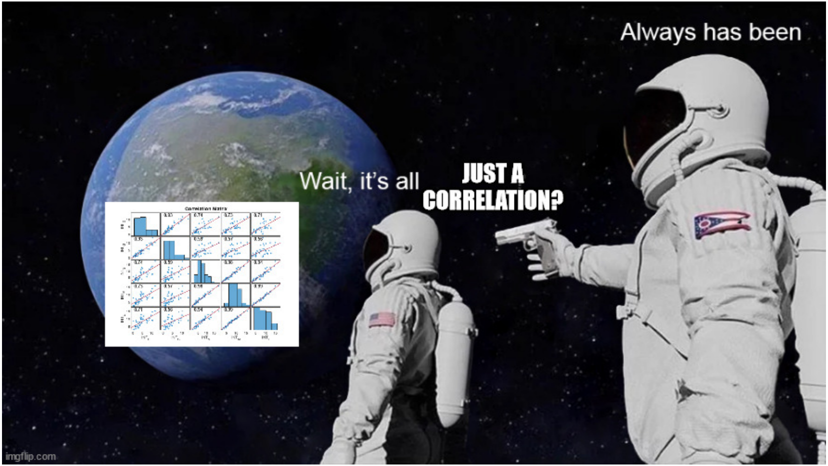
The Stuff Called Causality

Nikodem Lewandowski

University of Gdańsk

Classical Statistic's Realization



Bayesian approach to Causality

- Classical statistics emphasizes correlation over causation, treating relationships between variables as associations rather than causal links.
- Bayesian models offer a framework for explicitly modeling causal relationships, allowing for the incorporation of prior knowledge and beliefs about causality.
  - With that we can try to model causality in a **small world**
  - It is not trivial that the small world causal link will be true in a **big world**
- While Bayesian models facilitate causal inference, establishing causality in complex systems remains challenging and context-dependent.

WaffleDivorce dataset

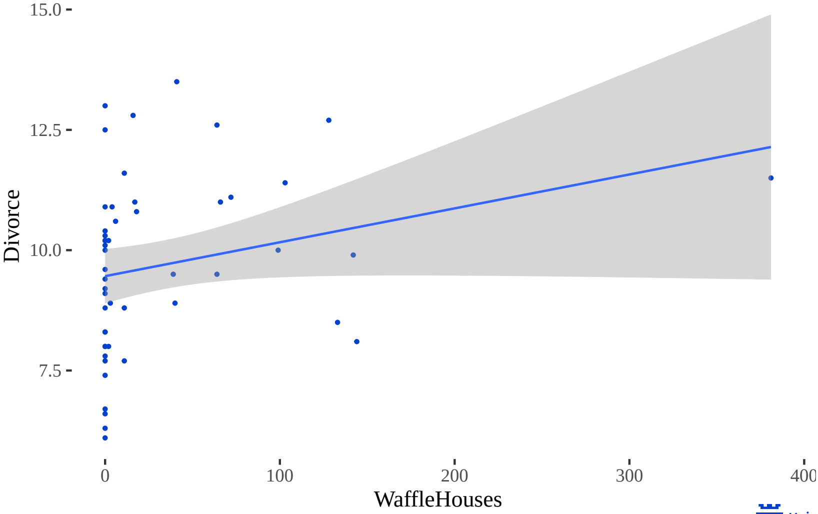
```r
data("WaffleDivorce")
d <- WaffleDivorce

head(d, n=4)
```

```
  Location Loc Population MedianAgeMarriage Marriage Marriage.SE Divorce
1  Alabama  AL       4.78              25.3     20.2        1.27    12.7
2   Alaska  AK       0.71              25.2     26.0        2.93    12.5
3  Arizona  AZ       6.33              25.8     20.3        0.98    10.8
4 Arkansas  AR       2.92              24.3     26.4        1.70    13.5
  Divorce.SE WaffleHouses South Slaves1860 Population1860 PropSlaves1860
1       0.79          128     1     435080         964201           0.45
2       2.05            0     0          0              0           0.00
3       0.74           18     0          0              0           0.00
4       1.22           41     1     111115         435450           0.26
```

