HW5

Samuel Pekofsky

Introduction

This dataset contains simulated data on Student Habits and other info, with 13 features that can serve as predictors of `exam_score`.

This is an excellent opportunity for PCA, as looking the top principal components may illuminate which habits and identifying information most strongly contribute to better exam scores.

While this is simulated data, it is quite similar to lots of real data, but cleaner and more organized, so for the sake of limiting the loading and pre-processing, I opted to use this dataset. This being said, the same ideas could easily apply to real student data.

As for my expectations, I expect `study_hours_per_day`, `attendance_percentage`, `sleep_hours`, `diet_quality`, `exercise_frequency`, and `mental_health_rating` to be the strongest predictors of exam scores. In other words, I would expect these to have the largest absolute value coefficients in the first several principal components. I expect anything implying time being spent outside of studying to not be highly impactful, as I believe that study time and sleep time will be the most impactful, and the rest can be chalked up to differences in time management more than differences in time spent on other things.

Setup

Load in the Data

```
data <- read_csv("https://github.com/pekofsky/schoolwork/raw/refs/heads/main/student_habits_performance
```

Pre-Processing to Maximize Feature Usage

Because of gender's categorical nature, I believe that it is better to remove it from this dataset, since it still has plenty of good features. Additionally, making a numerical scaling for gender logically wouldn't make sense.

With this being said, I believe that that the remaining categorical variables can be reasonably converted to numerical scaling variables, which is what I have done with them. My reasoning for this is that these categories can all be seen as a series of rankings, making numerical scaling more intuitive and logical than it would've been for gender.

```
# Remove 'gender'
data <- data %>% select(-gender)

# Recode 'part_time_job'
data$part_time_job <- ifelse(data$part_time_job == "No", 0,
                             ifelse(data$part_time_job == "Yes", 1, NA))

# Recode 'diet_quality'
data$diet_quality <- recode(data$diet_quality,
```

```
                              "Poor" = 0,
                              "Fair" = 1,
                              "Good" = 2)

# Recode 'parental_education_level'
data$parental_education_level <- recode(data$parental_education_level,
                                        "None" = 0,
                                        "High School" = 1,
                                        "Bachelor" = 2,
                                        "Master" = 3)

# Recode 'internet_quality'
data$internet_quality <- recode(data$internet_quality,
                                "Poor" = 0,
                                "Average" = 1,
                                "Good" = 2)

# Recode 'extracurricular_participation'
data$extracurricular_participation <- ifelse(data$extracurricular_participation == "No", 0, ifelse(data$
```

