STAT 231 Assignment 4

The purpose of this assignment is to use the software R to fit Gaussian models, do model checking, test hypotheses, and construct confidence intervals.

The code for this assignment is posted both as a text file called RCodeAssignment4.txt and an R file called RCodeAssignment4R.R which are posted in the Assignment 4 folder in the Assignments folder under Content on Learn.

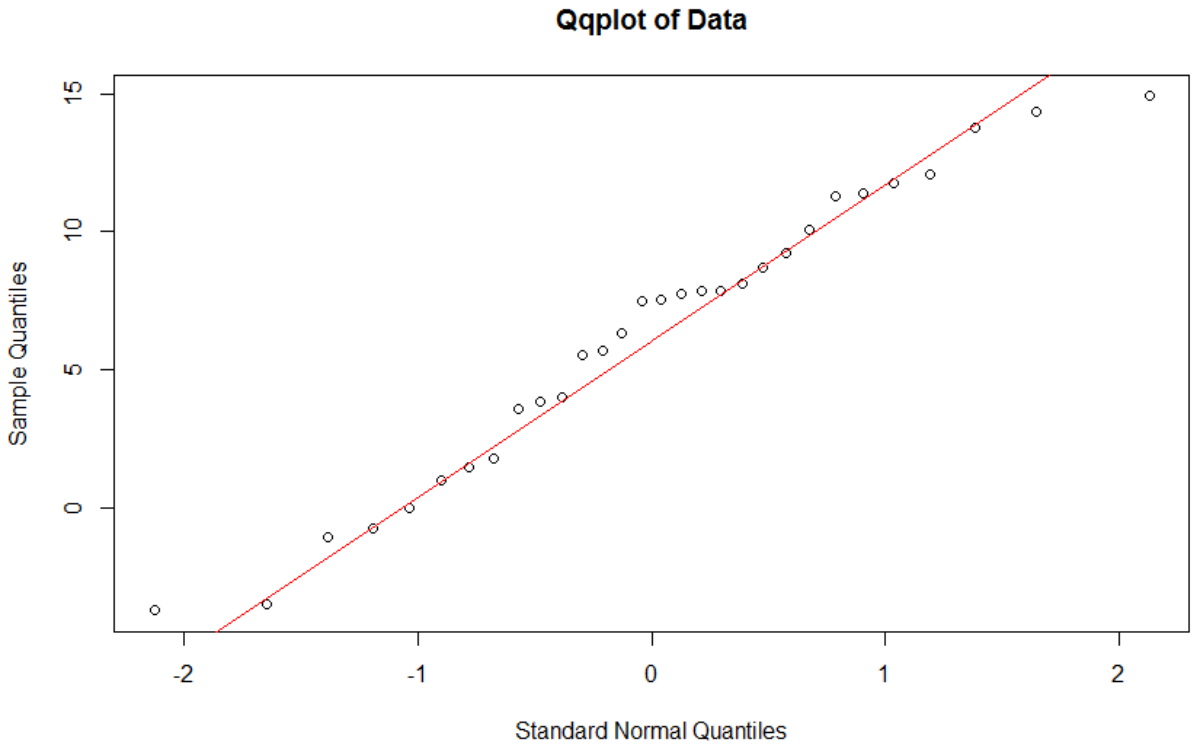
Problem 1: Run the following R code.

```
################################################################################
# Problem 1: One sample Gaussian model
id<-20456458
set.seed(id)
model<-sample(c(1:3),1)
cat("Model = ", model)
# Data are randomly generated from Gaussian distribution (model=1), Gamma distribution (model=2), or
# Poisson distribution (model=3)
if (model==1) {
mu<-id-10*trunc(id/10)                          # mu = last digit of ID
sigma<-max(1,trunc(id/10)-10*trunc(id/100))   # sig = second last digit of ID unless last digit is zero
cat("mu = ", mu, ", sigma = ", sigma)      # display values of mu and sigma
y<-sort(round(rnorm(30,mu,sigma),digits=2))  # 30 observations from G(mu,sig)
} else if (model==2) {
mu<-max(1,id-10*trunc(id/10))                # mu = last digit of ID unless it is zero
sigma<-(3*mu^2)^0.5
cat("mu = ", mu, ", sigma = ", sigma)      # display values of mu and sigma
y<-sort(round(rgamma(30,3,1/mu),digits=2))   # 30 observations from Gamma(3,1/mu)
