🧠 AI Developer Productivity Dataset Analysis

Burak ER, 122200065

June,2025

Dataset overview

- In this analysis, I examine the *ai_dev_productivity.csv* dataset which simulates the behavior and productivity of AI developers over 500 days.

Column Name	Description
`hours_coding`	Total focused hours spent on software development work (0–12 hours).
`coffee_intake_mg`	Daily caffeine intake in milligrams (0–600 mg).
`distractions`	Number of distractions (e.g., meetings, Slack notifications) (0–10).
`sleep_hours`	Number of hours of sleep the previous night (3–10 hours).
`commits`	Number of code commits pushed during the day (0–20).
`bugs_reported`	Number of bugs reported in code written that day (0–10).
`ai_usage_hours`	Number of hours spent using AI tools (e.g., ChatGPT, Copilot) (0–12).
`cognitive_load`	Self-reported mental strain on a scale of 1 to 10.
`task_success`	**Target column** — whether the daily productivity goal was achieved (0/1).

```
> dim(data)
[1] 500   9
> str(data)
'data.frame':	500 obs. of  9 variables:
 $ hours_coding    : num  5.99 4.72 6.3 8.05 4.53 4.53 8.16 6.53 4.06 6.09 ..
 $ coffee_intake_mg: int  600 568 560 600 421 429 600 600 409 567 ...
 $ distractions    : int  1 2 1 7 6 1 1 4 5 5 ...
 $ sleep_hours     : num  5.8 6.9 8.9 6.3 6.9 7.1 8.3 3.6 6.1 7.3 ...
 $ commits         : int  2 5 2 9 4 5 6 9 6 7 ...
 $ bugs_reported   : int  1 3 0 5 0 0 0 3 2 0 ...
 $ ai_usage_hours  : num  0.71 1.75 2.27 1.4 1.26 3.06 0.3 1.47 2.43 2.11 ...
 $ cognitive_load  : num  5.4 4.7 2.2 5.9 6.3 3.9 2.2 9.1 7 5.1 ...
 $ task_success    : int  1 1 1 0 1 1 1 0 0 1 ...
```

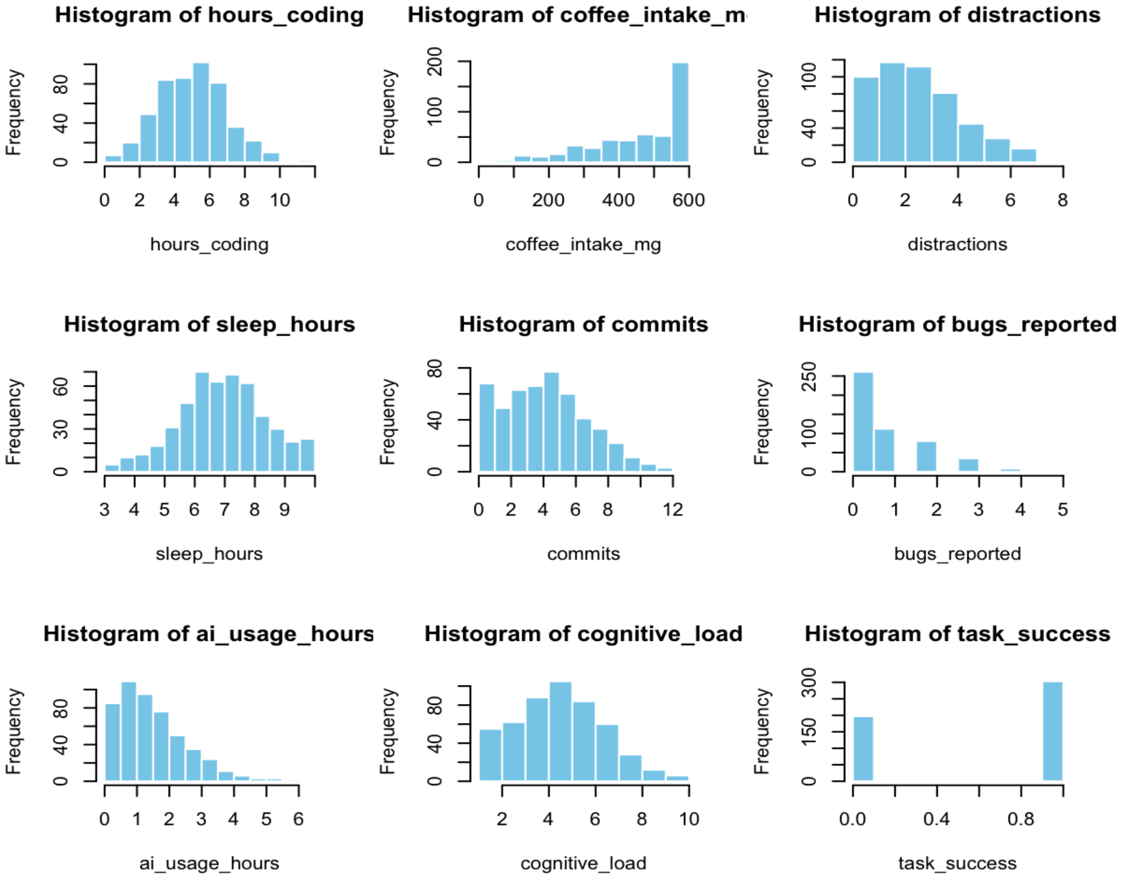
Dataset overview

```
par(mfrow=c(3, 3))
for (col_name in names(data)) {
  hist(data[[col_name]],
       main = paste("Histogram of", col_name),
       xlab = col_name,
       col = "skyblue",
       border = "white")
}
```

