# STAT 341: Tutorial 4 – Contour Plots and `optim`

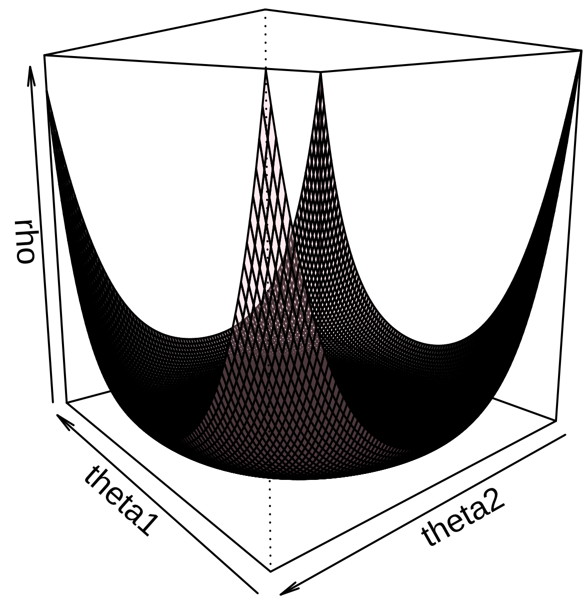
*Friday February 14, 2020*

## Part I: Plot the Mystery Function

$$\rho(\theta_1, \theta_2) = (\theta_1^2 + \theta_2^2 - 1)^3 - \theta_1^2\theta_2^3$$

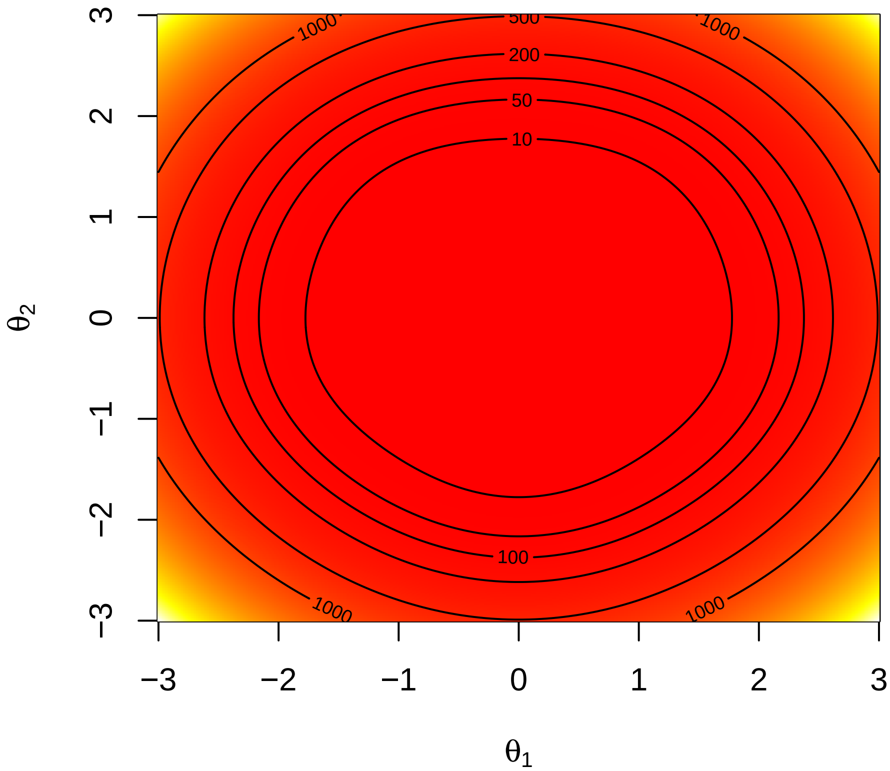
1. Plot the function as a 3D over the range $\theta_1, \theta_2 \in [-3, 3]$.

```
rho <- function(theta1, theta2) {
    (theta1^2 + theta2^2 - 1)^3 - (theta1^2) * (theta2^3)
}
theta1 <- seq(-3, 3, length = 100)
theta2 <- seq(-3, 3, length = 100)
Rho <- outer(theta1, theta2, "rho")
persp(theta1, theta2, Rho, theta = 230, phi = 10, col = adjustcolor("pink",
    alpha = 0.3), xlab = "theta1", ylab = "theta2", zlab = "rho")
```



