sqlDataModeling

January 21, 2022

1 SQL Data Modeling

1.1 Instructions

1. You have a single large table in a relational database with hundreds of columns that describe medical procedures performed, along with their dates and amounts.
   Ellipses (…), represent multiple rows of a similar type

1.1.1 TABLE: patient_procedure_info

patient_id	patient_attribute_1	…	patient_attribute_20
000000a	000000000	000000000	000000000
…	…	…	…
000000z	000000000	000000000	000000000

1.1.2 TABLE: diagnosis_info

patient_id	patient_diagnosis_code_1	…	patient_diagnosis_code_100
000000a	a00.000	a00.000	a00.000
…	…	…	…
000000z	a00.000	a00.000	a00.000

1.1.3 TABLE: visit_log

visit_id	patient_id	physician_id	physician_attribute_1	physician_attribute_20	visit_date	procedure_code	utilization_quantity	expenditure_amount
000000000000a	000000a	000000000	000000000	000000000	yyyy-mm-dd	000000000	00	$0000000000.00
…	…	…	…	…	…	…	…	…
000000000000z	000000z	000000000	000000000	000000000	yyyy-mm-dd	000000000	00	$0000000000.00

1.1.4 EXPLANATION of columns:

A **patient** has a series of attributes that have been updated as of the last doctor **visit**.
A **physician** has a series of attributes that are static across all visits and patients.

A patient has a list of associated health **diagnosis** codes updated as of the latest visit.
Example diagnosis_codes:

E11.9 – generic diabetes
C61 – prostate cancer
L89.103 – pressure ulcer

A **visit** is when a specific **patient** sees a specific **physician** who provides a specific **procedure** on a specific **date**

1.2 Questions

1.2.1 a. Write SQL to find the most common diagnosis_code. Explain your reasoning.

Functions to Synthsize Tables

- Note to self, possibily rewrite using numpy.

[8]:
```
def rand_benford_letter(list_length):
    """
    Accpepts int as the length of list to be returned.
    Function that returns a list of random letter(s) [a,z] in a randomized␣
 ↪log-normal distribution,
    in accordance with Benford's law:
    https://en.wikipedia.org/wiki/Benford%27s_law

    Parameters
    ----------
    list_length : int
        Number to specify length of randomized letter(s) to return.

    Returns
    -------
    list
        List of randomized log-normal distributed letter(s).

    See Also
    --------
    random.choices() : https://docs.python.org/3/library/random.html

    Examples
    --------
    >>> rand_benford_letter(3)
    ['d', 'b', 'k']
    >>> rand_benford_letter(1)
```

```
    ['k']
    >>> rand_benford_letter(10)
    ['r', 'n', 'p', 'q', 'y', 'z', 'z', 't', 'v', 'k']
    """

    import random
    import string
    import math

    # The seed() method is used to initialize the random number generator.
    # Default value is None, and if None, the generator uses the current system␣
 ↪time.
    random.seed()
    alpha_list = []
    benford = []

    for x in string.ascii_lowercase:
        alpha_list.append(x)
    # Comment out for non-log-normal distribution
    for i in range(26):
        benford.append(100 * math.log10(1+1/(i+1)))
    random.shuffle(benford)
    # return random.choice(alpha_list) # Uncomment for non-log-normal␣
 ↪distribution
    return random.choices(alpha_list, weights = benford, k = list_length)

def rand_number(number_length, decimal_length):
    """
    Accpepts int as the number of digits to be returned on both sides of the␣
 ↪decimal point.
    Function that returns a randomized float or an int of specified length.

    Parameters
    ----------
    number_length : int
        Number of digits to be returned to the left of the decimal place.

    decimal_length : int
        Number of digits to be returned to the right of the decimal place.

    Returns
    -------
    float, int, str
        Randomized number that corresponds to the decimal place input as␣
 ↪number_length,
        on both sides of the decimal spot. If all digits to the right of the␣
 ↪decimal are
```

```
        0, then returns an int. If digits to the left of the decimal are '0',␣
↪then returns
        a str, otherwise returns a float.

    See Also
    --------
    import random : [https://docs.python.org/3/library/random.html](https://
↪docs.python.org/3/library/random.html)

    Examples
    --------
    >>> rand_number(1, 3)
    3.011
    >>> rand_number(2, 3)
    '08.778'
    # Returns string to keep zero in tens spot.
    >>> rand_number(5, 3)
    29161
    # Radom number was 29161.000
    """

    import random

    digits = ''

    for i in range(number_length):
        digits += '9'
    digits = int(digits)
    # The seed() method is used to initialize the random number generator.
    # Default value is None, and if None, the generator uses the current system␣
↪time.
    random.seed()
    rand_num = round(random.triangular(0, digits), decimal_length)
    if rand_num.is_integer():
        rand_num = int(rand_num)
    if len(str(int(rand_num))) < len(str(int(digits))):
        rand_num = '0' + str(rand_num)
        return rand_num
    else:
        return rand_num

def random_letter():
    import random
    import string

    # The seed() method is used to initialize the random number generator.
```

```
    # Default value is None, and if None, the generator uses the current system time.
    random.seed()
    alpha_list = []

    for x in string.ascii_lowercase:
        alpha_list.append(x)
    return random.choice(alpha_list) # Uncomment for non-log-normal distribution

def rand_int(*number_length):
    import random

    max_int = ''
    for i in range(number_length[0]):
        max_int += '9'
    max_int = int(max_int)
    # The seed() method is used to initialize the random number generator.
    # Default value is None, and if None, the generator uses the current system time.
    random.seed()
    # rand_num = random.randint(0, max_int) # Uncomment for no weight in int range.

    # Comment out for no weight in int range.
    # if number_length[1].is_integer():
    try:
        rand_num = int(random.triangular(0, max_int, number_length[1]))
    # else:
    except:
        mode_num_weight = '1'
        for x in range(number_length[0] - 1):
            mode_num_weight += '0'
        mode_num_weight = int(mode_num_weight)
        rand_num = int(random.triangular(0, max_int, mode_num_weight))
    if len(str(int(rand_num))) < len(str(int(max_int))):
        rand_num = '0' + str(rand_num)
        return rand_num
    else:
        return rand_num

def rand_date(input_year, input_month, input_day):
    import random
    import datetime

    start_date = datetime.date(input_year, input_month, input_day)
    # end_date = datetime.date(2021, 2, 1)
    end_date = datetime.date.today()
```

```
    time_between_dates = end_date - start_date
    days_between_dates = time_between_dates.days
    random_number_of_days = random.randrange(days_between_dates)
    random_date = start_date + datetime.timedelta(days=random_number_of_days)
    return random_date
```

[9]:
```
def synthesize_attribute_list(attribute_name, max_num_of_attrib):
    attribute_name_list = []
    for x in range(max_num_of_attrib):
        # print(x + 1)
        attribute_name_list.append(attribute_name + str(x + 1))
    return attribute_name_list
```

[10]:
```
def synthesize_patient_id_data_batch(num_of_ids):
    patient_id_list = []
    for i in range(0, num_of_ids):
        patient_id_elem = str(rand_int(6)) + random_letter()
        patient_id_list.append(patient_id_elem)
    return patient_id_list

def synthesize_patient_id_data(num_unique_patients, num_synthetic_rows):
    patient_id_data_batch =␣
 ↪synthesize_patient_id_data_batch(num_unique_patients)
    def rand_elem():
        import random

        random.seed()
        patient_id_data_batch_elem = random.randint(1,␣
 ↪len(patient_id_data_batch) - 1)
        return patient_id_data_batch_elem

    num_unique_patients_to_add = num_synthetic_rows - num_unique_patients
    for i in range(num_unique_patients_to_add):
        patient_id_data_batch.append(patient_id_data_batch[rand_elem()])
    return patient_id_data_batch
```

[11]:
```
def synthesize_patient_attribute_data(num_of_ids):
    patient_attribute_list = []
    for i in range(0, num_of_ids):
        patient_attribute_list.append(rand_int(9))
    return patient_attribute_list
```

[12]:
```
def synthesize_patient_diagnosis_data_batch(num_of_ids):
    patient_diagnosis_code_list = []
    rand_benford_letter_list = rand_benford_letter(num_of_ids)
    for i in rand_benford_letter_list:
```

```
        patient_diagnosis_code_list.append(i + str(rand_number(2,3)))
    return patient_diagnosis_code_list

def synthesize_patient_diagnosis_data(num_unique_diagnosis, num_synthetic_rows):
    synthesize_patient_diagnosis_batch =␣
 ↪synthesize_patient_diagnosis_data_batch(num_unique_diagnosis)
    def rand_elem():
        import random

        random.seed()
        patient_diagnosis_batch_elem = random.randint(1,␣
 ↪len(synthesize_patient_diagnosis_batch) - 1)
        return patient_diagnosis_batch_elem

    num_unique_diagnosis_to_add = num_synthetic_rows - num_unique_diagnosis
    for i in range(num_unique_diagnosis_to_add):
        synthesize_patient_diagnosis_batch.
 ↪append(synthesize_patient_diagnosis_batch[rand_elem()])
    return synthesize_patient_diagnosis_batch
```

[13]:
```
def synthesize_physician_id_data_batch(num_of_ids):
    physician_id_list = []
    master_list = []
    physician_attribute_code_list =␣
 ↪synthesize_attribute_list('physician_attribute_', 20)

    for i in range(0, num_of_ids):
        physician_id_elem = str(rand_int(6)) + random_letter()
        physician_id_list.append(physician_id_elem)
        create_df_list = []
        for x in physician_attribute_code_list:
            create_df_list.append(str(synthesize_patient_attribute_data(1)).
 ↪strip('[]'))
        master_list.append(create_df_list)

    return physician_id_list, master_list

def synthesize_physician_id_data(num_unique_physicians, num_synthetic_rows):
    physician_id_and_attribute_data =␣
 ↪synthesize_physician_id_data_batch(num_unique_physicians)
    physician_id_data_batch = physician_id_and_attribute_data[0]
    physician_attribute_data_batch = physician_id_and_attribute_data[1]

    # helper function
    def rand_physician_id_elem():
        import random
```

```
        random.seed()
        physician_id_data_batch_elem = random.randint(1,␣
↪len(physician_id_data_batch) - 1)
        return physician_id_data_batch_elem

    num_unique_physicians_to_add = num_synthetic_rows - num_unique_physicians
    for i in range(num_unique_physicians_to_add):
        rand_element = rand_physician_id_elem()
        physician_id_data_batch.append(physician_id_data_batch[rand_element])
        physician_attribute_data_batch.
↪append(physician_attribute_data_batch[rand_element])

    return physician_id_data_batch, physician_attribute_data_batch
```

[14]:
```
def synthesize_visit_id_data(num_of_ids):
    visit_id_list = []
    for i in range(0, num_of_ids):
        visit_id_list.append(rand_int(6,1))
    return visit_id_list
```

[15]:
```
def synthesize_visit_date_data(num_of_ids):
    visit_date_list = []
    for i in range(0, num_of_ids):
        visit_date_list.append(rand_date(2016, 1, 1))
    return visit_date_list
```

[16]:
```
def synthesize_utilization_quantity_data(num_of_ids):
    utilization_quantity_list = []
    for i in range(0, num_of_ids):
        utilization_quantity_list.append(rand_int(2))
    return utilization_quantity_list
```

[17]:
```
## Update to use rand_int() to return lower figures.
def expenditure_amount_helper_func():
    some_var = rand_number(10, 2)
    some_var = str(some_var)

    if len(some_var) != 13:
        zero_to_add = 13 - len(str(some_var))
        for i in range(zero_to_add):
            some_var += '0'

    expenditure_amount_practice = '$' + some_var
    return expenditure_amount_practice

def synthesize_expenditure_amount_data(num_of_ids):
    expenditure_amount_list = []
```

```
    for i in range(0, num_of_ids):
        expenditure_amount_list.append(expenditure_amount_helper_func())
    return expenditure_amount_list
```

[18]:
```
print(rand_benford_letter.__doc__)
```

```
    Accpepts int as the length of list to be returned.
    Function that returns a list of random letter(s) [a,z] in a randomized log-
normal distribution,
    in accordance with Benford's law:
    https://en.wikipedia.org/wiki/Benford%27s_law

    Parameters
    ----------
    list_length : int
        Number to specify length of randomized letter(s) to return.

    Returns
    -------
    list
        List of randomized log-normal distributed letter(s).

    See Also
    --------
    random.choices() : https://docs.python.org/3/library/random.html

    Examples
    --------
    >>> rand_benford_letter(3)
    ['d', 'b', 'k']
    >>> rand_benford_letter(1)
    ['k']
    >>> rand_benford_letter(10)
    ['r', 'n', 'p', 'q', 'y', 'z', 'z', 't', 'v', 'k']
```

[19]:
```
rand_benford_letter(10)
```

[19]:
```
['y', 'y', 't', 'p', 'h', 't', 'p', 'k', 'a', 'h']
```

[20]:
```
print(rand_number.__doc__)
```

```
    Accpepts int as the number of digits to be returned on both sides of the
decimal point.
    Function that returns a randomized float or an int of specified length.
```

```
    Parameters
    ----------
    number_length : int
        Number of digits to be returned to the left of the decimal place.

    decimal_length : int
        Number of digits to be returned to the right of the decimal place.

    Returns
    -------
    float, int, str
        Randomized number that corresponds to the decimal place input as
number_length,
        on both sides of the decimal spot. If all digits to the right of the
decimal are
        0, then returns an int. If digits to the left of the decimal are '0',
then returns
        a str, otherwise returns a float.

    See Also
    --------
    import random : [https://docs.python.org/3/library/random.html](https://docs
.python.org/3/library/random.html)

    Examples
    --------
    >>> rand_number(1, 3)
    3.011
    >>> rand_number(2, 3)
    '08.778'
    # Returns string to keep zero in tens spot.
    >>> rand_number(5, 3)
    29161
    # Radom number was 29161.000
```

[21]:
```
rand_number(2,3)
```

[21]:
```
84.382
```

[22]:
```
rand_benford_letter(10)
```

[22]:
```
['u', 'h', 'm', 'x', 'u', 'v', 'q', 'i', 's', 'u']
```

[23]:
```
synthesize_patient_diagnosis_data(2, 10)
```

[23]:
```
['153.939',
 'n73.263',
```

```
 'n73.263',
 'n73.263',
 'n73.263',
 'n73.263',
 'n73.263',
 'n73.263',
 'n73.263',
 'n73.263']
```

Synthesize Tables

[24]:
```
import pandas as pd

def create_patient_procedure_info(num_synthetic_rows, num_unique_patients,␣
 ↪num_patient_attributes):
   ␣
 ↪#############################################################################################
    create_df_dict = {"patient_id" :␣
 ↪synthesize_patient_id_data(num_unique_patients, num_synthetic_rows)}
   ␣
 ↪#############################################################################################
    patient_attribute_list = synthesize_attribute_list('patient_attribute_',␣
 ↪num_patient_attributes)

    for x in patient_attribute_list:
        create_df_dict.update({x :␣
 ↪synthesize_patient_attribute_data(num_synthetic_rows)})
    patient_procedure_info = pd.DataFrame(create_df_dict)
    return patient_procedure_info

num_synthetic_rows = 5000
num_unique_patients = 5000
num_patient_attributes = 20

patient_procedure_info = create_patient_procedure_info(num_synthetic_rows,␣
 ↪num_unique_patients, num_patient_attributes)
patient_procedure_info
```

[24]:

	patient_id	patient_attribute_1	patient_attribute_2	patient_attribute_3	\
0	247119k	581673309	794009499	096095049	
1	158101w	033370982	626671986	284880017	
2	666206q	344993208	347424251	258921760	
3	657712a	035195972	104893580	821531639	
4	421272l	053596217	369510900	162837002	
…	…	…	…	…	
4995	662020b	255756305	572618727	538882994	
4996	334719c	250765798	210462956	091949067	

```
4997    141934e           222751182           146962002           310967501
4998    3994411           226795341           570982399           268047518
4999    332817y           510741114           584186529           110781997

     patient_attribute_4 patient_attribute_5 patient_attribute_6  \
0              748069799           173533249           012526968
1              267147920           519306550           089742106
2              505632997           436719098           180381710
3              278337280           188720910           600497420
4              455198606           419365579           071407284
…                  …                   …                   …
4995           405830451           194387648           335392726
4996           167762131           296571783           063112789
4997           292727434           164512650           160488879
4998           905625227           491598549           734829286
4999           039054516           210333118           322666051

     patient_attribute_7 patient_attribute_8 patient_attribute_9  …  \
0              757270224           406822643           578797346  …
1              150224620           084983348           273298233  …
2              470530211           628836008           318924462  …
3              323694175           625111654           334376694  …
4              260680124           422476730           383943412  …
…                  …                   …                   …  …
4995           658491025           738542082           580961466  …
4996           093510367           320606853           057184901  …
4997           613177789           856080686           299210417  …
4998           419304222           590200519           798065563  …
4999           854480375           552781690           602245118  …

     patient_attribute_11 patient_attribute_12 patient_attribute_13  \
0               042232225            448006145            077968688
1               643631238            419495113            689160370
2               568825507            090180828            728906941
3               460716829            085071447            266517702
4               038744234            687781287            382773894
…                   …                   …                   …
4995            452210129            057722480            078935950
4996            197246466            896769504            634095558
4997            176932485            117349187            501082705
4998            489536673            309049296            328135306
4999            082934992            850397512            074965340

     patient_attribute_14 patient_attribute_15 patient_attribute_16  \
0               355507698            517637301            037381403
1               493783092            484239726            576215536
2               177277987            262171072            341490581
```

```
3               089978298            169056478            310407579
4               602649877            107855703            494312684
…                     …                    …                    …
4995            180391053            197922956            259358035
4996            197233264            136490245            212073867
4997            460679243            496334945            546872923
4998            343561143            145155975            219470528
4999            084449498            427572094            247130743

     patient_attribute_17 patient_attribute_18 patient_attribute_19  \
0               615114839            865958375            400166584
1               040112780            759127250            186668205
2               305331714            674103841            178508048
3               079520989            342414334            462842838
4               929191876            407740616            647469278
…                     …                    …                    …
4995            055211710            163313103            379222102
4996            134704849            487118642            158940087
4997            589008919            124609841            723481695
4998            547597595            250267226            279733572
4999            498208362            380421065            162742726

     patient_attribute_20
0               772737972
1               463246366
2               327472162
3               729004993
4               227767400
…                     …
4995            387645274
4996            187151205
4997            524488786
4998            539690520
4999            735782075

[5000 rows x 21 columns]
```

[2430]:
```
import pandas as pd

def create_diagnosis_info(num_synthetic_rows, num_unique_patients,␣
 ↪num_unique_diagnosis, num_patient_diagnosis_codes):
   ␣
 ↪#########################################################################################
    create_df_dict = {"patient_id" :␣
 ↪synthesize_patient_id_data(num_unique_patients, num_synthetic_rows)}
   ␣
 ↪#########################################################################################
```

```
    patient_diagnosis_code_list =␣
 ↪synthesize_attribute_list('patient_diagnosis_code_',␣
 ↪num_patient_diagnosis_codes)

    for x in patient_diagnosis_code_list:
        create_df_dict.update({x :␣
 ↪synthesize_patient_diagnosis_data(num_unique_diagnosis, num_synthetic_rows)})
    diagnosis_info = pd.DataFrame(create_df_dict)
    return diagnosis_info

num_synthetic_rows = 5000
num_unique_patients = 5000
num_unique_diagnosis = 300
num_patient_diagnosis_codes = 100

diagnosis_info = create_diagnosis_info(num_synthetic_rows, num_unique_patients,␣
 ↪num_unique_diagnosis, num_patient_diagnosis_codes)
diagnosis_info
```

