Memory efficient data handling

1. Pandas

```
import pandas as pd

pd.read_csv()
```

Parameters:

Selection Params:
use_cols: Columns to use (positional or named)
skiprows: skip rows from start
skipfooter: skip rows at the end

Dtype Params:
dtype: change from default float 64 to 32 or 16
true_values: define value to be read as true
false_values: same with false

1. Pandas

```
import pandas as pd

pd.read_csv()
```

Parameters:

Chunk-wise reading:
chunksize: define chunksizes

Alternate parsing:
low_memory: Consumes less memory WHILE parsing
memory_map: loads the file in memory and parses from there

2. Appache parquet/pyArrow

parquet: column based data storage format from Hadoop ecosystem

pyArrow: in-memory layer to load data as parquet and process with pandas

3. pyTables

Allows working with hdf5 datasets on disk without using RAM

Results

- memory efficiency parameters in pandas seem to be useless
- BUT: reading a table as an iterator over text lowers RAM usage
- parquet/pyarrow is not memory but time efficient
- pyTable allocates less memory but is slower than pandas