"Try to be a rainbow in someone's cloud."

— **Maya Angelou**

Bayesian Statistics

CLASS 3

Goals for today

Understand how to do general linear regressions in a Bayesian framework

General understanding of a more complex Bayesian analysis

Regression modeling

We will be using the rstanarm package

- Can do many different types of regression including:
  - Multiple linear regression (stan_glm)
  - Logistic regression (stan_glm)
  - Poisson regression (stan_glm)
  - Negative binomial regression (stan_glm.nb)
  - Other generalized linear models (stan_glmer, stan_nler, stan_gamm4…..)
- Easy to program (syntax similar to the frequentist version in R) and puts noninformative priors on parameters (you can specify your own priors if you don’t like theirs)
- Also provides good visualizations to check your models
- You can still control number of iterations, the burn-in size, the number of chains, etc…

Ames Housing data

```
train_reg <- train %>%
dplyr::select(Sale_Price,  Lot_Area, Age,  Total_Bsmt_SF,  Garage_Area,
Gr_Liv_Area, Central_Air)
```

Stan Code

```
library(rstanarm)
library(bayesplot)
model1<-stan_glm(Sale_Price~.+I(Age^2),data=train_reg,
        seed=1097,refresh=0)
```

```
Model Info:
 function:     stan_glm
 family:       gaussian [identity]
 formula:      Sale_Price ~ . + I(Age^2)
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 2051
 predictors:   8
```

Comparison to frequentist

summary(model1)

Estimates:

	mean	sd
(Intercept)	40324.5	5684.8
Lot_Area	0.5	0.1
Age	-1672.8	98.8
Total_Bsmt_SF	38.6	2.6
Garage_Area	57.9	5.7
Gr_Liv_Area	63.2	2.2
Central_AirY	10851.0	3980.7
I(Age^2)	9.9	1.0
sigma	41449.3	647.2

model2<-lm(Sale_Price~.,data=train_reg)summary
summary(model2)

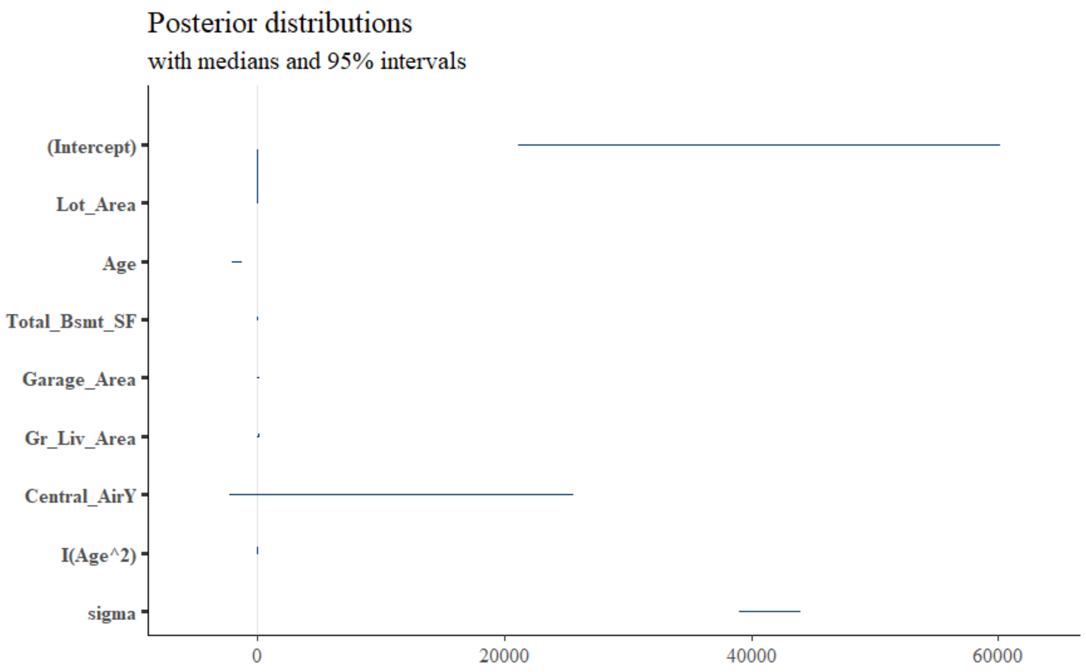
Coefficients:

	Estimate	Std. Error
(Intercept)	40396.2879	5649.3208
Lot_Area	0.5450	0.1142
Age	-1671.8214	100.1042
Total_Bsmt_SF	38.5593	2.5508
Garage_Area	58.0207	5.6101
Gr_Liv_Area	63.1880	2.2646
Central_AirY	10758.8634	3922.9604
I(Age^2)	9.8950	0.9961 ---

