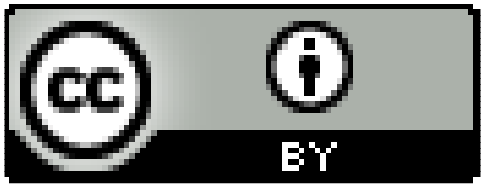
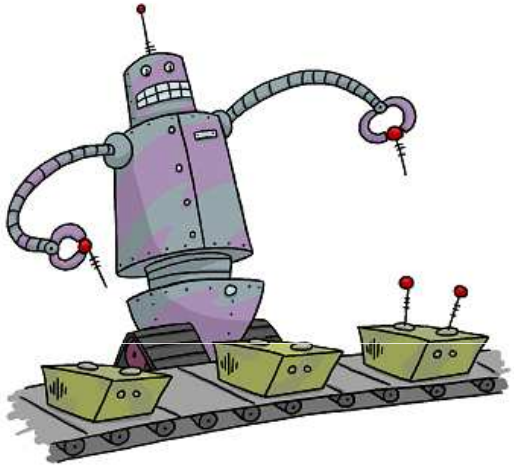
Automated Builds

Patterns

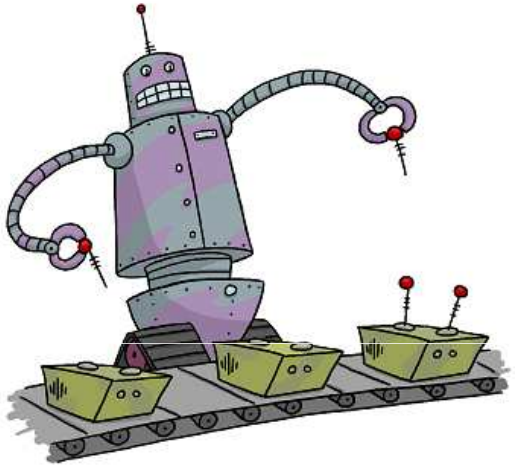


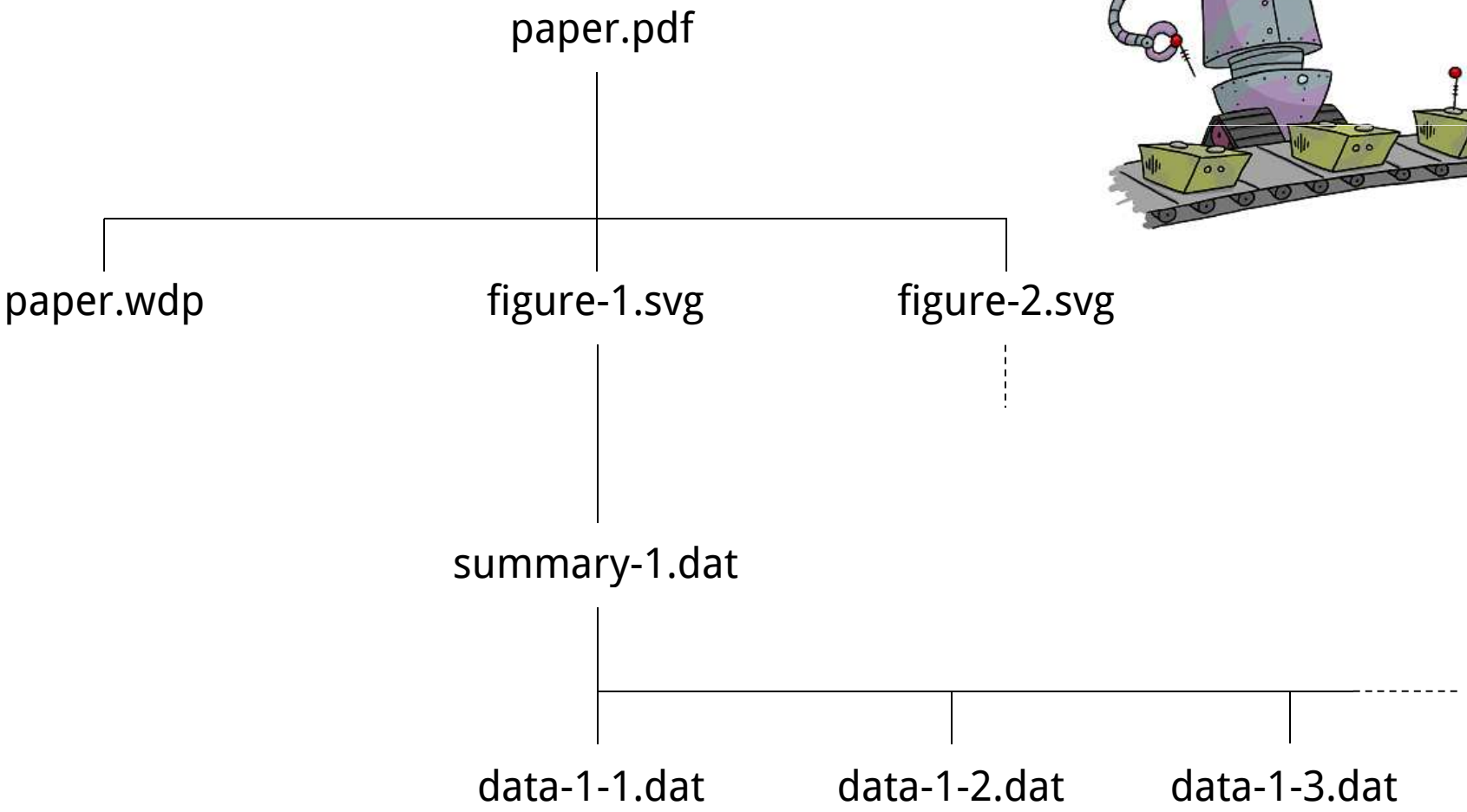
Copyright © Software Carpentry 2010
This work is licensed under the Creative Commons Attribution License
See http://software-carpentry.org/license.html for more information.

