Big Data Applications

Contents

Introduction

Pre-requisites

- Be comfortable with basic Python. Familiarity with Numpy and Pandas is beneficial but not required.
- Have an environment ready for the several hands-on exercises (e.g., own laptop or cloud resource)

Syllabus

Something broken?

If you encounter any problem or bug in these materials, please remember to add an issue to the course repo, explaining the problem and, potentially, its solution. By doing this, you will improve the instructions for future users. 🎉

Installation Instructions

Introduction

This document contains instructions for installation of the software we'll be using during the course. If you want to follow the training on your own machines, then please complete these instructions.

If you encounter any problem during installation and you manage to solve them, please remember to add an issue, explaining the problem and solution. By doing this you will be helping to improve the instructions for future users! :tada:

What we're installing

- the `Python` programming language (version `3.7` or greater)
- `git` for version control
- your favourite text editor
- `Apache Spark`
- `Jupyter notebooks`

Please ensure that you have a computer or a cloud-based workbench with all of these installed.

Linux

Package Manager

`Linux` users should be able to use their package manager to install all of the software from this page.

However note that if you are running an older `Linux` distribution you may get older versions with different look and features. We target Ubuntu 21.10 for the sake of homogeneity.

> **Tip**
>
> We recommend to upgrade the system and get the latest security packages: `sudo apt-get upgrade`
>
> Please do not forget to restart after the upgrade finishes.

You can test the `Python` version by issuing:

```
python3 --version
```

You should get an output similar to the following:

```
Python 3.9.7
```

Python via package manager

Recents versions of `Ubuntu` come with mostly up to date versions of all needed packages.

The version of `IPython` might be slightly out of date. Thus, you may wish to upgrade this using `pip`.

You should ensure that the following packages are installed using `apt-get`:

- `python3-pip`
- `jupyter`
- `ipython3`

```
sudo apt-update
sudo apt-get install -y gcc make perl python3-pip jupyter ipython3
```

You can test the installation by issuing:

```
pip --version
```

You should get an output similar to the following:

```
pip 20.3.4 from /usr/lib/python3/dist-packages/pip (python 3.9)
```

Editor

You have many different text editors suitable for programming at your fingertips. Here is an opinionated list of editors in case you do not already have a favourite:

- Visual Studio Code
- Atom
- Neovim
- Vim

Apache Spark

TL;DR

You can skip the whole Step-by-step Setup section by issuing the following commands in a terminal.

```
cd ~
wget https://dlcdn.apache.org/spark/spark-3.2.1/spark-3.2.1-bin-hadoop3.2.tgz
sudo tar -xzvf spark-3.2.1-bin-hadoop3.2.tgz
sudo apt-get install openjdk-8-jre -y
echo 'export SPARK_HOME=~/spark-3.2.1-bin-hadoop3.2' >> ~/.bashrc
echo 'export PATH=$SPARK_HOME/bin:$PATH' >> ~/.bashrc
echo 'export PYTHONPATH=$SPARK_HOME/python:$PYTHONPATH' >> ~/.bashrc
echo 'export PYSPARK_DRIVER_PYTHON=jupyter' >> ~/.bashrc
echo 'export PYSPARK_DRIVER_PYTHON_OPTS=notebook' >> ~/.bashrc
echo 'export PYSPARK_PYTHON=python3.9' >> ~/.bashrc
echo 'export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64' >> ~/.bashrc
exec bash
jupyter notebook --generate-config
sed -i '/# c.NotebookApp.use_redirect_file/s/$*True/False/g' ~/.jupyter/jupyter_notebook_config.py
sed -i '/c.NotebookApp.use_redirect_file/s/^#*\s*//g' ~/.jupyter/jupyter_notebook_config.py
```

Step-by-step Setup

We will setup Apache Spark now. You need to open the `downloads` page, and download a spark distribution.

Download Apache Spark

We suggest to choose the same options as on the screenshot below. If you see a newer version is available, please feel free to choose it.

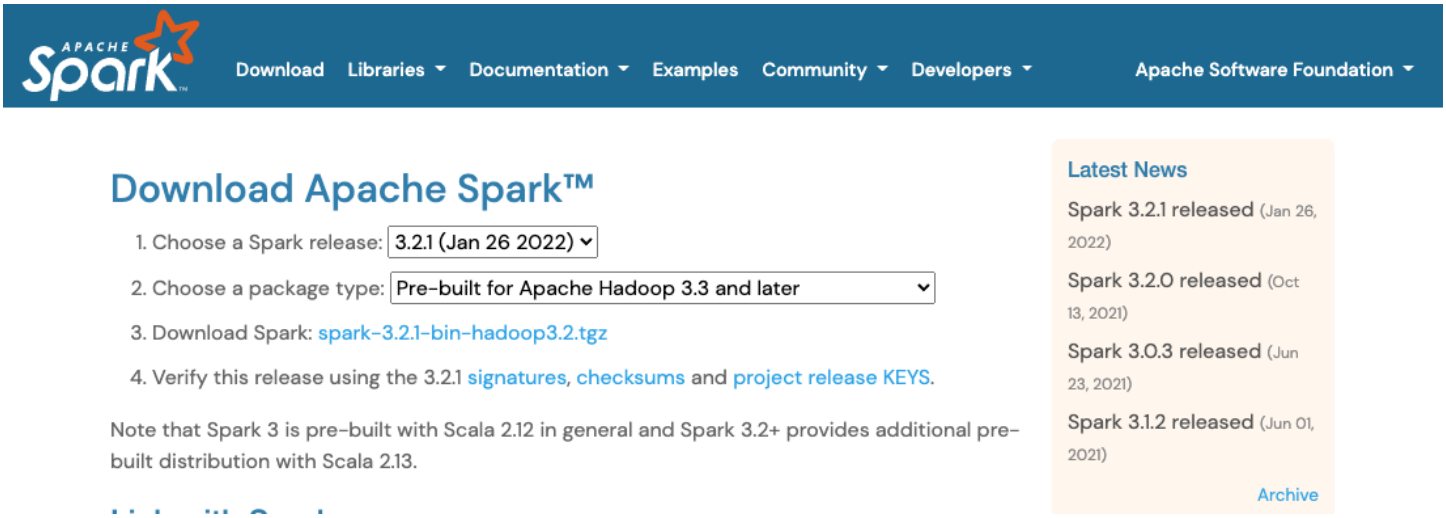
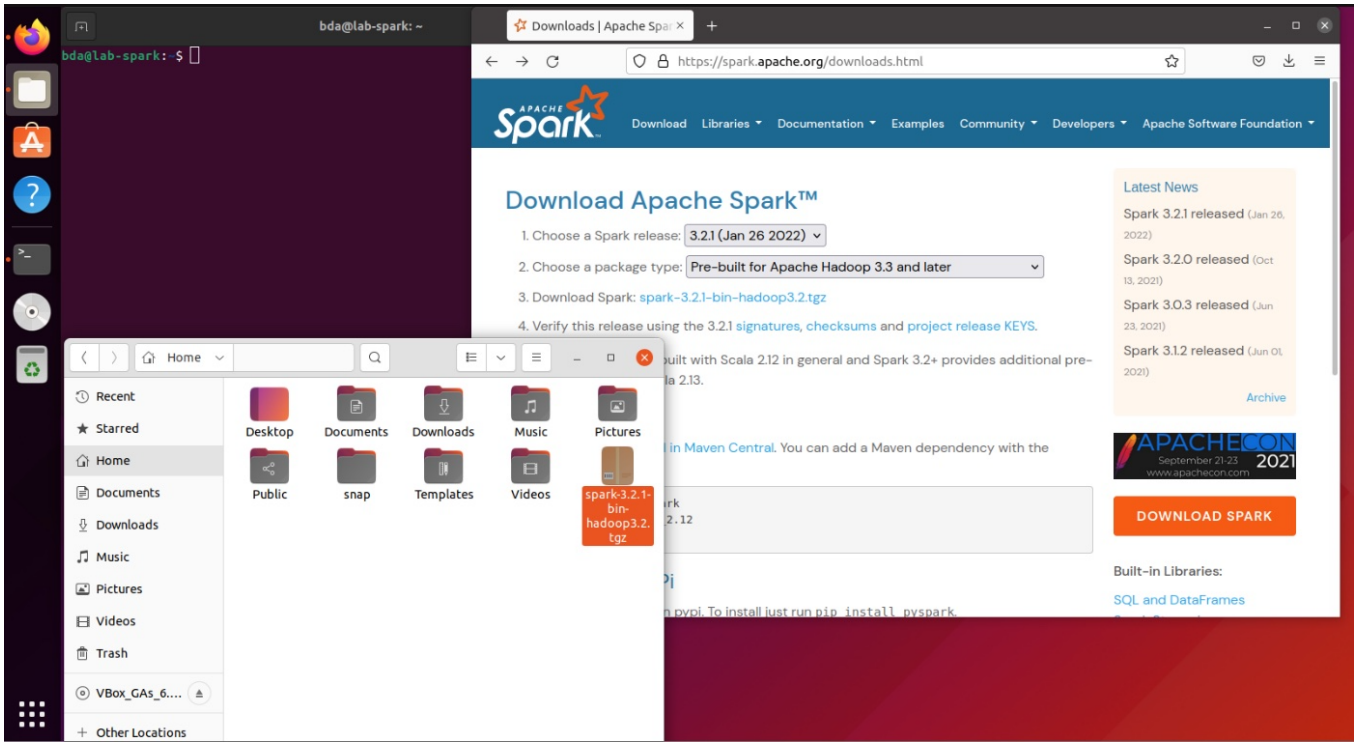


Image Link

Tip

We want the package in the right location, so open the file explorer and place it into your home folder.



Then go to your command line and issue the following to unzip the downloaded file:

```
sudo tar -xzvf spark-3.2.1-bin-hadoop3.2.tgz
```

Important

The above command assumes you downloaded version 3.2.1 with hadoop 3.2 binaries. Please ammend it as needed.

Setup JRE

Next step is installing a JRE (Java Runtime Engine), so that we can use the PySpark shell (or submit a job to a cluster):

```
sudo apt-get install openjdk-8-jre -y
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
```

Configure PySpark

We need to define several environment variables to let Spark find `Python` and use Jupyter notebooks when initiating the PySpark shell:

```
echo 'export SPARK_HOME=~/spark-3.2.1-bin-hadoop3.2' >> ~/.bashrc
echo 'export PATH=$SPARK_HOME/bin:$PATH' >> ~/.bashrc
echo 'export PYTHONPATH=$SPARK_HOME/python:$PYTHONPATH' >> ~/.bashrc
echo 'export PYSPARK_DRIVER_PYTHON=jupyter' >> ~/.bashrc
echo 'export PYSPARK_DRIVER_PYTHON_OPTS=notebook' >> ~/.bashrc
echo 'export PYSPARK_PYTHON=python3.9' >> ~/.bashrc
```

Note

PySpark requires the same minor version of Python in both driver and workers. This is why we specify what exact version to use in the `PYSPARK_PYTHON` variable.

> **Important**
>
> Please do not forget to reload your interactive shell session. [1]

Workaround for Jupyter Notebooks in Linux

If you start a PySpark shell (i.e., `pyspark`) and get a "Access to the file was denied after starting…", then you could try the following workround. [2]

```
jupyter notebook --generate-config
sed -i '/# c.NotebookApp.use_redirect_file/s/$*True/False/g' ~/.jupyter/jupyter_notebook_config.py
sed -i '/c.NotebookApp.use_redirect_file/s/^#*\s*//g' ~/.jupyter/jupyter_notebook_config.py
```

---

[1] https://www.delftstack.com/howto/linux/reload-bashrc/

[2] https://github.com/jupyter/notebook/issues/4353#issuecomment-570564277

Introduction to RDD

In this first session, we will introduce what is a Resilient Distributed Dataset (RDD). Please refer to the original paper for in-depth details [1].

To understand how Spark works, we need to understand the essence of RDD. Long story short, an RDD **represents a fault-tolerant collection of elements partitioned across the nodes of a cluster that can be operated in parallel**. We will chop the last sentence into pieces now.

Processing Logic Expressed as RDD Operations

We use RDDs to express a certain processic logic we want to apply to a dataset. For example, finding the average number of days required to recover from a certain disease.

