

Student	Ke Hao	30
---------	--------	----

Part 1

	Available	Marks
Generate WineSamples dataframe	5	2
Add proline categories	10	8
Add ratio of hue to colour intensity	5	4
Generate WineSummary	5	4
Add descriptive data for alcohol and phenols	10	6
Identify high/v. high proline levels	10	3
Commenting and use of white space	5	3
Comments It would have been easier to convert the xlsx file to a csv file and load it directly. <pre>> # 1.1 Load this data into a data frame wineData > wineData <- read.xlsx("WineSamples.xlsx",sheetIndex=1,header = T, encoding="UTF-8",stringsAsFactors = TRUE) Error in read.xlsx("WineSamples.xlsx", sheetIndex = 1, header = T, encoding = "UTF-8", : unused arguments (sheetIndex = 1, header = T, encoding = "UTF-8", stringsAsFactors = TRUE)</pre> Remember to make both your code and comments narrow enough to fit on a normal screen – 80-90 characters wide is good. Good use of cut() function. Descriptive statistics are fine, although it would be faster to use the aggregate() function. Q6: Section 6.1 has problems. <pre>+ level_h_vh2=level_h_vh2+1 + }else if(type[i]=="Type 3"){ + level_h_vh3=level_h_vh3+1 + } + } + }</pre> <pre>Error in count : object 'count' not found</pre> Some very good concise code in there, but it would have been good to see more of it implemented using functions.		

Part 2

	Available	Marks
Create the countryData dataframe	5	2
Create the CovidSymmary dataframe	15	12
Check the data for normality and process accordingly	10	7
Perform the appropriate t-tests	5	5
Perform the appropriate ANOVA tests	10	6
Commenting and use of white space	5	5
Comments Similar problem loading the data file. It works without the extra arguments. <pre>> # 1. Load this data into a data frame wineData > countryData <- read.xlsx("CovidData.xlsx", sheetIndex=1, header = T, encoding="UTF-8", stringsAsFactors = FALSE) Error in read.xlsx("CovidData.xlsx", sheetIndex = 1, header = T, encoding = "UTF-8", : unused arguments (sheetIndex = 1, header = T, encoding = "UTF-8", stringsAsFactors = FALSE)</pre> Good handling of the two datasets, and the method for displaying the range is very good. Your process for checking for normality is good. The t-test is good. ANOVA: The data extraction is good, but it's much better to put it in a dataframe rather than a table. There are problems with tab: <pre>Hit <Return> to see next plot: tab<-summary(fm) > k<-length(tab[[1]])-2 Error: object 'tab' not found > temp<-c(sum(tab[[1]][,1]),sum(tab[[1]][,2]),rep(NA,k)) Error: object 'tab' not found > tab[[1]][["Total"],]<-temp Error: object 'temp' not found > tab Error: object 'tab' not found > } Error: unexpected '}' in "}" > summary(tab\$fit)</pre> Post-hoc processing is good. Commenting and use of white space is good. Generally good, but more use of user-defined functions would be useful.		