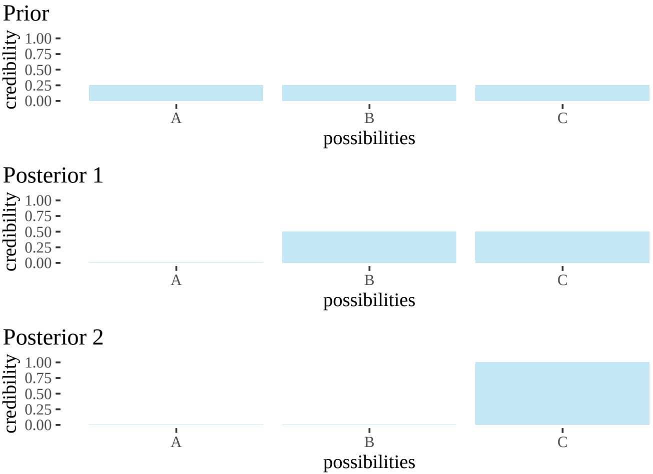
# Revision slides

Rafał Urbaniak, Nikodem Lewandowski
(LoPSE research group, University of Gdansk)

# Sherlock's naivete (l1/s2)

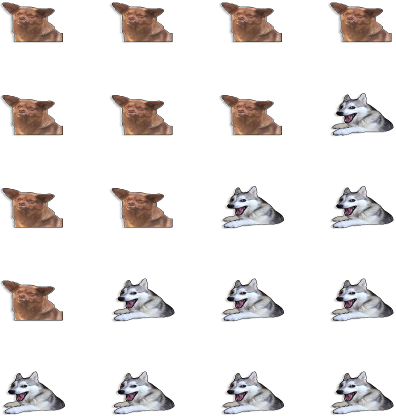
## A rather unhelpful piece of advice

"...when you have eliminated the impossible, whatever remains, however, improbable, must be the truth."



# Updating with new observations (l1/s8)

Ways to observe (h,c,h)



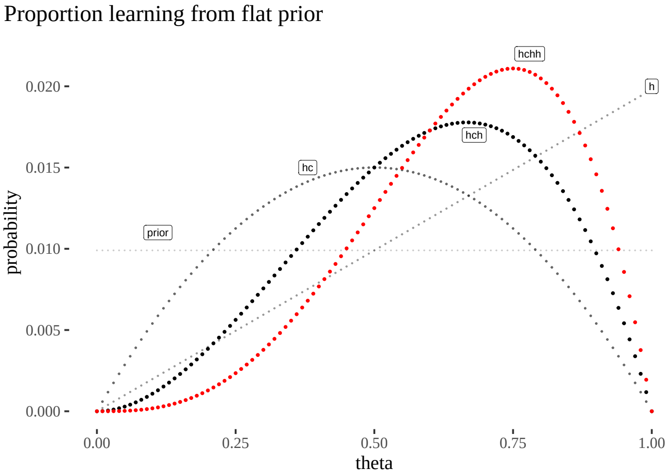
| hypothesis | h | c | h | (h,c,h) | h | (h,c,h,h) |
|---|---|---|---|---|---|---|
| four | 0 | 4 | 0 | 0 | 0 | 0 |
| three | 1 | 3 | 1 | 3 | 1 | 3 |
| two | 2 | 2 | 2 | 8 | 2 | 16 |
| one | 3 | 1 | 3 | 9 | 3 | 27 |
| zero | 4 | 0 | 4 | 0 | 4 | 0 |

# Now with probabilities (l1/s9)

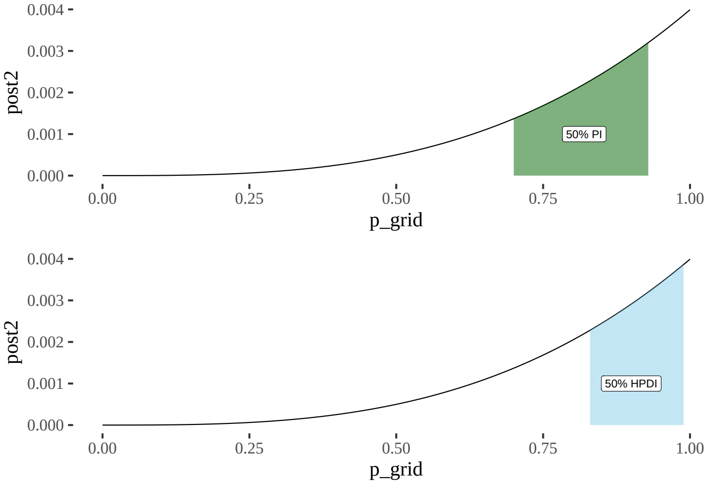
| p | ways0 | ways0pr | ways1 | ways1pr |
|---|---|---|---|---|
| 0.00 | 0 | 0.00 | 0 | 0.0000000 |
| 0.25 | 3 | 0.15 | 3 | 0.0652174 |
| 0.50 | 8 | 0.40 | 16 | 0.3478261 |
| 0.75 | 9 | 0.45 | 27 | 0.5869565 |
| 1.00 | 0 | 0.00 | 0 | 0.0000000 |

# The underlying mechanism (l1/s11)

plausibility(hypothesis $n$|data) $\propto$
ways hypothesis $n$ can produce data × prior plausibility of hypothesis $n$



