

Programming for Social Scientists

Working with files

Johan A. Dornschneider-Elkink

JavaScript Object Notation (JSON) files

File formats

```
{  
    "firstPref":{  
        "label":"How did you vote in the 2020 elections?",  
        "labels":{  
            "1":"Other Right",  
            "2":"Fianna Fail",  
            "3":"Fine Gael",  
            "4":"Green Party",  
            "5":"Sinn Fein",  
            "6":"Labour Party",  
            "7":"The Greens",  
            "8":"Conservative Party",  
            "9":"Ulster Unionist Party",  
            "10":"Social Democratic & Labour Party",  
            "11":"Democratic Unionist Party",  
            "12":"PUP",  
            "13":"Other Left",  
            "14":"Other Right",  
            "15":"No Answer"  
        }  
    }  
}
```

Comma Separated Values (CSV) files

| State | War Deaths | Population | Median Family Income | Per Capita Energy Consumption (BTUs) | Federal Aid to State (\$ per capita) | Rate, 2004 |
|----------|------------|------------|----------------------|--------------------------------------|--------------------------------------|------------|
| Alabama | 20 | 4525375 | 45768 | 22.3 | 5.8 | "1,310" |
| Alaska | 1 | 657755 | 66254 | 25.5 | 7.5 | "3,712" |
| Arizona | 12 | 5739879 | 48995 | 28.0 | 5.1 | "1,260" |
| Arkansas | 16 | 2750000 | 30045 | 18 | 8 | "1,412" |

Log (TXT) files

```
2024-04-01 10:54:11.760 [daemon] INFO: Kotlin compiler dae  
2024-04-01 10:54:11.807 [daemon] INFO: daemon JVM args: -D  
a.rmi.server.hostname=127.0.0.1 -Xmx512m -Dkotlin.environm  
ts=java.base/sun.nio.ch=ALL-UNNAMED  
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daem
```

Pickle (PKL) files



Log (TXT) files

```
from datetime import datetime

now = datetime.now().strftime("%d-%m-%Y %H:%M:%S ")

f = open("log.txt", "w")

f.write(now + "First logging line\n")
f.write(now + "Second logging line\n")

f.close()
```

```
2024-04-01 10:54:11.760 [daemon] INFO: Kotlin compiler`da
2024-04-01 10:54:11.807 [daemon] INFO: daemon JVM args: -D
a.rmi.server.hostname=127.0.0.1 -Xmx512m -Dkotlin.environn
ts=java.base/sun.nio.ch=ALL-UNNAMED
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daem
```

Log (TXT) files

```
f = open("log.txt", "r")
```

```
print("Log file contents:")
print(f.readline())
```

```
f.close()
```

```
from datetime import datetime

now = datetime.now().strftime("%d-%m-%Y %H:%M:%S ")

f = open("log.txt", "w")

f.write(now + "First logging line\n")
f.write(now + "Second logging line\n")

f.close()
```

```
2024-04-01 10:54:11.760 [daemon] INFO: Kotlin compiler`da
2024-04-01 10:54:11.807 [daemon] INFO: daemon JVM args: -D
a.rmi.server.hostname=127.0.0.1 -Xmx512m -Dkotlin.environn
ts=java.base/sun.nio.ch=ALL-UNNAMED
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daen
```

Log (TXT) files

```
f = open("log.txt", "r")
```

```
print("Log file contents:")
print(f.readlines())
```

```
f.close()
```

```
from datetime import datetime

now = datetime.now().strftime("%d-%m-%Y %H:%M:%S ")

f = open("log.txt", "w")

f.write(now + "First logging line\n")
f.write(now + "Second logging line\n")

f.close()
```

```
2024-04-01 10:54:11.760 [daemon] INFO: Kotlin compiler`da
2024-04-01 10:54:11.807 [daemon] INFO: daemon JVM args: -D
a.rmi.server.hostname=127.0.0.1 -Xmx512m -Dkotlin.environn
ts=java.base/sun.nio.ch=ALL-UNNAMED
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daen
```

