

ozanevkaya

2 Followers Lists About

Bu verilerin NE'si var ?

 ozanevkaya 18 hours ago · 6 min read

Veri ile uğraşan herkesin kulağına defalarca fısıldanan bir laf vardır;

“Model kurma kısmı neyse de, verinin hazırlanması büyük eziyet”

Tabi iş veriyi hazırlamaya gelince, önce elde ne var onu anlamak en önemlisi. Verilerde bir yamukluk/eksiklik var mı, göze çarpan aykırı durumlar neler (tabi gözle bakmak sadece ilk adım), değişken türleri neler vb.

Bu yazıda, kendi çapımda debelenirken öğrendiğim, **Keşifsel Veri Analizi** (Exploratory Data Analysis) noktasında faydası olabilecek bazı **R paketlerinden** ve ilgili kaynaklardan bahsediyor olacağım. İlerleyen bölümlerde kısaca EDA diye gösterilecek olan bu önemli süreç için kullanılabilecek birçok paket bulmak mümkün.

Paketlerin genel kurulumu ve kullanımı için, malumunuz önce şuna ihtiyaç var;

```
`` `{r }
```

```
install.packages("pckgname")
```

```
library(pckgname)
```

```
`` `
```

Arkasından verilerin NE'si var anlayabilmek adına ilgili paketlerden fonksiyonları R (RStudio) içerisinde kullanmaya başlayabiliyoruz. Bu yazıda bahsedilmek istenen paketler kısaca şunlar ;

1. funModeling

Paket yazarlarının da ifadesiyle;

“This package contains a set of functions related to exploratory data analysis, data preparation, and model performance”

Sadece EDA için değil, başka işler için de kullanılabilir bir paket. Temel bazı fonksiyonlar için paketin içinde yer alan heart_disease verisine şöyle bir bakalım;

status(heart_disease) ile değişkenlerin;

- isimlerini
- içerdikleri sıfır ve kayıp değerleri (miktar ve oransal olarak)
- sınıflarını ve unique karşılıklarını

görmek mümkün;

```
> status(heart_disease)
```

	variable	q_zeros	p_zeros	q_na	p_na	q_inf	p_inf	type	unique
age	age	0	0.0000000	0	0.0000000	0	0	integer	41
gender	gender	0	0.0000000	0	0.0000000	0	0	factor	2
chest_pain	chest_pain	0	0.0000000	0	0.0000000	0	0	factor	4
resting_blood_pressure	resting_blood_pressure	0	0.0000000	0	0.0000000	0	0	integer	50
serum_cholesterol	serum_cholesterol	0	0.0000000	0	0.0000000	0	0	integer	152
fasting_blood_sugar	fasting_blood_sugar	258	0.8514851	0	0.0000000	0	0	factor	2
resting_electro	resting_electro	151	0.4983498	0	0.0000000	0	0	factor	3
max_heart_rate	max_heart_rate	0	0.0000000	0	0.0000000	0	0	integer	91
exer_angina	exer_angina	204	0.6732673	0	0.0000000	0	0	integer	2
oldpeak	oldpeak	99	0.3267327	0	0.0000000	0	0	numeric	40
slope	slope	0	0.0000000	0	0.0000000	0	0	integer	3
num_vessels_flour	num_vessels_flour	176	0.5808581	4	0.01320132	0	0	integer	4
thal	thal	0	0.0000000	2	0.00660066	0	0	factor	3
heart_disease_severity	heart_disease_severity	164	0.5412541	0	0.0000000	0	0	integer	5
exer_angina	exer_angina	204	0.6732673	0	0.0000000	0	0	factor	2
has_heart_disease	has_heart_disease	0	0.0000000	0	0.0000000	0	0	factor	2

```
> |
```

status(heart_disease) output

describe(heart_disease) ise her bir değişken için daha detaylı bir resim sunuyor diyebiliriz;

```
> describe(heart_disease)
```

heart_disease

16 Variables 303 Observations

```

age
  n missing distinct    Info    Mean    Gmd      .05      .10      .25      .50      .75      .90      .95
 303      0        41    0.999    54.44   10.3     .40     .42     .48     .56     .61     .66     .68

lowest : 29 34 35 37 38, highest: 70 71 74 76 77

gender
  n missing distinct
 303      0        2

Value      female    male
Frequency      97     206
Proportion    0.32    0.68

```

Age ve gender için detaylar

Kayıp değer hangi değişkende var diye hızlıca bakma adına;

```

di=data_integrity(heart_disease)
# returns a summary
summary(di)

