Data Analysis for Research

Dr Polla Fattah

2025-04-20

Part 1: Foundations of Data Analysis

The Role of Data Analysis in Research Validation and Theory Testing

In scientific research, data analysis plays several critical roles:

1. **Testing Hypotheses**.
2. **Evaluating Models**.
3. **Generalizing Findings**.
4. **Validating Constructs.**
5. **Comparing Groups or Conditions.**
6. **Communicating Evidence.**

EDA vs. Inferential vs. Predictive Modeling

Understanding the types of data analysis is essential for selecting the correct approach based on your research question.

1. Exploratory Data Analysis (EDA): EDA is the **first stage of analysis**, where you get familiar with your dataset. EDA is **open-ended** and **non-confirmatory**. It helps you form hypotheses and decide which variables may be of interest for deeper analysis.
2. Inferential Statistics:Inferential analysis is about **making generalizations**. Common tools: t-tests, ANOVA, regression models, confidence intervals, p-values.
3. Predictive Modeling: Predictive modeling focuses on **forecasting outcomes**. The goal is not just to understand relationships, but to use them for making predictions about new or future data.

A Simulated Research Thesis Scenario

To ground our lecture in a realistic academic context, we will follow the case of **Sara**, a Master's student in Educational Psychology. Her thesis is focused on understanding how lifestyle factors and perceived academic support affect graduate student well-being and academic outcomes.

Sara's Research Motivation

Sara is tasked with investigating how lifestyle and psychological factors influence GPA, well-being, and dropout risk among graduate students.

Thesis Objective

Sara's thesis aims to:

- Explore links between lifestyle, academic pressure, and performance.
- Predict students at risk of dropout.
- Identify student clusters by behavior or performance.
- Investigate if psychological and lifestyle factors reveal deeper patterns like burnout.

Dataset in Use

First 15 Students with Simplified Column Names

Variable	Type
`student_id`	Categorical
`age`	Numeric
`gender`	Categorical
`study_hours_per_week`	Numeric
`sleep_hours_per_night`	Numeric
`stress_level`	Ordinal (1–10)
`gpa`	Numeric
`caffeine_intake_mg`	Numeric
`exercise_freq_per_week`	Numeric
`supervisor_support`	Ordinal (1–5)
`mental_health_score`	Numeric
`burnout_level`	Ordinal (1–10)
`satisfaction_academic`	Ordinal (1–5)
`considering_dropout`	Binary (0 = No, 1 = Yes)

ID	Age	Gender	StudyHrs	Sleep	Stress	GPA	Caffeine	Exercise	Support	MentalHealth	Burnout	Sa
S001	28	Female	39.9	5.3	5	2.75	224	0	4	63	3	2
S002	25	Male	32.2	7.1	4	3.48	203	3	5	55	7	3
S003	34	Male	14.4	5.9	10	3.57	130	4	4	65	8	2

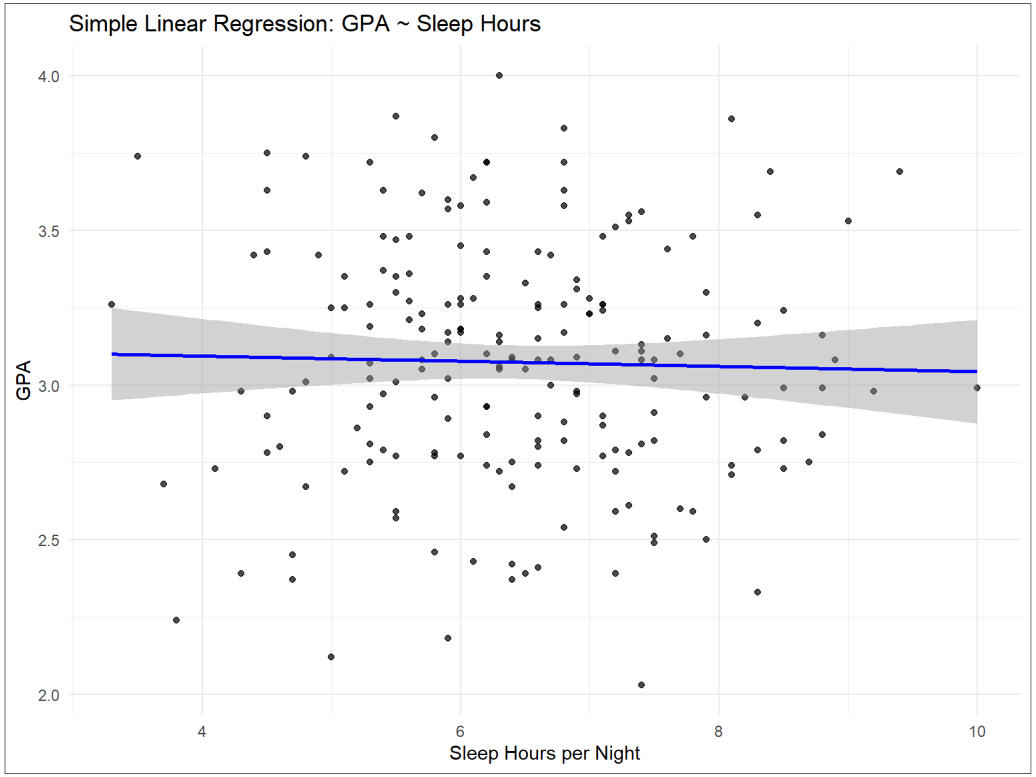
ID	Age	Gender	StudyHrs	Sleep	Stress	GPA	Caffeine	Exercise	Support	MentalHealth	Burnout	Sa
S004	32	Female	21.6	5.5	6	2.77	120	1	2	56	4	2
S005	29	Female	33.9	6.4	6	2.67	178	5	4	63	8	3
S006	34	Female	20.0	5.3	6	3.19	286	3	4	80	5	4
S007	26	Male	28.1	5.8	6	2.78	229	1	5	45	6	3
S008	28	Female	30.4	5.1	6	3.25	147	2	4	77	6	3
S009	31	Female	18.5	8.9	5	3.08	249	3	3	80	5	4
S010	24	Female	24.6	6.5	5	2.39	0	3	2	58	7	3
S011	28	Male	5.0	5.7	6	3.62	283	3	3	28	7	4
S012	32	Male	17.8	6.8	5	3.72	23	1	3	51	2	3
S013	32	Female	23.2	6.4	5	2.75	147	3	3	79	8	3
S014	29	Male	16.3	6.2	6	2.84	79	0	4	60	2	3
S015	26	Male	36.4	7.2	5	3.11	63	2	5	45	8	1

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```r
ggplot(data, aes(x = sleep_hours_per_night, y = gpa)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = TRUE, color = "blue") +
  labs(title = "Simple Linear Regression: GPA ~ Sleep Hours",
       x = "Sleep Hours per Night", y = "GPA") +
  theme_minimal()
```

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



Part 3: Regression and Predictive Modeling

Regression is one of the most fundamental tools in data analysis. It allows researchers to **quantify the relationship between a dependent variable (outcome)** and one or more **independent variables (predictors)**.
In research, regression is not just used for prediction—it is also widely used for **explaining patterns, testing hypotheses**, and **identifying significant influencing factors.**

3.1 What is Regression?

At its core, regression is a method of estimating the expected value of a response variable, Y, given one or more predictors, X. The **simplest form** of regression is **simple linear regression**, where:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

- β_0 is the intercept (value of Y when X = 0)
- β_1 is the slope (change in Y for one unit increase in X)
- ϵ is the random error term

This formula expands naturally to **multiple linear regression**, where we have more than one independent variable:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$$

Each coefficient β tells us the expected **change in Y per unit change in X**, **holding all other variables constant**.

3.2 Simple Demonstration of Regression in R

Before we dive into Sara's dataset, let's use a built-in dataset in R to build a basic regression model.
Example: Predicting car fuel efficiency (`mpg`) using weight (`wt`) from the `mtcars` dataset

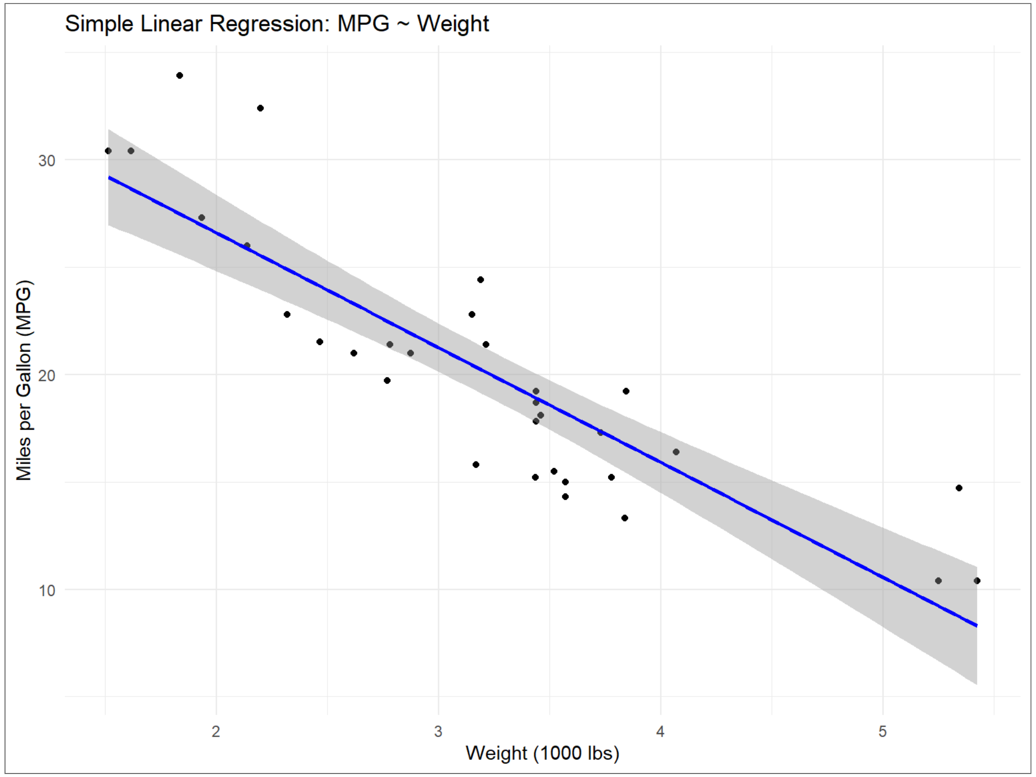
```r
# Load dataset
data(mtcars)

# Fit a simple linear regression model
model_simple <- lm(mpg ~ wt, data = mtcars)

# View model summary
summary(model_simple)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This tells us whether **car weight** has a statistically significant effect on **miles per gallon.** The slope of the line will tell us how much fuel efficiency decreases for each additional 1000 lbs.