[2430]:

	patient_id	patient_diagnosis_code_1	patient_diagnosis_code_2	\
0	502214w	d38.943	f63.869	
1	177493h	h41.585	s45.442	
2	236599b	r66.872	s22.362	
3	688451y	v64.174	o88.715	
4	016564i	l76.386	v56.508	
…	…	…	…	
4995	537306x	u63.722	c61.115	
4996	576932m	l33.786	j39.847	
4997	589525f	d85.61	b62.944	
4998	762117i	v64.174	y17.747	
4999	375814t	d48.313	f22.044	

	patient_diagnosis_code_3	patient_diagnosis_code_4	\
0	t30.04	q50.006	
1	s47.589	q56.461	
2	l38.168	g71.231	
3	c42.755	q50.829	
4	l55.711	t77.29	
…	…	…	
4995	h58.402	s23.66	
4996	v48.627	q50.106	
4997	o59.904	v46.583	
4998	t49.142	q50.898	
4999	d72.975	d87.767	

	patient_diagnosis_code_5	patient_diagnosis_code_6	\
0	g45.81	v68.362	

```
1                    y28.663                   v16.285
2                    g35.294                   h50.221
3                    g63.458                   e62.317
4                    i38.342                    l51.04
…                      …                       …
4995                 a59.931                   j34.372
4996                 i38.342                   w53.601
4997                 u41.839                   i46.307
4998                 y42.428                   x27.776
4999                 i38.342                   n74.625

     patient_diagnosis_code_7 patient_diagnosis_code_8  \
0                    x27.962                   j62.755
1                    l30.568                   v54.901
2                    x08.948                   x73.304
3                    f84.687                   a81.802
4                    x49.241                   o64.604
…                      …                       …
4995                 z43.984                   e39.532
4996                 l66.838                   y20.845
4997                 o42.892                   w74.641
4998                 j30.314                    g78.27
4999                 f61.866                   a59.571

     patient_diagnosis_code_9  … patient_diagnosis_code_91  \
0                    u45.492  …                    q55.98
1                    a88.021  …                   p49.244
2                    v31.845  …                   i56.781
3                    u67.078  …                   h73.789
4                     b69.89  …                   t61.234
…                      …  …                       …
4995                  z30.85  …                   l24.159
4996                 s18.298  …                   c37.896
4997                  u77.27  …                   y51.212
4998                 r57.091  …                   j26.148
4999                 s18.298  …                   w82.405

     patient_diagnosis_code_92 patient_diagnosis_code_93  \
0                     u71.596                   n76.585
1                     a63.734                   x23.779
2                     t24.303                     b17.5
3                     u87.263                   v45.972
4                     h47.801                   b28.249
…                       …                       …
4995                  a71.571                   g32.826
4996                  q19.946                   w71.151
4997                  i54.415                   q40.182
```

```
4998                   r78.961                   v45.972
4999                   h47.801                   m44.379

     patient_diagnosis_code_94 patient_diagnosis_code_95  \
0                      b44.093                   f76.279
1                      u73.381                   k41.259
2                      j42.237                   s65.474
3                      l48.868                   w41.314
4                      j08.261                   d92.389
…                        …                       …
4995                    y56.65                   w92.679
4996                   y15.453                   h45.077
4997                   z46.594                   f42.943
4998                   s88.074                   w92.679
4999                   d47.796                   s62.357

     patient_diagnosis_code_96 patient_diagnosis_code_97  \
0                      y53.903                   t53.138
1                      y46.492                   m33.728
2                      h25.724                    q75.79
3                      l40.174                   h90.453
4                      g59.003                   p64.041
…                        …                       …
4995                   j07.091                   u41.901
4996                   j07.091                   g63.835
4997                   y46.492                   g59.822
4998                   y52.807                    q75.79
4999                   y63.437                   p35.178

     patient_diagnosis_code_98 patient_diagnosis_code_99  \
0                      o46.152                   g36.922
1                      g39.731                   j48.709
2                      c69.448                   o54.064
3                      e42.364                   w45.952
4                      c45.689                    m27.45
…                        …                       …
4995                   s86.368                    k71.68
4996                   m72.713                   o48.997
4997                   b55.904                   k27.379
4998                   y37.386                   w55.794
4999                   c36.367                    r59.78

     patient_diagnosis_code_100
0                       r59.308
1                        y17.74
2                       a82.341
3                       o42.973
```

```
4                       a49.706
…                         …
4995                    n67.472
4996                    p35.996
4997                    r33.067
4998                    y60.259
4999                    g62.447

[5000 rows x 101 columns]
```

Solution to Question a. in Pandas

[2431]:
```
most_common_code = diagnosis_info.iloc[:,1:].apply(pd.Series.value_counts).
 ↪sum(1).idxmax()
most_common_cod_freq = diagnosis_info.iloc[:,1:].apply(pd.Series.value_counts).
 ↪loc[most_common_code].sum()

print("The most common diagnosis code is: \n\t{}: {}\n\tAssuming that the␣
 ↪patient_id is the primary key and does not repeat itself.".
 ↪format(most_common_code, most_common_cod_freq))
```

```
The most common diagnosis code is:
        z40.776: 168.0
        Assuming that the patient_id is the primary key and does not repeat
itself.
```

[25]:
```
import pandas as pd

def create_visit_log(num_synthetic_rows, num_unique_patients,␣
 ↪num_unique_physicians, num_physician_attributes):
    physician_data = synthesize_physician_id_data(num_unique_physicians,␣
 ↪num_synthetic_rows)

    create_df_dict = {"visit_id" : synthesize_visit_id_data(num_synthetic_rows)}
  ␣
 ↪#############################################################################################
    create_df_dict.update({"patient_id" :␣
 ↪synthesize_patient_id_data(num_unique_patients, num_synthetic_rows)})
  ␣
 ↪#############################################################################################
    create_df_dict.update({"physician_id" : physician_data[0]})

    ### <RESOLVED> physician_attribute_code_list needs to be the same value for␣
 ↪each unique physician_id </RESOLVED>
    physician_attribute_code_list =␣
 ↪synthesize_attribute_list('physician_attribute_', num_physician_attributes)

    for i in physician_attribute_code_list:
```

```
        create_df_dict.update({i : []})

    for x in range(num_synthetic_rows):
        col_list = []
        col_name = ''
        for y in range(len(physician_attribute_code_list)):
            col_name = physician_attribute_code_list[y]
            create_df_dict[col_name].append(physician_data[1][x][y])

    create_df_dict.update({"visit_date" :␣
 ↪synthesize_visit_date_data(num_synthetic_rows)})
    create_df_dict.update({"procedure_code" :␣
 ↪synthesize_patient_attribute_data(num_synthetic_rows)})
    create_df_dict.update({"utilization_quantity" :␣
 ↪synthesize_utilization_quantity_data(num_synthetic_rows)})
    create_df_dict.update({"expenditure_amount" :␣
 ↪synthesize_expenditure_amount_data(num_synthetic_rows)})

    visit_log = pd.DataFrame(create_df_dict)
    return visit_log

num_synthetic_rows = 50000
num_unique_patients = 5000
num_unique_physicians = 50
num_physician_attributes = 20

visit_log = create_visit_log(num_synthetic_rows, num_unique_patients,␣
 ↪num_unique_physicians, num_physician_attributes)
visit_log
```

[25]:

	visit_id	patient_id	physician_id	physician_attribute_1	\
0	104144	603533x	449552r	151639471	
1	022495	100743o	168084c	'052274667'	
2	079618	604765r	785651y	265712811	
3	410791	477022d	568801z	140050803	
4	483181	496649d	083725r	167711577	
…	…	…	…	…	
49995	229532	570469y	819744t	659119840	
49996	205388	408284j	819744t	659119840	
49997	239364	034003e	351274c	280294469	
49998	052016	120751z	649423l	611657985	
49999	03581	465187i	148635c	438917785	

	physician_attribute_2	physician_attribute_3	physician_attribute_4	\
0	362656211	219136646	371325203	
1	294780762	731281113	532794649	
2	171267920	250313361	132431879	

```
3                139567926             160950293             217020588
4                747990496             362622709             335974685
…                    …                     …                     …
49995            139031512           '066165474'             342832979
49996            139031512           '066165474'             342832979
49997          '085527275'             786052824             816258680
49998            844793326             128099380             429219068
49999            314552249             314299150             668454402

      physician_attribute_5 physician_attribute_6 physician_attribute_7  …  \
0                536385296             488532900             571652318  …
1              '095553196'             626722161             725871438  …
2                251859709             489332258             197199741  …
3                939978369           '045005942'             496645374  …
4                842952835             114197005           '089898280'  …
…                    …                     …                     …  …
49995            628623036           '045648425'             519087335  …
49996            628623036           '045648425'             519087335  …
49997            219671829             661090028             433810074  …
49998            305490366             607007100             478219501  …
49999            140925531             670829296             162837802  …

      physician_attribute_15 physician_attribute_16 physician_attribute_17  \
0                  532673919              843575291               127700892
1                  237275110              337936344               295173792
2                '095212194'            '028586650'               431820074
3                  204933700              618878232             '046916409'
4                  234691751              248462987               209361926
…                      …                      …                       …
49995              100876272              616764411               593244598
49996              100876272              616764411               593244598
49997              256749109              470771310               800816098
49998              514294045              108368187               143628948
49999              333592569            '019151278'               116600534

      physician_attribute_18 physician_attribute_19 physician_attribute_20  \
0                  156383177              824682516               828278104
1                  211146250              630224774               433740671
2                  346796669              328948327               524582530
3                  301442596              150907828               452356894
4                '065428882'            '076641644'               484375267
…                      …                      …                       …
49995              163231451              325450905               109032483
49996              163231451              325450905               109032483
49997            '041984427'              319888345               343085422
49998              215416746              740053922               586533342
49999              137044488            '086641748'               140910462
```

```
       visit_date procedure_code utilization_quantity expenditure_amount
0      2017-12-28      677531419                    78     $3357841023.70
1      2021-11-10      356979966                    52     $3310448780.70
2      2019-02-08      163780367                    26     $4438855575.60
3      2016-11-21      778198182                    11     $6259597539.12
4      2019-04-23      339046032                    41     $2206239039.41
…             …              …                     …                  …
49995  2020-02-10      523826857                    65     $5831417384.51
49996  2018-05-12      548663740                    67     $4603935944.04
49997  2020-07-08      320869744                    31     $2696317508.73
49998  2018-12-12      468706420                    11     $5971273216.88
49999  2017-10-24      753574712                    36     $6786703920.47

[50000 rows x 27 columns]
```

physician_attribute_n's are static and correspond with a unique physician_id

[2433]:
```
visit_log.groupby('physician_id').count()
```

[2433]:

physician_id	visit_id	patient_id	physician_attribute_1	\
036439c	353	353	353	
040702s	652	652	652	
062765n	46	46	46	
084502f	3253	3253	3253	
088034e	2554	2554	2554	
091208a	1018	1018	1018	
096731l	393	393	393	
119986g	76	76	76	
128011w	1	1	1	
144722h	702	702	702	
148376c	1080	1080	1080	
157088t	1212	1212	1212	
160368z	839	839	839	
166297x	267	267	267	
172381r	923	923	923	
178973m	2562	2562	2562	
188714q	1989	1989	1989	
191807d	1709	1709	1709	
196918b	48	48	48	
216410a	1140	1140	1140	
243277z	995	995	995	
249664v	34	34	34	
255661r	1141	1141	1141	
286682k	391	391	391	
294451d	824	824	824	
298032c	3156	3156	3156	

310699m	421	421	421
370259u	1131	1131	1131
371239k	711	711	711
372113w	1994	1994	1994
391937c	1324	1324	1324
431199d	67	67	67
485962l	487	487	487
502937q	102	102	102
549613p	1340	1340	1340
550555b	511	511	511
558504v	235	235	235
561606e	4663	4663	4663
565688t	477	477	477
574704k	150	150	150
609064x	2340	2340	2340
614271h	861	861	861
694645v	360	360	360
709574n	2760	2760	2760
717175o	589	589	589
752451k	257	257	257
761523d	683	683	683
789822b	208	208	208
797135h	833	833	833
855842w	138	138	138

physician_id	physician_attribute_2	physician_attribute_3	\
036439c	353	353	
040702s	652	652	
062765n	46	46	
084502f	3253	3253	
088034e	2554	2554	
091208a	1018	1018	
096731l	393	393	
119986g	76	76	
128011w	1	1	
144722h	702	702	
148376c	1080	1080	
157088t	1212	1212	
160368z	839	839	
166297x	267	267	
172381r	923	923	
178973m	2562	2562	
188714q	1989	1989	
191807d	1709	1709	
196918b	48	48	
216410a	1140	1140	

243277z	995	995
249664v	34	34
255661r	1141	1141
286682k	391	391
294451d	824	824
298032c	3156	3156
310699m	421	421
370259u	1131	1131
371239k	711	711
372113w	1994	1994
391937c	1324	1324
431199d	67	67
485962l	487	487
502937q	102	102
549613p	1340	1340
550555b	511	511
558504v	235	235
561606e	4663	4663
565688t	477	477
574704k	150	150
609064x	2340	2340
614271h	861	861
694645v	360	360
709574n	2760	2760
717175o	589	589
752451k	257	257
761523d	683	683
789822b	208	208
797135h	833	833
855842w	138	138

physician_id	physician_attribute_4	physician_attribute_5	\
036439c	353	353	
040702s	652	652	
062765n	46	46	
084502f	3253	3253	
088034e	2554	2554	
091208a	1018	1018	
096731l	393	393	
119986g	76	76	
128011w	1	1	
144722h	702	702	
148376c	1080	1080	
157088t	1212	1212	
160368z	839	839	
166297x	267	267	

172381r	923	923
178973m	2562	2562
188714q	1989	1989
191807d	1709	1709
196918b	48	48
216410a	1140	1140
243277z	995	995
249664v	34	34
255661r	1141	1141
286682k	391	391
294451d	824	824
298032c	3156	3156
310699m	421	421
370259u	1131	1131
371239k	711	711
372113w	1994	1994
391937c	1324	1324
431199d	67	67
485962l	487	487
502937q	102	102
549613p	1340	1340
550555b	511	511
558504v	235	235
561606e	4663	4663
565688t	477	477
574704k	150	150
609064x	2340	2340
614271h	861	861
694645v	360	360
709574n	2760	2760
717175o	589	589
752451k	257	257
761523d	683	683
789822b	208	208
797135h	833	833
855842w	138	138

physician_id	physician_attribute_6	physician_attribute_7	\
036439c	353	353	
040702s	652	652	
062765n	46	46	
084502f	3253	3253	
088034e	2554	2554	
091208a	1018	1018	
096731l	393	393	
119986g	76	76	

128011w	1	1	
144722h	702	702	
148376c	1080	1080	
157088t	1212	1212	
160368z	839	839	
166297x	267	267	
172381r	923	923	
178973m	2562	2562	
188714q	1989	1989	
191807d	1709	1709	
196918b	48	48	
216410a	1140	1140	
243277z	995	995	
249664v	34	34	
255661r	1141	1141	
286682k	391	391	
294451d	824	824	
298032c	3156	3156	
310699m	421	421	
370259u	1131	1131	
371239k	711	711	
372113w	1994	1994	
391937c	1324	1324	
431199d	67	67	
485962l	487	487	
502937q	102	102	
549613p	1340	1340	
550555b	511	511	
558504v	235	235	
561606e	4663	4663	
565688t	477	477	
574704k	150	150	
609064x	2340	2340	
614271h	861	861	
694645v	360	360	
709574n	2760	2760	
717175o	589	589	
752451k	257	257	
761523d	683	683	
789822b	208	208	
797135h	833	833	
855842w	138	138	

physician_id	physician_attribute_8	…	physician_attribute_15	\
036439c	353	…	353	
040702s	652	…	652	

062765n	46	…	46
084502f	3253	…	3253
088034e	2554	…	2554
091208a	1018	…	1018
096731l	393	…	393
119986g	76	…	76
128011w	1	…	1
144722h	702	…	702
148376c	1080	…	1080
157088t	1212	…	1212
160368z	839	…	839
166297x	267	…	267
172381r	923	…	923
178973m	2562	…	2562
188714q	1989	…	1989
191807d	1709	…	1709
196918b	48	…	48
216410a	1140	…	1140
243277z	995	…	995
249664v	34	…	34
255661r	1141	…	1141
286682k	391	…	391
294451d	824	…	824
298032c	3156	…	3156
310699m	421	…	421
370259u	1131	…	1131
371239k	711	…	711
372113w	1994	…	1994
391937c	1324	…	1324
431199d	67	…	67
485962l	487	…	487
502937q	102	…	102
549613p	1340	…	1340
550555b	511	…	511
558504v	235	…	235
561606e	4663	…	4663
565688t	477	…	477
574704k	150	…	150
609064x	2340	…	2340
614271h	861	…	861
694645v	360	…	360
709574n	2760	…	2760
717175o	589	…	589
752451k	257	…	257
761523d	683	…	683
789822b	208	…	208
797135h	833	…	833

```
855842w                          138  …                    138

             physician_attribute_16  physician_attribute_17  \
physician_id
036439c                         353                     353
040702s                         652                     652
062765n                          46                      46
084502f                        3253                    3253
088034e                        2554                    2554
091208a                        1018                    1018
096731l                         393                     393
119986g                          76                      76
128011w                           1                       1
144722h                         702                     702
148376c                        1080                    1080
157088t                        1212                    1212
160368z                         839                     839
166297x                         267                     267
172381r                         923                     923
178973m                        2562                    2562
188714q                        1989                    1989
191807d                        1709                    1709
196918b                          48                      48
216410a                        1140                    1140
243277z                         995                     995
249664v                          34                      34
255661r                        1141                    1141
286682k                         391                     391
294451d                         824                     824
298032c                        3156                    3156
310699m                         421                     421
370259u                        1131                    1131
371239k                         711                     711
372113w                        1994                    1994
391937c                        1324                    1324
431199d                          67                      67
485962l                         487                     487
502937q                         102                     102
549613p                        1340                    1340
550555b                         511                     511
558504v                         235                     235
561606e                        4663                    4663
565688t                         477                     477
574704k                         150                     150
609064x                        2340                    2340
614271h                         861                     861
694645v                         360                     360
```

709574n	2760	2760
717175o	589	589
752451k	257	257
761523d	683	683
789822b	208	208
797135h	833	833
855842w	138	138

physician_id	physician_attribute_18	physician_attribute_19	\
036439c	353	353	
040702s	652	652	
062765n	46	46	
084502f	3253	3253	
088034e	2554	2554	
091208a	1018	1018	
096731l	393	393	
119986g	76	76	
128011w	1	1	
144722h	702	702	
148376c	1080	1080	
157088t	1212	1212	
160368z	839	839	
166297x	267	267	
172381r	923	923	
178973m	2562	2562	
188714q	1989	1989	
191807d	1709	1709	
196918b	48	48	
216410a	1140	1140	
243277z	995	995	
249664v	34	34	
255661r	1141	1141	
286682k	391	391	
294451d	824	824	
298032c	3156	3156	
310699m	421	421	
370259u	1131	1131	
371239k	711	711	
372113w	1994	1994	
391937c	1324	1324	
431199d	67	67	
485962l	487	487	
502937q	102	102	
549613p	1340	1340	
550555b	511	511	
558504v	235	235	

561606e	4663		4663
565688t	477		477
574704k	150		150
609064x	2340		2340
614271h	861		861
694645v	360		360
709574n	2760		2760
717175o	589		589
752451k	257		257
761523d	683		683
789822b	208		208
797135h	833		833
855842w	138		138

physician_id	physician_attribute_20	visit_date	procedure_code	\
036439c	353	353	353	
040702s	652	652	652	
062765n	46	46	46	
084502f	3253	3253	3253	
088034e	2554	2554	2554	
091208a	1018	1018	1018	
096731l	393	393	393	
119986g	76	76	76	
128011w	1	1	1	
144722h	702	702	702	
148376c	1080	1080	1080	
157088t	1212	1212	1212	
160368z	839	839	839	
166297x	267	267	267	
172381r	923	923	923	
178973m	2562	2562	2562	
188714q	1989	1989	1989	
191807d	1709	1709	1709	
196918b	48	48	48	
216410a	1140	1140	1140	
243277z	995	995	995	
249664v	34	34	34	
255661r	1141	1141	1141	
286682k	391	391	391	
294451d	824	824	824	
298032c	3156	3156	3156	
310699m	421	421	421	
370259u	1131	1131	1131	
371239k	711	711	711	
372113w	1994	1994	1994	
391937c	1324	1324	1324	

431199d	67	67	67
485962l	487	487	487
502937q	102	102	102
549613p	1340	1340	1340
550555b	511	511	511
558504v	235	235	235
561606e	4663	4663	4663
565688t	477	477	477
574704k	150	150	150
609064x	2340	2340	2340
614271h	861	861	861
694645v	360	360	360
709574n	2760	2760	2760
717175o	589	589	589
752451k	257	257	257
761523d	683	683	683
789822b	208	208	208
797135h	833	833	833
855842w	138	138	138

physician_id	utilization_quantity	expenditure_amount
036439c	353	353
040702s	652	652
062765n	46	46
084502f	3253	3253
088034e	2554	2554
091208a	1018	1018
096731l	393	393
119986g	76	76
128011w	1	1
144722h	702	702
148376c	1080	1080
157088t	1212	1212
160368z	839	839
166297x	267	267
172381r	923	923
178973m	2562	2562
188714q	1989	1989
191807d	1709	1709
196918b	48	48
216410a	1140	1140
243277z	995	995
249664v	34	34
255661r	1141	1141
286682k	391	391
294451d	824	824

298032c	3156	3156
310699m	421	421
370259u	1131	1131
371239k	711	711
372113w	1994	1994
391937c	1324	1324
431199d	67	67
485962l	487	487
502937q	102	102
549613p	1340	1340
550555b	511	511
558504v	235	235
561606e	4663	4663
565688t	477	477
574704k	150	150
609064x	2340	2340
614271h	861	861
694645v	360	360
709574n	2760	2760
717175o	589	589
752451k	257	257
761523d	683	683
789822b	208	208
797135h	833	833
855842w	138	138

[50 rows x 26 columns]

Upload Tables to PostgreSQL DB

Import Dependencies

[2434]:
```
import pandas as pd
from pathlib import Path
import psycopg2
from psycopg2 import OperationalError
from dotenv import load_dotenv
import os
from IPython.display import IFrame
```

Load Postegres Password from `.env` File
To download PostgreSQL: https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

Postgres password saved in file `.env`:
`db_password = 'THE_POSTGRES_PASSWORD_HERE'`

[2435]:
```
load_dotenv()
db_password = os.getenv("db_password")
```

[2436]:
```
# db_password
```

[2437]:
```
def create_connection(db_name, db_user, db_password, db_host, db_port):
    connection = None
    try:
        connection = psycopg2.connect(
            database=db_name,
            user=db_user,
            password=db_password,
            host=db_host,
            port=db_port,
        )
        # print("Connection to PostgreSQL DB successful")
    except OperationalError as e:
        print(f"The error '{e}' occurred")
    return connection
```

