

# Data Cleaning

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.



## Types of Unclean Data

There are 2 kinds of unclean data

### 1. Dirty Data (Data with Quality issues):


Dirty data, also known as low quality data. Low quality data has content issues.

- Duplicated data
- Missing Data
- Corrupt Data
- Inaccurate Data

### 2. Messy Data (Data with tidiness (neatness) issues):

Messy data, also known as untidy data. Untidy data has structural issues. Tidy data has the following properties:

- Each variable forms a column
- Each observation forms a row
- Each observational unit forms a table



country	year	rate
Afghanistan	1999	745 / 19987071
Afghanistan	2000	2666 / 20595360
Brazil	1999	37737 / 172006362
Brazil	2000	80488 / 174504898
China	1999	212258 / 1272915272
China	2000	213766 / 1280428583

table3

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

```
In [1]: import pandas as pd
import numpy as np
```

```
In [2]: patients = pd.read_csv("patients.csv")
treatments = pd.read_csv("treatments.csv")
adverse_reactions = pd.read_csv("adverse_reactions.csv")
treatments_cut = pd.read_csv("treatments_cut.csv")
```

```
In [3]: # view datasets
patients.head(2)
```

Out[3]:

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	co
0	1	female	Zoe	Wellish	576 Brown Bear Drive	Rancho California	California	92390.0	U
1	2	female	Pamela	Hill	2370 University Hill Road	Armstrong	Illinois	61812.0	U

```
In [4]: treatments.head(2)
```

Out[4]:

	given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
0	veronika	jindrová	41u - 48u	-	7.63	7.20	NaN
1	elliott	richardson	-	40u - 45u	7.56	7.09	0.97

```
In [5]: treatments_cut.head(2)
```

```
Out[5]:
```

	given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
0	jožka	resanovič	22u - 30u	-	7.56	7.22	0.34
1	inunnguaq	heilmann	57u - 67u	-	7.85	7.45	NaN

```
In [6]: adverse_reactions.head(2)
```

```
Out[6]:
```

	given_name	surname	adverse_reaction
0	berta	napolitani	injection site discomfort
1	lena	baer	hypoglycemia

## 1. Write a summary for your data

This is a dataset about 500 patients of which 350 patients participated in a clinical trial. None of the patients were using Novodra (a popular injectable insulin) or Auralin (the oral insulin being researched) as their primary source of insulin before. All were experiencing elevated HbA1c levels.

All 350 patients were treated with Novodra to establish a baseline HbA1c level and insulin dose. After 4 weeks, which isn't enough time to capture all the change in HbA1c that can be attributed by the switch to Auralin or Novodra:

- 175 patients switched to Auralin for 24 weeks
- 175 patients continued using Novodra for 24 weeks

Data about patients feeling some adverse effects is also recorded.

## 2. Write Column descriptions

**Table -> patients :**

- **patient\_id** : the unique identifier for each patient in the Master Patient Index (i.e. patient database) of the pharmaceutical company that is producing Auralin
- **assigned\_sex** : the assigned sex of each patient at birth (male or female)
- **given\_name** : the given name (i.e. first name) of each patient
- **surname** : the surname (i.e. last name) of each patient
- **address** : the main address for each patient
- **city** : the corresponding city for the main address of each patient
- **state** : the corresponding state for the main address of each patient
- **zip\_code** : the corresponding zip code for the main address of each patient
- **country** : the corresponding country for the main address of each patient (all United states for this clinical trial)
- **contact** : phone number and email information for each patient

- `birthdate` : the date of birth of each patient (month/day/year). The inclusion criteria for this clinical trial is age  $\geq 18$  (there is no maximum age because diabetes is a growing problem among the elderly population)
- `weight` : the weight of each patient in pounds (lbs)
- `height` : the height of each patient in inches (in)
- `bmi` : the Body Mass Index (BMI) of each patient. BMI is a simple calculation using a person's height and weight. The formula is  $BMI = \frac{kg}{m^2}$  where kg is a person's weight in kilograms and m2 is their height in metres squared. A BMI of 25.0 or more is overweight, while the healthy range is 18.5 to 24.9. The inclusion criteria for this clinical trial is  $BMI \geq 16$  and  $BMI \leq 38$ .

#### Table -> `adverse_reactions`

- `given_name` : the given name of each patient in the Master Patient Index that took part in the clinical trial and had an adverse reaction (includes both patients treated Auralin and Novodra)
- `surname` : the surname of each patient in the Master Patient Index that took part in the clinical trial and had an adverse reaction (includes both patients treated Auralin and Novodra)
- `adverse_reaction` : the adverse reaction reported by the patient

### 3. Add any additional information

Additional useful information:

- Insulin resistance varies person to person, which is why both starting median daily dose and ending median daily dose are required, i.e., to calculate change in dose.
- It is important to test drugs and medical products in the people they are meant to help. People of different age, race, sex, and ethnic group must be included in clinical trials. This diversity is reflected in the patients table.

