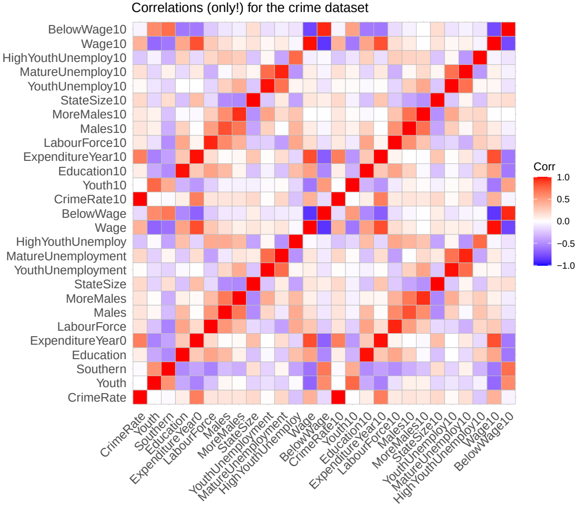
Linear Models

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

Predictions vs. Correlations

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



Correlation

Correlation is a statistical measure that indicates the extent to which two variables are related. In other words, it shows how strong the relationship is between two variables.

Spearman's rank correlation coefficient

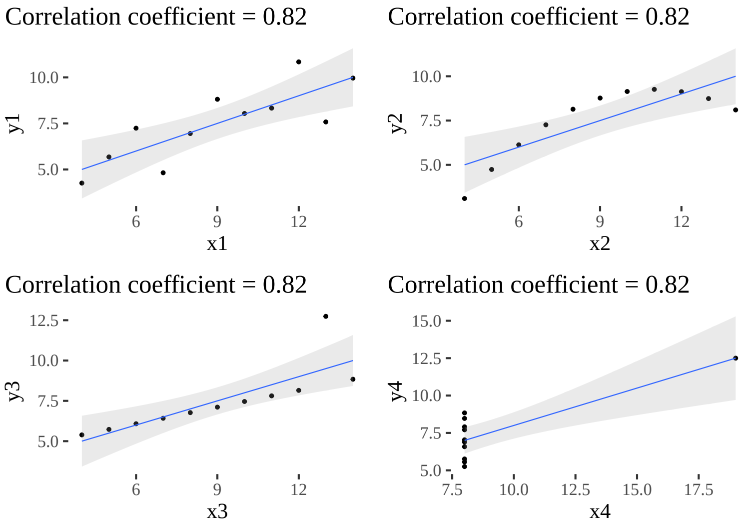
d_i is the difference in paired ranks and n is number of cases.

$$r_s = 1 - \frac{6 \sum d^2}{n(n^2 - 1)}$$

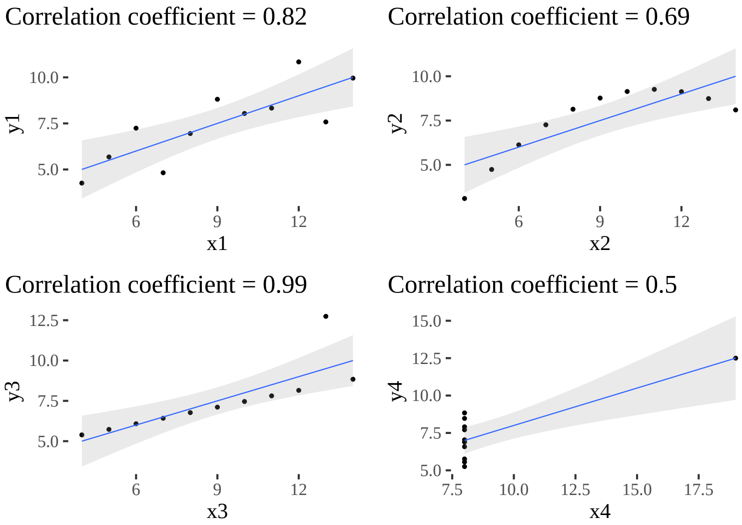
Pearson's correlation coefficient:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{(y_i - \bar{y})^2}}$$

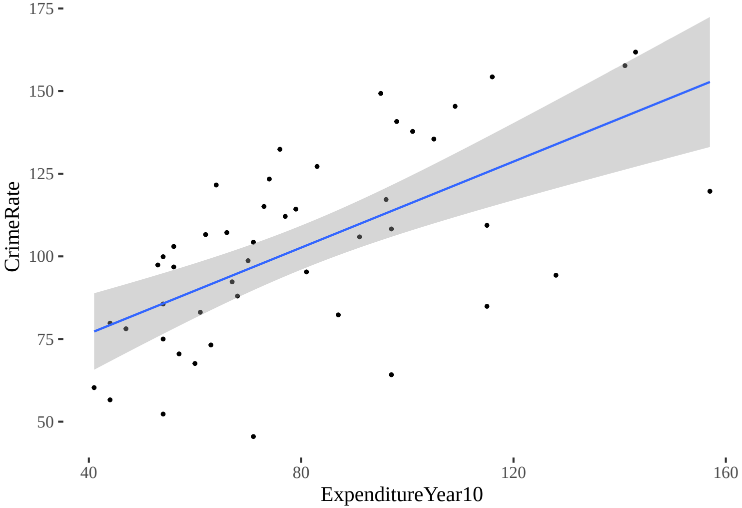
Anscombe's quartet (Pearson's correlation)