[2438]:
```
connection = create_connection(
    "postgres", "postgres", db_password, "127.0.0.1", "5432"
)
```

[2439]:
```
def create_database(connection, query):
    connection.autocommit = True
    cursor = connection.cursor()
    try:
        cursor.execute(query)
        print("Query executed successfully")
    except OperationalError as e:
        print(f"The error '{e}' occurred")
```

[2440]:
```
create_database_query = "CREATE DATABASE validate_health_db" # make sure to use␣
 ↪lower case
```

[2441]:
```
try:
    create_database(connection, create_database_query)
except:
    print('This database already exists.')
```

```
This database already exists.
```

[2442]:
```
def connection():
    return create_connection(
    "validate_health_db", "postgres", db_password, "127.0.0.1", "5432"
)
connection = connection()
```

[2443]:
```
def execute_query(connection, query):
    connection.autocommit = True
```

```
    cursor = connection.cursor()
    try:
        cursor.execute(query)
        print("Query executed successfully")
    except OperationalError as e:
        print(f"The error '{e}' occurred")
```

Create patient_procedure_info Table

[2444]:
```
patient_procedure_info_cols = patient_procedure_info.columns
patient_procedure_info_cols = list(patient_procedure_info_cols)
for e in range(len(patient_procedure_info_cols)):
    patient_procedure_info_cols[e] = '"{}"'.
 ↪format(patient_procedure_info_cols[e]) + " CHAR(15)" # For sake of␣
 ↪timeliness I have just used this datatpye.
patient_procedure_info_cols = str(patient_procedure_info_cols).strip("'[]").
 ↪replace("'", '')
print(patient_procedure_info_cols)
```

```
"patient_id" CHAR(15), "patient_attribute_1" CHAR(15), "patient_attribute_2"
CHAR(15), "patient_attribute_3" CHAR(15), "patient_attribute_4" CHAR(15),
"patient_attribute_5" CHAR(15), "patient_attribute_6" CHAR(15),
"patient_attribute_7" CHAR(15), "patient_attribute_8" CHAR(15),
"patient_attribute_9" CHAR(15), "patient_attribute_10" CHAR(15),
"patient_attribute_11" CHAR(15), "patient_attribute_12" CHAR(15),
"patient_attribute_13" CHAR(15), "patient_attribute_14" CHAR(15),
"patient_attribute_15" CHAR(15), "patient_attribute_16" CHAR(15),
"patient_attribute_17" CHAR(15), "patient_attribute_18" CHAR(15),
"patient_attribute_19" CHAR(15), "patient_attribute_20" CHAR(15)
```

[2445]:
```
next_query = """
DROP TABLE IF EXISTS patient_procedure_info;
"""

execute_query(connection, next_query)
```

```
Query executed successfully
```

[2446]:
```
create_patient_procedure_info = """
CREATE TABLE IF NOT EXISTS patient_procedure_info ({})
""".format(patient_procedure_info_cols)
```

[2447]:
```
# try:
execute_query(connection, create_patient_procedure_info)
# except:
    # print('This table already exists.')
```

```
Query executed successfully
```

[2448]:
```
csv_dataset = Path(str(Path.cwd()) + '/' + 'sqlDataModelingData' + '/' +␣
 ↪'patient_procedure_info.csv')
patient_procedure_info.to_csv(csv_dataset, index=False)
# patient_procedure_info.to_csv("patient_procedure_info.csv")
```

Import .csv Data Into patient_procedure_info Table

[2449]:
```
csv_dataset = """
COPY patient_procedure_info
FROM '{}'
DELIMITER ','
CSV HEADER;
""".format(str(csv_dataset))
```

[2450]:
```
execute_query(connection, csv_dataset)
# If an error gets thrown change the permissions to 'Everyone' ->'Full control'␣
 ↪in properties of the file (if using Windows).
# Use chmod if using macOS or Linux distro.
```

```
Query executed successfully
```

[2451]:
```
def execute_read_query(connection, query):
    result = None
    incrementor = 0
    try:
        cursor = connection.cursor()
        cursor.execute(query)
        result =cursor.fetchall()

        if len(result) < 10:
            for x in result:
                print(x)
        else:
            while incrementor < 15:
                print(result[incrementor])
                incrementor += 1
    except OperationalError as e:
        print(f"The error '{e}' occurred")

check_patient_procedure_info = """
SELECT *
FROM patient_procedure_info;
"""
# execute_read_query(connection, check_patient_procedure_info)
```

Create diagnosis_info Table

[2452]:
```
diagnosis_info_cols = diagnosis_info.columns
diagnosis_info_cols = list(diagnosis_info_cols)
```

```
for e in range(len(diagnosis_info_cols)):
    diagnosis_info_cols[e] = '"{}"'.format(diagnosis_info_cols[e]) + "␣
 ↪CHAR(15)" # For sake of timeliness I have just used this datatpye.
diagnosis_info_cols = str(diagnosis_info_cols).strip("'[]").replace("'", '')
print(diagnosis_info_cols)
```

```
"patient_id" CHAR(15), "patient_diagnosis_code_1" CHAR(15),
"patient_diagnosis_code_2" CHAR(15), "patient_diagnosis_code_3" CHAR(15),
"patient_diagnosis_code_4" CHAR(15), "patient_diagnosis_code_5" CHAR(15),
"patient_diagnosis_code_6" CHAR(15), "patient_diagnosis_code_7" CHAR(15),
"patient_diagnosis_code_8" CHAR(15), "patient_diagnosis_code_9" CHAR(15),
"patient_diagnosis_code_10" CHAR(15), "patient_diagnosis_code_11" CHAR(15),
"patient_diagnosis_code_12" CHAR(15), "patient_diagnosis_code_13" CHAR(15),
"patient_diagnosis_code_14" CHAR(15), "patient_diagnosis_code_15" CHAR(15),
"patient_diagnosis_code_16" CHAR(15), "patient_diagnosis_code_17" CHAR(15),
"patient_diagnosis_code_18" CHAR(15), "patient_diagnosis_code_19" CHAR(15),
"patient_diagnosis_code_20" CHAR(15), "patient_diagnosis_code_21" CHAR(15),
"patient_diagnosis_code_22" CHAR(15), "patient_diagnosis_code_23" CHAR(15),
"patient_diagnosis_code_24" CHAR(15), "patient_diagnosis_code_25" CHAR(15),
"patient_diagnosis_code_26" CHAR(15), "patient_diagnosis_code_27" CHAR(15),
"patient_diagnosis_code_28" CHAR(15), "patient_diagnosis_code_29" CHAR(15),
"patient_diagnosis_code_30" CHAR(15), "patient_diagnosis_code_31" CHAR(15),
"patient_diagnosis_code_32" CHAR(15), "patient_diagnosis_code_33" CHAR(15),
"patient_diagnosis_code_34" CHAR(15), "patient_diagnosis_code_35" CHAR(15),
"patient_diagnosis_code_36" CHAR(15), "patient_diagnosis_code_37" CHAR(15),
"patient_diagnosis_code_38" CHAR(15), "patient_diagnosis_code_39" CHAR(15),
"patient_diagnosis_code_40" CHAR(15), "patient_diagnosis_code_41" CHAR(15),
"patient_diagnosis_code_42" CHAR(15), "patient_diagnosis_code_43" CHAR(15),
"patient_diagnosis_code_44" CHAR(15), "patient_diagnosis_code_45" CHAR(15),
"patient_diagnosis_code_46" CHAR(15), "patient_diagnosis_code_47" CHAR(15),
"patient_diagnosis_code_48" CHAR(15), "patient_diagnosis_code_49" CHAR(15),
"patient_diagnosis_code_50" CHAR(15), "patient_diagnosis_code_51" CHAR(15),
"patient_diagnosis_code_52" CHAR(15), "patient_diagnosis_code_53" CHAR(15),
"patient_diagnosis_code_54" CHAR(15), "patient_diagnosis_code_55" CHAR(15),
"patient_diagnosis_code_56" CHAR(15), "patient_diagnosis_code_57" CHAR(15),
"patient_diagnosis_code_58" CHAR(15), "patient_diagnosis_code_59" CHAR(15),
"patient_diagnosis_code_60" CHAR(15), "patient_diagnosis_code_61" CHAR(15),
"patient_diagnosis_code_62" CHAR(15), "patient_diagnosis_code_63" CHAR(15),
"patient_diagnosis_code_64" CHAR(15), "patient_diagnosis_code_65" CHAR(15),
"patient_diagnosis_code_66" CHAR(15), "patient_diagnosis_code_67" CHAR(15),
"patient_diagnosis_code_68" CHAR(15), "patient_diagnosis_code_69" CHAR(15),
"patient_diagnosis_code_70" CHAR(15), "patient_diagnosis_code_71" CHAR(15),
"patient_diagnosis_code_72" CHAR(15), "patient_diagnosis_code_73" CHAR(15),
"patient_diagnosis_code_74" CHAR(15), "patient_diagnosis_code_75" CHAR(15),
"patient_diagnosis_code_76" CHAR(15), "patient_diagnosis_code_77" CHAR(15),
"patient_diagnosis_code_78" CHAR(15), "patient_diagnosis_code_79" CHAR(15),
"patient_diagnosis_code_80" CHAR(15), "patient_diagnosis_code_81" CHAR(15),
"patient_diagnosis_code_82" CHAR(15), "patient_diagnosis_code_83" CHAR(15),
```

```
"patient_diagnosis_code_84" CHAR(15), "patient_diagnosis_code_85" CHAR(15),
"patient_diagnosis_code_86" CHAR(15), "patient_diagnosis_code_87" CHAR(15),
"patient_diagnosis_code_88" CHAR(15), "patient_diagnosis_code_89" CHAR(15),
"patient_diagnosis_code_90" CHAR(15), "patient_diagnosis_code_91" CHAR(15),
"patient_diagnosis_code_92" CHAR(15), "patient_diagnosis_code_93" CHAR(15),
"patient_diagnosis_code_94" CHAR(15), "patient_diagnosis_code_95" CHAR(15),
"patient_diagnosis_code_96" CHAR(15), "patient_diagnosis_code_97" CHAR(15),
"patient_diagnosis_code_98" CHAR(15), "patient_diagnosis_code_99" CHAR(15),
"patient_diagnosis_code_100" CHAR(15)
```

[2453]:
```
next_query = """
DROP TABLE IF EXISTS diagnosis_info;
"""

execute_query(connection, next_query)
```

```
Query executed successfully
```

[2454]:
```
create_diagnosis_info_cols = """
CREATE TABLE IF NOT EXISTS diagnosis_info ({})
""".format(diagnosis_info_cols)
```

[2455]:
```
# try:
execute_query(connection, create_diagnosis_info_cols)
# except:
    # print('This table already exists.')
```

```
Query executed successfully
```

[2456]:
```
csv_dataset = Path(str(Path.cwd()) + '/' + 'sqlDataModelingData' + '/' +␣
 ↪'diagnosis_info.csv')
diagnosis_info.to_csv(csv_dataset, index=False)
# patient_procedure_info.to_csv("patient_procedure_info.csv")
```

Import .csv Data Into patient_procedure_info Table

[2457]:
```
csv_dataset = """
COPY diagnosis_info
FROM '{}'
DELIMITER ','
CSV HEADER;
""".format(str(csv_dataset))
```

[2458]:
```
execute_query(connection, csv_dataset)
# If an error gets thrown change the permissions to 'Everyone' ->'Full control'␣
 ↪in properties of the file (if using Windows).
# Use chmod if using macOS or Linux distro.
```

```
Query executed successfully
```

[2459]:
```
check_diagnosis_info = """
SELECT *
FROM diagnosis_info;
"""
# execute_read_query(connection, check_diagnosis_info)
```

Create visit_log Table

[2460]:
```
visit_log_cols = visit_log.columns
visit_log_cols = list(visit_log_cols)
for e in range(len(visit_log_cols)):
    visit_log_cols[e] = '"{}"'.format(visit_log_cols[e]) + " CHAR(15)" # For␣
 ↪sake of timeliness I have just used this datatpye.
visit_log_cols = str(visit_log_cols).strip("'[]").replace("'", '')
print(visit_log_cols)
```

```
"visit_id" CHAR(15), "patient_id" CHAR(15), "physician_id" CHAR(15),
"physician_attribute_1" CHAR(15), "physician_attribute_2" CHAR(15),
"physician_attribute_3" CHAR(15), "physician_attribute_4" CHAR(15),
"physician_attribute_5" CHAR(15), "physician_attribute_6" CHAR(15),
"physician_attribute_7" CHAR(15), "physician_attribute_8" CHAR(15),
"physician_attribute_9" CHAR(15), "physician_attribute_10" CHAR(15),
"physician_attribute_11" CHAR(15), "physician_attribute_12" CHAR(15),
"physician_attribute_13" CHAR(15), "physician_attribute_14" CHAR(15),
"physician_attribute_15" CHAR(15), "physician_attribute_16" CHAR(15),
"physician_attribute_17" CHAR(15), "physician_attribute_18" CHAR(15),
"physician_attribute_19" CHAR(15), "physician_attribute_20" CHAR(15),
"visit_date" CHAR(15), "procedure_code" CHAR(15), "utilization_quantity"
CHAR(15), "expenditure_amount" CHAR(15)
```

[2461]:
```
next_query = """
DROP TABLE IF EXISTS visit_log;
"""

execute_query(connection, next_query)
```

```
Query executed successfully
```

[2462]:
```
create_visit_log = """
CREATE TABLE IF NOT EXISTS visit_log ({})
""".format(visit_log_cols)
```

[2463]:
```
# try:
execute_query(connection, create_visit_log)
# except:
    # print('This table already exists.')
```

```
Query executed successfully
```

[2464]:
```
csv_dataset = Path(str(Path.cwd()) + '/' + 'sqlDataModelingData' + '/' +␣
 ↪'visit_log.csv')
visit_log.to_csv(csv_dataset, index=False)
# patient_procedure_info.to_csv("patient_procedure_info.csv")
```

Import .csv Data Into patient_procedure_info Table

[2465]:
```
csv_dataset = """
COPY visit_log
FROM '{}'
DELIMITER ','
CSV HEADER;
""".format(str(csv_dataset))
```

[2466]:
```
execute_query(connection, csv_dataset)
# If an error gets thrown change the permissions to 'Everyone' ->'Full control'␣
 ↪in properties of the file (if using Windows).
# Use chmod if using macOS or Linux distro.
```

```
Query executed successfully
```

[2467]:
```
check_diagnosis_info = """
SELECT *
FROM visit_log;
"""
# execute_read_query(connection, check_diagnosis_info)
```

[2468]:
```
# Helper Queries
```

[2469]:
```
# next_query = """
# DROP DATABASE IF EXISTS  validate_health_db;
# """

# execute_read_query(connection, next_query)
```

[2470]:
```
# next_query = """
# DROP TABLE IF EXISTS patient_procedure_info;
# """

# execute_query(connection, next_query)
```

[2471]:
```
# next_query = """
# DROP TABLE IF EXISTS diagnosis_info;
# """

# execute_query(connection, next_query)
```

[2472]:
```
# next_query = """
# DROP TABLE IF EXISTS visit_log;
# """

# execute_query(connection, next_query)
```

1.2.2 Answer Query

My assumption is that the `patient_id` has no dublicates in the `diagnosis_info` table. The code I created to synthesize the data may have a small probability that it has duplicate `diagnosis_code`'s in the same row. I am unsure if I this would be out of context or not for this dataset. But I know how to update the functions to synthesize the data well. If we were to do dealing with a very large dataset, we could also use the `md5()` function. That creates a hash of a string passed to it. We might consider doing this for tables in slightly different formats to reduce compute resources.

[2473]:
```
# next_query = """
# SELECT column1, COUNT(column1) AS number_count
# FROM (
#     (SELECT  patient_diagnosis_code_1
#     AS column1 FROM diagnosis_info )
#     union ALL
#     (SELECT  patient_diagnosis_code_2
#     AS column2 FROM diagnosis_info )
#     union ALL
#     (SELECT  patient_diagnosis_code_3
#     AS column3 FROM diagnosis_info )
#     union ALL
#     (SELECT  patient_diagnosis_code_4
#     AS column4 FROM diagnosis_info )
#     union ALL
#     (SELECT  patient_diagnosis_code_5
#     AS column5 FROM diagnosis_info )
#     ...
#     ...
#     ...
#      union ALL
#     (SELECT  patient_diagnosis_code_100
#     AS column100 FROM diagnosis_info )
#     )AS big_query
# GROUP BY column1 ORDER BY number_count desc
# LIMIT 1;
# """
# execute_read_query(connection, next_query)
```

[2474]:
```
def answer_query(diagnosis_info_df):
    patient_diagnosis_query_list = []
```

```
    for e in range(len(diagnosis_info_df.columns.to_list()[1:])):
        col_num = e+1
        if e == 0:
            patient_diagnosis_query_list.append("""
            SELECT column1, COUNT(column1) AS number_count
            FROM (
                (SELECT {}
                AS column{} FROM diagnosis_info)
                UNION ALL
            """.format(diagnosis_info_df.columns.to_list()[1:][e], col_num))
        elif e != len(diagnosis_info_df.columns.to_list()[1:]) - 1:
            patient_diagnosis_query_list.append("""
            (SELECT {}
            AS column{} FROM diagnosis_info)
            UNION ALL
            """.format(diagnosis_info_df.columns.to_list()[1:][e], col_num))
        else:
            patient_diagnosis_query_list.append("""
            (SELECT {}
            AS column{} FROM diagnosis_info)
            )AS big_query
            GROUP BY column1 ORDER BY number_count desc
            LIMIT 1;
            """.format(diagnosis_info_df.columns.to_list()[1:][e], col_num))
    patient_diagnosis_query_list = str(patient_diagnosis_query_list).
 ↪strip("'[]").replace("'", '').replace(",", "").replace('SELECT column1',␣
 ↪'SELECT column1,').replace('\\n', '').replace('    ', '\n')
    # print(patient_diagnosis_query_list)

    execute_read_query(connection, patient_diagnosis_query_list)

answer_query(diagnosis_info)
```

```
('z40.776        ', 168)
```

We see that the answer is validated in both SQL and Pandas.

[2475]:
```
most_common_code = diagnosis_info.iloc[:,1:].apply(pd.Series.value_counts).
 ↪sum(1).idxmax()
most_common_cod_freq = diagnosis_info.iloc[:,1:].apply(pd.Series.value_counts).
 ↪loc[most_common_code].sum()

print("The most common diagnosis code is: \n\t{}: {}\n\tAssuming that the␣
 ↪patient_id is the primary key and does not repeat itself.".
 ↪format(most_common_code, most_common_cod_freq))
```

```
The most common diagnosis code is:
        z40.776: 168.0
```

Assuming that the patient_id is the primary key and does not repeat itself.

1.2.3 b. Let's say in the future, it's possible that the number of "patient_diagnosis_code" columns will increase or decrease. Write code so that it's dynamic to handle any arbitrary number of "patient_diagnosis_code" columns.