### Types of Assessment

There are 2 types of assessment styles

- `Manual` - Looking through the data manually in google sheets
- `Programmatic` - By using pandas functions such as `info()`, `describe()` or `sample()`

### Steps in Assessment

There are 2 steps involved in Assessment

- Discover
- Documentation

```
In [7]: # export data for manual assessment

with pd.ExcelWriter('clinical_trials.xlsx') as writer:
    patients.to_excel(writer,sheet_name='patients')
    treatments.to_excel(writer,sheet_name='treatments')
    treatments_cut.to_excel(writer,sheet_name='treatment_cut')
    adverse_reactions.to_excel(writer,sheet_name='adverse_reactions')
```

## Issues with the dataset

### 1. Dirty Data ( Quality Related)

Table - Patients

- patient\_id = 9 has misspelled name 'Dsvid' instead of David accuracy

patient_id	gender	name	email	street	city	state
8	9 male	Dsvid	Gustafsson	1790 Nutter Street	Kansas City	MO
9	10 female	Sophie	Cabrera	3303 Anmoore Road	New York	New York

- state col sometimes contain full name and some times abbrivietation consistency

**state**

California
Illinois
Nebraska
NJ
AL
Florida
NV
CA
MO

- zip code col has entries with 4 digit validity

**zip\_code**

92390
61812
68467
7095
36303
32114
84728

- data missing for 12 patients in address,city, state,zip\_code ,country, contact completion

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	country	contact	birthdate
209	210	female	Lalita	Eldarkhan	Nan	Nan	Nan	Nan	Nan	Nan	8/14/1950
219	220	male	Mỹ	Quynh	Nan	Nan	Nan	Nan	Nan	Nan	4/9/1978
230	231	female	Elisabeth	Knudsen	Nan	Nan	Nan	Nan	Nan	Nan	9/23/1970
234	235	female	Martina	Tománkov	Nan	Nan	Nan	Nan	Nan	Nan	4/7/1936
242	243	male	John	O'Brian	Nan	Nan	Nan	Nan	Nan	Nan	2/25/1950
249	250	male	Benjamin	Mehler	Nan	Nan	Nan	Nan	Nan	Nan	10/30/1990
257	258	male	Jin	Kung	Nan	Nan	Nan	Nan	Nan	Nan	5/17/1990
264	265	female	Wafiyyah	Asfour	Nan	Nan	Nan	Nan	Nan	Nan	11/3/1980
269	270	female	Flavia	Fiorentino	Nan	Nan	Nan	Nan	Nan	Nan	10/9/1930
278	279	female	Generosa	Cabán	Nan	Nan	Nan	Nan	Nan	Nan	12/16/1900
286	287	male	Lewis	Webb	Nan	Nan	Nan	Nan	Nan	Nan	4/1/1979
296	297	female	Chi	Lâm	Nan	Nan	Nan	Nan	Nan	Nan	5/14/1990

- incorrect data type assigned to sex, zip code, birthdate validity

```

patient_id    503 non-null    int64
assigned_sex  503 non-null    object
given_name    503 non-null    object
surname       503 non-null    object
address       491 non-null    object
city          491 non-null    object
state         491 non-null    object
zip_code      491 non-null    float64
country       491 non-null    object
contact       491 non-null    object
birthdate     503 non-null    object
weight        503 non-null    float64
height        503 non-null    int64
bmi           503 non-null    float64

```

- duplicate entries by the name of John Doe accuracy

	patient_id	assigned_sex	given_name	surname	address	city	state
229	230	male	John	Doe	123 Main Street	New York	NY
237	238	male	John	Doe	123 Main Street	New York	NY
244	245	male	John	Doe	123 Main Street	New York	NY
251	252	male	John	Doe	123 Main Street	New York	NY
277	278	male	John	Doe	123 Main Street	New York	NY

- one patient has weight = 48 pounds accuracy

city	state	zip_code	country	contact	birthdate	weight	height	bmi
ster	OH	44691.0	United States	2145CamillaZaitseva@superrito.com	330-202-11/26/1938	48.8	63	19.1

- one patient has height = 27 inches accuracy

dress	city	state	zip_code	country	contact	birthdate	weight	height	bmi
1428 Turkey Pen Lane	Dothan	AL	36303.0	United States	7487TimNeudorf@cuvox.de	334-515-2/18/1928	192.3	27	26.1

Table - Treatments & Treatments\_cut

- given\_name and surname col is is all lower case consistency

given_name	surname
veronika	jindrová
elliott	richardson
yukitaka	takenaka

- remove u from Auralin and Novadra cols validity

auralin	novodra
41u - 48u	-
-	40u - 45u
-	39u - 36u
33u - 36u	-
-	33u - 29u

- '-' in novadra and Auralin col treated as nan validity

auralin	novodra
41u - 48u	-
-	40u - 45u
-	39u - 36u
33u - 36u	-

- missing values in hba1c\_change col completion

hba1c_change
0.97
0.35
0.32
0.38
0.38
0.34

- 1 duplicate entry by the name Joseph day accuracy

	given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
136	joseph	day	29u - 36u	-	7.7	7.19	NaN

- in hba1c\_change 9 instead of 4 accuracy

7.99	7.51	0.48	0.98
7.88	7.40		0.98
7.95	7.46		0.99
7.61	7.12		0.99
7.87	7.38		0.99

Table - Adverse\_reactions

- **given\_name and surname are all in lower case consistency**

given_name	surname
berta	napolitani
lena	baer
joseph	day

## 2. Messy Data ( Structural Related)

Table - Patients

- **contact col contains both phone and email**

contact
951-719-9170ZoeWellish@superrito.com
PamelaSHill@cuvox.de+1 (217) 569-3204
402-363-6804JaeMDebord@gustr.com
PhanBaLiem@jourrapide.com+1 (732) 636-8246
334-515-7487TimNeudorf@cuvox.de
386-334-5237RafaelCardosoCosta@gustr.com

Table - Treatments & Treatments\_cut

- **Auralin and Novadra col should be split into 2 cols start and end dose merge both the tables**

auralin	novodra
41u - 48u	-
-	40u - 45u

## Automatic Assessment

- head and tail
- sample
- info
- isnull
- duplicated
- describe



```
In [8]: # head
patients.head(1)
```

```
Out[8]:
```

