HW7Math158

Joshua Jansen-Montoya

2022-10-26

Problem 8.1

Researchers at National Institutes of Standards and Technology (NIST) collected pipeline data on ultrasonic measurements of the depth of defects in the Alaska pipeline in the field. The depth of the defects were then remeasured in the laboratory. These measurements were performed in six different batches. It turns out that this batch effect is not significant and so can be ignored in the analysis that follows. The laboratory measurements are more accurate than the in-field measurements, but more time consuming and expensive. We want to develop a regression equation for correcting the in-field measurements. 1. Fit a regression model Lab ~ Field. Check for non-constant variance. 2. We wish to use weights to account for the non-constant variance. Here we split the range of Field into 12 groups of size nine (except for the last group which has only eight values). Within each group, we compute the variance of Lab as varlab and the mean of Field as meanfield. Supposing pipeline is the name of your data frame, the following R code will make the needed computations:

```
library(faraway)
i <- order(pipeline$Field)
npipe <- pipeline[i,]
ff <- gl(12,9)[-108]
meanfield <- unlist(lapply(split(npipe$Field,ff),mean))
varlab <- unlist(lapply(split(npipe$Lab,ff),var))
```

Suppose we guess that the the variance in the response is linked to the predictor in the following way: $var(Lab) = a_0 Field^{a_1}$ Regress log(varlab) on log(meanfield) to estimate a_0 and a_1 . (You might choose to remove the last point.) Use this to determine appropriate weights in a WLS fit of Lab on Field. Show the regression summary. 3. An alternative to weighting is transformation. Find transformations on Lab and/or Field so that in the transformed scale the relationship is approximately linear with constant variance. You may restrict your choice of transformation to square root, log and inverse.

Answer 8.1

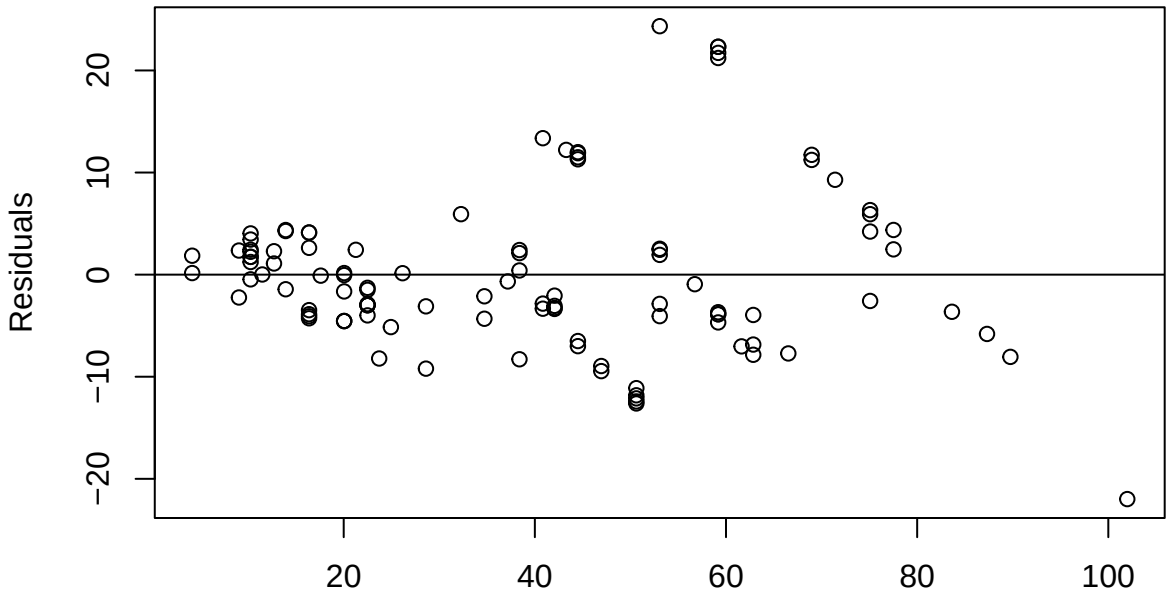
1. We can fit the regression model as follows,

```
library(faraway)
lmod = lm(Lab ~ Field, data = pipeline)
summary(lmod)
```

```
##
## Call:
## lm(formula = Lab ~ Field, data = pipeline)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.985  -4.072  -1.431   2.504  24.334
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.96750    1.57479  -1.249    0.214
## Field        1.22297    0.04107  29.778   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.865 on 105 degrees of freedom
## Multiple R-squared:  0.8941, Adjusted R-squared:  0.8931
## F-statistic: 886.7 on 1 and 105 DF,  p-value: < 2.2e-16
```

```
plot(fitted(lmod),residuals(lmod),xlab="Fitted",ylab="Residuals")
abline(h=0)
```



We can see that we have fairly constant variance but that we have a few points that may be problematic for our model. 2. We can regress log(varlab) on log(meanfield) using the following R-code

```
lmod <- lm(log(varlab)~log(meanfield), data = pipeline)
summary(lmod)
```

```
##
## Call:
## lm(formula = log(varlab) ~ log(meanfield), data = pipeline)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2038 -0.6729  0.1656  0.7205  1.1891
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.3538     1.5715  -0.225   0.8264
## log(meanfield)   1.1244     0.4617   2.435   0.0351 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.018 on 10 degrees of freedom
## Multiple R-squared:  0.3723, Adjusted R-squared:  0.3095
## F-statistic: 5.931 on 1 and 10 DF,  p-value: 0.03513
```

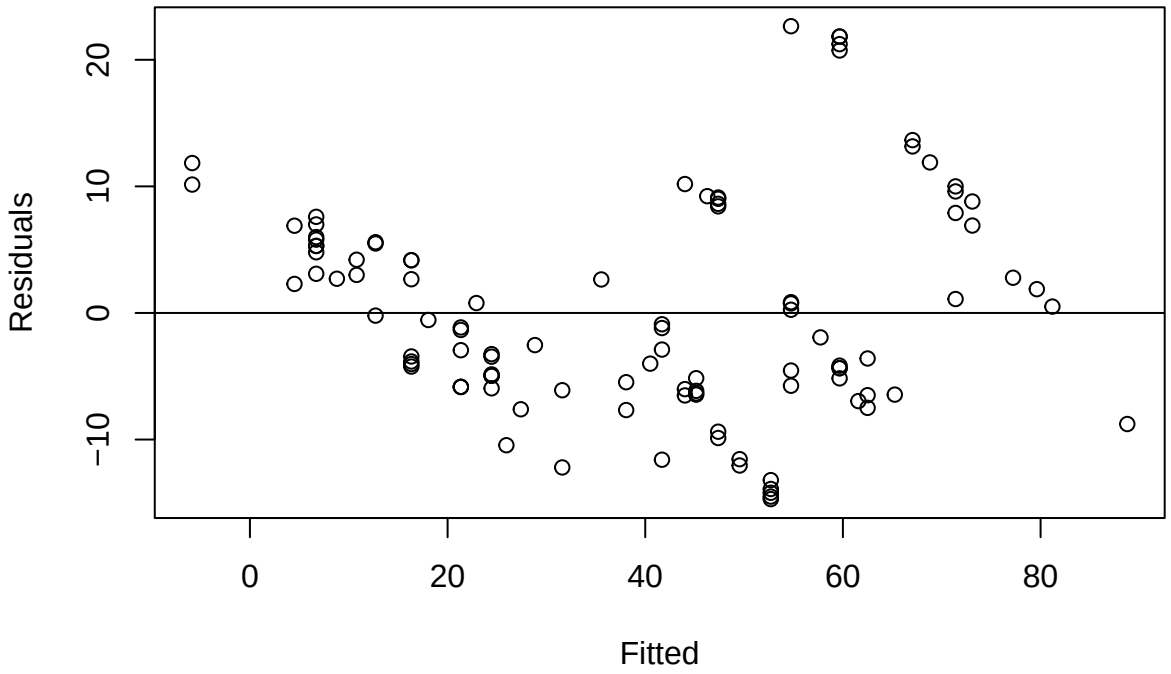
From there, we can make the following adjustment to the weights,

```
lmodWeight <- lm(Lab ~ Field, data = pipeline, weights=1/(0.8264*Field^0.0351))
summary(lmodWeight)
```

```
##
## Call:
## lm(formula = Lab ~ Field, data = pipeline, weights = 1/(0.8264 *
##     Field^0.0351))
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -22.383  -4.229  -1.500   2.593  25.035
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.97572    1.54904  -1.275    0.205
## Field        1.22321    0.04079  29.992   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.094 on 105 degrees of freedom
## Multiple R-squared:  0.8955, Adjusted R-squared:  0.8945
## F-statistic: 899.5 on 1 and 105 DF,  p-value: < 2.2e-16
```

3. We can apply the following transformations to our data so that it appears linear with approximately constant variance as follows.