# PI vs HPDI (l2/s8)

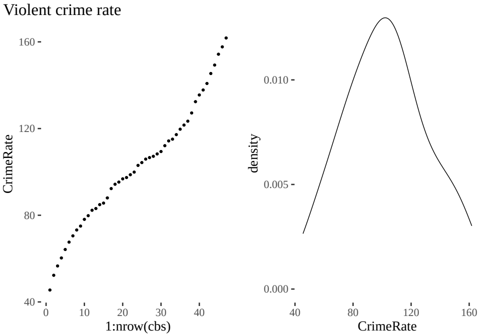


# Crime rates and normal distribution (l2/s16)

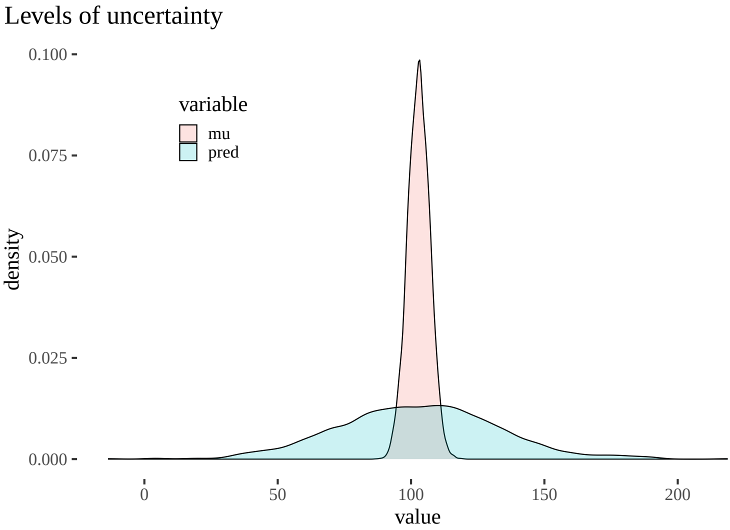
```r
cbs <-  read.csv(file = "../../datasets/CrimeByState.csv")
#these are registered violent incidents per 100k citizens
cbs$CrimeRate
```

```
##  [1]  45.5  52.3  56.6  60.3  64.2  67.6  70.5  73.2  75.0  78.1  79.8  82.3
## [13]  83.1  84.9  85.6  88.0  92.3  94.3  95.3  96.8  97.4  98.7  99.9 103.0
## [25] 104.3 105.9 106.6 107.2 108.3 109.4 112.1 114.3 115.1 117.2 119.7 121.6
## [37] 123.4 127.2 132.4 135.5 137.8 140.8 145.4 149.3 154.3 157.7 161.8
```

```r
cbsPlot <- grid.arrange(ggplot(cbs)+geom_point(aes(x=1:nrow(cbs),y = CrimeRate))+th+
                ggtitle("Violent crime rate"),
ggplot(cbs)+geom_density(aes(x=CrimeRate))+th, ncol=2)
```

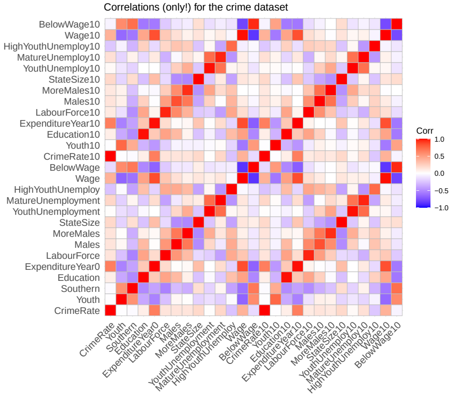


# Levels of uncertainty (l2/s25)

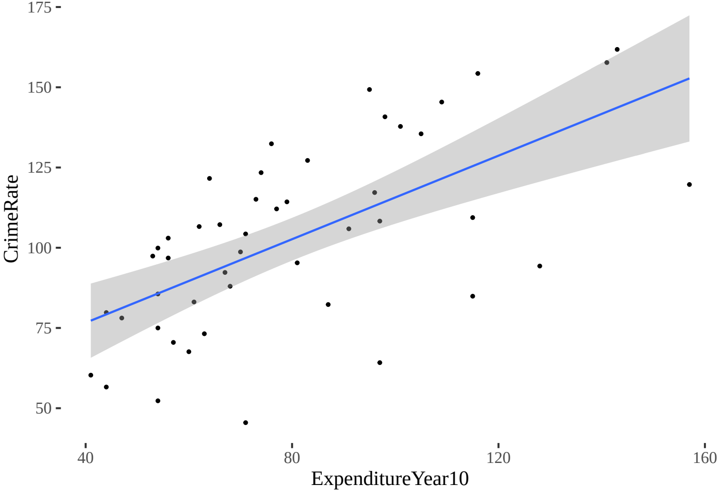


# Predictions vs. Correlations (l3/s2)

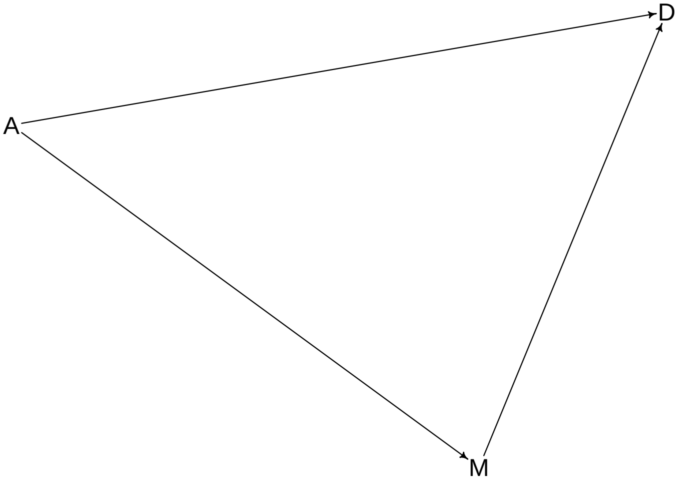
```
#these are registered violent incidents per 100k citizens
cors <- cor(cbs, method = 'spearman')
ggcorrplot(cors, method="square")+
  ggtitle("Correlations (only!) for the crime dataset")
```



