# Problem Set 2

Multivariate Visualization, Uncertainty, and Regression

[YOUR NAME]

Due Date: 2025-06-13

### Getting Set Up

Open `RStudio` and create a new RMarkDown file (`.Rmd`) by going to `File -> New File -> R Markdown...`. Accept defaults and save this file as `[LAST NAME]_ps2.Rmd` to your `code` folder.

Copy and paste the contents of this `.Rmd` file into your `[LAST NAME]_ps2.Rmd` file. Then change the `author: [Your Name]` to your name.

All of the following questions should be answered in this `.Rmd` file. There are code chunks with incomplete code that need to be filled in. To submit, compile (i.e., `knit as pdf`) the completed problem set and upload the PDF file to Blackboard on Friday by midnight. Be sure to check your knitted PDF for mistakes before submitting!

This problem set is worth 27 total points, plus 3.5 extra credit points. The point values for each question are indicated in brackets below. To receive full credit, you must have the correct code. In addition, some questions ask you to provide a written response in addition to the code.

You will be deducted 1 point for each day late the problem set is submitted, and 1 point for failing to submit in the correct format (i.e., not knitting as a PDF).

You are free to rely on whatever resources you need to complete this problem set, including lecture notes, lecture presentations, Google, your classmates... you name it. However, the final submission must be complete by you. There are no group assignments.

*Note that the professor will not respond to Campuswire posts after 2PM on Friday, so don't wait until the last minute to get started!*

**Good luck!**

**If you collaborated with a colleague and/or used AI for any help on this problem set, document here.** Write the names of your classmates and/or upload a PDF of your AI prompt and output with your problem set:

## Part 1: 2020 Presidential Election (10 points; +1 extra credit)

### Question 0 (0 points)

*Require `tidyverse` and load the `Pres2020_PV.Rds` data to an object called `pres`.*

```
require()
```

```
## Loading required package:
```

```
pres <- read_rds()
```

```
## Error in read_rds(): could not find function "read_rds"
```

## Question 1 [2 points]

*Consider the following hypothesis: "Most Americans don't pay very much attention to politics, and don't know who they will vote for until very close to the election. Therefore polling predictions should be more accurate closer to the election." Based on this hypothesis and theoretical intuition, which variable is the X variable and which is the Y variable(s)?*

- Write answer here

*Now let's first look at each variable by itself using univariate visualization. First, plot the total number of polls per start date in the data. NB: you will have convert* `StartDate` *to a* `date` *class with* `as.Date()`*. If you need help, see this post. Do you observe a pattern in the number of polls over time? Why do you think this is?*

```
pres %>%
  mutate(StartDate = as.Date(StartDate,'%m/%d/%Y')) %>% # Convert to date
  ggplot(aes(x = StartDate)) + # Visualize the variable using univariate principles
  geom_...() + # Choose the correct `geom`
  labs() # Make sure it is clearly labeled
```

```
## Error in pres %>% mutate(StartDate = as.Date(StartDate, "%m/%d/%Y")) %>% : could not find function "
```

- Write answer here

## Question 2 [2 points]

*Next, let's look at the other variables. Calculate the* **prediction error** *for Biden (call this variable* `demErr`*) and Trump (call this variable* `repErr`*) such that positive values mean that the poll* **overestimated** *the candidate's popular vote share (*`DemCertVote` *for Biden and* `RepCertVote` *for Trump).*

```
pres <- pres %>%
  mutate() # Create the two new variables
```

```
## Error in pres %>% mutate(): could not find function "%>%"
```

*Plot the Biden and Trump prediction errors on a single plot using* `geom_bar()`*, with red indicating Trump and blue indicating Biden (make sure to set alpha to some value less than 1 to increase the transparency!). Add vertical lines for the average prediction error for both candidates (colored appropriately) as well as a vertical line indicating no prediction error.*

```
pres %>%
  ggplot() + # Instantiate an EMPTY ggplot object
  geom_bar(aes(...), # Put the first variable in the first `geom_bar()`
           ...) + # Set the color and opacity
  geom_bar(aes(...), # Put the second variable in the second `geom_bar()`
```

```
          ...) + # Set the color and opacity
  labs(...) + # Make sure it is clearly labeled
  geom_vline(...) + # Put a black vertical line at 0
  geom_vline() + # Put a dashed blue vertical line at the Democrat prediction error
  geom_vline() + # Put a dashed red vertical line at the Republican prediction error
```

```
## Error in parse(text = input): <text>:11:0: unexpected end of input
## 9:   geom_vline() + # Put a dashed blue vertical line at the Democrat prediction error
## 10:   geom_vline() + # Put a dashed red vertical line at the Republican prediction error
##    ^
```

