Key for Chem 351: Final Exam

Please present responses to the following problems. Use a word processor or Rmarkdown to prepare your responses in the form of an organized and well-written narrative (that is, paragraphs and complete sentences) that explains how you are approaching the problem and why this approach is appropriate, and that presents your results. Where appropriate, be sure to state null and alternative hypotheses and to state clearly your conclusions. Be critical when evaluating the results of statistical tests. Do you trust the results or are there reasons that you find them suspicious? Be sure to annotate your work fully and completely so that I can assign partial credit where appropriate. In general this means you should include relevant code and output from R or, if you use Excel, submit electronic copies of your spreadsheets. Together, your narrative, your annotations, and your supporting documents must clearly guide me through your work.

You are free to use your textbook, the library, web resources, previous problem sets, and your notes and handouts while working on this exam. You are not free to discuss any portion of this exam with other students or with faculty members other than the instructor. This restriction applies to R as well. Please direct all questions about the exam or about the use of R to the instructor.

Data files (.RData) are available on the course's archives page. If you have trouble downloading any of these files, I can email them to you on request.

A **printed, not emailed** copy of your solutions is due in my office by 4:00 pm on Friday, December 16th.

Good Luck!

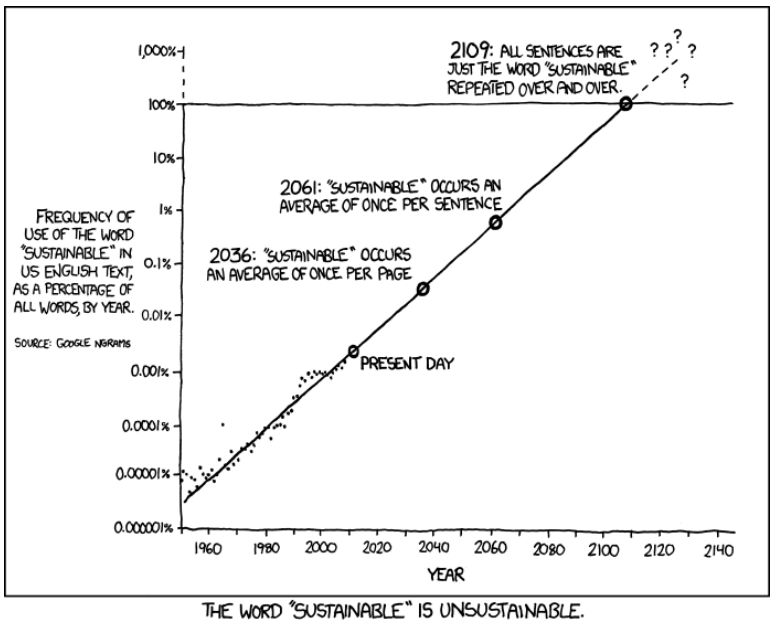


Figure 1: The problem with extrapolating data. (http://xkcd.com/1007/)

Note 1: In many cases there is more than one approach to solve these problems; the solutions here, therefore, outline my thoughts on how best to develop solutions. A few notes to this effect are scattered throughout this answer key.

Note 2: To save space, the questions are lightly edited to remove code that highlighted the structure of the .RData files.

Problem 1. A simple exercise in statistics is to analyze bags of M&Ms for the number of candies of each color. The file M&M.RData, which we explored earlier this semester, is a data frame that provides a typical set of results using a sample of 30 bags of M&Ms. Because the 30 bags do not contain identical numbers of M&Ms—note the last column of the data frame gives the total number of M&Ms in each bag—before answering the questions that follow you first must convert the raw data so that the results for each color are expressed in terms of the percentage of the total M&Ms in each bag.

(a) Report the mean, the standard deviation, the variance, the range, and the median for each color of M&Ms? (*Hint: make your life simpler by using the apply function, which was introduced in our last R session.*)

```
# begin by calculating the percents
per.mm = 100 * mm[, 1:6]/mm[ , 7]
# means (rounded to two decimal places)
round(apply(per.mm, 2, mean), digits = 2)
```

```
##   blue  brown  green orange    red yellow
##  11.75  25.81   6.52  12.75  17.64  25.52
```

```
# standard deviations (rounded to two decimal places)
round(apply(per.mm, 2, sd), digits = 2)
```

```
##   blue  brown  green orange    red yellow
##   6.67   5.00   4.74   4.98   6.60   7.59
```

```
# variances (rounded to two decimal places)
round(apply(per.mm, 2, var), digits = 2)
```

```
##   blue  brown  green orange    red yellow
##  44.55  25.04  22.50  24.83  43.55  57.67
```

```
# ranges (rounded to two decimal places)
round(apply(per.mm, 2, max) - apply(per.mm, 2, min), digits = 2)
```

```
##   blue  brown  green orange    red yellow
##  25.03  22.56  16.07  19.08  24.58  31.26
```

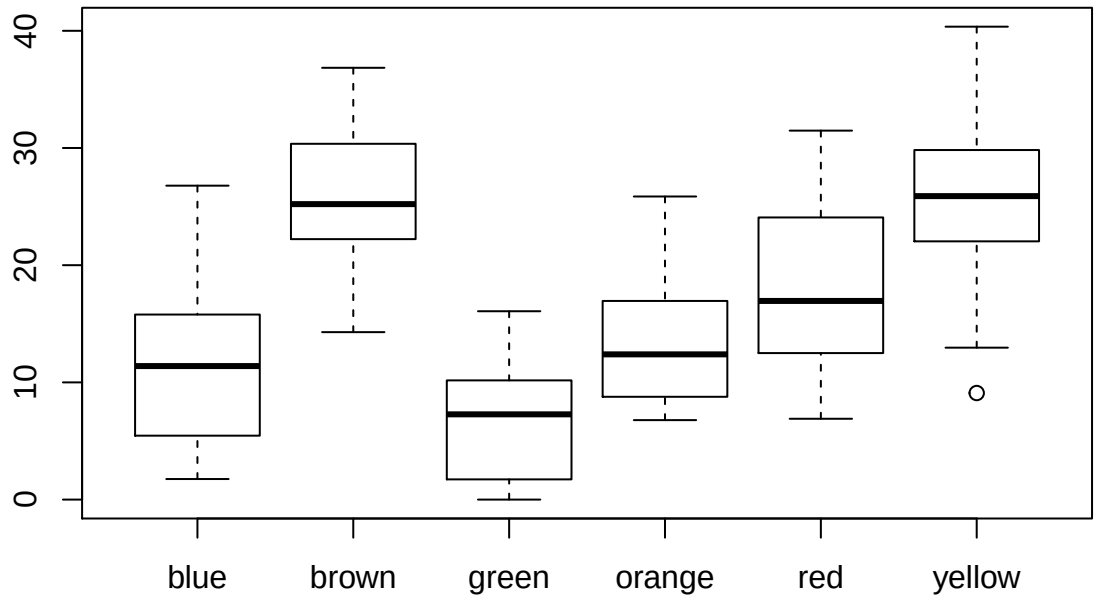
```
# medians (rounded to two decimal places)
round(apply(per.mm, 2, median), digits = 2)
```

```
##   blue  brown  green orange    red yellow
##  11.39  25.21   7.27  12.39  16.95  25.89
```

(b) Is there evidence for outliers in the data?

A boxplot is simple way to get a quick overview of the data; as shown below, the smallest value for yellow is a possible outlier.

```
boxplot(per.mm, labels = c("blue", "brown","green", "orange", "red", "yellow"))
```



Of course, this is just one result from a sample of 30 bags of M&Ms, or just 3.3% of the sample. A true outlier is a result that not consistent with the underlying distribution of results. We might reasonably ask if a single result from 30 trials is a true outlier. To evaluate this further we should use an appropriate statistical test, such as the Dixon Q-test or the Grubb test. As we see here

```
library(outliers)
dixon.test(per.mm$yellow)
```

```
##
##  Dixon test for outliers
##
## data:  per.mm$yellow
## Q = 0.17929, p-value = 0.9948
## alternative hypothesis: lowest value 9.09090909090909 is an outlier
```

```
grubbs.test(per.mm$yellow)
```

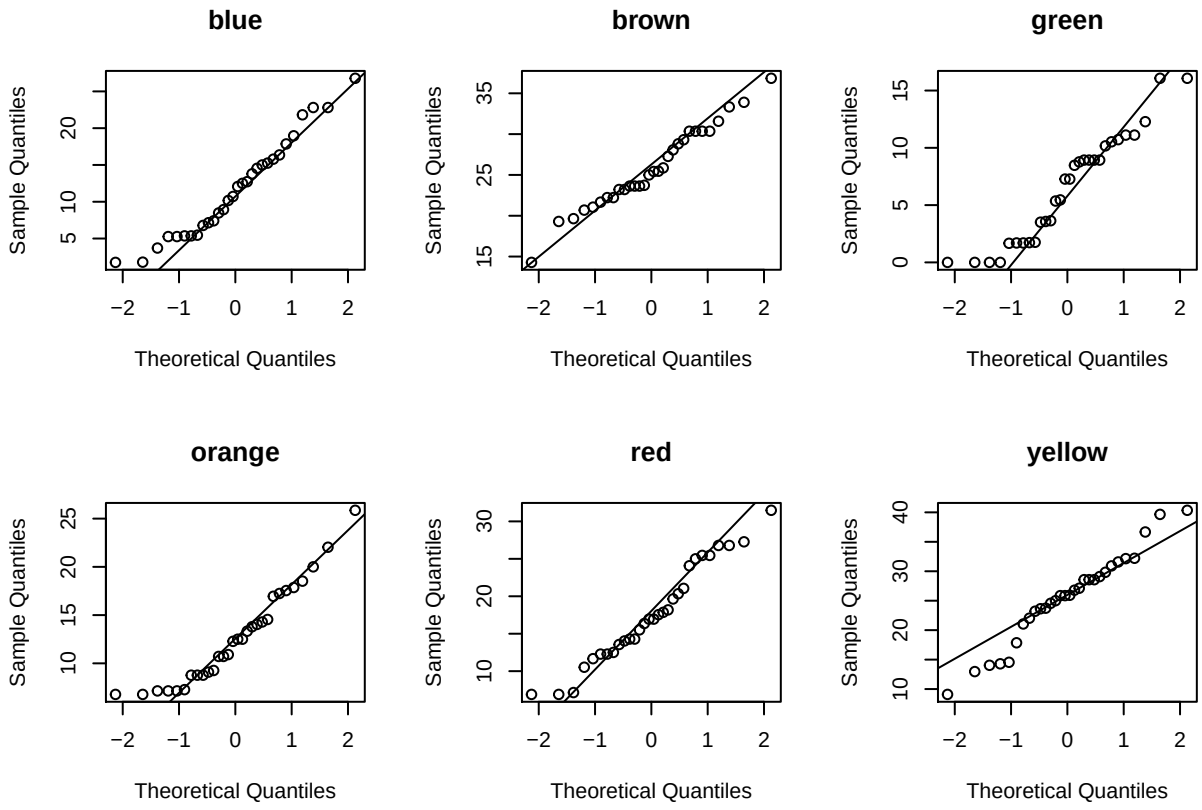
```
##
##  Grubbs test for one outlier
##
## data:  per.mm$yellow
## G = 2.16390, U = 0.83297, p-value = 0.3741
## alternative hypothesis: lowest value 9.09090909090909 is an outlier
```

the p-value for neither test provides convincing evidence that the smallest result for yellow M&Ms is an outlier.

(c) Is there evidence that the percentages of M&Ms of each color are distributed normally?

To evaluate this, let's examine a qq-plot for each color of M&Ms

```
old.par = par(mfrow = c(2, 3))
qqnorm(per.mm$blue, main = "blue"); qqline(per.mm$blue)
qqnorm(per.mm$brown, main = "brown"); qqline(per.mm$brown)
qqnorm(per.mm$green, main = "green"); qqline(per.mm$green)
qqnorm(per.mm$orange, main = "orange"); qqline(per.mm$orange)
qqnorm(per.mm$red, main = "red"); qqline(per.mm$red)
qqnorm(per.mm$yellow, main = "yellow"); qqline(per.mm$yellow)
```




```
par(old.par)
```

Most of these plots show evidence of a slight tail toward higher percentages (see, for example, blue, green, and orange) or a distribution that is a bit broader than expected (see, for example, yellow). On the whole, however, the distributions appear to be normally distributed, particularly given that 30 bags of M&Ms is not a large sample. Although it is tempting to use a histogram to evaluate the distribution of results, it is a less useful approach than a qqplot for at least two reasons: a histogram's shape may be quite sensitive to the number of bins used and there is no statistical parameter that you can test against the assumption of a normal distribution.

(d) What are the odds of obtaining a package of M&Ms without a green M&M?

```
round(100 * pnorm(0, mean(per.mm$green), sd(per.mm$green)), digits = 2)
```

```
## [1] 8.46
```

As shown above, the odds are approximately 8.5%. Note that in our set of 30 bags of M&Ms that

```
table(mm$green)
```

```
##
## 0 1 2 3 4 5 6 7 9
## 4 5 3 2 2 6 5 1 2
```

a total of 4 bags do not contain a green M&M; this is 13.3% of the sample, a result that is not far off from the prediction based on a normal distribution.

(e) Is there any difference, at $\alpha = 0.05$, between the percentage of blue and the percentage of red M&Ms found in a typical package of M&Ms?

Although it is tempting to use a paired t-test here, this data is best treated as unpaired as we are not interested in whether there is a correlation (direct or indirect) between the colors of M&Ms; thus, we first use

an F-test to see if we can pool variances

```
var.test(per.mm$blue, per.mm$red, alternative = "two.sided")
```

```
##
##  F test to compare two variances
##
## data:  per.mm$blue and per.mm$red
## F = 1.0228, num df = 29, denom df = 29, p-value = 0.952
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.486821 2.148917
## sample estimates:
## ratio of variances
##           1.022809
```

and then, a t-test using a pooled standard deviation

```
t.test(per.mm$blue, per.mm$red, mu = 0,  alternative = "two.sided", var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  per.mm$blue and per.mm$red
## t = -3.4316, df = 58, p-value = 0.001112
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -9.311093 -2.450435
## sample estimates:
## mean of x mean of y
##  11.75462  17.63538
```

The resulting p-value of 0.00111 is less than $\alpha = 0.05$, suggesting that the difference in the values is greater than the variance in results within each bag; thus, we find evidence that there is a significant difference in the percentage of blue and of red M&Ms in this sample of 30 bags.

Problem 2. The file Calibration.RData contains calibration data for the analysis of nitrite using two different spectrometers. The vector nitrite provides the concentrations of nitrite in several standards and the vectors SpectA and SpectB give the absorbance for each standard as measured on two spectrometers.

(a) Is there any evidence at $\alpha = 0.05$ that the two spectrometers provide different results?

Because the nitrite samples are at several levels of concentration, we will treat this as paired data; however, because the absorbance of each of the standards is measured three times (that is, we have replicates), we should first average the replicates as there is not a one-to-one correspondence between the replicates.

```
sa = c(mean(SpectA[1:3]), mean(SpectA[4:6]), mean(SpectA[7:9]), mean(SpectA[10:12]), mean(SpectA[13:15]))
sb = c(mean(SpectB[1:3]), mean(SpectB[4:6]), mean(SpectB[7:9]), mean(SpectB[10:12]), mean(SpectB[13:15]))
t.test(sa, sb, mu = 0, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  sa and sb
## t = 2.4015, df = 4, p-value = 0.07423
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.001404983  0.019404983
## sample estimates:
```

```
## mean of the differences
##                   0.009
```

At a confidence level of $\alpha = 0.05$, we do not have evidence that there is a significant difference between the results of the two spectrometers. *Note: Treating this data as unpaired is a serious error as the variance within each spectrometer's absorbance values includes both the variance inherent in measuring absorbance and the variance that results from using standards of different concentrations. To evaluate the two spectrometers relative to each other we must find a way to eliminate the variance due to the concentrations of the standards, which we accomplish by using a paired t-test.*

(b) Using the data for spectrometer A, calculate the value of ϵb for each standard. Is there any evidence at $\alpha = 0.05$ that the value of ϵb is dependent upon the concentration of nitrite?