[2505]:
```
next_query = """
DROP TABLE IF EXISTS diagnosis_info;
"""

execute_query(connection, next_query)
```

```
Query executed successfully
```

Load data from .csv to save time from resynthesizing

[2501]:
```
csv_dataset_diagnosis_info = Path(str(Path.cwd()) + '/' + 'sqlDataModelingData'␣
↪+ '/' + 'diagnosis_info_big.csv')
diagnosis_info = pd.read_csv(csv_dataset_diagnosis_info)
diagnosis_info
```

[2501]:

	patient_id	patient_diagnosis_code_1	patient_diagnosis_code_2	\
0	225060g	f85.017	u28.99	
1	071785r	l66.369	x16.654	
2	336965v	f56.808	r59.611	
3	552897v	w29.355	e85.09	
4	513589u	r35.468	t24.885	
…	…	…	…	
999995	375515f	f74.913	z65.127	
999996	326080a	f43.121	m53.722	
999997	181999t	d31.717	p74.88	
999998	154917g	z28.341	p19.159	
999999	199071g	d54.617	b29.62	

	patient_diagnosis_code_3	patient_diagnosis_code_4	\
0	u46.112	t62.079	
1	o18.756	158.12	
2	u57.346	l45.517	
3	g70.515	l46.544	
4	n68.069	c15.39	
…	…	…	
999995	g70.515	n74.497	
999996	p69.661	c75.52	
999997	p30.185	g12.45	
999998	o18.756	n71.478	
999999	x52.023	g38.497	

```
       patient_diagnosis_code_5 patient_diagnosis_code_6  \
0                       v32.744                  o37.903
1                       b72.492                  y62.185
2                        v47.34                   a27.27
3                        l55.23                  c68.187
4                        y28.22                  h20.734
…                           …                        …
999995                  p69.111                   u83.89
999996                  y44.903                  h49.681
999997                   i36.16                    r65.7
999998                  y89.096                  f24.957
999999                  n27.981                  l46.692

       patient_diagnosis_code_7 patient_diagnosis_code_8  \
0                       l19.883                  j74.748
1                       s66.956                  a65.981
2                       s14.248                  g40.708
3                       m59.463                  f57.083
4                       s20.394                   f20.87
…                           …                        …
999995                  j67.959                  p19.583
999996                  b48.439                  b38.282
999997                  d13.021                  o44.854
999998                  z47.722                  o44.968
999999                  d51.207                  x59.285

       patient_diagnosis_code_9  … patient_diagnosis_code_241  \
0                        k50.35  …                    t23.631
1                       z53.482  …                    x23.233
2                       z46.246  …                    v45.286
3                       u31.221  …                    v08.252
4                       a90.584  …                    x82.029
…                           …  …                        …
999995                  o55.208  …                    v85.515
999996                  z86.373  …                    a66.451
999997                  z50.411  …                    f45.097
999998                  f27.381  …                    t45.616
999999                  k66.482  …                    a63.646

       patient_diagnosis_code_242 patient_diagnosis_code_243  \
0                         k56.581                   q49.869
1                         b61.274                   m53.967
2                         y55.321                   y78.171
3                         w36.181                   x51.557
4                         w24.998                   r72.759
…                             …                         …
999995                    m56.212                   q37.387
```

```
999996                    d39.027                    m71.381
999997                    m56.212                    o30.463
999998                    w43.495                    v68.816
999999                    w36.181                    l17.507

       patient_diagnosis_code_244 patient_diagnosis_code_245  \
0                         p82.075                    i47.211
1                         t60.884                    c59.821
2                         w41.289                    n54.524
3                         w23.794                    j19.963
4                         d76.026                    m69.563
…                           …                        …
999995                    a59.986                    p65.627
999996                    c81.724                    j83.612
999997                    h59.296                    s61.325
999998                    t34.082                    n43.981
999999                    e74.191                    c53.658

       patient_diagnosis_code_246 patient_diagnosis_code_247  \
0                         b57.141                    g51.954
1                         f91.819                    c73.313
2                         f27.123                     h84.64
3                          e68.55                     n26.99
4                         u41.789                    i51.421
…                           …                        …
999995                    h62.428                    c28.287
999996                    b68.078                    u21.347
999997                    b35.261                    e31.748
999998                    u42.091                    x57.034
999999                    c45.844                    j64.549

       patient_diagnosis_code_248 patient_diagnosis_code_249  \
0                          w42.14                    q51.264
1                         r45.241                    s52.367
2                         x54.168                    q66.029
3                         f78.392                    g65.868
4                         r33.531                    y44.278
…                           …                        …
999995                     x49.53                    i39.681
999996                    d71.307                    s22.208
999997                    k23.645                    v44.286
999998                    a39.008                    v68.961
999999                    a42.681                    s72.205

       patient_diagnosis_code_250
0                         c87.897
1                          b45.89
```

```
2                       w29.481
3                       b15.069
4                       v29.581
…                         …
999995                  q29.383
999996                  b73.302
999997                  q29.383
999998                  z46.091
999999                  v33.308

[1000000 rows x 251 columns]
```

[2502]:
```
# num_synthetic_rows = 1000000
# num_unique_patients = 1000000
# num_unique_diagnosis = 300
# num_patient_diagnosis_codes = 250

# diagnosis_info = create_diagnosis_info(num_synthetic_rows,␣
 ↪num_unique_patients, num_unique_diagnosis, num_patient_diagnosis_codes)
# diagnosis_info
# diagnosis_info.head()
```

[2503]:
```
# Save Dataframe to reduce time to resynthesize
# csv_dataset_diagnosis_info = Path(str(Path.cwd()) + '/' +␣
 ↪'sqlDataModelingData' + '/' + 'diagnosis_info_big.csv')
# diagnosis_info.to_csv(csv_dataset_diagnosis_info, index=False)
```

Recreate diagnosis_info Table

[2504]:
```
diagnosis_info_cols = diagnosis_info.columns
diagnosis_info_cols = list(diagnosis_info_cols)
for e in range(len(diagnosis_info_cols)):
    diagnosis_info_cols[e] = '"{}"'.format(diagnosis_info_cols[e]) + "␣
 ↪CHAR(15)" # For sake of timeliness I have just used this datatpye.
diagnosis_info_cols = str(diagnosis_info_cols).strip("'[]").replace("'", '')
print(diagnosis_info_cols)
```

```
"patient_id" CHAR(15), "patient_diagnosis_code_1" CHAR(15),
"patient_diagnosis_code_2" CHAR(15), "patient_diagnosis_code_3" CHAR(15),
"patient_diagnosis_code_4" CHAR(15), "patient_diagnosis_code_5" CHAR(15),
"patient_diagnosis_code_6" CHAR(15), "patient_diagnosis_code_7" CHAR(15),
"patient_diagnosis_code_8" CHAR(15), "patient_diagnosis_code_9" CHAR(15),
"patient_diagnosis_code_10" CHAR(15), "patient_diagnosis_code_11" CHAR(15),
"patient_diagnosis_code_12" CHAR(15), "patient_diagnosis_code_13" CHAR(15),
"patient_diagnosis_code_14" CHAR(15), "patient_diagnosis_code_15" CHAR(15),
"patient_diagnosis_code_16" CHAR(15), "patient_diagnosis_code_17" CHAR(15),
"patient_diagnosis_code_18" CHAR(15), "patient_diagnosis_code_19" CHAR(15),
"patient_diagnosis_code_20" CHAR(15), "patient_diagnosis_code_21" CHAR(15),
```

```
"patient_diagnosis_code_22" CHAR(15), "patient_diagnosis_code_23" CHAR(15),
"patient_diagnosis_code_24" CHAR(15), "patient_diagnosis_code_25" CHAR(15),
"patient_diagnosis_code_26" CHAR(15), "patient_diagnosis_code_27" CHAR(15),
"patient_diagnosis_code_28" CHAR(15), "patient_diagnosis_code_29" CHAR(15),
"patient_diagnosis_code_30" CHAR(15), "patient_diagnosis_code_31" CHAR(15),
"patient_diagnosis_code_32" CHAR(15), "patient_diagnosis_code_33" CHAR(15),
"patient_diagnosis_code_34" CHAR(15), "patient_diagnosis_code_35" CHAR(15),
"patient_diagnosis_code_36" CHAR(15), "patient_diagnosis_code_37" CHAR(15),
"patient_diagnosis_code_38" CHAR(15), "patient_diagnosis_code_39" CHAR(15),
"patient_diagnosis_code_40" CHAR(15), "patient_diagnosis_code_41" CHAR(15),
"patient_diagnosis_code_42" CHAR(15), "patient_diagnosis_code_43" CHAR(15),
"patient_diagnosis_code_44" CHAR(15), "patient_diagnosis_code_45" CHAR(15),
"patient_diagnosis_code_46" CHAR(15), "patient_diagnosis_code_47" CHAR(15),
"patient_diagnosis_code_48" CHAR(15), "patient_diagnosis_code_49" CHAR(15),
"patient_diagnosis_code_50" CHAR(15), "patient_diagnosis_code_51" CHAR(15),
"patient_diagnosis_code_52" CHAR(15), "patient_diagnosis_code_53" CHAR(15),
"patient_diagnosis_code_54" CHAR(15), "patient_diagnosis_code_55" CHAR(15),
"patient_diagnosis_code_56" CHAR(15), "patient_diagnosis_code_57" CHAR(15),
"patient_diagnosis_code_58" CHAR(15), "patient_diagnosis_code_59" CHAR(15),
"patient_diagnosis_code_60" CHAR(15), "patient_diagnosis_code_61" CHAR(15),
"patient_diagnosis_code_62" CHAR(15), "patient_diagnosis_code_63" CHAR(15),
"patient_diagnosis_code_64" CHAR(15), "patient_diagnosis_code_65" CHAR(15),
"patient_diagnosis_code_66" CHAR(15), "patient_diagnosis_code_67" CHAR(15),
"patient_diagnosis_code_68" CHAR(15), "patient_diagnosis_code_69" CHAR(15),
"patient_diagnosis_code_70" CHAR(15), "patient_diagnosis_code_71" CHAR(15),
"patient_diagnosis_code_72" CHAR(15), "patient_diagnosis_code_73" CHAR(15),
"patient_diagnosis_code_74" CHAR(15), "patient_diagnosis_code_75" CHAR(15),
"patient_diagnosis_code_76" CHAR(15), "patient_diagnosis_code_77" CHAR(15),
"patient_diagnosis_code_78" CHAR(15), "patient_diagnosis_code_79" CHAR(15),
"patient_diagnosis_code_80" CHAR(15), "patient_diagnosis_code_81" CHAR(15),
"patient_diagnosis_code_82" CHAR(15), "patient_diagnosis_code_83" CHAR(15),
"patient_diagnosis_code_84" CHAR(15), "patient_diagnosis_code_85" CHAR(15),
"patient_diagnosis_code_86" CHAR(15), "patient_diagnosis_code_87" CHAR(15),
"patient_diagnosis_code_88" CHAR(15), "patient_diagnosis_code_89" CHAR(15),
"patient_diagnosis_code_90" CHAR(15), "patient_diagnosis_code_91" CHAR(15),
"patient_diagnosis_code_92" CHAR(15), "patient_diagnosis_code_93" CHAR(15),
"patient_diagnosis_code_94" CHAR(15), "patient_diagnosis_code_95" CHAR(15),
"patient_diagnosis_code_96" CHAR(15), "patient_diagnosis_code_97" CHAR(15),
"patient_diagnosis_code_98" CHAR(15), "patient_diagnosis_code_99" CHAR(15),
"patient_diagnosis_code_100" CHAR(15), "patient_diagnosis_code_101" CHAR(15),
"patient_diagnosis_code_102" CHAR(15), "patient_diagnosis_code_103" CHAR(15),
"patient_diagnosis_code_104" CHAR(15), "patient_diagnosis_code_105" CHAR(15),
"patient_diagnosis_code_106" CHAR(15), "patient_diagnosis_code_107" CHAR(15),
"patient_diagnosis_code_108" CHAR(15), "patient_diagnosis_code_109" CHAR(15),
"patient_diagnosis_code_110" CHAR(15), "patient_diagnosis_code_111" CHAR(15),
"patient_diagnosis_code_112" CHAR(15), "patient_diagnosis_code_113" CHAR(15),
"patient_diagnosis_code_114" CHAR(15), "patient_diagnosis_code_115" CHAR(15),
"patient_diagnosis_code_116" CHAR(15), "patient_diagnosis_code_117" CHAR(15),
```

```
"patient_diagnosis_code_118" CHAR(15), "patient_diagnosis_code_119" CHAR(15),
"patient_diagnosis_code_120" CHAR(15), "patient_diagnosis_code_121" CHAR(15),
"patient_diagnosis_code_122" CHAR(15), "patient_diagnosis_code_123" CHAR(15),
"patient_diagnosis_code_124" CHAR(15), "patient_diagnosis_code_125" CHAR(15),
"patient_diagnosis_code_126" CHAR(15), "patient_diagnosis_code_127" CHAR(15),
"patient_diagnosis_code_128" CHAR(15), "patient_diagnosis_code_129" CHAR(15),
"patient_diagnosis_code_130" CHAR(15), "patient_diagnosis_code_131" CHAR(15),
"patient_diagnosis_code_132" CHAR(15), "patient_diagnosis_code_133" CHAR(15),
"patient_diagnosis_code_134" CHAR(15), "patient_diagnosis_code_135" CHAR(15),
"patient_diagnosis_code_136" CHAR(15), "patient_diagnosis_code_137" CHAR(15),
"patient_diagnosis_code_138" CHAR(15), "patient_diagnosis_code_139" CHAR(15),
"patient_diagnosis_code_140" CHAR(15), "patient_diagnosis_code_141" CHAR(15),
"patient_diagnosis_code_142" CHAR(15), "patient_diagnosis_code_143" CHAR(15),
"patient_diagnosis_code_144" CHAR(15), "patient_diagnosis_code_145" CHAR(15),
"patient_diagnosis_code_146" CHAR(15), "patient_diagnosis_code_147" CHAR(15),
"patient_diagnosis_code_148" CHAR(15), "patient_diagnosis_code_149" CHAR(15),
"patient_diagnosis_code_150" CHAR(15), "patient_diagnosis_code_151" CHAR(15),
"patient_diagnosis_code_152" CHAR(15), "patient_diagnosis_code_153" CHAR(15),
"patient_diagnosis_code_154" CHAR(15), "patient_diagnosis_code_155" CHAR(15),
"patient_diagnosis_code_156" CHAR(15), "patient_diagnosis_code_157" CHAR(15),
"patient_diagnosis_code_158" CHAR(15), "patient_diagnosis_code_159" CHAR(15),
"patient_diagnosis_code_160" CHAR(15), "patient_diagnosis_code_161" CHAR(15),
"patient_diagnosis_code_162" CHAR(15), "patient_diagnosis_code_163" CHAR(15),
"patient_diagnosis_code_164" CHAR(15), "patient_diagnosis_code_165" CHAR(15),
"patient_diagnosis_code_166" CHAR(15), "patient_diagnosis_code_167" CHAR(15),
"patient_diagnosis_code_168" CHAR(15), "patient_diagnosis_code_169" CHAR(15),
"patient_diagnosis_code_170" CHAR(15), "patient_diagnosis_code_171" CHAR(15),
"patient_diagnosis_code_172" CHAR(15), "patient_diagnosis_code_173" CHAR(15),
"patient_diagnosis_code_174" CHAR(15), "patient_diagnosis_code_175" CHAR(15),
"patient_diagnosis_code_176" CHAR(15), "patient_diagnosis_code_177" CHAR(15),
"patient_diagnosis_code_178" CHAR(15), "patient_diagnosis_code_179" CHAR(15),
"patient_diagnosis_code_180" CHAR(15), "patient_diagnosis_code_181" CHAR(15),
"patient_diagnosis_code_182" CHAR(15), "patient_diagnosis_code_183" CHAR(15),
"patient_diagnosis_code_184" CHAR(15), "patient_diagnosis_code_185" CHAR(15),
"patient_diagnosis_code_186" CHAR(15), "patient_diagnosis_code_187" CHAR(15),
"patient_diagnosis_code_188" CHAR(15), "patient_diagnosis_code_189" CHAR(15),
"patient_diagnosis_code_190" CHAR(15), "patient_diagnosis_code_191" CHAR(15),
"patient_diagnosis_code_192" CHAR(15), "patient_diagnosis_code_193" CHAR(15),
"patient_diagnosis_code_194" CHAR(15), "patient_diagnosis_code_195" CHAR(15),
"patient_diagnosis_code_196" CHAR(15), "patient_diagnosis_code_197" CHAR(15),
"patient_diagnosis_code_198" CHAR(15), "patient_diagnosis_code_199" CHAR(15),
"patient_diagnosis_code_200" CHAR(15), "patient_diagnosis_code_201" CHAR(15),
"patient_diagnosis_code_202" CHAR(15), "patient_diagnosis_code_203" CHAR(15),
"patient_diagnosis_code_204" CHAR(15), "patient_diagnosis_code_205" CHAR(15),
"patient_diagnosis_code_206" CHAR(15), "patient_diagnosis_code_207" CHAR(15),
"patient_diagnosis_code_208" CHAR(15), "patient_diagnosis_code_209" CHAR(15),
"patient_diagnosis_code_210" CHAR(15), "patient_diagnosis_code_211" CHAR(15),
"patient_diagnosis_code_212" CHAR(15), "patient_diagnosis_code_213" CHAR(15),
```

```
"patient_diagnosis_code_214" CHAR(15), "patient_diagnosis_code_215" CHAR(15),
"patient_diagnosis_code_216" CHAR(15), "patient_diagnosis_code_217" CHAR(15),
"patient_diagnosis_code_218" CHAR(15), "patient_diagnosis_code_219" CHAR(15),
"patient_diagnosis_code_220" CHAR(15), "patient_diagnosis_code_221" CHAR(15),
"patient_diagnosis_code_222" CHAR(15), "patient_diagnosis_code_223" CHAR(15),
"patient_diagnosis_code_224" CHAR(15), "patient_diagnosis_code_225" CHAR(15),
"patient_diagnosis_code_226" CHAR(15), "patient_diagnosis_code_227" CHAR(15),
"patient_diagnosis_code_228" CHAR(15), "patient_diagnosis_code_229" CHAR(15),
"patient_diagnosis_code_230" CHAR(15), "patient_diagnosis_code_231" CHAR(15),
"patient_diagnosis_code_232" CHAR(15), "patient_diagnosis_code_233" CHAR(15),
"patient_diagnosis_code_234" CHAR(15), "patient_diagnosis_code_235" CHAR(15),
"patient_diagnosis_code_236" CHAR(15), "patient_diagnosis_code_237" CHAR(15),
"patient_diagnosis_code_238" CHAR(15), "patient_diagnosis_code_239" CHAR(15),
"patient_diagnosis_code_240" CHAR(15), "patient_diagnosis_code_241" CHAR(15),
"patient_diagnosis_code_242" CHAR(15), "patient_diagnosis_code_243" CHAR(15),
"patient_diagnosis_code_244" CHAR(15), "patient_diagnosis_code_245" CHAR(15),
"patient_diagnosis_code_246" CHAR(15), "patient_diagnosis_code_247" CHAR(15),
"patient_diagnosis_code_248" CHAR(15), "patient_diagnosis_code_249" CHAR(15),
"patient_diagnosis_code_250" CHAR(15)
```

[2506]:
```
create_diagnosis_info_cols = """
CREATE TABLE IF NOT EXISTS diagnosis_info ({})
""".format(diagnosis_info_cols)
```

[2507]:
```
# try:
execute_query(connection, create_diagnosis_info_cols)
# except:
    # print('This table already exists.')
```

```
Query executed successfully
```

[2481]:
```
# csv_dataset = Path(str(Path.cwd()) + '/' + 'sqlDataModelingData' + '/' +␣
 ↪'diagnosis_info.csv')
# diagnosis_info.to_csv(csv_dataset, index=False)
# patient_procedure_info.to_csv("patient_procedure_info.csv")
```

Import .csv Data Into patient_procedure_info Table

[2508]:
```
csv_dataset = """
COPY diagnosis_info
FROM '{}'
DELIMITER ','
CSV HEADER;
""".format(str(csv_dataset_diagnosis_info))
```

[2509]:
```
execute_query(connection, csv_dataset)
# If an error gets thrown change the permissions to 'Everyone' ->'Full control'␣
 ↪in properties of the file (if using Windows).
```

```
# Use chmod if using macOS or Linux distro.
```

```
Query executed successfully
```

1.2.4 Answer Query 2

Perhaps a purely SQL query was, what was sought after. In the event that this is the case. I would just use a subquery leveraging the information schema and the colums. Surely there are additional techniques as well.

[2484]:
```
answer_query(diagnosis_info)
```

```
('t76.061        ', 37351)
```

We see that the answer is validated in both SQL and Pandas.

[2485]:
```
most_common_code = diagnosis_info.iloc[:,1:].apply(pd.Series.value_counts).
 ↪sum(1).idxmax()
most_common_cod_freq = diagnosis_info.iloc[:,1:].apply(pd.Series.value_counts).
 ↪loc[most_common_code].sum()

print("The most common diagnosis code is: \n\t{}: {}\n\tAssuming that the␣
 ↪patient_id is the primary key and does not repeat itself.".
 ↪format(most_common_code, most_common_cod_freq))
```

```
The most common diagnosis code is:
        t76.061: 37351.0
        Assuming that the patient_id is the primary key and does not repeat
itself.
```

1.2.5 c. Write SQL to load this data into new tables, such that it works for regression or machine learning. Explain your reasoning.

1.2.6 EXPLANATION of columns:

A **patient** has a series of attributes that have been updated as of the last doctor **visit**.
A **physician** has a series of attributes that are static across all visits and patients.

A patient has a list of associated health **diagnosis** codes updated as of the latest visit.
Example diagnosis_codes:
E11.9 – generic diabetes
C61 – prostate cancer
L89.103 – pressure ulcer
A **visit** is when a specific **patient** sees a specific **physician** who provides a specific **procedure** on a specific **date**

2 Data Cleaning/Feature Engineering Answer

Given the explanation of the columns, the `patient_attribute`'s & `patient_diagnosis`'s can only be updated for the most recent visit. That is making the assumption that the `visit_id` column is

numerical and ascending, corresponding to an instance that a patient sees an individual physician with a `procedure_code` on a certain date. The `visit_id` could be considered a surrogate key. All of these make up a candidate key for the `visit_log`. The goal with these three tables, would be to merge them into a single table. If this is done everytime the *patient* has a *visit*, we can have both the `patient_attribute`'s & `patient_diagnosis`'s data for each **unique** `visit_id`. But *ex post facto*, we can only join with the most recent `visit_id` value. Once this is done, if we are going to do ML/DL or other high level analysis involving Maths, we will need to numerically encode the categorical data using either one hot encoding or label encoding. I am not aware of any SQL techniques that allow us to do that, with out some extremely strenous queries. Because the data needs to enter a Python or R IDE most likely, in order to perform the technical analysis such as ML/DL. The preferred method would be to merge as much as the data as possible (as well as dropping known *dud* columns) into a single table before importing it to Python or R. A simple `Select * from <table_name_here>` and cleaning the data within python would still be a valid solution, although it may be less efficient. But in certain instances certain Python functions may be more efficient than SQL queries (and vice versa). Because a merged table would have a column `visit_date` that is a time value, we could change the index to this as well. That way we would have the data in a proper time series format. Using principal component analysis (PCA), probably a good idea, if we are suggesting we are dealing with potentially hundreds of columns. Some additional exploratory analysis is probably warranted as well, looking for outliers and the overall distribution of the data.

Load data from `.csv`'s to save time on resynthesizing data.

