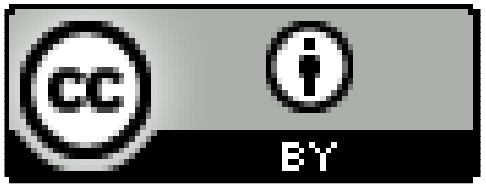
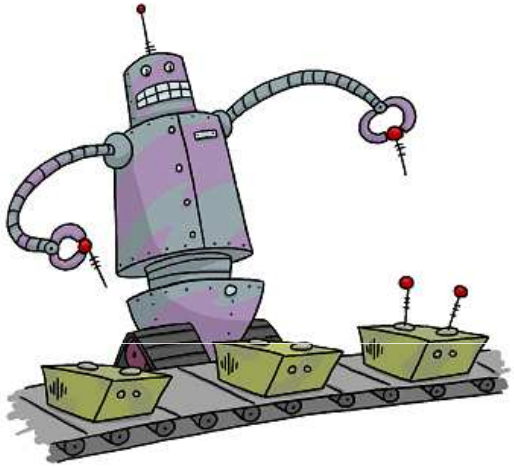
Automated Builds

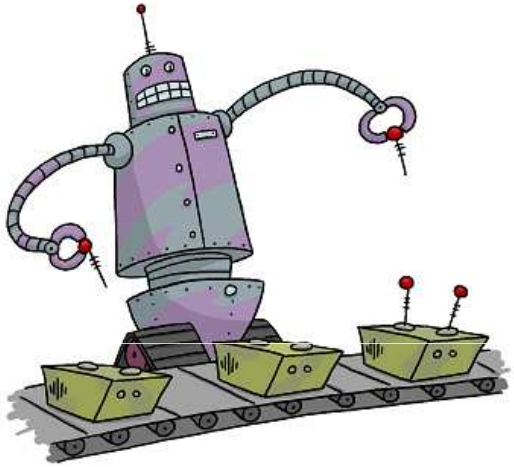
Basics

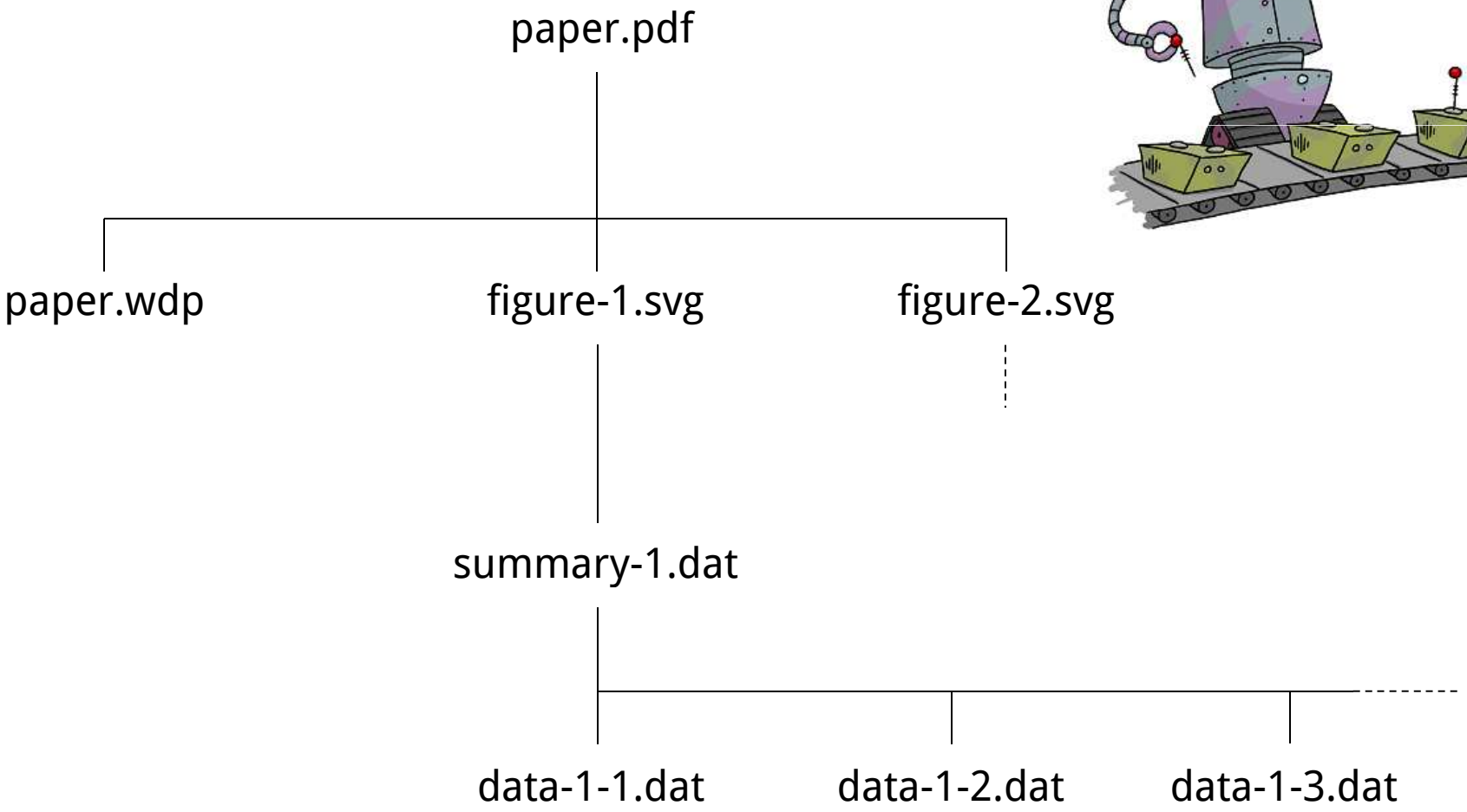


Manage tasks and dependencies

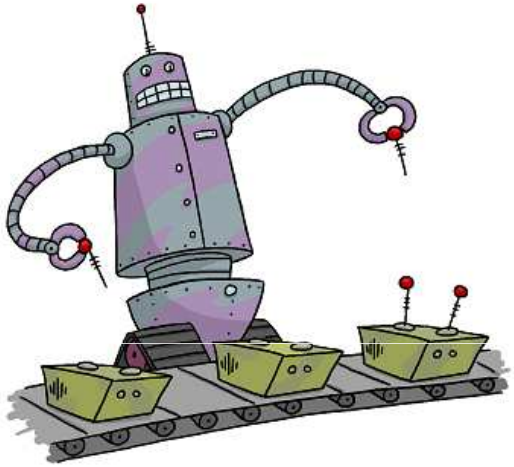


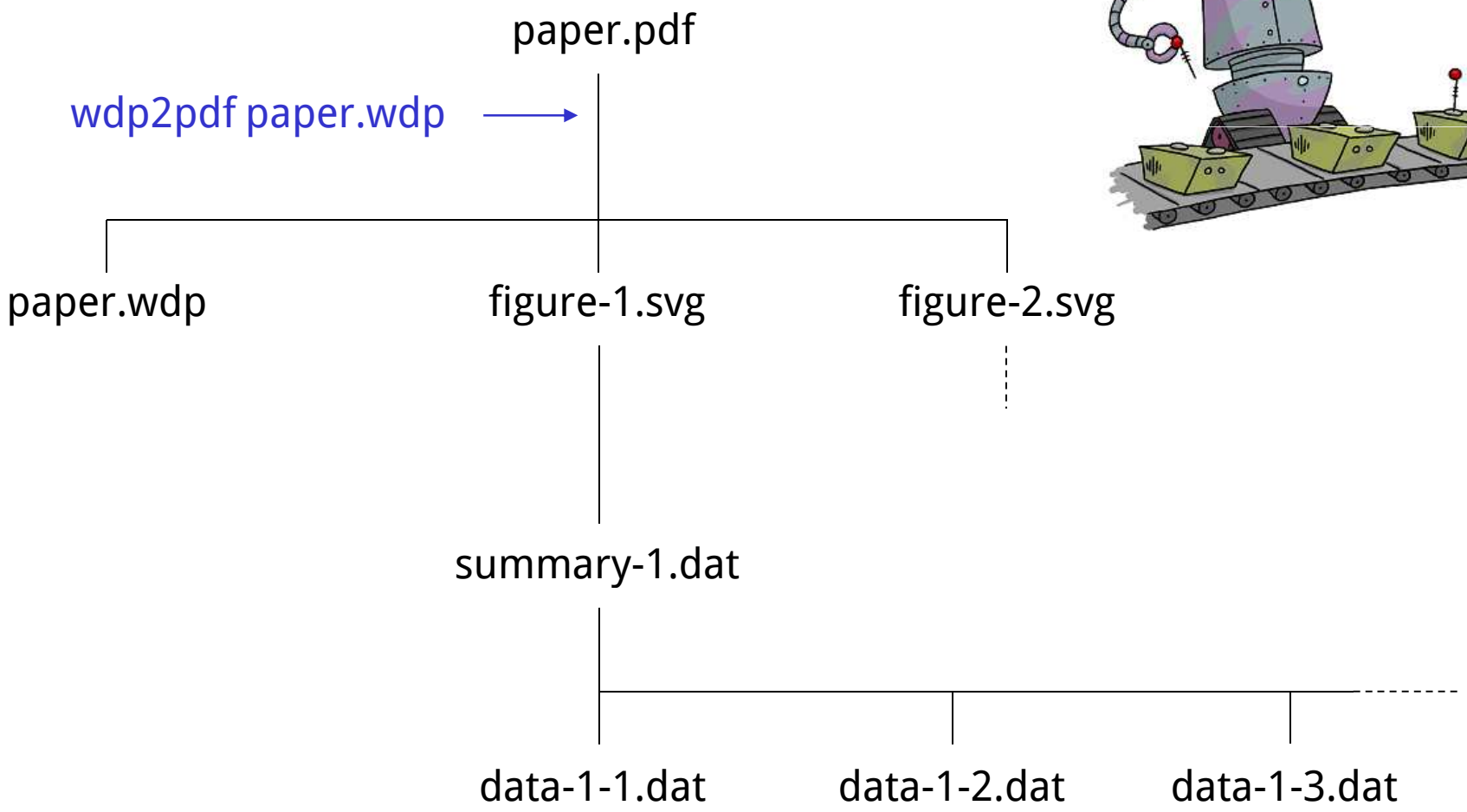
Manage tasks and dependencies



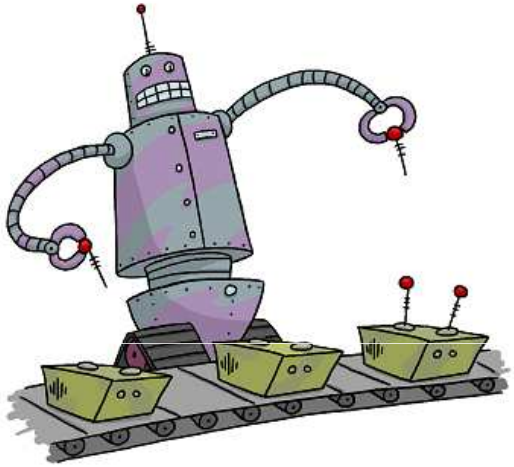


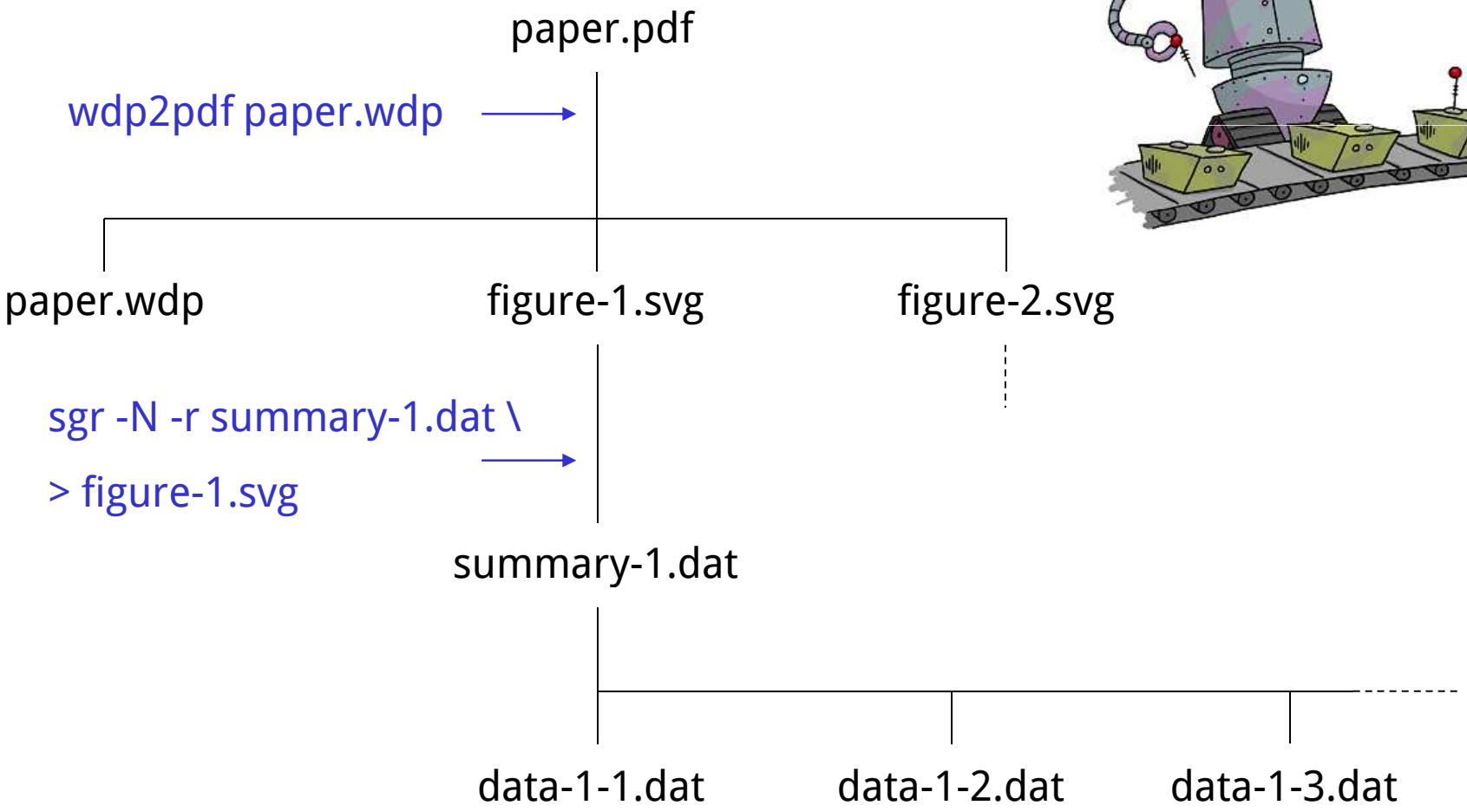
Manage tasks and dependencies



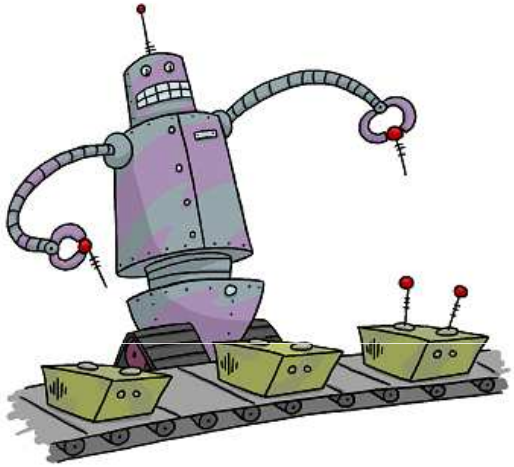


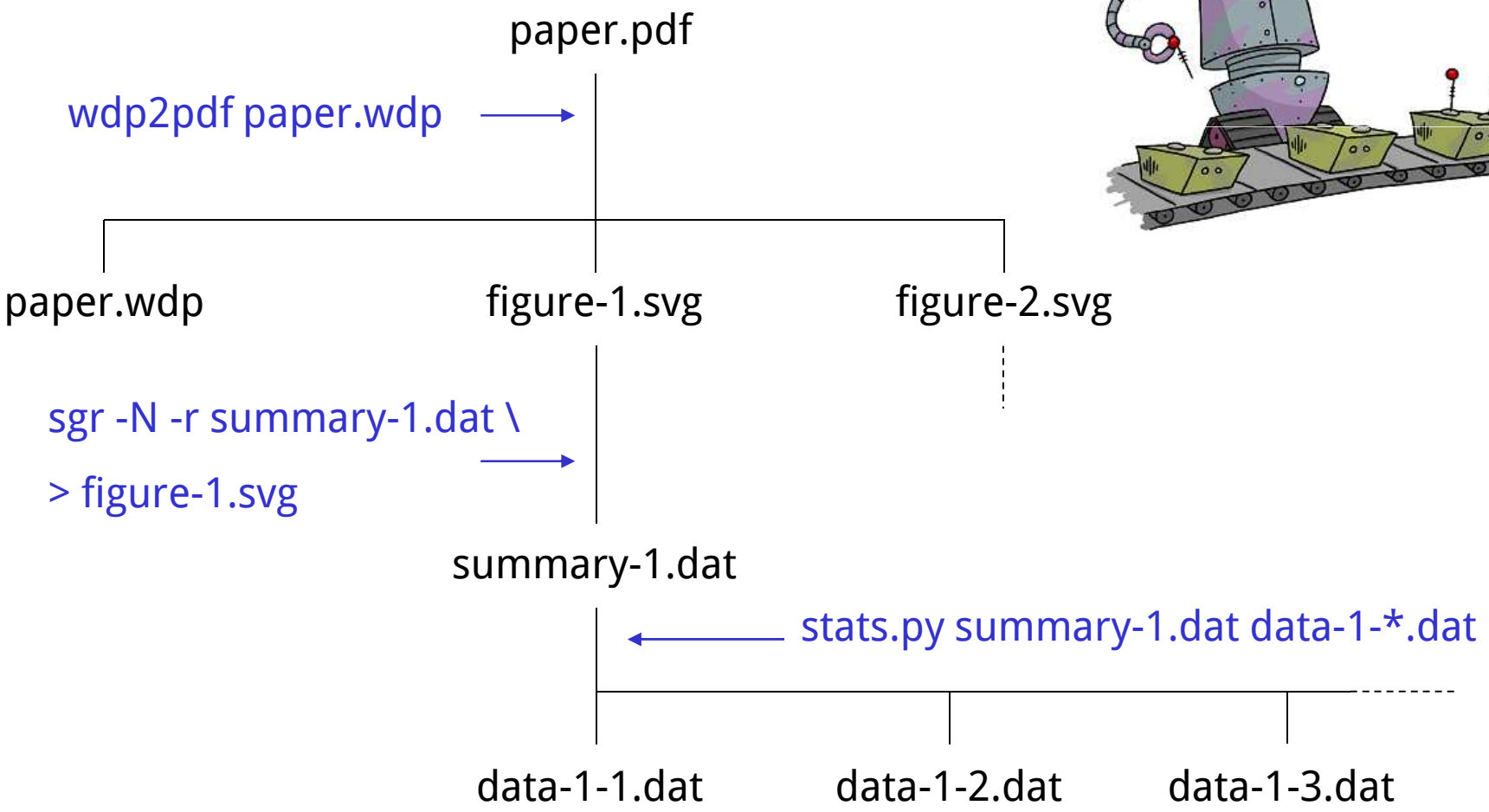
Manage tasks and dependencies



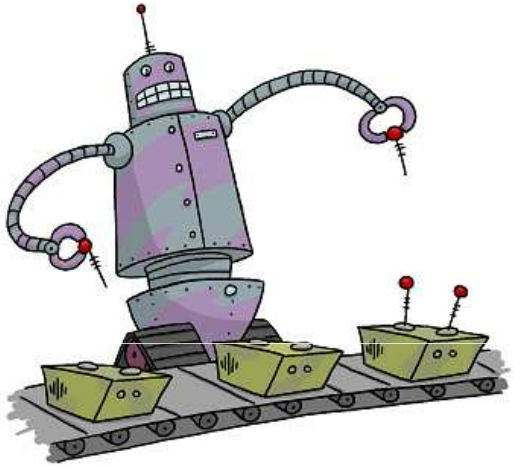


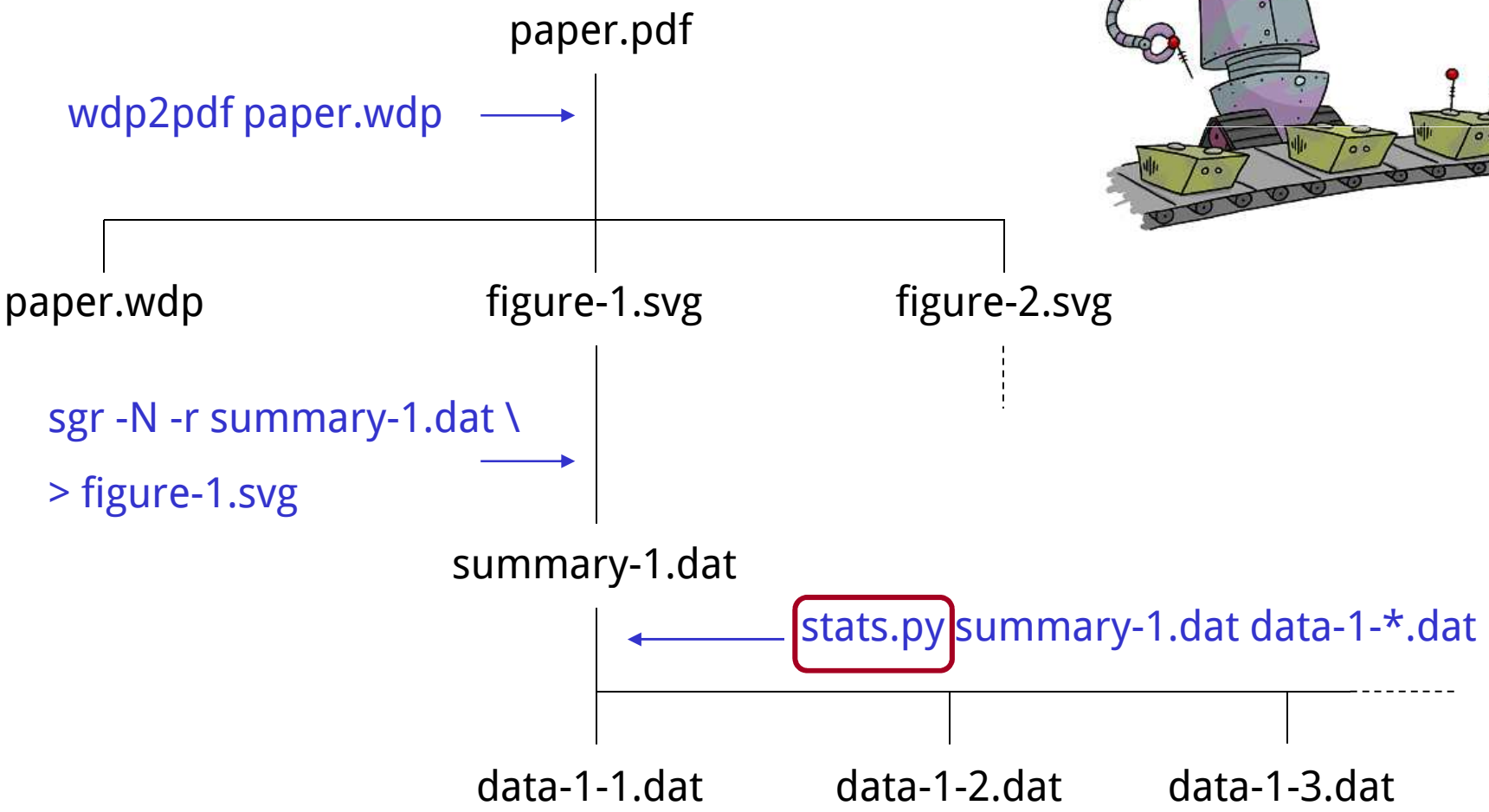
Manage tasks and dependencies



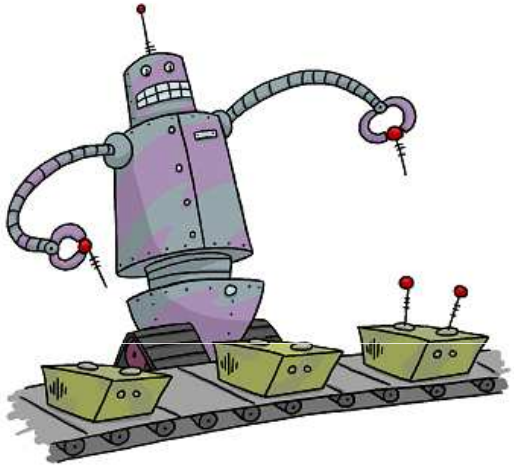


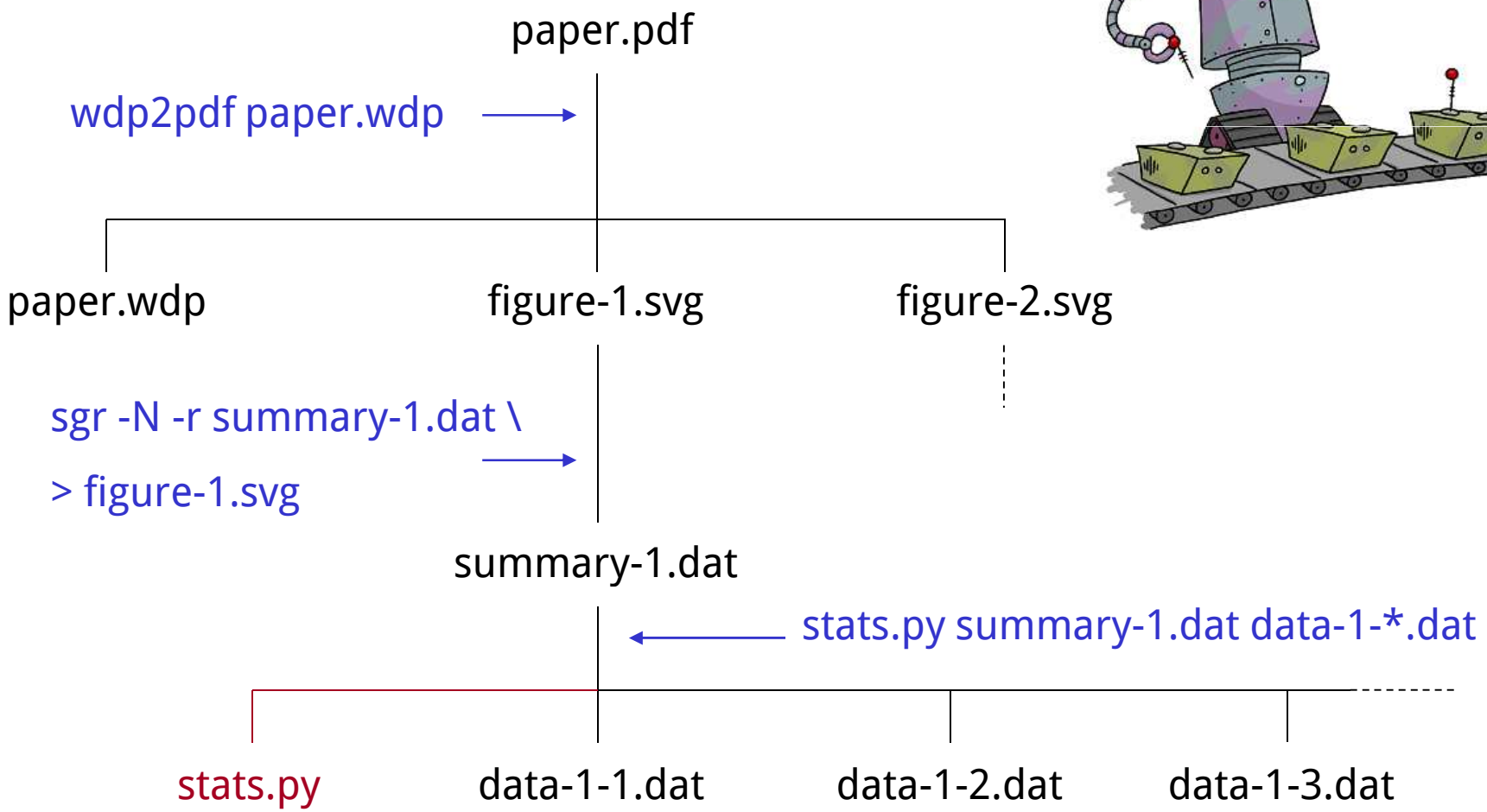
Manage tasks and dependencies



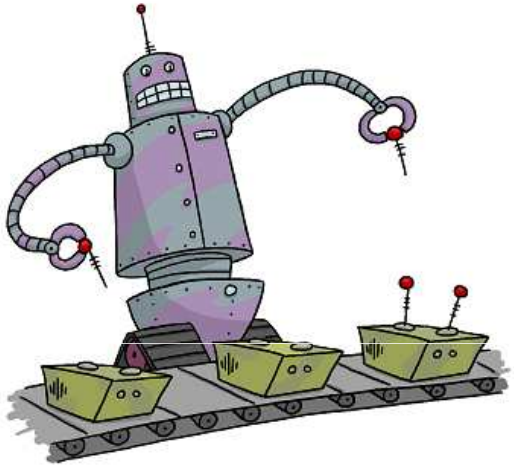