```

ile sonuç alınabilir;

```

○ {Numerical with NA} num_vessels_flour
○ {Categorical with NA} thal

```

Birçok açıklayıcı istatistik değerine ulaşmak için;

```

profiling_num(heart_disease)

```

```

> profiling_num(heart_disease)
  variable      mean    std_dev variation_coef  p_01  p_05  p_25  p_50  p_75  p_95  p_99  skewness kurtosis  iqr
1      age  54.4389439  9.0386624    0.1660330  35.00  40.0  48.0  56.0  61.0  68.0  71.00 -0.2080241  2.465477  13.0
2 resting_blood_pressure 131.6897690 17.5997477    0.1336455 100.00 108.0 120.0 130.0 140.0 160.0 180.00  0.7025346  3.845881  20.0
3   serum_cholesterol 246.6930693  51.7769175    0.2098840 149.00 175.1 211.0 241.0 275.0 326.9 406.74  1.1298741  7.398208  64.0
4    max_heart_rate 149.6072607  22.8750033    0.1529004  95.02 108.1 133.5 153.0 166.0 181.9 191.96 -0.5347844  2.927602  32.5
5      exer_angina  0.3267327  0.4697945    1.4378558  0.00  0.0  0.0  0.0  1.0  1.0  1.00  0.7388506  1.545900  1.0
6      oldpeak  1.0396040  1.1610750    1.1168436  0.00  0.0  0.0  0.8  1.6  3.4  4.20  1.2634255  4.530193  1.6
7      slope  1.6006601  0.6162261    0.3849825  1.00  1.0  1.0  2.0  2.0  3.0  3.00  0.5057957  2.363050  1.0
8 num_vessels_flour  0.6722408  0.9374383    1.3944978  0.00  0.0  0.0  0.0  1.0  3.0  3.00  1.1833771  3.234941  1.0
9 heart_disease_severity 0.9372937  1.2285357    1.3107265  0.00  0.0  0.0  0.0  2.0  3.0  4.00  1.0532483  2.843788  2.0
  range_98      range_80
1   [35, 71]    [42, 66]
2   [100, 180]  [110, 152]
3   [149, 406.74] [188.8, 308.8]
4   [95.02, 191.96] [116, 176.6]
5   [0, 1]      [0, 1]
6   [0, 4.2]    [0, 2.8]
7   [1, 3]      [1, 2]
8   [0, 3]      [0, 2]
9   [0, 4]      [0, 3]

```

Descriptive Statistics

Sayısal çıktılar dışında ggplot2 tabanlı birçok görseli de hızlıca elde etmek mümkün. Bu temel fonksiyonlar dışında daha birçok özelliği olan bu paket için şu kaynak iyi bir başlangıç aslında;

funModeling quick-start {#quick_start}

This package contains a set of functions related to exploratory data analysis, data preparation, and model performance...

cran.r-project.org

2. dlookr

Oldukça güncel paketlerden birisi ve son zamanda yeni versiyonu oluşturulmuş.

Temel bazı fonksiyonlara örnek olarak,

- `diagnose()`: provides basic diagnostic information for variables.
- `describe()`: provides descriptive statistics for variables
- `diagnose_category()`: provides detailed diagnostic information for categorical variables.
- `diagnose_numeric()`: provides detailed diagnostic information for numerical variables.
- `diagnose_outlier()` / `plot_outlier()` provide information and visualization of outliers.

Aynı veri ile yola devam etmeye çalışalım;

```
diagnose(heart_disease)
```

ile **missing** ve **unique** bilgilerine erişebiliyoruz. Bu arada dikkatli gözlerin farkettiği üzere, çıktısı bir **tibble df** formunda;

```
> diagnose(heart_disease)
# A tibble: 16 × 6
  variables      types missing_count missing_percent unique_count unique_rate
  <chr>         <chr>         <int>         <dbl>         <int>         <dbl>
1 age          integer         0             0             41          0.135
2 gender        factor         0             0              2         0.00660
3 chest_pain    factor         0             0              4         0.0132
4 resting_blood_pressure integer         0             0             50          0.165
```

