LAPORAN POST TEST

Ekstraksi, Transformasi, dan Loading dengan R

Hagan Sadina Rahman

3321600005

PRODI SAINS DATA TERAPAN

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2021

➢ TUJUAN:

1. Ekstraksi Data

Ekstraksi data adalah proses dimana data diambil atau diekstrak dari berbagai sistem operasional, baik menggunakan *query,* atau aplikasi ETL. Terdapat beberapa fungsi ekstraksi data, yaitu :

1. Ekstraksi data secara otomatis dari aplikasi sumber.
2. Penyaringan atau seleksi data hasil ekstraksi.
3. Pengiriman data dari berbagai *platform* aplikasi ke sumber data.
4. Perubahan format *layout* data dari format aslinya.
5. Penyimpanan dalam *file* sementara untuk penggabungan dengan hasil ekstraksi dari sumber lain.

Jika ingin membuat strategi data yang kompleks berfungsi, data yang digunakan harus bisa bergerak bebas di antara sistem dan aplikasi.

Data harus diekstrak terlebih dahulu dari sumbernya sebelum dipindahkan ke tempat yang lain. Pada langkah pertama proses ETL ini, data terstruktur dan tidak terstruktur diimpor dan dikonsolidasikan ke dalam satu wadah penyimpanan.

Data mentah dapat diekstraksi dari berbagai sumber berikut ini:

- **Database yang ada dan legacy system.**
- **Cloud, hybrid, dan on-premises environments.**
- **Aplikasi penjualan dan pemasaran.**
- **Mobile devices dan apps.**
- **CRM systems.**
- **Data storage platforms.**
- **Data warehouses.**
- **Analytics tools.**

2. Transformasi

Setelah data telah diambil melalui proses **extract,** selanjutnya dilakukan ***cleaning data*** dengan menghilangkan data yang tidak dibutuhkan (misalnya data anomali). Kemudian mengubah data dari bentuk aslinya menjadi bentuk yang sesuai dengan kebutuhan.

Prinsip-prinsip transformasi data dalam prosesnya yaitu :

a. **Leakage** (kebocoran) terjadi ketika proses ETL mengunduh data secara lengkap dari sumber data, namun pada kenyataannya terdapat beberapa record data yang hilang.

b. **Recoverability** (pemulihan) berarti bahwa selama proses ETL harus ***robust.***

Robust merupakan kemampuan algoritma untuk mengembalikan hasil yang benar, sehingga jika terjadi kegagalan, hal tersebut bisa segera dipulihkan tanpa kehilangan atau kerusakan data.

Transformasi ETL merupakan pembersihan dan mempersiapkan agregasi untuk analisis. Langkah ini sangat penting dalam proses ETL karena membantu memastikan data yang akan diolah sepenuhnya siap dan kompatibel.

Proses transformasi ETL terbagi menjadi beberapa proses sebagai berikut:

- **Pembersihan: data yang tidak konsisten dihilangkan.**
- **Standardisasi: memasang aturan pemformatan ke kumpulan data.**
- **Deduplikasi: data yang sama dibuang atau dikecualikan.**
- **Verifikasi: data yang tidak dapat digunakan dihapus dan anomali ditandai.**
- **Pengurutan: data diatur menurut jenisnya.**
- **Tugas lainnya - aturan tambahan yang dapat meningkatkan kualitas data.**

3. Loading / Memuat Data

Proses terakhir dalam ETL, yaitu memuat data yang sudah diubah ke tujuan baru. Data tersebut dapat dimuat sekaligus *(full load)* atau interval terjadwal *(incremental load).*

Full loading

Untuk *full loading* ETL, semua yang berasal dari transformasi menjadi catatan baru dan unik di gudang data. *Full load* berguna untuk menghasilkan kumpulan data yang tumbuh secara eksponensial dan sulit untuk diatur.

Incremental loading

Metode yang ini kurang komprehensif, tetapi lebih mudah dikelola. Incremental loading membandingkan data yang masuk dengan data yang sudah ada. Dan hanya akan menghasilkan data tambahan jika ditemukan data yang unik dan baru.

➢ METODOLOGI:

ETL atau *Extract Transform Load* adalah proses integrasi data yang menggabungkan data dari berbagai sumber ke dalam satu penyimpanan yang konsisten dan dimuat ke dalam gudang data atau sistem lainnya. Singkatnya, sistem ETL adalah dasar dari pengolahan data, khususnya big data. ETL pertama kali diperkenalkan pada tahun 1970-an untuk mengintegrasikan proses pemuatan data ke dalam superkomputer untuk dianalisis lebih lanjut. Sejak akhir 1980 hingga pertengahan 200, ETL menjadi proses utama untuk membuat gudang data yang mendukung aplikasi *business intelligence* (BI). Di masa sekarang, ETL lebih direkomendasikan untuk menyimpan data yang lebih kecil dan tidak memerlukan pembaruan terlalu sering. Alternatifnya, kamu bisa menggunakan data integrasi lain, seperti ELT, CDC, dan virtualisasi data untuk mengolah data real time dan selalu berubah.

Pada kasus yang saya kerjakan berbeda-berbeda maka juga menggunaka penyelesain yang berbeda-beda untuk data cuaca saya mengambil parameter yang terpenting dari data tersebut dengan metode uji data, untuk data kualitas udara saya menormalisasi data dengan membuat range antara nol hingga satu, dan untuk data crypto saya merubah data menjadi data nominal

➢ SUMBER DATA:

1. Data yang Digunakan:

a. Data Cuaca di Australia

b. Data Kualitas udara di Earlwood Australia

c. Data Crypto Market

2. Sumber Data:

Kaggle

3. Link:

https://www.kaggle.com/learn

➢ PENETAPAN VARIABEL:

a. Data Cuaca Australia

1. Data : data waktu yang terdata saat mencata cuaca di Australia
2. Location : Data lokasi hasil cuaca di tempat Australia
3. Mintemp : Temperatur terrendah hasil cuaca di tempat Australia
4. Maxtemp : Temperatur tertinggi hasil cuaca di tempat Australia
5. Rainfall : Tingkat curah hujan hasil cuaca di tempat Australia
6. Evaporation : tingkat penguapan hasil cuaca di tempat Australia
7. Sunshine : Tingkat sinar matahari hasil cuaca di tempat Australia
8. WindGusdir : Arah mata angin rata-rata cuaca di tempat Australia
9. Windgustspeed : Kecepatan angin rata-ratahasil cuaca di tempat Australia
10. WindDir9am : Arah mata angin jam 9Am cuaca di tempat Australia
11. Windgustspeed9am: Kecepatan angin pada pukul 9am hasil cuaca di tempat Australia
12. Windspeed3pm : Kecepatan angin pada pukul 3pm hasil cuaca di tempat Australia
13. Humadity3pm : Tingkat Kelembaban udara pada pukul 3pm hasil cuaca di tempat Australia
14. Humadity9pm : Tingkat Kelembaban udara pada pukul 9pm hasil cuaca di tempat Australia
15. Pressure3pm : Tingkat Tekanan udara pada pukul 3pm hasil cuaca di tempat Australia

16. Pressure9pm : Tingkat Tekanan udara pada pukul 9pm hasil cuaca di tempat Australia
17. Cloud9am : Tingkat awan pada pukul 9pm hasil cuaca di tempat Australia
18. Cloud3pm : Tingkat awan pada pukul 3pm hasil cuaca di tempat Australia
19. Temp9am : Tingkat temperature pada pukul 9pm hasil cuaca di tempat Australia
20. Temp3pm : Tingkat temperature pada pukul 3pm hasil cuaca di tempat Australia
21. Raintoday : Keterangan apakah hari ini hujan pada hasil cuaca di tempat Australia
22. Raintomorrow : Keterangan apakah besok hari hujan pada hasil cuaca di tempat Australia

b. Data Cuaca Australia

1. EARLWOOD WDR 1h average [°] : data Wind direction degree pada pengujian kualitas udara di Ealwood Australia
2. EARLWOOD TEMP 1h average [°C] : data tingkat temperature pada pengujian kualitas udara di Ealwood Australia
3. EARLWOOD WSP 1h average [m/s] : data weather system processor pada pengujian kualitas udara di Ealwood Australia
4. EARLWOOD NO 1h average [pphm] : data Nitrogen monoksida processor pada pengujian kualitas udara di Ealwood Australia
5. EARLWOOD NO2 1h average [pphm] : data Nitrogen oksida pada pengujian kualitas udara di Ealwood Australia
6. EARLWOOD OZONE 1h average [pphm] : data Ozon selama 1 jam pada pengujian kualitas udara di Ealwood Australia
7. EARLWOOD OZONE 4h rolling average [pphm] : data Ozon selama 4 jam pada pengujian kualitas udara di Ealwood Australia
8. EARLWOOD PM10 1h average [µg/m³] : data partikulat, PM10 = 150 µgram/m pada pengujian kualitas udara di Ealwood Australia
9. EARLWOOD PM2.5 1h average [µg/m³] : data partikulat, PM2.5 = 65 µgram/m3 pada pengujian kualitas udara di Ealwood Australia
10. EARLWOOD HUMID 1h average [%] : data Kelembaban pada pengujian kualitas udara di Ealwood Australia
11. EARLWOOD SD1 1h average [°] : data strom dati pada pengujian kualitas udara di Ealwood Australia

c. Data Crypto Market

1. Slug : Data pembelian koin pada Market crypto
2. Asset : Data nickname crypto currency pada Market crypto: data pada Market crypto
3. Name : Data Nama crypto currency pada Market crypto: data pada Market crypto
4. Date : Data Tanggal pencatatan pada Market crypto
5. Ranknow : Data Ranking terkini pada Market crypto
6. Open : Data value awal crypto pada Market crypto
7. High : Data value Tertinggi crypto pada Market crypto
8. Low : Data value Terendah crypto pada Market crypto
9. Close : Data Value penutup pada Market crypto
10. Volume :Data Volume crypto pada Market crypto
11. Market : Data Market crypto pada Market crypto
12. Clode Ratio : Data Clode Ratio awal crypto pada Market crypto
13. Spread : Data Persebaran crypto pada Market crypto

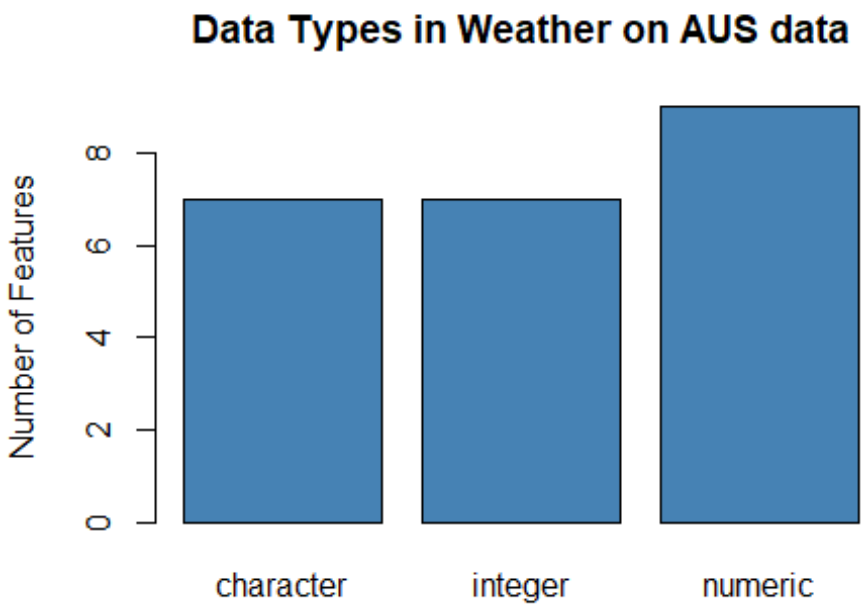
weatherAUS

```
data = read.csv("D:/Hagan/PENS/Project/R/Post Test/weatherAUS.csv", header = T
RUE)
#melihat data
View(data)
```

