Option Greeks Visualized

Arun Sadanand

2025-01-03

1. Option Payoff

The famous Black-Scholes-Metron model provides a simple and elegant approach to value a European option. The value of a call option before maturity is,

$C(S,\tau) = Se^{-d\tau}\Phi(d_1) - Ke^{-r\tau}\Phi(d_2)$

And the value of a put option is,

$P(S,\tau) = -Se^{-d\tau}\Phi(-d_1) + Ke^{-r\tau}\Phi(-d_2)$

In both cases,

$d_1 = \frac{log(\frac{S}{K}) + \left(r - d + \frac{1}{2}\sigma^2\right)\tau}{\sigma\sqrt{\tau}}$

And,

$d_2 = \frac{log(\frac{S}{K}) + \left(r - d - \frac{1}{2}\sigma^2\right)\tau}{\sigma\sqrt{\tau}} = d_1 - \sigma\sqrt{\tau}$

Where,

- S is the price of the underlying stock
- K is the strike price of the option
- τ is the time remaining to maturity of the option
- σ is the volatility of the underlying stock; the Black-Scholes-Metron framework assumes flat volatility across strikes
- r is the risk-free interest rate
- d is the annual dividend rate of the underlying stock
- Φ is the standard normal distribution function

At maturity, when $\tau = 0$, the option value decomposes into the familiar 'hockey-stick' payoff diagram.

$C(S,\tau = 0) = max\left(0, S - K\right)$

And,

$P(S,\tau = 0) = max\left(0, K - S\right)$

Lets examine how an option's value evolves as it approaches maturity:

```
# Set up the parameters for a single stock option

K <- 100 # Strike price of $100
vol <- 0.25 # Implied vol of 25%
```

```r
# We will examine option value across a range of prices for the underlying
# ie. we want to examine the option value at different levels of 'moneyness'
S <- seq(from = K-K/4, to = K+K/4, length.out = 1000)

# We will examine option value as time progresses,
# ie. as time to maturity declines from 3 months to 1 day
taus <- c(12/52, 10/52, 8/52, 6/52, 4/52, 2/52, 5/365, 1/365, 0)
```

```r
source("options.R") # My functions for option price and greeks

call_payoffs <- array(NA, dim = c(length(taus), length(S)))
for(i in seq_along(taus)) {
  call_payoffs[i,] <- opt_bs_price(S, K, taus[i], vol)
}

put_payoffs <- array(NA, dim = c(length(taus), length(S)))
for(i in seq_along(taus)) {
  put_payoffs[i,] <- opt_bs_price(S, K, taus[i], vol, is_call = FALSE)
}
```

```r
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

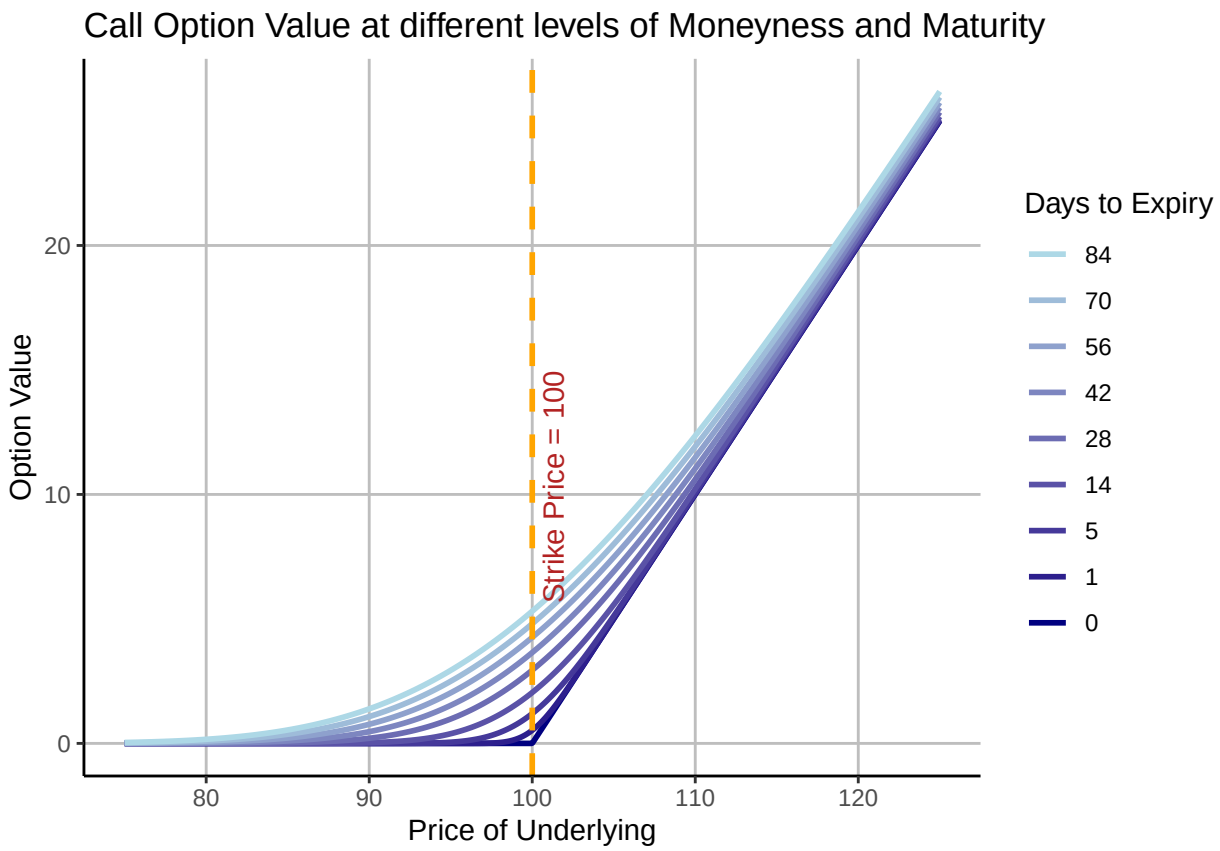
```r
# A function to convert matrix format into 'tidy' format for tidyverse workflows
make_it_tidy <- function(matx, S, taus) {
  df <- as.data.frame(matx)
  colnames(df) <- S
  df$Maturity <- taus
  tdf <- df %>%
    pivot_longer(cols = -Maturity,
                 names_to = "UnderlyingValue",
                 values_to = "OptCalc") %>%
    mutate(UnderlyingValue = as.numeric(UnderlyingValue))
  return(tdf)
}
```