There a couple of ways to approach this, but the easiest for us to work with is an analysis of variance. To accomplish this we will calculate ϵb for each trial, create a vector that assigns each such value to a particular standard, create a data frame to hold these values, and then complete the analysis of variance calculation.

```
eb = SpectA/nitrite
std = c(rep("Std 1", 3), rep("Std 2", 3), rep("Std 3", 3),
        rep("Std 4", 3), rep("Std 5", 3))
eb.df = data.frame(eb, std)
eb.aov = aov(eb ~ std, data = eb.df)
summary(eb.aov)
```

```
##             Df    Sum Sq   Mean Sq F value  Pr(>F)
## std          4 0.0002738 6.846e-05   9.416 0.00201 **
## Residuals   10 0.0000727 7.270e-06
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With a p-value of 0.0020, there is evidence to suggest that there is a significant variability in ϵb values. *Note: Another approach to this problem is to complete a linear regression analysis that predicts ϵb as a function of the concentration of nitrite; if there is a relationship between the two, then the model's parameter for the slope will be significantly different from zero. A linear regression of absorbance as a function of the concentration of nitrite is not useful here as the slope gives an estimate for ϵb, but does not provide evidence for whether there is a relationship between ϵb and the concentration of nitrite.

Problem 3. Alexander Moore and Nathan Bower, a student and a faculty member at Colorado College, published a paper in which they analyzed four bison and four cow patties for several parameters, including moisture content, nitrogen, carbon, and total chlorophyll. Students collected and analyzed both fresh patties and aged patties from bison and cows that were maintained on a nature preserve and on a ranch. The data for this problem are stored as a dataframe in the file Patties.RData. Note that some of the variables are numeric and some are categorical.

(a) As the patties age in the field they undergo a slow decomposition that produces some obvious and subtle changes in composition. For example, a statistical analysis is not needed to show that weathered patties have significantly less moisture and total chlorophyll. But what about carbon and nitrogen? Using a suitable statistical analysis determine at $\alpha = 0.05$ if there is any significant time-dependent change in percent nitrogen. Repeat your analysis for percent carbon.

Because the experimental design is balanced—the four fresh patties and the four weathered patties have one trial each for bison:ranch, bison:nature preserve, cow:ranch, and cow:nature preserve—a paired t-test will meet our needs.

```
t.test(PattyData$nitrogen[c(1, 3, 5, 7)], PattyData$nitrogen[c(2, 4, 6, 8)], mu = 0, paired = TRUE)
```

```
##
##  Paired t-test
##
```

```
## data:  PattyData$nitrogen[c(1, 3, 5, 7)] and PattyData$nitrogen[c(2, 4, 6, 8)]
## t = 1.9542, df = 3, p-value = 0.1457
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0785688  0.3285688
## sample estimates:
## mean of the differences
##                   0.125
```

```
t.test(PattyData$carbon[c(1, 3, 5, 7)], PattyData$carbon[c(2, 4, 6, 8)], mu = 0, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  PattyData$carbon[c(1, 3, 5, 7)] and PattyData$carbon[c(2, 4, 6, 8)]
## t = -0.085894, df = 3, p-value = 0.937
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.610158  7.210158
## sample estimates:
## mean of the differences
##                    -0.2
```

For both nitrogen and for carbon the p-value is greater than $\alpha = 0.05$, suggesting that uncertainty in the measurements is sufficient to explain any differences between the results.

(b) There obviously is quite a bit of difference in the total chlorophyll content of these patties. Build a linear model that predicts the amount of total chlorophyll as a function of animal, habitat, and age. Include all first-order and all binary interaction terms in your initial model, but exclude the ternary interaction. Your final model should include only those terms that you find statistically significant. Given that the factors are not continuous variables—for example, the age of a patty is assigned using the binary classification of fresh or weathered even though the actual ages of the patties are continuous—be forgiving in your choice of p-values. Evaluate your model by comparing your model's predicted amount of total chlorophyll to the actual total chlorophyll.

To create a suitable model we first must create vectors for each of the independent variables (animal, habitat, and age); these are best set up as vectors consisting of 1s and –1s, although other choices will lead you to the same general conclusions. Next, we build a linear model to predict the dependent variable (total chlorophyll) as a function of our independent variables; note that we subtract out the unwanted ternary interaction term.

```
animal = c(rep(1, 4), rep(-1, 4))
habitat = rep(c(1, 1, -1, -1), 2)
age = rep(c(1, -1), 4)
tc.lm = lm(PattyData$totchlor ~ animal * habitat * age - animal:habitat:age)
summary(tc.lm)
```

```
##
## Call:
## lm(formula = PattyData$totchlor ~ animal * habitat * age - animal:habitat:age)
##
## Residuals:
##      1      2      3      4      5      6      7      8
##  25.62 -25.62 -25.62  25.62 -25.62  25.62  25.62 -25.62
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      334.12      25.62  13.039   0.0487 *
```

```
## animal           150.87      25.62   5.888   0.1071
## habitat           15.87      25.62   0.620   0.6469
## age              319.63      25.62  12.473   0.0509 .
## animal:habitat    31.12      25.62   1.215   0.4385
## animal:age       144.88      25.62   5.654   0.1115
## habitat:age       14.87      25.62   0.580   0.6652
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 72.48 on 1 degrees of freedom
## Multiple R-squared:  0.9956, Adjusted R-squared:  0.9689
## F-statistic:  37.4 on 6 and 1 DF,  p-value: 0.1245
```

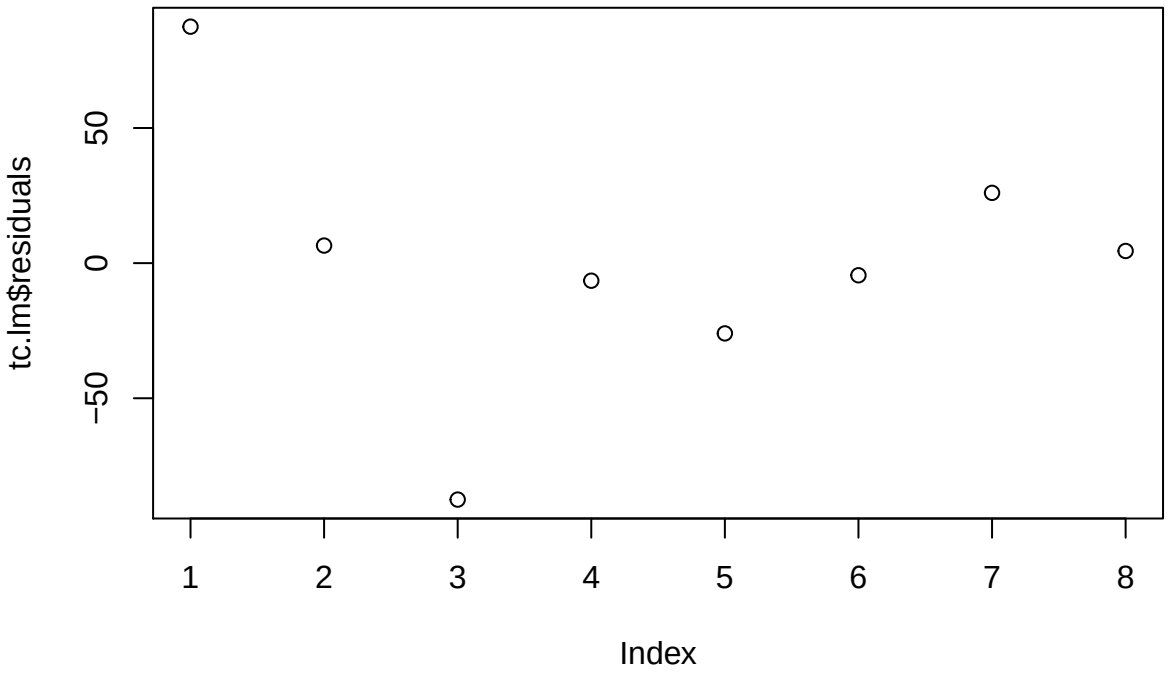
The model's coefficients cleanly fall into two distinct groups: those with p-values less than 0.12 and those greater than 0.42. To fine-tune our model, we will keep as the intercept, animal, age, and animal:age as parameters.

