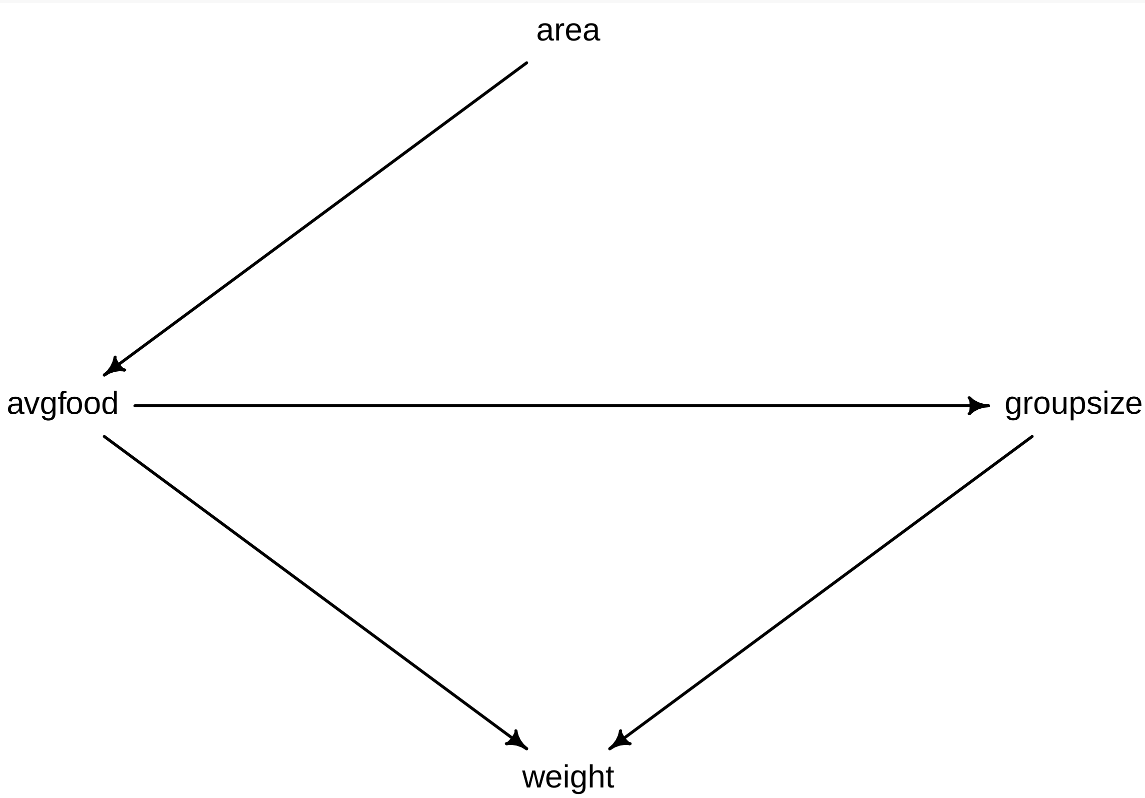
Statistical Rethinking Winter 2020 – Homework Week 3

Alessandro Gentilini `.@gmail` (just auditing)

