36-613 Homework 3, Fall 2022

[Your Name Here] Style Guide: [insert style guide here]

Due Wednesday, Sept 21st, 2022 (11:59 PM EDT) on Gradescope

Contents

Problem 1: Describing distributions with histograms and density plots (22 points)

In this homework, we will work with the Guardian's list of 1000 Songs to Hear Before You Die. The data is available here. Here's the code to load in the dataset, as well as fix an issue with the `YEAR` variable that contains `,` in some numbers...

```
library(tidyverse)
# Read in the data
songs <-
  read_csv("https://raw.githubusercontent.com/ryurko/DataViz-36613-Fall22/main/data/1000songs.csv") %>%
  # Clean the year variable:
  mutate(YEAR = as.numeric(str_remove_all(YEAR, ",")))
```

a. (6pts) Using `geom_histogram()`, create a histogram of `YEAR`, where you specify a value for `binwidth` that you think is appropriate (in lecture we changed the number of bins, but for this problem you'll specify the `binwidth` instead). Defend your choice in 1-2 sentences. Make sure your histogram has appropriate titles/labels and a non-default color. Then, describe the marginal distribution of `YEAR` in 1-3 sentences.

```
# PUT YOUR CODE HERE
```

[PUT YOUR ANSWER HERE]

b. (6pts) In this part, make the exact same graph you made in Part A, but with these changes:

- Add a smoothed density to the histogram using `geom_density()`. Be sure to add `aes(y = after_stat(density))` within `geom_histogram()`.
- Add a vertical line at the mean of `YEAR` using `geom_vline(aes(xintercept = mean(YEAR))`.
- Make sure that the histogram bars `fill`, smoothed density line, and vertical line all have different colors.
- Make sure the title and labels are appropriate.

After you've made your plot, answer the following question. If you tried to just use `geom_density()` *without* adding `aes(y = after_stat(density))` within `geom_histogram()`, you wouldn't be able to see the density curve overlayed on top of the histogram. Why is that? Explain in 1-2 sentences. (**Hint**: Try getting rid of `aes(y = after_stat(density))` and see what happens.)

```
# PUT YOUR CODE HERE
```

[PUT YOUR ANSWER HERE]

c. (6pts) Now copy-and-paste your Part B code here. Then, add one of the following to your `ggplot` code:

- `facet_wrap(~ THEME)`
- `facet_grid(~ THEME)`
- `facet_grid(THEME ~ .)`

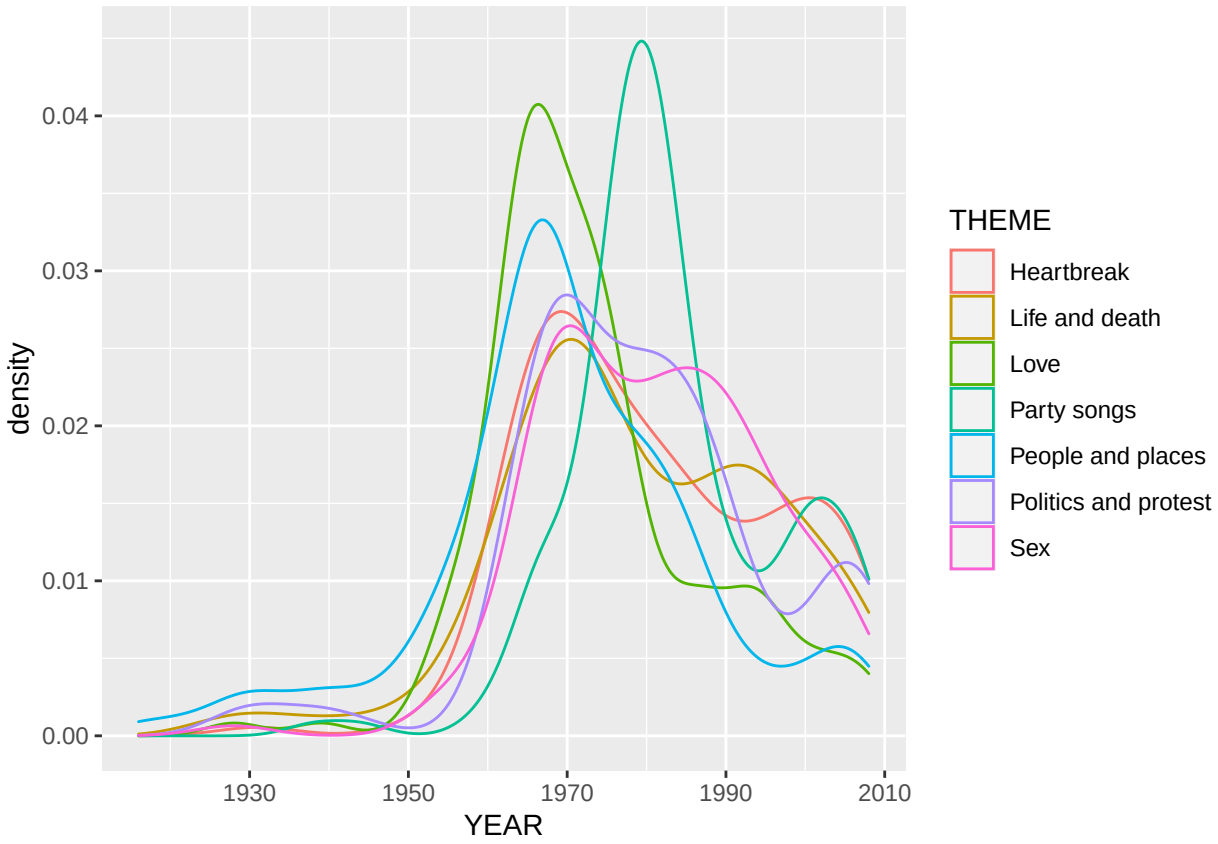
Choose the specification of `facet_wrap()` or `facet_grid()` that you believe gives you the best ability to compare these conditional distributions; mention why you made your choice in one sentence. **You don't need vertical lines in this plot, so delete the piece of your code adding vertical lines, but be sure that density curves are still added to your plot.** Then, compare and contrast the different conditional distributions of `YEAR` given `THEME` in 1-4 sentences.

```
# PUT YOUR CODE HERE
```

[PUT YOUR ANSWER HERE]

d. (4pts) Let's say that Korg is taking 36-613, and he excitedly makes the following plot after listening to a 36-613 lecture:

```
songs %>%
  ggplot(aes(x = YEAR, color = THEME)) +
  geom_density()
```



Korg makes the following argument for why he thinks this is a good plot:

"If you think about it, having histograms *and* smoothed densities is just a waste of data-ink: we really just need the smoothed density to get an idea of what the distribution is. Also, I've heard that overlaying smoothed densities is a great way to compare distributions, so why not just do that? I'd say my plot is better than the Part C plot for comparing different `YEAR` distributions, because it puts everything in one plot, rather than seven different plots."

Do you agree or disagree with Korg's argument? After saying whether you agree or disagree, explain in 1-3 sentences.

[**PUT YOUR ANSWER HERE**]

Problem 2: Automatic histograms (22 points)

As you may have seen in Problem 1A, it can be tedious to figure out what the best number of `bins` or `binwidth` is for a histogram.

Fortunately, there are many different "rules" for either choosing the number of bins k or the binwidth h. We will consider the following rules:

- Sturges' formula (which gives you the number of bins)
- The Rice rule (which gives you the number of bins)
- Scott's rule (which gives you the binwidth)
- Freedman-Diaconis' rule (which gives you the binwidth)

All of these rules are described on the Wikipedia page linked above. In fact, the `hist()` function (how you can make a histogram in `R` without `ggplot()`) already has the first, third, and fourth rules available.

Unfortunately, these rules are *not* readily available in `ggplot`; however, they can be readily coded up, which is what you'll have to do in this problem.

a. (12pts) In this part, write four functions:

- `get_sturges_bin_num(x)`: Returns the number of bins according to Sturges' rule.
- `get_rice_bin_num(x)`: Returns the number of bins according to the Rice rule.
- `get_scotts_binwidth(x)`: Returns the binwidth according to Scott's rule.
- `get_fd_binwidth(x)`: Returns the binwidth according to Freedman-Diaconis' rule.

