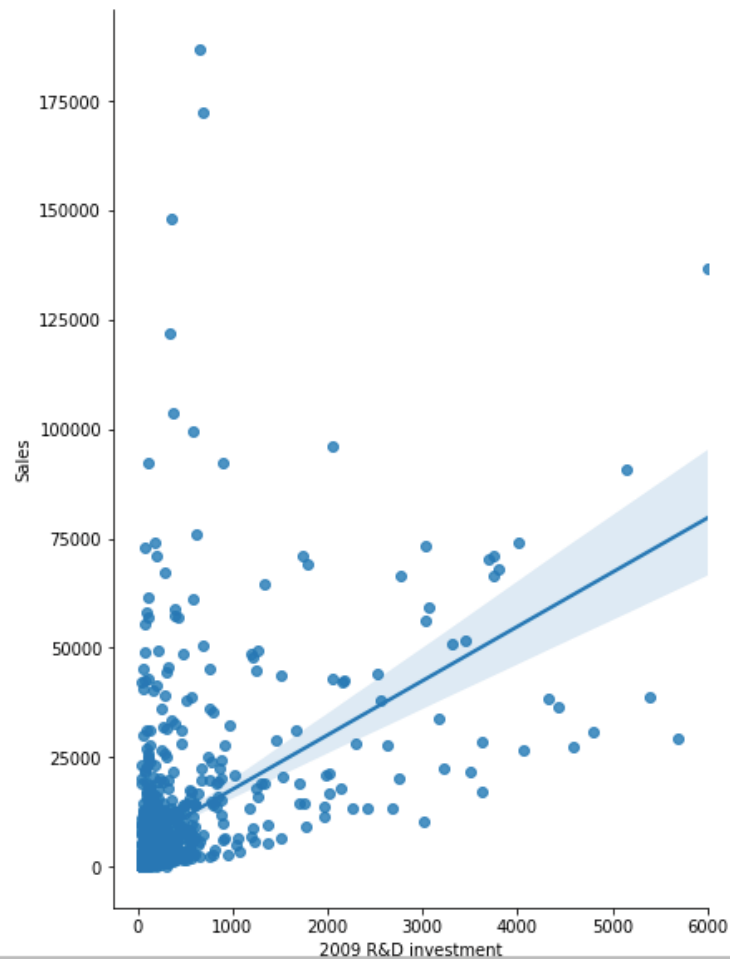
## 4.2 Project the data

Although the data can be imported successfully, however, it may not suitable for the algorithm. Further visualizations of data need to be done. Since the data

mining objects focus on the relation between R&D investment and Sales revenue, a plot diagram can be drawn by Python:

```
# Visuallization
```

```
sns.pairplot(TargetData, x_vars=['2009 R&D investment'], y_vars='Sales',kind="reg", height=8,plt.show())
```



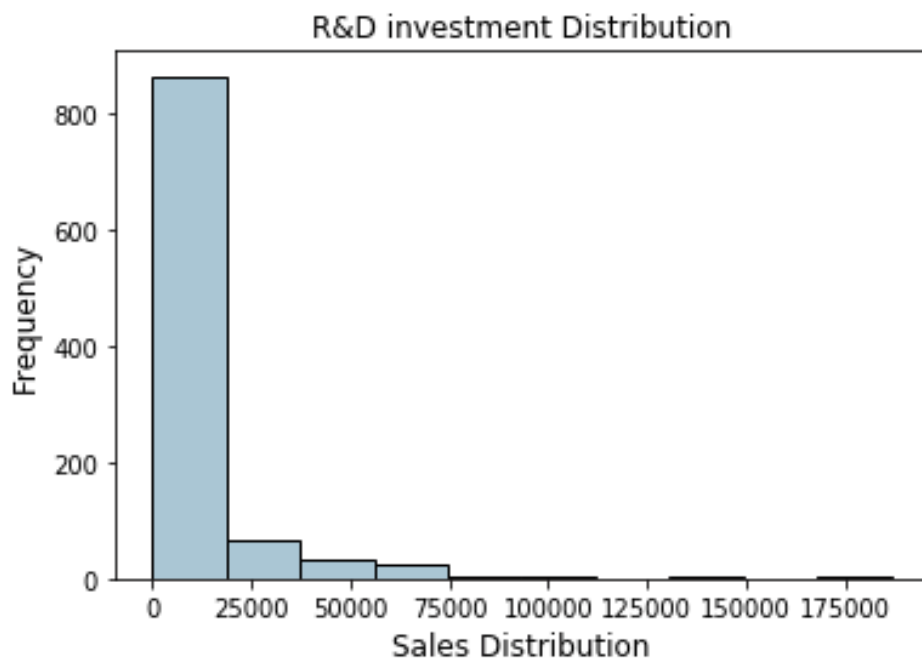
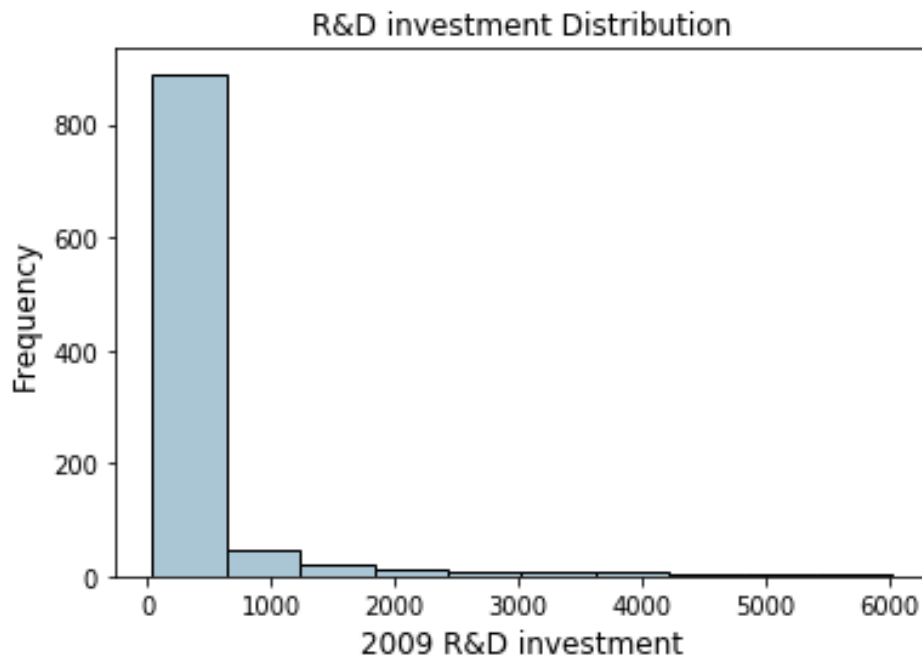
As can be seen, the data contains extreme values in both axes. If we look further on the distribution of both variables we can find:

```
# Not clear, more visulizaiton
```

```
fig, ax = plt.subplots()
TargetData['2009 R&D investment'].hist(color='#A9C5D3',edgecolor='black',grid=False)
ax.set_title('R&D investment Distribution', fontsize=12)
ax.set_xlabel('2009 R&D investment', fontsize=12)
ax.set_ylabel('Frequency', fontsize=12)

fig1, ax1 = plt.subplots()
TargetData['Sales'].hist(color='#A9C5D3',edgecolor='black',grid=False)
ax1.set_title('R&D investment Distribution', fontsize=12)
ax1.set_xlabel('Sales Distribution', fontsize=12)
ax1.set_ylabel('Frequency', fontsize=12)
```





