# POL501 - Answers to Problem Set 3

[redacted]

2024-11-13

## Table of Contents

## Question 1

```
# Set up the working directory
# Use the '/' character or '\' to separate folders
setwd("C:/Users[redacted]Desktop/Fall 2024/1. POL 501/Problem Set/Problem Set
3")

# Confirm the working directory
getwd()

## [1] "C:/Users/ssgal/Desktop/Fall 2024/1. POL 501/Problem Set/Problem Set
3"

# Load the dataset
load('dataframe-pew.RData')

# Check the loaded objects (the created function and the dataframe should be
printed as a result)
ls()

## [1] "confidence_interval" "df_clean"
```

### Answer to (1.a)

```
# Question 1.a: Use the function `confidence_interval` to compute the mean of
`CRIMESAFE` with a 95% confidence level
crimesafe_CI_z_95 <- confidence_interval(data = df_clean, var_name =
'CRIMESAFE', confidence_level = 0.95, method = "z")

print(crimesafe_CI_z_95)

## Sample Mean of        Estimate             MOE  Lower CI Bound  Upper CI Bound
##     "CRIMESAFE"         "2.575"         "0.023"         "2.552"         "2.598"
```

**Explanation/Justification**: The sample mean of the variable CRIMESAFE is 2.575, which is used as the center value for the confidence interval. The margin of error (MOE) is 0.023, which represents the range within which the true population mean is likely to fall, given the sample data and the 95% confidence level.

The lower and upper CI bounds can be obtained by subtracting or adding the MOE from the sample mean of the variable CRIMESAFE, respectively.

Lower CI Bound = Sample Mean of CRIMESAFE (Estimate) – MOE = 2.575 - 0.023 = 2.552

Upper CI Bound = Sample Mean of CRIMESAFE (Estimate) + MOE = 2.575 + 0.023 = 2.598

95% CI = [2.552, 2.598]

Therefore, we can say that we are 95% confident that the true population mean of CRIMESAFE lies between 2.552 and 2.598.

## Answer to (1.b)

```
# Question 1.b: Use the function `confidence_interval` to compute the mean of `CRIMESAFE` with a 99% confidence level
crimesafe_CI_z_99 <- confidence_interval( data = df_clean, var_name = 'CRIMESAFE', confidence_level = 0.99, method = "z")

print(crimesafe_CI_z_99)

## Sample Mean of         Estimate              MOE  Lower CI Bound  Upper CI Bound
##      "CRIMESAFE"          "2.575"           "0.03"         "2.545"         "2.606"
```

**Explanation/Justification**: As in (1.a), the sample mean of the variable CRIMESAFE is 2.575. However, the margin of error (MOE) is 0.03, which represents the range within which the true population mean is likely to fall, given the sample data and the 99% confidence level.

Lower CI Bound = Sample Mean of CRIMESAFE (Estimate) – MOE = 2.575 - 0.03 = 2.545

Upper CI Bound = Sample Mean of CRIMESAFE (Estimate) + MOE = 2.575 + 0.03 = 2.605 (2.606)

99% CI = [2.545, 2.606]

Therefore, we can say that we are 99% confident that the true population mean of CRIMESAFE lies between 2.545 and 2.606. This interval is broader than the 95% CI from (1.a), reflecting the increased certainty required for a 99% confidence level. A higher confidence level increases the interval's width because it must encompass a broader range of values to provide greater certainty that it includes the true mean.

## Answer to (1.c)

```
# Question 1.c: Use the `dplyr` command 'filter' to create a data frame for
the responses of Democrats and Republicans
# Create one data frame for Democrats and another for Republicans

df_dems <- filter(df_clean, PARTY == 2)
df_rep <- filter(df_clean, PARTY == 1)
```

**Explanation/Justification**: Using the 'filter' function in the 'dplyr' package, we can separate the 'df_clean' dataset into two groups: Democrats (df_dems) and Republicans (df_rep). This is done by filtering on the PARTY variable, where 'PARTY == 2' indicates Democrats and 'PARTY == 1' indicates Republicans, according to the survey questionnaire (2024 National Public Opinion Reference Survey Online Questionnaire). This approach allows us to compare CRIMESAFE perceptions across these two parties.

