



Experis®
ManpowerGroup

The Art of Workflowing Your Clinical Submission Scripts

Brian Varney, Experis
Kevin Putschko, Experis

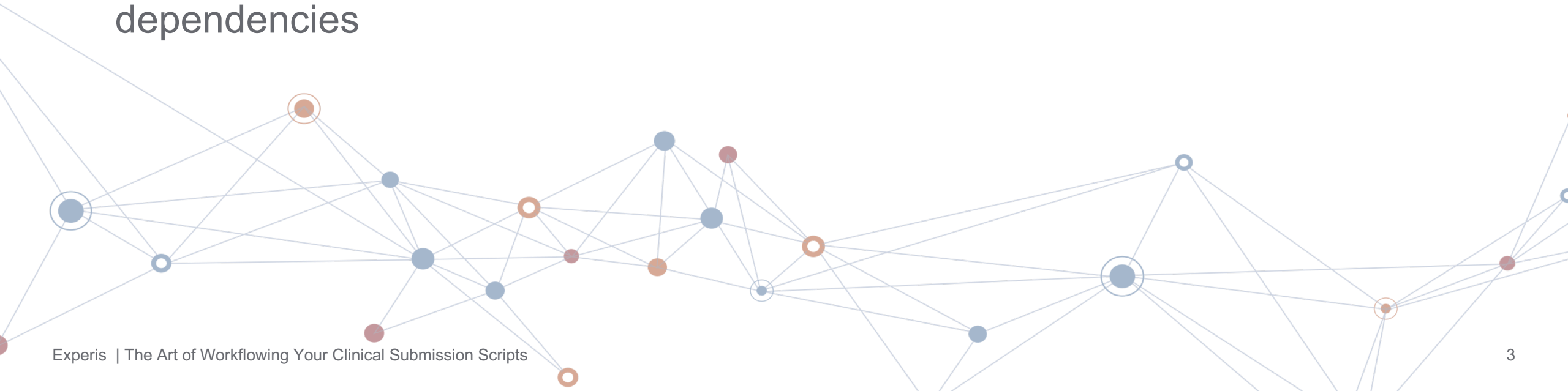


Agenda

- 1 What do we mean by a workflow?
- 2 SAS approach using '%include'
- 3 R approach using source function
- 4 The R targets package
- 5 Selected 'targets' functions
- 6 High-level code review
- 7 Visual network
- 8 Conclusions

What Do We Mean by a Workflow?

- Running a series of programs in a repeatable and convenient manner
- Typically uses a driver program
- Global variables and options are set at the beginning of the workflow
- The order that the programs are executed typically matters because of dependencies



SAS Approach Using %include

Example Situation:

A driver program has been set up to run the ADaM data set builds in SAS. This consists of selected ADaM data set builds and quality checks.

In SAS this would typically consist of:

- %include statement to set up global variables and options
- %include statement for each build program
- %include statement for each quality check

SAS Approach Using %include (continued)

Example Situation SAS Code:

```
/*  Program:  driver program for Study ABC ADaM Data Set Builds  */  
%include "c:\...\programs\studyabc\general\setup.sas";  
%include "c:\...\programs\studyabc\ADaM\setup.sas";  
%include "c:\...\programs\studyabc\ADaM\adsl.sas";  
%include "c:\...\programs\studyabc\ADaM\adae.sas";  
%include "c:\...\programs\studyabc\ADaM\quality_checks.sas";
```

SAS Enterprise Guide is another tactic, but we will not include it in this presentation.

R Approaches

We will look at two approaches using R.

