Predict Winners Earned in Professional women's Tennis Matches

Mitchell Rapaport

(Section: Tuesday 5 - 5:50 pm)

Dante Coletta

(Section: Tuesday 5 - 5:50 pm)

Introduction

This project focuses on the number of winners earned in a match by female players competing in the 2013 U.S. Open in a single match, given the attributes in the data set

“Tennis Major Tournament Match Statistics” from the UC Irvine Machine Learning Repository. We will investigate if the number of winners earned can be predicted by these predictors: break points won, aces won, first-serve percentage, first serves won, unforced errors and net points won. Particularly which single predictor is the “best” for predicting first-serve percentage.

Questions of Interest

Which subset of predictors is the best for predicting winners earned using Mallow’s C_P as criteria? What is the 95% confidence interval for the number of winners earned by a player with the average amounts of break points won, aces won, first-serve percentage, first serves won, unforced errors, and net points won?

Regression Method

First, we compile a predictor correlation matrix to better understand the relationships between predictors. Then we check for leverage points and outliers. Next, we see if the outliers are influential and remove any that are influential, however, none of them seemed to be so. We then perform a stepwise regression with all the predictor variables and then perform the best subsets regression. We determine the best model for the stepwise regression by picking the model with the smallest AIC value and we determine the best model for the best subsets regression by selecting the model with a C_p value near the number of parameters and also passes the p-test for each predictor variable. After finding the best model, we check the L.I.N.E. conditions.

For finding the confidence interval we will use the predict() function on whatever model we find best. We have to keep in mind that any transformations done on Y will require us to do the inverse transformation on the interval. For example, if we take log(Y) we must take e^(interval) to put the interval in the correct units.

Regression Analysis, Results, and Interpretation

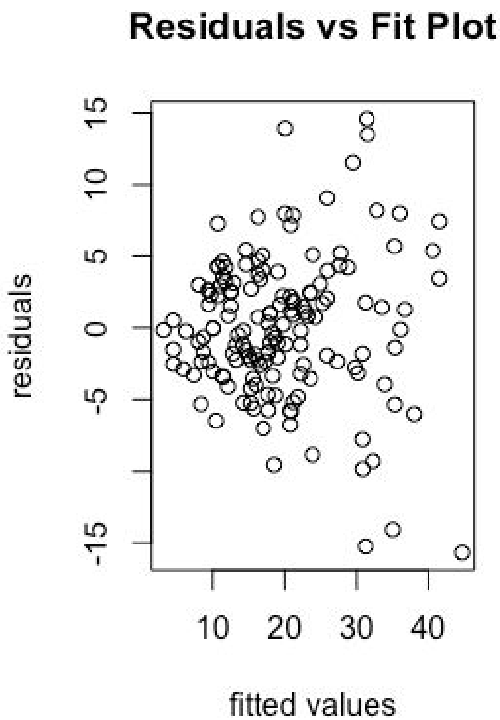
Let’s define our variables:

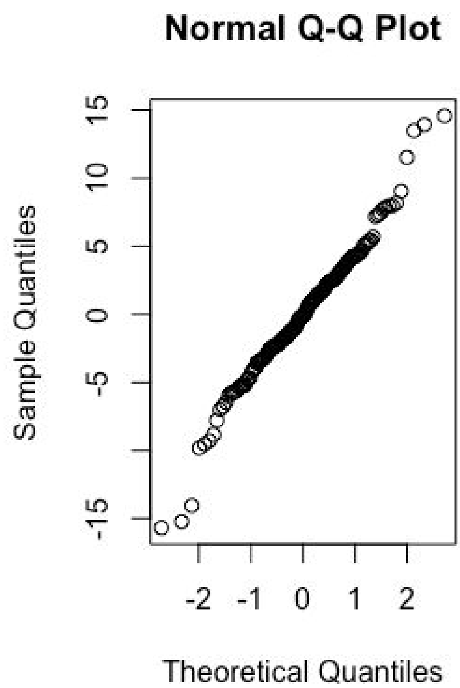
- Y = # of winners (shots that the opponent cannot return) earned

