Problem Set 4

Regression Part 1

[YOUR NAME]

Due Date: 2024-07-19

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 `[YOUR NAME]_ps4.Rmd` to your `code` folder.

Copy and paste the contents of this `.Rmd` file into your `[YOUR NAME]_ps4.Rmd` file. Then change the `author: [Your Name]` on line 2 to your name.

We will be using a new dataset called `youtube_individual.rds` which can be found on the course github page (https://github.com/jbisbee1/ISP_Data_Science_2024/raw/main/data/youtube_individual.rds). 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

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.

This problem set is worth 10 total points, plus **two** extra credit questions, each worth **two** points. In addition, there are additional opportunities for extra credit totaling **another 4 points**. Note that these additional EC opportunities are **very hard**. They are designed for the students who claimed that the course is easy on the midterm evals. I encourage you all to attempt all extra credit, but don't worry if you don't get the super hard ones.

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 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. To submit, email the knitted output to Eun Ji Kim (kej990804@snu.ac.kr (mailto:kej990804@snu.ac.kr)) **as a PDF** by the start of class on Friday, July 12th. If you need help converting to a PDF, see this tutorial (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/Psets/ISP_pset_0_HELPER.pdf).

Good luck!

*Copy the link to ChatGPT you used here: _____

Question 0

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

```
require(tidyverse)
```

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

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts t
o become errors
```

```
yt <- read_rds("https://github.com/jbisbee1/ISP_Data_Science_2024/raw/main/data/youtube_
individual.rds")
```

Question 1 [2 points]

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?

Start by answering all three of these research questions, and explaining your thinking.

> I assume that the recommendation algorithm is designed to suggest videos that users will likely watch. I assume that the algorithm learns about what the user likes to watch on the basis of what they are currently watching, and on the basis of what they historically have chosen to watch. Thus for both questions 1 and 2, I hypothesize that the relationship between the average ideology of recommendations and the average ideology of either the current video or the user's watch history are both positive. This means that users who are currently watching conservative videos will be recommended more conservative videos, and that users who historically watch conservative videos will be recommended more conservative videos. However, I think that the user's watch history contains more information about their preferences, and therefore should be the stronger relationship.

Question 2 [1 point]

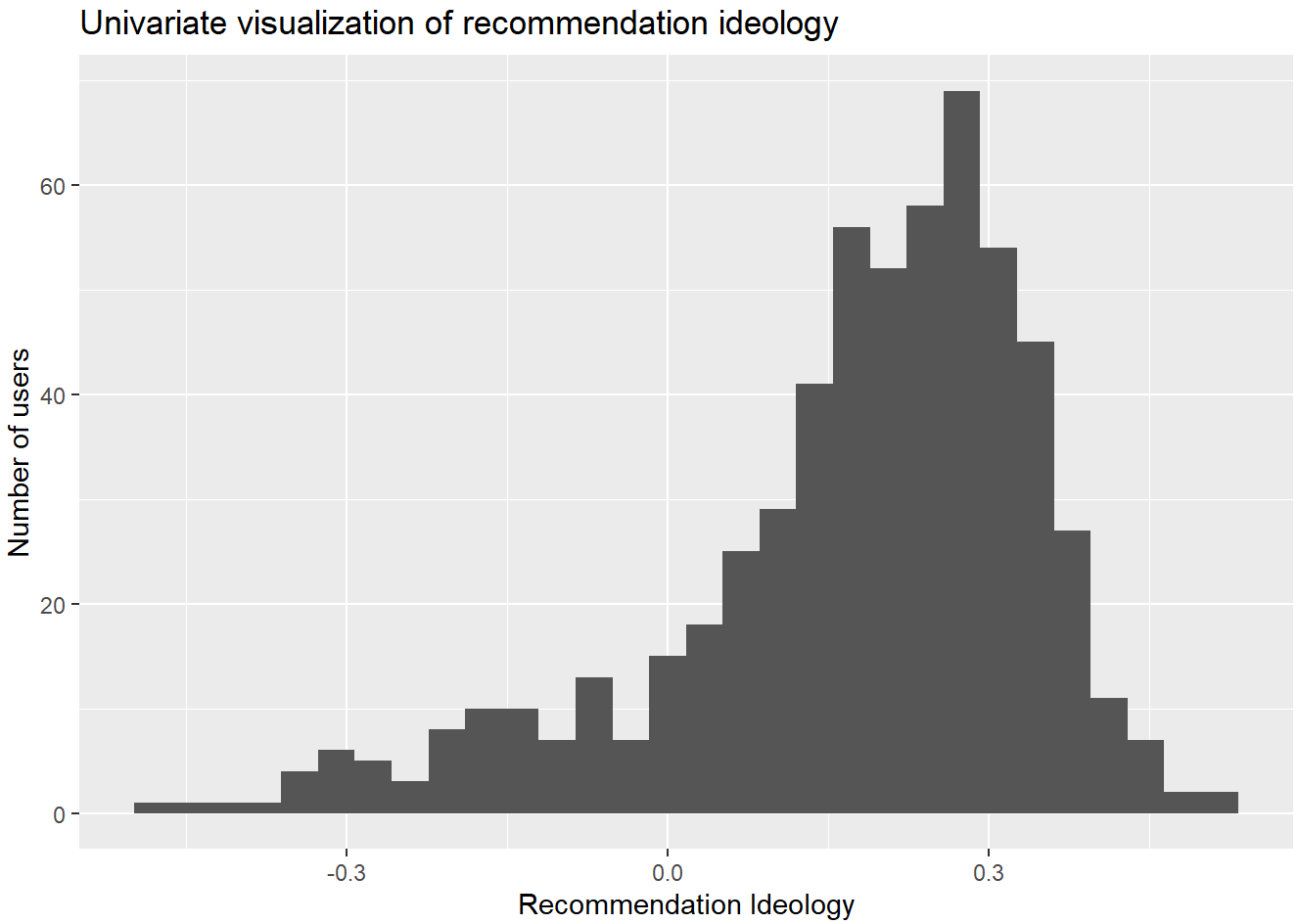
Based on your previous answer, which variables are the X (predictors) and which are the Y (outcome) variables?

> The predictors are the current video ideology and the watch history ideology. The outcome variable is the recommendation ideology.

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

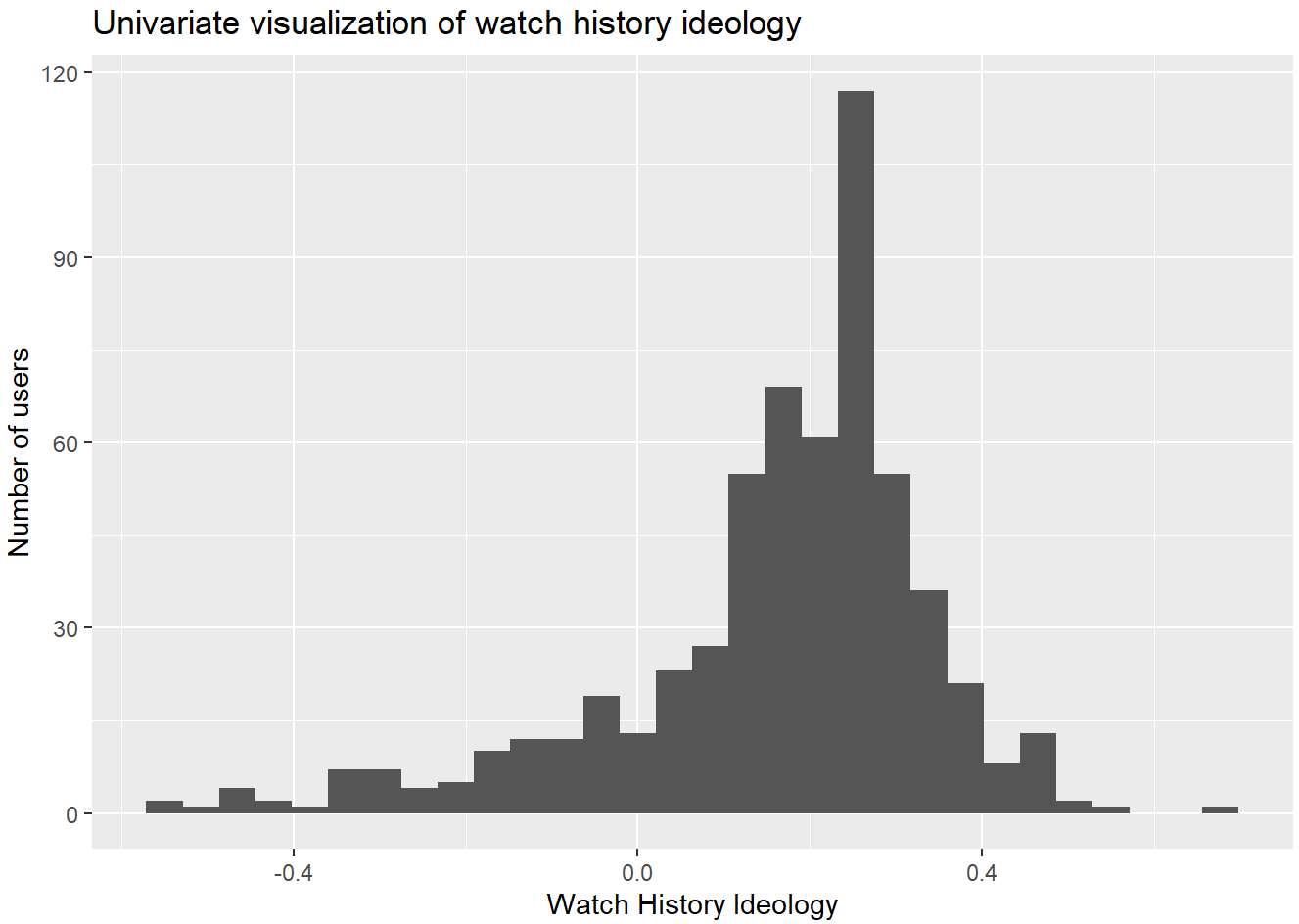
```
# Y: average_recommendation_ideo
yt %>%
  ggplot(aes(x = ideo_recommendation)) + # Put the outcome variable on the x-axis
  geom_histogram() +  # Choose the best geom_...() to visualize based on the variable's type
  labs(x = 'Recommendation Ideology', # Provide clear labels to help a stranger understand!
       y = 'Number of users',
       title = 'Univariate visualization of recommendation ideology')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



