Lab 04

CSCI E-106 TA's

10/03/2019

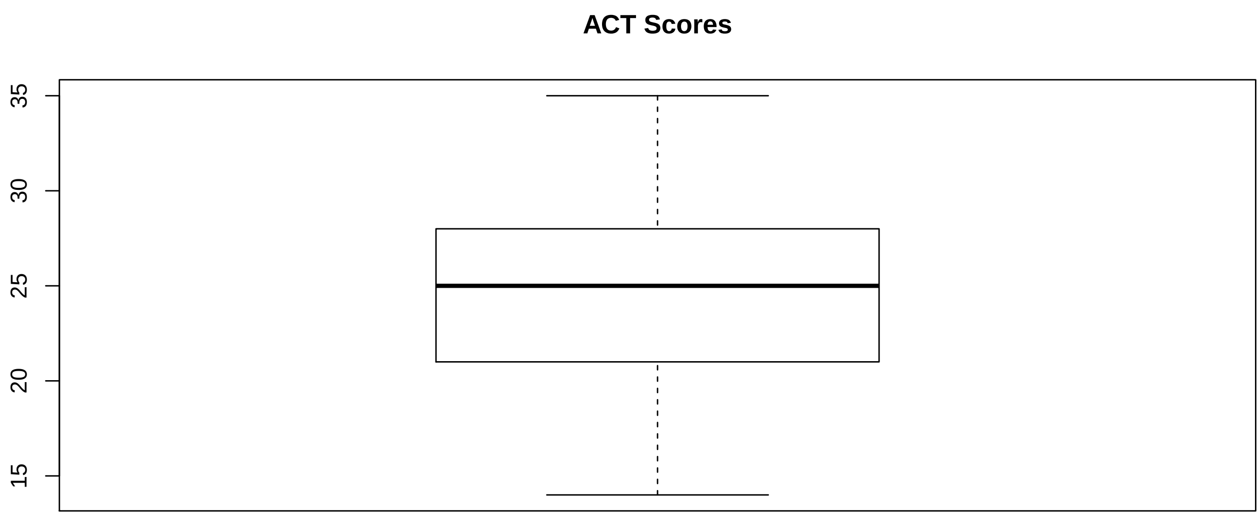
```
## ggplot2 loaded properly
## knitr loaded properly
## MASS loaded properly
## formatR loaded properly
```

(Textbook 3.3) Refer to Grade point average Problem 1.19.

Please use dataset titled: CH01PR19.txt**

a. Prepare a box plot for the ACT scores Xi. Are there any noteworthy features in this plot?

```r
Dataset_1_19 = read.table("CH01PR19.txt", header = FALSE, sep = "", col.names = c("V1",
    "V2"))

boxplot(Dataset_1_19$V2, main = "ACT Scores")
```

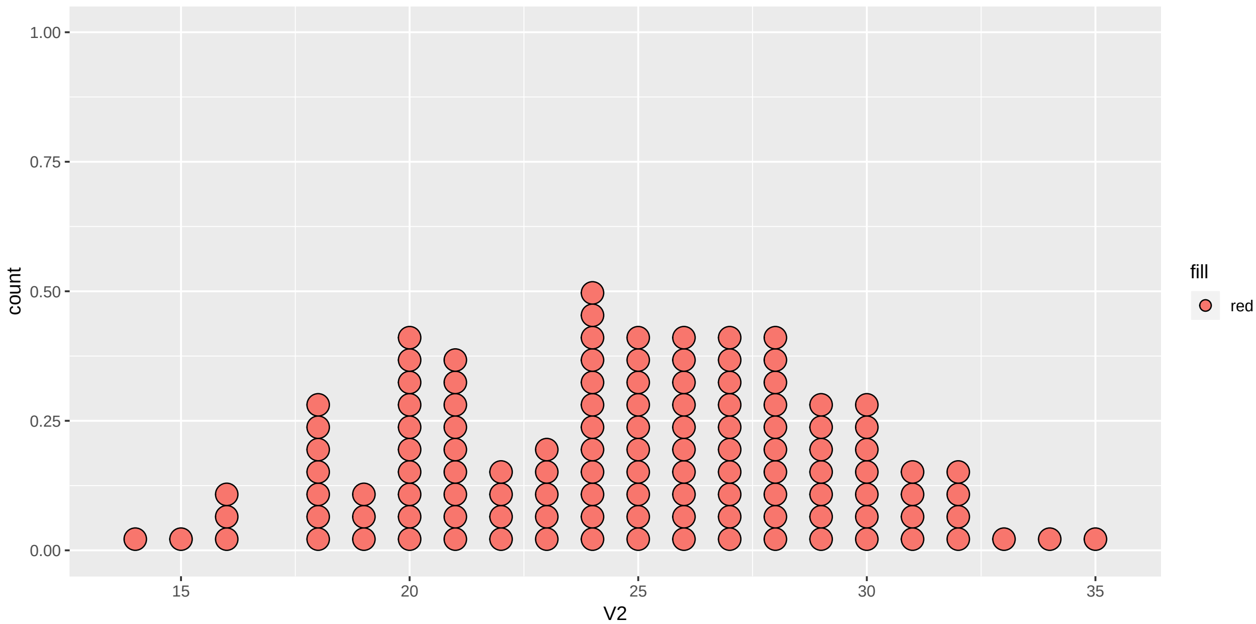


Part A Conclusion: We do not see any outliers. We see symmetric distributions in this case.

b. Prepare a dot plot of the residuals. What information does this plot provide?

```r
ggplot(Dataset_1_19, aes(x = V2, fill = "red")) + geom_dotplot(dotsize = 0.7)
```

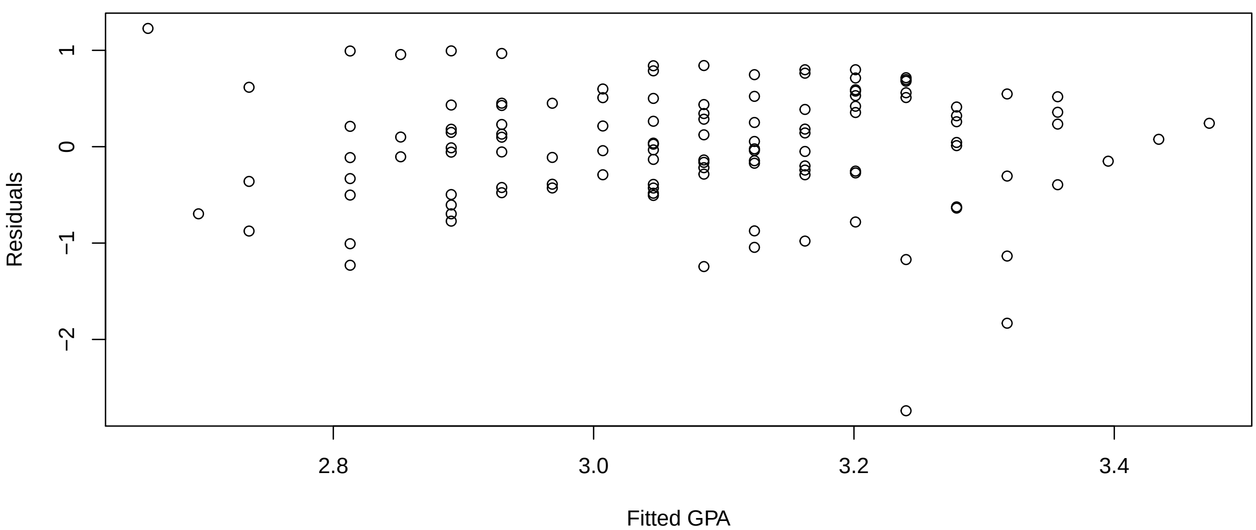
```
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```



Part B Conclusion: Again, we do not see any outliers. We see symmetric distributions in this case.

c. Plot the residual e_i against the fitted values $Yhat_i$. What departures from regression model (2.1) can be studied from this plot? What are your findings?

```
lmfit19 = lm(V1 ~ V2, data = Dataset_1_19)

plot(lmfit19$fitted.values, lmfit19$residuals, xlab = "Fitted GPA", ylab = "Residuals")
```



Part C Conclusion: We do not see any outliers or any non-linearity in our plot. Thus, we can say that we have a constant variance.

d. Prepare a normal probability plot of the residuals. Also obtain the coefficient of correlation between the ordered residuals and their expected values under normality. Test the reasonableness of the normality assumption here using Table B.6 and a = .05. What do you conclude?

```
summary(lmfit19)
```

```
##
```

```
## Call:
## lm(formula = V1 ~ V2, data = Dataset_1_19)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.74004 -0.33827  0.04062  0.44064  1.22737
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.11405    0.32089   6.588  1.3e-09 ***
## V2           0.03883    0.01277   3.040  0.00292 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6231 on 118 degrees of freedom
## Multiple R-squared:  0.07262,	Adjusted R-squared:  0.06476
## F-statistic:  9.24 on 1 and 118 DF,  p-value: 0.002917
```

```r
ei = lmfit19$residuals

ri = rank(ei)

zr = (ri - 0.375)/(120 + 0.25)
print(zr)
```

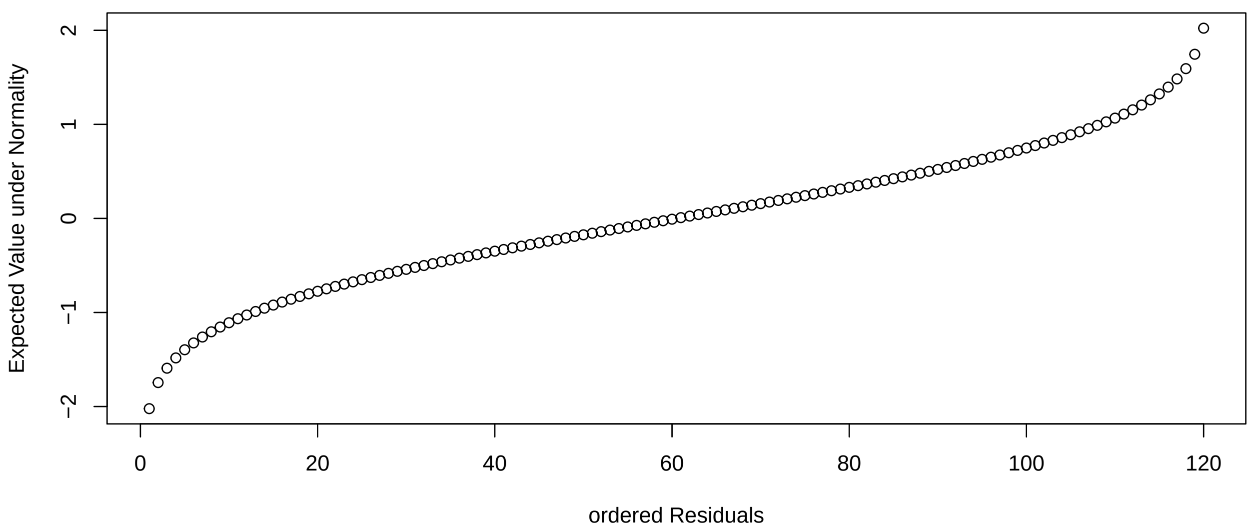
```
##           1           2           3           4           5           6
## 0.969854470 0.994802495 0.836798337 0.204781705 0.529106029 0.820166320
##           7           8           9          10          11          12
## 0.221413721 0.928274428 0.005197505 0.512474012 0.653846154 0.645530146
##          13          14          15          16          17          18
## 0.495841996 0.454261954 0.354469854 0.329521830 0.745322245 0.262993763
##          19          20          21          22          23          24
## 0.371101871 0.104989605 0.179833680 0.728690229 0.811850312 0.911642412
##          25          26          27          28          29          30
## 0.687110187 0.462577963 0.096673597 0.238045738 0.903326403 0.554054054
##          31          32          33          34          35          36
## 0.953222453 0.246361746 0.296257796 0.637214137 0.396049896 0.487525988
##          37          38          39          40          41          42
## 0.761954262 0.479209979 0.703742204 0.803534304 0.362785863 0.138253638
##          43          44          45          46          47          48
## 0.154885655 0.088357588 0.038461538 0.196465696 0.046777547 0.121621622
##          49          50          51          52          53          54
## 0.537422037 0.978170478 0.271309771 0.861746362 0.562370062 0.337837838
##          55          56          57          58          59          60
## 0.770270270 0.712058212 0.612266112 0.113305613 0.504158004 0.878378378
##          61          62          63          64          65          66
## 0.346153846 0.279625780 0.662162162 0.853430353 0.129937630 0.188149688
##          67          68          69          70          71          72
## 0.229729730 0.695426195 0.063409563 0.778586279 0.570686071 0.437629938
##          73          74          75          76          77          78
## 0.254677755 0.387733888 0.870062370 0.412681913 0.603950104 0.445945946
##          79          80          81          82          83          84
## 0.936590437 0.520790021 0.737006237 0.579002079 0.055093555 0.795218295
##          85          86          87          88          89          90
## 0.545738046 0.312889813 0.587318087 0.895010395 0.961538462 0.213097713
```

