Homework 2 KEY

Jacob Martin

Spring 2022

knitr::opts\_chunk$set(echo = TRUE,  
 fig.width=10,   
 fig.height=6,   
 fig.align = "center")  
  
# Load the needed package(s) here  
pacman::p\_load(tidyverse, gridExtra)  
  
# Change the default theme below:  
theme\_set(theme\_bw())

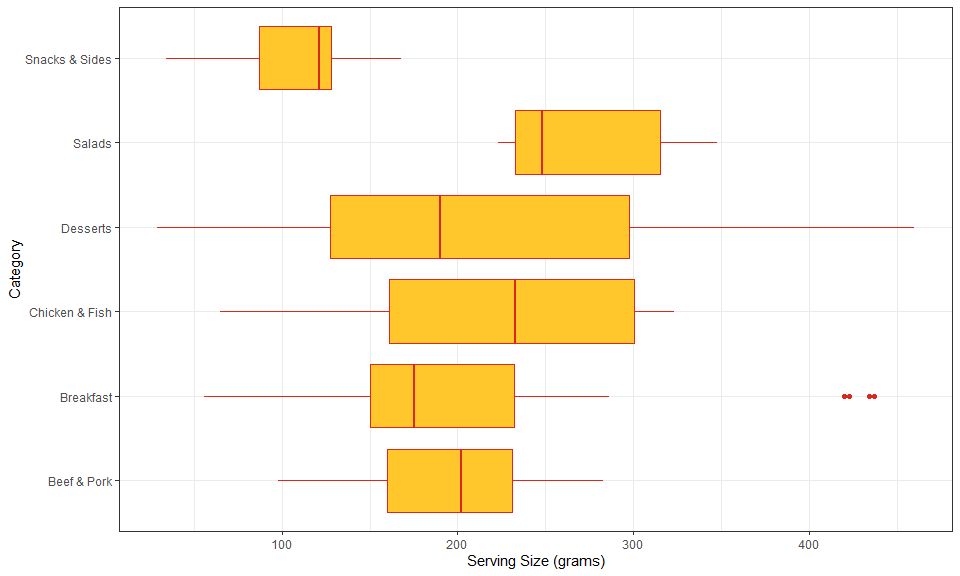
## Part 1: McDonald’s Nutrition

The “fast food menu.csv” data set contains the nutritional information on many different non-drink options offered at McDonald’s. Use an appropriate theme for the graphs.

### 1a) Calories by serving size

Create horizontal box plots of serving\_size\_grams by Category. Fill the box plots with all the same color, but use a different color than the default. Change the label of the x-axis to “Serving Size (grams)”. Which Category has the largest and smallest serving sizes overall?

# Start by reading in the data here.   
# Name the data set McD  
McD <- read.csv("fast food menu.csv")  
  
# Create the boxplot below  
ggplot(data = McD,  
 mapping = aes(x = serving\_size\_grams,  
 y = Category)) +  
   
 # Using McDonald's yellow and red to color the box plots  
 geom\_boxplot(fill = "#FFC72C",  
 color = "#DA291C") +   
   
 labs(x = "Serving Size (grams)")



Snacks have the smallest typical serving size and salads, surprisingly, have the largest typical serving size!

### 1b) Calories by Category, Serving Size, and Nutritional Info

Create and SAVE create 6 similar scatterplots with:

* Calories on the y-axis
* Category indicated by color
* serving\_size\_grams represented by size.
* Each graph should have points and a single (straight) regression line, without the shaded region.