5	serum_cholesterol	integer	0	0	152	0.502
6	fasting_blood_sugar	factor	0	0	2	0.00660
7	resting_electro	factor	0	0	3	0.00990
8	max_heart_rate	integer	0	0	91	0.300
9	exer_angina	integer	0	0	2	0.00660
10	oldpeak	numeric	0	0	40	0.132
11	slope	integer	0	0	3	0.00990
12	num_vessels_flour	integer	4	1.32	5	0.0165
13	thal	factor	2	0.660	4	0.0132
14	heart_disease_severity	integer	0	0	5	0.0165
15	exer_angina	factor	0	0	2	0.00660
16	has_heart_disease	factor	0	0	2	0.00660

Buradan da görüldüğü üzere, **num_vessels_flour** ve **thal** için düşük de olsa bir kayı gözlem sözkonusu. Tabi 303 gözlem arasında düşük bir oran olduğunu da söylemekte fayda var. Tek bir değişken ya da seçili değişkenler için de bu fonksiyonu kullanmak mümkün,

```
# Select columns by name
diagnose(heart_disease, num_vessels_flour)
```

Değişkenleri numeric ya da categoric olarak farklı değerlendirmek ve daha fazla bilgi sahibi olmak için ise **diagnose_numeric()** / **diagnose_category()** kullanılabilir;

```
describe(heart_disease)
```

```
> describe(heart_disease)
# A tibble: 9 × 26
  variable      n  na    mean    sd se_mean  IQR skewness kurtosis  p00  p01  p05  p10  p20  p25  p30
  <chr>      <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 age        303    0  54.4   9.04  0.519   13  -0.209 -0.523   29  35   40   42   45   48   50
2 resting_blood... 303    0  132.  17.6  1.01   20   0.706  0.880   94 100  108  110  120  120  120
3 serum_cholest... 303    0  247.  51.8  2.97   64   1.14  4.49  126 149  175. 189. 204 211  218
4 max_heart_rate  303    0  150.  22.9  1.31  32.5  -0.537 -0.0535  71  95.0 108. 116  130 134. 141.
5 exer_angina    303    0   0.327 0.470 0.0270  1   0.743 -1.46    0  0    0    0    0    0    0
6 oldpeak       303    0   1.04  1.16 0.0667  1.6  1.27  1.58    0  0    0    0    0    0    0
7 slope        303    0   1.60  0.616 0.0354  1   0.508 -0.628    1  1    1    1    1    1    1
8 num_vessels_f... 299    4   0.672 0.937 0.0542  1   1.19  0.259    0  0    0    0    0    0    0
9 heart_disease... 303    0   0.937 1.23 0.0706  2   1.06 -0.139    0  0    0    0    0    0    0
# ... with 10 more variables: p40 <dbl>, p50 <dbl>, p60 <dbl>, p70 <dbl>, p75 <dbl>, p80 <dbl>, p90 <dbl>, p95 <dbl>,
#   p99 <dbl>, p100 <dbl>
```

Descriptive Statistics

```
> # For numeric variables
> diagnose_numeric(heart_disease)

> # For categoric variables
> diagnose_category(heart_disease)
```

```
> diagnose_numeric(heart_disease)
# A tibble: 9 × 10
  variables      min      Q1    mean median     Q3    max  zero minus outlier
  <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <int>
1 age          29    48  54.4    56    61    77      0      0      0
2 resting_blood_pressure  94   120  132.   130   140   200      0      0      9
3 serum_cholesterol    126   211  247.   241   275   564      0      0      5
4 max_heart_rate       71   134. 150.   153   166   202      0      0      1
5 exer_angina          0      0  0.327    0      1      1    204      0      0
6 oldpeak            0      0  1.04    0.8    1.6    6.2     99      0      5
7 slope              1      1  1.60     2      2      3      0      0      0
8 num_vessels_flour      0      0  0.672    0      1      3    176      0     20
9 heart_disease_severity  0      0  0.937    0      2      4    164      0      0
```

Numeric variables diagnose

```
> diagnose_category(heart_disease)
# A tibble: 19 × 6
  variables      levels      N    freq ratio rank
  <chr>      <chr> <int> <int> <dbl> <int>
1 gender      male    303   206 68.0     1
2 gender      female  303    97 32.0     2
3 chest_pain    4    303   144 47.5     1
4 chest_pain    3    303    86 28.4     2
5 chest_pain    2    303    50 16.5     3
6 chest_pain    1    303    23  7.59    4
7 fasting_blood_sugar 0    303   258 85.1     1
8 fasting_blood_sugar 1    303    45 14.9     2
9 resting_electro 0    303   151 49.8     1
10 resting_electro 2    303   148 48.8     2
11 resting_electro 1    303     4  1.32     3
12 thal         3    303   166 54.8     1
13 thal         7    303   117 38.6     2
14 thal         6    303    18  5.94     3
15 thal         NA    303     2  0.660    4
16 exer_angina  0    303   204 67.3     1
17 exer_angina  1    303    99 32.7     2
18 has_heart_disease no    303   164 54.1     1
19 has_heart_disease yes   303   139 45.9     2
```