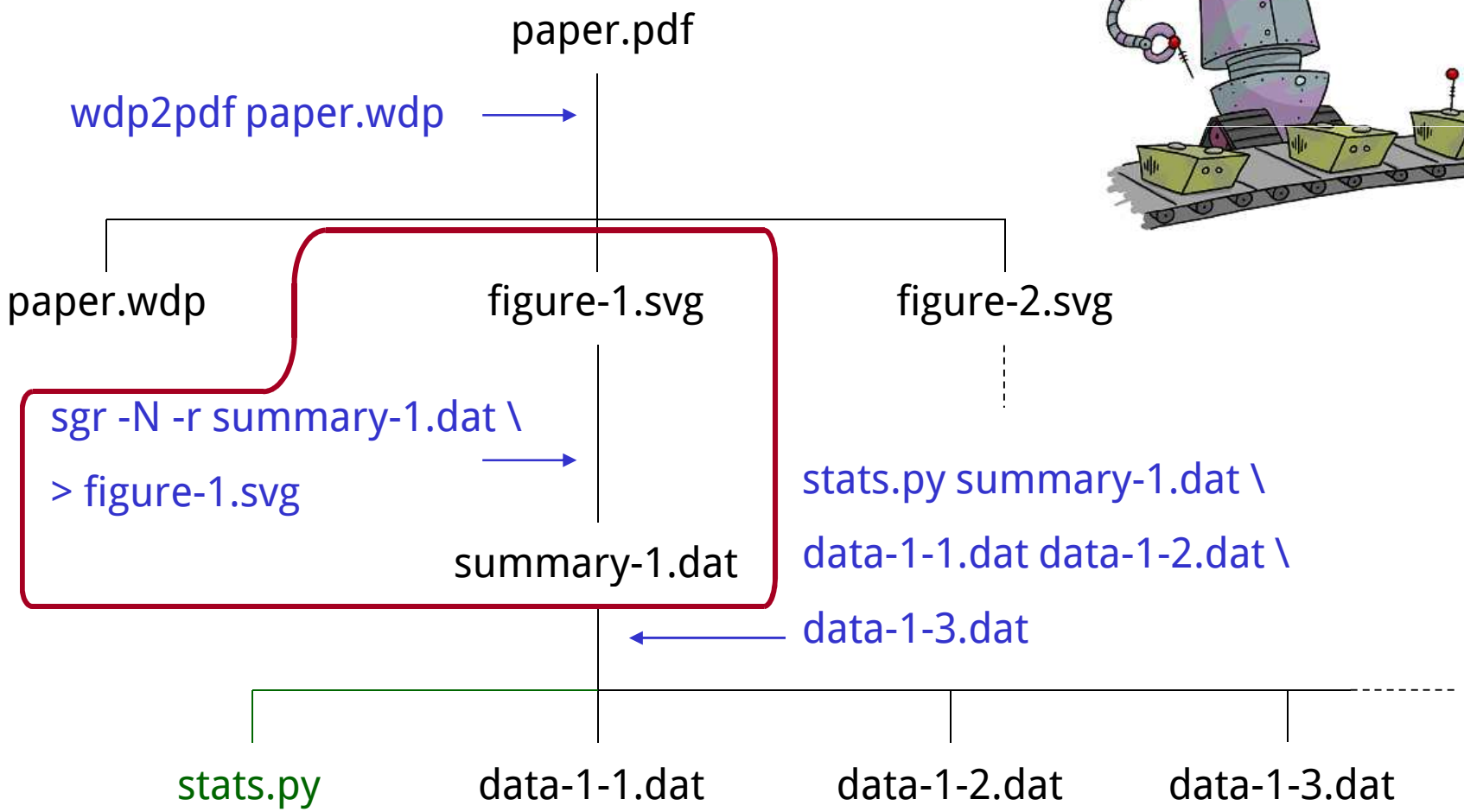
Manage tasks and dependencies





Manage tasks and dependencies





1. What is newer than what?

```
$ ls -t *.dat *.svg
summary-1.dat    figure-1.svg
$
```

1. What is newer than what?

```
$ ls -t *.dat *.svg
summary-1.dat    figure-1.svg
$
```

1. What is newer than what?

```
$ ls -t *.dat *.svg
summary-1.dat    figure-1.svg
$
```

Data file is newer than SVG image

1. What is newer than what?

```
$ ls -t *.dat *.svg
summary-1.dat    figure-1.svg
$
```

2. Put this in a *Makefile* called `hello.mk`

```
# hello.mk
figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg
```

1. What is newer than what?

```
$ ls -t *.dat *.svg
summary-1.dat    figure-1.svg
$
```

2. Put this in a *Makefile* called `hello.mk`

```
# hello.mk
figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg
```

Comment

