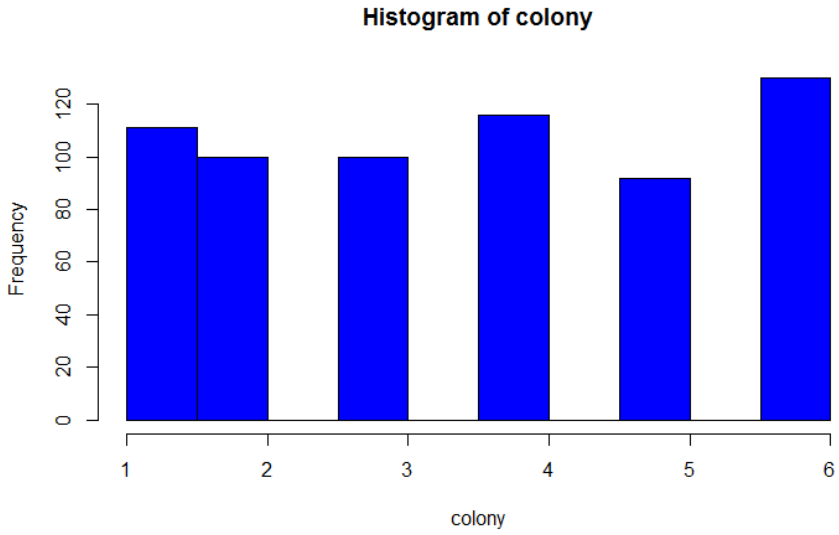
Q6

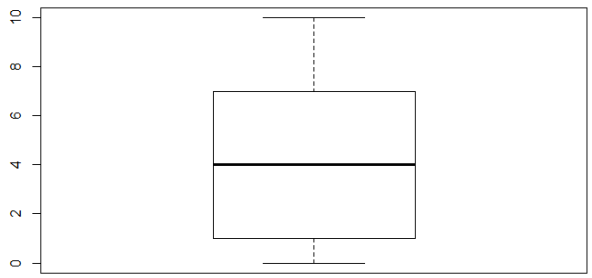
a)
We first examine the data. We import the data with codes in appendix 1.1 .

Then, we draw a histogram to examine the number of ants in each colony:
(appendix 1.2)



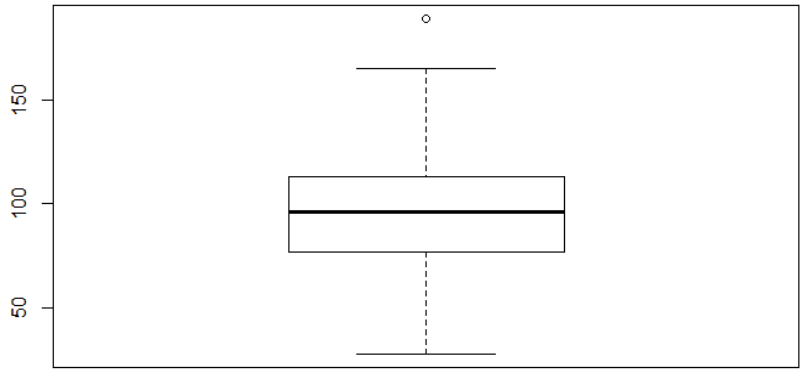
From the plot, we can see that the number of ants investigated in each colony are close to each other.

Then we examine the distribution of the distance by plotting a box plot:
(appendix 1.3)



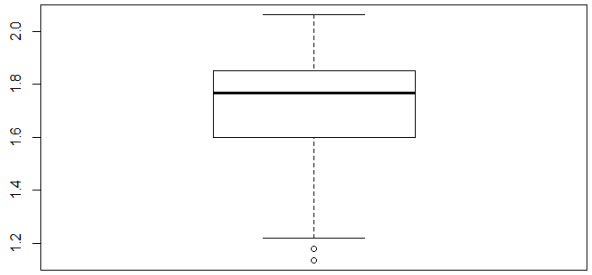
From the plot, we know that the median of distance is about 4. Half of the ants search for food within the distance 4 while some ants go very far in search for food.