Then, Spark makes its magic to **schedule** and **execute** our data processing logic on a distributed fashion.

If we look behind the scenes, the Spark runtime requires to determine the order of execution of RDDs, and make sure the execution is fault-tolerant. It uses the following pieces of information for the aforementioned purposes:

- Lineage
  - Dependencies on parent RDDs
- Fault-tolerance
  - The partitions that makes the whole dataset: used to execute in **parallel** to speed up the computation with **executors**.
  - The function for computing all the rows in the dataset: provided by users (i.e., "you"). Each **executor** in the cluster execute this function against each row in each partition.

With the above information, Spark can reproduce the RDD in case of failure scenarios.

Quick Demo

Prepare a Jupyter Kernel

You will need to prepare a Jupyter kernel in order to run this notebook on your local environment.

> **Note**
>
> This course relies on Poetry as package manager. Commands below can might slightly differ if you are using a different strategy (e.g., Pipenv or Conda, just to mention a few).

```
poetry shell
ipython kernel install --name "bda-labs" --user
```

▶ Details

Initialize Spark

We need to do some preparation first. In the next snippet, we will import the SparkContext and SparkConf classes.

Now we we proceed to create the **configuration** for our application and a **context** which tells Spark how to access the execution environment (local or a cluster).

```
from pyspark import SparkContext, SparkConf

# we use "local" to indicate we're running in local mode
conf = SparkConf().setAppName("intro-to-rdd").setMaster("local")
sc = SparkContext(conf=conf)
```

```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform
(file:/opt/hostedtoolcache/Python/3.9.12/x64/lib/python3.9/site-
packages/pyspark/jars/spark-unsafe_2.12-3.2.1.jar) to constructor
java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of
org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective
access operations
WARNING: All illegal access operations will be denied in a future release
```

```
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
```

```
22/05/19 05:06:41 WARN NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
```

SparkConf

We triggered a couple of actions with the above code. To start with, we created a `SparkConfig` object and chained two important configurations: - The Spark application name. - The URL where the `master node` can be located. **Setting it to `"local"` is a special configuration to indicate we are running on the same host where this code is being executed.**

Note: it is important to notice that Spark runs on a JVM. When a `SparkConf` is created, it will load values from `spark.*` Java system properties. If we make such properties available to the process running Spark, then they would be picked up.

Let's inspect a couple of common Spark configuration properties:

```
print(f"spark.app.name: {conf.get('spark.app.name')}")
print(f"spark.master: {conf.get('spark.master')}")
print(f"spark.home: {conf.get('spark.home')}")
```

```
spark.app.name: intro-to-rdd
spark.master: local
spark.home: None
```

So far so good. We get the same value we chained to the `SparkConf` object before.

Homework/self-research:

- Why don't we have a value for `spark.home`?
- Should we care?

Should we want to know all the properties available, then the following snippet might help.

```
for p in sorted(conf.getAll(), key=lambda p: p[0]):
    print(p)
```

```
('spark.app.name', 'intro-to-rdd')
('spark.master', 'local')
```

SparkContext

`SparkContext` is at the hearth of Spark. It is the main entry point for Spark that representes the connection of the "driver program" to a Spark cluster. **You must get familiarized with it**.

We can pass several parameters to it, from which two of them are mandatory: `master` and `appName`. In our case, we passed such information wrapped into a `SparkConf` object.

> **Attention**
>
> - `SparkContext` is **always** created on the driver and **it is not serialized**. Thus, it cannot be shipped to workers.
> - We can have one `SparkContext` per application at most. Try to run the cell where we created the `SparkContext` and see what happens!

> **Note**
>
> Actually, only one `SparkContext` can be active per JVM. If we want to create a new one, we must call the `stop()` method to our existing `SparkContext` object first.

Another consequence of passing a `SparkConf` object to the `SparkContext` constructor is **immutable configuration**. The `SparkConf` object is cloned and can no longer be modified. Let's see it in action:

```
# Is our SparkConf object the same we get from SparkContext?
print(f"conf == sc.getConf() --> {conf == sc.getConf()}")
```

```
conf == sc.getConf() --> False
```

Define an RDD

Let's start by defining a very simple RDD. We can think of it as data points indicating the recovery days for a certain disease.

```
recovery_days_per_disease = [("disease_1", [3, 6, 9, 10]), ("disease_2", [11, 11,
10, 9])]
rdd = sc.parallelize(recovery_days_per_disease)
```

Now we can operate on it. For example, we can compute the average.

```
rdd.mapValues(lambda x: sum(x) / len(x)).collect()
```

```
[Stage 0:>                                                          (0 + 1) / 1]


```

```
[('disease_1', 7.0), ('disease_2', 10.25)]
```

References

[1] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauly, Michael J Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, 15–28. 2012.

Introduction to the DataFrame API

In this section, we will introduce the DataFrame and Dataset APIs.

We will use a small subset from the Record Linkage Comparison Data Set, borrowed from UC Irvine Machine Learning Repository. It consists of several CSV files with match scores for patients in a Germany hospital, but we will use only one of them for the sake of simplicity. Please consult [1] and [2] for more details regarding the data sets and research.

Setup

- Setup a `SparkSession` to work with the Dataset and DataFrame API
- Unzip the `scores.zip` file located under `data` folder.

```
from pyspark import SparkContext, SparkConf

conf = SparkConf().setAppName("intro-to-df").setMaster("local")
sc = SparkContext(conf=conf)
# Avoid polluting the console with warning messages
sc.setLogLevel("ERROR")
```

```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform
(file:/opt/hostedtoolcache/Python/3.9.12/x64/lib/python3.9/site-
packages/pyspark/jars/spark-unsafe_2.12-3.2.1.jar) to constructor
java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of
org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective
access operations
WARNING: All illegal access operations will be denied in a future release
```

```
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
```

```
22/05/19 05:06:46 WARN NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
```

