7. Chapter 7 Solutions

7E1. The criteria for this measure of "uncertainty" are:

(1) Continuity. The measure is not discrete, but it may be bounded. So it can have a minimum and maximum, but it must be continuous in between.
(2) Increasing with the number of events. When more distinct things can happen, and all else is equal, there is more uncertainty than when fewer things can happen.
(3) Additivity. This is hardest one to understand, for most people. Additivity is desirable because it means we can redefine event categories without changing the amount of uncertainty.

7E2. This is a simple calculation in R, very much like the example on page 192:

R code 7.1

```
# define probabilities of heads and tails
p <- c( 0.7 , 0.3 )

# compute entropy
-sum( p*log(p) )
```

```
[1] 0.6108643
```

7E3. Similar to the problem above, but now with four types of events:

R code 7.2

```
# define probabilities of sides
p <- c( 0.2 , 0.25 , 0.25 , 0.3 )

# compute entropy
-sum( p*log(p) )
```

```
[1] 1.376227
```

7E4. When events are impossible (have probability of zero), they just fall out of the calculation:

R code 7.3

```
# define probabilities of sides
p <- c( 1/3 , 1/3 , 1/3 )

# compute entropy
-sum( p*log(p) )
```

```
[1] 1.098612
```

7M1. The definition of AIC is:

$$\begin{aligned}\text{AIC} &= -2 \log \Pr(\text{data}|\text{MAP estimates}) + 2p \\ &= D_{\text{train}} + 2p\end{aligned}$$

where p is the number of parameters in the model. WAIC is a notational mess, but once you understand each part, then the notation makes sense:

$$\begin{aligned} \text{WAIC} &= -2(\text{lppd} - p_{\text{WAIC}}) \\ &= -2\left(\sum_i \log \Pr(y_i) - \sum_i V(y_i)\right) \end{aligned}$$

where $\Pr(y_i)$ is the likelihood of observation i, averaging over the posterior distribution, and $V(y_i)$ is the variance in the log-likelihood of observation i, taking the variance over the posterior distribution.

Comparing these definitions, each has a term that expresses the fit to the training sample, as well as a term that expresses some penalty for flexibility in fitting to the sample. For AIC, the fit term is simply the "plug-in deviance," D_{train}, and the penalty term is twice the number of parameters, p. For WAIC, the fit term is a fully Bayesian log-likelihood that averages over the posterior prediction for each observation separately. This is lppd. The penalty term is the sum of the variances of each log-likelihood.

From more general to less general: WAIC, AIC. When the posterior predictive mean is a good representation of the posterior predictive distribution, and the priors are effectively flat or overwhelmed by the amount of data, then WAIC and AIC will tend to agree.

7M2. Model selection means ranking models by information criteria (or any other criteria) and choosing the highest ranked model. Model averaging is instead weighting each model by its relative distance from the best model and creating a posterior predictive distribution comprising predictions from each model in proportion to these weights, a prediction "ensemble." Model selection discards information about model uncertainty. Model averaging retains some of that information, but it still discards some information about model uncertainty, because it compresses the full information about the set of models into a single predictive distribution. Since the full set of ranks and information criteria values cannot be recovered from the averaged predictive distribution, some of that information has been lost. In other words, different model comparison sets with different ranks and information criteria values can produce nearly identical prediction ensembles. But those different model comparison sets may be different in other important ways. For example, inspecting the full model comparison set, with the structure of each model, often reveals why some models fit better than others.

7M3. When one model is fit to fewer (or different) observations, it is being judged on a different target than the other models. If fewer observations are used, the model will usually appear to perform better, because the deviance will smaller due to having less to predict. Less to predict means less prediction error, and deviance (as well as information criteria) is a kind of accumulated error. We have to be careful about this, because most of R's black-box regression functions like `lm`, `glm`, and `glmer` will automatically and silently drop incomplete cases, reducing the number of observations the model is fit to. This is bad behavior for scientific software, but unfortunately it is the norm.

