UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

BIG DATA

(6CS030)

Coursework Element 2: Time Constrained Assessment

Student Id : 1928579

Student Name : Shreejan Shrestha

Group : C3G1

Supervisor : Mr. Rupak Koirala | Jnaneshwar Bohara

Cohort : 3

Submitted on : 2020/05/29

Table of Contents

Table of Figures

Report

Introduction to data quality

Set of data is considered as a high quality if it can fit, represent and provide the real-world related information in accurate way. Because of the presence of the redundancy from various sources, spelling errors, invalid and missing information while entering the data, the quality of data gets reduced and hinders the consistency and accuracy of better results. It is the fundamental requirement to have good quality of data in order to avoid ambiguous information and wrong interpretation. Quality data will always lead to information gain and good clarification of data (Rahm & Do, 2000).

Summary of Key features relating to data quality in:

Rahm paper

Data are always imported and integrated from various sources which have the higher possibilities of redundant and invalid data, errors of spellings, missing values, etc. resulting in inconsistence results. In order to maintain the quality of data, data must be taken through data cleaning process which includes detecting and removing the errors from both individual sources and when merging numerous sources. In order to avoid misleading output and wrong interpretation, data warehouses are required, which provides support to handle and filter huge amount of data. As the data are imported from the multiple sources there is always high possibilities of data inconsistence because of which maintaining the quality of data is always challenging. For cleaning the data, there are various tools available but also some of data transformation work are always needed to be done manually. Before forming the cluster of data in the data warehouses, data must go through the process of extraction, translation & integration, filtering and loading. Data should be cleaned and transformed in such a reliable way that it will be reusable for other sources and would not create much difficulties while querying the data.

The quality of data in single-source depends on the quality and level of data controlled by schema, maintaining the integrity constraints. If the data are lacked of schema, restriction on entering and storing of data will occur because of the high possibilities of irregularities.

Schema-related problems occurs if the scheme has been poorly designed whereas instance-specific problem is related to data entry errors. As the sources of data are developed independently, the data cleaning process becomes troublesome when integrating from multi-source, as data are characterized differently or overlap. Mostly, schema-related problems occur because of naming conflicts when same term is used for different objects and vice-versa whereas instance problems occurs with the different types, representation and interpretation of data of various sources (Erhard & Hong, 2000).

Kim paper

Because of the increase of transfer of large amount data and information and its increasing storing capacity, the demand of data warehouse is also increasing. To generate the précised result, quality of data must be taken into consideration. Data comes from multiple sources hence will have many errors in the form of dirty data. In the given article, “A Taxonomy of Dirty Data”, author has defined dirty data as, “Any missing or wrong data and redundant data are dirty data”. Dirty data can lead to misleading information that is why, every data must be cleaned or removed if it cannot be corrected by any measures. Dirty data can get generated by uncountable ways. Some of them are due to typing errors, misinterpretation of data or can be because of bugs while processing the data. Author has defined three types of dirty data. First is data missing, second is wrong data and last one is unusable data.

Author has shown the confused definition for Null data and whether to consider it as missing data or not depends on the situation. But even after knowing the value, if it is not replaced then it will be considered as a dirty data. Wrong data can be considered as those types of data that is different from the true value during the time of process. Example for wrong data can be in the form of using character, strings in place of int or vice-versa. Unusable data usually occurs when we merge data from multiple of sources or data representation are not in consistent flow while entering data. Example, if the salary of on person is 1000 in one database and 100000 in other, it will create a confusion while we join in two databases. Techniques to cover and quantify most of the dirty data taxonomy should be developed to measure the excellence of data in the datasets (KIM, 2003).

Conclude with a discussion of whether they agree or disagree

After reading both articles, they both agree with each other's opinion regarding dirty data and the definition of dirty data is similar in some way. Both articles have marked the issue in the data and focused in dirty data, types or forms of dirty data and measures to clean the data in order to obtain the best result. Both the author has spoken about the importance of data warehouse because of the generation of large amount of data in daily basis. Both papers have discussed the possible issue and errors that can occur in the data and have clearly interpret the issues and their solutions in their own way. Though the methods and process of identifying and analyzing the dirty data are not exactly same but have the precisely same aim which is handling the dirty data and ways to clean the data.