Log (TXT) files

```
f = open("log.txt", "r")
```

```
print("Log file contents:")
print(f.read())
```

```
f.close()
```

```
from datetime import datetime

now = datetime.now().strftime("%d-%m-%Y %H:%M:%S ")

f = open("log.txt", "w")

f.write(now + "First logging line\n")
f.write(now + "Second logging line\n")

f.close()
```

```
2024-04-01 10:54:11.760 [daemon] INFO: Kotlin compiler`da
2024-04-01 10:54:11.807 [daemon] INFO: daemon JVM args: -D
a.rmi.server.hostname=127.0.0.1 -Xmx512m -Dkotlin.environn
ts=java.base/sun.nio.ch=ALL-UNNAMED
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daem
```

Log (TXT) files

```
f = open("log.txt", "r")
```

```
print("Log file contents:")
```

```
for line in f:
```

```
    print(line.rstrip())
```

```
f.close()
```

```
from datetime import datetime  
  
now = datetime.now().strftime("%d-%m-%Y %H:%M:%S ")  
  
f = open("log.txt", "w")  
  
f.write(now + "First logging line\n")  
f.write(now + "Second logging line\n")  
  
f.close()
```

```
2024-04-01 10:54:11.760 [daemon] INFO: Kotlin compiler`da  
2024-04-01 10:54:11.807 [daemon] INFO: daemon JVM args: -D  
a.rmi.server.hostname=127.0.0.1 -Xmx512m -Dkotlin.environm  
ts=java.base/sun.nio.ch=ALL-UNNAMED  
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daem
```

Log (TXT) files

```
with open("log.txt", "r") as log:  
    for line in log.readlines():  
        print(line.rstrip())
```

```
from datetime import datetime  
  
now = datetime.now().strftime("%d-%m-%Y %H:%M:%S ")  
  
f = open("log.txt", "w")  
  
f.write(now + "First logging line\n")  
f.write(now + "Second logging line\n")  
  
f.close()
```

```
2024-04-01 10:54:11.760 [daemon] INFO: Kotlin compiler`da  
2024-04-01 10:54:11.807 [daemon] INFO: daemon JVM args: -D  
a.rmi.server.hostname=127.0.0.1 -Xmx512m -Dkotlin.environm  
ts=java.base/sun.nio.ch=ALL-UNNAMED  
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daem
```

Log (TXT) files

```
now = datetime.now( ).strftime( "%d-%m-%Y %H:%M:%S " )

f = open( "log.txt", "a" )

f.write( now + "Third logging line\n" )

f.close()
```

```
2024-04-01 10:54:11.760 [daemon] INFO: Kotlin compiler`da
2024-04-01 10:54:11.807 [daemon] INFO: daemon JVM args: -D
a.rmi.server.hostname=127.0.0.1 -Xmx512m -Dkotlin.environn
ts=java.base/sun.nio.ch=ALL-UNNAMED
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daen
```

```
from datetime import datetime

now = datetime.now( ).strftime( "%d-%m-%Y %H:%M:%S " )

f = open( "log.txt", "w" )

f.write( now + "First logging line\n" )
f.write( now + "Second logging line\n" )

f.close()
```


JavaScript Object Notation (JSON) files

```
options = {"iterations": 1000,  
"strategies": ["sticker", "predator",  
"hunter"]}
```

```
{  
  "firstPref":{  
    "label":"How did you vote in the 2020 elections?",  
    "labels":{  
      "1":"Other Right",  
      "2":"Fianna Fail",  
      "3":"Fine Gael",  
      "4":"Green Party",  
      "5":"Sinn Féin",  
      "6":"Labour",  
      "7":"UCD",  
      "8":"Solidarity",  
      "9":"Other Left",  
      "10":"None",  
      "11":"Other",  
      "12":"Don't know"}  
  }  
}
```