7M4. As a prior becomes more concentrated around particular parameter values, the model becomes less flexible in fitting the sample. One way to remember this is to think of the prior as representing previous learning from previous observations. So a more concentrated, or peaked, prior represents more previous data. As the model becomes less flexible, the effective number of parameters declines.

To perform some simple experiments to demonstrate this, try out this code, changing the value for `sigma` in the data list. The smaller `sigma`, the more concentrated the prior and the smaller the effective number of parameters should be.

R code
7.4

```
y <- rnorm(10) # execute just once, to get data

# repeat this, changing sigma each time
m <- quap(
    alist(
        y ~ dnorm(mu,1),
        mu ~ dnorm(0,sigma)
    ), data=list(y=y,sigma=10) )
WAIC(m)
```

7M5. Informative priors reduce overfitting by reducing the sensitivity of a model to a sample. Some of the information in a sample is *irregular*, not a recurring feature of the process of interest.

7M6. If a prior is overly informative, then even regular features of a sample will not be learned by a model. In this case, the model may underfit the sample. In case this sounds like a terrifying balancing act, in practice there is usually a broad family of priors which achieve practically indistinguishable estimates from the same sample. The last "hard" practice problem for this chapter provides an example.

7H1/7H2. It's probably obvious from the prompt that the original curve was hand-drawn and not the result of any fitting procedure. And you can see there is a high outlier point, while the rest of the points show a general increase. But let's take this seriously and polish our modeling skills.

Lots of combinations of models will produce the same general inference. I'll start with comparing linear to polynomial models. Then I'll try a spline. Then I'll consider a robust regression to cope better with the outlier.

Here is a linear fit, together with a quadratic and cubic fit:

R code
7.5

```
library(rethinking)
data(Laffer)
d <- Laffer
d$T <- standardize( d$tax_rate )
d$R <- standardize( d$tax_revenue )

# linear model
m7H1a <- quap(
    alist(
        R ~ dnorm( mu , sigma ),
        mu <- a + b*T,
        a ~ dnorm( 0 , 0.2 ),
        b ~ dnorm( 0 , 0.5 ),
        sigma ~ dexp(1)
    ) , data=d )

# quadratic model
m7H1b <- quap(
    alist(
        R ~ dnorm( mu , sigma ),
        mu <- a + b*T + b2*T^2,
        a ~ dnorm( 0 , 0.2 ),
        c(b,b2) ~ dnorm( 0 , 0.5 ),
        sigma ~ dexp(1)
    ) , data=d )

# cubic model
m7H1c <- quap(
    alist(
        R ~ dnorm( mu , sigma ),
        mu <- a + b*T + b2*T^2 + b3*T^3,
        a ~ dnorm( 0 , 0.2 ),
        c(b,b2,b3) ~ dnorm( 0 , 0.5 ),
        sigma ~ dexp(1)
    ) , data=d )
```

As a first check, let's see how these models compare on purely predictive criteria, like PSIS:

R code 7.6

```
compare( m7H1a , m7H1b , m7H1c , func=PSIS )
```

```
Some Pareto k values are very high (>1)
      PSIS    SE dPSIS  dSE pPSIS weight
m7H1a 93.3 26.88   0.0   NA   8.3   0.78
m7H1b 96.5 32.37   3.2 5.75  10.9   0.16
m7H1c 98.3 31.82   5.0 5.25  11.8   0.06
```

Not much support for the wiggly models over the linear one. But PSIS also warns about high leverage points. You can probably guess from scatterplot which point is causing the problem. If you peek at the pointwise *k* values, you'll see:

R code 7.7

```
PSISk(m7H1a)
```

```
 [1]  0.61  0.46  0.44  0.07 -0.03  0.09  0.28  0.29 -0.07  0.04  0.43  1.85
[13]  0.34  0.34 -0.11 -0.07 -0.13 -0.08  0.31 -0.01 -0.02 -0.02  0.05  0.04
[25]  0.07  0.26  0.28  0.23  0.10
```

Point 12 is much over 1. That's the nation with the very high tax revenue value. I think it's Norway, actually, and it results from an accounting trick involving oil revenue.