Anscombe's quartet (Spearman's correlation)

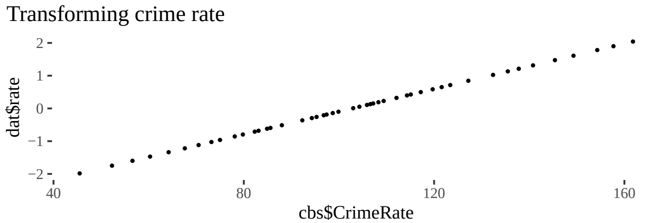


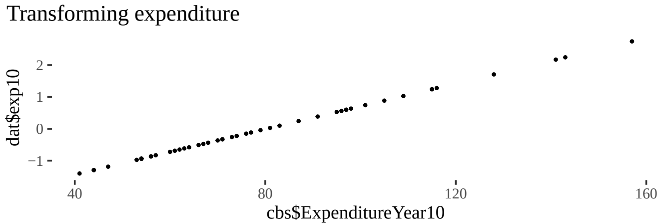
Linear model



Linear model

```
dat <- list(
  rate = (cbs$CrimeRate - mean(cbs$CrimeRate))/
    sd(cbs$CrimeRate),
  exp10 = standardize(cbs$ExpenditureYear10))
```





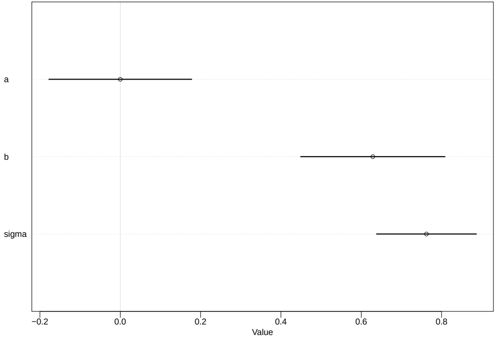
Linear model

```
expenditureModel <- quap(
  alist(
    rate ~ dnorm( mu , sigma ) ,
    mu <- a + b * exp10,
    a ~ dnorm( 0 ,  3) ,
    b ~ dnorm( 0 , 3 ) ,
    sigma ~ dexp(1 )
  ), data = dat
)

precis(expenditureModel)
```

```
##               mean         sd       5.5%     94.5%
## a     1.555453e-05 0.11112402 -0.1775821 0.1776132
## b     6.288045e-01 0.11232387  0.4492893 0.8083197
## sigma 7.623511e-01 0.07768734  0.6381917 0.8865105
```

Linear model

```
plot(precis(expenditureModel))
```




```
sd(cbs$ExpenditureYear10)
```

```
## [1] 27.96132
```

```
c(.45, .63, .81) * sd(cbs$CrimeRate)
```

```
## [1] 13.00197 18.20276 23.40355
```

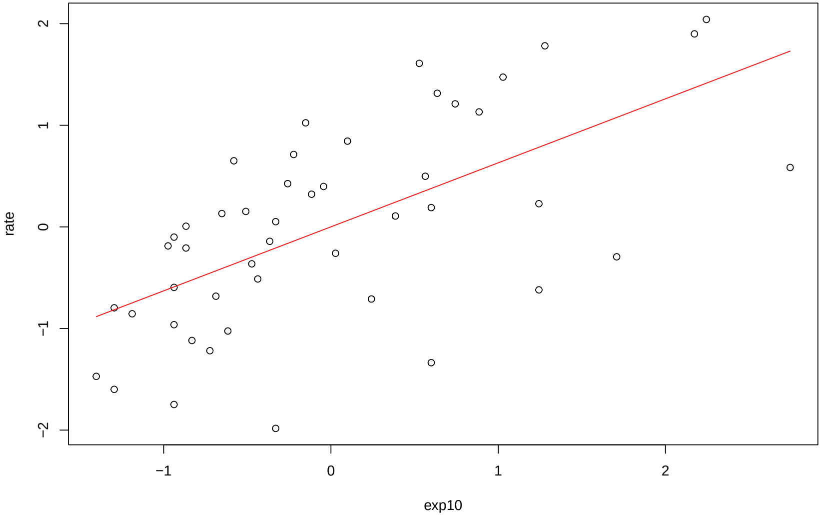
Uncertainty about predicted means

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

head(post, n = 4)

a_map <- mean(post$a)
b_map <- mean(post$b)
c(a_map, b_map)

x = dat$exp10
plot ( rate ~  exp10, data = dat)
curve( a_map + b_map * x , add = TRUE, col = "red")
```

