

## Lesson 09

Christopher A. Swenson (chris@cswenson.com)

11/27/2021

### Word recall (log-transforming a predictor)

Load the wordrecall data. Fit a simple linear regression model of prop on time. Display scatterplot of the data and add the regression line. Display residual plot with fitted (predicted) values on the horizontal axis. Display normal probability plot of the residuals and add a diagonal line to the plot. The argument “datax” determines which way round to plot the axes (false by default, which plots the data on the vertical axis, or true, which plots the data on the horizontal axis). Load the nortest package to access Anderson-Darling normality test. Create log(time) variable and fit a simple linear regression model of prop on log(time). Repeat diagnostic plots and normality test. Create prop<sup>-1.25</sup> variable and fit a simple linear regression model of prop<sup>-1.25</sup> on time. Repeat diagnostic plots and normality test. Use prop on log(time) model to find: 95% prediction interval for prop at time 1000. 95% confidence interval for expected change in prop for a 10-fold increase in time.

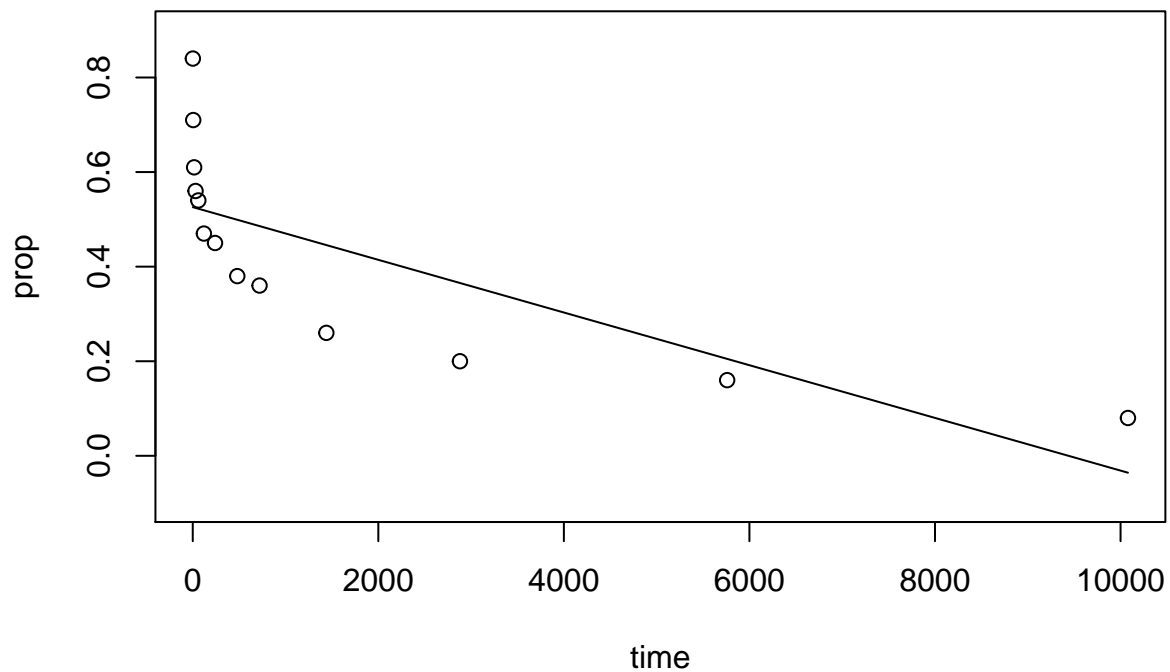
```
wordrecall <- read.table("./Data/wordrecall.txt", header=T)
attach(wordrecall)

model.1 <- lm(prop ~ time)
summary(model.1)

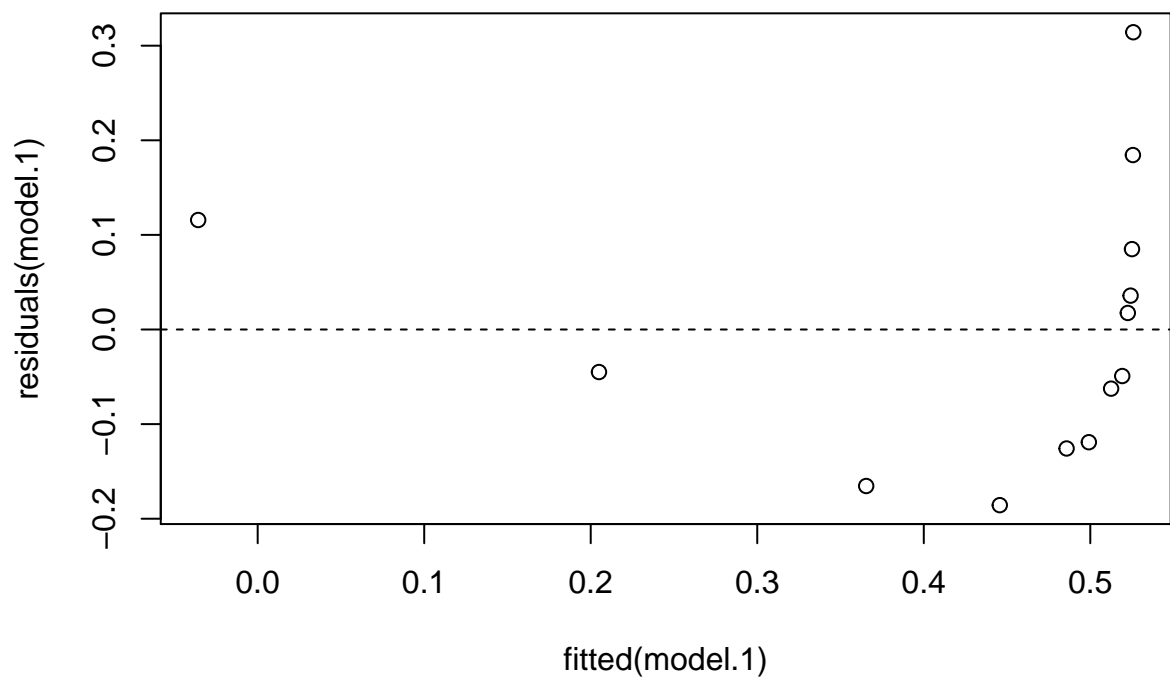
##
## Call:
## lm(formula = prop ~ time)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18564 -0.11913 -0.04495  0.08496  0.31418
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.259e-01  4.881e-02  10.774 3.49e-07 ***
## time        -5.571e-05  1.457e-05  -3.825  0.00282 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1523 on 11 degrees of freedom
## Multiple R-squared:  0.5709, Adjusted R-squared:  0.5318
## F-statistic: 14.63 on 1 and 11 DF,  p-value: 0.002817
#
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  5.259e-01  4.881e-02  10.774 3.49e-07 ***
# time        -5.571e-05  1.457e-05  -3.825  0.00282 **
# Multiple R-squared:  0.5709, Adjusted R-squared:  0.5318

plot(x=time, y=prop, ylim=c(-0.1, 0.9),
```

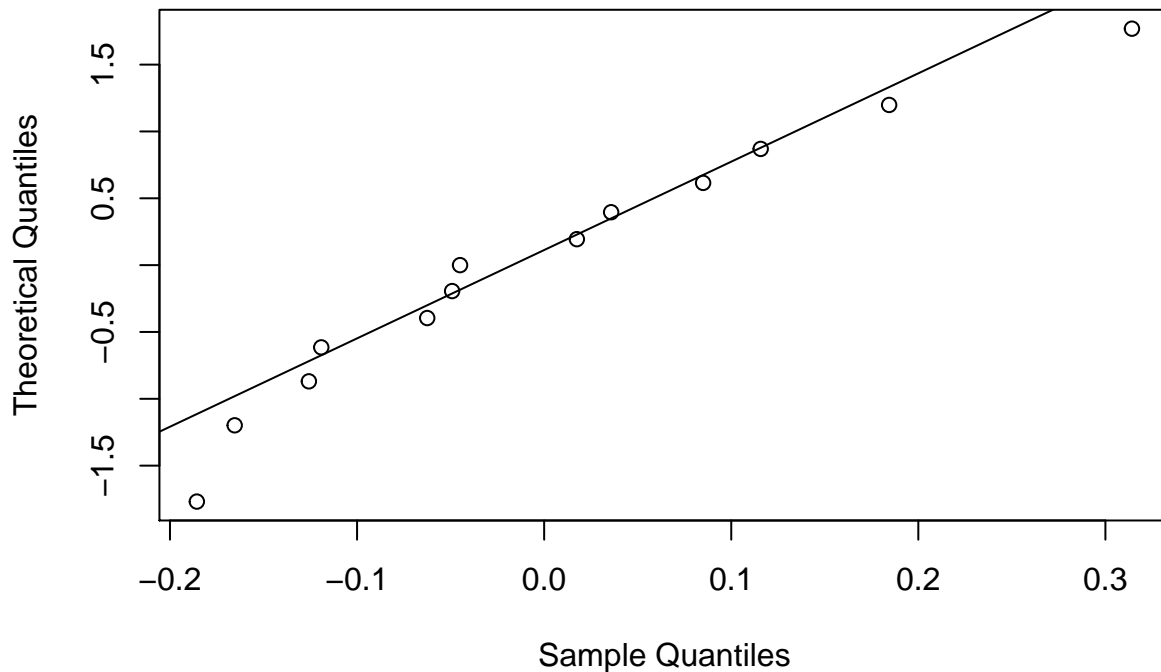
```
panel.last = lines(sort(time), fitted(model.1)[order(time)]))
```



```
plot(x=fitted(model.1), y=residuals(model.1),
     panel.last = abline(h=0, lty=2))
```



```
qqnorm(residuals(model.1), main="", datax=TRUE)
qqline(residuals(model.1), datax=TRUE)
```



```
library(nortest)
ad.test(residuals(model.1)) # A = 0.262, p-value = 0.6426
```

```
##
## Anderson-Darling normality test
##
## data: residuals(model.1)
## A = 0.26197, p-value = 0.6426
```

```
lntime <- log(time)
```

```
model.2 <- lm(prop ~ lntime)
summary(model.2)
```

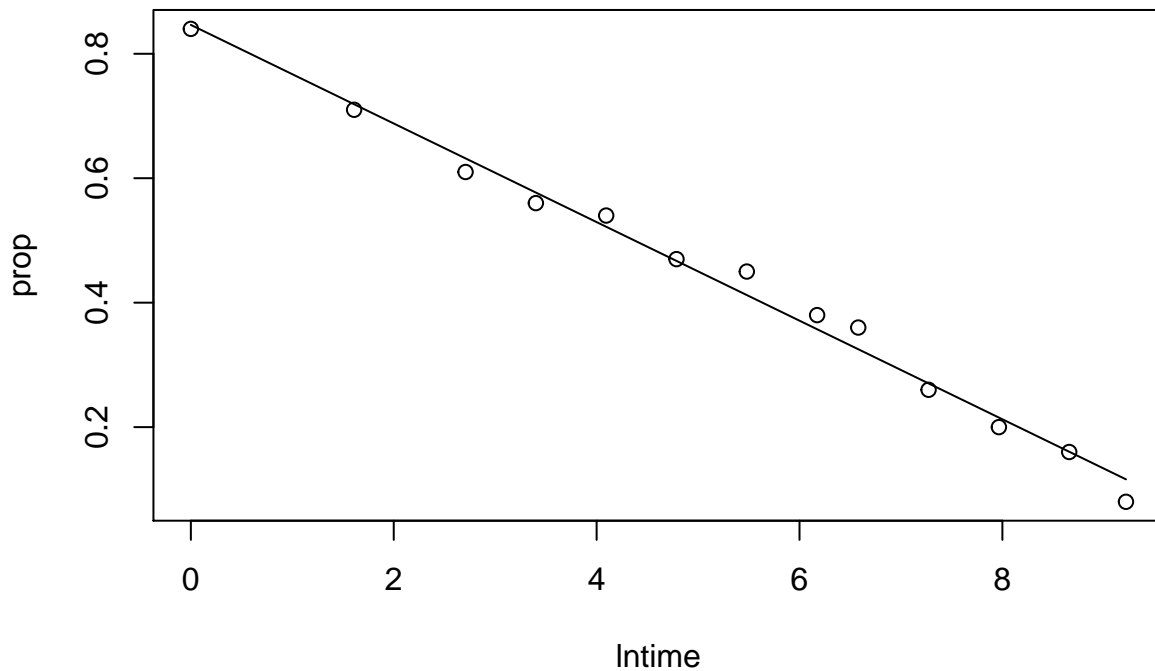
```
##
## Call:
## lm(formula = prop ~ lntime)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.036077 -0.015330 -0.006415  0.017967  0.037799
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.846415   0.014195   59.63 3.65e-15 ***
## lntime      -0.079227   0.002416  -32.80 2.53e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02339 on 11 degrees of freedom
## Multiple R-squared:  0.9899, Adjusted R-squared:  0.989
## F-statistic: 1076 on 1 and 11 DF, p-value: 2.525e-12
```

```

#               Estimate Std. Error t value Pr(>|t|)
# (Intercept)  0.846415   0.014195   59.63 3.65e-15 ***
# lntime       -0.079227   0.002416  -32.80 2.53e-12 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.02339 on 11 degrees of freedom
# Multiple R-squared:  0.9899, Adjusted R-squared:  0.989
# F-statistic: 1076 on 1 and 11 DF, p-value: 2.525e-12

plot(x=lntime, y=prop,
     panel.last = lines(sort(lntime), fitted(model.2)[order(lntime)]))

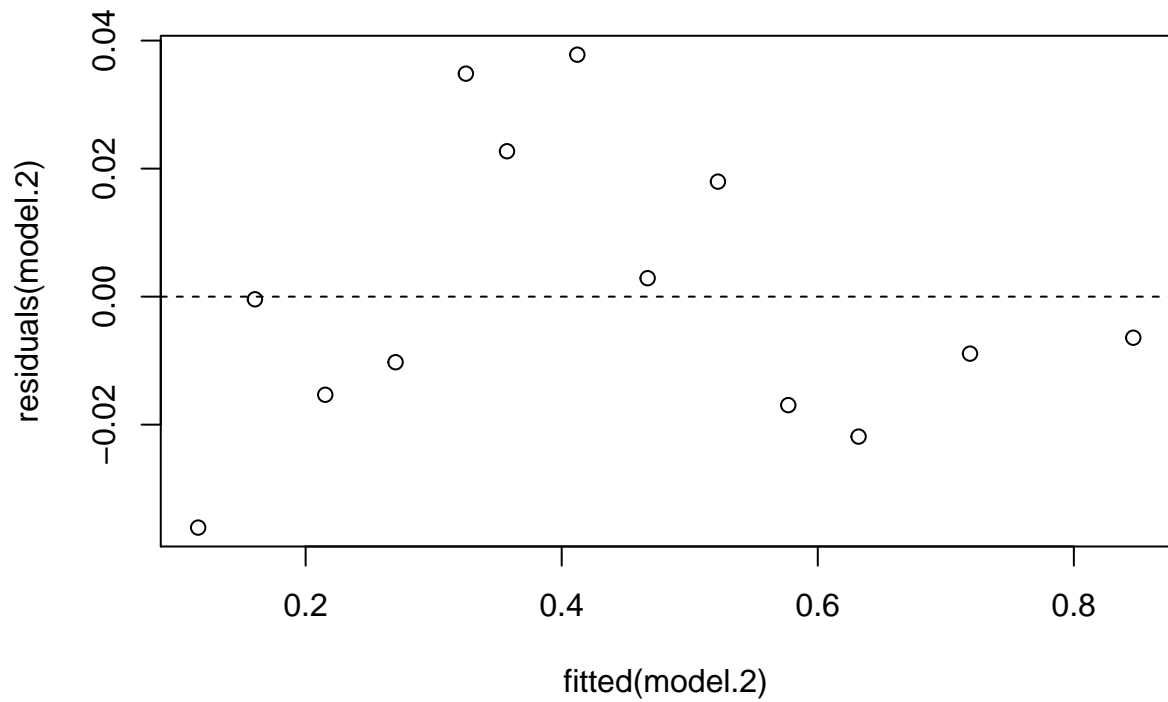
```