```
##          91          92          93          94          95          96 
## 0.944906445 0.071725572 0.678794179 0.595634096 0.786902287 0.919958420 
##          97          98          99         100         101         102 
## 0.429313929 0.420997921 0.404365904 0.163201663 0.021829522 0.030145530 
##         103         104         105         106         107         108 
## 0.470893971 0.620582121 0.379417879 0.628898129 0.287941788 0.720374220 
##         109         110         111         112         113         114 
## 0.845114345 0.321205821 0.753638254 0.670478170 0.171517672 0.146569647 
##         115         116         117         118         119         120 
## 0.013513514 0.986486486 0.828482328 0.886694387 0.080041580 0.304573805
```

```
# residual standard error = .6231 which is found from our summary
zr1 = sqrt(0.6231) * qnorm(zr)

cor.test(zr, zr1)
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  zr and zr1
## t = 54.569, df = 118, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9724678 0.9865674
## sample estimates:
##       cor 
## 0.9807569
```

```
plot(ri, zr1, xlab = "ordered Residuals", ylab = "Expected Value under Normality")
```



Part D Conclusion: So we see our rse: .6231
H0: Normal Ha: Not normal
r = .987 If rse >= .987 conclude H0, otherwise Ha. So we conclude Ha.

e. Conduct the Brown-Forsythe test to determine whether or not the error variance varies with the level of X. Divide the data into the two groups, X < 26, X >= 26, and use a = .01. State the decision rule and conclusion. Does your conclusion support your preliminary findings in part (c)?

```
summary(lmfit19)
```

```
##
## Call:
## lm(formula = V1 ~ V2, data = Dataset_1_19)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.74004 -0.33827  0.04062  0.44064  1.22737
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.11405    0.32089   6.588  1.3e-09 ***
## V2           0.03883    0.01277   3.040  0.00292 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6231 on 118 degrees of freedom
## Multiple R-squared:  0.07262,    Adjusted R-squared:  0.06476
## F-statistic:  9.24 on 1 and 118 DF,  p-value: 0.002917
```

```r
df = data.frame(cbind(Dataset_1_19[, 1], Dataset_1_19[, 2], ei))
df1 = df[df[, 2] < 26, ]
df2 = df[df[, 2] >= 26, ]

summary(df1[, 3])
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.243728 -0.390900 -0.032900  0.005155  0.427581  1.227371
```

```r
summary(df2[, 3])
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -2.740036 -0.262709  0.142618 -0.006092  0.520464  0.798791
```

```r
# n1
n1 = length(df1[, 3])
print(n1)
```

```
## [1] 65
```

```r
# n2
n2 = length(df2[, 3])
print(n2)
```

```
## [1] 55
```

```r
d1 = abs(df1[, 3] + 0.0329)
d2 = abs(df2[, 3] - 0.142618)

# calculate means for our answer
mean(d1)
```

```
## [1] 0.4379603
```

```r
mean(d2)
```

```
## [1] 0.5065161
```

```r
s2 = (var(d1) * (65 - 1) + var(d2) * (55 - 1))/(120 - 2)
print(s2)
```

```
## [1] 0.1741184
```

```r
# calculate s
s = sqrt(s2)
print(s)
```

```
## [1] 0.4172749
```

```r
# testStastic = (mean.d1 - mean.d2) / (s * sqrt((1/n1)+1/n2)
testStastic = (0.43796 - 0.50652)/(0.417275 * sqrt((1/65) + (1/55)))
print(testStastic)
```

```
## [1] -0.8968005
```

```r
t = qt(0.995, 118)
print(t)
```

```
## [1] 2.618137
```

Part E Notes: We need to put our answer together

n1 = 65, mean.d1 = .43796 n2 = 55, mean.d2 =.50652 s = .417275 test Statistic = (.43796 - .50652)/ .417275 q(1/65)+(1/55) = -.8968005 t =(.995,118) = 2.61814

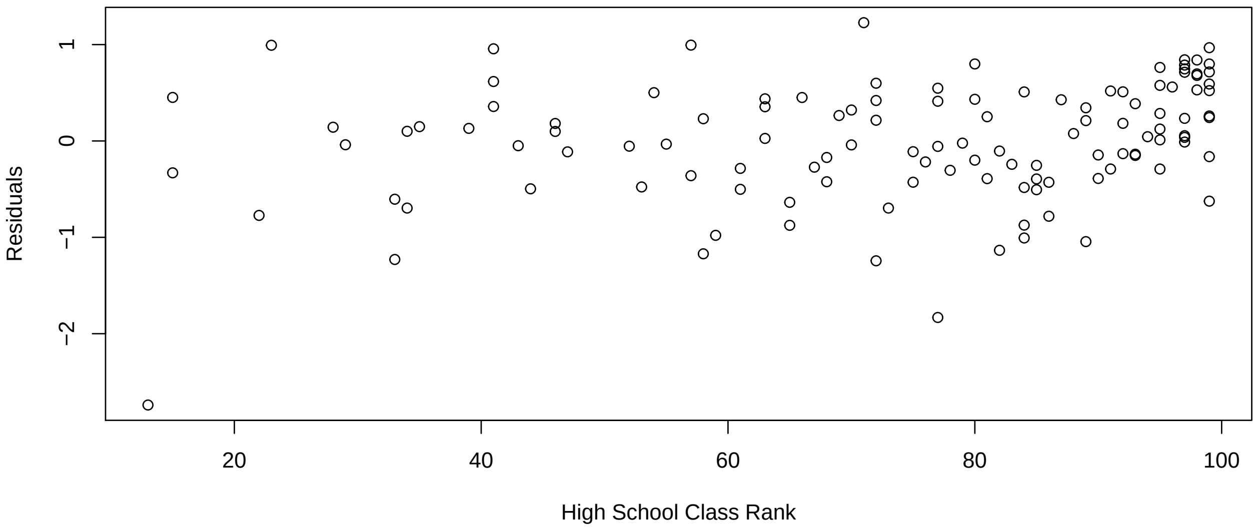
Part E Conclusion: If abs(testStatistic) <= 2.61814 conclude error variance constant, otherwise error variance not constant. Thus, we conclude error variance constant.

F. Information is given below for each student on two variables not included in the model, namely, intelligence test score (X2) and high school class rank percentile (X3). (Note that larger class rank percentiles indicate higher standing in the class, e.g., 1 % is near the bottom of the class and 99% is near the top of the class.) Plot the residuals against X2 and X3 on separate graphs to ascertain whether the model can be improved by including either of these variables. What do you conclude?

*Please use dataset titled: **CH03PR03.txt**

```r
Dataset_3_3 = read.table("CH03PR03.txt", header = FALSE, sep = "", col.names = c("V1",
    "V2", "V3", "V4"))

plot(Dataset_3_3$V3, ei, xlab = "Intelligence Test Score", ylab = "Residuals")
```


```r
plot(Dataset_3_3$V4, ei, xlab = "High School Class Rank", ylab = "Residuals")
```



Part F Conclusion: X2 is highly correlated with error term, but X3 doesn't show any correlation or pattern. X2 could be added to the model.

(Textbook 3.4) Refer to Copier maintenance Problem 1.20

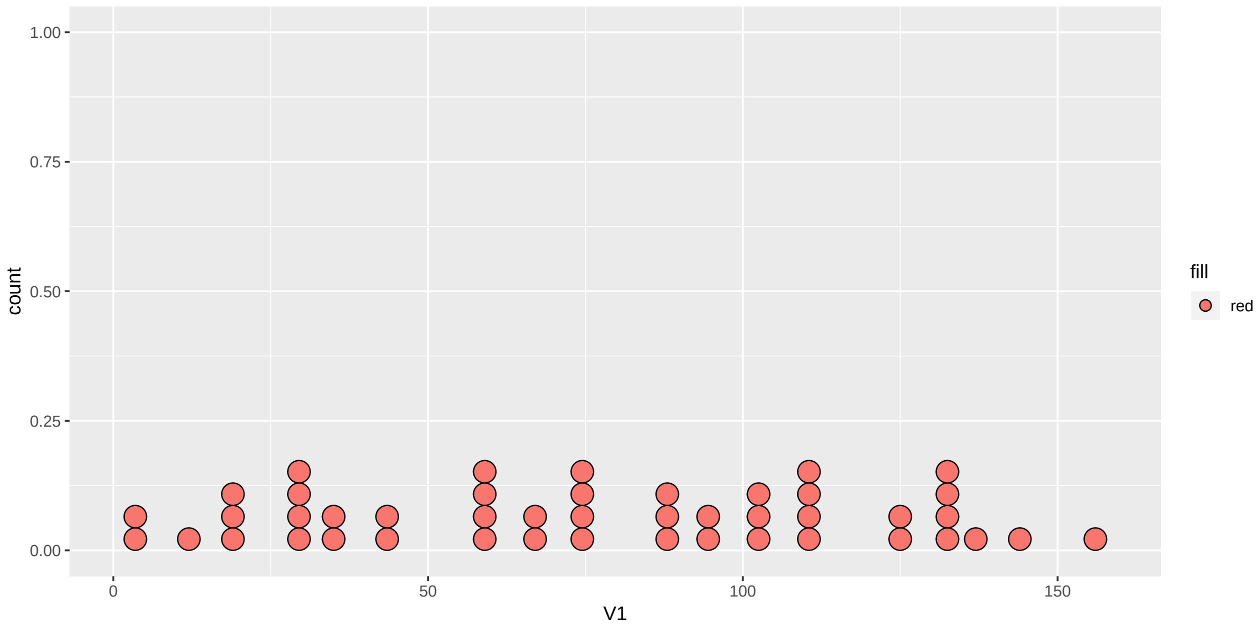
Please use dataset titled: ***CH01PR20.txt***