```
##             a         b     sigma
## 1  0.06371085 0.5202144 0.6999054
## 2  0.01687718 0.5799770 0.9360311
## 3 -0.09687310 0.7266234 0.6381939
## 4 -0.02995279 0.4264947 0.7966586
```

```
## [1] 0.0008306219 0.6299220794
```

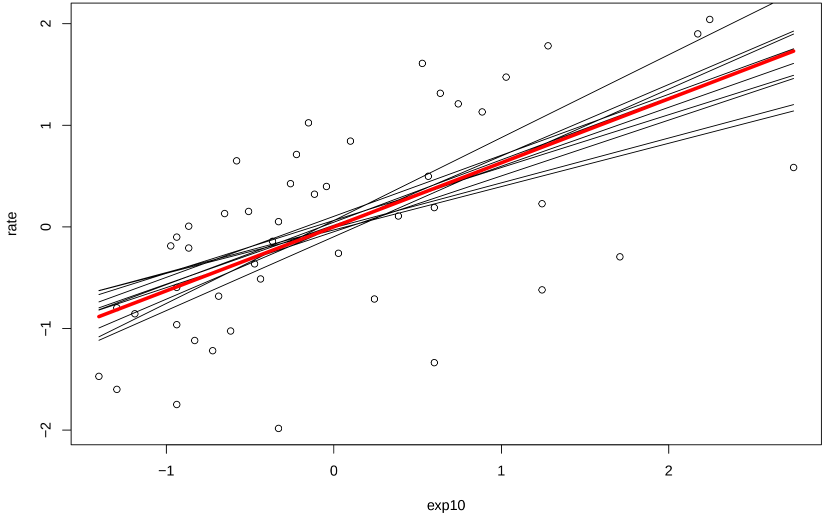
Uncertainty about predicted means



Uncertainty about predicted means

```r
x = dat$exp10
post10 <- extract.samples(expenditureModel, n = 10)
plot ( rate ~  exp10, data = dat)
for ( i in 1:10) {
  curve( post$a[i] + post$b[i] * x, add = TRUE)
}
curve( a_map + b_map * x , add = TRUE, col = "red", lwd = 4)
```

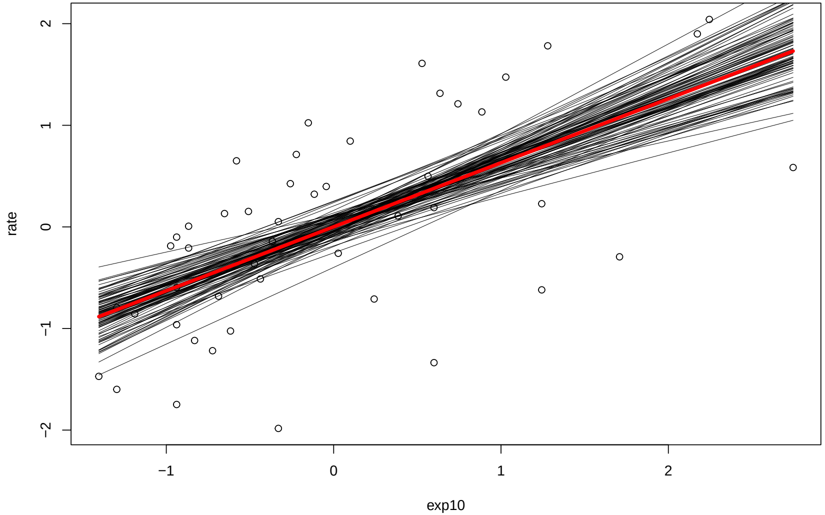
Uncertainty about predicted means



Uncertainty about predicted means

```r
x = dat$exp10
post100 <- extract.samples(expenditureModel, n = 100)
plot ( rate ~  exp10, data = dat)
for ( i in 1:100) {
  curve( post100$a[i] + post100$b[i] * x, add = TRUE, lwd = .01)
}
curve( a_map + b_map * x , add = TRUE, col = "red", lwd = 4)
```

Uncertainty about predicted means

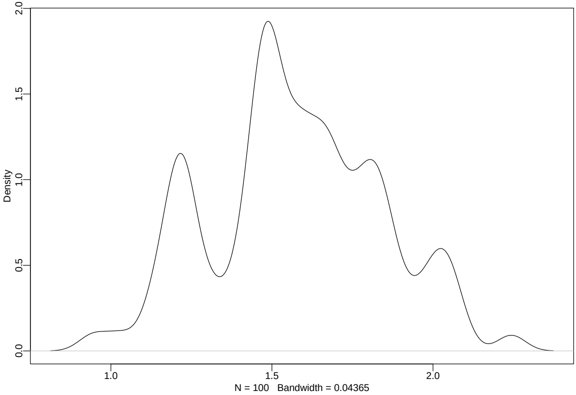


Uncertainty about predicted means

