STT 301: Take-home exam

Shawn Santo

October 10, 2018

Take-home exam is due Monday, October 15 at 12:40pm EST.

Required packages: `ggplot2`

Instructions and Rubric

- Instructions
  - submit only your .Rmd file in the dropbox folder on D2L by the date and time above
  - formatting is at your discretion
  - you must work alone; do not communicate with your classmates or any other human; do not post code on Slack (only questions about directions should be posted)
  - questions about directions and understanding a question are okay, but do not ask me for help with your code or to give you the answer
  - you may use your notes, book, the R help, Google, etc
- Rubric
  - this part of the exam is worth 60 points
  - partial credit is possible on all parts
  - late submissions will not be accepted
  - commenting on code, output, or plots is not required unless specified in the question; however, it may help you receive partial credit if I can better understand your thought process

```
# load required package
library(ggplot2)
```

Data

The data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars (origins). The analysis determined the quantities of 13 constituents found in each of the three types of wines. The data is from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/wine).

The data set has 14 columns.

1. Origin
2. Alcohol
3. Malic acid
4. Ash
5. Alcalinity of ash
6. Magnesium
7. Total phenols
8. Flavanoids
9. Nonflavanoid phenols
10. Proanthocyanins
11. Color intensity
12. Hue
13. OD280/OD315 of diluted wines
14. Proline

Copy the below code into an R chunk in your .Rmd file.

```
# read in the wine data
URL <- "https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data"
wine <- read.csv(URL, stringsAsFactors = F, header = F)

# assign column names
colnames(wine) <- c("origin", "alcohol", "acid",
                    "ash", "alcalinity", "magnesium",
                    "phenols", "flavanoids", "nonflavanoid",
                    "proanthocyanins", "color.int", "hue",
                    "od", "proline")

# change origin variable to a factor
wine$origin <- as.factor(wine$origin)

str(wine)
```

```
'data.frame':   178 obs. of  14 variables:
 $ origin         : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
 $ alcohol        : num  14.2 13.2 13.2 14.4 13.2 ...
 $ acid           : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
 $ ash            : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
 $ alcalinity     : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
 $ magnesium      : int  127 100 101 113 118 112 96 121 97 98 ...
 $ phenols        : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
 $ flavanoids     : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
 $ nonflavanoid   : num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
 $ proanthocyanins: num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
 $ color.int      : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
 $ hue            : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
 $ od             : num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
 $ proline        : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
```

