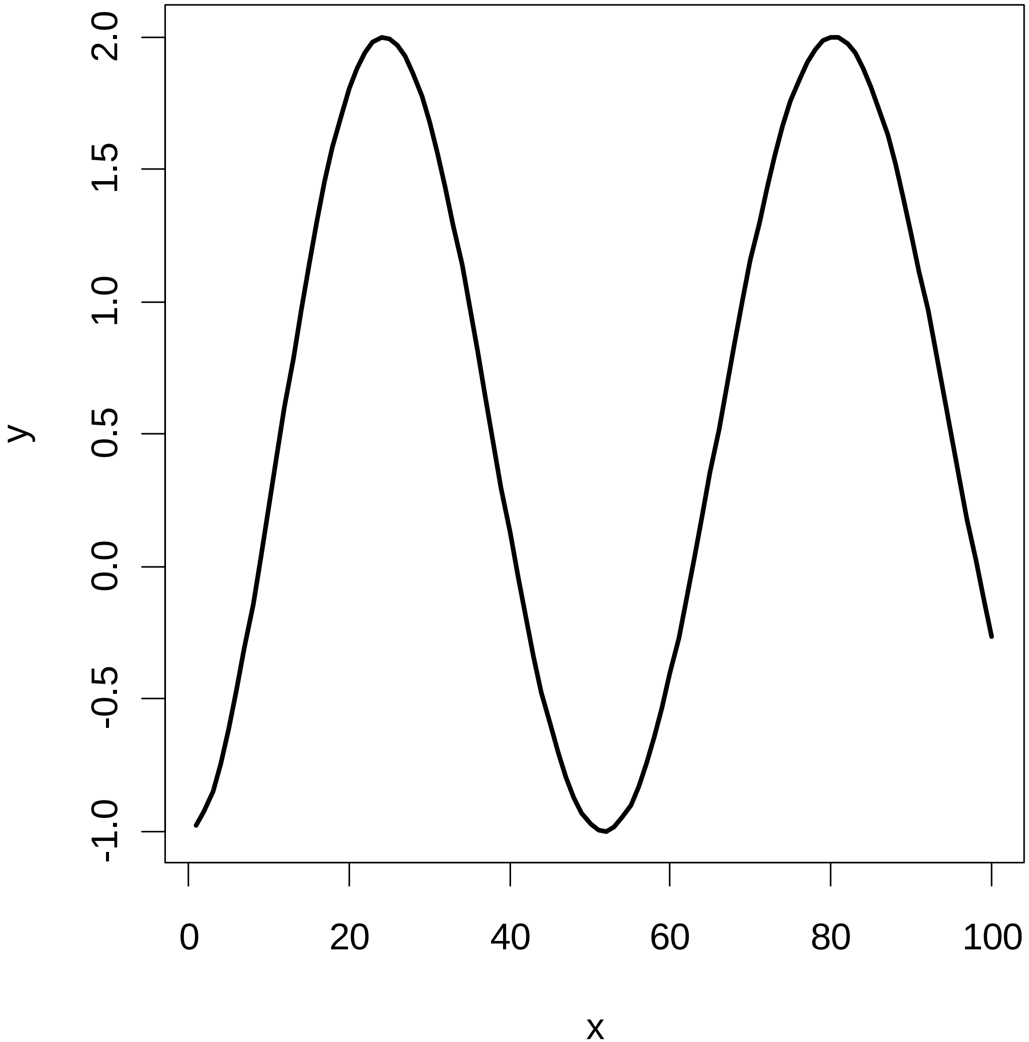
Использование методов численной оптимизации для оценки коэффициентов нелинейной модели

Рассмотрим зависимость

$$y = -1.5 \cdot \cos\frac{x^{0.92}}{6} + 0.5$$



Предположим, что зависимость нам неизвестна, и мы видим только график и имеем числовые векторы x и y

