Exploring Assumptions 2

Code

Hide

```
library(ggplot2)
library(car)
library(pastecs)
library(psych)
library(cowplot)
library(gridExtra)
data<-read.delim('/home/atrides/Desktop/Applied-Statistics-with-R-master/statistics_with_R/05/Data_Files/RExam.dat',header=TRUE)
head(data, 15)
```

	exam <int>	computer <int>	lectures <dbl>	numeracy <int>	uni <int>
1	18	54	75.0	7	0
2	30	47	8.5	1	0
3	40	58	69.5	6	0
4	30	37	67.0	6	0
5	40	53	44.5	2	0
6	15	48	76.5	8	0
7	36	49	70.0	3	0
8	40	49	18.5	7	0
9	63	45	43.5	4	0
10	31	62	100.0	6	0

1-10 of 15 rows Previous **1** 2 Next

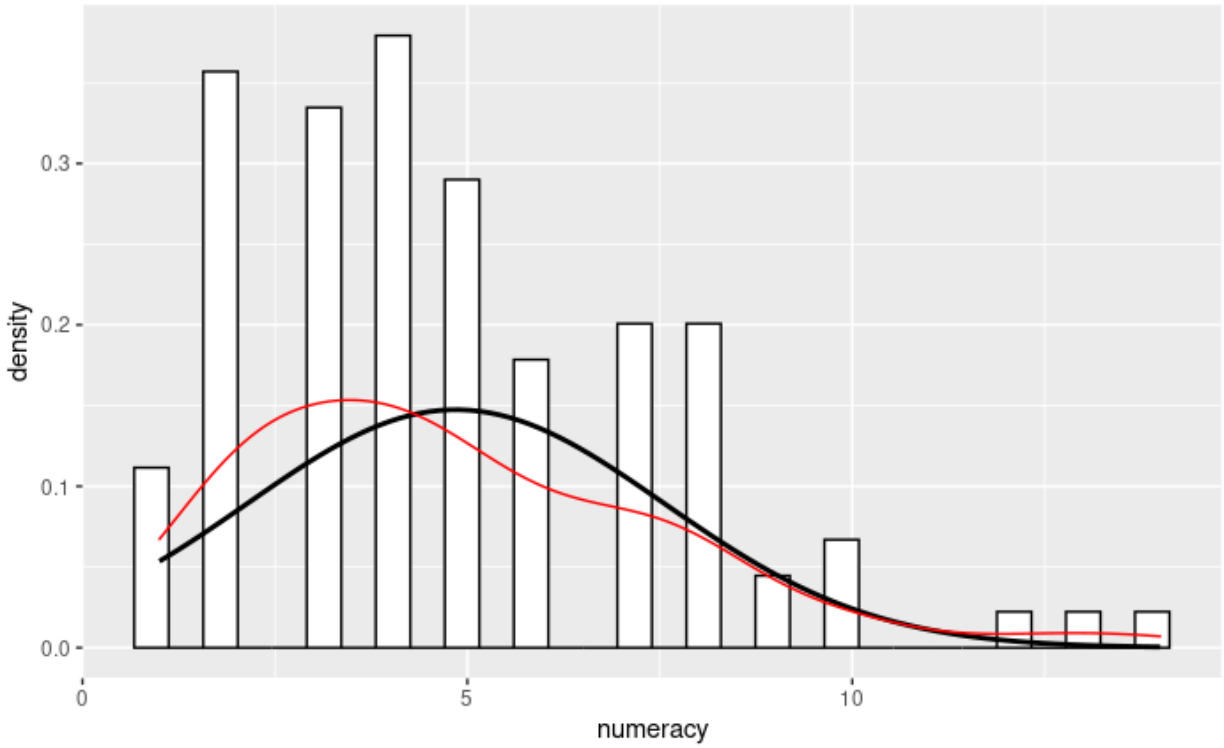
Hide

```
data$uni <-factor(data$uni,levels = c(0:1), labels = c('DunceTown','Sussex'))

is.factor(data$uni)
```

```
[1] TRUE
```

