Unit 8 activities

Guilherme Amorim

2024-10-16

```
# 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
```

Unit 8 notes - non-parametric tests

Mann-Whitney U (2 independent samples)

R

Using Health data, we want to know whether the median systolic BP of males and females is same.

```
wilcox.test(sbp~sex_1,Health_Data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  sbp by sex_1
## W = 5705.5, p-value = 0.1682
## alternative hypothesis: true location shift is not equal to 0
```

p-value 0.1682, therefore the null hypothesis cannot be rejected and we conclude that there is no statistically significant difference in median systolic BP between males and females.

Python

```
stats.mannwhitneyu(Health_Data_python.sbp[Health_Data_python.sex=="Male"],Health_Data_python.sbp[Health_
```

```
## MannwhitneyuResult(statistic=4535.5, pvalue=0.16823128740168025)
```

Wilcoxon Signed Rank (2 paired samples)

"The null hypothesis focuses on that there is no change in the post-test scores compared to the pre-test score due to the training."

R

```
wilcox.test(Health_Data$pre_test, Health_Data$post_test, exact = FALSE, paired = TRUE)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  Health_Data$pre_test and Health_Data$post_test
## V = 0, p-value = 8.288e-07
## alternative hypothesis: true location shift is not equal to 0
```

p-value <0.001, therefore we reject the null hypothesis and conclude that there is a statistically significant change in post-test scores compared to the pre-test score due to the training.

Python

```
stats.wilcoxon(Health_Data_python.pre_test.dropna(), Health_Data_python.post_test.dropna())
```

```
## WilcoxonResult(statistic=0.0, pvalue=4.656612873077393e-10)
```

Kruskal-Wallis test (>2 paired or unpaired samples)

"The null hypothesis is that the systolic BP has the same central tendency across the religious groups. (There is no difference in rank sums)."

R

```
kruskal.test(sbp~religion, data = Health_Data)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  sbp by religion
## Kruskal-Wallis chi-squared = 0.054144, df = 2, p-value = 0.9733
```

p-value >0.05, therefore we fail to reject the null hypothesis and conclude that there is no statistically significant difference in systolic BP across religious groups.

Python

```
stats.kruskal(Health_Data_python["sbp"][Health_Data_python["religion_2"]=="MUSLIM"], Health_Data_python
```

```
## KruskalResult(statistic=0.07881706485516941, pvalue=0.9942521768125786)
```

Unit 8 - data activity

1. Compute mean, median and mode of variables sbp, dbp and income
2. Find out the five-figure summary of income variable and present it using a Boxplot
3. Run a suitable hypothesis test to see if there is any association between systolic blood pressure and presence and absence of peptic ulcer.

