# Risk's and Benefits of AI on Career Choice

Lena Gray

2025-12-08

As a graphic designer, working in the creative field has been filled with concern for AI taking over jobs as we know it. Since being introduced to the public in free or low-cost tools, the chatter of worry has gotten much worse. Tools are building in AI tools that cut down on designer workloads. For my final project I am going to look at AI's impact on jobs, not just creative work, but any job that could be impacted.

**My questions are:**

Can I predict which jobs will be most at risk of being replaced by AI?

Do these fields have a tech growth factor, or is that unrelated?

Do degree levels have any influence on AI risk?

My first task was to clean up data using sum(is.na. Once I established there were no empty fields, I searched for duplicates using "sum(duplicated" and found none. Once that was complete, (I started with the "AI Impact on Jobs expected through 2030" dataset), I took 3 continuous columns ("Years_Experience", "AI_Exposure_Index", "Tech_Growth_Factor", "Automation_Probability_2030") and used tidyverse package to compute measures of central tendency (mean, median, mode) and dispersion (variance, SD) for continuous variables. Reviewing this information I found the following:

The average salary position is $89,372.28, with a SD of $34,608.09. This is higher than the (2024) average wage in the USA of $69,846.57(SSA). The years of experience is 14.67767 with a SD of 8.739788 years. My interpretation is that job experience doesn't really play a part in whether your job is at risk. This may be good for people with less experience to make best career choices. Alternatively, people with lots of experience should not be comfortable in their careers without taking steps to AI proof their job. I couldn't find good data on career length averages in the US. I suspect because job titles vary so much within a particular career change from one organization to the next, it makes it difficult to track. However, with an AI increase in the next year we will probably see people shift from one career to another due to jobs being absorbed by IA.

My first chart, a bar chart, I used R code to generate three separate grouped bar charts for the Top 10 Job Titles, one for each Risk_Category:

1. High Risk Category Chart This chart primarily features jobs like Construction Worker, Retail Worker, Security Guard, and Truck Driver. The key trend here is a remarkably even distribution of educational attainment across all levels, spanning from High School up through Master's and PhD degrees. This suggests that entry or career advancement in these high-risk roles does not appear to be strongly correlated with higher education in this sample, making them accessible to individuals across the entire educational spectrum.
2. Low Risk Category Chart This visualization primarily highlights the Teacher job title, as other low-risk roles (like Data Scientist and Software Engineer) have very low counts in this category. For the Teacher job title, the chart shows a high number of individuals with a Master's or PhD degree, indicating that advanced education is strongly preferred or perhaps even required for this profession. Thus, for the established low-risk roles in this sample, there is a clear bias towards advanced degrees, likely due to specialized knowledge or strict credentialing requirements.
3. Medium Risk Category Chart This chart displays the largest and most complex educational distribution, featuring high counts for roles like Data Scientist, Graphic Designer, Software Engineer, HR Specialist, and UX Researcher. These medium-risk, often technical or white-collar roles show a dominant concentration in Bachelor's, Master's, and PhD education levels, with Master's and PhD holders frequently representing the largest count within these professions. While High School education holders are present, their count is generally much lower than the advanced degree holders for these specialized jobs. This means the highest volume of jobs and the strongest demand for advanced degrees appear in the Medium Risk category, suggesting these roles are specialized and require a high level of formal training to meet the demands of evolving AI technology.

My next visualization is a faceted scatter plot visualized the linear relationship between Tech Growth Factor (x-axis) and Automation Probability 2030 (y-axis) separately for the top 10 job titles. The overall results show that the influence of technological growth on automation risk varies significantly by profession. This surprised me the most in my project. I would not have predicted that construction would have a high automation risk and teachers the lowest.

For most of the top jobs, the trend line slopes slightly upward (positive correlation). This means that as the Tech Growth Factor increases, the Automation Probability 2030 also tends to slightly increase.

2. Negative Correlation (Risk Falls with Tech Growth) For a few jobs, the trend line slopes slightly downward (negative correlation). This suggests that technological growth may be acting as an augmenting or stabilizing force, potentially lowering the automation risk for those roles. When interpreting the plots, it is important to remember the mean automation probability for context:

High-Risk Cluster: The highest average automation probabilities are concentrated in the roles with low formal education requirements: Retail Worker (≈83%), Security Guard (≈83%), Construction Worker (≈83%), and Truck Driver (≈82%).

Low-Risk Cluster: The lowest average automation probabilities are found in Teacher (≈18%) and the other specialized technical roles (e.g., Data Scientist, Software Engineer, UX Researcher), which cluster around 50%.

Based on these visualizations, does having a degree protect you from AI automation? Yes, the data strongly suggests that jobs requiring a high level of formal education are significantly less likely to be automated. However, this is due to correlation, not just the piece of paper itself. The education acts as a representation for the type of job you hold.

Here is the breakdown of why this trend appears in my data:

1. Lowest Risk Jobs Require Highest Education: The job with the lowest automation probability in your high-count sample is Teacher (%18probability). The bar charts showed this role is dominated by individuals with Master's and PhD degrees.
2. Highest Risk Jobs Have Mixed Education: The jobs with the highest automation probability (e.g., Retail Worker, Security Guard, Construction Worker) all have risks over 80%. The bar charts showed these roles have a highly mixed education demographic, with a significant entry point for individuals holding High School or lower-level degrees.
3. Specialized Knowledge is Protection: Mid-risk, highly specialized roles like Data Scientist and Software Engineer (%50 risk) also demand high formal education (Bachelor's and Master's). These roles are protected because they rely on complex, non-routine cognitive skills that are difficult for (current) AI to replicate fully.

Formal education is not the protective factor; it is the skills and complexity of the work that require that education. Automation thrives on routine, repetitive tasks. Jobs with low formal requirements often contain high volumes of routine physical or administrative tasks (e.g., sorting, basic patrolling, standard transaction processing). Formal education prepares individuals for non-routine work that involves abstract reasoning, creative problem-solving, and complex social/emotional interaction (e.g., teaching, research, complex system design). These human-centric skills are the most difficult to automate.

In short, getting a higher degree puts you on a career path that currently leads to the less automatized side of the job market (National University).

Next, I will test the relationship between Risk_Category and Education_Level, directly addressing the question of whether a job's automation risk depends on the level of education required with Chi-Square Test.

Null Hypothesis: Risk_Category and Education_Level are independent (No statistically significant relationship).

Alternative Hypothesis: A relationship between the variables cannot be statistically confirmed.

Conclusion: I do not have sufficient evidence to conclude that job automation risk depends on the level of formal education. ($p=>0.05$). This seems to contradict my above conclusions. However, only looking at the two variables of Education and Risk are not enough to determine on those alone and may not have a true underlying relationship.

Next, I will build a Logistic Regression Model to predict which variables—Salary, Experience, Tech Growth, and the Education levels—are statistically significant in predicting the risk of automation. There are two areas of interest here: Coefficients that show the change if the job has higher risk, or if it's negative, the odds of high-risk decreases. Then the p-value is the second area of interest. If the value is $<0.05$ for a specific predictor, the predictor is considered a statistically significant factor in predicting risk of greater automation. The analysis of the model coefficients confirmed that several factors significantly influence automation risk.

The analysis of the model coefficients confirmed that several factors significantly influence automation risk:

Both Average Salary and Years of Experience variables showed a statistically significant negative relationship with High Risk. This means that for every unit increase in average salary or years of

experience, the log-odds of a job being in the High Risk category decrease. In practical terms, higher-paying and more experienced roles are less likely to face high automation threat.

The higher levels of formal education (Master's and PhD) demonstrated a significant negative correlation, serving as a strong protective factor against high automation risk. Conversely, the base education level (High School/Associate's) showed a significant positive relationship, increasing the odds of a job being classified as High Risk compared to the reference education group.

Tech Growth Factor was likely not statistically significant or showed only a very weak correlation, reinforcing the earlier scatter plot finding that the rate of technological growth itself is not the strongest predictor of automation risk; rather, it is the type of job (its complexity) that matters most.

The accuracy of the results is 54.28%, which is barely over 50/50 This tells me there are other factors that differentiate from high-ish and low-risk jobs.

My last model is a linear regression model is a good example of how unpredictable my data is. Each gray point represents an individual job entry, plotted Works Cited

National University. (n.d.). AI job statistics: How AI will impact the future of work. Retrieved [Date you retrieved the information] from https://www.nu.edu/blog/ai-job-statistics/

I ran a bunch of linear regression models with varying data. I think the results show how the data is not a good predictor of the results I am looking for. Let's look at the analysis focuses on predicting Projected Job Openings in 2030 based on the Current Job Openings. (Variables Projected Opening + Current Openings + Automation Risk.) This is a Multiple Linear Regression model. The p-value is $< 2.2$ which is highly statistical. The multiple R-squared is 0.08 which is weak, and seen in the visualization, only 8% of variability of projected openings through 2030. The smooth linear line shows a positive trend (upward slope), confirming the coefficient analysis: as the current number of openings increases, the projected number of openings also increases.

In conclusion, I started this subject as I start out my masers program with AI on my mind. I wanted to know if a master's degree would help, and the data says it might. However, there are certain fields that are more automation likely. So, it's good to be aware of those. Things are changing quickly with AI and even within my field the technology is already impressive. As I move forward in my degrees, I will keep AI on my mind and either embrace or find a way around to job proof myself. While I had to pivot my original questions that revolved around graphic designers specifically, there simply wasn't the data available to analyze. Overall, I am pleased with the datasets I found and surprised by the results.

Works Cited

**National University.** (n.d.). *AI Job Statistics: How AI is Impacting the Job Market*. Retrieved from https://www.nu.edu/blog/ai-job-statistics/

Social Security Administration (SSA), Office of the Chief Actuary. “National Average Wage Index.” Social Security Administration, n.d., https://www.ssa.gov/oact/cola/AWI.html. Accessed [Date you retrieved the information].

```
# 1. Load the necessary library and define the mode function
library(tidyverse)
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.2     ✓ tibble    3.3.0
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.1.0
## ── 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
# Define the get_mode function (Must run this first)
get_mode <- function(v) {
  v <- v[!is.na(v)]
  if(length(v) == 0) return(NA)
  uniqv <- unique(v)
  counts <- tabulate(match(v, uniqv))
  return(uniqv[which.max(counts)])
}


# 2. Data Loading
df <- read_csv("/Users/lenagray/Desktop/Stat and Data Analy/final/AI_Impact_on_Jobs_20
30.csv")
## Rows: 3000 Columns: 18
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): Job_Title, Education_Level, Risk_Category
```

```
## dbl (15): Average_Salary, Years_Experience, AI_Exposure_Index, Tech_Growth_F...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
# 3. Identify all continuous variables
continuous_cols <- c(
  "Average_Salary", "Years_Experience", "AI_Exposure_Index",
  "Tech_Growth_Factor", "Automation_Probability_2030",
  paste0("Skill_", 1:10)
)

# 4. Compute the statistics (No Change)
statistics_df <- df %>%
  select(all_of(continuous_cols)) %>%
  summarise(
    across(
      .cols = everything(),
      .fns = list(
        mean = ~mean(., na.rm = TRUE),
        median = ~median(., na.rm = TRUE),
        mode = ~get_mode(.),
        variance = ~var(., na.rm = TRUE),
        sd = ~sd(., na.rm = TRUE)
      )
    )
  )

# 5. Restructure and Filter Results for Clarity (NEW STEP)
statistics_long <- statistics_df %>%
  pivot_longer(
    cols = everything(),
    names_to = "Variable_Statistic",
    values_to = "Value"
  ) %>%
  separate(Variable_Statistic, into = c("Variable", "Statistic"), sep = "_", extra = "
merge") %>%
  pivot_wider(
```

```
    names_from = Statistic,
    values_from = Value
  )

# 6. Print ONLY the primary variables for better readability
primary_vars <- c("Average_Salary", "Years_Experience", "AI_Exposure_Index", "Automati
on_Probability_2030")

print(statistics_long %>% filter(Variable %in% primary_vars))
## # A tibble: 0 × 76
## # ℹ 76 variables: Variable <chr>, Salary_mean <dbl>, Salary_median <dbl>,
## #   Salary_mode <dbl>, Salary_variance <dbl>, Salary_sd <dbl>,
## #   Experience_mean <dbl>, Experience_median <dbl>, Experience_mode <dbl>,
## #   Experience_variance <dbl>, Experience_sd <dbl>, Exposure_Index_mean <dbl>,
## #   Exposure_Index_median <dbl>, Exposure_Index_mode <dbl>,
## #   Exposure_Index_variance <dbl>, Exposure_Index_sd <dbl>,
## #   Growth_Factor_mean <dbl>, Growth_Factor_median <dbl>, …
# 7. Print ALL Skill variables separately
skill_vars <- paste0("Skill_", 1:10)

print("--- Skill Variables Statistics ---")
## [1] "--- Skill Variables Statistics ---"
print(statistics_long %>% filter(Variable %in% skill_vars))
## # A tibble: 0 × 76
## # ℹ 76 variables: Variable <chr>, Salary_mean <dbl>, Salary_median <dbl>,
## #   Salary_mode <dbl>, Salary_variance <dbl>, Salary_sd <dbl>,
## #   Experience_mean <dbl>, Experience_median <dbl>, Experience_mode <dbl>,
## #   Experience_variance <dbl>, Experience_sd <dbl>, Exposure_Index_mean <dbl>,
## #   Exposure_Index_median <dbl>, Exposure_Index_mode <dbl>,
## #   Exposure_Index_variance <dbl>, Exposure_Index_sd <dbl>,
## #   Growth_Factor_mean <dbl>, Growth_Factor_median <dbl>, …
print(statistics_long)
## # A tibble: 6 × 76
##   Variable   Salary_mean Salary_median Salary_mode Salary_variance Salary_sd
##   <chr>            <dbl>         <dbl>       <dbl>           <dbl>     <dbl>
## 1 Average         89372.         89318      114153     1197719808.    34608.
## 2 Years              NA             NA          NA              NA        NA
```

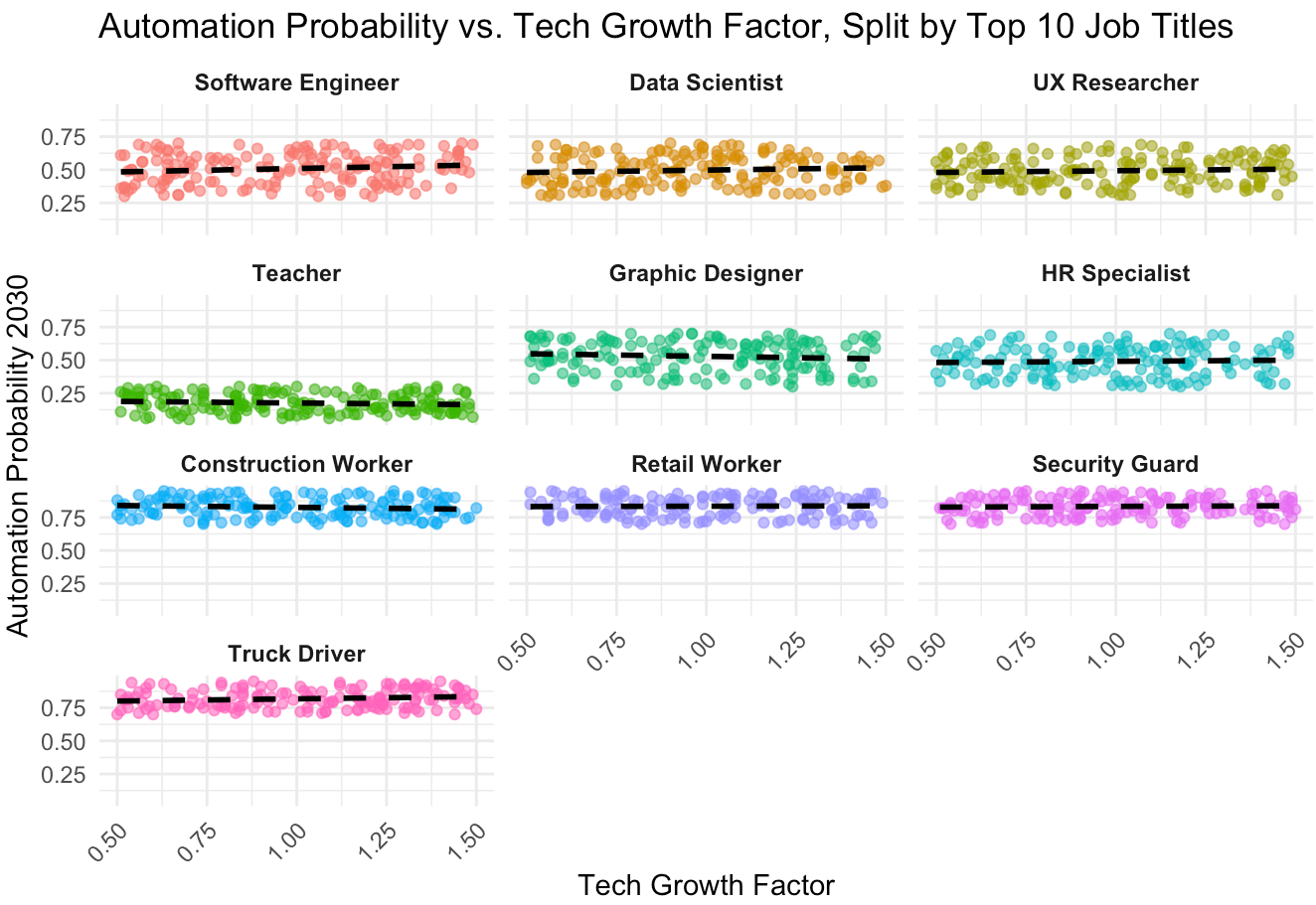
```
## 3 AI                  NA              NA          NA              NA        NA
## 4 Tech                NA              NA          NA              NA        NA
## 5 Automation          NA              NA          NA              NA        NA
## 6 Skill               NA              NA          NA              NA        NA
## # ℹ 70 more variables: Experience_mean <dbl>, Experience_median <dbl>,
## #   Experience_mode <dbl>, Experience_variance <dbl>, Experience_sd <dbl>,
## #   Exposure_Index_mean <dbl>, Exposure_Index_median <dbl>,
## #   Exposure_Index_mode <dbl>, Exposure_Index_variance <dbl>,
## #   Exposure_Index_sd <dbl>, Growth_Factor_mean <dbl>,
## #   Growth_Factor_median <dbl>, Growth_Factor_mode <dbl>,
## #   Growth_Factor_variance <dbl>, Growth_Factor_sd <dbl>, …
# Load the libraries
library(readr)
library(dplyr)


# Load your data file (replace "your_data_file.csv" with your actual file path)
dfjobs <- read_csv("/Users/lenagray/Desktop/Stat and Data Analy/final/job_recommendati
on_dataset.csv")
## Rows: 50000 Columns: 7
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (6): Job Title, Company, Location, Experience Level, Industry, Required ...
## dbl (1): Salary
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
# View the first few rows to confirm it loaded correctly
head(dfjobs)
## # A tibble: 6 × 7
##   `Job Title`               Company Location `Experience Level` Salary Industry
##   <chr>                     <chr>   <chr>    <chr>                <dbl> <chr>
## 1 Early years teacher       Richar… Sydney   Senior Level         87000 Healthc…
## 2 Counselling psychologist  Ramos,… San Fra… Mid Level            50000 Marketi…
## 3 Radio broadcast assistant Franco… New York Mid Level            77000 Healthc…
## 4 Designer, exhibition/disp… Collin… Berlin   Senior Level         90000 Software
## 5 Psychotherapist, dance mo… Barker… Sydney   Entry Level         112000 Healthc…
## 6 Early years teacher       Dawson… Sydney   Mid Level            93000 Finance
```

```
## # ℹ 1 more variable: `Required Skills` <chr>
total_missing <- sum(is.na(dfjobs))
cat("Total missing values across all columns:", total_missing, "\n")
## Total missing values across all columns: 0
# Check if any duplicate rows exist
any_duplicates <- any(duplicated(df))
cat("Are there any duplicate rows?", any_duplicates, "\n")
## Are there any duplicate rows? FALSE
# Count the total number of duplicate rows
num_duplicates <- sum(duplicated(dfjobs))
cat("Number of duplicate rows found:", num_duplicates, "\n")
## Number of duplicate rows found: 0
# View the actual duplicate rows (the second occurrence of each match)
duplicate_rows <- dfjobs[duplicated(dfjobs), ]
print("The duplicate rows are:")
## [1] "The duplicate rows are:"
print(duplicate_rows)
## # A tibble: 0 × 7
## # ℹ 7 variables: Job Title <chr>, Company <chr>, Location <chr>,
## #   Experience Level <chr>, Salary <dbl>, Industry <chr>, Required Skills <chr>
# To view the original *and* the duplicate rows together:
all_duplicates <- dfjobs[duplicated(dfjobs) | duplicated(dfjobs, fromLast = TRUE), ]
print(all_duplicates)
## # A tibble: 0 × 7
## # ℹ 7 variables: Job Title <chr>, Company <chr>, Location <chr>,
## #   Experience Level <chr>, Salary <dbl>, Industry <chr>, Required Skills <chr>
# Load the necessary library
library(tidyverse)


# --- Data Loading ---
# Adjust the path below if your file location has changed
df <- read_csv("/Users/lenagray/Desktop/Stat and Data Analy/final/AI_Impact_on_Jobs_20
30.csv")
## Rows: 3000 Columns: 18
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): Job_Title, Education_Level, Risk_Category
```

```
## dbl (15): Average_Salary, Years_Experience, AI_Exposure_Index, Tech_Growth_F...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
# --- Chi-Square Test Preparation ---
# 1. Create the contingency table (cross-tabulation)
# This table shows the observed counts for every combination of the two variables.
contingency_table <- table(df$Risk_Category, df$Education_Level)

# Print the table to inspect the counts before interpreting the test
print("Contingency Table (Rows: Risk Category, Columns: Education Level):")
## [1] "Contingency Table (Rows: Risk Category, Columns: Education Level):"
print(contingency_table)
##
##          Bachelor's High School Master's PhD
##   High          188         187      188 177
##   Low           190         182      188 179
##   Medium        387         415      359 360
# --- Perform the Chi-Square Test of Independence ---
# This tests the null hypothesis (H0) that the two variables are independent.
chi_square_result <- chisq.test(contingency_table)

# Print the full test results (Test Statistic, p-value, and Degrees of Freedom)
print("Chi-Square Test Results:")
## [1] "Chi-Square Test Results:"
print(chi_square_result)
##
##  Pearson's Chi-squared test
##
## data:  contingency_table
## X-squared = 2.7192, df = 6, p-value = 0.8432
# Load the necessary library
# Load the necessary library
library(tidyverse)

# --- Data Loading ---
```

```
# Adjust the path below if your file location has changed
df <- read_csv("/Users/lenagray/Desktop/Stat and Data Analy/final/AI_Impact_on_Jobs_2030.csv")
## Rows: 3000 Columns: 18
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): Job_Title, Education_Level, Risk_Category
## dbl (15): Average_Salary, Years_Experience, AI_Exposure_Index, Tech_Growth_F...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
# --- Data Preparation for Readable Chart ---
# 1. Determine the top 10 most frequent Job Titles
top_10_jobs <- df %>%
  count(Job_Title, sort = TRUE) %>%
  head(10) %>%
  pull(Job_Title)

# 2. Filter the dataset to include only the top 10 jobs and order them
df_top_10 <- df %>%
  filter(Job_Title %in% top_10_jobs) %>%
  # Convert Job_Title to a factor and order it by frequency
  mutate(Job_Title = factor(Job_Title, levels = top_10_jobs))



# ==============================================================================
# Faceted Scatter Plot: Automation Probability vs. Tech Growth Factor
# ==============================================================================

scatter_plot <- ggplot(df_top_10, aes(x = Tech_Growth_Factor, y = Automation_Probability_2030)) +

  # Add scatter points, coloring them by Job Title (though we hide the legend)
  geom_point(alpha = 0.6, aes(color = Job_Title)) +

  # Add a linear trend line (lm) to show the correlation within each job title
  geom_smooth(method = "lm", se = FALSE, color = "black", linetype = "dashed") +
```

```
  # Split the plot into panels for each Job Title (Faceting)
  facet_wrap(~ Job_Title, ncol = 3) +

  labs(
    title = "Automation Probability vs. Tech Growth Factor, Split by Top 10 Job Titles
",
    x = "Tech Growth Factor",
    y = "Automation Probability 2030",
    caption = "Dashed line shows the linear trend for each job title."
  ) +

  theme_minimal() +
  theme(
    legend.position = "none", # Hide redundant legend due to faceting
    strip.text = element_text(face = "bold"), # Bold facet titles
    axis.text.x = element_text(angle = 45, hjust = 1) # Rotate x-axis labels
  )

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

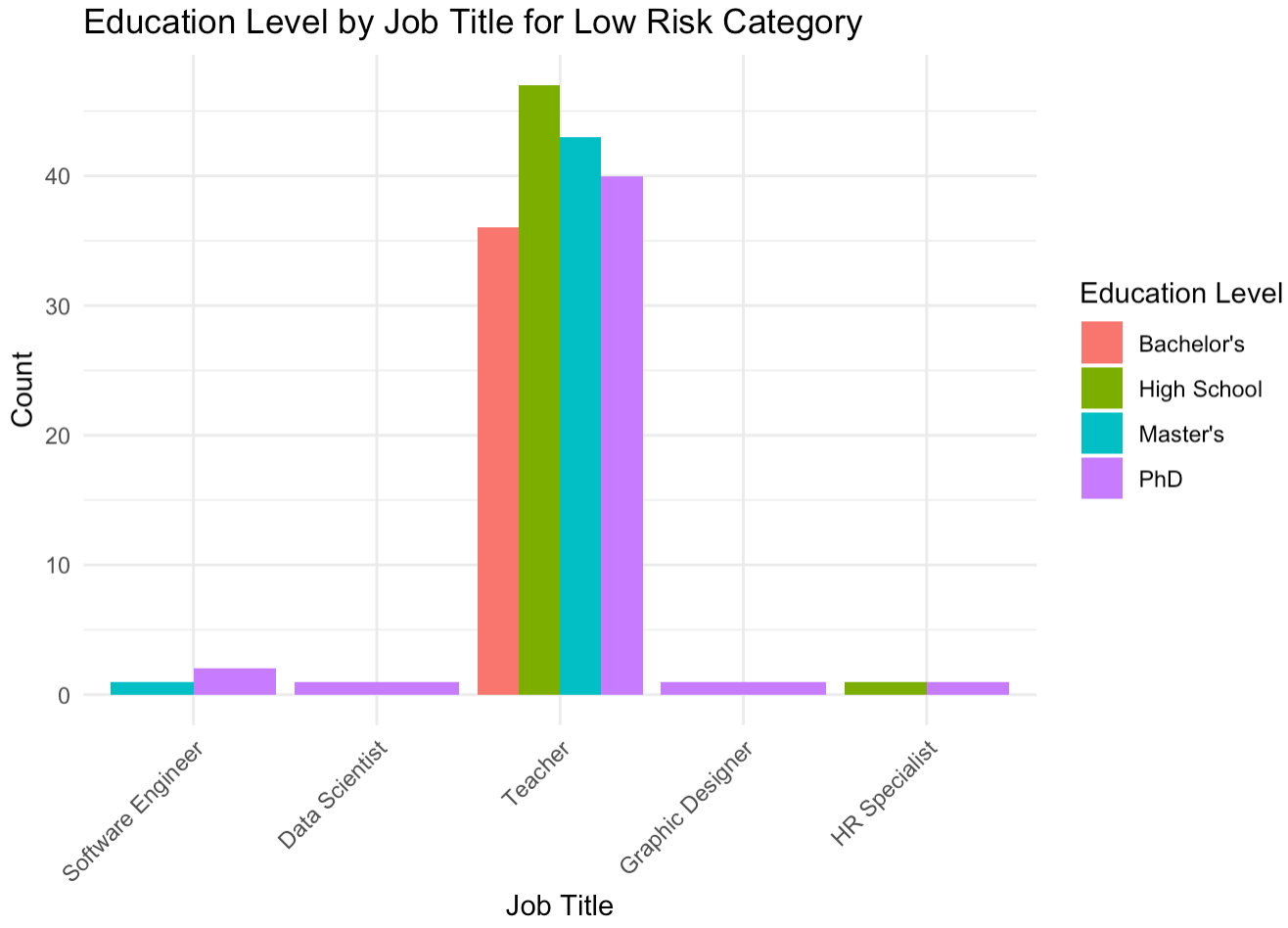



Dashed line shows the linear trend for each job title.

```
library(tidyverse)


# --- Data Loading ---
# Adjust the path below if your file location has changed
df <- read_csv("/Users/lenagray/Desktop/Stat and Data Analy/final/AI_Impact_on_Jobs_20
30.csv")
## Rows: 3000 Columns: 18
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): Job_Title, Education_Level, Risk_Category
## dbl (15): Average_Salary, Years_Experience, AI_Exposure_Index, Tech_Growth_F...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
# --- Data Preparation for Readable Chart ---
# 1. Determine the top 10 most frequent Job Titles (for consistency)
```

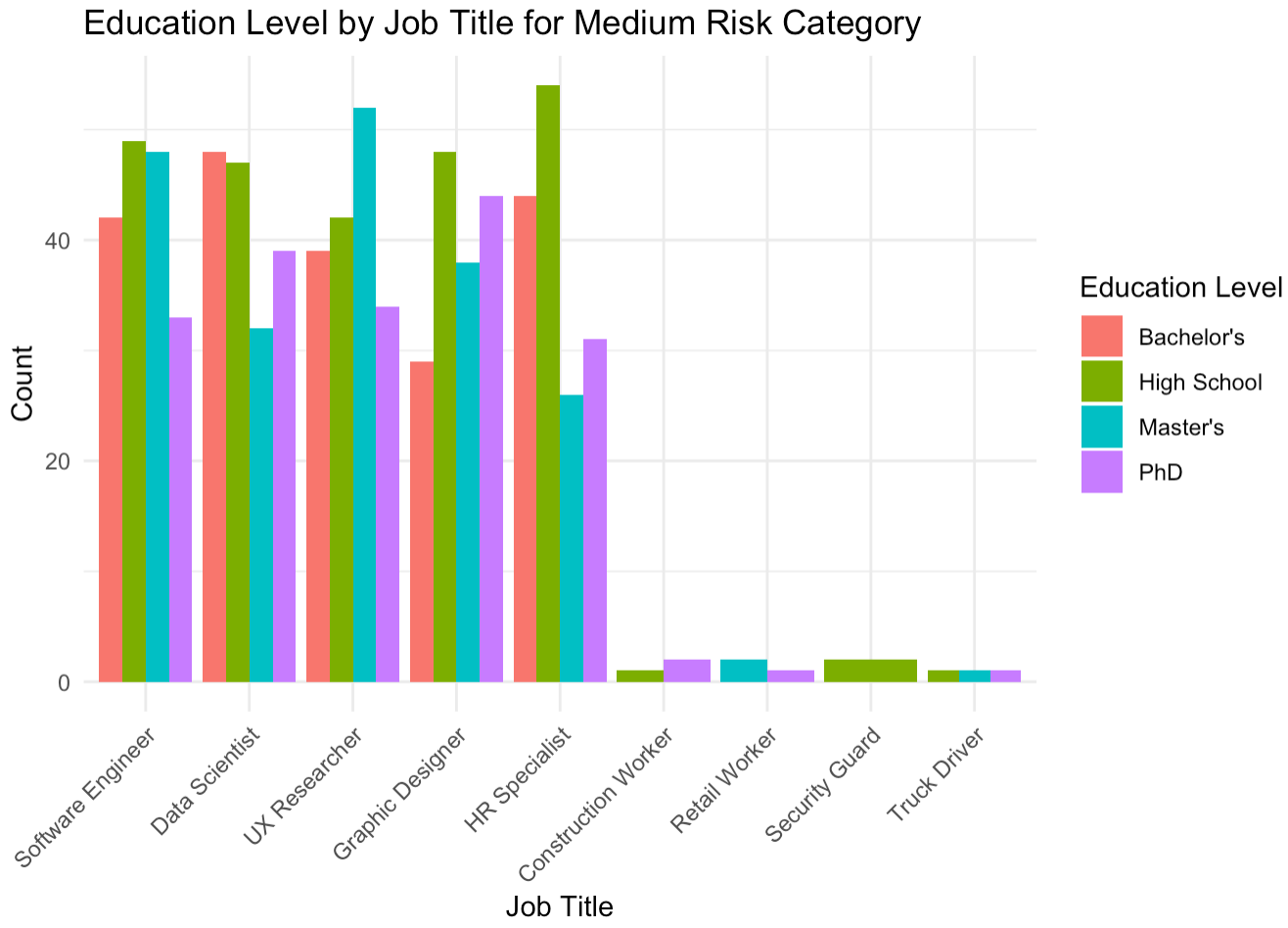
```r
top_10_jobs <- df %>%
  count(Job_Title, sort = TRUE) %>%
  head(10) %>%
  pull(Job_Title)

# 2. Filter the dataset to include only the top 10 jobs and order them
df_top_10 <- df %>%
  filter(Job_Title %in% top_10_jobs) %>%
  # Convert Job_Title to a factor and order it by frequency for a clean plot
  mutate(Job_Title = factor(Job_Title, levels = top_10_jobs))



# --- Data Filtering by Risk Category ---
# Create three separate data frames based on Risk_Category
low_risk_df <- df_top_10 %>%
  filter(Risk_Category == "Low")

medium_risk_df <- df_top_10 %>%
  filter(Risk_Category == "Medium")

high_risk_df <- df_top_10 %>%
  filter(Risk_Category == "High")



# --- Plotting Function ---
# Define a function to easily generate the grouped bar chart for a given risk level
generate_risk_chart <- function(data, risk_level) {
  ggplot(data, aes(x = Job_Title, fill = Education_Level)) +
    geom_bar(position = "dodge") +
    labs(
      title = paste("Education Level by Job Title for", risk_level, "Risk Category"),
      x = "Job Title",
      y = "Count",
      fill = "Education Level"
    ) +
    # Ensure all three charts use the same y-axis scale for direct comparison
    # (You may need to manually adjust the y-limit if the largest count is very high)
```

```
    # coord_cartesian(ylim = c(0, max(df_top_10 %>% count(Job_Title) %>% pull(n)))) +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}


# ================================================================================
# Generate and Print the 3 Separate Charts
# ================================================================================

# Chart 1: Low Risk
low_risk_chart <- generate_risk_chart(low_risk_df, "Low")
print(low_risk_chart)
```

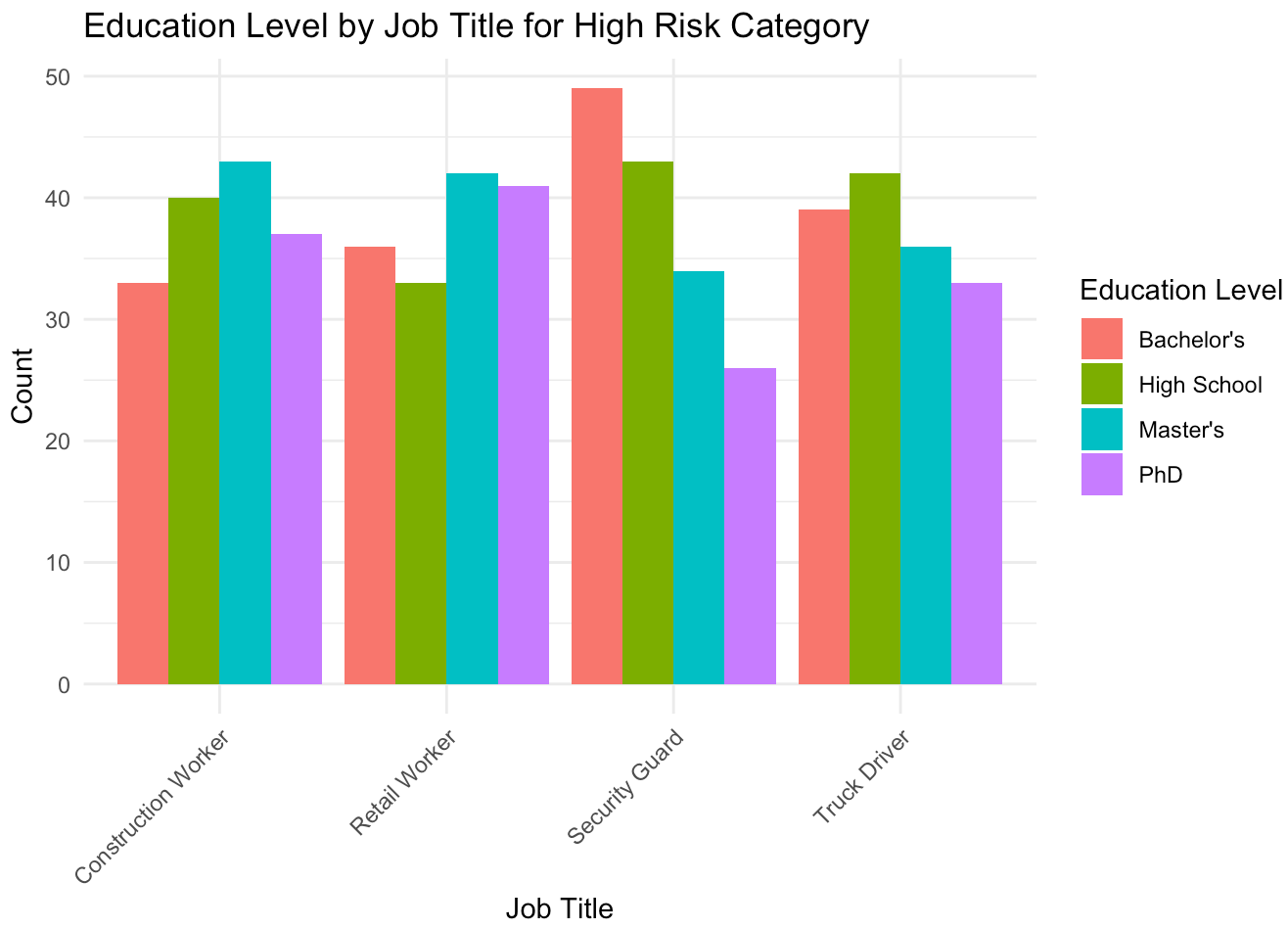



```
# Chart 2: Medium Risk
medium_risk_chart <- generate_risk_chart(medium_risk_df, "Medium")
```

```
print(medium_risk_chart)
```

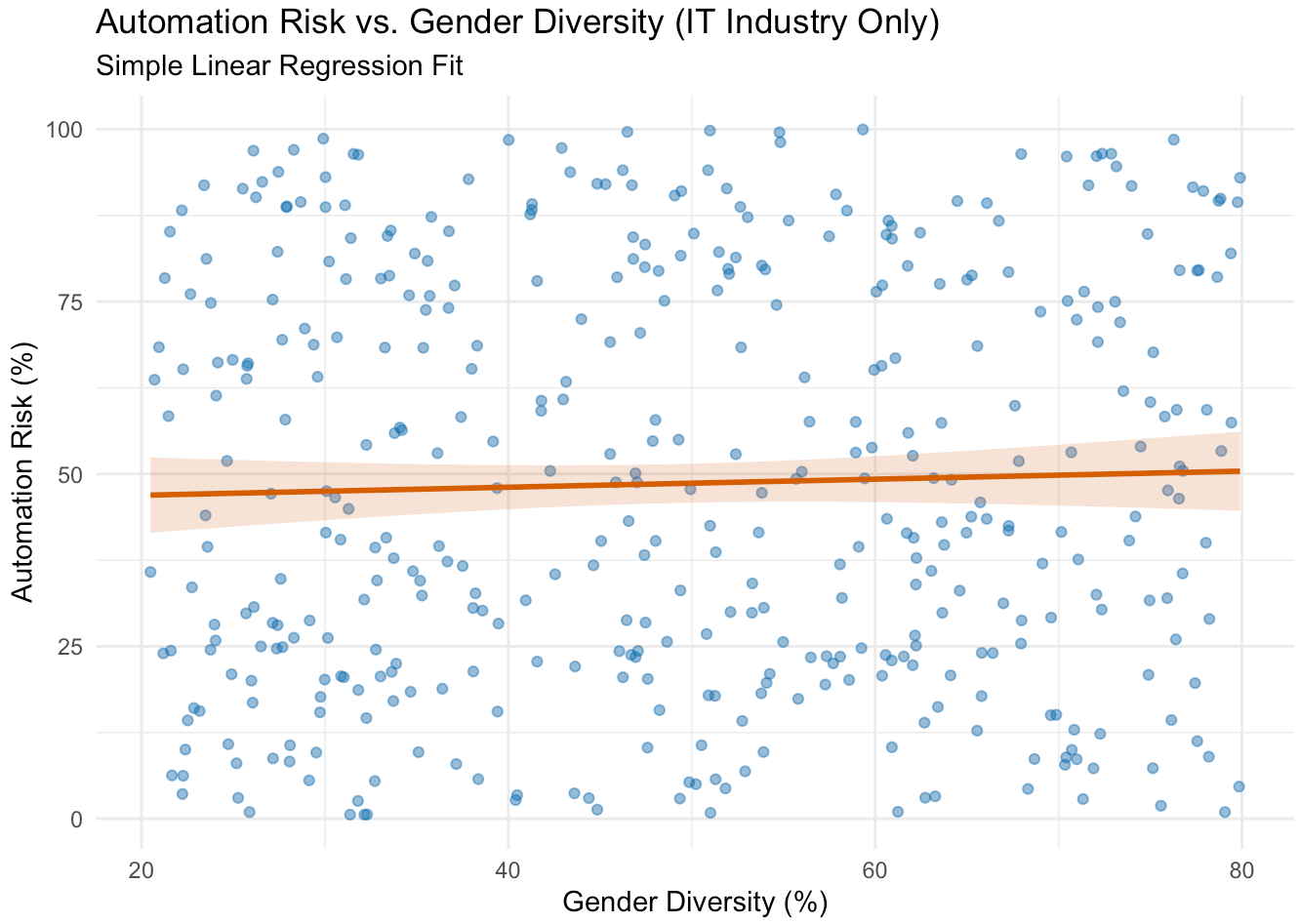



```
# Chart 3: High Risk
high_risk_chart <- generate_risk_chart(high_risk_df, "High")
print(high_risk_chart)
```



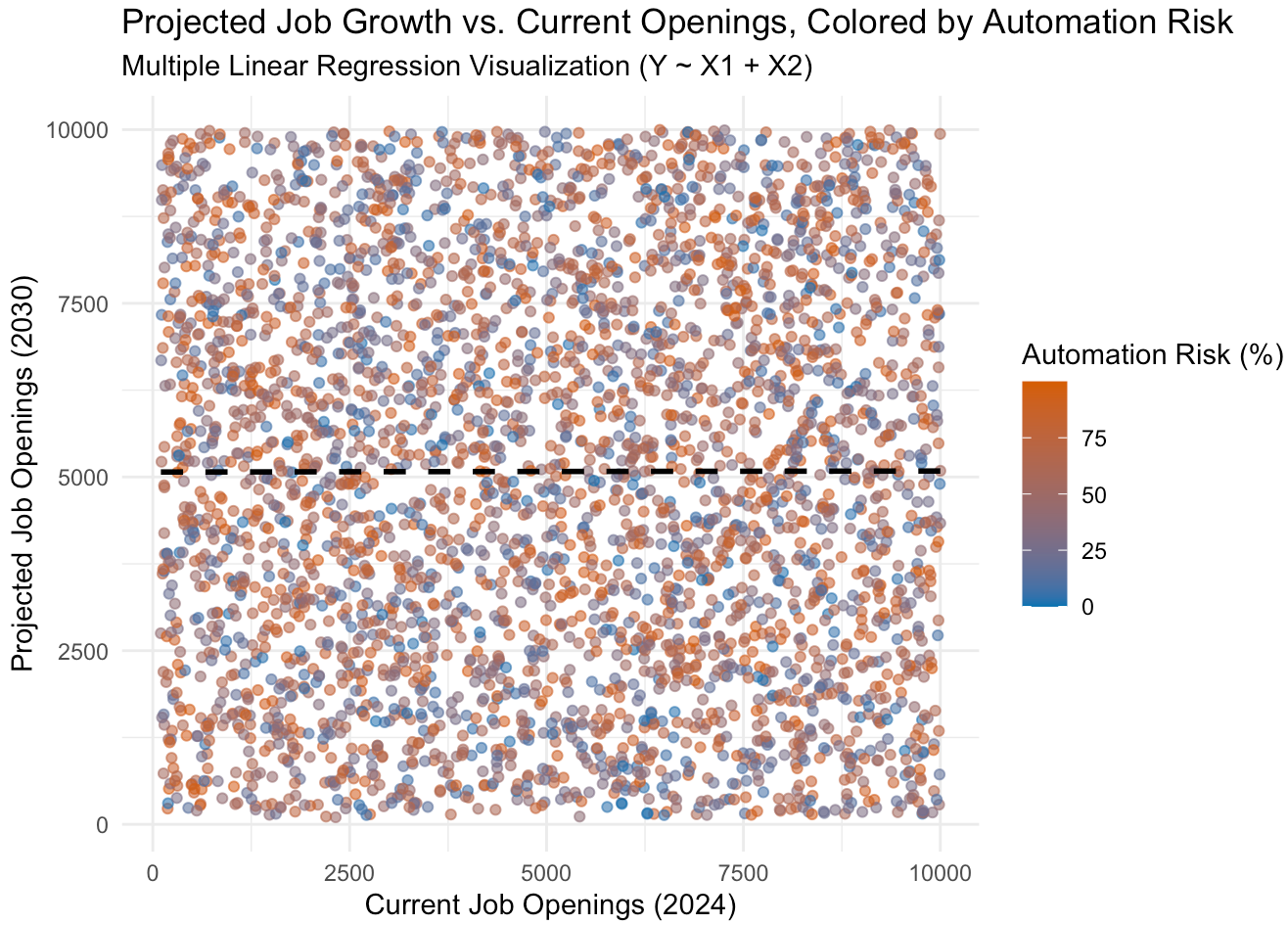

```
# 1. Load the necessary library
library(tidyverse)


# 2. Read the data file
# Make sure the file "ai_job_trends_dataset.csv" is in your working directory.
df <- read_csv("/Users/lenagray/Desktop/Stat and Data Analy/final/ai_job_trends_dataset.csv")
## Rows: 3713 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (6): Job Title, Industry, Job Status, AI Impact Level, Required Educatio...
## dbl (7): Median Salary (USD), Experience Required (Years), Job Openings (202...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
# 3. Data Filtering and Preparation
```

```
df_plot_ready <- df %>%
  # Filter the data for IT Industry only
  filter(Industry == "IT") %>%

  # Select the required continuous columns and remove any rows with missing values (NA
)
  select(
    `Automation Risk (%)`,
    `Gender Diversity (%)`
  ) %>%
  drop_na()

# 4. Create the Linear Regression Visualization for IT Jobs
visualization_it_filtered <- df_plot_ready %>%
  ggplot(aes(x = `Gender Diversity (%)`, y = `Automation Risk (%)`)) +

  # Scatter plot of the data points
  geom_point(alpha = 0.5, color = "#0072B2") +

  # Add the fitted linear regression line (with 95% confidence interval)
  geom_smooth(method = "lm", color = "#D55E00", fill = "#D55E00", alpha = 0.2) +

  labs(
    title = "Automation Risk vs. Gender Diversity (IT Industry Only)",
    subtitle = "Simple Linear Regression Fit",
    x = "Gender Diversity (%)",
    y = "Automation Risk (%)"
  ) +
  theme_minimal()

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




```
# 1. Load the necessary library
library(tidyverse)


# 2. Read the data file
df <- read_csv("/Users/lenagray/Desktop/Stat and Data Analy/final/ai_job_trends_datase
t.csv")
## Rows: 3713 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (6): Job Title, Industry, Job Status, AI Impact Level, Required Educatio...
## dbl (7): Median Salary (USD), Experience Required (Years), Job Openings (202...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
# Select the required continuous columns and remove any rows with missing values (NA)
df_plot_ready <- df %>%
```

```
  select(
    `Projected Openings (2030)`,
    `Job Openings (2024)`,
    `Automation Risk (%)`
  ) %>%
  drop_na()

# 4. Create the Multiple Linear Regression Visualization
visualization <- df_plot_ready %>%
  ggplot(aes(x = `Job Openings (2024)`, y = `Projected Openings (2030)`, color = `Auto
mation Risk (%)`)) +

  # Scatter plot: Color is mapped to Automation Risk (%)
  geom_point(alpha = 0.6) +

  # Add the fitted linear regression line (overall model trend)
  geom_smooth(method = "lm", color = "black", linetype = "dashed", se = FALSE) +

  # Define the color scale (low risk = blue, high risk = red)
  scale_color_gradient(low = "#0072B2", high = "#D55E00") +

  labs(
    title = "Projected Job Growth vs. Current Openings, Colored by Automation Risk",
    subtitle = "Multiple Linear Regression Visualization (Y ~ X1 + X2)",
    x = "Current Job Openings (2024)",
    y = "Projected Job Openings (2030)",
    color = "Automation Risk (%)"
  ) +
  theme_minimal()

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


Projected Job Growth vs. Current Openings, Colored by Automation Risk
Multiple Linear Regression Visualization (Y ~ X1 + X2)
Projected Job Openings (2030)
10000
7500
5000
2500
0
0
2500
5000
7500
10000
Current Job Openings (2024)
Automation Risk (%)
75
50
25
0