Understanding Job Satisfaction Among the American Workforce

Misbah Arshad (other contriubtors to original project: Panji Al Alam, Colley Buwana, Hazna Faiza, Sag

Abstract

This study examines job satisfaction within the U.S. workforce, using survey data from the IPUMS Higher Ed dataset. The objective is to identify key demographic and professional factors that serve as predictors of job satisfaction in STEM fields. Job satisfaction was assessed through survey responses on a four-point Likert scale, ranging from very satisfied to very dissatisfied. Employing logistic regression and cross-validation this research aims to uncover the main drivers of job satisfaction, offering insights into improving workplace experiences within STEM professions.

Methods

Data The study utilizes the IPUMS Higher Ed dataset with samples from 2013, which provides detailed information on the science and engineering workforce in the U.S. The dataset includes 98,051 observations across 56 variables.

Demographic variables include age, gender, and race; lifestyle variables encompass factors such as the number of children; and job-related characteristics include hours worked, benefits available, and company size.

The data was cleaned by changing the observations indicating no information or no answer (coded as 97, 98, or 99) as non-available. To make the exploratory data analysis understandable, the variables "Gender, Majors, and Degrees" were edited from number-coded data to categorical. Race and Job Satisfaction were altered from four-level categorical observations into binary variables.

A comprehensive list of variables is documented in the metadata.

Research Question The primary focus of this study is to identify the most significant predictors of job satisfaction in STEM professions.

Analytical Approach The analysis began with exploratory data analysis (EDA) to investigate demographic trends, salary distributions, and overall satisfaction levels. Logistic regression was then used to assess the relationship between various factors and job satisfaction. To refine the model and identify the most relevant predictors, LASSO regression and cross-validation techniques were applied.

Cleaning the Data Map Major and Job Codes to Actual Descriptions

```
major_mapping <- tibble(
  id = c(198895, 226395, 298895, 318730, 338785, 398895, 419295, 429295, 438995, 449995,
         459395, 527250, 537260, 547280, 567350, 587995, 611995, 699995, 719995, 799995),
  major = c("Computer and mathematical sciences", "Biological sciences",
            "Other biological, agricultural, environmental life sciences", "Chemistry, except biochemis
            "Physics and astronomy", "Other physical and related sciences", "Economics",
            "Political and related sciences", "Psychology", "Sociology and anthropology",
            "Other social sciences", "Chemical engineering", "Civil engineering",
            "Electrical, electronics and communications engineering", "Mechanical engineering",
            "Other engineering", "Health-related fields", "Other science and engineering-related",
            "Management and administration", "Other non-science and engineering")
```

```
)

job_mapping <- tibble(
  id = c(182965, 192895, 222205, 282885, 293995, 311930, 333305, 382995, 393995, 412320,
         432360, 482995, 483995, 505005, 520850, 530860, 540890, 560940, 582800, 611995,
         621995, 631995, 651995, 711410, 711995, 735995, 799995),
  job = c("Postsecondary teachers-Computer and math sciences", "Computer scientists and mathematicians"
          "Biological and medical scientists", "Postsecondary teachers-Life related sciences",
          "Other life and related scientists", "Chemists, except biochemists", "Physicists and astronom
          "Postsecondary teachers-Physical and related sciences", "Other physical and related scientist
          "Economists", "Psychologists", "Postsecondary teachers-Social and related sciences",
          "Other social scientists", "Other engineers", "Chemical engineers", "Civil engineers",
          "Electrical or computer hardware engineers", "Mechanical engineers",
          "Postsecondary teachers - engineering", "Health-related occupations",
          "Science and engineering managers", "Science and engineering pre-college teachers",
          "Science and engineering pre-college teachers", "Top and mid-level managers, executives, admi
          "Other management related occupations", "Non-science and engineering pre-college and post-sec
          "Other Non-science and engineering occupations")
)
```

Replacement Function to Clean Data

```
replace_codes <- function(col, major_mapping, job_mapping) {
  if (is.numeric(col)) {
    col <- ifelse(col %in% major_mapping$id, major_mapping$major[match(col, major_mapping$id)], col)
    col <- ifelse(col %in% job_mapping$id, job_mapping$job[match(col, job_mapping$id)], col)
  }
  return(col)
}
```

Create clean dataset

```
job_sat_cleaned <- job_sat %>%
  dplyr::select(-PERSONID, -PTWTFT, -NRREA, -WTREASN, -FSDED, -FSDK, -FSDOD, -FSDOE, -FSHHS, -FSNIH, -FS
  filter(SALARY != 9999998) %>%
  mutate(
    GENDER = if_else(GENDER == 2, "Male", "Female"),
    RACETH = if_else(RACETH == 2, "White", "Non-White"),
    JOBSATIS = if_else(JOBSATIS %in% c(1, 2), 1, 0)
  ) %>%
  mutate(across(everything(), ~ replace_codes(., major_mapping, job_mapping)))


check <- job_sat_cleaned |>
  summarize(across(everything(), ~ sum(is.na(.))))

write_csv(job_sat_cleaned, "job_sat_cleaned.csv")
job_sat_cleaned <- read.csv("job_sat_cleaned.csv")
```

Exploratory Data Analysis: Demographics

Race and Gender In the dataset, there are 56,448 male participants (representing 57.7% of the total sample) and 41,603 female participants (representing 42.3% of the total sample). Additionally, 60,676 participants identify as white (making up 61.8% of the total sample), while 37,375 participants identify as non-white (representing 38.2% of the total sample).

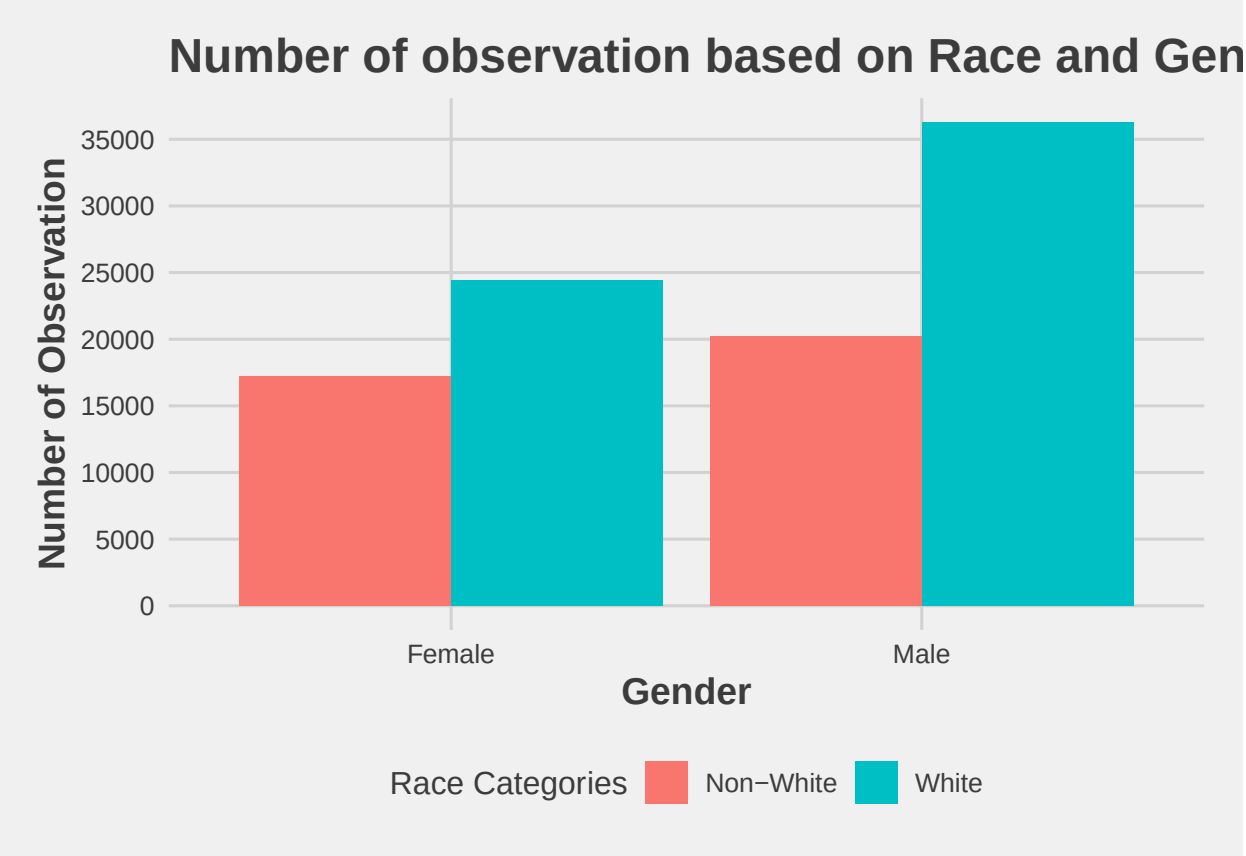
```r
gender_table <- table(job_sat_cleaned$GENDER)
gender_table
```

```
##
## Female   Male
##  41603  56448
```

```r
race_table <- table(job_sat_cleaned$RACETH)
race_table
```

```
##
## Non-White     White
##     37375     60676
```

```r
racsex_graph <- ggplot(data = job_sat_cleaned, aes(x = GENDER, fill = RACETH)) +
  labs(title = "Number of observation based on Race and Gender",
       x = "Gender",
       y = "Number of Observation",
       fill = "Race Categories") +
  scale_y_continuous(breaks = seq(0, 50000, by = 5000)) +
  geom_bar(position = "dodge") +
  theme_fivethirtyeight() +
  theme(axis.title = element_text(size = 14, face = "bold"))

racsex_graph
```



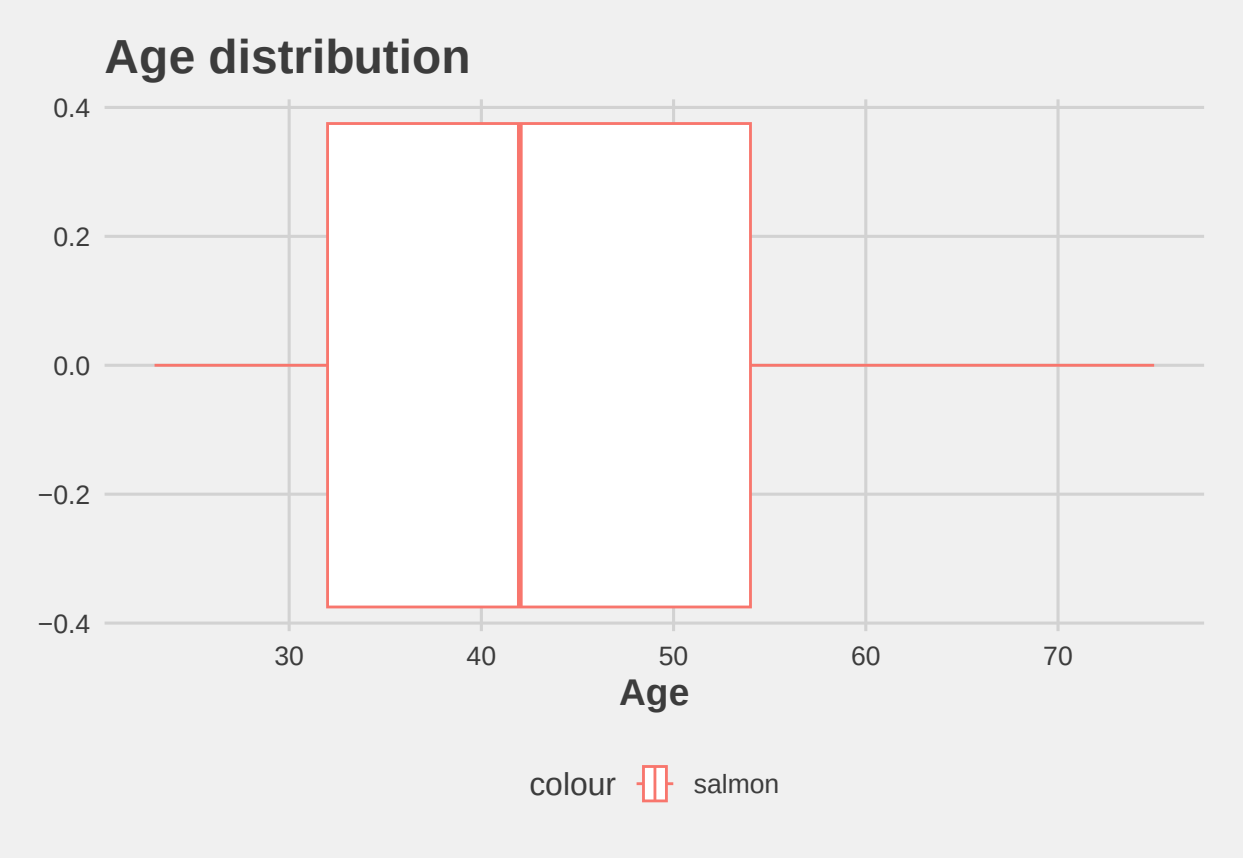

```r
ggsave("3. Outputs/Number of Observation based on Gender and Race.png",
```

```
       racsex_graph)
```

```
## Saving 6.5 x 4.5 in image
```

Age The average age of the total respondents was around 43 years old. Additionally, the spread of data for the age of respondents is similar, although female respondents seem to make up a younger cohort relative to the male respondents. This aligns with the overall systemic change in STEM jobs recruiting more women than before – it is expected for there to be lower representation of older female employees.

```
age_distribution <- job_sat_cleaned |>
  ggplot(aes(x = AGE)) + geom_boxplot(aes(color = "salmon")) + theme_classic() +
  labs(title = "Age distribution",
       x = "Age") +
  theme_fivethirtyeight() +
  theme(axis.title = element_text(size = 14, face = "bold"))

print(age_distribution)
```




```
ggsave("3. Outputs/Number of Observations: Age.png", age_distribution)
```

```
## Saving 6.5 x 4.5 in image
```

Age by Gender

```
# Calculate average age for each gender
avg_age <- job_sat_cleaned %>%
  group_by(GENDER) %>%
  summarise(mean_age = mean(AGE, na.rm = TRUE))
```

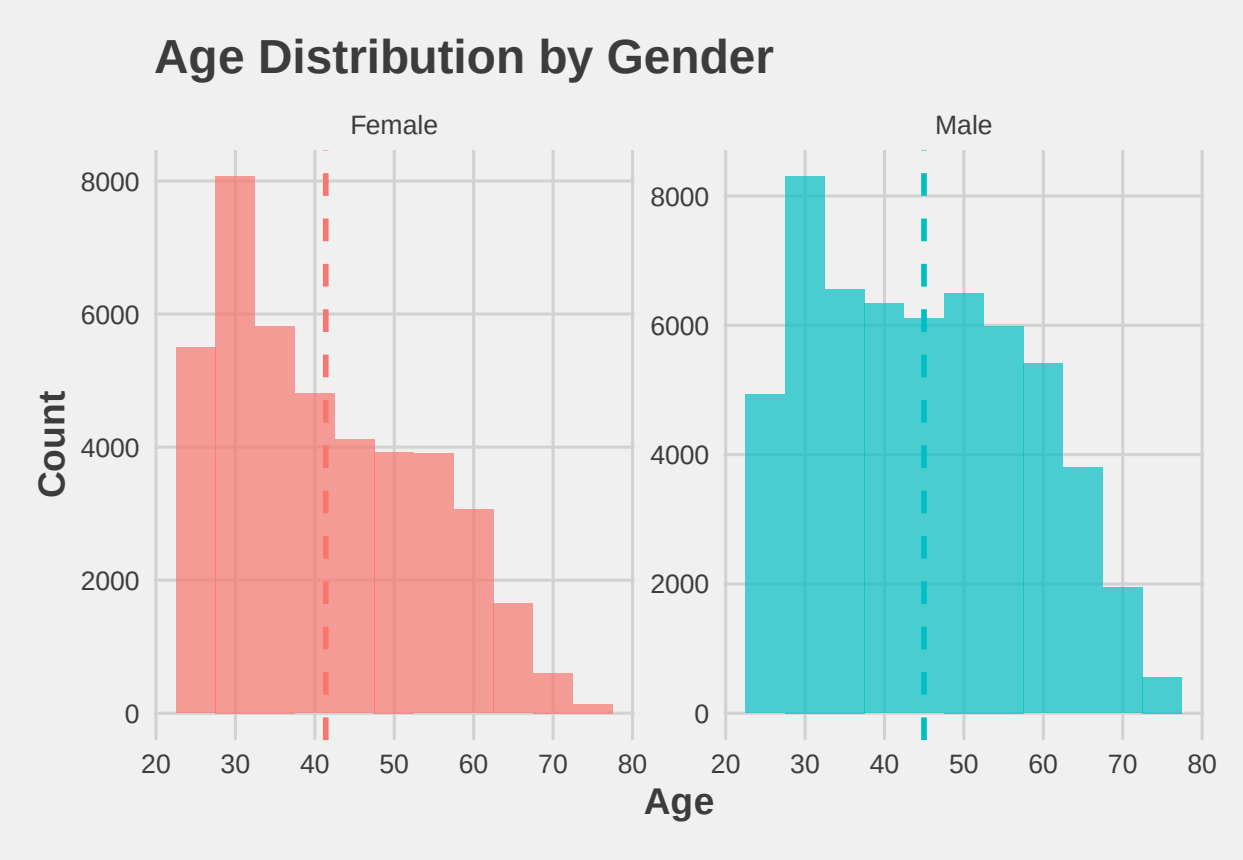
```
avg_age
```

```
## # A tibble: 2 x 2
##   GENDER mean_age
##   <chr>     <dbl>
## 1 Female     41.4
## 2 Male       45.0
```

```
# Plot
agesex_graph <- ggplot(job_sat_cleaned, aes(x = AGE, fill = GENDER)) +
  geom_histogram(binwidth = 5, alpha = 0.7, position = "identity") +
  facet_wrap(~ GENDER, scales = "free_y") +
  geom_vline(data = avg_age, aes(xintercept = mean_age, color = GENDER),
             linetype = "dashed", size = 1) +
  labs(
    title = "Age Distribution by Gender",
    x = "Age",
    y = "Count",
    fill = "Gender",
    color = "Average Age"
  ) + theme_fivethirtyeight() +
  theme(axis.title = element_text(size = 14, face = "bold")) +
  theme(legend.position = "none")
```

```
## 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.
```

```
print(agesex_graph)
```

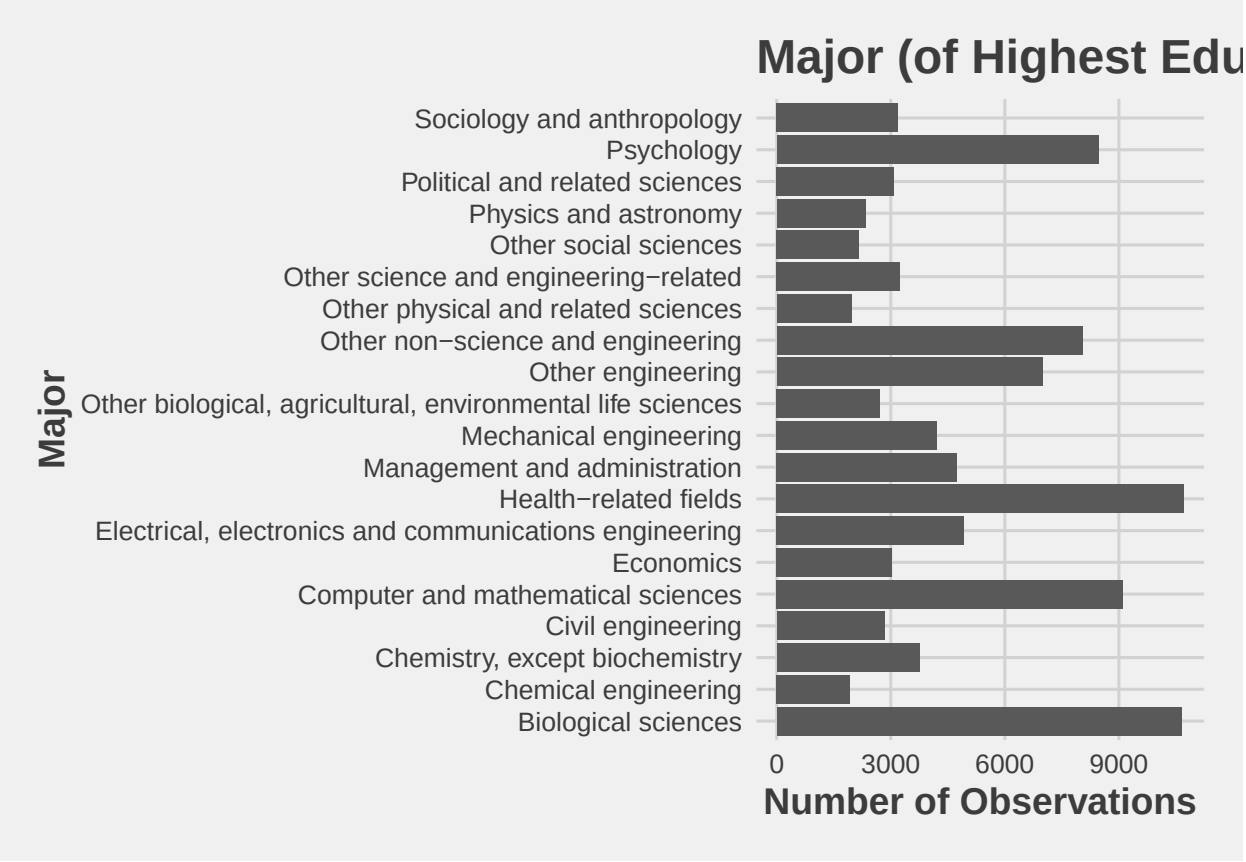



```
ggsave("3. Outputs/Age by Sex.png", agesex_graph)
```

```
## Saving 6.5 x 4.5 in image
```

Education - Major As expected, most respondents have the highest education in STEM fields, while majors associated with humanities (social sciences, economics) had a lower representation among all respondents. More than 18,000 respondents fall into Health-related fields and biological sciences.

```
edmajor_graph <- ggplot(job_sat_cleaned, aes(x = NDGMED)) +
  geom_bar() +
  labs(
    title = "Major (of Highest Education)",
    x = "Major",
    y = "Number of Observations"
  ) +
  coord_flip() +
  theme_fivethirtyeight() +
  theme(axis.title = element_text(size = 14, face = "bold"))

print(edmajor_graph)
```




```
ggsave("3. Outputs/Major of Highest Education.png",
       edmajor_graph)
```

```
## Saving 6.5 x 4.5 in image
```

Exploratory Data Analysis: Salary Distribution

Clean the data for degrees to make it interpretable.

```
# Mutating degree
job_sat_cleaned <- job_sat_cleaned |>
  mutate(
    DGRDG = case_when(
      DGRDG == 1 ~ "Bachelor",
      DGRDG == 2 ~ "Master",
      DGRDG == 3 ~ "Doctorate",
      DGRDG == 4 ~ "Professional"),
    DGRDG = factor(DGRDG,
                   levels = c("Bachelor", "Master", "Doctorate", "Professional"))
  )
```

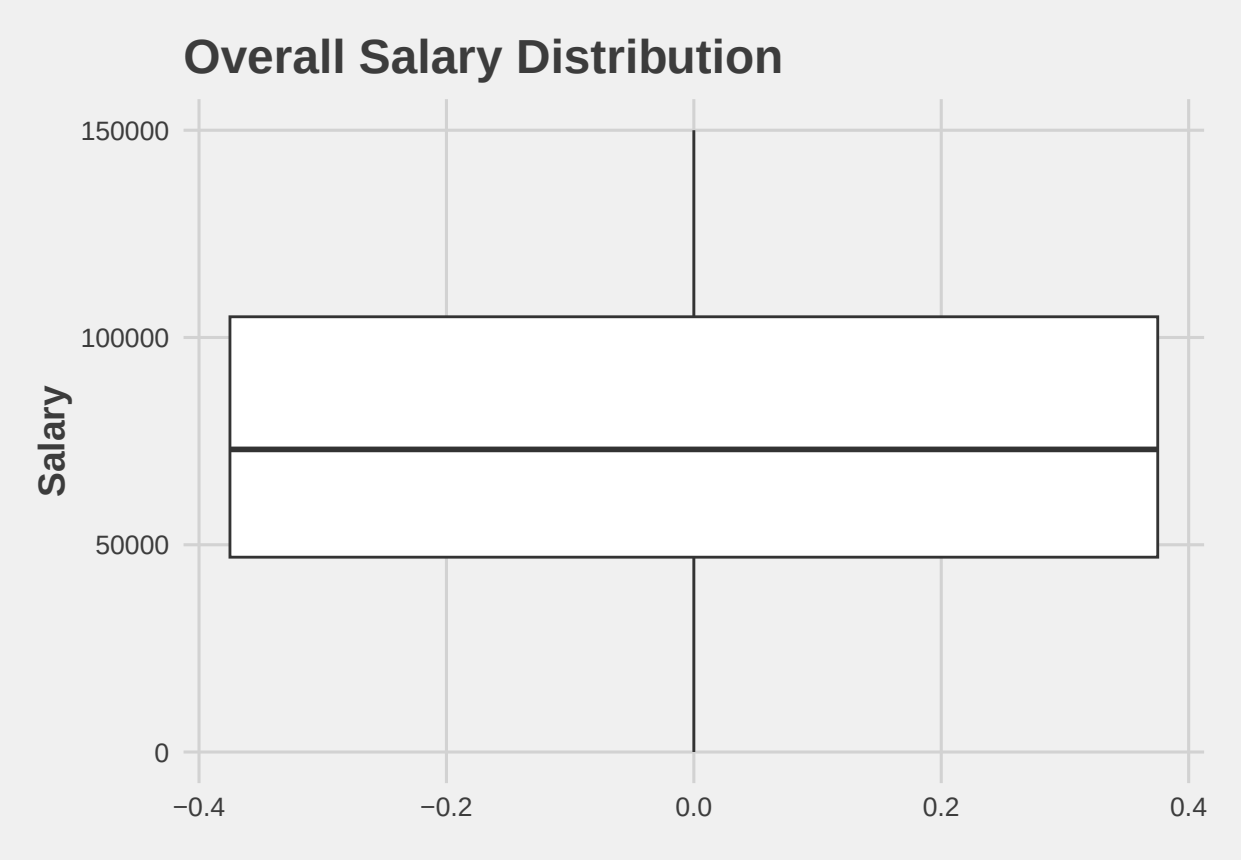
Salary Distribution The average salary across respondents is USD 77,294.

```
# Salary summary table
salary_summary <- summary(job_sat_cleaned$SALARY)
salary_summary
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##       0   47000   73000   77295  105000  150000
```

```
salary_distribution <- job_sat_cleaned |>
  ggplot(aes(x = SALARY)) +
  geom_boxplot() +
  theme_classic() +
  labs(title = "Overall Salary Distribution",
       x = "Salary") +
  theme_fivethirtyeight() +
  theme(axis.title = element_text(size = 14, face = "bold"))+
  coord_flip()

print(salary_distribution)
```




```
ggsave("3. Outputs/Salary_graph.png", salary_distribution)
```

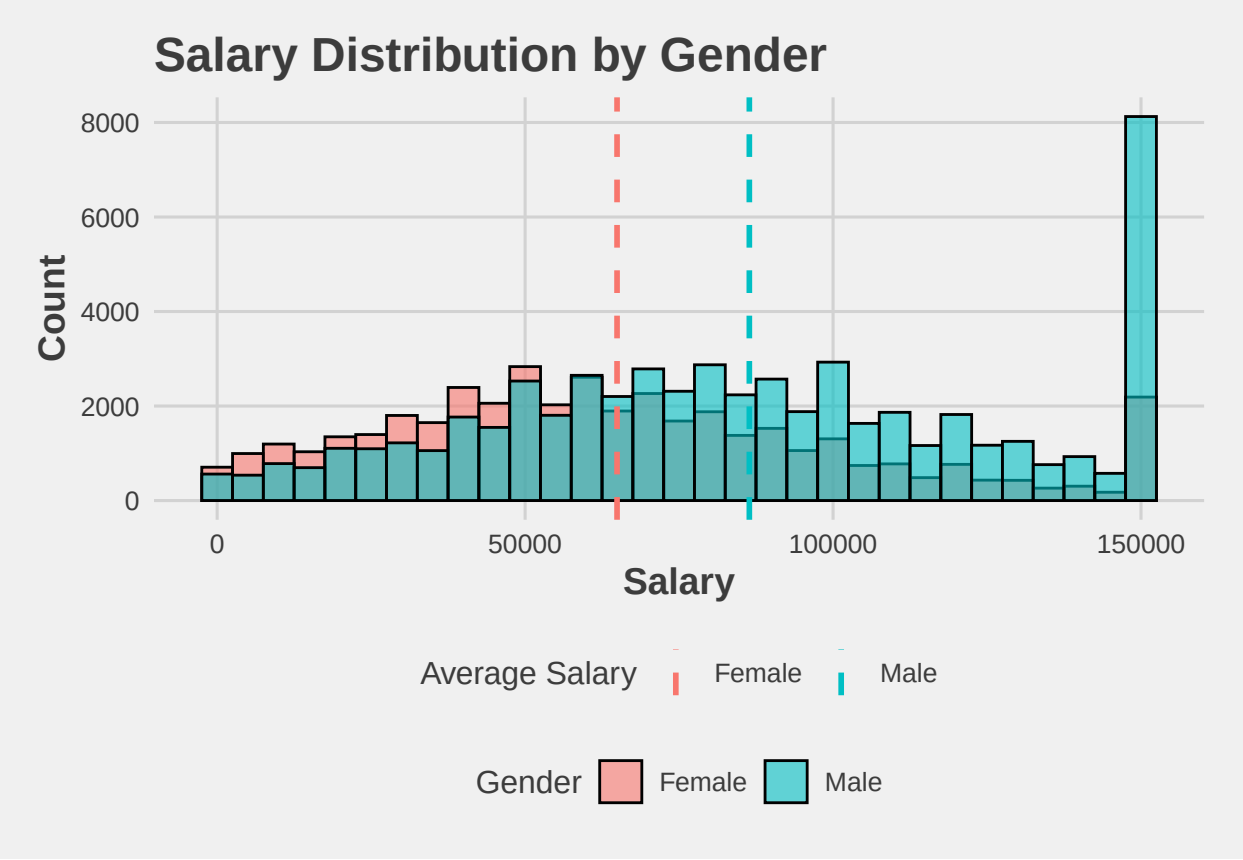
```
## Saving 6.5 x 4.5 in image
```

Salary by Gender The disaggregated data by gender underscores the gender pay gap between female and male respondents. In this case, male respondents on average almost make as much as the upper quartile of their female counterparts – around USD 90,000. Female respondents on average make a little more than the lower quartile of male respondents surveyed – less than USD 70,000.

```
avg_salary <- job_sat_cleaned %>%
  group_by(GENDER) %>%
  summarise(mean_salary = mean(SALARY, na.rm = TRUE))

avg_salary
```

```
## # A tibble: 2 x 2
##   GENDER mean_salary
##   <chr>        <dbl>
## 1 Female      64927.
## 2 Male        86409.
```

```
# Histogram by gender
salary_by_gender <- ggplot(job_sat_cleaned, aes(x = SALARY, fill = GENDER)) +
  geom_histogram(binwidth = 5000, position = "identity", alpha = 0.6, color = "black") +
  geom_vline(data = avg_salary, aes(xintercept = mean_salary, color = GENDER),
             linetype = "dashed", size = 1) +
  theme_classic() +
  labs(
    title = "Salary Distribution by Gender",
    x = "Salary",
    y = "Count",
    fill = "Gender",
    color = "Average Salary"
  ) + theme_fivethirtyeight() +
  theme(axis.title = element_text(size = 14, face = "bold"))

print(salary_by_gender)
```




```
ggsave("3. Outputs/Salary_graph_histogram.png", salary_by_gender)
```

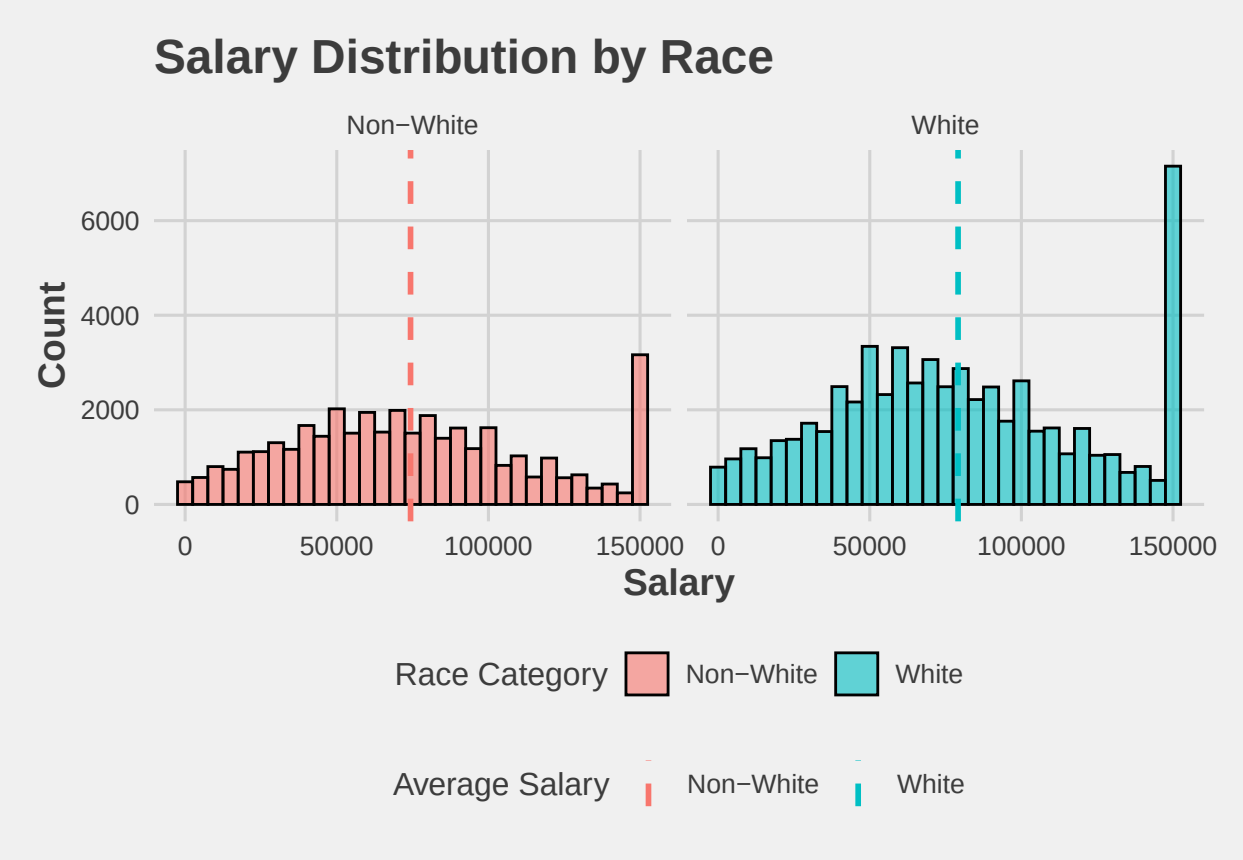
```
## Saving 6.5 x 4.5 in image
```

Salary by Race The disparity between white and non-white is smaller with a 4,796 difference between the two groups. It is important to note that the majority of the non-white group is made up of respondents who identify as Asian.

```
# Calculate average salary for each race
avg_salary_race <- job_sat_cleaned %>%
  group_by(RACETH) %>%
  summarise(mean_salary = mean(SALARY, na.rm = TRUE))

avg_salary_race
```

```
## # A tibble: 2 x 2
##   RACETH    mean_salary
##   <chr>           <dbl>
## 1 Non-White      74326.
## 2 White          79123.
```

```
# Create the salary distribution by race plot with facets and average lines
salary_race_graph <- ggplot(job_sat_cleaned, aes(x = SALARY, fill = RACETH)) +
  geom_histogram(binwidth = 5000, position = "identity", alpha = 0.6, color = "black") +
  geom_vline(data = avg_salary_race, aes(xintercept = mean_salary, color = RACETH),
             linetype = "dashed", size = 1) +
  facet_grid(cols = vars(RACETH)) +
  theme_classic() +
  labs(
    title = "Salary Distribution by Race",
    x = "Salary",
    y = "Count",
    fill = "Race Category",
    color = "Average Salary"
  ) + theme_fivethirtyeight() +
  theme(axis.title = element_text(size = 14, face = "bold"))

# Print and save the graph
print(salary_race_graph)
```




```
ggsave("3. Outputs/Salary_by_race.png", salary_race_graph)
```

```
## Saving 6.5 x 4.5 in image
```

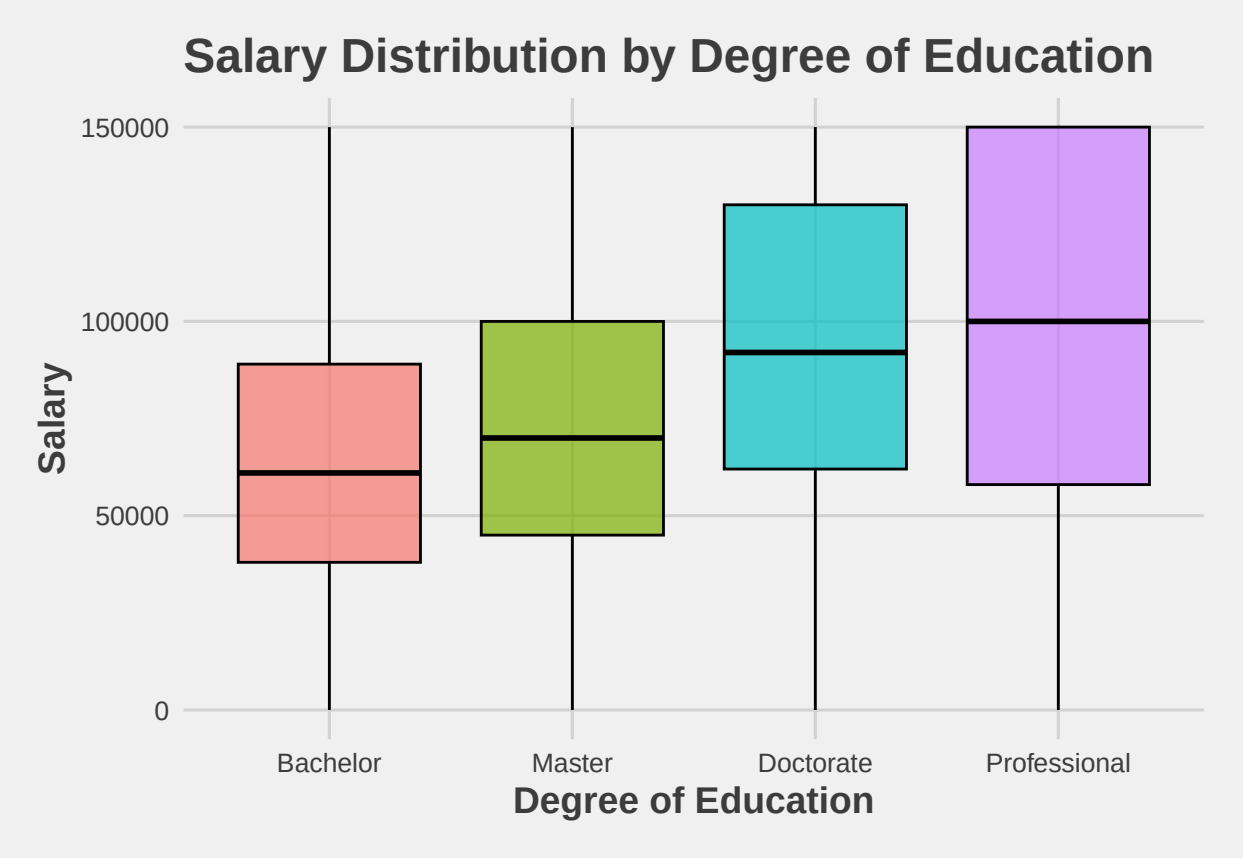
Salary by Degree The salary distribution by degree is as expected, it increases for each additional degree. Professional degree holders had the highest salary on average, around 97,000 while respondents with a Bachelors degree had the lowest salaries, around 65,000.

```
# Calculate average salary for each degree
avg_salary_degree <- job_sat_cleaned %>%
  group_by(DGRDG) %>%
  summarise(mean_salary = mean(SALARY, na.rm = TRUE))

avg_salary_degree
```

```
## # A tibble: 4 x 2
##   DGRDG        mean_salary
##   <fct>              <dbl>
## 1 Bachelor          65563.
## 2 Master            73235.
## 3 Doctorate         93453.
## 4 Professional      97996.
```

```
# Salary distribution by degree (with facets and average lines)
salary_degree_boxplot <- ggplot(job_sat_cleaned, aes(x = DGRDG, y = SALARY, fill = DGRDG)) +
  geom_boxplot(alpha = 0.7, color = "black") +
  labs(
    title = "Salary Distribution by Degree of Education",
```

```
    x = "Degree of Education",
    y = "Salary",
    fill = "Degree of Education"
  ) +
  theme_fivethirtyeight() +
  theme(axis.title = element_text(size = 14, face = "bold")) +
  theme(legend.position = "none")

print(salary_degree_boxplot)
```




```
ggsave("3. Outputs/Salary_by_degree.png", salary_degree_boxplot)
```

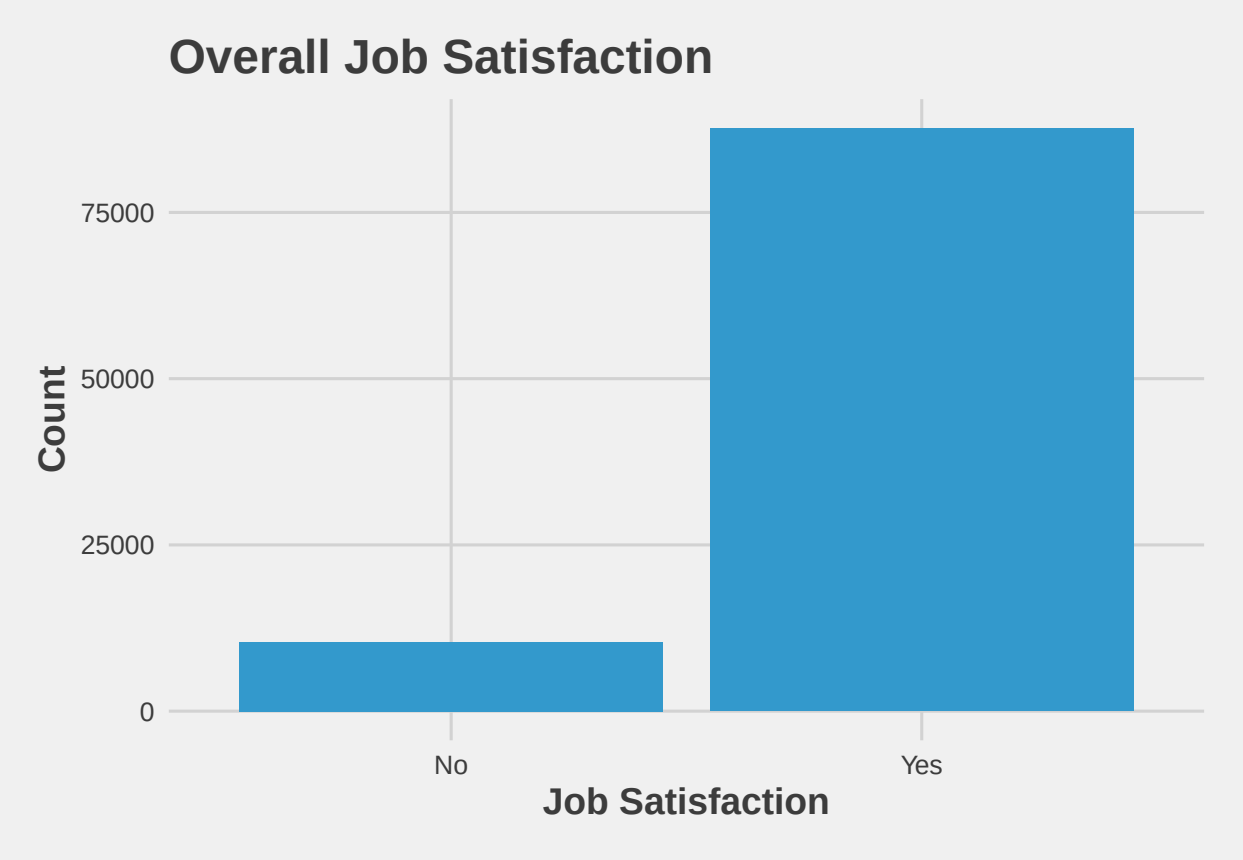
```
## Saving 6.5 x 4.5 in image
```

Results: Job Satisfaction

Overall Job Satisfaction This dataset explored job satisfaction in four ways (very satisfied (1), somewhat satisfied (2), somewhat dissatisfied (3), and very dissatisfied (4)), which was consolidated into a binary variable – collapsing 'very satisfied' and 'somewhat satisfied' into satisfied (1) and collapsing 'very dissatisfied' and 'somewhat dissatisfied' into dissatisfied (0) – for ease of retrieving outcomes based on probability. In the original dataset, most respondents were "somewhat satisfied" and the split between them being dissatisfied was fairly. Therefore, the analysis is slightly skewed due to the uneven distribution of responses 1/2 and 3/4 but still captures the overall sentiment.

Across all job categories, more than 82% of respondents were satisfied with their jobs – with the lowest being 'Other non-science and engineering' (83%). The job category of 'top and mid-level managers, executives and administrators' had the highest proportion of workers satisfied with their jobs (94%).

```
job_satisfaction <- job_sat_cleaned |>
  mutate(JOBSATIS = factor(JOBSATIS, labels = c("No", "Yes"))) |> # Relabel 0/1 to No/Yes
  ggplot(aes(x = JOBSATIS)) +
  geom_bar(fill = "#3399CC") +
  theme_classic() +
  labs(title = "Overall Job Satisfaction",
    x = "Job Satisfaction",
    y = "Count"
  ) + theme_fivethirtyeight() +
  theme(axis.title = element_text(size = 14, face = "bold"))

print(job_satisfaction)
```

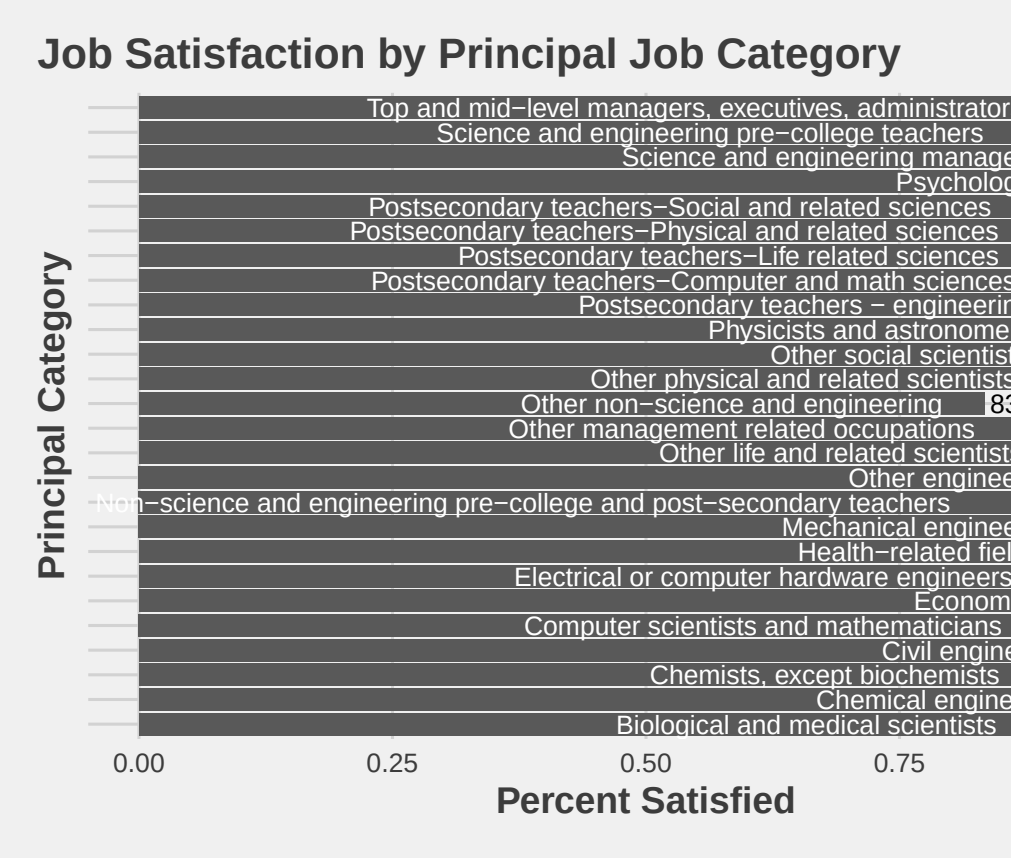



```
ggsave("3. Outputs/Job Satisfaction.png", job_satisfaction)
```

```
## Saving 6.5 x 4.5 in image
```

```
satisfaction_principal <- job_sat_cleaned |>
  group_by(NOCPR) |> summarize(percentage = mean(JOBSATIS)) |>
  ggplot(aes(x = NOCPR, y = percentage)) +
  geom_col() +
  ylim(0,1) + coord_flip() +
  geom_text(aes(label = NOCPR),
            hjust = 1.1,
            color = "white",
```

```
            size = 3.5) +
  geom_text(aes(label = scales::percent(percentage, accuracy = 1)),
            hjust = -0.1,
            color = "black",
            size = 3.5) +
  labs(x = "Principal Category", y = "Percent Satisfied",
       title = "Job Satisfaction by Principal Job Category") +
  theme_fivethirtyeight() +
  theme(axis.text.y = element_blank()) +
  theme(axis.title.x = element_text(size = 14, face = "bold")) +
  theme(axis.title.y = element_text(size = 14, face = "bold")) +
  theme(plot.title = element_text(hjust = 0, vjust = 1, size = 16, face = "bold"),
    plot.title.position = "plot")

print(satisfaction_principal)
```



Job Satisfaction by Occupation

```
ggsave("3. Outputs/Job Satisfaction by Principal Job.png", satisfaction_principal)
```

```
## Saving 6.5 x 4.5 in image
```

What characteristics drive job satisfaction in the STEM fields?

Logistic Regression: Determinants of Job Satisfaction The logistic regression analysis aims to identify the determinants of job satisfaction among the surveyed individuals. The dependent variable, job satisfaction, is binary, taking 1 if the individual is satisfied with their job and 0 otherwise. Several independent variables were included in the model to assess their impact on job satisfaction.

The result identifies several significant predictors of job satisfaction. Age, gender, race (weakly significant), number of children, highest degree obtained, hours worked per week, job benefits, and various occupational categories are shown as influential covariates. Older individuals are slightly less likely to report being satisfied with their jobs, which might be due to higher expectations or accumulated work-related stress over time. White individuals are slightly less satisfied with their jobs than other racial groups, which may reflect differences in job experiences or workplace environments.

Individuals with more children tend to report lower job satisfaction. Interestingly, higher educational attainment is associated with lower job satisfaction, suggesting that highly educated individuals have higher expectations of what they are going to get out of the job or face more demanding job roles. More hours worked per week significantly decreases job satisfaction, highlighting the importance of work-life balance.

Job benefits play a crucial role in increasing job satisfaction. The availability of a pension plan and access to a profit-sharing plan are strong positive predictors of job satisfaction, indicating that financial security and rewards are highly valued by employees. Occupational roles also significantly influence job satisfaction, with postsecondary teachers in computer and mathematical sciences, psychologists, and managers reporting higher satisfaction levels. These findings suggest that job characteristics, demographic factors, and occupational roles collectively impact job satisfaction.

```
#Dataframe for independent variables
independent_vars <- job_sat_cleaned %>% dplyr::select(-JOBSATIS)
x_cat <- sparse.model.matrix(~., data=independent_vars)[,-1]
Y <- as.factor(job_sat_cleaned$JOBSATIS)

formula <- as.formula(paste("JOBSATIS ~", paste(names(independent_vars), collapse = " + ")))
logit_model <- glm(formula, data = job_sat_cleaned, family = "binomial")
summary(logit_model)


tidy_results <- tidy(logit_model)
write.csv(tidy_results, "logit_model_results.csv", row.names = FALSE)
```

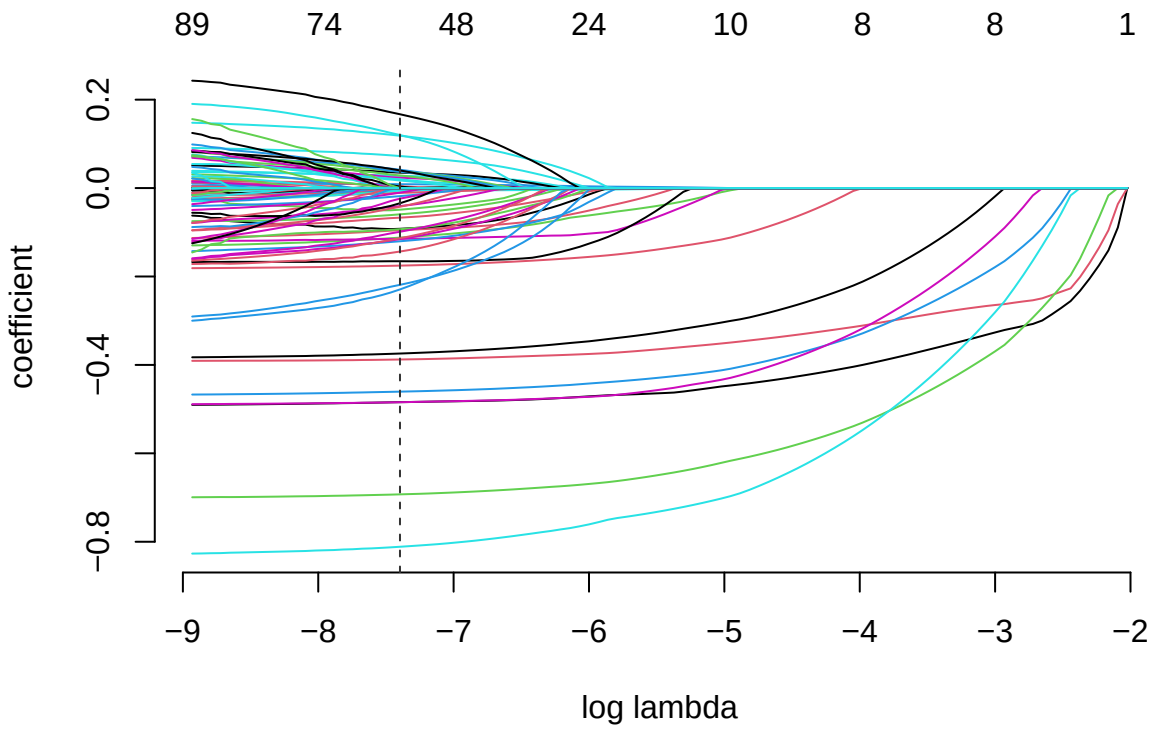
Lasso Regression: Top Predictors of Job Satisfaction (with Penalties) The LASSO regression analysis identified key predictors of job satisfaction by applying a penalty to less significant variables, enabling variable selection and regularization. Results indicate that 54 variables were retained, with a lambda value of 0.0028, accounting for 44% of the variance in job satisfaction.

Age and gender emerged as significant predictors, with older individuals reporting slightly lower satisfaction and males showing marginally reduced satisfaction compared to females. Educational attainment also plays a crucial role, with degrees in fields such as computer and mathematical sciences, psychology, and engineering positively influencing job satisfaction. For instance, post-secondary teachers in computer and mathematical sciences and psychologists report higher satisfaction, reflecting positive occupational experiences.

Job characteristics, including weekly working hours and access to benefits like pension plans and profit-sharing, significantly impact satisfaction. Additional factors such as work-related training and job security further contribute, underscoring the importance of professional development opportunities and perceived stability in enhancing satisfaction.

These findings emphasize actionable strategies to improve employee well-being. Organizations can enhance satisfaction by offering competitive benefits, ensuring manageable working hours, supporting professional growth, and addressing the unique needs of diverse demographic groups. Prioritizing these factors can help cultivate a more satisfied, engaged, and productive workforce.

```
lasso1 <- gamlr(x_cat, y=Y,family="binomial", lambda.min.ratio=1e-3)
plot(lasso1)
```




```
dev1 <- lasso1$deviance[which.min(AICc(lasso1))]
dev1_0 <- lasso1$deviance[1]
1-dev1/dev1_0
```

```
##     seg78
## 0.4458119
```

```
which.min(AICc(lasso1))
```

```
## seg78
##    78
```

```
summary(lasso1)[56,]
```

```
##
## binomial gamlr with 99 inputs and 100 segments.
```

```
##            lambda par df        r2     aicc
## seg56 0.002848023  19 19 0.4403152 37149.15
```

```
### Get csv table
best_model_index <- which.min(AICc(lasso1))
best_model_coefficients <- coef(lasso1, s = best_model_index)

coeff_df <- as.data.frame(as.matrix(best_model_coefficients))
coeff_df <- cbind(Variable = rownames(coeff_df), Coefficient = coeff_df)
colnames(coeff_df) <- c("Variable", "Coefficient")

coeff_df <- coeff_df[coeff_df$Coefficient != 0,]

lasso_results <- write.csv(coeff_df, "lasso_model_coefficients.csv", row.names = FALSE)
```

Decision Trees: Predicting Job Satisfaction in a Non-Linear Fashion The decision tree model is a machine learning technique used to predict a principal's overall job satisfaction by analyzing several factors that may influence their perception of their job. These factors include their satisfaction with responsibilities, opportunities for advancement, salary, and sense of contribution to society.

A decision tree works by recursively splitting the data into smaller, more specific groups based on the values of various predictor variables. For instance, it might ask questions like, "Is the principal satisfied with their level of responsibility?" or "Is the principal satisfied with their opportunity for advancement?" Based on the answers, the tree places principals into different "branches" that represent subgroups with similar characteristics. The model then predicts the overall job satisfaction for each group based on the average job satisfaction of the principals within that group.

Decision Tree Methodology:

- If a principal is dissatisfied with both their responsibility (SATRESP < 2.5) and their opportunity for advancement (SATADV < 3.5), the model predicts their overall job satisfaction to be 29% on average. This suggests that principals in this group are likely to report much lower levels of satisfaction with their jobs.
- Conversely, if a principal is satisfied with their responsibility (SATRESP >= 2.5), their opportunity for advancement (SATADV >= 3.5), salary (SATSAL >= 2.5), and sense of societal contribution (SATSOC >= 2.5), the model predicts their job satisfaction to be 97% on average. This indicates that principals in this group are much more likely to feel positive and fulfilled in their roles.

```
job_tree <- tree(JOBSATIS ~ ., data = job_sat_cleaned,
                 control = tree.control(nobs = nrow(job_sat_cleaned),
                                        mincut = 5, minsize = 10, mindev = 0.01))
```

```
## Warning in tree(JOBSATIS ~ ., data = job_sat_cleaned, control =
## tree.control(nobs = nrow(job_sat_cleaned), : NAs introduced by coercion
```

```
cv_tree <- cv.tree(job_tree, K=90)
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion

## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion

## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

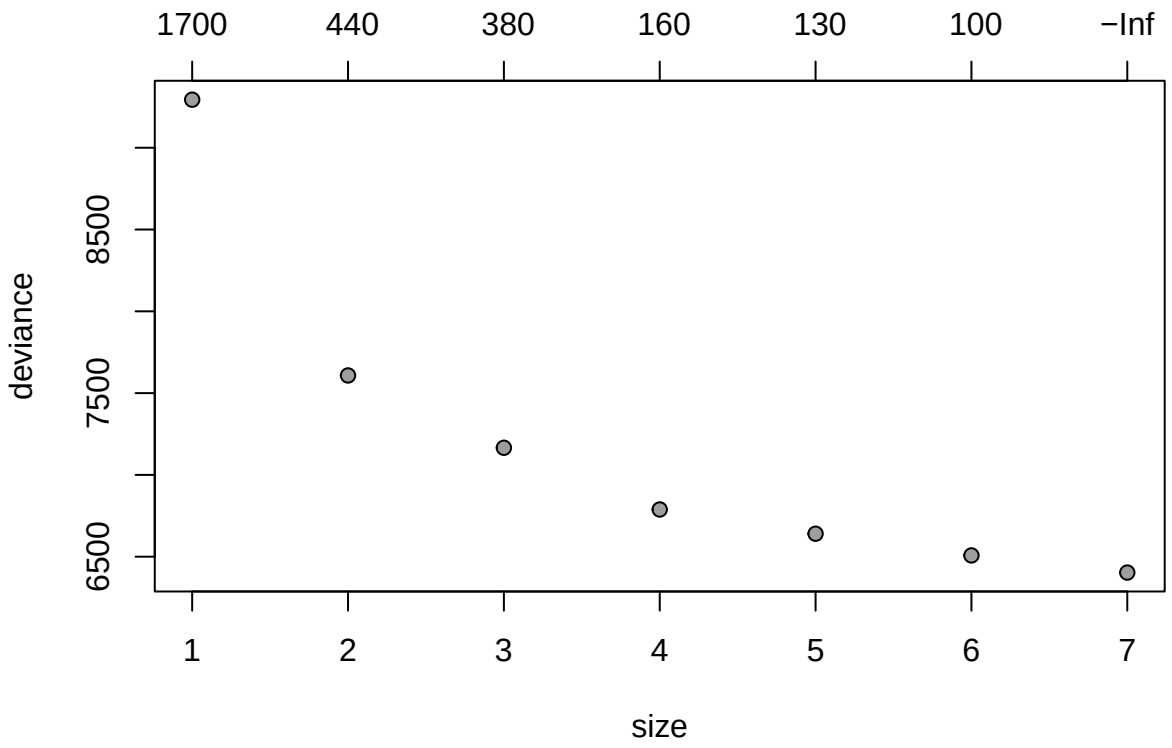
```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

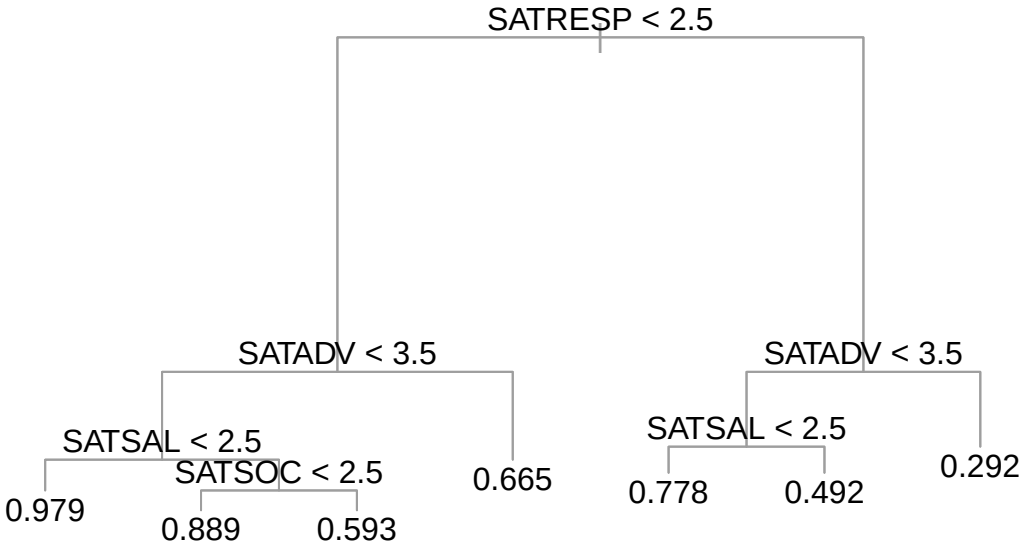
```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
cv_tree$size
cv_tree$dev
best_nodes <- cv_tree$size[which.min(cv_tree$dev)]

plot(cv_tree, pch=21, bg=8, type="p")
```




```
job_pruned <- prune.tree(job_tree, best=best_nodes)
plot(job_pruned, col=8)
text(job_pruned, digits=3, cex=1)
```



Conclusion

In this project, a comprehensive dataset from IPUMS Higher Ed was utilized to analyze job satisfaction within the U.S. workforce, with a particular focus on STEM fields. The analysis considered a range of demographic, lifestyle, and job-related factors to identify the key drivers of job satisfaction.

The logistic regression and LASSO reveals that job satisfaction is significantly influenced by age, gender, race, educational attainment, hours worked, and job benefits. Additionally, decision tree regression revealed that satisfaction with job responsibilities and opportunities for advancement are critical determinants of overall job satisfaction.

The findings emphasize the importance of addressing these factors to enhance employee satisfaction and well-being. Organizations can use these insights to develop strategies that promote a more engaged and productive workforce by offering competitive job benefits, maintaining a healthy work-life balance, and providing opportunities for professional growth.