1. What is newer than what?

```
$ ls -t *.dat *.svg
summary-1.dat    figure-1.svg
$
```

2. Put this in a *Makefile* called `hello.mk`

```
# hello.mk
figure-1.svg : summary-1.dat
        sgr -N -r summary-1.dat > figure-1.svg
```

Rule

1. What is newer than what?

```
$ ls -t *.dat *.svg
summary-1.dat    figure-1.svg
$
```

2. Put this in a *Makefile* called `hello.mk`

```
# hello.mk
figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg
```

Target

1. What is newer than what?

```
$ ls -t *.dat *.svg
summary-1.dat    figure-1.svg
$
```

2. Put this in a *Makefile* called `hello.mk`

```
# hello.mk
figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg
```

Prerequisite

1. What is newer than what?

```
$ ls -t *.dat *.svg
summary-1.dat    figure-1.svg
$
```

2. Put this in a *Makefile* called `hello.mk`

```
# hello.mk
figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg
```

Action

1. What is newer than what?

```
$ ls -t *.dat *.svg
summary-1.dat    figure-1.svg
$
```

2. Put this in a *Makefile* called `hello.mk`

```
# hello.mk
figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg
```

Must indent with a single tab character

3. Run GNU Make from the shell

```
$ gmake -f hello.mk
```

3. Run GNU Make from the shell

```
$ gmake -f hello.mk
```

-f filename

3. Run GNU Make from the shell

```
$ gmake -f hello.mk
sgr -N -r summary-1.dat > figure-1.svg
$
```

3. Run GNU Make from the shell

```
$ gmake -f hello.mk
sgr -N -r summary-1.dat > figure-1.svg
$
```

Prerequisite is newer than target

3. Run GNU Make from the shell

```
$ gmake -f hello.mk
sgr -N -r summary-1.dat > figure-1.svg
$
```

Prerequisite is newer than target

So Make executes the action