Create a SparkSession to work with the DataFrame API

```
from pyspark.sql import SparkSession

spark = SparkSession(sc)
```

```
help(SparkSession)
```

```
Help on class SparkSession in module pyspark.sql.session:

class SparkSession(pyspark.sql.pandas.conversion.SparkConversionMixin)
 |  SparkSession(sparkContext, jsparkSession=None)
 |
 |  The entry point to programming Spark with the Dataset and DataFrame API.
 |
 |  A SparkSession can be used create :class:`DataFrame`, register
:class:`DataFrame` as
 |  tables, execute SQL over tables, cache tables, and read parquet files.
 |  To create a :class:`SparkSession`, use the following builder pattern:
 |
 |  .. autoattribute:: builder
 |     :annotation:
 |
 |  Examples
 |  --------
 |  >>> spark = SparkSession.builder \
 |  ...     .master("local") \
 |  ...     .appName("Word Count") \
 |  ...     .config("spark.some.config.option", "some-value") \
 |  ...     .getOrCreate()
 |
 |  >>> from datetime import datetime
 |  >>> from pyspark.sql import Row
 |  >>> spark = SparkSession(sc)
 |  >>> allTypes = sc.parallelize([Row(i=1, s="string", d=1.0, l=1,
 |  ...     b=True, list=[1, 2, 3], dict={"s": 0}, row=Row(a=1),
 |  ...     time=datetime(2014, 8, 1, 14, 1, 5))])
```

```
 |  >>> df = allTypes.toDF()
 |  >>> df.createOrReplaceTempView("allTypes")
 |  >>> spark.sql('select i+1, d+1, not b, list[1], dict["s"], time, row.a '
 |  ...           'from allTypes where b and i > 0').collect()
 |  [Row((i + 1)=2, (d + 1)=2.0, (NOT b)=False, list[1]=2,         dict[s]=0,
time=datetime.datetime(2014, 8, 1, 14, 1, 5), a=1)]
 |  >>> df.rdd.map(lambda x: (x.i, x.s, x.d, x.l, x.b, x.time, x.row.a,
x.list)).collect()
 |  [(1, 'string', 1.0, 1, True, datetime.datetime(2014, 8, 1, 14, 1, 5), 1, [1, 2,
3])]
 |
 |  Method resolution order:
 |      SparkSession
 |      pyspark.sql.pandas.conversion.SparkConversionMixin
 |      builtins.object
 |
 |  Methods defined here:
 |
 |  __enter__(self)
 |      Enable 'with SparkSession.builder.(...).getOrCreate() as session: app'
syntax.
 |
 |      .. versionadded:: 2.0
 |
 |  __exit__(self, exc_type, exc_val, exc_tb)
 |      Enable 'with SparkSession.builder.(...).getOrCreate() as session: app'
syntax.
 |
 |      Specifically stop the SparkSession on exit of the with block.
 |
 |      .. versionadded:: 2.0
 |
 |  __init__(self, sparkContext, jsparkSession=None)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |
 |  createDataFrame(self, data, schema=None, samplingRatio=None, verifySchema=True)
 |      Creates a :class:`DataFrame` from an :class:`RDD`, a list or a
:class:`pandas.DataFrame`.
 |
 |      When ``schema`` is a list of column names, the type of each column
 |      will be inferred from ``data``.
 |
 |      When ``schema`` is ``None``, it will try to infer the schema (column names
and types)
 |      from ``data``, which should be an RDD of either :class:`Row`,
 |      :class:`namedtuple`, or :class:`dict`.
 |
 |      When ``schema`` is :class:`pyspark.sql.types.DataType` or a datatype
string, it must match
 |      the real data, or an exception will be thrown at runtime. If the given
schema is not
 |      :class:`pyspark.sql.types.StructType`, it will be wrapped into a
 |      :class:`pyspark.sql.types.StructType` as its only field, and the field name
will be "value".
 |      Each record will also be wrapped into a tuple, which can be converted to
row later.
 |
 |      If schema inference is needed, ``samplingRatio`` is used to determined the
ratio of
 |      rows used for schema inference. The first row will be used if
``samplingRatio`` is ``None``.
 |
 |      .. versionadded:: 2.0.0
 |
 |      .. versionchanged:: 2.1.0
 |         Added verifySchema.
 |
 |      Parameters
 |      ----------
 |      data : :class:`RDD` or iterable
 |          an RDD of any kind of SQL data representation (:class:`Row`,
 |          :class:`tuple`, ``int``, ``boolean``, etc.), or :class:`list`, or
 |          :class:`pandas.DataFrame`.
 |      schema : :class:`pyspark.sql.types.DataType`, str or list, optional
 |          a :class:`pyspark.sql.types.DataType` or a datatype string or a list of
 |          column names, default is None.  The data type string format equals to
 |          :class:`pyspark.sql.types.DataType.simpleString`, except that top level
struct type can
 |          omit the ``struct<>`` and atomic types use ``typeName()`` as their
format, e.g. use
 |          ``byte`` instead of ``tinyint`` for
:class:`pyspark.sql.types.ByteType`.
 |          We can also use ``int`` as a short name for
:class:`pyspark.sql.types.IntegerType`.
 |      samplingRatio : float, optional
 |          the sample ratio of rows used for inferring
 |      verifySchema : bool, optional
```

```
 |          verify data types of every row against schema. Enabled by default.
 |
 |      Returns
 |      -------
 |      :class:`DataFrame`
 |
 |      Notes
 |      -----
 |      Usage with spark.sql.execution.arrow.pyspark.enabled=True is experimental.
 |
 |      Examples
 |      --------
 |      >>> l = [('Alice', 1)]
 |      >>> spark.createDataFrame(l).collect()
 |      [Row(_1='Alice', _2=1)]
 |      >>> spark.createDataFrame(l, ['name', 'age']).collect()
 |      [Row(name='Alice', age=1)]
 |
 |      >>> d = [{'name': 'Alice', 'age': 1}]
 |      >>> spark.createDataFrame(d).collect()
 |      [Row(age=1, name='Alice')]
 |
 |      >>> rdd = sc.parallelize(l)
 |      >>> spark.createDataFrame(rdd).collect()
 |      [Row(_1='Alice', _2=1)]
 |      >>> df = spark.createDataFrame(rdd, ['name', 'age'])
 |      >>> df.collect()
 |      [Row(name='Alice', age=1)]
 |
 |      >>> from pyspark.sql import Row
 |      >>> Person = Row('name', 'age')
 |      >>> person = rdd.map(lambda r: Person(*r))
 |      >>> df2 = spark.createDataFrame(person)
 |      >>> df2.collect()
 |      [Row(name='Alice', age=1)]
 |
 |      >>> from pyspark.sql.types import *
 |      >>> schema = StructType([
 |      ...    StructField("name", StringType(), True),
 |      ...    StructField("age", IntegerType(), True)])
 |      >>> df3 = spark.createDataFrame(rdd, schema)
 |      >>> df3.collect()
 |      [Row(name='Alice', age=1)]
 |
 |      >>> spark.createDataFrame(df.toPandas()).collect()  # doctest: +SKIP
 |      [Row(name='Alice', age=1)]
 |      >>> spark.createDataFrame(pandas.DataFrame([[1, 2]])).collect()  # doctest:
+SKIP
 |      [Row(0=1, 1=2)]
 |
 |      >>> spark.createDataFrame(rdd, "a: string, b: int").collect()
 |      [Row(a='Alice', b=1)]
 |      >>> rdd = rdd.map(lambda row: row[1])
 |      >>> spark.createDataFrame(rdd, "int").collect()
 |      [Row(value=1)]
 |      >>> spark.createDataFrame(rdd, "boolean").collect() # doctest:
+IGNORE_EXCEPTION_DETAIL
 |      Traceback (most recent call last):
 |          ...
 |      Py4JJavaError: ...
 |
 |  newSession(self)
 |      Returns a new :class:`SparkSession` as new session, that has separate
SQLConf,
 |      registered temporary views and UDFs, but shared :class:`SparkContext` and
 |      table cache.
 |
 |      .. versionadded:: 2.0
 |
 |  range(self, start, end=None, step=1, numPartitions=None)
 |      Create a :class:`DataFrame` with single :class:`pyspark.sql.types.LongType`
column named
 |      ``id``, containing elements in a range from ``start`` to ``end``
(exclusive) with
 |      step value ``step``.
 |
 |      .. versionadded:: 2.0.0
 |
 |      Parameters
 |      ----------
 |      start : int
 |          the start value
 |      end : int, optional
 |          the end value (exclusive)
 |      step : int, optional
 |          the incremental step (default: 1)
 |      numPartitions : int, optional
```

```
 |          the number of partitions of the DataFrame
 |
 |      Returns
 |      -------
 |      :class:`DataFrame`
 |
 |      Examples
 |      --------
 |      >>> spark.range(1, 7, 2).collect()
 |      [Row(id=1), Row(id=3), Row(id=5)]
 |
 |      If only one argument is specified, it will be used as the end value.
 |
 |      >>> spark.range(3).collect()
 |      [Row(id=0), Row(id=1), Row(id=2)]
 |
 |  sql(self, sqlQuery)
 |      Returns a :class:`DataFrame` representing the result of the given query.
 |
 |      .. versionadded:: 2.0.0
 |
 |      Returns
 |      -------
 |      :class:`DataFrame`
 |
 |      Examples
 |      --------
 |      >>> df.createOrReplaceTempView("table1")
 |      >>> df2 = spark.sql("SELECT field1 AS f1, field2 as f2 from table1")
 |      >>> df2.collect()
 |      [Row(f1=1, f2='row1'), Row(f1=2, f2='row2'), Row(f1=3, f2='row3')]
 |
 |  stop(self)
 |      Stop the underlying :class:`SparkContext`.
 |
 |      .. versionadded:: 2.0
 |
 |  table(self, tableName)
 |      Returns the specified table as a :class:`DataFrame`.
 |
 |      .. versionadded:: 2.0.0
 |
 |      Returns
 |      -------
 |      :class:`DataFrame`
 |
 |      Examples
 |      --------
 |      >>> df.createOrReplaceTempView("table1")
 |      >>> df2 = spark.table("table1")
 |      >>> sorted(df.collect()) == sorted(df2.collect())
 |      True
 |
 |  ----------------------------------------------------------------------
 |  Class methods defined here:
 |
 |  getActiveSession() from builtins.type
 |      Returns the active :class:`SparkSession` for the current thread, returned
by the builder
 |
 |      .. versionadded:: 3.0.0
 |
 |      Returns
 |      -------
 |      :class:`SparkSession`
 |          Spark session if an active session exists for the current thread
 |
 |      Examples
 |      --------
 |      >>> s = SparkSession.getActiveSession()
 |      >>> l = [('Alice', 1)]
 |      >>> rdd = s.sparkContext.parallelize(l)
 |      >>> df = s.createDataFrame(rdd, ['name', 'age'])
 |      >>> df.select("age").collect()
 |      [Row(age=1)]
 |
 |  ----------------------------------------------------------------------
 |  Readonly properties defined here:
 |
 |  catalog
 |      Interface through which the user may create, drop, alter or query
underlying
 |      databases, tables, functions, etc.
 |
 |      .. versionadded:: 2.0.0
 |
 |      Returns
```

```
 |      -------
 |      :class:`Catalog`
 |
 |  conf
 |      Runtime configuration interface for Spark.
 |
 |      This is the interface through which the user can get and set all Spark and
Hadoop
 |      configurations that are relevant to Spark SQL. When getting the value of a
config,
 |      this defaults to the value set in the underlying :class:`SparkContext`, if
any.
 |
 |      Returns
 |      -------
 |      :class:`pyspark.sql.conf.RuntimeConfig`
 |
 |      .. versionadded:: 2.0
 |
 |  read
 |      Returns a :class:`DataFrameReader` that can be used to read data
 |      in as a :class:`DataFrame`.
 |
 |      .. versionadded:: 2.0.0
 |
 |      Returns
 |      -------
 |      :class:`DataFrameReader`
 |
 |  readStream
 |      Returns a :class:`DataStreamReader` that can be used to read data streams
 |      as a streaming :class:`DataFrame`.
 |
 |      .. versionadded:: 2.0.0
 |
 |      Notes
 |      -----
 |      This API is evolving.
 |
 |      Returns
 |      -------
 |      :class:`DataStreamReader`
 |
 |  sparkContext
 |      Returns the underlying :class:`SparkContext`.
 |
 |      .. versionadded:: 2.0
 |
 |  streams
 |      Returns a :class:`StreamingQueryManager` that allows managing all the
 |      :class:`StreamingQuery` instances active on `this` context.
 |
 |      .. versionadded:: 2.0.0
 |
 |      Notes
 |      -----
 |      This API is evolving.
 |
 |      Returns
 |      -------
 |      :class:`StreamingQueryManager`
 |
 |  udf
 |      Returns a :class:`UDFRegistration` for UDF registration.
 |
 |      .. versionadded:: 2.0.0
 |
 |      Returns
 |      -------
 |      :class:`UDFRegistration`
 |
 |  version
 |      The version of Spark on which this application is running.
 |
 |      .. versionadded:: 2.0
 |
 |  ----------------------------------------------------------------------
 |  Data and other attributes defined here:
 |
 |  Builder = <class 'pyspark.sql.session.SparkSession.Builder'>
 |      Builder for :class:`SparkSession`.
 |
 |
 |  builder = <pyspark.sql.session.SparkSession.Builder object>
 |
 |  ----------------------------------------------------------------------
 |  Data descriptors inherited from
```

```
pyspark.sql.pandas.conversion.SparkConversionMixin:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)
```

Unzip the scores file, if it was not done already

```
from os import path
scores_zip = path.join("data", "scores.zip")
scores_csv = path.join("data", "scores.csv")

%set_env SCORES_ZIP=$scores_zip
%set_env SCORES_CSV=$scores_csv
```

```
env: SCORES_ZIP=data/scores.zip
env: SCORES_CSV=data/scores.csv
```

```
%%bash
command -v unzip >/dev/null 2>&1 || { echo >&2 "unzip command is not installed.
Aborting."; exit 1; }
[[ -f "$SCORES_CSV" ]] && { echo "file data/$SCORES_CSV already exist. Skipping.";
exit 0; }

[[ -f "$SCORES_ZIP" ]] || { echo "file data/$SCORES_ZIP does not exist.
Aborting."; exit 1; }

echo "Unzip file $SCORES_ZIP"
unzip "$SCORES_ZIP" -d data
```

```
Unzip file data/scores.zip
```

```
Archive:  data/scores.zip
```

```
  inflating: data/scores.csv
```

```
  inflating: data/__MACOSX/._scores.csv
```

```
! head "$SCORES_CSV"
```

```
"id_1","id_2","cmp_fname_c1","cmp_fname_c2","cmp_lname_c1","cmp_lname_c2","cmp_sex"
,"cmp_bd","cmp_bm","cmp_by","cmp_plz","is_match"
37291,53113,0.833333333333333,?,1,?,1,1,1,1,0,TRUE
39086,47614,1,?,1,?,1,1,1,1,1,TRUE
70031,70237,1,?,1,?,1,1,1,1,1,TRUE
84795,97439,1,?,1,?,1,1,1,1,1,TRUE
36950,42116,1,?,1,1,1,1,1,1,1,TRUE
42413,48491,1,?,1,?,1,1,1,1,1,TRUE
25965,64753,1,?,1,?,1,1,1,1,1,TRUE
49451,90407,1,?,1,?,1,1,1,1,0,TRUE
39932,40902,1,?,1,?,1,1,1,1,1,TRUE
```