At first glance we can observe

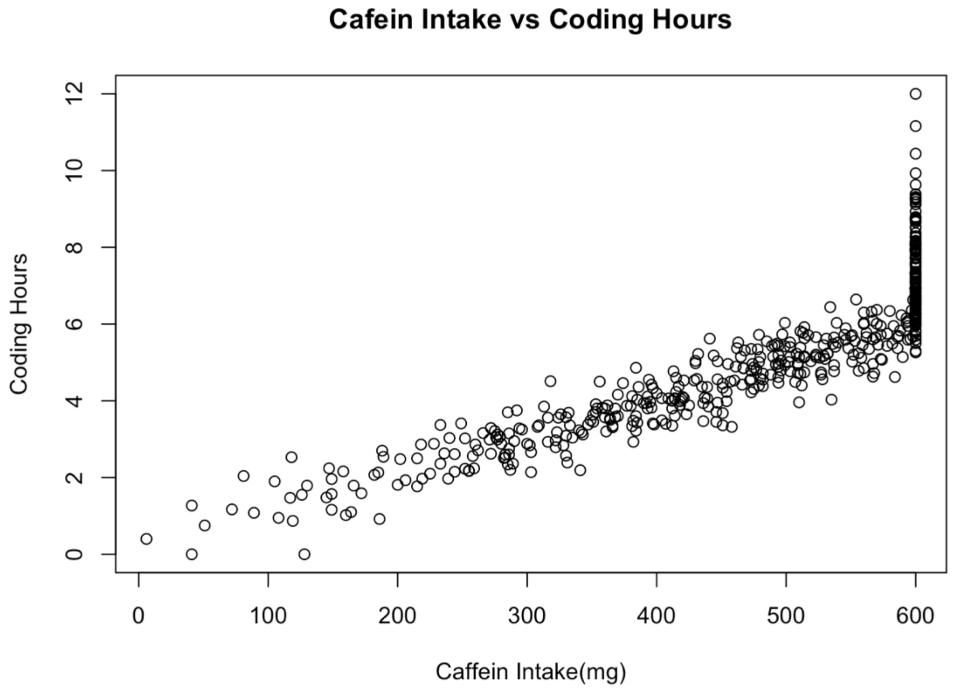
- slightly normally distributed among `hours_coding`, *`sleep_hours`*, *`cognitive_laod`* columns.
- And *succeded tasks* are more than failure.
- `coffe_inatake_mg` is generally based on 550-600 mg.
- etc.