Plotting the Relationship

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



This visualization helps students understand regression as a **line of best fit**.

3.3 Our Simulated Case: Modeling GPA from Lifestyle Factors

We now return to Sara's dataset. Her hypothesis is:

> *"Students with higher sleep hours and more supervisor support tend to have higher GPAs."*

Let's test this hypothesis using **multiple linear regression**, including `sleep_hours_per_night` and `supervisor_support` as predictors.
We will also include `gender` as a categorical predictor and later explore interaction effects.

Build the regression model

```
model_gpa <- lm(gpa ~ sleep_hours_per_night +
      supervisor_support + gender, data = data)
```

```
##
## Call:
## lm(formula = gpa ~ sleep_hours_per_night + supervisor_support +
##     gender, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10811 -0.22345 -0.01845  0.24533  0.92524
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.898920   0.179471  16.153   <2e-16 ***
## sleep_hours_per_night -0.003178   0.022310  -0.142   0.8869
## supervisor_support     0.066845   0.027268   2.451   0.0151 *
```

This output tells us: The **coefficient estimates** (e.g., how much GPA changes per hour of sleep)

3.4 Interpreting Regression Output

Key things to look for in the summary:

- **Coefficients (`Estimate`)**: The direction and size of the effect
- **`Pr(>|t|)` (p-value)**: Whether the effect is statistically significant
- **`R-squared`**: Proportion of variability in GPA explained by the model
- **Residual standard error**: The average prediction error

Part 4: Classification Analysis

In this section, we turn from continuous outcomes (like GPA) to **categorical classification problems.** Our outcome of interest is whether a student is **considering dropping out** — a binary variable.

4.1 Research Hypothesis

> *"High stress and low supervisor support increase the likelihood of considering dropout."*

We will investigate whether we can **predict which students are at risk of dropout** using their stress levels, supervisor support, and other lifestyle or academic indicators.

4.2 Understanding Classification

Classification is the process of predicting which category or group a data point belongs to. In our case, we want to predict:

- `considering_dropout` = 0 (No)
- `considering_dropout` = 1 (Yes)

We will use both: - A **baseline logistic regression** model to demonstrate odds and interpretation. - **Advanced models**: Support Vector Machine (SVM) and Random Forests.

4.3 Preparing the Data

Before fitting models, ensure categorical variables are in proper format.

```
# Convert gender to factor if not already
data$gender <- as.factor(data$gender)
data$considering_dropout <- as.factor(data$considering_dropout)

# View dropout distribution
table(data$considering_dropout)
```

```
##
##   0   1
## 173  27
```

4.4 Logistic Regression (Baseline)

```r
# Fit logistic regression model
model_logit <- glm(considering_dropout ~ stress_level + supervisor_support +
         sleep_hours_per_night,
                   data = data, family = "binomial")

# Model summary
summary(model_logit)
```

```
##
## Call:
## glm(formula = considering_dropout ~ stress_level + supervisor_support +
##     sleep_hours_per_night, family = "binomial", data = data)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)             0.9238     1.4436   0.640   0.5222
## stress_level           -0.1060     0.1137  -0.933   0.3509
## supervisor_support     -0.3729     0.2107  -1.769   0.0769 .
## sleep_hours_per_night  -0.1406     0.1741  -0.807   0.4194
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

Interpreting Odds Ratios

```
# Calculate odds ratios
exp(coef(model_logit))
```

```
##          (Intercept)          stress_level    supervisor_support
##            2.5187485             0.8994023             0.6887683
## sleep_hours_per_night
##            0.8688320
```

This helps us understand how each variable affects the odds of dropout: - Values >1 increase odds of dropout. - Values <1 decrease odds of dropout.

4.5 Classification with Support Vector Machines (SVM)

Support Vector Machines are powerful models for classification, especially when classes are not linearly separable.

```r
# Load library
library(e1071)

# Train SVM model
model_svm <- svm(considering_dropout ~ stress_level + supervisor_support +
         sleep_hours_per_night,
                 data = data, kernel = "radial", probability = TRUE)

# Predict on training data
svm_preds <- predict(model_svm, data, probability = TRUE)

# Confusion matrix
table(Predicted = svm_preds, Actual = data$considering_dropout)
```

```
##          Actual
## Predicted   0   1
##         0 173  27
##         1   0   0
```

4.6 Classification with Random Forest

Random Forest is a tree-based ensemble method that works well with mixed data types and nonlinear relationships.

```
# Load library
library(randomForest)
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

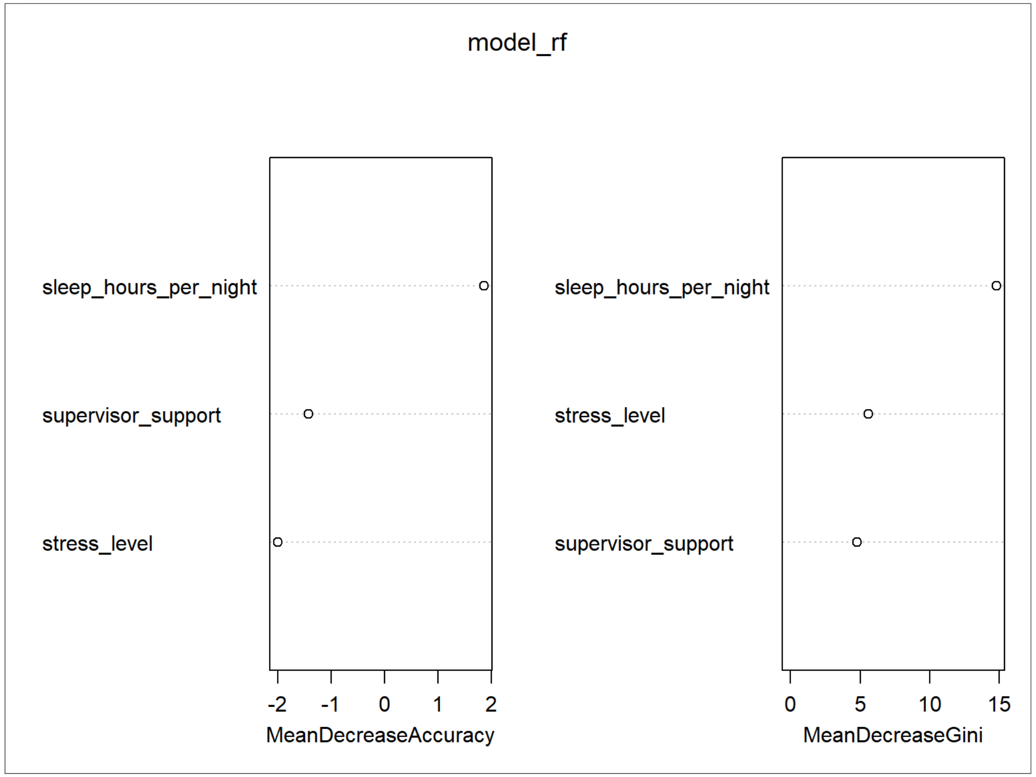
```
# Train random forest model
model_rf <- randomForest(considering_dropout ~ stress_level + supervisor_support +
         sleep_hours_per_night,
                        data = data, ntree = 100, importance = TRUE)
```

```r
# Predict and evaluate
rf_preds <- predict(model_rf, data)

# Confusion matrix
table(Predicted = rf_preds, Actual = data$considering_dropout)
```

```
##          Actual
## Predicted   0   1
##         0 173  23
##         1   0   4
```

Variable Importance

```r
# Plot variable importance
varImpPlot(model_rf)
```



This shows which variables were most influential in classifying students.

4.7 Model Performance and Accuracy Metrics

After training a classification model (SVM, Random Forest, Logistic Regression, etc.), we need to assess how well the model performs. This involves measuring the quality of its predictions compared to the true labels.

Core Concept: The Confusion Matrix

A **confusion matrix** is a table that shows the performance of a classification model:

	Predicted No	**Predicted Yes**
Actual No	TN (True Neg)	FP (False Pos)
Actual Yes	FN (False Neg)	TP (True Pos)

From this, we derive the following key metrics:

Performance Measures

1. Accuracy: The percentage of correct predictions out of all cases.

   $$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

   However, **accuracy can be misleading in imbalanced datasets.** For example, if 90% of students are not considering dropout, a model that always predicts “No” will have 90% accuracy but is useless.
2. Precision and Recall
   Precision: How many predicted positives were correct?

   $$\text{Precision} = \frac{TP}{TP + FP}$$

   High precision means **low false positives.**
   Recall (Sensitivity): How many actual positives were detected?

   $$\text{Recall} = \frac{TP}{TP + FN}$$

   High recall means **low false negatives**, which is crucial if your goal is to **catch at-risk students**.
3. **F1 Score**: The harmonic mean of precision and recall. It balances both in one metric.

   $$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Code: Evaluating Model Performance in R

Let's apply these metrics to our **Random Forest model**:

```r
# Load packages
library(caret)
library(pROC)

# Create confusion matrix
conf_matrix <- confusionMatrix(rf_preds, data$considering_dropout, positive = "1")
conf_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 173  23
##          1   0   4
##
##                Accuracy : 0.885
##                  95% CI : (0.8325, 0.9257)
##     No Information Rate : 0.865
##     P-Value [Acc > NIR] : 0.2382
##
##                   Kappa : 0.2313
##
```

Extracting precision, recall, F1

```r
# Confusion matrix details
conf_matrix$byClass[c("Precision", "Recall", "F1")]
```

```
## Precision    Recall        F1
## 1.0000000 0.1481481 0.2580645
```