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	cour
0	1	female	Zoe	Wellish	576 Brown Bear Drive	Rancho California	California	92390.0	Un St

```
In [9]: # tail
patients.tail(1)
```

```
Out[9]:
```

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	countr
502	503	male	Pat	Gersten	2778 North Avenue	Burr	Nebraska	68324.0	Unite State

```
In [10]: # sample
treatments.sample()
```

```
Out[10]:
```

	given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
263	rebecca	jephcott	53u - 63u	-	7.96	7.57	0.39

In [11]: *# info*

patients.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 503 entries, 0 to 502
Data columns (total 14 columns):
#   Column          Non-Null Count  Dtype
---  -
0   patient_id      503 non-null    int64
1   assigned_sex    503 non-null    object
2   given_name      503 non-null    object
3   surname         503 non-null    object
4   address         491 non-null    object
5   city            491 non-null    object
6   state           491 non-null    object
7   zip_code        491 non-null    float64
8   country         491 non-null    object
9   contact         491 non-null    object
10  birthdate       503 non-null    object
11  weight          503 non-null    float64
12  height          503 non-null    int64
13  bmi             503 non-null    float64
dtypes: float64(3), int64(2), object(9)
memory usage: 55.1+ KB
```

In [12]: *# Patients data 'Address' column as 12 Null values*

patients[patients['address'].isnull()]

Out[12]:

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	country
209	210	female	Lalita	Eldarkhanov	NaN	NaN	NaN	NaN	NaN
219	220	male	Mỹ	Quynh	NaN	NaN	NaN	NaN	NaN
230	231	female	Elisabeth	Knudsen	NaN	NaN	NaN	NaN	NaN
234	235	female	Martina	Tománková	NaN	NaN	NaN	NaN	NaN
242	243	male	John	O'Brian	NaN	NaN	NaN	NaN	NaN
249	250	male	Benjamin	Mehler	NaN	NaN	NaN	NaN	NaN
257	258	male	Jin	Kung	NaN	NaN	NaN	NaN	NaN
264	265	female	Wafiyyah	Asfour	NaN	NaN	NaN	NaN	NaN
269	270	female	Flavia	Fiorentino	NaN	NaN	NaN	NaN	NaN
278	279	female	Generosa	Cabán	NaN	NaN	NaN	NaN	NaN
286	287	male	Lewis	Webb	NaN	NaN	NaN	NaN	NaN
296	297	female	Chĩ	Lâm	NaN	NaN	NaN	NaN	NaN

In [13]: *# Table 2*treatments.info() *# missing values in hba1c*

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 280 entries, 0 to 279
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   given_name      280 non-null   object
1   surname         280 non-null   object
2   auralin         280 non-null   object
3   novodra         280 non-null   object
4   hba1c_start     280 non-null   float64
5   hba1c_end       280 non-null   float64
6   hba1c_change    171 non-null   float64
dtypes: float64(3), object(4)
memory usage: 15.4+ KB
```

In [14]: *# Table 3*treatments\_cut.info() *# missing values in hba1c*

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70 entries, 0 to 69
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   given_name      70 non-null   object
1   surname         70 non-null   object
2   auralin         70 non-null   object
3   novodra         70 non-null   object
4   hba1c_start     70 non-null   float64
5   hba1c_end       70 non-null   float64
6   hba1c_change    42 non-null   float64
dtypes: float64(3), object(4)
memory usage: 4.0+ KB
```

In [15]: *# table 4*

adverse\_reactions.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34 entries, 0 to 33
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  -
0   given_name      34 non-null   object
1   surname         34 non-null   object
2   adverse_reaction 34 non-null   object
dtypes: object(3)
memory usage: 944.0+ bytes
```

**deduplicated**

```
In [16]: # table 1
patients.duplicated().sum()
```

Out[16]: 0

```
In [17]: patients.duplicated(subset=['given_name', 'surname']).sum()
```

Out[17]: 5

```
In [18]: patients[patients.duplicated(subset=['given_name', 'surname'])]
```

Out[18]:

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	country	
<b>229</b>	230	male	John	Doe	123 Main Street	New York	NY	12345.0	United States	jr
<b>237</b>	238	male	John	Doe	123 Main Street	New York	NY	12345.0	United States	jr
<b>244</b>	245	male	John	Doe	123 Main Street	New York	NY	12345.0	United States	jr
<b>251</b>	252	male	John	Doe	123 Main Street	New York	NY	12345.0	United States	jr
<b>277</b>	278	male	John	Doe	123 Main Street	New York	NY	12345.0	United States	jr

```
In [19]: # table 2
treatments[treatments.duplicated()]
```

Out[19]:

	given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
<b>136</b>	joseph	day	29u - 36u	-	7.7	7.19	NaN

```
In [20]: treatments[treatments.duplicated(subset=['given_name', 'surname'])]
```

Out[20]:

	given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
<b>136</b>	joseph	day	29u - 36u	-	7.7	7.19	NaN

```
In [21]: # table 3

treatments_cut[treatments_cut.duplicated()]
```

```
Out[21]:
```

given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
------------	---------	---------	---------	-------------	-----------	--------------

```
In [22]: treatments_cut[treatments_cut.duplicated(subset=['given_name', 'surname'])]
```

```
Out[22]:
```

given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
------------	---------	---------	---------	-------------	-----------	--------------

```
In [23]: # Table 4

adverse_reactions.duplicated().sum()
```

```
Out[23]: 0
```