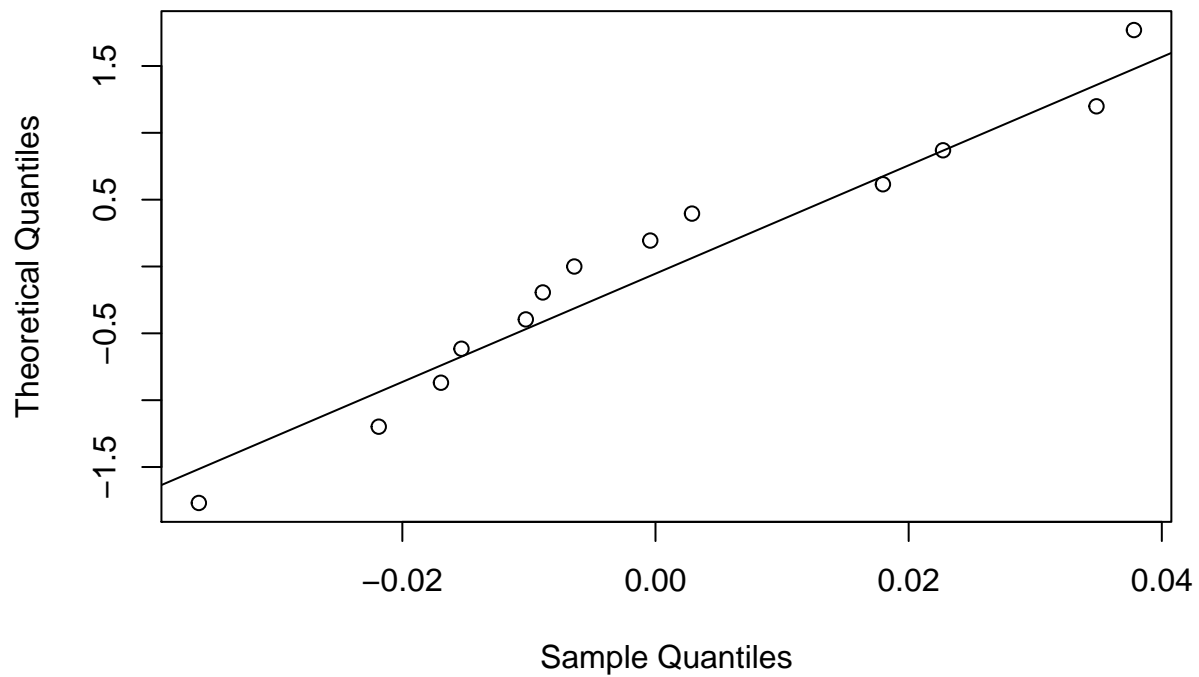
```

plot(x=fitted(model.2), y=residuals(model.2),
     panel.last = abline(h=0, lty=2))

```



```
qqnorm(residuals(model.2), main="", datax=TRUE)
qqline(residuals(model.2), datax=TRUE)
```



```
ad.test(residuals(model.2)) # A = 0.3216, p-value = 0.4869
```

```
##
## Anderson-Darling normality test
##
## data: residuals(model.2)
## A = 0.32157, p-value = 0.4869
```

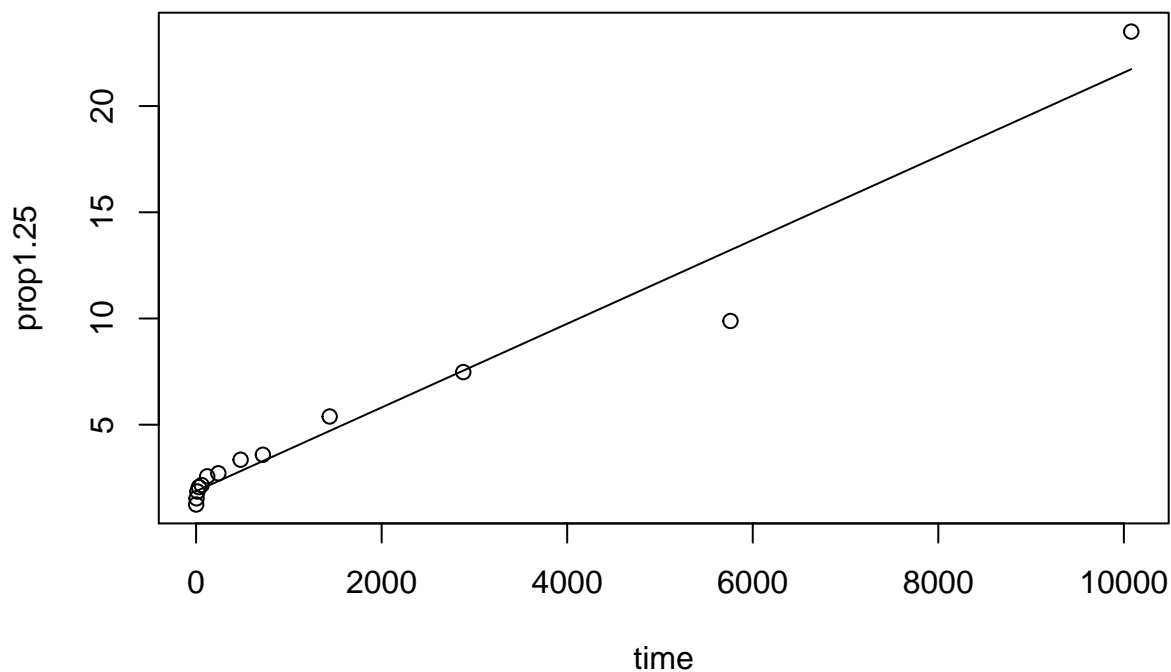
```
prop1.25 <- prop^-1.25
```

```
model.3 <- lm(prop1.25 ~ time)  
summary(model.3)
```

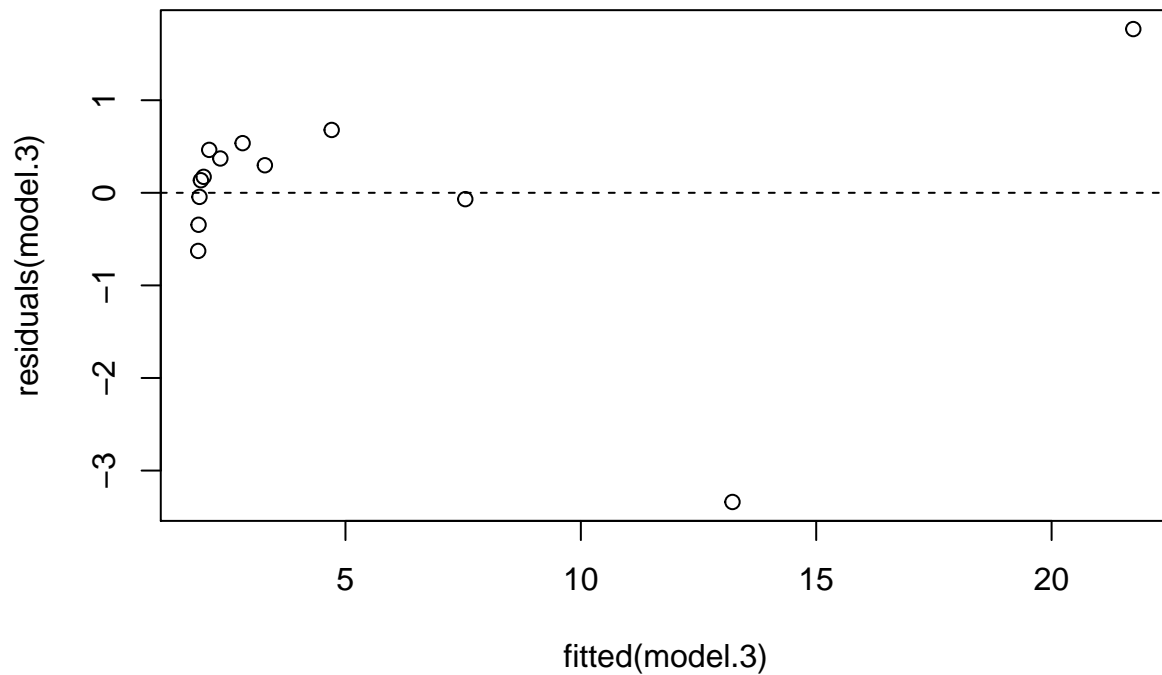
```
##  
## Call:  
## lm(formula = prop1.25 ~ time)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -3.3392 -0.0686  0.1726  0.4638  1.7685   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept) 1.8693698  0.3869678   4.831 0.000527 ***  
## time         0.0019708  0.0001155  17.067 2.91e-09 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 1.207 on 11 degrees of freedom  
## Multiple R-squared:  0.9636, Adjusted R-squared:  0.9603   
## F-statistic: 291.3 on 1 and 11 DF,  p-value: 2.909e-09
```

```
#              Estimate Std. Error t value Pr(>|t|)      
# (Intercept) 1.8693698  0.3869678   4.831 0.000527 ***  
# time         0.0019708  0.0001155  17.067 2.91e-09 ***  
# Multiple R-squared:  0.9636, Adjusted R-squared:  0.9603
```

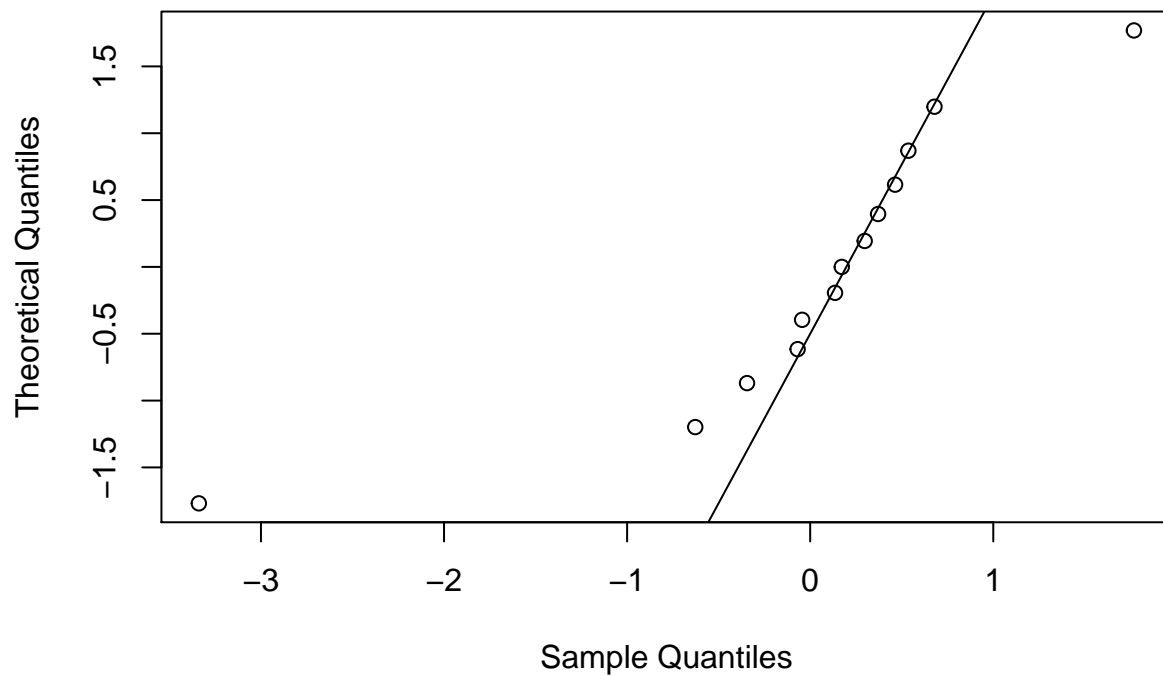
```
plot(x=time, y=prop1.25,  
      panel.last = lines(sort(time), fitted(model.3)[order(time)]))
```



```
plot(x=fitted(model.3), y=residuals(model.3),
     panel.last = abline(h=0, lty=2))
```



```
qqnorm(residuals(model.3), main="", datax=TRUE)
qqline(residuals(model.3), datax=TRUE)
```



```
ad.test(residuals(model.3)) # A = 1.191, p-value = 0.002584
```

```
##
## Anderson-Darling normality test
##
```

```
## data: residuals(model.3)
## A = 1.191, p-value = 0.002584

predict(model.2, interval="prediction",
        newdata=data.frame(lntime=log(1000)))

##          fit          lwr          upr
## 1 0.2991353 0.2449729 0.3532978

#          fit          lwr          upr
# 1 0.2991353 0.2449729 0.3532978

confint(model.2)[2,]*log(10) # 95% CI for 10-fold increase in time

##          2.5 %          97.5 %
## -0.1946689 -0.1701845

#          2.5 %          97.5 %
# -0.1946689 -0.1701845

detach(wordrecall)
```