Both of them have problems with skewed distribution. So, the data need to be projected by log transformation.

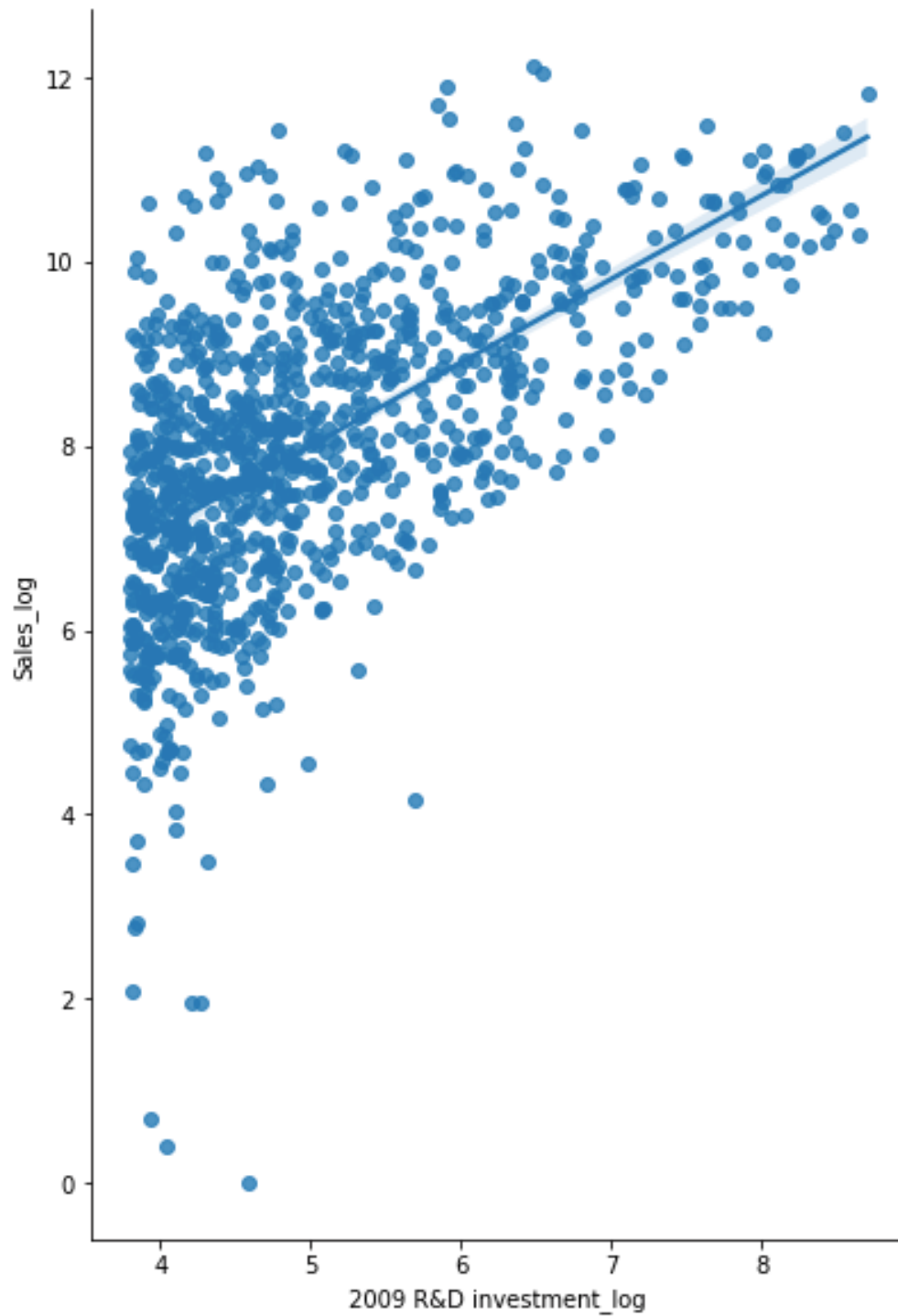
```
# Use log to project the data
logData = TargetData

logData['2009 R&D investment_log'] = np.log((1+logData['2009 R&D investment']))
logData[['Company ID', '2009 R&D investment', '2009 R&D investment_log']].iloc[4:9]

logData['Sales_log'] = np.log((1+logData['Sales']))
logData[['Company ID', 'Sales', 'Sales_log']].iloc[4:9]
```