*Do you observe a systematic bias toward one candidate or the other?*

- Write answer here

## Question 3 [2 points]

*Plot the average prediction error for Trump (red) and Biden (blue) by start date using* ***geom_point()*** *and add two curvey lines of best fit using* ***geom_smooth()****. Make sure that the curvey line for Trump is also red, and the curvey line for Biden is also blue!*

```
pres %>%
  mutate(...) %>% # Convert to date
  group_by(...) %>% # Calculate the average error for Biden and Trump by date
  summarise(...,
            ...) %>%
  ggplot() + # Instantiate an empty ggplot
  geom_point(aes(x = ...,y = ...), # Put the first variable in the first `geom_point()`
           ...) + # Set the color
  geom_point(aes(x = ...,y = ...), # Put the second variable in the second `geom_point()`
           ...) + # Set the color
  geom_smooth(aes(x = ...,y = ...), # Put the first variable in the first geom_smooth()
              ...) + # Set the color
  geom_smooth(aes(x = ...,y = ...), # Put the second variable in the second geom_smooth()
              ...) + # Set the color
  labs(...) + # Make sure it is clearly labeled
  geom_hline(...) # Add a horizontal dashed line at 0
```

```
## Error in pres %>% mutate(...) %>% group_by(...) %>% summarise(..., ...) %>% : could not find functio
```

*What pattern do you observe over time, if any? Does this support the hypothesis presented in Question 1 above?*

- Write answer here

## Question 4 [2 points]

*Can we do better by aggregating state-level polls? Load the [Pres2020_StatePolls.Rds] to an object called* ***state****. First, create two new variables* ***demErr*** *and* ***repErr*** *just as you did in Question 2. Then recreate the same overtime plot comparing Biden and Trump prediction errors as you did in Question 3. What do you observe?*

```
state <- read_rds(...) # Read in the data
```

```
## Error in read_rds(...): could not find function "read_rds"
```

```
state <- state %>%
  mutate(...) # Create the two new variables for Democrat and Republican prediction errors
```

```
## Error in state %>% mutate(...): could not find function "%>%"
```

```
state %>%
  mutate(...) %>% # Convert to date
  group_by(...) %>% # Calculate the average error for Biden and Trump by date
  summarise(...,
            ...) %>%
  ggplot() + # Instantiate an empty ggplot
  geom_point(aes(x = ...,y = ...), # Put the first variable in the first `geom_point()`
             ...) + # Set the color
  geom_point(aes(x = ...,y = ...), # Put the second variable in the second `geom_point()`
             ...) + # Set the color
  geom_smooth(aes(x = ...,y = ...), # Put the first variable in the first geom_smooth()
              ...) + # Set the color
  geom_smooth(aes(x = ...,y = ...), # Put the second variable in the second geom_smooth()
              ...) + # Set the color
  labs(...) + # Make sure it is clearly labeled
  geom_hline(...) # Add a horizontal dashed line at 0
```

```
## Error in state %>% mutate(...) %>% group_by(...) %>% summarise(..., ...) %>% : could not find functi
```

- Write answer here

## Question 5 [2 points]

*One other explanation for inaccurate state polls is that some states do not have many polls run. Calculate the anti-Trump/pro-Biden bias for each state by subtracting the* ***repErr*** *from the* ***demErr*** *(call this new variable* ***bidenBias****). Then calculate the average bias by state AND calculate the number of polls in that state. Finally, plot the relationship between the number of polls and the extent of bias. Does the data support the theory that states with more polls were predicted more accurately?*

```
state <- state %>%
  mutate(...) # Create the bidenBias variable here
```

```
## Error in state %>% mutate(...): could not find function "%>%"
```

```
state %>%
  group_by(...) %>%
  summarise(...,   # Calculate the average bidenBias by state
            ...) %>% # Calculate the number of polls by state
  ungroup() %>%
  ggplot(aes(x = ...,      # Put the correct variable on the x-axis
```

```
            y = ...)) +   # Put the correct variable on the y-axis
  geom_...() + # Choose the correct geom
  geom_...(...) + # Add a straight line of best fit
  labs(...) # Give it some good labels
```

```
## Error in state %>% group_by(...) %>% summarise(..., ...) %>% ungroup() %>% : could not find function
```