Lets look for other insights about the dataset



Coffee impact on coding hours ☕🤝🧑‍💻

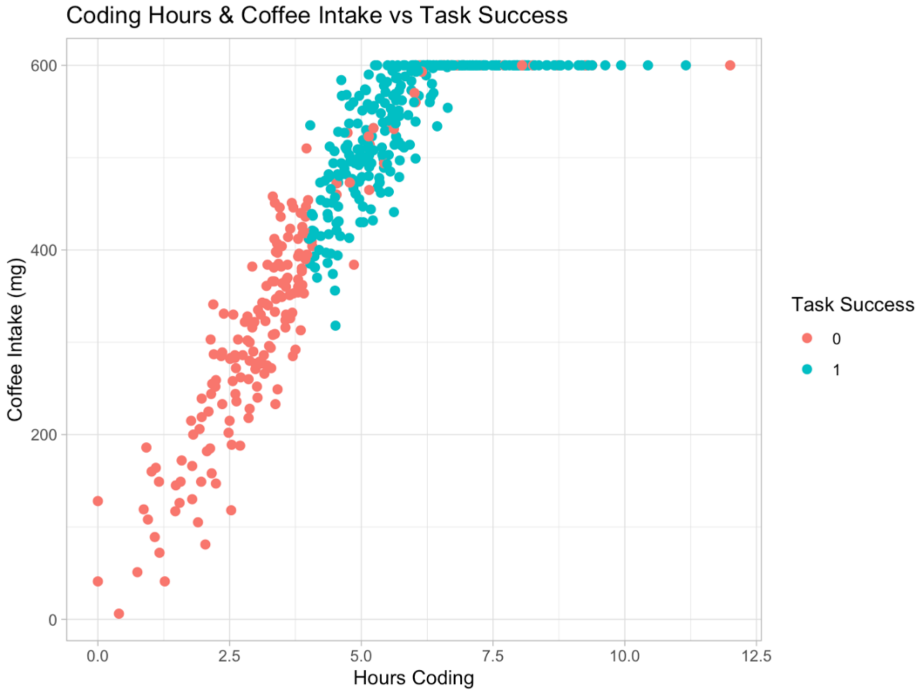
- `hours_coding` is highly correlated with `coffee_intake_mg` column with correlation coefficient of 0.89

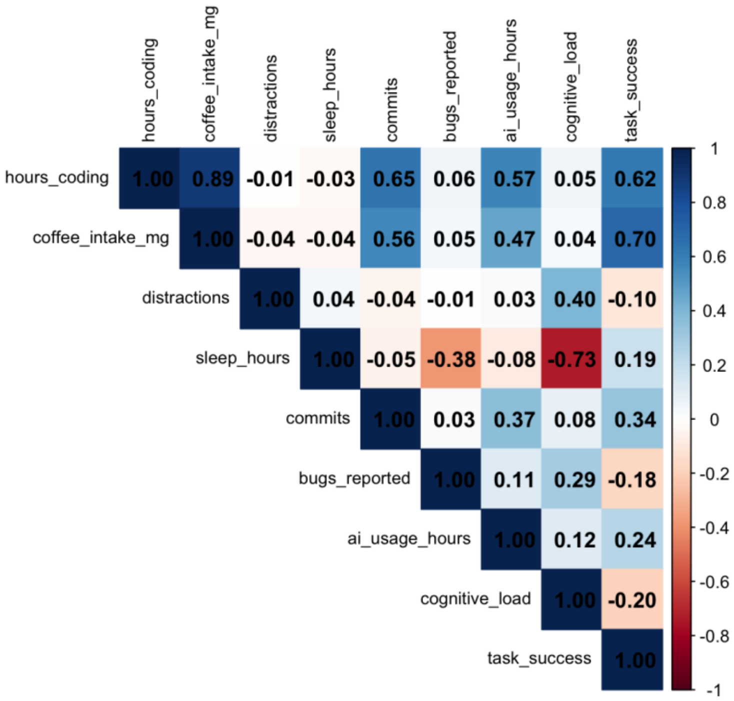



```
> cor(data$coffee_intake_mg, data$hours_coding)
[1] 0.8898159
```

Determination + coffee = success 💪🏻

- As we can see the the `task_success(which we are going to predict)` is highly correlated with `coffe_intake_mg` and `hours_coding`



