

```

# Project: Grit Analysis for the 2024 Industrial Maintenance Data
# 01/20/2025
# Carl Lee Tolbert
# Email: tolbert.research@tuta.com

# Disclaimer for Academic Use of R Code

# This R program is provided as supplementary material to support the findings of the
academic paper
# titled: U.S. Industrial Maintenance Workers: The Implications of Interest and Effort on
Leadership,
# Maturity, Retention, and Job Satisfaction.

# The code is made available "as is" for research and educational purposes only.

# Users are encouraged to adapt or modify the code as needed, provided proper attribution
is given.
# No Warranty: The author make no warranties, express or implied, regarding the accuracy,
completeness,
# or functionality of this code. While efforts have been made to ensure its reliability,
errors or inaccuracies
# may remain. Users are responsible for verifying the code's suitability for their
specific applications.

# Limitation of Liability: The authors shall not be held liable for any damages resulting
from the use or inability
# to use this code, including but not limited to direct, indirect, incidental, or
consequential damages.

# License: This code is licensed under the MIT License.

# Please cite the code and or the data set as appropriate.Acknowledgment: We welcome
feedback and suggestions to improve
# this program. For inquiries or to report issues, please contact the corresponding author
at tolbert.research@tuta.com.

library(tidyverse)
library(magrittr)
library(janitor)
library(ggthemes)
library(readxl)
library(PMCMRplus)
library(car)
theme_set(theme_light())
theme_update(plot.title = element_text(hjust = 0.5),
  strip.background =element_rect(fill="#F0A000"),
  strip.text = element_text(colour = 'black'))

#####
# Reading and preparing data #
#####

setwd('C:/Analysis/Grit in Maintenance/second dataset') # Change this to your path

d = read_tsv('data.tsv') %>%
  rowwise() %>%
  mutate(Interest_12 = mean(c(GS2, GS3, GS5, GS7, GS8, GS11)),
    Effort_12 = 6 - mean(c(GS1, GS4, GS6, GS9, GS10, GS12)),
    Grit_12 = mean(c(Interest_12, Effort_12)),
    Interest = mean(c(GS2, GS5, GS7, GS8)),
    Effort = 6 - mean(c(GS4, GS6, GS9, GS12)),
    Grit = mean(c(Interest, Effort)),

    Extraversion = mean(c(E1, 6-E2, E3, 6-E4, E5, 6-E6, E7, 6-E8, E9, 6-E10)),

```

```

        Neuroticism = mean(c(N1, 6-N2, N3, 6-N4, N5, N6, N7, N8, N9, N10)),
        Agreeableness = mean(c(6-A1, A2, 6-A3, A4, 6-A5, A6, 6-A7, A8, A9, A10)),
        Conscientiousness = mean(c(C1, 6-C2, C3, 6-C4, C5, 6-C6, C7, 6-C8, C9, C10)),
        Openness = mean(c(O1, 6-O2, O3, 6-O4, O5, 6-O6, O7, O8, O9, O10))) %>%
filter(familysize < 30, age < 100)

d_2024 = read_xlsx('Grit in Engineering and Maintenance (combined).xlsx') %>%
  rowwise() %>%
  mutate(Interest = 6 - mean(c(Q1, Q3, Q5, Q6)),
         Effort = mean(c(Q2, Q4, Q7, Q8)),
         Grit = mean(c(Interest, Effort)),
         Experience_30 = Years >= 30,
         Experience_20 = Years >= 20,
         Experience_10 = Years >= 10) %>%
  ungroup() %>%
  rename(Retention = Q9,
         Satisfaction = Q10) %>%
  filter(Years < 75)

d = d %>% mutate(
  introelapse_log = log(introelapse),
  testelapse_log = log(testelapse),
  surveyelapse_log = log(surveyelapse)
)

# This shows why 3 variables above were log-transformed
ggplot(d) +
  aes(introelapse) +
  geom_histogram(fill = 'orange', color='black')
ggplot(d) +
  aes(log(introelapse)) +
  geom_histogram(fill = 'orange', color='black')
ggplot(d) +
  aes(testelapse) +
  geom_histogram(fill = 'orange', color='black')
ggplot(d) +
  aes(log(testelapse)) +
  geom_histogram(fill = 'orange', color='black')
ggplot(d) +
  aes(surveyelapse) +
  geom_histogram(fill = 'orange', color='black')
ggplot(d) +
  aes(log(surveyelapse)) +
  geom_histogram(fill = 'orange', color='black')

#####
# Comparison with 2024 data #
#####

t.test(d[d$gender == 1, 'Interest'], d[d$gender == 2, 'Interest'])
t.test(d[d$gender == 1, 'Effort'], d[d$gender == 2, 'Effort'])

t.test(d_2024$Interest, d$Interest)
t.test(d_2024$Effort, d$Effort)
t.test(d_2024$Grit, d$Grit)

#####
# Correlation tests #
#####

correlation_tests = function(var_list) {
  # Create a directory for plots
  if (!dir.exists('correlation plots')) dir.create('correlation plots')

  # Initialize an empty dataframe to store results

