Lab 1 Homework Assignment

Ryan Avery and Team Member2

April 04, 2018

Probabilities should not exceed 1. θ include - controls if chunk is knitted results - controls if return vals printed echo - controls if code is shown eval - controls if code is run cache - speeds creation

Problem 1

1 (a)

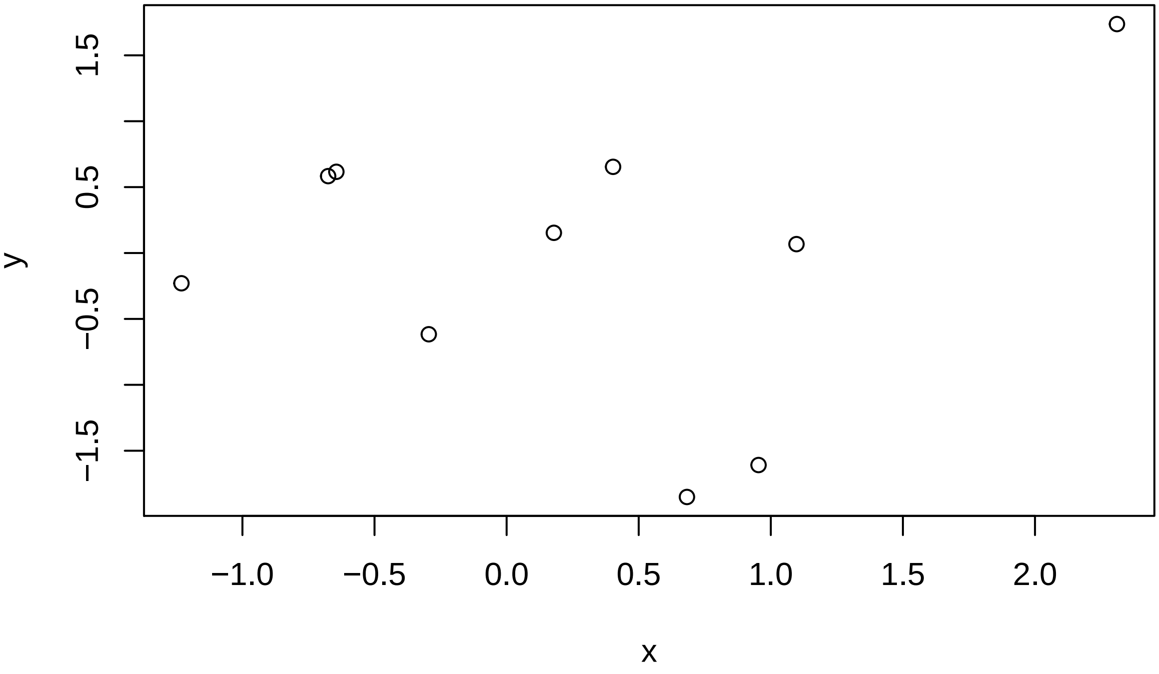
```
x = rnorm(10)
y = rnorm(10)
x
```

```
##  [1] -0.6444091  2.3102155 -0.6755040  0.6825787  0.9535888  0.4029056
##  [7]  0.1788246 -1.2308606  1.0971769 -0.2946461
```

```
y
```

```
##  [1]  0.61628424  1.73701684  0.58340869 -1.85079313 -1.60855875
##  [6]  0.65393962  0.15323370 -0.22955652  0.06746623 -0.61622723
```

```
plot(x,y)
```



1 (b)

1. Write an R function that returns the following dispersion measures:

- Estimator of standard deviation (SD):

$$\text{SD} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

```r
s.d <- function(x){
    n <- length(x) # Sample size
    s2 <- sum((x - mean(x))^2)/(n-1) # sample variance
    s.d <- sqrt(s2) # sample standard deviation
    return(s.d)
}
```

- Estimator of mean absolute deviation (MAD):

$$\text{MAD} = \frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}|$$

```r
mean.abs.d <- function(x){
    n <- length(x) # Sample size
    m <- sum(abs(x - mean(x)))/n # mean average deviation
    return(m)
}
```