Loading the Scores CSV file into a DataFrame

We are going to use the Reader API

```
help(spark.read)
```

```
Help on DataFrameReader in module pyspark.sql.readwriter object:

class DataFrameReader(OptionUtils)
 |  DataFrameReader(spark)
 |
 |  Interface used to load a :class:`DataFrame` from external storage systems
 |  (e.g. file systems, key-value stores, etc). Use :attr:`SparkSession.read`
 |  to access this.
 |
```

```
 |  .. versionadded:: 1.4
 |
 |  Method resolution order:
 |      DataFrameReader
 |      OptionUtils
 |      builtins.object
 |
 |  Methods defined here:
 |
 |  __init__(self, spark)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |
 |  csv(self, path, schema=None, sep=None, encoding=None, quote=None, escape=None,
comment=None, header=None, inferSchema=None, ignoreLeadingWhiteSpace=None,
ignoreTrailingWhiteSpace=None, nullValue=None, nanValue=None, positiveInf=None,
negativeInf=None, dateFormat=None, timestampFormat=None, maxColumns=None,
maxCharsPerColumn=None, maxMalformedLogPerPartition=None, mode=None,
columnNameOfCorruptRecord=None, multiLine=None, charToEscapeQuoteEscaping=None,
samplingRatio=None, enforceSchema=None, emptyValue=None, locale=None, lineSep=None,
pathGlobFilter=None, recursiveFileLookup=None, modifiedBefore=None,
modifiedAfter=None, unescapedQuoteHandling=None)
 |      Loads a CSV file and returns the result as a  :class:`DataFrame`.
 |
 |      This function will go through the input once to determine the input schema
if
 |      ``inferSchema`` is enabled. To avoid going through the entire data once,
disable
 |      ``inferSchema`` option or specify the schema explicitly using ``schema``.
 |
 |      .. versionadded:: 2.0.0
 |
 |      Parameters
 |      ----------
 |      path : str or list
 |          string, or list of strings, for input path(s),
 |          or RDD of Strings storing CSV rows.
 |      schema : :class:`pyspark.sql.types.StructType` or str, optional
 |          an optional :class:`pyspark.sql.types.StructType` for the input schema
 |          or a DDL-formatted string (For example ``col0 INT, col1 DOUBLE``).
 |
 |      Other Parameters
 |      ----------------
 |      Extra options
 |          For the extra options, refer to
 |          `Data Source Option <https://spark.apache.org/docs/latest/sql-data-
sources-csv.html#data-source-option>`_
 |          in the version you use.
 |
 |          .. # noqa
 |
 |      Examples
 |      --------
 |      >>> df = spark.read.csv('python/test_support/sql/ages.csv')
 |      >>> df.dtypes
 |      [('_c0', 'string'), ('_c1', 'string')]
 |      >>> rdd = sc.textFile('python/test_support/sql/ages.csv')
 |      >>> df2 = spark.read.csv(rdd)
 |      >>> df2.dtypes
 |      [('_c0', 'string'), ('_c1', 'string')]
 |
 |  format(self, source)
 |      Specifies the input data source format.
 |
 |      .. versionadded:: 1.4.0
 |
 |      Parameters
 |      ----------
 |      source : str
 |          string, name of the data source, e.g. 'json', 'parquet'.
 |
 |      Examples
 |      --------
 |      >>> df =
spark.read.format('json').load('python/test_support/sql/people.json')
 |      >>> df.dtypes
 |      [('age', 'bigint'), ('name', 'string')]
 |
 |  jdbc(self, url, table, column=None, lowerBound=None, upperBound=None,
numPartitions=None, predicates=None, properties=None)
 |      Construct a :class:`DataFrame` representing the database table named
``table``
 |      accessible via JDBC URL ``url`` and connection ``properties``.
 |
 |      Partitions of the table will be retrieved in parallel if either ``column``
or
 |      ``predicates`` is specified. ``lowerBound``, ``upperBound`` and
``numPartitions``
```

```
 |      is needed when ``column`` is specified.
 |
 |      If both ``column`` and ``predicates`` are specified, ``column`` will be
used.
 |
 |      .. versionadded:: 1.4.0
 |
 |      Parameters
 |      ----------
 |      table : str
 |          the name of the table
 |      column : str, optional
 |          alias of ``partitionColumn`` option. Refer to ``partitionColumn`` in
 |          `Data Source Option <https://spark.apache.org/docs/latest/sql-data-
sources-jdbc.html#data-source-option>`_
 |          in the version you use.
 |      predicates : list, optional
 |          a list of expressions suitable for inclusion in WHERE clauses;
 |          each one defines one partition of the :class:`DataFrame`
 |      properties : dict, optional
 |          a dictionary of JDBC database connection arguments. Normally at
 |          least properties "user" and "password" with their corresponding values.
 |          For example { 'user' : 'SYSTEM', 'password' : 'mypassword' }
 |
 |      Other Parameters
 |      ----------------
 |      Extra options
 |          For the extra options, refer to
 |          `Data Source Option <https://spark.apache.org/docs/latest/sql-data-
sources-jdbc.html#data-source-option>`_
 |          in the version you use.
 |
 |          .. # noqa
 |
 |      Notes
 |      -----
 |      Don't create too many partitions in parallel on a large cluster;
 |      otherwise Spark might crash your external database systems.
 |
 |      Returns
 |      -------
 |      :class:`DataFrame`
 |
 |  json(self, path, schema=None, primitivesAsString=None, prefersDecimal=None,
allowComments=None, allowUnquotedFieldNames=None, allowSingleQuotes=None,
allowNumericLeadingZero=None, allowBackslashEscapingAnyCharacter=None, mode=None,
columnNameOfCorruptRecord=None, dateFormat=None, timestampFormat=None,
multiLine=None, allowUnquotedControlChars=None, lineSep=None, samplingRatio=None,
dropFieldIfAllNull=None, encoding=None, locale=None, pathGlobFilter=None,
recursiveFileLookup=None, allowNonNumericNumbers=None, modifiedBefore=None,
modifiedAfter=None)
 |      Loads JSON files and returns the results as a :class:`DataFrame`.
 |
 |      `JSON Lines <http://jsonlines.org/>`_ (newline-delimited JSON) is supported
by default.
 |      For JSON (one record per file), set the ``multiLine`` parameter to
``true``.
 |
 |      If the ``schema`` parameter is not specified, this function goes
 |      through the input once to determine the input schema.
 |
 |      .. versionadded:: 1.4.0
 |
 |      Parameters
 |      ----------
 |      path : str, list or :class:`RDD`
 |          string represents path to the JSON dataset, or a list of paths,
 |          or RDD of Strings storing JSON objects.
 |      schema : :class:`pyspark.sql.types.StructType` or str, optional
 |          an optional :class:`pyspark.sql.types.StructType` for the input schema
or
 |          a DDL-formatted string (For example ``col0 INT, col1 DOUBLE``).
 |
 |      Other Parameters
 |      ----------------
 |      Extra options
 |          For the extra options, refer to
 |          `Data Source Option <https://spark.apache.org/docs/latest/sql-data-
sources-json.html#data-source-option>`_
 |          in the version you use.
 |
 |          .. # noqa
 |
 |      Examples
 |      --------
 |      >>> df1 = spark.read.json('python/test_support/sql/people.json')
 |      >>> df1.dtypes
```

```
 |      [('age', 'bigint'), ('name', 'string')]
 |      >>> rdd = sc.textFile('python/test_support/sql/people.json')
 |      >>> df2 = spark.read.json(rdd)
 |      >>> df2.dtypes
 |      [('age', 'bigint'), ('name', 'string')]
 |
 |  load(self, path=None, format=None, schema=None, **options)
 |      Loads data from a data source and returns it as a :class:`DataFrame`.
 |
 |      .. versionadded:: 1.4.0
 |
 |      Parameters
 |      ----------
 |      path : str or list, optional
 |          optional string or a list of string for file-system backed data
sources.
 |      format : str, optional
 |          optional string for format of the data source. Default to 'parquet'.
 |      schema : :class:`pyspark.sql.types.StructType` or str, optional
 |          optional :class:`pyspark.sql.types.StructType` for the input schema
 |          or a DDL-formatted string (For example ``col0 INT, col1 DOUBLE``).
 |      **options : dict
 |          all other string options
 |
 |      Examples
 |      --------
 |      >>> df =
spark.read.format("parquet").load('python/test_support/sql/parquet_partitioned',
 |      ...     opt1=True, opt2=1, opt3='str')
 |      >>> df.dtypes
 |      [('name', 'string'), ('year', 'int'), ('month', 'int'), ('day', 'int')]
 |
 |      >>> df =
spark.read.format('json').load(['python/test_support/sql/people.json',
 |      ...     'python/test_support/sql/people1.json'])
 |      >>> df.dtypes
 |      [('age', 'bigint'), ('aka', 'string'), ('name', 'string')]
 |
 |  option(self, key, value)
 |      Adds an input option for the underlying data source.
 |
 |      .. versionadded:: 1.5
 |
 |  options(self, **options)
 |      Adds input options for the underlying data source.
 |
 |      .. versionadded:: 1.4
 |
 |  orc(self, path, mergeSchema=None, pathGlobFilter=None,
recursiveFileLookup=None, modifiedBefore=None, modifiedAfter=None)
 |      Loads ORC files, returning the result as a :class:`DataFrame`.
 |
 |      .. versionadded:: 1.5.0
 |
 |      Parameters
 |      ----------
 |      path : str or list
 |
 |      Other Parameters
 |      ----------------
 |      Extra options
 |          For the extra options, refer to
 |          `Data Source Option <https://spark.apache.org/docs/latest/sql-data-
sources-orc.html#data-source-option>`_
 |          in the version you use.
 |
 |          .. # noqa
 |
 |      Examples
 |      --------
 |      >>> df = spark.read.orc('python/test_support/sql/orc_partitioned')
 |      >>> df.dtypes
 |      [('a', 'bigint'), ('b', 'int'), ('c', 'int')]
 |
 |  parquet(self, *paths, **options)
 |      Loads Parquet files, returning the result as a :class:`DataFrame`.
 |
 |      .. versionadded:: 1.4.0
 |
 |      Parameters
 |      ----------
 |      paths : str
 |
 |      Other Parameters
 |      ----------------
 |      **options
 |          For the extra options, refer to
```

```
 |          `Data Source Option <https://spark.apache.org/docs/latest/sql-data-
sources-parquet.html#data-source-option>`_
 |          in the version you use.
 |
 |          .. # noqa
 |
 |      Examples
 |      --------
 |      >>> df = spark.read.parquet('python/test_support/sql/parquet_partitioned')
 |      >>> df.dtypes
 |      [('name', 'string'), ('year', 'int'), ('month', 'int'), ('day', 'int')]
 |
 |  schema(self, schema)
 |      Specifies the input schema.
 |
 |      Some data sources (e.g. JSON) can infer the input schema automatically from
data.
 |      By specifying the schema here, the underlying data source can skip the
schema
 |      inference step, and thus speed up data loading.
 |
 |      .. versionadded:: 1.4.0
 |
 |      Parameters
 |      ----------
 |      schema : :class:`pyspark.sql.types.StructType` or str
 |          a :class:`pyspark.sql.types.StructType` object or a DDL-formatted
string
 |          (For example ``col0 INT, col1 DOUBLE``).
 |
 |      >>> s = spark.read.schema("col0 INT, col1 DOUBLE")
 |
 |  table(self, tableName)
 |      Returns the specified table as a :class:`DataFrame`.
 |
 |      .. versionadded:: 1.4.0
 |
 |      Parameters
 |      ----------
 |      tableName : str
 |          string, name of the table.
 |
 |      Examples
 |      --------
 |      >>> df = spark.read.parquet('python/test_support/sql/parquet_partitioned')
 |      >>> df.createOrReplaceTempView('tmpTable')
 |      >>> spark.read.table('tmpTable').dtypes
 |      [('name', 'string'), ('year', 'int'), ('month', 'int'), ('day', 'int')]
 |
 |  text(self, paths, wholetext=False, lineSep=None, pathGlobFilter=None,
recursiveFileLookup=None, modifiedBefore=None, modifiedAfter=None)
 |      Loads text files and returns a :class:`DataFrame` whose schema starts with
a
 |      string column named "value", and followed by partitioned columns if there
 |      are any.
 |      The text files must be encoded as UTF-8.
 |
 |      By default, each line in the text file is a new row in the resulting
DataFrame.
 |
 |      .. versionadded:: 1.6.0
 |
 |      Parameters
 |      ----------
 |      paths : str or list
 |          string, or list of strings, for input path(s).
 |
 |      Other Parameters
 |      ----------------
 |      Extra options
 |          For the extra options, refer to
 |          `Data Source Option <https://spark.apache.org/docs/latest/sql-data-
sources-text.html#data-source-option>`_
 |          in the version you use.
 |
 |          .. # noqa
 |
 |      Examples
 |      --------
 |      >>> df = spark.read.text('python/test_support/sql/text-test.txt')
 |      >>> df.collect()
 |      [Row(value='hello'), Row(value='this')]
 |      >>> df = spark.read.text('python/test_support/sql/text-test.txt',
wholetext=True)
 |      >>> df.collect()
 |      [Row(value='hello\nthis')]
 |
```

```
 |  ----------------------------------------------------------------------
 |  Data descriptors inherited from OptionUtils:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)
```

```
help(spark.read.csv)
```

```
Help on method csv in module pyspark.sql.readwriter:

csv(path, schema=None, sep=None, encoding=None, quote=None, escape=None,
comment=None, header=None, inferSchema=None, ignoreLeadingWhiteSpace=None,
ignoreTrailingWhiteSpace=None, nullValue=None, nanValue=None, positiveInf=None,
negativeInf=None, dateFormat=None, timestampFormat=None, maxColumns=None,
maxCharsPerColumn=None, maxMalformedLogPerPartition=None, mode=None,
columnNameOfCorruptRecord=None, multiLine=None, charToEscapeQuoteEscaping=None,
samplingRatio=None, enforceSchema=None, emptyValue=None, locale=None, lineSep=None,
pathGlobFilter=None, recursiveFileLookup=None, modifiedBefore=None,
modifiedAfter=None, unescapedQuoteHandling=None) method of
pyspark.sql.readwriter.DataFrameReader instance
    Loads a CSV file and returns the result as a  :class:`DataFrame`.

    This function will go through the input once to determine the input schema if
    ``inferSchema`` is enabled. To avoid going through the entire data once,
disable
    ``inferSchema`` option or specify the schema explicitly using ``schema``.

    .. versionadded:: 2.0.0

    Parameters
    ----------
    path : str or list
        string, or list of strings, for input path(s),
        or RDD of Strings storing CSV rows.
    schema : :class:`pyspark.sql.types.StructType` or str, optional
        an optional :class:`pyspark.sql.types.StructType` for the input schema
        or a DDL-formatted string (For example ``col0 INT, col1 DOUBLE``).

    Other Parameters
    ----------------
    Extra options
        For the extra options, refer to
        `Data Source Option <https://spark.apache.org/docs/latest/sql-data-sources-
csv.html#data-source-option>`_
        in the version you use.

        .. # noqa

    Examples
    --------
    >>> df = spark.read.csv('python/test_support/sql/ages.csv')
    >>> df.dtypes
    [('_c0', 'string'), ('_c1', 'string')]
    >>> rdd = sc.textFile('python/test_support/sql/ages.csv')
    >>> df2 = spark.read.csv(rdd)
    >>> df2.dtypes
    [('_c0', 'string'), ('_c1', 'string')]
```

```
scores = spark.read.csv(scores_csv)
```

```
scores
```

```
DataFrame[_c0: string, _c1: string, _c2: string, _c3: string, _c4: string, _c5:
string, _c6: string, _c7: string, _c8: string, _c9: string, _c10: string, _c11:
string]
```

```
help(scores.show)
```

```
Help on method show in module pyspark.sql.dataframe:

show(n=20, truncate=True, vertical=False) method of pyspark.sql.dataframe.DataFrame
instance
    Prints the first ``n`` rows to the console.

    .. versionadded:: 1.3.0

    Parameters
    ----------
    n : int, optional
        Number of rows to show.
    truncate : bool or int, optional
        If set to ``True``, truncate strings longer than 20 chars by default.
        If set to a number greater than one, truncates long strings to length
``truncate``
        and align cells right.
    vertical : bool, optional
        If set to ``True``, print output rows vertically (one line
        per column value).

    Examples
    --------
    >>> df
    DataFrame[age: int, name: string]
    >>> df.show()
    +---+-----+
    |age| name|
    +---+-----+
    |  2|Alice|
    |  5|  Bob|
    +---+-----+
    >>> df.show(truncate=3)
    +---+----+
    |age|name|
    +---+----+
    |  2| Ali|
    |  5| Bob|
    +---+----+
    >>> df.show(vertical=True)
    -RECORD 0-----
     age  | 2
     name | Alice
    -RECORD 1-----
     age  | 5
     name | Bob
```

We can look at the head of the DataFrame calling the `show` method.

scores.show()

Can anyone spot what's wrong with the above data?

