

Dataset 3: The Concrete Dataset

Optimizing Concrete Strength: An Dashboard Using R & Shiny

Tejashwini Saravanan

ISM6358 Decision Making with Business
Intelligence and Analytics

Summer 2025

Project Objective

- Understand how material composition impacts concrete strength
- Build predictive analytics using R
- Design a decision support dashboard
- Recommend optimal material mixes

Dataset Overview

- Given Source: UCI Machine Learning Repository
- 1,030 samples of concrete mixtures
- Inputs (8): Cement, Slag, Fly Ash, Water, Superplasticizer, Coarse Agg, Fine Agg, Age
- Target Variable: Concrete Compressive Strength (MPa)

Load & Explore the Data in RStudio

Goal: Load the dataset and inspect structure and summary statistics

- **Downloaded Excel file → Converted to CSV**
- **Imported to R using read.csv()**
- **Previewed using head() and str()**
- **Age and Strength converted to numeric types**

```
8 library(dplyr)
9
10 concrete <- read.csv("C:/Users/Tejashwini Saravanan/Copy of Concrete_Data.csv")
11
12 str(concrete)
13
14 colSums(is.na(concrete))
15
16
17 head(concrete)
18
```

OUTPUT SUMMARY FROM THE CODE:

1030 ROWS, 9 VARIABLES

NO MISSING VALUES

COMPRESSIVE STRENGTH IS THE TARGET VARIABLE

Clean & Prepare the Data

GOAL: CLEAN COLUMN NAMES AND PREPARE DATASET FOR EDA

- **Code used:**

```
names(concrete) <- c("Cement", "BlastFurnaceSlag", "FlyAsh", "Water",  
                    "Superplasticizer", "CoarseAggregate", "FineAggregate",  
                    "Age", "CompressiveStrength")
```

```
names(concrete)
```

```
[1] "Cement"           "BlastFurnaceSlag"  "FlyAsh"           "Water"  
[5] "Superplasticizer" "CoarseAggregate"  "FineAggregate"    "Age"  
[9] "CompressiveStrength"
```

- **Check for 0 values in Superplasticizer:**

```
sum(concrete$Superplasticizer == 0)
```

```
> sum(concrete$Superplasticizer == 0)  
[1] 379
```

379 ROWS HAVE 0 SUPERPLASTICIZER : NOTE FOR MODELING.



Objective:

- Ensure column names are clean and readable.
- Identify anomalies or special cases in the data.

Actions Taken:

- Cleaned column names using names() function.
- Verified correct names like Cement, Water, Age, etc.
- Checked for 0 values in Superplasticizer:
- Found 379 rows with 0 → Consider during modeling (could indicate absence)

Output:

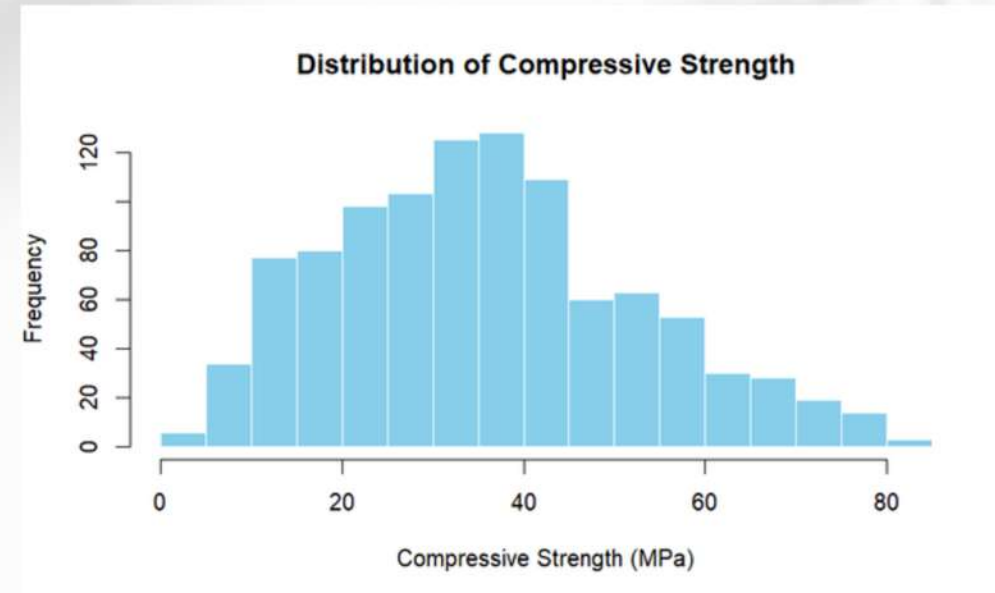
- Data ready for visualization and modeling
- No NAs, minimal cleaning needed