Manage tasks and dependencies

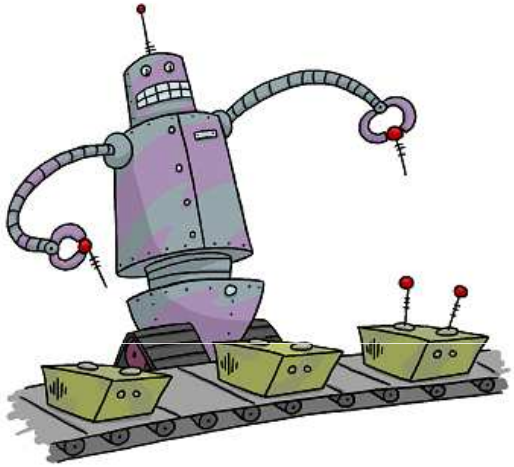


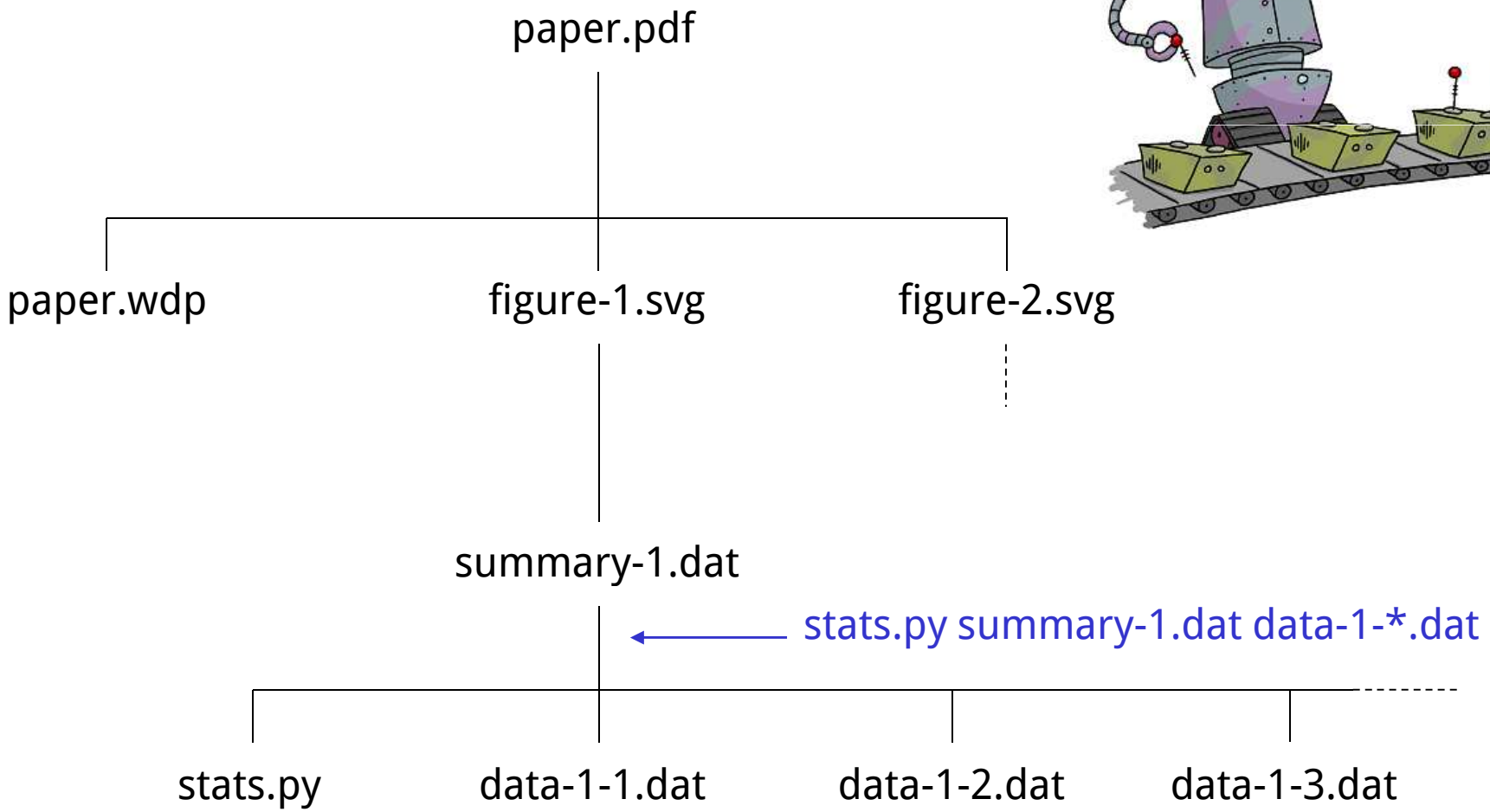
Manage tasks and dependencies



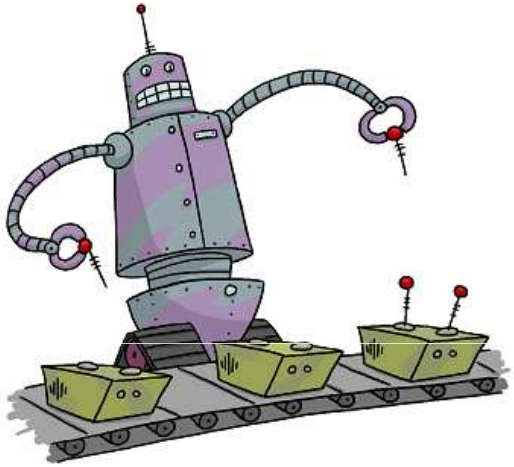


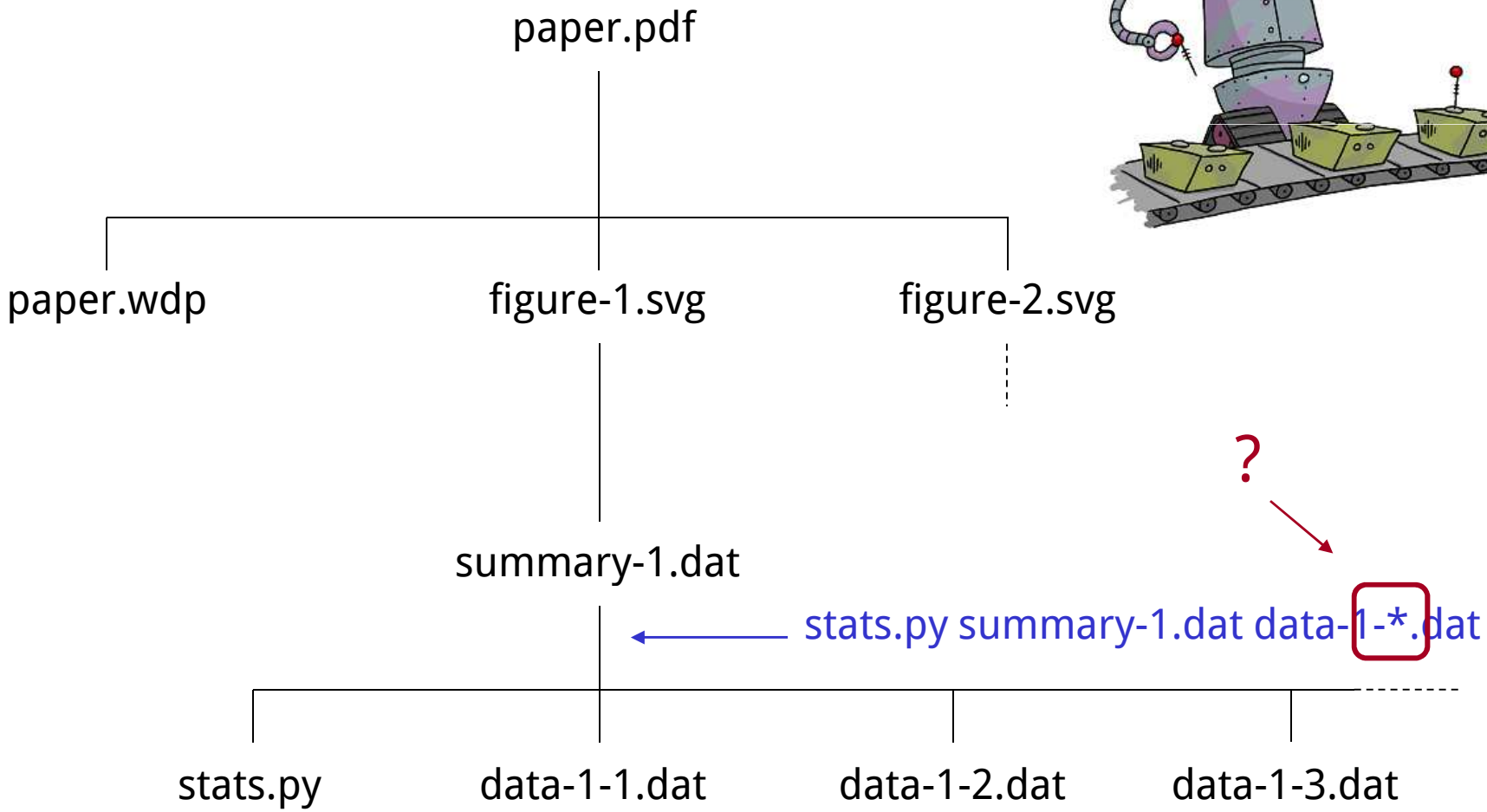
Manage tasks and dependencies



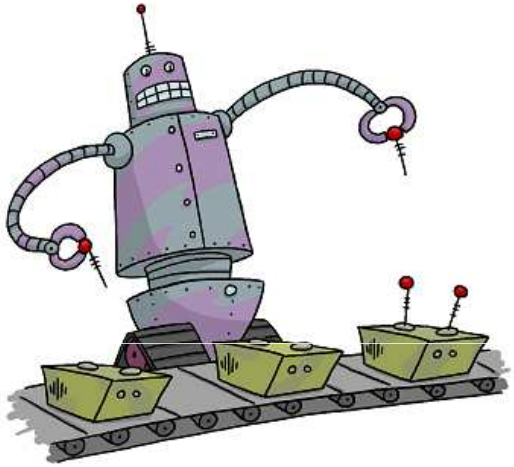


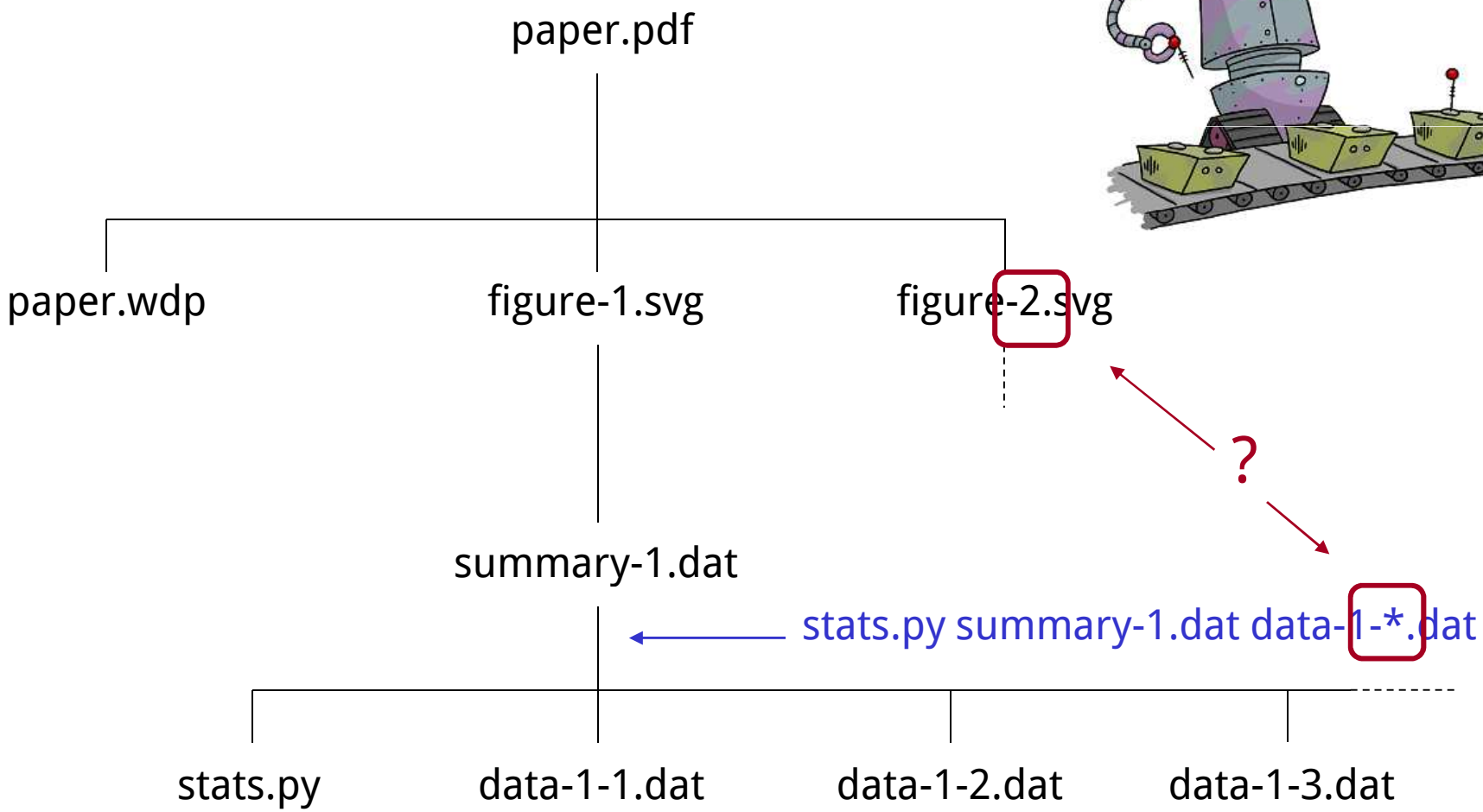
Manage tasks and dependencies

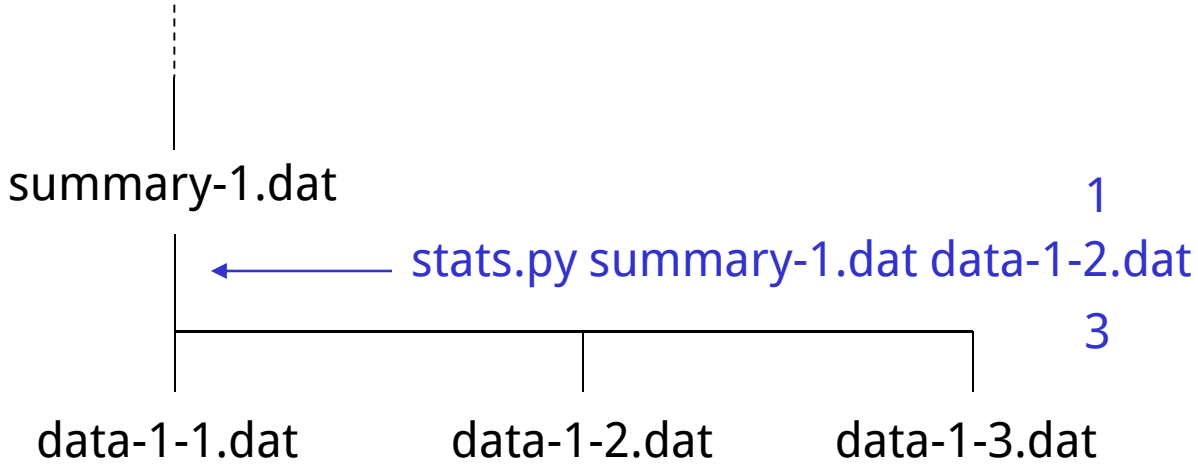




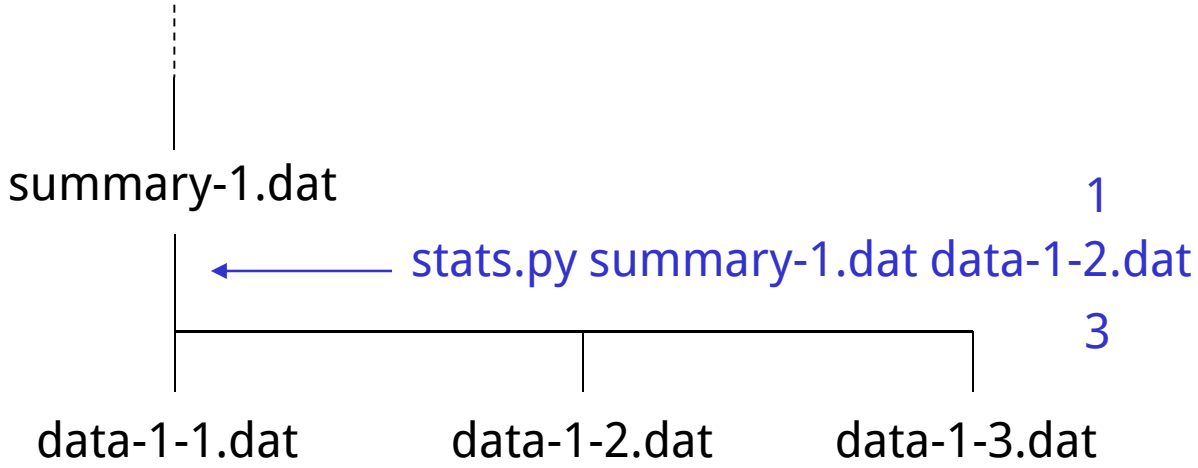
Manage tasks and dependencies



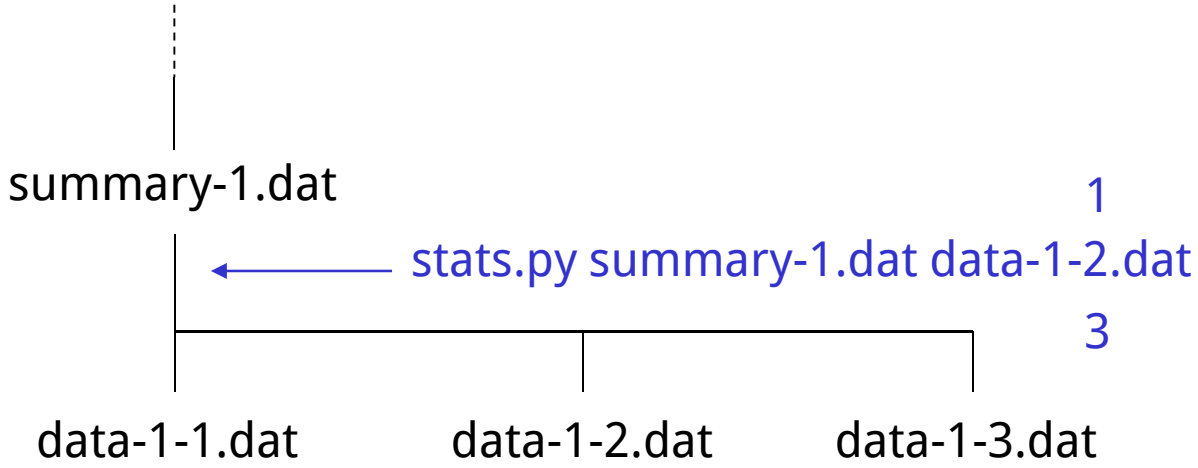




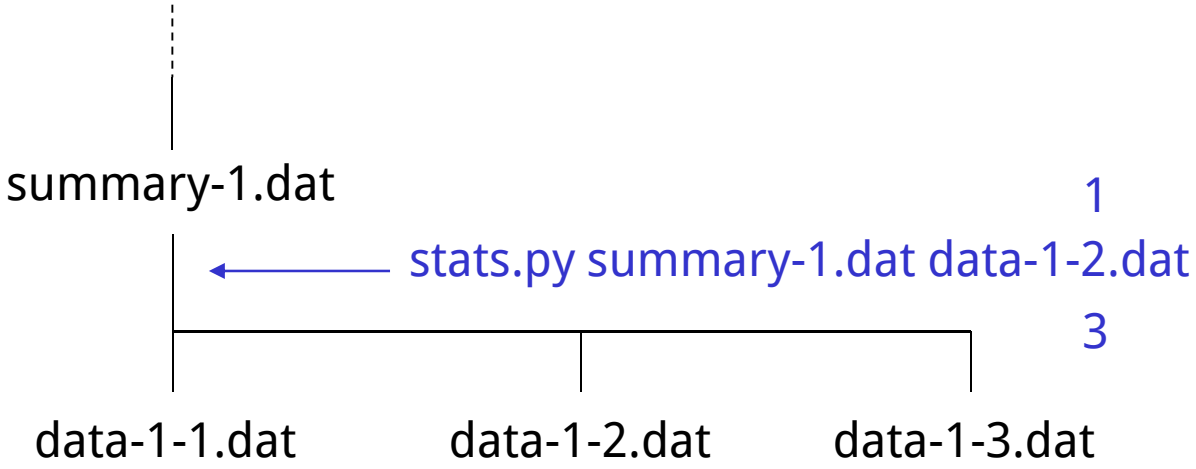

```
