Artificial Intelligence Opinion Survey

DATA 490 Independent Study

Mentor: Dr. Nicholas Dietrich
Aadarsha Gopala Reddy, Darren Lo,
Ethan Love, Jose Mancilla, Santo Sumo

April 20, 2023

Contents

1. Load Data

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.4     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)

# Load data. Top row is column name.
edu <- read.csv("prolific_edu.csv")
health <- read.csv("prolific_health.csv")
retail <- read.csv("prolific_retail.csv")
tech <- read.csv("prolific_tech.csv")
qualtrics <- read.csv("qualtrics_data.csv")
```

2. Data Cleaning

```r
# Combine data into one data frame after mutating Age to be one data type
edu <- edu %>% mutate(Age = as.character(Age))
health <- health %>% mutate(Age = as.character(Age))
retail <- retail %>% mutate(Age = as.character(Age))
tech <- tech %>% mutate(Age = as.character(Age))

combined <- bind_rows(edu, health, retail, tech)
# export combined data to csv
# write.csv(combined, "combined_non_qualtrics.csv")
```

```r
# combine qualtrics and combined data using qualtrics data's
#     ProlificID column and combined data's Participant id
combined <- left_join(qualtrics, combined,
  by = c("ProlificID" = "Participant.id")
)
# rename "Duration..in.seconds." column to "Duration"
colnames(combined)[
  colnames(combined) == "Duration..in.seconds."
] <- "Duration"
# remove Age.x and keep only Age.y column and rename Age.y to Age
combined <- combined %>%
  select(-Age.x) %>%
  rename(Age = Age.y)
# remove Status.x and Status.y columns
combined <- combined %>%
  select(-Status.x) %>%
  select(-Status.y)
# remove Finished, Progress, UserLanguage, DistributionChannel,
# Nationality, and Consent columns
combined <- combined %>%
  select(-Finished) %>%
  select(-Progress) %>%
  select(-UserLanguage) %>%
  select(-DistributionChannel) %>%
  select(-Nationality) %>%
  select(-Consent)


# remove rows where Submission.id is NA
combined <- combined %>%
  filter(!is.na(Submission.id))
# Keep only rows which say "United States" in "Country.of.residence" column
combined <- combined %>%
  filter(combined$Country.of.residence == "United States")
# Replace all cells that say "Information Technology" and
# "Science, Technology, Engineering & Mathematics" to "STEM/IT"
# in Employment.sector column
combined$Employment.sector[
  combined$Employment.sector == "Information Technology"
] <- "STEM/IT"
combined$Employment.sector[
  combined$Employment.sector == "Science, Technology, Engineering & Mathematics"
```

```
] <- "STEM/IT"

# change all cells in column EnhanceHurt that say
# "AI will neither enhance nor detract from my work" to "neither",
# "AI will enhance my work" to "enhance", and
# "AI will detract from my work" to "detract"
combined$EnhanceHurt[
  combined$EnhanceHurt == "AI will neither enhance nor detract from my work"
] <- "neither"
combined$EnhanceHurt[
  combined$EnhanceHurt == "AI will enhance my work"
] <- "enhance"
combined$EnhanceHurt[
  combined$EnhanceHurt == "AI will detract from my work"
] <- "detract"