Exploratory Data Analysis (EDA)

GOAL: VISUALIZE DISTRIBUTIONS, RELATIONSHIPS, AND CORRELATIONS

Histogram: Distribution of Compressive Strength

```
hist(concrete$CompressiveStrength,  
     main = "Distribution of Compressive Strength",  
     xlab = "Compressive Strength (MPa)",  
     col = "skyblue", border = "white")
```

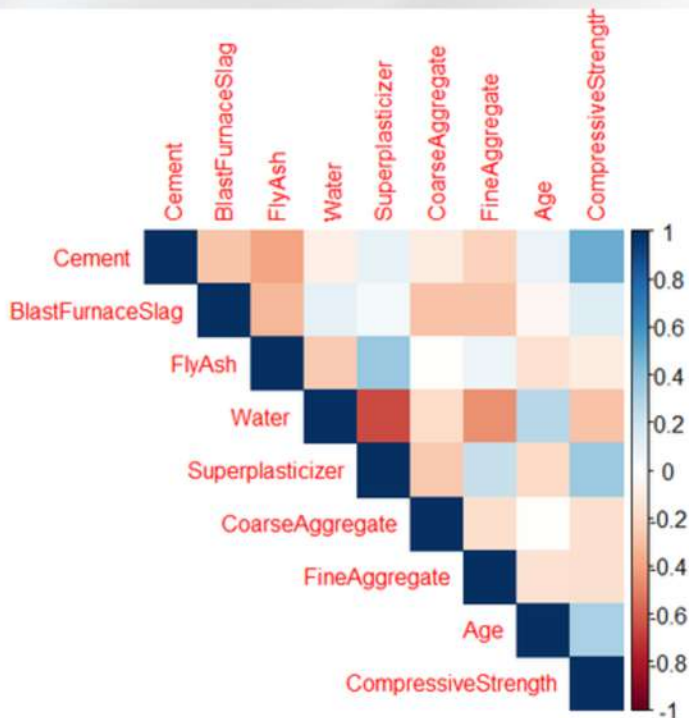


Objective:

- Explore relationships, distributions, and correlations between variables.

Visualizations Performed:

- Histogram of Compressive Strength
- Slight right skew, most values 20–50 MPa
- Correlation Heatmap using corrplot:
 - Cement, Age, Water have strong correlations with strength
- Scatter Plots:
 - Cement \uparrow \rightarrow Strength \uparrow
 - Water \uparrow \rightarrow Strength \downarrow (inversely related)

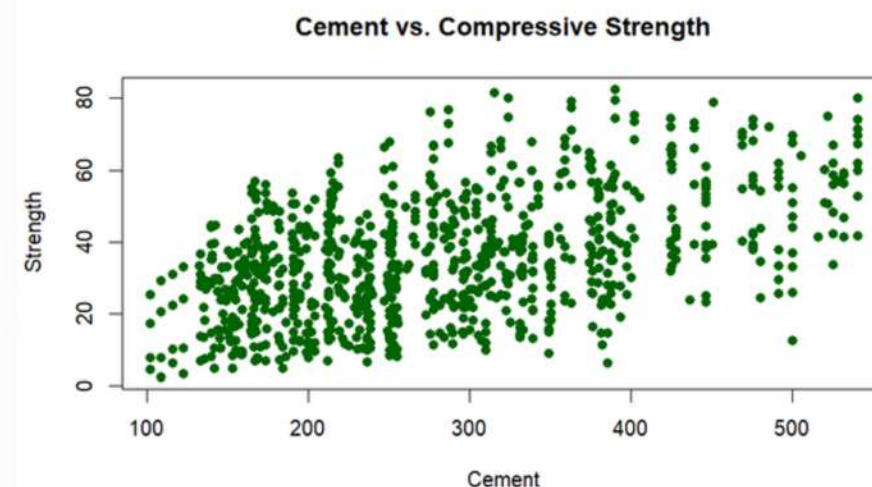


CORRELATION HEATMAP

```
library(corrplot)  
cor_matrix <- cor(concrete)  
corrplot(cor_matrix, method = "color", type = "upper", tl.cex = 0.8)
```

