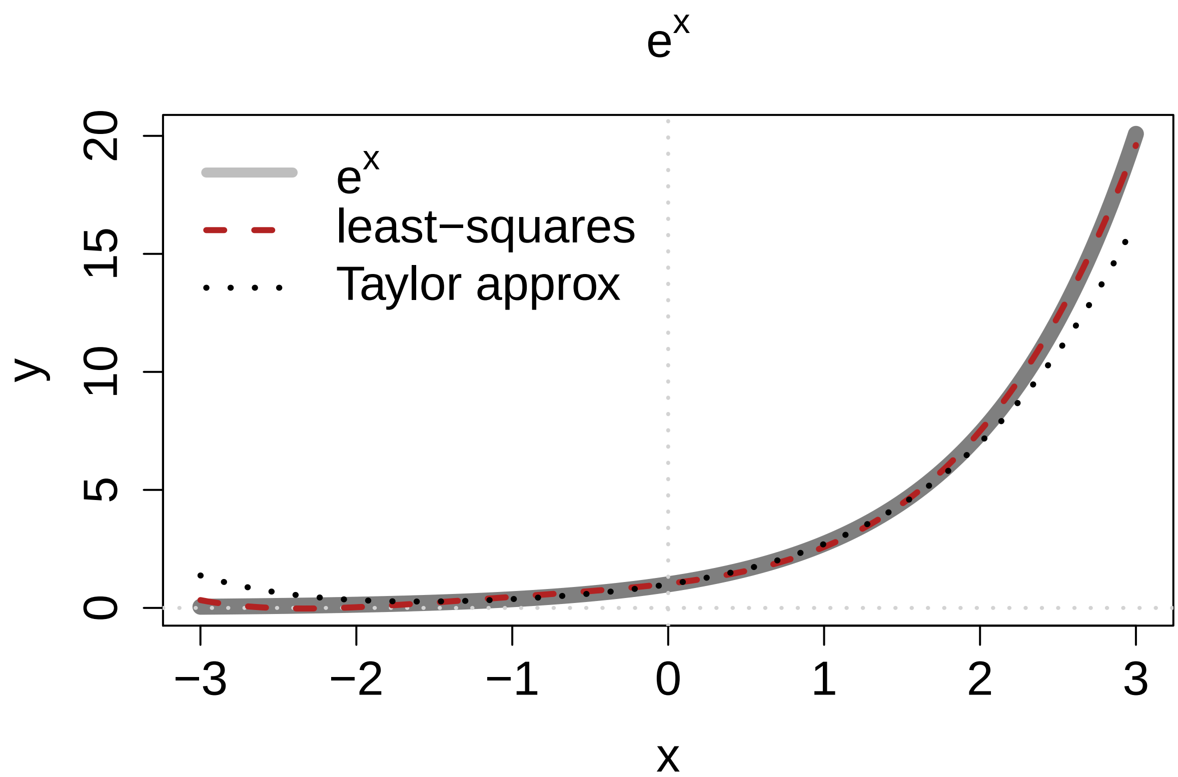
```r
x <- seq(-3, 3, length.out = 1000)
y <- exp(x)
# We fit our "model" as before
#
fit <- lm(y~x + I(x^2) + I(x^3) + I(x^4))
#
# We can now compare coefficients to the Taylor series values
# First the fit
round(fit$coefficients,3)
```

```
## (Intercept)           x      I(x^2)      I(x^3)      I(x^4)
##       1.029       0.788       0.435       0.269       0.062
```

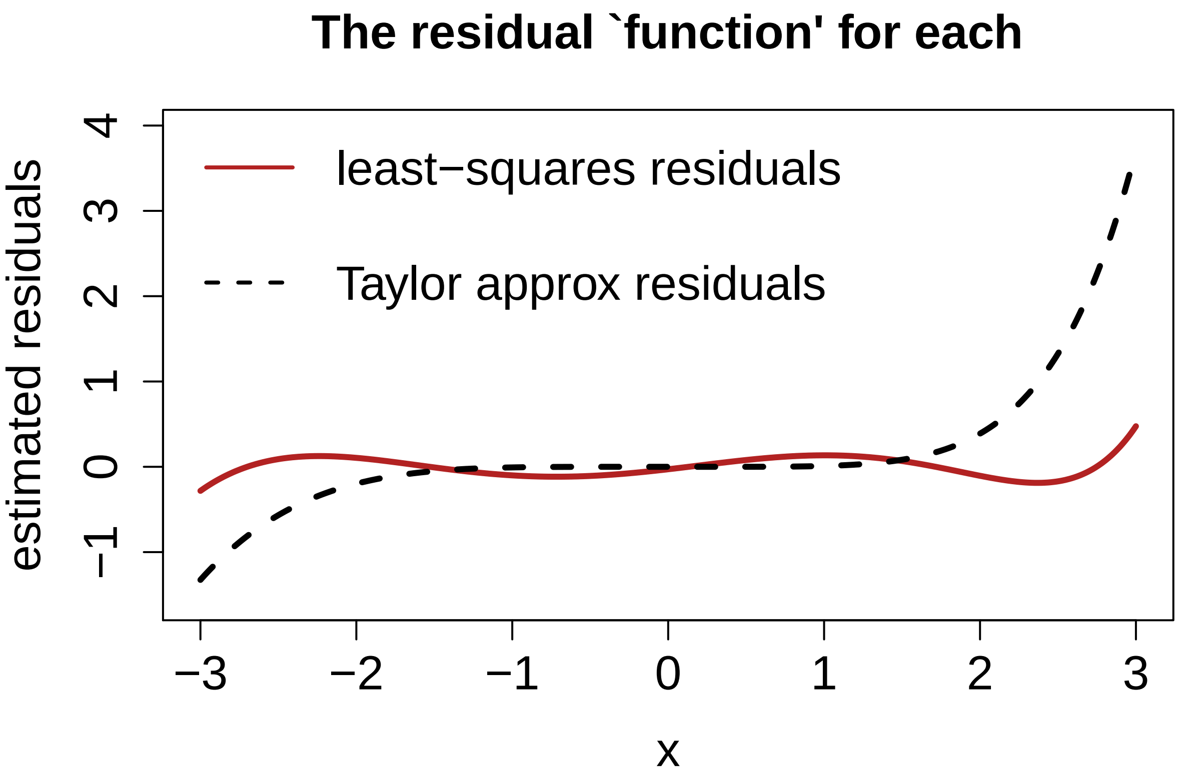
```r
# and now the Taylor coefficients
round(sapply(0:4, function(x) 1/factorial(x)), 3)
```

```
## [1] 1.000 1.000 0.500 0.167 0.042
```

```r
# We can compare them visually
plot(x, y, col=adjustcolor("black", 0.5), type="l",
     main = expression(e^x), xlab="x", ylab="y" , lwd=8,
     cex.lab=1.5 , cex.axis=1.5 , cex.main=1.5)
abline(h=0, col="lightgrey", lty=3 , lwd=2)
abline(v=0, col="lightgrey", lty=3, lwd=2)

# Note that the x's are in order 1:N so order(x) was unnecessary here
lines(x, fit$fitted.values, col="firebrick", lty=2, lwd=3)