```
tc.lm = lm(PattyData$totchlor ~ animal * age)
summary(tc.lm)
```

```
##
## Call:
## lm(formula = PattyData$totchlor ~ animal * age)
##
## Residuals:
##     1     2     3     4     5     6     7     8
##  87.5   6.5 -87.5  -6.5 -26.0  -4.5  26.0   4.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   334.12      22.91  14.587 0.000128 ***
## animal        150.87      22.91   6.587 0.002751 **
## age           319.62      22.91  13.954 0.000153 ***
## animal:age    144.87      22.91   6.325 0.003198 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 64.79 on 4 degrees of freedom
## Multiple R-squared:  0.9858, Adjusted R-squared:  0.9752
## F-statistic:  92.7 on 3 and 4 DF,  p-value: 0.0003752
```

To evaluate the model, we examine a plot of the residual errors, which, despite two large residual errors for the first and the third sample (bison:fresh and bison:weathered), shows that the residual errors are not particularly large.

```
plot(tc.lm$residuals)
```

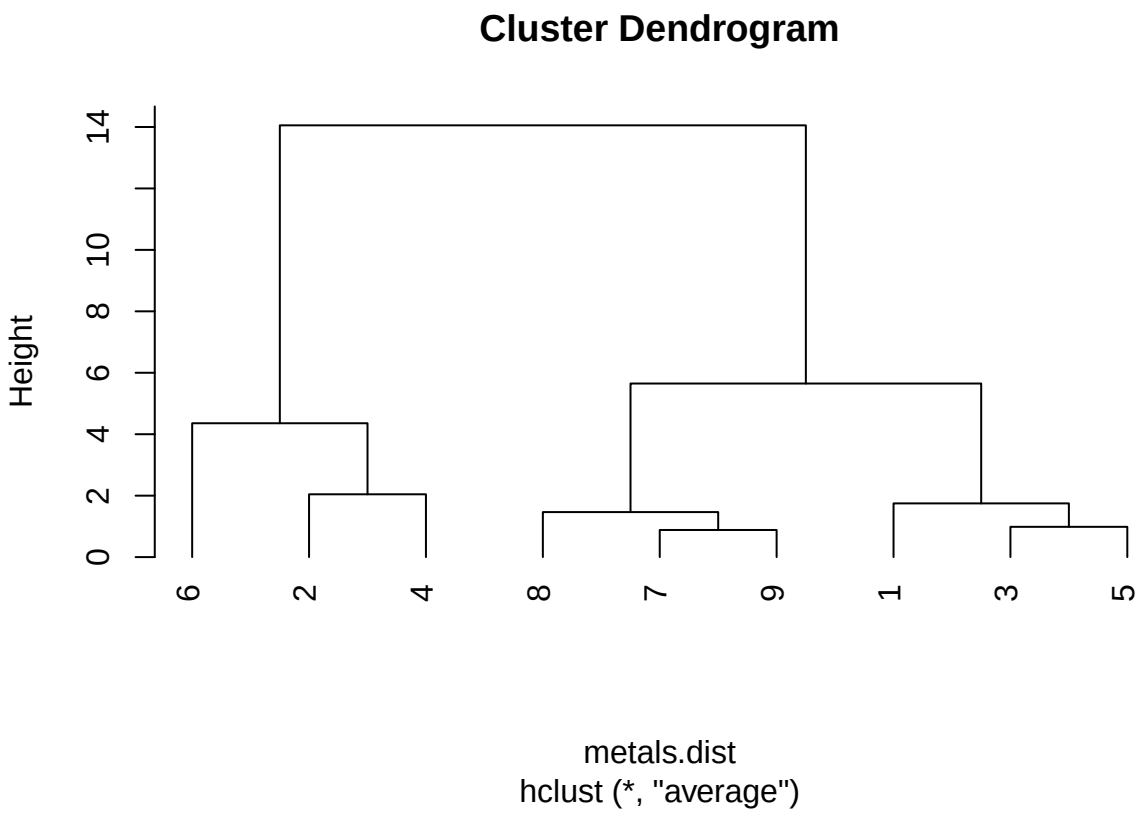


Problem 4. The file SedMetal.RData contains results for the concentrations, in ppm, of four metals in nine sediment samples.

(a) Using a cluster analysis, assign the nine sediments into two to four groups.

To complete the cluster analysis we first create a distance matrix and then choose a clustering method; the result using is shown here when using the average distance, but all three methods yield the same general result.

```
metals.dist = dist(metals)
metals.clust = hclust(metals.dist, method = "average")
plot(metals.clust, hang = -1)
```



Based on this dendrogram, we see that sites 7, 8, and 9 form a distinct cluster, that sites 1, 3, and 5 form a distinct second cluster, and that sites 2, 4, and 6 form a distinct third cluster; as well, we see that site 6 is at a greater distance from sites 2 and 4 than site 8 is from 7 and 9 or that site 1 is from 3 and 5.

(b) Complete a principal component analysis of the data. How much of the variance in the data is explained by the first two principal components? Prepare score and loading plots for the first two principal components. Are your results consistent with your cluster analysis results? Explain.

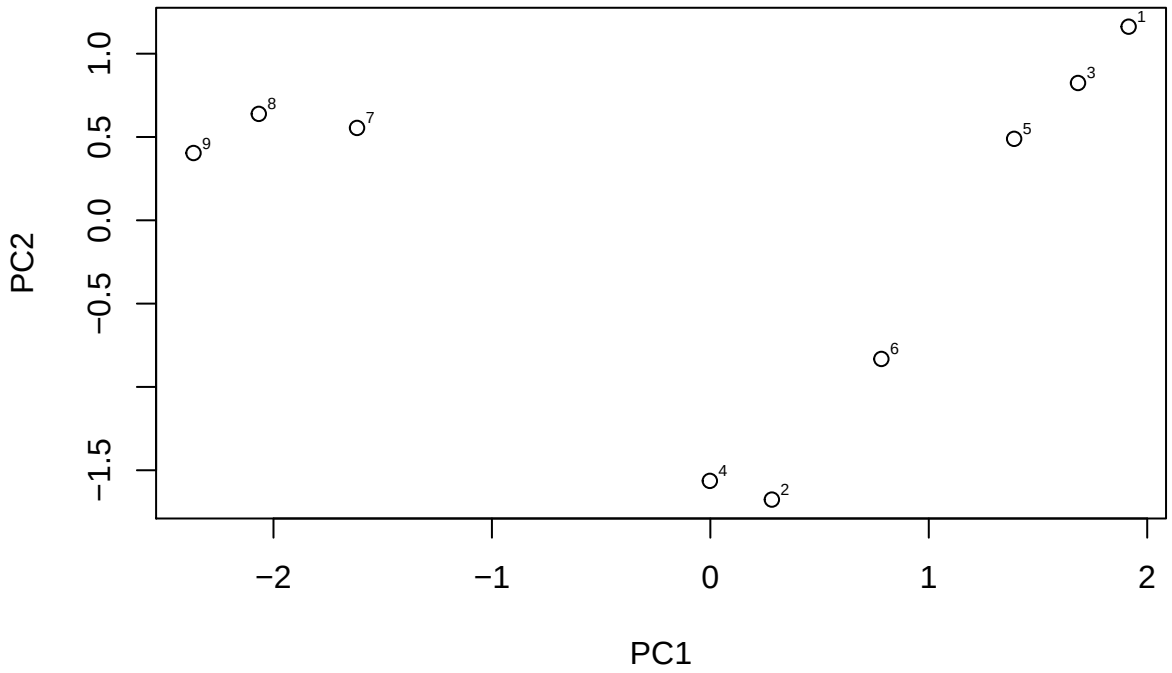
To complete the PCA analysis we will center and scale the data; thus

