Estimating statistical model parameters with maximum likelihood in R

As was covered in lecture there is a general recipe for estimating statistical model parameters in `R`.

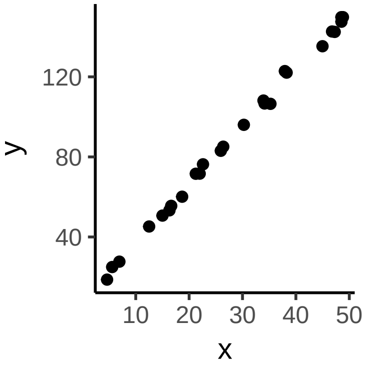
1. load data
2. write a custom likelihood function
3. estimate parameter values by minimizing the negative log likelihood

The code required to accomplish these steps on a simulated data set are below.

```
### Simulating data
# creating a uniformly distributed set of values for an independent variable x
# and values for a variable y that is linearly dependent on x
N=25
x=runif(N,min=0,max=50)
y=3*x+5

# add some "noise" to y and put the variables in a dataframe
y=y+rnorm(N,mean=0,sd=3)
df=data.frame(x=x,y=y)

# plot our observations
library(ggplot2)
ggplot(df,aes(x=x,y=y))+geom_point()+theme_classic()
```



```
### Custom likelihood function
nllike<-function(p,x,y){
  B0=p[1]
  B1=p[2]
  sigma=exp(p[3])

  expected=B0+B1*x

  nll=-sum(dnorm(x=y,mean=expected,sd=sigma,log=TRUE))
  return(nll)
}
```

```r
### estimate parameters by minimizing the negative log likelihood
initialGuess=c(1,1,1)
fit=optim(par=initialGuess,fn=nllike,x=df$x,y=df$y)

# fit is a variable that contains a list describing the result of minimization
# $par is a vector of most likely parameter values in the order specified by
#   your custom likelihood function
# $value is the minimum negative log likelihood found by optim
# $counts gives us the number of parameter combinations tried by optim
# $convergence is a code with information about the optim run; 0 means success

print(fit)
```

```
## $par
## [1] 7.7760671 2.9041521 0.6976173
##
## $value
## [1] 52.92398
##
## $counts
## function gradient
##      236       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```