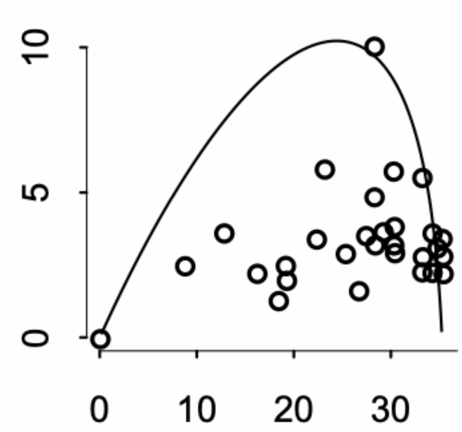
# SCI 2025: Homework 5

## Setup

The first two homework problems this week are from Chapter 7 of the textbook.

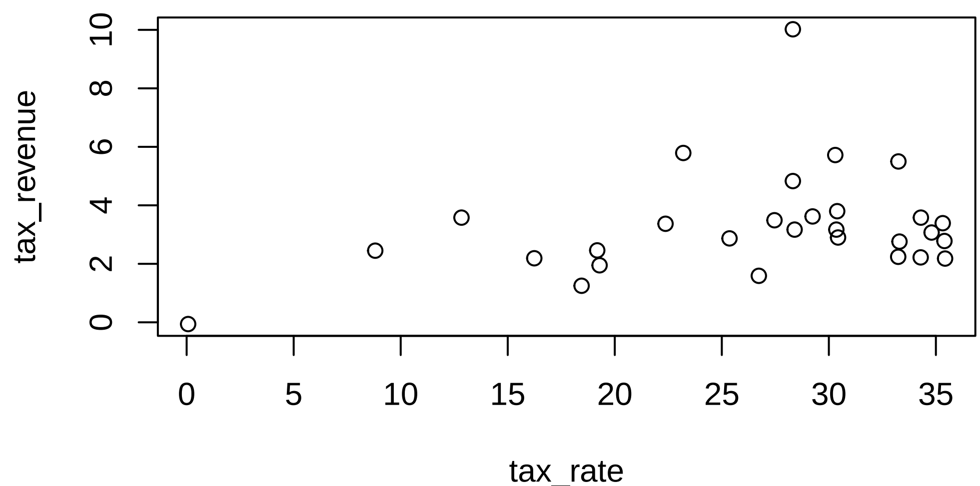
## Question 1: 7H1

In 2007, *The Wall Street Journal* published an editorial ("We're Number One, Alas") with a graph of corporate tax rates in 29 countries plotted against tax revenue. A badly fit curve was drawn in (reconstructed below), seemingly by hand, to make the argument that the relationship between tax rate and tax revenue increases and then declines, such that higher tax rates can actually produce less tax revenue.




I want you to actually fit a curve to these data, found in `data(Laffer)`. Consider models that use tax rate to predict tax revenue. Compare, using WAIC or PSIS, a straight-line model to any curved models you like. What do you conclude about the relationship between tax rate and tax revenue?

```
library(rethinking)
data(Laffer)

plot(tax_revenue ~ tax_rate, data = Laffer)
```




```
m1 <- quap(
    alist(
        tax_revenue ~ dnorm(mu, sigma),
        mu <- a +
        b * (tax_rate - mean(tax_rate)),
        a ~ dnorm(3, 4),
        b ~ dnorm(0, 0.1),
        sigma ~ dexp(1)
    ),
    data = Laffer
```

