BE601 - DATA ANALYTICS I

Describing Data: Numerical

Uyen Vu

29/08/2022

# Measures of central tendency and location

1. Mean, Median, and Mode (describe central tendency)

* arithmetic mean:
* median: odd n vs even n
* mode: most regular observation \*\*
  + one mode: unimodal distribution
  + two modes: bimodal distribution
  + two modes: multimodal distribution

1. Example

#exp 2.1 (p65)  
demand <- c(60,84,65,67,75,72,80,85,63,82,70,75)  
mean(demand)

## [1] 73.16667

median(demand)

## [1] 73.5

df <- as.data.frame(table(demand))  
colnames(df) <- c("value","freq")  
df <- df[order(-df[, "freq"]), ]  
df #mode is 75

## value freq  
## 7 75 2  
## 1 60 1  
## 2 63 1  
## 3 65 1  
## 4 67 1  
## 5 70 1  
## 6 72 1  
## 8 80 1  
## 9 82 1  
## 10 84 1  
## 11 85 1

#exp 2.2 (p65)  
eps <- c(0.05,0.05,0.081,0.136,0.232,0.207,0.12,0.142)  
mean(eps)

## [1] 0.12725

median(eps)

## [1] 0.128

df <- as.data.frame(table(eps))  
colnames(df) <- c("value","freq")  
df <- df[order(-df[, "freq"]), ]  
df

## value freq  
## 1 0.05 2  
## 2 0.081 1  
## 3 0.12 1  
## 4 0.136 1  
## 5 0.142 1  
## 6 0.207 1  
## 7 0.232 1

1. Geometric mean
2. Percentiles and Quartiles

* five-number summary: minimum < Q1 < Median < Q3 < maximum
* **box-and-whisker plot** is used to describe five-number summary

#exp 2.5  
demand <- c(60,84,65,67,75,72,80,85,63,82,70,75)  
demand <- sort(demand)  
demand

## [1] 60 63 65 67 70 72 75 75 80 82 84 85

fivenum(demand)

## [1] 60.0 66.0 73.5 81.0 85.0

#exp 2.6  
shopping <- c(18,46,45,20,33,33,21,31,23,34,42,38,31,38,21,37,37,30,42,34,34,18,30,48,51,52,19,37,30,25,42,41,34,50,52,50,19,21,34,25,18,25,25,43,59,37,23,23,40,31,45,51,45,60,30,40,37,21,34,34,42,43,60,40,37,20,40,18,21,52,18,68,28,57,63,57,63,31,67,25,69,34,69,57,69,57,70,18,70,70,71,73,73,71,70,69,68,64,59,18,47,52,55,25)  
shopping <- sort(shopping)  
shopping

## [1] 18 18 18 18 18 18 18 19 19 20 20 21 21 21 21 21 23 23 23 25 25 25 25 25 25  
## [26] 28 30 30 30 30 31 31 31 31 33 33 34 34 34 34 34 34 34 34 37 37 37 37 37 37  
## [51] 38 38 40 40 40 40 41 42 42 42 42 43 43 45 45 45 46 47 48 50 50 51 51 52 52  
## [76] 52 52 55 57 57 57 57 59 59 60 60 63 63 64 67 68 68 69 69 69 69 70 70 70 70  
## [101] 71 71 73 73

q1\_position <- 0.25\*(length(shopping)+1)  
q1 <- 28 + 0.25\*(30-28)  
q1

## [1] 28.5

fivenum(shopping)

## [1] 18 29 39 56 73

# Shape of distribution

1. **skewness**

* Skewness is positive: skewed-right (mean > median)
* skewness is negative: skewed-left (mean < median)

1. R

* Base R does not have the function to calculate skewness
* Package **“moment”** with the command **skewness()**

# Measures of variability

1. **Range** = max - min
2. **IQR** and **box-and-whisker plot**

* Interquartile range (IQR) = Q3 - Q1 (remove the lowest 25% of the data and the highest 25% of the data)
* Box-and-Whisker Plot