	Company ID	Sales	Sales_log
4	303	506.0	6.228511
5	403	2574.0	7.853605
6	473	387.0	5.961005
7	885	4033.0	8.302514
8	421	55.0	4.025352

After the projection, the extreme values issue is addressed well as can be seen in the table. The correlation and plot diagram looks much better as well.



## 5. Data-mining method(s) selection

### 5.1 Match and discuss the objectives of data mining to data mining methods

According to the data mining objectives, a pattern of input and its influence on companies is the main target. By sufficient preparation on the data set, it is explicit that we have detailed data related in continuous form. In other words, the aim is to predict a numeric variable by using the information in the numeric type. It's can be key evidence in choosing a proper data mining method.

## 5.2 Select the appropriate data-mining method(s) based on discussion

Based on the discussion above, the regression method in supervised learning is selected as the data mining method in this case. The pattern can be generated by the regression algorithm so that the data mining outcome can be assessed correspondingly.

## 6. Data-mining algorithm(s) selection

### 6.1 Conduct exploratory analysis and discuss

As discussed above, regression is chosen as the data mining method. However, it needs to explore a variety of specific algorithms since there are multiple specific algorithms or models of regression. Larose (2015) suggested that according to different occasions and assumptions, proper regression models should be correctly chosen.

Since this is an academic practice and there are no practical restrictions like human capital or schedule, regression models are open to choose. The main factor that should consider is suitability. With a data set less than a million samples, the SDG regressor is waived as it's for data set that contains a large number of samples. Linear regression is considered as an initial algorithm in the data mining process. Then, a further and deeper exploration should be conducted. Because the data set has several columns, to expect a more accurate regression result, an algorithm like the random forest, neural network, and gradient boosting tree can be options in further process. Ridge regression is also a great regression model as it has an improvement on ordinary least squares (OLS) model. If there are issues of overfitting and multicollinearity, ridge regression and Lasso regression can be also chosen as optimizing algorithms (Hoerl et al., 1975).

### 6.2 Select data-mining algorithms based on discussion

Firstly, linear regression will be conducted to gain essential data mining outcomes. Then, multiple regression will be adopted in the model. By doing so, we expected a model with higher fitness and fewer errors. Finally, an exploration of random forest algorithm and ridge regression is planned to analyze.

### 6.3 Build/Select appropriate model(s) and choose relevant parameter(s)

The model building and parameters choosing are finished on the Python platform. The data set is reduced according to the chosen algorithm initially. Then the train and test set are generated by setting the ratio parameter “frac”. After all, it’s ready for the implementation of the algorithm.

```
# Linear Reg
# Remove the columns which have less correlation with target
columnsl = logData.columns.tolist()
columnsl = [c for c in columnsl if c not in ["Sales_log", 'Company', 'Company I
targetl = "Sales_log"]
```

```
# Construct train and test sets
train = logData.sample(frac=0.8, random_state=1)
test = logData.loc[~logData.index.isin(train.index)]

print(train.shape)
print(test.shape)

(800, 39)
(200, 39)
```

## 7. Data Mining

### 7.1 Create and justify test designs

Before the process of building the model, it’s necessary to evaluate again the result assessment. In order to have a convincing result, a train set and test set are built. As shown above, an 80-20 training/test split is conducted. Normally, a 70-30 ratio is accepted. However, in this case, the data set has an entire quantity of 1000 which is not relatively large. Putting 80% of the data in the training set can ensure the models’ effectiveness. By repeating the process by choosing different training and testing sets, the results can be more convincing (Mueller & Guido, 2016).

### 7.2 Conduct data mining

Initially, a linear regression model is conducted. Since the data type and null, missing error issues were addressed in the data preparation step, the process runs well with no exceptions or errors reported. Calculation inconsistencies are also avoided due to sufficient data preparation. Processing time is also reasonable as the size of data is not extremely large. The mean squared error (MSR) is set as the measurement of the prediction result. It can be easily understood that low MSR represents a better fitting.

```
# Reg process and error measurement  
  
model = LinearRegression()  
model.fit(train[columnsl], train[targetl])  
predicitons = model.predict(test[columnsl])  
print(mean_squared_error(predicitons, test[targetl]))
```

1.9461604469442058

In Python, the linear algorithm is successfully conducted and the MSE is at a relatively low level. An acceptable outcome is achieved.

