```
knitr::opts_chunk$set(comment='     ')
```

```
library(causalsim)
```

```
      Loading required package: dagitty
      Loading required package: ggdag


      Attaching package: 'ggdag'
      The following object is masked from 'package:stats':

          filter
```

```
nams <- c('z','zx','zy','cov','x','y','m','i', 'xd', 'yd', 'zd', 'col')
mat <- matrix(0, length(nams), length(nams))
rownames(mat) <- nams
colnames(mat) <- nams

# confounding back-door path
mat['zx','z'] <- 3
mat['zy','z'] <- 3
mat['x','zx'] <- 1
mat['y','zy'] <- 2

# direct effect
mat['y','x'] <- 3

# indirect effect
mat['m','x'] <- 1
mat['y','m'] <- 1

# Instrumental variable
mat['x','i'] <- 2

# 'Covariate'
mat['y','cov'] <- 2

# descendant of X
mat['xd','x'] <- 1

# descendant of Y
mat['yd','y'] <- 1

# descendant of z -- imperfect control
mat['zd','z'] <- 2

# collider
mat['col','y'] <- 1
mat['col','x'] <- 1


# independent error
diag(mat) <- 1
```

```
mat # not in lower diagonal form
```

```
     z zx zy cov x y m i xd yd zd col
z    1  0  0   0 0 0 0 0  0  0  0   0
zx   3  1  0   0 0 0 0 0  0  0  0   0
zy   3  0  1   0 0 0 0 0  0  0  0   0
cov  0  0  0   1 0 0 0 0  0  0  0   0
x    0  1  0   0 1 0 0 2  0  0  0   0
y    0  0  2   2 3 1 1 0  0  0  0   0
m    0  0  0   0 1 0 1 0  0  0  0   0
i    0  0  0   0 0 0 0 1  0  0  0   0
xd   0  0  0   0 1 0 0 0  1  0  0   0
yd   0  0  0   0 0 1 0 0  0  1  0   0
zd   2  0  0   0 0 0 0 0  0  0  1   0
col  0  0  0   0 1 1 0 0  0  0  0   1
```

```
dag <- to_dag(mat) # can be permuted to lower-diagonal form
dag # this allows us to iteratively work out the covariance matrix
```

```
     i z zx x m cov zy y col zd yd xd
i    1 0  0 0 0   0  0 0   0  0  0  0
z    0 1  0 0 0   0  0 0   0  0  0  0