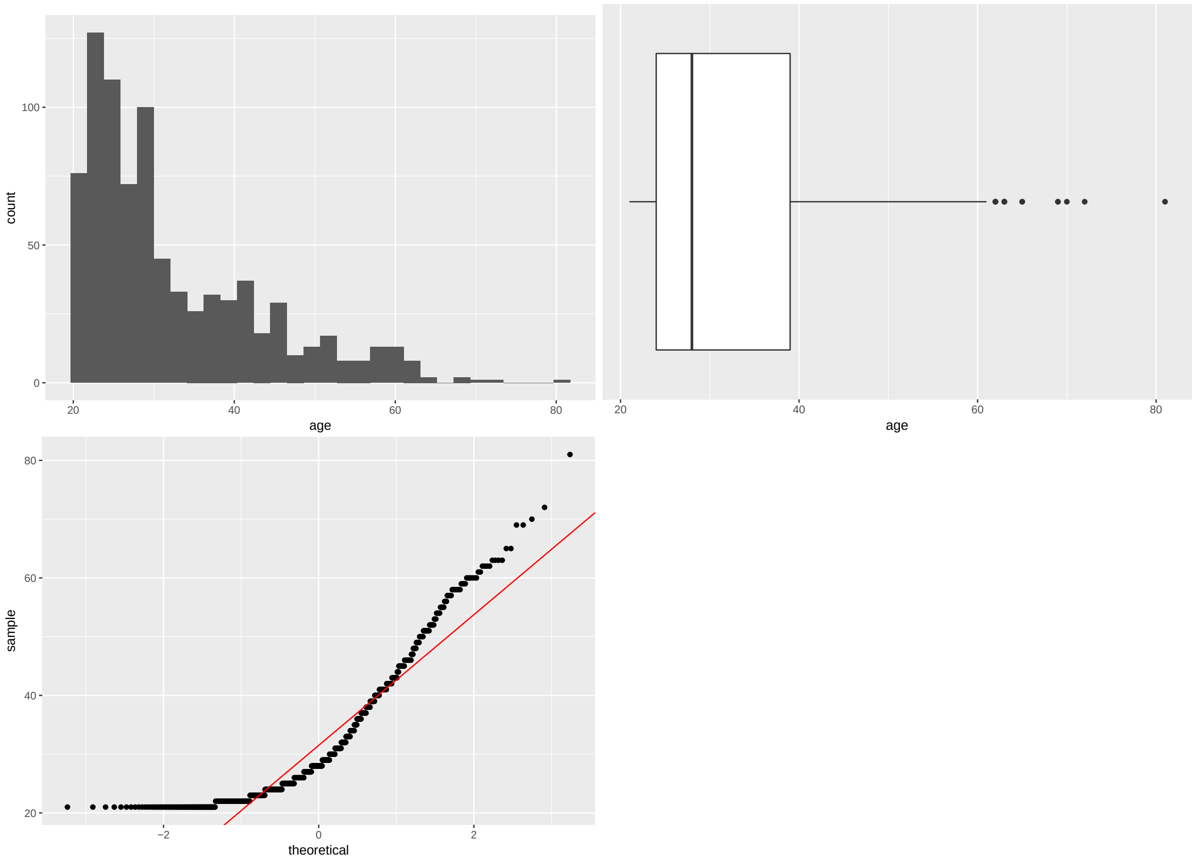
2. Construct box-plots, histograms, QQ-plots and kernel density estimates for these variables. Comment on features such as the distribution and outliers in these plots.
When asked to construct a graph, you should always precede your graph by the R command/function that generated it properly annotated.

```r
library(MASS)
pima2 = rbind(Pima.tr, Pima.tr2, Pima.te)

x = pima2$age
var.name = 'age'
```

```r
library(ggplot2)

ggplot(pima2, aes(x=age)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```r
ggplot(pima2, aes(x = factor(0), y = age)) + geom_boxplot() + xlab("") +
    scale_x_discrete(breaks = NULL) + coord_flip()

y     <- quantile(pima2$age, c(0.25, 0.75)) # Find the 1st and 3rd quartiles
x     <- qnorm( c(0.25, 0.75))         # Find the matching normal values on the x-axis
slope <- diff(y) / diff(x)             # Compute the line slope
int   <- y[1] - slope * x[1]           # Compute the line intercept

ggplot(pima2, aes(sample=age)) + stat_qq() +
    geom_abline(intercept=int, slope=slope, color='red')
```


count
100
50
0
20
40
60
80
age
20
40
60
80
age
sample
80
60
40
20
−2
0
2
theoretical