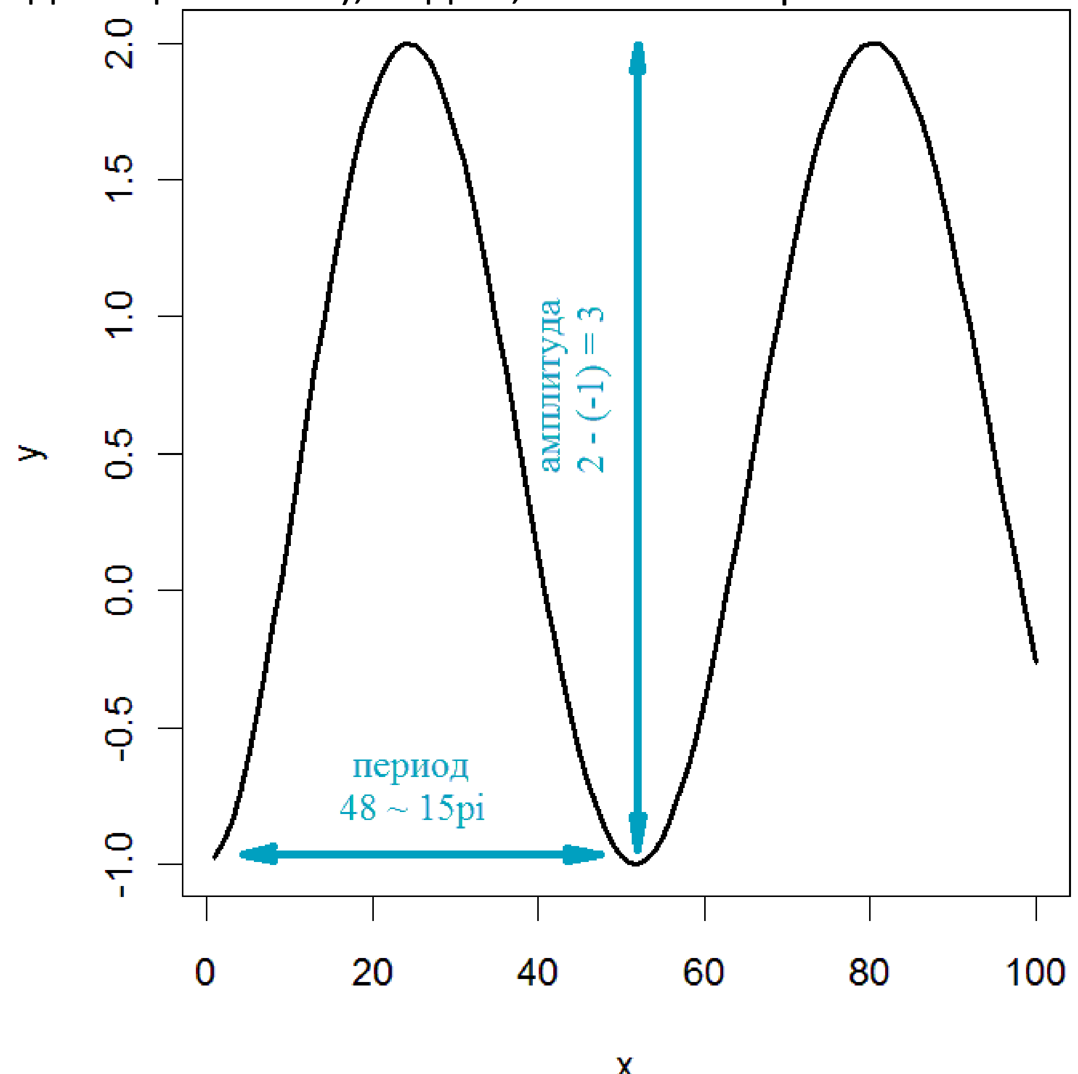
Так как имеют место гармонические колебания можно предположить зависимость вида $f(z) = \sin z$

Измерив период колебаний (ок. 48 единиц или 15π), видим, что он в 7.5 раз больше периода обычного синуса, так что корректируем зависимость:

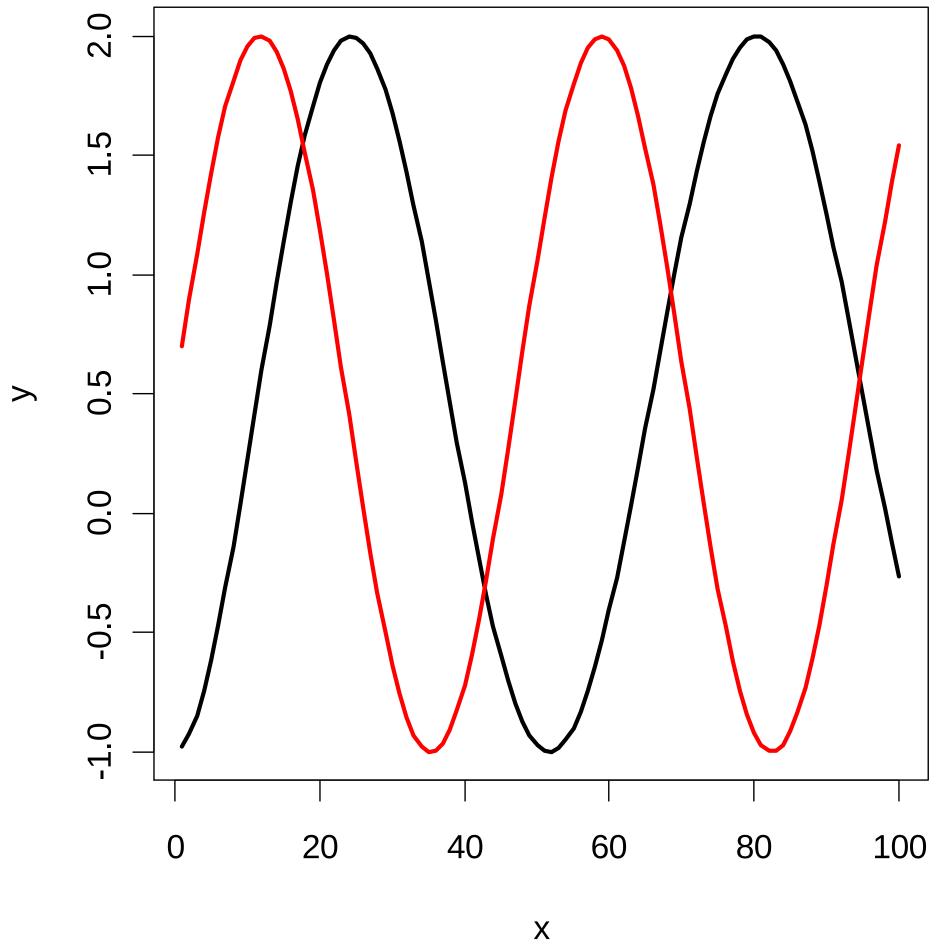
$$f(z) = \sin \frac{z}{7.5}$$

Амплитуда колебаний (3 ед.) в полтора раза шире, чем у синуса, а их центр находится на уровне 0.5, поэтому окончательный вид нашего первого предположения следующий:

$$f(z) = 1.5 \sin \frac{z}{7.5} + 0.5$$



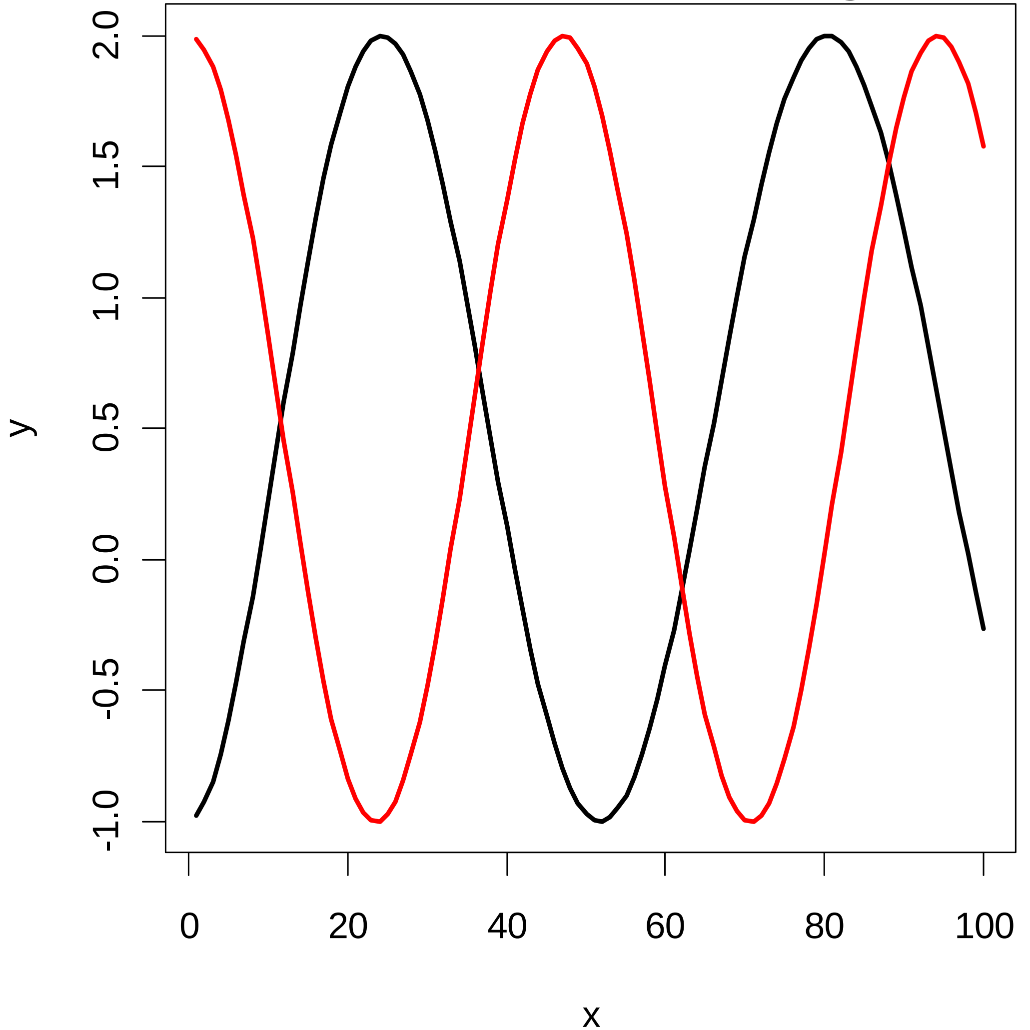
Рисуем график $f(z) = 1.5\sin\frac{z}{7.5} + 0.5$