- Write answer here

### Extra Credit [1 point]

*Do polls that underestimate Trump's support overestimate Biden's support? Investigate this question using both the national data (`pres`) and the state data (`state`). Use a scatterplot to test, combined with a (straight) line of best fit. Then, calculate the proportion of polls that (1) underestimate both Trump and Biden, (2) underestimate Trump and overestimate Biden, (3) overestimate Trump and underestimate Biden, (4) overestimate both candidates. In these analyses, define "overestimate" as prediction errors greater than or equal to zero, whereas "underestimate" should be prediction errors less than zero. What do you conclude? Is there any evidence of an anti-Trump bias in national polling? What about state polling?*

```
# National scatterplot
# INSERT CODE HERE

# National proportions: 4 different types of polls
# INSERT CODE HERE

# State scatterplot
# INSERT CODE HERE

# State proportions: 4 different types of polls
# INSERT CODE HERE
```

- Write answer here

## Part 2: New York/Villanova Knicks (8 points; 1 extra credit point)

### Question 0

Require `tidyverse` and load the `game_summary.rds` data to an object called `games`.

```
# INSERT CODE HERE
```

### Question 1 [2 points]

How many points, on average, did the New York Knicks score at home and away games in the 2017 season? Calculate this answer and also plot the multivariate relationship. Explain why your chosen visualization is justified. Draw two vertical lines for the average points at home and away.

```
# Create extra object to plot vertical lines for average points at home and away
vertLines <- games %>%
filter() %>% # Filter to the 2017 season (yearSeason) AND to the New York Knicks (nameTeam)
  group_by() %>% # Group by the location of the game
  summarise() # Calculate the average points (pts)
```

```
## Error in games %>% filter() %>% group_by() %>% summarise(): could not find function "%>%"
```

```
games %>%
  filter() %>% # Filter to the 2017 season (yearSeason) AND to the New York Knicks (nameTeam)
  ggplot() + # Create a multivariate plot comparing points scored between home and away games
  geom_...() + # Choose the appropriate geom_... for this plot (i.e., geom_histogram(), geom_density(),
  labs(title = '', # Add clear descriptions for the title, subtitle, axes, and legend
       subtitle = '',
       x = '',
       y = '',
       color = '') +
  geom_vline() # add vertical lines for the average points scored at home and away.
```

```
## Error in games %>% filter() %>% ggplot(): could not find function "%>%"
```

Write answer here

## Question 2 [2 points]

Now recreate the same plot for the 2018, 2019, and combined seasons. Imagine that you work for the Knicks organization and Scott Perry (the GM), asks you if the team scores more points at home or away? Based on your analysis, what would you tell him?

```
# By season
vertLines <- games %>%
filter() %>% # Filter to the New York Knicks (nameTeam)
  group_by() %>% # Group by the location and the season
  summarise() # Calculate the average points (pts)
```

```
## Error in games %>% filter() %>% group_by() %>% summarise(): could not find function "%>%"
```

```
games %>%
  filter() %>% # Filter to the New York Knicks (nameTeam)
  ggplot() + # Create a multivariate plot comparing points scored between home and away games
  geom_...() + # Choose the appropriate geom_... for this plot (i.e., geom_histogram(), geom_density(),
  labs(title = '', # Add clear descriptions for the title, subtitle, axes, and legend
       subtitle = '',
       x = '',
       y = '',
       color = '') +
  facet_wrap() + # Create separate panels for each season (facet_wrap())
  geom_vline() # add vertical lines for the average points scored at home and away.
```

```
## Error in games %>% filter() %>% ggplot(): could not find function "%>%"
```

```
# Over all seasons combined
vertLines <- games %>%
filter() %>% # Filter to the New York Knicks (nameTeam)
  group_by() %>% # Group by the location
  summarise() # Calculate the average points (pts)
```

```
## Error in games %>% filter() %>% group_by() %>% summarise(): could not find function "%>%"
```

```
games %>%
  filter() %>% # Filter to the New York Knicks (nameTeam)
  ggplot() + # Create a multivariate plot comparing points scored between home and away games
  geom_...() + # Choose the appropriate geom_... for this plot (i.e., geom_histogram(), geom_density(),
  labs(title = '', # Add clear descriptions for the title, subtitle, axes, and legend
       subtitle = '',
       x = '',
       y = '',
       color = '') +
  geom_vline() # add vertical lines for the average points scored at home and away.
```

```
## Error in games %>% filter() %>% ggplot(): could not find function "%>%"
```