Categoric variables

Ayrık değerlere ilk bakış için diagnose_outlier() / plot_outlier() fonksiyonları faydalı olabilir,

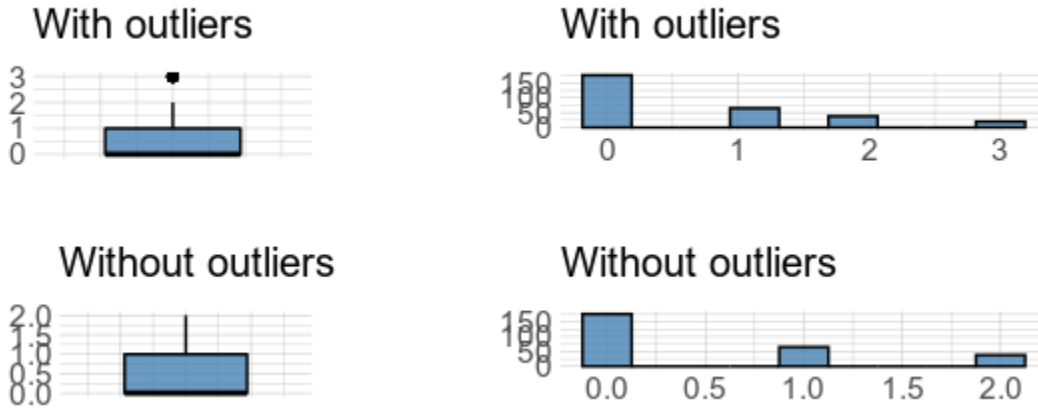
```
> diagnose_outlier(heart_disease)
  variables outliers_cnt outliers_ratio outliers_mean with_mean without_mean
1 age          0          0.000000      NaN    54.4389439    54.4389439
2 resting_blood_pressure  9          2.970297    181.5556 131.6897690    130.1632653
3 serum_cholesterol      5          1.650165    438.2000 246.6930693    243.4798658
4 max_heart_rate        1          0.330033    71.0000 149.6072607    149.8675497
5 exer_angina          0          0.000000      NaN     0.3267327     0.3267327
```

5	exer_angina	0	0.000000	NaN	0.5207527	0.5207527
6	oldpeak	5	1.650165	4.9200	1.0396040	0.9744966
7	slope	0	0.000000	NaN	1.6006601	1.6006601
8	num_vessels_flour	20	6.600660	3.0000	0.6722408	0.5053763
9	heart_disease_severity	0	0.000000	NaN	0.9372937	0.9372937

Outliers Detection

En fazla ayırık değer num_vessels_flour da olabilir gibi duruyor, tabi işin sonraki adımında ilgili testler ile bundan emin olmak lazım. Ayrıca olası bir scaling için mean değerlerdeki farklılıklar da (with_mean vs without_mean) önem arzedeabilir. Plot fonksiyonu her bir değişken için ayrı ayrı şu tarz görseller döküyor bizim için;

Outlier Diagnosis Plot (num_vessels_flour)



Outlier Diagnostic for num_vessels_flour

Burada boxplot grafikleri de bir ayıklama gerektiğine işaret eder yönde. Daha nice fonksiyon kullanımı için, bakınız;

Exploratory Data Analysis

Choonghyun Ryu After you have acquired the data, you should do the following: Diagnose data quality. If there is a...

cran.r-project.org

3. skimr

Yine yazarlarının deyimiyle;

skimr is designed to provide summary statistics about variables in data frames, tibbles, data tables and vectors. It is opinionated in its defaults, but easy to modify.