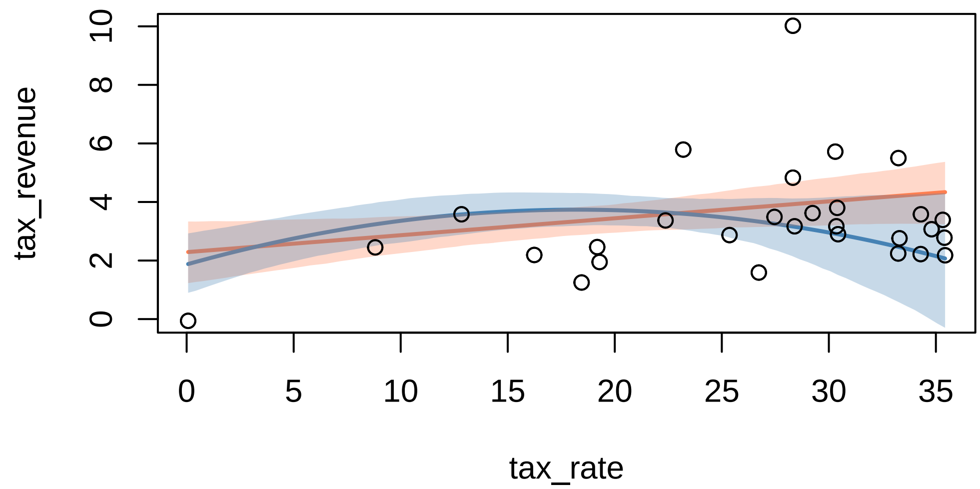
```r
)

precis(m1)
```

```
           mean         sd       5.5%     94.5%
a     3.30442861 0.30482599 2.8172578 3.7915994
b     0.05890947 0.03353172 0.0053193 0.1124996
sigma 1.64632381 0.20777497 1.3142593 1.9783883
```

```r
m2 <- quap(
    alist(
        tax_revenue ~ dnorm(mu, sigma),
        mu <- a +
        b * (tax_rate - mean(tax_rate)) +
        b2 * (tax_rate - mean(tax_rate))^2,
        a ~ dnorm(3, 4),
        b ~ dnorm(0, 0.1),
        b2 ~ dnorm(0, 0.1),
        sigma ~ dexp(1)
    ),
    data = Laffer
)

precis(m2)
```

```
            mean          sd        5.5%         94.5%
a      3.716721251 0.362527007  3.13733307  4.2961094270
b      0.004631741 0.043186378 -0.06438843  0.0736519136
b2    -0.005600083 0.002953547 -0.01032042 -0.0008797447
sigma  1.566555105 0.197841544  1.25036611  1.8827441027
```

```r
# plot predictions from both models

tax_rate_seq <- seq(min(Laffer$tax_rate), max(Laffer$tax_rate), length.out = 100)

mu_m1 <- link(m1, data = data.frame(tax_rate = tax_rate_seq))
mu_m2 <- link(m2, data = data.frame(tax_rate = tax_rate_seq))

# plot the predictions
plot(tax_revenue ~ tax_rate, data = Laffer)
lines(tax_rate_seq, colMeans(mu_m1), col = "coral", lwd = 2)
lines(tax_rate_seq, colMeans(mu_m2), col = "steelblue", lwd = 2)

# plot the uncertainty
shade(apply(mu_m1, 2, PI), tax_rate_seq,
 col = col.alpha("coral", 0.3))
shade(apply(mu_m2, 2, PI), tax_rate_seq,
 col = col.alpha("steelblue", 0.3))

# plot the data points
points(tax_revenue ~ tax_rate, data = Laffer)
```




```r
compare(m1, m2, func = PSIS)
```

```
       PSIS       SE    dPSIS      dSE     pPSIS     weight
m2 129.0241 29.46101 0.000000       NA  9.940347 0.92492266
m1 134.0464 32.77710 5.022383 3.965333 11.354141 0.07507734
```

```r
PSIS(m1, pointwise = TRUE)
```

```
       PSIS      lppd    penalty  std_err           k
1  5.303823 -2.651911 0.60878017 27.43141 0.524144855
2  3.035570 -1.517785 0.03392397 27.43141 0.351817266
3  3.455392 -1.727696 0.07064392 27.43141 0.330126162
4  3.027930 -1.513965 0.02543700 27.43141 0.315460961
5  2.962310 -1.481155 0.02016531 27.43141 0.199730121
6  3.246534 -1.623267 0.03065990 27.43141 0.198261913
7  3.943471 -1.971735 0.07379821 27.43141 0.276356114
8  4.066534 -2.033267 0.05019944 27.43141 0.086807867
```

```
9    2.911443  -1.455722 0.01768738 27.43141 -0.039651451
10   2.919709  -1.459854 0.01715984 27.43141 -0.119315833
11   5.881819  -2.940909 0.19392642 27.43141  0.333555718
12 30.622000 -15.311000 6.94268696 27.43141  1.627746849
13   4.868694  -2.434347 0.10834320 27.43141  0.204380680
14   3.679519  -1.839760 0.03467787 27.43141 -0.004660070
15   2.874283  -1.437141 0.01712232 27.43141 -0.029488236
16   2.891038  -1.445519 0.01681016 27.43141 -0.101114283
17   2.880593  -1.440297 0.01722258 27.43141 -0.008171197
18   2.903924  -1.451962 0.01774024 27.43141 -0.061088398
19   4.243777  -2.121888 0.08523757 27.43141  0.203781276
20   2.922573  -1.461287 0.01705556 27.43141 -0.192590953
21   3.029783  -1.514892 0.01854975 27.43141 -0.055538119
22   2.906416  -1.453208 0.01744528 27.43141 -0.078277200
23   2.974501  -1.487250 0.02013106 27.43141  0.012418737
24   3.103899  -1.551950 0.02507948 27.43141  0.097017629
25   3.240194  -1.620097 0.02792814 27.43141  0.175610646
26   3.748513  -1.874257 0.05289165 27.43141  0.274962741
27   3.864557  -1.932279 0.06640172 27.43141  0.264093409
28   4.028495  -2.014248 0.08748393 27.43141  0.277494935
29   3.345113  -1.672556 0.03897502 27.43141  0.213651965
```

