MIMIC III SQL

AI 395T - AI in Healthcare - Dr. Ying Ding

Joseph Skrovan, 2/10/2025

Notes:

- My searches are all related to diabetes mellitus. In the US the median HbA1C measurement of patients admitted to the hospital (ICU) in the MIMIC III dataset is in the pre diabetic range. 19% of ICU admission receive a diabetes diagnosis code.
- The Jupyter notebook and support programs I wrote are available from this Github repo: git@github.com:jskrovan2/aihc.git
  - sql.ipynb - Code to sql searches
  - I downloaded the MIMIC III dataset and accessed it locally with the duckdb Python module. I wrote and used the following programs:
    - convert_to_parquet.py to translate <table>.csv.gz file into the parquet format for fast reading by duckdb.
    - distill_chartevents.py and distill_noteevents.py to make smaller tables with only the rows containing data I wanted to use.
      - The full tables were loading fine in Python programs but crashed my Jupyter notebook when I tried to load them.
- I used a combination of Python and SQL to find data I was interested in. Full Python code details can be found in the sql.ipynb file. Some setup is not mentioned in the slides. For example tables are loaded at the start, but is the code is not particularly interesting (the last two tables are distilled):

```
for table in ('LABEVENTS', 'DIAGNOSES_ICD', 'D_ICD_DIAGNOSES', 'D_ITEMS', 'PATIENTS',
              'ADMISSIONS', 'PRESCRIPTIONS', 'chartevents_height_weight_systolic_diastolic',
              'noteevents_metformin'):
    con.execute(f"CREATE TABLE {table} AS SELECT * FROM '{table}.parquet'")
```

- I had very little prior experience with SQL searches prior to this assignment:
  - I received help in creating my duckdb SQL searches from ChatGPT.
  - I did not send MIMIC III data to ChatGPT.

ICD9 Diabetes Codes

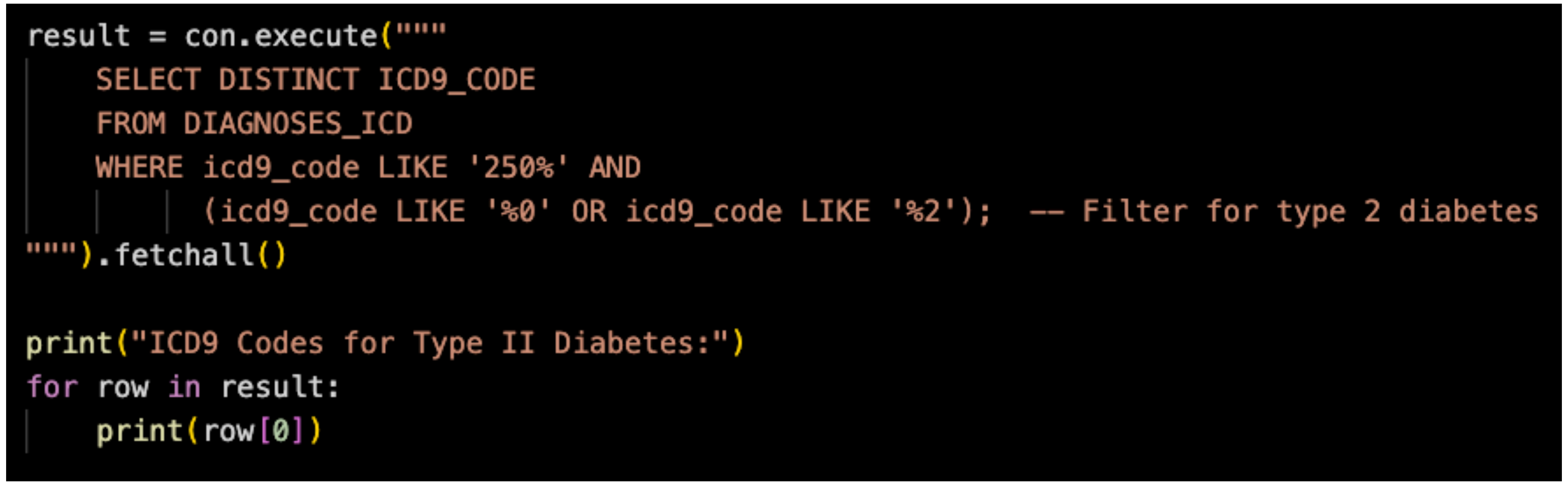
Find ICD9 code for Diabetes diagnoses. I will primarily be searching for patient data with these codes.

Note: While not precise, future slides report “type II or unspecified type” as “type II” and “not stated as uncontrolled” as “controlled” to simplify descriptions.

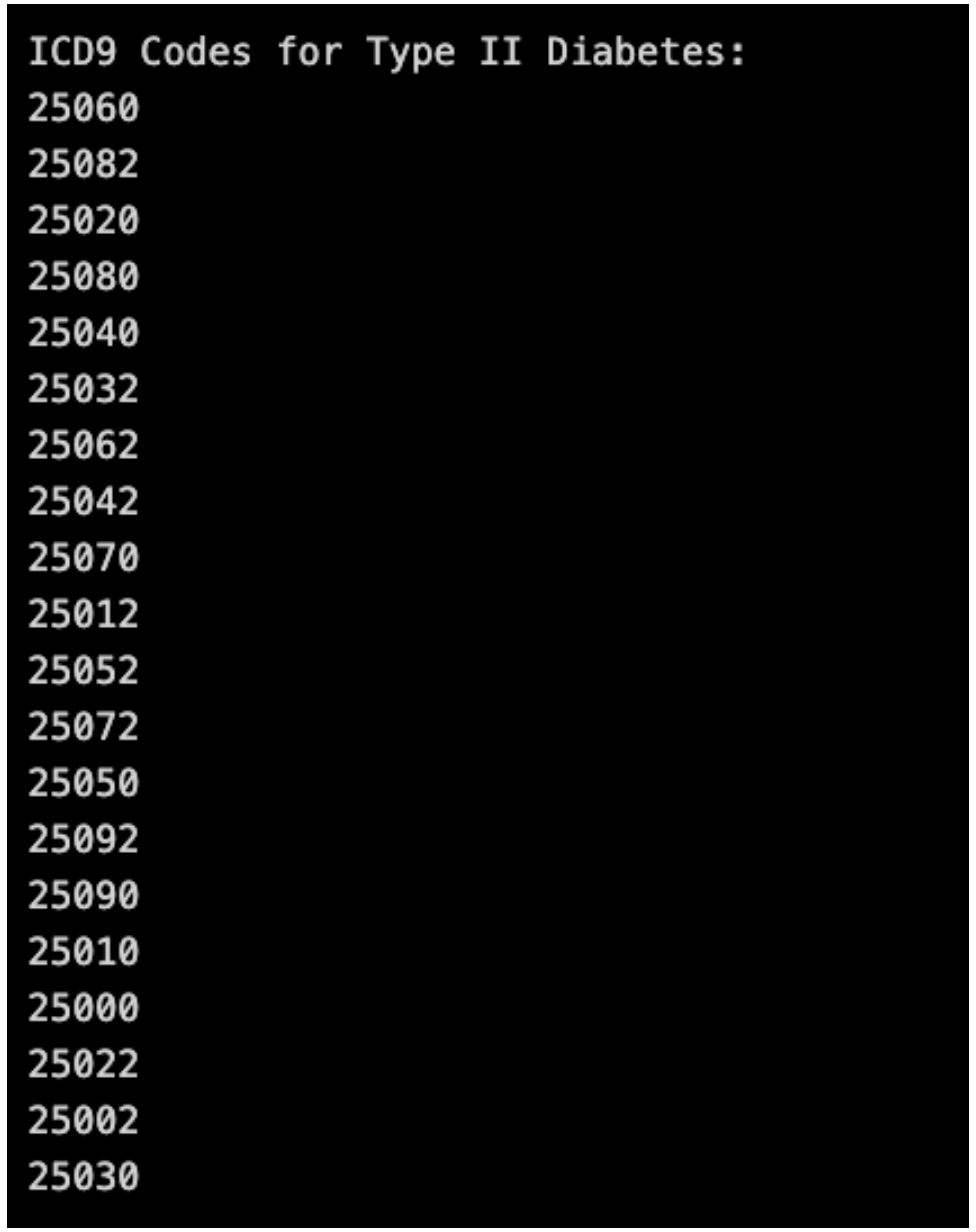
ICD9 Diabetes Codes:

```
250**:  diabetes mellitus: first 3 digits is the diabetes code
250x*:                     fourth digit specifies the type of complication
250*0:                     fifth digit, 0, type II or unspecified type, not stated as uncontrolled
250*2:                     fifth digit, 2, type II or unspecified type, uncontrolled
```

Search #1



```
result = con.execute("""
    SELECT DISTINCT ICD9_CODE
    FROM DIAGNOSES_ICD
    WHERE icd9_code LIKE '250%' AND
          (icd9_code LIKE '%0' OR icd9_code LIKE '%2');  -- Filter for type 2 diabetes
""").fetchall()

print("ICD9 Codes for Type II Diabetes:")
for row in result:
    print(row[0])
```



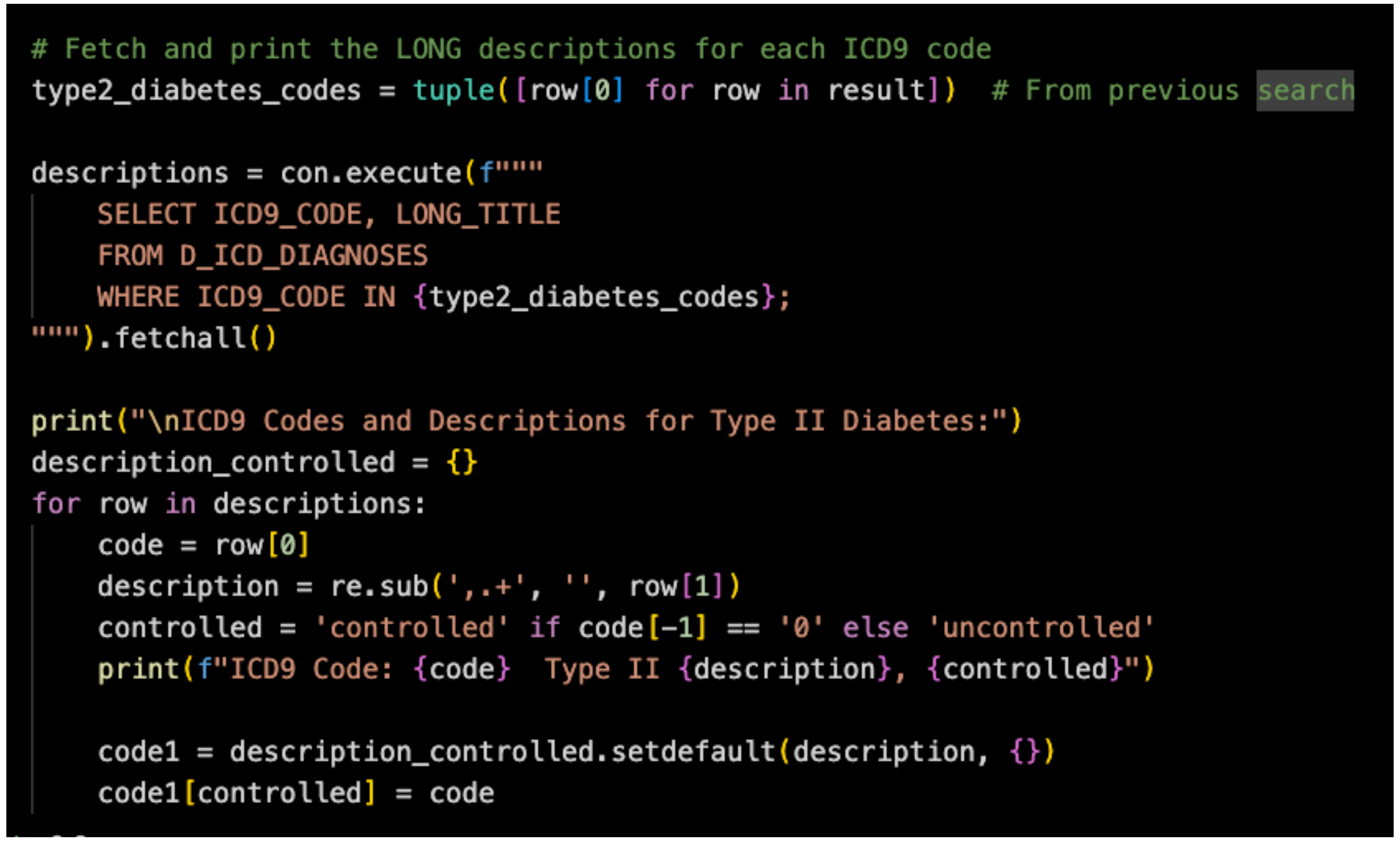
```
ICD9 Codes for Type II Diabetes:
25060
25082
25020
25080
25040
25032
25062
25042
25070
25012
25052
25072
25050
25092
25090
25010
25000
25022
25002
25030
```