3. Run GNU Make from the shell

```
$ gmake -f hello.mk
sgr -N -r summary-1.dat > figure-1.svg
$
```

4. Run Make again

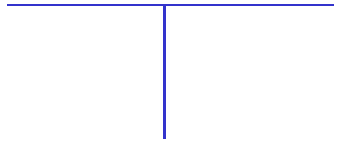
```
$ gmake -f hello.mk
$
```

3. Run GNU Make from the shell

```
$ gmake -f hello.mk
sgr -N -r summary-1.dat > figure-1.svg
$
```

4. Run Make again

```
$ gmake -f hello.mk
$
```



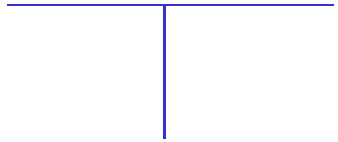
Target is newer than prerequisite

3. Run GNU Make from the shell

```
$ gmake -f hello.mk
sgr -N -r summary-1.dat > figure-1.svg
$
```

4. Run Make again

```
$ gmake -f hello.mk
$
```



Target is newer than prerequisite

So action not executed

Usually have multiple rules per file

Usually have multiple rules per file

```
# double.mk

figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg

figure-2.svg : summary-2.dat
	sgr -N -r summary-2.dat > figure-2.svg
```

