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

— [**Maya Angelou**]

Bayesian Statistics

CLASS 3

Goals for today

Understand how to do general linear regression and Logistic regression in a Bayesian framework

Ames Housing data

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

You can create this data and export (use seed from summer).

Data into format

```
y_train = train_reg['Sale_Price']
x6_train = pd.get_dummies(x_train['Central_Air'],
drop_first=True ,dtype=int)
x6_train = pd.to_numeric(x6_train['Y'], errors='coerce)
x1_train=x_train['Lot_Area']
x2_train=x_train['Age']
x3_train=x_train['Total_Bsmt_SF']
x4_train=x_train['Garage_Area']
x5_train=x_train['Gr_Liv_Area
x7_train=x_train['Age']**2
```

Create model

```
with pm.Model() as model1:
 # Priors for coefficients
	beta1 = pm.Normal("beta1", mu=0, sigma=10000)
	beta2 = pm.Normal("beta2", mu=0, sigma=10000
	beta3 = pm.Normal("beta3", mu=0, sigma=10000)
	beta4 = pm.Normal("beta4", mu=0, sigma=10000)
	beta5 = pm.Normal("beta5", mu=0, sigma=10000
	beta6 = pm.Normal("beta6", mu=0, sigma=10000)
	beta7 = pm.Normal("beta7", mu=0, sigma=10000)
	intercept = pm.Normal("intercept", mu=0, sigma=100000)
# Standard deviation of residuals
	sigma = pm.HalfNormal("sigma", sigma=10000)
	mu = intercept + beta1*x1_train + beta2*x2_train +
beta3*x3_train + beta4*x4_train + beta5*x5_train +
beta6*x6_train+beta7*x7_train
```

```
	y_obs = pm.Normal("y_obs", mu=mu, sigma=sigma,
observed=y_train
	trace = pm.sample(2000, tune=1000,
random_seed=10976, return_inferencedata=True)
	trace.extend(pm.sample_posterior_predictive(trace))

ppc=trace.posterior_predictive['y_obs']
az.summary(trace)
```

Comparison to frequentist

	mean	sd	hdi_3%	hdi_97%	m
beta1	0.544	0.114	0.334	0.766	
beta2	-1669.620	98.790	-1857.909	-1489.848	
beta3	38.562	2.484	33.869	43.176	
beta4	58.352	5.667	47.523	68.782	
beta5	63.218	2.306	58.876	67.557	
beta6	9419.832	3778.773	2189.750	16352.377	
beta7	9.843	0.985	7.949	11.625	
intercept	41474.250	5551.936	30850.490	51764.329	
sigma	41273.826	634.582	40125.058	42497.164	