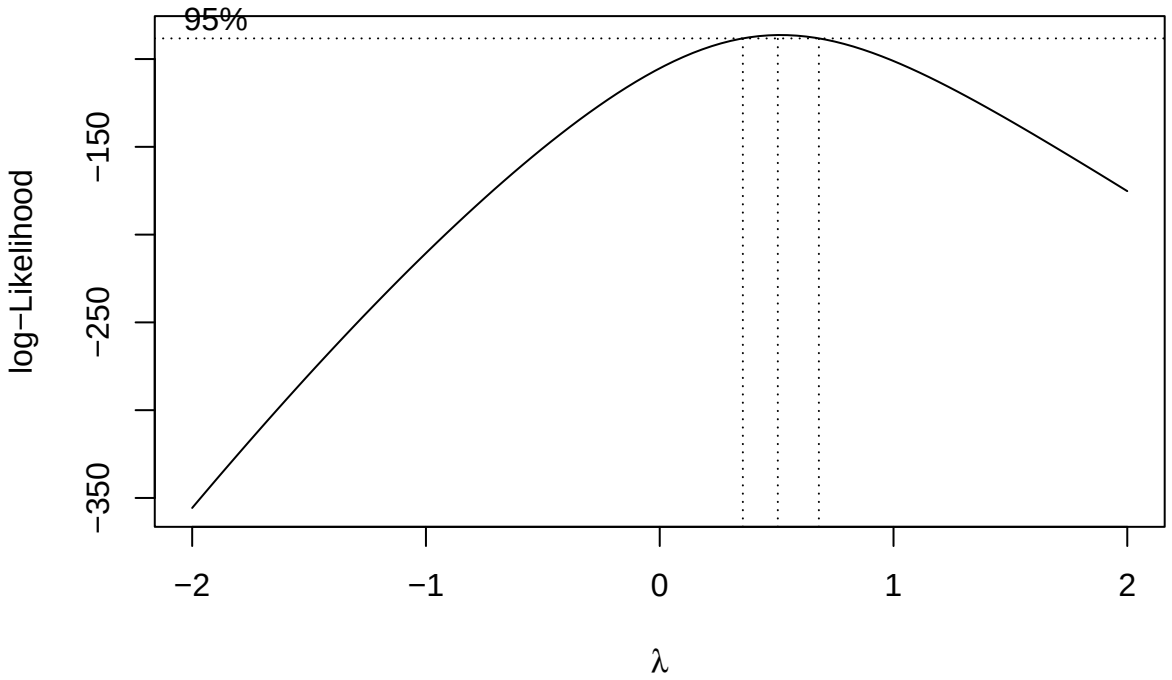
```
lmod = lm(Lab ~ sqrt(Field), data = pipeline)
plot(fitted(lmod),residuals(lmod),xlab="Fitted",ylab="Residuals")
abline(h=0)
```



thus,

we will take the square root relationship, as we can also verify this with our box cox plot as follows

```
field <- pipeline$Field
lab <- pipeline$Lab
library(MASS)
bc <- boxcox(lab ~ field)
```




```
(lambda <- bc$x[which.max(bc$y)])
```

```
## [1] 0.5050505
```

which gives us ≈ 0.5 corresponding to a square root. ## Problem 8.2 Using the divusa data, fit a regression model with divorce as the response and unemployed, femlab, marriage, birth and military as predictors. 1. Make two graphical checks for correlated errors. What do you conclude? 2. Allow for serial correlation with an AR(1) model for the errors. (Hint: Use maximum likelihood to estimate the parameters in the GLS fit by gls(..., method="ML", ...)). What is the estimated correlation and is it significant? Does the GLS model change which variables are found to be significant? 3. Speculate why there might be correlation in the errors.

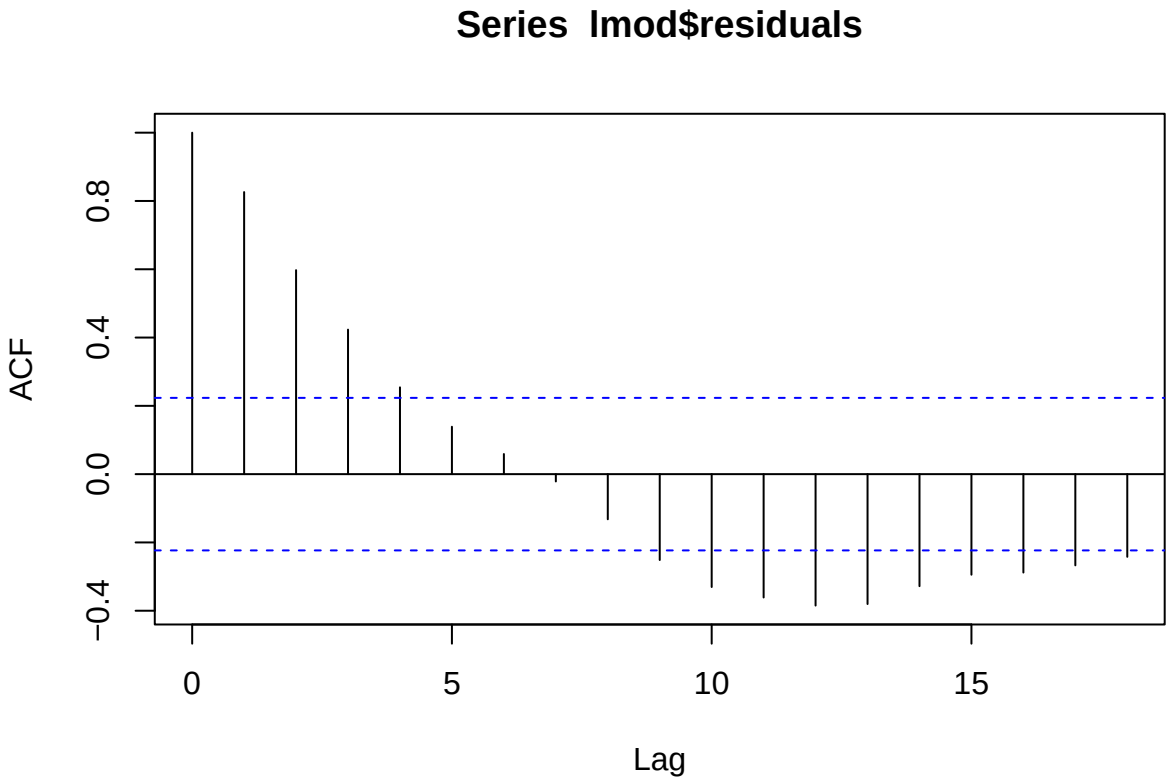
Answer 8.2

1. Fitting the regression model as desired, we obtain,

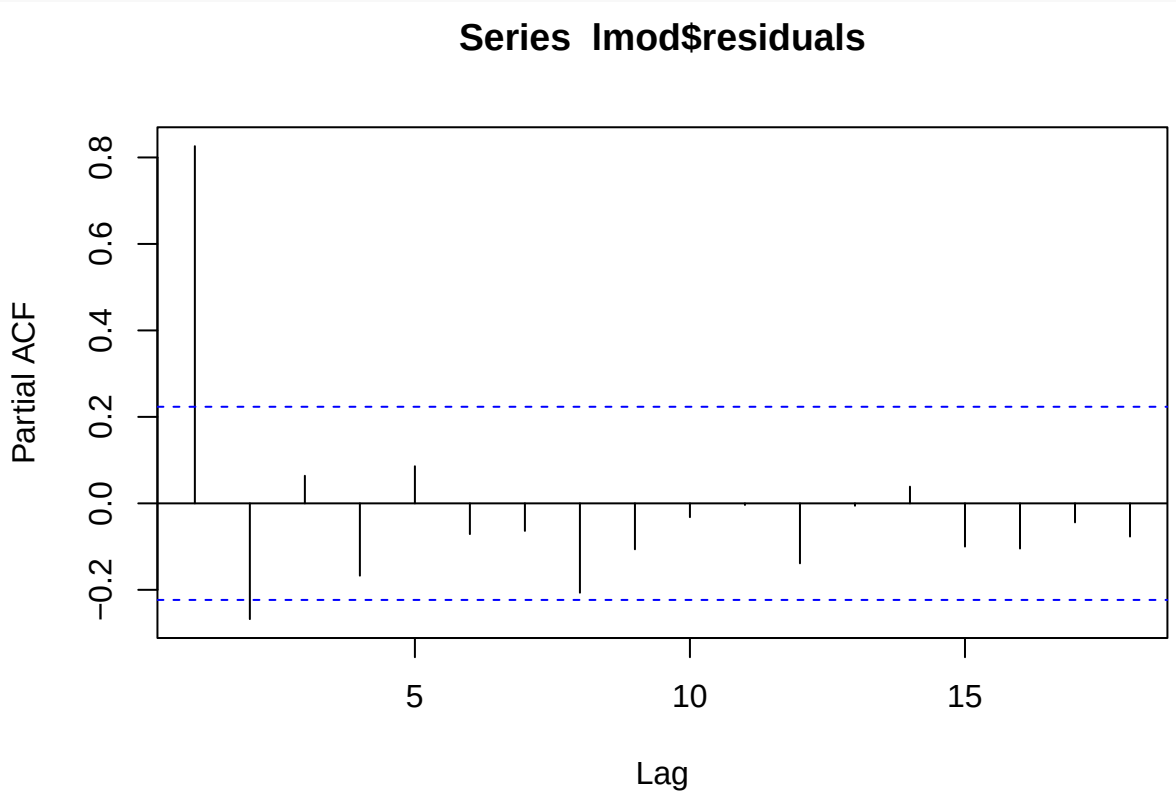
```
library(faraway)
lmod <- lm(divorce ~ unemployed + femlab + marriage + birth +military, data = divusa)
```

Now, we can graphically check for correlated errors as follows,

```
acf(lmod$residuals)
```




```
pacf(lmod$residuals)
```



We can note that there appears to be a higher correlation to the residuals in the for our lower lag values, and thus, we can say that we do have correlated errors. 2. Using the hint in the problem statement, we find that,

```
  require(nlme)
```

```
## Loading required package: nlme
```

```
  glsMod <- gls(divorce ~ unemployed + femlab + marriage + birth +military, data = divusa,method="ML", c
  #summary(glsMod)
  glsMod
```

```
## Generalized least squares fit by maximum likelihood
##   Model: divorce ~ unemployed + femlab + marriage + birth + military
##   Data: divusa
##   Log-likelihood: -81.97613
##
## Coefficients:
## (Intercept)  unemployed      femlab    marriage       birth    military
## -7.05968162  0.10764313  0.31208493  0.16432630 -0.04990919  0.01794640
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi
## 0.9715486
## Degrees of freedom: 77 total; 71 residual
## Residual standard error: 2.907665
```

Looking at the results of our gls, we can see that we get a correlation of apparently 0.97155, which indicates high autocorrelation errors and we can note that it is significant. Now, comparing the coefficients with those of our earlier linear model, we can see that we obtain,

```
  summary(lmod)
```

```
##
## Call:
## lm(formula = divorce ~ unemployed + femlab + marriage + birth +
##     military, data = divusa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8611 -0.8916 -0.0496  0.8650  3.8300
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.48784    3.39378   0.733   0.4659
## unemployed  -0.11125    0.05592  -1.989   0.0505 .
## femlab       0.38365    0.03059  12.543  < 2e-16 ***
## marriage     0.11867    0.02441   4.861 6.77e-06 ***
## birth       -0.12996    0.01560  -8.333 4.03e-12 ***
## military    -0.02673    0.01425  -1.876   0.0647 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.65 on 71 degrees of freedom
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9152
## F-statistic: 165.1 on 5 and 71 DF,  p-value: < 2.2e-16
```

We can see that the significance of the coefficients appears to change with which variables end up being

significant. 3. When we have correlation in the errors, that usually indicates a some kind of underlying trend to the errors that our gls outputs. However, looking at our data, we can consider the fact that some of our data may be correlated due to the fact that we are taking annual trends over years, and thus, there is a decent change that one years data may not be independent of the data of another year if there are consistent global trends that are not captured in our data, since we are dealing with a time series over the 1920s.