## Mammal gestation (log-transforming the response)

Load the mammgest data. Fit a simple linear regression model of Gestation on Birthwgt. Display scatterplot of the data and add the regression line. Display residual plot with fitted (predicted) values on the horizontal axis. Display normal probability plot of the residuals and add a diagonal line to the plot. Apply the Anderson-Darling normality test using nortest package. Create  $\log(\text{Gestation})$  variable and fit a simple linear regression model of  $\log(\text{Gestation})$  on Birthwgt. Repeat diagnostic plots and normality test. Use  $\log(\text{Gestation})$  on Birthwgt model to find: 95% prediction interval for Gestation for a Birthwgt of 50. 95% confidence interval for proportional change in median Gestation for a 1 pound increase in Birthwgt. 95% confidence interval for proportional change in median Gestation for a 10 pound increase in Birthwgt.

```
mammgest <- read.table("./Data/mammgest.txt", header=T)
attach(mammgest)

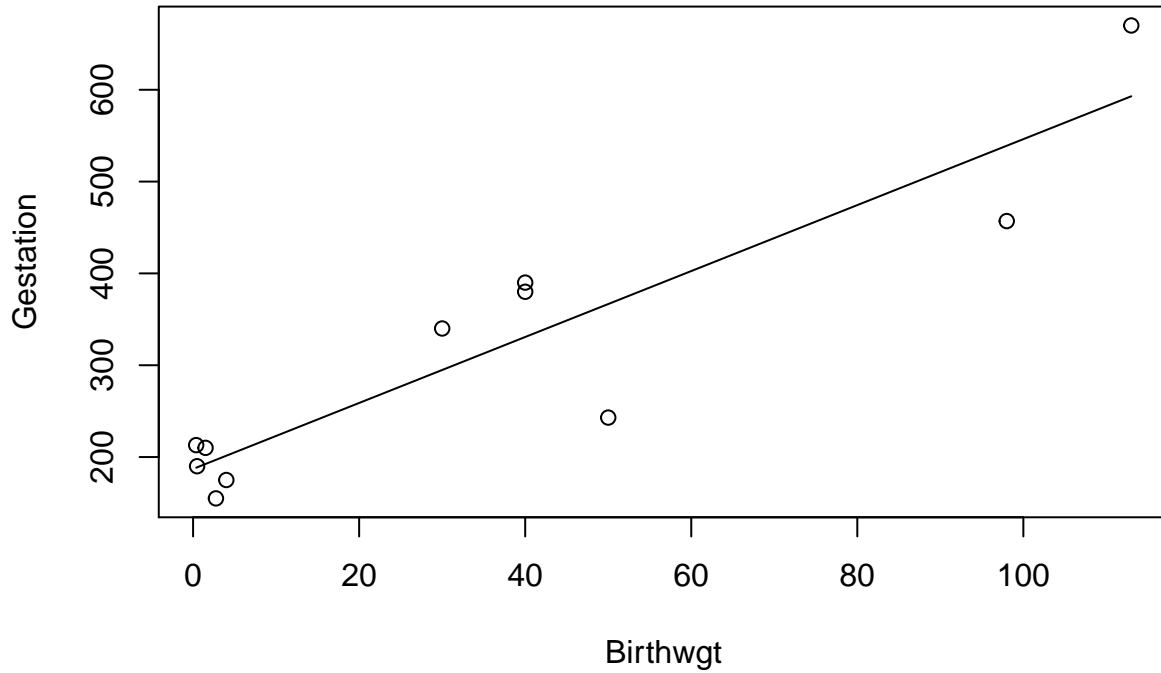
model.1 <- lm(Gestation ~ Birthwgt)
summary(model.1)

##
## Call:
## lm(formula = Gestation ~ Birthwgt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -123.65  -34.20   17.53   47.22   77.09
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  187.0837    26.9426   6.944 6.73e-05 ***
## Birthwgt       3.5914     0.5247   6.844 7.52e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.09 on 9 degrees of freedom
## Multiple R-squared:  0.8388, Adjusted R-squared:  0.8209
## F-statistic: 46.84 on 1 and 9 DF, p-value: 7.523e-05
```

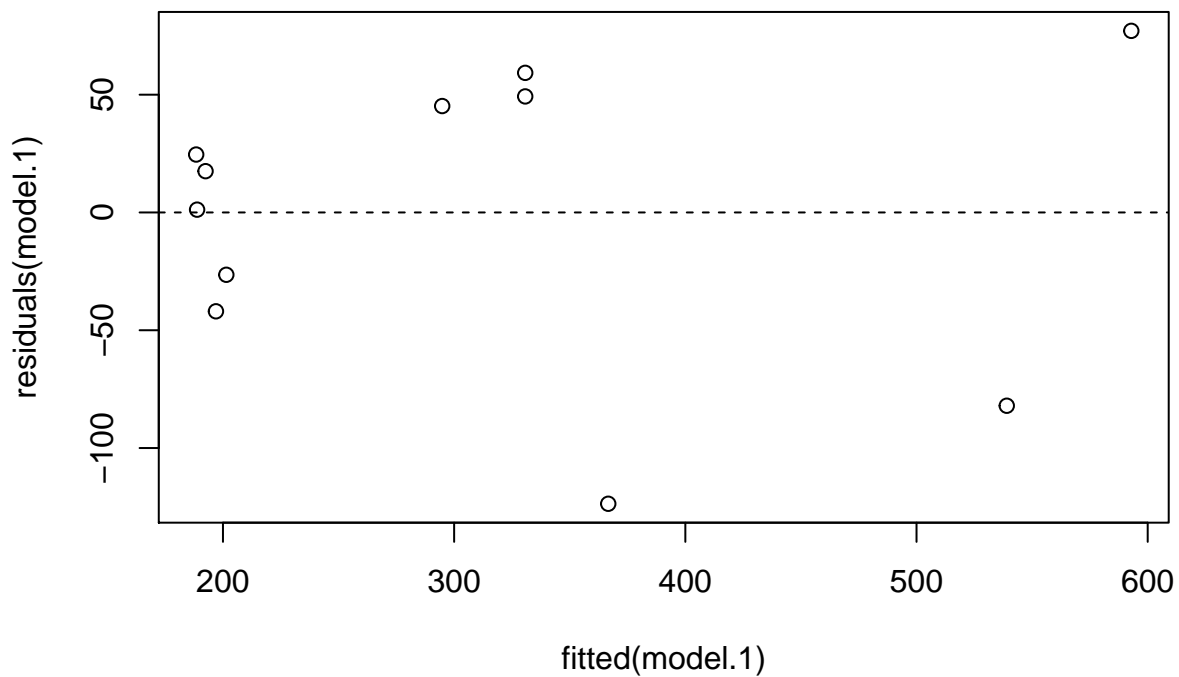


```
#           Estimate Std. Error t value Pr(>|t|)
# (Intercept) 187.0837    26.9426   6.944 6.73e-05 ***
# Birthwgt     3.5914     0.5247   6.844 7.52e-05 ***
# Multiple R-squared:  0.8388, Adjusted R-squared:  0.8209
```

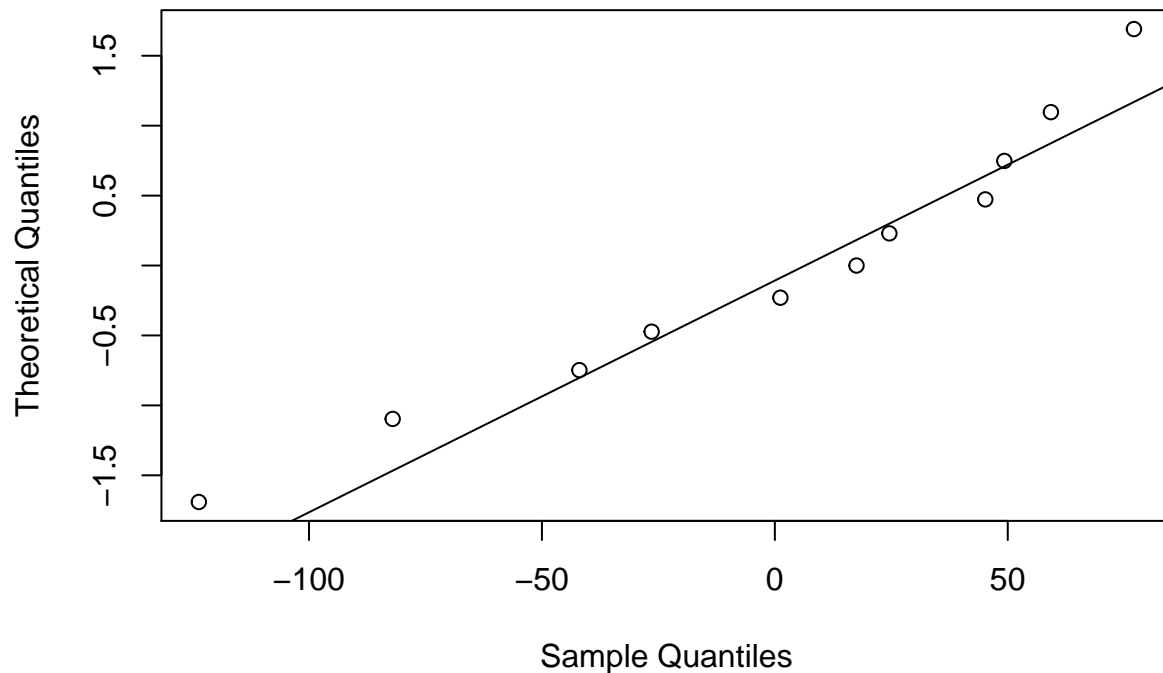
```
plot(x=Birthwgt, y=Gestation,
      panel.last = lines(sort(Birthwgt), fitted(model.1)[order(Birthwgt)]))
```



```
plot(x=fitted(model.1), y=residuals(model.1),
      panel.last = abline(h=0, lty=2))
```



```
qqnorm(residuals(model.1), main="", datax=TRUE)
qqline(residuals(model.1), datax=TRUE)
```



```
ad.test(residuals(model.1)) # A = 0.3116, p-value = 0.503
```

```
##
## Anderson-Darling normality test
##
## data: residuals(model.1)
## A = 0.31159, p-value = 0.503
```

```
lnGest <- log(Gestation)
```

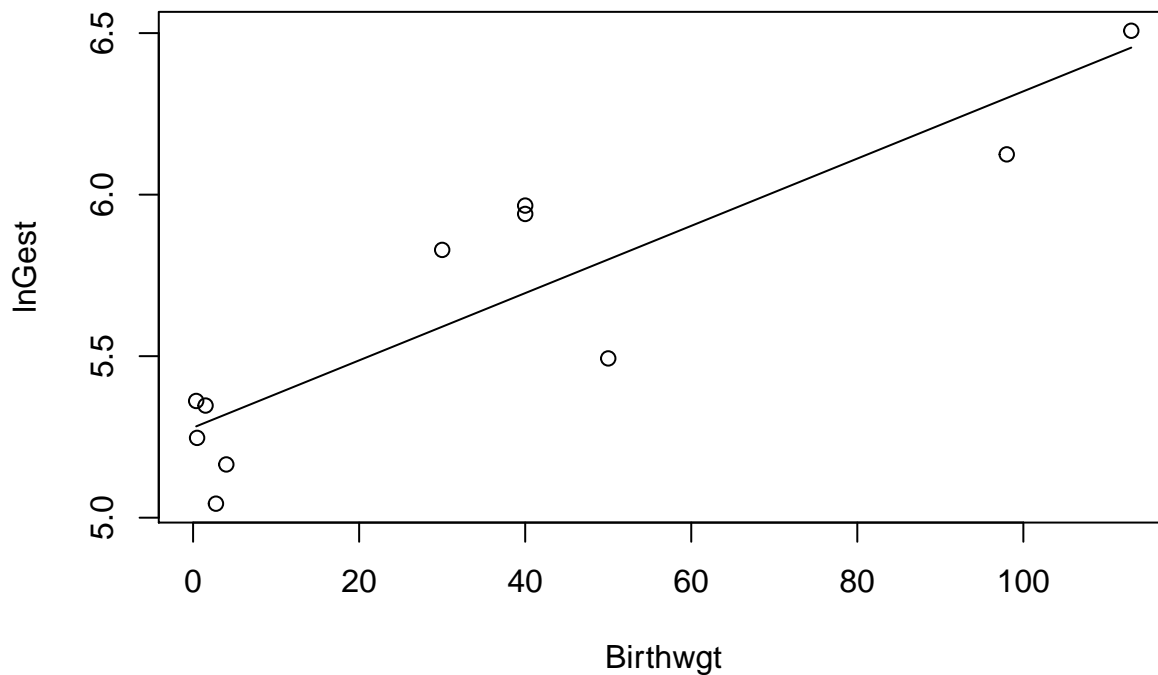
```
model.2 <- lm(lnGest ~ Birthwgt)
summary(model.2)
```

```
##
## Call:
## lm(formula = lnGest ~ Birthwgt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.3063 -0.1650  0.0521  0.1582  0.2709
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.278817   0.088177  59.866 5.1e-13 ***
## Birthwgt     0.010410   0.001717   6.062 0.000188 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2163 on 9 degrees of freedom
## Multiple R-squared:  0.8033, Adjusted R-squared:  0.7814
```

