HW 4

SDS322E

September 21, 2022

Ethan Chang - ehc586

Please submit as a PDF or HTML file on Canvas before the due date.

For all questions, include the R commands/functions that you used to find your answer. Answers without supporting code will not receive credit.

> Review of how to submit this assignment
>
> All homework assignments will be completed using R Markdown. These `.Rmd` files consist of >text/syntax (formatted using Markdown) alongside embedded R code. When you have completed the assignment (by adding R code inside codeblocks and supporting text outside of the codeblocks), create your document as follows (assuming you are using the edupod server and submitting HTML):

> - Click the arrow next to the “Knit” button (above)
> - Choose “Knit to HTML”
> - Go to Files pane and put checkmark next to the correct HTML file
> - Click on the blue gear icon (“More”) and click Export
> - Download the file and then upload to Canvas
> - To submit a PDF, open your HTML file and print it to a pdf, then upload the pdf as your submission.

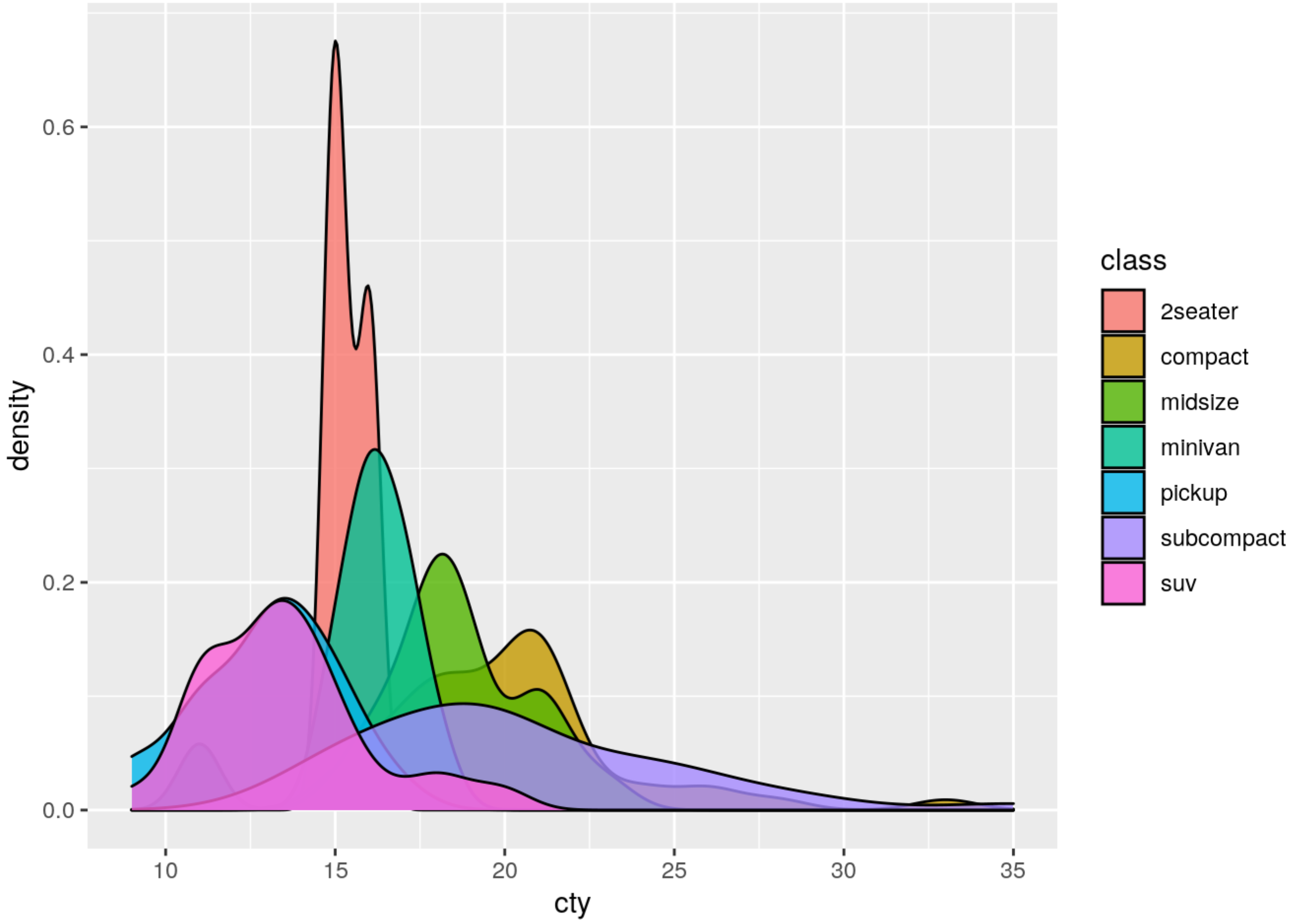
Question 1 (1pts)

Load the `ggplot2` package, which contains the `mpg` dataset. It contains data from 38 popular car models in 1999 and in 2008. **Do the following:**

- Make a density plot of `cty` (city miles-per-gallon) and fill it by `class` of car.
- Add an alpha value of .8 to increase transparency.