- x_1 = # of break points (the receiving player wins the game by scoring the next point) won
- x_2 = # of aces (legal serves that are not touched by the receiver) won
- x_3 = first serve percentage
- x_4 = # of first serves won
- x_5 = # of unforced errors (lost points by making a mistake in a situation where you should be in full control)
- x_6 = # of net points (points won on approaching the net) won

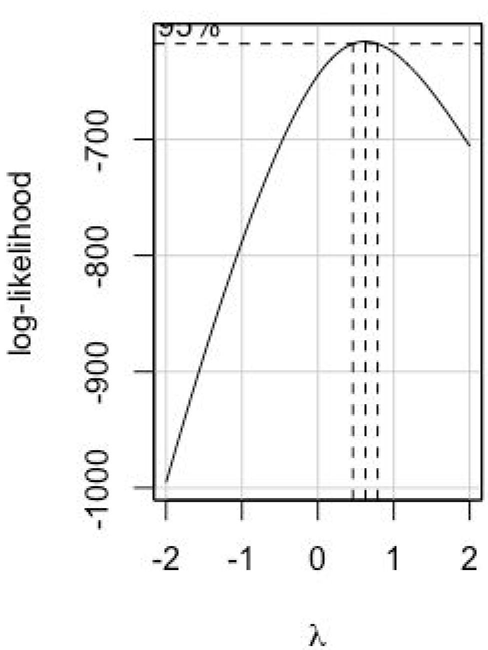
First, we made a scatterplot matrix with all the variables and to see if multicollinearity was a problem. None of the correlations between any two predictor variables were strong, so we continued. We found 3 leverage points and 4 outliers, but none of the outliers were influential so they were kept in the model. This was indicated by the small change in the slope parameter estimates and p-values. Then we used a stepwise regression to find that the best fit used x_1, x_2, x_5, and x_6 based on the AIC value of each model. We tried some different interactions to include in the model but none seemed to make a difference.

Next, we conducted the best subsets regression and used Mallow's C_P to pick the best model. The 4- and 5-predictor model both had similar C_ps near each of their respected amount of parameters, so we picked the 4-predictor model with x_1, x_2, x_5, and x_6 because the p-value for each x-value was below 0.05, whereas for the 5-predictor model this wasn't the case. At this point, we choose the regression model with x_1, x_2, x_5, and x_6 as our predictors because this model proved best in the stepwise regression and the best subsets regression.
Next, we set up a residual vs. fit plot and Normal QQ Plot to check our LINE assumptions

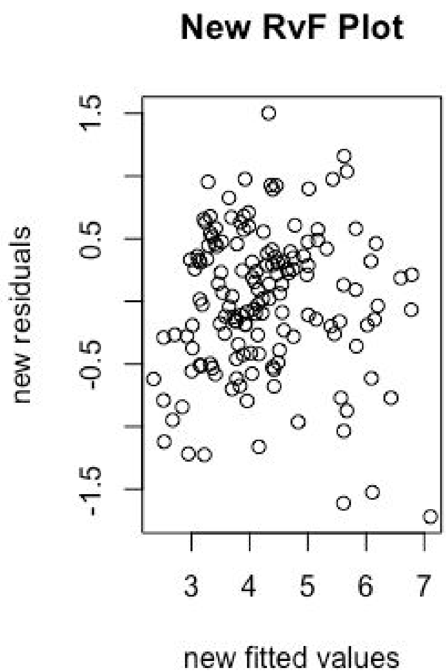


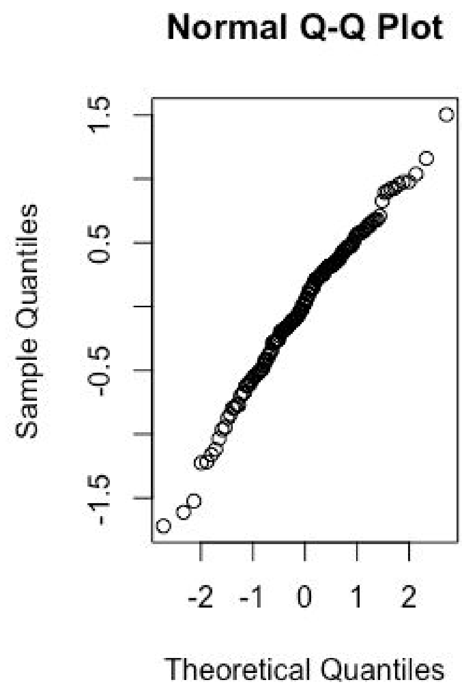


The RvF plot seems to show a bit of fanning and the Normal QQ plot is not completely linear. With these facts in mind, we use the boxcox function to figure out how to transform our Y.