# multiple.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py summary-1.dat data-1-1.dat data-1-2.dat data-1-3.dat
```



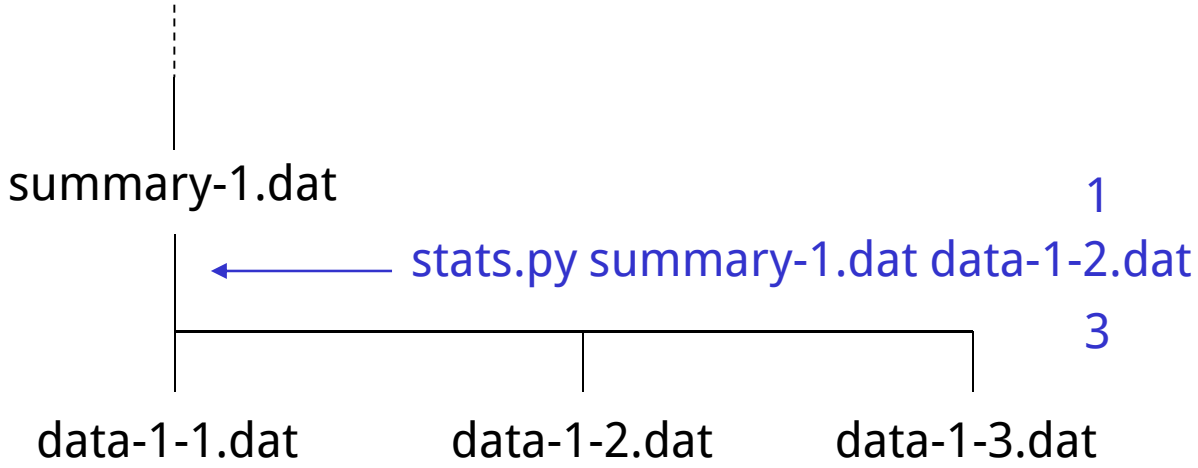

```
# target-variable.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py summary-1.dat data-1-1.dat data-1-2.dat data-1-3.dat
```




```
# target-variable.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py summary-1.dat data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ data-1-1.dat data-1-2.dat data-1-3.dat
```

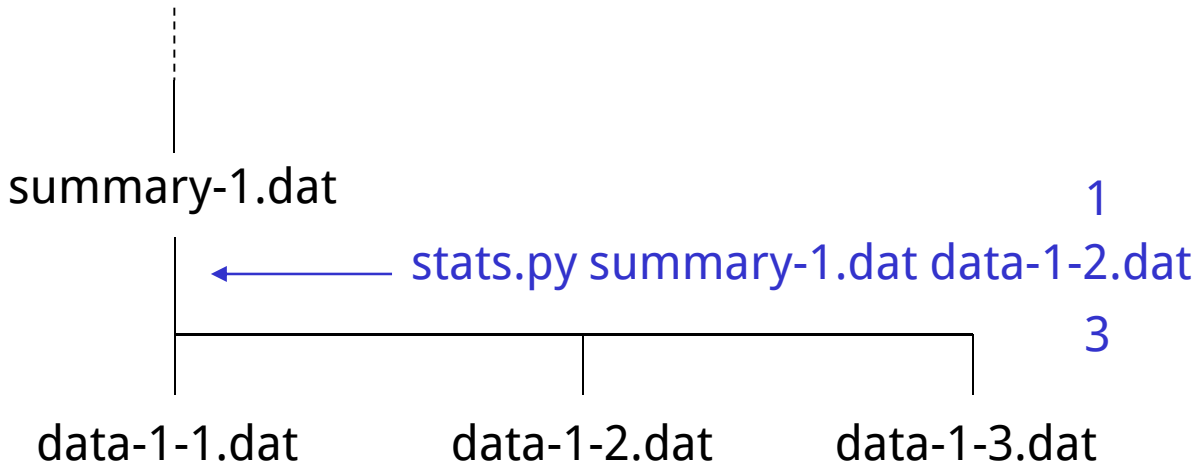



```