Eyeballing the plot, **which two classes have the most overlap in this distributions? Which class has the least variation in `cty` mpg and which has the most?**

```
library(ggplot2)
data(mpg)
ggplot(mpg, aes(x = cty, fill = class)) + geom_density(alpha = 0.8)
```



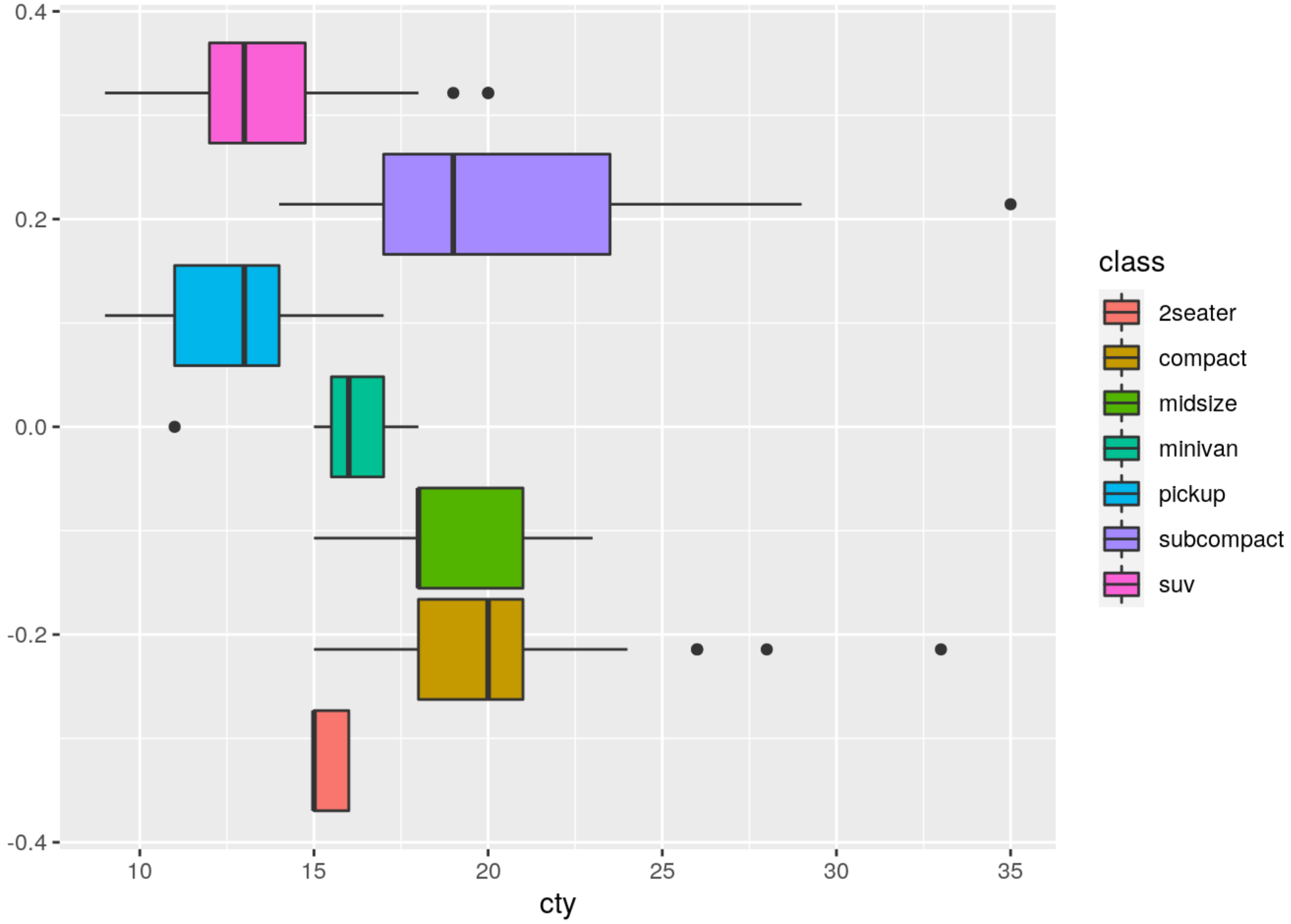
Answer: From the density plot, it appears that the **suv** and **pickup** classes have the most overlap in this distribution. **2seater** has the least variation in `cty` mpg while **subcompact** has the most.

Question 2 (1pts)

The figure from Question 1.1 is not particularly attractive: the curves overlap to a degree that it is difficult to see what is going on in any single group.

Make a new plot which could be used to answer Question 1.1, but which is easier to read and preserves all of the information in the density plot. This could involve changing the `geom_` to something else, faceting on a variable, etc. Do whatever you feel best facilitates the comparisons from Question 1.1, and e**xplain why you feel that your new figure is better.**

```
ggplot(mpg, aes(x = cty, fill = class)) + geom_boxplot()
```

