CPSC541_BYOD_Project

Yuting_Qiu

9/19/2019

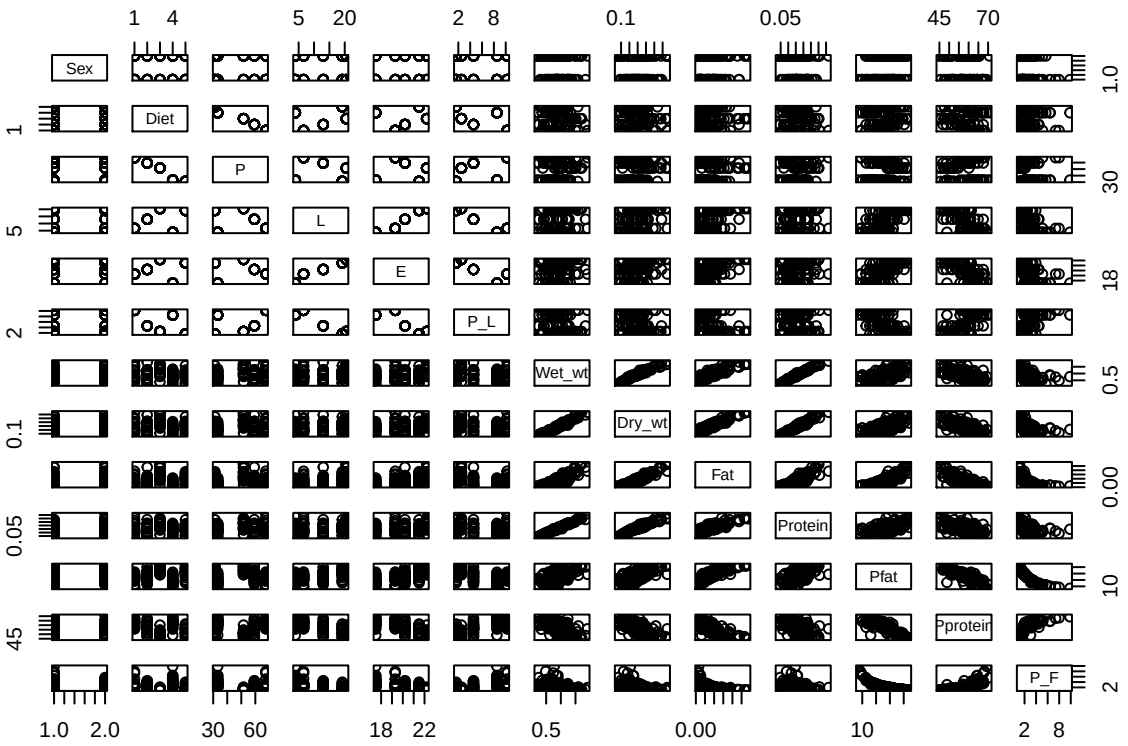
read in the data and basic processing

```
data1 <-read.csv("Moatt et al Data_S1.csv",h=T)
data2 <- data1[data1$Include == "y",] #only inlude =y
data2$Diet <- as.factor(data2$Diet)
data3 <- data2[-which(is.na(data2$Pprotein )),] # remove the nas
```

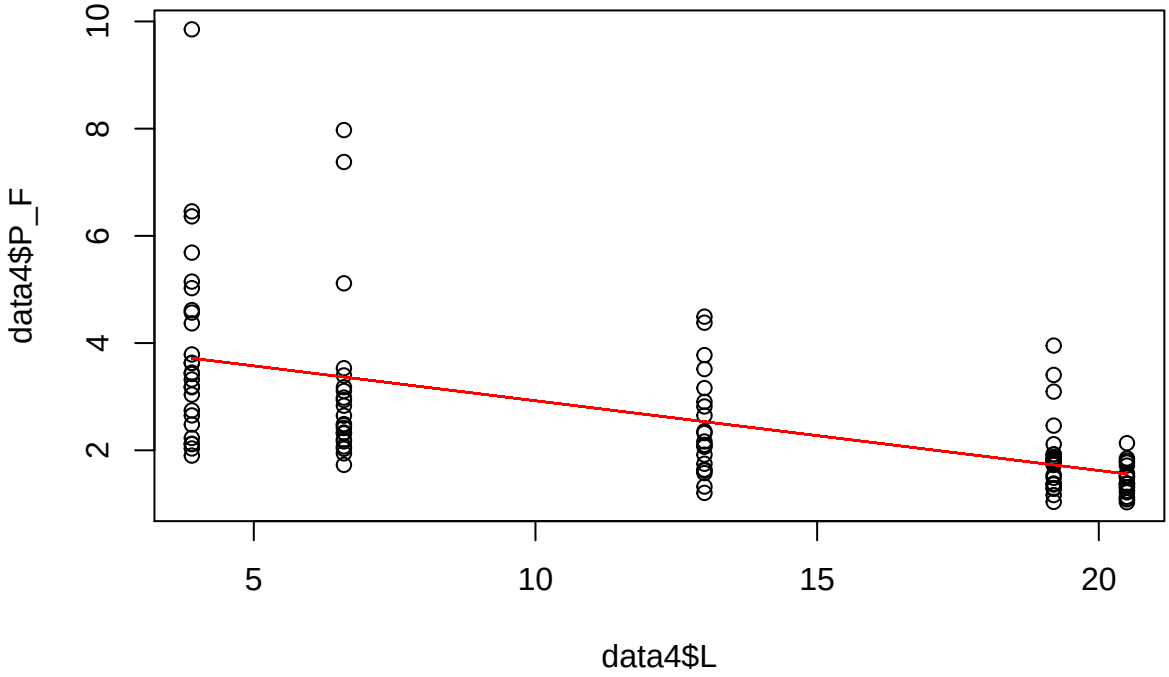
```
str(data3)
```

```
## 'data.frame':    113 obs. of  30 variables:
##  $ FID           : Factor w/ 116 levels "MT001","MT002",..: 25 26 27 28 31 29 30 34 33 32 ...
##  $ TANKID        : Factor w/ 116 levels "M11M","M12L",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Include       : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Sex           : Factor w/ 2 levels "f","m": 2 1 1 1 2 2 2 1 1 1 ...
##  $ Diet          : Factor w/ 5 levels "1","2","3","4",..: 1 2 2 2 3 3 3 4 4 4 ...
##  $ P             : num  67.5 59.3 59.3 59.3 51.6 51.6 51.6 33.2 33.2 33.2 ...
##  $ L             : num  6.6 13 13 13 20.5 20.5 20.5 3.9 3.9 3.9 ...
##  $ E             : num  19.3 20.2 20.2 20.2 22.2 22.2 22.2 17.5 17.5 17.5 ...
##  $ P_L           : num  10.23 4.56 4.56 4.56 2.52 ...
##  $ Tank          : Factor w/ 44 levels "M1","M11","M12",..: 2 3 3 3 4 4 4 5 5 5 ...
##  $ Family        : Factor w/ 9 levels "f102","f105",..: 4 4 1 8 8 4 1 8 1 4 ...
##  $ Wet_wt        : num  0.708 1.842 0.905 0.899 1.712 ...
##  $ Dry_wt        : num  0.197 0.587 0.296 0.277 0.659 ...
##  $ Fat_free_wt   : num  0.146 0.35 0.201 0.203 0.383 ...
##  $ Ashed_wt      : num  0.0206 0.0654 0.0483 0.0464 0.0823 0.0522 0.0493 0.0264 0.0469 0.0189 ...
##  $ Fat           : num  0.0508 0.2364 0.0943 0.0745 0.2754 ...
##  $ Protein       : num  0.125 0.285 0.153 0.156 0.301 ...
##  $ Pwater        : num  72.2 68.1 67.3 69.2 61.5 63.9 66 73.2 71.1 72.9 ...
##  $ Pfat          : num  25.8 40.3 31.9 26.9 41.8 35.8 33.3 24.3 12.5 14.4 ...
##  $ Pash          : num  10.5 11.1 16.4 16.7 12.5 ...
##  $ Pprotein      : num  63.7 48.6 51.8 56.4 45.7 ...
##  $ Energy_fat    : num  2.01 9.36 3.73 2.95 10.91 ...
##  $ Energy_protein: num  2.97 6.76 3.63 3.71 7.13 4.12 3.8 3.54 3.06 1.56 ...
##  $ Total_energy  : num  4.98 16.12 7.36 6.66 18.04 ...
##  $ P_F           : num  2.47 1.21 1.62 2.1 1.09 ...
##  $ Diff_P_F      : num  -7.76 -3.36 -2.94 -2.46 -1.42 ...
##  $ Testes_wt     : num  NA NA NA NA NA 0.0049 0.0074 NA NA NA ...
##  $ Wt_body       : num  NA 1.9 0.97 0.96 1.88 1.09 0.99 0.94 0.81 0.49 ...
##  $ Sperm_y       : Factor w/ 2 levels "n","y": 1 1 1 1 1 2 2 1 1 1 ...
##  $ P_end         : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 2 2 2 2 ...
```

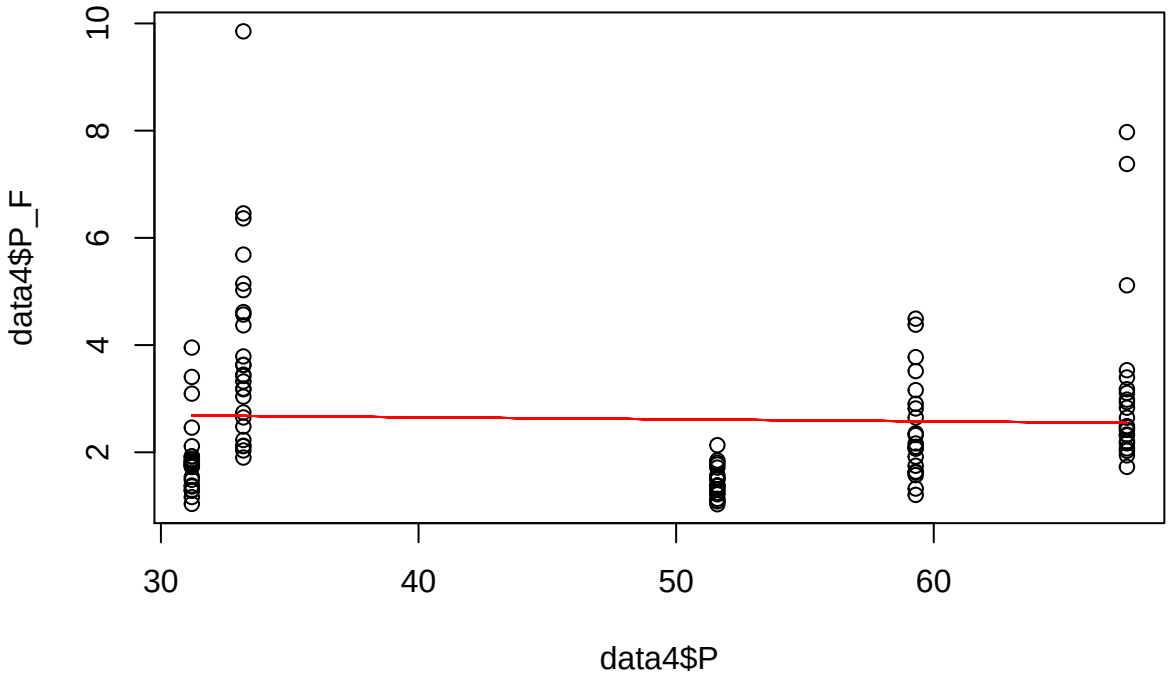
```
data4 <- data3[,c(4,5,6,7,8,9,12,13,16,17,19,21,25)] # the possible columns that we are interested in
pairs(data4) # making scatterplot matrices of the data
```



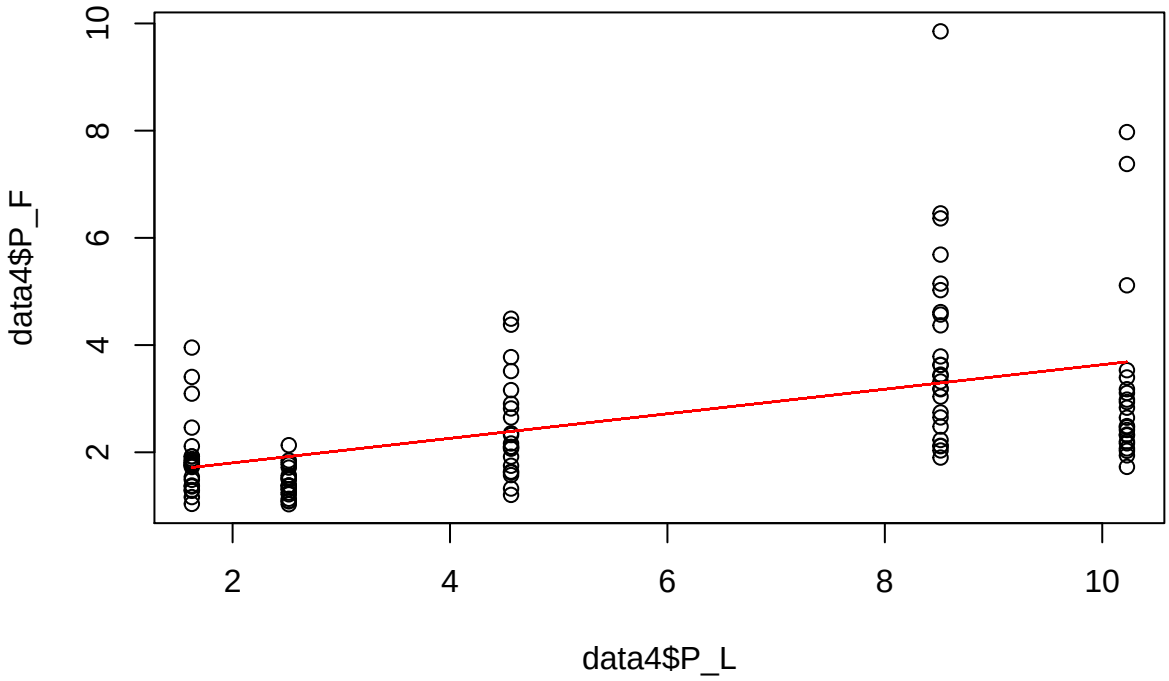

```
# fit some basic model to see the results
# effect on lipid intake on body protein:lipid intake

M_lipid <- lm(P_F ~ L, data4)
M_protein <- lm(P_F ~ P, data4)
M_PLratio <- lm(P_F ~ P_L, data4)


plot(data4$L, data4$P_F)
lines(data4$L, predict(M_lipid),col="red")
```




```
plot(data4$P, data4$P_F)
lines(data4$P, predict(M_protein),col="red")
```




```
plot(data4$P_L, data4$P_F)
lines(data4$P_L, predict(M_PLratio), col="red")
```




```
summary(M_lipid)
```

```
##
## Call:
## lm(formula = P_F ~ L, data = data4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8164 -0.6154 -0.2126  0.1966  6.1348
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.22488    0.24341  17.357  < 2e-16 ***
## L           -0.13008    0.01738  -7.483 1.83e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.235 on 111 degrees of freedom
## Multiple R-squared:  0.3353, Adjusted R-squared:  0.3293
## F-statistic:    56 on 1 and 111 DF,  p-value: 1.833e-11
```

```
summary(M_protein)
```

```
##
```

```
## Call:
## lm(formula = P_F ~ P, data = data4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6487 -0.9410 -0.4771  0.5576  7.1738
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.803960   0.488356   5.742 8.28e-08 ***
## P           -0.003778   0.009822  -0.385    0.701
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.514 on 111 degrees of freedom
## Multiple R-squared:  0.001331,   Adjusted R-squared:  -0.007666
## F-statistic: 0.1479 on 1 and 111 DF,  p-value: 0.7013
```

```
summary(M_PLratio)
```

```
##
## Call:
## lm(formula = P_F ~ P_L, data = data4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9589 -0.7494 -0.2898  0.2116  6.5582
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.34443    0.23665   5.681 1.09e-07 ***
## P_L          0.22903    0.03626   6.316 5.72e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.299 on 111 degrees of freedom
## Multiple R-squared:  0.2644, Adjusted R-squared:  0.2577
## F-statistic: 39.89 on 1 and 111 DF,  p-value: 5.717e-09
```

evaluate the diet on body composion

```
data3$P <- as.numeric(as.character(data3$P))
data3$L <- as.numeric(as.character(data3$L))
data3$E <- as.numeric(as.character(data3$E))
data3$P_F <- as.numeric(as.character(data3$P_F))

m.diet1 <- lm(P_F ~ Diet, data = data3)
m.diet2 <- lm(P_F ~ P + L + E + P_L, data = data3)
m.diet3 <- lm(P_F ~  P_L, data = data3)
m.diet4 <- lm(P_F ~  P_L + L, data = data3)
m.diet5 <- lm(P_F ~  P_L + P, data = data3)
m.diet6 <- lm(P_F ~  P_L + P + L, data = data3)
m.diet7 <- lm(P_F ~  Diet + Sex, data = data3)
```

```r
m.diet8 <- lm(P_F ~  Diet + Sex + Tank, data = data3)
m.diet9 <- lm(P_F ~  Diet + Tank, data = data3)
m.diet9 <- lm(P_F ~  E, data = data3)
m.diet10 <- lm(P_F ~  E + P_L, data = data3)
m.diet11 <- lm(E ~  P_L, data = data3)
m.diet12 <- lm(P_F ~  L, data = data3)
m.diet13 <- lm(P_F ~  P, data = data3)


anova(m.diet1)
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Diet        4  88.226 22.0566   14.31 2.096e-09 ***
## Residuals 108 166.466  1.5414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m.diet2)
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value   Pr(>F)
## P           1   0.339   0.339  0.2199   0.6400
## L           1  87.747  87.747 56.9289 1.51e-11 ***
## E           1   0.055   0.055  0.0355   0.8508
## P_L         1   0.085   0.085  0.0553   0.8146
## Residuals 108 166.466   1.541
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m.diet3)
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## P_L         1  67.329  67.329  39.888 5.717e-09 ***
## Residuals 111 187.363   1.688
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m.diet4)
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value    Pr(>F)
```

```
## P_L         1  67.329  67.329  44.305 1.142e-09 ***
## L           1  20.197  20.197  13.290  0.000409 ***
## Residuals 110 167.166   1.520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m.diet5)
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## P_L         1  67.329  67.329  42.847 1.947e-09 ***
## P           1  14.512  14.512   9.235  0.002967 **
## Residuals 110 172.851   1.571
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m.diet6)
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## P_L         1  67.329  67.329 44.0773 1.275e-09 ***
## P           1  14.512  14.512  9.5001  0.002602 **
## L           1   6.351   6.351  4.1580  0.043859 *
## Residuals 109 166.500   1.528
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m.diet9)
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value   Pr(>F)
## E           1  87.518  87.518   58.11 9.02e-12 ***
## Residuals 111 167.174   1.506
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m.diet10)
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## E           1  87.518  87.518 57.7147 1.071e-11 ***
## P_L         1   0.371   0.371  0.2444     0.622
## Residuals 110 166.803   1.516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m.diet11)
```

```
## Analysis of Variance Table
##
## Response: E
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## P_L         1 230.265 230.265  270.39 < 2.2e-16 ***
## Residuals 111  94.528   0.852
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(m.diet1)
```

```
##
## Call:
## lm(formula = P_F ~ Diet, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9835 -0.6784 -0.1642  0.2677  5.9677
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.1469     0.2647  11.889  < 2e-16 ***
## Diet2        -0.6788     0.3788  -1.792 0.075905 .
## Diet3        -1.7003     0.3836  -4.433 2.24e-05 ***
## Diet4         0.7378     0.3596   2.051 0.042650 *
## Diet5        -1.2491     0.3664  -3.409 0.000919 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.242 on 108 degrees of freedom
## Multiple R-squared:  0.3464, Adjusted R-squared:  0.3222
## F-statistic: 14.31 on 4 and 108 DF,  p-value: 2.096e-09
```

```r
summary(m.diet2)
```

```
##
## Call:
## lm(formula = P_F ~ P + L + E + P_L, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9835 -0.6784 -0.1642  0.2677  5.9677
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.31109   25.57927   0.051    0.959
## P           -0.01128    0.02410  -0.468    0.641
## L           -0.24789    0.65022  -0.381    0.704
## E            0.27262    1.83145   0.149    0.882
## P_L         -0.10058    0.42789  -0.235    0.815
```

```
##
## Residual standard error: 1.242 on 108 degrees of freedom
## Multiple R-squared:  0.3464, Adjusted R-squared:  0.3222
## F-statistic: 14.31 on 4 and 108 DF,  p-value: 2.096e-09
```

```
summary(m.diet3)
```

```
##
## Call:
## lm(formula = P_F ~ P_L, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9589 -0.7494 -0.2898  0.2116  6.5582
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.34443    0.23665   5.681 1.09e-07 ***
## P_L          0.22903    0.03626   6.316 5.72e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.299 on 111 degrees of freedom
## Multiple R-squared:  0.2644, Adjusted R-squared:  0.2577
## F-statistic: 39.89 on 1 and 111 DF,  p-value: 5.717e-09
```

```
summary(m.diet4)
```

```
##
## Call:
## lm(formula = P_F ~ P_L + L, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9445 -0.6575 -0.1965  0.2622  6.0067
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.60030    1.18879   4.711 7.25e-06 ***
## P_L         -0.12035    0.10183  -1.182 0.239779
## L           -0.18721    0.05135  -3.646 0.000409 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.233 on 110 degrees of freedom
## Multiple R-squared:  0.3437, Adjusted R-squared:  0.3317
## F-statistic:  28.8 on 2 and 110 DF,  p-value: 8.754e-11
```

```
summary(m.diet5)
```

```
##
## Call:
```

```
## lm(formula = P_F ~ P_L + P, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8944 -0.6247 -0.2363  0.1607  6.0568
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.374806   0.408774   5.810 6.18e-08 ***
## P_L          0.270298   0.037532   7.202 7.81e-11 ***
## P           -0.026514   0.008725  -3.039  0.00297 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.254 on 110 degrees of freedom
## Multiple R-squared:  0.3213, Adjusted R-squared:  0.309
## F-statistic: 26.04 on 2 and 110 DF,  p-value: 5.51e-10
```

```
summary(m.diet6)
```

```
##
## Call:
## lm(formula = P_F ~ P_L + P + L, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9952 -0.6657 -0.1613  0.2618  5.9560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.112948   1.401993   3.647 0.000409 ***
## P_L         -0.041383   0.157268  -0.263 0.792940
## P           -0.008207   0.012434  -0.660 0.510582
## L           -0.151738   0.074414  -2.039 0.043859 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 109 degrees of freedom
## Multiple R-squared:  0.3463, Adjusted R-squared:  0.3283
## F-statistic: 19.25 on 3 and 109 DF,  p-value: 4.363e-10
```

```
summary(m.diet7) # Adjusted R-squared:  0.3428
```

```
##
## Call:
## lm(formula = P_F ~ Diet + Sex, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9758 -0.5996 -0.2217  0.2020  5.7626
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   3.3892     0.2852  11.884  < 2e-16 ***
## Diet2        -0.6903     0.3730  -1.851 0.066962 .
## Diet3        -1.6518     0.3784  -4.365 2.94e-05 ***
## Diet4         0.7005     0.3546   1.975 0.050789 .
## Diet5        -1.2491     0.3608  -3.462 0.000773 ***
## Sexm         -0.4847     0.2315  -2.094 0.038667 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.223 on 107 degrees of freedom
## Multiple R-squared:  0.3721, Adjusted R-squared:  0.3428
## F-statistic: 12.68 on 5 and 107 DF,  p-value: 1.112e-09
```

```
summary(m.diet9)
```

```
##
## Call:
## lm(formula = P_F ~ E, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0383 -0.6191 -0.1843  0.2717  5.9129
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.0236     1.3691   9.513 4.75e-16 ***
## E            -0.5191     0.0681  -7.623 9.02e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.227 on 111 degrees of freedom
## Multiple R-squared:  0.3436, Adjusted R-squared:  0.3377
## F-statistic: 58.11 on 1 and 111 DF,  p-value: 9.02e-12
```

```
summary(m.diet11)
```

```
##
## Call:
## lm(formula = E ~ P_L, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2949 -0.2685 -0.2122  0.8656  1.2312
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.40049    0.16809  133.26   <2e-16 ***
## P_L         -0.42355    0.02576  -16.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9228 on 111 degrees of freedom
## Multiple R-squared:  0.709,  Adjusted R-squared:  0.7063
## F-statistic: 270.4 on 1 and 111 DF,  p-value: < 2.2e-16
```

```r
summary(m.diet12)
```

```
##
## Call:
## lm(formula = P_F ~ L, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8164 -0.6154 -0.2126  0.1966  6.1348
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.22488    0.24341  17.357  < 2e-16 ***
## L           -0.13008    0.01738  -7.483 1.83e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.235 on 111 degrees of freedom
## Multiple R-squared:  0.3353, Adjusted R-squared:  0.3293
## F-statistic:    56 on 1 and 111 DF,  p-value: 1.833e-11
```

```r
summary(m.diet13)
```

```
##
## Call:
## lm(formula = P_F ~ P, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6487 -0.9410 -0.4771  0.5576  7.1738
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.803960   0.488356   5.742 8.28e-08 ***
## P           -0.003778   0.009822  -0.385    0.701
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.514 on 111 degrees of freedom
## Multiple R-squared:  0.001331,   Adjusted R-squared:  -0.007666
## F-statistic: 0.1479 on 1 and 111 DF,  p-value: 0.7013
```

```r
summary(lm(E~L, data = data3)) # Adjusted R-squared:  0.9465
```

```
##
## Call:
## lm(formula = E ~ L, data = data3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45894 -0.23585 -0.00534  0.14324  0.67455
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.996197   0.077659  218.86   <2e-16 ***
## L            0.246857   0.005546   44.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.394 on 111 degrees of freedom
## Multiple R-squared:  0.9469, Adjusted R-squared:  0.9465
## F-statistic:  1981 on 1 and 111 DF,  p-value: < 2.2e-16
```

```
summary(lm(P_F~L, data = data3)) # Adjusted R-squared:  0.3293
```

```
##
## Call:
## lm(formula = P_F ~ L, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8164 -0.6154 -0.2126  0.1966  6.1348
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.22488    0.24341  17.357  < 2e-16 ***
## L           -0.13008    0.01738  -7.483 1.83e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.235 on 111 degrees of freedom
## Multiple R-squared:  0.3353, Adjusted R-squared:  0.3293
## F-statistic:    56 on 1 and 111 DF,  p-value: 1.833e-11
```

```
summary(lm(P_F~P, data = data3)) # Adjusted R-squared:  -0.007666
```

```
##
## Call:
## lm(formula = P_F ~ P, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6487 -0.9410 -0.4771  0.5576  7.1738
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.803960   0.488356   5.742 8.28e-08 ***
## P           -0.003778   0.009822  -0.385    0.701
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.514 on 111 degrees of freedom
## Multiple R-squared:  0.001331,   Adjusted R-squared:  -0.007666
## F-statistic: 0.1479 on 1 and 111 DF,  p-value: 0.7013
```

```r
summary(lm(P_F~E, data = data3)) # Adjusted R-squared:  0.3377
```

```
##
## Call:
## lm(formula = P_F ~ E, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0383 -0.6191 -0.1843  0.2717  5.9129
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.0236     1.3691   9.513 4.75e-16 ***
## E            -0.5191     0.0681  -7.623 9.02e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.227 on 111 degrees of freedom
## Multiple R-squared:  0.3436, Adjusted R-squared:  0.3377
## F-statistic: 58.11 on 1 and 111 DF,  p-value: 9.02e-12
```

```r
summary(lm(P_F~P_L, data = data3)) # Adjusted R-squared:  0.2577
```

```
##
## Call:
## lm(formula = P_F ~ P_L, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9589 -0.7494 -0.2898  0.2116  6.5582
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.34443    0.23665   5.681 1.09e-07 ***
## P_L          0.22903    0.03626   6.316 5.72e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.299 on 111 degrees of freedom
## Multiple R-squared:  0.2644, Adjusted R-squared:  0.2577
## F-statistic: 39.89 on 1 and 111 DF,  p-value: 5.717e-09
```

```r
anova(m.diet3, m.diet4)
```

```
## Analysis of Variance Table
##
## Model 1: P_F ~ P_L
## Model 2: P_F ~ P_L + L
##   Res.Df    RSS Df Sum of Sq     F   Pr(>F)
## 1    111 187.36
## 2    110 167.17  1    20.198 13.29 0.000409 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m.diet3, m.diet4)
```

```
## Analysis of Variance Table
##
## Model 1: P_F ~ P_L
## Model 2: P_F ~ P_L + L
##   Res.Df    RSS Df Sum of Sq     F   Pr(>F)
## 1    111 187.36
## 2    110 167.17  1    20.198 13.29 0.000409 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m.diet3, m.diet6) # it is important both P and L amonunt in the diet and the P_L ratio
```

```
## Analysis of Variance Table
##
## Model 1: P_F ~ P_L
## Model 2: P_F ~ P_L + P + L
##   Res.Df    RSS Df Sum of Sq     F   Pr(>F)
## 1    111 187.36
## 2    109 166.50  2    20.863 6.829 0.001606 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m.diet1, m.diet7) # beside diet, sex is also an important facotor
```

```
## Analysis of Variance Table
##
## Model 1: P_F ~ Diet
## Model 2: P_F ~ Diet + Sex
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    108 166.47
## 2    107 159.92  1    6.5502 4.3828 0.03867 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m.diet7, m.diet8) # tank is not signigficant with sex and diet
```

```
## Analysis of Variance Table
##
## Model 1: P_F ~ Diet + Sex
## Model 2: P_F ~ Diet + Sex + Tank
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    107 159.92
## 2     69 105.75 38    54.167 0.9301 0.5886
```

```r
anova(m.diet1, m.diet9) # tank is not signficant
```

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

```
## Model 1: P_F ~ Diet
## Model 2: P_F ~ E
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    108 166.47
## 2    111 167.17 -3  -0.70809 0.1531 0.9275
```

```
# test for collinearity in the P, L and P_L
summary(lm(P_F ~ P + L + P_L, data = data3))
```

```
##
## Call:
## lm(formula = P_F ~ P + L + P_L, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9952 -0.6657 -0.1613  0.2618  5.9560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.112948   1.401993   3.647 0.000409 ***
## P           -0.008207   0.012434  -0.660 0.510582
## L           -0.151738   0.074414  -2.039 0.043859 *
## P_L         -0.041383   0.157268  -0.263 0.792940
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 109 degrees of freedom
## Multiple R-squared:  0.3463, Adjusted R-squared:  0.3283
## F-statistic: 19.25 on 3 and 109 DF,  p-value: 4.363e-10
```

```
summary(lm(P_F ~  L + P_L + P, data = data3))
```

```
##
## Call:
## lm(formula = P_F ~ L + P_L + P, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9952 -0.6657 -0.1613  0.2618  5.9560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.112948   1.401993   3.647 0.000409 ***
## L           -0.151738   0.074414  -2.039 0.043859 *
## P_L         -0.041383   0.157268  -0.263 0.792940
## P           -0.008207   0.012434  -0.660 0.510582
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 109 degrees of freedom
## Multiple R-squared:  0.3463, Adjusted R-squared:  0.3283
## F-statistic: 19.25 on 3 and 109 DF,  p-value: 4.363e-10
```

```r
summary(lm(P_F ~ P_L + P + L , data = data3))
```

```
##
## Call:
## lm(formula = P_F ~ P_L + P + L, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9952 -0.6657 -0.1613  0.2618  5.9560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.112948   1.401993   3.647 0.000409 ***
## P_L         -0.041383   0.157268  -0.263 0.792940
## P           -0.008207   0.012434  -0.660 0.510582
## L           -0.151738   0.074414  -2.039 0.043859 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 109 degrees of freedom
## Multiple R-squared:  0.3463, Adjusted R-squared:  0.3283
## F-statistic: 19.25 on 3 and 109 DF,  p-value: 4.363e-10
```

re-fit, try poisson errors

```r
m_lipid.2 <- glm(P_F ~ L, data = data4, family = poisson(link = identity)) # non-interger, cannot run p
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.468504

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.205584

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.622481

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.099329

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.092593

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.380952

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.534928

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.650089

## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.147410

## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.615385

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.486392
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.166902
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.071878
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.106599
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.423622
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.743132
## Warning in dpois(y, mu, log = TRUE): non-integer x = 6.363636
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.953271
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.405914
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.460465
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.296647
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.094118
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.175926
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.901198
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.164345
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.761354
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.037417
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.812785
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.788136
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.568528
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.714396
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.805882
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.748981
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.323387
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.904762
```

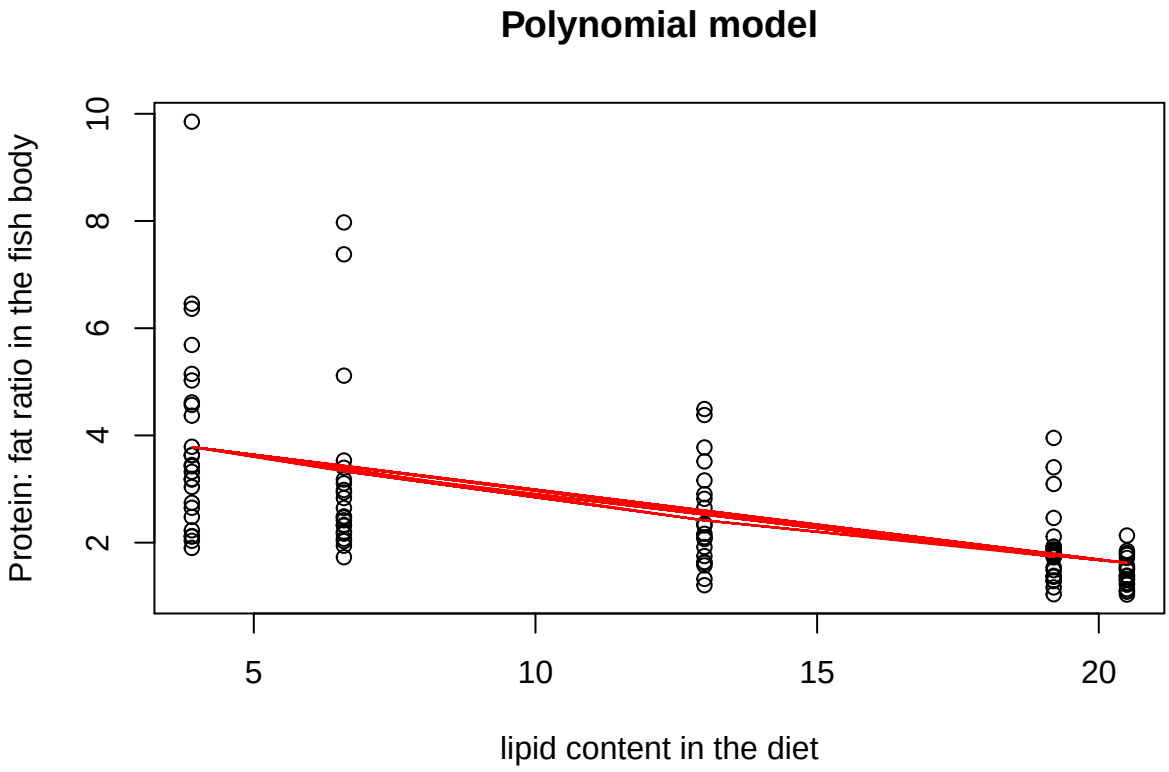
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.330561
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.489107
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.531496
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.923345
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.812371
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.763911
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.281507
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.121101
## Warning in dpois(y, mu, log = TRUE): non-integer x = 6.457143
## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.024876
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.031043
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.215513
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.246590
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.067138
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.380392
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.492386
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.942359
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.416275
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.316062
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.164742
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.750547
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.923935
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.113333
## Warning in dpois(y, mu, log = TRUE): non-integer x = 9.852349
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.630178
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.368421
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.132480
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.381484
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.132530
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.574422
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.515012
## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.114754
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.177033
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.028195
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.199812
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.831761
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.982759
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.638914
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.321199
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.098404
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.507792
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.369376
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.487854
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.034381
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.628842
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.645707
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.727852
## Warning in dpois(y, mu, log = TRUE): non-integer x = 7.378049
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.396040
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.918367
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.775401
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.648810
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.038339
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.188119
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.380468
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.937337
## Warning in dpois(y, mu, log = TRUE): non-integer x = 7.972973
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.085666
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.771451
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.570184
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.229525
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.480649
## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.687500
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.540894
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.873926
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.781513
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.842054
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.362084
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.724771
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.113905
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.448819
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.320359
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.313302
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.311430
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.847737
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.353591
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.817323
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.160000
```

try poly-nominal