```
# Semi-descriptive var names for DF and cols
Dataset_1_20 = read.table("CH01PR20.txt", header = FALSE, sep = "", col.names = c("V1",
    "V2"))
```

a. Prepare a dot plot for the number of copiers serviced XI. What information is provided by this plot? Are there any outlying cases with respect to this variable?

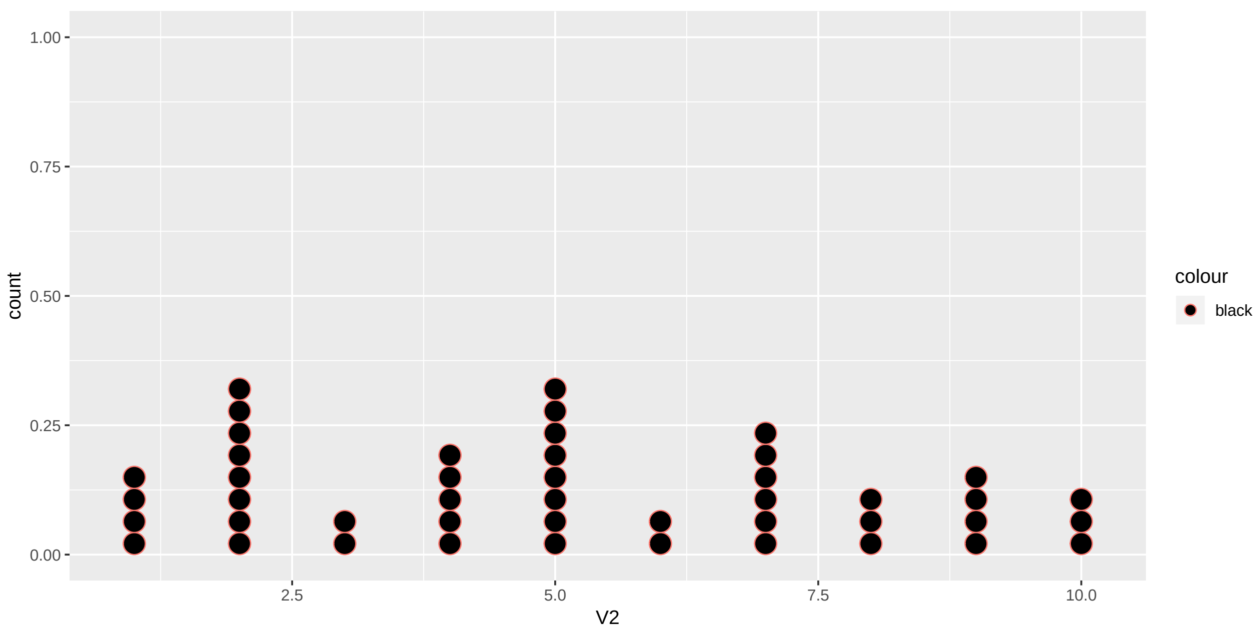
```
par(mfrow = c(1, 2))
ggplot(Dataset_1_20, aes(x = V1, fill = "red")) + geom_dotplot(dotsize = 0.7)
```

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




```r
ggplot(Dataset_1_20, aes(x = V2, color = "black")) + geom_dotplot(dotsize = 0.7)
```

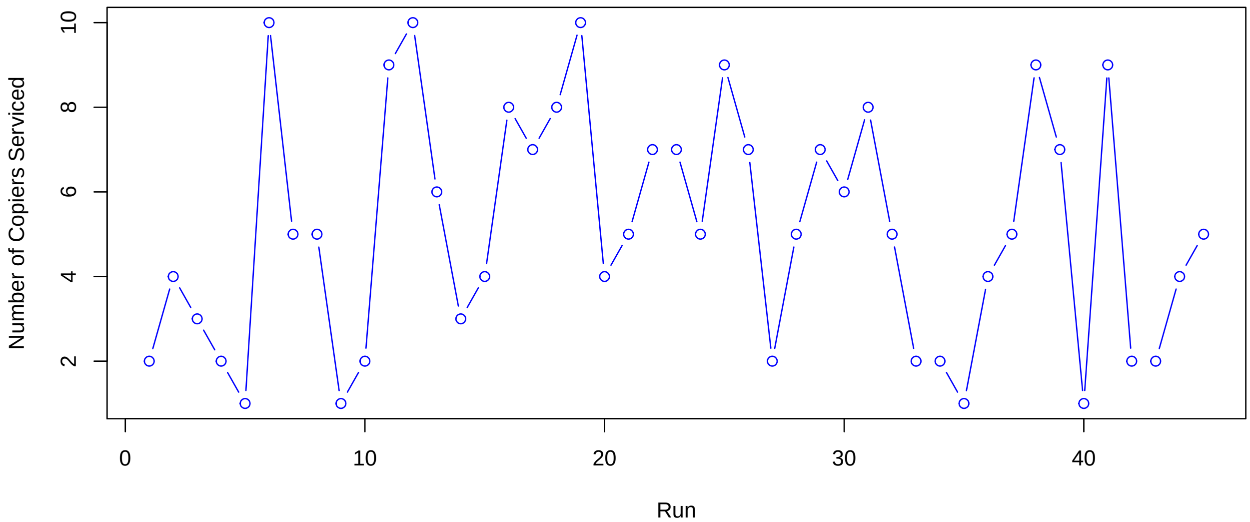
```
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```



Note: There are no outliers here on either plots.

b. The cases are given in time order. Prepare a time plot for the number of copiers serviced. What does your plot show?

```r
plot(Dataset_1_20$V2, type = "b", col = "blue", xlab = "Run", ylab = "Number of Copiers Serviced")
```



We do not see a time effect.

c. Prepare a stem-and-leaf plot of the residuals. Are there any noteworthy features in this plot?

```
stem(Dataset_1_20$V2)
```

```
##
##   The decimal point is at the |
##
##    1 | 0000
##    2 | 00000000
##    3 | 00
##    4 | 00000
##    5 | 00000000
##    6 | 00
##    7 | 000000
##    8 | 000
##    9 | 0000
##   10 | 000
```

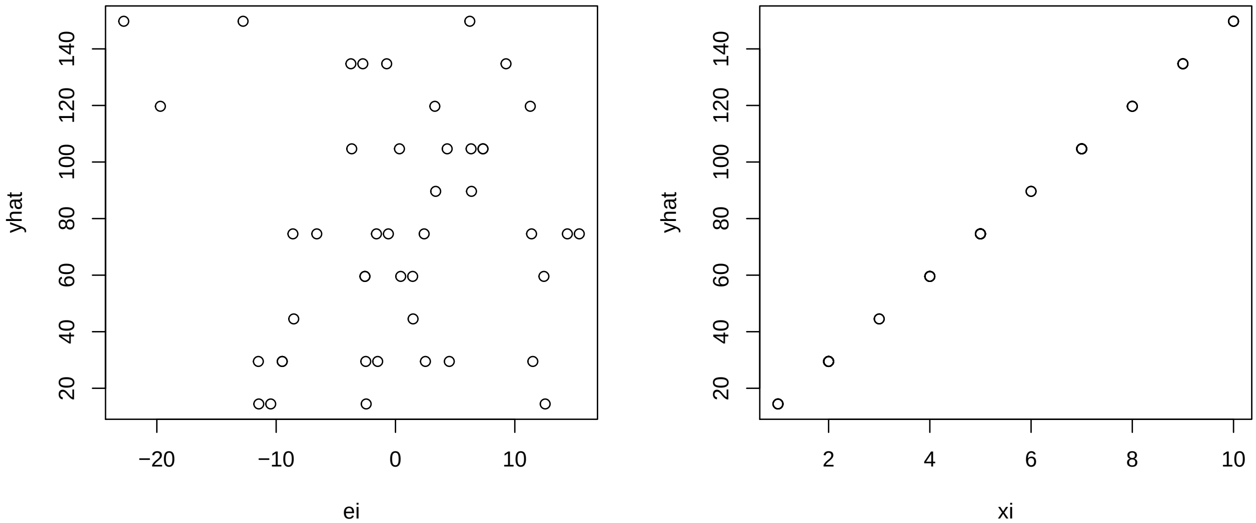
```
stem(Dataset_1_20$V1)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##    0 | 3428
##    2 | 00778246
##    4 | 1677
##    6 | 01682347
##    8 | 69036
##   10 | 0159122
##   12 | 3711247
##   14 | 46
```

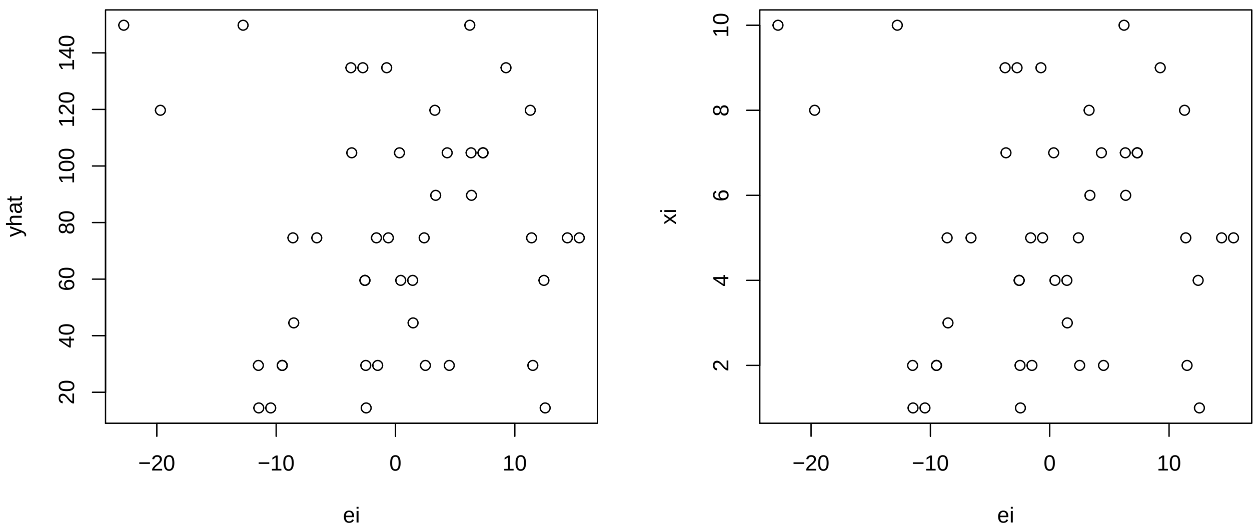
We do not see any outliers with the plot of the residuals. If anything, it is roughly normal or a little slightly right skewed.

d. Prepare residual plots of ei versus Y; and ei versus Xi on separate graphs. Do these plots provide the same information? What departures from regression model (2.1) can be studied from these plots? State your findings.

