PREDICTION AND UNCERTAINTY

SCHOOL OF
CULTURE AND SOCIETY
AARHUS UNIVERSITY

14 APRIL 2022

SIGURD FYHN SØRENSEN
STUDENT TEACHER

SOLIDUM PETIT IN PROFUNDIS · UNIVERSITAS ARHUSIENSIS ·

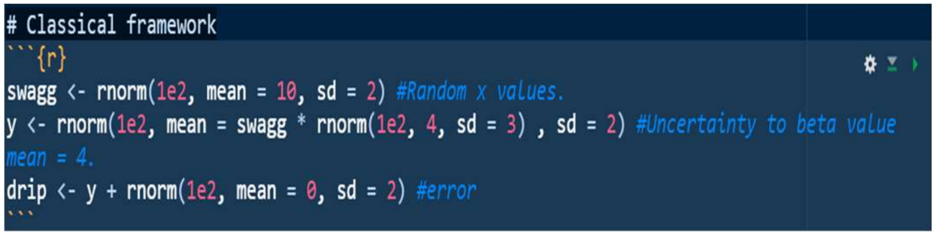
CLASSICAL

Classical (frequentist framework):

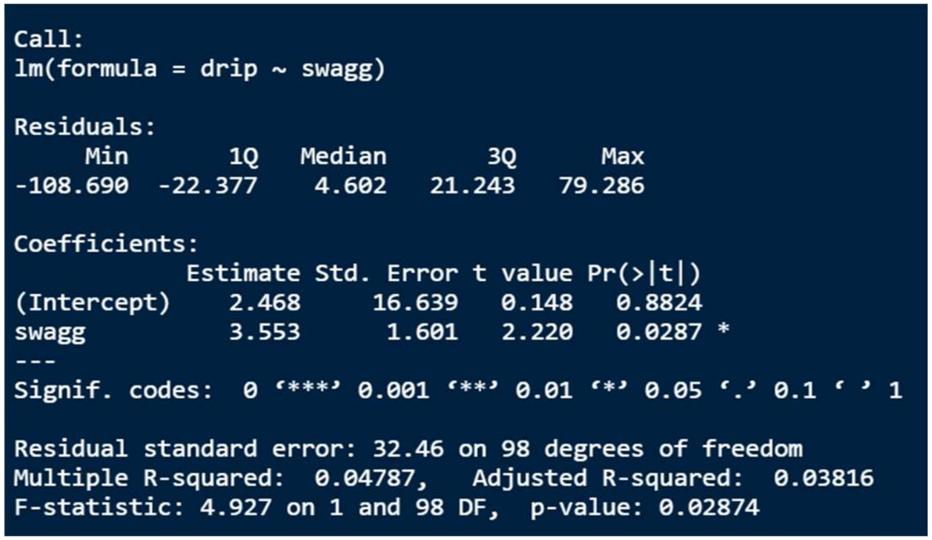
- Hypothesis testing & significance value.

- “If H_0 is true, then we would expect to get a result as extreme as the one obtained from our sample 2.9% of the time. Since that p-value is smaller than our alpha level of 5%, we reject the null hypothesis in favor of the alternate hypothesis.”

- <u>frequentist</u> interpretation that views probability as the <u>limit</u> of the relative frequency of an event after many trials.



~~~
# Classical framework
```{r}
swagg <- rnorm(1e2, mean = 10, sd = 2) #Random x values.
y <- rnorm(1e2, mean = swagg * rnorm(1e2, 4, sd = 3) , sd = 2) #Uncertainty to beta value
mean = 4.
drip <- y + rnorm(1e2, mean = 0, sd = 2) #error
```
~~~



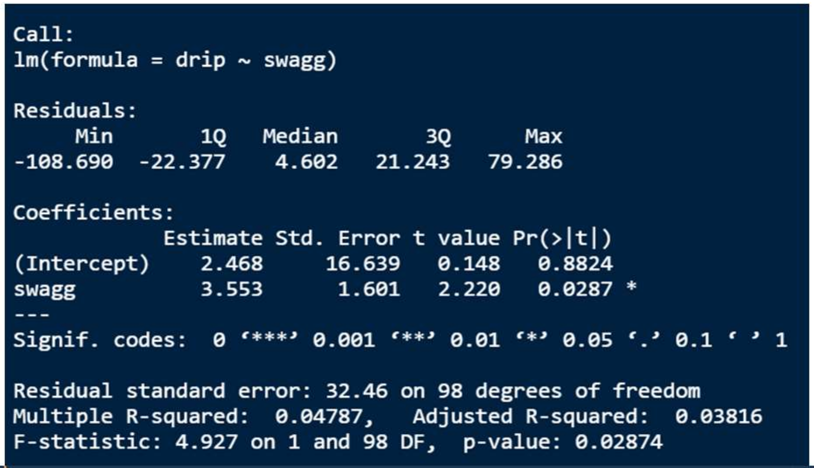
```
Call:
lm(formula = drip ~ swagg)

Residuals:
     Min       1Q   Median       3Q      Max 
-108.690  -22.377    4.602   21.243   79.286 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)    2.468     16.639   0.148   0.8824  
swagg          3.553      1.601   2.220   0.0287 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32.46 on 98 degrees of freedom
Multiple R-squared:  0.04787,	Adjusted R-squared:  0.03816 
F-statistic: 4.927 on 1 and 98 DF,  p-value: 0.02874
```

CLASSICAL

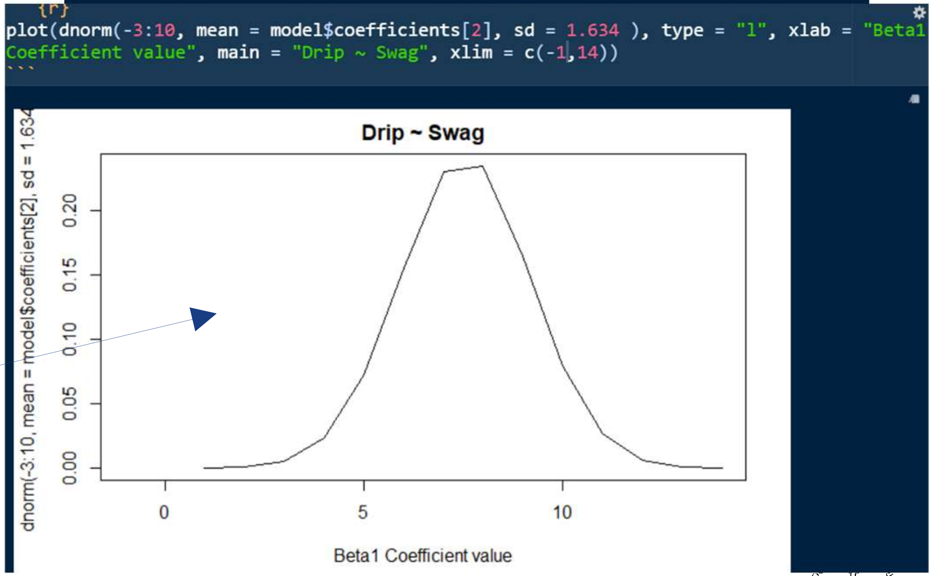
- Don’t just look at the p-value (*).
- Our swag estimate of 3.553 has uncertainty surrounding it. (**Standard Error**)

- Looking at the distribution of our possible beta values is much more informative.
  - While not exclusive to Bayesian statistics this approach is more prevalent in bayes.