- Question marks
- Column names
- `Float` and `Int` in the same column

Let's check the schema of our DataFrame

```
help(scores.printSchema)
```

```
Help on method printSchema in module pyspark.sql.dataframe:

printSchema() method of pyspark.sql.dataframe.DataFrame instance
    Prints out the schema in the tree format.

    .. versionadded:: 1.3.0

    Examples
    --------
    >>> df.printSchema()
    root
     |-- age: integer (nullable = true)
     |-- name: string (nullable = true)
    <BLANKLINE>
```

```
scores.printSchema()
```

```
root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
 |-- _c2: string (nullable = true)
 |-- _c3: string (nullable = true)
 |-- _c4: string (nullable = true)
 |-- _c5: string (nullable = true)
 |-- _c6: string (nullable = true)
 |-- _c7: string (nullable = true)
 |-- _c8: string (nullable = true)
 |-- _c9: string (nullable = true)
 |-- _c10: string (nullable = true)
 |-- _c11: string (nullable = true)
```

Why everythin is a `String`?

Managing Schema and Null Values

```
scores_df = (
    spark.read
        .option("header", "true")
        .option("nullValue", "?")
        .option("inferSchema", "true")
        .csv(scores_csv)
)
```

```
[Stage 2:>                                                          (0 + 1) / 1]
```

```
scores_df.printSchema()
```

```
root
 |-- id_1: integer (nullable = true)
 |-- id_2: integer (nullable = true)
 |-- cmp_fname_c1: double (nullable = true)
 |-- cmp_fname_c2: double (nullable = true)
 |-- cmp_lname_c1: double (nullable = true)
 |-- cmp_lname_c2: double (nullable = true)
 |-- cmp_sex: integer (nullable = true)
 |-- cmp_bd: integer (nullable = true)
 |-- cmp_bm: integer (nullable = true)
 |-- cmp_by: integer (nullable = true)
 |-- cmp_plz: integer (nullable = true)
 |-- is_match: boolean (nullable = true)
```

```
scores_df.show(5)
```

```
+-----+-----+-----------------+------------+------------+------------+-------+-----
-+------+------+-------+--------+
| id_1| id_2|
cmp_fname_c1|cmp_fname_c2|cmp_lname_c1|cmp_lname_c2|cmp_sex|cmp_bd|cmp_bm|cmp_by|cm
p_plz|is_match|
+-----+-----+-----------------+------------+------------+------------+-------+-----
-+------+------+-------+--------+
|37291|53113|0.833333333333333|        null|         1.0|        null|      1|
1|     1|     1|      0|    true|
|39086|47614|              1.0|        null|         1.0|        null|      1|
1|     1|     1|      1|    true|
|70031|70237|              1.0|        null|         1.0|        null|      1|
1|     1|     1|      1|    true|
|84795|97439|              1.0|        null|         1.0|        null|      1|
1|     1|     1|      1|    true|
|36950|42116|              1.0|        null|         1.0|         1.0|      1|
1|     1|     1|      1|    true|
+-----+-----+-----------------+------------+------------+------------+-------+-----
-+------+------+-------+--------+
only showing top 5 rows
```

Transformations and Actions

Creating a DataFrame does not cause any distributed computation in the cluster. **A DataFrame is un data set representing an intermediate step in a computation**.

For operatring data (in a distributed manner), we have two type of operations: **transformations** and **actions**:

- Transformations: lazy evaluation. They're not computed immediately, but they are recorded as a **lineage** for query play optimization.
- Actions: distributed computation occurs after invoking an action

```
# how many?
scores_df.count()
```

```
574913
```