```
metals.pca = prcomp(metals, center = TRUE, scale = TRUE)
summary(metals.pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.6438 1.0661 0.36569 0.16625
## Proportion of Variance 0.6755 0.2842 0.03343 0.00691
## Cumulative Proportion  0.6755 0.9597 0.99309 1.00000
```

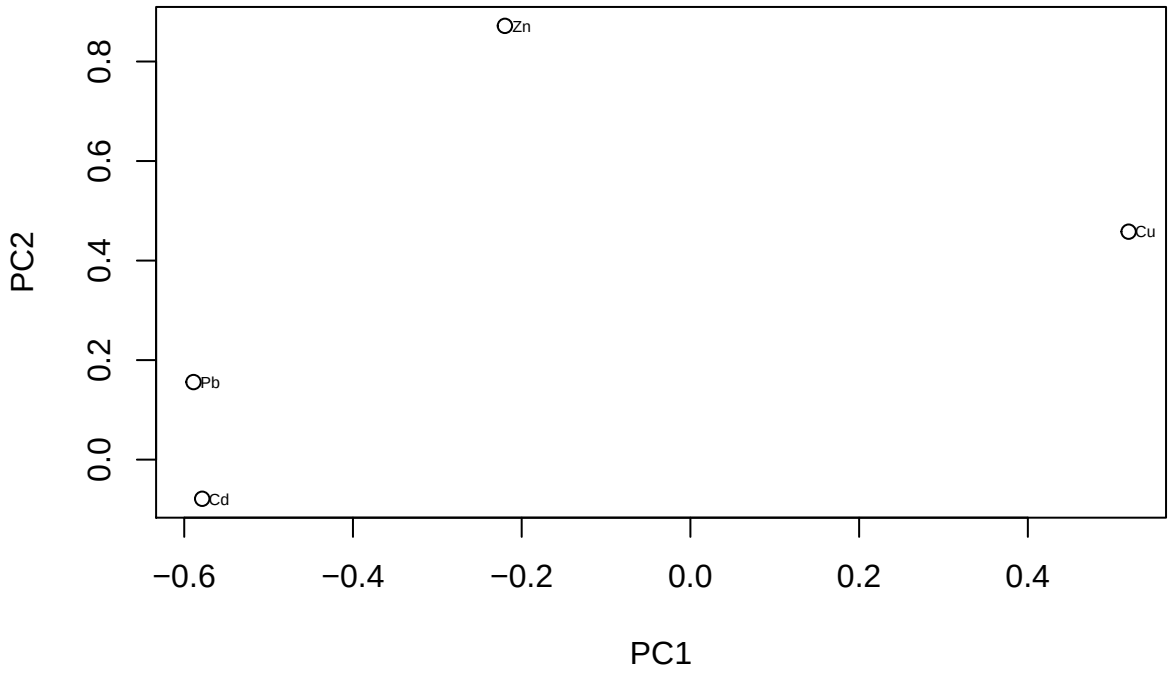
we find that 96% of the variances is explained by the first two principal components. The resulting scores plot is shown here with the individual sediment samples identified

```
plot(metals.pca$x)
text(x = metals.pca$x[, 1] + 0.06,
     y = metals.pca$x[ , 2] + 0.06,
     labels = c(1:9), cex = 0.5)
```



The scores plot matches our cluster results in that we see three distinct clusters, each of identical to our results from part (a); we also see that site 6 is somewhat removed from sites 2 and 4. Thee loadings plot, which is shown here

```
plot(metals.pca$rotation)
text(x = metals.pca$rotation[, 1] + 0.02,
     y = metals.pca$rotation[ , 2],
     labels = c("Zn", "Cd", "Pb", "Cu"), cex = 0.5)
```



suggests that Zn has a high positive correlation with the second principal component, that Cu has modest positive correlations with both principal components, and that Pb and Cd have a high negative correlation with the first principal component.

(c) A tenth sediment sample has the following concentrations: Zn, 21.8 ppm; Cd, 1.3 ppm; Pb, 0.5 ppm; Cu, 3.3 ppm. To which class of sediments samples does this sample belong? Clearly explain your reasoning.

With a relatively low level of Zn and relatively medium levels of Pb and of Cd, sample 10 is most like samples 2 and 4.

Problem 5. One way to monitor a chemical reaction is to follow the absorbance of any one reactant or product as a function of time. Things become complicated, however, when all the reactants and products absorb. The data in the file KinAnal.RData contains results for the study of the reaction

$$A + B \rightarrow C$$

The object CalData is a 25×22 data frame containing the spectra for 25 standards that contain known concentrations of A, B, and C recorded at 22 wavelengths (*note: the standards are stable because they lack a catalyst needed to initiate the reaction*). The object ConcData is a 25×3 data frame that gives the concentrations of A, B and C in these 25 standards. Finally, the object KinData is a 30×22 data frame of spectra recorded at 30 times during the reaction at the same 22 wavelengths used for the standards and the corresponding times are provided in the vector time.

(a) Complete a principal component analysis for both the calibration data and the kinetic data and report the number of significant components present in each data set.

```
caldata.pca = prcomp(CalData, center = TRUE, scale = TRUE)
summary(caldata.pca)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6
## Standard deviation     4.4028 1.28056 0.97543 0.15277 0.02191 0.01544
## Proportion of Variance 0.8811 0.07454 0.04325 0.00106 0.00002 0.00001
## Cumulative Proportion  0.8811 0.95565 0.99890 0.99996 0.99998 0.99999
##                            PC7      PC8      PC9     PC10     PC11
## Standard deviation     0.008226 0.004892 0.004444 0.002186 0.001331
## Proportion of Variance 0.000000 0.000000 0.000000 0.000000 0.000000
## Cumulative Proportion  1.000000 1.000000 1.000000 1.000000 1.000000
##                           PC12      PC13      PC14      PC15      PC16
## Standard deviation     0.001103 0.0009223 0.0006735 0.0005655 0.0004885
## Proportion of Variance 0.000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  1.000000 1.0000000 1.0000000 1.0000000 1.0000000
##                           PC17      PC18      PC19      PC20     PC21
## Standard deviation     0.0004011 0.0003179 0.0002837 0.0002172 0.000109
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.000000
## Cumulative Proportion  1.0000000 1.0000000 1.0000000 1.0000000 1.000000
##                           PC22
## Standard deviation     3.341e-05
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

```
kindata.pca = prcomp(KinData, center = TRUE, scale = TRUE)
summary(kindata.pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     4.0652 2.2642 0.57686 0.10333 0.05969 0.01142
## Proportion of Variance 0.7512 0.2330 0.01513 0.00049 0.00016 0.00001
## Cumulative Proportion  0.7512 0.9842 0.99934 0.99982 0.99998 0.99999
##                            PC7      PC8      PC9     PC10    PC11
## Standard deviation     0.009643 0.006304 0.005496 0.004563 0.00432
## Proportion of Variance 0.000000 0.000000 0.000000 0.000000 0.00000
```

```
## Cumulative Proportion  0.999990 1.000000 1.000000 1.000000 1.00000
##                            PC12    PC13     PC14     PC15     PC16
## Standard deviation     0.003073 0.00279 0.002528 0.001685 0.001456
## Proportion of Variance 0.000000 0.00000 0.000000 0.000000 0.000000
## Cumulative Proportion  1.000000 1.00000 1.000000 1.000000 1.000000
##                            PC17     PC18      PC19      PC20      PC21
## Standard deviation     0.001216 0.001042 0.0008595 0.0006522 0.0003999
## Proportion of Variance 0.000000 0.000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  1.000000 1.000000 1.0000000 1.0000000 1.0000000
##                             PC22
## Standard deviation     0.0003525
## Proportion of Variance 0.0000000
## Cumulative Proportion  1.0000000
```

For the calibration data, two principal components account for more than 95% of the variance and three principal components account for more than 99% of the variance; these results are consistent with our knowledge that the data set includes three species. For the kinetic data, two principal components account for almost 99% of the variance; given that we expect a strong correlation between the concentrations of the two reactants, two principal components is not a surprising outcome.

(b) Using the calibration data, complete a multiwavelength linear regression to find the calibration constant (ϵb) matrix and determine the relative error in replicating the concentrations of A, B, and C in the standards.

To complete the multiwavelength linear regression we use the functions available in the script file MLRScript.R; thus

```
source("MLRScript.R")
findeb(CalData, ConcData)
head(eb)
```

```
##    X234.39  X240.29   X246.2   X252.12  X258.03  X263.96  X269.89  X275.82
## A 24.16585 31.57166 29.23892 16.162465 4.107344 1.546053 1.830854 2.805452
## B 20.34509 16.10717 11.69286  9.270915 8.418389 7.513722 7.280666 8.785526
## C 53.32504 55.11233 44.64176 22.900416 9.701110 7.593804 8.201661 9.934636
##     X281.76    X287.7   X293.65   X299.6  X305.56  X311.52  X317.48
## A  4.310507  6.297496  8.689106 11.32666 13.78819 15.76476 16.71422
## B 12.434560 17.541045 23.488025 31.19541 38.60322 46.67863 52.86848
## C 13.035763 17.525766 23.687941 30.23370 39.19249 46.91863 57.42485
##    X323.45  X329.43  X335.41   X341.39   X347.37   X353.36   X358.86
## A 16.47176 14.89336 12.06777  8.850427  5.523459  2.662910  1.051243
## B 54.98776 49.96450 43.33391 32.137886 16.644939  6.142968  1.853636
## C 67.50079 75.16682 80.44013 82.051042 77.749475 69.282112 57.150424
```

```
findconc(CalData, eb)
head(pred.conc)
```

```
##         A       B       C
## 1 0.27698 0.09426 0.06951
## 2 0.27067 0.02946 0.01306
## 3 0.12310 0.02795 0.12916
## 4 0.11700 0.15521 0.04072
## 5 0.42683 0.06071 0.12717
## 6 0.20200 0.15554 0.06839
```

Note: The order in which you pass objects to the function ***findeb*** *is critical. If you use the command* ***findeb(ConcData, CalData)****, then the eb matrix is returned in its transposed form (that is, it has three*

rows instead of three columns); this creates a problem when you use the function ***findconc*** *as it requires the non-transposed eb matrix.*

The percent relative error in predicting concentration is $100 \times \frac{\text{predicted}-\text{actual}}{\text{actual}}$; thus

```
rel.err = 100 * (pred.conc - ConcData)/ConcData
head(rel.err)
```

```
##            A         B          C
## 1  0.3550725  4.733333  0.7391304
## 2 -1.9311594 13.307692  0.4615385
## 3 -3.8281250  7.500000  2.5079365
## 4 -8.5937500  1.444444 -0.6829268
## 5 -1.6520737  4.672414  0.9285714
## 6  1.0000000  1.660131 -0.8840580
```

or an average percent relative error for each species of

```
round(apply(rel.err, 2, mean), digits = 2)
```

```
##     A     B     C
## -0.22  0.81  2.15
```

All three average relative errors are reasonably small, giving us confidence in our linear regression results.

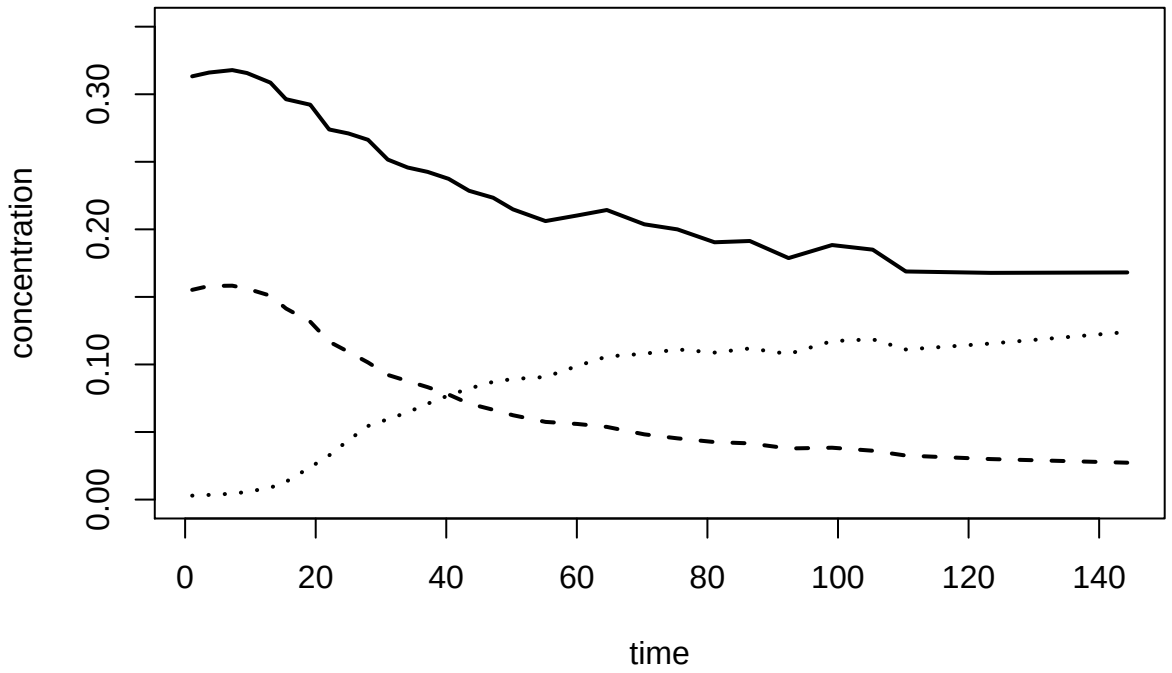
(c) Using your calibration constant matrix, determine the concentrations of A, B, and C in the kinetic data. Create a single plot that shows how the concentrations of A, B, and C change as a function of time and comment on the result.

Using our ϵb values, we now calculate the concentrations of A, B, and C in the kinetic data; thus

```
findconc(KinData, eb)
head(pred.conc)
```

```
##         A       B       C
## 1 0.31325 0.15522 0.00292
## 2 0.31606 0.15808 0.00345
## 3 0.31788 0.15831 0.00437
## 4 0.31568 0.15610 0.00567
## 5 0.30857 0.15088 0.00884
## 6 0.29633 0.14140 0.01318
```

```
plot(time, pred.conc[ , 1], type = "l", lwd = 2, lty = 1, xlab = "time",
     ylab = "concentration", ylim = c(0, 0.35))
lines(time, pred.conc[ , 2], lwd = 2, lty = 2)
lines(time, pred.conc[ , 3], lwd = 2, lty = 3)
```