R

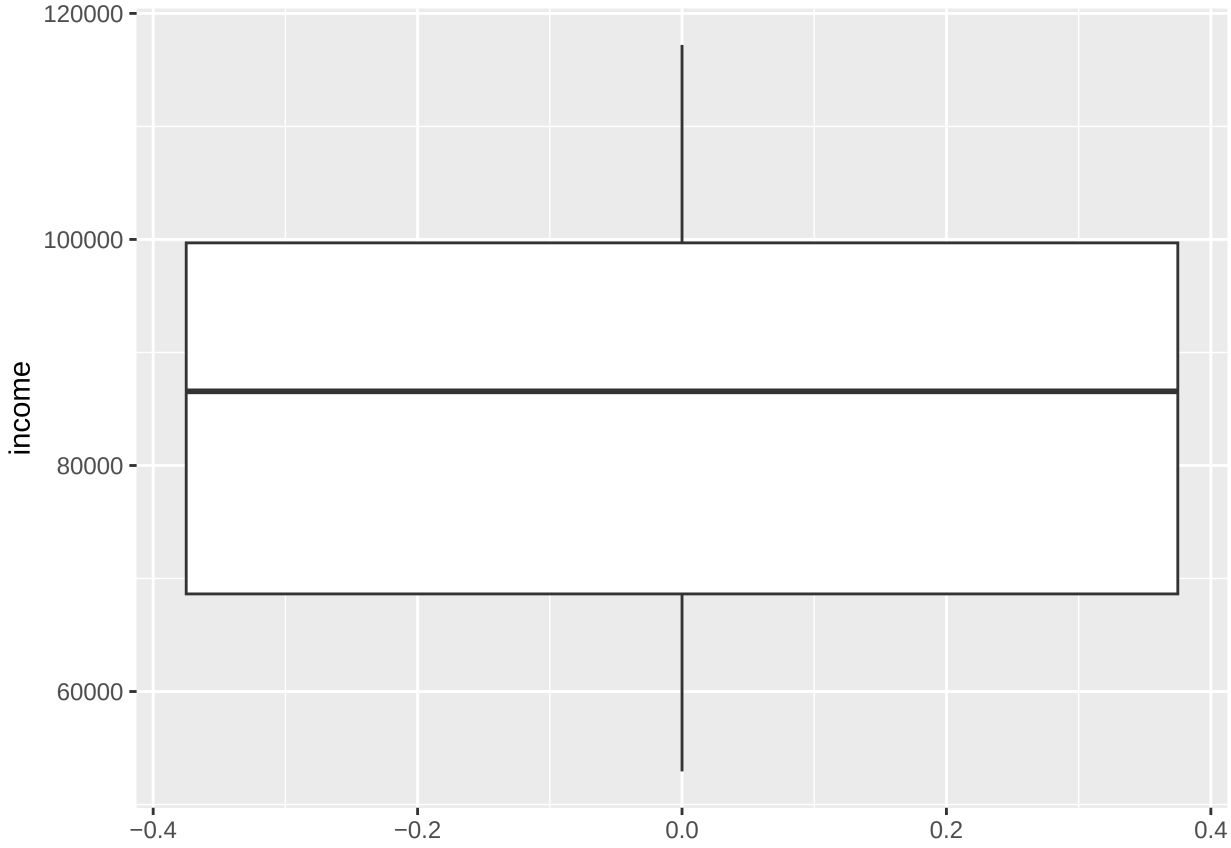
```
## Compute mean, median and mode of variables sbp, dbp and income

getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

Health_Data%>%
  select(sbp, dbp, income)%>%
  pivot_longer(c(sbp, dbp, income), values_to = "value", names_to = "variable")%>%
  group_by(variable)%>%
  summarise(mean=mean(value),
            median=median(value),
            mode=getmode(value))
```

```
## # A tibble: 3 x 4
##   variable    mean median  mode
##   <chr>      <dbl>  <dbl> <dbl>
## 1 dbp         82.8     82    80
## 2 income   85194.  86560. 79774
## 3 sbp        128.     123   120
```

```
## Find out the five-figure summary of income variable and present it using a Boxplot

Health_Data%>%
  select(income)%>%
  summarise(min=min(income),
            Q1=quantile(income, 0.25),
            median=median(income),
            Q3=quantile(income, 0.75),
            max=max(income))
```

```
## # A tibble: 1 x 5
##     min     Q1 median     Q3    max
##   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 52933 68636. 86560. 99696. 117210
```

```
Health_Data%>%
  ggplot(aes(income))+
  geom_boxplot()+
  coord_flip()
```



```
## Run a suitable hypothesis test to see if there is any association between systolic blood pressure an

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

```
##
##  Two Sample t-test
##
## data:  sbp by pepticulcer
## t = 1.2772, df = 208, p-value = 0.8985
## alternative hypothesis: true difference in means between group 1 and group 2 is less than 0
## 95 percent confidence interval:
##      -Inf 10.21382
## sample estimates:
## mean in group 1 mean in group 2
##        131.3171        126.8639
```

p-value 0.8985, suggesting no statistically significant difference in systolic blood pressure between people with and without peptic ulcer

Python