Top 3 data quality issues

1. Inappropriate information

These are some of the information which do not contribute directly while analyzing the data. Rather these kinds of data create hurdle while processing the data in DBMS. In the figure below, we can see the clear example of inappropriate information present in the dataset. These kinds of data will not have any definite structure and will not get fit into any attributes.

Ebola 2014-2016									
ONS Kaggle Copyright Reserved [from Nomis on 14 feb December 2017]									
SEX									
age	Total								
	Aged-19-75								
Country	Date	Suspected_cases	Probable_cases	Confirmed_cases	Total_cases	Suspected_deaths	Probable_deaths	Confirmed_deaths	Total_deaths
Guinea	8/29/2014	25	141	482	648	2	141	287	430
Guinea	8/29/2014	3	1	15	19	0	1	6	7

Figure 1 Inappropriate Information

In above figure, from row 1 to 6 is an inappropriate information and one of the data qualities issues.

2. Characters and symbols

Use of characters and symbols are always the best example of dirty data. It can be present in any part of dataset from header to in any rows. Characters like “!”, “#”, “/”, space are the common examples.

16	Guinea	9/5/2014	56	152	604	812	3	152	362	517
17	Guinea	9/5/2014	!	!	1	1	!	!	0	0
18	Guinea	9/8/2014	2	0	1	3	0	0	0	0
19	Guinea	9/8/2014	47	151	664	862	4	151	400	555

Figure 2 Use of character inside the dataset

Use of characters is invalid at any place and hence throw an error message. It is not identified as null value or does not provide any specific details that is why, it is also data quality issue.

3. Data quality issue while integrating from multiple source

While the data from multiple sources are integrated, the problem of redundancy, data representation might be different can occur.

Customer (source 1)

CID	*Name*	*Street*	*City*	*Sex*
11	Kristen Smith	2 Hurley Pl	South Fork, MN 48503	0
24	Christian Smith	Hurley St 2	S Fork MN	1

Client (source 2)

Cno	*LastName*	*FirstName*	*Gender*	*Address*	*Phone/Fax*
24	Smith	Christoph	M	23 Harley St, Chicago IL, 60633-2394	333-222-6542 / 333-222-6599
493	Smith	Kris L.	F	2 Hurley Place, South Fork MN, 48503-5998	444-555-6666

Figure 3 Integrating two data from two different sources (Rahm & Hong, 2000)

In above figure, same thing has been represented in two different way. In source 1, attribute ‘Sex’ has been used whereas in source 2, attribute ‘Gender’ has been used for denoting same data. This type of issue will generate the confusion while analyzing the data after integrating the data.

Sample Data

Document the problem in the data:

1. The column HIREDATE is in object type.

In employees table, column ‘HIREDATE’ is in object type, which means it is stored as a string format. But, as it is representing date, it must be in datetime format.

This can be considered as an issue, as it is in string format, if we have to analyze the ‘HIREDATE’ COLUMN on the basis of date and time, it will concatenate the data because it is in string format and makes it difficult to calculate the result on the basis of datetime.

To solve this issue, we can change the data type of the column ‘HIREDATE’ from type object to datatime type using pandas **.to_datatime** function.

2. Value error in ‘COMMISSION_PCT’ column.

The COMMISSION_PCT column has the float64 as a data type but if we analyze the ‘COMMISSION_PCT’ column in employee table, there are couple of int values, which carries large amount.

This can be considered as an issue, as the entire column carries the value of commission in the form of percentage, if the int value is ignored, it can result in wrong interpretation and wrong calculation because percentage cannot be more than 100. Hence, it will give the misleading output.

To solve this issue, as it is the form of invalid data/ data type error, the int value can be converted into percentage or it must be dropped. We can drop using pandas’s **df.drop(‘row_index’,axis=0)** function.