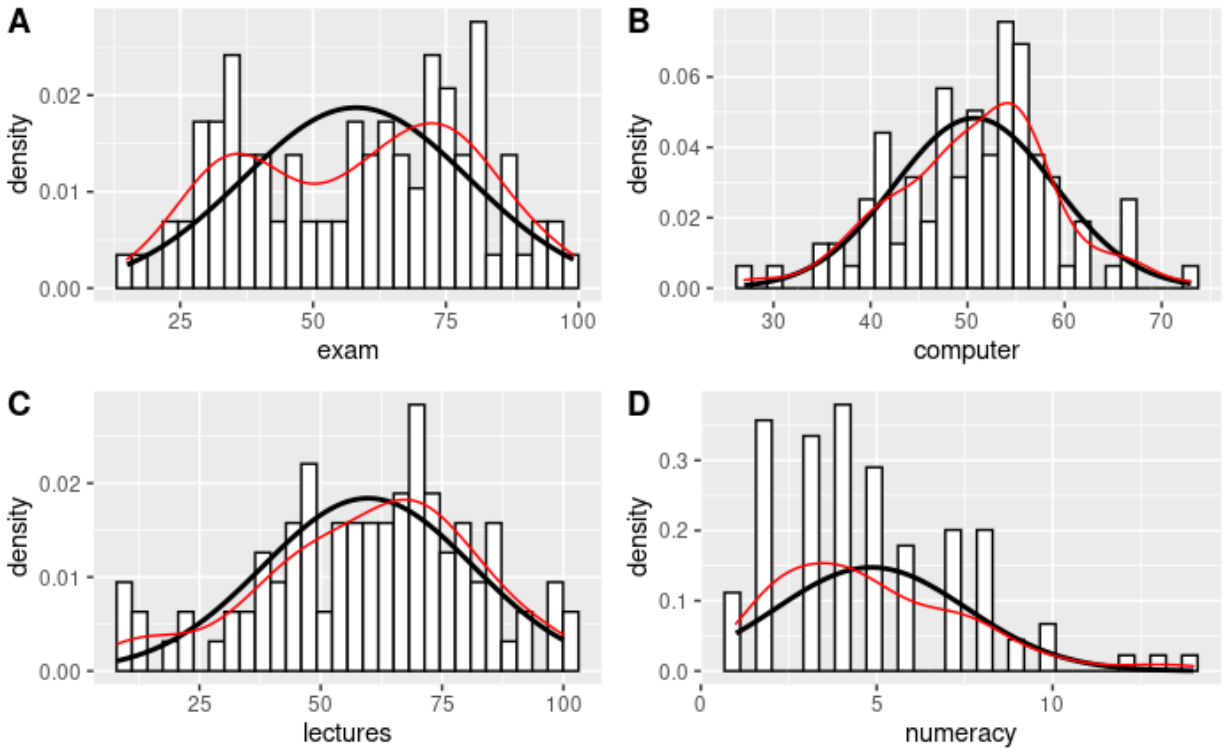
Hide

```
# exam marks histogram
examplot<-ggplot(data, aes(exam))
examplot<-examplot+geom_histogram(colour='black',bins=30,fill='white',aes(y
=..density..))
examplot<-examplot+
  stat_function(fun=dnorm, args=list(mean=mean(data$exam,na.rm = TRUE),sd=sd
(data$exam,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')



# computer literacy
complot<-ggplot(data, aes(computer))
complot<-complot+geom_histogram(colour='black',bins=30,fill='white',aes(y=..
density..))
complot<-complot+
  stat_function(fun=dnorm, args=list(mean=mean(data$computer,na.rm = TRUE),s
d=sd(data$computer,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')



# lectures attended
lectureplot<-ggplot(data, aes(lectures))
lectureplot<-lectureplot+geom_histogram(colour='black',bins=30,fill='white',
aes(y=..density..))
lectureplot<-lectureplot+
  stat_function(fun=dnorm, args=list(mean=mean(data$lectures,na.rm = TRUE),s
d=sd(data$lectures,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')



# numeracy
numeracyplot<-ggplot(data, aes(numeracy))
numeracyplot<-numeracyplot+geom_histogram(colour='black',bins=30,fill='white
',aes(y=..density..))
numeracyplot<-numeracyplot+
  stat_function(fun=dnorm, args=list(mean=mean(data$numeracy,na.rm = TRUE),s
d=sd(data$numeracy,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')
numeracyplot
```