SCATTER PLOT: CEMENT VS. STRENGTH

```
plot(concrete$Cement, concrete$CompressiveStrength,  
     main = "Cement vs. Compressive Strength",  
     xlab = "Cement", ylab = "Strength", col = "darkgreen", pch = 16)
```



Insights:

- Cement and Age have positive influence on strength
- Water and Fly Ash may have negative or weak influence
- Some features are highly correlated \rightarrow possible multicollinearity

Linear Regression Modeling

Goal: Predict compressive strength using all variables

Predictive Modeling (Linear Regression)

Objective:

- Predict concrete compressive strength using a linear regression model

Actions Taken:

- Built model: `lm(CompressiveStrength ~ ., data = concrete)`
- Viewed model summary

Key Model Stats:

- Adjusted R²: 0.6125 → explains ~61% of variance
- Residual Std Error: 10.4
- F-statistic: Strong model overall
- Most variables are statistically significant ($p < 0.05$)

Visualization:

- Actual vs. Predicted plot with red reference line
- Model shows decent linear fit but room for improvement

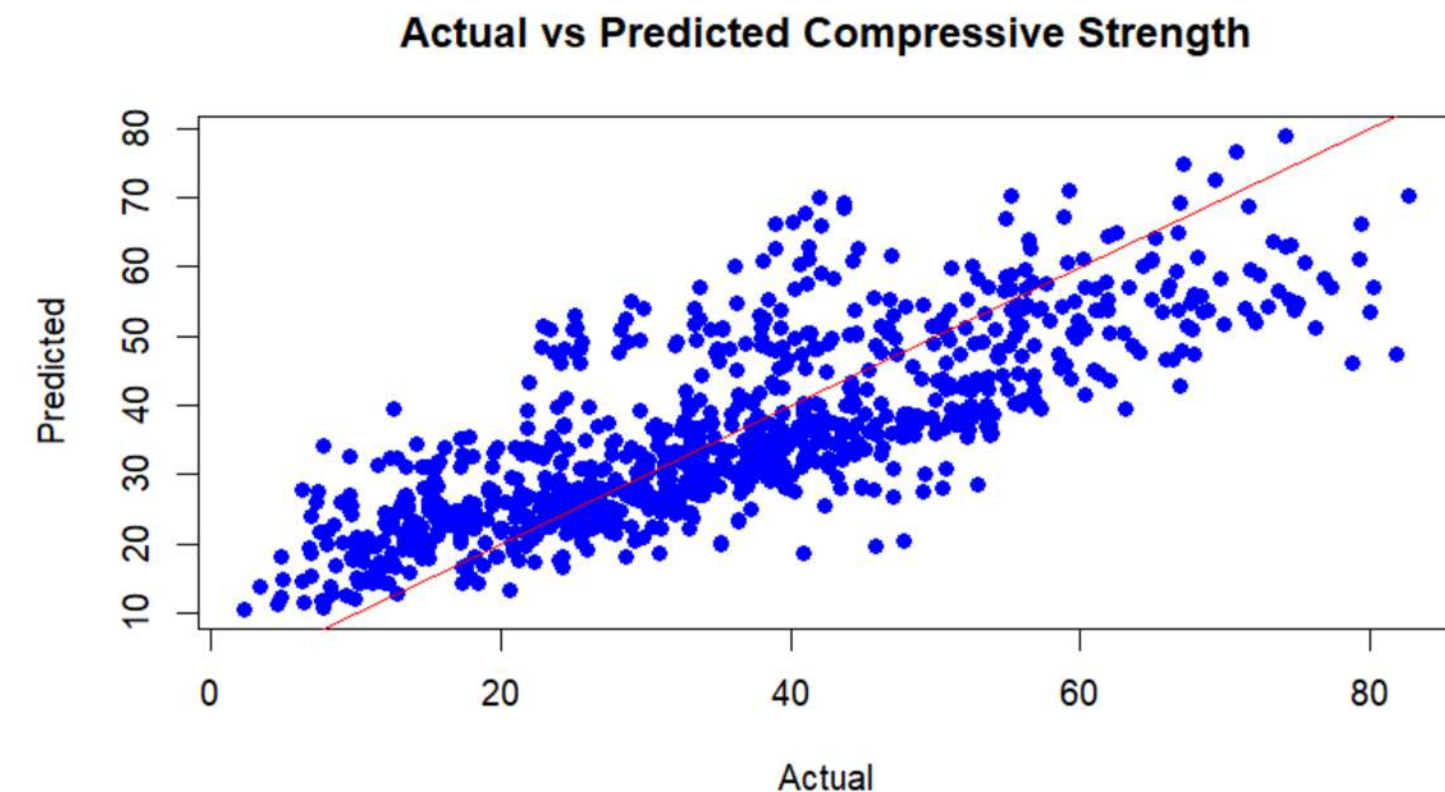
```
model <- lm(CompressiveStrength ~ ., data = concrete)
summary(model)

concrete$predicted <- predict(model)

plot(concrete$CompressiveStrength, concrete$predicted,
      main = "Actual vs Predicted Compressive Strength",
      xlab = "Actual", ylab = "Predicted", col = "blue", pch = 16)
abline(0, 1, col = "red")
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.4 on 1021 degrees of freedom
Multiple R-squared: 0.6155, Adjusted R-squared: 0.6125
F-statistic: 204.3 on 8 and 1021 DF, p-value: < 2.2e-16



From R Notebook to Shiny Dashboard

Why Shiny?

- To make the data interactive and allow real-time filtering & visualization.
- Move from static R notebook (.Rmd) to an app (.R script) using Shiny and shinydashboard.

R Markdown (RMD) File for Analysis

- started our analysis in an RMarkdown file (.Rmd).
- Performed:
 - Data import
 - Cleaning
 - EDA
 - Predictive modeling (Linear Regression)
 - Visualizations with ggplot2

Exported Cleaned Code to Shiny Script

- Reused the cleaned and structured code from .Rmd file.
- Converted it into an app.R script for Shiny app deployment.
- We modularized it into two parts:
 - ui – the user interface (layout, sliders, charts)
 - server – the logic (filtering, plotting, model response)

