MA214 Applied Statistics - Project 1 Week 2

C4 Group 7

2026-02-11

Project 1 week2 worksheet

1: Importing and Inspecting the Data

Objective: Load your dataset and understand its structure.

Things to consider:

- What variables are available? Are there enough predictors for modeling?
- What are their data types?
- Are there missing values?
- Do any variables need cleaning or recoding?

```
library(ggplot2)
library(dplyr)
library(readr)
library(tidyr)

data <- read.csv("ai_dev_productivity.csv")
head(data)
```

```
##   hours_coding coffee_intake_mg distractions sleep_hours commits bugs_reported
## 1         5.99              600            1         5.8       2             1
## 2         4.72              568            2         6.9       5             3
## 3         6.30              560            1         8.9       2             0
## 4         8.05              600            7         6.3       9             5
## 5         4.53              421            6         6.9       4             0
## 6         4.53              429            1         7.1       5             0
##   ai_usage_hours cognitive_load task_success
## 1           0.71            5.4            1
## 2           1.75            4.7            1
## 3           2.27            2.2            1
## 4           1.40            5.9            0
## 5           1.26            6.3            1
## 6           3.06            3.9            1
```

```
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 ...
```

```
colSums(is.na(data))
```

```
##     hours_coding coffee_intake_mg     distractions      sleep_hours
##                0                0                0                0
##          commits    bugs_reported   ai_usage_hours   cognitive_load
##                0                0                0                0
##     task_success
##                0
```

```
summary(data)
```

```
##   hours_coding    coffee_intake_mg  distractions    sleep_hours
##  Min.   : 0.000   Min.   :  6.0    Min.   :0.000   Min.   : 3.000
##  1st Qu.: 3.600   1st Qu.:369.5    1st Qu.:2.000   1st Qu.: 6.100
##  Median : 5.030   Median :500.5    Median :3.000   Median : 6.950
##  Mean   : 5.016   Mean   :463.2    Mean   :2.976   Mean   : 6.976
##  3rd Qu.: 6.275   3rd Qu.:600.0    3rd Qu.:4.000   3rd Qu.: 7.900
##  Max.   :12.000   Max.   :600.0    Max.   :8.000   Max.   :10.000
##     commits       bugs_reported   ai_usage_hours   cognitive_load
##  Min.   : 0.000   Min.   :0.000   Min.   :0.0000   Min.   : 1.000
##  1st Qu.: 3.000   1st Qu.:0.000   1st Qu.:0.6975   1st Qu.: 3.175
##  Median : 5.000   Median :0.000   Median :1.2600   Median : 4.400
##  Mean   : 4.608   Mean   :0.858   Mean   :1.5109   Mean   : 4.498
##  3rd Qu.: 6.000   3rd Qu.:2.000   3rd Qu.:2.0700   3rd Qu.: 5.800
##  Max.   :13.000   Max.   :5.000   Max.   :6.3600   Max.   :10.000
##   task_success
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :1.000
##  Mean   :0.606
##  3rd Qu.:1.000
##  Max.   :1.000
```

Group Discussion: Note any variables that need recoding or cleaning.

- `task_success` is stored as 0/1 numeric but is really categorical, so it needs to be recoded as a factor.
- `coffee_intake_mg` appears to be capped at 600, which could affect analysis.
- There are no missing values in the dataset.
- All other variables are numeric and don't need cleaning.

2. Data Transformation

Objective: Prepare your data for analysis by performing necessary transformations.

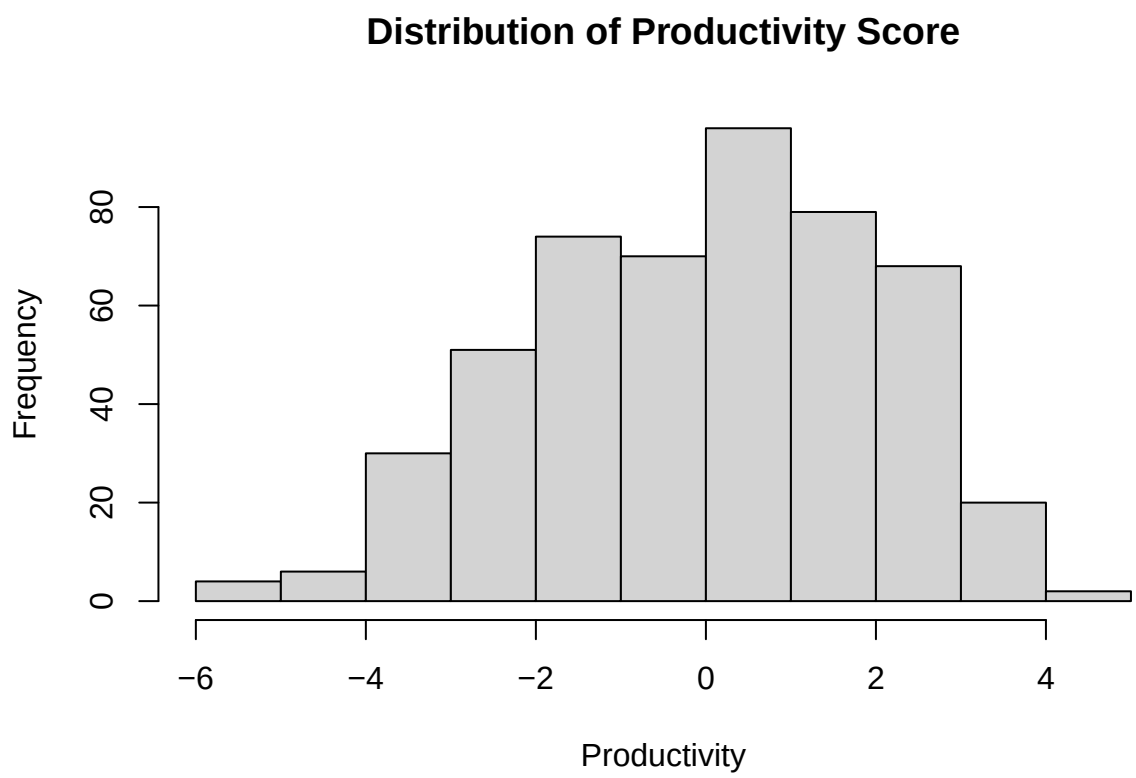
Consider:

- Do variables need unit changes or standardization?
- Would a log or other transformation improve visualization or interpretation?
- Do you need to create new variables for analysis?

```
# Recode task_success as a labeled factor
data$task_success <- factor(data$task_success, levels = c(0,
    1), labels = c("Failure", "Success"))

# Composite productivity score using z-scores of output
# variables hours_coding excluded here, used as a predictor
# instead
data <- data |>
    mutate(productivity = scale(commits) - scale(bugs_reported) +
        scale(as.numeric(task_success)))
data$productivity <- as.numeric(data$productivity)

summary(data$productivity)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -5.4218 -1.4251  0.2497  0.0000  1.5611  4.6859
```

```
hist(data$productivity, main = "Distribution of Productivity Score",
    xlab = "Productivity")
```



Group Discussion: Explain why each transformation was applied.

- We recoded `task_success` from 0/1 to "Failure"/"Success" so it is treated as a category, not a number.

- We created a composite productivity score by standardizing `commits`, `bugs_reported`, and `task_success` using z-scores. This puts them on the same scale so no single variable dominates. We subtracted `bugs_reported` because more bugs means lower productivity.
- We excluded `hours_coding` from the productivity score and kept it as a predictor, since more hours spent does not necessarily mean more productive.

3. Data Summarization

Objective: Summarize your dataset to understand patterns and distributions.

Tasks:

- Produce overall summaries using table, statistics
- Produce grouped summaries if applicable (e.g., by category)
- Produce meaningful summaries by plots