ICD9 Diabetes Code Labels

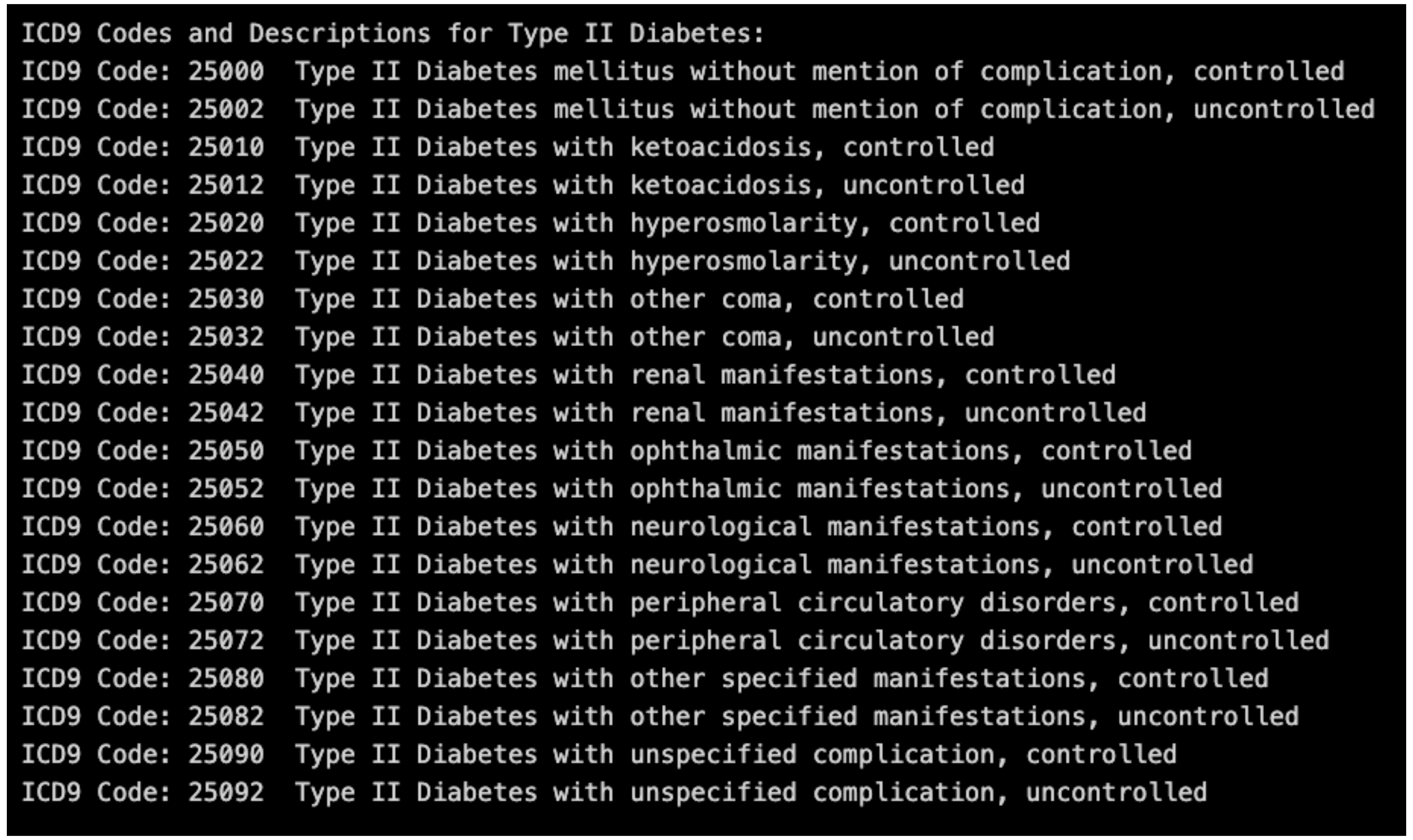
Find ICD9 code LONG_TITLEs (text) from D_ICD_DIAGNOSES.

The codes mention the major complications (conditions likely caused by diabetes.)

Search #2



```python
# Fetch and print the LONG descriptions for each ICD9 code
type2_diabetes_codes = tuple([row[0] for row in result])  # From previous search

descriptions = con.execute(f"""
    SELECT ICD9_CODE, LONG_TITLE
    FROM D_ICD_DIAGNOSES
    WHERE ICD9_CODE IN {type2_diabetes_codes};
""").fetchall()

print("\nICD9 Codes and Descriptions for Type II Diabetes:")
description_controlled = {}
for row in descriptions:
    code = row[0]
    description = re.sub(',.+', '', row[1])
    controlled = 'controlled' if code[-1] == '0' else 'uncontrolled'
    print(f"ICD9 Code: {code}  Type II {description}, {controlled}")

    code1 = description_controlled.setdefault(description, {})
    code1[controlled] = code
```



```
ICD9 Codes and Descriptions for Type II Diabetes:
ICD9 Code: 25000  Type II Diabetes mellitus without mention of complication, controlled
ICD9 Code: 25002  Type II Diabetes mellitus without mention of complication, uncontrolled
ICD9 Code: 25010  Type II Diabetes with ketoacidosis, controlled
ICD9 Code: 25012  Type II Diabetes with ketoacidosis, uncontrolled
ICD9 Code: 25020  Type II Diabetes with hyperosmolarity, controlled
ICD9 Code: 25022  Type II Diabetes with hyperosmolarity, uncontrolled
ICD9 Code: 25030  Type II Diabetes with other coma, controlled
ICD9 Code: 25032  Type II Diabetes with other coma, uncontrolled
ICD9 Code: 25040  Type II Diabetes with renal manifestations, controlled
ICD9 Code: 25042  Type II Diabetes with renal manifestations, uncontrolled
ICD9 Code: 25050  Type II Diabetes with ophthalmic manifestations, controlled
ICD9 Code: 25052  Type II Diabetes with ophthalmic manifestations, uncontrolled
ICD9 Code: 25060  Type II Diabetes with neurological manifestations, controlled
ICD9 Code: 25062  Type II Diabetes with neurological manifestations, uncontrolled
ICD9 Code: 25070  Type II Diabetes with peripheral circulatory disorders, controlled
ICD9 Code: 25072  Type II Diabetes with peripheral circulatory disorders, uncontrolled
ICD9 Code: 25080  Type II Diabetes with other specified manifestations, controlled
ICD9 Code: 25082  Type II Diabetes with other specified manifestations, uncontrolled
ICD9 Code: 25090  Type II Diabetes with unspecified complication, controlled
ICD9 Code: 25092  Type II Diabetes with unspecified complication, uncontrolled
```

ICD9 Diabetes Codes:

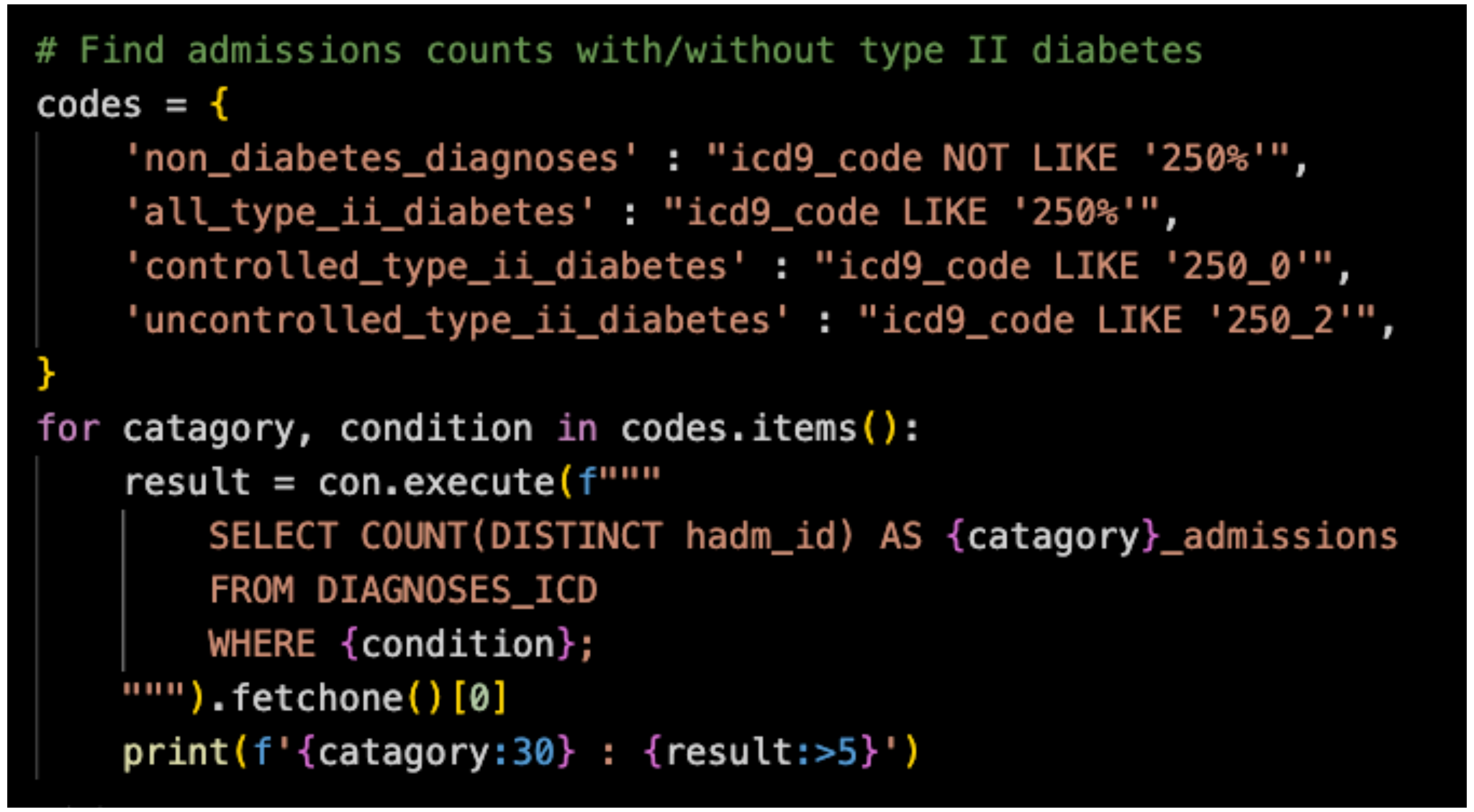
```
250**:  diabetes mellitus: first 3 digits is the diabetes code
250x*:      fourth digit specifies the type of complication
250*0:      fifth digit, 0, type II or unspecified type, not stated as uncontrolled
250*2:      fifth digit, 2, type II or unspecified type, uncontrolled
```

ICD9 Diabetes Admission Counts

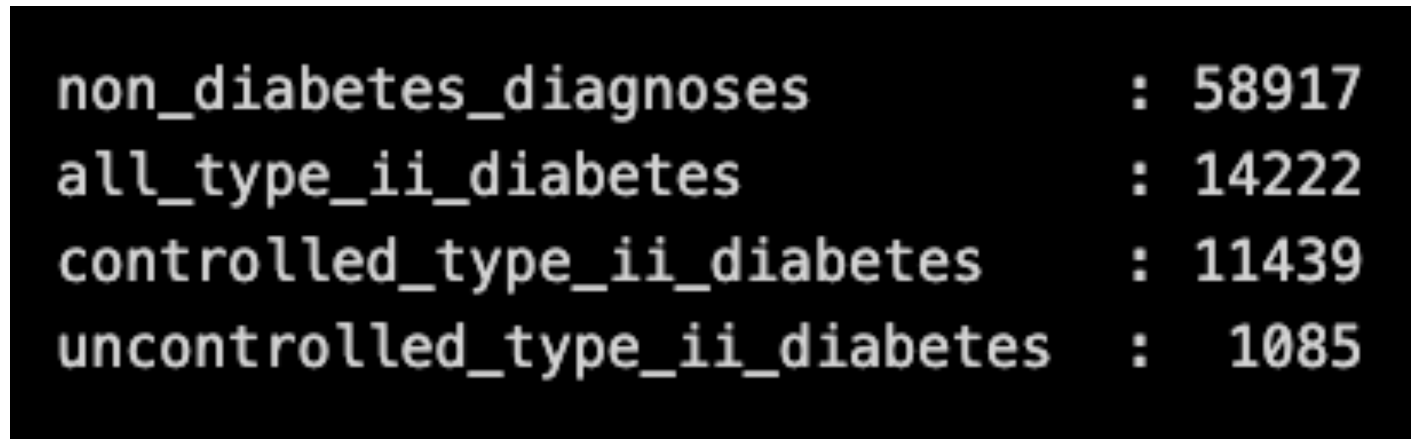
Almost 1/5 of ICU diagnoses are for diabetes mellitus.

A 7.6% of patients diagnosed with diabetes were not aware they were diabetic. I'm assuming here that uncontrolled diabetics were unaware, as Metformin is effective and inexpensive (see slide #.)

Search #3



```
# Find admissions counts with/without type II diabetes
codes = {
    'non_diabetes_diagnoses' : "icd9_code NOT LIKE '250%'",
    'all_type_ii_diabetes' : "icd9_code LIKE '250%'",
    'controlled_type_ii_diabetes' : "icd9_code LIKE '250_0'",
    'uncontrolled_type_ii_diabetes' : "icd9_code LIKE '250_2'",
}
for catagory, condition in codes.items():
    result = con.execute(f"""
        SELECT COUNT(DISTINCT hadm_id) AS {catagory}_admissions
        FROM DIAGNOSES_ICD
        WHERE {condition};
    """).fetchone()[0]
    print(f'{catagory:30} : {result:>5}')
```



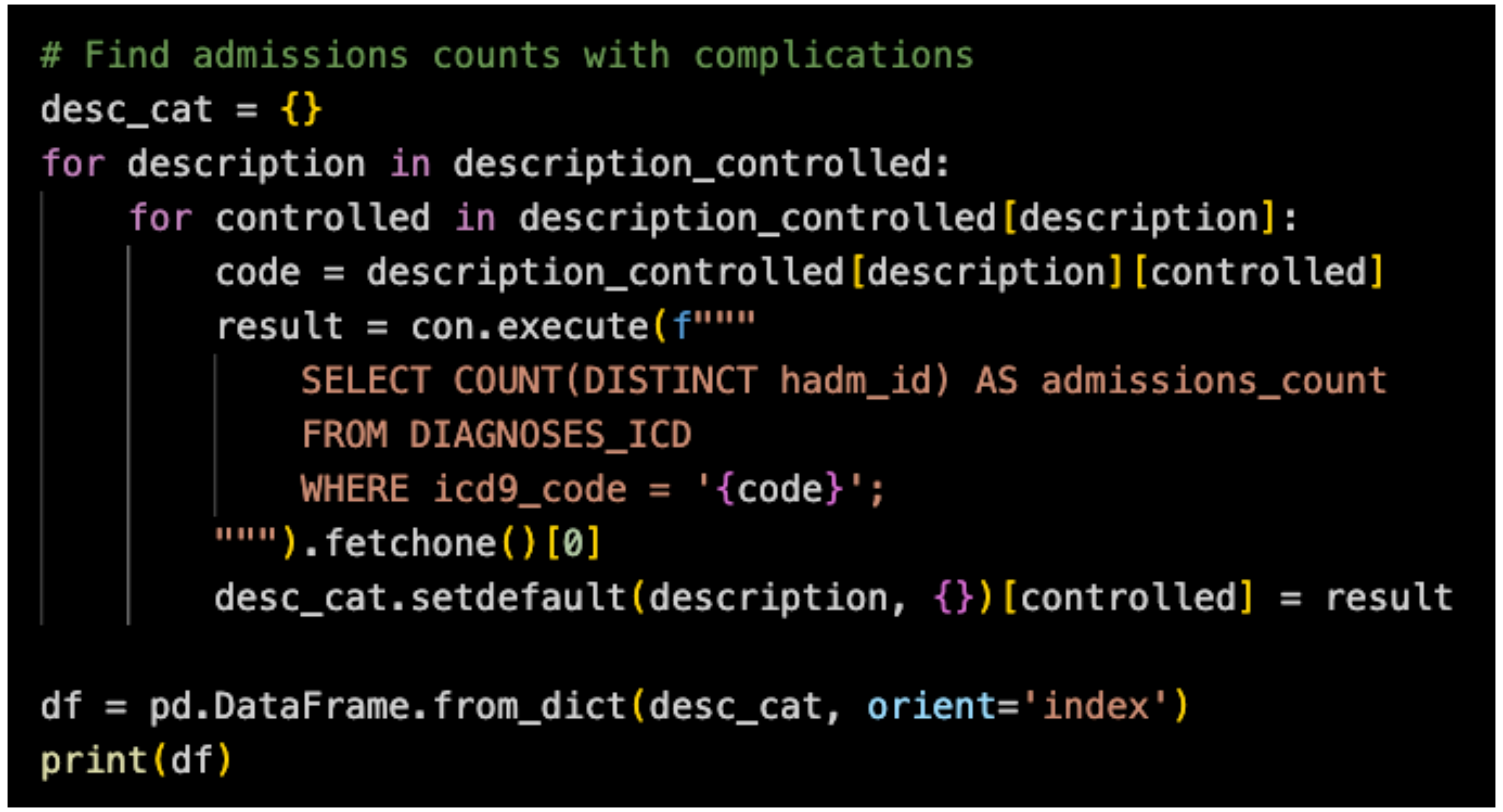
```
non_diabetes_diagnoses         : 58917
all_type_ii_diabetes           : 14222
controlled_type_ii_diabetes    : 11439
uncontrolled_type_ii_diabetes  :  1085
```