**describe**

```
In [24]: patients.describe()
```

```
Out[24]:
```

	patient_id	zip_code	weight	height	bmi
<b>count</b>	503.000000	491.000000	503.000000	503.000000	503.000000
<b>mean</b>	252.000000	49084.118126	173.434990	66.634195	27.483897
<b>std</b>	145.347859	30265.807442	33.916741	4.411297	5.276438
<b>min</b>	1.000000	1002.000000	48.800000	27.000000	17.100000
<b>25%</b>	126.500000	21920.500000	149.300000	63.000000	23.300000
<b>50%</b>	252.000000	48057.000000	175.300000	67.000000	27.200000
<b>75%</b>	377.500000	75679.000000	199.500000	70.000000	31.750000
<b>max</b>	503.000000	99701.000000	255.900000	79.000000	37.700000

```
In [25]: patients[patients['weight'] == 48.8]
```

```
Out[25]:
```

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	country
<b>210</b>	211	female	Camilla	Zaitseva	4689 Briarhill Lane	Wooster	OH	44691.0	United States

```
In [26]: patients[patients['height']== 27]
```

```
Out[26]:
```

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	country
4	5	male	Tim	Neudorf	1428 Turkey Pen Lane	Dothan	AL	36303.0	United States

```
In [27]: # table 2
treatments.describe()
```

```
Out[27]:
```

	hba1c_start	hba1c_end	hba1c_change
count	280.000000	280.000000	171.000000
mean	7.985929	7.589286	0.546023
std	0.568638	0.569672	0.279555
min	7.500000	7.010000	0.200000
25%	7.660000	7.270000	0.340000
50%	7.800000	7.420000	0.380000
75%	7.970000	7.570000	0.920000
max	9.950000	9.580000	0.990000

```
In [28]: treatments.sort_values('hba1c_start') # max
```

```
Out[28]:
```

	given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
270	mika	martinsson	34u - 43u	-	7.50	7.17	0.33
113	kari	laatikainen	39u - 43u	-	7.50	7.11	NaN
126	jowita	wisniewska	- 22u - 23u		7.50	7.08	0.92
53	nasser	mansour	- 33u - 31u		7.51	7.06	0.95
105	finlay	sheppard	- 31u - 30u		7.51	7.17	0.34
...	...	...	...	...	...	...	...
25	benoit	bonami	- 44u - 43u		9.82	9.40	0.92
171	justyna	kowalczyk	24u - 34u	-	9.84	9.44	NaN
81	robert	wagner	43u - 49u	-	9.84	9.52	0.32
75	mackenzie	mckay	- 44u - 43u		9.87	9.48	0.39
166	annie	allen	36u - 42u	-	9.95	9.58	0.37

280 rows × 7 columns

```
In [29]: treatments.sort_values('hba1c_change',na_position='first')
```

Out[29]:

	given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
0	veronika	jindrová	41u - 48u	-	7.63	7.20	NaN
2	yukitaka	takenaka	- 39u - 36u		7.68	7.25	NaN
8	saber	ménard	- 54u - 54u		8.08	7.70	NaN
9	asia	woźniak	30u - 36u	-	7.76	7.37	NaN
10	joseph	day	29u - 36u	-	7.70	7.19	NaN
...	...	...	...	...	...	...	...
49	jackson	addison	- 42u - 42u		7.99	7.51	0.98
17	gina	cain	- 36u - 36u		7.88	7.40	0.98
32	laura	ehrllichmann	- 43u - 40u		7.95	7.46	0.99
245	wu	sung	- 47u - 48u		7.61	7.12	0.99
138	giovana	rocha	- 23u - 21u		7.87	7.38	0.99

280 rows × 7 columns

```
In [30]: # table 3
```

```
treatments_cut.describe() # same prblem as treatments table
```

Out[30]:

	hba1c_start	hba1c_end	hba1c_change
count	70.000000	70.000000	42.000000
mean	7.838000	7.443143	0.518810
std	0.423007	0.418706	0.270719
min	7.510000	7.020000	0.280000
25%	7.640000	7.232500	0.340000
50%	7.730000	7.345000	0.370000
75%	7.860000	7.467500	0.907500
max	9.910000	9.460000	0.970000

## Note - Assessing Data is an Iterative Process

### Data Quality Dimensions

- Completeness -> is data missing?
- Validity -> is data invalid -> negative height -> duplicate patient id
- Accuracy -> data is valid but not accurate -> weight -> 1kg
- Consistency -> both valid and accurate but written differently -> New Youk and NY

- labelling is only for Dirty data

## Order of severity

Completeness <- Validity <- Accuracy <- Consistency

## Data Cleaning Order

1. Quality (Dirty data)-> Completeness ( Missing values)
2. Tidiness ( Messy data)
3. Quality -> Validity
4. Quality -> Accuracy
5. Quality -> Consistency

## Steps involved in Data cleaning

- Define
- Code
- Test

Always make sure to create a copy of your pandas dataframe before you start the cleaning process

In [31]: *# Making the copies*

```
patients_df = patients.copy()
treatments_df = treatments.copy()
treatments_cut_df = treatments_cut.copy()
adverse_reactions_df = adverse_reactions.copy()
```

In [32]: `patients.head(1)` *# original*

Out[32]:

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	cour
0	1	female	Zoe	Wellish	576 Brown Bear Drive	Rancho California	California	92390.0	Un St





```
In [33]: patients_df.head(1) # copy
```

```
Out[33]:
```