# The Taylor approx
ytaylor = 1 + x + x^2/2  + x^3/6 + x^4 /24
lines(x, ytaylor, col="black", lty=3, lwd=3)

# Add a legend
legend("topleft",  bty='n', cex=1.5,
       legend = c(expression(e^x), "least-squares", "Taylor approx"),
       col = c("grey", "firebrick", "black"),
       lty=c(1, 2,3 ), lwd=c(5, 3, 3),
       text.width = 1.5
)
```




```r
# We can look at the residuals
taylor_resids <- y - ytaylor
residlim <- extendrange(c(taylor_resids, fit$residuals))

# The residuals from ls
plot(x, fit$residuals, type="l" , lwd=3, col="firebrick",  ylim=residlim,
     main = "The residual `function' for each",
     xlab="x", ylab="estimated residuals" ,
     cex.axis=1.5 , cex.lab=1.5, cex.main=1.5)
# Taylor residuals
lines(x, taylor_resids, lty=2 , lwd=3, col="black")

# Add a legend
legend("topleft", bty="n", cex=1.5,
       legend = c("least-squares residuals", "", "Taylor approx residuals"),
       col = c("firebrick","","black"),
       lty = c(1,0,2), lwd=2,
       text.width = 2
)
```




```r
# Let's look at the coefficients
x <- seq(-3, 3, length.out = 1000)