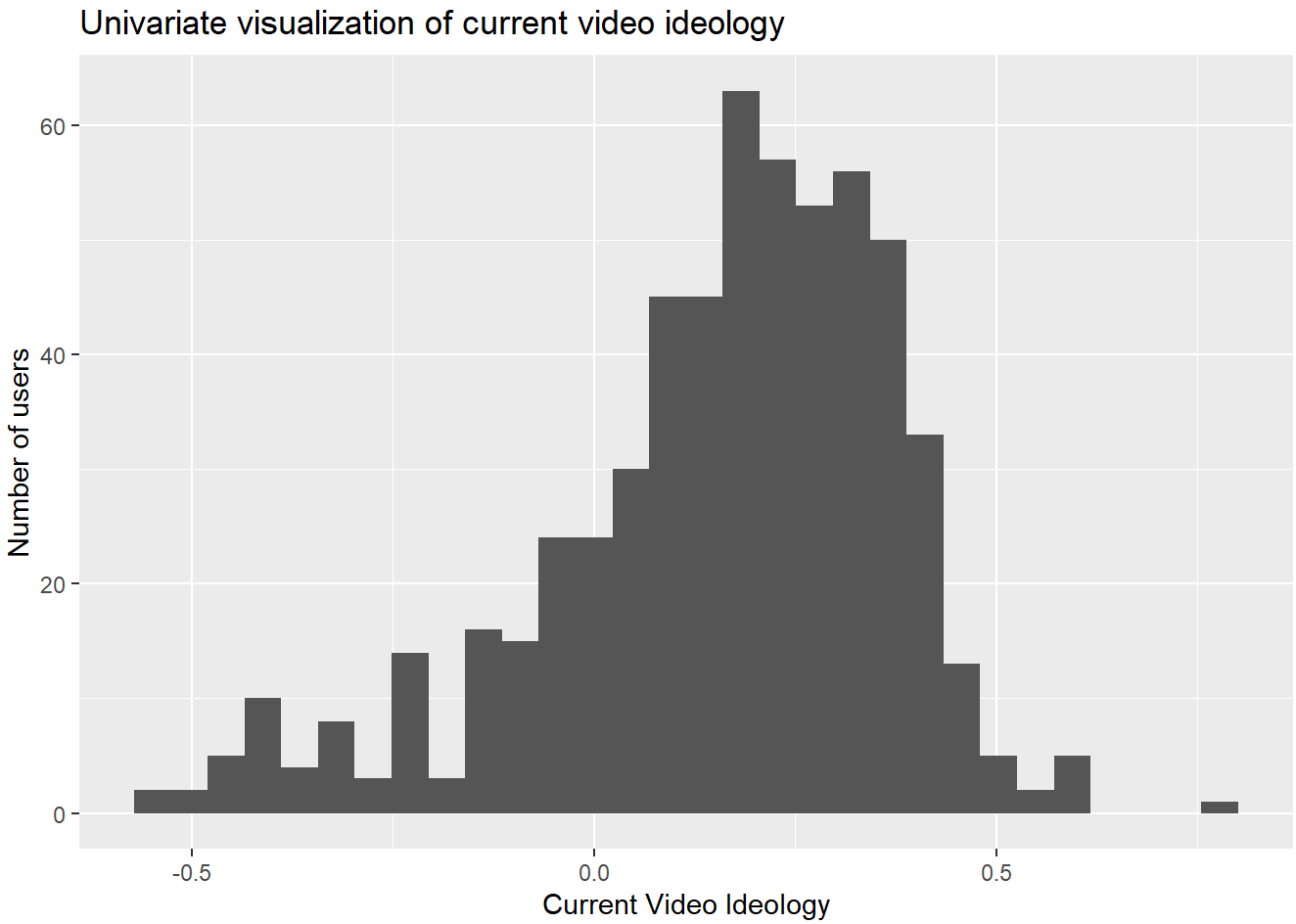

```
# X2: average_watch_ideo
yt %>%
  ggplot(aes(x = ideo_watch)) + # Put the first explanatory variable on the x-axis
  geom_histogram() +  # Choose the best geom_...() to visualize based on the variable's type
  labs(x = 'Watch History Ideology', # Provide clear labels to help a stranger understand!
       y = 'Number of users',
       title = 'Univariate visualization of watch history ideology')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# X2: average_current_ideo
yt %>%
  ggplot(aes(x = ideo_current)) + # Put the second explanatory variable on the x-axis
  geom_histogram() +  # Choose the best geom_...() to visualize based on the variable's type
  labs(x = 'Current Video Ideology', # Provide clear labels to help a stranger understand!
       y = 'Number of users',
       title = 'Univariate visualization of current video ideology')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

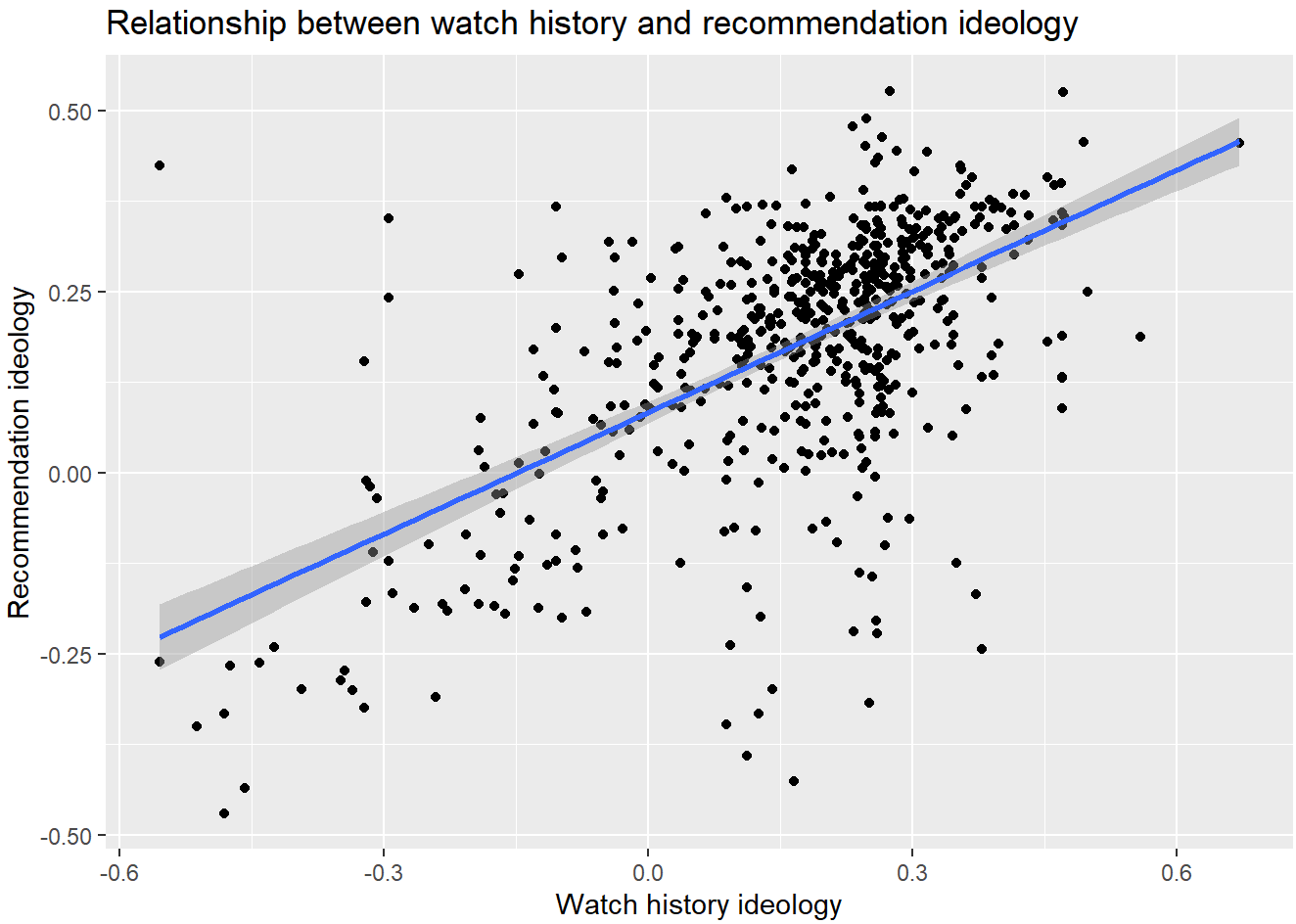


Question 3 [1 point]

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?