Problem 8.3

For the salmonella dataset, fit a linear model with colonies as the response and log(dose+1) as the predictor. Check for lack of fit.

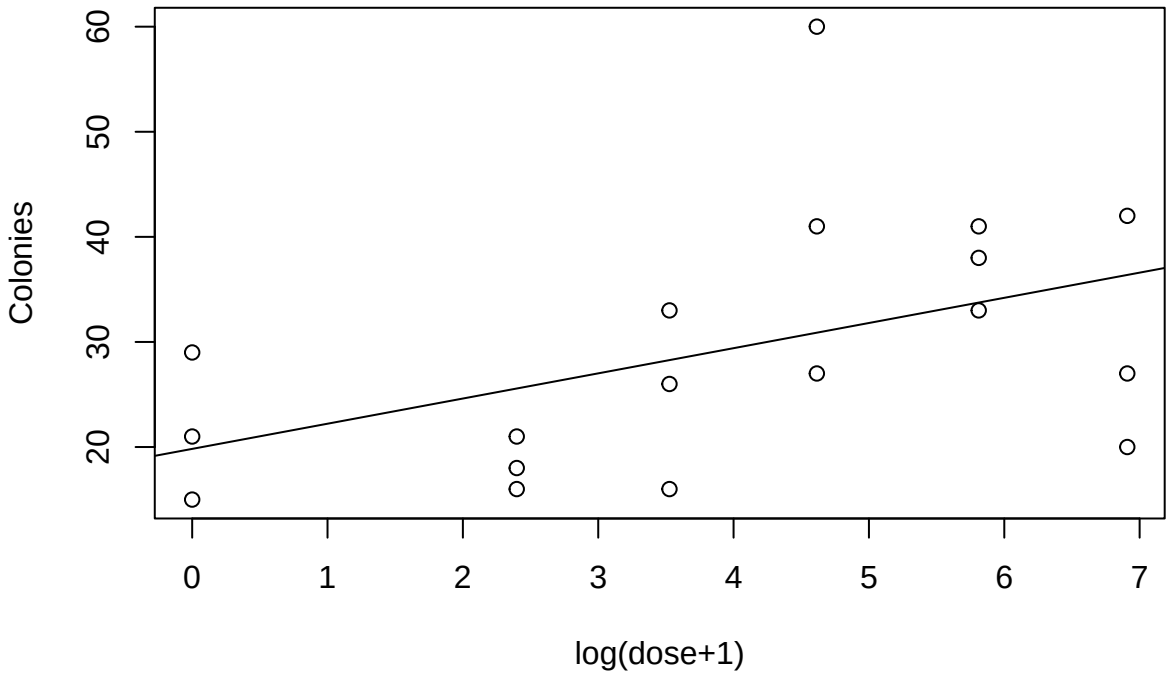
Answer 8.3

We can fit the linear model as follows and follow the process described in the book to check for lack of fit,

```
library(faraway)
lmodColonies <- lm(colonies ~ log(dose+1), data = salmonella)
summary(lmodColonies)
```

```
##
## Call:
## lm(formula = colonies ~ log(dose + 1), data = salmonella)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.376  -6.882  -1.509   5.400  29.119
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     19.823      5.064   3.915  0.00123 **
## log(dose + 1)    2.396      1.128   2.125  0.04955 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.84 on 16 degrees of freedom
## Multiple R-squared:  0.2201, Adjusted R-squared:  0.1713
## F-statistic: 4.514 on 1 and 16 DF,  p-value: 0.04955
```

```
plot(colonies ~ log(dose+1), salmonella,xlab="log(dose+1)", ylab="Colonies")
abline(coef(lmodColonies))
```



From these diagnostic plots, it appears that there may be a lack of fit, especially as indicated by the multiple R^2 which is 0.2201. Further looking into the apparent lack of fit, we can see that,

```
lmoda <- lm(colonies ~ factor(log(dose+1)), salmonella)
anova(lmodColonies, lmoda)
```

```
## Analysis of Variance Table
##
## Model 1: colonies ~ log(dose + 1)
## Model 2: colonies ~ factor(log(dose + 1))
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     16 1881.1
## 2     12 1091.3  4    789.73 2.1709 0.1342
```

We can see that as our pr(>F) is fairly small, although we do not have a large R^2, we can be confident that we do not have an issue of lack of fit from our model, as the Pr(>F) is not within our boundaries for significance.

