CIA-ACADEMIC INFORMATION-GRADES AND ACADEMIC PROGRESSION

Janhvee Awasthi 24221120

2025-04-12

SUBMITTED TO DR.DURGANSH SHARMA

BUSINESS PROBLEM

Data Inaccuracy and Errors: Manual entry or poorly integrated systems can result in incorrect or outdated student information, leading to confusion and errors in decision-making.

1.Data Fragmentation: Academic data is often stored across various platforms, such as spreadsheets, paper records, and separate databases, making it difficult to get a comprehensive view of a student's academic history.

2.Difficulty in Accessing Information: With multiple sources of information and no centralized system, it can be time-consuming for administrators or faculty to retrieve student records when needed, leading to delays in academic planning, interventions, or decision-making.

3.Lack of Insights and Analytics: Without a structured system, analyzing student performance trends, identifying areas of improvement, or monitoring academic progression across different stages becomes a complex and tedious task.

4.Inefficient Tracking and Monitoring: Tracking students' academic progression over time (e.g., from class 10 to post-graduation) is often manual, leading to difficulties in identifying students who may need extra support or recognition.

INFROMATION SYSTEM OBJECTIVES

The objectives of the INFORMATION SYSTEM are as follows:

1.Centralized Data Management: To centralize and store all students' academic information in one system, making it easy to manage, access, and update data as needed.

2.Efficient Record Keeping: To ensure accurate, organized, and up-to-date records of students' academic history, including class 10th, 12th, graduation, and post-graduation results.

3.Data Integrity and Security: To maintain the integrity and security of student data, ensuring that it is protected from unauthorized access or accidental loss.

4.Data Accessibility: To allow authorized personnel (e.g., academic coordinators, administrators) to access student data quickly and efficiently for monitoring progress and making informed decisions.

5.Automated Reporting and Insights: To automate the generation of reports that provide insights into student performance and progression, helping identify trends, gaps, or areas requiring attention.

6.Support Data-Driven Decisions: To assist in making data-driven decisions regarding student performance, academic interventions, and institutional policies.

7.Enhanced Academic Tracking: To track and visualize individual students' academic progression, from initial school levels to post-graduation, enabling personalized academic guidance.

8.Improved Communication: To facilitate communication among faculty, students, and administrators regarding academic performance and updates.

WHAT HAS BEEN SHOWCASED

*THE ACADEMIC PROGRESSION OF 10 STUDENTS HAS BEEN SHOWCASED, WHICH REPRESNTS THAT HOW A STUDENT HAS PERFORMED OVER A PERIOD OF TIME. STARTING FROM CLASS 10TH UPTO POST GRADUTION LEVEL. THRIUGH LINE CHART THE PERFORMANCE OF THE STUDENT HAS BEEN SHOWN.

*THE ATTENDANCE OF ALL THE STUDENTS ARE ABOVE 75%WHICH MEANS THAT ALL 15 STUDENTS ARE IN THE RISK FREE ZONE AND CAN APPEAR FOR EXAM.FOR SHOWING ATTENDANCE OF STUDENTS SCATTER PLOT HAS BEEN USED FOR DERIVING CORRECT CONCLUSION.

*COURSE WISE ATTENDANCE OF STUDENT HAVE BEEN SHOWCASED THRUGH BAR PLOTS.

*THE PERSONAL DETAILS OF A THE STUDENTS HAS ALSO BEEN SHOWN.

*THE GRADES OF HTE STUDENTS HAVE BEEN SHOWN THROUGH VISULAISATION. THE TOTAL MARKS ALONG WITH WHICH STUDENT HAS PASSED IN WHICH SUBJECT AND WHICH STUDENT HAS FAILED.

INSIGHTS

1.Academic Performance Trends:

The portal provides an overview of student performance across different stages (Class 10, Class 12, Graduation, Post Graduation), allowing administrators to track academic progression over time.

Insights can be drawn about students' improvement or decline in performance at each academic level, helping identify patterns such as consistent high achievers or students who are struggling.

2.Identification of At-Risk Students:

By analyzing the academic data, the portal helps identify students whose performance has dropped or who are consistently underperforming. This allows for timely intervention, such as tutoring, academic counseling, or other support mechanisms.

Students who are falling behind in certain subjects or stages can be flagged for additional academic assistance or personalized learning paths.

3.Performance of Individual Students:

The portal allows for a deep dive into the individual performance of each student, providing detailed reports on their marks, grades, and overall progression. This information can be used for personal academic mentoring and counseling.

4.Academic progression By analyzing past academic data, the portal can offer predictive insights into future performance, helping educators and administrators to forecast graduation success or potential challenges faced by students.

CONCLUSION

The visual insights derived from the system—such as performance trends, academic drops, and attendance correlations—enable timely academic interventions and personalized support for students. Moreover, the ability to monitor performance from secondary school through post-graduation gives a holistic view of each student's academic journey.

From improving academic outcomes to aiding administrative efficiency, the Knowledge Portal represents a scalable, intelligent solution that aligns with modern educational needs. With further development, it holds the potential to integrate predictive analytics, behavioral tracking, and even AI-driven academic guidance—making it a cornerstone of digital education transformation.

#CODES

```
# First 10 students (IDs 7001 to 7010)
Student_ID_Table_1 <- data.frame(
  student_id = c(7001, 7002, 7003, 7004, 7005, 7006, 7007, 7008, 7009, 7010),
  personal_details = c(
    "Details of Anaya Singh",
    "Details of Arjun Deshmukh",
    "Details of Diya Reddy",
    "Details of Krish Malhotra",
    "Details of Isha Patel",
    "Details of Aditya Menon",
    "Details of Saanvi Dubey",
    "Details of Reyansh Kapoor",
    "Details of Tanya Bansal",
    "Details of Yuvan Mehta"
  ),
  academic_details = c(
    "Academic details for Anaya",
    "Academic details for Arjun",
    "Academic details for Diya",
    "Academic details for Krish",
    "Academic details for Isha",
    "Academic details for Aditya",
    "Academic details for Saanvi",
    "Academic details for Reyansh",
    "Academic details for Tanya",
    "Academic details for Yuvan"
  ),
  work_experience = c(
    "Interned at Deloitte",
    "Interned at IBM",
    "Interned at Cognizant",
    "Interned at Oracle",
    "Interned at Microsoft",
    "Interned at Capgemini",
    "Interned at EY",
    "Interned at SAP",
    "Interned at PwC",
    "Interned at Google"
  ),
  pre_review = c(
    "Quick learner",
```

```
    "Strategic thinker",
    "Excellent communicator",
    "Innovative mind",
    "Disciplined worker",
    "Analytical thinker",
    "Collaborative team player",
    "Goal-oriented",
    "Detail-oriented",
    "Exceptional coder"
  ),
  admission_id = c(3001, 3002, 3003, 3004, 3005, 3006, 3007, 3008, 3009, 3010),
  stringsAsFactors = FALSE
)

# Original 5 students (IDs 7011 to 7015)
Student_ID_Table_2 <- data.frame(
  student_id = c(7011, 7012, 7013, 7014, 7015),
  personal_details = c(
    "Details of Aarav Sharma",
    "Details of Kavya Iyer",
    "Details of Rohan Verma",
    "Details of Meera Nair",
    "Details of Vihaan Joshi"
  ),
  academic_details = c(
    "Academic details for Aarav",
    "Academic details for Kavya",
    "Academic details for Rohan",
    "Academic details for Meera",
    "Academic details for Vihaan"
  ),
  work_experience = c(
    "Interned at TCS",
    "Interned at Wipro",
    "Interned at Infosys",
    "Interned at HCL",
    "Interned at Accenture"
  ),
  pre_review = c(
    "Dedicated learner",
    "Creative thinker",
    "Problem solver",
    "Team player",
    "Critical thinker"
  ),
  admission_id = c(3011, 3012, 3013, 3014, 3015),
  stringsAsFactors = FALSE
)

# Merge both data frames into one
Student_ID_Table <- rbind(Student_ID_Table_1, Student_ID_Table_2)

# View the combined data frame
```

```r
print(Student_ID_Table)
```

```
##    student_id          personal_details             academic_details
## 1        7001    Details of Anaya Singh   Academic details for Anaya
## 2        7002 Details of Arjun Deshmukh   Academic details for Arjun
## 3        7003     Details of Diya Reddy    Academic details for Diya
## 4        7004 Details of Krish Malhotra   Academic details for Krish
## 5        7005     Details of Isha Patel    Academic details for Isha
## 6        7006   Details of Aditya Menon  Academic details for Aditya
## 7        7007   Details of Saanvi Dubey  Academic details for Saanvi
## 8        7008 Details of Reyansh Kapoor Academic details for Reyansh
## 9        7009   Details of Tanya Bansal   Academic details for Tanya
## 10       7010    Details of Yuvan Mehta   Academic details for Yuvan
## 11       7011   Details of Aarav Sharma   Academic details for Aarav
## 12       7012     Details of Kavya Iyer   Academic details for Kavya
## 13       7013    Details of Rohan Verma   Academic details for Rohan
## 14       7014     Details of Meera Nair   Academic details for Meera
## 15       7015   Details of Vihaan Joshi  Academic details for Vihaan
##          work_experience               pre_review admission_id
## 1   Interned at Deloitte             Quick learner         3001
## 2        Interned at IBM         Strategic thinker         3002
## 3  Interned at Cognizant    Excellent communicator         3003
## 4     Interned at Oracle           Innovative mind         3004
## 5  Interned at Microsoft        Disciplined worker         3005
## 6  Interned at Capgemini        Analytical thinker         3006
## 7         Interned at EY Collaborative team player         3007
## 8        Interned at SAP             Goal-oriented         3008
## 9        Interned at PwC           Detail-oriented         3009
## 10    Interned at Google         Exceptional coder         3010
## 11       Interned at TCS         Dedicated learner         3011
## 12     Interned at Wipro          Creative thinker         3012
## 13   Interned at Infosys            Problem solver         3013
## 14       Interned at HCL               Team player         3014
## 15 Interned at Accenture          Critical thinker         3015
```

```r
#ATTENDANCE
# Create Course_Attendance data frame for 15 students
Course_Attendance <- data.frame(
  course_id = rep(1, 15),
  student_id = 7001:7015,
  attendance_date = rep("2023-08-02", 15),
  status = c(
    "Absent", "Present", "Present", "Absent", "Present",
    "Present", "Absent", "Present", "Absent", "Present",
    "Absent", "Present", "Present", "Absent", "Present"
  ),
  remarks = c(
    "Medical reason", "", "", "Personal emergency", "",
    "", "Travel issues", "", "Family function", "",
    "Flu symptoms", "", "", "Late arrival", ""
  ),
  stringsAsFactors = FALSE
)
```

```r
# View the data frame
print(Course_Attendance)
```

```
##    course_id student_id attendance_date  status            remarks
## 1          1       7001      2023-08-02  Absent     Medical reason
## 2          1       7002      2023-08-02 Present
## 3          1       7003      2023-08-02 Present
## 4          1       7004      2023-08-02  Absent Personal emergency
## 5          1       7005      2023-08-02 Present
## 6          1       7006      2023-08-02 Present
## 7          1       7007      2023-08-02  Absent      Travel issues
## 8          1       7008      2023-08-02 Present
## 9          1       7009      2023-08-02  Absent    Family function
## 10         1       7010      2023-08-02 Present
## 11         1       7011      2023-08-02  Absent       Flu symptoms
## 12         1       7012      2023-08-02 Present
## 13         1       7013      2023-08-02 Present
## 14         1       7014      2023-08-02  Absent       Late arrival
## 15         1       7015      2023-08-02 Present
```