```r
call_payoffs_tdf <- make_it_tidy(call_payoffs, S, taus)
put_payoffs_tdf <- make_it_tidy(put_payoffs, S, taus)
```

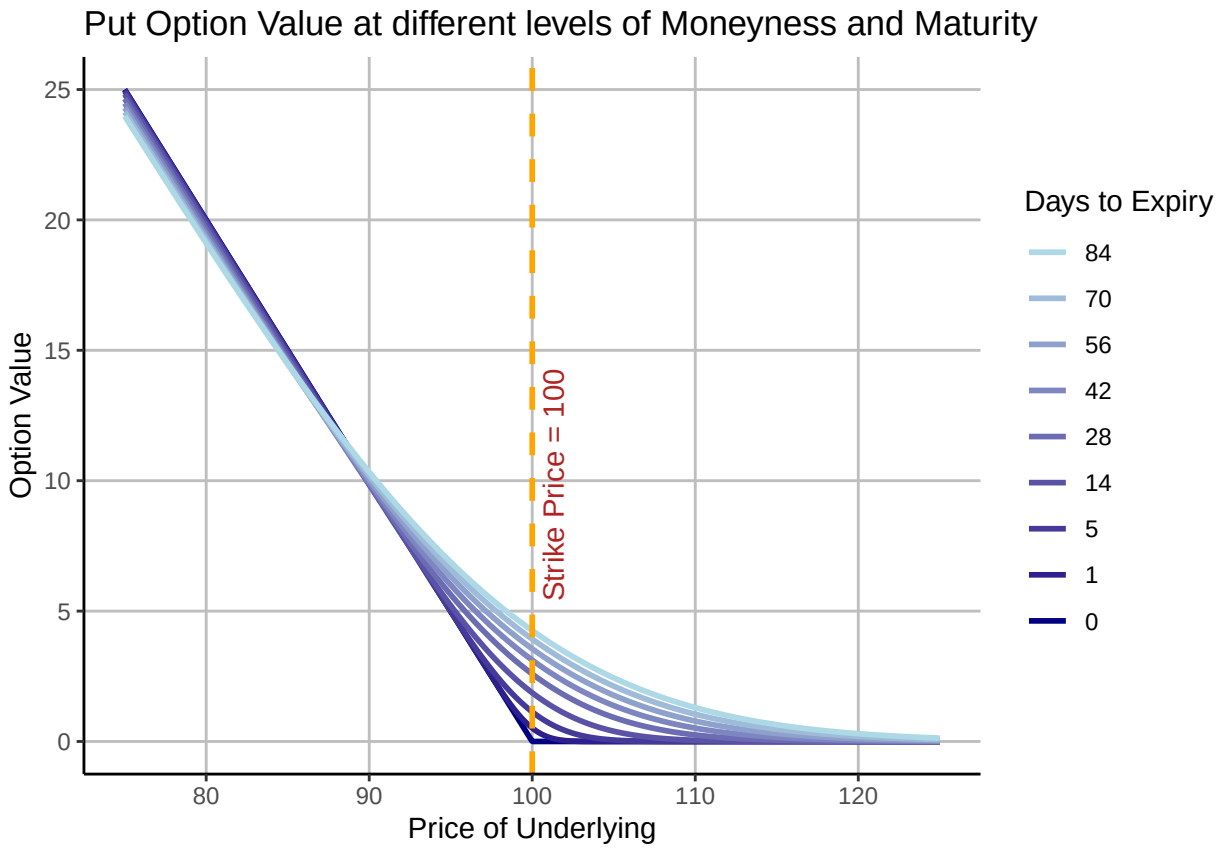
```r
library(ggplot2)
library(glue)
library(scales) # used to pick colors

# A function to make our plot given a tidy dataframe
make_plot <- function(tdf, opt_type, y_label) {
```

```
  breaks <- unique(floor(tdf$Maturity * 365)) # Convert tau to days
  colors <- gradient_n_pal(c("lightblue", "navy"))(seq(0, 1, length.out = length(breaks))) # pick color
  opt_plot <- ggplot(tdf,
                     aes(x = UnderlyingValue,
                         y = OptCalc,
                         group = Maturity,
                         color = factor(floor(Maturity * 365)))) +
    geom_line(linewidth = 1) +
    scale_color_manual(values = colors,
                       breaks = as.character(breaks),
                       labels = breaks) +
    labs(title = glue("{opt_type} {y_label} at different levels of Moneyness and Maturity"),
         x = "Price of Underlying",
         y = glue("{y_label}"),
         color = "Days to Expiry") +
    geom_vline(xintercept = K, linetype = "dashed", color = "orange", linewidth = 1) +
    annotate("text", x = K, y = max(tdf$OptCalc),
             label = glue("Strike Price = {K}"), color = "firebrick", angle = 90,
             hjust = 2.2, vjust = 1.5, size = 4) +
    theme_classic() +
    theme(panel.grid.major = element_line(color = "gray", linewidth = 0.5))
  return(opt_plot)
}
```