# target-variable.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py summary-1.dat data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ data-1-1.dat data-1-2.dat data-1-3.dat
```




```
# target-variable.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py summary-1.dat data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ data-1-1.dat data-1-2.dat data-1-3.dat
```

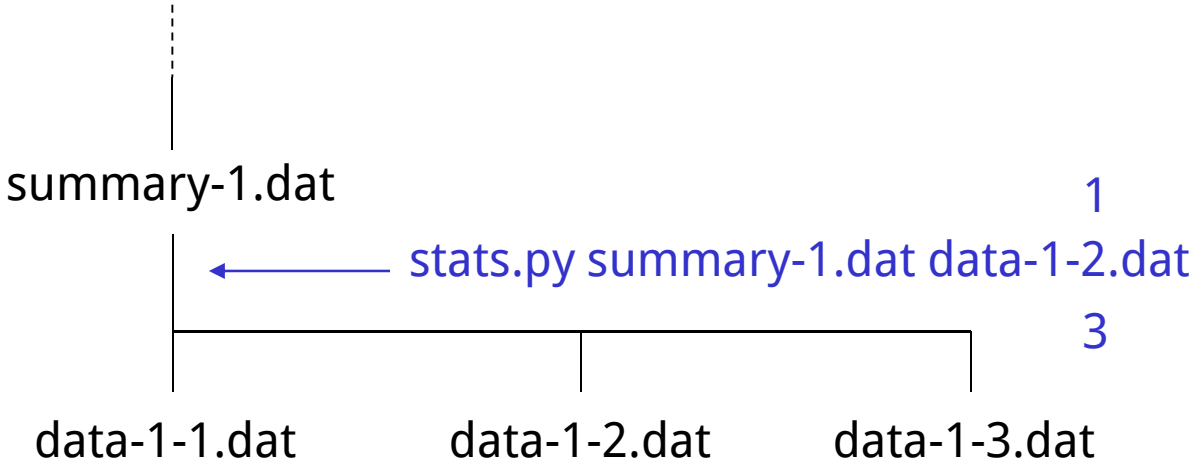
Automatic variable




```
# target-variable.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py summary-1.dat data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ data-1-1.dat data-1-2.dat data-1-3.dat
```

