PS_ Descriptive Statistics for TAs

Pablo E. Gutiérrez-Fonseca

2024-02-08

R practice.

Install packages.

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.2

## Warning: package 'ggplot2' was built under R version 4.3.2

## Warning: package 'tibble' was built under R version 4.3.1

## Warning: package 'tidyr' was built under R version 4.3.1

## Warning: package 'readr' was built under R version 4.3.2

## Warning: package 'purrr' was built under R version 4.3.1

## Warning: package 'dplyr' was built under R version 4.3.1

## Warning: package 'stringr' was built under R version 4.3.1

## Warning: package 'forcats' was built under R version 4.3.2

## Warning: package 'lubridate' was built under R version 4.3.2

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(lessR)
```

```
## Warning: package 'lessR' was built under R version 4.3.2
```

```
##
## lessR 4.3.0                         feedback: gerbing@pdx.edu
## --------------------------------------------------------------
## > d <- Read("")   Read text, Excel, SPSS, SAS, or R data file
##   d is default data frame, data= in analysis routines optional
##
## Learn about reading, writing, and manipulating data, graphics,
## testing means and proportions, regression, factor analysis,
## customization, and descriptive statistics from pivot tables
##   Enter:  browseVignettes("lessR")
##
## View changes in this and recent versions of lessR
##   Enter: news(package="lessR")
##
## Interactive data analysis
##   Enter: interact()
##
##
## Attaching package: 'lessR'
##
## The following objects are masked from 'package:dplyr':
##
##     recode, rename
```

```r
library(DT)
library(e1071)
```

```
##
## Attaching package: 'e1071'
##
## The following object is masked from 'package:lessR':
##
##     kurtosis
```

Load the water pollution data into R.

```
##      Type              Density          Strength
##  Length:20          Min.   :0.8700   Min.   :390.0
##  Class :character   1st Qu.:0.9375   1st Qu.:482.5
##  Mode  :character   Median :0.9800   Median :555.0
##                     Mean   :0.9740   Mean   :541.5
##                     3rd Qu.:1.0200   3rd Qu.:602.5
##                     Max.   :1.0400   Max.   :650.0
```

1. the **MEAN Strength**.

```r
round(mean(df_strenth$Strength),2)
```

```
## [1] 541.5
```

1.1. the **MEDIAN Strength**.

```r
round(median(df_strenth$Strength),2)
```

```
## [1] 555
```

2. Now calculate the **STANDARD DEVIATION** for the Strength.

```r
round(sd(df_strenth$Strength),2)
```

```
## [1] 73.86
```

2.1

```r
round(var(df_strenth$Strength),2)
```

```
## [1] 5455.53
```

3. What is the **inter-quartile range** for Strength column?

```r
# Calculate the IQR for the specified column
round(iqr_value <- IQR(df_strenth$Strength),2)
```

```
## [1] 120
```

```r
# Print the IQR
print(iqr_value)
```

```
## [1] 120
```

4. Using the Interquartile range technique, how many **OUTLIER** years are there in your **Strenght** column?

```r
# Calculate the quartiles and IQR
q1 <- quantile(df_strenth$Strength, 0.25, na.rm = TRUE)
q3 <- quantile(df_strenth$Strength, 0.75, na.rm = TRUE)
iqr <- q3 - q1

# Define the lower and upper bounds for outliers
lower_bound <- q1 - 1.5 * iqr
upper_bound <- q3 + 1.5 * iqr

# Identify outlier years
outlier_years <- df_strenth$Strength < lower_bound | df_strenth$Strength > upper_bound

# Count the number of outlier years
num_outliers <- sum(outlier_years)

# Print the number of outlier years
print(num_outliers)
```

```
## [1] 0
```

5. Now, for the **Density** column, calculate the **skewness, standard error of skewness (SES), kurtosis, standard error of kurtosis (SEK)**, and perform the **normality test with Shapiro-Wilk test**. Based on all of these parameters, determine if the Density is normally distributed.

```r
# Calculate Pearson's skew
skew <- skew(df_strenth$Density)
# Print the skew rounded to 2 decimal places
print(round(skew, 2))
```

```
## [1] -0.48
```

5.1. Calculate the **standard error of skew (ses)** for this data so we can determine the significance of our skew.