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	cour
0	1	female	Zoe	Wellish	576 Brown Bear Drive	Rancho California	California	92390.0	Un Sta

## Define

- replace all missing values of patients df with no data
- sub hba1c\_start from hba1c\_end to get all the change values
- in patients table we will use regex to separate email and phone

```
In [34]: patients_df[patients_df['address'].isnull()]
```

```
Out[34]:
```

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	country
209	210	female	Lalita	Eldarkhanov	NaN	NaN	NaN	NaN	NaN
219	220	male	Mỹ	Quynh	NaN	NaN	NaN	NaN	NaN
230	231	female	Elisabeth	Knudsen	NaN	NaN	NaN	NaN	NaN
234	235	female	Martina	Tománková	NaN	NaN	NaN	NaN	NaN
242	243	male	John	O'Brian	NaN	NaN	NaN	NaN	NaN
249	250	male	Benjamin	Mehler	NaN	NaN	NaN	NaN	NaN
257	258	male	Jin	Kung	NaN	NaN	NaN	NaN	NaN
264	265	female	Wafiyyah	Asfour	NaN	NaN	NaN	NaN	NaN
269	270	female	Flavia	Fiorentino	NaN	NaN	NaN	NaN	NaN
278	279	female	Generosa	Cabán	NaN	NaN	NaN	NaN	NaN
286	287	male	Lewis	Webb	NaN	NaN	NaN	NaN	NaN
296	297	female	Chĩ	Lâm	NaN	NaN	NaN	NaN	NaN

```
In [35]: # code
```

```
patients_df.fillna('No data',inplace=True)
```

In [36]: `# test``patients_df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 503 entries, 0 to 502
Data columns (total 14 columns):
#   Column          Non-Null Count  Dtype
---  -
0   patient_id      503 non-null    int64
1   assigned_sex    503 non-null    object
2   given_name      503 non-null    object
3   surname         503 non-null    object
4   address         503 non-null    object
5   city            503 non-null    object
6   state           503 non-null    object
7   zip_code        503 non-null    object
8   country         503 non-null    object
9   contact         503 non-null    object
10  birthdate       503 non-null    object
11  weight          503 non-null    float64
12  height          503 non-null    int64
13  bmi             503 non-null    float64
dtypes: float64(2), int64(2), object(10)
memory usage: 55.1+ KB
```

In [37]: `# table 2``treatments_df.head(2)`

Out[37]:

	given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
0	veronika	jindrová	41u - 48u	-	7.63	7.20	NaN
1	elliott	richardson	-	40u - 45u	7.56	7.09	0.97

In [38]:

`treatments_df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 280 entries, 0 to 279
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   given_name      280 non-null    object
1   surname         280 non-null    object
2   auralin         280 non-null    object
3   novodra         280 non-null    object
4   hba1c_start     280 non-null    float64
5   hba1c_end       280 non-null    float64
6   hba1c_change    171 non-null    float64
dtypes: float64(3), object(4)
memory usage: 15.4+ KB
```

In [39]: `# code`

```
treatments_df['hba1c_change'] = treatments_df['hba1c_start'] - treatments_df['h
```

In [40]: `#test`

```
treatments_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 280 entries, 0 to 279
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   given_name      280 non-null   object
1   surname         280 non-null   object
2   auralin         280 non-null   object
3   novodra         280 non-null   object
4   hba1c_start     280 non-null   float64
5   hba1c_end       280 non-null   float64
6   hba1c_change    280 non-null   float64
dtypes: float64(3), object(4)
memory usage: 15.4+ KB
```

In [41]: `# table 3`

```
treatments_cut_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70 entries, 0 to 69
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   given_name      70 non-null   object
1   surname         70 non-null   object
2   auralin         70 non-null   object
3   novodra         70 non-null   object
4   hba1c_start     70 non-null   float64
5   hba1c_end       70 non-null   float64
6   hba1c_change    42 non-null   float64
dtypes: float64(3), object(4)
memory usage: 4.0+ KB
```

In [42]: `treatments_cut_df['hba1c_change'] = treatments_cut_df['hba1c_start'] - treatmen`

In [43]: `treatments_cut_df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70 entries, 0 to 69
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   given_name      70 non-null    object
1   surname         70 non-null    object
2   auralin         70 non-null    object
3   novodra         70 non-null    object
4   hba1c_start     70 non-null    float64
5   hba1c_end       70 non-null    float64
6   hba1c_change    70 non-null    float64
dtypes: float64(3), object(4)
memory usage: 4.0+ KB
```

In [44]: `# tiddyness`

`patients_df.head() # contact`

Out[44]:

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	c
0	1	female	Zoe	Wellish	576 Brown Bear Drive	Rancho California	California	92390.0	
1	2	female	Pamela	Hill	2370 University Hill Road	Armstrong	Illinois	61812.0	
2	3	male	Jae	Debord	1493 Poling Farm Road	York	Nebraska	68467.0	
3	4	male	Liêm	Phan	2335 Webster Street	Woodbridge	NJ	7095.0	
4	5	male	Tim	Neudorf	1428 Turkey Pen Lane	Dothan	AL	36303.0	

In [45]: `import re`

In [46]: `# re - Regular expression`

```
def find_contact_details(text: str) -> tuple:
    # if the value is NaN, then return it
    if pd.isna(text):
        return np.nan

    # create the phone number pattern
    phone_number_pattern = re.compile(r"(\+[\d]{1,3}\s)?(\(?[\d]{3}\)?\s)?-?[\d]{3}-[\d]{4}"
    # find the phone number from the value/text, as a result we will get a list
    phone_number = re.findall(phone_number_pattern, text)