Problem 8.4

For the cars dataset, fit a linear model with distance as the response and speed as the predictor. Check for lack of fit.

Answer 8.4

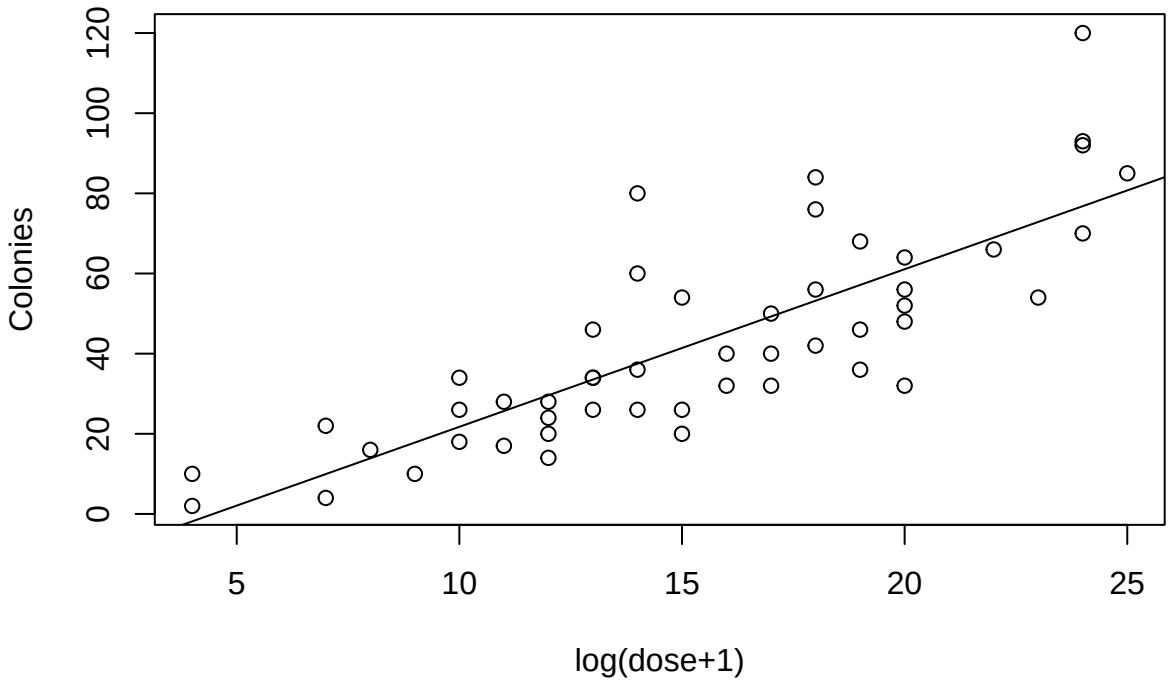
We can fit the linear model as follows and follow the process described in the book to check for lack of fit,

```
library(faraway)
lmodCars <- lm(dist ~ speed, data = cars)
summary(lmodCars)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

```
plot(dist ~ speed, cars,xlab="log(dose+1)", ylab="Colonies")
abline(coef(lmodCars))
```



Now, we can check for a lack of fit as follows,

```
lmodaCars <- lm(dist ~ factor(speed), cars)
anova(lmodCars, lmodaCars)
```

```
## Analysis of Variance Table
##
## Model 1: dist ~ speed
## Model 2: dist ~ factor(speed)
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     48 11353.5
## 2     31  6764.8 17    4588.7 1.2369 0.2948
```

Looking at these results and at our graphs, we can note that although we have a lower R^2 value, our anova test indicates to us that we have a good fitting linear model for this dataset, as indicated by the larger Pr(>F) than our significance level.