Residual standard error: 41420 on 2043 degrees of freedom

Visuals

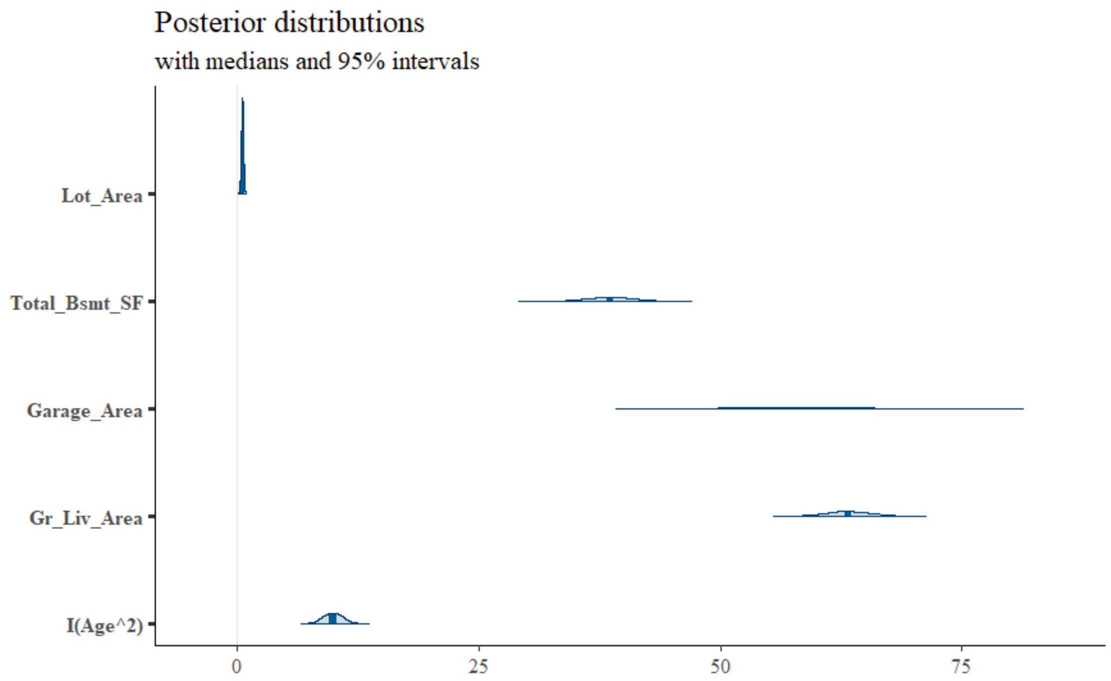
sims <- as.array(model1) ### array with dimensions 1000 x 4 x 9
plot_title <- ggtitle("Posterior distributions", "with medians and 95% intervals")mcmc_areas(sims, prob = 0.95) + plot_title



Only selected variables

sims2<-sims[,,-c(1,3,7,9)]

plot_title <- ggtitle("Posterior distributions", "with medians and 95% intervals") mcmc_areas(sims2, prob = 0.95) + plot_title



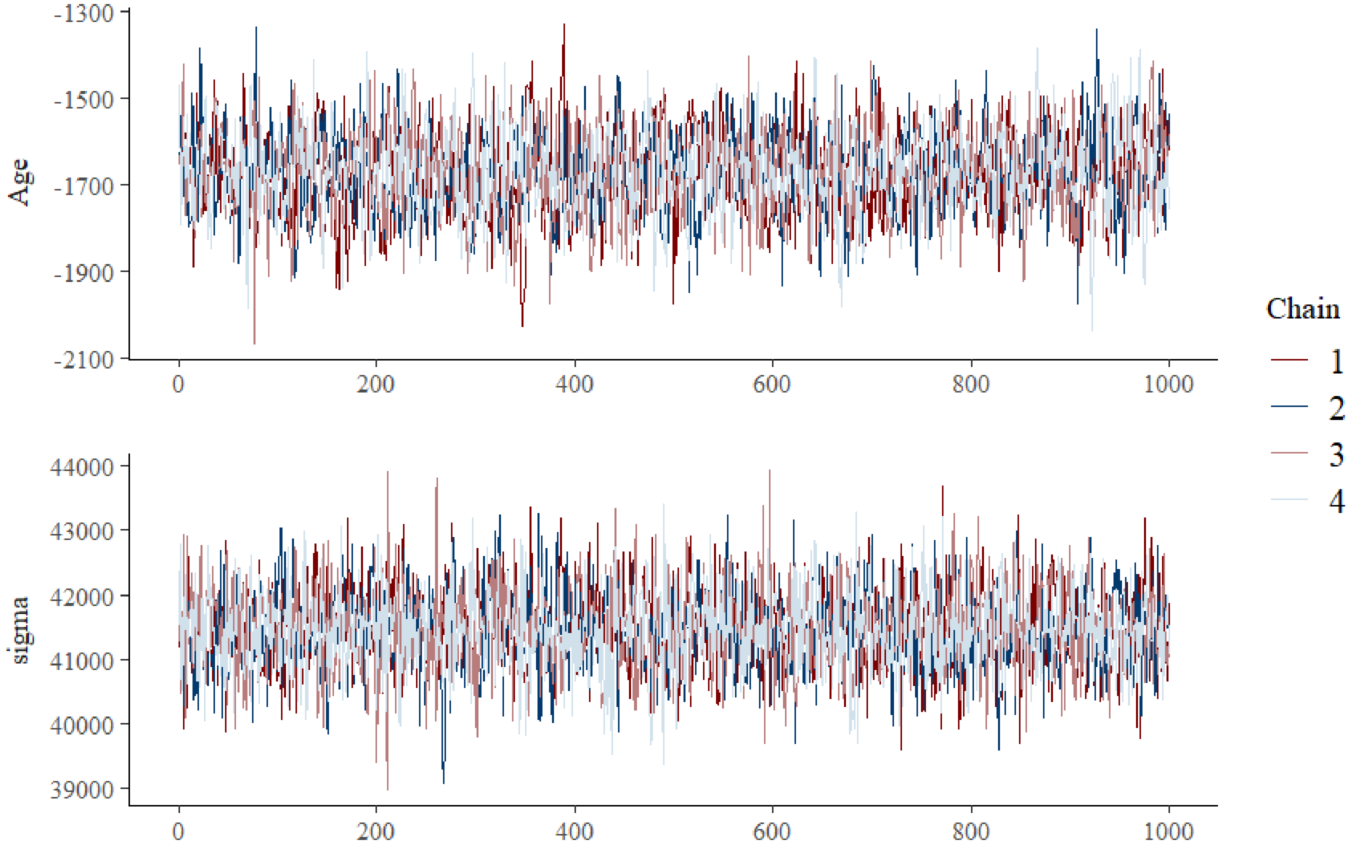
Posterior intervals (credible intervals)

```
quantile(sims[,,3], probs = c(.025,.975))  ###for Age

 2.5%          97.5%
-1861.409      --1481.822
```