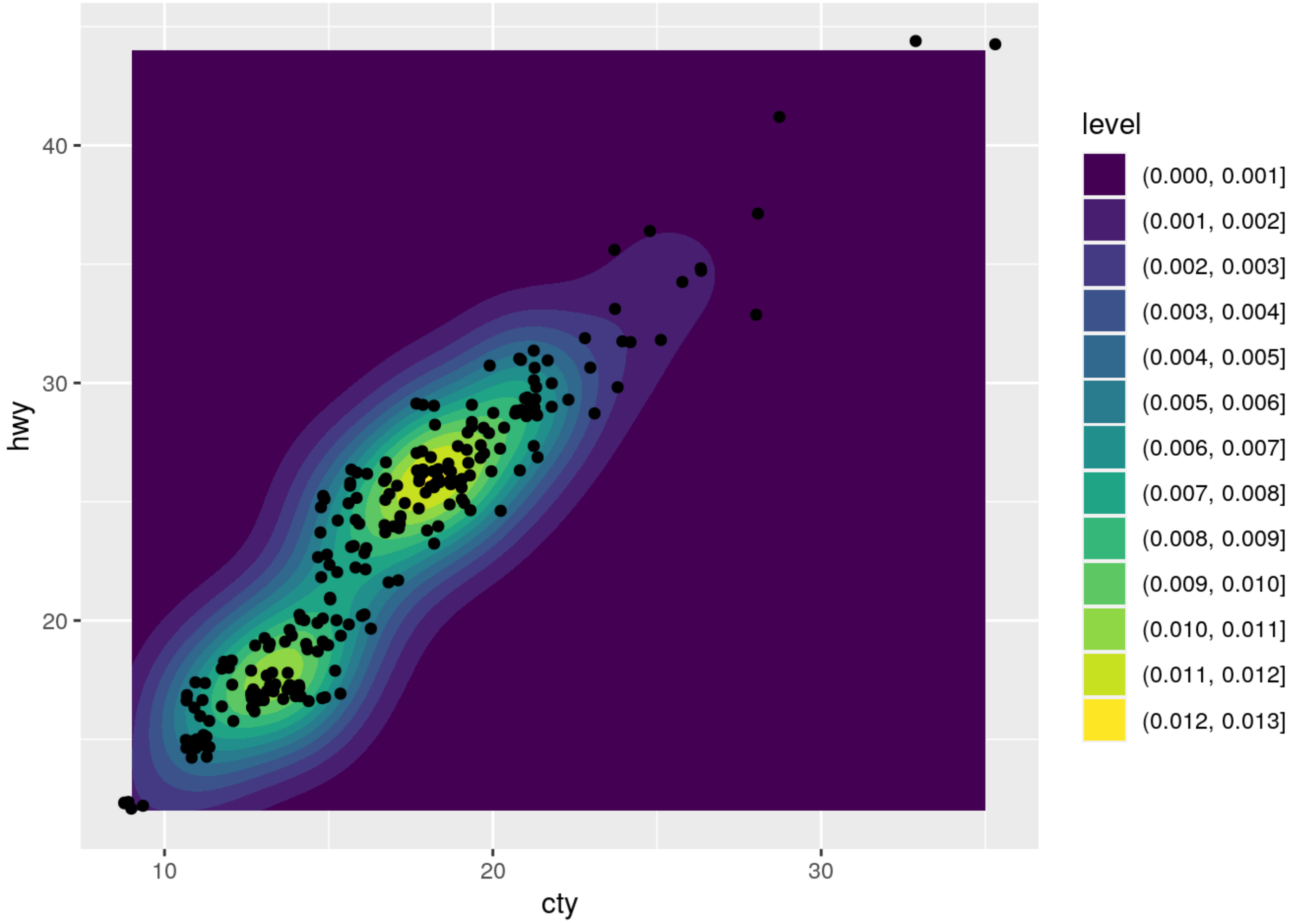


Answer: Using boxplots like this is better than the density plots from question 1 as it allows the users to easily compare the distributions of the `cty` mpgs of all the available classes easily, making it easier to notice overlaps and differences in variations without obscuring vision.

Question 3 (1pts)

Make a two-dimensional density plot (using, e.g., `geom_density2d_filled`) to visualize the joint distribution of `cty` and `hwy`. Add `geom_jitter()` to show the data points but give a slight amount of random noise vertically and horizontally so that the points don't totally overlap. Describe which areas of the plot show the greatest density of cars and which show the least density of cars in terms of their city and highway mpg.

```
ggplot(mpg, aes(x = cty, y = hwy)) + geom_density_2d_filled() +
    geom_jitter()
```



Answer: From the plot, it can be seen that the areas with the highest density of cars have relatively low city and highway mpgs (city between 10-25 and highway between 10-30), creating two concentrated peaks at around (12,18) and (19,26). The lowest density of cars appear to be in high city and low highway, or high highway and low city mpgs, off the diagonal.