ICD9 Diabetes Counts with Complications

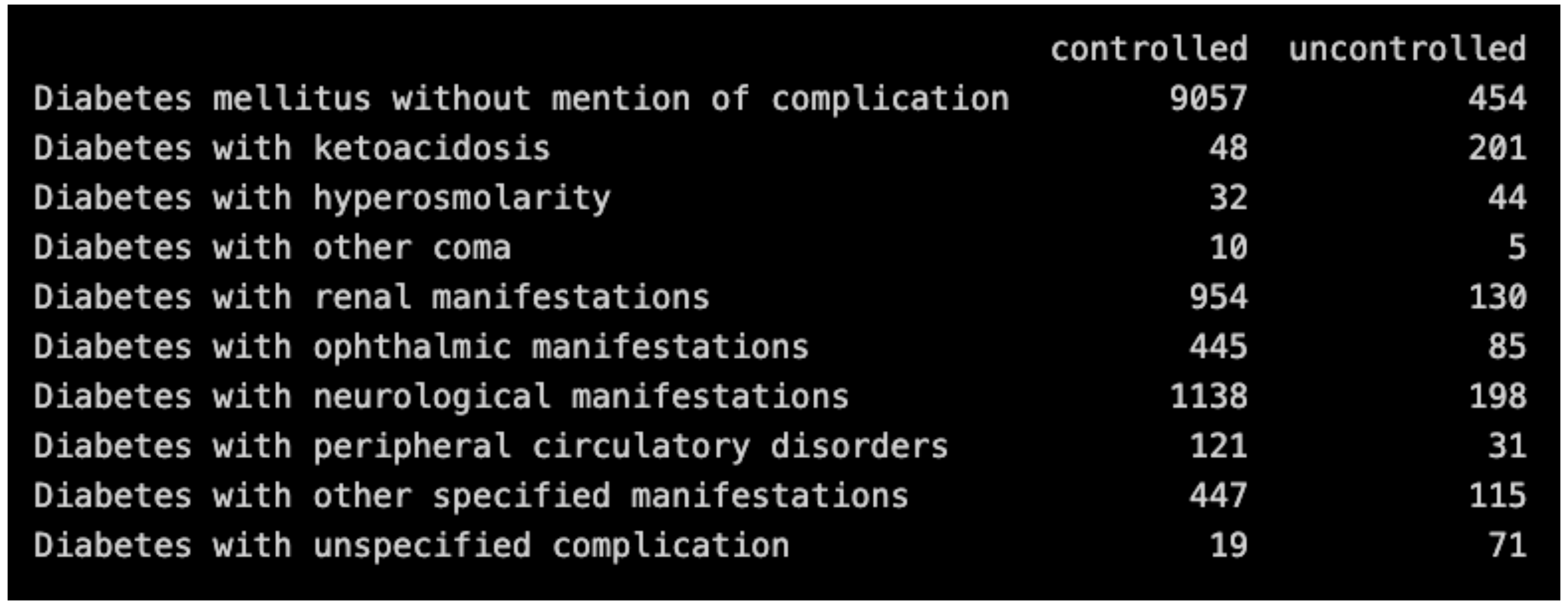
This search finds the counts of controlled and uncontrolled diabetes for each of the major complications indicated by the ICD9 code.

Just at first glance it is evident that risk of complications is much higher in uncontrolled diabetics. See next slide for pie charts of this data.

Search #4



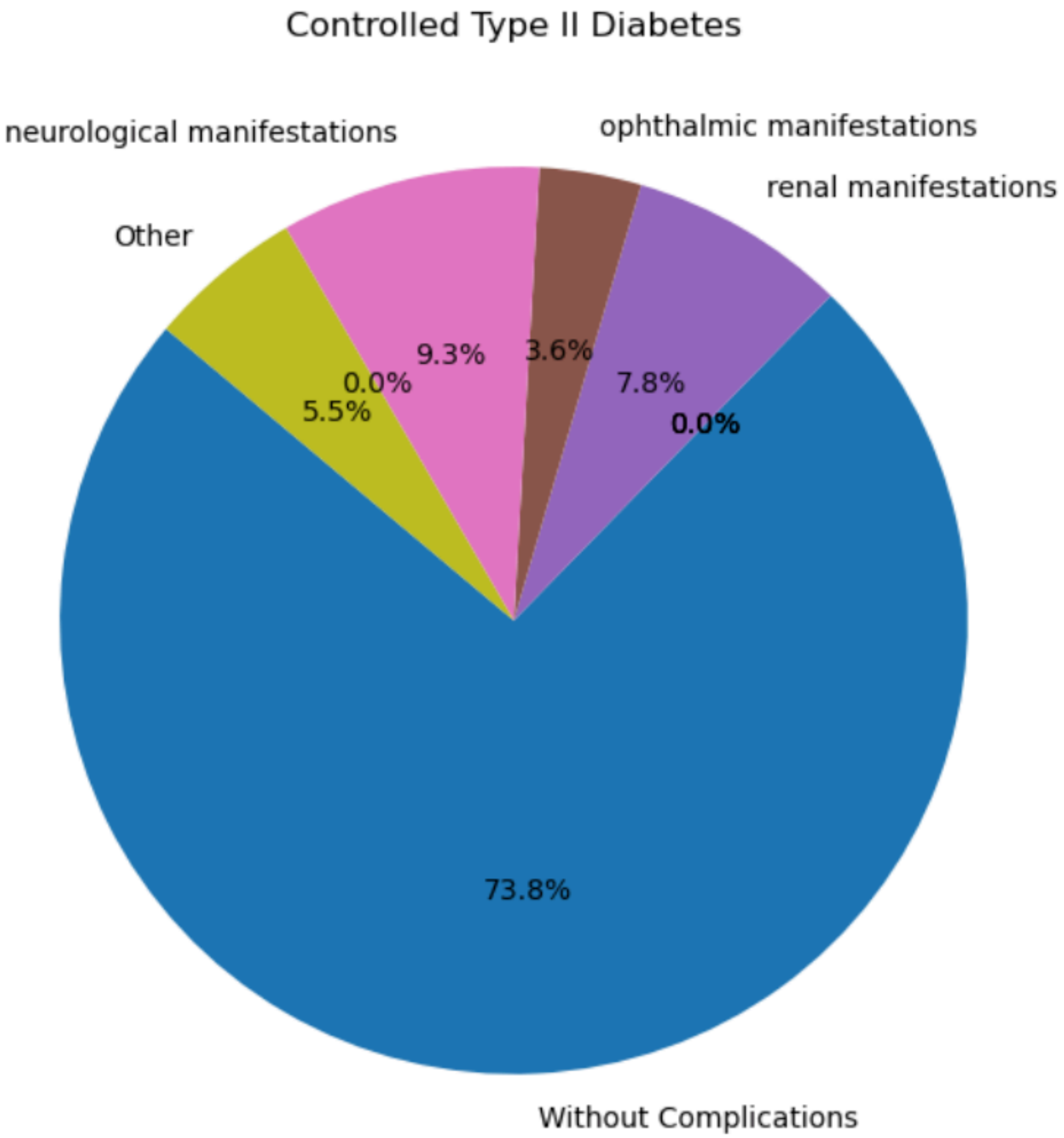
```python
# Find admissions counts with complications
desc_cat = {}
for description in description_controlled:
    for controlled in description_controlled[description]:
        code = description_controlled[description][controlled]
        result = con.execute(f"""
            SELECT COUNT(DISTINCT hadm_id) AS admissions_count
            FROM DIAGNOSES_ICD
            WHERE icd9_code = '{code}';
        """).fetchone()[0]
        desc_cat.setdefault(description, {})[controlled] = result

df = pd.DataFrame.from_dict(desc_cat, orient='index')
print(df)
```

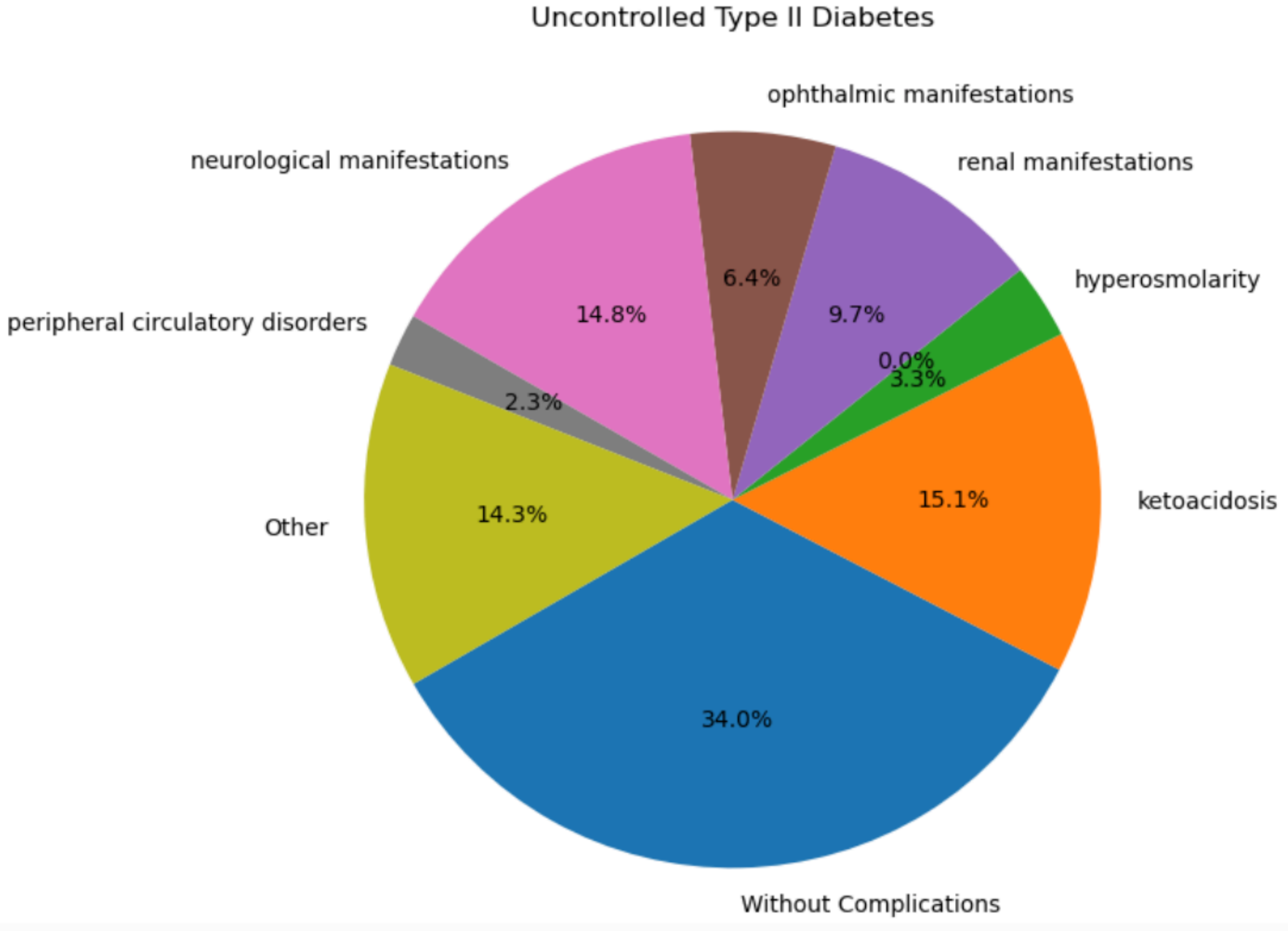


```
                                                  controlled  uncontrolled
Diabetes mellitus without mention of complication       9057           454
Diabetes with ketoacidosis                                48           201
Diabetes with hyperosmolarity                             32            44
Diabetes with other coma                                  10             5
Diabetes with renal manifestations                       954           130
Diabetes with ophthalmic manifestations                  445            85
Diabetes with neurological manifestations               1138           198
Diabetes with peripheral circulatory disorders           121            31
Diabetes with other specified manifestations             447           115
Diabetes with unspecified complication                    19            71
```

ICD9 Diabetes Complications

This shows complications increasing to nearly 2/3 of untreated diabetes. The most alarming slice, for diabetic ketoacidosis, is not present (less than 1%) in treated type II diabetics.