ROC Curve

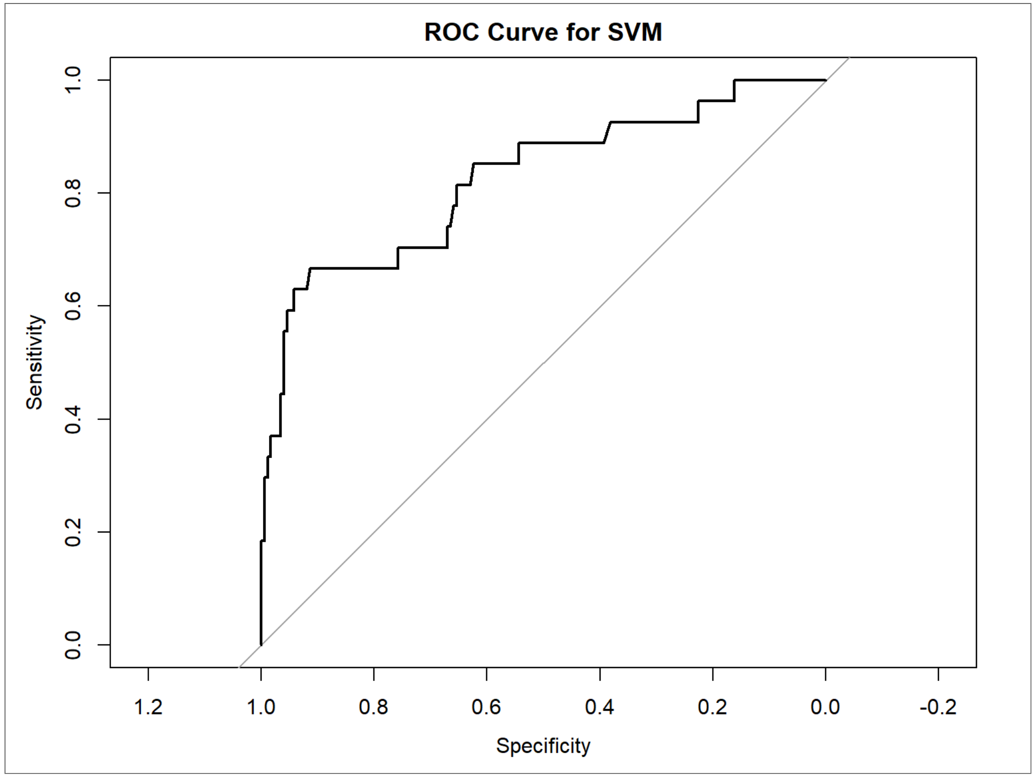
** ROC Curve and AUC (Area Under the Curve)**: The** ROC curve** plots the true positive rate vs. false positive rate for various thresholds.

- AUC = 1: perfect model
- AUC = 0.5: random guess
- AUC > 0.8: good model

ROC Curve and AUC for SVM

To compute AUC, we need predicted probabilities, not just labels:

```
# Predicted probabilities from SVM
svm_probs <- attr(predict(model_svm, data, probability = TRUE), "probabilities")[, "1"]

# Compute ROC and plot
roc_curve <- roc(response = data$considering_dropout, predictor = svm_probs)
plot(roc_curve, main = "ROC Curve for SVM")
```



```r
auc(roc_curve)
```

```
## Area under the curve: 0.8246
```

Part 5: Clustering and Pattern Discovery

Clustering is an unsupervised machine learning technique used to discover **groups or patterns** in data without predefined labels. It is especially useful in exploratory research when you want to identify subpopulations or behavioral profiles.

5.1 Research Objective

> *"Are there distinct lifestyle-academic profiles among students?"*

We aim to group students based on behavioral and academic factors such as: - Study habits - Sleep duration - Physical activity - Caffeine use - Stress and GPA

These groups can reveal insights like "low GPA, high stress" vs. "balanced lifestyle, high GPA" profiles.

5.2 Preparing the Data

Clustering algorithms, especially **K-means**, are sensitive to variable scales. We must: - Select numeric variables only - Remove missing values - Scale (normalize) the data

```
library(dplyr)

# Select relevant continuous variables
cluster_vars <- data %>%
  select(study_hours_per_week, sleep_hours_per_night,
         exercise_freq_per_week, caffeine_intake_mg,
         stress_level, gpa) %>%
  na.omit()

# Scale the data
cluster_scaled <- scale(cluster_vars)
```

5.3 Determining the Optimal Number of Clusters

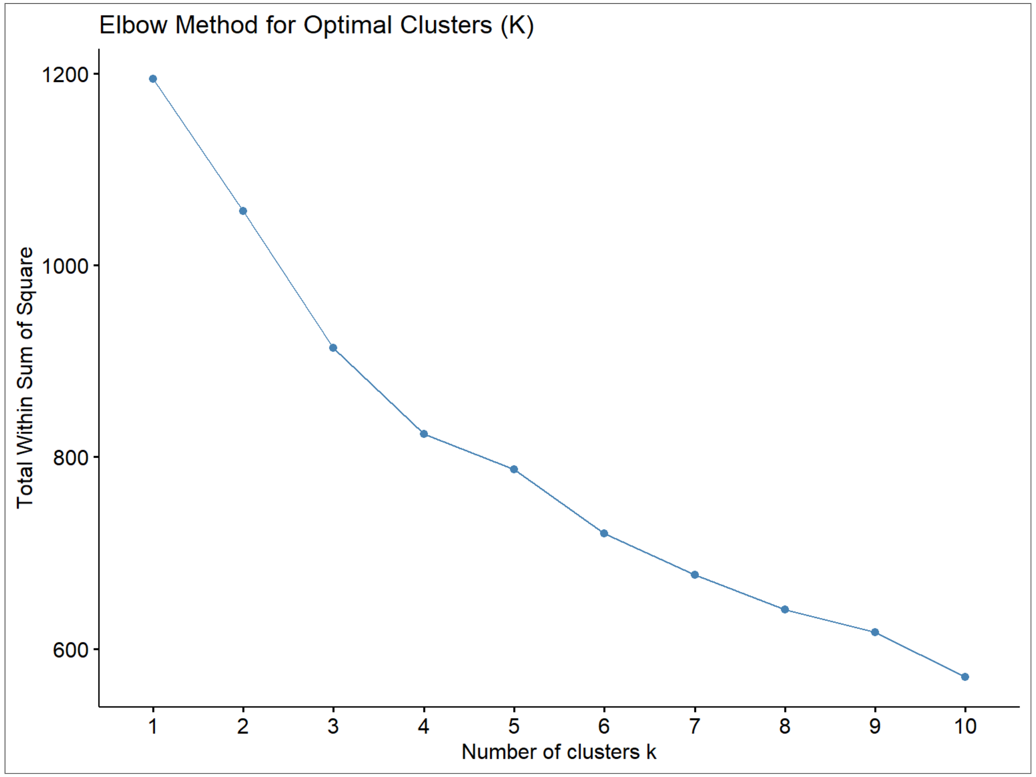
We'll use two common methods to help decide how many clusters (k) to use:

1. Elbow Method

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.4.3
```

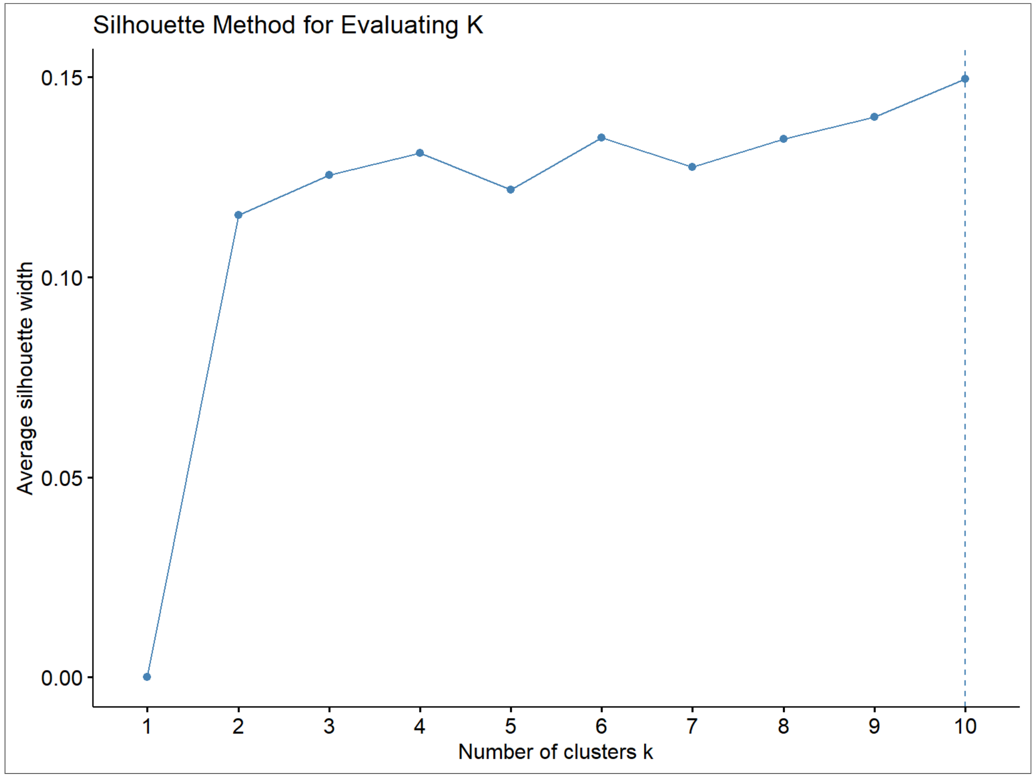
```
# Elbow method visualization
fviz_nbclust(cluster_scaled, kmeans, method = "wss") +
  labs(title = "Elbow Method for Optimal Clusters (K)")
```



Elbow Method for Optimal Clusters (K)
Total Within Sum of Square
1200
1000
800
600
1
2
3
4
5
6
7
8
9
10
Number of clusters k

2. Silhouette Method