```
## Compute mean, median and mode of variables sbp, dbp and income
"SBP:"
```

```
## 'SBP:'
```

```
"Mean: "+ str(Health_Data_python["sbp"].mean())
```

```
## 'Mean: 127.73333333333333'
```

```
"Median: " +  str(Health_Data_python["sbp"].median())
```

```
## 'Median: 123.0'
```

```
"Mode: " + str(stats.mode(Health_Data_python["sbp"]).mode)
```

```
## 'Mode: 120.0'
```

```
"DBP:"
```

```
## 'DBP:'
```

```
"Mean: "+ str(Health_Data_python["dbp"].mean())
```

```
## 'Mean: 82.76666666666667'
```

```
"Median: " +  str(Health_Data_python["dbp"].median())
```

```
## 'Median: 82.0'
```

```
"Mode: " + str(stats.mode(Health_Data_python["dbp"]).mode)
```

```
## 'Mode: 74.0'
```

```
"Income:"
```

```
## 'Income:'
```

```
"Mean: "+ str(Health_Data_python["income"].mean())
```

```
## 'Mean: 85194.4857142857'
```

```
"Median: " +  str(Health_Data_python["income"].median())
```

```
## 'Median: 86560.5'
```

```
"Mode: " + str(stats.mode(Health_Data_python["income"]).mode)
```

```
## 'Mode: 52933.0'
```

```
## Find out the five-figure summary of income variable and present it using a Boxplot
print("Min:", Health_Data_python["income"].min())
```

```
## Min: 52933.0
```

```
print("Q1:", np.percentile(Health_Data_python["income"],0.25))
```

```
## Q1: 53195.295
```

```
print("Median:", Health_Data_python["income"].median())
```

```
## Median: 86560.5
```

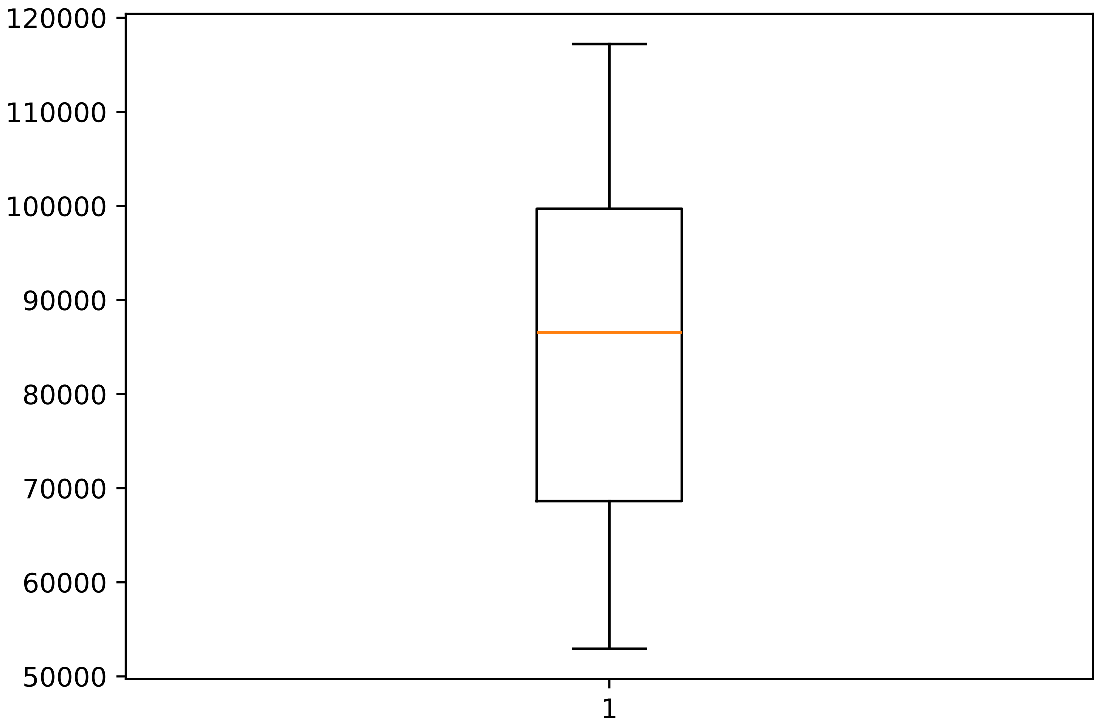
```
print("Q3:", np.percentile(Health_Data_python["income"],0.75))
```