```
yt %>%
  ggplot(aes(x = ideo_watch,  # Put the first explanatory variable on the x-axis
             y = ideo_recommendation)) +  # Put the outcome variable on the y-axis
  geom_point() + # Choose the best geom_...() for visualizing this type of multivariate relationship
  geom_smooth(method = 'lm') +  # Add a straight line of best fit
  labs(x = 'Watch history ideology', # Provide clear labels to help a stranger understand!
       y = 'Recommendation ideology',
       title = 'Relationship between watch history and recommendation ideology')
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Yes the data strongly supports my theory. There is clear evidence of a positive relationship between a user's watch history ideology and the recommendation ideology they are being shown.

Question 4 [1 point]

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

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

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)
```

```
## # A tibble: 2 × 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   0.0834   0.00771      10.8 5.74e-25
## 2 ideo_watch    0.558    0.0313       17.8 3.31e-57
```

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.

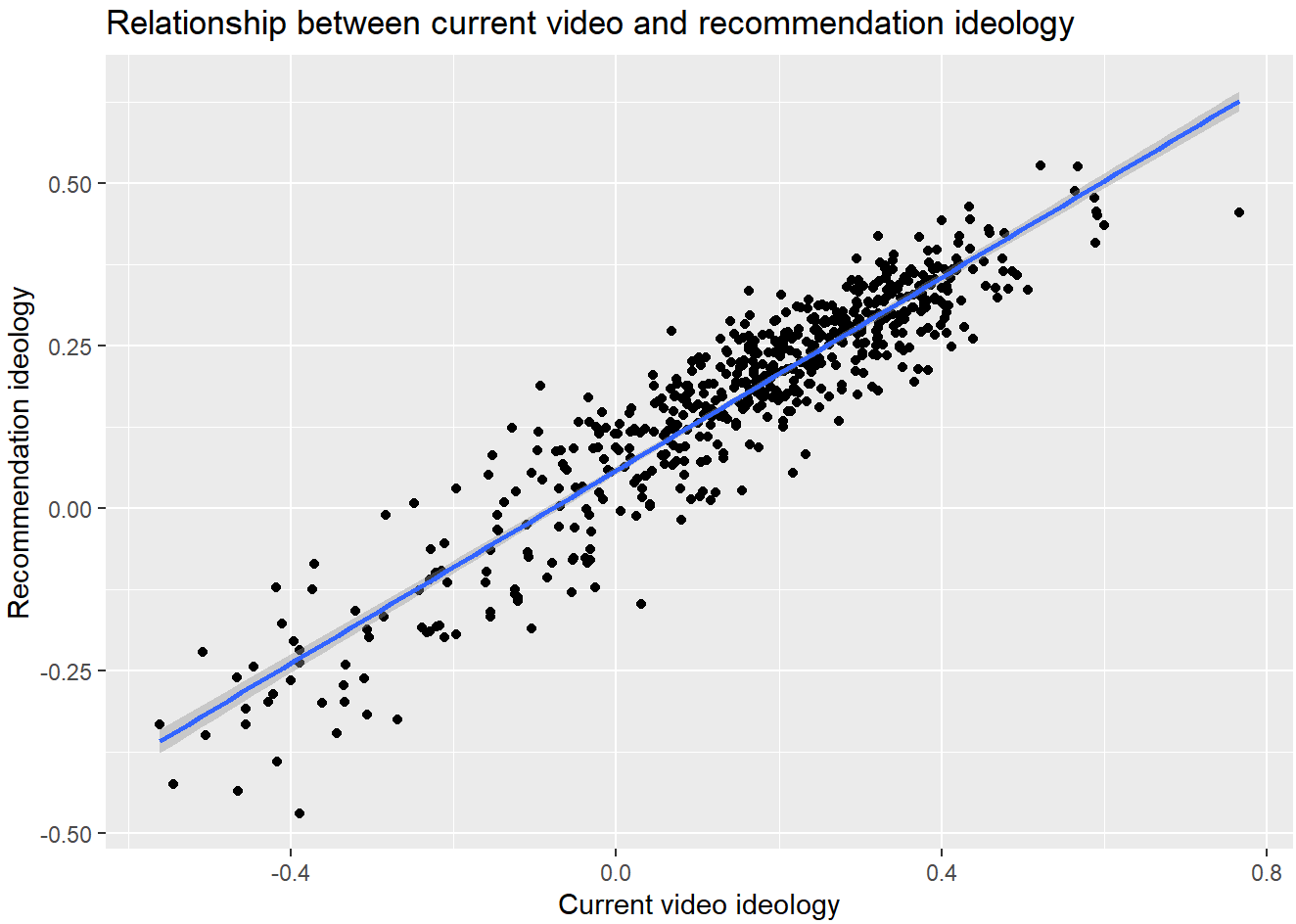
> According to this model, the average ideology of a user's recommendations is 0.08 when their watch history ideology is zero, meaning a totally moderate watch history. The model also tells us that the average ideology of recommendations shown to users increases by 0.558 units when their watch history ideology increases by 1 unit. I am more than 99.99% confident that this positive relationship is not zero, meaning it is a strong relationship.

Question 5 [1 point]

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.

```
yt %>%
  ggplot(aes(x = ideo_current,  # Put the second explanatory variable on the x-axis
             y = ideo_recommendation)) +  # Put the outcome variable on the y-axis
  geom_point() + # Choose the best geom_...() for visualizing this type of multivariate relationship
  geom_smooth(method = 'lm') +  # Add a straight line of best fit
  labs(x = 'Current video ideology', # Provide clear labels to help a stranger understand!
       y = 'Recommendation ideology',
       title = 'Relationship between current video and recommendation ideology')
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



This relationship is even stronger, suggesting a very strong positive relationship between the ideology of the video the user is currently watching and the videos they are being recommended by YouTube's algorithm.

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

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

tidy(model_current)
```

```
## # A tibble: 2 × 5
##   term          estimate std.error statistic   p.value
##   <chr>            <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    0.0580   0.00325       17.9 2.89e- 57
## 2 ideo_current   0.743    0.0121        61.2 9.94e-257
```

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

According to this model, the average ideology of a user's recommendations is 0.058 when the current video ideology is zero, meaning the recommendations that are shown on a totally moderate video are a tiny bit conservative. The model also tells us that the average ideology of recommendations shown to users increases by 0.743 units when the ideology of the current video increases by 1 unit. I am more than 99.99% confident that this positive relationship is not zero, meaning it is a strong relationship.

EC #1 [2 points]

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

While both watch history and current video ideology are strong predictors of the ideology of recommendations, the relationship between the current video and recommendations is stronger than the relationship between watch history and recommendations. The beta coefficient is 0.743, compared to 0.558, and the p-value is effectively zero, compared to 3.31e-57. While model variables are very strong predictors, it appears that the current video ideology is a stronger predictor of recommendation ideology.

Question 7 [2 points + 1 EC point]

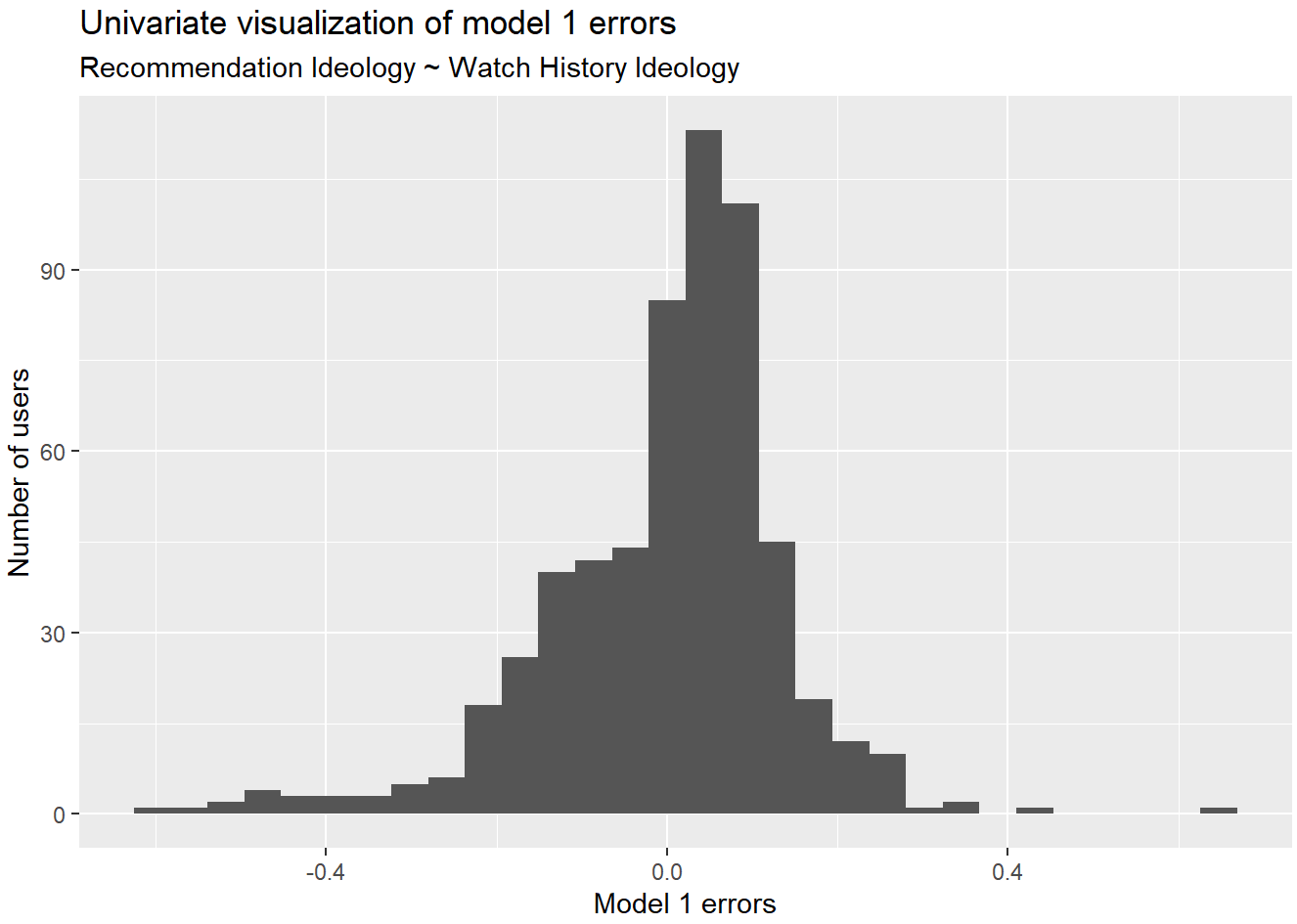
Now let's evaluate the models. Start by calculating the "mistakes" (i.e., the "errors" or the "residuals") generated by both models and saving these as new columns (`errors_watch` *and* `errors_current` *) in the* `yt` *dataset.*