```
pred_mean_100_2.5 <- post100$a + post100$b * 2.5
HPDI(pred_mean_100_2.5)
```

```
##     |0.89    0.89|
## 1.193394 2.036221
```

```
dens (pred_mean_100_2.5)
```

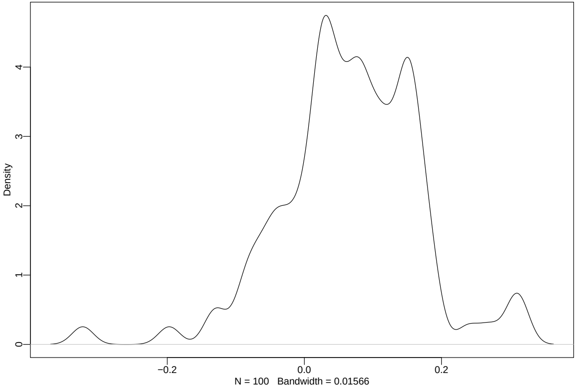


Uncertainty about predicted means

```
pred_mean_100_.1 <- post100$a + post100$b * 0.1
HPDI(pred_mean_100_.1)
```

```
##       |0.89       0.89|
## -0.09210891  0.18177355
```

```
dens (pred_mean_100_.1)
```



Uncertainty about predicted means

```
exp_seq <- seq(-2,3,by = .1)
mu <- link(expenditureModel, data = data.frame(exp10 = exp_seq))
str(mu)
```

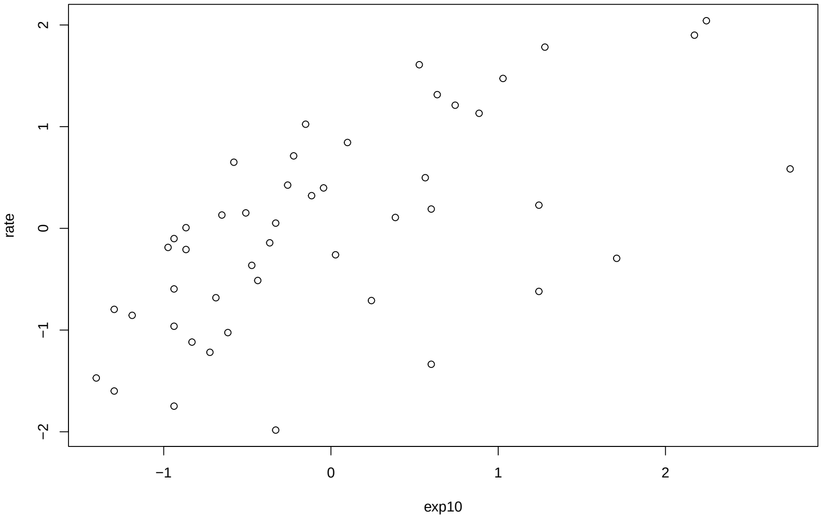
```
##  num [1:1000, 1:51] -1.311 -1.502 -0.889 -1.107 -1.097 ...
```

```
mu[1:5,1:5]
```

```
##            [,1]      [,2]       [,3]       [,4]       [,5]
## [1,] -1.3114914 -1.238302 -1.1651130 -1.0919238 -1.0187346
## [2,] -1.5015673 -1.431772 -1.3619775 -1.2921826 -1.2223877
## [3,] -0.8887324 -0.839011 -0.7892896 -0.7395682 -0.6898468
## [4,] -1.1065074 -1.051116 -0.9957245 -0.9403331 -0.8849416
## [5,] -1.0973350 -1.041725 -0.9861154 -0.9305056 -0.8748959
```

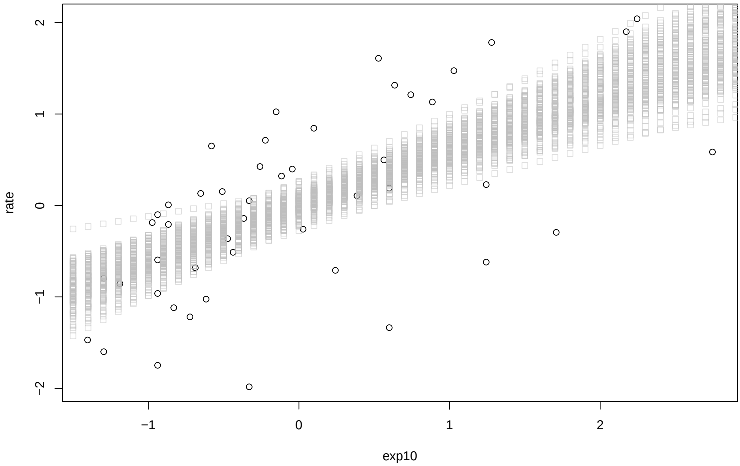
Uncertainty about predicted means

```
plot ( rate ~  exp10, data = dat)
```



Uncertainty about predicted means

```
plot ( rate ~  exp10, data = dat)
for (i in 1:100){
  points(exp_seq, mu[i,], pch =.01,
         col = col.alpha("grey",.5))
}
```



Uncertainty about predictions