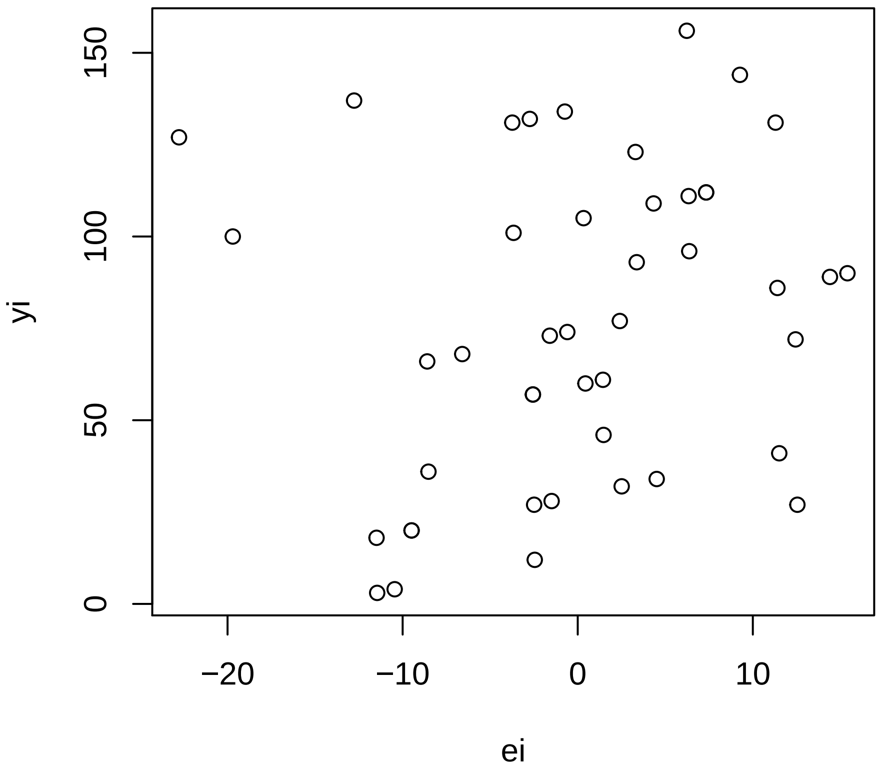
```r
f_1_20 = lm(V1 ~ V2, data = Dataset_1_20)
ei = f_1_20$residuals
yhat = f_1_20$fitted.values
yi = Dataset_1_20$V1
xi = Dataset_1_20$V2

par(mfrow = c(1, 2))
plot(ei, yhat)
plot(xi, yhat)
```




```r
plot(ei, yhat)
plot(ei, xi)
```




```r
plot(ei, yi)
```



In this case if you compare them then most of the plots look identical.

e. Prepare a normal probability plot of the residuals. Also obtain the coefficient of correlation between the ordered residuals and their expected values under normality. Does the normality assumption appear to be tenable here? Use Table B.6 and a = .10.

```
# there are two ways that this can be done

# long way to do this:

anova(f_1_20)
```

```
## Analysis of Variance Table
##
## Response: V1
##           Df Sum Sq Mean Sq F value    Pr(>F)
## V2         1  76960   76960  968.66 < 2.2e-16 ***
## Residuals 43   3416      79
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MSE = 79

summary(f_1_20)
```

```
##
## Call:
## lm(formula = V1 ~ V2, data = Dataset_1_20)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.7723  -3.7371   0.3334   6.3334  15.4039
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.5802     2.8039  -0.207    0.837
## V2           15.0352     0.4831  31.123   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.914 on 43 degrees of freedom
## Multiple R-squared:  0.9575, Adjusted R-squared:  0.9565
## F-statistic: 968.7 on 1 and 43 DF,  p-value: < 2.2e-16
```

```r
ei_rank = rank(ei)
z1 = (ei_rank - 0.375)/(45 + 0.25)
exp_rank = sqrt(MSE) * qnorm(z1)
part_e = data.frame(ei, ei_rank, z1, exp_rank)

# see all results
part_e
```

```
##             ei ei_rank         z1    exp_rank
## 1   -9.4903394     7.0 0.14640884  -9.3500293
## 2    0.4391645    24.0 0.52209945   0.4926145
## 3    1.4744125    26.0 0.56629834   1.4839527
## 4   11.5096606    41.0 0.89779006  11.2796464
## 5   -2.4550914    18.0 0.38950276  -2.4941628
## 6  -12.7723238     3.0 0.05801105 -13.9695002
## 7   -6.5960836    11.0 0.23480663  -6.4271285
## 8   14.4039164    44.0 0.96408840  16.0008575
## 9  -10.4550914     6.0 0.12430939 -10.2544052
## 10   2.5096606    28.0 0.61049724   2.4941628
## 11   9.2629243    38.0 0.83149171   8.5333491
## 12   6.2276762    33.0 0.72099448   5.2066894
## 13   3.3686684    30.0 0.65469613   3.5377721
## 14  -8.5255875    10.0 0.21270718  -7.0844521
## 15  12.4391645    42.0 0.91988950  12.4819464
## 16 -19.7018277     2.0 0.03591160 -16.0008575
## 17   0.3334204    23.0 0.50000000   0.0000000
## 18  11.2981723    39.0 0.85359116   9.3500293
## 19 -22.7723238     1.0 0.01381215 -19.5769662
## 20  -2.5608355    15.5 0.33425414  -3.8058910
## 21  -8.5960836     9.0 0.19060773  -7.7830270
## 22  -3.6665796    13.0 0.27900552  -5.2066894
## 23   4.3334204    31.0 0.67679558   4.0775194
## 24  -0.5960836    22.0 0.47790055  -0.4926145
## 25  -0.7370757    21.0 0.45580110  -0.9867480
## 26   7.3334204    36.5 0.79834254   7.4280027
## 27 -11.4903394     4.0 0.08011050 -12.4819464
## 28  -1.5960836    19.0 0.41160221  -1.9858486
## 29   6.3334204    34.0 0.74309392   5.8032206
## 30   6.3686684    35.0 0.76519337   6.4271285
## 31   3.2981723    29.0 0.63259669   3.0107749
## 32  15.4039164    45.0 0.98618785  19.5769662
## 33  -9.4903394     8.0 0.16850829  -8.5333491
## 34  -1.4903394    20.0 0.43370166  -1.4839527
```

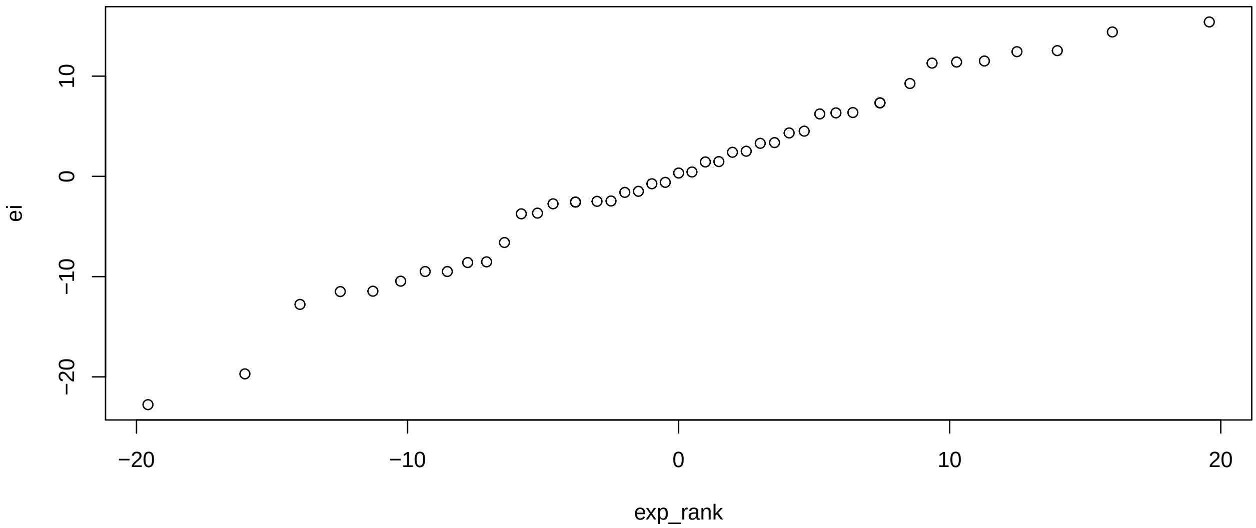
```
## 35 -11.4550914     5.0 0.10220994 -11.2796464
## 36  -2.5608355    15.5 0.33425414  -3.8058910
## 37  11.4039164    40.0 0.87569061  10.2544052
## 38  -2.7370757    14.0 0.30110497  -4.6327504
## 39   7.3334204    36.5 0.79834254   7.4280027
## 40  12.5449086    43.0 0.94198895  13.9695002
## 41  -3.7370757    12.0 0.25690608  -5.8032206
## 42   4.5096606    32.0 0.69889503   4.6327504
## 43  -2.4903394    17.0 0.36740331  -3.0107749
## 44   1.4391645    25.0 0.54419890   0.9867480
## 45   2.4039164    27.0 0.58839779   1.9858486
```

```r
print(part_e)
```