- Our first approach will use the `source()` function.
 - The `source()` function is very similar to the `%include` in SAS
- Our second approach will use the `targets` package.
 - The `targets` package provides a more robust and visual approach

R Approach Using source() Function

Example Situation R Code:

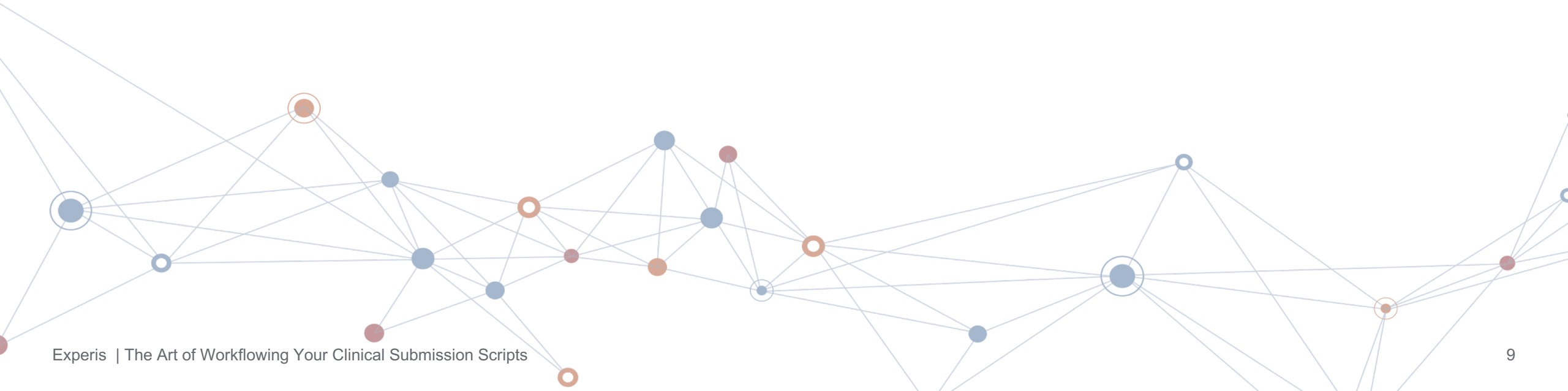
```
# Program: driver program for Study ABC ADaM Data Set Builds  
source("c:/.../programs/studyabc/general/setup.R")  
source("c:/.../programs/studyabc/ADaM/setup.R")  
source("c:/.../programs/studyabc/ADaM/adsl.R")  
source("c:/.../programs/studyabc/ADaM/adae.R")  
source("c:/.../programs/studyabc/ADaM/quality_checks.R")
```

The R Targets Package

- Evolved from the drake package
- A make-like pipeline toolkit for statistics and data science in R
- Easy to configure parallel computing
- Enhances reproducibility and reduces burdens of repeated computation and data management
- Provides tangible evidence that the output aligns with code and data

The R Targets Package (continued)

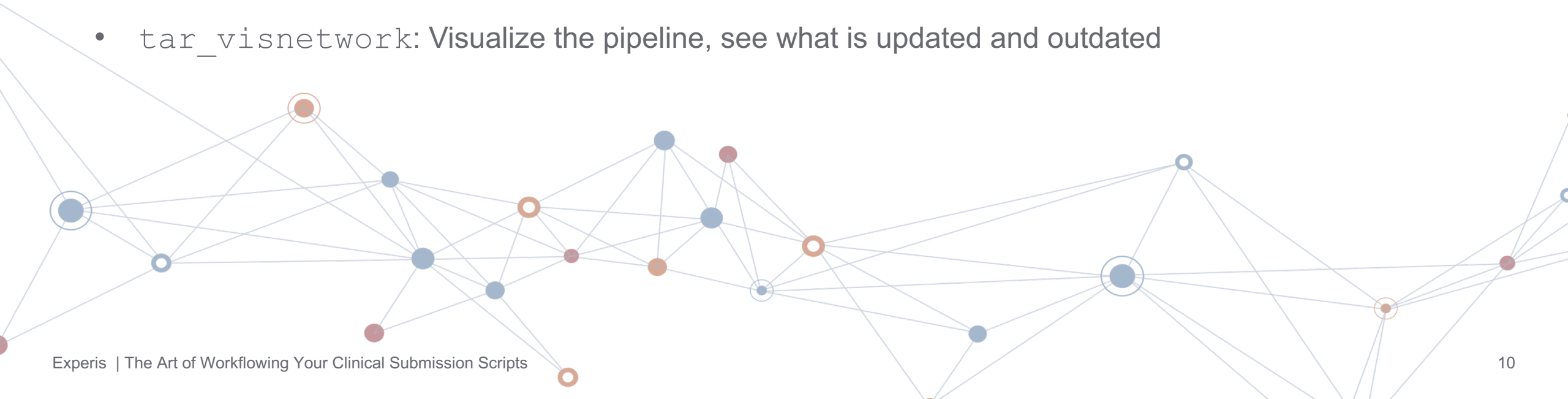
- Expects programmers to adopt a function-oriented style of programming
- Can display the workflow graphically



Selected Targets Package Functions

- **Selected Targets Functions**

- `tar_script`: Define the structure of your workflow pipeline
- `tar_target`: Define the individual steps of the pipeline
- `tar_make`: Run the pipeline, automatically skipping steps when it can
- `tar_visnetwork`: Visualize the pipeline, see what is updated and outdated



Code Synopsis

```
1
2 # Load Functions to Help Define the Pipeline
3 library(targets) # To use tar_script, tar_option_set, and tar_target
4 library(dplyr)   # To use %>% in command definitions if desired
5
6
7 ## Define the Pipeline ----
8 targets::tar_script(      # Create '_targets.R' file in project/working directory
9   ask = FALSE,           # Set 'ask' to TRUE to confirm overwrite of existing file
10  code = {                # Use {...} to wrap many operations in the pipeline
11
12
13    # -- Load Packages & Personal Functions --
14    # These are the packages and functions that will be required in the pipeline
15    tar_option_set(packages = c("dplyr", "ggplot2", "stringr", "readr",
16                                "janitor", "broom", "tibble"))
17
18    source("code/pipeline_fns.R")|
19
```

Code Synopsis (continued)

```
20
21 # -- Create a Pipeline Definition with `list()` --
22 # Use a list() to hold the `tar_target()` pipeline steps
23 # 'Name' gives the target an informative name so we know what the process is doing
24 # 'Command' is the R code we want to run at this step
25 # 'Format' means that the pipeline will monitor this file for changes
26 list(
27
28   # Specifying format = "file" can monitor any type of file for changes
29   tar_target(
30     name      = monitor_adam_ads1,
31     command   = "data/ADaM/ads1.rds",
32     format    = "file"),
33
34   tar_target(
35     name      = monitor_sdtm_ae,
36     command   = "data/SDTM/ae.rds",
37     format    = "file"),
38
```

Code Synopsis (continued)