```r
fviz_nbclust(cluster_scaled, kmeans, method = "silhouette") +
  labs(title = "Silhouette Method for Evaluating K")
```



5.4 Performing K-Means Clustering

Assuming both methods suggest **3 clusters**, we run the clustering:

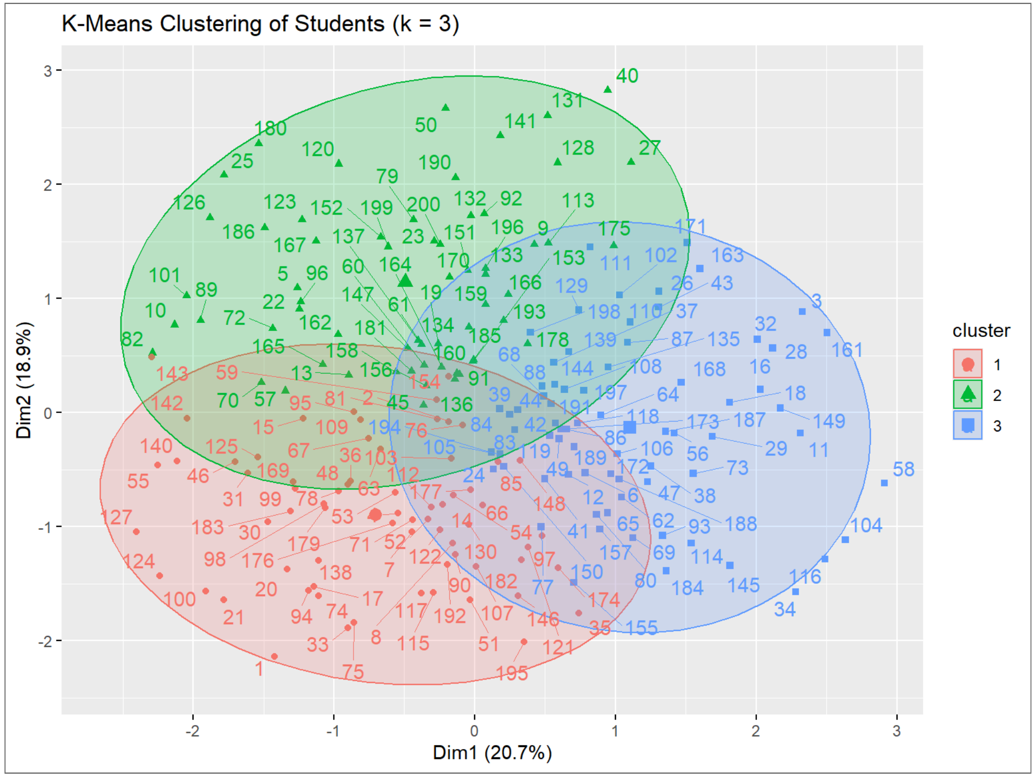
```r
# Set seed for reproducibility
set.seed(123)

# Apply K-means with 3 clusters
kmeans_model <- kmeans(cluster_scaled, centers = 3, nstart = 25)

# View cluster assignments
table(kmeans_model$cluster)
```

```
##
##  1  2  3
## 68 61 71
```

5.5 Visualizing K-Means Clusters

We use `fviz_cluster()` to plot the clustering results and see how students group together based on the variables we selected.

```r
fviz_cluster(kmeans_model, data = cluster_scaled,
             ellipse.type = "norm",
             main = "K-Means Clustering of Students (k = 3)",
             repel = TRUE)
```



5.6 Interpreting the Clusters

To understand what each cluster represents, we can **add cluster labels** to the original data and examine average characteristics per cluster.

```
# Add cluster label to unscaled data
clustered_data <- cluster_vars
clustered_data$cluster <- factor(kmeans_model$cluster)

# Summarize by cluster
cluster_summary <- clustered_data %>%
  group_by(cluster) %>%
  summarise(across(everything(), mean, .names = "avg_{.col}"))

cluster_summary
```

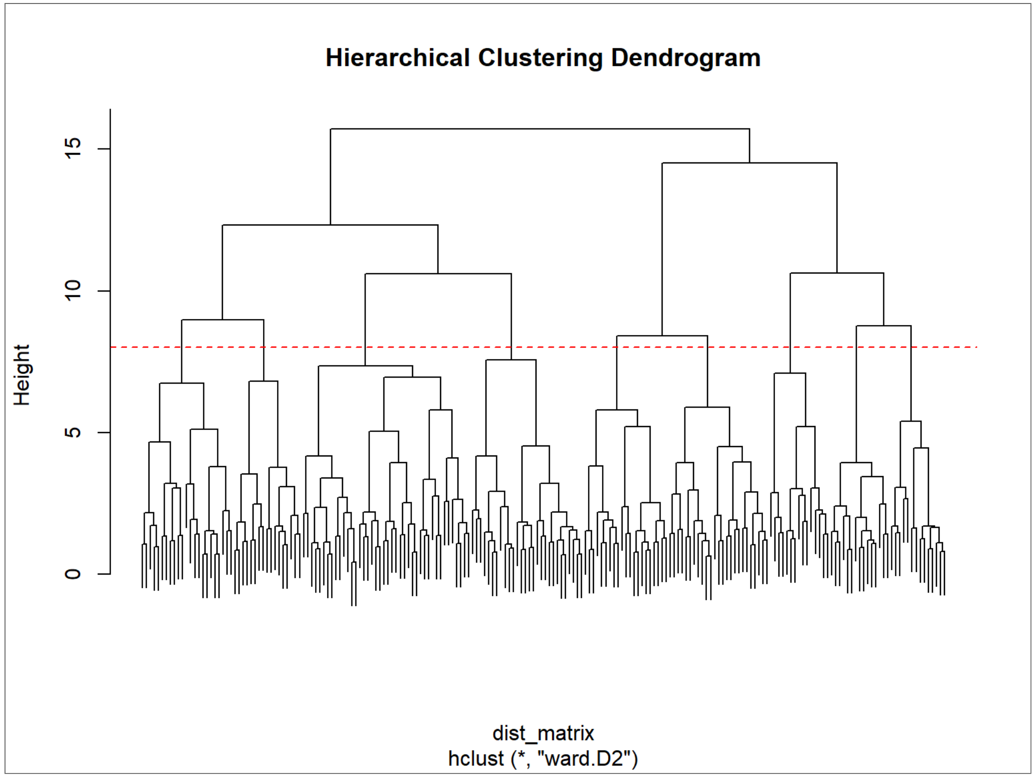
```
## # A tibble: 3 × 7
##   cluster avg_study_hours_per_week avg_sleep_hours_per_…¹ avg_exercise_freq_pe…²
##   <fct>                      <dbl>                  <dbl>                  <dbl>
## 1 1                           28.2                   5.80                   1.60
## 2 2                           25.3                   7.18                   3.95
## 3 3                           22.5                   6.39                   2.41
## # ℹ abbreviated names: ¹avg_sleep_hours_per_night, ²avg_exercise_freq_per_week
## # ℹ 3 more variables: avg_caffeine_intake_mg <dbl>, avg_stress_level <dbl>,
## #   avg_gpa <dbl>
```

This gives insight such as: - Cluster 1: Low stress, high GPA, moderate sleep - Cluster 2: High caffeine, low sleep, high stress - Cluster 3: High study hours, balanced profile

5.7 Bonus: Hierarchical Clustering

We can also use **hierarchical clustering** to visualize relationships between students using a dendrogram.

```r
# Compute distance matrix
dist_matrix <- dist(cluster_scaled)

# Apply hierarchical clustering
hclust_model <- hclust(dist_matrix, method = "ward.D2")

# Plot the dendrogram
plot(hclust_model, labels = FALSE, main = "Hierarchical Clustering Dendrogram")
abline(h = 8, col = "red", lty = 2)  # Example cut line
```


Hierarchical Clustering Dendrogram
Height
15
10
5
0
dist_matrix
hclust (*, "ward.D2")

Part 6: Dimensionality Reduction

In many datasets, particularly those involving **psychological or survey data**, variables can be highly correlated. For example, high stress may co-occur with burnout and low satisfaction. In such cases, it is useful to **reduce the dimensionality** of the data while preserving the essential patterns.

6.1 Research Objective

“Can we reduce psychological variables into latent wellbeing components?”

Rather than analyzing stress, burnout, and satisfaction separately, we want to discover **underlying dimensions** such as “mental strain” or “academic wellbeing” that explain most of the variation in the data.

6.2 What is Principal Component Analysis (PCA)?

PCA is a mathematical technique that transforms correlated variables into a smaller number of uncorrelated variables called **principal components** (PCs). Each principal component is a **linear combination** of the original variables.
Key goals: **- Reduce noise and redundancy - Visualize complex data in fewer dimensions - Identify latent constructs**

6.3 Variables for PCA

We'll use three psychological indicators from the dataset:

- `stress_level`
- `burnout_level`
- `satisfaction_academic`

6.4 Running PCA on the Dataset

First, we select and scale the relevant variables.

```r
# Load required library
library(factoextra)

# Select and scale relevant variables
pca_vars <- data %>%
  select(stress_level, burnout_level, satisfaction_academic) %>%
  na.omit()

pca_scaled <- scale(pca_vars)
```

6.5 Performing PCA

We now run PCA on the scaled variables.