3. Formatting error.

In the JOB_ID column, sales representative has been abbreviated as ‘**SA_REP**’. In the entire column of JOB_ID, one row has been misspelled as ‘**SALES_REP**’ in place of ‘**SA_REP**’ because of which, one entry from the sales representative has been recorded separately.

This can be considered as an issue. If we sum the total commission got by all the sales representative, one record will not get added and will get counted separately as 'SALES_REP'.

To solve this issue, the misspelled value can be updated from 'SALES_REP' to 'SA_REP' by using pandas **.replace** function. Below is the pandas 's code.

```
dataset['JOB_ID'].replace('SALES_REP', 'SA_REP', inplace=True)
```

4. Typo error.

Similar to JOB_ID column, DEPT_ID also has one type error. In the department table, primary key for Sales representatives is 80. As it is a primary key of department table, it has been used as a foreign key in employees table. But in employees table, one of the sale representatives has been recorded with wrong sales representative's id i.e. 85. The id of sales representative has been mistyped to 85 in place of 80.

This can be considered as an issue because, as he has been assigned a wrong sales id, his work will not be recorded in his working department.

To solve this issue, the mistyped value can be replaced by the correct sales id by using pandas .replace function. Below is the pandas 's code

```
dataset['DEPT_ID'].replace(85,80, inplace=True)
```

5. Missing value

In employee table, DEPT_ID column has one missing value in 80th row.

This can be considered as an issue because missing data can affect the statistical analysis process and can create bias in the dataset.

To solve this issue:

As we know, in employee table, 'DEPT_ID' column is a foreign key. Though 'DEPT_ID' has not been assigned to the 80th row, if we see the value of 'JOB_ID' column for the same row, 'SA_REP' has been assigned. As the id for every 'SA_REP' is 80, we can replace the missing value with 80 considering this logic. We can replace the missing value with known value directly in excel. Using query or code just to replace one missing value will be tedious steps.

6. Unknow string format and invalid data format in ‘HIREDATE’ column.

The ‘HIREDATE’ column have invalid date like **31-APR-2005** and **30-feb-05** and unknown string format and mistyping error like **18-MAI-2003** and **01-AUG-0204**. This error can also be observed, if we try to change the data type of ‘HIREDATE’ column from object data type to datetime data type. Pandas will display the ValueError message while changing the datatype if we do not solve above issues.

This can be considered as an issue, as errors like this will hinders the analysis process based on date and time by throwing value error messages and stop our analysis procedure.

This issue can be solved in multiple ways. Some of them are follows:

- **Ignoring the mistyped or invalid data**

we can ignore those kinds of data and avoid the ValueError message by using pandas **to_datetime** function. errors=’ignore’ will ignore those kinds of wrong data format from dataset. Below is the pandas’s code

```
dataset['HIREDATE']= pd.to_datetime(dataset.HIREDATE,errors='ignore')
```

- **Converting the mistyped and invalid data into NaT.**

We can convert all the invaid data into Not a Time (NaT). errors=’coerce’ will change the wrong data format into NaT

```
dataset['HIREDATE']= pd.to_datetime(dataset.HIREDATE,errors='coerce')
```

- **Replacing the individual invalid data into valid form**

Using the replace fucntion from pandas, we can replace each invalid value to valid form.

```
dataset['HIREDATE'].replace('18-MAI-2003','18-MAY-2003')
```

7. Null values in COMMISSION_PCT

If we look the COMMISSION_PCT column, this column is giving information regarding the commission given to different departments. But if we analyze in a real case scenario, commissions are not provided to all the departments of an organization and even in 'COMMISSION_PCT' column, commission has been provided to the sales department and few other departments, not to all the departments. If we analyze this case, it cannot be considered as a missing value, as not every department gets a commission. We cannot drop the column because dropping this column will erase all data of commission given to other departments. And we cannot take mean, medium or mode to fill the empty rows, as rows which have null value are those who did not get the commission, if we use mean or mode the whole document will have false information regarding the commission granted.