```
# Calculating errors
yt <- yt %>%
  mutate(preds_watch = predict(model_watch), # Get the predicted values from the first m
odel (Yhat)
         preds_current = predict(model_current)) %>% # Get the predicted values from the
second model (Yhat)
  mutate(errors_watch = ideo_recommendation - preds_watch, # Calculate errors for the fi
rst model (Y - Yhat)
         errors_current = ideo_recommendation - preds_current) # Calculate errors for th
e second model (Y - Yhat)
```

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()` *.*

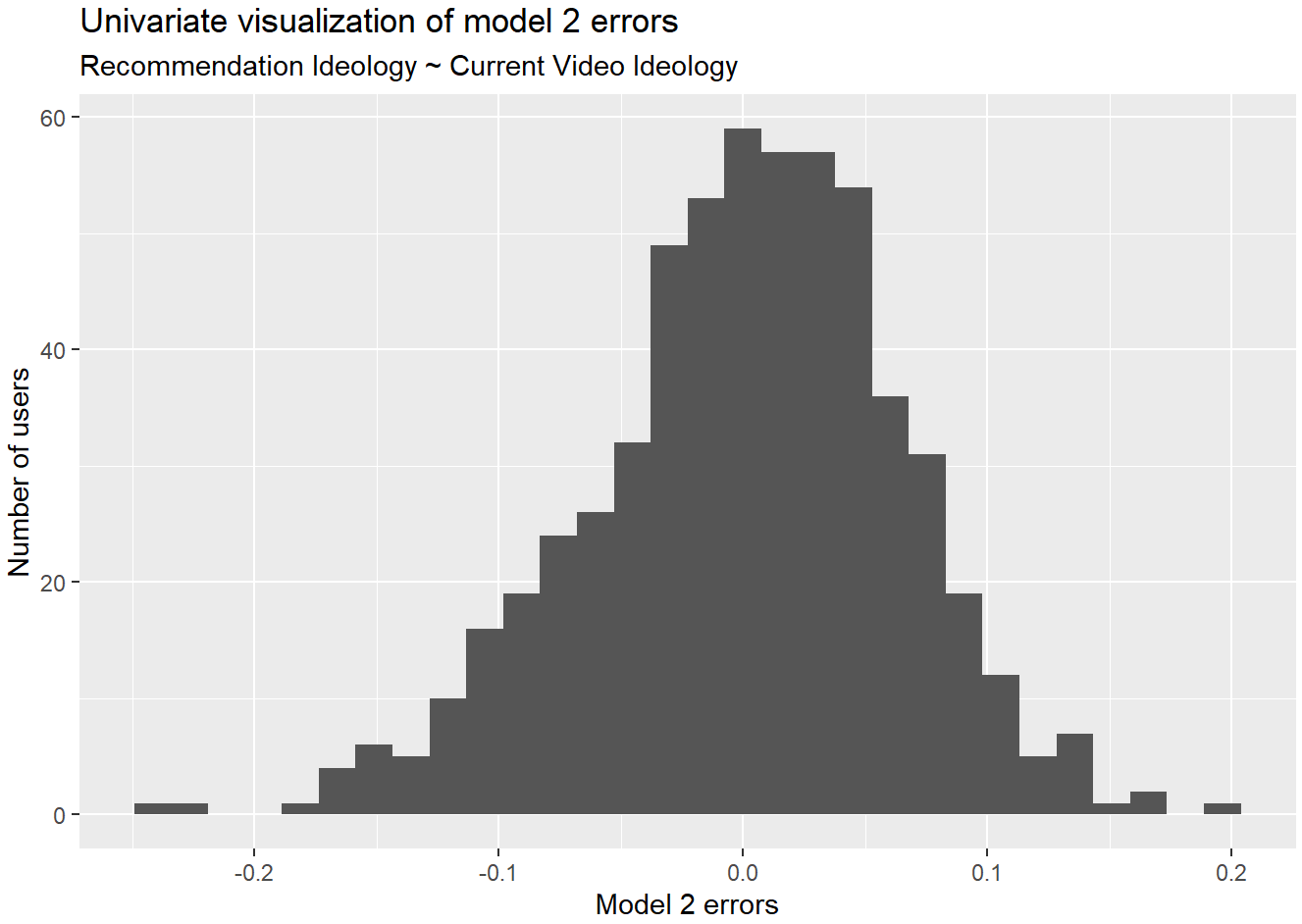
```
# Univariate visualization of watch history model errors
yt %>%
  ggplot(aes(x = errors_watch)) + # Put the errors from the first model on the x-axis
  geom_histogram() +  # Choose the best geom_...() to visualize based on the variable's
type
  labs(x = 'Model 1 errors', # Provide clear labels to help a stranger understand!
       y = 'Number of users',
       title = 'Univariate visualization of model 1 errors',
       subtitle = 'Recommendation Ideology ~ Watch History Ideology')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Univariate visualization of current video model errors
yt %>%
  ggplot(aes(x = errors_current)) + # Put the errors from the second model on the x-axis
  geom_histogram() +  # Choose the best geom_...() to visualize based on the variable's
type
  labs(x = 'Model 2 errors', # Provide clear labels to help a stranger understand!
       y = 'Number of users',
       title = 'Univariate visualization of model 2 errors',
       subtitle = 'Recommendation Ideology ~ Current Video Ideology')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



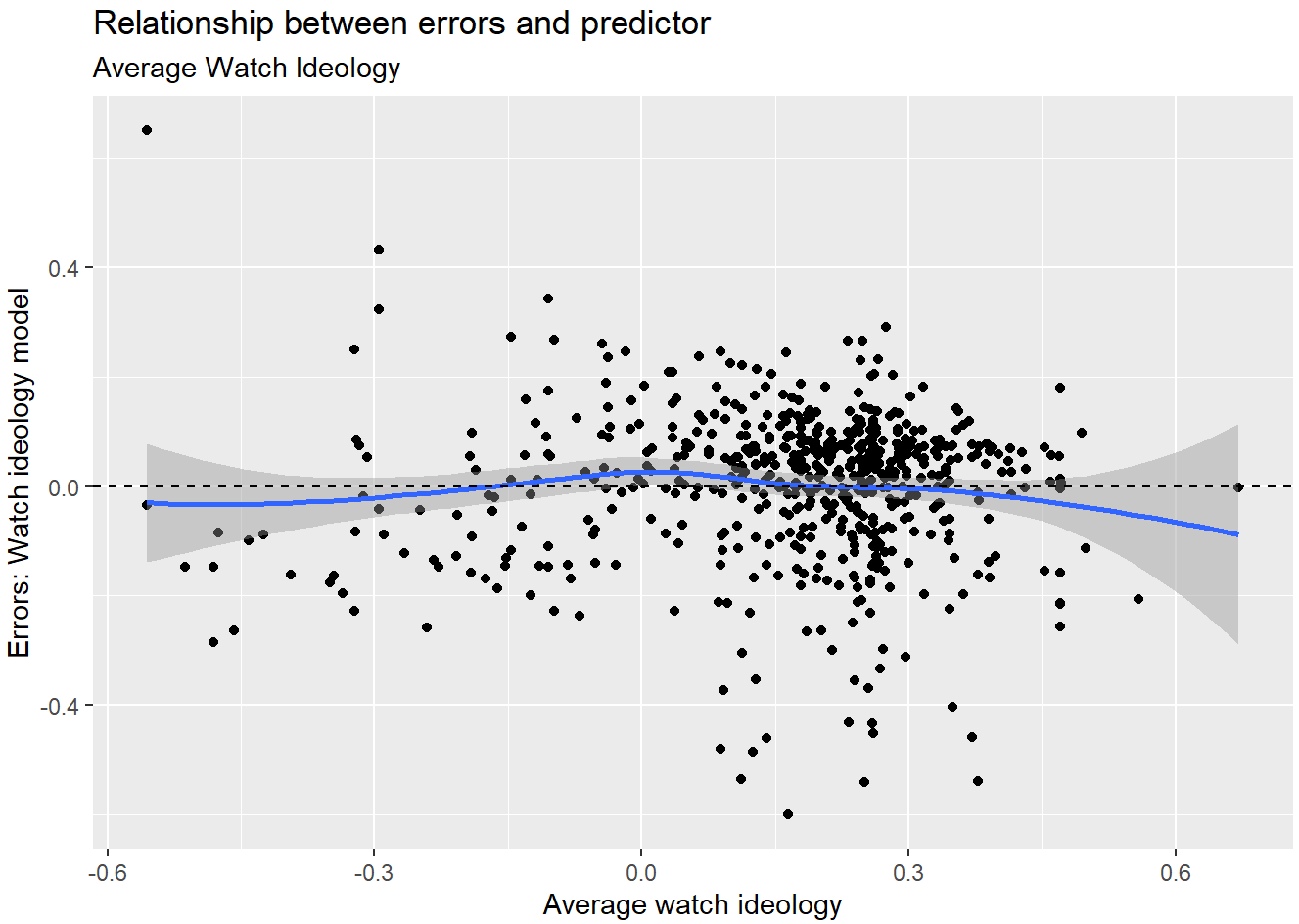

```
# EC [1 point]: Plot both errors on a single plot. Hint: use pivot_longer().
yt %>%
  select(ResponseId,matches('errors')) %>%
  pivot_longer(names_to = 'model',
               values_to = 'errors',
               cols = -ResponseId) %>%
  ggplot(aes(x = errors,
             fill = model)) +
  geom_density(alpha = .3) +
  labs(x = 'Model errors',
       y = 'Density',
       fill = 'Model',
       title = 'Comparing Model 1 and 2 errors')
```

Both models appear to do fairly well. However, the current video model's errors are more tightly clustered around zero, and are also more symmetrically distributed. As such, I would argue that model 2 is the better fit.