We see that taking the square root of Y is the best way to fix the normality of the Normal QQ plot and slight fanning in the RvF plot. This is because lambda is the power that we take Y to, and it is about 0.5. Now our RvF plot is more evenly but randomly distributed. The new Norm QQ plot is more linear.





Our new Linear model is $\ln(Y) \sim x_1+x_2+x_5+x_6$. This model has a very similar R^2 value as before we took the ln(Y), and all the p-values are lower than any significance level. The R^2 value is 0.75, which indicates that these predictors explain 75% of the variance in the response: winners earned.

Next we want to find a 95% confidence interval for the predicted number of winners earned with an average amount of break points won, aces, unforced errors, and net points won. Using R we get an interval of [4.163858, 4.351212]. However, this is for our new model where the sqrt(Y) is the response, so we must square each end of the interval to put it back into original units. So, we are 95% confident the prediction will be in the interval [17.33771, 18.93305].

Conclusion

We can say that break points won, aces, unforced errors, and net points won can decently predict a woman tennis player's amount of winners earned in a match in the

2013 US Open. This response is important in tennis because scoring points is heavily based on unforced errors and winners. Of these, winners are more controlled by a player by working on certain parts of the game. An individual player wants to increase their winners earned, so it is useful to know which part of the game they are lacking in and to change their strategy (such as serving, or net points). This is something that a tennis coach could pay attention to in each individual game. We could add more predictors to make our guess more accurate. There are probably a lot of different variables to consider adding such as court positioning.

Appendix

```
> tennis<-read.csv(file='USOpen-women-2013.csv')
> View(tennis)
>
> y1=tennis$WNR.1
> y2=tennis$WNR.2
> y=c(y1,y2)
>
> x11=tennis$BPW.1
> x12=tennis$BPW.2
> x1=c(x11,x12)
>
> x21=tennis$ACE.1
> x22=tennis$ACE.2
> x2=c(x21,x22)
>
> x31=tennis$FSP.1
> x32=tennis$FSP.2
> x3=c(x31,x32)
>
> x41=tennis$FSW.1
> x42=tennis$FSW.2
> x4=c(x41,x42)
>
> x51=tennis$UFE.1
> x52=tennis$UFE.2
> x5=c(x51,x52)
>
> x61=tennis$NPW.1
```

```
> x62=tennis$NPW.2
> x6=c(x61,x62)
> x6[is.na(x6)]<-0
>
> df=data.frame('x1'=x1,'x2'=x2,'x3'=x3,'x4'=x4,'x5'=x5,'x6'=x6,'y'=y)
> View(df)
> x1=df$x1
> x2=df$x2
> x3=df$x3
> x4=df$x4
> x5=df$x5
> x6=df$x6
> y=df$y
>
> pairs(y~x1+x2+x3+x4+x5+x6)
> cor(df)
> fit.all=lm(y~x1+x2+x3+x4+x5+x6)
> summary(fit.all)

Call:
lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6)

Residuals:
     Min       1Q   Median       3Q      Max
-15.1638  -3.0162  -0.0397   2.9848  14.8731

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.02367    3.11919   0.008   0.9940
x1           1.28023    0.20523   6.238 4.60e-09 ***
x2           1.62037    0.17376   9.325  < 2e-16 ***
x3          -0.05672    0.05074  -1.118   0.2655
x4           0.10156    0.05853   1.735   0.0848 .
x5           0.30181    0.04286   7.042 7.04e-11 ***
x6           0.32860    0.06185   5.313 3.97e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.025 on 145 degrees of freedom
```

```
Multiple R-squared:  0.7624,	Adjusted R-squared:  0.7526
F-statistic: 77.55 on 6 and 145 DF,  p-value: < 2.2e-16

>
> #find leverage points
> h=hatvalues(fit.all)
> p=7
> n=length(y)
> which(h>3*p/n)
 64  84 140
 64  84 140
> #find outliers using studentized deleted residuals
> rs=rstudent(fit.all)
> which(abs(rs)>3)
 64 127 130 138
 64 127 130 138
>
> #check if outliers are influential
> df1=df[-c(64,127,130,138),]
> summary(lm(df1$y~df1$x1+df1$x2+df1$x3+df1$x4+df1$x5+df1$x6))

Call:
lm(formula = df1$y ~ df1$x1 + df1$x2 + df1$x3 + df1$x4 + df1$x5 +
    df1$x6)

Residuals:
     Min       1Q   Median       3Q      Max
-13.2918  -2.6600  -0.2394   3.0172  12.7798

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.41511    2.78472  -1.226    0.222
df1$x1       1.27373    0.18156   7.015 8.81e-11 ***
df1$x2       1.82088    0.15360  11.855  < 2e-16 ***
df1$x3      -0.01270    0.04497  -0.282    0.778
df1$x4       0.07603    0.05137   1.480    0.141
df1$x5       0.28476    0.03958   7.194 3.40e-11 ***
df1$x6       0.50224    0.07317   6.864 1.95e-10 ***
---
```