Each of these functions should take in a quantitative vector `x` (e.g., `songs$YEAR`). To get you started, there is some template code below that you can edit. **After you've written your functions, report what each function returns when you input `songs$YEAR`.** Here are a few other hints to help you with this problem:

- Sturges' rule and the Rice rule involve the ceiling function; the ceiling function in `R` is `ceiling(x)`.
- The Rice rule and Scott's rule involves the cubic root; note that $\sqrt[3]{n} = n^{1/3}$, which you can type in `R`.
- Sturges' rule also uses the $\log_2(x)$ function; this is `log2(x)` in `R`.
- Freedman-Diaconis' rule involves the IQR, which you can compute using the `IQR()` function.

To be clear, this problem requires minimal coding; each function only requires 3-4 lines of code (some of which I already give you in the template code below). This problem is just meant to demonstrate that sometimes we can write mini functions to help us make graphs (as we will see in Part B). **Remember to report what the each function returns when you input the `songs$YEAR` variable!**

```
# TEMPLATE CODE - MODIFY THIS
get_sturges_bin_num <- function(x) {
    # number of observations
    n <- length(x)
    # thus, the number of bins is
    k <- ?
    return(k)
}
#TEMPLATE CODE - MODIFY THIS
get_rice_bin_num <- function(x) {
    # number of observations
    n <- length(x)
    # thus, the number of bins is
    k <- ?
    return(k)
}
get_scotts_binwidth <- function(x) {
    # the standard deviation is
    sigma <- ?
    # number of observations
    n <- length(x)
    # thus, the binwidth is
    h <- ?
    return(h)
}
get_fd_binwidth <- function(x) {
```

```
  # the IQR is
    iqr <-  ?
    # number of observations
    n <- length(x)
    # thus, the binwidth is
    h <- ?
    return(h)
}
# REPORT WHAT YOU GET FOR EACH OF THESE FUNCTIONS HERE (after uncommenting)
#get_sturges_bin_num(songs$YEAR)
#get_rice_bin_num(songs$YEAR)
#get_scotts_binwidth(songs$YEAR)
#get_fd_binwidth(songs$YEAR)
```

[PUT YOUR ANSWER HERE]

b. (10pts) Now make a histogram of `songs$YEAR` using each of the four rules from Part A. Specifically, make these four histograms:

- A histogram of `songs$YEAR`, specifying `bins` using Sturges' rule.
- A histogram of `songs$YEAR`, specifying `bins` using the Rice rule.
- A histogram of `songs$YEAR`, specifying `binwidth` using Scott's rule.
- A histogram of `songs$YEAR`, specifying `binwidth` using Freedman-Diaconis' rule.

Make these histograms using `ggplot`, and please place four histograms in a grid by following template code below with the `patchwork` package (make sure to install it in the Console first using `install.packages("patchwork")`). **Be sure to add appropriate titles for each plot so that it is clear which histogram corresponds to which rule.**

After you've made your histograms, discuss the differences among the four histograms in 1-3 sentences.

Remember that you need to specify `bins` for Sturges' rule and the Rice rule but `binwidth` for the other two rules!

```
# PUT YOUR CODE FOR THE STURGES PLOT HERE
# sturges_plot <- ?

# PUT YOUR CODE FOR THE RICE PLOT HERE
#rice_plot <- ?

# PUT YOUR CODE FOR THE SCOTT PLOT HERE
#scotts_plot <- ?

# PUT YOUR CODE FOR THE FREEDMAN-DIACONIS PLOT HERE
#fd_plot <- ?

# Display all arranged: (uncomment these lines!)
#library(patchwork)
#sturges_plot + rice_plot + scotts_plot + fd_plot
```

[PUT YOUR ANSWER HERE]

Problem 3: Violin plots: Transparency Side-by-Side (16 points)

In this problem we will continue to use the `songs` dataset, this time using violin plots to visualize the dataset.

a. (5pts) First, make a violin plot of the `YEAR` variable using `geom_violin()`; you can place `YEAR` on the x-axis or y-axis (your choice). However, be sure that the other axis does not display any numbers, which can be otherwise distracting. Within `geom_violin()`, specify a `fill` color of your choice. Make sure that your plot is appropriately labeled and titled. All you need to do here is make the plot (and include your code to make the plot, of course). I won't make you interpret the marginal distribution of `YEAR` based on this plot; we already did that earlier with histograms. (**Hint**: To make numbers not display on one of the axes, try setting that axis variable, `x` or `y`, equal to `""`.)

```
# PUT YOUR CODE AND PLOT HERE
```

b. (5pts) Now copy-and-paste your code from Part A here and do the the following:

- add `alpha = .5` within `geom_violin()`,
- add a boxplot layer on top of your violin plot with `+ geom_boxplot(width = .2, alpha = .5)`.