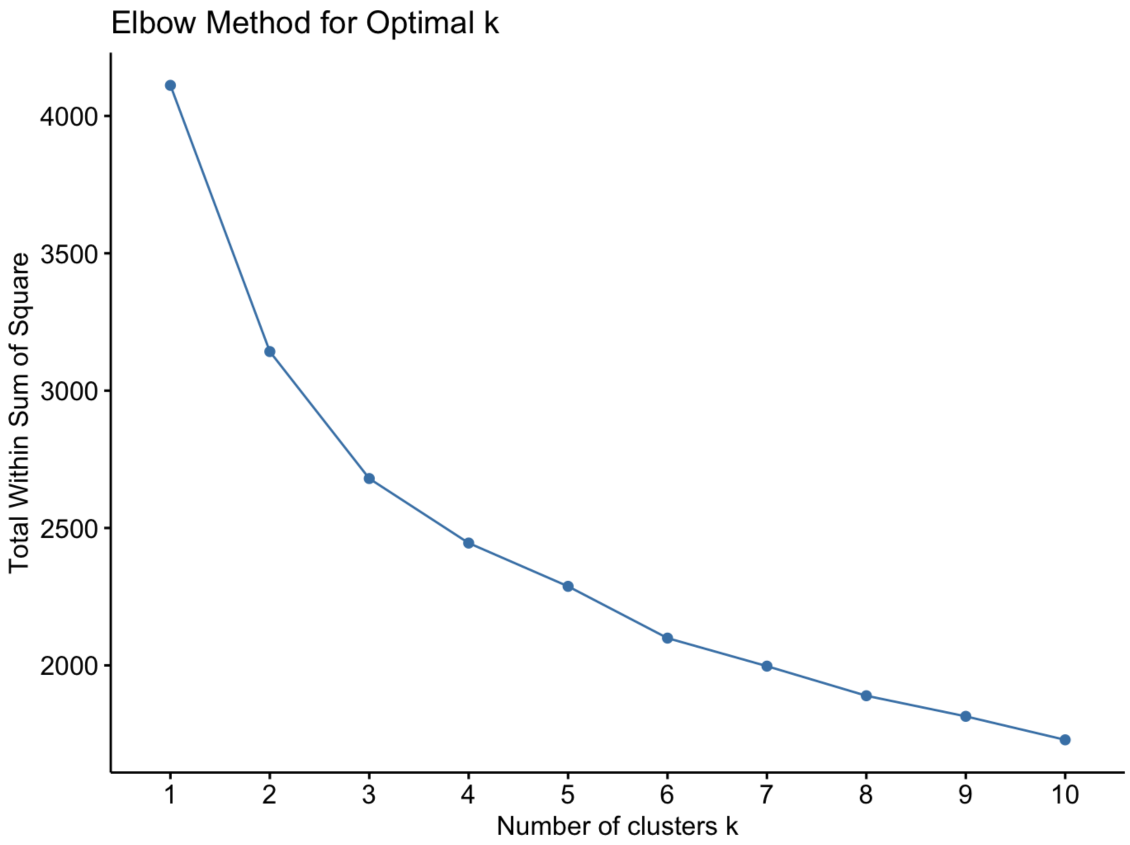



```
ggplot(data, aes(x = hours_coding, y = coffee_intake_mg, color = factor(task_success))) +
  geom_point(alpha = 1, size = 2) +
  labs(title = "Coding Hours & Coffee Intake vs Task Success", x = "Hours Coding", y = "Coffee Intake (mg)", color = "Task Success") +
  theme_light()
```