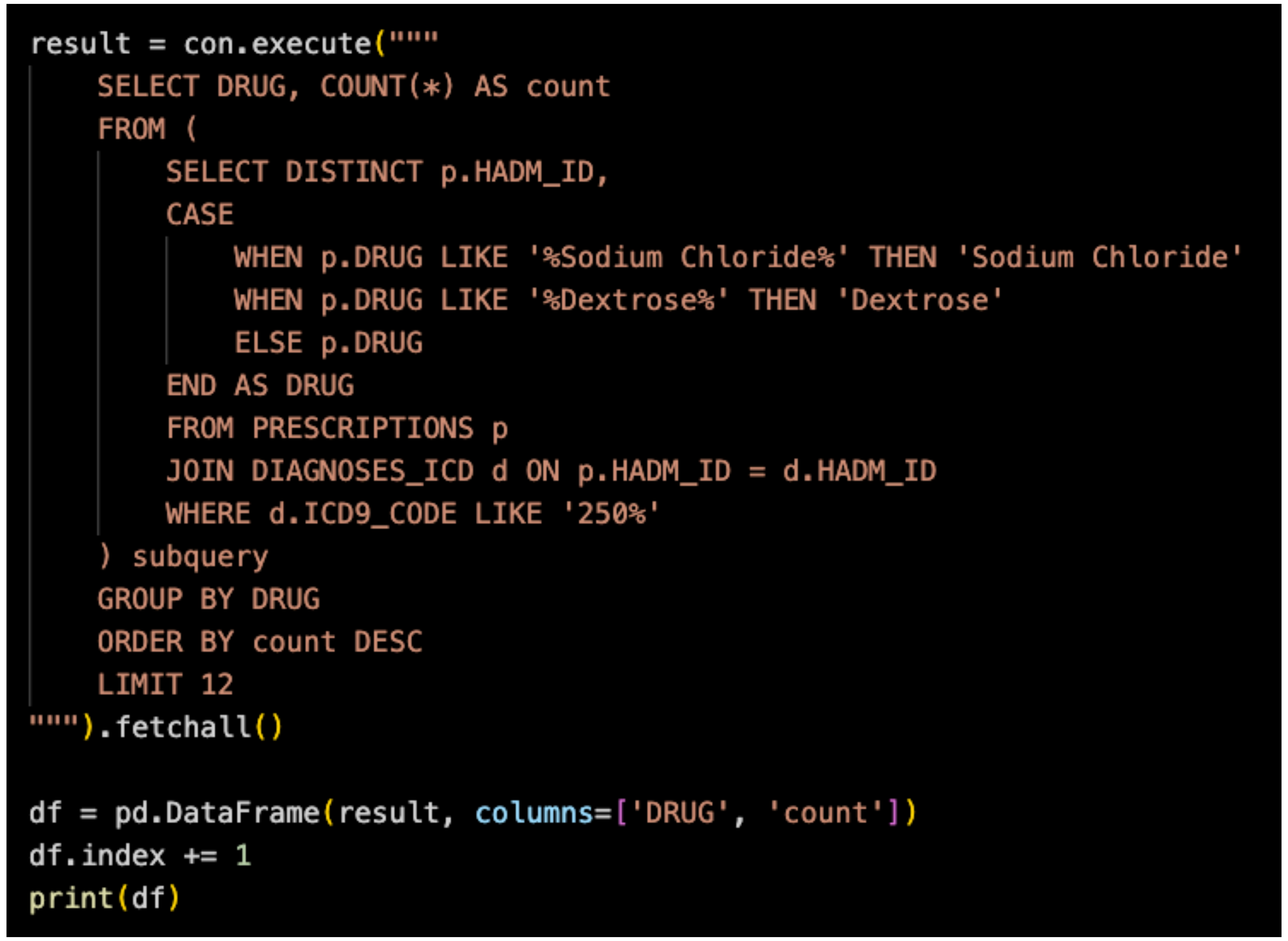

Controlled Type II Diabetes


Uncontrolled Type II Diabetes

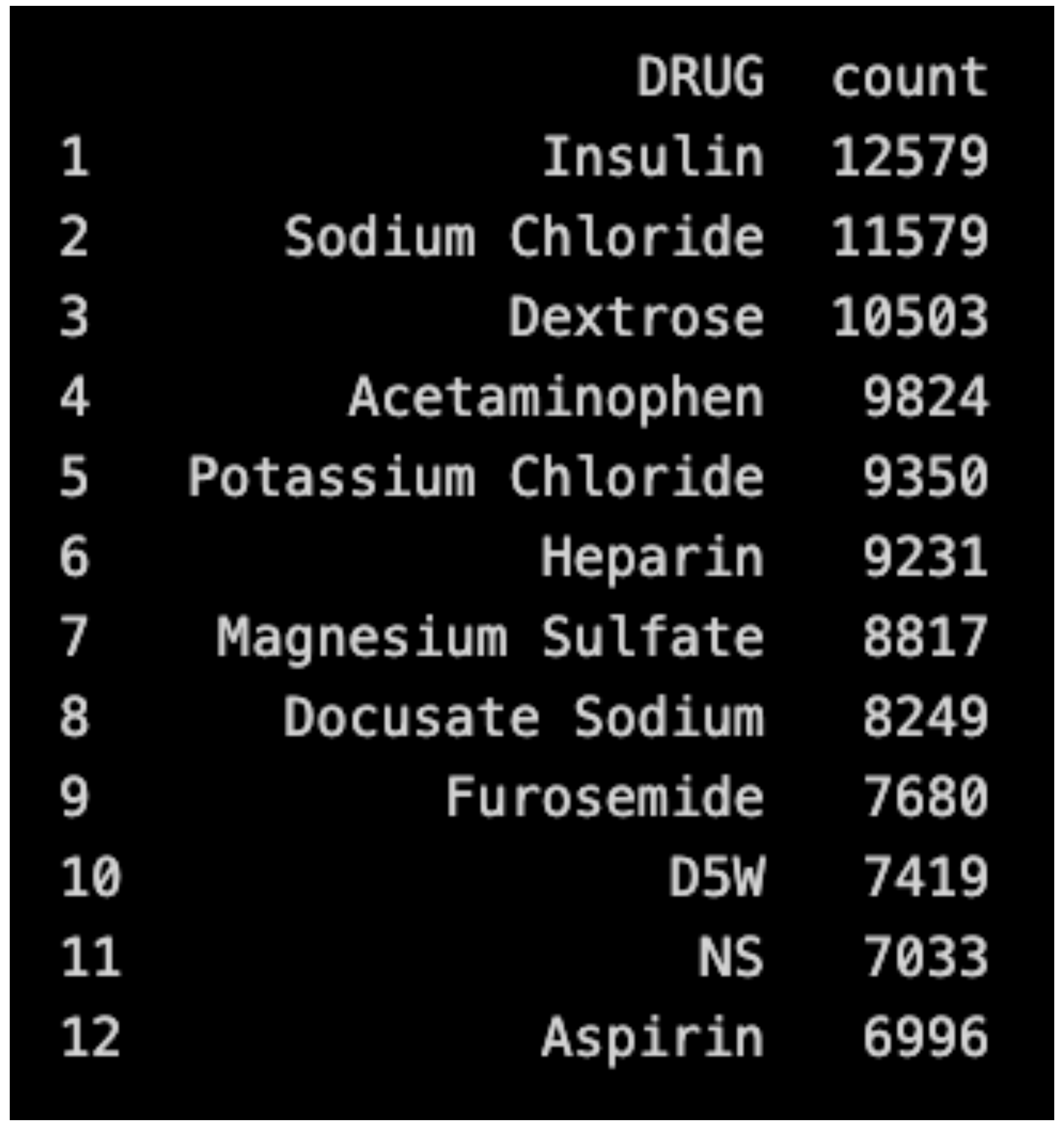
Most Common Drugs Administered to ICU Diabetic Patients

The top five are in simplified terms: insulin (what is lacking), salt (for dehydration), sugar to raise blood sugar, acetaminophen for pain and fever, potassium (replacing potassium in blood stream that enters cells with added insulin.) But I'm not a doctor! :D

Search #5



```
result = con.execute("""
    SELECT DRUG, COUNT(*) AS count
    FROM (
        SELECT DISTINCT p.HADM_ID,
        CASE
            WHEN p.DRUG LIKE '%Sodium Chloride%' THEN 'Sodium Chloride'
            WHEN p.DRUG LIKE '%Dextrose%' THEN 'Dextrose'
            ELSE p.DRUG
        END AS DRUG
        FROM PRESCRIPTIONS p
        JOIN DIAGNOSES_ICD d ON p.HADM_ID = d.HADM_ID
        WHERE d.ICD9_CODE LIKE '250%'
    ) subquery
    GROUP BY DRUG
    ORDER BY count DESC
    LIMIT 12
""").fetchall()

df = pd.DataFrame(result, columns=['DRUG', 'count'])
df.index += 1
print(df)
```



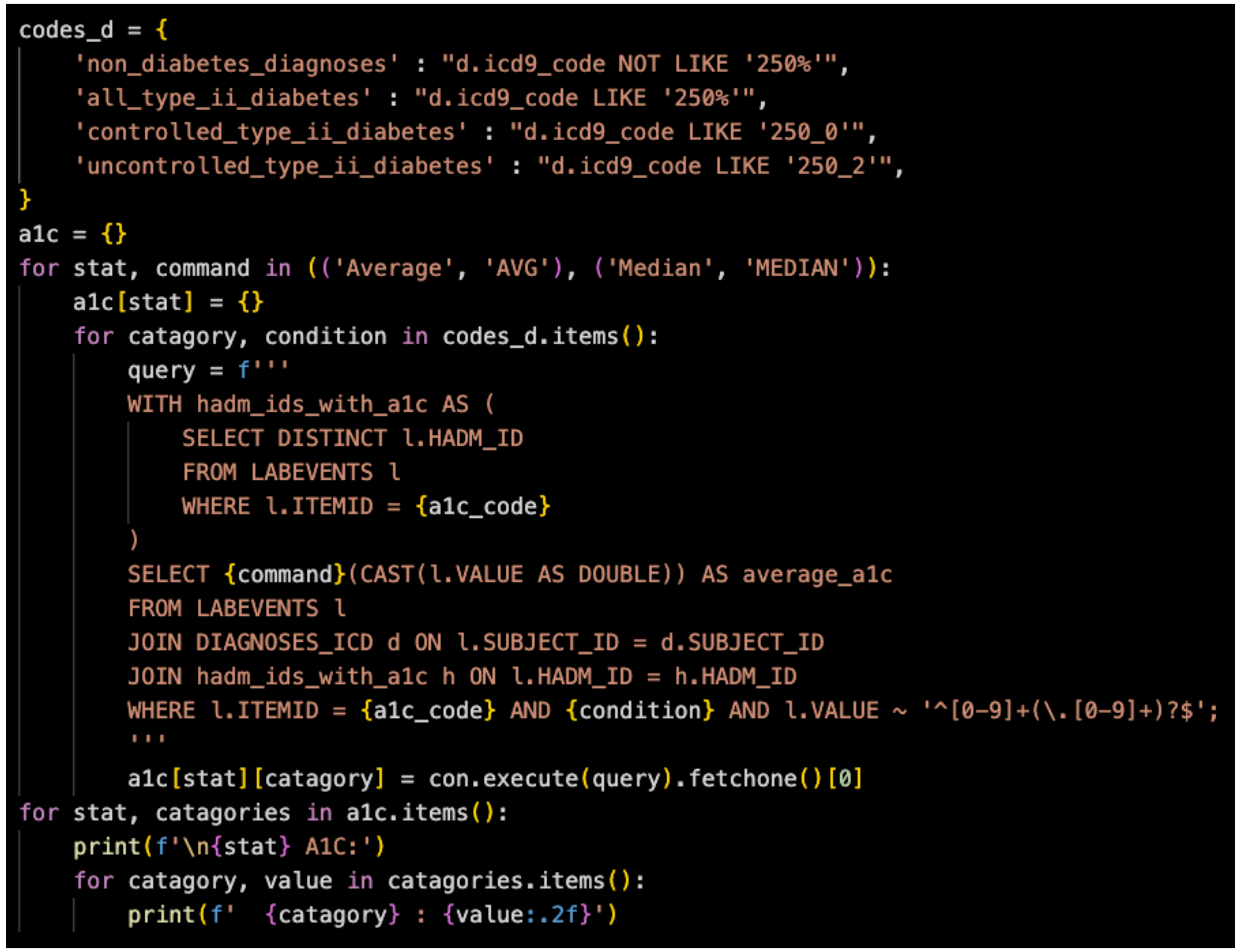
```
                 DRUG  count
1             Insulin  12579
2     Sodium Chloride  11579
3            Dextrose  10503
4       Acetaminophen   9824
5  Potassium Chloride   9350
6             Heparin   9231
7   Magnesium Sulfate   8817
8     Docusate Sodium   8249
9          Furosemide   7680
10                D5W   7419
11                 NS   7033
12            Aspirin   6996
```