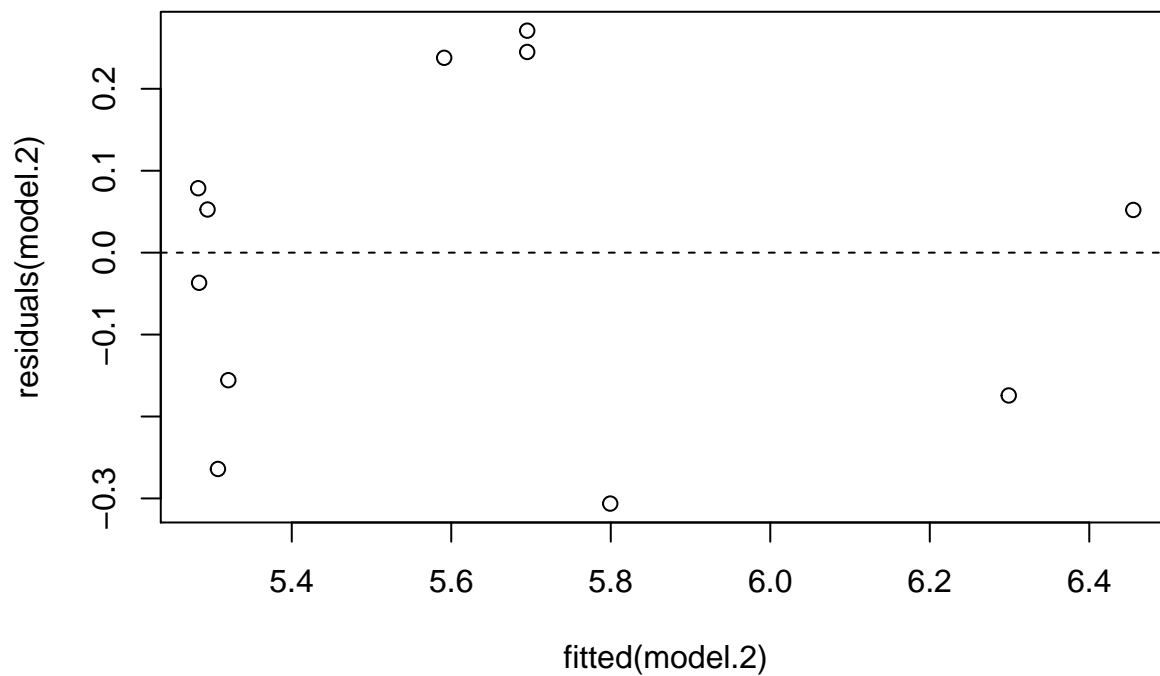
```
## F-statistic: 36.75 on 1 and 9 DF, p-value: 0.0001878
```

```
#           Estimate Std. Error t value Pr(>|t|)
# (Intercept) 5.278817  0.088177  59.866  5.1e-13 ***
# Birthwgt    0.010410  0.001717   6.062 0.000188 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.2163 on 9 degrees of freedom
# Multiple R-squared:  0.8033, Adjusted R-squared:  0.7814
# F-statistic: 36.75 on 1 and 9 DF, p-value: 0.0001878
```

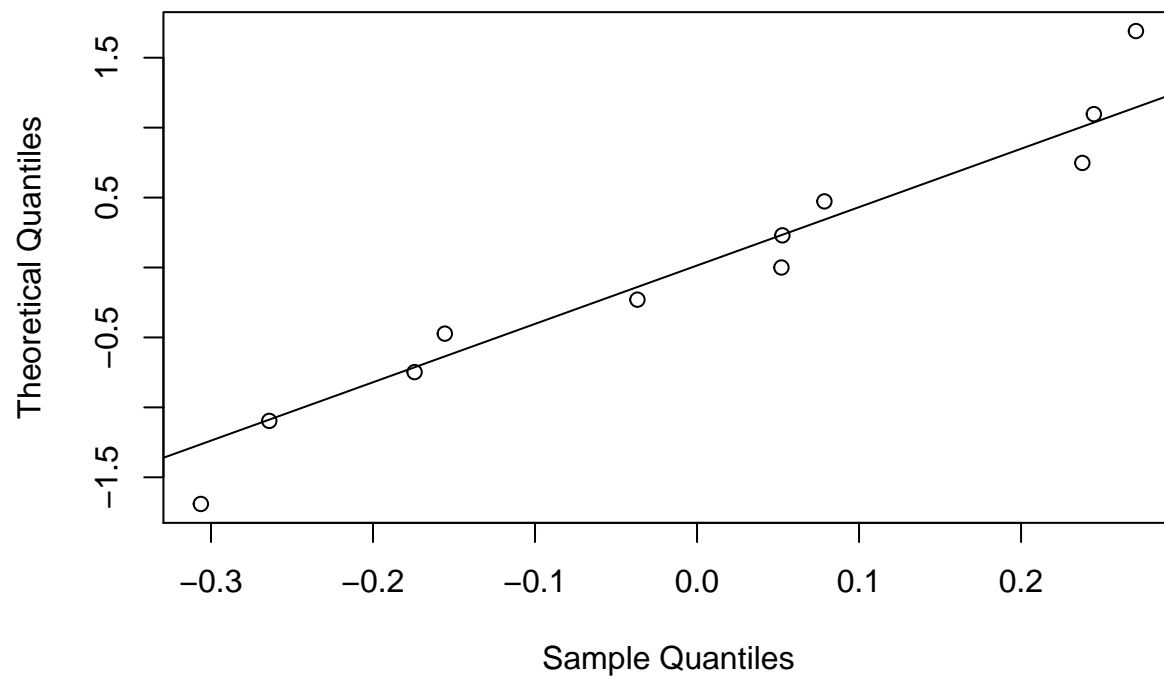
```
plot(x=Birthwgt, y=lnGest,
     panel.last = lines(sort(Birthwgt), fitted(model.2)[order(Birthwgt)]))
```



```
plot(x=fitted(model.2), y=residuals(model.2),
     panel.last = abline(h=0, lty=2))
```



```
qqnorm(residuals(model.2), main="", datax=TRUE)
qqline(residuals(model.2), datax=TRUE)
```



```
ad.test(residuals(model.2)) # A = 0.3135, p-value = 0.4963
```

```
##
##  Anderson-Darling normality test
##
## data:  residuals(model.2)
## A = 0.31355, p-value = 0.4963
```

```

exp(predict(model.2, interval="prediction",
           newdata=data.frame(Birthwgt=50)))

##          fit          lwr          upr
## 1 330.0781 197.3013 552.2092

#          fit          lwr          upr
# 1 330.0781 197.3013 552.2092

# proportional change in median gestation for 1-unit increase in birthwgt
exp(coefficients(model.2)[2]) # 1.010465

## Birthwgt
## 1.010465

exp(confint(model.2)[2,]) # 95% CI

##      2.5 %      97.5 %
## 1.006547 1.014398

#      2.5 %      97.5 %
# 1.006547 1.014398

# proportional change in median gestation for 10-unit increase in birthwgt
exp(10*coefficients(model.2)[2]) # 1.109714

## Birthwgt
## 1.109714

exp(10*confint(model.2)[2,]) # 95% CI

##      2.5 %      97.5 %
## 1.067429 1.153674

#      2.5 %      97.5 %
# 1.067429 1.153674

detach(mammgest)

```

## Shortleaf pine trees (log-transforming both response and predictor)

Load the shortleaf data. Fit a simple linear regression model of Vol on Diam. Display scatterplot of the data and add the regression line. Display residual plot with fitted (predicted) values on the horizontal axis. Display normal probability plot of the residuals and add a diagonal line to the plot. Apply the Anderson-Darling normality test using the nortest package. Create log(Diam) variable and fit a simple linear regression model of Vol on log(Diam). Repeat diagnostic plots and normality test. Create log(Vol) variable and fit a simple linear regression model of log(Vol) on log(Diam). Repeat diagnostic plots and normality test. Use log(Vol) on log(Diam) model to find: 95% confidence interval for median Vol for a Diam of 10. 95% confidence interval for proportional change in median Vol for a 2-fold increase in Diam.

```

shortleaf <- read.table("./Data/shortleaf.txt", header=T)
attach(shortleaf)

model.1 <- lm(Vol ~ Diam)
summary(model.1)

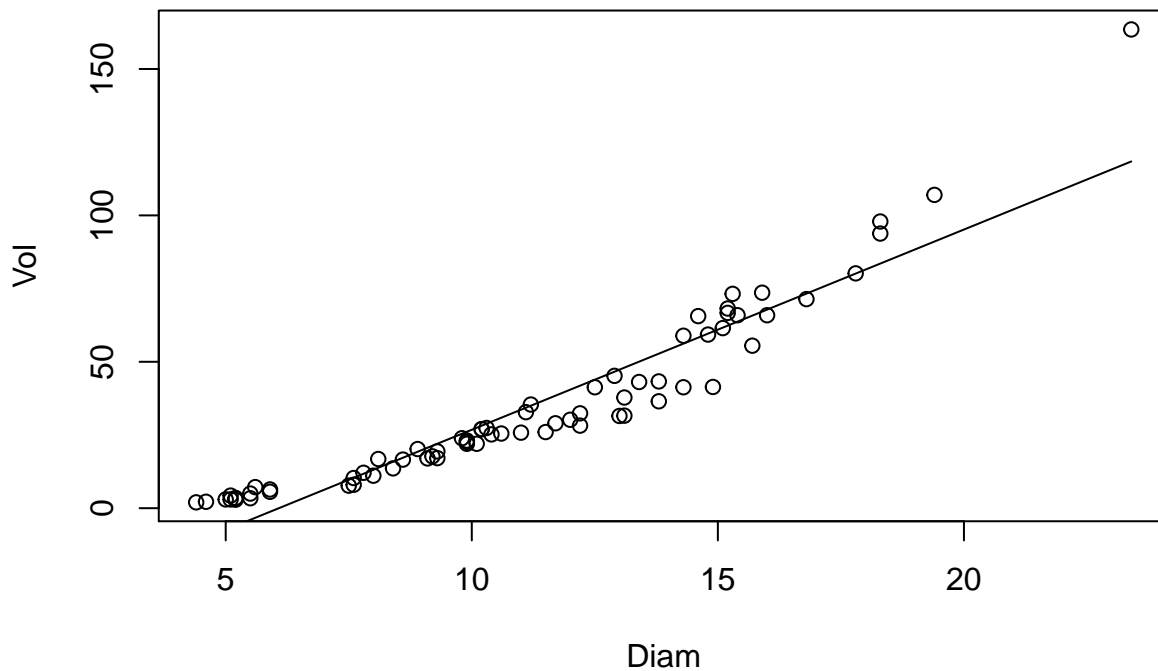
##
## Call:

```

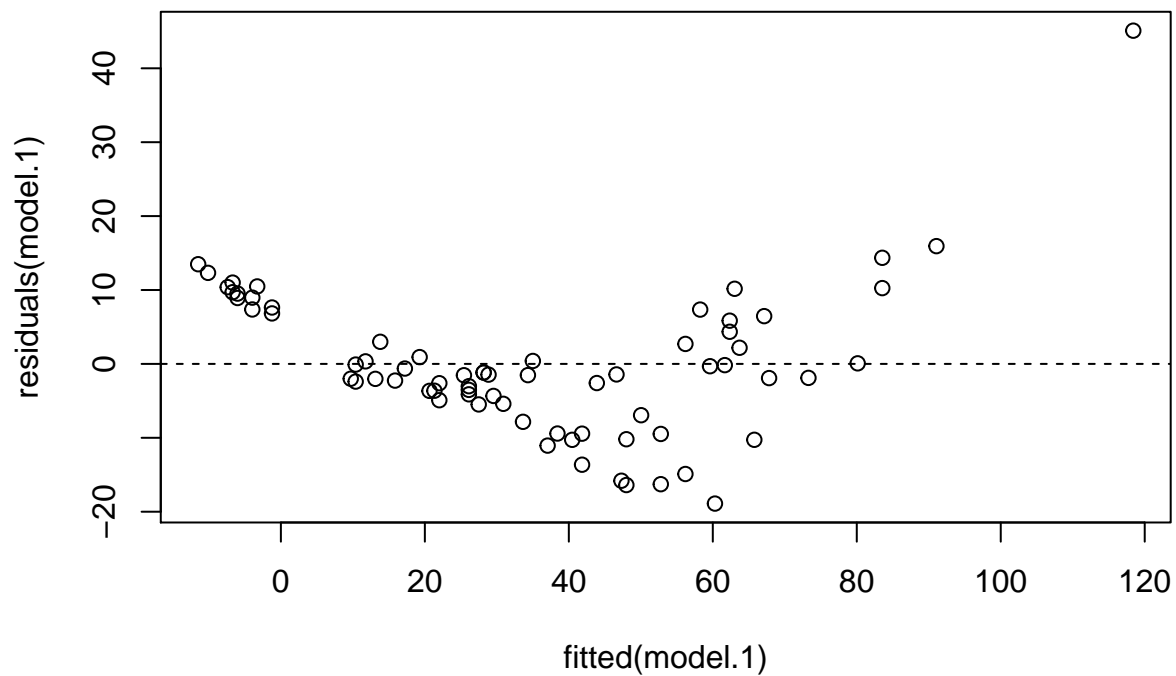
```
## lm(formula = Vol ~ Diam)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.899  -4.768  -1.438   6.740  45.089
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -41.5681     3.4269  -12.13  <2e-16 ***
## Diam         6.8367     0.2877   23.77  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.875 on 68 degrees of freedom
## Multiple R-squared:  0.8926, Adjusted R-squared:  0.891
## F-statistic: 564.9 on 1 and 68 DF,  p-value: < 2.2e-16
```

```
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept) -41.5681     3.4269  -12.13  <2e-16 ***
# Diam         6.8367     0.2877   23.77  <2e-16 ***
# ---
# Multiple R-squared:  0.8926, Adjusted R-squared:  0.891
```