```
# Run PCA
pca_result <- prcomp(pca_scaled, center = TRUE, scale. = TRUE)

# View PCA results
summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2    PC3
## Standard deviation     1.0455 0.9990 0.9534
## Proportion of Variance 0.3644 0.3327 0.3030
## Cumulative Proportion  0.3644 0.6970 1.0000
```

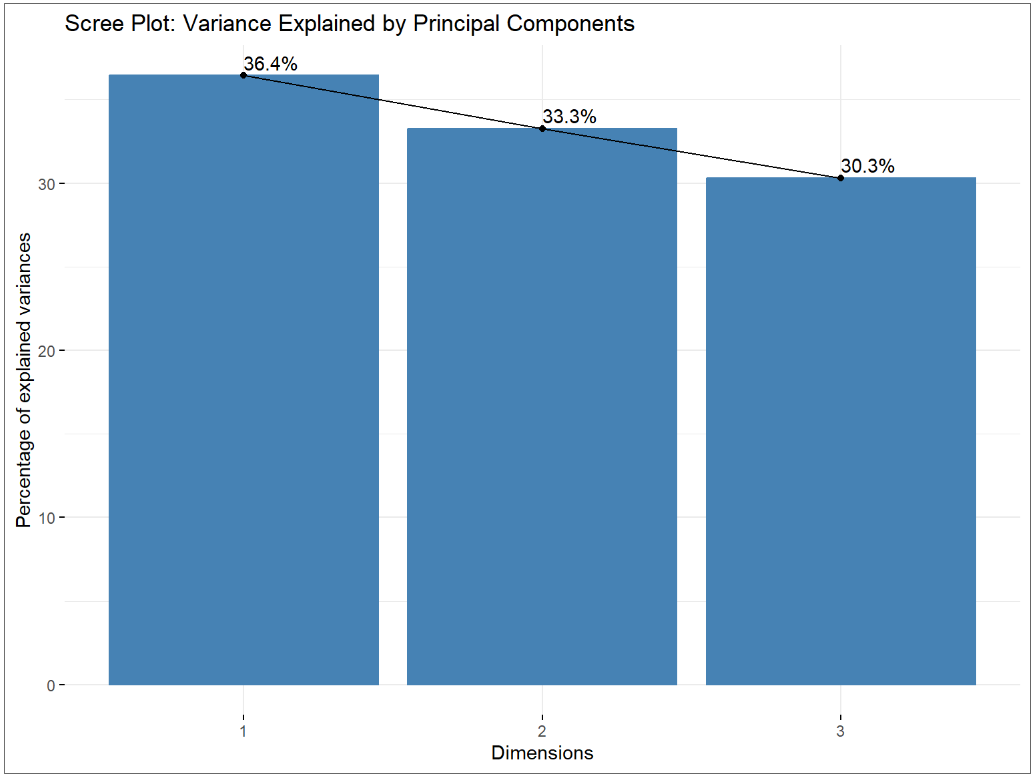
The summary shows:

- **Standard deviation** of each principal component
- **Proportion of variance** explained by each PC
- **Cumulative proportion** — how many components explain a sufficient portion of total variance

6.6 Visualizing PCA Results

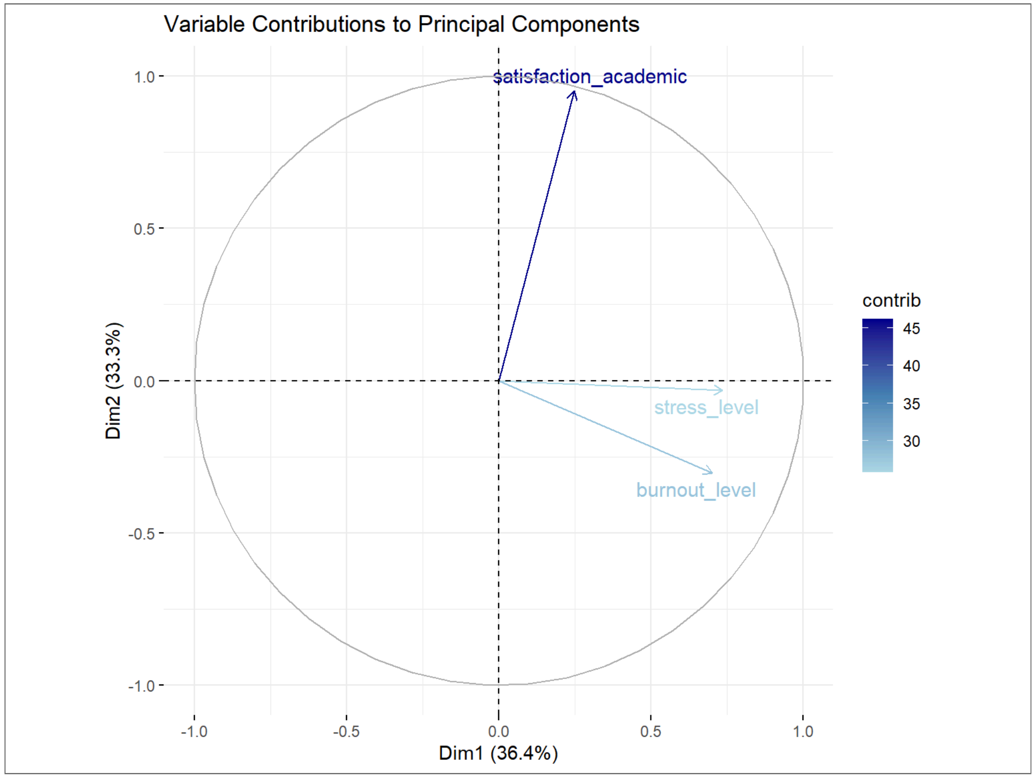
Scree Plot: How much variance each component explains

```r
fviz_eig(pca_result, addlabels = TRUE, barfill = "steelblue") +
  labs(title = "Scree Plot: Variance Explained by Principal Components")
```



Contribution of Each Variable to PCs

```
fviz_pca_var(pca_result, col.var = "contrib",
             gradient.cols = c("lightblue", "steelblue", "darkblue"),
             repel = TRUE) +
  labs(title = "Variable Contributions to Principal Components")
```



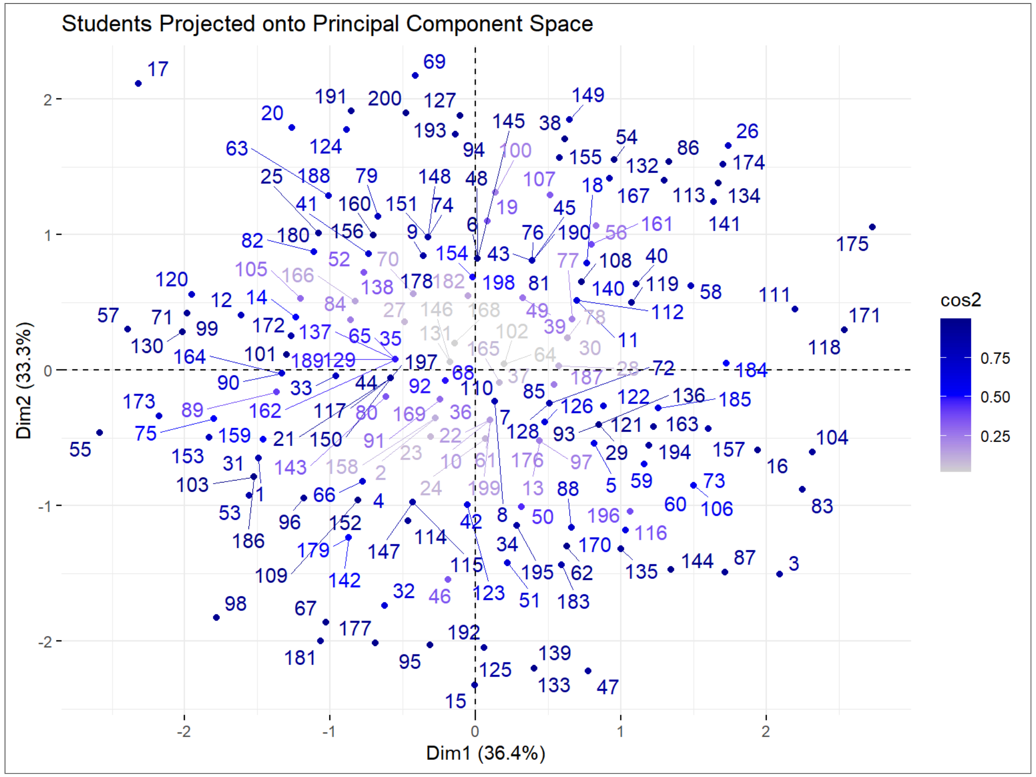
This plot helps interpret the **loadings** — how much each original variable contributes to the principal components:

- PC1 may represent **psychological strain**
- PC2 might capture **academic satisfaction**

6.7 Projecting Individuals (Optional)

You can also project each student (individual) onto the PC space.

```r
fviz_pca_ind(pca_result,
             col.ind = "cos2", # color by quality of representation
             gradient.cols = c("lightgray", "blue", "darkblue"),
             repel = TRUE) +
  labs(title = "Students Projected onto Principal Component Space")
```



This visualization can help you cluster or segment students based on their psychological profiles.

6.8 Interpretation in Research

In your thesis, PCA allows you to:

- Reduce dimensionality before clustering or regression
- Build **composite wellbeing indices**
- Interpret latent constructs (e.g., mental strain, engagement)

You can also use PCA as a **preprocessing step** to reduce multicollinearity in predictive models.

Part 7: Reproducible Research and Ethical Modeling

One of the most important (and often overlooked) aspects of data analysis is not just conducting the analysis, but **documenting it clearly** and **sharing it responsibly**. This ensures that others can verify, reproduce, and build upon your findings — a core principle of scientific integrity.

7.1 Reproducible Research with R Markdown

Reproducible research means that anyone with access to your code and data can **run your analysis and get the same results**.
R Markdown supports reproducibility by combining: - Narrative text (your explanations and interpretations) - R code (your analysis) - Output (tables, plots, summaries)
All in one dynamic document.
Benefits: - Transparency - Easier collaboration with supervisors or peers - Faster review and publication processes

Example Header in R Markdown