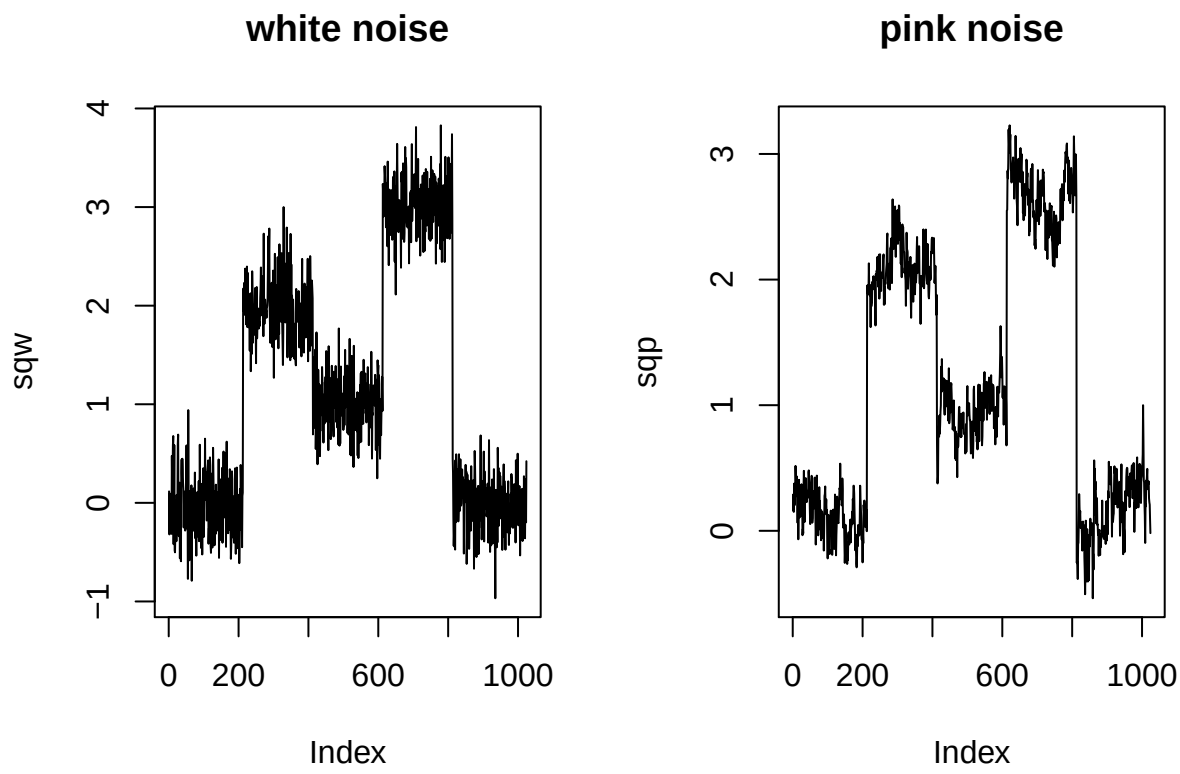
where the solid line is A, the dashed line is B, and the dotted line is C. Note that the initial increase in the concentrations of A and of B must be an artifact of the data. Note, as well, that the concentrations of A and of B decrease at a similar rate and that the concentration of C is tracking toward the initial concentration of B; both observations are consistent with the reaction's stoichiometry and the smaller concentration of B relative to that of a.

Problem 6. In class we considered several ways to improve the signal-to-noise ratio using data created by adding noise drawn from a Gaussian distribution to a pure signal. This sort of noise is known as white noise. There are other classes of noise, including pink noise, which has very different characteristics. The file squareWave.RData contains three vectors, each with a length of 1024: sq is a pure signal that is the summation of three square waves; sqw is the same data with white noise added to it; sqp is the same data with pink noise added to it.

(a) For both the white noise and the pink noise report the signal-to-noise ratio for the square wave with the greatest signal.

First, let's look at the raw data so that we can identify useful regions for evaluating the signal and the noise.

```
old.par = par(mfrow = c(1, 2))
plot(sqw, type = "l", main = "white noise")
plot(sqp, type = "l", main = "pink noise")
```




```
par(old.par)
```

Using the first 100 points for the noise and 100 points from the center of the third square wave gives the following signal-to-noise ratios for white noise and pink noise, respectively

```
sn.white1 = mean(sqw[650:749])/sd(sqw[1:100])
round(sn.white1, digits = 2)
```

```
## [1] 8.99
```

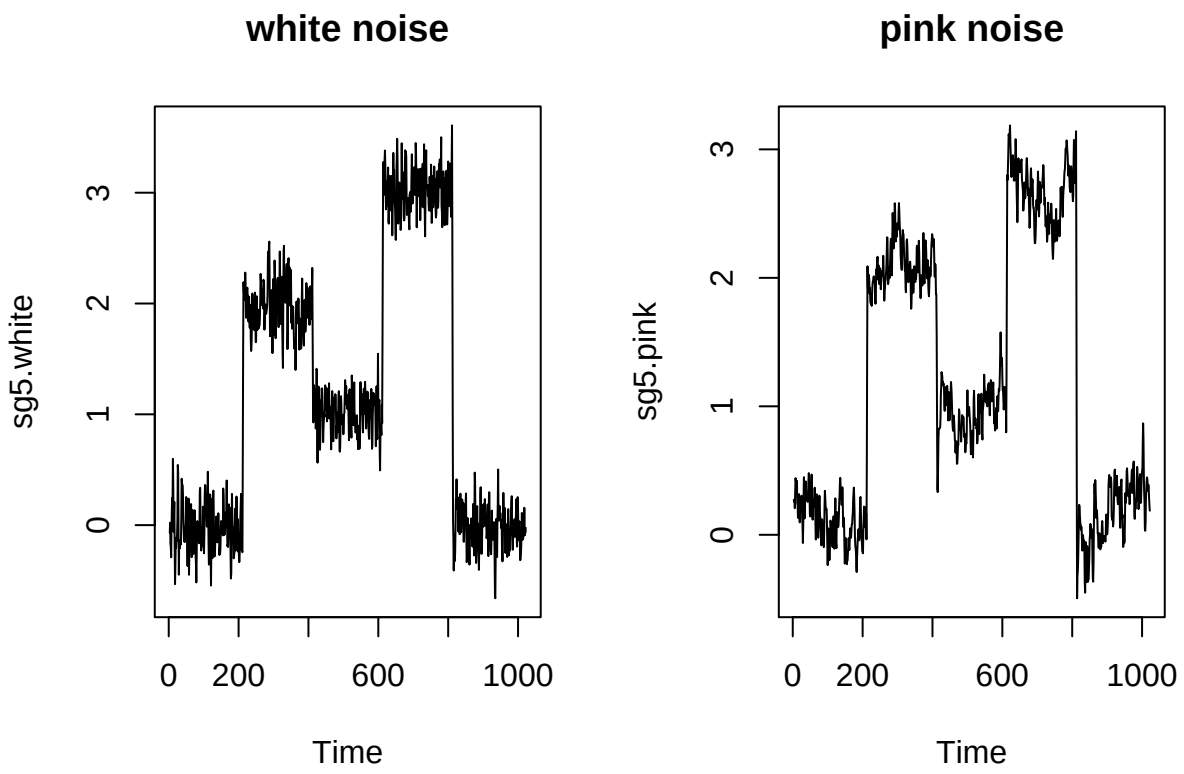
```
sn.pink1 = mean(sqp[650:749])/sd(sqp[1:100])
round(sn.pink1, digits = 2)
```

```
## [1] 15.93
```

(b) Apply a five-point Savitzky-Golay filter to both the white noise and the pink noise and reevaluate the signal-to-noise ratio.

First, let's apply the filter and examine the resulting data

```
sg5.white = filter(sqw, c(-3, 12, 17, 12, -3)/35)
sg5.pink = filter(sqp, c(-3, 12, 17, 12, -3)/35)
old.par = par(mfrow = c(1, 2))
plot(sg5.white, type = "l", main = "white noise")
plot(sg5.pink, type = "l", main = "pink noise")
```




```
par(old.par)
```

and then calculate the signal-to-noise ratios for the data with, respectively, white noise and pink noise.

```
sn.white2 = mean(sg5.white[650:749], na.rm = TRUE)/sd(sg5.white[1:100], na.rm = TRUE)
round(sn.white2, digits = 2)
```

```
## [1] 13.44
```