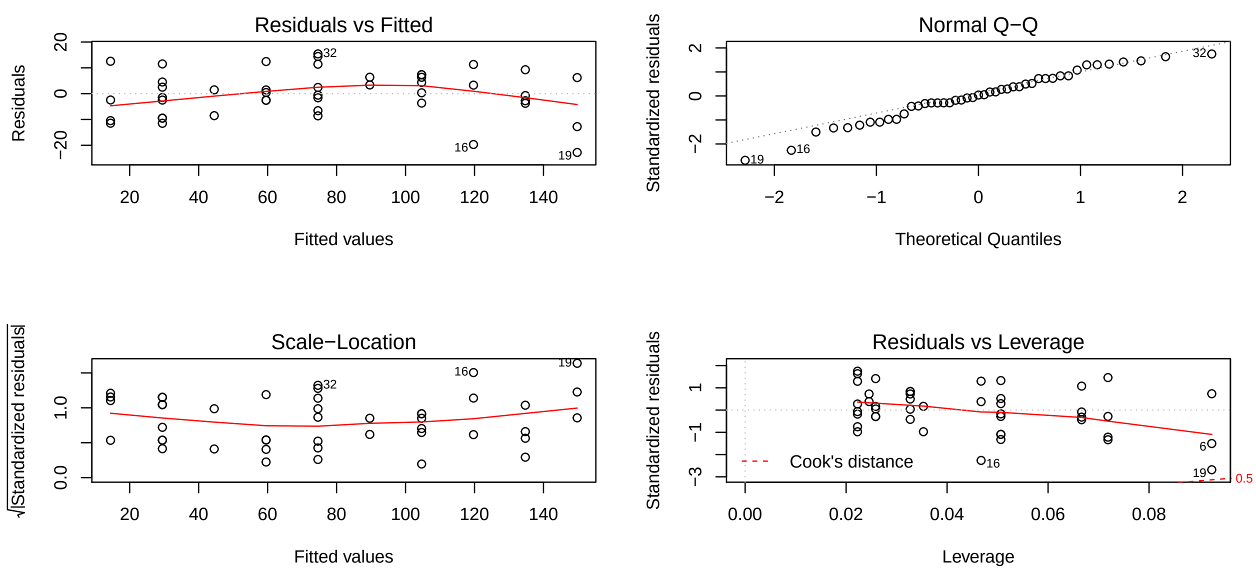
```
##             ei ei_rank         z1    exp_rank
## 1   -9.4903394     7.0 0.14640884  -9.3500293
## 2    0.4391645    24.0 0.52209945   0.4926145
## 3    1.4744125    26.0 0.56629834   1.4839527
## 4   11.5096606    41.0 0.89779006  11.2796464
## 5   -2.4550914    18.0 0.38950276  -2.4941628
## 6  -12.7723238     3.0 0.05801105 -13.9695002
## 7   -6.5960836    11.0 0.23480663  -6.4271285
## 8   14.4039164    44.0 0.96408840  16.0008575
## 9  -10.4550914     6.0 0.12430939 -10.2544052
## 10   2.5096606    28.0 0.61049724   2.4941628
## 11   9.2629243    38.0 0.83149171   8.5333491
## 12   6.2276762    33.0 0.72099448   5.2066894
## 13   3.3686684    30.0 0.65469613   3.5377721
## 14  -8.5255875    10.0 0.21270718  -7.0844521
## 15  12.4391645    42.0 0.91988950  12.4819464
## 16 -19.7018277     2.0 0.03591160 -16.0008575
## 17   0.3334204    23.0 0.50000000   0.0000000
## 18  11.2981723    39.0 0.85359116   9.3500293
## 19 -22.7723238     1.0 0.01381215 -19.5769662
## 20  -2.5608355    15.5 0.33425414  -3.8058910
## 21  -8.5960836     9.0 0.19060773  -7.7830270
## 22  -3.6665796    13.0 0.27900552  -5.2066894
## 23   4.3334204    31.0 0.67679558   4.0775194
## 24  -0.5960836    22.0 0.47790055  -0.4926145
## 25  -0.7370757    21.0 0.45580110  -0.9867480
## 26   7.3334204    36.5 0.79834254   7.4280027
## 27 -11.4903394     4.0 0.08011050 -12.4819464
## 28  -1.5960836    19.0 0.41160221  -1.9858486
## 29   6.3334204    34.0 0.74309392   5.8032206
## 30   6.3686684    35.0 0.76519337   6.4271285
## 31   3.2981723    29.0 0.63259669   3.0107749
## 32  15.4039164    45.0 0.98618785  19.5769662
## 33  -9.4903394     8.0 0.16850829  -8.5333491
## 34  -1.4903394    20.0 0.43370166  -1.4839527
## 35 -11.4550914     5.0 0.10220994 -11.2796464
## 36  -2.5608355    15.5 0.33425414  -3.8058910
## 37  11.4039164    40.0 0.87569061  10.2544052
## 38  -2.7370757    14.0 0.30110497  -4.6327504
## 39   7.3334204    36.5 0.79834254   7.4280027
## 40  12.5449086    43.0 0.94198895  13.9695002
```

```
## 41  -3.7370757    12.0 0.25690608  -5.8032206
## 42   4.5096606    32.0 0.69889503   4.6327504
## 43  -2.4903394    17.0 0.36740331  -3.0107749
## 44   1.4391645    25.0 0.54419890   0.9867480
## 45   2.4039164    27.0 0.58839779   1.9858486
```

```r
# show in a plot
plot(exp_rank, ei)
```




```r
# short way to do the same as above and plot
par(mfrow = c(2, 2))
plot(f_1_20)
```




```r
# getting correlation information
cor.test(exp_rank, ei)
```

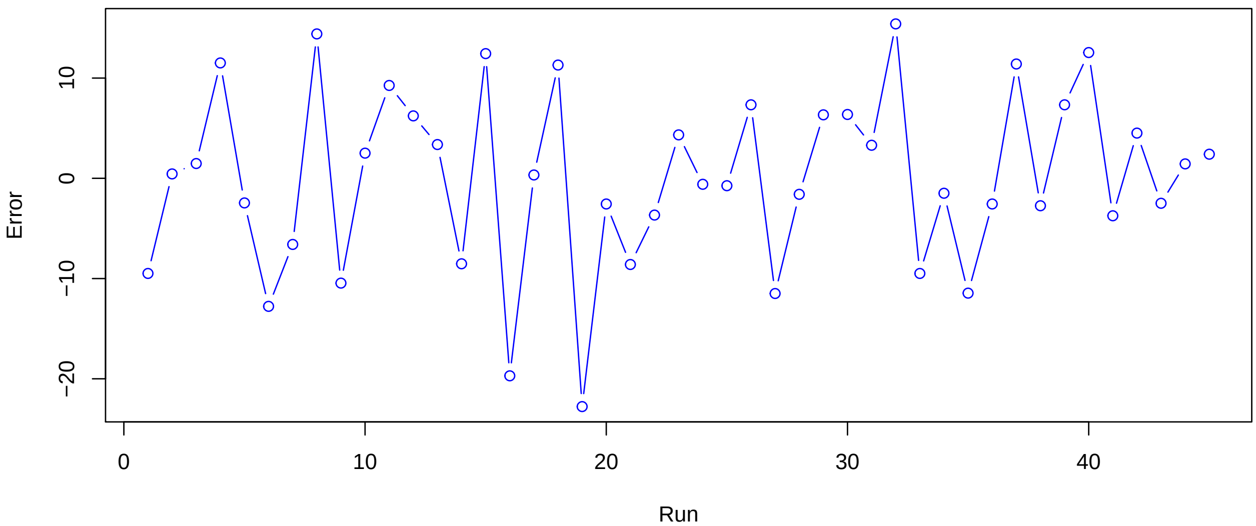
```
##
##  Pearson's product-moment correlation
##
## data:  exp_rank and ei
```

```
## t = 44.176, df = 43, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9802438 0.9940660
## sample estimates:
##       cor
## 0.9891615
```

We see here the distribution is normal with no outliers. We also reject the null as it is normal.

f. Prepare a time plot of the residuals to ascertain whether the error terms are correlated over time. What is your conclusion?

```
plot(ei, type = "b", col = "blue", xlab = "Run", ylab = "Error")
```



We see no correlation with time.

g. Assume that (3.10) is applicable and conduct the Breusch-Pagan test to determine whether or not the error variance varies with the level of X. Use a = .05. State the alternatives,decision rule, and conclusion.

```
ei2 = ei^2
f = lm(ei2 ~ xi)
summary(f)
```

```
##
## Call:
## lm(formula = ei2 ~ xi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -101.32  -69.40  -41.29   54.59  410.04
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   41.818     32.732   1.278    0.208
## xi             6.672      5.639   1.183    0.243
##
## Residual standard error: 104.1 on 43 degrees of freedom
## Multiple R-squared:  0.03153,    Adjusted R-squared:  0.009004
```

```
## F-statistic:   1.4 on 1 and 43 DF,  p-value: 0.2433
```

```
# to find SSE(R) and SSR(R)
anova(lm(ei2 ~ xi))
```

```
## Analysis of Variance Table
##
## Response: ei2
##           Df Sum Sq Mean Sq F value Pr(>F)
## xi         1  15155   15155  1.3998 0.2433
## Residuals 43 465556   10827
```

```
# to find SSE(F) and SSR(F)
anova(f_1_20)
```

```
## Analysis of Variance Table
##
## Response: V1
##           Df Sum Sq Mean Sq F value    Pr(>F)
## V2         1  76960   76960  968.66 < 2.2e-16 ***
## Residuals 43   3416      79
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# chi-squared: [SSR(R)/2] / [SSE/n]^2
chi = (15155/2)/((3416/45))^2
print(chi)
```

```
## [1] 1.314968
```

```
# p
p = 1 - pchisq(1.314968, 2, 45)
print(p)
```

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

SSR(R) = 15155 SSE(R) = 46556 df = 43

SSR(F) = 76960 SSE(F)= 3416 df= 43

After all of our above we would see that we will accept our null as the error variance is constant.

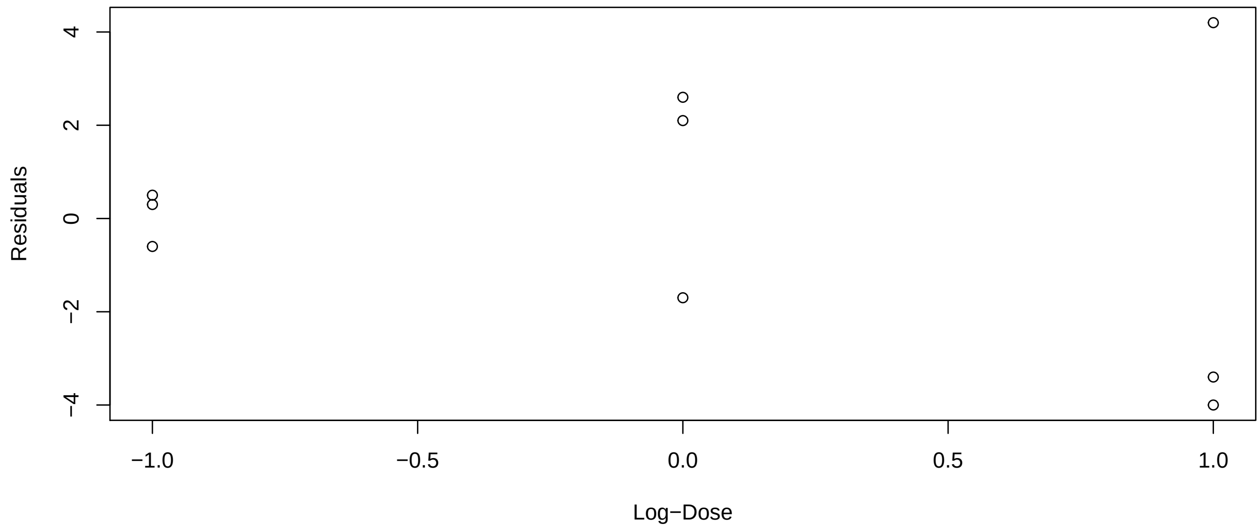
(Textbook 3.11) Drug concentration.

A pharmacologist employed linear regression model (2.1) to study the relation between the concentration of a drug in plasma (Y) and the log-dose of the drug (X).The residuals and log-dose levels follow.