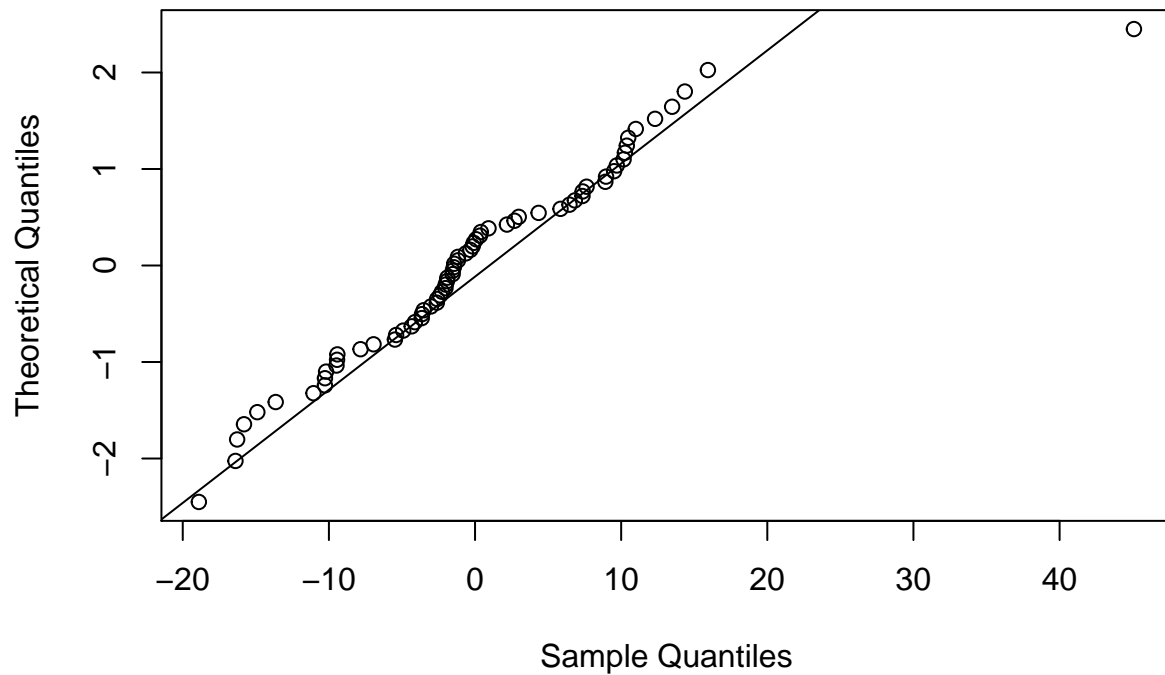
```
plot(x=Diam, y=Vol,
      panel.last = lines(sort(Diam), fitted(model.1)[order(Diam)]))
```



```
plot(x=fitted(model.1), y=residuals(model.1),
      panel.last = abline(h=0, lty=2))
```



```
qqnorm(residuals(model.1), main="", datax=TRUE)
qqline(residuals(model.1), datax=TRUE)
```



```
ad.test(residuals(model.1)) # A = 0.9913, p-value = 0.01215
```

```
##
## Anderson-Darling normality test
##
## data: residuals(model.1)
## A = 0.99132, p-value = 0.01215
```

```

lnDiam <- log(Diam)

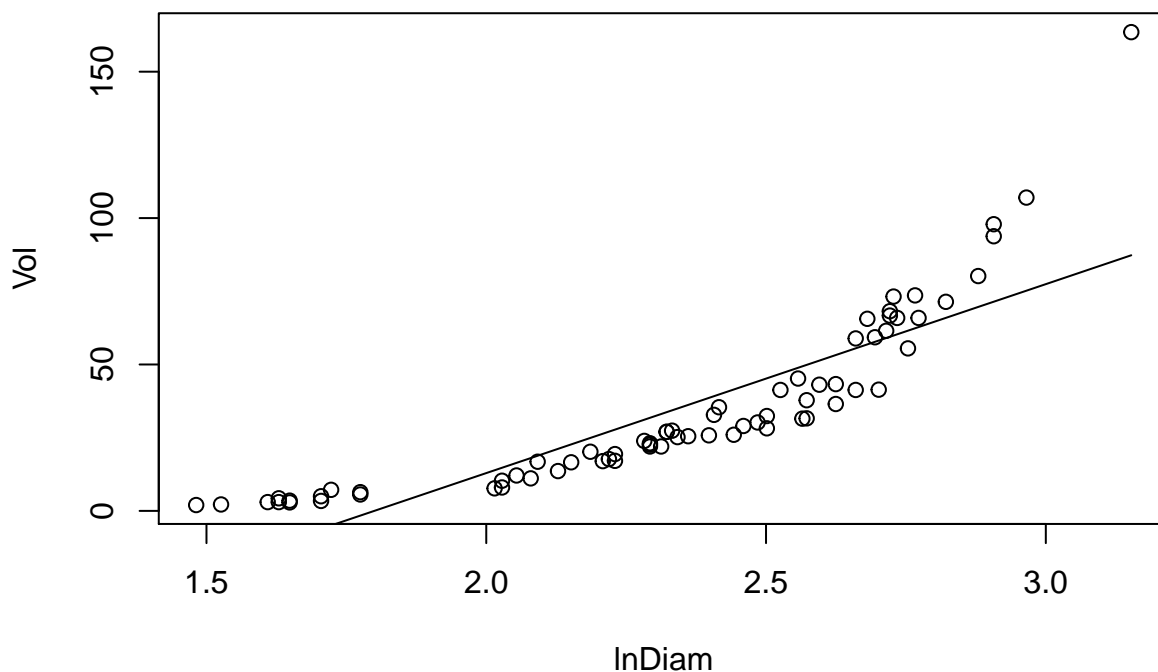
model.2 <- lm(Vol ~ lnDiam)
summary(model.2)

##
## Call:
## lm(formula = Vol ~ lnDiam)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.264  -9.665  -5.793   8.741  76.198
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -116.162     10.830  -10.73 2.88e-16 ***
## lnDiam         64.536       4.562   14.15 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.17 on 68 degrees of freedom
## Multiple R-squared:  0.7464, Adjusted R-squared:  0.7427
## F-statistic: 200.2 on 1 and 68 DF,  p-value: < 2.2e-16

#              Estimate Std. Error t value Pr(>|t|)
# (Intercept) -116.162     10.830  -10.73 2.88e-16 ***
# lnDiam       64.536       4.562   14.15 < 2e-16 ***
# Multiple R-squared:  0.7464, Adjusted R-squared:  0.7427

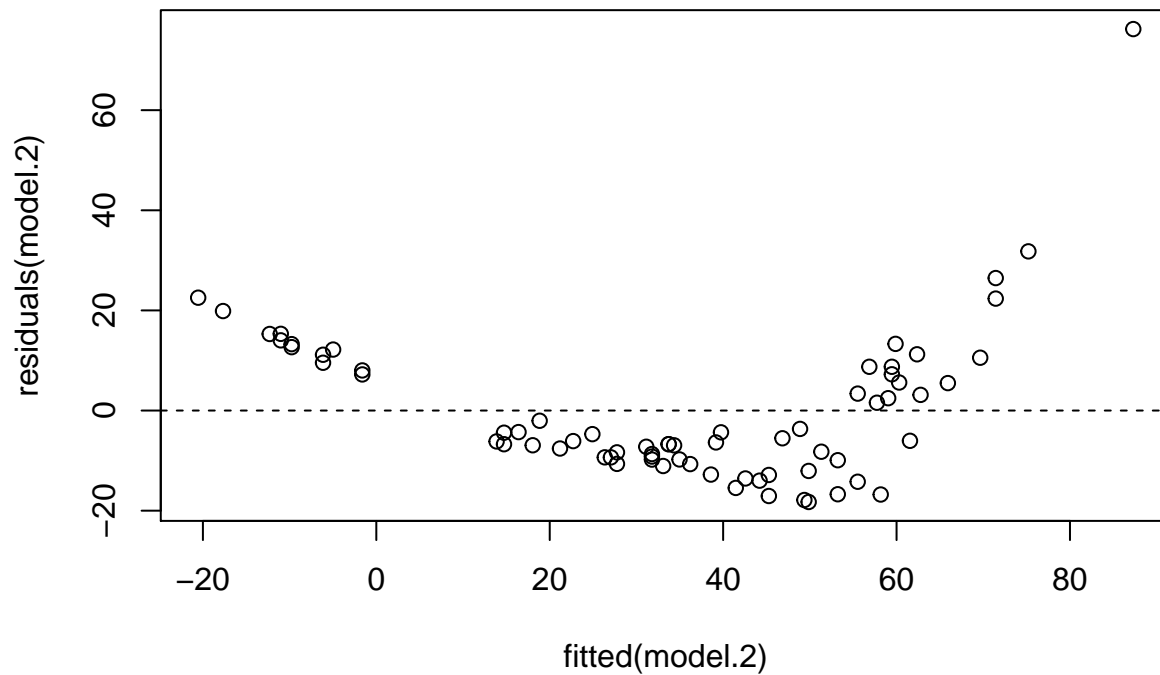
plot(x=lnDiam, y=Vol,
     panel.last = lines(sort(lnDiam), fitted(model.2)[order(lnDiam)]))