```
sim_rate <- sim(expenditureModel,
                data = data.frame(exp10 = exp_seq))

str(sim_rate)
```

```
##  num [1:1000, 1:51] -0.994 -0.976 -0.944 -0.98 -0.177 ...
```

```
sim_rate[1:5,1:5]
```

```
##            [,1]        [,2]      [,3]       [,4]        [,5]
## [1,] -0.9944427  0.09925683 -1.645604 -0.7608645 -1.58630334
## [2,] -0.9759735 -0.52167234 -2.109010 -0.2201428 -0.26959651
## [3,] -0.9442590 -1.54479304 -2.302338  0.2869969  0.05418225
## [4,] -0.9797929 -1.64638349 -1.357738 -1.3867955 -1.77313715
## [5,] -0.1766870 -1.95249812 -1.887265 -0.3244647 -0.45262283
```

Uncertainty about predictions

```r
mu_sim <- apply(sim_rate, 2, mean)
str(mu_sim)
```

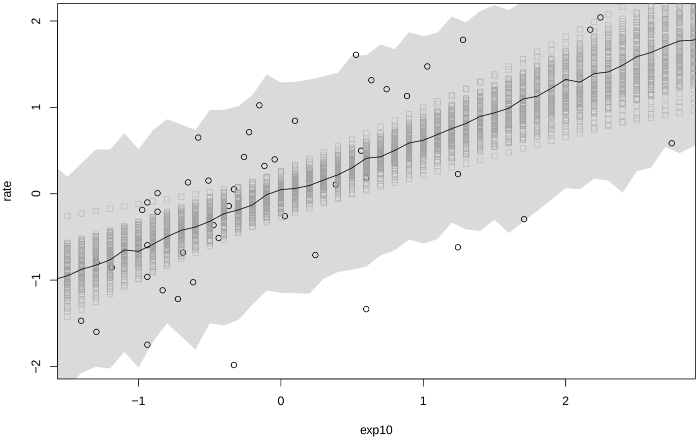
```
##  num [1:51] -1.24 -1.198 -1.119 -1.064 -0.998 ...
```

```r
hpdi_sim <- apply(sim_rate, 2, HPDI, prob = .89)

str(hpdi_sim)
```

```
##  num [1:2, 1:51] -2.5612 -0.0114 -2.4274 0.1438 -2.3868 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:2] "|0.89" "0.89|"
##   ..$ : NULL
```

Uncertainty about predictions

```
plot ( rate ~  exp10, data = dat)
for (i in 1:100){
  points(exp_seq, mu[i,], pch =.01,
         col = col.alpha("grey",.5))
}
lines( exp_seq, mu_sim)
shade( hpdi_sim, exp_seq)
```



Overfitting and complexity

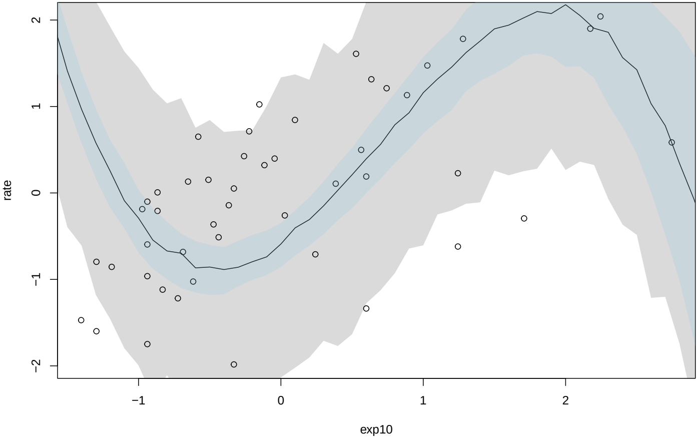
```
dat$exp10_2 <- dat$exp10^2
dat$exp10_3 <- dat$exp10^3


expenditureModelPoly <- quap(
  alist(
    rate ~ dnorm( mu , sigma ) ,
    mu <- a + b1 * exp10 + b2 + exp10_2 +
      b3 * exp10_3,
    a ~ dnorm( 0 ,  3) ,
    c(b1, b2, b3) ~ dnorm( 0 , 3 ) ,
    sigma ~ dexp(1 )
  ), data = dat
)
```

Overfitting and complexity

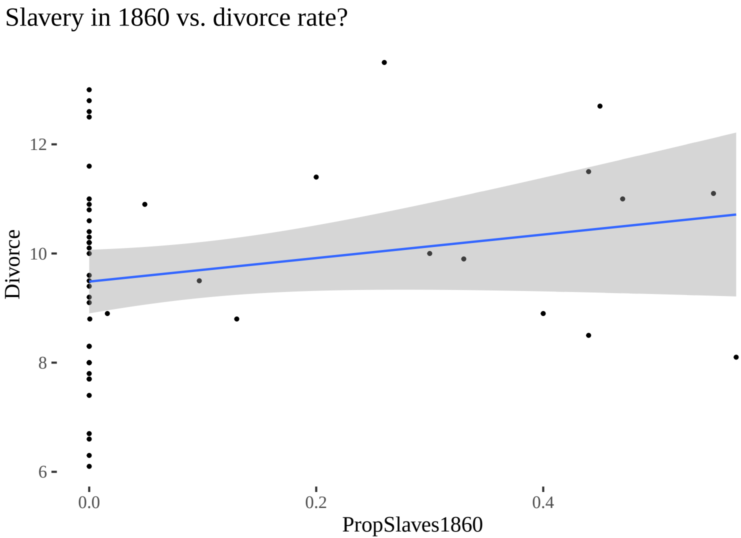
```
pred_df <- list(exp10 = exp_seq,
                exp10_2 = exp_seq^2,
                exp10_3 = exp_seq^3
                )