```
## Q3: 53742.0175
```

```
print("Max:", Health_Data_python["income"].max())
```

```
## Max: 117210.0
```

```
fig = plt.boxplot(Health_Data_python["income"])
plt.show()
```



```
## Run a suitable hypothesis test to see if there is any association between systolic blood pressure an

stats.ttest_ind(Health_Data_python["sbp"][Health_Data_python["pepticulcer"]=="Yes"],Health_Data_python[
```

```
## TtestResult(statistic=1.2772129020220484, pvalue=0.2029509444350035, df=208.0)
```

Unit 8 - scenario-based exercise (95% confidence interval)

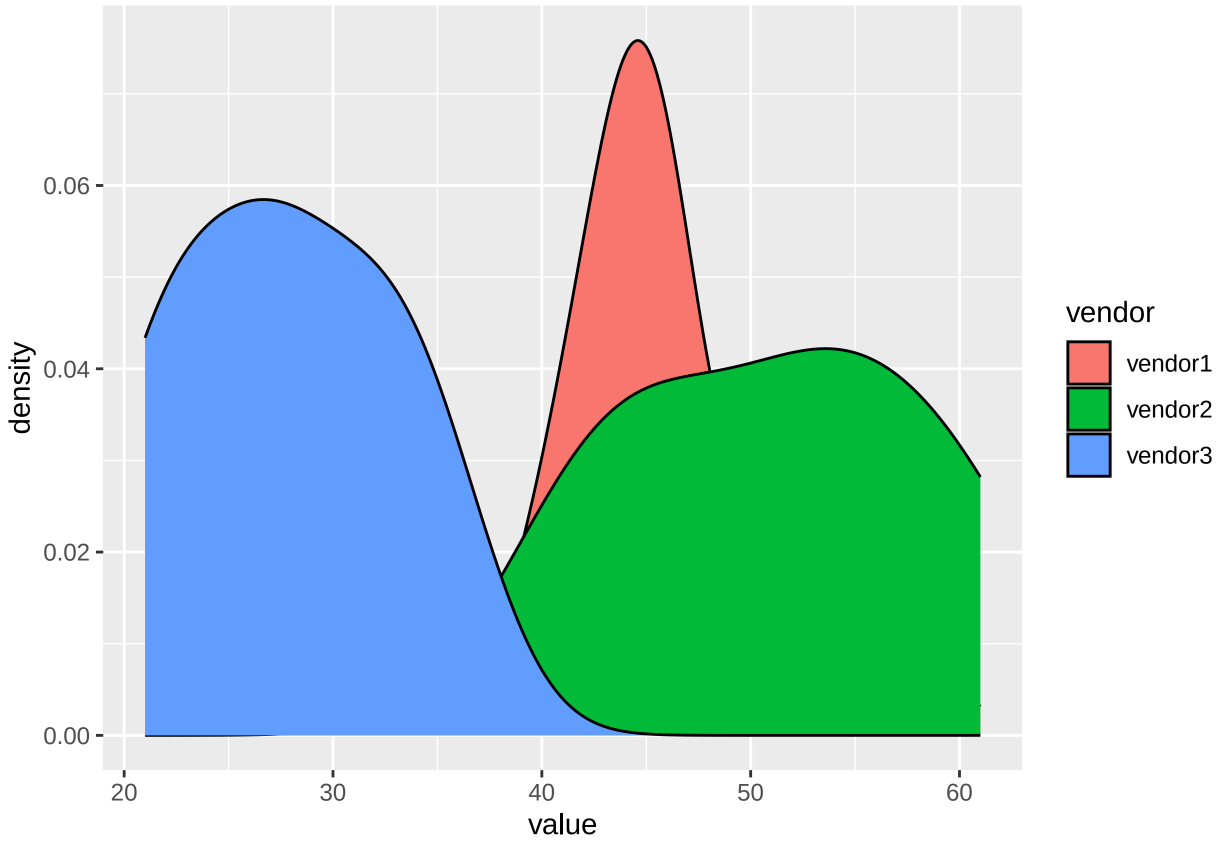
An HR organisation wishes to introduce a new set of training programs to increase the work efficiency of their employees. To do this, the organisation hires 3 vendors such that each vendor will train a group of employees with a one-week program. It is assumed that the employees across the three groups have same work efficiencies before the start of the program. The following table summarises the scores obtained post the one-week training. Each group has 8 employees.

Now, find out the mean efficiency of each vendor and determine which vendor shows a higher significance of improvement in employee efficiency statistically at the 95% level.

R

Visualise distributions