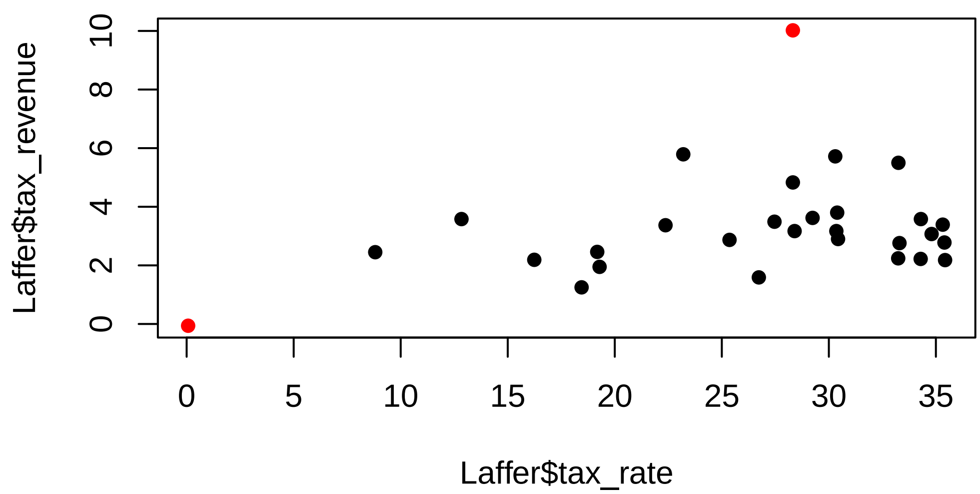
```r
PSIS(m2, pointwise = TRUE)
```

```
       PSIS      lppd    penalty std_err          k
1  4.118592 -2.059296 0.40079901 33.4103 0.56728919
2  3.106388 -1.553194 0.05377965 33.4103 0.36561995
3  3.332112 -1.666056 0.07026044 33.4103 0.41389622
4  3.199253 -1.599626 0.04638946 33.4103 0.15380664
5  3.209538 -1.604769 0.04333672 33.4103 0.07201807
```

```
6   3.788333  -1.894166 0.09167638 33.4103  0.29467664
7   4.884556  -2.442278 0.21698565 33.4103  0.55615441
8   4.848776  -2.424388 0.13224186 33.4103  0.51367768
9   2.828713  -1.414357 0.01697010 33.4103 -0.11018969
10  3.091676  -1.545838 0.02583439 33.4103  0.06495498
11  5.061489  -2.530745 0.21490724 33.4103  0.67452652
12 36.620217 -18.310109 9.82289903 33.4103  2.37887545
13  4.718953  -2.359476 0.10407490 33.4103  0.49752018
14  3.337822  -1.668911 0.03044802 33.4103  0.25725681
15  2.796211  -1.398106 0.01527789 33.4103 -0.12118424
16  2.893656  -1.446828 0.01673025 33.4103 -0.12745846
17  2.770995  -1.385498 0.01501782 33.4103  0.01400079
18  2.779815  -1.389908 0.01542166 33.4103 -0.03824941
19  4.660619  -2.330309 0.13372508 33.4103  0.54358463
20  2.868438  -1.434219 0.01610821 33.4103 -0.03976526
21  3.011870  -1.505935 0.01876830 33.4103  0.04148996
22  2.823693  -1.411846 0.01773730 33.4103  0.08869721
23  2.840967  -1.420484 0.01908699 33.4103  0.22833856
24  2.865013  -1.432506 0.01949503 33.4103  0.28263589
25  3.030324  -1.515162 0.02427983 33.4103  0.27676074
26  3.504328  -1.752164 0.04866387 33.4103  0.37761371
27  3.473130  -1.736565 0.05873391 33.4103  0.41321457
28  3.461051  -1.730525 0.07385313 33.4103  0.43470118
29  2.977563  -1.488781 0.02877360 33.4103  0.36019494
```

```
# color code the points by pareto k statistic
plot(Laffer$tax_rate, Laffer$tax_revenue,
 col = ifelse(
    PSIS(m2, pointwise = TRUE)$k > 0.7,
     "red",
```

```
      "black"), pch = 16)
```