Usually have multiple rules per file

```
# double.mk

figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg

figure-2.svg : summary-2.dat
	sgr -N -r summary-2.dat > figure-2.svg
```

Force it to run

```
$ touch *.dat
$
```

Usually have multiple targets per file

```
# double.mk

figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg

figure-2.svg : summary-2.dat
	sgr -N -r summary-2.dat > figure-2.svg
```

Force it to run

```
$ touch *.dat
$
```


Update timestamps on files

Usually have multiple targets per file

```
# double.mk

figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg

figure-2.svg : summary-2.dat
	sgr -N -r summary-2.dat > figure-2.svg
```

Force it to run

```
$ touch *.dat
$ gmake -f double.mk
sgr -N -r summary-1.dat > figure-1.svg
$
```

Usually have multiple targets per file

```
# double.mk

figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg

figure-2.svg : summary-2.dat
	sgr -N -r summary-2.dat > figure-2.svg
```

Force it to run

```
$ touch *.dat
$ gmake -f double.mk
sgr -N -r summary-1.dat > figure-1.svg
$
```

Why isn't `figure-2.svg` rebuilt?

First rule in Makefile is *default rule*

First rule in Makefile is *default rule*

Make only does this unless told otherwise

First rule in Makefile is *default rule*

Make only does this unless told otherwise

Force it to rebuild `figure-2.svg` explicitly

First rule in Makefile is *default rule*

Make only does this unless told otherwise

Force it to rebuild `figure-2.svg` explicitly

```
$ gmake -f double.mk figure-2.svg
sgr -N -r summary-2.dat > figure-2.svg
$
```

First rule in Makefile is *default rule*

Make only does this unless told otherwise

Force it to rebuild `figure-2.svg` explicitly

```
$ gmake -f double.mk figure-2.svg
sgr -N -r summary-2.dat > figure-2.svg
$
```

Better than typing commands

one by one, but only slightly

Introduce a *phony target*

Introduce a *phony target*

Doesn't correspond to a file

Introduce a *phony target*

Doesn't correspond to a file

So never up to date

Introduce a *phony target*

Doesn't correspond to a file

So never up to date

But things can depend on it

Introduce a *phony target*

Doesn't correspond to a file

So never up to date

But things can depend on it

```
# phony.mk

all : figure-1.svg figure-2.svg

figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg

figure-2.svg : summary-2.dat
	sgr -N -r summary-2.dat > figure-2.svg
```

Introduce a *phony target*

Doesn't correspond to a file

So never up to date

But things can depend on it

```
# phony.mk

all : figure-1.svg figure-2.svg

figure-1.svg : summary-1.dat
	sgr -N -r summary-1.dat > figure-1.svg

figure-2.svg : summary-2.dat
	sgr -N -r summary-2.dat > figure-2.svg
```

"`make all`" rebuilds everything

"make all" rebuilds everything

```
$ touch *.dat
$ gmake -f phony.mk
sgr -N -r summary-1.dat > figure-1.svg
sgr -N -r summary-2.dat > figure-2.svg
$
```

"make all" rebuilds everything

```
$ touch *.dat
$ gmake -f phony.mk
sgr -N -r summary-1.dat > figure-1.svg
sgr -N -r summary-2.dat > figure-2.svg
$
```

Order is not guaranteed

"make all" rebuilds everything

```
$ touch *.dat
$ gmake -f phony.mk
sgr -N -r summary-1.dat > figure-1.svg
sgr -N -r summary-2.dat > figure-2.svg
$
```

Order is not guaranteed

Could re-create `figure-2.svg` first

"make all" rebuilds everything

```
$ touch *.dat
$ gmake -f phony.mk
sgr -N -r summary-1.dat > figure-1.svg
sgr -N -r summary-2.dat > figure-2.svg
$
```

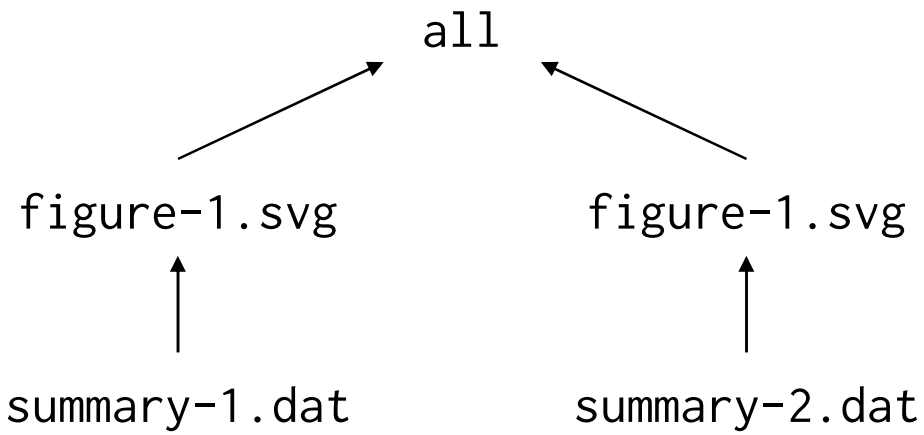
- Order is not guaranteed
- Could re-create `figure-2.svg` first
- Or re-create both in parallel

"make all" rebuilds everything

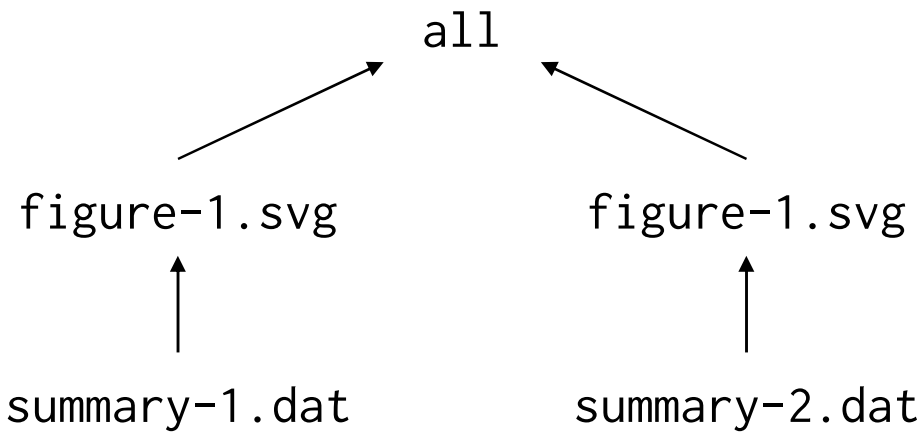
```
$ touch *.dat
$ gmake -f phony.mk
sgr -N -r summary-1.dat > figure-1.svg
sgr -N -r summary-2.dat > figure-2.svg
$
```

