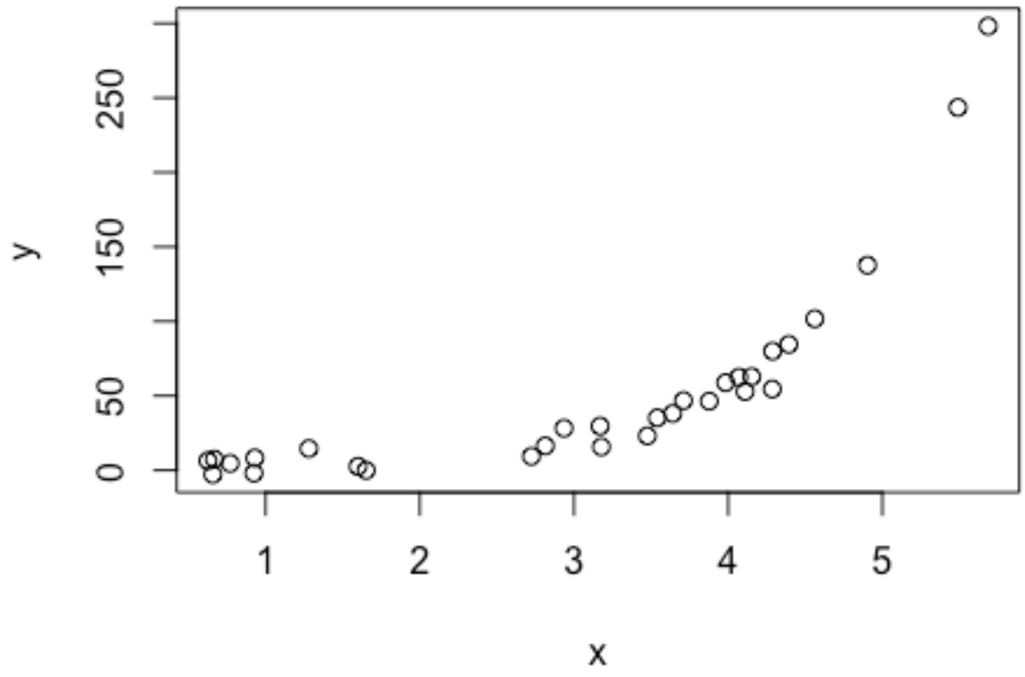
HW 6 script

Ian Douglas 3/30/2019

Question 1

Part A

```
n <- 30
set.seed(63)
x <- runif(n, 0.5, 6)
y <- exp(x) + rnorm(n, sd = 6)
plot(x,y)
```



This does not look like a linear relationship

Part B

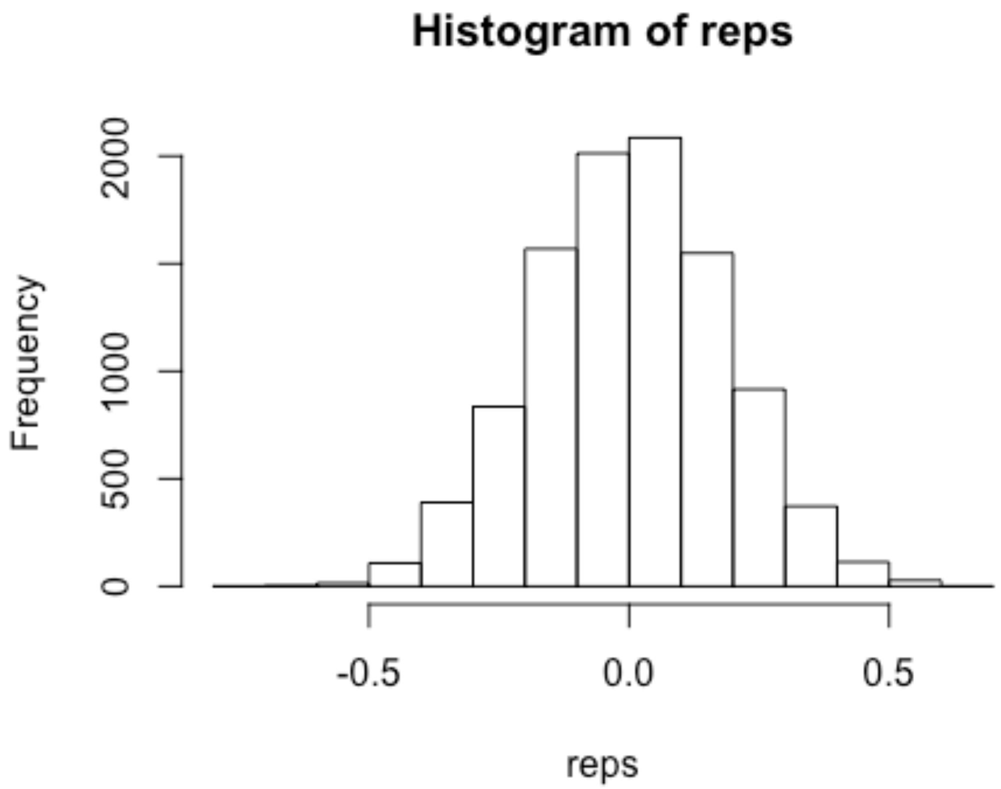
```
Pearson <- cor(x,y)
Spearman <- cor(x,y, method = "spearman")
```

```
out <- cbind(Pearson, Spearman)
sprmn<-out[,2]
out

##         Pearson Spearman
## [1,] 0.7568039 0.963515
```