```yaml
---
title: "Sara's Thesis Data Analysis"
author: "Sara Ahmed"
output: html_document
---
```

Best Practices for Reproducibility

- Always **set a seed** for randomness (e.g., `set.seed(123)`)
- Include **all data cleaning and transformation steps**
- Use **relative paths** and consistent folder structures
- Label code chunks clearly and avoid overly long scripts

7.2 Ethical Considerations in Data Modeling

Beyond technical quality, every research project must consider **ethical implications** in how data is used, shared, and interpreted.

1. Bias in Modeling

Models may reflect or even amplify existing biases if not properly evaluated.

Examples: - Using GPA as the only measure of success may disadvantage students with learning differences. - Models trained on unbalanced data can misclassify underrepresented groups.

Mitigation strategies:

- Analyze model performance across subgroups (e.g., gender)
- Report fairness metrics (e.g., equal opportunity, demographic parity)
- Avoid using sensitive attributes (e.g., race, religion) as predictors without justification

2. Privacy and Data Security

Respect for participant privacy is a fundamental research principle. This includes:

- Removing personally identifiable information (PII)
- Masking or anonymizing student IDs before sharing
- Not publishing raw datasets without consent or clearance

If sharing data, always check with:

- University data governance guidelines
- Ethical review board policies
- Informed consent agreements

3. Responsible Code Sharing

If you publish your thesis or results online:

- Include a `README.md` explaining the purpose of the scripts
- Host code in a public or private Git repository (e.g., GitHub, GitLab)
- Share only derived or anonymized data when allowed

Licensing your code with an open license (e.g., MIT, GPL) also helps others reuse it properly.

Part 8: Beyond R — Other Tools for Data Analysis

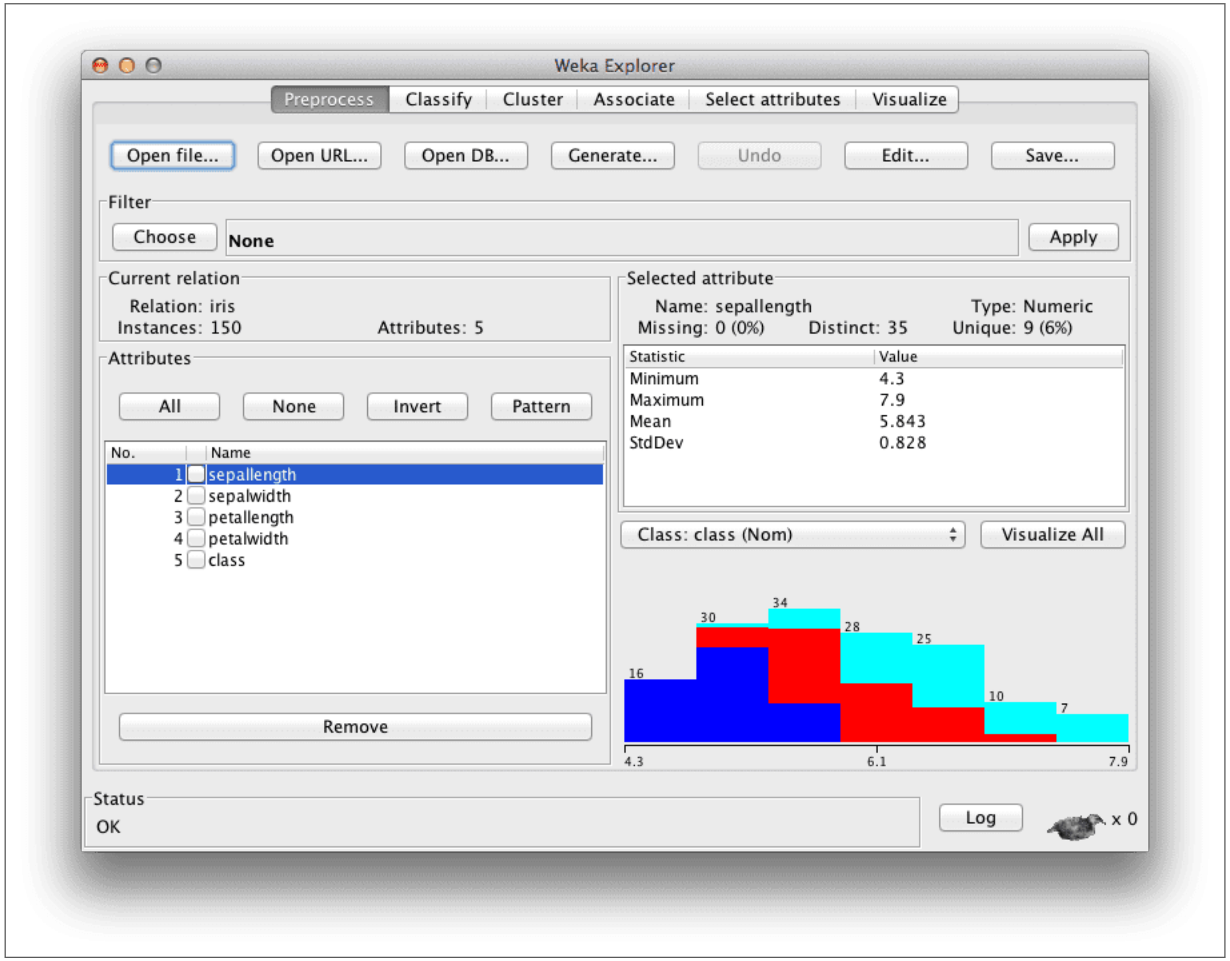
