PS_ Probability and Hypothesis Testing for TAs

Pablo E. Gutiérrez-Fonseca

2023-08-12

R practice.

Install packages.

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

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Load the water pollution data into R.

```
##    Species           Mammalian.Size.group     BodyWt            BrainWt
##  Length:62          Length:62              Min.   :   0.005   Min.   :   0.14
##  Class :character   Class :character       1st Qu.:   0.600   1st Qu.:   4.25
##  Mode  :character   Mode  :character       Median :   3.342   Median :  17.25
##                                            Mean   : 198.790   Mean   : 283.13
##                                            3rd Qu.:  48.202   3rd Qu.: 166.00
##                                            Max.   :6654.000   Max.   :5712.00
##
##   NonDreaming        Dreaming       TotalSleep       LifeSpan
##  Min.   : 2.100   Min.   :0.000   Min.   : 2.60   Min.   :  2.000
##  1st Qu.: 6.100   1st Qu.:0.900   1st Qu.: 8.05   1st Qu.:  6.625
##  Median : 8.300   Median :1.800   Median :10.45   Median : 15.100
##  Mean   : 8.541   Mean   :1.941   Mean   :10.53   Mean   : 19.878
##  3rd Qu.:11.000   3rd Qu.:2.500   3rd Qu.:13.20   3rd Qu.: 27.750
##  Max.   :17.900   Max.   :6.600   Max.   :19.90   Max.   :100.000
##  NA's   :13       NA's   :11      NA's   :4       NA's   :4
##    Gestation         Predation        Exposure         Danger
##  Min.   : 12.00   Min.   :1.000   Min.   :1.000   Min.   :1.000
```

```
##  1st Qu.: 35.75   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000
##  Median : 79.00   Median :3.000   Median :2.000   Median :2.000
##  Mean   :142.35   Mean   :2.871   Mean   :2.419   Mean   :2.613
##  3rd Qu.:207.50   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :645.00   Max.   :5.000   Max.   :5.000   Max.   :5.000
##  NA's   :4
```

1. Based on the summary results in JMP, mammals spend most of their time in which type of sleep phase (dreaming or non-dreaming)?

```
mean(df_mammals$NonDreaming, na.rm = TRUE)
```

```
## [1] 8.540816
```

```
mean(df_mammals$Dreaming, na.rm = TRUE)
```

```
## [1] 1.941176
```

2. Which type of sleep phase has the highest variability across the species included here (dreaming or non-dreaming)?
   Non-dreaming

```
sd(df_mammals$NonDreaming, na.rm = TRUE)
```

```
## [1] 3.744046
```

```
sd(df_mammals$Dreaming, na.rm = TRUE)
```

```
## [1] 1.445016
```

3. Enter the p-value for the goodness of fit test for the **Dreaming variable** (NOTE….just enter the number (no letters, symbols, equal signs etc)… also be careful of decimal places.

```
shapiro.test(df_mammals$Dreaming)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_mammals$Dreaming
## W = 0.87556, p-value = 7.067e-05
```

4. Based on this goodness of fit test, is the **dreaming variable** normally distributed? **No**

5. Enter the new p-value for the goodness of fit test for the **Dreaming variable** when the outliers are excluded (NOTE….just enter the number (no letters, symbols, equal signs etc)… also be careful of decimal places.

```
quartiles <- quantile(na.omit(df_mammals$Dreaming), probs = c(0.25, 0.75))
IQR <- IQR(na.omit(df_mammals$Dreaming))
Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

new_Dreaming <- subset(df_mammals$Dreaming, df_mammals$Dreaming > Lower & df_mammals$Dreaming < Upper)

shapiro.test(new_Dreaming)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  new_Dreaming
## W = 0.95543, p-value = 0.06599
```

6. Based on this goodness of fit test, is the dreaming variable normally distributed once outliers are removed? **Yes.**

7. Enter the new p-values for the goodness of fit test for the **log transformed Dreaming variable**.

```r
log_Dreaming <- log(df_mammals$Dreaming[df_mammals$Dreaming > 0])
shapiro.test(log_Dreaming)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log_Dreaming
## W = 0.98568, p-value = 0.8003
```