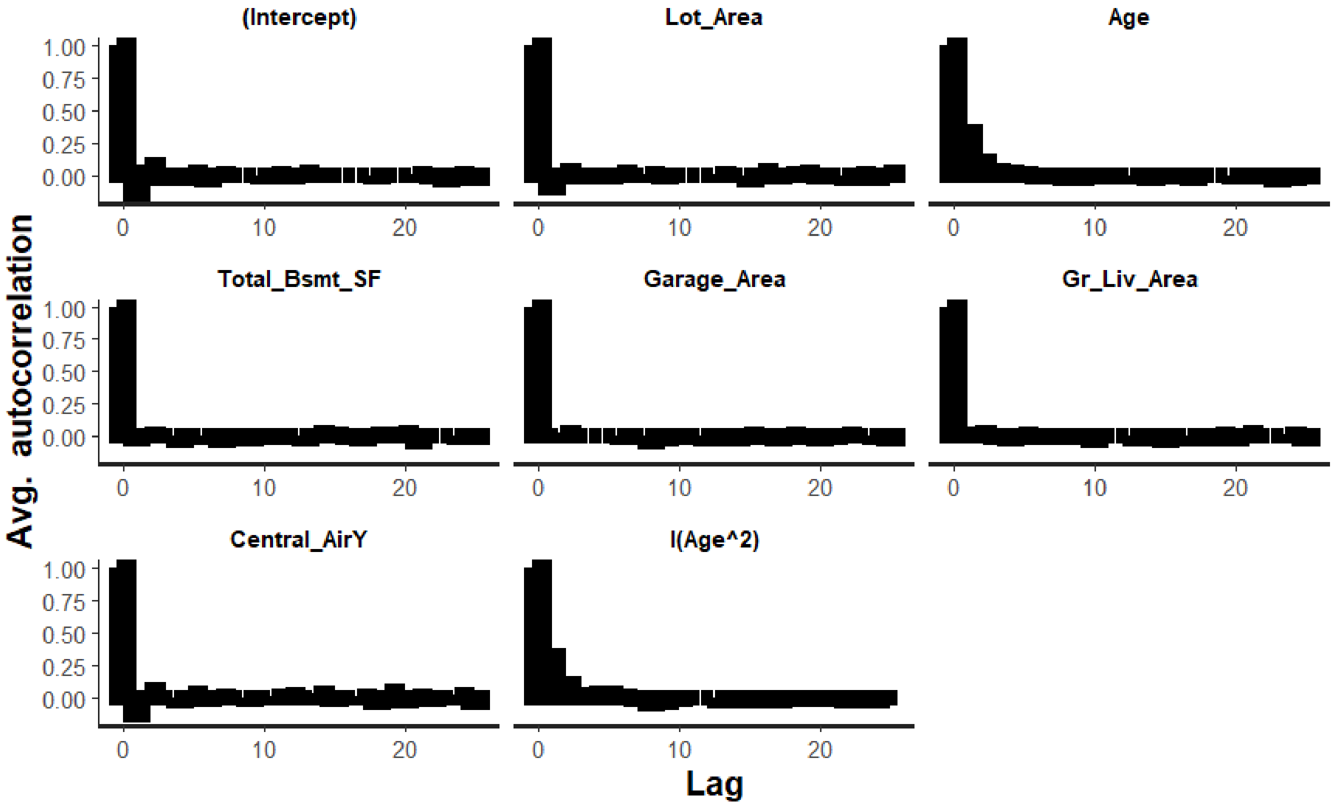
Traceplots

```
color_scheme_set("mix-blue-red")
mcmc_trace(sims, pars = c("Age", "sigma"),  facet_args = list(ncol = 1, strip.position = "left"))
```