PCA and Visualizations

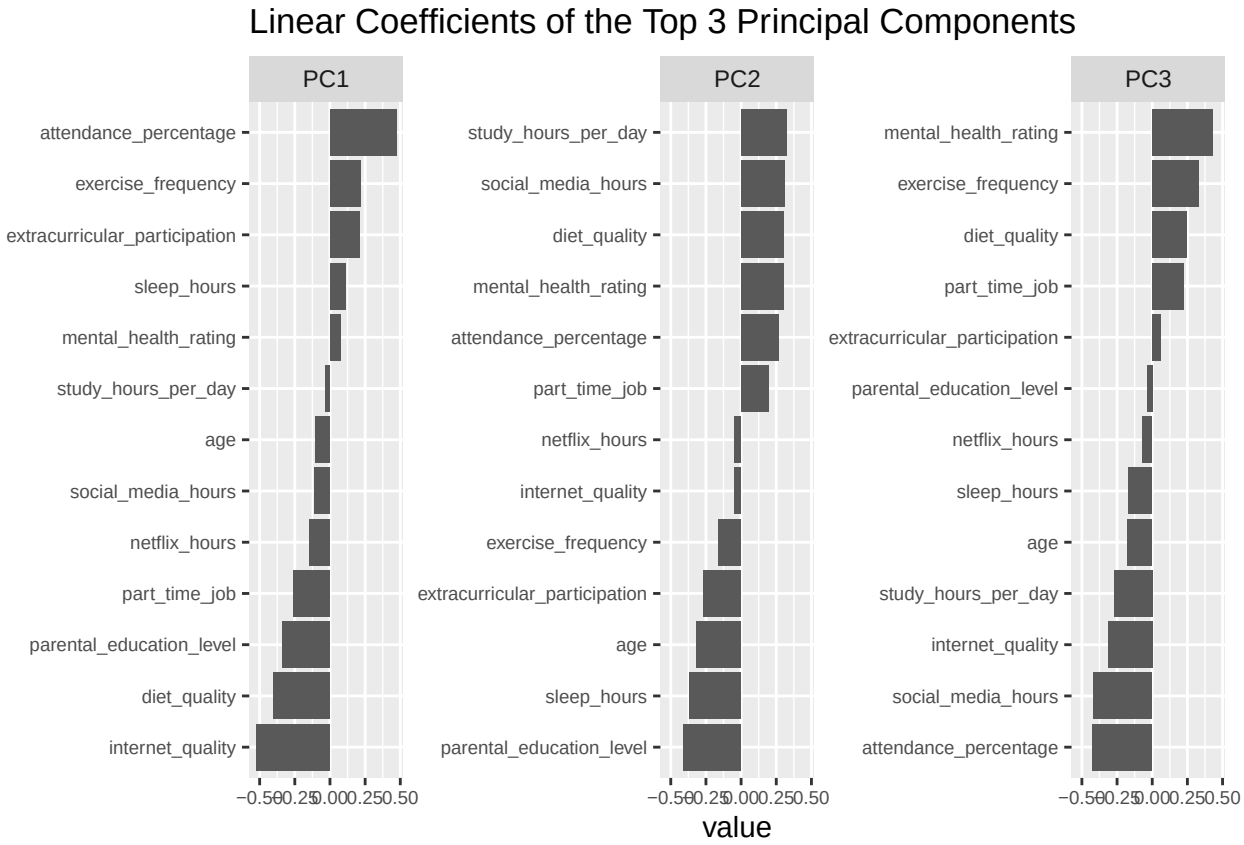
Creating pca_prep

```
# Make recipe
pca_rec <- recipe(~., data = data) %>%
  update_role(student_id, exam_score, new_role = "id") %>%
  step_normalize(all_predictors()) %>%
  step_pca(all_predictors())

# Prepare
pca_prep <- prep(pca_rec)
```

PC1 and PC2

In order to properly evaluate my predictions, it makes sense to view the first 3 principal components.

```
# Get top 3 principal components
components <- tidy(pca_prep, 2) %>%
  filter(component %in% str_c("PC", 1:3)) %>%
  mutate(terms = reorder_within(terms, value, component))

# Plot
ggplot(components, aes(value, terms)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~component, scales = "free_y") +
  scale_y_reordered() +
  labs(title = "Linear Coefficients of the Top 3 Principal Components",
       y = NULL) +
  theme(axis.text = element_text(size = 7))
```



As I expected, `attendance_percentage` is quite impactful, and `netflix_hours` is not, however, I greatly underestimated the effect of `internet_quality`. `study_hours_per_day` seems to be quite significant, but I expected this to be the most impactful by far, which it does not seem to be.

Exam Score Heatmap with PC1 and PC2

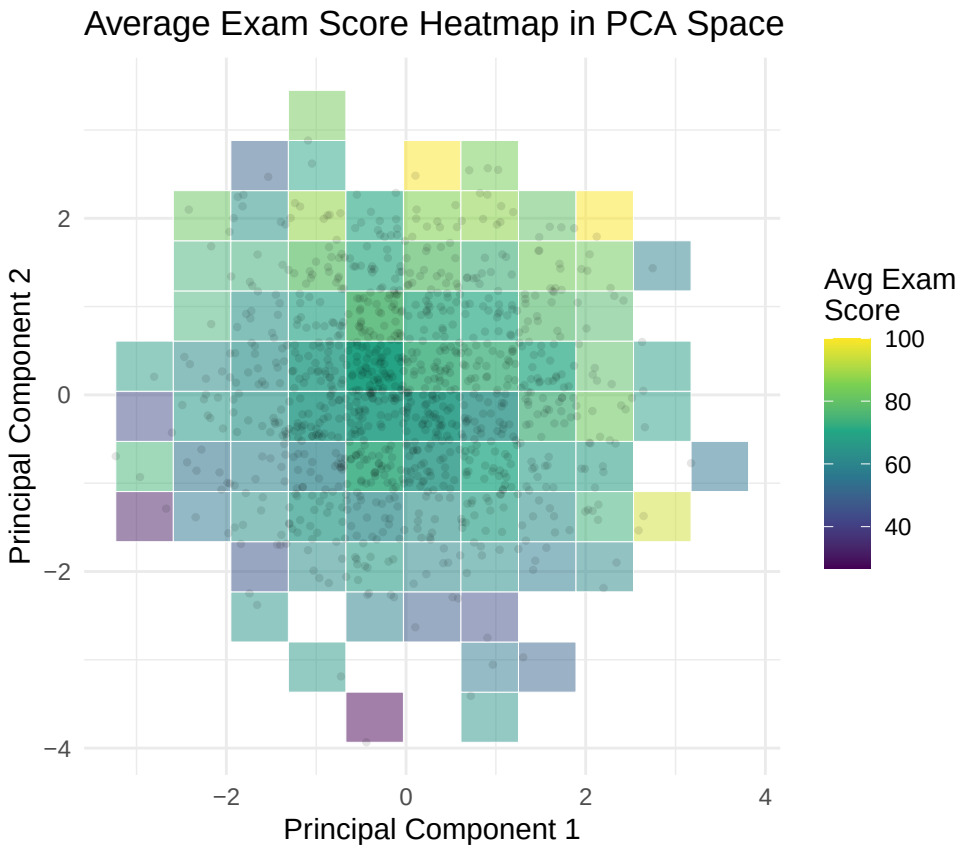
In order to see how well principal components 1 and 2 relate to exam_scores, I believe that a heatmap that averages subsets of exam scores would be most insightful, as it would be much easier to read than a ton of individual points.

