Explore and Summarize Data

Requires Changes

8 SPECIFICATIONS REQUIRE CHANGES

This is a very good first submission for this project. Nice work!

When you see the number of requirements that have to be fixed, you may get the sense that there is much work to do. However, many of the changes are minor and others overlap (that is, when you fix one then another requirement will be met). The main issue to be addressed is: Reorganizing your plots and commentary. Other issues are: code commenting, and some of the plot options used. Each of these are explained in the sections below.

Note: Given that the structure of your project is sound (there are no major changes required), it will not take long to make the changes required to meet all of the requirements *(Also Suggestions **are not** required changes).*

Overall, very nice work!! Best wishes for your re-submission!

Code Functionality

✓

All code is functional (e.g. No Error is produced and RMD document is not prevented from being knit.)

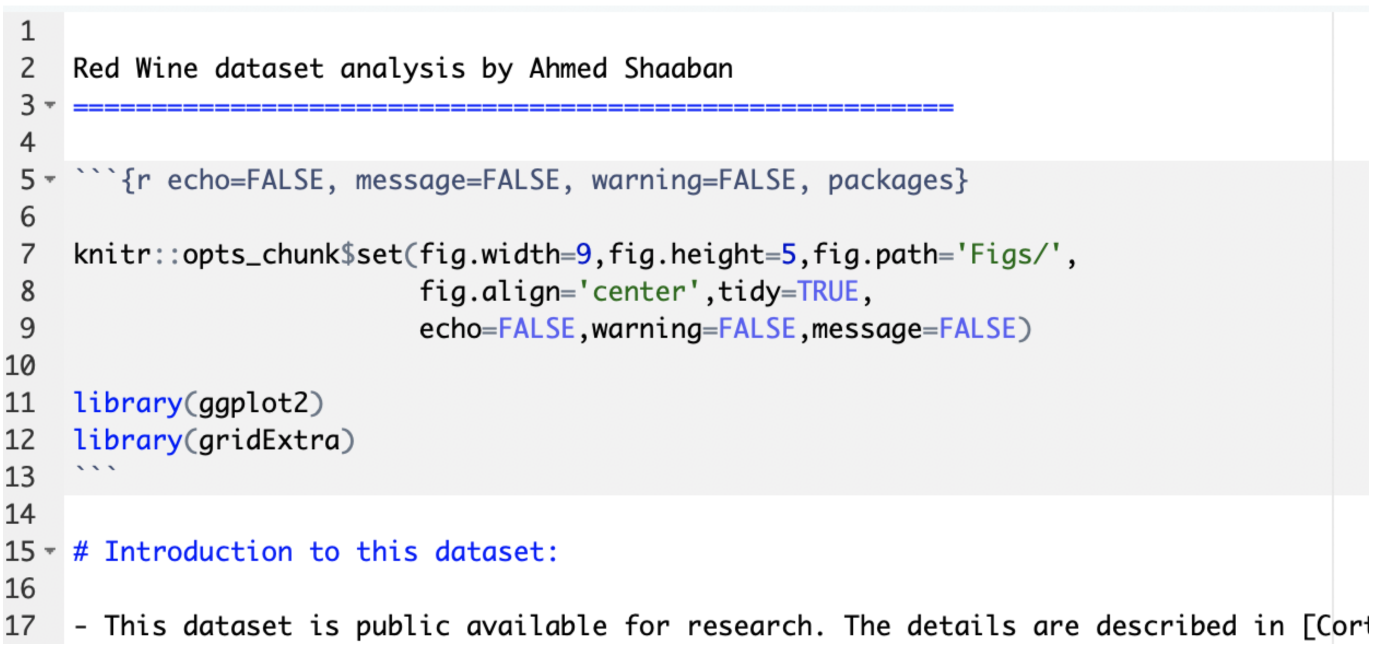
Your code knits as expected and code, errors and warnings are suppressed using chunk options. Very nice work!

Suggestion (Recommended)

You can set knitr options for the figure dimensions. You do this by adding:

```
knitr::opts_chunk$set(fig.width=9,fig.height=5,fig.path='Figs/',
                      fig.align='center',tidy=TRUE,
                      echo=FALSE,warning=FALSE,message=FALSE)
```

to the first code chunk. This sets the **global** heights and widths for the plots in your project.