Autocorrelation

stan_ac(model1) ### Age could be centered

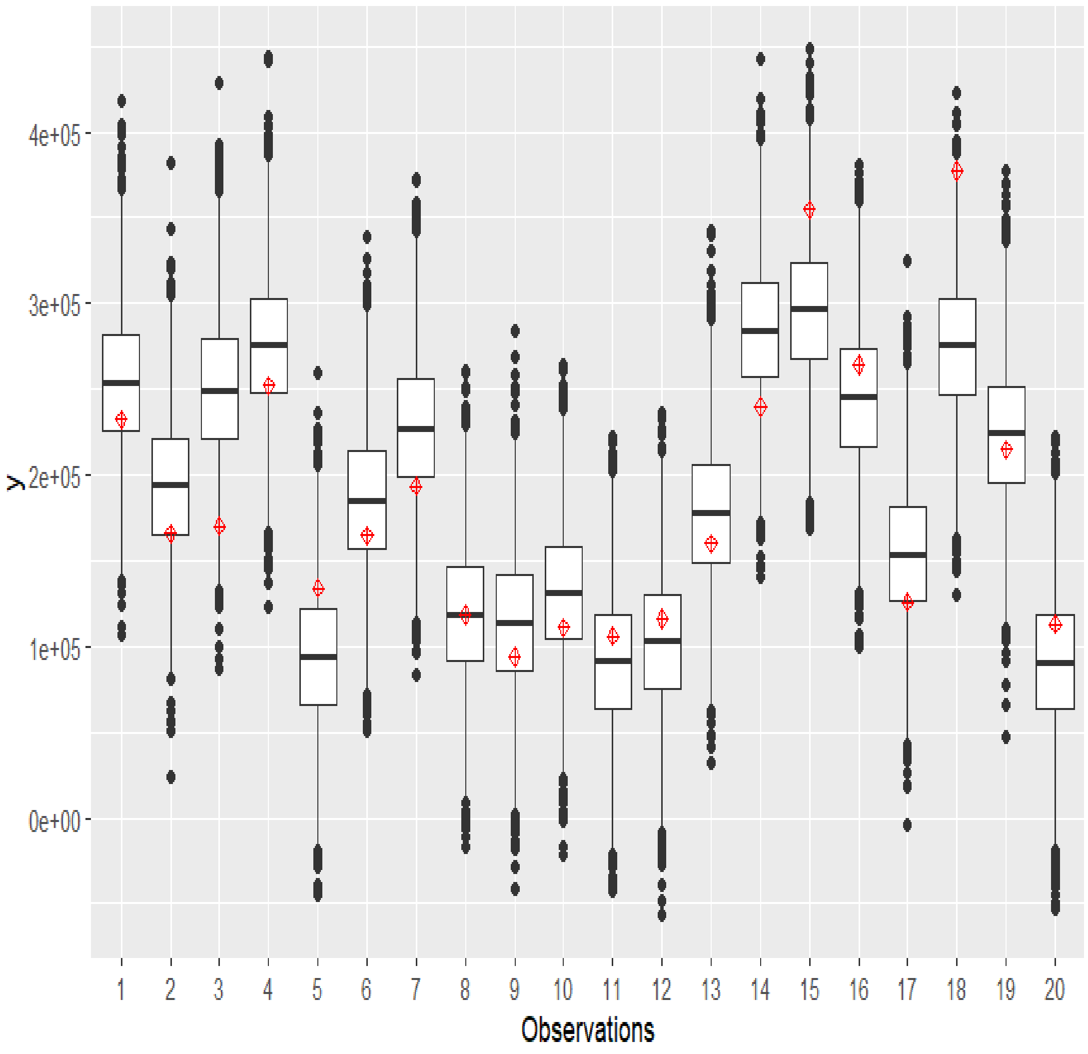


Fit of the model

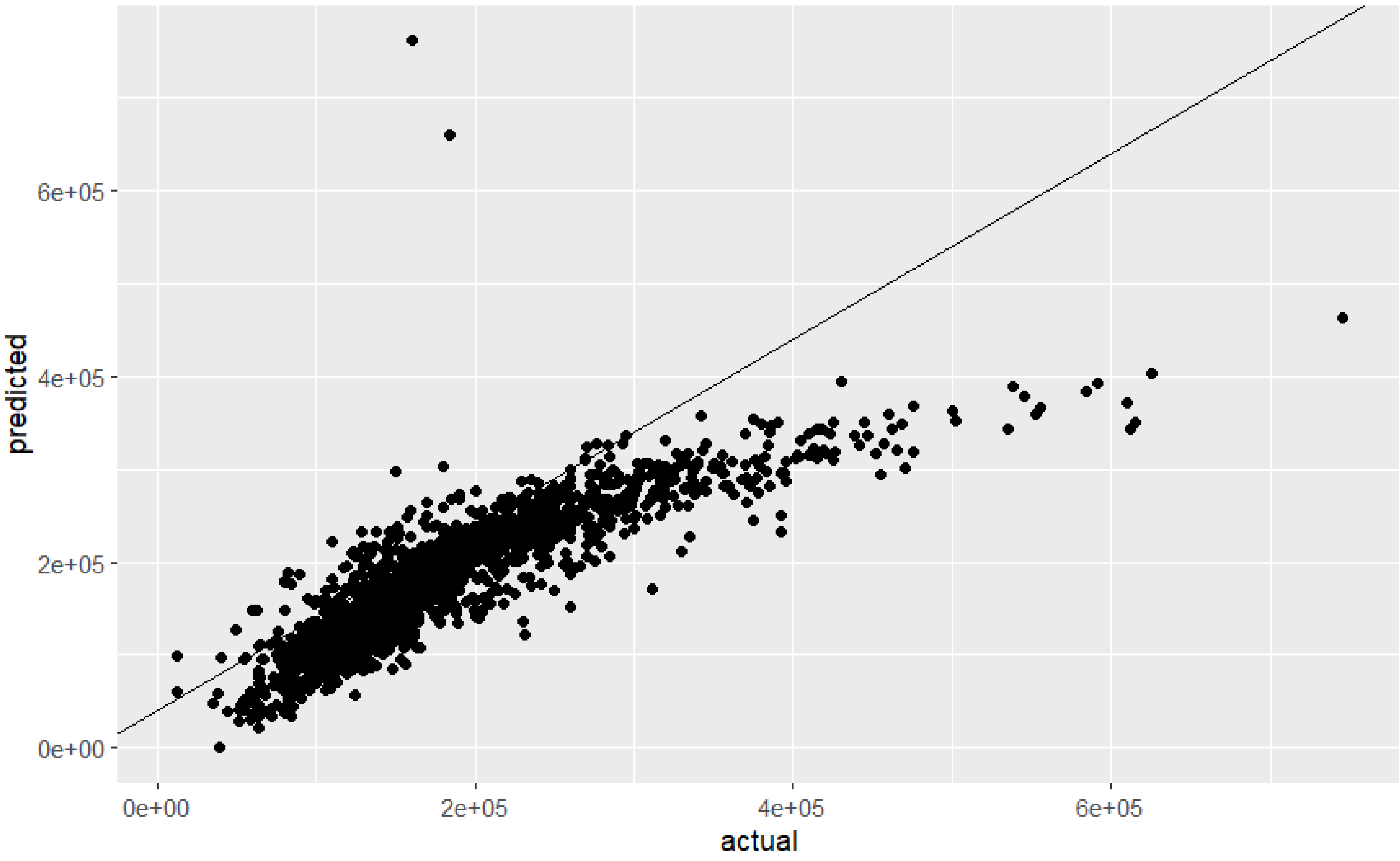
```
pred.y<- posterior_predict(model1)
dim(pred.y)
### For first 20 observations in training data
pred.y2<-
data.frame(x=c(rep(1:20,each=nrow(pred.y))),y=as.numeric(
pred.y[,1:20]))

actual.y=data.frame(x2=1:20,y2=train_reg$Sale_Price[1:20])

ggplot(pred.y2,aes(x=as.factor(x),y=y))+geom_boxplot()+ge
om_point(data=actual.y,aes(x=as.factor(x2),y=y2),color="re
d",shape=9)+  labs(x="Observations")
```



post.mean=apply(pred.y,2,mean)plot.dat=data.frame(actual=train_reg$Sale_Price,predicted=post.mean)
ggplot(plot.dat,aes(x=actual,y=predicted))+ geom_point()+geom_abline(slope=1,intercept = coef(model1)[1])
###Underpredicts the larger house values