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindSpeed3pm	Humidity9am
1	2008-12-01	Albury	13.4	22.9	0.6	NA	NA	W	44	W	WNW	20	24	71
2	2008-12-02	Albury	7.4	25.1	0.0	NA	NA	WNW	44	NNW	WSW	4	22	44
3	2008-12-03	Albury	12.9	25.7	0.0	NA	NA	WSW	46	W	WSW	19	26	38
4	2008-12-04	Albury	9.2	28.0	0.0	NA	NA	NE	24	SE	E	11	9	45
5	2008-12-05	Albury	17.5	32.3	1.0	NA	NA	W	41	ENE	NW	7	20	82
6	2008-12-06	Albury	14.6	29.7	0.2	NA	NA	WNW	56	W	W	19	24	55
7	2008-12-07	Albury	14.3	25.0	0.0	NA	NA	W	50	SW	W	20	24	49
8	2008-12-08	Albury	7.7	26.7	0.0	NA	NA	W	35	SSE	W	6	17	48
9	2008-12-09	Albury	9.7	31.9	0.0	NA	NA	NNW	80	SE	NW	7	28	42
10	2008-12-10	Albury	13.1	30.1	1.4	NA	NA	W	28	S	SSE	15	11	58
11	2008-12-11	Albury	13.4	30.4	0.0	NA	NA	N	30	SSE	ESE	17	6	48
12	2008-12-12	Albury	15.9	21.7	2.2	NA	NA	NNE	31	NE	ENE	15	13	89
13	2008-12-13	Albury	15.9	18.6	15.6	NA	NA	W	61	NNW	NNW	28	28	76
14	2008-12-14	Albury	12.6	21.0	3.6	NA	NA	SW	44	W	SSW	24	20	65
15	2008-12-15	Albury	8.4	24.6	0.0	NA	NA	NA	NA	S	WNW	4	30	57
16	2008-12-16	Albury	9.8	27.7	NA	NA	NA	WNW	50	NA	WNW	NA	22	50
17	2008-12-17	Albury	14.1	20.9	0.0	NA	NA	ENE	22	SSW	E	11	9	69
18	2008-12-18	Albury	13.5	22.9	16.8	NA	NA	W	63	N	WNW	6	20	80
19	2008-12-19	Albury	11.2	22.5	10.6	NA	NA	SSE	43	WSW	SW	24	17	47
20	2008-12-20	Albury	9.8	25.6	0.0	NA	NA	SSE	26	SE	NNW	17	6	45

Pada ETL ini menggunakan data cuaca di australia yang mana sumber data didapatkan dari kaggle dan tipe data csv, menggunakan syntax read.csv untuk membaca data tersebut lalu menampilkan data tersebut dengan view agar nantinya kita dapat menampilkan data di dalam R Studio

```
data_types <- function(frame) {
  res <- lapply(frame, class)
  res_frame <- data.frame(unlist(res))
  barplot(table(res_frame), main="Data Types in Weather on AUS data", col="ste
elblue", ylab="Number of Features")
}
data_types(data)
```



dengan syntax data_type kita bisa melihat struktur data dan persebaran data pada data cuaca di australia menggunakan tipe data apa aja terlihat terdapat 3 tipe data yang ada yaitu karakter dengan jumlah tujuh data, integer dengan tujuh data, dan numerik dengan sembilan data yang nantinya untuk menjadi gambaran pengolahan data nya seperti apa

```
str(data)

## 'data.frame':    145460 obs. of  23 variables:
##  $ Date         : chr  "2008-12-01" "2008-12-02" "2008-12-03" "2008-12-04"
...
##  $ Location     : chr  "Albury" "Albury" "Albury" "Albury" ...
##  $ MinTemp      : num  13.4 7.4 12.9 9.2 17.5 14.6 14.3 7.7 9.7 13.1 ...
##  $ MaxTemp      : num  22.9 25.1 25.7 28 32.3 29.7 25 26.7 31.9 30.1 ...
##  $ Rainfall     : num  0.6 0 0 0 1 0.2 0 0 0 1.4 ...
##  $ Evaporation  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Sunshine     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ WindGustDir  : chr  "W" "WNW" "WSW" "NE" ...
##  $ WindGustSpeed: int  44 44 46 24 41 56 50 35 80 28 ...
##  $ WindDir9am   : chr  "W" "NNW" "W" "SE" ...
##  $ WindDir3pm   : chr  "WNW" "WSW" "WSW" "E" ...
##  $ WindSpeed9am : int  20 4 19 11 7 19 20 6 7 15 ...
##  $ WindSpeed3pm : int  24 22 26 9 20 24 24 17 28 11 ...
##  $ Humidity9am  : int  71 44 38 45 82 55 49 48 42 58 ...
##  $ Humidity3pm  : int  22 25 30 16 33 23 19 19 9 27 ...
##  $ Pressure9am  : num  1008 1011 1008 1018 1011 ...
##  $ Pressure3pm  : num  1007 1008 1009 1013 1006 ...
##  $ Cloud9am     : int  8 NA NA NA 7 NA 1 NA NA NA ...
```

```
##  $ Cloud3pm     : int  NA NA 2 NA 8 NA NA NA NA NA ...
##  $ Temp9am      : num  16.9 17.2 21 18.1 17.8 20.6 18.1 16.3 18.3 20.1 ...
##  $ Temp3pm      : num  21.8 24.3 23.2 26.5 29.7 28.9 24.6 25.5 30.2 28.2 ..
.
##  $ RainToday    : chr  "No" "No" "No" "No" ...
##  $ RainTomorrow : chr  "No" "No" "No" "No" ...
```

disini kita bisa melihat struktur data dan juga persebaran data dengan gambaran spesifik nama data tersebut dengan tipe data yang ada terdapat tiga tipe data yang tertampil yaitu karakter, integer, dan numerik dengan total baris data sebanyak 145460 baris dengan 23 parameter yang tertampil

```
data %>% filter( RainToday == "Yes" )
```

Description: df [31,880 x 23]

Humidity9am <int>	Humidity3pm <int>	Pressure9am <dbl>	Pressure3pm <dbl>	Cloud9am <int>	Cloud3pm <int>	Temp9am <dbl>	Temp3pm <dbl>	RainToday <chr>	RainTomorrow <chr>
58	27	1007.0	1005.7	NA	NA	20.1	28.2	Yes	No
89	91	1010.5	1004.2	8	8	15.9	17.0	Yes	Yes
76	93	994.3	993.0	8	8	17.4	15.8	Yes	Yes
65	43	1001.2	1001.8	NA	7	15.8	19.8	Yes	No
80	65	1005.8	1002.2	8	1	18.0	21.5	Yes	Yes
47	32	1009.4	1009.7	NA	2	15.5	21.0	Yes	No
78	70	1005.6	1003.4	8	8	12.5	18.2	Yes	No
43	28	1007.9	1003.9	NA	NA	22.8	33.0	Yes	No
41	21	1023.3	1019.7	NA	NA	18.0	27.6	Yes	No
57	23	1021.3	1018.0	NA	NA	21.5	29.6	Yes	No

1-10 of 31,880 rows | 14-23 of 23 columns

Previous 1 2 3 4 5 6 _ 100 Next

Pada tampilan ini menampilkan data dengan RainToday == yes, bisa dilihat memunculkan data sebanyak 31880 data dengan 23 parameter dari total keseluruhan data sebesar 145460 yang mana data ini akan digunakan sebagai gambaran nantinya untuk pengolahan data lebih lanjut

```
data %>% filter( RainToday == "No" )
```

Humidity9am <int>	Humidity3pm <int>	Pressure9am <dbl>	Pressure3pm <dbl>	Cloud9am <int>	Cloud3pm <int>	Temp9am <dbl>	Temp3pm <dbl>	RainToday <chr>	RainTomorrow <chr>
71	22	1007.7	1007.1	8	NA	16.9	21.8	No	No
44	25	1010.6	1007.8	NA	NA	17.2	24.3	No	No
38	30	1007.6	1008.7	NA	2	21.0	23.2	No	No
45	16	1017.6	1012.8	NA	NA	18.1	26.5	No	No
82	33	1010.8	1006.0	7	8	17.8	29.7	No	No
55	23	1009.2	1005.4	NA	NA	20.6	28.9	No	No
49	19	1009.6	1008.2	1	NA	18.1	24.6	No	No
48	19	1013.4	1010.1	NA	NA	16.3	25.5	No	No
42	9	1008.9	1003.6	NA	NA	18.3	30.2	No	Yes
48	22	1011.8	1008.7	NA	NA	20.4	28.8	No	Yes

1-10 of 110,319 rows | 14-23 of 23 columns

Previous 1 2 3 4 5 6 _ 100 Next

Pada tampilan ini menampilkan data dengan RainToday == No, bisa dilihat memunculkan data sebanyak 110319 data dengan 23 parameter dari total keseluruhan data sebesar 145460 yang mana data ini akan digunakan sebagai gambaran nantinya untuk pengolahan data lebih lanjut

```
data %>% filter(Humidity3pm <= 18, RainToday == "Yes", WindGustSpeed <= 40 )
```

Humidity9am <int>	Humidity3pm <int>	Pressure9am <dbl>	Pressure3pm <dbl>	Cloud9am <int>	Cloud3pm <int>	Temp9am <dbl>	Temp3pm <dbl>	RainToday <chr>	RainTomorrow <chr>
29	15	1021.8	1019.4	3	1	15.2	19.9	Yes	No
68	14	1015.7	1014.5	2	1	16.1	25.7	Yes	No
45	17	1020.9	1018.6	0	1	14.3	22.0	Yes	No
58	15	1015.5	1011.4	NA	NA	23.4	34.1	Yes	No
79	15	1022.4	1018.8	NA	NA	19.0	31.8	Yes	No
28	18	1016.3	1014.5	NA	NA	32.0	35.7	Yes	No
54	12	1027.6	1024.3	1	2	15.5	27.4	Yes	No
84	15	1010.6	1008.6	7	NA	15.8	24.0	Yes	No
66	15	1013.6	1010.2	NA	NA	20.2	31.4	Yes	Yes
1	1	1021.5	1019.4	NA	NA	18.4	21.9	Yes	No

1-10 of 12 rows | 14-23 of 23 columns Previous 1 2 Next

Pada tampilan ini menampilkan data dengan Raintoday = yes , Humadity <= 18 dengan WindGustSpeed <= 40, bisa dilihat memunculkan data sebanyak 10 data dengan 23 parameter dari total keseluruhan data sebesar 145460 yang mana data ini akan digunakan sebagai gambaran nantinya untuk pengolahan data lebih lanjut

Humidity9am <int>	Humidity3pm <int>	Pressure9am <dbl>	Pressure3pm <dbl>	Cloud9am <int>	Cloud3pm <int>	Temp9am <dbl>	Temp3pm <dbl>	RainToday <chr>	RainTomorrow <chr>
45	16	1017.6	1012.8	NA	NA	18.1	26.5	No	No
41	12	1015.1	1010.3	NA	NA	20.7	33.9	No	No
38	16	1017.8	1013.7	NA	NA	17.2	26.6	No	No
48	16	1014.1	1012.1	NA	NA	24.2	33.2	No	No
40	8	1011.6	1006.9	NA	NA	25.6	41.5	No	No
35	16	1019.7	1017.4	NA	NA	16.0	25.8	No	No
34	17	1019.7	1016.2	NA	NA	20.9	30.5	No	No
39	10	1015.8	1010.6	NA	NA	22.0	34.4	No	No
44	10	1016.5	1014.6	NA	NA	21.2	32.1	No	No
48	12	1017.7	1014.6	NA	NA	23.4	36.5	No	No

1-10 of 3,340 rows | 14-23 of 23 columns Previous 1 2 3 4 5 6 … 100 Next

Pada tampilan ini menampilkan data %>% filter(Humidity3pm <= 18, RainToday == "No" , WindGustSpeed <= 40), Humadity <= 18 dengan WindGustSpeed <= 40, bisa dilihat memunculkan data sebanyak 3340 data dengan 23 parameter dari total keseluruhan data sebesar 14 5460 yang mana data ini akan digunakan sebagai gambaran nantinya untuk pengolahan data lebih l anjut

```
arrange(data,Humidity3pm, Sunshine, RainToday)
```

Description: df [145,460 x 23]

Date <chr>	Location <chr>	MinTemp <dbl>	MaxTemp <dbl>	Rainfall <dbl>	Evaporation <dbl>	Sunshine <dbl>	WindGustDir <chr>	WindGustSpeed <int>	WindDir9am <chr>
2015-10-15	Woomera	17.7	38.1	0.0	14.4	7.2	N	50	N
2015-10-05	Woomera	17.3	37.1	0.0	14.8	10.9	NNW	70	N
2013-10-20	Woomera	19.4	40.0	0.0	19.0	11.2	NNW	61	N
2014-11-13	Woomera	23.4	41.6	0.0	12.0	*NA*	SSW	69	NNE
2009-09-30	Woomera	14.4	31.9	0.0	18.8	4.3	WNW	76	NNE
2010-01-22	Mildura	21.7	42.9	0.0	12.4	9.2	WSW	52	N
2015-07-21	AliceSprings	-1.9	22.8	0.0	4.0	10.5	N	37	NNW
2014-12-16	WaggaWagga	21.8	36.1	0.4	11.4	10.8	W	72	ENE
2009-10-01	Cobar	14.7	34.9	0.0	10.0	11.4	NNW	43	NNE
2013-10-16	Woomera	13.4	34.5	0.0	8.0	11.4	WSW	74	N