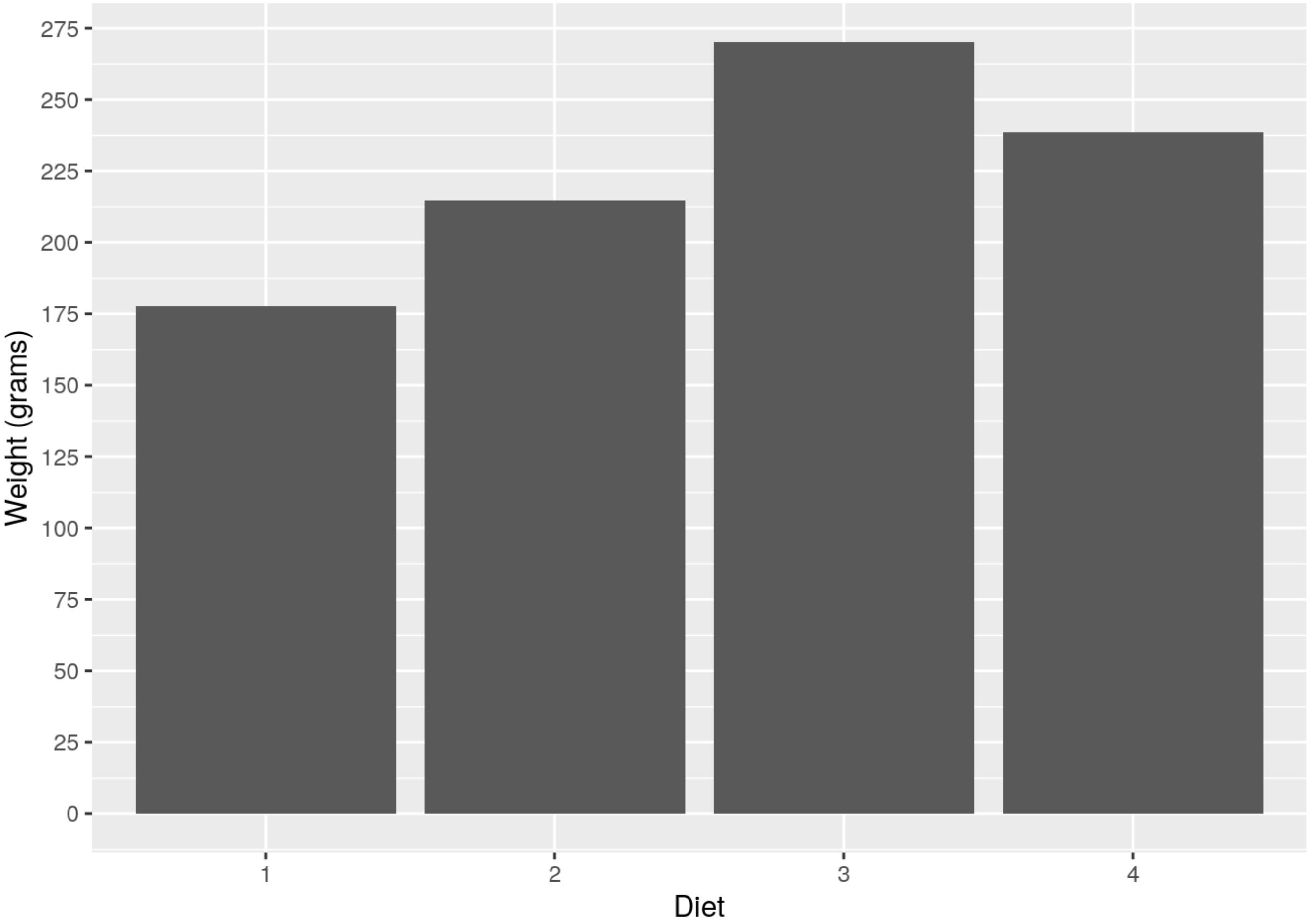
Question 4 (1pts)

We will now revisit the `ChickWeight` dataset, which contains information about the weights (in grams)of chicks on different diets over time (at 2-day intervals). Review the previous homework to refresh yourself on what you have done with this dataset.

A scatterplot might not be the best way to visualize this data: It calls attention to the relationship between weight and time, but it can be hard to see the differences between diets. A more traditional approach for comparing groups would be to use barplot of group means with standard error bars showing +/- 1 standard error.

Create a plot using `geom_bar` where each bar’s height corresponds to the average chick weight for each of the four diet conditions at the end of the study (i.e., first subset the data so that `Time == 21`). Rename the y-axis to include units (e.g., using `ylab()` or `labs()`) and make the major tick marks go from 0 to 275 by 25 (with `breaks =` in `scale_y_continuous()`).

```
library(tidyverse)
ggplot(filter(ChickWeight, ChickWeight$Time == 21),
    aes(x = Diet, y = weight)) + geom_bar(stat = "summary") +
    ylab("Weight (grams)") + scale_y_continuous(breaks = seq(0,
    275, by = 25))
```



Using this plot, what approximately is the average weight for Diet 1?