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 [2 points]: Create two plots side-by-side using facet_wrap(). This is SUPER HARD, so don't worry if you can't get it.

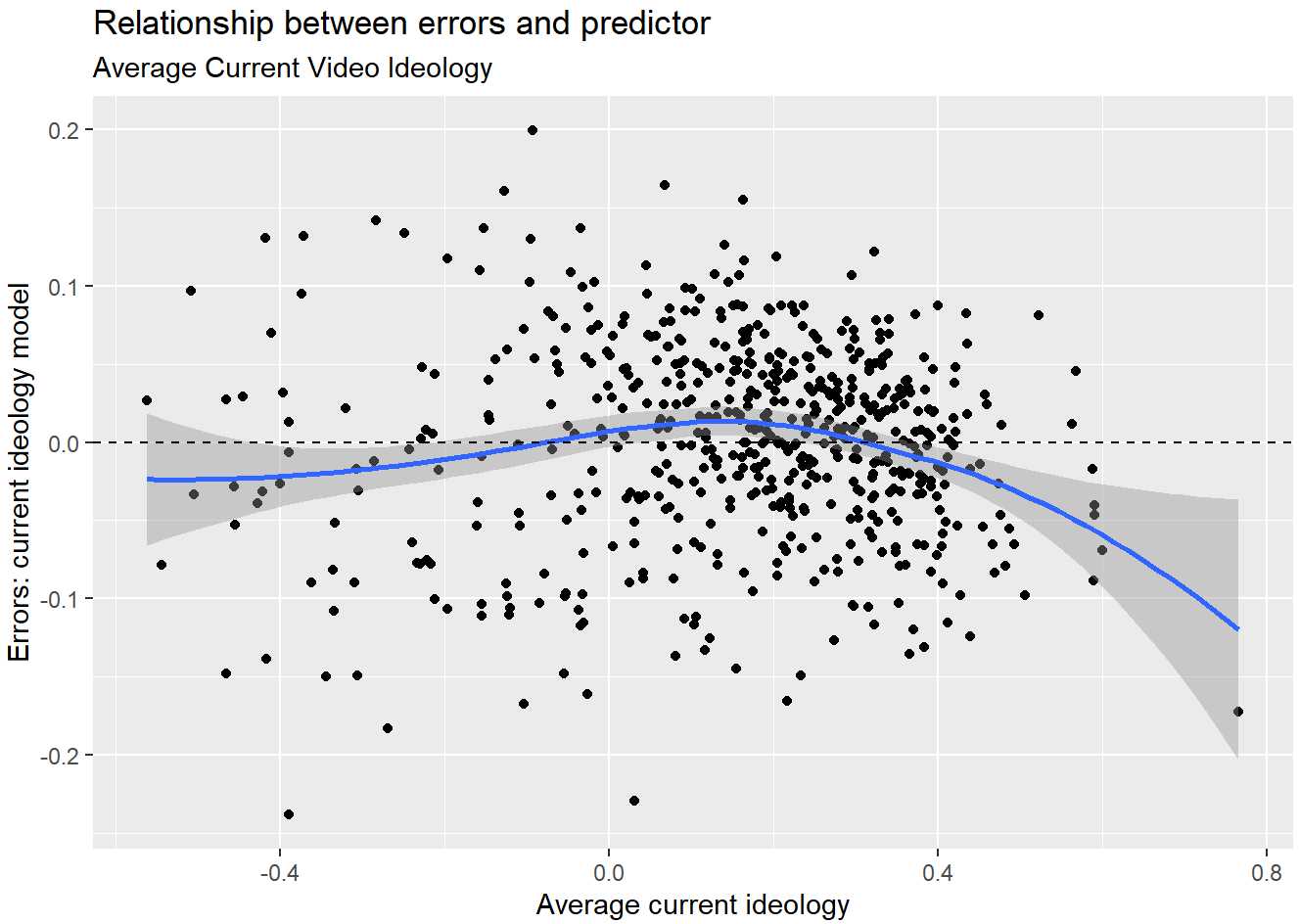
```
# Multivariate visualization of watch history errors
yt %>%
  ggplot(aes(x = ideo_watch,       # Put the predictor on the x-axis
             y = errors_watch)) + # Put the errors on the y-axis
  geom_point() + # Choose the best geom_...()
  geom_smooth() + # Add a curvey line of best fit
  geom_hline(yintercept = 0,linetype = 'dashed') + # Add a horizontal dashed line at zer
o
  labs(x = 'Average watch ideology', # Give it clear labels
       y = 'Errors: Watch ideology model',
       title = 'Relationship between errors and predictor',
       subtitle = 'Average Watch Ideology')
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



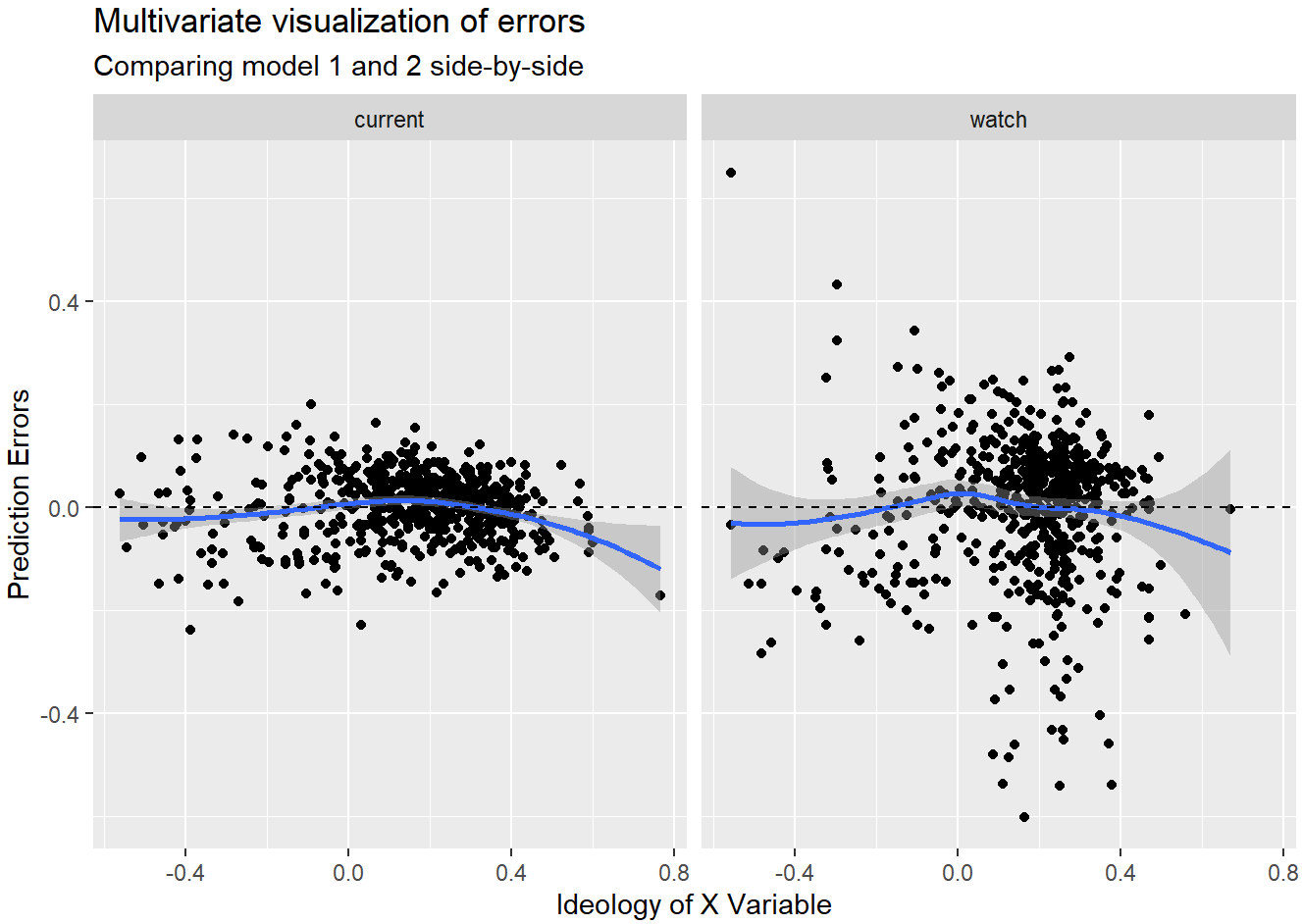