    # if length is 0, then the regex can't find any ph number, then define with
    if len(phone_number) <= 0:
        phone_number = np.nan
    # if the country code is attached with the ph number, for that case, the first
    # element will be the country code and the 2nd element will be the actual
    # number. So, get that ph number
    elif len(phone_number) >= 2:
        phone_number = phone_number[1]
    # else, we will get the ph number. Grab it.
    else:
        phone_number = phone_number[0]

    # if we found the ph number (with/without country code), then remove that
    # after removing the ph number, the remaining string might be the email address
    possible_email_add = re.sub(phone_number_pattern, "", text).strip()

    # then return the ph number and the email address
    return phone_number, possible_email_add
```

In [ ]:

In [47]: `patients_df['phone'] = patients_df["contact"].apply(lambda x: find_contact_details(x))`  
`patients_df['email'] = patients_df["contact"].apply(lambda x: find_contact_details(x))`

In [48]: `patients_df.head(2)`

Out[48]:

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	country
0	1	female	Zoe	Wellish	576 Brown Bear Drive	Rancho California	California	92390.0	USA
1	2	female	Pamela	Hill	2370 University Hill Road	Armstrong	Illinois	61812.0	USA

```
In [49]: patients_df.drop(columns='contact',inplace=True)
```

```
In [50]: patients_df.head(2)
```

Out[50]:

	patient_id	assigned_sex	given_name	surname	address	city	state	zip_code	co
0	1	female	Zoe	Wellish	576 Brown Bear Drive	Rancho California	California	92390.0	l s
1	2	female	Pamela	Hill	2370 University Hill Road	Armstrong	Illinois	61812.0	l s



```
In [51]: # merging two tables
```

```
treatments_df = pd.concat([treatments_df,treatments_cut_df])
```

```
In [52]: treatments_df
```

Out[52]:

	given_name	surname	auralin	novodra	hba1c_start	hba1c_end	hba1c_change
0	veronika	jindrová	41u - 48u	-	7.63	7.20	0.43
1	elliott	richardson	- 40u - 45u		7.56	7.09	0.47
2	yukitaka	takenaka	- 39u - 36u		7.68	7.25	0.43
3	skye	gormanston	33u - 36u	-	7.97	7.62	0.35
4	alissa	montez	- 33u - 29u		7.78	7.46	0.32
...	...	...	...	...	...	...	...
65	rovzan	kishiev	32u - 37u	-	7.75	7.41	0.34
66	jakob	jakobsen	- 28u - 26u		7.96	7.51	0.45
67	bernd	schneider	48u - 56u	-	7.74	7.44	0.30
68	berta	napolitani	- 42u - 44u		7.68	7.21	0.47
69	armina	sauvé	36u - 46u	-	7.86	7.40	0.46

350 rows × 7 columns

In [54]: *# Use melt function*

```
treatments_df = treatments_df.melt(id_vars=['given_name',
                                             'surname', 'hba1c_start',
                                             'hba1c_end', 'hba1c_change'],
                                   var_name='type',
                                   value_name='dosage_range')
```

In [55]: *treatments\_df # here we have variable and value columns*

Out[55]:

	given_name	surname	hba1c_start	hba1c_end	hba1c_change	type	dosage_range
0	veronika	jindrová	7.63	7.20	0.43	auralin	41u - 48u
1	elliot	richardson	7.56	7.09	0.47	auralin	-
2	yukitaka	takenaka	7.68	7.25	0.43	auralin	-
3	skye	gormanston	7.97	7.62	0.35	auralin	33u - 36u
4	alissa	montez	7.78	7.46	0.32	auralin	-
...	...	...	...	...	...	...	...
695	rovzan	kishiev	7.75	7.41	0.34	novodra	-
696	jakob	jakobsen	7.96	7.51	0.45	novodra	28u - 26u
697	bernd	schneider	7.74	7.44	0.30	novodra	-
698	berta	napolitani	7.68	7.21	0.47	novodra	42u - 44u
699	armina	sauvé	7.86	7.40	0.46	novodra	-

700 rows × 7 columns

In [56]: *# remove "-" from dose range , it givs 350 rows*

```
treatments_df = treatments_df[treatments_df['dosage_range'] != '-']
```

In [57]: `treatments_df`

Out[57]:

	given_name	surname	hba1c_start	hba1c_end	hba1c_change	type	dosage_range
0	veronika	jindrová	7.63	7.20	0.43	auralin	41u - 48u
3	skye	gormanston	7.97	7.62	0.35	auralin	33u - 36u
6	sophia	haugen	7.65	7.27	0.38	auralin	37u - 42u
7	eddie	archer	7.89	7.55	0.34	auralin	31u - 38u
9	asia	woźniak	7.76	7.37	0.39	auralin	30u - 36u
...	...	...	...	...	...	...	...
688	christopher	woodward	7.51	7.06	0.45	novodra	55u - 51u
690	maret	sultygov	7.67	7.30	0.37	novodra	26u - 23u
694	lixue	hsueh	9.21	8.80	0.41	novodra	22u - 23u
696	jakob	jakobsen	7.96	7.51	0.45	novodra	28u - 26u
698	berta	napolitani	7.68	7.21	0.47	novodra	42u - 44u

350 rows × 7 columns

In [58]: `treatments_df['dosage_range'].str.split('-') # split data into two columns`

Out[58]:

0	[41u , 48u]
3	[33u , 36u]
6	[37u , 42u]
7	[31u , 38u]
9	[30u , 36u]
...	...
688	[55u , 51u]
690	[26u , 23u]
694	[22u , 23u]
696	[28u , 26u]
698	[42u , 44u]

Name: dosage\_range, Length: 350, dtype: object

In [60]: `treatments_df['dosage_range'].str.split('-').str.get(0) # startig range`

Out[60]:

0	41u
3	33u
6	37u
7	31u
9	30u
...	...
688	55u
690	26u
694	22u
696	28u
698	42u

Name: dosage\_range, Length: 350, dtype: object