} else if (model==3) {
mu<-max(1,id-10*trunc(id/10))                # mu = last digit of ID unless it is zero
sigma<-mu^0.5
cat("mu = ", mu, ", sigma = ", sigma)      # display values of mu and sigma
y<- sort(round(rpois(30,mu),digits=2))   # 30 observations from Poisson(mu)
}
# Check Gaussian assumption using qqplot
qqnorm(y,xlab="Standard Normal Quantiles",main="Qqplot of Data")
qqline(y,col="red",lwd=1.5)  # add line for comparison
mu0<-mu+1
cat("mu0 = ", mu0)      # display value of mu0
# test hypothesis mu=mu0 and obtain 95% confidence interval for mu
t.test(y,mu=mu0,conf.level=0.95)
#
```

```
sigma0<-sigma+2
cat("sigma0 = ", sigma0)      # display value of sigma0
# test hypothesis sigma=sigma0 and obtain 95% confidence interval for sigma
df<-length(y)-1     # degrees of freedom
s2<-var(y)
cat("sample variance = ", s2)         # display sample variance
chitest<-s2*df/sigma0^2
q<-pchisq(chitest,df)
cat("p-value for testing sigma=sigma0: ", min(2*q,2*(1-q)))
p<-0.95               # p=0.95 for 95% confidence interval
a<-qchisq((1-p)/2,df) # lower value from Chi-squared dist'n
b<-qchisq((1+p)/2,df) # upper value from Chi-squared dist'n
cat("95% confidence interval for sigma squared:  ",c(s2*df/b,s2*df/a))
cat("95% confidence interval for sigma: ",c(sqrt(s2*df/b),sqrt(s2*df/a)))
##############################################################################################
```

Verify that you obtain the following output:

```
> cat("Model = ", model)
Model =  1

mu =  8 , sigma =  5
```



```
> cat("mu0 = ", mu0)       # display value of mu0
mu0 =  9

> t.test(y,mu=mu0,conf.level=0.95)

        One Sample t-test

data:  y
t = -2.8669, df = 29, p-value = 0.007643
alternative hypothesis: true mean is not equal to 9
95 percent confidence interval:
 4.336738 8.219928
sample estimates:
mean of x
 6.278333

> cat("sigma0 = ", sigma0)       # display value of sigma0
sigma0 =  7

> cat("sample variance = ", s2)          # display sample variance
sample variance =  27.03675

> cat("p-value for testing sigma=sigma0: ", min(2*q,2*(1-q)))
p-value for testing sigma=sigma0:  0.04895834

> cat("95% confidence interval for sigma squared: ",c(s2*df/b,s^2*df/a))
95% confidence interval for sigma squared:   17.14843 48.86035

> cat("95% confidence interval for sigma: ",c(sqrt(s2*df/b),sqrt(s^2*df/a)))
95% confidence interval for sigma:  4.141067 6.990018
```

Problem 2: Run the following R code.

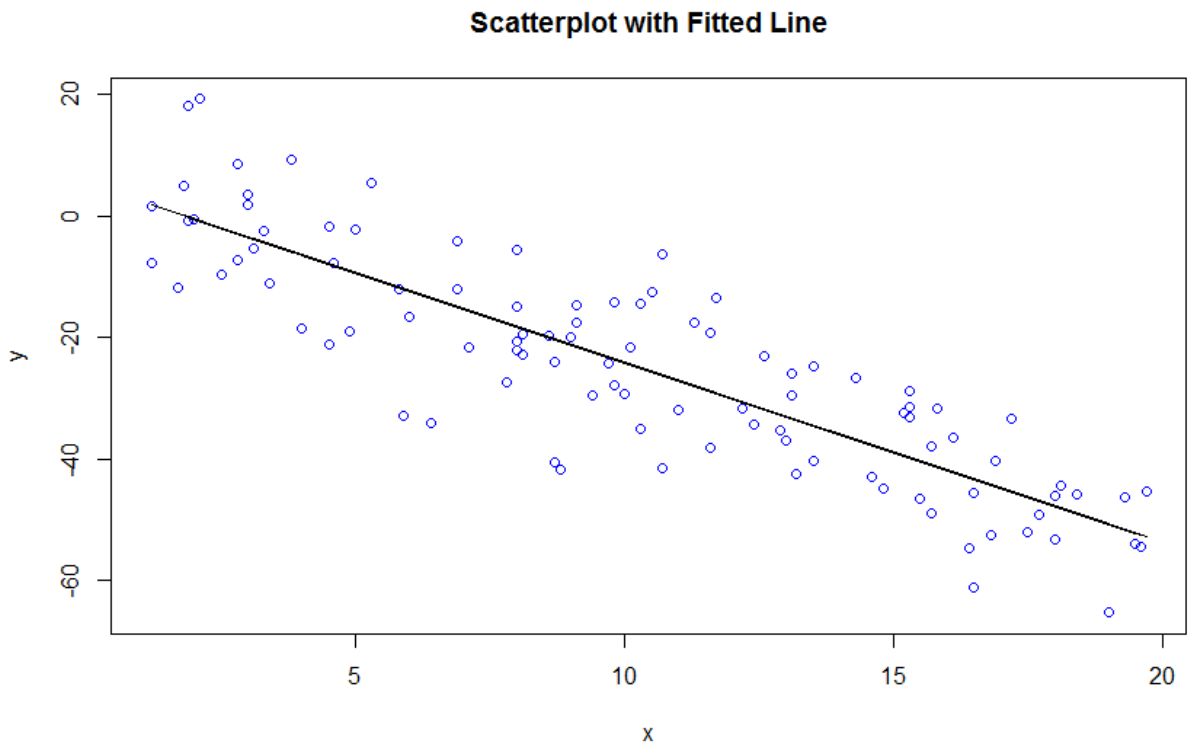
```
##############################################################################################
# Problem 2: Linear regression model
set.seed(id)
x<-round(runif(100,1,20),digits=1)
alpha<-rnorm(1,0,5)
beta<- rnorm(1,0,5)
# display values of alpha and beta
cat("alpha = ", alpha, ", beta = ", beta)
model<-sample(c(1:4),1)
cat("Model = ", model)
# Data are randomly generated depending on the value of the variable model
if (model==1) {
y<-round(alpha+beta*x+rnorm(100,0,10),digits=1)
} else if (model==2) {
y<-round(alpha+beta*x+rnorm(100,0,x),digits=1)
} else if (model==3) {
y<-round(alpha+beta*((x-10)/5)^2+rnorm(100,0,3),digits=1)
} else if (model==4)  {
y<-round(alpha+beta*x+3*rt(100,2),digits=1) }
# display sample correlation
cat("sample correlation = ", cor(x,y))
RegModel<-lm(y~x)
# estimates and p-value for test of no relationship
summary(RegModel)
n<-length(x)                    # n=sample size
r<- RegModel$residuals     # get residuals
se<-sqrt(sum(r^2)/(n-2))   # estimate of sigma
cat("estimate of sigma",se)
betahat<-RegModel$coefficients[2] # estimate of slope
alphahat<-RegModel$coefficients[1] # estimate of intercept
xbar=mean(x)
Sxx<-sum(x^2)-sum(x)^2/n # value of Sxx
# 95% confidence interval for slope beta
a95<-qt(0.975,n-2) # value from t table for 95% confidence interval
pm<-a95*se/sqrt(Sxx)
cat("95% confidence interval for slope:  ", c(betahat-pm,betahat+pm))
#
# 90% confidence interval for mean response at x=5
muhat5<-alphahat+betahat*5     # fitted value for x=5
a90<-qt(0.95,n-2) # value from t table for 90% confidence interval
```

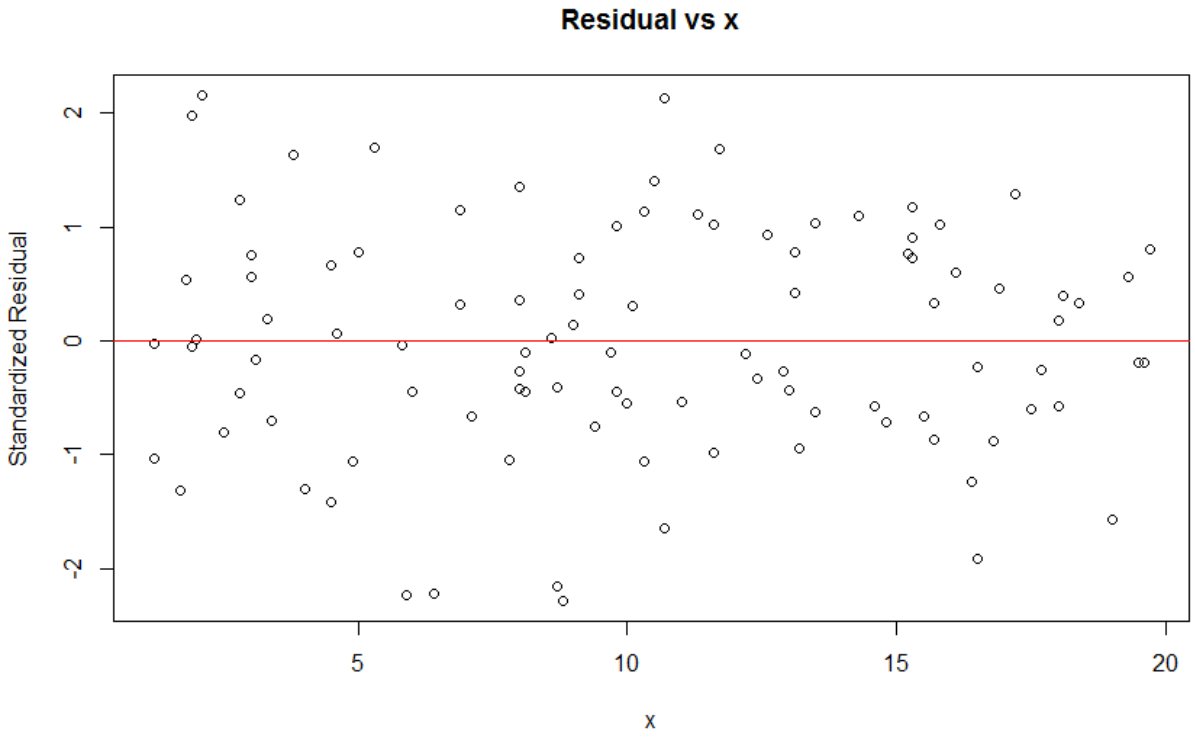
```
pm<-a90*se*sqrt(1/n+(5-xbar)^2/Sxx)
cat("90% confidence interval for mean response at x=5: ",c(muhat5-pm,muhat5+pm))
# 99% prediction interval for response at x=2
muhat2<-alphahat+betahat*2     # predicted value for x=2
a99<-qt(0.995,n-2) # value from t table for 99% prediction interval
pm<-a99*se*sqrt(1+1/n+(2-xbar)^2/Sxx)
cat("99% prediction interval for response at x=2: ",c(muhat2-pm,muhat2+pm))
 # Scatterplot of data with fitted line
muhat<-RegModel$fitted.values
# muhat is the vector of fitted responses
plot(x,y,col="blue")
title(main="Scatterplot with Fitted Line")
points(x,muhat,type="l")
# Residual plots for checking fit of the model
rstar <- r/se   # the standardized residuals
plot(x,rstar,xlab="x",ylab="Standardized Residual")
abline(0,0,col="red",lwd=1.5)
title(main="Residual vs x")
plot(muhat,rstar,xlab="Muhat",ylab="Standardized Residual")
abline(0,0,col="red",lwd=1.5)
title(main="Residual vs Muhat")
qqnorm(rstar,main="")
qqline(rstar,col="red",lwd=1.5)  # add line for comparison
title(main="Qqplot of Residuals")
################################################################################################
```

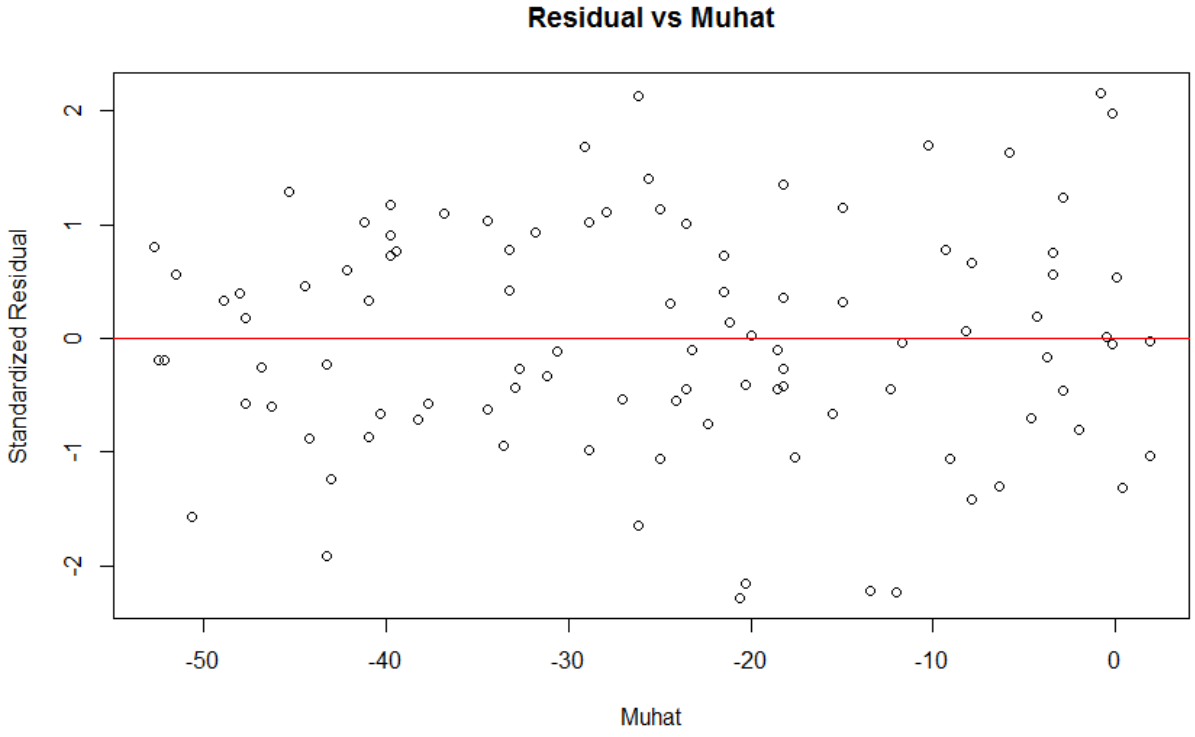
Verify that you obtain the following output:

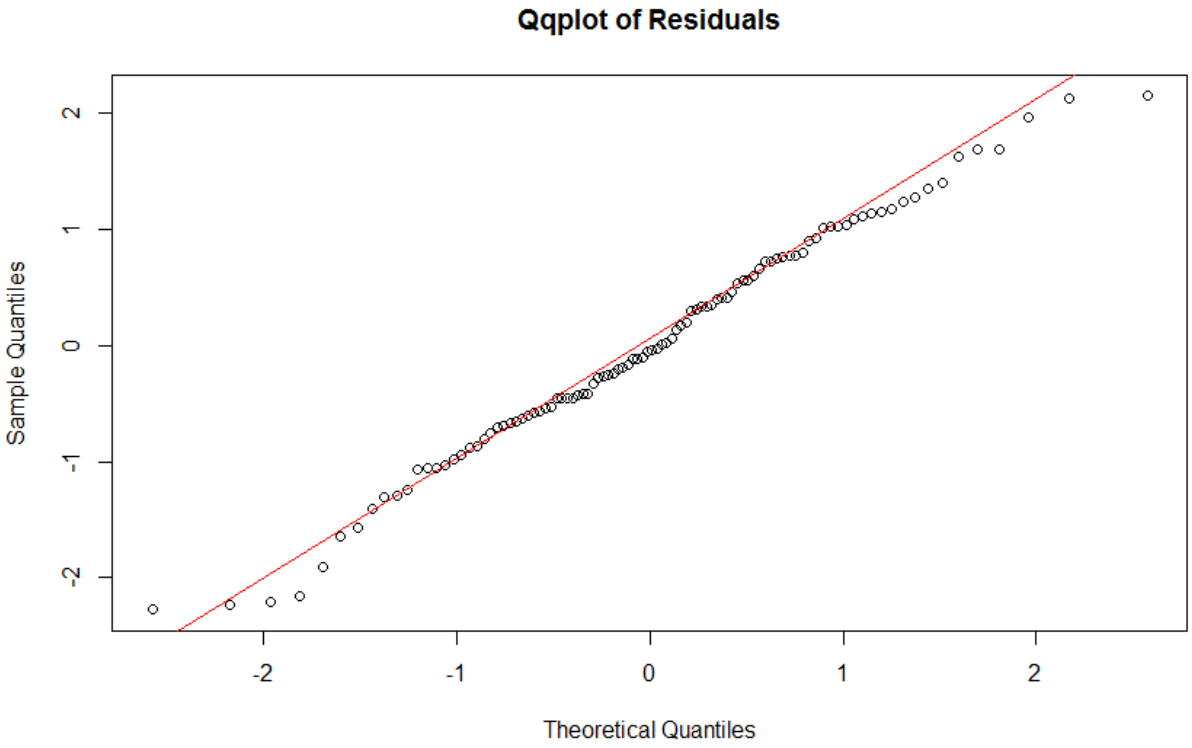
```
> cat("alpha = ", alpha, ", beta = ", beta)
alpha =  2.293382 , beta =  -2.601833

> cat("Model = ", model)
Model =  1

> cat("sample correlation = ", cor(x,y))
sample correlation =  -0.8625526

> RegModel<-lm(y~x)
> # estimates and p-value for test of no relationship
> summary(RegModel)

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-21.2398  -5.9288  -0.4623   7.0616  20.0681
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.435      2.021    2.69   0.0084 **
x             -2.954      0.175  -16.88   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.326 on 98 degrees of freedom
Multiple R-squared: 0.744,     Adjusted R-squared: 0.7414
F-statistic: 284.8 on 1 and 98 DF,  p-value: < 2.2e-16

> cat("estimate of sigma",se)
estimate of sigma 9.326383

> cat("95% confidence interval for slope:  ", c(betahat-a*se/sqrt(Sxx),betaha
t+a*se/sqrt(Sxx)))
95% confidence interval for slope:   -3.301399 -2.606673

> cat("90% confidence interval for mean response at x=5: ",c(muhat5-pm,muhat5
+pm))
90% confidence interval for mean response at x=5:  -11.507 -7.162702

> cat("99% prediction interval for response at x=2: ",c(muhat2-pm,muhat2+pm))
99% prediction interval for response at x=2:  -25.38452 24.43904
```




Residual vs x
Standardized Residual
2
1
0
-1
-2
5
10
15
20
x


Residual vs Muhat
Standardized Residual
2
1
0
-1
-2
-50
-40
-30
-20
-10
0
Muhat

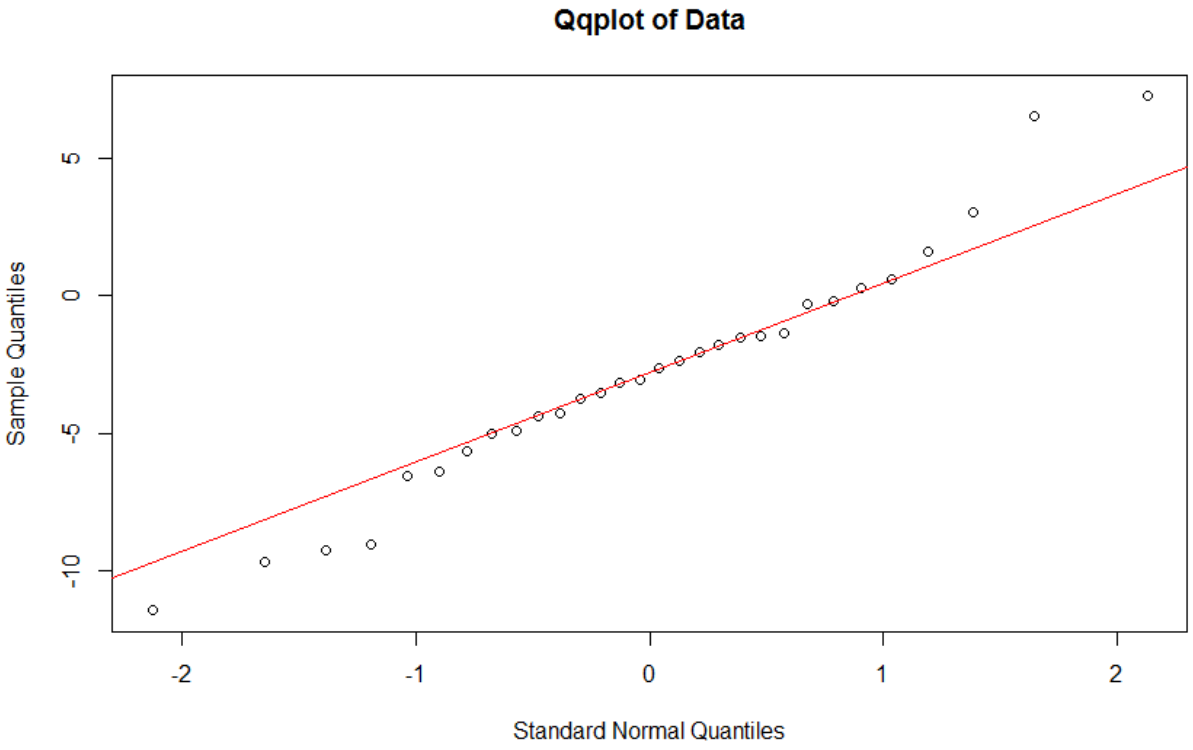

Qqplot of Residuals
Sample Quantiles
Theoretical Quantiles
-2
-1
0
1
2

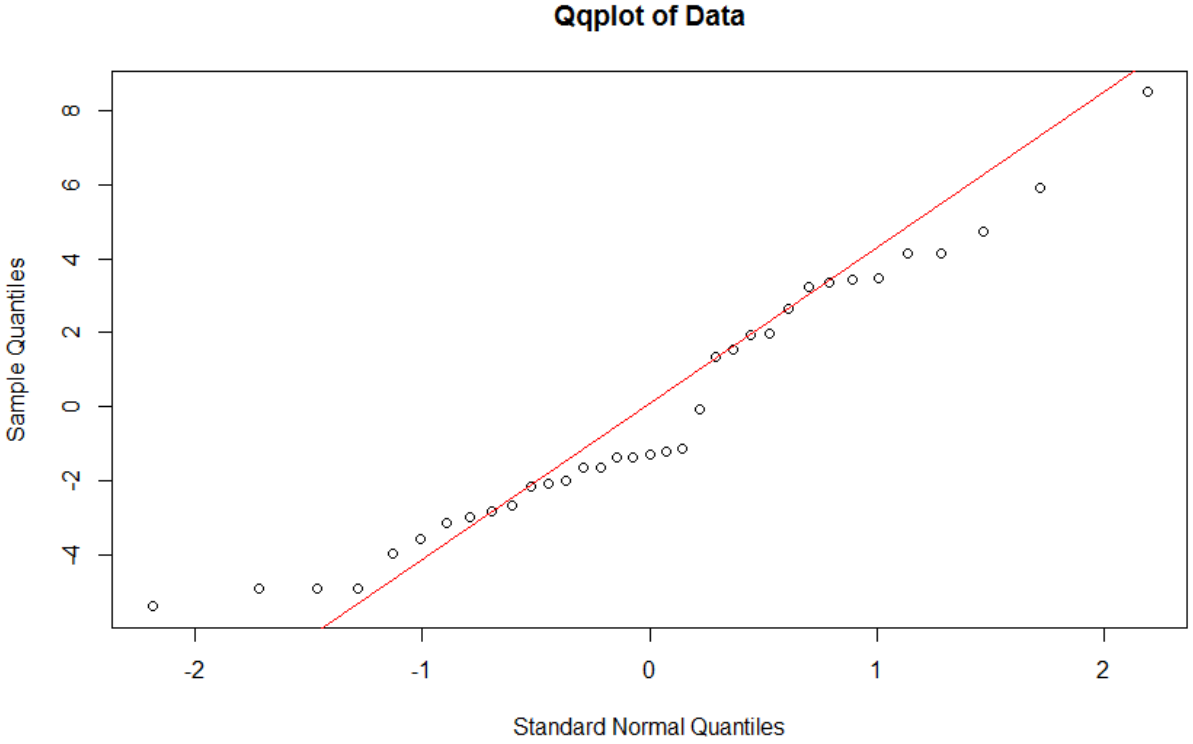
Problem 3: Run the following R code.

```
################################################################################
# Problem 3: Two sample Gaussian model
set.seed(id)
mu1<-rnorm(1,0,2)
mu2<-rnorm(1,0,2)
sigma<-max(1,trunc(id/10)-10*trunc(id/100))   # sig = second last digit of ID unless last digit is zero
# display values of mu1, mu2 and sigma
cat("mu1 = ", mu1, ", mu2 = ", mu2,", sigma = ", sigma)
# Generate data
y1<-sort(round(rnorm(30,mu1,sigma),digits=2))  # 30 observations from G(mu1,sig)
y2<-sort(round(rnorm(35,mu2,sigma),digits=2))  # 35 observations from G(mu2,sig)
# Test hypothesis of no difference in the means
t.test(y1,y2,mu=0,var.equal=TRUE,conf.level=0.95)
s1<-sd(y1)
s2<-sd(y2)
sp<-((29*s1^2+34*s2^2)/63)^0.5
cat("sample sd for sample 1 = ", s1, ", sample sd for sample 2 = ", s2)
cat("pooled estimate of sigma = ", sp)
# Check Gaussian assumption using qqplots
qqnorm(y1,xlab="Standard Normal Quantiles",main="Qqplot of Data")
qqline(y1,col="red",lwd=1.5)  # add line for comparison
qqnorm(y2,xlab="Standard Normal Quantiles",main="Qqplot of Data")
qqline(y2,col="red",lwd=1.5)  # add line for comparison
################################################################################
```

Verify that you obtain the following output:

```
> cat("mu1 = ", mu1, ", mu2 = ", mu2,", sigma = ", sigma)
mu1 =  -3.171255 , mu2 =  -0.371706 , sigma =  5

> t.test(y1,y2,mu=0,var.equal=TRUE,conf.level=0.95)
        Two Sample t-test
data:  y1 and y2
t = -2.7641, df = 63, p-value = 0.007477
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.6087821 -0.7410274
sample estimates:
 mean of x  mean of y
-2.8103333 -0.1354286

> cat("sample sd for sample 1 = ", s1, ", sample sd for sample 2 = ", s2)
sample sd for sample 1 =  4.298046 , sample sd for sample 2 =  3.503656
> cat("pooled estimate of sigma = ", sp)
pooled estimate of sigma =  3.889532
```


Qqplot of Data
Sample Quantiles
5
0
-5
-10
-2
-1
0
1
2
Standard Normal Quantiles


Qqplot of Data
Sample Quantiles
8
6
4
2
0
-2
-4
-2
-1
0
1
2
Standard Normal Quantiles

Run the R code for the 3 problems above again except modify the line

"id<-20456458"

in Problem 1 by replacing the number 20456458 with your UWaterloo ID number.

When you run the R code with your ID number you will generate 7 new plots. Export these 7 plots as .png files using RStudio (See Introduction to R and RStudio Section 6).

Download the Assignment 4 Template which is posted as a Word document on Learn. Fill in the required information and plots based on the output for the data generated using your ID number. Your assignment must follow the template exactly. See Assignment 4 Example posted on Learn.

Create a .pdf file for the answer to EACH problem.
Upload your assignment to Crowdmark one problem at a time using the link which was emailed to you.