```r
# Calculate the skewness and standard error of skew
ses <- round(sqrt(6/length(df_strenth$Density)),2)
ses
```

```
## [1] 0.55
```

5.2. Based on the ses technique, is this **skew significant**?
No

5.3. Now determine the **kurtosis for the Yearly Mean Snow Depth data**.

```r
kurtosis <- round(kurtosis(df_strenth$Density, type = 1) - 3, 2)
kurtosis
```

```
## [1] -3.91
```

5.4. Calculate the **standard error of kurtosis (sek)** for this data so we can determine the significance of our skew.

```
# Calculate standard error of kurtosis (SES)
ses_kurt <- round(sqrt(24/length(df_strenth$Density)),2)
ses_kurt
```

```
## [1] 1.1
```

5.6. Based on the sek technique, is this **kurtosis significant**? **No**

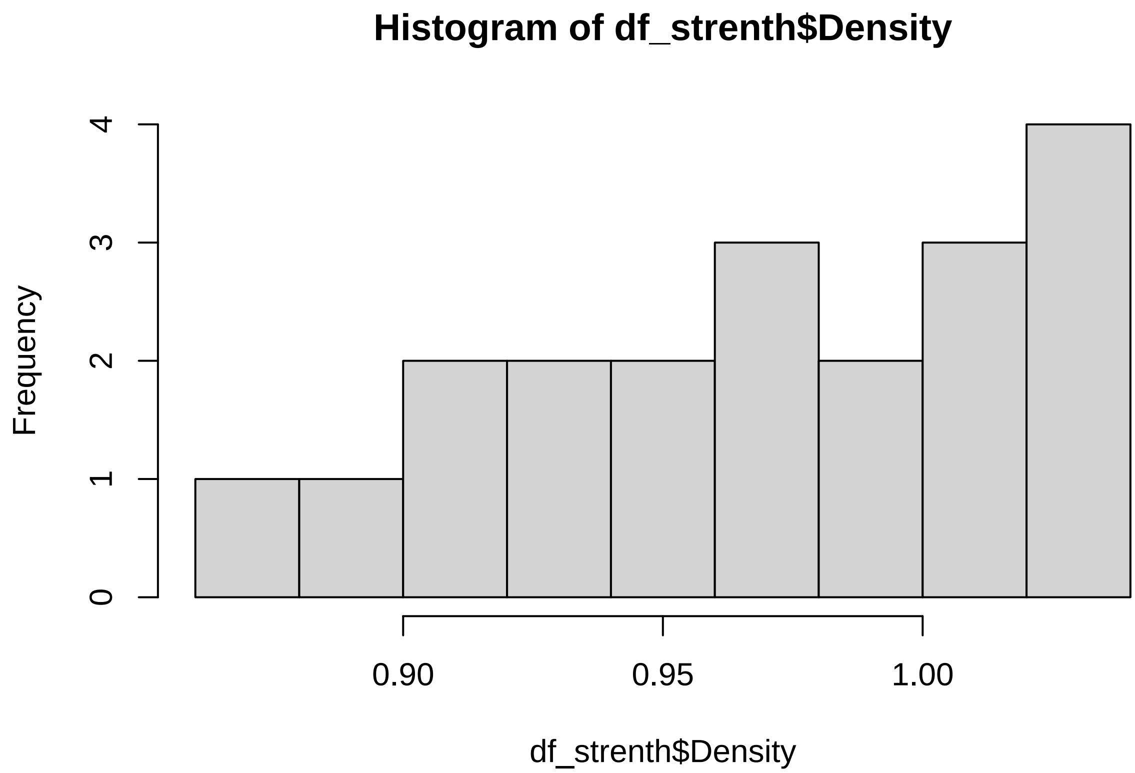
5.7. Based on all of these parameters, determine if the Yearly Mean Snow Depth is normally distributed. Pay special attention to the shape (distribution) of the data (plotted on a histogram). **No**

5.8. Is your data normally distributed? How can you tell? Be sure to report your certainty of this conclusion.

```
shapiro.test(df_strenth$Density)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_strenth$Density
## W = 0.93122, p-value = 0.163
```

```
hist(df_strenth$Density, breaks = 10)
```



Histogram of df_strenth$Density
Frequency
0
1
2
3
4
0.90
0.95
1.00
df_strenth$Density