```
m_lipid.3 <- lm(P_F ~ poly(L,2), data4)
plot(data4$L, data4$P_F, xlab = "lipid content in the diet", ylab = "Protein: fat ratio in the fish body
lines(data4$L, predict(m_lipid.3),col="red")
```

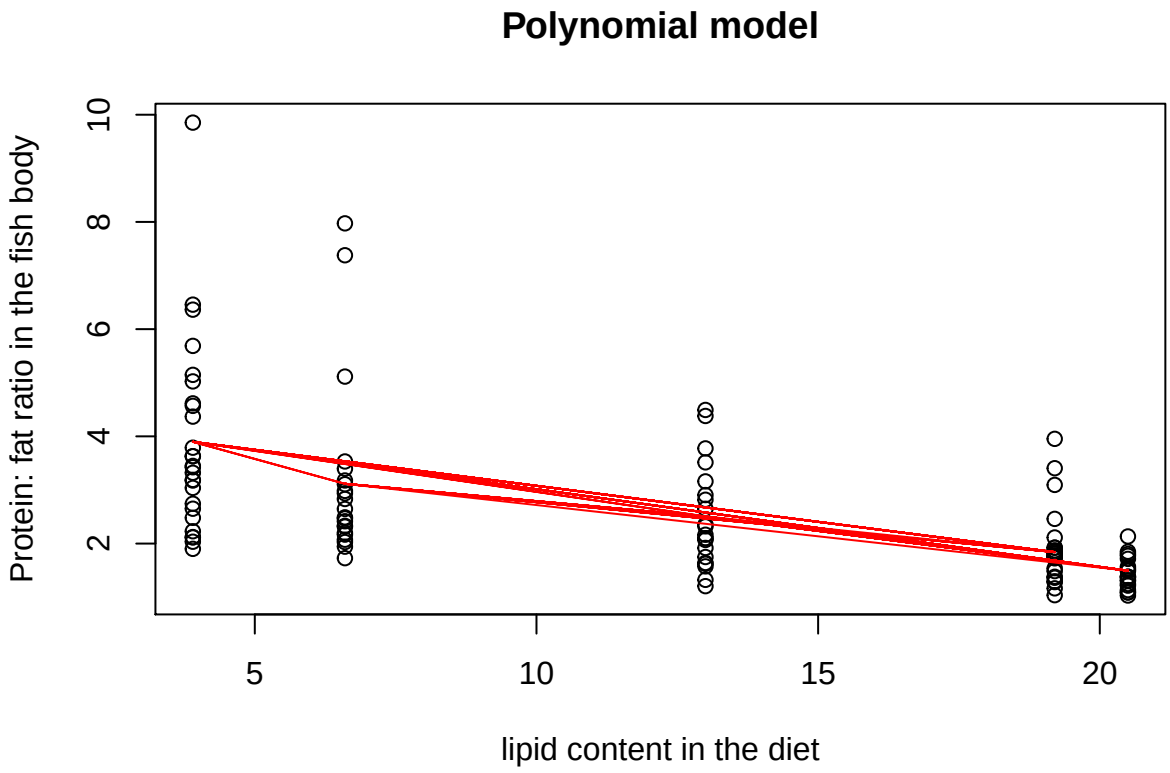



```
summary(m_lipid.3)
```

```
##
## Call:
## lm(formula = P_F ~ poly(L, 2), data = data4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8835 -0.6087 -0.2406  0.1865  6.0677
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.6243     0.1165  22.522  < 2e-16 ***
## poly(L, 2)1  -9.2414     1.2386  -7.461 2.13e-11 ***
## poly(L, 2)2   0.7227     1.2386   0.583    0.561
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.239 on 110 degrees of freedom
## Multiple R-squared:  0.3374, Adjusted R-squared:  0.3253
## F-statistic:    28 on 2 and 110 DF,  p-value: 1.479e-10
```

```
m_lipid.4 <- lm(P_F ~ poly(L,3), data4)
plot(data4$L, data4$P_F, xlab = "lipid content in the diet", ylab = "Protein: fat ratio in the fish body
lines(data4$L, predict(m_lipid.4),col="red")
```




```
summary(m_lipid.4)
```

```
##
## Call:
## lm(formula = P_F ~ poly(L, 3), data = data4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9966 -0.6464 -0.1807  0.2775  5.9545
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.6243     0.1163  22.561  < 2e-16 ***
## poly(L, 3)1  -9.2414     1.2365  -7.474 2.07e-11 ***
## poly(L, 3)2   0.7227     1.2365   0.584    0.560
## poly(L, 3)3  -1.4563     1.2365  -1.178    0.241
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 109 degrees of freedom
## Multiple R-squared:  0.3457, Adjusted R-squared:  0.3277
## F-statistic:  19.2 on 3 and 109 DF,  p-value: 4.572e-10
```

```
anova(m_lipid.3, m_lipid.4)
```

```
## Analysis of Variance Table
##
## Model 1: P_F ~ poly(L, 2)
## Model 2: P_F ~ poly(L, 3)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    110 168.77
## 2    109 166.65  1    2.1209 1.3872 0.2414
```

```
# not significant, p is large, complicate model is not necessary better than the simple model
anova(m_lipid.3, M_lipid)
```

```
## Analysis of Variance Table
##
## Model 1: P_F ~ poly(L, 2)
## Model 2: P_F ~ L
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    110 168.77
## 2    111 169.29 -1  -0.52232 0.3404 0.5608
```

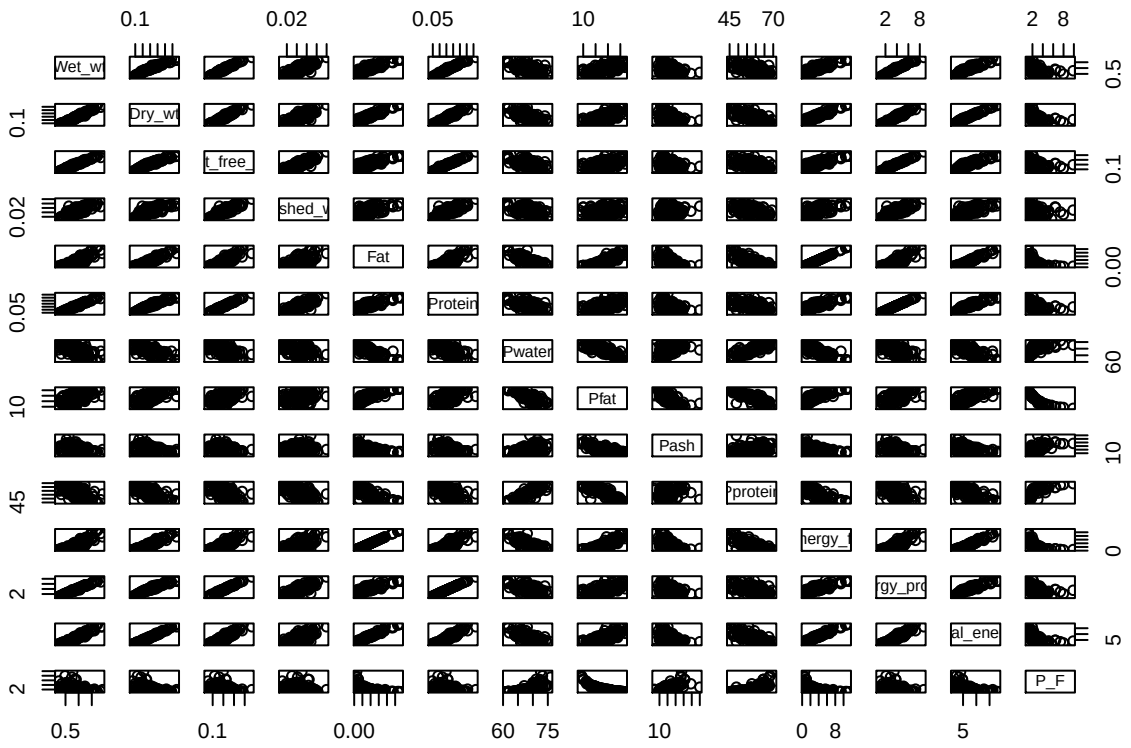
```
# not signifciant, a simple model is good enough
# but we can still see the influence of lipid intake on the protein_ratio for the fish. more lipid inta
```

look at the body componient

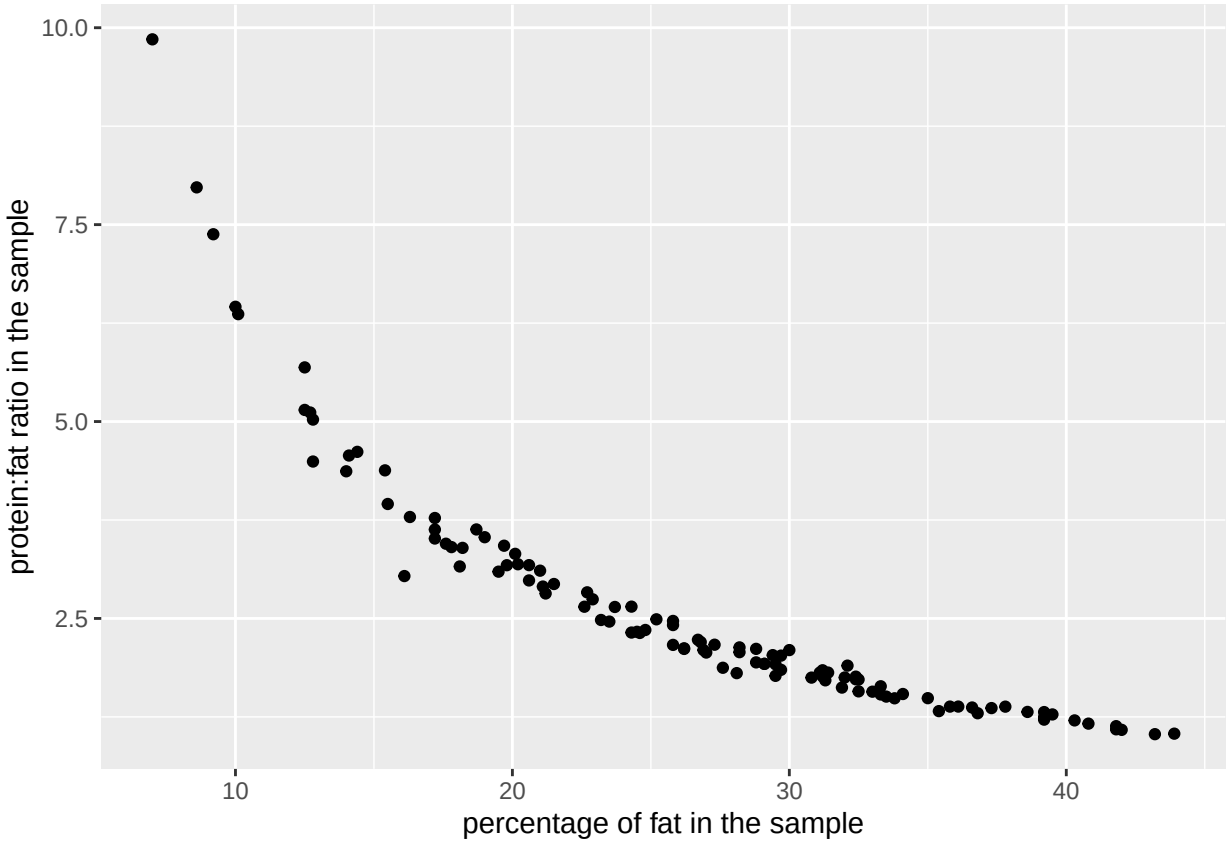
```
data5 <- data3[,12:25]
str(data5)
```

```
## 'data.frame':    113 obs. of  14 variables:
##  $ Wet_wt        : num  0.708 1.842 0.905 0.899 1.712 ...
##  $ Dry_wt        : num  0.197 0.587 0.296 0.277 0.659 ...
##  $ Fat_free_wt   : num  0.146 0.35 0.201 0.203 0.383 ...
##  $ Ashed_wt      : num  0.0206 0.0654 0.0483 0.0464 0.0823 0.0522 0.0493 0.0264 0.0469 0.0189 ...
##  $ Fat           : num  0.0508 0.2364 0.0943 0.0745 0.2754 ...
##  $ Protein       : num  0.125 0.285 0.153 0.156 0.301 ...
##  $ Pwater        : num  72.2 68.1 67.3 69.2 61.5 63.9 66 73.2 71.1 72.9 ...
##  $ Pfat          : num  25.8 40.3 31.9 26.9 41.8 35.8 33.3 24.3 12.5 14.4 ...
##  $ Pash          : num  10.5 11.1 16.4 16.7 12.5 ...
##  $ Pprotein      : num  63.7 48.6 51.8 56.4 45.7 ...
##  $ Energy_fat    : num  2.01 9.36 3.73 2.95 10.91 ...
##  $ Energy_protein: num  2.97 6.76 3.63 3.71 7.13 4.12 3.8 3.54 3.06 1.56 ...
##  $ Total_energy  : num  4.98 16.12 7.36 6.66 18.04 ...
##  $ P_F           : num  2.47 1.21 1.62 2.1 1.09 ...
```

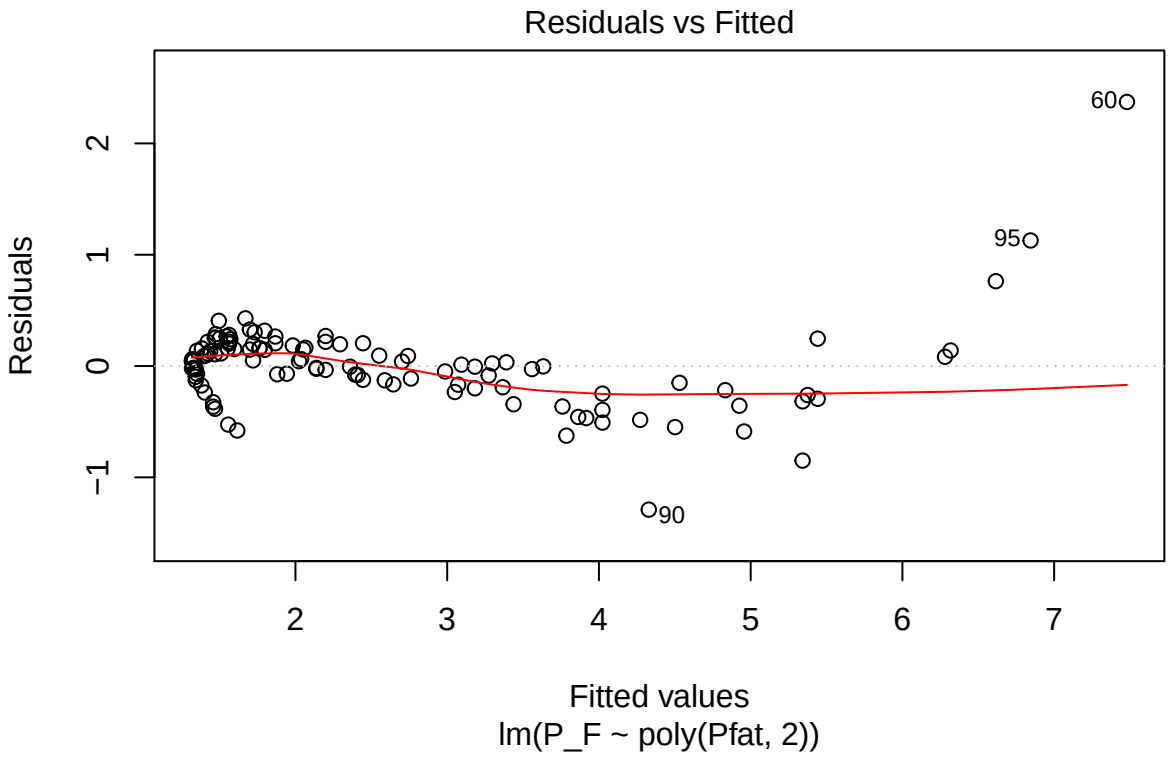
```
pairs(data5)
```

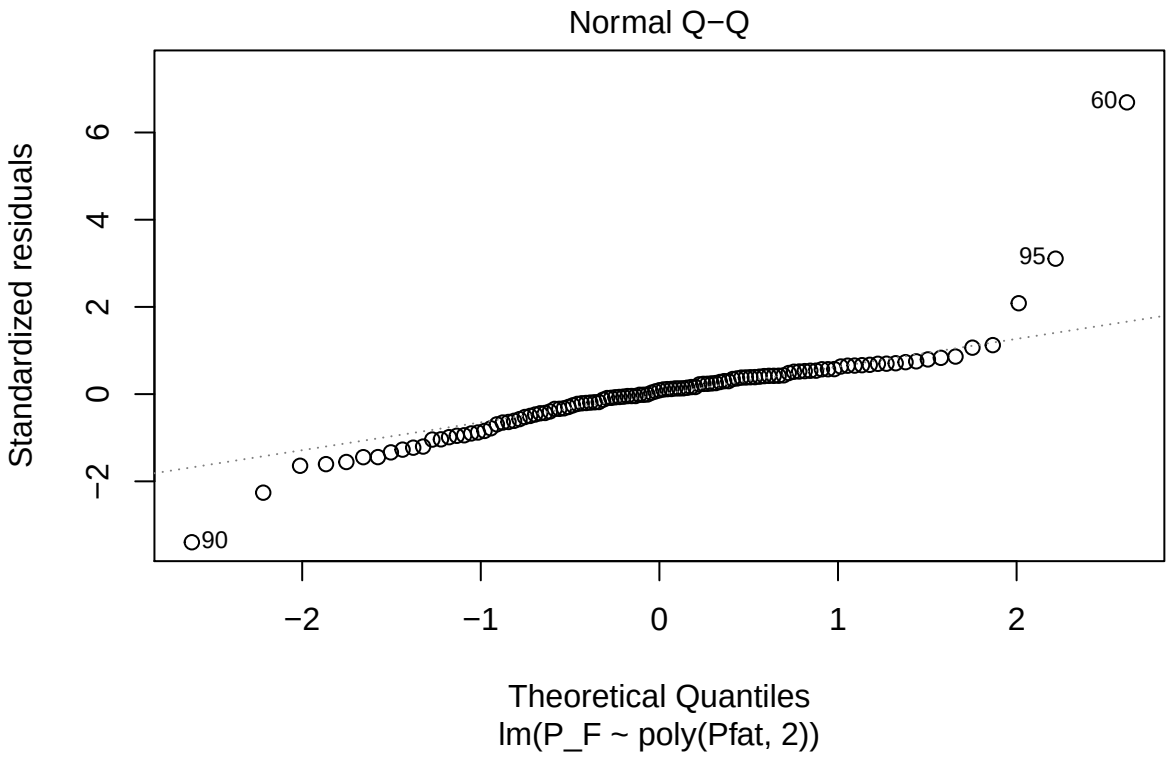


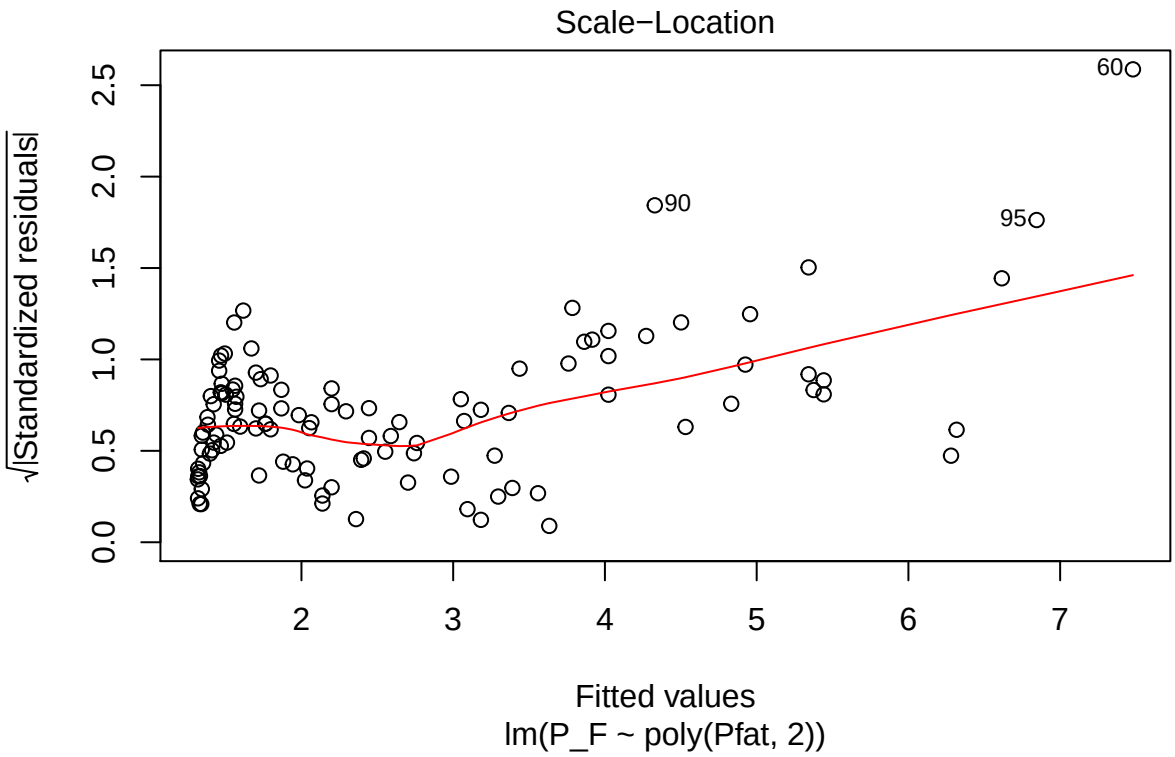

```
# what can explain protein:fat ration in the sample
# fat percentage first, looks like ploy normal

library(ggplot2)
p1 = ggplot(data5, aes(x=Pfat, y=P_F))
p1 = p1 + geom_point() + ylab("protein:fat ratio in the sample") +
  xlab("percentage of fat in the sample")
p1
```

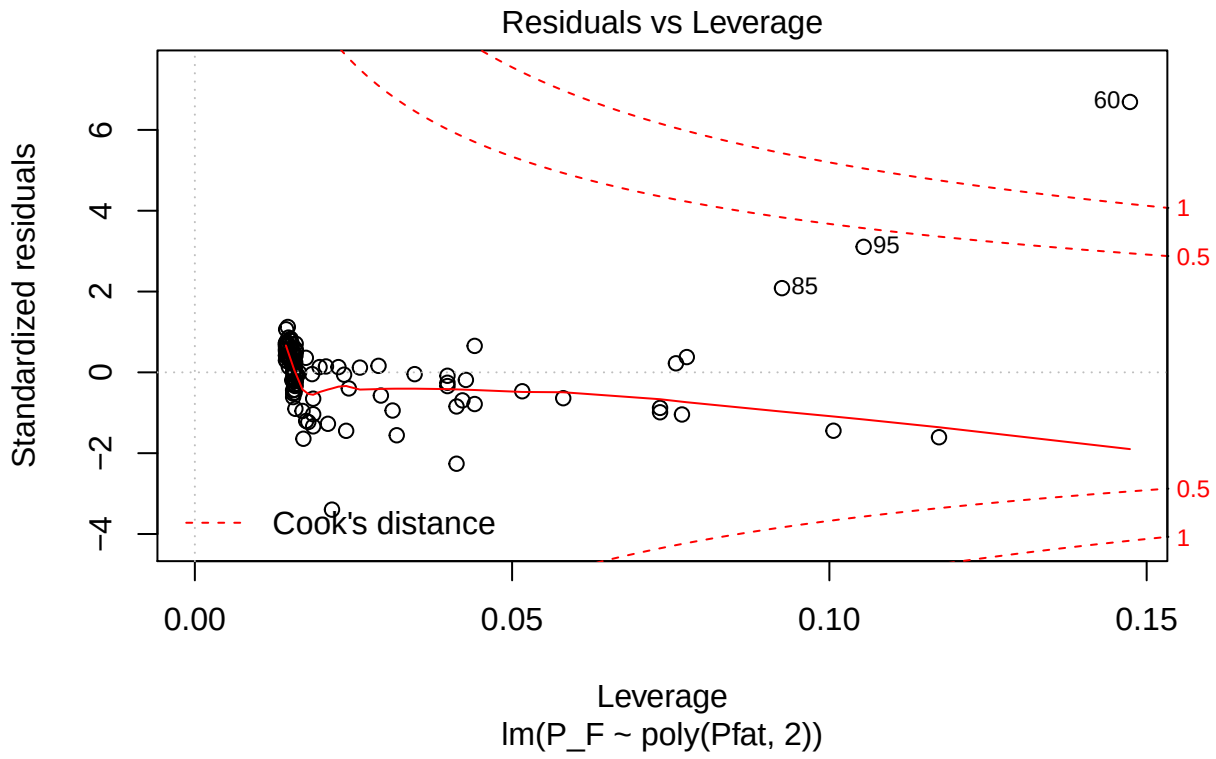



```
# fit polynomial to the P:L sample and
fit1 <- lm(P_F~poly(Pfat,2), data5)
plot(fit1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(P_F ~ poly(Pfat, 2))
60
95
90
2
1
0
−1
2
3
4
5
6
7


Normal Q−Q
Standardized residuals
6
4
2
0
−2
60
95
90
−2
−1
0
1
2
Theoretical Quantiles
lm(P_F ~ poly(Pfat, 2))


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
2
3
4
5
6
7
60
90
95
Fitted values
lm(P_F ~ poly(Pfat, 2))




```
summary(fit1)
```

```
##
## Call:
## lm(formula = P_F ~ poly(Pfat, 2), data = data5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29018 -0.16754  0.03355  0.16052  2.37258
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.62429    0.03612   72.65   <2e-16 ***
## poly(Pfat, 2)1 -14.16022    0.38399  -36.88   <2e-16 ***
## poly(Pfat, 2)2   6.16127    0.38399   16.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.384 on 110 degrees of freedom
## Multiple R-squared:  0.9363, Adjusted R-squared:  0.9352
## F-statistic: 808.7 on 2 and 110 DF,  p-value: < 2.2e-16
```

CHECK FOR OUTLIERS

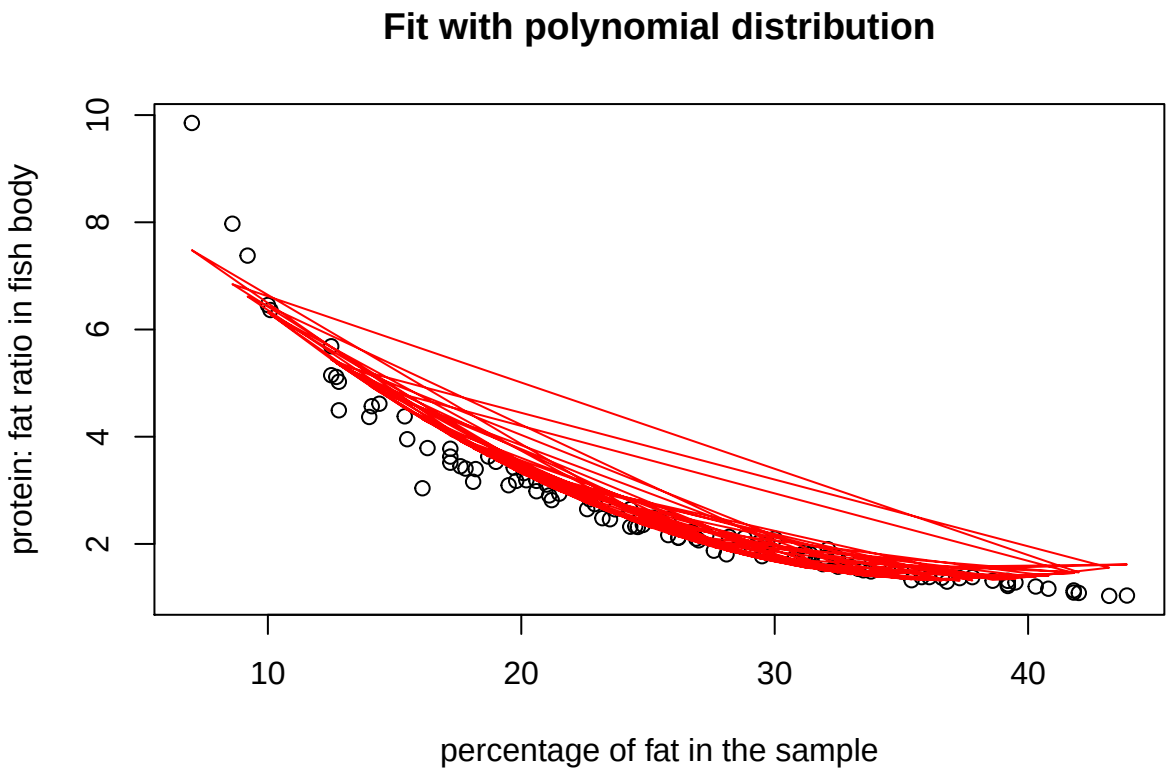
```r
influence.measures(fit1)
```

```
## Influence measures of
##   lm(formula = P_F ~ poly(Pfat, 2), data = data5) :
##
##        dfb.1_ dfb.p.P.2.1 dfb.p.P.2.2    dffit cov.r   cook.d    hat inf
## 1    0.066993   -4.53e-03   -0.059589  0.08977 1.030 2.70e-03 0.0159
## 2   -0.045036   -7.17e-02   -0.068209 -0.10875 1.077 3.97e-03 0.0516
## 3    0.028052    1.77e-02   -0.013298  0.03573 1.040 4.29e-04 0.0144
## 4    0.015360    8.96e-04   -0.013494  0.02046 1.043 1.41e-04 0.0157
## 5   -0.096510   -1.70e-01   -0.197109 -0.27778 1.080 2.57e-02 0.0733
## 6    0.012392    1.34e-02    0.003037  0.01847 1.048 1.15e-04 0.0197
## 7    0.028055    2.22e-02   -0.007199  0.03649 1.041 4.48e-04 0.0150
## 8    0.050839   -1.22e-02   -0.043672  0.06812 1.036 1.56e-03 0.0159
## 9   -0.075289    1.20e-01   -0.090773 -0.16807 1.057 9.45e-03 0.0441
## 10  -0.054607    7.50e-02   -0.035737 -0.09940 1.049 3.31e-03 0.0293
## 11   0.022195    1.88e-02   -0.003726  0.02934 1.042 2.90e-04 0.0155
## 12   0.045686    4.76e-03   -0.039636  0.06067 1.037 1.24e-03 0.0156
## 13   0.050661    1.05e-02   -0.042014  0.06665 1.036 1.49e-03 0.0153
## 14   0.003091   -1.91e-03   -0.001879  0.00409 1.044 5.62e-06 0.0155
## 15   0.008314   -6.37e-03   -0.003653  0.01109 1.044 4.14e-05 0.0158
## 16   0.010050   -4.02e-03   -0.007873  0.01338 1.044 6.02e-05 0.0157
## 17   0.021841   -4.07e-02    0.044161  0.06393 1.111 1.37e-03 0.0758   *
## 18  -0.138388    1.73e-01   -0.051714 -0.22716 0.994 1.70e-02 0.0238
## 19  -0.114360    1.12e-01    0.013995 -0.16102 1.005 8.61e-03 0.0175
## 20  -0.031874    1.05e-02    0.026182 -0.04258 1.041 6.09e-04 0.0158
## 21  -0.005465   -6.51e-03   -0.002659 -0.00891 1.052 2.67e-05 0.0235
## 22  -0.085435    6.74e-02    0.035024 -0.11432 1.021 4.36e-03 0.0158
## 24  -0.047269    3.57e-02    0.021447 -0.06298 1.037 1.33e-03 0.0157
## 25   0.101020    6.60e-02   -0.045034  0.12882 1.011 5.52e-03 0.0144
## 26  -0.008533    5.77e-04    0.007590 -0.01143 1.044 4.40e-05 0.0159
## 27   0.071073    4.89e-02   -0.028545  0.09087 1.027 2.76e-03 0.0145
## 28  -0.161933   -3.25e-01   -0.464625 -0.58949 1.084 1.14e-01 0.1173   *
## 29   0.065956    3.78e-02   -0.035661  0.08398 1.029 2.36e-03 0.0143
## 30  -0.121375    1.40e-01   -0.022720 -0.18685 1.004 1.16e-02 0.0210
## 31  -0.090191    1.27e-01   -0.066371 -0.16926 1.035 9.56e-03 0.0312
## 32   0.039535    2.22e-02   -0.021875  0.05035 1.038 8.52e-04 0.0144
## 33  -0.018320   -3.59e-03    0.015288 -0.02413 1.043 1.96e-04 0.0154
## 34   0.037774    1.91e-02   -0.023153  0.04824 1.038 7.82e-04 0.0144
## 35  -0.004118   -4.25e-03   -0.000639 -0.00595 1.047 1.19e-05 0.0185
## 36  -0.041537    2.52e-02    0.025719 -0.05495 1.038 1.01e-03 0.0155
## 37  -0.019837    4.29e-03    0.017181 -0.02659 1.043 2.38e-04 0.0159
## 38   0.048481   -6.61e-03   -0.042837  0.06503 1.037 1.42e-03 0.0159
## 39  -0.006783    5.74e-03    0.002254 -0.00917 1.045 2.83e-05 0.0162
## 40   0.039627    1.23e-02   -0.030600  0.05155 1.038 8.93e-04 0.0150
## 41   0.059828    3.23e-02   -0.034572  0.07626 1.031 1.95e-03 0.0144
## 42   0.049821    2.74e-02   -0.028184  0.06348 1.035 1.35e-03 0.0144
## 43  -0.017914   -2.69e-02   -0.022472 -0.03936 1.073 5.21e-04 0.0427
## 44  -0.004234    9.22e-05    0.003763 -0.00566 1.044 1.08e-05 0.0158
## 45   0.036931   -6.93e-02    0.076036  0.10931 1.110 4.01e-03 0.0775   *
## 46  -0.080992    1.26e-01   -0.090147 -0.17484 1.051 1.02e-02 0.0412
## 47  -0.144006   -2.77e-01   -0.371856 -0.48563 1.079 7.78e-02 0.1006   *
## 48  -0.032463   -4.76e-02   -0.037681 -0.06885 1.067 1.59e-03 0.0398
```

```
## 49  -0.024560   -3.60e-02   -0.028508 -0.05209 1.068 9.12e-04 0.0398
## 50   0.010818    7.55e-04   -0.009478  0.01440 1.044 6.98e-05 0.0157
## 51  -0.037741    4.75e-02   -0.015023 -0.06250 1.049 1.31e-03 0.0243
## 52  -0.221376    3.44e-01   -0.246399 -0.47790 0.929 7.33e-02 0.0412
## 53   0.036034    9.94e-03   -0.028587  0.04706 1.039 7.44e-04 0.0151
## 54   0.053978   -3.65e-03   -0.048012  0.07233 1.035 1.75e-03 0.0159
## 55  -0.019192    3.93e-03    0.016684 -0.02573 1.043 2.23e-04 0.0159
## 56  -0.061773   -1.02e-01   -0.104067 -0.15822 1.079 8.39e-03 0.0581
## 57   0.061610    3.96e-02   -0.028343  0.07852 1.031 2.07e-03 0.0144
## 58   0.039774    1.23e-02   -0.030714  0.05175 1.038 8.99e-04 0.0150
## 59  -0.006159    1.34e-04    0.005473 -0.00824 1.044 2.28e-05 0.0158
## 60   0.881265   -1.96e+00    2.885820  3.59622 0.251 2.58e+00 0.1474   *
## 61  -0.000755    6.65e-04    0.000214 -0.00103 1.045 3.56e-07 0.0164
## 62  -0.149823    2.13e-01   -0.114390 -0.28410 0.993 2.66e-02 0.0318
## 63  -0.085887   -1.52e-01   -0.175412 -0.24721 1.086 2.04e-02 0.0733   *
## 64   0.013832    1.54e-02    0.004358  0.02114 1.049 1.50e-04 0.0207
## 65   0.065761    1.36e-02   -0.054536  0.08651 1.030 2.51e-03 0.0153
## 66   0.026241    1.84e-02   -0.010141  0.03359 1.041 3.79e-04 0.0145
## 67  -0.127332    1.34e-01    0.000615 -0.18484 0.997 1.13e-02 0.0186
## 68  -0.066475    1.04e-01   -0.076025 -0.14511 1.059 7.05e-03 0.0422
## 69  -0.001417    9.40e-04    0.000794 -0.00188 1.044 1.19e-06 0.0155
## 70   0.081503    3.09e-02   -0.058982  0.10524 1.022 3.70e-03 0.0148
## 71   0.036811    1.73e-03   -0.032419  0.04908 1.040 8.09e-04 0.0157
## 72   0.022369   -9.45e-03   -0.017199  0.02976 1.043 2.98e-04 0.0157
## 73  -0.049604    3.29e-02    0.027788 -0.06568 1.036 1.45e-03 0.0155
## 74   0.053868    4.26e-02   -0.013823  0.07007 1.034 1.65e-03 0.0150
## 75  -0.030744    7.36e-03    0.026410 -0.04119 1.041 5.70e-04 0.0159
## 76   0.106602    4.41e-02   -0.074223  0.13717 1.008 6.26e-03 0.0147
## 77   0.023897    1.95e-02   -0.005301  0.03127 1.042 3.29e-04 0.0151
## 78   0.012328    1.44e-02    0.005380  0.01971 1.051 1.31e-04 0.0226
## 79   0.034004    3.35e-02    0.002338  0.04781 1.042 7.68e-04 0.0175
## 80   0.075290    2.59e-02   -0.056391  0.09758 1.025 3.18e-03 0.0149
## 81  -0.098490    1.04e-01    0.000476 -0.14298 1.017 6.81e-03 0.0186
## 82   0.023159   -7.13e-03   -0.019273  0.03096 1.043 3.22e-04 0.0158
## 84   0.062726    4.32e-02   -0.025193  0.08020 1.030 2.16e-03 0.0145
## 85   0.209067   -4.12e-01    0.493943  0.67604 1.004 1.48e-01 0.0925   *
## 86  -0.090632    8.50e-02    0.017772 -0.12552 1.020 5.26e-03 0.0170
## 87   0.049011    1.75e-02   -0.036306  0.06344 1.036 1.35e-03 0.0148
## 88  -0.061706    6.49e-02    0.000298 -0.08958 1.035 2.69e-03 0.0186
## 89  -0.027845    1.21e-02    0.021198 -0.03702 1.042 4.61e-04 0.0156
## 90  -0.339908    4.00e-01   -0.079001 -0.53120 0.753 8.50e-02 0.0216   *
## 92  -0.021168    1.50e-02    0.010771 -0.02809 1.043 2.65e-04 0.0156
## 93   0.015339    2.00e-02    0.011522  0.02774 1.058 2.59e-04 0.0289
## 94  -0.012145    6.80e-03    0.008043 -0.01608 1.043 8.69e-05 0.0155
## 95   0.321984   -6.56e-01    0.837215  1.11121 0.872 3.79e-01 0.1054   *
## 96  -0.102098   -1.82e-01   -0.216111 -0.30071 1.081 3.01e-02 0.0768
## 97   0.012584    4.48e-03   -0.009322  0.01629 1.043 8.92e-05 0.0148
## 98   0.032692    2.47e-02   -0.010038  0.04221 1.040 5.99e-04 0.0148
## 99   0.040680    1.44e-03   -0.035905  0.05428 1.039 9.89e-04 0.0158
## 100 -0.040805    1.49e-02    0.032785 -0.05442 1.039 9.95e-04 0.0157
## 101  0.062832   -9.99e-02    0.075753  0.14026 1.063 6.59e-03 0.0441
## 102  0.039027    3.45e-02   -0.004366  0.05224 1.039 9.17e-04 0.0159
## 103 -0.017139   -2.37e-03    0.014687 -0.02270 1.043 1.73e-04 0.0155
## 104  0.054194    2.98e-02   -0.030657  0.06905 1.033 1.60e-03 0.0144
```

```
## 105  0.069262    3.81e-02   -0.039181  0.08825 1.028 2.61e-03 0.0144
## 106  0.011220    1.40e-02    0.006911  0.01924 1.055 1.24e-04 0.0260
## 107  0.063568    4.45e-02   -0.024566  0.08138 1.030 2.22e-03 0.0145
## 108  0.078734    2.17e-02   -0.062463  0.10282 1.024 3.53e-03 0.0151
## 109 -0.116875    1.18e-01    0.009832 -0.16612 1.004 9.16e-03 0.0179
## 110  0.005891   -4.24e-03   -0.002918  0.00782 1.044 2.06e-05 0.0156
## 111 -0.004167   -5.83e-03   -0.004082 -0.00824 1.065 2.29e-05 0.0346
## 112 -0.008085   -1.19e-02   -0.009385 -0.01715 1.070 9.89e-05 0.0398
## 113  0.036585    1.39e-02   -0.026476  0.04724 1.039 7.50e-04 0.0148
## 114 -0.001512    2.75e-04    0.001323 -0.00203 1.044 1.38e-06 0.0159
## 115 -0.057887    3.44e-02    0.036485 -0.07659 1.033 1.97e-03 0.0155
## 116 -0.157157    1.49e-01    0.027975 -0.21849 0.971 1.57e-02 0.0171
```

```
plot(data5$Pfat, data5$P_F, xlab = "percentage of fat in the sample", ylab = "protein: fat ratio in fis
lines(data5$Pfat, predict(fit1), col="red")
```



since tank is a significant factor, we need to regress it out

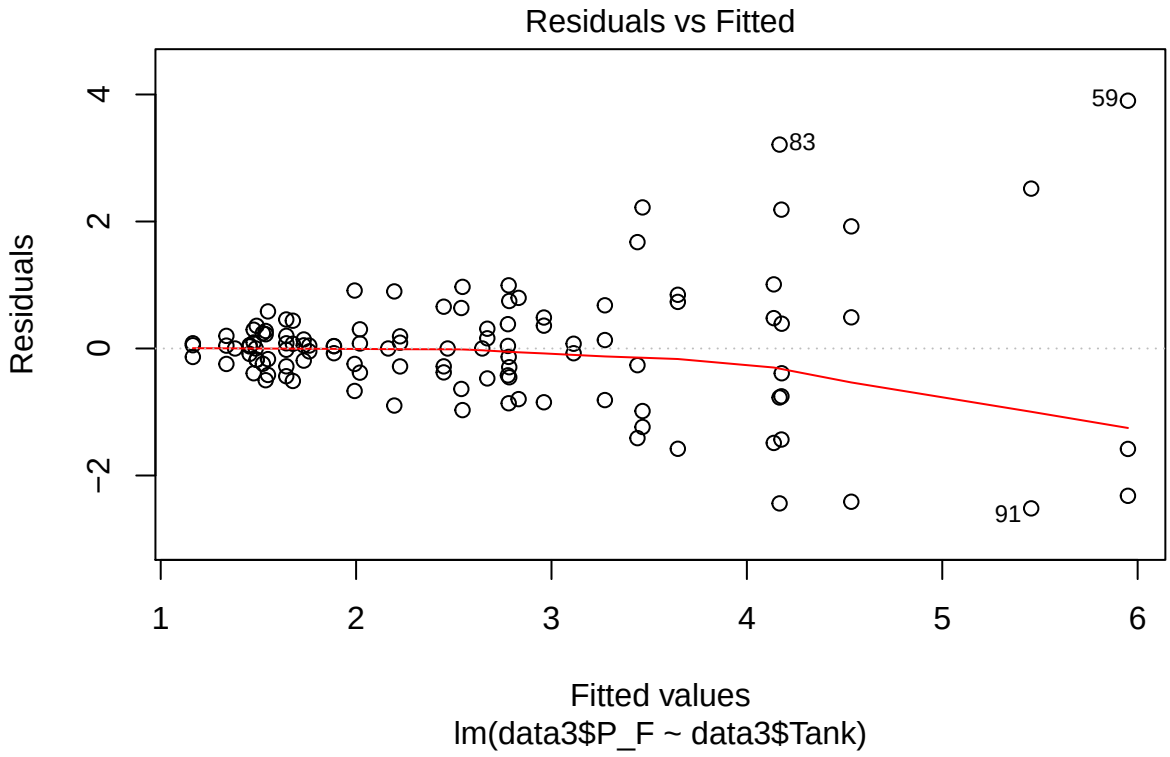
```
model_tank <- lm(data3$P_F ~ data3$Tank)
anova(model_tank)
```

