Isolator data learning

Huy Pham

02 July, 2021

Data loading

Limit ground motions to only those with scale factor less than 20.

```
dataPath <- './imStudyData_manualbandpass.csv'
isol.full <- read.csv(dataPath, header=TRUE) %>%
  filter(GMScale <= 20) %>% filter(GMSTfb <= 5)
```

Organize outputs. Currently, we record maximum interstory drift and any collapse between the three levels.

```
isol.full$maxDrift <- pmax(isol.full$driftMax1, isol.full$driftMax2, isol.full$driftMax3)
isol.full$collapse <- ((isol.full$collapseDrift1 | isol.full$collapseDrift2) |
  isol.full$collapseDrift3) %>%
  as.integer()
```

Optionally, gather the dimensionless variables

```
zetaRef     <- c(0.02, 0.05, 0.10, 0.20, 0.30, 0.40, 0.50)
BmRef       <- c(0.8, 1.0, 1.2, 1.5, 1.7, 1.9, 2.0)
tmp <- unlist(approx(zetaRef, BmRef, isol.full$zetaM)[2])
isol.full$Bm  <- tmp

g <- 386.4

# nondims
isol.full$TfbRatio <- isol.full$Tfb/isol.full$Tm
isol.full$mu2Ratio <- isol.full$mu2/(isol.full$GMSTm / isol.full$Bm)
# isol.full$gapRatio <- isol.full$moatGap/(isol.full$mu2 * g * isol.full$Tm^2)
isol.full$gapRatio <- isol.full$moatGap*4*pi^2/((isol.full$GMSTm/isol.full$Bm) *
                                              g * isol.full$Tm^2)
isol.full$T2Ratio <- isol.full$T2/isol.full$Tm
isol.full$Qm <- isol.full$mu2*g
```

Collect intensity measures

Function to get design spectral acceleration

```
getDesignSa <- function(Tquery, S1) {
  Ss <- 2.2815
  Tshort <- S1/Ss
  if (Tquery < Tshort) {
    SaTquery <- S1
  } else {
    SaTquery <- S1/Tquery
```

```
  }
  return(SaTquery)
}
```

Collect the structure IMs

```
isol.full$S1Dm <- isol.full$moatGap*4*pi^2*isol.full$Bm/(g*isol.full$Tm)
isol.full$Sm <- mapply(getDesignSa, isol.full$Tm, isol.full$S1) * isol.full$Bm
# isol.full$Sfb <- mapply(getDesignSa, isol.full$Tfb, isol.full$S1)
```

Collect the ground motion IMs

Ground motion IMs are currently GMSTm, GMST2, GMSavg.

Dimensionless variables:

1. $\frac{Sa_{avg}(T)}{S_{1,amp,M}}$
2. $\frac{Sa_{avg}(T)}{S_M}$
3. $\frac{Sa(T_2,\zeta=5\%)}{S_{1,amp,M}}$
4. $\frac{Sa(T_2,\zeta=5\%)}{S_M}$
5. $\frac{Sa(T_M,\zeta=5\%)}{S_{1,amp,M}}$
6. $\frac{Sa(T_M,\zeta=5\%)}{S_M}$
7. $\frac{IP(T_M)}{S_{1,amp,M} \cdot D_M}$
8. $\frac{PGA}{S_{1,amp,M} \cdot g}$
9. $\frac{PGV}{S_{1,amp,M} \cdot T_M \cdot g}$
10. $\frac{FIV3}{S_{1,amp,M} \cdot T_M \cdot g}$
11. $\frac{IP(T_M)}{S_M \cdot D_M}$
12. $\frac{FIV3}{S_M \cdot T_M \cdot g}$

```
isol.full$Pi1 <- isol.full$GMSavg/isol.full$S1Dm
isol.full$Pi2 <- isol.full$GMSavg/isol.full$Sm
isol.full$Pi3 <- isol.full$GMST2/isol.full$S1Dm
isol.full$Pi4 <- isol.full$GMST2/isol.full$Sm
isol.full$Pi5 <- isol.full$GMSTm/isol.full$S1Dm
isol.full$Pi6 <- isol.full$GMSTm/isol.full$Sm
isol.full$Pi7 <- isol.full$IPTm/(isol.full$S1Dm*g*isol.full$moatGap)
isol.full$Pi8 <- isol.full$PGA/(isol.full$S1Dm*g)
isol.full$Pi9 <- isol.full$PGV/(isol.full$S1Dm*g*isol.full$Tm)
isol.full$Pi10 <- isol.full$FIV3Tm/(isol.full$S1Dm*g*isol.full$Tm)
isol.full$Pi11 <- isol.full$IPTm/(isol.full$Sm*g*isol.full$moatGap)
isol.full$Pi12 <- isol.full$FIV3Tm/(isol.full$Sm*g*isol.full$Tm)
```

Logit regression: Collapse

Split data into test and training set.

```r
set.seed(1)

isol.train <- isol.full %>% sample_frac(0.8)
isol.test <- isol.full %>% setdiff(isol.train)
```

Make functions for logistic regression and plotting with respect to collapse.

```r
logiStudy <- function(piVar, train, test) {
  logitCollapse <- glm(paste("collapse ~ ", piVar), family=binomial(link = "logit"),
                      data = train)
  summary(logitCollapse)
  confint(logitCollapse)
  test.prob <- logitCollapse %>% predict(test, type = "response")
  test.collapse <- ifelse(test.prob > 0.5, 1, 0)

  test.accuracy <- mean(test.collapse == test$collapse)

  return(list(classification = logitCollapse, accuracy = test.accuracy))
}

logiPlot <- function(dataSet, xvar) {
    ggplot(data = dataSet, aes_string(x = xvar, y = "collapse")) +
    geom_point(alpha = 0.2) +
    geom_smooth(method = "glm", method.args = list(family = "binomial")) +
    labs(
      title = "Logistic Regression Model",
      x = xvar,
      y = "Probability of collapse"
    )
}
```

First fit

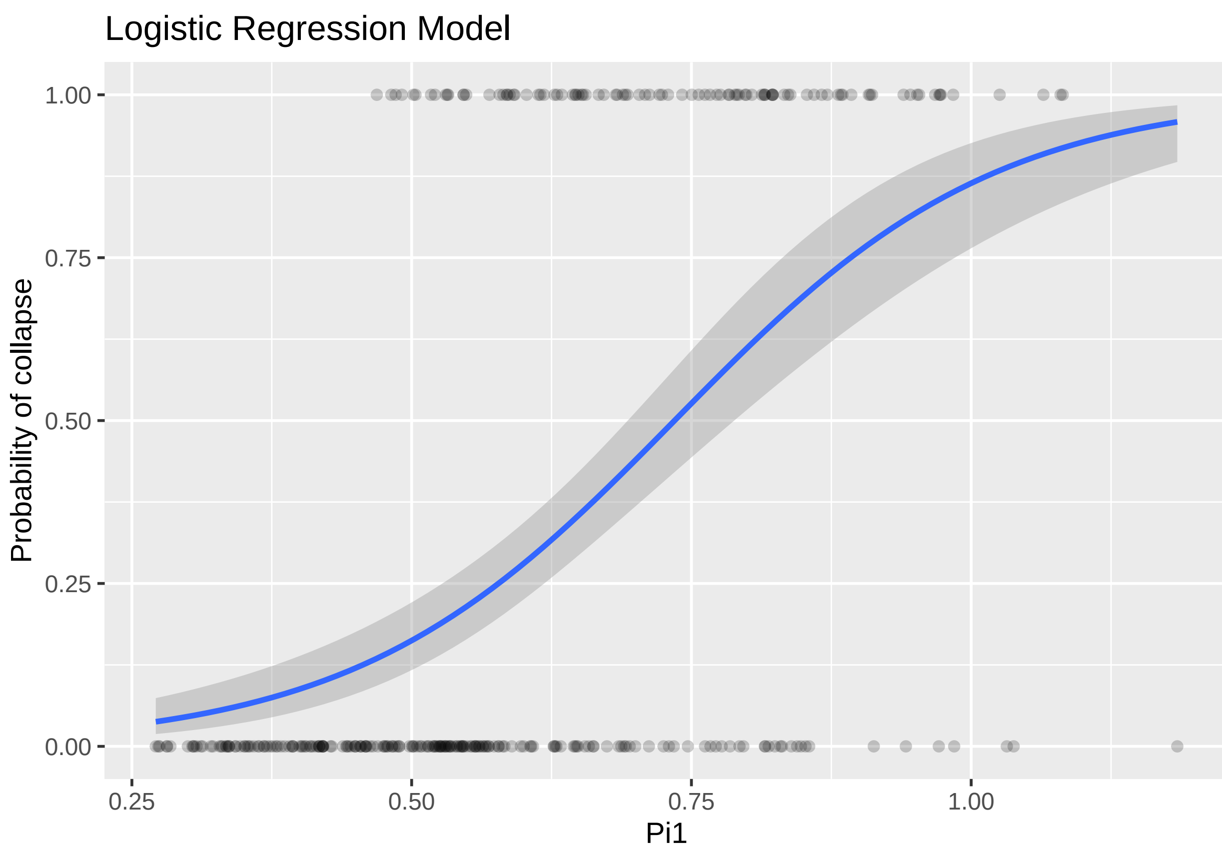
Variable: $\pi_1 = \frac{Sa_{avg}(T)}{S_{1,amp,M}}$

```r
logi1 <- logiStudy("Pi1", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```r
logiPlot(isol.train, "Pi1")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Second fit

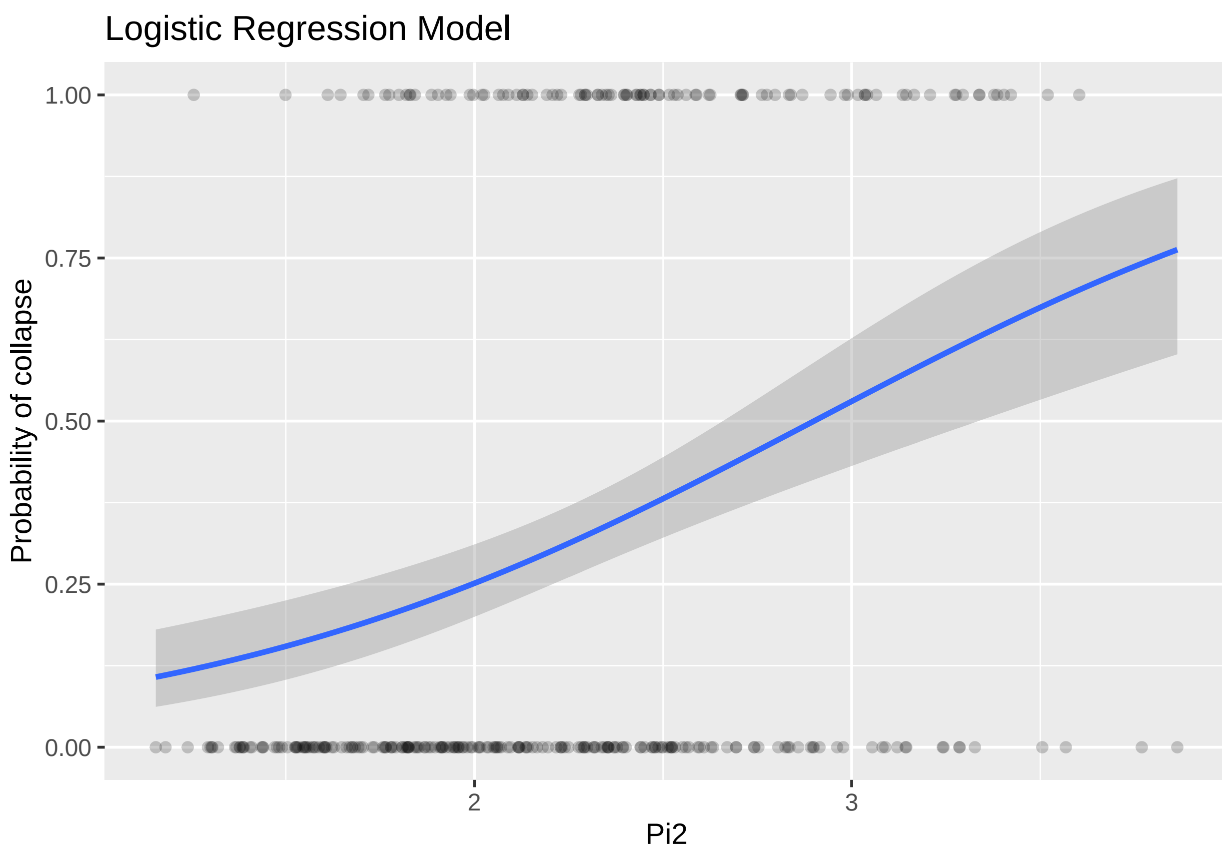
Variable: $\pi_2 = \frac{Sa_{avg}(T)}{S_M}$

```
logi2 <- logiStudy("Pi2", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```
logiPlot(isol.train, "Pi2")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Third fit