8. Is this **log transformed Dreaming variable** normally distributed?
Yes

9. Based on these results, would you transform (outlier transform, log transform or no transform) your **dreaming variable**? Assume that outliers removed are incorrect data values and that all values fit the range necessary for a log transform) **Outlier transform**

10. Now go back and repeat these procedures for the **Non-dreaming variable**. However, instead of entering answers for every one of these steps, just summarize what you would do with this variable: Based on these results, would you transform (outlier transform, log transform or no transform) your non-dreaming variable? Assume that outliers removed are incorrect data values)

```r
shapiro.test(df_mammals$NonDreaming)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_mammals$NonDreaming
## W = 0.98257, p-value = 0.6763
```

no transform

11. Is the distribution of the Non dreaming variable normally distributed for the small mammalian size group? **Yes**

From the df_mammals make two groups (large and small mammals)

```
df_mammals_large <- df_mammals %>%
  filter(Mammalian.Size.group == 'large')

mean_NonDreaming_large <- mean(df_mammals_large$NonDreaming, na.rm = TRUE)
sd_NonDreaming_large <- sd(df_mammals_large$NonDreaming, na.rm = TRUE)
```

```
df_mammals_small <- df_mammals %>%
  filter(Mammalian.Size.group == 'small')

# Calculate mean and standard deviation of the NonDreaming column
mean_NonDreaming_small <- mean(df_mammals_small$NonDreaming, na.rm = TRUE)
sd_NonDreaming_small <- sd(df_mammals_small$NonDreaming, na.rm = TRUE)
```

12. Is the distribution of the Non dreaming variable normally distributed for the large mammalian size group? Normality test Large Mammals

```
shapiro.test(df_mammals_large$NonDreaming)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_mammals_large$NonDreaming
## W = 0.88465, p-value = 0.1191
```

13. Calculate the z score for the **Baboon** based on its non dreaming measured value for the small class (i.e. use the mean and std from the small class to calculate this z score)

```
mean_NonDreaming_Baboon <- df_mammals$NonDreaming[df_mammals$Species == "Baboon"]

# Calculate Z-scores for the NonDreaming column
z_scores <- (mean_NonDreaming_Baboon - mean_NonDreaming_small) / sd_NonDreaming_small

# Print the Z-scores
print(z_scores)
```

```
## [1] -0.2011717
```

14. Calculate the z score for the **Patas monkey** based on its non dreaming measured value for the small class (i.e. use the mean and std from the small class to calculate this z score)

```r
mean_NonDreaming_Patas_monkey <- df_mammals$NonDreaming[df_mammals$Species == "Patas monkey"]

# Calculate Z-scores for the NonDreaming column
z_scores <- (mean_NonDreaming_Patas_monkey - mean_NonDreaming_small) / sd_NonDreaming_small

# Print the Z-scores
print(z_scores)
```

```
## [1] 0.07001131
```

15. Calculate the z score for the **Rhesus monkey** based on its non dreaming measured value for the small class (i.e. use the mean and std from the small class to calculate this z score)

```r
mean_NonDreaming_Rhesus_monkey <- df_mammals$NonDreaming[df_mammals$Species == "Rhesus monkey"]

# Calculate Z-scores for the NonDreaming column
z_scores <- (mean_NonDreaming_Rhesus_monkey - mean_NonDreaming_small) / sd_NonDreaming_small

# Print the Z-scores
print(z_scores)
```

```
## [1] -0.4120919
```

16. Calculate the z score for the **Roe deer** based on its non dreaming measured value for the small class (i.e. use the mean and std from the small class to calculate this z score)

```r
mean_NonDreaming_Roe_deer <- df_mammals$NonDreaming[df_mammals$Species == "Roe deer"]

# Calculate Z-scores for the NonDreaming column
z_scores <- (mean_NonDreaming_Roe_deer - mean_NonDreaming_small) / sd_NonDreaming_small

# Print the Z-scores
print(z_scores)
```

```
## [1] -2.280242
```

