9/13/2020

1) Please answer the following questions. These assume you have read and worked through the exercises up to Lecture03\_Descriptives and explotingAssumptions. Where necessary, please print your R output as your answer. To print: first print your data from the command prompt, take a screen shot (command+shift+4, for mac; PrintScreen on Windows),

1. Please identify the level of measurement for each of the following. (4 points)

* Medal standing of countries in Olympic games :Ordinal
* # of medals of each country in Olympic games :Ratio
* Hostility, as measured by Cook-Medley Hostility Scale interval but may be ordinal so its multi level
* Memory ability, as measured by the number of words recalled from an initially memorized list: Interval
* Ranking of MLB teams: Ordinal
* Social security numbers: Nominal
* Air distance between Indy and other cities in US: ratio
* Time to clotting of samples of human blood :Ratio
* Give examples that you will come across in your project (e.g. research area)

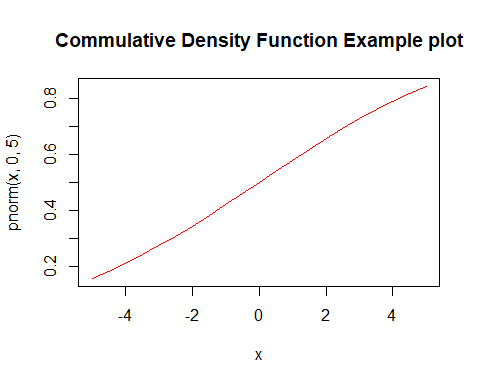
**Nominal: Type 2 Diabetes, Alzheimer’s, Dementia, Gender, Patient Id**

**Ordinal: Age of diagnosis**

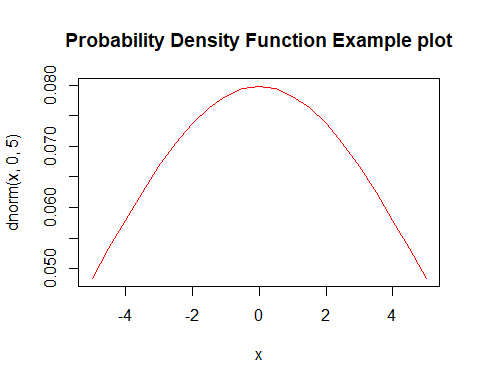
**Interval : Date of birth**

**Ratio: cholesterol, glucose, blood pressure, height, weight etc**

# 2) Using R, plot a normal distribution as a both a PDF and CDF.  
  
# creating example data and plotting cdf and pdf  
  
x=seq(from=-5,to=5,by=0.5)  
plot(x,pnorm(x,0,5),type="l",main="Commulative Density Function Example plot",col="red")



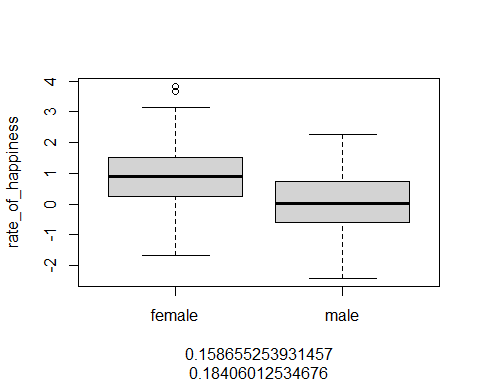
plot(x,dnorm(x,0,5),type="l",main="Probability Density Function Example plot",col="red")



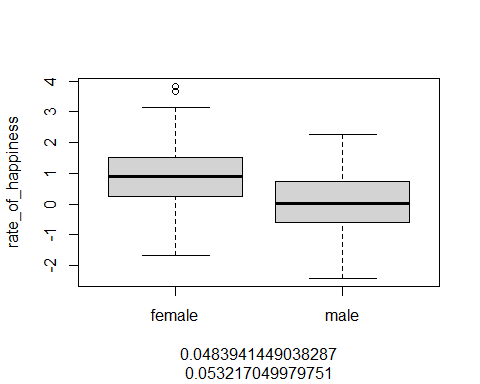
# another example by creating example data   
set.seed(2000)  
sample\_score<-data.frame(gender = factor(rep(c("male","female"), each=200)),   
 rate\_of\_happiness = c(rnorm(200),rnorm(200, mean=.8)))  
  
tail(sample\_score,5)

## gender rate\_of\_happiness  
## 396 female 0.2409993  
## 397 female 2.2296293  
## 398 female 1.5270909  
## 399 female 0.3865732  
## 400 female 1.0606204

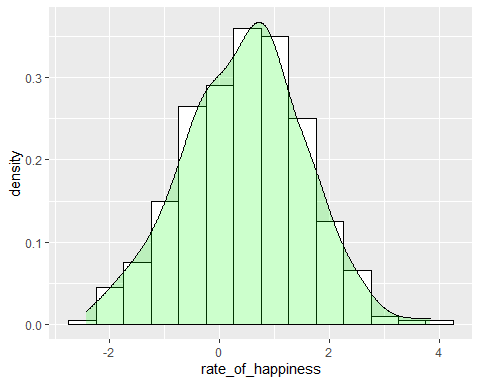
plot(sample\_score,pnorm(x,0,5),type="l")



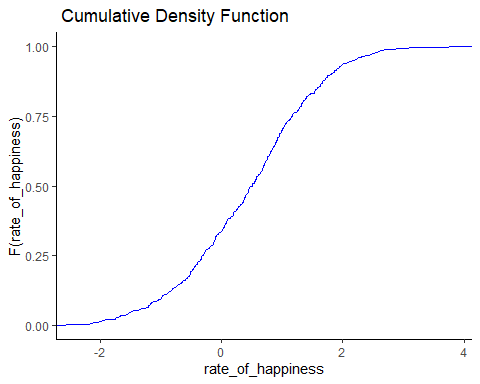
plot(sample\_score,dnorm(x,0,5),type="l")  
  
library(ggplot2)



# Histogram overlaid with kernel density curve  
ggplot(sample\_score, aes(x=rate\_of\_happiness)) +   
 geom\_histogram(aes(y=..density..), # Histogram with density instead of count on y-axis  
 binwidth=.5,  
 colour="black", fill="white") +  
 geom\_density(alpha=.2, fill="green") # Overlay with transparent density plot



# Basic ECDF plot  
ggplot(sample\_score, aes(rate\_of\_happiness)) + stat\_ecdf(geom = "line",color="blue")+  
labs(title=" Cumulative Density Function",  
 y = "F(rate\_of\_happiness)", x="rate\_of\_happiness")+  
theme\_classic()



#4) Calculate the Z-score from a score of 80 from a distribution with a mean of 85 and standard deviation of 10. What percent of scores would fall below this score? What percent of scores would fall above this score?   
# z= scale(x,mean,sd)  
z=scale(80,85,10)  
z

## [,1]  
## [1,] -0.5  
## attr(,"scaled:center")  
## [1] 85  
## attr(,"scaled:scale")  
## [1] 10

#pnorm(x, mean = , sd = , lower.tail= )  
pnorm(80, mean =85 , sd =10 , lower.tail=-0.5 )

## [1] 0.6914625

# Since z-score is negative and pnorm is 0.6915 i.e. (50+19.15)=69.15% of score would fall below mean

mathematically:

z = (x – μ) / σ

= (80-85)/10

= -0.5

# 5) Please go to Canvas and download the 'Programming.csv' data set. Please refer to the table below to define the variables in the data  
  
#loading library  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Loading and Exploring file  
my\_data <- read.csv("programming.csv")  
  