Automatic variable

"the target of this rule"

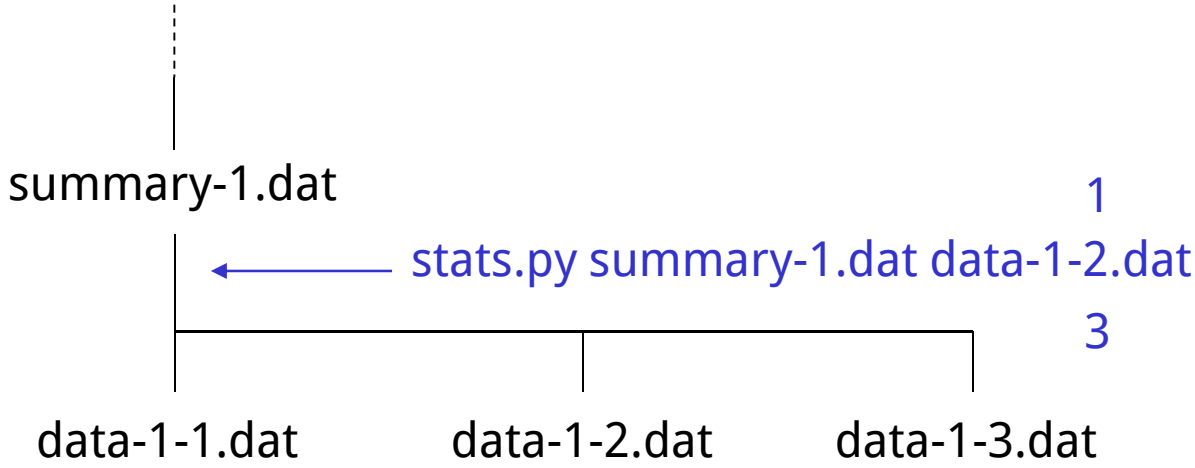



```
# target-variable.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py summary-1.dat data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ data-1-1.dat data-1-2.dat data-1-3.dat
```

Automatic variable

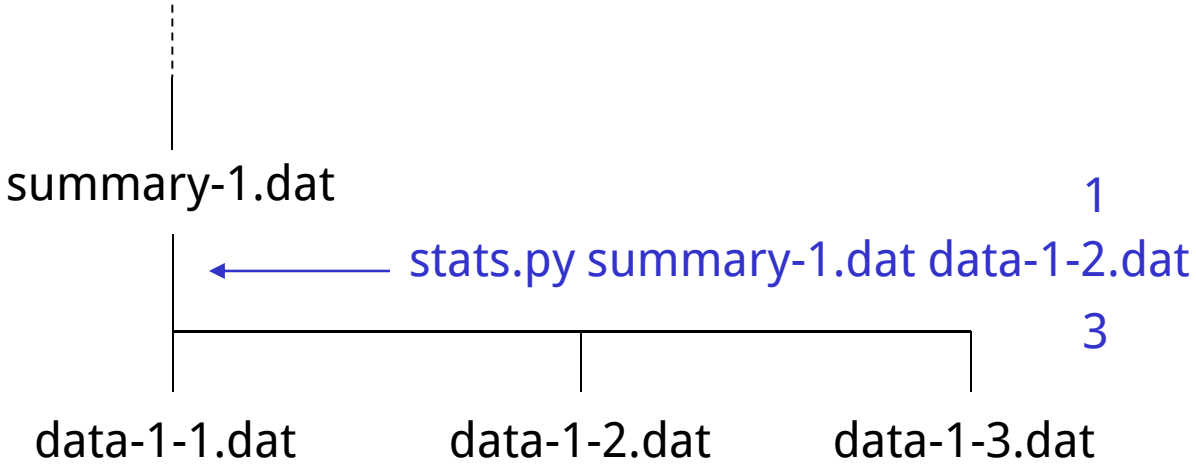
"the target of this rule"

No, there isn't a more readable form

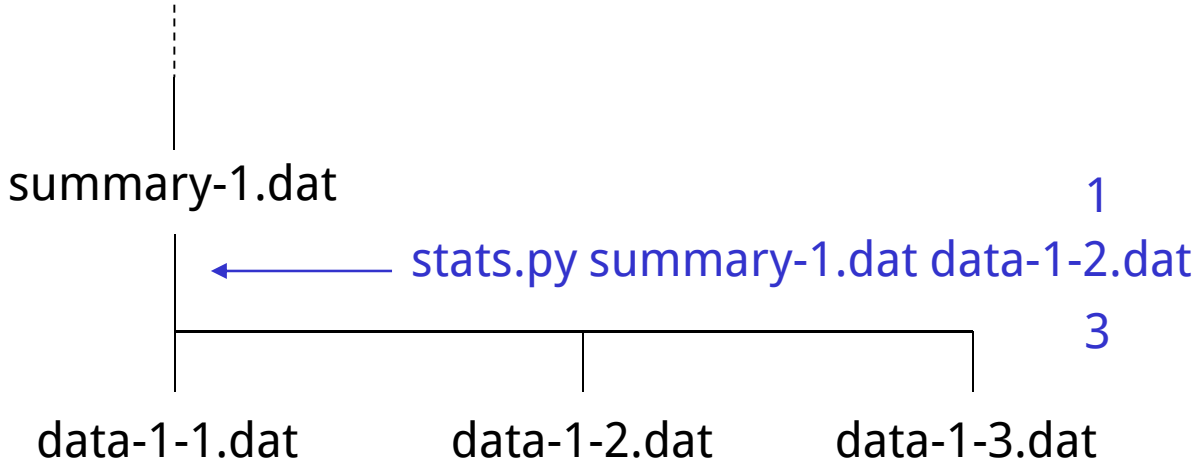



```
# target-variable.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ data-1-1.dat data-1-2.dat data-1-3.dat
```

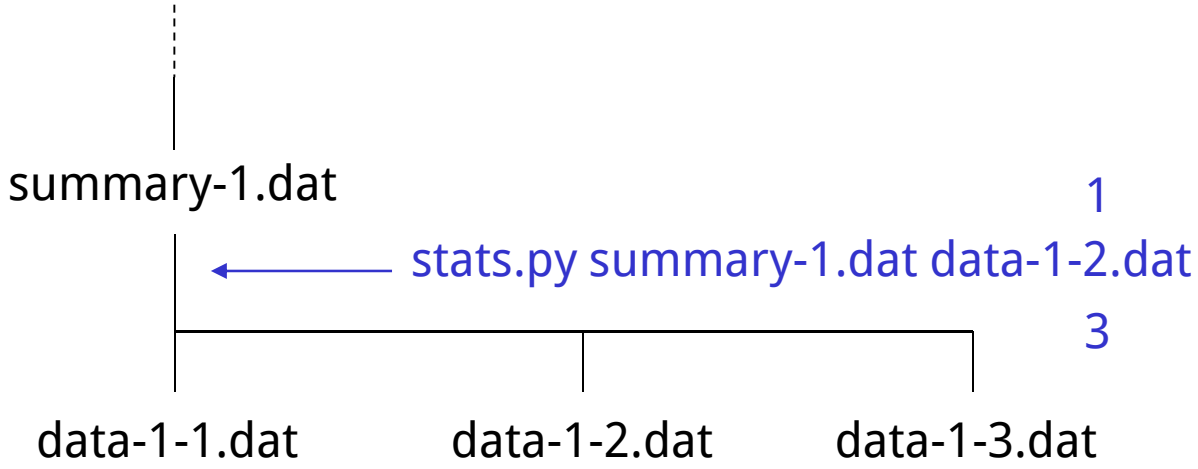
Still a lot of redundancy




```
# variables.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
	stats.py $@ data-1-1.dat data-1-2.dat data-1-3.dat
```