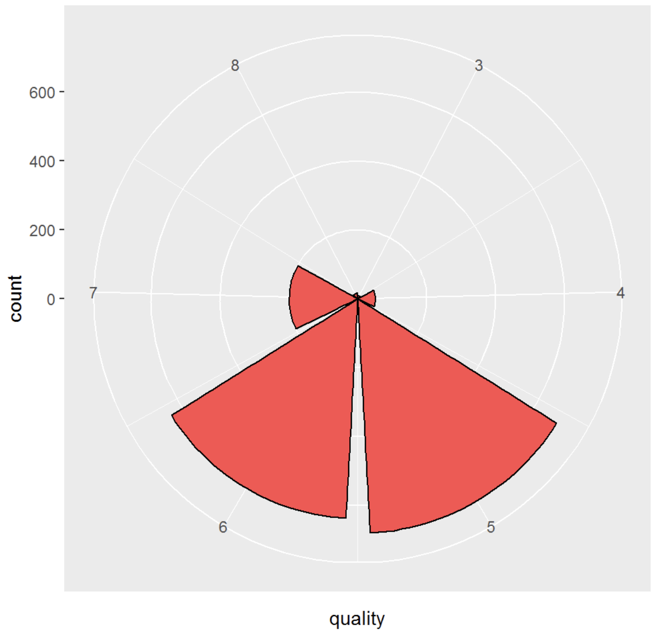
~~~

Red Wine dataset analysis by Ahmed Shaaban
==

```{r echo=FALSE, message=FALSE, warning=FALSE, packages}

knitr::opts_chunk$set(fig.width=9,fig.height=5,fig.path='Figs/',
                      fig.align='center',tidy=TRUE,
                      echo=FALSE,warning=FALSE,message=FALSE)

library(ggplot2)
library(gridExtra)
~~~

# Introduction to this dataset:

- This dataset is public available for research. The details are described in [Cort
```

This approach makes your plot sizes uniform, page-width and centered.

✓

The project almost never uses repetitive code where a function would be more appropriate. The code references variables by name instead of using constants or column numbers.

Suggestion

There is some repetition of code. You may benefit from developing functions for those blocks. If you want to explore the option of developing your own functions, this post is a good place to start

Project Readability

🔄

Markdown syntax is used in the RMD file to improve readability of the knitted file.

Very Minor Changes

1.

If you don't leave *an empty line* between text and code chunks for plots then the text will wrap with the plot:
```

dioxide and total sulfur dioxide, except pH and density are asymmetrically distributed

Display the quailty grades

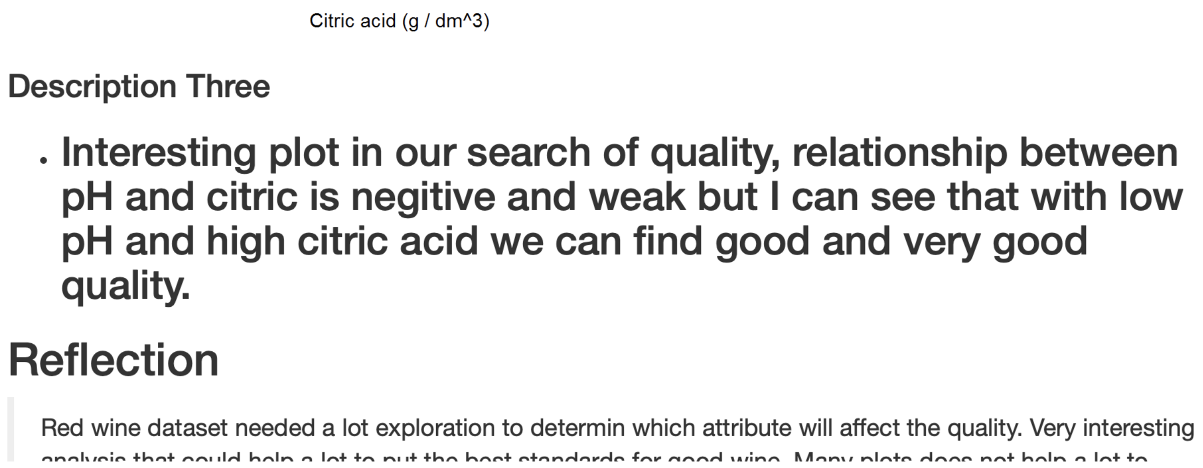


We can

see that most of the quality rating given were 5 and 6

## 2.

Also, you get odd formatting if there is not an empty line between page divider syntax and text:



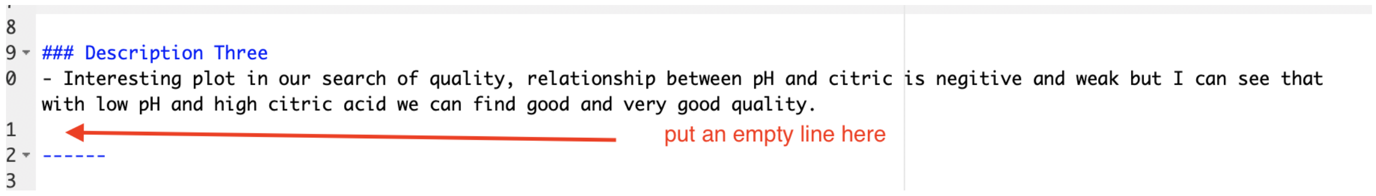
Citric acid (g / dm^3)

### Description Three

- Interesting plot in our search of quality, relationship between pH and citric is negitive and weak but I can see that with low pH and high citric acid we can find good and very good quality.

## Reflection

> Red wine dataset needed a lot exploration to determin which attribute will affect the quality. Very interesting



```
### Description Three
- Interesting plot in our search of quality, relationship between pH and citric is negitive and weak but I can see that
with low pH and high citric acid we can find good and very good quality.

