Another View of Ordinary Regression

PirateGruntTV

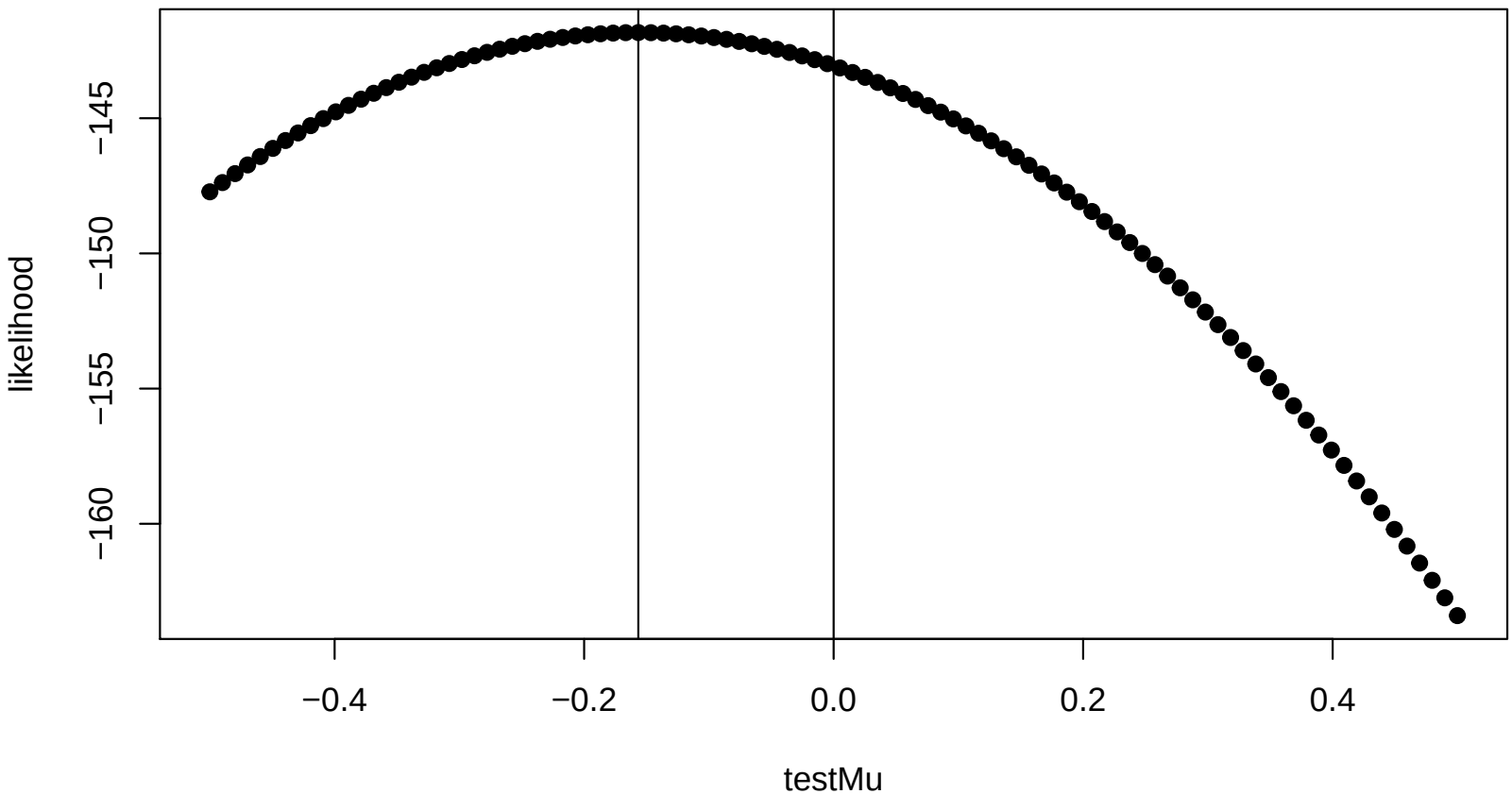
September 30, 2013

Another view of linear regression

```
set.seed(1234)
N = 100
e = rnorm(N, mean = 0, sd = 1)
lnLike = function(x, mu, sigma) {
    n = length(x)
    lnLike = -n/2 * log(2 * pi)
    lnLike = lnLike - n/2 * log(sigma^2)
    lnLike = lnLike - 1/(2 * sigma^2) * sum((x -
        mu)^2)
    lnLike
}
```