1-10 of 145,460 rows | 1-10 of 23 columns Previous 1 2 3 4 5 6 _ 100 Next

Pada tampilan ini menampilkan arrange(data,Humidity3pm, Sunshine, RainToday) yang mana akan mengurutkan berdasarkan kecocokan data pada ketiga parameter tersebut yaitu Humidity3pm, Sunshine, RainToday agar pembacaan data lebih mudah dan akan data ini akan digunakan sebagai gambaran nantinya untuk pengolahan data lebih lanjut

```
arrange(data,desc(Humidity3pm), Sunshine, RainToday,Pressure3pm)
```

Description: df [145,460 x 23]

Date <chr>	Location <chr>	MinTemp <dbl>	MaxTemp <dbl>	Rainfall <dbl>	Evaporation <dbl>	Sunshine <dbl>	WindGustDir <chr>	WindGustSpeed <int>	WindDir9am <chr>
2015-07-12	Watsonia	0.9	11.8	1.0	1.8	0.0	SW	48	N
2011-01-13	Watsonia	21.7	27.0	0.4	3.0	0.0	NNE	39	NNE
2013-06-12	Watsonia	7.8	12.6	0.8	0.4	0.0	SE	19	*NA*
2016-09-09	Watsonia	16.3	16.7	0.0	5.4	0.0	N	44	NNE
2016-07-05	Watsonia	7.3	12.3	1.0	1.8	0.0	SSW	24	*NA*
2012-03-03	Watsonia	15.5	19.1	0.0	6.2	0.0	ENE	24	SSW
2011-10-24	Watsonia	12.7	18.6	0.2	6.0	0.0	SSW	39	SW
2015-04-07	Watsonia	10.5	16.4	0.0	2.6	0.0	SSW	54	SW
2016-06-27	Watsonia	3.1	10.0	0.8	0.6	0.0	N	22	ENE
2014-06-02	Dartmoor	11.7	13.3	0.6	0.2	0.0	NW	15	*NA*

1-10 of 145,460 rows | 1-10 of 23 columns Previous 1 2 3 4 5 6 _ 100 Next

Pada tampilan ini menampilkan arrange(data, desc(Humidity3pm), Sunshine, RainToday,Pressure3pm) yang mana akan mengurutkan berdasarkan kecocokan data pada ketiga parameter tersebut yaitu Humidity3pm dengan pengurutan dari terkecil hingga kebesar, Sunshine, RainToday agar pembacaan data lebih mudah dan akan data ini akan digunakan sebagai gambaran nantinya untuk pengolahan data lebih lanjut

```
data %>% filter( WindDir9am == "N")
```

Date <chr>	Location <chr>	MinTemp <dbl>	MaxTemp <dbl>	Rainfall <dbl>	Evaporation <dbl>	Sunshine <dbl>	WindGustDir <chr>	WindGustSpeed <int>	WindDir9am <chr>
2008-12-18	Albury	13.5	22.9	16.8	*NA*	*NA*	W	63	N
2008-12-28	Albury	20.1	32.7	0.0	*NA*	*NA*	WNW	48	N
2009-01-22	Albury	24.4	34.0	0.6	*NA*	*NA*	NW	98	N
2009-02-03	Albury	21.5	37.7	0.0	*NA*	*NA*	*NA*	*NA*	N
2009-02-08	Albury	28.3	40.2	0.0	*NA*	*NA*	NW	52	N
2009-03-28	Albury	9.1	28.9	0.0	*NA*	*NA*	NNW	24	N
2009-04-27	Albury	4.5	11.5	3.2	*NA*	*NA*	NW	26	N
2009-05-03	Albury	4.6	18.9	0.0	*NA*	*NA*	S	15	N
2009-06-18	Albury	0.5	14.7	0.0	*NA*	*NA*	N	11	N
2009-07-01	Albury	8.3	13.3	8.4	*NA*	*NA*	NW	52	N

1-10 of 11,758 rows | 1-10 of 23 columns Previous 1 2 3 4 5 6 _ 100 Next

Pada tampilan ini menampilkan data %>% filter(WindDir9am == "N") yang mana
Pada tampilan ini menampilkan data dengan data %>% filter(WindDir9am == "N"), bisa dilihat m emunculkan data sebanyak 11758 data dengan 23 parameter dari total keseluruhan data sebesar 145 460 yang mana data ini akan digunakan sebagai gambaran nantinya untuk pengolahan data lebih la njut

data %>% filter(WindDir3pm == "N")

WindDir3pm <chr>	WindSpeed9am <int>	WindSpeed3pm <int>	Humidity9am <int>	Humidity3pm <int>	Pressure9am <dbl>	Pressure3pm <dbl>	Cloud9am <int>	Cloud3pm <int>	Temp9am <dbl>
N	17	22	38	28	1013.6	1008.1	NA	1	24.5
N	13	17	52	31	1009.9	1006.8	NA	NA	22.8
N	15	19	57	23	1021.3	1018.0	NA	NA	21.5
N	9	22	82	74	1012.7	1008.0	NA	4	19.9
N	7	7	50	13	1016.5	1013.6	NA	NA	17.4
N	2	17	60	26	1023.8	1020.6	NA	NA	14.0
N	2	13	99	75	1015.5	1012.7	7	8	6.9
N	0	11	98	64	1023.0	1019.5	7	5	3.2
N	0	19	95	53	1023.1	1018.4	8	NA	7.1
N	6	13	94	87	1015.7	1015.3	8	7	9.4

1-10 of 8,890 rows | 11-20 of 23 columns Previous 1 2 3 4 5 6 … 100 Next

Pada tampilan ini menampilkan data dengan data %>% filter(WindDir3am == "N"), bisa dilihat memunculkan data sebanyak 8990 data dengan 23 parameter dari total keseluruhan data sebesar 145460 yang mana data ini akan digunakan sebagai gambaran nantinya untuk pengolahan data lebih lanjut

```
nacols <- function(df) {

    colnames(df)[unlist(lapply(df, function(x) anyNA(x)))]

}

cat('There are',length(nacols(df)),'columns with NA values.50% of columns are
NA filled which disturbs the data quality')



## There are 0 columns with NA values.50% of columns are NA filled which distu
rbs the data quality
```

disini kita bisa lihat bahwa tidak terdapat data kosong yang lebih dari 50% yang mana data ini nantinya sangat baik untuk diolah

```
sum(is.na(data))/(nrow(data)*ncol(data))

## [1] 0.1025975
```

Disini kita bisa melihat teradpat 10% NA value dari keseluruhan data yang ada yang mana data ini nantinya sebagai gambaran kedepannya untuk pengolahan data selanjutnya

```
missing_data <- as.data.frame(sort(sapply(data, function(x) sum(is.na(x))),dec
reasing = T))
colnames(missing_data)[1] <- "Missing_values"
missing_data$Percentage <- (missing_data$Missing_values/nrow(data))*100
missing_data$Variables <- rownames(missing_data)
missing_data <- missing_data[c(3,1,2)]
rownames(missing_data)<-c()
head(missing_data,15)
```

```
##        Variables Missing_values Percentage
## 1       Sunshine          69835  48.009762
## 2    Evaporation          62790  43.166506
## 3       Cloud3pm          59358  40.807095
## 4       Cloud9am          55888  38.421559
## 5    Pressure9am          15065  10.356799
## 6    Pressure3pm          15028  10.331363
## 7     WindDir9am          10566   7.263853
## 8    WindGustDir          10326   7.098859
## 9  WindGustSpeed          10263   7.055548
## 10   Humidity3pm           4507   3.098446
## 11    WindDir3pm           4228   2.906641
## 12       Temp3pm           3609   2.481094
## 13  RainTomorrow           3267   2.245978
## 14      Rainfall           3261   2.241853
## 15     RainToday           3261   2.241853
```

disini kita bisa melihat persentasi datang yang hilang dari tiap parameter atau dari tiap kolom yang ada dengan NA value tertinggi pada variabel Sunshine dengan Missing Value 69835 data persentase 48% dan yang kedua Evaporation dengan mising value 62790 dengan persentase 43% yang mana keseluruhan data dengan persentase diatas 30% sebanyak empat data yaitu data Sunshine, Evaporation, Cloud3pm, Cloud9am yang mana pada output data ini bisa menjadi gambaran kedepanya untuk pengolahan data selanjutnya

```
library(DataExplorer)
```

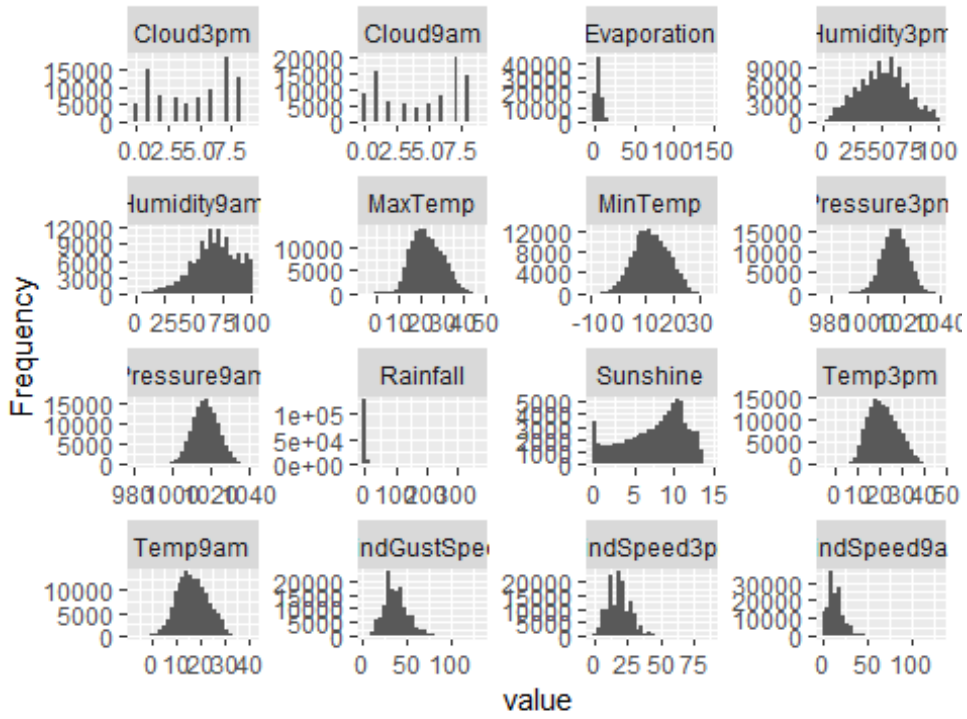
```
## Warning: package 'DataExplorer' was built under R version 4.1.2
```

```
introduce(data)
```

```
##     rows columns discrete_columns continuous_columns all_missing_columns
## 1 145460      23               7                  16                   0
##   total_missing_values complete_rows total_observations memory_usage
## 1               343248         56420            3345580     22923080
```

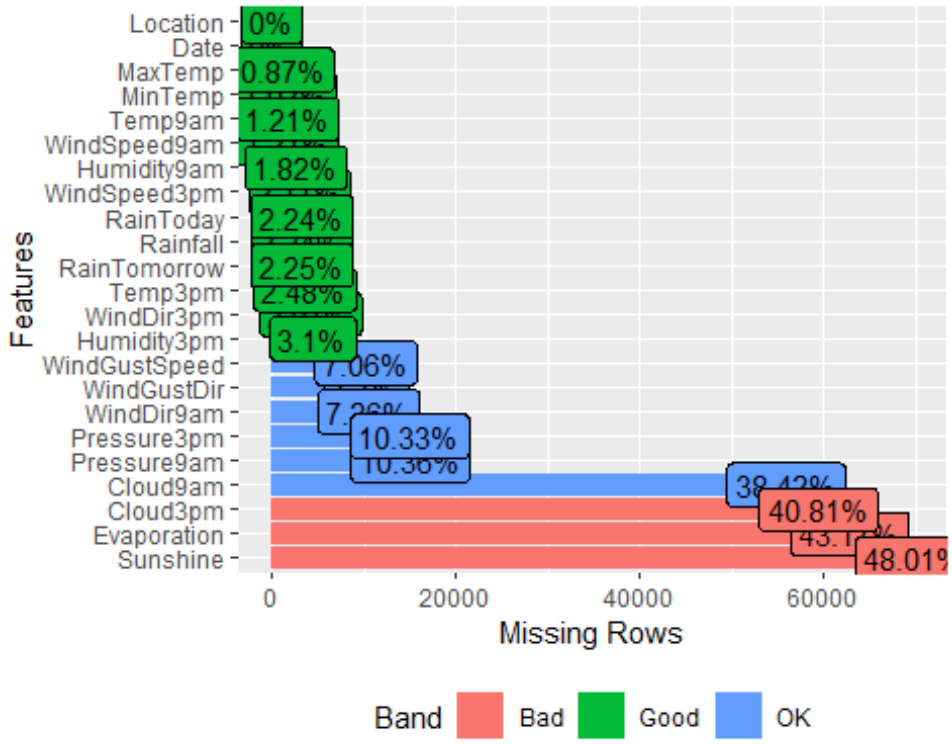
dengan library dataExplorer kita bisa mengetahui data yang kita seperti apa dengan total baris sebanyak 145460 baris, 23 kolom, variabel dalam bentuk karakter sebanyak tujuh variabel, variabel numerik sebanyak enam belas variabel, kolom yang semua data kosong sebanyak nol data, data yang hilang sebanyak 343248 data, baris yang terisi penuh sebanyak 56420 baris, dengan total observasi 3345580 data yang mana pada output data ini bisa menjadi gambaran kedepanya untuk pengolahan data selanjutnya

```
plot_histogram(data)
```