```
make_plot(call_payoffs_tdf, "Call", "Option Value")
```



```
make_plot(put_payoffs_tdf, "Put", "Option Value")
```



2. Option Delta

'Delta' measures the sensitivity of the option price to the price of the underlying stock. Differentiating the option value with respect to S tells us what delta should look like:

$\Delta_C = \frac{\partial C(S,\tau)}{\partial S} = e^{-d\tau}\Phi(d_1)$

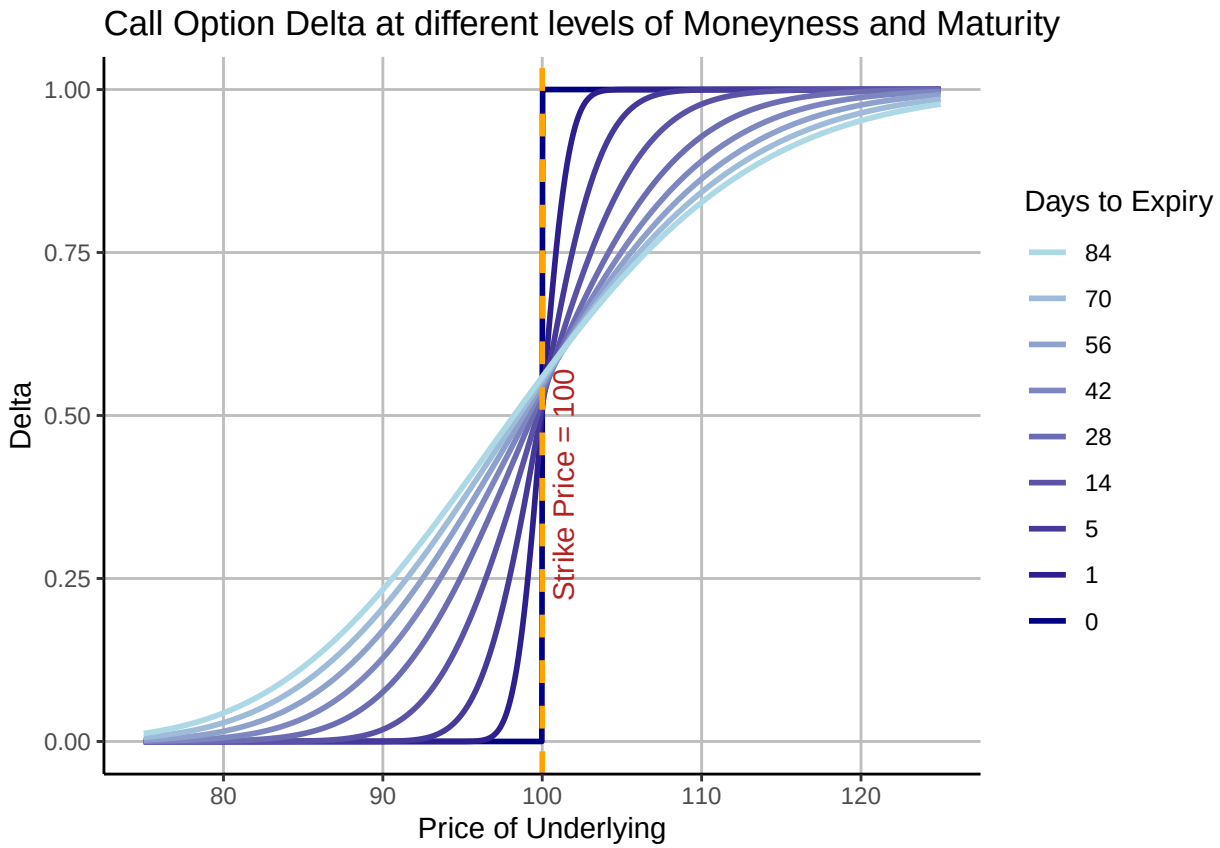
And,

$\Delta_P = \frac{\partial P(S,\tau)}{\partial S} = -e^{-d\tau}\Phi(-d_1) = e^{-d\tau}(\Phi(d_1) - 1)$

We can confirm what delta looks like using our finite difference function.

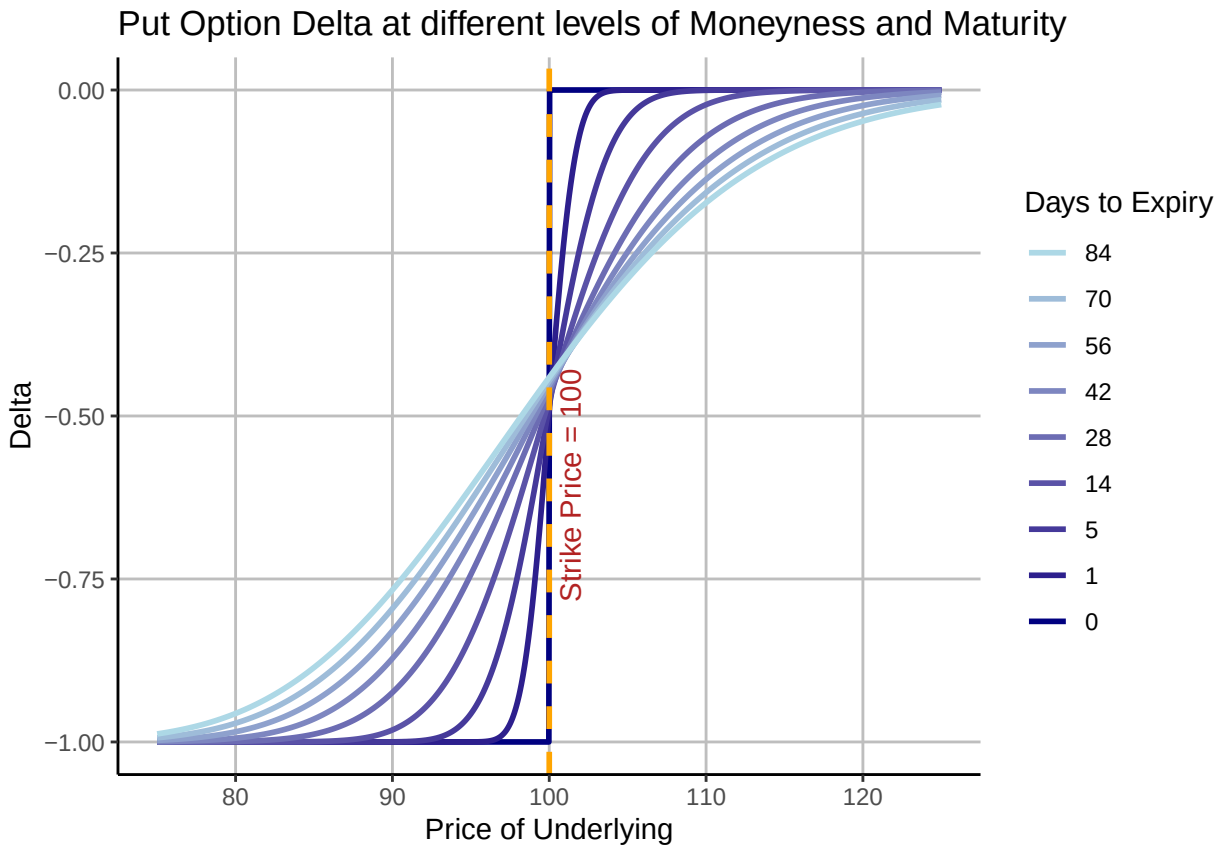
```
call_deltas <- array(NA, dim = c(length(taus), length(S)))
for(i in seq_along(taus)) {
  call_deltas[i,] <- opt_get_delta(S, K, taus[i], vol)
}