Tam anlamıyla verinin NE'si var diye göz atıyoruz aslında. Tek başına


```
> skim(heart_disease)
```

bile bize çok şey anlatıyor. Numeric değerler için histogram bile cabası (çok küçük görünüyorsa da). Aşağıda iki ayrı görselde verilen çıktılar, ortak bir şekilde çıktı karşımıza çıkıyor;

```
> skim(heart_disease)
```






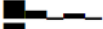



```
— Data Summary —
Name                Values
Number of rows      heart_disease
Number of columns    303
                    16

Column type frequency:
  factor              7
  numeric             9

Group variables      None
```

```
— Variable type: factor —
skim_variable  n_missing complete_rate ordered n_unique top_counts
1 gender        0           1      FALSE         2 mal: 206, fem: 97
2 chest_pain    0           1      FALSE         4 4: 144, 3: 86, 2: 50, 1: 23
3 fasting_blood_sugar 0           1      FALSE         2 0: 258, 1: 45
4 resting_electro 0           1      FALSE         3 0: 151, 2: 148, 1: 4
5 thal          2         0.993 FALSE         3 3: 166, 7: 117, 6: 18
6 exer_angina    0           1      FALSE         2 0: 204, 1: 99
7 has_heart_disease 0           1      FALSE         2 no: 164, yes: 139
```

skim() output, part 1

```
— Variable type: numeric —
skim_variable  n_missing complete_rate  mean  sd  p0  p25  p50  p75  p100 hist
1 age          0           1    54.4  9.04  29  48  56  61  77  
2 resting_blood_pressure 0           1    132. 17.6  94 120 130 140 200 
3 serum_cholesterol    0           1    247. 51.8 126 211 241 275 564 
4 max_heart_rate       0           1    150. 22.9  71 134. 153 166 202 
5 exer_angina          0           1    0.327 0.470  0  0  0  1  1  
6 oldpeak             0           1    1.04  1.16  0  0  0.8 1.6 6.2 
7 slope              0           1    1.60  0.616  1  1  2  2  3  
8 num_vessels_flour    4         0.987    0.672 0.937  0  0  0  1  3  
9 heart_disease_severity 0           1    0.937 1.23  0  0  0  2  4  
```

skim() output, part 2

Daha fazlası için;

GitHub - ropensci/skimr: A frictionless, pipeable approach to dealing with summary statistics

skimr provides a frictionless approach to summary statistics which conforms to the principle of least surprise...

github.com

ropensci/skimr

pipeable approach to dealing with summary statistics

19 Issues 836 Stars 72 Forks

4. DataExplorer

Temel olarak 3 farklı özelliğinden söz edilen bu paket yine oldukça güncel;

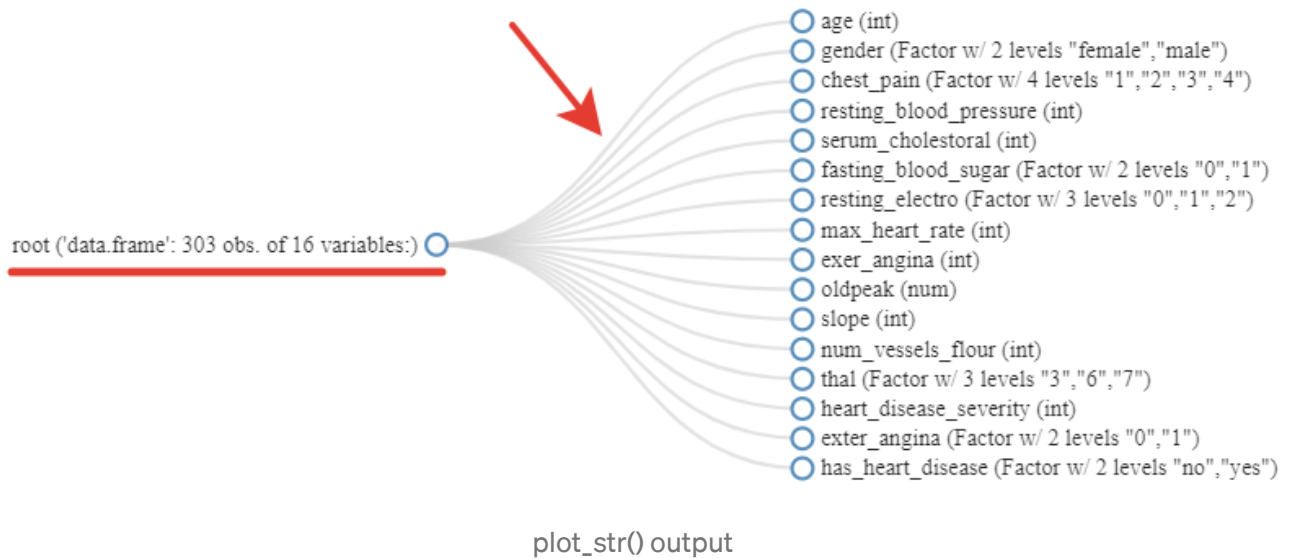
*There are 3 main goals for **DataExplorer**:*

- *Exploratory Data Analysis (EDA)*
- *Feature Engineering*
- *Data Reporting*

Temel bazı fonksiyonlar ile şunları görebilmek gerçekten ilginç ve keyifli;

```
> plot_str(heart_disease)
```

Veri setinin temel özellikleri ve değişkenlere dair ilk bilgileri buradan okuyabiliyoruz. Bir anlamda str()'nin görselleştirilmiş hali aslında (hangisi faktör, hangisi numeric vb.)



