PgDip Unit 7 activities

Guilherme Amorim

2024-10-16

Unit 7 - notes activites (parametric tests)

```
# initial setup


library(haven)
library(skimr)
library(tidyverse)
library(here)
```

```
## Warning: package 'here' was built under R version 4.3.3
```

```
Health_Data<-read_sav(here("Datasets/Health Data.sav"))
```

```
# first install packages (from terminal)

# pip3 install numpy

# pip3 install pandas

# pip3 install matplotlib

# pip3 install pyreadstat

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from scipy import stats


Health_Data_python = pd.read_spss("C:/Users/guilhermep/Documents/PgDip/Coding/Module 2/pgdip_module2_pra
```

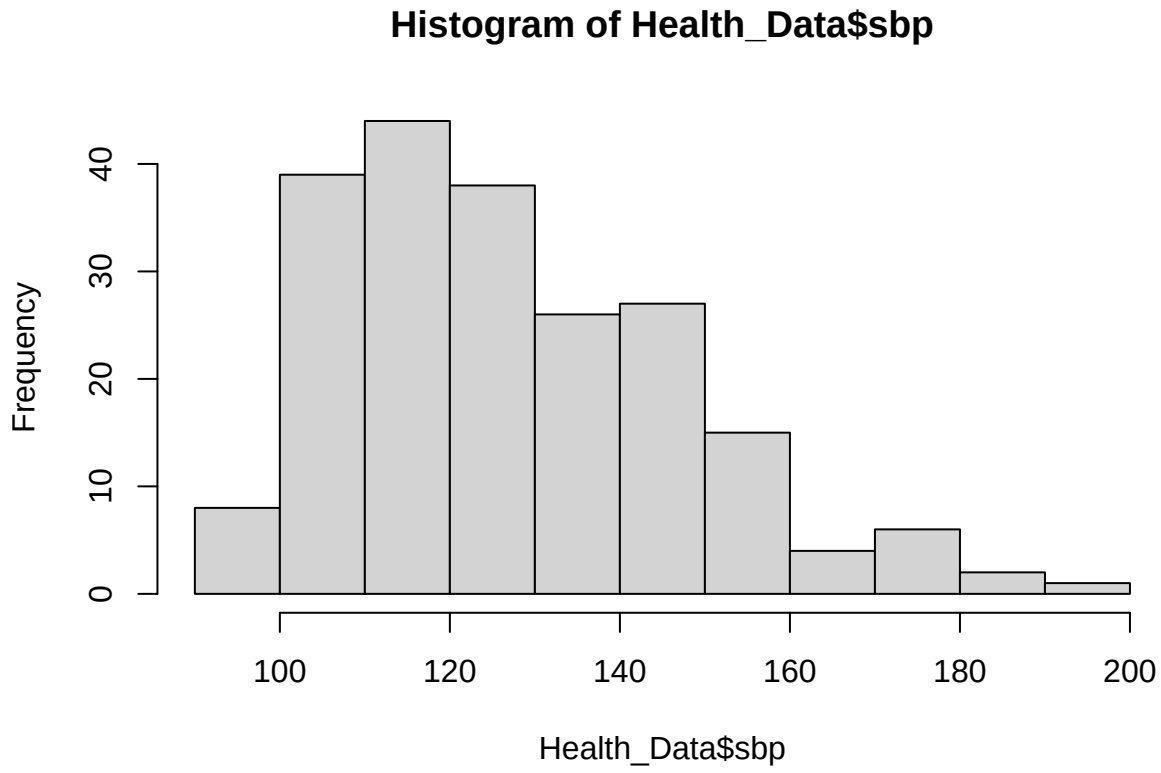
Normality testing (Shapiro-Wilk)

R

```r
shapiro.test(Health_Data$sbp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Health_Data$sbp
## W = 0.95474, p-value = 3.345e-06
```

```r
hist(Health_Data$sbp)
```



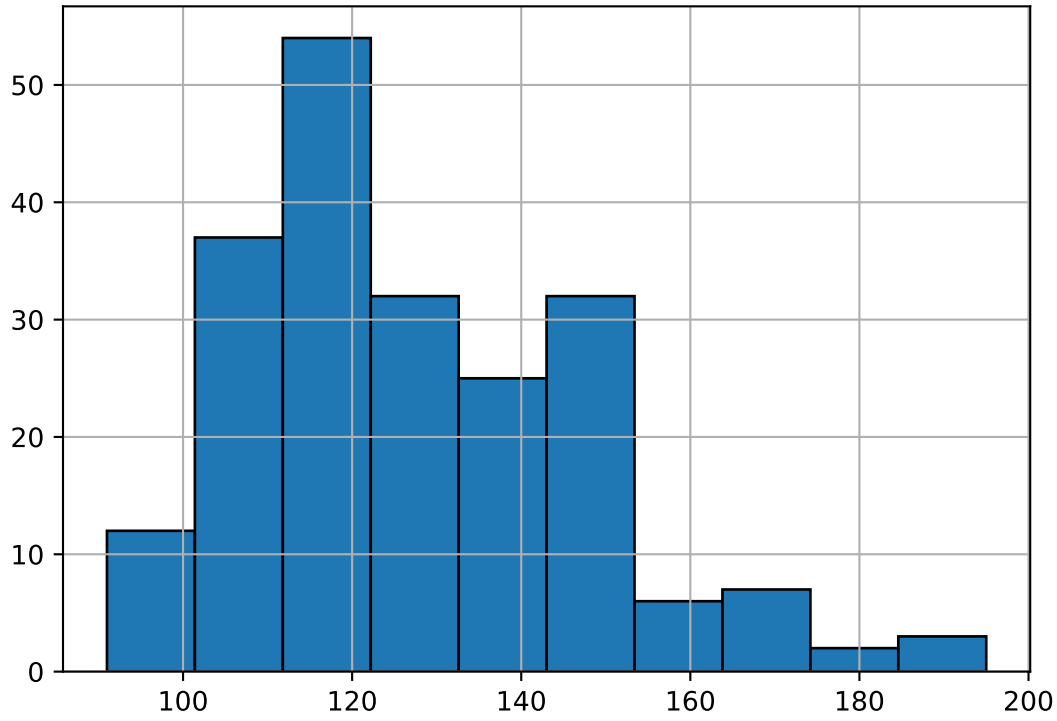
The null hypothesis of the Shapiro-Wilk test is that the data is normally distributed. A p-value of 3.345e-06 (<0.001) indicates that the null hypothesis was rejected at a 5% alpha, suggesting the data is non-normally distributed. This is confirmed by visual inspection of the histogram which shows a right-skewed distribution