# Linear model (l3/s6)

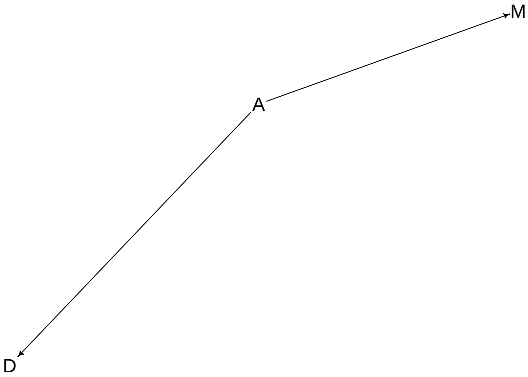


# DAG and divorce rate (l4/s2)



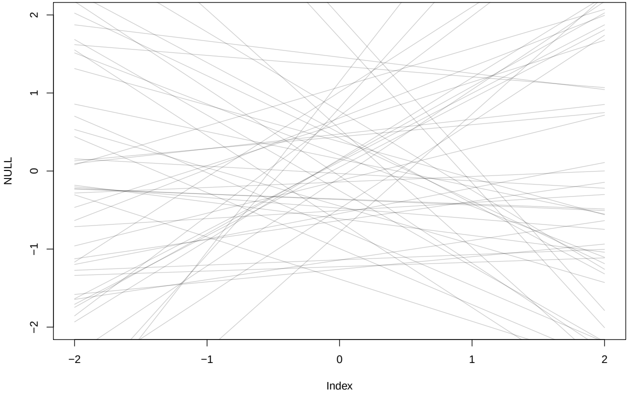
# DAG and divorce rate (l4,s3)

```
dagWaffles2 <- dagitty(
  "dag{
  A -> D; A -> M
  }"
)

drawdag(dagWaffles2, goodarrow = TRUE, cex = 2, radius = 3)
```



# Check your priors! (l4/s23)

```
prior <- extract.prior(milk_try2)
xseq <- seq(-2,2,length.out = 30)
mu <- link(milk_try2, post = prior, data = list(N = xseq))

plot( NULL, xlim = c(-2,2), ylim = c(-2,2))
for (i in 1:50 ) lines (xseq, mu[i,], col = col.alpha("black", .2))
```



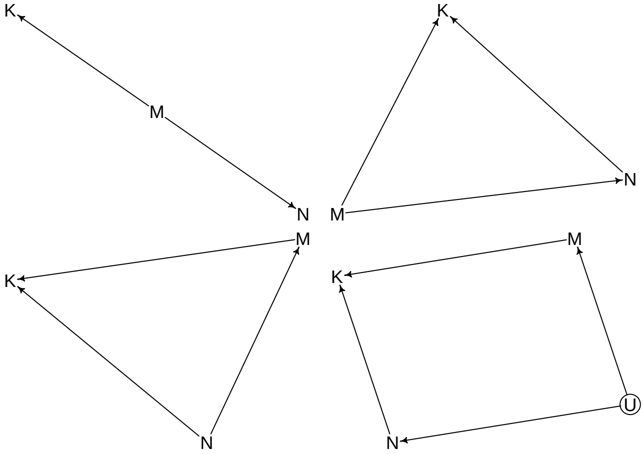
# Now with both predictors (l4/s30)

```r
milk_mn <- quap(
  alist(
    K ~ dnorm( mu, sigma),
    mu <- a + bN * N + bM * M,
    a ~ dnorm(0, .2),
    bM ~ dnorm( 0, .5),
    bN ~ dnorm( 0, .5),
sigma ~ dexp(1)
  ), data = dc
)
```

```
##              mean        sd       5.5%      94.5%
## a      0.06800057 0.1340001 -0.1461574  0.2821585
## bM    -0.70298209 0.2207912 -1.0558492 -0.3501150
## bN     0.67511465 0.2483024  0.2782794  1.0719499
## sigma  0.73802943 0.1324686  0.5263190  0.9497399
```

# Now with DAGs (l4/s35)

```
par(mfrow = c(2, 2))
drawdag(milkDAG1a, cex = 2, radius = 5)
drawdag(milkDAG1, cex = 2, radius = 5)
drawdag(milkDAG2, cex = 2, radius = 5)
drawdag(milkDAG3, cex = 2, radius = 5)
```



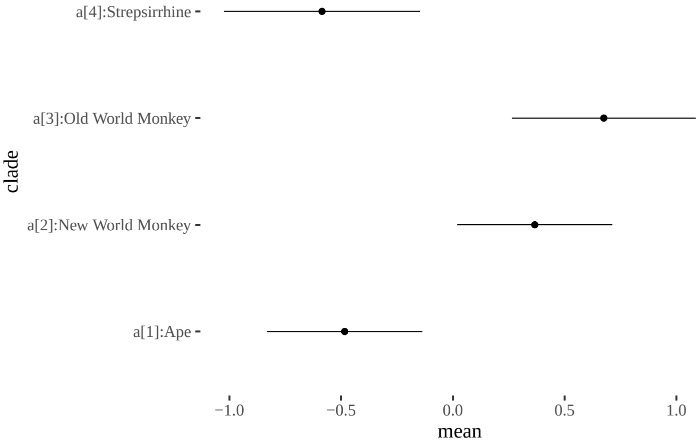
# Proper way of dealing with binary predictors (l4/s40)

```
data(Howell1)
d <- Howell1

d$sex <- ifelse( d$male==1 , 2 , 1 )
str( d$sex )
```

```
##  num [1:544] 2 1 1 2 1 2 1 2 1 2 ...
```

```
heightByGender <- quap(
  alist(
    height ~ dnorm( mu , sigma ) ,
    mu <- a[sex] ,
    a[sex] ~ dnorm( 178 , 20 ) ,
    sigma ~ dunif( 0 , 50 )
  ) , data=d )

heightByGenderWrong <- quap(
  alist(
    height ~ dnorm( mu , sigma ) ,
    mu <- a + b * male ,
    a ~ dnorm( 178 , 20 ) ,
    b ~ dnorm( 0 , 10 ) ,
    sigma ~ dunif( 0 , 50 )
  ) , data=d )
```

# Multiple predictors (l4/s47)



# Selection-distortion effect (l5/s4)

```
N <- 800 #proposals/candidates
p <- .5  #proportion to select
# uncorrelated newsworthiness/
#looks and trustworthiness/kindness
nwl <- rnorm(N)
twk <- rnorm(N)

s <- nwl + twk # total score
q <- quantile( s , 1-p ) # top 10% threshold
selected <- ifelse( s >= q , TRUE , FALSE )
cor( twk[selected] , nwl[selected] )
```