We can use the `collect` action to return `Array` with all the `Row` objects in our DataFrame.

```
scores_df.collect()
```

```
[Stage 7:>                                                          (0 + 1) / 1]
```

```
[Row(id_1=37291, id_2=53113, cmp_fname_c1=0.833333333333333, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=0, is_match=True),
 Row(id_1=39086, id_2=47614, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=70031, id_2=70237, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=84795, id_2=97439, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36950, id_2=42116, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42413, id_2=48491, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25965, id_2=64753, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=49451, id_2=90407, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=39932, id_2=40902, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46626, id_2=47940, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=48948, id_2=98379, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=4767, id_2=4826, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=45463, id_2=69659, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=11367, id_2=13169, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=10782, id_2=89636, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=0, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26206, id_2=39147, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16662, id_2=27083, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18823, id_2=30204, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38545, id_2=85418, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28963, id_2=39172, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38918, id_2=44578, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23499, id_2=40452, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46469, id_2=94155, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25816, id_2=29809, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28777, id_2=39482, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18914, id_2=18916, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=15105, id_2=16971, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=75313, id_2=76006, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29624, id_2=90371, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=30127, id_2=30732, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=3791, id_2=61656, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=89734, id_2=89736, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42161, id_2=44157, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=94601, id_2=94602, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=14532, id_2=75581, cmp_fname_c1=0.888888888888889, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=None, is_match=True),
 Row(id_1=34195, id_2=35649, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=13862, id_2=99105, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=68990, id_2=68991, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36430, id_2=47306, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20417, id_2=87185, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27458, id_2=33456, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=53806, id_2=57851, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30216, id_2=69283, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=7275, id_2=8294, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=15821, id_2=16913, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31625, id_2=38784, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17146, id_2=40456, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=90142, id_2=90144, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16615, id_2=29839, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16628, id_2=22975, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=13297, id_2=84993, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=44473, id_2=44474, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26756, id_2=61838, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=60032, id_2=65967, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=84808, id_2=84810, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19845, id_2=41058, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5947, id_2=72075, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18313, id_2=24363, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=17676, id_2=44054, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=10784, id_2=14152, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=10996, id_2=72815, cmp_fname_c1=0.25, cmp_fname_c2=None,
cmp_lname_c1=0.25, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=1, is_match=True),
 Row(id_1=43920, id_2=72193, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28464, id_2=69899, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25306, id_2=28302, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=13260, id_2=34548, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=83862, id_2=85842, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19563, id_2=30319, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41967, id_2=83896, cmp_fname_c1=0.875, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=1, is_match=True),
 Row(id_1=1763, id_2=64724, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38604, id_2=44034, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=43214, id_2=48853, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=48021, id_2=95459, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=3751, id_2=3752, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=11585, id_2=11972, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=1857, id_2=1884, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38118, id_2=59788, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17431, id_2=20446, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33395, id_2=49625, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27885, id_2=32872, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29875, id_2=42368, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23243, id_2=76268, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=48837, id_2=63106, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35400, id_2=43913, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2702, id_2=5962, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35709, id_2=43859, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=9885, id_2=86926, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=4375, id_2=61891, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=35701, id_2=84881, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27907, id_2=41092, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=48106, id_2=72933, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=0, cmp_bm=0, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34697, id_2=41334, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=76874, id_2=79017, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35670, id_2=48804, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=69439, id_2=69448, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=45793, id_2=94527, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31899, id_2=37198, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46074, id_2=58332, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=96438, id_2=96451, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17247, id_2=26452, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17690, id_2=22832, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36098, id_2=52230, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=21325, id_2=59892, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42976, id_2=66077, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=5237, id_2=70862, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16472, id_2=64283, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26353, id_2=48169, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=44394, id_2=88861, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41154, id_2=54204, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25738, id_2=34891, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30066, id_2=31819, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40080, id_2=93119, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30242, id_2=41430, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=14820, id_2=41337, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29446, id_2=43844, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=1806, id_2=1841, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=49842, id_2=49901, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=34563, id_2=44884, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33375, id_2=36820, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=15516, id_2=59867, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=7182, id_2=7464, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21362, id_2=31859, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=73204, id_2=79717, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=9675, id_2=12966, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=95061, id_2=95062, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22181, id_2=93515, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=54522, id_2=57955, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=60061, id_2=69354, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=53795, id_2=60343, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36671, id_2=78410, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=8709, id_2=86907, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22914, id_2=27526, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=43915, id_2=63663, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=23841, id_2=47307, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=43371, id_2=88853, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=95169, id_2=96851, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34902, id_2=36439, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18578, id_2=42842, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32467, id_2=46952, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5591, id_2=12689, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16899, id_2=39730, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29154, id_2=37137, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46608, id_2=72200, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36769, id_2=66551, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=70036, id_2=70105, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33837, id_2=65099, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=20670, id_2=31856, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=61649, id_2=61650, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=37668, id_2=54114, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17101, id_2=28010, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30032, id_2=40979, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40652, id_2=81009, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34391, id_2=45453, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36276, id_2=80445, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2935, id_2=66776, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=58410, id_2=62593, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=74914, id_2=83540, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29970, id_2=33037, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=12259, id_2=27752, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25011, id_2=48687, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16366, id_2=18740, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39219, id_2=48188, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36582, id_2=36583, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2429, id_2=7016, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16483, id_2=44045, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19420, id_2=24675, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38035, id_2=46626, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=44992, id_2=98411, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19399, id_2=55603, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23675, id_2=47325, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=55746, id_2=66041, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=24347, id_2=98319, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17876, id_2=39365, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19139, id_2=57328, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16714, id_2=28977, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=64640, id_2=64663, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=50519, id_2=88880, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26586, id_2=55632, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20519, id_2=48517, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=57469, id_2=57493, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=86982, id_2=92793, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28178, id_2=30983, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=0, cmp_plz=1,
is_match=True),
 Row(id_1=23803, id_2=64324, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35260, id_2=47619, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5387, id_2=6224, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25949, id_2=60398, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20226, id_2=23314, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=602, id_2=11685, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=92058, id_2=92060, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16849, id_2=16850, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26306, id_2=27069, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26391, id_2=34318, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=63818, id_2=63819, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=43247, id_2=43425, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28198, id_2=29133, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20791, id_2=53616, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29686, id_2=46077, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46999, id_2=47898, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33350, id_2=37708, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=45998, id_2=48186, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20237, id_2=22805, cmp_fname_c1=0.8, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34622, id_2=38557, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=11949, id_2=90217, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=44820, id_2=45170, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=53407, id_2=72438, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=97906, id_2=97908, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=43712, id_2=47848, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27383, id_2=56085, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=43121, id_2=87888, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36148, id_2=83788, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=79145, id_2=79512, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19979, id_2=54219, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19437, id_2=21685, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5232, id_2=9096, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27388, id_2=32254, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23021, id_2=27965, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=48394, id_2=94700, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41619, id_2=55165, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22505, id_2=25630, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=52735, id_2=52736, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=37069, id_2=50644, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=68894, id_2=68895, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42179, id_2=49006, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=63967, id_2=69198, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32288, id_2=60704, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24968, id_2=27220, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39419, id_2=70320, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=1910, id_2=1911, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=11749, id_2=11750, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=4526, id_2=4527, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25853, id_2=34935, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28308, id_2=29605, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23344, id_2=31571, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=24794, id_2=32043, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31867, id_2=66588, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32396, id_2=76097, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32941, id_2=43719, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=75103, id_2=98319, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31400, id_2=35079, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=67564, id_2=70134, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40612, id_2=43953, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29855, id_2=31092, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=52998, id_2=53931, cmp_fname_c1=0.888888888888889, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=1, is_match=True),
 Row(id_1=31299, id_2=64387, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16735, id_2=39325, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18368, id_2=23121, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21037, id_2=23421, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26111, id_2=32515, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=54400, id_2=99470, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=61454, id_2=86296, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=13961, id_2=13962, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=64540, id_2=98201, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=3357, id_2=14253, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16063, id_2=30837, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=68824, id_2=68825, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19651, id_2=24224, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26003, id_2=66696, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23150, id_2=23617, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=37785, id_2=71137, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21590, id_2=23574, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5541, id_2=30805, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23777, id_2=39379, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=46712, id_2=48895, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19489, id_2=24546, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20244, id_2=23112, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32487, id_2=61025, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38327, id_2=80814, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18284, id_2=21122, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20885, id_2=28431, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18557, id_2=51346, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25641, id_2=33507, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38284, id_2=43351, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42473, id_2=77470, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=1443, id_2=50993, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=0, cmp_bm=0, cmp_by=0, cmp_plz=1,
is_match=True),
 Row(id_1=23660, id_2=45960, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17470, id_2=27076, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=3995, id_2=8136, cmp_fname_c1=0.8, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40022, id_2=41819, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41786, id_2=45527, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24498, id_2=24924, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25565, id_2=29549, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40715, id_2=40966, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=11232, id_2=12689, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=82435, id_2=82437, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17427, id_2=17681, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26224, id_2=29550, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=1285, id_2=37362, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=0, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24000, id_2=27538, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29341, id_2=36402, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31942, id_2=62256, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=92167, id_2=92169, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=49729, id_2=92089, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=50629, id_2=59769, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20569, id_2=53434, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21121, id_2=49540, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38858, id_2=90187, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=44605, id_2=47245, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21990, id_2=22749, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22702, id_2=54006, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=9738, id_2=90175, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=186, id_2=289, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=0.9,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=10357, id_2=11108, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23961, id_2=28388, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=45851, id_2=47988, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=90688, id_2=90689, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27794, id_2=32772, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22236, id_2=26621, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33996, id_2=42465, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17091, id_2=22115, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=1897, id_2=36282, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=50130, id_2=55615, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22713, id_2=48001, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22395, id_2=30474, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=59566, id_2=61228, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25070, id_2=26532, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40707, id_2=47270, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=62471, id_2=90737, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34769, id_2=46880, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=76010, id_2=76150, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38158, id_2=49607, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=30041, id_2=76138, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=82153, id_2=82154, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29826, id_2=34707, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40294, id_2=66086, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=32673, id_2=43931, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=57061, id_2=57062, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19447, id_2=24533, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=65692, id_2=74896, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17905, id_2=18706, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40300, id_2=72312, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=53855, id_2=57443, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=56272, id_2=56636, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35060, id_2=75119, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21456, id_2=28411, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=7373, id_2=77467, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22595, id_2=50421, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21392, id_2=31364, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23696, id_2=26446, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39875, id_2=47054, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=6502, id_2=10378, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24235, id_2=53126, cmp_fname_c1=0.875, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=0, is_match=True),
 Row(id_1=33076, id_2=37960, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=71570, id_2=77788, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22397, id_2=64389, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35598, id_2=78035, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=47315, id_2=48166, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=90927, id_2=97474, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33488, id_2=39378, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20117, id_2=22423, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=23073, id_2=36855, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=15387, id_2=26208, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23775, id_2=59331, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=9448, id_2=9450, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5449, id_2=35434, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32494, id_2=33617, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5599, id_2=99138, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22535, id_2=27083, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22079, id_2=28195, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24502, id_2=56085, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=70554, id_2=95607, cmp_fname_c1=None, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=0,
cmp_plz=0, is_match=True),
 Row(id_1=24762, id_2=31256, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16269, id_2=60498, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39634, id_2=47956, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=997, id_2=55903, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=60384, id_2=60385, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=10737, id_2=10738, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20672, id_2=29548, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17641, id_2=46042, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=41967, id_2=85682, cmp_fname_c1=0.875, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=1, is_match=True),
 Row(id_1=25964, id_2=66481, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=37257, id_2=70423, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27457, id_2=27458, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=70517, id_2=75766, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=10744, id_2=92906, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25282, id_2=44441, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16467, id_2=28735, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16088, id_2=23919, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2855, id_2=65169, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=34355, id_2=63181, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35164, id_2=84066, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=87785, id_2=89119, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22482, id_2=50106, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38482, id_2=74742, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24414, id_2=46042, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=18423, id_2=21547, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39933, id_2=42741, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=14962, id_2=14965, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=4914, id_2=59133, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=14586, id_2=87818, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17182, id_2=33452, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=90697, id_2=90698, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2481, id_2=2482, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38173, id_2=75342, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39326, id_2=48511, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=855, id_2=1051, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=74534, id_2=74535, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36010, id_2=76798, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25434, id_2=32078, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=58532, id_2=63818, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=11420, id_2=11421, cmp_fname_c1=1.0, cmp_fname_c2=None,
cmp_lname_c1=0.923076923076923, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1,
cmp_by=1, cmp_plz=1, is_match=True),
 Row(id_1=20842, id_2=27795, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=86540, id_2=88794, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=0, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=4674, id_2=8441, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24482, id_2=39885, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=95226, id_2=95304, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24688, id_2=31468, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20936, id_2=80134, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=20879, id_2=54252, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21138, id_2=32885, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=94328, id_2=94356, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=82102, id_2=82103, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=54824, id_2=54912, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24925, id_2=27380, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5510, id_2=11945, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=85905, id_2=92162, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=33717, id_2=34574, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2918, id_2=4306, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38621, id_2=44930, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17951, id_2=18276, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=171, id_2=175, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30447, id_2=50601, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31458, id_2=67251, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2760, id_2=38702, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29419, id_2=38138, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17894, id_2=35515, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=61065, id_2=99469, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=47481, id_2=90263, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21711, id_2=57493, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17287, id_2=22744, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16522, id_2=59759, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25398, id_2=35288, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=97432, id_2=99254, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46864, id_2=48891, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=64225, id_2=65335, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=45779, id_2=94316, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=76018, id_2=76163, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=22356, id_2=61618, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20419, id_2=36118, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=23826, id_2=65085, cmp_fname_c1=0.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19981, id_2=24931, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36132, id_2=44759, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26784, id_2=45247, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25333, id_2=38019, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=56156, id_2=56773, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2204, id_2=2783, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27630, id_2=31624, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21323, id_2=22913, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=8191, id_2=92167, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=59857, id_2=59861, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=6389, id_2=76817, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=9175, id_2=9176, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17529, id_2=48504, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=15818, id_2=17435, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21854, id_2=38046, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=469, id_2=2395, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34321, id_2=49882, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=56767, id_2=92361, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=63374, id_2=63872, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=57090, id_2=57102, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=82498, id_2=83478, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28736, id_2=83734, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24996, id_2=47686, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=54905, id_2=58039, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=69883, id_2=92811, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=45657, id_2=49484, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=83867, id_2=83868, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27509, id_2=30467, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=84438, id_2=92235, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=37874, id_2=37875, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36602, id_2=49481, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18821, id_2=20498, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27759, id_2=29777, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=54511, id_2=57951, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=63213, id_2=90027, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=14851, id_2=76283, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26411, id_2=29391, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=12412, id_2=40085, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32153, id_2=70347, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=19868, id_2=40707, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23107, id_2=60797, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26391, id_2=42564, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20854, id_2=20855, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33722, id_2=46074, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=75523, id_2=94687, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=43944, id_2=48596, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2644, id_2=30213, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=67983, id_2=67985, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=47222, id_2=49269, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21786, id_2=48156, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20385, id_2=22076, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=60404, id_2=62256, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=53159, id_2=54401, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=37317, id_2=80774, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46682, id_2=82491, cmp_fname_c1=0.888888888888889, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=1, is_match=True),
```

```