Pada data ini menampilkan persebaran data pada data cuaca australia bisa terlihat pada gambar grafik yang mana pada output data ini bisa menjadi gambaran kedepanya untuk pengolahan data selanjutnya

```
plot_missing(data)
```



Pada data ini menampilkan untuk mengetahui baris yang kosong pada data dengan grafik yang mana terdapat tiga keterangan Band, Good, dana OK yang mana masih cenderung aman untuk diolah yang mana pada output data ini bisa menjadi gambaran kedepanya untuk pengolahan data selanjutnya

```
summary(data)
```

```
##      Date              Location            MinTemp          MaxTemp      
##  Length:145460      Length:145460      Min.   :-8.50   Min.   :-4.80  
##  Class :character   Class :character   1st Qu.: 7.60   1st Qu.:17.90  
##  Mode  :character   Mode  :character   Median :12.00   Median :22.60  
##                                        Mean   :12.19   Mean   :23.22  
##                                        3rd Qu.:16.90   3rd Qu.:28.20  
##                                        Max.   :33.90   Max.   :48.10  
##                                        NA's   :1485    NA's   :1261   
##     Rainfall        Evaporation        Sunshine     WindGustDir       
##  Min.   :  0.000   Min.   :  0.00   Min.   : 0.00   Length:145460     
##  1st Qu.:  0.000   1st Qu.:  2.60   1st Qu.: 4.80   Class :character  
##  Median :  0.000   Median :  4.80   Median : 8.40   Mode  :character  
##  Mean   :  2.361   Mean   :  5.47   Mean   : 7.61                     
##  3rd Qu.:  0.800   3rd Qu.:  7.40   3rd Qu.:10.60                     
##  Max.   :371.000   Max.   :145.00   Max.   :14.50                     
##  NA's   :3261      NA's   :62790    NA's   :69835
```

```
##  WindGustSpeed     WindDir9am         WindDir3pm         WindSpeed9am
##  Min.   :  6.00   Length:145460      Length:145460      Min.   :  0.00
##  1st Qu.: 31.00   Class :character   Class :character   1st Qu.:  7.00
##  Median : 39.00   Mode  :character   Mode  :character   Median : 13.00
##  Mean   : 40.03                                         Mean   : 14.04
##  3rd Qu.: 48.00                                         3rd Qu.: 19.00
##  Max.   :135.00                                         Max.   :130.00
##  NA's   :10263                                          NA's   :1767
##   WindSpeed3pm    Humidity9am      Humidity3pm      Pressure9am
##  Min.   : 0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 980.5
##  1st Qu.:13.00   1st Qu.: 57.00   1st Qu.: 37.00   1st Qu.:1012.9
##  Median :19.00   Median : 70.00   Median : 52.00   Median :1017.6
##  Mean   :18.66   Mean   : 68.88   Mean   : 51.54   Mean   :1017.6
##  3rd Qu.:24.00   3rd Qu.: 83.00   3rd Qu.: 66.00   3rd Qu.:1022.4
##  Max.   :87.00   Max.   :100.00   Max.   :100.00   Max.   :1041.0
##  NA's   :3062    NA's   :2654     NA's   :4507     NA's   :15065
##   Pressure3pm        Cloud9am        Cloud3pm        Temp9am
##  Min.   : 977.1   Min.   :0.00    Min.   :0.00    Min.   :-7.20
##  1st Qu.:1010.4   1st Qu.:1.00    1st Qu.:2.00    1st Qu.:12.30
##  Median :1015.2   Median :5.00    Median :5.00    Median :16.70
##  Mean   :1015.3   Mean   :4.45    Mean   :4.51    Mean   :16.99
##  3rd Qu.:1020.0   3rd Qu.:7.00    3rd Qu.:7.00    3rd Qu.:21.60
##  Max.   :1039.6   Max.   :9.00    Max.   :9.00    Max.   :40.20
##  NA's   :15028    NA's   :55888   NA's   :59358   NA's   :1767
##     Temp3pm       RainToday         RainTomorrow
##  Min.   :-5.40   Length:145460      Length:145460
##  1st Qu.:16.60   Class :character   Class :character
##  Median :21.10   Mode  :character   Mode  :character
##  Mean   :21.68
##  3rd Qu.:26.40
##  Max.   :46.70
##  NA's   :3609
```

Pada data ini menampilkan summary dari tiap variabel data yang ada ,yang yang mana pada output data ini bisa menjadi gambaran kedepanya untuk pengolahan data selanjutnya

```
most_na_columns<-missing_data$Variables[1:50]
most_na_columns
```

```
##  [1] "Sunshine"      "Evaporation"   "Cloud3pm"      "Cloud9am"
##  [5] "Pressure9am"   "Pressure3pm"   "WindDir9am"    "WindGustDir"
##  [9] "WindGustSpeed" "Humidity3pm"   "WindDir3pm"    "Temp3pm"
## [13] "RainTomorrow"  "Rainfall"      "RainToday"     "WindSpeed3pm"
## [17] "Humidity9am"   "WindSpeed9am"  "Temp9am"       "MinTemp"
## [21] "MaxTemp"       "Date"          "Location"      NA
## [25] NA              NA              NA              NA
## [29] NA              NA              NA              NA
## [33] NA              NA              NA              NA
## [37] NA              NA              NA              NA
## [41] NA              NA              NA              NA
## [45] NA              NA              NA              NA
## [49] NA              NA
```

Pada tampilan data dengan syntax ini kita bisa melihat urutan data yang hilang dengan deskripsi Mengatasi mising value dengan mengganti dengan nilai rata-rata yang mana untuk tipe data numerik

```
rawdata = data
for(i in 1:ncol(rawdata)) {
  rawdata[is.na(rawdata[,i]), i] <-  mean(rawdata[,i], na.rm = TRUE)
}
```

```
## Warning in mean.default(rawdata[, i], na.rm = TRUE): argument is not numeri
c or
## logical: returning NA

## Warning in mean.default(rawdata[, i], na.rm = TRUE): argument is not numeri
c or
## logical: returning NA

## Warning in mean.default(rawdata[, i], na.rm = TRUE): argument is not numeri
c or
## logical: returning NA

## Warning in mean.default(rawdata[, i], na.rm = TRUE): argument is not numeri
c or
## logical: returning NA

## Warning in mean.default(rawdata[, i], na.rm = TRUE): argument is not numeri
c or
## logical: returning NA
```

```
## Warning in mean.default(rawdata[, i], na.rm = TRUE): argument is not numeri
c or
## logical: returning NA

## Warning in mean.default(rawdata[, i], na.rm = TRUE): argument is not numeri
c or
## logical: returning NA

View(rawdata)
```

Filter

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindSpeed3pm	Humidity9am
1	2008-12-01	Albury	13.4	22.9	0.600000	5.468232	7.611178	W	44.00000	W	WNW	20.00000	24.00000	
2	2008-12-02	Albury	7.4	25.1	0.000000	5.468232	7.611178	WNW	44.00000	NNW	WSW	4.00000	22.00000	
3	2008-12-03	Albury	12.9	25.7	0.000000	5.468232	7.611178	WSW	46.00000	W	WSW	19.00000	26.00000	
4	2008-12-04	Albury	9.2	28.0	0.000000	5.468232	7.611178	NE	24.00000	SE	E	11.00000	9.00000	
5	2008-12-05	Albury	17.5	32.3	1.000000	5.468232	7.611178	W	41.00000	ENE	NW	7.00000	20.00000	
6	2008-12-06	Albury	14.6	29.7	0.200000	5.468232	7.611178	WNW	56.00000	W	W	19.00000	24.00000	
7	2008-12-07	Albury	14.3	25.0	0.000000	5.468232	7.611178	W	50.00000	SW	W	20.00000	24.00000	
8	2008-12-08	Albury	7.7	26.7	0.000000	5.468232	7.611178	W	35.00000	SSE	W	6.00000	17.00000	
9	2008-12-09	Albury	9.7	31.9	0.000000	5.468232	7.611178	NNW	80.00000	SE	NW	7.00000	28.00000	
10	2008-12-10	Albury	13.1	30.1	1.400000	5.468232	7.611178	W	28.00000	S	SSE	15.00000	11.00000	
11	2008-12-11	Albury	13.4	30.4	0.000000	5.468232	7.611178	N	30.00000	SSE	ESE	17.00000	6.00000	
12	2008-12-12	Albury	15.9	21.7	2.200000	5.468232	7.611178	NNE	31.00000	NE	ENE	15.00000	13.00000	
13	2008-12-13	Albury	15.9	18.6	15.600000	5.468232	7.611178	W	61.00000	NNW	NNW	28.00000	28.00000	
14	2008-12-14	Albury	12.6	21.0	3.600000	5.468232	7.611178	SW	44.00000	W	SSW	24.00000	20.00000	
15	2008-12-15	Albury	8.4	24.6	0.000000	5.468232	7.611178	NA	40.03523	S	WNW	4.00000	30.00000	
16	2008-12-16	Albury	9.8	27.7	2.360918	5.468232	7.611178	WNW	50.00000	NA	WNW	14.04343	22.00000	
17	2008-12-17	Albury	14.1	20.9	0.000000	5.468232	7.611178	ENE	22.00000	SSW	E	11.00000	9.00000	
18	2008-12-18	Albury	13.5	22.9	16.800000	5.468232	7.611178	W	63.00000	N	WNW	6.00000	20.00000	
19	2008-12-19	Albury	11.2	22.5	10.600000	5.468232	7.611178	SSE	43.00000	WSW	SW	24.00000	17.00000	
20	2008-12-20	Albury	9.8	25.6	0.000000	5.468232	7.611178	SSE	26.00000	SE	NNW	17.00000	6.00000	

pada kasus ini saya mengganti nilai nilai yang kosong dengan rata0rata data untuk data numerik, dikarenakan rata-rata sangat cocok untuk data numerik, dan untuk data kosong pada tipe data karakter akan tetap dibiarkan menjadi NA Value terdapat warning messsage dikarenakan jika terdapat data kosong pada tipe data selain numerik maka akan dikembalikkan kembali dalam bentuk NA

mengatasi data yang kosong dengan menghapus nilai data yang terdapat NA

```
rawdata3 = rawdata
rawdata3 <- na.omit(rawdata)
```

```
View(rawdata3)
```

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindSpeed3pm	Humidity9am
1	2008-12-01	Albury	13.4	22.9	0.6	5.468232	7.611178	W	44	W	WNW	20	24	
2	2008-12-02	Albury	7.4	25.1	0.0	5.468232	7.611178	WNW	44	NNW	WSW	4	22	
3	2008-12-03	Albury	12.9	25.7	0.0	5.468232	7.611178	WSW	46	W	WSW	19	26	
4	2008-12-04	Albury	9.2	28.0	0.0	5.468232	7.611178	NE	24	SE	E	11	9	
5	2008-12-05	Albury	17.5	32.3	1.0	5.468232	7.611178	W	41	ENE	NW	7	20	
6	2008-12-06	Albury	14.6	29.7	0.2	5.468232	7.611178	WNW	56	W	W	19	24	
7	2008-12-07	Albury	14.3	25.0	0.0	5.468232	7.611178	W	50	SW	W	20	24	
8	2008-12-08	Albury	7.7	26.7	0.0	5.468232	7.611178	W	35	SSE	W	6	17	
9	2008-12-09	Albury	9.7	31.9	0.0	5.468232	7.611178	NNW	80	SE	NW	7	28	
10	2008-12-10	Albury	13.1	30.1	1.4	5.468232	7.611178	W	28	S	SSE	15	11	
11	2008-12-11	Albury	13.4	30.4	0.0	5.468232	7.611178	N	30	SSE	ESE	17	6	
12	2008-12-12	Albury	15.9	21.7	2.2	5.468232	7.611178	NNE	31	NE	ENE	15	13	
13	2008-12-13	Albury	15.9	18.6	15.6	5.468232	7.611178	W	61	NNW	NNW	28	28	
14	2008-12-14	Albury	12.6	21.0	3.6	5.468232	7.611178	SW	44	W	SSW	24	20	
17	2008-12-17	Albury	14.1	20.9	0.0	5.468232	7.611178	ENE	22	SSW	E	11	9	
18	2008-12-18	Albury	13.5	22.9	16.8	5.468232	7.611178	W	63	N	WNW	6	20	
19	2008-12-19	Albury	11.2	22.5	10.6	5.468232	7.611178	SSE	43	WSW	SW	24	17	
20	2008-12-20	Albury	9.8	25.6	0.0	5.468232	7.611178	SSE	26	SE	NNW	17	6	

data yang masih kosong atau data karakter yang kosong yang mana akan dihapus karena untuk mengurangi kesalahan dalam pengolahan data nantinya

pada data ini disiapkan yang mana akan diambil variabel yang terbaik untuk nantinya bisa menjadi data diolah