```

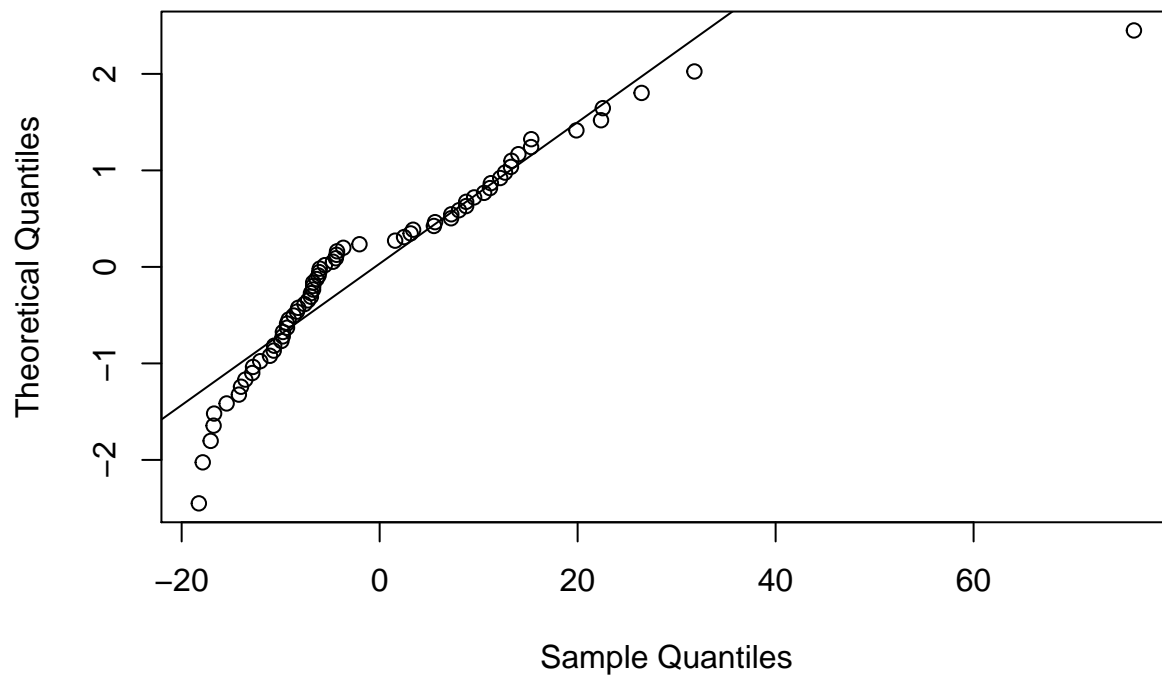




```
plot(x=fitted(model.2), y=residuals(model.2),
     panel.last = abline(h=0, lty=2))
```



```
qqnorm(residuals(model.2), main="", datax=TRUE)
qqline(residuals(model.2), datax=TRUE)
```



```
ad.test(residuals(model.2)) # A = 2.3845, p-value = 4.273e-06
```

```
##
## Anderson-Darling normality test
##
```

```
## data: residuals(model.2)
## A = 2.3845, p-value = 4.273e-06

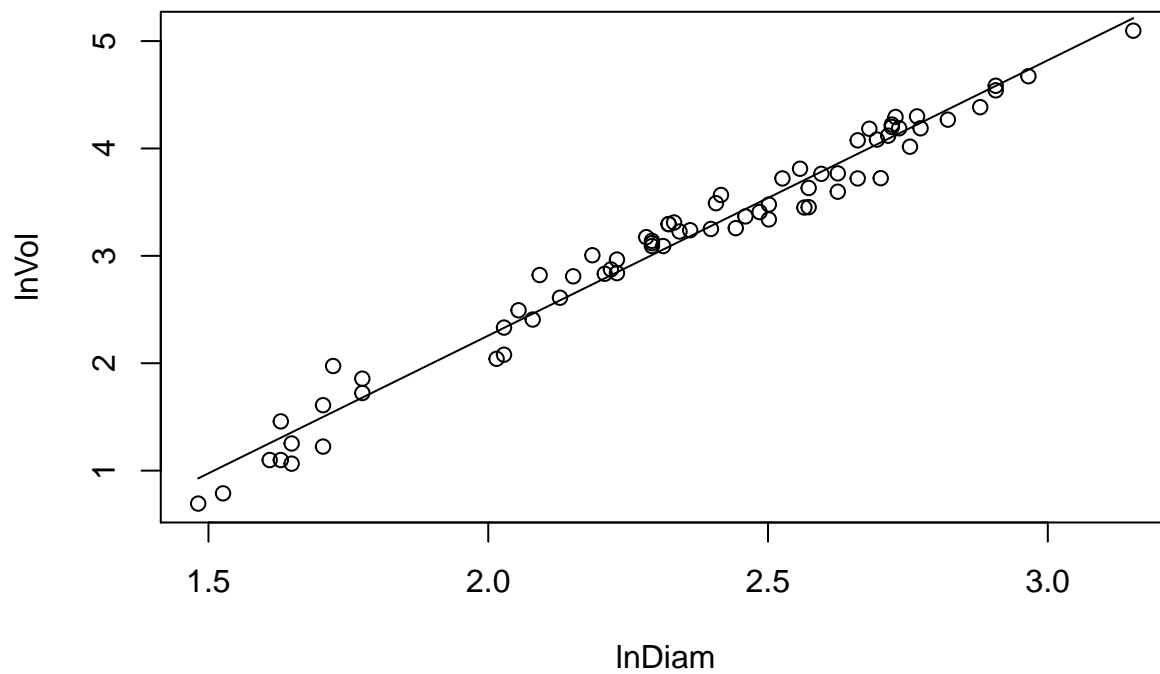
lnVol <- log(Vol)

model.3 <- lm(lnVol ~ lnDiam)
summary(model.3)

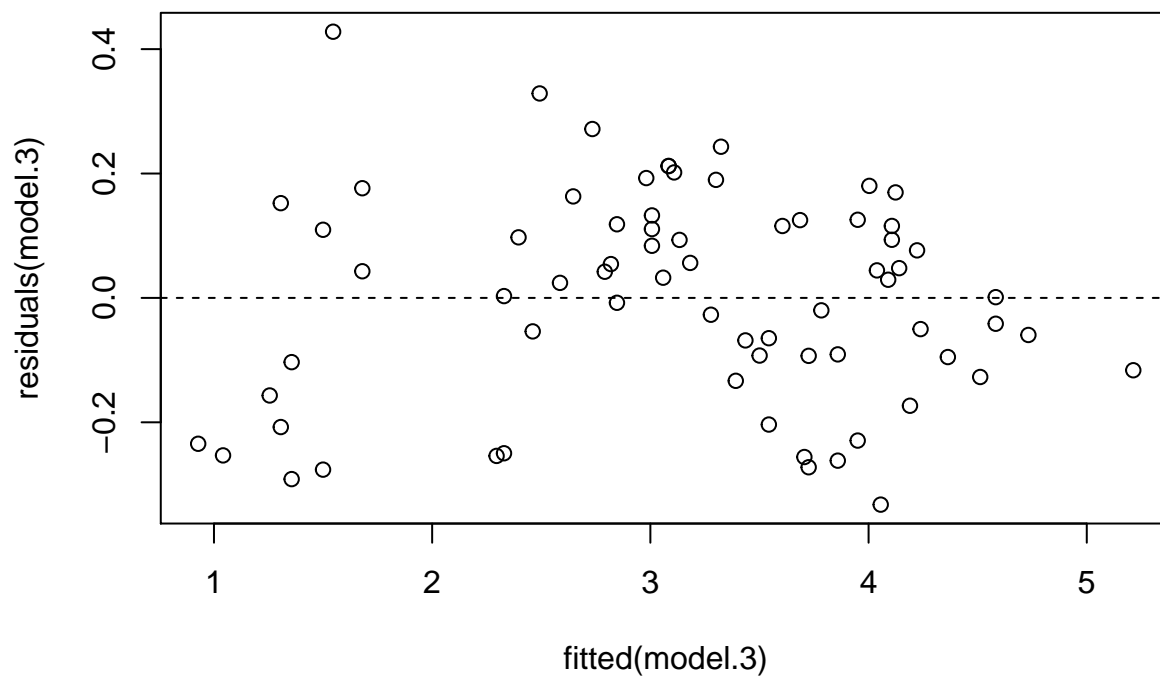
##
## Call:
## lm(formula = lnVol ~ lnDiam)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.3323 -0.1131  0.0267  0.1177  0.4280
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.8718     0.1216  -23.63  <2e-16 ***
## lnDiam         2.5644     0.0512   50.09  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1703 on 68 degrees of freedom
## Multiple R-squared:  0.9736, Adjusted R-squared:  0.9732
## F-statistic: 2509 on 1 and 68 DF, p-value: < 2.2e-16

#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -2.8718     0.1216  -23.63  <2e-16 ***
# lnDiam         2.5644     0.0512   50.09  <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.1703 on 68 degrees of freedom
# Multiple R-squared:  0.9736, Adjusted R-squared:  0.9732
# F-statistic: 2509 on 1 and 68 DF, p-value: < 2.2e-16

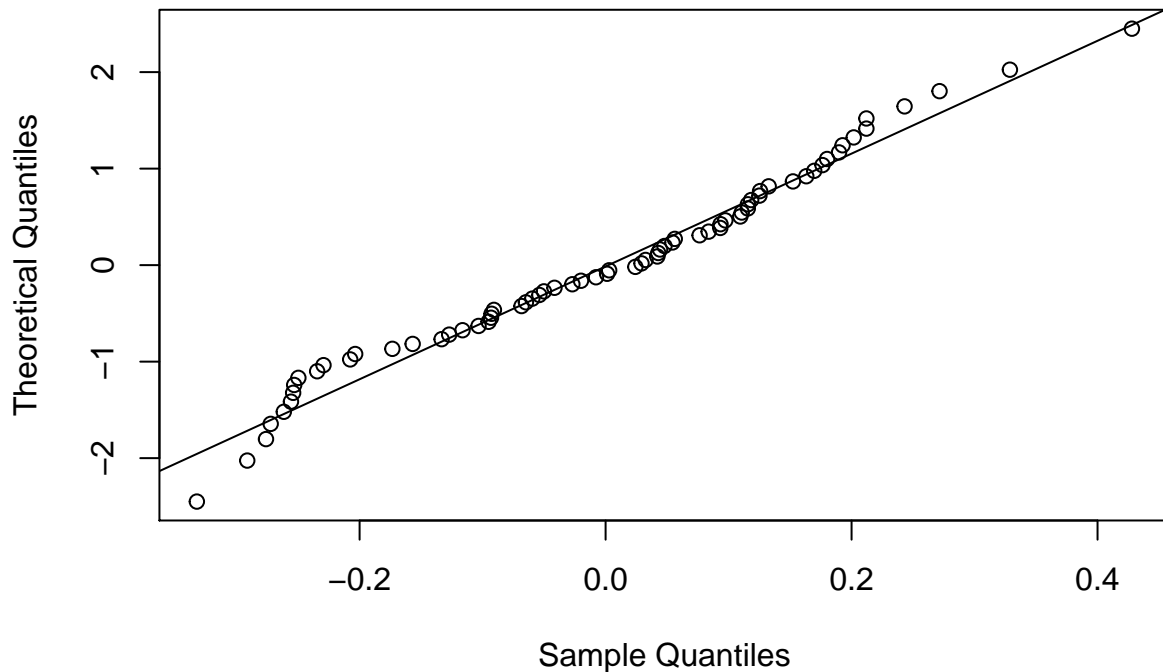
plot(x=lnDiam, y=lnVol,
     panel.last = lines(sort(lnDiam), fitted(model.3)[order(lnDiam)]))
```



```
plot(x=fitted(model.3), y=residuals(model.3),
     panel.last = abline(h=0, lty=2))
```



```
qqnorm(residuals(model.3), main="", datax=TRUE)
qqline(residuals(model.3), datax=TRUE)
```



```
ad.test(residuals(model.3)) # A = 0.5309, p-value = 0.1692
```

```
##
## Anderson-Darling normality test
##
## data: residuals(model.3)
## A = 0.53086, p-value = 0.1692
```

```
exp(predict(model.3, interval="confidence",
            newdata=data.frame(lnDiam=log(10))))
```

```
##      fit      lwr      upr
## 1 20.75934 19.92952 21.62372
```

```
#      fit      lwr      upr
# 1 20.75934 19.92952 21.62372
```

```
# proportional change in median Vol for 2-fold increase in Diam
2^(coefficients(model.3)[2]) # 5.915155
```

```
## lnDiam
## 5.915155
```

```
2^(confint(model.3)[2,]) # 95% CI
```

```
##      2.5 %      97.5 %
## 5.510776 6.349207
```

```
#      2.5 %      97.5 %
# 5.510776 6.349207
```

```
detach(shortleaf)
```

## Underground air quality (interactions)

Load the swallows data. Load the car package to access 3D scatterplots. Create interaction variables and fit a multiple linear regression model of Vent on O2 + CO2 + Type + TypeO2 + TypeCO2 + CO2O2. Use anova function to display anova table with sequential (type I) sums of squares. Calculate partial F-statistic and p-value. Display residual plot with fitted (predicted) values on the horizontal axis. Fit a multiple linear regression model of Vent on O2 + CO2 + Type. Display residual plot with fitted (predicted) values on the horizontal axis. Display normal probability plot of the residuals and add a diagonal line to the plot. Apply the Anderson-Darling normality test using the nortest package.

```
swallows <- read.table("./Data/allswallows.txt", header=T)
attach(swallows)

## The following object is masked from package:datasets:
##
##      CO2

library(car)

## Loading required package: carData
scatter3d(Vent ~ O2 + CO2, subset=Type==1) # adult

## Loading required namespace: rgl
## Loading required namespace: mgcv
scatter3d(Vent ~ O2 + CO2, subset=Type==0) # nestling
#scatter3d(Vent ~ O2 + CO2, subset=Type==0, revolutions=3, speed=0.5, grid=F)

TypeO2 <- Type*O2
TypeCO2 <- Type*CO2
CO2O2 <- CO2*O2

model.1 <- lm(Vent ~ O2 + CO2 + Type + TypeO2 + TypeCO2 + CO2O2)
summary(model.1)

##
## Call:
## lm(formula = Vent ~ O2 + CO2 + Type + TypeO2 + TypeCO2 + CO2O2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -379.89 -104.91   -7.99  116.71  442.00
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -18.399    160.007  -0.115   0.9086
## O2              1.189     9.854   0.121   0.9041
## CO2            54.281    25.987   2.089   0.0378 *
## Type          111.658    157.742   0.708   0.4797
## TypeO2         -7.008     9.560  -0.733   0.4642
## TypeCO2         2.311     7.126   0.324   0.7460
## CO2O2          -1.449     1.593  -0.909   0.3642
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 165.6 on 233 degrees of freedom
```

```
## Multiple R-squared:  0.272, Adjusted R-squared:  0.2533
## F-statistic: 14.51 on 6 and 233 DF,  p-value: 4.642e-14
```

```
#           Estimate Std. Error t value Pr(>|t|)
# (Intercept) -18.399    160.007  -0.115   0.9086
# O2           1.189     9.854    0.121   0.9041
# CO2          54.281    25.987    2.089   0.0378 *
# Type        111.658    157.742    0.708   0.4797
# TypeO2       -7.008     9.560   -0.733   0.4642
# TypeCO2       2.311     7.126    0.324   0.7460
# CO2O2        -1.449     1.593   -0.909   0.3642
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 165.6 on 233 degrees of freedom
# Multiple R-squared:  0.272, Adjusted R-squared:  0.2533
# F-statistic: 14.51 on 6 and 233 DF,  p-value: 4.642e-14

anova(model.1) # Sequential (type I) SS
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: Vent
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## O2          1  93651   93651   3.4156 0.06585 .
## CO2         1 2247696 2247696 81.9762 < 2e-16 ***
## Type        1   5910    5910   0.2156 0.64288
## TypeO2       1  14735   14735   0.5374 0.46425
## TypeCO2      1   2884    2884   0.1052 0.74598
## CO2O2        1  22664   22664   0.8266 0.36421
## Residuals 233 6388603   27419
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#           Df Sum Sq Mean Sq F value Pr(>F)
# O2          1  93651   93651   3.4156 0.06585 .
# CO2         1 2247696 2247696 81.9762 < 2e-16 ***
# Type        1   5910    5910   0.2156 0.64288
# TypeO2       1  14735   14735   0.5374 0.46425
# TypeCO2      1   2884    2884   0.1052 0.74598
# CO2O2        1  22664   22664   0.8266 0.36421
# Residuals 233 6388603   27419
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

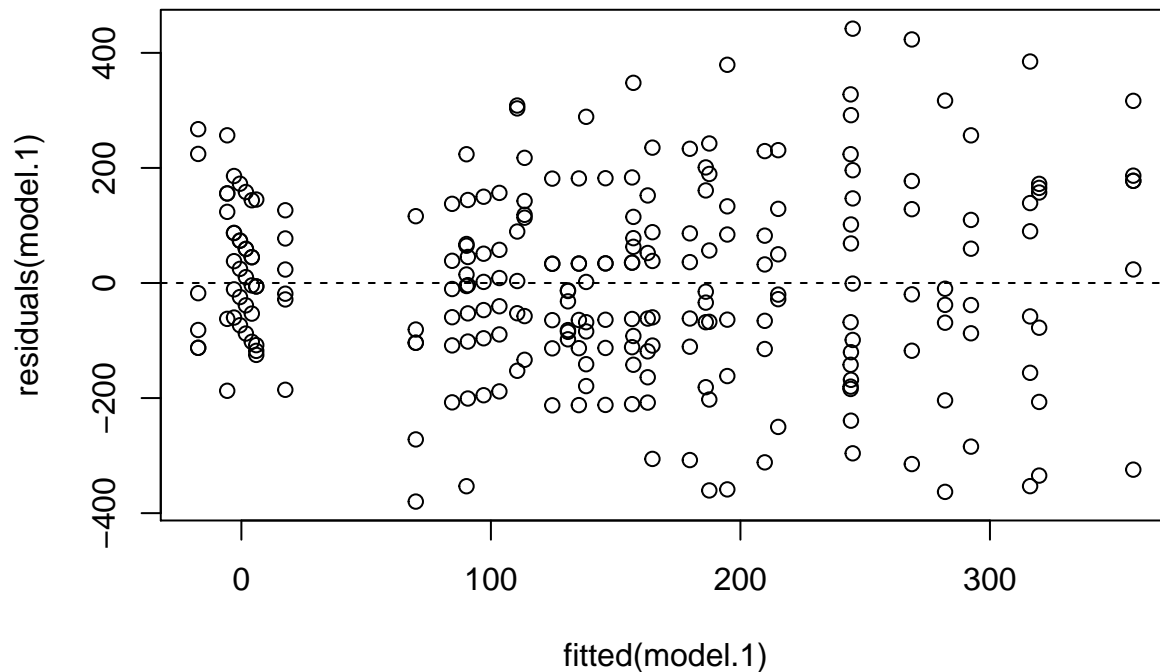
```
((14735+2884+22664)/3)/27419 # F-stat = 0.4897212
```