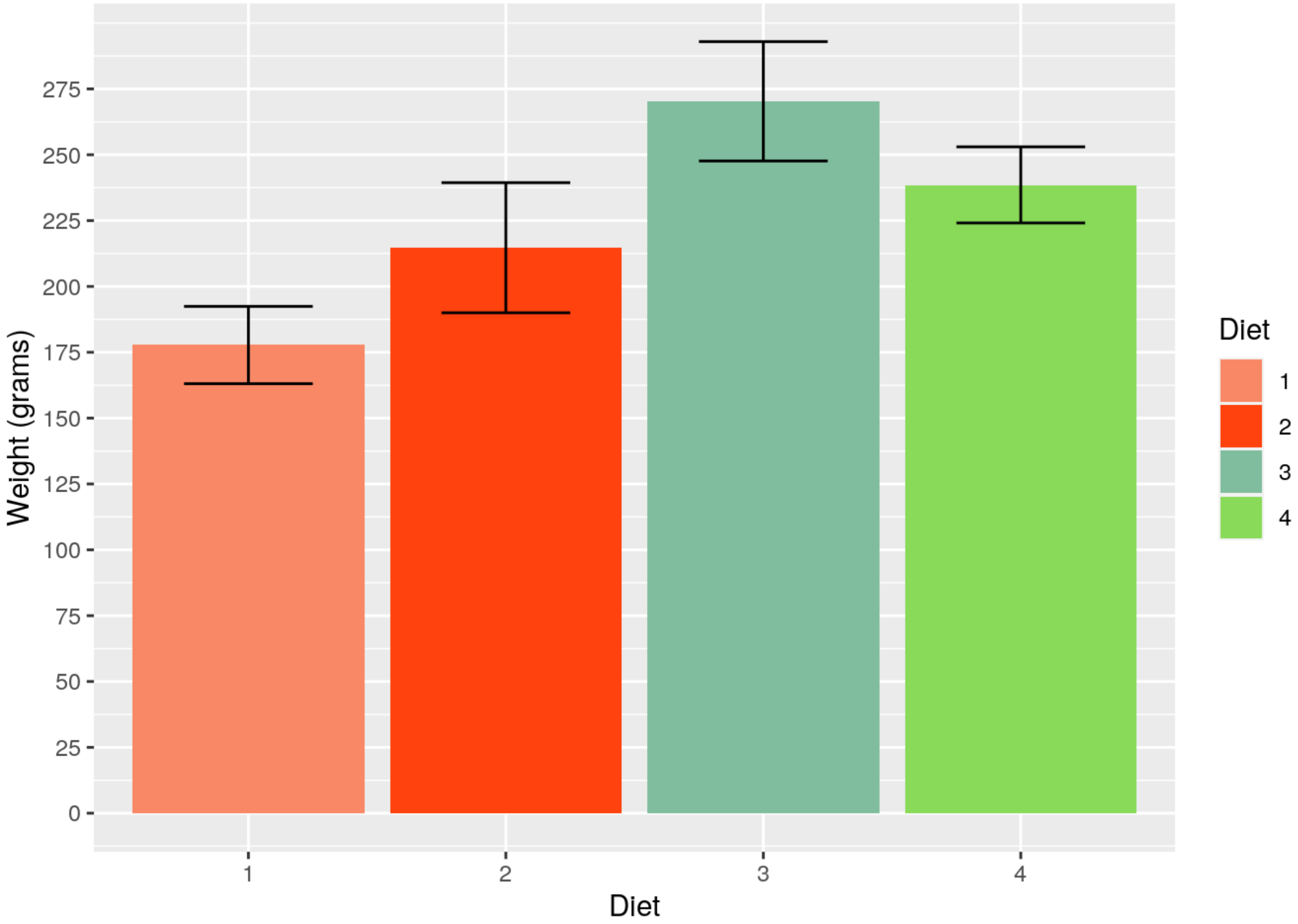
Answer: Diet 1 is around 175 grams on average.

Question 5 (1.5pts)

Make the following modifications to your previous plot:

- Add error bars showing the mean plus/minus one standard error using `geom_errorbar(stat="summary")` .
- Make the error-bars skinnier by adding a `width=` argument.
- Color the bars (not the error bars, but the bar chart bars) by diet and change from the default color scheme using a `scale_fill_` or a `scale_color_`

```
library(ggthemes)
ggplot(filter(ChickWeight, ChickWeight$Time == 21),
    aes(x = Diet, y = weight, fill = Diet)) + geom_bar(stat = "summary") +
    ylab("Weight (grams)") + scale_y_continuous(breaks = seq(0,
    275, by = 25)) + geom_errorbar(stat = "summary",
    width = 0.5) + scale_fill_canva()
```



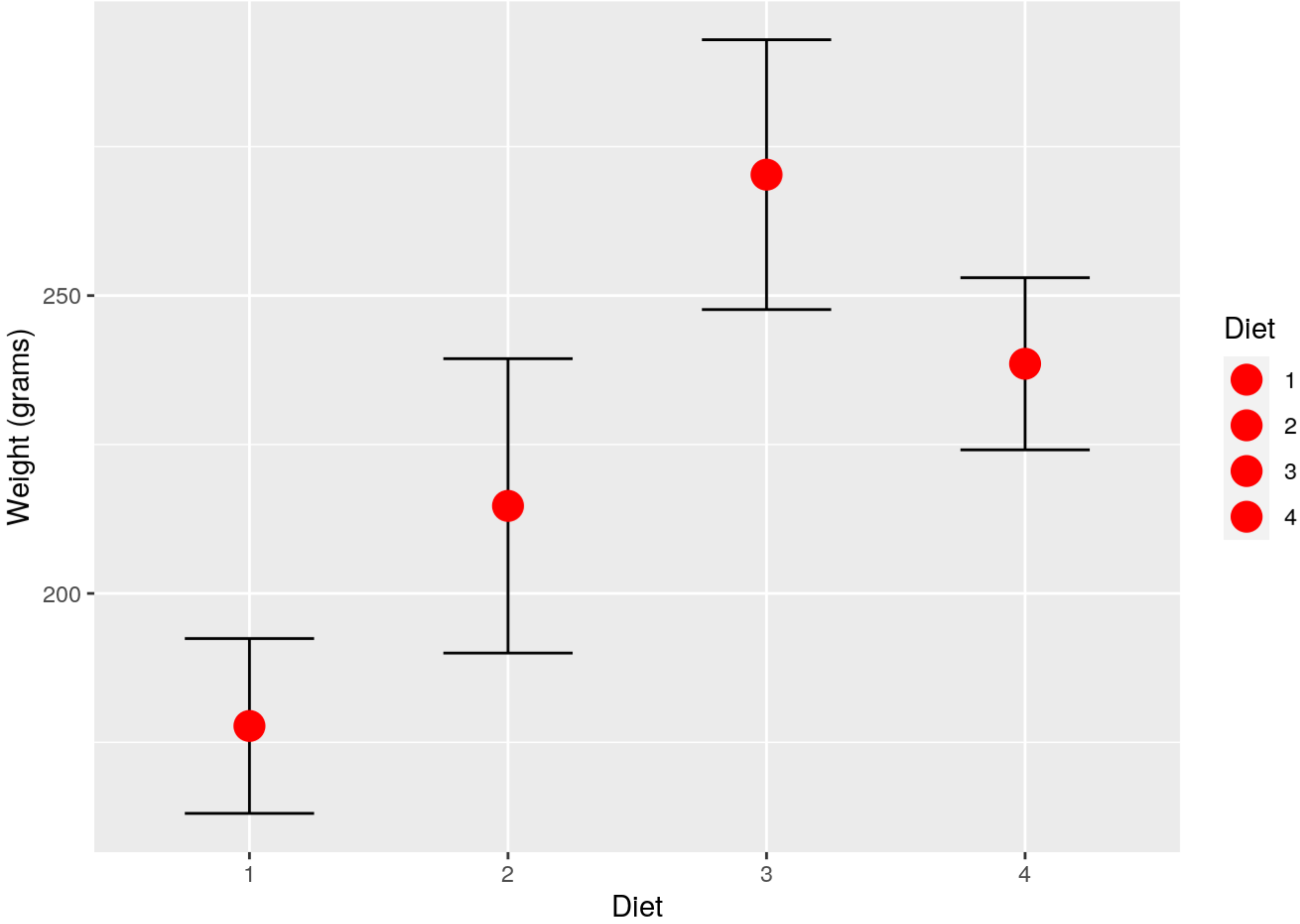
Question 6 (1.5pts)