We then use the same approach to examine the mass of ants:
(appendix 1.4)



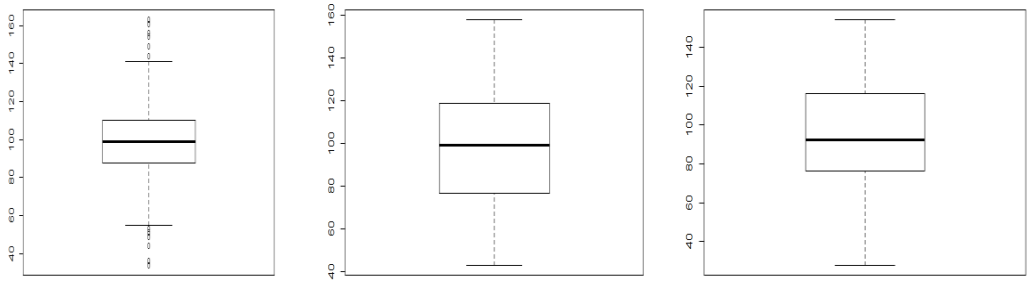
We can see that most ants weighs around 100.

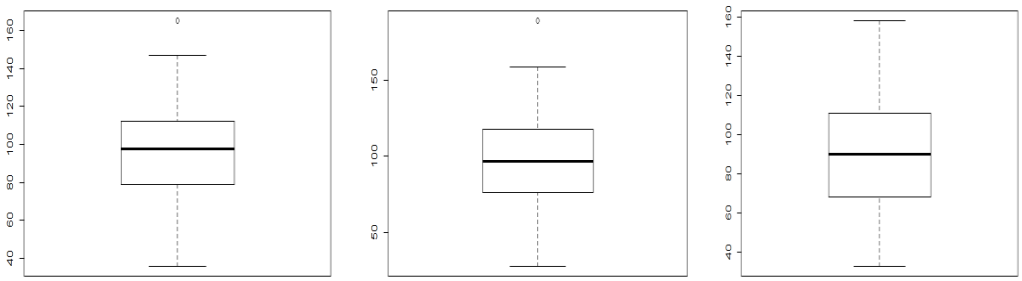
We then examine the headwidth of ants
(appendix 1.5)



The plot shows that nearly half of the ants have headwidth wider than 1.8 .

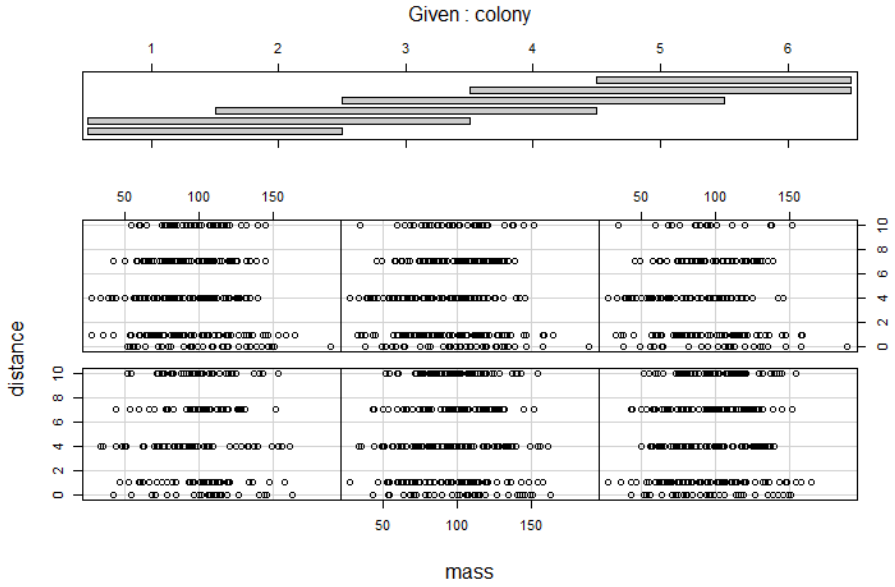
We are also interested in the distribution of mass within each colony. Thus, with codes in appendix 1.6, we can draw boxplots for the mass distribution within each colony:





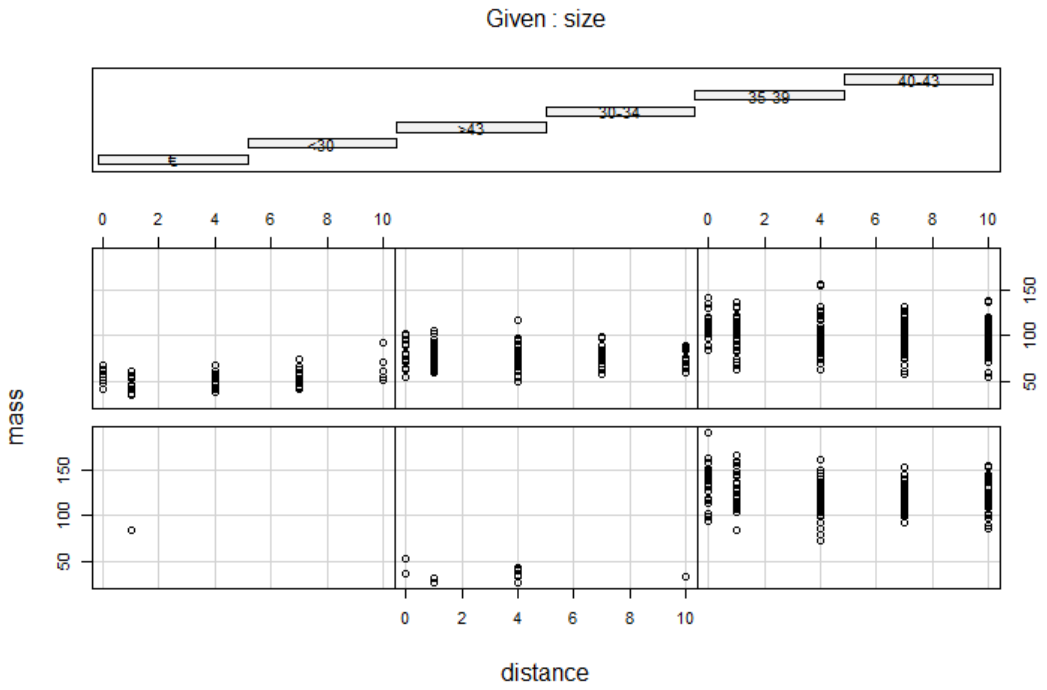
We can see different colonies have slight differences in their mass.

We further study given the colony, if there is some relationship between mass and distance. We use codes in appendix 1.7 to draw a coplot.



It seems that among different colonies, the relationship between distance and mass looks alike with each other.

We then study that given size, if there is any relationship between distance and mass. We use codes in appendix 1.8 to draw a coplot:



The plot shows that only few ants are of small size. In other occasions, it seems that the distance an ants goes has nothing to do on its mass. For ants with different distances have nearly the same distribution.

At the same time, we notice that there is a strange data with a strange size. Further examination shows that many of its values are NA. Since this observation provides poor information, we simply delete it.

To further clarify the data, we will run an linear regression and perform regression diagnostics.

b)

We first perform the regression with codes in appendix 2.1 and get the following summary:

```
Call:
lm(formula = mass ~ colony + distance + size)

Residuals:
    Min      1Q  Median      3Q     Max
-51.077 -10.000   0.008   8.626  63.319

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  40.9588     4.5313   9.039  < 2e-16 ***
colony       -0.4368     0.3468  -1.260 0.208302
distance     -0.7287     0.1780  -4.094 4.78e-05 ***
size>43      86.9063     4.4233  19.647  < 2e-16 ***
size30-34    17.9485     4.7288   3.796 0.000161 ***
size35-39    40.4844     4.4373   9.124  < 2e-16 ***
size40-43    64.1592     4.3725  14.673  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.32 on 641 degrees of freedom
Multiple R-squared:  0.6823,    Adjusted R-squared:  0.6793
F-statistic: 229.4 on 6 and 641 DF,  p-value: < 2.2e-16
```

After performing the regression, it is necessary to perform regression diagnostic.