Logistic regression

```
new.dat=titanic_train[complete.cases(titanic_train),]

titanic.model<-stan_glm(Survived~ Sex+Age + Fare+ Sex:Fare,data=new.dat,family =
binomial(link="logit"),prior = normal(0,100), prior_intercept = normal(0,100),
seed=03786,refresh=0,QR=T)

summary(titanic.model)
```

Model Info:
 function: stan_glm
 family: binomial [logit]
 formula: Survived ~ Sex + Age + Fare + Sex:Fare
 algorithm: sampling
 sample: 4000 (posterior sample size)
 priors: see help('prior_summary')
 observations: 714
 predictors: 5

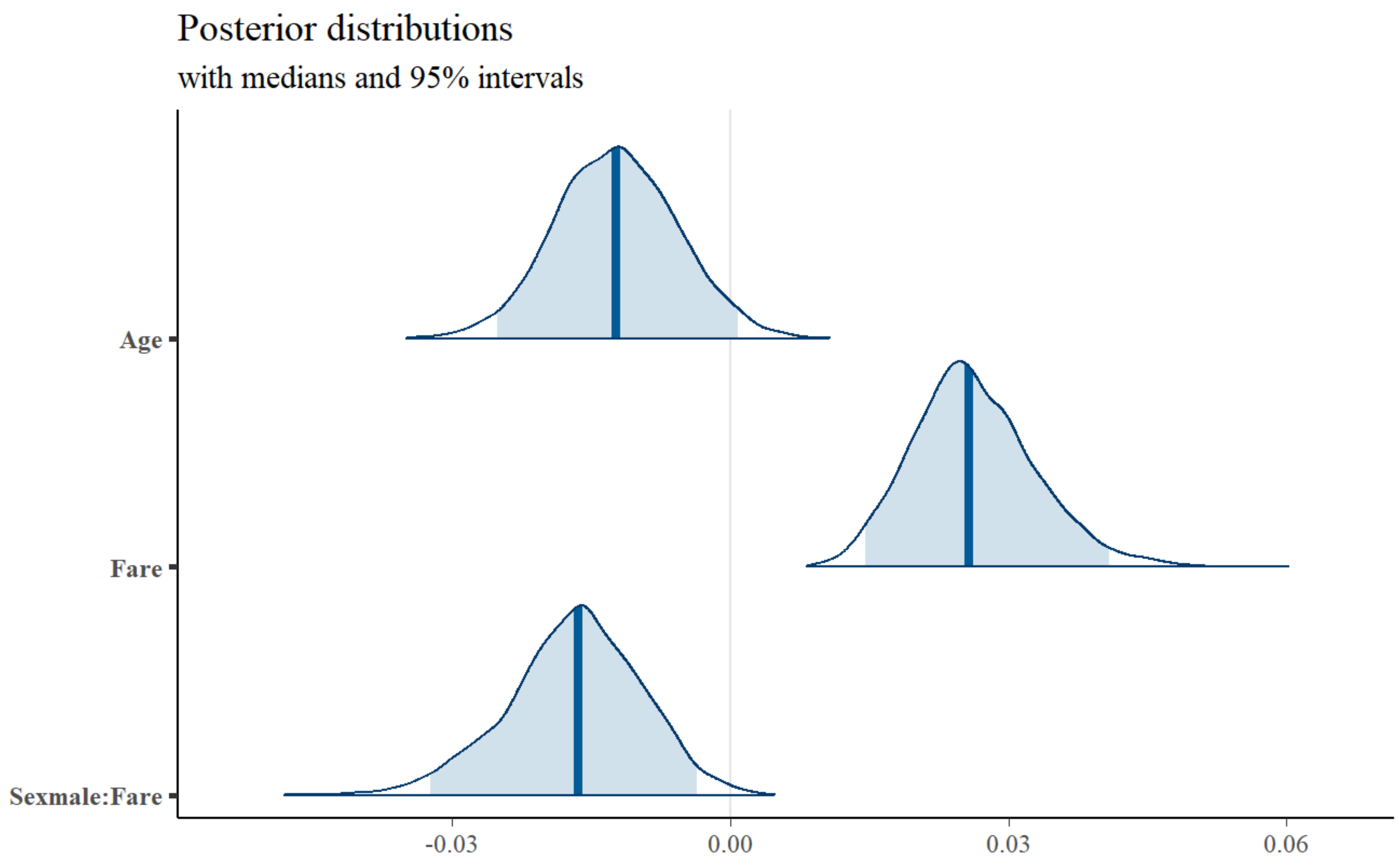
Estimates:

	mean	sd
(Intercept)	0.6	0.3
Sexmale	-1.9	0.3
Age	0.0	0.0
Fare	0.0	0.0
Sexmale:Fare	0.0	0.0

** very small values (need to pull of individually if want to see values

Posterior distributions

```
sims <- as.array(titanic.model)
sims2<-sims[,,3:5]
plot_title <- ggtitle("Posterior distributions“,
 "with medians and 95% intervals")
mcmc_areas(sims2, prob = 0.95) + plot_title
```



