



Programming for Social Scientists

Johan A. Dornschneider-Elkink

Working with files

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,Pe  
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,16,2750000,39945,18.8,5.9,"1,412",81.3,69
```

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.envirom  
ts=java.base/sun.nio.ch=ALL-UNNAMED  
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daem
```

Pickle (PKL) files



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,Pe  
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,16,2750000,39945,18.8,5.9,"1,412",81.3,69
```

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

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()
```

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

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,Pe  
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,16,2750000,39945,18.8,5.9,"1,412",81.3,69
```

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.envirom  
ts=java.base/sun.nio.ch=ALL-UNNAMED  
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daem
```

Pickle (PKL) files



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",
```

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",
```

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",
```

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,Pe  
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,16,2750000,39945,18.8,5.9,"1,412",81.3,69
```

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.envirom  
ts=java.base/sun.nio.ch=ALL-UNNAMED  
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daem
```

Pickle (PKL) files



Binary files

```
000002d0  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(cfg)
```



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,Pe  
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,16,2750000,39945,18.8,5.9,"1,412",81.3,69
```

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.envirom  
ts=java.base/sun.nio.ch=ALL-UNNAMED  
2024-04-01 10:54:11.823 [daemon] INFO: daemon args: --daem
```

Pickle (PKL) files



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,Pe  
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,16,2750000,39945,18.8,5.9,"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,Pe
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,16,2750000,39945,18.8,5.9,"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}
]
```

boats.csv:

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

```
State,War Deaths,Population,Median Family Income,Pe
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,16,2750000,39945,18.8,5.9,"1,412",81.3,69.
```

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,Pe
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,16,2750000,39945,18.8,5.9,"1,412",81.3,69.
```