To solve this issue, we can convert all the null values in the form of zero '0' so that there will not be any empty value in 'COMMISSION_PCT' column.

Below is the pandas's code to solve this null value issue.

```
dataset['COMMISSION_PCT'].fillna(value=0, inplace=True)
```

Evidence

Provide evidence to show there are issues.

1.Evidence for the column ‘HIREDATE’ is in object type as an issue.

```
dataset.dtypes
```

```
EMP_ID              int64
FIRST_NAME         object
LAST_NAME          object
EMAIL              object
HIREDATE           object
SALARY              int64
JOB_ID             object
COMMISSION_PCT    float64
MANAGER_ID        float64
DEPT_ID           float64
dtype: object
```

Figure 4 HIREDATE column in object datatype

2.Evidence of Value error in COMMISSION_PCT column as there is int value in float datatype.

```
dataset.dtypes
```

```
EMP_ID              int64
FIRST_NAME         object
LAST_NAME          object
EMAIL              object
HIREDATE           object
SALARY              int64
JOB_ID             object
COMMISSION_PCT    float64
MANAGER_ID        float64
DEPT_ID           float64
dtype: object
```

Figure 5: Showing COMMISSION_PCT has float64 datatype

EMP_ID	FIRST_NAME	LAST_NAME	EMAIL	HIREDATE	SALARY	JOB_ID	COMMISSION_PCT	MANAGER_ID	DEPT_ID
143	Randall	Matos	RMATOS@example.co.uk	15-Mar-06	2600	ST_CLERK		124	50
144	Peter	Vargas	PVARGAS@example.co.uk	9-Jul-06	2500	ST_CLERK	10200	124	50
145	John	Russell	JRUSSEL@example.co.uk	1-Oct-04	14000	SA_MAN	0.4	100	80
146	Karen	Partners	KPARTNER@example.co.uk	5-Jan-05	13500	SA_MAN	0.3	100	80