[27]:
```
import pandas as pd
from pathlib import Path

csv_patient_procedure_info_big = Path(str(Path.cwd()) + '/' +␣
 ↪'sqlDataModelingData' + '/' + 'patient_procedure_info_big.csv')
patient_procedure_info = pd.read_csv(csv_dataset_patient_procedure_info_big)
```

[30]:
```
csv_dataset_visit_log_info_big = Path(str(Path.cwd()) + '/' +␣
 ↪'sqlDataModelingData' + '/' + 'visit_log_info_big.csv')
visit_log = pd.read_csv(csv_dataset_visit_log_info_big)
```

[29]:
```
# # Uncomment if did not execute above code cells
# csv_dataset_diagnosis_info = Path(str(Path.cwd()) + '/' +␣
 ↪'sqlDataModelingData' + '/' + 'diagnosis_info_big.csv')
# diagnosis_info = pd.read_csv(csv_dataset_diagnosis_info)
# diagnosis_info
```

[29]:

	patient_id	patient_diagnosis_code_1	patient_diagnosis_code_2	\
0	225060g	f85.017	u28.99	
1	071785r	l66.369	x16.654	
2	336965v	f56.808	r59.611	
3	552897v	w29.355	e85.09	
4	513589u	r35.468	t24.885	
…	…	…	…	
999995	375515f	f74.913	z65.127	

```
999996    326080a                 f43.121                 m53.722
999997    181999t                 d31.717                  p74.88
999998    154917g                 z28.341                 p19.159
999999    199071g                 d54.617                  b29.62

       patient_diagnosis_code_3 patient_diagnosis_code_4  \
0                        u46.112                  t62.079
1                        o18.756                   l58.12
2                        u57.346                  l45.517
3                        g70.515                  l46.544
4                        n68.069                   c15.39
…                          …                      …
999995                   g70.515                  n74.497
999996                   p69.661                   c75.52
999997                   p30.185                   g12.45
999998                   o18.756                  n71.478
999999                   x52.023                  g38.497

       patient_diagnosis_code_5 patient_diagnosis_code_6  \
0                        v32.744                  o37.903
1                        b72.492                  y62.185
2                         v47.34                   a27.27
3                         l55.23                  c68.187
4                         y28.22                  h20.734
…                          …                      …
999995                   p69.111                   u83.89
999996                   y44.903                  h49.681
999997                    i36.16                    r65.7
999998                   y89.096                  f24.957
999999                   n27.981                  l46.692

       patient_diagnosis_code_7 patient_diagnosis_code_8  \
0                        l19.883                  j74.748
1                        s66.956                  a65.981
2                        s14.248                  g40.708
3                        m59.463                  f57.083
4                        s20.394                   f20.87
…                          …                      …
999995                   j67.959                  p19.583
999996                   b48.439                  b38.282
999997                   d13.021                  o44.854
999998                   z47.722                  o44.968
999999                   d51.207                  x59.285

       patient_diagnosis_code_9  … patient_diagnosis_code_241  \
0                         k50.35  …                    t23.631
1                        z53.482  …                    x23.233
```

```
2                      z46.246  …                    v45.286
3                      u31.221  …                    v08.252
4                      a90.584  …                    x82.029
…                        …  …                       …
999995                 o55.208  …                    v85.515
999996                 z86.373  …                    a66.451
999997                 z50.411  …                    f45.097
999998                 f27.381  …                    t45.616
999999                 k66.482  …                    a63.646

       patient_diagnosis_code_242 patient_diagnosis_code_243  \
0                        k56.581                    q49.869
1                        b61.274                    m53.967
2                        y55.321                    y78.171
3                        w36.181                    x51.557
4                        w24.998                    r72.759
…                          …                        …
999995                   m56.212                    q37.387
999996                   d39.027                    m71.381
999997                   m56.212                    o30.463
999998                   w43.495                    v68.816
999999                   w36.181                    l17.507

       patient_diagnosis_code_244 patient_diagnosis_code_245  \
0                        p82.075                    i47.211
1                        t60.884                    c59.821
2                        w41.289                    n54.524
3                        w23.794                    j19.963
4                        d76.026                    m69.563
…                          …                        …
999995                   a59.986                    p65.627
999996                   c81.724                    j83.612
999997                   h59.296                    s61.325
999998                   t34.082                    n43.981
999999                   e74.191                    c53.658

       patient_diagnosis_code_246 patient_diagnosis_code_247  \
0                        b57.141                    g51.954
1                        f91.819                    c73.313
2                        f27.123                     h84.64
3                         e68.55                     n26.99
4                        u41.789                    i51.421
…                          …                        …
999995                   h62.428                    c28.287
999996                   b68.078                    u21.347
999997                   b35.261                    e31.748
999998                   u42.091                    x57.034
```

```
999999                    c45.844                    j64.549

       patient_diagnosis_code_248 patient_diagnosis_code_249  \
0                          w42.14                    q51.264
1                         r45.241                    s52.367
2                         x54.168                    q66.029
3                         f78.392                    g65.868
4                         r33.531                    y44.278
…                             …                          …
999995                     x49.53                    i39.681
999996                    d71.307                    s22.208
999997                    k23.645                    v44.286
999998                    a39.008                    v68.961
999999                    a42.681                    s72.205

       patient_diagnosis_code_250
0                         c87.897
1                          b45.89
2                         w29.481
3                         b15.069
4                         v29.581
…                             …
999995                    q29.383
999996                    b73.302
999997                    q29.383
999998                    z46.091
999999                    v33.308

[1000000 rows x 251 columns]
```

[2490]:
```
# # If error "TypeError: 'str' object is not callable" rerun `def␣
 ↪create_patient_procedure_info()` cell
# num_synthetic_rows_big = 1000000
# num_unique_patients_big = 1000000
# num_patient_attributes_big = 20

# patient_procedure_info =␣
 ↪create_patient_procedure_info(num_synthetic_rows_big,␣
 ↪num_unique_patients_big, num_patient_attributes_big)
# # patient_procedure_info
```

[31]:
```
patient_procedure_info
```

[31]:
```
       patient_id  patient_attribute_1  patient_attribute_2  \
0         085438o            266701801            561141537
1         562093s            257540053            799284514
2          09083l            128531780            341899964
```

```
3         469329e            625682632            846781971
4         241375i            496790665            415149861
…           …                  …                  …
999995    457818a            448405486            773124192
999996    478256b            513481307            459905329
999997    106251l            107169551            475643838
999998    145729n            219726900            190785745
999999    287440q            359699891            158424833

        patient_attribute_3  patient_attribute_4  patient_attribute_5  \
0                 378523569            524778076            207383775
1                 971214051            507050318            590687290
2                 332497794            111818856            539132348
3                 708727431            331026534            477791945
4                 338282099            487025965            303664991
…                       …                    …                    …
999995             89902046            129856610            671490875
999996            444475509            111945753            307558014
999997             86429536            127007358            158791959
999998            571844326            238197030            352775476
999999             95100225            384017845            306876946

        patient_attribute_6  patient_attribute_7  patient_attribute_8  \
0                 135923455            158434084            443757160
1                 505463649            574386954            126906730
2                 388811485             97339458            165910260
3                 747924904            259059225            595868602
4                 675968280            279647969            102401375
…                       …                    …                    …
999995            595064117             19744093            339796010
999996              4713362            186216781            400697014
999997            262414028            561858247            513230727
999998            375995606            254361870            342905964
999999            467544512            181464886            259997143

        patient_attribute_9  …  patient_attribute_11  patient_attribute_12  \
0                  98863970  …             330811590             611322247
1                 412860355  …             537746710             378552970
2                 259923355  …             284613042             136198493
3                 272925734  …             141196005             413983441
4                 157512903  …             179384263             377731329
…                       …  …                     …                     …
999995            510488949  …             136685279              43900292
999996            466325402  …             270632537             516888049
999997             48469671  …             771389669             307309415
999998            131734180  …             401431068             220203655
999999            150209037  …             348033848             144550128
```

```
        patient_attribute_13  patient_attribute_14  patient_attribute_15  \
0                   61499057             397184884             212031962
1                  103651209             351199007             130008527
2                  541516146             407810817             359216133
3                  703146817             218908022              68227461
4                  284428341             130088202             184635896
…                        …                     …                     …
999995              81186371             179586237             555898268
999996             731644684             636225868             259721529
999997             124325628             159673740             128915683
999998             432804398             360871262             430066513
999999              87388368             351675390             411057156

        patient_attribute_16  patient_attribute_17  patient_attribute_18  \
0                  697284928             364199575             165814960
1                  478815157             854743050             685365284
2                  112158815             480717921             483253666
3                  534328355             375256160             288233730
4                  224827083             412025904             166845718
…                        …                     …                     …
999995             345967774             509428344             510689588
999996             478798968             898451407             669088655
999997             568913611             563919140             556488459
999998             666294802             413290644             291445205
999999             807172308             301883157             288408952

        patient_attribute_19  patient_attribute_20
0                  445470105             979862284
1                  479612880             822190852
2                  276635212             105019323
3                  211544097             312594472
4                  715667180             219829158
…                        …                     …
999995             169227150             386827780
999996              82911786             184128686
999997             109725301             743427052
999998             427057004             364637295
999999             165644045             151989854

[1000000 rows x 21 columns]
```

[32]:
```
# Save Dataframe to reduce time to resynthesize
# csv_dataset_diagnosis_patient_procedure_info = Path(str(Path.cwd()) + '/' +␣
 ↪'sqlDataModelingData' + '/' + 'patient_procedure_info_big.csv')
# patient_procedure_info.to_csv(csv_dataset_diagnosis_patient_procedure_info,␣
 ↪index=False)
```

[33]:
```
# # If error "TypeError: 'str' object is not callable" rerun `def␣
 ↪create_visit_log()` cell
# num_synthetic_rows = 1000000
# num_unique_patients = 25000
# num_unique_physicians = 50
# num_physician_attributes = 20

# visit_log = create_visit_log(num_synthetic_rows, num_unique_patients,␣
 ↪num_unique_physicians, num_physician_attributes)
# # visit_log
```

[34]:
```
visit_log
```

[34]:

	visit_id	patient_id	physician_id	physician_attribute_1	\
0	571556	126631r	077122z	213339669	
1	981861	106769h	351313g	443309582	
2	662	896171a	677717o	652303571	
3	344121	447684m	252781h	'057510660'	
4	398692	353001w	403117s	116712546	
…	…	…	…	…	
999995	738995	647992r	310622c	416972191	
999996	337313	257182a	526187n	381011306	
999997	27706	249614a	526187n	381011306	
999998	68701	140025c	200862j	257013399	
999999	41823	690954a	142429c	342713798	

	physician_attribute_2	physician_attribute_3	physician_attribute_4	\
0	361069306	260484338	195196675	
1	102701953	186063455	470156746	
2	477503269	726171566	'085831619'	
3	100126810	319617934	595903833	
4	517263535	559781016	863954446	
…	…	…	…	
999995	'088827951'	152620479	270189206	
999996	390204240	653045895	172459250	
999997	390204240	653045895	172459250	
999998	123366508	581515859	570900644	
999999	577039116	161964311	210553540	

	physician_attribute_5	physician_attribute_6	physician_attribute_7	…	\
0	'073133483'	217356313	738461107	…	
1	'077609962'	518671919	456230491	…	
2	416260535	778852452	496072780	…	
3	360145090	452124871	181340822	…	
4	382515049	193478753	500437280	…	
…	…	…	…	…	
999995	124652681	240625172	217286931	…	

```
999996             150281607             329401888             220679493  …
999997             150281607             329401888             220679493  …
999998             440257745             231251581             311800828  …
999999             208511361             106707287             567700697  …

       physician_attribute_15 physician_attribute_16 physician_attribute_17  \
0                   473017335              168666608              625430745
1                   423373300              454297925              686406426
2                  '026070609'             878376300              410674573
3                  '036285290'             806925155              573617610
4                   175005975              871049202              428207054
…                         …                      …                      …
999995              352683757              313220904              228512226
999996              364381852              737423120              731366381
999997              364381852              737423120              731366381
999998              284074619              174856004              425764530
999999             '098225186'             150971517             '067983110'

       physician_attribute_18 physician_attribute_19 physician_attribute_20  \
0                   414719468              509141525             '088305540'
1                   194262959              745893766               114997030
2                  '084685415'             171528679               137514323
3                   418924497             '073210818'              136867142
4                   183893258              715637372               476920243
…                         …                      …                      …
999995              209698269              514313229              344955137
999996              922761201              218618340              670182000
999997              922761201              218618340              670182000
999998              813192033              117773466              218757938
999999              777147386              193467145              270534266

        visit_date procedure_code utilization_quantity expenditure_amount
0       2019-02-05      129548318                   12     $4279229475.82
1       2018-10-24      615335837                   12     $7733399417.75
2       2021-08-06      165988850                   44     $3686078418.63
3       2019-02-14      502580089                   16     $8624388702.64
4       2016-09-25      283256179                   39     $7618742576.66
…              …              …                    …                  …
999995  2017-06-22      275529688                   14     $3200744465.59
999996  2019-10-15      669454843                   14     $4883886315.06
999997  2017-04-26      253374846                   75     $3623515575.41
999998  2021-09-27      194211324                   13     $1767061234.31
999999  2021-03-23      102707926                   33     $1952614612.35

[1000000 rows x 27 columns]
```

[35]:
```
# Save Dataframe to reduce time to resynthesize
# csv_dataset_visit_log_info_big = Path(str(Path.cwd()) + '/' +␣
 ↪'sqlDataModelingData' + '/' + 'visit_log_info_big.csv')
# visit_log.to_csv(csv_dataset_visit_log_info_big, index=False)
```

Join Tables

[36]:
```
# patient_id's were synthesized on a per table basis, in the interest of time I␣
 ↪will update the tables synthesizing at a later time. I have increased the␣
 ↪sample size as a quick fix for visualizing the concept
# of manipulating the data.
# We have selected to do an inner join because I am assuming we will want to␣
 ↪drop all null values after any join anyways, if we are to use these columns␣
 ↪for feature selection for a DL/ML model.
temp_table = pd.merge(patient_procedure_info, diagnosis_info, how='inner',␣
 ↪on='patient_id')
temp_table
```

[36]:

	patient_id	patient_attribute_1	patient_attribute_2	\
0	397058b	274981585	210972870	
1	191364x	83877215	684831920	
2	045665c	96826681	264583161	
3	178670i	756235105	198557569	
4	606951v	390073676	130276993	
…	…	…	…	
51378	063361n	347233166	617768986	
51379	117709q	804605903	409950747	
51380	366102p	163611643	202751167	
51381	628814z	429764401	400584187	
51382	141882l	235530947	405710022	

	patient_attribute_3	patient_attribute_4	patient_attribute_5	\
0	392036160	263766783	123270201	
1	549787476	62892538	79306802	
2	270977051	215425721	826746753	
3	330374472	164570002	117083590	
4	148784979	810264752	383595104	
…	…	…	…	
51378	188537886	338719724	244567289	
51379	600792460	611938806	166553566	
51380	798204304	326218438	218682901	
51381	182140176	338579258	369630897	
51382	760767499	544271069	549710282	

	patient_attribute_6	patient_attribute_7	patient_attribute_8	\
0	284939629	390371704	911663031	
1	726250742	395155872	839243486	
2	334999978	398976658	34173973	

```
3                497377383            318067493            214752906
4                176713079             48849052            591394801
…                    …                    …                    …
51378            613352007            407860377            633069328
51379            398511771            442206487            253577341
51380             48696998            180281850            145323398
51381            382430587            706685911            813405326
51382            358036520            334274155            566162173

       patient_attribute_9  …  patient_diagnosis_code_241  \
0                657428568  …                      j29.039
1                540777411  …                      a35.666
2                307954880  …                      t50.978
3                390188408  …                      v85.515
4                253726046  …                      v37.934
…                    …  …                          …
51378            149589278  …                      a50.261
51379            125423561  …                      r12.001
51380            147465686  …                      y63.197
51381            222663429  …                      j49.602
51382            208232425  …                      k75.215

       patient_diagnosis_code_242  patient_diagnosis_code_243  \
0                          c14.008                      m67.009
1                          j54.242                      o39.345
2                          p53.745                      h54.798
3                          y57.612                      h49.113
4                          w41.071                      o33.123
…                            …                          …
51378                      y53.205                      r41.524
51379                      m56.212                      m20.558
51380                      d89.445                       m40.11
51381                      h67.111                      l47.103
51382                      v34.489                      i33.972

       patient_diagnosis_code_244  patient_diagnosis_code_245  \
0                          t74.378                      i37.787
1                          c81.724                      m53.575
2                          f40.237                      j59.166
3                          c75.481                      l63.209
4                           q59.07                      r43.346
…                            …                          …
51378                       w77.99                      n29.553
51379                       n47.47                      p04.675
51380                      l66.308                      w43.862
51381                      h45.466                      f69.719
51382                      t74.378                      f58.908
```

```
       patient_diagnosis_code_246  patient_diagnosis_code_247  \
0                          f64.921                     y63.448
1                          b26.939                     a64.954
2                          b53.217                      a63.26
3                          b23.898                     s43.305
4                          c96.426                     c83.234
…                              …                           …
51378                      c08.676                     y63.448
51379                      o08.365                     o48.517
51380                      u31.947                       a43.2
51381                      f34.579                     i64.223
51382                      i56.458                     d64.996

       patient_diagnosis_code_248  patient_diagnosis_code_249  \
0                          y54.579                     b58.382
1                          a55.079                     v68.961
2                          l40.871                     s08.546
3                           i71.42                     h47.255
4                          a38.639                     z59.919
…                              …                           …
51378                      p15.565                     s53.818
51379                      a23.686                     v71.845
51380                      c62.994                      s82.43
51381                      l52.643                      s82.43
51382                      k23.645                     b11.712

       patient_diagnosis_code_250
0                          r12.053
1                          q29.383
2                          b49.754
3                          v21.362
4                          b47.778
…                              …
51378                      v33.308
51379                      c33.556
51380                      r55.896
51381                      r27.502
51382                      q29.383

[51383 rows x 271 columns]
```

[ ]:
```
# Another way to merge the tables with the same results.
```

[ ]:
```
# final_table = pd.merge(visit_log, temp_table, how='inner', on='patient_id')
# final_table
```

[ ]:
```
# the_list =  final_table[['visit_id', 'patient_id','visit_date']].
 ↪groupby('visit_id').max().sort_values(['patient_id', 'visit_date',␣
 ↪'visit_id'], ascending=False).reset_index().drop_duplicates('patient_id')
# the_list
```

[37]:
```
# temp_table_2 = pd.merge(the_list, final_table, how='inner', on=['visit_id',␣
 ↪'patient_id'])
# temp_table_2
```

[38]:
```
the_list =  visit_log[['visit_id', 'patient_id','visit_date']].
 ↪groupby('visit_id').max().sort_values(['patient_id', 'visit_date',␣
 ↪'visit_id'], ascending=False).reset_index().drop_duplicates('patient_id')
the_list
```

[38]:
```
        visit_id patient_id  visit_date
0         438268    995397t  2022-01-12
173       188881    989301i  2021-12-18
182        12209    988280z  2021-08-07
198       209879    986722m  2022-01-19
251       412459    985906u  2021-09-10
…            …         …          …
567586    552631    011365j  2022-01-14
567603    902540    011301x  2020-02-27
567611    462796    010539s  2021-11-04
567628    577634    010274b  2021-04-11
567632    439140    010105c  2020-01-21

[24414 rows x 3 columns]
```

[39]:
```
temp_table_2 = pd.merge(the_list, visit_log, how='inner', on=['visit_id',␣
 ↪'patient_id', 'visit_date'])
temp_table_2
```

[39]:
```
        visit_id patient_id  visit_date physician_id physician_attribute_1  \
0         438268    995397t  2022-01-12      947215u              125371005
1         209879    986722m  2022-01-19      241455q              349467935
2         228224    978627w  2021-12-23      142429c              342713798
3          91578    974877p  2022-01-10      466252i              778288811
4         262362    970314e  2022-01-18      122643g              321871413
…            …         …          …           …                      …
16336     552631    011365j  2022-01-14      336236x              672652337
16337     902540    011301x  2020-02-27      270665t            '034167386'
16338     462796    010539s  2021-11-04      518043b            '088656953'
16339     577634    010274b  2021-04-11      225484f              565789468
16340     439140    010105c  2020-01-21      317440m              417430100

       physician_attribute_2 physician_attribute_3 physician_attribute_4  \
```

```
0                642696595             701036218             121786055
1                417485295             297515151             132638169
2                577039116             161964311             210553540
3                282809148             885640462           '056433527'
4                643122189             360827320             610481295
…                    …                     …                     …
16336            238109558             318237483             623418120
16337            291810804             315591277             574879429
16338            311599633             116844824             336111541
16339            734712900             484598592             268882732
16340            569284426             380225165             327139641

      physician_attribute_5 physician_attribute_6  … physician_attribute_14  \
0                335339923             133842553  …              239149762
1                697947068             498635911  …              306845371
2                208511361             106707287  …              644775978
3                119243199             351496452  …              294442135
4                662258915             577492616  …              502684729
…                    …                     …  …                      …
16336            805718649             295719272  …              226486213
16337          '068229243'             491374778  …              514051485
16338            290385337             327434085  …              292074215
16339          '029135013'             272712316  …              260326493
16340            206453553           '071726372'  …              592237565

      physician_attribute_15 physician_attribute_16 physician_attribute_17  \
0                '037348141'              534732542             224054089
1                  463312998              230213596             151194454
2                '098225186'              150971517           '067983110'
3                  206679700              476321598             200741812
4                  486669481              479564611             418339586
…                      …                      …                     …
16336              140165735              236737287             700216246
16337              359674720              659054335             645664800
16338              422109364              703053783             113696969
16339              584788034            '061462880'             132861880
16340              403517614              485196506             550133711

      physician_attribute_18 physician_attribute_19 physician_attribute_20  \
0                621814388            '070350322'             555407763
1                608852682              428752727             387903063
2                777147386              193467145             270534266
3              '077376573'              338424059             184951678
4                440546172              229116947             290957771
…                    …                      …                     …
16336            192655661              265505216           '070620524'
16337            180517543              371724500             736406351
```

```
16338              728187522              718822394              205305821
16339              702133806              593580193              488650001
16340              309526918              223997834              407985202

      procedure_code utilization_quantity expenditure_amount
0          267566490                   61     $6461678881.95
1          744877542                   63     $5577937419.70
2          227209317                   63     $7098451654.77
3          617407886                   35     $5410349450.68
4          335307018                   31     $6644506056.70
…                …                    …                  …
16336      279162700                   51     $6362018975.44
16337       45744478                   49     $4495635800.28
16338       84770168                   17     $5215096263.96
16339       57861148                   41     $2867173573.09
16340      330632378                   48     $5805334396.44

[16341 rows x 27 columns]
```

