Computational Statistics & Probability

Problem Set 3 - Information Criteria and Interactions

Author: Neelesh Bhalla
Collaborators: Nils Marthiensen, Chia-Jung Chang

2022-11-30

1. Collider Bias and Information Criteria

Return to the textbook example in §6.3.1, which explores the relationship between age, marriage and happiness.

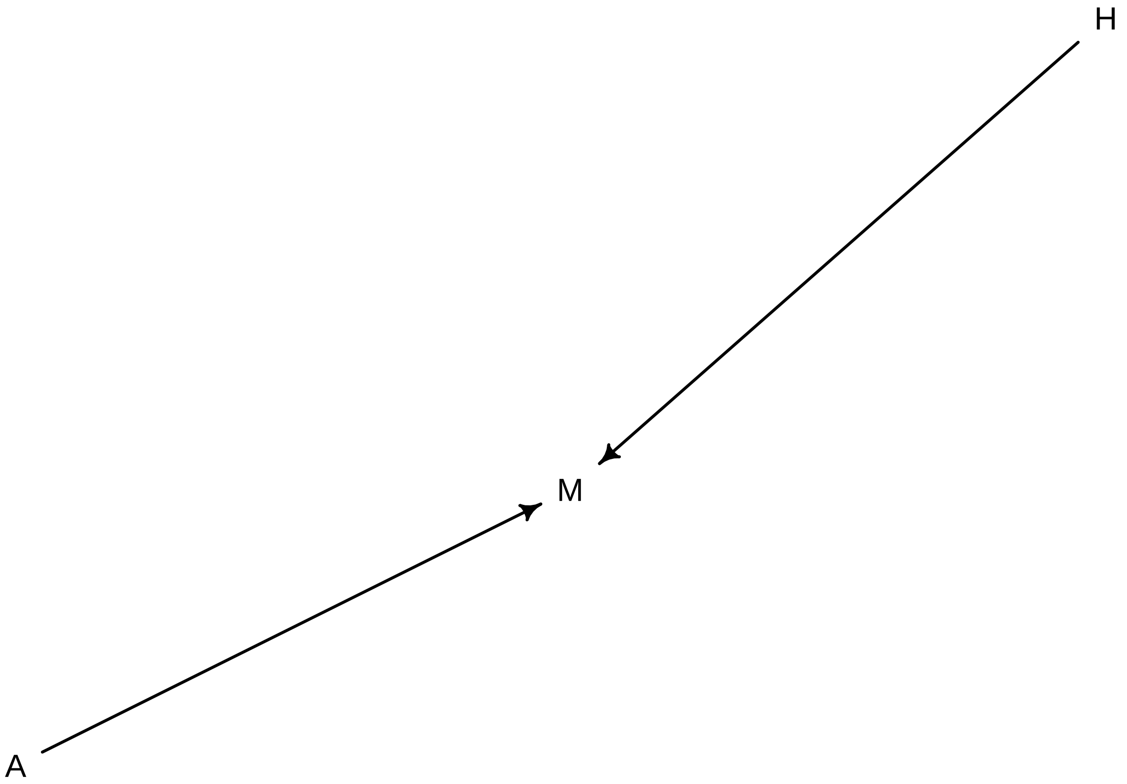
```
library(rethinking)
```

```
## Loading required package: rstan

## Loading required package: StanHeaders

##
## rstan version 2.26.13 (Stan version 2.26.1)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
## For within-chain threading using `reduce_sum()` or `map_rect()` Stan functions,
## change `threads_per_chain` option:
## rstan_options(threads_per_chain = 1)

## Loading required package: cmdstanr

## This is cmdstanr version 0.5.3

## - CmdStanR documentation and vignettes: mc-stan.org/cmdstanr

## - CmdStan path: /Users/neelesh/.cmdstan/cmdstan-2.30.1

## - CmdStan version: 2.30.1

##
## A newer version of CmdStan is available. See ?install_cmdstan() to install it.
## To disable this check set option or environment variable CMDSTANR_NO_VER_CHECK=TRUE.

## Loading required package: parallel

## rethinking (Version 2.21)

##
## Attaching package: 'rethinking'

## The following object is masked from 'package:rstan':
##
##     stan
## The following object is masked from 'package:stats':
##
##     rstudent
```

```
d <- sim_happiness( seed=1515 , N_years=1000)
d2 <- d[ d$age>17 , ] # only adults
d2$A <- (d2$age - 18) / (65 - 18)
d2$mid <- d2$married + 1
precis(d)[,1:4]
```

```
##                     mean          sd      5.5%     94.5%
## age        3.300000e+01 18.7688832  4.000000 62.000000
## married    2.930769e-01  0.4553486  0.000000  1.000000
## happiness -1.000070e-16  1.2144211 -1.789474  1.789474
```

```
# drawing the D.A.G.
library(dagitty)
dag_q1 <- dagitty('dag{ H -> M <- A }')
drawdag( dag_q1 )
```



a) Which model is expected to make better predictions according to these information criteria?

```
# recalling model m6.9 that considers the effect of age and marriage status on happiness
m6.9 <- quap(
alist(
happiness ~ dnorm( mu , sigma ),
mu <- a[mid] + bA*A,
a[mid] ~ dnorm( 0 , 1 ),
bA ~ dnorm( 0 , 2 ),
sigma ~ dexp(1)
) , data=d2 )
```

```r
precis(m6.9,depth=2)
```

```
##             mean         sd       5.5%      94.5%
## a[1]  -0.1947491 0.06521259 -0.2989714 -0.0905268
## a[2]   1.2161249 0.08876809  1.0742563  1.3579934
## bA    -0.7325968 0.11708179 -0.9197161 -0.5454775
## sigma  1.0199308 0.02325838  0.9827594  1.0571022
```

```r
# model m6.9 is quite sure that age is negatively associated with happiness.
```

```r
# recalling model m6.10 that omits marriage status
m6.10 <- quap(
alist(
happiness ~ dnorm( mu , sigma ),
mu <- a + bA*A,
a ~ dnorm( 0 , 1 ),
bA ~ dnorm( 0 , 2 ),
sigma ~ dexp(1)
) , data=d2 )
precis(m6.10)
```

```
##                mean         sd       5.5%     94.5%
## a     -9.798009e-06 0.07674935 -0.1226701 0.1226505
## bA     3.917361e-05 0.13225839 -0.2113353 0.2114136
## sigma  1.213175e+00 0.02766008  1.1689688 1.2573811
```

```r
# model m6.10 in contrast, finds no association between age and happiness.
```

```r
# Notes from the book (pg 184):
# The pattern above is exactly what we should expect when we condition on a collider.
# The collider is marriage status. It a common consequence of age and happiness.
# As a result, when we condition on it, we induce a spurious association between the two causes.
# So it looks like, to model m6.9, that age is negatively associated with happiness.
# But this is just a statistical association, not a causal association.
# Once we know whether someone is married or not, then their age does provide information
# about how happy they are.
```

Compare the two models, m6.9 and m6.10, using both PSIS and WAIC.

```r
# comparing models using PSIS
compare( m6.9, m6.10, func=PSIS)
```

```
##           PSIS       SE    dPSIS     dSE    pPSIS       weight
## m6.9  2771.068 37.13856   0.0000      NA 3.752925 1.000000e+00
## m6.10 3102.028 27.71311 330.9597 34.1829 2.415411 1.358367e-72
```

```r
# comparing models using WAIC
compare( m6.9, m6.10, func=WAIC)
```

```
##           WAIC       SE    dWAIC      dSE    pWAIC       weight
## m6.9  2771.393 36.93196   0.0000       NA 3.960565 1.000000e+00
## m6.10 3101.906 27.68192 330.5132 34.02321 2.330043 1.698146e-72
```