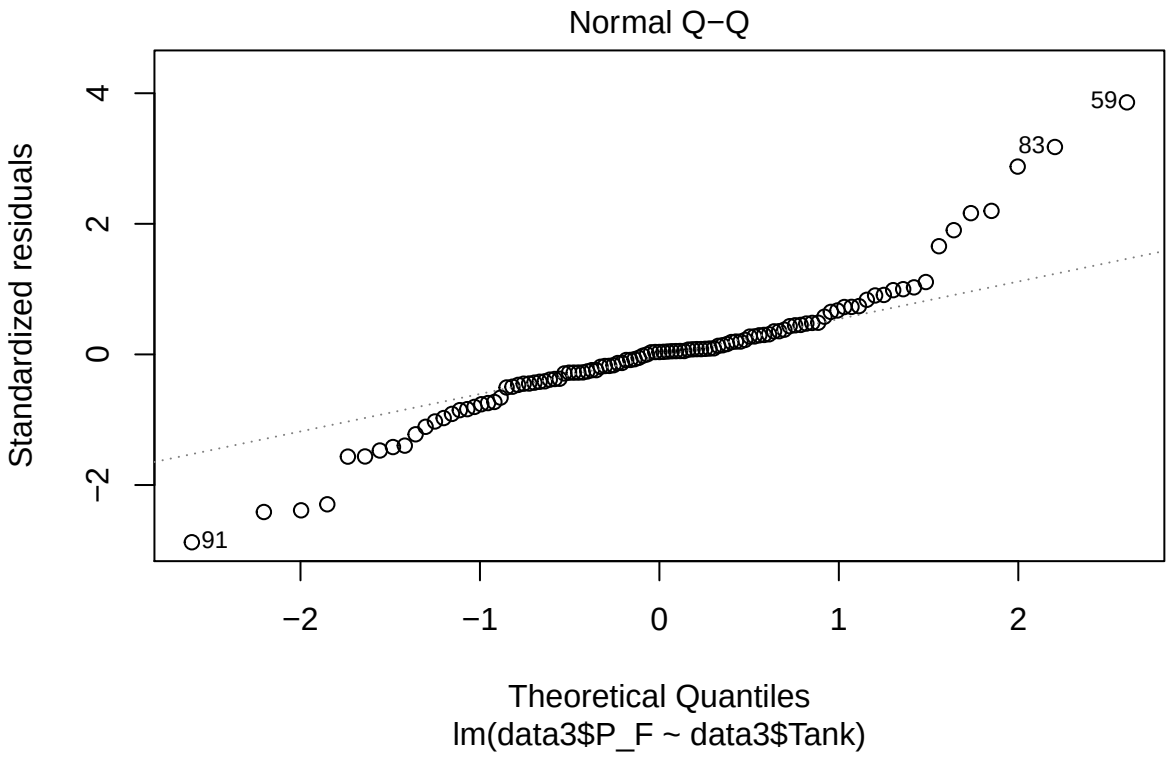
```
## Analysis of Variance Table
##
## Response: data3$P_F
##            Df Sum Sq Mean Sq F value   Pr(>F)
```

```
## data3$Tank 43 148.94  3.4638  2.2601 0.001225 **
## Residuals  69 105.75  1.5326
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

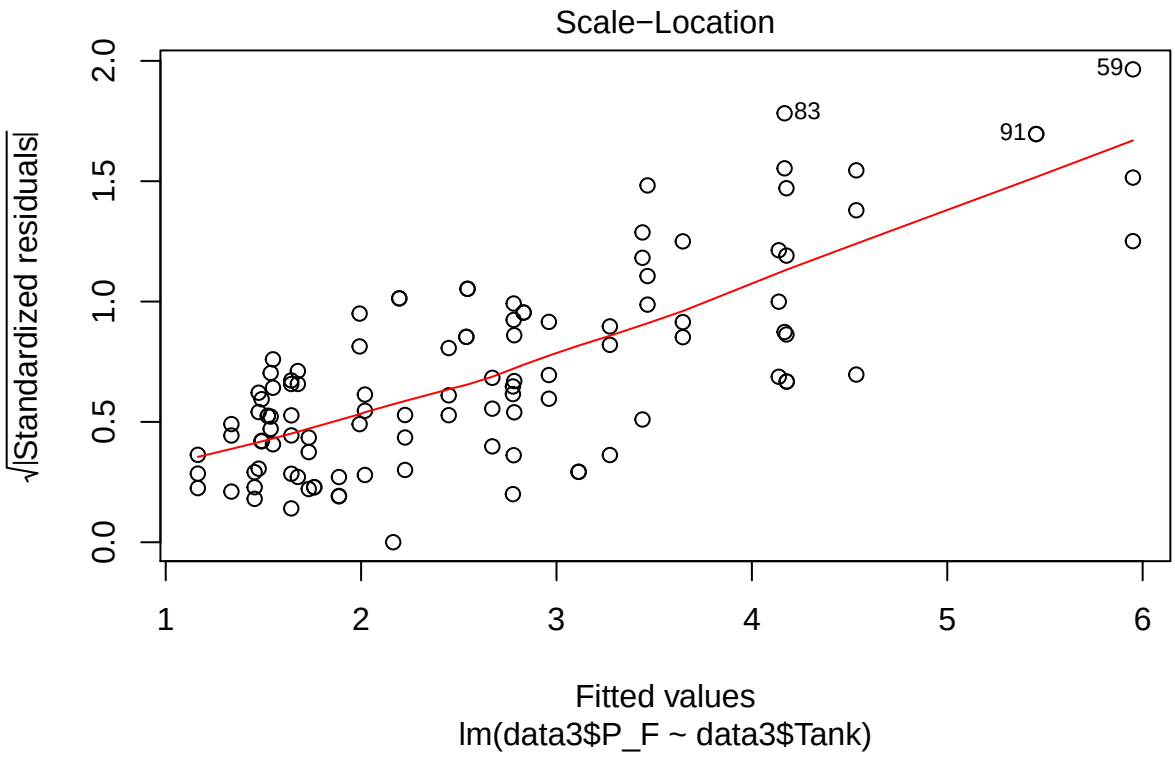
```r
tank_resid = residuals(model_tank)
data3$tank_resid<- tank_resid
plot(model_tank)
```

```
## Warning: not plotting observations with leverage one:
##   1, 11, 81, 90
```

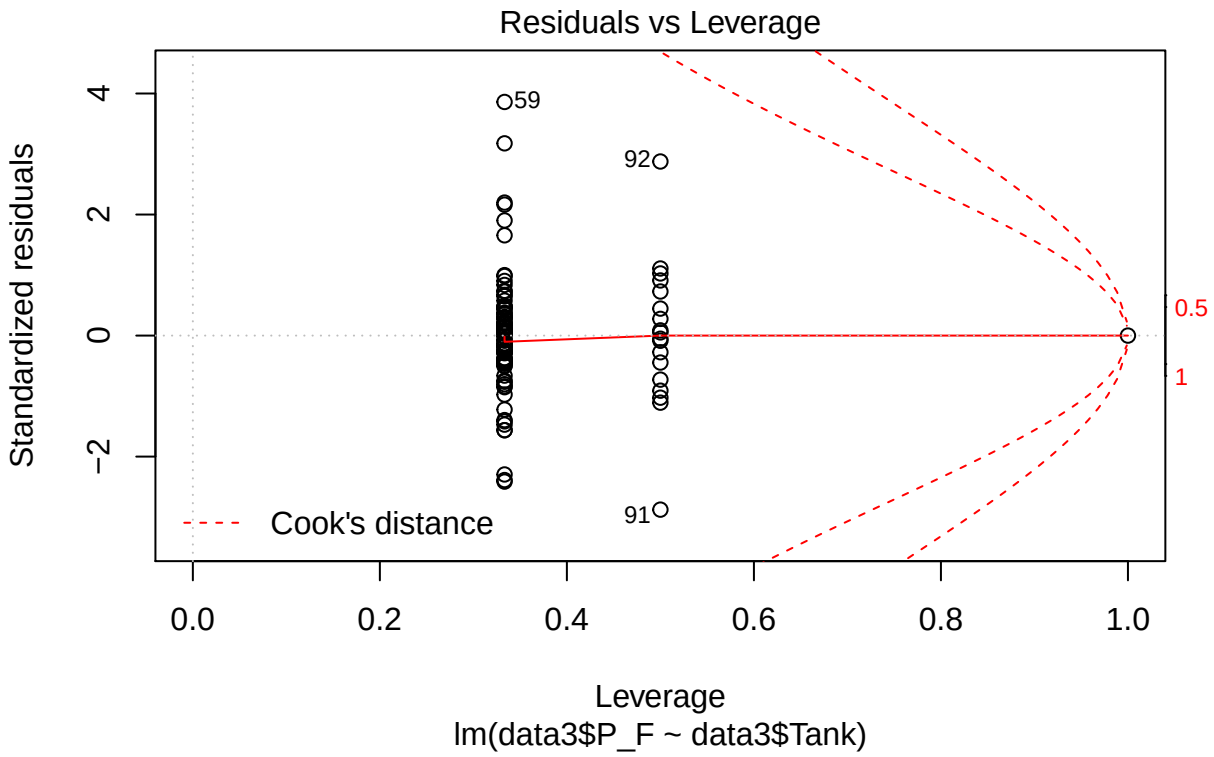




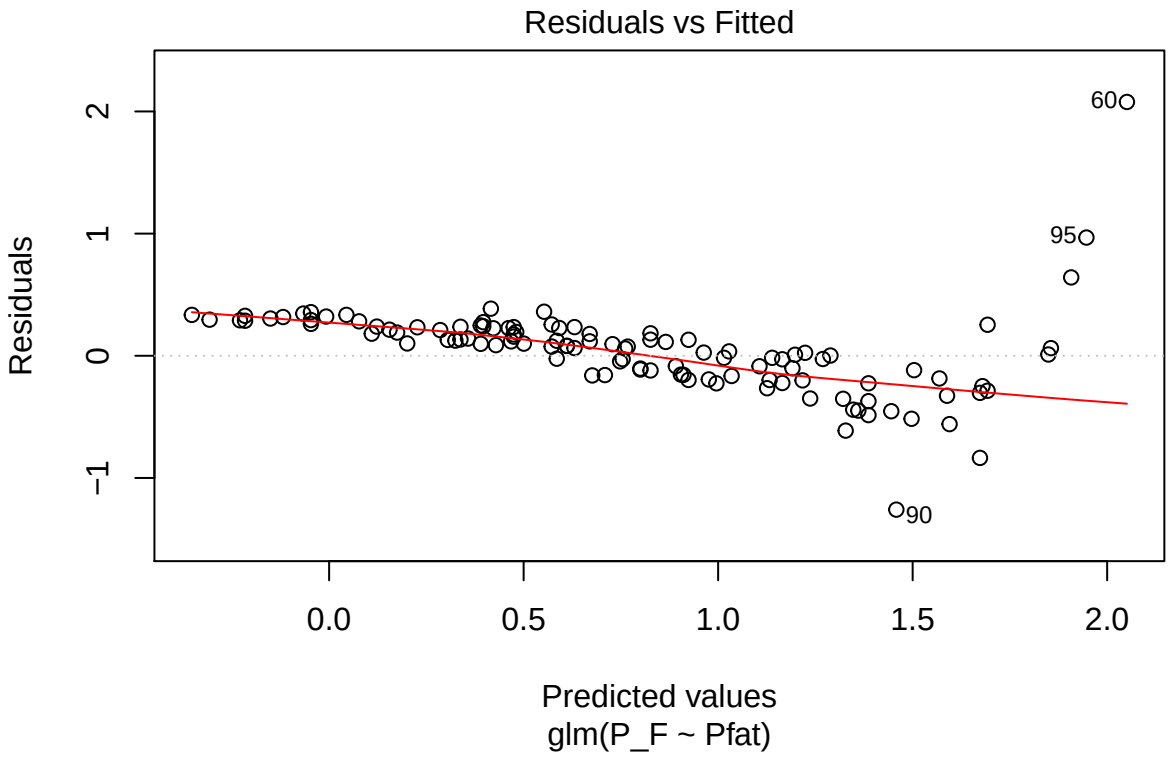

```
## Warning: not plotting observations with leverage one:
##   1, 11, 81, 90
```

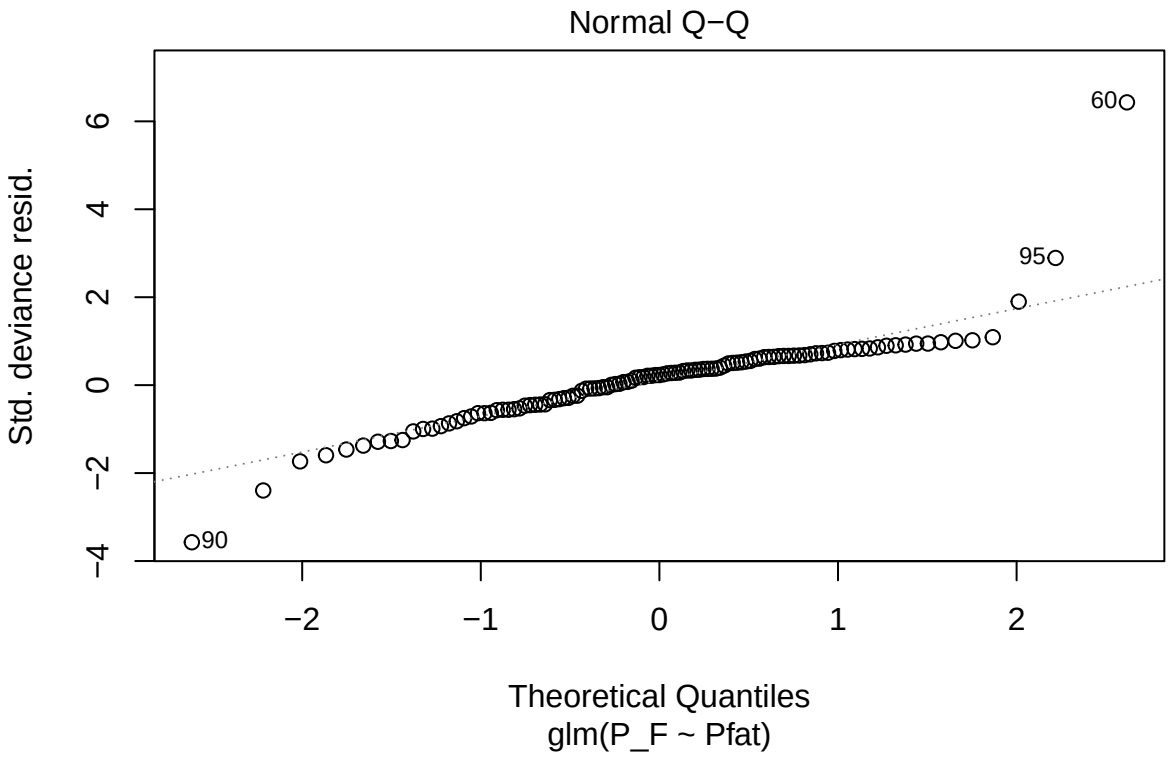


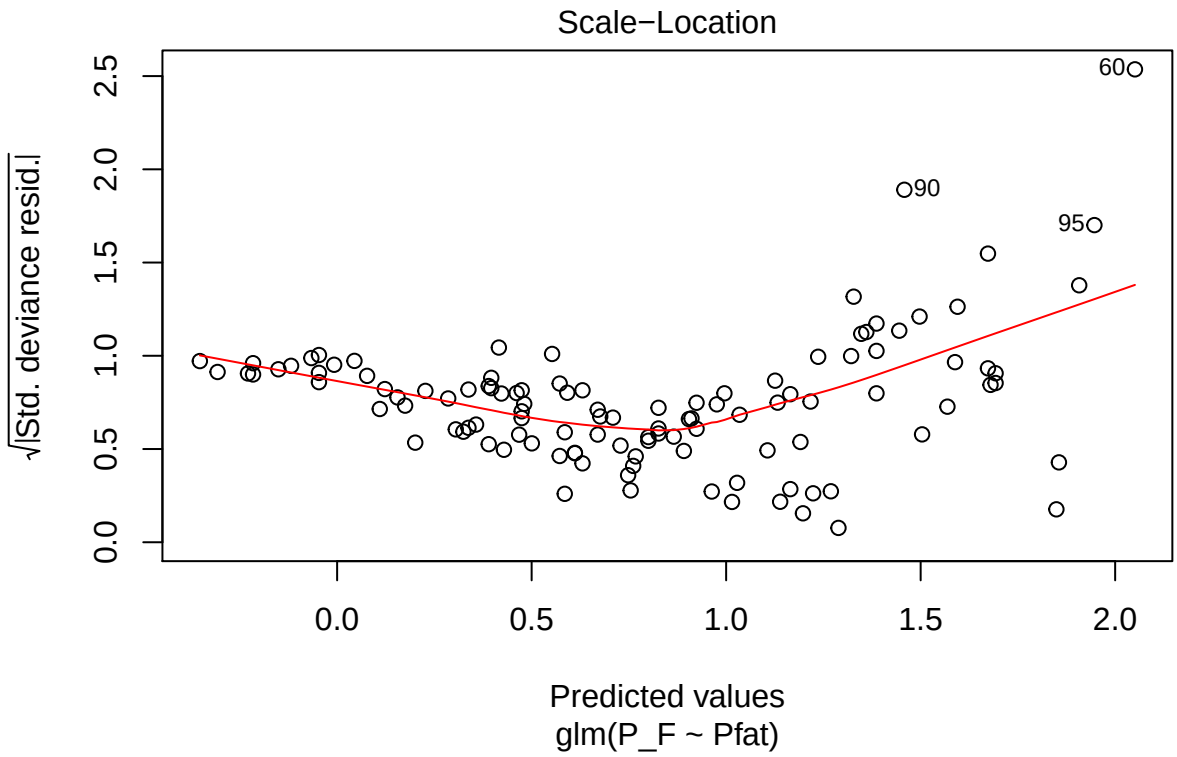

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```

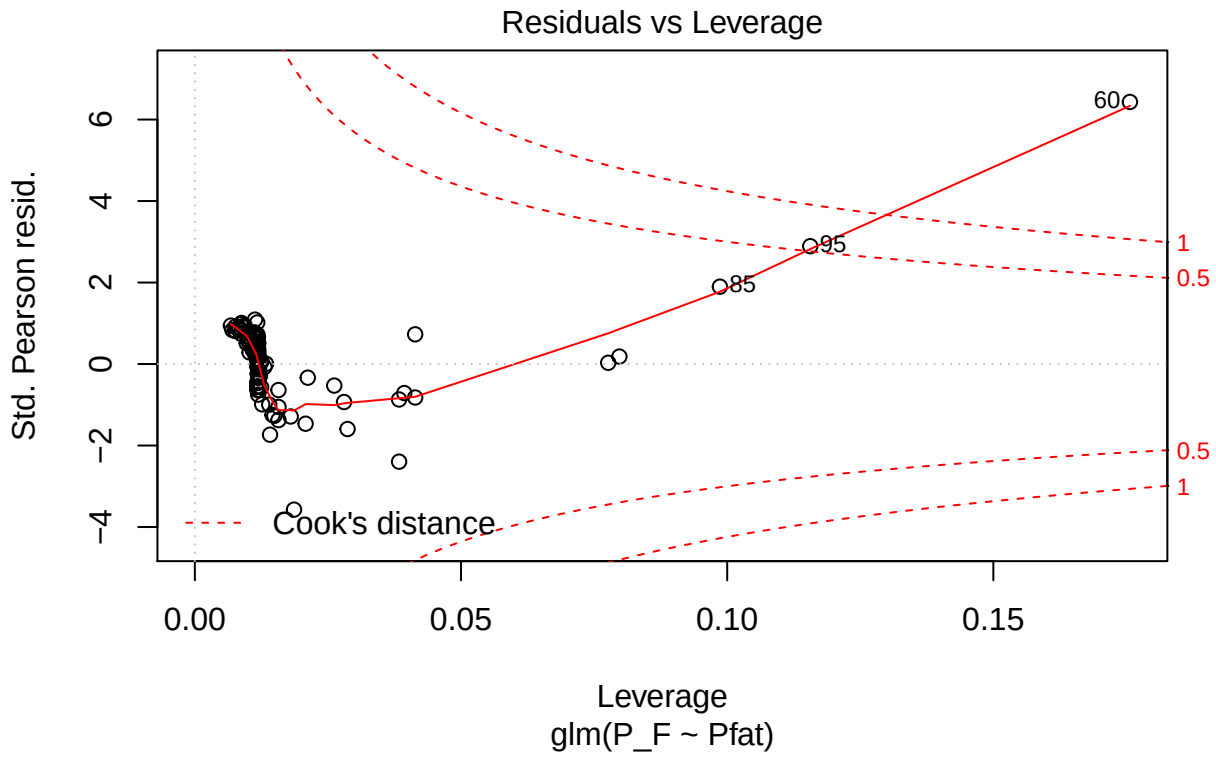



```
# fit the exponential model to the data
fit_exp <- glm(P_F ~ Pfat, data = data5, family = gaussian(link=log))
plot(fit_exp)
```


Residuals vs Fitted
Residuals
Predicted values
glm(P_F ~ Pfat)
60
95
90
2
1
0
−1
0.0
0.5
1.0
1.5
2.0


Normal Q−Q
Std. deviance resid.
Theoretical Quantiles
glm(P_F ~ Pfat)
60
95
90
−4
−2
0
2
4
6
−2
−1
0
1
2


Scale−Location
√|Std. deviance resid.|
0.0
0.5
1.0
1.5
2.0
2.5
60
90
95
0.0
0.5
1.0
1.5
2.0
Predicted values
glm(P_F ~ Pfat)




```
# check residuals
# res_exp <- residuals(fit_exp)/sqrt(summary(fit_exp)$dispersion)

# ggplot(res_exp, aes(x=predict(fit_exp), y=res_exp)) + geom_point()
```

after the first feedback

```
# esclude the unwanted columns
data_ready <- data3[,-c(1:3,5:8,11,22:24,26:31)]

# compare tank and family, tank is significant
model_blocking <- lm(data3$P_F ~ data3$Tank + data3$Family)
anova(model_blocking)
```

```
## Analysis of Variance Table
##
## Response: data3$P_F
##              Df  Sum Sq Mean Sq F value   Pr(>F)
## data3$Tank   43 148.944  3.4638  2.1275 0.003352 **
## data3$Family  8   6.432  0.8040  0.4938 0.855961
## Residuals    61  99.316  1.6281
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_blocking <- lm(data3$P_F ~  data3$Family + data3$Tank)
anova(model_blocking)
```

```
## Analysis of Variance Table
##
## Response: data3$P_F
##              Df  Sum Sq Mean Sq F value   Pr(>F)
## data3$Family  8  10.440  1.3050  0.8015 0.603570
## data3$Tank   43 144.936  3.3706  2.0702 0.004475 **
## Residuals    61  99.316  1.6281
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# how anout sex
model_blocking <- lm(data3$P_F ~  data3$Sex + data3$Tank)
anova(model_blocking)
```

```
## Analysis of Variance Table
##
## Response: data3$P_F
##            Df  Sum Sq Mean Sq F value   Pr(>F)
## data3$Sex   1  11.872 11.8719  7.7463 0.006938 **
## data3$Tank 42 137.072  3.2636  2.1295 0.002606 **
## Residuals  69 105.748  1.5326
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# both sex and tank is significant, family is not significant and is excluded in the prior code

anova(lm(data3$P_F ~ data3$Pfat + data3$Tank))
```

```
## Analysis of Variance Table
##
## Response: data3$P_F
##            Df  Sum Sq Mean Sq  F value Pr(>F)
## data3$Pfat  1 200.512 200.512 399.7325 <2e-16 ***
## data3$Tank 43  20.071   0.467   0.9305 0.5943
## Residuals  68  34.110   0.502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lm(data3$P_F ~ data3$Pfat + data3$Sex))
```

```
## Analysis of Variance Table
##
## Response: data3$P_F
##             Df  Sum Sq Mean Sq  F value Pr(>F)
## data3$Pfat   1 200.512 200.512 411.0543 <2e-16 ***
## data3$Sex    1   0.523   0.523   1.0716 0.3029
## Residuals  110  53.658   0.488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# try to conduct stepwise regression
data_transformed <- na.omit(data_ready)
model_null <- lm(P_F ~ 1, data=data_transformed)
model_full <- lm(P_F ~ ., data = data_transformed)
step(model_null, scope = list(upper = formula(model_full), lower = formula(model_null)), direction = 'f
```

```
## Start:  AIC=93.83
## P_F ~ 1
##
##               Df Sum of Sq     RSS     AIC
## + Pfat         1   200.512  54.181 -79.062
## + Pprotein     1   146.626 108.066  -1.045
## + Pwater       1   124.434 130.259  20.061
## + Fat          1   114.943 139.749  28.008
## + Pash         1   103.215 151.478  37.115
## + Dry_wt       1    77.318 177.375  54.949
## + P_L          1    67.329 187.363  61.140
## + Wet_wt       1    58.823 195.869  66.157
## + Protein      1    52.936 201.757  69.503
## + Fat_free_wt  1    44.769 209.924  73.987
## + Tank        43   148.944 105.748  80.505
## + Ashed_wt     1    13.461 241.232  89.696
## + Sex          1    11.872 242.820  90.438
## <none>                     254.692  93.832
##
## Step:  AIC=-79.06
## P_F ~ Pfat
##
##               Df Sum of Sq    RSS     AIC
## + Fat          1    5.9534 48.227 -90.216
## + Dry_wt       1    1.7658 52.415 -80.807
## + Pwater       1    1.0597 53.121 -79.294
## <none>                     54.181 -79.062
## + Wet_wt       1    0.8085 53.372 -78.761
## + Protein      1    0.7010 53.480 -78.534
## + Fat_free_wt  1    0.5675 53.613 -78.252
## + Sex          1    0.5227 53.658 -78.158
## + Ashed_wt     1    0.1600 54.021 -77.397
## + P_L          1    0.0603 54.120 -77.188
## + Pprotein     1    0.0298 54.151 -77.124
## + Pash         1    0.0285 54.152 -77.122
## + Tank        43   20.0708 34.110 -45.352
##
## Step:  AIC=-90.22
## P_F ~ Pfat + Fat
##
##               Df Sum of Sq    RSS      AIC
## + Fat_free_wt  1    7.4289 40.798 -107.119
## + Dry_wt       1    7.4289 40.798 -107.119
## + Ashed_wt     1    6.7417 41.486 -105.231
## + Wet_wt       1    5.7240 42.503 -102.492
## + Protein      1    5.5263 42.701 -101.968
## <none>                     48.227  -90.216
```

```
## + Pprotein     1    0.8207 47.406  -90.155
## + Pash         1    0.8101 47.417  -90.130
## + P_L          1    0.5205 47.707  -89.442
## + Sex          1    0.2188 48.008  -88.729
## + Pwater       1    0.0103 48.217  -88.240
## + Tank        43   18.3447 29.882  -58.303
##
## Step:  AIC=-107.12
## P_F ~ Pfat + Fat + Fat_free_wt
##
##            Df Sum of Sq    RSS      AIC
## + Pprotein  1    1.1291 39.669 -108.290
## + Pash      1    1.1188 39.679 -108.261
## + Ashed_wt  1    0.9867 39.812 -107.885
## + Protein   1    0.9862 39.812 -107.884
## + Sex       1    0.8536 39.945 -107.508
## <none>                  40.798 -107.119
## + Wet_wt    1    0.2887 40.510 -105.921
## + Pwater    1    0.0205 40.778 -105.176
## + P_L       1    0.0153 40.783 -105.161
## + Tank     43   14.8552 25.943  -72.278
##
## Step:  AIC=-108.29
## P_F ~ Pfat + Fat + Fat_free_wt + Pprotein
##
##            Df Sum of Sq    RSS      AIC
## + Sex       1    0.9702 38.699 -109.088
## <none>                  39.669 -108.290
## + P_L       1    0.5383 39.131 -107.834
## + Pash      1    0.2695 39.400 -107.061
## + Pwater    1    0.1351 39.534 -106.676
## + Wet_wt    1    0.0404 39.629 -106.405
## + Ashed_wt  1    0.0003 39.669 -106.291
## + Protein   1    0.0003 39.669 -106.291
## + Tank     43   14.0206 25.648  -71.568
##
## Step:  AIC=-109.09
## P_F ~ Pfat + Fat + Fat_free_wt + Pprotein + Sex
##
##            Df Sum of Sq    RSS      AIC
## <none>                  38.699 -109.088
## + Pwater    1    0.4049 38.294 -108.277
## + P_L       1    0.3843 38.315 -108.216
## + Pash      1    0.2123 38.487 -107.710
## + Wet_wt    1    0.0125 38.686 -107.125
## + Ashed_wt  1    0.0008 38.698 -107.091
## + Protein   1    0.0008 38.698 -107.090
## + Tank     42   13.0504 25.648  -71.568

##
## Call:
## lm(formula = P_F ~ Pfat + Fat + Fat_free_wt + Pprotein + Sex,
##     data = data_transformed)
##
```

```
## Coefficients:
## (Intercept)         Pfat          Fat  Fat_free_wt     Pprotein
##     6.48586     -0.22609     22.37837     -7.92262      0.03346
##        Sexm
##    -0.19282
```

```
step(model_full, scope = list(upper = formula(model_full), lower = formula(model_null)), direction = 'ba
```

```
## Start:  AIC=-70.53
## P_F ~ Sex + P_L + Tank + Wet_wt + Dry_wt + Fat_free_wt + Ashed_wt +
##     Fat + Protein + Pwater + Pfat + Pash + Pprotein
##
##
## Step:  AIC=-70.53
## P_F ~ Sex + P_L + Tank + Wet_wt + Dry_wt + Fat_free_wt + Ashed_wt +
##     Protein + Pwater + Pfat + Pash + Pprotein
##
##
## Step:  AIC=-70.53
## P_F ~ Sex + Tank + Wet_wt + Dry_wt + Fat_free_wt + Ashed_wt +
##     Protein + Pwater + Pfat + Pash + Pprotein
##
##
## Step:  AIC=-70.53
## P_F ~ Tank + Wet_wt + Dry_wt + Fat_free_wt + Ashed_wt + Protein +
##     Pwater + Pfat + Pash + Pprotein
##
##               Df Sum of Sq    RSS      AIC
## - Tank        43   13.3774 37.071 -105.945
## - Pfat         1    0.1768 23.870  -71.687
## - Pash         1    0.2268 23.920  -71.451
## - Pprotein     1    0.2553 23.949  -71.316
## <none>                     23.693  -70.527
## - Fat_free_wt  1    0.7482 24.442  -69.014
## - Ashed_wt     1    0.7590 24.452  -68.965
## - Protein      1    0.7828 24.476  -68.854
## - Pwater       1    0.8565 24.550  -68.515
## - Wet_wt       1    0.9003 24.594  -68.313
## - Dry_wt       1    1.6687 25.362  -64.837
##
## Step:  AIC=-105.94
## P_F ~ Wet_wt + Dry_wt + Fat_free_wt + Ashed_wt + Protein + Pwater +
##     Pfat + Pash + Pprotein
##
##               Df Sum of Sq    RSS     AIC
## - Fat_free_wt  1   0.05043 37.121 -107.79
## - Ashed_wt     1   0.05867 37.130 -107.77
## - Protein      1   0.06334 37.134 -107.75
## - Pfat         1   0.12266 37.194 -107.57
## - Pash         1   0.18340 37.254 -107.39
## - Pprotein     1   0.21132 37.282 -107.30
## <none>                     37.071 -105.94
## - Wet_wt       1   2.14341 39.214 -101.59
## - Pwater       1   2.29725 39.368 -101.15
```

```
## - Dry_wt       1   2.34199 39.413 -101.02
##
## Step:  AIC=-107.79
## P_F ~ Wet_wt + Dry_wt + Ashed_wt + Protein + Pwater + Pfat +
##     Pash + Pprotein
##
##            Df Sum of Sq    RSS     AIC
## - Pfat      1    0.1204 37.242 -109.43
## - Pash      1    0.1801 37.301 -109.24
## - Pprotein  1    0.2078 37.329 -109.16
## <none>                  37.121 -107.79
## - Ashed_wt  1    2.0350 39.156 -103.76
## - Wet_wt    1    2.1008 39.222 -103.57
## - Pwater    1    2.2475 39.369 -103.15
## - Dry_wt    1    2.4644 39.586 -102.53
## - Protein   1    5.9089 43.030  -93.10
##
## Step:  AIC=-109.43
## P_F ~ Wet_wt + Dry_wt + Ashed_wt + Protein + Pwater + Pash +
##     Pprotein
##
##            Df Sum of Sq    RSS      AIC
## <none>                  37.242 -109.426
## - Ashed_wt  1    1.9547 39.196 -105.645
## - Wet_wt    1    2.2159 39.457 -104.895
## - Pwater    1    2.3206 39.562 -104.595
## - Dry_wt    1    2.4313 39.673 -104.279
## - Protein   1    6.2680 43.510  -93.848
## - Pash      1   16.5341 53.776  -69.910
## - Pprotein  1   23.2643 60.506  -56.585

##
## Call:
## lm(formula = P_F ~ Wet_wt + Dry_wt + Ashed_wt + Protein + Pwater +
##     Pash + Pprotein, data = data_transformed)
##
## Coefficients:
## (Intercept)       Wet_wt       Dry_wt     Ashed_wt      Protein
##     -3.5310       8.2526      12.9655     -35.9563     -57.7010
##      Pwater         Pash     Pprotein
##     -0.2718       0.2532       0.3545
```

```
step(model_full, scope = list(upper = formula(model_full), lower = formula(model_null)), direction = 'b
```