## Answer to (1.d)

```
# Question 1.d: Compute the sample mean for `CRIMESAFE` and 95% confidence
intervals for Democrats and Republicans
crimesafe_CI_z_95_dems <- confidence_interval(data = df_dems, var_name =
'CRIMESAFE', confidence_level = 0.95, method = "z")

cat('\nResults for Democrats:\n')

##
## Results for Democrats:

print(crimesafe_CI_z_95_dems)

## Sample Mean of    Estimate          MOE Lower CI Bound Upper CI Bound
##    "CRIMESAFE"     "2.526"       "0.04"        "2.486"        "2.566"

crimesafe_CI_z_95_reps <- confidence_interval(data = df_rep, var_name =
'CRIMESAFE', confidence_level = 0.95, method = "z")

cat('\nResults for Republicans:\n')

##
## Results for Republicans:

print(crimesafe_CI_z_95_reps)

## Sample Mean of    Estimate          MOE Lower CI Bound Upper CI Bound
##    "CRIMESAFE"     "2.596"      "0.041"        "2.556"        "2.637"
```

**Explanation/Justification:** For Democrats, the mean perception of community safety is 2.526 with an MOE of 0.04. So,

95% CI for Democrats = [2.486, 2.566].

For Republicans, the mean is 2.596 with an MOE of 0.041. So,

95% CI for Republicans = [2.556, 2.637].

In these results, Republicans have a larger sample mean and a slightly wider 95% confidence interval compared to Democrats.

## Answer to (1.e)

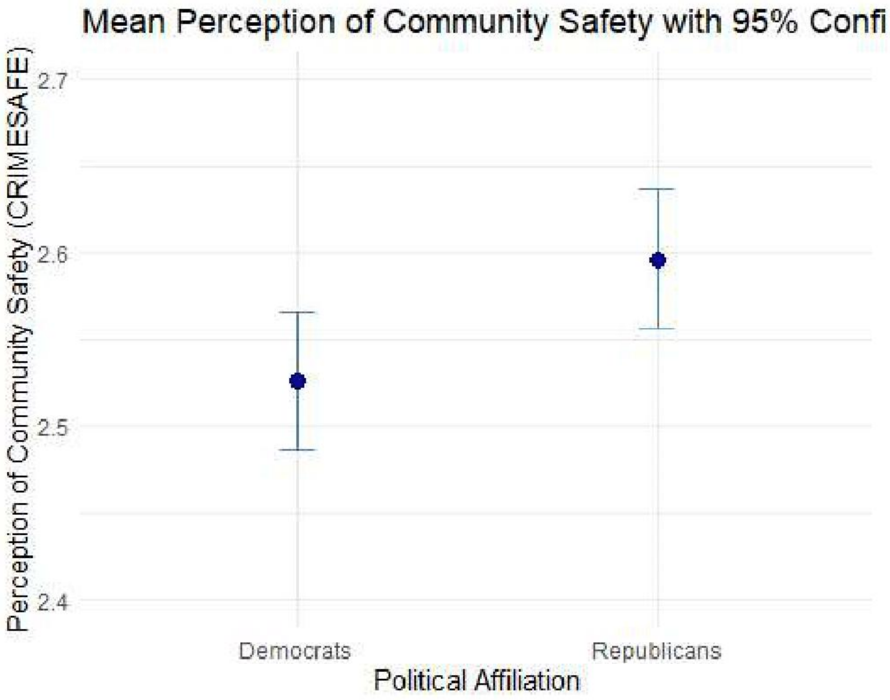
```
# Question 1.e: Create a data frame with the results of (1.d) and plot the
results
# Extract the values and combine them into a data frame for plotting
combined_crimesafe_CI_95 <- cbind(c(Group = 'Democrats',
crimesafe_CI_z_95_dems),
                                  c(Group = 'Republicans',
crimesafe_CI_z_95_reps))

# Convert the combined matrix into a data frame with proper column names
df_combined_CI <- as.data.frame(t(combined_crimesafe_CI_95), stringsAsFactors
= FALSE)

# Rename columns for clarity
colnames(df_combined_CI) <- c("Group", "Sample_Mean", "Estimate", "MOE",
"Lower", "Upper")

# Convert relevant columns to numeric
df_combined_CI <- df_combined_CI %>%
  mutate(Estimate = as.numeric(Estimate),
         Lower = as.numeric(Lower),
         Upper = as.numeric(Upper))

# Plot using ggplot2 with a dot and whiskers for confidence interval
ggplot(df_combined_CI, aes(x = Group, y = Estimate)) +
  geom_errorbar(aes(ymin = Lower, ymax = Upper), width = 0.1, color =
"steelblue") + # Whiskers for CI
  geom_point(size = 3, color = "navyblue") + # Dot for the sample mean
  labs(
    title = "Mean Perception of Community Safety with 95% Confidence
Interval",
    x = "Political Affiliation",
    y = "Perception of Community Safety (CRIMESAFE)"
  ) +
  theme_minimal() +
  ylim(2.4, 2.7)  # Set the y-axis limits here
```