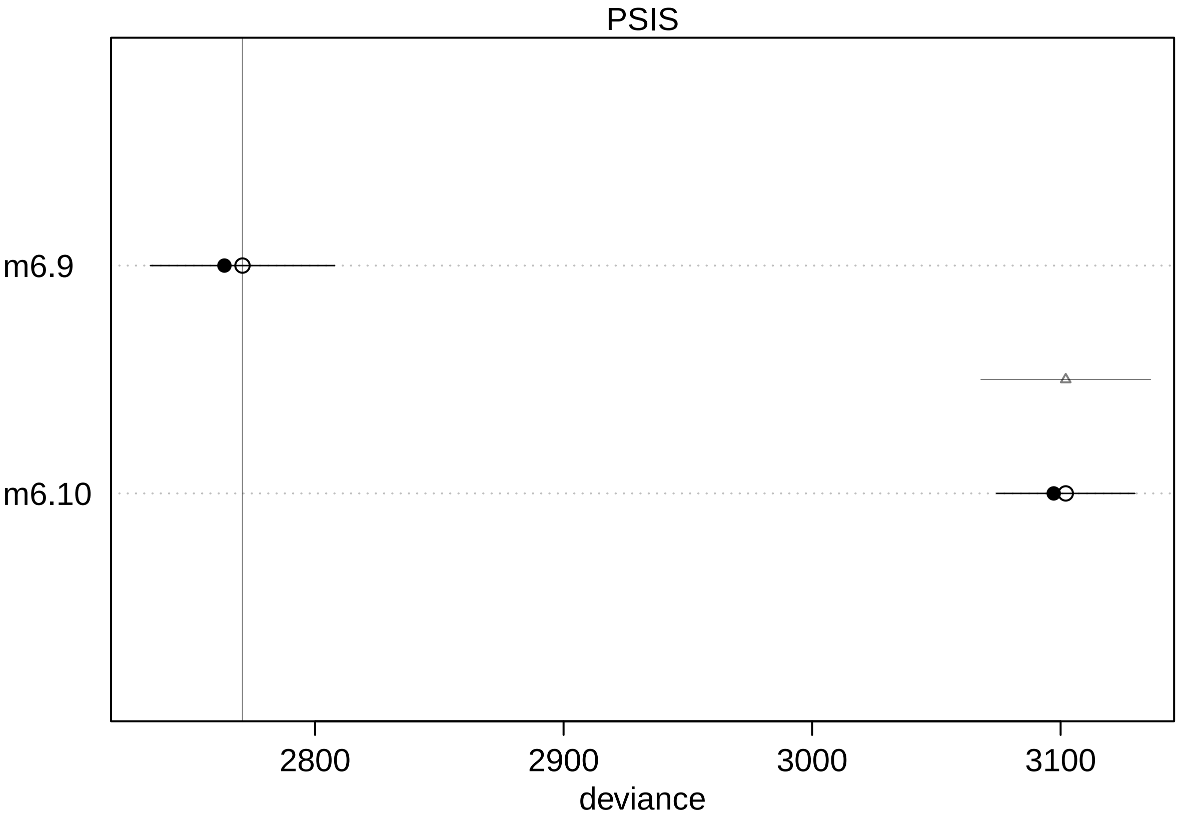
a) Which model is expected to make better predictions according to these information criteria?

```r
# Both PSIS and WAIC values suggest that model m6.9 (that considers the effect of age and
# marriage status on happiness) is expected to make better predictions than model m6.10 (that
# omits marriage status)
# Smaller values of PSIS and WAIC are better.
```

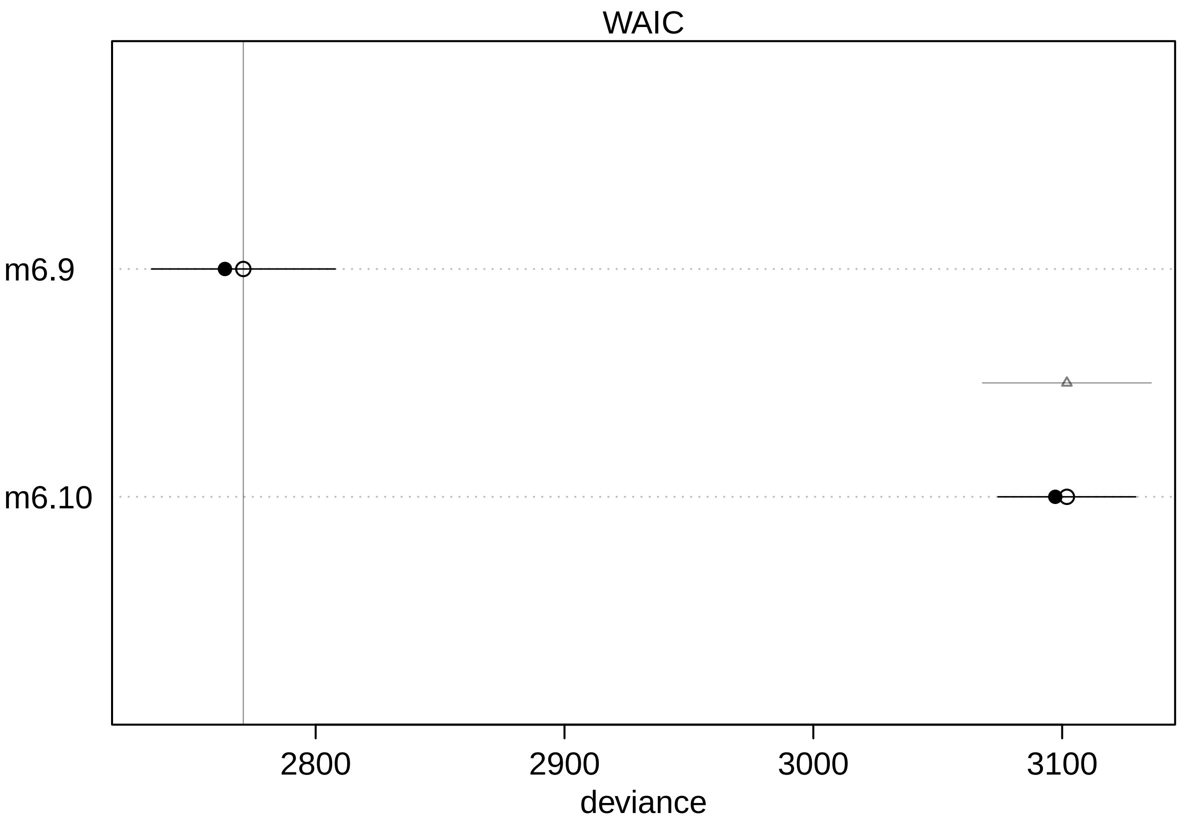
b) On the basis of the causal model, how should you interpret the parameter estimates from the model preferred by PSIS and WAIC?

```
# The pWAIC and pPSIS are the penalty terms.
# These values are close to the number of dimensions (3 for m6.9 and 2 for m6.10)
# in the posterior of each model

# The columns dWAIC and dPSIS reflect the difference between each model's WAIC/PSIS and
# the best WAIC/PSIS in the set. So it's zero for the best model and then the
# differences with the other models tell you how far apart each is from the top model.
# Model m6.9 is about 330 units of deviance smaller than models m6.10.

# SE is the approximate standard error of each WAIC/PSIS. In a very approximate sense,
# we expect out-of-sample accuracy to be normally distributed with mean equal to the
# reported WAIC/PSIS value and a standard deviation equal to the standard error.
# To judge whether two models are easy to distinguish, we don't use their standard
# errors but rather the standard error of their difference.
```

```
# visually understanding PSIS based difference in two models
plot( compare( m6.9 , m6.10, func=PSIS ) )
```



```
# visually understanding WAIC based difference in two models
plot( compare( m6.9 , m6.10, func=WAIC ) )
```



```
# The filled points are the in-sample deviance values. The open points are the WAIC values.
# Each model does better in-sample than it is expected to do out-ofsample.
# The line segments show the standard error of each WAIC. These are the values
# of SE in the table above. So we can see how much better m6.9 is than m6.10.
# The standard error of the difference in WAIC between the two models is shown by the lighter
# line segment with the triangle on it, between m6.9 and m6.10.
```