Take your code from 2.2 and do the following:

- Replace the barplot with a figure that has a single point for each diet (as in a scatterplot) at the same value on the y-axis as the barplot.
- Remove the `breaks =` argument from `scale_y_continuous`.
- Make the points larger than their default by setting `size = 5` and color them all red.
- Put them *on top* of the error bars.

Do you prefer this figure or the barplot?

```
ggplot(filter(ChickWeight, ChickWeight$Time == 21),
    aes(x = Diet, y = weight, fill = Diet)) + geom_errorbar(stat = "summary",
    width = 0.5) + geom_point(stat = "summary", size = 5,
    color = "red") + ylab("Weight (grams)") + scale_y_continuous() +
    scale_fill_canva()
```



Answer: I prefer the barplot as it is more visually appealing when compared to the this figure, and has a better scale.

Question 7 (1pts)

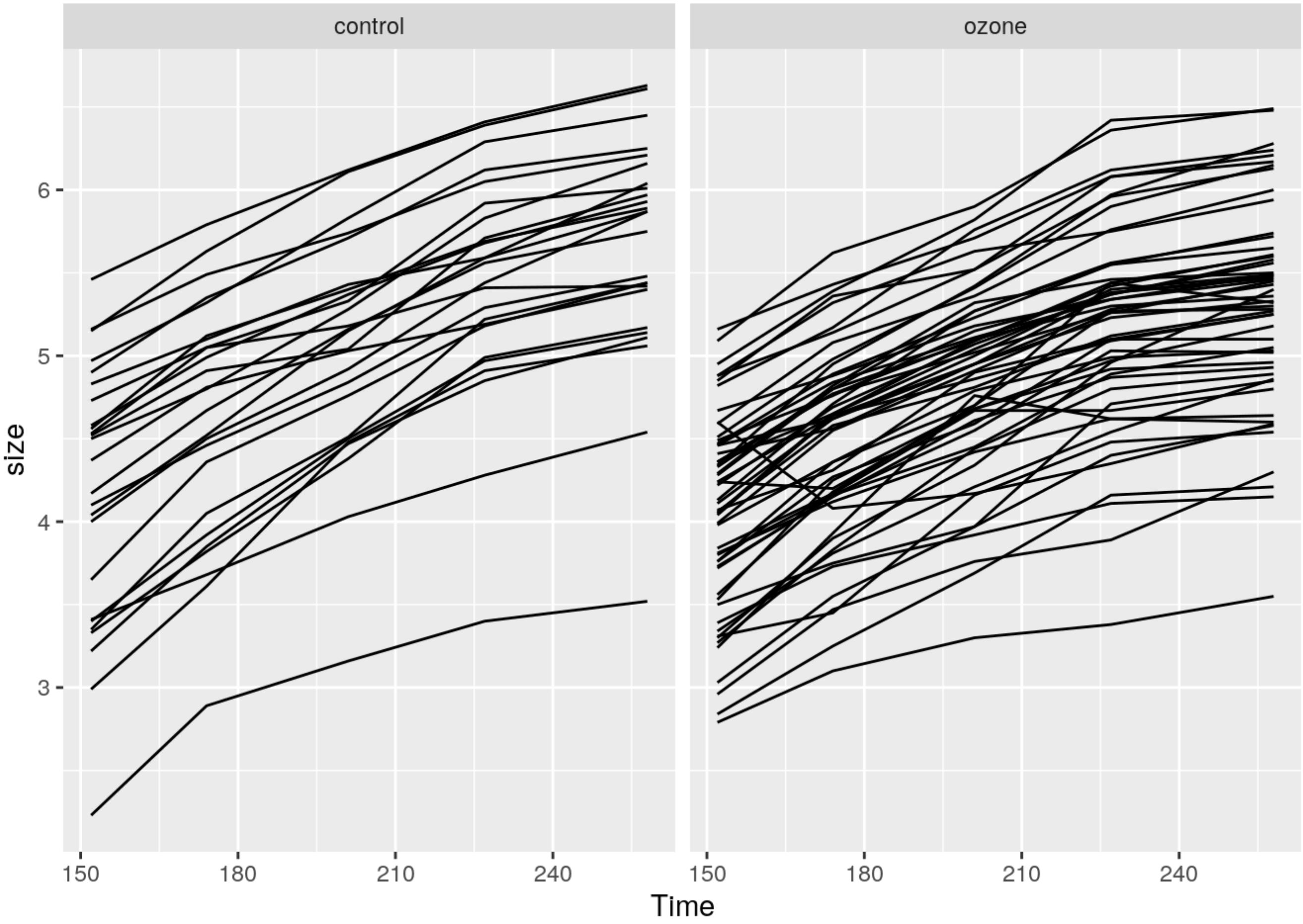
The data set `Sitka` contains repeated measurements of tree size for 79 Sitka spruce trees, which were grown either in ozone-enriched chambers or under control conditions. It contains four columns: `size` measures the size of the tree (height times diameter squared, on a log scale). `Time` measures the time, in days since Jan. 1, 1988. `tree` indicates the tree we are working with, consecutively numbered from 1 to 79. `treat` indicates the treatment trees were subjected to, either ozone for an ozone-enriched chamber or control.