Не попали в период колебаний

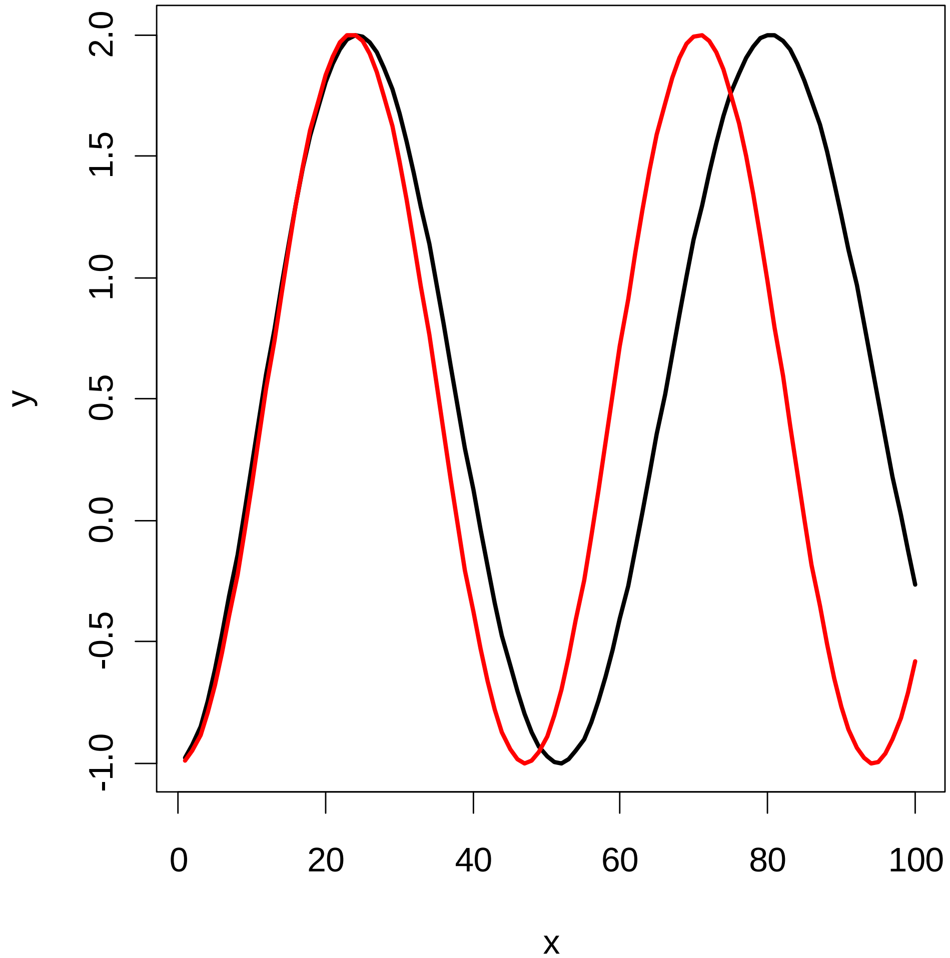
Тут можно вспомнить, что кроме синуса гармонические колебания даёт и косинус

Рисуем график $f(z) = 1.5\,cos\ \frac{z}{7.5} + 0.5$



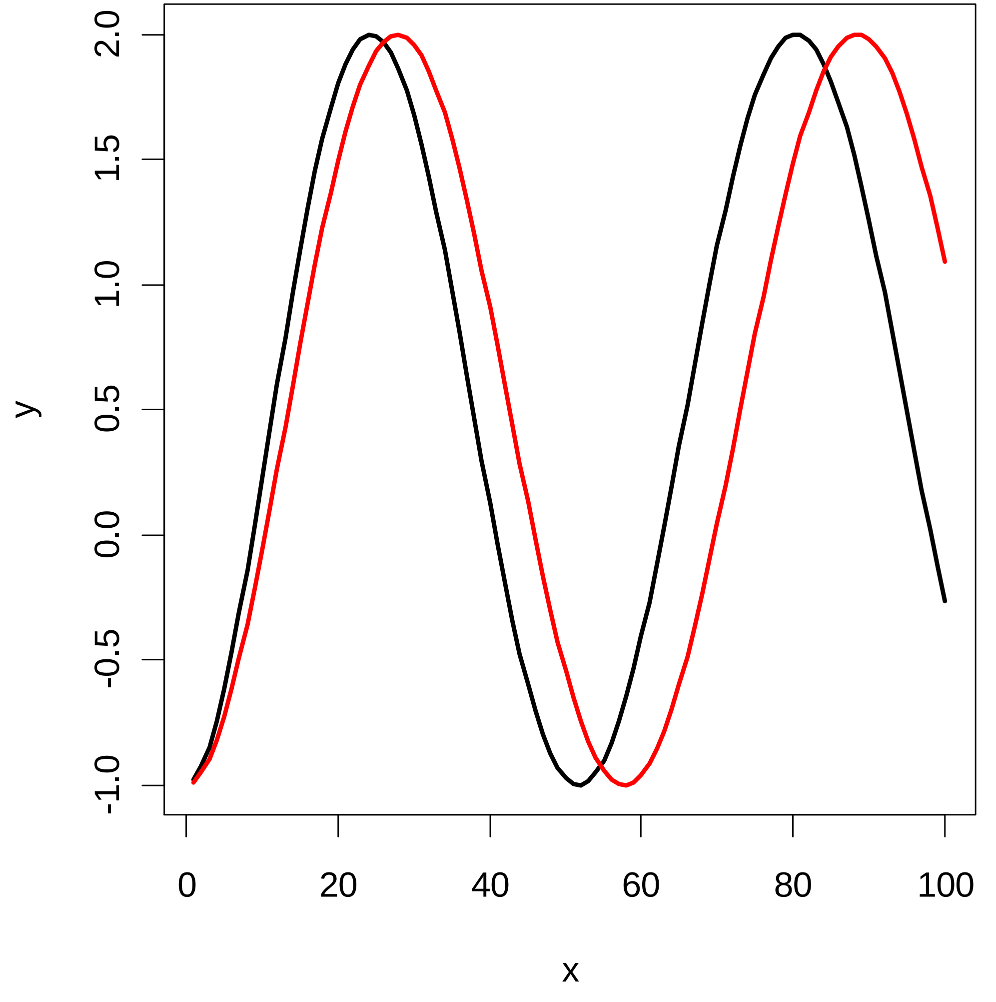
Уже лучше, надо только отразить от горизонтальной оси, для чего впереди приписываем минус