Compared to Part A, you should see that the violin-area itself is somewhat transparent. Plus you can now see a boxplot on top. It appears some of the outlier dots are more transparent than others. Why does this happen? And what does impact does `width` have on the boxplot? Explain in 1-4 sentences.

```
# PUT YOUR CODE AND PLOT HERE
```

[PUT YOUR ANSWER HERE]

c. (6pts) For this part, copy-and-paste your code from Part B, and then change your code accordingly such that you display side-by-side violins with overlaid boxplots showing the conditional distribution of `YEAR` given `THEME`. Furthermore, `fill` each violin/box according to `THEME`, such that each violin/box is colored by theme. Instead of specifying the `fill` within the `geom_violin()` layer, you can specify it within the initial `ggplot()` function so that the `fill` aesthetic is **shared** across both `geom_violin()` and `geom_boxplot()` layers, e.g., `ggplot(songs, aes(x = YEAR, y = THEME, fill = THEME)) + ...`

Again make sure that your plot is appropriately titled/labeled. Your final plot should have `YEAR` on one axis, `THEME` on another axis, and the violins/boxes filled by `THEME` (with some transparency).

```
# PUT YOUR CODE AND PLOT HERE
```

Problem 4: Whining about density estimates and histograms (40 points)

In this problem, we will work with a dataset of red wines. The dataset is provided here by Kaggle. Documentation for all of the variables is available at that link as well. The help documentation is also available on the course GitHub here for reference.

Here is the code to read in the data into `R`:

```
wine <- read_csv("https://raw.githubusercontent.com/ryurko/DataViz-36613-Fall22/main/data/wineQualityRe
```

The following problems focus on examining the distribution of `volatile.acidity` with density estimates and histograms.

a. (8pts) First we will plot the marginal distribution of `volatile.acidity`, and visually assess if it follows a Normal distribution. To do this, complete the following steps:

- Create a histogram of `volatile.acidity` on the **density scale**. Make the color something other than gray or black, and (as usual) be sure your plot is properly titled and labeled.

```
# PUT YOUR CODE HERE
```

- To see how well a Normal distribution fits to `volatile.acidity`, write code that defines the sample mean and variance `volatile.acidity`; define these as `mean_est` and `var_est`, respectively. Then, using the code below (that you have to modify), define the variable `acidity_norm_pdf`, which is the estimated Normal density for `volatile.acidity`:

```
# first, define the mean and variance: (uncomment these lines!)
#mean_est <- ?
#var_est <- ?

# then, compute the Normal density
wine$acidity_norm_pdf <- dnorm(wine$volatile.acidity,
                               mean = 0, sd = 1)
```

Currently, the above code fits a standard Normal density (with mean 0, standard deviation 1), and you need to replace the mean and sd arguments appropriately. **Remember that sd is for standard deviation, which is the square root of the variance.**

- Finally, add the Normal density curve to your histogram of `volatile.acidity`. To do this, first copy-and-paste your histogram code, and then add the line `+ geom_line(aes(volatile.acidity, acidity_norm_pdf))`.

```
# PUT YOUR CODE AND PLOT HERE
```

Ultimately, you should end up with a histogram of `volatile.acidity` with a Normal density curve added to the plot. After you've made your plot, describe the distribution of `volatile.acidity` in 1-2 sentences, and then discuss whether or not you think this distribution appears to be Normally distributed (say Yes or No, and then provide a 1-2 sentence explanation).