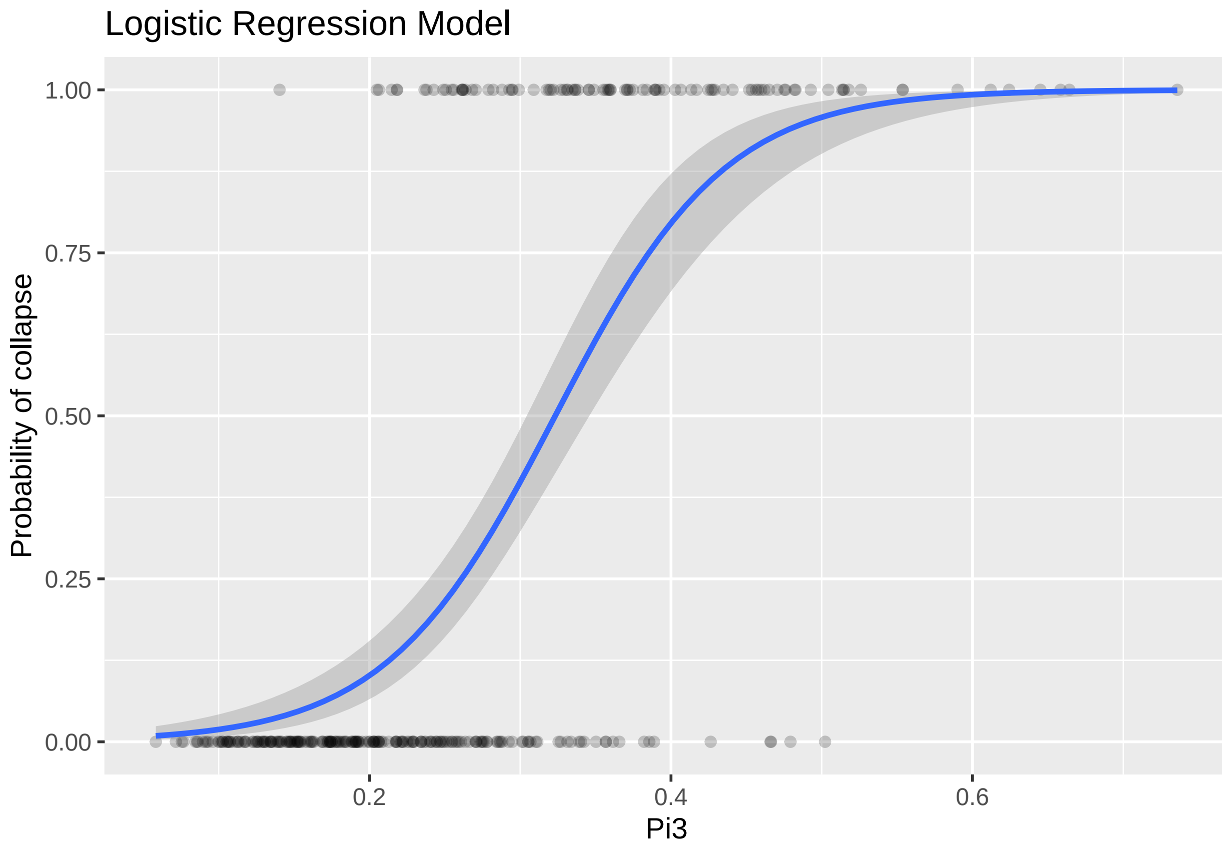
Variable: $\pi_3 = \frac{Sa(T_2)}{S_{1,amp,M}}$

```
logi3 <- logiStudy("Pi3", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```
logiPlot(isol.train, "Pi3")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Fourth fit

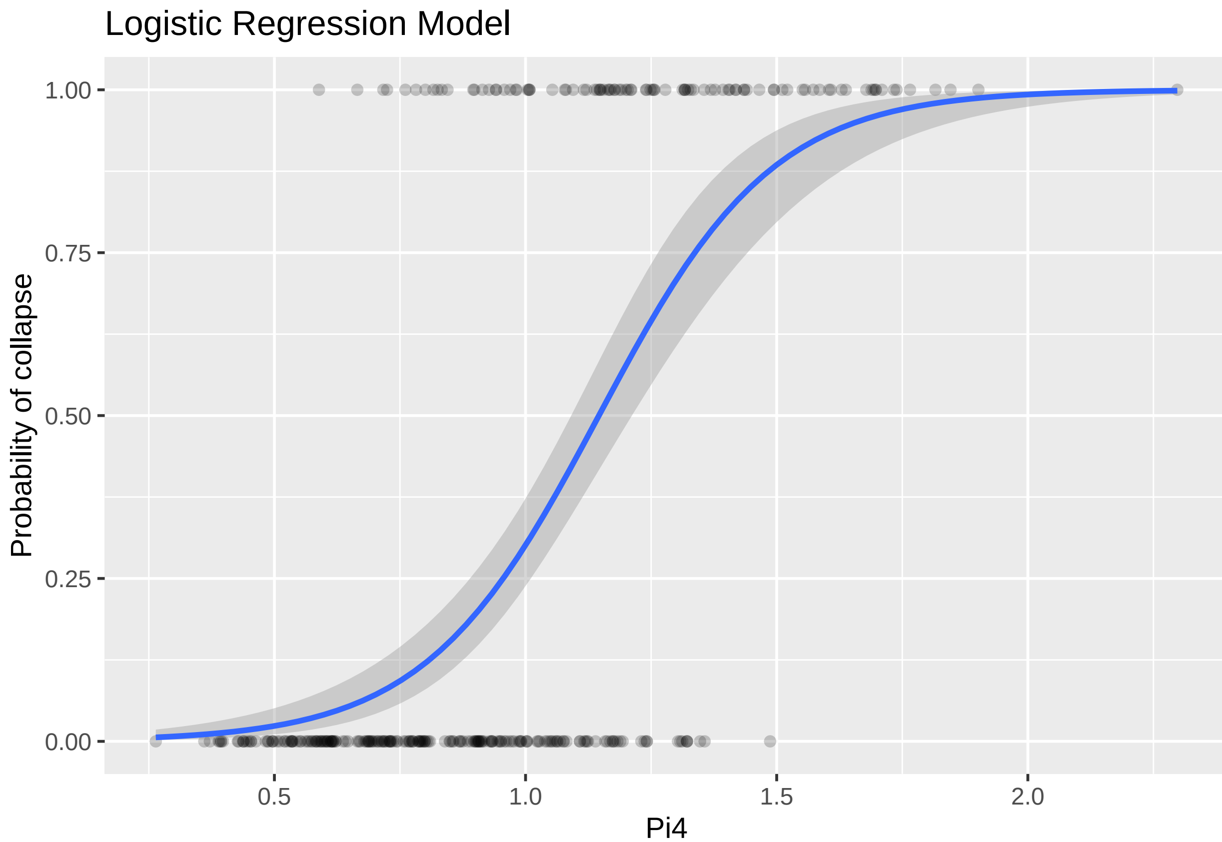
Variable: $\pi_4 = \frac{Sa(T_2)}{S_M}$

```
logi4 <- logiStudy("Pi4", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```
logiPlot(isol.train, "Pi4")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Fifth fit

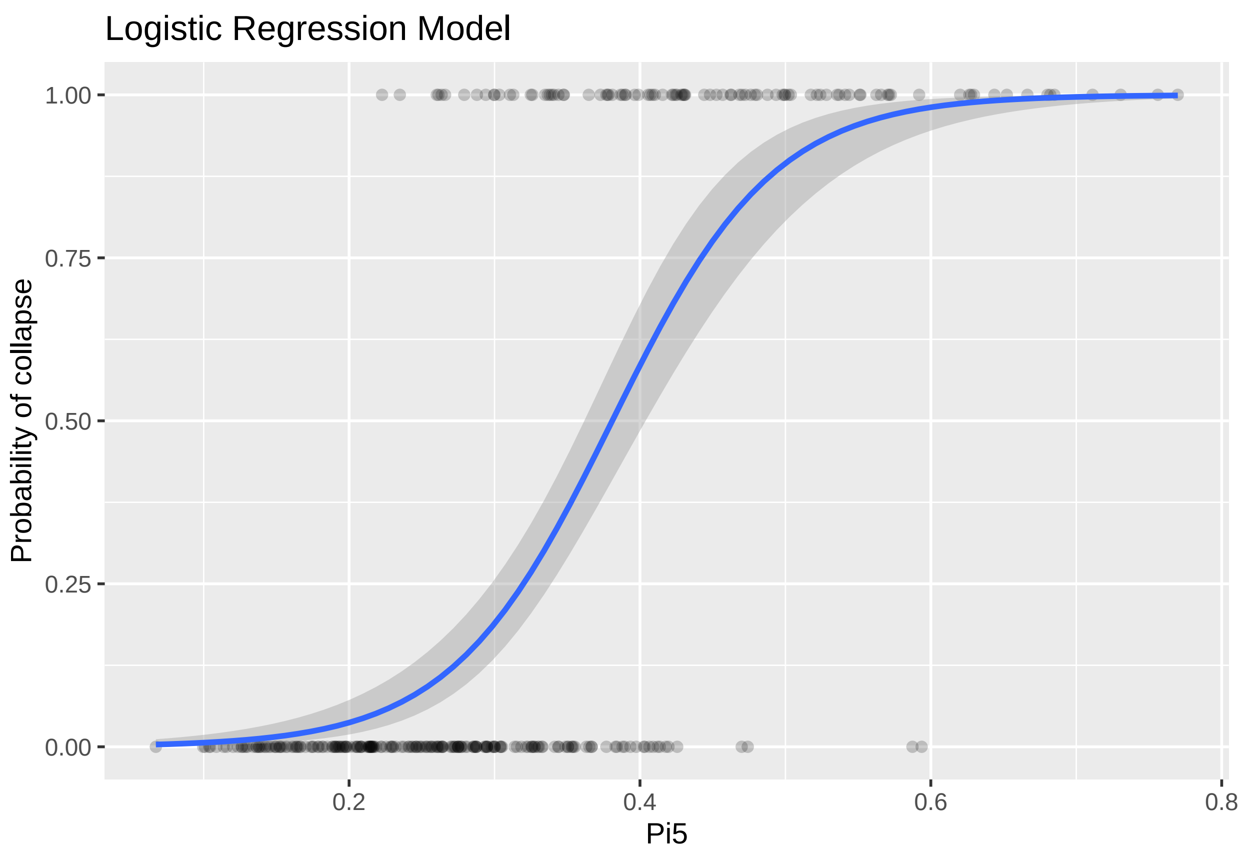
Variable: $\pi_5 = \frac{Sa(T_M)}{S_{1,amp,M}}$

```
logi5 <- logiStudy("Pi5", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```
logiPlot(isol.train, "Pi5")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Sixth fit

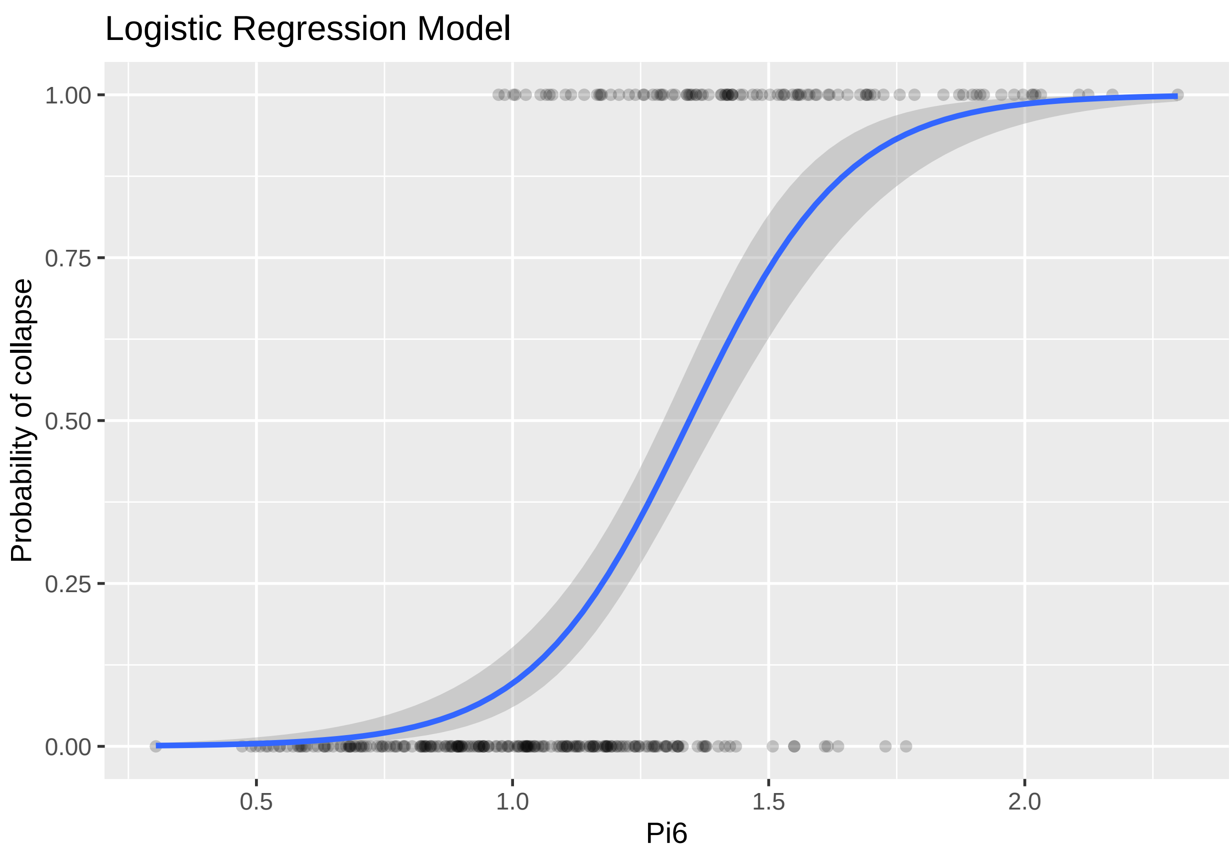
Variable: $\pi_6 = \frac{Sa(T_M)}{S_M}$

```
logi6 <- logiStudy("Pi6", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```
logiPlot(isol.train, "Pi6")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Seventh fit

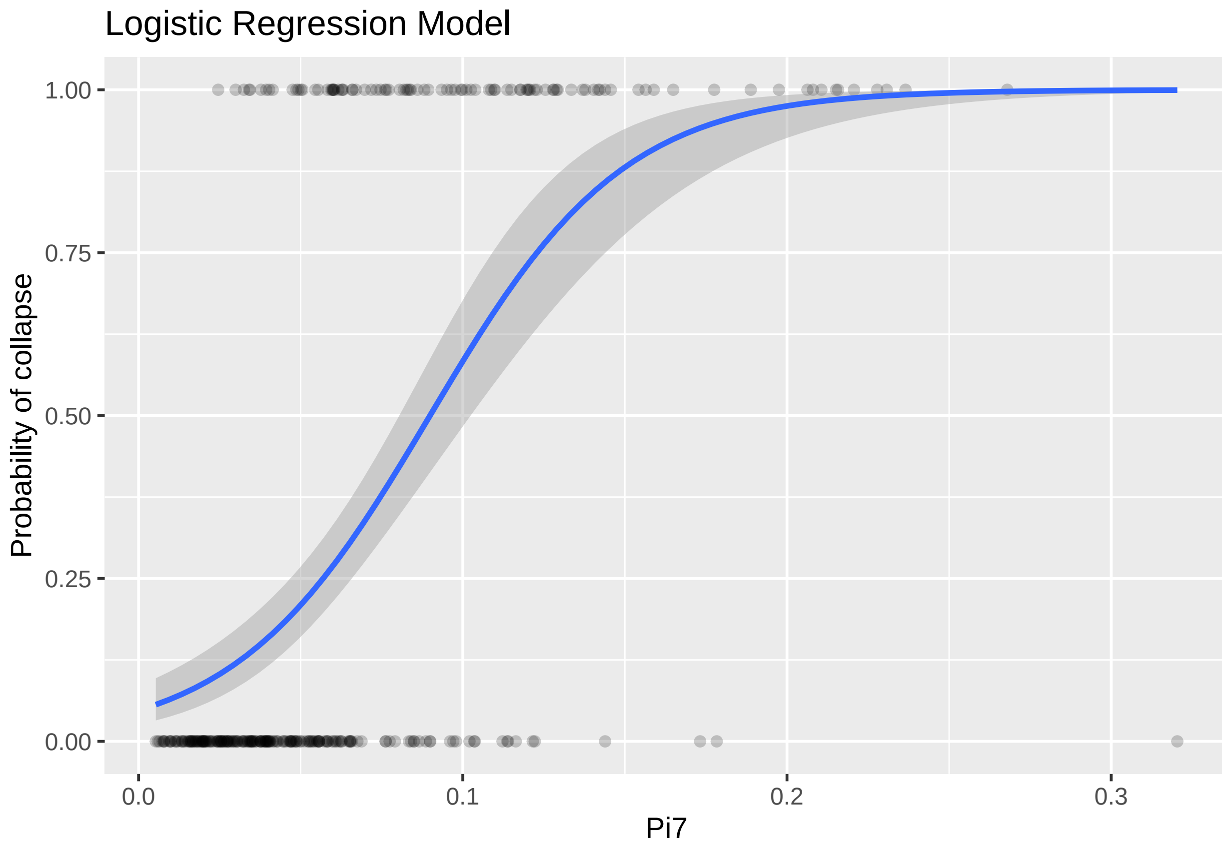
Variable: $\frac{IP(T_M)}{S_{1,amp,M} \cdot D_M}$

```r
logi7 <- logiStudy("Pi7", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```r
logiPlot(isol.train, "Pi7")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Eighth fit

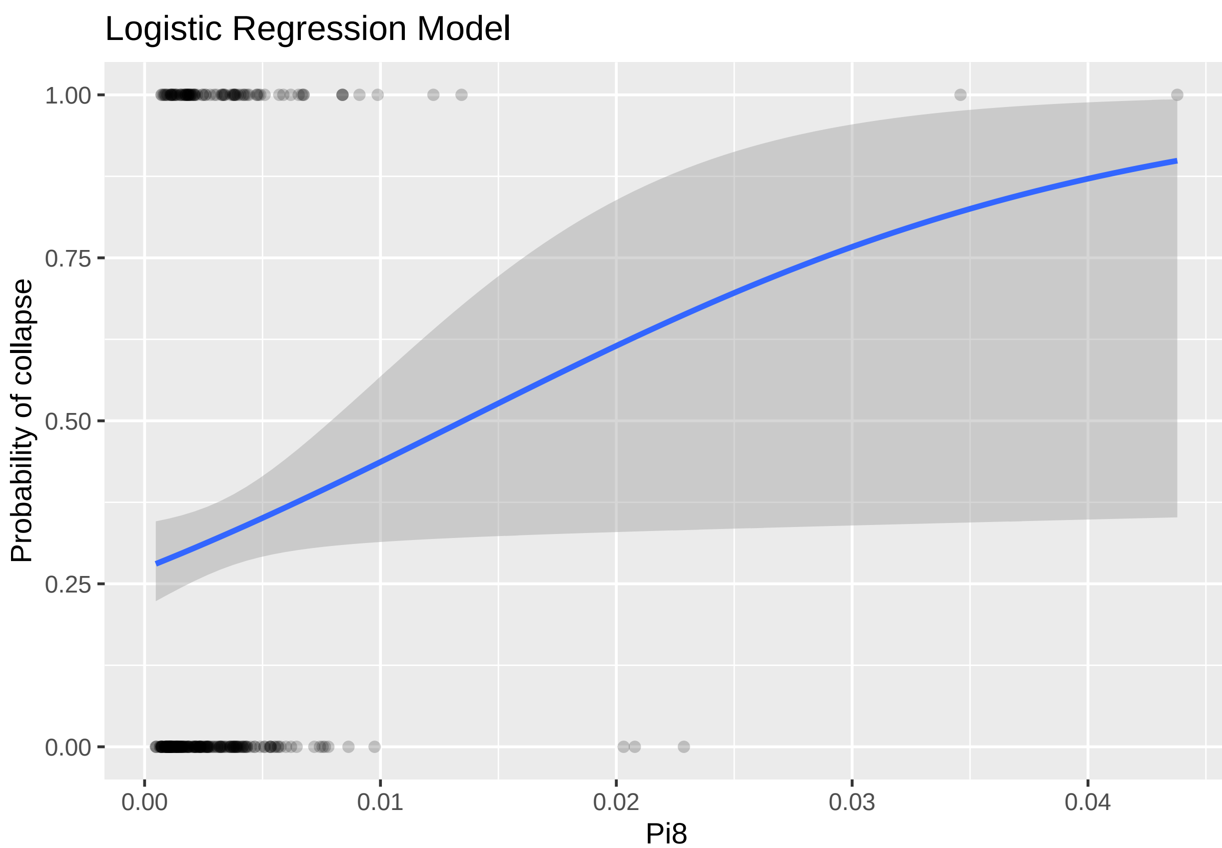
Variable: $\frac{PGA}{S_{1,amp,M} \cdot g}$