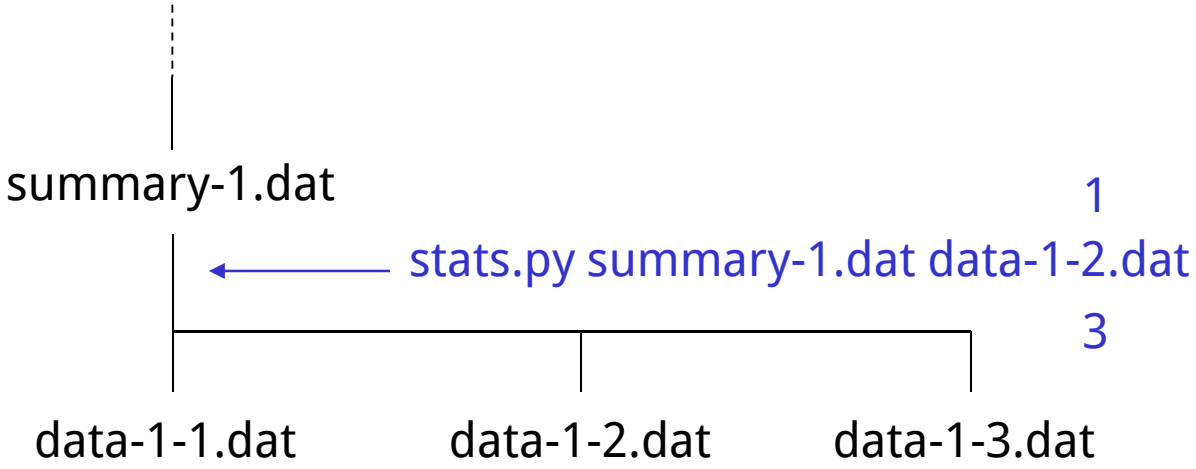



```
# variables.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ $^
```




```
# variables.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ $^
```

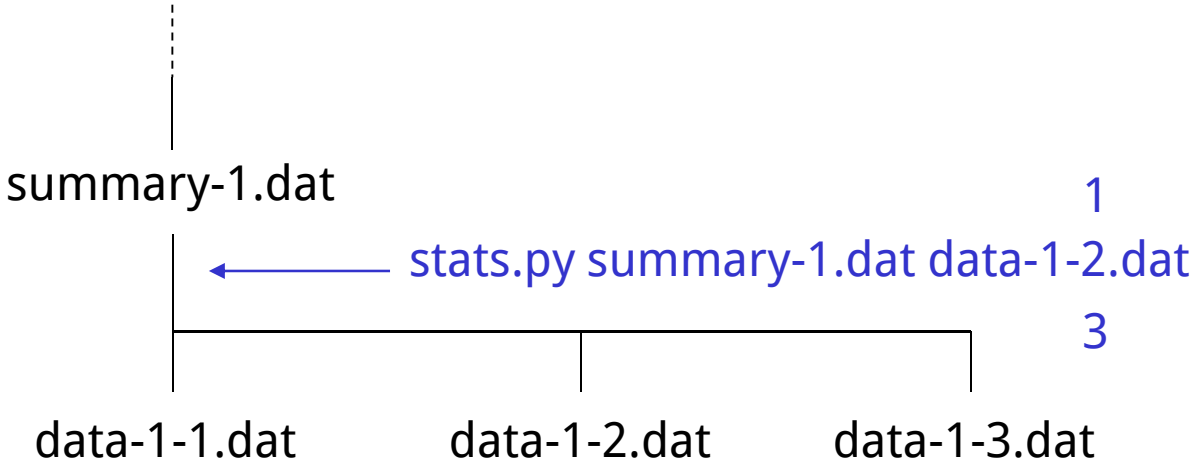
All prerequisites of this rule




```
# variables.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ $^
```

All prerequisites of this rule

$< is "the first prerequisite"

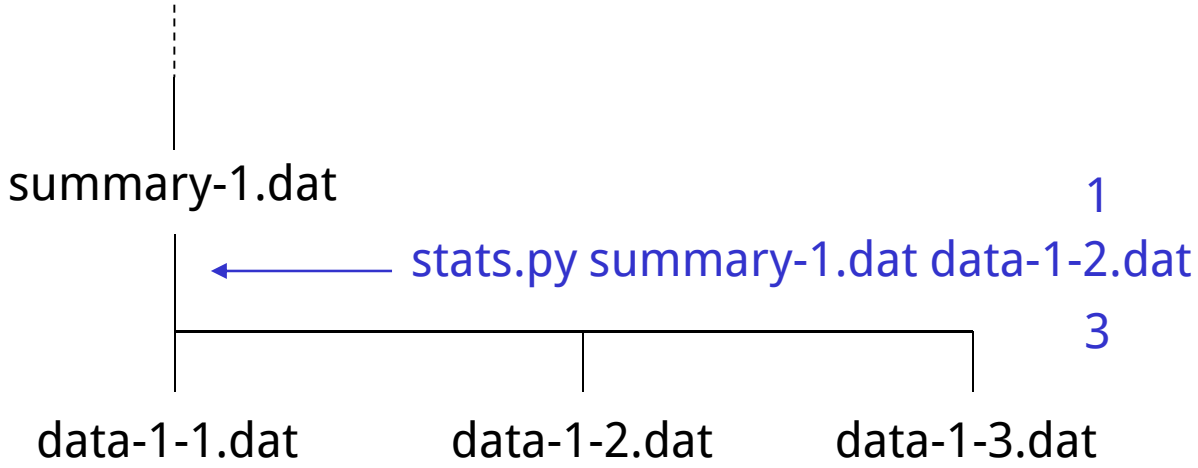



```
# variables.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ $^
```

All prerequisites of this rule

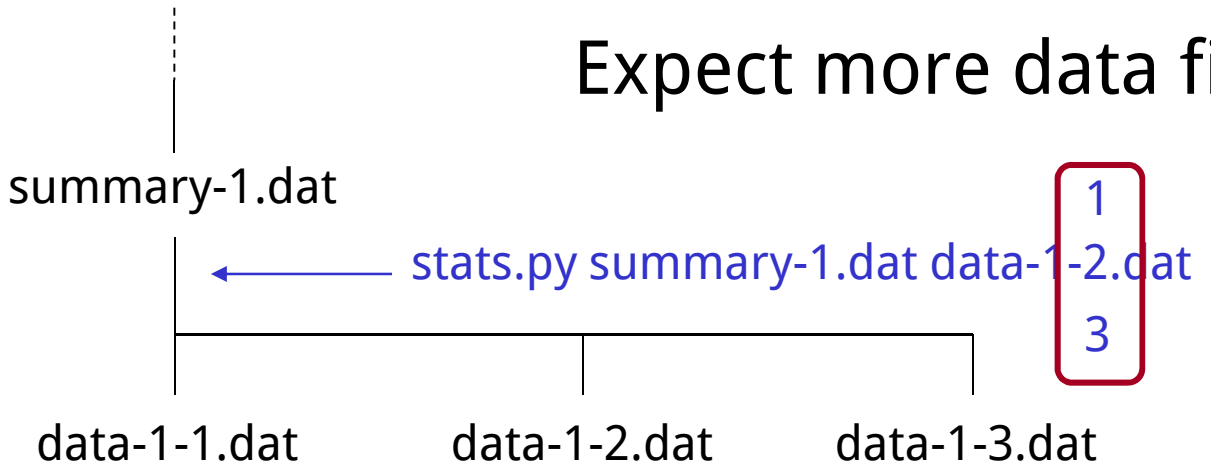
$< is "the first prerequisite"

$? is "all out-of-date prerequisites"

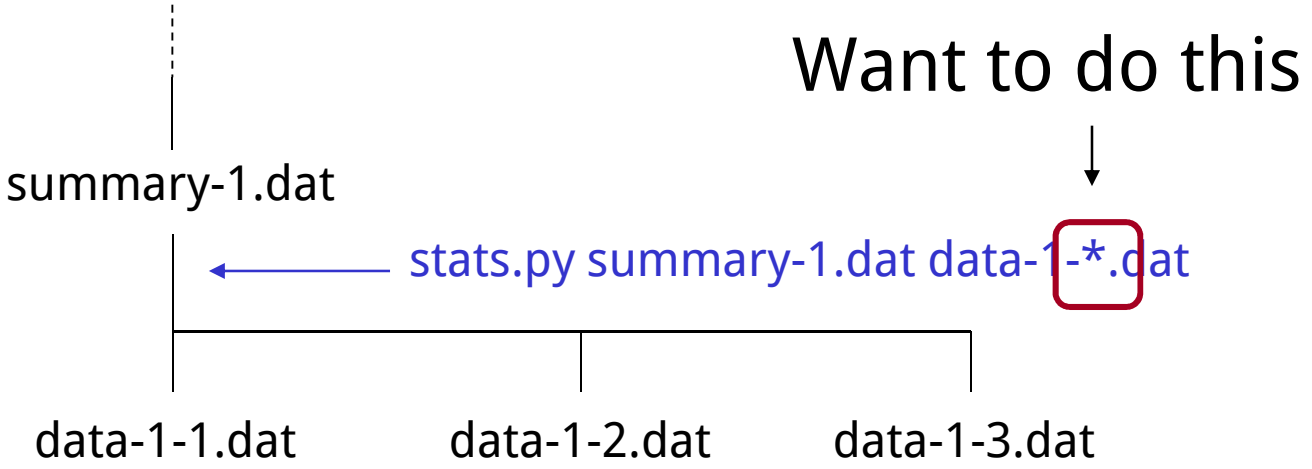



```
# variables.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ $^
```