```
## Start:  AIC=-70.53
## P_F ~ Sex + P_L + Tank + Wet_wt + Dry_wt + Fat_free_wt + Ashed_wt +
##     Fat + Protein + Pwater + Pfat + Pash + Pprotein
##
##
## Step:  AIC=-70.53
## P_F ~ Sex + P_L + Tank + Wet_wt + Dry_wt + Fat_free_wt + Ashed_wt +
##     Protein + Pwater + Pfat + Pash + Pprotein
##
```

```
##
## Step:  AIC=-70.53
## P_F ~ Sex + Tank + Wet_wt + Dry_wt + Fat_free_wt + Ashed_wt +
##     Protein + Pwater + Pfat + Pash + Pprotein
##
##
## Step:  AIC=-70.53
## P_F ~ Tank + Wet_wt + Dry_wt + Fat_free_wt + Ashed_wt + Protein +
##     Pwater + Pfat + Pash + Pprotein
##
##               Df Sum of Sq    RSS      AIC
## - Tank        43   13.3774 37.071 -105.945
## - Pfat         1    0.1768 23.870  -71.687
## - Pash         1    0.2268 23.920  -71.451
## - Pprotein     1    0.2553 23.949  -71.316
## <none>                     23.693  -70.527
## - Fat_free_wt  1    0.7482 24.442  -69.014
## - Ashed_wt     1    0.7590 24.452  -68.965
## - Protein      1    0.7828 24.476  -68.854
## - Pwater       1    0.8565 24.550  -68.515
## - Wet_wt       1    0.9003 24.594  -68.313
## - Dry_wt       1    1.6687 25.362  -64.837
##
## Step:  AIC=-105.94
## P_F ~ Wet_wt + Dry_wt + Fat_free_wt + Ashed_wt + Protein + Pwater +
##     Pfat + Pash + Pprotein
##
##               Df Sum of Sq    RSS      AIC
## - Fat_free_wt  1    0.0504 37.121 -107.791
## - Ashed_wt     1    0.0587 37.130 -107.766
## - Protein      1    0.0633 37.134 -107.752
## - Pfat         1    0.1227 37.194 -107.572
## - Pash         1    0.1834 37.254 -107.387
## - Pprotein     1    0.2113 37.282 -107.303
## <none>                     37.071 -105.945
## + Sex          1    0.6213 36.450 -105.855
## + P_L          1    0.3973 36.674 -105.162
## - Wet_wt       1    2.1434 39.214 -101.593
## - Pwater       1    2.2972 39.368 -101.151
## - Dry_wt       1    2.3420 39.413 -101.022
## + Tank        43   13.3774 23.693  -70.527
##
## Step:  AIC=-107.79
## P_F ~ Wet_wt + Dry_wt + Ashed_wt + Protein + Pwater + Pfat +
##     Pash + Pprotein
##
##               Df Sum of Sq    RSS      AIC
## - Pfat         1    0.1204 37.242 -109.426
## - Pash         1    0.1801 37.301 -109.244
## - Pprotein     1    0.2078 37.329 -109.160
## <none>                     37.121 -107.791
## + Sex          1    0.5973 36.524 -107.624
## + P_L          1    0.3478 36.773 -106.855
## + Fat_free_wt  1    0.0504 37.071 -105.945
```

```
## + Fat          1    0.0504 37.071 -105.945
## - Ashed_wt     1    2.0350 39.156 -103.761
## - Wet_wt       1    2.1008 39.222 -103.571
## - Pwater       1    2.2475 39.369 -103.149
## - Dry_wt       1    2.4644 39.586 -102.528
## - Protein      1    5.9089 43.030  -93.100
## + Tank        43   12.6796 24.442  -69.014
##
## Step:  AIC=-109.43
## P_F ~ Wet_wt + Dry_wt + Ashed_wt + Protein + Pwater + Pash +
##     Pprotein
##
##               Df Sum of Sq    RSS      AIC
## <none>                     37.242 -109.426
## + Sex          1    0.6079 36.634 -109.285
## + P_L          1    0.3442 36.897 -108.475
## + Pfat         1    0.1204 37.121 -107.791
## + Fat_free_wt  1    0.0481 37.194 -107.572
## + Fat          1    0.0481 37.194 -107.572
## - Ashed_wt     1    1.9547 39.196 -105.645
## - Wet_wt       1    2.2159 39.457 -104.895
## - Pwater       1    2.3206 39.562 -104.595
## - Dry_wt       1    2.4313 39.673 -104.279
## - Protein      1    6.2680 43.510  -93.848
## + Tank        43   12.7329 24.509  -70.705
## - Pash         1   16.5341 53.776  -69.910
## - Pprotein     1   23.2643 60.506  -56.585

##
## Call:
## lm(formula = P_F ~ Wet_wt + Dry_wt + Ashed_wt + Protein + Pwater +
##     Pash + Pprotein, data = data_transformed)
##
## Coefficients:
## (Intercept)       Wet_wt       Dry_wt     Ashed_wt      Protein
##     -3.5310       8.2526      12.9655     -35.9563     -57.7010
##      Pwater         Pash     Pprotein
##     -0.2718       0.2532       0.3545
```

```
#### FORWARD
# Coefficients:
# (Intercept)         Pfat          Fat  Fat_free_wt     Pprotein         Sexm
#    6.48586     -0.22609     22.37837     -7.92262      0.03346     -0.19282


#### BACKWARD
# Coefficients:
# (Intercept)       Wet_wt  Fat_free_wt     Ashed_wt      Protein       Pwater         Pfat         Pas
# -5.537e-16    2.498e-01   -2.212e+00    3.723e-01    1.660e+00   -7.440e-02   -1.180e+00   -1.105e-01

#### BOTH
# Coefficients:
# (Intercept)       Wet_wt       Dry_wt     Ashed_wt      Protein       Pwater         Pash     Pprotei
#    -3.5310       8.2526      12.9655     -35.9563     -57.7010      -0.2718       0.2532       0.3545
```

```
#########
# why is tank not show up here????
# why P_L not showing up????
```

since there is no significant value of outliers may I can use the original dataset

```
anova(lm(P_F~Sex, data = data3))
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value  Pr(>F)
## Sex         1  11.872 11.8719   5.427 0.02164 *
## Residuals 111 242.820  2.1876
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm(P_F~Wet_wt, data = data3))
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Wet_wt      1  58.823  58.823  33.335 7.158e-08 ***
## Residuals 111 195.869   1.765
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm(P_F~Dry_wt, data = data3))
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Dry_wt      1  77.318  77.318  48.385 2.561e-10 ***
## Residuals 111 177.375   1.598
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm(P_F~Ashed_wt, data = data3))
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value Pr(>F)
## Ashed_wt    1  13.461 13.4606  6.1937 0.0143 *
## Residuals 111 241.232  2.1733
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lm(P_F~Fat, data = data3))
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df Sum Sq Mean Sq F value    Pr(>F)
## Fat         1 114.94 114.943  91.297 3.797e-16 ***
## Residuals 111 139.75   1.259
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lm(P_F~Protein, data = data3))
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value   Pr(>F)
## Protein     1  52.936  52.936  29.123 3.89e-07 ***
## Residuals 111 201.757   1.818
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lm(P_F~Pfat, data = data3))
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Pfat        1 200.512 200.512  410.79 < 2.2e-16 ***
## Residuals 111  54.181   0.488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lm(P_F~Pprotein, data = data3))
```

```
## Analysis of Variance Table
##
## Response: P_F
##            Df Sum Sq Mean Sq F value    Pr(>F)
## Pprotein    1 146.63 146.626  150.61 < 2.2e-16 ***
## Residuals 111 108.07   0.974
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lm(P_F~Energy_fat, data = data3))
```

```
## Analysis of Variance Table
##
## Response: P_F
##             Df Sum Sq Mean Sq F value    Pr(>F)
```

```
## Energy_fat   1 114.96 114.959  91.319 3.774e-16 ***
## Residuals  111 139.73   1.259
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm(P_F~Energy_protein, data = data3))
```

```
## Analysis of Variance Table
##
## Response: P_F
##                 Df  Sum Sq Mean Sq F value    Pr(>F)
## Energy_protein   1  52.879  52.879  29.084 3.952e-07 ***
## Residuals      111 201.813   1.818
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm(P_F~Total_energy, data = data3))
```

```
## Analysis of Variance Table
##
## Response: P_F
##               Df  Sum Sq Mean Sq F value   Pr(>F)
## Total_energy   1  94.755  94.755  65.762 7.45e-13 ***
## Residuals    111 159.938   1.441
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# all are significant

summary(lm(P_F~Sex, data = data3)) # Adjusted R-squared:  0.03802
```

```
##
## Call:
## lm(formula = P_F ~ Sex, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9081 -1.0216 -0.2955  0.5033  6.9068
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.9456     0.1959   15.04   <2e-16 ***
## Sexm         -0.6483     0.2783   -2.33   0.0216 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.479 on 111 degrees of freedom
## Multiple R-squared:  0.04661,    Adjusted R-squared:  0.03802
## F-statistic: 5.427 on 1 and 111 DF,  p-value: 0.02164
```

```r
summary(lm(P_F~Wet_wt, data = data3)) # Adjusted R-squared:  0.224
```

```
##
## Call:
## lm(formula = P_F ~ Wet_wt, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6715 -0.9126 -0.1749  0.3538  6.9165
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.4402     0.3384  13.120  < 2e-16 ***
## Wet_wt       -1.9484     0.3375  -5.774 7.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.328 on 111 degrees of freedom
## Multiple R-squared:  0.231,  Adjusted R-squared:  0.224
## F-statistic: 33.34 on 1 and 111 DF,  p-value: 7.158e-08
```

```r
summary(lm(P_F~Dry_wt, data = data3)) # Adjusted R-squared:  0.2973
```

```
##
## Call:
## lm(formula = P_F ~ Dry_wt, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6177 -0.8343 -0.1100  0.3087  6.7062
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.4174     0.2839  15.560  < 2e-16 ***
## Dry_wt       -5.9995     0.8625  -6.956 2.56e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.264 on 111 degrees of freedom
## Multiple R-squared:  0.3036, Adjusted R-squared:  0.2973
## F-statistic: 48.38 on 1 and 111 DF,  p-value: 2.561e-10
```

```r
summary(lm(P_F~Ashed_wt, data = data3)) # Adjusted R-squared:  0.04432
```

```
##
## Call:
## lm(formula = P_F ~ Ashed_wt, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6942 -0.9553 -0.4548  0.4113  7.3167
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.4551     0.3615   9.558 3.74e-16 ***
## Ashed_wt    -18.3158     7.3595  -2.489   0.0143 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.474 on 111 degrees of freedom
## Multiple R-squared:  0.05285,    Adjusted R-squared:  0.04432
## F-statistic: 6.194 on 1 and 111 DF,  p-value: 0.0143
```

```
summary(lm(P_F~Protein + Fat, data = data3)) # Adjusted R-squared:  0.4895
```

```
##
## Call:
## lm(formula = P_F ~ Protein + Fat, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9630 -0.6123 -0.3845  0.2133  5.6189
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.2257     0.3089  10.443  < 2e-16 ***
## Protein       9.3827     2.9118   3.222  0.00167 **
## Fat         -24.8072     3.1058  -7.987 1.46e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.077 on 110 degrees of freedom
## Multiple R-squared:  0.4986, Adjusted R-squared:  0.4895
## F-statistic:  54.7 on 2 and 110 DF,  p-value: < 2.2e-16
```

```
summary(lm(P_F~Pfat, data = data3)) #Adjusted R-squared:  0.7854
```

```
##
## Call:
## lm(formula = P_F ~ Pfat, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1555 -0.3889 -0.1671  0.1623  4.2706
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.649456   0.209190   31.79   <2e-16 ***
## Pfat        -0.152525   0.007525  -20.27   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6987 on 111 degrees of freedom
## Multiple R-squared:  0.7873, Adjusted R-squared:  0.7854
## F-statistic: 410.8 on 1 and 111 DF,  p-value: < 2.2e-16
```

```r
summary(lm(P_F~Pprotein, data = data3)) # Adjusted R-squared:  0.5719
```

```
##
## Call:
## lm(formula = P_F ~ Pprotein, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5389 -0.6073 -0.1505  0.3330  5.0889
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.94122    0.86592  -9.171 2.89e-15 ***
## Pprotein     0.18341    0.01494  12.272  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9867 on 111 degrees of freedom
## Multiple R-squared:  0.5757, Adjusted R-squared:  0.5719
## F-statistic: 150.6 on 1 and 111 DF,  p-value: < 2.2e-16
```

```r
summary(lm(P_F~Energy_fat, data = data3)) # Adjusted R-squared:  0.4464
```

```
##
## Call:
## lm(formula = P_F ~ Energy_fat, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4423 -0.7005 -0.2020  0.2393  6.0496
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.04566    0.18238  22.182  < 2e-16 ***
## Energy_fat  -0.41177    0.04309  -9.556 3.77e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.122 on 111 degrees of freedom
## Multiple R-squared:  0.4514, Adjusted R-squared:  0.4464
## F-statistic: 91.32 on 1 and 111 DF,  p-value: 3.774e-16
```

```r
summary(lm(P_F~Energy_protein, data = data3)) # Multiple R-squared:  0.2076
```

```
##
## Call:
## lm(formula = P_F ~ Energy_protein, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5793 -0.8335 -0.2517  0.3506  7.0257
```

```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.3487     0.3440  12.642  < 2e-16 ***
## Energy_protein  -0.4374     0.0811  -5.393 3.95e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.348 on 111 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.2005
## F-statistic: 29.08 on 1 and 111 DF,  p-value: 3.952e-07
```

```
summary(lm(P_F~Total_energy, data = data3)) # Adjusted R-squared:  0.3664
```

```
##
## Call:
## lm(formula = P_F ~ Total_energy, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6255 -0.8193 -0.1199  0.3587  6.4405
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.3754     0.2437  17.956  < 2e-16 ***
## Total_energy  -0.2368     0.0292  -8.109 7.45e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.2 on 111 degrees of freedom
## Multiple R-squared:  0.372,  Adjusted R-squared:  0.3664
## F-statistic: 65.76 on 1 and 111 DF,  p-value: 7.45e-13
```

```
cor.test(data3$Wet_wt, data3$Dry_wt)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data3$Wet_wt and data3$Dry_wt
## t = 49.152, df = 111, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9678868 0.9846632
## sample estimates:
##       cor
## 0.9777901
```

```
data_ready <- data3[,-c(1:3,5:8,11,22:24,26:31)]

# since both sex and tank is significant, we need to regress them out first
# then scale the residuals for log regression
```

```r
model.tank_sex <- lm(data_ready$P_F ~ data_ready$Sex + data_ready$Tank)
anova(model.tank_sex)
```

```
## Analysis of Variance Table
##
## Response: data_ready$P_F
##                 Df  Sum Sq Mean Sq F value   Pr(>F)
## data_ready$Sex    1  11.872 11.8719  7.7463 0.006938 **
## data_ready$Tank  42 137.072  3.2636  2.1295 0.002606 **
## Residuals        69 105.748  1.5326
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
resid(model.tank_sex) -> data_ready$resid

library(caret)
```

```
## Loading required package: lattice
```

```r
data_stand <- preProcess(data_ready, method = c("center", "scale", "YeoJohnson","nzv"))
data_transformed <- predict(data_stand, data_ready)
rm.out<-function(x)(ifelse(x >3, NA,x ))

dev.new()
pairs(data_ready)

model.5 <- lm(resid ~ poly(Pfat,2), data_ready)
plot(data_ready$Pfat, data_ready$resid)
lines(data_ready$Pfat, predict(model.5),col="red")
summary(model.5)
```

```
##
## Call:
## lm(formula = resid ~ poly(Pfat, 2), data = data_ready)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5866 -0.3846  0.1873  0.5059  1.5164
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     9.221e-18  7.116e-02   0.000        1
## poly(Pfat, 2)1 -4.685e+00  7.564e-01  -6.194 1.04e-08 ***
## poly(Pfat, 2)2  4.568e+00  7.564e-01   6.039 2.15e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7564 on 110 degrees of freedom
## Multiple R-squared:  0.4049, Adjusted R-squared:  0.394
## F-statistic: 37.42 on 2 and 110 DF,  p-value: 4.015e-13
```

```r
#fit_exp <- glm(data_ready$resid ~ Pfat, data = data_ready, family = gaussian(link=log))
#plot(fit_exp)
# regress out the factor is not going to work, with residuals and the map look to be much more noiser

# how about look femanle and male fish seperately?
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
data_female <- filter(data_ready, Sex =="f")
data_male <- filter(data_ready, Sex =="m")

p <- ggplot(data_ready,aes(Pfat,P_F,colour = Sex)) +
  geom_point() +
  ylab("Protein: fat ratio in the fish body") +
  xlab("Percentage of fat in the fish body") +
  ggtitle("Distribution of female and male fish")

p


fit_exp.f <- glm(P_F ~ Pfat, data = data_female, family = gaussian(link=log))
plot(fit_exp.f)

fit_exp.m <- glm(P_F ~ Pfat, data = data_male, family = gaussian(link=log))
plot(fit_exp.m)

model.5 <- lm(resid ~ poly(Pfat,2), data_ready)
plot(data_ready$Pfat, data_ready$resid)


library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

##
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
# pairs plot to visualize data by sex (3 categories)

sexplot<-ggpairs(data_ready[,-3], aes(colour=Sex, alpha=0.8), title ="Pairs plot for fish dataset by se
  theme_grey(base_size = 8)
pdf("sexplot.pdf")
sexplot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
dev.off()
```

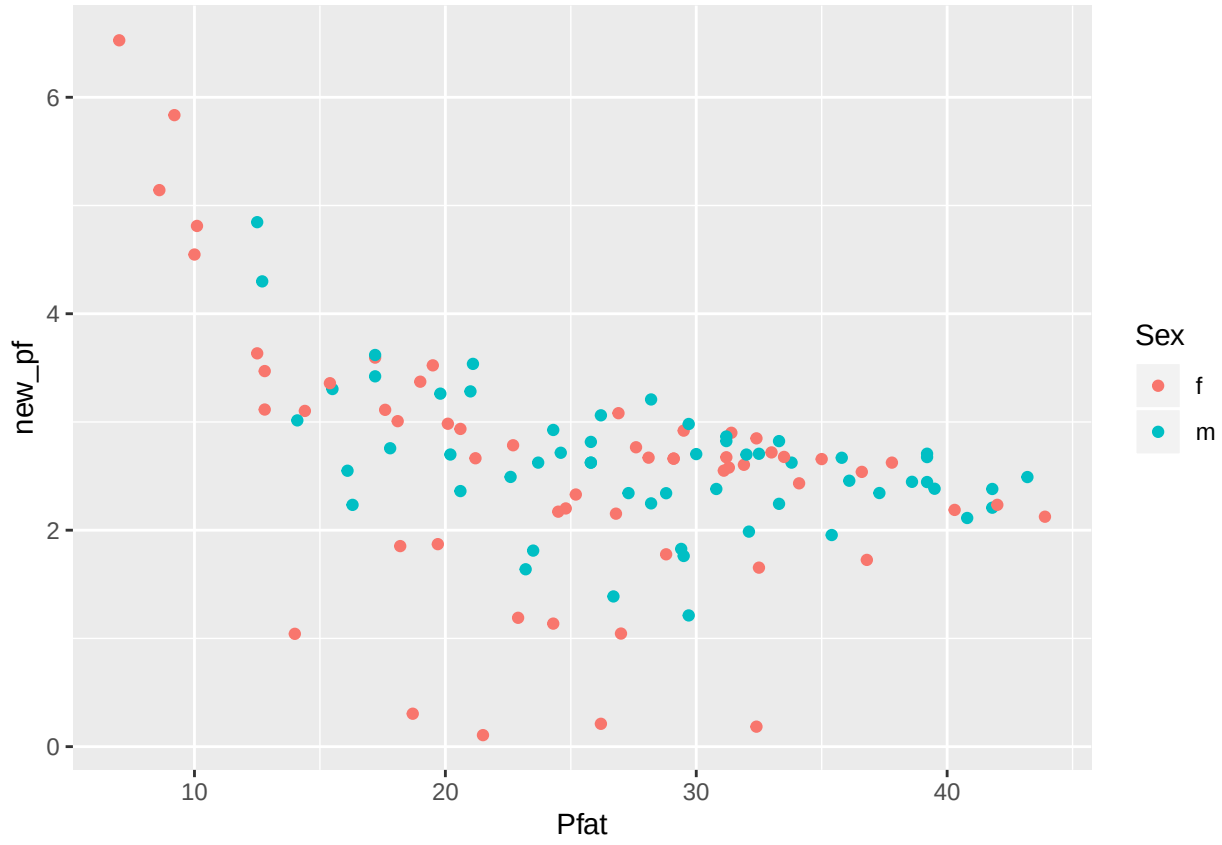
```
## pdf
##   2
```

try to adjust for tank

```
grand_mean <- mean(data_ready[["P_F"]])
grand_mean #[1] 2.62429
```

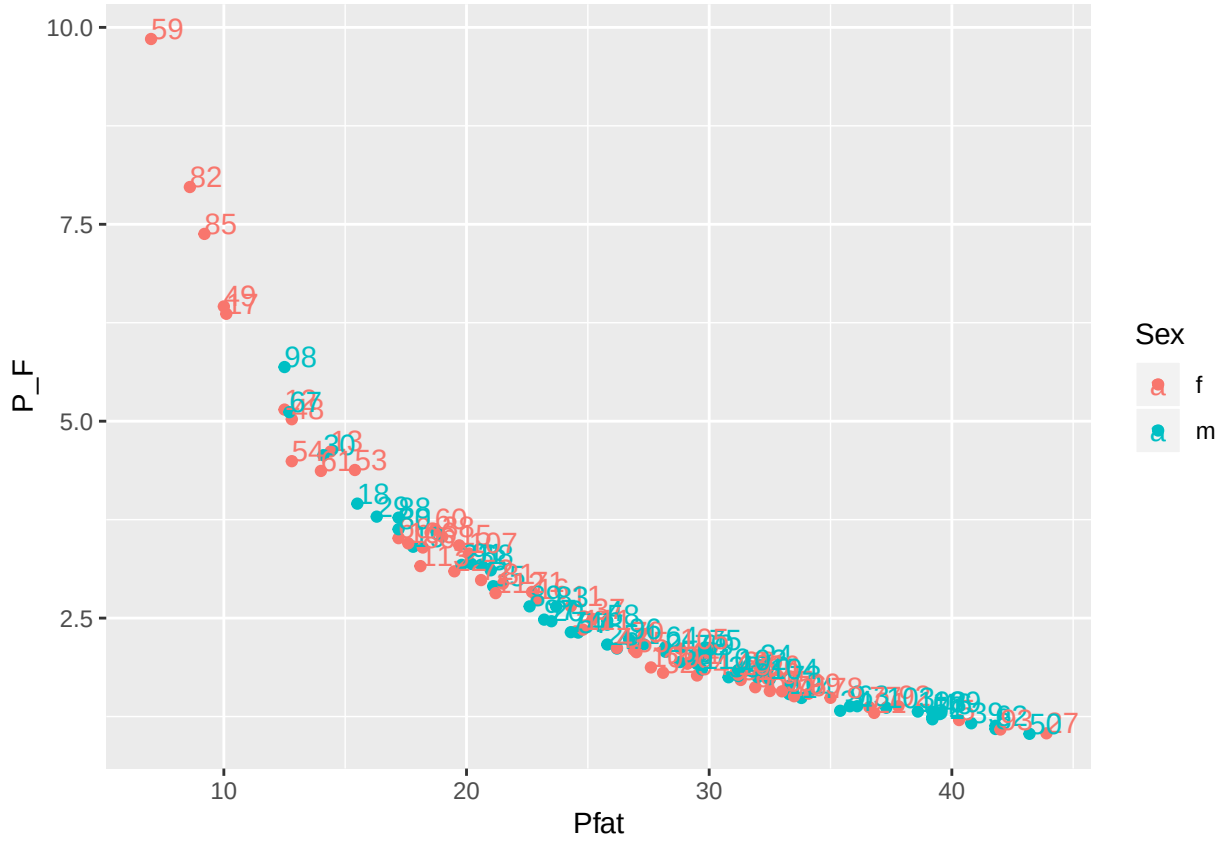
```
## [1] 2.62429
```

```r
# extract mean for each tank and then correct it
tank_level <- levels(data3$Tank)
tank_mean <- NULL

as.data.frame(aggregate(data_ready[,14], list(data_ready$Tank),mean)) -> tank_mean
names(tank_mean) <- c("tankid", "tank_mean")
tank_mean$grand_mean <- 2.62429
tank_mean$diff <- tank_mean$tank_mean - tank_mean$grand_mean

data_correction <- data_ready[,-15]


# goal is to put diff of tank_mean into the correlation datafrmae
data_correction <- merge(x=data_correction, y=tank_mean[,c(1,4)], by.x = "Tank", by.y = "tankid")
data_correction$new_pf <- data_correction$P_F - data_correction$diff # new_pf is the one thay has been

# plot it after correction
library(ggplot2)
p <- ggplot(data_correction,aes(Pfat,new_pf,colour = Sex)) + geom_point()
p
```

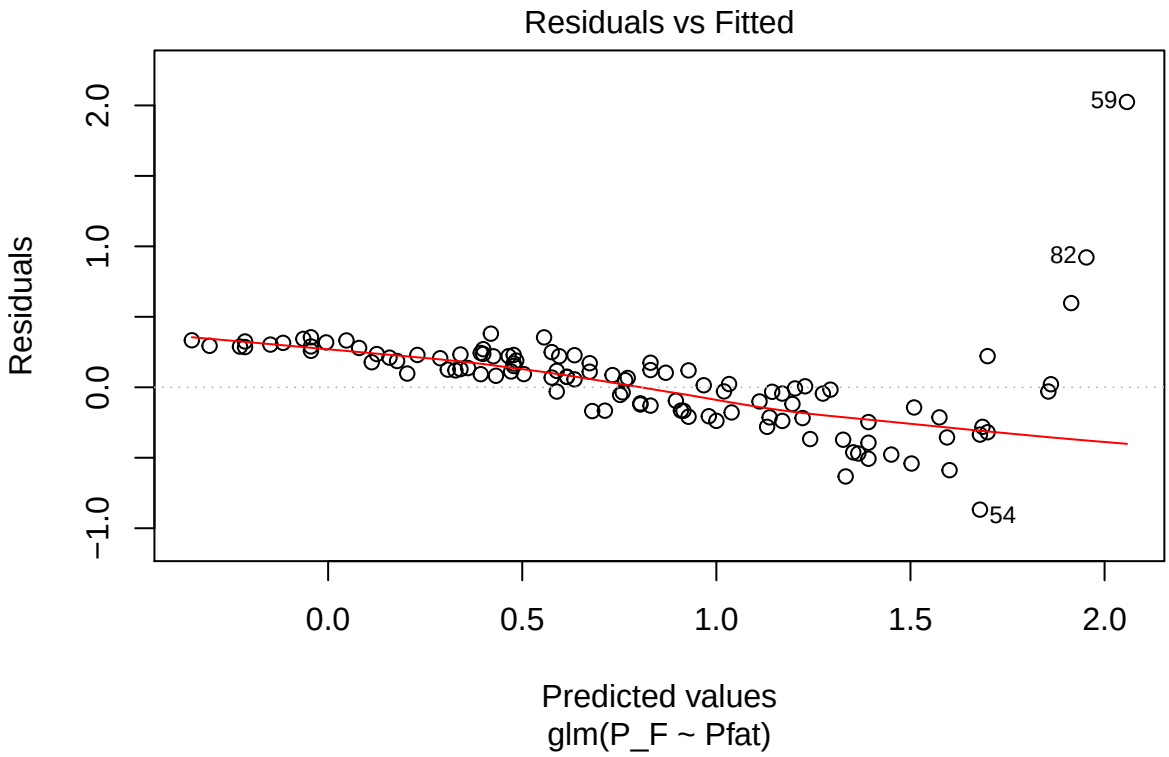


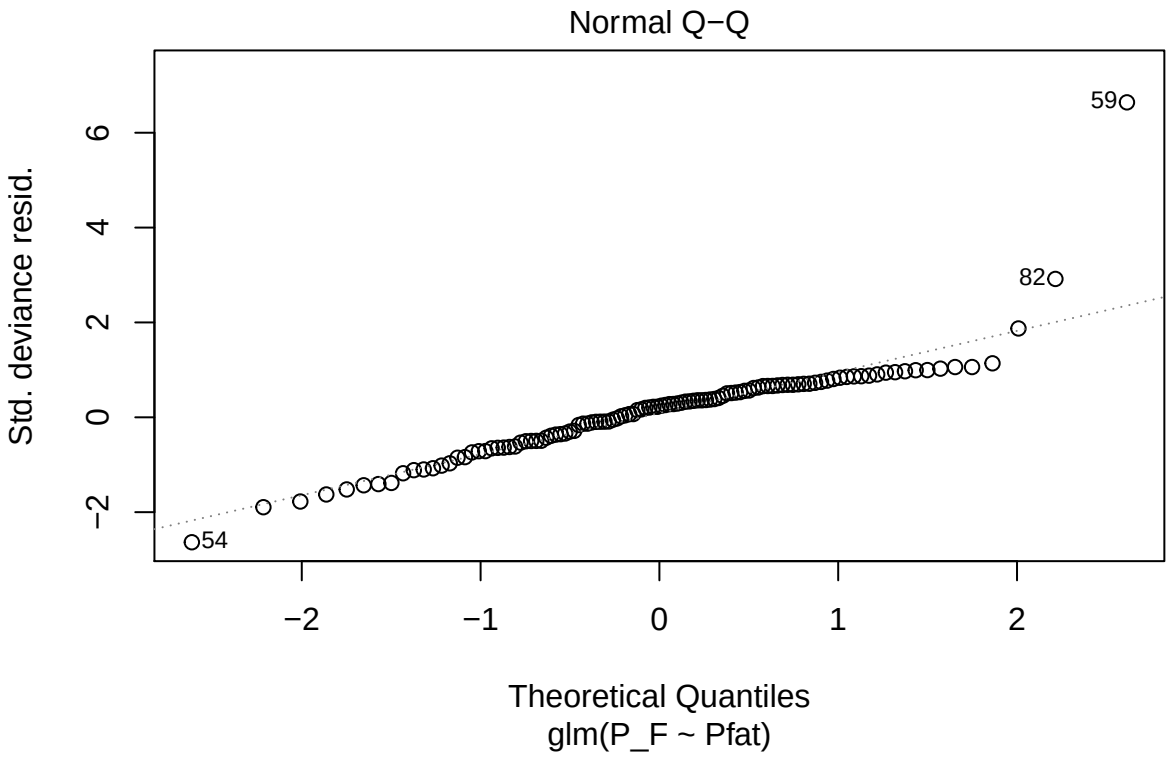

```r
# it is a distastet

# it seems that p_fat = 16.1, p_diff = 3.03 is an outlier it is data point 90, need to removeit
data_correction <- data_correction[-90,]
```

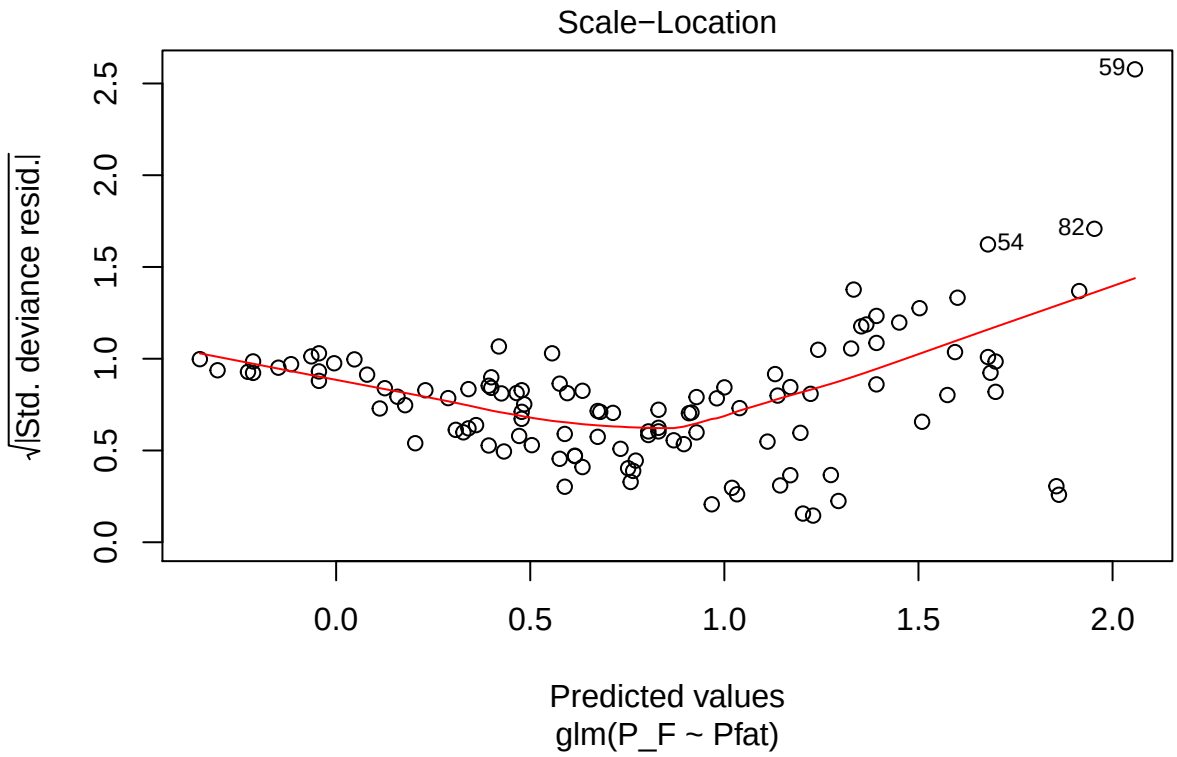
```
p <- ggplot(data_correction,aes(Pfat,P_F,colour = Sex)) + geom_point() + geom_text(aes(label=rownames(da
p
```

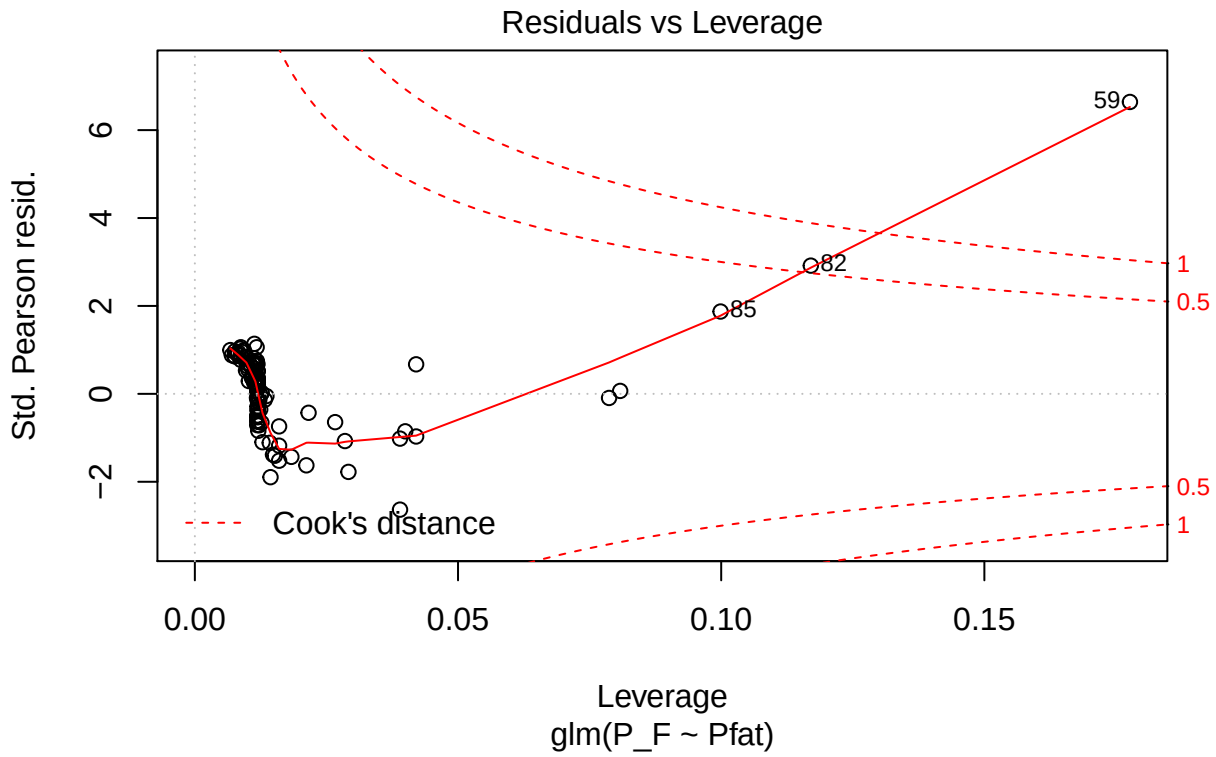



```
# now I am able to detect the two outliers, are they truely outliers????

# try differnt fit and compare them
data_f <- data_ready[data_ready$Sex=="f",]
data_m <- data_ready[data_ready$Sex=="m",]

# fit log link to it
fit_exp <- glm(P_F ~ Pfat, data = data_correction, family = gaussian(link=log))
plot(fit_exp)
```

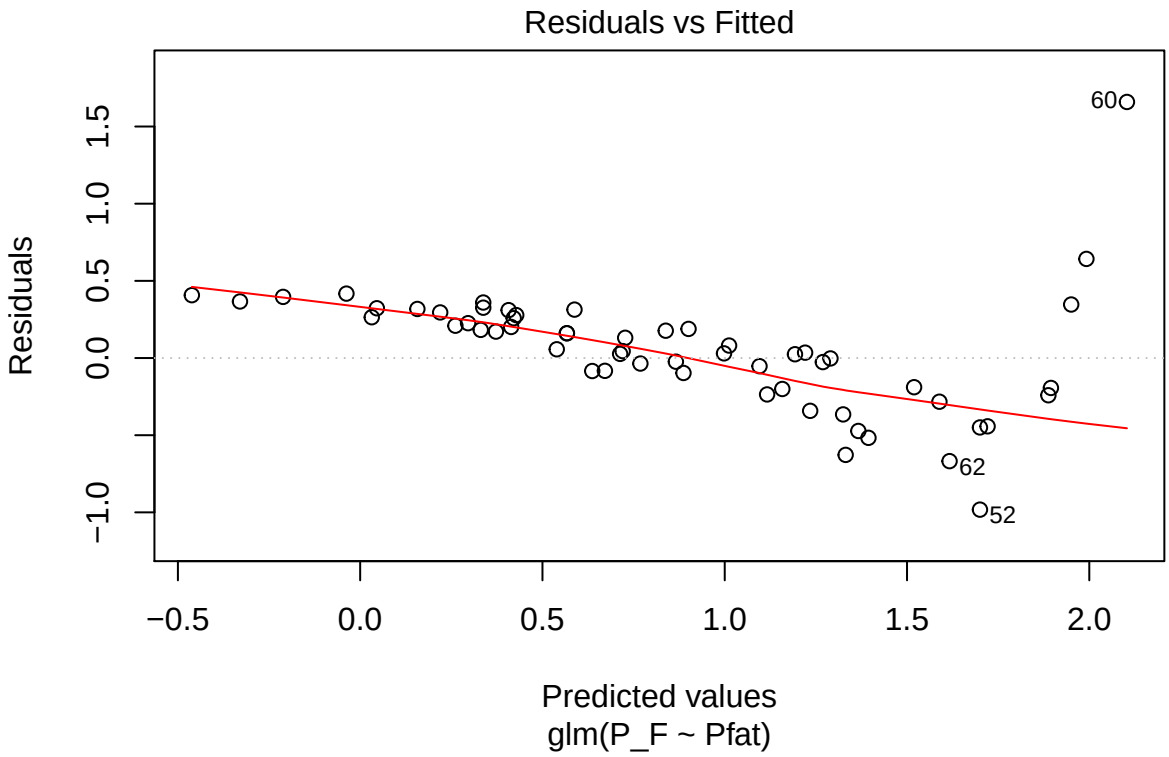

Residuals vs Fitted
Residuals
Predicted values
glm(P_F ~ Pfat)
59
82
54
2.0
1.0
0.0
−1.0
0.0
0.5
1.0
1.5
2.0

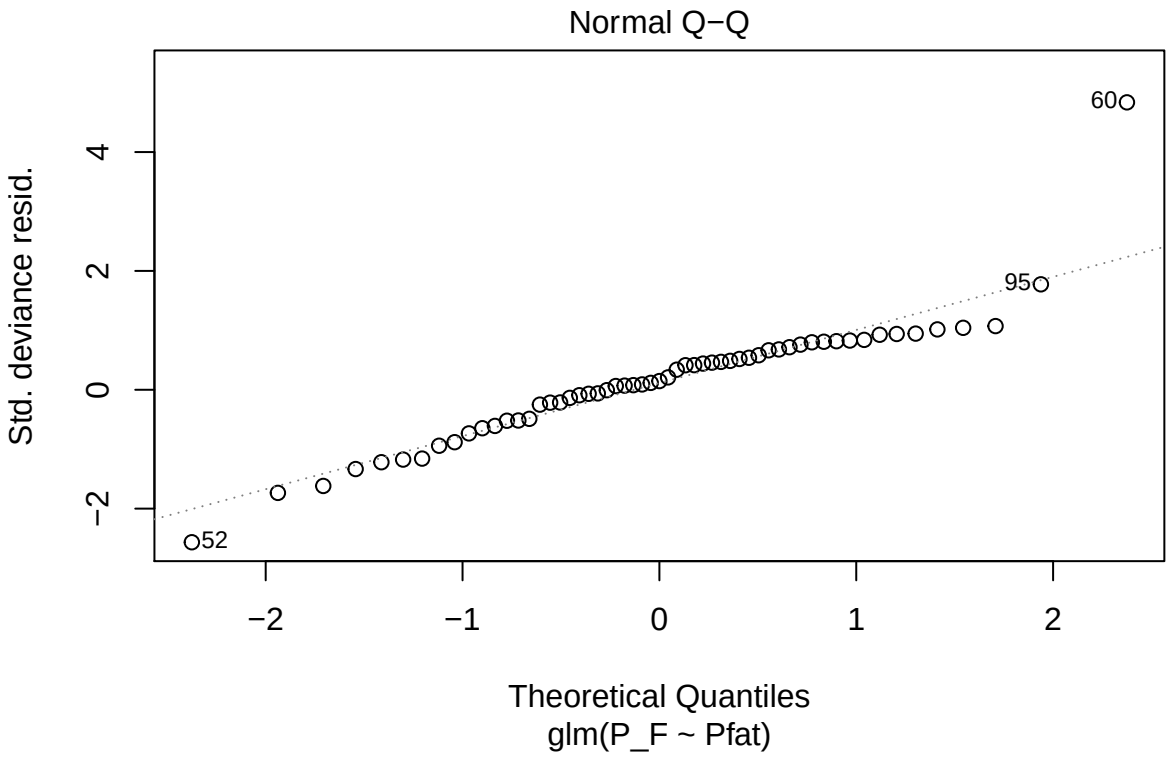

Normal Q−Q
Std. deviance resid.
Theoretical Quantiles
glm(P_F ~ Pfat)
59
82
54
−2
0
2
4
6
−2
−1
0
1
2

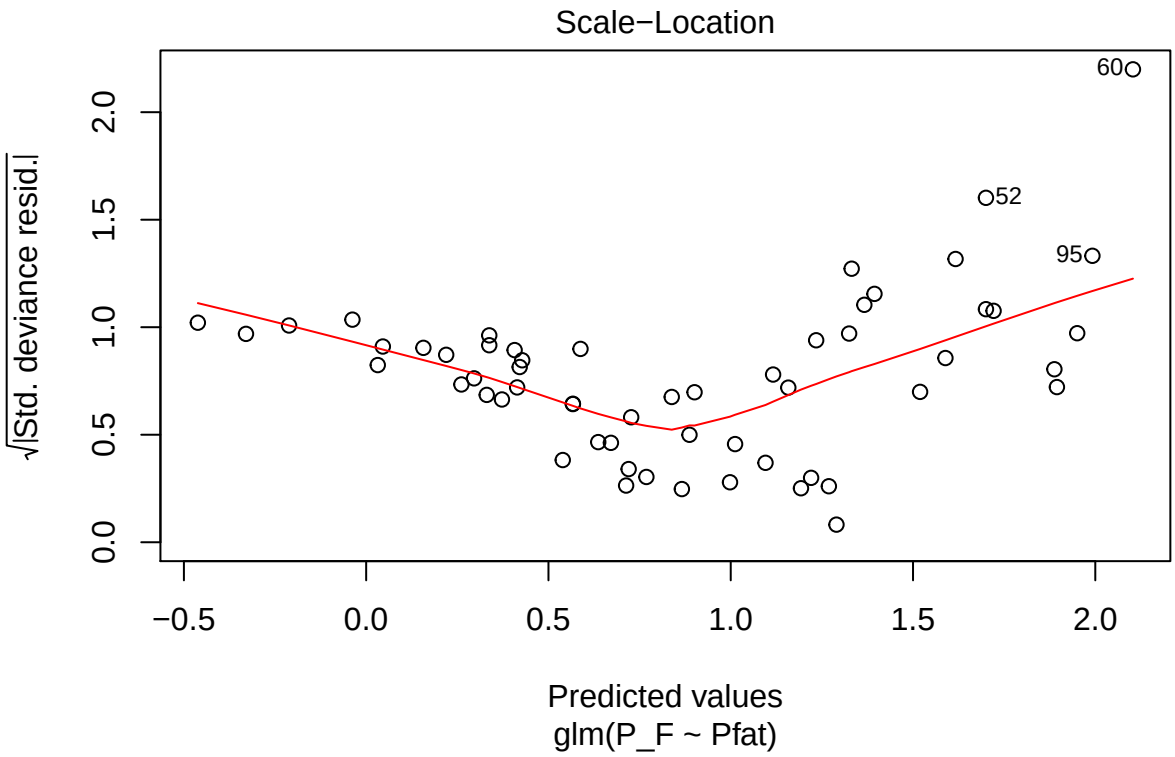

Scale−Location
√|Std. deviance resid.|
0.0
0.5
1.0
1.5
2.0
2.5
59
54
82
0.0
0.5
1.0
1.5
2.0
Predicted values
glm(P_F ~ Pfat)

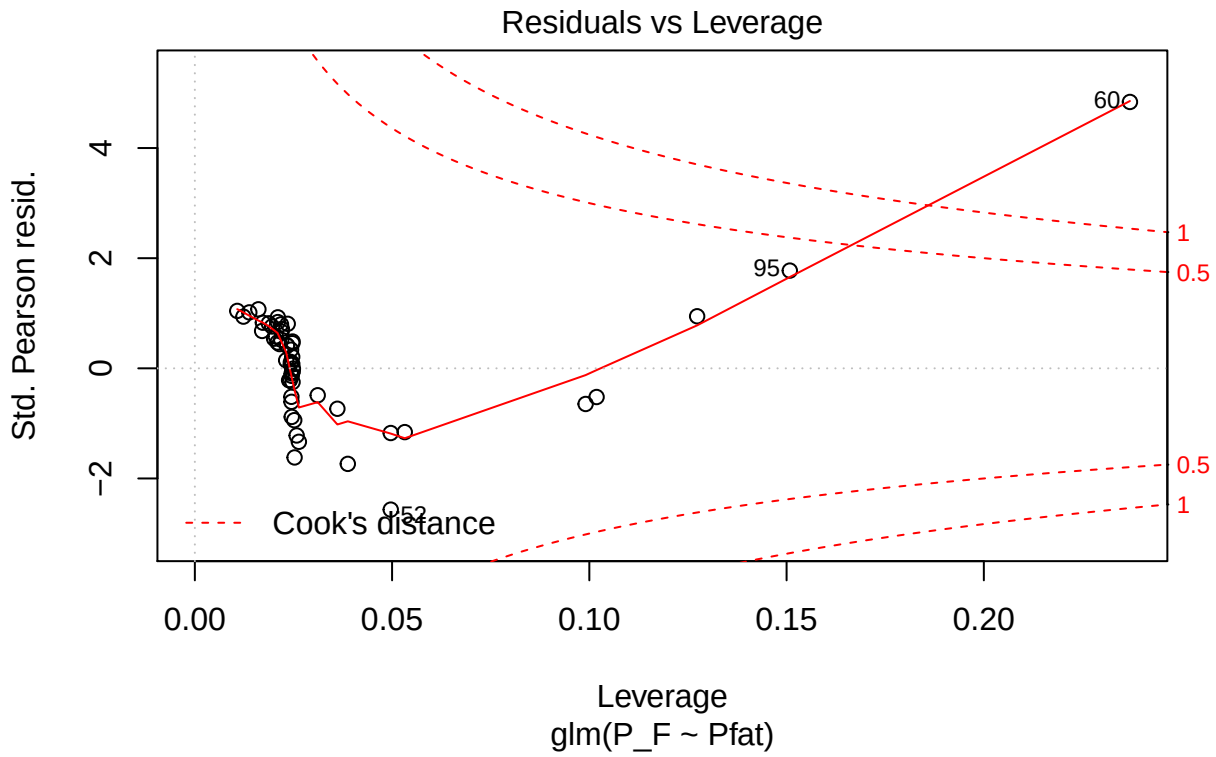



```
fit_exp_f <- glm(P_F ~ Pfat, data = data_f, family = gaussian(link=log))
plot(fit_exp_f)
```

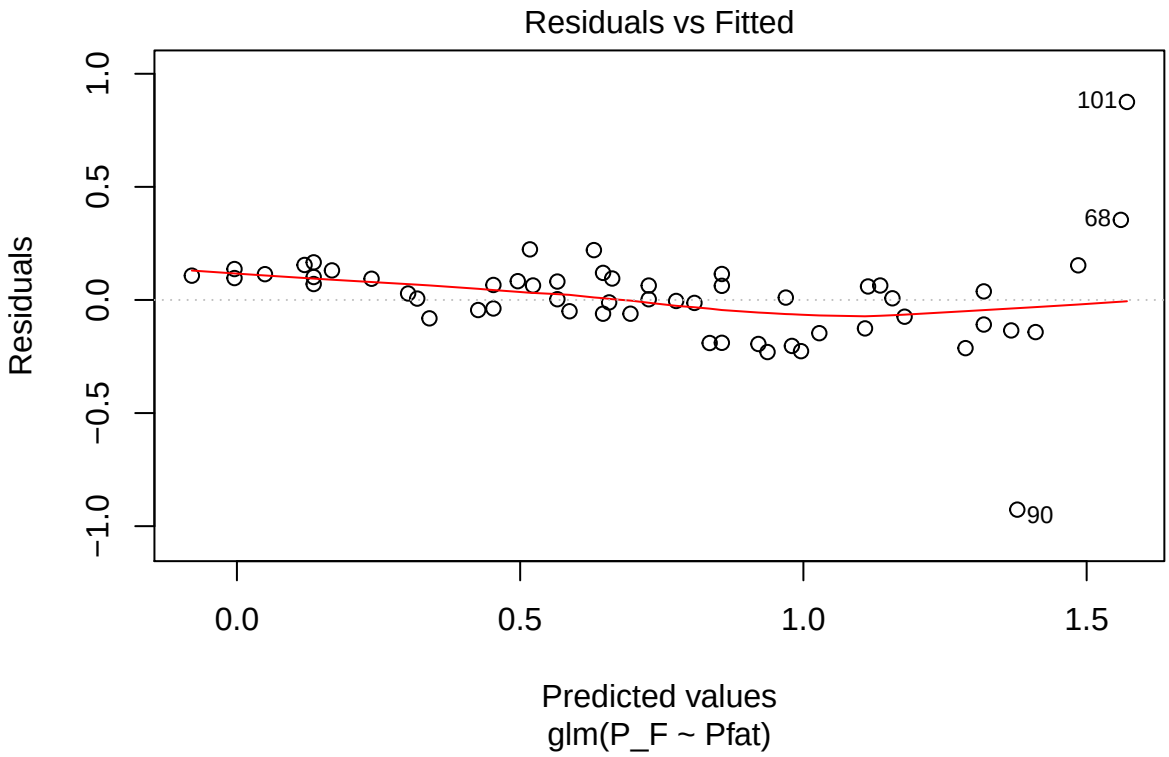

Residuals vs Fitted
Residuals
Predicted values
glm(P_F ~ Pfat)
60
62
52
−1.0
0.0
0.5
1.0
1.5
−0.5
0.0
0.5
1.0
1.5
2.0

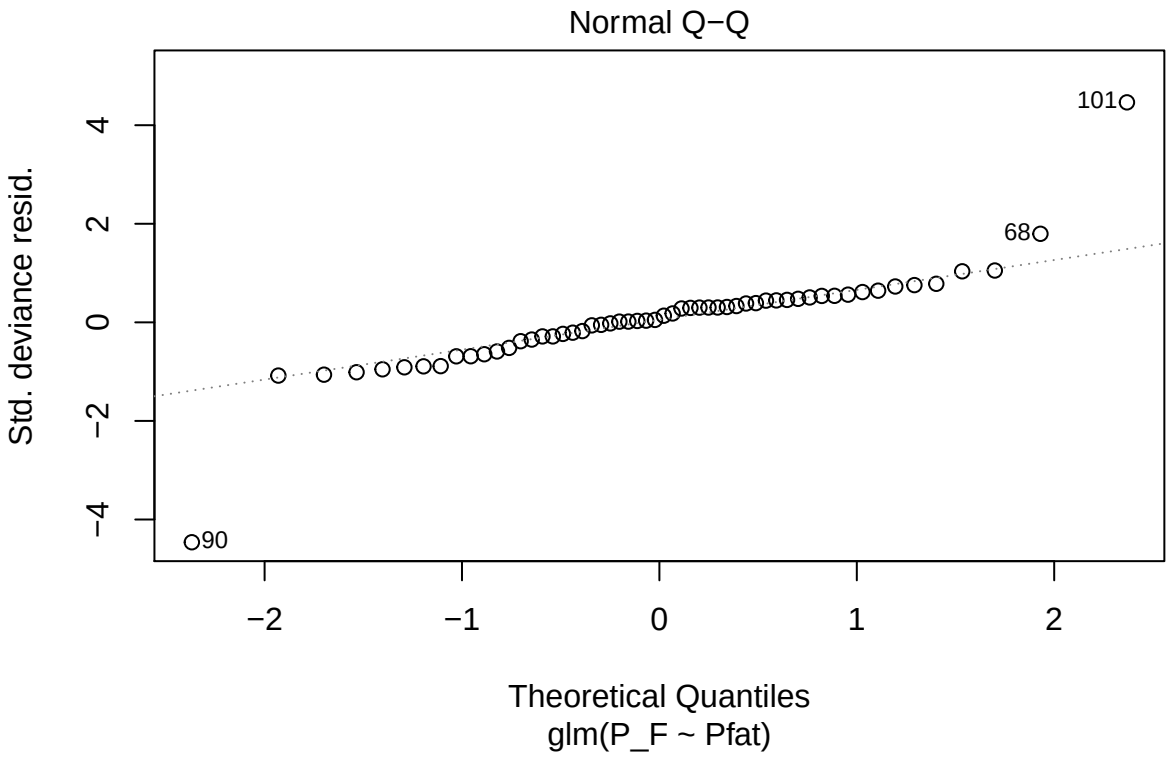

Normal Q−Q
Std. deviance resid.
Theoretical Quantiles
glm(P_F ~ Pfat)
60
95
52
−2
0
2
4
−2
−1
0
1
2

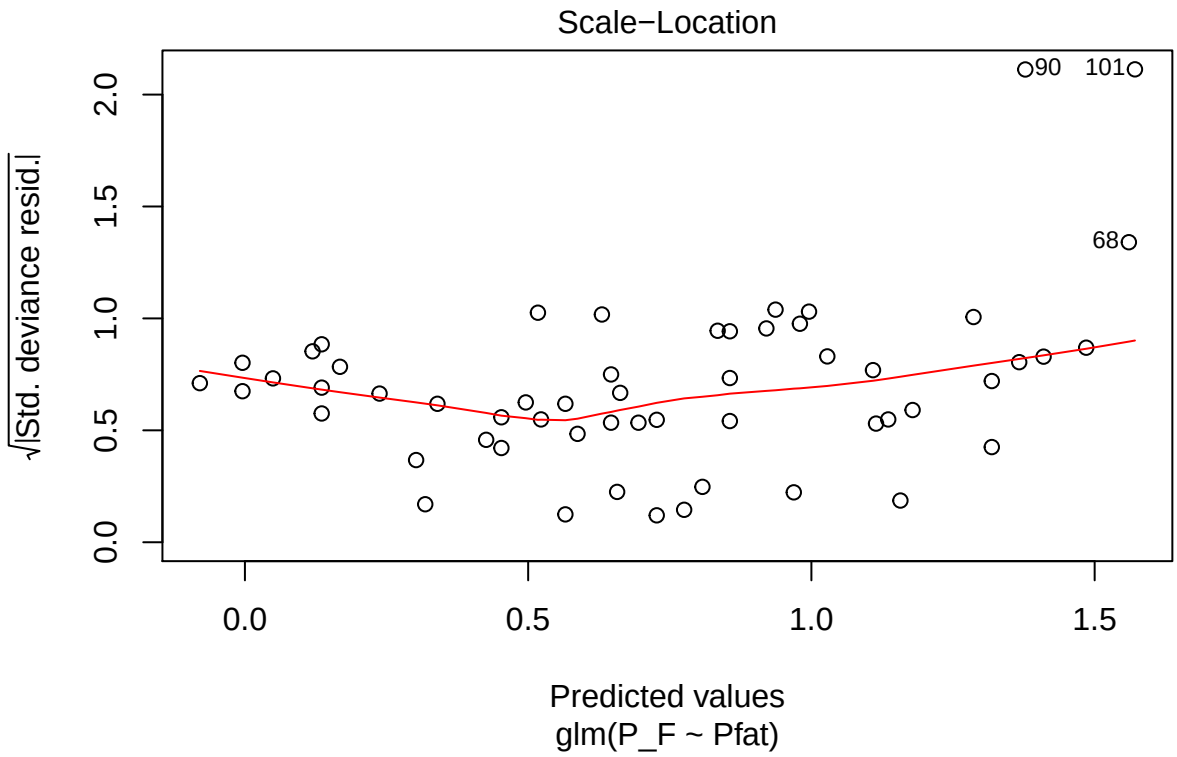

Scale−Location
√|Std. deviance resid.|
0.0
0.5
1.0
1.5
2.0
−0.5
0.0
0.5
1.0
1.5
2.0
60
52
95
Predicted values
glm(P_F ~ Pfat)

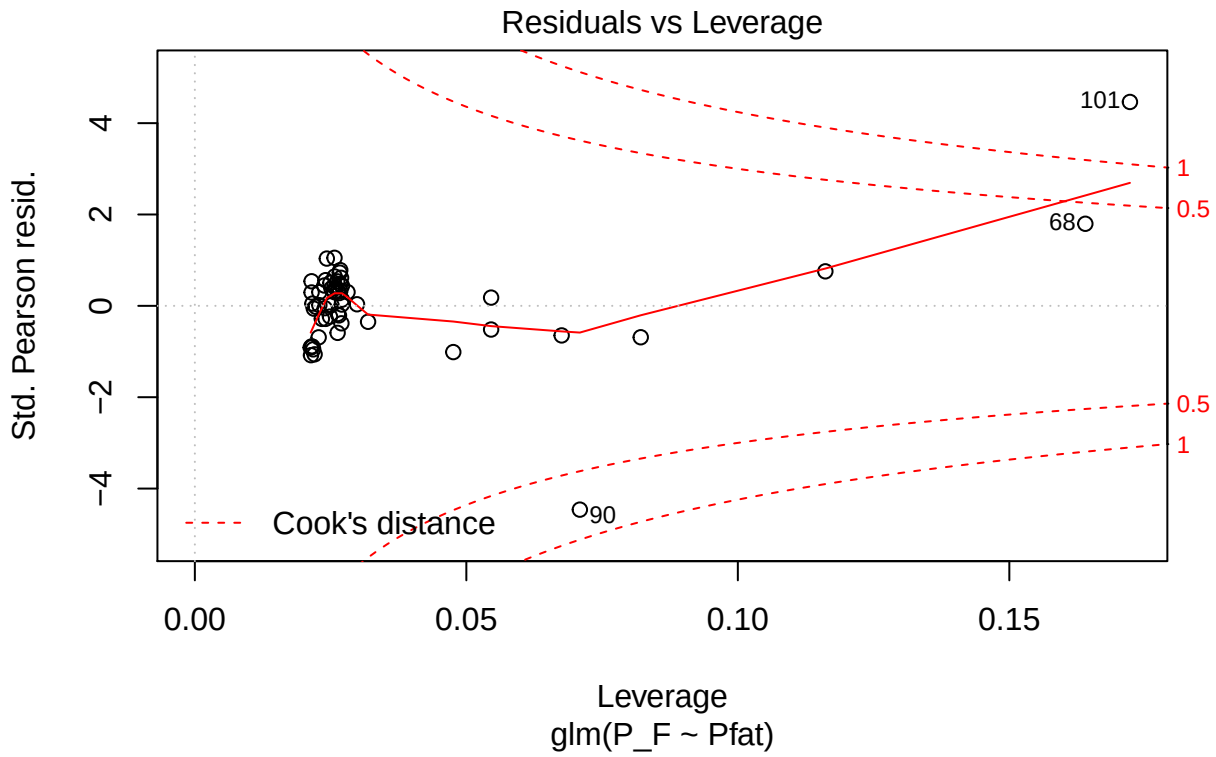



```
fit_exp_m <- glm(P_F ~ Pfat, data = data_m, family = gaussian(link=log))
plot(fit_exp_m)
```

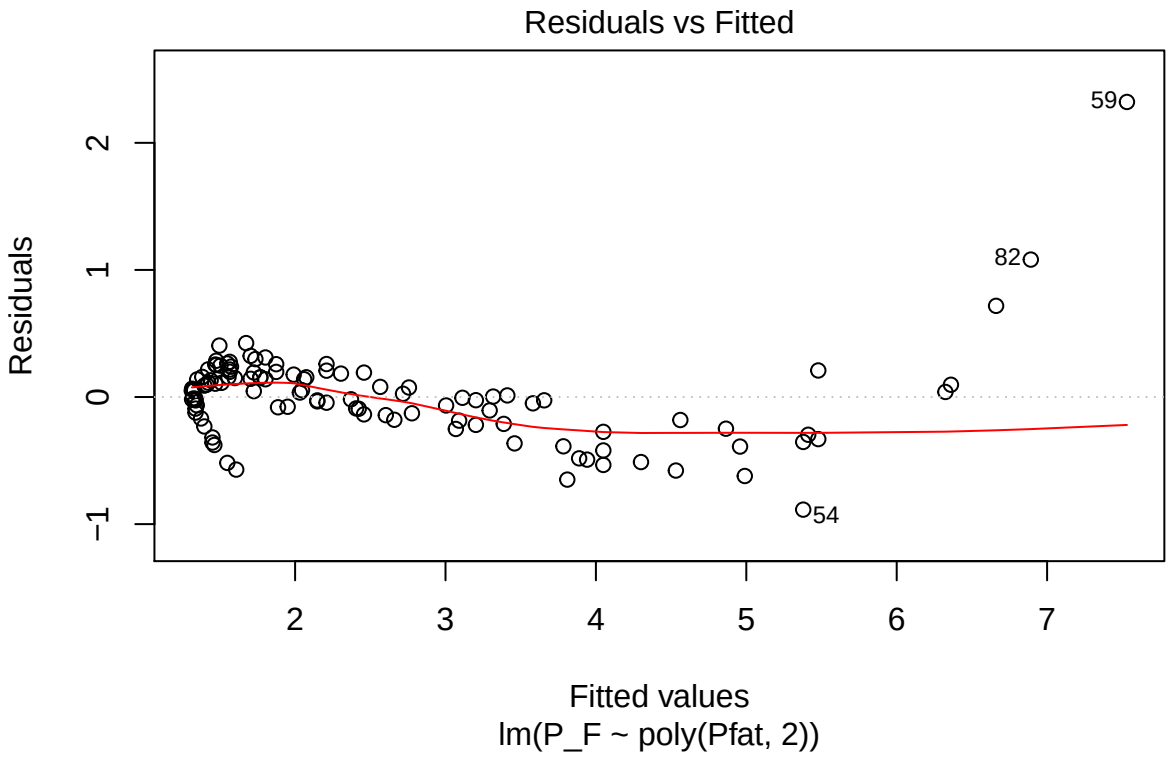

Residuals vs Fitted
Residuals
1.0
0.5
0.0
−0.5
−1.0
101
68
90
0.0
0.5
1.0
1.5
Predicted values
glm(P_F ~ Pfat)

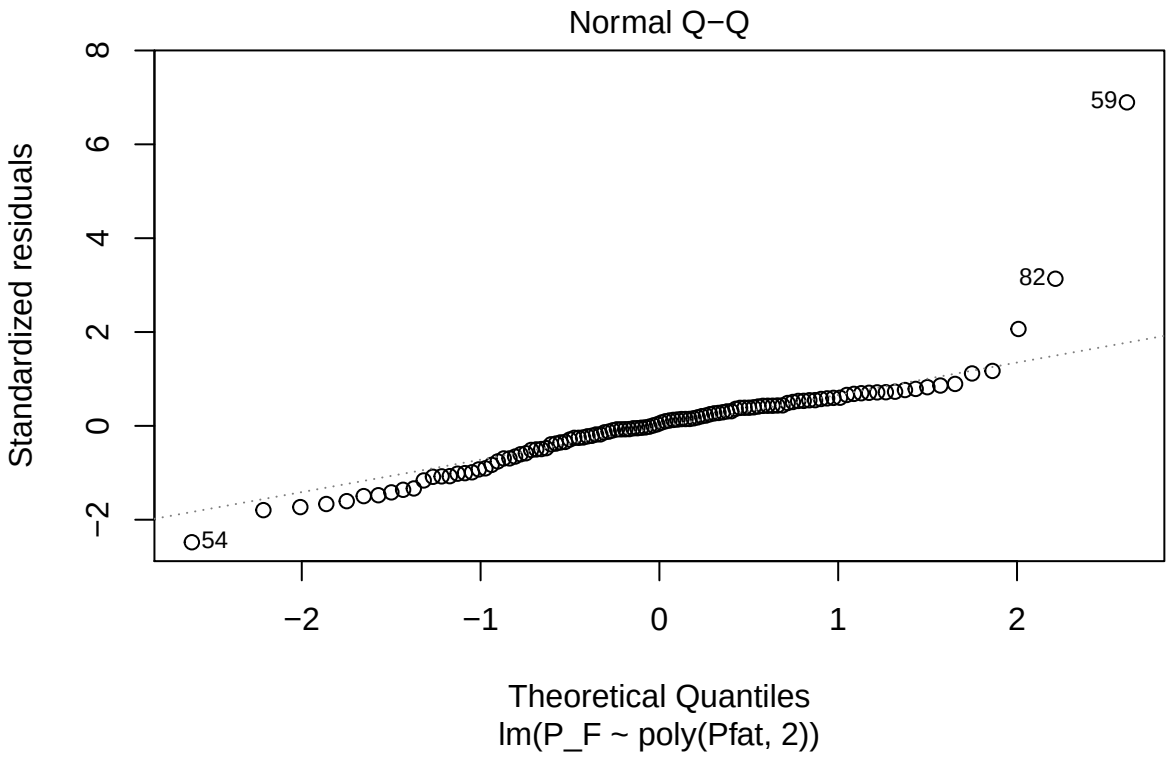

Normal Q−Q
Std. deviance resid.
Theoretical Quantiles
glm(P_F ~ Pfat)
101
68
90
−4
−2
0
2
4
−2
−1
0
1
2

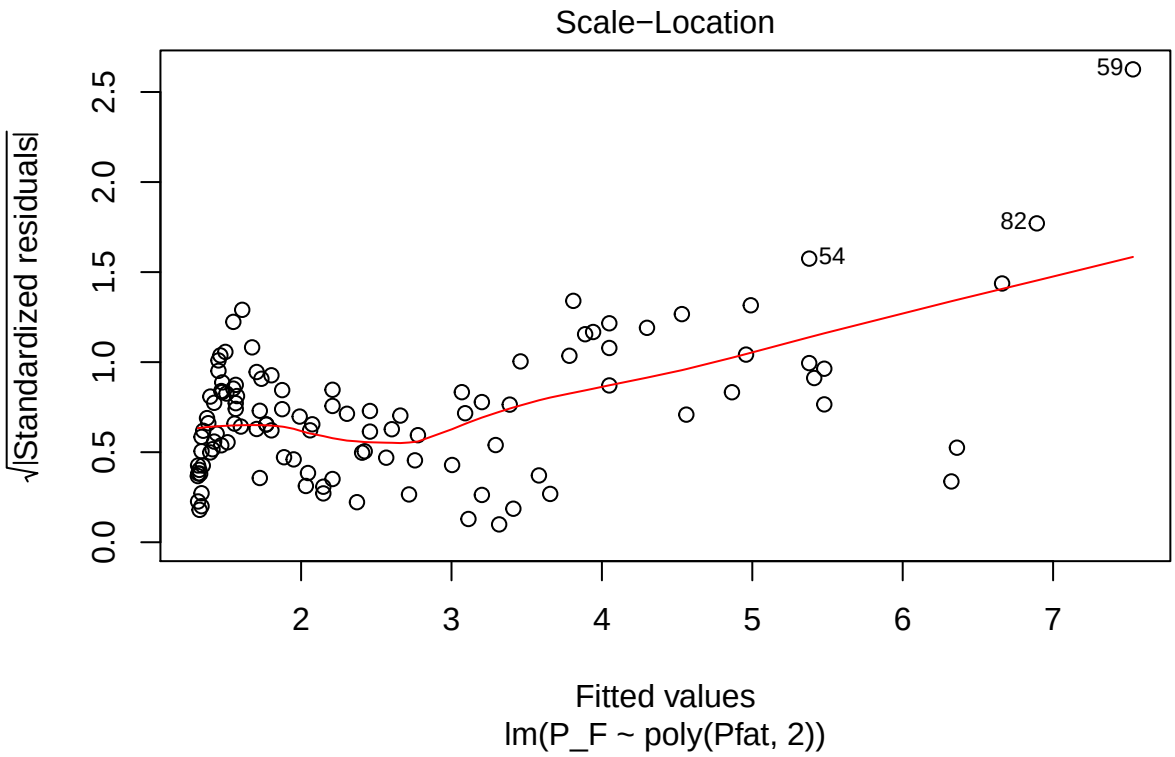

Scale−Location
90
101
68
√|Std. deviance resid.|
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
1.5
Predicted values
glm(P_F ~ Pfat)

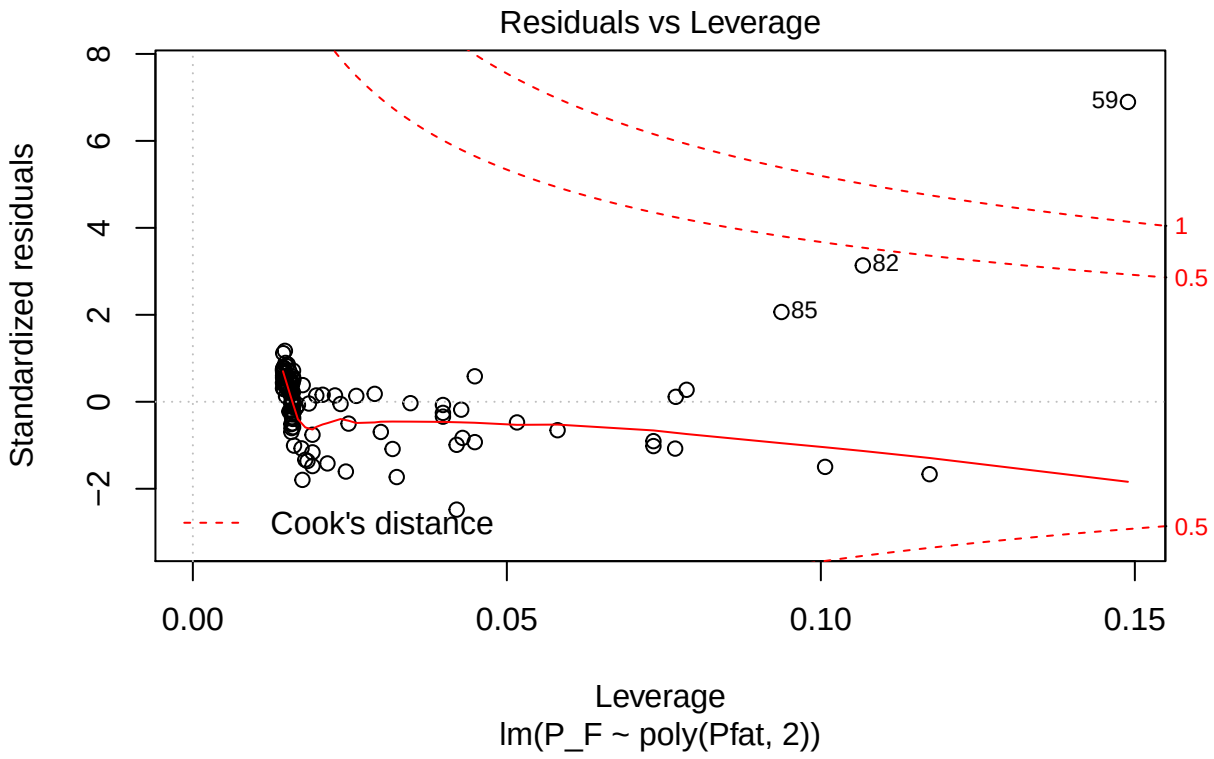