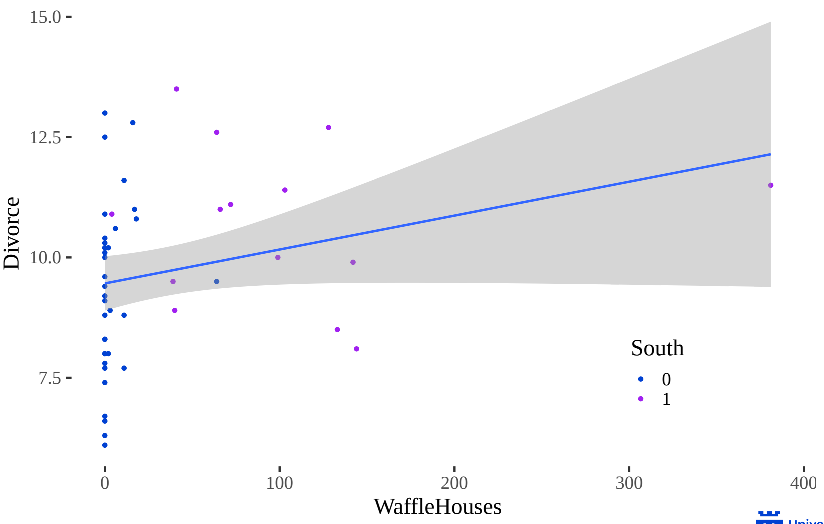
```r
print(paste('The Dataset length:', nrow(d)))
```

```
[1] "The Dataset length: 50"
```

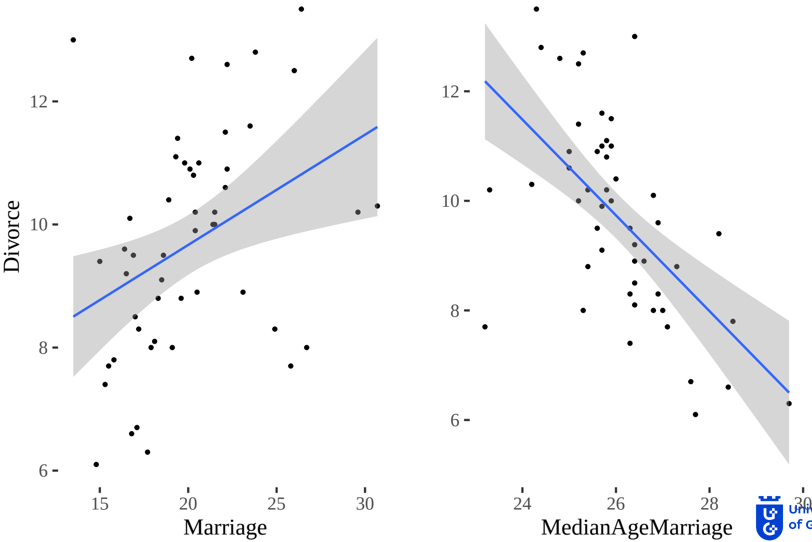
Waffle Houses vs. Divorce Rate?



Waffle Houses vs. Divorce Rate?



Confounding: first stab



University of Gdańsk

Let's First Standardize

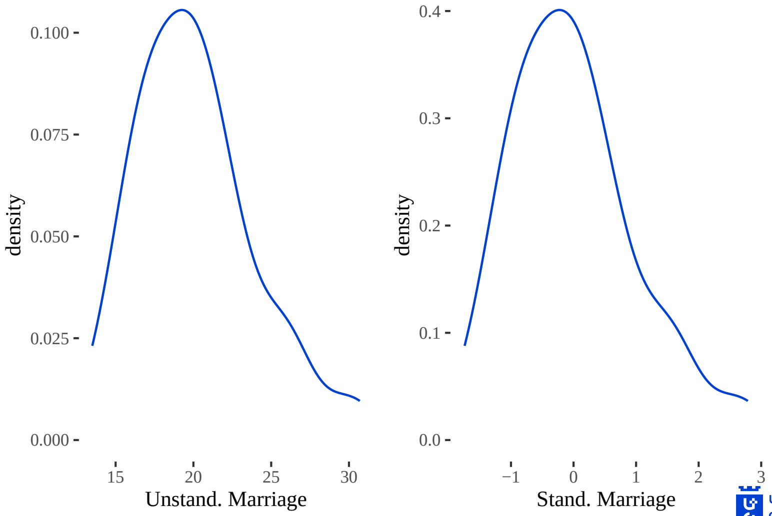
```r
marriage_stnd1 <- standardize(d$Marriage)
marriage_stnd2 <- (d$Marriage - mean(d$Marriage)) / sd(d$Marriage)

as.vector(marriage_stnd1)[1:5] == marriage_stnd2[1:5]
```

```
[1] TRUE TRUE TRUE TRUE TRUE
```

$$z = \frac{x - \mu}{\sigma}$$

- We standardize to:
  - enable intuitive comparison of different variables
  - improve performance of some models
  - mitigate the influence of outliers

