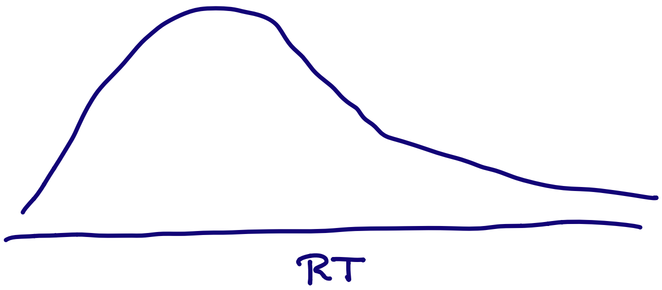
Lecture 7 - the ex-Gaussian model

In the previous lecture, we learned that RT distributions are usually skewed to the right,

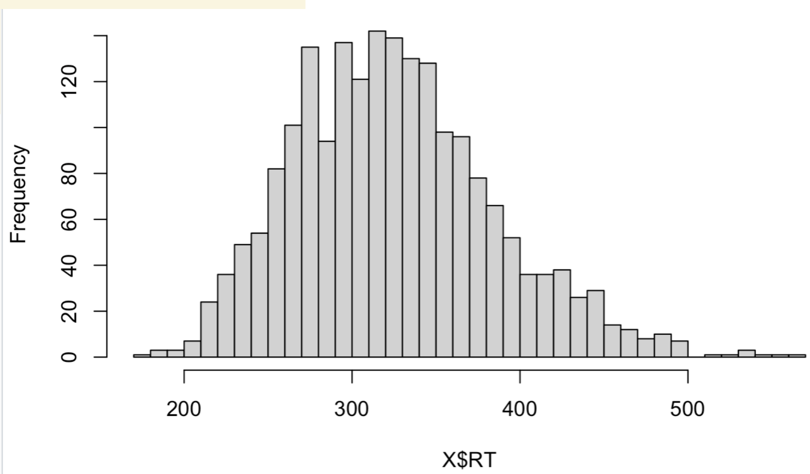
i.e,



Today, we will try to fit some RT data with a normal distribution, then learn how to extend the normal model with a "tail" to account for the positive skew.

The data come from Schwarz (2001), who did a number comparison task

```r
# load Schwarz (2001) data
X = read.csv("https://raw.githubusercontent.com/tomfaulkenberry/courses/master/summer2021/mathpsych/schwarz-A.csv")

# plot histogram of RTs
hist(X$RT, breaks=30)
```



Recall: to fit a model, we'll use MLE

```r
# Step 1: define an objective function (NLL)
nll.normal = function(data, pars){
  mu = pars[1]
  sigma = pars[2]
  return(-sum(log(dnorm(data,
                        mean = mu,
                        sd = sigma))))

}

# Step 2: use sample mean/sd as initial parameters for mu/sigma
initPar = c(mean(X$RT), sd(X$RT))

# Step 3: perform optimization
model1 = optim(par = initPar,
               fn = nll.normal,
               data = X$RT)
```

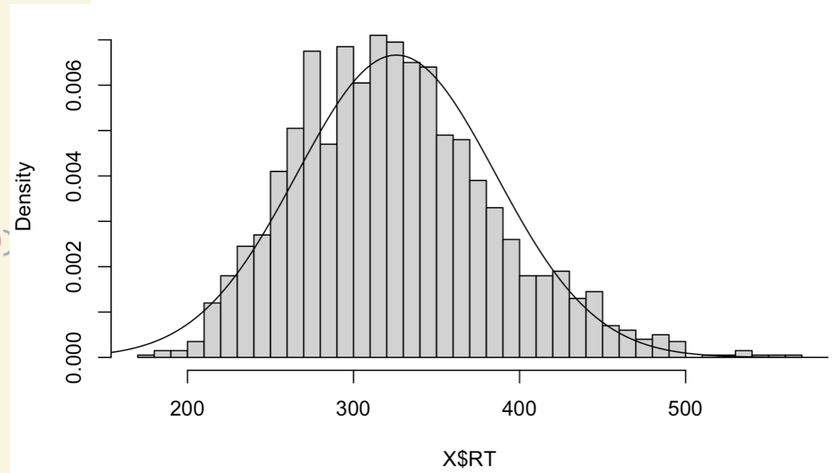
Normal model has two parameters:

μ = "mean"

σ = "standard deviation"

Once we fit the model, let's see how it fits:

```r
# extract parameters
mu = model1$par[1]
sigma = model1$par[2]

# plot model against raw data
hist(X$RT, breaks = 30, probability = T)
x = seq(from = 0, to = 600, length.out=200)
lines(x, dnorm(x, mean=mu, sd=sigma))

# compute BIC
k = 2 # two parameters
N = length(X$RT)
BIC1 = k*log(N) + 2*model1$value
```

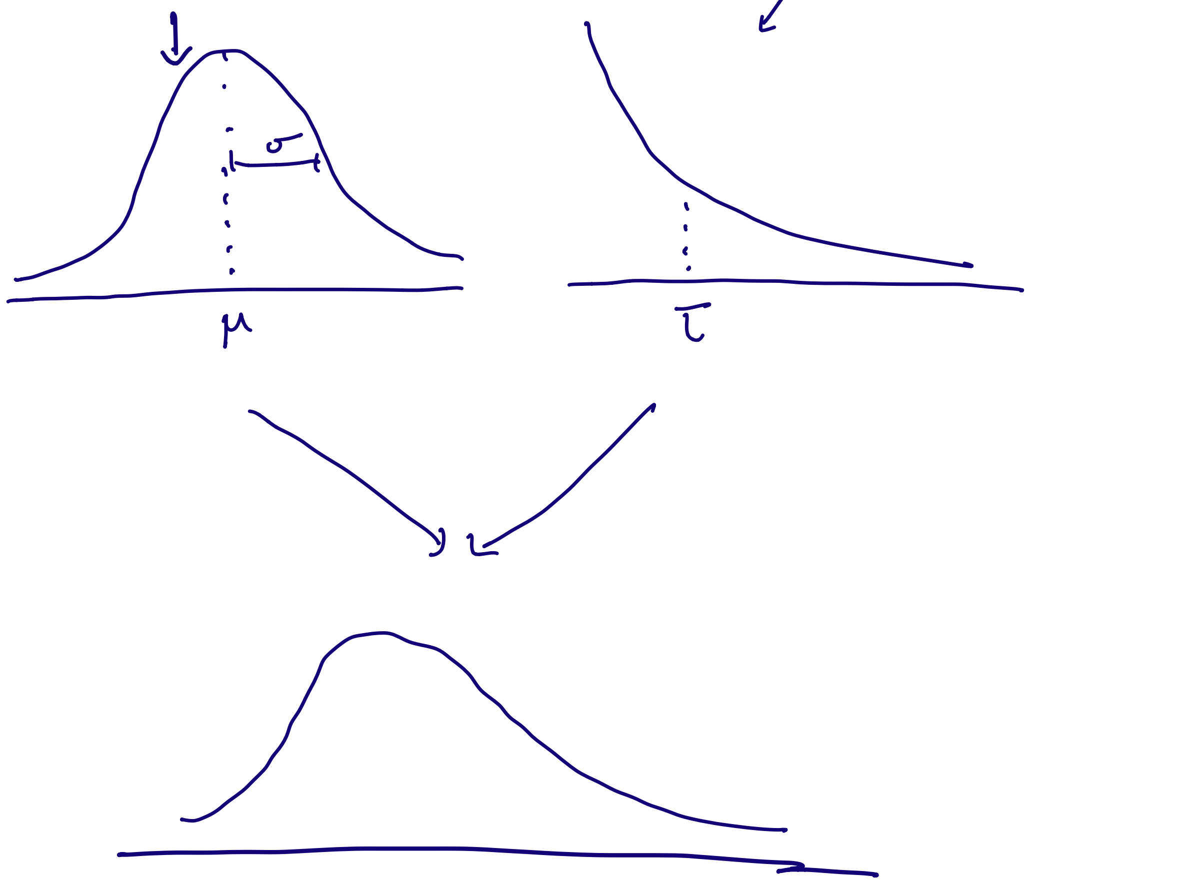


recall: $BIC = k \log N - 2 \log \hat{L}$

Since optim minimizes negative log likelihood, the arithmetic becomes + in the code.

The ex-Gaussian model

idea: combine normal distribution with exponential tail



Three parameters:

* μ = mean of normal component
* σ = sd of normal component
* τ = mean of "tail" component

Density function:

$$f(x \mid \mu, \sigma, \tau) = \frac{1}{\tau\sqrt{2\pi}} \exp\left(\frac{\sigma^2}{2\tau^2} - \frac{x-\mu}{\tau}\right) \int_{-\infty}^{\frac{x-\mu}{\sigma} - \frac{\sigma}{\tau}} \exp\left(\frac{-y^2}{2}\right) dy$$

We can make our own ex-Gaussian density in R:

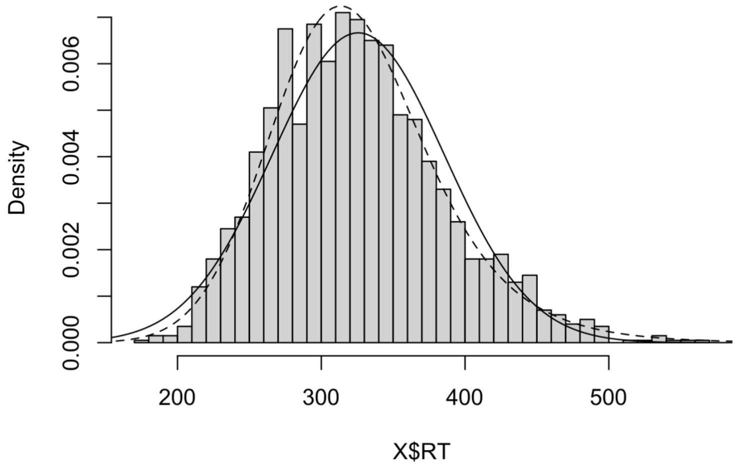
```r
dexg = function(x, mu, sigma, tau){
  return((1/tau)*exp((sigma^2/(2*tau^2))-(x-mu)/tau)*pnorm((x-mu)/sigma-(sigma/tau)))
  }
```

Now perform MLE using the usual methods:

```r
# Step 1: define objective function (NLL)
nll.exg = function(data,pars){
  mu = pars[1]
  sigma = pars[2]
  tau = pars[3]
  return(-sum(log(dexg(data, mu, sigma, tau))))}

# Step 2: function to give initial guess for parameters
# note: this is from Heathcote (2004)
init.exg = function(data){
  require("moments")
  tau = 0.8*sd(data)
  mu = mean(data) - skewness(data)
  sigma = sqrt(var(data)-tau^2)
  return(c(mu, sigma, tau))
}

initPar = init.exg(X$RT)

# Step 3: perform optimization
model2 = optim(par=initPar,
               fn = nll.exg,
               data = X$RT)
```

define a function which gives good guesses for initial values of μ, σ, τ

```r
# extract parameters
mu = model2$par[1]
sigma = model2$par[2]
tau = model2$par[3]

# add to plot to compare with normal fit
lines(x, dexg(x, mu, sigma, tau), lty=2)

# compute BIC
k = 3 # ex-Gaussian has three parameters
N = length(X$RT)
BIC2 = k*log(N) + 2*model2$value
```



much better fit from ex-Gaussian (dashed line) (verify with BIC)