Python

```python
stats.shapiro(Health_Data_python["sbp"])
```

```
## ShapiroResult(statistic=0.9547387361526489, pvalue=3.345598997839261e-06)
```

```
Health_Data_python["sbp"].hist(edgecolor="black")
plt.show()
```



Same results with Python code

t-tests

R

```
# One sample t-test
res<-t.test(Health_Data$dbp, mu=80)
print(res)
```

```
##
##  One Sample t-test
##
## data:  Health_Data$dbp
## t = 3.4124, df = 209, p-value = 0.0007732
## alternative hypothesis: true mean is not equal to 80
## 95 percent confidence interval:
##  81.16832 84.36502
## sample estimates:
## mean of x
##  82.76667
```

Mean: 82.77, reference value: 80, p-value <0.001, suggesting that the mean diastolic BP of participants in this sample is statistically significantly different from 80 at a significance level of 0.05

```
# Independent (two-sample) t-test

t.test(age~diabetes,data = Health_Data, var.equal=TRUE, alternative="less")
```

```
##
##  Two Sample t-test
##
## data:  age by diabetes
## t = 1.4146, df = 208, p-value = 0.9207
## alternative hypothesis: true difference in means between group 1 and group 2 is less than 0
## 95 percent confidence interval:
##      -Inf 3.854124
## sample estimates:
## mean in group 1 mean in group 2
##        27.91111        26.13333
```

p-value 0.92, suggesting no statistically significant difference in age between diabetics and non-diabetics

```
# Paired (two-sample) t-test

res<-t.test(Health_Data$pre_test,Health_Data$post_test,paired = TRUE)
res
```

```
##
##  Paired t-test
##
## data:  Health_Data$pre_test and Health_Data$post_test
## t = -15.092, df = 31, p-value = 7.84e-16
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -42.46114 -32.35136
## sample estimates:
## mean difference
##       -37.40625
```

p-value <0.001, showing a statistically significant difference in in mean scores before and after the training

Python

```
# One sample t-test
stats.ttest_1samp(Health_Data_python["dbp"], 80)
```

```
## TtestResult(statistic=3.4123606697811057, pvalue=0.000773217288977576, df=209)
```

Same result with Python

```
# Independent (two-sample) t-test

stats.ttest_ind(Health_Data_python["age"][Health_Data_python["diabetes"]=="Yes"],Health_Data_python["ag
```

```
## TtestResult(statistic=1.4146321247846514, pvalue=0.15867161234642121, df=208.0)
```

```
# Paired (two-sample) t-test
stats.ttest_rel(Health_Data_python["pre_test"].dropna(),Health_Data_python["post_test"].dropna())
```

```
## TtestResult(statistic=-15.092417315184948, pvalue=7.8404934223232925e-16, df=31)
```

Same result with Python, but here NAs had to be dropped

Anova

R

```
res.aov<-aov(income~religion_2, data = Health_Data)
summary(res.aov)
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## religion_2    1 1.316e+09 1.316e+09   4.256 0.0404 *
## Residuals   208 6.434e+10 3.093e+08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Health_Data%>%
  group_by(religion_2)%>%
  summarise(mean=mean(income))
```

```
## # A tibble: 4 x 2
##   religion_2      mean
##   <dbl+lbl>      <dbl>
## 1 1 [MUSLIM]    88181.
## 2 2 [HINDU]     79166.
## 3 3 [Christian] 79406.
## 4 4 [BUDDHISM]  84797.
```

The output suggests that there are statistically significant differences in mean income across the 4 religious groups included in the variable religion_2, but does not specify which.

Upon further exploration, we can see that group 1 (MUSLIM) has a mean of 88180, group 2 (HINDU) 79166, group 3 (Christian) 79405, and group 4 (BUDDHISM) 84796

Python

```
stats.f_oneway(Health_Data_python["income"][Health_Data_python["religion_2"]=="MUSLIM"], Health_Data_py
```

```
## F_onewayResult(statistic=3.6419410495096636, pvalue=0.013640822883462755)
```