put_deltas <- array(NA, dim = c(length(taus), length(S)))
for(i in seq_along(taus)) {
  put_deltas[i,] <- opt_get_delta(S, K, taus[i], vol, is_call = FALSE)
}

call_deltas_tdf <- make_it_tidy(call_deltas, S, taus)
put_deltas_tdf <- make_it_tidy(put_deltas, S, taus)
```

```
make_plot(call_deltas_tdf, "Call Option", "Delta")
```



Delta is always positive for a call. Delta tends to 1 the more in the money the option is.

```
make_plot(put_deltas_tdf, "Put Option", "Delta")
```



Delta is always negative for a put. Delta tends to -1 the more in the money the option is.

3. Option Gamma

Gamma measures the sensitivity of the option's delta to the price of the underlying stock.

$\Gamma_C = \frac{\partial \Delta_C}{\partial S} = \frac{\partial^2 C(S,\tau)}{\partial S^2}$

And,

$\Gamma_P = \frac{\partial \Delta_P}{\partial S} = \frac{\partial^2 P(S,\tau)}{\partial S^2}$

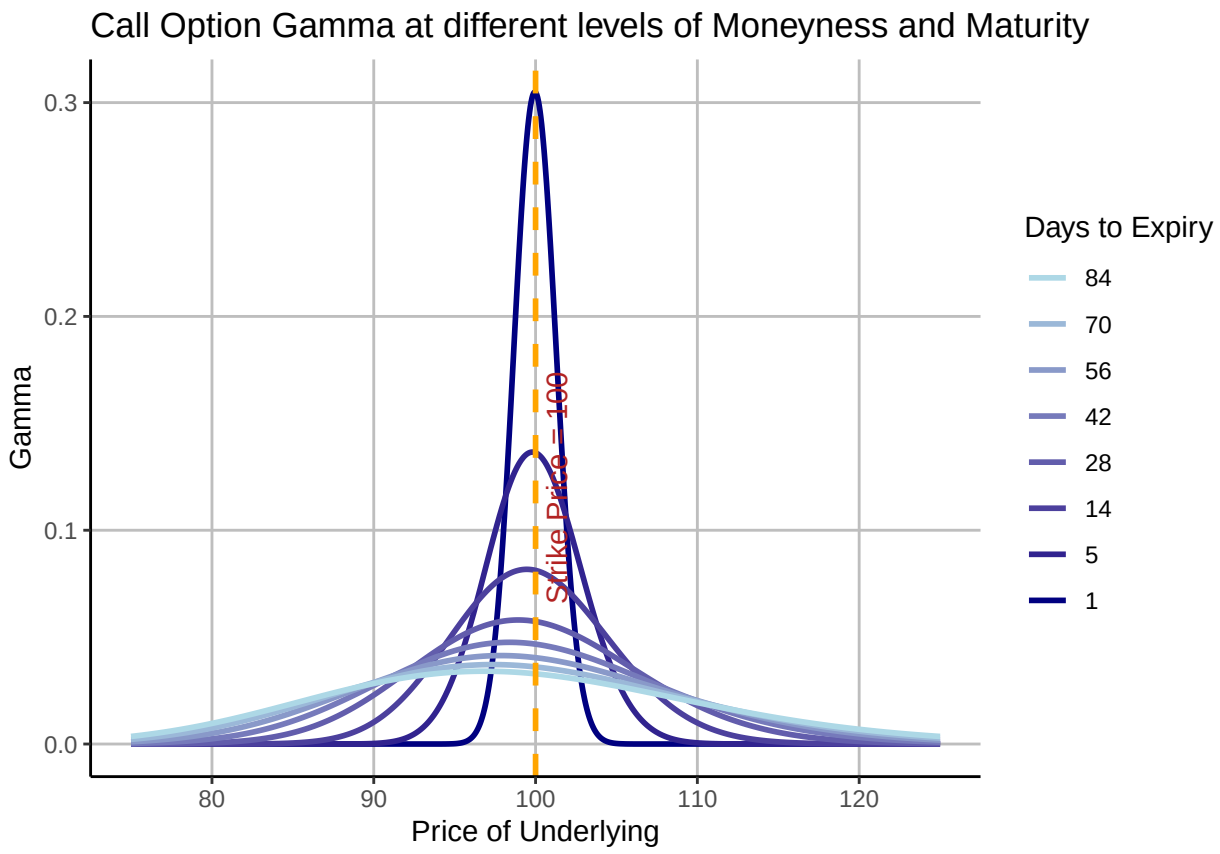
Since Δ_C and Δ_P resemble the shape of the normal distribution function Φ, Γ_C and Γ_C should resemble the shape of the normal density function ϕ. Additionally, Γ_C and Γ_P should look identical. At $\tau \to 0$, Γ should resemble the impulse function.

This is confirmed by our finite difference second derivative function.

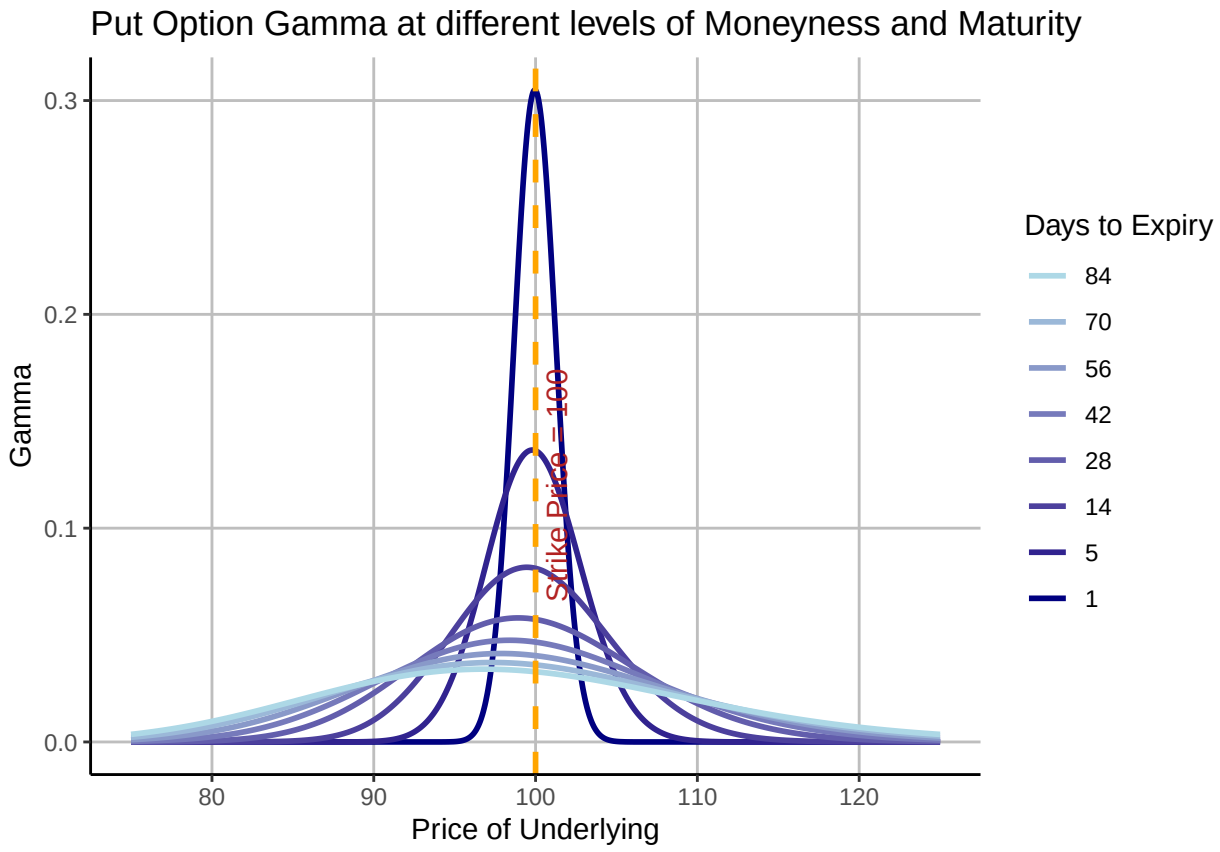
```
taus <- taus[1:length(taus)-1] # remove '0' so that chart shape for other values of tau are clearly vis

call_gammas <- array(NA, dim = c(length(taus), length(S)))
for(i in seq_along(taus)) {
  call_gammas[i,] <- opt_get_gamma(S, K, taus[i], vol)
}

put_gammas <- array(NA, dim = c(length(taus), length(S)))
for(i in seq_along(taus)) {
  put_gammas[i,] <- opt_get_gamma(S, K, taus[i], vol, is_call = FALSE)
}
```