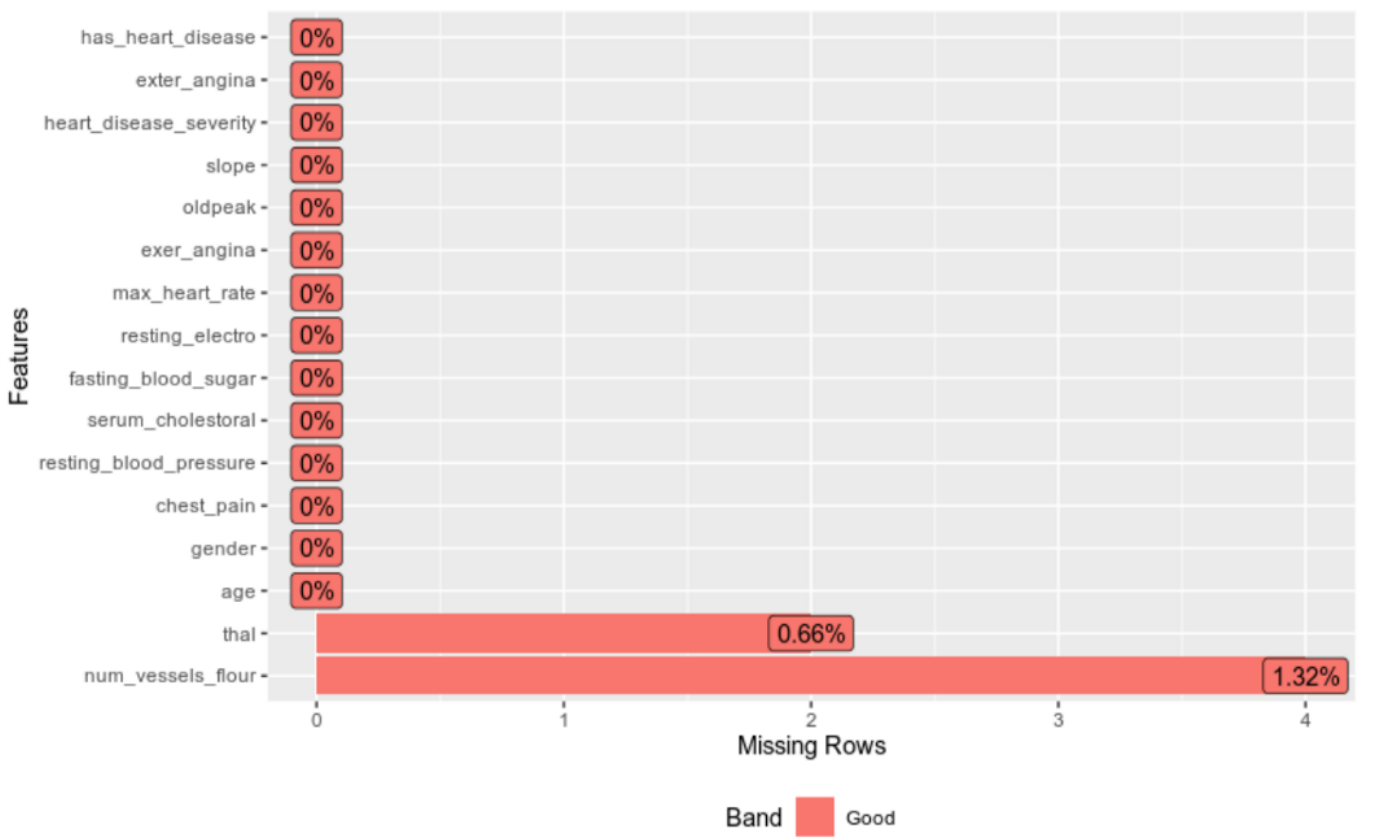
Şimdi size verimizi kısaca tanıtmak gerekirse;

```
introduce(heart_disease)
```

```
> introduce(heart_disease)
  rows columns discrete_columns continuous_columns all_missing_columns
1  303     16           7           9              0
total_missing_values complete_rows total_observations memory_usage
```