Write answer here

## Question 3 [2 points]

Scott Perry thanks you for your answer, but is a well-trained statistician in his own right, and wants to know how confident you are in your claim. Bootstrap sample the data 1,000 times to provide him with a more sophisticated answer. How confident are you in your conclusion that the Knicks score more points at home games than away games? Make sure to `set.seed(123)` to ensure you get the same answer every time you `knit` your code!

```
set.seed(123) # Set the seed!
forBS <- games %>% # To make things easier, create a new data object that is filtered to just the Knick
    filter() # Filter to the Knicks (nameTeam)
```

```
## Error in games %>% filter(): could not find function "%>%"
```

```
bsRes <- NULL # Instantiate an empty object to store data from the loop
for(i in 1:1000) { # Loop 1,000 times
  bsRes <- forBS %>%
    sample_n() %>% # Sample the data with replacement using all possible rows
    group_by() %>% # Group by the location of the game
    summarise() %>% # Calculate the average points (pts)
    ungroup() %>% # Best practices!
    spread() %>% # Spread the data to get one column for average points at home and another for average
    mutate(, # Calculate the difference between home and away points
           ) %>% # Save the bootstrap index
    bind_rows(bsRes) # Append the result to the empty object from above
}
```

```
## Error in forBS %>% sample_n() %>% group_by() %>% summarise() %>% ungroup() %>% : could not find funct
```

```
# Calculate the confidence
bsRes %>%
  summarise(, # Calculate the proportion of bootstrap simulations where the home points are greater tha
            ) # Calculate the overall average difference
```

```
## Error in bsRes %>% summarise(, ): could not find function "%>%"
```

Write answer here

## Question 4 [2 points]

Re-do this analysis for three other statistics of interest to Scott: total rebounds (treb), turnovers (tov), and field goal percent (pctFG). Do you notice anything strange in these results? What might explain it?

```
bsRes <- NULL # Instantiate an empty object to store data from the loop
for(i in 1:1000) { # Loop 1,000 times
  bsRes <- forBS %>%
    sample_n() %>% # Sample the data with replacement using all possible rows
    group_by() %>% # Group by the location of the game
    summarise(, # Calculate the average total rebounds (treb)
              , # Calculate the average turnovers (tov)
              ) %>% # Calculate the average field goal shooting percentage (pctFG)
    ungroup() %>% # Best practices!
    pivot_wider(, # Pivot wider to get each measure in its own column for home and away games
                ) %>% # Use the values from the variables you created above
    mutate(, # Calculate the difference between home and away total rebounds
           , # Calculate the difference between home and away turnovers
           , # Calculate the difference between home and away field goal percentages
           ) %>% # Save the bootstrap index
    bind_rows(bsRes) # Append the result to the empty object from above
}
```

```
## Error in forBS %>% sample_n() %>% group_by() %>% summarise(, , ) %>% ungroup() %>% : could not find
```

```
# Calculate the confidence
bsRes %>%
  summarise(, # Calculate the confidence for rebounds being greater than zero
            , # Calculate the confidence for turnovers being greater than zero
            )
```

```
## Error in bsRes %>% summarise(, , ): could not find function "%>%"
```