## Question 2: 7H2

In the `Laffer` data, there is one country with a high tax revenue that is an outlier. Use PSIS and WAIC to measure the importance of this outlier in the models you fit in the previous problem. Then use robust regression with a Student's t distribution to revisit the curve fitting problem. How much does a curved relationship depend upon the outlier point?

```
WAIC(m2, pointwise = TRUE)
```

```
        WAIC      lppd    penalty  std_err
1   3.965729 -1.657608 0.32525647 25.64038
2   3.090515 -1.492015 0.05324243 25.64038
3   3.290353 -1.586264 0.05891288 25.64038
4   3.208038 -1.557038 0.04698118 25.64038
5   3.215429 -1.563513 0.04420152 25.64038
```

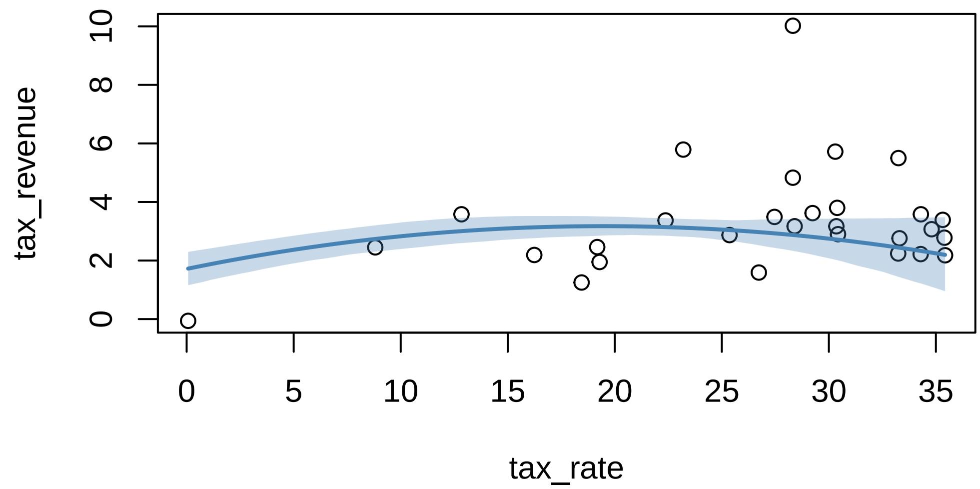
```
6   3.791172 -1.809616 0.08596987 25.64038
7   4.862558 -2.240890 0.19038879 25.64038
8   4.829668 -2.292247 0.12258662 25.64038
9   2.830868 -1.396151 0.01928264 25.64038
10  3.097284 -1.519994 0.02864732 25.64038
11  4.984448 -2.311791 0.18043258 25.64038
12 29.237711 -8.254144 6.36471168 25.64038
13  4.761622 -2.260666 0.12014531 25.64038
14  3.351024 -1.644567 0.03094561 25.64038
15  2.803659 -1.384371 0.01745858 25.64038
16  2.900508 -1.430582 0.01967159 25.64038
17  2.780226 -1.373479 0.01663441 25.64038
18  2.791082 -1.379119 0.01642197 25.64038
19  4.715694 -2.208886 0.14896091 25.64038
20  2.874757 -1.418500 0.01887889 25.64038
21  3.015304 -1.485747 0.02190535 25.64038
22  2.831947 -1.397502 0.01847109 25.64038
23  2.843299 -1.402221 0.01942817 25.64038
24  2.862377 -1.410330 0.02085825 25.64038
25  3.025829 -1.486384 0.02653060 25.64038
26  3.486142 -1.695042 0.04802925 25.64038
27  3.446970 -1.667931 0.05555361 25.64038
28  3.424146 -1.644909 0.06716385 25.64038
29  2.964967 -1.453502 0.02898139 25.64038
```

