

# Data Analysis

[Code ▼](#)

Ankit Grover

This is an R Markdown (<http://rmarkdown.rstudio.com/>) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

[Hide](#)

```
library(tidyverse) # Dataframe manipulation
```

```
Warning: package 'tidyverse' was built under R version 4.4.1Warning: package 'tidyr' was built under R version 4.4.1— Attaching core tidyverse packages —————
— tidyverse 2.0.0 —
✓ dplyr      1.1.4    ✓ readr      2.1.5
✓ forcats    1.0.0    ✓ stringr    1.5.1
✓ ggplot2    3.4.4    ✓ tibble     3.2.1
✓ lubridate  1.9.3    ✓ tidyr      1.3.1
✓ purrr      1.0.2    — Conflicts —————
- tidyverse_conflicts() —
✗ dplyr::filter() masks stats::filter()
✗ dplyr::lag()    masks stats::lag()
i Use the [?]8;;http://conflicted.r-lib.org/[?]8;;[?] to force all conflicts to become errors
```

[Hide](#)

```
library(ggplot2) # Plot
library(dplyr) # For pipe uage
library(datarium)
```

```
Warning: package 'datarium' was built under R version 4.4.1
```

[Hide](#)

```
library(summarytools) #Summary stuff
```

Warning: couldn't connect to display ":0"system might not have X11 capabilities; in case of errors when using dfSummary(), set st\_options(use.x11 = FALSE)

Attaching package: 'summarytools'

The following object is masked from 'package:tibble':

view

Hide

```
library(skimr)
library(corrplot)
```

Warning: package 'corrplot' was built under R version 4.4.1corrplot 0.95 loaded

Hide

```
# I have no idea why require is used instead of library ngl.
require(summarytools)
require(skimr)
require(psych)
```

Loading required package: psych

Attaching package: 'psych'

The following objects are masked from 'package:ggplot2':

%+%, alpha

Hide

```
require(rstatix)
```

Loading required package: rstatix

Attaching package: 'rstatix'

The following object is masked from 'package:stats':

filter

Hide

```
require(emmeans)
```

```
Loading required package: emmeans  
Welcome to emmeans.  
Caution: You lose important information if you filter this package's results.  
See '? untidy'
```

[Hide](#)

```
require(car)
```

```
Loading required package: car  
Loading required package: carData  
  
Attaching package: 'car'  
  
The following object is masked from 'package:psych':  
  
  logit  
  
The following object is masked from 'package:dplyr':  
  
  recode  
  
The following object is masked from 'package:purrr':  
  
  some
```

[Hide](#)

```
require(multcomp)
```

```
Loading required package: multcomp
Loading required package: mvtnorm
Loading required package: survival
Loading required package: TH.data
Loading required package: MASS
```

```
Attaching package: 'MASS'
```

```
The following object is masked from 'package:rstatix':
```

```
  select
```

```
The following object is masked from 'package:dplyr':
```

```
  select
```

```
Attaching package: 'TH.data'
```

```
The following object is masked from 'package:MASS':
```

```
  geyser
```

Hide

```
require(pastecs)
```

```
Loading required package: pastecs
```

```
Attaching package: 'pastecs'
```

```
The following objects are masked from 'package:dplyr':
```

```
  first, last
```

```
The following object is masked from 'package:tidyr':
```

```
  extract
```

Hide

```
require(gmodels)
```

```
Loading required package: gmodels
```

Hide

```
data <- read.csv("hororr_dataset.csv")
View(data)
```

[Hide](#)

```
supply(data,class) # Take a look at the type of variable first
```

id	p_age	p_gender	p_experience	r_model
"integer"	"integer"	"character"	"integer"	"character"
r_errors	interaction_length	trust	liking	agency
"character"	"numeric"	"integer"	"integer"	"integer"
interact_again	distance			
"character"	"numeric"			

[Hide](#)

```
skim(data) # Interesting we don't have missing data so no need to impute.
```

#### — Data Summary —

	Values
Name	data
Number of rows	200
Number of columns	12
Column type frequency:	
character	4
numeric	8
Group variables	
	None

skim_variable <chr>	n_missing <int>	complete_rate <dbl>	m.. <int>	m.. <int>	empty <int>	n_unique <int>	whitespace <int>
1 p_gender	0	1	1	3	0	5	0
2 r_model	0	1	8	14	0	3	0
3 r_errors	0	1	5	8	0	2	0
4 interact_again	0	1	2	3	0	2	0
4 rows							

skim_variable <chr>	n_missing <int>	complete_rate <dbl>	mean <dbl>	sd <dbl>	p0 <dbl>	p25 <dbl>
1 id	0	1	100.50000	57.879185	1.000000	50.750000
2 p_age	0	1	33.11000	10.693439	18.000000	24.000000
3 p_experience	0	1	4.13500	1.512685	1.000000	3.000000
4 interaction_length	0	1	11.81080	3.152829	3.647874	9.758559
5 trust	0	1	4.40500	1.248507	1.000000	3.000000

skim_variable <chr>	n_missing <int>	complete_rate <dbl>	mean <dbl>	sd <dbl>	p0 <dbl>	p25 <dbl>
6 liking	0	1	5.18500	1.252445	2.000000	4.000000
7 agency	0	1	4.54000	1.373987	1.000000	4.000000
8 distance	0	1	78.93722	20.685481	25.487588	61.745718

Hide

# I guess we need to be careful to transform our data into their respective types huh?

```
# Numeric -->Discrete(id), Continuous Vars(distance,interaction_length)
# Categorical--> Binary(interact_again,r_error),
# contd. Ordinal(p_experience, trust, liking, agency),Nominal(p_gender,r_model)
```

# Reason: Likert scale values are ordered! Age is continuous! Rest are categories.

```
data$id <- as.integer(data$id)
```

# Continuous

```
data$distance <- as.numeric(data$distance)
data$interaction_length <- as.numeric(data$interaction_length)
```

#Binary

```
data$interact_again <- factor(data$interact_again,levels=c("yes","no"))
data$r_errors <- factor(data$r_errors, levels=c("no_error","error"))
# data$interact_again <- ifelse(data$interact_again == "yes", 1, 0)
# data$r_errors <- ifelse(data$r_errors == "error", 1, 0)
```

# Ordinal

```
data$p_experience <- factor(data$p_experience, ordered = TRUE, levels = 1:7)
data$trust <- factor(data$trust, ordered = TRUE, levels = 1:7)
data$liking <- factor(data$liking, ordered = TRUE, levels = 1:7)
data$agency <- factor(data$agency, ordered = TRUE, levels = 1:7)
```

# Nominal

```
data$p_gender <- factor(data$p_gender)
data$r_model <- factor(data$r_model)
```

# model gender and interact\_again

Hide


```
skim(data)
```

#### — Data Summary —

Name	Values
data	data
Number of rows	200
Number of columns	12
Column type frequency:	
factor	8
numeric	4
Group variables	
None	


skim_variable <chr>	n_missing <int>	complete_rate <dbl>	ordered <lgl>	n_uniq... <int>	top_counts <chr>
1 p_gender	0	1	FALSE	5	M: 96, F: 88, N.R: 8, NB: 5
2 p_experience	0	1	TRUE	7	4: 57, 5: 43, 3: 33, 6: 27
3 r_model	0	1	FALSE	3	zoo: 71, mec: 66, hum: 63
4 r_errors	0	1	FALSE	2	err: 108, no_: 92
5 trust	0	1	TRUE	7	5: 60, 4: 51, 3: 41, 6: 30
6 liking	0	1	TRUE	6	5: 60, 6: 47, 4: 40, 7: 35
7 agency	0	1	TRUE	7	5: 61, 4: 41, 6: 37, 3: 33
8 interact_again	0	1	FALSE	2	yes: 121, no: 79

8 rows



skim_variable <chr>	n_missing <int>	complete_rate <dbl>	mean <dbl>	sd <dbl>	p0 <dbl>	p25 <dbl>
1 id	0	1	100.50000	57.879185	1.000000	50.750000
2 p_age	0	1	33.11000	10.693439	18.000000	24.000000
3 interaction_length	0	1	11.81080	3.152829	3.647874	9.758559
4 distance	0	1	78.93722	20.685481	25.487588	61.745718

4 rows | 1-9 of 11 columns



## Descriptive Statistics of Variables

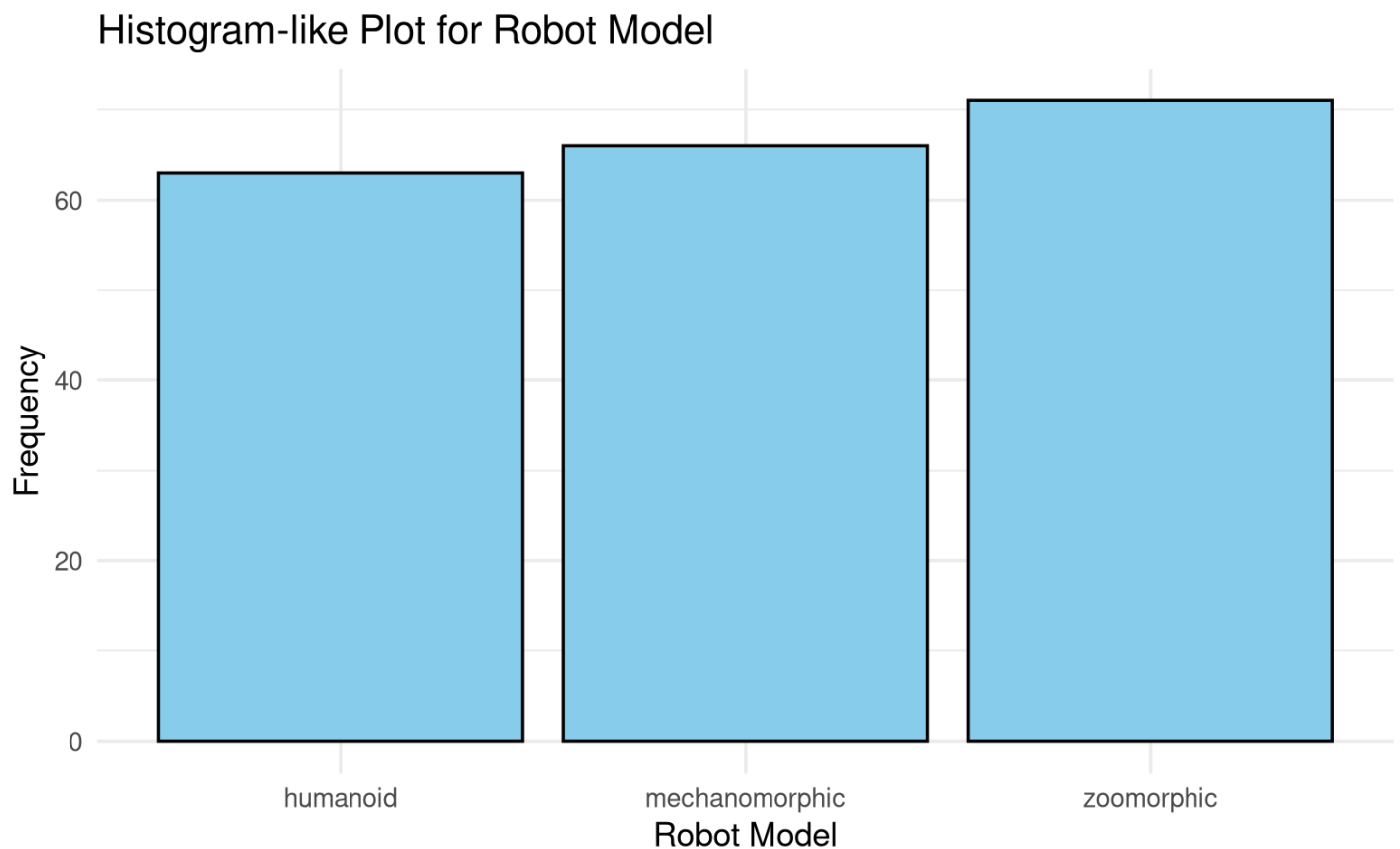
View the distribution of the 2X2 Between group variables.

We see r\_model seems to have roughly similar number of models. We have robots with more errors Hmm. This seems purposefully done to see some effect. But well that's why we have 2X2 design!

## INDPV Distributions

Hide

```
data %>%  
  ggplot(aes(x = r_model)) +  
  geom_bar(stat = "count", fill = "skyblue", color = "black") +  
  labs(title = "Histogram-like Plot for Robot Model", x = "Robot Model", y = "Frequency") +  
  theme_minimal()
```



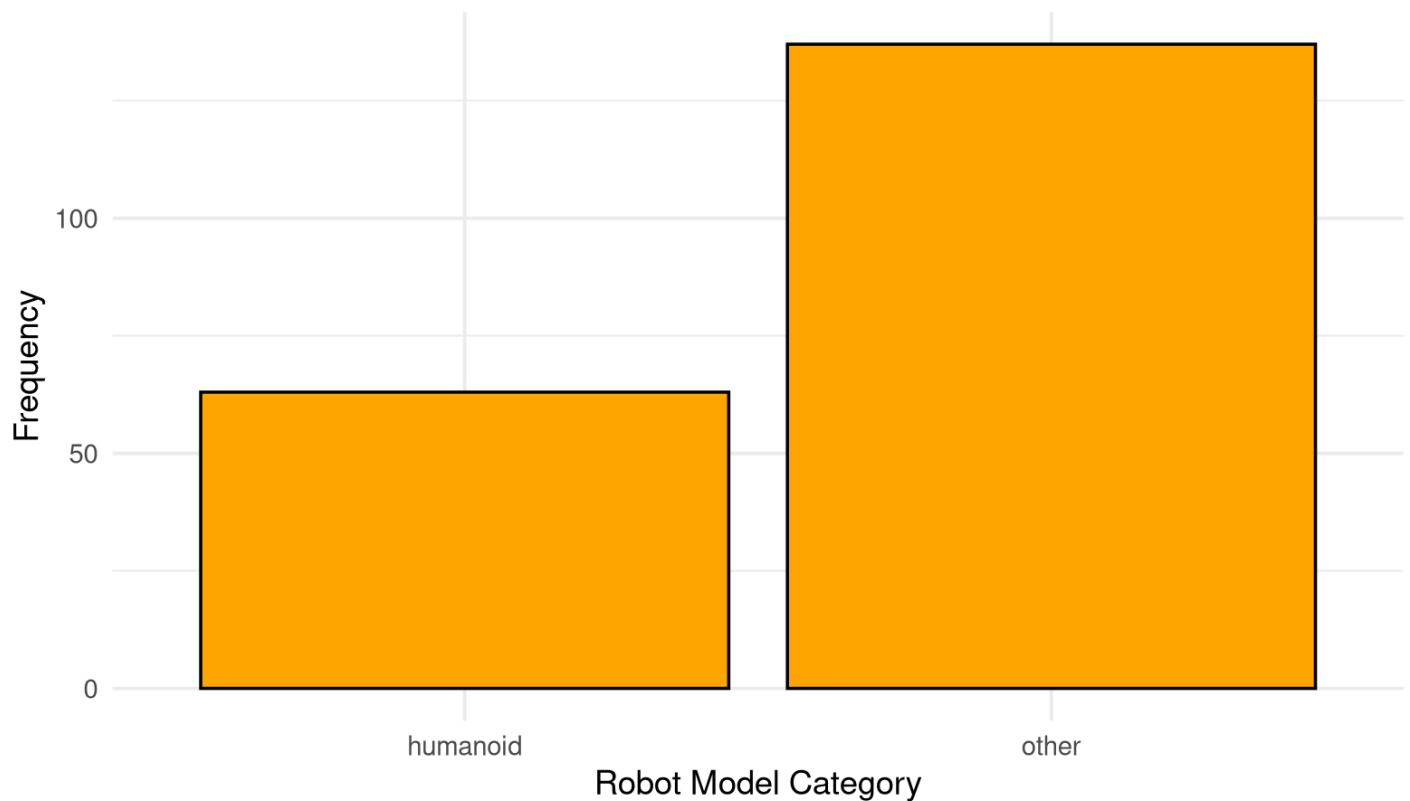
Hide



```
data <- data %>%
  mutate(robot_model = ifelse(r_model == "humanoid", "humanoid", "other"))

data %>%
  ggplot(aes(x = robot_model)) +
  geom_bar(fill = "orange", color = "black") +
  labs(title = "Count of Humanoids vs Other Robot Models", x = "Robot Model Category", y = "Frequency") +
  theme_minimal()
```

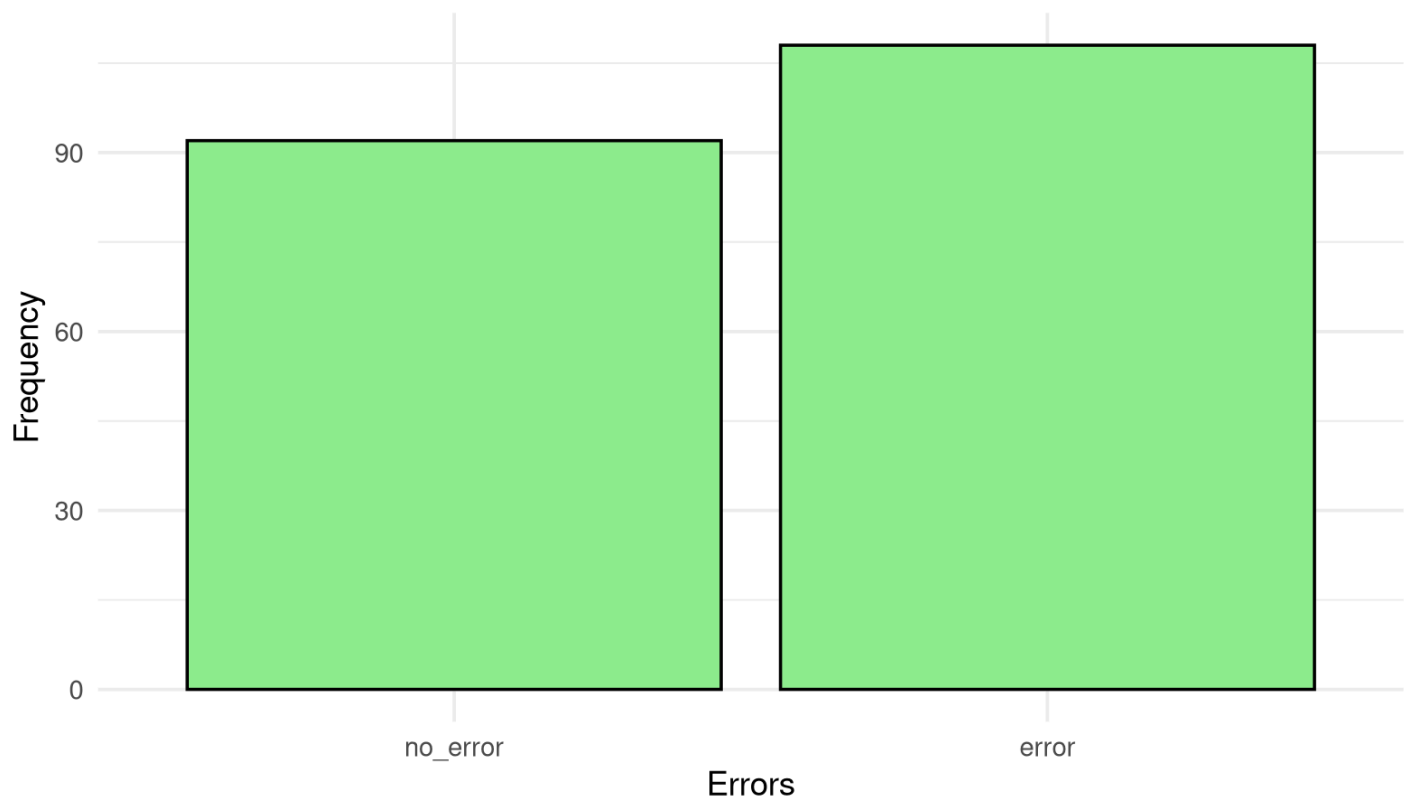
Count of Humanoids vs Other Robot Models



Hide

```
data %>%
  ggplot(aes(x = r_errors)) +
  geom_bar(stat = "count", fill = "lightgreen", color = "black") +
  labs(title = "Histogram-like Plot for Robot errors", x = "Errors", y = "Frequency") +
  theme_minimal()
```

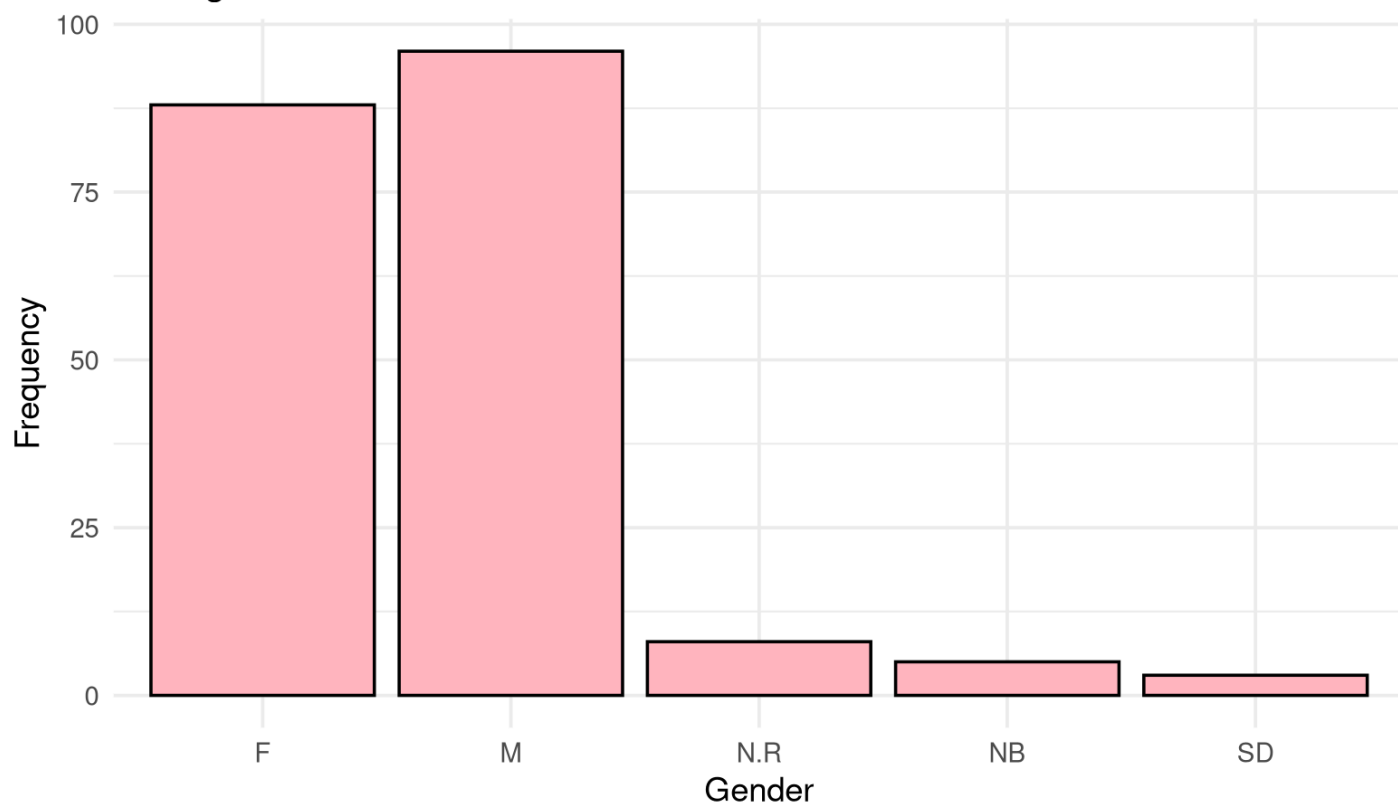
Histogram-like Plot for Robot errors



Hide

```
data %>%  
  ggplot(aes(x = p_gender)) +  
  geom_bar(stat = "count", fill = "lightpink", color = "black") +  
  labs(title = "Histogram-like Plot for Gender", x = "Gender", y = "Frequency") +  
  theme_minimal()
```

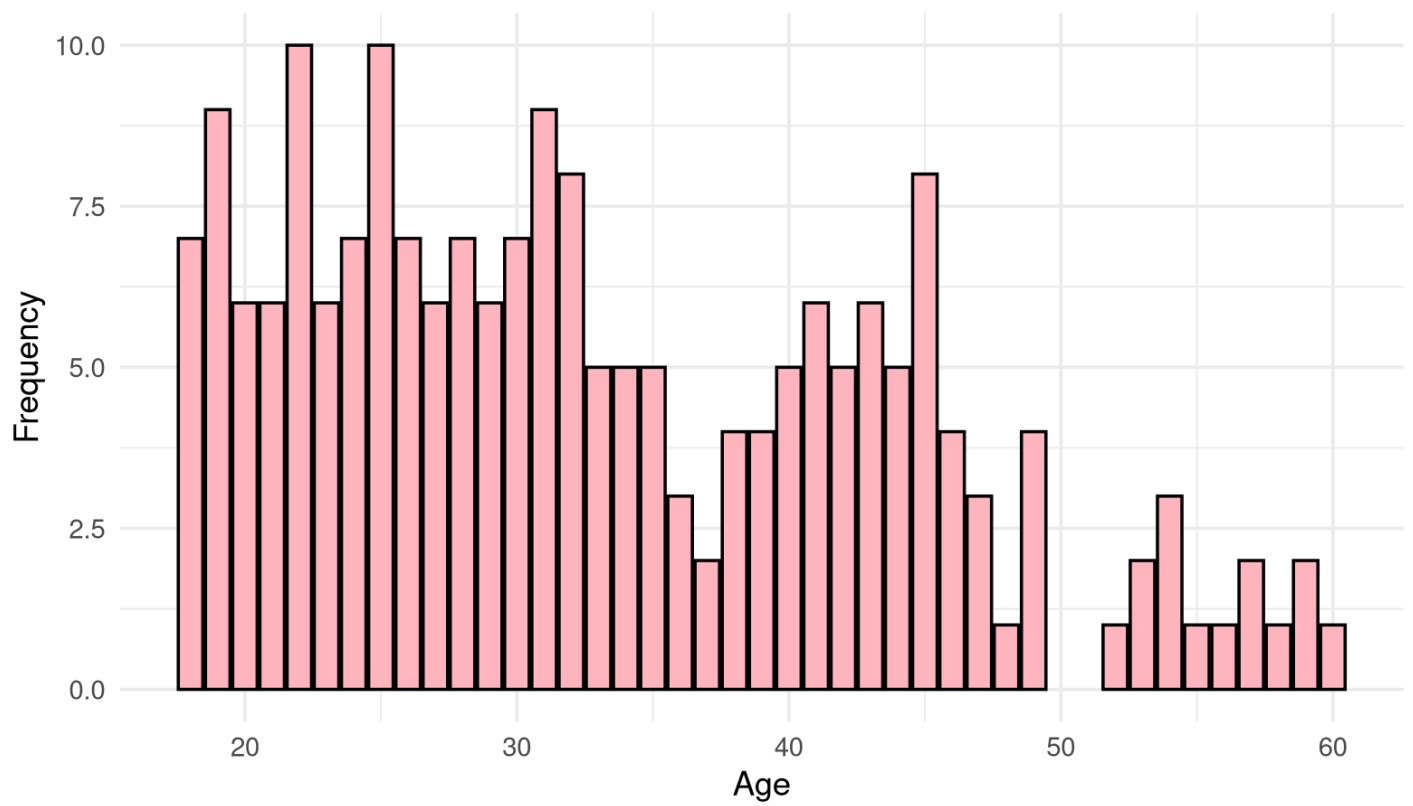
# Histogram-like Plot for Gender



Hide

```
data %>%
  ggplot(aes(x = p_age)) +
  geom_bar(stat = "count", fill = "lightpink", color = "black") +
  labs(title = "Histogram-like Plot for Age", x = "Age", y = "Frequency") +
  theme_minimal()
```

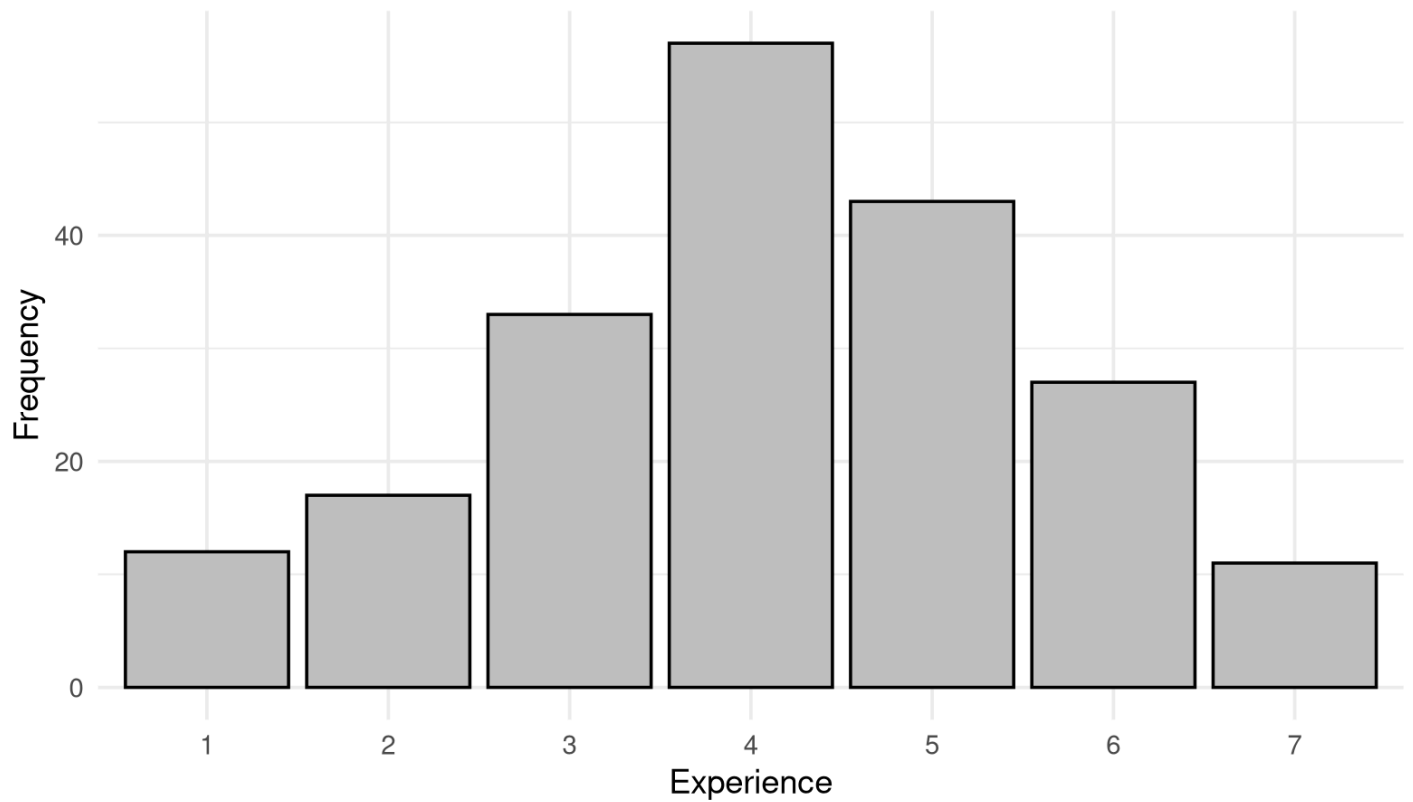
Histogram-like Plot for Age



Hide

```
data %>%  
  ggplot(aes(x = p_experience)) +  
  geom_bar(stat = "count", fill = "grey", color = "black") +  
  labs(title = "Histogram-like Plot for Experience", x = "Experience", y = "Frequency") +  
  theme_minimal()
```

Histogram-like Plot for Experience



Hide

```
# count_not_male <- data %>%  
#   filter(p_gender != "M") %>% # Exclude males  
#   summarize(count = n())  
#  
# count_male <- data %>%  
#   filter(p_gender == "M") %>% # Include only males  
#   summarize(count = n())  
#  
# count_not_male  
# count_male
```

## DEPV Distribution Visualized

Hide

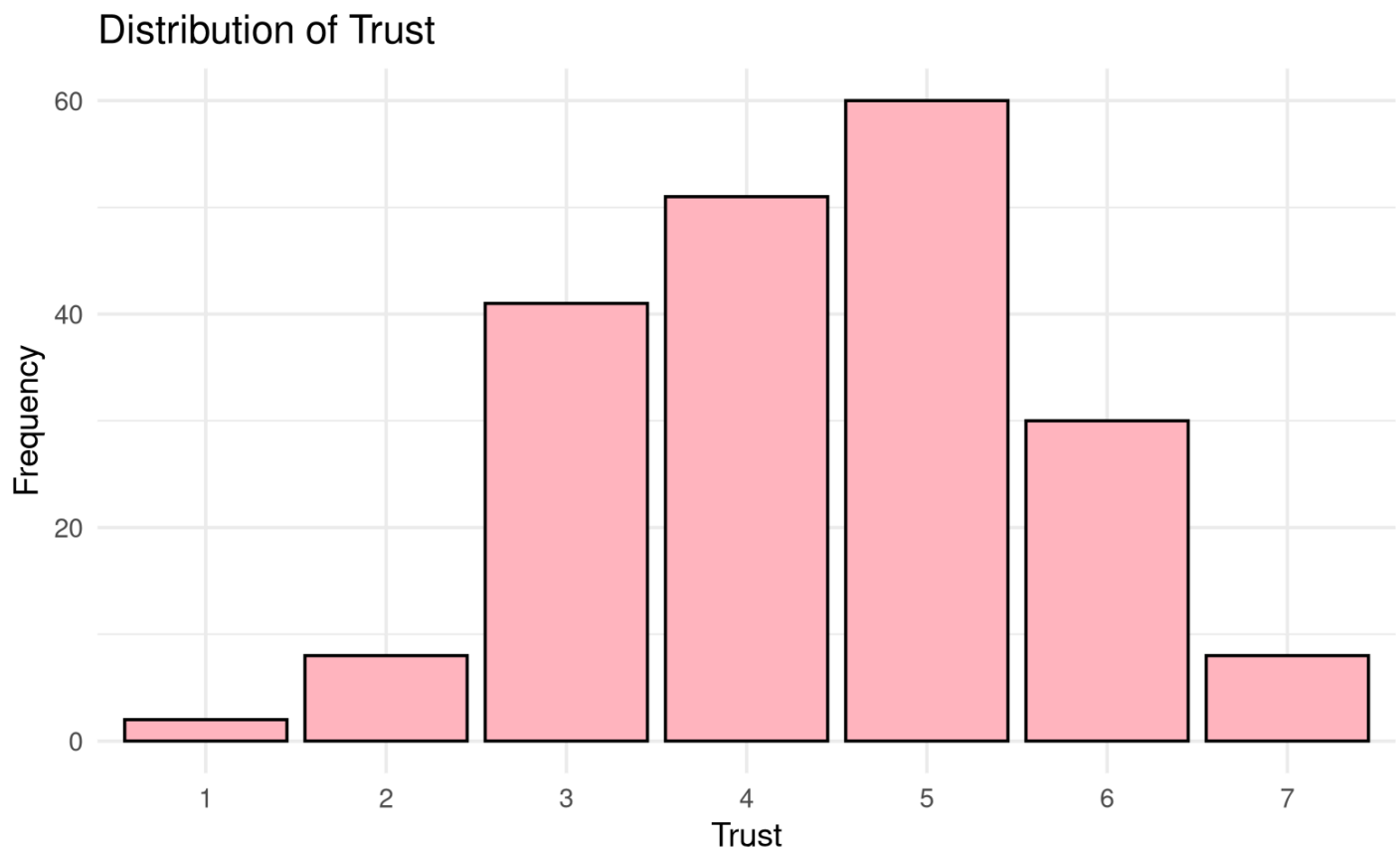
```

category_colors <- c("lightblue", "lightgreen", "lightpink", "orange", "purple")
continuous_color <- "steelblue"

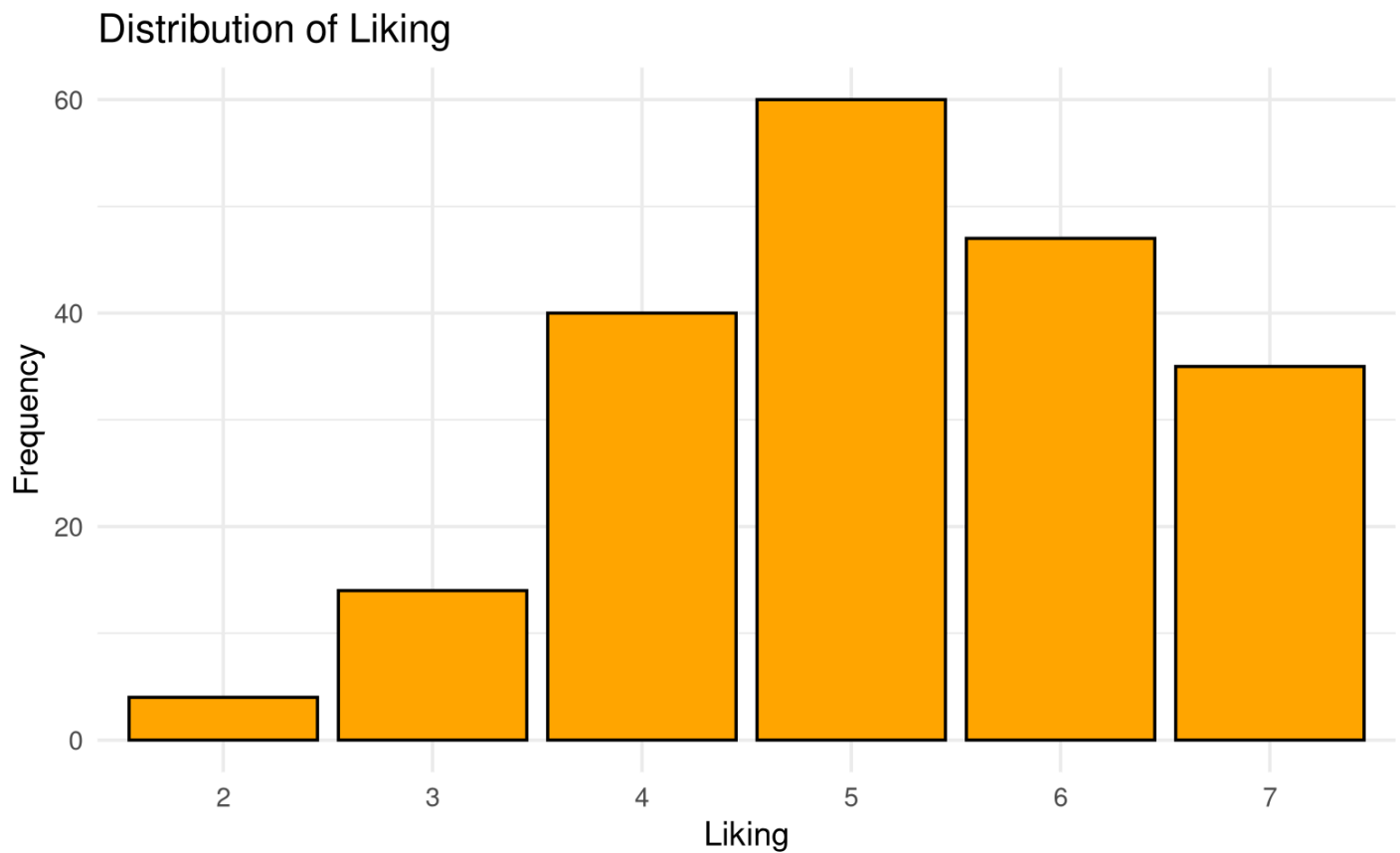
# # Plot for r_model
# data %>%
#   ggplot(aes(x = r_model)) +
#   geom_bar(stat = "count", fill = category_colors[1], color = "black") +
#   labs(title = "Distribution of Robot Model", x = "Robot Model", y = "Frequency") +
#   theme_minimal()
#
# # Plot for r_errors
# data %>%
#   ggplot(aes(x = r_errors)) +
#   geom_bar(stat = "count", fill = category_colors[2], color = "black") +
#   labs(title = "Distribution of Robot Errors", x = "Robot Errors", y = "Frequency") +
#   theme_minimal()

# Plot for trust
data %>%
  ggplot(aes(x = trust)) +
  geom_bar(fill = category_colors[3], color = "black") +
  labs(title = "Distribution of Trust", x = "Trust", y = "Frequency") +
  theme_minimal()

```



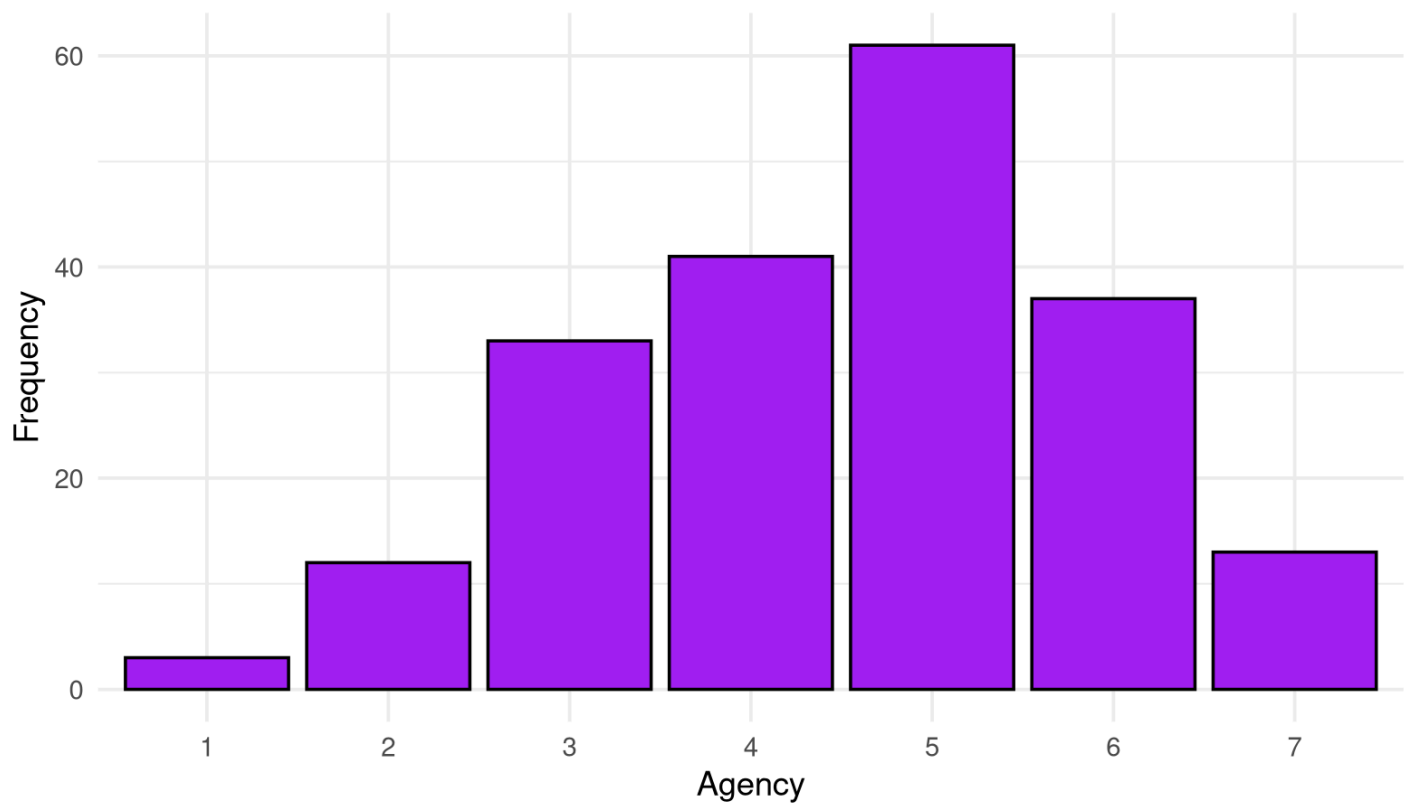
```
# Plot for liking
data %>%
  ggplot(aes(x = liking)) +
  geom_bar(fill = category_colors[4], color = "black") +
  labs(title = "Distribution of Liking", x = "Liking", y = "Frequency") +
  theme_minimal()
```



Hide

```
# Plot for agency
data %>%
  ggplot(aes(x = agency)) +
  geom_bar(fill = category_colors[5], color = "black") +
  labs(title = "Distribution of Agency", x = "Agency", y = "Frequency") +
  theme_minimal()
```

Distribution of Agency

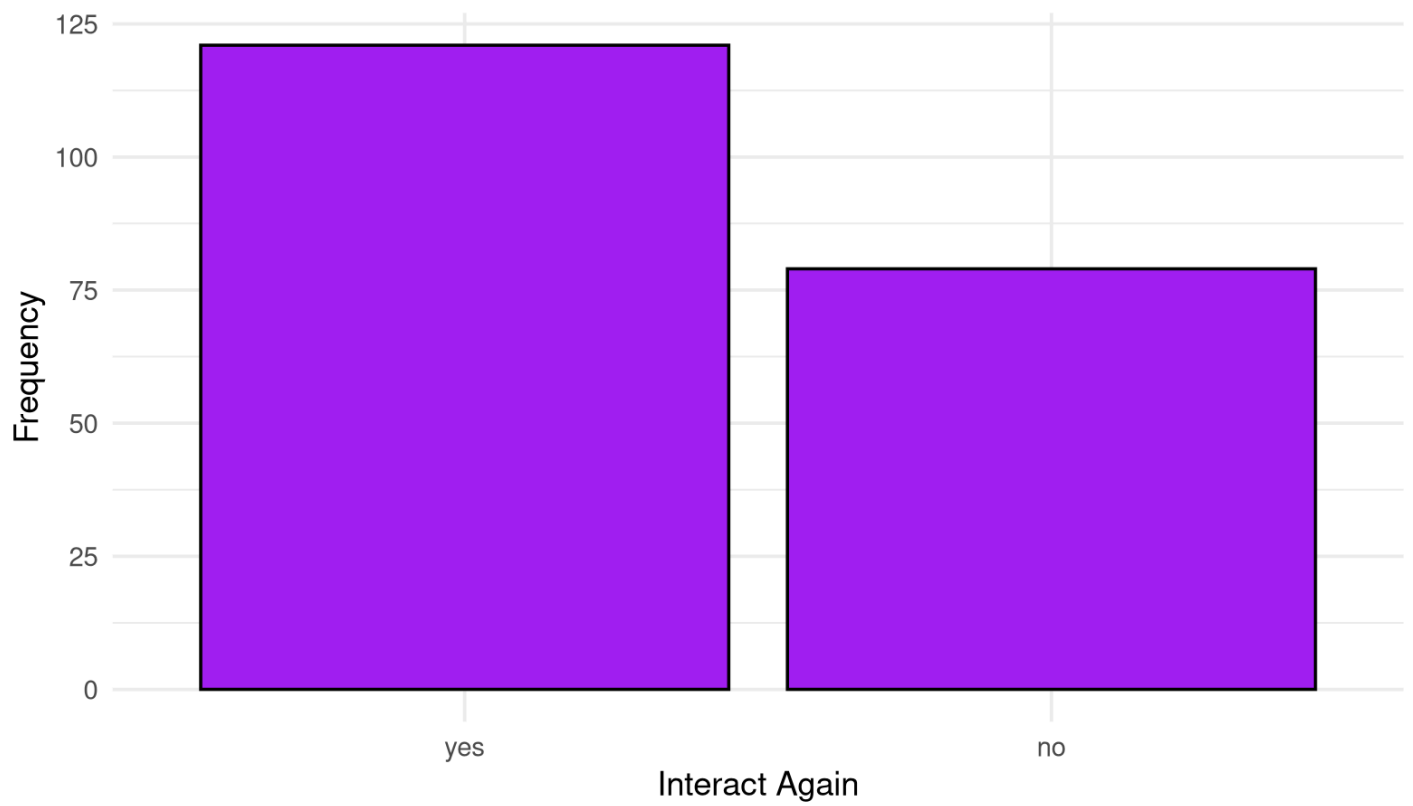


Hide

```
# Plot for interact_again
data %>%
  ggplot(aes(x = interact_again)) +
  geom_bar(fill = category_colors[5], color = "black") +
  labs(title = "Distribution of Interact Again", x = "Interact Again", y = "Frequency") +
  theme_minimal()
```



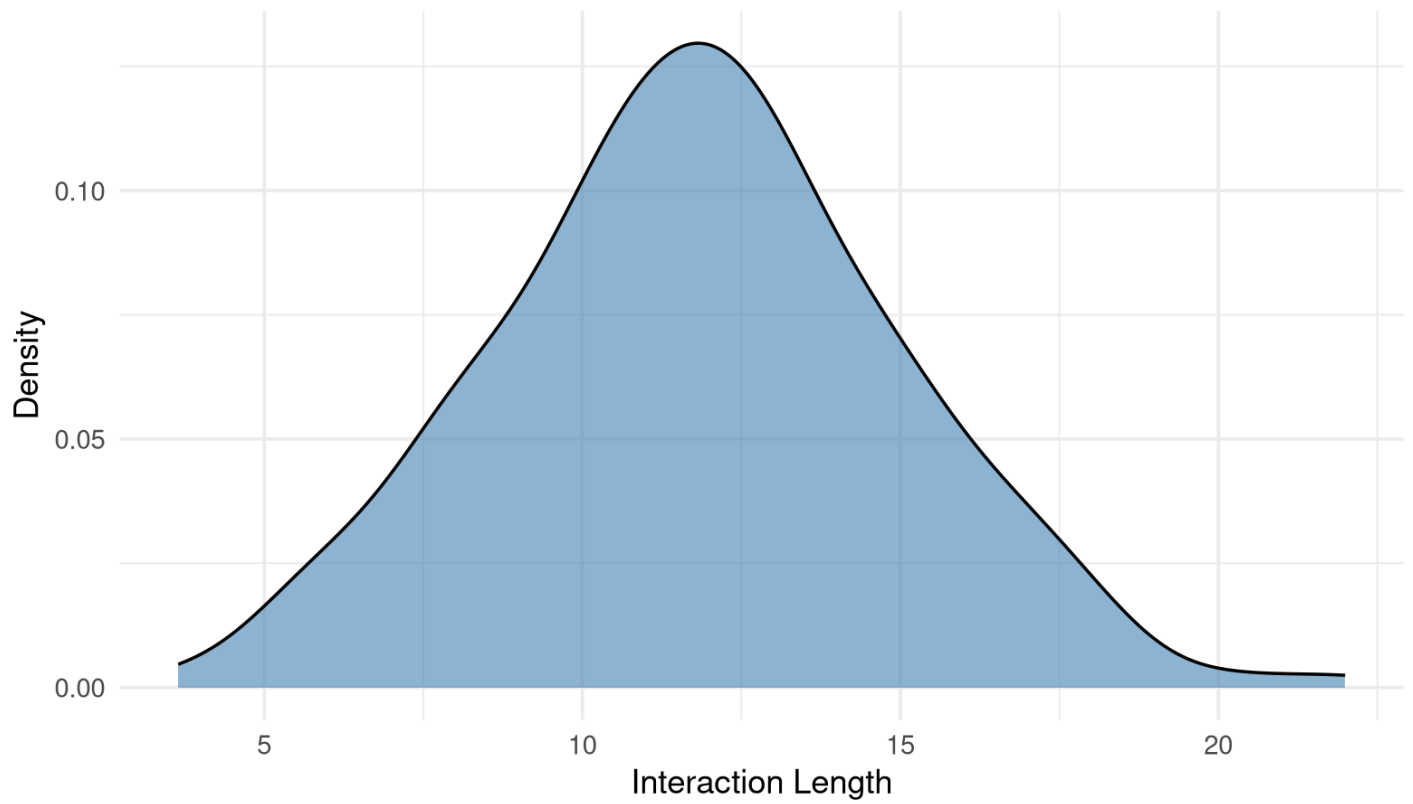
Distribution of Interact Again



Hide

```
# Density plot for interaction_length
data %>%
  ggplot(aes(x = interaction_length)) +
  geom_density(fill = continuous_color, color = "black", alpha = 0.6) +
  labs(title = "Density Plot of Interaction Length", x = "Interaction Length", y = "Density") +
  theme_minimal()
```

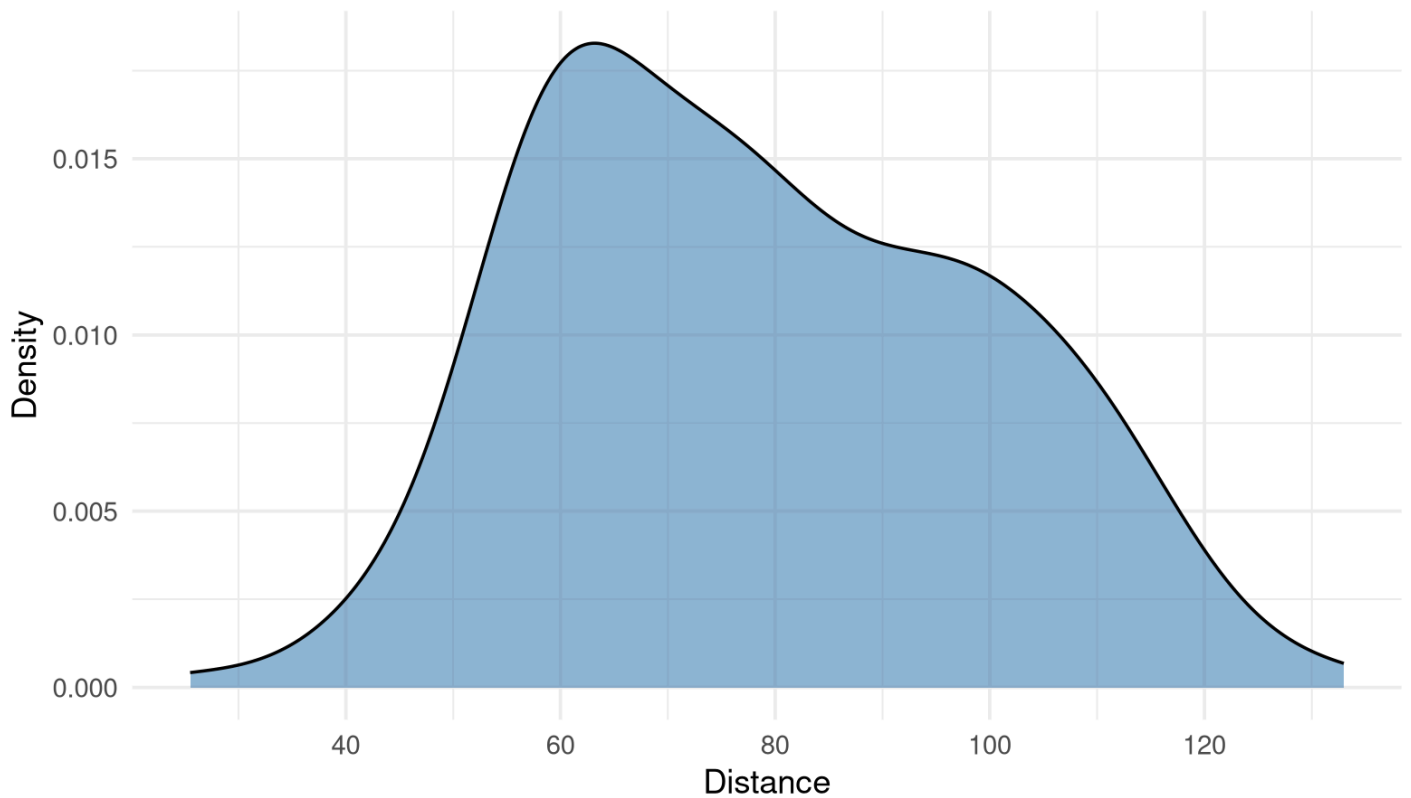
Density Plot of Interaction Length



Hide

```
# Density plot for distance
data %>%
  ggplot(aes(x = distance)) +
  geom_density(fill = continuous_color, color = "black", alpha = 0.6) +
  labs(title = "Density Plot of Distance", x = "Distance", y = "Density") +
  theme_minimal()
```

Density Plot of Distance



Hide

NA  
NA  
NA

From this it seems that the INDPV have somewhat negatively skewed dist.

The participant's distance to the robot is resembling almost a mixture of distributions (but we will check this later.)

Wierdly the interaction length is approximately unimodal. Maybe this is because of the task itself idk. At least I thought maybe people experienced with robots would interact more. But let's find out?

I wonder what could effect the distance? Maybe age, gender, experience .

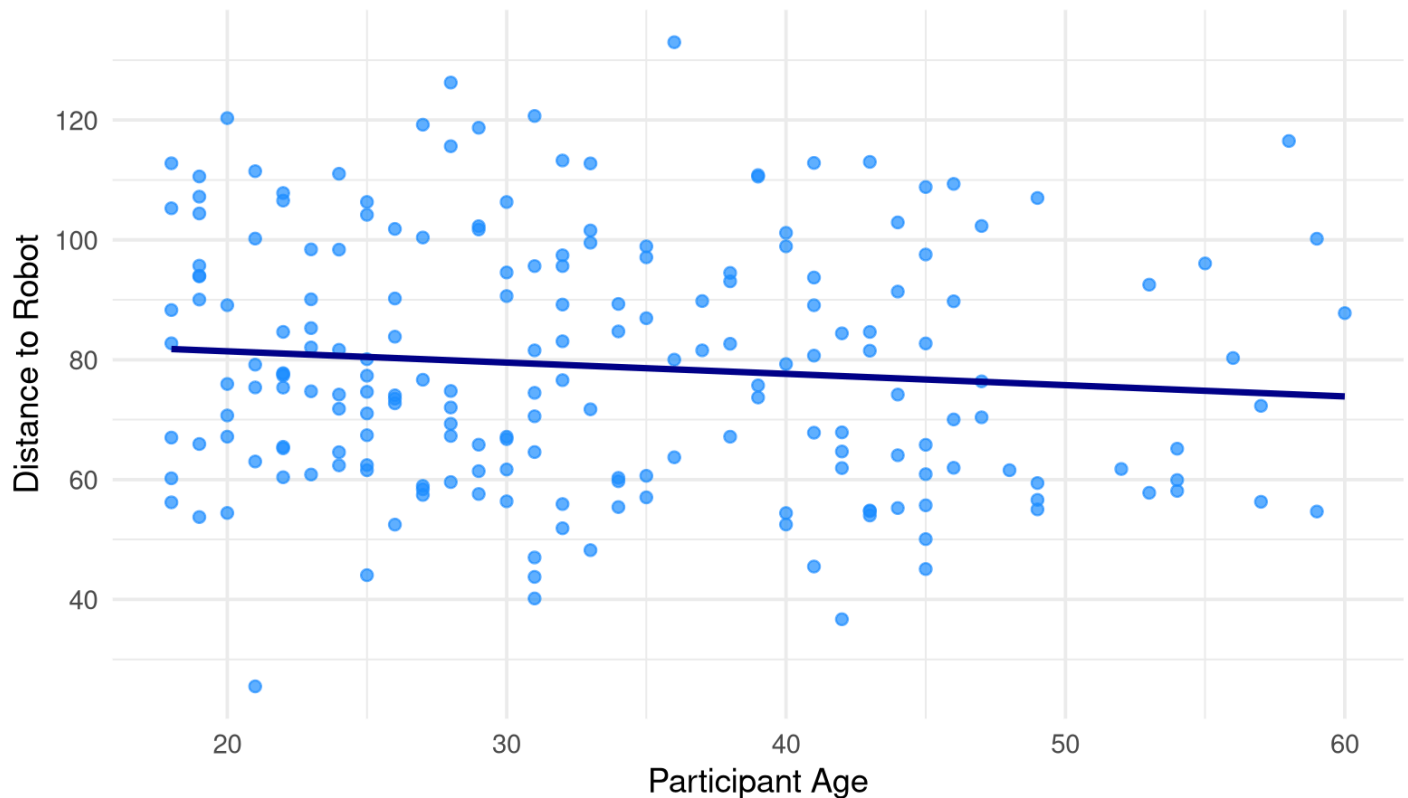
## Robot distance interactions with INDPVs

Note: This EDA isn't used to come up with our Hypothesis as this would imply sort of cheating. This is only done for sanity purposes.

Hide

```
data %>%
  ggplot(aes(x = p_age, y = distance)) +
  geom_point(color = "dodgerblue", alpha = 0.7) +
  geom_smooth(method = "lm", color = "darkblue", se = FALSE) +
  labs(title = "Scatter Plot of Age vs. Distance",
       x = "Participant Age",
       y = "Distance to Robot") +
  theme_minimal()
```

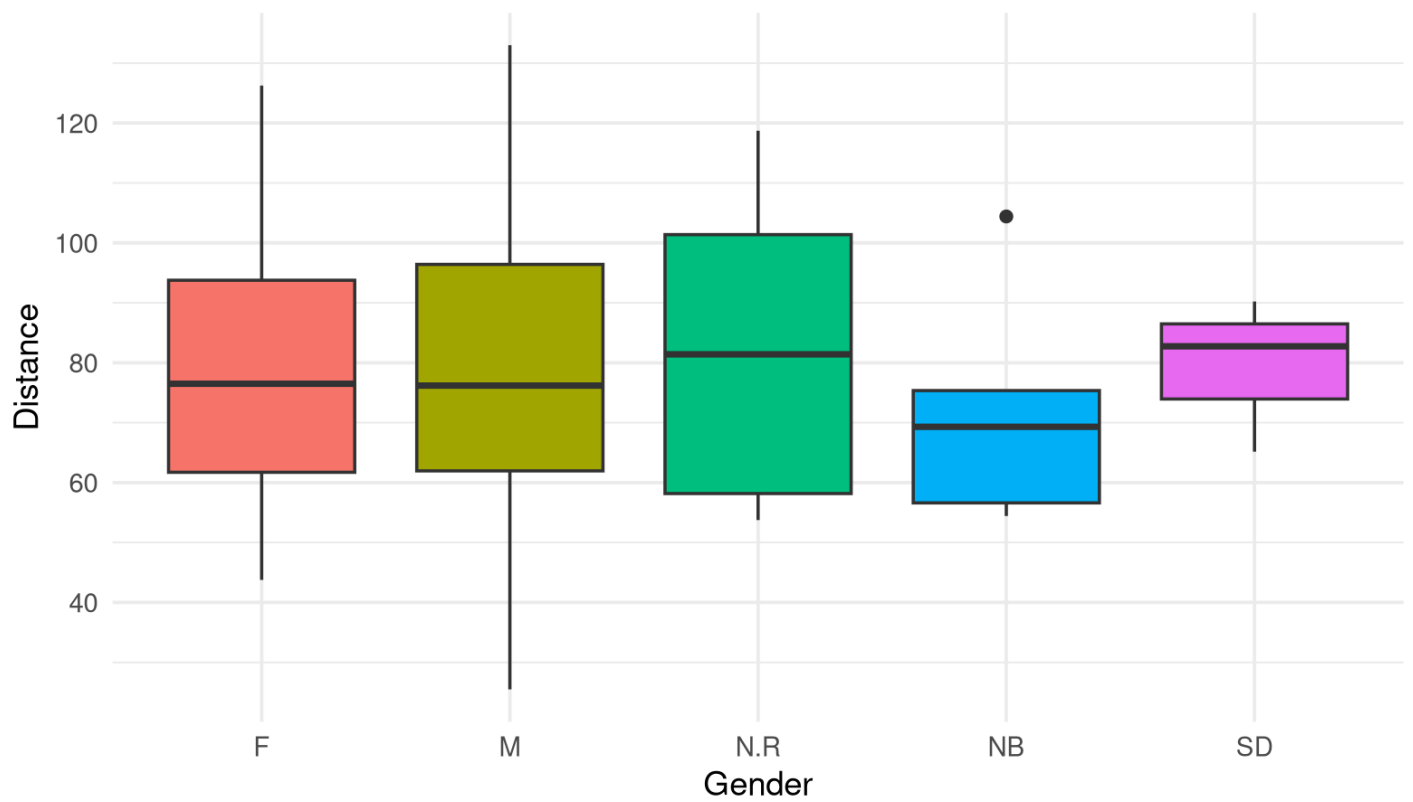
Scatter Plot of Age vs. Distance



Hide

```
data %>%
  ggplot(aes(x = p_gender, y = distance, fill = p_gender)) +
  geom_boxplot() +
  labs(title = "Distance to Robot by Gender",
       x = "Gender",
       y = "Distance") +
  theme_minimal() +
  theme(legend.position = "none")
```

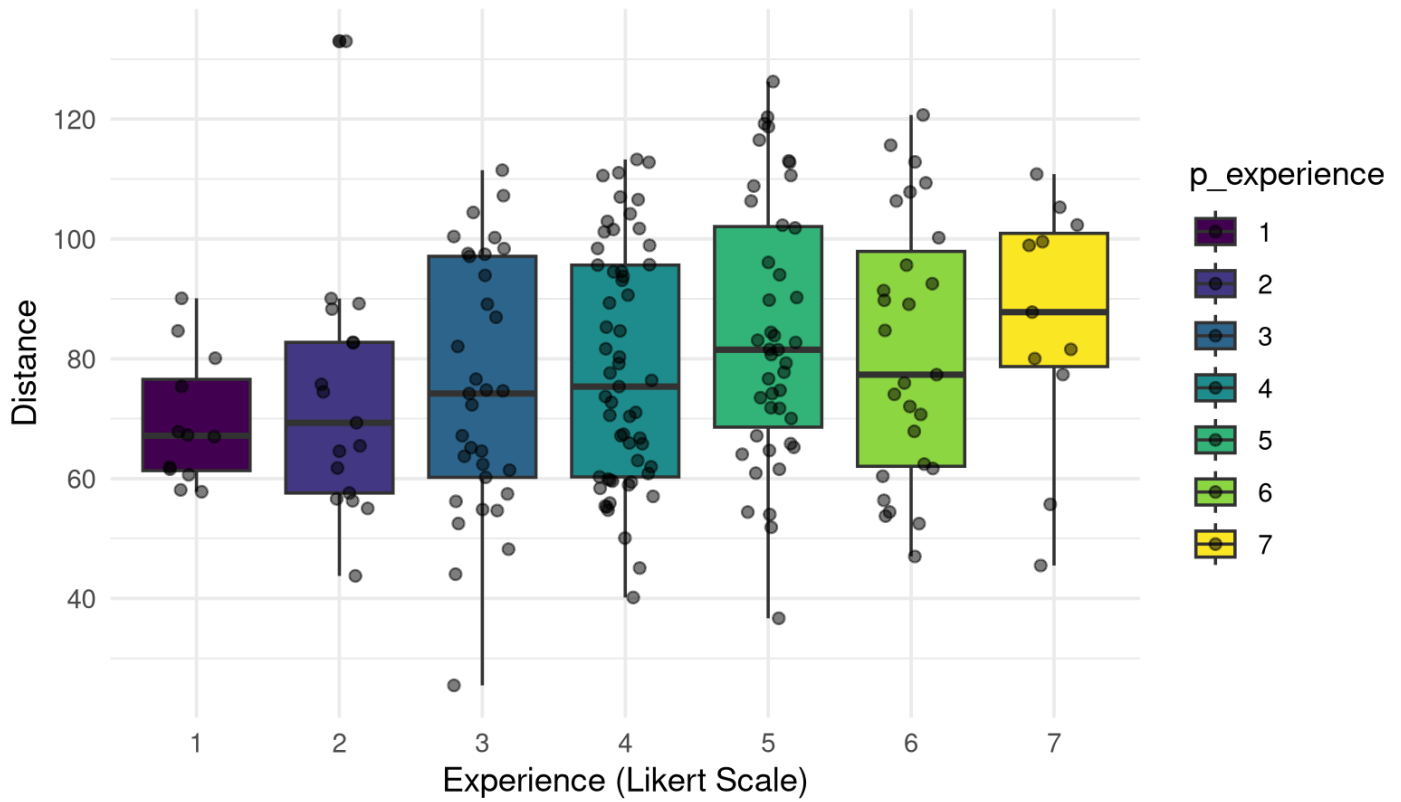
Distance to Robot by Gender



Hide

```
data %>%
  ggplot(aes(x = p_experience, y = distance, fill = p_experience)) +
  geom_boxplot() +
  geom_jitter(width = 0.2, alpha = 0.5, color = "black") + # Jittered points
  labs(title = "Box Plot of Experience vs. Distance",
        x = "Experience (Likert Scale)",
        y = "Distance") +
  theme_minimal()
```

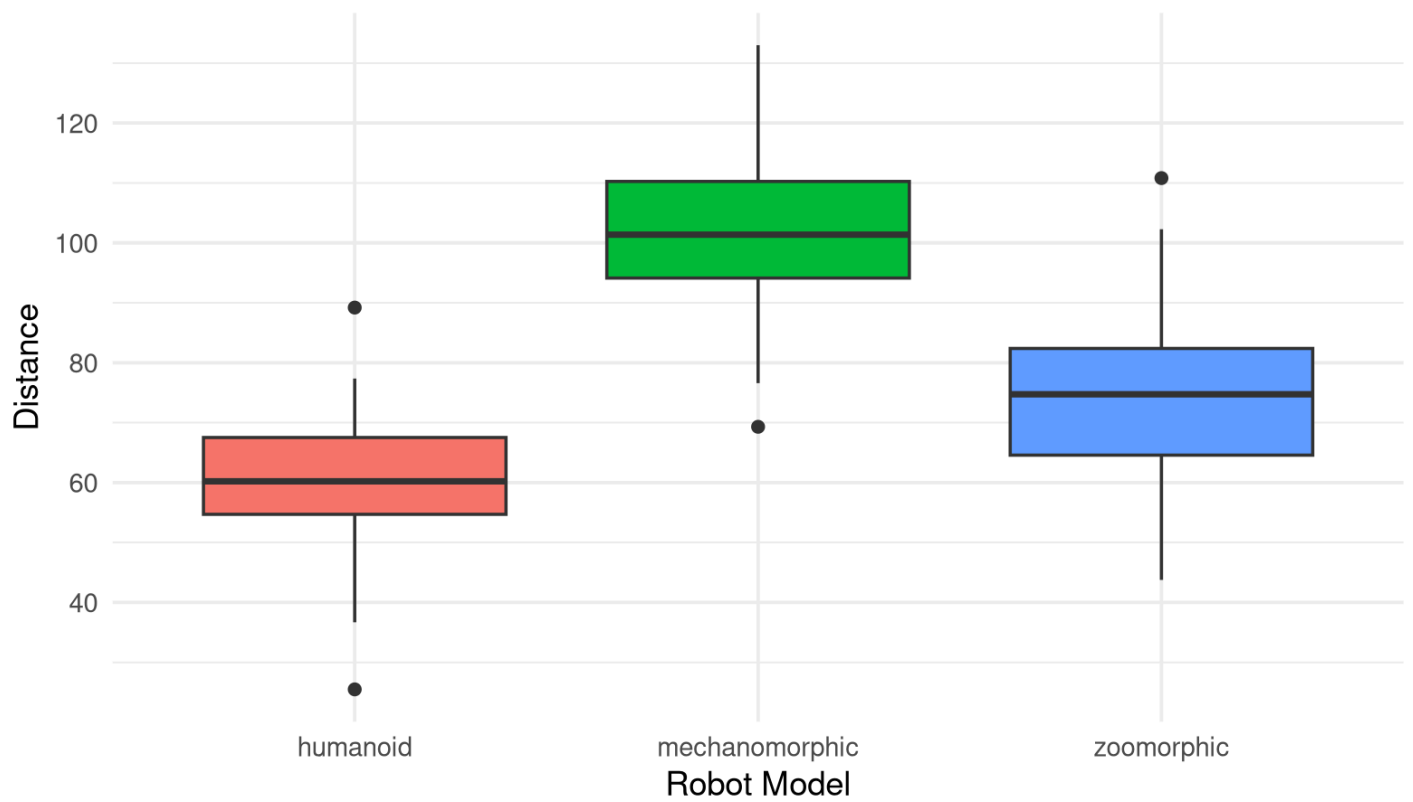
Box Plot of Experience vs. Distance



Hide

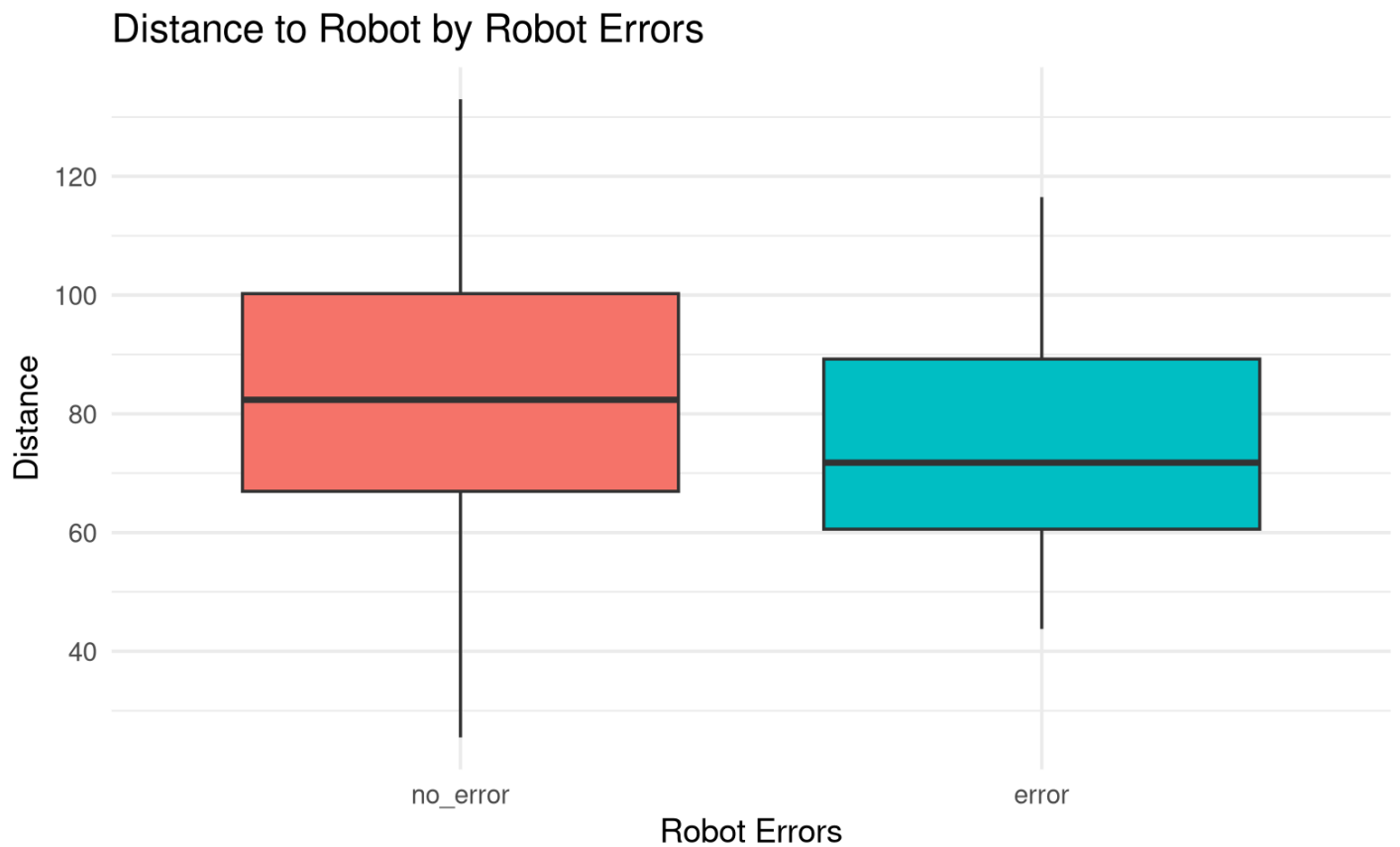
```
data %>%
  ggplot(aes(x = r_model, y = distance, fill = r_model)) +
  geom_boxplot() +
  labs(title = "Distance to Robot by Robot Model",
       x = "Robot Model",
       y = "Distance") +
  theme_minimal() +
  theme(legend.position = "none")
```

## Distance to Robot by Robot Model



Hide

```
data %>%  
  ggplot(aes(x = r_errors, y = distance, fill = r_errors)) +  
  geom_boxplot() +  
  labs(title = "Distance to Robot by Robot Errors",  
        x = "Robot Errors",  
        y = "Distance") +  
  theme_minimal() +  
  theme(legend.position = "none")
```



## RQ1: Is there a relationship between the type of model and distance?

$H_0$ : Participants will maintain a greater distance from mechanomorphic robots compared to humanoid and zoomorphic robots

Hide

```
summary_stats <- data %>%
  group_by(r_model) %>%
  summarise(
    mean_distance = mean(distance, na.rm = TRUE),
    sd_distance = sd(distance, na.rm = TRUE)
  )
summary_stats
```

r_model <fctr>	mean_distance <dbl>	sd_distance <dbl>
humanoid	60.66308	10.90826
mechanomorphic	101.47798	12.12504
zoomorphic	74.19891	13.12046
3 rows		

We use Kruskal's test since distance is not-normal as seen from the dist.



[Hide](#)

```
kruskal_result <- kruskal.test(distance ~ r_model, data = data)
kruskal_result
```

Kruskal-Wallis rank sum test

```
data: distance by r_model
Kruskal-Wallis chi-squared = 131.66, df = 2, p-value < 2.2e-16
```

Based on the plots and test we can accept our  $H_0$ .

A Kruskal-Wallis test indicated a statistically significant difference in distance based on the type of robot models,  $\chi^2(2) = 131.66, p < .001$ .

Let's try a post-hoc pairwise comparison's test to see which model effects distance.

[Hide](#)

```
library(FSA)
```

```
Warning: package 'FSA' was built under R version 4.4.1Registered S3 methods overwritten by 'FSA':
A':
```

```
  method      from
  confint.boot car
  hist.boot    car
## FSA v0.9.5. See citation('FSA') if used in publication.
## Run fishR() for related website and fishR('IFAR') for related book.
```

```
Attaching package: 'FSA'
```

```
The following object is masked from 'package:car':
```

```
bootCase
```

```
The following object is masked from 'package:psych':
```

```
headtail
```

[Hide](#)

```
dunn_test_results <- dunnTest(distance ~ r_model, data = data, method = "bonferroni")
dunn_test_results
```

Dunn (1964) Kruskal-Wallis multiple comparison  
p-values adjusted with the Bonferroni method.

Comparison <chr>	Z <dbl>	P.unadj <dbl>	P.adj <dbl>
humanoid - mechanomorphic	-11.314822	1.108264e-29	3.324793e-29
humanoid - zoomorphic	-4.263988	2.008102e-05	6.024306e-05
mechanomorphic - zoomorphic	7.339495	2.144015e-13	6.432046e-13
3 rows			

Post-hoc Dunn's tests indicated significant differences between the robot models on distance: humanoid vs. mechanomorphic ( $Z = -11.31$ ,  $p < .001$ ), humanoid vs. zoomorphic ( $Z = -4.26$ ,  $p < .001$ ), and mechanomorphic vs. zoomorphic ( $Z = 7.34$ ,  $p < .001$ ). These results suggest that all pairs of robot models differ significantly regarding participant distance.

#### Conclusion:

Mechanamorphic robots the *highest mean distance* ( $M = 101.48$  ), indicating that participants kept the furthest away from this model on average.

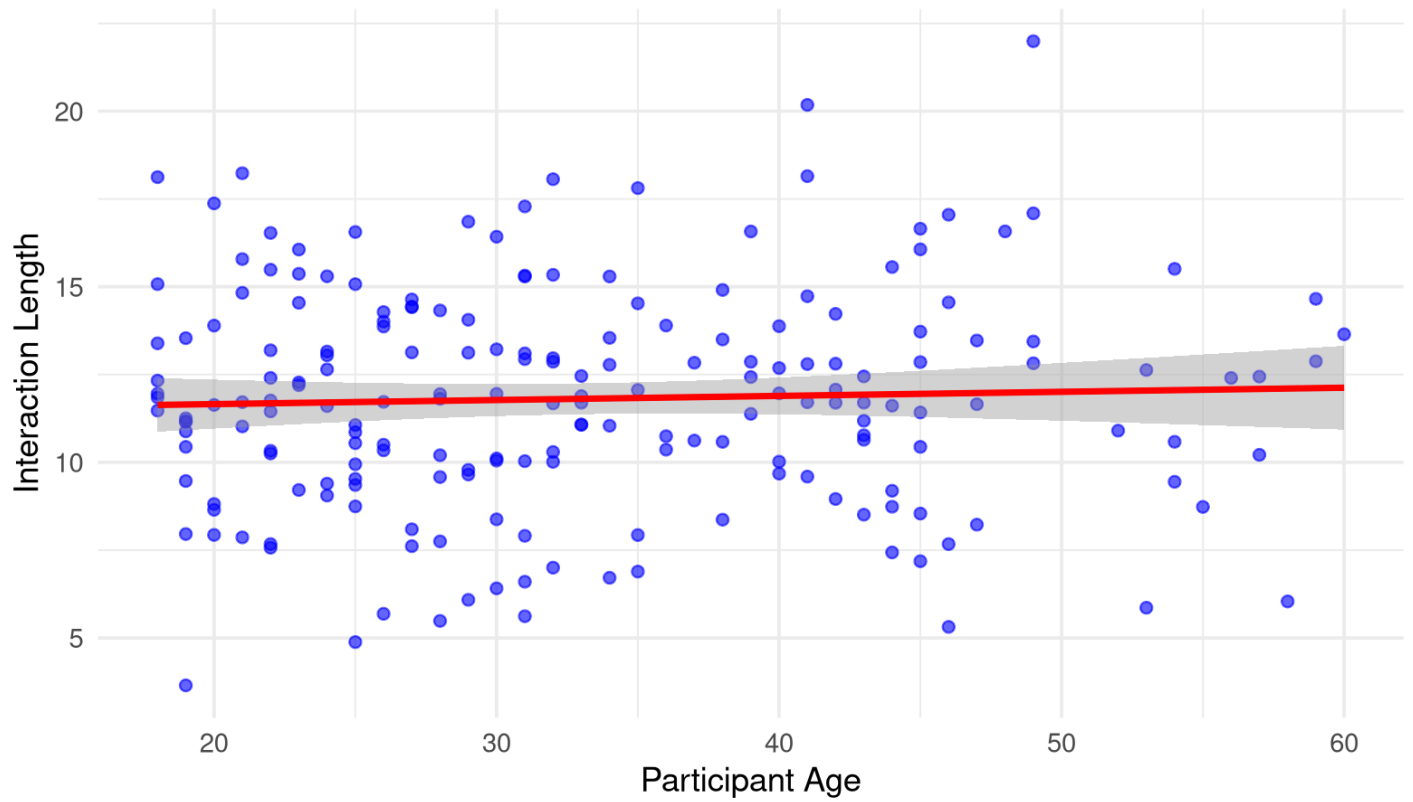
Humanoid's had the *lowest mean distance* ( $M = 74.20$  ), indicating that participants were most comfortable around this model on average.

## Interaction Length interaction with INDPV

[Hide](#)

```
data %>%
  ggplot(aes(x = p_age, y = interaction_length)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_smooth(method = "lm", se = TRUE, color = "red") +
  labs(title = "Scatter Plot of Age vs. Interaction Length",
       x = "Participant Age",
       y = "Interaction Length") +
  theme_minimal()
```

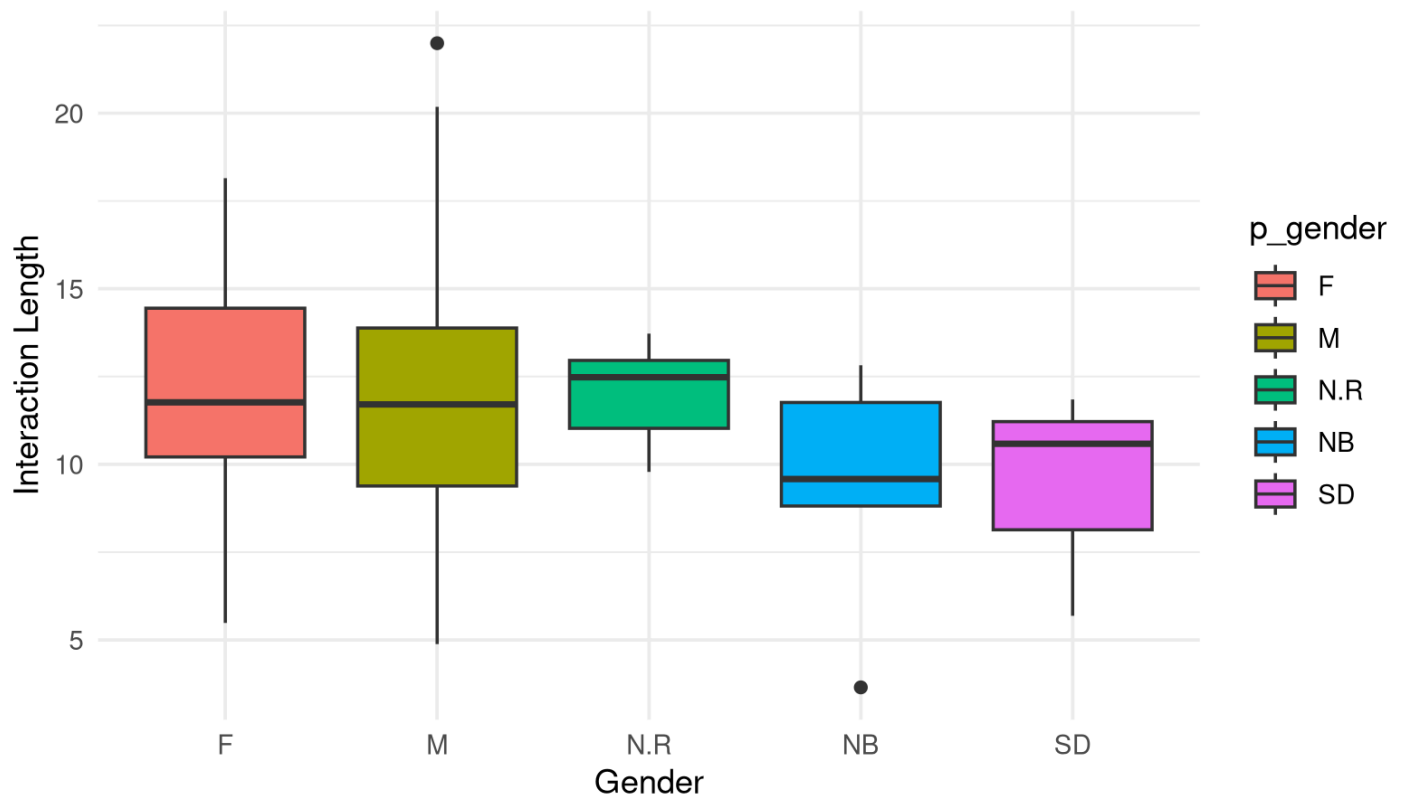
Scatter Plot of Age vs. Interaction Length



Hide

```
data %>%  
  ggplot(aes(x = p_gender, y = interaction_length, fill = p_gender)) +  
  geom_boxplot() +  
  labs(title = "Box Plot of Interaction Length by Gender",  
        x = "Gender",  
        y = "Interaction Length") +  
  theme_minimal()
```

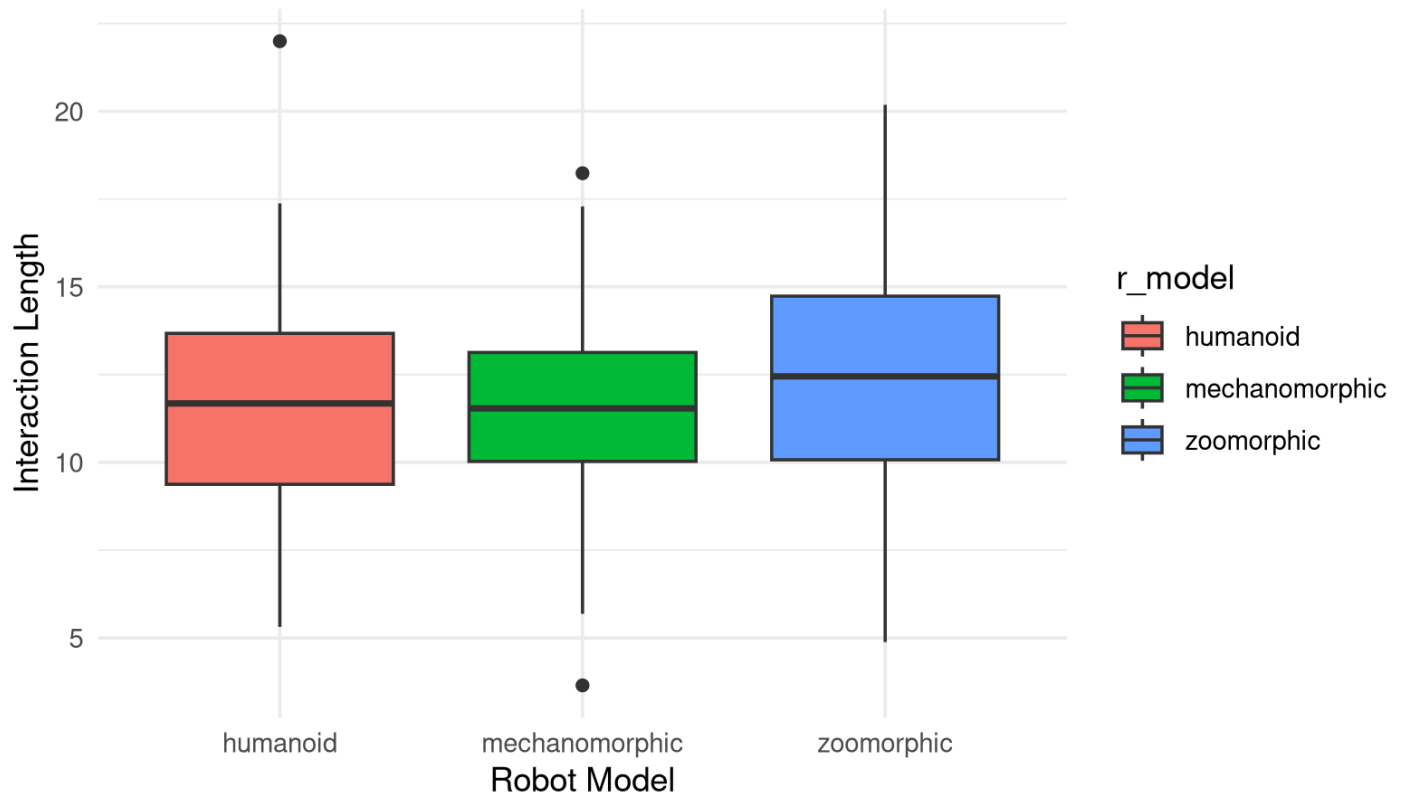
Box Plot of Interaction Length by Gender



Hide

```
data %>%
  ggplot(aes(x = r_model, y = interaction_length, fill = r_model)) +
  geom_boxplot() +
  labs(title = "Box Plot of Interaction Length by Robot Model",
        x = "Robot Model",
        y = "Interaction Length") +
  theme_minimal()
```

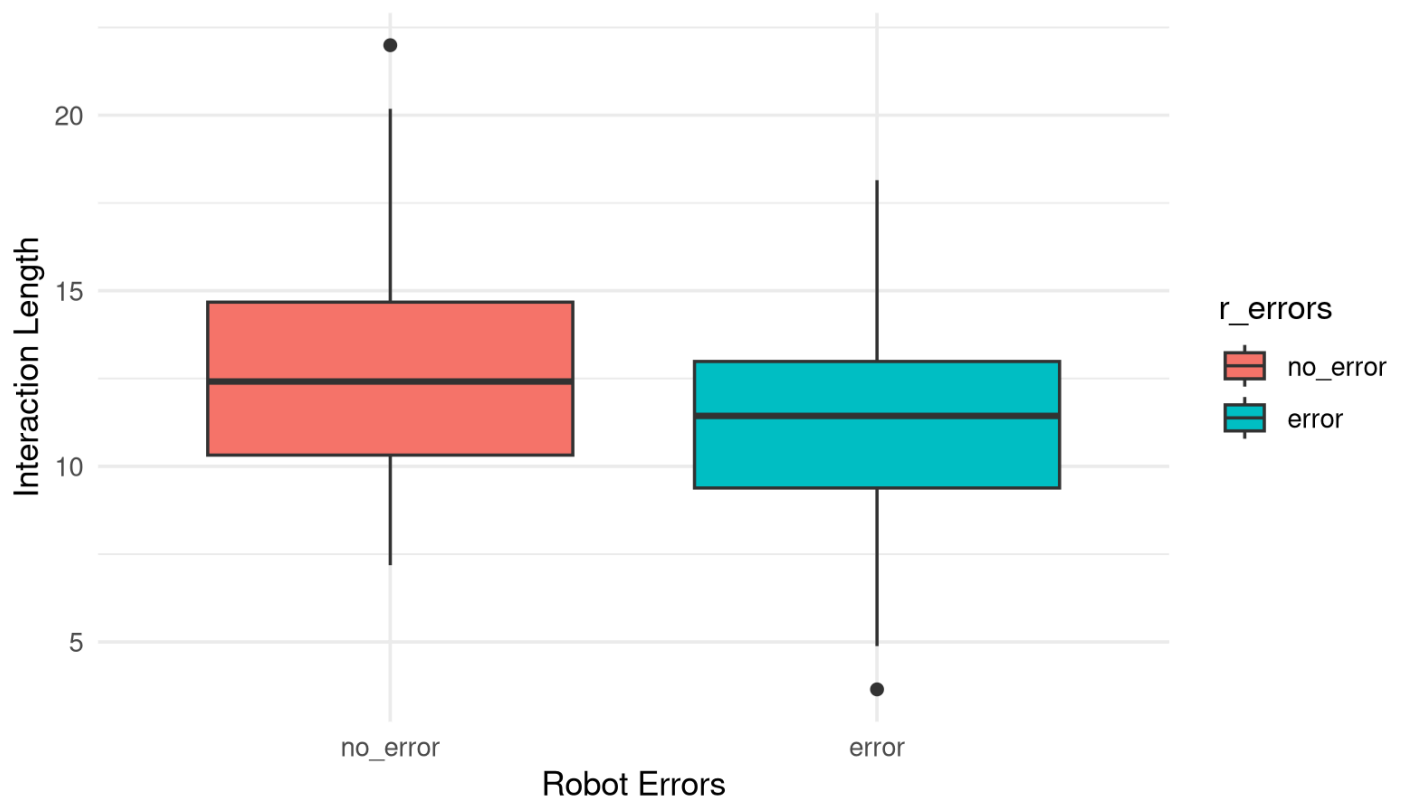
Box Plot of Interaction Length by Robot Model



Hide

```
data %>%
  ggplot(aes(x = r_errors, y = interaction_length, fill = r_errors)) +
  geom_boxplot() +
  labs(title = "Box Plot of Interaction Length by Robot Errors",
        x = "Robot Errors",
        y = "Interaction Length") +
  theme_minimal()
```

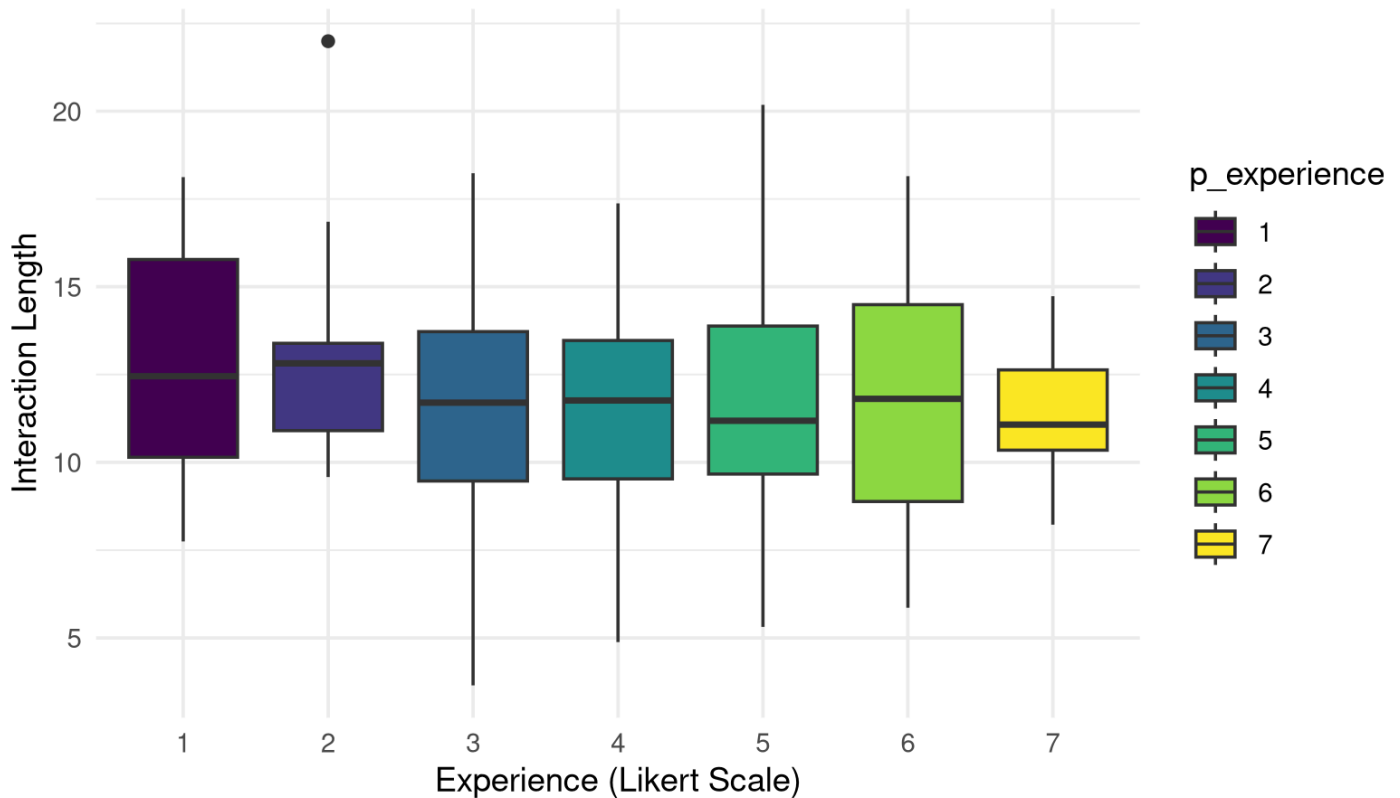
Box Plot of Interaction Length by Robot Errors



Hide

```
data %>%  
  ggplot(aes(x = p_experience, y = interaction_length, fill = p_experience)) +  
  geom_boxplot() +  
  labs(title = "Box Plot of Interaction Length by Experience",  
        x = "Experience (Likert Scale)",  
        y = "Interaction Length") +  
  theme_minimal()
```

Box Plot of Interaction Length by Experience



## RQ2: Does robot model type effect the interaction length?

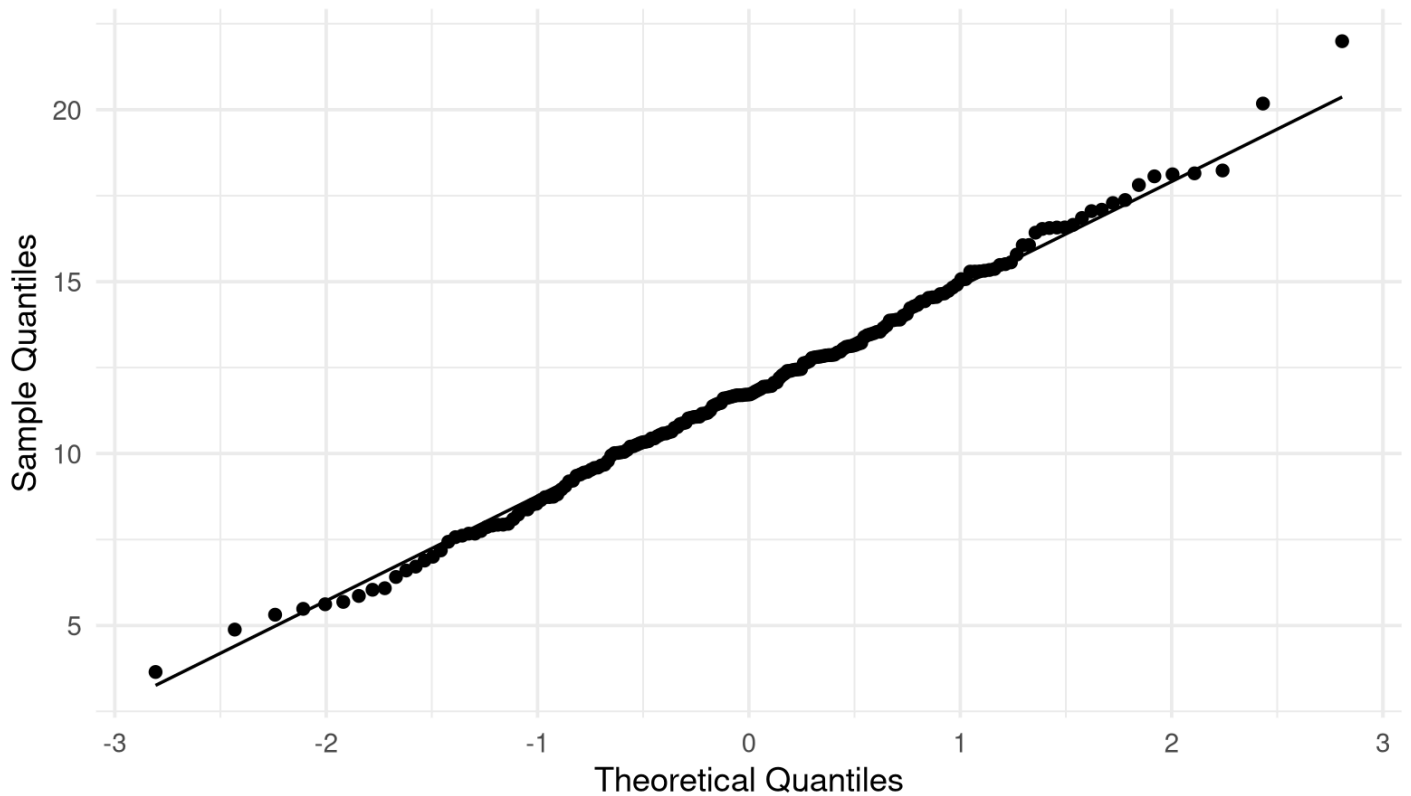
$H_0$ : Interaction length will be longer with humanoid robots than with mechanomorphic and zoomorphic robots.

Check for normality assumption of interaction\_length using QQ-Plot. The QQ-Plot of interaction length does not deviate much so it seems to assume normality.

Hide

```
ggplot(data, aes(sample = interaction_length)) +  
  stat_qq() +  
  stat_qq_line() +  
  labs(title = "QQ Plot of Interaction Length", x = "Theoretical Quantiles", y="Sample Quantile  
s") +  
  theme_minimal()
```

## QQ Plot of Interaction Length



Doing a Bartlett's test to see if variances are equal. (Homogeneity?)

Hide

```
bartlett.test(interaction_length ~ r_model, data = data)
```

Bartlett test of homogeneity of variances

data: interaction\_length by r\_model  
Bartlett's K-squared = 1.225, df = 2, p-value = 0.542

Since p-value > .05 . Variances across groups are equal so we proceed with ANOVA.

Hide

```
anova_result <- aov(interaction_length ~ r_model, data = data)  
summary(anova_result)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
r_model	2	35.1	17.572	1.782	0.171
Residuals	197	1943.0	9.863		

An ANOVA was conducted to examine the effect of robot model type on interaction length. The results indicated that there was no significant effect of robot model type on interaction length,  $F(2, 197) = 1.782, p = 0.171$  . No post-hoc analysis was performed

Conclusion:



The type of robot model doesn't have an effect on the interaction length.

## RQ3: Does gender effect the interaction length?

$H_0$ : Female participants will have a shorter interaction length with robots than male participants.

Hide

```
# Test for normality
data %>%
  group_by(p_gender) %>%
  summarise(p_value = shapiro.test(interaction_length)$p.value)
```

p_gender <fctr>	p_value <dbl>
F	0.2550924
M	0.3607553
N.R	0.4860042
NB	0.4919180
SD	0.3726541
5 rows	

Since  $p > .05$ . We fail to reject  $H_0$  of equal variances. (Homogeneity of Variances)

Hide

```
# Test for Homogeneity
leveneTest(interaction_length ~ p_gender, data = data)
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  4  1.1933  0.315
      195
```

Based on the same let's perform an ANOVA test.

A one-way ANOVA was conducted to examine the effect of gender on interaction length. There was no significant effect of gender on interaction length,  $F(4, 195) = 1.32, p = .265$ . Thus  $H_0$  was rejected.

Hide

```
summary(anova_result)
```

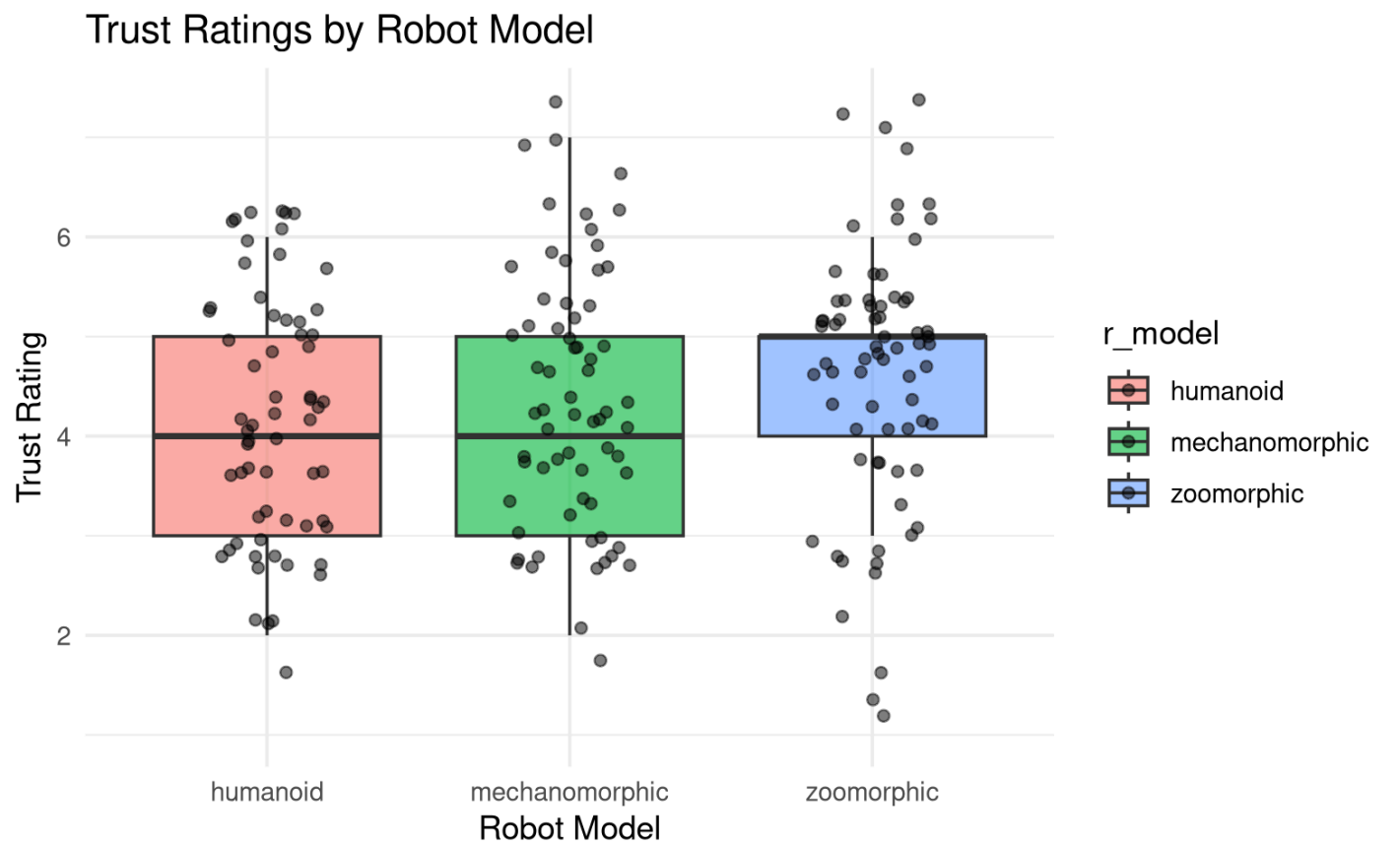
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
p_gender	4	52	13.006	1.317	0.265
Residuals	195	1926	9.877		

Conclusion:

Gender doesn't not have an effect on our interaction length.

RQ4: Does robot-model effect Trustworthiness?

H<sub>0</sub>: Humanoid robots will be perceived as more trustworthy than zoomorphic and mechanomorphic robots.



Hide

```
trust_summary <- data %>%
  group_by(r_model) %>%
  summarize(
    mean_trust = mean(as.numeric(trust), na.rm = TRUE),
    sd_trust = sd(as.numeric(trust), na.rm = TRUE)
  )
trust_summary
```

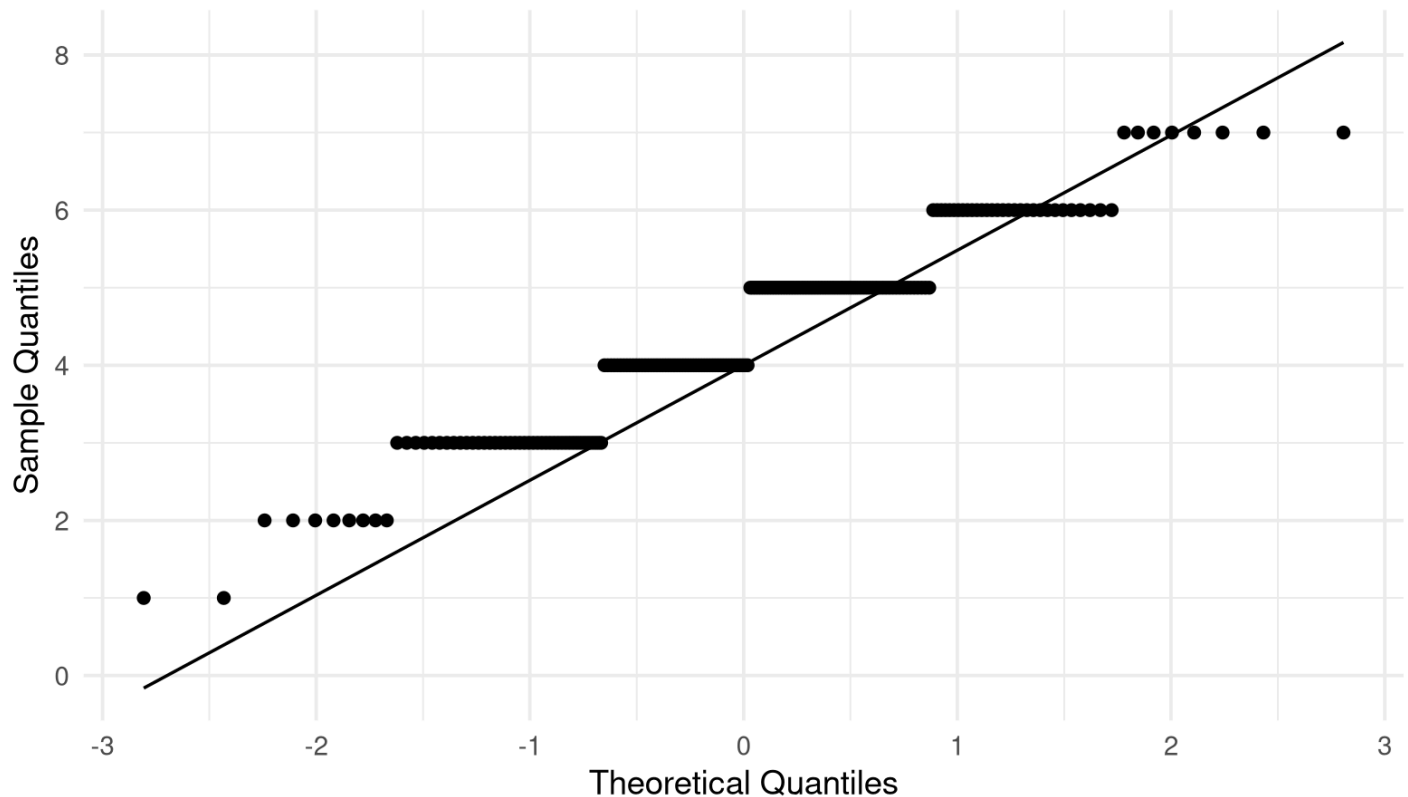
r_model <fctr>	mean_trust <dbl>	sd_trust <dbl>
humanoid	4.174603	1.185119
mechanomorphic	4.409091	1.264635
zoomorphic	4.605634	1.270308

3 rows

Hide

```
ggplot(data, aes(sample = as.numeric(trust))) +  
  stat_qq() +  
  stat_qq_line() +  
  labs(title = "QQ Plot of Trust", x = "Theoretical Quantiles", y = "Sample Quantiles") +  
  theme_minimal()
```

QQ Plot of Trust



If I look at the QQ-Plot we can see Ordinal data doesn't follow our normality assumptions (remember it's Likert scale data!). Let's do a statistical test to see whether our hypothesis is true or false.

Use non-parametric Kruskal-Wallis test since data is not normally distributed.

Hide

```
kruskal.test(trust ~ r_model, data = data)
```

Kruskal-Wallis rank sum test

data: trust by r\_model

Kruskal-Wallis chi-squared = 5.0597, df = 2, p-value = 0.07967

A Kruskal-Wallis test was conducted to examine differences in trustworthiness ratings across robot models. The results were not statistically significant

$\chi^2(2) = 5.06$  ,  $p = .08$  . suggesting that trust ratings do not significantly differ between robot models.

Conclusion:

The type of Robot Models does not effect trust ratings of our robot.