It'll be useful to plot the posterior predictions of each model.

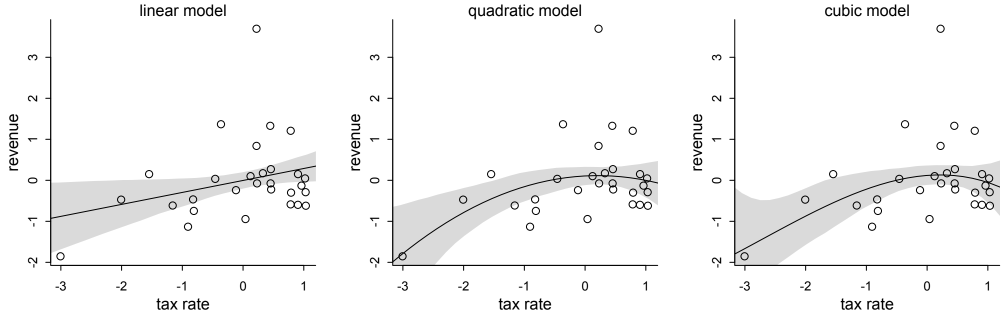
R code 7.8

```
T_seq <- seq( from=-3.2 , to=1.2 , length.out=30 )
la <- link( m7H1a , data=list(T=T_seq) )
lb <- link( m7H1b , data=list(T=T_seq) )
lc <- link( m7H1c , data=list(T=T_seq) )

plot( d$T , d$R , xlab="tax rate" , ylab="revenue" )
mtext( "linear model" )
lines( T_seq , colMeans(la) )
shade( apply( la , 2 , PI ) , T_seq )

plot( d$T , d$R , xlab="tax rate" , ylab="revenue" )
mtext( "quadratic model" )
lines( T_seq , colMeans(lb) )
shade( apply( lb , 2 , PI ) , T_seq )
```

```
plot( d$T , d$R , xlab="tax rate" , ylab="revenue" )
mtext( "cubic model" )
lines( T_seq , colMeans(lc) )
shade( apply( lc , 2 , PI ) , T_seq )
```



The polynomial models do bend the right way, but they aren't bending much. If anything, they offer stronger evidence that more tax produces more revenue, not that very high tax reduces revenue.

For the sake of the exercise, let's consider a basis spline. Refer back to Chapter 4, the last section, if you've forgotten splines.

R code 7.9

```
num_knots <- 15
knot_list <- quantile( d$T , probs=seq(0,1,length.out=num_knots) )

library(splines)
B <- bs(d$T,
    knots=knot_list[-c(1,num_knots)] ,
    degree=3 , intercept=TRUE )

m7H1s <- quap(
    alist(
        R ~ dnorm( mu , sigma ) ,
        mu <- a + B %*% w ,
        a ~ dnorm(0,1),
        w ~ dnorm(0,1),
        sigma ~ dexp(1)
    ), data=list( R=d$R , B=B ) ,
    start=list( w=rep( 0 , ncol(B) ) ) )
```

The coefficients aren't interpretable, but let's compare to the previous models:

R code 7.10

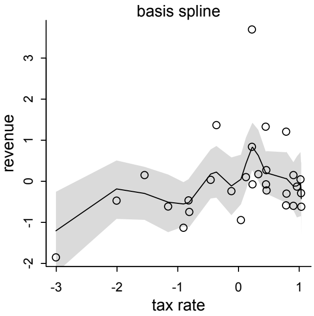
```
compare( m7H1a , m7H1b , m7H1c , m7H1s , func=PSIS )
```

```
Some Pareto k values are very high (>1)
       PSIS    SE dPSIS  dSE pPSIS weight
m7H1a  93.1 26.81   0.0   NA   8.2   0.60
m7H1c  95.1 28.45   2.1 2.57  10.2   0.21
m7H1b  95.4 30.30   2.4 3.74  10.3   0.18
m7H1s 102.4 31.75   9.3 6.06  17.7   0.01
```

Terrible. What do the predictions look like?