------
```

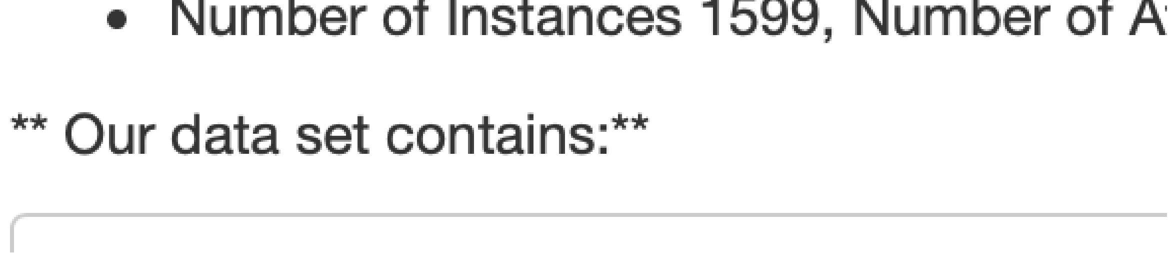
put an empty line here

3.

Finally, this is particular to *R Markdown*, if you use *Markdown* syntax and leave a space between the *Markdown* character and the text then the *Markdown* character is printed as regular text. That is:

```
** Our data set contains:**
```

prints as:

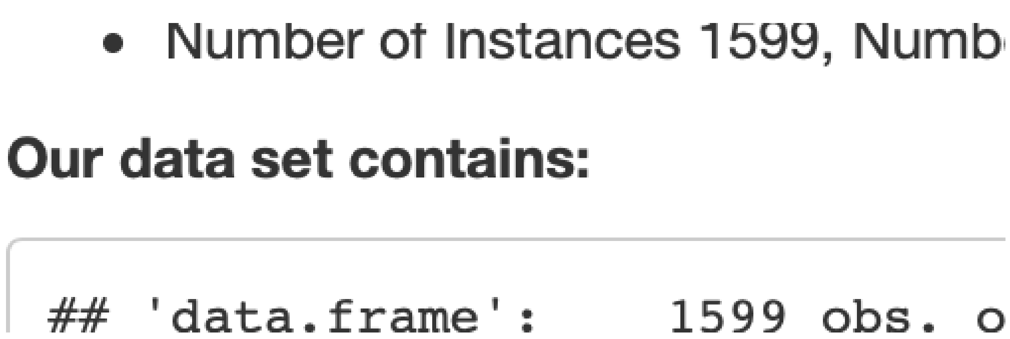

- Number of Instances 1599, Number of A

** Our data set contains:**

whereas:

```
**Our data set contains:**
```

prints as:


- Number of Instances 1599, Numb

Our data set contains:

```
## 'data.frame':    1599 obs. o
```

All complex code is adequately explained with comments. It is always clear what the code is doing and how and why any unusual coding decisions were made.

Important Change

There is very little commenting of code.

Commenting code is a requirement for this project, and a good habit to develop, because it communicates to colleagues, or to yourself at some future time, the intent of the code that you have written (in a succinct way that avoids you having to examine the code line by line).

It is difficult to overstate the importance of clear code commenting, in a work environment, in data analysis.

To fix this, it is best to write brief comments for each chunk of code, and more detailed comments for complex code.

For example:

```
# Bar Chart in Polar Coordinates - Quality
ggplot(pf, aes(quality))+
  geom_bar(color = 'black', fill = '#ff4d4d')+
  coord_polar()
```

It takes about 5 minutes to comment a script like this, but it saves much more time for a colleague when they are reading your code.

The code uses formatting techniques in a consistent and effective manner to improve code readability. All lines are shorter than 80 characters.

Very Minor Changes

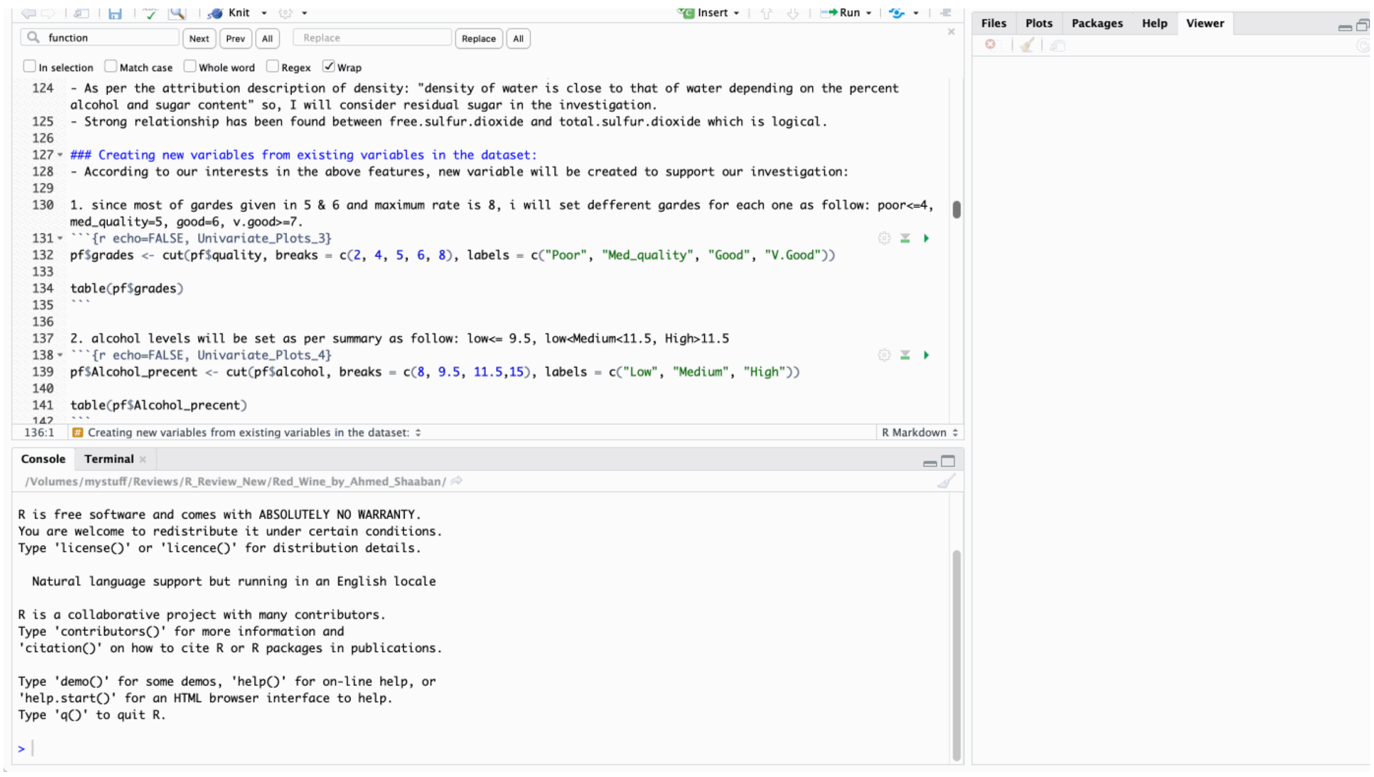
Some of your lines of code go over the required *80 characters* limit.

The reason for this limit? *Readability of Code.*

The optimal number of characters in a line in a book is about 60 characters. Shorter than that, it is difficult to read because you are skipping from line to line. Longer than that, it is difficult to read because your eye *"loses the start of the line"*, which means you have to search for it to move to the next line.

Parsing code is difficult enough, especially when reading other people's, so standards are routinely followed to make the process easier.

R has a handy tool that helps you to fix this. That is, you can add *guides* that mark where *80 characters* end. You do this by going to `Global Options`, which is under the `Tools` menu option. Then select `Code` and click the `Display` tab, as follows:



~~~
- As per the attribution description of density: "density of water is close to that of water depending on the percent
alcohol and sugar content" so, I will consider residual sugar in the investigation.
- Strong relationship has been found between free.sulfur.dioxide and total.sulfur.dioxide which is logical.

### Creating new variables from existing variables in the dataset:
- According to our interests in the above features, new variable will be created to support our investigation:

1. since most of gardes given in 5 & 6 and maximum rate is 8, i will set defferent gardes for each one as follow: poor<=4,
med_quality=5, good=6, v.good>=7.
```{r echo=FALSE, Univariate_Plots_3}
pf$grades <- cut(pf$quality, breaks = c(2, 4, 5, 6, 8), labels = c("Poor", "Med_quality", "Good", "V.Good"))

table(pf$grades)
~~~

2. alcohol levels will be set as per summary as follow: low<= 9.5, low<Medium<11.5, High>11.5
```{r echo=FALSE, Univariate_Plots_4}
pf$Alcohol_precent <- cut(pf$alcohol, breaks = c(8, 9.5, 11.5,15), labels = c("Low", "Medium", "High"))

table(pf$Alcohol_precent)
```
```