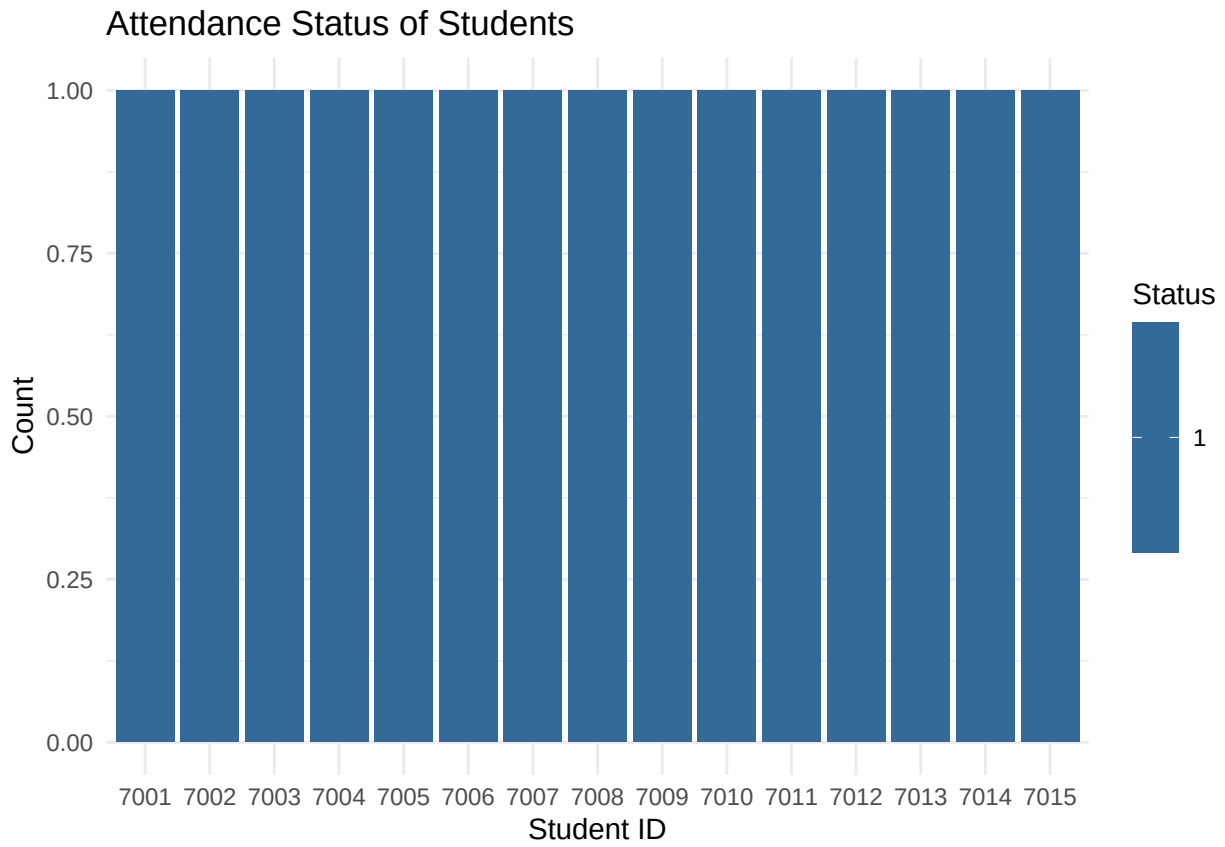
```r
#BAR CHART FOR ATTENDANCE
# Load necessary library
library(tidyr)
```

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

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```r
# Aggregate data: count of Present and Absent for each student
attendance_summary <- aggregate(status ~ student_id + status, data = Course_Attendance, length)

# Plot bar chart
ggplot(attendance_summary, aes(x = factor(student_id), y = status, fill = status)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Attendance Status of Students",
    x = "Student ID",
    y = "Count",
    fill = "Status"
  ) +
  theme_minimal()
```



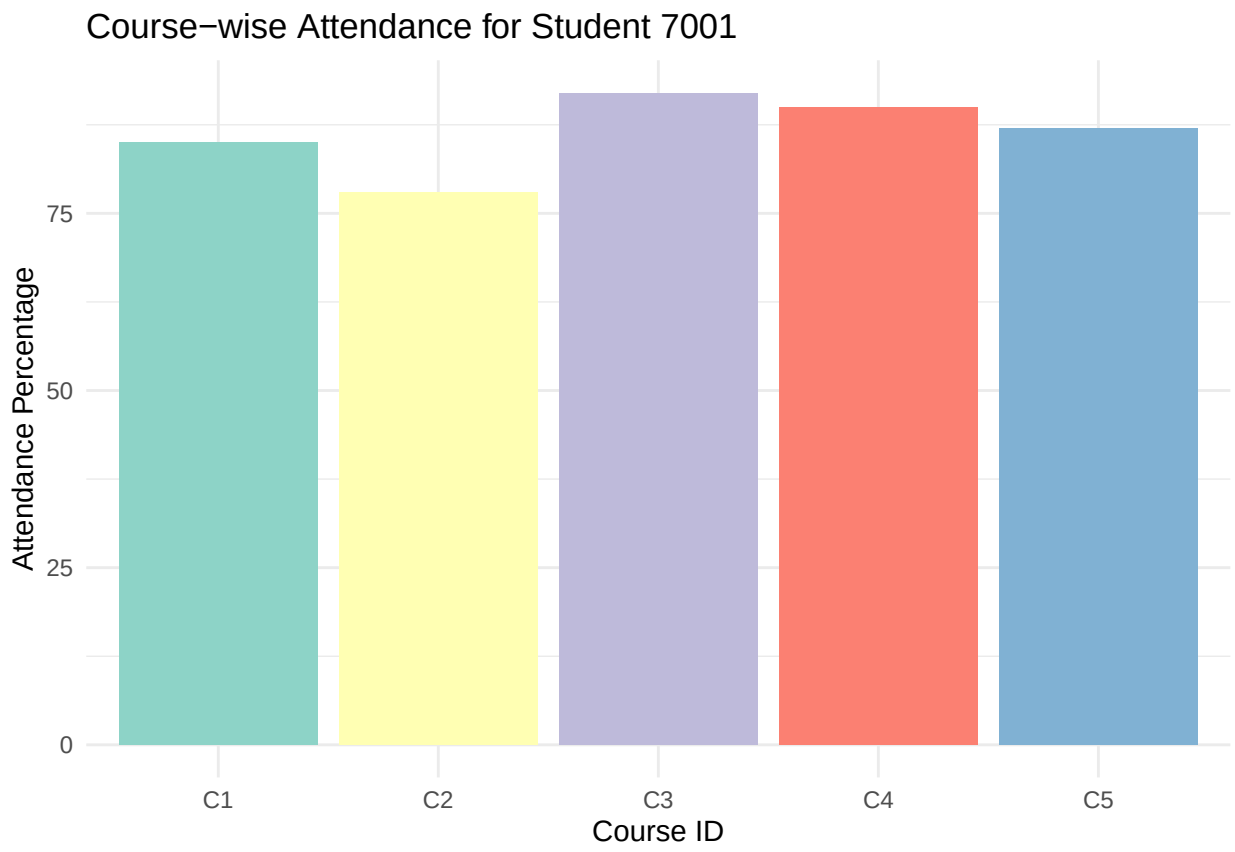

```
# Given attendance percentages
attendance_percentages <- c(85, 78, 78, 92, 98, 78, 85, 89, 90, 79, 88, 90, 94, 95, 84)

# Add attendance_percentage column to Course_Attendance
Course_Attendance$attendance_percentage <- attendance_percentages

# View the updated data frame
print(Course_Attendance)
```

```
##    course_id student_id attendance_date  status            remarks
## 1          1       7001      2023-08-02  Absent     Medical reason
## 2          1       7002      2023-08-02 Present
## 3          1       7003      2023-08-02 Present
## 4          1       7004      2023-08-02  Absent Personal emergency
## 5          1       7005      2023-08-02 Present
## 6          1       7006      2023-08-02 Present
## 7          1       7007      2023-08-02  Absent      Travel issues
## 8          1       7008      2023-08-02 Present
## 9          1       7009      2023-08-02  Absent    Family function
## 10         1       7010      2023-08-02 Present
## 11         1       7011      2023-08-02  Absent       Flu symptoms
## 12         1       7012      2023-08-02 Present
## 13         1       7013      2023-08-02 Present
## 14         1       7014      2023-08-02  Absent       Late arrival
## 15         1       7015      2023-08-02 Present
##    attendance_percentage
```

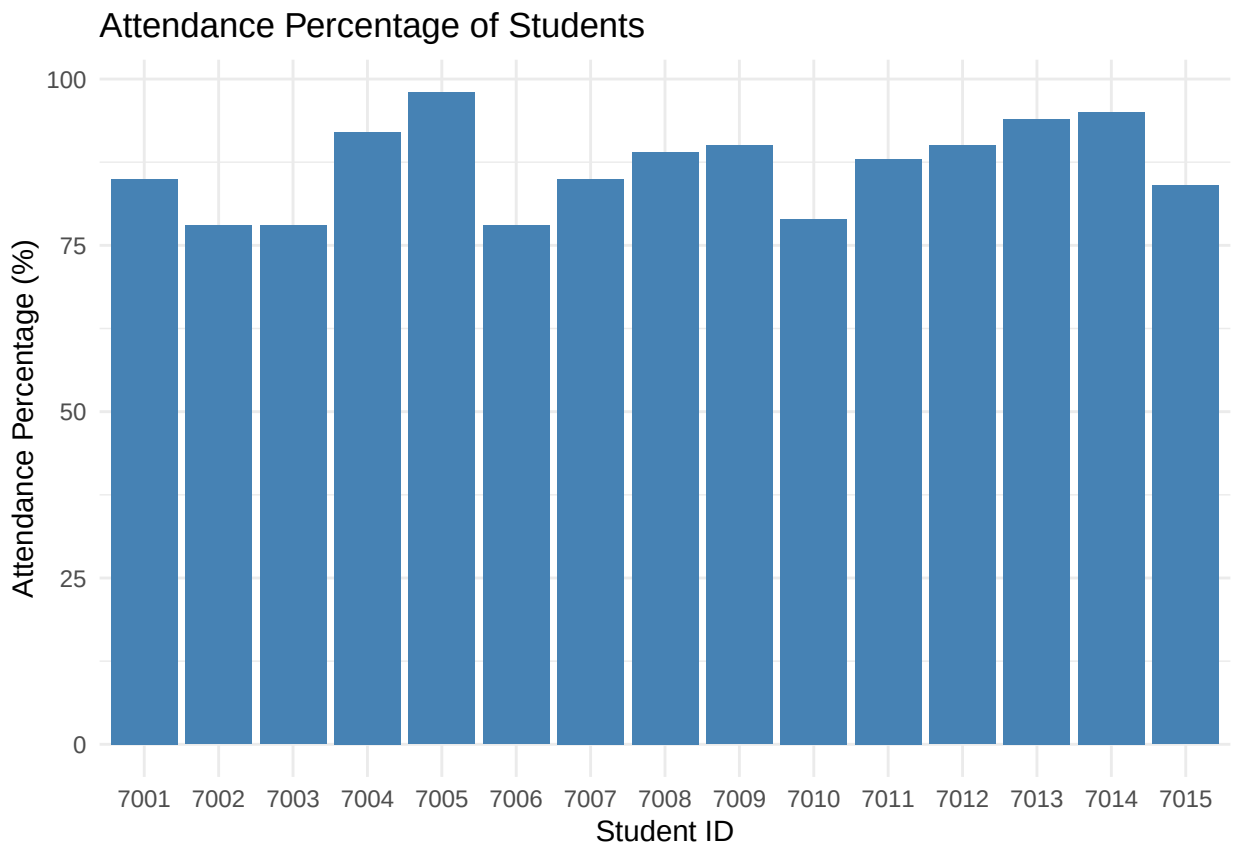
```
## 1                    85
## 2                    78
## 3                    78
## 4                    92
## 5                    98
## 6                    78
## 7                    85
## 8                    89
## 9                    90
## 10                   79
## 11                   88
## 12                   90
## 13                   94
## 14                   95
## 15                   84
```

```r
#Course wise attendance of each student
# Create a data frame for course-wise attendance
Course_Attendance_Percentage <- data.frame(
  student_id = rep(7001:7015, each = 5),
  course_id = rep(c("C1", "C2", "C3", "C4", "C5"), times = 15),
  attendance_percent = c(
    85, 78, 92, 90, 87,  # Student 7001
    80, 70, 75, 88, 90,  # Student 7002
    88, 85, 82, 91, 80,  # Student 7003
    60, 65, 70, 72, 68,  # Student 7004
    95, 90, 88, 93, 94,  # Student 7005
    76, 80, 84, 78, 75,  # Student 7006
    85, 88, 83, 79, 86,  # Student 7007
    80, 82, 78, 81, 80,  # Student 7008
    70, 74, 72, 80, 79,  # Student 7009
    92, 91, 95, 94, 90,  # Student 7010
    68, 70, 72, 75, 71,  # Student 7011
    89, 90, 85, 88, 91,  # Student 7012
    77, 79, 82, 80, 75,  # Student 7013
    80, 85, 88, 91, 87,  # Student 7014
    75, 72, 70, 78, 79   # Student 7015
  )
)

# BAR PLOT FOR COURSE WISE ATTENDANCE FOR FIRST STUDENT
# Load necessary library
library(ggplot2)

# Filter data for the first student (student_id = 7001)
first_student_attendance <- subset(Course_Attendance_Percentage, student_id == 7001)

# Create a bar plot for the first student's course-wise attendance
ggplot(first_student_attendance, aes(x = course_id, y = attendance_percent, fill = course_id)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  labs(
    title = "Course-wise Attendance for Student 7001",
    x = "Course ID",
    y = "Attendance Percentage"
```

```
  ) +
  theme_minimal() +
  scale_fill_brewer(palette = "Set3")
```




```
# AGGREGATE COURSE ATTENDANCE FOR ALL STUDENTS
library(ggplot2)

# Create bar plot for attendance percentages
ggplot(Course_Attendance, aes(x = factor(student_id), y = attendance_percentage)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(
    title = "Attendance Percentage of Students",
    x = "Student ID",
    y = "Attendance Percentage (%)"
  ) +
  theme_minimal()
```

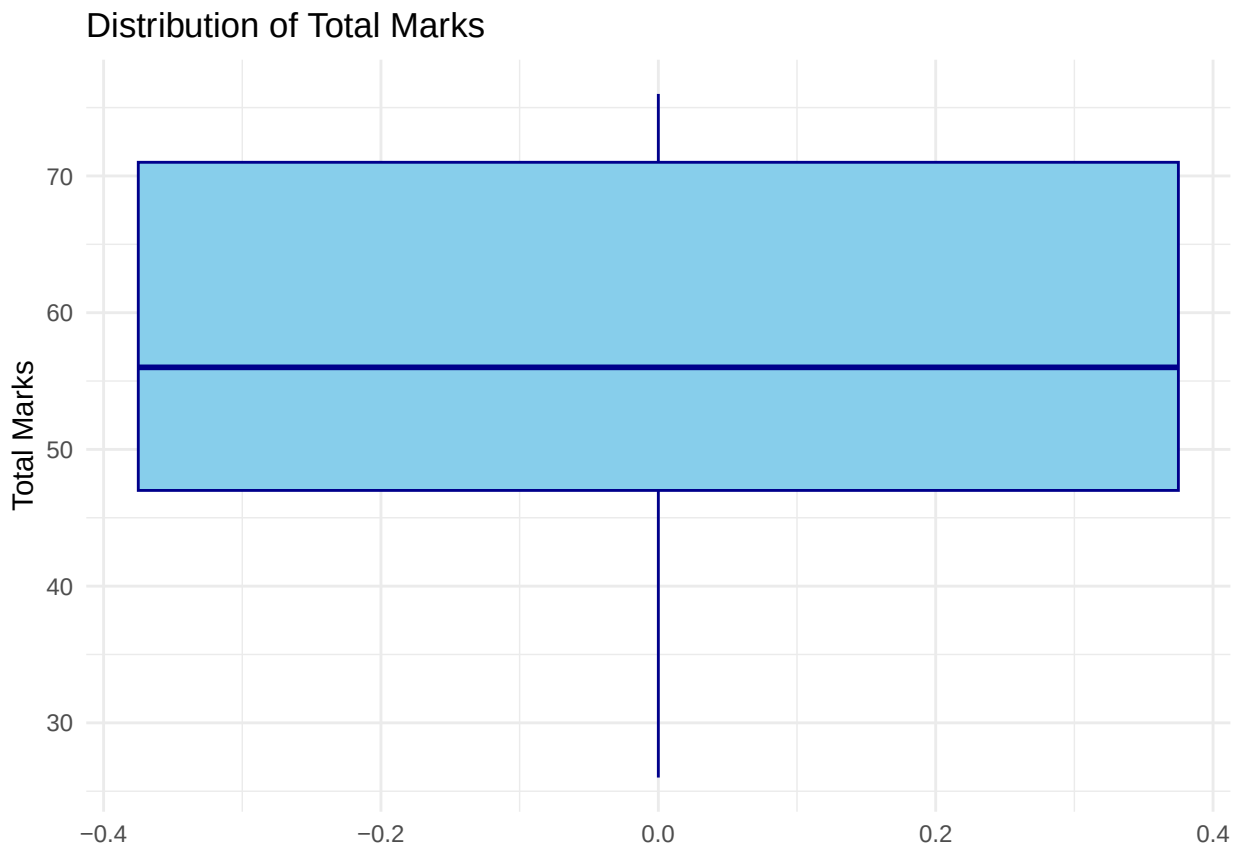