```

```

results = data.frame(
  Variable = character(),
  Variable_2 = character(),
  Correlation = numeric(),
  CI_Lower = numeric(),
  CI_Upper = numeric(),
  P_Value = character(),
  stringsAsFactors = FALSE
)

# Loop through each variable and conduct two tests
for (var in var_list) {
  for (var2 in c('Interest', 'Effort')) {
    if (var != var2) { # Avoid self-correlation
      test_result = cor.test(d[[var]], d[[var2]])

      # Save results
      new_row = data.frame(
        Variable = var,
        Variable_2 = var2,
        Correlation = test_result$estimate %>% round(3),
        CI_Lower = test_result$conf.int[1] %>% round(2),
        CI_Upper = test_result$conf.int[2] %>% round(2),
        P_Value = ifelse(test_result$p.value < 0.001, '< 0.001',
                          as.character(round(test_result$p.value, 3)))
      )
      results = results %>%
        bind_rows(new_row)

      # Save scatter plot
      plot = ggplot(d, aes_string(x=var, y=var2)) +
        geom_jitter(shape=1, alpha=.7, color='midnightblue', width=.1, height=.1) +
        geom_smooth(method='lm', se=FALSE) +
        labs(title = paste(var, 'vs', var2),
              x = var,
              y = var2,
              subtitle = paste('p-value:', new_row$P_Value))

      plot_filename = paste0('correlation plots/', var, '_vs_', var2, '.png')
      ggsave(plot_filename, plot)
    }
  }
}

return(results)
}

corr_results = correlation_tests(c('Extraversion', 'Neuroticism', 'Agreeableness',
                                  'Conscientiousness', 'Openness', 'age', 'education',
                                  'familysize', 'introelapse_log', 'testelapse_log',
                                  'surveyelapse_log'))

corr_results %>% write_csv('Correlation tests.csv')
corr_results

#####
# ANOVA tests #
#####

anova_tests = function(cat_vars, labels = list()) {

  # Create a directory for plots and post hoc tests
  if (!dir.exists('anova_plots')) dir.create('anova_plots')
  if (!dir.exists('post_hoc_tests')) dir.create('post_hoc_tests')