JavaScript Object Notation (JSON) files

```
options = {"iterations": 1000,  
"strategies": ["sticker", "predator",  
"hunter"]}
```

```
import json  
  
with open("config.json", "w") as config:  
    json.dump(options, config, indent=2)
```

```
{  
    "firstPref":{  
        "label":"How did you vote in the 2020 elections?",  
        "labels":{  
            "1":"Other Right",  
            "2":"Fianna Fail",  
            "3":"Fine Gael",  
            "4":"Green Party",  
            "5":"Sinn Fein",  
            "6":"Labour",  
            "7":"UCD",  
            "8":"Solidarity",  
            "9":"Pillar",  
            "10":"Other Left",  
            "11":"None",  
            "12":"Don't know"}  
    }  
}
```

JavaScript Object Notation (JSON) files

```
options = {"iterations": 1000,  
"strategies": ["sticker", "predator",  
"hunter"]}
```

```
import json  
  
with open("config.json", "r") as config:  
    cfg = json.load(config)  
    print(cfg)
```

```
{  
  "firstPref":{  
    "label":"How did you vote in the 2020 elections?",  
    "labels":{  
      "1":"Other Right",  
      "2":"Fianna Fail",  
      "3":"Fine Gael",  
      "4":"Green Party",  
      "5":"Labour Party",  
      "6":"Sinn Féin",  
      "7":"Solidarity",  
      "8":"People Before Profit",  
      "9":"Workers Party",  
      "10":"Other Left",  
      "11":"None",  
      "12":"Don't know",  
      "13":"Other",  
      "14":"Other (please specify)",  
      "15":"Other (please specify)",  
      "16":"Other (please specify)",  
      "17":"Other (please specify)",  
      "18":"Other (please specify)"  
    }  
  }  
}  
[{"label": "How did you vote in the 2020 elections?", "value": 1}, {"label": "None", "value": 2}, {"label": "Don't know", "value": 3}, {"label": "Other", "value": 4}, {"label": "Other (please specify)", "value": 5}, {"label": "Other", "value": 6}, {"label": "Other", "value": 7}, {"label": "Other", "value": 8}, {"label": "Other", "value": 9}, {"label": "Other", "value": 10}, {"label": "Other", "value": 11}, {"label": "Other", "value": 12}, {"label": "Other", "value": 13}, {"label": "Other", "value": 14}, {"label": "Other", "value": 15}, {"label": "Other", "value": 16}, {"label": "Other", "value": 17}, {"label": "Other", "value": 18}], [{"label": "Other Right", "value": 1}, {"label": "Fianna Fail", "value": 2}, {"label": "Fine Gael", "value": 3}, {"label": "Green Party", "value": 4}, {"label": "Labour Party", "value": 5}, {"label": "Sinn Féin", "value": 6}, {"label": "Solidarity", "value": 7}, {"label": "People Before Profit", "value": 8}, {"label": "Workers Party", "value": 9}, {"label": "Other Left", "value": 10}, {"label": "Other", "value": 11}, {"label": "Other", "value": 12}, {"label": "Other", "value": 13}, {"label": "Other (please specify)", "value": 14}, {"label": "Other (please specify)", "value": 15}, {"label": "Other (please specify)", "value": 16}, {"label": "Other (please specify)", "value": 17}, {"label": "Other (please specify)", "value": 18}], [{"label": "Other Right", "value": 1}, {"label": "Fianna Fail", "value": 2}, {"label": "Fine Gael", "value": 3}, {"label": "Green Party", "value": 4}, {"label": "Labour Party", "value": 5}, {"label": "Sinn Féin", "value": 6}, {"label": "Solidarity", "value": 7}, {"label": "People Before Profit", "value": 8}, {"label": "Workers Party", "value": 9}, {"label": "Other Left", "value": 10}, {"label": "Other", "value": 11}, {"label": "Other", "value": 12}, {"label": "Other", "value": 13}, {"label": "Other (please specify)", "value": 14}, {"label": "Other (please specify)", "value": 15}, {"label": "Other (please specify)", "value": 16}, {"label": "Other (please specify)", "value": 17}, {"label": "Other (please specify)", "value": 18}]]
```