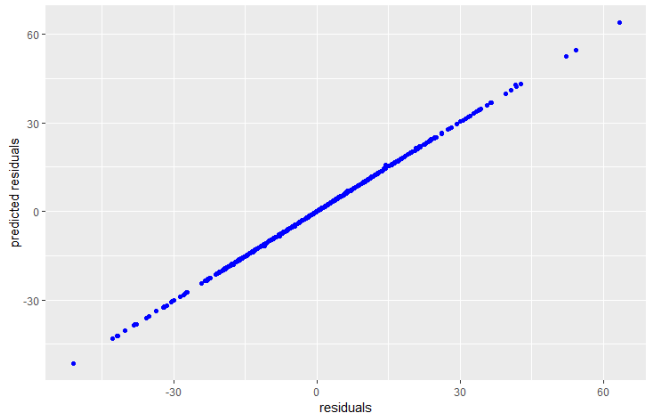
I)

We first determine if there are influential observations. From the summary of the regression, we know there are 7 coefficients and 649 observations. Thus, the average of the leverages is $(7+1)/649 \approx 0.012$. We list all leverages (appendix 2.2), here is some of the leverages:

```
         19          62         364         255         259         264         267         272          80
0.012211704 0.012384420 0.012471509 0.012661136 0.012661136 0.012661136 0.012661136 0.012661136 0.012852344
         84          95          96          98         499         509         477         479         482
0.012852344 0.012852344 0.012852344 0.012852344 0.013127689 0.013127689 0.013708236 0.013708236 0.013708236
        488         496         497         173         183         485         486         489         491
0.013708236 0.013708236 0.013708236 0.013875598 0.013875598 0.015166760 0.015166760 0.015166760 0.015166760
        493          83          91          99         329         224         225         599         601
0.015166760 0.016114896 0.016114896 0.016114896 0.018206143 0.018495475 0.018495475 0.018941646 0.018941646
        613         622         289         293         338         343         232         235         236
0.018941646 0.018941646 0.019417037 0.019417037 0.019424661 0.019424661 0.019629038 0.019629038 0.019629038
        242         189         208         534         539         421         422         427         428
0.019629038 0.019791325 0.019791325 0.020360538 0.020360538 0.020701984 0.020701984 0.020701984 0.020701984
        430         434         440         445         532         153         154         158         163
0.020701984 0.020701984 0.020701984 0.020701984 0.020763144 0.020858249 0.020858249 0.020858249 0.020858249
        561         113         118         390         397         400         412         451         452
0.020982767 0.021190448 0.021190448 0.021742967 0.021742967 0.021742967 0.021742967 0.022090414 0.022090414
        471          40          45         515         550         262         269         273           4
0.022090414 0.022148645 0.022148645 0.022190585 0.022629831 0.023192013 0.023192013 0.023192013 0.023614405
         13         181         503         507          81         598         602         607         207
0.023614405 0.024336269 0.024642862 0.024642862 0.026505360 0.077406865 0.077406865 0.077406865 0.078301788
        441         533         387         393         549          25          38          42         490
0.078810412 0.079050828 0.079183058 0.079183058 0.079847143 0.082041025 0.082041025 0.082041025 0.085353356
> |
```