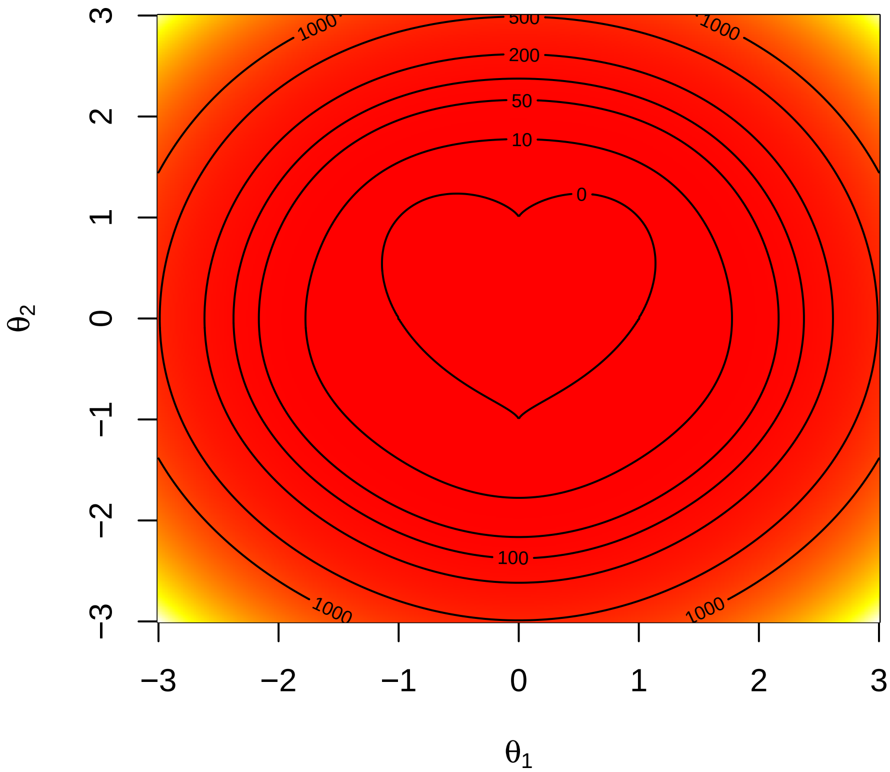
2. Create a 2D contour plot of the function over the range $\theta_1, \theta_2 \in [-3, 3]$.

```
theta1 <- seq(-3, 3, length = 500)
theta2 <- seq(-3, 3, length = 500)
Rho <- outer(theta1, theta2, "rho")
image(theta1, theta2, Rho, col = heat.colors(1000), xlab = bquote(theta[1]),
    ylab = bquote(theta[2]), main = "")
contour(theta1, theta2, Rho, add = T, levels = c(10, 50, 100, 200, 500,
    1000))
```



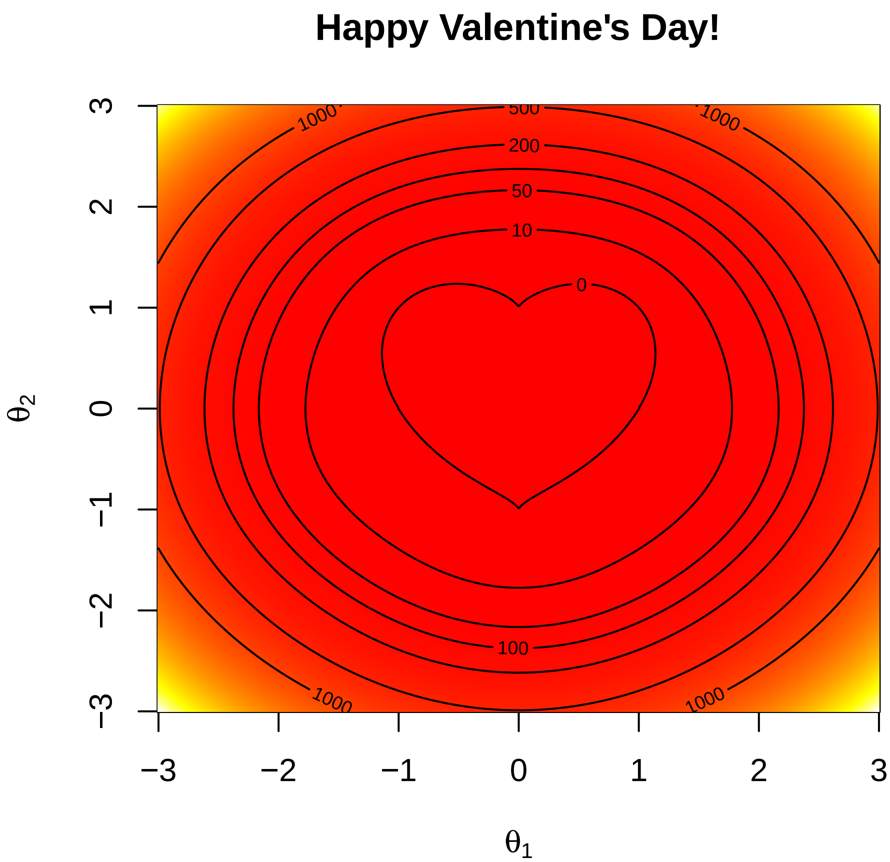
3. Add the 0 contour

```
image(theta1, theta2, Rho, col = heat.colors(1000), xlab = bquote(theta[1]),
    ylab = bquote(theta[2]), main = "")
contour(theta1, theta2, Rho, add = T, levels = c(0, 10, 50, 100, 200, 500,
    1000))
```