Then, the second algorithm will be implemented. In the multi-regression model, more independent variables are included in the regression model. By setting the same ratio of training and testing set, the algorithm is also conducted successfully without exception.

```

: # E1: Reg process and error measurement

Elmodel = LinearRegression()
Elmodel.fit(E1train[E1columnsl], E1train[E1targetl])
predicitons = Elmodel.predict(E1test[E1columnsl])
print(mean_squared_error(predicitons, E1test[E1targetl]))

a, b = Elmodel.coef_, Elmodel.intercept_
print(a,b)
print(E1train[E1columnsl])
# As can be seen, the result is better with more related attributes in

```

```

1.4781840190269646
[ 1.42915488e-05 -2.16614228e-06 -3.33937841e-07 -7.53134115e-04
 8.48902773e-01] 4.029547301671737

```

	Operating profit	Employees	Market cap	R&D spend 2008 \
340	122.0	10000	2065.0	87.39
810	5149.0	19835	95983.0	1729.64
473	6085.0	30000	50539.0	129.42
56	114.0	1600	2303.0	44.33
617	431.0	34900	9571.0	261.32
..	...	...	...	...
137	422.0	26000	6891.0	53.25
536	-67.0	26803	1357.0	165.40
646	791.0	58600	7905.0	296.76
628	170.0	29500	47.0	268.75
212	-3.0	4355	902.0	63.40

	2009 R&D investment_log
340	4.769922
810	5.144292
473	4.363608
56	4.093678
617	4.875107

As can be seen, the result of the multi-regression model shows an MSE of 1.478 which is lower than the former one. It indicates that this result has better fitness.

Afterward, the random forest algorithm is conducted. Random forest algorithm has an idea of bagging which means combine multiple decision trees in determining the final output (Mueller & Guido, 2016). For those data which shows less linear relationship globally but partly related, this algorithm can reveal it appropriately. The following snapshot shows the process of conducting the model.

```
# Exploration2: Random forest algrithm
# Optimize the algorithm
# adjust the value of min_samples_leaf=40

E2modelR = RandomForestRegressor(n_estimators=100, min_samples_leaf=40, random_s
E2modelR.fit(E1train[E1columnsl], E1train[E1targetl])
predictionsByRF = E2modelR.predict(E1test[E1columnsl])
print(mean_squared_error(predictionsByRF, E1test[E1targetl]))

# It seems not more effective in this case
```

1.5984971495880627

The parameter “min\_sample\_leaf” indicates the minimum number of samples for each leaf nodes. A lower value makes the model reflect well of noises in data (Srivastava, 2015). After several attempts, the parameter value is set to “40” and get the prediction result with MSE of 1.59 subsequently. It is acceptable as a qualified outcome because its MSE is lower than simple linear regression model.

By using pySpark furthermore, model data is initially selected. Based on previous big data analysis, proper subset can be chosen and selected.

```
model_data = output.select("features", "Sales")
model_data.show()
```

```
+-----+-----+
|          features|    Sales|
+-----+-----+
|[2439.0,228.89,37...| 7321.0|
|[3592.0,196.04,44...| 9175.0|
|[2775.0,190.47,38...| 9343.0|
|[3661.0,785.25,43...|14205.0|
|[668.0,173.54,103...| 2574.0|
|[97.0,53.74,1888....|  387.0|
|[237.0,102.7,1110...| 4033.0|
|[-29.0,58.9,403.0...|   55.0|
|[130.0,43.76,1304...| 2841.0|
|[308.0,96.67,4164...| 7597.0|
|[1247.0,75.52,272...| 5441.0|
|[-2471.0,67.53,56...| 4718.0|
|[1115.0,47.09,168...| 5262.0|
|[306.0,598.85,932...| 2769.0|
|[6355.0,98.46,116...|22762.0|
|[146.0,134.16,115...| 2448.0|
|[-65.0,54.36,3310...|  567.0|
|[236.0,56.36,1810...| 2166.0|
|[683.0,502.89,282...| 7539.0|
|[121.0,120.57,971...| 6177.0|
+-----+-----+
only showing top 20 rows
```

Then, train and test data set are divided.

```
train_data,test_data = model_data.randomSplit([0.8,0.2])
```



```
: train_data, test_data = model_data.randomSplit([0.8, 0.2])
train_data.show()
```

```
+-----+-----+
|          features |    Sales |
+-----+-----+
| [-4325.0, 6013.74, ... | 136559.0 |
| [-2599.0, 3445.06, ... | 51655.0 |
| [-2595.0, 559.0, 19... | 38690.0 |
| [-2471.0, 67.53, 56... | 4718.0 |
| [-2223.0, 2516.12, ... | 44265.0 |
| [-2221.0, 1459.8, 1... | 29106.0 |
| [-2181.0, 331.91, 7... | 6513.0 |
| [-1842.0, 1227.52, ... | 5602.0 |
| [-1759.0, 3029.78, ... | 56121.0 |
| [-1683.0, 257.66, 1... | 36040.0 |
| [-1646.0, 590.13, 4... | 6072.0 |
| [-1520.0, 3307.93, ... | 50776.0 |
| [-1413.0, 1300.6, 5... | 18939.0 |
| [-1355.0, 117.28, 1... | 42849.0 |
| [-1220.0, 2055.98, ... | 43018.0 |
| [-1207.0, 2770.59, ... | 66521.0 |
| [-1181.0, 214.47, 5... | 2202.0 |
| [-1122.0, 412.85, 4... | 7674.0 |
| [-1058.0, 437.07, 1... | 3245.0 |
| [-1057.0, 59.26, 51... | 4410.0 |
+-----+-----+
only showing top 20 rows
```

Then, the data model can be conducted. Result can be shown as follow:

```
In [86]: print("Coefficients: {} Intercept: {}".format(lrModel.coef, lrModel.intercept))

Coefficients: [2.1667281934710645, 2.78137712307252, 0.11388333781690699, 0.2203143616168274] In
tercept: 184.02481116980144
```