Patients’ HbA1C Measurements

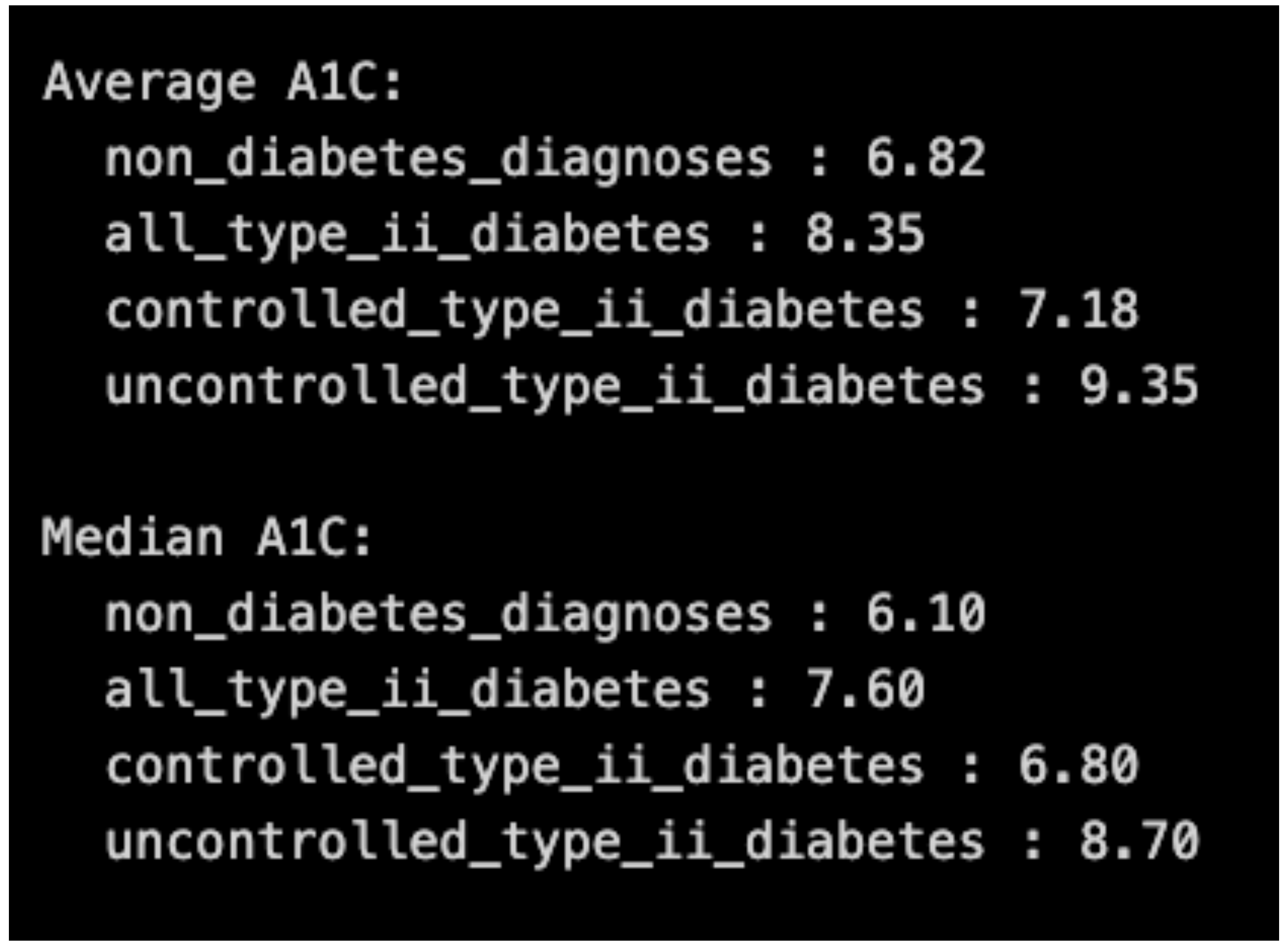
It’s clear that HbA1C (abbreviated, “a1c”) is much higher in undiagnosed (uncontrolled) diabetics. It’s also suggested that many people not diagnosed with diabetes are also in the diabetic range, and for some reason did not get a diabetes diagnosis code.

Note: Over half of ICU patients without a diabetic diagnosis are in the pre-diabetic range of A1C 5.7-6.4%.

Search #6



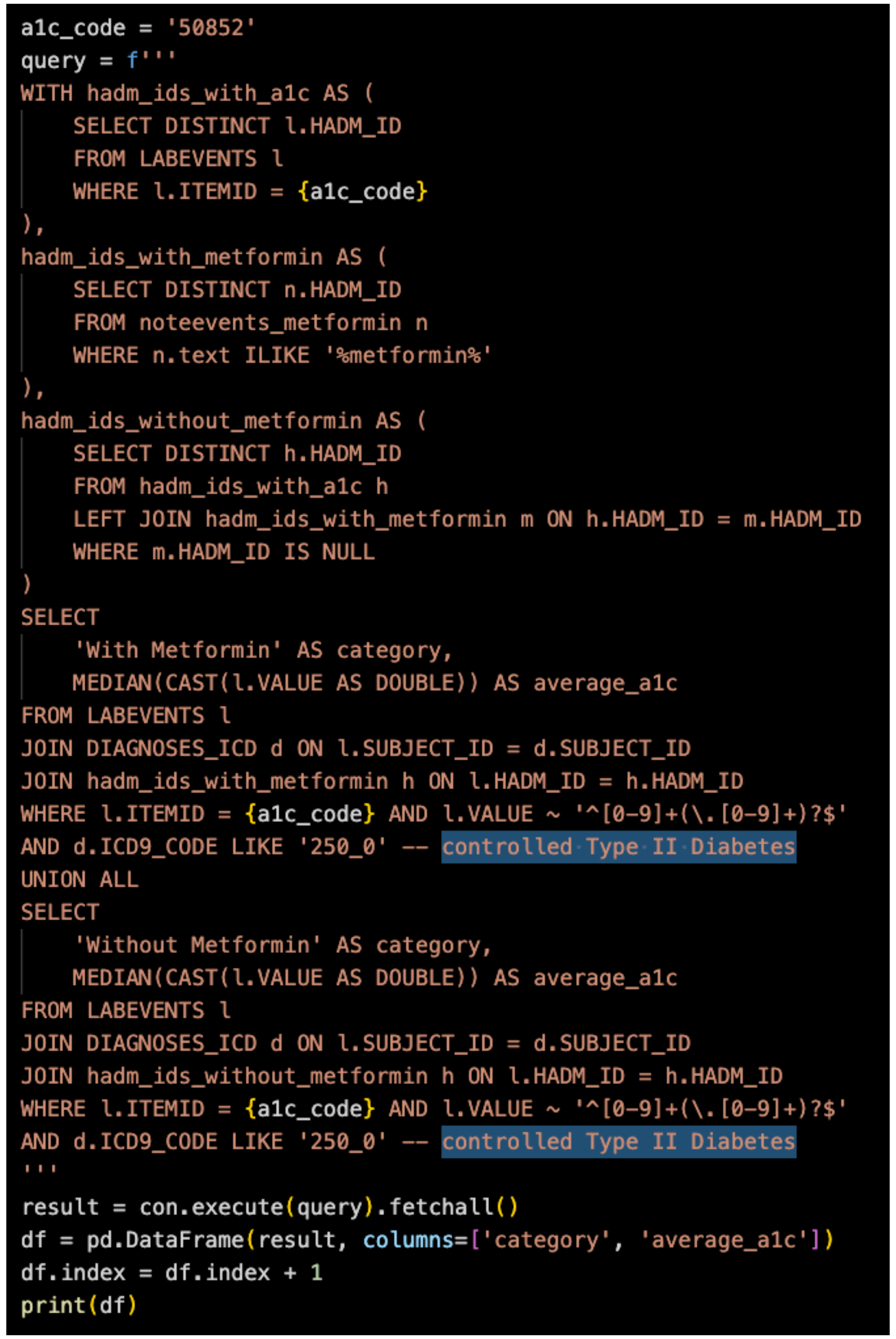
```
codes_d = {
    'non_diabetes_diagnoses' : "d.icd9_code NOT LIKE '250%'",
    'all_type_ii_diabetes' : "d.icd9_code LIKE '250%'",
    'controlled_type_ii_diabetes' : "d.icd9_code LIKE '250_0'",
    'uncontrolled_type_ii_diabetes' : "d.icd9_code LIKE '250_2'",
}
a1c = {}
for stat, command in (('Average', 'AVG'), ('Median', 'MEDIAN')):
    a1c[stat] = {}
    for catagory, condition in codes_d.items():
        query = f'''
        WITH hadm_ids_with_a1c AS (
            SELECT DISTINCT l.HADM_ID
            FROM LABEVENTS l
            WHERE l.ITEMID = {a1c_code}
        )
        SELECT {command}(CAST(l.VALUE AS DOUBLE)) AS average_a1c
        FROM LABEVENTS l
        JOIN DIAGNOSES_ICD d ON l.SUBJECT_ID = d.SUBJECT_ID
        JOIN hadm_ids_with_a1c h ON l.HADM_ID = h.HADM_ID
        WHERE l.ITEMID = {a1c_code} AND {condition} AND l.VALUE ~ '^[0-9]+(\.[0-9]+)?$';
        '''
        a1c[stat][catagory] = con.execute(query).fetchone()[0]
for stat, catagories in a1c.items():
    print(f'\n{stat} A1C:')
    for catagory, value in catagories.items():
        print(f'  {catagory} : {value:.2f}')
```



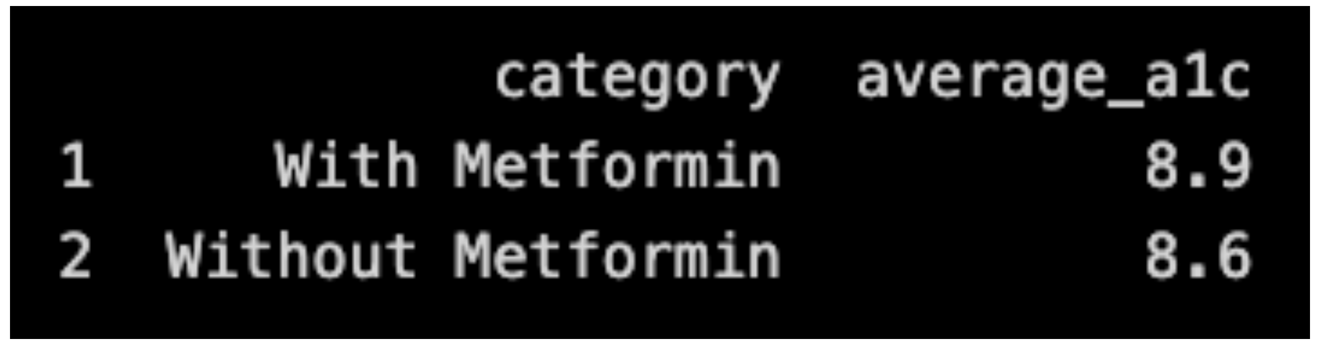
```
Average A1C:
  non_diabetes_diagnoses : 6.82
  all_type_ii_diabetes : 8.35
  controlled_type_ii_diabetes : 7.18
  uncontrolled_type_ii_diabetes : 9.35

Median A1C:
  non_diabetes_diagnoses : 6.10
  all_type_ii_diabetes : 7.60
  controlled_type_ii_diabetes : 6.80
  uncontrolled_type_ii_diabetes : 8.70
```

Patients That Discussed Metformin

Search #7



```
a1c_code = '50852'
query = f'''
WITH hadm_ids_with_a1c AS (
    SELECT DISTINCT l.HADM_ID
    FROM LABEVENTS l
    WHERE l.ITEMID = {a1c_code}
),
hadm_ids_with_metformin AS (
    SELECT DISTINCT n.HADM_ID
    FROM noteevents_metformin n
    WHERE n.text ILIKE '%metformin%'
),
hadm_ids_without_metformin AS (
    SELECT DISTINCT h.HADM_ID
    FROM hadm_ids_with_a1c h
    LEFT JOIN hadm_ids_with_metformin m ON h.HADM_ID = m.HADM_ID
    WHERE m.HADM_ID IS NULL
)
SELECT
    'With Metformin' AS category,
    MEDIAN(CAST(l.VALUE AS DOUBLE)) AS average_a1c
FROM LABEVENTS l
JOIN DIAGNOSES_ICD d ON l.SUBJECT_ID = d.SUBJECT_ID
JOIN hadm_ids_with_metformin h ON l.HADM_ID = h.HADM_ID
WHERE l.ITEMID = {a1c_code} AND l.VALUE ~ '^[0-9]+(\.[0-9]+)?$'
AND d.ICD9_CODE LIKE '250_0' -- controlled Type II Diabetes
UNION ALL
SELECT
    'Without Metformin' AS category,
    MEDIAN(CAST(l.VALUE AS DOUBLE)) AS average_a1c
FROM LABEVENTS l
JOIN DIAGNOSES_ICD d ON l.SUBJECT_ID = d.SUBJECT_ID
JOIN hadm_ids_without_metformin h ON l.HADM_ID = h.HADM_ID
WHERE l.ITEMID = {a1c_code} AND l.VALUE ~ '^[0-9]+(\.[0-9]+)?$'
AND d.ICD9_CODE LIKE '250_0' -- controlled Type II Diabetes
'''
result = con.execute(query).fetchall()
df = pd.DataFrame(result, columns=['category', 'average_a1c'])
df.index = df.index + 1
print(df)
```



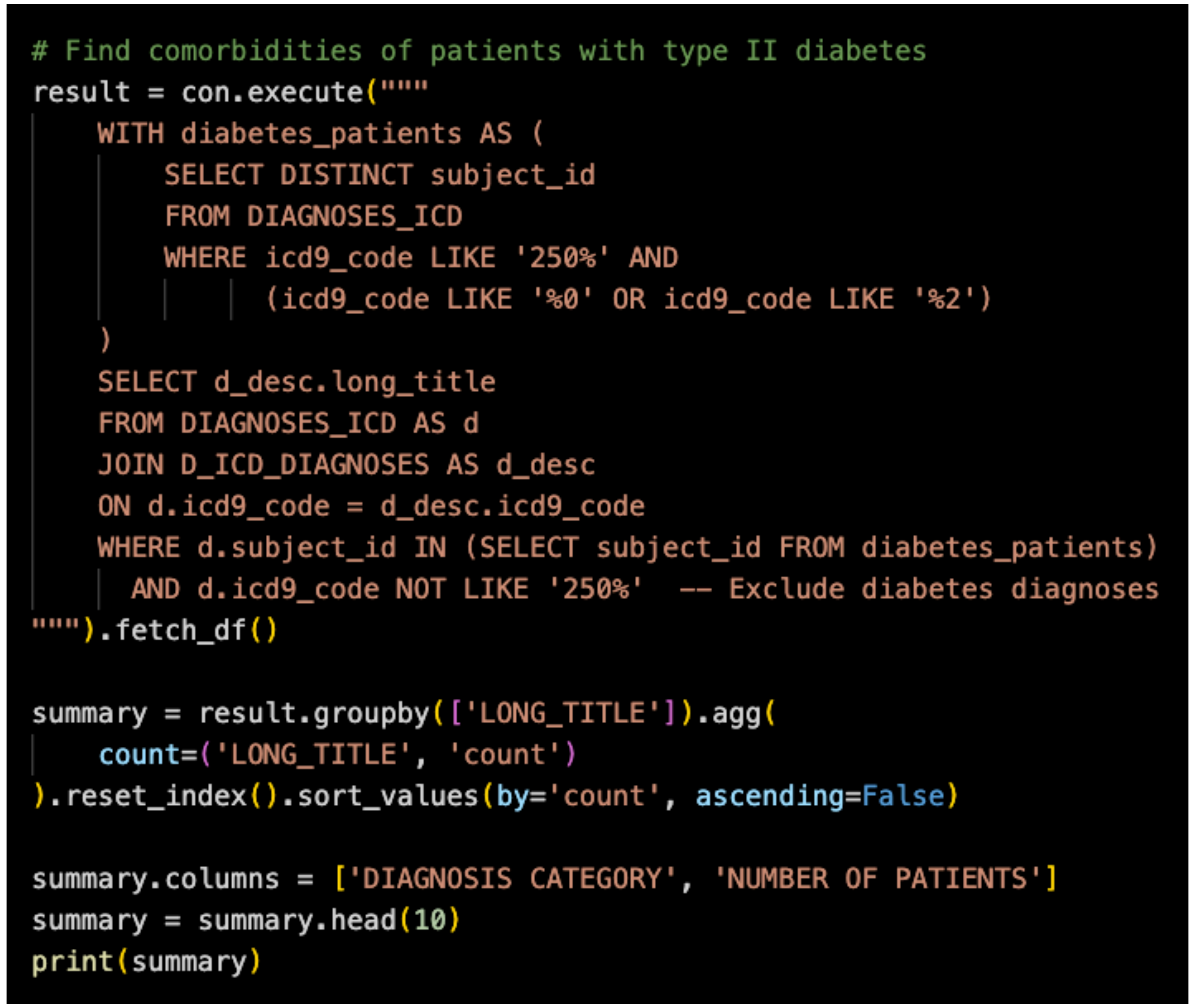
```
            category  average_a1c
1     With Metformin          8.9
2  Without Metformin          8.6
```

This search compares average measured A1C in patients that have the drug “metformin” mentioned in their notes with the implication that likely these patients mentioned that they are currently taking Metformin.