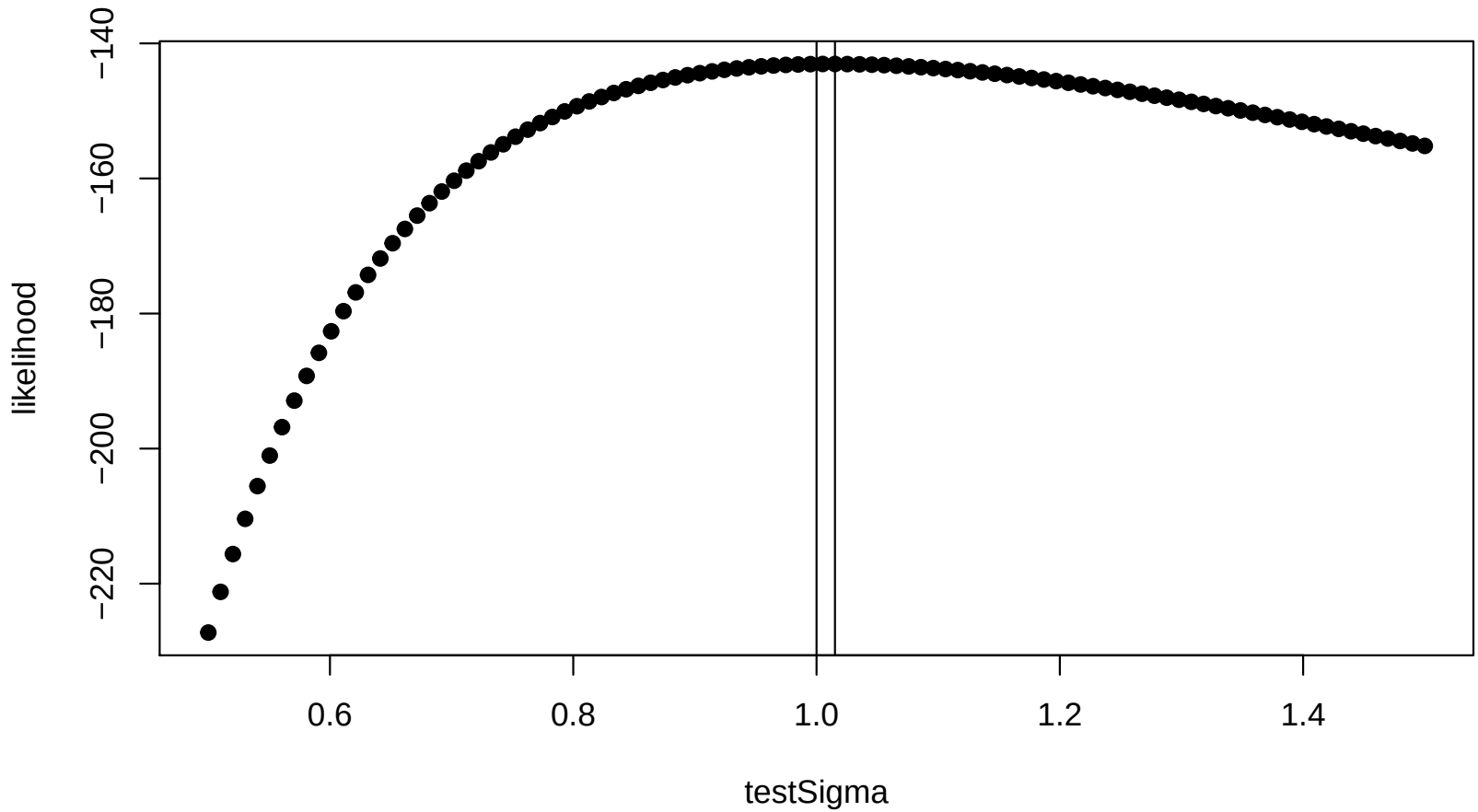
```
testMu = seq(-0.5, 0.5, length.out = 100)
likelihood = sapply(testMu, lnLike, x = e,
    sigma = 1)
testMu[likelihood == max(likelihood)]

## [1] -0.1566
```

```
plot(likelihood ~ testMu, pch = 19)
abline(v = 0)
abline(v = testMu[likelihood == max(likelihood)])
```