**Explanation/Justification**: The 95% confidence intervals for the mean of CRIMESAFE of Democrats and Republicans show slight differences in perceptions of community safety in terms of crime. It is visualized using a dot and whiskers plot with vertical error bars from the 'geom_errorbar' function in 'ggplot2' package of R. The plot clearly contrasts the means and confidence intervals, with error bars that almost do not overlap, reinforcing the idea that Republicans generally feel safer in their communities.

However, the minimal overlap of the confidence intervals indicates that the difference in perceptions may not be large enough to draw definitive conclusions without further analysis such as hypothesis testing. This graphical representation helps highlight the disparity in perceived safety across political affiliations more clearly than numbers alone.

# Question 2

## Answer to (2.a)

```
# Question 2.a: Create a new variable in the dataframe `df_clean` that equals
1 if the respondent selected
# "Not too safe" or "Not at all safe", and zero otherwise. Then, recreate the
subsample data frames for Democrats and Republicans.
# Finally, interpret the mean of this variable.

df_clean <- df_clean %>%
  mutate(
    notsafe_binary = if_else(CRIMESAFE %in% c(4, 5), 1, 0)
  )

# Create subsamples for Democrats and Republicans
df_dems <- filter(df_clean, PARTY == 2)
df_rep <- filter(df_clean, PARTY == 1)

# Describe the mean of the new variable - entire sample
mean(df_clean$notsafe_binary)

## [1] 0.107136

# Describe the mean of the new variable for Democrats
mean(df_dems$notsafe_binary)

## [1] 0.1097493

# Describe the mean of the new variable for Republicans
mean(df_rep$notsafe_binary)

## [1] 0.0984148
```

**Explanation/Justification**: Based on the survey questionnaire (2024 National Public Opinion Reference Survey Online Questionnaire), we can assign a value of 1 if respondents reported “Not too safe” or “Not at all safe” (corresponding to CRIMESAFE values of 4 or 5) and 0 otherwise. This converts the five-level CRIMESAFE variable into a new binary variable that captures whether a respondent feels unsafe. This makes it easier to analyze the proportion of respondents who feel unsafe.

The means represent the proportion of respondents in each group, Democrats (df_dems) and Republicans (df_rep), who feel unsafe. For the entire sample, about 10.71% feel unsafe (mean = 0.107136). Among Democrats, about 10.97% feel unsafe (mean = 0.1097493), while about 9.84% of Republicans report feeling unsafe (mean = 0.0984148). These proportions allow us to compare the feeling of unsafe between the two groups, and a slightly higher proportion of Democrats report feeling unsafe compared to Republicans.