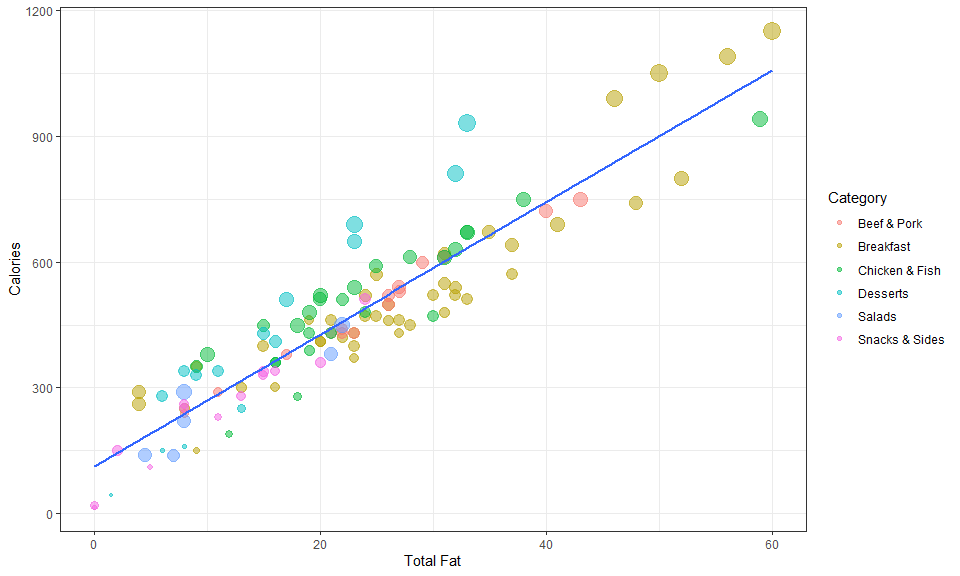
Make sure to change the labels on the axes & legends to be more readable, if necessary

1. Total Fat
2. Saturated Fat
3. Cholesterol
4. Sodium
5. Protein
6. Sugars

add + guides(size = “none”) to each of the six plots to hide the legend for the size aesthetic

#### Part 1bi. Total Fat

# Create, save, and display the scatterplot with x = Total Fat below  
gg\_cals\_tf <-   
   
 ggplot(data = McD,  
 mapping = aes(x = Total.Fat,   
 y = Calories)) +  
   
 geom\_point(mapping = aes(color = Category,  
 size = serving\_size\_grams),  
 alpha = 0.5) +   
   
 geom\_smooth(method = "lm",  
 se = F) +  
   
 labs(x = "Total Fat") +  
   
 guides(size = "none")  
   
gg\_cals\_tf



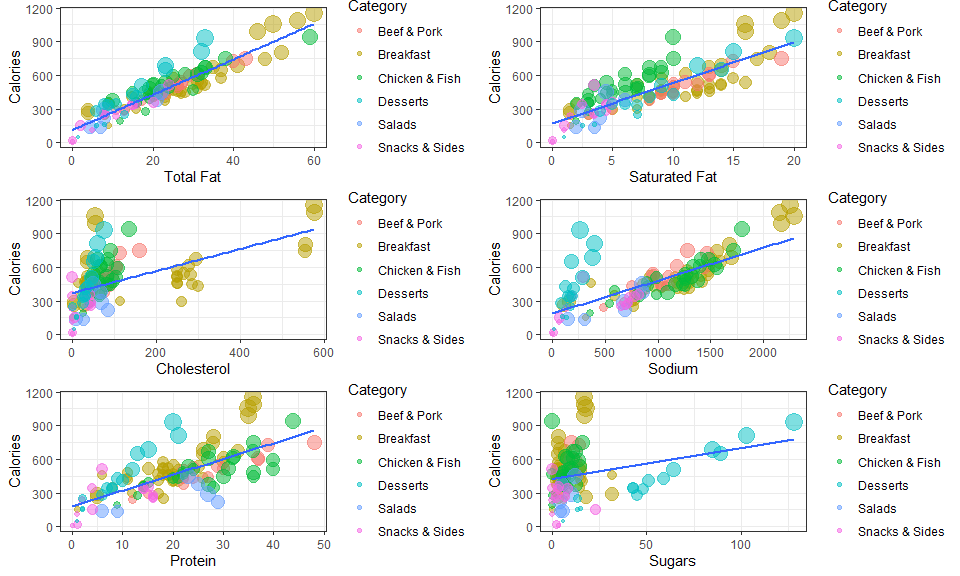
#### Part 1bii. - vi. Other 5 scatterplots

## Create and save but don't display the scatterplot with  
  
# ii) x = Saturated Fat below  
gg\_cals\_sf <-   
   
 ggplot(data = McD,  
 mapping = aes(x = Saturated.Fat,   
 y = Calories)) +  
   
 geom\_point(mapping = aes(color = Category,  
 size = serving\_size\_grams),  
 alpha = 0.5) +   
   
 geom\_smooth(method = "lm",  
 se = F) +  
   
 labs(x = "Saturated Fat") +  
   
 guides(size = "none")  
  