Better summary information

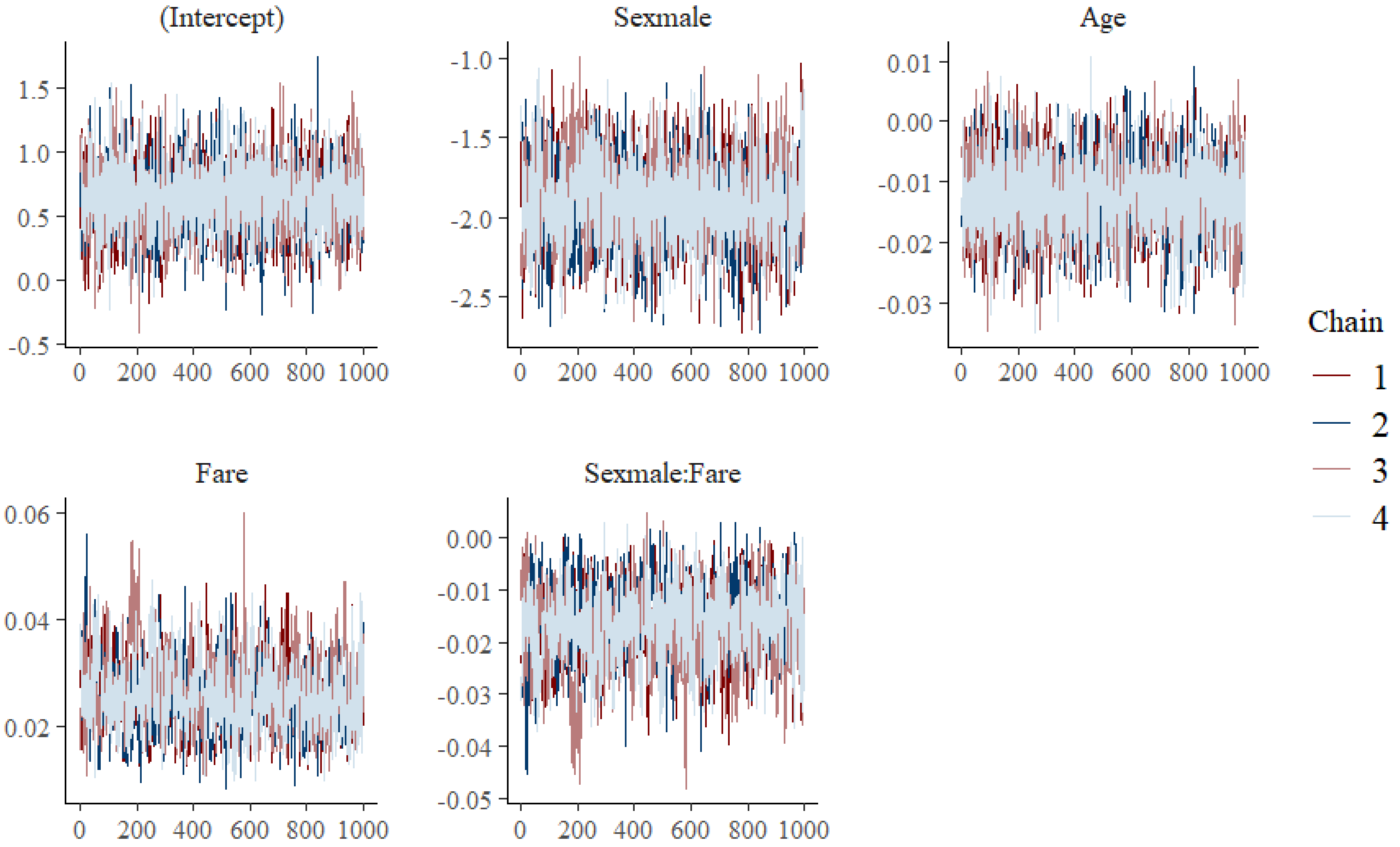
```
quant.fun1 <- function(x){
   temp=quantile(x,probs = 0.025)
   return(temp)}

quant.fun2 <- function(x){
   temp=quantile(x,probs = 0.975)
   return(temp)}

correct.output<-
matrix(c(apply(sims,3,mean),apply(sims,3,median),apply(sims,
3,sd),apply(sims,3,quant.fun1),apply(sims,3,quant.fun2)),ncol=
5)
colnames(correct.output)<-c("mean","median","standard
error","2.5%","97.5%")
rownames(correct.output)<-
c("Intercept","Male","Age","Fare","Sex:Fare")
correct.output
```

	mean	median	standard error	2.5%	97.5%
Intercept	0.62760901	0.62569814	0.274586657	0.08038008	1.1531670579
Male	-1.88719348	-1.88241875	0.270073216	-2.43506413	-1.3655037403
Age	-0.01223392	-0.01227592	0.006584693	-0.02496053	0.0006976183
Fare	0.02667329	0.02626842	0.006970298	0.01425691	0.0418701352
Sex:Fare	-0.01719808	-0.01704796	0.007482674	-0.03303847	-0.0035268331