[PUT YOUR ANSWER HERE]

b. (5pts) Now we'll formally test whether `volatile.acidity` follows a Normal distribution. Do this using the Kolmogorov-Smirnov (KS) test. Specifically: Test whether `volatile.acidity` follows a Normal distribution whose mean and standard deviation are equal to the sample mean and sample standard deviation of `volatile.acidity` observed in the data, respectively. In your answer, after providing the appropriate code for the KS test, state your formal conclusion from the test. (For this problem, just ignore the warning message that is displayed.)

```
# PUT YOUR CODE HERE
```

[PUT YOUR ANSWER HERE]

c. (6 points) Create two density plots of `volatile.acidity`. (You can use the `density` function in base `R`, or you can use the `geom_density` function in `ggplot`.) In the first plot, use a small bandwidth, so that many local features of the distribution are shown, and the density estimate is somewhat "jagged" / "rigid" / not smooth. In the second, use a larger bandwidth, so that the local features of the distribution are smoothed out. Include both plots in your answer, and compare and contrast the two plots (e.g., discuss which features are easier or more difficult to see in each of the plots) in 1-3 sentences. You may have to try a few bandwidth values to figure out what is a relatively small or large bandwidth. For this problem, instead of modifying the `adjust` parameter like we did in lecture, you should directly change the `bw` input for `geom_density()` instead.

Make sure that each plot is appropriately titled/labeled such that we know which is the "large bandwidth" plot and which is the "small bandwidth" plot. Furthermore, be sure to include the *units* of the `volatile.acidity` variable in your labels; you'll have to check the data documentation to see what the units are for this variable.

```
# PUT YOUR CODE AND PLOTS HERE
```

[PUT YOUR ANSWER HERE]

d. (15 points) For the remainder of this question, we will also work with the variable `quality`. Look at the help documentation for this variable. From this description, is this variable a quantitative/continuous variable, quantitative/discrete variable, categorical/nominal variable, or categorical/ordinal variable? After answering this, use the `class` function to determine what type of variable this is in `R`. In your answer, report what the class is.

```
# PUT YOUR CODE HERE
```

[PUT YOUR ANSWER HERE]

Regardless of your answer, please run the following code to treat `quality` as a factor:

```
wine <- wine %>% mutate(quality = factor(quality))
```

Then, to assess if the distribution of `volatile.acidity` differs depending on the `quality` of the wine, make four different plots:

- A "stacked histogram" of `volatile.acidity` (where the bars are colored according to the wine `quality`). To do this, first make a histogram of `volatile.acidity`, and then specify `fill` as `quality`.

```
# PUT YOUR CODE AND PLOT HERE
```

- A "conditional density plot," where there are six density curves of `volatile.acidity` – one for each wine quality grade – on a single plot, with each curve colored according to the wine quality). Let `R` choose the bandwidth automatically.

```
# PUT YOUR CODE AND PLOT HERE
```

- Facetted histograms of `volatile.acidity`, facetted by `quality`.

```
# PUT YOUR CODE AND PLOT HERE
```

- Next create a ridgeline plot. To make ridgeline plots, you'll need to install the ggridges R package; do that now by running `install.packages("ggridges")` in your Console. I've found that you need to restart RStudio after installing the package. Once you do that, you should be able to uncomment the following code, but remember to add appropriate title/labels:

```
# PUT YOUR CODE AND PLOT HERE
```

- Notice the message that appeared regarding a `joint bandwidth`. By default, `geom_density_ridges()` estimates a single bandwidth to use for each density in the plot. Alternatively, you can force `geom_density_ridges()` to estimate a different bandwidth for each density by doing the following:
  - Copy-and-paste your above `ggridges` code
  - Add `height = stat(density)` within `aes()`.
  - Add `stat = "density"` within `geom_density_ridges()`.
  - Make another ridgeline plot by following the above three steps. Then, compare and contrast the plot you made here with the previous ridgeline plot. In particular: Which smoothed densities look similar, and which smoothed densities look different?

```
# PUT YOUR CODE AND PLOT HERE
```

[PUT YOUR ANSWER HERE]

All you have to do for this part is create the five plots above (as well as answer the preliminary questions about `quality` and the difference between your two ridgeline plots). We'll interpret the plots in the next part.

e. (6pts) After looking at your graphs in Part D, Korg asks you: "Do you think the distribution of `volatile.acidity` depends on the quality of the wine?" Use one or more of your graphs in Part D to answer this question in 1-4 sentences. Be sure that your answer discusses whether the center, spread, and shape of `volatile.acidity` depend (or not depend) on `quality`, and mention which graph(s) you used to arrive at each of your observations.

[PUT YOUR ANSWER HERE]