[40]:
```
final_table = pd.merge(temp_table, temp_table_2, how='inner', on='patient_id')
final_table
```

[40]:
```
   patient_id  patient_attribute_1  patient_attribute_2  patient_attribute_3  \
0      235144e            491820920            351716859            758305368
1      187187s            233023006            462720672            433154671
2      122982u            407832513            763190710            864323922
3      086562c            259178398            561473643            338673896
4      371775q            222551832            112960909            278848385
5      154295h            322517676             55362525            213855271
6      161369l            239114831            168922969            167307877
7      069605h             62902948            324673404            509055121
8      086941b            534836017            636937192            490083050
9      086941b            534836017            636937192            490083050
10     771591f            435717761            154084860            465392799
11     274822m             85950052            275062467            608262874
12     197734d            298750914            251838286            694036133
13     421256v            668833752            397549477            357459520
14     265030w             98220911            310872059            577601188
15     133365w            250838928            120254706            317507207
16     133365w            490745762            572473663            337606046
17     124973s            566205548            293142485            921996585
18     545820s            662470974            320319259            664974275
19     171358m            129022222            487458482            144172706
20     335935t            664448244             50357716            120261965
21     138278z            325417670            133662547            211211441
22     220476g            897357181            660363133            255201638
23     258675w            288518063            583469064             27540594
```

24	455764k	464007226	181668298	73839675
25	108905b	308206143	588334324	227265748
26	094428c	507350135	262111125	248528464
27	284788i	886085482	82253935	282779008
28	550045a	338387952	214761866	641319472
29	238236g	433656636	115161923	211786561
30	291577j	199083744	463736819	631238551
31	147514d	134621430	179873788	199229985
32	126599a	104829867	245442472	136541003
33	101791i	200435082	524798204	163551751
34	239611w	392581251	236830354	169029599
35	089052o	283646953	737061155	619611009
36	090205t	182785974	353002953	748701742
37	200571m	470624302	156280161	70408800
38	078611b	581119973	182090145	765892853
39	074991o	795057104	295778707	867129202
40	378564w	350998172	501421949	470412767
41	185704g	471358882	24289522	819842201
42	273123e	521973859	424271755	861841710
43	182148a	106319175	155324983	333398876
44	276029e	616721516	199686482	392388174
45	152790h	510305177	800362489	120536797
46	734281h	705509513	491297803	248424599
47	338968d	456487594	339731385	328177243
48	102901x	121224550	857179427	202094278
49	135325b	320950842	207208941	409175861
50	210418a	320410274	802066797	212197784
51	198645z	541330130	335463850	400848945
52	236718n	221321617	226059305	749069407
53	092571c	109037704	372495562	189695228
54	267912c	411830763	381297223	512758639
55	416229h	78654066	571092963	231648814
56	230992j	296563583	546577810	199349869

	patient_attribute_4	patient_attribute_5	patient_attribute_6	\
0	282929741	359098303	321947746	
1	24595839	210452757	854387490	
2	180148318	20550896	264519029	
3	71929468	88413455	297796224	
4	313895747	110759676	553755250	
5	418478974	356214927	360710548	
6	173053377	463935240	221401206	
7	498137303	802904048	224069137	
8	562775838	684177826	684067136	
9	562775838	684177826	684067136	
10	449285016	291257964	811264366	
11	255101567	144842219	751794912	

12	329997878	331089126	436488443
13	206960421	164841331	576771152
14	791108525	869733992	671702743
15	177286799	628067723	366751999
16	87075643	633032348	241343638
17	443055542	304410758	262200495
18	216846135	130606377	50269321
19	303799501	715969850	546782838
20	129677193	385272215	875850313
21	607753344	217235323	902585177
22	307994431	441098530	303986170
23	474788761	402863910	142911773
24	133972714	208121873	976040967
25	84819004	500085669	42344601
26	394357647	468437737	111888850
27	210747611	288252285	174671847
28	357948419	63968113	88986520
29	183716733	238494910	716599450
30	135086866	123274954	433105540
31	38114816	287285134	984481025
32	257481277	395294004	270335521
33	447291308	162477166	374851743
34	93321060	214544347	99434859
35	109501094	553861085	168379092
36	300161207	47590773	578703834
37	430141888	92652817	519318716
38	419864323	789633352	184882215
39	358494092	464053933	56224861
40	225162512	271683146	490183622
41	558955328	548469345	122324141
42	562471722	650652576	222771357
43	237443816	206489912	277149257
44	265090308	287259012	589791980
45	338512297	350418000	293364846
46	194734610	238269927	346540075
47	239223918	390521193	131798413
48	343556868	462048611	160524838
49	394185012	501698966	200105019
50	310604685	315995784	458761181
51	152388420	345320294	119210142
52	691389724	381424491	287208019
53	564761069	598877941	260798744
54	485856575	244603065	44811655
55	893974881	84406539	131357407
56	184409660	741399176	859474305

```
    patient_attribute_7  patient_attribute_8  patient_attribute_9  …  \
```

0	280811044	800603368	225702331	…
1	58894591	203142322	132887613	…
2	672980714	782714256	122328371	…
3	942614837	762110980	232209084	…
4	262577549	526716343	750799553	…
5	634876483	181342422	509642175	…
6	57768114	172000780	217381567	…
7	356229855	718104378	377733883	…
8	614289553	551085458	146861445	…
9	614289553	551085458	146861445	…
10	144848316	824650461	874316081	…
11	333796831	405676847	408072791	…
12	226781765	335576948	431518852	…
13	140044277	365903075	625730550	…
14	87642023	458952904	723012917	…
15	112960118	73249000	544822005	…
16	476687853	269518941	52516489	…
17	86632318	181767310	659380225	…
18	54004267	520069861	116585607	…
19	113084249	319128598	690581580	…
20	675411241	468543403	604372748	…
21	75856910	270204379	214210264	…
22	133780673	88910753	409969593	…
23	145064464	152594281	131728720	…
24	349654134	232777085	628819621	…
25	251467881	195539018	317319304	…
26	142680350	356958893	513080135	…
27	461100936	210028907	421226860	…
28	522205689	596866578	299204654	…
29	578816269	779011345	459014760	…
30	239700949	535661600	242159492	…
31	295093847	515477054	657973450	…
32	645942725	655682017	42599360	…
33	572322537	899164819	153580838	…
34	460851056	516204194	391574201	…
35	773308317	171553474	717028488	…
36	107483893	424798327	361053526	…
37	192506344	427285211	516577139	…
38	469384553	307053148	139628823	…
39	299375115	246001120	273830206	…
40	753047789	347307398	48635849	…
41	488074059	695689571	733569645	…
42	44666549	98906488	963292720	…
43	577651908	518535818	376363235	…
44	335809590	707195895	329810426	…
45	344380311	460914119	347209756	…
46	330474180	200046479	435122077	…

```
47            817284672            384041995            500631483  …
48            334210941            121632684            186725308  …
49            547310039             94760984            799086416  …
50            321250171            316020564            764648083  …
51            342093041            377117787            141576588  …
52            132982448            373015896            370554207  …
53            795212512            253171455            274740545  …
54            103053105            366916619            547002984  …
55            230885292            474492672            324026883  …
56            273349401             87380294            695232018  …

    physician_attribute_14  physician_attribute_15  physician_attribute_16  \
0                592237565               403517614               485196506
1                366488910               352683757               313220904
2                306845371               463312998               230213596
3                226486213               140165735               236737287
4                292074215               422109364               703053783
5                234973919               264983325               138603617
6              '090150635'               281710128               189043109
7                167284597               284074619               174856004
8                137088705               514359687               438052595
9                137088705               514359687               438052595
10               260326493               584788034             '061462880'
11               894445928             '026070609'               878376300
12               514051485               359674720               659054335
13               120462051               592709477               355998426
14             '061015097'               170643473               286507098
15               366488910               352683757               313220904
16               366488910               352683757               313220904
17               502684729               486669481               479564611
18               329706746               214771007               776675340
19               329706746               214771007               776675340
20               206103306               498468959               230710865
21               239149762             '037348141'               534732542
22               226486213               140165735               236737287
23               239149762             '037348141'               534732542
24               129640395               225322451               225115555
25               717050131               541913102               297662975
26               644775978             '098225186'               150971517
27               226486213               140165735               236737287
28               226486213               140165735               236737287
29               366488910               352683757               313220904
30               120462051               592709477               355998426
31               378679039               667586082               328266116
32               398207051               528380985               781149794
33               378679039               667586082               328266116
34             '061015097'               170643473               286507098
```

35	206103306	498468959	230710865
36	167284597	284074619	174856004
37	275128080	623099901	356268220
38	502684729	486669481	479564611
39	448849565	183734749	494170288
40	760093683	468894298	281494499
41	234973919	264983325	138603617
42	894445928	'026070609'	878376300
43	378679039	667586082	328266116
44	644775978	'098225186'	150971517
45	109974083	867850122	592203895
46	'090150635'	281710128	189043109
47	644775978	'098225186'	150971517
48	514051485	359674720	659054335
49	366488910	352683757	313220904
50	137088705	514359687	438052595
51	168277406	156731802	129039318
52	448849565	183734749	494170288
53	292074215	422109364	703053783
54	123009837	'027165272'	'088011226'
55	129640395	225322451	225115555
56	167284597	284074619	174856004

	physician_attribute_17	physician_attribute_18	physician_attribute_19	\
0	550133711	309526918	223997834	
1	228512226	209698269	514313229	
2	151194454	608852682	428752727	
3	700216246	192655661	265505216	
4	113696969	728187522	718822394	
5	131083046	208963349	907072186	
6	454357906	105903980	563698439	
7	425764530	813192033	117773466	
8	449448065	'097042681'	584744600	
9	449448065	'097042681'	584744600	
10	132861880	702133806	593580193	
11	410674573	'084685415'	171528679	
12	645664800	180517543	371724500	
13	234333792	254929558	258713224	
14	'070308940'	616592727	446524025	
15	228512226	209698269	514313229	
16	228512226	209698269	514313229	
17	418339586	440546172	229116947	
18	114197339	469250963	117331672	
19	114197339	469250963	117331672	
20	460436772	396914090	301452797	
21	224054089	621814388	'070350322'	
22	700216246	192655661	265505216	

23	224054089	621814388	'070350322'
24	589140742	581041072	847153394
25	146797976	477042533	218918601
26	'067983110'	777147386	193467145
27	700216246	192655661	265505216
28	700216246	192655661	265505216
29	228512226	209698269	514313229
30	234333792	254929558	258713224
31	618355794	540446893	'095956073'
32	183014477	115196124	582253720
33	618355794	540446893	'095956073'
34	'070308940'	616592727	446524025
35	460436772	396914090	301452797
36	425764530	813192033	117773466
37	268595276	653635552	239403120
38	418339586	440546172	229116947
39	109056137	198457132	672209110
40	584060801	677555570	514787734
41	131083046	208963349	907072186
42	410674573	'084685415'	171528679
43	618355794	540446893	'095956073'
44	'067983110'	777147386	193467145
45	731229962	175608161	617488583
46	454357906	105903980	563698439
47	'067983110'	777147386	193467145
48	645664800	180517543	371724500
49	228512226	209698269	514313229
50	449448065	'097042681'	584744600
51	120552610	446412871	175419320
52	109056137	198457132	672209110
53	113696969	728187522	718822394
54	911043096	'075512512'	556943761
55	589140742	581041072	847153394
56	425764530	813192033	117773466

	physician_attribute_20	procedure_code	utilization_quantity	\
0	407985202	506939460	25	
1	344955137	624064682	27	
2	387903063	258721052	21	
3	'070620524'	525246749	32	
4	205305821	709326627	15	
5	114504390	660277225	16	
6	556516249	243571702	15	
7	218757938	190471766	18	
8	196603799	871649368	12	
9	196603799	871649368	12	
10	488650001	818805997	49	

11	137514323	215108633	35
12	736406351	168828843	54
13	299791337	225018538	47
14	490809134	514790453	74
15	344955137	110941404	45
16	344955137	110941404	45
17	290957771	123553175	49
18	250683503	505919477	26
19	250683503	186174947	13
20	383763804	509246779	14
21	555407763	705328042	66
22	'070620524'	109750285	55
23	555407763	251416010	40
24	284905869	611085157	71
25	'090650226'	412071355	60
26	270534266	508011328	28
27	'070620524'	119245111	45
28	'070620524'	452881668	63
29	344955137	456045948	63
30	299791337	132564262	7
31	375172832	606082145	5
32	560351611	120794504	28
33	375172832	917113098	31
34	490809134	96567247	65
35	383763804	525015568	9
36	218757938	448821274	18
37	127993463	391419255	40
38	290957771	275628236	9
39	178034600	502878727	35
40	202044964	784757646	35
41	114504390	104573011	10
42	137514323	653461700	40
43	375172832	148210430	24
44	270534266	462973896	95
45	156022748	56792080	7
46	556516249	220954176	61
47	270534266	379463711	63
48	736406351	346936684	48
49	344955137	197543270	3
50	196603799	561360842	18
51	739871361	212500678	15
52	178034600	620556273	29
53	205305821	385878079	26
54	'060566421'	298342936	10
55	284905869	491007969	12
56	218757938	836948478	66

	expenditure_amount
0	$2558842390.99
1	$4806229377.46
2	$6290304098.84
3	$5995269511.33
4	$6174218093.22
5	$6202288864.12
6	$6163872613.74
7	$7902110422.60
8	$3441749593.52
9	$3441749593.52
10	$3679749970.36
11	$5985986160.98
12	$5258633579.72
13	$3830178525.03
14	$8915193240.07
15	$2030395841.65
16	$2030395841.65
17	$7116572485.18
18	$3718288041.26
19	$8405725600.62
20	$5090366707.07
21	$7509134702.11
22	$6679169406.62
23	$6553260338.66
24	$5081575027.82
25	$3695662077.39
26	$2700335307.08
27	$7418334830.96
28	$5839326672.03
29	$0967362855.63
30	$7000623684.11
31	$2643683171.52
32	$0629686347.08
33	$7940050009.81
34	$5656443183.12
35	$5668960061.06
36	$5820371920.39
37	$3734538162.78
38	$2534170100.18
39	$4995551253.40
40	$4785284010.44
41	$6288862303.05
42	$5233087861.61
43	$1856336073.19
44	$2889684409.82
45	$4107868655.53

```
46      $7004692962.20
47      $2583257749.76
48      $5463285931.01
49      $3721300880.41
50      $7290114256.10
51      $5190827878.61
52      $4347125756.20
53      $2271063052.55
54      $8469425457.73
55      $5874188734.29
56      $4252624796.51

[57 rows x 297 columns]
```

[41]:
```
final_table.dtypes.value_counts()
```

[41]:
```
object    274
int64      23
dtype: int64
```

[42]:
```
final_table.astype('string').dtypes
```

[42]:
```
patient_id                string
patient_attribute_1       string
patient_attribute_2       string
patient_attribute_3       string
patient_attribute_4       string
                           …
physician_attribute_19    string
physician_attribute_20    string
procedure_code            string
utilization_quantity      string
expenditure_amount        string
Length: 297, dtype: object
```

[43]:
```
final_table = final_table.convert_dtypes()
final_table.dtypes.value_counts()
```

[43]:
```
string    274
Int64      23
dtype: int64
```

[44]:
```
final_table.index = pd.to_datetime(final_table['visit_date'])
final_table.drop(columns='visit_date', inplace=True)
```

[50]:
```
final_table.head()
```

[50]:

visit_date	patient_id	patient_attribute_1	patient_attribute_2	\
2020-05-24	235144e	491820920	351716859	
2021-06-08	187187s	233023006	462720672	
2021-10-29	122982u	407832513	763190710	
2021-06-20	086562c	259178398	561473643	
2018-07-30	371775q	222551832	112960909	

visit_date	patient_attribute_3	patient_attribute_4	patient_attribute_5	\
2020-05-24	758305368	282929741	359098303	
2021-06-08	433154671	24595839	210452757	
2021-10-29	864323922	180148318	20550896	
2021-06-20	338673896	71929468	88413455	
2018-07-30	278848385	313895747	110759676	

visit_date	patient_attribute_6	patient_attribute_7	patient_attribute_8	\
2020-05-24	321947746	280811044	800603368	
2021-06-08	854387490	58894591	203142322	
2021-10-29	264519029	672980714	782714256	
2021-06-20	297796224	942614837	762110980	
2018-07-30	553755250	262577549	526716343	

visit_date	patient_attribute_9	…	physician_attribute_14	\
2020-05-24	225702331	…	592237565	
2021-06-08	132887613	…	366488910	
2021-10-29	122328371	…	306845371	
2021-06-20	232209084	…	226486213	
2018-07-30	750799553	…	292074215	

visit_date	physician_attribute_15	physician_attribute_16	\
2020-05-24	403517614	485196506	
2021-06-08	352683757	313220904	
2021-10-29	463312998	230213596	
2021-06-20	140165735	236737287	
2018-07-30	422109364	703053783	

visit_date	physician_attribute_17	physician_attribute_18	\
2020-05-24	550133711	309526918	
2021-06-08	228512226	209698269	
2021-10-29	151194454	608852682	
2021-06-20	700216246	192655661	
2018-07-30	113696969	728187522	

```
            physician_attribute_19  physician_attribute_20  procedure_code  \
visit_date
2020-05-24               223997834               407985202       506939460
2021-06-08               514313229               344955137       624064682
2021-10-29               428752727               387903063       258721052
2021-06-20               265505216             '070620524'       525246749
2018-07-30               718822394               205305821       709326627

            utilization_quantity  expenditure_amount
visit_date
2020-05-24                    25      $2558842390.99
2021-06-08                    27      $4806229377.46
2021-10-29                    21      $6290304098.84
2021-06-20                    32      $5995269511.33
2018-07-30                    15      $6174218093.22

[5 rows x 296 columns]
```

[53]:
```
final_table.columns
# final_table.columns.to_list()
```

[53]:
```
Index(['patient_id', 'patient_attribute_1', 'patient_attribute_2',
       'patient_attribute_3', 'patient_attribute_4', 'patient_attribute_5',
       'patient_attribute_6', 'patient_attribute_7', 'patient_attribute_8',
       'patient_attribute_9',
       …
       'physician_attribute_14', 'physician_attribute_15',
       'physician_attribute_16', 'physician_attribute_17',
       'physician_attribute_18', 'physician_attribute_19',
       'physician_attribute_20', 'procedure_code', 'utilization_quantity',
       'expenditure_amount'],
      dtype='object', length=296)
```