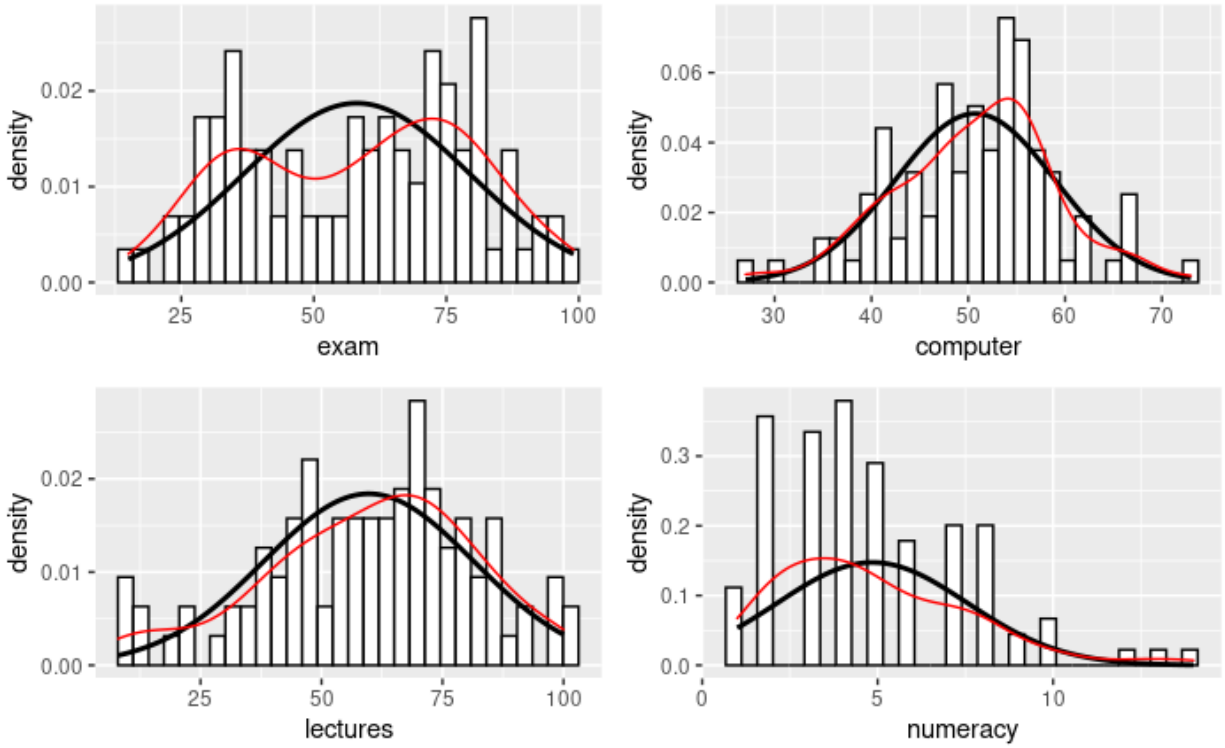
Hide

```
# using cowplot library, combined multiple plot
plot_grid(examplot, complot, lectureplot, numeracyplot, labels = "AUTO")
```



Hide

```
# using gridExtra library, compined multiple plots
grid.arrange(examplot, complot,lectureplot,numeracyplot, ncol=2,nrow=2)
```



Hide

```
stat.desc(data[,cbind('exam', 'computer','lectures','numeracy')], norm=TRUE,
basic=FALSE)
```

	exam <dbl>	computer <dbl>	lectures <dbl>	numeracy <dbl>
median	60.000000000	51.5000000	62.00000000	4.000000e+00
mean	58.100000000	50.7100000	59.76500000	4.850000e+00
SE.mean	2.131557026	0.8260035	2.16847774	2.705681e-01
CI.mean.0.95	4.229471584	1.6389702	4.30273029	5.368657e-01
var	454.353535354	68.2281818	470.22957071	7.320707e+00
std.dev	21.315570256	8.2600352	21.68477740	2.705681e+00
coef.var	0.366877285	0.1628877	0.36283406	5.578723e-01
skewness	-0.103804261	-0.1690671	-0.40984494	9.327151e-01
skew.2SE	-0.215022696	-0.3502098	-0.84896287	1.932049e+00
kurtosis	-1.147658459	0.2208250	-0.28463568	7.634927e-01

1-10 of 13 rows Previous **1** 2 Next

Hide

```
# Running the analysis on different groups, i.e as pandas groupby

by(data[,cbind('exam', 'computer', 'lectures','numeracy')], data$uni, describe)
```

```
data$uni: DunceTown
```

	vars <dbl>	n <dbl>	mean <dbl>	sd <dbl>	median <dbl>	trimmed <dbl>	mad <dbl>	min <dbl>	max <dbl>
exam	1	50	40.18	12.59	38.0	39.85	12.60	15	66
computer	2	50	50.26	8.07	49.0	50.05	8.90	35	67
lectures	3	50	56.26	23.77	60.5	56.90	20.02	8	100
numeracy	4	50	4.12	2.07	4.0	4.00	2.22	1	9

4 rows | 1-10 of 13 columns

```
------------------------------------------------------------------------
----------------------
data$uni: Sussex
```

	vars <dbl>	n <dbl>	mean <dbl>	sd <dbl>	median <dbl>	trimmed <dbl>	mad <dbl>	min <dbl>	max <dbl>
exam	1	50	76.02	10.21	75.00	75.70	8.90	56.0	99
computer	2	50	51.16	8.51	54.00	51.62	5.93	27.0	73
lectures	3	50	63.27	18.97	65.75	63.99	20.76	12.5	100
numeracy	4	50	5.58	3.07	5.00	5.28	2.97	1.0	14

4 rows | 1-10 of 13 columns

Hide

```
# or also,
by(data[,cbind('exam', 'computer', 'lectures','numeracy')], data$uni, stat.desc, basic=FALSE, norm=TRUE)
```

```
data$uni: DunceTown
                    exam   computer    lectures    numeracy
median        38.0000000 49.0000000  60.5000000  4.00000000
mean          40.1800000 50.2600000  56.2600000  4.12000000
SE.mean        1.7803210  1.1410021   3.3619491  0.29226770
CI.mean.0.95   3.5776890  2.2929295   6.7560897  0.58733393
var          158.4771429 65.0942857 565.1351020  4.27102041
std.dev       12.5887705  8.0681030  23.7725704  2.06664472
coef.var       0.3133094  0.1605273   0.4225484  0.50161280
skewness       0.2906760  0.2121230  -0.2904291  0.48165960
skew.2SE       0.4317816  0.3150960  -0.4314149  0.71547621
kurtosis      -0.7230849 -0.6779460  -0.5634849 -0.65166313
kurt.2SE      -0.5462122 -0.5121147  -0.4256518 -0.49226083
normtest.W     0.9721662  0.9776351   0.9697413  0.94081692
normtest.p     0.2828984  0.4571105   0.2259072  0.01451518
------------------------------------------------------------------------------
----------------------
data$uni: Sussex
                    exam    computer    lectures     numeracy
median        75.0000000 54.00000000  65.7500000  5.000000000
mean          76.0200000 51.16000000  63.2700000  5.580000000
SE.mean        1.4432079  1.20284018   2.6827191  0.434332704
CI.mean.0.95   2.9002348  2.41719783   5.3911258  0.872824247
var          104.1424490 72.34122449 359.8490816  9.432244898
std.dev       10.2050208  8.50536445  18.9696885  3.071196004
coef.var       0.1342413  0.16625028   0.2998212  0.550393549
skewness       0.2559866 -0.50635339  -0.3429407  0.746369109
skew.2SE       0.3802527 -0.75215735  -0.5094177  1.108686183
kurtosis      -0.4609644  0.96404781  -0.4233827 -0.006440059
kurt.2SE      -0.3482086  0.72823358  -0.3198197 -0.004864766
normtest.W     0.9837115  0.94392221   0.9817164  0.932346126
normtest.p     0.7151182  0.01931372   0.6262649  0.006786803
```

Hide

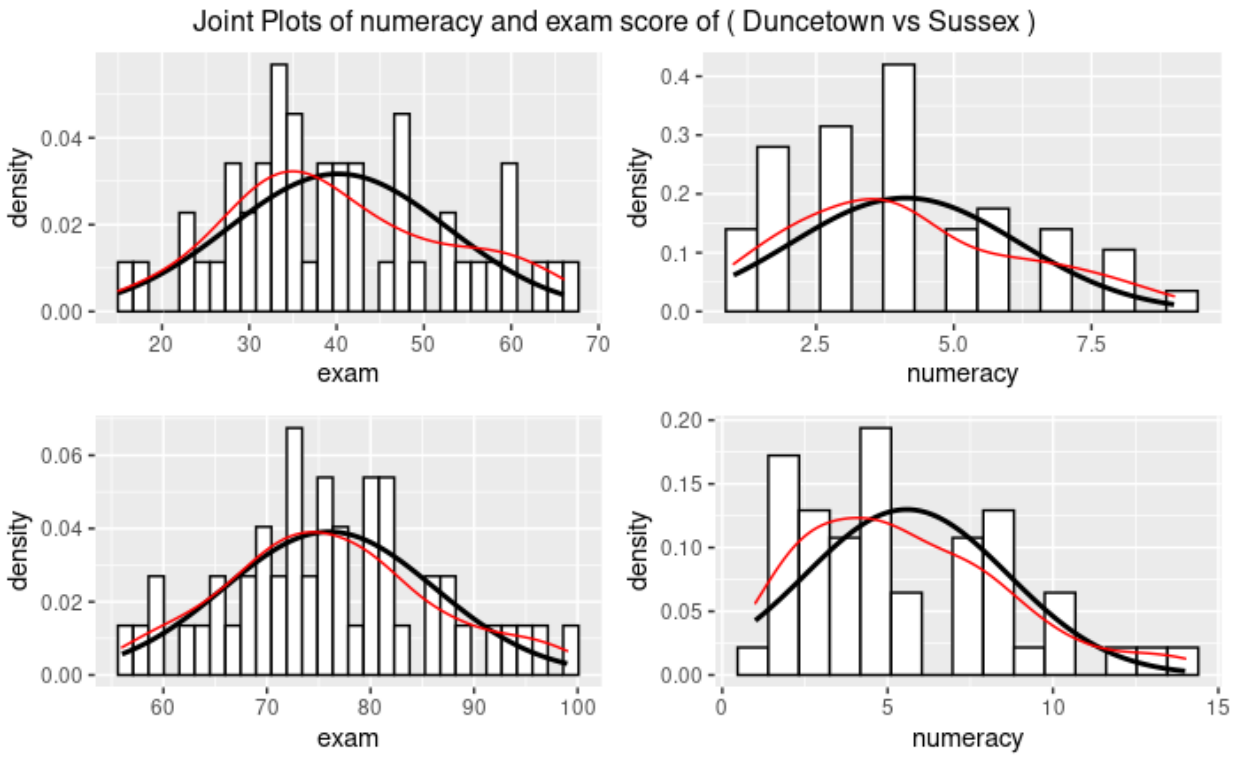
```
# dividing data as per uni
dunceData<-subset(data, data$uni=='DunceTown')
Sussex<-subset(data, data$uni=='Sussex')
```

Hide

```
# DunceTown PLot
# exam marks histogram
examplot1<-ggplot(dunceData, aes(exam))
examplot1<-examplot1+geom_histogram(colour='black',bins=30,fill='white',aes
(y=..density..))
examplot1<-examplot1+
  stat_function(fun=dnorm, args=list(mean=mean(dunceData$exam,na.rm = TRUE),
sd=sd(dunceData$exam,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')


# numeracy
numeracyplot1<-ggplot(dunceData, aes(numeracy))
numeracyplot1<-numeracyplot1+geom_histogram(colour='black',bins=15,fill='whi
te',aes(y=..density..))
numeracyplot1<-numeracyplot1+
  stat_function(fun=dnorm, args=list(mean=mean(dunceData$numeracy,na.rm = TR
UE),sd=sd(dunceData$numeracy,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')


# Sussex PLot
# exam marks histogram
examplot2<-ggplot(Sussex, aes(exam))
examplot2<-examplot2+geom_histogram(colour='black',bins=30,fill='white',aes
(y=..density..))
examplot2<-examplot2+
  stat_function(fun=dnorm, args=list(mean=mean(Sussex$exam,na.rm = TRUE),sd=
sd(Sussex$exam,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')


# numeracy
numeracyplot2<-ggplot(Sussex, aes(numeracy))
numeracyplot2<-numeracyplot2+geom_histogram(colour='black',bins=15,fill='whi
te',aes(y=..density..))
numeracyplot2<-numeracyplot2+
  stat_function(fun=dnorm, args=list(mean=mean(Sussex$numeracy,na.rm = TRU
E),sd=sd(Sussex$numeracy,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')
```

Hide

```
# plotting above things
grid.arrange(examplot1,numeracyplot1,examplot2,numeracyplot2,ncol=2, nrow=2,
top='Joint Plots of numeracy and exam score of ( Duncetown vs Sussex )')
```

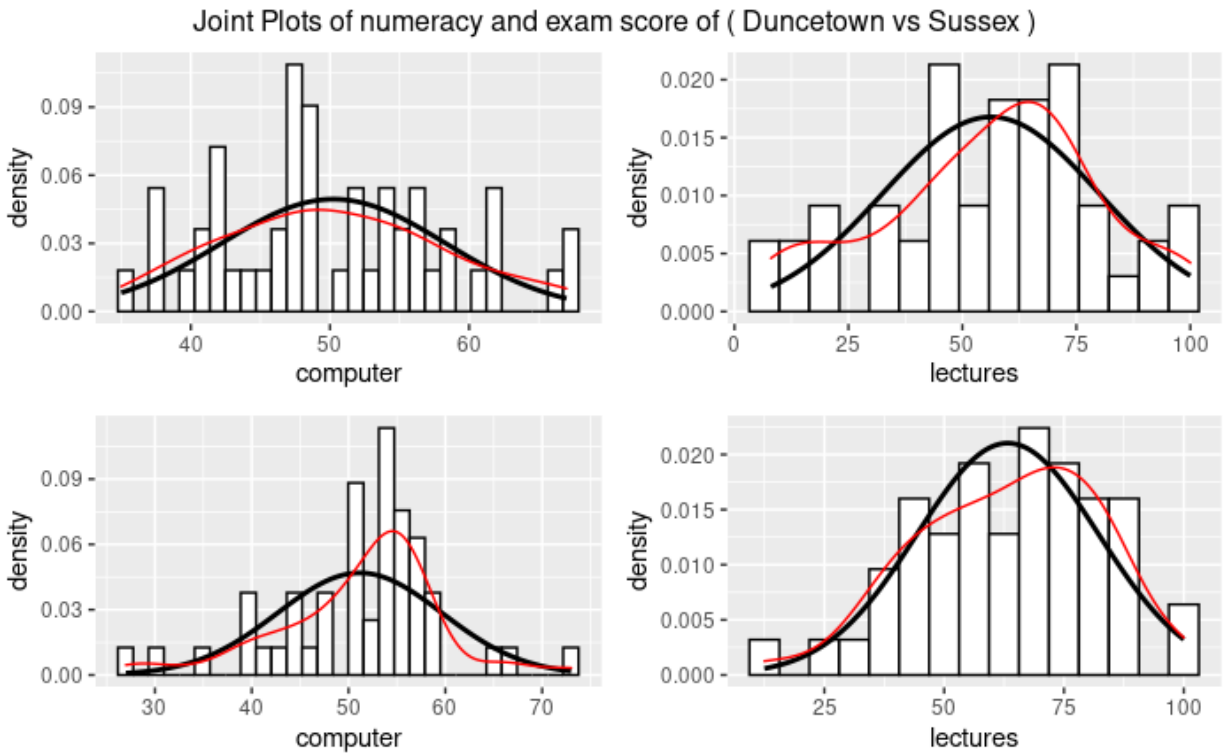

Joint Plots of numeracy and exam score of (Duncetown vs Sussex)
density
exam
numeracy
0.00
0.02
0.04
0.06
0.0
0.1
0.2
0.3
0.4
0.05
0.10
0.15
0.20
20
30
40
50
60
70
2.5
5.0
7.5
0
5
10
15
60
70
80
90
100

Hide

```
# DunceTown PLot
# computer marks histogram
computerplot1<-ggplot(dunceData, aes(computer))
computerplot1<-computerplot1+geom_histogram(colour='black',bins=30,fill='whi
te',aes(y=..density..))
computerplot1<-computerplot1+
  stat_function(fun=dnorm, args=list(mean=mean(dunceData$computer,na.rm = TR
UE),sd=sd(dunceData$computer,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')


# lectures
lecplot1<-ggplot(dunceData, aes(lectures))
lecplot1<-lecplot1+geom_histogram(colour='black',bins=15,fill='white',aes(y
=..density..))
lecplot1<-lecplot1+
  stat_function(fun=dnorm, args=list(mean=mean(dunceData$lectures,na.rm = TR
UE),sd=sd(dunceData$lectures,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')


# Sussex PLot
# exam marks histogram
computerplot2<-ggplot(Sussex, aes(computer))
computerplot2<-computerplot2+geom_histogram(colour='black',bins=30,fill='whi
te',aes(y=..density..))
computerplot2<-computerplot2+
  stat_function(fun=dnorm, args=list(mean=mean(Sussex$computer,na.rm = TRU
E),sd=sd(Sussex$computer,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')


# lectures
lecplot2<-ggplot(Sussex, aes(lectures))
lecplot2<-lecplot2+geom_histogram(colour='black',bins=15,fill='white',aes(y
=..density..))
lecplot2<-lecplot2+
  stat_function(fun=dnorm, args=list(mean=mean(Sussex$lectures,na.rm = TRU
E),sd=sd(Sussex$lectures,na.rm = TRUE)), colour='black', size=1)+
  geom_density(colour='red')
```

Hide

```
# plotting above things
grid.arrange(computerplot1,lecplot1,computerplot2,lecplot2,ncol=2, nrow=2, t
op='Joint Plots of numeracy and exam score of ( Duncetown vs Sussex )')
```



Hide

```
# Doing Statistical tests for normality assumptions

# Shapiro-wilk test whole Data
print(shapiro.test(data$exam)) # non -normal
```

```
    Shapiro-Wilk normality test

data:  data$exam
W = 0.96131, p-value = 0.004991
```

Hide

```
print(shapiro.test(data$computer)) # normal
```

```
    Shapiro-Wilk normality test

data:  data$computer
W = 0.98706, p-value = 0.4413
```

Hide

```
print(shapiro.test(data$lectures)) # normal
```

```

    Shapiro-Wilk normality test

data:  data$lectures
W = 0.97698, p-value = 0.07712
```

Hide

```
print(shapiro.test(data$numeracy)) # non-normal
```

```

    Shapiro-Wilk normality test

data:  data$numeracy
W = 0.92439, p-value = 2.424e-05
```

Hide

```

# Shapiro-wilk test Duncetown Data
print(shapiro.test(dunceData$exam)) # normal
```

```

    Shapiro-Wilk normality test

data:  dunceData$exam
W = 0.97217, p-value = 0.2829
```

Hide

```
print(shapiro.test(dunceData$computer)) # normal
```

```

    Shapiro-Wilk normality test

data:  dunceData$computer
W = 0.97764, p-value = 0.4571
```

Hide

```
print(shapiro.test(dunceData$lectures)) # normal
```

```
    Shapiro-Wilk normality test

data:  dunceData$lectures
W = 0.96974, p-value = 0.2259
```

Hide

```
print(shapiro.test(dunceData$numeracy)) # non-normal
```

```
    Shapiro-Wilk normality test

data:  dunceData$numeracy
W = 0.94082, p-value = 0.01452
```

Hide

```
# Shapiro-wilk test Sussex Data
print(shapiro.test(Sussex$exam)) # normal
```

```
    Shapiro-Wilk normality test

data:  Sussex$exam
W = 0.98371, p-value = 0.7151
```

Hide

```
print(shapiro.test(Sussex$computer)) # non-normal
```

```
    Shapiro-Wilk normality test

data:  Sussex$computer
W = 0.94392, p-value = 0.01931
```

Hide

```
print(shapiro.test(Sussex$lectures)) # normal
```

```
    Shapiro-Wilk normality test

data:  Sussex$lectures
W = 0.98172, p-value = 0.6263
```

Hide

```
print(shapiro.test(Sussex$numeracy)) # non-normal
```

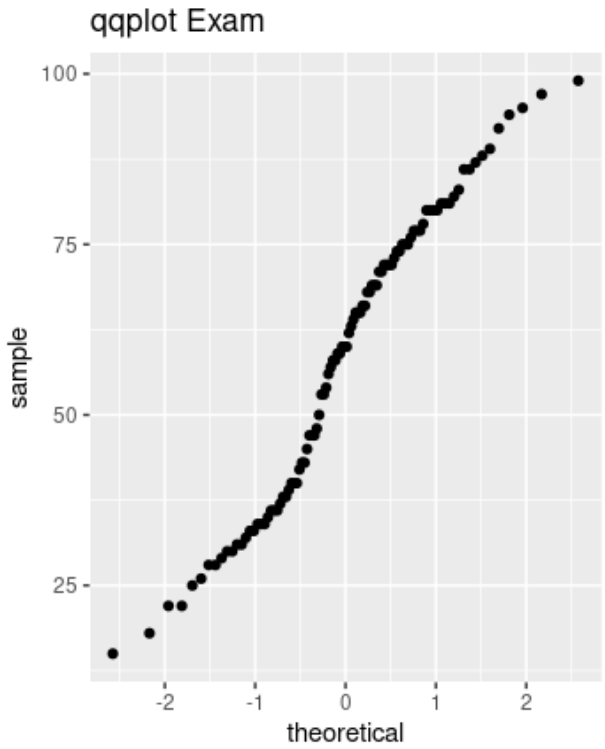
```
    Shapiro-Wilk normality test

data:  Sussex$numeracy
W = 0.93235, p-value = 0.006787
```

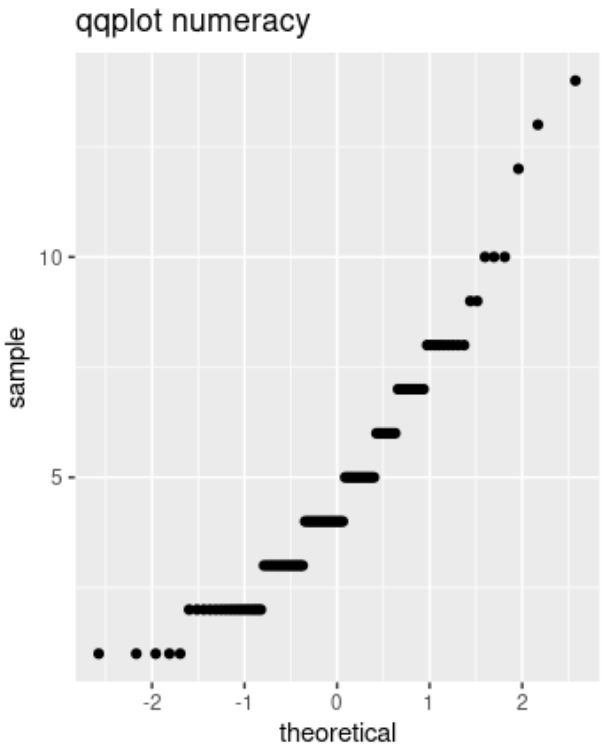
Hide

```
# qqplots
qqplot_exam<- ggplot(data, aes(sample=exam))
qqplot_exam<-qqplot_exam+stat_qq()
qqplot_exam<-qqplot_exam+ggtitle('qqplot Exam')


qqplot_numeracy<- ggplot(data, aes(sample=numeracy))
qqplot_numeracy<-qqplot_numeracy+stat_qq()
qqplot_numeracy<-qqplot_numeracy+ggtitle('qqplot numeracy')
```

Hide

```
grid.arrange(qqplot_exam,qqplot_numeracy,ncol=2)
```





Hide

```
# Doing Statistical tests for Homogeneity of variance assumption
# leveneTest() from car package

print(leveneTest(data$exam, data$uni, center = median))  # assumptions met
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  2.0886 0.1516
      98
```

Hide

```
print(leveneTest(data$numeracy, data$uni, center = median))  # assumptions n
ot  met
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value  Pr(>F)
group  1   5.366 0.02262 *
      98
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```