2. Laffer Curve

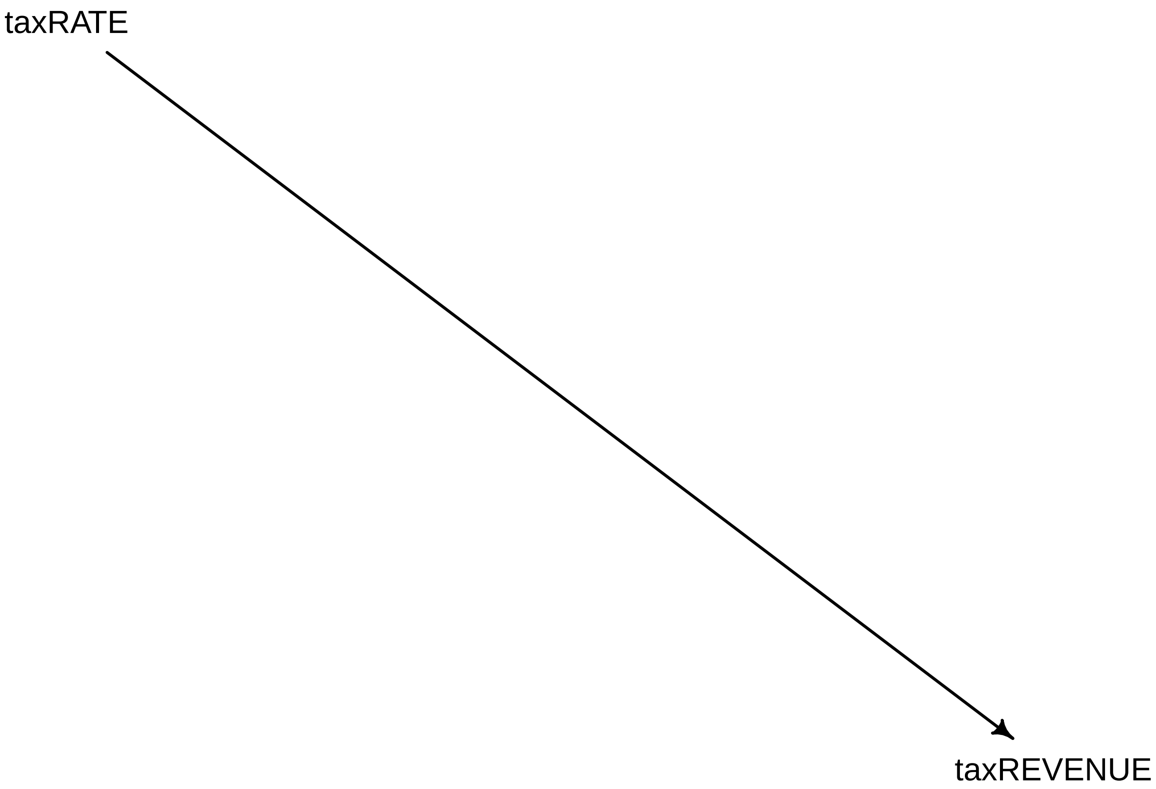
In 2007 The Wall Street Journal published an editorial arguing that raising corporate tax rates increases government revenues only to a point, after which higher tax rates produce less revenue for governments. The editorial included the following graph of corporate tax rates in 29 countries plotted against tax revenue, over which a Laffer curve was drawn. The data used in this plot are available in the rethinking package.

```
library(rethinking)
data(Laffer)
d3 <- Laffer
precis( d3 )[,1:4]
```

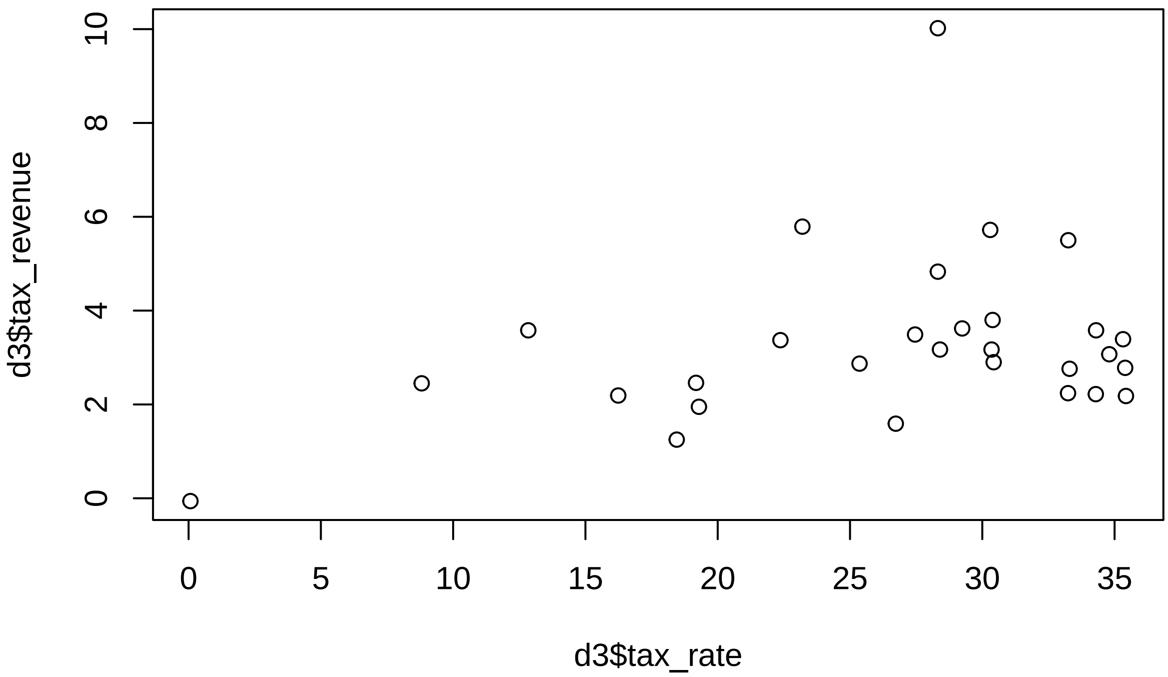
```
##                  mean       sd    5.5%   94.5%
## tax_rate    26.382069 8.753579 10.9862 35.3568
## tax_revenue  3.306207 1.816491  1.4336  5.7522
```

a) Using this data, fit a basic regression that uses tax rate to predict tax revenue. Simulate and justify your priors.

```
library(dagitty)
dag_q2 <- dagitty('dag{ taxRATE -> taxREVENUE }')
drawdag( dag_q2 )
```

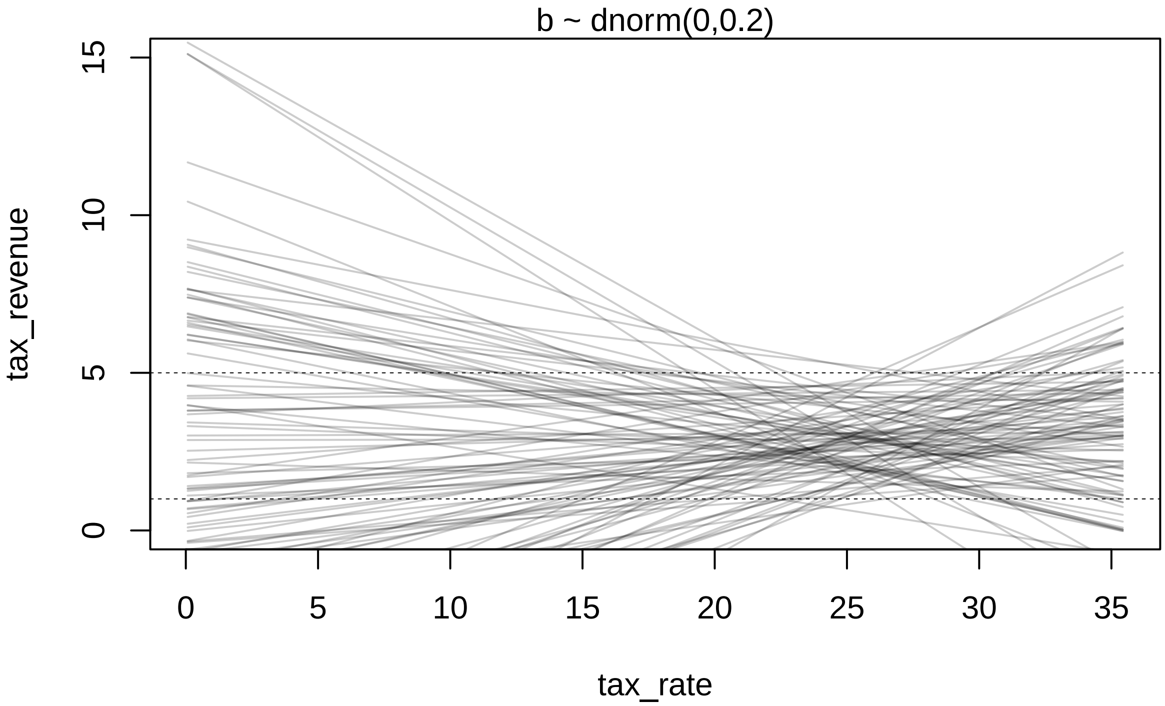


```
# plotting the data
plot( d3$tax_revenue ~ d3$tax_rate )
```