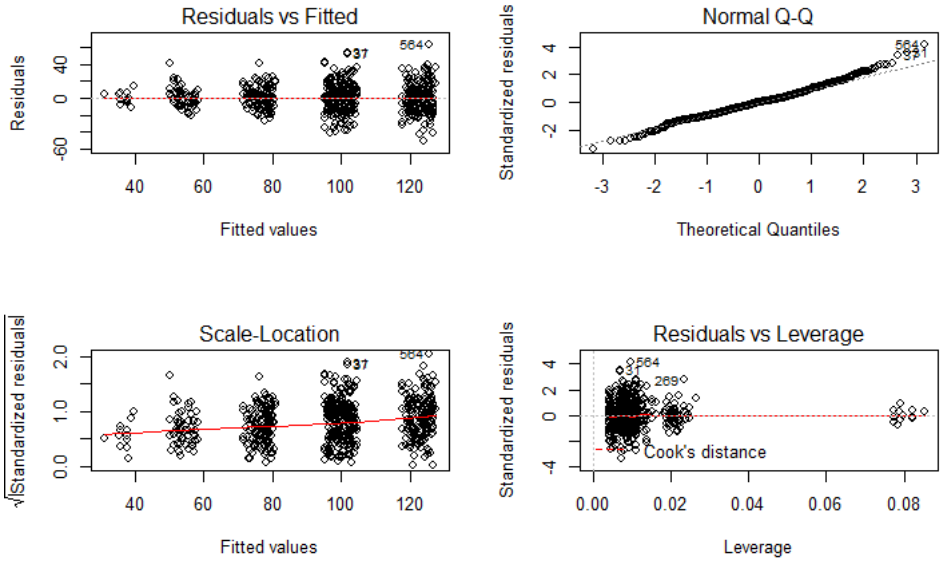
We can find out that some of the leverages are pretty high. Thus, it is suspected that their might be influential observations.

We then plot residuals vs predicted residuals (appendix 2.3) :



We can find out that all residuals are close to there predicted residuals. Thus, it is reasonable to conclude that there is no outliers on x-axis.

However, we do suspect that there ming be outlying Y. To verify this, we further run regression diagnostics with appdendix 2.4 .



From these plots, we notice that observation 564 and 37 have standardized residuals larger than 3. While from appendix 2.5, we find they have rather small leverages. Thus, we can believe that observation 564 and 37 are outliers on Y-axis. We shall remove them and regress again (appendix 2.6):

```
Call:
lm(formula = mass ~ colony + distance + size)

Residuals:
    Min      1Q  Median      3Q     Max
-50.642  -9.936   0.321   8.725  54.530

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  40.7290     4.4357   9.182  < 2e-16 ***
colony       -0.4174     0.3401  -1.227 0.220165
distance     -0.6814     0.1744  -3.906 0.000104 ***
size>43      86.4732     4.3305  19.968  < 2e-16 ***
size30-34    17.9148     4.6281   3.871 0.000120 ***
size35-39    40.4693     4.3428   9.319  < 2e-16 ***
size40-43    63.8840     4.2797  14.927  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.99 on 639 degrees of freedom
Multiple R-squared:  0.6886,    Adjusted R-squared:  0.6857
F-statistic: 235.5 on 6 and 639 DF,  p-value: < 2.2e-16
```

II)

The second thing we would like to do is to detect nonlinearity.

To detect non-linearity, we do partial regression for the model. If the linearity holds, each partial regression shall possess a very small intercept and a slope that is extremely close to β_j.

Since there are 6 explanatory variables in the model, we do 6 partial regressions. (appendix 2.7)

And hence we obtain the result of 6 regressions:

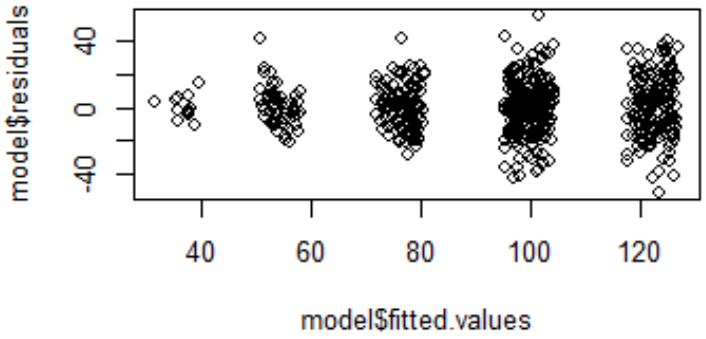
```
+ }
  (Intercept) xlm$residuals
-2.807811e-15 -4.174047e-01
  (Intercept) xlm$residuals
 1.220754e-15 -6.814084e-01
  (Intercept) xlm$residuals
 6.836929e-16  8.647324e+01
  (Intercept) xlm$residuals
-9.210585e-16  1.791480e+01
  (Intercept) xlm$residuals
 1.486874e-15  4.046930e+01
  (Intercept) xlm$residuals
-1.655048e-15  6.388402e+01
>
```

We can see that all these 6 models have very small intercept and a slope that is close to each's β_j. Thus, I believe that the true data is linear.

III)

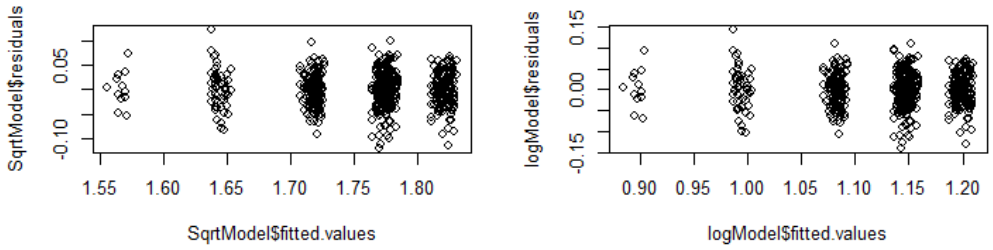
The third thing to do is to check the assumptions on e. To do this, we plot residuals vs fitted values (appendix 2.8).

The following picture is the result:



The picture shows that in the original data, the data does not have the same variance. Actually, as the fitted values grows bigger, their variance grows larger. Thus, some transformation will be performed on the data.

Because we know little about mass and σ, we shall try both $\mathrm{h}(\mathrm{mass}) = \sqrt{mass}$ and $h(mass) = \log(mass)$. We transform mass in these two ways and get the result (appendix 2.10):



What is surprising is that these two models has nearly the same effect. Both of the variances after transformation is nearly a constant. We then further investigate the summary of these two models:

```
-0.120383 -0.021756  0.002993  0.022066  0.122092

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.5764506  0.0103311 152.593  < 2e-16 ***
colony      -0.0013013  0.0007921  -1.643  0.10091
distance    -0.0012928  0.0004063  -3.182  0.00153 **
size>43      0.2560160  0.0100861  25.383  < 2e-16 ***
size30-34    0.0781297  0.0107791   7.248 1.22e-12 ***
size35-39    0.1524947  0.0101147  15.076  < 2e-16 ***
size40-43    0.2096961  0.0099677  21.037  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03492 on 639 degrees of freedom
Multiple R-squared:  0.7362,    Adjusted R-squared:  0.7337
F-statistic: 297.2 on 6 and 639 DF,  p-value: < 2.2e-16

> plot(SqrtModel$fitted.values,SqrtModel$residuals)
> mass=log(mass_copy) #mass_copy is a copy of original mass vector
> logModel=lm(mass~colony+distance+size)
> summary(logModel)

Call:
lm(formula = mass ~ colony + distance + size)

Residuals:
      Min        1Q    Median        3Q       Max
-0.140718 -0.024633  0.003606  0.025289  0.143438

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.9086976  0.0118107  76.939  < 2e-16 ***
colony      -0.0015446  0.0009056  -1.706  0.08855 .
distance    -0.0014165  0.0004645  -3.050  0.00239 **
size>43      0.3027047  0.0115306  26.252  < 2e-16 ***
size30-34    0.0971108  0.0123229   7.880 1.41e-14 ***
size35-39    0.1857919  0.0115634  16.067  < 2e-16 ***
size40-43    0.2511080  0.0113953  22.036  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03992 on 639 degrees of freedom
Multiple R-squared:  0.742,     Adjusted R-squared:  0.7396
F-statistic: 306.3 on 6 and 639 DF,  p-value: < 2.2e-16
```