Write answer here

## Extra Credit [1 point]

Now Scott is asking for a similar analysis of other teams. Calculate the difference between home and away points for every team in the league and prepare a summary table that includes both the average difference for each team, as well as your confidence about whether the difference is not zero. Based on these data, would you argue that there is an **overall** home court advantage in terms of points across the NBA writ

large? Visualize these summary results by plotting the difference on the x-axis, the teams (reordered) on the y-axis, and the points colored by whether you are more than 90% confident in your answer. How should we interpret confidence levels less than 50%?

```
# INSERT CODE HERE
```

Write answer here

# Part 3: Youtube Bias (9 points; 1.5 extra credit points)

We will be using a new dataset called `youtube_individual.rds` which can be found on the course github page. The codebook for this dataset is produced below. All ideology measures are coded such that negative values indicate more liberal content and positive values indicate more conservative content.

| Name | Description |
|---|---|
| ResponseId | A unique code for each respondent to the survey |
| ideo_recommendation | The average ideology of all recommendations shown to the respondent |
| ideo_current | The average ideology of all current videos the respondent was watching when they were shown recommendations |
| ideo_watch | The average ideology of all videos the respondent has ever watched on YouTube (their "watch history") |
| nReccs | The total number of recommendations the respondent was shown during the survey |
| YOB | The year the respondent was born |
| education | The respondent's highest level of education |
| gender | The respondent's gender |
| income | The respondent's total household income |
| party_id | The respondent's self-reported partisanship |
| ideology | The respondent's self-reported ideology |
| race | The respondent's race |
| age | The respondent's age at the time of the survey |

## Question 0

*Require* `tidyverse` *and load the* `youtube_individual.rds` *data to an object called* `yt`*.*

```
# INSERT CODE HERE
```

## Question 1 [1 point]

*We are interested in how the YouTube recommendation algorithm works. These data are collected from real users, logged into their real YouTube accounts, allowing us to see who gets recommended which videos. We will investigate three research questions in this problem set:*

*1. What is the relationship between average ideology of recommendations shown to each user, and the average ideology of all the videos the user has watched?*

*2. What is the relationship between the average ideology of recommendations shown to each user, and the average ideology of the current video the user was watching when they were shown the recommendation?*

*3. Which of these relationships is stronger? Why?*

*Start by answering all three of these research questions, and explaining your thinking. Be very precise about your assumptions!*

Write answer here

## Question 2 [2 points]

*Based on your previous answer, which variables are the X (predictors) and which are the Y (outcome) variables?*

Write answer here

*Now create univariate visualizations of all three variables, making sure to label your plots clearly.*

```
# Univariate visualization of Y
# INSERT CODE HERE

# Univariate visualization of X1
# INSERT CODE HERE

# Univariate visualization of X2
# INSERT CODE HERE
```

## Question 3 [2 points]

*Let's focus on the first research question. Create a multivariate visualization of the relationship between these two variables, making sure to put the X variable on the x-axis, and the Y variable on the y-axis. Add a straight line of best fit. Does the data support your theory?*

```
# Multviariate visualization of Y and X1
# INSERT CODE HERE
```

Write answer here

*Now run a linear regression using the `lm()` function and save the result to an object called `model_watch`.*

```
model_watch <- lm(formula = ..., # Write the regression equation here (remember to use the tilde ~!)
             data = ...) # Indicate where the data is stored here.
```

```
## Error: '...' used in an incorrect context
```

*Using either the `summary()` function (from base `R`) or the `tidy()` function (from the `broom` package), print the regression result.*

```
require(broom)
```

```
## Loading required package: broom
```

```
tidy(model_watch)
```

```
## Error: object 'model_watch' not found
```

*In a few sentences, summarize the results of the regression output. This requires you to translate the statistical measures into plain English, making sure to refer to the units for both the $X$ and $Y$ variables. In addition, you must determine whether the regression result supports your hypothesis, and discuss your confidence in your answer, referring to the p-value.*

Write answer here

## Question 4 [2 points]

*Now let's do the same thing for the second research question. First, create the multivariate visualization and determine whether it is consistent with your theory.*

```
# Multviariate visualization of Y and X2
# INSERT CODE HERE
```