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

#create decision tree c5.0 tree based model
rawdata3 = rawdata3 %>%
  mutate(across(where(is.character), as.factor))
str(rawdata3)

## 'data.frame':    123710 obs. of  23 variables:
##  $ Date         : Factor w/ 3417 levels "2007-11-01","2007-11-02",..: 378 3
79 380 381 382 383 384 385 386 387 ...
##  $ Location     : Factor w/ 47 levels "Adelaide","Albury",..: 2 2 2 2 2 2 2
 2 2 2 ...
##  $ MinTemp      : num  13.4 7.4 12.9 9.2 17.5 14.6 14.3 7.7 9.7 13.1 ...
##  $ MaxTemp      : num  22.9 25.1 25.7 28 32.3 29.7 25 26.7 31.9 30.1 ...
##  $ Rainfall     : num  0.6 0 0 0 1 0.2 0 0 0 1.4 ...
##  $ Evaporation  : num  5.47 5.47 5.47 5.47 5.47 ...
##  $ Sunshine     : num  7.61 7.61 7.61 7.61 7.61 ...
```

```
##  $ WindGustDir  : Factor w/ 16 levels "E","ENE","ESE",..: 14 15 16 5 14 15
14 14 7 14 ...
##  $ WindGustSpeed: num  44 44 46 24 41 56 50 35 80 28 ...
##  $ WindDir9am   : Factor w/ 16 levels "E","ENE","ESE",..: 14 7 14 10 2 14 1
3 11 10 9 ...
##  $ WindDir3pm   : Factor w/ 16 levels "E","ENE","ESE",..: 15 16 16 1 8 14 1
4 14 8 11 ...
##  $ WindSpeed9am : num  20 4 19 11 7 19 20 6 7 15 ...
##  $ WindSpeed3pm : num  24 22 26 9 20 24 24 17 28 11 ...
##  $ Humidity9am  : num  71 44 38 45 82 55 49 48 42 58 ...
##  $ Humidity3pm  : num  22 25 30 16 33 23 19 19 9 27 ...
##  $ Pressure9am  : num  1008 1011 1008 1018 1011 ...
##  $ Pressure3pm  : num  1007 1008 1009 1013 1006 ...
##  $ Cloud9am     : num  8 4.45 4.45 4.45 7 ...
##  $ Cloud3pm     : num  4.51 4.51 2 4.51 8 ...
##  $ Temp9am      : num  16.9 17.2 21 18.1 17.8 20.6 18.1 16.3 18.3 20.1 ...
##  $ Temp3pm      : num  21.8 24.3 23.2 26.5 29.7 28.9 24.6 25.5 30.2 28.2 ..
.
##  $ RainToday    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 2 ...
##  $ RainTomorrow : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 2 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:21750] 15 16 27 65 66 76 87 89
 90 96 ...
##   ..- attr(*, "names")= chr [1:21750] "15" "16" "27" "65" ...
```

disini variabel karakter diubah menjadi tipe data faktor

```
predictor <- RainTomorrow~MinTemp+MaxTemp+Rainfall+Evaporation+Sunshine+WindGu
stSpeed+WindDir3pm+WindDir9am+WindSpeed3pm+WindSpeed9am+Humidity3pm+Pressure3p
m+Cloud9am+Cloud3pm+Temp3pm+Temp9am+RainToday
```

membuat variabel predictor yang mana isinya terdapat variabel-variabel pada data cuaca di australia ,data raintomrrorw mejadi kelas

```
set.seed(1234)
#apply fold validation
fold <- cut(seq(1, nrow(rawdata3)), breaks = 10, labels=FALSE)
for(i in 1:10){
  testindexes <- which(fold==i, arr.ind = TRUE)
  testdata <- rawdata3[testindexes,]
  traindata <- rawdata3[-testindexes, ]}
```

menguji data dengan dataset dibagi menjadi 10 bagian, setiap bagian akan diuji ke seluruhan model lalu dihitung akurasi yang nantinya akan dijumlahkan dan di cek rata-rata akurasi tersebut, dengan metode ten fold validation akan membuat data lebih baik saat dicek akurasinya dalam pemodelan dibandingkan dengan pemodelan split 80/20

```
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3
.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.0.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts
() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(tidyrules)
library(C50)

## Warning: package 'C50' was built under R version 4.1.2

library(pander)

## Warning: package 'pander' was built under R version 4.1.2

library(dplyr)

#create decision tree menggunakan algoritma C5.0
treec5 <- C5.0(predictor, data=traindata)
treec5

##
## Call:
## C5.0.formula(formula = predictor, data = traindata)
##
## Classification Tree
## Number of samples: 111339
## Number of predictors: 17
##
## Tree size: 1715
##
## Non-standard options: attempt to group attributes

summary(treec5)

##
## Call:
## C5.0.formula(formula = predictor, data = traindata)
##
##
## C5.0 [Release 2.07 GPL Edition]      Sun Dec 05 08:51:24 2021
## -------------------------------
##
## Class specified by attribute `outcome'
##
```

```
## Read 111339 cases (18 attributes) from undefined.data
##
## Decision tree:
##
## Humidity3pm <= 66:
## :...Sunshine <= 7.5:
## :   :...Pressure3pm <= 1014.3:
## :   :   :...WindGustSpeed > 54:
## :   :   :   :...Humidity3pm > 52:
## :   :   :   :   :...Pressure3pm <= 998.1: Yes (80/9)
## :   :   :   :   :   Pressure3pm > 998.1:
## :   :   :   :   :   :...WindDir3pm in {ENE,NE,S,W,WNW}: Yes (258/76)
## :   :   :   :   :       WindDir3pm in {ESE,SSE}: No (32/12)
## :   :   :   :   :       WindDir3pm = NNE:
## :   :   :   :   :       :...Pressure3pm <= 1005.3: No (10/4)
## :   :   :   :   :       :   Pressure3pm > 1005.3: Yes (17/1)
## :   :   :   :   :       WindDir3pm = WSW:
## :   :   :   :   :       :...Humidity3pm <= 54: No (8/2)
## :   :   :   :   :       :   Humidity3pm > 54: Yes (71/17)
## :   :   :   :   :       WindDir3pm = E:

##
## SubTree [S79]
##
## WindSpeed3pm > 26: No (8)
## WindSpeed3pm <= 26:
## :...Sunshine <= 3.5: No (7)
##     Sunshine > 3.5:
##     :...WindDir3pm in {E,ENE,ESE,N,NE,NNE,NNW,NW,S,SE,SSE,SSW,
##         :              WSW}: Yes (16/3)
##         WindDir3pm = SW:
##         :...Pressure3pm <= 1017.7: Yes (2)
##         :   Pressure3pm > 1017.7: No (3)
##         WindDir3pm = W:
##         :...WindSpeed3pm <= 20: No (5)
##         :   WindSpeed3pm > 20: Yes (4/1)
##         WindDir3pm = WNW:
##         :...Humidity3pm <= 72: No (6)
##             Humidity3pm > 72: Yes (3)
##
##
## Evaluation on training data (111339 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##    1713 12883(11.6%)   <<
##
##
```

```
##     (a)   (b)    <-classified as
##    ----  ----
##   82930  3244    (a): class No
##    9639 15526    (b): class Yes
##
##
##  Attribute usage:
##
##  100.00% Humidity3pm
##   90.19% Sunshine
##   89.52% WindGustSpeed
##   66.22% RainToday
##   41.63% Pressure3pm
##   29.08% Rainfall
##   15.97% WindDir9am
##   15.37% WindDir3pm
##   14.30% MinTemp
##   12.62% Cloud3pm
##   10.61% WindSpeed3pm
##    9.57% Temp3pm
##    8.38% Temp9am
##    5.52% WindSpeed9am
##    3.97% Evaporation
##    3.55% Cloud9am
##    1.90% MaxTemp
##
##
## Time: 5.0 secs
```

pada output kali ini akan menghasilkan atribut-atribut yang penting dari total keseluruhan data, bisa kita lihat bahwa atribut yang paling penting ddalam data ini yaitu atribut humidity3pm dengan persentase 100% atau bisa dibilang tingkat pengaruh data humadity3pm dengan data lainnya berpengaruh 100%, attribut yang kedua yaitu atribut sunshime dengan persentase 90,19% atau bisa dibilang tingkat pengaruh data sunshine dengan data lainnya berpengaruh 90,19%, dan data ketiga yaitu Raintoday dengan persentase sebesar 66,22% atau bisa dibilang data Raintoday tingkat pengaruh dengan data lainnya berpengaruh 66,22%, data dengan tingkat pengaruh terrendah yaitu data Maxtemp dengan persentase 1.90% atau data Maxtemp hanya memiliki pengaruh dengan data lainnya sebesar 1.90% terlihat data yang keluar akan mengurut dari yang terbesar pengaruhnya ke data lainnya yaitu data Humadity3pm hingga data yang sedikit pengaruhnya dengan data lainnya yang paling bawah yaitu Atribut Maxtemp, dari data ini saya akan mengambil 10 data yang memiliki pengaruh tinggi yang mana dari 10 data ini akan memudahkan pengolahan data nantinya

```
library(dplyr)
labelsWeather= c("Humidity3pm","Sunshine","WindGustSpeed", "RainToday","Pressu
re3pm", "Rainfall", "WindDir9am", "WindDir3pm")
weather_new = rawdata3 %>%
  select(labelsWeather)

## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(labelsWeather)` instead of `labelsWeather` to silence this me
ssage.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.

weather_new = weather_new %>%
  mutate(across(where(is.double), as.integer))

View(weather_new)
attach(weather_new)
```

	Humidity3pm	Sunshine	WindGustSpeed	RainToday	Pressure3pm	Rainfall	WindDir9am	WindDir3pm
1	22	7	44	No	1007	0	W	WNW
2	25	7	44	No	1007	0	NNW	WSW
3	30	7	46	No	1008	0	W	WSW
4	16	7	24	No	1012	0	SE	E
5	33	7	41	No	1006	1	ENE	NW
6	23	7	56	No	1005	0	W	W
7	19	7	50	No	1008	0	SW	W
8	19	7	35	No	1010	0	SSE	W
9	9	7	80	No	1003	0	SE	NW
10	27	7	28	Yes	1005	1	S	SSE
11	22	7	30	No	1008	0	SSE	ESE
12	91	7	31	Yes	1004	2	NE	ENE
13	93	7	61	Yes	993	15	NNW	NNW
14	43	7	44	Yes	1001	3	W	SSW
17	82	7	22	No	1010	0	SSW	E
18	65	7	63	Yes	1002	16	N	WNW
19	32	7	43	Yes	1009	10	WSW	SW
20	26	7	26	No	1017	0	SE	NNW

menyimpan data yang sudah diolah kedalam CSV

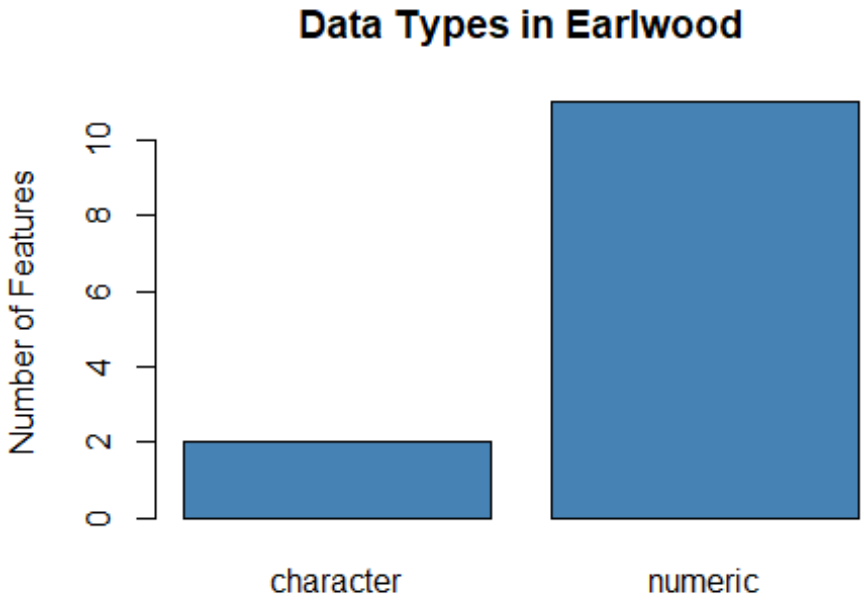
```
write.csv(rawdata3,"D:/Hagan/PENS/Project/R/Post Test/ weather update.csv")
```

Earlwood

Hagan

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(openxlsx)
library(readxl)
Earlwood = read_xls("D:/Hagan/PENS/Project/R/Post Test/Earlwood_Air_Data_17_18
.xls")
View(Earlwood)
str(Earlwood)

## tibble [8,784 x 13] (S3: tbl_df/tbl/data.frame)
##  $ Date                                      : chr [1:8784] "01/01/2017" "01/
01/2017" "01/01/2017" "01/01/2017" ...
##  $ Time                                      : chr [1:8784] "01:00" "02:00" "
03:00" "04:00" ...
##  $ EARLWOOD WDR 1h average [°]               : num [1:8784] 152 134 132 126 1
08 ...
##  $ EARLWOOD TEMP 1h average [°C]             : num [1:8784] 22.6 22.6 22.6 22
.7 22.8 23 23.6 24.3 24.6 25.2 ...
##  $ EARLWOOD WSP 1h average [m/s]             : num [1:8784] 0.4 0.3 0.3 0.2 0
.6 0.5 0.5 0.4 0.4 1 ...
##  $ EARLWOOD NO 1h average [pphm]             : num [1:8784] 0 NA 0 0 0 0 0 0
0 0 ...
##  $ EARLWOOD NO2 1h average [pphm]            : num [1:8784] 0.4 NA 0.6 0.5 0.
3 0.3 0.3 0.3 0.5 0.5 ...
##  $ EARLWOOD OZONE 1h average [pphm]          : num [1:8784] 2 NA 1.7 1.7 2.1
2.1 2.5 3 2.7 2.7 ...
##  $ EARLWOOD OZONE 4h rolling average [pphm]: num [1:8784] 2.1 2.2 2 1.8 1.8
 1.9 2.1 2.4 2.6 2.7 ...
##  $ EARLWOOD PM10 1h average [µg/m³]          : num [1:8784] 23.6 21 20 21.4 2
1.5 23.5 16.8 16.5 19.9 20.3 ...
##  $ EARLWOOD PM2.5 1h average [µg/m³]         : num [1:8784] 7 6.6 7.2 7.1 4.3
 8.6 5.9 8 7.8 6 ...
```

```
##  $ EARLWOOD HUMID 1h average [%]           : num [1:8784] 87.2 87.2 87 87.2
 86.8 85.7 80.2 74.9 74.8 73.5 ...
##  $ EARLWOOD SD1 1h average [°]             : num [1:8784] 49 46.6 47.4 53.7
 41.6 ...

data_types <- function(frame) {
  res <- lapply(frame, class)
  res_frame <- data.frame(unlist(res))
  barplot(table(res_frame), main="Data Types in Earlwood", col="steelblue", yl
ab="Number of Features")
}
data_types(Earlwood)
```