Originally, I used only large translucent points, but since there was a large variance in exam scores in most areas of the graph, I realized a heatmap that relied on the average exam scores in the areas of each tile would be much easier to gain insights from.

This heatmap still includes points under it however, so the viewer can get an idea of the distribution of data, since some tiles only have a couple points, while others have several dozen.

```
# Get PCA scores, join with original data to also have exam scores
scores <- bake(pca_prep, data) %>%
  inner_join(data) %>%
  select(PC1, PC2, exam_score)

# Create a heatmap showing average exam scores in the PCA space
# Largely generated by Claude

# Calculate the ranges for PC1 and PC2
pc1_range <- range(scores$PC1)
pc2_range <- range(scores$PC2)
```

```r
# Get Variances
variances <- tidy(pca_prep, 2, type = "variance")

# Function that automatically determines appropriate bin sizes
# to generate approximately the desired number of tiles
create_heatmap <- function(data, x_bins = 6, y_bins = 7) {
  # Calculate bin widths based on data range and desired number of bins
  x_width <- (max(data$PC1) - min(data$PC1)) / x_bins
  y_width <- (max(data$PC2) - min(data$PC2)) / y_bins

  # Create binned data with average exam scores per bin
  binned_data <- data %>%
    mutate(
      PC1_bin = floor((PC1 - min(PC1)) / x_width) * x_width + min(PC1) + x_width/2,
      PC2_bin = floor((PC2 - min(PC2)) / y_width) * y_width + min(PC2) + y_width/2
    ) %>%
    group_by(PC1_bin, PC2_bin) %>%
    summarize(
      avg_score = mean(exam_score, na.rm = TRUE),
      count = n(),
      .groups = "drop"
    )

  # Create the heatmap
  ggplot(binned_data, aes(x = PC1_bin, y = PC2_bin)) +
    geom_tile(aes(fill = avg_score, alpha = count), color = "white", linewidth = 0.1) +
    scale_fill_viridis_c(option = "viridis", name = "Avg Exam\nScore") +
    scale_alpha_continuous(range = c(0.5, 1), guide = "none") +
    coord_fixed(sqrt(variances$value[2] / variances$value[1])) +
    labs(
      title = "Average Exam Score Heatmap in PCA Space",
      x = "Principal Component 1",
      y = "Principal Component 2"
    ) +
    theme_minimal() +
    # Add original points with reduced size and alpha for context
    geom_point(data = scores, aes(x = PC1, y = PC2), size = 0.8, alpha = 0.1, color = "black")
}

# Use function to show heatmap
create_heatmap(scores, x_bins = 10, y_bins = 12)
```



From viewing this visualization, it is clear that there is a positive trend between average exam scores and the values for principal components 1 and 2, as one would expect. This being said, the relationship is far from perfect, as there are many pairs of tiles that are not in line with this trend.

Conclusion

Through the use of PCA on this simulated dataset, I have demonstrated some of the many ways that the relationships between high-dimensional personal data and performance metrics can be visually encoded. While my process would most easily translate to real student data, it could very well translate to similar areas, such as workplace efficiency and athletic performance.