Correlation matrix

Clustering the data

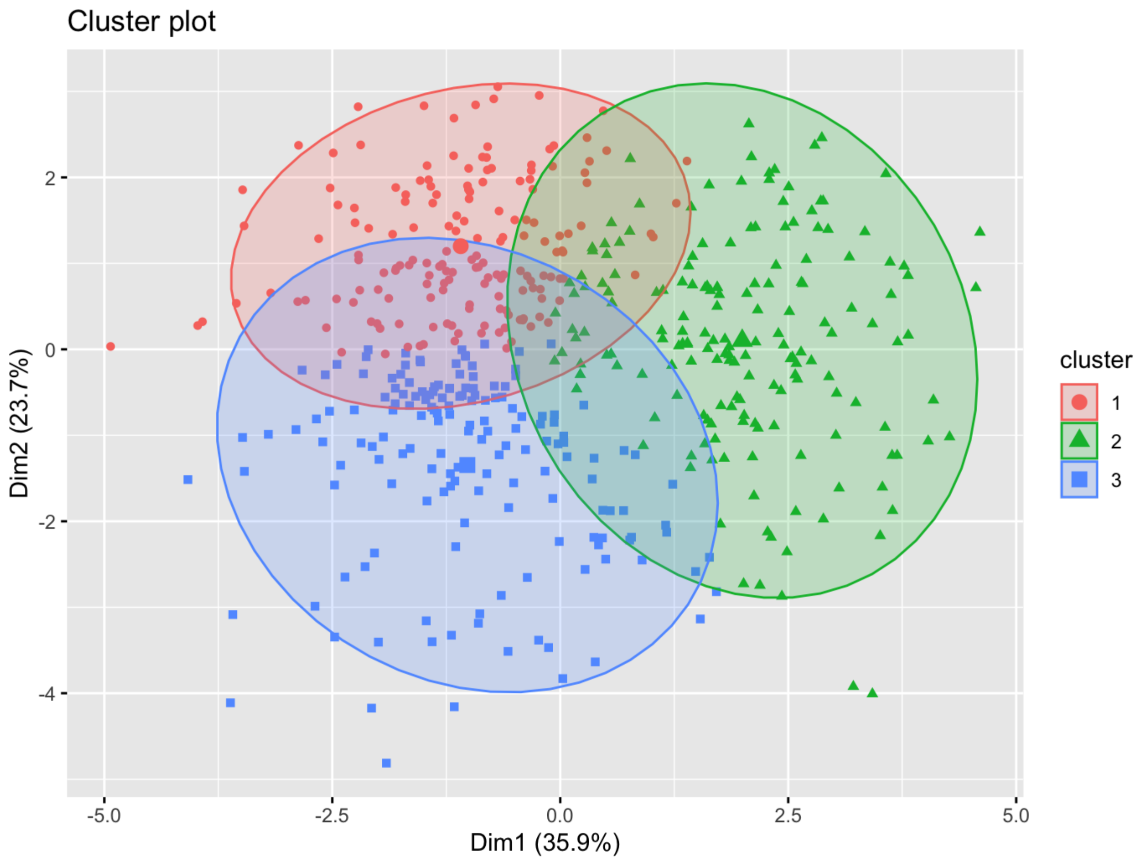
- Since we will cluster values from `ai_usage_hours and coffee_intake_mg`, which can differ in type, I chose to **scale** the data.
- After scaling is done, I perform **elbow method** to chose the best k for the algorithm.
- I picked **3** for the k.




```
data2 <- data[c("hours_coding", "coffee_intake_mg", "distractions", "sleep_hours", "commits", "bugs
#Scale the data before clustering
scaled_data <- scale(data2)
#Elbow method for optimal k to do kmeans
fviz_nbclust(scaled_data, kmeans, method = "wss") + labs(title = "Elbow Method for Optimal k")
```

Clustering the data 🗂️

- Performing **K-means** algorithm to cluster the data
- Visualizing the clusters by applying **PCA** reducing dimentions to **2**.




```r
#Applying kmeans with k=3
kmeans_result <- kmeans(scaled_data, centers = 3)
#PCA to visualize the clusters
fviz_cluster(kmeans_result, data = scaled_data,
             geom = "point",
             ellipse.type = "norm",)
```

What are these clusters?

```
> #Avg values for the clusters
> aggregate(. ~ cluster, data = data2, FUN = mean)
  cluster hours_coding coffee_intake_mg distractions sleep_hours  commits bugs_reported ai_usage_hours cognitive_load
1       1     6.049759         543.7169     2.506024    8.004217 5.795181     0.3614458      1.7200602       3.212048
2       2     3.045838         306.3873     3.017341    7.070520 2.578035     0.6531792      0.8110983       4.282659
3       3     6.065901         548.6584     3.416149    5.813665 5.565217     1.5900621      2.0472671       6.055901
> |
```

- The K-means clustering algorithm grouped the developer's daily activities into **3 distinct clusters**, each representing a specific **work pattern.**

✅ **Cluster 1 – Balanced and Productive Days**

- Coding hours: **6.05**
- Coffee intake: **543 mg**
- Sleep hours: **8.00**
- Cognitive load: **3.21**

📌 These are the developer's most **balanced days**: well-rested, moderately caffeinated, and highly productive with low mental effort. AI tools are used efficiently, and bug reports remain low. These days reflect a **sustainable and healthy work mode**.

⚠️ **Cluster 2 – Low Energy / Unproductive Days**

- Coding hours: **3.04 (lowest)**
- Coffee intake: **306 mg**
- Sleep hours: **7.07**
- Cognitive load: **4.28**

📌 This cluster represents **low-performance days** with reduced coding activity, lower caffeine intake, and minimal AI usage. Commits are fewer, and task success is less likely. These days may indicate **lack of motivation, distractions, or recovery phases**.

🔥 **Cluster 3 – Overloaded and Fatigued Days**

- Coding hours: **6.07**
- Coffee intake: **548 mg**
- Sleep hours: **5.81 (lowest)**
- Cognitive load: **6.05 (highest)**

📌 These are the **high-pressure days** where the developer pushes productivity at the cost of sleep. Caffeine consumption and AI usage are at their peak. However, cognitive load increases significantly, and more bugs are reported. This pattern could signal **burnout risk** or **unsustainable overworking**.

Detecting outliers 🔍

- I used IQR method to detect the outliers. Each value can be seen below.
- No outliers on **`bugs_reported`**

```
> outliers_list <- lapply(data, function(x) boxplot.stats(x)$out);
> outliers_list
```

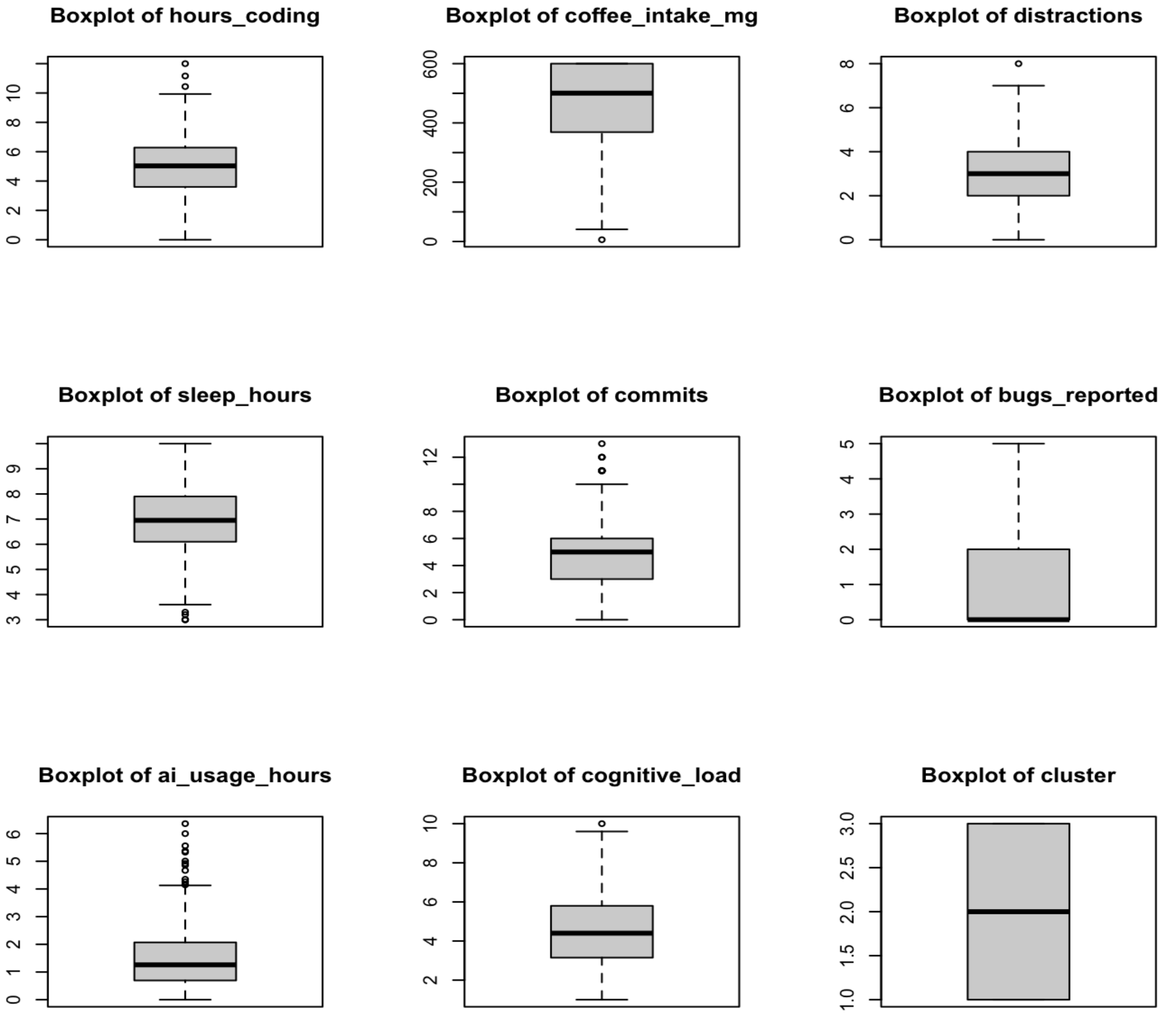
```
$hours_coding
[1] 10.44 12.00 11.16

$coffee_intake_mg
[1] 6

$distractions
[1] 8

$sleep_hours
[1] 3.2 3.0 3.0 3.0 3.3

$commits
 [1] 12 12 11 11 13 12 11 11 11 11
```

```
$bugs_reported
integer(0)

$ai_usage_hours
 [1] 4.26 5.33 5.56 5.37 6.00 4.16 4.92 4.67 5.01 4.15 4.86 4.35 6.36

$cognitive_load
[1] 10

$task_success
integer(0)
```

```
> sapply(outliers_list, length)
    hours_coding coffee_intake_mg     distractions      sleep_hours          commits    bugs_reported   ai_usage_hours   cognitive_load 
               3                1                1                5               10                0               13                1 
    task_success 
               0 
```

# of outliers for each column

Detecting outliers 🔍



Logistic regression 📊

```
> log_model <- glm(task_success ~ coffee_intake_mg, data = data, family = "binomial")
> summary(log_model)

Call:
glm(formula = task_success ~ coffee_intake_mg, family = "binomial",
    data = data)

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -7.183274   0.662291  -10.85   <2e-16 ***
coffee_intake_mg  0.016559   0.001402   11.81   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 670.50  on 499  degrees of freedom
Residual deviance: 374.74  on 498  degrees of freedom
AIC: 378.74

Number of Fisher Scoring iterations: 5
```

- - I applied a **logistic regression** model to evaluate the impact of `coffee_intake_mg` on the binary outcome task_success.
- coffee intake is a **strong and statistically significant predictor**

($\beta = 0.0166$(mg), $p < 0.001$).

Logistic regression

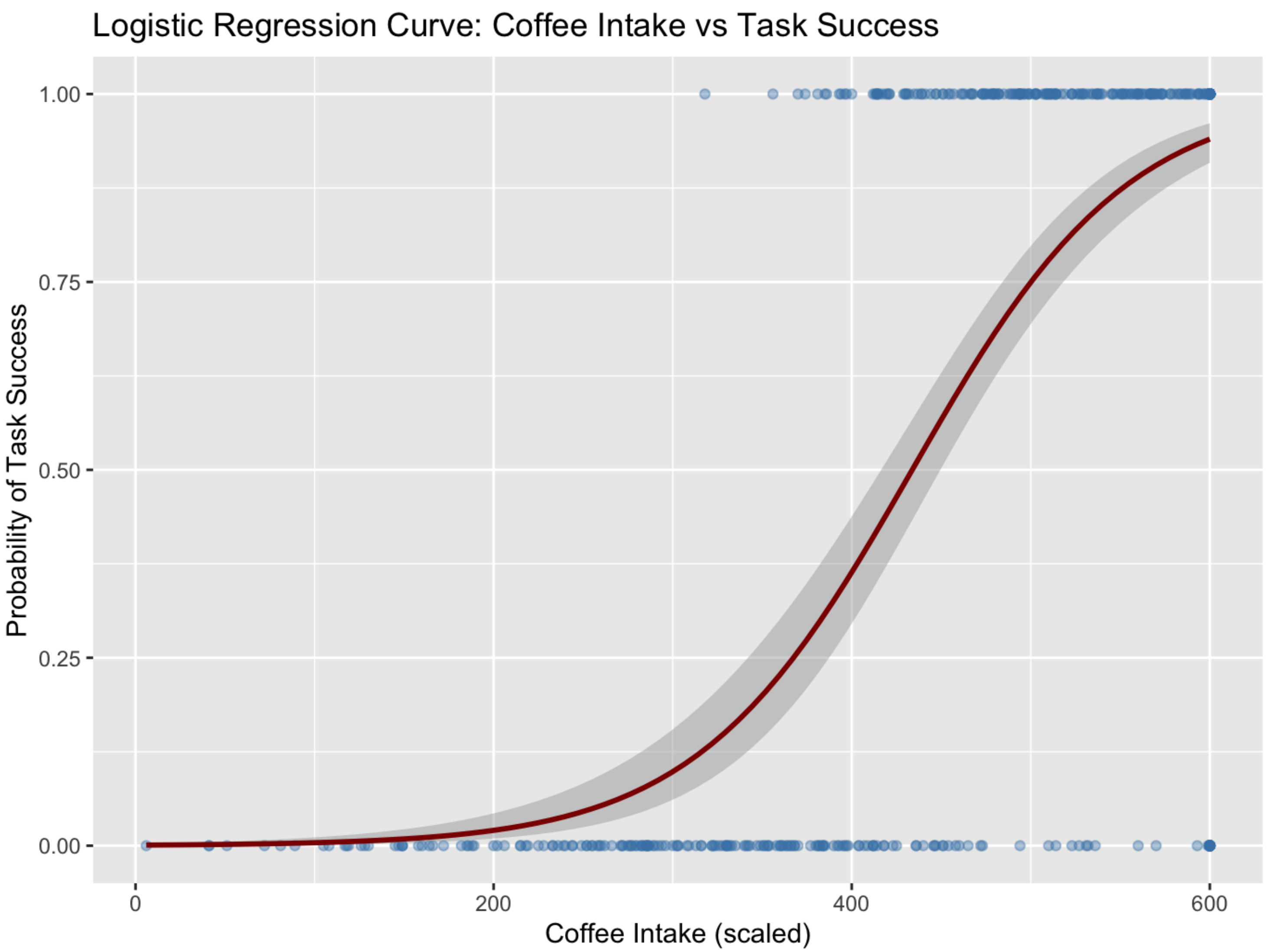
```
> #Odds ratio and confint
> exp(coef(log_model))
  (Intercept) coffee_intake_mg
 0.0007591783     1.0166973342
> exp(confint(log_model))
Waiting for profiling to be done...                      2.5 %     97.5 %
(Intercept)      0.000193198 0.00260499
coffee_intake_mg 1.014048574 1.01965048
>
> #labeling the data based on predictions that comes from logistic reg
> data$predicted_prob <- predict(log_model, type = "response")
> data$predicted_class <- ifelse(scaled_data$predicted_prob > 0.5, 1, 0)
> #confusion matrix
> conf_matrix <- table(Predicted = data$predicted_class, Actual = scaled_data$task_success)
> conf_matrix
         Actual
Predicted   0   1
        0 152  30
        1  45 273
>
> #accuracy of the model
> accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
> print(paste("Accuracy Ratio: ", round(accuracy * 100, 2), "%"))
[1] "Accuracy Ratio:  85 %"
>
> library(pROC)
> #AUC ROC
> roc_obj <- roc(scaled_data$task_success, scaled_data$predicted_prob)
Setting levels: control = 0, case = 1
Setting direction: controls < cases> auc_val <- auc(roc_obj); auc_val
Area under the curve: 0.8873
```

The odds ratio for `coffee_intake_mg` is 1.0167, indicating that each additional mg of coffee increases the odds of **task success** by **1.67%.**

The logistic regression model achieved an **accuracy of 85%**, correctly classifying 425 out of 500 cases.

Additionally, the **AUC value of 0.8873** indicates strong discriminatory power, confirming that coffee intake is a robust predictor of task success.

Logistic regression 📊



Conclusions

- We observed the **interactions** between the parameters and could have explored them even further.
- **Clustering** uncovered distinct work patterns characterized by variations of circumstances.
- **Outlier detection** identified extreme values except `bugs_reported`.
- **Correlation analysis** indicated that `task_success` is strongly associated with both caffeine intake and `hours_coding`. Also we can see when cognitive_load is high there is a strong reason: sleep_hours

We might say that these findings emphasize productivity is not solely about time investment, but about balancing focus, mental effort, and behavioral support like caffeine.