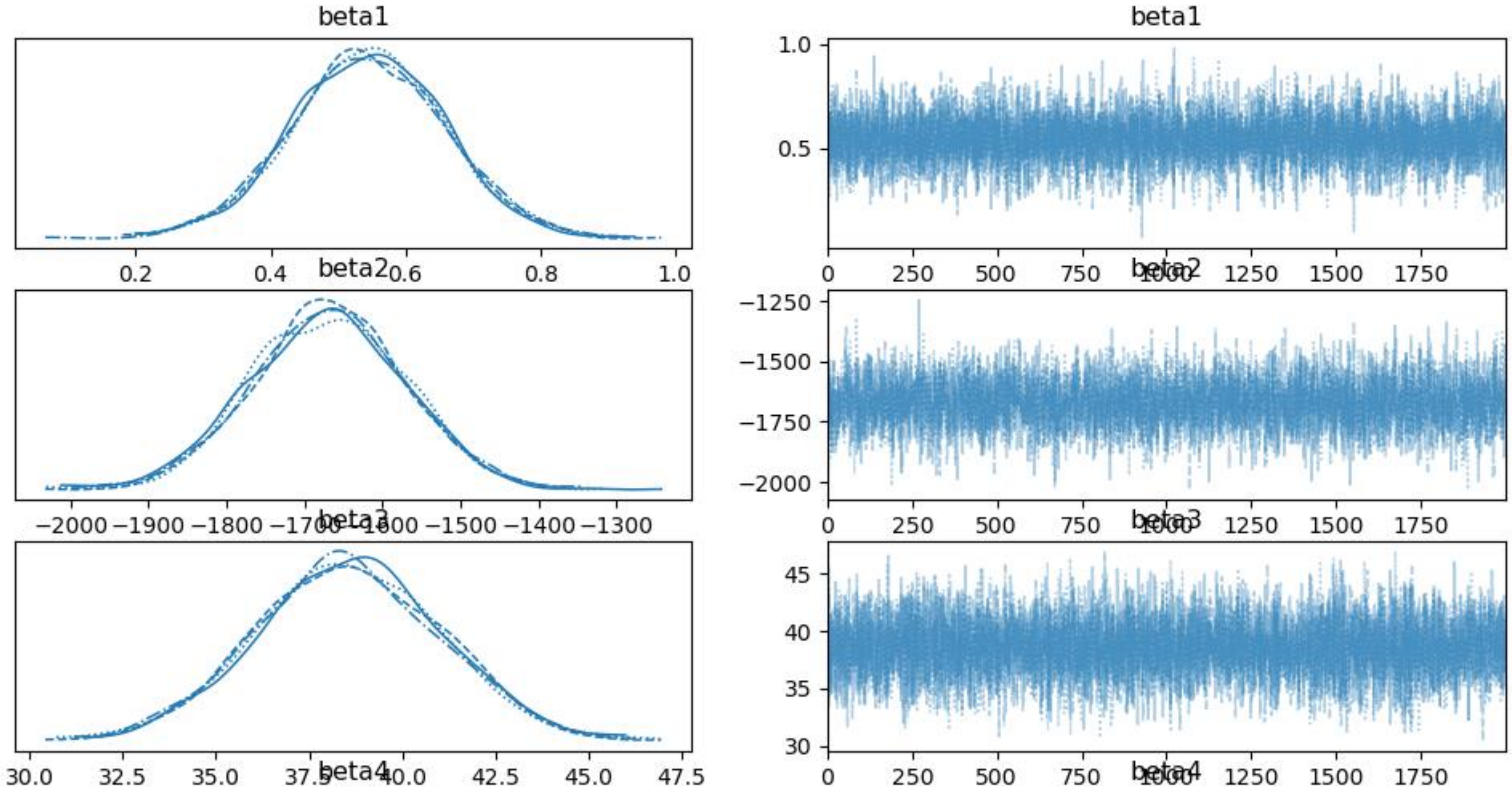
model2<-lm(Sale_Price~.,data=train_reg)summary