Genel bir özet

Bu çıktıları görselleştirmek yine mümkün. Hatta kayıp değerler için şu görsele şöyle bir göz atabiliriz;



Missing values for variables

Yine ggplot2 tabanlı, çok endişelenme diyen bir görüntü var burada. **Tabi kayıp değerler belli yüzdeleri aştıkça GOOD oluyor mu size önce OK, sonra da BAD!**

Yine buraya sığmayacak nice marifeti olan bu paket için bakınız;

Introduction to DataExplorer

Boxuan Cui This document introduces the package DataExplorer, and shows how it can help you with different tasks...

cran.r-project.org

5. summarytools

Son olarak kısaca bahsedeceğim paket yine EDA'dan fazlasına imkan tanıyor gibi görünüyor.

summarytools provides a coherent set of functions centered on data exploration and simple reporting.

Özellikle de tabular çıktılarının güzelliği birçok şeyi kolay yapmaya imkan tanıyor sanki. Kısaca birkaç fonksiyona bakalım;

```
freq(heart_disease)
```

bize şöyle bir çıktı döküyor (categoric değişkenler üzerinden);

```
> freq(heart_disease)
gender frequency percentage cumulative_perc
1 male      206      67.99      67.99
2 female    97      32.01     100.00

chest_pain frequency percentage cumulative_perc
1 4      144      47.52      47.52
2 3       86      28.38      75.90
3 2       50      16.50      92.40
4 1       23       7.59     100.00

fasting_blood_sugar frequency percentage cumulative_perc
1 0      258      85.15      85.15
2 1       45      14.85     100.00

resting_electro frequency percentage cumulative_perc
1 0      151      49.83      49.83
2 2      148      48.84      98.67
3 1       4       1.32     100.00

thal frequency percentage cumulative_perc
3 166      54.79      54.79
7 117      38.61      93.40
6 18       5.94      99.34
<NA> 2       0.66     100.00

exter_angina frequency percentage cumulative_perc
1 0      204      67.33      67.33
2 1       99      32.67     100.00

has_heart_disease frequency percentage cumulative_perc
1 no      164      54.13      54.13
2 yes     139      45.87     100.00

[1] "Variables processed: gender, chest_pain, fasting_blood_sugar, resting_electro, thal, exter_angina, has_heart_disease"
>
```

Ama daha da iyisi, `view(dfSummary())` ile şöyle bir rapor oluşturabiliyoruz.

(Not: Bu çıktı direk olarak viewer penceresinde açılmakla beraber, bir html sayfası olarak büyütüp görmek mümkün. Aşağıda oradan indirilmiş ve jpeg'e çevrilmiş halini görüyoruz)

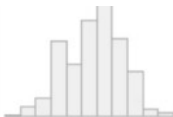
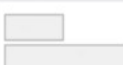
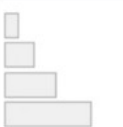
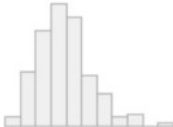
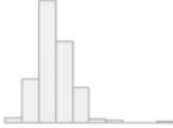
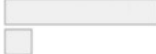

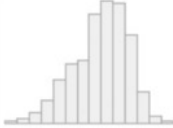

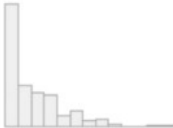


Data Frame Summary



heart_disease


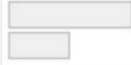
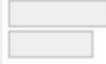
Dimensions: 303 x 16

Duplicates: 0

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	gender	male: 206 (67.99%) female: 97 (32.01%)	67.99% 32.01%		206 97	0