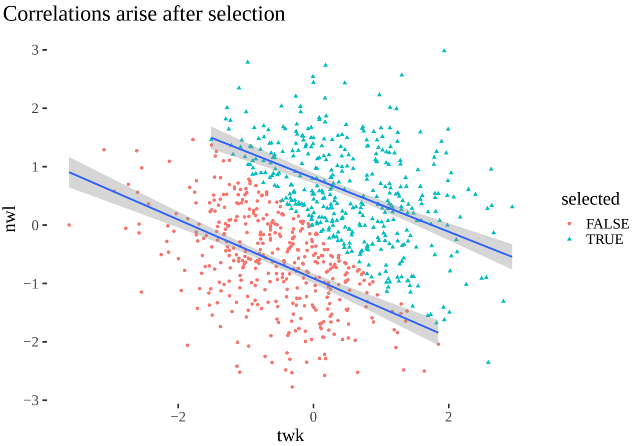
```
## [1] -0.4649455
```

```
cor( twk[!selected] , nwl[!selected] )
```

```
## [1] -0.5135988
```

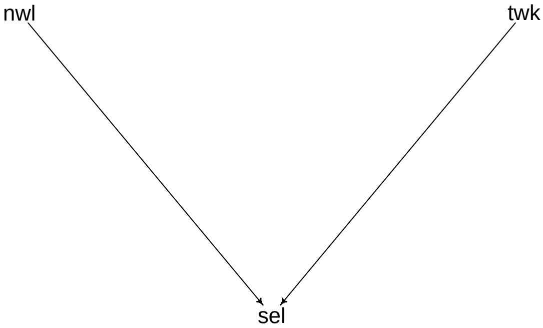
# Selection-distortion effect (l5/s5)

```
ggplot() + geom_point(aes(
  x = twk, y = nwl, color = selected, shape = selected))+
geom_smooth(aes(
  x = twk, y = nwl, group = selected), method = "lm")+th+
  ggtitle("Correlations arise after selection")
```



# Collider bias (l5/s8)

```
newsDAG <- dagitty (
  "dag{
  nwl -> sel <- twk
  }"
)
coordinates(newsDAG) <- list(
  x=c(nwl=0,sel=1,twk=2) , y=c(nwl=0,sel=1,twk=0) )
drawdag(newsDAG, cex = 2,
        radius = 3, goodarrow = TRUE, xlim = c(-.2,2.2), ylim = c(-1.2,.2))
```



# Post-treatment bias (l5/s9)

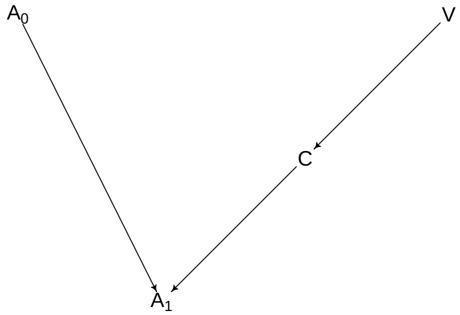
Blindly tossing in predictors is never a good idea

```r
set.seed(21)
# number of subjects
N <- 100
# simulate initial aggression levels
aggression0 <- rnorm(N,1,.4)
#simulate vaccine
vaccine <- rep( 0:1 , each=N/2)
#simulate fungus
cordyceps <- rbinom( N , size=1 , prob=0.95 - vaccine * 0.5 )
# assign vaccines and simulate cordyceps and aggression
aggression1 <- aggression0 + rnorm(N, 1.2 + 2 *cordyceps, .4)
# compose a clean data frame
d <- data.frame( aggression0=aggression0 , aggression1=aggression1 ,
                 vaccine=vaccine , cordyceps=cordyceps )
precis(d)[,-5]
```

```
##                 mean        sd      5.5%    94.5%
## aggression0 1.028994 0.4108600 0.3452403 1.702803
## aggression1 3.703373 1.0376606 1.8749605 5.038115
## vaccine     0.500000 0.5025189 0.0000000 1.000000
## cordyceps   0.740000 0.4408440 0.0000000 1.000000
```

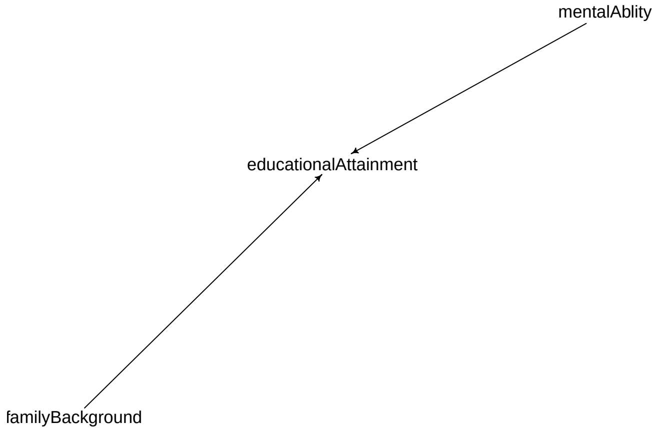
# C d-separates V from A1 (l5/s14)

```
aggressionDAG <- dagitty( "dag {
A_0 -> A_1
C -> A_1
V -> C
}")
coordinates( aggressionDAG ) <- list( x=c(A_0=0,V=1.5,C=1,A_1=.5) ,
                                  y=c(A_0=0,V=0,C=.5,A_1=1) )
drawdag( aggressionDAG,  cex = 2, radius = 3, goodarrow = TRUE,
         xlim = c(-.3,1.7), ylim = c(-1.2,.2))
```