Get used to this visualization instead of tables.



```
Call:
lm(formula = drip ~ swagg)

Residuals:
     Min       1Q   Median       3Q      Max 
-108.690  -22.377    4.602   21.243   79.286 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)    2.468     16.639   0.148   0.8824  
swagg          3.553      1.601   2.220   0.0287 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32.46 on 98 degrees of freedom
Multiple R-squared:  0.04787,	Adjusted R-squared:  0.03816 
F-statistic: 4.927 on 1 and 98 DF,  p-value: 0.02874
```



```
{r}
plot(dnorm(-3:10, mean = model$coefficients[2], sd = 1.634 ), type = "l", xlab = "Beta1
Coefficient value", main = "Drip ~ Swag", xlim = c(-1,14))
```
```
```

PROBABILITIES

Bayesian statistics deals a lot with probabilities.

- We won’t got deep into it but appreciate this for a minute.

The paradox of the shark bite.

P(‘dying within two years’|’head bitten off by shark’) = 1

P(’head bitten off by shark’| ‘dying within two years’) = .0001

- Inverting conditional probabilities makes a huge difference.

Classical Framework :

P(“How many died from shark bites **(D)**”| “ People die from shark bites **(H)**”)

we can’t say anything directly about the hypothesis but how well the data conform to the hypothesis.

PROBABILITIES DICE

Hypothesis: I have a fair die

Classical Framework :

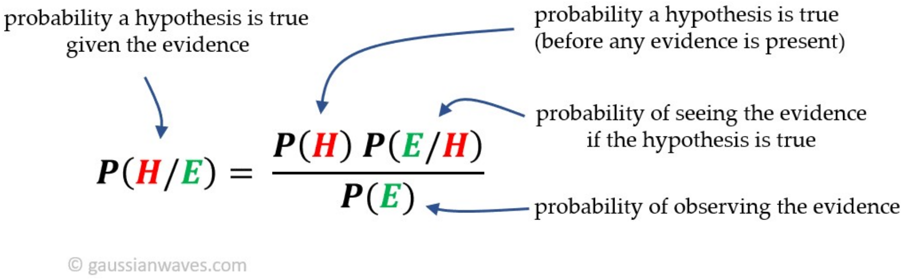
Let us symbolize some data by D and a hypothesis by H. We can talk about P(D|H), the probability of obtaining some data given a hypothesis; for example, P('getting 5 threes in 25 rolls of a die'|'I have a fair die')

how well the data conform to the hypothesis.

Bayesian Framework:

P('I have a fair die'|'I obtained 5 threes in 25 rolls'). Not allowed in classical framework for several reasons.

$$\text{P('I have a fair die'|'I obtained 5 threes in 25 rolls').} = \frac{\text{P('I obtained 5 threes in 25 rolls'|'I've a fair die')} \cdot P('I've\ a\ fair\ die')}{P(\text{'I obtained 5 threes in 25 rolls'})}$$



probability a hypothesis is true given the evidence

probability a hypothesis is true (before any evidence is present)

probability of seeing the evidence if the hypothesis is true

$$P(H/E) = \frac{P(H)\, P(E/H)}{P(E)}$$

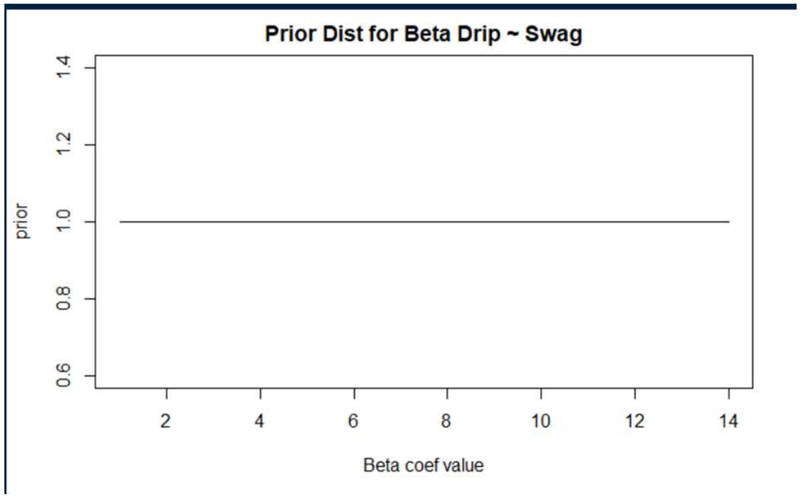
probability of observing the evidence

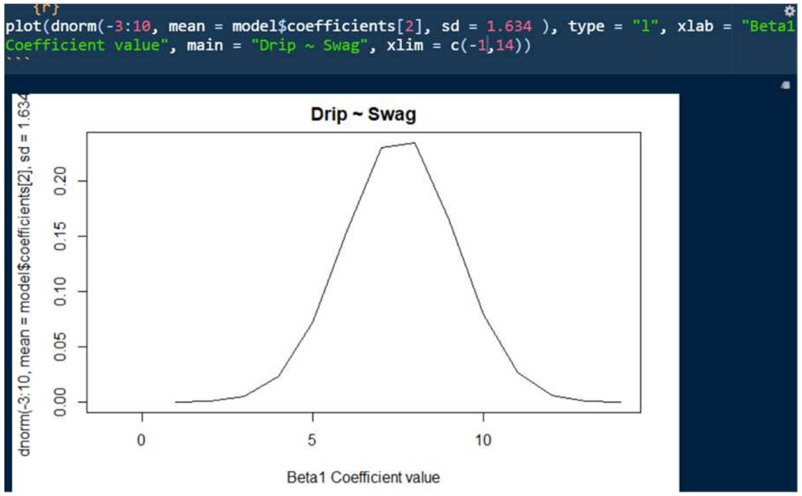
© gaussianwaves.com

BAYESIAN PRIOR AND POSTERIOR:

<u>Bayesian inference involves three steps that go beyond classical estimation.</u>

1. we can include additional information into the model using a *prior distribution.*
2. The data and model are combined to form a *posterior distribution. (Combining our data and prior)*





```
{r}
plot(dnorm(-3:10, mean = model$coefficients[2], sd = 1.634 ), type = "l", xlab = "Beta1
Coefficient value", main = "Drip ~ Swag", xlim = c(-1,14))
```

BAYESIAN:

Bayesian inference involves three steps that go beyond classical estimation.

