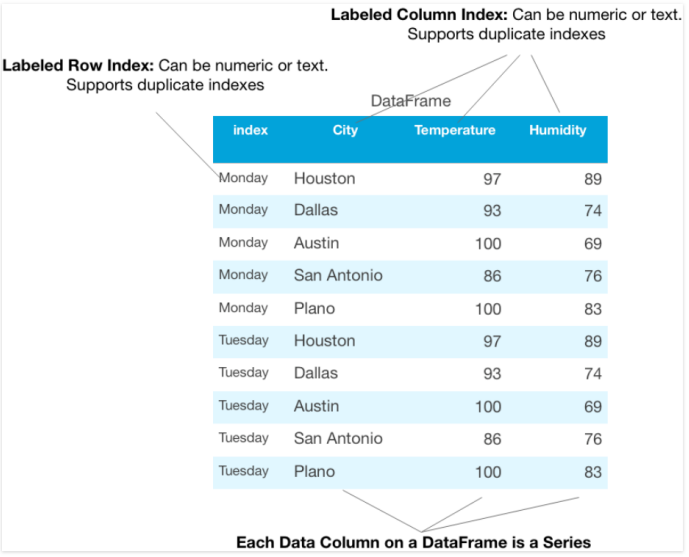
DataFrame

`DataFrame` objekti saab kõige paremini ette kujutada tavalise tabelina, sest `DataFrame` on andmete tabelistruktuur.

Igas tabelis on alati read ja veerud.

`DataFrame` objekti veergudeks on `Series` objektid, mille read on nende otsesed elemendid.



In [ ]:
```
import pandas as pd
```

In [ ]:
```
df_students = pd.DataFrame({
    'Student': ['John', 'Henry', 'Jane', 'James'],
    'Math': [87, 78, 95, 65],
    'Programming': [85, 88, 92, 72]
})
df_students
```

Out[ ]:

	Student	Math	Programming
0	John	87	85
1	Henry	78	88
2	Jane	95	92
3	James	65	72

Veerg `DataFrame` -s – see on `Series`

In [ ]:
```
type(df_students['Math'])
```

Out[ ]:
```
pandas.core.series.Series
```

In [ ]:
```
weather = [["Tallinn",-2,-3, 0],["Narva",1,-1,0],["Tartu",2,1,0]]
labels = ["City","Monday","Tuesday","Wednesday"]

df_weather = pd.DataFrame(weather, columns = labels)
df_weather
```

Out[ ]:

	City	Monday	Tuesday	Wednesday
0	Tallinn	-2	-3	0
1	Narva	1	-1	0
2	Tartu	2	1	0

`DataFrame` 'i objektil on 2 indeksit: ridade ja veergude järgi.

Kui ridade indeks ei ole selgelt määratud (nt veerg, mille järgi neid ehitada), siis pandas määrab täisavulise indeksi `RangeIndex` `0` kuni `N-1` , kus `N` on ridade arv tabelis. Atribuudi index abil saab vaadata, kuidas ridu nimetatakse. `shape` näitab, kui palju on andmefailis ridu ja veerge. See tagastab paar väärtust (n_rows, n_columns). Kõigepealt lähevad read `rows` , siis veerud `columns` vaadata veergude nimed

`dtypes` näitab, millist tüüpi andmed on igas veerus. On võimalik kindlaks teha, kas peab teisendama veerutüüpi, kuna juhtub, et numbrilised andmed laaditakse alla tekstina ja näiteks liitmisel võib tekkida probleeme. '1'+'1'='11'. Tüüp object on tekst.

In [ ]:
```
df_students.columns
```

Out[ ]:
```
Index(['Student', 'Math', 'Programming'], dtype='object')
```

In [ ]:
```
df_students.index
```

Out[ ]:
```
RangeIndex(start=0, stop=4, step=1)
```

In [ ]:
```
df_students.dtypes
```

Out[ ]:
```
Student        object
Math            int64
Programming     int64
dtype: object
```

In [ ]:
```
len(df_students)
```

Out[ ]:
```
4
```

Väärtuste hajumise, keskmise väärtuse ja mediaani teadasaamiseks kasutatakse `describe()` meetodit

In [ ]:
```
df_students.describe()
```

Out[ ]:

	Math	Programming
count	4.000000	4.000000
mean	81.250000	84.250000
std	12.867919	8.655441
min	65.000000	72.000000
25%	74.750000	81.750000
50%	82.500000	86.500000
75%	89.000000	89.000000
max	95.000000	92.000000

Samuti on võimalik `describe` meetodi abil saada teavet eraldi veeru kohta (numbri- ja reaandmete jaoks on kuvatakse erinev teave).

In [ ]:
```
df_students.Student.describe()
```

Out[ ]:
```
count         4
unique        4
top       Henry
freq          1
Name: Student, dtype: object
```

In [ ]:
```
df_students['Math'].describe()
```

Out[ ]:
```
count     4.000000
mean     81.250000
std      12.867919
min      65.000000
25%      74.750000
50%      82.500000
75%      89.000000
max      95.000000
Name: Math, dtype: float64
```

`info` - meetod, mis näitab kogu andmete (ridade) hulka, mitte-nullandmete arvu ja nende tüüpi.

In [ ]:
```
df_students.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Student      4 non-null      object
 1   Math         4 non-null      int64 
 2   Programming  4 non-null      int64 
dtypes: int64(2), object(1)
memory usage: 224.0+ bytes
```

Juurdepääs ridadele indeksi alusel on võimalik mitmel viisil:

- `loc` - kasutatakse juurdepääsuks reamärgi järgi
- `iloc` - kasutatakse juurdepääsuks arvväärtuse järgi (alates 0)

In [ ]:
```
df_students.iloc[0]
```

Out[ ]:
```
Student        John
Math             87
Programming      85
Name: 0, dtype: object
```

Andmete filtreerimine

Filtreerida `DataFrame` 'i read veeru väärtuste järgi

In [ ]:
```
df_students[df_students.Math > 90]
```

Out[ ]:

	Student	Math	Programming
2	Jane	95	92

In [ ]:
```
df_students[df_students['Math'] > 90]
```

Out[ ]:

	Student	Math	Programming
2	Jane	95	92

Veergude poole saab pöörduda, kasutades Pythoni sõnaraamatute atribuuti või notatsiooni.

In [ ]:
```
df_students[df_students.Math > 80][['Student', 'Math']]
```

Out[ ]:

	Student	Math
0	John	87
2	Jane	95

Võib kasutada ka päringukeelt `query` meetodi abil (& on and, | on or).

In [ ]:
```
df_students.query('Math > 80 & Programming > 80')
```

Out[ ]:

	Student	Math	Programming
0	John	87	85
2	Jane	95	92

Filtrite väärtused saab salvestada muutujasse ja päringus kasutada muutujat `@` sümboli abil.

In [ ]:
```
math_points = 80
programming_points = 90
df_students.query('Math > @math_points & Programming > @programming_points')
```

Out[ ]:

	Student	Math	Programming
2	Jane	95	92

Andmete lugemine ja kirjutamine¶

Pakett `pandas` toetab kõiki populaarsemaid andmete säilitamise formaate.

Andmete lugemine toimub `pandas.read_<type>` ja kirjutamine `DataFrame.to_<type>` meetodiga.

Nendel meetoditel on täiendavad parameetrid, mida saab vaadata ametlikust dokumentatsioonist: https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html (https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html)

In [ ]:
```
url = "https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population"
df_info_wiki = pd.read_html(url)
```

In [ ]:
```
df_population = df_info_wiki[0]
df_population.head()
```

Out[ ]:

	Rank	Country / Dependency	Region	Population	Percentage of the world	Date	Source (official or from the United Nations)	Notes
0	–	World	NaN	7939935000	100%	30 Mar 2022	UN projection[1]	NaN
1	1	China	Asia	1412600000	NaN	31 Dec 2021	National annual estimate[2]	The population figure refers to mainland China...
2	2	India	Asia	1374747725	NaN	30 Mar 2022	National population clock[3]	The figure includes the population of Jammu an...
3	3	United States	Americas	332591674	NaN	30 Mar 2022	National population clock[4]	The figure includes the 50 states and the Dist...
4	4	Indonesia	Asia[b]	272248500	NaN	1 Jul 2021	National annual estimate[5]	NaN

Kõige sagedamini tuleb töötada `csv` -failidega. Lugeda andmed csv -failist ja muuta DataFrame-iks saab `read_csv` funktsiooniga.

Kaggle —uuringukonkursside korraldamise süsteem ning andmetöötluse ja masinõppe spetsialistide suhtlusvõrgustik. See kuulub Google'i korporatsioonile (alates 2017. aasta märtsist). Keskkond on korraldatud avaliku veebiplatvormina, kus kasutajad ja organisatsioonid saavad avaldada andmekogumeid, uurida ja luua mudeleid, suhelda teiste andmespetsialistide ja masinõppe inseneridega, korraldada andmete uurimise konkursse ja osaleda neis. Süsteemi on paigutatud avaandmete kogumid ning esitatud pilvetööriistad andmete töötlemiseks ja masinõppeks. Realiseeritud on ka õpperessursse, on olemas tööandjate tööpakkumiste jaotis, kus samuti on võimalik konkursside korraldamine parimate kandidaatide valimiseks (wikipedia).

Fail programmeerimiskeelte kohta. Link: https://www.kaggle.com/datasets/fernandol/countries-of-the-world (https://www.kaggle.com/datasets/fernandol/countries-of-the-world)

Andmekogumi kohta (dataset)

Tavaline küsimus informaatika ja tarkvaraarenduse algajatele ja kursis olevatele inimestele: milline programmeerimiskeel on parim ja/või populaarseim? Ühetähenduslikku vastust on väga raske anda, sest on näiliselt määramatu arv näitajaid, mille järgi saab määrata "parimat" või "kõige populaarsemat" programmeerimiskeelt.

Üks selliseid näitajaid, mida saab kasutada "populaarse" programmeerimiskeele määramiseks, on projektide ja failide arv, mis on loodud seda programmeerimiskeelt kasutades. Kuna GitHub on kõige populaarsem üldkasutatav ühiskasutus- ja failijagamisplatvorm, on GitHubi hoidlate, projektide ja probleemide jaoks kasutatavate keelte analüüs hea keele populaarsuse näitaja.

Andmestik sisaldab statistikat programmeerimiskeelte kohta, mida kasutatakse GitHubi projektide loomisel. Andmed on aastatest 2011-2022.

Allikas: https://www.kaggle.com/datasets/isaacwen/github-programming-languages-data?select=prs.csv (https://www.kaggle.com/datasets/isaacwen/github-programming-languages-data?select=prs.csv)

Fail on salvestatud kaustas data

In [ ]:
```
df = pd.read_csv('data/projects_github.csv')
df.shape
```

Out[ ]:
```
(3462, 4)
```

3462 rida ja veergu DataFrame-is

`head` ja `tail` - saab vaadata esimesi n-ridu või viimaseid n-ridu (vaikimisi n=5). `sample` tagastab juhuslikud kirjed (vaikimisi n=1).

In [ ]:
```
df.head()
```

Out[ ]:

	name	year	quarter	count
0	Ruby	2011	3	632
1	PHP	2011	3	484
2	Python	2011	3	423
3	JavaScript	2011	3	367
4	Java	2011	3	216

In [ ]:
```
df.tail(3)
```

Out[ ]:

	name	year	quarter	count
3459	HTML	2022	1	134
3460	Scala	2022	1	110
3461	Shell	2022	1	103

In [ ]:
```
df.sample()
```

Out[ ]:

	name	year	quarter	count
780	PLpgSQL	2015	2	329

Teavet allalaaditud DataFrame kohta.

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3462 entries, 0 to 3461
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   name     3462 non-null   object
 1   year     3462 non-null   int64 
 2   quarter  3462 non-null   int64 
 3   count    3462 non-null   int64 
dtypes: int64(3), object(1)
memory usage: 108.3+ KB
```

Unikaalsete väärtuste saamiseks kasutatakse meetodit `unique`

In [ ]:
```
df['name'].unique()
```

Out[ ]:
```
array(['Ruby', 'PHP', 'Python', 'JavaScript', 'Java', 'C++', 'C', 'Shell',
       'C#', 'Objective-C', 'Scala', 'Erlang', 'HTML', 'Haskell',
       'Groovy', 'TypeScript', 'CoffeeScript', 'Go', 'Lua', 'Perl',
       'ColdFusion', 'Emacs Lisp', 'CSS', 'Clojure', 'Objective-J',
       'Elixir', 'Web Ontology Language', 'Puppet', 'XSLT', 'Vim script',
       'nesC', 'Nix', 'VimL', 'Delphi', 'Scheme', 'Jupyter Notebook',
       'TSQL', 'Common Lisp', 'Pascal', 'SQF', 'DM', 'Kotlin', 'OCaml',
       'Julia', 'Smalltalk', 'PowerShell', 'ActionScript', 'Dart', 'Vala',
       'Haxe', 'Swift', 'Raku', 'Rust', 'Makefile', 'TeX', 'R',
       'Assembly', 'Mako', 'D', 'Fortran', 'F#', 'Logos', 'PLSQL',
       'CMake', 'Apex', 'QML', 'Objective-C++', 'Pan', 'Tcl', 'Liquid',
       'SQLPL', 'Racket', 'SQL', 'Vim Snippet',
       'Common Workflow Language', 'Eagle', 'Protocol Buffer',
       'Standard ML', 'SCSS', 'XQuery', 'Dockerfile',
       'GCC Machine Description', 'PLpgSQL', 'Xtend', 'Matlab',
       'ApacheConf', 'Arduino', 'YAML', 'Visual Basic', 'Gherkin', 'JSON',
       'PureScript', 'ooc', 'MATLAB', 'PostScript', 'ASP', 'Thrift',
       'HCL', 'SourcePawn', 'Rich Text Format', 'Processing', 'SaltStack',
       'Elm', 'Batchfile', 'Perl 6', 'Smarty', 'Groff', 'Hack', 'CWeb',
       'AutoIt', 'Mustache', 'Perl6', 'Cuda', 'GDScript', 'Verilog',
       'Jsonnet', 'mIRC Script', 'Crystal', 'BitBake', 'Vue', 'Jinja',
       'Roff', 'IDL', 'Mathematica', 'Eiffel', 'OpenEdge ABL', 'LLVM',
       'GLSL', 'Markdown', 'Slash', 'FORTRAN', 'API Blueprint',
       'RobotFramework', 'WebAssembly', 'Nim', 'Cucumber',
       'SystemVerilog', 'Game Maker Language', 'NSIS', 'Nginx',
       'LiveScript', 'Rascal', 'Stylus', 'YARA', 'Ceylon', 'Modelica',
       'MoonScript', 'Prolog', 'Starlark', 'Zephir', 'SAS', 'GAP',
       'Visual Basic .NET', 'DIGITAL Command Language', 'M4', 'VHDL',
       'Coq', '1C Enterprise', 'AGS Script', 'Pug', 'ABAP', 'FreeMarker',
       'KiCad', 'AutoHotkey', 'Ada', 'Component Pascal', 'OpenSCAD',
       'SWIG', 'Uno', 'LabVIEW', 'F*', 'Meson', 'PureBasic', 'Nunjucks',
       'COBOL', 'UnrealScript', 'Twig', 'Lean', 'Genshi', 'Blade',
       'CodeQL', 'q', 'Pawn', 'Reason', 'POV-Ray SDL', 'BlitzBasic',
       'Svelte'], dtype=object)
```

In [ ]:
```
# unique values with drop_duplicates
df['name'].drop_duplicates().head(10)
```

Out[ ]:
```
0           Ruby
1            PHP
2         Python
3     JavaScript
4           Java
5            C++
12             C
13         Shell
14            C#
15    Objective-C
Name: name, dtype: object
```

Unikaalsete väärtuste loendi ja nende andmekogus esinemise sageduse vaatamiseks võib kasutada meetodit `value_counts()` .

In [ ]:
```
df['name'].value_counts().head(10)
```

Out[ ]:
```
Ruby          43
Python        43
JavaScript    43
Java          43
PHP           43
C++           43
HTML          42
C#            42
C             42
Shell         42
Name: name, dtype: int64
```

Rühmitamine ja sorteerimine

`groupby` meetod andmete rühmitamiseks ühe või mitme veeru järgi.

`sort_values` andmete sorteerimine. Vaikimisi kasvav sortimine (ascending=True), kahanev sortimine (ascending=False)

In [ ]:
```
df.groupby('name')['count'].sum().sample(10)
```

Out[ ]:
```
name
Java               3731712
Nginx                  630
Protocol Buffer       2140
AutoHotkey             496
POV-Ray SDL            315
Processing            1778
Eagle                 1307
MoonScript             101
OpenSCAD               225
HTML               1180280
Name: count, dtype: int64
```

Andmete rühmitamine programmeerimiskeele 'nimi' järgi ja veeru 'count' summa arvutamine

In [ ]:
```
df_sort = df.sort_values(by=["count"], ascending=False).head(10)
df_sort
```

Out[ ]:

	name	year	quarter	count
1497	JavaScript	2017	1	397530
1619	JavaScript	2017	2	388477
1373	JavaScript	2016	4	381082
1141	JavaScript	2016	2	373028
1259	JavaScript	2016	3	370929
1737	JavaScript	2017	3	356997
1018	JavaScript	2016	1	354843
1852	JavaScript	2017	4	334136
909	JavaScript	2015	4	276229
1965	JavaScript	2018	1	265358

Kahanev sorteerimine count veeru järgi.

Näide. Top 10 on kõige populaarsemad programmeerimiskeeled.

In [ ]:
```
df.groupby('name')['count'].sum().groupby(level=0).head().sort_values(ascending=False).head(10)
```

Out[ ]:
```
name
JavaScript    7512246
Python        5904273
Java          3731712
Ruby          2887567
PHP           2474738
Go            2401902
C++           2145920
TypeScript    1994032
C#            1181788
HTML          1180280
Name: count, dtype: int64
```

In [ ]:
```
df.groupby('name')['count'].sum().nlargest(10)
```

Out[ ]:
```
name
JavaScript    7512246
Python        5904273
Java          3731712
Ruby          2887567
PHP           2474738
Go            2401902
C++           2145920
TypeScript    1994032
C#            1181788
HTML          1180280
Name: count, dtype: int64
```

Kasutatud linkid

http://buildandteach.com/pandas-tutorials/ (http://buildandteach.com/pandas-tutorials/)

https://www.kaggle.com/code/residentmario/creating-reading-and-writing/tutorial (https://www.kaggle.com/code/residentmario/creating-reading-and-writing/tutorial)

https://smysl.io/blog/pandas/ (https://smysl.io/blog/pandas/)