# Causality creeps (l5/s17)

## Example: status attainment tradition

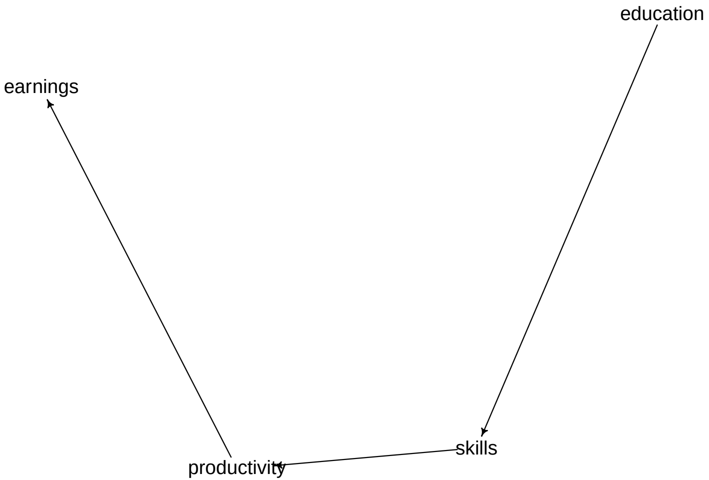


## Implicit Wisconsin model

Students follow their own aspirations.

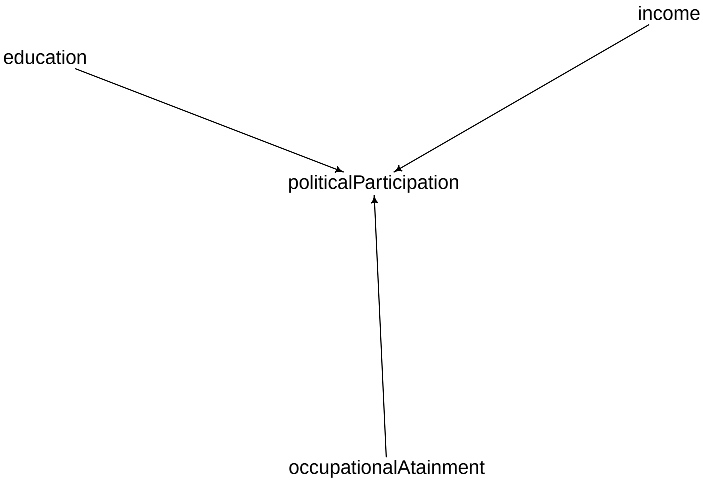
# Causality creeps (l5/s19)

## Example: economic theory of human capital



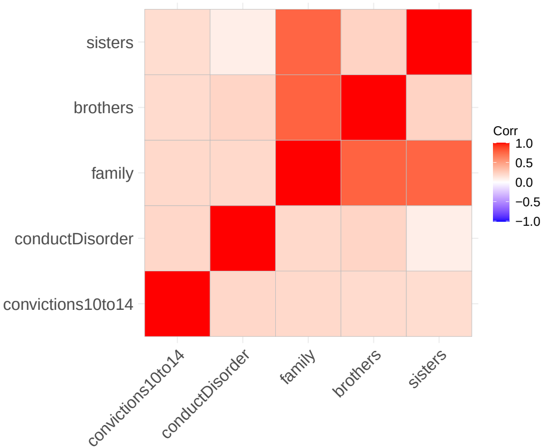
# Causality creeps (l5/s21)

## Example: political participation



# Multicolinearity (l6/s3)

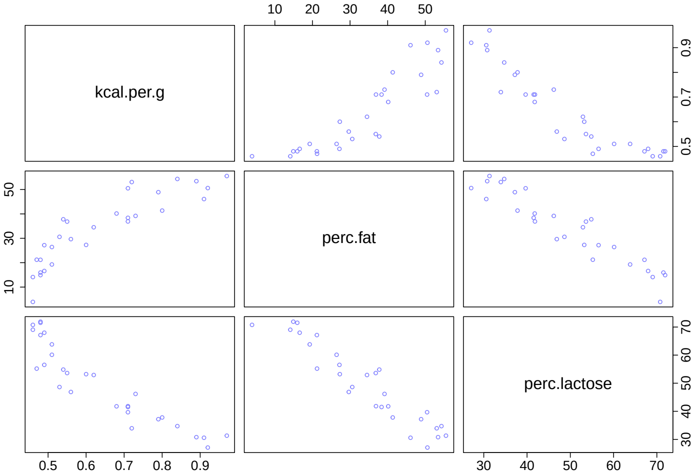
```
cors <- cor(small, method = "spearman")
ggcorrplot(cors) + corSize
```



Let's ignore it for simplicity now (bad practice in general)

# Multicolinearity and milk (l6/s12)

```
pairs( ~ kcal.per.g + perc.fat + perc.lactose,
       data=d , col=rangi2, cex.axis = 1.6)
```



# Confounding (l6/s16)

## The notion

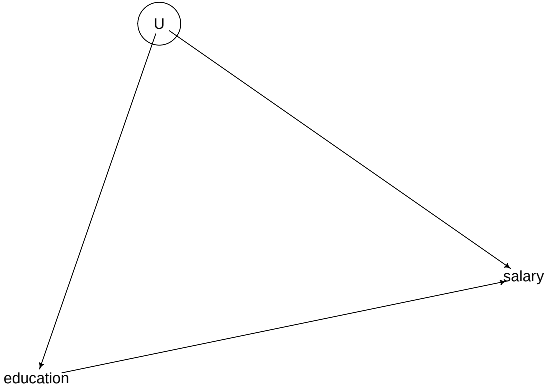
Context in which the association between an outcome and a predictor is not the same as it would be had we experimentally intervened on the predictor.

## But when?

- Sometimes, because we didn't condition on a variable.
- Sometimes, because we **did** condition on a variable, too!