```r
logi8 <- logiStudy("Pi8", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```r
logiPlot(isol.train, "Pi8")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Ninth fit

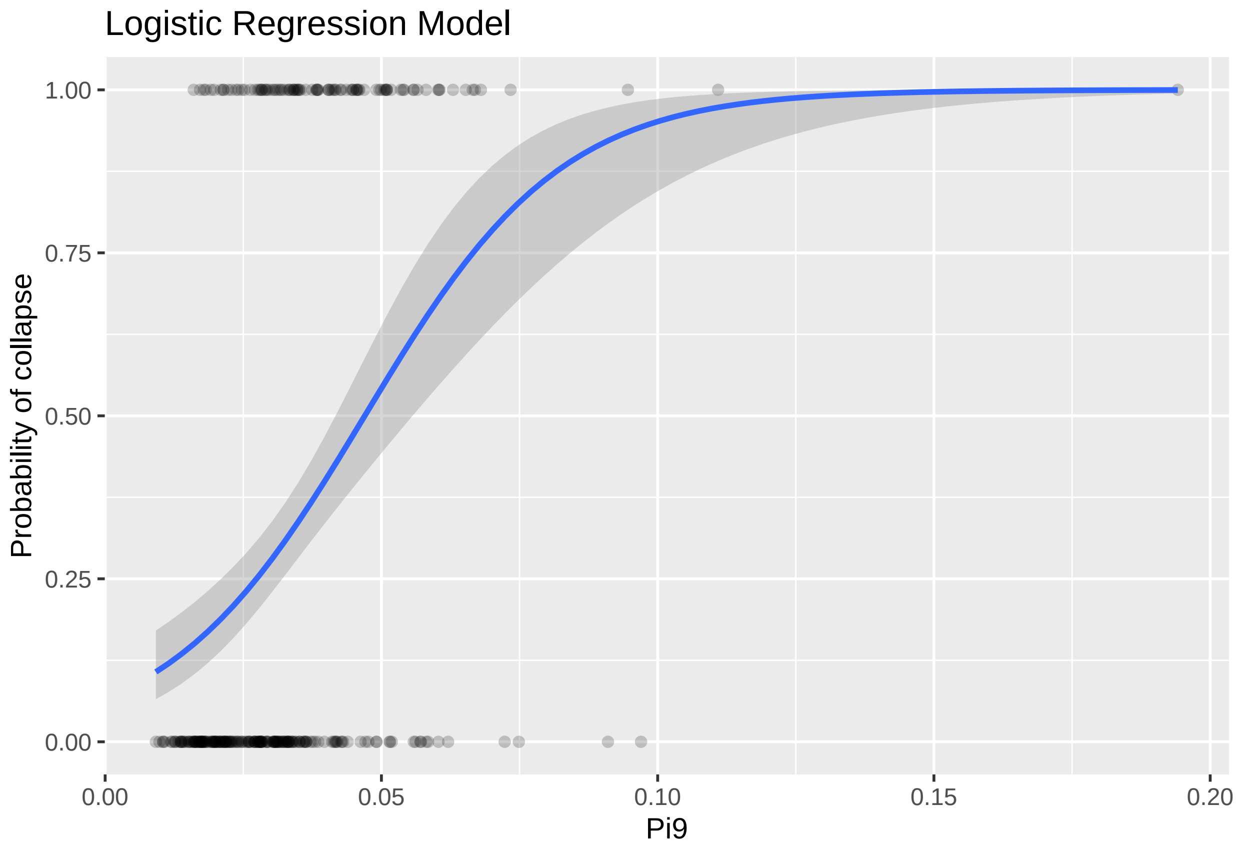
Variable: $\frac{PGV}{S_{1,amp,M} \cdot T_M \cdot g}$

```r
logi9 <- logiStudy("Pi9", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```r
logiPlot(isol.train, "Pi9")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Tenth fit

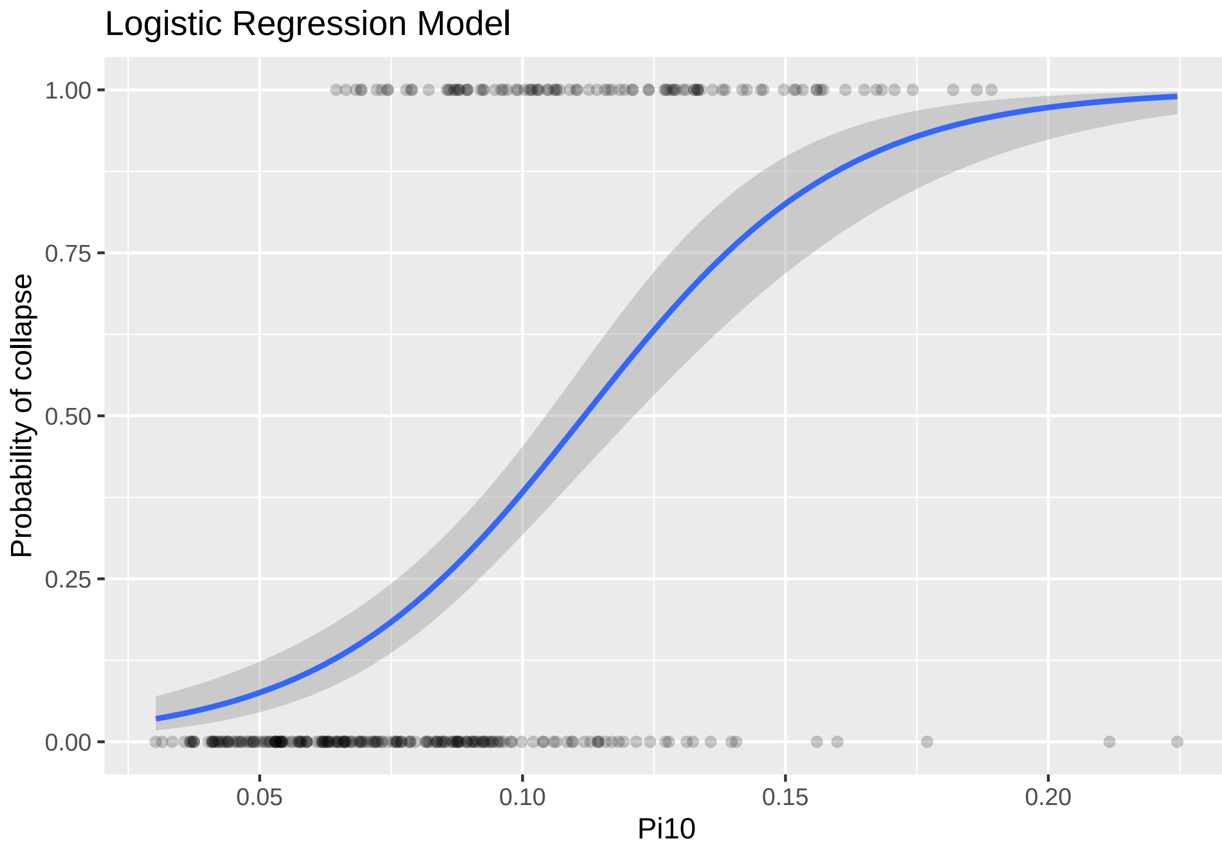
Variable: $\frac{FIV3}{S_{1,amp,M} \cdot T_M \cdot g}$

```r
logi10 <- logiStudy("Pi10", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```r
logiPlot(isol.train, "Pi10")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Eleventh fit

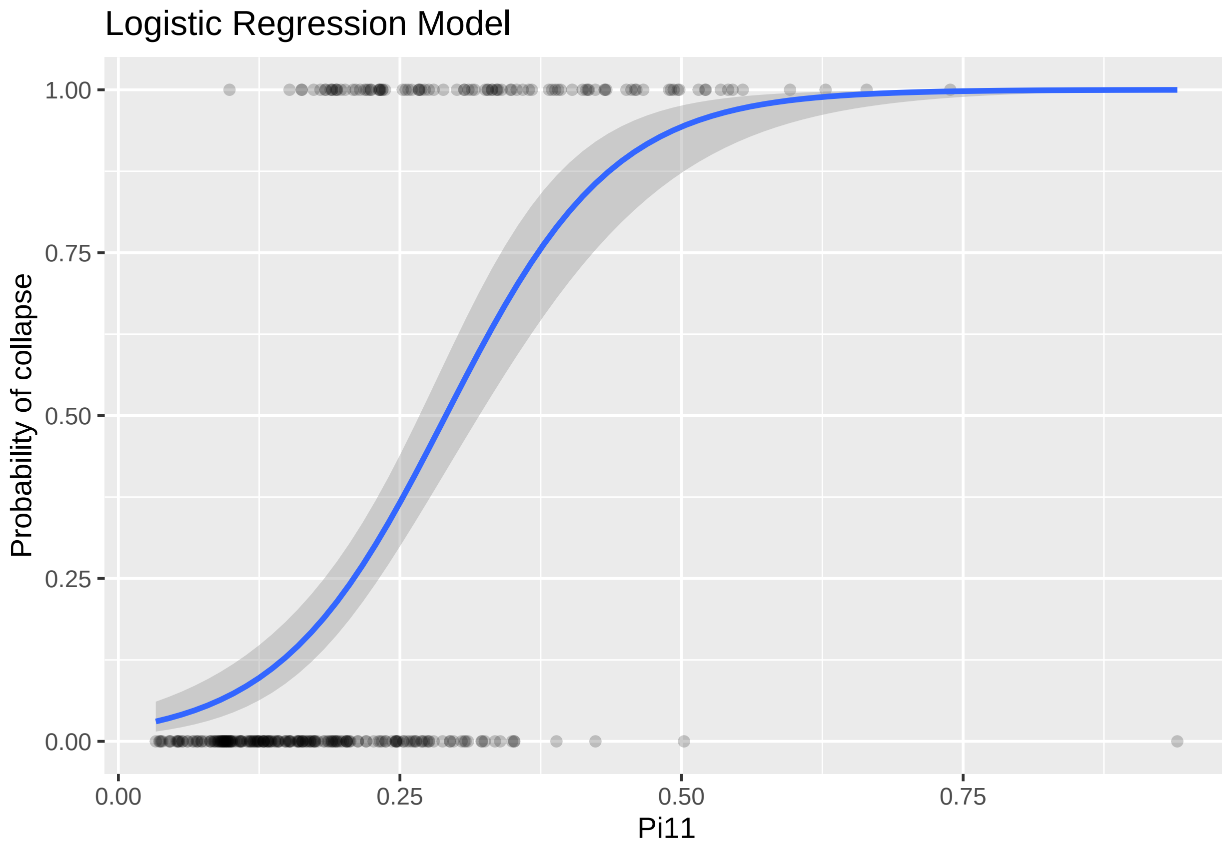
Variable: $\frac{IP(T_M)}{S_M \cdot D_M}$

```
logi11 <- logiStudy("Pi11", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```
logiPlot(isol.train, "Pi11")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Twelfth fit

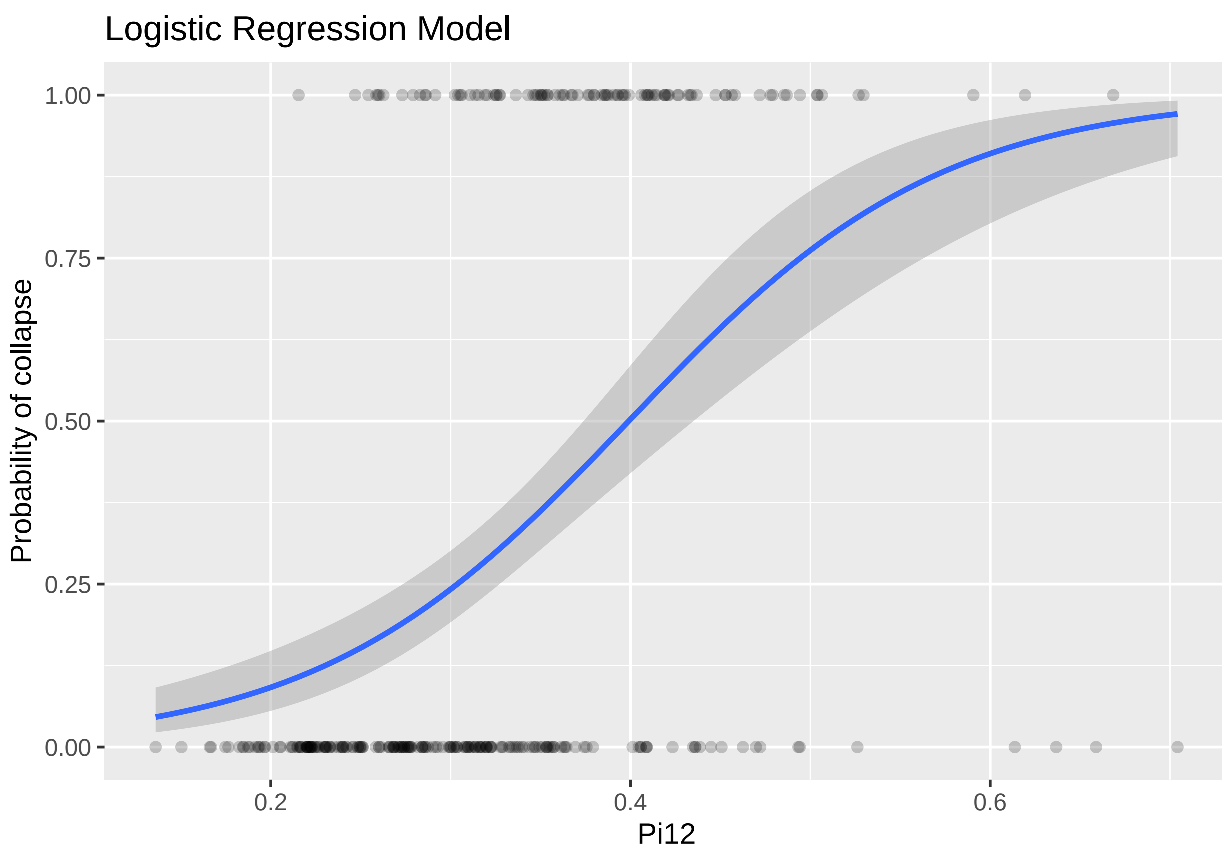
Variable: $\frac{FIV3}{S_M \cdot T_M \cdot g}$

```
logi12 <- logiStudy("Pi12", isol.train, isol.test)
```

```
## Waiting for profiling to be done...
```

```
logiPlot(isol.train, "Pi12")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Quick FWL check