```
1 library(shiny)
2 library(shinydashboard)
3 library(tidyverse)
4
5
6 concrete <- read.csv("C:/Users/Tejashwini Saravanan/Copy of Concrete_Data.csv")
7 colnames(concrete) <- c("Cement", "BlastFurnaceSlag", "FlyAsh", "Water",
8   "Superplasticizer", "CoarseAggregate", "FineAggregate",
9   "Age", "CompressiveStrength")
10 concrete <- concrete %>% mutate(WC_Ratio = Water / Cement)
11
12
13 ui <- dashboardPage(
14   dashboardHeader(title = "Concrete Strength Dashboard"),
15   dashboardSidebar(
16     sliderInput("ageFilter", "Select Age (days):",
17       min = min(concrete$Age),
18       max = max(concrete$Age),
19       value = c(min(concrete$Age), max(concrete$Age))),
20
21     sliderInput("cementRange", "Select Cement Range (kg/m³):",
22       min = floor(min(concrete$Cement)),
23       max = ceiling(max(concrete$Cement)),
24       value = c(min(concrete$Cement), max(concrete$Cement)))
25   ),
26   dashboardBody(
27     tags$div(
28       style = "padding: 10px; text-align: center;",
29       tags$h2("The Concrete Dataset")
30     ),
31
32     fluidRow(
33       valueBoxOutput("avg_strength"),
34       valueBoxOutput("max_strength"),
35       valueBoxOutput("num_samples")
36     ),
37
38     fluidRow(
39       box(title = "Strength vs. Age", width = 6, plotOutput("plot1")),
40       box(title = "Strength vs. WC Ratio", width = 6, plotOutput("plot2"))
```

Dynamic KPI Cards: Insights by Mix ID

Purpose of KPI Cards:

- Summarize key numerical insights from the dataset in a quick, visual way.
- Help decision makers instantly spot important values like:
 - Average strength of concrete
 - Maximum strength observed
 - Minimum strength value
 - Standard deviation of strength (how much variability exists)

```
output$avg_strength <- renderValueBox({
  avg <- round(mean(filtered())$CompressiveStrength), 2)
  valueBox(avg, "Avg Strength (MPa)", icon = icon("chart-line"), color = "light-blue")
})

output$max_strength <- renderValueBox({
  max_val <- round(max(filtered())$CompressiveStrength), 2)
  valueBox(max_val, "Max Strength (MPa)", icon = icon("trophy"), color = "green")
})