*Please use dataset titled: **CH03PR11.txt**

a. Plot the residuals ei against Xi. What conclusions do you draw from the plot?

```
Dataset_3_11 = read.table("CH03PR11.txt", header = FALSE, sep = "", col.names = c("V1",
    "V2"))


plot(Dataset_3_11$V1, Dataset_3_11$V2, xlab = "Log-Dose", ylab = "Residuals")
```



Part A Conclusion: We see a non-constant variance here.

b. Assume that (3.10) is applicable and conduct the Breusch-Pagan test to determine whether or not the error variance varies with log-dose of the drug (X). Use a = .05. State the alternatives, decision rule, and conclusion. Does your conclusion support your preliminary findings in part (a)?

```
lmfit3_11 = lm(Dataset_3_11$V2^2 ~ Dataset_3_11$V1)
summary(lmfit3_11)
```

```
##
## Call:
## lm(formula = Dataset_3_11$V2^2 ~ Dataset_3_11$V1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7722 -2.2522  0.8444  1.1144  3.5611
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.6622     0.8451   7.883 0.000100 ***
## Dataset_3_11$V1   7.4167     1.0350   7.166 0.000183 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.535 on 7 degrees of freedom
## Multiple R-squared:   0.88,  Adjusted R-squared:  0.8629
## F-statistic: 51.35 on 1 and 7 DF,  p-value: 0.0001828
```

```
anova(lmfit3_11)
```

```
## Analysis of Variance Table
##
## Response: Dataset_3_11$V2^2
##                 Df Sum Sq Mean Sq F value    Pr(>F)
## Dataset_3_11$V1  1 330.04  330.04  51.348 0.0001828 ***
## Residuals        7  44.99    6.43
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
sse = sum(Dataset_3_11$V2^2)
print(sse)
```

```
## [1] 59.96
```

```r
# figure out chisquared
chisq = qchisq(0.95, 1)
print(chisq)
```

```
## [1] 3.841459
```

```r
# test stat:
x2 = (330.04/2)/(59.96/9)^2
print(x2)
```

```
## [1] 3.717906
```

Part B Notes: Ho: Error Variance is constant Ha: Error Variance is not constant

SSR = 330.04 (from summary above) SSE = 59.96 (from sse calculation above)

test statistic: (from calculation above) X^2 = (330.04/2) / (59.96/9)^2 = 3.717906

Chi-squared: 3.84

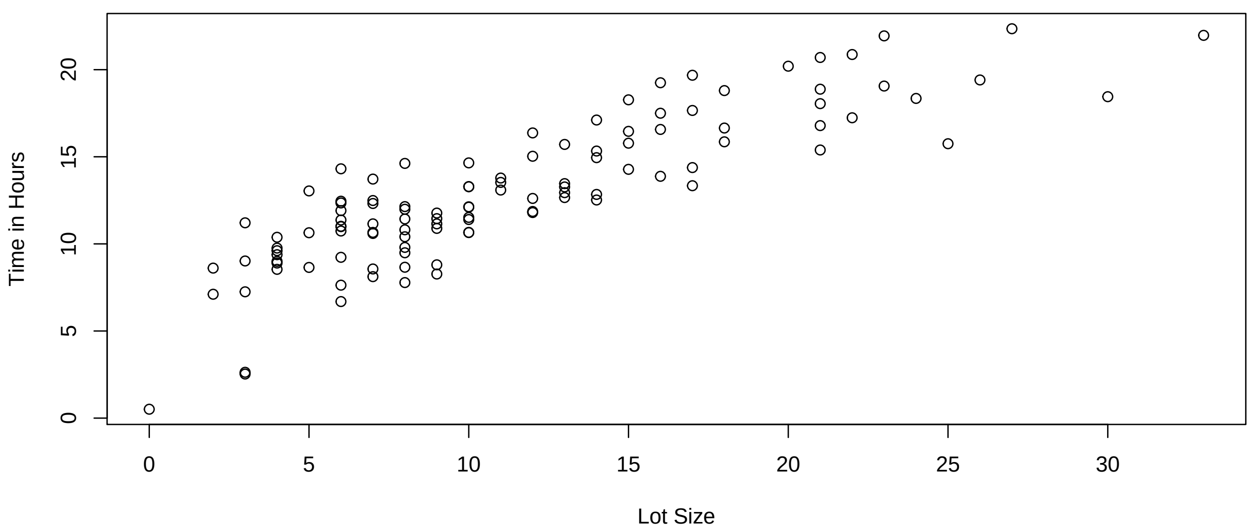
Part B Conclusion: If X^2 <= 3.84 conclude error variance constant, otherwise error variance not constant. So we conclude error variance not constant.

(Textbook 3.18)Production time. In a manufacturing study, the production times for 111 recent production runs were obtained. The table below lists for each run the production time in hours (Y) and the production lot size (X).

*Please use dataset titled: **CH03PR18.txt**

a. Prepare a scatter plot of the data. Does a linear relation appear adequate here? Would a transformation on X or Y be more appropriate here? Why?

```r
Dataset_3_18 = read.table("CH03PR18.txt", header = FALSE, sep = "", col.names = c("V1",
    "V2"))

plot(Dataset_3_18$V2, Dataset_3_18$V1, xlab = "Lot Size", ylab = "Time in Hours")
```



Part A Conclusion: A transformation to X is more suitable because the different levels seem to be pretty consistent.

b. Use the transformation X' = -JX and obtain the estimated linear regression function for the transformed data.

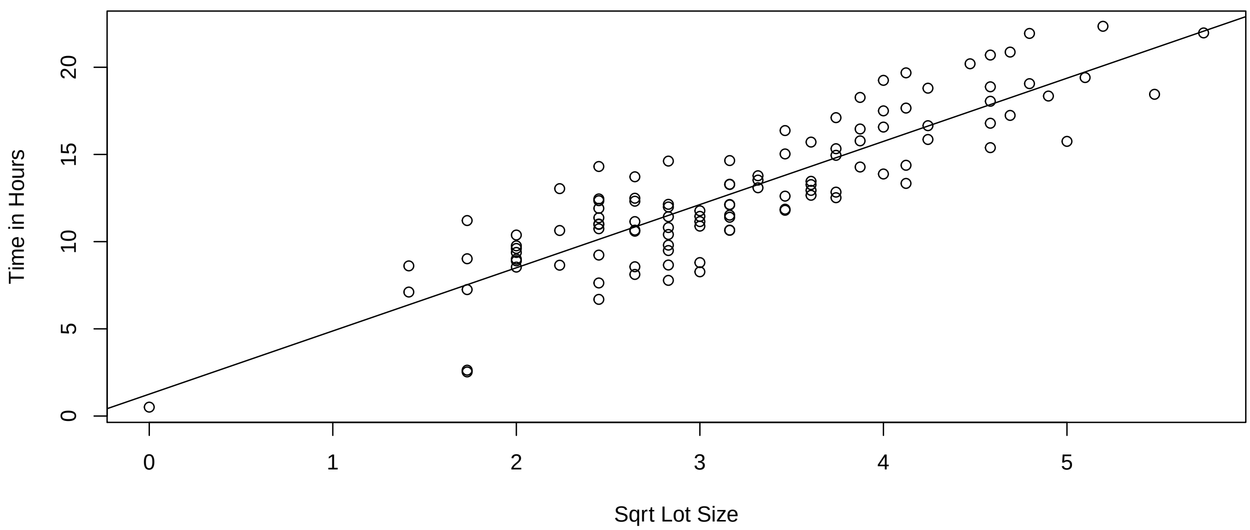
```
lmfit3_18 = lm(V1 ~ sqrt(V2), data = Dataset_3_18)
summary(lmfit3_18)
```

```
##
## Call:
## lm(formula = V1 ~ sqrt(V2), data = Dataset_3_18)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0008 -1.2161  0.0383  1.3367  4.1795
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.2547     0.6389   1.964   0.0521 .
## sqrt(V2)      3.6235     0.1895  19.124   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.99 on 109 degrees of freedom
## Multiple R-squared:  0.7704, Adjusted R-squared:  0.7683
## F-statistic: 365.7 on 1 and 109 DF,  p-value: < 2.2e-16
```

Part B Conclusion: The linear regression function is: y' = 1.2547-3.6235x'

c. Plot the estimated regression line and the transformed data. Does the regression line appear to be a good fit to the transformed data?

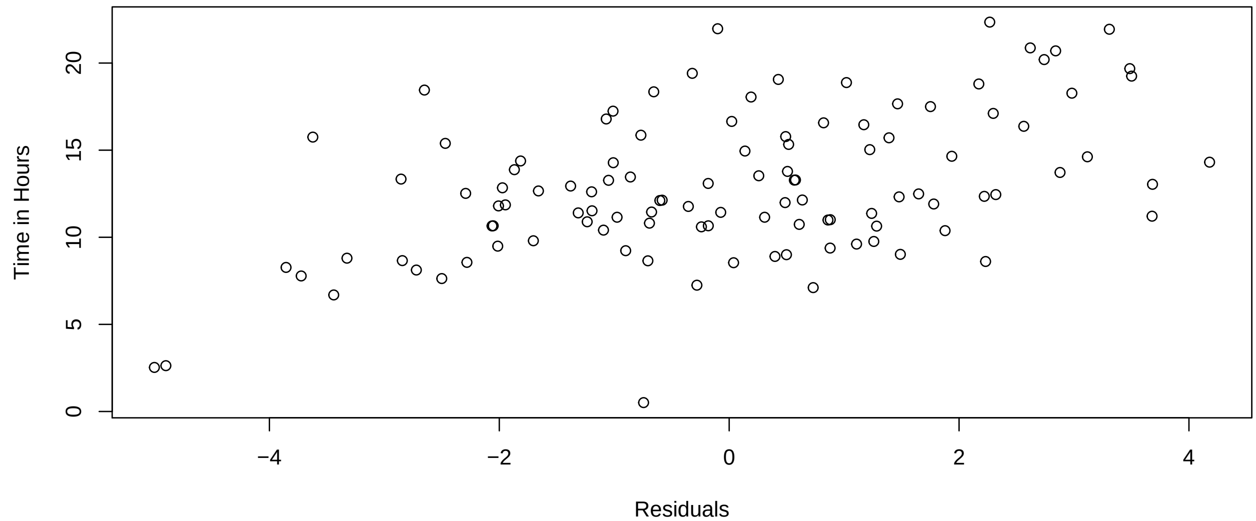
```
plot(sqrt(Dataset_3_18$V2), Dataset_3_18$V1, xlab = "Sqrt Lot Size", ylab = "Time in Hours")
abline(lmfit3_18)
```



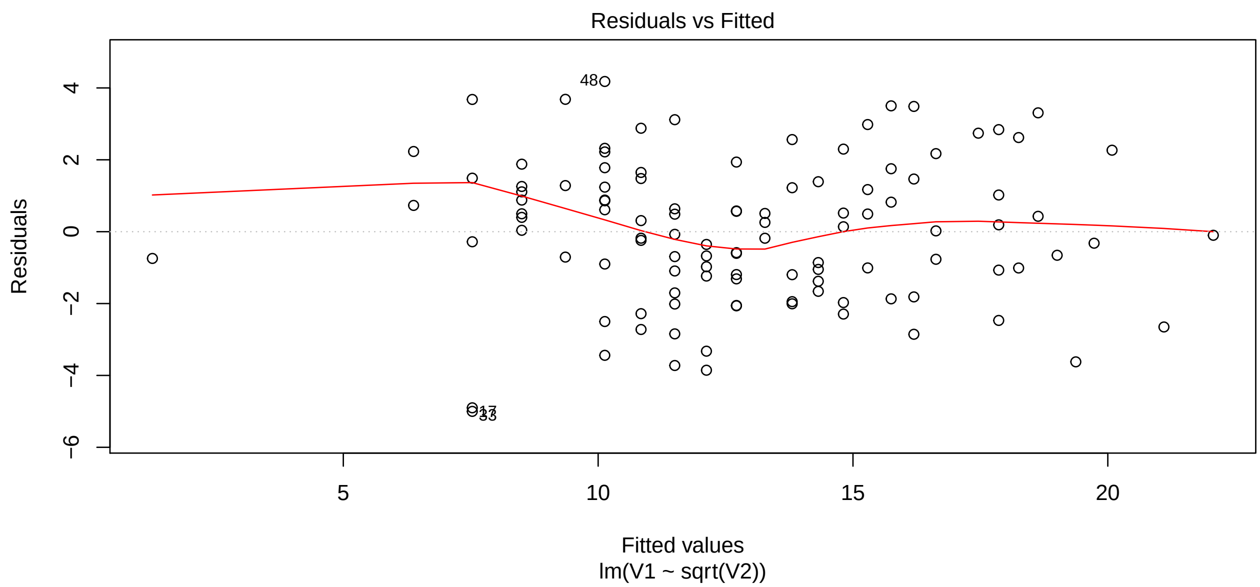
Part C Conclusion: Yes the linear regression appears to be a good fit.