```
# Multivariate visualization of current video errors
yt %>%
  ggplot(aes(x = ideo_current,       # Put the predictor on the x-axis
             y = errors_current)) + # Put the errors on the y-axis
  geom_point() + # Choose the best geom_...()
  geom_smooth() + # Add a curvey line of best fit
  geom_hline(yintercept = 0,linetype = 'dashed') + # Add a horizontal dashed line at zer
o
  labs(x = 'Average current ideology', # Give it clear labels
       y = 'Errors: current ideology model',
       title = 'Relationship between errors and predictor',
       subtitle = 'Average Current Video Ideology')
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
# EC [2 points]: Try to create two plots side-by-side. (SUPER HARD)
yt %>%
  select(ideo_watch,
         ideo_current,
         errors_watch,
         errors_current,
         ResponseId) %>%
  pivot_longer(cols = -ResponseId,
               names_to = c('variable','model'),
               names_pattern = '(.*)_(.*)') %>%
  pivot_wider(names_from = variable,
              values_from = value) %>%
  ggplot(aes(x = ideo,
             y = errors)) +
  geom_point() +
  geom_smooth() +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  facet_wrap(~model) +
  labs(x = 'Ideology of X Variable',
       y = 'Prediction Errors',
       title = 'Multivariate visualization of errors',
       subtitle = 'Comparing model 1 and 2 side-by-side')
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



The multivariate visualization of the errors indicates that the current video model is a better fit than the watch history model since the points are more tightly clustered around zero and the line is flatter.

Question 8 [2 points]

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! EC [1 point]: plot the errors by the model using geom_boxplot(). HINT: you'll need to use pivot_longer() to get the data shaped correctly.*

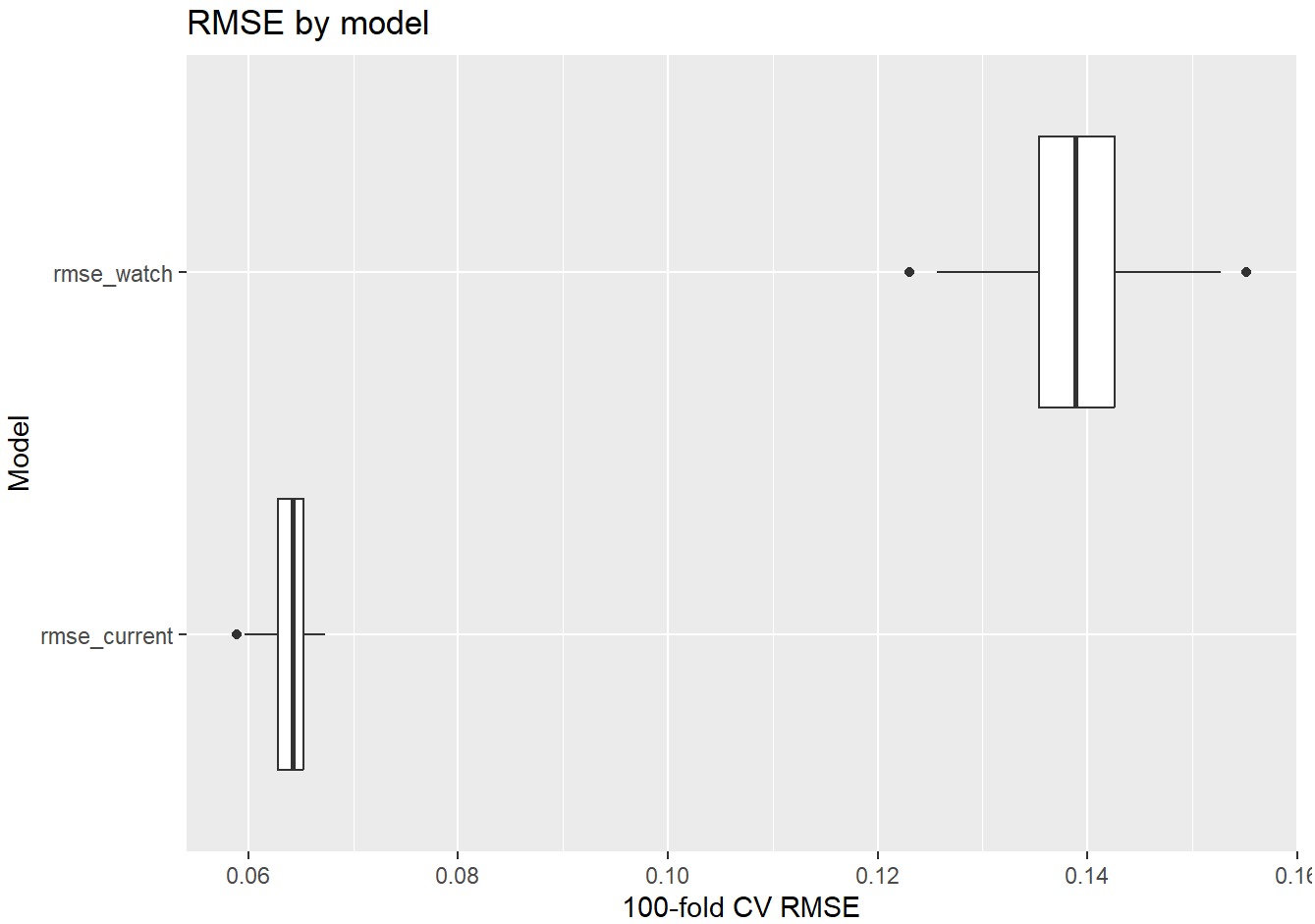
```
set.seed(123) # Set the seed to ensure replicability
cvRes <- NULL # Instantiate an empty object to save the results
for(i in 1:100) { # 100-fold cross validation
  # Create the training dataset
  train <- yt %>%
    sample_n(size = round(nrow(.)*.5), # set the size equal to half of the original data
set
             replace = F) # Make sure to NOT replace observations (unlike bootstrappin
g!)

  # Create the testing dataset
  test <- yt %>%
    anti_join(train) # Use anti_join() to make the test set contain every observation NO
T in the train set

  # Estimate model 1 on the training dataset
  mTmp_watch <- lm(ideo_recommendation ~ ideo_watch,
             data = train)

  # Estimate model 2 on the training dataset
  mTmp_current <- lm(ideo_recommendation ~ ideo_current,
             data = train)

  # Predict both models on the testing dataset
  test <- test %>%
    mutate(preds_watch = predict(mTmp_watch,newdata = test),
           preds_current = predict(mTmp_current,newdata = test))

  # Calculate the RMSE
  answer <- test %>%
    mutate(errors_watch = ideo_recommendation - preds_watch, # calculate the errors for
model 1
           errors_current = ideo_recommendation - preds_current) %>% # calculate the err
ors for model 2
    mutate(se_watch = errors_watch^2, # Square the errors
           se_current = errors_current^2) %>%
    summarise(mse_watch = mean(se_watch,na.rm=T), # Take the mean of the square errors
              mse_current = mean(se_current,na.rm=T)) %>%
    mutate(rmse_watch = sqrt(mse_watch), # Take the square root of the mean of the squar
e errors
           rmse_current = sqrt(mse_current),) %>%
    mutate(cvInd = i) # Add the cross validation index

  # Save the result
  cvRes <- cvRes %>%
    bind_rows(answer)
}

# Finally, calculate the RMSE value
mean(cvRes$rmse_watch)
```

```
## [1] 0.1392125
```

```
mean(cvRes$rmse_current)
```

```
## [1] 0.06399898
```

```
# EC [1 point]:
cvRes %>%
  select(rmse_watch,rmse_current,cvInd) %>%
  pivot_longer(cols = c(rmse_watch,rmse_current),
               names_to = 'model',
               values_to = 'rmse') %>%
  ggplot(aes(x = rmse,
             y = model)) +
  geom_boxplot() +
  labs(x = '100-fold CV RMSE',
       y = 'Model',
       title = 'RMSE by model')
```