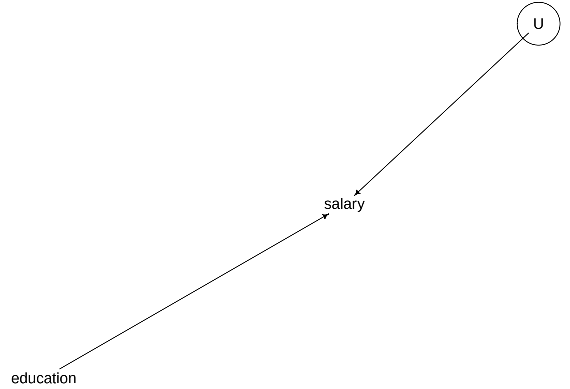
# Confounding (l6/s17)

## Non-causal paths

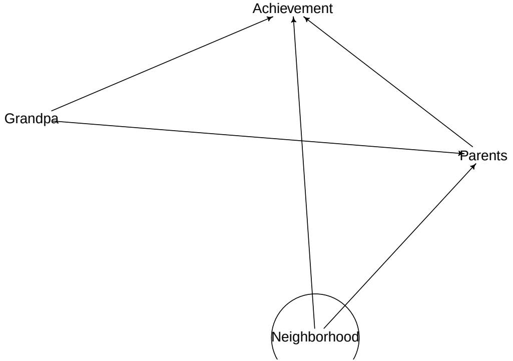


# Confounding (l6/s18)

## Experimenting

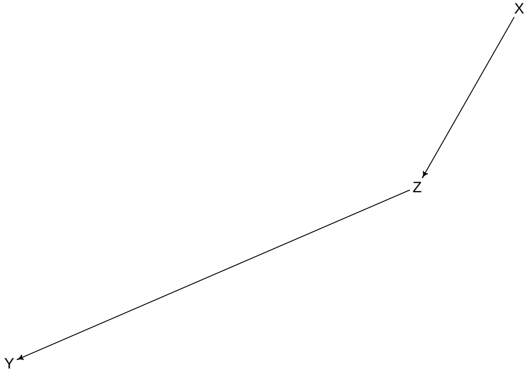


# DAG haunting (l6/s24)



# Shut the backdoor (l6/s29)

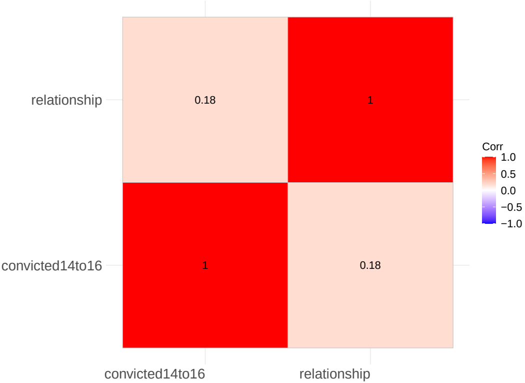
## The pipe/chain



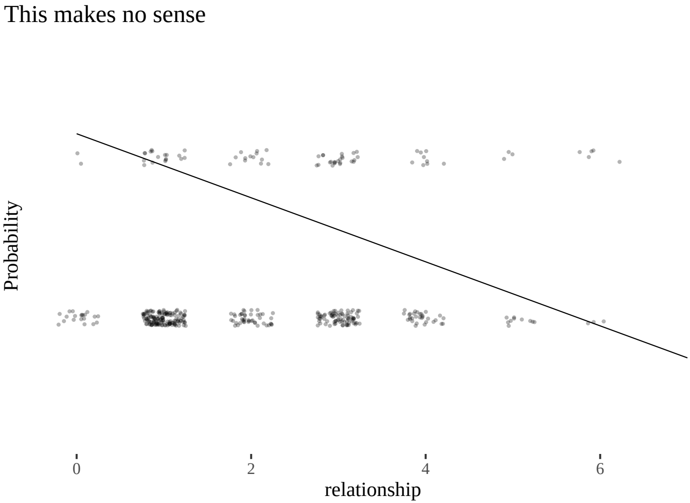
$I(X, Y|Z)$ (think fungi)

# Binary outcomes (l7/s3)

```
# 677 no, casual, steady, engaged,
#married, cohabiting 6:  convicted 14-16
data <- as.data.frame(read_xpt("crimeLife.xpt"))
small <- data[,c(6, 677)]

names(small) <- c("convicted14to16", "relationship")

cors <- cor(small, method = "spearman")
ggcorrplot(cors, lab= TRUE, lab_size = 5, tl.srt = 0) + corSize
```



# Why we need link functions (l7/s8)



## An oversimplification?

Throw cohabiting below engaged, treat as numeric. Never do at home!

# What are link functions anyway? (l7/s10)

## In general

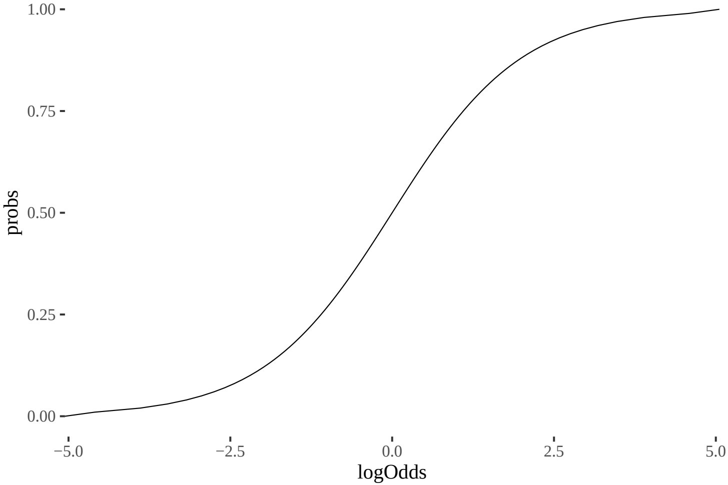
$$
\begin{aligned}
y_i &\sim \mathrm{Blah}(\theta_i, \phi) \\
f(\theta) &= \alpha + \beta(x_i - \bar{x})
\end{aligned}
$$

## Logit link

$$
\begin{aligned}
y_i &\sim \mathrm{Binomial}(n, p_i) \\
\mathrm{logit}(p_i) &= \alpha + \beta(x_i - \bar{x}) \\
\mathrm{logit}(p_i) &= \log\left(\frac{p_i}{1 - p_i}\right) \\
\log\left(\frac{p_i}{1 - p_i}\right) &= \alpha + \beta(x_i - \bar{x}) \\
p_i &= \frac{\exp(\alpha + \beta x_i)}{1 + \exp(\alpha + \beta x_i)}
\end{aligned}
$$