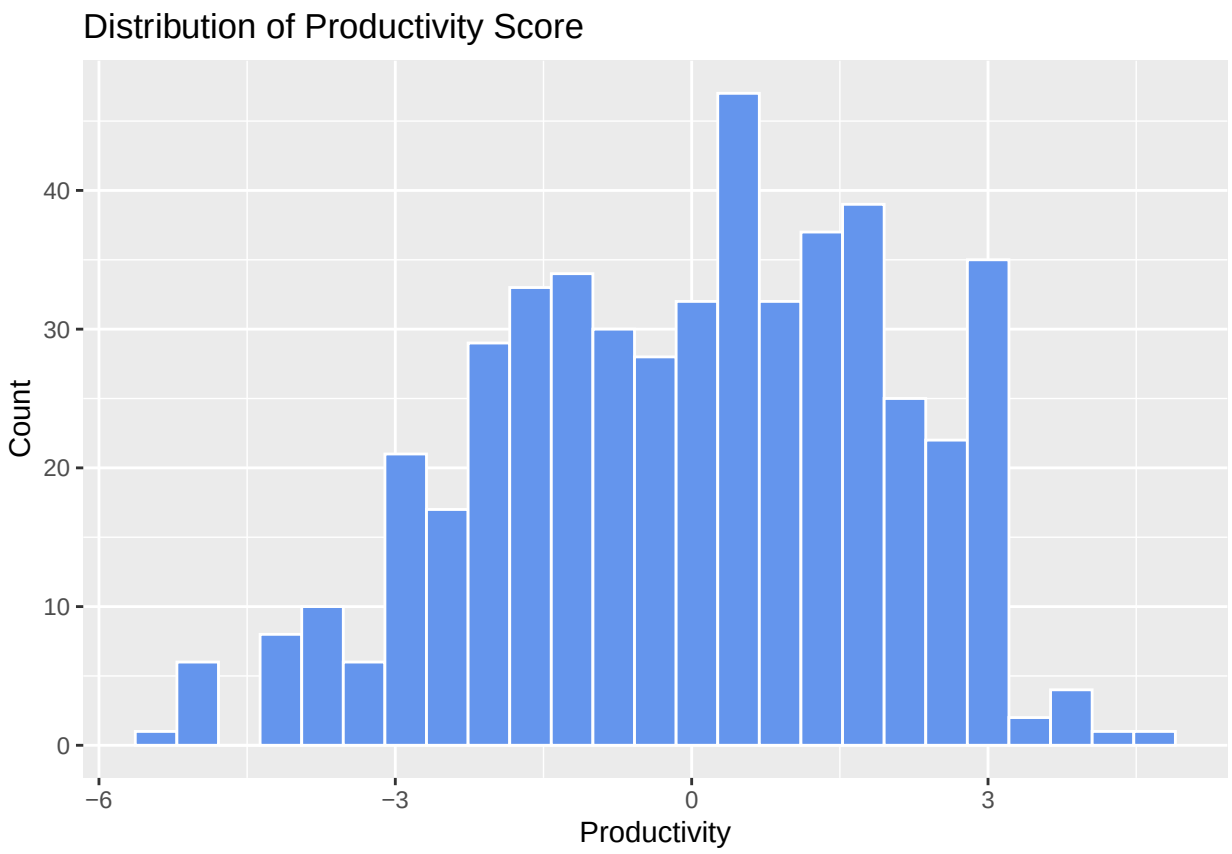
```
summary(data[, c("hours_coding", "coffee_intake_mg", "sleep_hours",
    "distractions", "ai_usage_hours", "cognitive_load", "productivity")])
```

```
##   hours_coding    coffee_intake_mg  sleep_hours      distractions
##  Min.   : 0.000   Min.   :  6.0    Min.   : 3.000   Min.   :0.000
##  1st Qu.: 3.600   1st Qu.:369.5    1st Qu.: 6.100   1st Qu.:2.000
##  Median : 5.030   Median :500.5    Median : 6.950   Median :3.000
##  Mean   : 5.016   Mean   :463.2    Mean   : 6.976   Mean   :2.976
##  3rd Qu.: 6.275   3rd Qu.:600.0    3rd Qu.: 7.900   3rd Qu.:4.000
##  Max.   :12.000   Max.   :600.0    Max.   :10.000   Max.   :8.000
##  ai_usage_hours   cognitive_load    productivity
##  Min.   :0.0000   Min.   : 1.000   Min.   :-5.4218
##  1st Qu.:0.6975   1st Qu.: 3.175   1st Qu.:-1.4251
##  Median :1.2600   Median : 4.400   Median : 0.2497
##  Mean   :1.5109   Mean   : 4.498   Mean   : 0.0000
##  3rd Qu.:2.0700   3rd Qu.: 5.800   3rd Qu.: 1.5611
##  Max.   :6.3600   Max.   :10.000   Max.   : 4.6859
```

```
# Compare mean predictors by task success
data |>
    group_by(task_success) |>
    summarise(mean_sleep = mean(sleep_hours), mean_coffee = mean(coffee_intake_mg),
        mean_distractions = mean(distractions), mean_ai_usage = mean(ai_usage_hours),
        mean_cognitive_load = mean(cognitive_load), mean_productivity = mean(productivity))
```

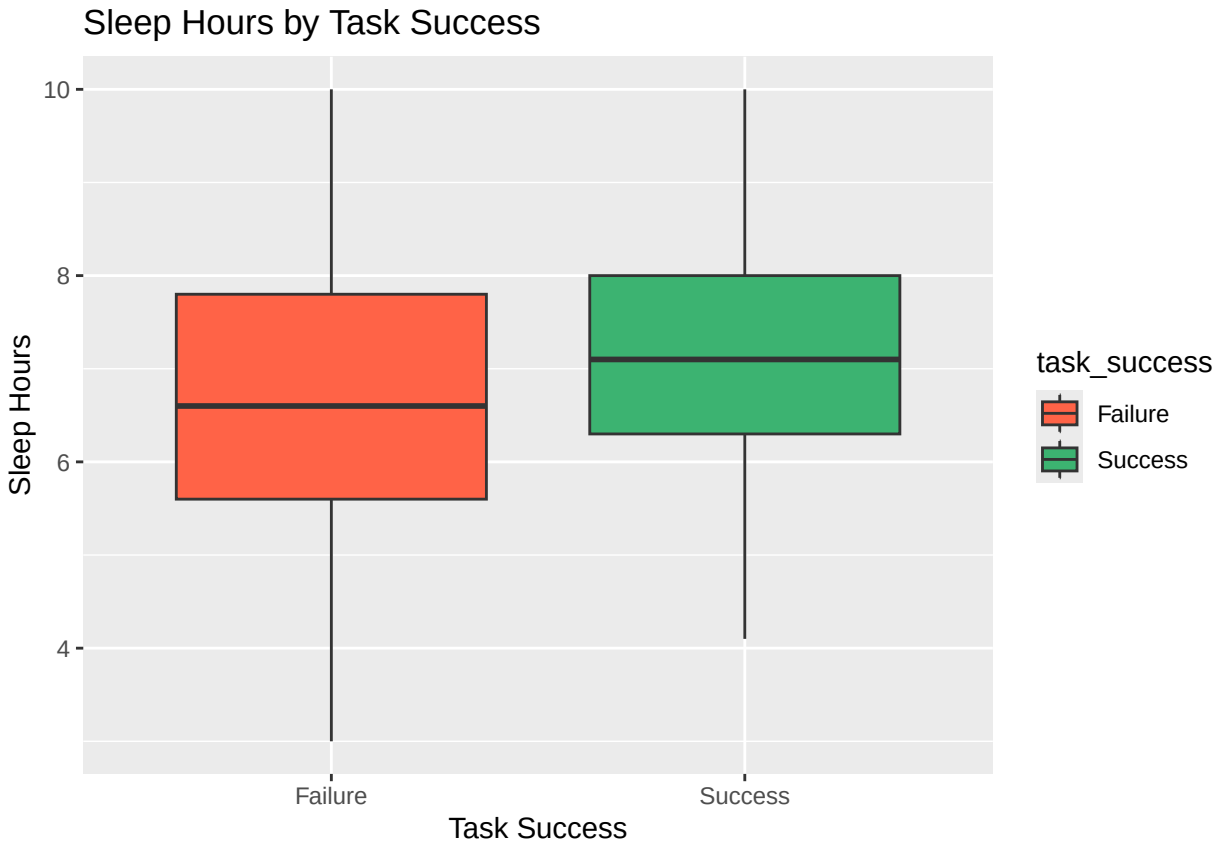
```
## # A tibble: 2 x 7
##   task_success mean_sleep mean_coffee mean_distractions mean_ai_usage
##   <fct>             <dbl>       <dbl>             <dbl>         <dbl>
## 1 Failure            6.64        341.              3.19          1.19
## 2 Success            7.19        543.              2.83          1.72
## # i 2 more variables: mean_cognitive_load <dbl>, mean_productivity <dbl>
```

```
ggplot(data, aes(x = productivity)) + geom_histogram(bins = 25,
    fill = "cornflowerblue", color = "white") + labs(title = "Distribution of Productivity Score",
    x = "Productivity", y = "Count")
```