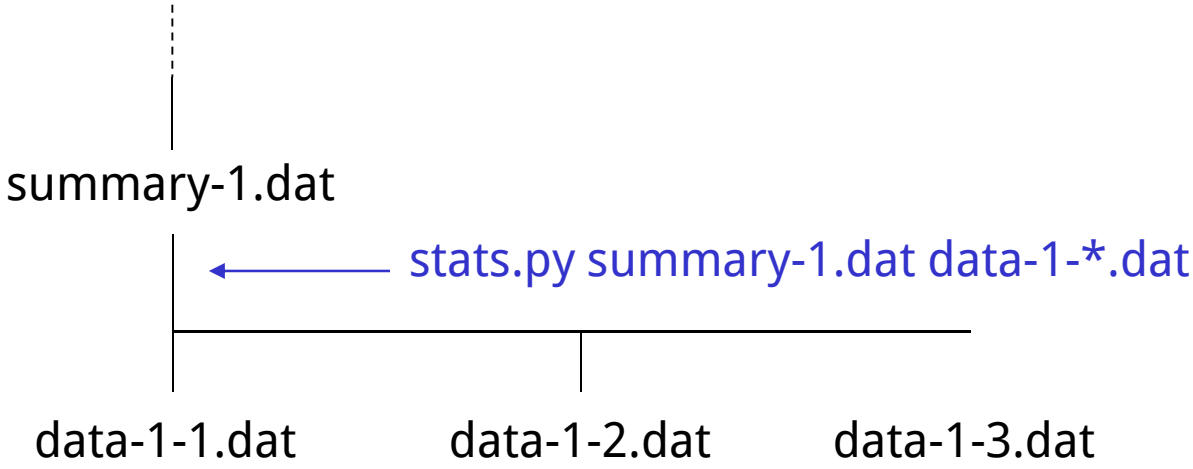
Expect more data files



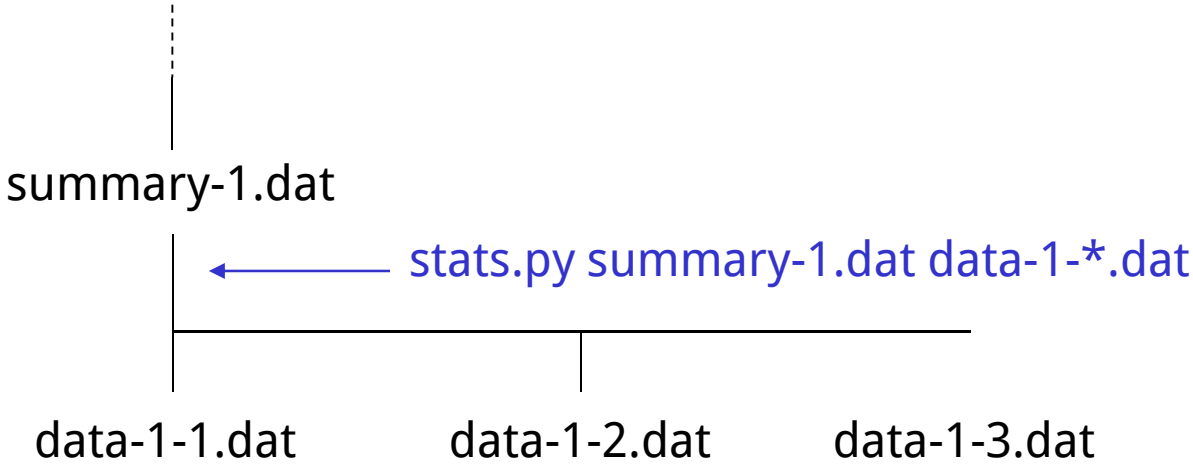

```
# variables.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ $^
```




```
# variables.mk

summary-1.dat : data-1-1.dat data-1-2.dat data-1-3.dat
        stats.py $@ $^
```

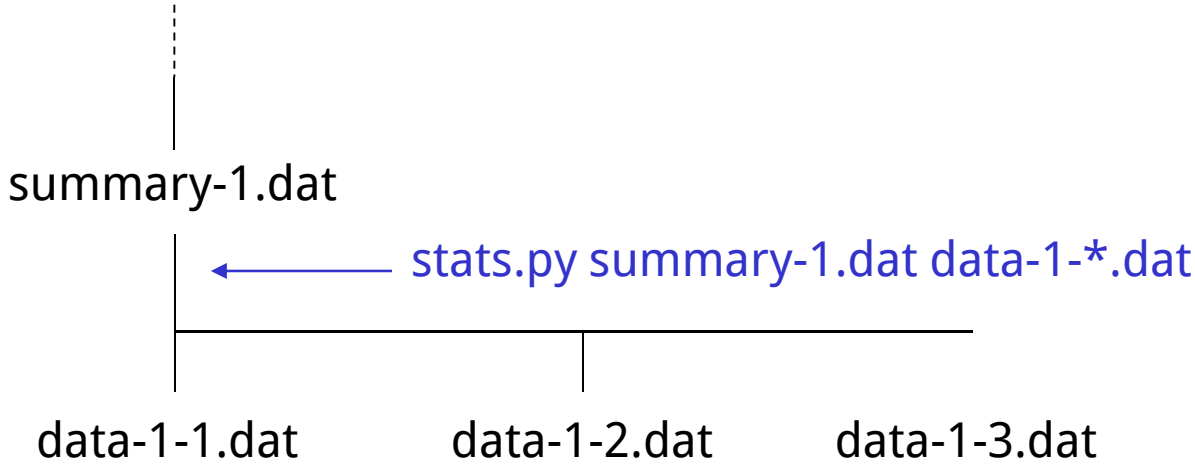



```
# wildcard.mk

summary-1.dat : data-1-*.dat
        stats.py $@ $^
```




```
# wildcard.mk

summary-1.dat : data-1-*.dat
        stats.py $@ $^
```

