Data Analysis Code with Output

Q5. The answers provided by the Speaker/Spot seemed credible and believable overall.

Q6. I believe the speaker/Spot provided accurate answers to all of my questions.

Q7. I felt comfortable and at ease throughout my interaction with the speaker/Spot.

Q8. The Speaker/Spot's responses felt socially appropriate for the situation.

Q9. The speaker/Spot responded in a way that made the interaction feel personal.

Q10. I did NOT enjoy interacting with the speaker/Spot.

Q11. For this question, select the "4" in the following scale

Q12. The Speaker/Spot felt unapproachable during the interaction.

Q13. The speaker/Spot 's communication style did NOT feel natural.

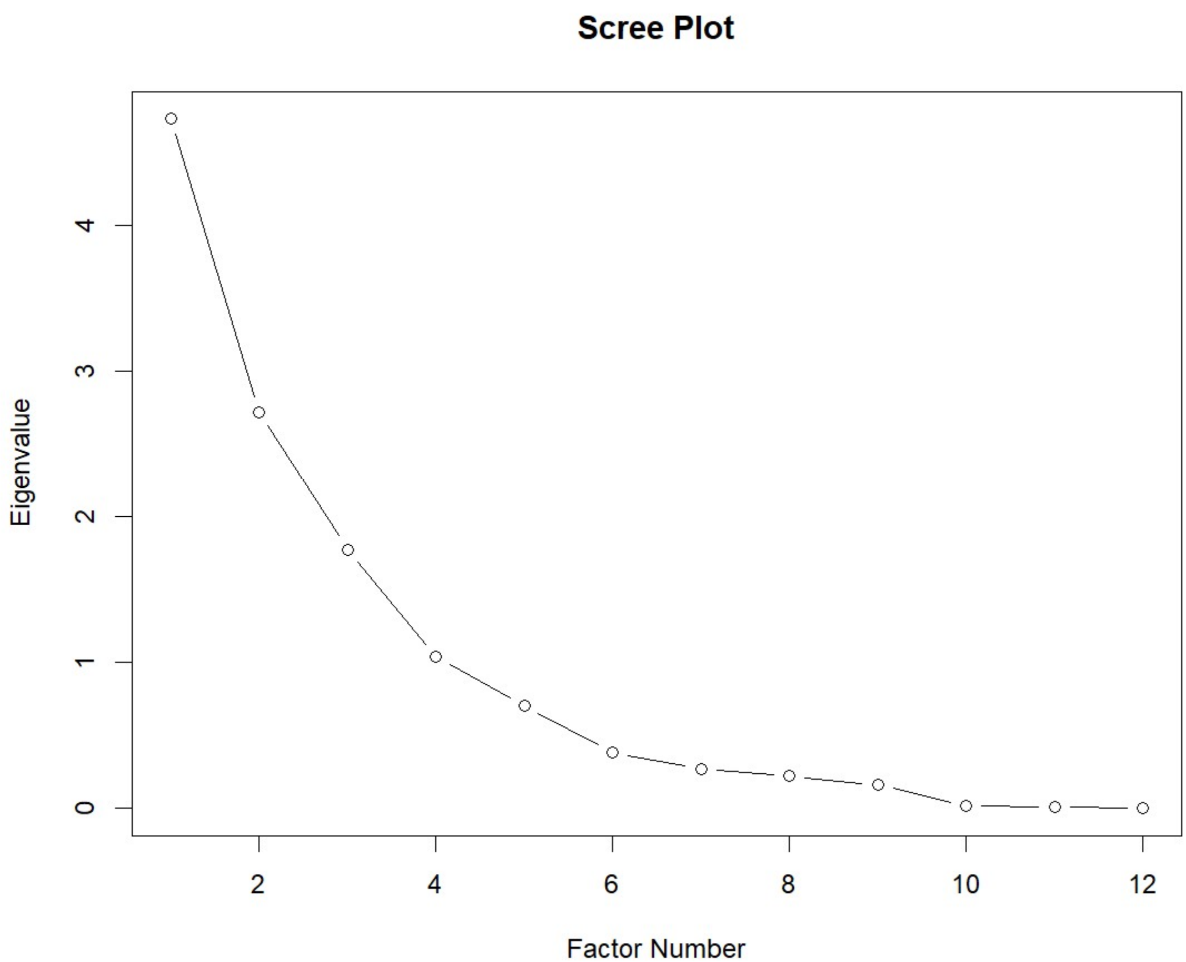
Q14. I believe the speaker/Spot provided wrong answers.

Q15. The speaker/Spot's responses were coherent and maintained a consistent tone and quality.

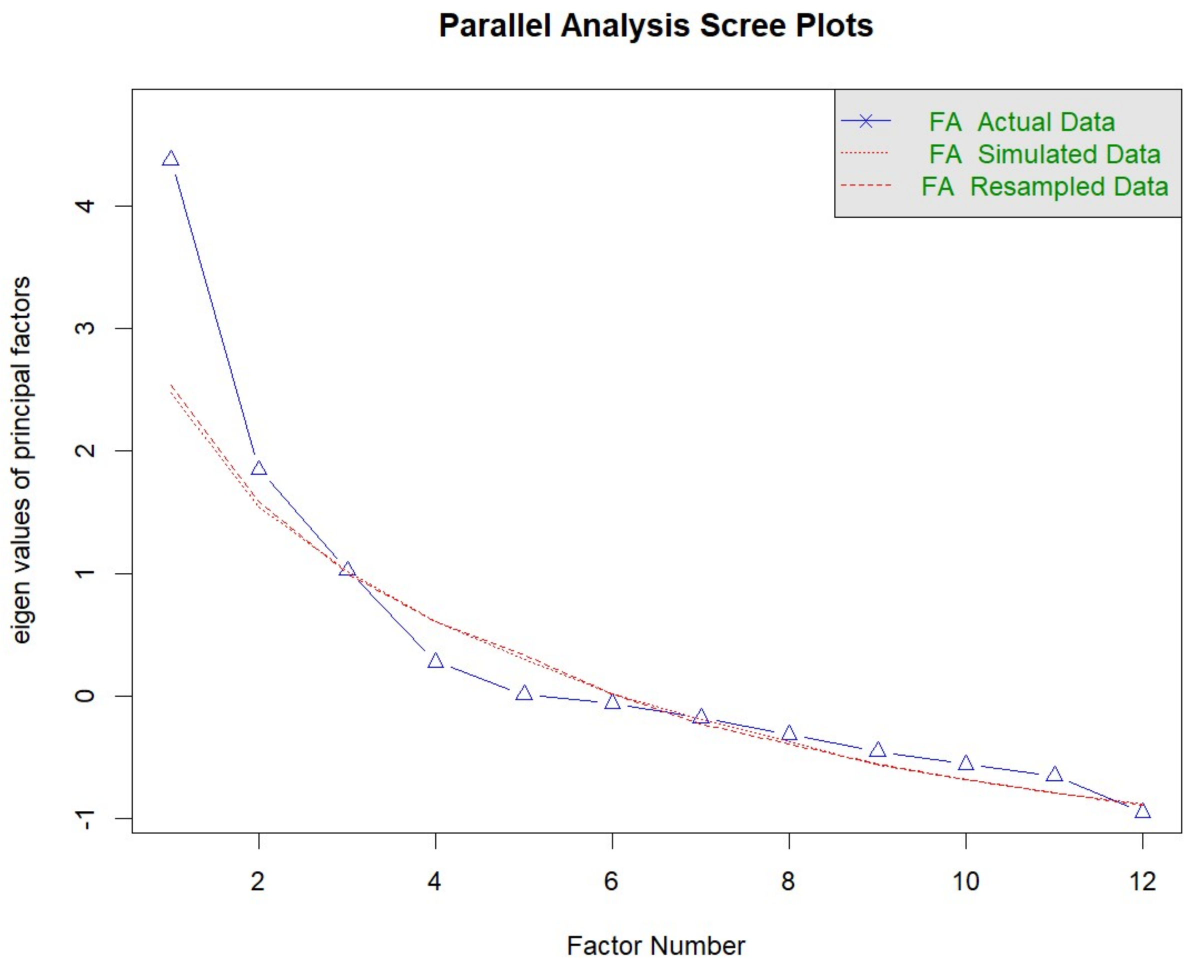
Q16. I enjoyed interacting with the speaker/Spot.

Q17. I think the speaker/Spot is approachable.

```r
# ------------------------------
# 6. Scree Plot
# ------------------------------
ev <- eigen(cor_mat)

plot(ev$values, type = "b",
     main = "Scree Plot",
     xlab = "Factor Number",
     ylab = "Eigenvalue")
```



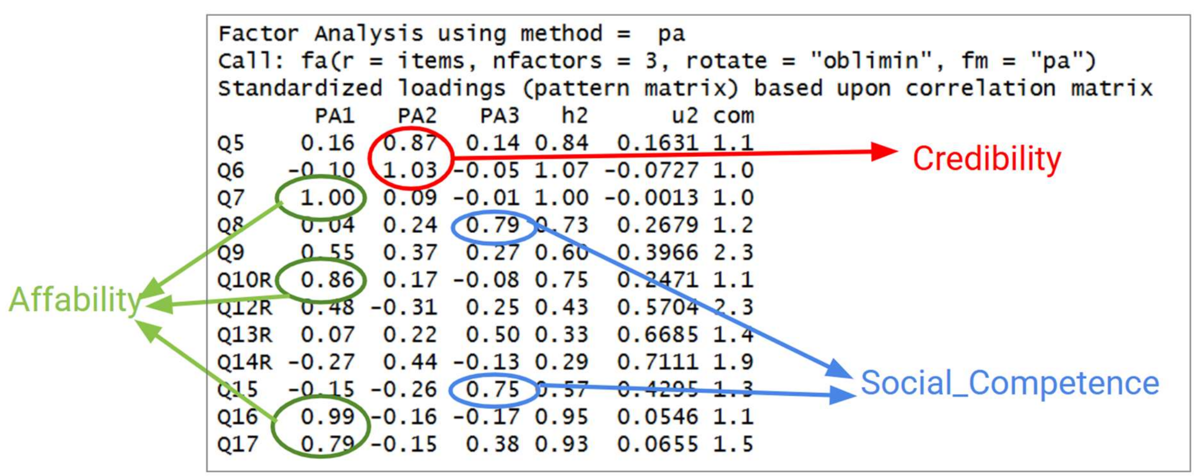

```r
fa.parallel(items, fa = "fa", fm = "pa")
fa(items, nfactors = 3, fm = "pa", rotate = "oblimin")
```


Parallel Analysis Scree Plots
FA Actual Data
FA Simulated Data
FA Resampled Data
eigen values of principal factors
Factor Number
-1
0
1
2
3
4
2
4
6
8
10
12

```r
library(psych)

# Run factor analysis
fa_result <- fa(items, nfactors = 3, fm = "pa", rotate = "oblimin")
```

```
> fa(items, nfactors = 3, fm = "pa", rotate = "oblimin")
Factor Analysis using method =  pa
Call: fa(r = items, nfactors = 3, rotate = "oblimin", fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
        PA1   PA2   PA3   h2      u2 com
Q5     0.16  0.87  0.14 0.84  0.1631 1.1
Q6    -0.10  1.03 -0.05 1.07 -0.0727 1.0
Q7     1.00  0.09 -0.01 1.00 -0.0013 1.0
Q8     0.04  0.24  0.79 0.73  0.2679 1.2
Q9     0.55  0.37  0.27 0.60  0.3966 2.3
Q10R   0.86  0.17 -0.08 0.75  0.2471 1.1
Q12R   0.48 -0.31  0.25 0.43  0.5704 2.3
Q13R   0.07  0.22  0.50 0.33  0.6685 1.4
Q14R  -0.27  0.44 -0.13 0.29  0.7111 1.9
Q15   -0.15 -0.26  0.75 0.57  0.4295 1.3
Q16    0.99 -0.16 -0.17 0.95  0.0546 1.1
Q17    0.79 -0.15  0.38 0.93  0.0655 1.5

                       PA1  PA2  PA3
SS loadings           4.10 2.52 1.88
Proportion Var        0.34 0.21 0.16
Cumulative Var        0.34 0.55 0.71
Proportion Explained  0.48 0.30 0.22
Cumulative Proportion 0.48 0.78 1.00
```

We made it for better representation:

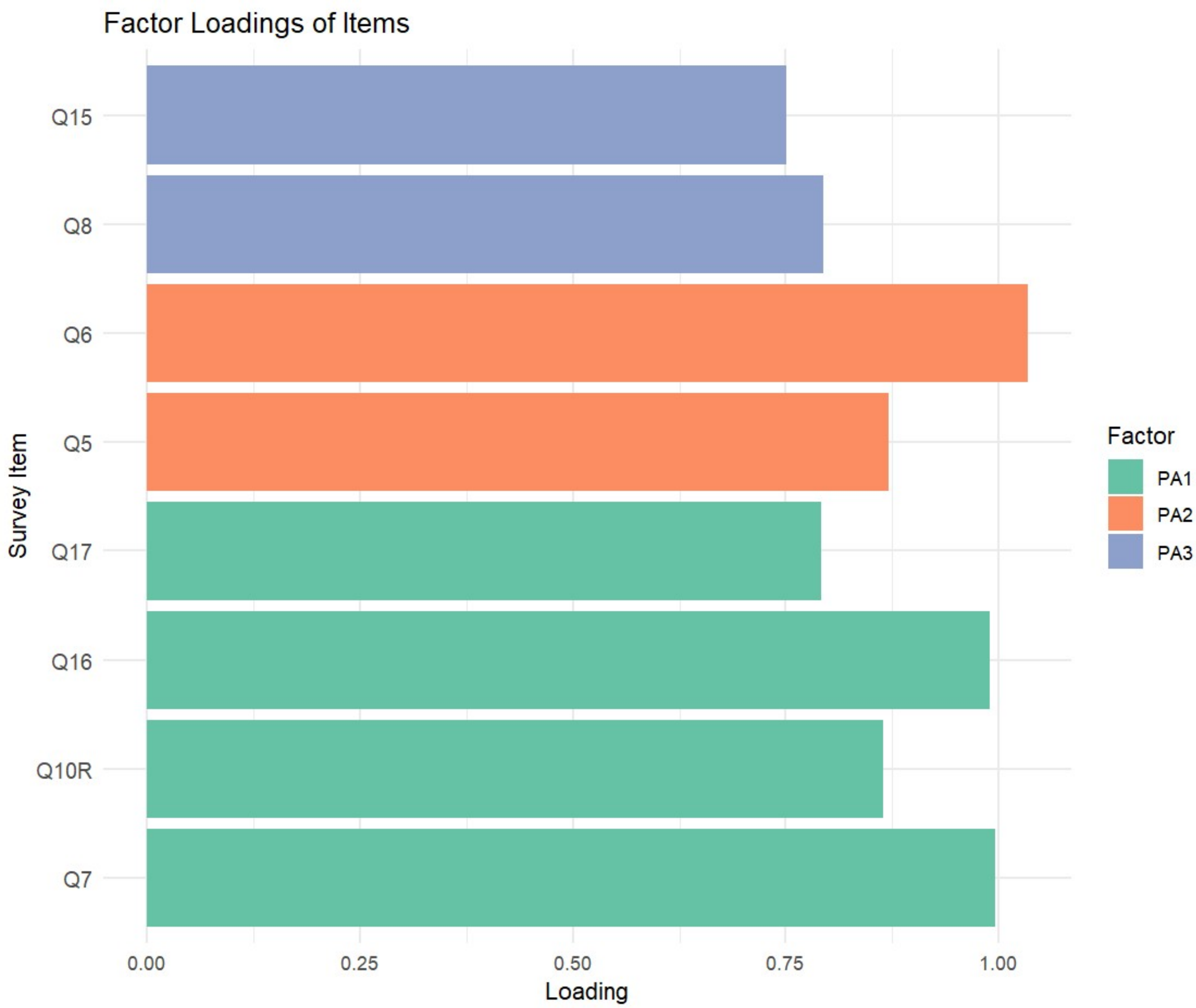


```
Factor Analysis using method =  pa
Call: fa(r = items, nfactors = 3, rotate = "oblimin", fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
        PA1   PA2   PA3   h2      u2 com
Q5     0.16  0.87  0.14 0.84  0.1631 1.1
Q6    -0.10  1.03 -0.05 1.07 -0.0727 1.0
Q7     1.00  0.09 -0.01 1.00 -0.0013 1.0
Q8     0.04  0.24  0.79 0.73  0.2679 1.2
Q9     0.55  0.37  0.27 0.60  0.3966 2.3
Q10R   0.86  0.17 -0.08 0.75  0.2471 1.1
Q12R   0.48 -0.31  0.25 0.43  0.5704 2.3
Q13R   0.07  0.22  0.50 0.33  0.6685 1.4
Q14R  -0.27  0.44 -0.13 0.29  0.7111 1.9
Q15   -0.15 -0.26  0.75 0.57  0.4295 1.3
Q16    0.99 -0.16 -0.17 0.95  0.0546 1.1
Q17    0.79 -0.15  0.38 0.93  0.0655 1.5
```

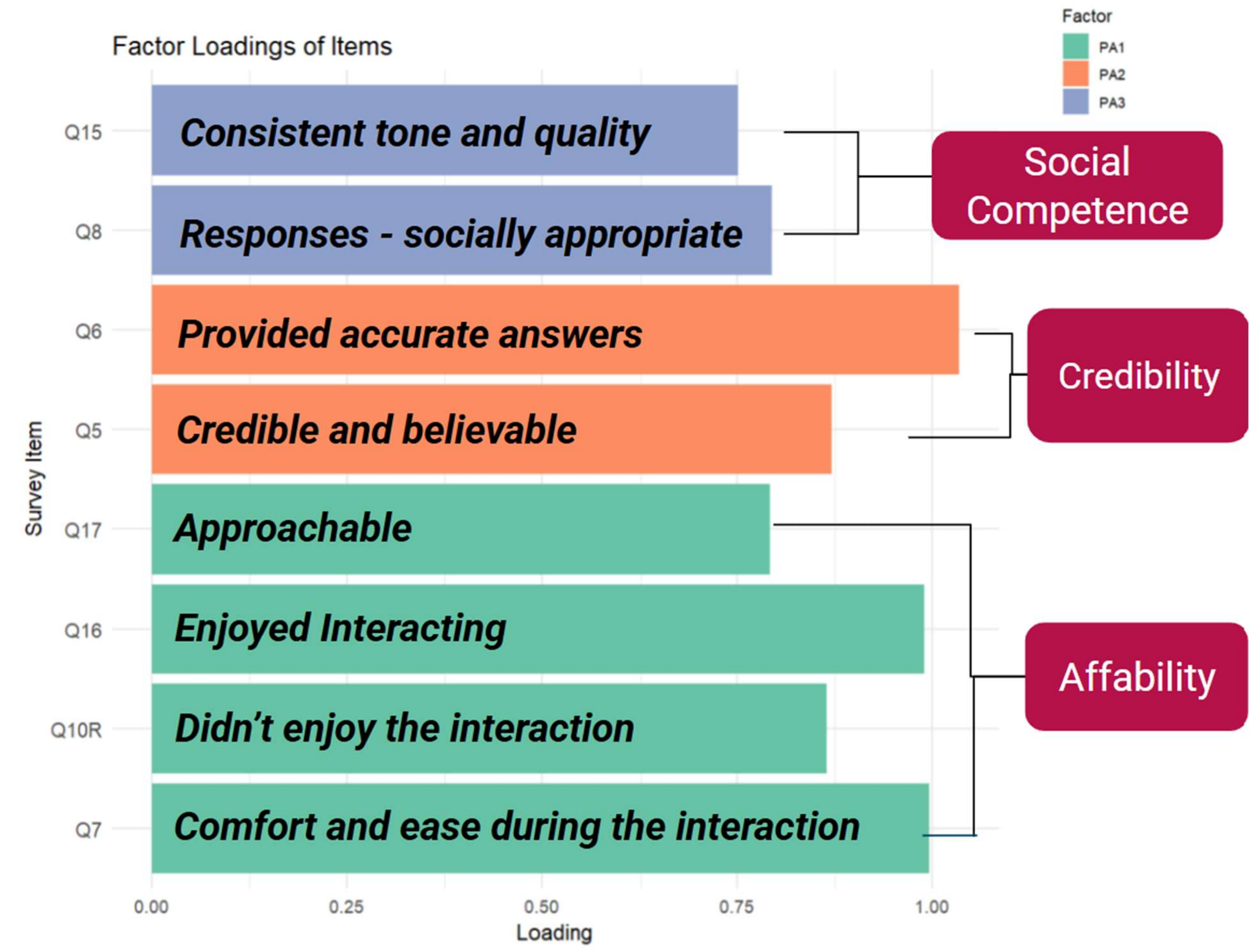
```r
# Threshold for strong loadings
threshold <- 0.7

# Initialize empty data frame to store results
summary_table <- data.frame(Factor = character(),
                            Item = character(),
                            Loading = numeric(),
                            stringsAsFactors = FALSE)

# Loop over each factor
for (factor in colnames(loadings_df)[1:(ncol(loadings_df)-1)]) {
  # Select items with loading above threshold
  temp <- loadings_df[abs(loadings_df[[factor]]) >= threshold, c("Item", factor)]
  if(nrow(temp) > 0){
    # Add factor name and rename loading column
    temp$Factor <- factor
    colnames(temp)[2] <- "Loading"
    # Reorder columns
    temp <- temp[, c("Factor", "Item", "Loading")]
    # Add to summary table
    summary_table <- rbind(summary_table, temp)
  }
}

# Reset row names
rownames(summary_table) <- NULL

# Print summary table
print(summary_table)
```

```
> print(summary_table)
  Factor Item   Loading
1    PA1   Q7 0.9965100
2    PA1 Q10R 0.8647890
3    PA1  Q16 0.9897476
4    PA1  Q17 0.7916480
5    PA2   Q5 0.8709457
6    PA2   Q6 1.0340417
7    PA3   Q8 0.7941966
8    PA3  Q15 0.7502706
```

```r
library(ggplot2)

# Assuming 'summary_table' is the data frame from previous code:
# Columns: Factor, Item, Loading

# Optional: reorder items by factor for better plotting
summary_table$Item <- factor(summary_table$Item, levels = unique(summary_table$Item))

# Plot
ggplot(summary_table, aes(x = Loading, y = Item, fill = Factor)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.7)) +
  labs(title = "Factor Loadings of Items",
       x = "Loading",
       y = "Survey Item") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2") +
  theme(axis.text.y = element_text(size = 10))
```



We made it for better representation:



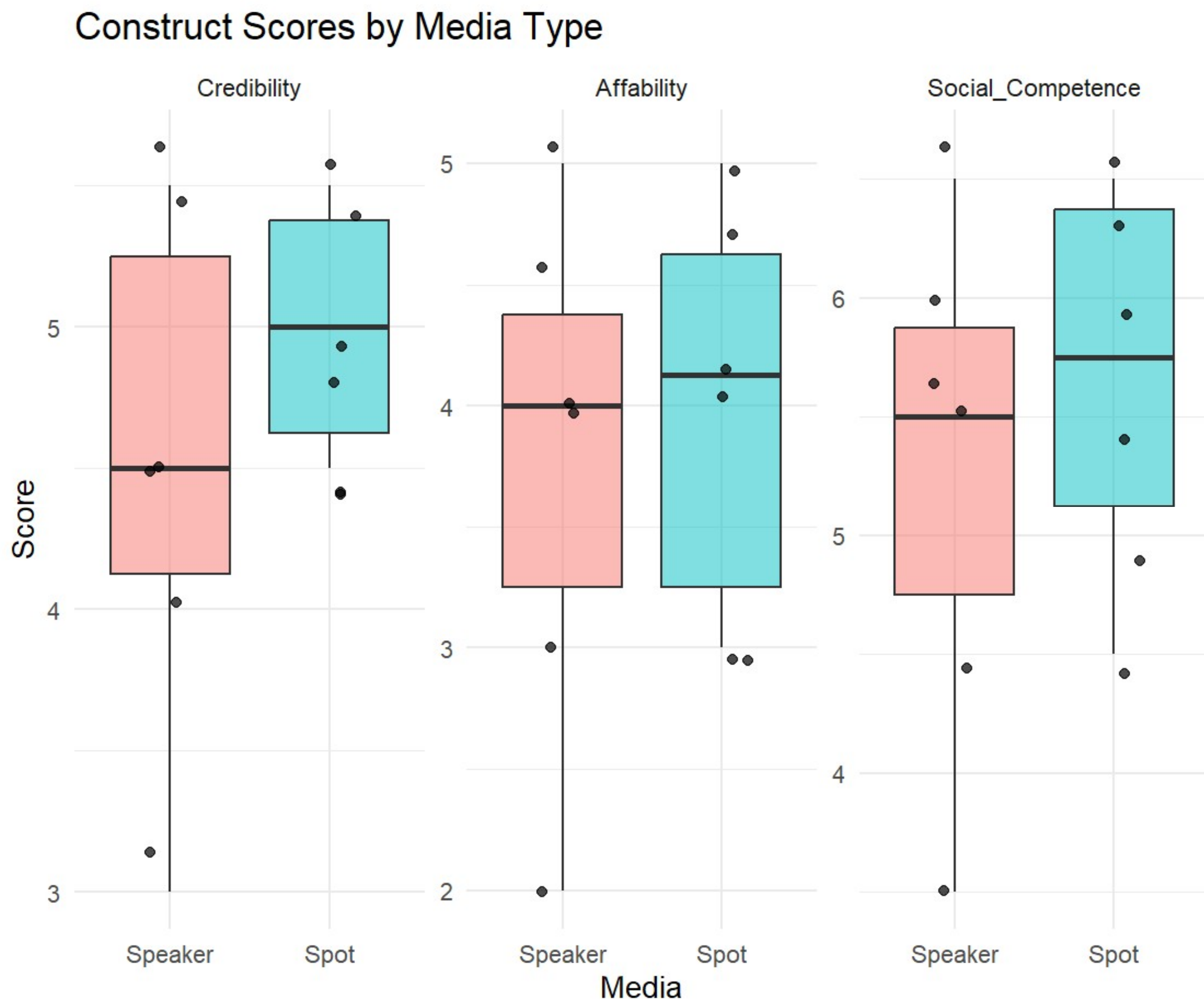
```r
# ------------------------------
# 1. Manual Cronbach's alpha function
# ------------------------------
cronbach_alpha <- function(df) {
  k <- ncol(df)
  item_var <- apply(df, 2, var, na.rm = TRUE)
  total_var <- var(rowSums(df, na.rm = TRUE), na.rm = TRUE)
  alpha <- (k / (k - 1)) * (1 - sum(item_var)/total_var)
  return(alpha)
}


# ------------------------------
# 3. Compute constructs
# ------------------------------
Credibility <- data[, c("Q5", "Q6")]
Affability    <- data[, c("Q7", "Q16", "Q17", "Q10R")]
Social_Competence  <- data[, c("Q8", "Q15")]

# ------------------------------
# 4. Compute Cronbach's alpha manually
# ------------------------------
cat("Cronbach's alpha for Credibility:", cronbach_alpha(Credibility), "\n")
cat("Cronbach's alpha for Affability:", cronbach_alpha(Affability), "\n")
cat("Cronbach's alpha for Social_Competence:", cronbach_alpha(Social_Competence), "\n")
```

```
> cat("Cronbach's alpha for Credibility:", cronbach_alpha(Credibility), "\n")
Cronbach's alpha for Credibility: 0.8933333
> cat("Cronbach's alpha for Affability:", cronbach_alpha(Affability), "\n")
Cronbach's alpha for Affability: 0.9419355
> cat("Cronbach's alpha for Social_Competence:", cronbach_alpha(Social_Competence), "\n")
Cronbach's alpha for Social_Competence: 0.5653105
```

```r
# ------------------------------
# 5. ANOVA example
# ------------------------------
data$Credibility <- rowMeans(Credibility, na.rm = TRUE)
data$Affability <- rowMeans(Affability, na.rm = TRUE)
data$Social_Competence<- rowMeans(Social_Competence, na.rm = TRUE)


constructs <- c("Credibility","Affability","Social_Competence")

for(c in constructs){
  cat("\n==========", c, "==========\n")
  print(summary(aov(data[[c]] ~ data$Media)))
}
```

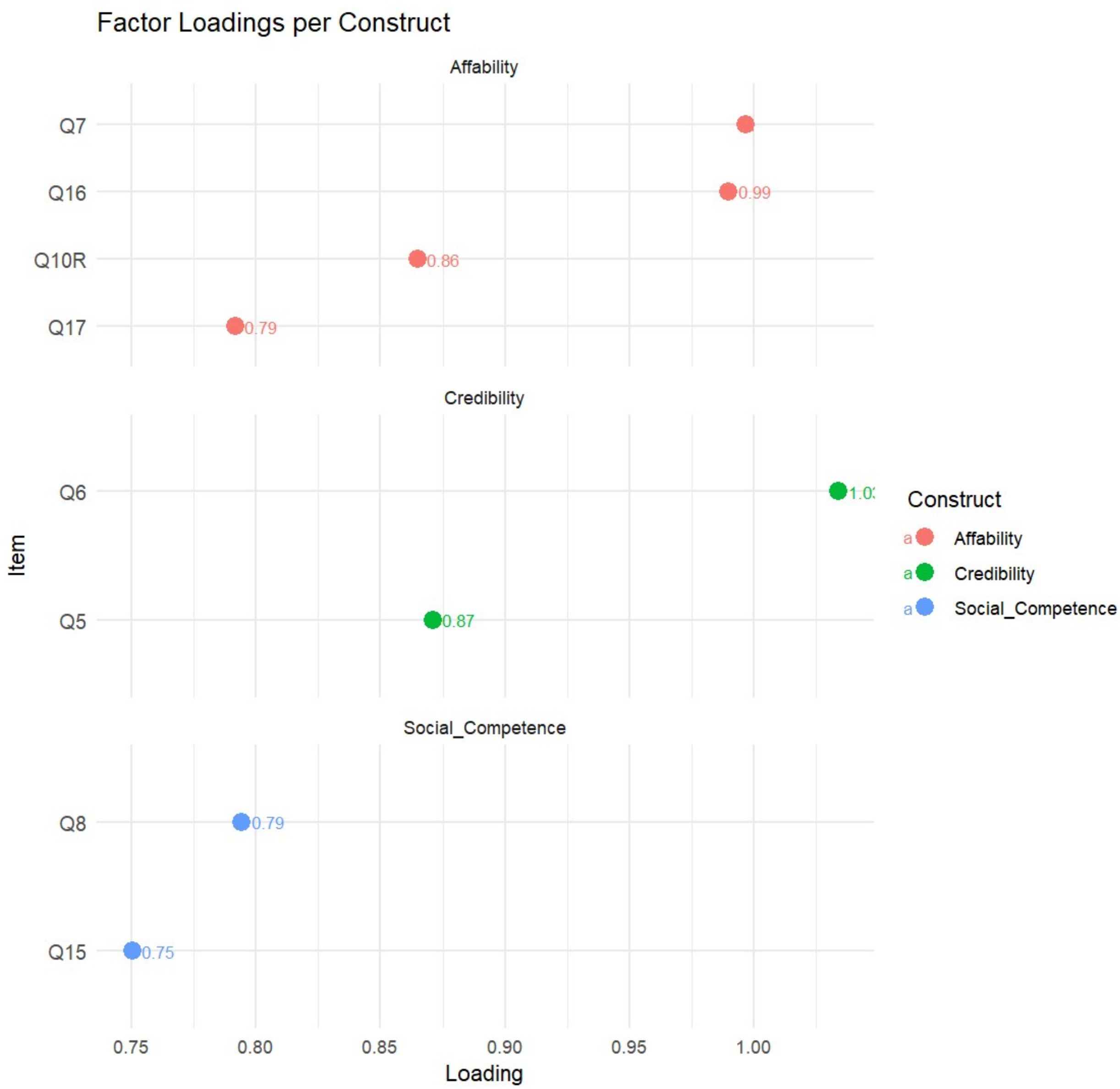
```
========== Credibility ==========
            Df Sum Sq Mean Sq F value Pr(>F)
data$Media   1   0.75    0.75   1.364   0.27
Residuals   10   5.50    0.55

========== Affability ==========
            Df Sum Sq Mean Sq F value Pr(>F)
data$Media   1  0.188  0.1875   0.197  0.666
Residuals   10  9.500  0.9500

========== Social_Competence ==========
            Df Sum Sq Mean Sq F value Pr(>F)
data$Media   1  0.521  0.5208   0.566  0.469
Residuals   10  9.208  0.9208
```

```r
library(ggplot2)
library(reshape2)

# Select the constructs for plotting
plot_data <- data[, c("Media", "Credibility", "Affability", "Social_Competence")]

# Convert data to long format for ggplot
plot_long <- melt(plot_data, id.vars = "Media",
                  variable.name = "Construct",
                  value.name = "Score")


# Boxplots with jittered points
ggplot(plot_long, aes(x = Media, y = Score, fill = Media)) +
  geom_boxplot(alpha = 0.5, outlier.shape = NA) +
  geom_jitter(width = 0.2, size = 2, alpha = 0.7) +
  facet_wrap(~Construct, scales = "free_y") +
  theme_minimal(base_size = 14) +
  labs(title = "Construct Scores by Media Type",
       x = "Media",
       y = "Score") +
  theme(legend.position = "none")
```



Construct Scores by Media Type
Credibility
Affability
Social_Competence
Score
5
4
3
5
4
3
2
6
5
4
Speaker
Spot
Speaker
Spot
Speaker
Spot
Media

```r
library(ggplot2)
library(dplyr)

# Example: summary_table from previous step
# Columns: Factor, Item, Loading
colnames(summary_table)
# Add a Construct column manually based on your EFA mapping
# Example mapping based on your previous factors
summary_table <- summary_table %>%
  mutate(Construct = case_when(
    Factor == "PA1" ~ "Affability",
    Factor == "PA2" ~ "Credibility",
    Factor == "PA3" ~ "Social_Competence"
  ))


ggplot(summary_table, aes(x = Loading, y = reorder(Item, Loading), color = Construct)) +
  geom_point(size = 4) +
  geom_text(aes(label = round(Loading, 2)), hjust = -0.3, size = 3) +
  facet_wrap(~Construct, scales = "free_y", ncol = 1) +
  labs(title = "Factor Loadings per Construct",
       x = "Loading",
       y = "Item") +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 10))
```



Factor Loadings per Construct
Affability
Q7
Q16
Q10R
Q17
0.99
0.86
0.79
Credibility
Q6
Q5
0.87
Social_Competence
Q8
Q15
0.79
0.75
Item
Loading
0.75
0.80
0.85
0.90
0.95
1.00
Construct
Affability
Credibility
Social_Competence