mu_poly_mean <- link(expenditureModelPoly, data = pred_df)
hpdi_poly_mean <- apply( mu_poly_mean, 2, HPDI, prob = .89)

mu_poly <- sim(expenditureModelPoly, data = pred_df)
mean_poly <- apply( mu_poly, 2, mean)
hpdi_poly <- apply( mu_poly, 2, HPDI, prob = .89)
```

Overfitting and complexity

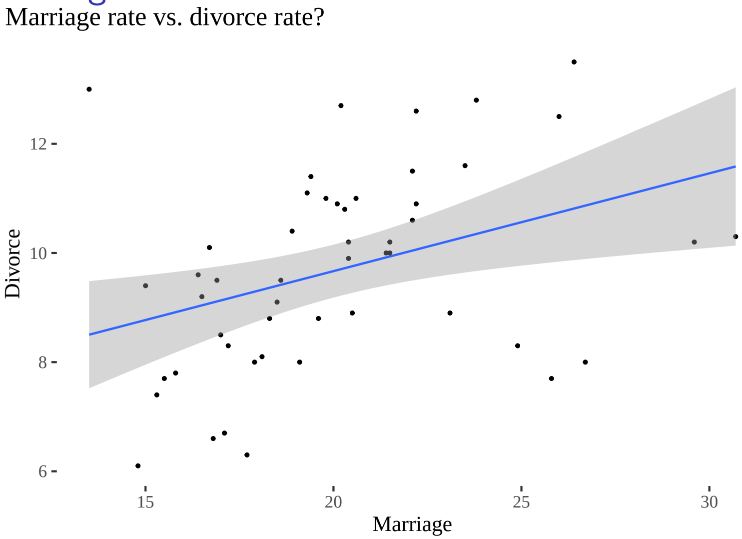
```
plot ( rate ~  exp10, data = dat)
lines( exp_seq, mean_poly)
shade( hpdi_poly, exp_seq)
shade( hpdi_poly_mean, exp_seq,
       col = col.alpha("skyblue",.2))
```



Confounding: first stab



Confounding: first stab

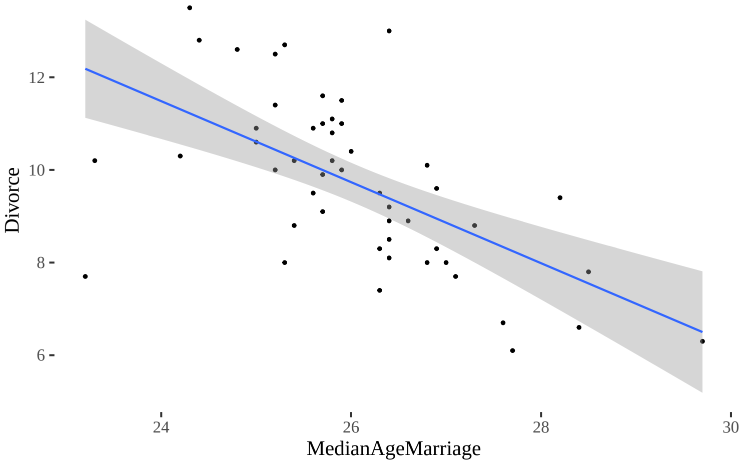


Question

Does marriage rate cause divorce rate?

Confounding: first stab

Median age at marriage vs. divorce rate?



Question

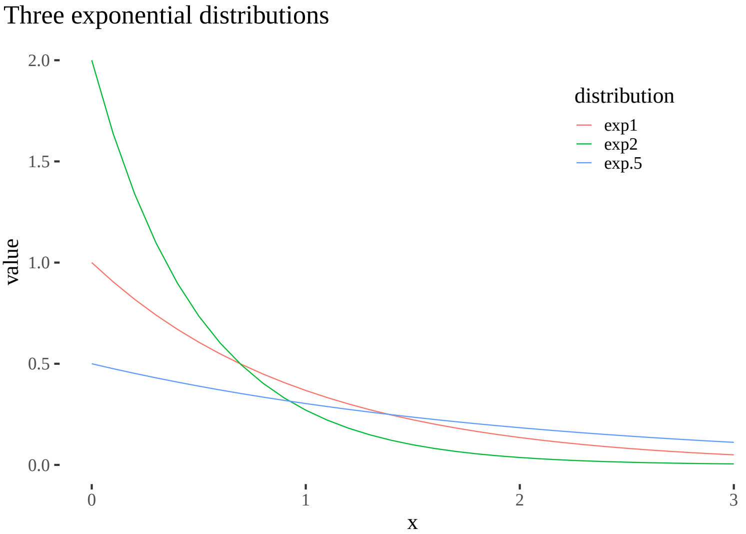
Does difference in median age of marriage impact divorce rate?

Confounding: first stab