Part C

```
#recalculate the Spearman correlation 9999 times
z <- c(x,y)
R <- 9999
n <- length(x)
K <- 1:length(z)
reps <- numeric(R)
for (i in 1:R) {
  k <- sample(K, size = n, replace = FALSE)
  x1 <- z[k]
  y1 <- z[-k]
  reps[i] <- cor(x1, y1, method = "spearman")
}
hist(reps)
```



Part D

```
p.val <- mean(c(sprmn, abs(reps) >= sprmn))
p.val

## [1] 9.63515e-05
```

Based on these findings I reject the Null hypothesis that the samples are independent.

This is because the observed p-value from the original sample is highly unlikely to have come from the population of permutation samples, for which independence is assumed. Thus, the original sample breaks this assumption of independence.

Part #E

```
cor.test(x, y, method="spearman")

##
##  Spearman's rank correlation rho
##
## data:  x and y
## S = 164, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.963515
```

the p-value derived from the permutation test is slightly larger, though it would have been even smaller if there had been more replications, because the expression:

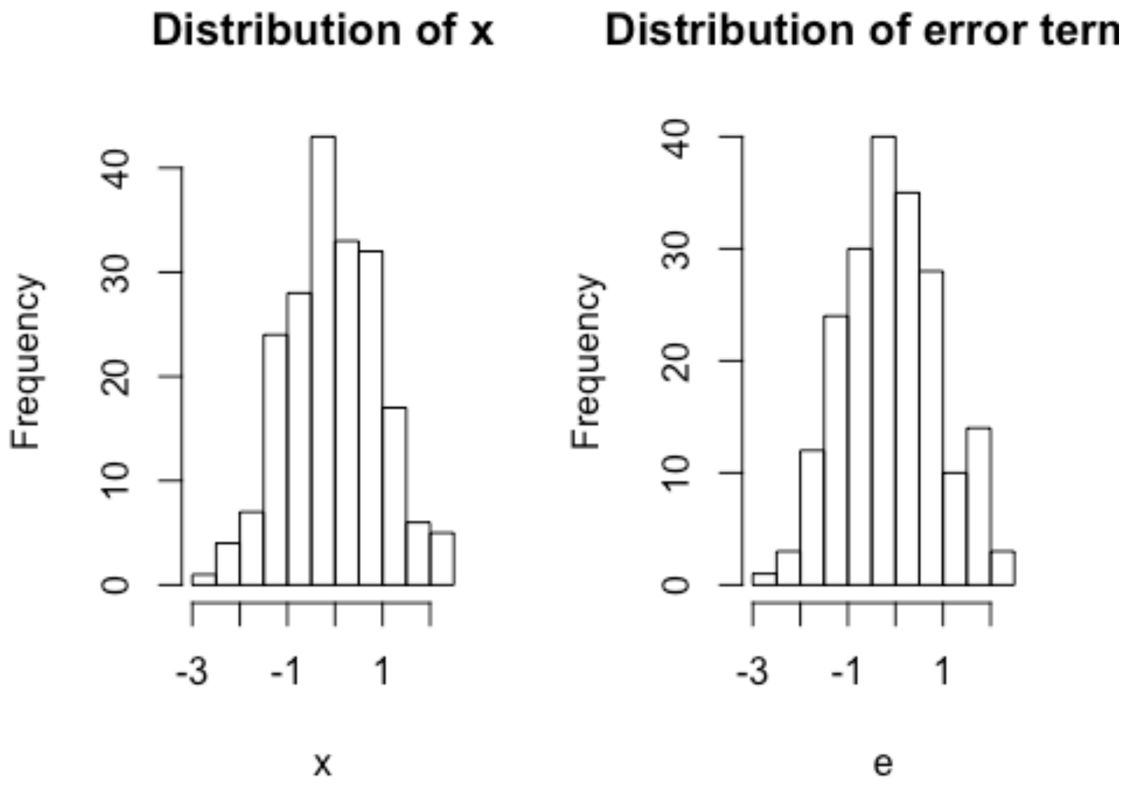
```
abs(reps) >= sprmn #Note: sprmn = .963515 is rho of the original sample
```

was a vector of all `FALSE`; thus the p-value would be larger with more replications.

Question 2

Part A

```
#simulate data
set.seed(437)
x <- rnorm(n=200)
e <- rnorm(n=200)
par(mfrow=c(1,2))
hist(x,main = "Distribution of x")
hist(e,main = "Distribution of error terms")
```



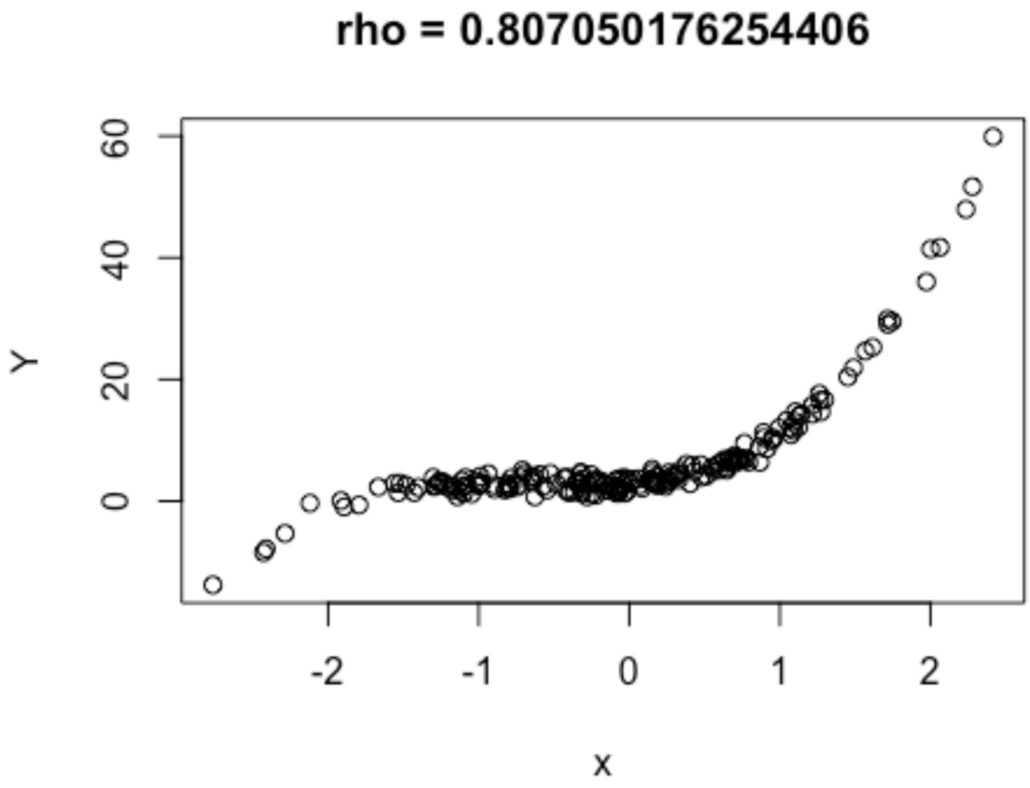
They look approximately normal.