```
/Volumes/mystuff/Reviews/R_Review_New/Red_Wine_by_Ahmed_Shaaban/

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

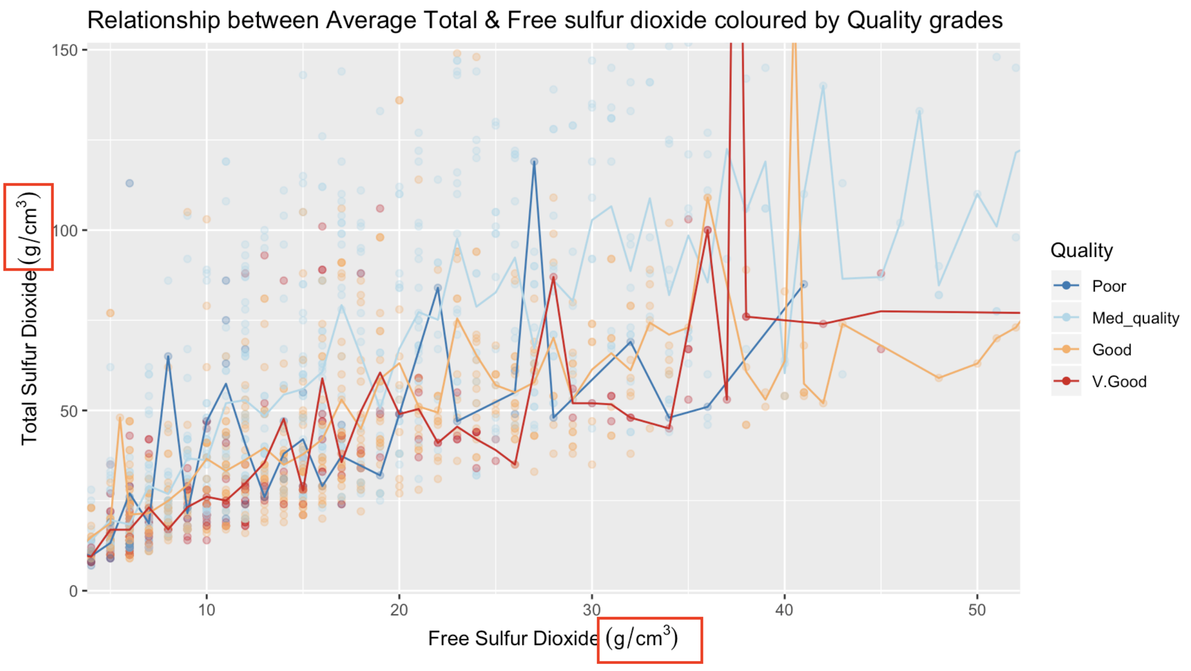
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

(Click images to enlarge them)

To *wrap* the code, put the cursor *after the* `+` *symbol* (or a *comma*) in a line and press *Return*.

Final Plots and Summary

✓

All plots are labeled appropriately (axis labels, plot titles, axis units) and can be read and interpreted easily. Plots are scaled appropriately.

Suggestion

If you wish, for labels that contains mathematical symbols, you can use the `expression()` method (which is similar to *LaTex*, if you are familiar with that):

```
+
  xlab(expression(Free~Sulfur~Dioxide~(g/cm^{3}))) +
```
```