```
call_gammas_tdf <- make_it_tidy(call_gammas, S, taus)
put_gammas_tdf <- make_it_tidy(put_gammas, S, taus)
```

```
make_plot(call_gammas_tdf, "Call Option", "Gamma")
```




```
make_plot(put_gammas_tdf, "Put Option", "Gamma")
```



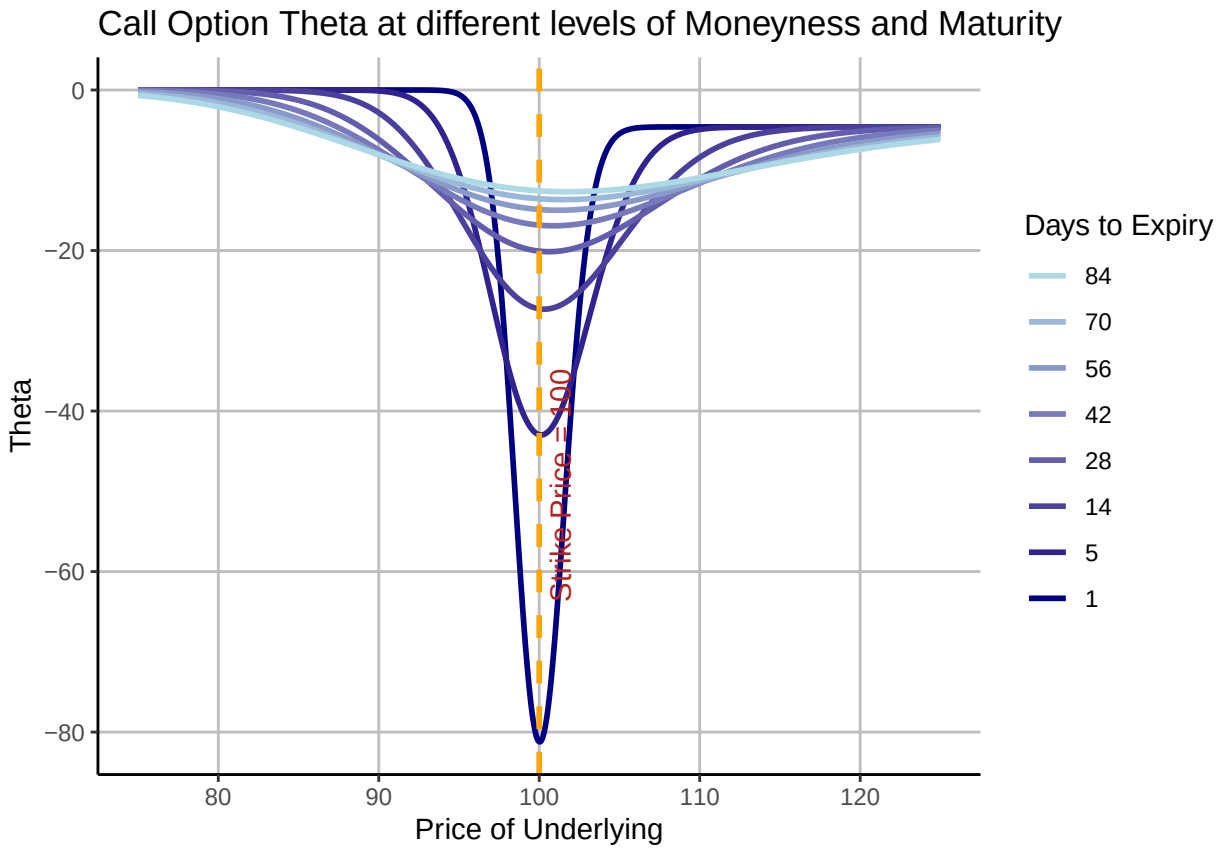
4. Option Theta

Theta measures the options 'time decay'. As the call option approaches expiry, its value 'crystallizes' into $S - K$ ($K - S$ for a put option) or 0.

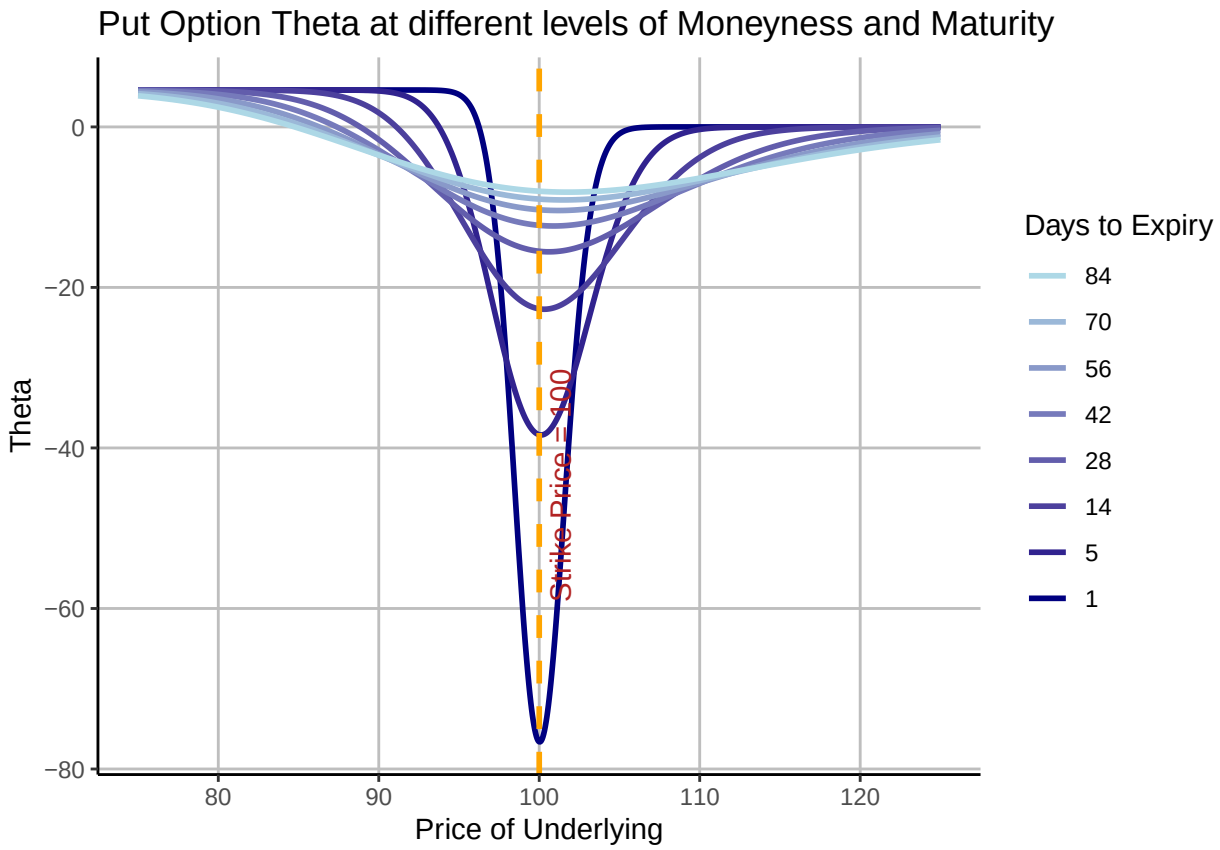
Theta is usually negative if we are long the option, but it can be positive if we are long a far in-the-money put option in a positive interest rate environment.