Figure 6 Showing COMMISSION column has int value

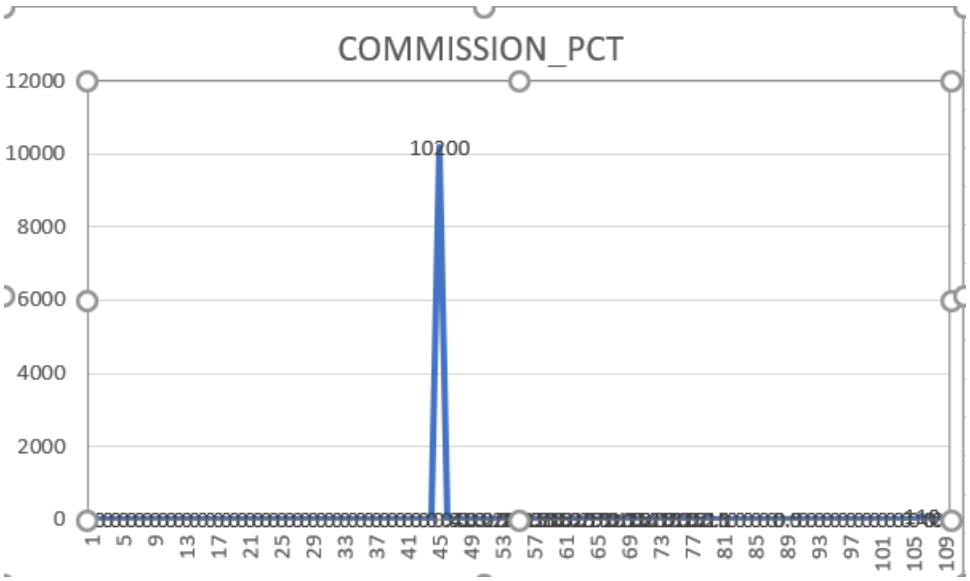


Figure 7 showing int value in the COMMISSION column which takes float value only

3.Evidence for Formatting error. When we count all ‘JOB_ID’, we can see SALES_REP has been recorded mistakenly in place of SA_REP

```
dataset.JOB_ID.value_counts()
```

```
SA_REP         29
ST_CLERK       21
SH_CLERK       19
IT_PROG         6
SA_MAN          5
PU_CLERK        5
FI_ACCOUNT      5
ST_MAN          5
AD_VP           2
PR_REP          1
ADMIN_ASST      1
SALES_REP       1
PU_MAN          1
HR_REP          1
FI_MGR          1
MK_REP          1
MK_MAN          1
AC_ACCOUNT      1
CLERK           1
AD_PRES         1
AC_MGR          1
Name: JOB_ID, dtype: int64
```

Figure 8 showing SALES_REP as formatting error

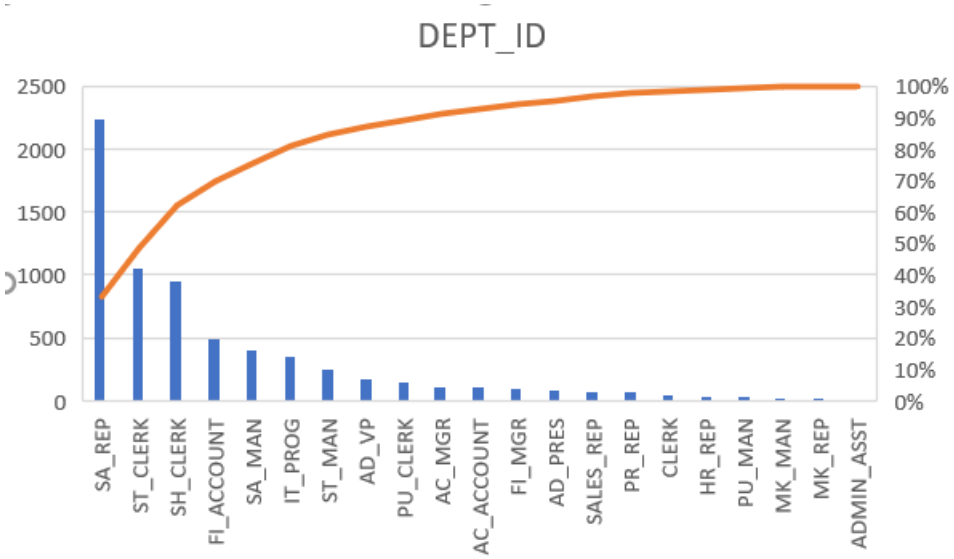


Figure 9 showing Pareto of SALES_REP as formatting error in place of SA_REP

4.Evidence of Typo-error. In place of 80, 85 has been mistyped hence has been recording one data separately.

```
dataset.DEPT_ID.value_counts()
```

```
50.0     45
80.0     33
30.0      6
100.0     6
60.0      6
90.0      3
110.0     2
20.0      2
10.0      1
70.0      1
40.0      1
55.0      1
85.0      1
Name: DEPT_ID, dtype: int64
```