Write answer here

*Second, run a new regression and save the result to* `model_current`*. Then print the result using either* `summary()` *or* `tidy()`*, as before.*

```
# [RUBRIC: 0.25 points - either right or wrong.]
model_current <- lm(formula = ..., # Write the regression equation here (remember to use the tilde ~!)
              data = ...) # Indicate where the data is stored here.
```

```
## Error: '...' used in an incorrect context
```

```
tidy(model_current)
```

```
## Error: object 'model_current' not found
```

*Finally, describe the result in plain English, and interpret it in light of your hypothesis. How confident are you?*

Write answer here

*Based* **ONLY** *on the preceding analysis, are you able to answer research question 3?*

Write answer here

## Question 5 [2 points + 0.5 EC points]

*Now let's evaluate the models. Start by calculating the "mistakes" (i.e., the "errors" or the "residuals") generated by both models and saving these as new columns (*`errors_watch` *and* `errors_current`*) in the* `yt` *dataset.*

```
# Calculating errors
yt <- yt %>%
  mutate(preds_watch = ..., # Get the predicted values from the first model (Yhat)
         preds_current = ...) %>% # Get the predicted values from the second model (Yhat)
  mutate(errors_watch = ..., # Calculate errors for the first model (Y - Yhat)
         errors_current = ...) # Calculate errors for the second model (Y - Yhat)
```

```
## Error in yt %>% mutate(preds_watch = ..., preds_current = ...) %>% mutate(errors_watch = ..., : coul
```

*Now create two univariate visualization of these errors. Based on this result, which model looks better? Why? EC [+1 point]: Plot both errors on the same graph using `pivot_longer()`.*

```
# Univariate visualization of watch history model errors
yt %>%
  ggplot(aes(x = ...)) + # Put the errors from the first model on the x-axis
  geom_...() +  # Choose the best geom_...() to visualize based on the variable's type
  labs(...) # Provide clear labels to help a stranger understand!
```

```
## Error in yt %>% ggplot(aes(x = ...)): could not find function "%>%"
```

```
# Univariate visualization of current video model errors
yt %>%
  ggplot(aes(x = ...)) + # Put the errors from the first model on the x-axis
  geom_...() +  # Choose the best geom_...() to visualize based on the variable's type
  labs(...) # Provide clear labels to help a stranger understand!
```

```
## Error in yt %>% ggplot(aes(x = ...)): could not find function "%>%"
```

```
# EC [0.5 points]: Plot both errors on a single plot. Hint: use pivot_longer().
```

Write answer here

*Finally, create a multivariate visualization of both sets of errors, comparing them against the $X$ variable. Based on this result, which model looks better? Why? EC [+1 point]: Create two plots side-by-side using facet_wrap(). This is SUPER HARD, so don't worry if you can't get it.*

```
# Multivariate visualization of watch history errors
yt %>%
  ggplot(aes(x = ...,       # Put the predictor on the x-axis
             y = ...)) + # Put the errors on the y-axis
  geom_...() + # Choose the best geom_...()
  geom_...() + # Add a curvey line of best fit
  geom_hline(...) + # Add a horizontal dashed line at zero
  labs(...) # Provide clear labels to help a stranger understand!
```

```
## Error in yt %>% ggplot(aes(x = ..., y = ...)): could not find function "%>%"
```

```
# Multivariate visualization of current video errors
yt %>%
  ggplot(aes(x = ...,       # Put the predictor on the x-axis
             y = ...)) + # Put the errors on the y-axis
  geom_...() + # Choose the best geom_...()
  geom_...() + # Add a curvey line of best fit
  geom_hline(...) + # Add a horizontal dashed line at zero
  labs(...) # Provide clear labels to help a stranger understand!
```

```
## Error in yt %>% ggplot(aes(x = ..., y = ...)): could not find function "%>%"
```

```
# EC [1 point]: Try to create two plots side-by-side. (SUPER HARD)
```

Write answer here

## Extra Credit [1 point]

*Calculate the* **R***oot* **M***ean* **S***quared* **E***rror (RMSE) using 100-fold cross validation with a 50-50 split for both models. How bad are the first model's mistakes on average? How bad are the second model's mistakes? Which model seems better? Remember to talk about the result in terms of the range of values of the outcome variable! Plot the errors by the model using geom_boxplot(). HINT: you'll need to use pivot_longer() to get the data shaped correctly.*

```
# INSERT CODE HERE
```

Write answer here