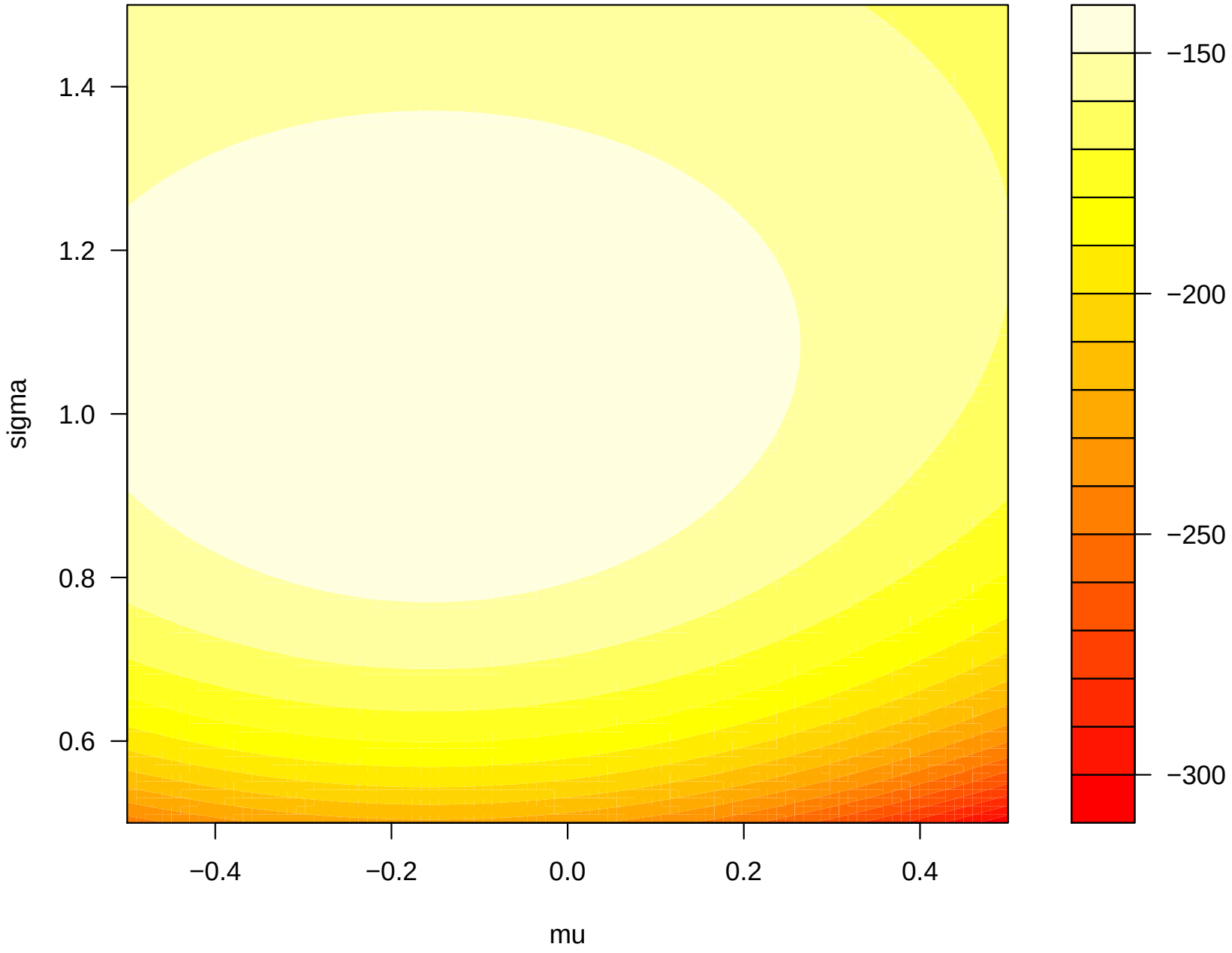
```
testSigma = seq(0.5, 1.5, length.out = 100)
likelihood = sapply(testSigma, lnLike, x = e,
    mu = 0)
testSigma[likelihood == max(likelihood)]

## [1] 1.015
```

```r
plot(likelihood ~ testSigma, pch = 19)
abline(v = 1)
abline(v = testSigma[likelihood == max(likelihood)])
```



```
params = expand.grid(mu = testMu, sigma = testSigma)
params$Likelihood = mapply(lnLike, params$mu,
    params$sigma, MoreArgs = list(x = e))
z = matrix(params$Likelihood, length(testMu),
    length(testSigma))
```

```
filled.contour(x = testMu, y = testSigma,
    z = z, color.palette = heat.colors, xlab = "mu",
    ylab = "sigma")
```


sigma
1.4
1.2
1.0
0.8
0.6
−0.4
−0.2
0.0
0.2
0.4
mu
−150
−200
−250
−300

Optimize for both parameters

```
lnLike2 = function(x, par) {
    mu = par[1]
    sigma = par[2]
    lnLike(x, mu, sigma)
}
optimFit = optim(par = c(-1, 4), fn = lnLike2,
    control = list(fnscale = -1), x = e)
optimFit$par

## [1] -0.1566  0.9994
```

Add a constant term to the normal variable e

```
B0 = 5
Y = B0 + e
```

This is equivalent to lm

```
optimFit = optim(par = c(-1, 4), fn = lnLike2,
    control = list(fnscale = -1), x = Y)
optimFit$par[[1]]

## [1] 4.843

lmFit = lm(Y ~ 1)
lmFit$coefficients[[1]]

## [1] 4.843
```

Now add a slope

```
X = as.double(1:length(e))
B1 = 1.5
Y = B0 + B1 * X + e
lnLike3 = function(par, Y, X) {
    B0 = par[1]
    B1 = par[2]
    sigma = par[3]
    x = Y - B0 - B1 * X
    mu = 0
    lnLike(x, mu, sigma)
}
```

```r
optimFit = optim(par = c(4, 1, 1), fn = lnLike3,
    control = list(fnscale = -1), Y = Y,
    X = X)
optimFit$par[1:2]

## [1] 4.404 1.509

lmFit = lm(Y ~ 1 + X)
lmFit$coefficients

## (Intercept)           X
##       4.406       1.509
```

Why does this matter?

piratetgrunt.com