Regress Pi1 on Pi2 to get residual

```
# pi.lm <- lm(Pi2 ~ Pi1, data = isol.full)
# Pi2.tilde <- pi.lm$residuals
# maxDrift <- isol.full$maxDrift
# fwl.lm <- lm(maxDrift ~ Pi2.tilde)
# summary(fwl.lm)
# plot(fwl.lm)
```

Power regression: maximum interstory drift

Create new data frame of the logarithm data.

```
isol.log <- data.frame(
  maxDrift = log(isol.full$maxDrift), Pi1 = log(isol.full$Pi1),  Pi2 = log(isol.full$Pi2),
  Pi3 = log(isol.full$Pi3), Pi4 = log(isol.full$Pi4), Pi5 = log(isol.full$Pi5),
  Pi6 = log(isol.full$Pi6), Pi7 = log(isol.full$Pi7)
)
```

Split data again for cross validation.

```
set.seed(1)

isol.log.train <- isol.log %>% sample_frac(0.8)
isol.log.test <- isol.log %>% setdiff(isol.log.train)
```

First regression

Variable: $\pi_1 = \frac{Sa_{avg}(T)}{S_{1,amp,M}}$

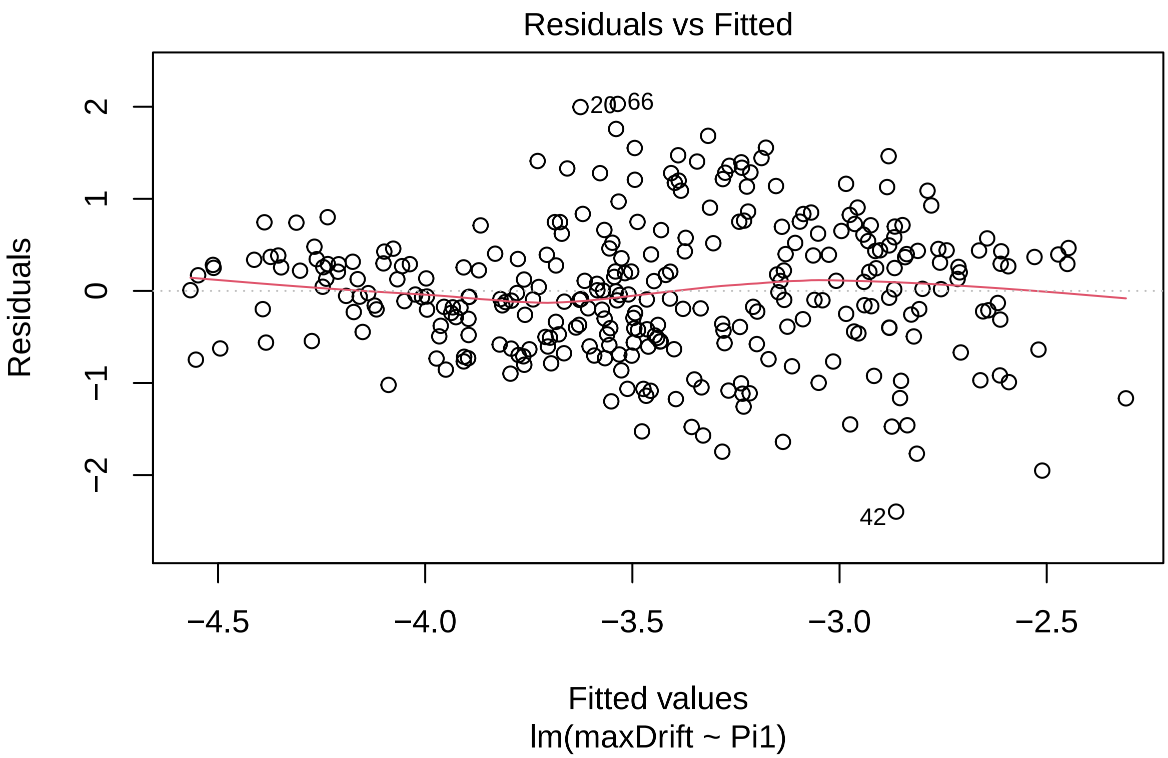
```r
# fit model
lnFitDrift1 <- lm(maxDrift ~ Pi1, data = isol.log.train)
summary(lnFitDrift1)
```

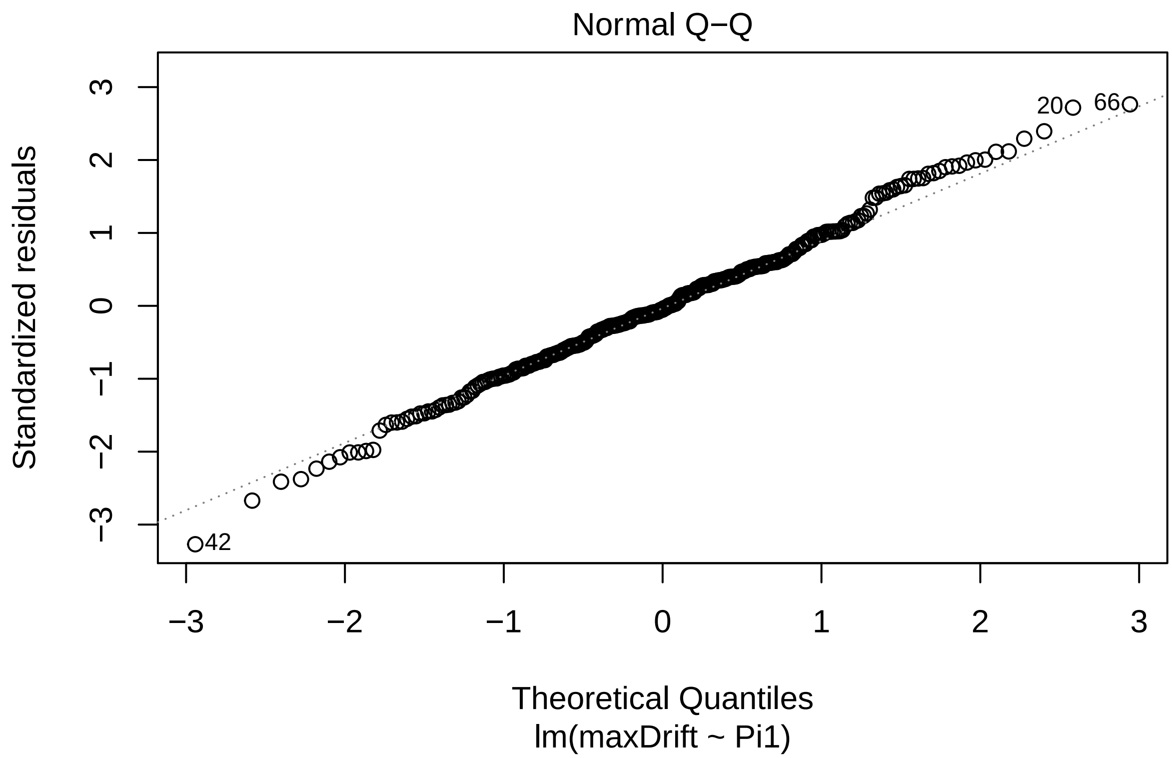
```
##
## Call:
## lm(formula = maxDrift ~ Pi1, data = isol.log.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.39699 -0.48150 -0.03862  0.43178  2.03032
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.56791    0.08464  -30.34   <2e-16 ***
## Pi1          1.53261    0.12966   11.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7359 on 305 degrees of freedom
## Multiple R-squared:  0.3142, Adjusted R-squared:  0.3119
## F-statistic: 139.7 on 1 and 305 DF,  p-value: < 2.2e-16
```

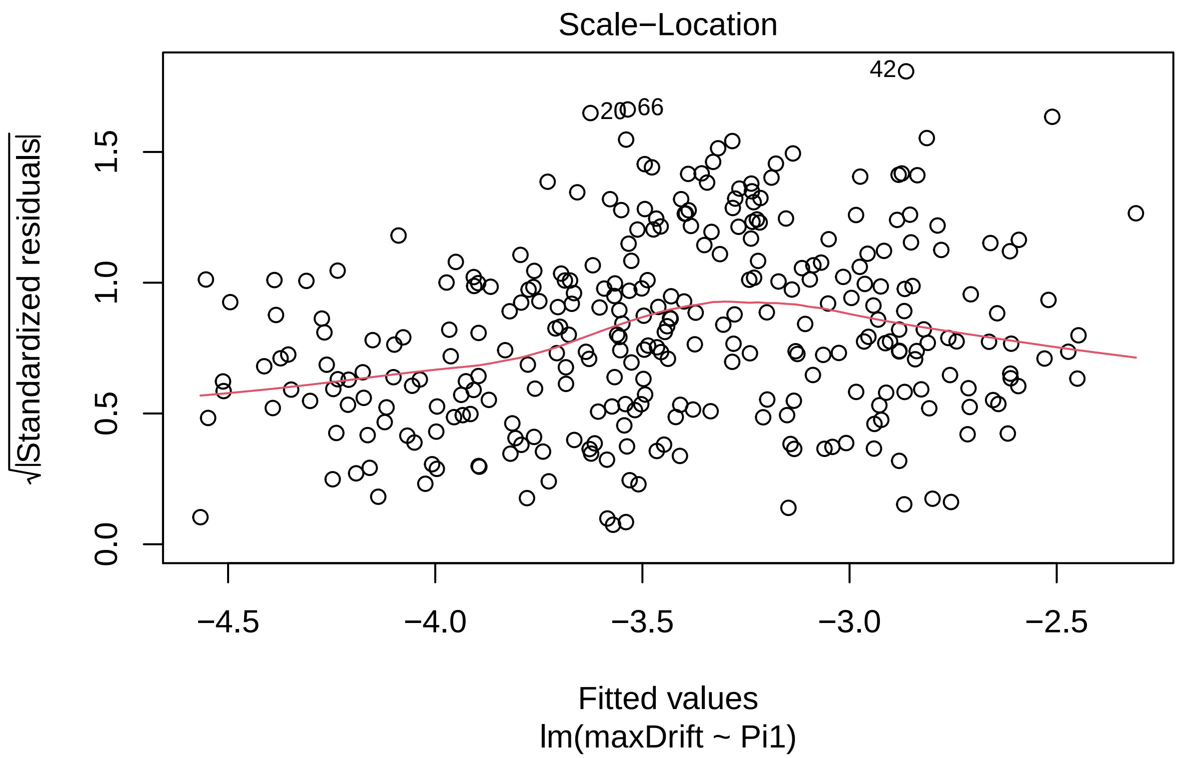
```r
confint(lnFitDrift1)
```

```
##                 2.5 %    97.5 %
## (Intercept) -2.734463 -2.401364
## Pi1          1.277471  1.787753
```

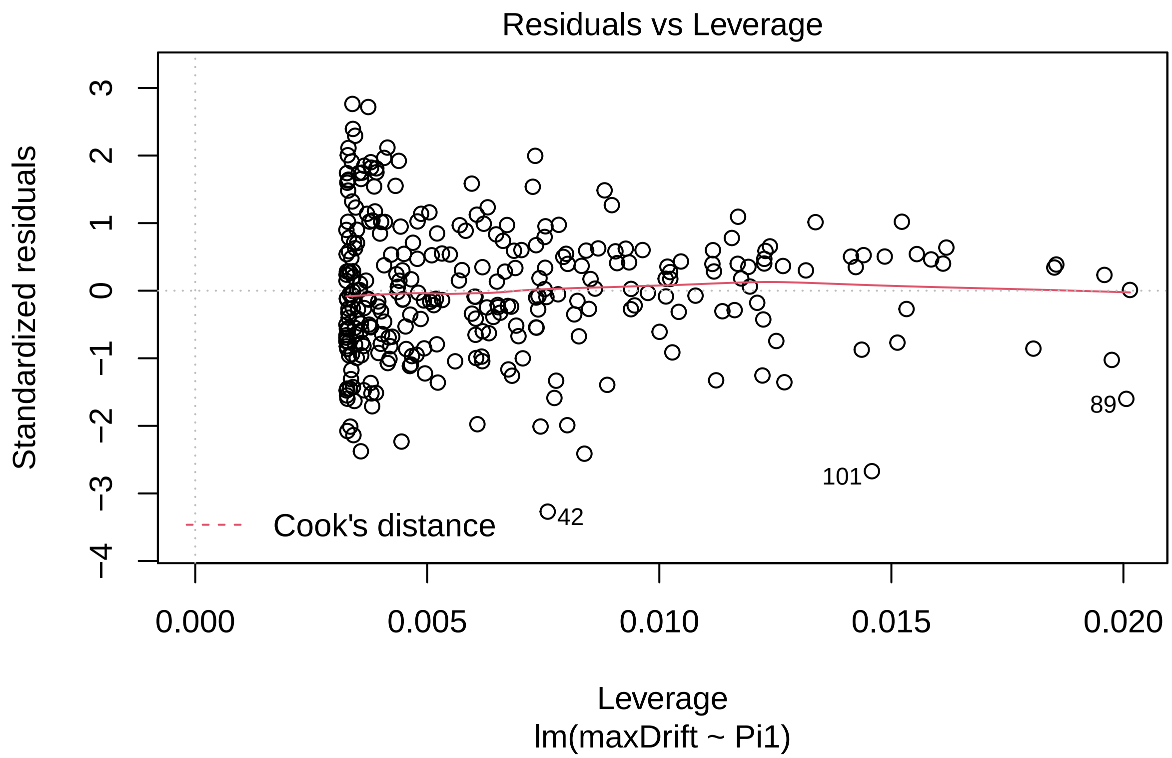
```r
plot(lnFitDrift1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(maxDrift ~ Pi1)
2
1
0
−1
−2
−4.5
−4.0
−3.5
−3.0
−2.5
20
66
42


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(maxDrift ~ Pi1)
3
2
1
0
−1
−2
−3
20
66
42



Scale−Location
42
20
66
√|Standardized residuals|
0.0
0.5
1.0
1.5
−4.5
−4.0
−3.5
−3.0
−2.5
Fitted values
lm(maxDrift ~ Pi1)




```r
sigma1 <- summary(lnFitDrift1)$sigma
```

```r
# check prediction
test1.drift <- predict(lnFitDrift1, isol.log.test)
test1.drift.comparison <- data.frame(cbind(actual = isol.log.test$maxDrift, predicted = test1.drift))
cor.1 <- cor(test1.drift.comparison)
```

Second regression

Variable: $\pi_2 = \frac{Sa_{avg}(T)}{S_M}$

```r
# fit model
lnFitDrift2 <- lm(maxDrift ~ Pi2, data = isol.log.train)
summary(lnFitDrift2)
```

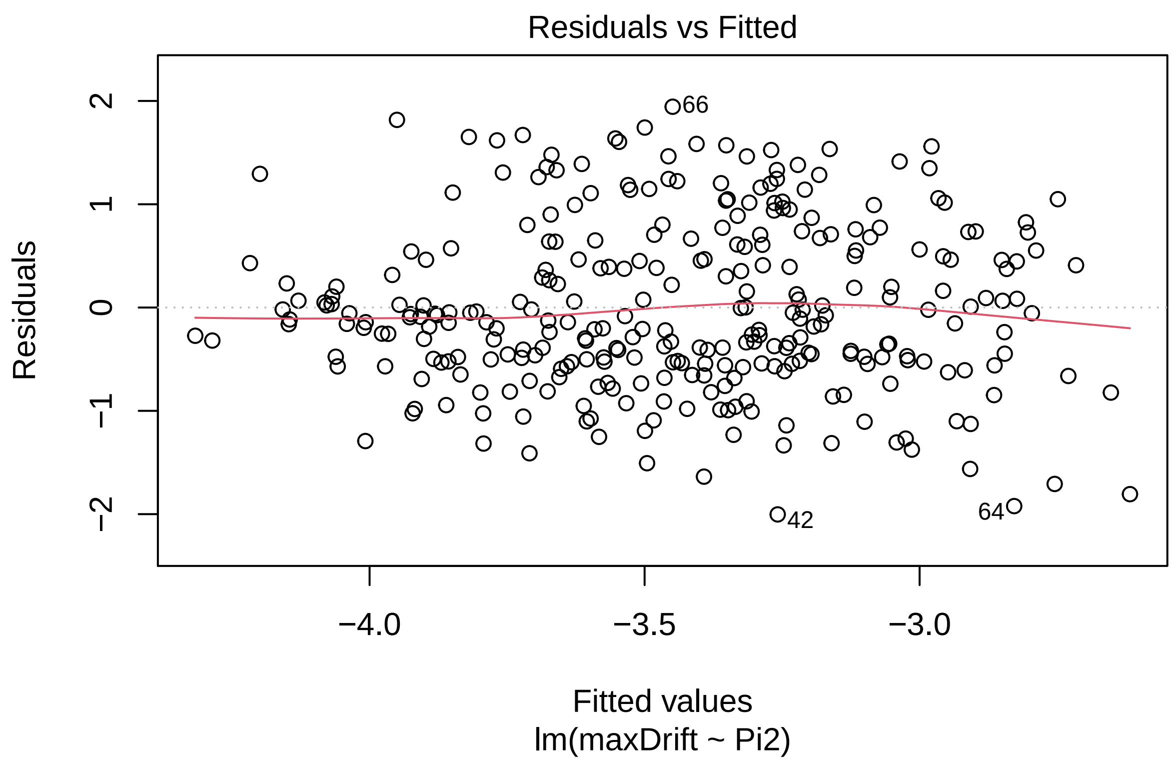
```
##
## Call:
## lm(formula = maxDrift ~ Pi2, data = isol.log.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0025 -0.5396 -0.1155  0.5805  1.9439
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4.5203     0.1480  -30.53  < 2e-16 ***
## Pi2           1.4079     0.1826    7.71  1.8e-13 ***
```

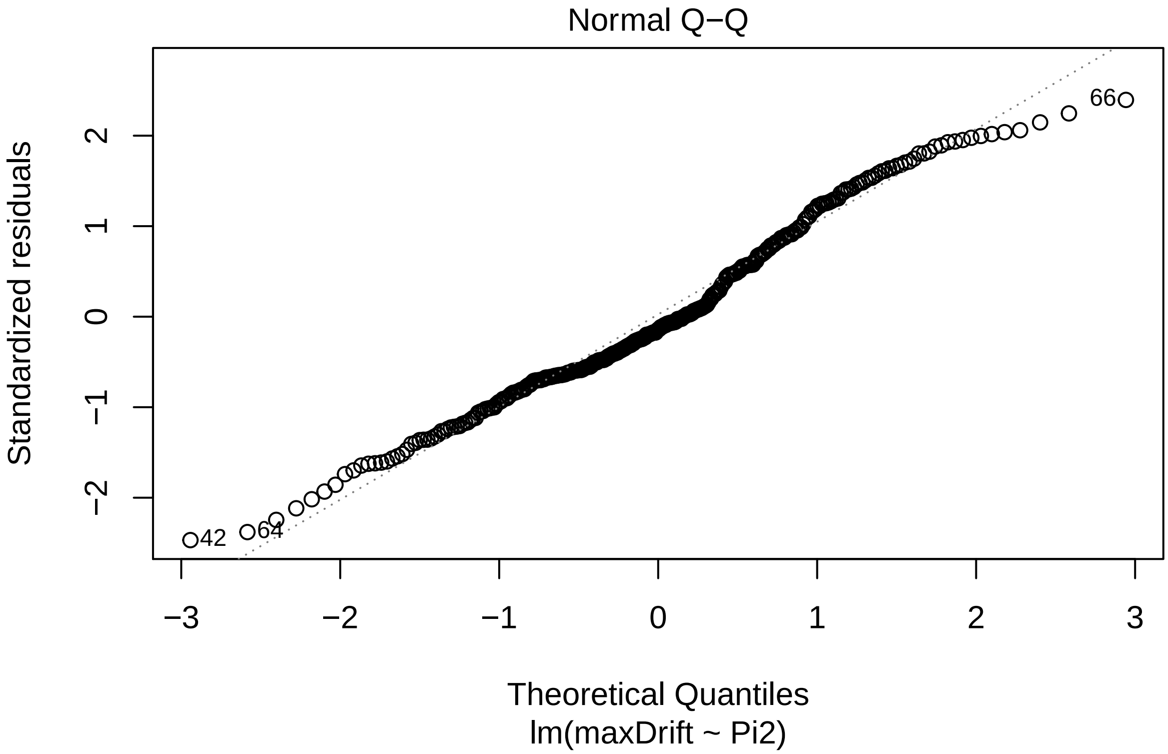
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.813 on 305 degrees of freedom
## Multiple R-squared:  0.1631, Adjusted R-squared:  0.1603
## F-statistic: 59.44 on 1 and 305 DF,  p-value: 1.797e-13
```

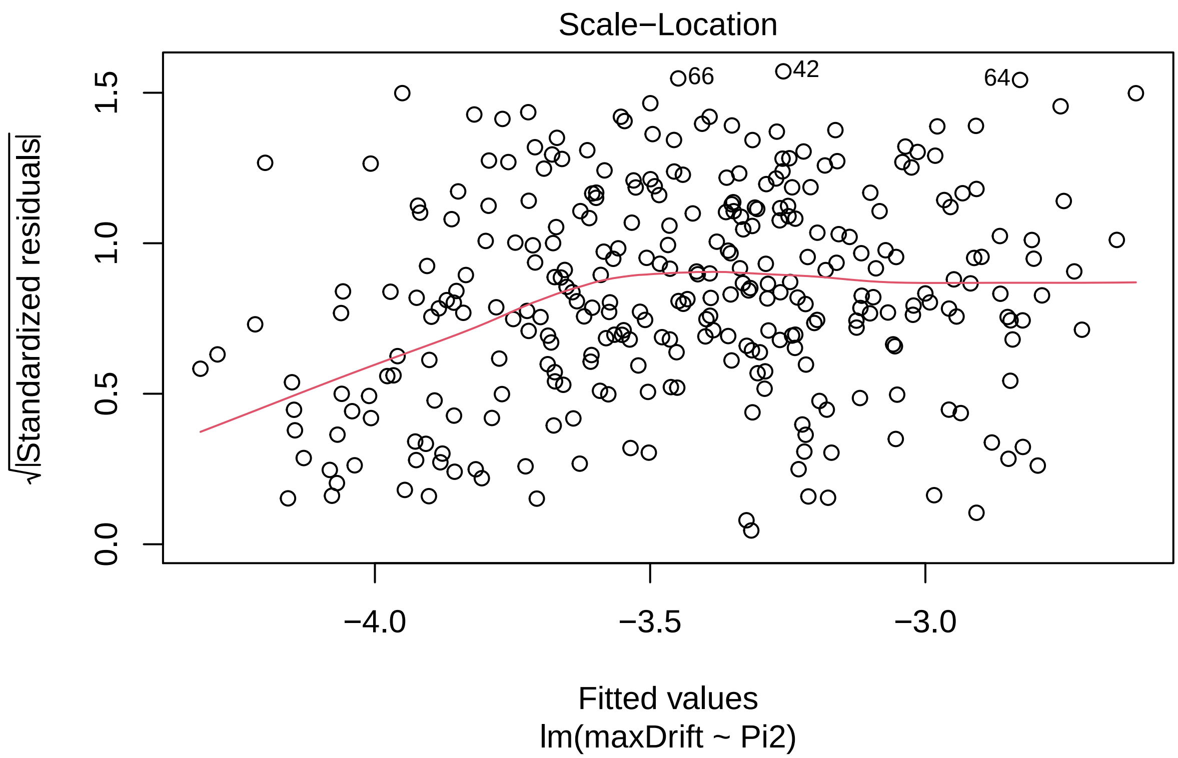
```
confint(lnFitDrift2)
```

```
##                  2.5 %    97.5 %
## (Intercept) -4.811602 -4.228988
## Pi2          1.048529  1.767216
```

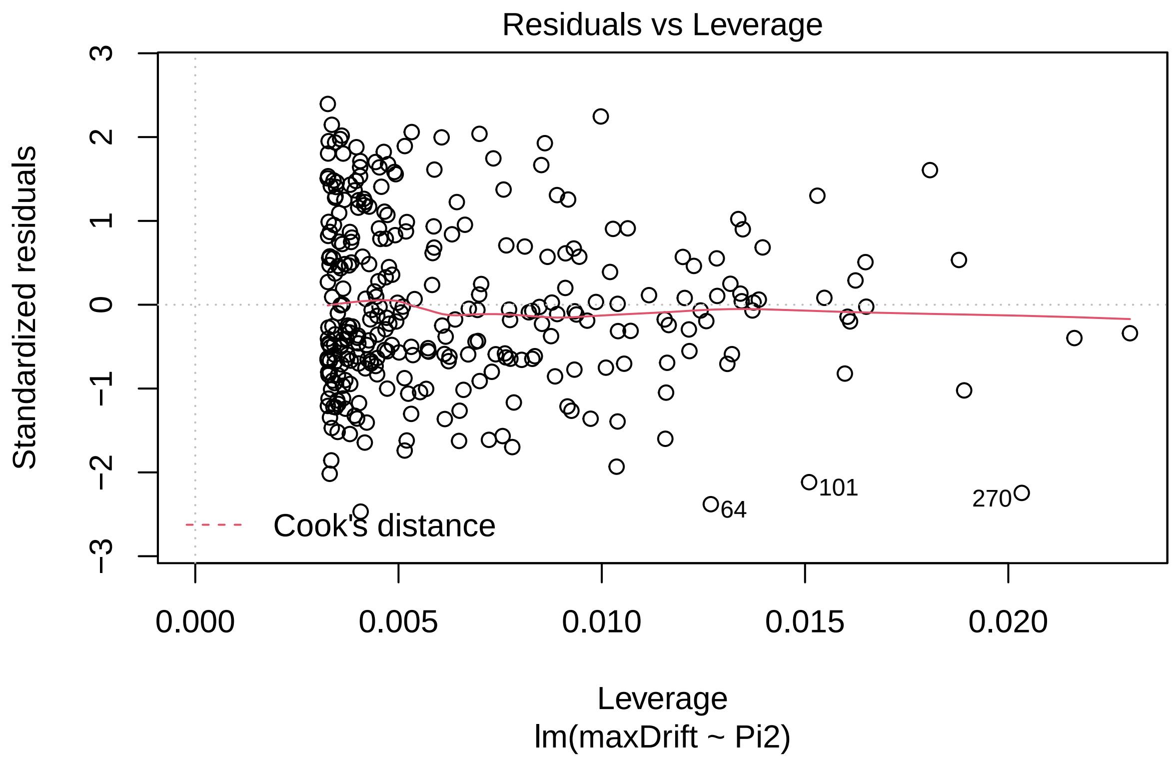
```
plot(lnFitDrift2)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(maxDrift ~ Pi2)
42
64
66
−3
−2
−1
0
1
2
3



Scale−Location
66
42
64
1.5
1.0
0.5
0.0
√|Standardized residuals|
−4.0
−3.5
−3.0
Fitted values
lm(maxDrift ~ Pi2)




```
sigma2 <- summary(lnFitDrift2)$sigma
```

Third regression

Variable: $\pi_3 = \frac{Sa(T_2)}{S_{1,amp,M}}$