Stand. and Unstand. Comparison



University of Gdańsk

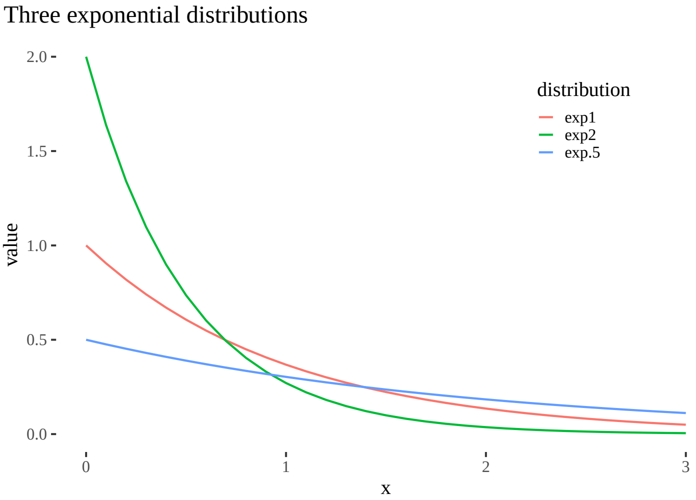
Divorce vs. MedianAgeMarriage

```
d$D <- standardize(d$Divorce)
d$M <- standardize(d$Marriage)
d$A <- standardize(d$MedianAgeMarriage)

ageModelWide <- quap(
  alist(
    D ~ dnorm(mu, sigma ) ,
    mu <- a + bA * A ,
    a ~ dnorm(0, 1),
    bA ~ dnorm( 0, 1),
    sigma ~ dexp( 1 )
  ), data = d
)
```

University of Gdańsk

Exponential distribution

$$f(x, rate) = rate \times e^{-rate \times x}$$



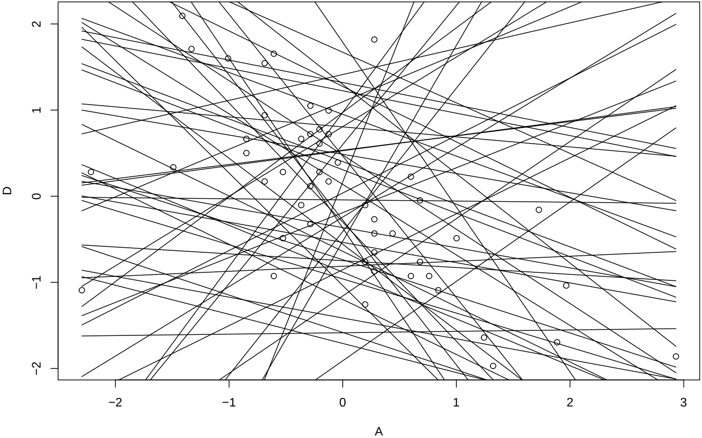
Prior Predictive Check

```r
prior <- as.data.frame(extract.prior(ageModelWide, n = 50))
head(prior, n = 2)
```

```
           a        bA    sigma
1 -0.3010122 -1.382887 2.179705
2 -0.2760757 -1.682699 0.833474
```

Prior Predictive Check

```
plot ( D ~  A, data = d)
for ( i in 1:50) {
  curve( prior$a[i] + prior$b[i] * x, add = TRUE)}
```



Revising our Priors

```
ageModelNarrow <- quap(
  alist(
    D ~ dnorm(mu, sigma ) ,
    mu <- a + bA * A ,
    a ~ dnorm(0, .5),
    bA ~ dnorm( 0, .5),
    sigma ~ dexp( .5 )
  ), data = d
)
```