```r
#GRADES
# Create Grade data frame for 15 students
Grade <- data.frame(
  student_id = 7001:7015,
  course_id = rep(1, 15),
  cia1 = c(17, 12, 14, 9, 18, 15, 13, 16, 11, 19, 10, 18, 20, 8, 14),
  cia2 = c(16, 11, 13, 8, 17, 16, 12, 15, 10, 18, 11, 17, 19, 7, 13),
  cia3 = c(18, 13, 15, 10, 19, 14, 14, 17, 9, 18, 13, 18, 19, 6, 15),
  cia4 = c(19, 12, 14, 11, 18, 13, 15, 16, 10, 17, 12, 19, 18, 5, 14),
  credits = rep(3, 15),
  grading_date = rep("2024-11-10", 15),
  stringsAsFactors = FALSE
)

# Calculate total_marks
Grade$total_marks <- Grade$cia1 + Grade$cia2 + Grade$cia3 + Grade$cia4

# Assign Pass/Fail based on total_marks (let's say passing is 50 or more)
Grade$status <- ifelse(Grade$total_marks >= 50, "Pass", "Fail")

# Assign remarks
Grade$remark <- ifelse(
  Grade$status == "Fail", "Needs improvement",
  ifelse(Grade$total_marks >= 70, "Excellent",
         ifelse(Grade$total_marks >= 60, "Good performance", "Satisfactory"))
)
```

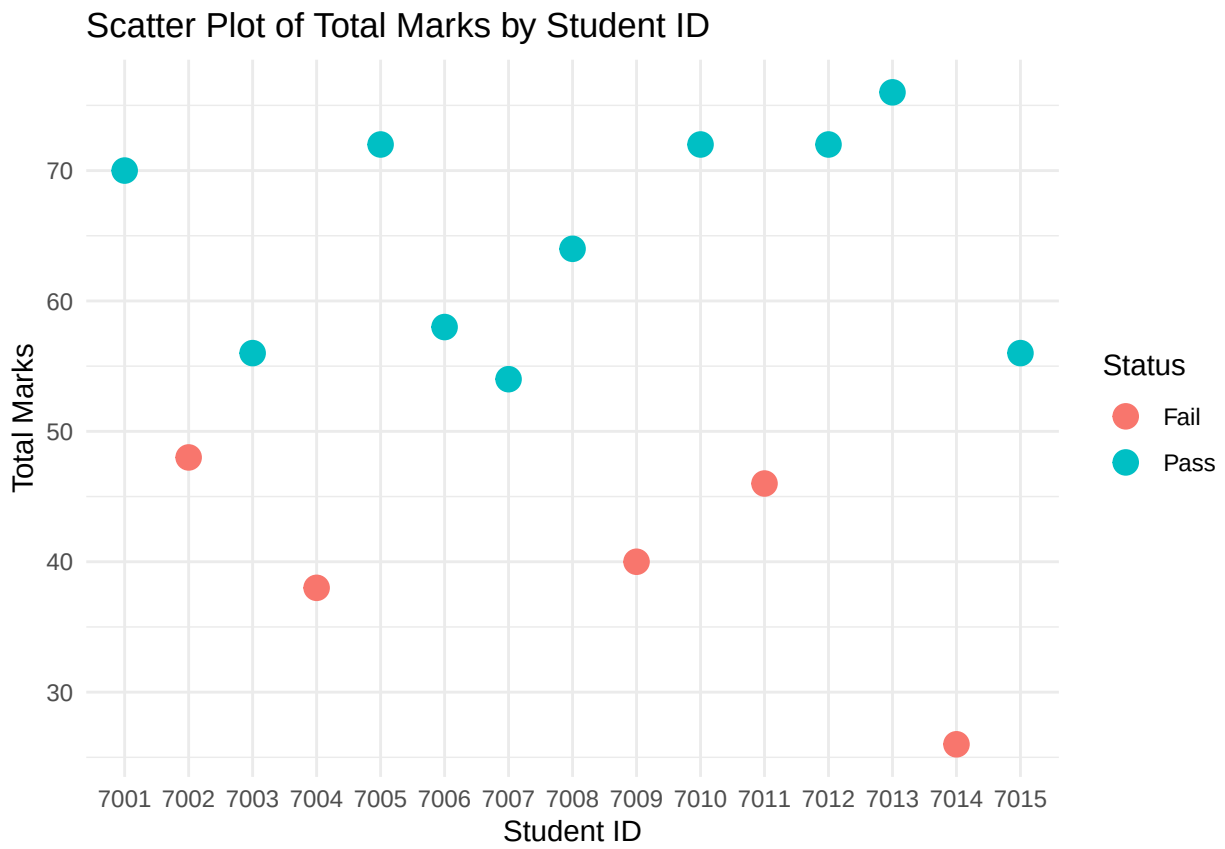
```r
# View the final Grade data frame
print(Grade)
```

```
##    student_id course_id cia1 cia2 cia3 cia4 credits grading_date total_marks
## 1        7001         1   17   16   18   19       3   2024-11-10          70
## 2        7002         1   12   11   13   12       3   2024-11-10          48
## 3        7003         1   14   13   15   14       3   2024-11-10          56
## 4        7004         1    9    8   10   11       3   2024-11-10          38
## 5        7005         1   18   17   19   18       3   2024-11-10          72
## 6        7006         1   15   16   14   13       3   2024-11-10          58
## 7        7007         1   13   12   14   15       3   2024-11-10          54
## 8        7008         1   16   15   17   16       3   2024-11-10          64
## 9        7009         1   11   10    9   10       3   2024-11-10          40
## 10       7010         1   19   18   18   17       3   2024-11-10          72
## 11       7011         1   10   11   13   12       3   2024-11-10          46
## 12       7012         1   18   17   18   19       3   2024-11-10          72
## 13       7013         1   20   19   19   18       3   2024-11-10          76
## 14       7014         1    8    7    6    5       3   2024-11-10          26
## 15       7015         1   14   13   15   14       3   2024-11-10          56
##    status            remark
## 1    Pass         Excellent
## 2    Fail Needs improvement
## 3    Pass      Satisfactory
## 4    Fail Needs improvement
## 5    Pass         Excellent
## 6    Pass      Satisfactory
## 7    Pass      Satisfactory
## 8    Pass  Good performance
## 9    Fail Needs improvement
## 10   Pass         Excellent
## 11   Fail Needs improvement
## 12   Pass         Excellent
## 13   Pass         Excellent
## 14   Fail Needs improvement
## 15   Pass      Satisfactory
```

```r
#GRADES VISUALISATION(box plot)
# Load ggplot2 for visualization

library(ggplot2)

# Create a box plot for total marks
ggplot(Grade, aes(y = total_marks)) +
  geom_boxplot(fill = "skyblue", color = "darkblue", outlier.color = "red", outlier.shape = 8) +
  labs(
    title = "Distribution of Total Marks",
    y = "Total Marks"
  ) +
  theme_minimal()
```




```r
#GRADES SCATTER PLOT
# Load ggplot2
library(ggplot2)

# Scatter plot: Student ID vs Total Marks
ggplot(Grade, aes(x = factor(student_id), y = total_marks, color = status)) +
  geom_point(size = 4) +
  labs(
    title = "Scatter Plot of Total Marks by Student ID",
    x = "Student ID",
    y = "Total Marks",
    color = "Status"
  ) +
  theme_minimal()
```

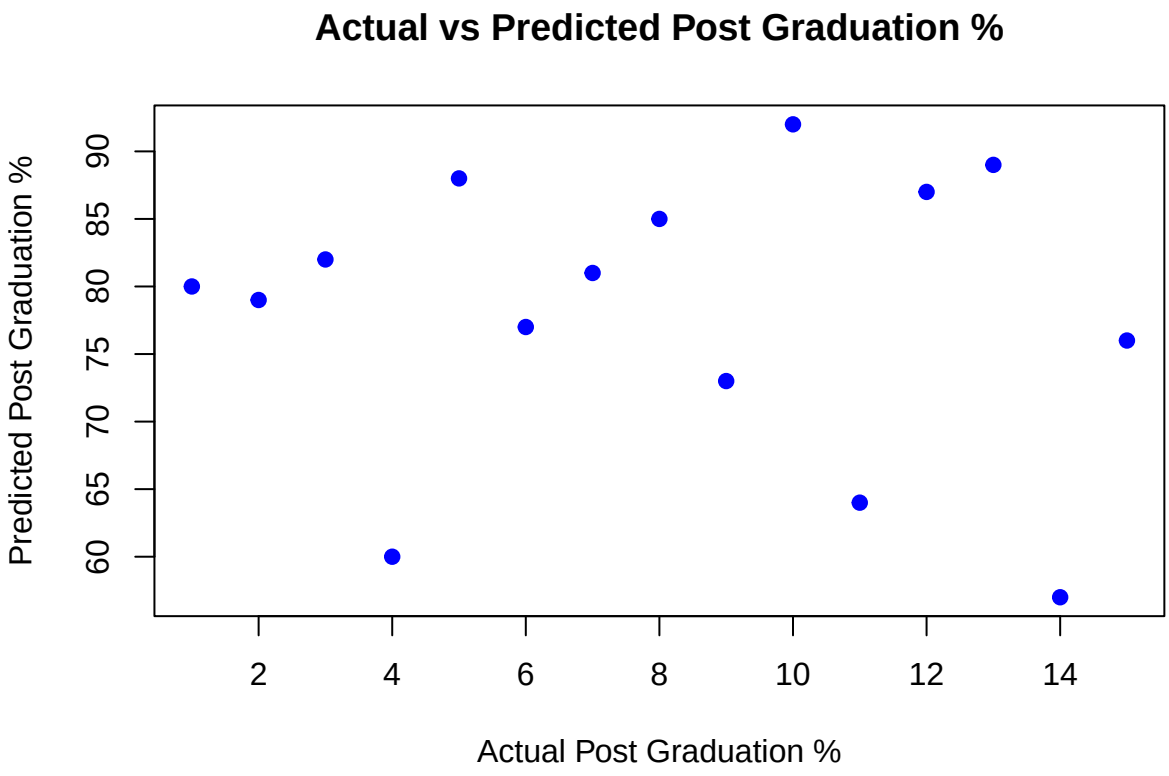



```
#Create Academic Progression data frame
    Academic_Progression <- data.frame(
      student_id = 7001:7015,
      class_10_percent = c(85, 78, 88, 65, 92, 76, 81, 79, 70, 95, 68, 90, 87, 60, 75),
      class_12_percent = c(82, 80, 85, 60, 90, 79, 83, 82, 72, 93, 66, 88, 89, 58, 77),
      graduation_percent = c(78, 76, 80, 58, 87, 75, 79, 84, 70, 91, 62, 85, 86, 55, 73),
      post_graduation_percent = c(80, 79, 82, 60, 88, 77, 81, 85, 73, 92, 64, 87, 89, 57, 76),
      stringsAsFactors = FALSE
    )

    # View the data frame
    print(Academic_Progression)
```

```
##    student_id class_10_percent class_12_percent graduation_percent
## 1        7001               85               82                78
## 2        7002               78               80                76
## 3        7003               88               85                80
## 4        7004               65               60                58
## 5        7005               92               90                87
## 6        7006               76               79                75
## 7        7007               81               83                79
## 8        7008               79               82                84
## 9        7009               70               72                70
## 10       7010               95               93                91
## 11       7011               68               66                62
## 12       7012               90               88                85
```

```
## 13       7013                87              89                86
## 14       7014                60              58                55
## 15       7015                75              77                73
##    post_graduation_percent
## 1                       80
## 2                       79
## 3                       82
## 4                       60
## 5                       88
## 6                       77
## 7                       81
## 8                       85
## 9                       73
## 10                      92
## 11                      64
## 12                      87
## 13                      89
## 14                      57
## 15                      76
```

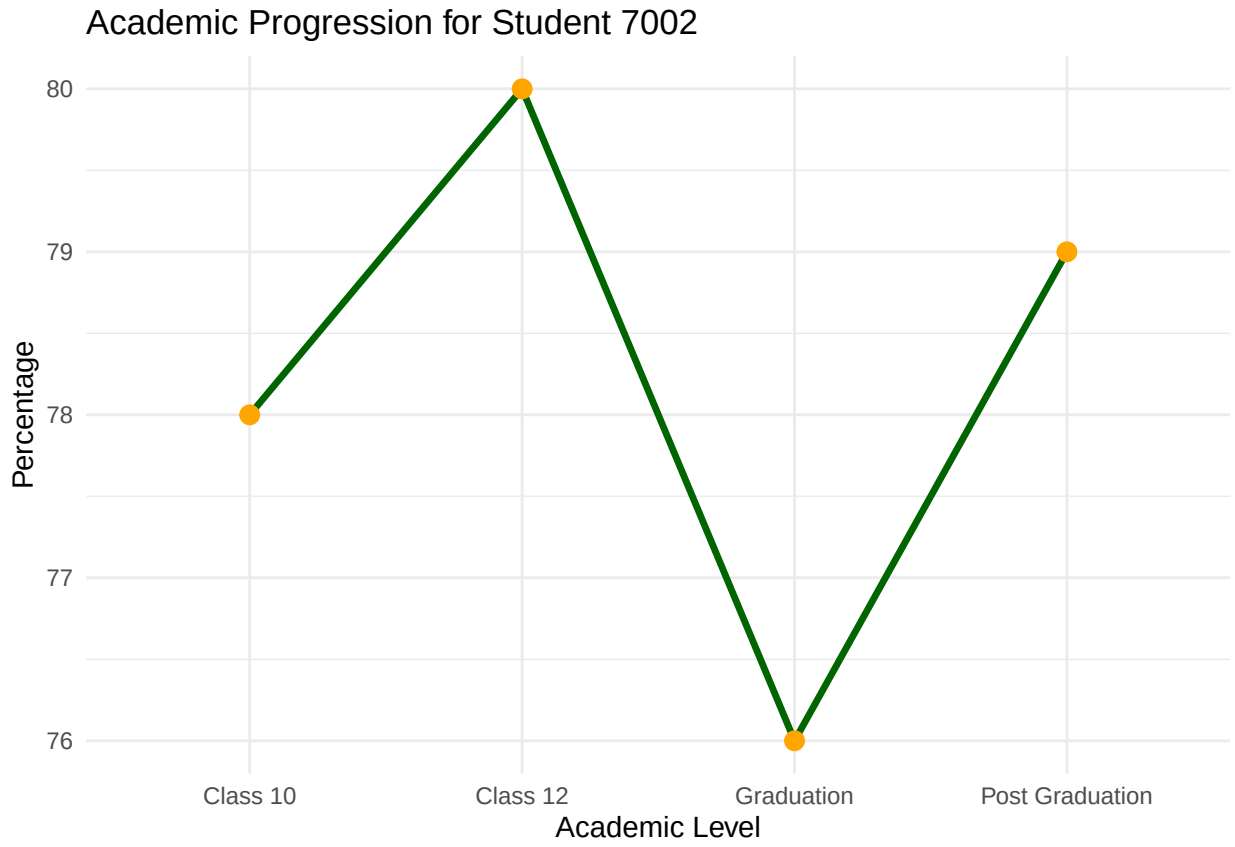
```r
#ACEDEMIC PROGRESSION USING MULTIPLE LINEAR REGRESSION
# Multiple Linear Regression: Predicting Post Graduation % based on past academics
model <- lm(post_graduation_percent ~ class_10_percent + class_12_percent + graduation_percent,
            data = Academic_Progression)

# Plot actual vs predicted
plot(Academic_Progression$post_graduation_percent, Academic_Progression$predicted_pg,
     xlab = "Actual Post Graduation %",
     ylab = "Predicted Post Graduation %",
     main = "Actual vs Predicted Post Graduation %",
     col = "blue", pch = 19)
abline(0, 1, col = "red", lwd = 2)
```




```
# Filter data for Student 7002
student2_progress <- Academic_Progression[Academic_Progression$student_id == 7002, ]

# Transform data to long format
student2_long <- pivot_longer(student2_progress,
                              cols = c(class_10_percent, class_12_percent, graduation_percent, post_gra
                              names_to = "Academic_Level",
                              values_to = "Percentage")

# Rename academic levels for display
student2_long$Academic_Level <- factor(student2_long$Academic_Level,
                                       levels = c("class_10_percent", "class_12_percent", "graduation_p
                                       labels = c("Class 10", "Class 12", "Graduation", "Post Graduatio

# Create line chart
ggplot(student2_long, aes(x = Academic_Level, y = Percentage, group = 1)) +
  geom_line(color = "darkgreen", size = 1.2) +
  geom_point(color = "orange", size = 3) +
  labs(
    title = "Academic Progression for Student 7002",
    x = "Academic Level",
    y = "Percentage"
  ) +
  theme_minimal()
```

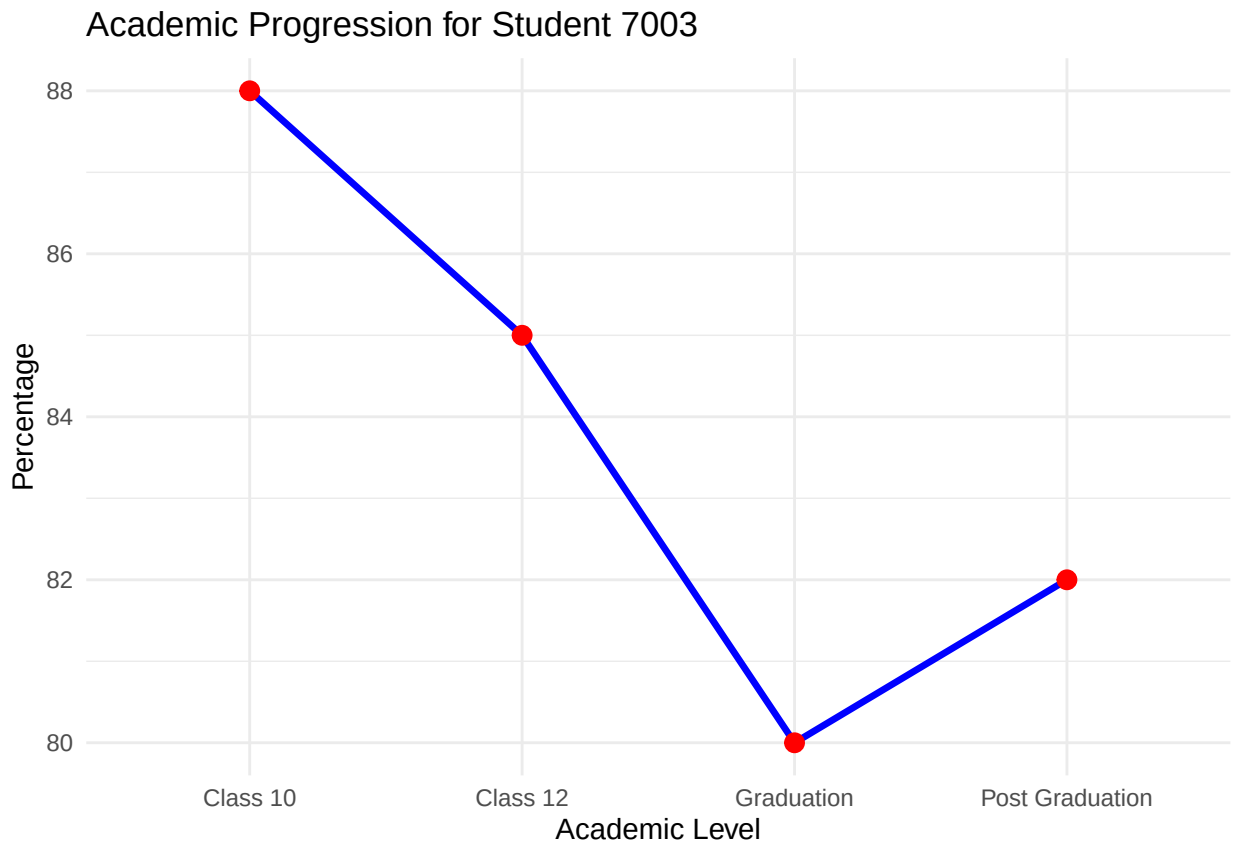
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
```

```
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```
#student 3 academic progression
# Load necessary library
library(ggplot2)
library(tidyr)
library(ggplot2)
library(tidyr)


# Filter data for Student 7003
student3_progress <- Academic_Progression[Academic_Progression$student_id == 7003, ]

# Transform data to long format
student3_long <- pivot_longer(student3_progress,
                              cols = c(class_10_percent, class_12_percent, graduation_percent, post_gra
                              names_to = "Academic_Level",
                              values_to = "Percentage")

# Rename academic levels for display
student3_long$Academic_Level <- factor(student3_long$Academic_Level,
                                       levels = c("class_10_percent", "class_12_percent", "graduation_p
                                       labels = c("Class 10", "Class 12", "Graduation", "Post Graduatio
```

```r
# Create line chart for Student 7003
ggplot(student3_long, aes(x = Academic_Level, y = Percentage, group = 1)) +
  geom_line(color = "blue", size = 1.2) +
  geom_point(color = "red", size = 3) +
  labs(
    title = "Academic Progression for Student 7003",
    x = "Academic Level",
    y = "Percentage"
  ) +
  theme_minimal()
```

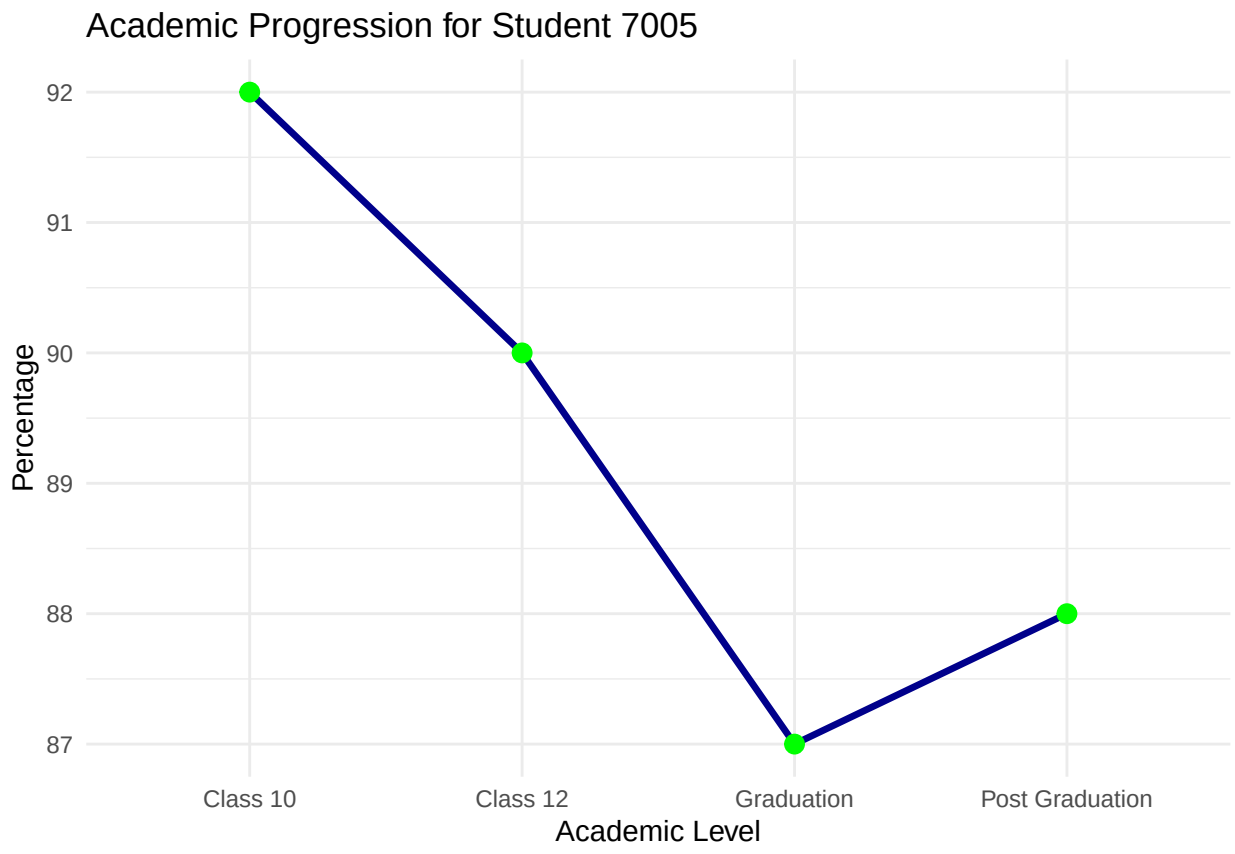



```r
#sudent 4
# Load necessary library
library(ggplot2)
library(tidyr)

# Filter data for Student 7004
student4_progress <- Academic_Progression[Academic_Progression$student_id == 7004, ]

# Transform data to long format
student4_long <- pivot_longer(student4_progress,
                              cols = c(class_10_percent, class_12_percent, graduation_percent, post_gra
                              names_to = "Academic_Level",
                              values_to = "Percentage")
```

```r
# Rename academic levels for display
student4_long$Academic_Level <- factor(student4_long$Academic_Level,
                                       levels = c("class_10_percent", "class_12_percent", "graduation_p
                                       labels = c("Class 10", "Class 12", "Graduation", "Post Graduatio

# Create line chart for Student 7004
ggplot(student4_long, aes(x = Academic_Level, y = Percentage, group = 1)) +
  geom_line(color = "purple", size = 1.2) +
  geom_point(color = "yellow", size = 3) +
  labs(
    title = "Academic Progression for Student 7004",
    x = "Academic Level",
    y = "Percentage"
  ) +
  theme_minimal()
```

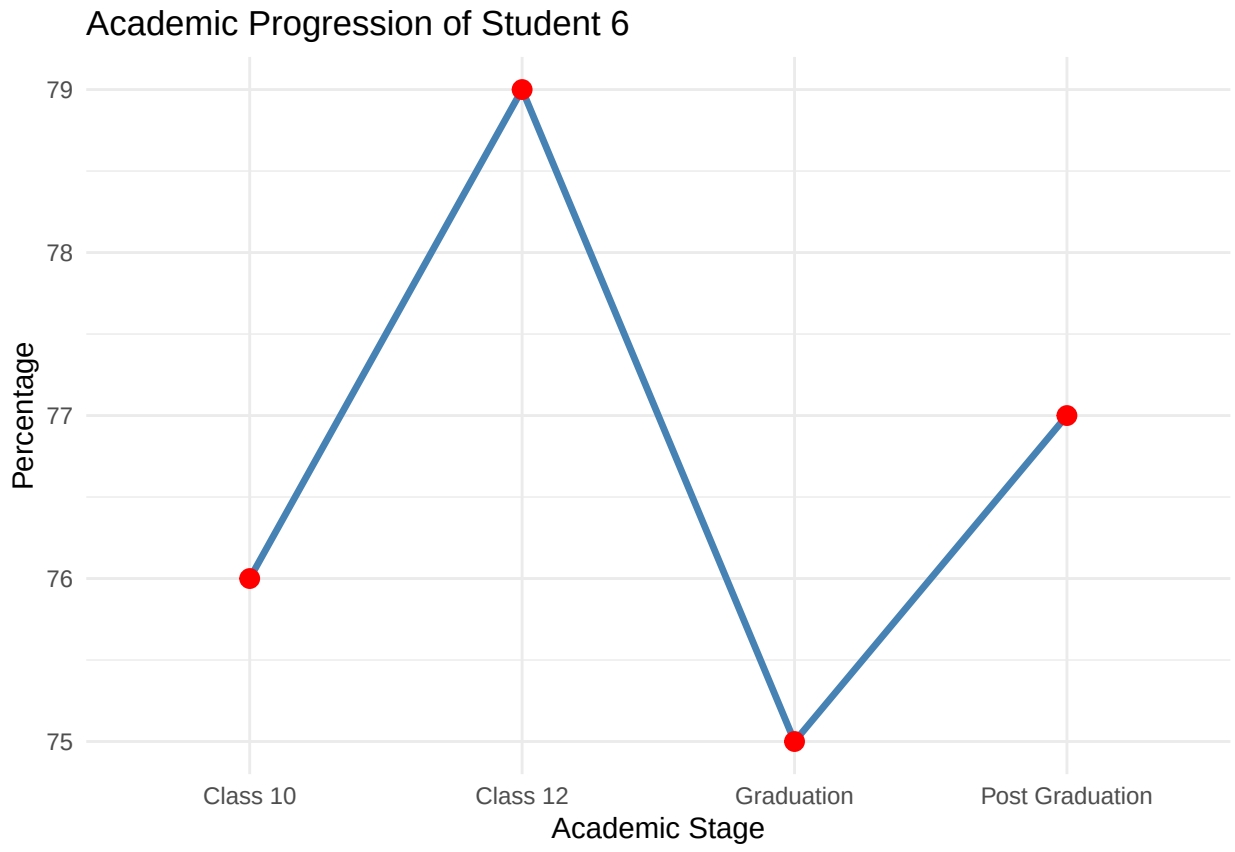

```r
#student 5 academic progression
# Load necessary library
library(ggplot2)
library(tidyr)

# Filter data for Student 7005
student5_progress <- Academic_Progression[Academic_Progression$student_id == 7005, ]

# Transform data to long format
```

```r
student5_long <- pivot_longer(student5_progress,
                              cols = c(class_10_percent, class_12_percent, graduation_percent, post_grad
                              names_to = "Academic_Level",
                              values_to = "Percentage")

# Rename academic levels for display
student5_long$Academic_Level <- factor(student5_long$Academic_Level,
                                       levels = c("class_10_percent", "class_12_percent", "graduation_p
                                       labels = c("Class 10", "Class 12", "Graduation", "Post Graduatio

#line chart for Student 7005
library(ggplot2)
library(tidyr)
ggplot(student5_long, aes(x = Academic_Level, y = Percentage, group = 1)) +
  geom_line(color = "darkblue", size = 1.2) +
  geom_point(color = "green", size = 3) +
  labs(
    title = "Academic Progression for Student 7005",
    x = "Academic Level",
    y = "Percentage"
  ) +
  theme_minimal()
```

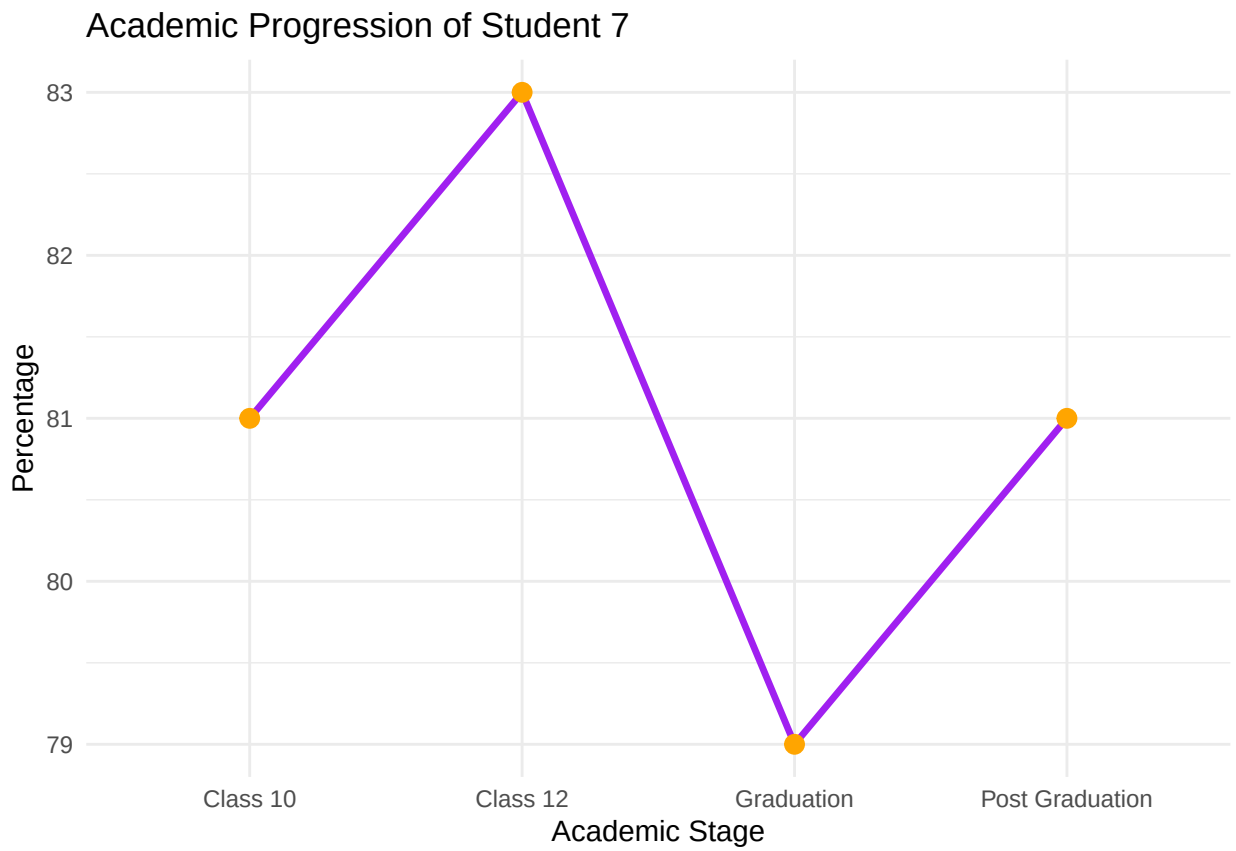



```r
#ACADEMIC PROGRESSION FOR STUDENT 6
# Load necessary libraries
```

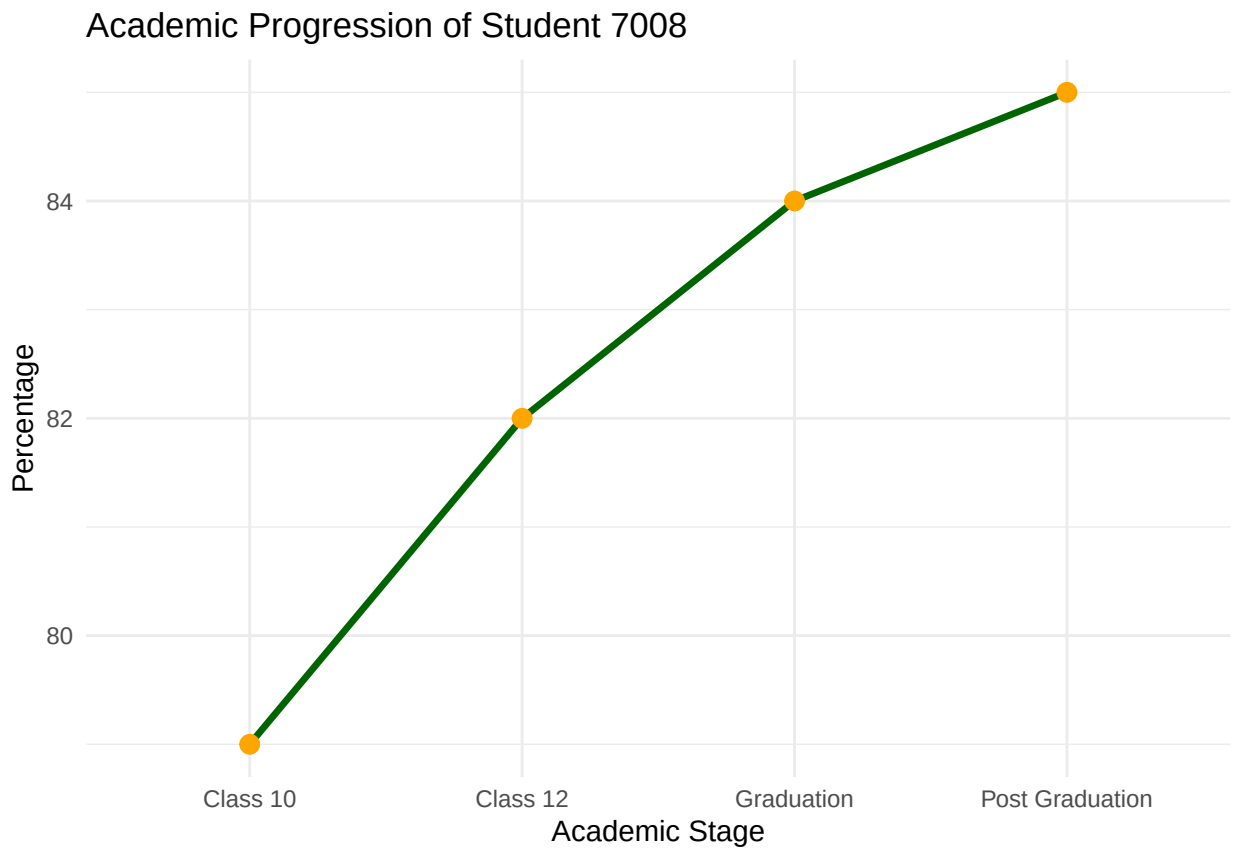
```r
library(ggplot2)
library(tidyr)

# Data for Student 6 from the Academic_Progression dataset
student6 <- Academic_Progression[Academic_Progression$student_id == 7006, ]

# Convert to long format for plotting
student6_long <- pivot_longer(student6,
                              cols = c(class_10_percent, class_12_percent, graduation_percent, post_gra
                              names_to = "Stage",
                              values_to = "Percentage")

# Rename academic stages for better labels
student6_long$Stage <- factor(student6_long$Stage,
                              levels = c("class_10_percent", "class_12_percent", "graduation_percent",
                              labels = c("Class 10", "Class 12", "Graduation", "Post Graduation"))

# Create the line chart
ggplot(student6_long, aes(x = Stage, y = Percentage, group = 1)) +
  geom_line(color = "steelblue", size = 1.2) +
  geom_point(color = "red", size = 3) +
  labs(title = "Academic Progression of Student 6",
       x = "Academic Stage",
       y = "Percentage") +
  theme_minimal()
```



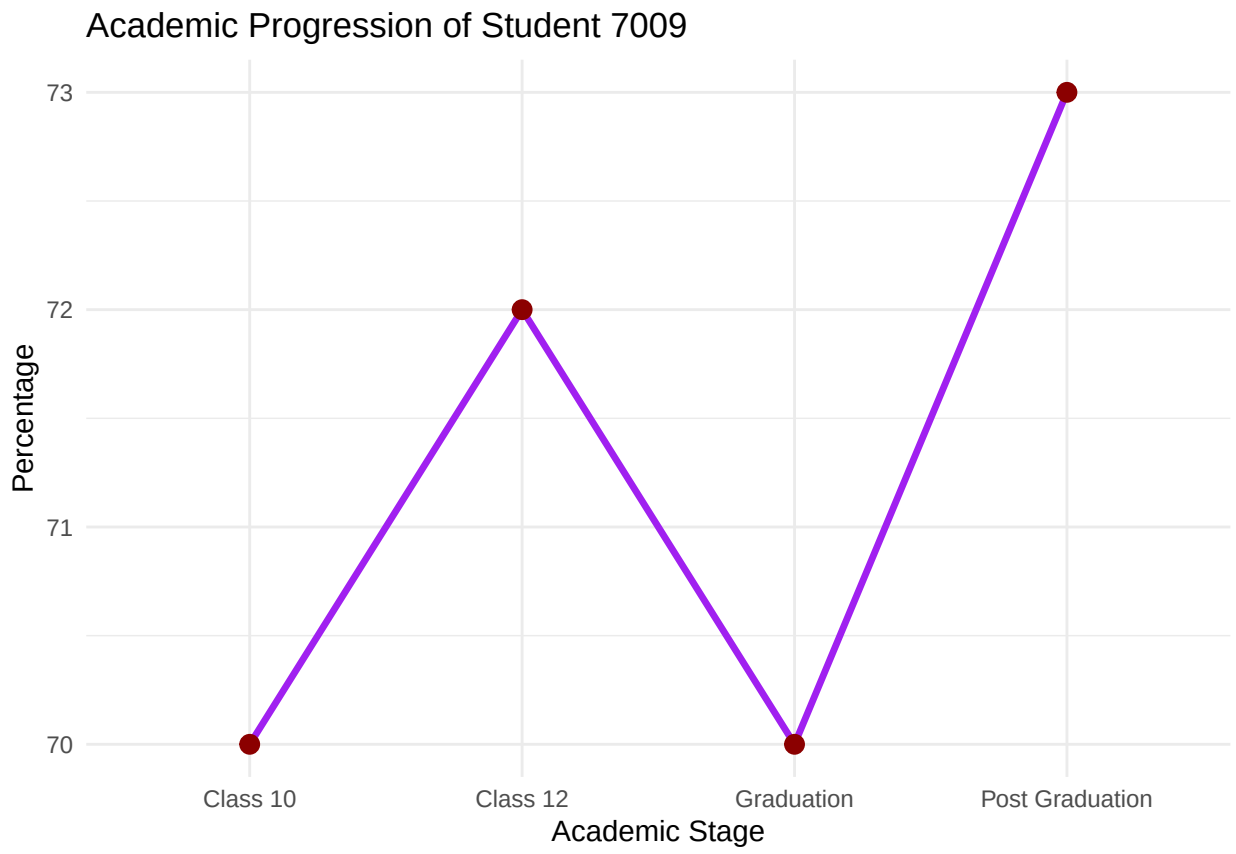
```r
#ACADEMIC PROGRESSION FOR STUDENT 7007
# Load necessary libraries
library(ggplot2)
library(tidyr)

# Data for Student 7 from the Academic_Progression dataset
student7 <- Academic_Progression[Academic_Progression$student_id == 7007, ]

# Convert to long format for plotting
student7_long <- pivot_longer(student7,
                              cols = c(class_10_percent, class_12_percent, graduation_percent, post_gra
                              names_to = "Stage",
                              values_to = "Percentage")

# Rename academic stages for better labels
student7_long$Stage <- factor(student7_long$Stage,
                              levels = c("class_10_percent", "class_12_percent", "graduation_percent",
                              labels = c("Class 10", "Class 12", "Graduation", "Post Graduation"))

# Create the line chart
ggplot(student7_long, aes(x = Stage, y = Percentage, group = 1)) +
  geom_line(color = "purple", size = 1.2) +
  geom_point(color = "orange", size = 3) +
  labs(title = "Academic Progression of Student 7",
       x = "Academic Stage",
       y = "Percentage") +
  theme_minimal()
```



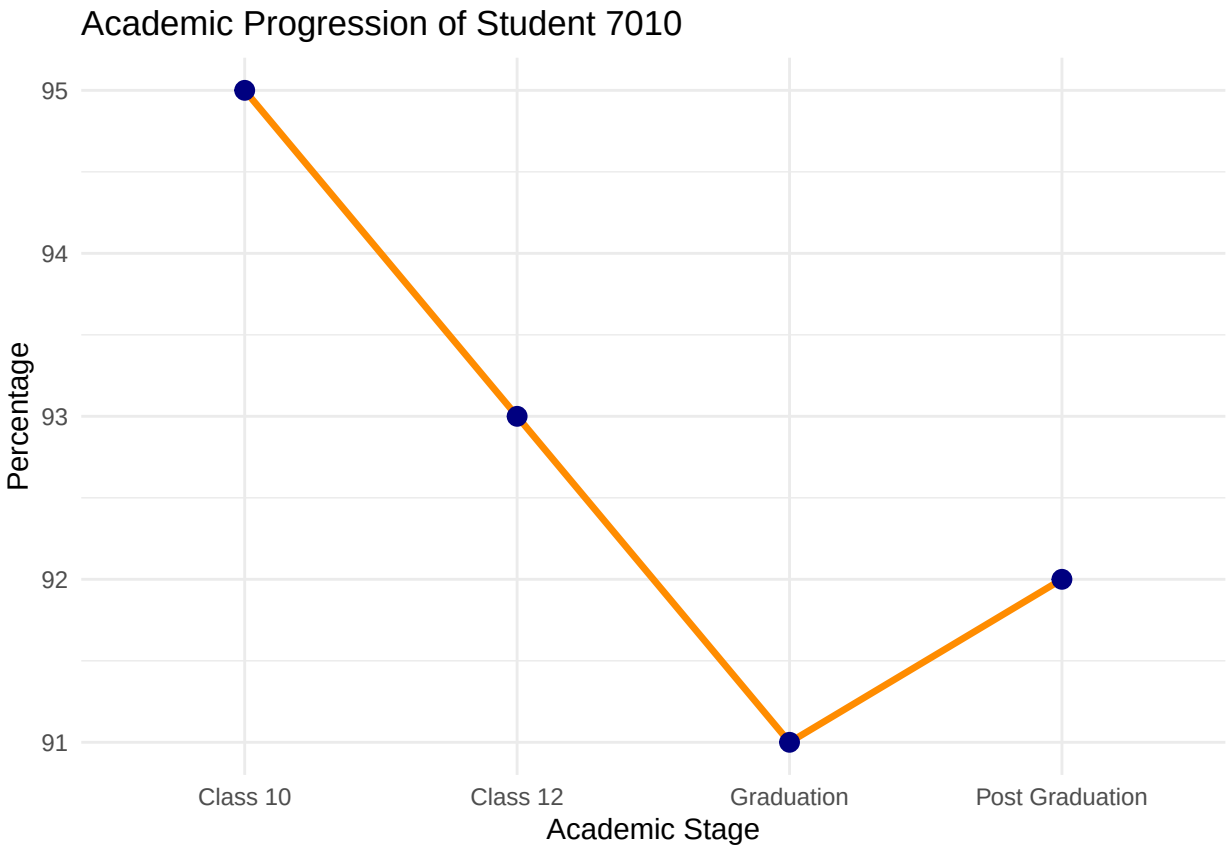

```
#ACADEMIC PROGRESSION FOR STUDENT 7008

# Data for Student 7008 from the Academic_Progression dataset
student8 <- Academic_Progression[Academic_Progression$student_id == 7008, ]

# Convert to long format for plotting
student8_long <- pivot_longer(student8,
                              cols = c(class_10_percent, class_12_percent, graduation_percent, post_gra
                              names_to = "Stage",
                              values_to = "Percentage")

# Rename academic stages for better labels
student8_long$Stage <- factor(student8_long$Stage,
                              levels = c("class_10_percent", "class_12_percent", "graduation_percent",
                              labels = c("Class 10", "Class 12", "Graduation", "Post Graduation"))

# Create the line chart
ggplot(student8_long, aes(x = Stage, y = Percentage, group = 1)) +
  geom_line(color = "darkgreen", size = 1.2) +
  geom_point(color = "orange", size = 3) +
  labs(title = "Academic Progression of Student 7008",
       x = "Academic Stage",
       y = "Percentage") +
  theme_minimal()
```




```
#ACADEMIC PROGRESSION FOR 7009
# Data for Student 7009 from the Academic_Progression dataset
student9 <- Academic_Progression[Academic_Progression$student_id == 7009, ]

# Convert to long format
student9_long <- pivot_longer(student9,
                              cols = c(class_10_percent, class_12_percent, graduation_percent, post_gra
                              names_to = "Stage",
                              values_to = "Percentage")

# Rename academic stages
student9_long$Stage <- factor(student9_long$Stage,
                              levels = c("class_10_percent", "class_12_percent", "graduation_percent",
                              labels = c("Class 10", "Class 12", "Graduation", "Post Graduation"))

# Plot line chart for Student 7009
ggplot(student9_long, aes(x = Stage, y = Percentage, group = 1)) +
  geom_line(color = "purple", size = 1.2) +
  geom_point(color = "darkred", size = 3) +
  labs(title = "Academic Progression of Student 7009",
       x = "Academic Stage",
       y = "Percentage") +
  theme_minimal()
```




```
#ACADEMIC PROGRESSION FOR STUDENT 10 7010
# Data for Student 7010 from the Academic_Progression dataset
student10 <- Academic_Progression[Academic_Progression$student_id == 7010, ]

# Convert to long format
student10_long <- pivot_longer(student10,
                               cols = c(class_10_percent, class_12_percent, graduation_percent, post_gra
                               names_to = "Stage",
                               values_to = "Percentage")

# Rename academic stages
student10_long$Stage <- factor(student10_long$Stage,
                               levels = c("class_10_percent", "class_12_percent", "graduation_percent",
                               labels = c("Class 10", "Class 12", "Graduation", "Post Graduation"))

# Plot line chart for Student 7010
ggplot(student10_long, aes(x = Stage, y = Percentage, group = 1)) +
  geom_line(color = "darkorange", size = 1.2) +
  geom_point(color = "navy", size = 3) +
  labs(title = "Academic Progression of Student 7010",
       x = "Academic Stage",
       y = "Percentage") +
  theme_minimal()
```


Academic Progression of Student 7010
Percentage
95
94
93
92
91
Class 10
Class 12
Graduation
Post Graduation
Academic Stage