y <- exp(x)
# We fit our "model" as before

fit1 <- lm(y~x + I(x^2) + I(x^3) + I(x^4))
fit2 <- lm(y~poly(x,4))

# The Taylor approximation
Taylor = sapply(0:4, function(x) 1/factorial(x))

# We can now compare coefficients to the Taylor series values
# First the fit
round(fit1$coefficients,3)
```

```
## (Intercept)           x      I(x^2)      I(x^3)      I(x^4)
##       1.029       0.788       0.435       0.269       0.062
```

```r
round(fit2$coefficients,3)
```

```
## (Intercept) poly(x, 4)1 poly(x, 4)2 poly(x, 4)3 poly(x, 4)4
##       3.346     123.146      77.778      34.882      12.153
```

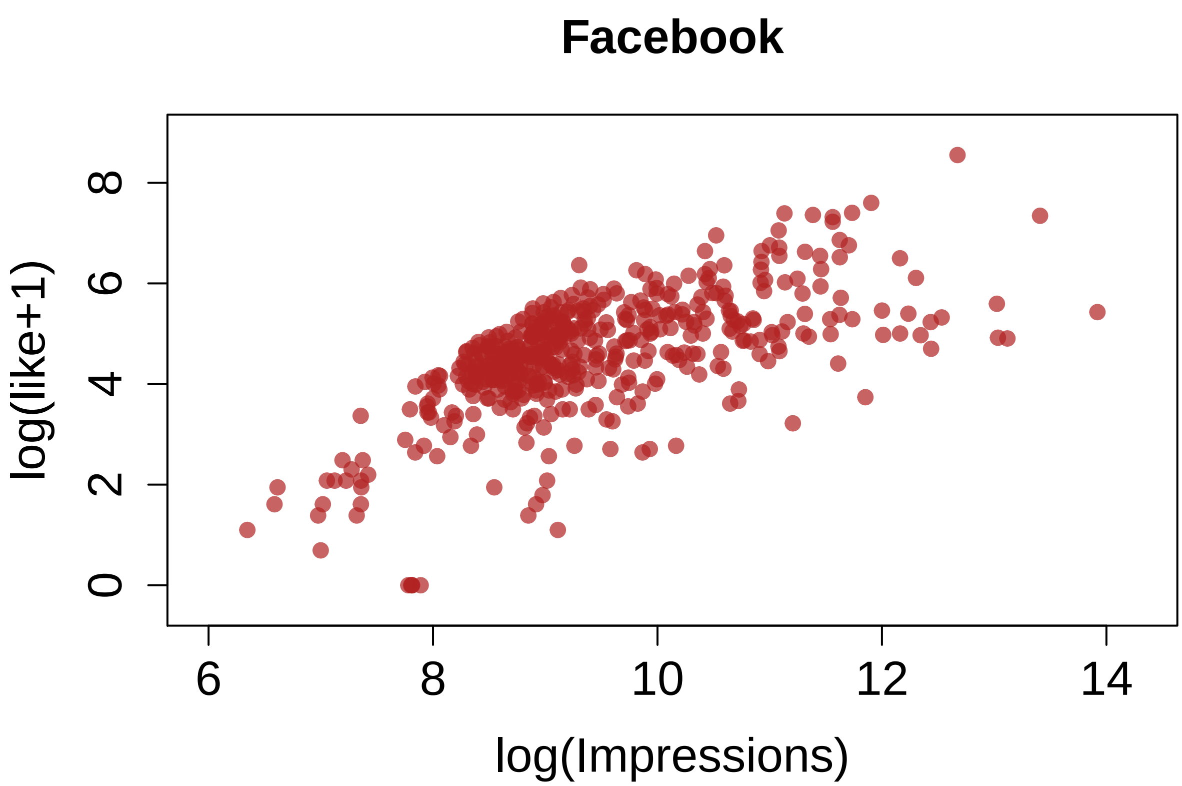
```r
round(Taylor, 3)
```

```
## [1] 1.000 1.000 0.500 0.167 0.042
```

Facebook example geometry

```
# Load the data into R and plot the data
facebook <- read.csv("stat-444-datasets/facebook.csv", header=TRUE)
fb <- na.omit(facebook)
# Log-transform data
fb$x = log(fb$Impressions)
fb$y = log(fb$like+1)
```

```
# Plotting the log-transformed data
xlim <- extendrange(fb$x)