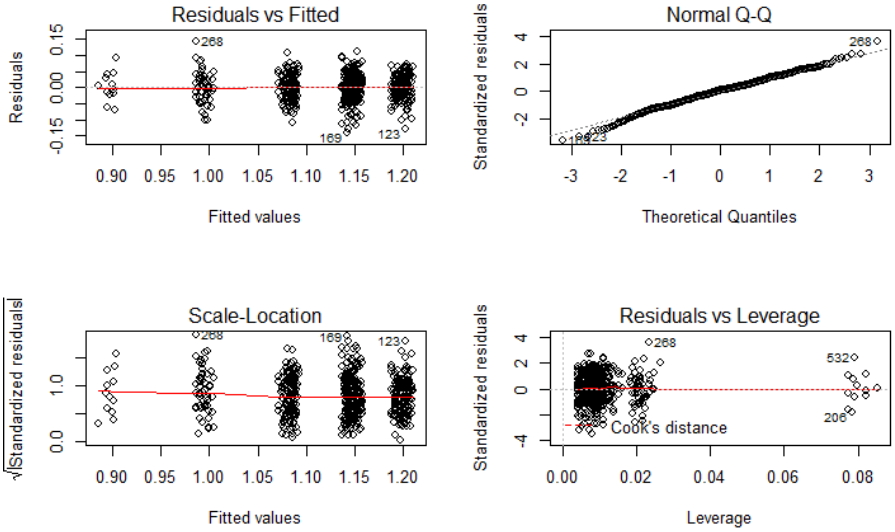
We find out that the log model is more significant.

Besides, the output of appendix 2.11 also shows that its linearity maintains well:

```
+ }
  (Intercept) xlm$residuals
-5.921865e-18 -1.544632e-03
  (Intercept) xlm$residuals
-1.087003e-17 -1.416535e-03
  (Intercept) xlm$residuals
 6.858027e-18  3.027047e-01
  (Intercept) xlm$residuals
 9.381696e-18  9.711076e-02
  (Intercept) xlm$residuals
 2.320000e-18  1.857919e-01
  (Intercept) xlm$residuals
  2.26491e-18   2.51108e-01
>
```

Thus, it is better to transform mass with the log function.

IV)
The last thing we would like to do is to check its normality.
The output of appendix 2.2 is:



Notice that the qq plot is nearly a straight line. Thus, we can treat e as something that is nearly the normal distribution.

c)
The summary of the final model is:

```
Call:
lm(formula = mass ~ colony + distance + size)

Residuals:
     Min       1Q   Median       3Q      Max
-0.56287 -0.09853  0.01442  0.10116  0.57375

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.634790   0.047243  76.939  < 2e-16 ***
colony      -0.006179   0.003622  -1.706  0.08855 .
distance    -0.005666   0.001858  -3.050  0.00239 **
size>43      1.210819   0.046123  26.252  < 2e-16 ***
size30-34    0.388443   0.049292   7.880 1.41e-14 ***
size35-39    0.743167   0.046253  16.067  < 2e-16 ***
size40-43    1.004432   0.045581  22.036  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1597 on 639 degrees of freedom
Multiple R-squared:  0.742,     Adjusted R-squared:  0.7396
F-statistic: 306.3 on 6 and 639 DF,  p-value: < 2.2e-16
```

From the summary, we can see that in this model, the mass of an ant increases with colony, however, it is not that significant. Thus, there is no very significant relationship between the colony and the mass.

Since the coefficient of distance is negative, it is suggested that going further does not help collect more energy. Or, we can say that going further for more food is not a good idea, because the further an ant goes, the less mass it has. Thus, a worker conservative strategy does not work.