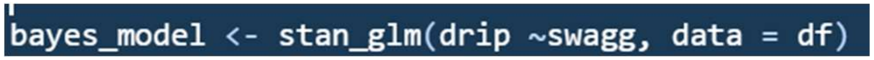
1. we can include additional information into the model using a *prior distribution.*

2. The data and model are combined to form a *posterior distribution.*

3. **we can propagate uncertainty in this distribution—that is, we can get simulation-based *predictions* for unobserved or future outcomes that accounts for uncertainty in the model parameters.**
    - What we will focus on today!

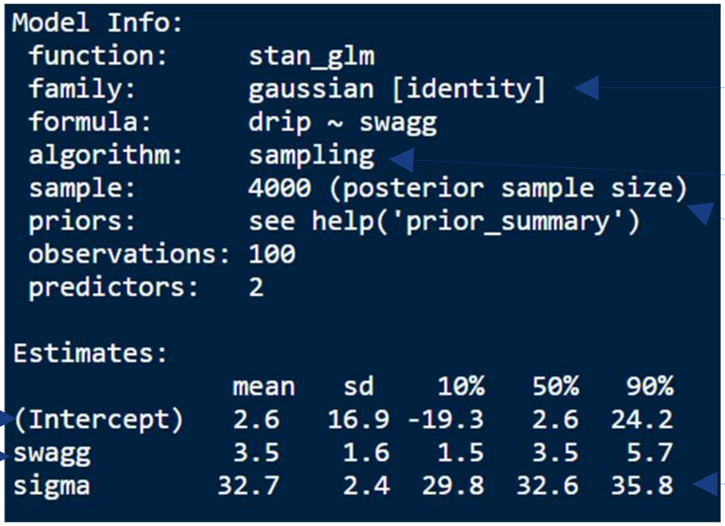
STAN_GLM()

The function for Bayesian:



```
bayes_model <- stan_glm(drip ~swagg, data = df)
```

Output:



```
Model Info:
 function:     stan_glm
 family:       gaussian [identity]
 formula:      drip ~ swagg
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 100
 predictors:   2

Estimates:
              mean   sd    10%   50%   90%
(Intercept)  2.6   16.9 -19.3   2.6  24.2
swagg        3.5    1.6   1.5   3.5   5.7
sigma       32.7    2.4  29.8  32.6  35.8
```

Generalized model with identity link.

It uses sampling.

We've 4000 samples.

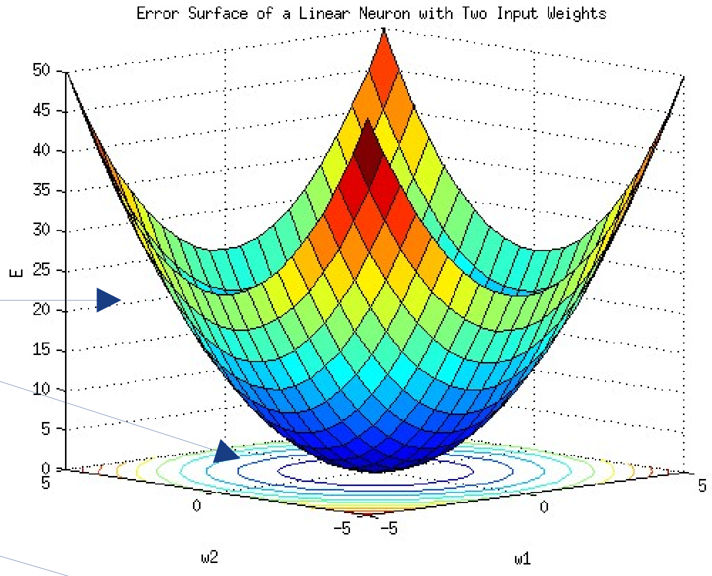
Posterior mean and their SD

Residual standard deviation. What is the spread of the residuals?

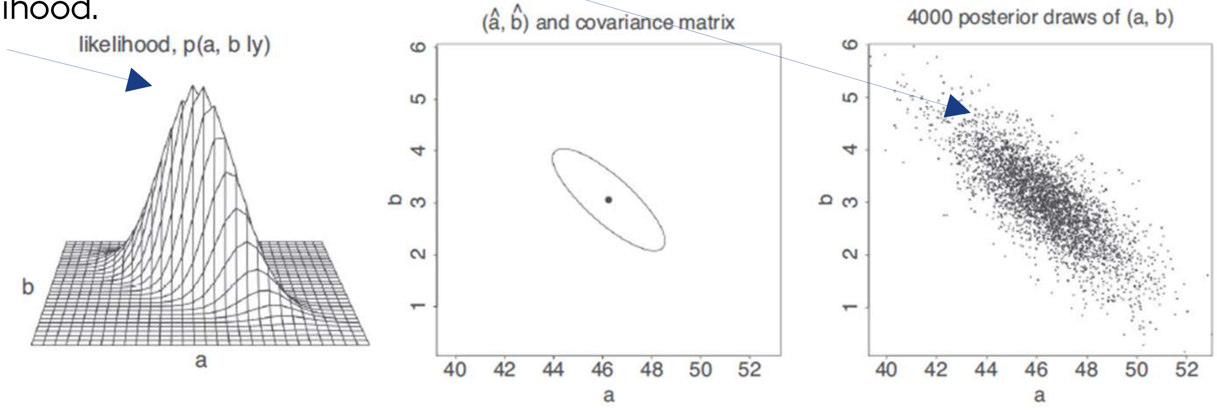
WHAT DOES THE SAMPLING MEAN?

- Last class we build optimizers the minimized our cost/loss function.
  - I.e., Find the θ that minimizes our error.

- We can visualize our cost/loss function in multiple ways.

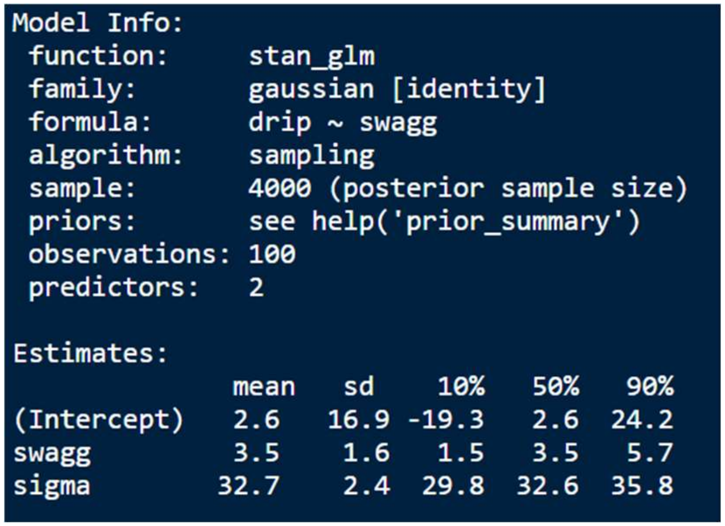


- Bayesian sampling works on some of the same principles. We draw random samples of alpha and beta values based on their likelihood.
  - Most samples around maximum likelihood.

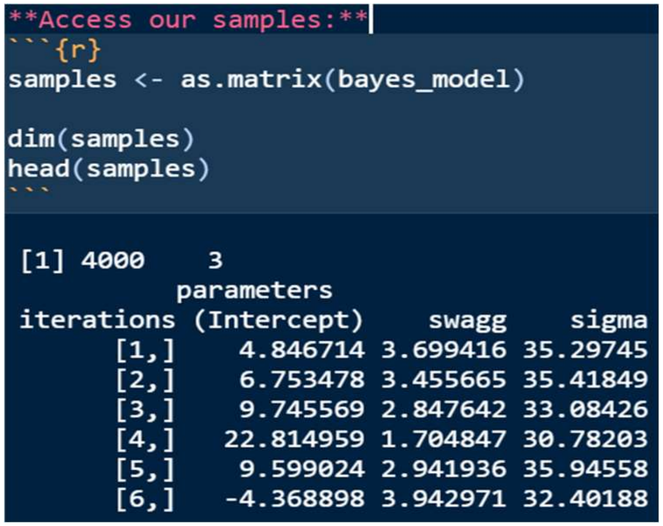


SAMPLING

- We sampled 4000 posterior samples.
  - I.e., we have 4000 different θ.

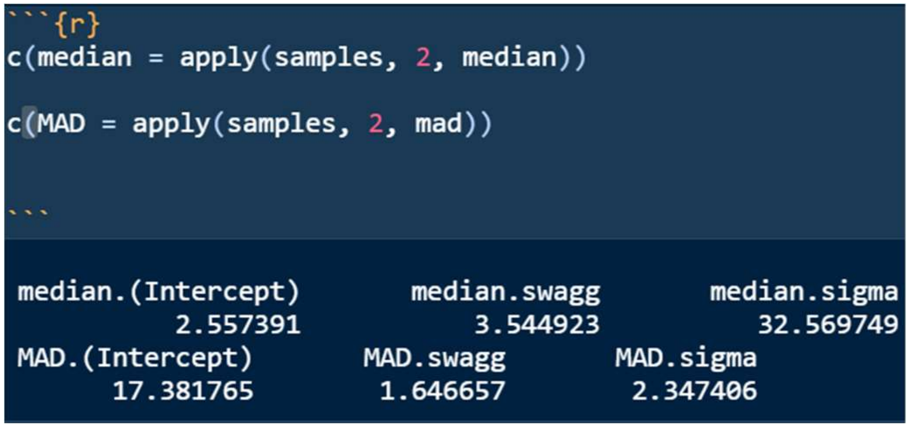


```
Model Info:
 function:     stan_glm
 family:       gaussian [identity]
 formula:      drip ~ swagg
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 100
 predictors:   2

Estimates:
              mean   sd   10%   50%   90%
(Intercept)  2.6   16.9 -19.3   2.6  24.2
swagg        3.5    1.6   1.5   3.5   5.7
sigma       32.7    2.4  29.8  32.6  35.8
```



~~~
**Access our samples:**
```{r}
samples <- as.matrix(bayes_model)

dim(samples)
head(samples)
```

 [1] 4000    3
          parameters
 iterations (Intercept)    swagg    sigma
       [1,]    4.846714 3.699416 35.29745
       [2,]    6.753478 3.455665 35.41849
       [3,]    9.745569 2.847642 33.08426
       [4,]   22.814959 1.704847 30.78203
       [5,]    9.599024 2.941936 35.94558
       [6,]   -4.368898 3.942971 32.40188
~~~

- We can summaries our samples of θ in many ways. But we want to show the uncertainty in our estimate.
  - Summary() uses Median and MAD.
  - Median and MAD can be difficult to visualize, so....

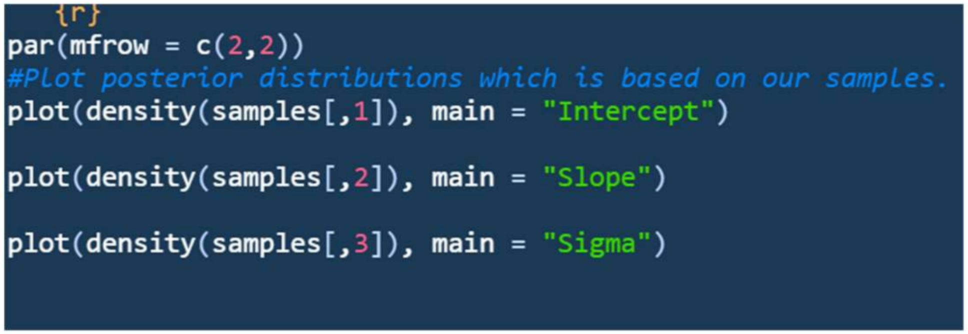


~~~
```{r}
c(median = apply(samples, 2, median))

c(MAD = apply(samples, 2, mad))


```

 median.(Intercept)        median.swagg        median.sigma
           2.557391            3.544923           32.569749
 MAD.(Intercept)        MAD.swagg        MAD.sigma
       17.381765         1.646657         2.347406
~~~

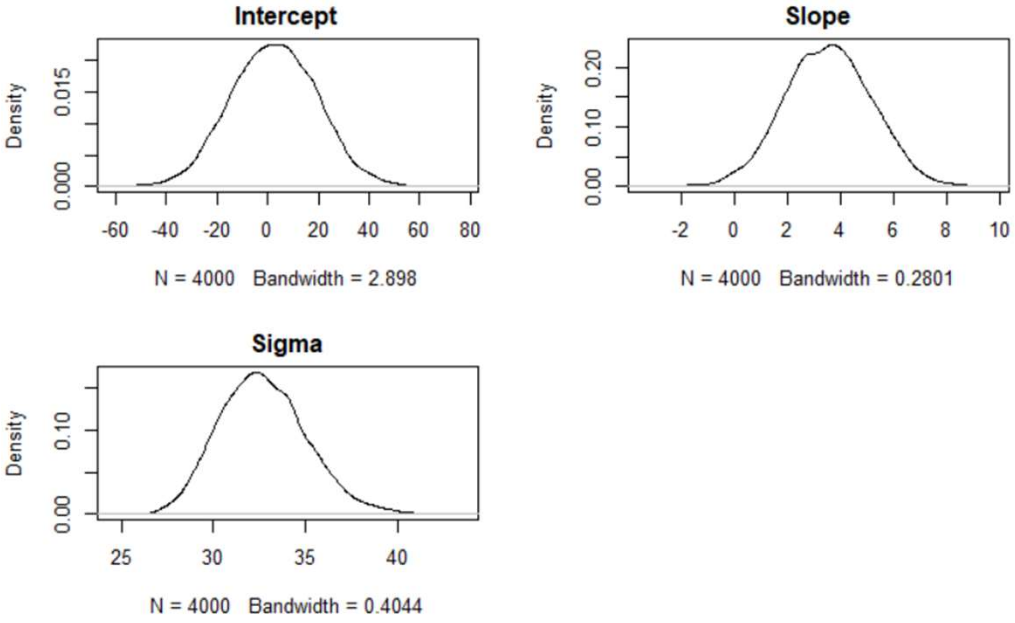
PLOT OUR POSTERIOR

- Plotting the posterior distribution which has the same median, mean and SD and MAD as the summary shows.

- Visualizations are easier to interpret.
  - *Top scientists always plot*.

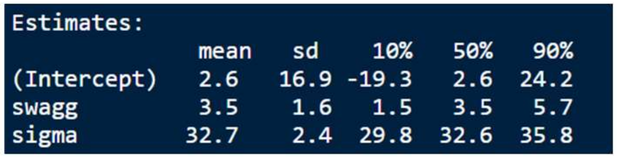


```
   {r}
par(mfrow = c(2,2))
#Plot posterior distributions which is based on our samples.
plot(density(samples[,1]), main = "Intercept")

plot(density(samples[,2]), main = "Slope")

plot(density(samples[,3]), main = "Sigma")
```

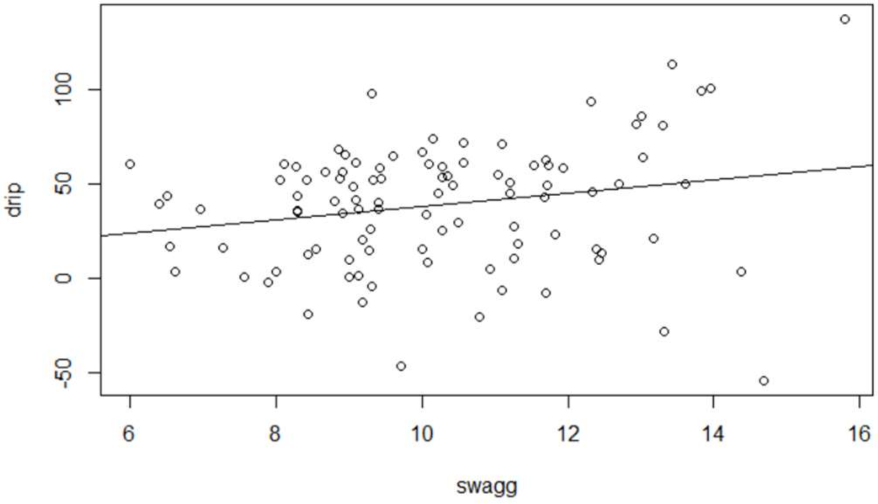


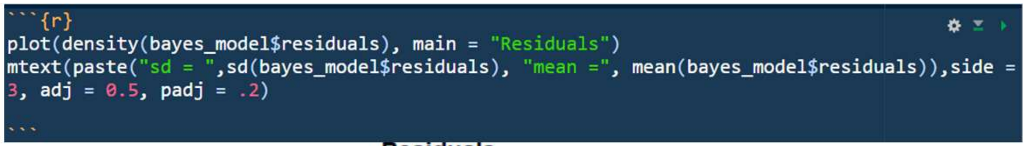
SIGMA / RESIDUAL STD

- Our sigma seemed rather large to let's investigate.
- Sigma is the SD of our error distribution surrounding 0.

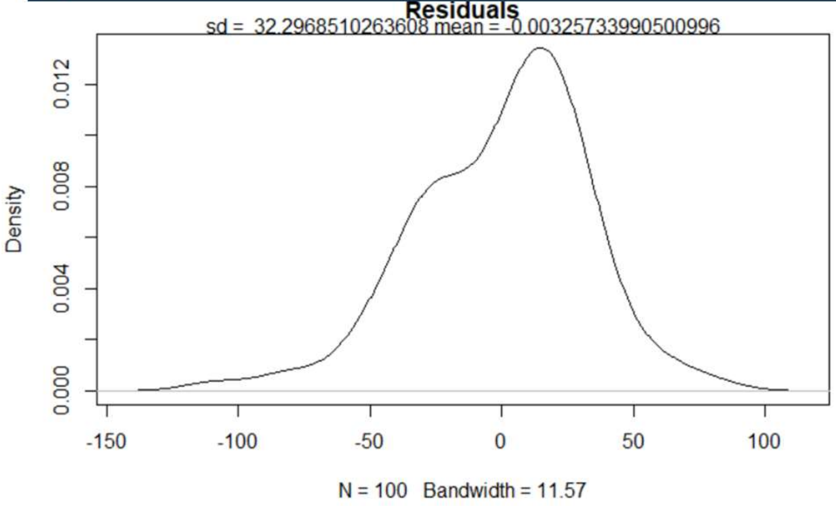


```
Estimates:
              mean   sd   10%   50%   90%
(Intercept)   2.6  16.9 -19.3   2.6  24.2
swagg         3.5   1.6   1.5   3.5   5.7
sigma        32.7   2.4  29.8  32.6  35.8
```





~~~
```{r}
plot(density(bayes_model$residuals), main = "Residuals")
mtext(paste("sd = ",sd(bayes_model$residuals), "mean =", mean(bayes_model$residuals)),side =
3, adj = 0.5, padj = .2)

```
~~~



PREDICTION & UNCERTAINTY.

- The *point prediction*, $\hat{a} + \hat{b} \cdot x_{new}$: Based on the fitted model, this is the best point estimate of the average value of y for new data points with this new value of *x*. We use $\hat{a}$ and $\hat{b}$ here because the point prediction ignores uncertainty.

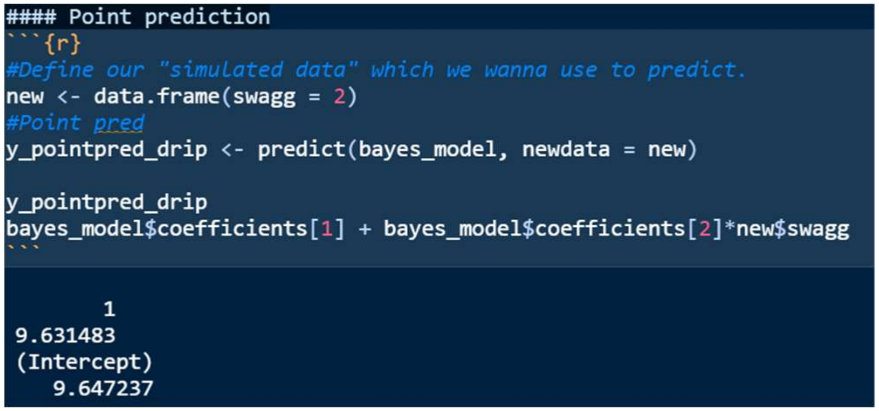
- The *linear predictor with uncertainty*, $a + b \cdot x_{new}$, propagating the inferential uncertainty in (*a*, *b*): This represents the distribution of uncertainty about the expected or average value of y for new data points with predictors *x*new.

- The *predictive distribution for a new observation*, $a + b \cdot x_{new} + error$: This represents uncertainty about a new observation y with predictors *x*new.

POINT PREDICTION

Point predictions disregard any uncertainty around our θ and any error.

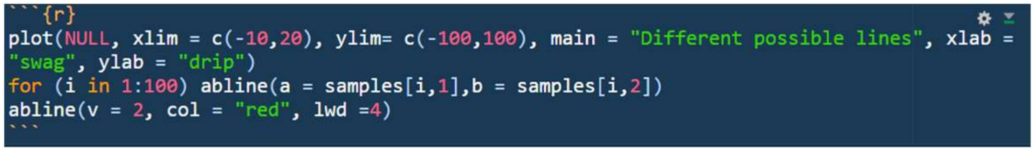
In R:



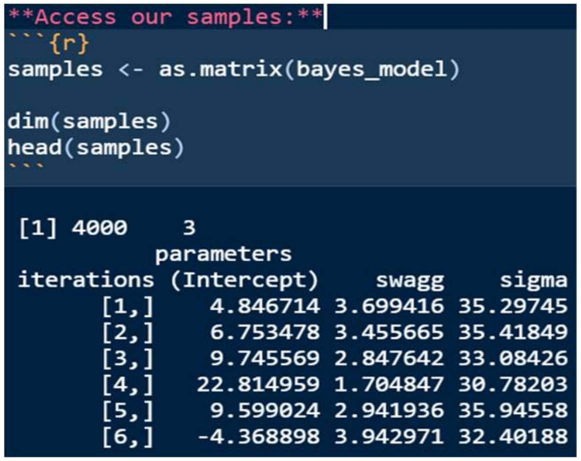
~~~
#### Point prediction
```{r}
#Define our "simulated data" which we wanna use to predict.
new <- data.frame(swagg = 2)
#Point pred
y_pointpred_drip <- predict(bayes_model, newdata = new)

y_pointpred_drip
bayes_model$coefficients[1] + bayes_model$coefficients[2]*new$swagg
```

       1
 9.631483
 (Intercept)
    9.647237
~~~

LINEAR PREDICTION

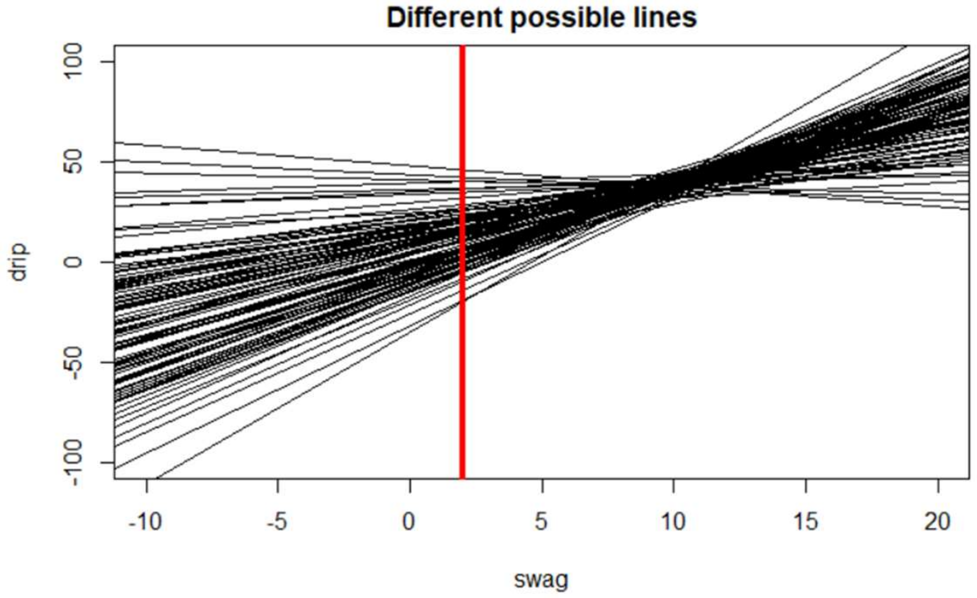
- Linear prediction takes the uncertainty about our coefficients into account.



~~~
```{r}
plot(NULL, xlim = c(-10,20), ylim= c(-100,100), main = "Different possible lines", xlab =
"swag", ylab = "drip")
for (i in 1:100) abline(a = samples[i,1],b = samples[i,2])
abline(v = 2, col = "red", lwd =4)
```
~~~

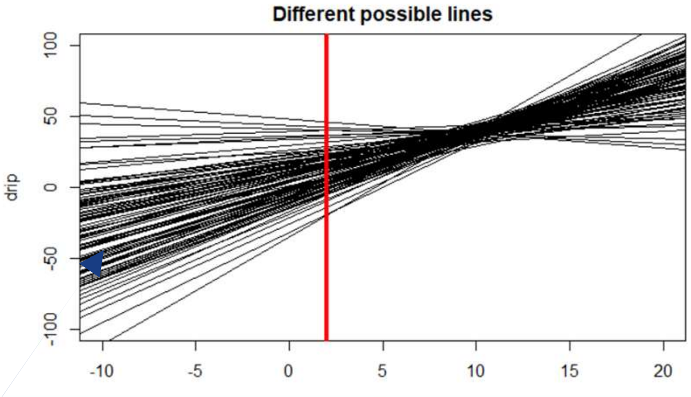


~~~
**Access our samples:**
```{r}
samples <- as.matrix(bayes_model)

dim(samples)
head(samples)
```

 [1] 4000    3
           parameters
 iterations (Intercept)    swagg    sigma
       [1,]    4.846714 3.699416 35.29745
       [2,]    6.753478 3.455665 35.41849
       [3,]    9.745569 2.847642 33.08426
       [4,]   22.814959 1.704847 30.78203
       [5,]    9.599024 2.941936 35.94558
       [6,]   -4.368898 3.942971 32.40188
~~~

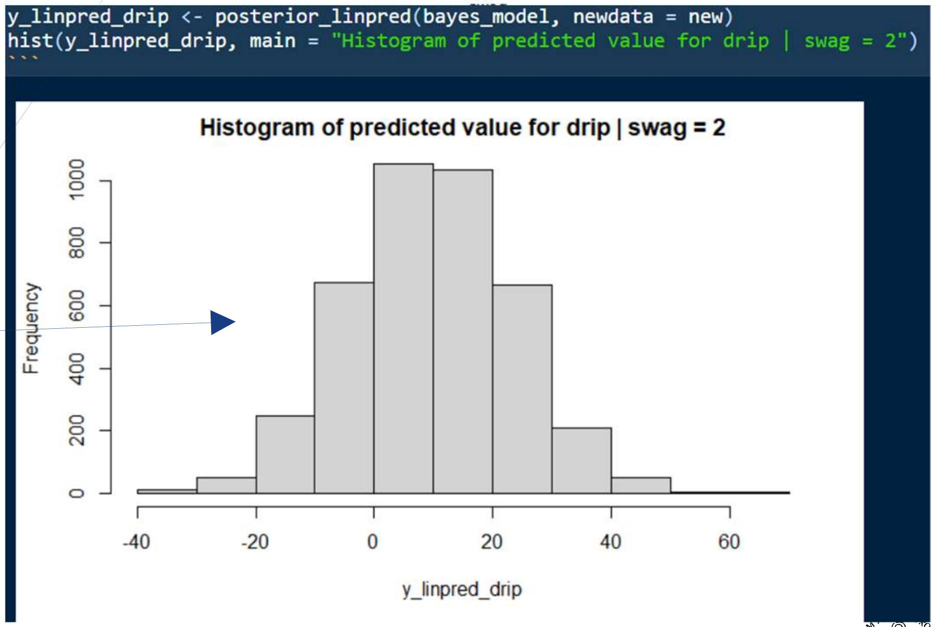


LINEAR PREDICTION

- Linear prediction takes the uncertainty about our coefficients into account.



- In R we calculate the linear prediction using *posterior_linpred(model, newdata = newdata).*
  - Returns a vector/list with predicted y-values given different θ.
  - We can summaries the different y-values as a hist() or density() plot.

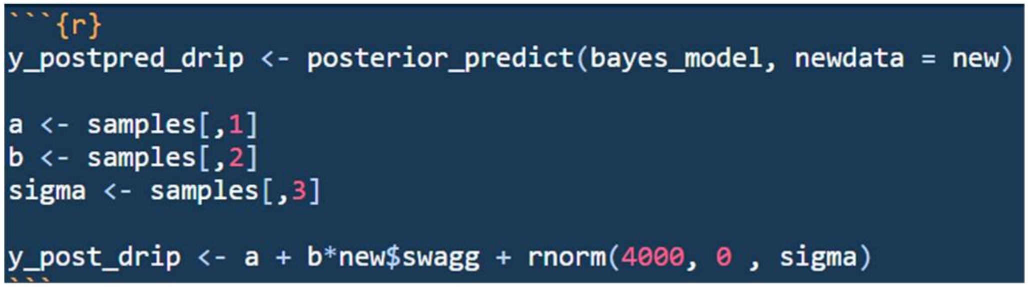


```
y_linpred_drip <- posterior_linpred(bayes_model, newdata = new)
hist(y_linpred_drip, main = "Histogram of predicted value for drip | swag = 2")
```

PREDICTIVE DISTRIBUTION

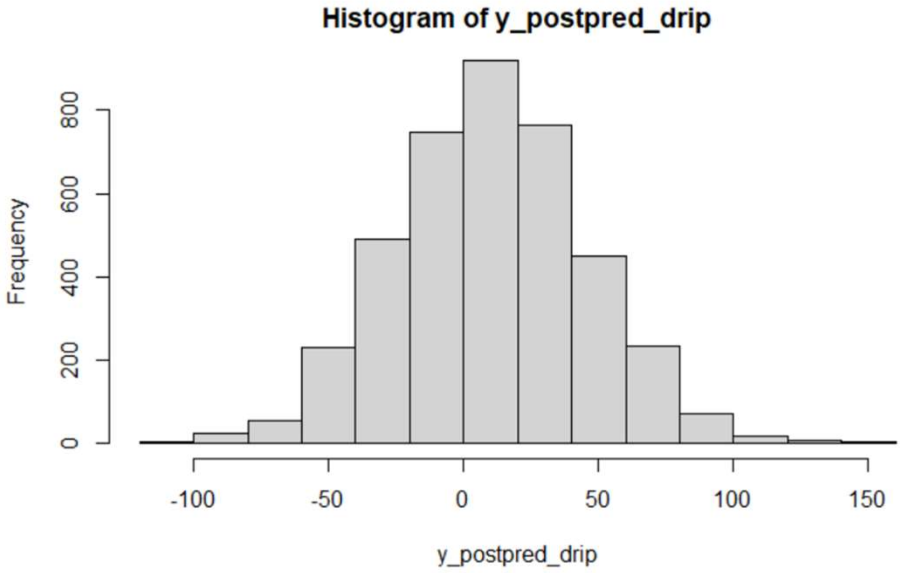
- However, we know that our observed data points isn't straight on the predicted line. Our data points are scattered around the line with some error.
- We can quantify that error using our sigma which denoted the residual standard deviation for our residuals.
  - High sigma = large spread around the linear prediction line.
  - Low sigma = low spread around the linear prediction line.