```r
# fit a robust regression model
m3 <- quap(
    alist(
        tax_revenue ~ dstudent(2, mu, sigma),
        mu <- a +
```

```r
        b * (tax_rate - mean(tax_rate)) +
        b2 * (tax_rate - mean(tax_rate))^2,
        a ~ dnorm(3, 4),
        b ~ dnorm(0, 0.1),
        b2 ~ dnorm(0, 0.1),
        sigma ~ dexp(1)
    ),
    data = Laffer
)

precis(m3)
```

```
              mean          sd         5.5%        94.5%
a      3.172481863 0.226417676  2.810622686  3.534341040
b      0.012931469 0.025074400 -0.027142265  0.053005202
b2    -0.003954157 0.001729148 -0.006717669 -0.001190645
sigma  0.715536231 0.150423672  0.475130151  0.955942312
```

```r
# plot the predictions from the robust regression model
mu_m3 <- link(m3, data = data.frame(tax_rate = tax_rate_seq))

plot(tax_revenue ~ tax_rate, data = Laffer)
lines(tax_rate_seq, colMeans(mu_m3),
 col = "steelblue", lwd = 2)
shade(apply(mu_m3, 2, PI), tax_rate_seq,
 col = col.alpha("steelblue", 0.3))
```




## Question 3

In machine learning, it is common to use cross-validation for model comparison and tuning of certain parameters. The simplest approach is the "train-test" split, where the data is split into a training set and a test set. The model is fit on the training set, and then the predictions are compared to the true values on the test set. It is typical to use around 70-80% of the data for the training set and the rest for the test set.

Here's how you can do a train-test split in R on the `Laffer` data:

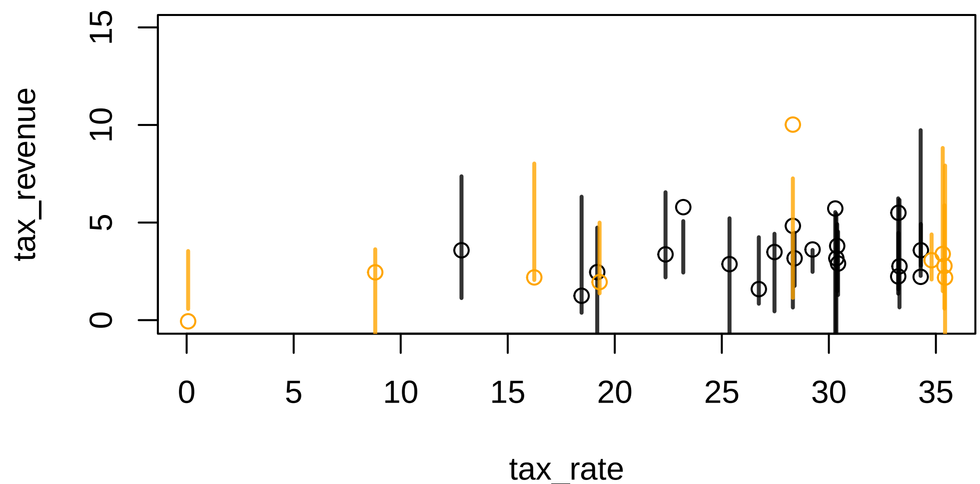
```
library(rethinking)
data(Laffer)

set.seed(123)
n <- nrow(Laffer)
train_idx <- sample(1:n, size = round(n * 0.7)) # random sample of 70% of the data
train_data <- Laffer[train_idx, ]
test_data <- Laffer[-train_idx, ]
```

