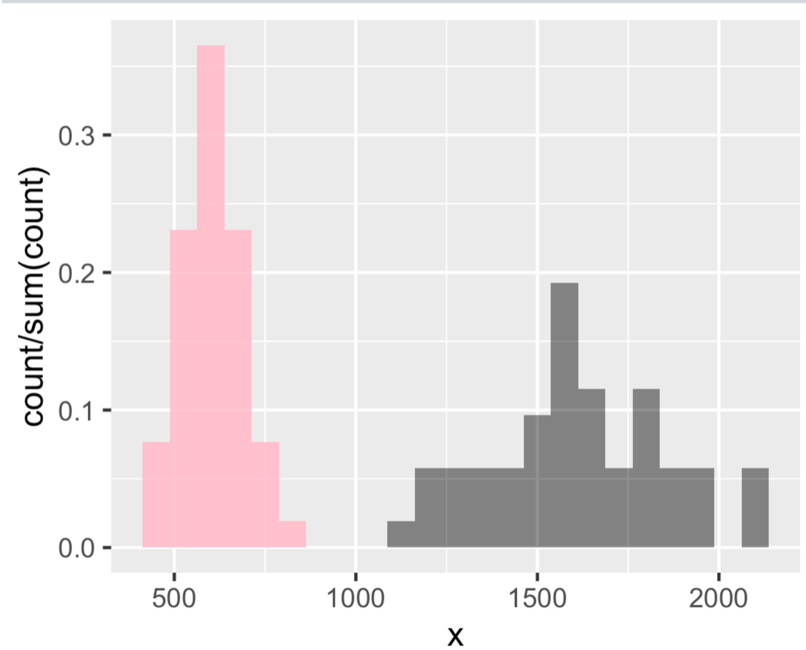
```r
# AD 616 Assignment 3
# Tiange Chang
library(survival)
library(extraDistr)
library(tidyverse)
library(MASS)
library(vcd)
library(fitdistrplus)
library(Rcpp)
library(matrixcalc)
library(MultiRNG)
library(rio)
```

```r
# 1. a
# install.packages("readxl")
library(readxl)

setwd('/Users/t/Desktop/AD 616/Assignment/3')

df <- read_excel("metad616_Assignment3_Problem1_Data.xlsx")
View(df)
summary(df)

# Distribution
x <- df$`Wait Time`
y <- df$`Inexperienced Agents`

ggplot() + geom_histogram(aes(x=x, y=..count../sum(..count..)),
                          fill='pink', alpha=.85, binwidth=75) +
           geom_histogram(aes(x=y, y=..count../sum(..count..)),
                          fill='black', alpha=.5, binwidth=75)

# Wait time model
wt <- fitdist(df$'Wait Time', distr='norm')
wt
# The mean of the wait time model is 606.75, the sd of the wait time model
# is 82.73913

# Inexperienced Agents model
ia <- fitdist(df$'Inexperienced Agents', distr='norm')
ia
# The mean of the inexperienced agents model is 1611.2308, the sd of the
# inexperienced agents model is 247.5434.

# goodness-of-fit statistics
gofstat(wt)
gofstat(ia)
```




```
> gofstat(wt)
Goodness-of-fit statistics
                             1-mle-norm
Kolmogorov-Smirnov statistic 0.05541960
Cramer-von Mises statistic   0.01754825
Anderson-Darling statistic   0.13712117

Goodness-of-fit criteria
                               1-mle-norm
Akaike's Information Criterion   610.8016
Bayesian Information Criterion   614.7041
```

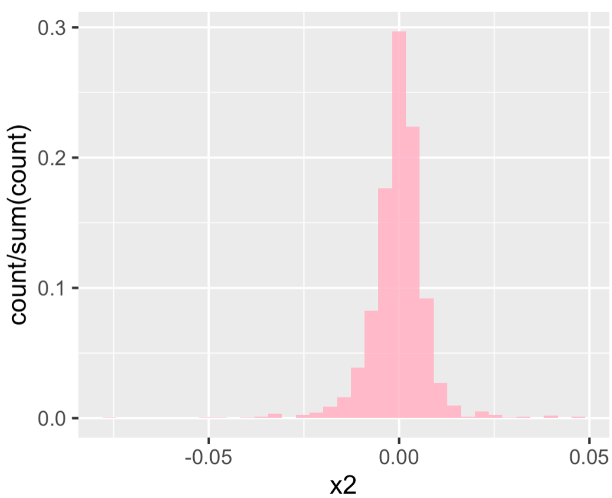
```
> gofstat(ia)
Goodness-of-fit statistics
                             1-mle-norm
Kolmogorov-Smirnov statistic 0.05064256
Cramer-von Mises statistic   0.01948200
Anderson-Darling statistic   0.14993809

Goodness-of-fit criteria
                               1-mle-norm
Akaike's Information Criterion   724.7745
Bayesian Information Criterion   728.6770
```

```r
# 1. b
ggplot(data=df) + geom_point(aes(x=x, y=y)) + theme_bw()
cor(df$'Wait Time', df$'Inexperienced Agents', method = 'spearman')
cormatrix <- matrix(c(1, 0.66, 0.66, 1), 2, 2)
is.positive.semi.definite(cormatrix)

# The Spearman correlation takes the intermediate step of calculating the rank
# of each observation in the data frame, then computing the Pearson correlation
# of those ranks. If these ranks match up perfectly, you could observe a
# Spearman correlation of 1, when the Pearson correlation would not achieve that
# result. In other words, Spearman correlation allows us to generate correlated
# values from different probability distributions. The correlation in this case
# is 0.6612163.

# 1. c
df_correlated <- data.frame(draw.d.variate.uniform(10000, 2, cormatrix))

df_correlated[,1] <- qnorm(df_correlated[,1], mean = wt$estimate[1],
                           sd = wt$estimate[2])

df_correlated[,2] <- qnorm(df_correlated[,2], mean = ia$estimate[1],
                           sd = ia$estimate[2])

cor(df_correlated[,1], df_correlated[,2])

total <- df_correlated[,1] + df_correlated[,2]

mean(total)
sd(total)
# The total mean and sd of the complaints by week is 2224.075 and 309.2281.
```

```r
# 1. d
mean(total) + qnorm(0.025) * sd(total)/(10000^0.5)
mean(total) + qnorm(0.975) * sd(total)/(10000^0.5)
# The 95% confidence interval is between 2218.014 - 2230.135.

# 1. e
sum((total < 600)/10000)
# The probability is less than 600 is 0.

# 1. f
sum((total > 2600)/10000)
# The probability is greater than 2600 is 0.1134.
```

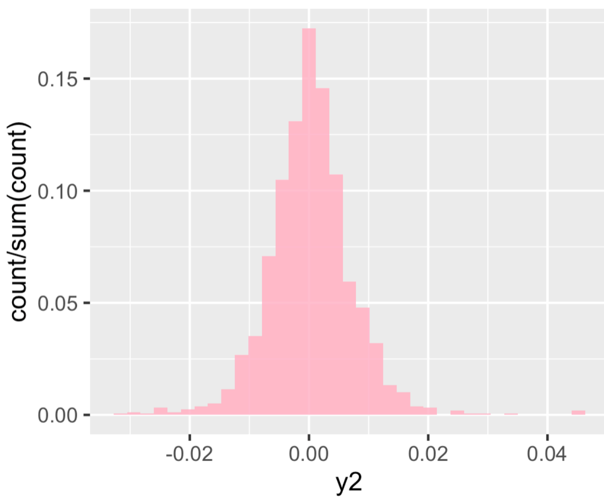
```r
# Re-run
wt2 <- rnorm(n = 10000, mean = wt$estimate[1], sd = wt$estimate[2])
ia2 <- rnorm(n = 10000, mean = ia$estimate[1], sd = ia$estimate[2])
total2 <- wt2 + ia2
mean(total2)
sd(total2)
# The total mean and sd of the complaints by week is 2216.624 and 260.8362.
mean(total2) + qnorm(0.025) * sd(total2)/(10000^0.5)
mean(total2) + qnorm(0.975) * sd(total2)/(10000^0.5)
# The 95% confidence interval is between 2211.512 - 2221.737.
sum((total2 < 600)/10000)
# The probability is less than 600 is 0.
sum((total2 > 2600)/10000)
