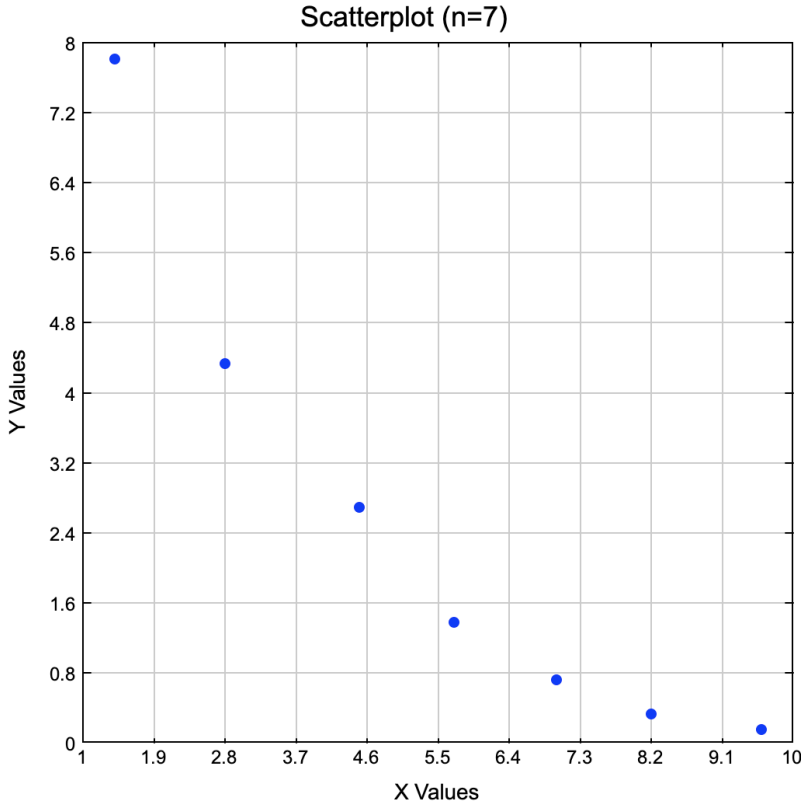
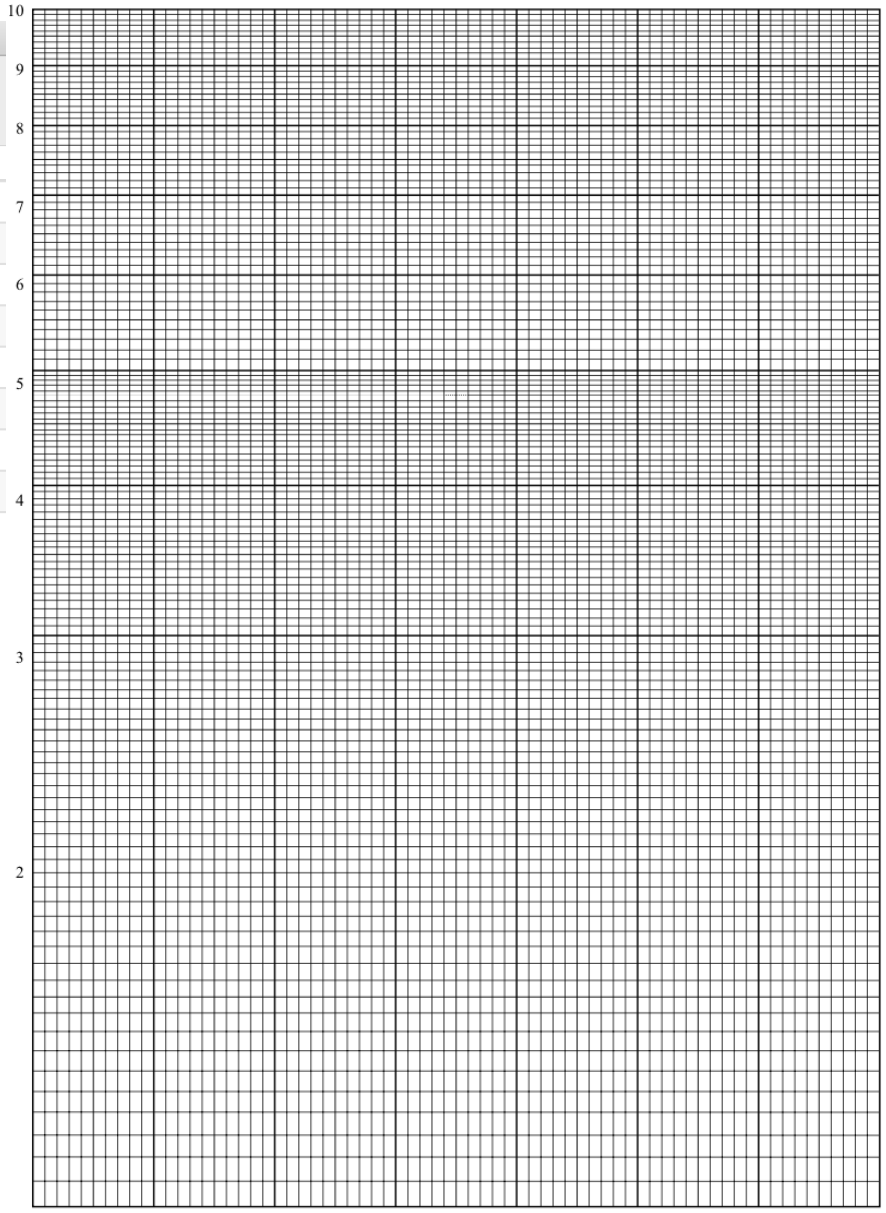
Clear Copy Paste

