BAYESIAN STATISTICS

CLASS 4

REVIEW OF BASICS

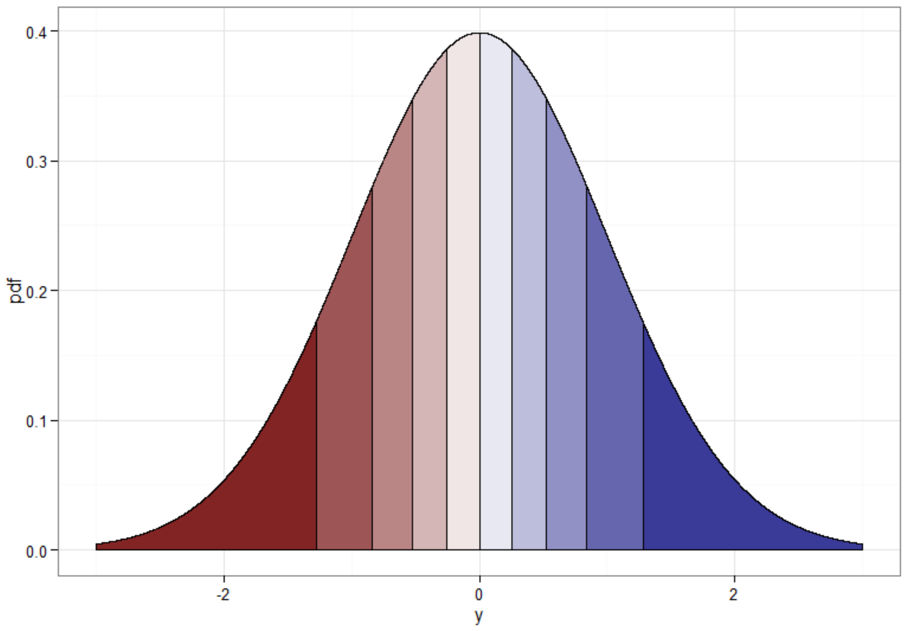
- In Bayesian Statistics, we use posterior distributions of the parameters to make inferences and create quantities of interest
- Need to define a distribution for the data that makes sense (sampling distribution: $P(Y|\theta)$)
- Based on the sampling distribution, choose an appropriate prior distribution for the parameter(s) (prior distribution: $p(\theta)$)
- These two distributions are put together (by Bayes theorem) to get the posterior distribution:

$$P(\theta|Y) = \frac{P(Y|\theta)P(\theta)}{\int P(Y|\theta)p(\theta)d\theta} \qquad \text{or} \qquad P(\theta|Y) \propto P(Y|\theta)P(\theta)$$

COMMON SAMPLING DISTRIBUTIONS WITH PRIORS

Sampling distribution: Binomial(n,p)
Prior: p~Beta(a,b) (a=1, b=1 for noniformative)

Sampling distribution: Normal(μ,σ)
Prior: μ~Normal(0,100) σ^2 ~ Inv-Gamma(0.001,0.001)



Sampling distribution: Poisson(λ)
Prior: λ~Gamma(a,b) (a=0.001, b=0.001 for noniformative)

STAN FILE

- Data
- Parameter
- Model

DATA

```
Data {
int y;
Real y;

}
```

DATA

Data {

Int y; Integers (takes on whole numbers)

Real y;

}

DATA

Data {

Int y; Integers (takes on whole numbers)

Real y; Real valued (numbers may take on values in an interval)

}

DATA

```
Data {
Int <lower=0, upper=1> y;
Real <lower=0, upper=1> y;

}
```

You can also indicate lower values and upper values for data (will give an error if someone tries inputting values that go beyond the limit).

DATA

```
Data {
Int <lower=0> n;
Real y[n];
Vector [n] y;

}
```

When your data is a vector (more than one observation), there are two ways to specify this.

DATA

```
Data {
Int <lower=0> n;
Int <lower=0> m;
Real y[n,m];
matrix [n,m] y;

}
```

When your data is a matrix (for example a dataframe), there are two ways to specify this. This data frame has n rows and m columns.

DATA

STAN file

```
data{
  int <lower=0> n;
  vector[n] y;
  matrix[n,5] x;
}
```

R code

```
regress.dat=list(n=nrow(x),x=x,y=ameshousing$Log_Price)
```

These two have to match up. Notice that both contain:
n, y and x (all with matching dimensions!)

PARAMETERS

```
parameters{
  real alpha;
  vector[5] beta;
  real<lower=0> sigma;
}
```

This is where you define ALL your parameters!! You can define them as just one number, a vector of numbers or a dataframe (same notation that was used in the "Data" section)

MODEL

```
model {
y ~normal(alpha + x*beta,sigma);
}
```

This is the main part of the problem (both previous sections were just basically setting things up). Here you should define the model and prior distributions. If you do not define a prior distribution, STAN will define an EXTREMELY vague prior over all possible values (as defined above).