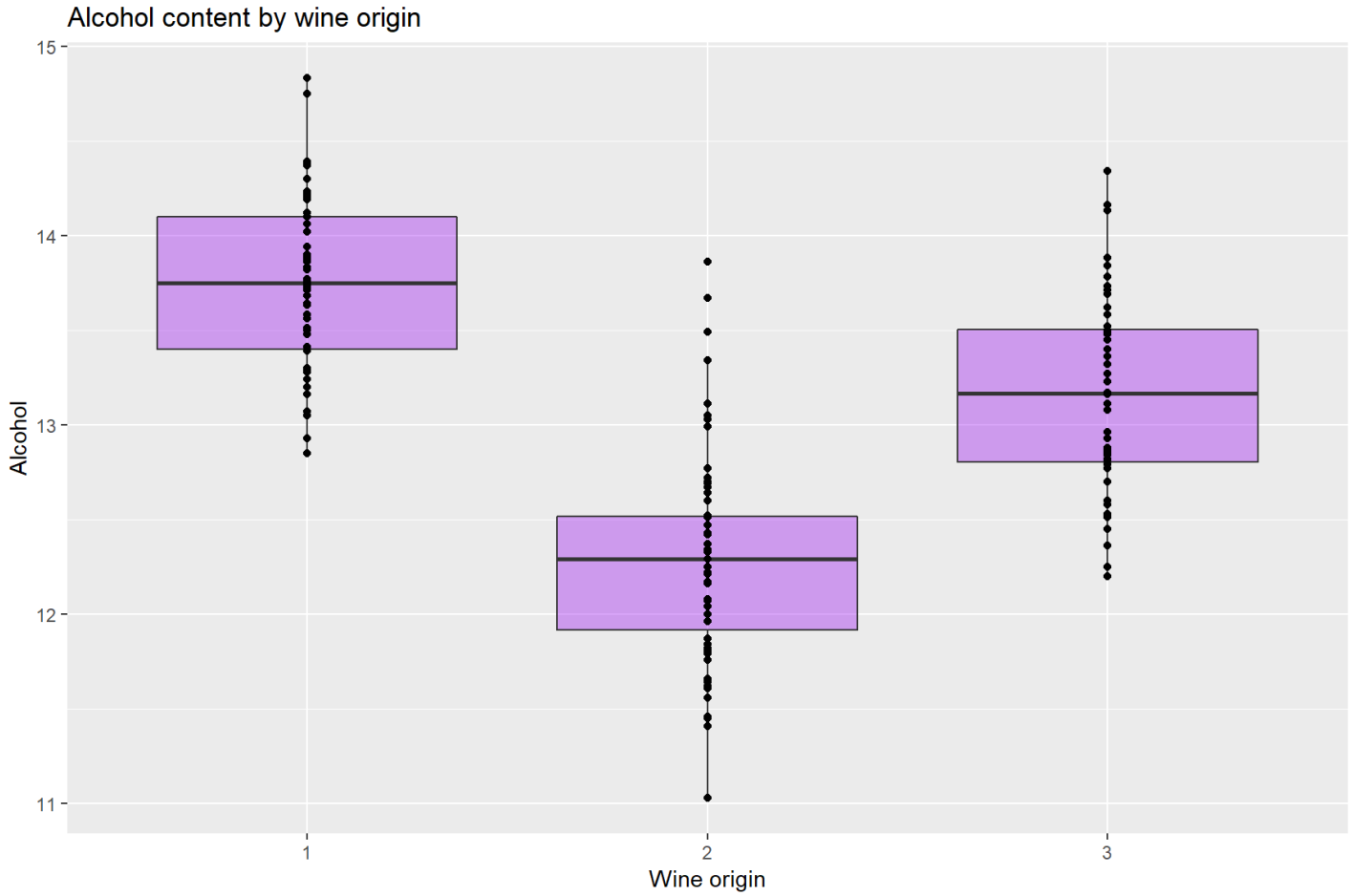
Question 1

Use the `wine` data frame to produce each of the plots below using `ggplot`. Your plot must look exactly the same to earn full credit. Details are provided when not obvious.

Plot 1 (10 points)

Details:

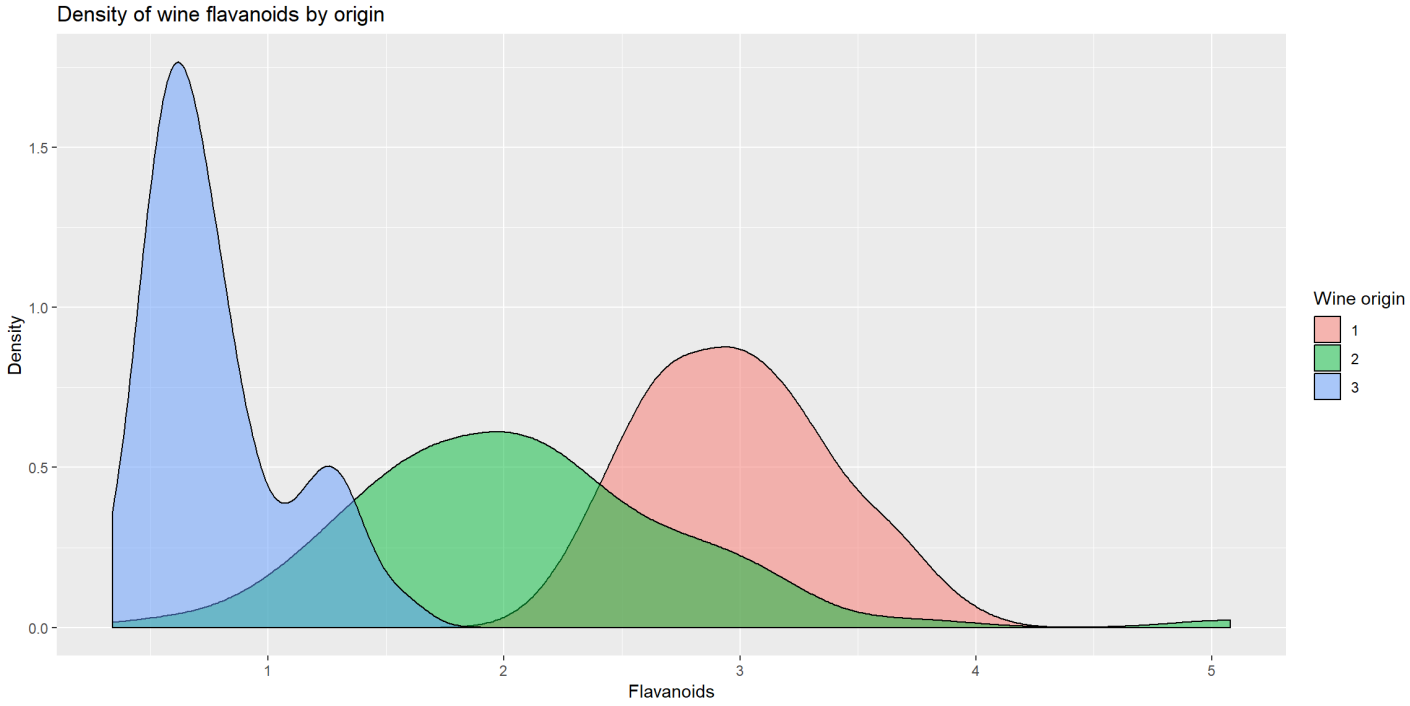
- include the chunk option: `fig.width=9, fig.height=6`
- color of boxes is purple
- transparancy is 0.4