```
## [1] 0.4897212
```

```
pf(0.49, 3, 233, lower.tail=F) # p-value = 0.6895548
```

```
## [1] 0.6895548
```

```
plot(x=fitted(model.1), y=residuals(model.1),
     panel.last = abline(h=0, lty=2))
```



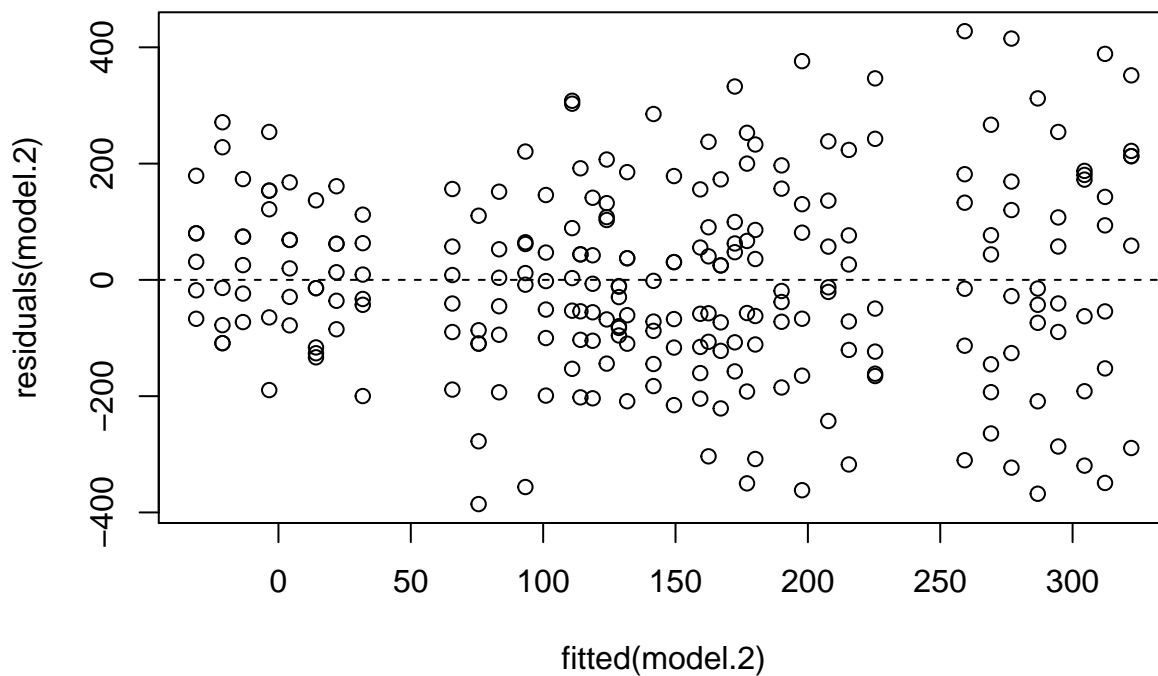
```
model.2 <- lm(Vent ~ O2 + CO2 + Type)
summary(model.2)
```

```
##
## Call:
## lm(formula = Vent ~ O2 + CO2 + Type)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -385.62 -107.76  -12.18   108.59   427.76
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  136.768    79.334   1.724   0.086 .
## O2           -8.834     4.765  -1.854   0.065 .
## CO2          32.258     3.551   9.084 <2e-16 ***
## Type          9.925    21.308   0.466   0.642
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 165 on 236 degrees of freedom
## Multiple R-squared:  0.2675, Adjusted R-squared:  0.2581
## F-statistic: 28.72 on 3 and 236 DF,  p-value: 7.219e-16
```

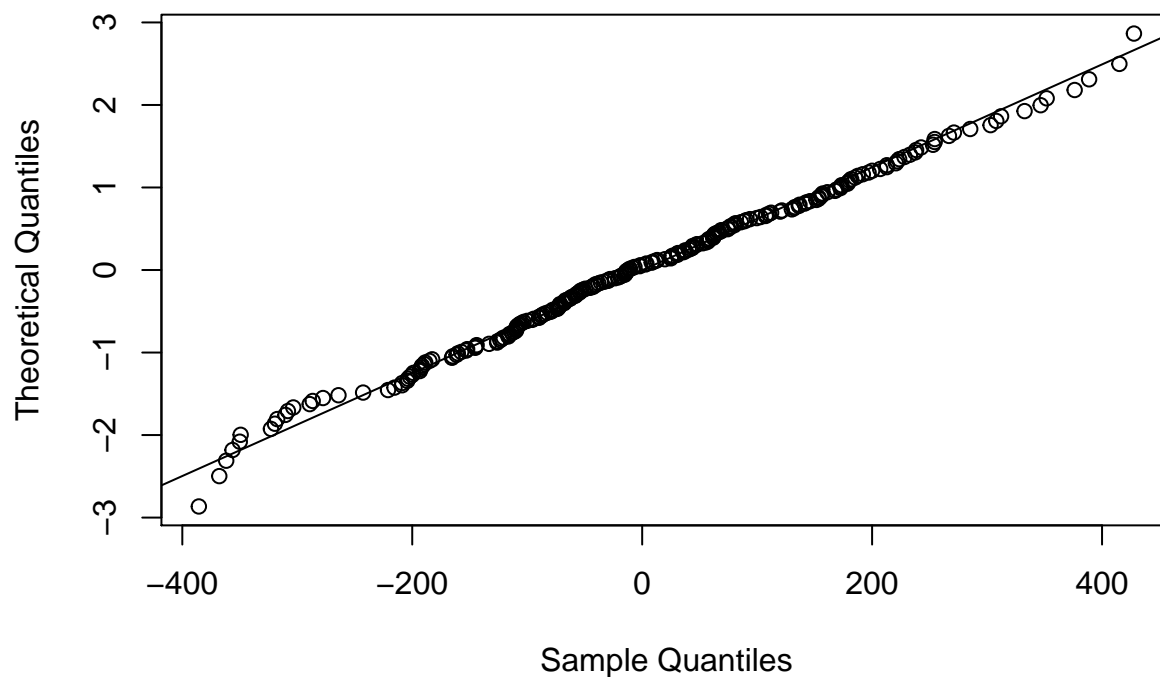
```
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  136.767    79.334   1.724   0.086 .
# O2           -8.834     4.765  -1.854   0.065 .
# CO2          32.258     3.551   9.084 <2e-16 ***
# Type          9.925    21.308   0.466   0.642
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 165 on 236 degrees of freedom
```

```
# Multiple R-squared:  0.2675, Adjusted R-squared:  0.2581
# F-statistic: 28.72 on 3 and 236 DF,  p-value: 7.219e-16
```

```
plot(x=fitted(model.2), y=residuals(model.2),
     panel.last = abline(h=0, lty=2))
```



```
qqnorm(residuals(model.2), main="", datax=TRUE)
qqline(residuals(model.2), datax=TRUE)
```



```
ad.test(residuals(model.2)) # A = 0.3175, p-value = 0.5358
```



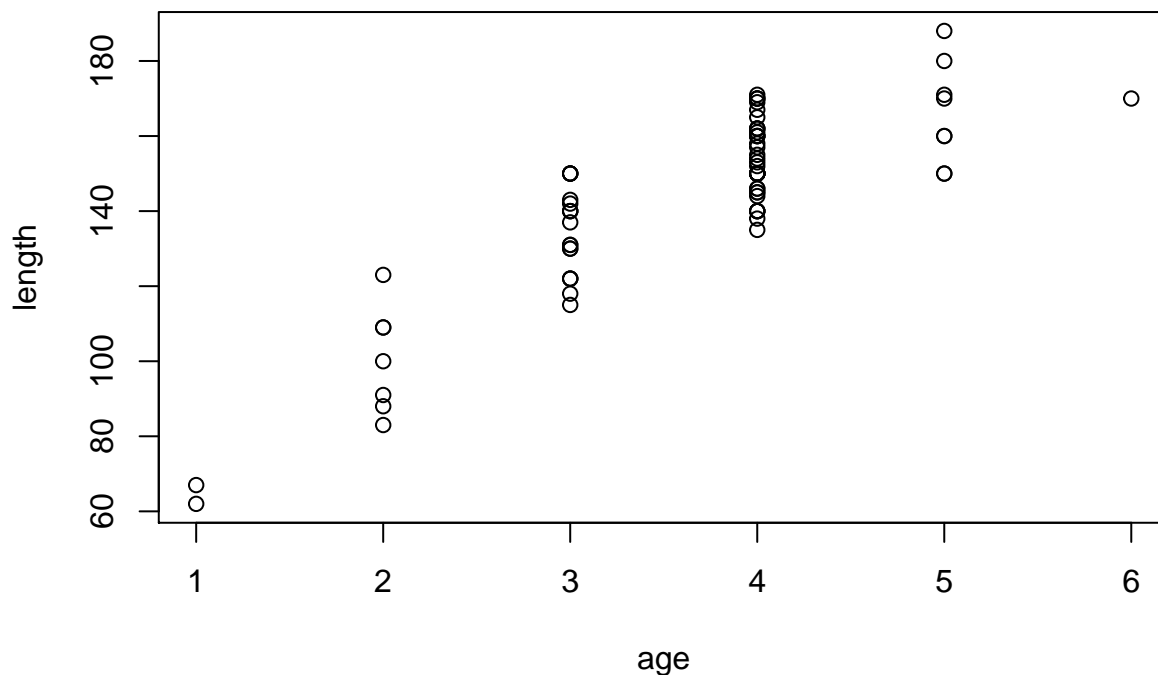
```
##
## Anderson-Darling normality test
##
## data: residuals(model.2)
## A = 0.31754, p-value = 0.5358
detach(swallows)
```

## Bluegill fish (polynomial regression)

Load the bluegills data. Display a scatterplot of the data. Create age-squared variable and fit a multiple linear regression model of length on age + agesq. Add quadratic regression line to the scatterplot. Find 95% prediction interval for length for an age of 5.

```
bluegills <- read.table("./Data/bluegills.txt", header=T)
attach(bluegills)

plot(x=age, y=length)
```



```
agesq <- age^2

model <- lm(length ~ age + agesq)
summary(model)

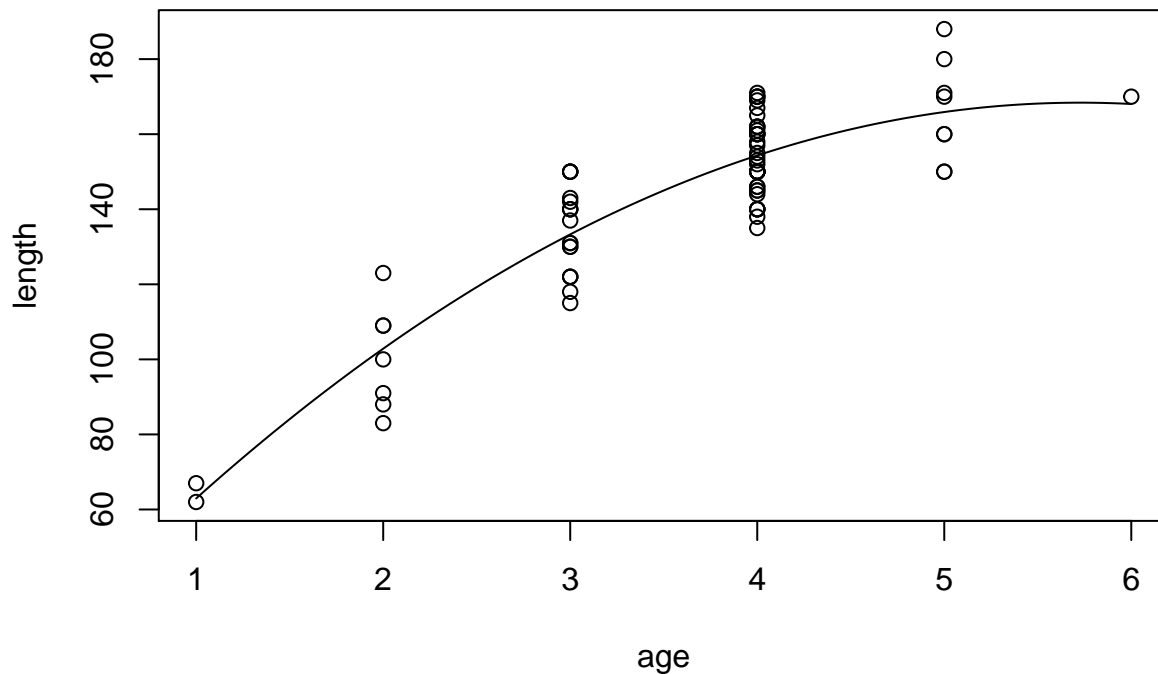
##
## Call:
## lm(formula = length ~ age + agesq)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.846  -8.321  -1.137   6.698  22.098
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 13.622 11.016 1.237 0.22
## age 54.049 6.489 8.330 2.81e-12 ***
## agesq -4.719 0.944 -4.999 3.67e-06 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.91 on 75 degrees of freedom
## Multiple R-squared: 0.8011, Adjusted R-squared: 0.7958
## F-statistic: 151.1 on 2 and 75 DF, p-value: < 2.2e-16
```

