Bayesian Test Quantities

Usually Bayesians like Andrew Gelman, Richard McElreath, and Jim Savage don't use the Kalmogorov-Smirnov test to determine goodness of fit. In *Bayesian Data Analysis, 3rd ed., the "Test Quantity" is introduced and contrasted with the more familiar "Test Statistic."

A test statistic is made using fixed parameters, but a test quantity results from all possible values of the parameters, weighted by their probabilities. That is, the posterior predictive distribution is used to find the test quantities.

The recommendation is for the analyst to choose test quantities that make sense based on the overall goals of the statistical inquiry at hand. Below we do that with the maximum breach size per year. Before using the model to predict the future, a check determines whether a data-based test quantity is similar to test quantities from simulated data drawn from the posterior predictive distribution.

Finding the Posterior Predictive Distribution

The data is loaded as before.

```
library(dplyr)
source("prep.R")
dd <- prep.data("~/data/privacyrights.org/select-all.csv")
d <- tbl_df(dd$prepped)
```

A new library, *lubridate*, facilitates the task of working with the dates in the data set. Again, if a library *foo* does not load, it can be installed via `install.packages("foo")`. The code below creates a `t` column in the data frame to match the `t` variable used by Edwards et al. in their unorthodox time series model. (Most time series models treat time as the unusual, one-way autocorrelated random variable it is, not as an unqualified predictor variable in a linear regression.)

```
library(lubridate)
d <- d %>% mutate(date=mdy(Date.Made.Public))
```

Then by using Richard McElreath's *rethinking* package, it's easy to use the powerful Stan probabilistic modeling tool via `map2stan` or the quick alternative `map`, which requires no C++ compilation step.

```
library(rethinking)
m.data <- data.frame(
       y=as.numeric(d$Total.Records),
       t=as.numeric(difftime(d$date, d$date[1], unit="weeks") / 52.25),
       kind=as.factor(d$Type.of.breach))

(m <- map(
     alist(
       log(y) ~ dnorm(mu, sigma),
       mu <- a + b * t,
       sigma ~ dunif(0, 50),
       a ~ dnorm(0, 10),
       b ~ dnorm(0, 10)),
     data=m.data))
```

```
##
```

```
## Maximum a posteriori (MAP) model fit
##
## Formula:
## log(y) ~ dnorm(mu, sigma)
## mu <- a + b * t
## sigma ~ dunif(0, 50)
## a ~ dnorm(0, 10)
## b ~ dnorm(0, 10)
##
## MAP values:
##       sigma           a           b
## 2.923758532 7.954853611 0.008025113
##
## Log-likelihood: -5855.76
```

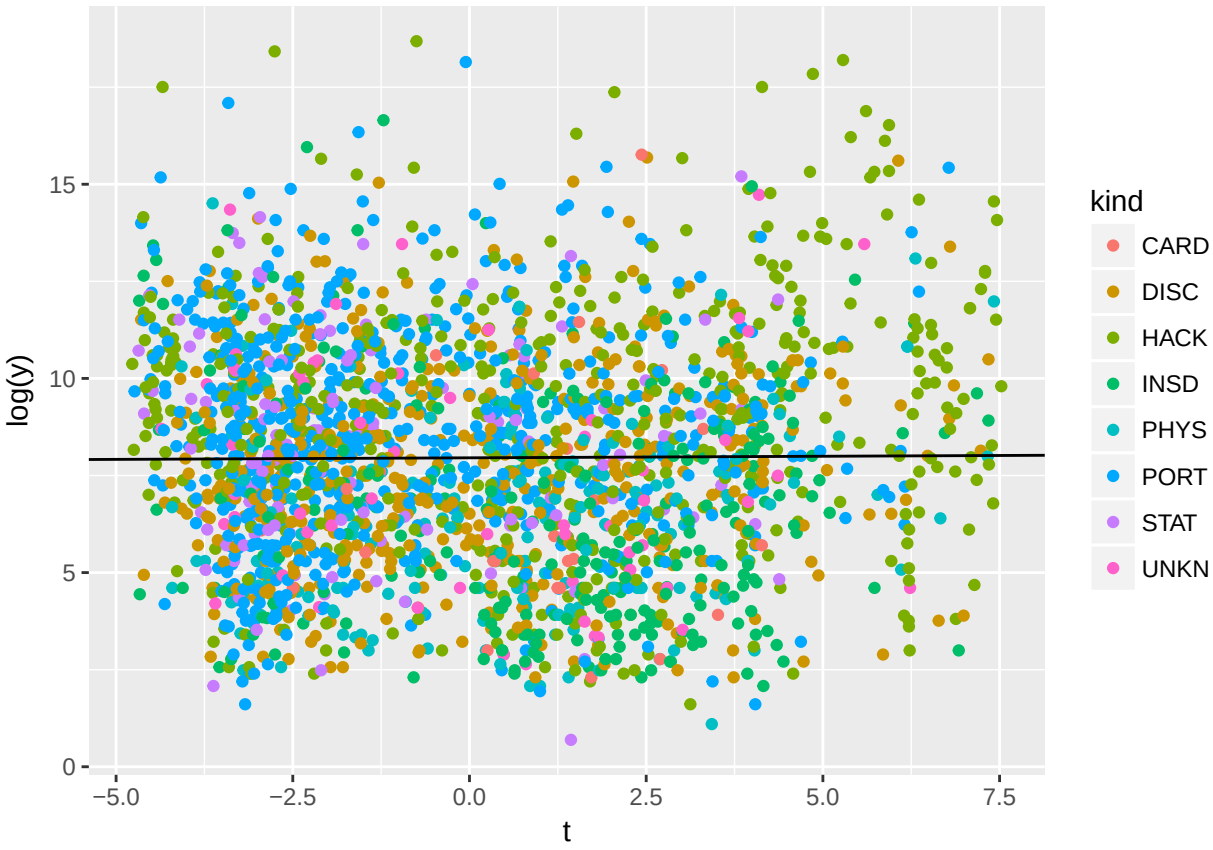
```
precis(m)
```

```
##       Mean StdDev  5.5% 94.5%
## sigma 2.92   0.04  2.86  2.99
## a     7.95   0.06  7.86  8.05
## b     0.01   0.02 -0.03  0.04
```

What does that look like?

```
library(ggplot2)

ggplot(m.data, aes(t, log(y), color=kind)) +
  geom_point() +
  geom_abline(intercept=coef(m)["a"], slope=coef(m)["b"])
```



Noisy and flat, like the results in Edwards et al.

Next, we have to figure out what questions we're interested in answering with this model as a tool for understanding the data.

Test Quantities

todo