Рисуем график $f(z) = -1.5\, cos\ \frac{z}{7.5} + 0.5$



Период увеличивается с ростом x, значит аргумент под косинусом в степени, меньшей 1 (степень >1 приводит с сокращению периодов)

Рисуем график $f(z) = -1.5\,cos\,\frac{z^{0.95}}{7.5} + 0.5$



Получилось не совсем то, что нужно, но можно пошевелить параметры для достижения лучшей подгонки

Оценим зависимость

$$f(z) = \theta_1 \cdot \cos\frac{z^{\theta_2}}{\theta_3} + \theta_4$$

В качестве критерия качества подгонки используем разность квадратов прогноза и фактического значения

```
mnk <- function(y, y.hat) sum((y - y.hat)^2) / length(y)
```

Оцениваемая модель

```
f <- function(z, theta) theta[1] * cos(z^theta[2] /
theta[3]) + theta[4]
```

Теперь можно записать функцию потерь, которая зависит только от theta

```
err <- function(theta) mnk(y, f(x, theta))
```

Подбираем коэффициенты

Начальные значения параметров, на которых мы остановились

```
theta <- c(-1.5, 0.95, 7.5, 0.5)
```

Оптимизация

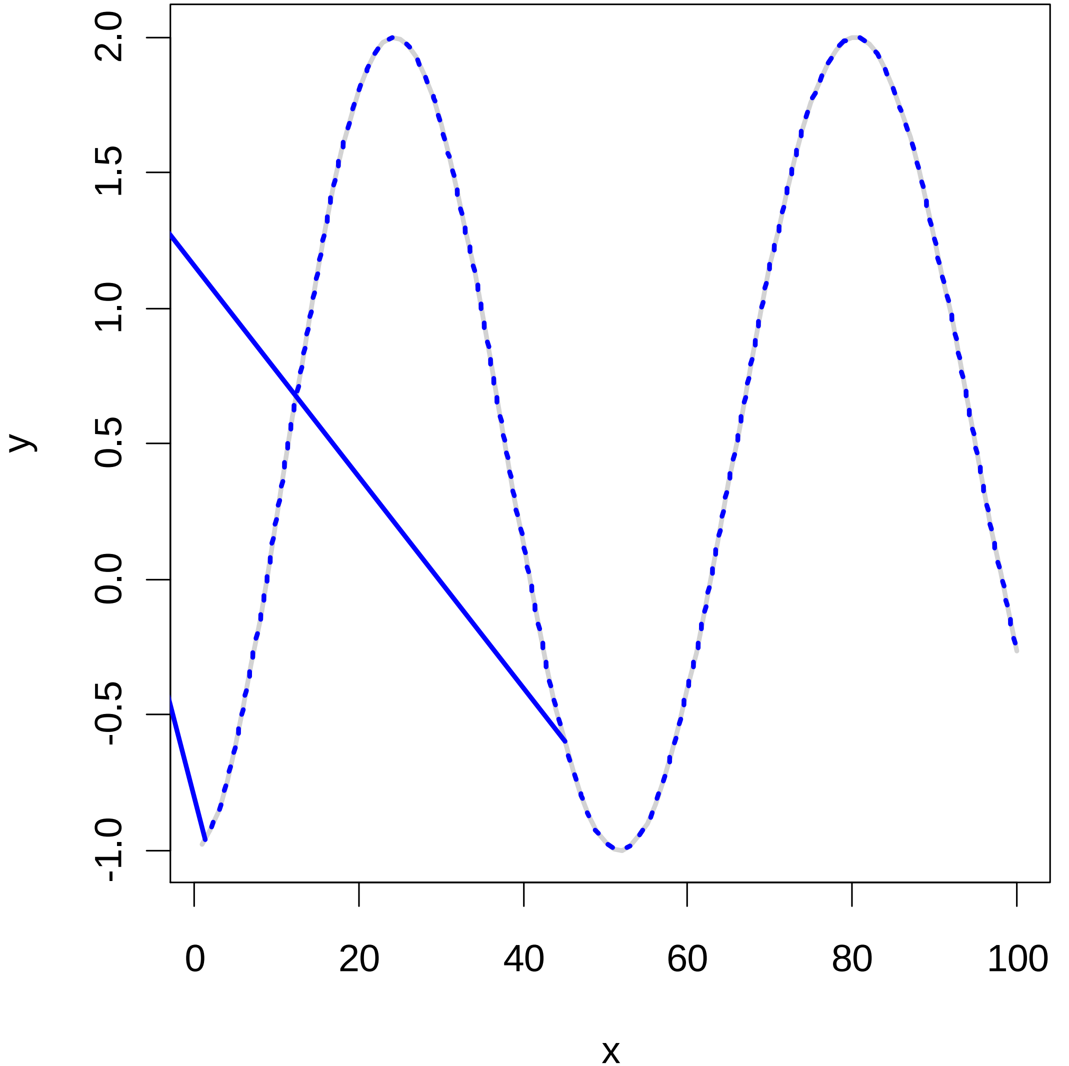
```
opt <- optim(fn = err, par = theta, method = "Nelder-Mead",
control = list(maxit = 5000))


opt
$par
[1] -1.5000931  0.9200136  6.0003736  0.4999908
$value
[1] 6.386102e-09
$counts
function gradient
     431       NA
$convergence
[1] 0
```

- ноль значит, что всё хорошо

```
$message
NULL
```

Рисуем график

```
plot(x, y, type = "l", col = "lightgray")
lines(x, f(x, opt$par), col = "blue")
```