After conducting the data mining process, a test result can be operated. RSME returned as an assessment of the model.

```
test_results.residuals.show()

print("RSME: {}".format(test_results.rootMeanSquaredError))
```

```
+-----+
| residuals |
+-----+
| 9582.9728288142 |
| 8057.155024593623 |
| 5084.750841164885 |
| 2405.3030640164357 |
| 4363.8952755482815 |
| 3352.949451407898 |
| 4885.680632894415 |
| 2658.886134866456 |
| 136.57619796838708 |
| 286.1448485644603 |
| 1276.8853721600867 |
| 8213.57484355157 |
| 415.7266239176206 |
| -1783.1804660854386 |
| -177.37302509661276 |
| 3125.0699703557884 |
| -569.2364175049333 |
| 25.06624752982492 |
| 698.3064807706432 |
| -235.56931989326358 |
+-----+
only showing top 20 rows

RSME: 8640.848968645289
```

### 7.3 Search for patterns

After the implementation of the algorithm, the specific pattern of result can be calculated by using the Python packages “pandas” and “numpy”. For linear regression, the coefficient value and interception can be returned by using the following sentences.

```
# Reg process and error measurement

model = LinearRegression()
model.fit(train[columnsl], train[targetl])
predicitons = model.predict(test[columnsl])
print(mean_squared_error(predicitons, test[targetl]))

a, b = model.coef_, model.intercept_
print(a,b)

1.9461604469442058
[0.88963582] 3.5492783501535055
```

So, the linear regression model formula can be drawn as the data mining pattern. Similarly, the second pattern is also successfully collected by executing certain sentences.

According to the random forest algorithm, the pattern result cannot be shown directly because the outcome contains many decision trees. The pattern of it can be treated as a combination of parameters.

In spark, the pattern can also be shown explicitly as below:

```
Coefficients: [2.1667281934710645,2.78137712307252,0.11388333781690699,0.2203143616168274] In  
tercept: 184.02481116980144
```

By knowing the number of corresponding coefficients and intercept, a linear regression model can be collected.

## 8. Interpretation

### 8.1 Study and discuss the mined patterns

Regarding the linear regression algorithm model, it is a classic method to predict numeric data. In our case, the correlation analysis is presented, and the mined pattern is obtained. As can be seen from above, the coefficient value is approximately 0.89, and interception is 3.55, and the evaluated regression line can be drawn by having these parameters. The coefficient value represents the strength of the independent variable's influence on the dependent variable. While 0.89 is relatively high and with the MSE of 1.94, it's a significant pattern that reflects the quantitative relationship between R&D investment and sales revenue after log projection.

In the second regression, the model contains multiple factors and results in a more complex regression pattern. By which we found that the coefficient value of it is exceedingly low. For instance, the operating profit's coefficient value is 1.42915488e-05 which is 0.000142 in decimal. However, the value is relatively low, it can be interpreted into insight into the character of the multiple regression model. The model is conducted including 5 independent variables and they have different measurement units. Besides the dependent variables in this model are log projected, they are conducted in the same model in which the algorithm treated them in different values without considering units (Larose, 2015). Consequently, the outcome patter may contain factors with low value. In this case, the access criteria MSE can tell the effectiveness of this model. Because there are more dimensions to describe the dependent variable, the fitness and prediction ability are improved compared with the first model.

In the random forest algorithm, the basic idea is to build a set of decision trees and make them block of “decision forest”. By running them parallelly at training time, the mean prediction of individual trees will be returned (Chakure, 2019). In the pattern, we can gain a set of the parameter value. To understand the random forest algorithm deeper, we have to know the meaning of parameters. The parameter “n\_estimators” represents the number of trees the model plans to build. More trees make the predictions stronger and convincing (Srivastava, 2015). The value of “min\_sample\_leaf” reflects the end node number of a single decision tree. The fewer nodes are more flexible and capable to catch the noises in the data set. After tuning these parameters, the model shows a better performance although it takes a longer time.

```
# Exploration2: Random forest algrithm
# Optimize the algorithm to 1

E2modelR = RandomForestRegressor(n_estimators=1000, min_samples_leaf=1, random_s
E2modelR.fit(E1train[E1columns], E1train[E1target])
predictionsByRF = E2modelR.predict(E1test[E1columns])
print(mean_squared_error(predictionsByRF, E1test[E1target]))

# It seems not more effective in this case
```

1.289031674690822

The MSE is reduced by 1.2. By reusing it with a combination with other data set or individual rows of data, it can predict the form of what parameters limited.

In pySpark, the result can be also gained as follows:

```
print("R2: {}".format(test_results.r2))
```

R2: 0.6879419956752286

The specific outcomes of the linear regression can be gain through the pySpark lib.

```
trainingSummary = lrModel.summary
```