AMES HOUSING DATA EXAMPLE

```
data{
  int <lower=0> n;
  vector[n] y;
  matrix[n,5] x;
}
parameters{
  real alpha;
  vector[5] beta;
  real<lower=0> tau;
}
```

```
transformed parameters{
real sigma;
sigma=sqrt(1/tau);
}
model {
alpha~normal(0,100);
beta~normal(0,100);
tau ~ gamma(0.001, 0.001);
y ~normal(alpha + x*beta,sigma);
}
```

AMES HOUSING DATA EXAMPLE

```
data{
  int <lower=0> n;
  vector[n] y;
  matrix[n,5] x;
}
parameters{
  real alpha;
  vector[5] beta;
  real<lower=0> tau;
}
```

```
transformed parameters{
real sigma;
sigma=sqrt(1/tau);
}
model {
alpha~normal(0,100);
beta~normal(0,100);
tau ~ gamma(0.001, 0.001);
y ~normal(alpha + x*beta,sigma);
}
```

AMES HOUSING DATA EXAMPLE

```
data{
  int <lower=0> n;
  vector[n] y;
  matrix[n,5] x;
}
parameters{
  real alpha;
  vector[5] beta;
  real<lower=0> tau;
}
```

```
transformed parameters{
real sigma;
sigma=sqrt(1/tau);
}
model {
alpha~normal(0,100);
beta~normal(0,100);
tau ~ gamma(0.001, 0.001);
y ~normal(alpha + x*beta,sigma);
}
```

VARIANCE IS AN INVERSE GAMMA!

ORIGINAL

```
> print(regress.stan)
Inference for Stan model: ameshousing.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

          mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
alpha    11.79    0.00 0.01  11.78  11.79  11.79  11.80  11.81  6588    1
beta[1]   0.12    0.00 0.01   0.10   0.11   0.12   0.12   0.14  5907    1
beta[2]   0.09    0.00 0.01   0.07   0.08   0.09   0.10   0.11  5643    1
beta[3]   0.06    0.00 0.01   0.04   0.05   0.06   0.06   0.08  5779    1
beta[4]   0.03    0.00 0.01   0.01   0.02   0.03   0.04   0.05  6864    1
beta[5]  -0.09    0.00 0.01  -0.10  -0.09  -0.09  -0.08  -0.07  5316    1
sigma     0.14    0.00 0.01   0.13   0.14   0.14   0.14   0.15  5456    1
lp__    440.42    0.05 1.89 435.92 439.39 440.76 441.80 443.12  1744    1
```

NEW

```
> print(regress2.stan)
Inference for Stan model: ameshousing2.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

          mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
alpha    11.79    0.00 0.01  11.77  11.79  11.79  11.80  11.81  7243    1
beta[1]   0.12    0.00 0.01   0.10   0.11   0.12   0.12   0.14  5759    1
beta[2]   0.09    0.00 0.01   0.07   0.08   0.09   0.10   0.11  4689    1
beta[3]   0.06    0.00 0.01   0.04   0.05   0.06   0.06   0.08  5365    1
beta[4]   0.03    0.00 0.01   0.01   0.02   0.03   0.03   0.05  6308    1
beta[5]  -0.09    0.00 0.01  -0.10  -0.09  -0.09  -0.08  -0.07  5255    1
tau      52.18    0.06 4.27  44.25  49.18  52.01  55.11  60.74  5682    1
sigma     0.14    0.00 0.01   0.13   0.13   0.14   0.14   0.15  5621    1
lp__    442.40    0.05 1.88 437.94 441.38 442.76 443.79 445.03  1613    1
```

CONJUGACY

- Some individuals prefer to have models with conjugacy:
  - Defining a prior that when combined with the data will produce a posterior distribution in the same family
    - For example:
    - If your data is binomial, defining a beta prior will result in a posterior that is also a beta distribution (however, parameters are "updated")
    - If your data is Poisson, defining a Gamma distribution on the mean will produce a posterior distribution that is also Gamma

POINT ESTIMATES

- Most common “point estimates” of the parameters are the mean of the posterior distribution or the median of the posterior distribution
  - The mean is the estimate under a “squared error loss”
  - The median is the estimate under an “absolute error loss”
  - There are other loss functions that will result in different point estimates, but these two are by far the most common

LOGISTIC REGRESSION

- The response variable is binary (i.e. a Bernoulli trial)
- The set up is similar to linear regression except when we specify the model, we will use a Bernoulli distribution instead of a normal distribution
- Let's take a look using the Titanic data set:
  - The response variable is Y which is whether or not they survived (0=not survived, 1=survived)
  - X1=Sex (Male=0)
  - X2=age
  - X3=Fare
  - X4=X1*X2

```
data{
  int <lower=0> n;
  int <lower=0,upper=1> y[n];
  vector [n] x1;
  vector [n] x2;
  vector [n] x3;
  vector [n] x4;
}
parameters{
  real alpha;
  real beta1;
  real beta2;
  real beta3;
  real beta4;
}
model {
alpha~normal(0,100);
beta1~normal(0,100);
beta2~normal(0,100);
beta3~normal(0,100);
beta4~normal(0,100);
y ~bernoulli_logit(alpha+beta1*x1 + beta2*x2 + beta3*x3+ beta4*x4);
}
```