17. Based on your knowledge that higher (absolute value) z-scores are more unusual for a given group's distribution, which class would you assign **Baboon** to (small or large)? (hint compare **Baboon's z scores** for the small and large group. . . . . . which group does the species z score fit "best" with. . . . .i.e. for which group is that species z score less extreme?

```r
# Calculate Z-scores for the NonDreaming column
z_scores <- (mean_NonDreaming_Baboon - mean_NonDreaming_small) / sd_NonDreaming_small
# Print the Z-scores
print(z_scores)
```

```
## [1] -0.2011717
```

```r
# Calculate Z-scores for the NonDreaming column
z_scores <- (mean_NonDreaming_Baboon - mean_NonDreaming_large) / sd_NonDreaming_large
# Print the Z-scores
print(z_scores)
```

```
## [1] 1.319504
```

small

18. Based on your knowledge that higher (absolute value) z-scores are more unusual for a given group's distribution, which class would you assign **Pata monkey** to (small or large)? (hint compare **Pata monkey's z scores** for the small and large group...... which group does the species z score fit "best" with.....i.e. for which group is that species z score less extreme?

```r
# Calculate Z-scores for the NonDreaming column
z_scores <- (mean_NonDreaming_Patas_monkey - mean_NonDreaming_small) / sd_NonDreaming_small
# Print the Z-scores
print(z_scores)
```

```
## [1] 0.07001131
```

```r
z_scores <- (mean_NonDreaming_Patas_monkey - mean_NonDreaming_large) / sd_NonDreaming_large
# Print the Z-scores
print(z_scores)
```

```
## [1] 1.620497
```

small

19. Based on your knowledge that higher (absolute value) z-scores are more unusual for a given group's distribution, which class would you assign **Rhesus monkey** to (small or large)? (hint compare **Rhesus monkey's z scores** for the small and large group...... which group does the species z score fit "best" with.....i.e. for which group is that species z score less extreme?

```r
# Calculate Z-scores for the NonDreaming column
z_scores <- (mean_NonDreaming_Rhesus_monkey - mean_NonDreaming_small) / sd_NonDreaming_small
# Print the Z-scores
print(z_scores)
```

```
## [1] -0.4120919
```

```r
z_scores <- (mean_NonDreaming_Rhesus_monkey - mean_NonDreaming_large) / sd_NonDreaming_large
# Print the Z-scores
print(z_scores)
```

```
## [1] 1.085398
```

small

20. Based on your knowledge that higher (absolute value) z-scores are more unusual for a given group's distribution, which class would you assign **Roe deer** to (small or large)? (hint compare **Roe deer's z scores** for the small and large group. . . . . . which group does the species z score fit "best" with. . . . .i.e. for which group is that species z score less extreme?

```
# Calculate Z-scores for the NonDreaming column
z_scores <- (mean_NonDreaming_Roe_deer - mean_NonDreaming_small) / sd_NonDreaming_small
# Print the Z-scores
print(z_scores)
```

```
## [1] -2.280242
```

```
z_scores <- (mean_NonDreaming_Roe_deer - mean_NonDreaming_large) / sd_NonDreaming_large
# Print the Z-scores
print(z_scores)
```

```
## [1] -0.9881076
```

large

19. How many observations in your data set would be labeled outliers by this IQR definition?

```
quartiles <- quantile(na.omit(df_mammals$NonDreaming), probs = c(0.25, 0.75))
IQR <- IQR(na.omit(df_mammals$NonDreaming ))
Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

new_NonDreaming <- subset(df_mammals$NonDreaming , df_mammals$NonDreaming  > Lower & df_mammals$NonDrea

# Calculate the number of outliers removed
num_outliers_removed <- length(na.omit(df_mammals$NonDreaming)) - length(new_NonDreaming)

# Print the number of outliers removed
cat("Number of outliers removed:", num_outliers_removed, "\n")
```

```
## Number of outliers removed: 0
```

20. Armed with your knowledge of what a z score is and how to calculate actual values from given z scores, enter the **upper non-dreaming** value beyond which I may have outliers (values greater than 2 standard deviations from the mean in the positive direction). It may help if you sketch this out so that you can visualize what you are calculating.

```r
# Calculate mean and standard deviation of NonDreaming column
mean_NonDreaming <- mean(df_mammals$NonDreaming, na.rm = TRUE)
sd_NonDreaming <- sd(df_mammals$NonDreaming, na.rm = TRUE)

# Calculate upper bound using z-score (z = 2)
z_score_threshold <- 2
upper_bound <- mean_NonDreaming + (z_score_threshold * sd_NonDreaming)

# Print the upper bound
cat("Upper bound for potential outliers:", upper_bound, "\n")
```

```
## Upper bound for potential outliers: 16.02891
```

21. Do the same for the **lower non-dreaming value** below which I might consider observations to be outliers based on the fact that they are more than 2 standard deviations under the mean.

```r
# Calculate lower bound using z-score (z = 2)
lower_bound <- mean_NonDreaming - (z_score_threshold * sd_NonDreaming)

# Print the lower bound
cat("Lower bound for potential outliers:", lower_bound, "\n")
```

```
## Lower bound for potential outliers: 1.052724
```

22. Using this 2-standard deviation threshold, how many potential outliers might be in this **non dreaming** data set?

```r
# Calculate the number of potential outliers above the upper bound
num_upper_outliers <- sum(df_mammals$NonDreaming > upper_bound, na.rm = TRUE)

# Calculate the number of potential outliers below the lower bound
num_lower_outliers <- sum(df_mammals$NonDreaming < lower_bound, na.rm = TRUE)

# Calculate the total number of potential outliers
total_num_outliers <- num_upper_outliers + num_lower_outliers

# Print the results
cat("Number of potential outliers above upper bound:", num_upper_outliers, "\n")
```

```
## Number of potential outliers above upper bound: 1
```

```r
cat("Number of potential outliers below lower bound:", num_lower_outliers, "\n")
```

```
## Number of potential outliers below lower bound: 0
```

```r
cat("Total number of potential outliers:", total_num_outliers, "\n")
```

```
## Total number of potential outliers: 1
```

30 and 31. Identify the **upper and lower z-scores** that cumulatively represent < 2% TOTAL chance of occurring. This doesn't require any calculations, just the determination of two z-scores (upper and lower) from the normal probability table (or excel function).

```r
# Find the z-score for the upper 1% cumulative probability (upper bound)
upper_bound_prob <- 0.99
upper_z_score <- qnorm(upper_bound_prob)

# Find the z-score for the lower 1% cumulative probability (lower bound)
lower_bound_prob <- 0.01
lower_z_score <- qnorm(lower_bound_prob)

# Print the results
cat("Upper z-score for < 2% cumulative probability:", upper_z_score, "\n")
```

```
## Upper z-score for < 2% cumulative probability: 2.326348
```

```r
cat("Lower z-score for < 2% cumulative probability:", lower_z_score, "\n")
```

```
## Lower z-score for < 2% cumulative probability: -2.326348
```

32. What is the actual **non-dreaming** value associated with the upper z score identified in the question above.

```r
# Calculate the actual non-dreaming value associated with the upper z-score
actual_upper_value <- mean_NonDreaming + (upper_z_score * sd_NonDreaming)

# Print the result
cat("Actual non-dreaming value associated with the upper z-score:", actual_upper_value, "\n")
```

```
## Actual non-dreaming value associated with the upper z-score: 17.25077
```

33. How many **non-dreaming** observations in your data set would be labeled as outliers by this zscore definition?

```r
# Calculate the number of non-dreaming observations labeled as outliers
num_outliers_zscore <- sum(df_mammals$NonDreaming > upper_bound | df_mammals$NonDreaming < lower_bound,

# Print the result
cat("Number of non-dreaming observations labeled as outliers (z-score definition):", num_outliers_zscore
```

```
## Number of non-dreaming observations labeled as outliers (z-score definition): 1
```