- Order is not guaranteed
- Could re-create `figure-2.svg` first
- Or re-create both in parallel
- Return to that idea later

Things can be both targets and prerequisites

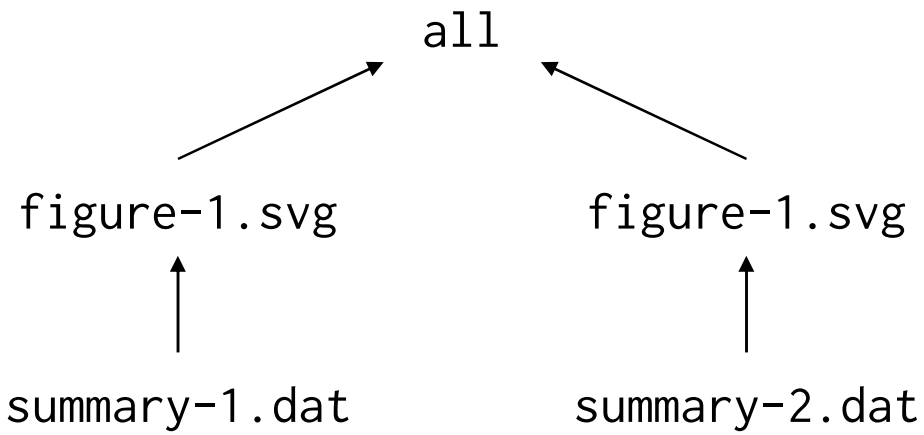


Things can be both targets and prerequisites



A *directed graph*

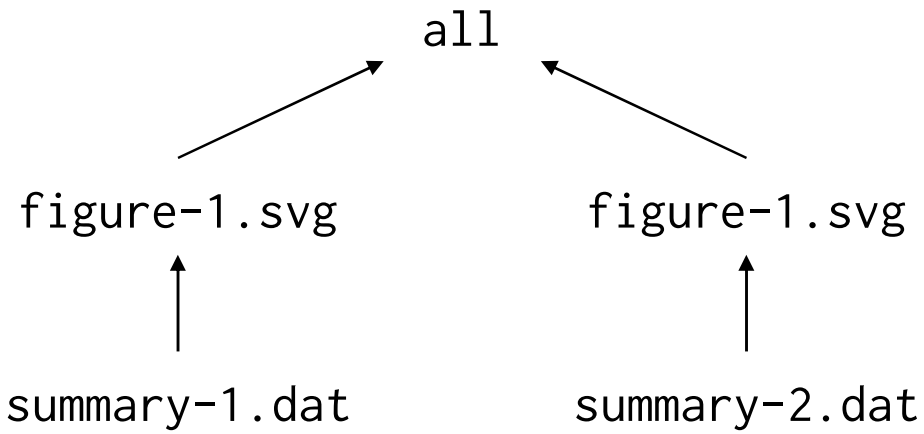
Things can be both targets and prerequisites



A *directed graph*

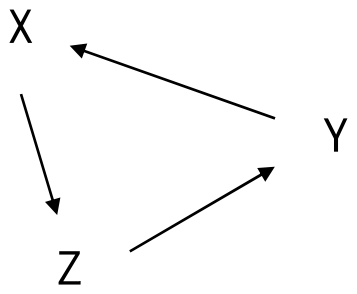
Must be *acyclic*

Things can be both targets and prerequisites

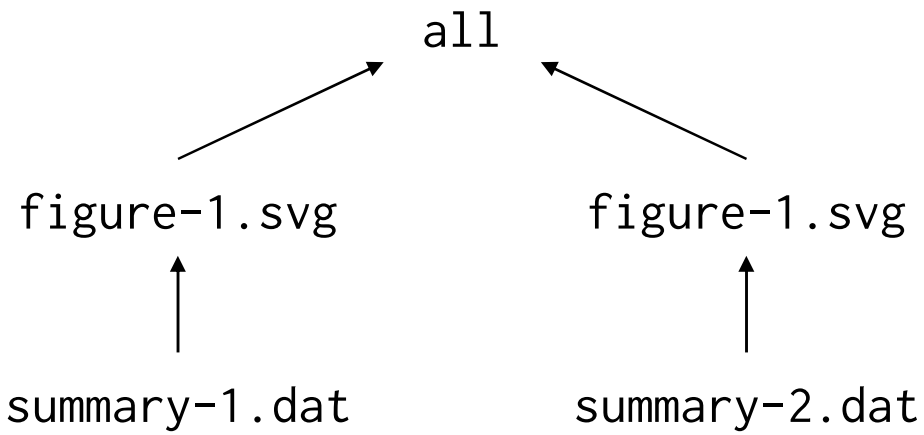


A *directed graph*

Must be *acyclic*

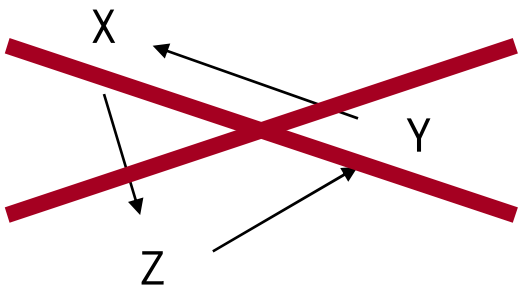


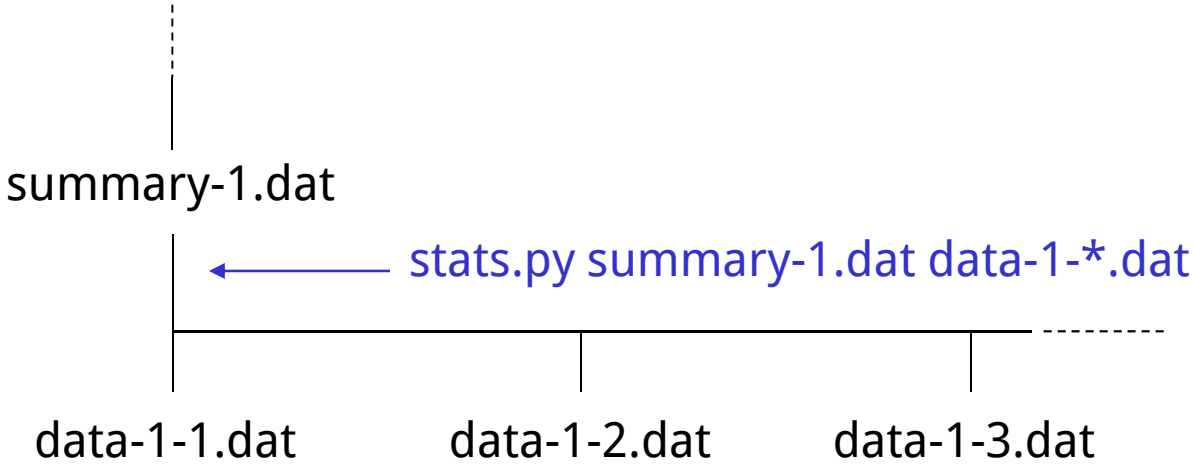
Things can be both targets and prerequisites

