Project 5

Justin Williams

2024-11-22

```
#import libraries and data
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflic
ts to become errors
```

```
library(dplyr)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(ggplot2)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(knitr)
library(rmarkdown)
hrdata_df <- read.csv("/Users/justinwilliams/Code/9050advresearch/Project 5/HRData.cs
v")
```

1. Run a simple regression analysis in which you predict PerfScoreID from EmpSatisfaction. Provide a summary of this analysis like what you would find in a journal article. Be sure to provide a table of results, a plot of the regression line, AND a written summary of the results in your response.

```
# Simple regression analysis
simple_model <- lm(PerfScoreID ~ EmpSatisfaction, data = hrdata_df)

# Model summary
simple_model_summary <- summary(simple_model)
```

```
# Model results data frame
simple_results_df <- data.frame(
  Predictor = rownames(coef(simple_model_summary)),
  Estimate = coef(simple_model_summary)[, "Estimate"],
  Std_Error = coef(simple_model_summary)[, "Std. Error"],
  t_value = coef(simple_model_summary)[, "t value"],
  P_value = coef(simple_model_summary)[, "Pr(>|t|)"]
)

print(simple_results_df)
```

```
##                       Predictor  Estimate  Std_Error  t_value      P_value
## (Intercept)         (Intercept) 2.2148700 0.13982121 15.84073 8.735181e-42
## EmpSatisfaction EmpSatisfaction 0.1960127 0.03499753  5.60076 4.719882e-08
```

```
#print results in a table
print("Simple Regression Results Data Frame:")
```

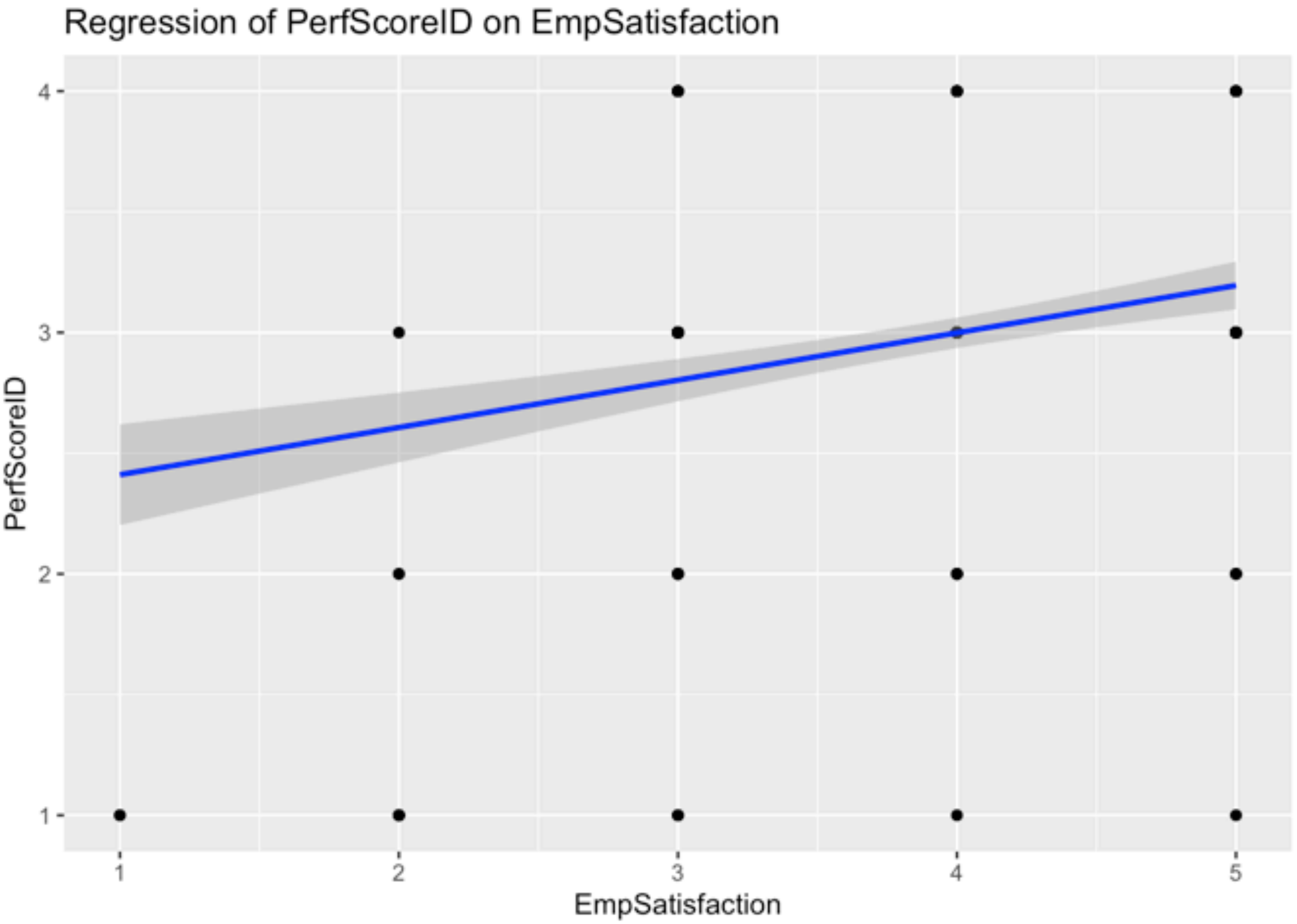
```
## [1] "Simple Regression Results Data Frame:"
```

```
kable(simple_results_df, format = "html") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
%>%
  row_spec(0, bold = TRUE, color = "white", background = "orange") %>%
  column_spec(1, bold = TRUE, color = "purple")
```

	Predictor	Estimate	Std_Error	t_value	P_value
(Intercept)	(Intercept)	2.2148700	0.1398212	15.84073	0
EmpSatisfaction	EmpSatisfaction	0.1960127	0.0349975	5.60076	0

```
# Plot of the regression line
ggplot(hrdata_df, aes(x = EmpSatisfaction, y = PerfScoreID)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Regression of PerfScoreID on EmpSatisfaction",
       x = "EmpSatisfaction",
       y = "PerfScoreID")
```

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



Summary

The regression analysis demonstrates that EmpSatisfaction significantly predicts PerfScoreID. The slope shows that for every one-unit increase in EmpSatisfaction, PerfScoreID increases by 0.196. The intercept suggests that when EmpSatisfaction is 0, the predicted PerfScoreID is 2.215. 9.22% of the variability in PerfScoreID is accounted for by EmpSatisfaction.

a. Is the regression weight statistically significant?

Yes, $p < 0.001$.

b. What is the regression equation?

PerfScoreID = 2.215 + 0.196 * EmpSatisfaction

c. Provide an interpretation of the slope in terms of the variables involved.

For every one-unit increase in EmpSatisfaction, PerfScoreID increases by 0.196.

d. Interpret the y-intercept.

When EmpSatisfaction is 0, the predicted PerfScoreID is 2.215.

e. What is the predicted PerfScoreID for someone with an average level of EmpSatisfaction?

The predicted PerfScoreID would be calculated as Y = 2.215 + 0.196 * the mean EmpSatisfaction.

f. What percentage of variance in PerfScoreID is explained by EmpSatisfaction?

EmpSatisfaction explains 9.22% of the variance in PerfScoreID.

2. Run a simple regression analysis in which you predict Absences from EngagementSurvey. Provide a summary of this analysis like what you would find in a journal article. Be sure to provide a table of results, a plot of the regression line, AND a written summary of the results in your response.

```
absences_model <- lm(Absences ~ EngagementSurvey, data = hrdata_df)

# Model summary
absences_model_summary <- summary(absences_model)

# Model results data frame
absences_results_df <- data.frame(
  Predictor = rownames(coef(absences_model_summary)),
  Estimate = coef(absences_model_summary)[, "Estimate"],
  Std_Error = coef(absences_model_summary)[, "Std. Error"],
  t_value = coef(absences_model_summary)[, "t value"],
  P_value = coef(absences_model_summary)[, "Pr(>|t|)"]
)

print(absences_results_df)
```

```
##                         Predictor    Estimate Std_Error    t_value      P_value
## (Intercept)           (Intercept) 10.50501508 1.7638177  5.9558395 7.032763e-09
## EngagementSurvey EngagementSurvey -0.06498126 0.4214634 -0.1541801 8.775684e-01
```

```
#print results in a table
print("Absences Regression Results Data Frame:")
```

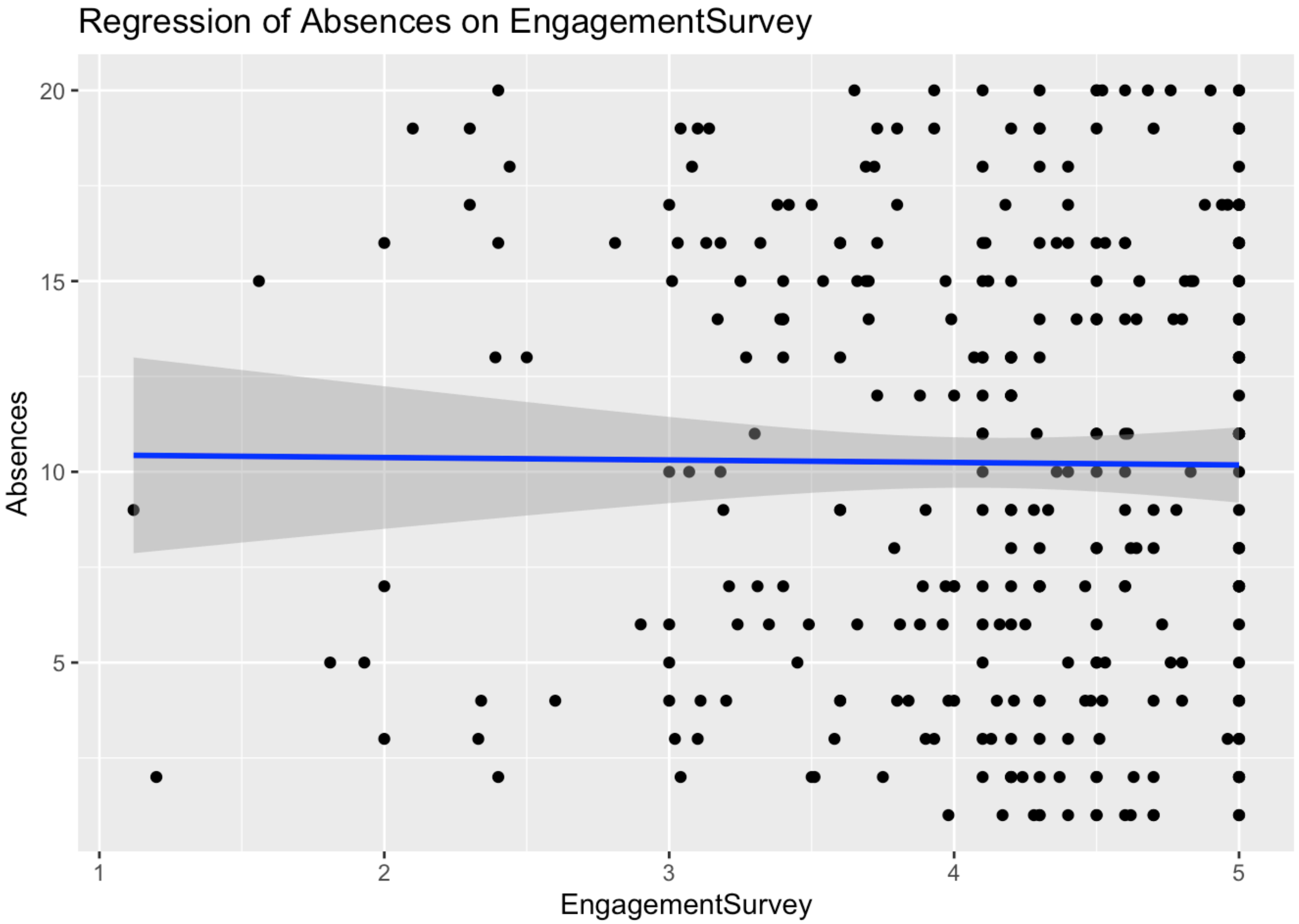
```
## [1] "Absences Regression Results Data Frame:"
```

```
kable(absences_results_df, format = "html") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
%>%
  row_spec(0, bold = TRUE, color = "white", background = "orange") %>%
  column_spec(1, bold = TRUE, color = "purple")
```

	Predictor	Estimate	Std_Error	t_value	P_value
(Intercept)	(Intercept)	10.5050151	1.7638177	5.9558395	0.0000000
EngagementSurvey	EngagementSurvey	-0.0649813	0.4214634	-0.1541801	0.8775684

```
# Plot of the regression line
ggplot(hrdata_df, aes(x = EngagementSurvey, y = Absences)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Regression of Absences on EngagementSurvey",
       x = "EngagementSurvey",
       y = "Absences")
```

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



Summary

The regression analysis aimed to predict Absences based on EngagementSurvey scores. Results indicated that the regression weight for EngagementSurvey was not statistically significant. This suggests that EngagementSurvey scores do not significantly predict Absences.

The model explained only 0.01% of the variance in Absences (R^2=0.0001), indicating that the predictive utility of EngagementSurvey is negligible. The y-intercept represents the predicted Absenceswhen EngagementSurvey is zero, a scenario unlikely in real life.

a. Is the regression weight statistically significant?

No, p = 0.878.

b. What is the regression equation?

Absences = 10.50502 - 0.06498 * EngagementSurvey

c. Provide an interpretation of the slope

For every one-unit increase in EngagementSurvey, the predicted Absences decrease by 0.065 units. This change is not statistically significant.

d.Interpret the y-intercept

When EngagementSurvey is zero, the predicted number of Absences is 10.505. This value may not be meaningful if EngagementSurvey cannot logically reach zero.

e. Predicted Absences for someone with an average EngagementSurvey

For an average EngagementSurvey score, the predicted number of Absences is 10.237.

f. Percentage of variance explained by EngagementSurvey

0.01% of the variance in Absences is explained by EngagementSurvey, showing no meaningful explanatory power.

3. Run a simple regression analysis in which you predict PerfScoreID from Department. Provide a summary of this

analysis like what you would find in a journal article. Be sure to provide a table of results, a plot of the regression line, AND a written summary of the results in your response.

```
# Simple regression analysis predicting PerfScoreID from Department
department_model <- lm(PerfScoreID ~ Department, data = hrdata_df)

# Model summary
department_model_summary <- summary(department_model)

# Model results data frame
department_results_df <- data.frame(
  Predictor = rownames(coef(department_model_summary)),
  Estimate = coef(department_model_summary)[, "Estimate"],
  Std_Error = coef(department_model_summary)[, "Std. Error"],
  t_value = coef(department_model_summary)[, "t value"],
  P_value = coef(department_model_summary)[, "Pr(>|t|)"]
)

print(department_results_df)
```

```
##                                                          Predictor      Estimate
## (Intercept)                                            (Intercept)  3.000000e+00
## DepartmentExecutive Office        DepartmentExecutive Office -5.035592e-16
## DepartmentIT/IS                                    DepartmentIT/IS  6.000000e-02
## DepartmentProduction             DepartmentProduction       -2.870813e-02
## DepartmentSales                                    DepartmentSales -1.612903e-01
## DepartmentSoftware Engineering DepartmentSoftware Engineering  9.090909e-02
##                                Std_Error       t_value      P_value
## (Intercept)                    0.1962772  1.528451e+01 1.494910e-39
## DepartmentExecutive Office     0.6206829 -8.112987e-16 1.000000e+00
## DepartmentIT/IS                0.2132116  2.814106e-01 7.785863e-01
## DepartmentProduction           0.2004587 -1.432122e-01 8.862172e-01
## DepartmentSales                0.2229559 -7.234181e-01 4.699775e-01
## DepartmentSoftware Engineering 0.2646601  3.434938e-01 7.314637e-01
```

```
#print results in a table
print("Department Regression Results Data Frame:")
```

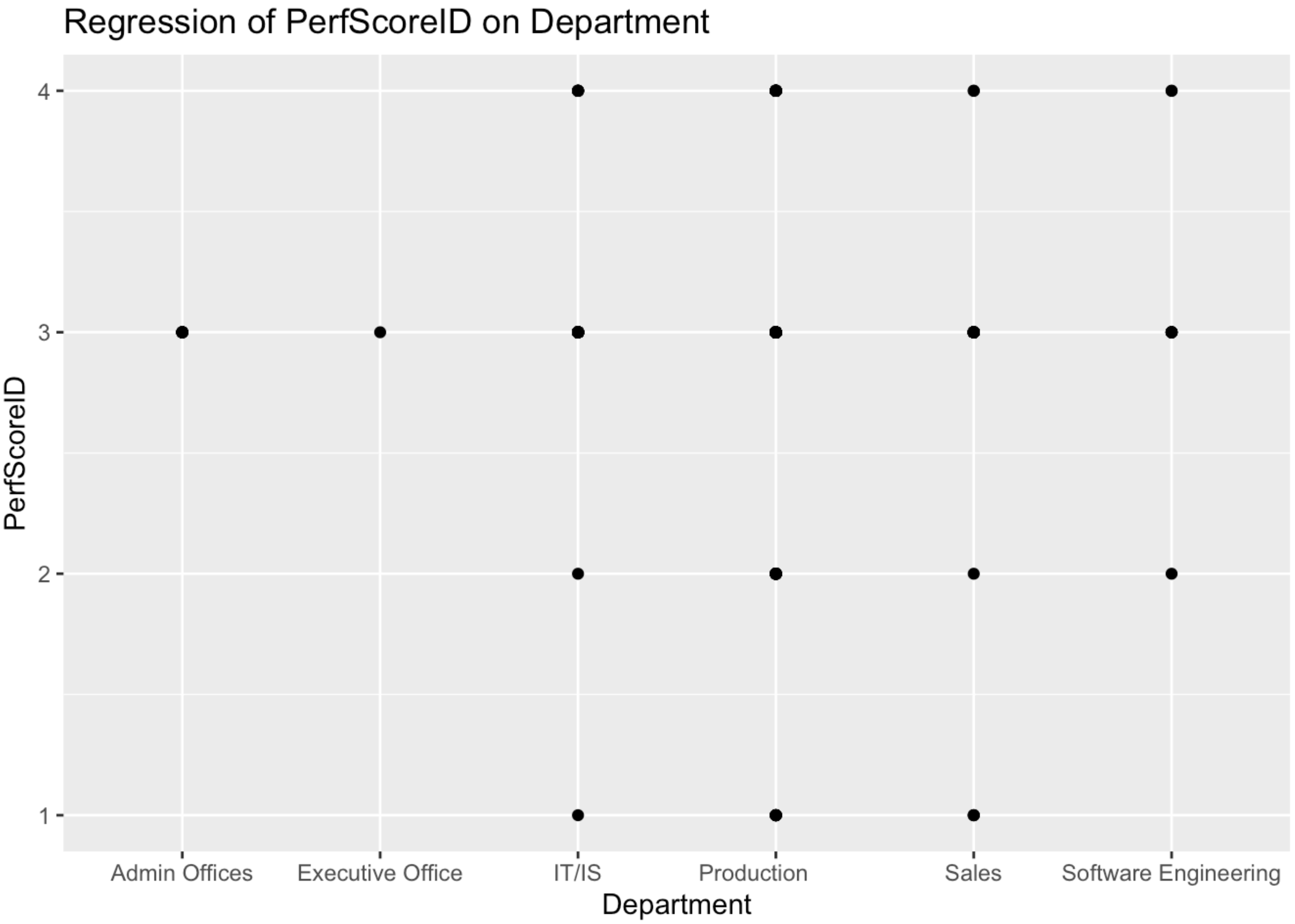
```
## [1] "Department Regression Results Data Frame:"
```

```
kable(department_results_df, format = "html") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
%>%
  row_spec(0, bold = TRUE, color = "white", background = "orange") %>%
  column_spec(1, bold = TRUE, color = "purple")
```

	Predictor	Estimate	Std_Error	t_value	P_value
(Intercept)	(Intercept)	3.0000000	0.1962772	15.2845071	0.0000000
DepartmentExecutive Office	DepartmentExecutive Office	0.0000000	0.6206829	0.0000000	1.0000000
DepartmentIT/IS	DepartmentIT/IS	0.0600000	0.2132116	0.2814106	0.7785863
DepartmentProduction	DepartmentProduction	-0.0287081	0.2004587	-0.1432122	0.8862172
DepartmentSales	DepartmentSales	-0.1612903	0.2229559	-0.7234181	0.4699775
DepartmentSoftware Engineering	DepartmentSoftware Engineering	0.0909091	0.2646601	0.3434938	0.7314637

```
# Plot of the regression line
ggplot(hrdata_df, aes(x = Department, y = PerfScoreID)) +
  geom_point() +
  geom_smooth(method = "lm", col = "purple") +
  labs(title = "Regression of PerfScoreID on Department",
       x = "Department",
       y = "PerfScoreID")
```

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



Summary

The plot shows PerfScoreID by Department with a regression line. This shows minimal variation across departments, consistent with the low R² value.

The overall model was not statistically significant as the R² was .0102, explaining only 1.02% of the variance in PerfScoreID. None of the departmental coefficients were statistically significant as their p values were greater than .05.

The regression sum of squares was small compared to the total sum of squares, indicating poor model fit.

The extremely low coefficients and non-significant p-values shows a lack of relationship between predictors and the outcome variable, reinforcing the need for more meaningful predictors to improve R².

a. Is the regression weight statistically significant?

The regression weights for all departments were not statistically significant.

b. What is the regression equation?

PerfScoreID = 3.00 + (-5.035592e - 16 · Department)

c. Provide an interpreatation of the slope in terms of the variables involved.

For each unit increase in the department coding, PerfScoreID is predicted to change by -5.035592e-16 units. This value indicates no meaningful relationship.

d. Interpret the y-intercept.

When department coding is zero, the predicted PerfScoreID is 3.00. This value is not meaningful as department coding cannot be zero.

e. What is the predicted PerfScoreID for someone from each of the Departments?

- Executive Office: 3.00
- IT/IS: 3.06
- Production: 2.97
- Sales: 2.84
- Software Engineering: 3.09

f. What percentage of variance in PerfScoreID is explained by Department?

Department explains only 1.02% of the variance in PerfScoreID, showing weak predictive power of department on performance scores.