 Row(id_1=30581, id_2=66655, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=15918, id_2=28652, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=237, id_2=256, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34135, id_2=36610, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=58317, id_2=63687, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=66149, id_2=69400, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31532, id_2=44889, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=93770, id_2=94477, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=45164, id_2=91906, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=68961, id_2=68962, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34823, id_2=39268, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24873, id_2=43411, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=66023, id_2=69314, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16177, id_2=21741, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=56288, id_2=56602, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=62457, id_2=62458, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=68669, id_2=68670, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38831, id_2=64391, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30176, id_2=71082, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2697, id_2=4019, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=60038, id_2=63882, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=81363, id_2=81364, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17148, id_2=44050, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22263, id_2=23867, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27585, id_2=40001, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21683, id_2=50569, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41172, id_2=44833, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28236, id_2=42795, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=86823, id_2=86836, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=9263, id_2=85266, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26443, id_2=28855, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25521, id_2=55757, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=16839, id_2=22717, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=84723, id_2=91792, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=71007, id_2=71069, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40266, id_2=44549, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=3193, id_2=81931, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2700, id_2=5260, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=64307, id_2=64309, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19236, id_2=24810, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2060, id_2=3624, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=79996, id_2=79997, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=55946, id_2=60389, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=6583, id_2=72861, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=31206, id_2=44047, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36137, id_2=49533, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16052, id_2=25846, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21186, id_2=25184, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27317, id_2=66599, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27377, id_2=67753, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=37622, id_2=64398, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16971, id_2=25515, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39586, id_2=74662, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=51860, id_2=57906, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26503, id_2=30240, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23005, id_2=29201, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23604, id_2=85030, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46584, id_2=64979, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=84821, id_2=84829, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=72659, id_2=84440, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26909, id_2=27982, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=56612, id_2=56618, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=78074, id_2=86885, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=4402, id_2=67212, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36509, id_2=38124, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2157, id_2=3618, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23461, id_2=34211, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34424, id_2=46607, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34416, id_2=36823, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5801, id_2=6182, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22885, id_2=25791, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18311, id_2=19457, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19936, id_2=60242, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=11750, id_2=14574, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18184, id_2=51081, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20655, id_2=27332, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=1821, id_2=8226, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=10262, id_2=10446, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20157, id_2=35863, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=70970, id_2=72768, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30941, id_2=66066, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=29612, id_2=45517, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=58489, id_2=62491, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=840, id_2=843, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19174, id_2=21841, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=56258, id_2=56622, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=3172, id_2=8345, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=39702, id_2=46895, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18316, id_2=19031, cmp_fname_c1=1.0, cmp_fname_c2=None,
cmp_lname_c1=0.875, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=1, is_match=True),
 Row(id_1=23582, id_2=41127, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29664, id_2=35558, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=81214, id_2=90267, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26669, id_2=41100, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=1056, id_2=1239, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=0, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=1933, id_2=58629, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35381, id_2=42350, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=55873, id_2=81632, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=16378, id_2=20757, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17529, id_2=44066, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=43571, id_2=49210, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31562, id_2=61946, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20699, id_2=37692, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40074, id_2=79122, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30024, id_2=71656, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=73551, id_2=73601, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17540, id_2=19310, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=96482, id_2=96492, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=49062, id_2=94168, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=11501, id_2=89332, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=45305, id_2=96797, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36171, id_2=40894, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41301, id_2=44100, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30532, id_2=37427, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16015, id_2=73086, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=78232, id_2=78233, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=37389, id_2=38794, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=28412, id_2=42341, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=69206, id_2=70444, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30362, id_2=77427, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=44741, id_2=46452, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21013, id_2=94896, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40857, id_2=87229, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5244, id_2=7846, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41783, id_2=64241, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=71633, id_2=71634, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33571, id_2=34721, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=12769, id_2=14083, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=11286, id_2=94858, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35706, id_2=36376, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21875, id_2=41145, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2440, id_2=20397, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=712, id_2=718, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19799, id_2=84004, cmp_fname_c1=0.833333333333333, cmp_fname_c2=None,
cmp_lname_c1=0.888888888888889, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1,
cmp_by=1, cmp_plz=1, is_match=True),
 Row(id_1=62005, id_2=62006, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=12576, id_2=12577, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5607, id_2=79036, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23026, id_2=31989, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=71076, id_2=73512, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=62630, id_2=64206, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38422, id_2=79052, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=51864, id_2=57920, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16983, id_2=28235, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=52332, id_2=62020, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=58132, id_2=58133, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=67491, id_2=67539, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=67, id_2=11613, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21057, id_2=29394, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=906, id_2=90011, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=8972, id_2=84365, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31559, id_2=56895, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=13389, id_2=93725, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=3649, id_2=13838, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41816, id_2=43823, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=72996, id_2=96440, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35408, id_2=37910, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32186, id_2=49529, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2188, id_2=2189, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21697, id_2=22879, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=11141, id_2=20712, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=56731, id_2=62057, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24761, id_2=27346, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29779, id_2=33524, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21615, id_2=24084, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=64006, id_2=69236, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=65587, id_2=67392, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34301, id_2=42989, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16408, id_2=51707, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18054, id_2=38303, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39110, id_2=47684, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25499, id_2=44077, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38614, id_2=50347, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34937, id_2=80975, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=89644, id_2=96716, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=14403, id_2=14406, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=36688, id_2=42397, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46393, id_2=46394, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=597, id_2=736, cmp_fname_c1=0.875, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27298, id_2=27300, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30429, id_2=36465, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2300, id_2=7507, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22029, id_2=74275, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2491, id_2=2497, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32889, id_2=46587, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33965, id_2=81272, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17382, id_2=48008, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27308, id_2=35747, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38554, id_2=52212, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=23407, id_2=25092, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34694, id_2=43312, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=14462, id_2=14463, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42238, id_2=73473, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32580, id_2=33893, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24742, id_2=34190, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=68921, id_2=68922, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22459, id_2=27503, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=6682, id_2=9562, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=9853, id_2=10417, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=520, id_2=2183, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22830, id_2=44092, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28557, id_2=67757, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=53693, id_2=75212, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=87951, id_2=87956, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32683, id_2=35309, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=25009, id_2=62264, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5221, id_2=63226, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16120, id_2=40697, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5801, id_2=13979, cmp_fname_c1=0.818181818181818, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=1, is_match=True),
 Row(id_1=13152, id_2=76252, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22627, id_2=38924, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=61237, id_2=81192, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20709, id_2=30407, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=65253, id_2=65254, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16396, id_2=39034, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=37933, id_2=79325, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=12551, id_2=12552, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=12716, id_2=95361, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=55594, id_2=55595, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31835, id_2=44563, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22135, id_2=40711, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41311, id_2=46124, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20936, id_2=73334, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32861, id_2=72636, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22779, id_2=26349, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=56257, id_2=56617, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=48789, id_2=57761, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=70716, id_2=71181, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=67970, id_2=67973, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41336, id_2=82862, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5172, id_2=5173, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20073, id_2=21045, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19259, id_2=29557, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=44450, id_2=53998, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=38149, id_2=41760, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=6705, id_2=9038, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=3318, id_2=8473, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5586, id_2=11867, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=57816, id_2=69404, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=30321, id_2=44211, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35461, id_2=38801, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33839, id_2=38160, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33083, id_2=36898, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=15672, id_2=29731, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21422, id_2=24380, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28642, id_2=30796, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=13902, id_2=93858, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40349, id_2=78932, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26782, id_2=39607, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=15307, id_2=48799, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18961, id_2=24782, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=71953, id_2=72160, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39744, id_2=52245, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25566, id_2=27319, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=82035, id_2=82036, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16753, id_2=29385, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26261, id_2=54032, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39851, id_2=46237, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35667, id_2=36588, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39349, id_2=49315, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=2836, id_2=12619, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=63887, id_2=66122, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=16655, id_2=17876, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=30036, id_2=34413, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39607, id_2=43423, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17750, id_2=20905, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25038, id_2=47763, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=83122, id_2=97035, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17566, id_2=22965, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42184, id_2=80784, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23902, id_2=32364, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=51447, id_2=53880, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=33985, id_2=74742, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=15810, id_2=17961, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38212, id_2=84085, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27296, id_2=44771, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5329, id_2=73727, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=9866, id_2=13721, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=1419, id_2=1423, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5613, id_2=72410, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=90713, id_2=90716, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=4833, id_2=7092, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=0.75,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25071, id_2=49749, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19048, id_2=22155, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=4045, id_2=4046, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29355, id_2=43812, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=7726, id_2=38074, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=48288, id_2=93852, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=51003, id_2=51004, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32707, id_2=63697, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=55737, id_2=60046, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=30621, id_2=32254, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=7525, id_2=74588, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=48490, id_2=49367, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42247, id_2=90528, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=45037, id_2=46462, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=35305, id_2=69564, cmp_fname_c1=0.8, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26209, id_2=41153, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19437, id_2=74432, cmp_fname_c1=0.888888888888889, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=1, is_match=True),
 Row(id_1=279, id_2=10025, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=52135, id_2=52423, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=906, id_2=63808, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30476, id_2=90260, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20853, id_2=46027, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=68640, id_2=68641, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16306, id_2=36465, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=93075, id_2=93076, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24827, id_2=38525, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=52615, id_2=52849, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27980, id_2=27981, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=93357, id_2=93371, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=45028, id_2=46454, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28876, id_2=31351, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16823, id_2=42831, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16781, id_2=16782, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25588, id_2=27186, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=7713, id_2=84996, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=4602, id_2=69545, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=62940, id_2=62942, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28017, id_2=59968, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=15539, id_2=29108, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=9253, id_2=85230, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=53723, id_2=60260, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40663, id_2=49858, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22860, id_2=27741, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25543, id_2=28078, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39988, id_2=47445, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38640, id_2=75380, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29951, id_2=42361, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=75379, id_2=76069, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16888, id_2=26747, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=14125, id_2=61169, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18958, id_2=28411, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=53390, id_2=96528, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32380, id_2=36273, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=5571, id_2=29071, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=44149, id_2=45928, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39627, id_2=48592, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26905, id_2=38215, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=47606, id_2=77691, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36354, id_2=43795, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29104, id_2=39197, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25077, id_2=27014, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=6075, id_2=75650, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=8242, id_2=75945, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17167, id_2=22089, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33865, id_2=51110, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24200, id_2=44765, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30406, id_2=32710, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=3081, id_2=5522, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=20203, id_2=47015, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=40343, id_2=45900, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=83394, id_2=83395, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26543, id_2=47197, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23983, id_2=37910, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39684, id_2=77682, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16478, id_2=25294, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32764, id_2=69294, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=21071, id_2=23879, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34949, id_2=59324, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25333, id_2=34983, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=56709, id_2=57000, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24005, id_2=56806, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=60126, id_2=63453, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=6355, id_2=69383, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16132, id_2=16995, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17903, id_2=28885, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=15687, id_2=42949, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=55788, id_2=56359, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=276, id_2=281, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41365, id_2=86005, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=57036, id_2=69423, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=27186, id_2=60839, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=57409, id_2=78063, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27494, id_2=66624, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=38452, id_2=64733, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=71061, id_2=73496, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39702, id_2=49580, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26519, id_2=56384, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=71307, id_2=71316, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=62869, id_2=67675, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20839, id_2=61071, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=12744, id_2=12745, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28607, id_2=31292, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16378, id_2=39613, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18624, id_2=53196, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26695, id_2=36574, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=15119, id_2=50151, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17873, id_2=25821, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26865, id_2=35413, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33075, id_2=45561, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24810, id_2=60661, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=68896, id_2=68897, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34083, id_2=83716, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=6585, id_2=9842, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34401, id_2=37837, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31801, id_2=39248, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19102, id_2=33215, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=7993, id_2=86164, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25422, id_2=39604, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27889, id_2=34329, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32871, id_2=43930, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19717, id_2=27127, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=9464, id_2=9465, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24358, id_2=45536, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42660, id_2=84992, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46182, id_2=58223, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28307, id_2=29605, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=60036, id_2=69279, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22931, id_2=60354, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=4308, id_2=10887, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18534, id_2=26384, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32763, id_2=42711, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36727, id_2=40362, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23277, id_2=28961, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=59463, id_2=86741, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17725, id_2=19301, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24518, id_2=27267, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=53183, id_2=57326, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=3327, id_2=74373, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23512, id_2=32518, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=76288, id_2=90354, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=9736, id_2=93062, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41067, id_2=46056, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=97974, id_2=97992, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=75295, id_2=77345, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=65985, id_2=69281, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=6484, id_2=96587, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=0, cmp_plz=1,
is_match=True),
 Row(id_1=15048, id_2=50086, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33871, id_2=60724, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21747, id_2=24586, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=81819, id_2=81820, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=61905, id_2=63343, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=14661, id_2=86559, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=0.9,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21040, id_2=29201, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=32855, id_2=43844, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31654, id_2=49634, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=32657, id_2=43718, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=16454, id_2=40007, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=64014, id_2=69303, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=34540, id_2=41327, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=43572, id_2=46828, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=12215, id_2=77454, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=43686, id_2=49498, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31244, id_2=75312, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=9238, id_2=89768, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=93769, id_2=94477, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=39349, id_2=43817, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=52437, id_2=52462, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=20791, id_2=70361, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=12694, id_2=12695, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=34159, id_2=38166, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=47494, id_2=98855, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29469, id_2=34085, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21308, id_2=57753, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=97878, id_2=98540, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=11252, id_2=85246, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=37391, id_2=49849, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=81280, id_2=81309, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=58997, id_2=58998, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=8629, id_2=9948, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=72757, id_2=72758, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=44861, id_2=46826, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42196, id_2=85526, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=86189, id_2=86192, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46925, id_2=66679, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
```

```
 Row(id_1=29565, id_2=37883, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26637, id_2=49048, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26811, id_2=32460, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42052, id_2=99591, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=30664, id_2=46578, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24972, id_2=33418, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21060, id_2=46042, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=82731, id_2=85413, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=33646, id_2=83655, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=0, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=21855, id_2=36336, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18963, id_2=28601, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=49793, id_2=57772, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=44161, id_2=49297, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=13226, id_2=14814, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=51803, id_2=51804, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=20310, id_2=30619, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18160, id_2=33246, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=65845, id_2=65851, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=17164, id_2=40701, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=29620, id_2=38124, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=23326, id_2=48780, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=21685, id_2=74432, cmp_fname_c1=0.888888888888889, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=1, is_match=True),
 Row(id_1=12250, id_2=12361, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22637, id_2=52379, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=82075, id_2=82076, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46254, id_2=59754, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=26802, id_2=64387, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=19034, id_2=43837, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=41370, id_2=42161, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
```

```
 Row(id_1=35327, id_2=47055, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=50928, id_2=50929, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=86259, id_2=86260, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=31413, id_2=39918, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25518, id_2=31391, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=18033, id_2=30504, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=87985, id_2=87990, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=22128, id_2=23052, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=28910, id_2=32363, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=90239, id_2=91588, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=24369, id_2=31627, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=27525, id_2=29593, cmp_fname_c1=1.0, cmp_fname_c2=1.0, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=38456, id_2=44565, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=56197, id_2=88150, cmp_fname_c1=1.0, cmp_fname_c2=None,
cmp_lname_c1=0.888888888888889, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1,
cmp_by=1, cmp_plz=0, is_match=True),
 ...]
```

