dklick dq1 final

January 29, 2024

Dan Klick 1/29/2024 DSC-510 Topic 1 DQ 1

“What are some examples of situations in which basic statistical concepts, such as population, sample, parameter, and statistic, could be useful for making decisions or drawing conclusions? How can Python be used to help analyze data in these situations?”

As an avid golfer and a lover of data and statistics, I spend a lot of time analyzing my stats related to my golf game. In this discussion post, I wanted to illustrate the importance of population size on statistics, specifically regarding minimizing the impact of outliers.

Since I am an amateur golfer (and not a great one at that), calculating average distance with clubs can be difficult without a properly sized population. The first generated table is a list of distance hit over 10 shots with each of my wedges, also included is the mean for each club. Since my skill level is not that great, there is a lot of variation within the data, which leads to means that are not accurate. By increasing the population size to 100 shots with each club, I can minimize the impact of the variation and get means that are far more accurate. For reference, from data over thousands of recorded shots, the average distance with my LW, SW, GW, and PW are as follows respectively: 65, 90, 115, and 120.

Rogel-Salazar, J. (2023). Statistics and data visualisation with python. CRC Press, Taylor & Francis Group.

[1]:
```
# import required libraries
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

[2]:
```
# import data sets from csv files
golf_data_small = pd.read_csv("C:/Users/Dan/Documents/
  ↪golf_data_small_population.csv")
golf_data_large = pd.read_csv("C:/Users/Dan/Documents/
  ↪golf_data_large_population.csv")
```

[3]:
```
# printing small population data set and calculated means
print(golf_data_small)
club_avg_small_LW = golf_data_small['LW'].mean()
club_avg_small_SW = golf_data_small['SW'].mean()
club_avg_small_GW = golf_data_small['GW'].mean()
club_avg_small_PW = golf_data_small['PW'].mean()
print('LW avg: ' + str(club_avg_small_LW))
print('SW avg: ' + str(club_avg_small_SW))
print('GW avg: ' + str(club_avg_small_GW))
print('PW avg: ' + str(club_avg_small_PW))
```

```
   Shot  LW  SW   GW   PW
0     1  40  80  115  125
1     2  68  81  114  122
2     3  66  85  100  120
3     4  42  86  105  120
4     5  45  87  128  111
5     6  50  85  112  136
6     7  72  92  116  127
7     8  65  77  114  121
8     9  82  80  115  126
9    10  65  85  120  119
LW avg: 59.5
SW avg: 83.8
GW avg: 113.9
PW avg: 122.7
```

[4]:

```
# printing large population data set and calculated means
print(golf_data_large)
club_avg_large_LW = golf_data_large['LW'].mean()
club_avg_large_SW = golf_data_large['SW'].mean()
club_avg_large_GW = golf_data_large['GW'].mean()
club_avg_large_PW = golf_data_large['PW'].mean()
print('LW avg: ' + str(club_avg_large_LW))
print('SW avg: ' + str(club_avg_large_SW))
print('GW avg: ' + str(club_avg_large_GW))
print('PW avg: ' + str(club_avg_large_PW))
```

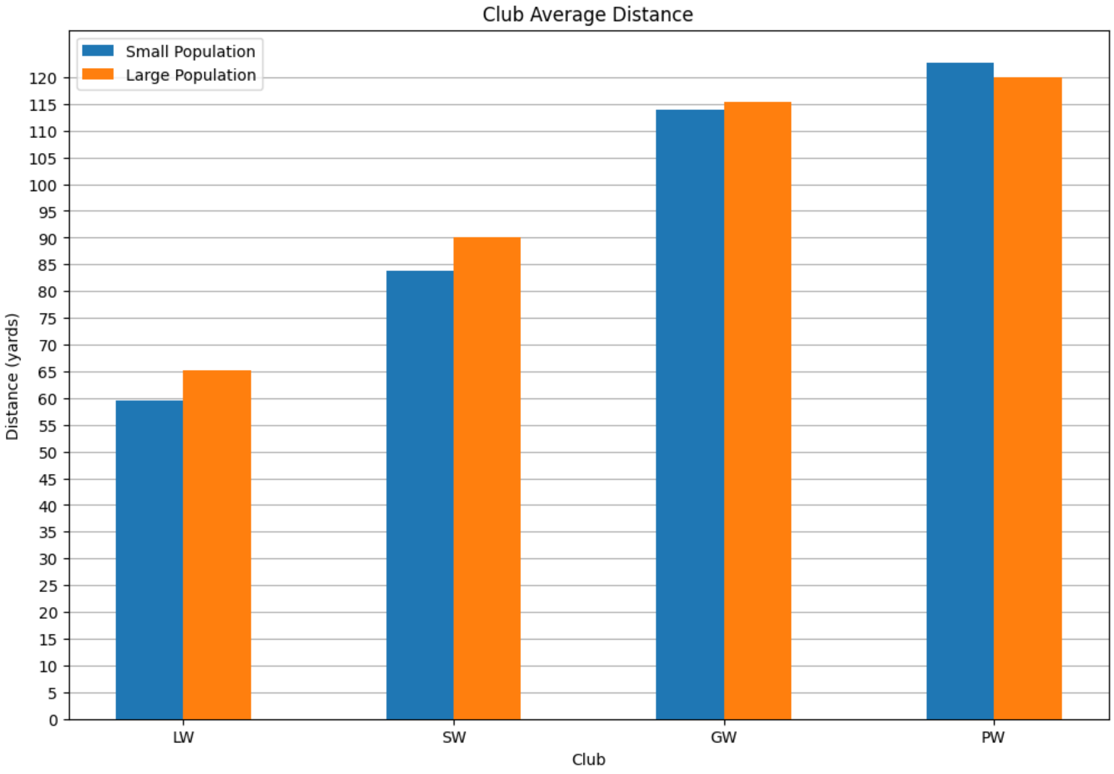
```
    Shot  LW  SW   GW   PW
0      1  53  76  108  113
1      2  59  80  102  107
2      3  60  77  100  114
3      4  50  82  103  111
4      5  57  81  104  115
..   …  ..  ..  …  …
95    96  66  92  115  116
96    97  60  92  116  121
97    98  66  92  113  121
98    99  64  93  117  118
99   100  67  94  110  119

[100 rows x 5 columns]
LW avg: 65.15
SW avg: 89.98
GW avg: 115.35
```

```
PW avg: 119.91
```

[5]:

```
# setting width of bars
barwidth = .25
fig = plt.subplots(figsize =(12, 8))
# setting height of bars
SMALL = [club_avg_small_LW, club_avg_small_SW, club_avg_small_GW, club_avg_small_PW]
LARGE = [club_avg_large_LW, club_avg_large_SW, club_avg_large_GW, club_avg_large_PW]
# setting bar position
bar1 = np.arange(len(SMALL))
bar2 = [x + barwidth for x in bar1]
plt.bar(bar1, SMALL, width = barwidth, label ='Small Population', zorder=3)
plt.bar(bar2, LARGE, width = barwidth, label ='Large Population', zorder=3)
plt.xlabel('Club')
plt.ylabel('Distance (yards)')
plt.xticks(bar1 + .5*barwidth,
           ['LW', 'SW', 'GW', 'PW'])
plt.yticks(np.arange(0, 125, step=5))
plt.grid(axis='y', zorder=0)
plt.title('Club Average Distance')
plt.legend()
plt.show()
```



[ ]: