Elements of Computing 2

Today by Clemens S.
Plan

- Data Sources
- Import & Create Databases
- Indexing
- (Machine Learning)
Databases

• US gov: https://data.gov

• South Bend: https://southbendin.gov/transparency-and-performance/

• Europe: https://ec.europa.eu/eurostat/data/database

• Germany: https://www.destatis.de/EN/Home/_node.html


• FED: https://www.federalreserve.gov/data.htm


• U.S. Census: https://www.census.gov/econ/

• Stock Prices: https://markets.wsj.com/
**Import & Create Databases**

`pandas.read_excel`

```python
pandas.read_excel(io, sheet_name=0, header=0, names=None, index_col=None, parse_cols=None, usecols=None, squeeze=False, dtype=None, engine=None, converters=None, true_values=None, false_values=None, skiprows=None, nrows=None, na_values=None, keep_default_na=True, verbose=False, parse_dates=False, date_parser=None, thousands=None, comment=None, skip_footer=0, skipfooter=0, convert_float=True, mangle_dupe_cols=True, **kwds)
```

Read an Excel file into a pandas DataFrame.

Support both .xls and .xlsx file extensions from a local filesystem or URL. Support an option to read a single sheet or a list of sheets.

`pandas.read_csv`

```python
pandas.read_csv(filepath_or_buffer, sep=', ', delimiter=None, header='infer', names=None, index_col=None, usecols=None, squeeze=False, prefix=None, mangle_dupe_cols=True, dtype=None, engine=None, converters=None, true_values=None, false_values=None, skipinitialspace=False, skiprows=None, na_values=None, keep_default_na=True, na_filter=True, verbose=False, skip_blank_lines=True, parse_dates=False, infer_datetime_format=False, keep_date_col=False, date_parser=None, dayfirst=False, iterator=False, chunksize=None, compression='infer', thousands=None, decimal='.', lineterminator=None, quotechar=None, quoting=0, doublequote=True, escapechar=None, comment=None, encoding=None, dialect=None, tupleize_cols=None, error_bad_lines=True, warn_bad_lines=True, delim_whitespace=False, low_memory=True, memory_map=False, float_precision=None)
```

Read a comma-separated values (csv) file into DataFrame.

Also supports optionally iterating or breaking of the file into chunks.

Additional help can be found in the online docs for IO Tools.

Source: [https://pandas.pydata.org/pandas-docs/stable/](https://pandas.pydata.org/pandas-docs/stable/)
Import & Create Databases

```python
pandas.DataFrame.to_sql
```

`DataFrame.to_sql(name, con, schema=None, if_exists='fail', index=True, index_label=None, chunksize=None, dtype=None, method=None)`  

Write records stored in a DataFrame to a SQL database.

Databases supported by SQLAlchemy [1] are supported. Tables can be newly created, appended to, or overwritten.
Indexing