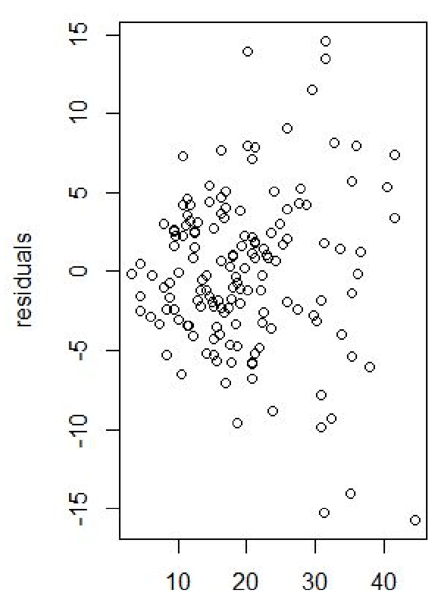
```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.343 on 141 degrees of freedom
Multiple R-squared:  0.8238,	Adjusted R-squared:  0.8163
F-statistic: 109.8 on 6 and 141 DF,  p-value: < 2.2e-16

> #p-values didn't change much and neither did slope values, so not influential
> #stepwise including different interactions
> mod0=lm(y~1)
> mod.upper=lm(y~x1+x2+x3+x4+x5+x6+I(x3*x4)+I(x3^2))
> step(mod0,scope=list(lower=mod0,upper=mod.upper))
Start:  AIC=704.09
y ~ 1

            Df Sum of Sq     RSS    AIC
+ x4         1    5634.3  9777.5 636.92
+ x2         1    5348.8 10063.0 641.30
+ x5         1    5028.6 10383.2 646.06
+ x6         1    3688.2 11723.6 664.51
+ I(x3 * x4)  1    3664.3 11747.4 664.82
+ x1         1    2692.3 12719.4 676.90
<none>                   15411.8 704.09
+ I(x3^2)     1      42.7 15369.1 705.67
+ x3         1      20.2 15391.6 705.89

Step:  AIC=636.92
y ~ x4

            Df Sum of Sq     RSS    AIC
+ x2         1    2546.3  7231.2 593.07
+ x6         1    1572.2  8205.3 612.28
+ I(x3 * x4)  1    1537.4  8240.0 612.92
+ x5         1    1349.5  8427.9 616.34
+ x3         1    1176.5  8601.0 619.43
+ I(x3^2)     1    1145.0  8632.4 619.99
+ x1         1     732.5  9045.0 627.09
<none>                    9777.5 636.92
- x4         1    5634.3 15411.8 704.09
```

```
Step:  AIC=593.07
y ~ x4 + x2

             Df Sum of Sq     RSS    AIC
+ x6          1   1517.08  5714.1 559.27
+ x5          1   1370.20  5861.0 563.13
+ x1          1    774.81  6456.3 577.84
+ I(x3 * x4)  1    506.29  6724.9 584.03
+ x3          1    446.45  6784.7 585.38
+ I(x3^2)     1    402.72  6828.4 586.36
<none>                     7231.2 593.07
- x2          1   2546.30  9777.5 636.92
- x4          1   2831.82 10063.0 641.30

Step:  AIC=559.27
y ~ x4 + x2 + x6

             Df Sum of Sq    RSS    AIC
+ x5          1   1026.08 4688.0 531.19
+ x1          1    475.04 5239.0 548.08
+ I(x3 * x4)  1    300.50 5413.6 553.06
+ x3          1    279.81 5434.3 553.64
+ I(x3^2)     1    250.43 5463.6 554.46
<none>                    5714.1 559.27
- x6          1   1517.08 7231.2 593.07
- x4          1   1592.30 7306.4 594.64
- x2          1   2491.21 8205.3 612.28

Step:  AIC=531.19
y ~ x4 + x2 + x6 + x5

             Df Sum of Sq    RSS    AIC
+ x1          1    994.89 3693.1 496.93
<none>                    4688.0 531.19
+ I(x3 * x4)  1     51.29 4636.7 531.52
+ x3          1     43.84 4644.1 531.76
+ I(x3^2)     1     33.09 4654.9 532.11
- x4          1    410.02 5098.0 541.93
- x5          1   1026.08 5714.1 559.27
```