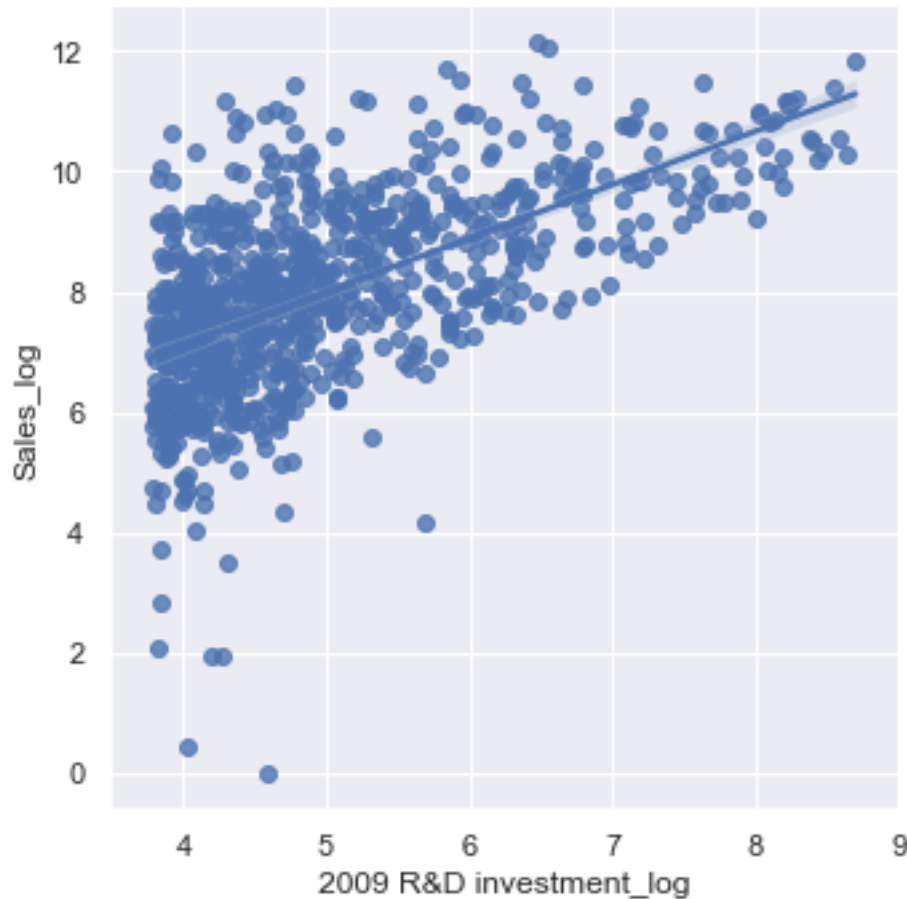
```
print("Deviance Residuals: " +str(trainingSummary.devianceResiduals))
print("Coefficient Standard Errors: " +str(trainingSummary.coefficientStandardErrors))
print("Explained Variance: " +str(trainingSummary.explainedVariance))
print("Features Col: " +str(trainingSummary.featuresCol))
print("Label Col: " +str(trainingSummary.labelCol))
print("Mean Absolute Error: " +str(trainingSummary.meanAbsoluteError))
print("Mean Squared Error: " +str(trainingSummary.meanSquaredError))
print("Num Instances: " +str(trainingSummary.numInstances))
print("Objective History: " +str(trainingSummary.objectiveHistory))
print("PValues: " +str(trainingSummary.pValues))
print("Prediction Col: " +str(trainingSummary.predictionCol))
print("R2: " +str(trainingSummary.r2))
print("Root Mean Squared Error: " +str(trainingSummary.rootMeanSquaredError))
print("TValues: " +str(trainingSummary.tValues))
print("Total Iterations: " +str(trainingSummary.totalIterations))
```

```
Deviance Residuals: [-47793.25395672888, 108275.00667536427]
Coefficient Standard Errors: [0.3188040680694874, 0.7248652755766982, 0.0075625832853041175,
0.04436131156257767, 521.6065436926772]
Explained Variance: 266081661.49671704
Features Col: features
Label Col: Sales
Mean Absolute Error: 4711.828761660059
Mean Squared Error: 124023729.85901089
Num Instances: 675
Objective History: [0.0]
PValues: [2.3733903731226746e-11, 0.00013627048195874458, 0.0, 8.667040143350846e-07, 0.72434
64371747206]
Prediction Col: prediction
R2: 0.6820763501165852
Root Mean Squared Error: 11136.594176812356
TValues: [6.796425800309419, 3.837095273821296, 15.058787919494277, 4.9663626672995, 0.352803
87754917836]
Total Iterations: 1
```

The results are similar with the former process, which shows acceptable outcome. The value of coefficient standard errors is relatively low on average which means the result is effective. Moving on to the prediction outcomes, R square is 0.68, which means almost 70% of the data can be predicted, or be explained, by the model. The model has a capacity to address the data mining problems and makes prediction.

## 8.2 Visualize the data, results, models, and patterns

The visualization of models can be done by utilizing Python seaborn packages. The simple linear regression model:



Multiple regression model visualization can be conducted into different diagrams according to different independent variables.



Since the random forest algorithm is operated by a large amount of decision trees, the visualization is not suitable and necessary in the model.

### 8.3 Interpret the results, models, and patterns

As the data mining algorithm proceeds, outcomes and results are meaningful and corresponding. They can address the data mining object significantly. From

the linear regression model, the pattern shows an obvious result of the quantitative relationship between R&D investment and sales revenue. It simply tells they contain a slightly strong positive correlation; input of R&D will benefit companies in most cases. However, the data mining objective is not fully covered as we need to predict companies' performance by giving several data attributes.

Adding factors "Operating profit", "Employees", "Market capital" and R&D input of the former period, the model becomes practically usable. By giving a company's information above, the company's future performance can be predicted. Also, it can be used reversely by giving factors and expect the quantity of R&D investment. In the second regression model, the pattern is acquired. One thing should be noticed is that the data is projected in previous steps. The reason for doing that to adjust their distributions. When the pattern is conducted in real, the input data should be projected as same as the model and the outcomes also need to be transformed back.