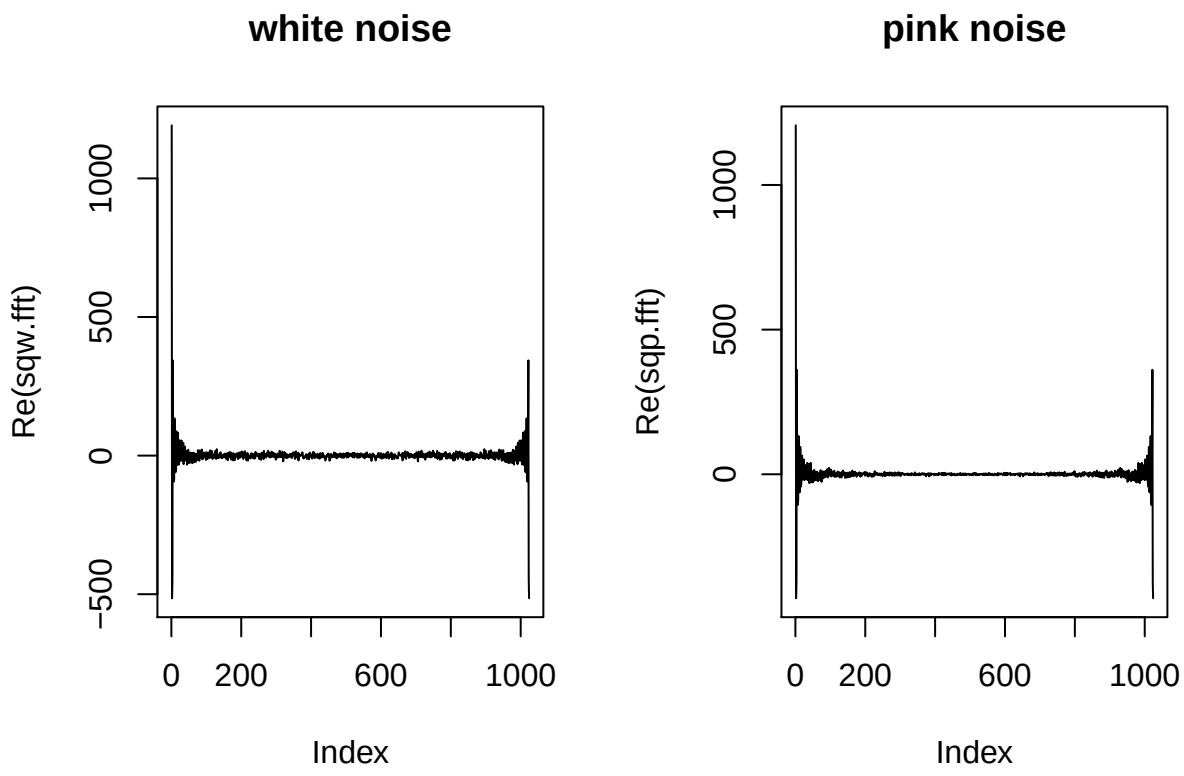
```
sn.pink2 = mean(sg5.pink[650:749], na.rm = TRUE)/sd(sg5.pink[1:100], na.rm = TRUE)
round(sn.pink2, digits = 2)
```

```
## [1] 18.06
```

(c) Apply a Fourier filter to both the white noise and the pink noise and reevaluate the signal-to-noise ratio.

First, let's complete the FFT and examine the data to identify how best to filter the noise.

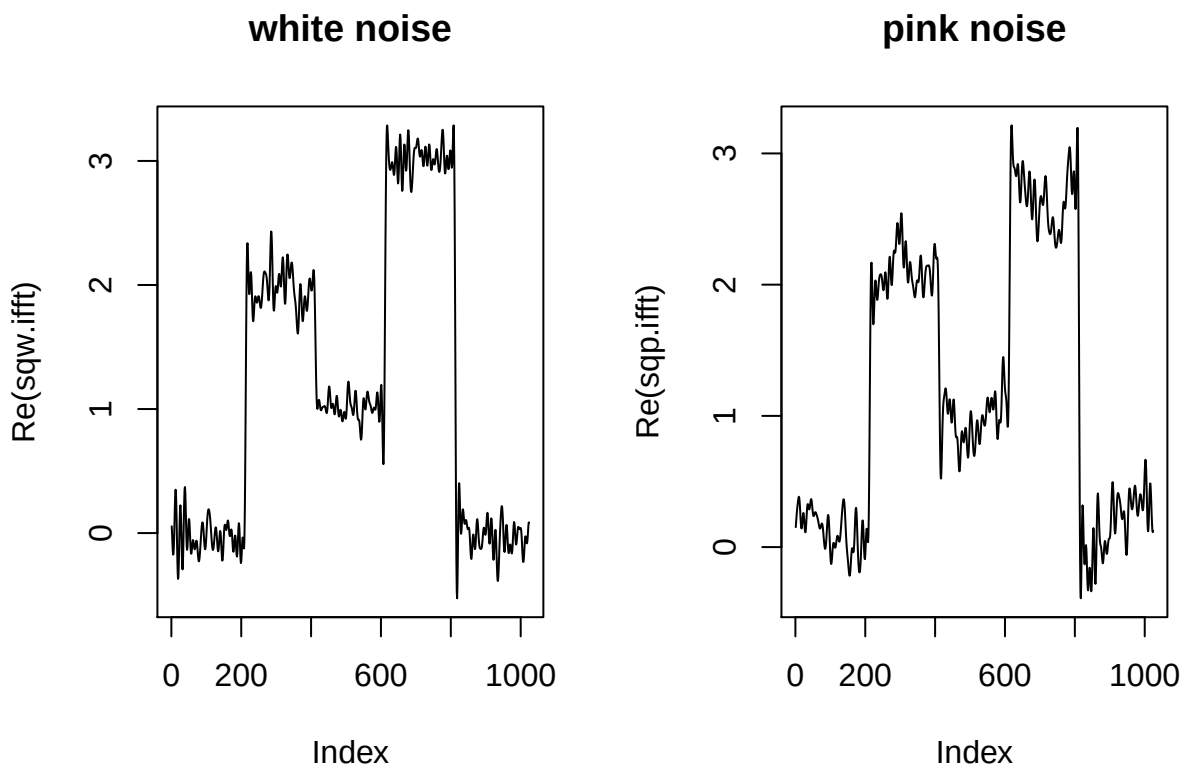
```
sqp.fft = fft(sqp)
sqw.fft = fft(sqw)
old.par = par(mfrow = c(1, 2))
plot(Re(sqw.fft), type = "l", main = "white noise")
plot(Re(sqp.fft), type = "l", main = "pink noise")
```




```
par(old.par)
```

Let's try filtering out all points from 101 through 924; that is, keeping the first and the last 100 points.

```
sqp.fft[101:924] = 0 + 0i
sqw.fft[101:924] = 0 + 0i
sqp.ifft = fft(sqp.fft, inverse = TRUE)/length(sqp.fft)
sqw.ifft = fft(sqw.fft, inverse = TRUE)/length(sqw.fft)
old.par = par(mfrow = c(1, 2))
plot(Re(sqw.ifft), type = "l", main = "white noise")
plot(Re(sqp.ifft), type = "l", main = "pink noise")
```




```
par(old.par)
```

The signal-to-noise ratios for white and for pink noise after Fourier filtering are, respectively

```
sn.white3 = mean(Re(sqw.ifft[650:749]))/sd(Re(sqw.ifft[1:100]))
round(sn.white3, digits = 2)
```

```
## [1] 18.61
```

```
sn.pink3 = mean(Re(sqp.ifft[650:749]))/sd(Re(sqp.ifft[1:100]))
round(sn.pink3, digits = 2)
```

```
## [1] 26
```

(d) Based on your work in (b) and in (c), comment on the general ability of these two approaches to filtering data for a signal that includes white noise. Repeat for a signal that includes pink noise.

Both approaches seem to do a good job of improving the signal-to-noise ratio. For the white noise, the original signal-to-noise ratio is 8.99, which improves to 13.44 when we apply the Savitzky-Golay filter and to 18.61 when we apply the Fourier filter. For the pink noise, the original signal-to-noise ratio is 15.93, which improves to 18.06 when we apply the Savitzky-Golay filter and to 26 when we apply the Fourier filter.

Note: Pink noise clearly is much more complicated than white noise in that distorts the shape of the square wave itself. As a result, depending on the range of points you selected your reported signal-to-noise ratios for the data with pink noise may differ signficantly from those reported here.