Now, what I would like you to do is fit a model of your choice (informed by the results of the previous questions) to the training data only (`train_data`). First, make predictions for the *training data* and plot those predictions, as well as the true values as points. Then, make predictions for the *test data* and plot those predictions, as well as the true values as points. Your model predictions should be on the y-axis and the tax rate should be on the x-axis. Be sure to visualize uncertainty in your predictions.

```
m3_train <- quap(
    alist(
        tax_revenue ~ dstudent(2, mu, sigma),
        mu <- a + b * (tax_rate - mean(tax_rate)) + b2 * (tax_rate - mean(tax_rate))^2,
        a ~ dnorm(3, 4),
        b ~ dnorm(0, 0.1),
        b2 ~ dnorm(0, 0.1),
        sigma ~ dexp(1)
    ),
    data = train_data
)
```

```
mu_m3_train <- sim(m3_train, data = data.frame(tax_rate = train_data$tax_rate))

plot(tax_revenue ~ tax_rate, data = train_data, xlim = range(Laffer$tax_rate), ylim = ra

for (i in 1:nrow(train_data)) {
    lines(x=rep(train_data$tax_rate[i], 2), y=HPDI(mu_m3_train[i, ]),
     col = col.alpha("black", 0.8), lwd = 2)
}

# now make predictions for the test data
mu_m3_test <- sim(m3_train, data = data.frame(tax_rate = test_data$tax_rate))
```

```
points(tax_revenue ~ tax_rate, data = test_data, col = "orange")

for (i in 1:nrow(test_data)) {
    lines(x=rep(test_data$tax_rate[i], 2), y=HPDI(mu_m3_test[i, ]),
     col = col.alpha("orange", 0.8), lwd = 2)
}
```

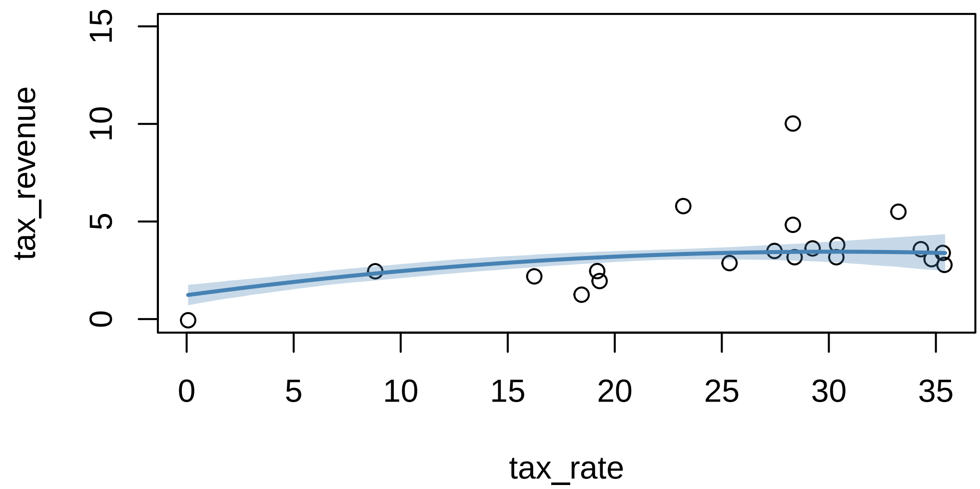



## Question 4

Now, repeat the procedure in Question 3, exactly, but *change the random seed* to some new number. I encourage you to do this multiple times. How much variability across seeds (different random splits) is there in: (a) the slope/curve relating tax rate to tax revenue? (b) the discrepancy between the training and test set predictions?