The `Array` will reside in local memory!!

Write to Disk

We are going to save the DataFrame into a different format: Parquet

```
scores_df.write.format("parquet").save("data/scores-parquet")
```

```
[Stage 8:>                                                          (0 + 1) / 1]
```

```
! ls data/scores-parquet
```

```
_SUCCESS  part-00000-9f997247-811a-40fc-a71a-6b5da6e3e29f-c000.snappy.parquet
```

```
scores_parquet = spark.read.parquet("data/scores-parquet")
```

```
scores_parquet.printSchema()
```

```
root
 |-- id_1: integer (nullable = true)
 |-- id_2: integer (nullable = true)
 |-- cmp_fname_c1: double (nullable = true)
 |-- cmp_fname_c2: double (nullable = true)
 |-- cmp_lname_c1: double (nullable = true)
 |-- cmp_lname_c2: double (nullable = true)
 |-- cmp_sex: integer (nullable = true)
 |-- cmp_bd: integer (nullable = true)
 |-- cmp_bm: integer (nullable = true)
 |-- cmp_by: integer (nullable = true)
 |-- cmp_plz: integer (nullable = true)
 |-- is_match: boolean (nullable = true)
```

```
scores_parquet.show(5)
```

```
+-----+-----+-----------------+------------+------------+------------+-------+-----
-+------+------+-------+--------+
| id_1| id_2|
cmp_fname_c1|cmp_fname_c2|cmp_lname_c1|cmp_lname_c2|cmp_sex|cmp_bd|cmp_bm|cmp_by|cm
p_plz|is_match|
+-----+-----+-----------------+------------+------------+------------+-------+-----
-+------+------+-------+--------+
|37291|53113|0.833333333333333|        null|         1.0|        null|      1|
1|     1|     1|      0|    true|
|39086|47614|              1.0|        null|         1.0|        null|      1|
1|     1|     1|      1|    true|
|70031|70237|              1.0|        null|         1.0|        null|      1|
1|     1|     1|      1|    true|
|84795|97439|              1.0|        null|         1.0|        null|      1|
1|     1|     1|      1|    true|
|36950|42116|              1.0|        null|         1.0|         1.0|      1|
1|     1|     1|      1|    true|
+-----+-----+-----------------+------------+------------+------------+-------+-----
-+------+------+-------+--------+
only showing top 5 rows
```

Analyzing Data

All good for now, but we don't load data for the sake of i, we do it because we want to run some analysis.

- First two column are Integer IDs. There represent the patients that were matched in the record.
- The next nine column are numeric values (int and double). They represnt match scores on different fields, such name, sex, birthday, and locations.
- The last column is a boolean value indicating whether or not the pair of patient records represented by the line was a match.

We could use this dataset to build a simple classifier that allows us to predict whether a record will be a match based on the values of the match scores for patient records.

Caching

Each time we process data (e.g., calling the `collect` method), Spark re-opens the file, parsea the rows, and then execute the requested action. It does not matter if we have filtered the data and created a smaller set of record.

We can use the `cache` method to indicate to store the DataFrame in memory.

```
help(scores_df.cache)
```

```
Help on method cache in module pyspark.sql.dataframe:

cache() method of pyspark.sql.dataframe.DataFrame instance
    Persists the :class:`DataFrame` with the default storage level
(`MEMORY_AND_DISK`).

    .. versionadded:: 1.3.0

    Notes
    -----
    The default storage level has changed to `MEMORY_AND_DISK` to match Scala in
2.0.
```

Spark is in-memory only. Myth or misconception?

“Spill” Storage levels:

- MEMORY_AND_DISK
- MEMORY
- MEMORY_SER

```
scores_cached = scores_df.cache()
```

```
scores_cached.count()
```

```
[Stage 11:>                                                         (0 + 1) / 1]
```

```
574913
```

```
scores_cached.take(10)
```

```
[Row(id_1=37291, id_2=53113, cmp_fname_c1=0.833333333333333, cmp_fname_c2=None,
cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=0, is_match=True),
 Row(id_1=39086, id_2=47614, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=70031, id_2=70237, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=84795, id_2=97439, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=36950, id_2=42116, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=42413, id_2=48491, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=25965, id_2=64753, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=49451, id_2=90407, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=0,
is_match=True),
 Row(id_1=39932, id_2=40902, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True),
 Row(id_1=46626, id_2=47940, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_lname_c1=1.0,
cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1, cmp_plz=1,
is_match=True)]
```

Query Plan

```
scores_cached.explain()
```

```
== Physical Plan ==
FileScan csv
[id_1#56,id_2#57,cmp_fname_c1#58,cmp_fname_c2#59,cmp_lname_c1#60,cmp_lname_c2#61,cm
p_sex#62,cmp_bd#63,cmp_bm#64,cmp_by#65,cmp_plz#66,is_match#67] Batched: false,
DataFilters: [], Format: CSV, Location: InMemoryFileIndex(1 paths)
[file:/home/runner/work/bda-course/bda-course/coursebook/modules/m2/dat...,
PartitionFilters: [], PushedFilters: [], ReadSchema:
struct<id_1:int,id_2:int,cmp_fname_c1:double,cmp_fname_c2:double,cmp_lname_c1:doubl
e,cmp_lname_c2...
```

GroupBy + OrderBy

```
from pyspark.sql.functions import col

scores_cached.groupBy("is_match").count().orderBy(col("count").desc()).show()
```

```
+--------+------+
|is_match| count|
+--------+------+
|   false|572820|
|    true|  2093|
+--------+------+
```

Aggregation Functions

In addition to `count`, we can also compute more complex aggregation like sums, mins, maxes, means, and standard deviation. How? we use `agg` method of the DataFrame API.

```
from pyspark.sql.functions import avg, stddev
```

```
aggregated = scores_cached.agg(avg("cmp_sex"), stddev("cmp_sex"))
```

```
aggregated.explain()
```

```
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- HashAggregate(keys=[], functions=[avg(cmp_sex#62), stddev_samp(cast(cmp_sex#62
as double))])
   +- Exchange SinglePartition, ENSURE_REQUIREMENTS, [id=#220]
      +- HashAggregate(keys=[], functions=[partial_avg(cmp_sex#62),
partial_stddev_samp(cast(cmp_sex#62 as double))])
         +- InMemoryTableScan [cmp_sex#62]
               +- InMemoryRelation [id_1#56, id_2#57, cmp_fname_c1#58,
cmp_fname_c2#59, cmp_lname_c1#60, cmp_lname_c2#61, cmp_sex#62, cmp_bd#63,
cmp_bm#64, cmp_by#65, cmp_plz#66, is_match#67], StorageLevel(disk, memory,
deserialized, 1 replicas)
                     +- FileScan csv
[id_1#56,id_2#57,cmp_fname_c1#58,cmp_fname_c2#59,cmp_lname_c1#60,cmp_lname_c2#61,cm
p_sex#62,cmp_bd#63,cmp_bm#64,cmp_by#65,cmp_plz#66,is_match#67] Batched: false,
DataFilters: [], Format: CSV, Location: InMemoryFileIndex(1 paths)
[file:/home/runner/work/bda-course/bda-course/coursebook/modules/m2/dat...,
PartitionFilters: [], PushedFilters: [], ReadSchema:
struct<id_1:int,id_2:int,cmp_fname_c1:double,cmp_fname_c2:double,cmp_lname_c1:doubl
e,cmp_lname_c2...
```

```
aggregated.show()
```

```
+------------------+--------------------+
|      avg(cmp_sex)|stddev_samp(cmp_sex)|
+------------------+--------------------+
|0.9550923357099248| 0.20710152240504734|
+------------------+--------------------+
```

SQL

ANSI 2003-compliant version or HiveQL.

```
scores_df.createOrReplaceTempView("scores")
```

```
# scores_cached.groupBy("is_match").count().orderBy(col("count").desc()).show()
spark.sql("""
    SELECT is_match, COUNT(*) cnt
    FROM scores
    GROUP BY is_match
    ORDER BY cnt DESC
""").show()
```

```
+--------+------+
|is_match|   cnt|
+--------+------+
|   false|572820|
|    true|  2093|
+--------+------+
```

```
spark.sql("""
    SELECT is_match, COUNT(*) cnt
    FROM scores
    GROUP BY is_match
    ORDER BY cnt DESC
""").explain()
```

```
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- Sort [cnt#2165L DESC NULLS LAST], true, 0
   +- Exchange rangepartitioning(cnt#2165L DESC NULLS LAST, 200),
ENSURE_REQUIREMENTS, [id=#330]
      +- HashAggregate(keys=[is_match#67], functions=[count(1)])
         +- Exchange hashpartitioning(is_match#67, 200), ENSURE_REQUIREMENTS,
[id=#327]
            +- HashAggregate(keys=[is_match#67], functions=[partial_count(1)])
               +- InMemoryTableScan [is_match#67]
                     +- InMemoryRelation [id_1#56, id_2#57, cmp_fname_c1#58,
cmp_fname_c2#59, cmp_lname_c1#60, cmp_lname_c2#61, cmp_sex#62, cmp_bd#63,
cmp_bm#64, cmp_by#65, cmp_plz#66, is_match#67], StorageLevel(disk, memory,
deserialized, 1 replicas)
                           +- FileScan csv
[id_1#56,id_2#57,cmp_fname_c1#58,cmp_fname_c2#59,cmp_lname_c1#60,cmp_lname_c2#61,cm
p_sex#62,cmp_bd#63,cmp_bm#64,cmp_by#65,cmp_plz#66,is_match#67] Batched: false,
DataFilters: [], Format: CSV, Location: InMemoryFileIndex(1 paths)
[file:/home/runner/work/bda-course/bda-course/coursebook/modules/m2/dat...,
PartitionFilters: [], PushedFilters: [], ReadSchema:
struct<id_1:int,id_2:int,cmp_fname_c1:double,cmp_fname_c2:double,cmp_lname_c1:doubl
e,cmp_lname_c2...
```

```
scores_cached.groupBy("is_match").count().orderBy(col("count").desc()).explain()
```

```
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- Sort [count#2364L DESC NULLS LAST], true, 0
   +- Exchange rangepartitioning(count#2364L DESC NULLS LAST, 200),
ENSURE_REQUIREMENTS, [id=#350]
      +- HashAggregate(keys=[is_match#67], functions=[count(1)])
         +- Exchange hashpartitioning(is_match#67, 200), ENSURE_REQUIREMENTS,
[id=#347]
            +- HashAggregate(keys=[is_match#67], functions=[partial_count(1)])
               +- InMemoryTableScan [is_match#67]
                     +- InMemoryRelation [id_1#56, id_2#57, cmp_fname_c1#58,
cmp_fname_c2#59, cmp_lname_c1#60, cmp_lname_c2#61, cmp_sex#62, cmp_bd#63,
cmp_bm#64, cmp_by#65, cmp_plz#66, is_match#67], StorageLevel(disk, memory,
deserialized, 1 replicas)
                           +- FileScan csv
[id_1#56,id_2#57,cmp_fname_c1#58,cmp_fname_c2#59,cmp_lname_c1#60,cmp_lname_c2#61,cm
p_sex#62,cmp_bd#63,cmp_bm#64,cmp_by#65,cmp_plz#66,is_match#67] Batched: false,
DataFilters: [], Format: CSV, Location: InMemoryFileIndex(1 paths)
[file:/home/runner/work/bda-course/bda-course/coursebook/modules/m2/dat...,
PartitionFilters: [], PushedFilters: [], ReadSchema:
struct<id_1:int,id_2:int,cmp_fname_c1:double,cmp_fname_c2:double,cmp_lname_c1:doubl
e,cmp_lname_c2...
```

Should I use Spark SQL or the DataFrame API

Depends on the query.

Pandas is my friend!

required packages: `pandas` and `numpy`

- poetry add pandas numpy
- pip install pandas numpy

```
scores_pandas = scores_df.toPandas()
```

```
[Stage 24:>                                                         (0 + 1) / 1]
```

```
scores_pandas.head()
```

	id_1	id_2	cmp_fname_c1	cmp_fname_c2	cmp_lname_c1	cmp_lname_c2	cmp_
0	37291	53113	0.833333	NaN	1.0	NaN	
1	39086	47614	1.000000	NaN	1.0	NaN	
2	70031	70237	1.000000	NaN	1.0	NaN	
3	84795	97439	1.000000	NaN	1.0	NaN	
4	36950	42116	1.000000	NaN	1.0	1.0	

```
scores_pandas.shape
```

```
(574913, 12)
```

References

[1] Irene Schmidtmann, Gaël Hammer, Murat Sariyar, Aslihan Gerhold-Ay, and Körperschaft des öffentlichen Rechts. Evaluation des krebsregisters nrw schwerpunkt record linkage. *Abschlußbericht vom*, 2009.

[2] Murat Sariyar, Andreas Borg, and Klaus Pommerening. Controlling false match rates in record linkage using extreme value theory. *Journal of biomedical informatics*, 44(4):648–654, 2011.

By Carlos Montemuiño