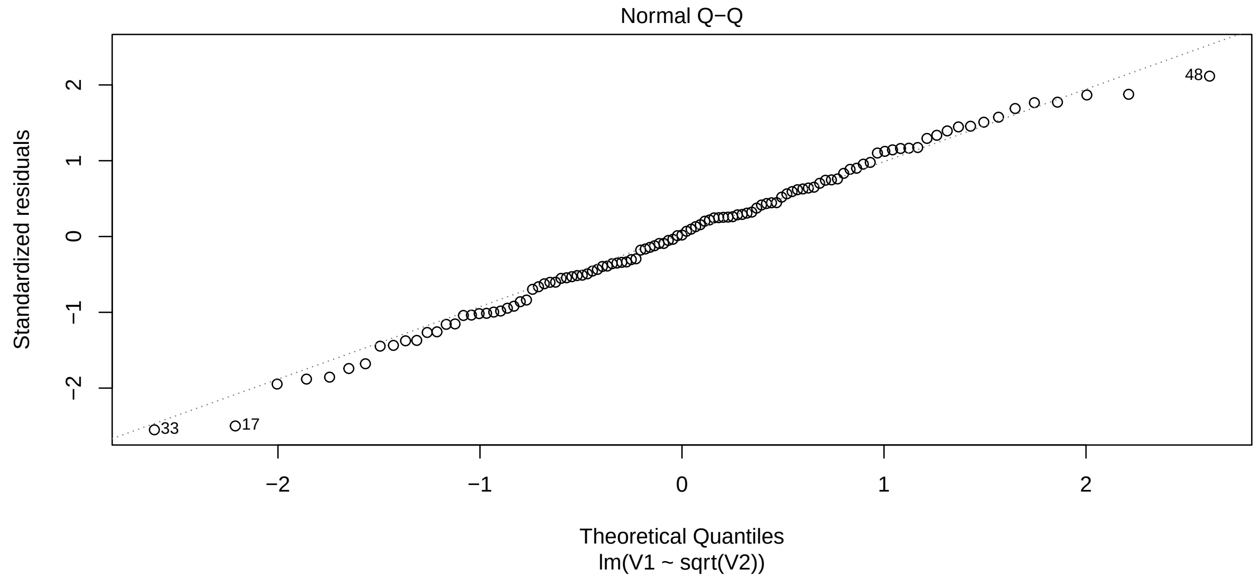
d. Obtain the residuals and plot them against the fitted values. Also prepare a normal probability plot. What do your plots show?

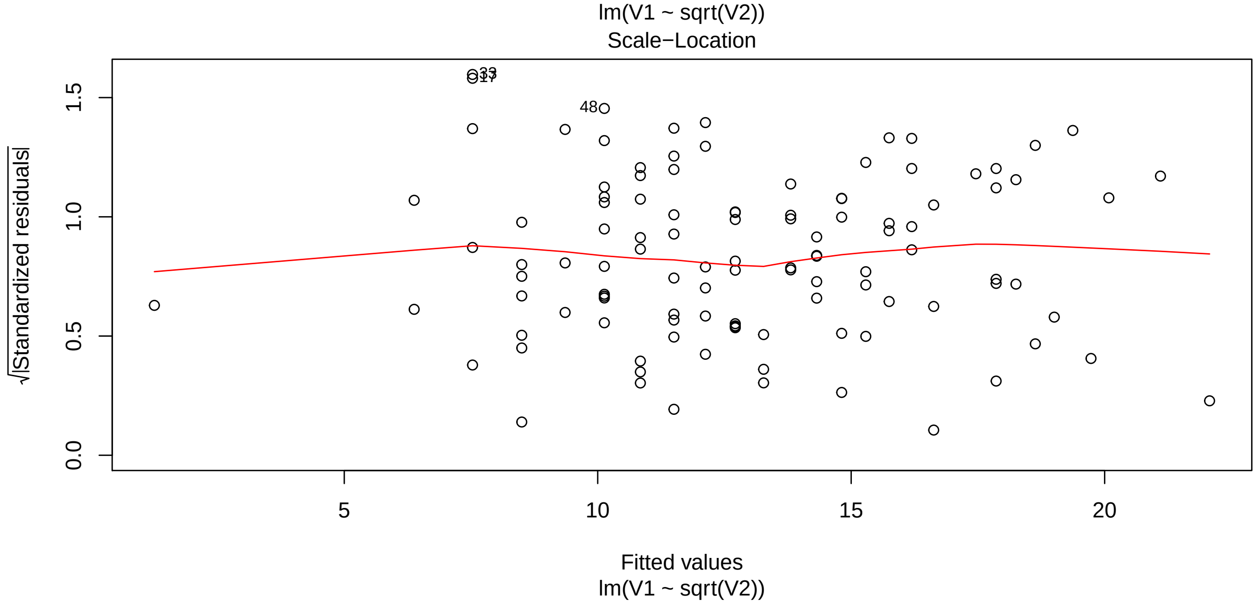
```r
plot(lmfit3_18$residuals, Dataset_3_18$V1, xlab = "Residuals", ylab = "Time in Hours")
```

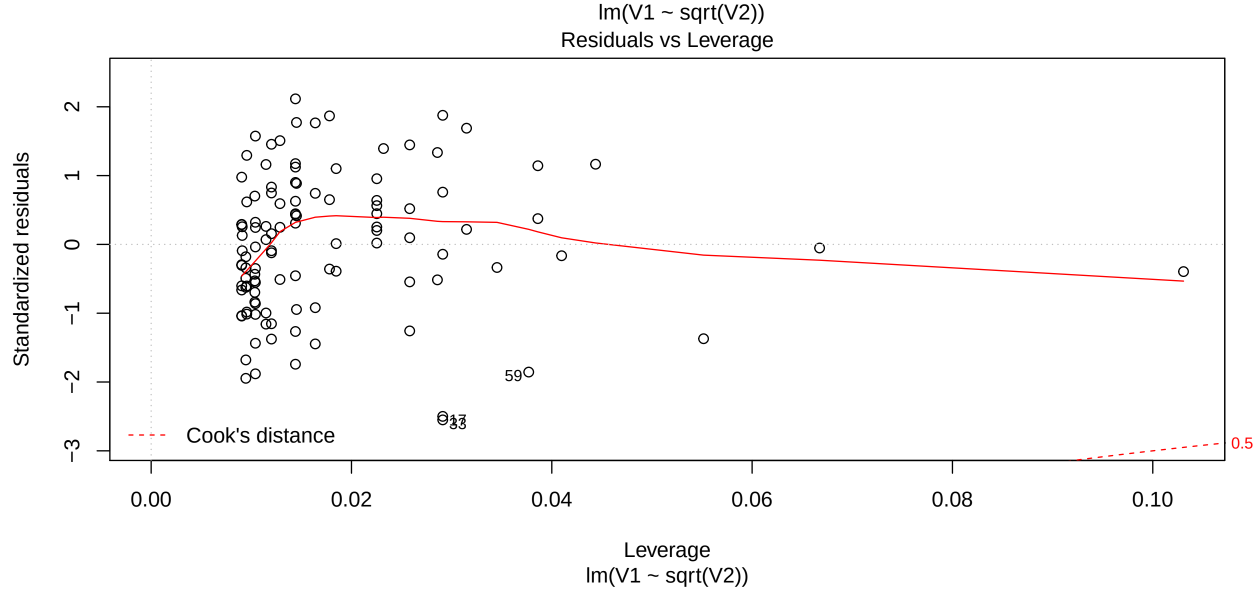



```r
plot(lmfit3_18)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(V1 ~ sqrt(V2))
48
33
17


Scale−Location
√|Standardized residuals|
Fitted values
lm(V1 ~ sqrt(V2))
33
17
48


Residuals vs Leverage
Standardized residuals
Leverage
lm(V1 ~ sqrt(V2))
Cook's distance
0.5
59
17
33

Part D Conclusion: The residuals plot (the first plot) shows that points are spread out without a systematic pattern. The normal probability plot shows that the points all fall pretty close to a straight line.

e. Express the estimated regression function in the original units.

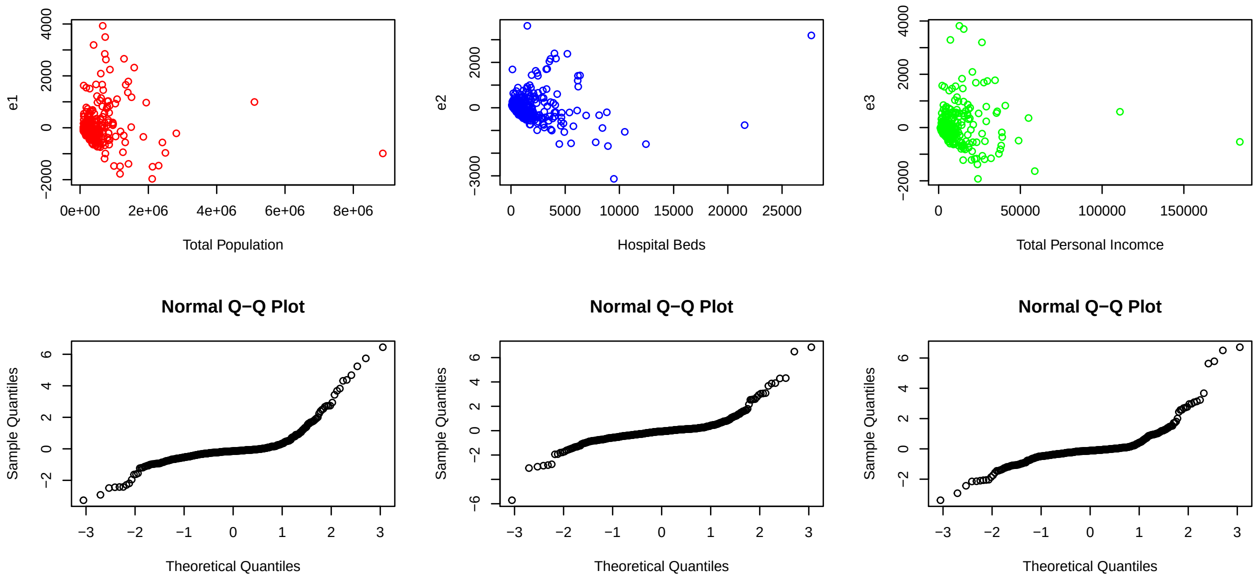
Part E conclusion: The estimated regression line expressed in original units is: yhat = 1.2547-3.6235*sqrt(x)

(Textbook 3.25)

Refer to the CDI data set in Appendix C.2 and Project 1.43. For each of the three fitted regression models, obtain the residuals and prepare a residual plot against X and a normal probability plot. Summarize your conclusions. Is linear regression model (2.1) more appropriate in one case than in the others?

Please use dataset titled: ***APPENC02.txt***