Plot 2 (10 points)

Details:

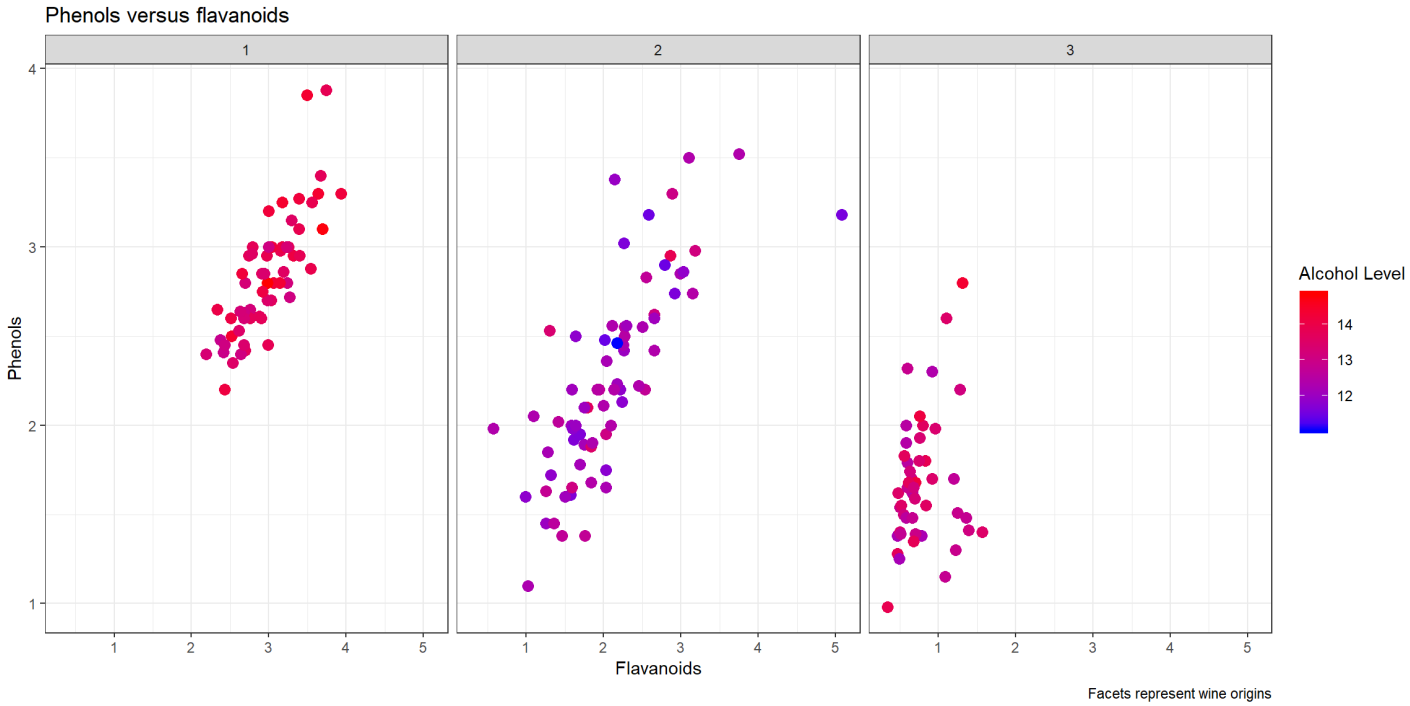
- include the chunk option: `fig.width=12, fig.height=6`
- transparancy is 0.5



Plot 3 (10 points)

Details:

- include the chunk option: `fig.width=12, fig.height=6`
- size is 3
- to change color scale: `scale_color_gradient(low = "blue", high = "red")`
- theme is bw



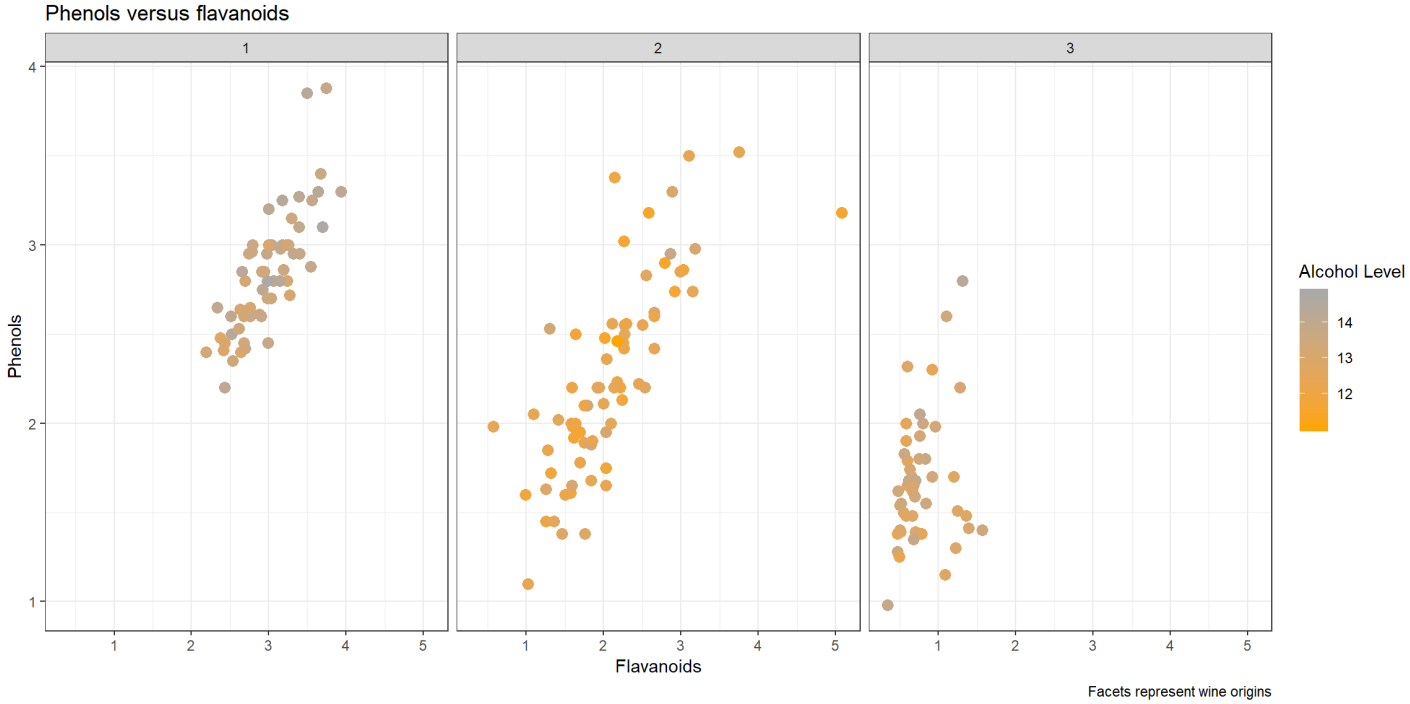
Plot 4 (10 points)

Use `wine` to create an original plot with `ggplot`. You may use additional packages. If you do use additional packages, be sure to load them with the library function in the chunk where you have `library(ggplot2)`.