In R:

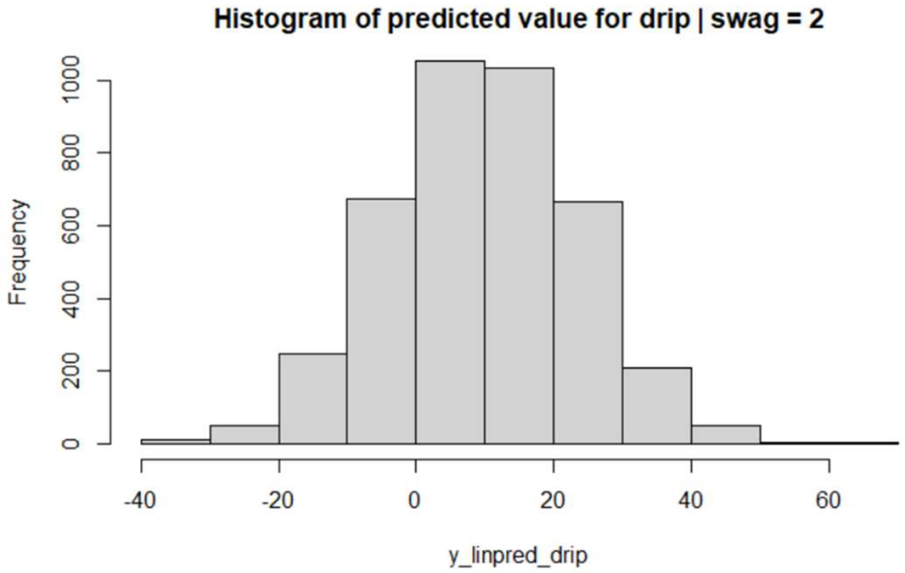


~~~
```{r}
y_postpred_drip <- posterior_predict(bayes_model, newdata = new)

a <- samples[,1]
b <- samples[,2]
sigma <- samples[,3]

y_post_drip <- a + b*new$swagg + rnorm(4000, 0 , sigma)
```
~~~

Posterior Predicton



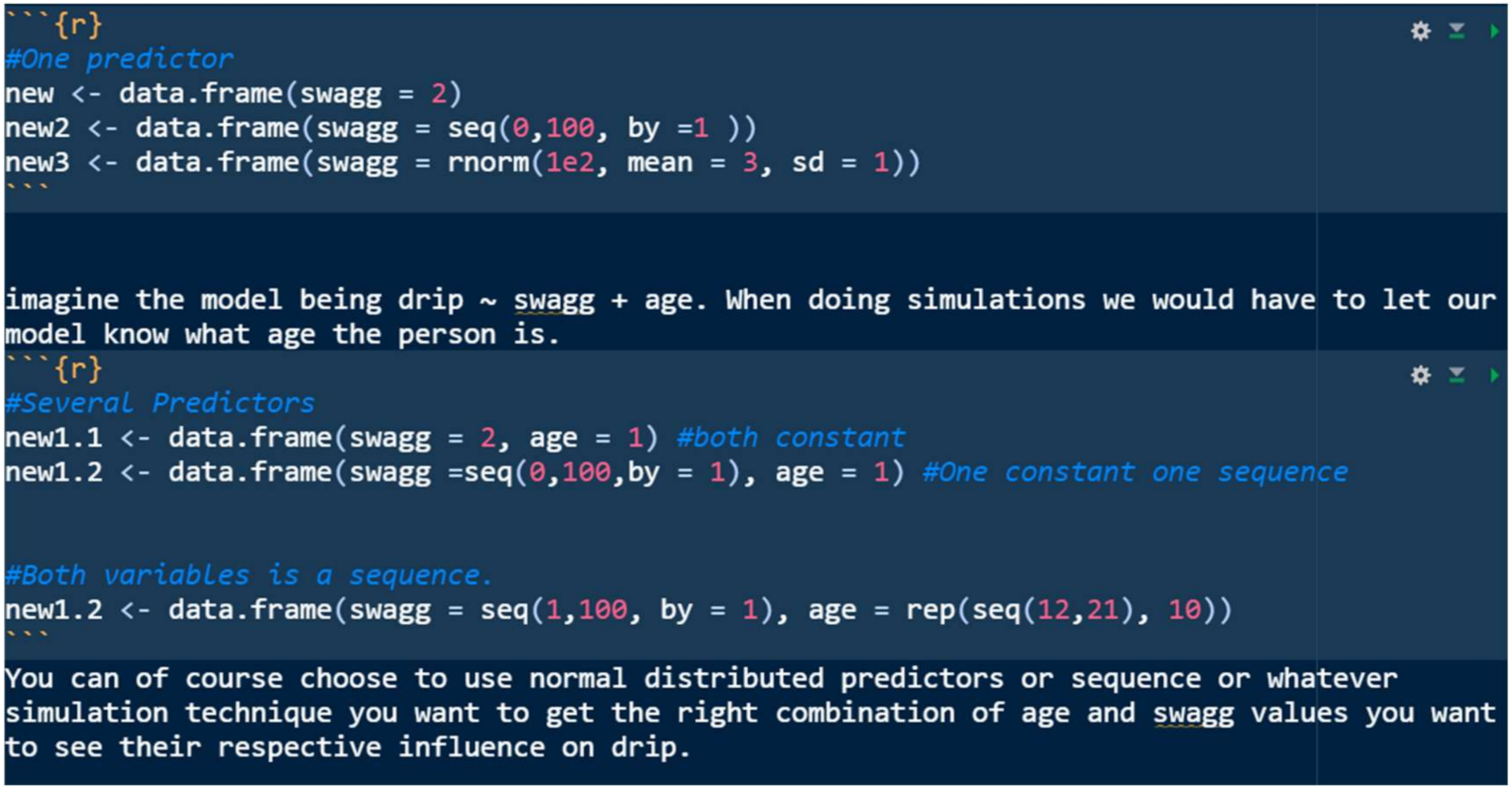
LINEAR PREDICTION



Larger spread in posterior prediction than linear prediction?

Why??

SEQUENCE OF VALUES



~~~
```{r}
#One predictor
new <- data.frame(swagg = 2)
new2 <- data.frame(swagg = seq(0,100, by =1 ))
new3 <- data.frame(swagg = rnorm(1e2, mean = 3, sd = 1))
```
~~~

imagine the model being drip ~ swagg + age. When doing simulations we would have to let our model know what age the person is.

~~~
```{r}
#Several Predictors
new1.1 <- data.frame(swagg = 2, age = 1) #both constant
new1.2 <- data.frame(swagg =seq(0,100,by = 1), age = 1) #One constant one sequence


#Both variables is a sequence.
new1.2 <- data.frame(swagg = seq(1,100, by = 1), age = rep(seq(12,21), 10))
```
~~~

You can of course choose to use normal distributed predictors or sequence or whatever simulation technique you want to get the right combination of age and swagg values you want to see their respective influence on drip.

EXERCISES:

9.2 (hint: check slides)

9.3 (Hint: check 9.1 and 9.2 in the chapter)

9.8

10.1

10.2

10.3 & 10.4

EXTRA Bayesian prior/posterior exercises; 9.5, 9.9 & 9.10