The comparison of RMSE clearly illustrates the superior fit of the current YouTube video compared to the watch history. The watch history model has an RMSE of roughly 0.14, while the current video model has an RMSE of roughly 0.064. Both of these are relatively small RMSE values when compared to the outcome variable, which ranges from -0.5 to +0.5. But the current video model is clearly superior.

EC #2 [2 points]

Let's try including both X variables into a single model. Run the regression and evaluate the errors as described just as you did before. Then evaluate the RMSE for ALL 3 MODELS using 100-fold cross validation with an 80-20 split. Does this combined model perform better than the two separate models? Worse? Why?

```
set.seed(123) # Set the seed to ensure replicability
cvRes <- NULL # Instantiate an empty object to save the results
for(i in 1:100) { # 100-fold cross validation
  # Create the training dataset
  train <- yt %>%
    sample_n(size = round(nrow(.)*.8),
             replace = F)

  # Create the testing dataset
  test <- yt %>%
    anti_join(train)

  # Estimate the regression on the training dataset
  mTmp_watch <- lm(ideo_recommendation ~ ideo_watch,
             data = train)
  mTmp_current <- lm(ideo_recommendation ~ ideo_current,
             data = train)
  mTmp_comb <- lm(ideo_recommendation ~ ideo_watch + ideo_current,
             data = train)


  # Predict the model on the testing dataset
  test <- test %>%
    mutate(preds_watch = predict(mTmp_watch,newdata = test),
           preds_current = predict(mTmp_current,newdata = test),
           preds_comb = predict(mTmp_comb,newdata = test))

  # Calculate the RMSE
  answer <- test %>%
    mutate(errors_watch = ideo_recommendation - preds_watch,
           errors_current = ideo_recommendation - preds_current,
           errors_comb = ideo_recommendation - preds_comb) %>%
    mutate(se_watch = errors_watch^2,
           se_current = errors_current^2,
           se_comb = errors_comb^2) %>%
    summarise(mse_watch = mean(se_watch,na.rm=T),
              mse_current = mean(se_current,na.rm=T),
              mse_comb = mean(se_comb,na.rm=T)) %>%
    mutate(rmse_watch = sqrt(mse_watch),
           rmse_current = sqrt(mse_current),
           rmse_comb = sqrt(mse_comb)) %>%
    mutate(cvInd = i)

  # Save the result
  cvRes <- cvRes %>%
    bind_rows(answer)
}

# Finally, calculate the RMSE value
mean(cvRes$rmse_watch)
```

```
## [1] 0.1391034
```

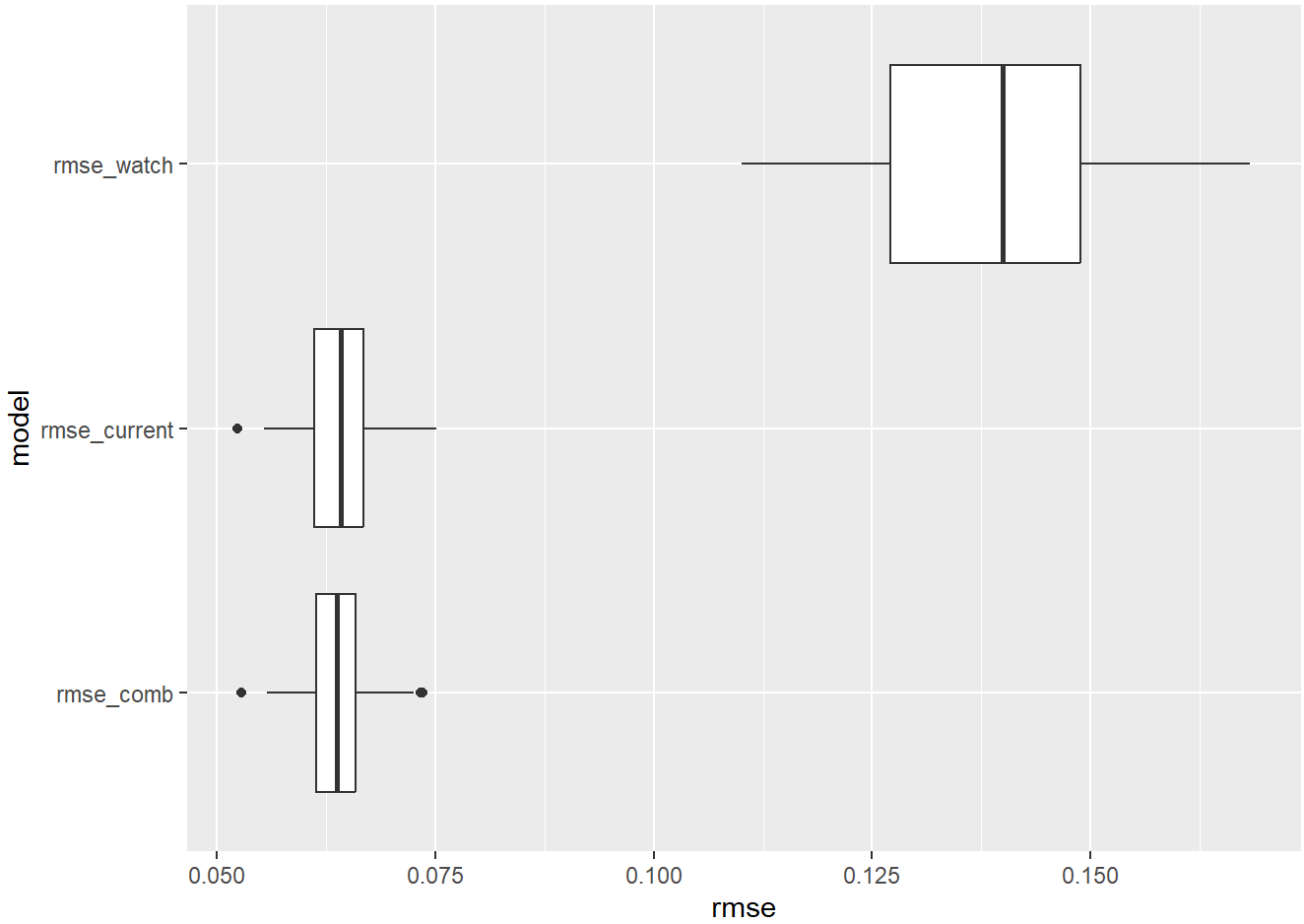
```
mean(cvRes$rmse_current)
```

```
## [1] 0.06430862
```

```
mean(cvRes$rmse_comb)
```

```
## [1] 0.06354252
```

```
cvRes %>%
  select(rmse_watch,rmse_current,rmse_comb,cvInd) %>%
  pivot_longer(cols = c(rmse_watch,rmse_current,rmse_comb),
               names_to = 'model',
               values_to = 'rmse') %>%
  ggplot(aes(x = rmse,
             y = model)) +
  geom_boxplot()
```



There isn’t much of an improvement when combining both X variables into a single regression. The average cross-validated RMSE declines from 0.064 to 0.063.