zx   0 3  1 0 0   0  0 0   0  0  0  0
x    2 0  1 1 0   0  0 0   0  0  0  0
m    0 0  0 1 1   0  0 0   0  0  0  0
cov  0 0  0 0 0   1  0 0   0  0  0  0
zy   0 3  0 0 0   0  1 0   0  0  0  0
y    0 0  0 3 1   2  2 1   0  0  0  0
col  0 0  0 1 0   0  0 1   1  0  0  0
zd   0 2  0 0 0   0  0 0   0  1  0  0
yd   0 0  0 0 0   0  0 1   0  0  1  0
xd   0 0  0 1 0   0  0 0   0  0  0  1
attr(,"class")
[1] "dag"    "matrix" "array"
```

```
#
# coefx(y ~ x, dag)  # with confounding
# coefx(y ~ x + z, dag)  # blocking back-door path
# coefx(y ~ x + zy, dag) # blocking with lower SE
# coefx(y ~ x + zx, dag) # blocking with worse SE
# coefx(y ~ x + zy + cov, dag)  # adding a 'covariate'
# coefx(y ~ x + zy + m, dag)  # including a mediator
# coefx(y ~ x + zy + xd, dag)  # including a descendant
# coefx(y ~ x + zx + i, dag)  # including an instrument
# coefx(y ~ x + zx + i + cov, dag) # I and C
#
# # plotting added-variable plot ellipse
# lines(
#   coefx(y ~ x + zy, mat),
#   lwd = 2, xv= 5,xlim = c(-5,10), ylim = c(-25, 50))
# lines(
#   coefx(y ~ x + zx, mat), new = FALSE,
#   col = 'red', xv = 5, lwd = 2)
# lines(
#   coefx(y ~ x + i, mat), new = FALSE,
```

```
#   col = 'dark green', xv = 5)

# putting results in a data frame
# for easier comparison of SEs

fmlas <- list(
  y ~ x,             # with confounding
  y ~ x + z,         # unconfounded
  y ~ x + zy,        # unconfounded using generating model
  y ~ x + zx,        # uncoufounded using assignment model
  y ~ x + zx + zy,   # 'doubly robust'
  y ~ x + zy + cov,  # adding a 'covariate' unrelated to x
  y ~ x + zy + m,    # adding a 'mediator'
  y ~ x + zy + xd,   # adding a 'descendant of X'
  y ~ x + zy + yd,   # adding a 'descendant of Y'
  y ~ x + zy + col,  # adding a 'collider'