2.0.1 Label Encoding

[54]:
```
final_table.describe()
```

[54]:
```
       patient_attribute_1  patient_attribute_2  patient_attribute_3  \
count         5.700000e+01         5.700000e+01         5.700000e+01
mean          3.792033e+08         3.638740e+08         4.032567e+08
std           2.077522e+08         2.148733e+08         2.440882e+08
min           6.290295e+07         2.428952e+07         2.754059e+07
25%           2.225518e+08         1.820901e+08         2.112114e+08
50%           3.383880e+08         3.246734e+08         3.376060e+08
75%           5.103052e+08         5.247982e+08         6.082629e+08
max           8.973572e+08         8.571794e+08         9.219966e+08
```

```
       patient_attribute_4  patient_attribute_5  patient_attribute_6  \
count         5.700000e+01         5.700000e+01         5.700000e+01
mean          3.239858e+08         3.676943e+08         3.891903e+08
std           1.887383e+08         2.146219e+08         2.675021e+08
min           2.459584e+07         2.055090e+07         4.234460e+07
25%           1.837167e+08         2.104528e+08         1.746718e+08
50%           3.037995e+08         3.453203e+08         2.977962e+08
75%           4.430555e+08         5.000857e+08         5.767712e+08
max           8.939749e+08         8.697340e+08         9.844810e+08

       patient_attribute_7  patient_attribute_8  patient_attribute_9  \
count         5.700000e+01         5.700000e+01         5.700000e+01
mean          3.585186e+08         4.022958e+08         4.091601e+08
std           2.346551e+08         2.185262e+08         2.343449e+08
min           4.466655e+07         7.324900e+07         4.259936e+07
25%           1.426804e+08         2.100289e+08         2.173816e+08
50%           3.304742e+08         3.730159e+08         3.777339e+08
75%           5.473100e+08         5.267163e+08         6.043727e+08
max           9.426148e+08         8.991648e+08         9.632927e+08

       patient_attribute_10  …  patient_attribute_14  patient_attribute_15  \
count          5.700000e+01  …          5.700000e+01          5.700000e+01
mean           3.651189e+08  …          3.829493e+08          3.978653e+08
std            2.431674e+08  …          2.241322e+08          2.212308e+08
min            4.436010e+07  …          7.424989e+06          6.536827e+07
25%            1.487695e+08  …          2.195811e+08          2.076032e+08
50%            3.192423e+08  …          3.689281e+08          3.777405e+08
75%            5.591636e+08  …          5.040083e+08          5.301183e+08
max            9.072941e+08  …          8.946992e+08          9.087290e+08

       patient_attribute_16  patient_attribute_17  patient_attribute_18  \
count          5.700000e+01          5.700000e+01          5.700000e+01
mean           3.341590e+08          3.869333e+08          4.139674e+08
std            2.439584e+08          2.323402e+08          2.408460e+08
min            2.821819e+07          1.441755e+07          8.223379e+07
25%            1.370289e+08          2.249093e+08          1.847429e+08
50%            2.472063e+08          3.252862e+08          4.124519e+08
75%            4.433485e+08          5.010221e+08          6.188832e+08
max            9.457811e+08          9.633612e+08          8.765670e+08

       patient_attribute_19  patient_attribute_20       visit_id  \
count          5.700000e+01          5.700000e+01      57.000000
mean           3.495272e+08          3.656962e+08  439096.192982
std            2.344611e+08          2.211175e+08  252402.353441
min            4.226594e+07          3.106059e+07   12439.000000
25%            1.845835e+08          1.668185e+08  220526.000000
```

```
50%            2.751355e+08          3.706184e+08  434778.000000
75%            5.222994e+08          4.836492e+08  628335.000000
max            9.860401e+08          9.631939e+08  937715.000000

       procedure_code  utilization_quantity
count    5.700000e+01             57.000000
mean     4.097584e+08             34.105263
std      2.392178e+08             21.763078
min      5.679208e+07              3.000000
25%      1.975433e+08             15.000000
50%      4.120714e+08             29.000000
75%      5.613608e+08             49.000000
max      9.171131e+08             95.000000

[8 rows x 23 columns]
```

[55]:
```
final_table_prep_df = final_table.copy(deep = True)
```

[56]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
```

[57]:
```
label = LabelEncoder()
```

- `Utilization_quantity` and `expenditure_amount` should not be label encoded. `expenditure_amount` needs to be changed to a float.

[58]:
```
final_table_prep_df = final_table_prep_df.apply(lambda col: label.
↪fit_transform(col.astype(str)), axis=0, result_type='expand')
```

[60]:
```
final_table_prep_df.head()
```

[60]:
```
            patient_id  patient_attribute_1  patient_attribute_2  \
visit_date
2020-05-24          32                   36                   28
2021-06-08          25                   11                   34
2021-10-29          12                   27                   51
2021-06-20           3                   14                   43
2018-07-30          46                   10                    0

            patient_attribute_3  patient_attribute_4  patient_attribute_5  \
visit_date
2020-05-24                   49                   21                   26
2021-06-08                   32                   17                    9
2021-10-29                   53                    7                    6
2021-06-20                   27                   50                   54
2018-07-30                   21                   26                    0
```

```
            patient_attribute_6  patient_attribute_7  patient_attribute_8  \
visit_date
2020-05-24                   24                   17                   49
2021-06-08                   48                   42                    8
2021-10-29                   17                   46                   48
2021-06-20                   22                   55                   46
2018-07-30                   38                   15                   37

            patient_attribute_9  …  physician_attribute_14  \
visit_date                       …
2020-05-24                   11  …                      24
2021-06-08                    3  …                      18
2021-10-29                    1  …                      16
2021-06-20                   12  …                      10
2018-07-30                   51  …                      15

            physician_attribute_15  physician_attribute_16  \
visit_date
2020-05-24                      15                      20
2021-06-08                      13                      14
2021-10-29                      17                       8
2021-06-20                       4                      10
2018-07-30                      16                      25

            physician_attribute_17  physician_attribute_18  \
visit_date
2020-05-24                      21                      12
2021-06-08                      12                      10
2021-10-29                       9                      20
2021-06-20                      26                       7
2018-07-30                       3                      26

            physician_attribute_19  physician_attribute_20  procedure_code  \
visit_date
2020-05-24                       8                      21              33
2021-06-08                      17                      17              44
2021-10-29                      15                      20              18
2021-06-20                      12                       1              38
2018-07-30                      26                      10              48

            utilization_quantity  expenditure_amount
visit_date
2020-05-24                     9                   6
2021-06-08                    11                  22
2021-10-29                     7                  41
2021-06-20                    16                  36
2018-07-30                     4                  38
```

[5 rows x 296 columns]

[ ]:

[68]:
```
# # One Hot Encoding Will Not Be Practical With This Many Columns

# import pandas as pd
# import numpy as np
# from sklearn.preprocessing import OneHotEncoder
# # creating instance of one-hot-encoder
# enc = OneHotEncoder(handle_unknown='ignore')
```

[69]:
```
# final_table_one_hot_encoding_df = final_table.copy(deep = True)
```

2.0.2 Stats Tests

[71]:
```
final_table_prep_df.describe()
```

[71]:

	patient_id	patient_attribute_1	patient_attribute_2	\
count	57.000000	57.000000	57.000000	
mean	26.385965	27.719298	27.859649	
std	16.097642	16.247807	16.389717	
min	0.000000	0.000000	0.000000	
25%	13.000000	14.000000	14.000000	
50%	26.000000	28.000000	28.000000	
75%	40.000000	41.000000	42.000000	
max	54.000000	55.000000	55.000000	

	patient_attribute_3	patient_attribute_4	patient_attribute_5	\
count	57.000000	57.000000	57.000000	
mean	27.631579	27.824561	27.842105	
std	16.193728	16.347917	16.368294	
min	0.000000	0.000000	0.000000	
25%	14.000000	14.000000	14.000000	
50%	28.000000	28.000000	28.000000	
75%	41.000000	42.000000	42.000000	
max	55.000000	55.000000	55.000000	

	patient_attribute_6	patient_attribute_7	patient_attribute_8	\
count	57.000000	57.000000	57.000000	
mean	27.789474	27.771930	27.701754	
std	16.310313	16.293094	16.234844	
min	0.000000	0.000000	0.000000	
25%	14.000000	14.000000	14.000000	
50%	28.000000	28.000000	28.000000	
75%	42.000000	42.000000	41.000000	

```
max              55.000000            55.000000            55.000000

       patient_attribute_9  …  physician_attribute_14  \
count            57.000000  …               57.000000
mean             27.122807  …               13.842105
std              16.412180  …                8.088328
min               0.000000  …                0.000000
25%              13.000000  …                7.000000
50%              27.000000  …               14.000000
75%              41.000000  …               20.000000
max              55.000000  …               28.000000

       physician_attribute_15  physician_attribute_16  physician_attribute_17  \
count               57.000000               57.000000               57.000000
mean                13.035088               13.929825               14.175439
std                  8.119564                7.837318                8.445461
min                  0.000000                0.000000                0.000000
25%                  6.000000                7.000000                7.000000
50%                 13.000000               14.000000               15.000000
75%                 20.000000               20.000000               21.000000
max                 28.000000               28.000000               28.000000

       physician_attribute_18  physician_attribute_19  physician_attribute_20  \
count               57.000000               57.000000               57.000000
mean                13.789474               13.491228               13.631579
std                  8.450987                8.498157                7.861824
min                  0.000000                0.000000                0.000000
25%                  7.000000                6.000000                8.000000
50%                 13.000000               13.000000               14.000000
75%                 21.000000               20.000000               19.000000
max                 28.000000               28.000000               28.000000

       procedure_code  utilization_quantity  expenditure_amount
count       57.000000             57.000000           57.000000
mean        27.000000             16.754386           26.298246
std         16.426242             10.586853           16.181962
min          0.000000              0.000000            0.000000
25%         13.000000              7.000000           12.000000
50%         27.000000             17.000000           26.000000
75%         41.000000             26.000000           40.000000
max         54.000000             35.000000           54.000000

[8 rows x 296 columns]
```

[144]:
```
import pingouin as pg
import scipy.stats as stats
```

[87]:
```
third_of_df = int((len(final_table_prep_df) / 3))
```

[101]:
```
final_table_prep_df = final_table_prep_df.sort_index(ascending=False)
```

[138]:
```
first_third_df = final_table_prep_df.iloc[:third_of_df]
# first_third_df
```

[106]:
```
second_third_df = final_table_prep_df.iloc[third_of_df:third_of_df +␣
↪third_of_df]
# second_third_df
```

[105]:
```
third_third_df = final_table_prep_df.iloc[third_of_df + third_of_df:]
# third_third_df
```

[139]:
```
# concat DF's into single DF
single_df = pd.concat([pd.concat([pd.DataFrame(first_third_df.
↪utilization_quantity.reset_index().drop(columns='visit_date')).
↪rename(columns={'utilization_quantity': 'utilization_quantity_1'}),
                                  second_third_df.utilization_quantity.
↪reset_index().drop(columns='visit_date').
↪rename(columns={'utilization_quantity': 'utilization_quantity_2'})], axis=1),
                       third_third_df.utilization_quantity.reset_index().
↪drop(columns='visit_date').rename(columns={'utilization_quantity':␣
↪'utilization_quantity_3'})], axis=1)
single_df
```

[139]:

	utilization_quantity_1	utilization_quantity_2	utilization_quantity_3
0	29	34	4
1	22	31	30
2	27	6	31
3	28	7	1
4	34	18	1
5	28	17	19
6	4	15	26
7	33	35	9
8	5	19	3
9	17	19	24
10	2	6	6
11	28	25	14
12	18	12	17
13	21	16	1
14	10	22	4
15	32	18	10
16	20	13	12
17	23	11	8
18	30	0	0

[141]:
```
df_melt = pd.melt(single_df.reset_index(), id_vars=['index'],␣
↪value_vars=['utilization_quantity_1', 'utilization_quantity_2',␣
↪'utilization_quantity_3'])
df_melt
```

[141]:

	index	variable	value
0	0	utilization_quantity_1	29
1	1	utilization_quantity_1	22
2	2	utilization_quantity_1	27
3	3	utilization_quantity_1	28
4	4	utilization_quantity_1	34
5	5	utilization_quantity_1	28
6	6	utilization_quantity_1	4
7	7	utilization_quantity_1	33
8	8	utilization_quantity_1	5
9	9	utilization_quantity_1	17
10	10	utilization_quantity_1	2
11	11	utilization_quantity_1	28
12	12	utilization_quantity_1	18
13	13	utilization_quantity_1	21
14	14	utilization_quantity_1	10
15	15	utilization_quantity_1	32
16	16	utilization_quantity_1	20
17	17	utilization_quantity_1	23
18	18	utilization_quantity_1	30
19	0	utilization_quantity_2	34
20	1	utilization_quantity_2	31
21	2	utilization_quantity_2	6
22	3	utilization_quantity_2	7
23	4	utilization_quantity_2	18
24	5	utilization_quantity_2	17
25	6	utilization_quantity_2	15
26	7	utilization_quantity_2	35
27	8	utilization_quantity_2	19
28	9	utilization_quantity_2	19
29	10	utilization_quantity_2	6
30	11	utilization_quantity_2	25
31	12	utilization_quantity_2	12
32	13	utilization_quantity_2	16
33	14	utilization_quantity_2	22
34	15	utilization_quantity_2	18
35	16	utilization_quantity_2	13
36	17	utilization_quantity_2	11
37	18	utilization_quantity_2	0
38	0	utilization_quantity_3	4
39	1	utilization_quantity_3	30
40	2	utilization_quantity_3	31

```
41      3  utilization_quantity_3      1
42      4  utilization_quantity_3      1
43      5  utilization_quantity_3     19
44      6  utilization_quantity_3     26
45      7  utilization_quantity_3      9
46      8  utilization_quantity_3      3
47      9  utilization_quantity_3     24
48     10  utilization_quantity_3      6
49     11  utilization_quantity_3     14
50     12  utilization_quantity_3     17
51     13  utilization_quantity_3      1
52     14  utilization_quantity_3      4
53     15  utilization_quantity_3     10
54     16  utilization_quantity_3     12
55     17  utilization_quantity_3      8
56     18  utilization_quantity_3      0
```

[142]:
```
longFormat = df_melt[['variable', 'value']]
print('Anova')
pg.anova(data=longFormat, dv='value', between='variable', detailed=True)
```

```
Anova
```

[142]:

	Source	SS	DF	MS	F	p-unc	np2
0	variable	962.561404	2	481.280702	4.890696	0.011166	0.153358
1	Within	5314.000000	54	98.407407	NaN	NaN	NaN

[151]:
```
from bioinfokit.analys import stat
import matplotlib.pyplot as plt
# perform multiple pairwise comparison (Tukey's HSD)
# unequal sample size data, tukey_hsd uses Tukey-Kramer test
res = stat()
res.tukey_hsd(df=df_melt, res_var='value', xfac_var='variable',␣
 ↪anova_model='value ~ C(variable)')
print('Tukey\'s')
res.tukey_summary
```

```
Tukey's
```

[151]:

	group1	group2	Diff	Lower	Upper	q-value	p-value
0	utilization_quantity_1	utilization_quantity_2	4.578947	-3.177471	12.335365	2.012003	0.337153
1	utilization_quantity_1	utilization_quantity_3	10.052632	2.296214	17.809049	4.417155	0.007952
2	utilization_quantity_2	utilization_quantity_3	5.473684	-2.282734	13.230102	2.405153	0.214435

[152]:
```
res = stat()
res.levene(df=df_melt, res_var='value', xfac_var='variable', center='mean')
print('levene_summary')
res.levene_summary
```

```
levene_summary
```

[152]:

	Parameter	Value
0	Test statistics (W)	0.2733
1	Degrees of freedom (Df)	2.0000
2	p value	0.7619

[154]:
```
## Two-tail test at 5% sig
#pg.ttest(df['I'], hypothesisVal, paired=False, tail='two-sided', confidence=0.
 ↪95).round(2)

print('95% C.I.')
ciTtest = pg.ttest(single_df['utilization_quantity_1'],␣
 ↪single_df['utilization_quantity_2'], paired=False, alternative='two-sided',␣
 ↪confidence=0.95)
ciTtest
```

```
95% C.I.
```

[154]:

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	1.445169	36	two-sided	0.15706	[-1.85, 11.0]	0.468875	0.711	0.290429

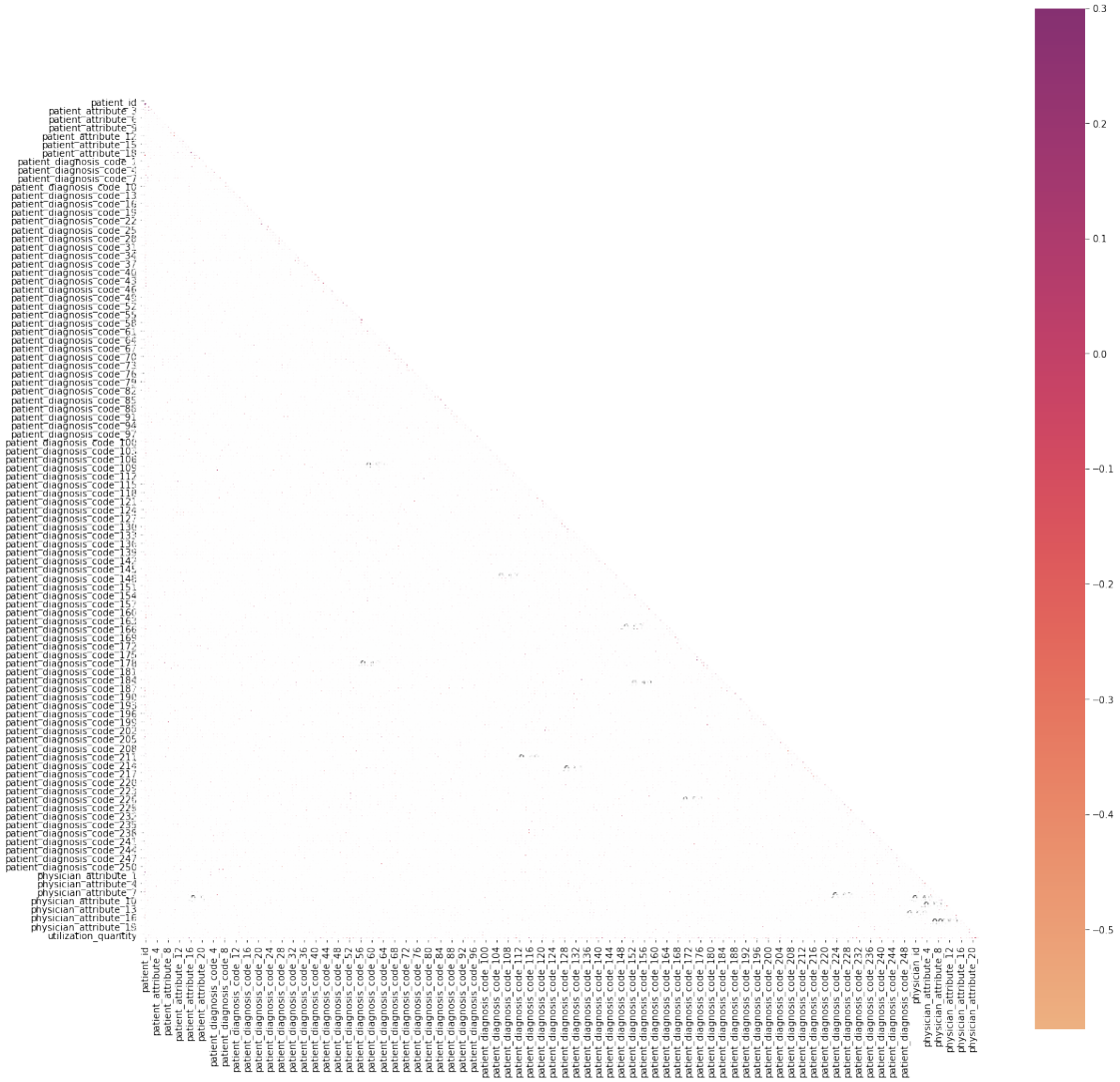
2.0.3 Correlation Matrix

We should take a subset or reduce the columns before plotting.

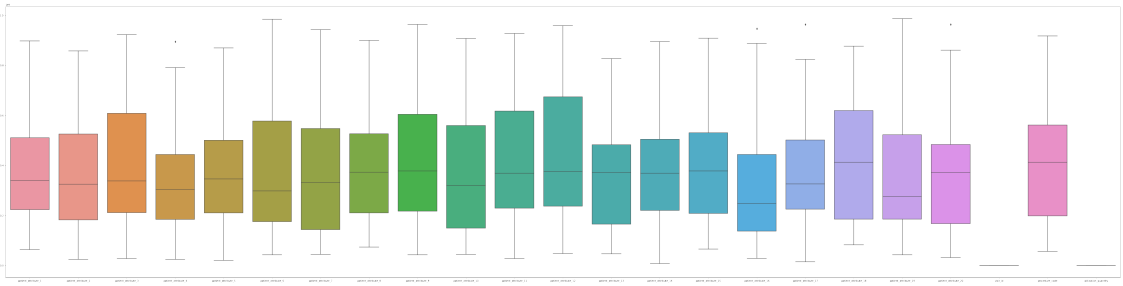
[155]:
```
from string import ascii_letters
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

sns.color_palette("flare", as_cmap=True)

# Compute the correlation matrix
corr = final_table_prep_df.corr()

# Generate a mask for the upper triangle
mask = np.triu(np.ones_like(corr, dtype=bool))

# Set up the matplotlib figure
```

```
f, ax = plt.subplots(figsize=(20, 20))
cmap = sns.color_palette("flare", as_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, annot=True)
```

[155]: <AxesSubplot:>



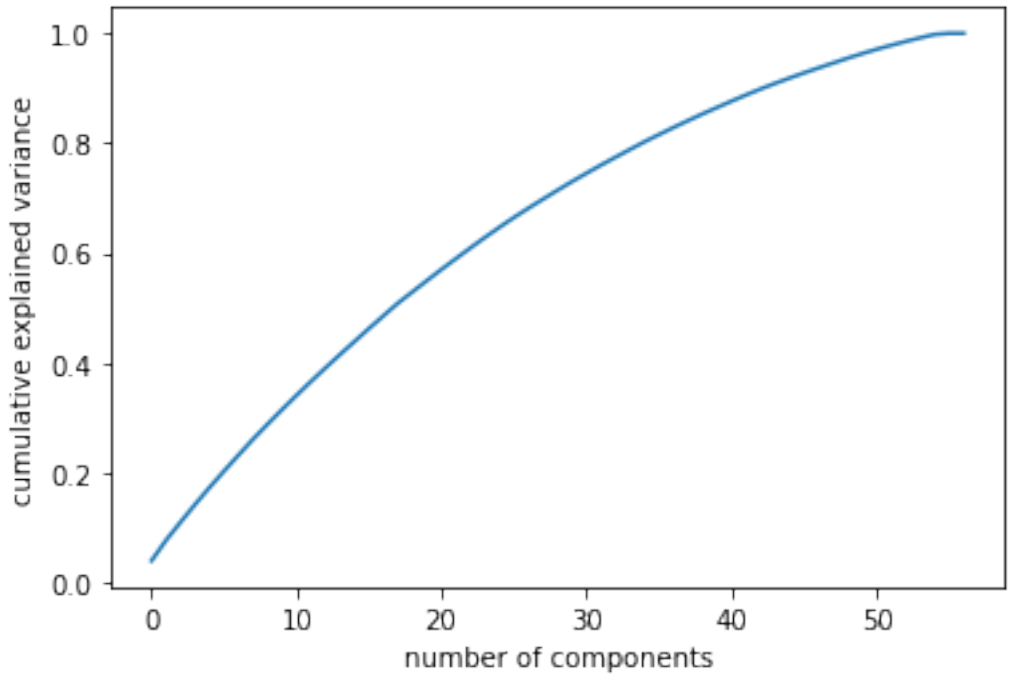
```
plt.figure(figsize=(80,20))

sns.boxplot(data=final_table)
plt.show()
```



2.0.4 PCA

[157]:
```
from sklearn.decomposition import PCA
```

[159]:
```
pca = PCA().fit(final_table_prep_df)
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance');
```



Select an appropriate amount of components

[160]:
```
n_components = 50

pca = PCA(n_components=n_components)
```

[161]:
```
pca.fit(final_table_prep_df)
X_pca = pca.transform(final_table_prep_df)
print("original shape:   ", final_table_prep_df.shape)
print("transformed shape:", X_pca.shape)
```