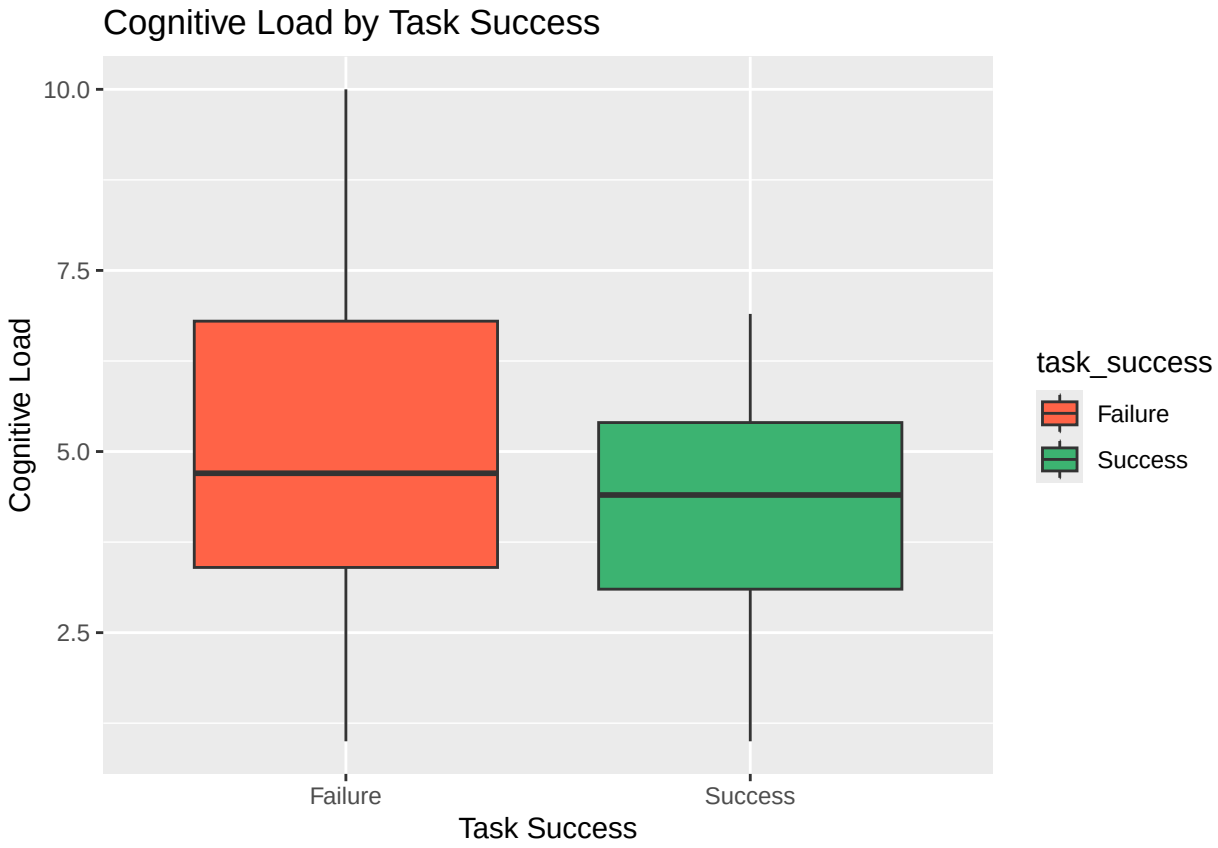
The productivity score is roughly bell-shaped and centered around 0, which is expected since it is built from z-scores.

```
ggplot(data, aes(x = task_success, y = sleep_hours, fill = task_success)) +
    geom_boxplot() + scale_fill_manual(values = c(Failure = "tomato",
    Success = "mediumseagreen")) + labs(title = "Sleep Hours by Task Success",
    x = "Task Success", y = "Sleep Hours")
```



Successful tasks tend to have higher sleep hours. The median sleep for successes is noticeably higher than for failures.

```
ggplot(data, aes(x = task_success, y = cognitive_load, fill = task_success)) +
    geom_boxplot() + scale_fill_manual(values = c(Failure = "tomato",
    Success = "mediumseagreen")) + labs(title = "Cognitive Load by Task Success",
    x = "Task Success", y = "Cognitive Load")
```



Failed tasks have much higher cognitive load on average. This suggests that mental strain hurts task outcomes.

Group Discussion: Comment on interesting patterns observed in summaries and plots.

- Successful tasks are associated with more sleep and lower cognitive load on average.
- The productivity score is roughly centered around 0, with a spread from about -3 to 5.
- The boxplots show a clear difference in cognitive load between success and failure groups, with failures having noticeably higher cognitive load.

4. Assumption Checking

Objective: Evaluate key modeling assumptions using visualizations and summary statistics.

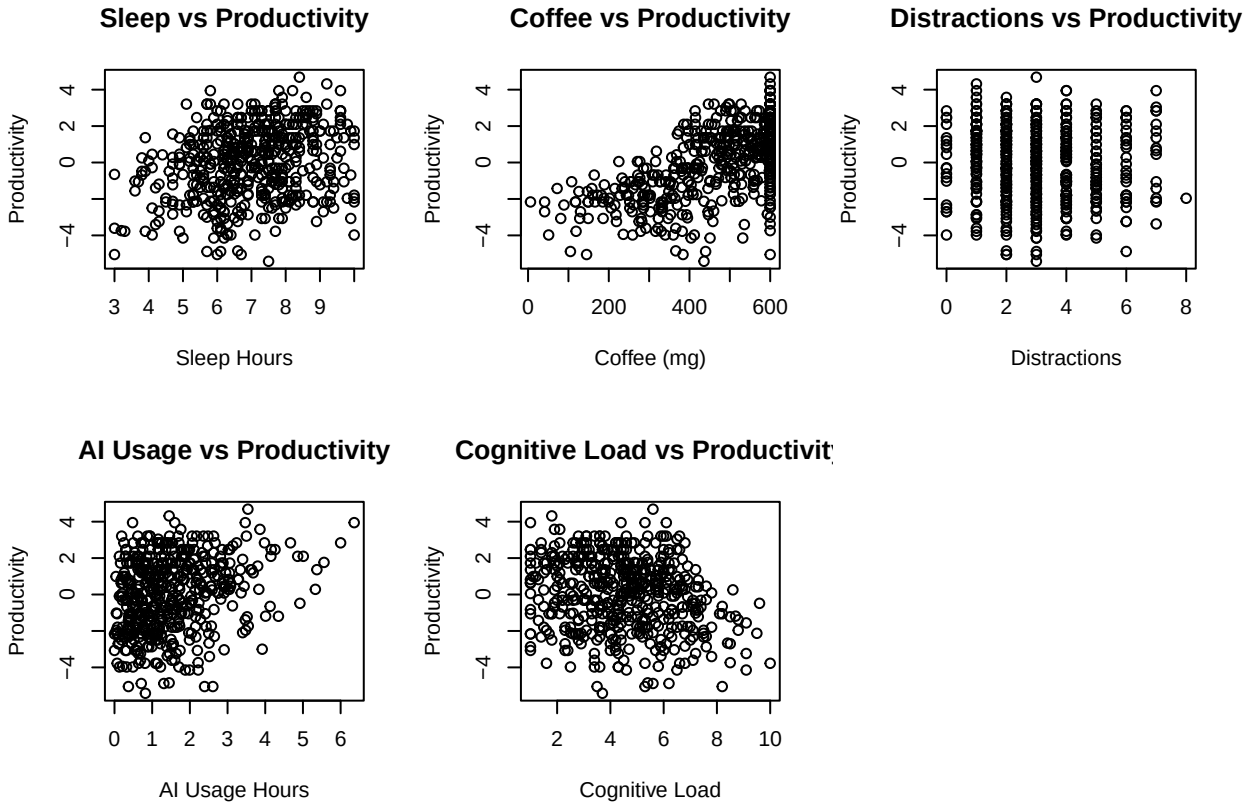
Check for:

- Linearity
- Normality
- Independence

Tools: Histograms, boxplots, scatterplots

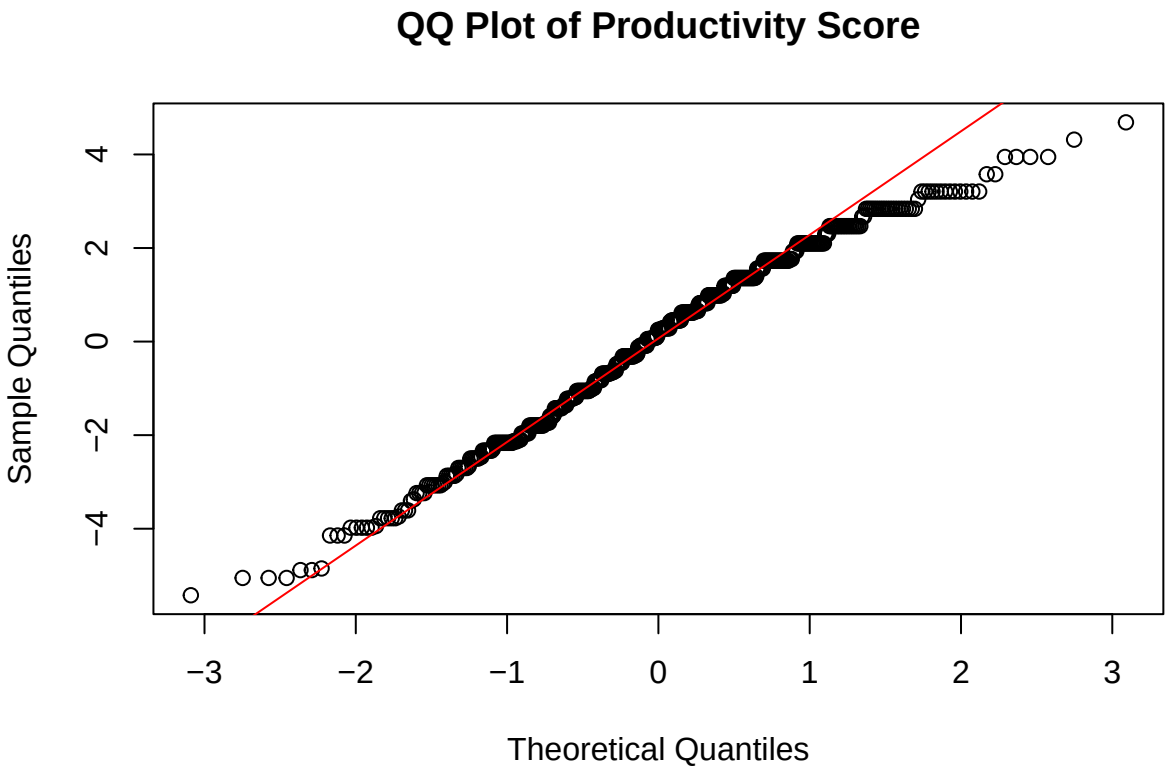
```r
# Linearity check
par(mfrow = c(2, 3))
plot(data$sleep_hours, data$productivity, main = "Sleep vs Productivity",
    xlab = "Sleep Hours", ylab = "Productivity")
plot(data$coffee_intake_mg, data$productivity, main = "Coffee vs Productivity",
    xlab = "Coffee (mg)", ylab = "Productivity")
```

```r
plot(data$distractions, data$productivity, main = "Distractions vs Productivity",
    xlab = "Distractions", ylab = "Productivity")
plot(data$ai_usage_hours, data$productivity, main = "AI Usage vs Productivity",
    xlab = "AI Usage Hours", ylab = "Productivity")
plot(data$cognitive_load, data$productivity, main = "Cognitive Load vs Productivity",
    xlab = "Cognitive Load", ylab = "Productivity")
par(mfrow = c(1, 1))
```