Row	1	2
1	1.4	7.81
2	2.8	4.33
3	4.5	2.69
4	5.7	1.38
5	7.0	0.72
6	8.2	0.33
7	9.6	0.15
8		
9		



Clear Copy Paste

Row	1	2
1	1.4	7.81
2	2.8	4.33
3	4.5	2.69
4	5.7	1.38
5	7.0	0.72
6	8.2	0.33
7	9.6	0.15
8		
9		



Sample Transformer - 2

Basic Transformation

Source column: 2

Operation: +

Constant

Column 3

Basic Transform

Advanced Transformation

Equation (see help for details):

log10(Col2)

*Example: Col1*Col2/cos(Col4)*

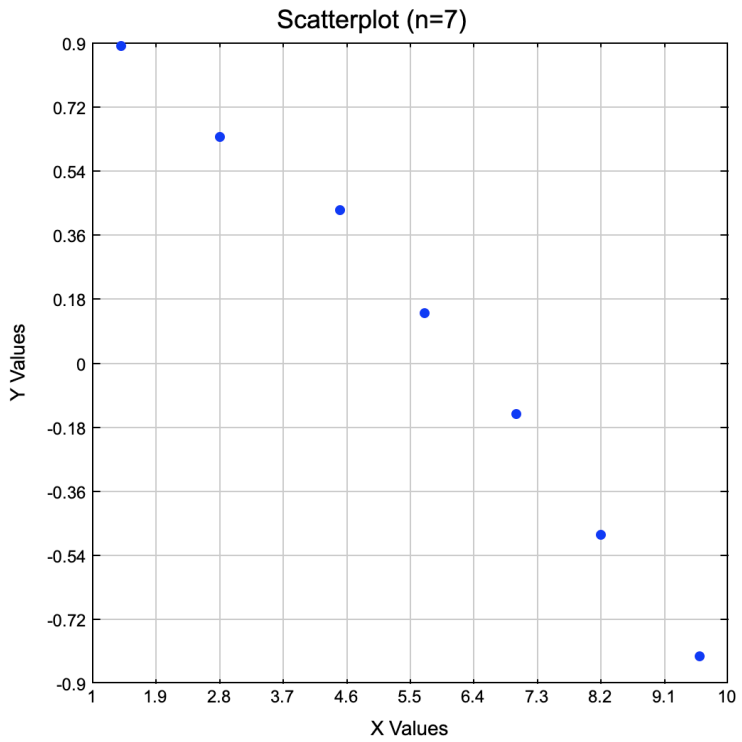
Advanced Transform

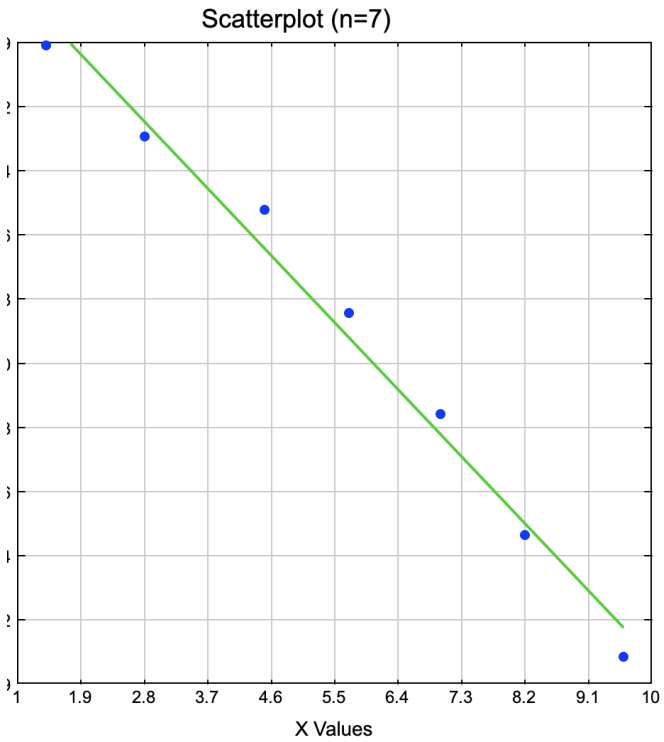
Row	Transformed
1	0.892651
2	0.6364879
3	0.4297523
4	0.1398791
5	-0.1426675
6	-0.4814861
7	-0.8239087

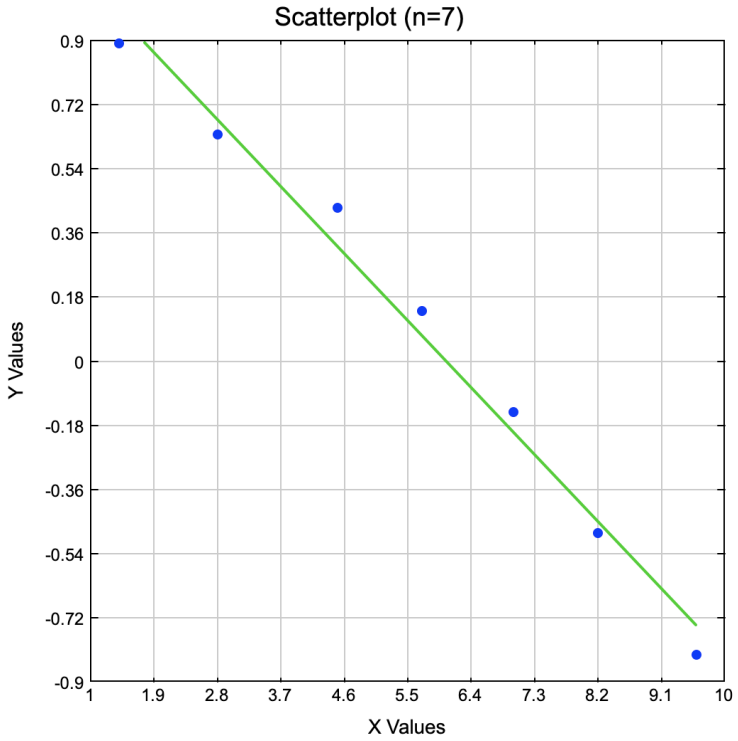
Statdisk Samp

Clear Copy Paste Save Print Data Tools

Row	1	2	3	4	5
1	1.4	7.81	0.892651		
2	2.8	4.33	0.6364879		
3	4.5	2.69	0.4297523		
4	5.7	1.38	0.1398791		
5	7.0	0.72	-0.1426675		
6	8.2	0.33	-0.4814861		
7	9.6	0.15	-0.8239087		







Convert

Correlation and Regression - 1

Significance: 0.05

Select the columns to be used for the x and v variables.

x variable column 1

y variable column 3

Evaluate

Scatterplot

Residual Plot

```
Sample size, n:      7
Degrees of freedom: 5

Correlation Results:
Correlation coeff, r: -0.9923699
Critical r:           ±0.7544919
P-value (two-tailed): 0.00001

Regression Results:
Y= b0 + b1x:
Y Intercept, b0:      1.262323
Slope, b1:            -0.2088151

Total Variation:       2.276716
Explained Variation:   2.242105
Unexplained Variation: 0.0346107
Standard Error:        0.0831994
Coeff of Det, R^2:     0.984798
```

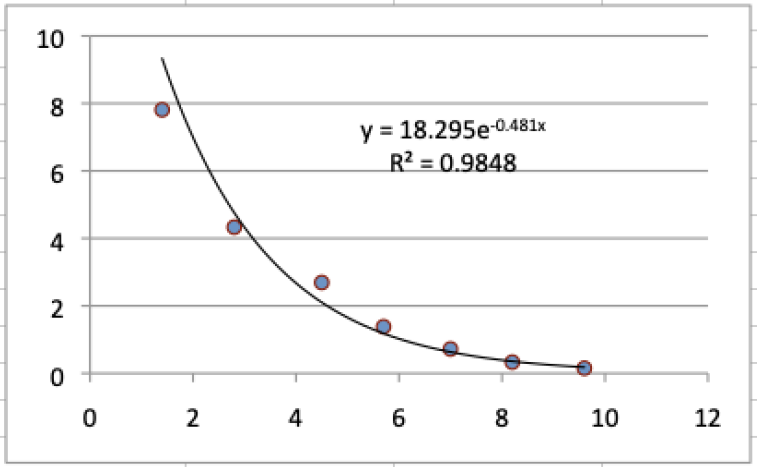
Enter the slope -0.208

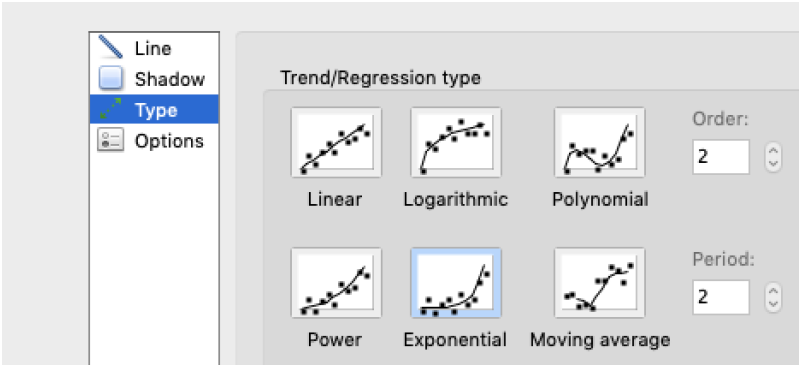
Enter the intercept 1.262

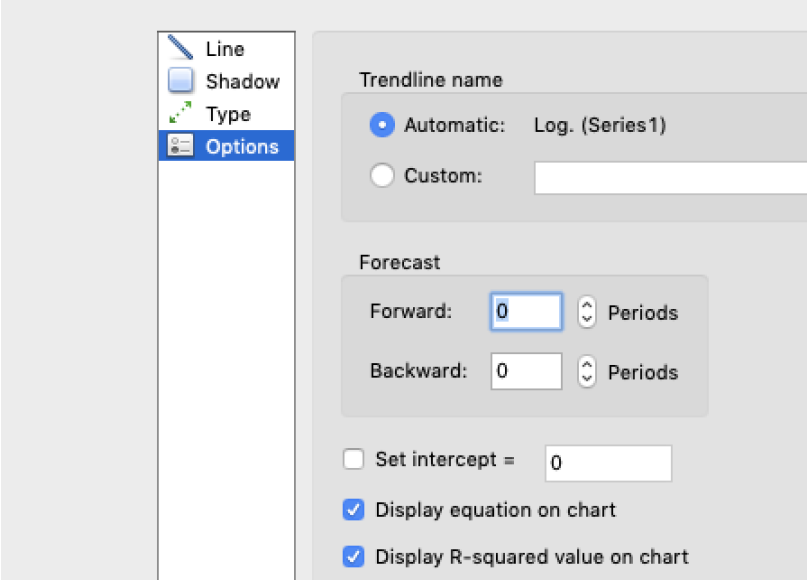
18.281002 (0.619441) ^ x

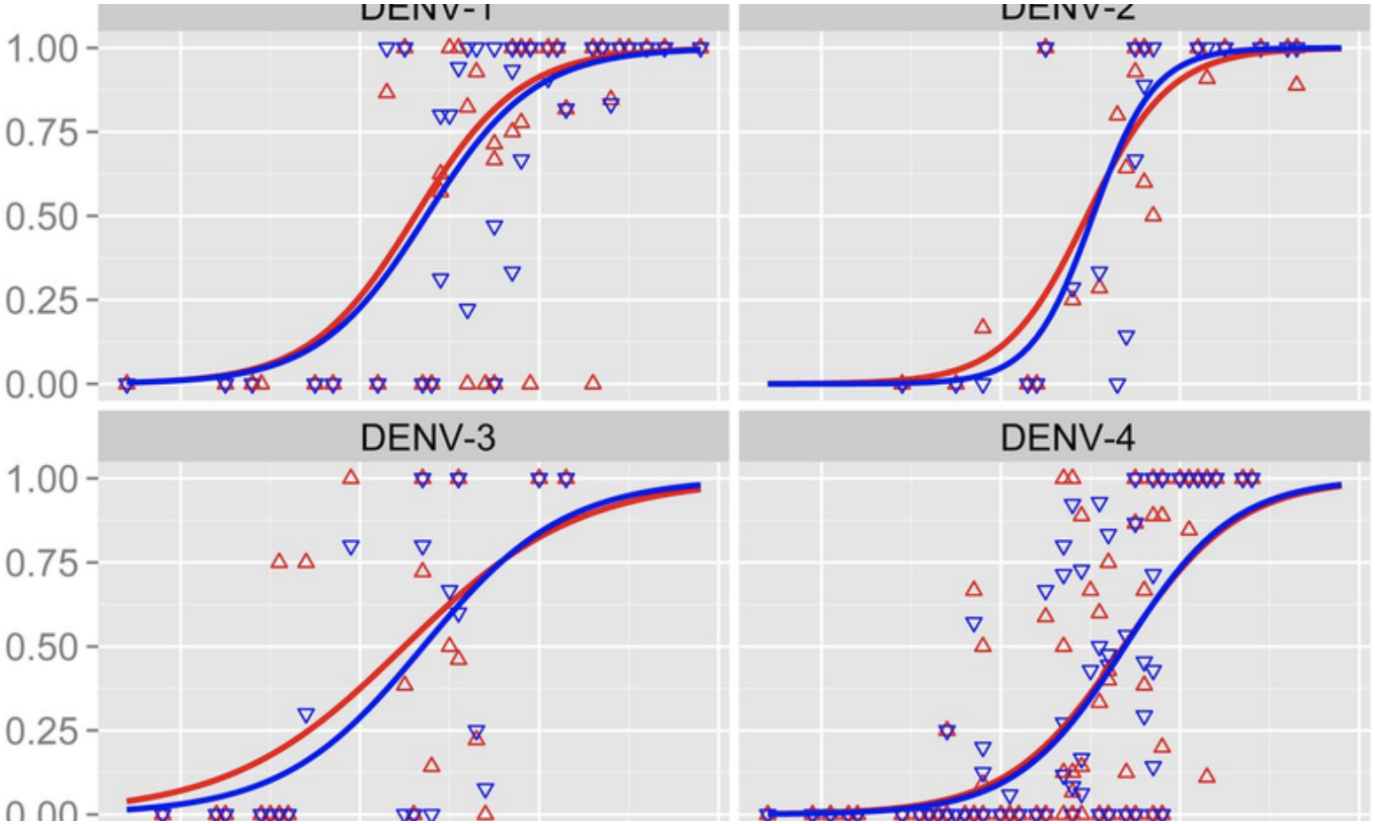
18.281002 e ^ -0.478938 x

	A	B
1	1.4	7.81
2	2.8	4.33
3	4.5	2.69
4	5.7	1.38
5	7	0.72
6	8.2	0.33
7	9.6	0.15
8		
9		
10		
11		
12		
13		

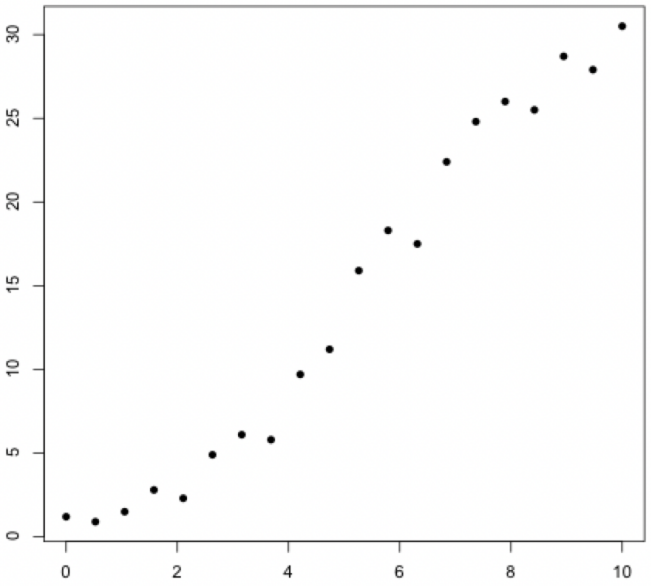


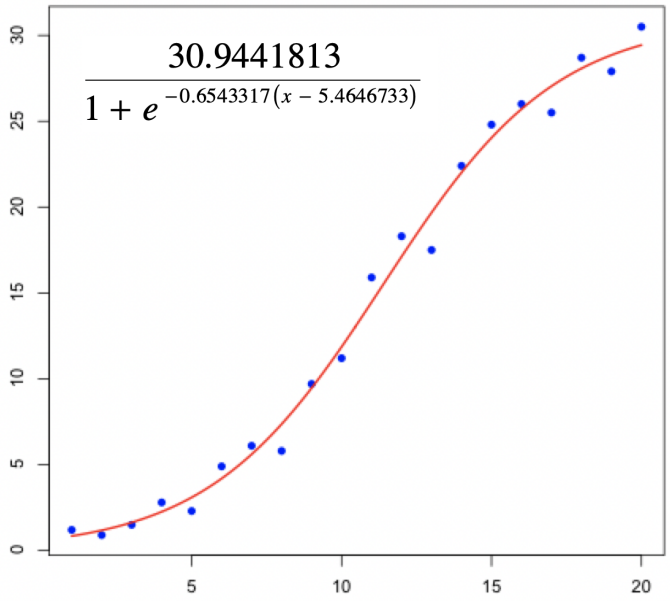





DENV-1
DENV-2
DENV-3
DENV-4
1.00
0.75
0.50
0.25
0.00


mth_150





The R Project for Statistical Computing

```
# Data
y <- c(1.2, 0.9, 1.5, 2.8, 2.3, 4.9, 6.1, 5.8, 9.7, 11.2,
       15.9, 18.3, 17.5, 22.4, 24.8, 26.0, 25.5, 28.7, 27.9, 30.5)

x <- 1:length(y)

# Plot raw data
plot(x, y, pch = 16, col = "blue",
     main = "Logistic Fit to Data",
     xlab = "x", ylab = "y")

# Logistic model function
logistic <- function(x, L, k, x0) {
  L / (1 + exp(-k * (x - x0)))
}

# Fit using nonlinear least squares
fit <- nls(y ~ L / (1 + exp(-k * (x - x0))),
           start = list(L = max(y), k = 0.3, x0 = mean(x)))

# Generate smooth curve
x_fit <- seq(min(x), max(x), length.out = 200)
y_fit <- predict(fit, newdata = list(x = x_fit))

# Add fitted sigmoid curve
lines(x_fit, y_fit, col = "red", lwd = 2)

# Optional: show fitted parameters
summary(fit)

coef(log_model)

      L           k          x0
30.9441813  0.6543317  5.4636733
```