```

```

# Initialize results dataframe
results = data.frame(
  Variable = character(),
  Dependent = character(),
  P_Value = character(),
  stringsAsFactors = FALSE
)

# Loop through each categorical variable
for (cat_var in cat_vars) {
  for (dep_var in c('Interest', 'Effort')) {

    # Ensure categorical variable is a factor
    d[[cat_var]] = as.factor(d[[cat_var]])
    d_filtered = d[d[[cat_var]] != '0', ]

    # Perform ANOVA
    formula = as.formula(paste(dep_var, '~', cat_var))
    aov_result = aov(formula, data = d_filtered)
    summary_result = summary(aov_result)

    # Extract F-statistic and p-value
    f_stat = summary_result[[1]]$`F value`[1]
    p_value = summary_result[[1]]$`Pr(>F)`[1]
    p_value_label = ifelse(p_value < 0.001, '< 0.001', as.character(round(p_value, 3)))

    # Add results to the dataframe
    results = results %>%
      bind_rows(data.frame(
        Variable = cat_var,
        Dependent = dep_var,
        P_Value = p_value_label
      ))

    #####
    # Plot
    #####

    # Combine Interest and Effort into a single plot
    d_melted = d_filtered %>%
      select(.data[[cat_var]], Interest, Effort) %>%
      pivot_longer(cols = c(Interest, Effort), names_to = 'Dependent', values_to =
'Value')

    # Extract p-values from the results dataframe
    p_value_interest = results %>%
      filter(Variable == cat_var & Dependent == 'Interest') %>%
      pull(P_Value)

    p_value_effort = results %>%
      filter(Variable == cat_var & Dependent == 'Effort') %>%
      pull(P_Value)

    # Get custom labels for the current variable if provided
    x_labels = labels[[cat_var]]

    # Create Combined Barplot
    plot = ggplot(d_melted, aes_string(x = cat_var, y = 'Value', fill = 'Dependent')) +
      stat_summary(fun = mean, geom = "bar", position = "dodge") +
      stat_summary(fun.data = mean_cl_boot, geom = "errorbar", position =
position_dodge(0.9), width = 0.2) +
      scale_fill_manual(values = c('Interest' = '#1b9e77', 'Effort' = '#d95f02')) +
      scale_x_discrete(labels = x_labels) +
      labs(

```

```

    title = cat_var,
    subtitle = paste('p-value (Interest):', p_value_interest,
                     '| p-value (Effort):', p_value_effort),
    x = '',
    y = 'Average score',
    fill = 'Dependent Variable'
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = 'top'
  )

  # Save the plot
  plot_filename = paste0('anova_plots/', cat_var, '_combined.png')
  ggsave(plot_filename, plot, height = 6, width = 8)
}}

```

```

# Pivot results into wide format
results = results %>%
pivot_wider(
  names_from = Dependent,
  values_from = P_Value
) %>%
  rename(`Interest (p-value)` = Interest,
         `Effort (p-value)` = Effort)

return(results)
}

```

```

anova_results = anova_tests(c('urban', 'religion', 'married', 'voted', 'race',
'education'),

```

```

  list(
    urban = c(
      '1' = 'Rural (countryside)',
      '2' = 'Suburban',
      '3' = 'Urban (town, city)'
    ),

    education = c(
      '1' = 'Less than high school',
      '2' = 'High school',
      '3' = 'University degree',
      '4' = 'Graduate degree'
    ),

    religion = c(
      '1' = 'Agnostic',
      '2' = 'Atheist',
      '3' = 'Buddhist',
      '4' = 'Christian (Catholic)',
      '5' = 'Christian (Mormon)',
      '6' = 'Christian (Protestant)',
      '7' = 'Christian (Other)',
      '8' = 'Hindu',
      '9' = 'Jewish',
      '10' = 'Muslim',
      '11' = 'Sikh',
      '12' = 'Other'
    ),

    married = c(
      '1' = 'Never married',
      '2' = 'Currently married',
      '3' = 'Previously married'
    )
  )

```

```

    ),
    voted = c(
      '1' = 'Yes',
      '2' = 'No'
    ),
    race = c(
      '1' = 'Asian',
      '2' = 'Arab',
      '3' = 'Black',
      '4' = 'Indigenous/Native American/White',
      '5' = 'Other'
    )
  ))

```

```

anova_results %>% write_csv('ANOVA tests.csv')
anova_results

```

```

#####
# VCL t-tests #
#####

```

```

VCL_tests = data.frame(
  Dependent = character(),
  Variable = character(),
  Word = character(),
  p_value = character(),
  Mean_Difference = numeric(),
  CI_Lower = numeric(),
  CI_Upper = numeric(),
  stringsAsFactors = FALSE
)

```

```

# Define variable-word mapping
words = c(
  'boat', 'incoherent', 'pallid', 'robot', 'audible', 'cuivocal',
  'paucity', 'epistemology', 'florted', 'decide', 'pastiche', 'verdid',
  'abysmal', 'lucid', 'betray', 'funny'
)

```

```

# Loop through both dependent variables and VCL variables
for (dep_var in c('Interest', 'Effort')) {
  for (i in seq_along(words)) {
    var = paste0('VCL', i)

```

```

    # Perform t-test
    test_result = t.test(
      d[d[[var]] == 1, dep_var],
      d[d[[var]] == 0, dep_var]
    )