JavaScript Object Notation (JSON) files

File formats

```
{  
    "firstPref":{  
        "label":"How did you vote in the 2020 elections?",  
        "labels":{  
            "1":"Other Right",  
            "2":"Fianna Fail",  
            "3":"Fine Gael",  
            "4":"Green Party",  
        }  
    }  
}
```

Comma Separated Values (CSV) files

| State | War Deaths | Population | Median Family Income | Per Capita Energy Consumption (BTUs) | Vote for GWB | Rate, 2004 | Federal Aid to State (\$ per capita) | Per capita Energy Consumption (BTUs) |
|----------|------------|------------|----------------------|--------------------------------------|--------------|------------|--------------------------------------|--------------------------------------|
| Alabama | 20 | 4525375 | 45768 | 22.3 | 5.8 | "1,310" | 71.4 | 78.0 |
| Alaska | 1 | 657755 | 66254 | 25.5 | 7.5 | "3,712" | 70.7 | 67.2 |
| Arizona | 12 | 5739879 | 48995 | 28.0 | 5.1 | "1,260" | 87.6 | 68.7 |
| Arkansas | 16 | 2750000 | 30045 | 18 | 8.5 | "1,412" | 81.3 | 69 |

Pickle (PKL) files



Log (TXT) files

```
2024-04-01 10:54:11.760 [daemon] INFO: Kotlin compiler dae  
2024-04-01 10:54:11.807 [daemon] INFO: daemon JVM args: -D  
a.rmi.server.hostname=127.0.0.1 -Xmx512m -Dkotlin.environment=java.base/sun.nio.ch=ALL-UNNAMED  
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daem
```

Binary files

| | | |
|----------|---|---------------------|
| 000002d0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 19 26 c0 d3 |&.. |
| 000002e0 | 3e 88 4a c4 58 90 72 09 d6 86 8d bb d4 72 52 54 | >.J.X.r.....rRT |
| 000002f0 | ee ef f3 10 0d a5 44 ff ce 08 ba b1 ac 84 ce bf |D..... |
| 00000300 | 7c e8 e8 0c 11 a2 d4 d1 cc 9d 89 65 43 f0 72 cf |eC.r. |
| 00000310 | 91 f7 53 45 b2 51 b6 d7 7c 13 d9 2a 3e ed 2c 2e | ..SE.Q.. ...*>.,. |
| 00000320 | 73 e8 5c d0 9e 77 65 e5 22 91 ee a2 51 f7 e7 2f | s.\..we."...Q.. |
| 00000330 | 2a 4a 68 db c9 ea c9 74 6d aa 04 93 77 33 48 89 | *Jh....tm...w3H. |
| 00000340 | bf 56 17 fd 11 77 b5 1c d9 67 f6 9d 09 3f a0 5b | .V....w....g...?..[|
| 00000350 | a3 9f 2c 89 13 91 b0 8e bc 45 19 ae 9a 46 b2 0e | ..,.....E....F.. |

Pickle (PKL) files

```
import pickle  
  
with open("config.pickle", "wb") as config:  
    pickle.dump(options, config)
```



Pickle (PKL) files

```
import pickle

with open("config.pickle", "rb") as config:
    cfg = pickle.load(config)
    print(config)
```



Comma Separate Values (CSV) files

```
data = [
    {"name": "Ruff Diamond", "speed": 5.0},
    {"name": "Silverwind", "speed": 6.0},
    {"name": "Mayfly", "speed": 5.5},
    {"name": "Fireball", "speed": 4.7}
]
```