1. **variance** and **standard deviation**

#exp 2.8, p74  
sales <- data.frame(Location\_1 = c(6,8,10,12,14,9,11,7,13,11),  
 Location\_2 = c(1,19,2,18,11,10,3,17,4,17),  
 Location\_3 = c(2,3,25,20,22,19,25,20,22,26),  
 Location\_4 = c(22,20,10,13,12,10,11,9,10,8))  
sales

## Location\_1 Location\_2 Location\_3 Location\_4  
## 1 6 1 2 22  
## 2 8 19 3 20  
## 3 10 2 25 10  
## 4 12 18 20 13  
## 5 14 11 22 12  
## 6 9 10 19 10  
## 7 11 3 25 11  
## 8 7 17 20 9  
## 9 13 4 22 10  
## 10 11 17 26 8

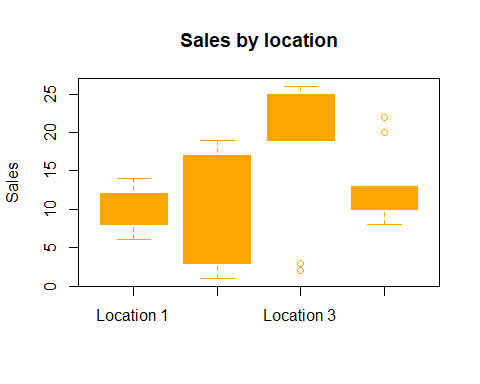
summary(sales)

## Location\_1 Location\_2 Location\_3 Location\_4   
## Min. : 6.00 Min. : 1.00 Min. : 2.00 Min. : 8.00   
## 1st Qu.: 8.25 1st Qu.: 3.25 1st Qu.:19.25 1st Qu.:10.00   
## Median :10.50 Median :10.50 Median :21.00 Median :10.50   
## Mean :10.10 Mean :10.20 Mean :18.40 Mean :12.50   
## 3rd Qu.:11.75 3rd Qu.:17.00 3rd Qu.:24.25 3rd Qu.:12.75   
## Max. :14.00 Max. :19.00 Max. :26.00 Max. :22.00

sapply(sales,IQR)

## Location\_1 Location\_2 Location\_3 Location\_4   
## 3.50 13.75 5.00 2.75

boxplot(sales$Location\_1, sales$Location\_2, sales$Location\_3, sales$Location\_4,  
 main = "Sales by location",  
 ylab = "Sales",  
 at = c(1,2,3,4),  
 names = c("Location 1","Location 2","Location 3","Location 4"),  
 col = "orange",  
 border = "orange")



#exp 2.9, p76  
sales <- c(6,8,10,12,14,9,11,7,13,11)  
mean(sales)

## [1] 10.1

mean <- rep(mean(sales), 10)  
df <- as.data.frame(cbind(sales,mean))  
df

## sales mean  
## 1 6 10.1  
## 2 8 10.1  
## 3 10 10.1  
## 4 12 10.1  
## 5 14 10.1  
## 6 9 10.1  
## 7 11 10.1  
## 8 7 10.1  
## 9 13 10.1  
## 10 11 10.1

df$s\_diff\_m <- df$sales - df$mean  
df$s\_diff\_m\_sq <- df$s\_diff\_m^2  
var <- sum(df$s\_diff\_m\_sq)/(length(sales)-1)  
var

## [1] 6.766667

sd <- var^0.5  
sd

## [1] 2.601282

var(sales)

## [1] 6.766667

sd(sales)

## [1] 2.601282

1. Coefficient of Variation

* to adjust standard deviation as percentage of mean (unadjusted sd is biased when we try to compare 2 datasets of means of large difference)

df <- data.frame(a=c(88, 85, 82, 97, 67, 77, 74, 86, 81, 95),  
 b=c(77, 88, 85, 76, 81, 82, 88, 91, 92, 99),  
 c=c(67, 68, 68, 74, 74, 76, 76, 77, 78, 84))  
df