```
  ylab(expression(Total~Sulfur~Dioxide~(g/cm^{3})))
```

where `~` is a space and `^{}` denotes a superscript and `[]` denotes a subscript.



Omitting 1% of top total & free sulfur.dioxide

✓

The project includes a Final Plots and Summary section containing three plots and commentary. All plots in this section reflect what has been explored in the main body of the analysis.

Again, excellent plot selection.

Note

As discussed earlier, ensure that you include these plots in **both** this *Final Plots* section **and** the appropriate section earlier in your report.

✓

The plots are well chosen and the plots fulfill at least 2 of the criteria. The plots are varied and reveal interesting trends and relationships.

✓

All plots have appropriately selected variables and are plotted in a way that accurately conveys the data/information (i.e findings in Final Plot 1 do not depend on the findings of Final Plot 2).

✓

The reasoning and findings from each plot are explained and the text about each plot is descriptive enough to stand alone. Comments reflect the contents of the plots that they are associated with.

Quality of Analysis

Questions and findings are placed between blocks of R code regularly so it is clear what the student was thinking throughout the analysis.

Minor Change

As noted, there are sequence of plots, or composite visualizations, without individual commentary for individual plots.

Reasoning is provided for the plots made throughout the analysis. Plots made follow a logical flow. Comments following plots accurately reflect the plots' contents.

Minor (but important) Change

It is difficult to follow the flow of your exploration because either: The commentary relates to a composite visualization, so the reader has to search the composite visualization for the relevant plot; or, The commentary follow a sequence of plots, so the reader has to scroll back and forth between commentary and the relevant plot.

Obviously, this is resolved by re-organizing your plots and commentary.

The project contains at least 20 visualizations. The visualizations are varied and show multiple comparisons and trends. Relevant statistics (e.g. mean, median, confidence intervals, correlations) are computed throughout the analysis when an inference is made about the data.

Minor Change

Currently there are 12 unique visualizations in your report:

1. Composite plots count as one visualization, and
2. Plots in the *Final Plots* section don't count (because they are supposed to be finalize versions of plots from earlier sections).

This issue will be easy to resolve:

1. In the *Univariate* plots section, remove the more interesting plots from the composite plot and plot them individual (with commentary), and
2. Include the plots in the *Final Plots* section in **both** the *Final Plots* section **and** the relevant section that they are appropriate for.

That will easily get you over the 20 visualization limit.

✓

The project appropriately uses univariate, bivariate, and multivariate plots to explore most of the expected relationships in the data set.

Your plot selection is excellent.

However, as noted below, the requirements for this *data exploration* are that you explain the data to the reader *as you are exploring it*. That is, instead of using composite plots, or plotting multiple plots in a row, **and then** adding commentary, the requirements specify that you should provide commentary for all individual plots that you find patterns in (using composite plots are fine, if the variables are not central to your exploration).

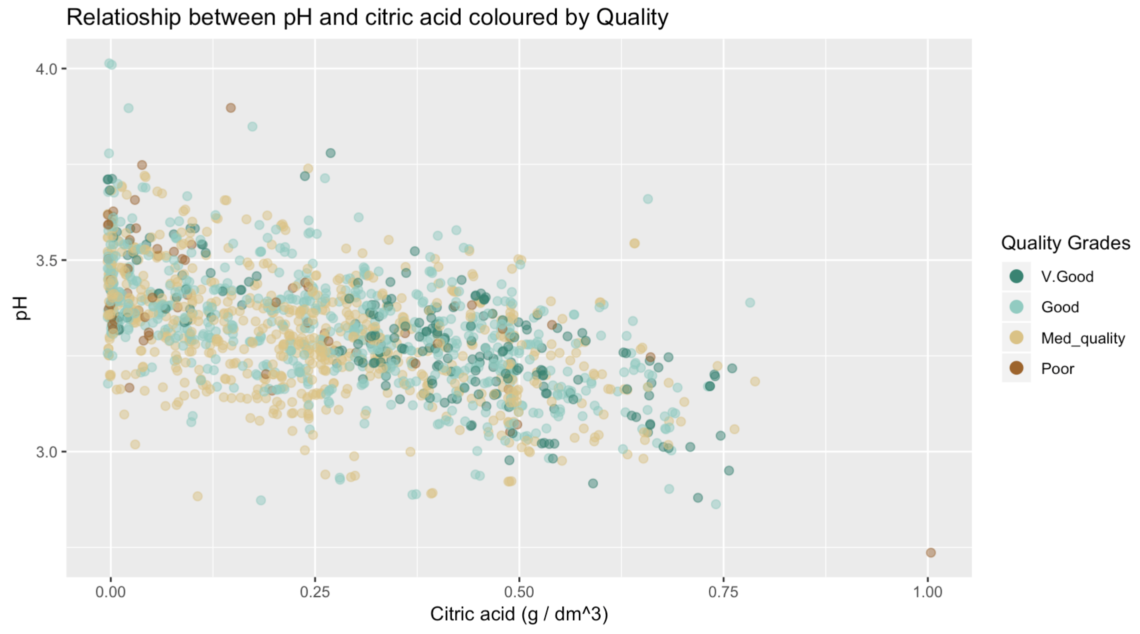
Visualizations made in the project depict the data in an appropriate manner that allows plots to be readily interpreted. Choice of plot type, variables, and aesthetic parameters (e.g. bin width, color, axis breaks) is appropriate.

As noted earlier, your plot selection is excellent. However, there as some issues with plot options that should be resolved:

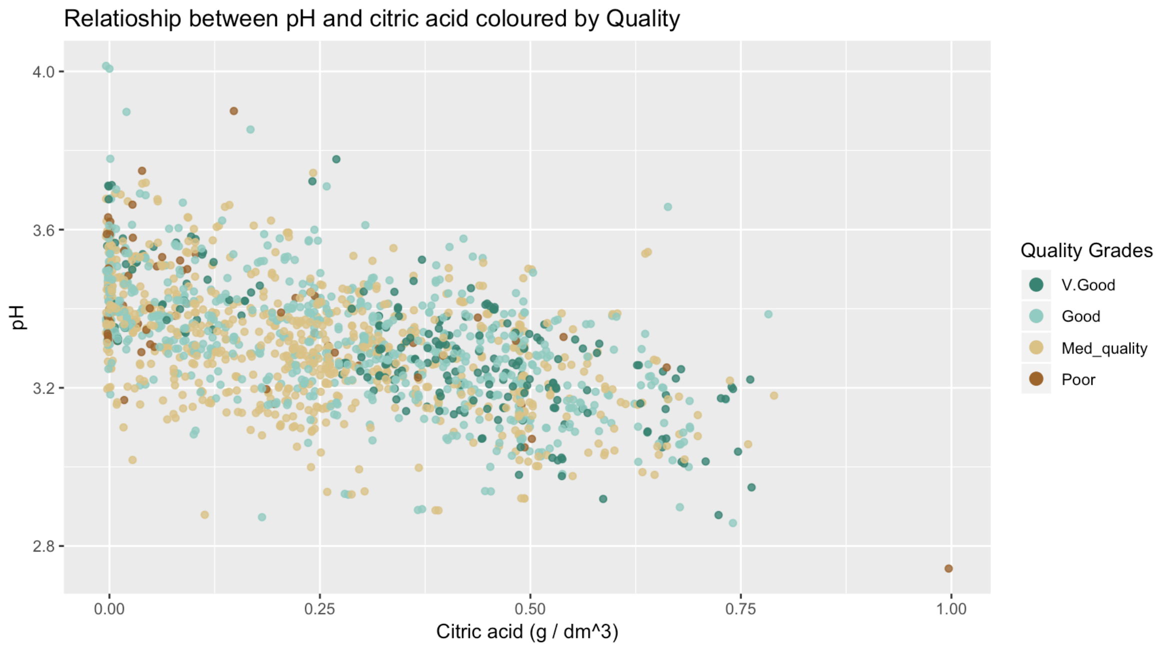
1. `size` option

Depending on the number of observations in the data, increasing the point size can lead to *overplotting* (points plotting on top of each other). That is the case here:

Plot Three



By removing the `size` option, and just using the `ggplot()` default point size, it allows you to use a higher level of `alpha` (because the points don't overlap, so you don't need a high level of transparency):



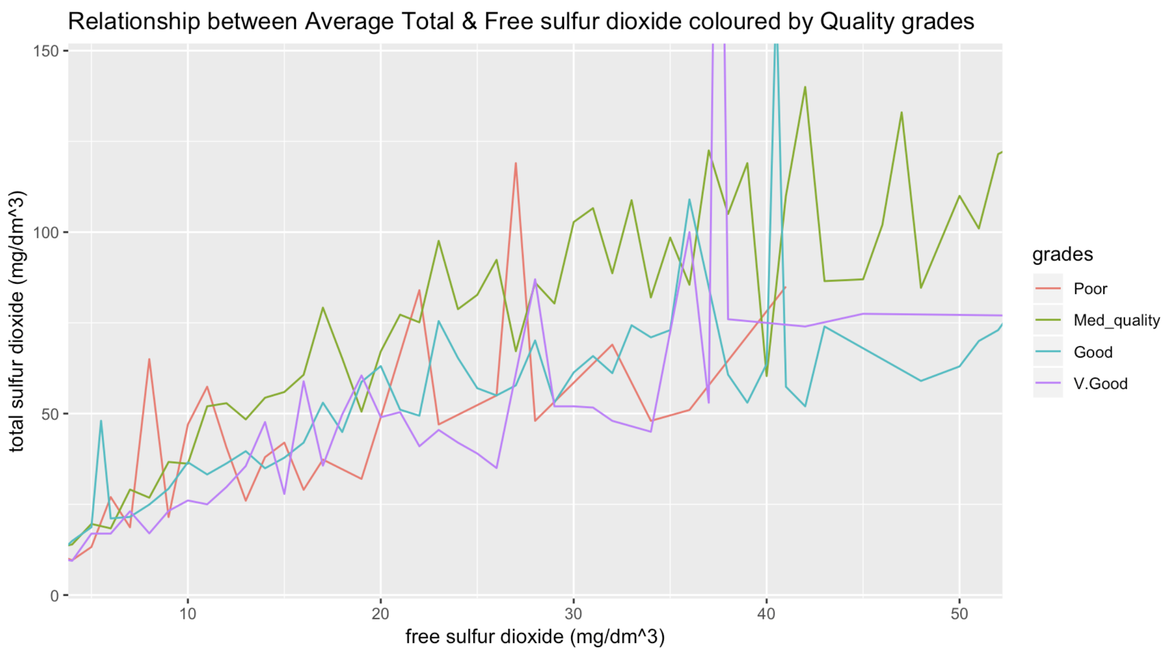
So, you get more clearly defined points with less overlapping.

2. Line Graphs

Line graphs are typically used for trends (in time series data) or models. The common feature with those types of plots is that *there is some underlying connection between the points that define the lines.*

When using line graphs to represent summary statistics, it is best to use them overlaid on the actual data - to give the reader context. So for this plot:

Plot One

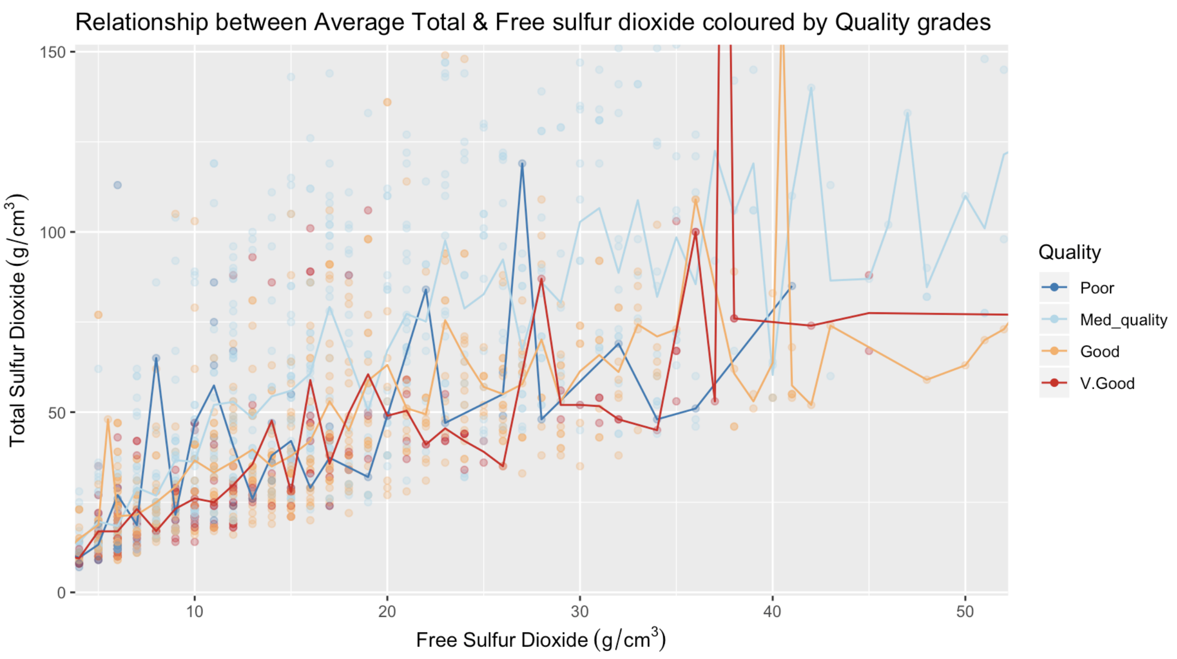


Omitting 1% of top total & free sulfur.dioxide

you would simply add the layer:

```
geom_point(alpha=0.3) +
```

before the layer for the means (so that the lines are on top of the points) - *where the high level of transparency is used to keep the focus on the mean lines.*



Omitting 1% of top total & free sulfur.dioxide

This allows the reader to see that the mean lines, for high values of *Free Sulfur Dioxide*, are actually just the only observation for that value - so that the mean equals that single value *(where the mean lines represent: The mean level of Total Sulfur Dioxide, for each value of Free Sulfur Dioxide, grouped by Quality)*

For each of the line graphs in your report, which all represent summary statistics, you should overlay them on the actual data.

---

3. Color Encoding

Color Encoding is a helpful, and important, tool for uncovering patterns in data. Using the correct color encoding for the data type can often help uncover a pattern that would otherwise not be very obvious. You did this for the final plot in the *Final Plots* section.

As you can see from the two plots above, the color encoding has been changed.

For the first plot, you have encoded the quality variable as discrete, which is correct. Nice work with that.

However, because quality is ordinal, you have to tell `ggplot()` to use *sequential* or *divergent* color encoding (this is optimal because it gives a sense of gradation to the different levels in the data) - and not *qualitative* color encoding (which is for general discrete variables), as is done in your code.

Where *sequential* color encoding is used for pure ordinal discrete data and *divergent* color encoding is used if the data is both ordinal and follows a diverging scale (think *"Good, Ok, Bad"* - which can be viewed as appropriate for this dataset). For example, for the plot above, you would use:

```
    scale_color_brewer(type = "div", palette = "RdYlBu", name="Quality", direction=-1) +
```

(where the `name` option changes the legend title and `direction=-1` changes the order of the colors) for example:

```
ggplot(pf, aes(free.sulfur.dioxide, total.sulfur.dioxide, color = grades))+
  geom_point(alpha=0.3) +
  geom_line(stat = 'summary', fun.y = mean)+
  coord_cartesian(xlim = c(min(pf$total.sulfur.dioxide),
                           quantile(pf$free.sulfur.dioxide, 0.99)),
                  ylim = c(min(pf$total.sulfur.dioxide),
                           quantile(pf$total.sulfur.dioxide, 0.99)))+
  ggtitle('Relationship between Average Total & Free sulfur dioxide coloured by Quality grade
s')+
  scale_color_brewer(type = "div", palette = "RdYlBu", name="Quality",
                     direction=-1) +
  guides(color=guide_legend(override.aes = list(alpha = 1)))
```

(where the `guides()` method controls the legend and `override.aes` over-rides the plots aesthetics (in this case, the `alpha` value)). which results in the second plot above. **Which more clearly highlights the difference between good and bad wines (because a neutral color is used for the *OK* wines) and the levels in *quality* are highlighted by the gradation in color.**

See here for palette options

Reflection

The section reflects on how the analysis was conducted and reports on the struggles and successes throughout the analysis. The section provides at least one idea or question for future work. The section explains any important decisions in the analysis and how those decisions affected the analysis.

Minor Change

The Reflection section is intended to be, in part, a discussion of your *personal experiences* working on this project (difficulty, or not, of learning *R*, of understand the data, of understanding design principles, etc.). It is not intended to be solely a summary of your *analysis* (which the bulk of yours is).

Also, the idea for future work needs to be an outline of a specific aspect of the topic that you would like to expand on.

✓

The project includes a Reflection section discussing the analysis performed.

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH