Lab 10: Lab-ubu – SOLUTIONS

PSTAT 100, Summer Session A 2025 with Ethan P. Marzban

MEMBER 1 (NetID 1) MEMBER 2 (NetID 2)
MEMBER 3 (NetID 3)

July 24, 2025

Required Packages

```
library(ottr)        # for checking test cases (i.e. autograding)
library(pander)      # for nicer-looking formatting of dataframe outputs
library(tidyverse)   # for graphs, data wrangling, etc.
```

Logistical Details

Logistical Details

- This lab is due by **11:59pm on Wednesday, July 30, 2025**.
- Collaboration is allowed, and encouraged!
  - If you work in groups, list ALL of your group members' names and NetIDs (not Perm Numbers) in the appropriate spaces in the YAML header above.
  - Please delete any "MEMBER X" lines in the YAML header that are not needed.
  - No more than 3 people in a group, please.
- Ensure your Lab properly renders to a `.pdf`; non-`.pdf` submissions will not be graded and will receive a score of 0.
- Ensure all test cases pass (test cases that have passed will display a message stating `"All tests passed!"`)

Lab Overview and Objectives

Welcome to another PSTAT 100 Lab! In this lab, we will cover the following:

- Clustering in `R`
- Missing Data

Part I: Clustering

Labubus are a moderately-sized plush doll that has completely taken the online market recently. If you'd like to read more about the history of Labubus, feel free to consult This NPR Article.

For the purposes of this lab, we'll explore a simulated dataset pertaining to *Labubus*. Specifically, we are imagining that someone has collectd 61 labubus and recorded the following attributes:

- The height of the *Labubu* (in centimeters)
- The weight of the *Labubu* (in grams)
- The `RBG` color specification of the *Labubu* (split across three columns; one for `R`, one for `G`, and one for `B`)

Additionally, it is known that each *Labubu* falls into one of six styles: Soymilk (`SM`), Lychee Berry (`LB`), Green Grape (`GG`), Sea Salt Coconut (`SSC`), Toffee (`T`), and Sesame Bean (`SB`) (names taken from this listing).

The full *Labubu* dataset can be found in the `labubu.csv` file, located in the `data/` subfolder.

```
labubu <- read.csv("data/labubu.csv")
```

! Question 1

Create a data frame called `labubu_num` that contains only the numerical columns from the `labubu` dataset.

Solution:

```
## replace this line with your code
labubu_num <- labubu %>% dplyr::select(where(is.numeric))
```

Answer Check:

```
# DO NOT EDIT THIS LINE
invisible({check("tests/q1.R")})
```

```
All tests passed!
```

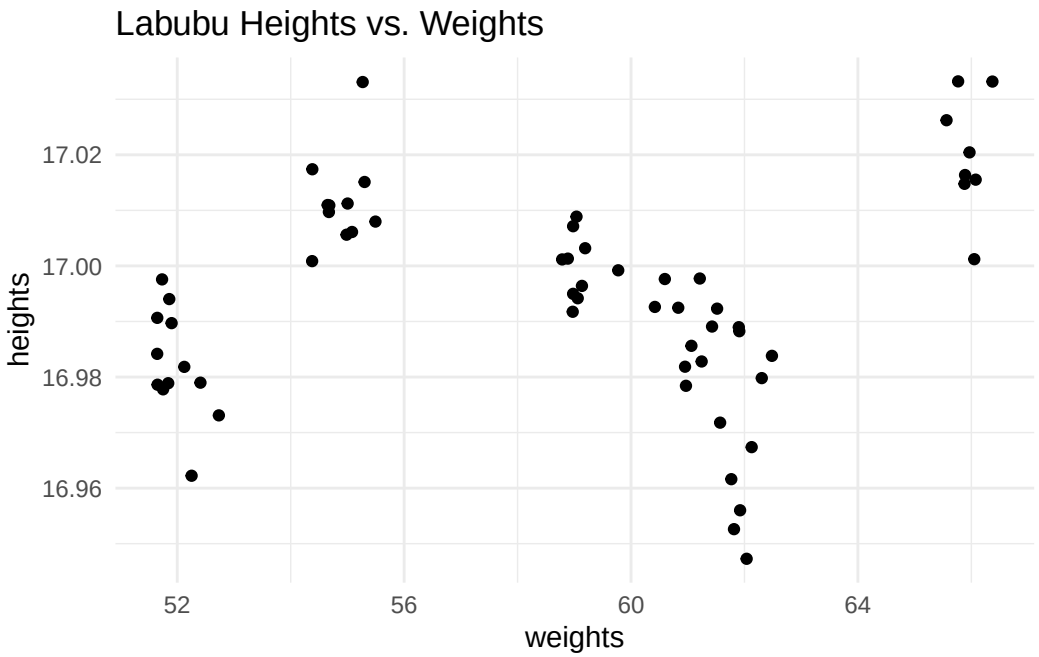
Let's see what happens if we try to visually classify based on only two variables.

! Question 2

Generate a scatterplot of the Labubus' heights (on the vertical axis) against their weights (on the horizontal) axis. Ensure your plot is presentation-quality. How many clusters do you see on the plot? **Don't** rely on the `type` column from the `labubu` dataframe - only use the plot you just generated!

Solution:

```
## replace this line with your code
labubu_num %>%
  ggplot(aes(x = weights, y = heights)) +
  geom_point() +
  theme_minimal() +
  ggtitle("Labubu Heights vs. Weights")
```



Replace this line with your answer
It appears as though there are around 4 clusters.

Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

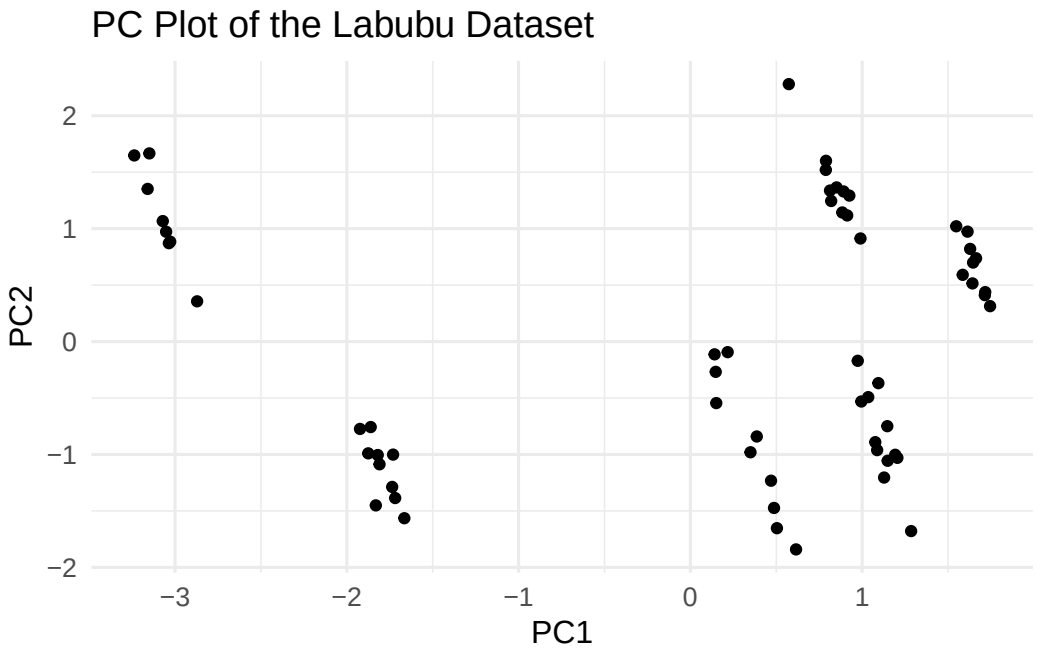
Hm, our number of clusters doesn't appear to agree with the number of different *Labubu* types we know to be present in the dataset. This *could* mean that two or more of the *Labubu* types are very similar to one another - however, it's dangerous to make that conclusion based solely on our work in Question 2 since we ignored several variables from the dataset. Let's remedy that!

! Question 3

Perform PCA on the `labubu_num` dataframe, taking care to *standardize* (not only mean-center) each column. Then, use this to generate a plot of the first two latent factors; how many clusters do you see on this PC plot?

Solution:

```
## replace this line with your code
prcomp(labubu_num, scale. = TRUE)$x[, 1:2] %>%
  data.frame() %>%
  ggplot(aes(x = PC1, y = PC2)) +
  geom_point() +
  ggtitle("PC Plot of the Labubu Dataset") +
  theme_minimal(base_size = 12)
```



Replace this line with your answer
Now, it appears as though there are 6 clusters.

Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

Wow- the clusters on this graph are pretty distinct! Nevertheless, let's run a K-means clustering algorithm on our PC-projected dataset. As a reminder, in `R` we use the `kmeans()` function to run the K-means clustering algorithm; take a look at the help file for this function before moving onto the next question on this lab.

! Question 4

Part (a)

Pass the matrix of the first two PCs into a call to `kmeans()`, and assign the result to a variable called `kmeans_labubu`. Use 6 clusters. **Make sure to set your seed to 100 before beginning; otherwise the autograder may not work.**

Solution:

```
set.seed(100)
## replace this line with your code
kmeans_labubu <- prcomp(
  labubu_num, scale. = TRUE)$x[, 1:2] %>%
  kmeans(centers = 6)
```

Answer Check:

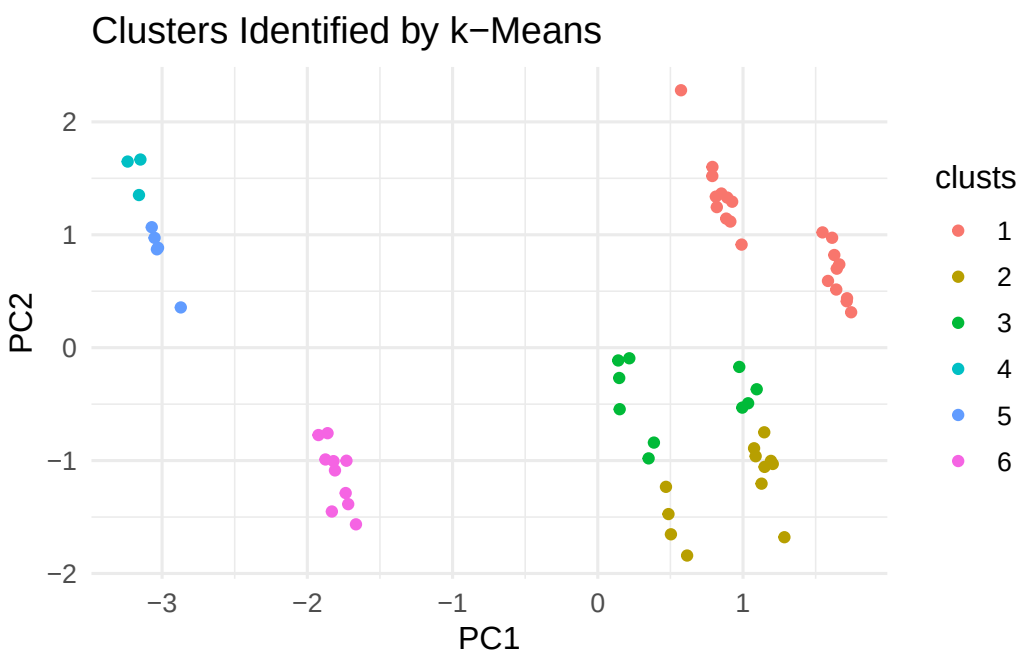
```
# DO NOT EDIT THIS LINE
invisible({check("tests/q4a.R")})
```

```
All tests passed!
```

Part (b)

Re-do the PC plot from Question 3, now coloring based on the clusters identified by `kmeans()`.

```
prcomp(labubu_num, scale. = TRUE)$x[, 1:2] %>%
  data.frame() %>%
  mutate(clusts = kmeans_labubu$cluster %>% factor()) %>%
  ggplot(aes(x = PC1, y = PC2)) +
  geom_point(aes(colour = clusts)) +
  ggtitle("Clusters Identified by k-Means") +
  theme_minimal(base_size = 12)
```



Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

Assuming you correctly completed the questions above, the following code should produce a cross-tabulation of each *Labubu*'s classification (as determined by K-means) and its true type (as classified by the `type` variable from the original `labubu` dataframe):

```
labubu %>%
  mutate(class = kmeans_labubu$cluster) %>%
  group_by(type, class) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = type,
              values_from = count,
              values_fill = 0) %>%
  pander()
```

class	GG	LB	SB	SM	SSC	T
6	10	0	0	0	0	0
1	0	11	0	10	0	0
2	0	0	4	0	8	0
3	0	0	6	0	4	0
4	0	0	0	0	0	3
5	0	0	0	0	0	5

! Question 5

Has K-means classified along the different types, as encoded by the `type` variable in the original `labubu` dataframe? Are there any two *Labubu* types that are so similar to each other that they are indistinguishable (by K-means) from one another? How can you tell?