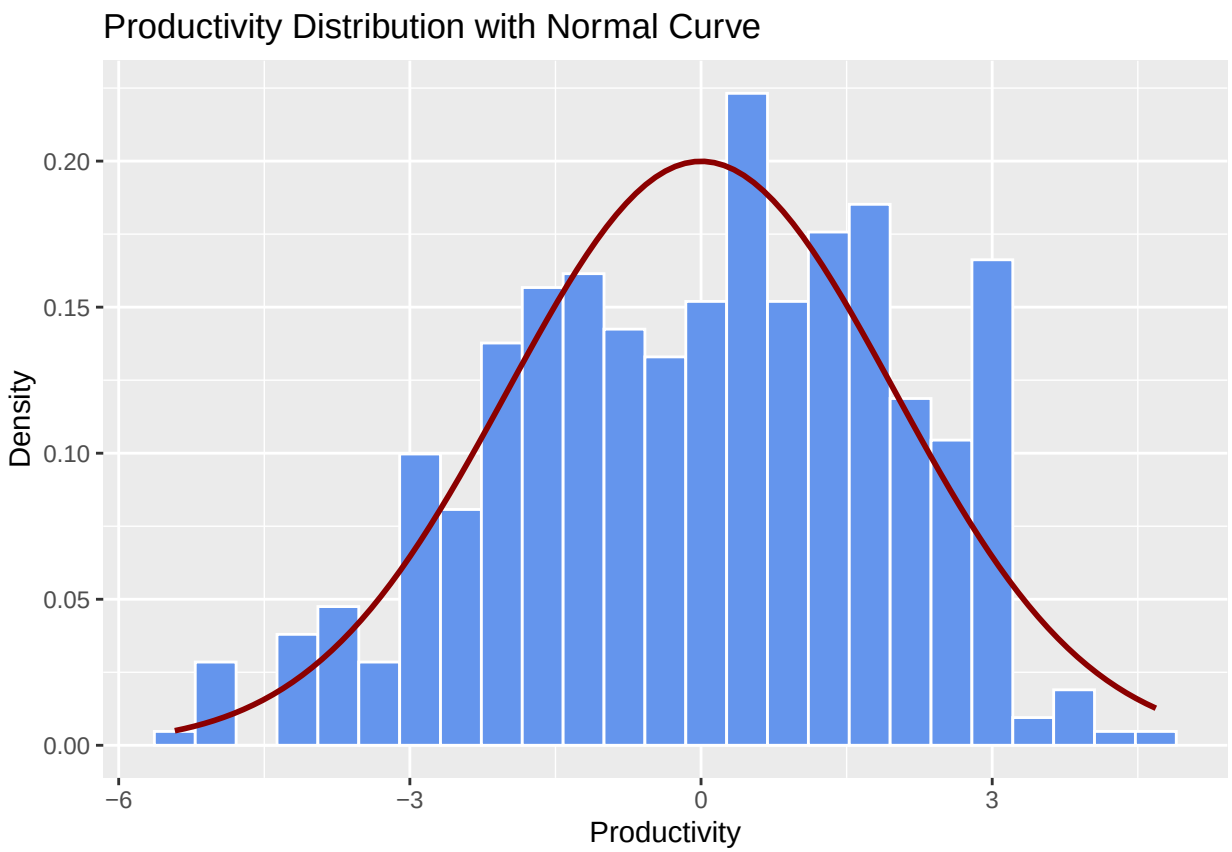
The scatterplots show roughly linear trends for most predictors. No strong curves are visible, so the linearity assumption looks reasonable.

```r
# Normality check
qqnorm(data$productivity, main = "QQ Plot of Productivity Score")
qqline(data$productivity, col = "red")
```



If the points fall along the red line, the data is approximately normal. Deviations at the tails suggest skewness or outliers.

```
ggplot(data, aes(x = productivity)) + geom_histogram(aes(y = after_stat(density)),
    bins = 25, fill = "cornflowerblue", color = "white") + stat_function(fun = dnorm,
    args = list(mean = mean(data$productivity), sd = sd(data$productivity)),
    color = "darkred", linewidth = 1) + labs(title = "Productivity Distribution with Normal Curve",
    x = "Productivity", y = "Density")
```



The histogram follows the red normal curve fairly well. Productivity appears approximately normal, so the normality assumption holds.

Group Discussion: Comment on whether assumptions are met and any potential issues.

- Linearity: The scatterplots show roughly linear relationships between most predictors and productivity. No strong curves are visible.
- Normality: The histogram of productivity follows the normal curve fairly well, so this assumption looks reasonable.
- Independence: Each row represents a separate observation, so independence is a safe assumption.
- One potential issue is that `coffee_intake_mg` is capped at 600, which creates a cluster of points at the top of its range.

5. Exploring Relationships Between Variables

Objective: Investigate associations and patterns between variables.

Questions:

Which variables appear associated?

Are relationships linear or nonlinear?

Are there outliers or unusual patterns?

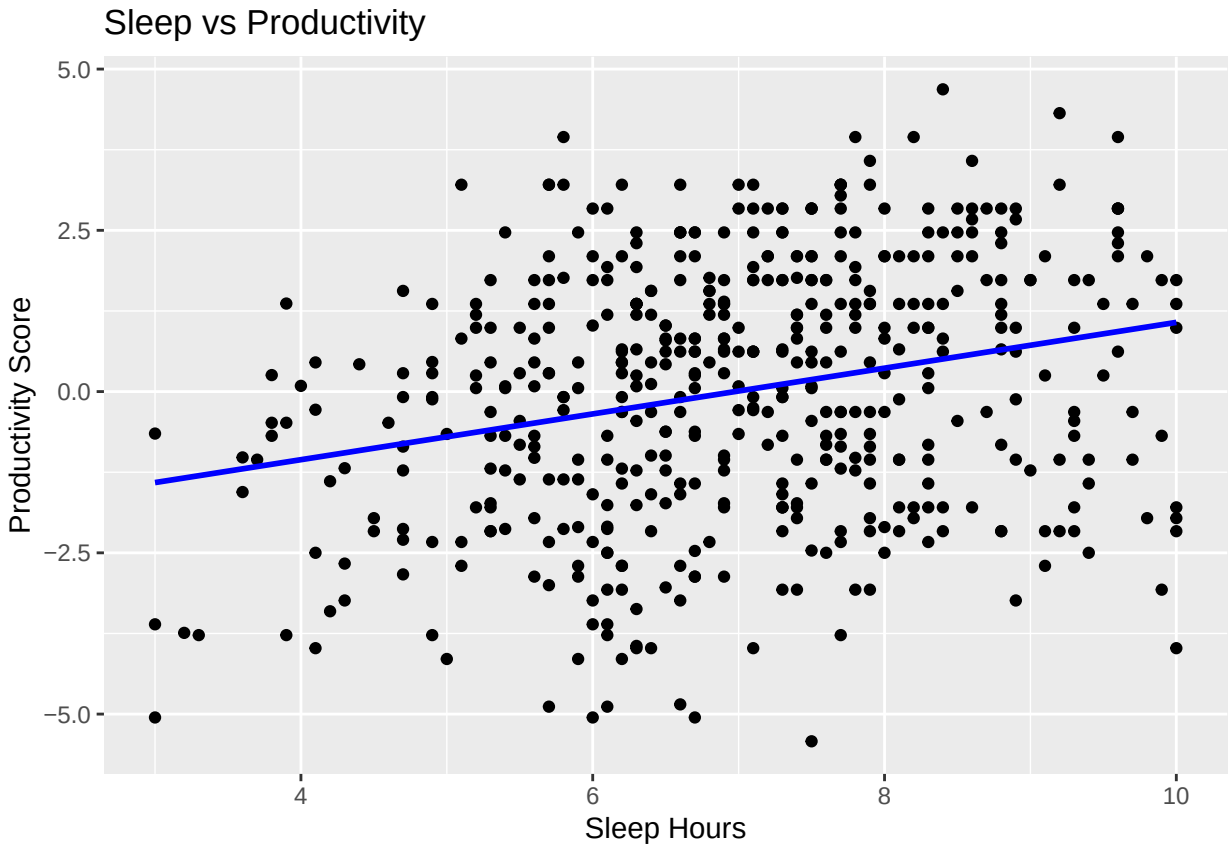
```
# Correlation Summary
predictors <- data |>
    select(hours_coding, coffee_intake_mg, sleep_hours, distractions,
        ai_usage_hours, cognitive_load, productivity)
round(cor(predictors, use = "complete.obs"), 3)
```

```
##                 hours_coding coffee_intake_mg sleep_hours distractions
## hours_coding           1.000           0.890      -0.025       -0.010
## coffee_intake_mg       0.890           1.000      -0.039       -0.036
## sleep_hours           -0.025          -0.039       1.000        0.041
## distractions          -0.010          -0.036       0.041        1.000
## ai_usage_hours         0.572           0.465      -0.084        0.029
## cognitive_load         0.051           0.037      -0.734        0.400
## productivity           0.605           0.606       0.259       -0.069
##                 ai_usage_hours cognitive_load productivity
## hours_coding             0.572          0.051        0.605
## coffee_intake_mg         0.465          0.037        0.606
## sleep_hours             -0.084         -0.734        0.259
## distractions             0.029          0.400       -0.069
## ai_usage_hours           1.000          0.120        0.249
## cognitive_load           0.120          1.000       -0.204
## productivity             0.249         -0.204        1.000
```

```r
pred_names <- c("hours_coding", "coffee_intake_mg", "sleep_hours",
    "distractions", "ai_usage_hours", "cognitive_load")
r_values <- sapply(pred_names, function(x) cor(data[[x]], data$productivity))
data.frame(predictor = pred_names, r = round(r_values, 3), R_squared = round(r_values^2,
    3))
```