```
# fit model
lnFitDrift3 <- lm(maxDrift ~ Pi3, data = isol.log.train)
summary(lnFitDrift3)
```

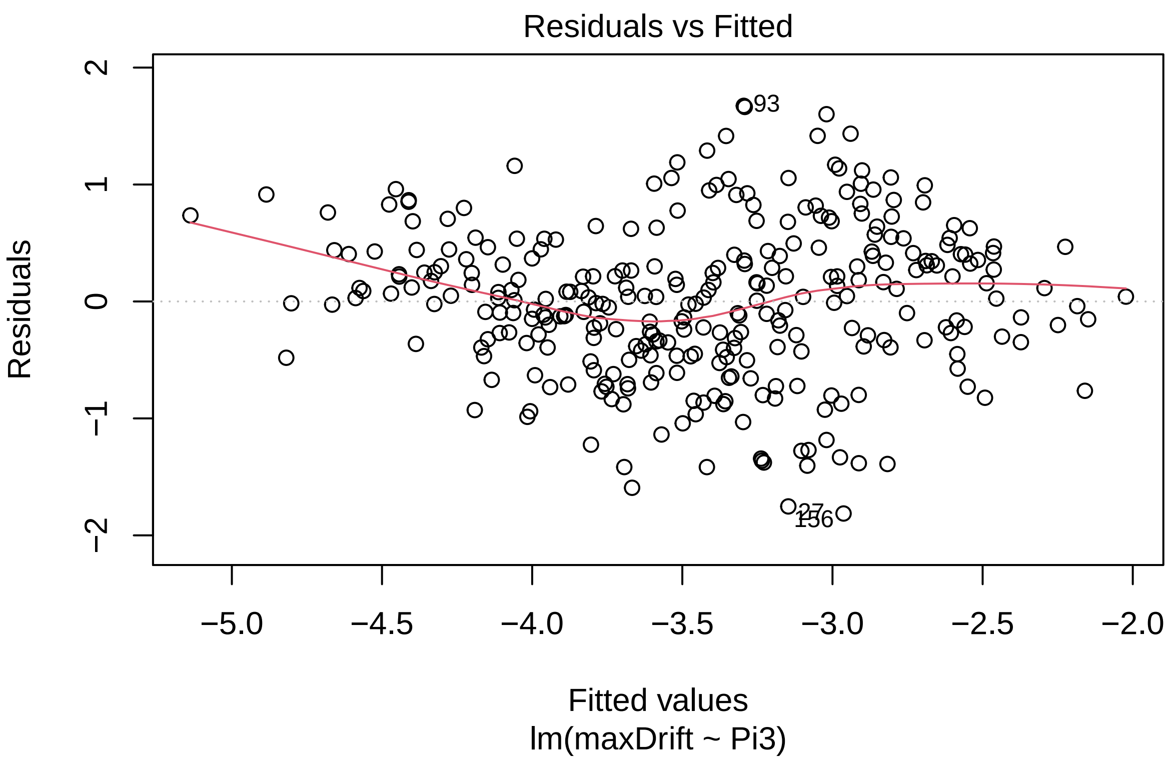
```
##
## Call:
## lm(formula = maxDrift ~ Pi3, data = isol.log.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.81325 -0.39324  0.03041  0.41360  1.67295
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.64583    0.11658  -14.12   <2e-16 ***
## Pi3          1.22902    0.07584   16.21   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6514 on 305 degrees of freedom
## Multiple R-squared:  0.4627, Adjusted R-squared:  0.4609
```

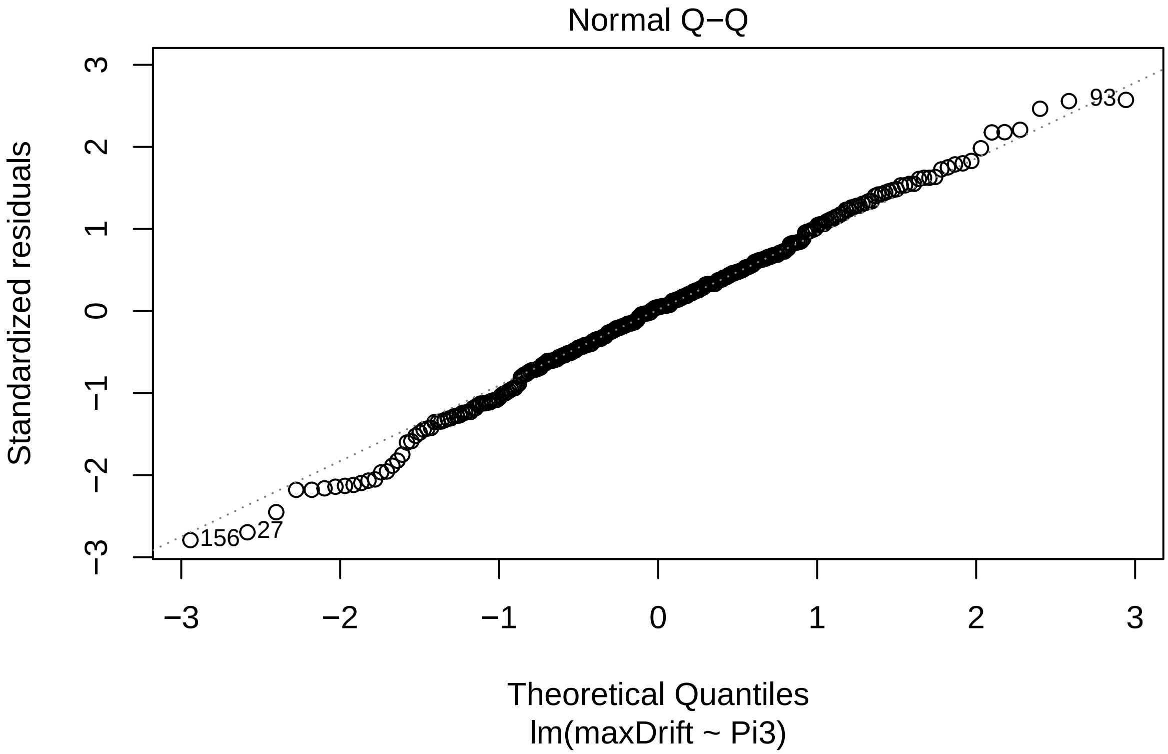
```
## F-statistic: 262.6 on 1 and 305 DF,  p-value: < 2.2e-16
```

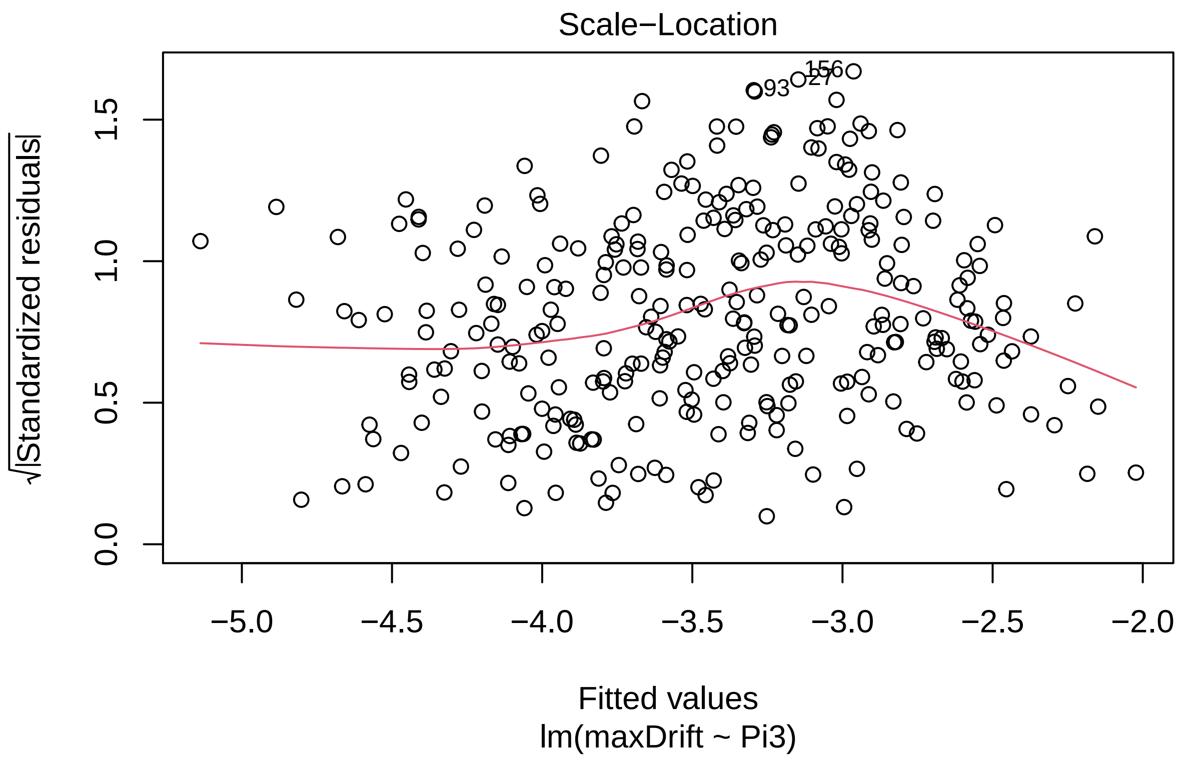
```r
confint(lnFitDrift3)
```

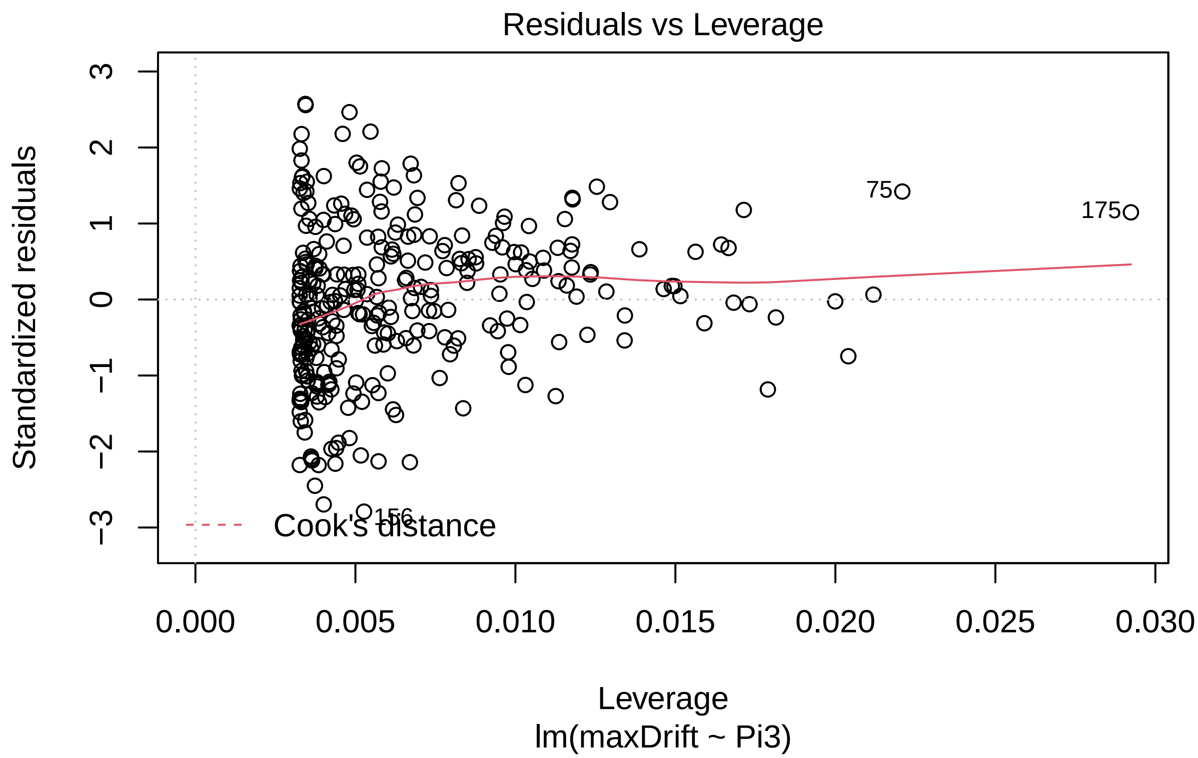
```
##                  2.5 %    97.5 %
## (Intercept) -1.875228 -1.416429
## Pi3          1.079793  1.378249
```

```r
plot(lnFitDrift3)
```




Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
−3
156
27
93
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(maxDrift ~ Pi3)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−5.0
−4.5
−4.0
−3.5
−3.0
−2.5
−2.0
Fitted values
lm(maxDrift ~ Pi3)
93
156
27




```r
sigma3 <- summary(lnFitDrift3)$sigma
```

Fourth regression

Variable: $\pi_4 = \frac{Sa(T_2)}{S_M}$

```r
# fit model
lnFitDrift4 <- lm(maxDrift ~ Pi4, data = isol.log.train)
summary(lnFitDrift4)
```

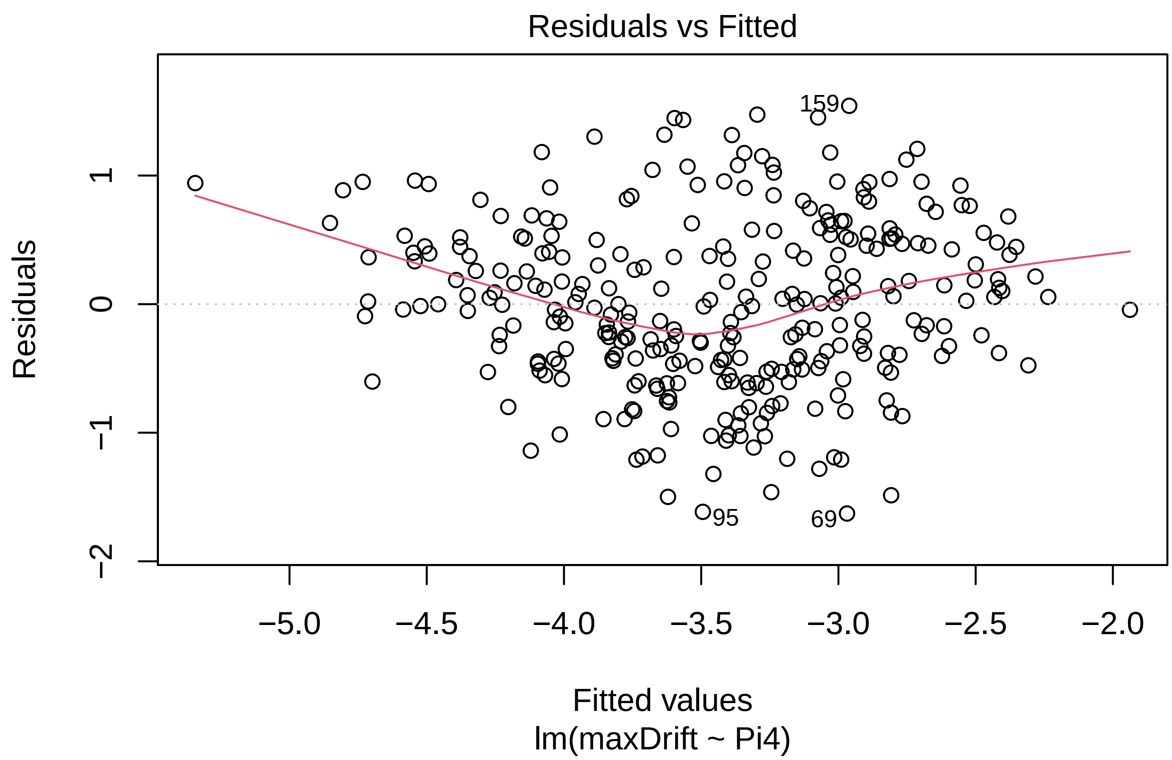
```
##
## Call:
## lm(formula = maxDrift ~ Pi4, data = isol.log.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6283 -0.4632 -0.0033  0.4768  1.5423
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.24694    0.03939  -82.43   <2e-16 ***
## Pi4          1.57386    0.09919   15.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6577 on 305 degrees of freedom
## Multiple R-squared:  0.4522, Adjusted R-squared:  0.4504
```

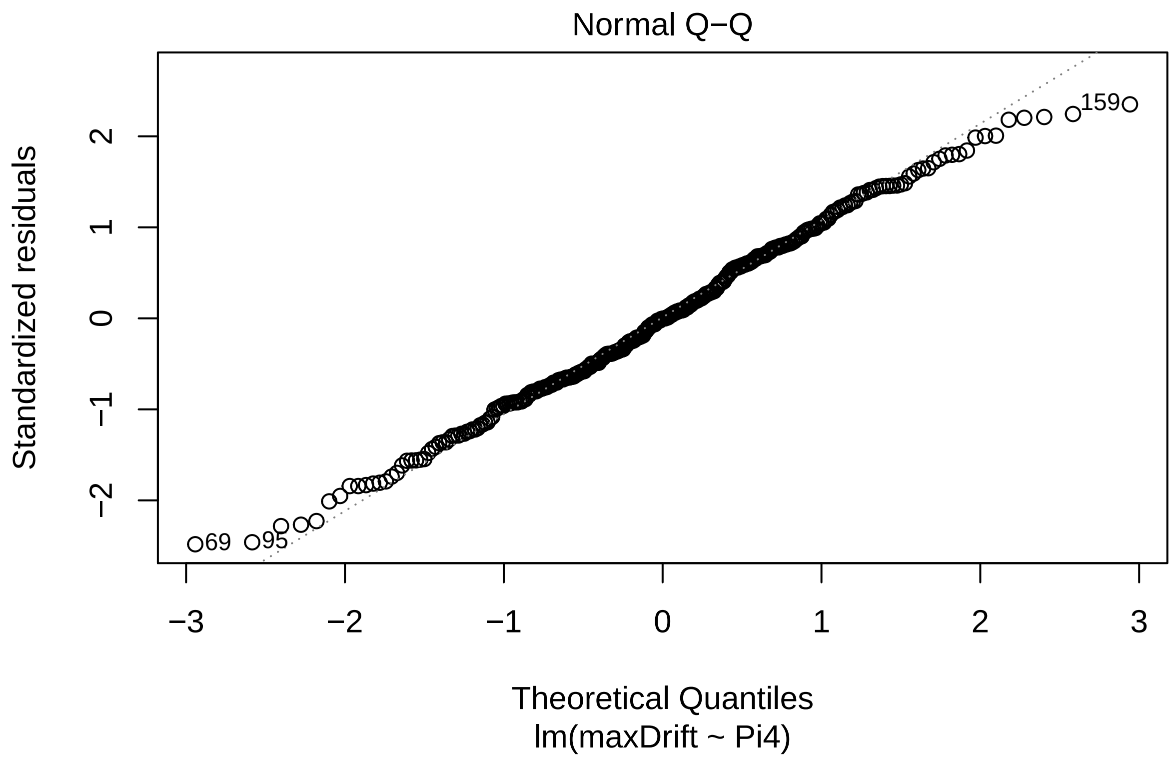
```
## F-statistic: 251.8 on 1 and 305 DF,  p-value: < 2.2e-16
```

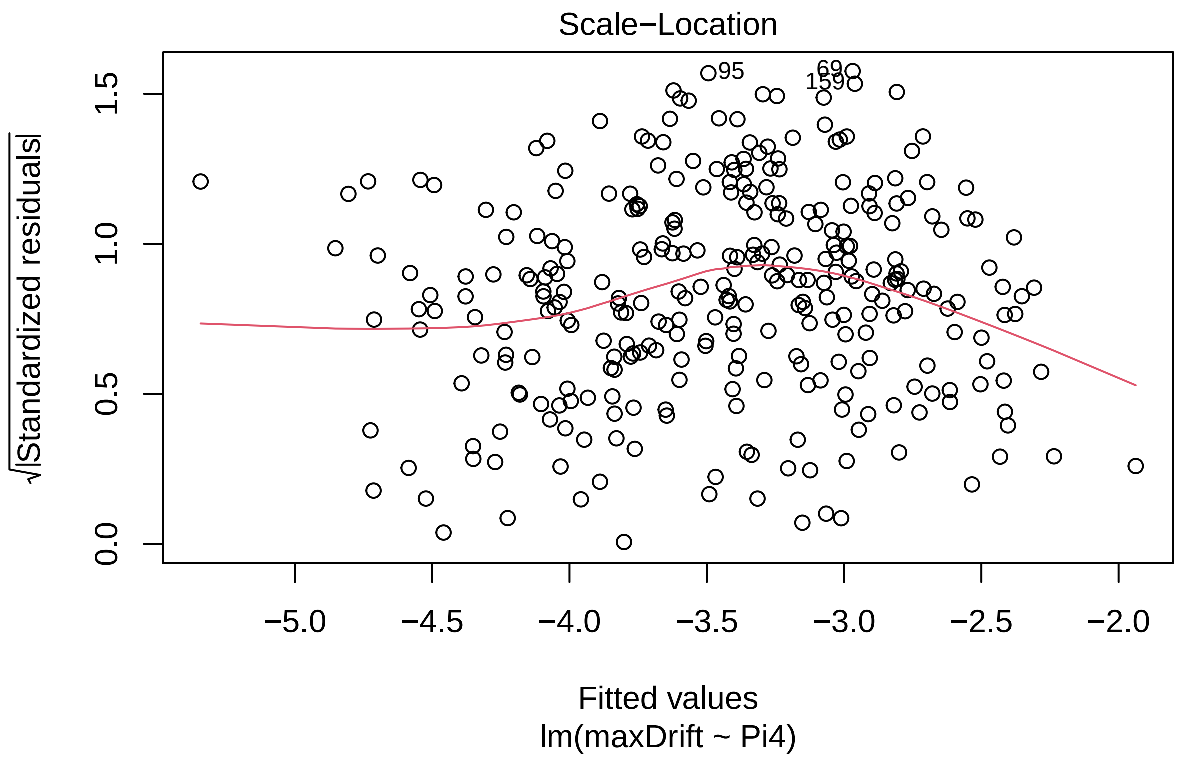
```r
confint(lnFitDrift4)
```