Вопрос

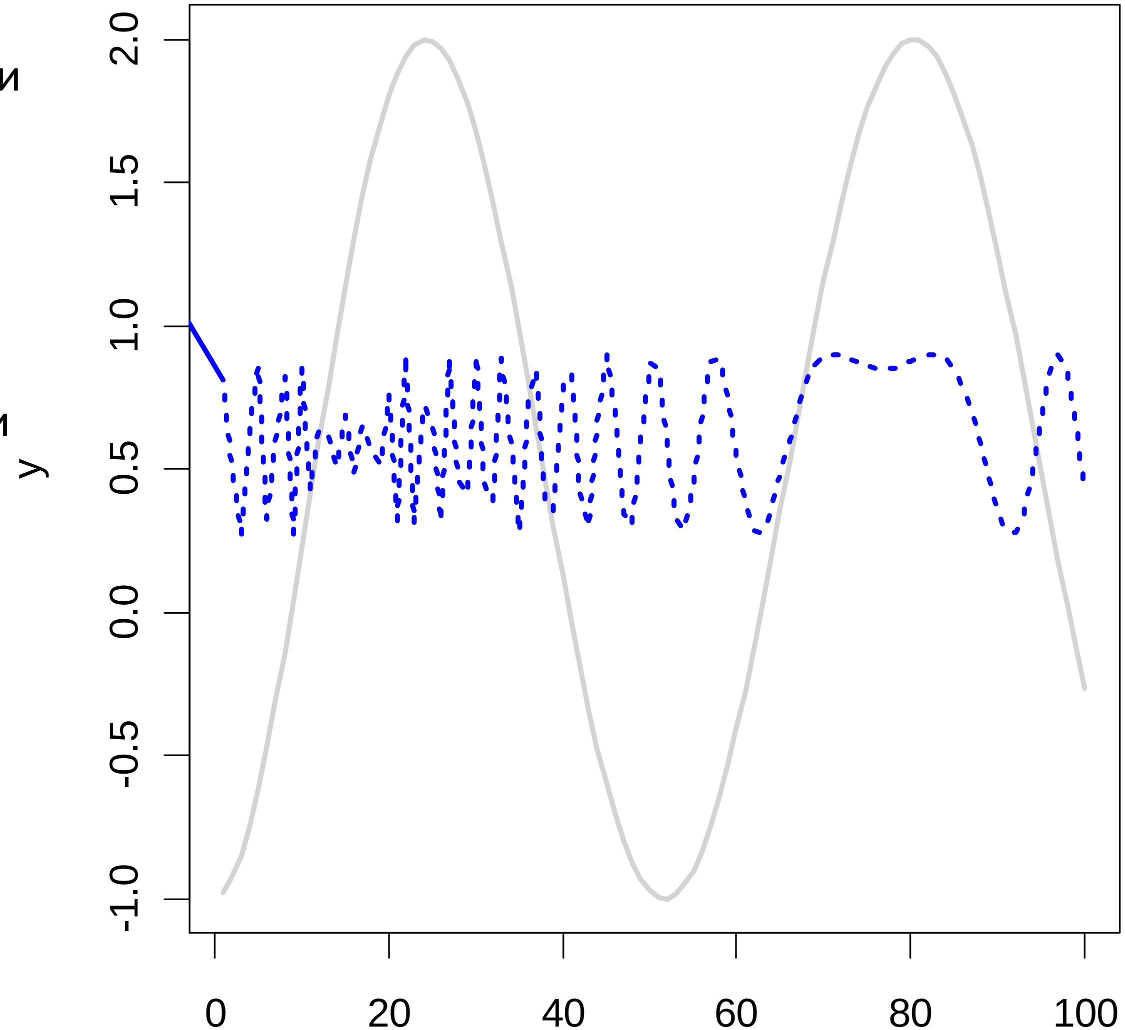
Можно ли не подбирать вручную начальные значения theta, а задать некий произвольный их набор? Например

```
theta <- c(1, 1, 1, 1)
opt <- optim(fn = err, par = theta, method = "Nelder-Mead",
control = list(maxit = 5000))

opt
$par
[1] 0.3152950 1.4299762 1.4716486 0.5866915
$value
[1] 0.9934666
$counts
function gradient
     345       NA
$convergence
[1] 0
$message
NULL
```

Ответ

Нет.

В отличие от классических задач машинного обучения, где функция потерь является строго выпуклой и имеет один локальный минимум, функция потерь в этой задаче имеет множество минимумов. Чтобы найти среди них глобальный, необходимо начать подбор параметров из точки, уже достаточно близкой к этому минимуму. Для этого и нужна ручная подгонка модели



Что дальше?

Подгонка и прогнозирование тренда