While R is a powerful and open-source language for data analysis, it is not the only option available to researchers. Depending on the project scope, field of study, available resources, and the user's background, other tools might be more suitable for certain types of analysis or visualization. This section introduces key alternatives and their unique strengths.

1. Python Ecosystem

Python has become one of the most popular languages in data science. Its rich ecosystem of libraries makes it highly flexible for everything from data wrangling to deep learning.

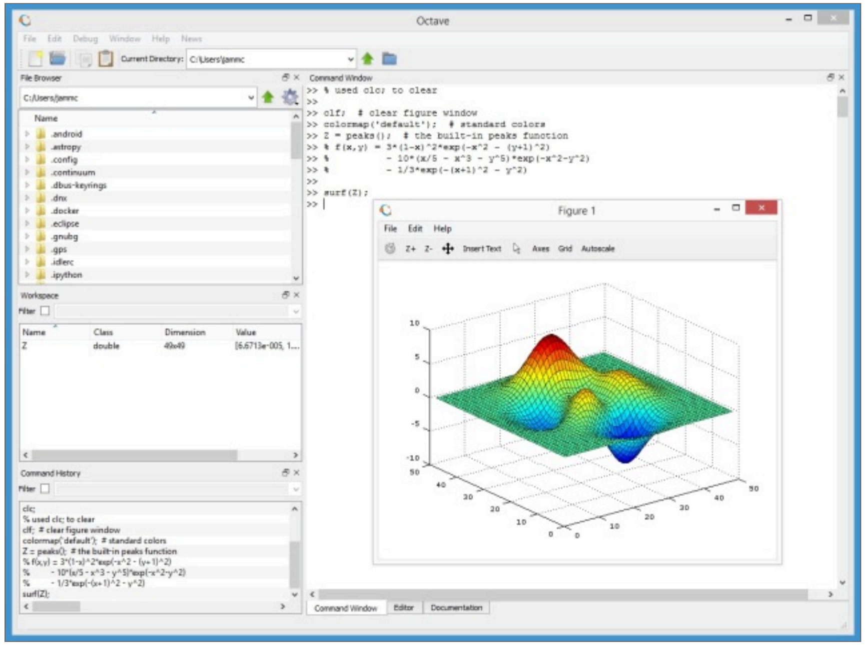
2. Weka

Weka is a beginner-friendly, GUI-based platform developed at the University of Waikato for teaching and prototyping machine learning.



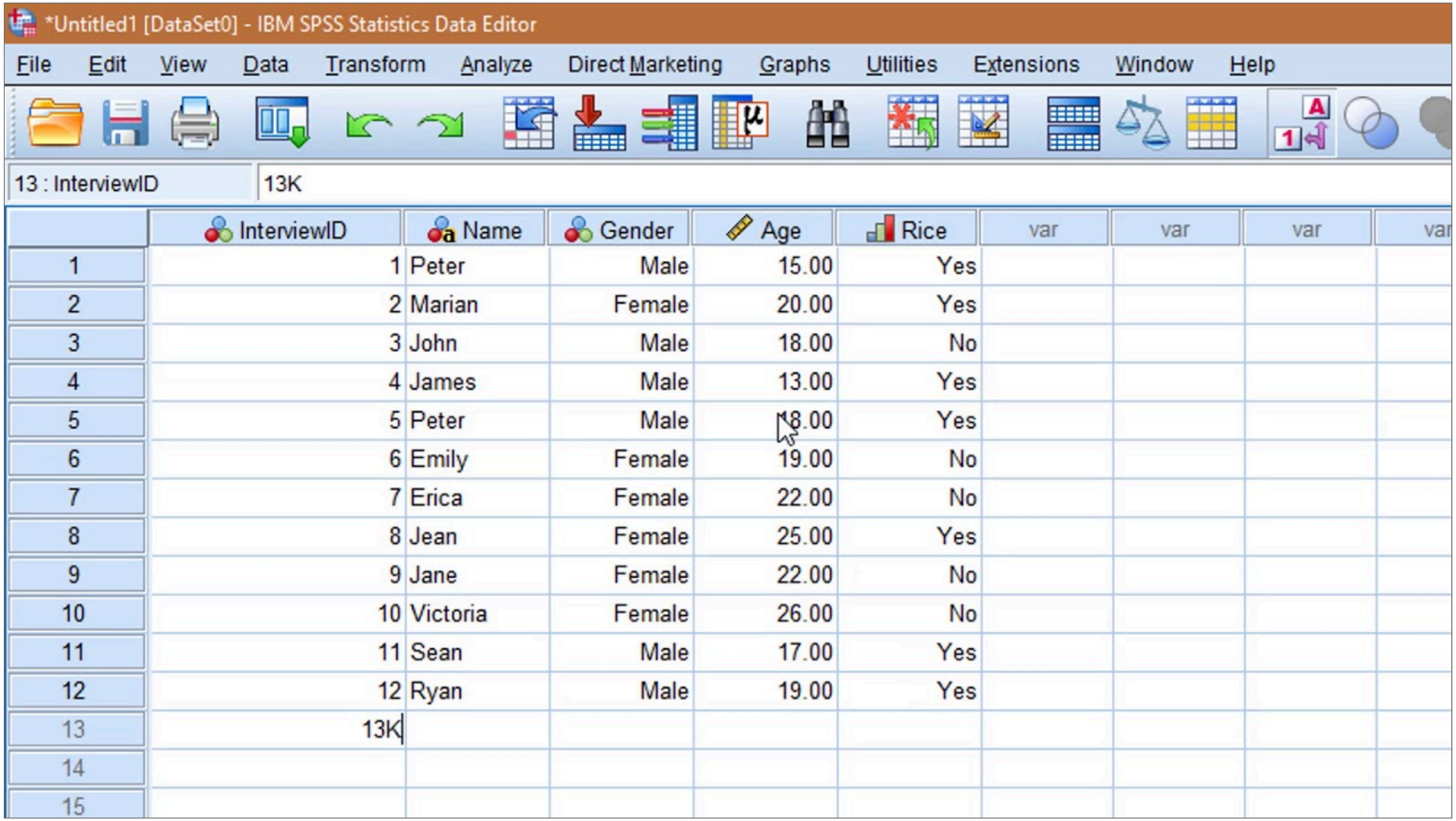
3. GNU Octave

GNU Octave is a high-level interpreted language, mostly compatible with MATLAB, used mainly for numerical computations.



4. SPSS

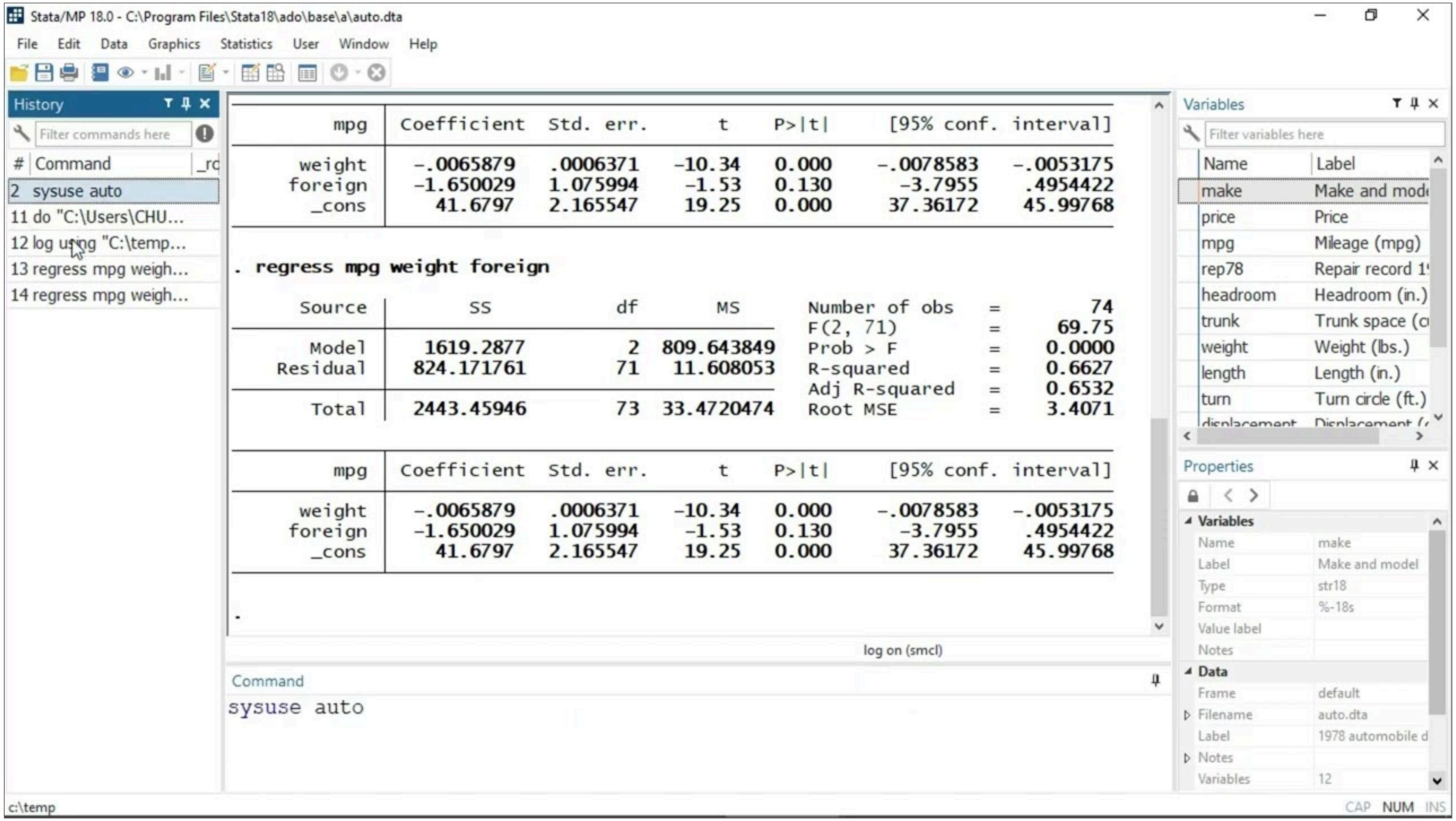
SPSS is a GUI-based software package from IBM, widely used in social sciences for statistical analysis and data management.



	InterviewID	Name	Gender	Age	Rice	var	var	var	var
1	1	Peter	Male	15.00	Yes				
2	2	Marian	Female	20.00	Yes				
3	3	John	Male	18.00	No				
4	4	James	Male	13.00	Yes				
5	5	Peter	Male	18.00	Yes				
6	6	Emily	Female	19.00	No				
7	7	Erica	Female	22.00	No				
8	8	Jean	Female	25.00	Yes				
9	9	Jane	Female	22.00	No				
10	10	Victoria	Female	26.00	No				
11	11	Sean	Male	17.00	Yes				
12	12	Ryan	Male	19.00	Yes				
13	13K								
14									
15									

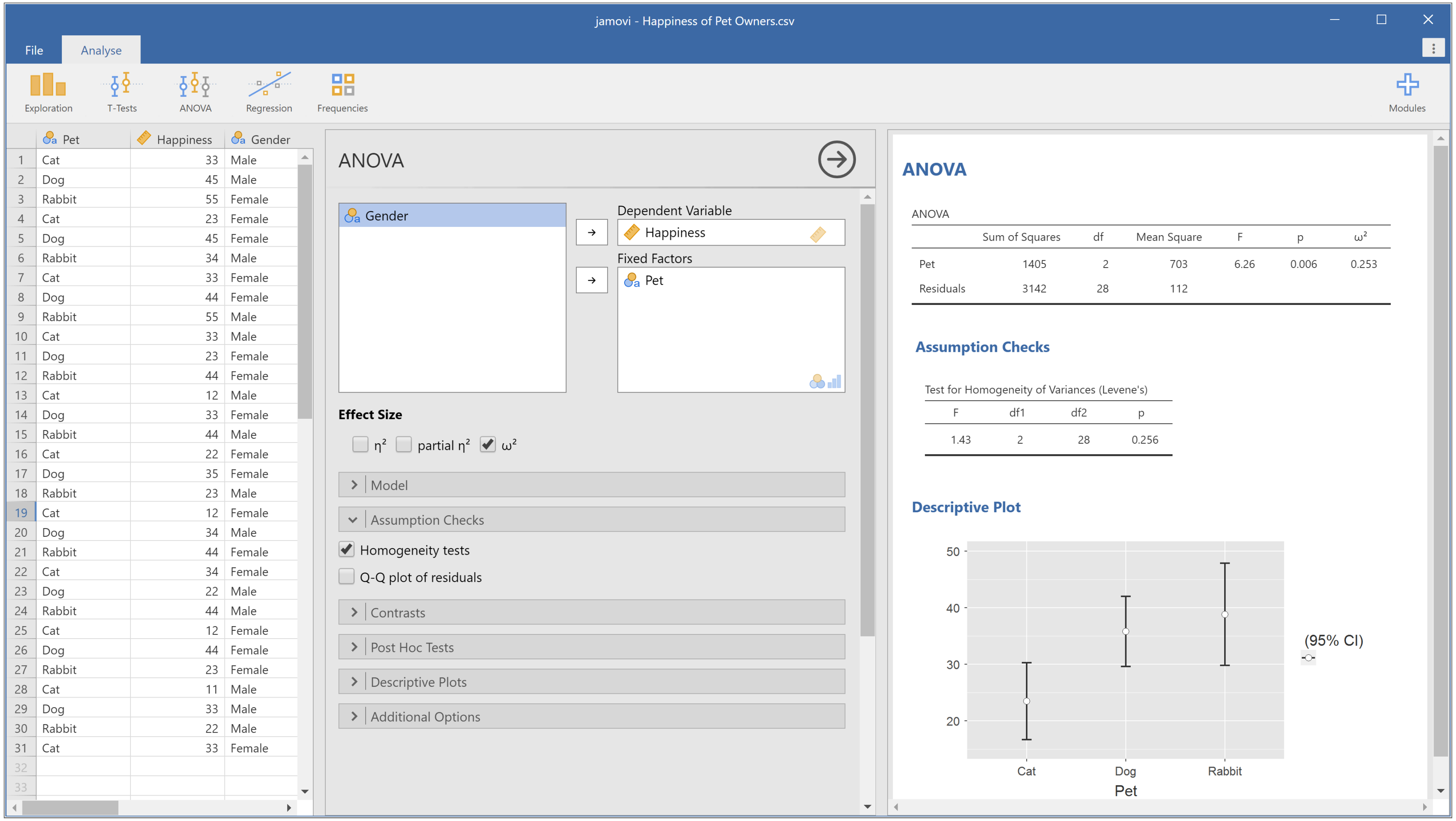
5. Stata

Stata is widely used in fields like economics, biostatistics, and epidemiology for its strong statistical modeling capabilities.



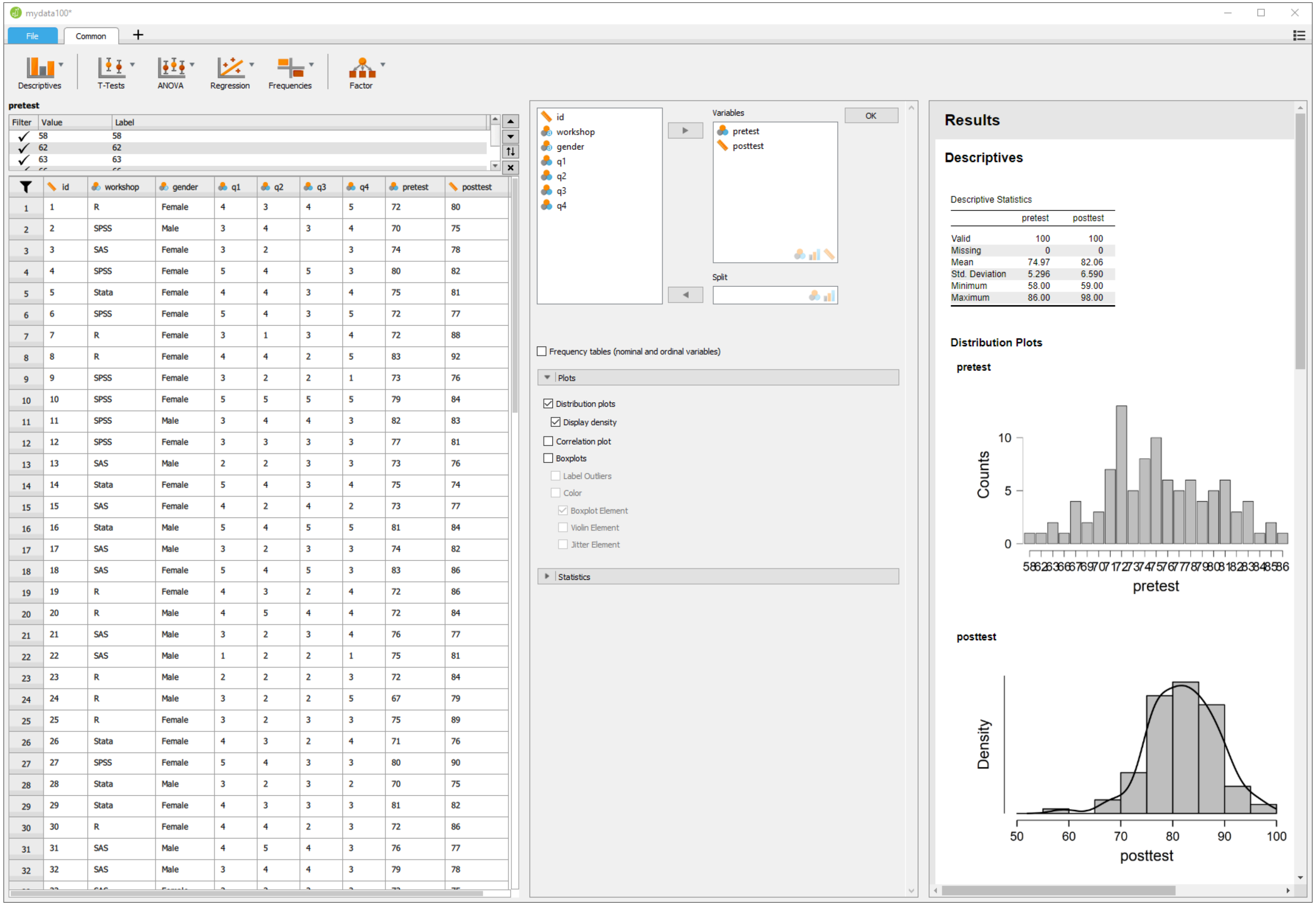
6. Jamovi

Jamovi is a user-friendly, open-source alternative to SPSS that is built on top of R. It offers a point-and-click interface while retaining full access to R scripting for advanced users.



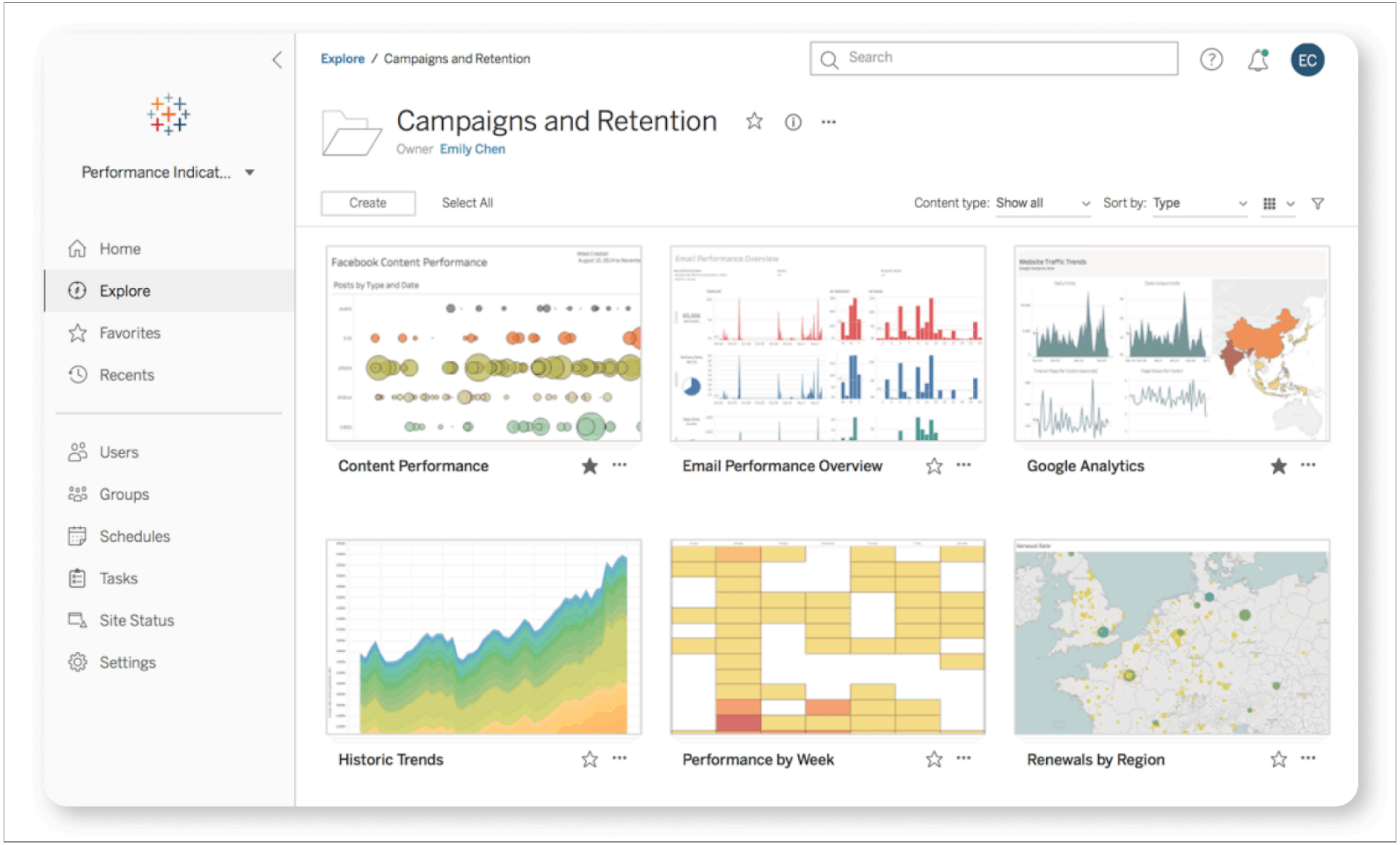
7. JASP

JASP is an open-source alternative to SPSS with a special focus on **Bayesian statistics**. It is increasingly used in psychology and social sciences.