```
# poly-nomial
fit_poly <- lm(P_F~poly(Pfat,2), data_correction)
plot(fit_poly)
```


Residuals vs Fitted
Residuals
Fitted values
lm(P_F ~ poly(Pfat, 2))
59
82
54
2
1
0
−1
2
3
4
5
6
7


Normal Q−Q
Standardized residuals
8
6
4
2
0
−2
59
82
54
−2
−1
0
1
2
Theoretical Quantiles
lm(P_F ~ poly(Pfat, 2))

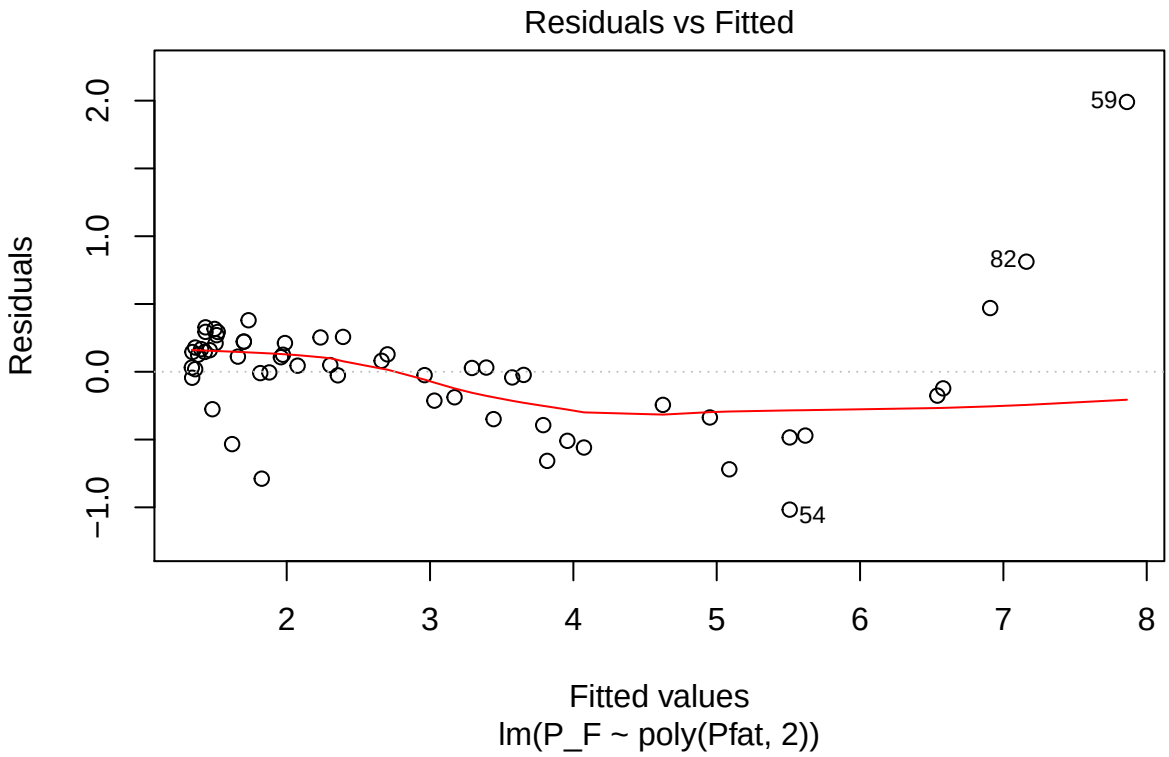

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
2
3
4
5
6
7
59
82
54
Fitted values
lm(P_F ~ poly(Pfat, 2))

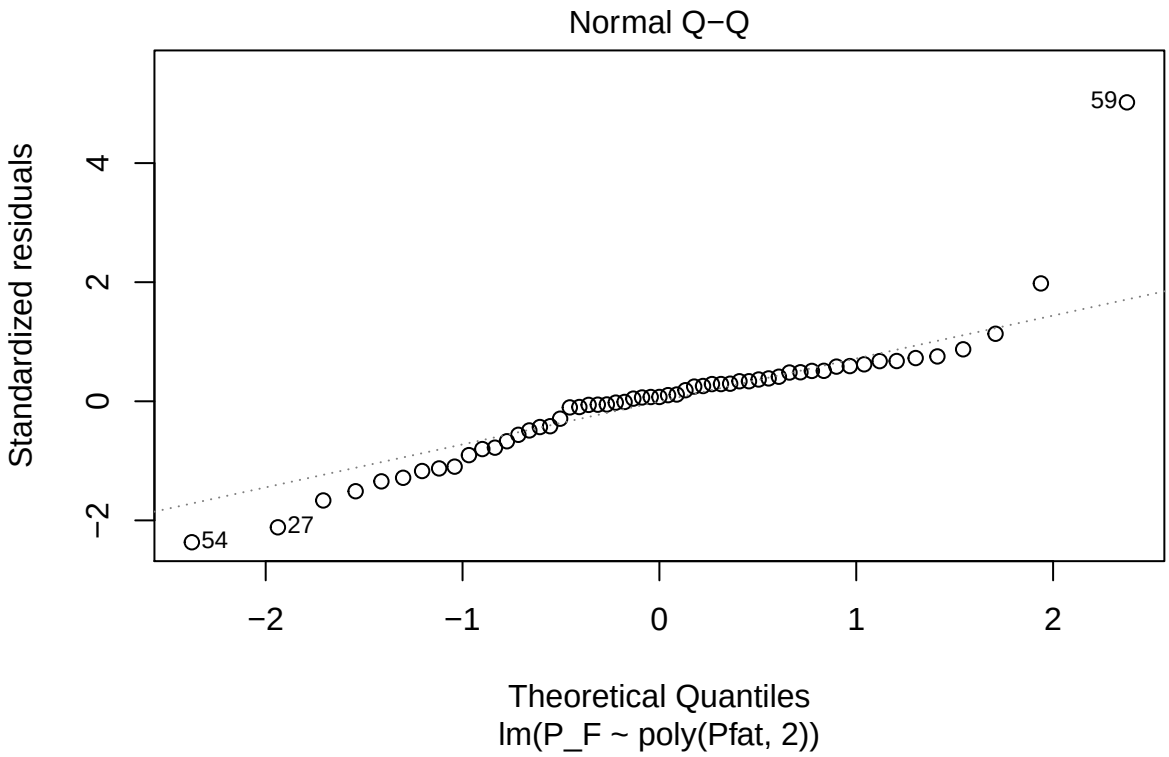


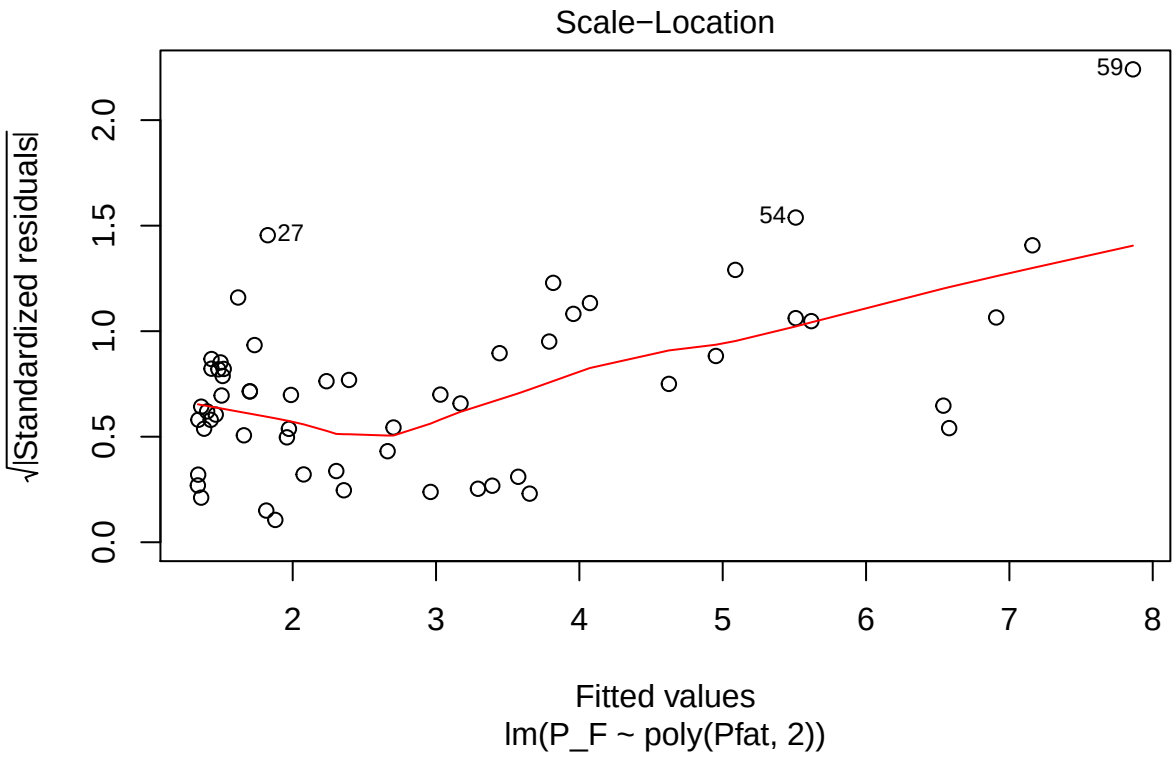

```
summary(fit_poly)
```

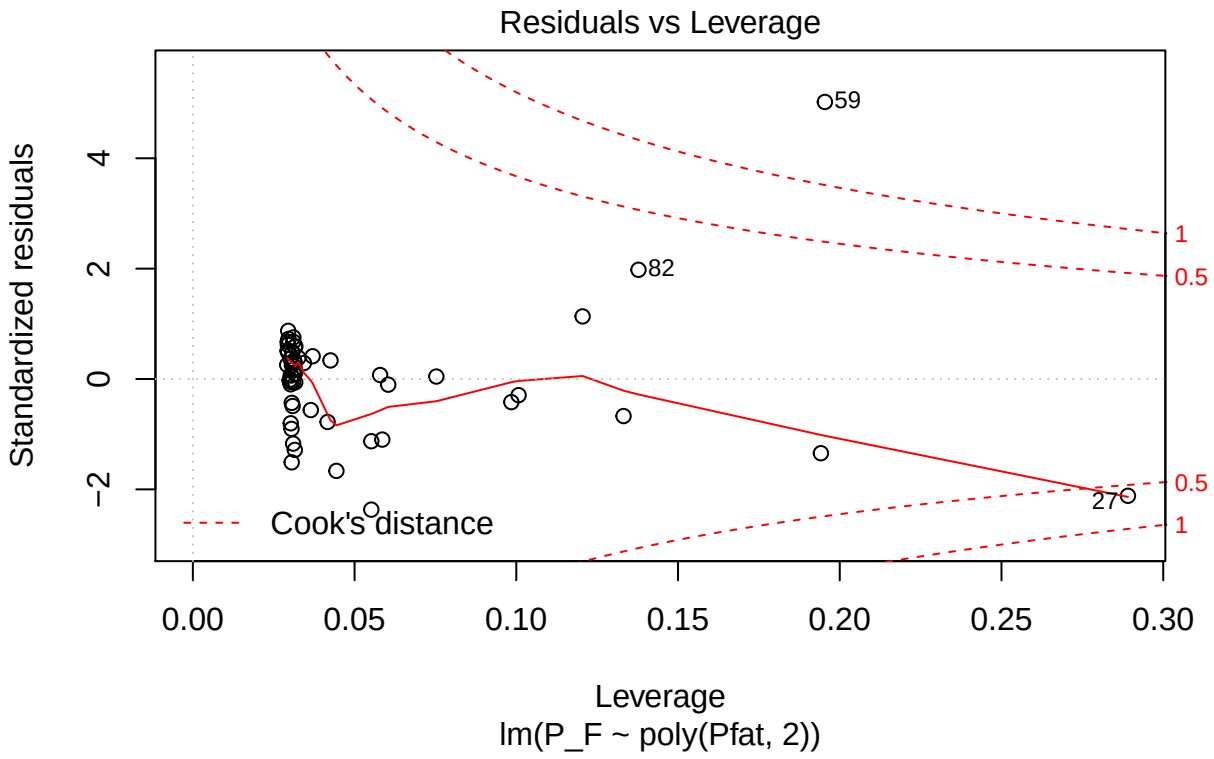
```
##
## Call:
## lm(formula = P_F ~ poly(Pfat, 2), data = data_correction)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.88596 -0.17953  0.01903  0.15745  2.32150
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.62059    0.03449   75.99   <2e-16 ***
## poly(Pfat, 2)1 -14.20221    0.36496  -38.91   <2e-16 ***
## poly(Pfat, 2)2   6.18859    0.36496   16.96   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.365 on 109 degrees of freedom
## Multiple R-squared:  0.943,  Adjusted R-squared:  0.9419
## F-statistic:   901 on 2 and 109 DF,  p-value: < 2.2e-16
```

```
fit_poly_f <- lm(P_F~poly(Pfat,2), data_correction[data_correction$Sex == "f", ])
plot(fit_poly_f)
```


Residuals vs Fitted
Residuals
2.0
1.0
0.0
−1.0
59
82
54
2
3
4
5
6
7
8
Fitted values
lm(P_F ~ poly(Pfat, 2))


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(P_F ~ poly(Pfat, 2))
59
27
54
−2
0
2
4
−2
−1
0
1
2

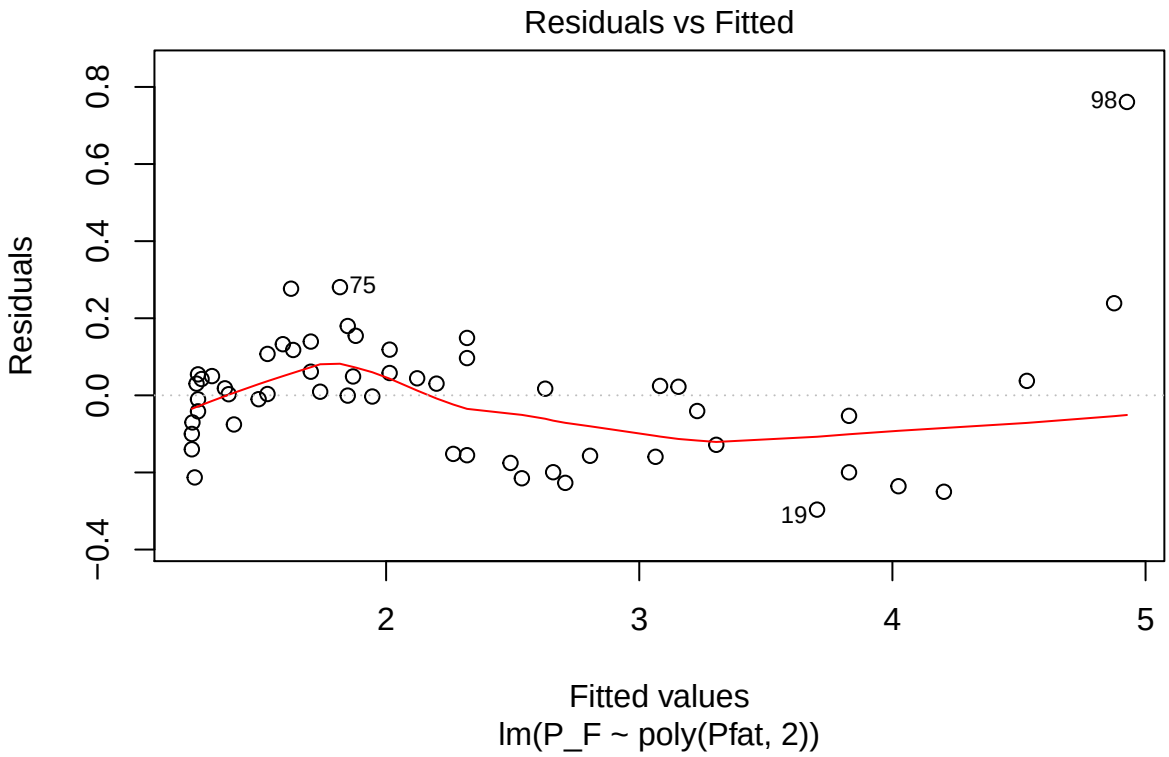

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2
3
4
5
6
7
8
27
54
59
Fitted values
lm(P_F ~ poly(Pfat, 2))

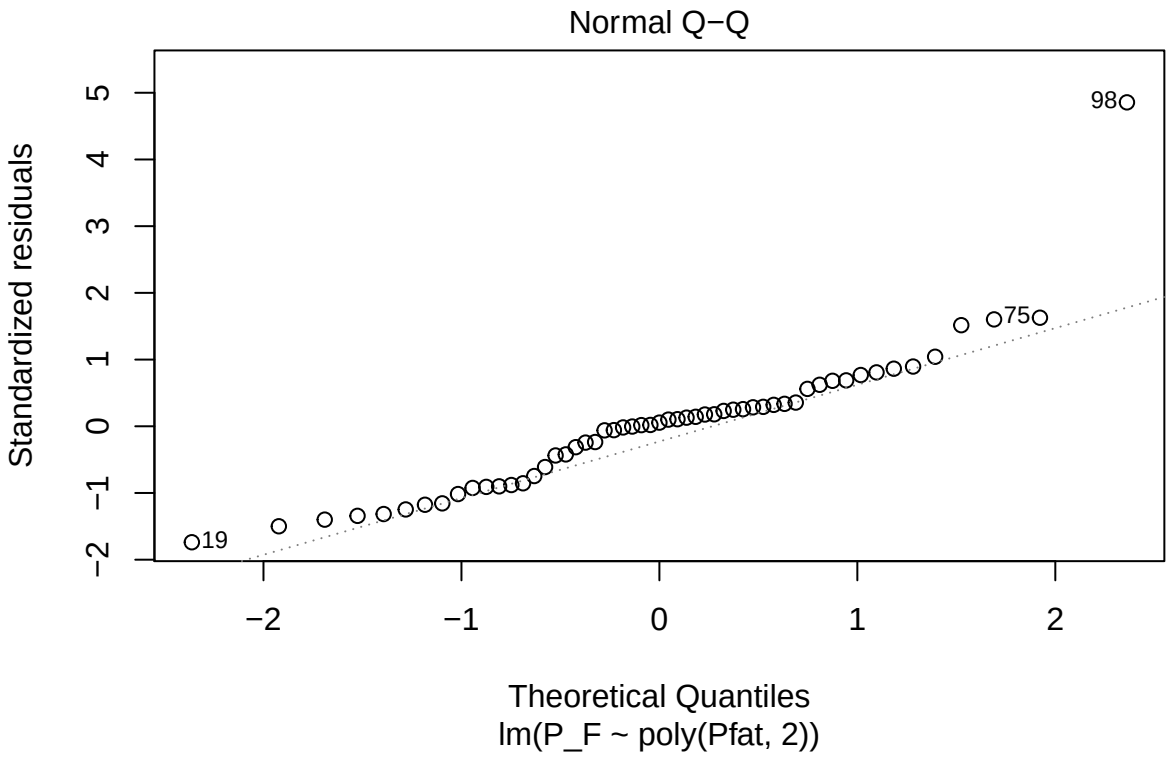


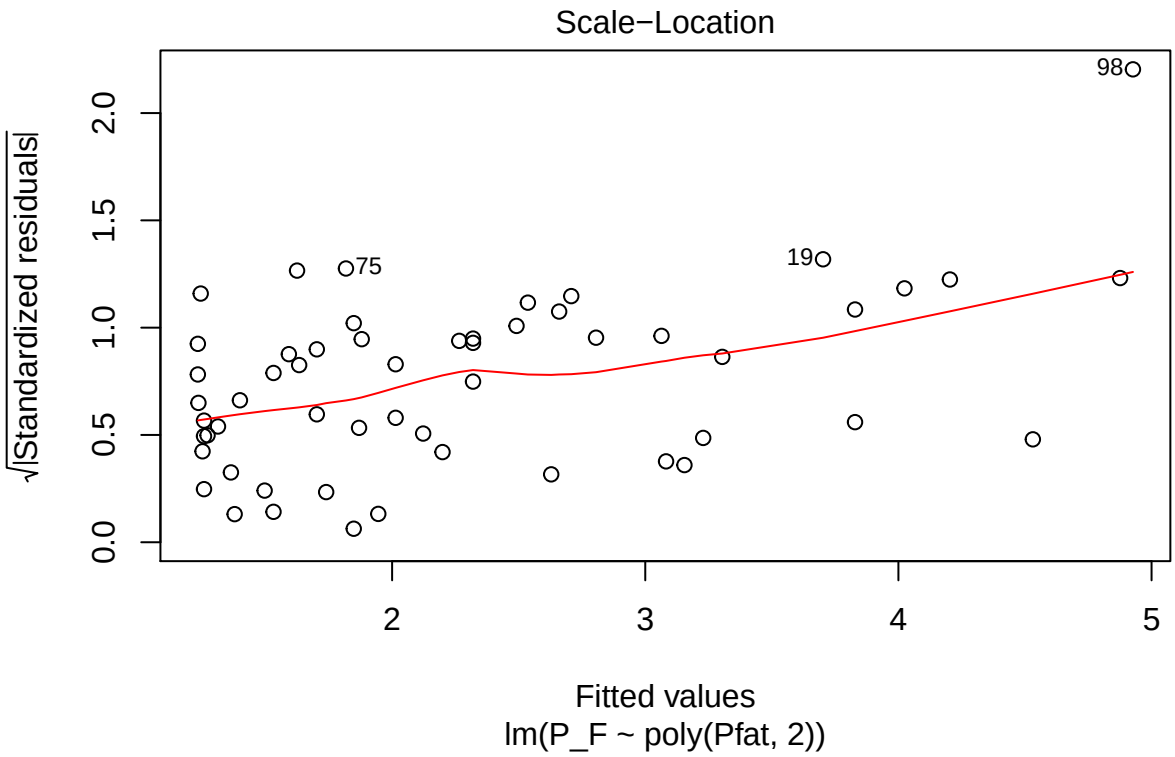

```
summary(fit_poly_f)
```

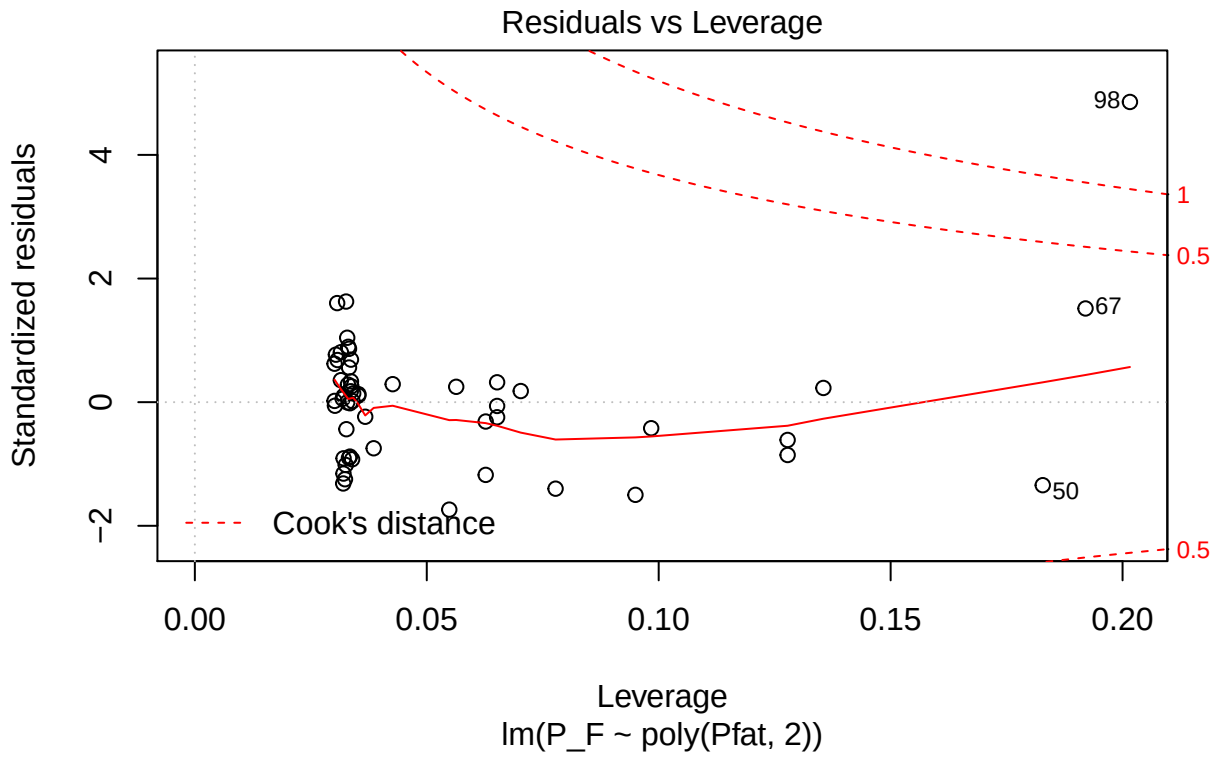
```
##
## Call:
## lm(formula = P_F ~ poly(Pfat, 2), data = data_correction[data_correction$Sex ==
##     "f", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.01689 -0.21296  0.03122  0.21045  1.99035
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.94556    0.05854   50.31   <2e-16 ***
## poly(Pfat, 2)1 -12.11912    0.44199  -27.42   <2e-16 ***
## poly(Pfat, 2)2   5.28207    0.44199   11.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.442 on 54 degrees of freedom
## Multiple R-squared:  0.9431, Adjusted R-squared:  0.941
## F-statistic: 447.3 on 2 and 54 DF,  p-value: < 2.2e-16
```

```
fit_poly_m <- lm(P_F~poly(Pfat,2), data_correction[data_correction$Sex == "m", ])
plot(fit_poly_m)
```


Residuals vs Fitted
Residuals
0.8
0.6
0.4
0.2
0.0
−0.4
98
75
19
2
3
4
5
Fitted values
lm(P_F ~ poly(Pfat, 2))


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(P_F ~ poly(Pfat, 2))
98
75
19
−2
−1
0
1
2
3
4
5

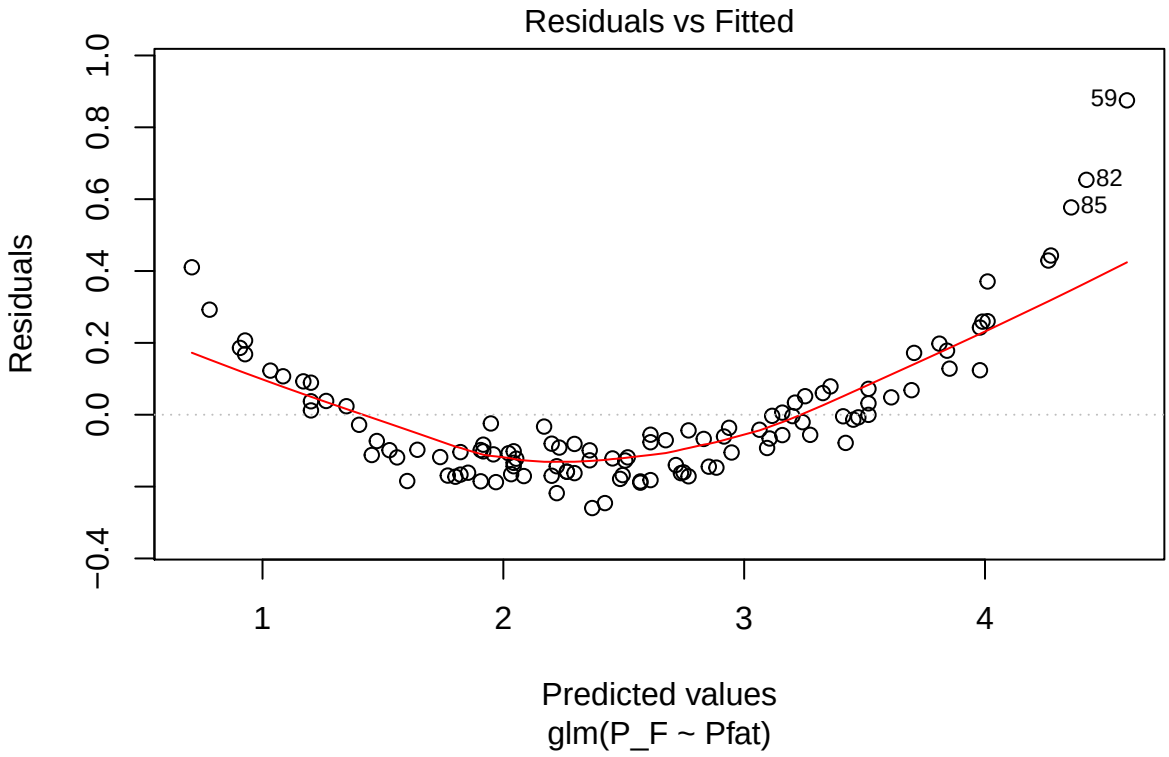

Scale−Location
98
75
19
0.0
0.5
1.0
1.5
2.0
√|Standardized residuals|
2
3
4
5
Fitted values
lm(P_F ~ poly(Pfat, 2))

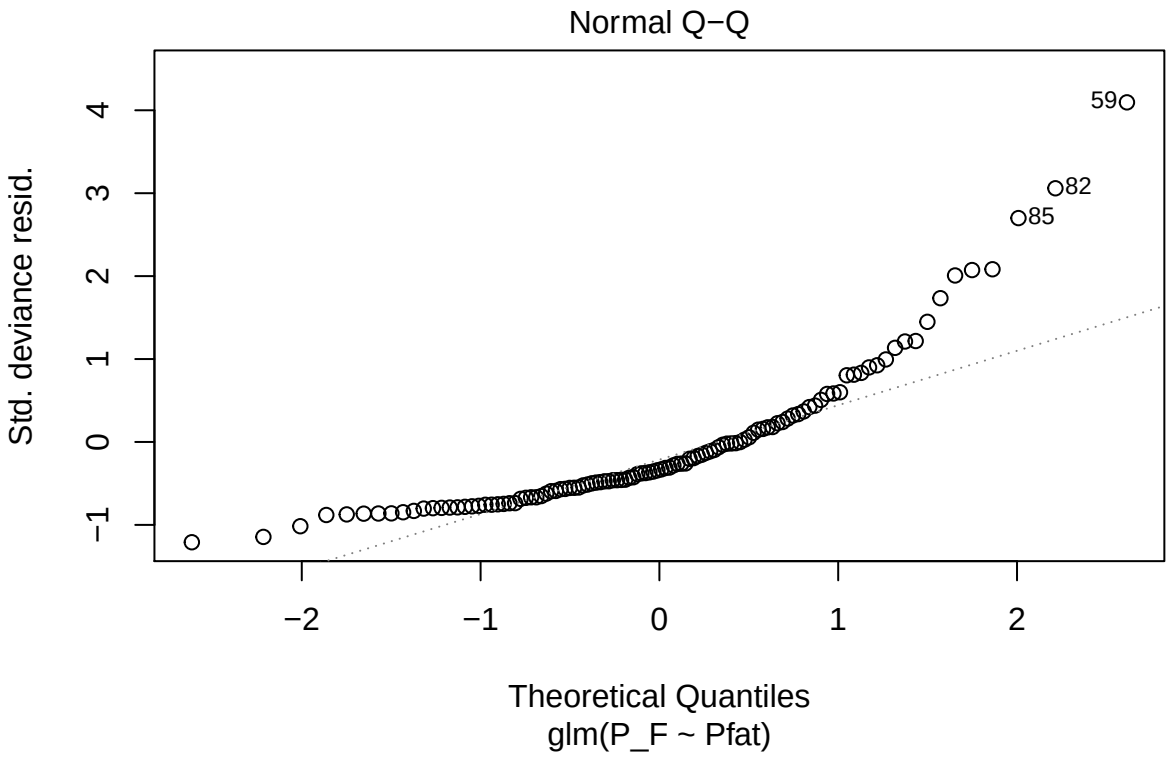


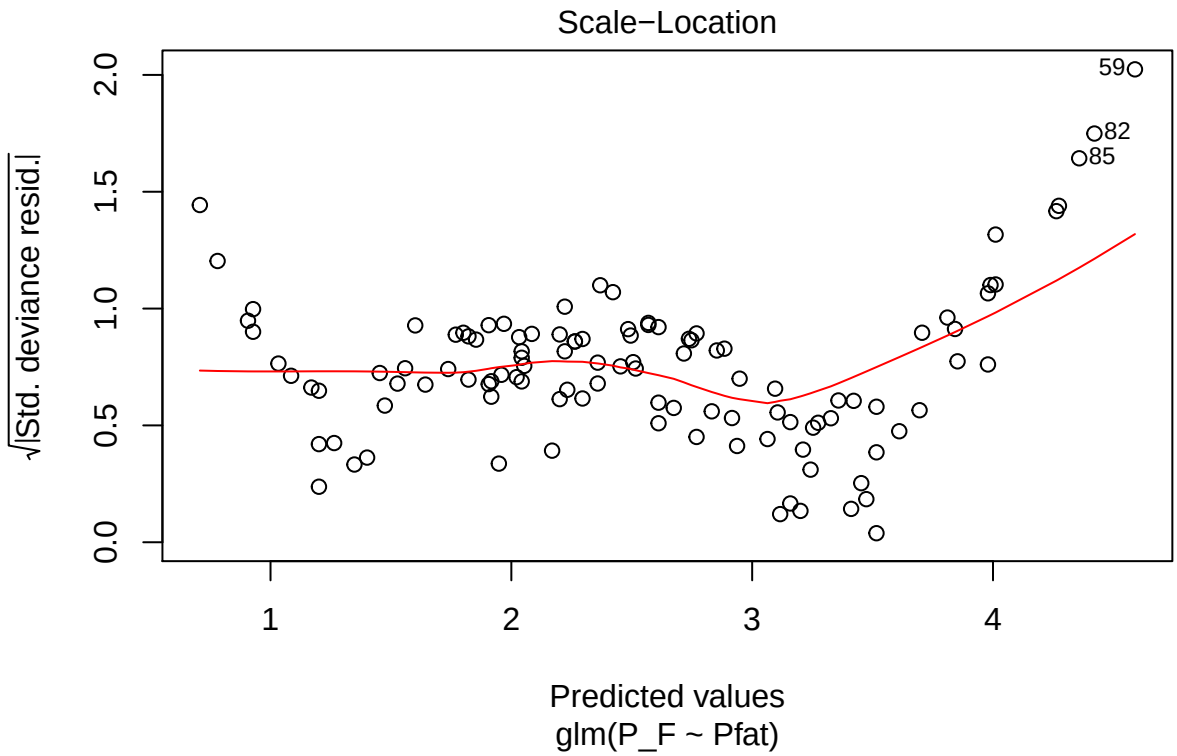

```
summary(fit_poly_m)
```

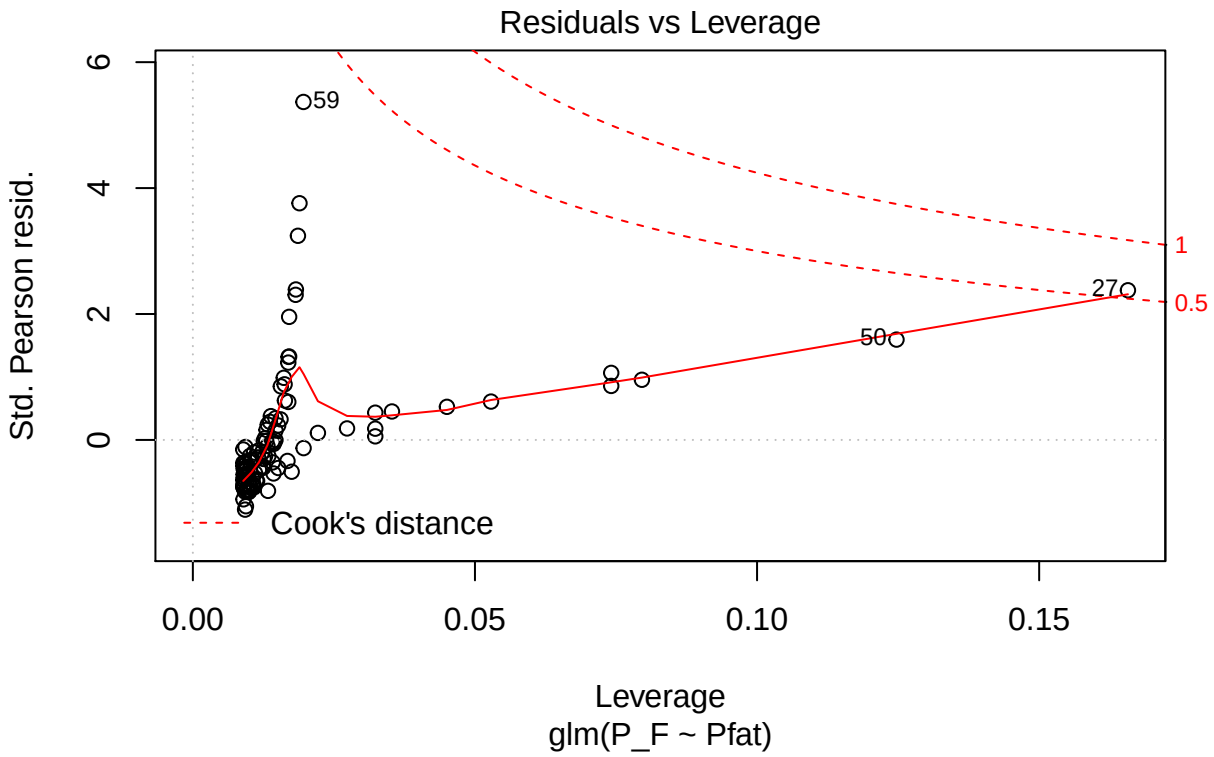
```
##
## Call:
## lm(formula = P_F ~ poly(Pfat, 2), data = data_correction[data_correction$Sex ==
##     "m", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29642 -0.13415  0.00943  0.05964  0.76104
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.28380    0.02365   96.57   <2e-16 ***
## poly(Pfat, 2)1 -7.08246    0.17538  -40.38   <2e-16 ***
## poly(Pfat, 2)2  2.27555    0.17538   12.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1754 on 52 degrees of freedom
## Multiple R-squared:  0.9719, Adjusted R-squared:  0.9708
## F-statistic: 899.6 on 2 and 52 DF,  p-value: < 2.2e-16
```