January 6, 2021

```r
library(rethinking)
```

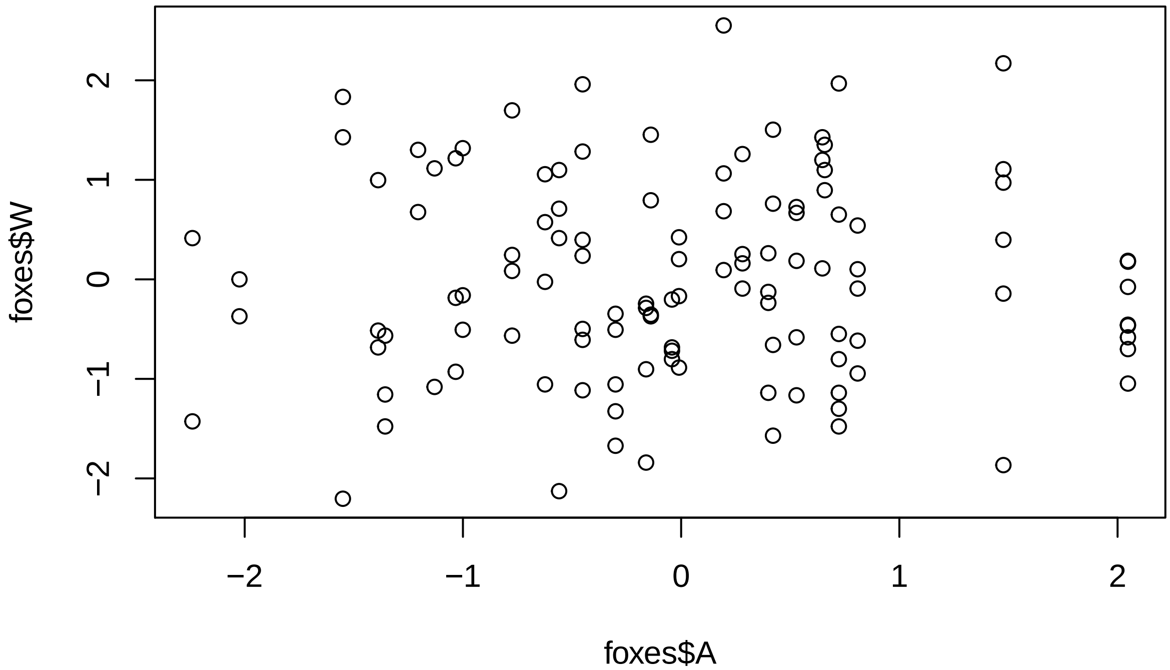
```r
library(dagitty)
dag <- dagitty( "dag {
    area -> avgfood
    avgfood -> groupsize
    avgfood -> weight
    groupsize -> weight
}")
coordinates(dag) <- list(
    x=c(avgfood=-1, groupsize=1, area=0, weight=0),
    y=c(avgfood=0, groupsize=0, area=-1, weight=1)
)
drawdag(dag)
```




```r
adjustmentSets( dag, exposure="area" , outcome="weight" )
```