```
In [61]: treatments_df['dosage_range'].str.split('-').str.get(1) # ending range
```

```
Out[61]: 0      48u
          3      36u
          6      42u
          7      38u
          9      36u
          ...
        688     51u
        690     23u
        694     23u
        696     26u
        698     44u
        Name: dosage_range, Length: 350, dtype: object
```

```
In [62]: treatments_df['dosage_start'] = treatments_df['dosage_range'].str.split('-').s
treatments_df['dosage_end'] = treatments_df['dosage_range'].str.split('-').str
```

C:\Users\user\AppData\Local\Temp\ipykernel\_10564\2612136710.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

```
treatments_df['dosage_start'] = treatments_df['dosage_range'].str.split('-').str.get(0) # starting range
```

C:\Users\user\AppData\Local\Temp\ipykernel\_10564\2612136710.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

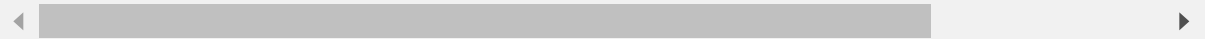
```
treatments_df['dosage_end'] = treatments_df['dosage_range'].str.split('-').str.get(1) # ending range
```

In [63]: `treatments_df`

Out[63]:

	given_name	surname	hba1c_start	hba1c_end	hba1c_change	type	dosage_range	d
0	veronika	jindrová	7.63	7.20	0.43	auralin	41u - 48u	
3	skye	gormanston	7.97	7.62	0.35	auralin	33u - 36u	
6	sophia	haugen	7.65	7.27	0.38	auralin	37u - 42u	
7	eddie	archer	7.89	7.55	0.34	auralin	31u - 38u	
9	asia	woźniak	7.76	7.37	0.39	auralin	30u - 36u	
...	...	...	...	...	...	...	...	...
688	christopher	woodward	7.51	7.06	0.45	novodra	55u - 51u	
690	maret	sultygov	7.67	7.30	0.37	novodra	26u - 23u	
694	lixue	hsueh	9.21	8.80	0.41	novodra	22u - 23u	
696	jakob	jakobsen	7.96	7.51	0.45	novodra	28u - 26u	
698	berta	napolitani	7.68	7.21	0.47	novodra	42u - 44u	

350 rows × 9 columns



In [65]: `treatments_df.drop(columns='dosage_range', inplace=True)`

C:\Users\user\anaconda3\lib\site-packages\pandas\core\frame.py:4906: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

return super().drop(

In [66]: `treatments_df`

Out[66]:

	given_name	surname	hba1c_start	hba1c_end	hba1c_change	type	dosage_start	dc
0	veronika	jindrová	7.63	7.20	0.43	auralin	41u	
3	skye	gormanston	7.97	7.62	0.35	auralin	33u	
6	sophia	haugen	7.65	7.27	0.38	auralin	37u	
7	eddie	archer	7.89	7.55	0.34	auralin	31u	
9	asia	woźniak	7.76	7.37	0.39	auralin	30u	
...	...	...	...	...	...	...	...	
688	christopher	woodward	7.51	7.06	0.45	novodra	55u	
690	maret	sultygov	7.67	7.30	0.37	novodra	26u	
694	lixue	hsueh	9.21	8.80	0.41	novodra	22u	
696	jakob	jakobsen	7.96	7.51	0.45	novodra	28u	
698	berta	napolitani	7.68	7.21	0.47	novodra	42u	

350 rows × 8 columns



In [67]: `# remove 'U' in dosage`

```
treatments_df['dosage_start'].str.replace('u', '')
```

Out[67]:

0	41
3	33
6	37
7	31
9	30
...	...
688	55
690	26
694	22
696	28
698	42

Name: dosage\_start, Length: 350, dtype: object

```
In [68]: treatments_df['dosage_start'] = treatments_df['dosage_start'].str.replace('u',
treatments_df['dosage_end'] = treatments_df['dosage_end'].str.replace('u', '')
```

C:\Users\user\AppData\Local\Temp\ipykernel\_10564\3184756727.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

```
treatments_df['dosage_start'] = treatments_df['dosage_start'].str.replace('u', '')
```

C:\Users\user\AppData\Local\Temp\ipykernel\_10564\3184756727.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

```
treatments_df['dosage_end'] = treatments_df['dosage_end'].str.replace('u', '')
```

```
In [69]: treatments_df # removed 'u from dosage'
```

Out[69]:

	given_name	surname	hba1c_start	hba1c_end	hba1c_change	type	dosage_start	dc
0	veronika	jindrová	7.63	7.20	0.43	auralin	41	
3	skye	gormanston	7.97	7.62	0.35	auralin	33	
6	sophia	haugen	7.65	7.27	0.38	auralin	37	
7	eddie	archer	7.89	7.55	0.34	auralin	31	
9	asia	woźniak	7.76	7.37	0.39	auralin	30	
...	...	...	...	...	...	...	...	...
688	christopher	woodward	7.51	7.06	0.45	novodra	55	
690	maret	sulygov	7.67	7.30	0.37	novodra	26	
694	lixue	hsueh	9.21	8.80	0.41	novodra	22	
696	jakob	jakobsen	7.96	7.51	0.45	novodra	28	
698	berta	napolitani	7.68	7.21	0.47	novodra	42	

350 rows × 8 columns



In [71]: `treatments_df.info()` *# here dosage is in "object"*

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 350 entries, 0 to 698
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  -
0   given_name      350 non-null    object
1   surname         350 non-null    object
2   hba1c_start     350 non-null    float64
3   hba1c_end       350 non-null    float64
4   hba1c_change    350 non-null    float64
5   type            350 non-null    object
6   dosage_start    350 non-null    object
7   dosage_end      350 non-null    object
dtypes: float64(3), object(5)
memory usage: 32.7+ KB
```