R code 7.11

```
mu <- link( m7H1s )
mu_PI <- apply( mu , 2 , PI )
plot( d$T , d$R , xlab="tax rate" , ylab="revenue" )
mtext( "basis spline" )
o <- order( d$T )
lines( d$T[o] , colMeans(mu)[o] )
shade( mu_PI[,o] , d$T[o] )
```



If you really want to interpret that as support for the curved relationship, good luck. You might try more rigid splines, with lower polynomial order, fewer knots, and tighter weights, to see if this can be made more sensible. But clearly this is a curve-fitting exercise, not science, at this point.

For the Student-t model, we just need to replace the normal likelihood. I'll use $\nu = 2$, so the tails are satisfyingly thick.

R code 7.12

```
# linear model with student-t
m7H1d <- quap(
    alist(
        R ~ dstudent( 2 , mu , sigma ),
        mu <- a + b*T,
        a ~ dnorm( 0 , 0.2 ),
        b ~ dnorm( 0 , 0.5 ),
        sigma ~ dexp(1)
    ) , data=d )
```

Comparing the previous models:

R code 7.13

```
compare( m7H1a , m7H1b , m7H1c , m7H1s , m7H1d , func=PSIS )
```

```
Some Pareto k values are very high (>1)
       PSIS    SE dPSIS   dSE pPSIS weight
m7H1d  75.0 13.70   0.0    NA   4.0      1
m7H1b  89.1 25.23  14.0 16.78   7.1      0
m7H1a  98.8 32.32  23.8 23.44  11.0      0
m7H1c 103.2 36.99  28.1 27.91  14.2      0
m7H1s 105.5 33.20  30.5 23.88  19.3      0
```

Oh PSIS likes this one. It also doesn't itself throw any Pareto-*k* warnings. If you like, try to polynomial models with the Student-t likelihood. Overall, there is no obvious support for a strongly increasing-then-decreasing relationship between tax rate and tax revenue.

7H3. To compute the entropies, we just need a function to compute the entropy. Information entropy, as defined in lecture and the book, is simply:

$$H(p) = -\sum_i p_i \log(p_i)$$

where p is a vector of probabilities summing to 1. In R code this would look like:

R code 7.14

```
H <- function(p) -sum(p*log(p))
```

I'll make a list of the birb distributions and then push each through the function above.

R code 7.15

```
IB <- list()
IB[[1]] <- c( 0.2 , 0.2 , 0.2 , 0.2 , 0.2 )
IB[[2]] <- c( 0.8 , 0.1 , 0.05 , 0.025 , 0.025 )
IB[[3]] <- c( 0.05 , 0.15 , 0.7 , 0.05 , 0.05 )
sapply( IB , H )
```

```
[1] 1.6094379 0.7430039 0.9836003
```

The first island has the largest entropy, followed by the third, and then the second in last place. Why is this? Entropy is a measure of the evenness of a distribution. The first islands has the most even distribution of birbs. This means you wouldn't be very surprised by any particular birb. The second island, in contrast, has a very uneven distribution of birbs. If you saw any birb other than the first species, it would be surprising.

Now we need K-L distance, so let's write a function for it:

R code 7.16

```
DKL <- function(p,q) sum( p*(log(p)-log(q)) )
```

This is the distance from q to p, regarding p as true and q as the model. Now to use each island as a model of the others, we need to consider the different ordered pairings. I'll just make a matrix and loop over rows and columns:

R code 7.17

```
Dm <- matrix( NA , nrow=3 , ncol=3 )
for ( i in 1:3 ) for ( j in 1:3 ) Dm[i,j] <- DKL( IB[[j]] , IB[[i]] )
round( Dm , 2 )
```

```
     [,1] [,2] [,3]
[1,] 0.00 0.87 0.63
[2,] 0.97 0.00 1.84
[3,] 0.64 2.01 0.00
```

The way to read this is each row as a model and each column as a true distribution. So the first island, the first row, has the smaller distances to the other islands. This makes sense, since it has the highest entropy. Why does that give it a shorter distance to the other islands? Because it is less surprised by the other islands, due to its high entropy.

7H4. I won't repeat the models here. They are in the text. Model `m6.9` contains both marriage status and age. Model `m6.10` contains only age. Model `m6.9` produces a confounded inference about the relationship between age and happiness, due to opening a collider path. To compare these models using WAIC:

R code 7.18

```
compare( m6.9 , m6.10 )
```

```
        WAIC pWAIC dWAIC weight    SE  dSE
m6.9  2714.0   3.7   0.0      1 37.54   NA
m6.10 3101.9   2.3 387.9      0 27.74 35.4
```

The model that produces the invalid inference, `m6.9`, is expected to predict much better. And it would. This is because the collider path does convey actual association. We simply end up mistaken about the causal inference. We should not use WAIC (or LOO) to choose among models, unless we have some clear sense of the causal model. These criteria will happily favor confounded models.

7H5. These are the models:

R code 7.19

```
library(rethinking)
data(foxes)
d <- foxes
d$W <- standardize(d$weight)
d$A <- standardize(d$area)
d$F <- standardize(d$avgfood)
d$G <- standardize(d$groupsize)

m1 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + bF*F + bG*G + bA*A,
        a ~ dnorm(0,0.2),
        c(bF,bG,bA) ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ), data=d )
m2 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + bF*F + bG*G,
        a ~ dnorm(0,0.2),
        c(bF,bG) ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ), data=d )
m3 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + bG*G + bA*A,
        a ~ dnorm(0,0.2),
        c(bG,bA) ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ), data=d )
m4 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + bF*F,
        a ~ dnorm(0,0.2),
        bF ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ), data=d )
m5 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + bA*A,
        a ~ dnorm(0,0.2),
        bA ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ), data=d )
```

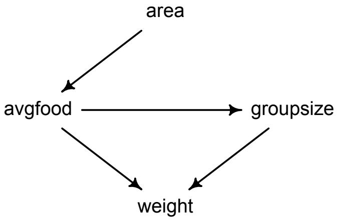
Comparing with WAIC:

R code 7.20

```
compare( m1 , m2 , m3 , m4 , m5 )
```

```
    WAIC pWAIC dWAIC weight    SE  dSE
m1 322.9   4.7   0.0   0.47 16.28   NA
m3 323.9   3.7   1.0   0.28 15.68 2.90
m2 324.1   3.9   1.2   0.25 16.14 3.60
m4 333.4   2.4  10.6   0.00 13.79 7.19
m5 333.7   2.7  10.8   0.00 13.79 7.24
```

To remind you, the DAG from the original problem is:



Notice that the top three models are m1, m3, and m2. They have very similar WAIC values. The differences are small and smaller in all cases than the standard error of the difference. WAIC sees these models are tied. This makes sense, given the DAG, because as long as a model has groupsize in it, we can include either avgfood or area or both and get the same inferences. Another way to think of this is that the influence of good, adjusting for group size, is (according to the DAG) the same as the influence of area, adjusting for group size, because the influence of area is routed entirely through food and group size. There are no backdoor paths.

What about the other two models, m4 and m5? These models are tied with one another, and both omit group size. Again, the influence of area passes entirely through food. So including only food or only area should produce the same inference—the total causal influence of area (or food) is just about zero. That's indeed what the posterior distributions suggest:

R code 7.21

```
coeftab(m4,m5)
```

```
      m4      m5
a          0       0
bF     -0.02      NA
sigma   0.99    0.99
bA        NA    0.02
nobs     116     116
```