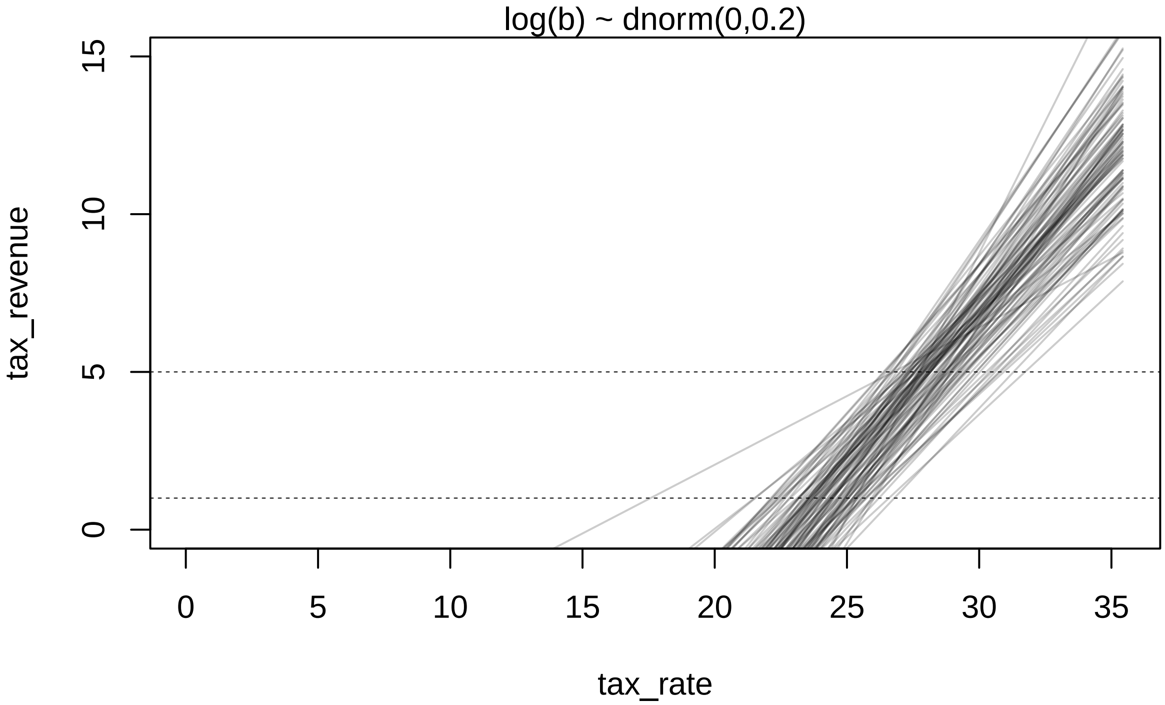
```
# Presenting prior predictive simulation for  tax_revenue & tax_rate model with
# normal-distribution of 'b'
# For starters, the intercept 'a' (mean tax_revenue) can assume a normal distribution
# with a mean of 3
# While the slope 'b' can be assumed to be normally distributed and centered at mean 0,
```

```r
# with SD of 0.2

N <- 100
a <- rnorm( N , 3 , 1 )
b <- rnorm( N , 0 , 0.2 )
plot( NULL , xlim=range(d3$tax_rate) , ylim=c(0,15) ,
xlab="tax_rate" , ylab="tax_revenue" )
abline( h=1 , lty=2, lwd=0.5 )
abline( h=5 , lty=2 , lwd=0.5 )
mtext( "b ~ dnorm(0,0.2)" )
xbar <- mean(d3$tax_rate)
for ( i in 1:N ) curve( a[i] + b[i]*(x - xbar) ,
from=min(d3$tax_rate) , to=max(d3$tax_rate) , add=TRUE ,
col=col.alpha("black",0.2) )
```



```r
# The values of b are completely random in first iteration suggesting tax_revenue could
# be related to tax_rate in any random fashion, which should not be the ideal case.
```

```r
# Attempting prior predictive simulation for tax_revenue & tax_rate model with
# log-normal-distribution of 'b'. Let's try restricting it to positive values
# assuming that average tax_revenue increases with average tax_rate, at least up to a point.
N <- 100
a <- rnorm( N , 3 , 1 )
b <- rlnorm( N , 0 , 0.2 )
# Prior predictive simulation for the tax_revenue and tax_rate model
plot( NULL , xlim=range(d3$tax_rate) , ylim=c(0,15) ,
xlab="tax_rate" , ylab="tax_revenue" )
abline( h=1 , lty=2, lwd=0.5 )
abline( h=5 , lty=2 , lwd=0.5 )
mtext( "log(b) ~ dnorm(0,0.2)" )
xbar <- mean(d3$tax_rate)
for ( i in 1:N ) curve( a[i] + b[i]*(x - xbar) ,
from=min(d3$tax_rate) , to=max(d3$tax_rate) , add=TRUE ,
col=col.alpha("black",0.2) )
```



```
# Posterior distribution for tax_revenue and tax_rate - basic linear model
# fit model
xbar <- mean(d3$tax_rate)
m2a <- quap(
alist(
tax_revenue ~ dnorm( mu , sigma ) ,
mu <- a + b*( tax_rate - xbar ) ,
a ~ dnorm( 3 , 1 ) ,
b ~ dlnorm( 0 , 0.2 ) ,
sigma ~ dunif( 0 , 4 )
) ,
data=d3 )
# the marginal posterior distributions is as follows
precis( m2a )
```

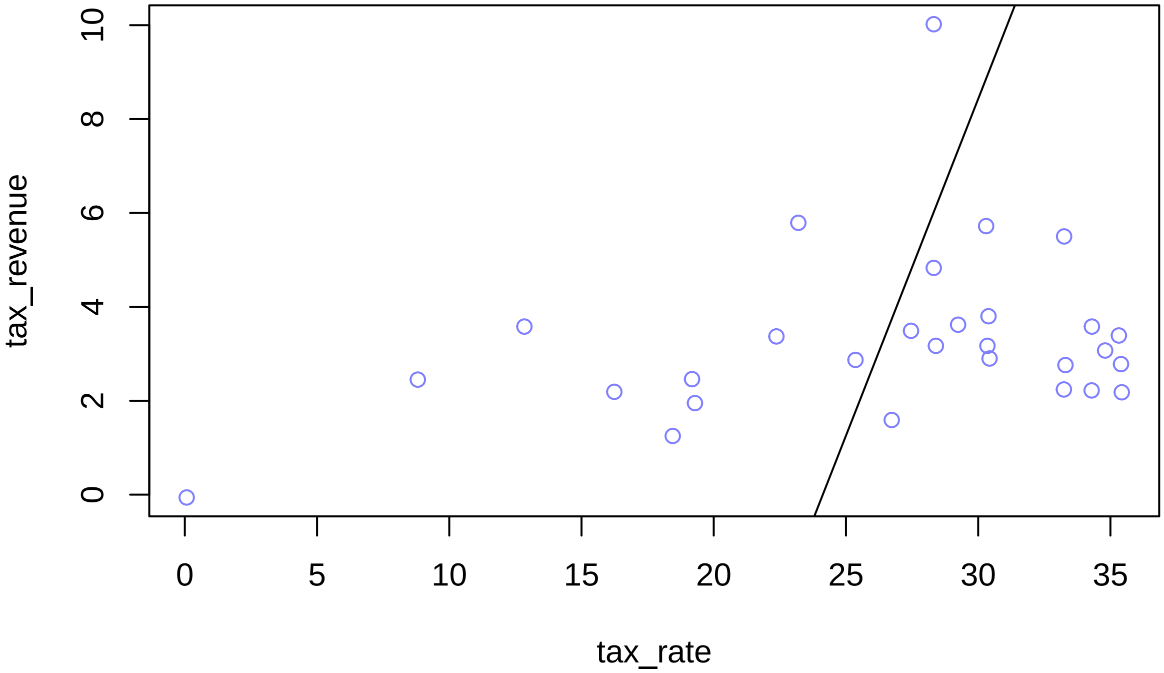
```
##            mean         sd      5.5%     94.5%
## a     3.2321651 0.49219863 2.4455366 4.0187935
## b     0.3600818 0.05836684 0.2668003 0.4533633
## sigma 3.0411687 0.57780854 2.1177191 3.9646184
```

```
round( vcov( m2a ) , 3 )
```

```
##            a      b  sigma
## a      0.242 -0.001 -0.012
## b     -0.001  0.003  0.024
## sigma -0.012  0.024  0.334
```

```
# the variance-covariance matrix
```

```
plot( tax_revenue ~ tax_rate , data=d3 , col=rangi2 )
post <- extract.samples( m2a )
a_map <- mean(post$a)
b_map <- exp(mean(post$b))
curve( a_map + b_map*(x - xbar) , add=TRUE )
```