The random forest algorithm is conducted after well concerned. It is not biased since there are multiple decision trees while in operation and each of them is trained on a subset of data. It can reduce unfairness effectively (Muller and Guido, 2016). Also, this algorithm is suitable for data that is not scaled well (Malik, 2019). In our case, it performs well and has a result with the least MSE. It can be regarded as the most suitable model. By having the combination of parameters, it is outstanding on giving a relationship between variables in this case and it also can be well used in intervention prediction. The business objectives can be well addressed.

#### 8.4 Assess and evaluate results, models, and patterns

In this case of data mining, the results' effectiveness is addressed by MSE. MSE reflects the average squared difference between the estimated values and the training values. Lower MSE represents for stronger effectiveness and significance. Specifically, the MSE of linear regression has proceeded after the operation of the regression model, the value of it is 1.94. While in multi-regression, the MSE reduced by 1.47 which indicates a better model according to the data set. After tuning parameters, the random forest algorithm gets an MSE of 1.28 as a result, it can be regarded as the most effective model in this case.

#### 8.5 Iterate prior steps (1 – 7) as required



The iteration step can be done by recondct the model by different train and data set. After a few attempts, it can be found that models have a well able to predict the data and keep the MSE at a low level meanwhile.

```
: # E1: Construct train and test sets
E1train = logData1.sample(frac=0.85, random_state=1)
E1test = logData1.loc[~logData1.index.isin(train.index)]

: # E1: Reg process and error measurement

E1model = LinearRegression()
E1model.fit(E1train[E1columns], E1train[E1target])
predicitons = E1model.predict(E1test[E1columns])
print(mean_squared_error(predicitons, E1test[E1target]))

a, b = E1model.coef_, E1model.intercept_
print(a,b)
print(E1train[E1columns])
# As can be seen, the result is better with more related attributes in

1.4796228391621873
```

While including a new algorithm model, ridge regression, is also a valid way to iterate the data mining process. Initial steps including situation understanding and data preprocessing. Ridge regression is an optimized version of the ordinary least square method and it's suitable in this case considering the former outcomes. The model is conducted as below.

```
# Exploration3: Ridge Regression
# Data prepration

E3columns = logData.columns.tolist()
E3columns = [c for c in E3columns if c not in ["Sales_log", 'Employees', 'Market cap', 'Company', 'Company ID',
E3target = "Sales_log"

poly=PolynomialFeatures(3)
#E3columns=poly.fit_transform(E3columns)

E3train = logData.sample(frac=0.8, random_state=1)
E3test = logData.loc[~logData.index.isin(train.index)]

clf=Ridge(alpha=1.0,fit_intercept = True)
clf.fit(E3train[E3columns], E3train[E3target])
clf.score(test[E3columns],test[E3target])

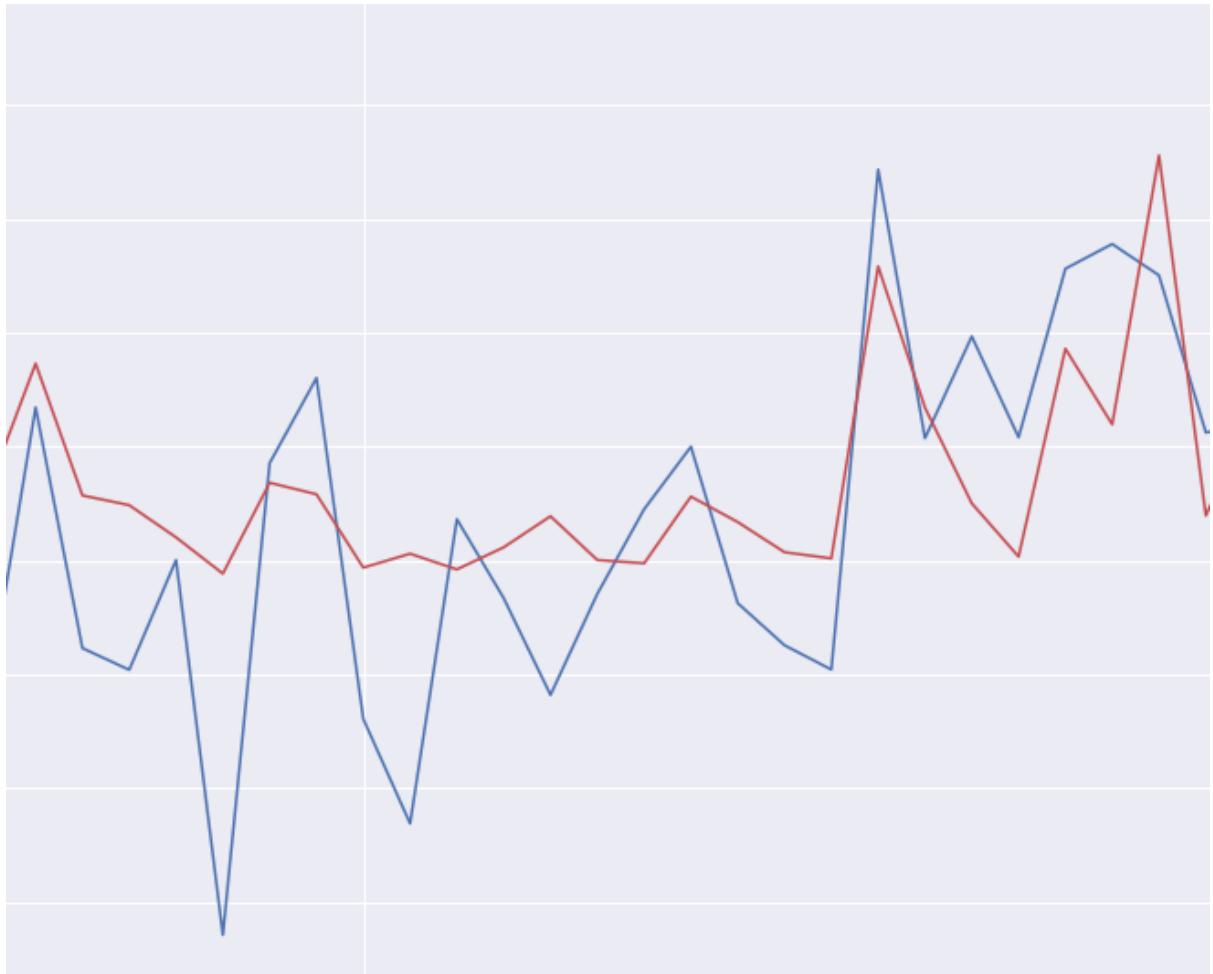
0.42280725688104903
```