## Answer to (2.b)

```
# Question 2.b: Using the subsample for Democrats, compute the p-value with
the null hypothesis
# that the true proportion for Democrats differs from the sample proportion
for Republicans.

prop_null_hypothesis <- mean(df_rep$notsafe_binary) # hypothesis = the sample
prop for reps
n_sample_size <- nrow(df_dems)

# Calculate the standard DEVIATION for the null hypothesis (replace with the
correct formula)
SD_null_hypot <- sqrt(prop_null_hypothesis * (1- prop_null_hypothesis))

# Run the z-test to test if the mean is significantly different from the
sample proportion for Republicans
#   The function will compute in the background the correct Standard Error.
z_test_result <- z.test(
  x = df_dems$notsafe_binary, # Here we declare the variable for our sample,
i.e., democrats
  mu = prop_null_hypothesis,  # Hypothesized population mean
  sigma.x = SD_null_hypot   # Population standard DEVIATION (given H0)
)

# Print the z-test result
print(z_test_result)

##
##  One-sample z-Test
##
## data:  df_dems$notsafe_binary
## z = 1.6121, p-value = 0.1069
## alternative hypothesis: true mean is not equal to 0.0984148
## 95 percent confidence interval:
##  0.0959693 0.1235293
## sample estimates:
## mean of x
## 0.1097493
```

**Explanation/Justification**: We can compute a one-sample z-test for the proportion of Democrats who feel “Not too safe” or “Not at all safe,” testing against the null hypothesis that this proportion does not differ from the corresponding sample proportion for Republicans. The null hypothesis ($H_0$) assumes that the true proportion for Democrats is equal to the sample proportion for Republicans (0.0984148).

呩 Null Hypothesis ($H_0$): The true proportion for Democrats = 0.0984148

呩 Alternative Hypothesis ($H_a$): The true proportion for Democrats ≠ 0.0984148

The standard DEVIATION for the null hypothesis

SD_null_hypot <- sqrt(prop_null_hypothesis * (1- prop_null_hypothesis))

Since the p-value (0.1069) is greater than 0.05, there is not enough evidence to conclude that the true mean is different from 0.0984148 at both the 5% (p-value < 0.05) and 1% (p-value < 0.01) significance levels. Thus, we would fail to reject the null hypothesis, and the observed difference in the mean is not statistically significant.

## Answer to (2.c)

```
# Question 2.c: Using the REPUBLICANS subsample, compute the p-value with the
null hypothesis
# that the true proportion differs from the estimated sample proportion for
Democrats.

prop_null_hypothesis <- mean(df_dems$notsafe_binary)
n_sample_size <- nrow(df_rep) # UPDATED CODE LINE

# Calculate the standard DEVIATION for the null hypothesis (replace with the
correct formula)
SD_null_hypot <- sqrt(prop_null_hypothesis * (1- prop_null_hypothesis))

# Run the z-test to test if the mean is significantly different from the
sample proportion for Democrats
z_test_result <- z.test(
  x = df_rep$notsafe_binary, # Here we declare the variable for our sample
  mu = prop_null_hypothesis,  # Hypothesized population mean
  sigma.x = SD_null_hypot  # Population standard DEVIATION (assuming known)
)

# Print the z-test result
print(z_test_result)

##
##  One-sample z-Test
##
## data:  df_rep$notsafe_binary
## z = -1.4109, p-value = 0.1583
## alternative hypothesis: true mean is not equal to 0.1097493
## 95 percent confidence interval:
##  0.08266981 0.11415978
## sample estimates:
## mean of x
## 0.0984148
```

**Explanation/Justification**: As in (2.b), we can conduct a similar one-sample z-test but for the Republican subsample, testing against the null hypothesis that the Republican proportion of respondents who feel unsafe does not differ from the estimated proportion for Democrats (0.1097493). The null hypothesis ($H_0$) posits that the true proportion for Republicans is equal to the sample proportion for Democrats.