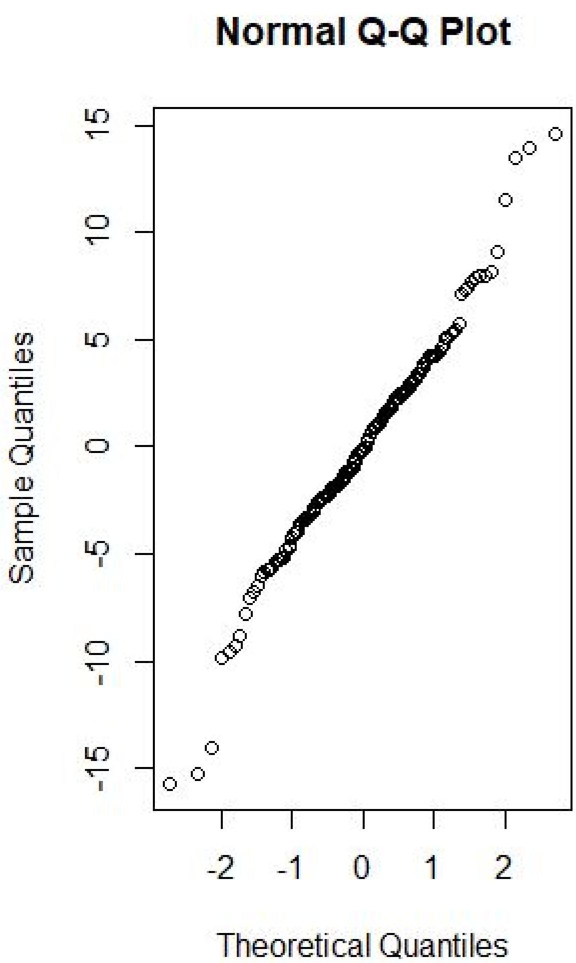
```
- x6          1   1172.97 5861.0 563.13
- x2          1   2515.13 7203.1 594.47

Step:  AIC=496.93
y ~ x4 + x2 + x6 + x5 + x1

             Df Sum of Sq    RSS    AIC
- x4          1     46.31 3739.4 496.83
<none>                    3693.1 496.93
+ I(x3 * x4)  1     34.54 3658.6 497.50
+ x3          1     31.55 3661.5 497.63
+ I(x3^2)     1     23.80 3669.3 497.95
- x6          1    736.55 4429.7 522.57
- x1          1    994.89 4688.0 531.19
- x5          1   1545.93 5239.0 548.08
- x2          1   2579.30 6272.4 575.45

Step:  AIC=496.83
y ~ x2 + x6 + x5 + x1

             Df Sum of Sq    RSS    AIC
<none>                    3739.4 496.83
+ x4          1     46.31 3693.1 496.93
+ I(x3 * x4)  1     21.92 3717.5 497.93
+ x3          1      1.83 3737.6 498.75
+ I(x3^2)     1      0.79 3738.6 498.79
- x6          1    777.44 4516.8 523.54
- x1          1   1358.60 5098.0 541.93
- x5          1   2514.02 6253.4 572.98
- x2          1   3003.28 6742.7 584.43

Call:
lm(formula = y ~ x2 + x6 + x5 + x1)

Coefficients:
(Intercept)           x2           x6           x5           x1  
    -2.7510       1.7399       0.3409       0.3458       1.3927  

> #best fit is y~x1+x2+x5+x6
```

```
> fit1=lm(y~x1+x2+x5+x6)
> summary(fit1)

Call:
lm(formula = y ~ x1 + x2 + x5 + x6)

Residuals:
    Min       1Q   Median       3Q      Max
-15.6920  -2.7889  -0.1421   2.9449  14.5864

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.75104    1.21035  -2.273   0.0245 *
x1           1.39269    0.19057   7.308 1.60e-11 ***
x2           1.73988    0.16013  10.866  < 2e-16 ***
x5           0.34582    0.03479   9.941  < 2e-16 ***
x6           0.34093    0.06167   5.528 1.44e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.044 on 147 degrees of freedom
Multiple R-squared:  0.7574,	Adjusted R-squared:  0.7508
F-statistic: 114.7 on 4 and 147 DF,  p-value: < 2.2e-16

> #best subsets
> library(leaps)
> mod=regsubsets(cbind(x1,x2,x3,x4,x5,x6),y)
> summary.mod=summary(mod)
> summary.mod$which
  (Intercept)    x1    x2    x3    x4    x5    x6
1        TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
2        TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
3        TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE
4        TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
5        TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
6        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
> summary.mod$cp
[1] 239.194379 115.167036  34.870314   6.083294   6.249346
[6]   7.000000
```