However, if there does exist an relationship between the mass and the colony, it is possible that colonies with larger numbers behaves more conservative on going further. This is because the mass goes smaller when number increases. Since we assume all colonies are of the same species, it is possible that colonies with larger number are more conservative thus leads to there smaller mass.

The coefficients of sizes are all positive, this meets the intuition, since size is connected to the mass.

APPENDIX

1.1

```{r}
#import data
ant=read.table("C:/Users/a/Desktop/hw5_ants.txt",header=T)
colony=ant$Colony
distance=ant$Distance
mass=ant$Mass
headwidth=ant$Headwidth
mheadwidth=ant$Headwidth..mm.
size=ant$Size.class
library(ggplot2)
```

1.2

```
hist(colony,col='blue',freq=T)
```

1.3

```
> boxplot(distance)
```

1.4

```
> boxplot(mass)
```

1.5

```
> boxplot(mheadwidth)
>
```

1.6

```
453    {r}
454  col1=NULL
455  col2=col1
456  col3=col1
457  col4=col1
458  col5=col1
459  col6=col1
460
461  for(k in 1:649){
462    if(colony[k]==1){
463      col1=c(mass[k],col1)
464    }
465    if(colony[k]==2){
466      col2=c(mass[k],col2)
467    }
468    if(colony[k]==3){
469      col3=c(mass[k],col3)
470    }
471    if(colony[k]==4){
472      col4=c(mass[k],col4)
473    }
474    if(colony[k]==5){
475      col5=c(mass[k],col5)
476    }
477    if(colony[k]==6){
478      col6=c(mass[k],col6)
479    }
480  }
481
482  boxplot(col1)
483  boxplot(col2)
484  boxplot)(col3)
485  boxplot(col4)
486  boxplot(col5)
487  boxplot(col6)
488
```

1.7

coplot(distance~mass|colony)

1.8

coplot(mass~distance|size)

2.1

```
model=lm(mass~colony+distance+size)
summary(model)
```

2.2

```
model=lm(mass~colony+distance+size)
summary(model)
```

2.3

```
predicted=model$residuals/(1-hatvalues(model))

data=data.frame(model$residuals,predicted)

ggplot(data,aes(model$residuals,predicted))+geom_point(colo
r="blue")+labs(x="residuals",y="predicted residuals")
```

2.4

```
par(mfrow=c(2,2))
plot(model)
```

2.5

```
hatvalues(model)[37]
hatvalues(model)[564]
```

2.6

```
mass=mass[-564]
colony=colony[-564]
distance=distance[-564]
size=size[-564]

mass=mass[-37]
colony=colony[-37]
distance=distance[-37]
size=size[-37]

mass_copy=mass

model=lm(mass~colony+distance+size)
summary(model)
```

2.7

```
x=model.matrix(model)

k=2
for(k in 2:7){
```

```
  temp=lm(mass~x[,-k])
  xlm=lm(x[,k]~x[,-k])
  s=lm(temp$residuals~xlm$residuals)
  print(s$coefficients)
}
```

2.8

```
plot(model$fitted.values,model$residuals)
```

2.9

```
mass=sqrt(mass_copy) #mass_copy is a copy of original mass
vector
SqrtModel=lm(mass~colony+distance+size)
summary(SqrtModel)
plot(SqrtModel$fitted.values,SqrtModel$residuals)
```

2.10

```
mass=log(mass_copy) #mass_copy is a copy of original mass
vector
logModel=lm(mass~colony+distance+size)
```

```
summary(logModel)
plot(logModel$fitted.values,logModel$residuals)
```

2.11

```
for(k in 2:7){

  temp=lm(mass~x[,-k])
  xlm=lm(x[,k]~x[,-k])
  s=lm(temp$residuals~xlm$residuals)
  print(s$coefficients)
}
```

2.12

```
par(mfrow=c(2,2))
plot(logModel)
```