呩 Null Hypothesis ($H_0$): The true proportion for Republicans = 0.1097493

呩 Alternative Hypothesis ($H_a$): The true proportion for Republicans ≠ 0.1097493

The standard DEVIATION for the null hypothesis

SD_null_hypot <- sqrt(prop_null_hypothesis * (1- prop_null_hypothesis))

Since the p-value (0.1583) is greater than 0.05, there is not enough evidence to conclude that the true mean is different from 0.0984148 at both the 5% (p-value < 0.05) and 1% (p-value < 0.01) significance levels. Thus, we would fail to reject the null hypothesis, and the observed difference in the mean is not statistically significant.

## Question 3: Hypothesis Test for Proportions at 1% Significance Level

**Don’t modify this chunk:**

```
# Set seed for reproducibility
set.seed(12345)

# Create a mock dataset with 35 respondents where each respondent either
supports (1) or does not support (0) a specific policy
respondents <- 35
true_prop <- 0.44
mock_data_q3 <- data.frame(support_policy = rbinom(n=respondents, size=1,
true_prop)) # size=1 means one trial, hence we are drawing 'n' bernoulli
random variables.
```

```
# Visualize the first few rows of the dataset
head(mock_data_q3)
```

```
##   support_policy
## 1              1
## 2              1
## 3              1
## 4              1
## 5              0
## 6              0
```

```
mean(mock_data_q3$support_policy)
```

```
## [1] 0.4
```

## Question 3.a: Sample Proportion and Standard Error

```
# Calculate the sample proportion
p_hat <- mean(mock_data_q3$support_policy) # Observed Sample Proportion

# Define the null hypothesis proportion
p_0 <- 0.43

# Compute the sample size
n <- nrow(mock_data_q3)

# Calculate the standard error under the null
standard_error_null_hypothesis <- sqrt(p_0 * (1 - p_0) / n)

# Print the results
cat("Sample Proportion: ", round(p_hat, 3), "\n")
```

```
## Sample Proportion:  0.4
```

```
cat("Standard Error: ", round(standard_error_null_hypothesis, 3), "\n")
```

```
## Standard Error:  0.084
```

**Explanation/Justification:** The sample proportion (p_hat) is calculated as the mean of the support_policy column, which represents the observed proportion of respondents who support the policy. Here, p_hat = 0.4, meaning 40% of the sample supports the policy.

We can compute the standard error (SE) using the formula, where p_0 = 0.43 is the null hypothesis proportion and n = 35 is the sample size.

Standard Error (SE) = sqrt(p * (1 - p) / n)

= sqrt(p_0 * (1 - p_0) / n) = sqrt(0.43 * (1 - 0.43) / 35) ≈ 0.0838

The standard error quantifies the variability of the sample proportion under the null hypothesis, and is calculated to be approximately 0.084, rounded to three decimal places.

## Question 3.b: Define the Hypotheses (Step 1)

吥 Null Hypothesis ($H_0$): **p = 0.43**

The null hypothesis states that the true population proportion of support for the policy is equal to 0.43.

吥 Alternative Hypothesis ($H_a$): **p ≠ 0.43**

Since this is a two-tailed test, the alternative hypothesis posits that the true population proportion is different from 0.43.

## Question 3.c: Compute the Test Statistic (Step 2)

```
# Define the null hypothesis proportion
p_0 <- 0.43

# Define sample proportion
p_hat <- mean(mock_data_q3$support_policy)

# Define Standard Error Under the Null
SE_H0 <- sqrt(p_0 * (1 - p_0) / n)

# Calculate the z-test statistic
z_statistic <- (p_hat - p_0) / SE_H0

# Print the test statistic
cat("Z-test Statistic: ", round(z_statistic, 3), "\n")

## Z-test Statistic:  -0.358
```

**Explanation/Justification**: The test statistic for this hypothesis test is calculated using a Z-test formula, where p_hat = 0.4 is the sample proportion, p_0 = 0.43 is the null hypothesis proportion, and SE_H0 = 0.084 is the standard error under the null hypothesis.