```r
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
)
```

Exponential distribution

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



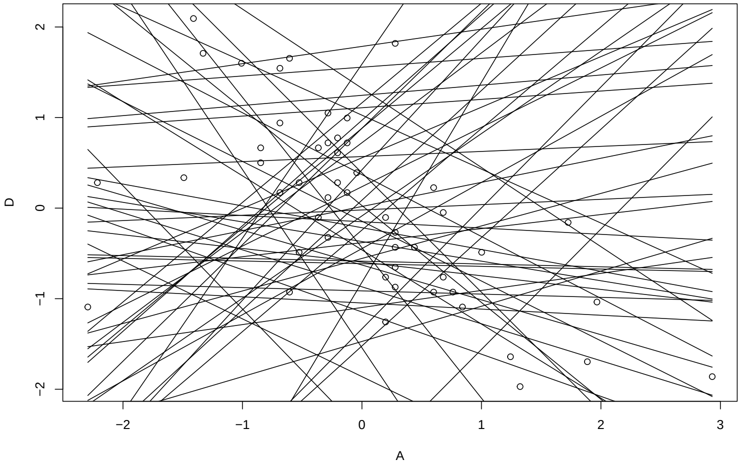
Prior predictive check

```r
set.seed(20)
prior <- as.data.frame(extract.prior(ageModelWide, n = 50))
head(prior, n = 2)
```

```
##            a          bA     sigma
## 1  1.1626853  1.09943524 0.4163377
## 2 -0.5859245 -0.03091713 0.2101736
```

Prior predictive check

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



Prior predictive check

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

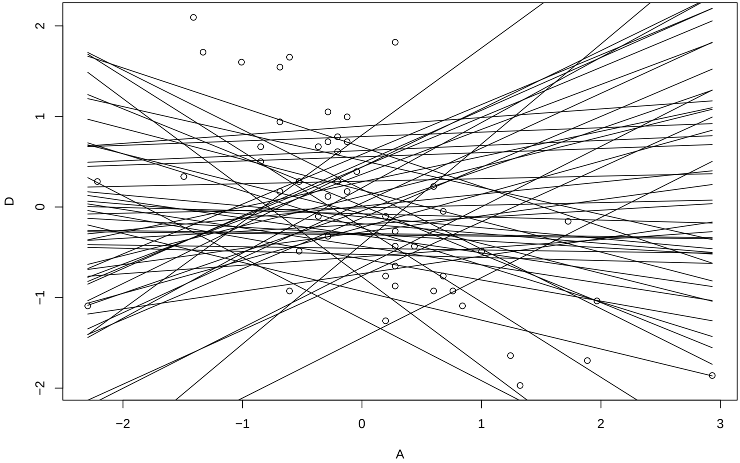
Prior predictive check

```
set.seed(20)
priorNarrow <- as.data.frame(extract.prior(ageModelNarrow, n = 50))
head(prior, n = 2)
```

```
##            a          bA     sigma
## 1  1.1626853  1.09943524 0.4163377
## 2 -0.5859245 -0.03091713 0.2101736
```

Prior predictive check

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



Posterior predictive check

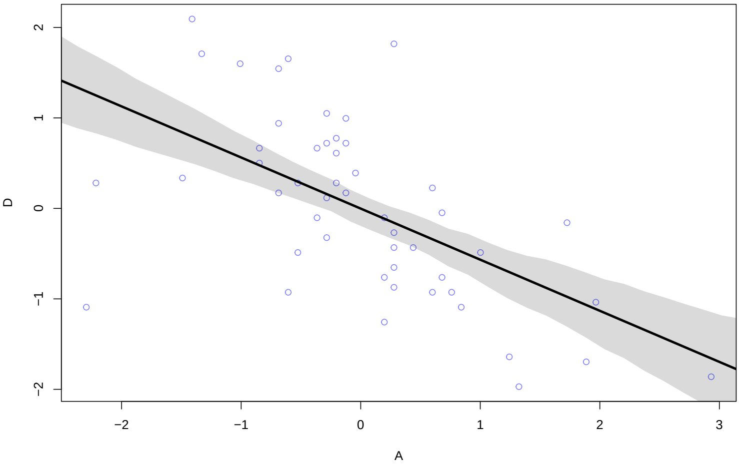
```r
A_range <- seq(-4,4, length.out = 50)
mu <- link(ageModelNarrow, data  = list(A = A_range))
str(mu)
```