summary(model2)

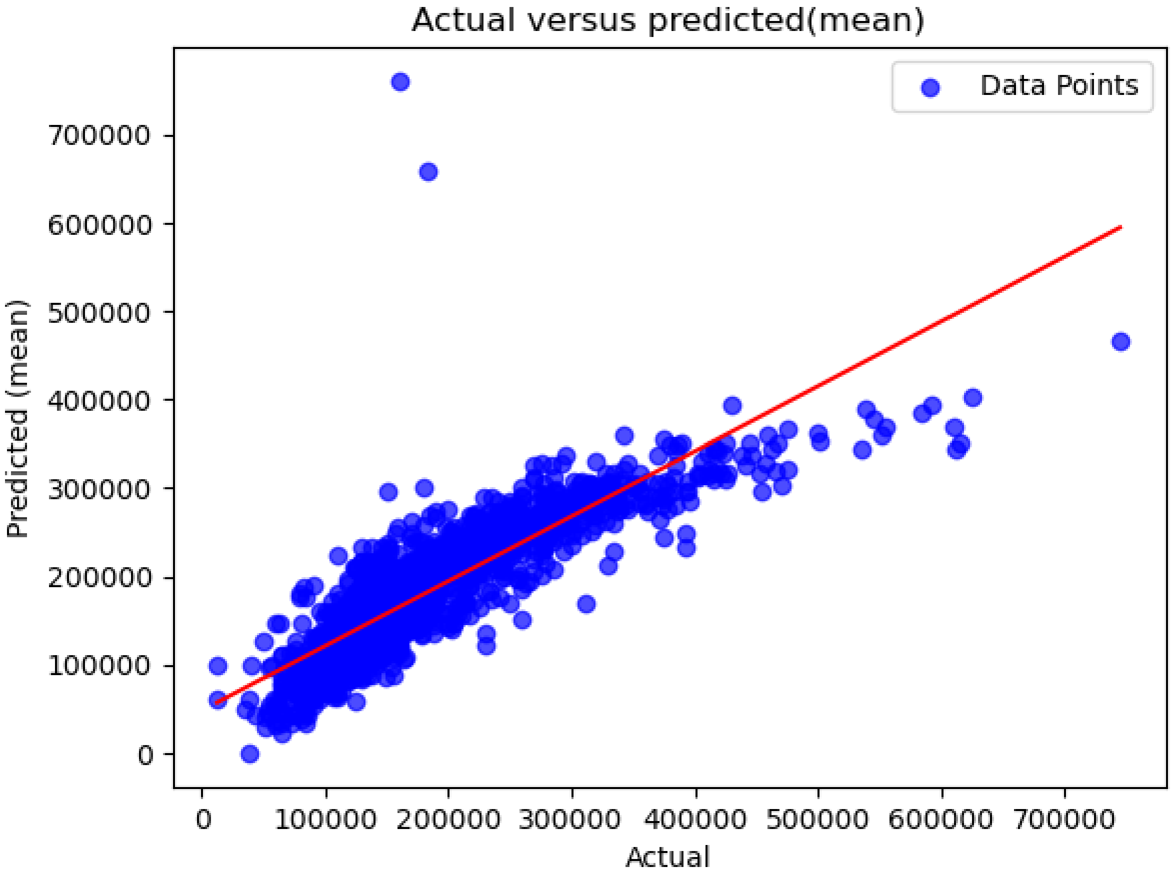
Coefficients:

	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 ---

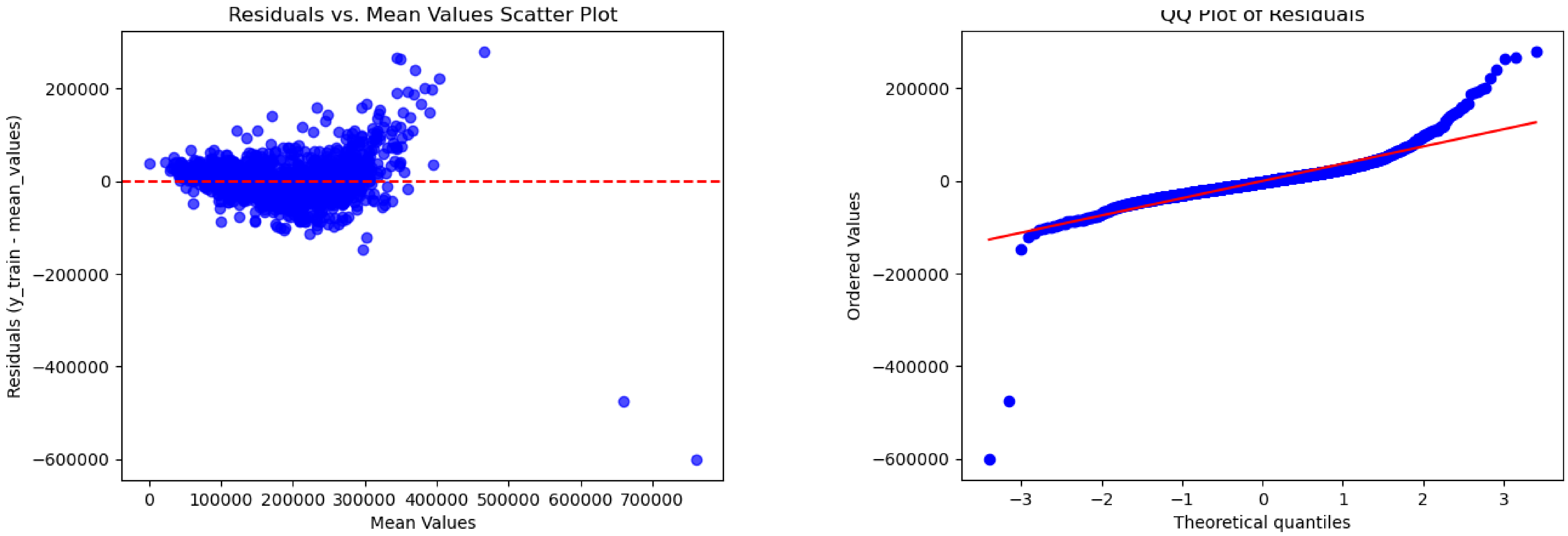
Residual standard error: 41420 on 2043 degrees of freedom

Traceplots


beta1
0.2
0.4
0.6
0.8
1.0
beta1
1.0
0.5
0
250
500
750
1000
1250
1500
1750
beta2
−2000
−1900
−1800
−1700
−1600
−1500
−1400
−1300
beta2
−1250
−1500
−1750
−2000
beta3
30.0
32.5
35.0
37.5
40.0
42.5
45.0
47.5
beta3
45
40
35
30
beta4
beta4


Actual versus predicted(mean)
Data Points
Predicted (mean)
0
100000
200000
300000
400000
500000
600000
700000
Actual

Checking assumptions (still have work to do)



Logistic regression

We will use the Bambi package to perform Logistic regression:

titanic_model = bmb.Model("Survived ~ sex + Age + Fare + sex_fare", data=titanic_train,family="bernoulli")

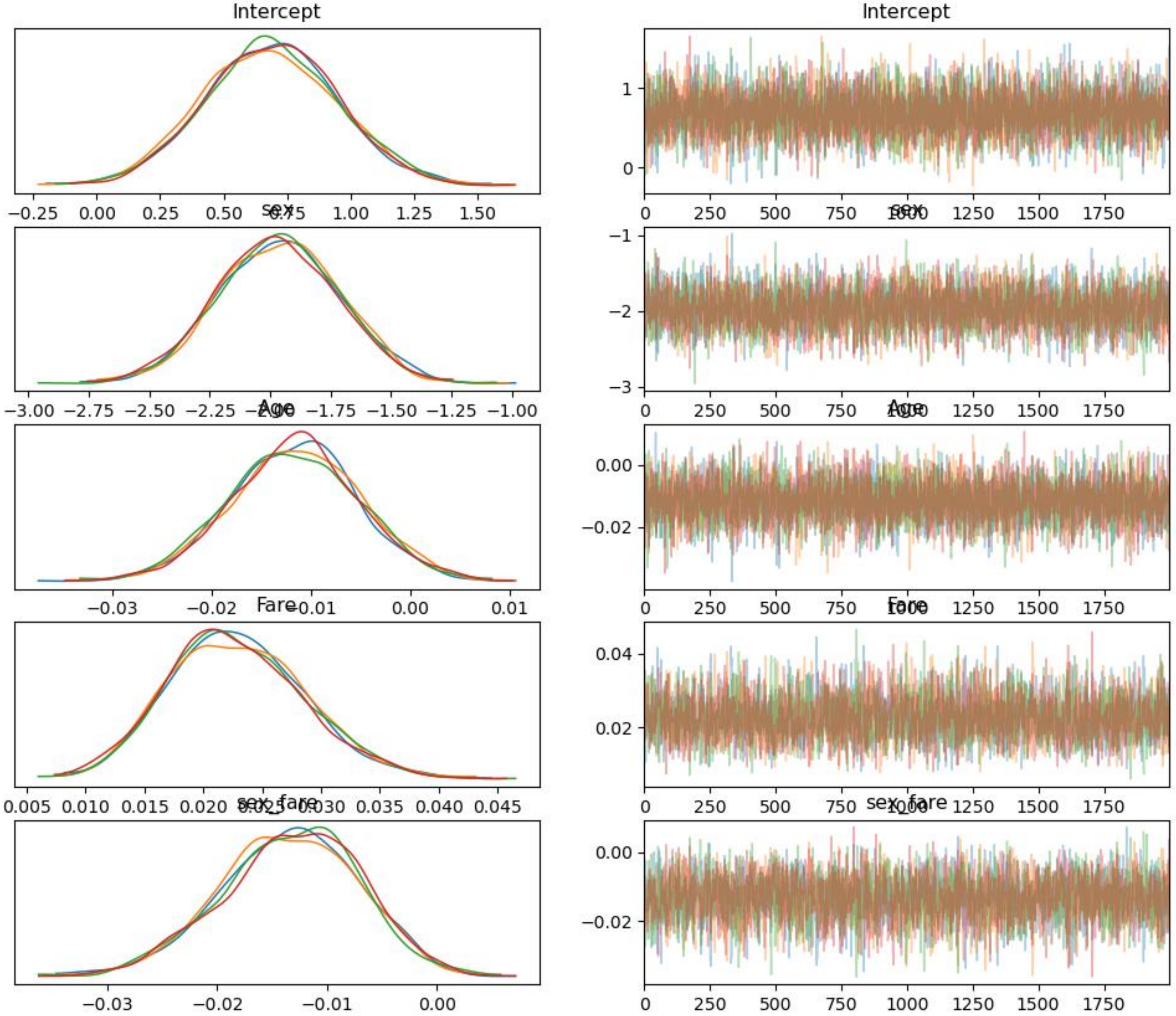
titanic_fitted = titanic_model.fit(draws=2000, target_accept=0.85, random_seed=56892, idata_kwargs={"log_likelihood": True})

Comparison: Bayesian to Frequentist

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
Intercept	0.686	0.263	0.205	1.189	0.004	0.003	5237.0	5871.0	1.0
sex	-1.966	0.249	-2.421	-1.495	0.004	0.003	3659.0	5122.0	1.0
Age	-0.012	0.006	-0.024	0.001	0.000	0.000	5189.0	4929.0	1.0
Fare	0.023	0.006	0.012	0.034	0.000	0.000	3010.0	4187.0	1.0
sex_fare	-0.013	0.006	-0.026	-0.002	0.000	0.000	2906.0	3963.0	1.0

```
             Estimate S
(Intercept)  0.642704
sex         -1.904051
Age         -0.011958
Fare         0.024752
sex_fare    -0.015736
```

MCMC plots



Intercept
Intercept
sex
sex
Age
Age
Fare
Fare
sex_fare
sex_fare

Self study:

Example from online

Hierarchical Model

A total of 30 rats were followed across time

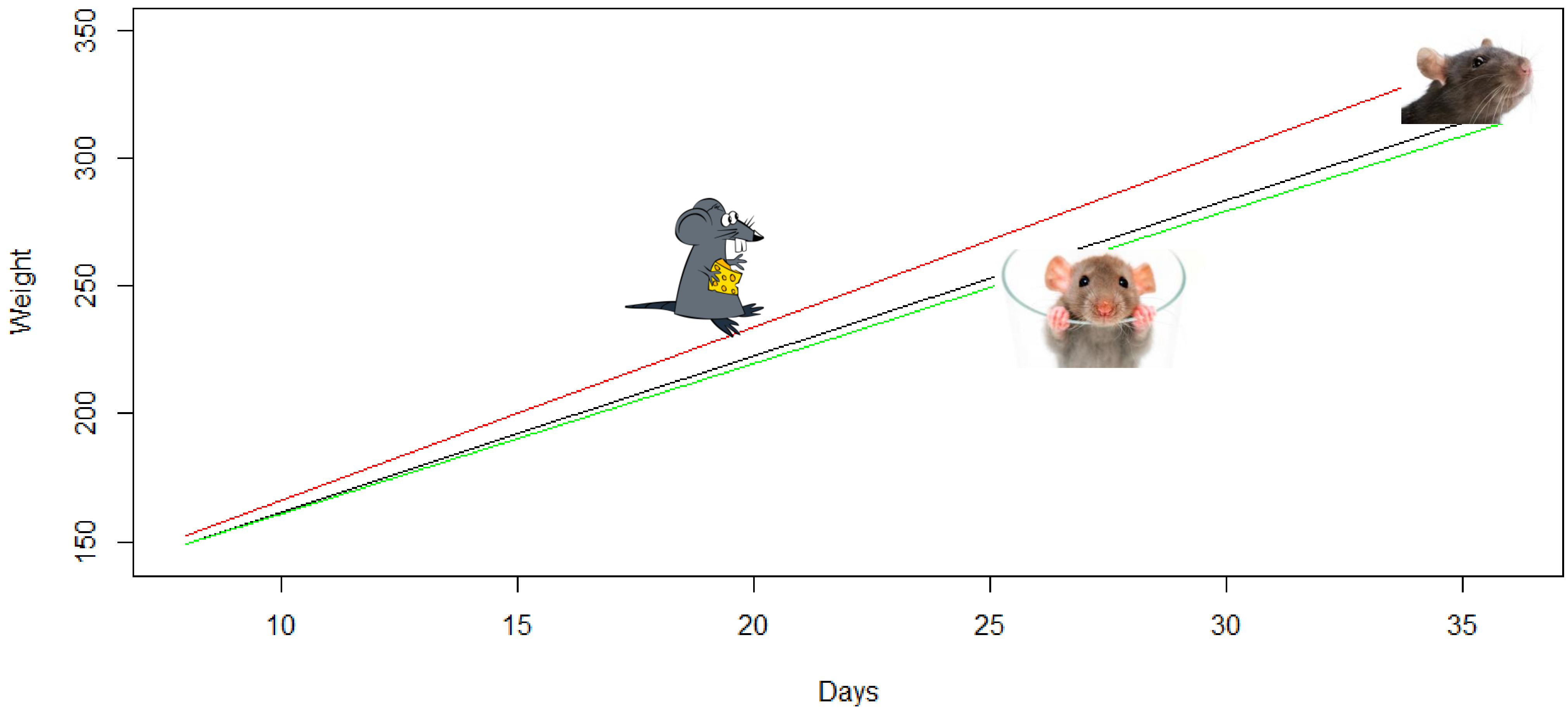
The response variable being measured was their weight