Z = (p_hat - p_0) / SE_H0 = (0.4 - 0.43) / 0.084 ≈ -0.3571

Therefore, Z-test Statistic is -0.358, rounded to three decimal places. This Z-test Statistic quantifies how far the observed sample proportion is from the hypothesized proportion in units of the standard error. A negative Z-test Statistic here indicates that the sample proportion is slightly less than the hypothesized 0.43, but this will be assessed for significance in the next question (3.d).

## Question 3.d: Determine the Critical Value and P-value (Step 3)

```
# Critical value for 1% significance level (two-tailed)
alpha <- 0.01
critical_value <- qnorm( 1 - (alpha / 2) )

# Calculate the z-test statistic
z_statistic <- (p_hat - p_0) / SE_H0

# Calculate the p-value. HINT: Use `pnorm(abs(z_statistic), lower.tail =
FALSE)` to compute Pr(Z > |Z_obs|).
p_value <- 2 * (1 - pnorm( abs(z_statistic), lower.tail = FALSE))

# Print the critical value and p-value
cat("Critical Value: ", round(critical_value, 3), "\n")
```

```
## Critical Value:  2.576
```

```
cat("P-value: ", round(p_value, 3), "\n")
```

```
## P-value:  1.28
```

**Explanation/Justification:** At a 1% significance level (α = 0.01), the critical value for a two-tailed test is calculated using the ‘qnorm’ function. The critical value represents the Z-test Statistic threshold beyond which we would reject the null hypothesis. For α = 0.01, we can find the critical value, 2.576.

Next, we can compute the p-value as the probability of obtaining a test statistic as extreme as Z = −0.358 under the null hypothesis. For a two-tailed test,

P-value = 2 × P(Z > |z|) = 2 × (1 − pnorm( I Z I )) = 2 × (1 − pnorm(0.358)) ≈ 1.28

At α = 0.01, since P-value = 1.28 > 0.01, we fail to reject $H_0$.

Since the p-value (1.28) is much greater than the significance level (α = 0.01), it suggests that the observed sample proportion is not significantly different from the hypothesized proportion of 0.43.

## Question 3.e: Conclusion (Step 4)

```
# Print the z-statistic, critical value and p-value
cat("Z-test Statistic: ", round(z_statistic, 3), "\n")

## Z-test Statistic:  -0.358

cat("Critical Value: ", round(critical_value, 3), "\n")

## Critical Value:  2.576

cat("P-value: ", round(p_value, 3), "\n")

## P-value:  1.28
```

**Explanation/Justification:** We can draw a conclusion using the test statistic (Z-test Statistic = -0.358), critical value (2.576), and p-value (1.28). Since the absolute value of the test statistic (0.358) is less than the critical value (2.576), and the p-value (1.28) is greater than 0.01 (α = 0.01), we **fail to reject the null hypothesis**. This means there is insufficient evidence to conclude that the true population proportion is different from 0.43 at the 1% significance level. We maintain the hypothesis that the population proportion is 0.43 because the observed sample proportion of 0.4 is likely to occur under the null hypothesis.

### (Optional) Verify your previous result running the following chunk:

```
prop_null_hypothesis <- 0.43
n_sample_size <- respondents

# Calculate the standard DEVIATION for the null hypothesis (replace with the
correct formula)
SD_null_hypot <- sqrt( prop_null_hypothesis*(1-prop_null_hypothesis) )    #
The formula for SD does not include the sample size

# Run the z-test to test if the mean is significantly different from the
sample proportion for Democrats
z_test_result <- z.test(
  x = mock_data_q3$support_policy,
  mu = prop_null_hypothesis,  # Hypothesized population mean
  sigma.x = SD_null_hypot  # Population standard DEVIATION (assuming known)
)

# Print results
print(z_test_result)

##
##  One-sample z-Test
##
```

```
## data:  mock_data_q3$support_policy
## z = -0.3585, p-value = 0.72
## alternative hypothesis: true mean is not equal to 0.43
## 95 percent confidence interval:
##  0.2359842 0.5640158
## sample estimates:
## mean of x
##       0.4
```

```r
# Print p-value
print(z_test_result$p.value)
```

```
## [1] 0.7199726
```