```
call_thetas <- array(NA, dim = c(length(taus), length(S)))
for(i in seq_along(taus)) {
  call_thetas[i,] <- opt_get_theta(S, K, taus[i], vol)
}

put_thetas <- array(NA, dim = c(length(taus), length(S)))
for(i in seq_along(taus)) {
  put_thetas[i,] <- opt_get_theta(S, K, taus[i], vol, is_call = FALSE)
}

call_thetas_tdf <- make_it_tidy(call_thetas, S, taus)
put_thetas_tdf <- make_it_tidy(put_thetas, S, taus)
```

```
make_plot(call_thetas_tdf, "Call Option", "Theta")
```




```
make_plot(put_thetas_tdf, "Put Option", "Theta")
```



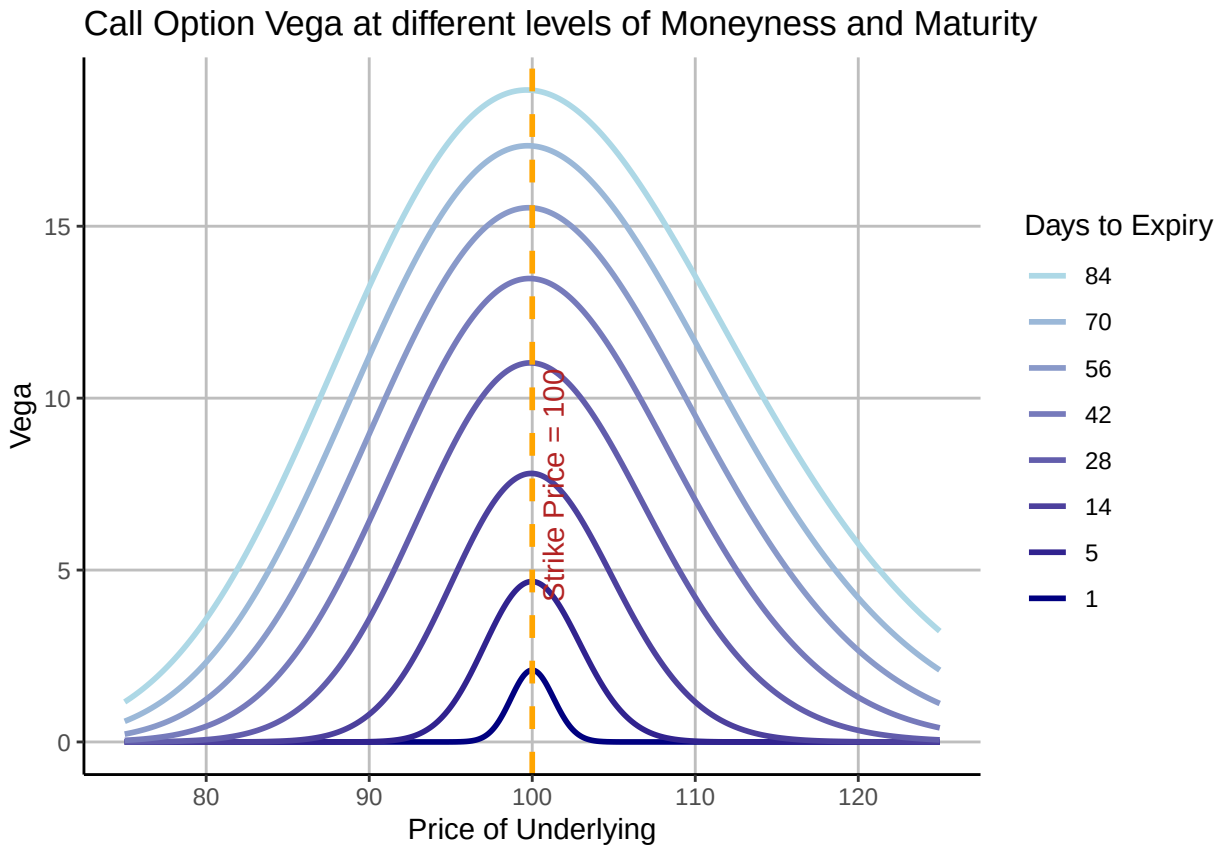
4. Option Vega

Vega measures the sensitivity of the option's price to the implied volatility of the underlying asset. The stock's volatility is not constant, but changes as market perceptions change.

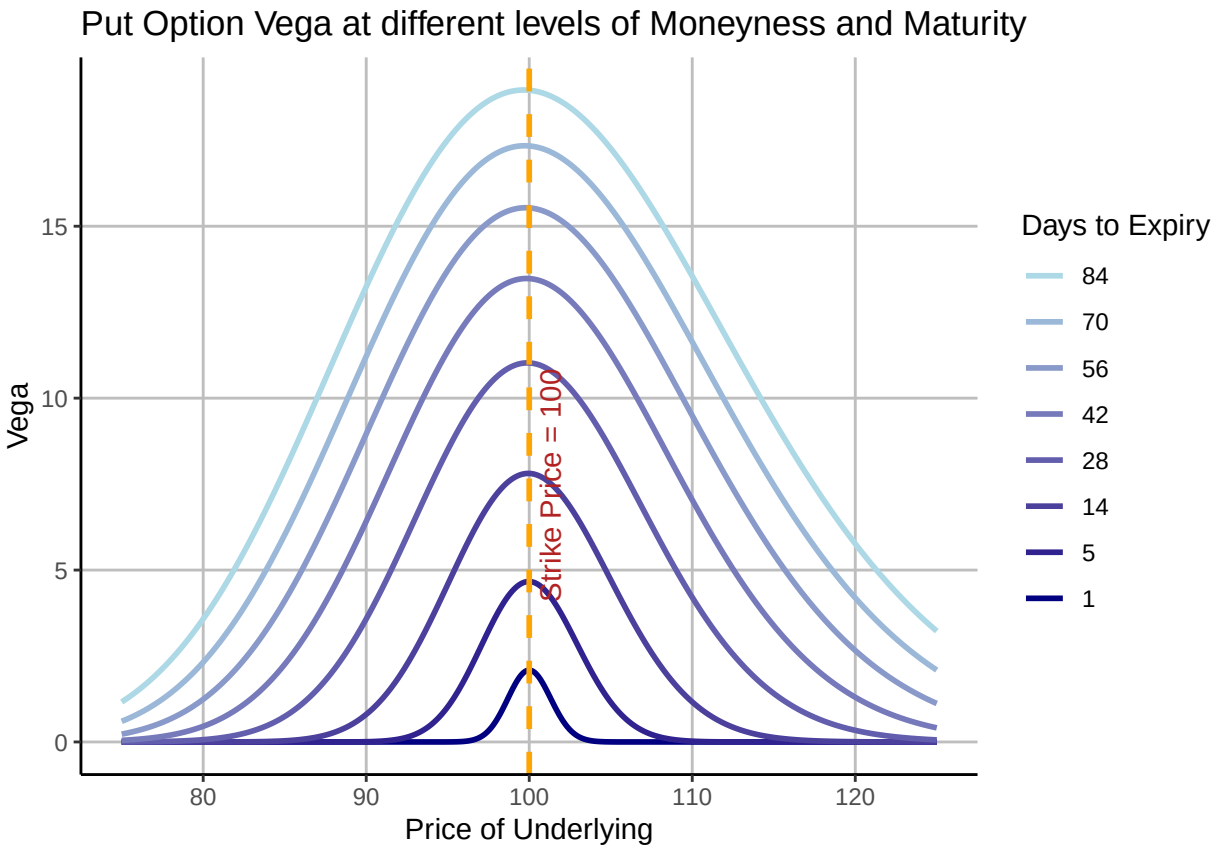
Volatility matters most when the option is 'near-the-money', as shown by how vega varies with S. Volatility also matters most when we are far from expiry.

```
call_vegas <- array(NA, dim = c(length(taus), length(S)))
for(i in seq_along(taus)) {
  call_vegas[i,] <- opt_get_vega(S, K, taus[i], vol)
}

put_vegas <- array(NA, dim = c(length(taus), length(S)))
for(i in seq_along(taus)) {
  put_vegas[i,] <- opt_get_vega(S, K, taus[i], vol, is_call = FALSE)
}

call_vegas_tdf <- make_it_tidy(call_vegas, S, taus)
put_vegas_tdf <- make_it_tidy(put_vegas, S, taus)
```

```
make_plot(call_vegas_tdf, "Call Option", "Vega")
```




```
make_plot(put_vegas_tdf, "Put Option", "Vega")
```


Put Option Vega at different levels of Moneyness and Maturity
Vega
15
10
5
0
80
90
100
110
120
Price of Underlying
Strike Price = 100
Days to Expiry
84
70
56
42
28
14
5
1