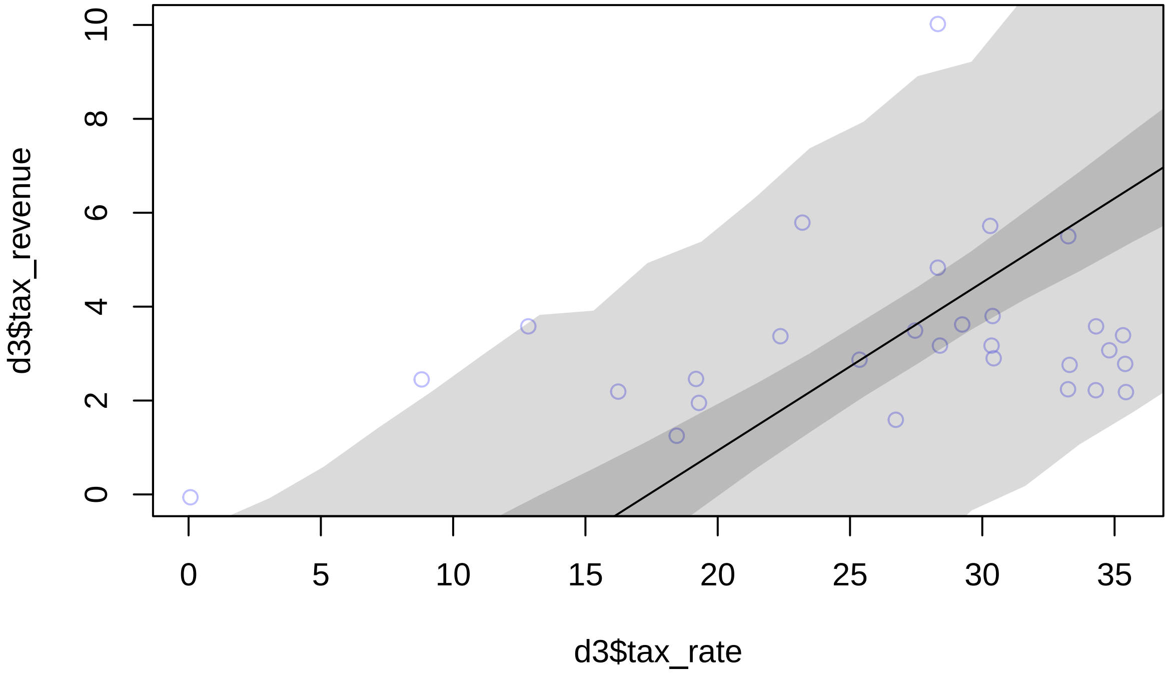
b) Now construct and fit any curved model you wish to the data. Plot your straight-line model and your new curved model. Each plot should include 89% PI intervals.

```
# Posterior distribution for tax_revenue and tax_rate making use of a
# QUADRATIC_POLYNOMIAL based model
d3$tax_rate_s <- ( d3$tax_rate - mean(d3$tax_rate) )/sd(d3$tax_rate)
d3$tax_rate_s2 <- d3$tax_rate_s^2
m2b_2 <- quap(
  alist(
tax_revenue ~ dnorm( mu , sigma ) ,
mu <- a + b1*tax_rate_s + b2*tax_rate_s2 ,
a ~ dnorm( 3 , 1 ) ,
b1 ~ dlnorm( 0 , 0.2 ) ,
b2 ~ dnorm( 0 , 0.2 ) ,
sigma ~ dunif( 0 , 4 )
) ,
data=d3 )
precis(m2b_2)
```

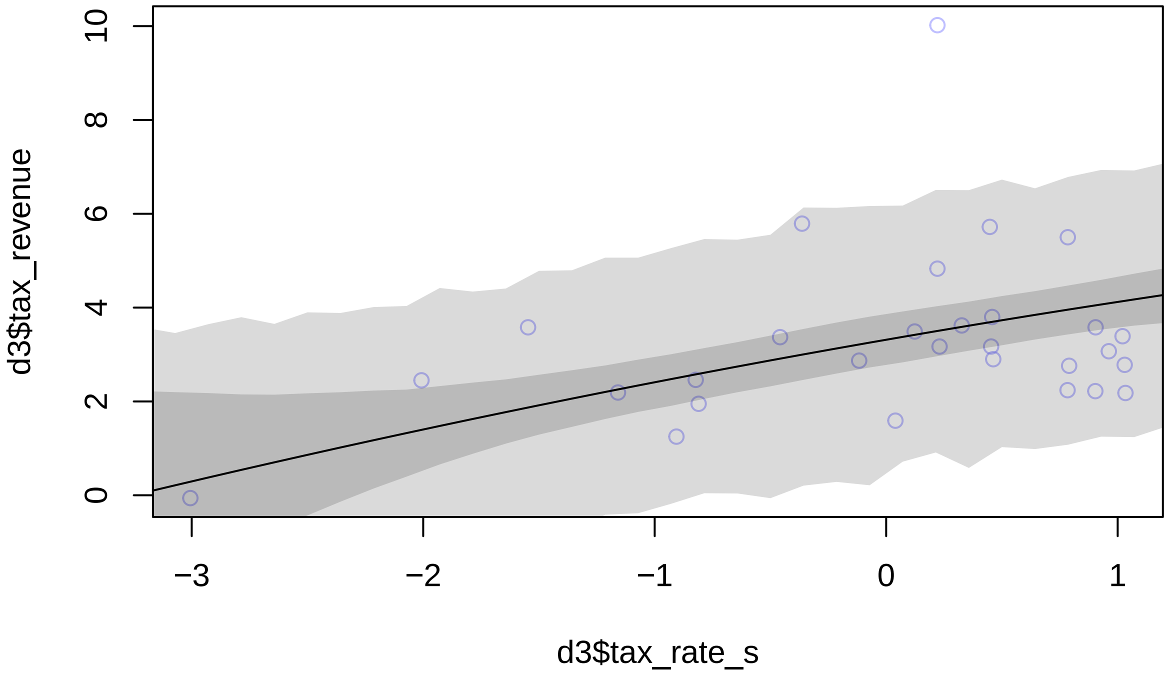
```
##                mean        sd       5.5%     94.5%
## a        3.32166224 0.3259466  2.8007367 3.8425878
## b1       0.85882695 0.1510481  0.6174230 1.1002309
## b2      -0.04947294 0.1383547 -0.2705904 0.1716445
## sigma    1.70678289 0.2267825  1.3443406 2.0692251
```

```
# Plotting the LINEAR model now
tax_rate.seq <- seq(from=-50, to=50, length.out=50)
```

```
pred_dat <- list( tax_rate=tax_rate.seq )
mu <- link( m2a , data=pred_dat )
mu.mean <- apply( mu , 2 , mean )
mu.PI <- apply( mu , 2 , PI , prob=0.89 )
sim.tax_revenue <- sim( m2a , data=pred_dat )
tax_revenue.PI <- apply( sim.tax_revenue , 2 , PI , prob=0.89 )

plot( d3$tax_revenue ~ d3$tax_rate , d , col=col.alpha(rangi2,0.5) )
lines( tax_rate.seq , mu.mean )
shade( mu.PI , tax_rate.seq )
shade( tax_revenue.PI , tax_rate.seq )
```



```
# Plotting the QUADRATIC POLYNOMIAL model now
tax_rate.seq <- seq(from=-3.5, to=3.5, length.out=50)
pred_dat <- list( tax_rate_s=tax_rate.seq , tax_rate_s2=tax_rate.seq^2 )
mu <- link( m2b_2 , data=pred_dat )
mu.mean <- apply( mu , 2 , mean )
mu.PI <- apply( mu , 2 , PI , prob=0.89 )
sim.tax_revenue <- sim( m2b_2 , data=pred_dat )
tax_revenue.PI <- apply( sim.tax_revenue , 2 , PI , prob=0.89 )

plot( d3$tax_revenue ~ d3$tax_rate_s , d , col=col.alpha(rangi2,0.5) )
lines( tax_rate.seq , mu.mean )
shade( mu.PI , tax_rate.seq )
shade( tax_revenue.PI , tax_rate.seq )
```



c) Using WAIC or PSIS, compare a straight-line model to your curved model. What conclusions would you draw from comparing your two models?

```
# comparing models using WAIC
compare( m2a, m2b_2, func=WAIC)
```