(Humid) kelembaban (SD) strom dati (PM) partikulat, PM10 = 150 µgram/m, NAB PM2.5 = 65 µgram/m3 (NO2) nitrogen oksida (NO) nitrogen monoksida (wsp) weather system processor (wdr)wind direction degree

```
#Earlwood %>% filter(Time == c("06:00"),`EARLWOOD TEMP 1h average [°C]`>= 23)

#Earlwood %>% filter(Time == c("20:00"),`EARLWOOD TEMP 1h average [°C]`>= 28)

#Earlwood %>% filter(`EARLWOOD WDR 1h average [°]`>= 100 )
```

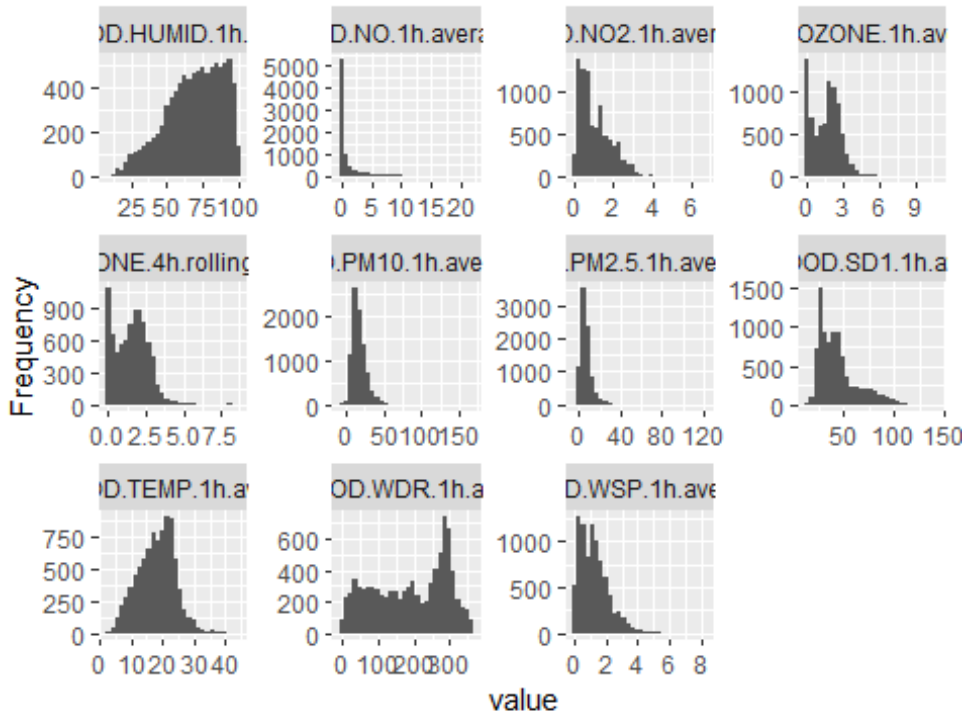
nitrogen oksida

```
#arrange(Earlwood, desc("EARLWOOD OZONE 1h average [pphm]") )

#sum(is.na(Earlwood))/(nrow(Earlwood)*ncol(Earlwood))
```

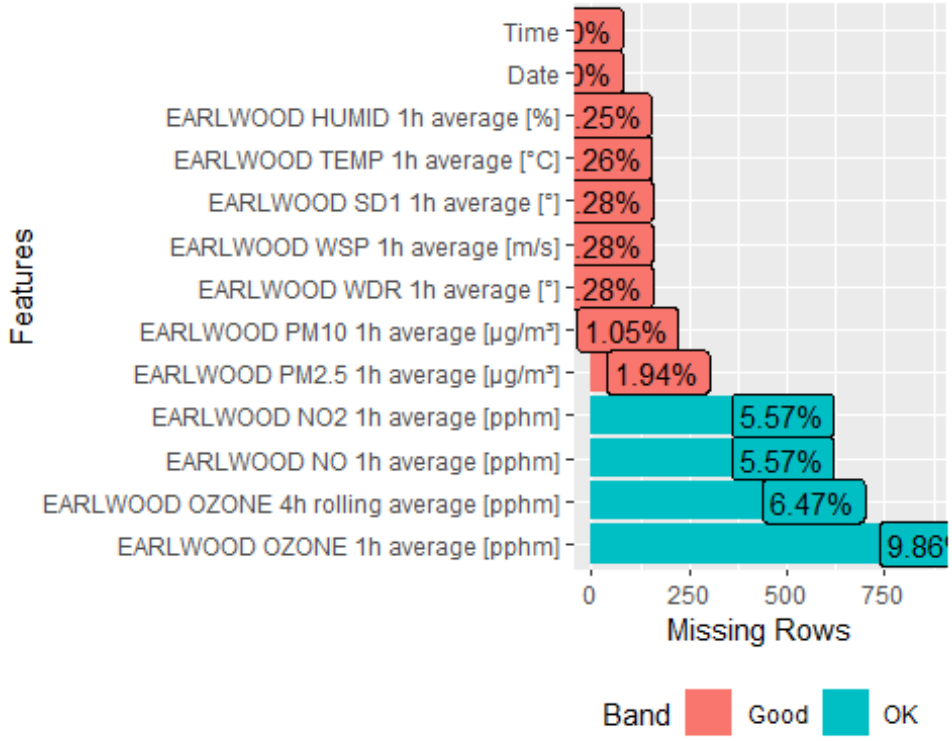
nacols <- function(df) { colnames(df)[unlist(lapply(df, function(x) anyNA(x)))] } cat('There are',length(nacols(df)),'columns with NA values.50% of columns are NA filled which disturbs the data quality')

missing_data <- as.data.frame(sort(sapply(Earlwood, function(x) sum(is.na(x))),decreasing = T)) colnames(missing_data)[1] <- "Missing_values" missing_data$Percentage <- -(missing_data$Missing_values/nrow(data))*100
missing_data$Variables <- rownames(missing_data) missing_data <- missing_data[c(3,1,2)]
rownames(missing_data)<-c()
head(missing_data,15)

```
library(DataExplorer)

## Warning: package 'DataExplorer' was built under R version 4.1.2

introduce(Earlwood)

## # A tibble: 1 x 9
##    rows columns discrete_columns continuous_columns all_missing_columns
##   <int>   <int>            <int>              <int>               <int>
## 1  8784      13                2                 11                   0
## # ... with 4 more variables: total_missing_values <int>, complete_rows <int
>,
## #   total_observations <int>, memory_usage <dbl>

plot_histogram(Earlwood)
```



```
plot_missing(Earlwood)
```




```
summary(Earlwood)

##      Date               Time           EARLWOOD WDR 1h average [°]
##  Length:8784        Length:8784        Min.   :  0.1
##  Class :character   Class :character   1st Qu.: 98.3
##  Mode  :character   Mode  :character   Median :201.4
##                                        Mean   :190.9
##                                        3rd Qu.:283.5
##                                        Max.   :360.0
##                                        NA's   :25
##  EARLWOOD TEMP 1h average [°C] EARLWOOD WSP 1h average [m/s]
##  Min.   : 2.60                 Min.   :0.000
##  1st Qu.:14.10                 1st Qu.:0.500
##  Median :18.60                 Median :1.100
##  Mean   :18.22                 Mean   :1.255
##  3rd Qu.:22.30                 3rd Qu.:1.800
##  Max.   :43.90                 Max.   :8.200
##  NA's   :23                    NA's   :25
##  EARLWOOD NO 1h average [pphm] EARLWOOD NO2 1h average [pphm]
##  Min.   :-0.100                Min.   :0.00
##  1st Qu.: 0.000                1st Qu.:0.40
##  Median : 0.200                Median :0.90
##  Mean   : 1.288                Mean   :1.08
##  3rd Qu.: 1.100                3rd Qu.:1.60
##  Max.   :21.700                Max.   :6.70
```

```
##  NA's   :489                    NA's   :489
##  EARLWOOD OZONE 1h average [pphm] EARLWOOD OZONE 4h rolling average [pphm]
##  Min.   : 0.000                   Min.   :0.000
##  1st Qu.: 0.500                   1st Qu.:0.600
##  Median : 1.700                   Median :1.600
##  Mean   : 1.633                   Mean   :1.609
##  3rd Qu.: 2.400                   3rd Qu.:2.300
##  Max.   :10.900                   Max.   :8.700
##  NA's   :866                      NA's   :568
##  EARLWOOD PM10 1h average [µg/m³] EARLWOOD PM2.5 1h average [µg/m³]
##  Min.   : -8.80                   Min.   : -2.500
##  1st Qu.: 11.30                   1st Qu.:  3.500
##  Median : 16.10                   Median :  5.900
##  Mean   : 18.04                   Mean   :  7.295
##  3rd Qu.: 22.90                   3rd Qu.:  9.400
##  Max.   :164.20                   Max.   :122.900
##  NA's   :92                       NA's   :170
##  EARLWOOD HUMID 1h average [%] EARLWOOD SD1 1h average [°]
##  Min.   :  9.60                Min.   : 12.62
##  1st Qu.: 55.90                1st Qu.: 28.61
##  Median : 71.40                Median : 39.66
##  Mean   : 69.26                Mean   : 44.18
##  3rd Qu.: 85.50                3rd Qu.: 51.70
##  Max.   :100.30                Max.   :142.81
##  NA's   :22                    NA's   :25
```