# defining variables  
glimpse(my\_data)

## Rows: 271  
## Columns: 10  
## $ Student <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, ...  
## $ Score <int> 11, 7, 12, 19, 2, 14, 4, 17, 3, 8, 10, 6, 17, 11, 3, 5, 8, ...  
## $ F <int> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0,...  
## $ O <int> 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## $ Prog <int> 0, 0, 0, 1, 0, 2, 1, 1, 0, 2, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,...  
## $ AB <int> 1, 0, 2, 2, 1, 1, 1, 2, 1, 1, 1, 0, 1, 2, 1, 1, 1, 2, 0, 2,...  
## $ BA <int> 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 2,...  
## $ H <int> 0, 0, 0, 2, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,...  
## $ K <int> 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,...  
## $ Lang <int> 2, 2, 2, 1, 2, 1, 1, 0, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2,...

# checking columns and exploring chunk of data  
names(my\_data)

## [1] "Student" "Score" "F" "O" "Prog" "AB" "BA"   
## [8] "H" "K" "Lang"

head(my\_data,5)

## Student Score F O Prog AB BA H K Lang  
## 1 1 11 1 0 0 1 0 0 0 2  
## 2 2 7 1 1 0 0 1 0 0 2  
## 3 3 12 0 1 0 2 0 0 0 2  
## 4 4 19 1 1 1 2 1 2 1 1  
## 5 5 2 1 1 0 1 0 0 0 2

# 5.1) Please calculate the mean and standard deviation for the Final Score in the examinations of all the of Freshmen versus non-Freshmen.   
  
# selecting required columns to work on  
  
F\_Student<-my\_data$F  
Final\_score<-my\_data$Score  
  
# creating dataframe and exploring  
  
Std\_All<-data.frame(F\_Student,Final\_score)  
  
names(Std\_All)

## [1] "F\_Student" "Final\_score"

head(Std\_All, 5)

## F\_Student Final\_score  
## 1 1 11  
## 2 1 7  
## 3 0 12  
## 4 1 19  
## 5 1 2

# selecting scores of all and calculating mean and sd for all  
  
fscore<-select(Std\_All,Final\_score)  
  
mean(Final\_score)

## [1] 10.46125

sd(Final\_score)

## [1] 4.145208

# selecting subset for FRESHMEN AND NONFRESHMEN FROM "F" COLUMN  
my\_data\_freshmen\_1<-subset(my\_data, F ==1)  
my\_data\_not\_freshmen\_0<-subset(my\_data, F ==0)  
  
# Calculating mean and sd for Freshmen  
  
mean(my\_data\_freshmen\_1$Score)

## [1] 10.45992

sd(my\_data\_freshmen\_1$Score)

## [1] 4.199914

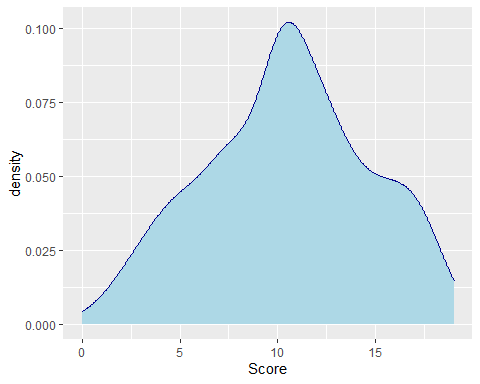
# Calculating mean and sd for NonFreshmen  
  
mean(my\_data\_not\_freshmen\_0$Score)

## [1] 10.47059

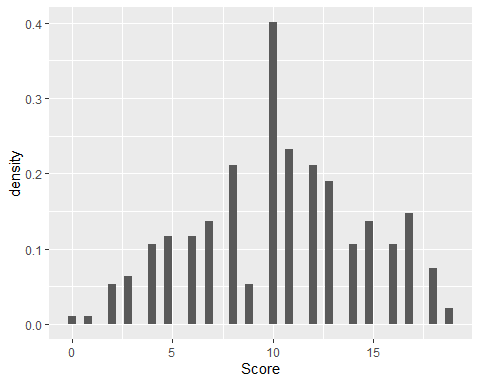
sd(my\_data\_not\_freshmen\_0$Score)