  
# iii) x = Cholesterol below  
gg\_cals\_chol <-   
   
 ggplot(data = McD,  
 mapping = aes(x = Cholesterol,   
 y = Calories)) +  
   
 geom\_point(mapping = aes(color = Category,  
 size = serving\_size\_grams),  
 alpha = 0.5) +   
   
 geom\_smooth(method = "lm",  
 se = F) +  
   
 guides(size = "none")  
  
  
# iv) x = Sodium  
gg\_cals\_na <-   
   
 ggplot(data = McD,  
 mapping = aes(x = Sodium,   
 y = Calories)) +  
   
 geom\_point(mapping = aes(color = Category,  
 size = serving\_size\_grams),  
 alpha = 0.5) +   
   
 geom\_smooth(method = "lm",  
 se = F) +  
   
 guides(size = "none")  
  
# v) x = Protein below  
gg\_cals\_pro <-   
   
 ggplot(data = McD,  
 mapping = aes(x = Protein,   
 y = Calories)) +  
   
 geom\_point(mapping = aes(color = Category,  
 size = serving\_size\_grams),  
 alpha = 0.5) +   
   
 geom\_smooth(method = "lm",  
 se = F) +  
   
 guides(size = "none")  
  
  
# vi) x = Sugars below  
gg\_cals\_sug <-   
   
 ggplot(data = McD,  
 mapping = aes(x = Sugars,   
 y = Calories)) +  
   
 geom\_point(mapping = aes(color = Category,  
 size = serving\_size\_grams),  
 alpha = 0.5) +   
   
 geom\_smooth(method = "lm",  
 se = F) +  
   
 guides(size = "none")

### Part 1c)

Present the 6 scatterplots using grid.arrange(), as described on page 1. Put all six together with 3 rows.

# Use grid.arrange below to place all 6 scatter plots together  
grid.arrange(gg\_cals\_tf,   
 gg\_cals\_sf,  
 gg\_cals\_chol,  
 gg\_cals\_na,  
 gg\_cals\_pro,  
 gg\_cals\_sug,  
 nrow = 3)



### Part 1d) Long Format

Run the code in the chunk below to transform the data into a “long” format.

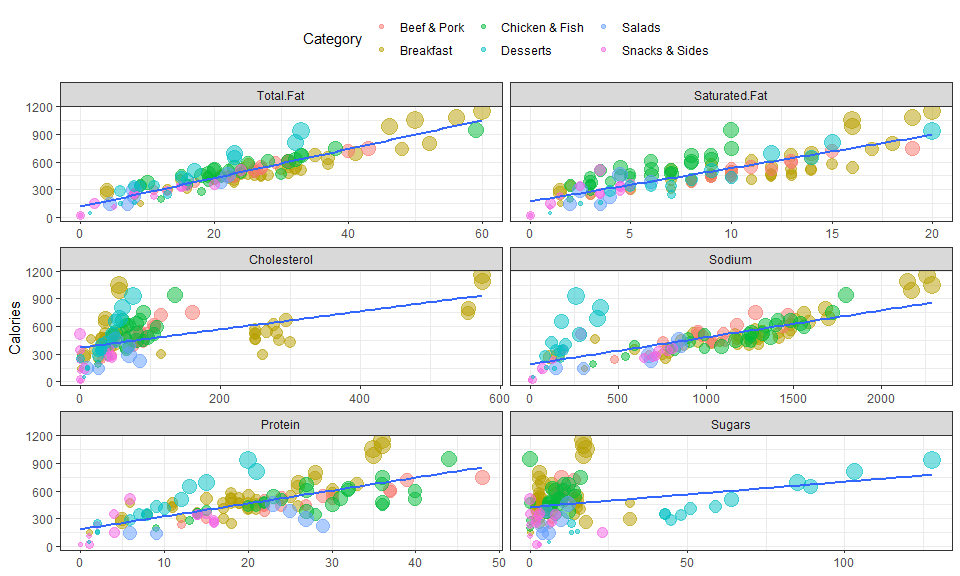
# Using pivot\_longer to create a new data set in the "long" format  
McD\_long <-   
 McD %>%   
   
 dplyr::select(Category, serving\_size\_grams, Calories, Total.Fat,   
 Saturated.Fat, Cholesterol, Sodium, Protein, Sugars) %>%   
   
 pivot\_longer(cols = Total.Fat:Sugars,  
 names\_to = "nutrition\_type",  
 values\_to = "amount") %>%   
   
 mutate(nutrition\_type = as\_factor(nutrition\_type))  
  