- Notice the smaller values for the priors in this version of the model

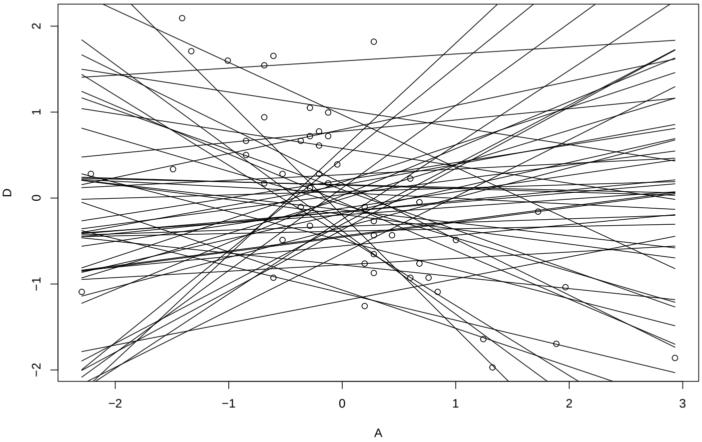
Prior predictive check V.2

```r
priorNarrow <- as.data.frame(extract.prior(ageModelNarrow, n = 50))
head(prior, n = 2)
```

```
           a        bA    sigma
1 -0.3010122 -1.382887 2.179705
2 -0.2760757 -1.682699 0.833474
```

Prior predictive check V.2

```
plot ( D ~  A, data = d)
for ( i in 1:50) {
  curve( priorNarrow$a[i] + priorNarrow$b[i] * x, add = TRUE)
}
```



Posterior predictive check

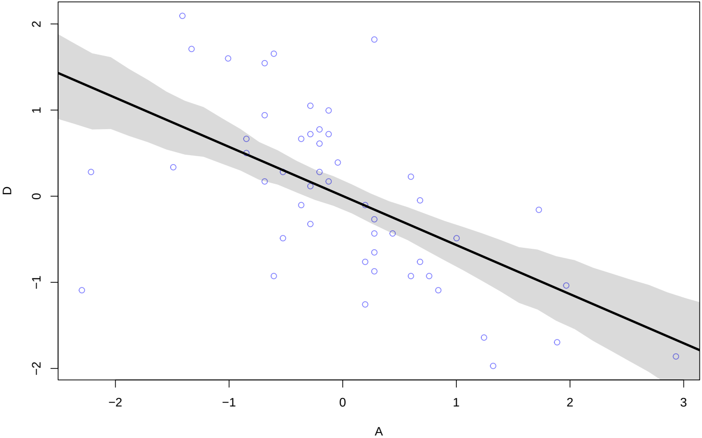
```
# creating a range of values of a length 50
A_range <- seq(-4,4, length.out = 50)

# link function for simulating mu values for 50 cases
mu <- link(ageModelNarrow, data  = list(A = A_range))
str(mu)
```

```
 num [1:1000, 1:50] 2.41 2.7 2.78 1.57 1.98 ...
```

```
mu_mean <- apply(mu, 2, mean)
mu_hpdi <- apply(mu, 2, HPDI)
```

Posterior predictive check

```
plot(D ~ A, data = d, col = rangi2)
lines(A_range, mu_mean, lwd = 3)
shade(mu_hpdi, A_range)
```



Posterior predictions

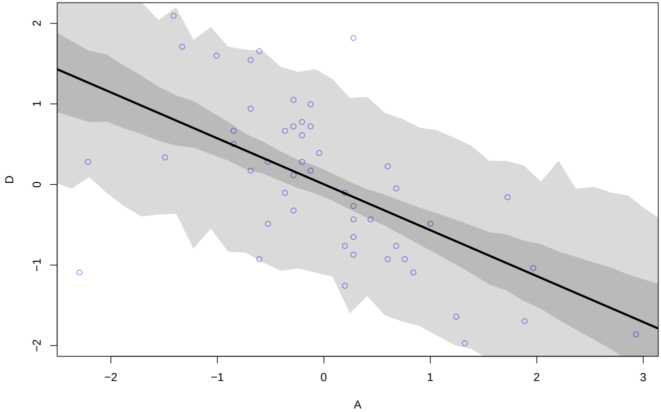
```r
# sim function for predicting values of a parameter in question
pred <- sim(ageModelNarrow, data  = list(A = A_range))
str(pred)
```

```
 num [1:1000, 1:50] 1.98 2.4 2.32 2.68 2.47 ...
```

```r
pred_hpdi <- apply(pred, 2, HPDI)
```