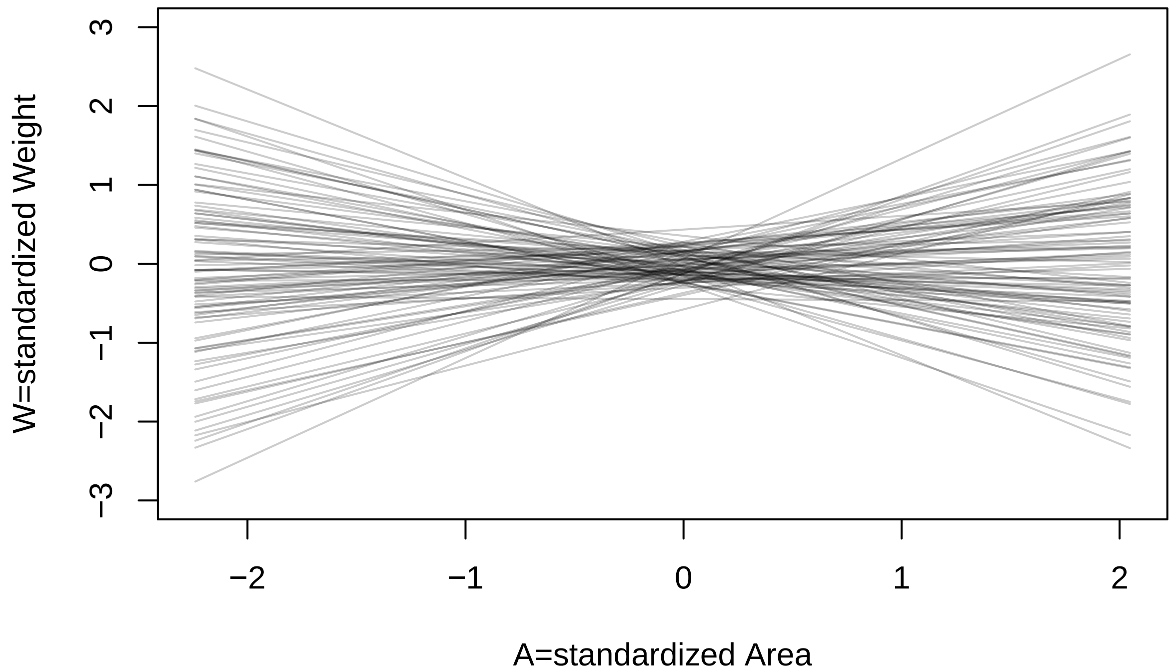
```
## {}
```

```r
data(foxes)
foxes$A <- standardize(foxes$area)
foxes$W <- standardize(foxes$weight)
foxes$F <- standardize(foxes$avgfood)
foxes$G <- standardize(foxes$groupsize)
plot(foxes$A,foxes$W)
```




```r
## R code 4.38
set.seed(2971)
N <- 100                   # 100 lines
a <- rnorm( N , 0 , .2 )
b_A <- rnorm( N , 0 , .5 )

## R code 4.39
plot( NULL , xlim=range(foxes$A) , ylim=c(-3,3) ,
    xlab="A=standardized Area" , ylab="W=standardized Weight" )
#abline( h=0 , lty=2 )
#abline( h=272 , lty=1 , lwd=0.5 )
#mtext( "b ~ dnorm(0,10)" )
#xbar <- mean(d2$weight)
for ( i in 1:N ) curve( a[i] + b_A[i]*x ,
    from=min(foxes$A) , to=max(foxes$A) , add=TRUE ,
    col=col.alpha("black",0.2) )
```




```
m1 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + b_A*A,
        a ~ dnorm( 0 , .2 ),
        b_A ~ dnorm( 0 , .5 ),
        sigma ~ dexp( 1 )
    ), data=foxes )
precis(m1)
```

```
##                mean         sd       5.5%     94.5%
## a     3.020478e-06 0.08361227 -0.1336255 0.1336316
## b_A   1.883613e-02 0.09090041 -0.1264403 0.1641125
## sigma 9.913178e-01 0.06467490  0.8879548 1.0946808
```

```
adjustmentSets( dag, exposure="avgfood" , outcome="weight" )
```

```
##  {}
```

```
foxes$F <- standardize(foxes$avgfood)
m2 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + b_A*A + b_F*F,
        a ~ dnorm( 0 , .2 ),
        b_A ~ dnorm( 0 , .5 ),
        b_F ~ dnorm( 0 , .5 ),
        sigma ~ dexp( 1 )
    ), data=foxes )
precis(m2)
```

```
##                 mean         sd       5.5%     94.5%
## a       2.093550e-06 0.08334407 -0.1331978 0.1332020
## b_A     1.461520e-01 0.17418831 -0.1322346 0.4245385
```

```
## b_F   -1.490596e-01 0.17418846 -0.4274464 0.1293272
## sigma  9.874685e-01 0.06444178  0.8844781 1.0904589
```

```
foxes$F <- standardize(foxes$avgfood)
m3 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + b_F*F,
        a ~ dnorm( 0 , .2 ),
        b_F ~ dnorm( 0 , .5 ),
        sigma ~ dexp( 1 )
    ), data=foxes )
precis(m3)
```

```
##                mean         sd       5.5%     94.5%
## a      4.373265e-06 0.08361896 -0.1336349 0.1336436
## b_F   -2.417790e-02 0.09090895 -0.1694680 0.1211122
## sigma  9.914138e-01 0.06470250  0.8880067 1.0948209
```

I am clueless… it seems to me that `W` is not caused by `F` or `A`. I did not grasped the chapter 6 and so I have no idea on how to proceed. I feel a lack of definitions, for example I did not understand what *conditioning* means.

The worst thing I can do: enumerate all the possibilities and try all the models. There are seven possible linear models:

```
AFG
001
010
011
100
101
110
111
```

The problems with this method are:

1. It's a black-box approach with no insights by myself.
2. Its doesn't scale well with a lot of features.
3. It does not leverage on chapter 6!

```
m001 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + b_G*G,
        a ~ dnorm( 0 , .2 ),
        b_G ~ dnorm( 0 , .5 ),
        sigma ~ dexp( 1 )
    ), data=foxes )
precis(m001)
```

```
##                mean         sd       5.5%       94.5%
## a      8.568912e-07 0.08274463 -0.1322410  0.13224275
## b_G   -1.558071e-01 0.08980043 -0.2993255 -0.01228864
## sigma  9.788930e-01 0.06389595  0.8767750  1.08101109
```

```r
m010 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + b_F*F,
        a ~ dnorm( 0 , .2 ),
        b_F ~ dnorm( 0 , .5 ),
        sigma ~ dexp( 1 )
    ), data=foxes )
precis(m010)
```

```
##                mean         sd       5.5%     94.5%
## a      2.290780e-07 0.08360006 -0.1336088 0.1336093
## b_F   -2.421165e-02 0.09088488 -0.1694632 0.1210399
## sigma  9.911424e-01 0.06465833  0.8878059 1.0944789
```

```r
m011 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + b_F*F+b_G*G,
        a ~ dnorm( 0 , .2 ),
        b_F ~ dnorm( 0 , .5 ),
        b_G ~ dnorm( 0 , .5 ),
        sigma ~ dexp( 1 )
    ), data=foxes )
precis(m011)
```

```
##                mean         sd       5.5%      94.5%
## a     -1.596832e-07 0.08013805 -0.1280762  0.1280759
## b_F    4.772541e-01 0.17912318  0.1909807  0.7635276
## b_G   -5.735267e-01 0.17914168 -0.8598298 -0.2872237
## sigma  9.420437e-01 0.06175252  0.8433513  1.0407362
```

```r
m100 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + b_A*A,
        a ~ dnorm( 0 , .2 ),
        b_A ~ dnorm( 0 , .5 ),
        sigma ~ dexp( 1 )
    ), data=foxes )
precis(m100)
```

```
##               mean         sd       5.5%     94.5%
## a     2.977806e-09 0.08360867 -0.1336228 0.1336228
## b_A   1.883408e-02 0.09089583 -0.1264350 0.1641032
## sigma 9.912661e-01 0.06466649  0.8879166 1.0946157
```

```r
m101 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + b_A*A+b_G*G,
        a ~ dnorm( 0 , .2 ),
        b_A ~ dnorm( 0 , .5 ),
        b_G ~ dnorm( 0 , .5 ),
        sigma ~ dexp( 1 )
    ), data=foxes )
```

```r
precis(m101)
```

```
##                mean         sd       5.5%      94.5%
## a      1.177493e-06 0.08013103 -0.1280637  0.1280660
## b_A    4.058526e-01 0.14536256  0.1735352  0.6381701
## b_G   -4.820001e-01 0.14537258 -0.7143335 -0.2496666
## sigma  9.419453e-01 0.06159405  0.8435061  1.0403845
```

```r
m110 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + b_A*A+b_F*F,
        a ~ dnorm( 0 , .2 ),
        b_A ~ dnorm( 0 , .5 ),
        b_F ~ dnorm( 0 , .5 ),
        sigma ~ dexp( 1 )
    ), data=foxes )
precis(m110)
```