McD\_long

## # A tibble: 702 x 5  
## Category serving\_size\_grams Calories nutrition\_type amount  
## <chr> <int> <int> <fct> <dbl>  
## 1 Breakfast 136 300 Total.Fat 13  
## 2 Breakfast 136 300 Saturated.Fat 5  
## 3 Breakfast 136 300 Cholesterol 260  
## 4 Breakfast 136 300 Sodium 750  
## 5 Breakfast 136 300 Protein 17  
## 6 Breakfast 136 300 Sugars 3  
## 7 Breakfast 135 250 Total.Fat 8  
## 8 Breakfast 135 250 Saturated.Fat 3  
## 9 Breakfast 135 250 Cholesterol 25  
## 10 Breakfast 135 250 Sodium 770  
## # ... with 692 more rows

Using McD\_long, create the same six scatterplots from part b), but using facet\_wrap instead of grid.arrange.

You should only need 1 ggplot function, and use 3 rows again. Make sure to use a good choice for the scales argument inside facet\_wrap.

# Use the transformed data and facet\_wrap to create a similar graph to 1c)  
ggplot(data = McD\_long,  
 mapping = aes(x = amount,  
 y = Calories)) +  
   
 geom\_point(mapping = aes(color = Category,  
 size = serving\_size\_grams),  
 alpha = 0.50) +   
   
 geom\_smooth(method = "lm",  
 se = F) +   
   
 facet\_wrap(facets = ~ nutrition\_type,  
 nrow = 3,  
 scales = "free\_x") +  
   
   
   
 # Add the two lines below to your code   
 theme(legend.position = "top") +   
   
 guides(size = "none") +  
   
 labs(x = NULL)



### Part 1e)

Which set of graphs do you prefer? Describe why

Answer may vary, but the graph in part d) is what I prefer since the legend only appears once in the plot, not once for each of the graphs, which takes up the majority of the space on the plot.

## Question 2: NFL Drive Data

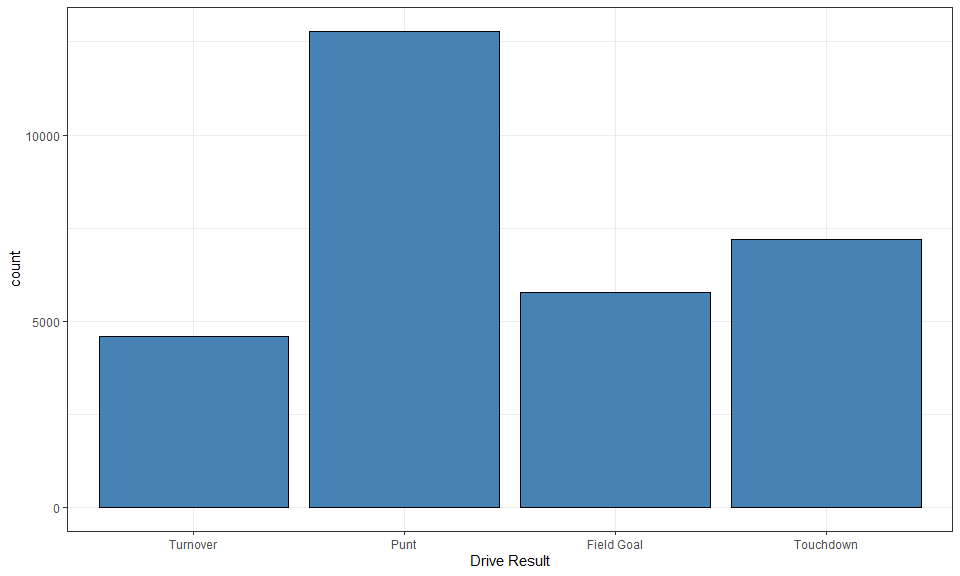
Use the NFL Drive data set contains how an NFL drive begins and ends for 30374 drives for the 2016 – 2021 NFL seasons. A “drive” in football is a series of plays where the same team is on offense and ends when the teams switch being on offense or defense.