Make line plots of tree size vs. time, for each tree, faceted by treatment. Use the same color for all lines. *Hint:* you will need to use the group aesthetic to tell ggplot that you want to have a separate line for each tree.

```
Sitka <- MASS::Sitka  ## You may need to install the MASS package to do this
head(Sitka)
```

```
##   size Time tree treat
## 1 4.51  152    1 ozone
## 2 4.98  174    1 ozone
## 3 5.41  201    1 ozone
## 4 5.90  227    1 ozone
## 5 6.15  258    1 ozone
## 6 4.24  152    2 ozone
```

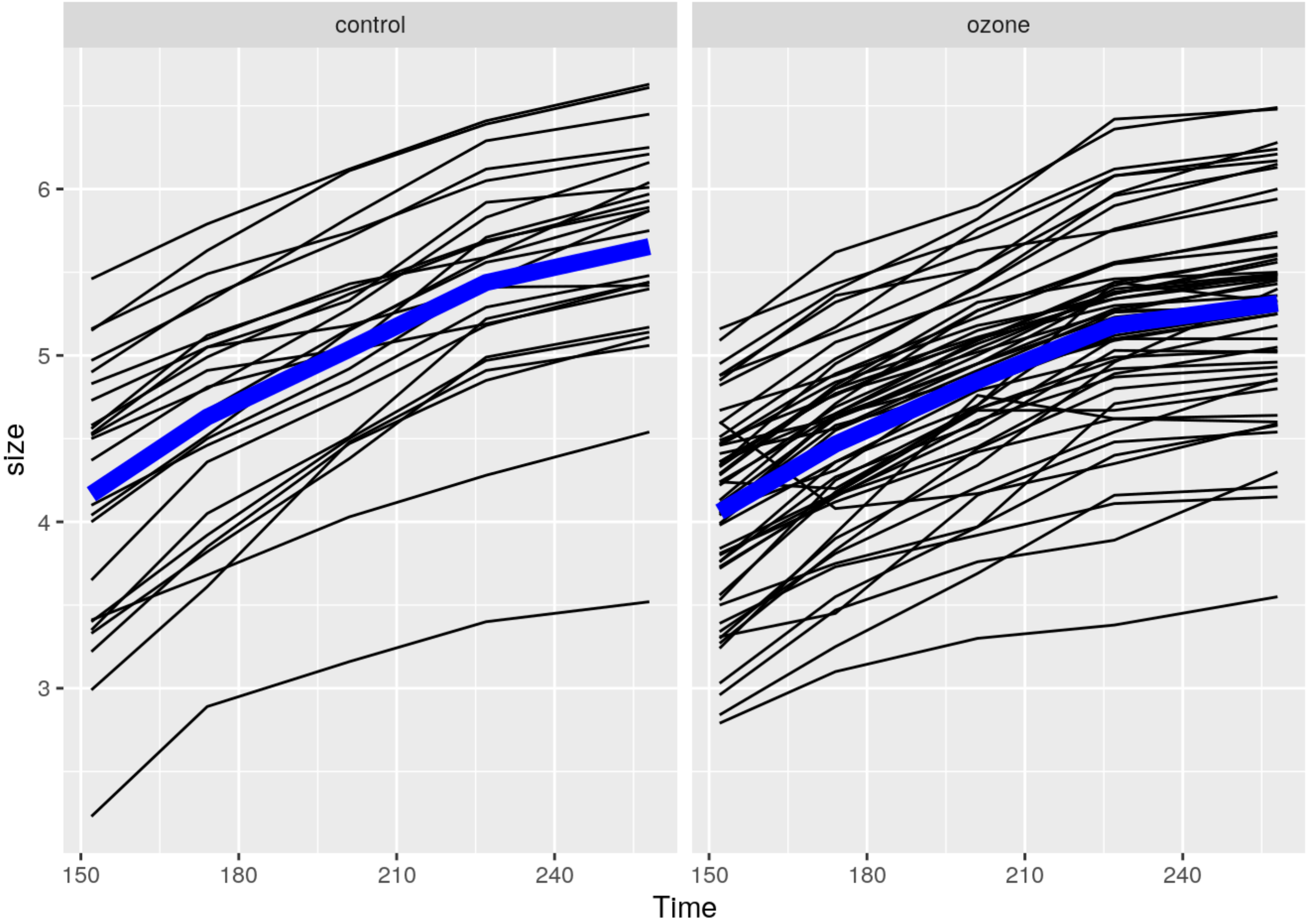
```
ggplot(Sitka, aes(x = Time, y = size)) + geom_line(mapping = aes(group = tree)) +
    facet_wrap(~treat)
```