## [1] 3.799803

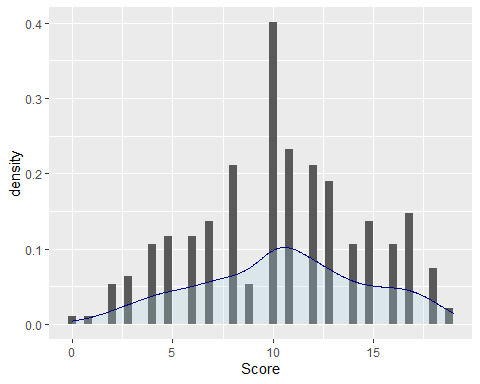
# 5.2) Please provide a qualitative assessment of the normality of the final exam scores in Freshmen and non-Freshmen groups.  
freshmen<- ggplot(my\_data\_freshmen\_1, aes(Score))   
freshmen+ geom\_density(color="darkblue", fill="lightblue")



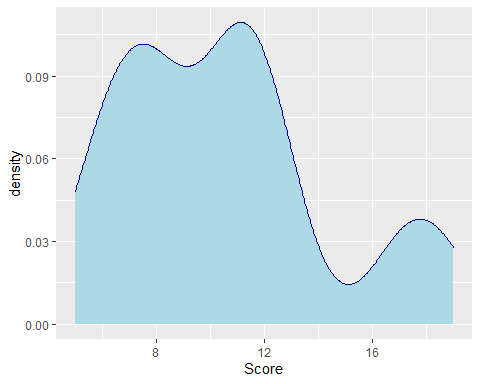
freshmen + geom\_histogram(aes(y=..density..),binwidth = 0.4)



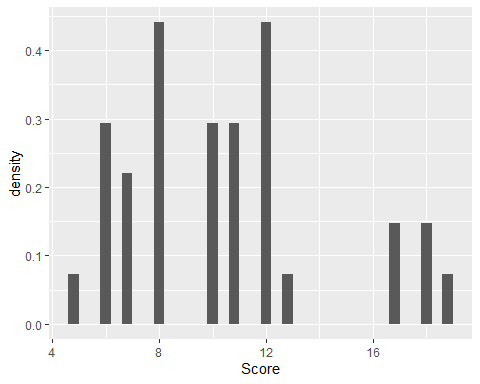
freshmen + geom\_histogram(aes(y=..density..),binwidth = 0.4) +  
 geom\_density(color="darkblue", fill="lightblue", alpha=0.25)



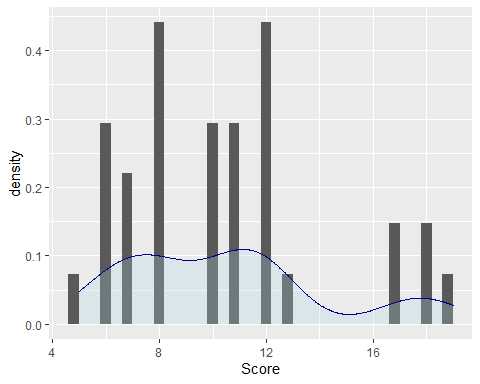
nonfreshmen<- ggplot(my\_data\_not\_freshmen\_0, aes(Score))   
nonfreshmen+ geom\_density(color="darkblue", fill="lightblue")



nonfreshmen + geom\_histogram(aes(y=..density..),binwidth = 0.4)



nonfreshmen + geom\_histogram(aes(y=..density..),binwidth = 0.4) +  
 geom\_density(color="darkblue", fill="lightblue", alpha=0.25)

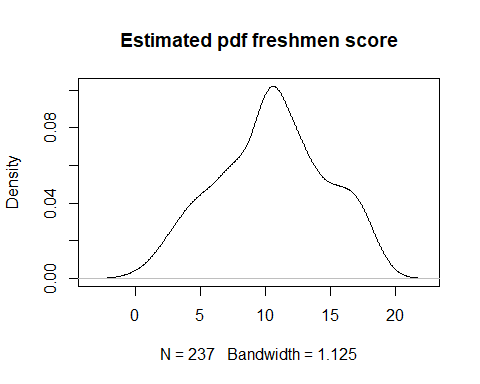


# In both cases data looks normal which means scores are normally ditributed in both groups but freshmen scores are far better normal than non freshmen which looks skewed towards left.  
  
# 5.3)Please assess statistically if final exam scores are normally distributed in Freshmen and non-Freshmen groups.   
  
# install.packages("pastecs")  
  
#loading library  
library(pastecs)

##   
## Attaching package: 'pastecs'

## The following objects are masked from 'package:dplyr':  
##   
## first, last

plot(density(my\_data\_freshmen\_1$Score),main="Estimated pdf freshmen score")



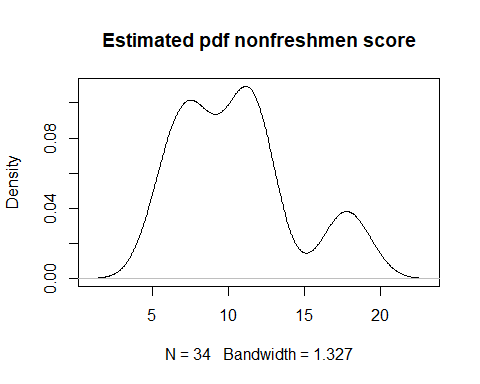
stat.desc(my\_data\_freshmen\_1$Score,basic=T,norm=TRUE)

## nbr.val nbr.null nbr.na min max   
## 2.370000e+02 1.000000e+00 0.000000e+00 0.000000e+00 1.900000e+01   
## range sum median mean SE.mean   
## 1.900000e+01 2.479000e+03 1.000000e+01 1.045992e+01 2.728137e-01   
## CI.mean.0.95 var std.dev coef.var skewness   
## 5.374613e-01 1.763928e+01 4.199914e+00 4.015246e-01 -1.100231e-01   
## skew.2SE kurtosis kurt.2SE normtest.W normtest.p   
## -3.479175e-01 -6.324080e-01 -1.003993e+00 9.804352e-01 2.327734e-03

shapiro.test(my\_data\_freshmen\_1$Score)

##   
## Shapiro-Wilk normality test  
##   
## data: my\_data\_freshmen\_1$Score  
## W = 0.98044, p-value = 0.002328

#ks.test(my\_data\_freshmen\_1$Score,"pnorm",mean(my\_data\_freshmen\_1$Score),sd(my\_data\_freshmen\_1$Score))  
  
  
  
plot(density(my\_data\_not\_freshmen\_0$Score),main="Estimated pdf nonfreshmen score")



stat.desc(my\_data\_not\_freshmen\_0$Score,basic=T,norm=TRUE)

## nbr.val nbr.null nbr.na min max   
## 34.000000000 0.000000000 0.000000000 5.000000000 19.000000000   
## range sum median mean SE.mean   
## 14.000000000 356.000000000 10.000000000 10.470588235 0.651660835   
## CI.mean.0.95 var std.dev coef.var skewness   
## 1.325813937 14.438502674 3.799802978 0.362902532 0.713960812   
## skew.2SE kurtosis kurt.2SE normtest.W normtest.p   
## 0.885690933 -0.378336090 -0.240092038 0.908690522 0.007844936

shapiro.test(my\_data\_not\_freshmen\_0$Score)

##   
## Shapiro-Wilk normality test  
##   
## data: my\_data\_not\_freshmen\_0$Score  
## W = 0.90869, p-value = 0.007845

#ks.test(my\_data\_not\_freshmen\_0$Score,"pnorm",mean(my\_data\_not\_freshmen\_0$Score),sd(my\_data\_not\_freshmen\_0$Score))  
  
# here p-value is is less than 0.05 so null hypothesis that data are normally distributed is rejected.