```

```

    # Extract relevant results
    VCL_tests = VCL_tests %>%
      bind_rows(data.frame(
        Dependent = dep_var,
        Variable = var,
        Word = words[i],
        p_value = ifelse(test_result$p.value < 0.001, '< 0.001',
as.character(round(test_result$p.value, 3))),
        Mean_Difference = ifelse(test_result$p.value < 0.1,
round(-1*diff(test_result$estimate), 3), NA),
        CI_Lower = ifelse(test_result$p.value < 0.1, round(test_result$conf.int[1], 3),
NA),
        CI_Upper = ifelse(test_result$p.value < 0.1, round(test_result$conf.int[2], 3),

```

```

NA),
    prop_know = paste0((sum(d[[var]] == 1) / nrow(d) * 100) %>% round(), '%')
  ))
}
}

# Display the results
print(VCL_tests)
VCL_tests %>% write_csv('VCL t-tests.csv')

#####
# Regression #
#####

interest_full_model = lm(Interest ~ Extraversion + Neuroticism + Agreeableness +
                          Conscientiousness + Openness + age + education + religion +
                          urban + surveyelapse_log + familysize + voted + married, d)

summary_1 = summary(interest_full_model)
summary_interest = data.frame(
  Term = rownames(coef(summary_1)),
  Estimate = coef(summary_1)[, "Estimate"] %>% round(3),
  vif = c(NA, vif(interest_full_model)) %>% round(2),
  `p-value` = coef(summary_1)[, "Pr(>|t|)"] %>% round(3)
)
summary_interest %>% write_csv('Regression interest.csv')

efforts_full_model = lm(Effort ~ Extraversion + Neuroticism + Agreeableness +
                        Conscientiousness + Openness + age + education + religion +
                        urban + surveyelapse_log + familysize + voted + married, d)

summary_2 = summary(efforts_full_model)
summary_effort = data.frame(
  Term = rownames(coef(summary_2)),
  Estimate = coef(summary_2)[, "Estimate"] %>% round(3),
  vif = c(NA, vif(efforts_full_model)) %>% round(2),
  `p-value` = coef(summary_2)[, "Pr(>|t|)"] %>% round(3)
)
summary_effort %>% write_csv('Regression effort.csv')

# Print in Console (Copy from Here)
cat(model_summary_text, sep = "\n")

#####
# Leadership + Experience #
#####

table(d_2024 %>% select(Leadership, Experience_10))

d_2024 %>%
  filter(!is.na(Leadership)) %>%
  ggplot() +
    aes(Years, Effort, color = Leadership) +
    geom_point() +
    facet_wrap(~ Leadership, ncol=1)

d_leader = d_2024 %>% filter(Leadership == T)
d_notleader = d_2024 %>% filter(Leadership == F)

cor.test(d_leader$Years, d_leader$Interest)
cor.test(d_notleader$Years, d_notleader$Interest)
cor.test(d_leader$Years, d_leader$Effort)
cor.test(d_notleader$Years, d_notleader$Effort)

```

```
t.test(Interest ~ Experience_10, d_leader)
t.test(Interest ~ Experience_10, d_notleader)
t.test(Interest ~ Experience_10, d_leader)
t.test(Interest ~ Experience_10, d_notleader)

d_2024 %>% filter(Leadership == F, Experience_10 == F) %>% select(Interest) %>% t.test()
d_2024 %>% filter(Leadership == T, Experience_10 == F) %>% select(Interest) %>% t.test()
d_2024 %>% filter(Leadership == F, Experience_10 == T) %>% select(Interest) %>% t.test()
d_2024 %>% filter(Leadership == T, Experience_10 == T) %>% select(Interest) %>% t.test()

d_2024 %>% filter(Leadership == F, Experience_10 == F) %>% select(Effort) %>% t.test()
d_2024 %>% filter(Leadership == T, Experience_10 == F) %>% select(Effort) %>% t.test()
d_2024 %>% filter(Leadership == F, Experience_10 == T) %>% select(Effort) %>% t.test()
d_2024 %>% filter(Leadership == T, Experience_10 == T) %>% select(Effort) %>% t.test()
```