In [72]: *# changing into 'int'*

```
treatments_df['dosage_start'] = treatments_df['dosage_start'].astype('int')
treatments_df['dosage_end'] = treatments_df['dosage_end'].astype('int')
```

C:\Users\user\AppData\Local\Temp\ipykernel\_10564\3541383869.py:3: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

```
treatments_df['dosage_start'] = treatments_df['dosage_start'].astype('int')
```

C:\Users\user\AppData\Local\Temp\ipykernel\_10564\3541383869.py:4: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

```
treatments_df['dosage_end'] = treatments_df['dosage_end'].astype('int')
```

In [73]: `treatments_df.info()` *# changed object to int*

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 350 entries, 0 to 698
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  -
0   given_name      350 non-null    object
1   surname         350 non-null    object
2   hba1c_start     350 non-null    float64
3   hba1c_end       350 non-null    float64
4   hba1c_change    350 non-null    float64
5   type            350 non-null    object
6   dosage_start    350 non-null    int32
7   dosage_end      350 non-null    int32
dtypes: float64(3), int32(2), object(3)
memory usage: 30.0+ KB
```

In [74]: `treatments_df`

Out[74]:

	given_name	surname	hba1c_start	hba1c_end	hba1c_change	type	dosage_start	dc
0	veronika	jindrová	7.63	7.20	0.43	auralin	41	
3	skye	gormanston	7.97	7.62	0.35	auralin	33	
6	sophia	haugen	7.65	7.27	0.38	auralin	37	
7	eddie	archer	7.89	7.55	0.34	auralin	31	
9	asia	woźniak	7.76	7.37	0.39	auralin	30	
...	...	...	...	...	...	...	...	...
688	christopher	woodward	7.51	7.06	0.45	novodra	55	
690	marek	sultygov	7.67	7.30	0.37	novodra	26	
694	lixue	hsueh	9.21	8.80	0.41	novodra	22	
696	jakob	jakobsen	7.96	7.51	0.45	novodra	28	
698	berta	napolitani	7.68	7.21	0.47	novodra	42	

350 rows × 8 columns



In [75]: `adverse_reactions_df`

Out[75]:

	given_name	surname	adverse_reaction
0	berta	napolitani	injection site discomfort
1	lena	baer	hypoglycemia
2	joseph	day	hypoglycemia
3	flavia	fiorentino	cough
4	manouck	wubbels	throat irritation
5	jasmine	sykes	hypoglycemia
6	louise	johnson	hypoglycemia
7	albinca	komavec	hypoglycemia
8	noe	aranda	hypoglycemia
9	sofia	hermansen	injection site discomfort
10	tegan	johnson	headache
11	abel	yonatan	cough
12	abdul-nur	isa	hypoglycemia
13	leon	scholz	injection site discomfort
14	gabriele	saenger	hypoglycemia
15	jia li	teng	nausea
16	jakob	jakobsen	hypoglycemia
17	christopher	woodward	nausea
18	ole	petersen	hypoglycemia
19	finley	chandler	headache
20	anenechi	chidi	hypoglycemia
21	miłosław	wiśniewski	injection site discomfort
22	lixue	hsueh	injection site discomfort
23	merci	leroux	hypoglycemia
24	kang	mai	injection site discomfort
25	elliot	richardson	hypoglycemia
26	clinton	miller	throat irritation
27	idalia	moore	hypoglycemia
28	xiuxiu	chang	hypoglycemia
29	alex	crawford	hypoglycemia
30	monika	lončar	hypoglycemia
31	steven	roy	headache
32	cecilie	nilsen	hypoglycemia
33	krisztina	magyar	hypoglycemia



```
In [76]: # add extra column , which is adverse reaction table to treatments_df

treatments_df = treatments_df.merge(adverse_reactions_df, how ='left' , on =['
```

```
In [77]: treatments_df
```

```
Out[77]:
```

	given_name	surname	hba1c_start	hba1c_end	hba1c_change	type	dosage_start	dc
0	veronika	jindrová	7.63	7.20	0.43	auralin	41	
1	skye	gormanston	7.97	7.62	0.35	auralin	33	
2	sophia	haugen	7.65	7.27	0.38	auralin	37	
3	eddie	archer	7.89	7.55	0.34	auralin	31	
4	asia	woźniak	7.76	7.37	0.39	auralin	30	
...	...	...	...	...	...	...	...	...
345	christopher	woodward	7.51	7.06	0.45	novodra	55	
346	marek	sultygov	7.67	7.30	0.37	novodra	26	
347	lixue	hsueh	9.21	8.80	0.41	novodra	22	
348	jakob	jakobsen	7.96	7.51	0.45	novodra	28	
349	berta	napolitani	7.68	7.21	0.47	novodra	42	

350 rows × 9 columns



```
In [ ]:
```

```
In [ ]:
```