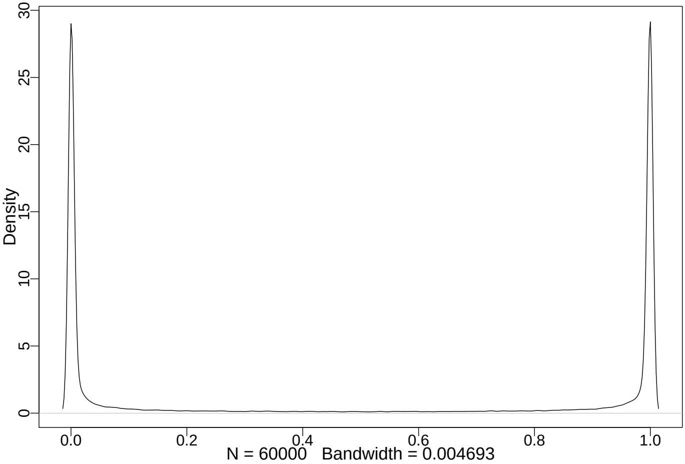
# Logit link (l7/s13)

```
ggplot()+geom_line(aes(y = probs, x = logOdds))+th
```



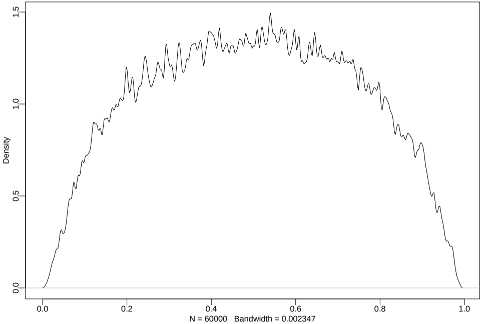
# Check your priors! (l7/s15)

```
prior <- extract.prior( crimeFactorial , n=1e4 )

p <- sapply( 1:6 , function(k) inv_logit( prior$a + prior$b[,k] ) )

dens( p , adj=0.1, cex.axis=1.3, cex.lab=1.5 )
```



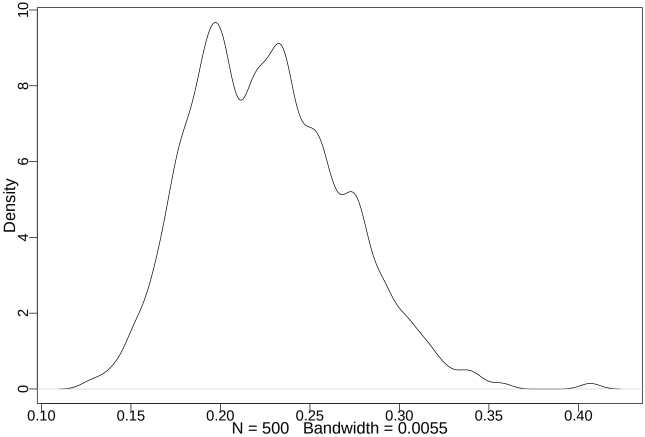
# Check your priors! (l7/s16)

```
crimeFactorialNarrow <- ulam(
  alist(
    conv ~ dbinom( 1 , p ) ,
    logit(p) <- a +  b[relFactor] ,
    a ~ dnorm( 0, 1.1),
    b[relFactor] ~ dnorm( 0 , .5 )
  ) , data=data, log_lik = TRUE )

priorN <- extract.prior( crimeFactorialNarrow , n=1e4 )

pN <- sapply( 1:6 , function(k) inv_logit( priorN$a + priorN$b[,k] ) )

dens( pN, adj=0.1 )
```

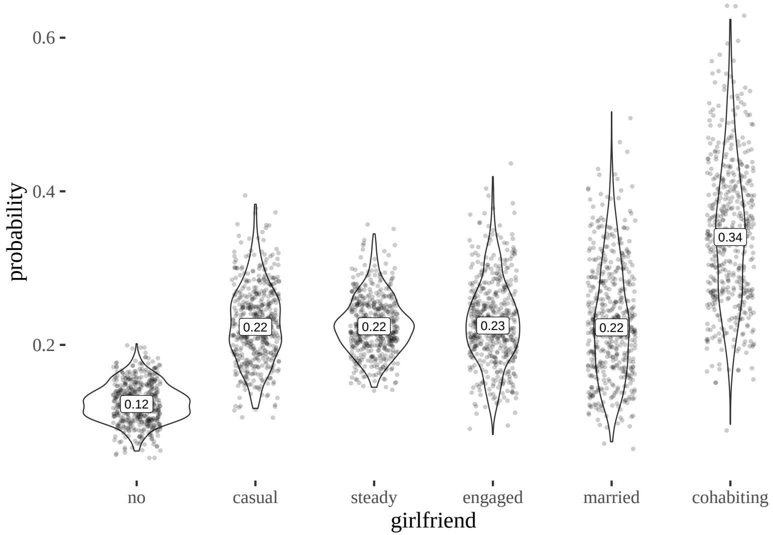


# Now the posteriors (l7/s18)

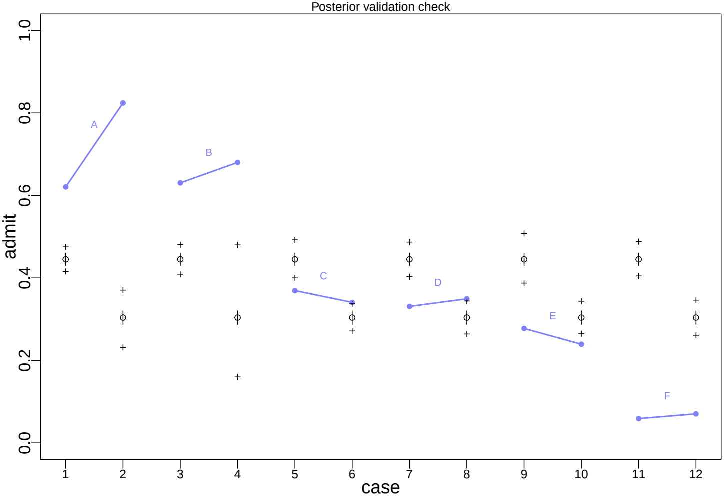
```
post <- extract.samples(crimeFactorialNarrow)

baseline <- inv_logit(post$a)

dens (baseline, cex.axis=1.3, cex.lab=1.5)
```