```
df_cdi = read.table("APPENC02.txt", header = FALSE, sep = "", col.names = c("id",
    "county", "state", "landArea", "totPop", "percAge18_34", "percAge65plus", "actPhysicians",
    "hospBeds", "totSerCrimes", "percHSgrads", "percBachDeg", "percBelowPov", "percUnempl",
    "perCapitaInc", "totPersIncome", "geoRegion"))


f_3.25_1 = lm(df_cdi$actPhysicians ~ df_cdi$totPop)
f_3.25_2 = lm(df_cdi$actPhysicians ~ df_cdi$hospBeds)
f_3.25_3 = lm(df_cdi$actPhysicians ~ df_cdi$totPersIncome)

e1 = f_3.25_1$residuals
e2 = f_3.25_2$residuals
e3 = f_3.25_3$residuals

# standardize residuals needed for QQ plot##
rs1 = rstandard(f_3.25_1)
rs2 = rstandard(f_3.25_2)
rs3 = rstandard(f_3.25_3)

par(mfrow = c(2, 3))

plot(df_cdi$totPop, e1, xlab = "Total Population", col = "red")
plot(df_cdi$hospBeds, e2, xlab = "Hospital Beds", col = "blue")
plot(df_cdi$totPersIncome, e3, xlab = "Total Personal Incomce", col = "green")

## QQ plot ##
qqnorm(rs1)
qqnorm(rs2)
qqnorm(rs3)
```



Problem 3.25 Conclusion: There does not seem to be one that is more visually significant than the others.

(Textbook 3.32)

Refer to the Prostate cancer data set in Appendix C.5. Build a regression model to predict PSA level (Y) as a function of cancer, Volume (X). The analysis should include an assessment of the degree to which the key regression assumptions are satisfied. If the regression assumptions are not met, include and justify appropriate remedial measures. Use the final model to estimate mean PSA level for a patient whose cancer volume is 20 cc. Assess the strengths and weaknesses of the final model.

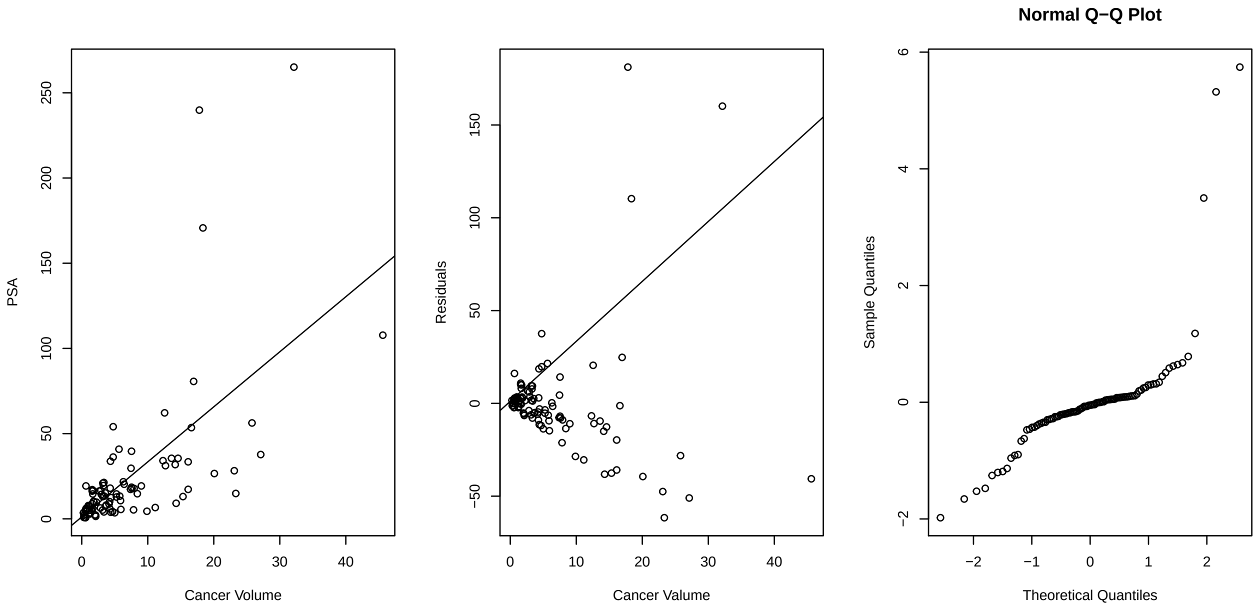
Please use dataset titled: ***APPENC05.txt***

```
df_prostate = read.table("APPENC05.txt", header = FALSE, sep = "", col.names = c("id",
    "psa", "cancerVol", "weight", "age", "benPros", "seminalVes", "capPen", "gleasonScore"))


f_3.32 = lm(df_prostate$psa ~ df_prostate$cancerVol)
summary(f_3.32)
```

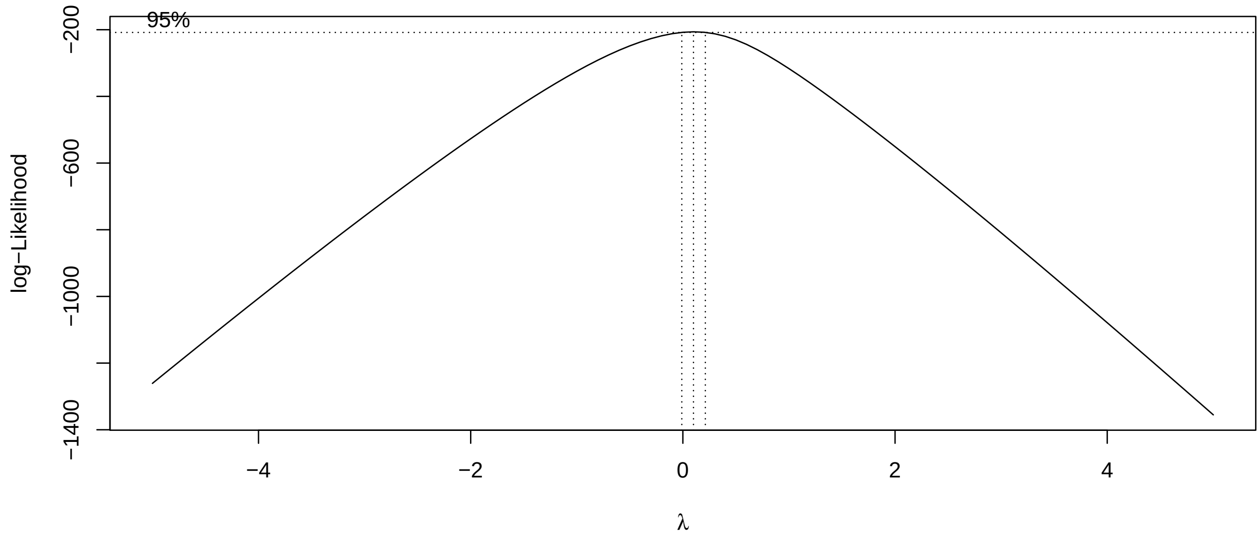
```
##
## Call:
## lm(formula = df_prostate$psa ~ df_prostate$cancerVol)
##
## Residuals:
##     Min       1Q  Median      3Q     Max
## -61.619  -9.023  -1.586   3.151 181.183
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.1249     4.3596   0.258    0.797
## df_prostate$cancerVol   3.2299     0.4148   7.786 8.47e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.03 on 95 degrees of freedom
## Multiple R-squared:  0.3896, Adjusted R-squared:  0.3831
```

```
## F-statistic: 60.63 on 1 and 95 DF,  p-value: 8.468e-12
```

```
e1 = f_3.32$residuals

# standardize residuals needed for QQ plot##
rs1 = rstandard(f_3.32)

par(mfrow = c(1, 3))

plot(df_prostate$cancerVol, df_prostate$psa, xlab = "Cancer Volume", ylab = "PSA")
abline(f_3.32)

plot(df_prostate$cancerVol, e1, xlab = "Cancer Valume", ylab = "Residuals")
abline(f_3.32)

## QQ plot ##
qqnorm(rs1)
```




```
par(mfrow = c(1, 1))

boxcox(f_3.32, lambda = seq(-5, 5, by = 0.1))
```

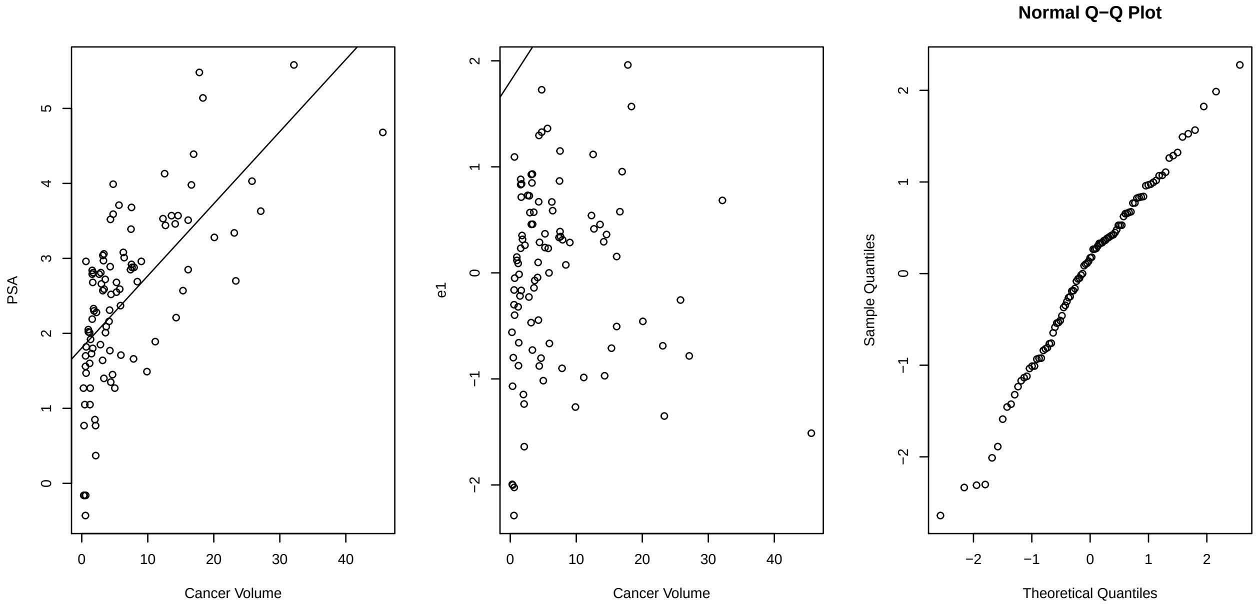



```r
# From our boxcox we see that we need to use the log transformation

f_3.32_2 = lm(log(df_prostate$psa) ~ df_prostate$cancerVol)
summary(f_3.32_2)
```

```
##
## Call:
## lm(formula = log(df_prostate$psa) ~ df_prostate$cancerVol)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2886 -0.6590  0.1493  0.5769  1.9610
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.80549    0.11899  15.174  < 2e-16 ***
## df_prostate$cancerVol  0.09619    0.01132   8.496 2.69e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8742 on 95 degrees of freedom
## Multiple R-squared:  0.4317, Adjusted R-squared:  0.4258
## F-statistic: 72.18 on 1 and 95 DF,  p-value: 2.688e-13
```

```r
e1 = f_3.32_2$residuals

# standardize residuals needed for QQ plot##
rs1 = rstandard(f_3.32_2)

par(mfrow = c(1, 3))

plot(df_prostate$cancerVol, log(df_prostate$psa), xlab = "Cancer Volume", ylab = "PSA")

abline(f_3.32_2)

plot(df_prostate$cancerVol, e1, xlab = "Cancer Volume")
```

```
abline(f_3.32_2)

qqnorm(rs1)
```



Problem 3.32: It still seems like we were getting a lot of outliers in our 2 graphs. Our normal plot is in line but not still not completely a great line.