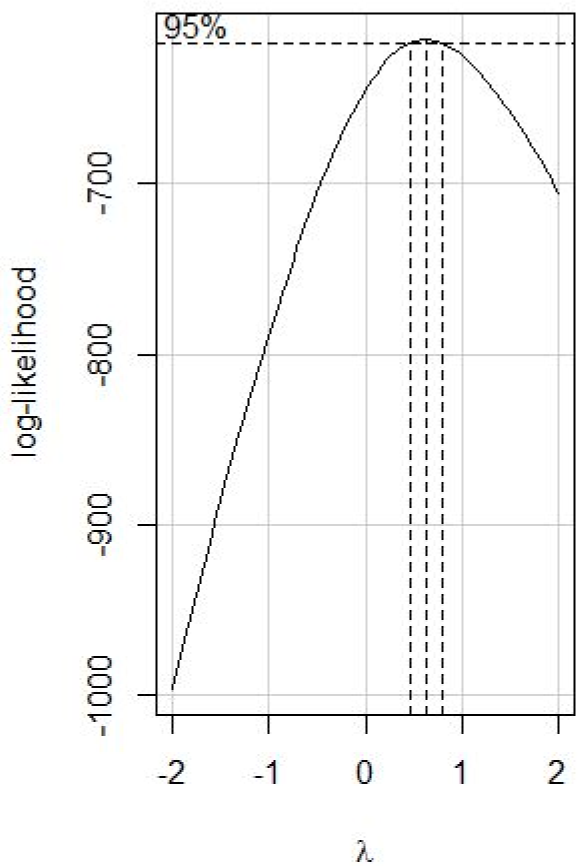
```
> fit2=lm(y~x1+x2+x4+x5+x6)
> summary(fit2)

Call:
lm(formula = y ~ x1 + x2 + x4 + x5 + x6)

Residuals:
    Min       1Q   Median       3Q      Max 
-15.5318  -2.8986  -0.0791   2.9117  14.4129 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -3.17278    1.24654  -2.545    0.012 *  
x1           1.28755    0.20530   6.271 3.83e-09 ***
x2           1.67774    0.16615  10.098  < 2e-16 ***
x4           0.06813    0.05036   1.353    0.178    
x5           0.31730    0.04059   7.818 9.81e-13 ***
x6           0.33325    0.06176   5.396 2.69e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.029 on 146 degrees of freedom
Multiple R-squared:  0.7604,	Adjusted R-squared:  0.7522 
F-statistic: 92.66 on 5 and 146 DF,  p-value: < 2.2e-16

> e=y-fitted(fit1)
> plot(fitted(fit1),e,xlab='fitted values',ylab='residuals')
```