State,War Deaths,Population,Median Family Income,Per
Rate, 2004",Federal Aid to State (\$ per capita),Pe
apita Energy Consumption (BTUs),Vote for GWB
Alabama,20,4525375,45768,22.3,5.8,"1,310",71.4,78.0
Alaska,1,657755,66254,25.5,7.5,"3,712",70.7,67.2,"1
Arizona,12,5739879,48995,28.0,5.1,"1,260",87.6,68.7
Arkansas,16,2750000,30045,18.8,5.0,"1,412",81.3,69

Comma Separate Values (CSV) files

```
import csv
```

```
with open("boats.csv", "w") as file:
```

```
    data_writer = csv.writer(file)
```

```
    data_writer.writerow(["name", "speed"])
```

```
    for row in data:
```

```
        data_writer.writerow(row.values())
```

```
data = [
    {"name": "Ruff Diamond", "speed": 5.0},
    {"name": "Silverwind", "speed": 6.0},
    {"name": "Mayfly", "speed": 5.5},
    {"name": "Fireball", "speed": 4.7}
]
```

State,War Deaths,Population,Median Family Income,Per Capita Rate, 2004",Federal Aid to State (\$ per capita),Per Capita Energy Consumption (BTUs),Vote for GWB
Alabama,20,4525375,45768,22.3,5.8,"1,310",71.4,78.0
Alaska,1,657755,66254,25.5,7.5,"3,712",70.7,67.2,"1
Arizona,12,5739879,48995,28.0,5.1,"1,260",87.6,68.7
Arkansas,16,2750000,30045,18.8,5.0,"1,412",81.3,60

Comma Separate Values (CSV) files

```
import csv
```

```
with open("boats.csv", "w") as file:  
  
    data_writer = csv.writer(file)  
  
    data_writer.writerow(["name", "speed"])  
  
    for row in data:  
        data_writer.writerow(row.values())
```

```
data = [  
    {"name": "Ruff Diamond", "speed": 5.0},  
    {"name": "Silverwind", "speed": 6.0},  
    {"name": "Mayfly", "speed": 5.5},  
    {"name": "Fireball", "speed": 4.7}  
]
```

boats.csv:

```
name,speed  
Ruff Diamond,5.0  
Silverwind,6.0  
Mayfly,5.5  
Fireball,4.7
```

State,War Deaths,Population,Median Family Income,Per Capita Rate, 2004",Federal Aid to State (\$ per capita),Per Capita Energy Consumption (BTUs),Vote for GWB
Alabama,20,4525375,45768,22.3,5.8,"1,310",71.4,78.0
Alaska,1,657755,66254,25.5,7.5,"3,712",70.7,67.2,"1
Arizona,12,5739879,48995,28.0,5.1,"1,260",87.6,68.7
Arkansas,16,2750000,30045,18.8,5.0,"1,412",81.3,60

Comma Separate Values (CSV) files

```
import csv  
  
boats = []  
  
with open("boats.csv", "r") as file:  
  
    data_reader = csv.DictReader(file)  
  
    for row in data_reader:  
        boats.append(row)  
  
print(boats)
```

```
data = [  
    {"name": "Ruff Diamond", "speed": 5.0},  
    {"name": "Silverwind", "speed": 6.0},  
    {"name": "Mayfly", "speed": 5.5},  
    {"name": "Fireball", "speed": 4.7}  
]
```

boats.csv:

```
name,speed  
Ruff Diamond,5.0  
Silverwind,6.0  
Mayfly,5.5  
Fireball,4.7
```

State,War Deaths,Population,Median Family Income,Per Capita Rate, 2004",Federal Aid to State (\$ per capita),Per Capita Energy Consumption (BTUs),Vote for GWB
Alabama,20,4525375,45768,22.3,5.8,"1,310",71.4,78.0
Alaska,1,657755,66254,25.5,7.5,"3,712",70.7,67.2,"1
Arizona,12,5739879,48995,28.0,5.1,"1,260",87.6,68.7
Arkansas,16,2750000,30045,18.8,5.0,"1,412",81.3,60