Solution:
Replace this line with your answer

For one thing, take a look at Classes 2 and 3: they are classified somewhat differently from how we might have classified them by hand. From the table, we see that these classes have actually split the Sesame Bean and Sea Salt Coconut types. Additionally, Class 1 seems to have grouped together the Lychee Berry and Soymilk Types. All in all, it appears as though the classification is NOT perfectly along type-lines, indicating that some types are more similar to each other than previously thought.

Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

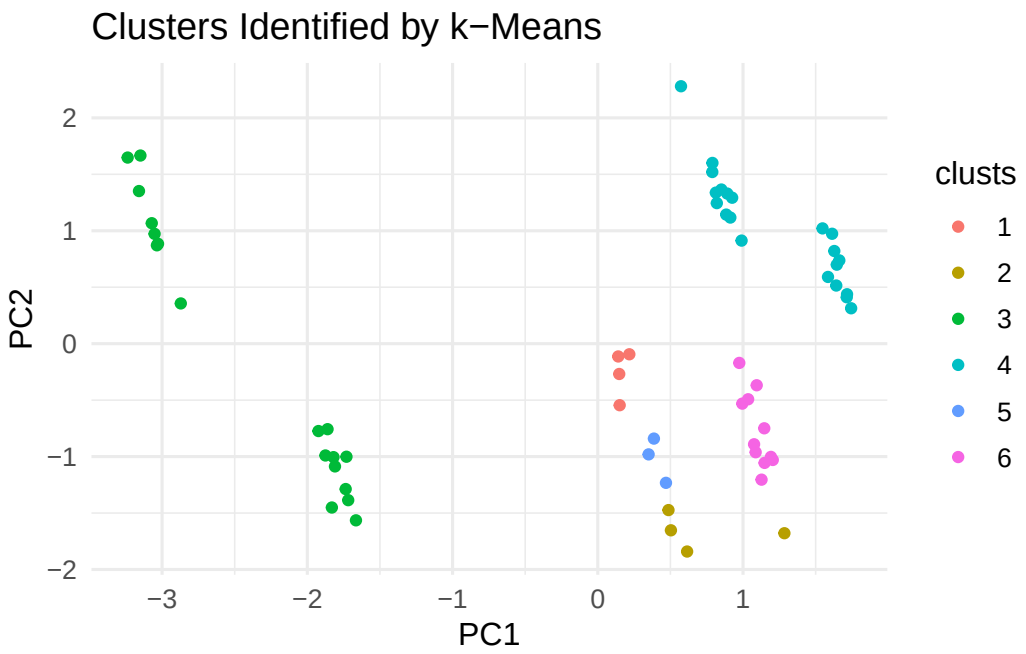
Something that is important to note is that there is a bit of randomness in the K-means algorithm. As such, let's see how this manifests itself in this particular dataset.

! Question 6

Re-run the K-means clustering algorithm, but this time **set your seed to 20 at the start**. Produce a plot like you did in Question 4(b), coloring points by the *new* classifications. Comment on any differences between this plot and the plot from 4(b) above.

```
set.seed(20)
## replace this line with your code
kmeans_labubu2 <- prcomp(
  labubu_num, scale. = TRUE)$x[, 1:2] %>%
  kmeans(centers = 6)

prcomp(labubu_num, scale. = TRUE)$x[, 1:2] %>%
  data.frame() %>%
  mutate(clusts = kmeans_labubu2$cluster %>% factor()) %>%
  ggplot(aes(x = PC1, y = PC2)) +
  geom_point(aes(colour = clusts)) +
  ggtitle("Clusters Identified by k-Means") +
  theme_minimal(base_size = 12)
```



Replace this line with your answer

The clustering has changed a fair amount. This is likely due to the fact that the standard deviations are relatively small, along with the fact that sample sizes are relatively small.

Answer Check:

There is no autograder for this question; your TA will manually check that your answers are correct.

So, there is some variability in the product of the K-means clustering algorithm. Again, recall that the K-means algorithm is only optimal in a local sense - in some cases (like with this particular simulated *Labubu* dataset), the final configurations may be dependent on the seed you set initially.

Submission Details

Congrats on finishing this PSTAT 100 lab! Please carry out the following steps:

Submission Details

1) Check that all of your tables, plots, and code outputs are rendering correctly in your final `.pdf`.

2) Check that you passed all of the test cases (on questions that have autograders). You'll know that you passed all tests for a particular problem when you get the message "All tests passed!".

3) Submit **ONLY** your `.pdf` to Gradescope. Make sure to **match ALL pages to the ONE question on Gradescope**; failure to do so will incur a penalty of 0.1 points.