  y ~ x + zx + i,    # adding an instrumental variable
  y ~ x + zx + i + cov,  # adding an instrumental variable and a covariate
  y ~ x + i,         # using an instrumental variable as a control
  y ~ x + zd,        # imperfect control for counfounding
  y ~ x + zd + i     # bias amplification
)
res <- lapply(fmlas, coefx, dag)
res <- lapply(res, function(ll) {
  ll$fmla <- paste(as.character(ll$fmla)[c(2,1,3)], collapse = ' ')
  ll$beta <- ll$beta[1]
  ll
})

df <- do.call(rbind.data.frame, res)
isnum <- sapply(df, is.numeric)
df[,isnum] <- round(df[,isnum], 2)

df[, c(5,1,4,2,3)]
```

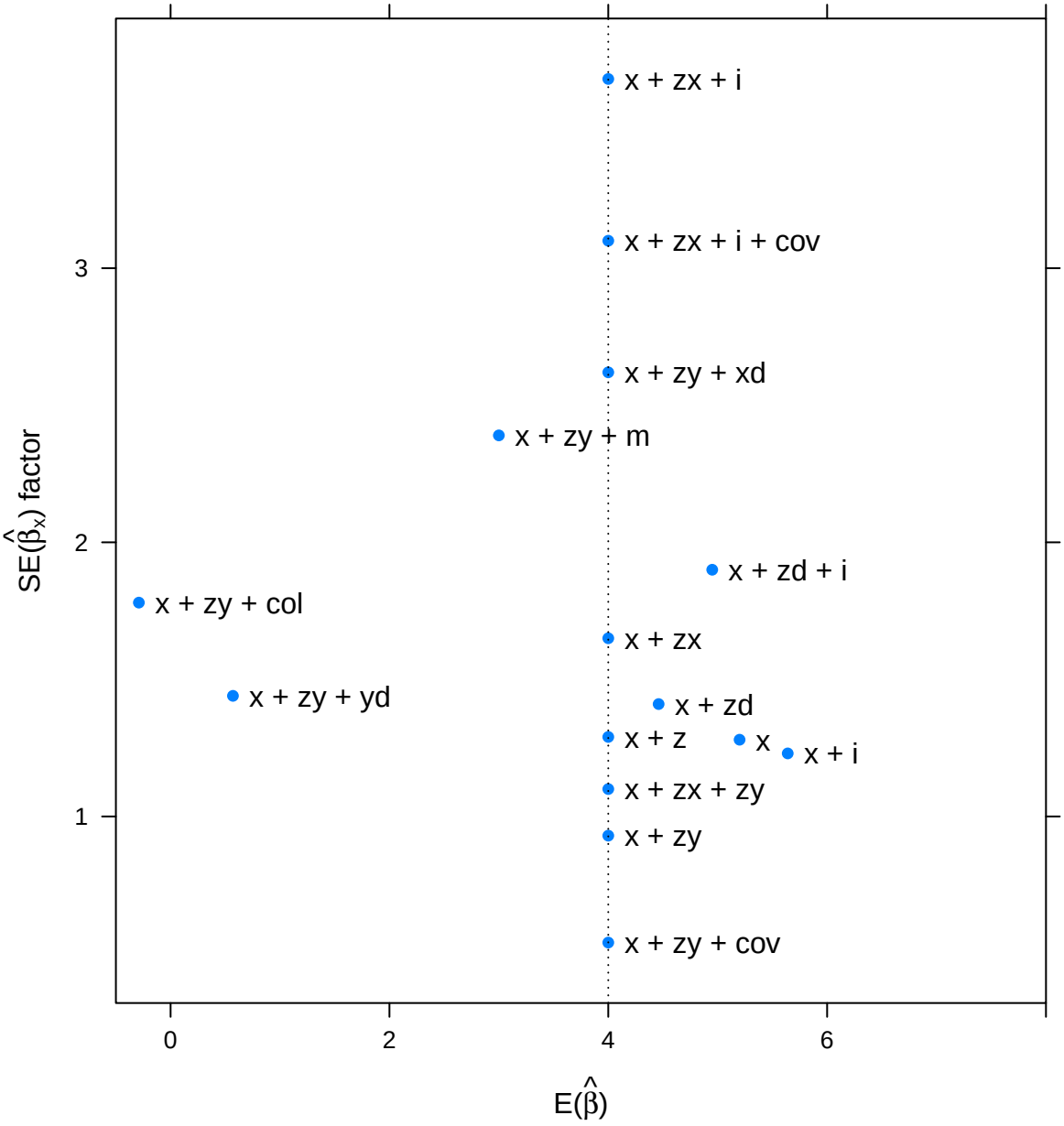
	fmla	beta	sd_betax_factor	sd_e	sd_x_avp
1	y ~ x	5.20	1.28	4.94	3.87
2	y ~ x + z	4.00	1.29	3.16	2.45
3	y ~ x + zy	4.00	0.93	2.45	2.63
4	y ~ x + zx	4.00	1.65	3.69	2.24
5	y ~ x + zx + zy	4.00	1.10	2.45	2.24
6	y ~ x + zy + cov	4.00	0.54	1.41	2.63
7	y ~ x + zy + m	3.00	2.39	2.24	0.93
8	y ~ x + zy + xd	4.00	2.62	2.45	0.93
9	y ~ x + zy + yd	0.57	1.44	0.93	0.64
10	y ~ x + zy + col	-0.29	1.78	0.93	0.52
11	y ~ x + zx + i	4.00	3.69	3.69	1.00
12	y ~ x + zx + i + cov	4.00	3.10	3.10	1.00
13	y ~ x + i	5.64	1.23	4.07	3.32
14	y ~ x + zd	4.46	1.41	3.94	2.79
15	y ~ x + zd + i	4.95	1.90	3.71	1.95

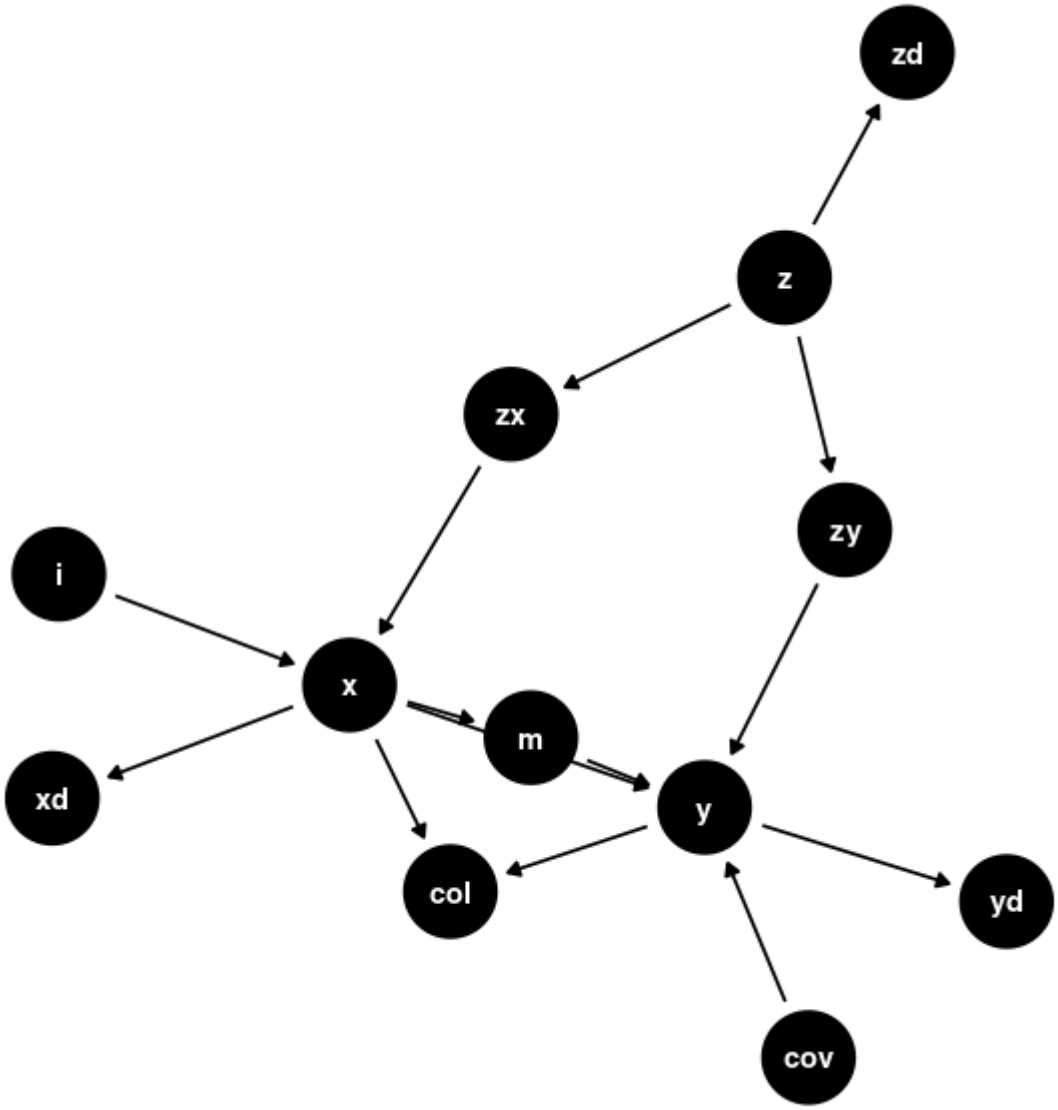
```
library(latticeExtra)
```

```
      Loading required package: lattice
```

```
library(latex2exp)
```

```
xyplot(sd_betax_factor ~ beta, df,
       # scales = list(y = list(log = 10)),
       xlim = c(-.5, 8),
       pch = 16,
       xlab = TeX('$E(\\hat{\\beta})$'),
       ylab = TeX('$SE(\\hat{\\beta}_x)$ factor'),
       labs = sub('y ~ ','  ',as.character(df$fmla))) +
  layer(panel.text(..., labels = labs, adj = 0)) +
  layer(panel.abline(v = 4, lty = 3))
```




zd
z
zx
zy
i
x
m
xd
y
col
yd
cov