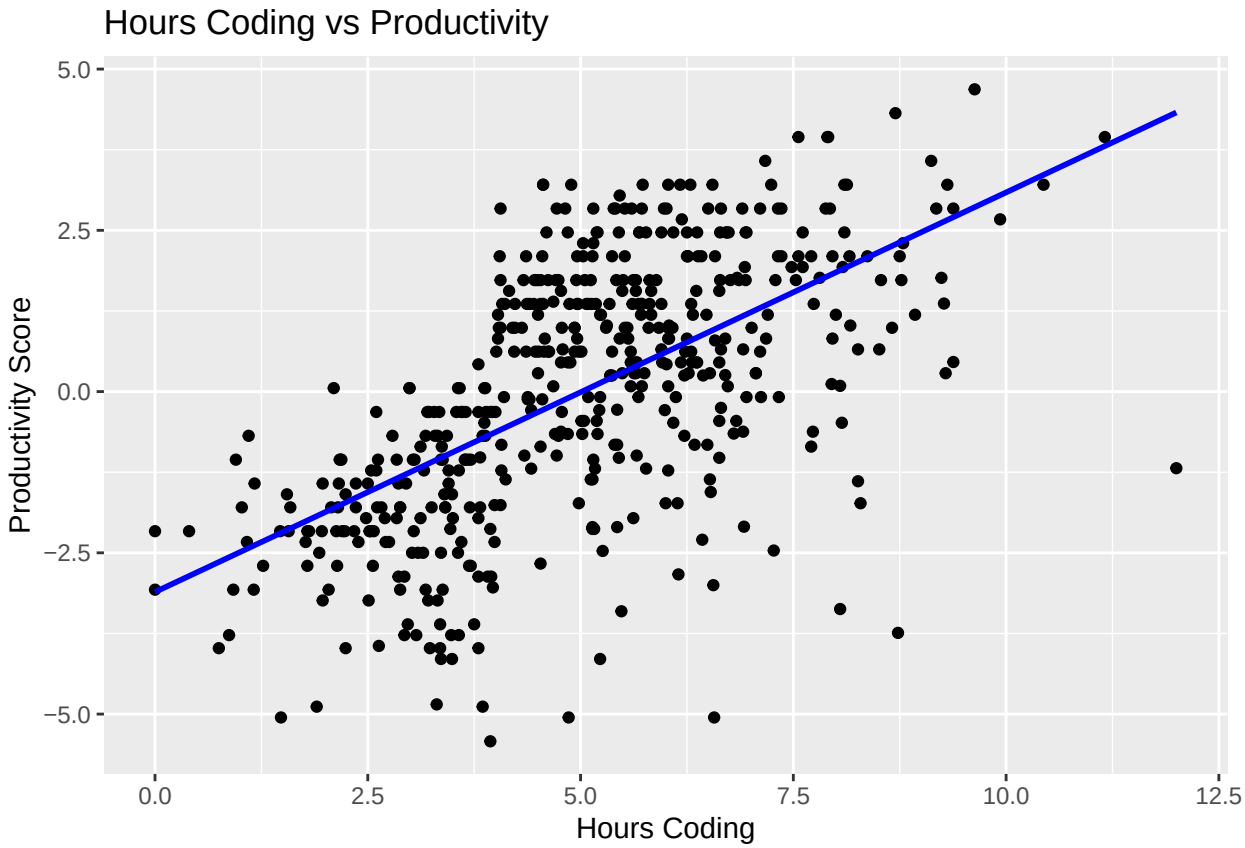
```
##                          predictor      r R_squared
## hours_coding         hours_coding  0.605     0.366
## coffee_intake_mg coffee_intake_mg  0.606     0.368
## sleep_hours           sleep_hours  0.259     0.067
## distractions         distractions -0.069     0.005
## ai_usage_hours     ai_usage_hours  0.249     0.062
## cognitive_load     cognitive_load -0.204     0.042
```

```r
# Predictor vs Productivity Scatterplots
ggplot(data, aes(x = sleep_hours, y = productivity)) + geom_point() +
    geom_smooth(method = "lm", se = FALSE, color = "blue") +
    labs(title = "Sleep vs Productivity", x = "Sleep Hours",
        y = "Productivity Score")
```



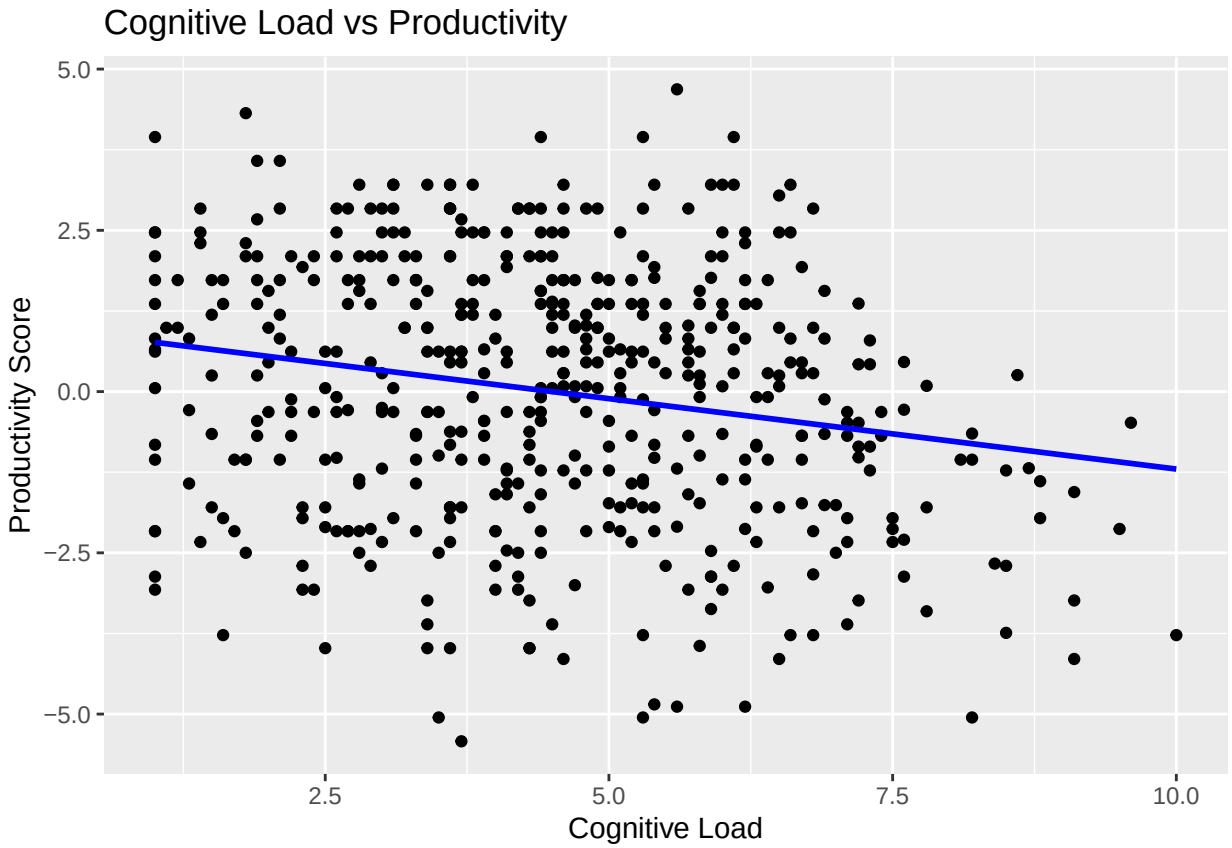
Positive correlation. More sleep is associated with higher productivity.

```
ggplot(data, aes(x = hours_coding, y = productivity)) + geom_point() +
    geom_smooth(method = "lm", se = FALSE, color = "blue") +
    labs(title = "Hours Coding vs Productivity", x = "Hours Coding",
        y = "Productivity Score")
```