```
# inverse.gamma
fit_gamma <- glm(P_F ~ Pfat, data = data_correction, family = Gamma(link = "identity"))
plot(fit_gamma)
```

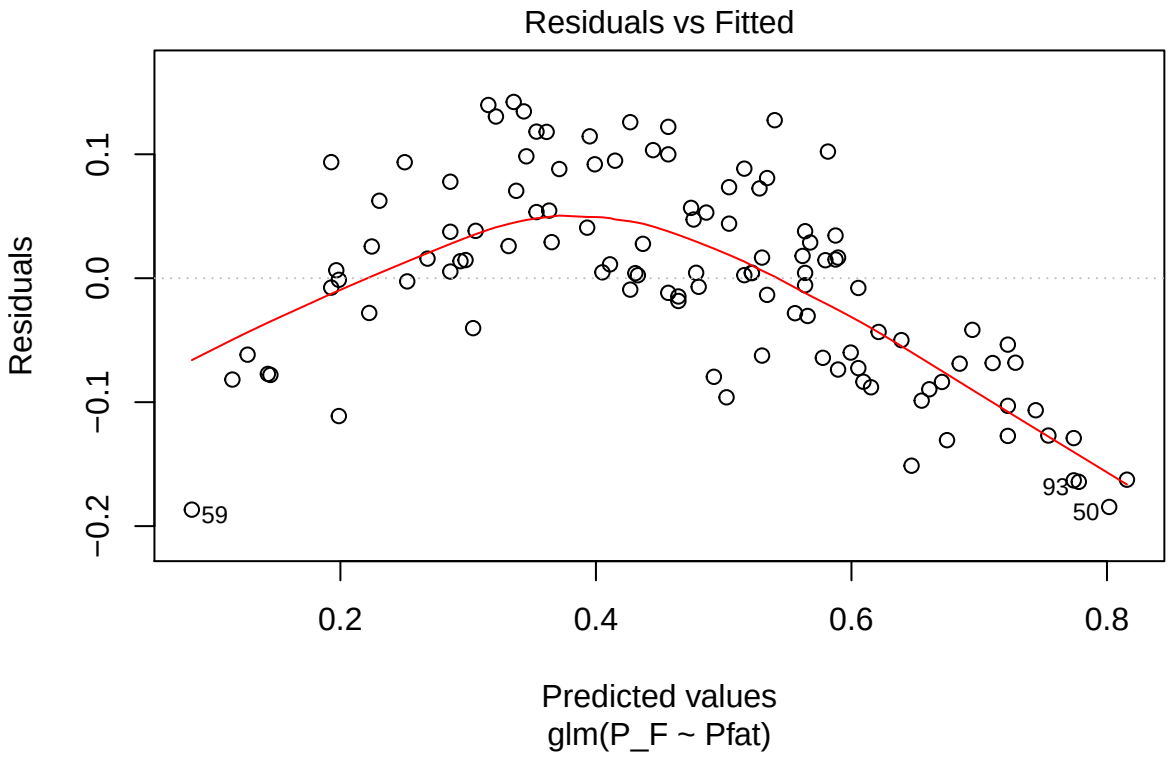

Residuals vs Fitted
Residuals
1.0
0.8
0.6
0.4
0.2
0.0
−0.4
59
82
85
1
2
3
4
Predicted values
glm(P_F ~ Pfat)

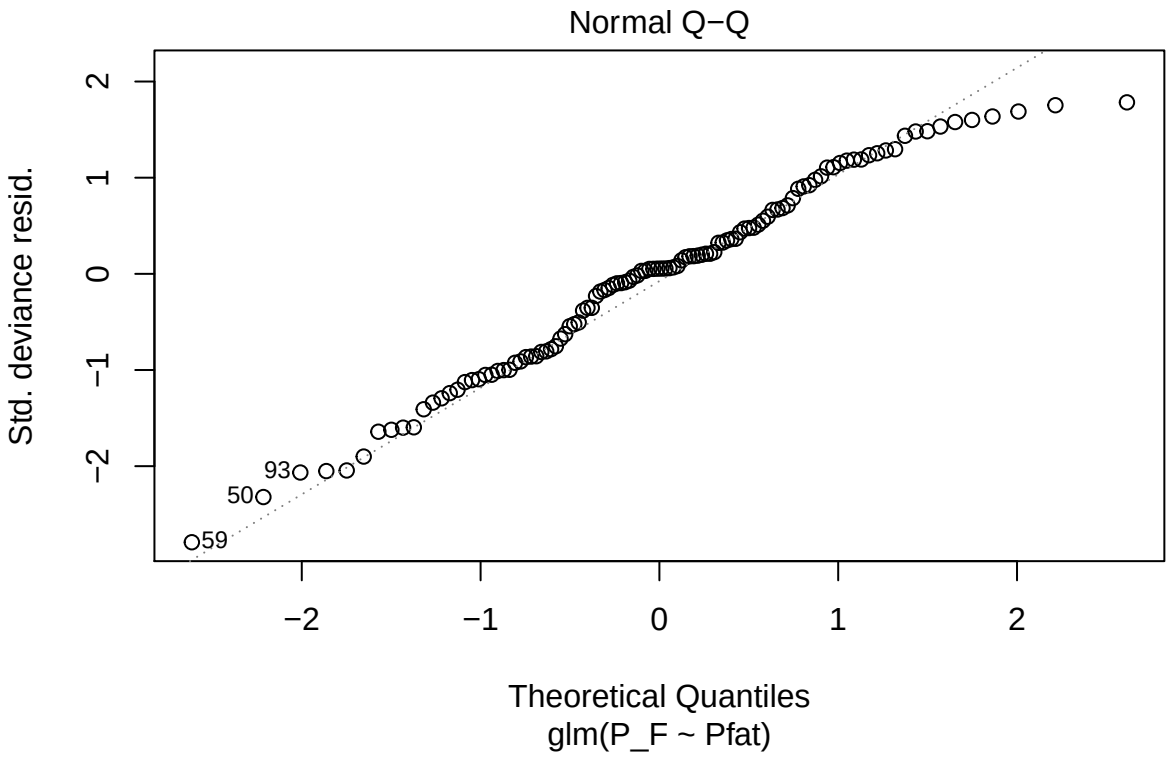

Normal Q−Q
Std. deviance resid.
Theoretical Quantiles
glm(P_F ~ Pfat)
59
82
85
−1
0
1
2
3
4
−2
−1
0
1
2

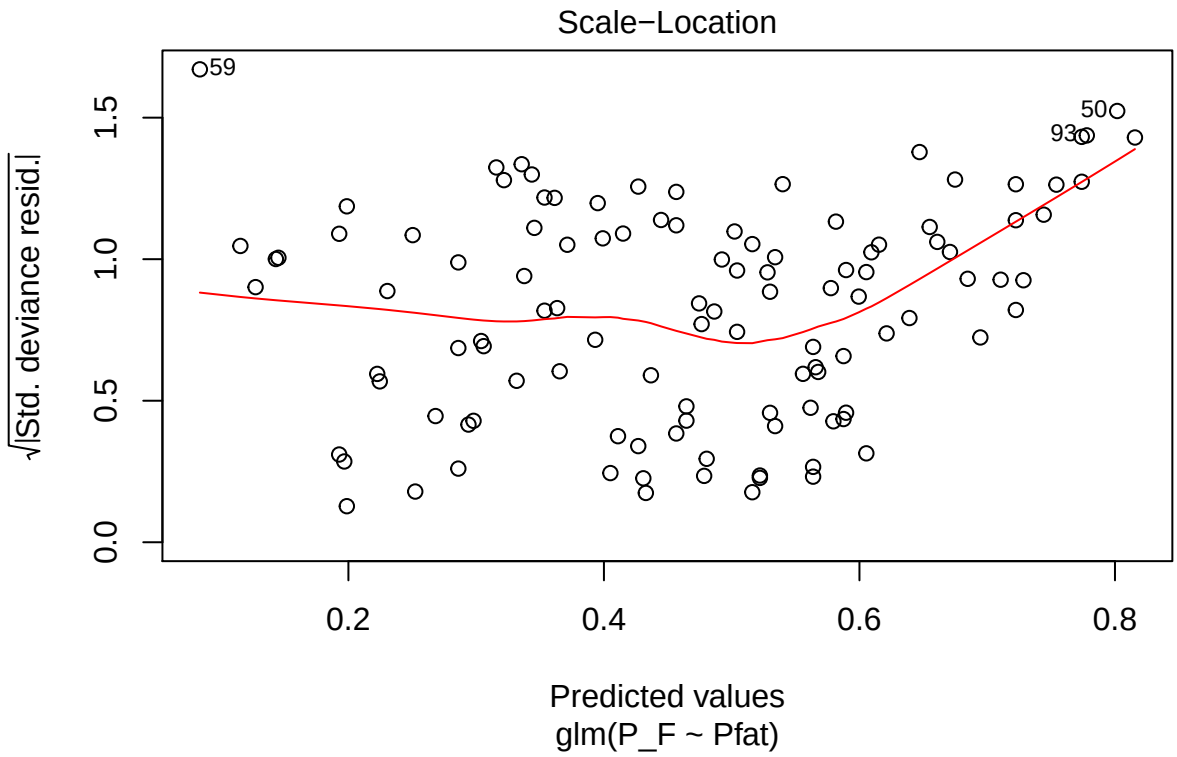

Scale−Location
√|Std. deviance resid.|
0.0
0.5
1.0
1.5
2.0
59
82
85
1
2
3
4
Predicted values
glm(P_F ~ Pfat)

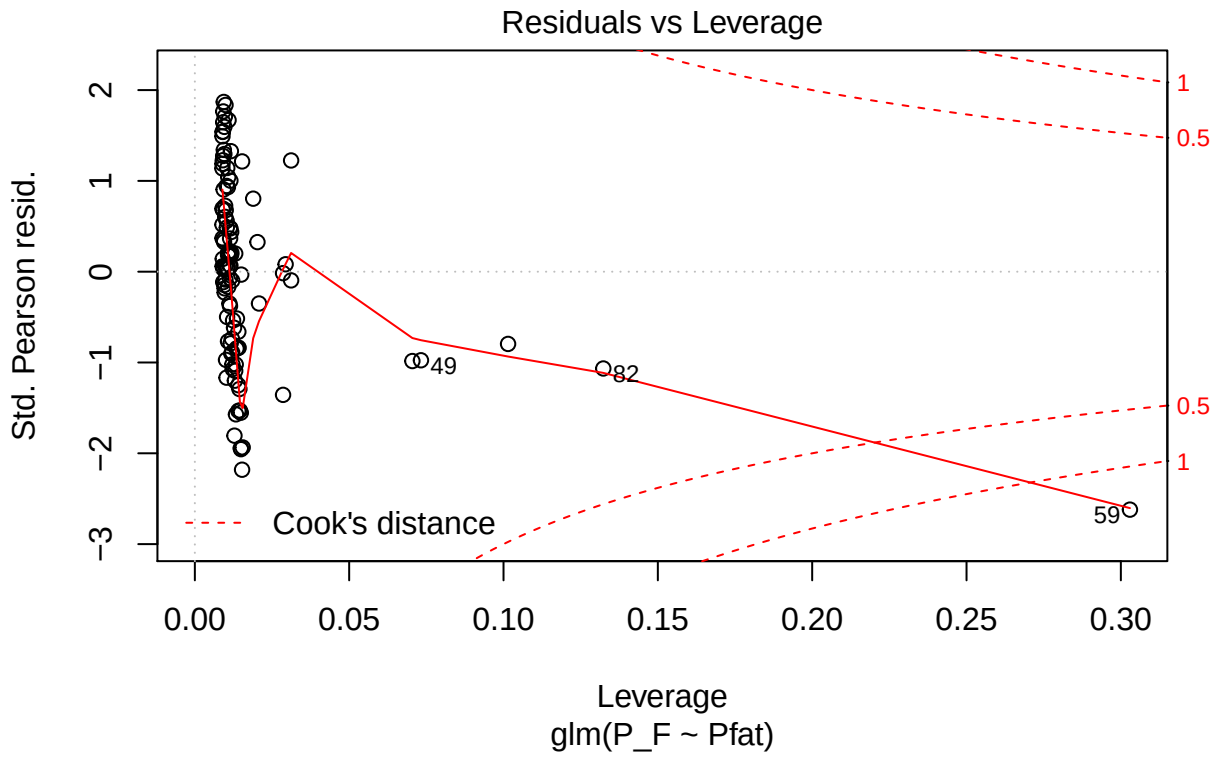



```
fit_inverse <- glm(P_F ~ Pfat, data = data_correction, family = Gamma(link = "inverse"))
plot(fit_inverse)
```


Residuals vs Fitted
Residuals
0.1
0.0
−0.1
−0.2
59
93
50
0.2
0.4
0.6
0.8
Predicted values
glm(P_F ~ Pfat)


Normal Q−Q
Std. deviance resid.
Theoretical Quantiles
glm(P_F ~ Pfat)
93
50
59
−2
−1
0
1
2


Scale−Location
59
50
93
√|Std. deviance resid.|
0.0
0.5
1.0
1.5
0.2
0.4
0.6
0.8
Predicted values
glm(P_F ~ Pfat)




```
# poission
fit_log_poi <- glm(P_F ~ Pfat, data = data_correction, family = poisson(link = log))
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.166902

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.071878

## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.106599

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.468504

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.205584

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.622481

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.099329

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.092593

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.380952

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.534928
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.650089
## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.147410
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.615385
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.486392
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.423622
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.743132
## Warning in dpois(y, mu, log = TRUE): non-integer x = 6.363636
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.953271
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.405914
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.460465
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.296647
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.094118
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.175926
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.901198
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.164345
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.761354
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.037417
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.812785
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.788136
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.568528
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.714396
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.805882
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.748981
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.323387
```

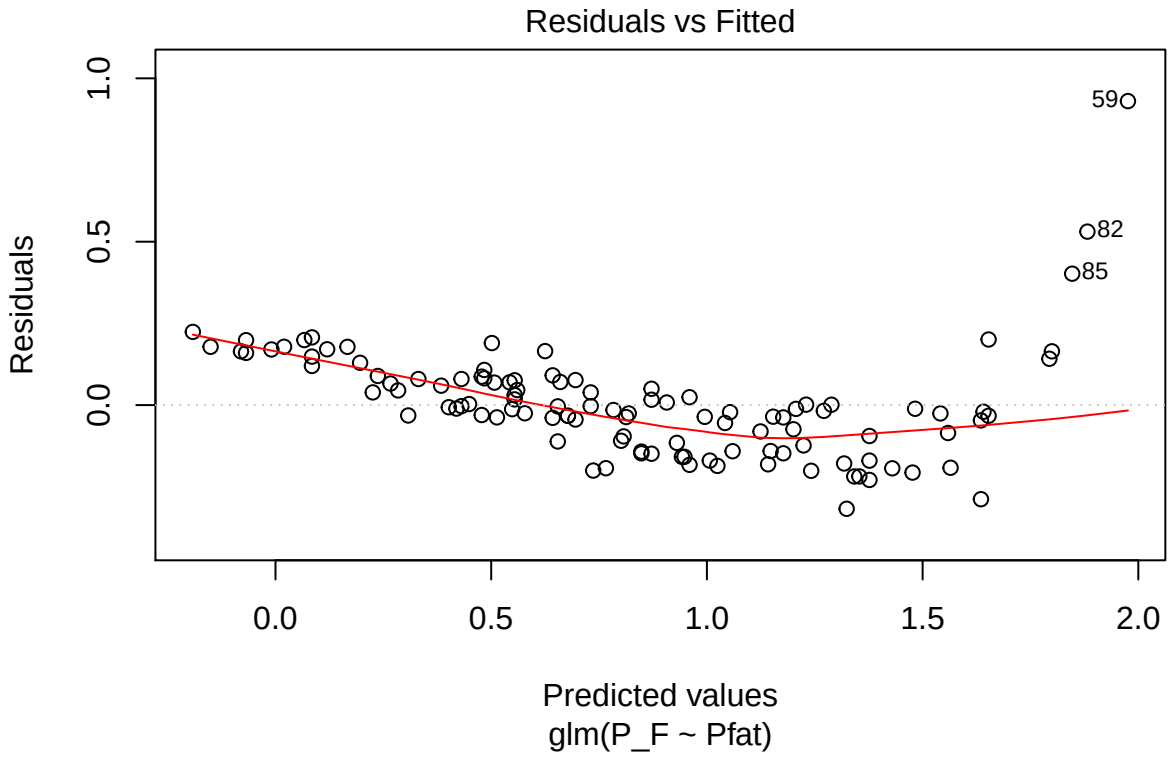
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.904762
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.330561
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.489107
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.531496
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.164742
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.750547
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.113333
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.923345
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.812371
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.923935
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.763911
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.281507
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.121101
## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.024876
## Warning in dpois(y, mu, log = TRUE): non-integer x = 6.457143
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.031043
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.215513
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.246590
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.380392
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.492386
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.067138
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.942359
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.316062
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.416275
```

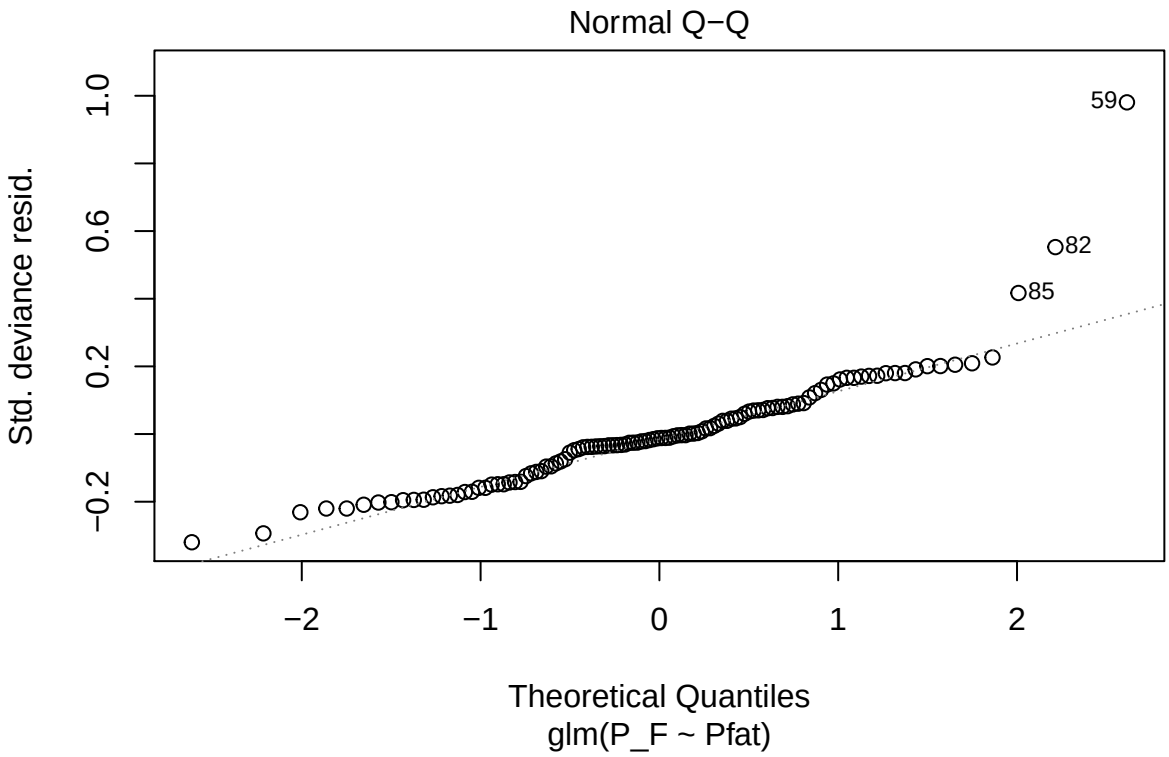
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 9.852349
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.630178
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.368421
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.132480
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.381484
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.132530
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.574422
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.515012
## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.114754
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.177033
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.028195
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.199812
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.831761
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.982759
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.638914
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.321199
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.098404
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.507792
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.369376
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.487854
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.034381
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.628842
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.937337
## Warning in dpois(y, mu, log = TRUE): non-integer x = 7.972973
```

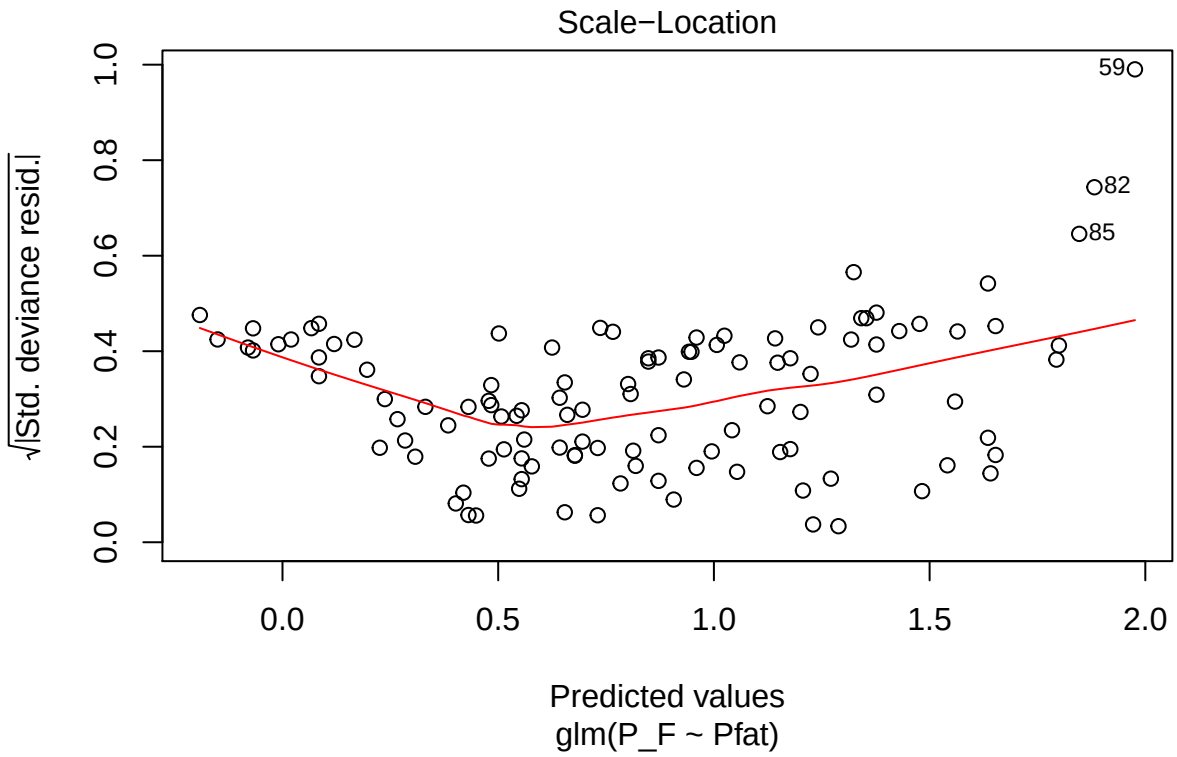
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.645707
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.727852
## Warning in dpois(y, mu, log = TRUE): non-integer x = 7.378049
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.396040
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.918367
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.775401
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.648810
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.188119
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.380468
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.085666
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.771451
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.570184
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.229525
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.480649
## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.687500
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.540894
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.873926
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.781513
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.842054
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.362084
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.724771
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.113905
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.448819
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.320359
```

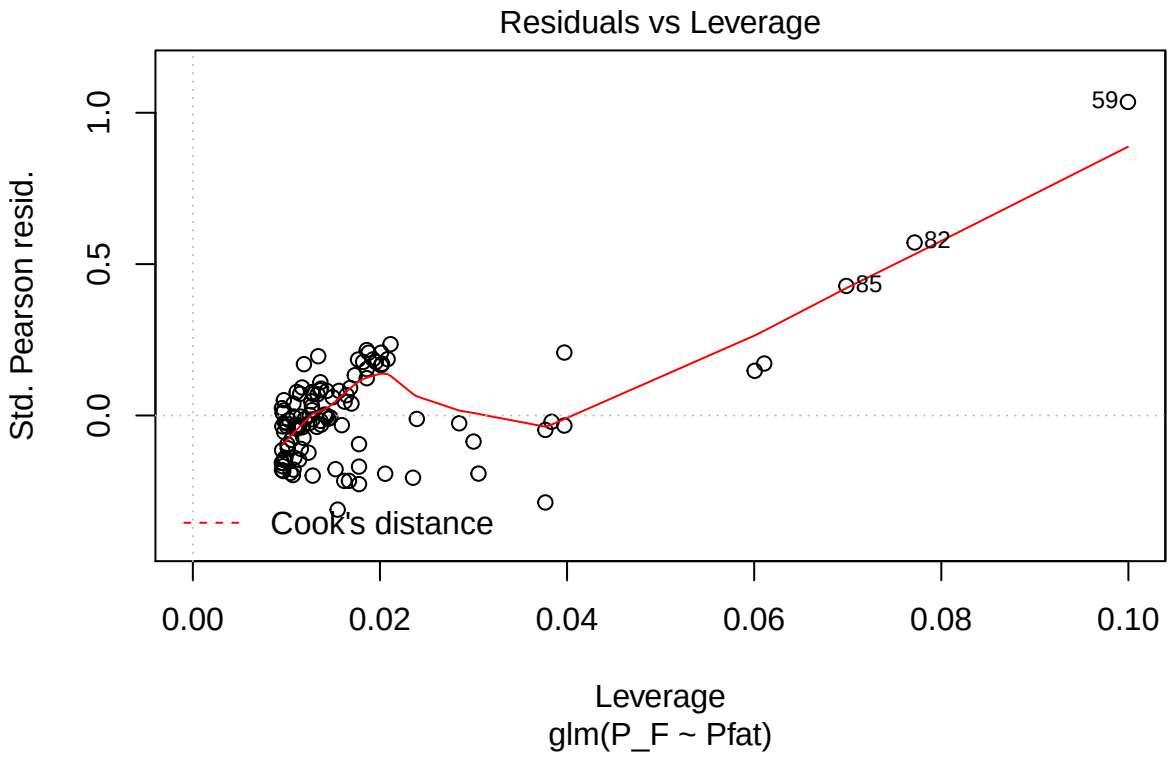
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.313302

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.311430

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.847737

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.353591

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.817323

## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.160000
```

```
plot(fit_log_poi)
```




Normal Q−Q
Std. deviance resid.
1.0
0.6
0.2
−0.2
59
82
85
−2
−1
0
1
2
Theoretical Quantiles
glm(P_F ~ Pfat)


Scale−Location
√|Std. deviance resid.|
1.0
0.8
0.6
0.4
0.2
0.0
59
82
85
0.0
0.5
1.0
1.5
2.0
Predicted values
glm(P_F ~ Pfat)




```
fit_log_poi_f <- glm(P_F ~ Pfat, data = data_correction[data_correction$Sex == "f",], family = poisson(
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.205584
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.622481
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.099329
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.650089
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.147410
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.615385
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.423622
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.743132
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 6.363636
```

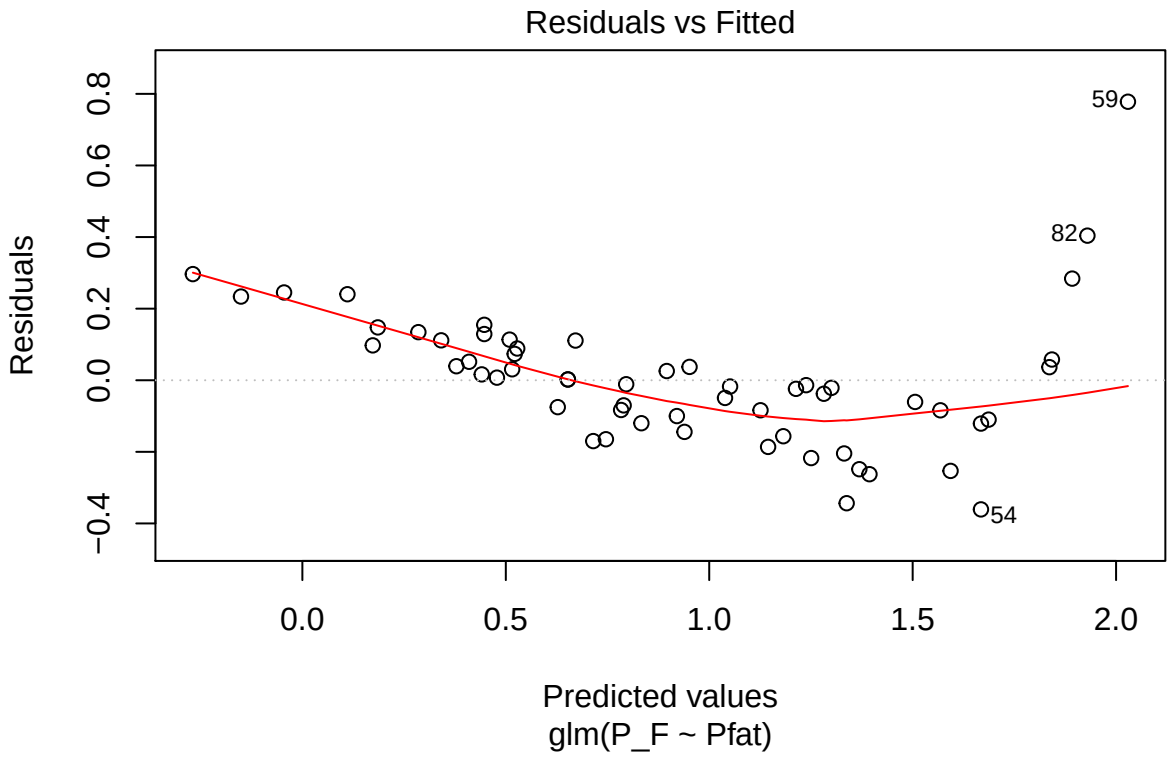
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.296647
```

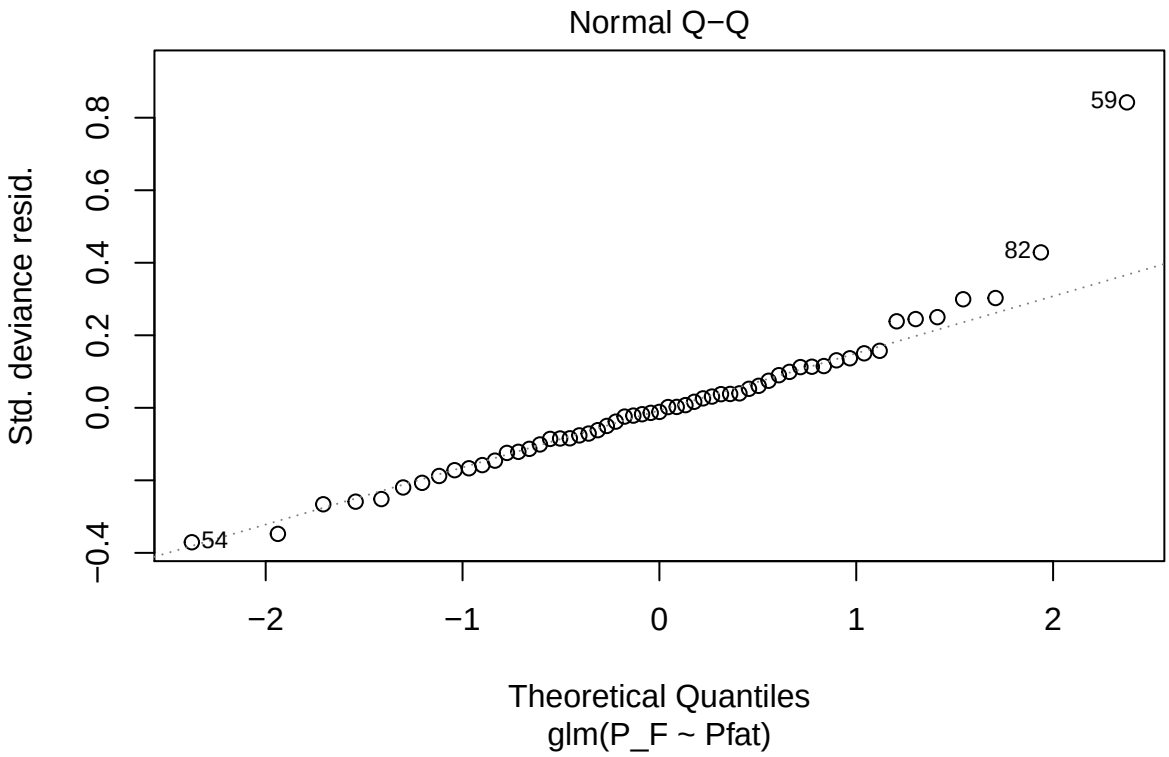
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.094118
```

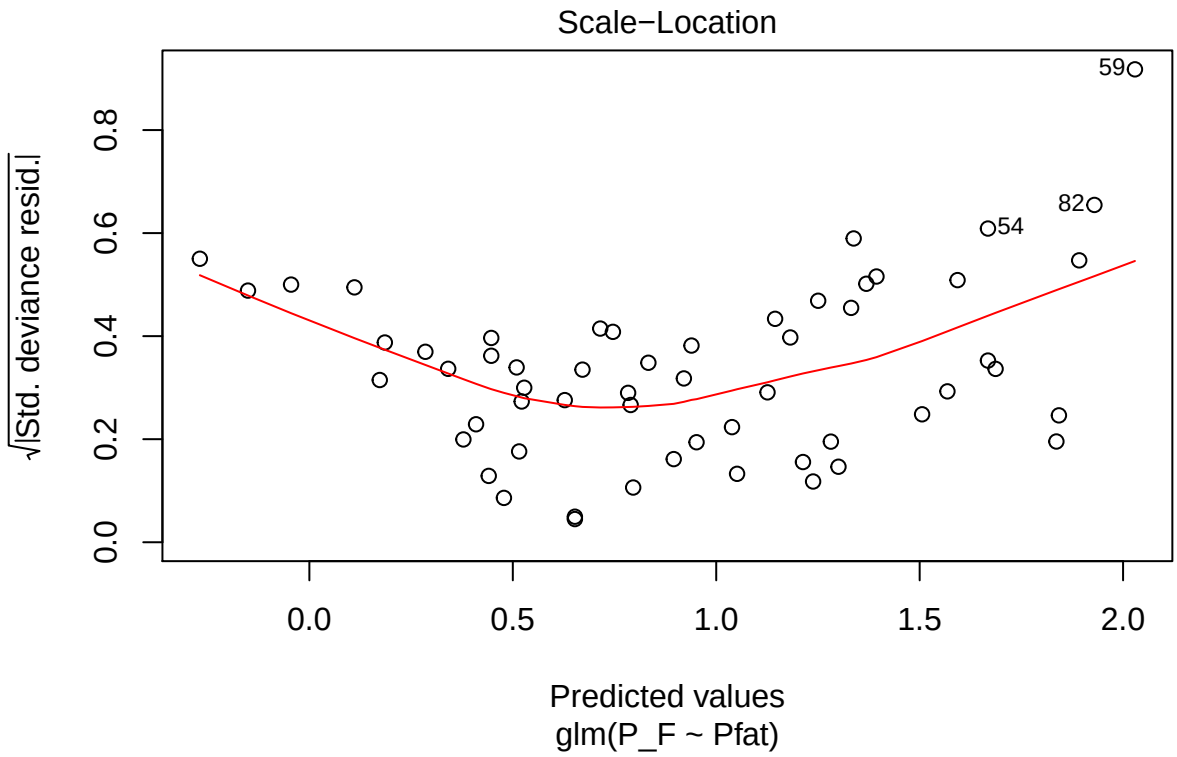
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.761354
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.037417
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.812785
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.714396
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.805882
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.330561
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.489107
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.531496
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.923345
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.812371
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.923935
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.121101
## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.024876
## Warning in dpois(y, mu, log = TRUE): non-integer x = 6.457143
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.380392
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.492386
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.067138
## Warning in dpois(y, mu, log = TRUE): non-integer x = 9.852349
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.630178
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.368421
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.574422
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.515012
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.199812
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.831761
```

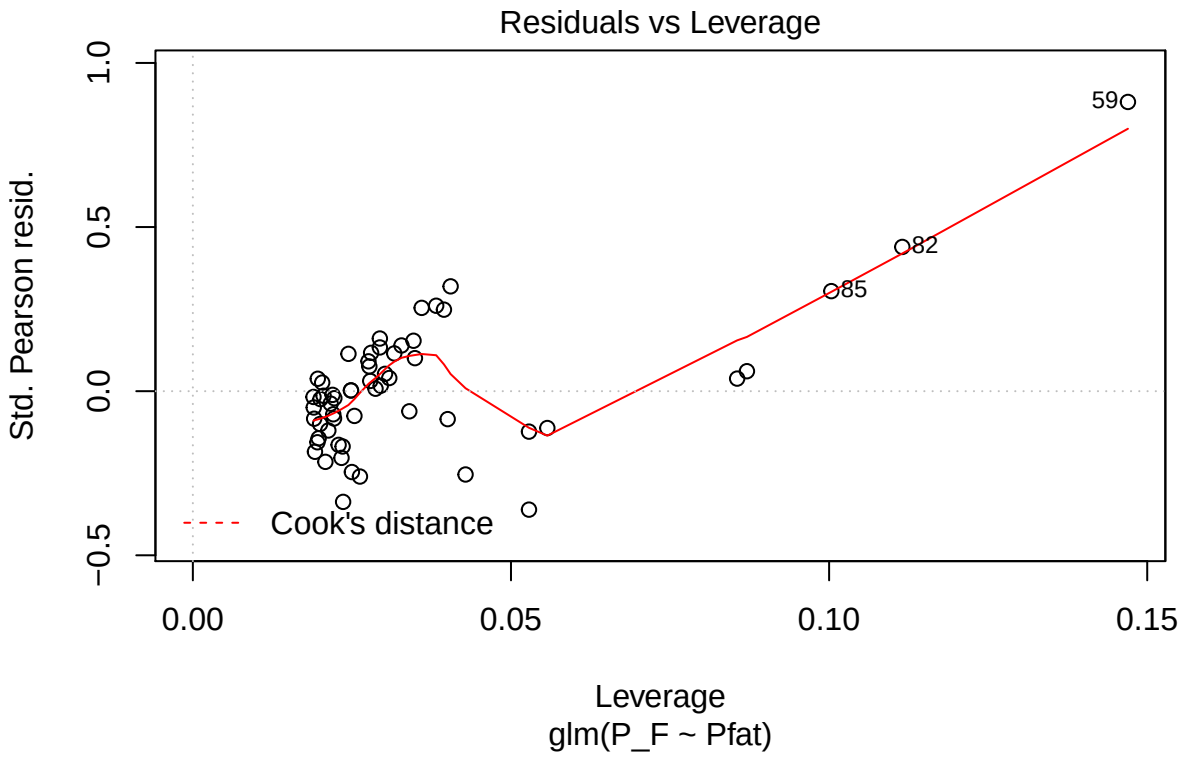
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.982759
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.507792
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.369376
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.487854
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.937337
## Warning in dpois(y, mu, log = TRUE): non-integer x = 7.972973
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.727852
## Warning in dpois(y, mu, log = TRUE): non-integer x = 7.378049
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.396040
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.380468
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.085666
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.771451
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.570184
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.540894
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.873926
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.781513
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.113905
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.448819
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.320359
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.353591
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.817323
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.160000
```

```
plot(fit_log_poi_f)
```




Normal Q−Q
Std. deviance resid.
−0.4
0.0
0.2
0.4
0.6
0.8
59
82
54
−2
−1
0
1
2
Theoretical Quantiles
glm(P_F ~ Pfat)


Scale−Location
59
82
54
√|Std. deviance resid.|
0.0
0.2
0.4
0.6
0.8
0.0
0.5
1.0
1.5
2.0
Predicted values
glm(P_F ~ Pfat)