Traceplots



Example from online

Hierarchical Model

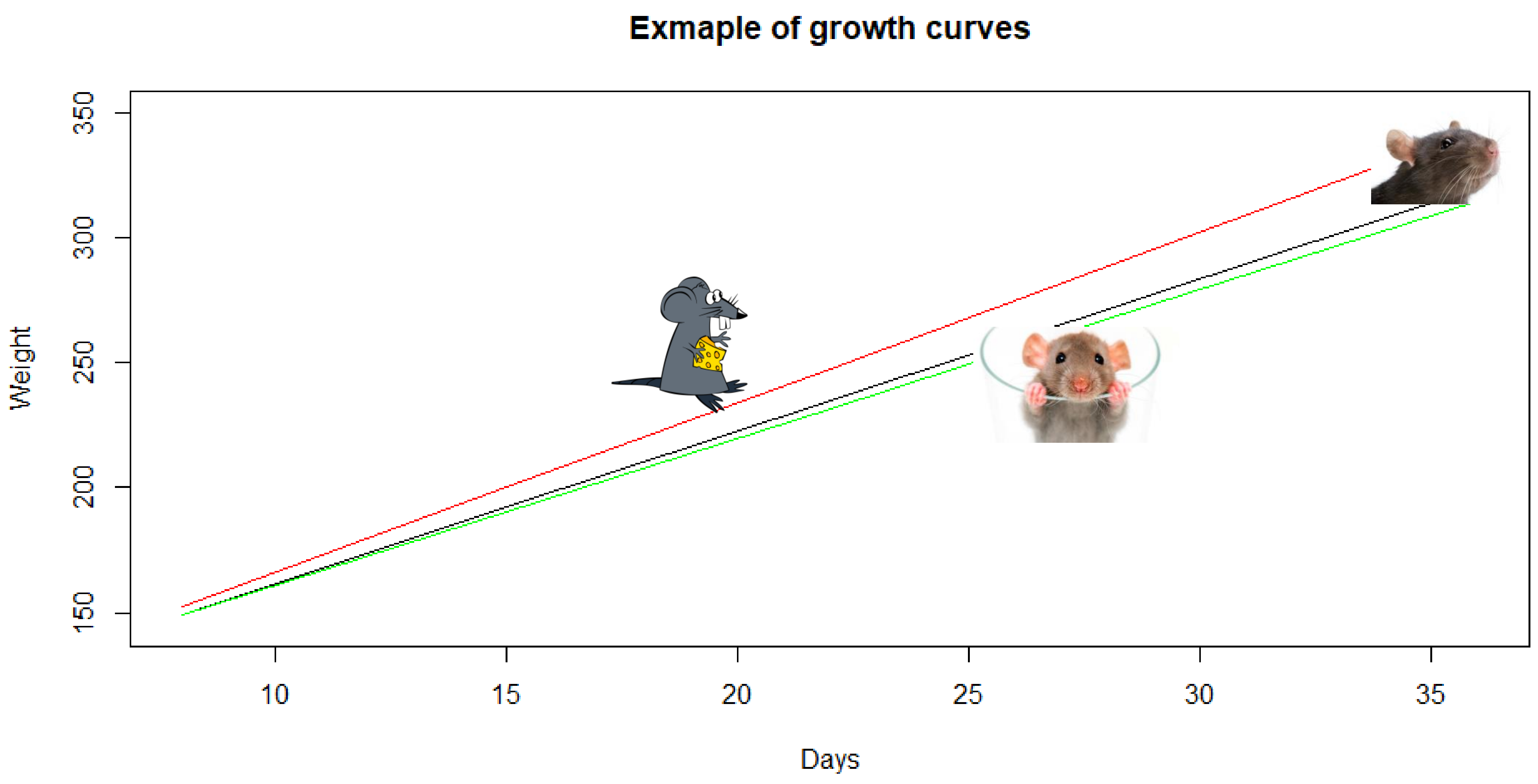
A total of 30 rats were followed across time

The response variable being measured was their weight

Weight was recorded on day 8, 15, 22, 29 and 36 (i.e. 5 measurements taken on each rat)

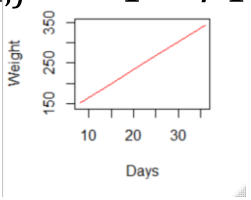
This can be viewed as ‘panel’ data or ‘cluster’ data

We will create a growth curve for each rat (assume a linear growth curve)

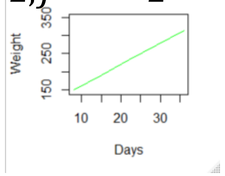


Exmaple of growth curves
Weight
350
300
250
200
150
10
15
20
25
30
35
Days

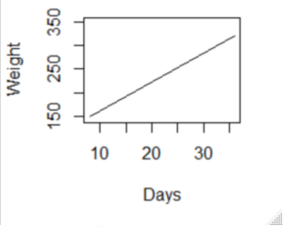
$y_{1,j} = \alpha_1 + \beta_1 x_j$



$y_{2,j} = \alpha_2 + \beta_2 x_j$

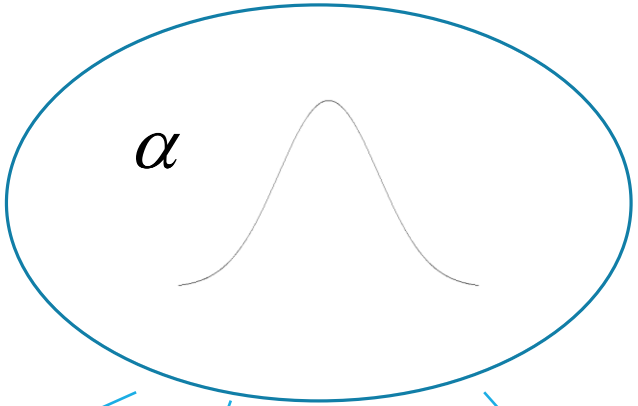


$y_{3,j} = \alpha_3 + \beta_3 x_j$

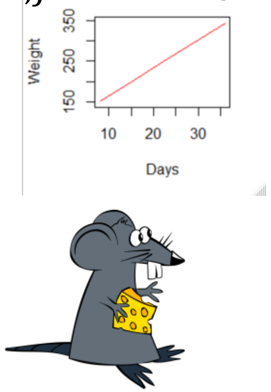


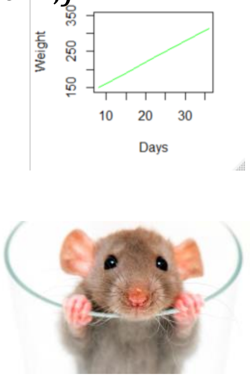
• • •

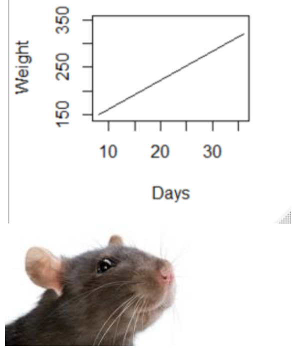
N=30 rats



$y_{1,j} = \alpha_1 + \beta_1 x_j$ $\qquad$ $y_{2,j} = \alpha_2 + \beta_2 x_j$ $\qquad$ $y_{3,j} = \alpha_3 + \beta_3 x_j$