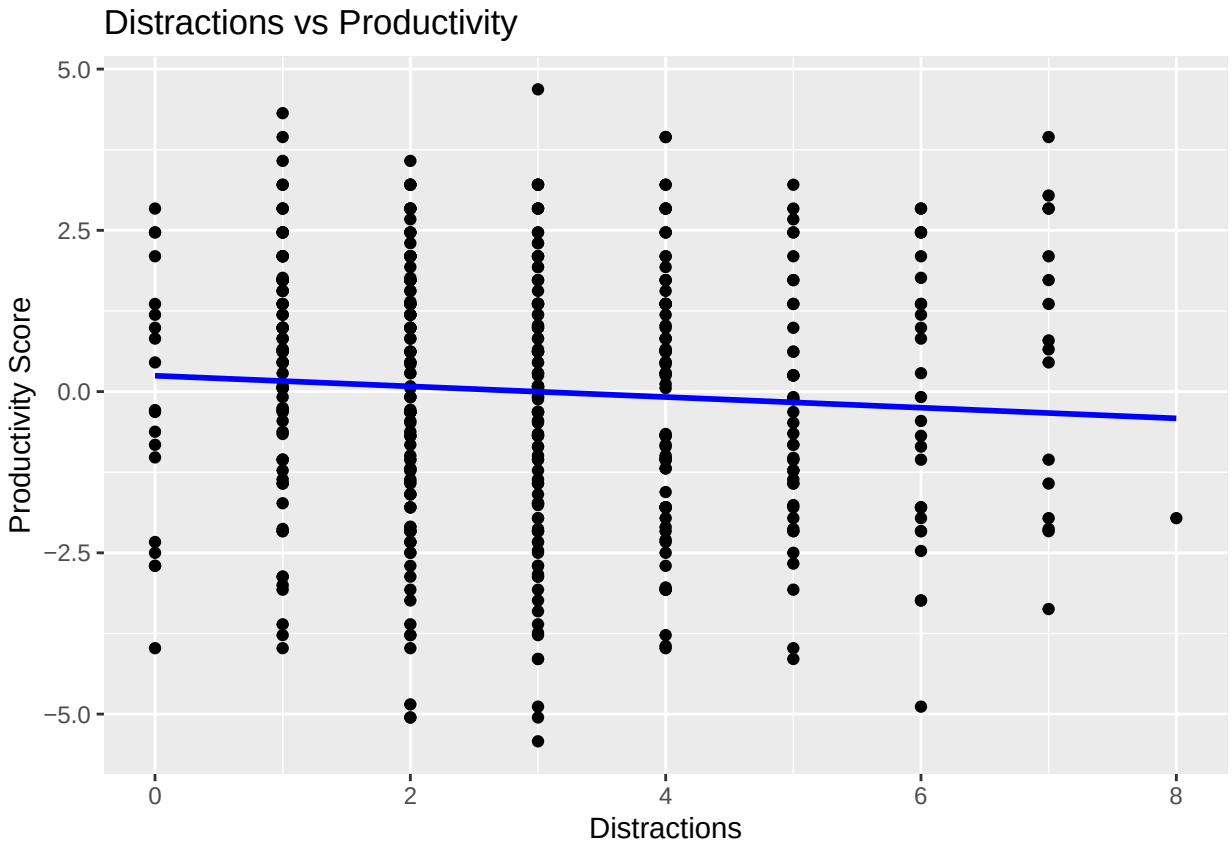
Positive correlation. Developers who code more hours tend to have higher productivity scores.

```
ggplot(data, aes(x = cognitive_load, y = productivity)) + geom_point() +
    geom_smooth(method = "lm", se = FALSE, color = "blue") +
    labs(title = "Cognitive Load vs Productivity", x = "Cognitive Load",
        y = "Productivity Score")
```



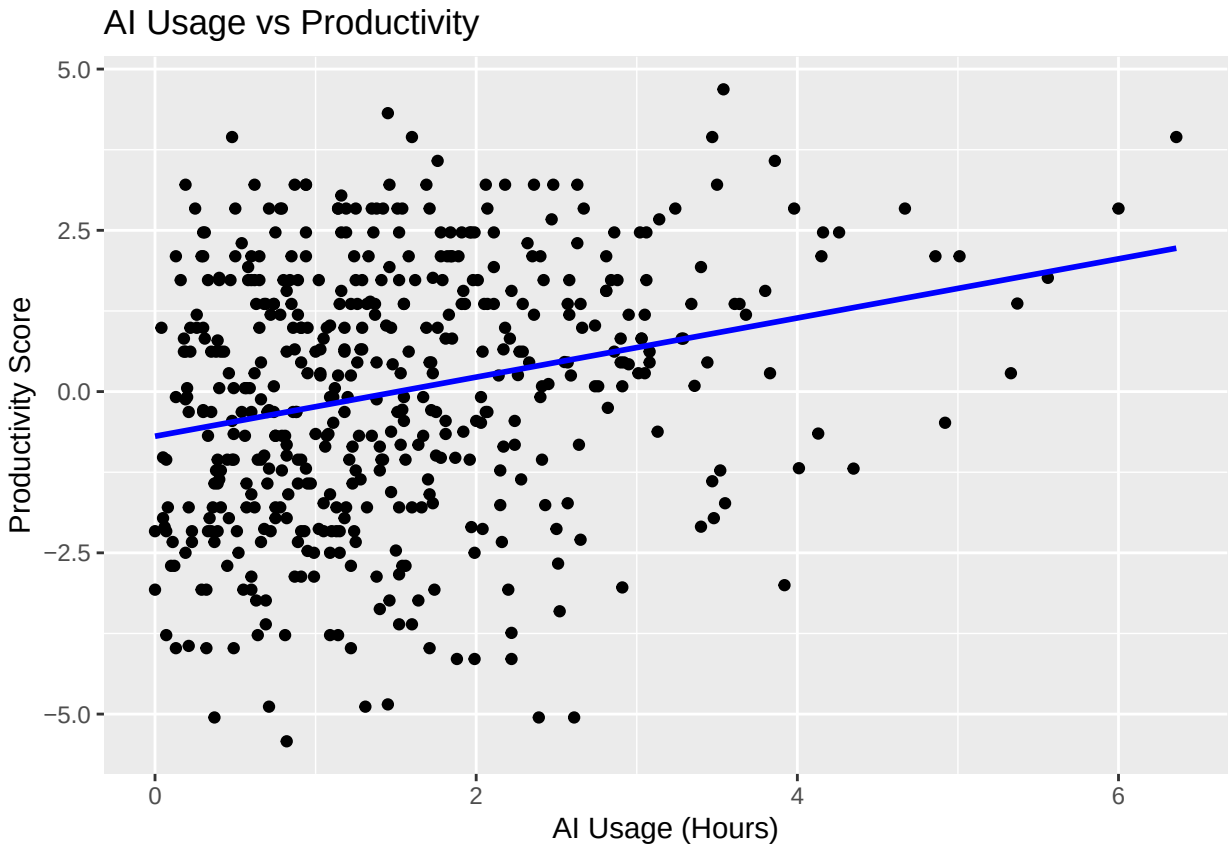
Negative correlation. Higher cognitive load is linked to lower productivity.

```
ggplot(data, aes(x = distractions, y = productivity)) + geom_point() +
    geom_smooth(method = "lm", se = FALSE, color = "blue") +
    labs(title = "Distractions vs Productivity", x = "Distractions",
        y = "Productivity Score")
```



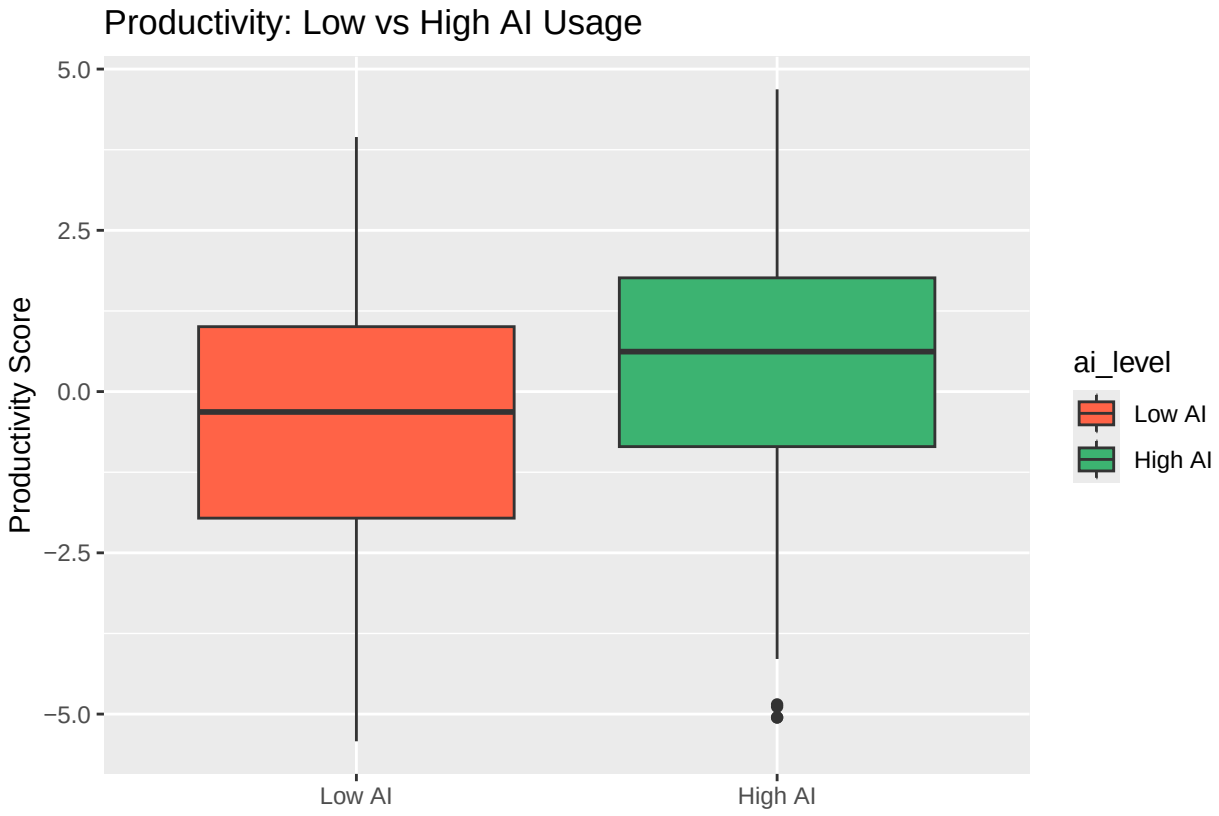
Weak negative correlation. More distractions tend to slightly lower productivity, but the relationship is not very strong.

```
# AI Usage Deep Dive
ggplot(data, aes(x = ai_usage_hours, y = productivity)) + geom_point() +
    geom_smooth(method = "lm", se = FALSE, color = "blue") +
    labs(title = "AI Usage vs Productivity", x = "AI Usage (Hours)",
        y = "Productivity Score")
```



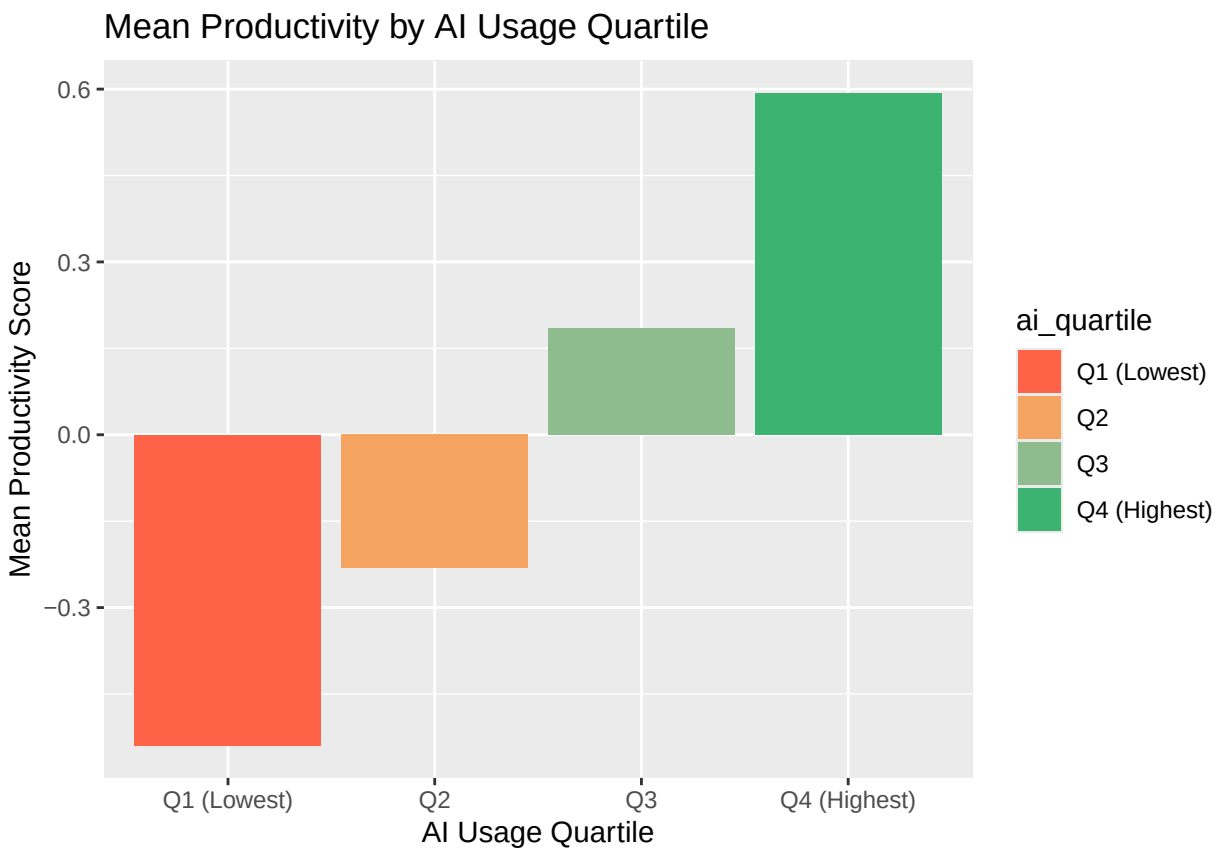
Moderate positive correlation. The trend line slopes upward, but the spread of points is wide.

```
data$ai_level <- factor(ifelse(data$ai_usage_hours <= median(data$ai_usage_hours),
    "Low AI", "High AI"), levels = c("Low AI", "High AI"))

ggplot(data, aes(x = ai_level, y = productivity, fill = ai_level)) +
    geom_boxplot() + scale_fill_manual(values = c(`Low AI` = "tomato",
    `High AI` = "mediumseagreen")) + labs(title = "Productivity: Low vs High AI Usage",
    x = "", y = "Productivity Score")
```



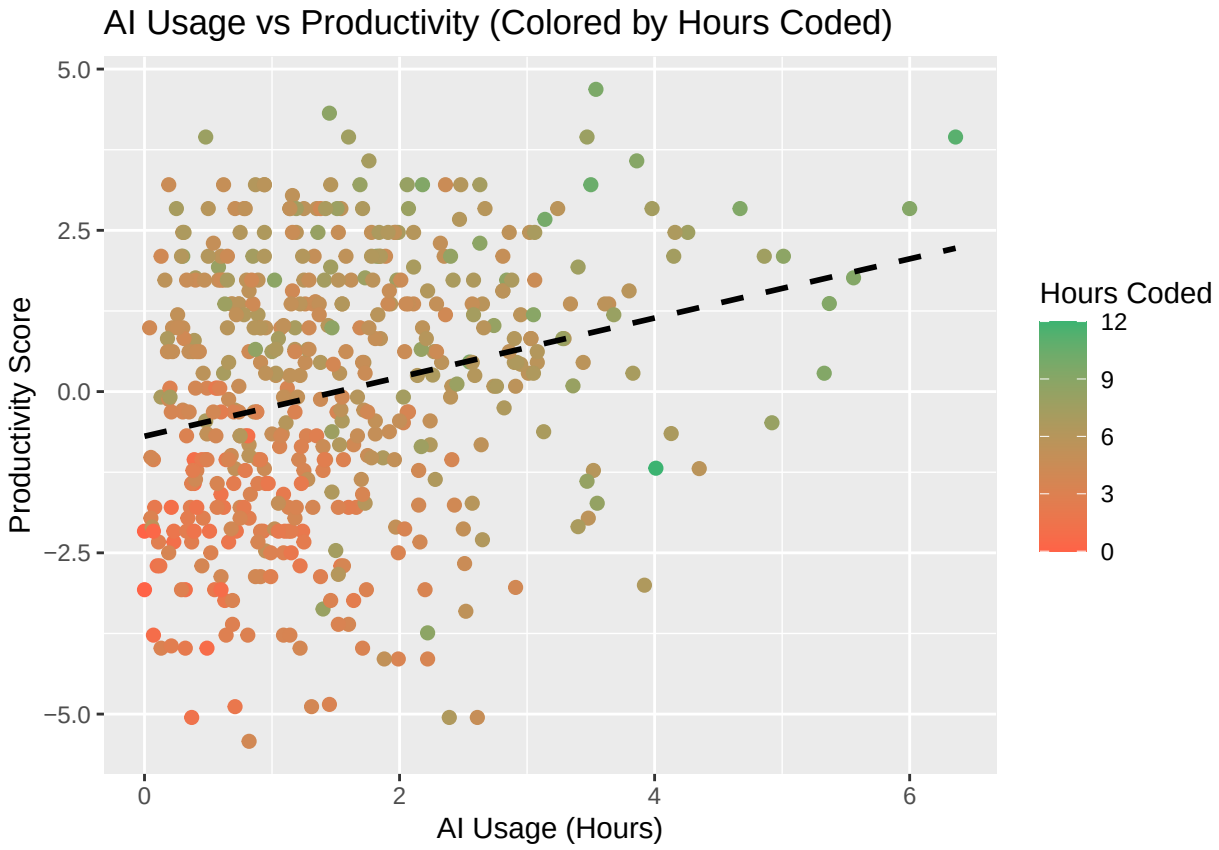
High AI users have a clearly higher median productivity than low AI users. The difference between the two groups is visible.

```
# Mean productivity by AI usage quartile
data$ai_quartile <- cut(data$ai_usage_hours, breaks = quantile(data$ai_usage_hours,
    probs = c(0, 0.25, 0.5, 0.75, 1)), include.lowest = TRUE,
    labels = c("Q1 (Lowest)", "Q2", "Q3", "Q4 (Highest)"))

ai_summary <- data |>
    group_by(ai_quartile) |>
    summarise(mean_productivity = mean(productivity))

ggplot(ai_summary, aes(x = ai_quartile, y = mean_productivity,
    fill = ai_quartile)) + geom_col() + scale_fill_manual(values = c(`Q1 (Lowest)` = "tomato",
    Q2 = "sandybrown", Q3 = "darkseagreen", `Q4 (Highest)` = "mediumseagreen")) +
    labs(title = "Mean Productivity by AI Usage Quartile", x = "AI Usage Quartile",
        y = "Mean Productivity Score")
```



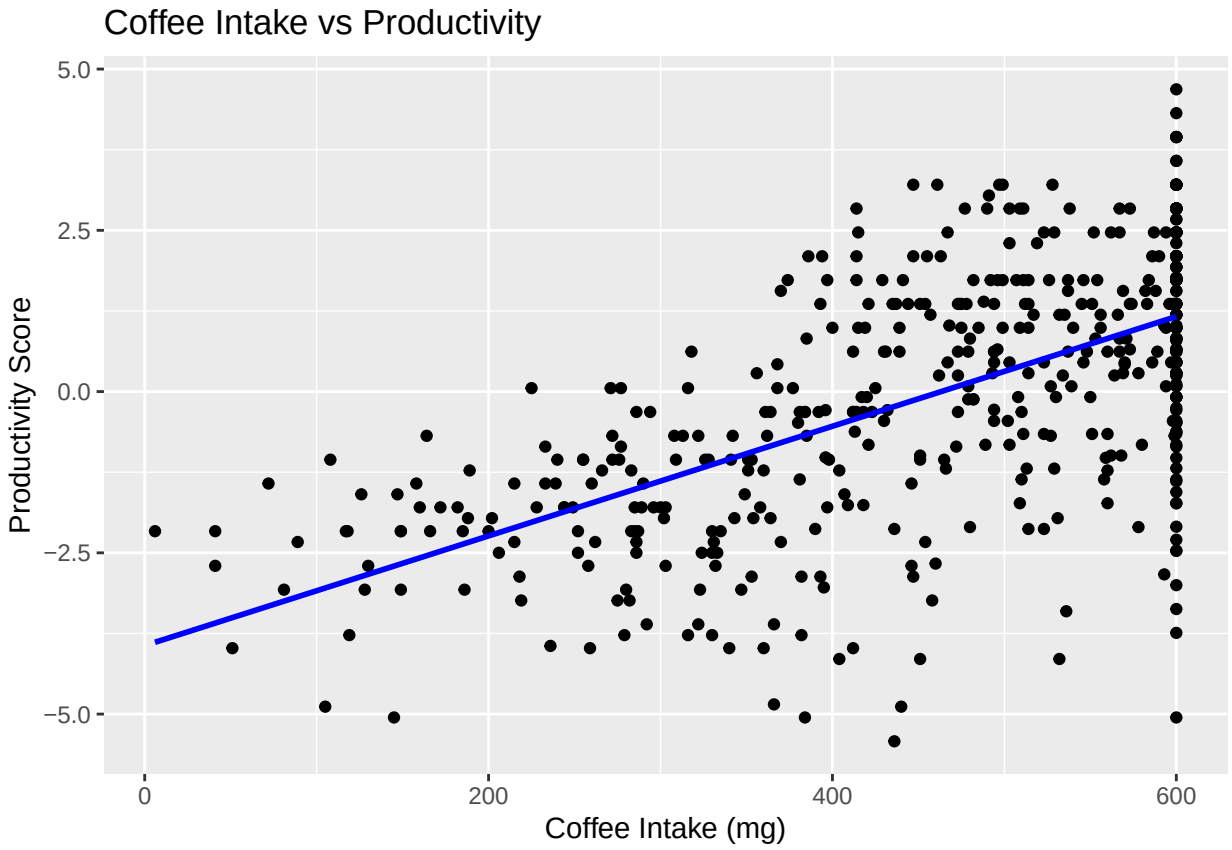
Mean productivity increases steadily from Q1 to Q4. This shows a clear trend that more AI usage is associated with better output.

```
# AI usage vs productivity, colored by hours coding
ggplot(data, aes(x = ai_usage_hours, y = productivity, color = hours_coding)) +
    geom_point(size = 2) + scale_color_gradient(low = "tomato",
    high = "mediumseagreen", name = "Hours Coded") + geom_smooth(method = "lm",
    se = FALSE, color = "black", linetype = "dashed") + labs(title = "AI Usage vs Productivity (Colored by
    x = "AI Usage (Hours)", y = "Productivity Score")
```



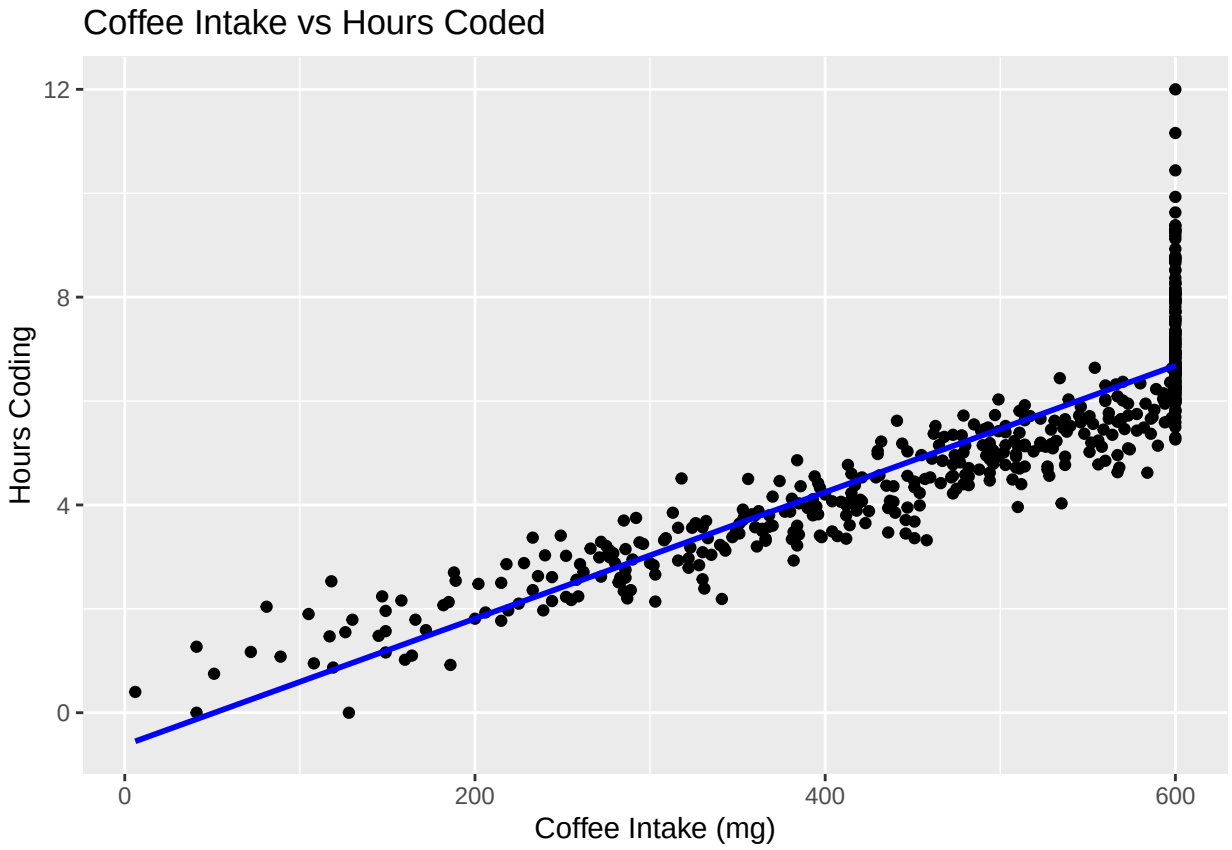
Green points (more hours coded) tend to cluster in the upper right, showing that developers who both use AI more and code more hours have the highest productivity.

```
# Coffee Intake Relationships
ggplot(data, aes(x = coffee_intake_mg, y = productivity)) + geom_point() +
    geom_smooth(method = "lm", se = FALSE, color = "blue") +
    labs(title = "Coffee Intake vs Productivity", x = "Coffee Intake (mg)",
        y = "Productivity Score")
```



Moderate positive correlation. Higher coffee intake is associated with higher productivity, though the cap at 600mg creates a cluster on the right side.

```
ggplot(data, aes(x = coffee_intake_mg, y = hours_coding)) + geom_point() +
    geom_smooth(method = "lm", se = FALSE, color = "blue") +
    labs(title = "Coffee Intake vs Hours Coded", x = "Coffee Intake (mg)",
        y = "Hours Coding")
```



Strong positive correlation. Developers who consume more coffee tend to code for more hours.

Group Discussion: Summarize the key relationships and any patterns found.

- Sleep hours and hours coding have the strongest positive correlations with productivity.
- Cognitive load has a negative relationship with productivity, meaning higher mental strain is linked to worse output.
- AI usage shows a clear positive effect on productivity, especially visible in the quartile bar chart where mean productivity increases steadily from Q1 to Q4.
- Coffee intake is positively correlated with hours coding, suggesting developers who drink more coffee tend to code longer.

Submit this document at the end of today's lab. If you downloaded any CSV files, please include them with your submission. **Each group should submit one set of files (Rmd, PDF, and any CSV files) to Blackboard.**

To convert this R Markdown document to PDF: In RStudio, click the **Knit** button at the top of the editor and select **Knit to PDF**. Make sure you have LaTeX installed (e.g., TinyTeX) to enable PDF generation.

Name	BUID	Present/Absent
Yunhao Zhou	U18926707	Present
Hibak Hussen	U15515562	Present
Muze Ren	U21890514	Present
Ian Sabia	U33871576	Present