Just like shell wildcard



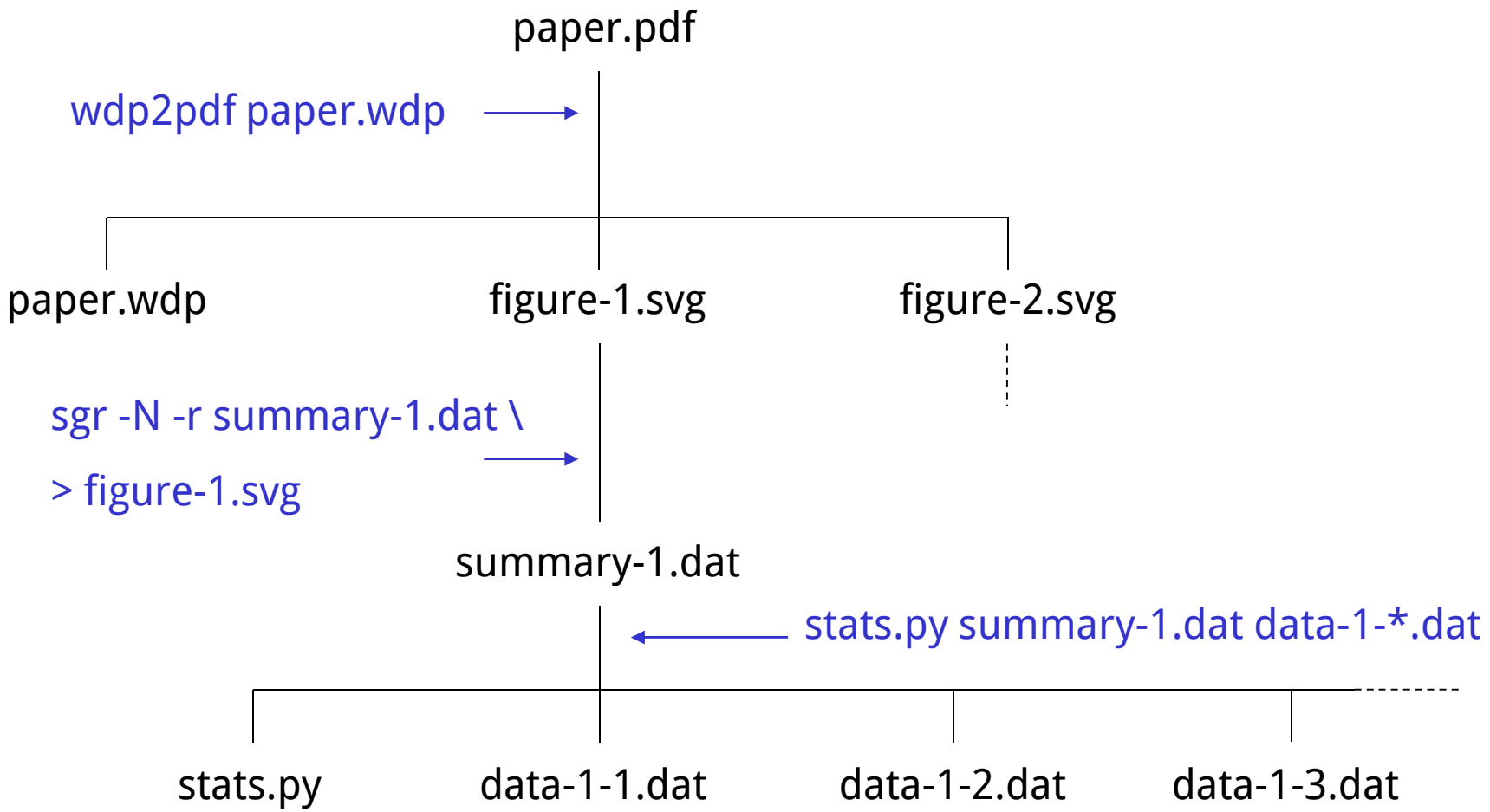

```
# wildcard.mk

summary-1.dat : data-1-*.dat
        stats.py $@ $^
```

Just like shell wildcard

Must use $^ in action, since

filenames not fixed in advance

The whole tree one more time



The makefile so far

```
paper.pdf : paper.wdp figure-1.svg figure-2.svg
        wdp2pdf $<

figure-1.svg : summary-1.dat
        sgr -N -r $@ $^

figure-2.svg : summary-2.dat
        sgr -N -r $@ $^

summary-1.dat : data-1-*.dat
        stats.py $@ $^

summary-2.dat : data-2-*.dat
        stats.py $@ $^
```

The makefile so far

```
paper.pdf : paper.wdp figure-1.svg figure-2.svg
        wdp2pdf $<

figure-1.svg : summary-1.dat
        sgr -N -r $@ $^

figure-2.svg : summary-2.dat
        sgr -N -r $@ $^

summary-1.dat : data-1-*.dat
        stats.py $@ $^

summary-2.dat : data-2-*.dat
        stats.py $@ $^
```

Still some redundancy

The makefile so far

```
paper.pdf : paper.wdp figure-1.svg figure-2.svg
        wdp2pdf $<

figure-1.svg : summary-1.dat
        sgr -N -r $@ $^

figure-2.svg : summary-2.dat
        sgr -N -r $@ $^

summary-1.dat : data-1-*.dat
        stats.py $@ $^

summary-2.dat : data-2-*.dat
        stats.py $@ $^
```

Still some redundancy

Fix in next episode

The makefile so far

```
paper.pdf : paper.wdp figure-1.svg figure-2.svg
        wdp2pdf $<

figure-1.svg : summary-1.dat
        sgr -N -r $@ $^

figure-2.svg : summary-2.dat
        sgr -N -r $@ $^

summary-1.dat : data-1-*.dat
        stats.py $@ $^

summary-2.dat : data-2-*.dat
        stats.py $@ $^
```

Still some redundancy

Fix in next episode

Doesn't handle summaries' dependency on `stats.py`

Option 1: add to existing rules

```
paper.pdf : paper.wdp figure-1.svg figure-2.svg
        wdp2pdf $<

figure-1.svg : summary-1.dat
        sgr -N -r $@ $^

figure-2.svg : summary-2.dat
        sgr -N -r $@ $^

summary-1.dat : stats.py data-1-*.dat
        stats.py $@ $^

summary-2.dat : stats.py data-2-*.dat
        stats.py $@ $^
```

Option 1: add to existing rules

```
⋮

summary-1.dat : stats.py data-1-*.dat
        stats.py $@ $^

summary-2.dat : stats.py data-2-*.dat
        stats.py $@ $^

⋮
```

$^ is now `stats.py data-1-1.dat data-1-1.dat` ...

Option 1: add to existing rules

```
⋮

summary-1.dat : stats.py data-1-*.dat
        stats.py $@ $^

summary-2.dat : stats.py data-2-*.dat
        stats.py $@ $^

⋮
```

`$^` is now `stats.py data-1-1.dat data-1-1.dat ...`

So the invocation of `stats.py` is wrong

Option 1: add to existing rules

```
⋮

summary-1.dat : stats.py data-1-*.dat
        stats.py $@ $^

summary-2.dat : stats.py data-2-*.dat
        stats.py $@ $^

⋮
```

`$^` is now `stats.py data-1-1.dat data-1-1.dat ...`

So the invocation of `stats.py` is wrong

Having it ignore one argument is an ugly hack

Option 2: make data files depend on `stats.py`

```
figure-2.svg : summary-2.dat
        sgr -N -r $@ $^

summary-1.dat : data-1-*.dat
        stats.py $@ $^

summary-2.dat : data-2-*.dat
        stats.py $@ $^

data-1-1.dat : stats.py
        touch $@

data-1-2.dat : stats.py
        touch $@
```

Option 2: make data files depend on `stats.py`

```
⋮

data-1-1.dat : stats.py
        touch $@

data-1-2.dat : stats.py
        touch $@

⋮
```

A *false dependency*

Option 2: make data files depend on `stats.py`

```
⋮

data-1-1.dat : stats.py
        touch $@

data-1-2.dat : stats.py
        touch $@

⋮
```

A *false dependency*

Updating raw data files triggers update of summary

Option 2: make data files depend on `stats.py`

```
⋮

data-1-1.dat : stats.py
        touch $@

data-1-2.dat : stats.py
        touch $@

⋮
```

A *false dependency*

Updating raw data files triggers update of summary

Back to listing all raw data files explicitly...

Option 3: add additional dependencies

```
paper.pdf : paper.wdp figure-1.svg figure-2.svg
        wdp2pdf $<

figure-1.svg : summary-1.dat
        sgr -N -r $@ $^

figure-2.svg : summary-2.dat
        sgr -N -r $@ $^

summary-1.dat : data-1-*.dat
        stats.py $@ $^

summary-2.dat : data-2-*.dat
        stats.py $@ $^

summary-1.dat : stats.py
summary-2.dat : stats.py
```

Option 3: add additional dependencies

```
⋮

summary-1.dat : data-1-*.dat
        stats.py $@ $^

summary-2.dat : data-2-*.dat
        stats.py $@ $^

summary-1.dat : stats.py
summary-2.dat : stats.py

⋮
```

Full set of dependencies is union of lists

Option 3: add additional dependencies

```
⋮

summary-1.dat : data-1-*.dat
        stats.py $@ $^

summary-2.dat : data-2-*.dat
        stats.py $@ $^

summary-1.dat : stats.py
summary-2.dat : stats.py

⋮
```

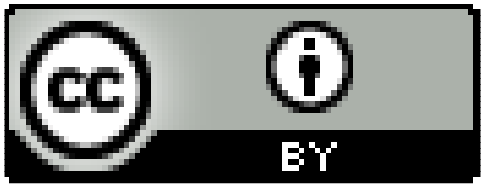
Full set of dependencies is union of lists

But `$^` in the action is still just `data-1-*.dat`

created by

Greg Wilson

August 2010



Copyright © Software Carpentry 2010
This work is licensed under the Creative Commons Attribution License
See http://software-carpentry.org/license.html for more information.