Figure 10 showing 85 as mistyped in DEPT_ID columns

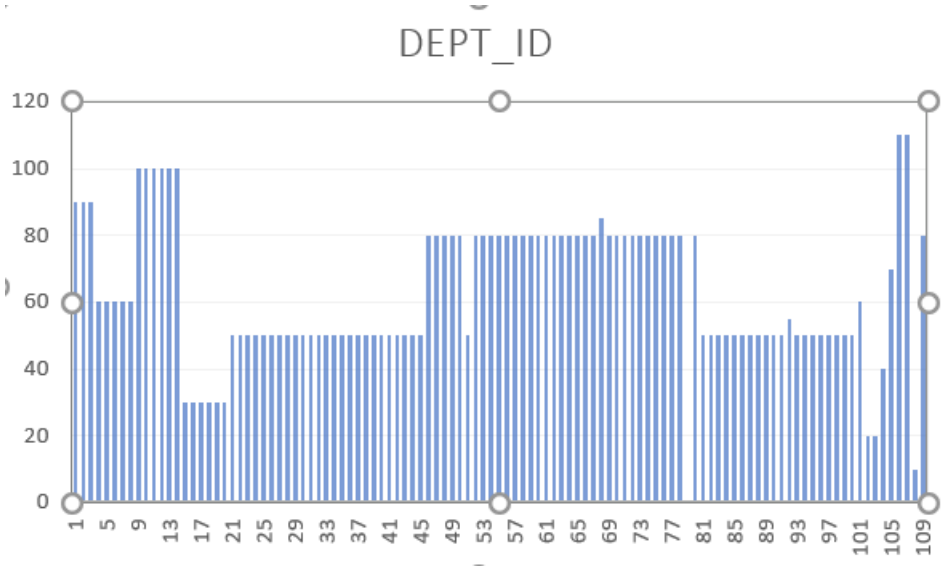


Figure 11 Shows Clustered Column of DEPT_ID which include mistyped value 85

5. Evidence of ‘DEPT_ID’ has a missing value

```
dataset.isna().sum()

EMP_ID             0
FIRST_NAME         0
LAST_NAME          0
EMAIL              0
HIREDATE           0
SALARY             0
JOB_ID             0
COMMISSION_PCT    71
MANAGER_ID         1
DEPT_ID            1
dtype: int64
```

Figure 12 showing DEPT_ID column has one missing value

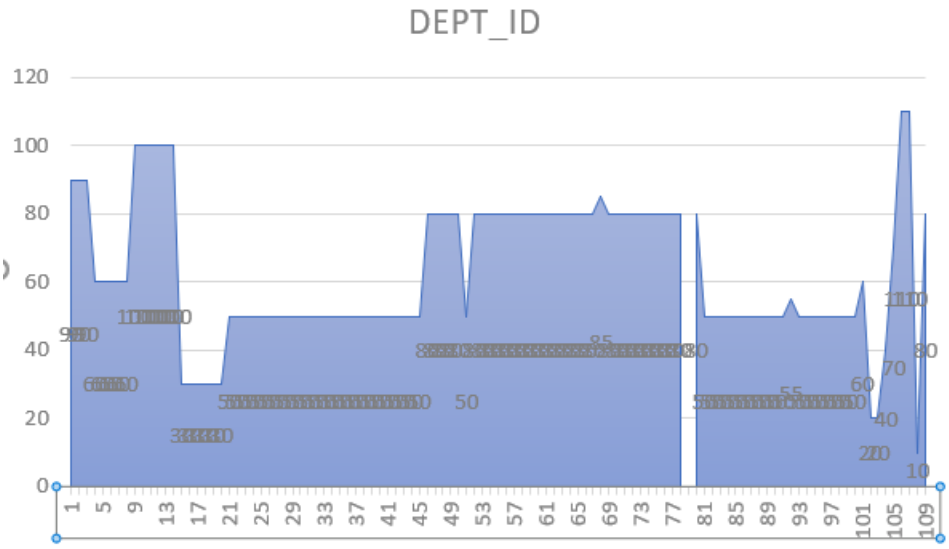


Figure 13 showing DEPT_ID has a missing value in 80th row

6.Evidence of unknow string format and invalid data in ‘HIREDATE’ column.