Posterior predictions

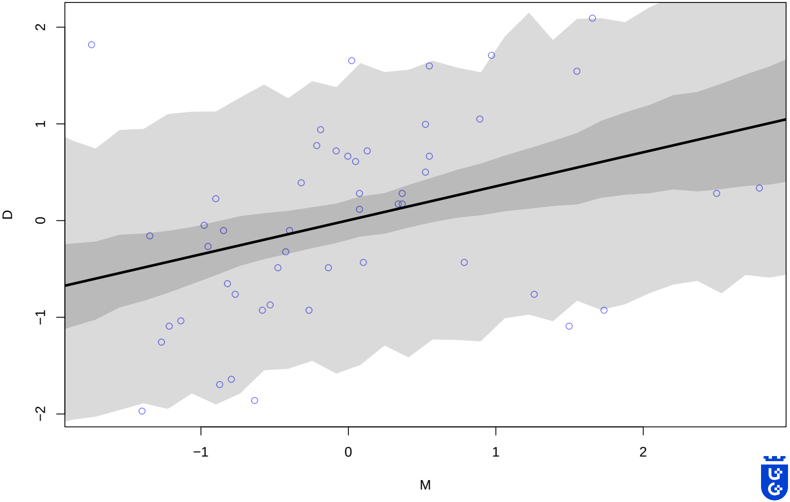
```
plot(D ~ A, data = d, col = rangi2)
lines(A_range, mu_mean, lwd = 3)
shade(mu_hpdi, A_range)
shade(pred_hpdi, A_range)
```



Now just marriage rate

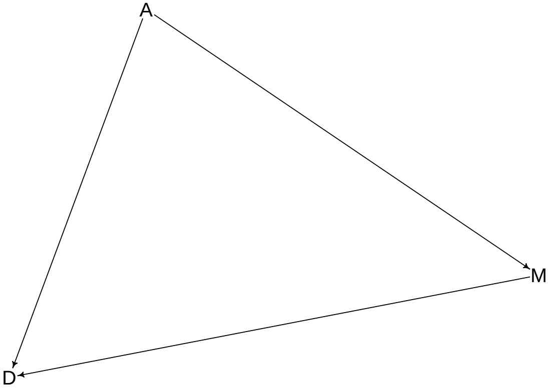
```r
marriageModelNarrow <- quap(
  alist(
    D ~ dnorm(mu, sigma ) ,
    mu <- m + bM * M ,
    m ~ dnorm(0, .5),
    bM ~ dnorm( 0, .5),
    sigma ~ dexp( .5 )
  ), data = d
)
```

Now just marriage rate

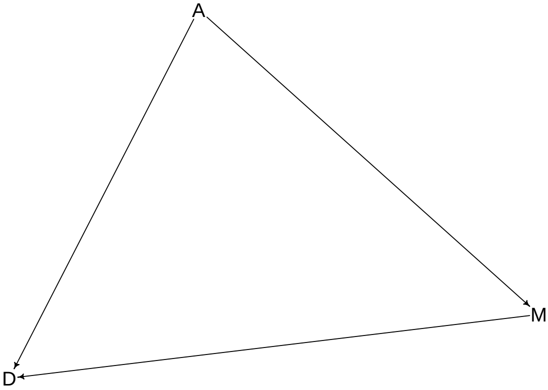


DAGs to the rescue!

```
dagWaffles1 <- dagitty(
  "dag{
  A -> D; A -> M; M -> D
  }"
)

drawdag(dagWaffles1, goodarrow = TRUE, cex = 2, radius = 3)
```



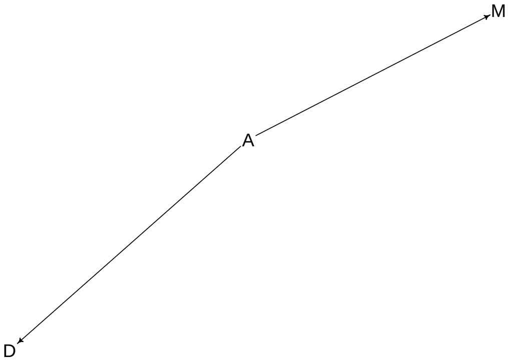
DAGs to the rescue!



- notice two causal paths from A to D
- regressing on either A or M tells us the total “influence”
- On this model, the path from M to D is not causal!

A Fork?

```
dagWaffles2 <- dagitty(
  "dag{
  A -> D; A -> M
  }" )
drawdag(dagWaffles2, goodarrow = TRUE, cex = 2, radius = 3)
```



- We can imply independency relation of D and M when we condition on A:

$$I(D, M)|A$$

How to Figure out Independencies?

- There is a function for that!
- First one doesn't give an output as there are no independencies to claim

```
impliedConditionalIndependencies(dagWaffles1)
impliedConditionalIndependencies(dagWaffles2)
```

```
D _||_ M | A
```