```
original shape:    (57, 296)
transformed shape: (57, 50)
```

[174]:
```
pca_col_list = []
for x in range(X_pca.shape[1]):
    pca_col_list.append('principal_component_' + str(x + 1))

print(pca_col_list)
```

```
['principal_component_1', 'principal_component_2', 'principal_component_3',
'principal_component_4', 'principal_component_5', 'principal_component_6',
'principal_component_7', 'principal_component_8', 'principal_component_9',
'principal_component_10', 'principal_component_11', 'principal_component_12',
'principal_component_13', 'principal_component_14', 'principal_component_15',
'principal_component_16', 'principal_component_17', 'principal_component_18',
'principal_component_19', 'principal_component_20', 'principal_component_21',
'principal_component_22', 'principal_component_23', 'principal_component_24',
'principal_component_25', 'principal_component_26', 'principal_component_27',
'principal_component_28', 'principal_component_29', 'principal_component_30',
'principal_component_31', 'principal_component_32', 'principal_component_33',
'principal_component_34', 'principal_component_35', 'principal_component_36',
'principal_component_37', 'principal_component_38', 'principal_component_39',
'principal_component_40', 'principal_component_41', 'principal_component_42',
'principal_component_43', 'principal_component_44', 'principal_component_45',
'principal_component_46', 'principal_component_47', 'principal_component_48',
'principal_component_49', 'principal_component_50']
```

[175]:
```
principalDf = pd.DataFrame(data = X_pca, columns = pca_col_list)
```

[176]:
```
principalDf
```

[176]:

	principal_component_1	principal_component_2	principal_component_3	\
0	-48.880887	21.842735	-13.476592	
1	-13.270611	131.567419	28.584155	
2	-58.846777	-28.722315	-25.971137	
3	-1.576397	50.090790	14.644794	
4	-49.932757	-48.642680	-38.803895	
5	-15.603633	14.985138	15.166020	
6	19.207320	36.276144	-54.290655	
7	-42.672611	60.412674	-73.233410	
8	12.873347	26.349804	32.066082	
9	8.640831	-33.332512	-67.345318	
10	16.990236	-57.319235	-31.121684	

11	8.665980	-37.973775	-39.290757
12	26.192674	-32.602075	-69.592311
13	4.808524	-60.735099	-5.349943
14	-17.944131	52.251157	12.902688
15	-23.638959	61.123144	-15.990389
16	-0.840292	36.286117	-18.196664
17	28.951547	-21.783298	-16.690228
18	-63.418632	-43.497748	58.307373
19	17.764891	-59.898155	-64.477860
20	-9.235194	-17.035624	71.284474
21	-44.651853	-91.071677	-62.552975
22	-42.004277	-10.522770	22.905265
23	0.016459	33.747760	-20.453422
24	18.388781	60.346935	-70.202987
25	5.157777	-81.973181	53.916530
26	15.241035	35.543331	22.420762
27	182.841637	28.697492	-4.811263
28	176.538865	13.778813	-12.186428
29	4.383635	16.830605	-13.001147
30	-9.444450	14.269142	55.495061
31	90.396508	-59.569904	76.886145
32	-22.118647	-7.529756	30.704398
33	14.127693	-56.294849	-18.390157
34	59.830809	16.908243	-47.271339
35	42.876106	68.066143	19.138052
36	-24.548633	-0.062064	59.124825
37	-18.705836	24.370669	-30.686849
38	-37.973031	4.843467	-42.865980
39	-67.489112	-9.166503	-26.868786
40	-3.396200	-40.793369	65.295118
41	-33.163382	18.407865	-64.911650
42	-7.060250	-52.726803	73.427259
43	-23.744352	-31.723233	-11.554626
44	41.343582	-47.647856	42.321184
45	-47.724089	74.211286	21.577556
46	-2.773300	12.212992	50.305651
47	20.701508	16.349755	-24.289612
48	-43.115717	-2.414131	-9.145750
49	8.311656	15.059440	30.613024
50	-46.900028	33.779268	61.511499
51	4.913132	17.926763	-9.092452
52	-26.854384	21.177803	67.671844
53	21.433957	-74.436130	2.105806
54	9.278442	-44.729473	-1.099414
55	-38.373923	-11.288728	-14.594885
56	26.025409	45.780049	29.435002

	principal_component_4	principal_component_5	principal_component_6	\
0	46.875666	53.646734	-25.066691	
1	20.109392	-27.828453	28.339414	
2	14.435185	94.232865	-5.411807	
3	58.040547	-73.296997	-35.952600	
4	90.103681	1.547324	-30.275774	
5	-10.525021	3.267108	53.589357	
6	7.500330	42.581156	63.604878	
7	-41.916531	38.637464	0.264449	
8	-49.983158	36.784169	45.341224	
9	-44.351129	-61.959298	-31.267409	
10	29.376647	96.260041	-49.709136	
11	-12.340338	-27.658439	27.941014	
12	-44.535867	19.414157	55.627132	
13	62.854142	2.188620	11.920657	
14	-45.666489	-1.571916	-22.221037	
15	38.368091	-58.156374	28.592373	
16	32.065195	33.133733	75.144702	
17	10.155712	9.885924	-32.304326	
18	-5.126953	1.046722	20.386059	
19	-11.136692	-49.625876	-37.200336	
20	58.416873	-19.102044	-58.481030	
21	-19.075217	8.779728	24.398571	
22	81.574815	11.358765	-29.026669	
23	8.212832	-59.583885	-65.670281	
24	51.408194	-15.314823	23.722825	
25	38.374712	29.272159	12.564649	
26	12.519355	-57.938972	1.552909	
27	54.292251	44.461142	14.387162	
28	54.453950	27.627685	4.900283	
29	-6.366114	-20.827750	110.649593	
30	-13.557755	-37.747951	-45.112558	
31	-53.890722	-4.457944	-3.000198	
32	54.552919	-4.558454	-30.124858	
33	-0.047338	-84.596711	15.320693	
34	-54.750641	-29.631472	-28.284694	
35	-47.444050	-14.490502	-38.322612	
36	-79.645434	12.567656	-14.405393	
37	-51.269465	-37.445560	-31.898740	
38	-20.825000	59.319389	33.909429	
39	-11.498458	41.866416	-26.071660	
40	39.666281	-11.261286	9.859553	
41	11.585130	-13.486415	-26.526907	
42	-61.523362	-19.144413	11.295382	
43	21.333756	15.758489	16.515597	
44	-39.570384	13.508143	-3.205530	
45	-55.768721	0.260507	6.580659	

46	35.418504	-26.587710	44.487366
47	-3.346250	4.513120	-71.160286
48	15.255131	-17.184157	-68.480540
49	-48.743361	67.813018	-0.544506
50	29.401813	2.189501	62.969672
51	-12.580004	-19.371199	-39.629341
52	24.120292	34.240944	22.007944
53	-65.724257	-39.071577	18.238363
54	9.968777	-56.506276	52.053616
55	-45.404358	-18.061055	40.389615
56	-53.827106	100.304830	-87.200222

	principal_component_7	principal_component_8	principal_component_9	\
0	-70.719680	-47.192284	4.546918	
1	-21.874305	-17.583118	-55.256456	
2	3.528834	15.234702	45.772591	
3	73.494188	-14.511567	-4.717472	
4	-37.222767	-42.850792	31.776348	
5	7.045368	18.344876	77.328792	
6	83.075553	-85.740653	-3.511476	
7	-60.059536	15.916831	-62.875032	
8	35.808318	17.560278	15.789674	
9	-6.779386	92.509991	-21.968460	
10	31.879517	48.349349	-23.490133	
11	-30.877222	9.463691	55.045708	
12	-40.248427	15.624062	-1.621276	
13	47.535158	-15.888253	-1.422237	
14	23.053089	18.921637	51.947322	
15	-3.325466	-11.759368	36.040814	
16	104.002713	9.468380	29.647386	
17	-42.251354	109.261572	8.394706	
18	-36.958925	28.767368	-3.985857	
19	11.158777	-1.948184	-2.542153	
20	26.406096	25.421072	62.505128	
21	38.217986	-20.927437	-17.313893	
22	28.845836	-50.689457	-8.673700	
23	6.479532	-40.226002	-20.508717	
24	30.856826	23.601208	-25.930425	
25	-20.618543	-39.109626	-13.845279	
26	-42.674298	-45.879241	103.895287	
27	-45.894401	0.093124	11.841819	
28	-38.027484	16.258716	7.304624	
29	6.601575	47.577190	-59.173398	
30	25.471089	36.455306	-40.969541	
31	18.433007	-11.045744	32.304705	
32	-14.909407	-11.456991	-56.797161	
33	75.665012	43.205531	-41.476161	

34	-49.171636	-35.291875	47.124820
35	-10.715928	-35.217586	9.814183
36	0.374013	-41.425015	19.658544
37	3.586149	12.978868	7.917179
38	31.419366	-55.203099	-4.510048
39	-85.361746	6.591837	-9.191131
40	-30.810583	22.272828	-12.500056
41	-46.031267	-31.034263	-18.241664
42	-21.255099	1.577355	-29.093019
43	31.549780	66.353358	57.620992
44	30.073641	-65.431242	-53.304508
45	10.659044	8.214976	-10.252305
46	-34.111682	-3.642187	-19.639536
47	27.442325	4.471311	-45.183736
48	25.593376	-12.306787	45.375863
49	-31.410855	-9.164763	15.968245
50	-7.517280	88.310302	6.955498
51	-1.853974	50.441066	-11.359228
52	-39.024975	9.468141	-58.872221
53	8.096856	-66.621379	-17.826622
54	-43.598437	-51.106412	-70.996623
55	-7.242581	-11.445365	74.984788
56	74.194223	11.983769	-32.512412

	principal_component_10	…	principal_component_41	\
0	-26.590218	…	15.695017	
1	-39.022550	…	0.227264	
2	27.399608	…	-16.835447	
3	-46.573815	…	-3.138534	
4	-64.145682	…	7.404468	
5	43.612245	…	4.736682	
6	36.710889	…	6.854724	
7	-13.732407	…	-36.751863	
8	-33.570906	…	-5.311587	
9	12.219288	…	-2.567010	
10	1.433552	…	-32.169231	
11	-10.104644	…	-26.120204	
12	-23.636398	…	6.917905	
13	-6.198610	…	-2.343208	
14	-58.639521	…	2.843400	
15	-41.553408	…	-40.032005	
16	-17.059537	…	28.219974	
17	-31.919171	…	41.358363	
18	27.660389	…	-14.283666	
19	-1.164075	…	0.110238	
20	-34.346996	…	19.313218	
21	17.972029	…	-28.359807	

```
22               49.606750  …              14.946131
23               20.889701  …              31.478109
24               34.638980  …              16.848990
25               -6.046652  …             -60.224977
26               -9.033263  …             -53.043955
27               27.042467  …              -0.289604
28               40.606378  …              -3.444329
29              -27.637450  …              30.304298
30               14.136094  …             -27.860698
31               28.833596  …             -17.066378
32               58.228942  …              23.928452
33              -39.519168  …             -27.125797
34              -31.479827  …              12.134819
35               -5.554557  …             -38.271941
36               -7.106521  …              15.664682
37               47.010533  …              36.472909
38               -5.909356  …             -16.514046
39              -24.418569  …              10.092944
40               50.944724  …               0.391198
41               -8.767526  …             -25.990724
42              -19.897813  …               3.189521
43              -32.577417  …               0.107612
44              -93.091527  …              33.730298
45              125.780714  …             -42.298165
46              -32.527527  …               2.387549
47               18.351705  …              -0.271713
48               72.172159  …              42.351785
49               20.357437  …              61.057161
50               29.324057  …              -5.588957
51                2.897716  …              -1.754638
52              -34.953195  …              10.225825
53               32.414940  …              13.160468
54               30.352953  …              15.354984
55                1.377795  …              38.915740
56              -45.197340  …             -18.766245

    principal_component_42  principal_component_43  principal_component_44  \
0                14.462637              -51.927053               34.724752
1                -8.312966               -9.355985               14.451431
2                66.850046                5.575892              -33.714682
3               -28.463691               27.838085              -32.199750
4                -8.468937               19.050927                3.130873
5                44.474804               -1.963655              -11.799322
6               -33.041464               26.910575                9.936821
7                -7.269639              -21.263062              -25.272922
8               -33.338693              -24.088189                1.277834
9               -26.964766               16.723323               21.033953
```

10	-36.181273	-30.842820	-14.376663
11	-25.697841	34.903829	-40.316473
12	-14.528333	4.425275	15.362330
13	-17.780803	20.544974	-0.069342
14	26.639769	-38.168389	-5.710445
15	8.979103	7.427925	18.572062
16	-5.377466	-7.240894	-15.950461
17	-12.532024	-15.467832	8.596200
18	-32.509142	6.735043	-18.566240
19	5.766555	-17.614313	-48.656043
20	13.329935	9.025804	-22.748021
21	1.347031	-31.443625	34.321945
22	-36.817912	-20.641576	-23.410960
23	31.827622	-9.355006	-7.630860
24	13.453301	11.225108	42.855720
25	34.461577	18.294912	14.767421
26	-13.001402	-9.724816	7.785444
27	5.789516	-4.510134	-5.603811
28	29.113087	3.506622	7.881983
29	20.499641	-14.538193	-35.512327
30	-16.565020	15.126368	28.863751
31	-42.814092	-47.633289	18.040533
32	21.944562	-21.884126	-8.189357
33	47.256337	0.635170	30.079621
34	-2.203665	-10.655692	-36.426568
35	7.723785	13.298120	-10.318591
36	13.529037	-10.033583	-17.532083
37	-41.575202	-18.691316	-21.249379
38	1.339977	23.729978	-19.956701
39	-9.796909	53.760830	-14.686833
40	-25.181445	40.591123	19.156812
41	-7.194770	-13.458268	45.832608
42	36.439853	2.386519	2.836421
43	6.240452	-37.262085	19.789417
44	35.887747	3.219592	13.862458
45	20.878043	-9.123144	7.099054
46	-41.839697	-4.347467	-0.493965
47	13.430726	-3.131525	-26.578627
48	8.212417	-18.867059	28.967987
49	-5.397128	55.170821	28.603552
50	7.946282	12.485571	-29.732737
51	27.824983	50.207858	11.160182
52	-15.511077	-40.983719	8.915612
53	-11.981669	-2.667285	-11.146071
54	14.168608	5.167964	-25.679236
55	-5.888645	20.392951	54.381757
56	-13.581759	38.522940	11.239940

	principal_component_45	principal_component_46	principal_component_47	\
0	-16.981720	20.153828	7.618229	
1	45.050641	-26.852482	10.045410	
2	33.833752	-26.303725	4.863265	
3	6.119078	5.512488	36.630182	
4	1.241066	47.956076	-17.689491	
5	-14.299257	7.233533	-18.768189	
6	23.752265	31.093931	25.204467	
7	-47.240714	21.977582	27.149894	
8	58.078505	32.161238	-2.074422	
9	36.871194	18.644010	21.914765	
10	-29.796870	0.730895	15.988961	
11	-16.549745	28.285119	-6.891262	
12	-7.604428	-25.652756	-2.388155	
13	-43.146722	1.704692	-27.832655	
14	8.326061	-21.487922	-16.709651	
15	2.329626	-42.878771	-38.859243	
16	-10.828856	-10.295709	10.202026	
17	0.861118	-18.264687	6.815166	
18	7.616245	-1.240029	-14.164715	
19	-9.426097	-19.110599	-23.053812	
20	9.848188	-2.090617	15.851046	
21	47.950932	-8.667250	-29.158735	
22	-18.308437	-24.487070	21.223096	
23	-15.373870	-15.409503	-4.663875	
24	-8.082416	29.153094	-48.333553	
25	-3.962178	-7.893952	53.033165	
26	-0.482258	11.530596	-20.995480	
27	5.374946	-24.071782	0.646669	
28	-3.707757	-7.451770	-1.678479	
29	-50.665000	24.923123	-17.481744	
30	-8.719709	7.079335	-17.537493	
31	4.706590	15.737817	-4.855943	
32	17.513004	59.805024	-10.597685	
33	-3.980467	17.123030	20.468862	
34	16.317733	41.897542	34.759752	
35	-2.306833	8.461107	-8.022030	
36	-25.845969	37.734407	-11.760961	
37	0.255337	-47.472269	18.673196	
38	9.261377	-23.308034	5.344914	
39	28.164200	-17.878876	-24.858259	
40	0.618010	-2.604614	6.455594	
41	-9.287496	-16.123463	18.474974	
42	-13.790047	3.794124	-15.134859	
43	19.118176	-2.141561	5.747536	
44	4.903936	-20.725198	35.221055	

45	-19.065498	-0.611830	27.306687
46	-23.257288	-25.077634	-10.240680
47	39.098170	21.652755	-13.582954
48	-6.494852	-9.769900	-9.681008
49	12.637014	11.834671	-3.888956
50	18.117402	-1.304077	13.533312
51	-15.910097	-8.965138	31.449559
52	3.331015	-0.313678	-2.380503
53	-11.193104	-41.339907	-19.046647
54	37.158891	-5.252395	-17.359090
55	-36.798816	5.963506	38.001773
56	-25.347970	-7.096327	-52.933024

	principal_component_48	principal_component_49	principal_component_50
0	-12.731995	-8.117939	-13.452872
1	31.180680	0.746771	-20.219628
2	-19.923386	0.939739	-22.406623
3	-6.058459	-22.572469	6.983921
4	-5.117387	-6.572617	9.116924
5	41.195413	-5.903178	1.992677
6	9.167542	-10.577310	-42.238238
7	-24.240324	-9.057959	38.093160
8	-8.930614	-13.197546	0.114904
9	-27.650368	28.685119	-49.883057
10	23.997583	-4.678408	-10.345915
11	6.123639	-27.008679	-4.622486
12	52.557259	12.348795	-0.755569
13	-17.147359	59.529850	-13.440179
14	-18.607128	-9.888135	-24.847147
15	-23.999683	11.196133	1.415956
16	-8.368312	4.476770	59.802441
17	32.304889	-16.857578	37.817254
18	-4.741091	45.221034	4.315663
19	17.010991	12.146786	-29.896337
20	-12.680120	17.118237	-1.226656
21	-23.790663	-21.302657	19.416198
22	-0.272361	-16.823879	1.802100
23	34.985320	19.029400	-18.191294
24	17.206753	-15.460076	8.082115
25	33.042384	-20.876852	-24.047427
26	-8.893913	18.416978	12.959211
27	-19.450671	-3.375714	-7.270161
28	-28.157017	0.886978	-15.814646
29	-26.087255	-2.337747	-28.718255
30	12.201315	-8.741172	-35.301131
31	14.610941	2.517970	18.476596
32	31.495722	29.394176	20.383954

33	-3.555045	14.542451	10.723984
34	5.867149	14.460906	2.418538
35	20.442411	-15.528300	31.948714
36	-38.154179	1.625859	-18.867610
37	-20.820743	-7.608999	-4.056460
38	7.870588	-1.453905	-13.963251
39	15.787396	8.454497	10.771048
40	-38.279674	-26.639401	24.926577
41	-11.196129	16.380661	8.387467
42	17.601608	-55.236571	-14.639770
43	6.750241	18.829772	16.058296
44	0.734616	43.834362	25.830567
45	1.659793	49.760470	19.660388
46	35.070725	11.473215	-13.589262
47	15.334939	-14.900562	5.640617
48	-3.782783	-34.409040	-6.672808
49	-12.025637	40.561748	-1.638205
50	20.037972	-4.428366	-5.538048
51	-35.522644	-25.376939	20.532788
52	-37.905341	-12.287951	-27.372334
53	-13.604107	-8.786806	8.727129
54	-13.933189	-19.608612	38.928242
55	17.096310	-26.761247	-7.847609
56	4.293402	-6.202063	11.535547

[57 rows x 50 columns]

2.0.5 Test Train Split

[177]:
```
from sklearn.model_selection import train_test_split

train, test = train_test_split(principalDf, test_size=0.2)
```

[ ]:
```

```