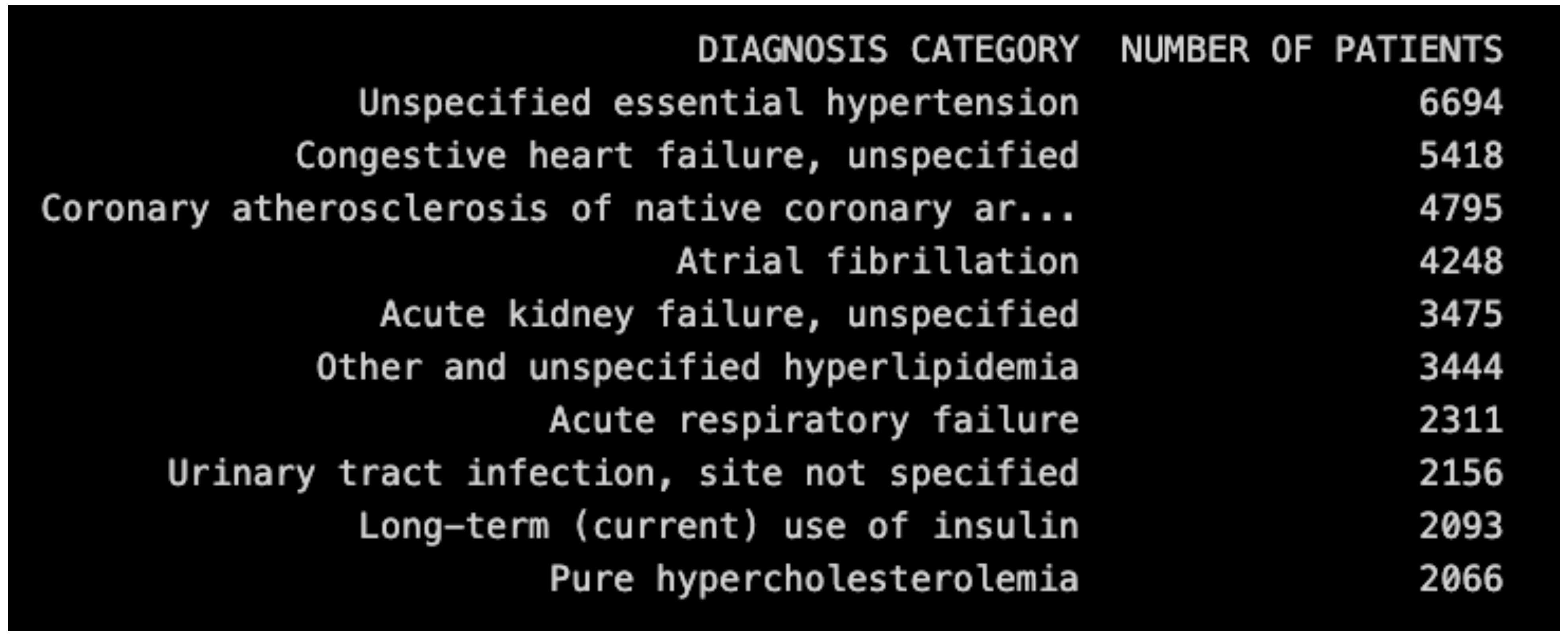
This search covers controlled diabetic diagnoses, so maybe the Metformin group has a higher average A1C compared to the non-metformin group already using more powerful drugs.

Comorbidities of Patients with type II diabetes

Search #8



```
# Find comorbidities of patients with type II diabetes
result = con.execute("""
    WITH diabetes_patients AS (
        SELECT DISTINCT subject_id
        FROM DIAGNOSES_ICD
        WHERE icd9_code LIKE '250%' AND
              (icd9_code LIKE '%0' OR icd9_code LIKE '%2')
    )
    SELECT d_desc.long_title
    FROM DIAGNOSES_ICD AS d
    JOIN D_ICD_DIAGNOSES AS d_desc
    ON d.icd9_code = d_desc.icd9_code
    WHERE d.subject_id IN (SELECT subject_id FROM diabetes_patients)
      AND d.icd9_code NOT LIKE '250%'  -- Exclude diabetes diagnoses
""").fetch_df()

summary = result.groupby(['LONG_TITLE']).agg(
    count=('LONG_TITLE', 'count')
).reset_index().sort_values(by='count', ascending=False)

summary.columns = ['DIAGNOSIS CATEGORY', 'NUMBER OF PATIENTS']
summary = summary.head(10)
print(summary)
```

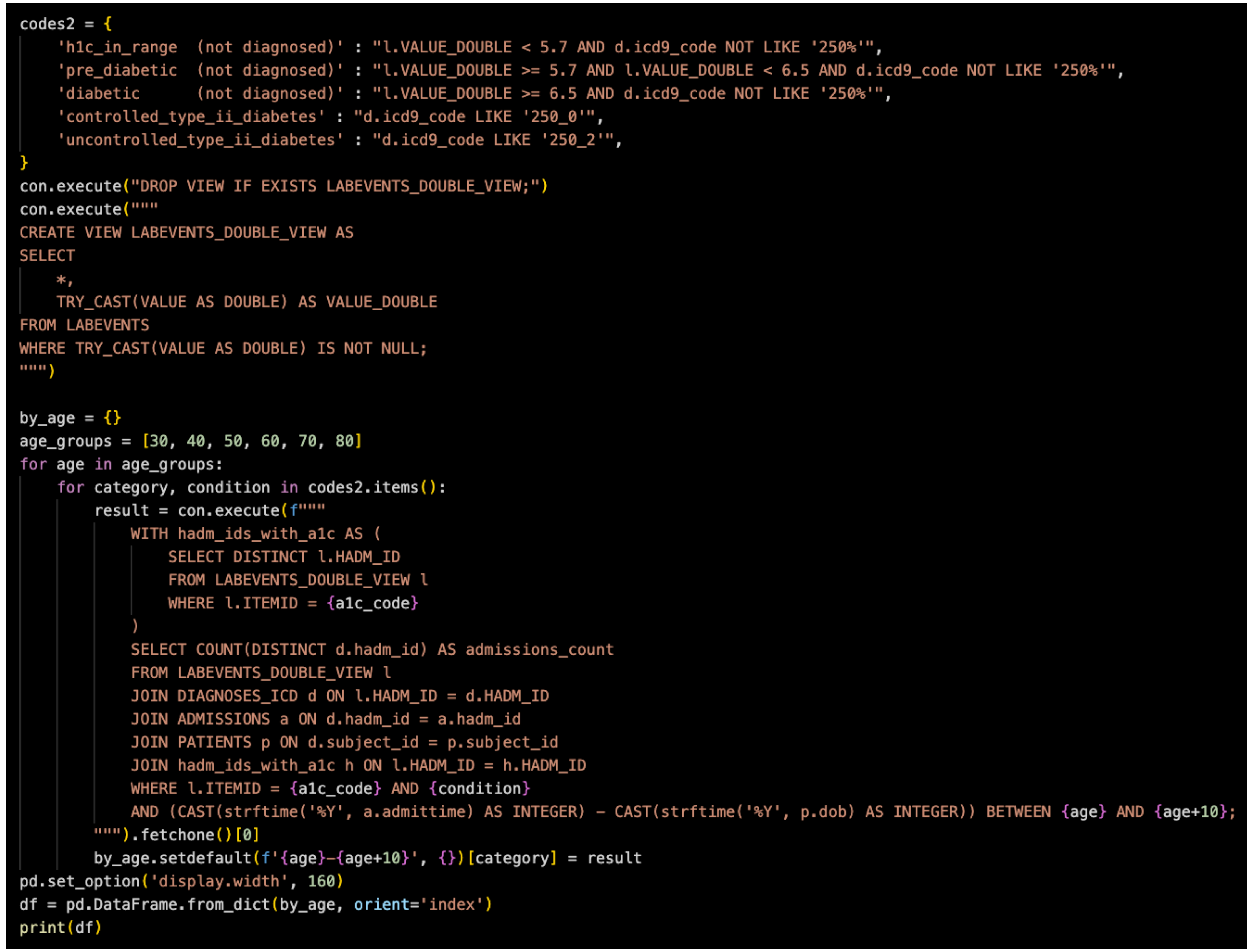


```
                           DIAGNOSIS CATEGORY  NUMBER OF PATIENTS
             Unspecified essential hypertension                6694
          Congestive heart failure, unspecified                5418
Coronary atherosclerosis of native coronary ar...               4795
                             Atrial fibrillation                4248
             Acute kidney failure, unspecified                 3475
            Other and unspecified hyperlipidemia               3444
                     Acute respiratory failure                 2311
     Urinary tract infection, site not specified               2156
             Long-term (current) use of insulin                2093
                    Pure hypercholesterolemia                  2066
```

Search for most common comorbidities of type 2 diabetics.