# export data to csv
# write.csv(combined, "combined_qualtrics.csv")
```

```
# Keep only rows which say "Compose an email" in "Attention" column
combined <- combined %>% filter(combined$Attention == "Compose an email")
# remove Attention column
combined <- combined %>% select(-Attention)
# export data to csv
# write.csv(combined, "combined_qualtrics_attentive.csv")
```

3. Data Exploration

The columns in the dataset are:

- *StartDate* - Date and time survey was started
- *EndDate* - Date and time survey was completed
- *IPAddress* - IP address of participant
- *Duration* - Duration of survey in seconds
- *RecordedDate* - Date and time survey was recorded
- *ResponseId* - Response ID
- *LocationLatitude* - Participant's location latitude
- *LocationLongitude* - Participant's location longitude
- *ProlificID* - Identification of the response on Prolific
- *Gender* - Gender of the participant
- *Education* - Education level of the participant
- *Salary* - Salary of the participant
- *AIKnowledge* - Knowledge of AI of the participant
- *UsedAI* - Whether the participant has used AI
- *TimeEnergy* - How much time and energy AI has saved the participant
- *SimilarTasks* - How much of the participant's tasks they think AI can do

- *EnhanceHurt* - Whether the participant thinks AI can enhance or hurt their work efficiency.
- *Comments* - Comments from the participant
- *Submission.id* - Submission ID
- *Started.at* - Date and time survey was started
- *Completed.at* - Date and time survey was completed
- *Reviewed.at* - Date and time survey was reviewed
- *Archived.at* - Date and time survey was archived
- *Time.taken* - Duration of survey in seconds
- *Completion.code* - Completion code
- *Total.approvals* - Total number of approvals
- *Employment.sector* - Employment sector
- *Age* - Age of the participant
- *Sex* - Sex of the participant
- *Ethnicity.simplified* - Ethnicity of the participant
- *Country.of.birth* - Country of birth of the participant
- *Country.of.residence* - Country of residence of the participant
- *Language* - Language of the participant
- *Student.status* - Whether the participant is a student
- *Employment.status* - Whether the participant is employed

4. Data Analysis

4.1. Average amount of time to complete survey

```
# Create a new data frame with only the columns we need
avg_time <- combined %>% select(Duration)
# Remove rows where Duration is NA
avg_time <- avg_time %>% filter(!is.na(Duration))
# Calculate the average time taken to complete the survey
avg_time <- avg_time %>% summarise(avg_time = mean(Duration))
# Convert to <x> minutes and <y> seconds
avg_time$avg_time <- avg_time$avg_time / 60
avg_time$avg_time <- paste0(
  floor(avg_time$avg_time),
  " minutes and ",
  round((avg_time$avg_time - floor(avg_time$avg_time)) * 60),
  " seconds"
)
# Print the average time taken to complete the survey
avg_time
```

```
##                   avg_time
## 1 2 minutes and 13 seconds
```

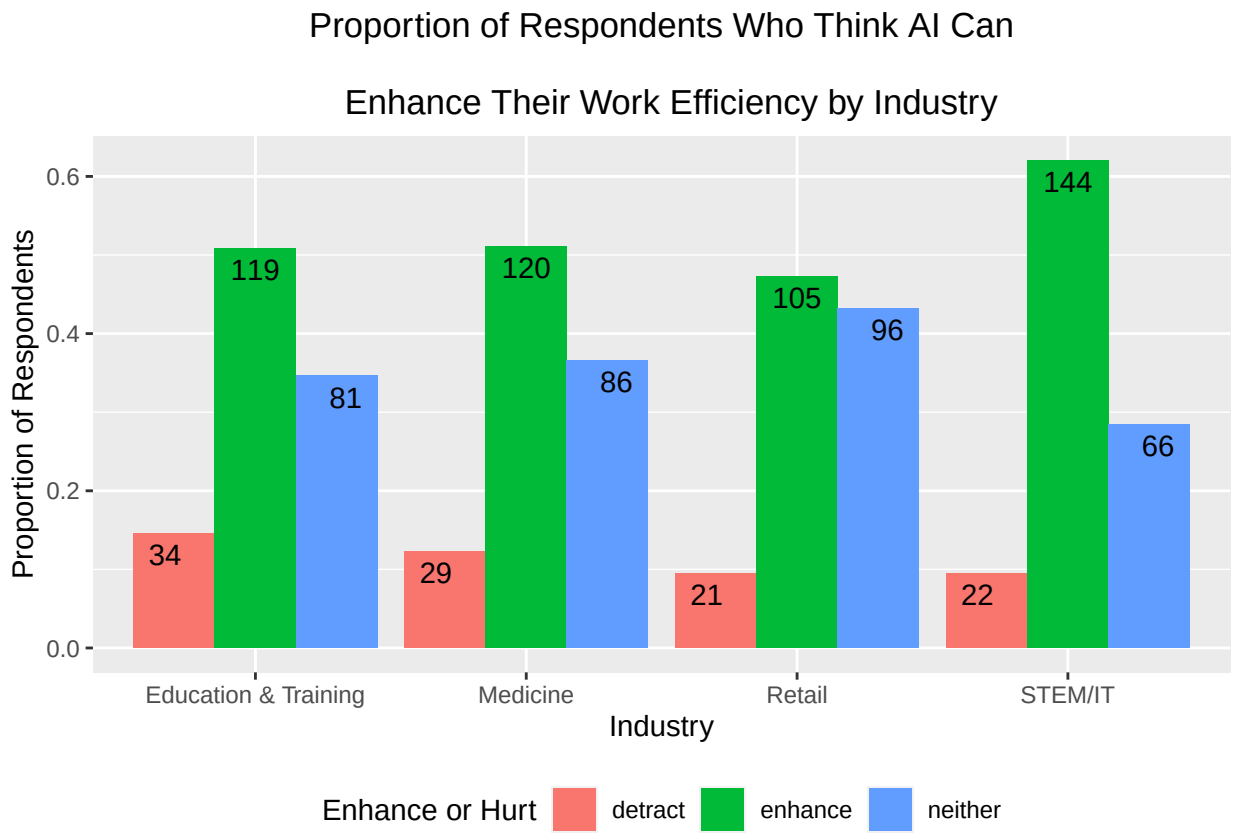
4.2. EnhanceHurt vs. Industry

```
# Create a new data frame with only the columns we need
enhancehurt_vs_industry <- combined %>%
  select(EnhanceHurt, Employment.sector)
```

```
# Remove rows where Employment.sector is NA
enhancehurt_vs_industry <- enhancehurt_vs_industry %>%
  filter(!is.na(Employment.sector))
# Remove rows where EnhanceHurt is NA
enhancehurt_vs_industry <- enhancehurt_vs_industry %>%
  filter(!is.na(EnhanceHurt))

# For each industry, calculate the proportion of
# respondents who think AI can enhance their work efficiency
enhancehurt_vs_industry <- enhancehurt_vs_industry %>%
  group_by(Employment.sector, EnhanceHurt) %>%
  summarize(count = n()) %>%
  mutate(prop = count / sum(count))
```

```
## `summarise()` has grouped output by 'Employment.sector'. You can override using
## the `.groups` argument.
```

```
# Visualize using different histogram for each industry.
# Show the number of respondents inside each bar.
# make legend bottom. Wrap x axis labels without changing plot size.
enhancehurt_vs_industry_plot <- ggplot(
  enhancehurt_vs_industry,
  aes(
    x = Employment.sector,
    y = prop, fill = EnhanceHurt
  )
) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = count),
    position = position_dodge(width = 1),
    vjust = 1.5
  ) +
  labs(
    x = "Industry", y = "Proportion of Respondents",
    fill = "Enhance or Hurt"
  ) +
  ggtitle("Proportion of Respondents Who Think AI Can\n
  Enhance Their Work Efficiency by Industry") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.position = "bottom"
  )

enhancehurt_vs_industry_plot
```



4.3. EnhanceHurt vs. Education

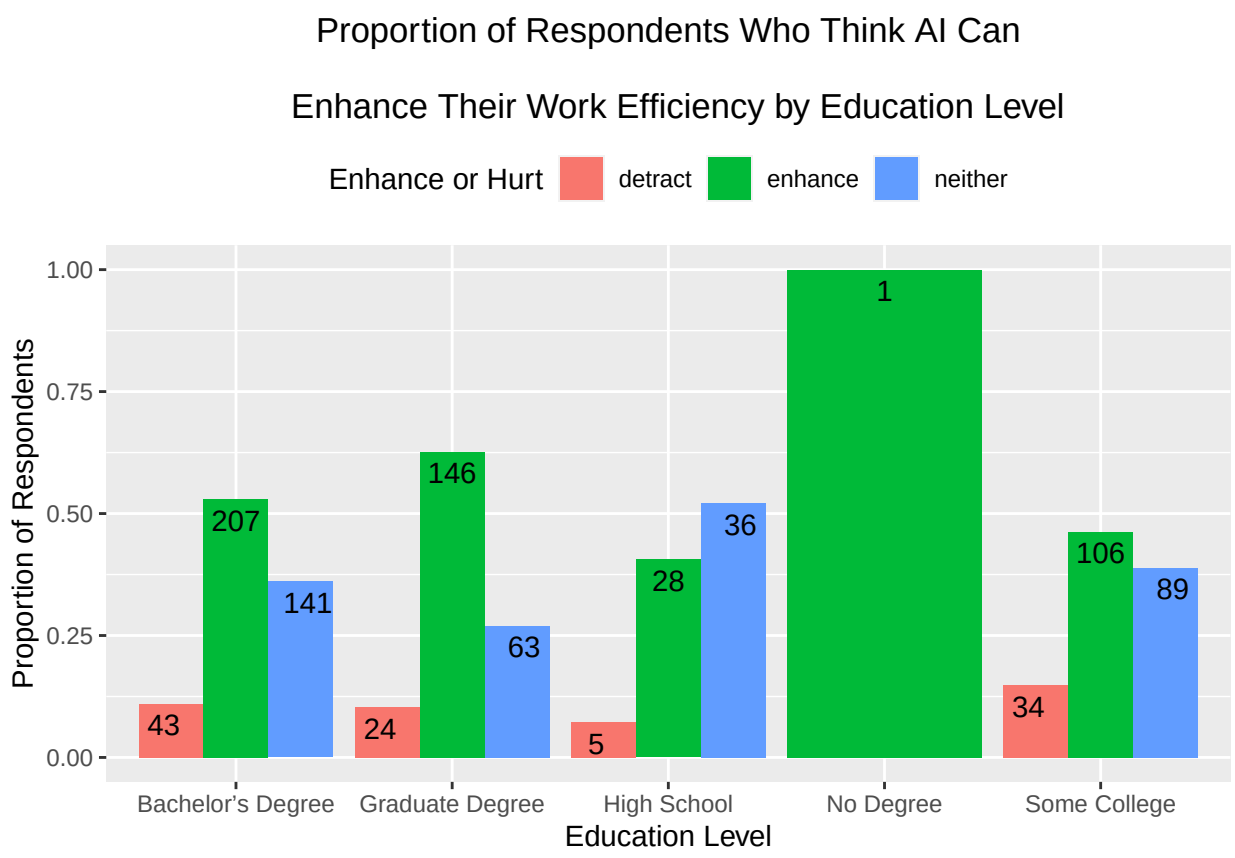
```
# Create a new data frame with only the columns we need
enhancehurt_vs_education <- combined %>%
  select(EnhanceHurt, Education)
```

```
# Remove rows where Education is NA
enhancehurt_vs_education <- enhancehurt_vs_education %>%
  filter(!is.na(Education))
# Remove rows where EnhanceHurt is NA
enhancehurt_vs_education <- enhancehurt_vs_education %>%
  filter(!is.na(EnhanceHurt))

# For each education level, calculate the proportion of
# respondents who think AI can enhance their work efficiency
enhancehurt_vs_education <- enhancehurt_vs_education %>%
  group_by(Education, EnhanceHurt) %>%
  summarize(count = n()) %>%
  mutate(prop = count / sum(count))
```

```
## `summarise()` has grouped output by 'Education'. You can override using the
## `.groups` argument.
```

```
# Visualize using different histogram for each education level.
# Show the number of respondents on each bar.
enhancehurt_vs_education_plot <- ggplot(enhancehurt_vs_education, aes(
```

```
  x = Education, y = prop,
  fill = EnhanceHurt
)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = count),
    position = position_dodge(width = 1),
    vjust = 1.5
  ) +
  labs(
    x = "Education Level", y = "Proportion of Respondents",
    fill = "Enhance or Hurt"
  ) +
  ggtitle("Proportion of Respondents Who Think AI Can\n
  Enhance Their Work Efficiency by Education Level") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(legend.position = "top")

enhancehurt_vs_education_plot
```



4.4. EnhanceHurt vs. Age

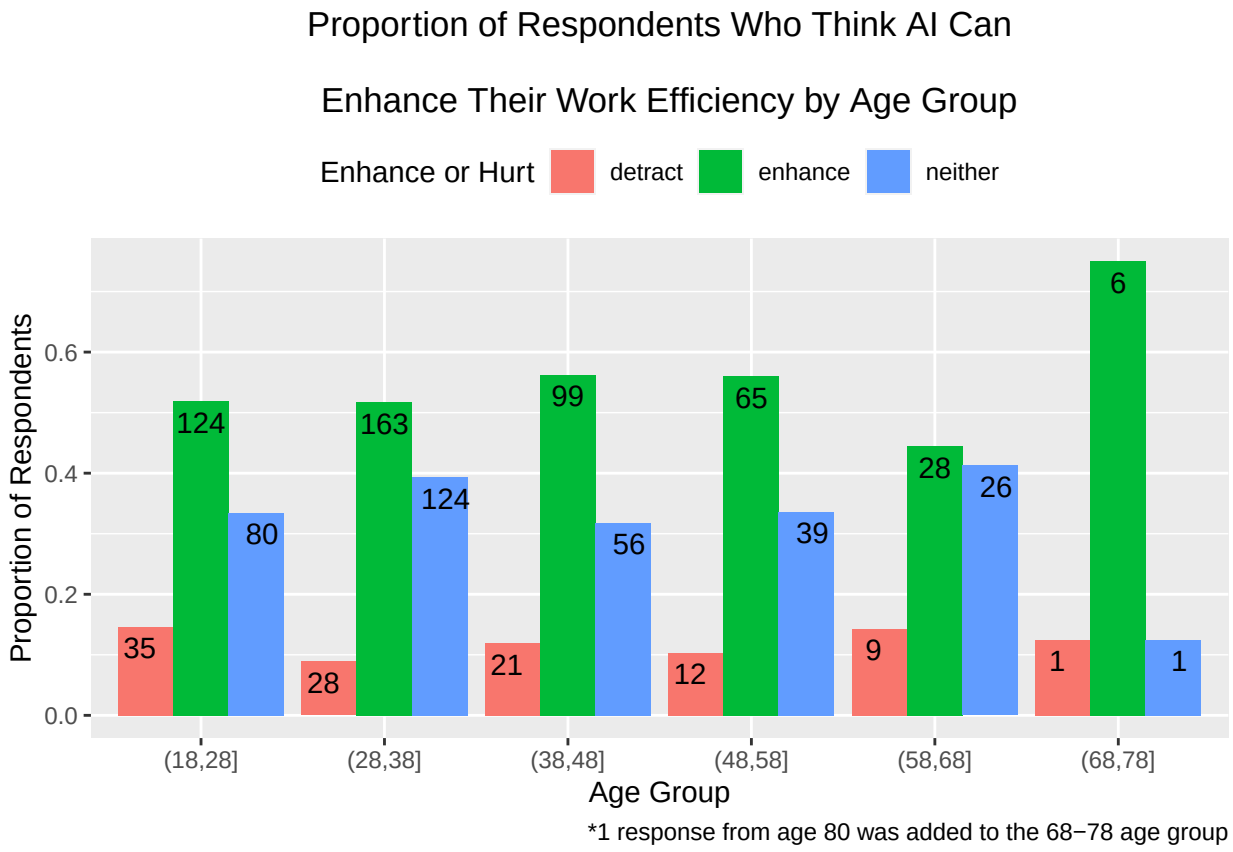
```
# Create a new data frame with only the columns we need
enhancehurt_vs_age <- combined %>% select(EnhanceHurt, Age)
```

```r
# Remove the Ages which say "DATA_EXPIRED" and "923"
enhancehurt_vs_age <- enhancehurt_vs_age %>%
  filter(enhancehurt_vs_age$Age != "DATA_EXPIRED")
enhancehurt_vs_age <- enhancehurt_vs_age %>%
  filter(enhancehurt_vs_age$Age != "923")
# Remove rows where Age is NA
enhancehurt_vs_age <- enhancehurt_vs_age %>%
  filter(!is.na(Age))
# Remove rows where EnhanceHurt is NA
enhancehurt_vs_age <- enhancehurt_vs_age %>%
  filter(!is.na(EnhanceHurt))

# Change age rows which say "79" and "80" to "73"
enhancehurt_vs_age$Age[enhancehurt_vs_age$Age == "79"] <- "73"
enhancehurt_vs_age$Age[enhancehurt_vs_age$Age == "80"] <- "73"

# Convert Age to numeric
enhancehurt_vs_age$Age <- as.numeric(as.character(enhancehurt_vs_age$Age))
# group ages by 5 years
enhancehurt_vs_age$Age <- cut(enhancehurt_vs_age$Age,
  breaks = seq(18, 80, by = 10)
)
# remove NA
enhancehurt_vs_age <- enhancehurt_vs_age %>% filter(!is.na(Age))

# create a new dataframe with the proportion of respondents who think AI can enhance their work efficie
enhancehurt_vs_age <- enhancehurt_vs_age %>%
  group_by(Age, EnhanceHurt) %>%
  summarize(count = n()) %>%
  mutate(prop = count / sum(count))
```

```
## `summarise()` has grouped output by 'Age'. You can override using the `.groups`
## argument.
```

```r
# Visualize using different histogram for each age group.
# Show the number of respondents on each bar.
enhancehurt_vs_age_plot <- ggplot(enhancehurt_vs_age, aes(
  x = Age, y = prop,
  fill = EnhanceHurt
)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = count),
    position = position_dodge(width = 1),
    vjust = 1.5
  ) +
  labs(
    x = "Age Group",
    y = "Proportion of Respondents",
    fill = "Enhance or Hurt",
    caption = "*1 response from age 80 was added to the 68-78 age group"
  ) +
  ggtitle("Proportion of Respondents Who Think AI Can\n
  Enhance Their Work Efficiency by Age Group") +
  theme(plot.title = element_text(hjust = 0.5)) +
```

```
  theme(legend.position = "top")

enhancehurt_vs_age_plot
```



4.5. EnhanceHurt vs. Salary

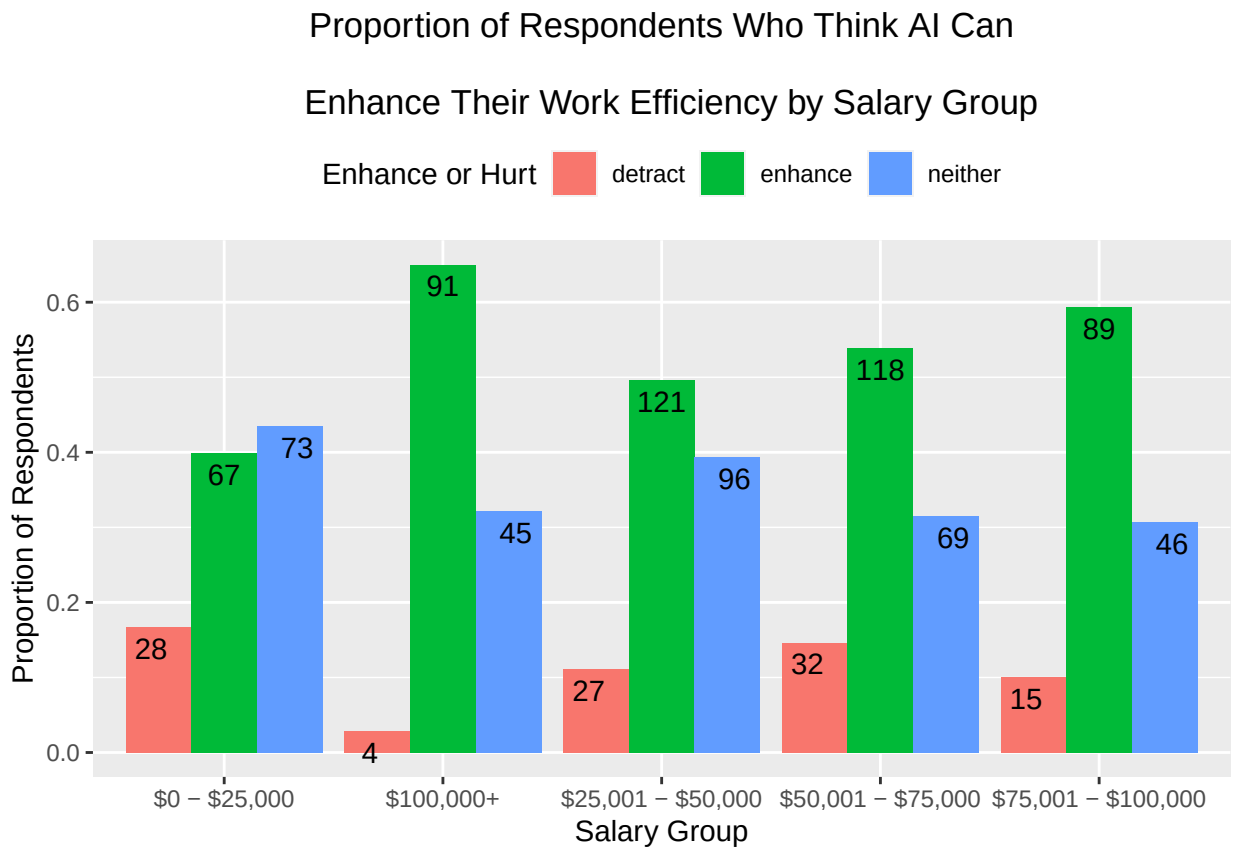
```
# Create a new data frame with only the columns we need
enhancehurt_vs_salary <- combined %>% select(EnhanceHurt, Salary)
```

```
# Remove rows where Salary is NA
enhancehurt_vs_salary <- enhancehurt_vs_salary %>%
  filter(!is.na(Salary))
# Remove rows where EnhanceHurt is NA
enhancehurt_vs_salary <- enhancehurt_vs_salary %>%
  filter(!is.na(EnhanceHurt))

# remove rows where Salary is empty
enhancehurt_vs_salary <- enhancehurt_vs_salary %>%
  filter(Salary != "")

# Create a new dataframe with the proportion of respondents who
# think AI can enhance their work efficiency for each salary group
enhancehurt_vs_salary <- enhancehurt_vs_salary %>%
  group_by(Salary, EnhanceHurt) %>%
  summarize(count = n()) %>%
```

```
  mutate(prop = count / sum(count))
```

```
## `summarise()` has grouped output by 'Salary'. You can override using the
## `.groups` argument.
```

```
# Visualize using different histogram for each salary group.
# Show the number of respondents on each bar.
enhancehurt_vs_salary_plot <- ggplot(enhancehurt_vs_salary, aes(
  x = Salary, y = prop,
  fill = EnhanceHurt
)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = count),
    position = position_dodge(width = 1),
    vjust = 1.5
  ) +
  labs(
    x = "Salary Group",
    y = "Proportion of Respondents",
    fill = "Enhance or Hurt"
  ) +
  ggtitle("Proportion of Respondents Who Think AI Can\n
  Enhance Their Work Efficiency by Salary Group") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(legend.position = "top")

enhancehurt_vs_salary_plot
```



4.6. EnhanceHurt vs. Gender

```
# Create a new data frame with only the columns we need
enhancehurt_vs_gender <- combined %>% select(EnhanceHurt, Gender)
```

```
# Remove rows where Gender is NA
enhancehurt_vs_gender <- enhancehurt_vs_gender %>%
  filter(!is.na(Gender))
# Remove rows where EnhanceHurt is NA
enhancehurt_vs_gender <- enhancehurt_vs_gender %>%
  filter(!is.na(EnhanceHurt))

table(enhancehurt_vs_gender$Gender)
```

```
##
##                  Female                    Male Prefer to self-describe:
##                     462                     451                          10
```

```
# remove rows where Gender is empty
enhancehurt_vs_gender <- enhancehurt_vs_gender %>%
  filter(Gender != "Prefer to self-describe")

# Create a new dataframe with the proportion of respondents who
# think AI can enhance their work efficiency for each gender
enhancehurt_vs_gender <- enhancehurt_vs_gender %>%
  group_by(Gender, EnhanceHurt) %>%
```

```
  summarize(count = n()) %>%
  mutate(prop = count / sum(count))
```

```
## `summarise()` has grouped output by 'Gender'. You can override using the
## `.groups` argument.
```