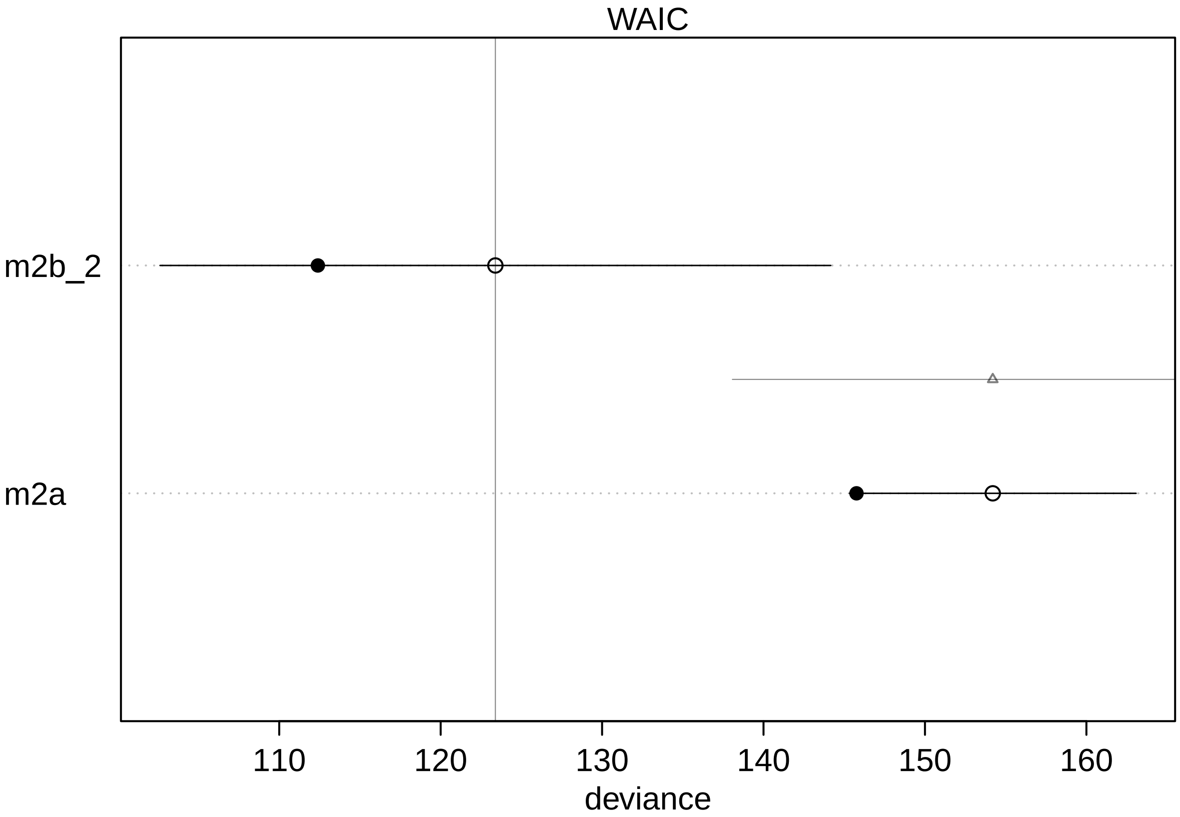
```
##            WAIC       SE  dWAIC      dSE    pWAIC       weight
## m2b_2 123.9816 21.41884  0.000       NA 5.819023 9.999995e-01
## m2a   152.9196  8.24238 28.938 17.61486 3.614630 5.202275e-07
```

```
# comparing models using PSIS
compare( m2a, m2b_2, func=PSIS)
```

```
## Some Pareto k values are high (>0.5). Set pointwise=TRUE to inspect individual points.

## Some Pareto k values are very high (>1). Set pointwise=TRUE to inspect individual points.

##            PSIS        SE   dPSIS      dSE    pPSIS       weight
## m2b_2 126.8757 24.333475  0.0000       NA 7.237916 9.999983e-01
## m2a   153.4591  8.690183 26.5834 19.60764 3.848416 1.688444e-06
```

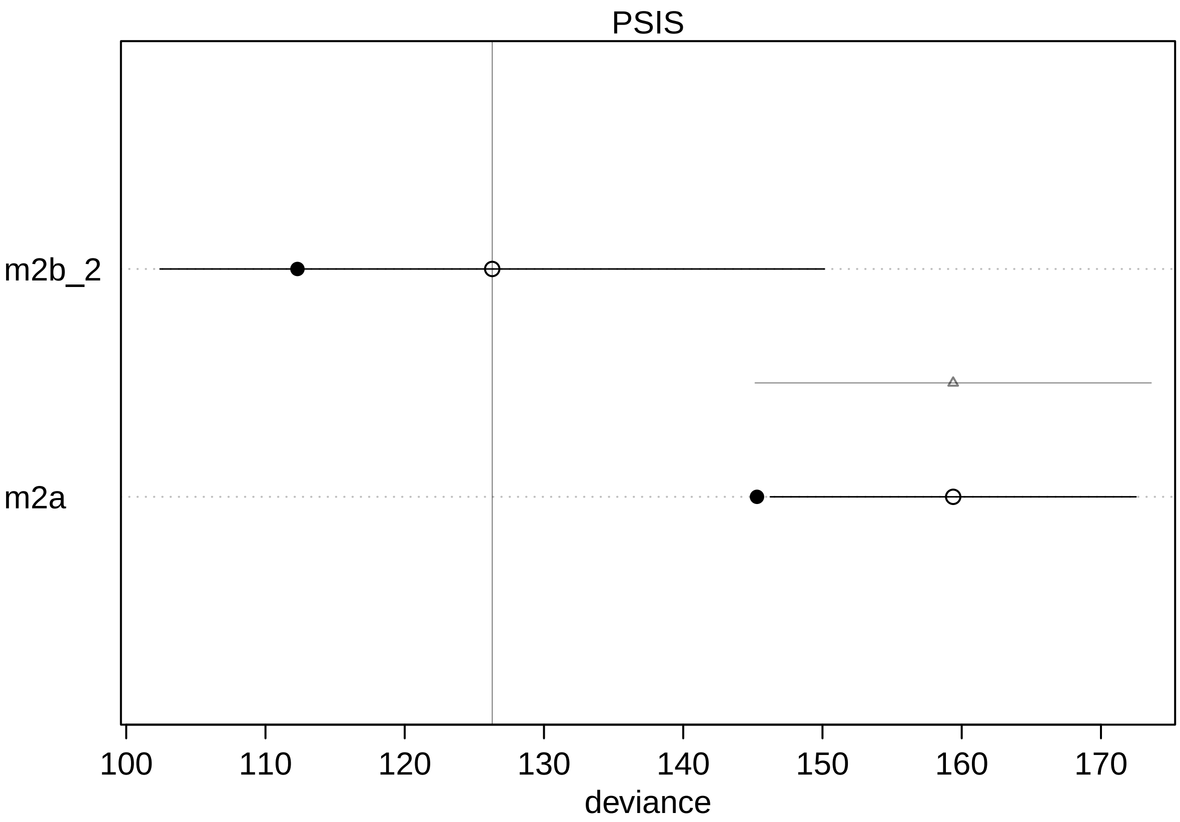
```
# Both PSIS and WAIC values suggest that QUADRATIC model m2b_2 is expected to make better
# predictions than LINEAR model m2a
# Smaller values of PSIS and WAIC are better.
# The pWAIC and pPSIS are the penalty terms.
# The columns dWAIC and dPSIS reflect the difference between each model's WAIC/PSIS and
# the best WAIC/PSIS in the set. So it's zero for the best model and then the
# differences with the other models tell you how far apart each is from the top model.
# Model m2b_2 is about 28 units of deviance smaller than models m6.10.

# SE is the approximate standard error of each WAIC/PSIS. In a very approximate sense,
# we expect out-of-sample accuracy to be normally distributed with mean equal to the
# reported WAIC/PSIS value and a standard deviation equal to the standard error.
# To judge whether two models are easy to distinguish, we don't use their standard
# errors but rather the standard error of their difference.
```

```
plot( compare( m2a, m2b_2, func=WAIC ) )
```



```
plot( compare( m2a, m2b_2, func=PSIS ) )
```

```
## Some Pareto k values are very high (>1). Set pointwise=TRUE to inspect individual points.
## Some Pareto k values are very high (>1). Set pointwise=TRUE to inspect individual points.
```