output$num_samples <- renderValueBox({
  n <- nrow(filtered())
  valueBox(n, "Filtered Mixes", icon = icon("flask"), color = "orange")
})
```

KPI	What It Shows	Why It's Important
Average Strength	Mean for selected mix	See how strong that specific mix usually is
Max Strength	Peak strength for mix	Identify best-case potential
Min Strength	Lowest strength	Know safety limits or failures
Std. Dev.	Strength variability	Evaluate reliability or consistency of the mix

The Concrete Dataset

35.82

Avg Strength (MPa)



82.6

Max Strength (MPa)



1030

Filtered Mixes



Interactive Sliders – Age & Cement Input

Purpose of These Filters

- Allows users to customize input values and explore their effect on concrete compressive strength.
- Enables hands-on decision-making for selecting ideal ingredient combinations.
- Makes the dashboard interactive and user-friendly, similar to professional tools like Tableau.

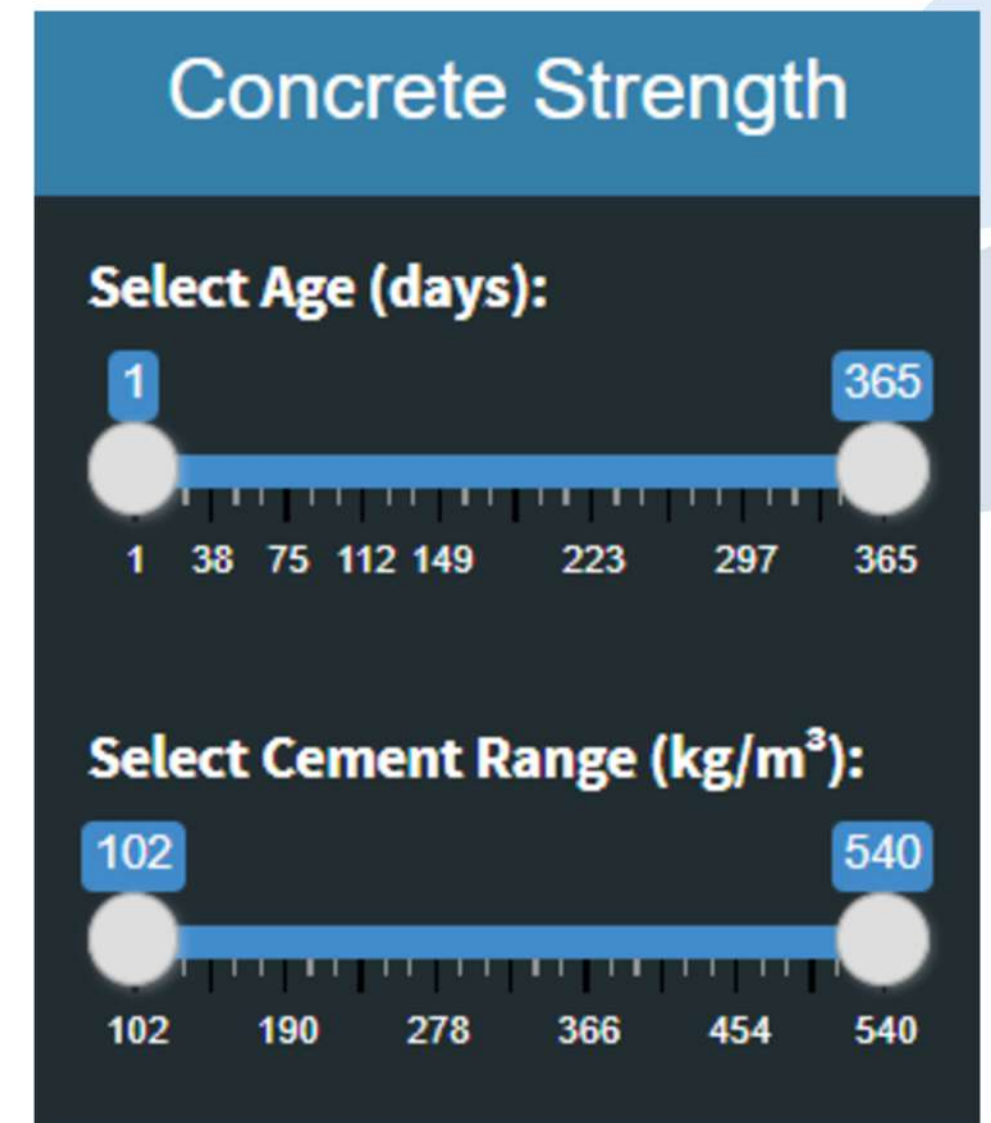
Filter 1: Cement Amount Slider

- Found on the left panel dashboard.
- Lets the user choose a specific cement quantity (kg/m³).
- Ranges from the minimum to maximum cement values in the dataset.
- Reactively updates all visualizations and summary stats.
- Useful for understanding:
 - How increasing or decreasing cement affects strength.
 - Finding the sweet spot for optimal cement usage.

Filter 2: Age of Concrete Slider

- Also located on the left sidebar.
- Allows users to select the age of concrete in days (curing time).
- Strength of concrete increases as it cures over time.
- Helps answer questions like:
 - "How strong will this mix be after 28 days?"
 - "Is this mix strong enough after just 7 days?"