### Import the NFL Drive data set below:  
nfl <- read.csv("NFL Drive.csv")  
  
  
# Run the code below after reading in your data to change the order of the groups to something more preferable:  
nfl <-   
 nfl %>%   
   
 filter(complete.cases(.)) %>%   
   
 mutate(drive\_start = factor(drive\_start,   
 levels = c("Kickoff", "Punt",   
 "Fumble", "Interception")),  
   
 drive\_end = factor(drive\_end,  
 levels = c("Turnover", "Punt",  
 "Field Goal", "Touchdown")))

### Part 2a) Bar Chart for Drive End

Create a single bar chart drive\_end with the counts for each outcome on the y-axis. Have the bars all be the same color other than the default color and the x-axis label of “Drive Result”.

# Create the bar chart for drive\_end below.  
ggplot(data = nfl,  
 mapping = aes(x = drive\_end)) +  
   
 geom\_bar(color = "black",  
 fill = "steelblue") +  
   
 labs(x = "Drive Result")

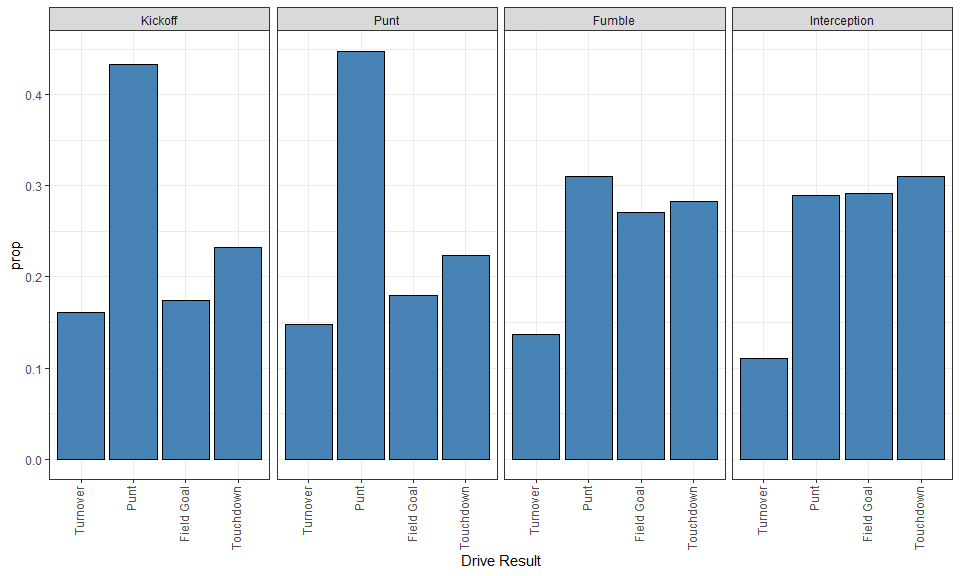


### Part 2b) Bar Chart by Drive Beginning

Create 4 bar charts for drive\_end: One for each way a drive can begin (drive\_start). Have the proportion displayed on the y-axis instead of the counts and all 4 charts in the same row.

Adjust the plots so the x-axis labels can be seen properly (necessary for most of the graphs). You might use this code: + theme(axis.text.x = element\_text(angle = 90, hjust = 1, vjust = 0.25)) (you can adjust the values in element\_text if you like).

# Write you code to create the bar charts of drive result with small multiples for each drive beginning and the proportion on the x-axis  
ggplot(data = nfl,  
 mapping = aes(x = drive\_end)) +  
   
 geom\_bar(mapping = aes(group = 1,  
 y = ..prop..),  
 fill = "steelblue",  
 color = "black") +   
   
 labs(x = "Drive Result") +  
   
 facet\_wrap(facets = ~ drive\_start,  
 nrow = 1) +   
   
 theme(axis.text.x = element\_text(angle = 90,  
 hjust = 1,  
 vjust = 0.25))



### Part 2c) Drive Result by Drive Start

Does it appear that teams are more likely to score (drive\_end = Touchdown or Field Goal) depending on how the drive started? Explain your answer just based on the graphs from part b).

It does appear that how drives begin do have an effect on how often a drive ends in a score.

The results of a drive if it begins from a punt or a kickoff appear to be about the same. Drives started after an interception or fumble also appear to have similar results.

Drives that start after the opponent turns the ball over (fumble or interception) are more likely to end in a scoring play than drives that being after a kickoff or punt.