

# Ricker\_polynomial.R

SIU850486795

2023-11-14

```
# Ricker_polynomial.R
# Polynomial regression for Ricker data

# Load necessary libraries
library(ggplot2)
library(car)

## Loading required package: carData
library(QuantPsyc)

## Loading required package: boot
##
## Attaching package: 'boot'
## The following object is masked from 'package:car':
##
##      logit
## Loading required package: MASS
##
## Attaching package: 'QuantPsyc'
## The following object is masked from 'package:base':
##
##      norm

# Read in data set
rickerdata <- read.table(header=T,colClasses=rep("numeric",2),text="
n logR
5  0.42
10 0.33
20 0.48
30 0.03
40 -0.18
50 -0.16
60 0.08
70 -1.20
80 -1.45
90 -1.72
100 -2.67
")

# Apply transformations here
```

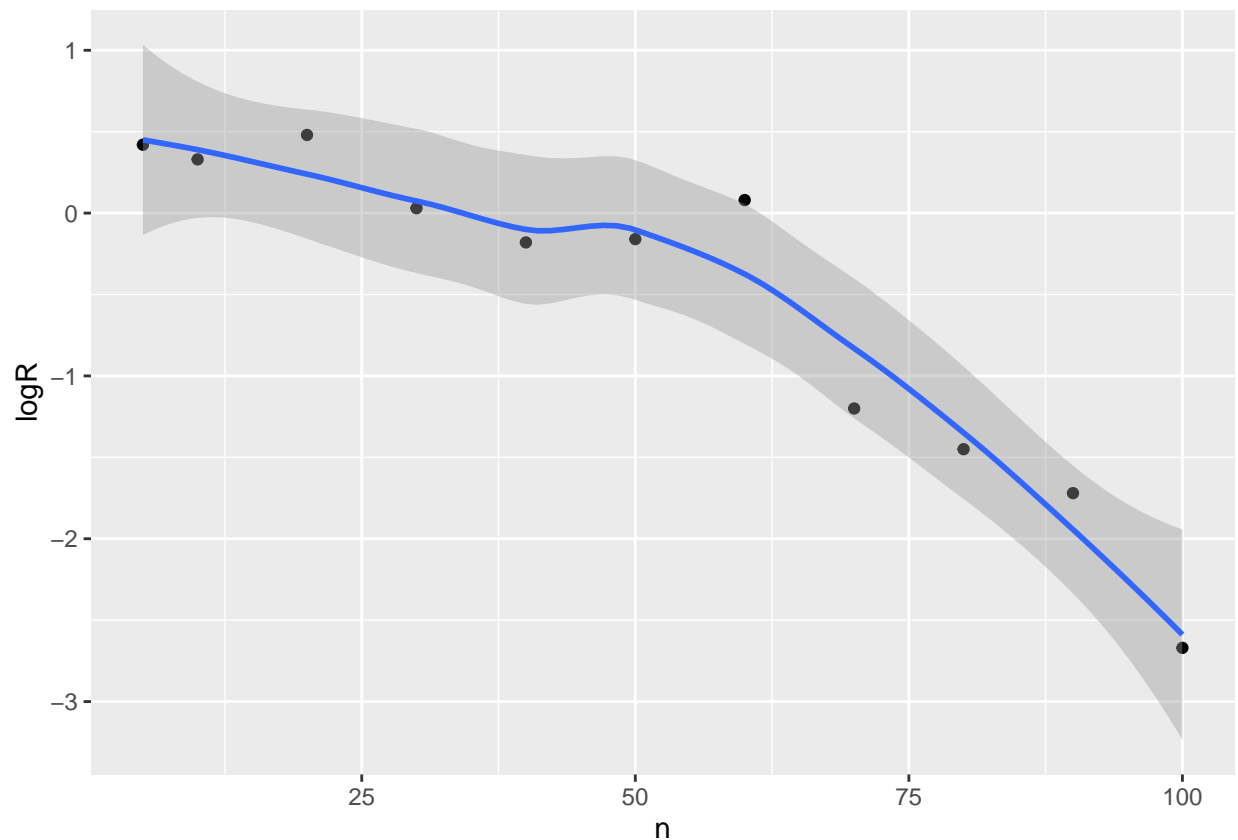
```
rickerdata <- transform(rickerdata,x=n-mean(n))
rickerdata <- transform(rickerdata,x2=x^2)
rickerdata <- transform(rickerdata,x3=x^3)
```

```
# Print data
rickerdata
```

```
##      n  logR      x      x2      x3
## 1    5  0.42 -45.4545455 2066.1157025 -9.391435e+04
## 2   10  0.33 -40.4545455 1636.5702479 -6.620671e+04
## 3   20  0.48 -30.4545455  927.4793388 -2.824596e+04
## 4   30  0.03 -20.4545455  418.3884298 -8.557945e+03
## 5   40 -0.18 -10.4545455  109.2975207 -1.142656e+03
## 6   50 -0.16  -0.4545455    0.2066116 -9.391435e-02
## 7   60  0.08   9.5454545   91.1157025  8.697408e+02
## 8   70 -1.20  19.5454545  382.0247934  7.466848e+03
## 9   80 -1.45  29.5454545  872.9338843  2.579123e+04
## 10  90 -1.72  39.5454545 1563.8429752  6.184288e+04
## 11 100 -2.67  49.5454545 2454.7520661  1.216218e+05
```