```
# load data
library(readxl)
vendor_scores <- read_excel("~/PgDip/Coding/Module 2/pgdip_module2_practice/Datasets/vendor_scores.xlsx

vendor_scores%>%
  pivot_longer(everything(), names_to = "vendor", values_to = "value")%>%
  ggplot(aes(value, fill=vendor))+
  geom_density()
```



Compute summary statistics

```
vendor_scores%>%
  pivot_longer(everything(), names_to = "vendor", values_to = "value")%>%
  group_by(vendor)%>%
  summarise(mean=mean(value),
            sd=sd(value),
            conf_int=paste0(round(mean-1.98*sd,2), "-", round(mean+1.98*sd,2)))
```

```
## # A tibble: 3 x 4
##   vendor   mean    sd conf_int
##   <chr>   <dbl> <dbl> <chr>
## 1 vendor1  44.7  8.58 27.73-61.7
## 2 vendor2  51    7.33 36.5-65.5
## 3 vendor3  27.7  5.28 17.25-38.17
```

Compute ANOVA

```
vendor_scores_pivoted<-vendor_scores%>%
  pivot_longer(everything(), names_to = "vendor", values_to = "value")

res.aov<-aov(value~vendor, data = vendor_scores_pivoted)
summary(res.aov)
```

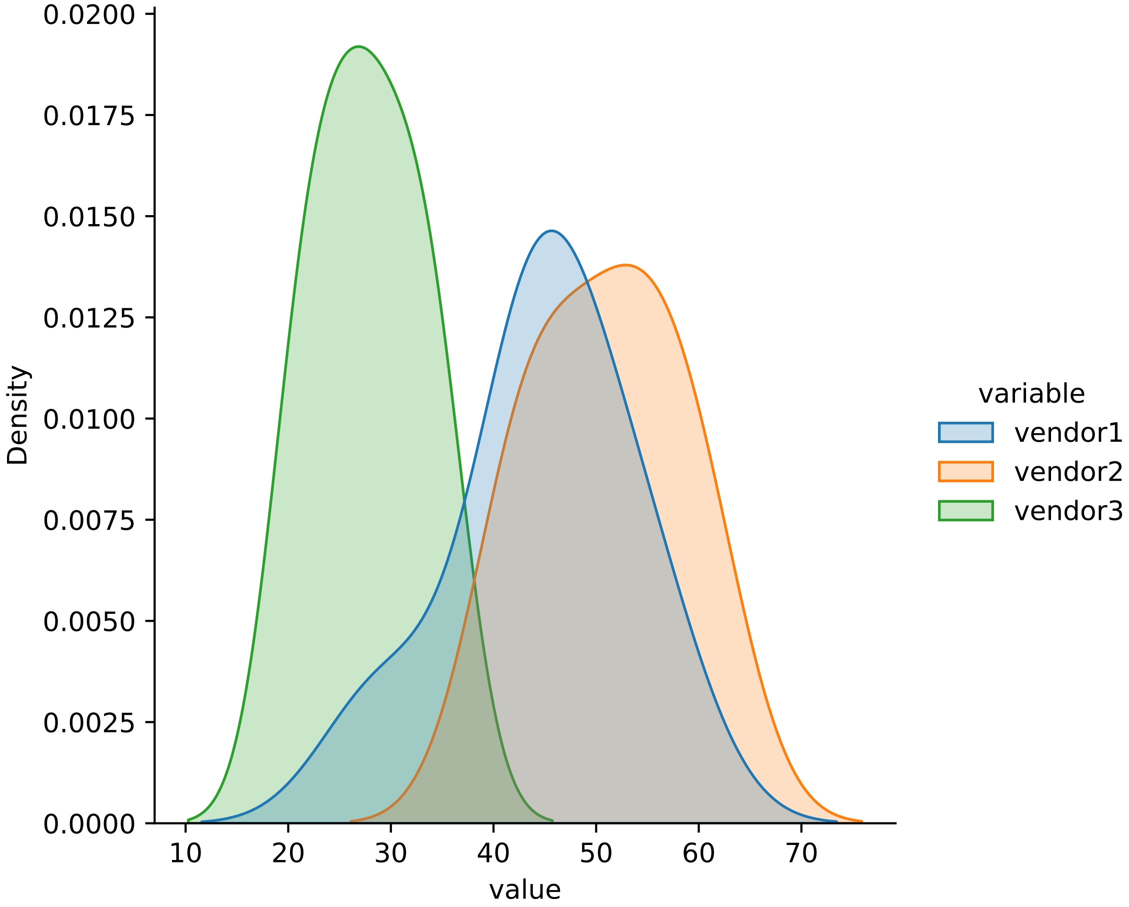
```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## vendor       2 2031.7  1015.9   19.64 2.98e-05 ***
## Residuals   18  930.9    51.7                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the results above, vendor 2 shows a statistically significant higher improvement in employee efficiency after training at the 95% level.