```
# Estimate Std. Error t value Pr(>|t|)
# (Intercept) 13.622 11.016 1.237 0.22
# age 54.049 6.489 8.330 2.81e-12 ***
# agesq -4.719 0.944 -4.999 3.67e-06 ***
# ---
# Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 10.91 on 75 degrees of freedom
# Multiple R-squared: 0.8011, Adjusted R-squared: 0.7958
# F-statistic: 151.1 on 2 and 75 DF, p-value: < 2.2e-16
```

```
newX <- seq(min(age), max(age), length=100)
newXsq <- newX**2

plot(x=age, y=length,
     panel.last = lines(newX,
                        predict(model,
                               newdata=data.frame(age=newX, agesq=newXsq))))
```



```
predict(model, interval="prediction",
        newdata=data.frame(age=5, agesq=25))
```

```
## fit lwr upr
```

```
## 1 165.9023 143.487 188.3177
```

```
#      fit      lwr      upr  
# 1 165.9023 143.487 188.3177
```

```
detach(bluegills)
```

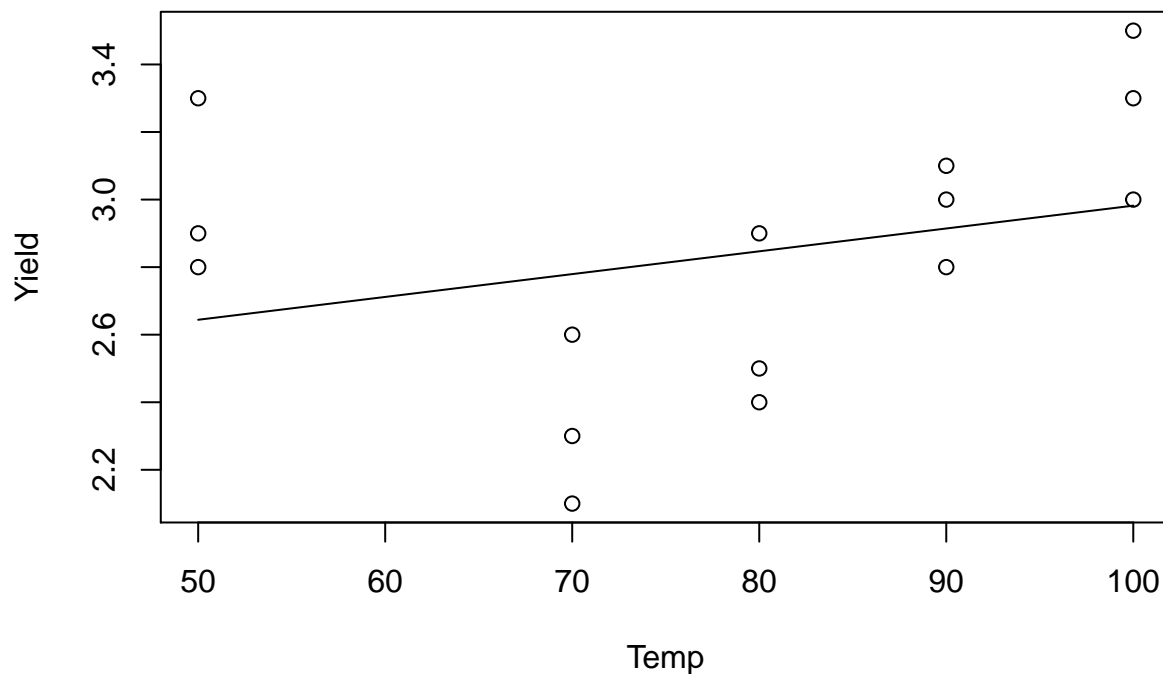
## Experiment yield (polynomial regression)

Load the yield data. Fit a simple linear regression model of Yield on Temp. Display a scatterplot of the data and add the simple linear regression line. Create Temp-squared variable and fit a multiple linear regression model of Yield on Temp + Tempsq. Display a scatterplot of the data and add quadratic regression line to the scatterplot.

```
yield <- read.table("./Data/yield.txt", header=T)  
attach(yield)
```

```
model.1 <- lm(Yield ~ Temp)  
summary(model.1)
```

```
##  
## Call:  
## lm(formula = Yield ~ Temp)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -0.67928 -0.26306  0.05315  0.22072  0.65586   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept)  2.306306   0.469075   4.917 0.000282 ***  
## Temp         0.006757   0.005873   1.151 0.270641      
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.3913 on 13 degrees of freedom  
## Multiple R-squared:  0.09242,    Adjusted R-squared:  0.0226   
## F-statistic: 1.324 on 1 and 13 DF,  p-value: 0.2706  
  
#      Estimate Std. Error t value Pr(>|t|)      
# (Intercept)  2.306306   0.469075   4.917 0.000282 ***  
# Temp         0.006757   0.005873   1.151 0.270641      
# ---  
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
#  
# Residual standard error: 0.3913 on 13 degrees of freedom  
# Multiple R-squared:  0.09242,    Adjusted R-squared:  0.0226   
# F-statistic: 1.324 on 1 and 13 DF,  p-value: 0.2706  
  
plot(x=Temp, y=Yield,  
      panel.last = lines(sort(Temp), fitted(model.1)[order(Temp)]))
```



```
Tempsq <- Temp^2
```

```
model.2 <- lm(Yield ~ Temp + Tempsq)
summary(model.2)
```

```
##
## Call:
## lm(formula = Yield ~ Temp + Tempsq)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37113 -0.15567 -0.04536  0.15790  0.35258
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.9604811  1.2589183   6.323 3.81e-05 ***
## Temp        -0.1537113  0.0349408  -4.399 0.000867 ***
## Tempsq       0.0010756  0.0002329   4.618 0.000592 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2444 on 12 degrees of freedom
## Multiple R-squared:  0.6732, Adjusted R-squared:  0.6187
## F-statistic: 12.36 on 2 and 12 DF, p-value: 0.001218
```

```
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  7.9604811  1.2589183   6.323 3.81e-05 ***
# Temp        -0.1537113  0.0349408  -4.399 0.000867 ***
# Tempsq       0.0010756  0.0002329   4.618 0.000592 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.2444 on 12 degrees of freedom
```

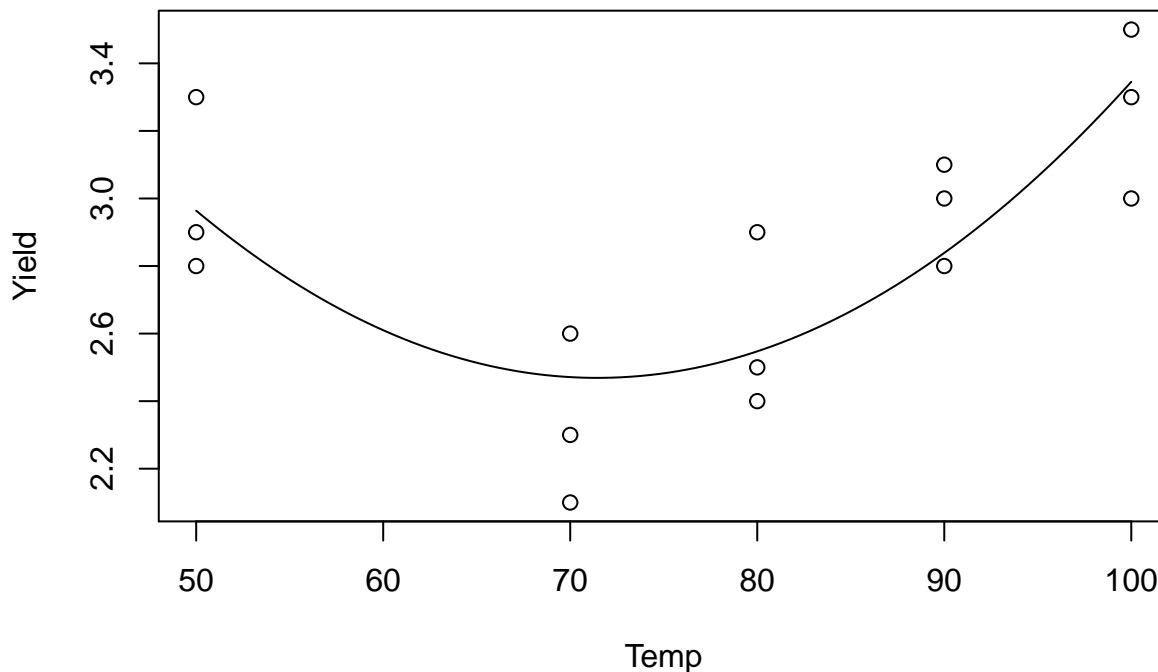
```

# Multiple R-squared:  0.6732, Adjusted R-squared:  0.6187
# F-statistic: 12.36 on 2 and 12 DF,  p-value: 0.001218

newX <- seq(min(Temp), max(Temp), length=100)
newXsq <- newX**2

plot(x=Temp, y=Yield,
     panel.last = lines(newX,
                        predict(model.2,
                                newdata=data.frame(Temp=newX, Tempsq=newXsq))))

```



```
detach(yield)
```

## Chemical odor (polynomial regression)

Load the odor data. Create squared variables and fit a multiple linear regression model of Odor on Temp + Ratio + Height + Tempsq + Ratiosq + Heightsq. Fit a multiple linear regression model of Odor on Temp + Ratio + Height + Tempsq + Ratiosq.

```

odor <- read.table("./Data/odor.txt", header=T)
attach(odor)

Tempsq <- Temp^2
Ratiosq <- Ratio^2
Heightsq <- Height^2

model.1 <- lm(Odor ~ Temp + Ratio + Height + Tempsq + Ratiosq + Heightsq)
summary(model.1)

```

```

##
## Call:
## lm(formula = Odor ~ Temp + Ratio + Height + Tempsq + Ratiosq +
##      Heightsq)

```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.625  -9.625  -1.375   4.021  28.875
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -30.667     10.840  -2.829  0.0222 *
## Temp         -12.125      6.638  -1.827  0.1052
## Ratio        -17.000      6.638  -2.561  0.0336 *
## Height       -21.375      6.638  -3.220  0.0122 *
## Tempsq        32.083      9.771   3.284  0.0111 *
## Ratiosq       47.833      9.771   4.896  0.0012 **
## Heightsq       6.083      9.771   0.623  0.5509
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.77 on 8 degrees of freedom
## Multiple R-squared:  0.8683, Adjusted R-squared:  0.7695
## F-statistic: 8.789 on 6 and 8 DF, p-value: 0.003616
```

```
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -30.667     10.840  -2.829  0.0222 *
# Temp         -12.125      6.638  -1.827  0.1052
# Ratio        -17.000      6.638  -2.561  0.0336 *
# Height       -21.375      6.638  -3.220  0.0122 *
# Tempsq        32.083      9.771   3.284  0.0111 *
# Ratiosq       47.833      9.771   4.896  0.0012 **
# Heightsq       6.083      9.771   0.623  0.5509
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 18.77 on 8 degrees of freedom
# Multiple R-squared:  0.8683, Adjusted R-squared:  0.7695
# F-statistic: 8.789 on 6 and 8 DF, p-value: 0.003616
```

```
model.2 <- lm(Odor ~ Temp + Ratio + Height + Tempsq + Ratiosq)
summary(model.2)
```

```
##
## Call:
## lm(formula = Odor ~ Temp + Ratio + Height + Tempsq + Ratiosq)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.933  -9.635  -4.067   4.620  26.933
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -26.923      8.707  -3.092 0.012884 *
## Temp         -12.125      6.408  -1.892 0.091024 .
## Ratio        -17.000      6.408  -2.653 0.026350 *
## Height       -21.375      6.408  -3.336 0.008720 **
## Tempsq        31.615      9.404   3.362 0.008366 **
## Ratiosq       47.365      9.404   5.036 0.000703 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.12 on 9 degrees of freedom
## Multiple R-squared:  0.8619, Adjusted R-squared:  0.7852
## F-statistic: 11.23 on 5 and 9 DF,  p-value: 0.001169

#           Estimate Std. Error t value Pr(>|t|)
# (Intercept) -26.923      8.707  -3.092 0.012884 *
# Temp        -12.125      6.408  -1.892 0.091024 .
# Ratio       -17.000      6.408  -2.653 0.026350 *
# Height      -21.375      6.408  -3.336 0.008720 **
# Tempsq       31.615      9.404   3.362 0.008366 **
# Ratiosq      47.365      9.404   5.036 0.000703 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 18.12 on 9 degrees of freedom
# Multiple R-squared:  0.8619, Adjusted R-squared:  0.7852
# F-statistic: 11.23 on 5 and 9 DF,  p-value: 0.001169

detach(odor)
```