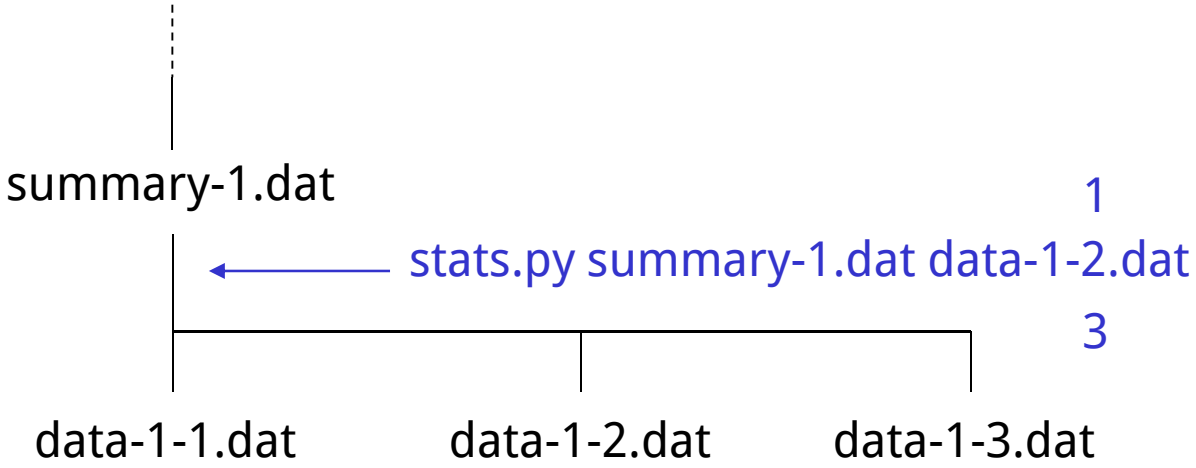


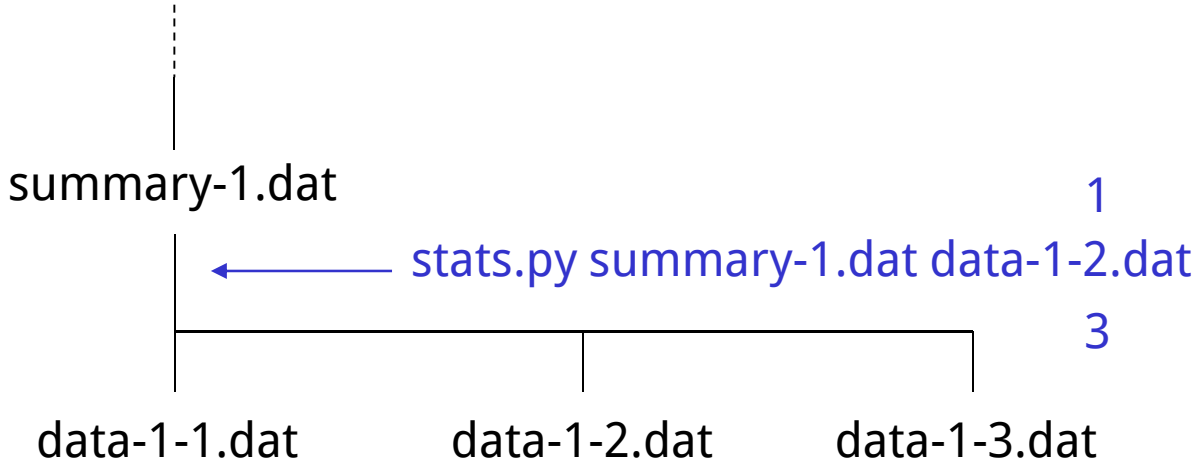
A *directed graph*

Must be *acyclic*

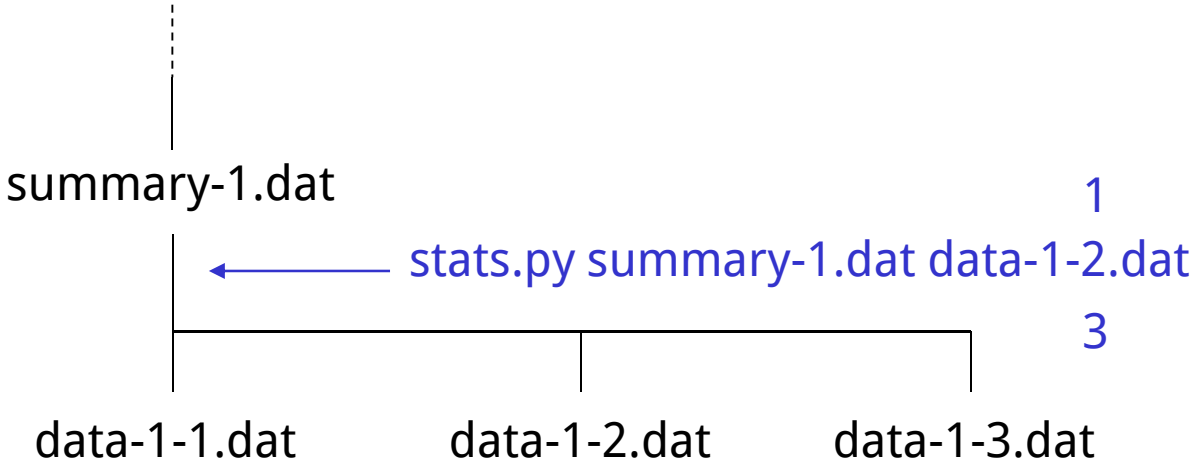



summary-1.dat
stats.py summary-1.dat data-1-*.dat
data-1-1.dat
data-1-2.dat
data-1-3.dat


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

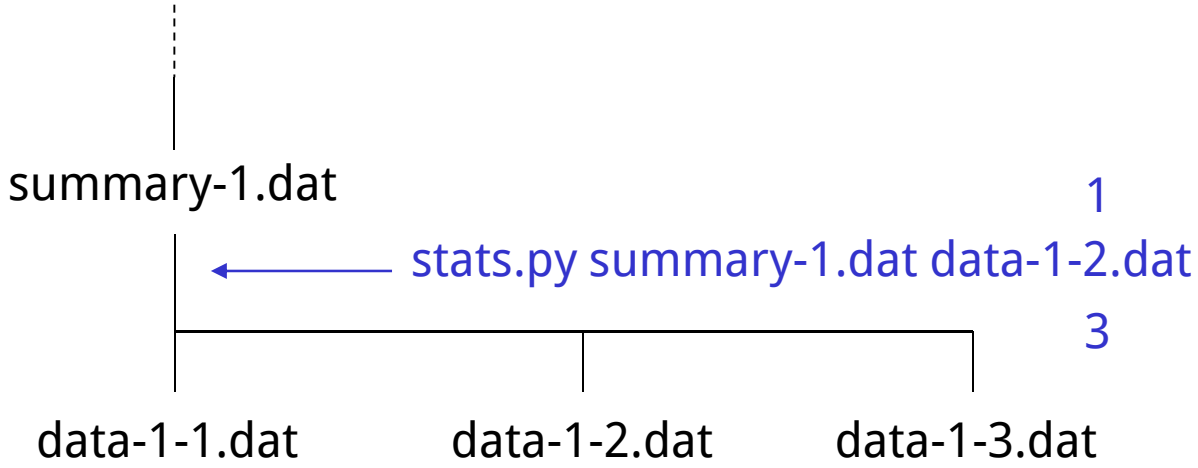



```
# multiple.mk

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
```




```
# 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
```

How to generalize to any number of files?




```
# 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
```

How to generalize to any number of files?

And get rid of repeated filenames

created by

Greg Wilson

August 2010