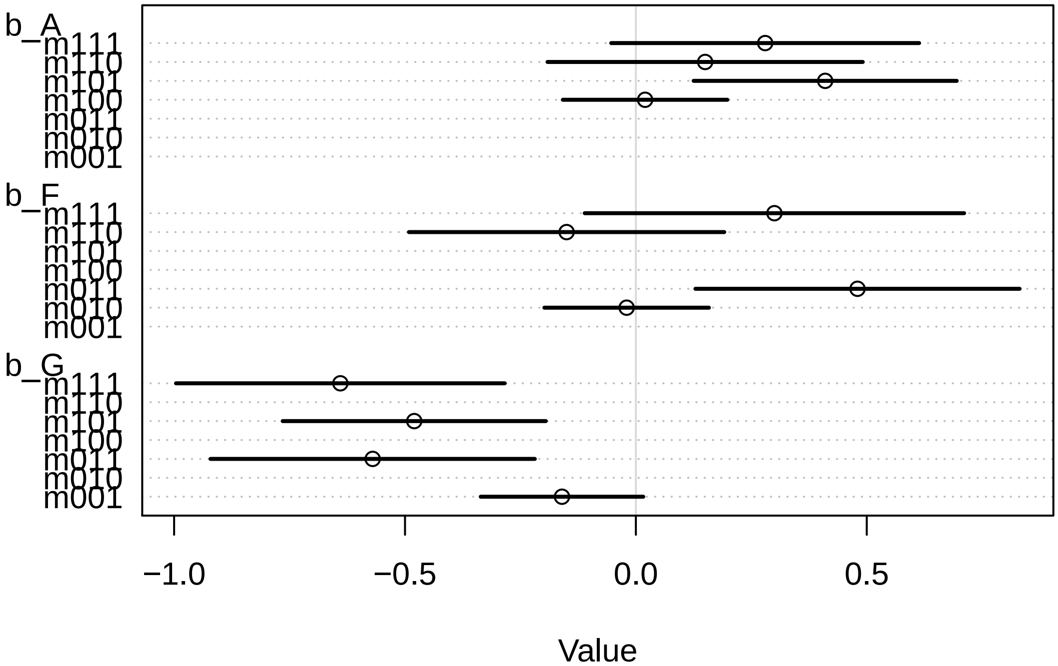
```
##                mean         sd       5.5%     94.5%
## a     -4.411609e-08 0.08334400 -0.1331999 0.1331998
## b_A    1.461357e-01 0.17418822 -0.1322507 0.4245221
## b_F   -1.490376e-01 0.17418838 -0.4274243 0.1293491
## sigma  9.874675e-01 0.06444162  0.8844773 1.0904576
```

```r
m111 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + b_A*A+b_F*F+b_G*G,
        a ~ dnorm( 0 , .2 ),
        b_A ~ dnorm( 0 , .5 ),
        b_F ~ dnorm( 0 , .5 ),
        b_G ~ dnorm( 0 , .5 ),
        sigma ~ dexp( 1 )
    ), data=foxes )
precis(m111)
```

```
##                mean         sd         5.5%      94.5%
## a     -7.064481e-08 0.07936205 -0.126835950  0.1268358
## b_A    2.782378e-01 0.17011235  0.006365425  0.5501102
## b_F    2.968990e-01 0.20960032 -0.038082762  0.6318808
## b_G   -6.396196e-01 0.18161492 -0.929875301 -0.3493639
## sigma  9.312069e-01 0.06100017  0.833716812  1.0286969
```

And now I look at the graphical summary:

```r
plot( coeftab( m001 , m010 , m011, m100, m101, m110, m111 ) , pars=c("b_A","b_F","b_G") )
```



So `b_G` is consistently negative across all the models: smaller group gives weightier foxes. `b_A` is consistently positive accross all the models: larger area gives weightier foxes. But `b_F` seems not so influential on the weight!