Big-O Complexity Chart

Operations

Elements

n = 100:
O(n) = 100s
O(lg n) = 2s

n = 10 000:
O(n) = 10 000s
O(lg n) = 4s

Source: http://bigocheatsheet.com/
Indexing

<table>
<thead>
<tr>
<th>title</th>
<th>rowid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>1</td>
</tr>
<tr>
<td>AndroidDev</td>
<td>3</td>
</tr>
<tr>
<td>Java</td>
<td>6</td>
</tr>
<tr>
<td>Kotlin</td>
<td>5</td>
</tr>
<tr>
<td>ObjectiveC</td>
<td>9</td>
</tr>
<tr>
<td>SQLite</td>
<td>4</td>
</tr>
<tr>
<td>Swift</td>
<td>8</td>
</tr>
<tr>
<td>iOS</td>
<td>2</td>
</tr>
<tr>
<td>iOSDev</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>rowid</th>
<th>title</th>
<th>description</th>
<th>created</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Android</td>
<td>Android apps</td>
<td>2017-06-22 15:11:14</td>
</tr>
<tr>
<td>2</td>
<td>iOS</td>
<td>iOS Apps</td>
<td>2017-06-22 15:11:15</td>
</tr>
<tr>
<td>3</td>
<td>AndroidDev</td>
<td>For android development</td>
<td>2017-06-22 15:11:16</td>
</tr>
<tr>
<td>4</td>
<td>SQLite</td>
<td>The SQLite database</td>
<td>2017-06-22 15:11:17</td>
</tr>
<tr>
<td>5</td>
<td>Kotlin</td>
<td>Kotlin Language</td>
<td>2017-06-22 15:11:18</td>
</tr>
<tr>
<td>6</td>
<td>Java</td>
<td>Java Language</td>
<td>2017-06-22 15:11:19</td>
</tr>
<tr>
<td>7</td>
<td>iOSDev</td>
<td>For iOS development</td>
<td>2017-06-22 15:11:20</td>
</tr>
<tr>
<td>8</td>
<td>Swift</td>
<td>Swift Language</td>
<td>2017-06-22 15:11:21</td>
</tr>
<tr>
<td>9</td>
<td>ObjectiveC</td>
<td>Objective-C development</td>
<td>2017-06-22 15:11:22</td>
</tr>
</tbody>
</table>

One binary search to find the rowid for "Kotlin", another to find the data.

Source: https://medium.com/@JasonWyatt/squeezing-performance-from-sqlite-indexes-indexes-c4e175f3c346
Machine Learning
Why bother?

**MODEL 1**
- Solid bars
- Traditional design
- Weight: 10.3 kilograms
- Displacement: 0.8 micrometers

**MODEL 2**
- Uniform lattice
- Smart design with ALM
- Weight: 4.1 kilograms
- Displacement: 4.2 micrometers

**MODEL 3**
- Evolved lattice
- Evolutionary design with ALM
- Weight: 2.9 kilograms
- Displacement: 6.1 micrometers
Machine Learning

“A computer program is said to learn from experience, E, with respect to some class of tasks, T, and performance measure, P, if its performance at tasks in T, as measured by P, improves with experience, E.”

Tom Mitchell, *Machine Learning*
Machine Learning

Learning

- Supervised
  - Classification
  - Regression

- Unsupervised
  - Clustering
  - Outlier detection

- Reinforcement learning

Source: Prof N. Chawla
Machine Learning

Source: Prof N. Chawla
Machine Learning
Machine Learning

Underfitting

Overfitting

![Graph showing underfitting and overfitting with error vs model complexity]

- Test error
- Training error

Source: Prof N. Chawla
Regression

- **L** Linear relationship
  - The relationship between X and the mean of Y is (approximately) linear.
- **I** Independent observations
  - Features conditionally independent. Errors are independent of each other.
- **N** Normally distributed
  - Probability distribution of the error is approximately normal.
- **E** Equal variance across X’s
  - The variance of error is the same for any value of X.

Source: Prof N. Chawla
Decision Trees

Training Data

<table>
<thead>
<tr>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Cheat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Single</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Married</td>
<td>100K</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>Single</td>
<td>70K</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Married</td>
<td>120K</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Divorced</td>
<td>95K</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>Married</td>
<td>60K</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>Divorced</td>
<td>220K</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>Single</td>
<td>85K</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Married</td>
<td>75K</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>Single</td>
<td>90K</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Model: Decision Tree

Splitting features

Source: Prof N. Chawla
Data
Resources

• **Deep Learning:** [https://www.fast.ai/](https://www.fast.ai/)
• Deep Learning: [https://www.deeplearning.ai/](https://www.deeplearning.ai/)
• Little bit of everything: [https://scikit-learn.org/stable/](https://scikit-learn.org/stable/)
• Deep Learning: [https://keras.io/](https://keras.io/)
• Automated: [https://github.com/EpistasisLab/tpot](https://github.com/EpistasisLab/tpot)
• Challenges: [https://www.kaggle.com/](https://www.kaggle.com/)
• Prof Chawla: Machine Learning (CSE Course at ND)