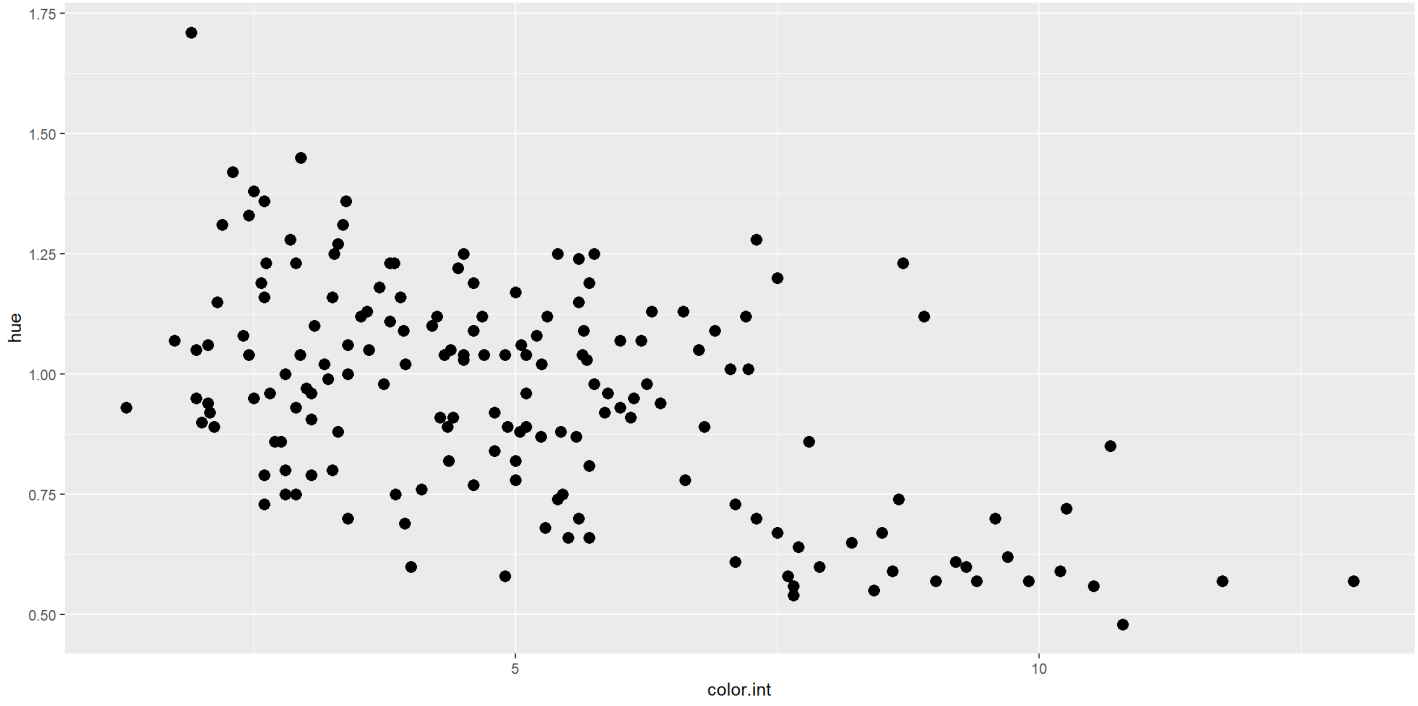
For some inspiration see the Top 50 ggplot2 Visualizations (http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html). To earn full credit, your plot cannot be a simple extension of the three plots above. It should contain at least three variables, and it should be well labeled and easy to understand.

Below are just two examples of plots that will earn 0 points.

Bad plot 1



Bad plot 2



Question 2 (20 points)

Write a function named `df.numeric.summary` that takes any data frame as an input, and for each numeric variable in the data frame, the function returns the variable's mean, median, and inter-quartile range in the form of a list. The list should have the names of the corresponding variables. Below are two examples for you to see the function in action.

Hint: `quantile` *will compute the quantile for a vector of data given a probability.*

```
df.numeric.summary(mtcars[, 1:4])
```

```
$mpg
    Mean   Median      IQR
20.09062 19.20000  7.37500

$cyl
  Mean Median    IQR
6.1875 6.0000 4.0000

$disp
    Mean   Median      IQR
230.7219 196.3000 205.1750

$hp
    Mean   Median      IQR
146.6875 123.0000  83.5000
```

```
df.numeric.summary(wine[, 1:4])
```

```
$alcohol
    Mean   Median      IQR
13.00062 13.05000  1.31500

$acid
    Mean   Median      IQR
2.336348 1.865000 1.480000

$ash
    Mean   Median      IQR
2.366517 2.360000 0.347500
```