Part B

generate Y using

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + e$$

```
B <- matrix(rpois(n=4,lambda=4),ncol=1)
Y <- B[1,] + B[2,]*x +B[3,]*x^2 + B[4,]*x^3 + e
plot(x,Y, main = paste("rho =",cor(x,Y,method="spearman")))
```



Part C

```
#best polynomial search using bestglm with BIC and AIC
#Create dataframe of predictors: {x^1,x^2,...,x^15}
X <- matrix(rep(0,times = 200*15),ncol = 15)
for (i in 1:15) {
  X[,i] <- x^i
}
#attach Y
df <- as.data.frame(cbind(X,Y))
names(df) <- c(names(df)[1:ncol(df)-1], "y")
library(bestglm)

## Loading required package: leaps

bestBIC <- bestglm(df, IC="BIC")
bestAIC <- bestglm(df, IC = "AIC")
```

Best model according to BIC:

```
bestBIC$BestModel

##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Coefficients:
## (Intercept)           V1           V2           V3
##       2.904        2.035        4.010        2.011
```

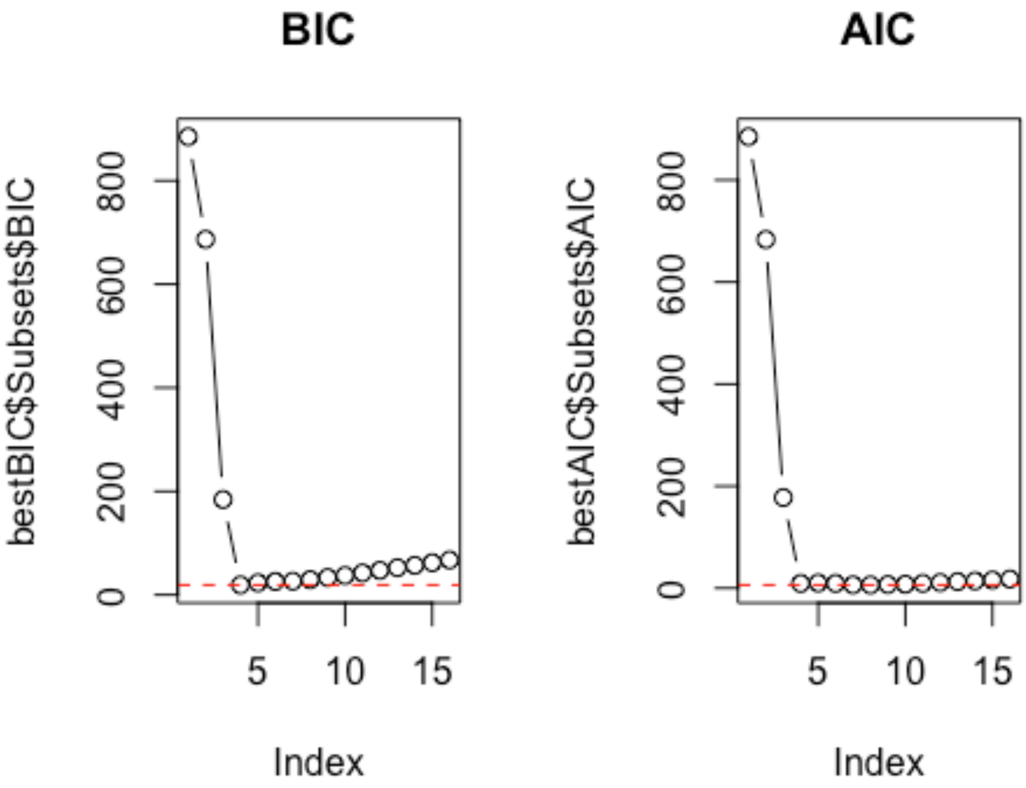
Best model according to AIC:

```
bestAIC$BestModel
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Coefficients:
## (Intercept)           V1           V2           V3          V10
##   2.8494207    1.6675486    4.1173775    2.3618339   -0.0001678
##         V11          V13          V15
##  -0.0066763    0.0019272   -0.0001404
```