4. Add an appropriate plot title

```r
image(theta1, theta2, Rho, col = heat.colors(1000), xlab = bquote(theta[1]),
    ylab = bquote(theta[2]), main = "Happy Valentine's Day!")
contour(theta1, theta2, Rho, add = T, levels = c(0, 10, 50, 100, 200, 500,
    1000))
```



## Part II: Find the Minimum of the Mystery Function

1. Determine the gradient of $\rho(\theta_1, \theta_2)$ and write the corresponding `rho` and `gradient` functions for use with `optim`.

```
rho <- function(theta) {
    theta1 <- theta[1]
    theta2 <- theta[2]
    (theta1^2 + theta2^2 - 1)^3 - (theta1^2) * (theta2^3)
}

grad <- function(theta) {
    theta1 <- theta[1]
    theta2 <- theta[2]
    c(6 * theta1 * (theta1^2 + theta2^2 - 1)^2 - 2 * theta1 * theta2^3,
      6 * theta2 * (theta1^2 + theta2^2 - 1)^2 - 3 * (theta1^2) * (theta2^2))
}
```

$$\nabla\rho(\theta_1,\theta_2) = \begin{bmatrix} \frac{\partial\rho}{\partial\theta_1} \\ \frac{\partial\rho}{\partial\theta_2} \end{bmatrix}$$

where
$$\frac{\partial\rho}{\partial\theta_1} = 3(\theta_1^2+\theta_2^2-1)^2(2\theta_1) - 2\theta_1\theta_2^3$$

$$= 6\theta_1(\theta_1^2+\theta_2^2-1)^2 - 2\theta_1\theta_2^3$$

and

$$\frac{\partial\rho}{\partial\theta_2} = 3(\theta_1^2+\theta_2^2-1)^2(2\theta_2) - 3\theta_1^2\theta_2^2$$

$$= 6\theta_2(\theta_1^2+\theta_2^2-1) - 3\theta_1^2\theta_2^2$$

2. Explore the different optimization `methods` available to you in by checking out `? optim`

3. Using `Nelder-Mead` to find $\text{argmin}_{\boldsymbol{\theta}\in\mathbb{R}}\, \rho(\boldsymbol{\theta})$, and start the algorithm from $\widehat{\boldsymbol{\theta}}_0 = (3, 3)$.

```
optim(par = c(3, 3), fn = rho, gr = grad, method = "Nelder-Mead")
```

```
## $par
## [1] -0.0003073409 -0.0016774222
##
## $value
## [1] -0.9999913
##
## $counts
## function gradient
##       59       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

4. Using `BFGS` to find $\text{argmin}_{\boldsymbol{\theta}\in\mathbb{R}}\, \rho(\boldsymbol{\theta})$, and start the algorithm from $\widehat{\boldsymbol{\theta}}_0 = (3, 3)$.

```
optim(par = c(3, 3), fn = rho, gr = grad, method = "BFGS")
```

```
## $par
## [1] 5.841424e-09 1.386031e-08
##
## $value
## [1] -1
##
## $counts
## function gradient
##       29       13
##
## $convergence
## [1] 0
##
## $message
## NULL
```

5. Using `CG` to find $\text{argmin}_{\boldsymbol{\theta}\in\mathbb{R}}\, \rho(\boldsymbol{\theta})$, and start the algorithm from $\widehat{\boldsymbol{\theta}}_0 = (3, 3)$.

```
optim(par = c(3, 3), fn = rho, gr = grad, method = "CG")
```

```
## $par
## [1] 7.216951e-08 8.968249e-08
##
## $value
## [1] -1
##
## $counts
```

```
## function gradient
##       40       15
##
## $convergence
## [1] 0
##
## $message
## NULL
```

6. Using L-BFGS-B to find $\text{argmin}_{\boldsymbol{\theta}\in\mathbb{R}}\, \rho(\boldsymbol{\theta})$, and start the algorithm from $\widehat{\boldsymbol{\theta}}_0 = (3, 3)$.

```r
optim(par = c(3, 3), fn = rho, gr = grad, method = "L-BFGS-B")
```

```
## $par
## [1] 0.7862829 0.9629960
##
## $value
## [1] -0.3897013
##
## $counts
## function gradient
##       19       19
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

7. Using SANN to find $\text{argmin}_{\boldsymbol{\theta}\in\mathbb{R}}\, \rho(\boldsymbol{\theta})$, and start the algorithm from $\widehat{\boldsymbol{\theta}}_0 = (3, 3)$.

```r
optim(par = c(3, 3), fn = rho, gr = grad, method = "SANN")
```

```
## $par
## [1] 3 3
##
## $value
## [1] 4670
##
## $counts
## function gradient
##    10000       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```