```
##                 2.5 %    97.5 %
## (Intercept) -3.324458 -3.169427
## Pi4          1.378687  1.769043
```

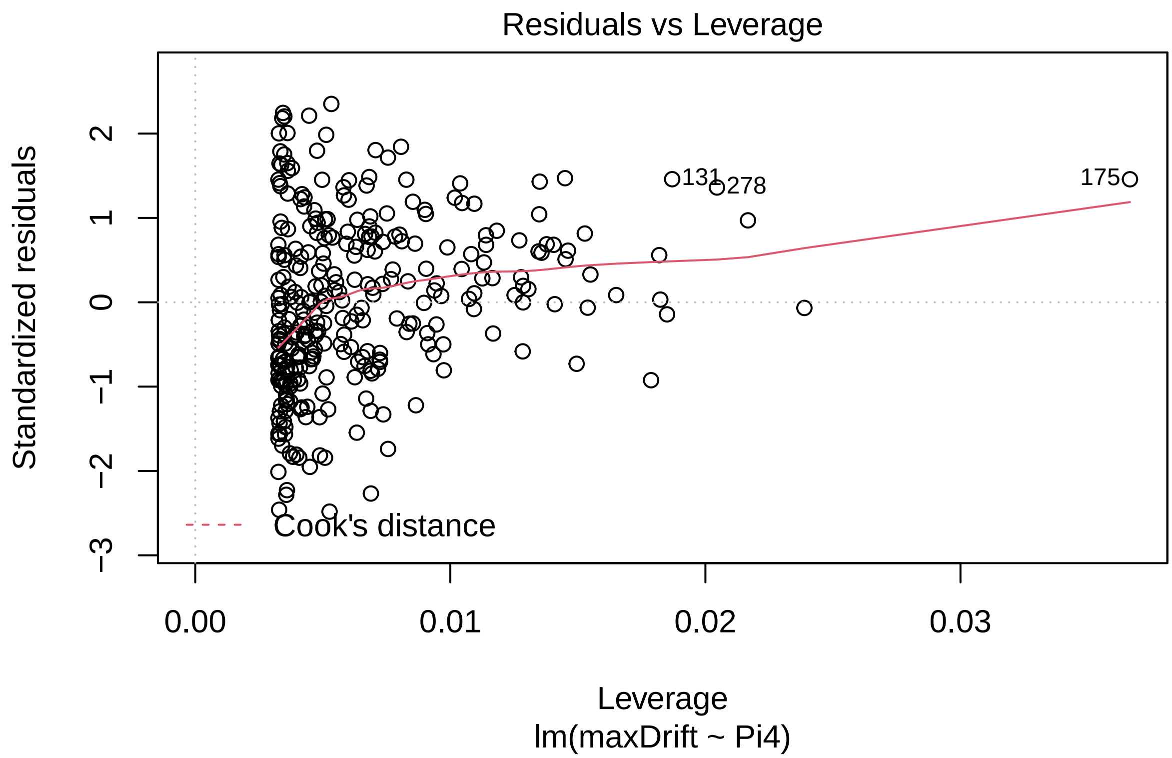
```r
plot(lnFitDrift4)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(maxDrift ~ Pi4)
69
95
159
−3
−2
−1
0
1
2
3



Scale−Location
95
69
159
√|Standardized residuals|
0.0
0.5
1.0
1.5
−5.0
−4.5
−4.0
−3.5
−3.0
−2.5
−2.0
Fitted values
lm(maxDrift ~ Pi4)




```
sigma4 <- summary(lnFitDrift4)$sigma
```

Fifth regression

Variable: $\pi_5 = \frac{Sa(T_M)}{S_{1,amp,M}}$

```
# fit model
lnFitDrift5 <- lm(maxDrift ~ Pi5, data = isol.log.train)
summary(lnFitDrift5)
```

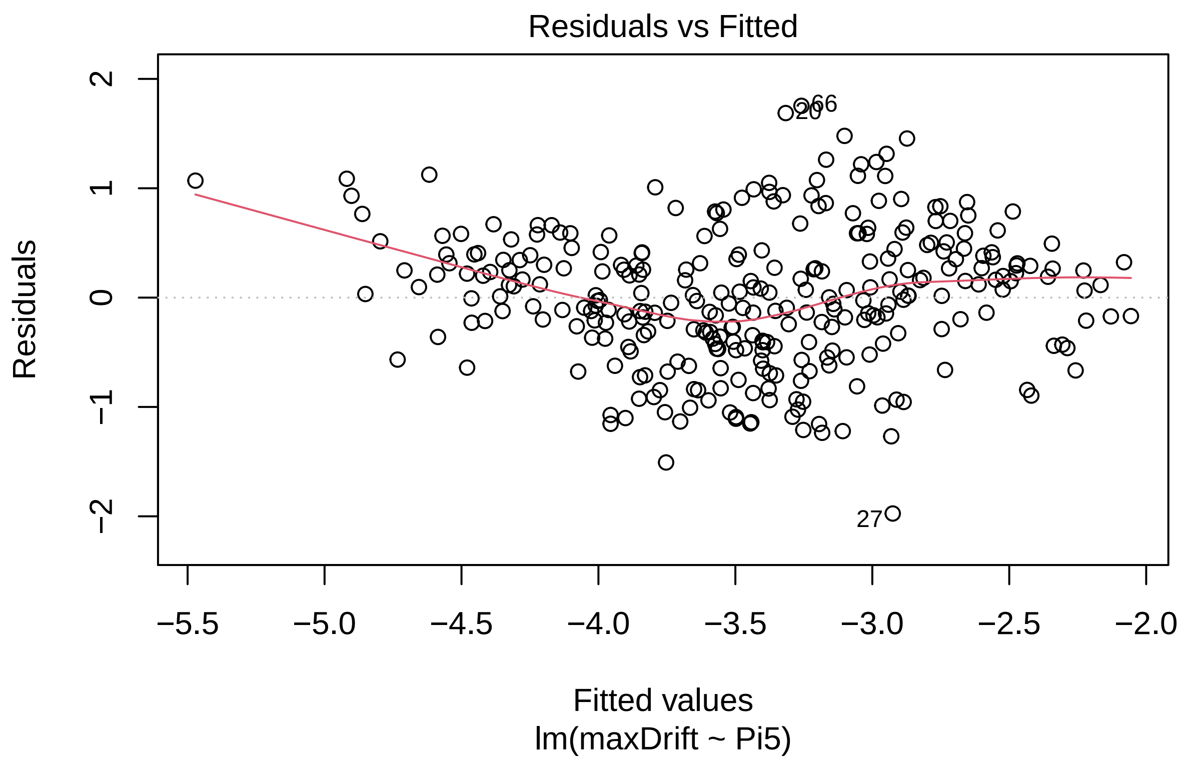
```
##
## Call:
## lm(formula = maxDrift ~ Pi5, data = isol.log.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.97430 -0.40754  0.01804  0.39447  1.75363
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.68932    0.10463  -16.14   <2e-16 ***
## Pi5          1.40028    0.07887   17.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6232 on 305 degrees of freedom
## Multiple R-squared:  0.5083, Adjusted R-squared:  0.5066
```

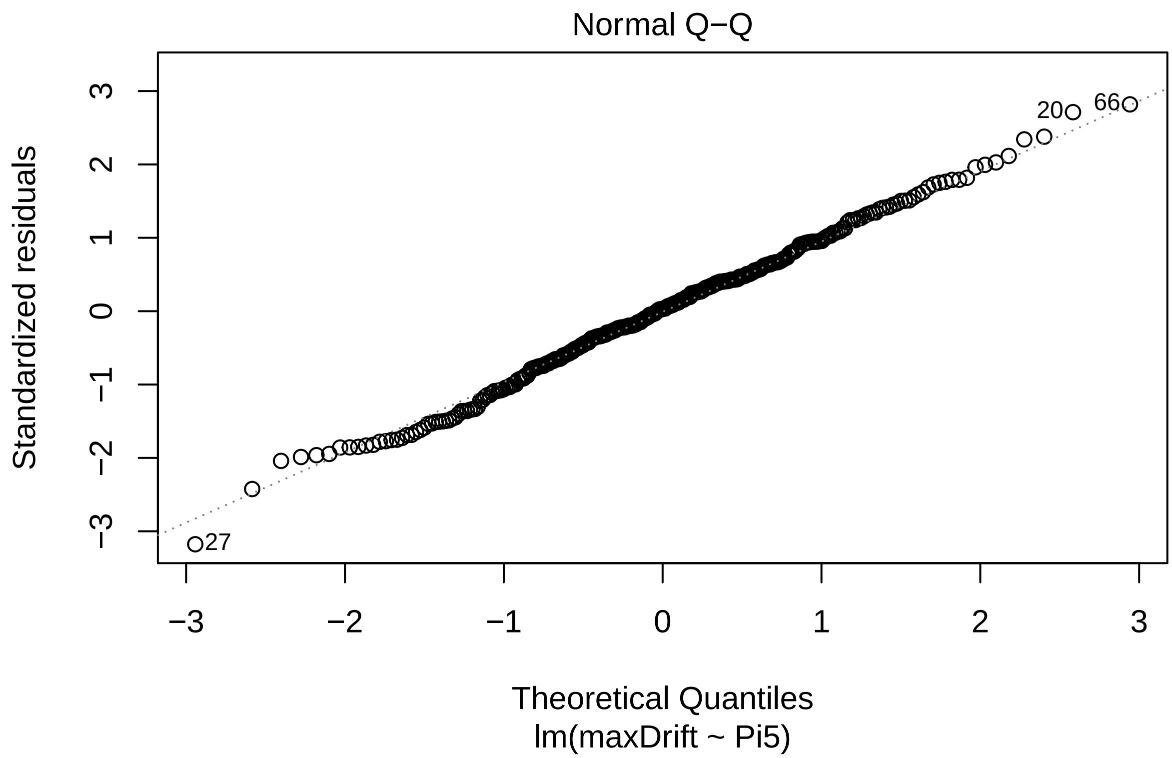
```
## F-statistic: 315.2 on 1 and 305 DF,  p-value: < 2.2e-16
```

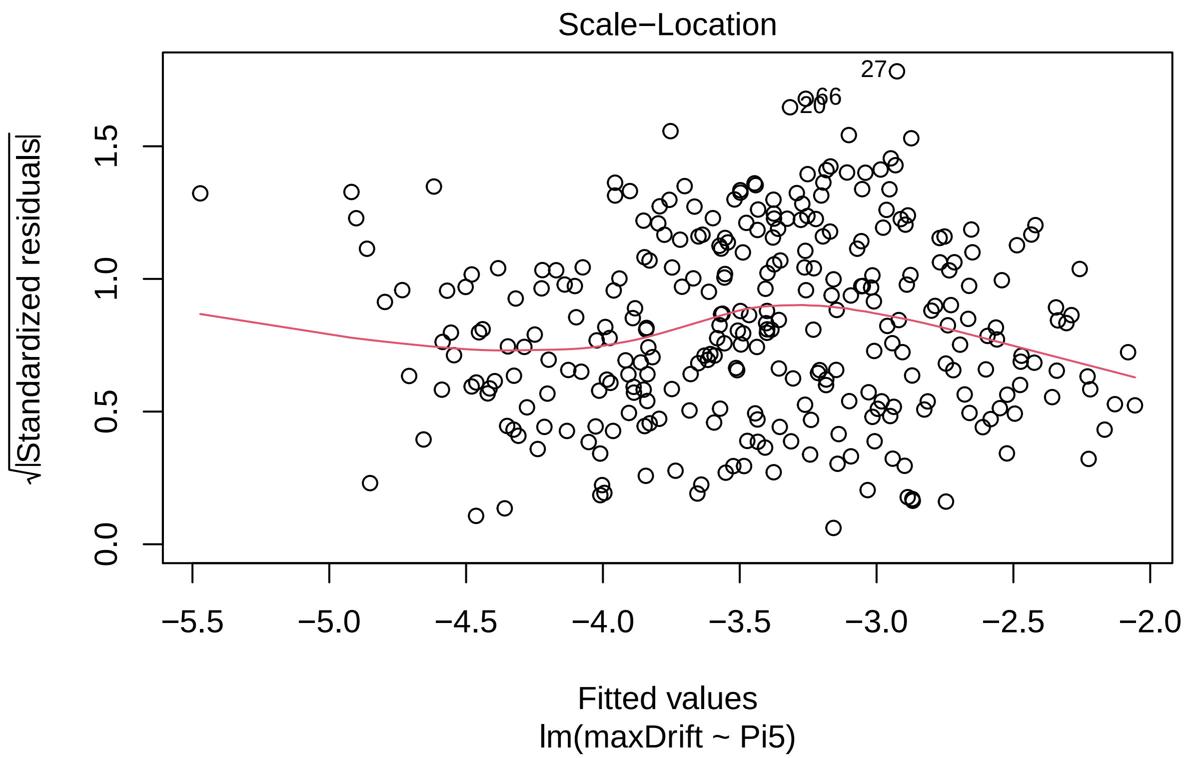
```r
confint(lnFitDrift5)
```

```
##                   2.5 %    97.5 %
## (Intercept) -1.895218 -1.483424
## Pi5          1.245083  1.555468
```

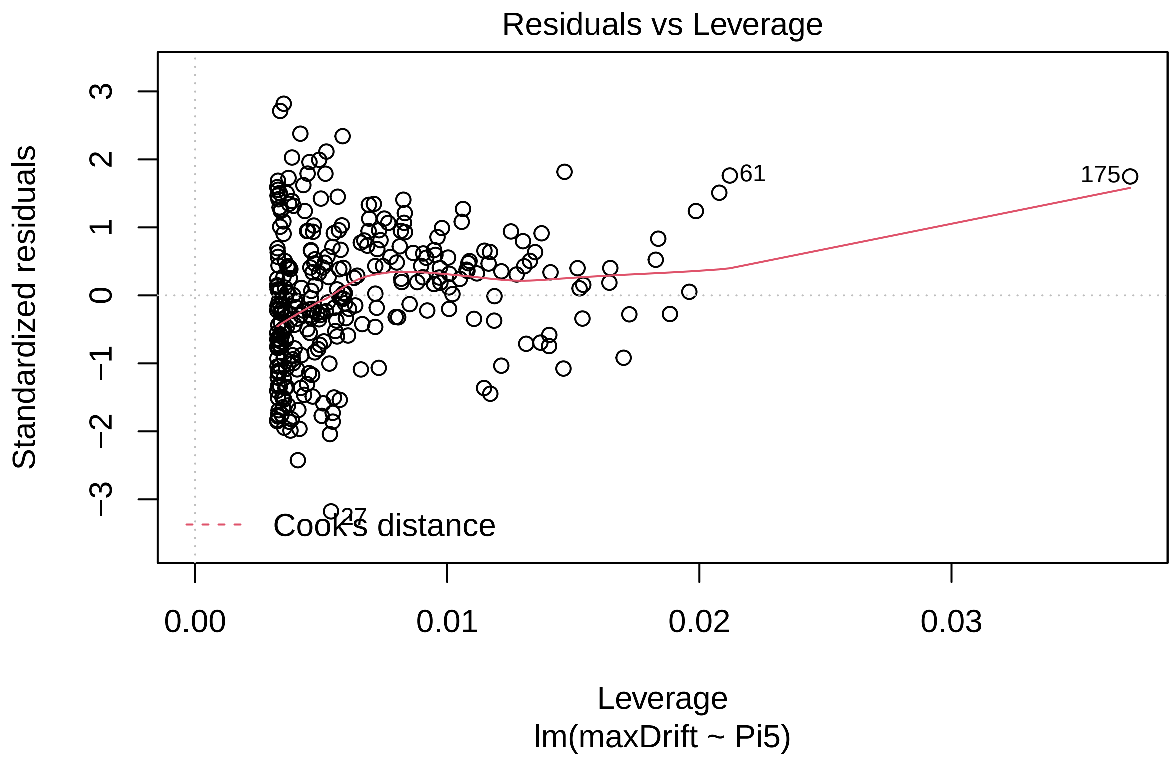
```r
plot(lnFitDrift5)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(maxDrift ~ Pi5)
27
20
66
−3
−2
−1
0
1
2
3



Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−5.5
−5.0
−4.5
−4.0
−3.5
−3.0
−2.5
−2.0
27
20
66
Fitted values
lm(maxDrift ~ Pi5)