mydata100*
File
Common
Descriptives
T-Tests
ANOVA
Regression
Frequencies
Factor
pretest
Filter Value Label
id workshop gender q1 q2 q3 q4 pretest posttest
Variables
pretest
posttest
Split
OK
Frequency tables (nominal and ordinal variables)
Plots
Distribution plots
Display density
Correlation plot
Boxplots
Label Outliers
Color
Boxplot Element
Violin Element
Jitter Element
Statistics
Results
Descriptives
Descriptive Statistics
pretest posttest
Valid 100 100
Missing 0 0
Mean 74.97 82.06
Std. Deviation 5.296 6.590
Minimum 58.00 59.00
Maximum 86.00 98.00
Distribution Plots
pretest
Counts
pretest
posttest
Density
posttest

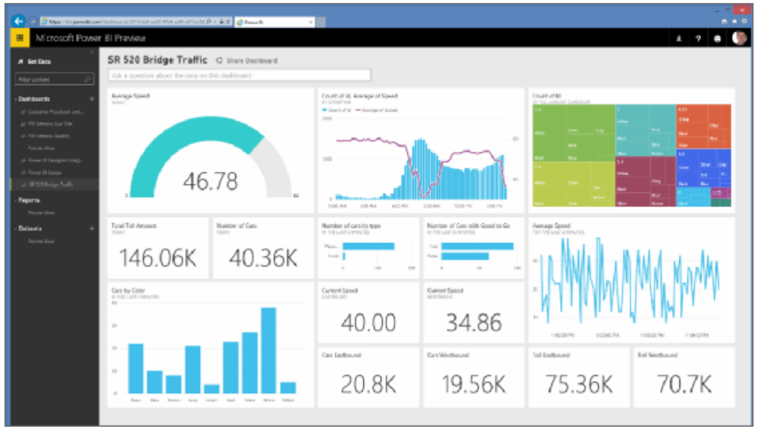
8. Tableau

Tableau is a leading business intelligence and data visualization platform, used to turn data into interactive dashboards and compelling visuals.


Explore / Campaigns and Retention
Search
EC
Performance Indicat...
Campaigns and Retention
Owner Emily Chen
Create
Select All
Content type: Show all
Sort by: Type
Home
Explore
Favorites
Recents
Users
Groups
Schedules
Tasks
Site Status
Settings
Facebook Content Performance
Posts by Type and Date
Content Performance
Email Performance Overview
Website Traffic Trends
Google Analytics
Historic Trends
Performance by Week
Renewals by Region

9. Microsoft Power BI

Microsoft Power BI is a powerful data visualization tool designed for business intelligence, with deep integration across Microsoft Office tools.



Choosing the Right Tool

Tool	Best For	Notes
R	Academic data analysis and stats	Open-source, rich packages
Python	Machine learning, automation, flexibility	Powerful and widely supported
SPSS/Stata	Social science and economic modeling	GUI-based but expensive
Jamovi/JASP	Stats education and Bayesian analysis	Good for reproducibility
Tableau/Power BI	Interactive dashboards and business reporting	Best for storytelling and exec reports
Octave	Engineering and math modeling	MATLAB compatibility for free
Weka	Teaching machine learning	Great for non-coders, limited scalability