```
# Plot data and fit smooth line
ggplot(rickerdata,aes(n,logR))+
  geom_point()+geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```

# Polynomial regression analysis
regout <- lm(logR~x+x2+x3,data=rickerdata)
summary(regout)

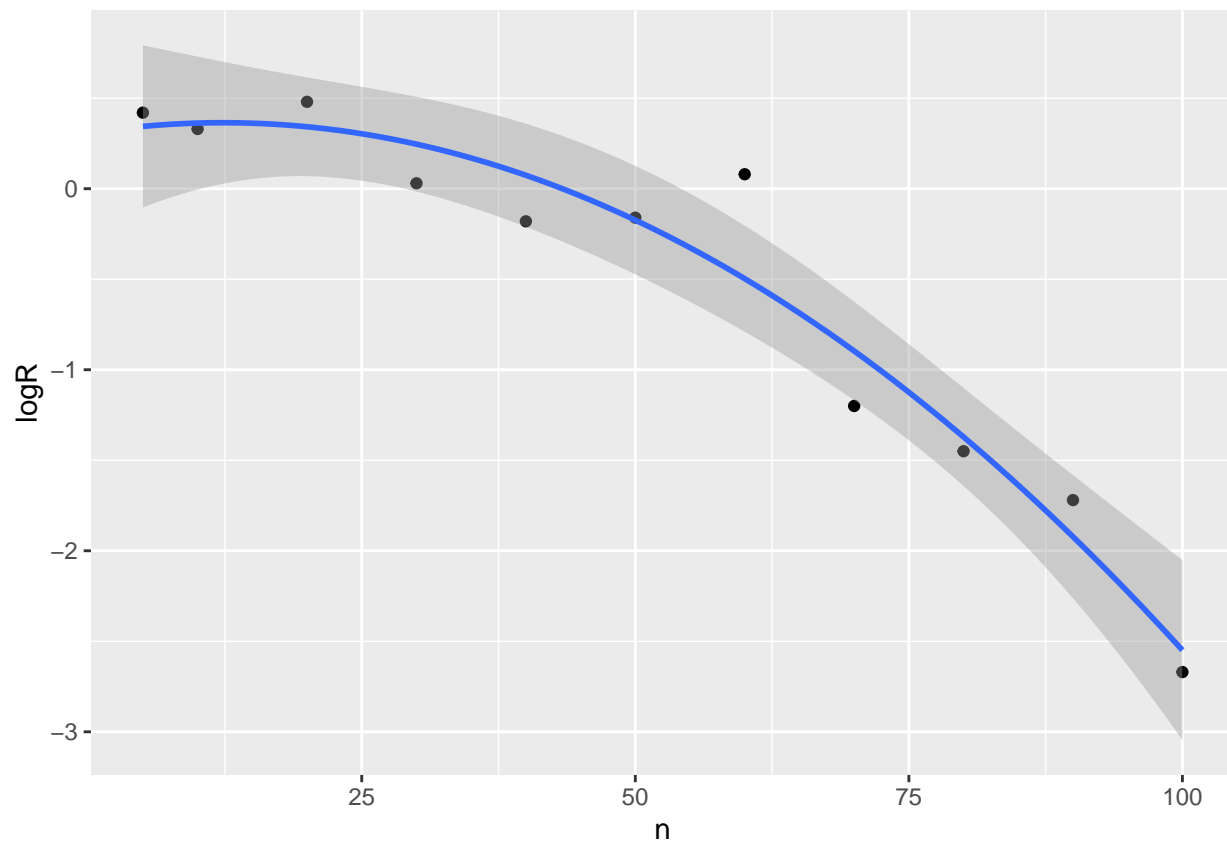
##
## Call:
## lm(formula = logR ~ x + x2 + x3, data = rickerdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37038 -0.14460 -0.03357  0.10204  0.54251
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.955e-01  1.356e-01  -1.442   0.1926
## x            -2.424e-02  7.660e-03  -3.164   0.0158 *
## x2           -3.644e-04  1.097e-04  -3.321   0.0127 *
## x3           -2.843e-06  4.334e-06  -0.656   0.5328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2898 on 7 degrees of freedom
## Multiple R-squared:  0.946, Adjusted R-squared:  0.9229
## F-statistic: 40.89 on 3 and 7 DF, p-value: 8.326e-05

# Look at Type I tests to determine order of polynomial
anova(regout)

## Analysis of Variance Table
##
## Response: logR
##      Df Sum Sq Mean Sq  F value    Pr(>F)
## x      1  9.2244   9.2244 109.8063 1.57e-05 ***
## x2     1  1.0457   1.0457  12.4474 0.009623 **
## x3     1  0.0361   0.0361   0.4303 0.532803
## Residuals 7 0.5880   0.0840
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Preceding analysis suggests second-order polynomial adequate
# Plot the data and second-order polynomial
ggplot(rickerdata,aes(n,logR))+
  geom_point()+
  stat_smooth(method="lm",formula=y~poly(x,2))