```
##  num [1:1000, 1:50] 2.36 3.04 2.13 2.13 2.84 ...
```

```r
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

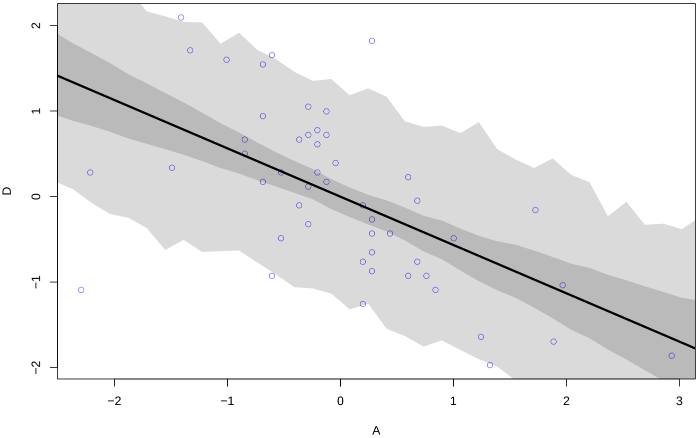
```r
pred <- sim(ageModelNarrow, data  = list(A = A_range))
str(pred)
```

```
##  num [1:1000, 1:50] 1.99 3.56 3.78 2.79 1.76 ...
```

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

Posterior predictions

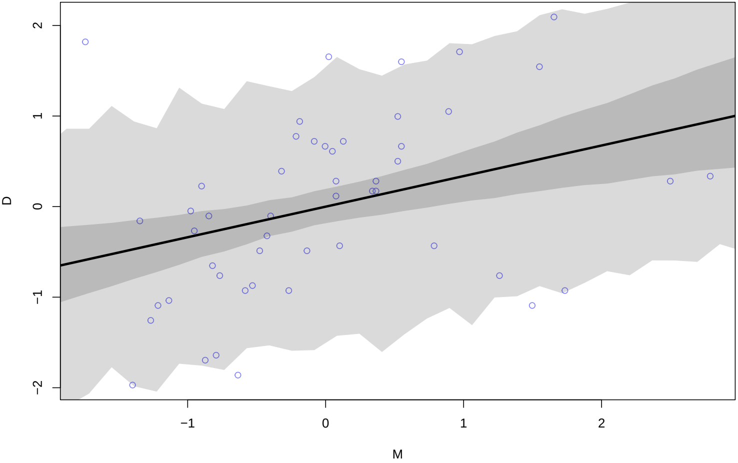
```
plot(D ~ A, data = d, col = rangi2)
lines(A_range, mu_mean, lwd = 3)
shade(mu_hpdi, A_range)
shade(pred_hpdi, A_range)
```



Now just marriage rate

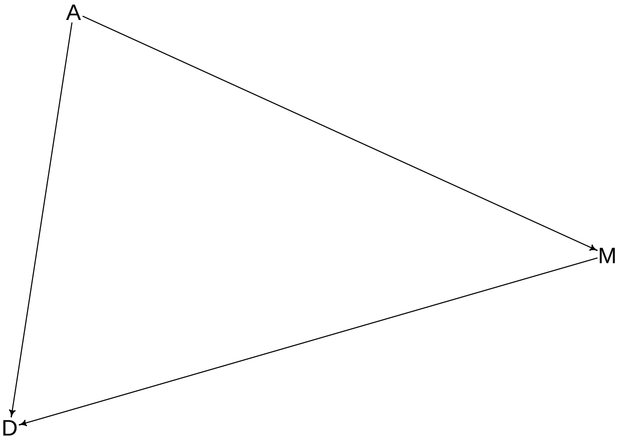
```
marriageModelNarrow <- quap(
  alist(
    D ~ dnorm(mu, sigma ) ,
    mu <- m + bM * M ,
    m ~ dnorm(0, .5),
    bM ~ dnorm( 0, .5),
    sigma ~ dexp( .5 )
  ), data = d
)
```

Now just marriage rate

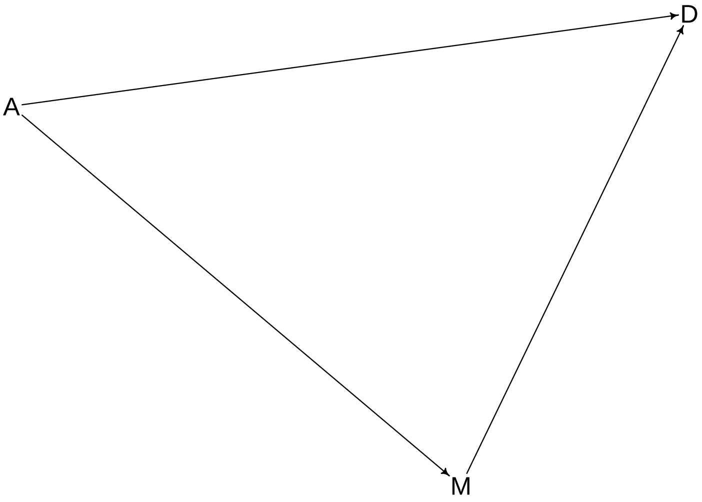


DAGs to the rescue!

```
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!