# Now the posteriors (l7/s20)

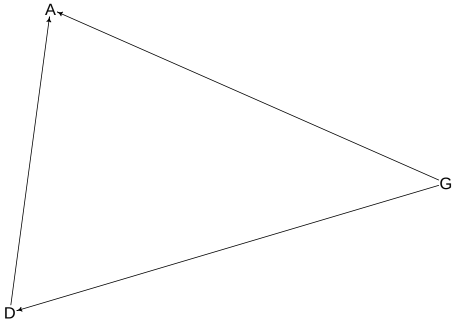


# UC Berkeley admissions (l7/s31)



# Within departments (l7/s36)

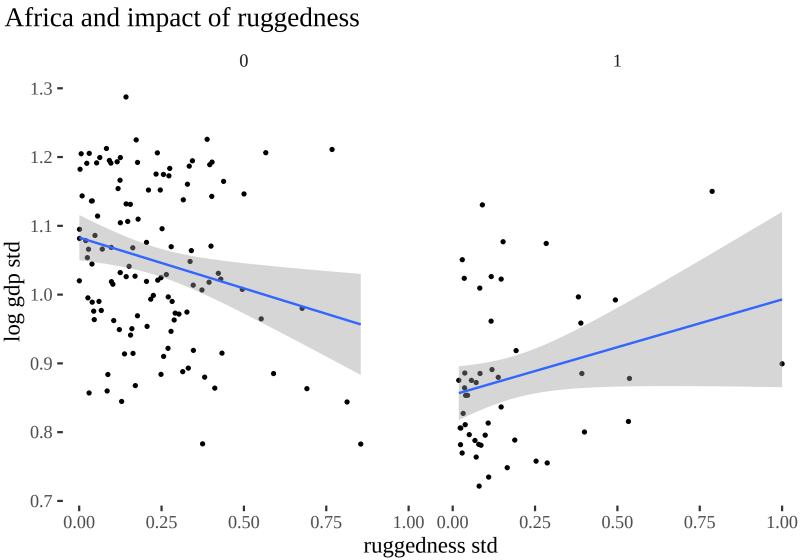
```
ucbDAG <- dagitty(
  "dag{
  G -> D; G -> A; D -> A
  }"
)
drawdag(ucbDAG, goodarrow = TRUE, cex = 2, radius = 3)
```



```
adjustmentSets(ucbDAG, exposure = "G",
               outcome = "A", effect = "direct")
```

```
## { D }
```

# African economy and bad geography (l9/s3)



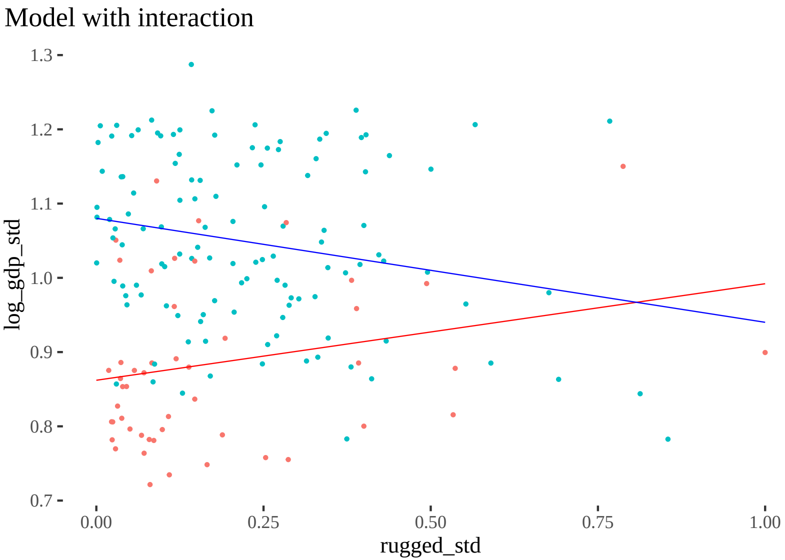
# African economy and bad geography (l9/s6)

```
africaNoInteraction <- quap(
  alist(
    log_gdp_std ~ dnorm( mu , sigma ) ,
    mu <- a + b*( rugged_std - 0.215 ) ,
    a ~ dnorm( 1 , .1 ) ,
    b ~ dnorm( 0 , .3 ) ,
    sigma ~ dexp( 1 )
  ) , data=dd )
```

# African economy and bad geography (l9/s12)

```
dd$cid <- ifelse( dd$cont_africa==1 , 1 , 2 )

africaInteraction <- quap(
  alist(
    log_gdp_std ~ dnorm( mu , sigma ) ,
    mu <- a[cid] + b[cid]*( rugged_std - 0.215 ) ,
    a[cid] ~ dnorm( 1 , 0.1 ) ,
    b[cid] ~ dnorm( 0 , 0.3 ) ,
    sigma ~ dexp( 1 )
  ) , data=dd )
```

# African economy and bad geography (l9/s14)



# Tulips (l9/s18)

```
tulipsInteraction <- quap(
  alist(
    blooms_std ~ dnorm( mu , sigma ) ,
    mu <- a + bw*water_cent + bs*shade_cent +
      bws*water_cent*shade_cent ,
    a ~ dnorm( 0.5 , 0.25 ) ,
    bw ~ dnorm( 0 , 0.25 ) ,
    bs ~ dnorm( 0 , 0.25 ) ,
    bws ~ dnorm( 0 , 0.25 ) ,
    sigma ~ dexp( 1 )
  ) , data=d )
```