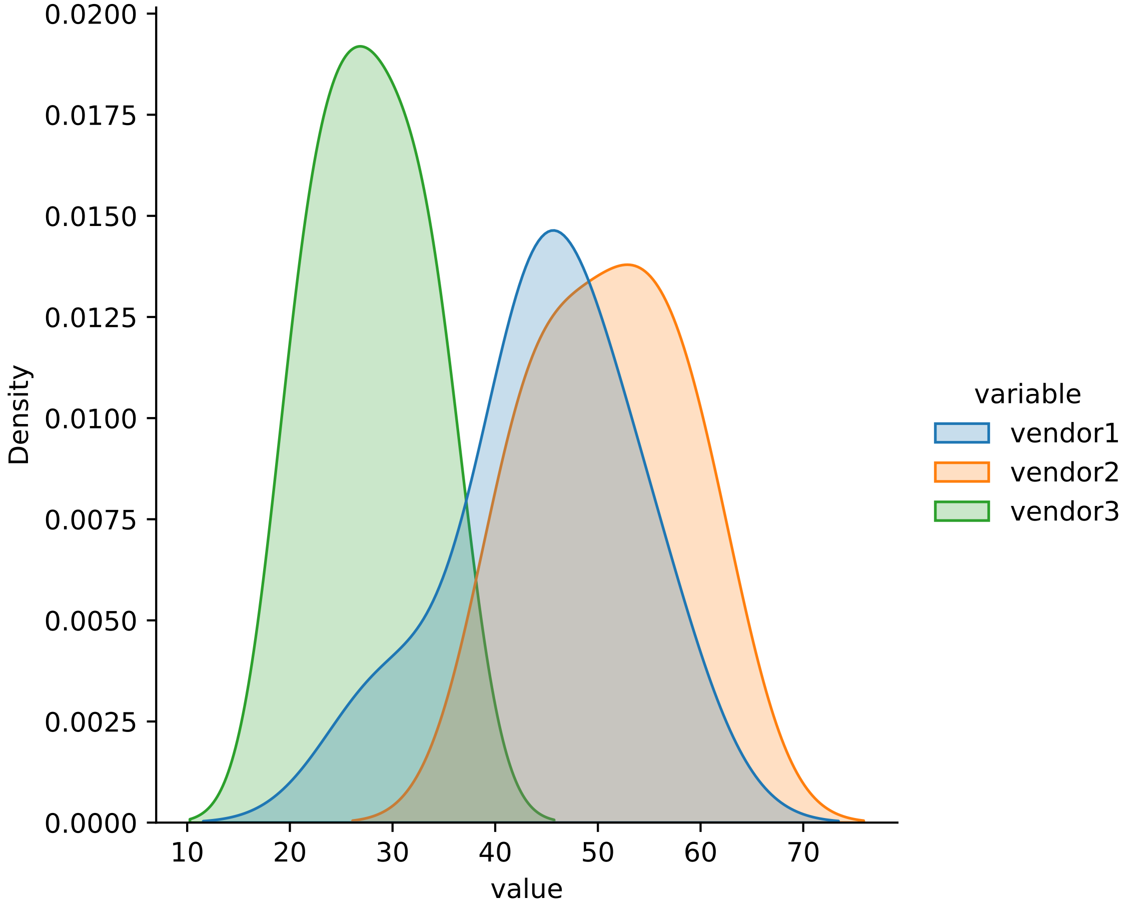
Python

Visualise distributions

```
# load data
vendor_scores = pd.read_excel("C:/Users/guilhermep/Documents/PgDip/Coding/Module 2/pgdip_module2_practi

# load seaborn for visualisation
import seaborn as sns



vendor_scores["id"] = vendor_scores.index

sns.displot(data=pd.melt(vendor_scores, id_vars="id", value_vars=["vendor1", "vendor2", "vendor3"]), x=
```


0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
Density
10
20
30
40
50
60
70
value
variable
vendor1
vendor2
vendor3

```
plt.show()
```



```
#
```

Compute summary statistics

```
print("Means:")
```

```
## Means:
```

```
pd.melt(vendor_scores, id_vars="id", value_vars=["vendor1", "vendor2", "vendor3"]).groupby("variable").m
```

```
##             id      value
## variable
## vendor1   3.0  44.714286
## vendor2   3.0  51.000000
## vendor3   3.0  27.714286
```

```
print("SDs:")
```

```
## SDs:
```

```