```r
sigma5 <- summary(lnFitDrift5)$sigma
```

Sixth regression

Variable: $\pi_6 = \frac{Sa(T_M)}{S_M}$

```r
# fit model
lnFitDrift6 <- lm(maxDrift ~ Pi6, data = isol.log.train)
summary(lnFitDrift6)
```

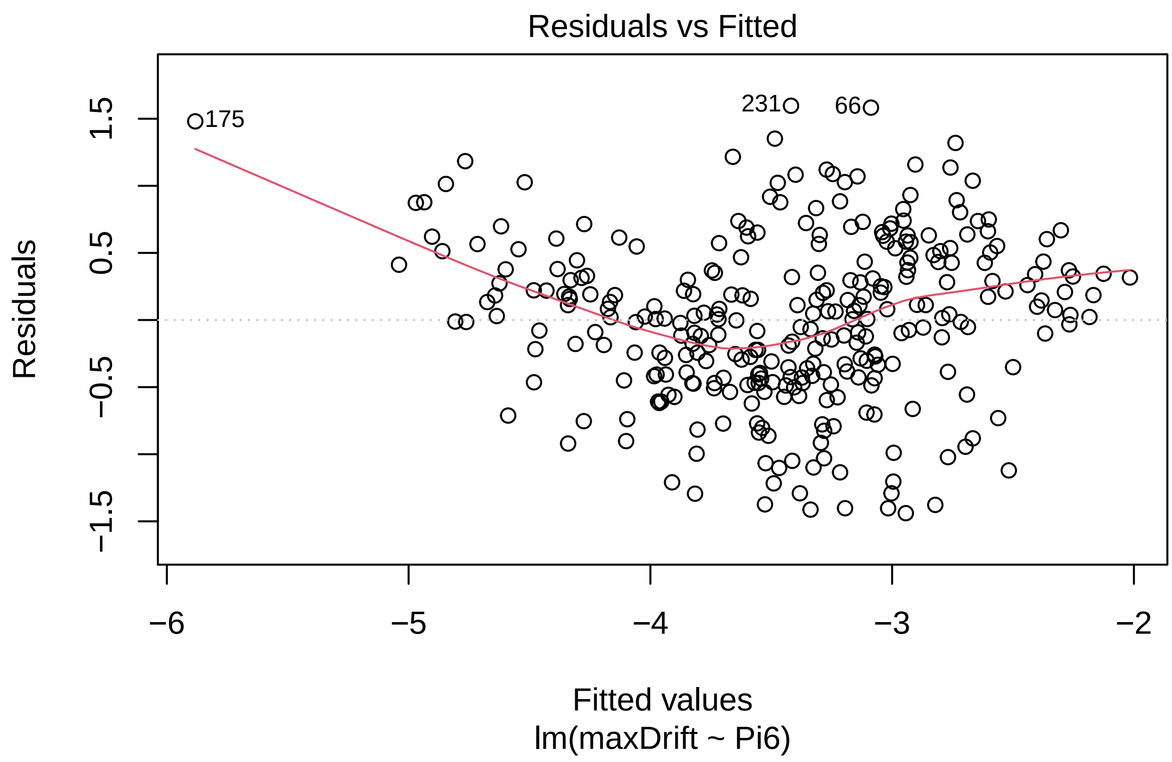
```
##
## Call:
## lm(formula = maxDrift ~ Pi6, data = isol.log.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.43924 -0.42174  0.02268  0.42626  1.59656
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.60615    0.03623  -99.52   <2e-16 ***
## Pi6          1.91019    0.10445   18.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6137 on 305 degrees of freedom
## Multiple R-squared:  0.523,  Adjusted R-squared:  0.5215
```

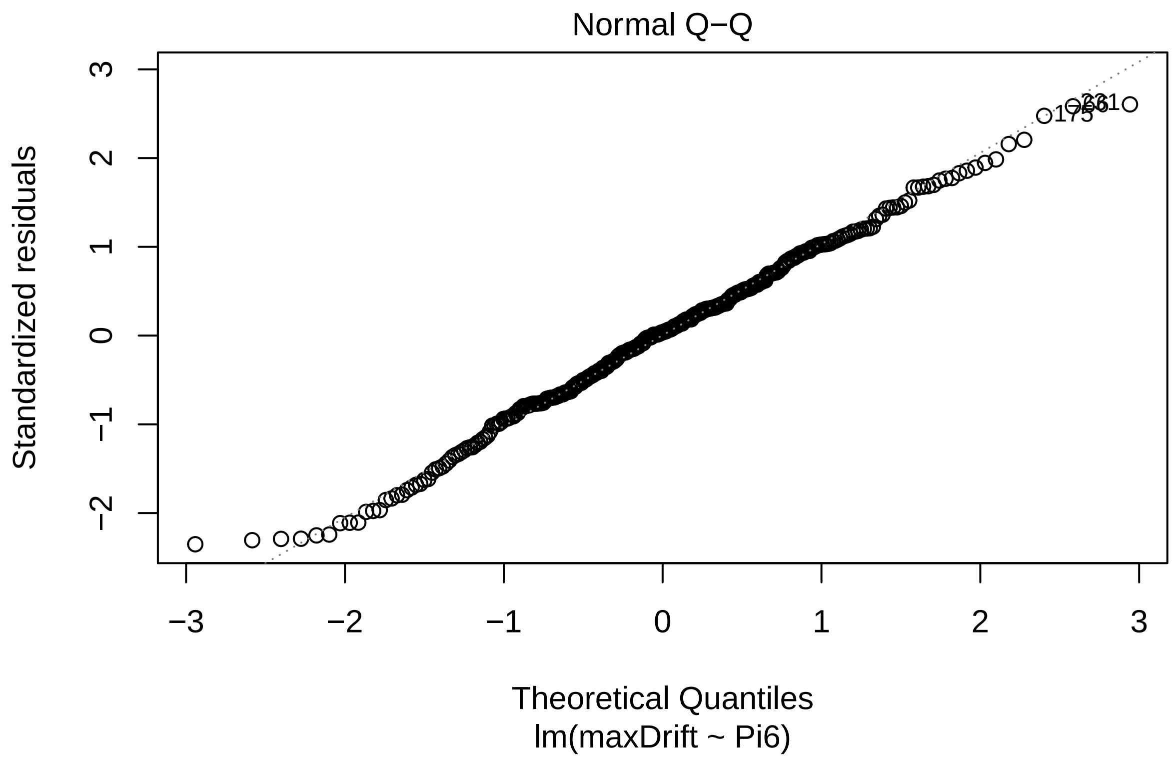
```
## F-statistic: 334.5 on 1 and 305 DF,  p-value: < 2.2e-16
```

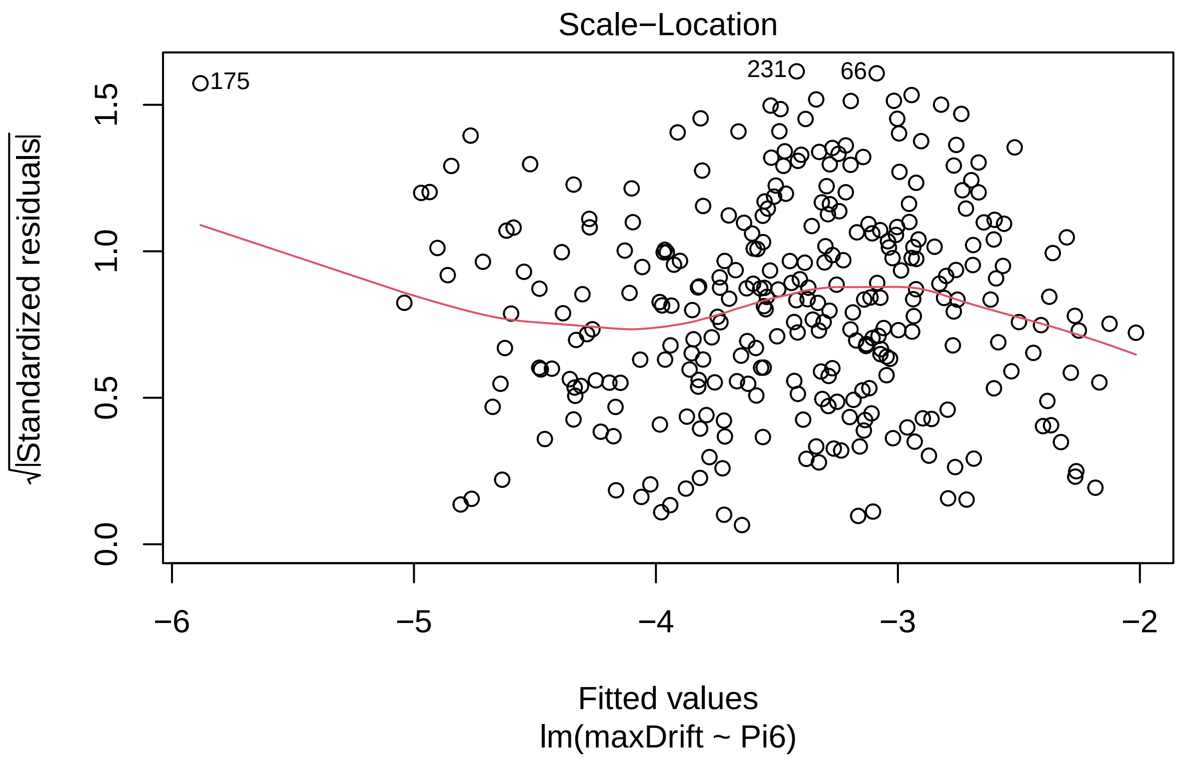
```r
confint(lnFitDrift6)
```

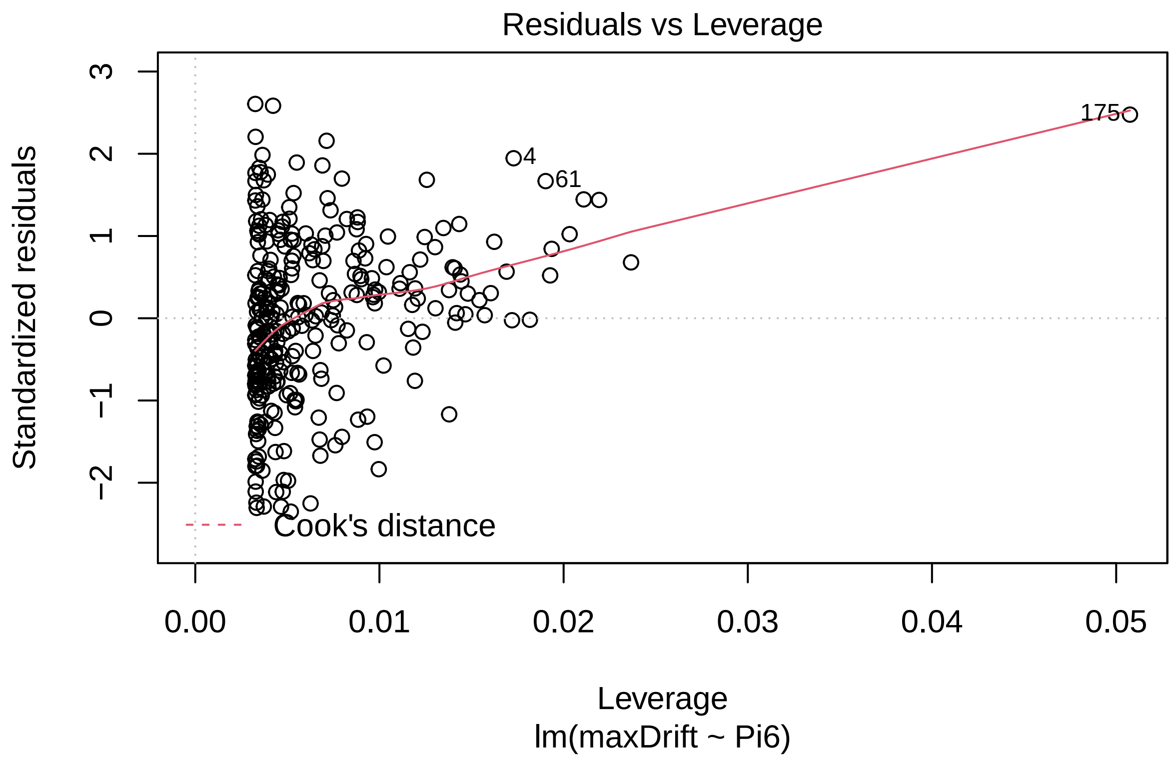
```
##                  2.5 %    97.5 %
## (Intercept) -3.677446 -3.534845
## Pi6          1.704666  2.115723
```

```r
plot(lnFitDrift6)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(maxDrift ~ Pi6)
175
231
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
175
231
66
−6
−5
−4
−3
−2
Fitted values
lm(maxDrift ~ Pi6)




```
sigma6 <- summary(lnFitDrift6)$sigma
```

PCA

Intensity measures

Run PCA on the various spectral accelerations.

```
im.vars <- subset(isol.full, select=c(
  ST1,ST2,GMSavg,GMS1,GMST1,GMST2,GMSTm
))

# perform PCA, centering and scaling the data
im.pca <- prcomp(im.vars, center = TRUE, scale. = TRUE)
```

Outlier detection

```
# # compute the distance from the origin given the first three PCs
# r2 <- im.pca$x[,1]^2 + im.pca$x[,2]^2 + im.pca$x[,3]^2
# # get the indices of the points sorted in decreasing distance from the origin
# r2 <- order(r2, decreasing=TRUE)
#
#
# plot(im.pca$x[,1], im.pca$x[,2], asp=1, col=point.col)
# points(im.pca$x[r2[1:3],1], im.pca$x[r2[1:3],2], col='red', pch=5)
#
```

```r
# # output outlier indices
# r2[1:3]
```

Linear models

Univariate

Look at one spectral acceleration in its performance of predicting maximum drift.

```r
# fit <- lm(GMSavg ~ maxDrift, data=isol.full)
# summary(fit)
#
# par(mfrow = c(2,2))
# plot(fit, col = point.col)
```

EHW standard errors

```r
# library(car)
# fit.hc0 = sqrt(diag(hccm(fit, type="hc0")))
# fit.hc1 = sqrt(diag(hccm(fit, type="hc1")))
# fit.hc2 = sqrt(diag(hccm(fit, type="hc2")))
# fit.hc3 = sqrt(diag(hccm(fit, type="hc3")))
# fit.hc4 = sqrt(diag(hccm(fit, type="hc4")))
# fit.coef = summary(fit)$coef
# tvalues = fit.coef [ ,1] /
# cbind(fit.coef[ ,2] , fit.hc0 , fit.hc1 ,
# fit.hc2 , fit.hc3 , fit.hc4)
# colnames(tvalues) = c("ols",  "hc0", "hc1", "hc2", "hc3", "hc4")
# round(tvalues , 2)
```

Probit regression

```r
# im.full <- subset(isol.full, select=c(
#   ST1,ST2,GMSavg,GMS1,GMST1,GMST2,GMSTm,collapse
# ))
#
# im.probit <- glm(collapse ~ ., family = binomial(link="probit"), data=im.full)
# summary(im.probit)
```