By assessing the outcome, ridge regression uses the “score” as an indicator. According to the documentation, “score” represents the value of R square. It implies that 42% of the change is explained by the model. Considering the data set’s complexity, the result is acceptable. The visualization of the result can be



drawn in Python.





As can be seen, the model does not perform well when the change of input swings vigorously. To make the model performs better, the data can be reduced accordingly and transformed in a reasonable interval (Larose, 2015).

Finally, a prediction of spark linear regression model can be conducted as iteration. The value of predicted sales can be shown in the table below. By reflecting the data mining object, the prediction process can be successfully operated.

```

predict_model = lr.fit(test_data)
test_results = predict_model.evaluate(test_data)
test_results.predictions.show(10)

```

features	Sales	prediction
[-4139.0,113.73,6...]	9727.0	-5055.002249873933
[-1772.0,2305.05,...]	28041.0	13196.948462855447
[-1562.0,1343.43,...]	18938.0	7117.2139763285395
[-1251.0,58.41,39...]	1014.0	-2813.7622470495385
[-1183.0,660.15,1...]	5833.0	-2001.4562657553784
[-1005.0,156.67,2...]	40318.0	40000.664442029716
[-902.0,104.65,59...]	11608.0	6778.187502537703
[-618.0,45.39,123...]	3364.0	876.5156684784098
[-437.0,56.16,320...]	3686.0	4522.337622857567
[-365.0,54.8,1500...]	1719.0	2233.0975923063816

only showing top 10 rows

## Reference

- Cancino, C. A., Paz, A. I., Ramaprasad, A., & Syn, T. (2018). Technological innovation for sustainable growth: An ontological perspective. *Journal of Cleaner Production*, 179, 31-41. doi:10.1016/j.jclepro.2018.01.059
- Solow, R. M. (1956). A Contribution to the Theory of Economic Growth. *The Quarterly Journal of Economics*, 70(1), 65-94.
- Romer, P. (1989). Endogenous Technological Change. doi:10.3386/w3210
- Vadim, K. (2018). Overview of different approaches to solving problems of Data Mining. *Procedia Computer Science*, 123, 234-239. doi:10.1016/j.procs.2018.01.036
- Olafsson, S. (2003). Focused issue on operations research and data mining. *Computers & Operations Research*, 30(13), 2079-2080. doi:10.1016/s0305-0548(03)00174-6
- Sculley, D., & Pasanek, B. M. (2008). Meaning and mining: The impact of implicit assumptions in data mining for the humanities. *Literary and Linguistic Computing*, 23(4), 409-424. doi:10.1093/lilc/fqn019
- Freitas, A. (1996). A framework for data-parallel knowledge discovery in databases. *IEE Colloquium on Knowledge Discovery and Data Mining*. doi:10.1049/ic:19961111
- Li, H., & SAS Data Science Blog. (2017, April 12). Which machine learning algorithm should I use? Retrieved May 15, 2020, from <https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/>
- Larose, D. T. (2015). *Data mining methods and models*. Hoboken, NJ: Wiley-Interscience.
- Hoerl, A., Kennard, R., & Baldwin, K. (1975). Ridge Regression: Some Simulations. *Communications in Statistics - Simulation and Computation*, 4(2), 105-123. doi:10.1080/03610917508548342
- Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: A Guide for Data Scientists*. Beijing: O'Reilly et Associates.
- Srivastava, T. (2019, September 04). Tune a Random Forest model's parameters for Machine Learning. Retrieved May 15, 2020, from <https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/>
- Chakure, A. (2020, April 07). Random Forest and its Implementation. Retrieved May 15, 2020, from <https://towardsdatascience.com/random-forest-and-its-implementation-71824ced454f>
- Srivastava, T. (2019, September 04). Tune a Random Forest model's parameters for Machine Learning. Retrieved May 15, 2020, from <https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/>

Malik, U. (2019). Random Forest Algorithm with Python and Scikit-Learn. Retrieved May 15, 2020, from <https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/>

## Disclaimer

I acknowledge that the submitted work is my own original work in accordance with the University of Auckland guidelines and policies on academic integrity and copyright.

I also acknowledge that I have appropriate permission to use the data that I have utilised in this project. (For example, if the data belongs to an organisation and the data has not been published in the public domain then the data must be approved by the rights holder.) This includes permission to upload the data file to Canvas. The University of Auckland bears no responsibility for the student's misuse of data.