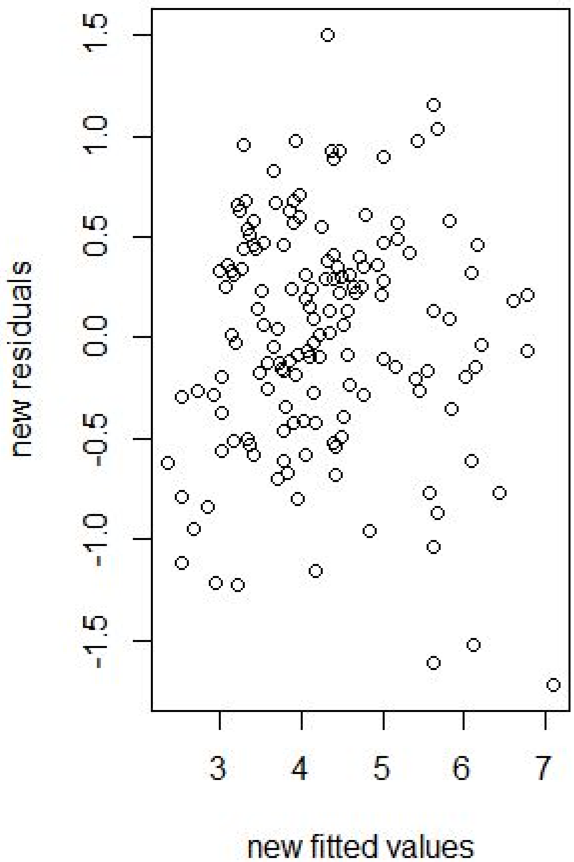
```
> qqnorm(e)
```




```
> boxCox(fit1)
```

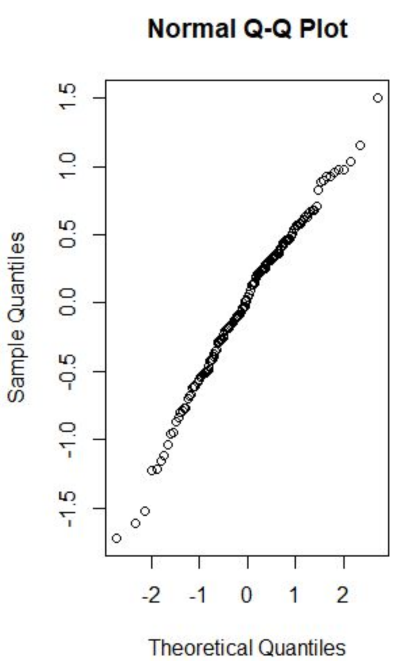



```
> #transform y to fix non normality and slight fanning
> newfit=lm(sqrt(y)~x1+x2+x5+x6)
> new_e=sqrt(y)-fitted(newfit)
> plot(fitted(newfit),new_e,xlab='new fitted values',ylab='new residuals')
```



new residuals
new fitted values

> qqnorm(new_e)



Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles

```
> #passes LINE conditions with transformations
> #sqrt for Y from boxcox
> summary(newfit)

Call:
lm(formula = sqrt(y) ~ x1 + x2 + x5 + x6)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.71953 -0.36153  0.03004  0.38625  1.50297 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.698666   0.140244  12.112  < 2e-16 ***
x1          0.170214   0.022081   7.708 1.76e-12 ***
x2          0.191766   0.018554  10.336  < 2e-16 ***
x5          0.039119   0.004031   9.705  < 2e-16 ***
x6          0.039899   0.007146   5.584 1.11e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5844 on 147 degrees of freedom
Multiple R-squared:  0.7524,	Adjusted R-squared:  0.7456 
F-statistic: 111.7 on 4 and 147 DF,  p-value: < 2.2e-16

> #95% CI for mean data
> new = data.frame(x1 =mean(x1) , x2 = mean(x2),x5=mean(x5),x6=mean(x6))
> c.i. = predict(newfit, new, interval = 'confidence', level = .95)
> #must square it because of our sqrt transformation
> c.i.=c.i.^2
> c.i.
      fit      lwr      upr
1 18.1266 17.33771 18.93305
```