## a b c  
## 1 88 77 67  
## 2 85 88 68  
## 3 82 85 68  
## 4 97 76 74  
## 5 67 81 74  
## 6 77 82 76  
## 7 74 88 76  
## 8 86 91 77  
## 9 81 92 78  
## 10 95 99 84

CV <- sapply(df, function(x) sd(x)/mean(x)\*100)  
CV

## a b c   
## 11.012892 8.330843 7.154009

1. Chebyshev’s theorem and the empirical rule

* Chebyshev’s theorem: For any population with unknown distribution, with mean and sd, k > 1 is the number of sd, the percentage of observations that lie within the interval **[mean +/- sd\*k]** is **100[1 - 1/k^2]%**
* Empirical rule: For large population with bell-shaped distribution:
  + approximately 68% of the obs are in the interval [mean +/- 1sd]
  + approximately 95% of the obs are in the interval [mean +/- 2sd]
  + almost all of the obs are in the interval [mean +/- 3sd]

#exp 2.13  
mean <- 1200  
sd <- 50  
#If shape of distribution is unknown: apply Chebyshev theorem  
interval1\_min <- mean - 1.5\*sd  
interval1\_min

## [1] 1125

interval1\_max <- mean + 1.5\*sd  
interval1\_max

## [1] 1275

interval2\_min <- mean - 2\*sd  
interval2\_min

## [1] 1100

interval2\_max <- mean + 2\*sd  
interval2\_max

## [1] 1300

interval3\_min <- mean - 3\*sd  
interval3\_min

## [1] 1050

interval3\_max <- mean + 3\*sd  
interval3\_max

## [1] 1350

k1\_5 <- 100\*(1-1/(1.5^2))  
k1\_5

## [1] 55.55556

k2 <- 100\*(1-1/(2^2))  
k2

## [1] 75

k3 <- 100\*(1-1/3^2)  
k3

## [1] 88.88889

#If shape of distribution is known as bell-shape: apply empirical rule

1. z-score

* percentile & quartiles: location of a value **relative to the entire dataset**
* z-Score: location of a value **relative to the mean of distribution**
  + number of sd that a value is from the mean
  + 0: value > mean
  + =0: value = mean
  + <0: value < mean
  + **z-score = (value of x - mean)/sd**

#exp 2.14  
mean <- 1200  
sd <- 50  
z\_1120 <- (1120-1200)/50  
z\_1120

## [1] -1.6

# Weighted mean and measure of grouped data

* Approximate mean and variance for grouped data is used when we don’t know the exact value (e.g. a person is asked to choose his age range between 20 - 29 rather than giving an exact value)

# Measures of relationships between variables

* **scatterplot** is used as a graphical way of describing relationship
* **covariance** and **correlation** are used as numerical ways

post <- c(16,31,27,23,15,17,17,18,14)  
interaction <- c(165,314,280,195,137,286,199,128,462)  
df <- as.data.frame(cbind(post,interaction))  
df$mean\_post <- mean(df$post)  
df$mean\_int <- mean(df$interaction)  
df$mean\_post\_diff <- df$post - df$mean\_post  
df$mean\_int\_diff <- df$interaction - df$mean\_int  
df$mean\_diff\_prod <- df$mean\_post\_diff\*df$mean\_int\_diff  
cov\_manual <- sum(df$mean\_diff\_prod)/(nrow(df)-1)  
cov\_manual

## [1] 81.54167

cov\_r <- cov(post,interaction)  
cov\_r

## [1] 81.54167

df$mean\_post\_diff\_sq <- df$mean\_post\_diff^2  
df$mean\_int\_diff\_sq <- df$mean\_int\_diff^2  
var\_post <- sum(df$mean\_post\_diff\_sq)/(nrow(df)-1)  
var\_int <- sum(df$mean\_int\_diff\_sq)/(nrow(df)-1)  
cor\_manual <- cov\_manual/(var\_post^0.5 \* var\_int^0.5)  
cor\_manual

## [1] 0.1298312

cor\_r <- cor(post,interaction, method = "pearson")  
cor\_r

## [1] 0.1298312