```
server <- function(input, output) {  
  filtered <- reactive({  
    concrete %>%  
      filter(Age >= input$ageFilter[1],  
             Age <= input$ageFilter[2],  
             Cement >= input$cementRange[1],  
             Cement <= input$cementRange[2])  
  })  
}
```



WHY THESE FILTERS MATTER (DECISION-MAKING CONTEXT)

ENGINEERS AND BUILDERS CAN TEST COMBINATIONS TO:
INCREASE DURABILITY

REDUCE UNNECESSARY COSTS (E.G., TOO MUCH CEMENT)

ENSURE STRENGTH AFTER SPECIFIC CURING PERIODS

ENCOURAGES DATA-DRIVEN EXPERIMENTATION WITH REAL OUTCOMES.

Key Visualizations in Dashboard

Strength vs. Age

- **What it shows:**

Scatter plot with linear trendline of concrete age vs. compressive strength.

- **Insight:**

As concrete cures over time, its strength increases steadily especially within the first few weeks.

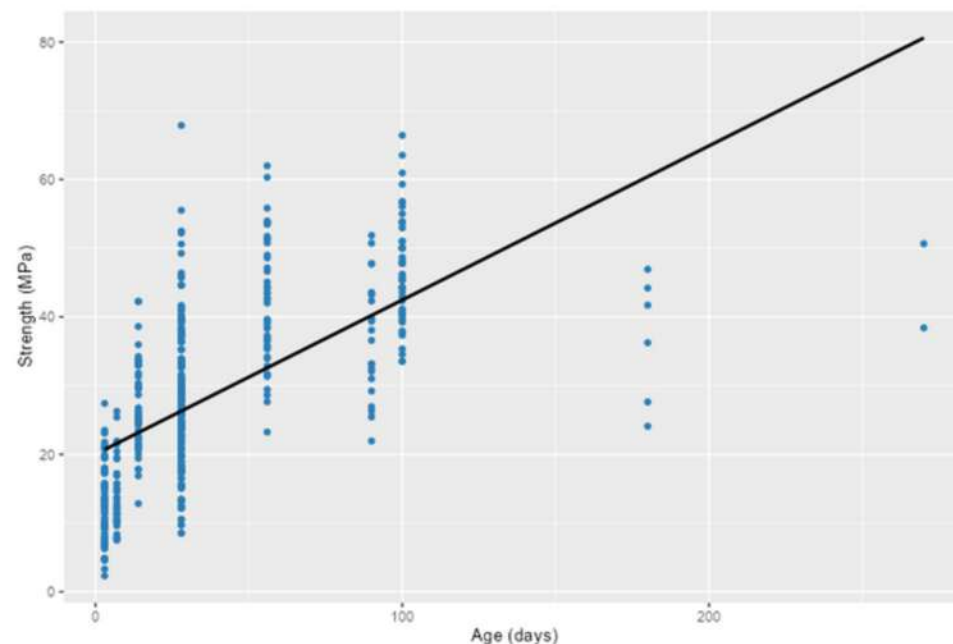
- **Why important:**

Helps construction engineers plan optimal curing periods to achieve required strength.

- **Dashboard Feature:**

Reacts to age and cement filters, so users can see how age interacts with other variables.

Strength vs. Age



Strength vs. Water-Cement (WC) Ratio

- **What it shows:**

Scatter plot showing the inverse relationship between water-to-cement ratio and strength.

- **Insight:**

Lower WC ratios generally result in stronger concrete, but too little water can hurt workability.

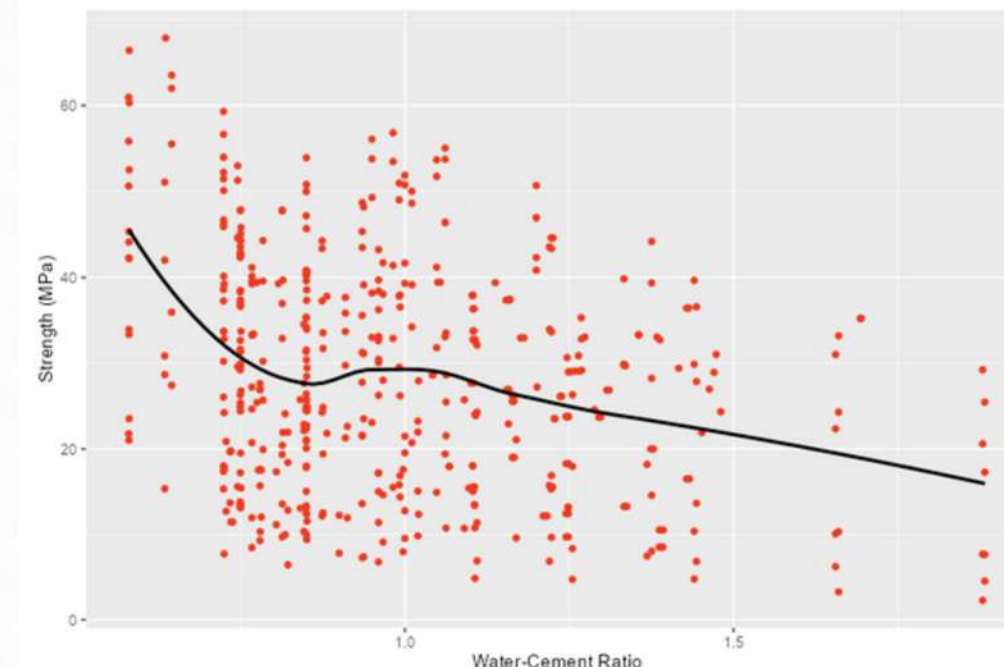
- **Why important:**

Balancing water and cement is key to mix design—this chart visualizes trade offs clearly.

- **Dashboard Feature:**

Dynamic and filtered in real-time based on selected input ranges.

Strength vs. WC Ratio



Key Visualizations in Dashboard

Strength vs. Cement

- **What it shows:**

Plot of how compressive strength varies with cement content in the mix.

- **Insight:**

Strength increases with cement content up to a point, but returns may diminish beyond that.

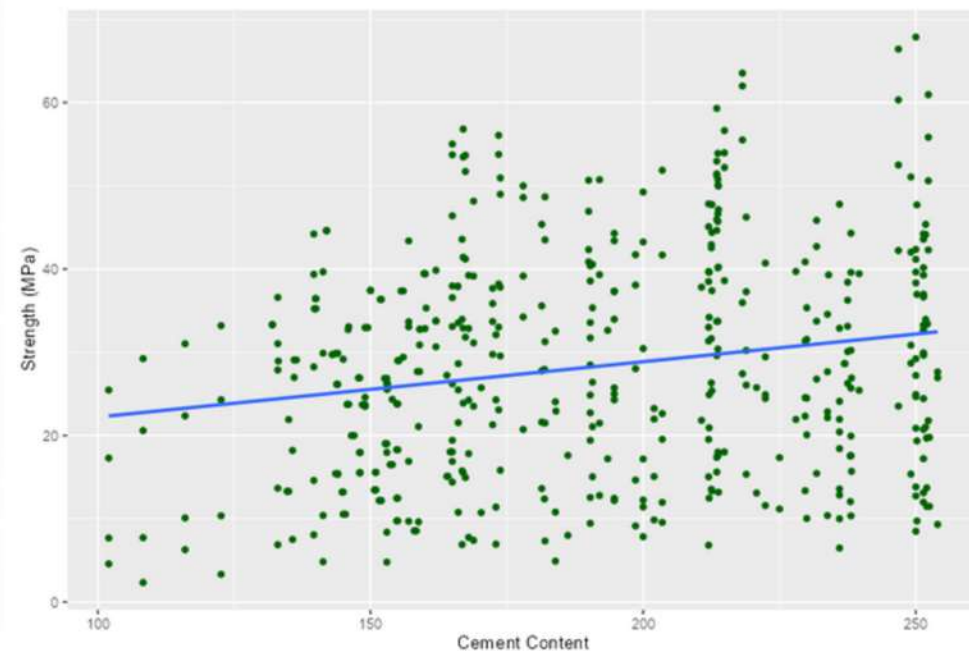
- **Why important:**

Encourages cost-effective cement use without overdesigning the mix.

- **Dashboard Feature:**

Users can explore impact using cement slider filter.

Strength vs. Cement



Strength vs. Fly Ash

- **What it shows:**

Chart analyzing contribution of fly ash, a supplementary material, to overall strength.

- **Insight:**

Fly ash boosts long-term strength, especially in later ages, and reduces environmental impact.

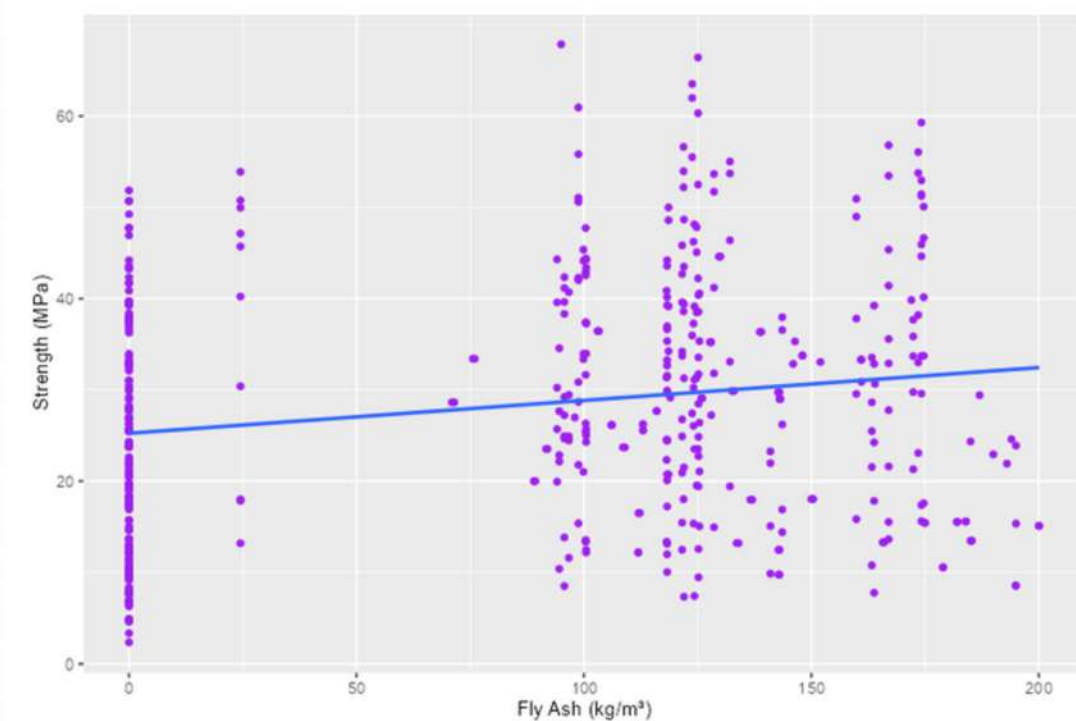
- **Why important:**

Highlights sustainable materials in concrete mix optimization.

- **Dashboard Feature:**

Works in combination with Age and Cement filters for layered insights.

Strength vs. Fly Ash



Recommendations and 3W's

Final Recommendations

- Use cement efficiently higher cement content improves strength but with diminishing returns.
- Aim for a low Water-Cement (W/C) ratio to improve strength without compromising mix workability.
- Consider age and curing time strength increases with time, especially in early days.
- Fly ash can be a sustainable substitute that contributes to strength over time.
- Leverage interactive dashboards for testing ingredient combinations before field use.

What Went Wrong

- Faced a deployment error:
"app.R did not return a shiny.appobj object"
Resolved by placing app.R and the dataset in the same folder, then publishing through shinyapps.io.
- Had to restructure file paths and ensure no folders or file names had spaces (e.g., changed to ShinyConcreteDashboard).
- Initially struggled with reactive filtering, but debugging helped me learn more about how reactive() and input\$ work in Shiny.

What I Learned

- Learned how to clean, model, and visualize data in R and Shiny for decision-making.
- Understood how different concrete ingredients influence strength through EDA and modeling.
- Gained experience using linear regression and applying it in a real-world material science context.
- Developed skills in building interactive dashboards using Shiny, ggplot2, and shinydashboard.

What I Would Change

- Add more interactive controls in the dashboard (e.g., dropdowns for mix type).
- Experiment with non-linear models like Random Forest for better prediction accuracy.
- Include cost metrics alongside strength to optimize for both strength and affordability.
- Improve UI styling for better visual design and mobile responsiveness.

The background features a collection of architectural blueprints and rolled-up plans. The blueprints are spread out, showing various technical drawings, dimensions, and labels such as '1', '2000', 'CH3000', '1650', '5600', '2300', 'A', 'B', 'h', '145', '725', '906', '808', '2400', and 'AC'. The rolled-up plans are stacked, with their ends showing circular openings. The overall scene is set against a light, neutral background with a subtle gradient.

Thank You.