most_na_columns_Earlwood<-missing_data$Variables[1:50] most_na_columns_Earlwood

Mengatasi mising value dengan mengganti dengan nilai rata-rata yang mana untuk tipe data numerik

```
rawEarlwood = Earlwood
for(i in 3:ncol(rawEarlwood)) {
  rawEarlwood[is.na(rawEarlwood[,i]), i] <-  mean(rawEarlwood[,i], na.rm = TRU
E)
}

## Warning in mean.default(rawEarlwood[, i], na.rm = TRUE): argument is not nu
meric
## or logical: returning NA

## Warning in mean.default(rawEarlwood[, i], na.rm = TRUE): argument is not nu
meric
## or logical: returning NA

## Warning in mean.default(rawEarlwood[, i], na.rm = TRUE): argument is not nu
meric
## or logical: returning NA

## Warning in mean.default(rawEarlwood[, i], na.rm = TRUE): argument is not nu
```

```
meric
## or logical: returning NA

## Warning in mean.default(rawEarlwood[, i], na.rm = TRUE): argument is not nu
meric
## or logical: returning NA

## Warning in mean.default(rawEarlwood[, i], na.rm = TRUE): argument is not nu
meric
## or logical: returning NA

## Warning in mean.default(rawEarlwood[, i], na.rm = TRUE): argument is not nu
meric
## or logical: returning NA

## Warning in mean.default(rawEarlwood[, i], na.rm = TRUE): argument is not nu
meric
## or logical: returning NA

## Warning in mean.default(rawEarlwood[, i], na.rm = TRUE): argument is not nu
meric
## or logical: returning NA

## Warning in mean.default(rawEarlwood[, i], na.rm = TRUE): argument is not nu
meric
## or logical: returning NA

## Warning in mean.default(rawEarlwood[, i], na.rm = TRUE): argument is not nu
meric
## or logical: returning NA

View(rawEarlwood)
```

terdapat warning messsage dikarenakan jika terdapat data kosong pada tipe data selain numerik maka akan dikembalikkan kembali dalam bentuk NA

mengatasi data yang kosong dengan menghapus nilai data yang terdapat NA

```
rawEarlwood3 = rawEarlwood
rawEarlwood3 <- na.omit(rawEarlwood)

View(rawEarlwood3)

labelsEarlwood= c("EARLWOOD WDR 1h average [°]","EARLWOOD TEMP 1h average [°C]
","EARLWOOD WSP 1h average [m/s]", "EARLWOOD WSP 1h average [m/s]","EARLWOOD N
O 1h average [pphm]", "EARLWOOD NO2 1h average [pphm]", "EARLWOOD NO2 1h avera
ge [pphm]", "EARLWOOD OZONE 1h average [pphm]","EARLWOOD OZONE 4h rolling aver
age [pphm]", "EARLWOOD PM10 1h average [µg/m³]","EARLWOOD PM2.5 1h average [µg
/m³]","EARLWOOD HUMID 1h average [%]","EARLWOOD SD1 1h average [°]")
```

```
Earlwoodupdate = rawEarlwood3 %>%
  select(labelsEarlwood)

## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(labelsEarlwood)` instead of `labelsEarlwood` to silence this
message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.

View(Earlwoodupdate)

library(dplyr)
#library(ggplot2)
library(readr)
```

normalisasi data atribut agar menjadi range dari 0 hingga 1

```
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

`EARLWOOD.WDR.1h.average.[°]` <- normalize(Earlwoodupdate$`EARLWOOD.WDR.1h.ave
rage.[°]`)

## Warning: Unknown or uninitialised column: `EARLWOOD.WDR.1h.average.[°]`.

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

`EARLWOOD.TEMP.1h.average.[°C]` <- normalize(Earlwoodupdate$`EARLWOOD.TEMP.1h.
average.[°C]`)

## Warning: Unknown or uninitialised column: `EARLWOOD.TEMP.1h.average.[°C]`.

## Warning: no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

`EARLWOOD.WSP.1h.average.[m/s]` <- normalize(Earlwoodupdate$`EARLWOOD.WSP.1h.a
verage.[m/s]`)

## Warning: Unknown or uninitialised column: `EARLWOOD.WSP.1h.average.[m/s]`.

## Warning: no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf
```

```
`EARLWOOD.NO.1h.average.[pphm]` <- normalize(Earlwoodupdate$`EARLWOOD.NO.1h.average.[pphm]`)

## Warning: Unknown or uninitialised column: `EARLWOOD.NO.1h.average.[pphm]`.

## Warning: no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

`EARLWOOD.NO2.1h.average.[pphm]` <- normalize(Earlwoodupdate$`EARLWOOD.NO2.1h.average.[pphm]`)

## Warning: Unknown or uninitialised column: `EARLWOOD.NO2.1h.average.[pphm]`.

## Warning: no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

`EARLWOOD.OZONE.4h.rolling.average.[pphm]` <- normalize(Earlwoodupdate$`EARLWOOD.OZONE.4h.rolling.average.[pphm]`)

## Warning: Unknown or uninitialised column: `EARLWOOD.OZONE.4h.rolling.averag
e.
## [pphm]`.

## Warning: no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

`EARLWOOD.PM10.1h.average.[µg/m³]` <- normalize(Earlwoodupdate$`EARLWOOD.PM10.1h.average.[µg/m³]`)

## Warning: Unknown or uninitialised column: `EARLWOOD.PM10.1h.average.[µg/m³]
`.

## Warning: no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

`EARLWOOD.PM2.5.1h.average.[µg/m³]` <- normalize(Earlwoodupdate$`EARLWOOD.PM2.5.1h.average.[µg/m³]`)

## Warning: Unknown or uninitialised column: `EARLWOOD.PM2.5.1h.average.[µg/m³
]`.
```

```
## Warning: no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

`EARLWOOD.HUMID.1h.average.[%]` <- normalize(Earlwoodupdate$`EARLWOOD.HUMID.1h
.average.[%]`)

## Warning: Unknown or uninitialised column: `EARLWOOD.HUMID.1h.average.[%]`.

## Warning: no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

`EARLWOOD.SD1.1h.average.[°]` <- normalize(Earlwoodupdate$`EARLWOOD.SD1.1h.ave
rage.[°]`)

## Warning: Unknown or uninitialised column: `EARLWOOD.SD1.1h.average.[°]`.

## Warning: no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

`EARLWOOD.OZONE.1h.average.[pphm]`<- normalize(Earlwoodupdate$`EARLWOOD.OZONE.
1h.average.[pphm]`)

## Warning: Unknown or uninitialised column: `EARLWOOD.OZONE.1h.average.[pphm]
`.

## Warning: no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

View(Earlwoodupdate)

#write.csv(Earlwoodupdate,"D:/Hagan/PENS/Project/R/Post Test/ Earlwood normali
ze.csv")
```

crypto_markets

Hagan

```
library(RMySQL)

## Warning: package 'RMySQL' was built under R version 4.1.2

## Loading required package: DBI

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(stringr)
mysqlconnection = dbConnect(MySQL(), user="root", password="hagan", dbname = "
crypto money", host="localhost")
crypto_markets = dbReadTable(mysqlconnection, "`crypto_markets`")
crypto_markets %>% head()

##          Slug Asset        Name       Date Ranknow    Open    High     Low
## 1        slug asset        name       date ranknow    open    high     low
## 2 target-coin   TGT Target Coin 09/29/2017     607 0.02896 0.05476 0.02896
## 3 target-coin   TGT Target Coin 09/30/2017     607 0.04178 0.04619 0.03143
## 4 target-coin   TGT Target Coin 10/01/2017     607 0.03176 0.03595 0.02104
## 5 target-coin   TGT Target Coin 10/02/2017     607 0.02837 0.05459 0.02041
## 6 target-coin   TGT Target Coin 10/03/2017     607 0.02252 0.03222 0.02021
##     Close Volume Market Clode_Ratio Spread
## 1   close volume market close_ratio spread
## 2 0.04177  69996      0      0.4966   0.03
## 3 0.03174   5725      0      0.0209   0.01
## 4 0.02838   5012      0      0.4924   0.01
## 5 0.02252   8010      0      0.0617   0.03
## 6 0.02035   1787      0      0.0123   0.01

attach(crypto_markets)
```

Pada ETL ini menggunakan data Crypto market yang mana sumber data didapatkan dari kaggle dan tipe data sql, menggunakan syntax my sql connection untuk membaca data tersebut menggunakan database yang telah tersimpan di databse lokalserver laptop lalu dibaca menggunakan dbreadtable lalu menampilkan data tersebut dengan view agar nantinya kita dapat menampilkan data di dalam R Studio

merubah tipe data

```
crypto_markets_update <- transform(crypto_markets,
                                   Open = as.double(Open),
                                   High = as.double(High),
                                   Ranknow = as.double(Ranknow),
                                   Low= as.double(Low),
                                   Close= as.double(Close),
                                   Volume= as.double(Volume),
                                   Market = as.double(Market),
                                   Clode_Ratio = as.double(Clode_Ratio),
                                   Spread= as.double(Spread))

## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs introd
uced
## by coercion

## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs introd
uced
## by coercion

## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs introd
uced
## by coercion

## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs introd
uced
## by coercion

## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs introd
uced
## by coercion

## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs introd
uced
## by coercion

## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs introd
uced
## by coercion

## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs introd
uced
## by coercion

## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs introd
uced
## by coercion
```

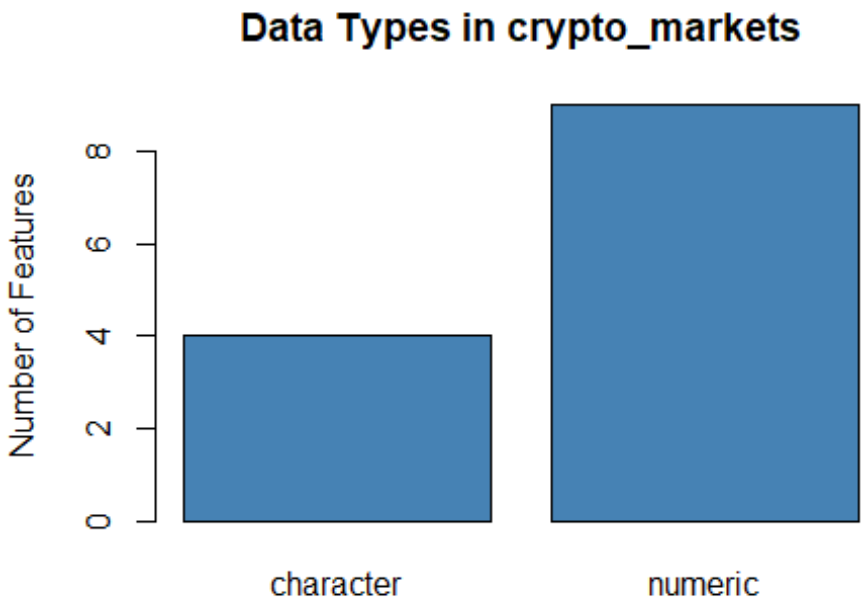
```
crypto_markets_update[crypto_markets == 0] <- NA

crypto_markets_update %>% head()

##          Slug Asset        Name       Date Ranknow    Open    High     Low
## 1        slug asset        name       date      NA      NA      NA      NA
## 2 target-coin   TGT Target Coin 09/29/2017     607 0.02896 0.05476 0.02896
## 3 target-coin   TGT Target Coin 09/30/2017     607 0.04178 0.04619 0.03143
## 4 target-coin   TGT Target Coin 10/01/2017     607 0.03176 0.03595 0.02104
## 5 target-coin   TGT Target Coin 10/02/2017     607 0.02837 0.05459 0.02041
## 6 target-coin   TGT Target Coin 10/03/2017     607 0.02252 0.03222 0.02021
##     Close Volume Market Clode_Ratio Spread
## 1      NA     NA     NA          NA     NA
## 2 0.04177  69996     NA      0.4966   0.03
## 3 0.03174   5725     NA      0.0209   0.01
## 4 0.02838   5012     NA      0.4924   0.01
## 5 0.02252   8010     NA      0.0617   0.03
## 6 0.02035   1787     NA      0.0123   0.01
```

pada server database data yang seharusnya terbaca sebagai integer tetapi terbaca sebagai data karakter, maka data diubah terlebih dahulu menjadi tipe data double dengan syntax transform agar data bisa kita olah karena berbentuk tipe data double, dan juga saya merubah data "0" menjadi data NA atau data kosong yang mana jikalau data terbentuk data "0" data tidak akan terbaca sebagai missing value

```
str(crypto_markets_update)

## 'data.frame':    1588 obs. of  13 variables:
##  $ Slug       : chr  "slug" "target-coin" "target-coin" "target-coin" ...
##  $ Asset      : chr  "asset" "TGT" "TGT" "TGT" ...
##  $ Name       : chr  "name" "Target Coin" "Target Coin" "Target Coin" ...
##  $ Date       : chr  "date" "09/29/2017" "09/30/2017" "10/01/2017" ...
##  $ Ranknow    : num  NA 607 607 607 607 607 607 607 607 607 ...
##  $ Open       : num  NA 0.029 0.0418 0.0318 0.0284 ...
##  $ High       : num  NA 0.0548 0.0462 0.036 0.0546 ...
##  $ Low        : num  NA 0.029 0.0314 0.021 0.0204 ...
##  $ Close      : num  NA 0.0418 0.0317 0.0284 0.0225 ...
##  $ Volume     : num  NA 69996 5725 5012 8010 ...
##  $ Market     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Clode_Ratio: num  NA 0.4966 0.0209 0.4924 0.0617 ...
##  $ Spread     : num  NA 0.03 0.01 0.01 0.03 0.01 0.01 0.01 0.01 0.01 ...
```

disini kita bisa melihat struktur data dan juga persebaran data dengan gambaran spesifik nama data tersebut dengan tipe data yang ada dan juga sedikit isi data yang diperlihatkan

```
data_types <- function(frame) {
  res <- lapply(frame, class)
  res_frame <- data.frame(unlist(res))
  barplot(table(res_frame), main="Data Types in crypto_markets", col="steelblu
e", ylab="Number of Features")
}
data_types(crypto_markets_update)
```