Plots

```
par(mfrow=c(1,2))
plot(bestBIC$Subsets$BIC, main = "BIC",type="b")
abline(h = min(bestBIC$Subsets$BIC), col = 2, lty = 2)
plot(bestAIC$Subsets$AIC, main = "AIC",type="b")
abline(h = min(bestAIC$Subsets$AIC), col = 2, lty = 2)
```



###Conclusion ###The BIC identifies the following 3-predictor model (plus the intercept):

$$Y_{hat} = 2.904 + 2.035X + 4.010X^2 + 2.011X^3$$

###The AIC identifies the following 7-predictor model (plus the intercept):

$$\begin{aligned} &Y \\ &= 2.849 + 1.668X + 4.117X^2 + 2.362X^3 - 0.0002X^{10} - 0.007X^{11} + 0.002X^{13} \\ &- 0.0001X^{15} \end{aligned}$$

Part D

```
#run the stepAIC function to identify best models
library(MASS)
min.mod <- lm(y ~ 1, data = df)
max.mod <- lm(y ~ ., data = df)
scp <- list(lower = min.mod, upper = max.mod)
fwdBIC <- stepAIC(min.mod,
                  direction = 'forward',
                  scope = scp,
                  k = log(200))
```

```
## Start:  AIC=890.24
## y ~ 1
##
```

```
##        Df Sum of Sq     RSS    AIC
## + V3    1   10660.7  6037.2 692.07
## + V1    1    9531.7  7166.2 726.36
## + V5    1    7627.4  9070.5 773.49
## + V7    1    5205.6 11492.3 820.82
## + V9    1    3546.0 13151.9 847.80
## + V2    1    2919.6 13778.3 857.10
## + V11   1    2455.1 14242.8 863.73
## + V13   1    1755.3 14942.6 873.33
## + V4    1    1442.3 15255.6 877.47
## + V15   1    1308.8 15389.1 879.22
## + V6    1     484.8 16213.1 889.65
## <none>              16697.9 890.24
## + V14   1     112.5 16585.4 894.19
## + V8    1      89.3 16608.6 894.47
## + V12   1      38.5 16659.4 895.08
## + V10   1       0.1 16697.8 895.54
##
## Step:  AIC=692.07
## y ~ V3
##
##        Df Sum of Sq    RSS    AIC
## + V2    1    5560.8  476.4 189.49
## + V4    1    4909.0 1128.2 361.90
## + V6    1    4002.1 2035.1 479.89
## + V8    1    3241.0 2796.2 543.43
## + V10   1    2622.3 3414.9 583.41
## + V12   1    2141.5 3895.7 609.76
## + V14   1    1784.4 4252.7 627.30
## + V9    1    1148.7 4888.5 655.16
## + V7    1    1139.5 4897.7 655.54
## + V11   1    1120.7 4916.5 656.30
## + V13   1    1079.8 4957.4 657.96
## + V5    1    1056.3 4980.9 658.90
## + V15   1    1038.2 4999.0 659.63
## + V1    1     591.7 5445.5 676.74
## <none>              6037.2 692.07
##
## Step:  AIC=189.49
## y ~ V3 + V2
##
##        Df Sum of Sq    RSS     AIC
## + V1    1   273.131 203.28  24.447
## + V5    1   172.774 303.64 104.697
## + V7    1   124.845 351.57 134.009
## + V9    1    93.379 383.03 151.154
## + V11   1    73.380 403.03 161.333
## + V13   1    60.591 415.82 167.580
## + V15   1    52.253 424.16 171.551
## + V14   1    31.597 444.81 181.061
## + V12   1    29.843 446.57 181.848
## + V10   1    27.007 449.40 183.114
## + V8    1    22.830 453.58 184.965
## + V6    1    17.450 458.96 187.323
## <none>              476.41 189.488
## + V4    1    11.545 464.87 189.880
##
## Step:  AIC=24.45
## y ~ V3 + V2 + V1
##
##        Df Sum of Sq    RSS    AIC
## <none>              203.28 24.447
## + V5    1   1.14069 202.14 28.620
## + V7    1   0.76935 202.51 28.987
## + V15   1   0.62366 202.66 29.131
## + V9    1   0.62111 202.66 29.133
## + V13   1   0.59072 202.69 29.163
## + V11   1   0.58077 202.70 29.173
## + V4    1   0.44787 202.83 29.304
## + V14   1   0.23662 203.04 29.512
## + V6    1   0.23189 203.05 29.517
## + V12   1   0.09557 203.18 29.651
## + V8    1   0.04019 203.24 29.706
## + V10   1   0.00605 203.27 29.739
```