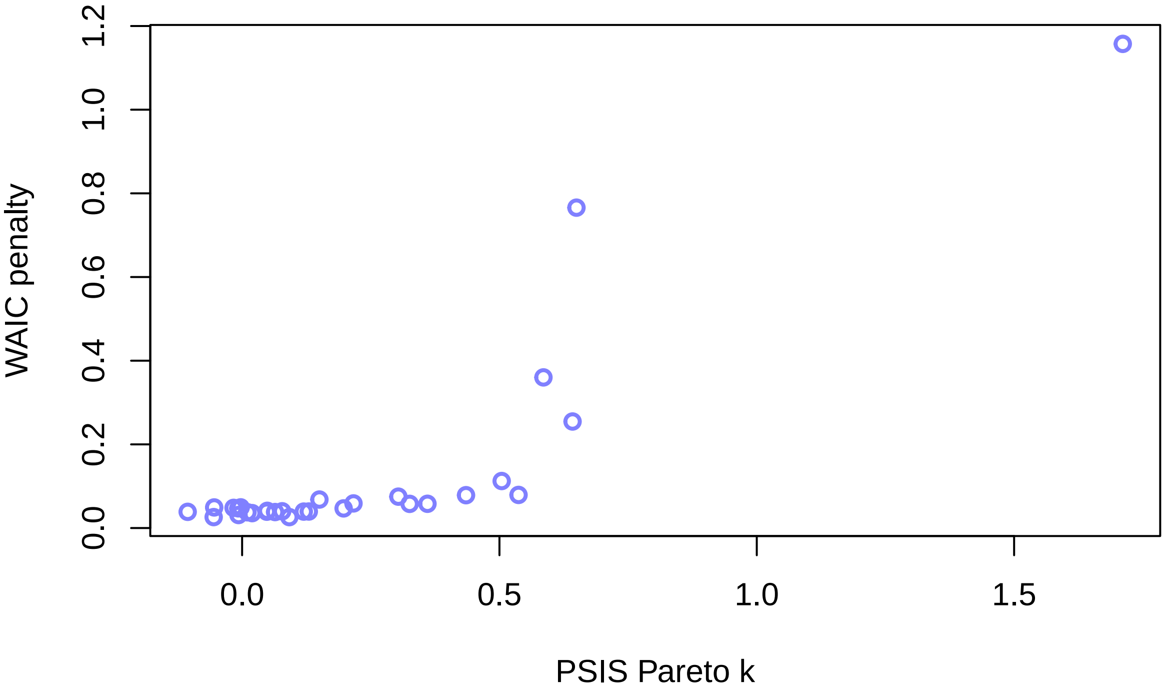
```
# When we look at the plots, we realise that model m2b_2 has a higher SE (~24) as compared to
# the SE (~8) of m2a . But m2b_2 is still a favorable choice of model because of its low PSIS and
# WAIC values
```

d) There is one country with a high tax revenue which is an outlier. Use PSIS and WAIC to measure the importance of this outlier in the two models you fit.

```
set.seed(24071847)
PSIS_m2a <- PSIS(m2a,pointwise=TRUE)
```

```
## Some Pareto k values are very high (>1). Set pointwise=TRUE to inspect individual points.
```

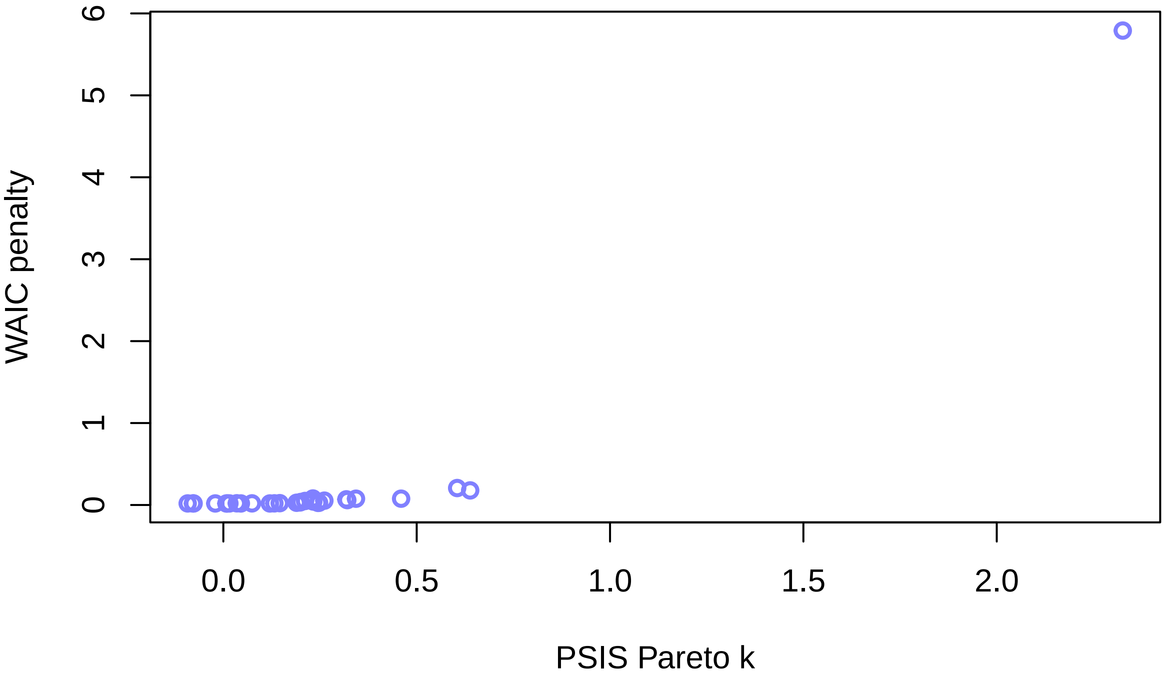
```
set.seed(24071847)
WAIC_m2a <- WAIC(m2a,pointwise=TRUE)
plot( PSIS_m2a$k , WAIC_m2a$penalty , xlab="PSIS Pareto k" ,
ylab="WAIC penalty" , col=rangi2 , lwd=2 )
```



```
set.seed(24071847)
PSIS_m2b_2 <- PSIS(m2b_2,pointwise=TRUE)
```

```
## Some Pareto k values are very high (>1). Set pointwise=TRUE to inspect individual points.
```

```
set.seed(24071847)
WAIC_m2b_2 <- WAIC(m2b_2,pointwise=TRUE)
plot( PSIS_m2b_2$k , WAIC_m2b_2$penalty , xlab="PSIS Pareto k" ,
ylab="WAIC penalty" , col=rangi2 , lwd=2 )
```