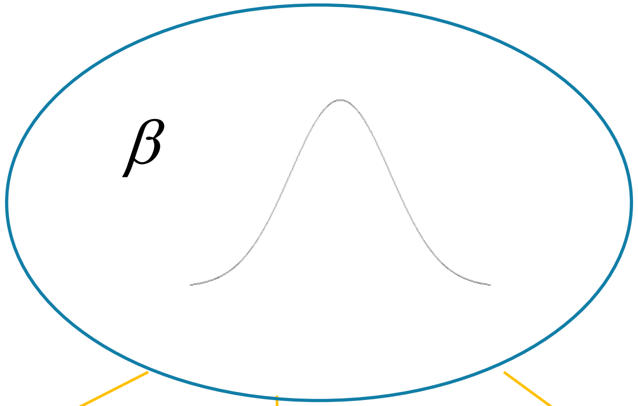




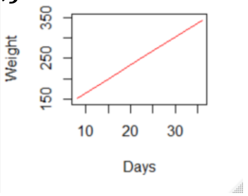


• • •

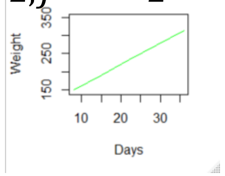
N=30 rats



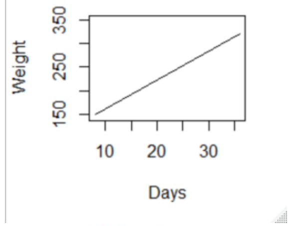
$y_{1,j} = \alpha_1 + \beta_1 x_j$



$y_{2,j} = \alpha_2 + \beta_2 x_j$



$y_{3,j} = \alpha_3 + \beta_3 x_j$



• • •

N=30 rats

The model

$$Y_{i,j} \sim Normal(\alpha_i + \beta_i\left(x_j - \bar{x}\right), \sigma_Y)$$

$$\alpha_i \sim Normal(\mu_\alpha, \sigma_\alpha)$$

$$\beta_i \sim Normal(\mu_\beta, \sigma_\beta)$$

The model

$$Y_{i,j} \sim Normal(\alpha_i + \beta_i(x_j - \bar{x}), \sigma_Y)$$

$$\alpha_i \sim Normal(\mu_\alpha, \sigma_\alpha)$$

$$\beta_i \sim Normal(\mu_\beta, \sigma_\beta)$$

The model

$$Y_{i,j} \sim Normal(\alpha_i + \beta_i(x_j - \bar{x}), \sigma_Y)$$

$$\alpha_i \sim Normal(\mu_\alpha, \sigma_\alpha)$$

$$\beta_i \sim Normal(\mu_\beta, \sigma_\beta)$$

The model

$$Y_{i,j} \sim Normal(\alpha_i + \beta_i(x_j - \bar{x}), \sigma_Y)$$

$$\alpha_i \sim Normal(\mu_\alpha, \sigma_\alpha)$$

$$\beta_i \sim Normal(\mu_\beta, \sigma_\beta)$$

Need prior distributions

Priors

$$\mu_\alpha \sim Normal(0,100)$$

$$\mu_\beta \sim Normal(0,100)$$

$$\sigma_Y^2 \sim Inv - Gamma(0.001,0.001)$$

$$\sigma_\alpha^2 \sim Inv - Gamma(0.001,0.001)$$

$$\sigma_\beta^2 \sim Inv - Gamma(0.001,0.001)$$

```
data {
  int<lower=0> N; // Number of rats
  int<lower=0> Npts; // Number of data points
  int<lower=0> rat[Npts]; // Lookup index for rat
  real x[Npts];
  real y[Npts];
  real xbar;
}
parameters {
  real alpha[N];
  real beta[N];
  real mu_alpha;
  real mu_beta;
  real <lower=0> sigmasq_y;
  real <lower=0> sigmasq_alpha;
  real <lower=0> sigmasq_beta;
}
transformed parameters {
  real<lower=0> sigma_y;
  real<lower=0> sigma_alpha;
  real<lower=0> sigma_beta;
  sigma_y = sqrt(sigmasq_y);
  sigma_alpha = sqrt(sigmasq_alpha);
  sigma_beta = sqrt(sigmasq_beta);
}
model {
  mu_alpha ~ normal(0, 100);
  mu_beta ~ normal(0, 100);
  sigmasq_y ~ inv_gamma(0.001, 0.001);
  sigmasq_alpha ~ inv_gamma(0.001, 0.001);
  sigmasq_beta ~ inv_gamma(0.001, 0.001);
  alpha ~ normal(mu_alpha, sigma_alpha);
  beta ~ normal(mu_beta, sigma_beta);
  for (n in 1:Npts){
    int irat;
    irat = rat[n];
    y[n] ~ normal(alpha[irat] + beta[irat] * (x[n] - xbar), sigma_y);
  }
}
generated quantities {
  real alpha0;
  alpha0 = mu_alpha - xbar * mu_beta;
}
```

```
print(rats.stan,pars = c("mu_alpha","mu_beta","sigma_y","sigma_alpha","sigma_beta","alpha0"))
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff
mu_alpha	242.48	0.04	2.88	236.78	240.62	242.47	244.35	248.18	4910
mu_beta	6.18	0.00	0.11	5.97	6.11	6.18	6.26	6.40	4394
sigma_y	6.11	0.01	0.47	5.28	5.80	6.09	6.41	7.10	2190
sigma_alpha	14.96	0.03	2.24	11.38	13.38	14.70	16.25	20.12	4194
sigma_beta	0.53	0.00	0.10	0.37	0.47	0.52	0.59	0.75	2670
alpha0	106.44	0.06	3.74	98.99	103.97	106.50	108.99	113.67	4461

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

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)

Thank you!