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

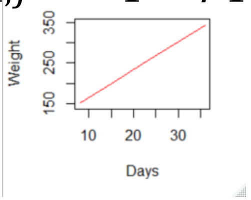
This can be viewed as ‘panel’ data or ‘cluster’ data

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

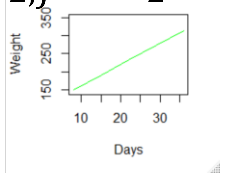
Exmaple of growth curves



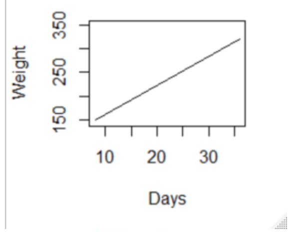
$y_{1,j} = \alpha_1 + \beta_1 x_j$



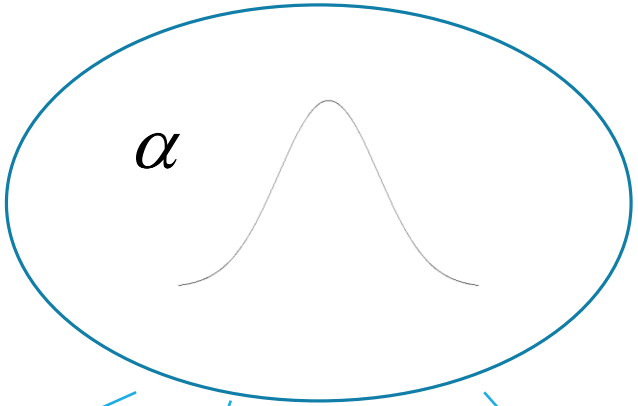
$y_{2,j} = \alpha_2 + \beta_2 x_j$



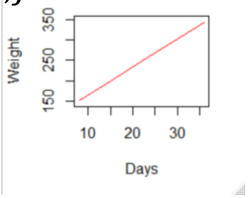
$y_{3,j} = \alpha_3 + \beta_3 x_j$



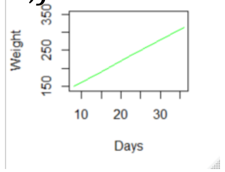
N=30 rats



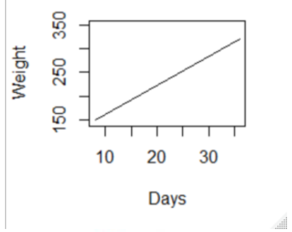
$y_{1,j} = \alpha_1 + \beta_1 x_j$



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

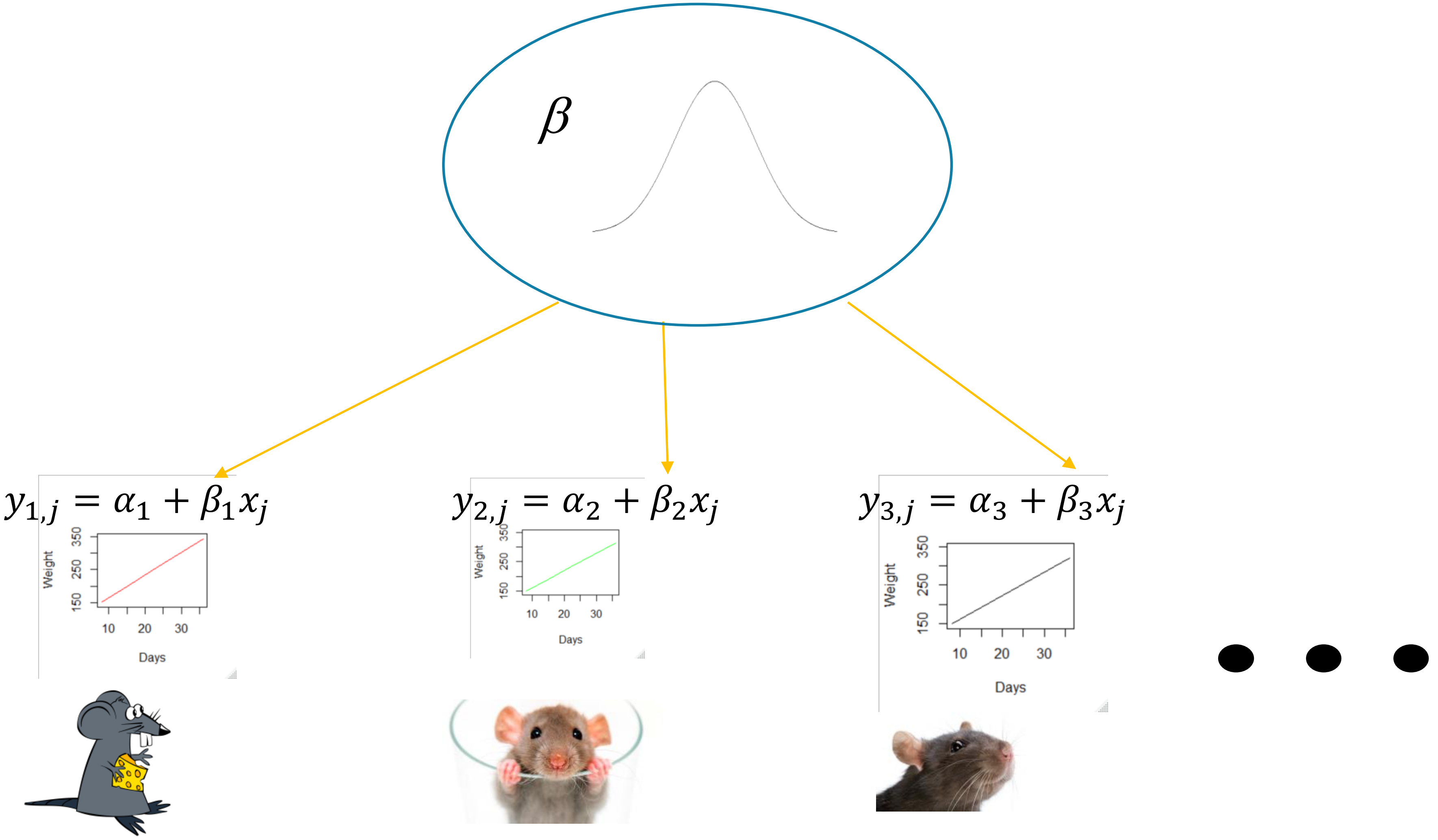


$y_{3,j} = \alpha_3 + \beta_3 x_j$



• • •

N=30 rats



β
$y_{1,j} = \alpha_1 + \beta_1 x_j$
$y_{2,j} = \alpha_2 + \beta_2 x_j$
$y_{3,j} = \alpha_3 + \beta_3 x_j$
Weight
150 250 350
10 20 30
Days

N=30 rats

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

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

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

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

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!