ylim <- extendrange(fb$y)
plot(fb$x, fb$y,
     xlim = xlim, ylim = ylim,
     main = "Facebook",
     xlab = "log(Impressions)",
     ylab = "log(like+1)",
     pch=19,
     col=adjustcolor("firebrick", 0.7),
     cex.main=1.5 , cex.axis=1.5 , cex.lab=1.5
)
```

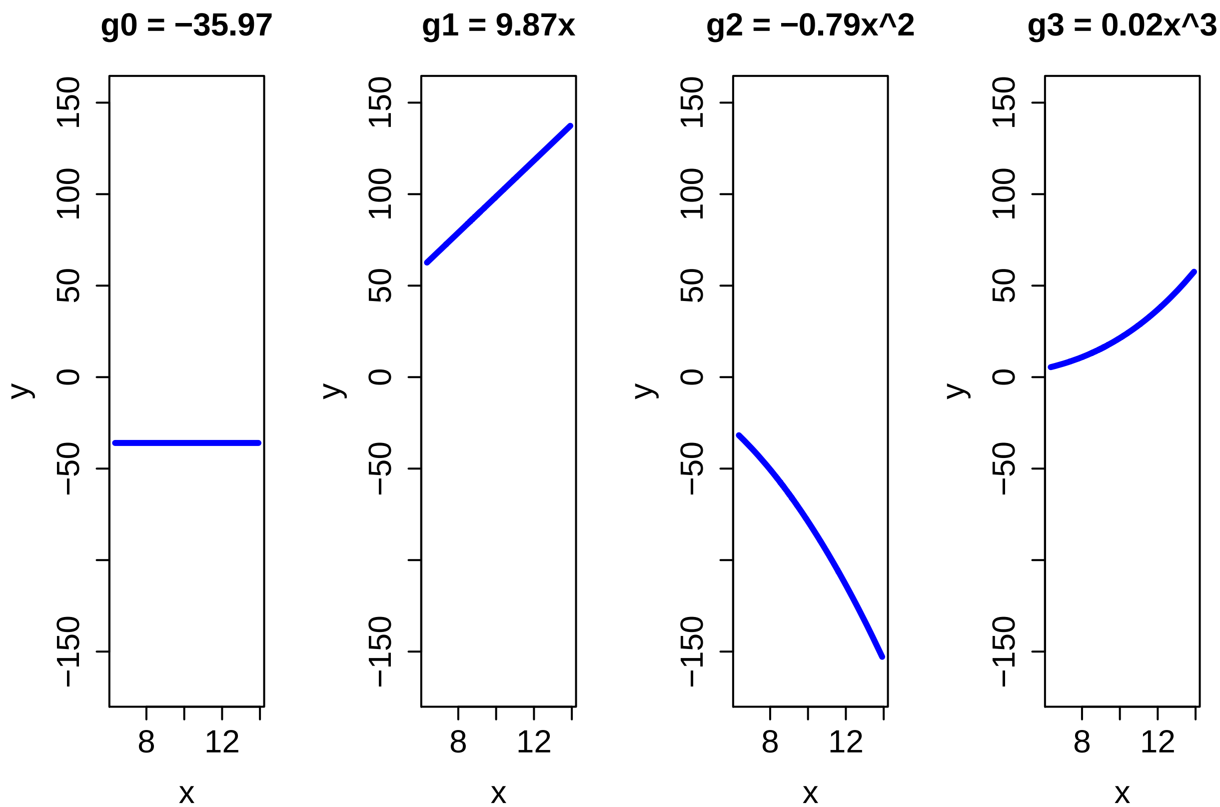



```
facebook.fit3 <- lm(y ~ x + I(x^2) + I(x^3), data=fb)
coefs = coef(facebook.fit3)
print(round(coefs, 3))
```

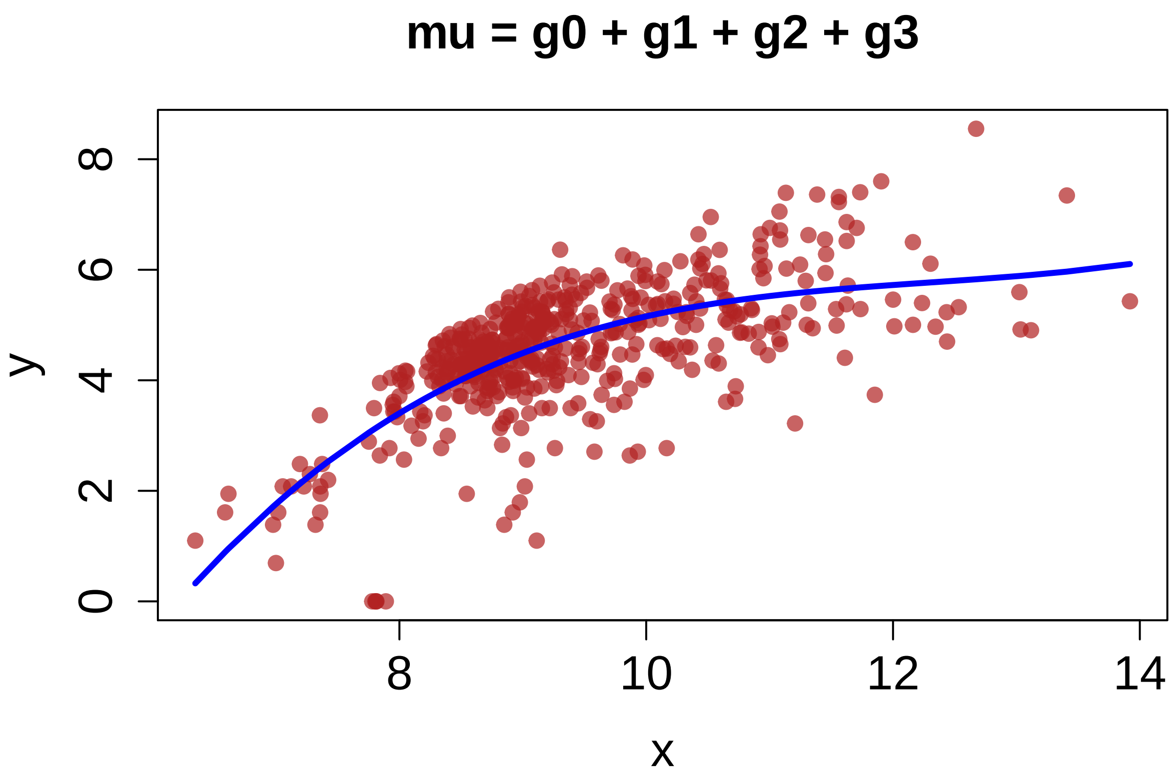
```
## (Intercept)           x      I(x^2)      I(x^3)
##     -35.974       9.866      -0.789       0.021
```

```
coefs = unname(coefs)
Xorder <- order(fb$x)
n <- length(Xorder)
g0 <- coefs[1] * rep(1, n)   # constant term  x^0
g1 <- coefs[2] * fb$x        # linear term    x
g2 <- coefs[3] * fb$x^2      # squared term   x^2
g3 <- coefs[4] * fb$x^3      # cubed term     x^3
```

```
par(mfrow=c(1,4))
glim <- extendrange(c(g0, g1, g2, g3))
plot(fb$x[Xorder], g0,
     ylim=glim, xlab="x", ylab="y",
     col="blue", lwd=3, type="l",
     main=paste0("g0 = ", round(coefs[1],2)),
     cex.main=1.5 , cex.axis=1.5 , cex.lab=1.5)
plot(fb$x[Xorder], g1[Xorder],
     ylim=glim, xlab="x", ylab="y",
     col="blue", lwd=3, type="l",
     main=paste0("g1 = ", round(coefs[2],2), "x"),
     cex.main=1.5 , cex.axis=1.5 , cex.lab=1.5)
plot(fb$x[Xorder], g2[Xorder],
     ylim=glim, xlab="x", ylab="y",
     col="blue", lwd=3, type="l",
     main=paste0("g2 = ", round(coefs[3],2), "x^2"),
     cex.main=1.5 , cex.axis=1.5 , cex.lab=1.5)
plot(fb$x[Xorder], g3[Xorder],
     ylim=glim, xlab="x", ylab="y",
     col="blue", lwd=3, type="l",
     main=paste0("g3 = ", round(coefs[4],2), "x^3"),
     cex.main=1.5 , cex.axis=1.5 , cex.lab=1.5)
```




```
par(mfrow=c(1,1))
plot(fb$x, fb$y,
     xlab = "x", ylab = "y",
     main = "mu = g0 + g1 + g2 + g3",
     pch=19, col=adjustcolor("firebrick", 0.7),
     cex.main=1.5 , cex.axis=1.5 , cex.lab=1.5)
lines(fb$x[Xorder], g0[Xorder] + g1[Xorder] + g2[Xorder] + g3[Xorder],
      col="blue", lwd=3)
```


mu = g0 + g1 + g2 + g3
y
0
2
4
6
8
8
10
12
14
x