```
fit_log_poi_m <- glm(P_F ~ Pfat, data = data_correction[data_correction$Sex == "m",], family = poisson(
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.166902
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.071878
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.106599
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.468504
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.092593
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.380952
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.534928
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.486392
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.953271
```

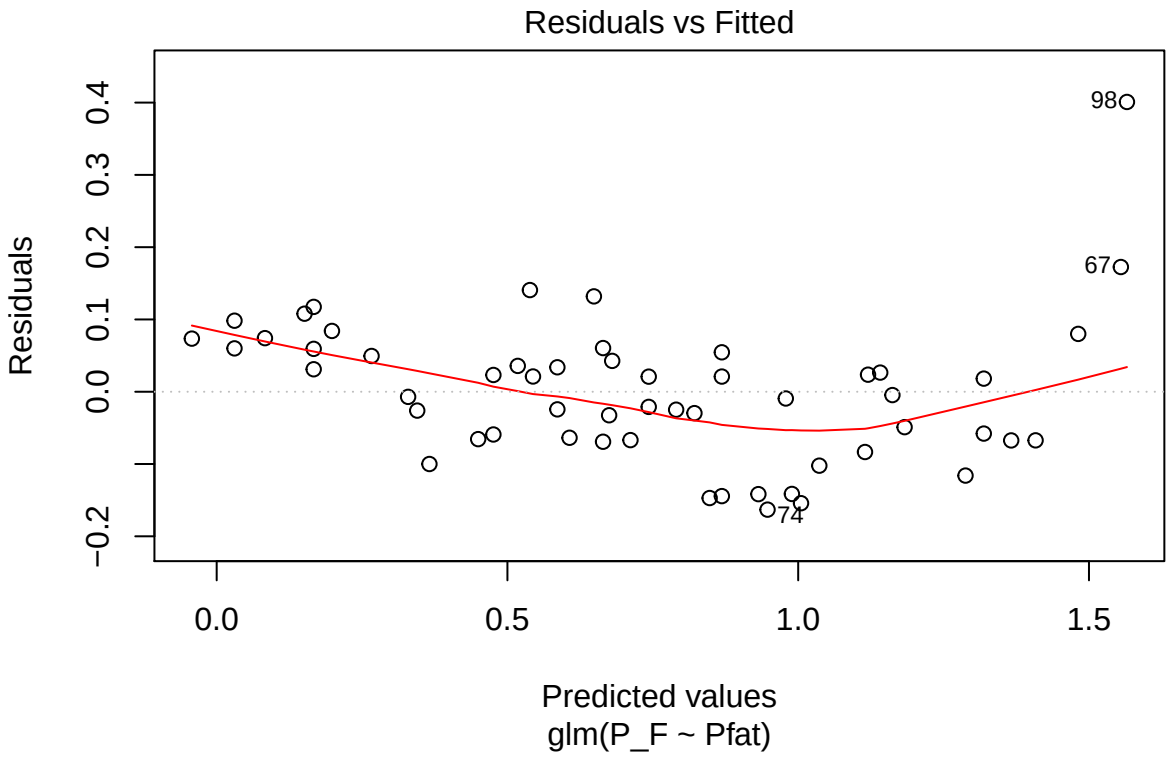
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.405914
```

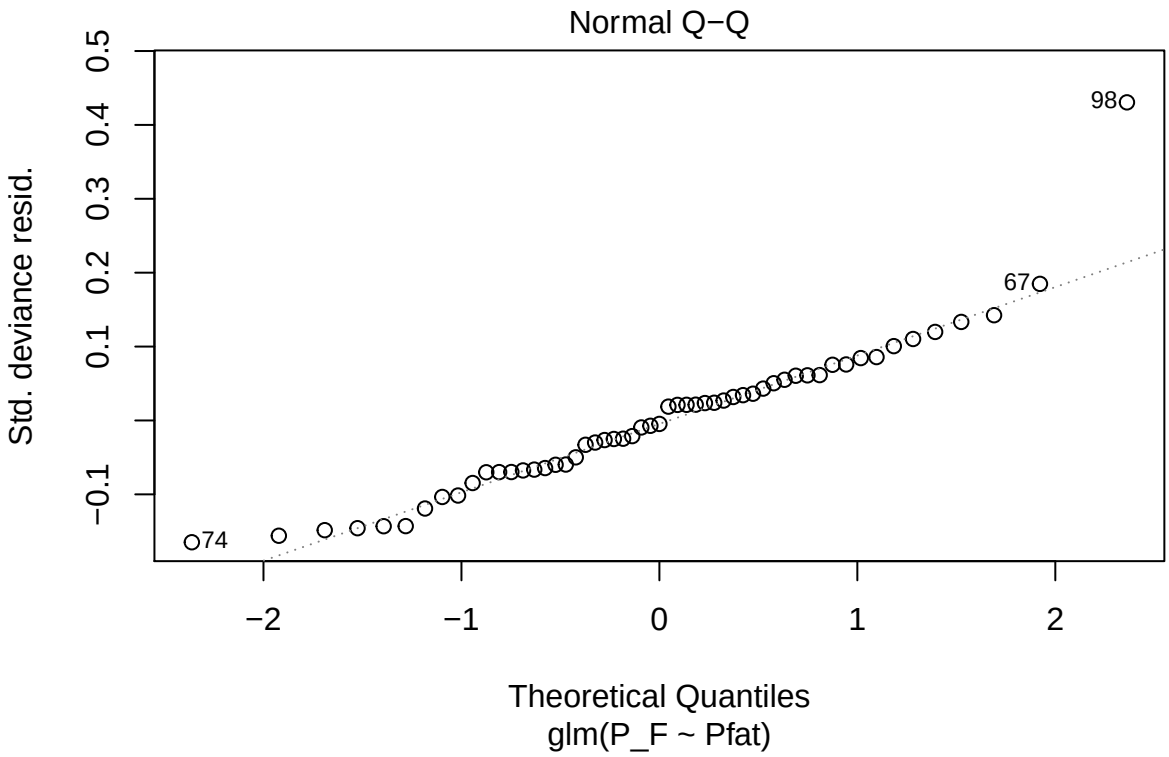
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.460465
```

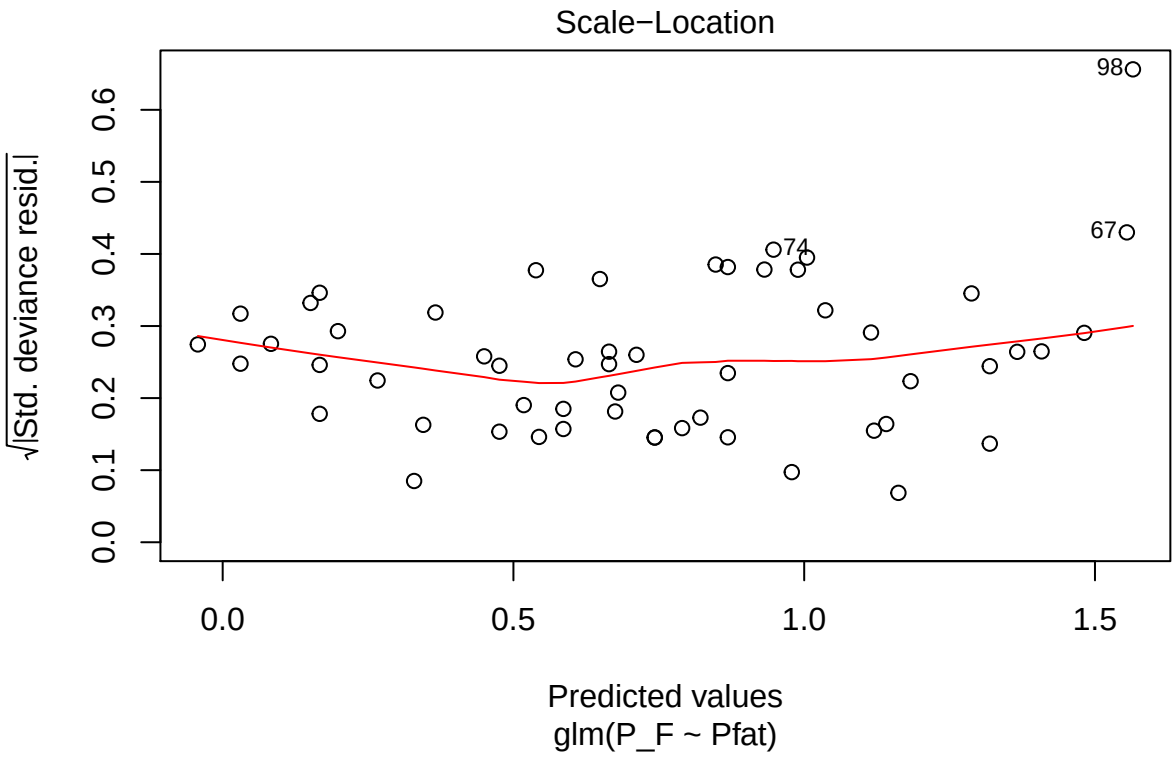
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.175926
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.901198
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.164345
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.788136
## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.568528
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.748981
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.323387
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.904762
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.164742
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.750547
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.113333
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.763911
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.281507
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.031043
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.215513
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.246590
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.942359
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.316062
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.416275
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.132480
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.381484
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.132530
## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.114754
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.177033
```

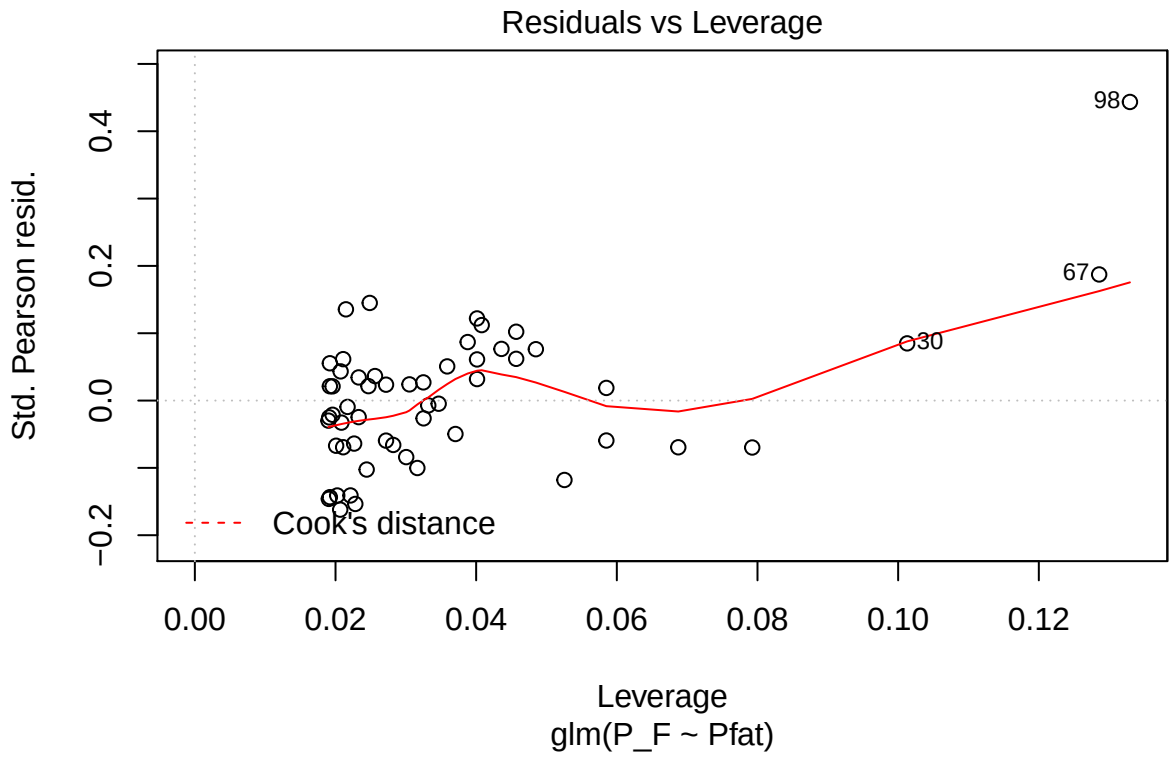
```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.028195

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.638914

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.321199

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.098404

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.034381

## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.628842

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.645707

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.918367

## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.775401

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.648810

## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.188119

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.229525

## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.480649

## Warning in dpois(y, mu, log = TRUE): non-integer x = 5.687500

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.842054

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.362084

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.724771

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.313302

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.311430

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.847737
```

```
plot(fit_log_poi_m)
```


Residuals vs Fitted
Residuals
Predicted values
glm(P_F ~ Pfat)
98
67
74
−0.2
0.0
0.1
0.2
0.3
0.4
0.0
0.5
1.0
1.5


Normal Q−Q
Std. deviance resid.
0.5
0.4
0.3
0.2
0.1
−0.1
98
67
74
−2
−1
0
1
2
Theoretical Quantiles
glm(P_F ~ Pfat)


Scale−Location
98
67
74
√|Std. deviance resid.|
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.0
0.5
1.0
1.5
Predicted values
glm(P_F ~ Pfat)




```
AIC(fit_exp) #[1] 77.66344  this seem to be a good model
```

```
## [1] 77.66344
```

```
AIC(fit_exp_f) # 91.33888
```

```
## [1] 59.17236
```

```
AIC(fit_exp_m) # -8.987106
```

```
## [1] -8.987106
```

```
AIC(fit_log_poi) #[1] Inf
```

```
## [1] Inf
```

```
AIC(fit_inverse) #[1] -54.14454
```

```
## [1] -54.14454
```

```r
AIC(fit_poly) # [1] 97.01369
```

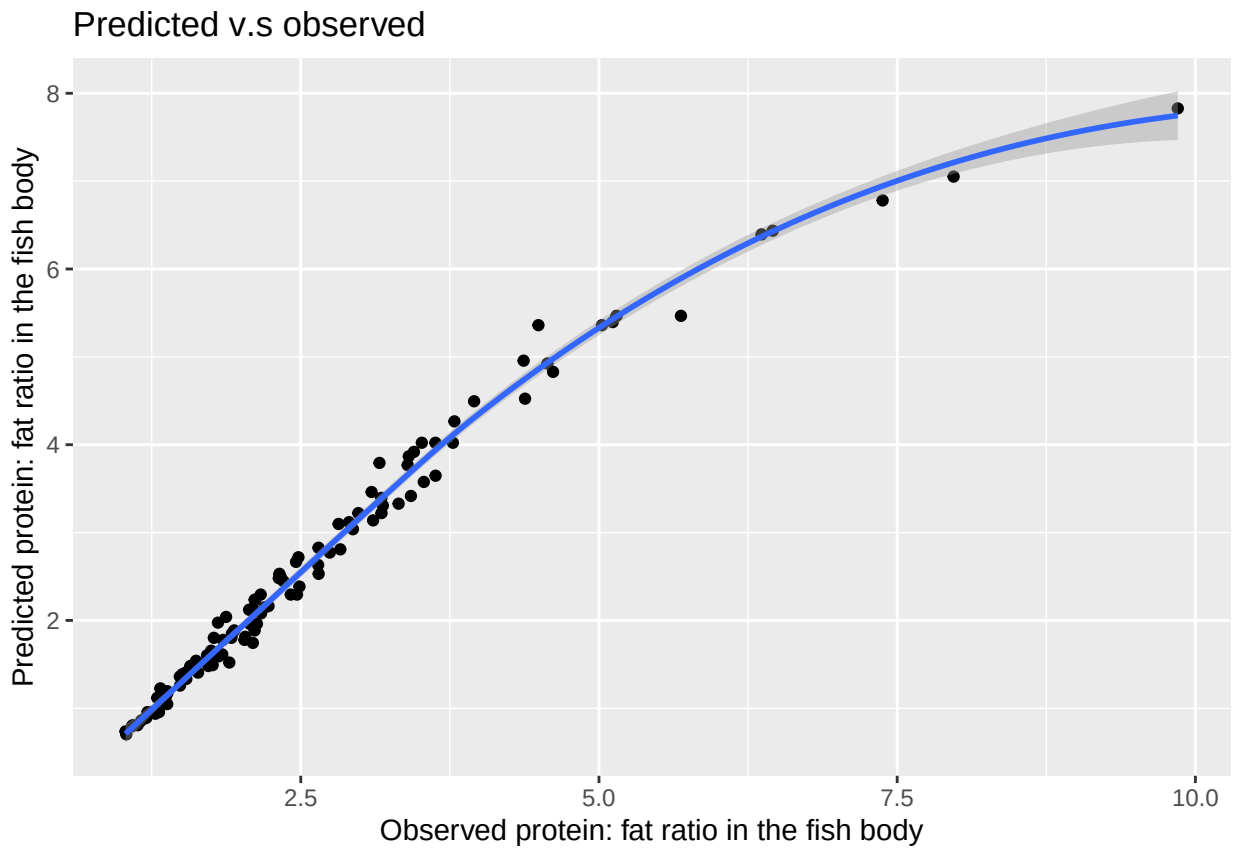
```
## [1] 97.01369
```

```r
AIC(fit_gamma) #[1] 140.0517
```

```
## [1] 140.0517
```

```r
# fit_exp is the model that we are going to use
fit_exp <- glm(P_F ~ Pfat, data = data_correction, family = gaussian(link=log))
data_correction$predict_fitexp <- predict(fit_exp, type = "response")
exp_plot <-  ggplot(data = data_correction,aes(x=P_F, y=predict_fitexp)) + geom_point() +
  geom_smooth() +
  xlab("Observed protein: fat ratio in the fish body") +
  ylab("Predicted protein: fat ratio in the fish body") +
  ggtitle("Predicted v.s observed")
exp_plot
```

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

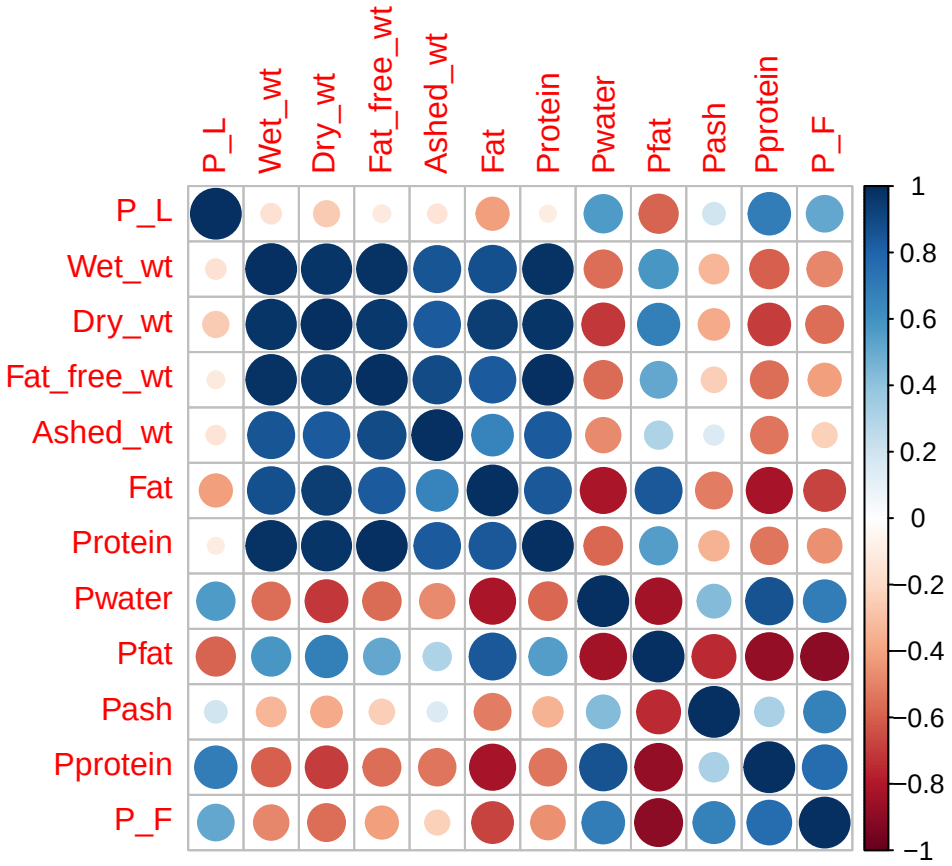


it is possible to predict the P_F ratio of a fish based on their diet

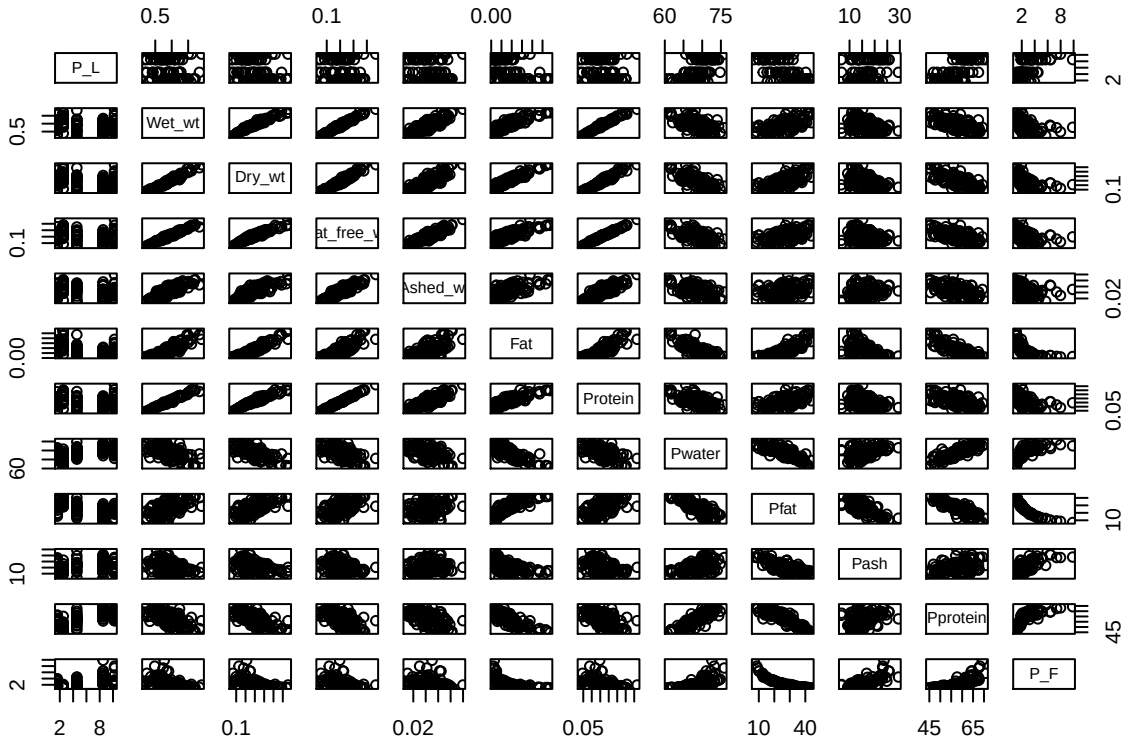
```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corr <- cor(data_correction[,3:14])
corrplot(corr, method="circle")
```




```
pairs(data_correction[,3:14])
```




```
## that would be a logisrtic regression later
```

simulation test

Simulation

Please simulate a set of 1000 new observations based on your best model. How do these simulated observations compare to the real observations?

```
# objective: new y based on the model fit, we need intercept and 5 parameters and also error
# need a dataframe of x, and also random error ()

summary(data_correction) # mean, median and variance, min and max, mean and sd can give a normal distri
```

```
##       Tank    Sex          P_L              Wet_wt          Dry_wt
##  M1      : 3   f:57   Min.   : 1.625   Min.   :0.1711   Min.   :0.0499
##  M12     : 3   m:55   1st Qu.: 2.517   1st Qu.:0.7007   1st Qu.:0.2002
##  M13     : 3          Median : 4.562   Median :0.9034   Median :0.2874
##  M14     : 3          Mean   : 5.562   Mean   :0.9347   Mean   :0.2998
##  M4      : 3          3rd Qu.: 8.513   3rd Qu.:1.2154   3rd Qu.:0.3932
##  M5      : 3          Max.   :10.227   Max.   :1.9092   Max.   :0.6636
##  (Other):94
##   Fat_free_wt        Ashed_wt            Fat              Protein
##  Min.   :0.0394   Min.   :0.00790   Min.   :0.00800   Min.   :0.0305
```

```
##  1st Qu.:0.1584   1st Qu.:0.03335   1st Qu.:0.03665   1st Qu.:0.1252
##  Median :0.2087   Median :0.04625   Median :0.07555   Median :0.1585
##  Mean   :0.2121   Mean   :0.04516   Mean   :0.08766   Mean   :0.1670
##  3rd Qu.:0.2748   3rd Qu.:0.05670   3rd Qu.:0.11663   3rd Qu.:0.2192
##  Max.   :0.4670   Max.   :0.09980   Max.   :0.28670   Max.   :0.3673
##
##      Pwater           Pfat            Pash          Pprotein
##  Min.   :60.60   Min.   : 7.00   Min.   : 6.71   Min.   :44.55
##  1st Qu.:66.47   1st Qu.:20.02   1st Qu.:12.36   1st Qu.:52.31
##  Median :69.00   Median :26.95   Median :15.12   Median :57.46
##  Mean   :68.63   Mean   :26.48   Mean   :15.84   Mean   :57.68
##  3rd Qu.:70.90   3rd Qu.:32.42   3rd Qu.:18.45   3rd Qu.:62.62
##  Max.   :75.90   Max.   :43.90   Max.   :29.45   Max.   :70.72
##
##       P_F             diff               new_pf       predict_fitexp
##  Min.   :1.031   Min.   :-1.459908   Min.   :0.1065   Min.   :0.7039
##  1st Qu.:1.696   1st Qu.:-0.981446   1st Qu.:2.2276   1st Qu.:1.4888
##  Median :2.118   Median :-0.175830   Median :2.6591   Median :2.1284
##  Mean   :2.621   Mean   :-0.004366   Mean   :2.6250   Mean   :2.5818
##  3rd Qu.:3.176   3rd Qu.: 0.690621   3rd Qu.:2.9818   3rd Qu.:3.3450
##  Max.   :9.852   Max.   : 3.326026   Max.   :6.5263   Max.   :7.8278
##
```

```r
Pfat <- seq(min(data_correction$Pfat), max(data_correction$Pfat),length.out = 1000)
grid <- expand.grid(Pfat)
names(grid) <- c("Pfat")

summary(fit_exp)
```

```
##
## Call:
## glm(formula = P_F ~ Pfat, family = gaussian(link = log), data = data_correction)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.86813  -0.16542   0.07801   0.22264   2.02460
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.514625   0.026822   93.75   <2e-16 ***
## Pfat        -0.065279   0.001422  -45.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1130409)
##
##     Null deviance: 254.519  on 111  degrees of freedom
## Residual deviance:  12.434  on 110  degrees of freedom
## AIC: 77.663
##
## Number of Fisher Scoring iterations: 4
```

```r
# (Dispersion parameter for gaussian family taken to be 0.1130409)
# it is the same as mean square error

set.seed(777)
grid$error <- rnorm(n=1000, mean=0, sd=0.1130409)
head(grid)
```

```
##       Pfat       error
## 1 7.000000  0.05536587
## 2 7.036937 -0.04505148
## 3 7.073874  0.05774540
## 4 7.110811 -0.04508207
## 5 7.147748  0.18523855
## 6 7.184685  0.07022937
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
```

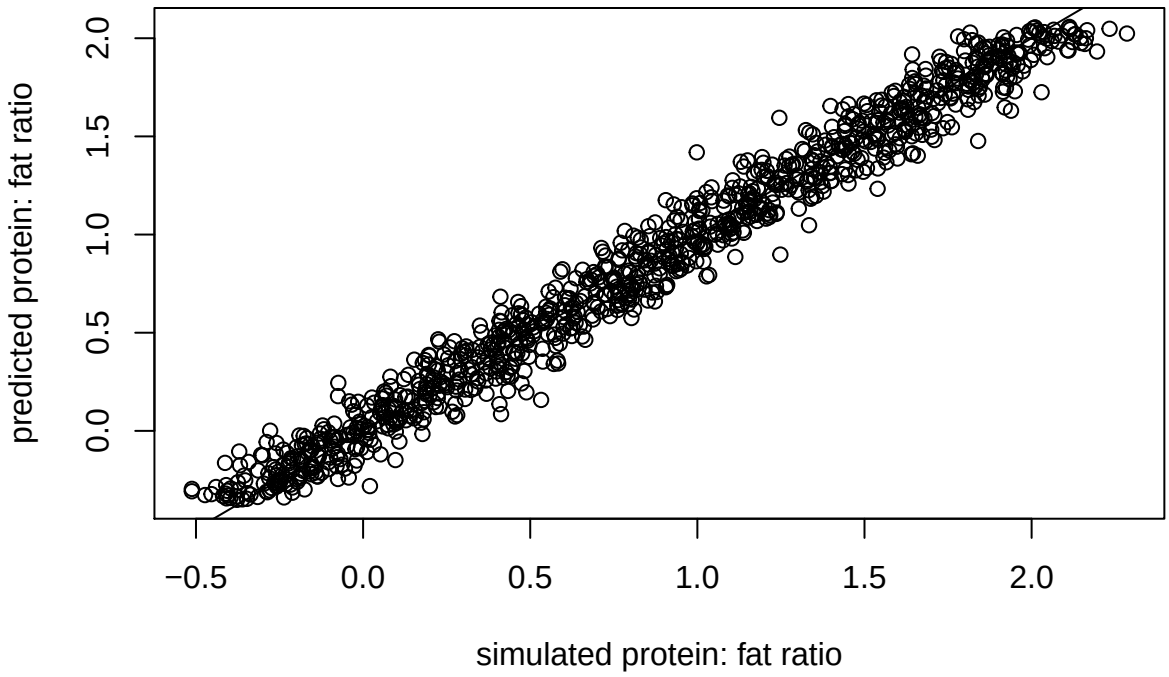
```
## v tibble  2.1.3     v purrr   0.3.2
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x gridExtra::combine() masks dplyr::combine()
## x dplyr::filter()      masks stats::filter()
## x dplyr::lag()         masks stats::lag()
## x purrr::lift()        masks caret::lift()
```

```r
grid_sim <- grid %>%
  mutate(y1 = 2.514625 -0.065279*Pfat + error) %>%
  mutate(y2 = exp(y1))

grid_sim$pred <- predict(fit_exp, newdata = grid_sim) # based on the new dataframe of the dataframe, ca

plot(x=grid_sim$y1, y=grid_sim$pred, xlab = "simulated protein: fat ratio", ylab = "predicted protein: f
cor.test(x = grid_sim$y1, y=grid_sim$pred, method = "pearson")
```

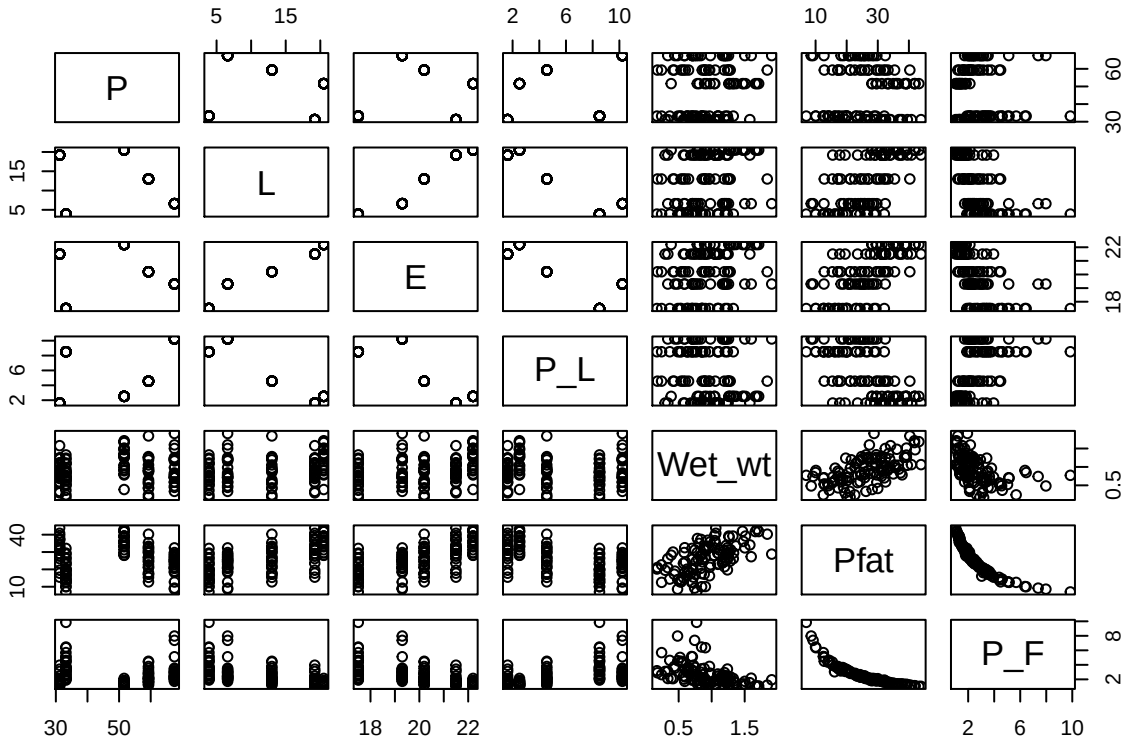
```
##
##  Pearson's product-moment correlation
##
## data:  grid_sim$y1 and grid_sim$pred
## t = 196.59, df = 998, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9856714 0.9888041
## sample estimates:
##       cor
## 0.9873336
```

```
abline(a=0,b=1)
```

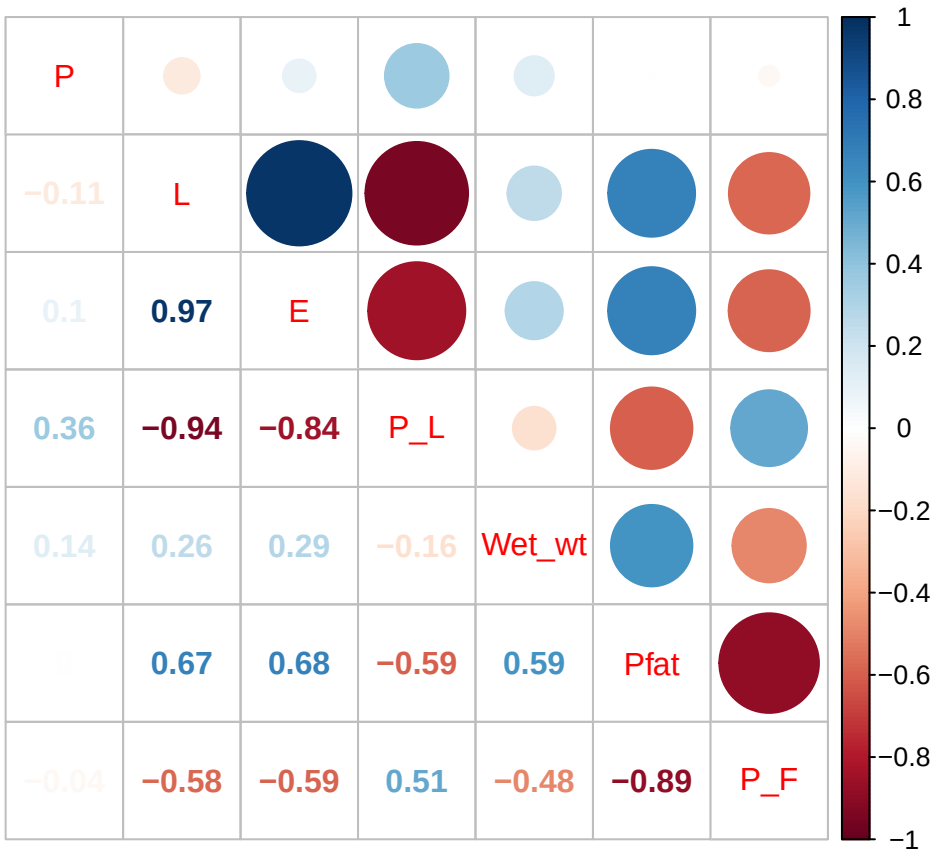



```
# this looks great
```

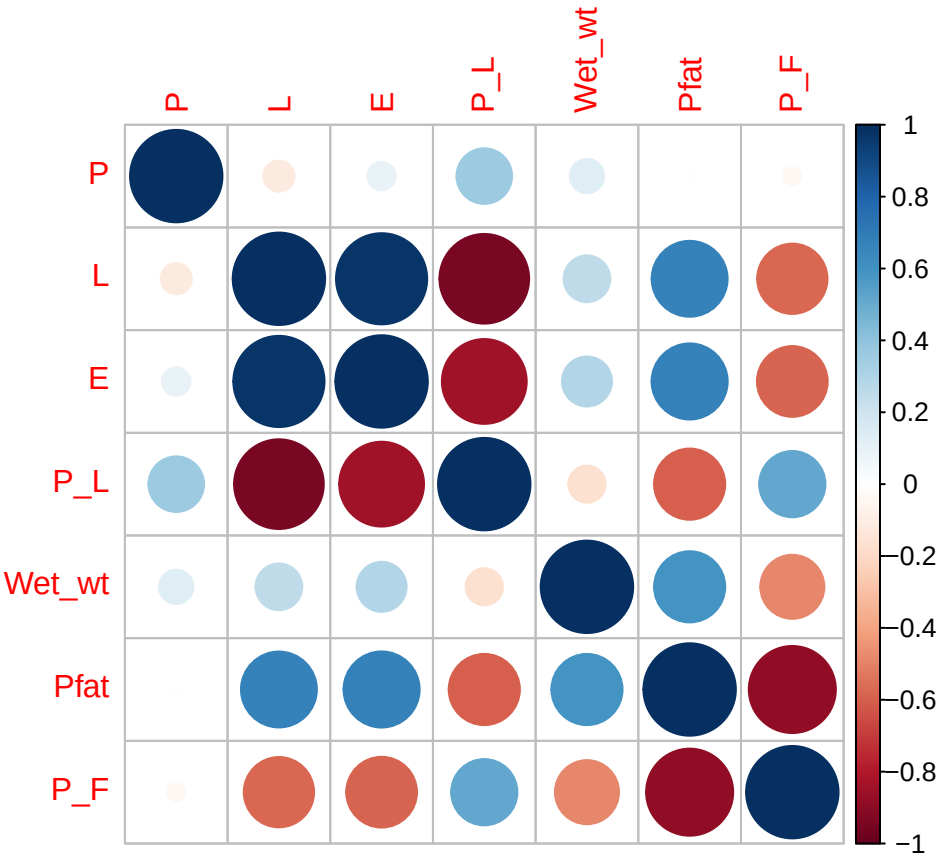
```
# see if we are able to find PL ratio based on the diet the fish take
data6 <- data3[,c(4:12,25)]
data6 <- data3[,c(6:9,12,19,25)]
pairs(data6)
```




```
# based on the paired plot, it seems that overall, lipid is a very good indicator, and the inclusion of
library(corrplot)
corr <- cor(data6)
corrplot.mixed(corr)
```




```
corrplot(corr, method="circle")
```


P
L
E
P_L
Wet_wt
Pfat
P_F
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1