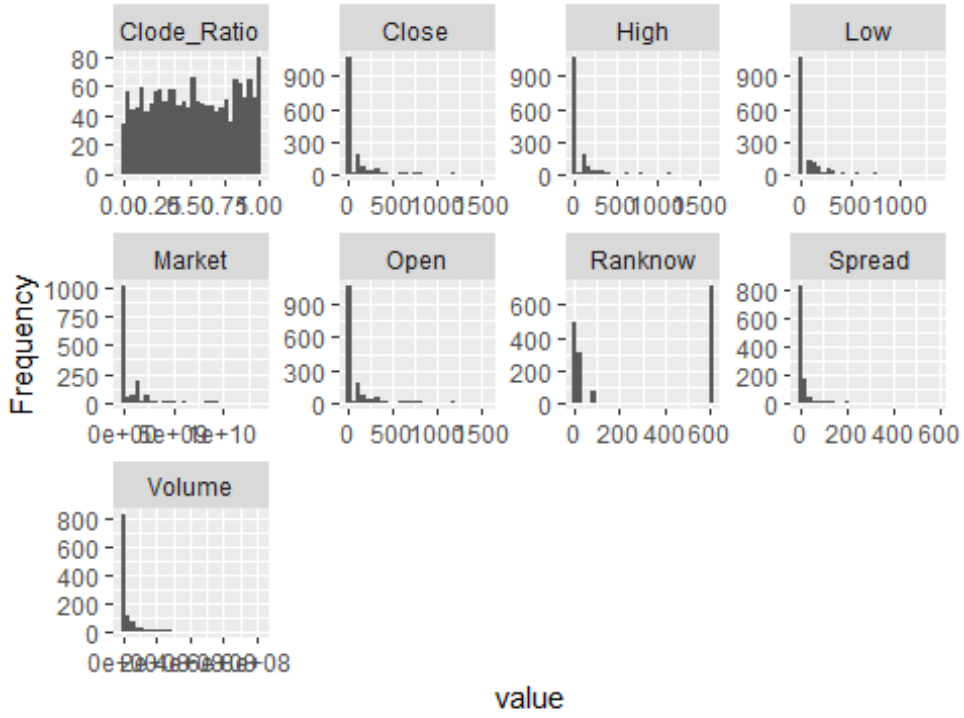

```
sum(is.na(crypto_markets_update))/(nrow(crypto_markets_update)*ncol(crypto_mar
kets_update))
```

```
## [1] 0.04887619
```

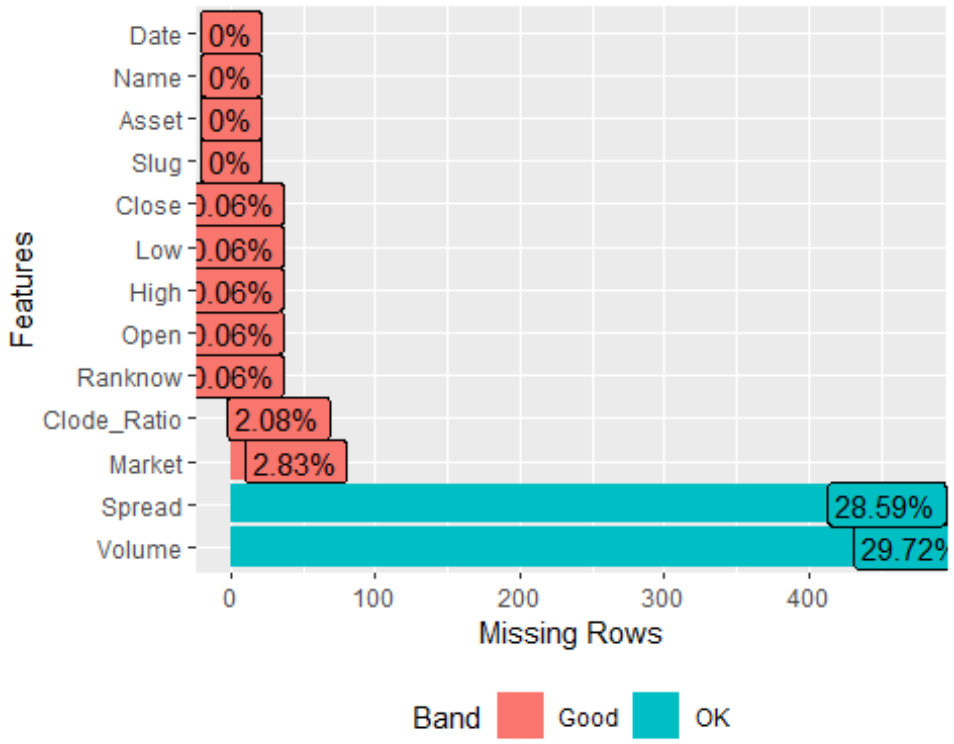
nacols <- function(df) { colnames(df)[unlist(lapply(df, function(x) anyNA(x)))] } cat('There are',length(nacols(df)),'columns with NA values.50% of columns are NA filled which disturbs the data quality')

missing_data <- as.data.frame(sort(sapply(crypto_markets_update, function(x) sum(is.na(x))),decreasing = T))
colnames(missing_data)[1] <- "Missing_values" missing_data$Percentage <- -(missing_data$Missing_values/nrow(data))*100
missing_data$Variables <- rownames(missing_data) missing_data <- missing_data[c(3,1,2)]
rownames(missing_data)<-c()
missing_data %>% head()

```
library(DataExplorer)

## Warning: package 'DataExplorer' was built under R version 4.1.2

introduce(crypto_markets_update)

##   rows columns discrete_columns continuous_columns all_missing_columns
## 1 1588      13                4                   9                   0
##   total_missing_values complete_rows total_observations memory_usage
## 1                 1009           742               20644       213384

plot_histogram(crypto_markets_update)
```



```
plot_missing(crypto_markets_update)
```




```
summary(crypto_markets_update)
##      Slug               Asset               Name               Date          
##  Length:1588        Length:1588        Length:1588        Length:1588       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##     Ranknow            Open                High                 Low          
##  Min.   :  1.0   Min.   :   0.0031   Min.   :   0.0032   Min.   :   0.0029  
##  1st Qu.:  6.0   1st Qu.:   0.0152   1st Qu.:   0.0184   1st Qu.:   0.0125  
##  Median : 13.0   Median :   0.5591   Median :   0.5991   Median :   0.4876  
##  Mean   :279.4   Mean   : 102.3470   Mean   : 108.0613   Mean   :  96.5417  
##  3rd Qu.:605.0   3rd Qu.: 115.8100   3rd Qu.: 121.6950   3rd Qu.: 110.9000  
##  Max.   :609.0   Max.   :1555.5900   Max.   :1642.2200   Max.   :1371.3900  
##  NA's   :1       NA's   :1           NA's   :1           NA's   :1          
```

```
##      Close                Volume               Market           Clode_Ratio
##  Min.   :   0.0031   Min.   :       14   Min.   :9.361e+05   Min.   :0.0006
##  1st Qu.:   0.0151   1st Qu.:     8612   1st Qu.:1.195e+07   1st Qu.:0.2631
##  Median :   0.5640   Median :   100952   Median :5.235e+07   Median :0.5175
##  Mean   : 103.1197   Mean   : 25670517   Mean   :9.337e+08   Mean   :0.5189
##  3rd Qu.: 116.4500   3rd Qu.: 18831100   3rd Qu.:1.239e+09   3rd Qu.:0.7944
##  Max.   :1550.8500   Max.   :816872000   Max.   :1.361e+10   Max.   :1.0000
##  NA's   :1           NA's   :472         NA's   :45          NA's   :33
##      Spread
##  Min.   :  0.01
##  1st Qu.:  0.02
##  Median :  0.46
##  Mean   : 16.12
##  3rd Qu.: 11.44
##  Max.   :588.08
##  NA's   :454
```

most_na_columns_crypto_markets_update<-missing_data$Variables[1:50]
most_na_columns_crypto_markets_update %>% head()

Mengatasi mising value dengan mengganti dengan nilai rata-rata yang mana untuk tipe data numerik

```
rawcrypto_markets_update = crypto_markets_update
for(i in 3:ncol(rawcrypto_markets_update)) {
  rawcrypto_markets_update[is.na(rawcrypto_markets_update[,i]), i] <-  mean(ra
wcrypto_markets_update[,i], na.rm = TRUE)
}

## Warning in mean.default(rawcrypto_markets_update[, i], na.rm = TRUE): argum
ent
## is not numeric or logical: returning NA

## Warning in mean.default(rawcrypto_markets_update[, i], na.rm = TRUE): argum
ent
## is not numeric or logical: returning NA
```

```
rawcrypto_markets_update %>% head()
```

```
##          Slug Asset        Name       Date  Ranknow      Open      High
## 1        slug asset        name       date 279.3705 102.34703 108.06126
## 2 target-coin   TGT Target Coin 09/29/2017 607.0000   0.02896   0.05476
## 3 target-coin   TGT Target Coin 09/30/2017 607.0000   0.04178   0.04619
## 4 target-coin   TGT Target Coin 10/01/2017 607.0000   0.03176   0.03595
## 5 target-coin   TGT Target Coin 10/02/2017 607.0000   0.02837   0.05459
## 6 target-coin   TGT Target Coin 10/03/2017 607.0000   0.02252   0.03222
##        Low     Close   Volume    Market Clode_Ratio   Spread
## 1 96.54167 103.11972 25670517 933671579   0.5189242 16.12096
## 2  0.02896   0.04177    69996 933671579   0.4966000  0.03000
## 3  0.03143   0.03174     5725 933671579   0.0209000  0.01000
## 4  0.02104   0.02838     5012 933671579   0.4924000  0.01000
## 5  0.02041   0.02252     8010 933671579   0.0617000  0.03000
## 6  0.02021   0.02035     1787 933671579   0.0123000  0.01000
```

terdapat warning messsage dikarenakan jika terdapat data kosong pada tipe data selain numerik maka akan dikembalikkan kembali dalam bentuk NA

mengatasi data yang kosong dengan menghapus nilai data yang terdapat NA

```
rawcrypto_markets_new = rawcrypto_markets_update
rawcrypto_markets_new <- na.omit(rawcrypto_markets_update)

library(dplyr)

labels= c("Asset","Name","Date", "Open","High", "Low", "Close")
dropcrypto_markets_new = rawcrypto_markets_update %>%
  select(labels)

## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(labels)` instead of `labels` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.

dropcrypto_markets_new$open <- dropcrypto_markets_new[,4]*0.8*15000
dropcrypto_markets_new$high <- dropcrypto_markets_new[,5]*0.8*15000
dropcrypto_markets_new$low <- dropcrypto_markets_new[,6]*0.8*15000
dropcrypto_markets_new$close <- dropcrypto_markets_new[,7]*0.8*15000

labels= c("Asset","Name","Date", "open","high", "low", "close")
crypto_markets_new = dropcrypto_markets_new %>%
  select(labels)
```

```
crypto_markets_new <- crypto_markets_new[crypto_markets_new$Asset != "asset",
] %>% head()

crypto_markets_new %>% head()

##   Asset        Name       Date   open   high    low  close
## 2   TGT Target Coin 09/29/2017 347.52 657.12 347.52 501.24
## 3   TGT Target Coin 09/30/2017 501.36 554.28 377.16 380.88
## 4   TGT Target Coin 10/01/2017 381.12 431.40 252.48 340.56
## 5   TGT Target Coin 10/02/2017 340.44 655.08 244.92 270.24
## 6   TGT Target Coin 10/03/2017 270.24 386.64 242.52 244.20
## 7   TGT Target Coin 10/04/2017 244.80 330.48 207.24 233.28
```

menyimpan data yang sudah diolah kedalam CSV

```
#write.csv(crypto_markets_new,"D:/Hagan/PENS/Project/R/Post Test/ crypto_marke
ts_new.csv")
```

➢ Evaluasi:

Pada program yang saya buat masih belum bisa membaca banyak data yang diharapkan seperti html, xml, json. Sehingga tidak banyak data yang bisa di eksplorasi dan kedepannya saya diharapkan bisa maksimal dalam pengolahan data tersebut

➢ Kesimpulan:

Pada program ini saya membaca tipe data xls, csv, sql dari file yang diinputkan yang mana tersebut masih berantakan lalu saya melakukan pengolahan data dengan syntax menggunakan Bahasa pemrograman R yang membuat data bisa dipakai untuk kebutuhan nantinya