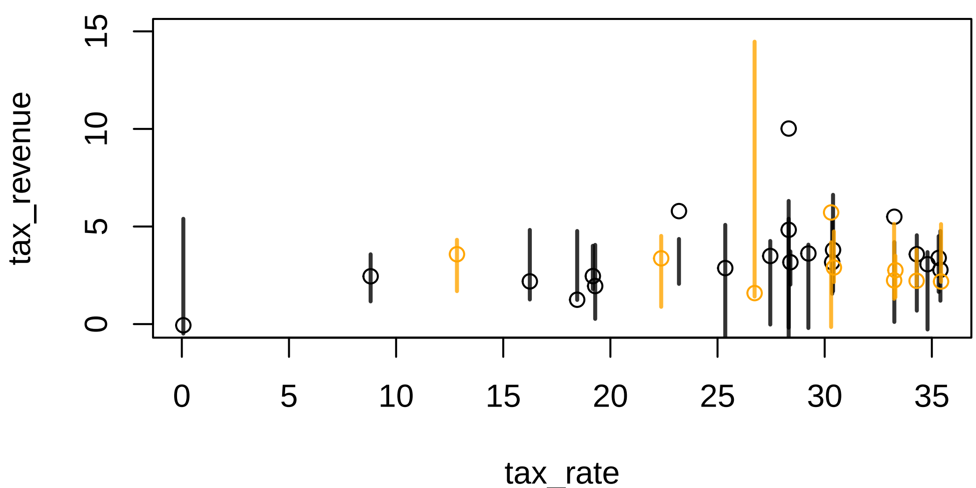
```
set.seed(3001)
n <- nrow(Laffer)
train_idx <- sample(1:n, size = round(n * 0.7)) # random sample of 70% of the data
```

```r
train_data <- Laffer[train_idx, ]
test_data <- Laffer[-train_idx, ]

m3_train <- quap(
    alist(
        tax_revenue ~ dstudent(2, mu, sigma),
        mu <- a + b * (tax_rate - mean(tax_rate)) + b2 * (tax_rate - mean(tax_rate))^2,
        a ~ dnorm(3, 4),
        b ~ dnorm(0, 0.1),
        b2 ~ dnorm(0, 0.1),
        sigma ~ dexp(1)
    ),
    data = train_data
)

# visualize the function relating tax rate to tax revenue

tax_rate_seq <- seq(min(Laffer$tax_rate), max(Laffer$tax_rate), length.out = 100)

mu_m3 <- link(m3_train, data = data.frame(tax_rate = tax_rate_seq))

plot(tax_revenue ~ tax_rate, data = train_data,
 xlim = range(Laffer$tax_rate),
  ylim = range(Laffer$tax_revenue)*1.5)

lines(tax_rate_seq, colMeans(mu_m3), col = "steelblue", lwd = 2)
shade(apply(mu_m3, 2, PI), tax_rate_seq,
 col = col.alpha("steelblue", 0.3))
```




```
# compare predictions for test vs train data

mu_m3_train <- sim(m3_train, data = data.frame(tax_rate = train_data$tax_rate))

plot(tax_revenue ~ tax_rate,
 data = train_data,
  xlim = range(Laffer$tax_rate),
   ylim = range(Laffer$tax_revenue)*1.5)

for (i in 1:nrow(train_data)) {
    lines(
        x=rep(train_data$tax_rate[i], 2),
         y=HPDI(mu_m3_train[i, ]),
          col = col.alpha("black", 0.8),
           lwd = 2)
}
```

```r
# now make predictions for the test data
mu_m3_test <- sim(m3_train, data = data.frame(tax_rate = test_data$tax_rate))

points(tax_revenue ~ tax_rate,
 data = test_data,
  col = "orange")

for (i in 1:nrow(test_data)) {
    lines(x=rep(test_data$tax_rate[i], 2), y=HPDI(mu_m3_test[i, ]),
     col = col.alpha("orange", 0.8), lwd = 2)
}
```




## Question 5

Based on what you have learned so far in this course, how do you imagine that model comparison via information criteria and/or cross-validation can support causal inference, and answering scientific questions? Where do you think it could go wrong?

- colliders will be preferred because they help us to predict the outcome, e.g., conditioning on hospitalization helps to predict the probably of having a heart attack, even though we have causality reversed