EMP_ID	FIRST_NAME	LAST_NAME	EMAIL	HIREDATE	SALARY	JOB_ID	COMMISSION_PCT	MANAGER_ID	DEPT_ID
115	Alexander	Khoo	AKHOO@example.co.uk	18-MAI-2003	3100	PU_CLERK		114	30
116	Shelli	Baida	SBAIDA@example.co.uk	24-Dec-05	2900	PU_CLERK		114	30
117	Sigal	Tobias	STOBIAS@example.co.uk	24-Jul-05	2800	PU_CLERK		114	30
118	Guy	Himuro	GHIMURO@example.co.uk	15-Nov-06	2600	PU_CLERK		114	30
119	Karen	Colmenares	KCOLMENA@example.co.uk	10-Aug-07	2500	PU_CLERK		114	30
120	Matthew	Weiss	MWEISS@example.co.uk	18-Jul-04	8000	ST_MAN		100	50
121	Adam	Fripp	AFRIPP@example.co.uk	31-APR-2005	8200	ST_MAN		100	50
122	Payam	Kaufling	PKAUFLIN@example.co.uk	1-May-03	7900	ST_MAN		100	50
123	Shanta	Vollman	SVOLLMAN@example.co.uk	10-Oct-05	6500	ST_MAN		100	50
124	Kevin	Mourgos	KMOURGOS@example.co.uk	16-Nov-07	5800	ST_MAN		100	50
125	Julia	Nayer	JNAYER@example.co.uk	16-Jul-05	3200	ST_CLERK		120	50
126	Irene	Mikkilineni	IMIKKILI@example.co.uk	28-Sep-06	2700	ST_CLERK		120	50
127	James	Landry	JLANDRY@example.co.uk	14-Jan-07	2400	ST_CLERK		120	50
128	Steven	Markle	SMARKLE@example.co.uk	8-Mar-08	2200	ST_CLERK		120	50
129	Laura	Bissot	LBISSOT@example.co.uk	20-Aug-05	3300	ST_CLERK		121	50
130	Mozhe	Atkinson	MATKINSO@example.co.uk	30-Oct-05	2800	ST_CLERK		121	50
131	James	Marlow	JAMRLOW@example.co.uk	30-FEB-05	2500	ST_CLERK		121	50
157	Patrick	Sully	PSULLY@example.co.uk	4-Mar-04	9500	SA_REP	0.35	146	80
158	Allan	McEwen	AMCEWEN@example.co.uk	01-AUG-0204	9000	SA_REP	0.35	146	80
159	Lindsey	Smith	LSMITH@example.co.uk	10-Mar-05	8000	SA_REP	0.3	146	80

Figure 14 Screenshot of excel sheet showing wrong date format in HIREDATE column

```
dataset['HIREDATE']= pd.to_datetime(dataset.HIREDATE)
```

```
pandas\_libs\tslibs\parsing.pyx in pandas._libs.tslibs.parsing.parse_datetime_string()

c:\python37\lib\site-packages\dateutil\parser\_parser.py in parse(timestr, parserinfo, **kwargs)
   1356         return parser(parserinfo).parse(timestr, **kwargs)
   1357     else:
-> 1358         return DEFAULTPARSER.parse(timestr, **kwargs)
   1359
   1360

c:\python37\lib\site-packages\dateutil\parser\_parser.py in parse(self, timestr, default, ignoretz, tzinfos, **kwargs)
    647
    648         if res is None:
--> 649             raise ValueError("Unknown string format:", timestr)
    650
    651         if len(res) == 0:

ValueError: ('Unknown string format:', '18-MAI-2003')
```

Figure 15 Screenshot of error message while datatype of HIREDATE column

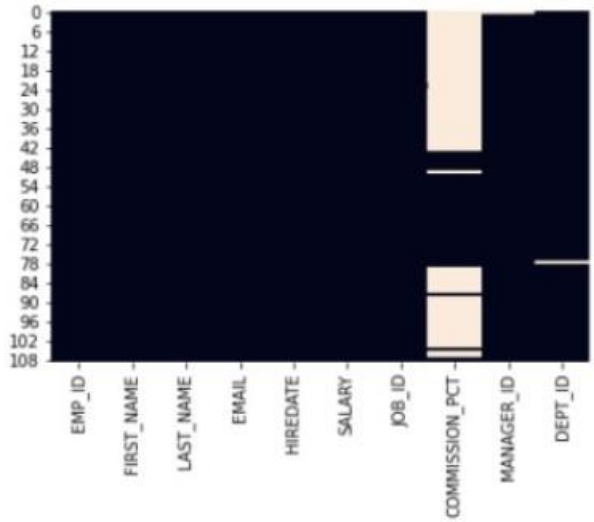
7. Evidence of Null-value in COMMISSION_PCT column

EMP_ID	FIRST_NAME	LAST_NAME	EMAIL	HIREDATE	SALARY	JOB_ID	COMMISSION_PCT	MANAGER_ID	DEPT_ID
100	Steven	King	SKING@ex	17-Jun-03	24000	AD_PRES			90
101	Neena	Kochhar	NKOCHHA	21-Sep-05	17000	AD_VP		100	90
102	Lex	DeHaan	LDEHAAN(	13-Jan-01	17000	AD_VP		100	90
103	Alexander	Hunold	AHUNOLD	3-Jan-06	9000	IT_PROG		102	60
104	Bruce	Ernst	BERNST@(	21-May-07	6000	IT_PROG		103	60
105	David	Austin	DAUSTIN@	25-Jun-05	4800	IT_PROG		103	60
106	Valli	Pataballa	VPATABAL	5-Feb-66	4800	IT_PROG		103	60
107	Diana	Lorentz	DLORENTZ	7-Feb-07	4200	IT_PROG		103	60
108	Nancy	Greenberg	NGREENBI	17-Aug-02	120080	FI_MGR		101	100
109	Daniel	Faviet	DFAVIET@	16-Aug-02	9000	FI_ACCOUNT		108	100
110	John	Chen	JCHEN@ex	28-Sep-05	8200	FI_ACCOUNT		108	100
111	Ismael	Sciarra	ISCIARRA@	30-Sep-05	7700	FI_ACCOUNT		108	100
112	JoseManuel	Urman	JMURMAN	7/3/2006	7800	FI_ACCOUNT		108	100
113	Luis	Popp	LPOPP@ex	7-Dec-07	6900	FI_ACCOUNT		108	100
114	Den	Raphaely	DRAPHEAL	7-Dec-02	11000	PU_MAN		100	30
115	Alexander	Khoo	AKHOO@e	18-MAI-2003	3100	PU_CLERK		114	30
116	Shelli	Baida	SBAIDA@e	24-Dec-05	2900	PU_CLERK		114	30
117	Sigal	Tobias	STOBIAS@	24-Jul-05	2800	PU_CLERK		114	30
118	Guy	Himuro	GHIMURO	15-Nov-06	2600	PU_CLERK		114	30
119	Karen	Colmenares	KCOLMEN	10-Aug-07	2500	PU_CLERK		114	30
120	Matthew	Weiss	MWEISS@	18-Jul-04	8000	ST_MAN		100	50
121	Adam	Fripp	AFRIPP@e	31-APR-2005	8200	ST_MAN		100	50
122	Payam	Kaufling	PKAUFLIN(	1-May-03	7900	ST_MAN		100	50
123	Shanta	Vollman	SVOLLMAN	10-Oct-05	6500	ST_MAN		100	50
124	Kevin	Mourgos	KMOURGC	16-Nov-07	5800	ST_MAN		100	50
125	Julia	Nayer	JNAYER@e	16-Jul-05	3200	ST_CLERK		120	50
126	Irene	Mikkilineni	IMIKKILI@	28-Sep-06	2700	ST_CLERK		120	50

Figure 16 Showing Null-value in COMMISSION_PCT column

```
sns.heatmap(emp.isnull(),cbar=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x20b7ffd5880>
```



White boxes are the null values.

Figure 17 Showing Null-value in COMMISSION_PCT as well as in DEPT_ID column

References

Erhard, R. & Hong, D., 2000. Data cleaning: Problems and Current Approaches. *IEEE Data Eng. Bull,* Volume 23, pp. 3-13.

KIM, W., 2003. A Taxonomy of Dirty Data. *Data Mining and Knowledge Discovery,* Volume 7, pp. 81-99.

Rahm, E. & Do, H., 2000. Data Cleaning: Problems and Current Approaches. *IEEE Data Eng. Bull,* Volume 23, pp. 3-13.

Rahm & Hong, 2000. Data Cleaning: Problems and Current Approaches. *IEEE Data Eng. Bull.,* Volume 23, pp. 3-13.