The wonderful package [ggdag](https://ggdag.netlify.com/) can easily make DAG like this:

However, what we really want to include in publications is something like this:

The second one can include subscript and superscript, among many others. After some tweaking, I found a solution, not perfect but usable for now.

—————————————————————–  
library(dagitty)  
library(ggdag)  
library(ggraph)  
library(cowplot)  
library(dplyr)  
  
“`{r, echo=FALSE}  
dag <- dagify(Y1 ~ X + Z1 + Z0 + U + P,  
              Y0 ~ Z0 + U,  
              X ~ Y0 + Z1 + Z0 + P,  
              Z1 ~ Z0,  
              P ~ Y0 + Z1 + Z0,  
              exposure = “X”,  
              outcome = “Y1”)  
  
  
dag %>%   
  tidy\_dagitty(layout = “auto”, seed = 12345) %>%  
  arrange(name) %>%   
  ggplot(aes(x = x, y = y, xend = xend, yend = yend)) +  
  geom\_dag\_point() +  
  geom\_dag\_edges() +  
  geom\_dag\_text(parse = TRUE, label = c(“P”, “U”, “X”, expression(Y[0]), expression(Y[1]), expression(Z[0]), expression(Z[1]))) +  
  theme\_dag()  
—————————————————————–

Here the trick is to sort the tidy version of the DAG data by “name”, then we can assign labels by the order of the name of the nodes. I hope a more automated approach can be developed in the future.

By the way, with thee package [latex2exp](https://github.com/stefano-meschiari/latex2exp), it is straightforward to use LaTeX instead of plotmath commands.