# The probability is greater than 2600 is 0.0697.
# All the number decreased if we ignored the correlation.
```

```r
# 2. a
df2 <- read_excel('metad616_Assignment3_Problem2_Data.xlsx')
View(df2)

x2 <- df2$`MXP log-return`

ggplot() + geom_histogram(aes(x=x2, y=..count../sum(..count..)),
                          fill='pink', alpha=.95, bins=35)

mxp <- fitdist(df2$'MXP log-return', distr='norm')
gofstat(mxp)
```

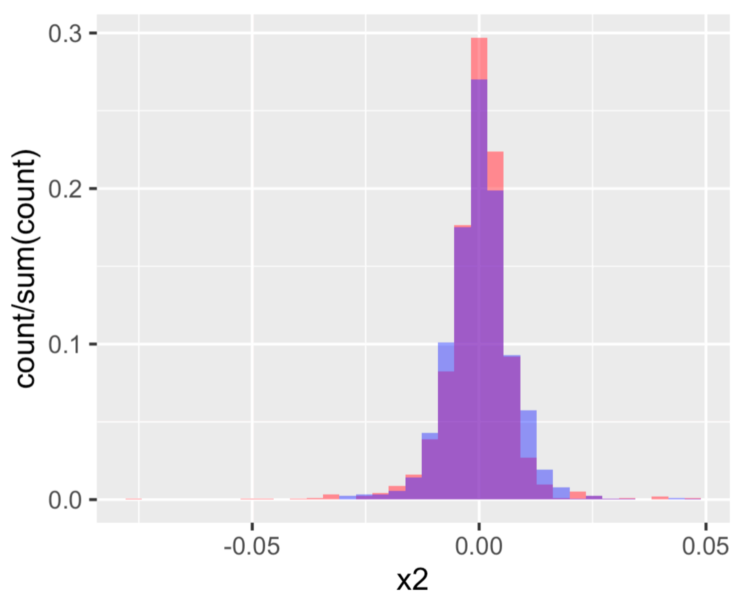



```
> gofstat(mxp)
Goodness-of-fit statistics
                             1-mle-norm
Kolmogorov-Smirnov statistic  0.09396289
Cramer-von Mises statistic    6.49311914
Anderson-Darling statistic   38.75094307

Goodness-of-fit criteria
                               1-mle-norm
Akaike's Information Criterion  -10810.11
Bayesian Information Criterion  -10799.40
```

```r
# 2. b
y2 <- df2$`YEN log-return`

ggplot() + geom_histogram(aes(x=y2, y=..count../sum(..count..)),
                          fill='pink', alpha=.95, bins=35)

yen <- fitdist(df2$'YEN log-return', distr='norm')
gofstat(yen)

ggplot() + geom_histogram(aes(x=x2, y=..count../sum(..count..)),
                          fill='red', alpha=.45, bins=35) +
           geom_histogram(aes(x=y2, y=..count../sum(..count..)),
                          fill='blue', alpha=.45, bins=35)

# I observed that two data sets are almost the same, but MXP has a higher return
# rate of around 0.3.
```




```
> gofstat(yen)
Goodness-of-fit statistics
                             1-mle-norm
Kolmogorov-Smirnov statistic 0.05226285
Cramer-von Mises statistic   1.62628221
Anderson-Darling statistic   9.52130149

Goodness-of-fit criteria
                               1-mle-norm
Akaike's Information Criterion  -11037.93
Bayesian Information Criterion  -11027.21
```



```r
# 3. a
time_rec <- runif(10000, 2, 5)
labor_rec <- 20*time_rec/60

time_rec
labor_rec

time <- c()
labor <- c()

for(i in 1:10000){
  u <- runif(1, 0, 1)
  if(u < 0.3){ # renew
    time_renew <- rgamma(1, shape = 3, scale = 10)
    labor_renew <- 30*time_renew/60
    time <- c(time, time_renew)
    labor <- c(labor, labor_renew)
  }
  else if (u < 0.8){ # register
    p <- runif(1, 0, 1)
    time_reg <- ifelse(p < 0.2, runif(1, 1, 15),
                       ifelse(p < 0.6, runif(1, 15, 25),
                              ifelse(p < 0.95, runif(1, 25, 30),
                                     runif(1, 30, 35))))
    labor_reg <- 40*time_reg/60
    time <- c(time, time_reg)
    labor <- c(labor, labor_reg)
  }
  else{ # leave
    time <- c(time, 0)
    labor <- c(labor, 0)
  }
}
total_time <- time_rec + time
total_labor <- labor_rec + labor

mean(total_time, na.rm = T)
sd(total_labor, na.rm=T)
# The mean and standard deviation of the service time per customer is 22.91402
# and 8.184378.
```

```r
# 3. b
sum(total_time < 8, na.rm = T)/10000
sum(total_time > 20, na.rm = T)/10000
# The probability that the service time per customer will be less than 8 minutes
# is 0.2228. The probability that the service time per customer will exceed 20
# minutes is 0.6044.

# 3. c
mean(total_labor, na.rm = T)
sd(total_labor, na.rm = T)
# The mean and standard deviation of the labor cost per customer to deliver this
# service is 12.62951 and 8.184378.
```