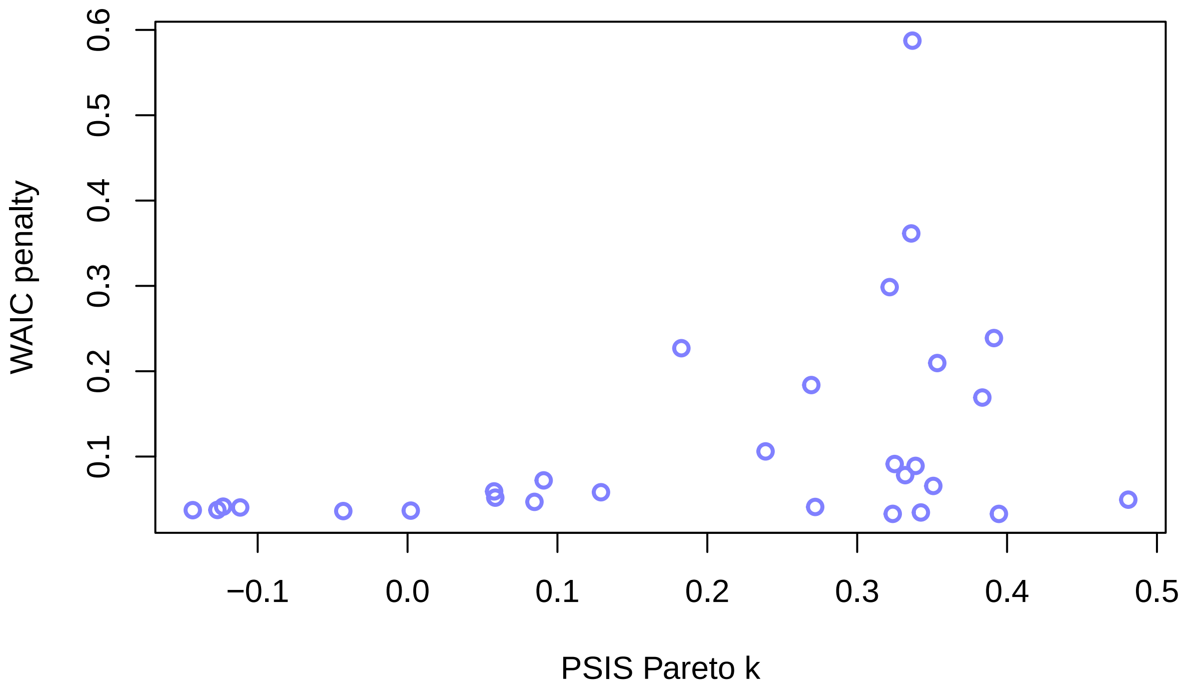


```
# In both of these cases, there is an identifiable outlier with high Pareto k value as well as
# high WAIC penalty.
# We need to account for this outlier because Points like these are highly influential and
# potentially hurt prediction.
```

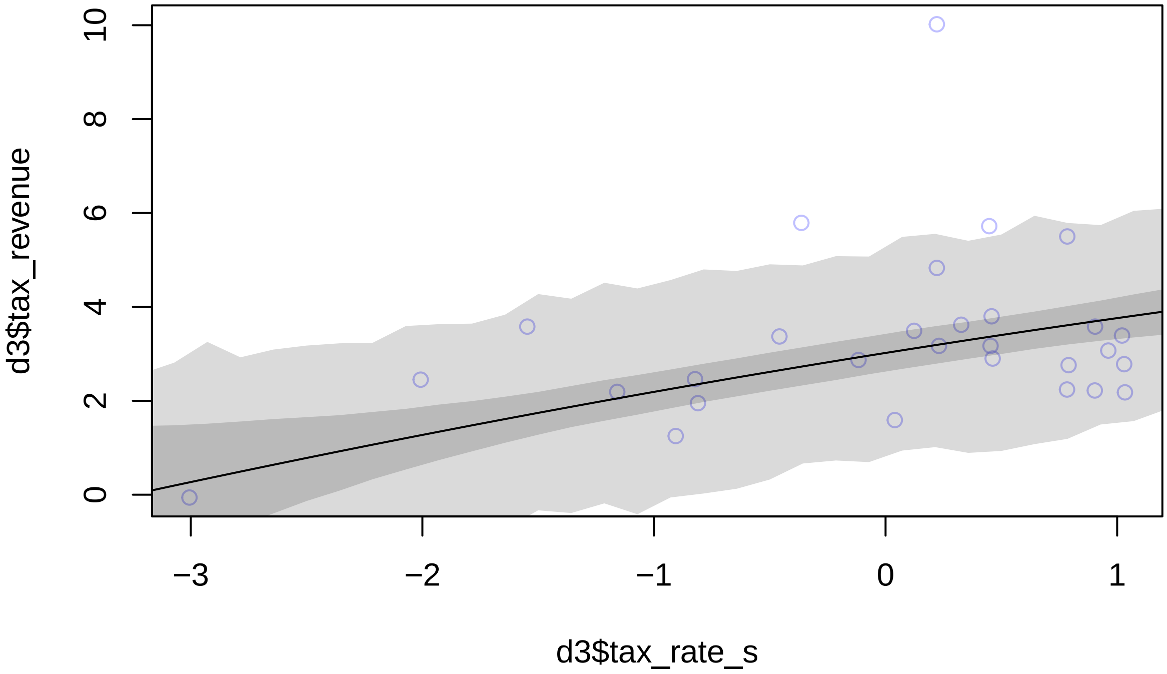
```
# Let's re-estimate the rate-revenue model model using a Student-t distribution with  = 4
# we call the model m2b_2t
m2b_2t <- quap(
alist(
tax_revenue ~ dstudent( 4 , mu , sigma ) ,
mu <- a + b1*tax_rate_s + b2*tax_rate_s2 ,
a ~ dnorm( 3 , 1 ) ,
b1 ~ dlnorm( 0 , 0.2 ) ,
b2 ~ dnorm( 0 , 0.2 ) ,
sigma ~ dunif( 0 , 4 )
) , data = d3 )
precis(m2b_2t)
```

```
##              mean        sd       5.5%     94.5%
## a      3.00860735 0.2522217  2.6055083 3.4117064
## b1     0.78652992 0.1326318  0.5745588 0.9985011
## b2    -0.04162556 0.1059588 -0.2109682 0.1277170
## sigma  1.07598878 0.1995687  0.7570394 1.3949382
```

```
# When we compute PSIS now, PSIS(m2b_2t), we don't get any warnings about Pareto k values.
set.seed(24071847)
PSIS_m2b_2t <- PSIS(m2b_2t,pointwise=TRUE)
set.seed(24071847)
WAIC_m2b_2t <- WAIC(m2b_2t,pointwise=TRUE)
plot( PSIS_m2b_2t$k , WAIC_m2b_2t$penalty , xlab="PSIS Pareto k" ,
ylab="WAIC penalty" , col=rangi2 , lwd=2 )
```



```
# NOW, Plotting the QUADRATIC model m2b_2t (with Student's T distribution)
tax_rate.seq <- seq(from=-3.5, to=3.5, length.out=50)
pred_dat <- list( tax_rate_s=tax_rate.seq , tax_rate_s2=tax_rate.seq^2 )
mu <- link( m2b_2t , data=pred_dat )
mu.mean <- apply( mu , 2 , mean )
mu.PI <- apply( mu , 2 , PI , prob=0.89 )
sim.tax_revenue <- sim( m2b_2t , data=pred_dat )
tax_revenue.PI <- apply( sim.tax_revenue , 2 , PI , prob=0.89 )

plot( d3$tax_revenue ~ d3$tax_rate_s , d , col=col.alpha(rangi2,0.5) )
lines( tax_rate.seq , mu.mean )
shade( mu.PI , tax_rate.seq )
shade( tax_revenue.PI , tax_rate.seq )
```



```
# From the above graph as well, we can see that now the 89% PI intervals for m2b_2t (with
# Student-t distribution) is much more narrow (confident) than that for the model b2b_2
```

e) Given your analysis, what conclusions do you draw about the relationship between tax rate and tax revenue? Do your conclusions support the original Laffer curve plot used in the editorial?

```
# From the analysis, it is easy to conclude that a polynomial regressor (degree 2) is a better
# predictor of tax_revenue than a linear regressor (degree 1).
# This is much in lines with the Laffer curve plot which also somewhat resembles a quadratic
# polynomial (degree 2)
# Even though with significantly high PSIS/WAIC values, the relationship between tax_revenue
# and tax_rate is certainly better explained by a degree two polynomial than a degree one
# linear curve.
```