Diabetes Diagnosis Categories by Age Groups

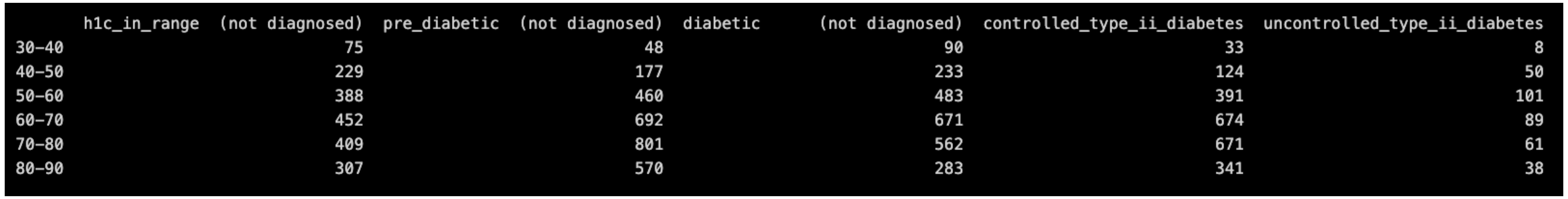
Search #9



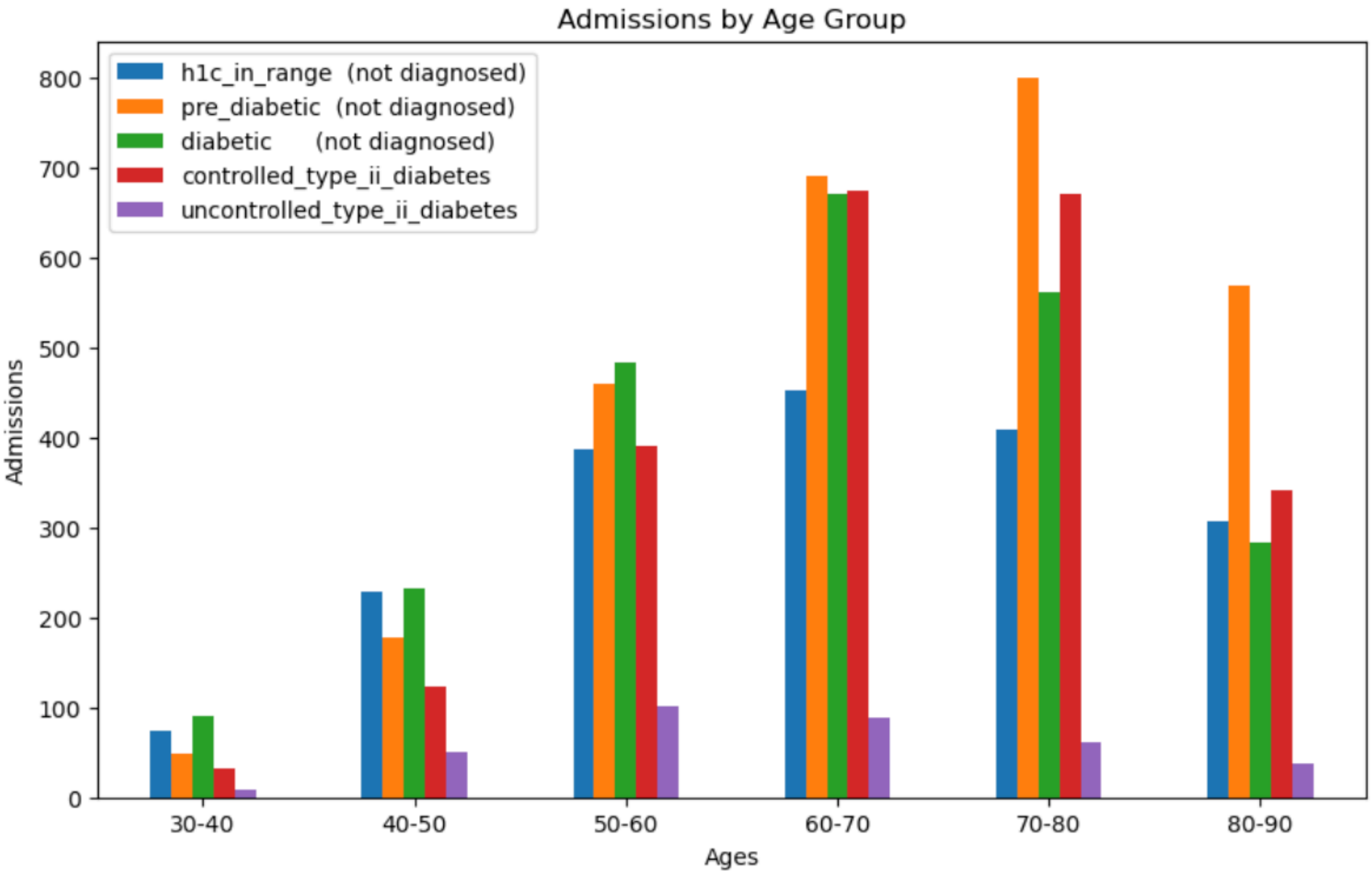
```
codes2 = {
    'h1c_in_range  (not diagnosed)' : "l.VALUE_DOUBLE < 5.7 AND d.icd9_code NOT LIKE '250%'",
    'pre_diabetic  (not diagnosed)' : "l.VALUE_DOUBLE >= 5.7 AND l.VALUE_DOUBLE < 6.5 AND d.icd9_code NOT LIKE '250%'",
    'diabetic      (not diagnosed)' : "l.VALUE_DOUBLE >= 6.5 AND d.icd9_code NOT LIKE '250%'",
    'controlled_type_ii_diabetes' : "d.icd9_code LIKE '250_0'",
    'uncontrolled_type_ii_diabetes' : "d.icd9_code LIKE '250_2'",
}
con.execute("DROP VIEW IF EXISTS LABEVENTS_DOUBLE_VIEW;")
con.execute("""
CREATE VIEW LABEVENTS_DOUBLE_VIEW AS
SELECT
    *,
    TRY_CAST(VALUE AS DOUBLE) AS VALUE_DOUBLE
FROM LABEVENTS
WHERE TRY_CAST(VALUE AS DOUBLE) IS NOT NULL;
""")

by_age = {}
age_groups = [30, 40, 50, 60, 70, 80]
for age in age_groups:
    for category, condition in codes2.items():
        result = con.execute(f"""
            WITH hadm_ids_with_a1c AS (
                SELECT DISTINCT l.HADM_ID
                FROM LABEVENTS_DOUBLE_VIEW l
                WHERE l.ITEMID = {a1c_code}
            )
            SELECT COUNT(DISTINCT d.hadm_id) AS admissions_count
            FROM LABEVENTS_DOUBLE_VIEW l
            JOIN DIAGNOSES_ICD d ON l.HADM_ID = d.HADM_ID
            JOIN ADMISSIONS a ON d.hadm_id = a.hadm_id
            JOIN PATIENTS p ON d.subject_id = p.subject_id
            JOIN hadm_ids_with_a1c h ON l.HADM_ID = h.HADM_ID
            WHERE l.ITEMID = {a1c_code} AND {condition}
            AND (CAST(strftime('%Y', a.admittime) AS INTEGER) - CAST(strftime('%Y', p.dob) AS INTEGER)) BETWEEN {age} AND {age+10};
        """).fetchone()[0]
        by_age.setdefault(f'{age}-{age+10}', {})[category] = result
pd.set_option('display.width', 160)
df = pd.DataFrame.from_dict(by_age, orient='index')
print(df)
```

Searching for diabetes diagnosis by age group. Also including “undiagnosed as diabetes” yet with A1C measured in the diabetic range.

Bar graph of this data on next slide.

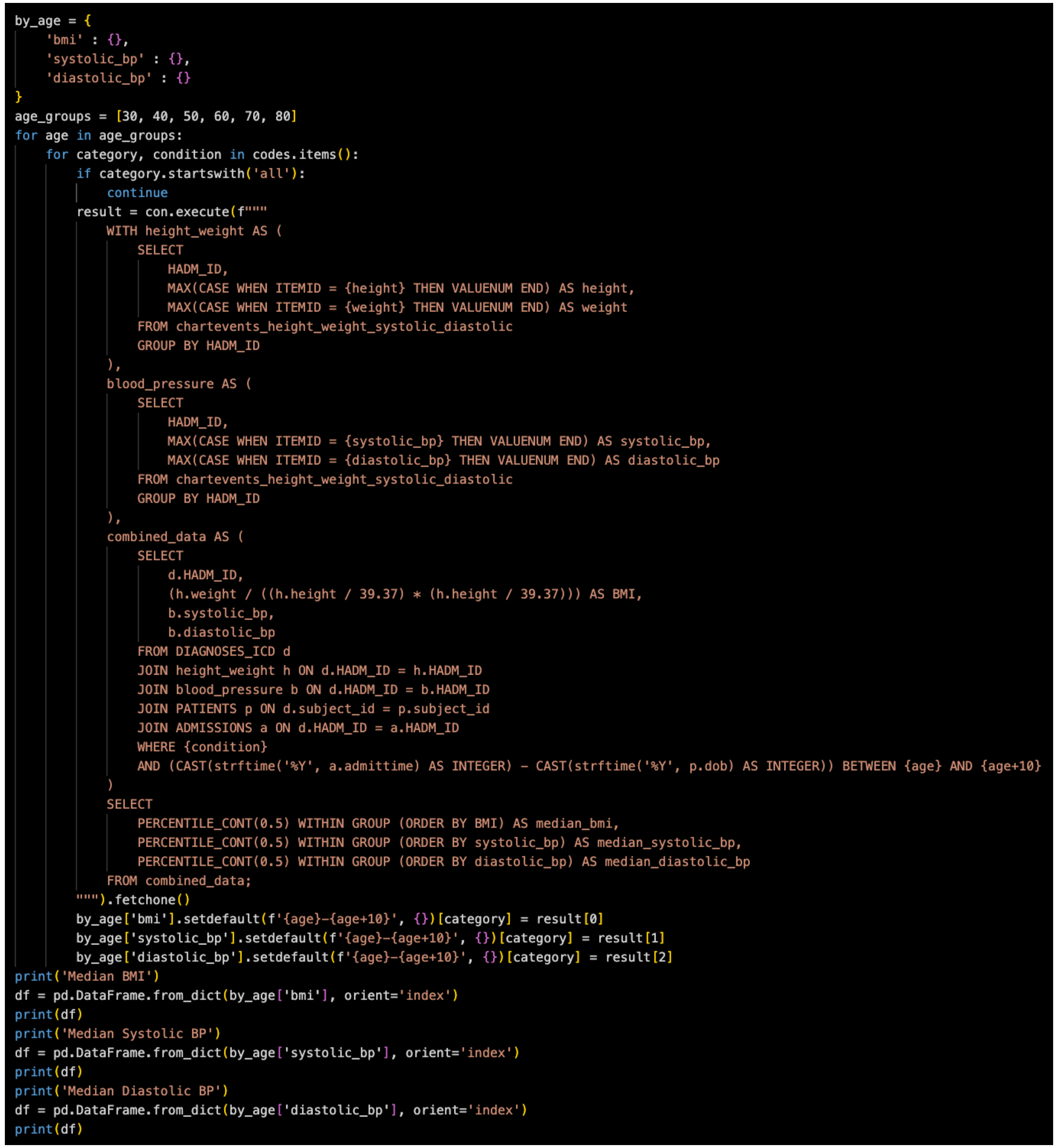


```
       h1c_in_range  (not diagnosed)  pre_diabetic  (not diagnosed)  diabetic      (not diagnosed)  controlled_type_ii_diabetes  uncontrolled_type_ii_diabetes
30-40                             75                             48                             90                           33                              8
40-50                            229                            177                            233                          124                             50
50-60                            388                            460                            483                          391                            101
60-70                            452                            692                            671                          674                             89
70-80                            409                            801                            562                          671                             61
80-90                            307                            570                            283                          341                             38
```


Admissions by Age Group
h1c_in_range (not diagnosed)
pre_diabetic (not diagnosed)
diabetic (not diagnosed)
controlled_type_ii_diabetes
uncontrolled_type_ii_diabetes
Admissions
0
100
200
300
400
500
600
700
800
30-40
40-50
50-60
60-70
70-80
80-90
Ages