Question 8 (2pts)

Using another `geom_line` (so two `geom_line` s in the expression) with an appropriate `stat` , modify the previous plot to include the average `size` of the tree for each value of `Time` ; there should be one new line for each of `ozone` and `control` which tracks the size of the average tree over time, in addition to separate lines tracking the growth of each individual tree. Color this line blue and set its size to be `3` .

```
ggplot(Sitka, aes(x = Time, y = size)) + geom_line(mapping = aes(group = tree)) +
    facet_wrap(~treat) + geom_line(stat = "summary",
    color = "blue", size = 3)
```



How do the average growth trajectories for the two treatments compare to each other?

Answer: When comparing the average growth trajectories of the two treatments, it can be seen that both treatments have roughly the same average growth trajectory, with `control` potentially having a slightly higher one and shifted up a bit. `Control` starts slightly higher than `ozone` (`control` beginning at around 4.1, while `ozone` begins at around 4), and potentially increases slightly more as it ends at a location higher than `ozone` when compared to the beginning (`control` ends at around 5.6, while `ozone` ends at around 5.3).

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.6 LTS
## 
## Matrix products: default
## BLAS:   /stor/system/opt/R/R-4.0.3/lib/R/lib/libRblas.so
## LAPACK: /stor/system/opt/R/R-4.0.3/lib/R/lib/libRlapack.so
## 
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C              
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8    
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8   
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                 
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C            
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C       
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
##  [1] ggthemes_4.2.4  forcats_0.5.1   stringr_1.4.0   dplyr_1.0.9    
##  [5] purrr_0.3.4     readr_2.1.2     tidyr_1.2.0     tibble_3.1.8   
##  [9] tidyverse_1.3.2 ggplot2_3.3.6  
## 
## loaded via a namespace (and not attached):
##  [1] lubridate_1.8.0     assertthat_0.2.1    digest_0.6.29      
##  [4] utf8_1.2.2          R6_2.5.1            cellranger_1.1.0   
##  [7] backports_1.4.1     reprex_2.0.1        evaluate_0.15      
## [10] httr_1.4.3          highr_0.9           pillar_1.8.0       
## [13] rlang_1.0.4         readxl_1.4.0        googlesheets4_1.0.0
## [16] rstudioapi_0.13     jquerylib_0.1.4     rmarkdown_2.14     
## [19] labeling_0.4.2      googledrive_2.0.0   munsell_0.5.0      
## [22] broom_1.0.0         compiler_4.0.3      modelr_0.1.8       
## [25] xfun_0.31           pkgconfig_2.0.3     htmltools_0.5.3    
## [28] tidyselect_1.1.2    fansi_1.0.3         viridisLite_0.4.0  
## [31] crayon_1.5.1        tzdb_0.3.0          dbplyr_2.2.1       
## [34] withr_2.5.0         MASS_7.3-58         grid_4.0.3         
## [37] jsonlite_1.8.0      gtable_0.3.0        lifecycle_1.0.1    
## [40] DBI_1.1.3           magrittr_2.0.3      formatR_1.12       
## [43] scales_1.2.0        cli_3.3.0           stringi_1.7.8      
## [46] cachem_1.0.6        farver_2.1.1        fs_1.5.2           
## [49] xml2_1.3.3          bslib_0.4.0         ellipsis_0.3.2     
## [52] generics_0.1.3      vctrs_0.4.1         tools_4.0.3        
## [55] glue_1.6.2          hms_1.1.1           fastmap_1.1.0      
## [58] yaml_2.3.5          colorspace_2.0-3    gargle_1.2.0       
## [61] rvest_1.0.2         isoband_0.2.5       knitr_1.39         
## [64] haven_2.5.0         sass_0.4.2
```

```
## [1] "2022-09-21 15:16:54 CDT"
##                                       sysname 
##                                       "Linux" 
##                                       release 
##                          "4.15.0-191-generic" 
##                                       version 
## "#202-Ubuntu SMP Thu Aug 4 01:49:29 UTC 2022" 
##                                      nodename 
##                  "educcomp04.ccbb.utexas.edu" 
##                                       machine 
##                                      "x86_64" 
##                                         login 
##                                     "unknown" 
##                                          user 
##                                      "ehc586" 
##                                effective_user 
##                                      "ehc586"
```