pd.melt(vendor_scores, id_vars="id", value_vars=["vendor1", "vendor2", "vendor3"]).groupby("variable").s
```

```
##                 id     value
## variable
## vendor1   2.160247  8.577379
## vendor2   2.160247  7.325754
## vendor3   2.160247  5.282496
```

```
print("95% CIs:")
```

```
## 95% CIs:
```

```
print("Vendor 1: ", stats.t.interval(0.95, len(vendor_scores.vendor1)-1, loc=np.mean(vendor_scores.vendo
```

```
## Vendor 1:  (36.78153332654523, 52.6470381020262)
```

```
print("Vendor 2: ", stats.t.interval(0.95, len(vendor_scores.vendor2)-1, loc=np.mean(vendor_scores.vendo
```

```
## Vendor 2:  (44.22480788212335, 57.77519211787665)
```

```
print("Vendor 3: ", stats.t.interval(0.95, len(vendor_scores.vendor3)-1, loc=np.mean(vendor_scores.vendo
```

```
## Vendor 3:  (22.828791935417676, 32.599779493153754)
```

Compute anova

```
stats.f_oneway(vendor_scores.vendor1,vendor_scores.vendor2,vendor_scores.vendor3)
```

```
## F_onewayResult(statistic=19.643646408839746, pvalue=2.9848866902345244e-05)
```