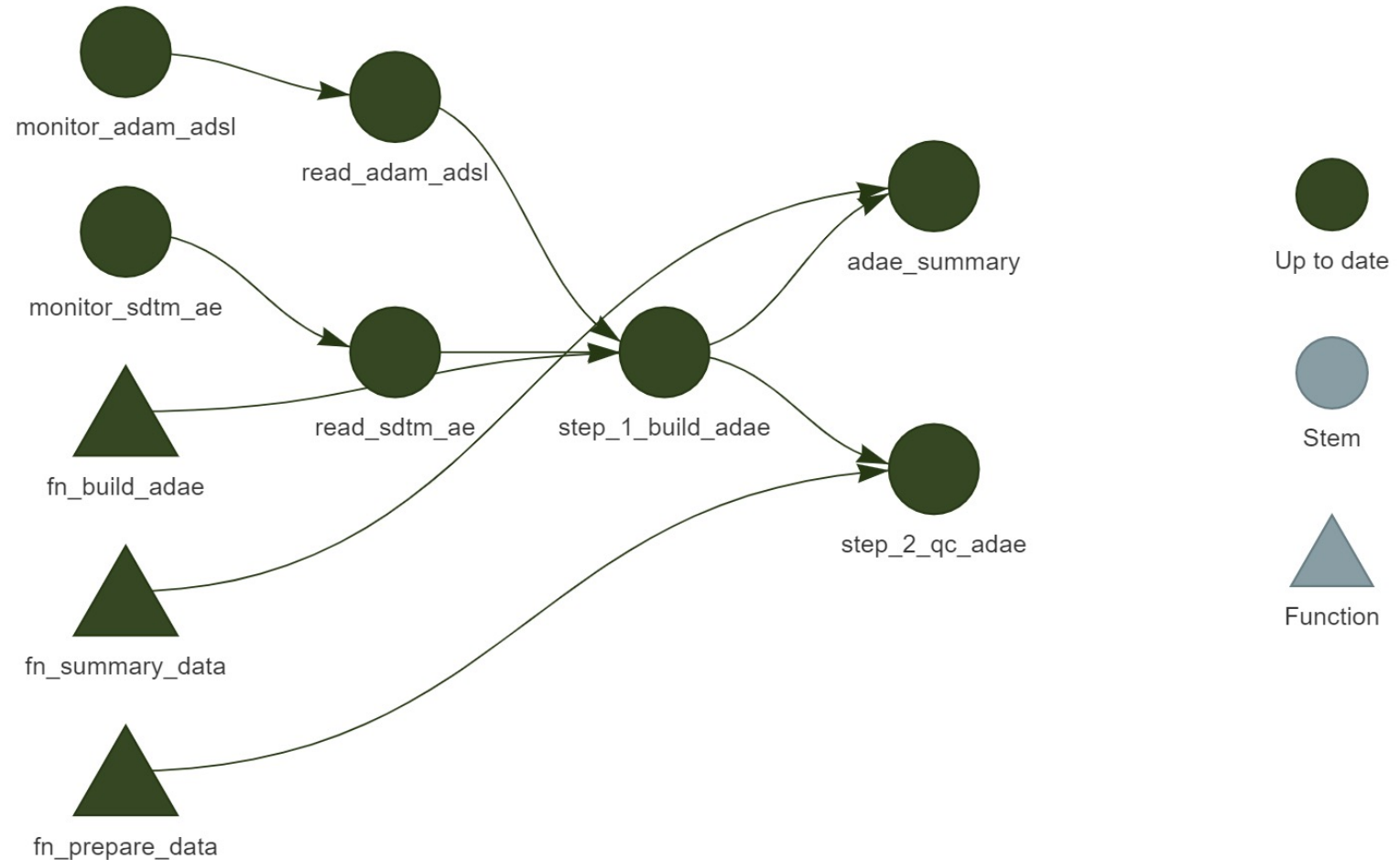
```
39     # Load the Data
40     tar_target(
41       name      = read_adam_ads1,
42       command   = readRDS(monитор_adam_ads1)
43     ),
44
45     tar_target(
46       name      = read_sdtm_ae,
47       command   = монитор_sdtm_ae %>% readRDS()
48     ),
49
```

Code Synopsis (continued)

```
50 # The 'command' argument accepts any amount of R code, here we begin using
51 # our personal functions
52 tar_target(
53   name      = step_1_build_adae,
54   command   = fn_build_adae(read_adam_ads1, read_sdtm_ae)),
55
56
57 # But using custom functions will make it easier to debug the pipeline
58 tar_target(
59   name      = step_2_qc_adae,
60   command   = fn_prepare_data(step_1_build_adae)),
61
62
63 # No need to keep track of sequence order, 'targets' will track it for us
64 # Notice the following steps are not in sequential order:
65 tar_target(name = adae_summary, fn_summary_data(step_1_build_adae))
66 )
67 }
68
```

Results from tar_visnetwork() Function

```
# Run and view the Pipeline  
# Build Everything  
tar_make()  
# View the Pipeline  
tar_visnetwork()
```



Conclusions

- The *targets* package offers a simplified method of managing complex data pipelines.
- There is a learning curve in:
 - building your operations around functions
 - defining appropriate target steps
- However, the time is well invested in allowing you to better maintain up-to-date results without having to run the entire pipeline.

Thank you

Stop by **Booth #33** to see a demo of Accel2R!

– 4PM today –



Experis[®]
ManpowerGroup

Accel2R

Powered by Experis[®]