```



```
# Polynomial regression with second-order polynomial
```

```
regout2 <- lm(logR~x+x2,data=rickerdata)
```

```
summary(regout2)
```

```
##
```

```
## Call:
```

```
## lm(formula = logR ~ x + x2, data = rickerdata)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -0.30372 -0.16858 -0.03253  0.10676  0.57698
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -0.1865353  0.1300048  -1.435  0.18925
```

```
## x           -0.0289045  0.0027298 -10.589 5.53e-06 ***
```

```
## x2           -0.0003790  0.0001035  -3.661  0.00639 **
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Residual standard error: 0.2793 on 8 degrees of freedom
```

```
## Multiple R-squared:  0.9427, Adjusted R-squared:  0.9284
```

```
## F-statistic: 65.81 on 2 and 8 DF,  p-value: 1.078e-05
```

```
# 95% confidence intervals for regression coefficients
```

```
confint(regout2)
```

```
##                2.5 %        97.5 %
## (Intercept) -0.4863269680  0.1132563915
## x           -0.0351994386 -0.0226096529
## x2          -0.0006177359 -0.0001402637
```

```
# Standardized regression coefficients
lm.beta(regout2)
```

```
##          x          x2
## -0.8983195 -0.3105803
```

```
# Tolerance values (1/vif)
tol <- 1/vif(regout2)
tol
```

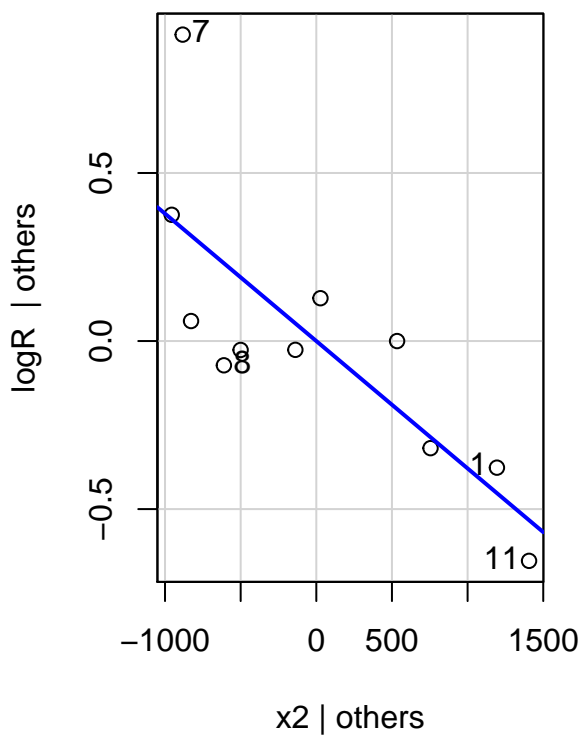
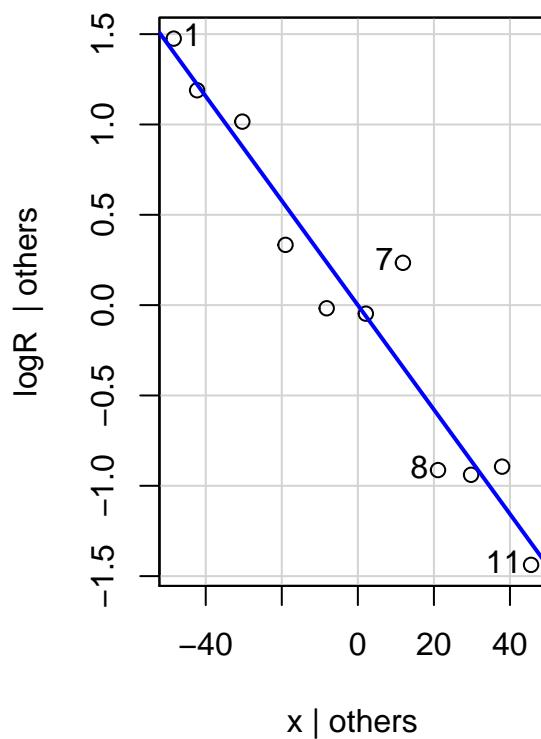
```
##          x          x2
## 0.9950481 0.9950481
```

```
# Variance inflation factors
vif(regout2)
```

```
##          x          x2
## 1.004977 1.004977
```

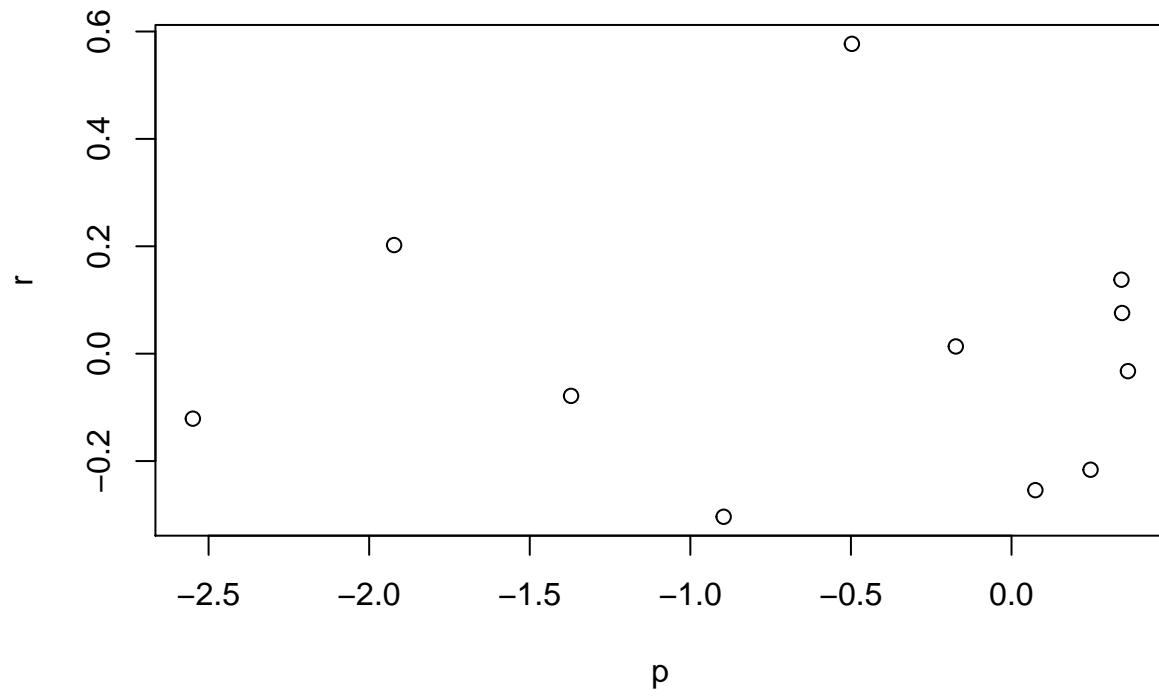
```
# Residual-residual plots
avPlots(regout2)
```

## Added-Variable Plots



```
# Diagnostic plots to check regression assumptions
p <- predict(regout2)
r <- resid(regout2)
```

```
plot(p,r)
```



```
qqnorm(r)
```

**Normal Q-Q Plot**