BMI and Blood Pressure of Diabetics

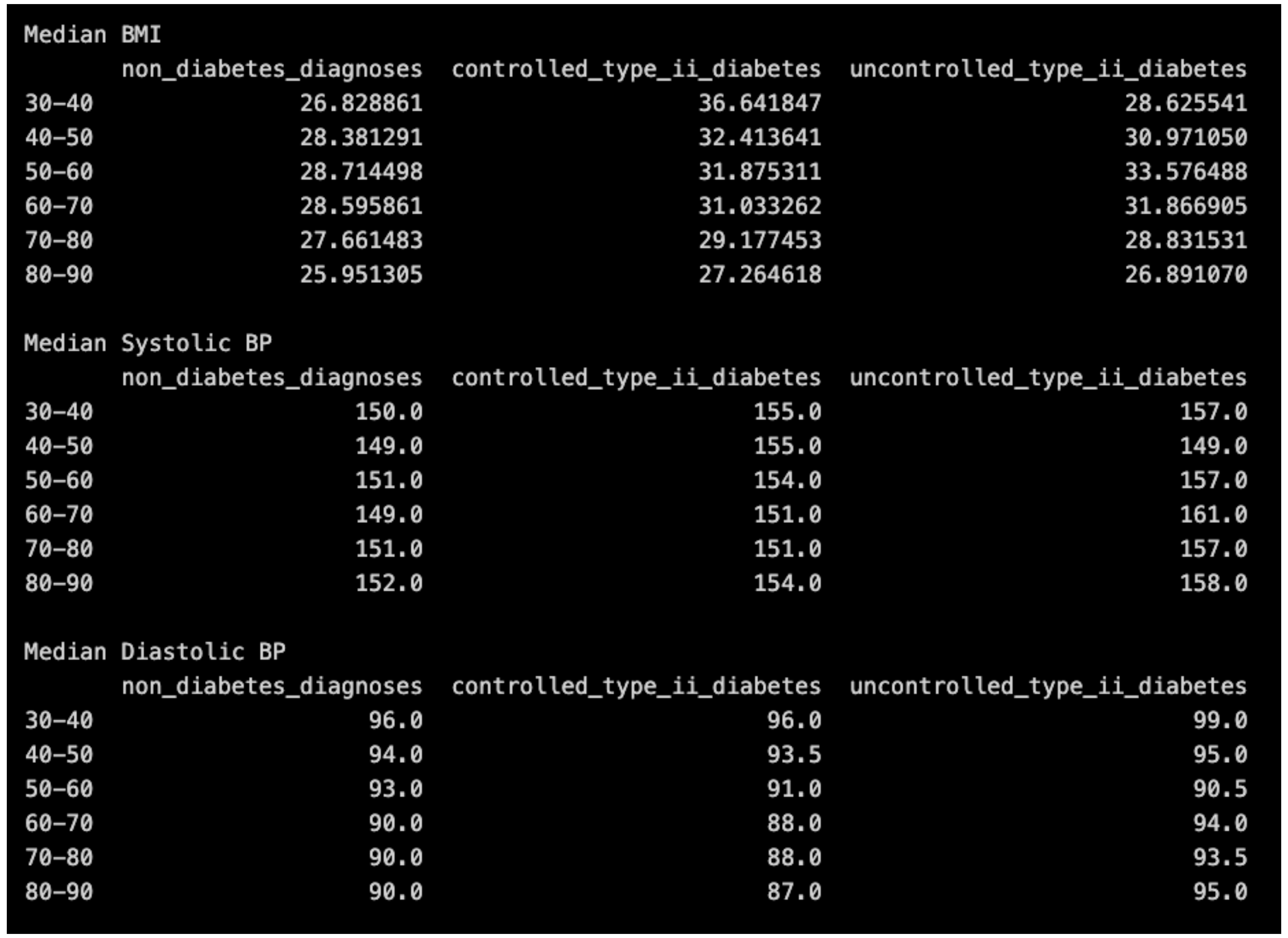
Search #10



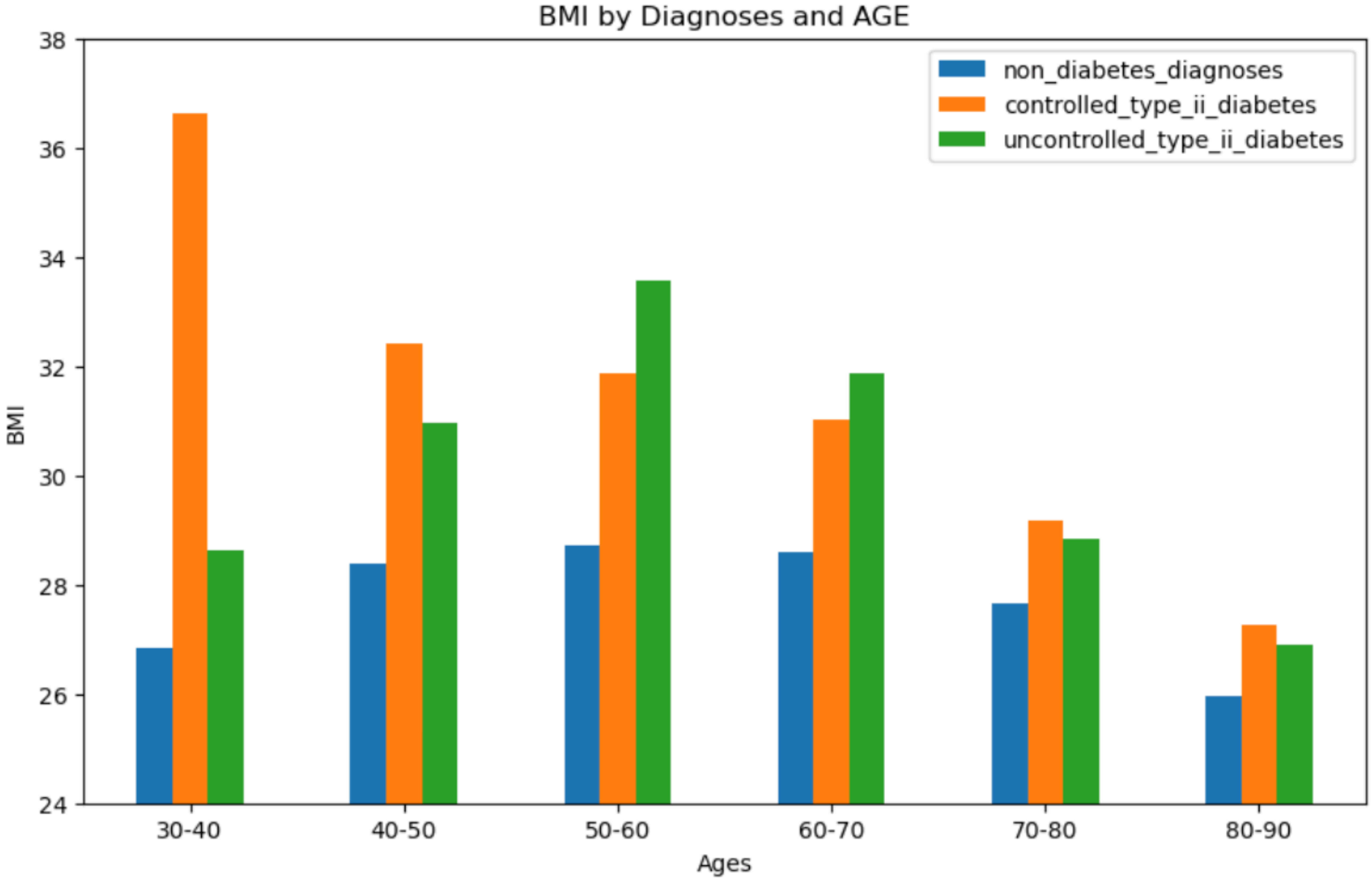
```
by_age = {
    'bmi' : {},
    'systolic_bp' : {},
    'diastolic_bp' : {}
}
age_groups = [30, 40, 50, 60, 70, 80]
for age in age_groups:
    for category, condition in codes.items():
        if category.startswith('all'):
            continue
        result = con.execute(f"""
            WITH height_weight AS (
                SELECT
                    HADM_ID,
                    MAX(CASE WHEN ITEMID = {height} THEN VALUENUM END) AS height,
                    MAX(CASE WHEN ITEMID = {weight} THEN VALUENUM END) AS weight
                FROM chartevents_height_weight_systolic_diastolic
                GROUP BY HADM_ID
            ),
            blood_pressure AS (
                SELECT
                    HADM_ID,
                    MAX(CASE WHEN ITEMID = {systolic_bp} THEN VALUENUM END) AS systolic_bp,
                    MAX(CASE WHEN ITEMID = {diastolic_bp} THEN VALUENUM END) AS diastolic_bp
                FROM chartevents_height_weight_systolic_diastolic
                GROUP BY HADM_ID
            ),
            combined_data AS (
                SELECT
                    d.HADM_ID,
                    (h.weight / ((h.height / 39.37) * (h.height / 39.37))) AS BMI,
                    b.systolic_bp,
                    b.diastolic_bp
                FROM DIAGNOSES_ICD d
                JOIN height_weight h ON d.HADM_ID = h.HADM_ID
                JOIN blood_pressure b ON d.HADM_ID = b.HADM_ID
                JOIN PATIENTS p ON d.subject_id = p.subject_id
                JOIN ADMISSIONS a ON d.HADM_ID = a.HADM_ID
                WHERE {condition}
                AND (CAST(strftime('%Y', a.admittime) AS INTEGER) - CAST(strftime('%Y', p.dob) AS INTEGER)) BETWEEN {age} AND {age+10}
            )
            SELECT
                PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY BMI) AS median_bmi,
                PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY systolic_bp) AS median_systolic_bp,
                PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY diastolic_bp) AS median_diastolic_bp
            FROM combined_data;
        """).fetchone()
        by_age['bmi'].setdefault(f'{age}-{age+10}', {})[category] = result[0]
        by_age['systolic_bp'].setdefault(f'{age}-{age+10}', {})[category] = result[1]
        by_age['diastolic_bp'].setdefault(f'{age}-{age+10}', {})[category] = result[2]
print('Median BMI')
df = pd.DataFrame.from_dict(by_age['bmi'], orient='index')
print(df)
print('Median Systolic BP')
df = pd.DataFrame.from_dict(by_age['systolic_bp'], orient='index')
print(df)
print('Median Diastolic BP')
df = pd.DataFrame.from_dict(by_age['diastolic_bp'], orient='index')
print(df)
```

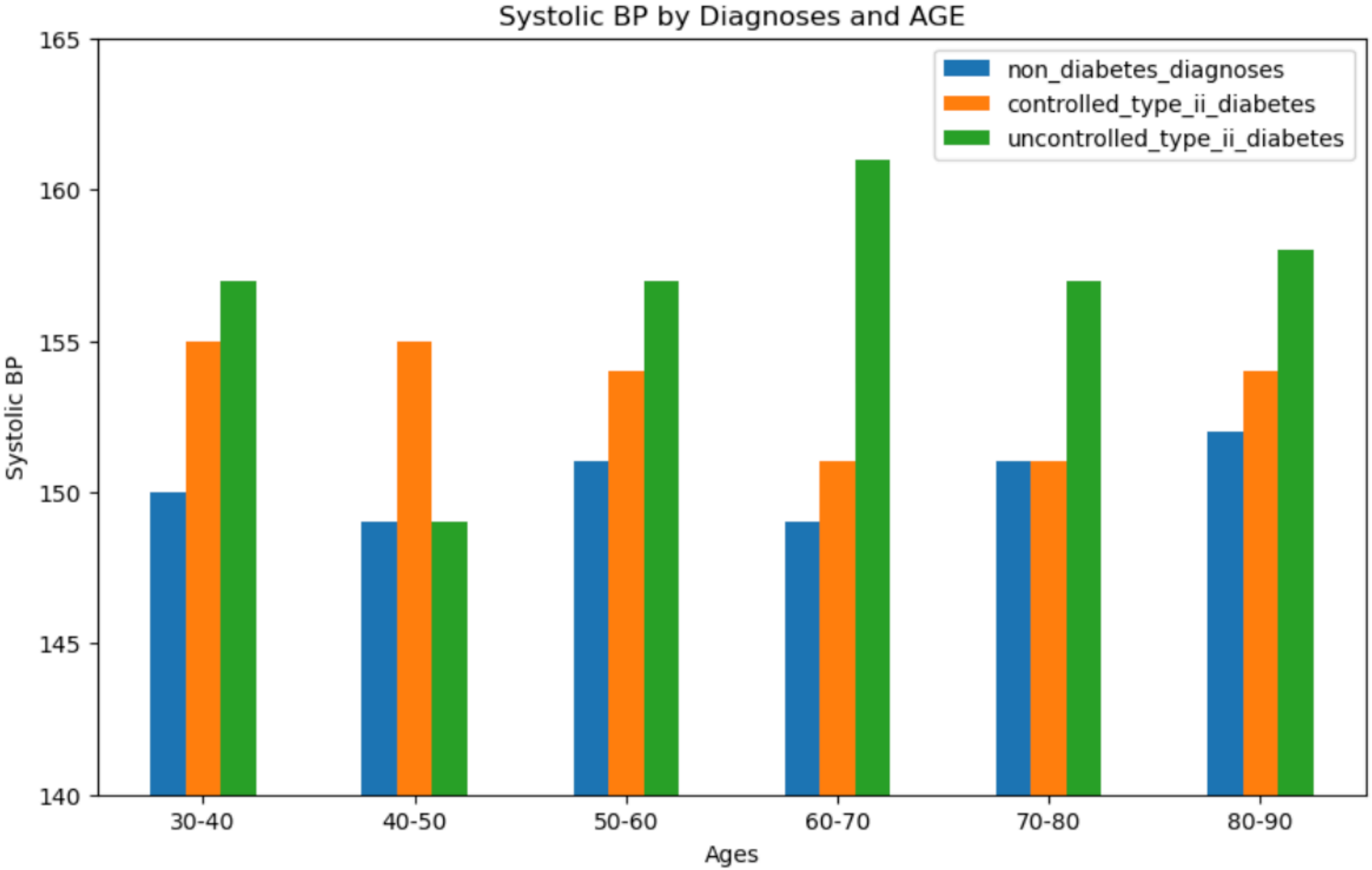
Searching for BMI and blood pressure median values for diabetic categories.

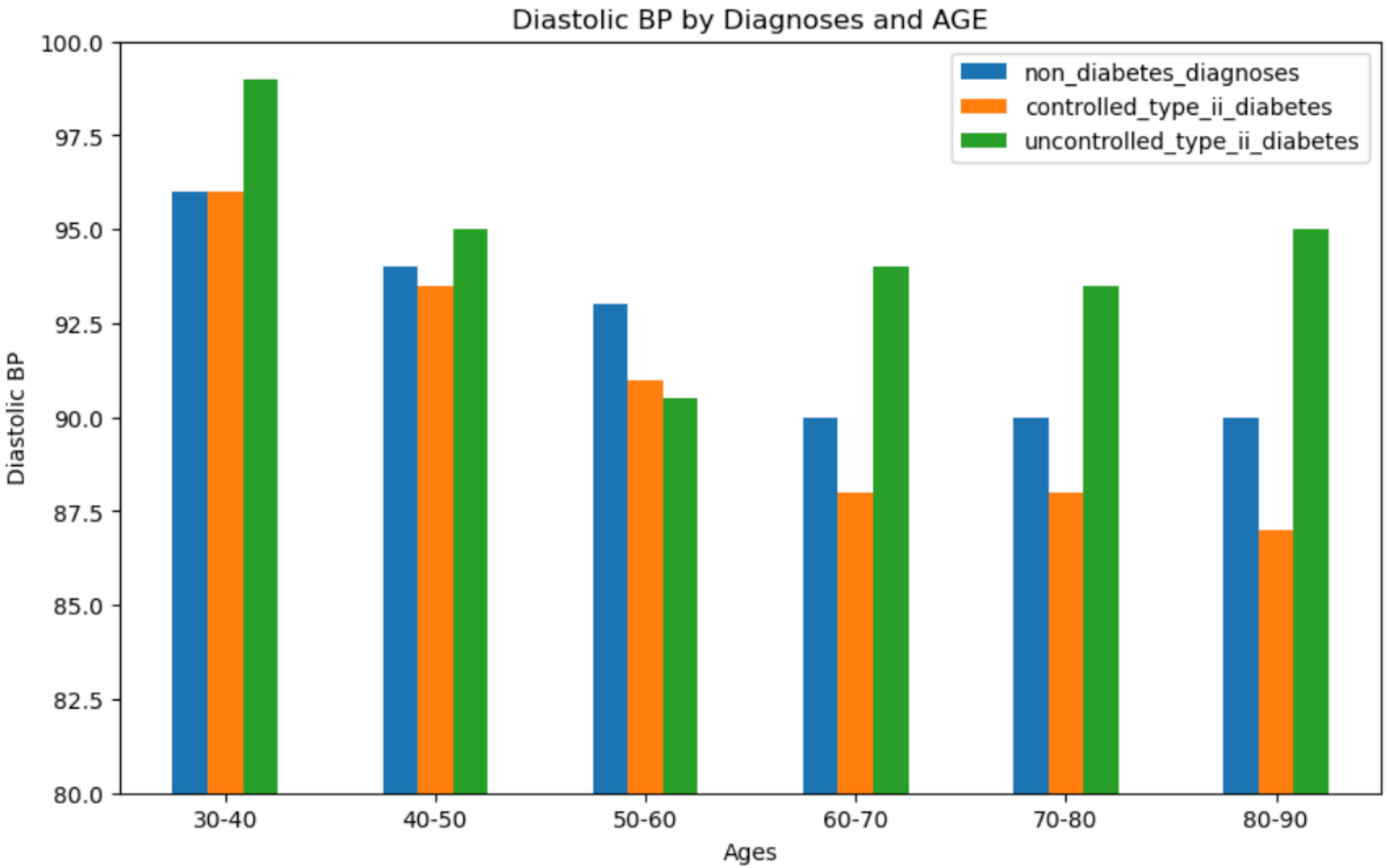
See graphs on following slides.



```
Median BMI
       non_diabetes_diagnoses  controlled_type_ii_diabetes  uncontrolled_type_ii_diabetes
30-40               26.828861                    36.641847                      28.625541
40-50               28.381291                    32.413641                      30.971050
50-60               28.714498                    31.875311                      33.576488
60-70               28.595861                    31.033262                      31.866905
70-80               27.661483                    29.177453                      28.831531
80-90               25.951305                    27.264618                      26.891070

Median Systolic BP
       non_diabetes_diagnoses  controlled_type_ii_diabetes  uncontrolled_type_ii_diabetes
30-40                   150.0                        155.0                          157.0
40-50                   149.0                        155.0                          149.0
50-60                   151.0                        154.0                          157.0
60-70                   149.0                        151.0                          161.0
70-80                   151.0                        151.0                          157.0
80-90                   152.0                        154.0                          158.0

Median Diastolic BP
       non_diabetes_diagnoses  controlled_type_ii_diabetes  uncontrolled_type_ii_diabetes
30-40                    96.0                         96.0                           99.0
40-50                    94.0                         93.5                           95.0
50-60                    93.0                         91.0                           90.5
60-70                    90.0                         88.0                           94.0
70-80                    90.0                         88.0                           93.5
80-90                    90.0                         87.0                           95.0
```


BMI by Diagnoses and AGE
38
36
34
32
30
28
26
24
BMI
30-40
40-50
50-60
60-70
70-80
80-90
Ages
non_diabetes_diagnoses
controlled_type_ii_diabetes
uncontrolled_type_ii_diabetes


Systolic BP by Diagnoses and AGE
non_diabetes_diagnoses
controlled_type_ii_diabetes
uncontrolled_type_ii_diabetes
Systolic BP
165
160
155
150
145
140
30-40
40-50
50-60
60-70
70-80
80-90
Ages


Diastolic BP by Diagnoses and AGE
non_diabetes_diagnoses
controlled_type_ii_diabetes
uncontrolled_type_ii_diabetes
Diastolic BP
100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
30-40
40-50
50-60
60-70
70-80
80-90
Ages