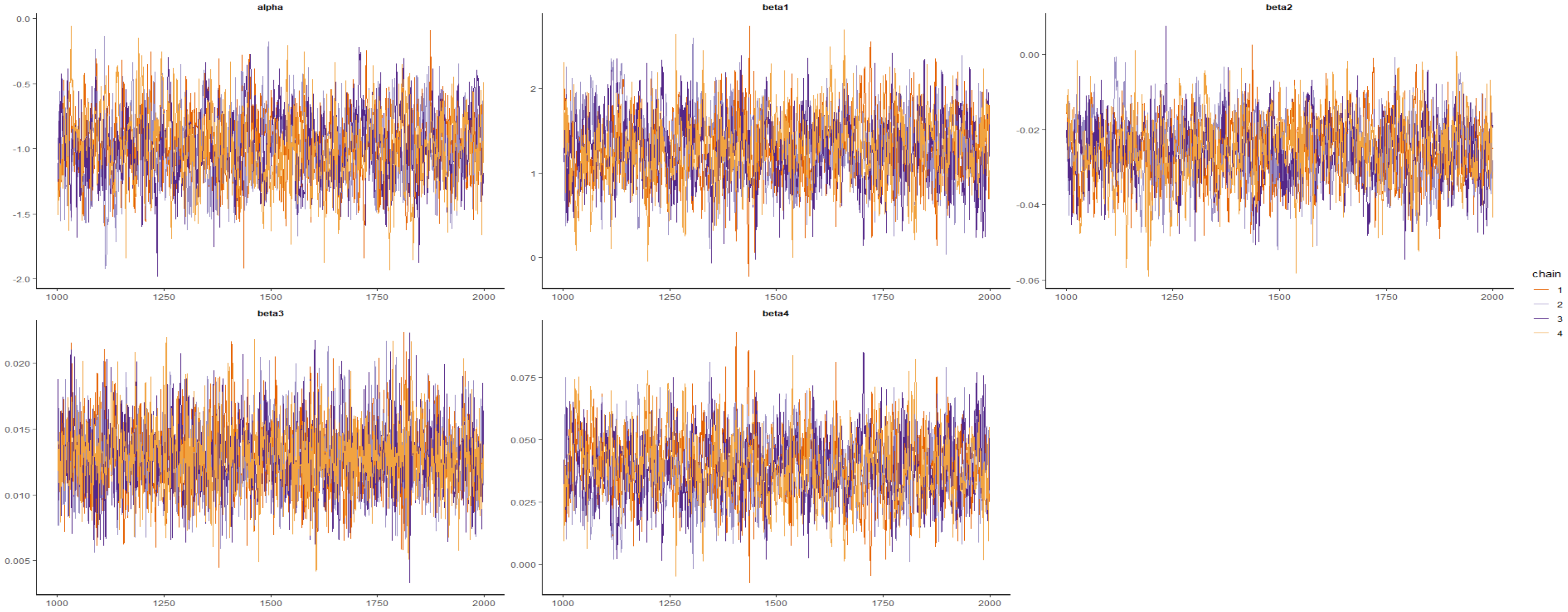
R CODE

```
library(titanic)
##Get rid of NA's
new.dat=titanic_train[complete.cases(titanic_train),]

###Get variables
y=new.dat$Survived
x1=ifelse(new.dat$Sex=="male",0,1)
x2=new.dat$Age
x3=new.dat$Fare
x4=x1*x2


###Put data together
titanic.dat<-list(n=length(y),y=y,x1=x1,x2=x2,x3=x3,x4=x4)
##Run STAN
titanic.stan<-stan(file='Q:\\My Drive\\Bayesian\\Code\\titanic.stan',data=titanic.dat,seed=03786)
```

TRACEPLOTS



SUMMARY INFO

```
> print(titanic.stan)
Inference for Stan model: titanic.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

         mean se_mean   sd    2.5%     25%     50%     75%   97.5% n_eff Rhat
alpha   -0.99    0.01 0.28   -1.52   -1.18   -0.98   -0.79   -0.45  1012    1
beta1    1.29    0.01 0.42    0.47    1.01    1.29    1.57    2.11  1072    1
beta2   -0.03    0.00 0.01   -0.04   -0.03   -0.03   -0.02   -0.01  1019    1
beta3    0.01    0.00 0.00    0.01    0.01    0.01    0.01    0.02  3667    1
beta4    0.04    0.00 0.01    0.01    0.03    0.04    0.05    0.07  1129    1
lp__  -356.66    0.04 1.58 -360.59 -357.47 -356.31 -355.48 -354.59  1398    1
```

WRAP-UP

- Bayesian statistics can be used to perform the same analysis as you can do as a frequentist
- With vague priors, you will expect to see similar results from Bayes to frequentist
- Advantages of Bayesian
  - Easier to compute probability intervals
  - Easier to find quantities such as probabilities or transformations, such as CV
  - Easier to handle complex models (need make sure everything is specified correctly and ensure convergence of the MCMC…so samples can be used)