```
fwdAIC <- stepAIC(min.mod,
                  direction = 'forward',
                  scope = scp)
```

```
## Start:  AIC=886.94
## y ~ 1
## 
##         Df Sum of Sq     RSS    AIC
## + V3    1   10660.7  6037.2 685.48
## + V1    1    9531.7  7166.2 719.76
## + V5    1    7627.4  9070.5 766.89
## + V7    1    5205.6 11492.3 814.22
## + V9    1    3546.0 13151.9 841.20
## + V2    1    2919.6 13778.3 850.51
## + V11   1    2455.1 14242.8 857.14
## + V13   1    1755.3 14942.6 866.73
## + V4    1    1442.3 15255.6 870.88
## + V15   1    1308.8 15389.1 872.62
## + V6    1     484.8 16213.1 883.05
## <none>              16697.9 886.94
## + V14   1     112.5 16585.4 887.59
## + V8    1      89.3 16608.6 887.87
## + V12   1      38.5 16659.4 888.48
## + V10   1       0.1 16697.8 888.94
## 
## Step:  AIC=685.48
## y ~ V3
## 
##         Df Sum of Sq    RSS    AIC
## + V2    1    5560.8  476.4 179.59
## + V4    1    4909.0 1128.2 352.01
## + V6    1    4002.1 2035.1 470.00
## + V8    1    3241.0 2796.2 533.54
## + V10   1    2622.3 3414.9 573.52
## + V12   1    2141.5 3895.7 599.86
## + V14   1    1784.4 4252.7 617.40
## + V9    1    1148.7 4888.5 645.26
## + V7    1    1139.5 4897.7 645.64
## + V11   1    1120.7 4916.5 646.41
## + V13   1    1079.8 4957.4 648.06
## + V5    1    1056.3 4980.9 649.01
## + V15   1    1038.2 4999.0 649.74
## + V1    1     591.7 5445.5 666.84
## <none>              6037.2 685.48
## 
## Step:  AIC=179.59
## y ~ V3 + V2
## 
##         Df Sum of Sq    RSS     AIC
## + V1    1   273.131 203.28  11.253
## + V5    1   172.774 303.64  91.504
## + V7    1   124.845 351.57 120.816
## + V9    1    93.379 383.03 137.960
## + V11   1    73.380 403.03 148.139
## + V13   1    60.591 415.82 154.387
## + V15   1    52.253 424.16 158.358
## + V14   1    31.597 444.81 167.868
## + V12   1    29.843 446.57 168.655
## + V10   1    27.007 449.40 169.921
## + V8    1    22.830 453.58 171.772
## + V6    1    17.450 458.96 174.129
## + V4    1    11.545 464.87 176.686
## <none>              476.41 179.593
```

```
##
## Step:  AIC=11.25
## y ~ V3 + V2 + V1
##
##        Df Sum of Sq    RSS    AIC
## <none>              203.28 11.253
## + V5    1   1.14069 202.14 12.128
## + V7    1   0.76935 202.51 12.495
## + V15   1   0.62366 202.66 12.639
## + V9    1   0.62111 202.66 12.641
## + V13   1   0.59072 202.69 12.671
## + V11   1   0.58077 202.70 12.681
## + V4    1   0.44787 202.83 12.812
## + V14   1   0.23662 203.04 13.021
## + V6    1   0.23189 203.05 13.025
## + V12   1   0.09557 203.18 13.159
## + V8    1   0.04019 203.24 13.214
## + V10   1   0.00605 203.27 13.248
```

The output shows that the two methods did come to different results.

That said, they confirm the importance of the cubed term:

```
#BIC used in forward stepwise selection results in:
fwdBIC$coefficients

## (Intercept)          V3          V2          V1
##    2.904387    2.011071    4.010162    2.034678

#AIC used in forward stepwise selection results in:
fwdAIC$coefficients

## (Intercept)          V3          V2          V1
##    2.904387    2.011071    4.010162    2.034678
```