1	age [integer]	Mean (sd) : 54.4 (9) min ≤ med ≤ max: 29 ≤ 56 ≤ 77 IQR (CV) : 13 (0.2)	41 distinct values		303 (100.0%)	0 (0.0%)
2	gender [factor]	1. female 2. male	97 (32.0%) 206 (68.0%)		303 (100.0%)	0 (0.0%)
3	chest_pain [factor]	1. 1 2. 2 3. 3 4. 4	23 (7.6%) 50 (16.5%) 86 (28.4%) 144 (47.5%)		303 (100.0%)	0 (0.0%)
4	resting_blood_pressure [integer]	Mean (sd) : 131.7 (17.6) min ≤ med ≤ max: 94 ≤ 130 ≤ 200 IQR (CV) : 20 (0.1)	50 distinct values		303 (100.0%)	0 (0.0%)
5	serum_cholesterol [integer]	Mean (sd) : 246.7 (51.8) min ≤ med ≤ max: 126 ≤ 241 ≤ 564 IQR (CV) : 64 (0.2)	152 distinct values		303 (100.0%)	0 (0.0%)
6	fasting_blood_sugar [factor]	1. 0 2. 1	258 (85.1%) 45 (14.9%)		303 (100.0%)	0 (0.0%)
7	resting_electro [factor]	1. 0 2. 1 3. 2	151 (49.8%) 4 (1.3%) 148 (48.8%)		303 (100.0%)	0 (0.0%)
8	max_heart_rate [integer]	Mean (sd) : 149.6 (22.9) min ≤ med ≤ max: 71 ≤ 153 ≤ 202 IQR (CV) : 32.5 (0.2)	91 distinct values		303 (100.0%)	0 (0.0%)
9	exer_angina [integer]	Min : 0 Mean : 0.3 Max : 1	0 : 204 (67.3%) 1 : 99 (32.7%)		303 (100.0%)	0 (0.0%)
10	oldpeak [numeric]	Mean (sd) : 1 (1.2) min ≤ med ≤ max: 0 ≤ 0.8 ≤ 6.2 IQR (CV) : 1.6 (1.1)	40 distinct values		303 (100.0%)	0 (0.0%)
11	slope [integer]	Mean (sd) : 1.6 (0.6) min ≤ med ≤ max: 1 ≤ 2 ≤ 3 IQR (CV) : 1 (0.4)	1 : 142 (46.9%) 2 : 140 (46.2%) 3 : 21 (6.9%)		303 (100.0%)	0 (0.0%)
12	num_vessels_flour [integer]	Mean (sd) : 0.7 (0.9) min ≤ med ≤ max: 0 ≤ 0 ≤ 3 IQR (CV) : 1 (1.4)	0 : 176 (58.9%) 1 : 65 (21.7%) 2 : 38 (12.7%) 3 : 20 (6.7%)		299 (98.7%)	4 (1.3%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
13	thal [factor]	1. 3 2. 6 3. 7	166 (55.1%) 18 (6.0%) 117 (38.9%)		301 (99.3%)	2 (0.7%)
14	heart_disease_severity	Mean (sd) : 0.9 (1.2) min ≤ med ≤ max:	0 : 164 (54.1%) 1 : 55 (18.2%) 2 : 32 (10.3%) 3 : 20 (6.5%) 4 : 10 (3.2%) 5 : 5 (1.6%) 6 : 2 (0.6%) 7 : 1 (0.3%)		303	0

14	[integer]	0 ≤ 0 ≤ 4 IQR (CV) : 2 (1.3)	2 : 36 (11.9%) 3 : 35 (11.6%) 4 : 13 (4.3%)		(100.0%)	(0.0%)
15	exter_angina [factor]	1. 0 2. 1	204 (67.3%) 99 (32.7%)		303 (100.0%)	0 (0.0%)
16	has_heart_disease [factor]	1. no 2. yes	164 (54.1%) 139 (45.9%)		303 (100.0%)	0 (0.0%)

Generated by [summarytools](#) 1.0.0 (R version 3.6.3)
2021-09-05

Page 2

Numeric olanlar için histogram, categoric olanlar için barplot bu raporda otomatik ekleniyor. Buradaki histogram görüntüsü skim() ile elde edilenden biraz daha kullanışlı. Ayrıca yine gözlemlerin oranlarını, temel istatistik değerlerini ve kayıp veri oranlarını tek bir tabloda toplamış oluyoruz.

Tabi yine bunlar daha yolun başı, yine ne numaraları var kim bilir. Devamına bakmak isteyenler için;

Introduction to summarytools

summarytools provides a coherent set of functions centered on data exploration and simple reporting.

cran.r-project.org

Bu yolda nice araçlar daha var ve yenileri gelecek olsa gerek. “Bu verilerin NE’si var?” ile kendimce bir başlangıç yaptım, bakalım hangi sıklıkta ve hangi ölçekte devamını getirebileceğim, ben de merakla bekliyorum. Öneri ve görüşleriniz için yazarsanız sevinirim.

Son olarak bir de ingilizce yazı ekleyerek şimdilik kaçıyorum...

Discovering the Treasures of 22 R Exploratory Analysis Packages

Find out the fastest and most informative ways to do EDA in R

towardsdatascience.com



Data Science

Exploratory Data Analysis

R

Data Visualization



[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app

