Effect of increasing sample size

20 marks

In R there are functions that allow calculation of the density (or probability mass) function $f(x)$, the cumulative distribution function $F(x)$, and the quantile function $Q_X(p)$; there are also functions that will generate pseudo-random observations for each distribution. For example for a $N(0,1)$ distribution, the functions are `dnorm(...)`, `pnorm(...)`, `qnorm(...)`, and `rnorm(...)` respectively. To see all of the distributions for which these functions are built-in see `help("distributions")`.

In this question, you will be generating pseudo-random numbers from three different distributions, and four different sample sizes n:

- Gaussian or $N(0,1)$, Student (3) or t_3, and the χ^2_3 distribution.
- $n \in \{50, 100, 1000, 10000\}$

The goal is to compare different visualizations across distributions and to assess the effect of increasing sample size.

Note: So that we will all be looking at the same pictures, we will set a "seed" for the pseudo-random number generation. Be sure to set the seed as shown in each case below.

a. **(3 marks)** Complete (and hand in) the following code to generate the data that we will be considering

```r
set.seed(314159)
# The normal data
z50 <- rnorm(50)
z100 <- rnorm(100)
z1000 <- rnorm(1000)
z10000 <- rnorm(10000)
zlims <- extendrange(c(z50, z100, z1000, z10000))

# The student t (3) data
t50 <- rt(50,3)
t100 <- rt(100,3)
t1000 <- rt(1000,3)
t10000 <- rt(10000,3)
tlims <- extendrange(c(t50, t100, t1000, t10000))

# The Chi-squared (3) data
c50 <- rchisq(50, 3)
c100 <- rchisq(100,3)
c1000 <- rchisq(1000,3)
c10000 <- rchisq(10000,3)
clims <- extendrange(c(c50, c100, c1000, c10000))
```

You will be using these data to answer the remaining parts of this question.

b. For each of the following arrange the corresponding visualizations of the underlying densities in a 2×2 array (e.g. via `savePar <- par(mfrow=c(2,2))`. Each plot in any given array should share the same data limits, the same underlying distribution, and be labelled according to the distribution that generated the sample, and the size of that sample. For each display type (i.e. quantile plot, boxplot, etc.) there should be three arrays (one for each generating distribution) where only the sample size n varies within array.

Fill all regions with “grey50”.

For each array, comment on how the quality of the display changes as n increases.

i. **(4 marks)** *quantile plots*. Produce the three arrays of changing n, one for each distribution ($N(0,1)$, t_3, and χ^2_3). Submit each arrangement of the four displayed plots and comment on how the quality of the display changes as n increases.

```r
library(ggplot2)
```

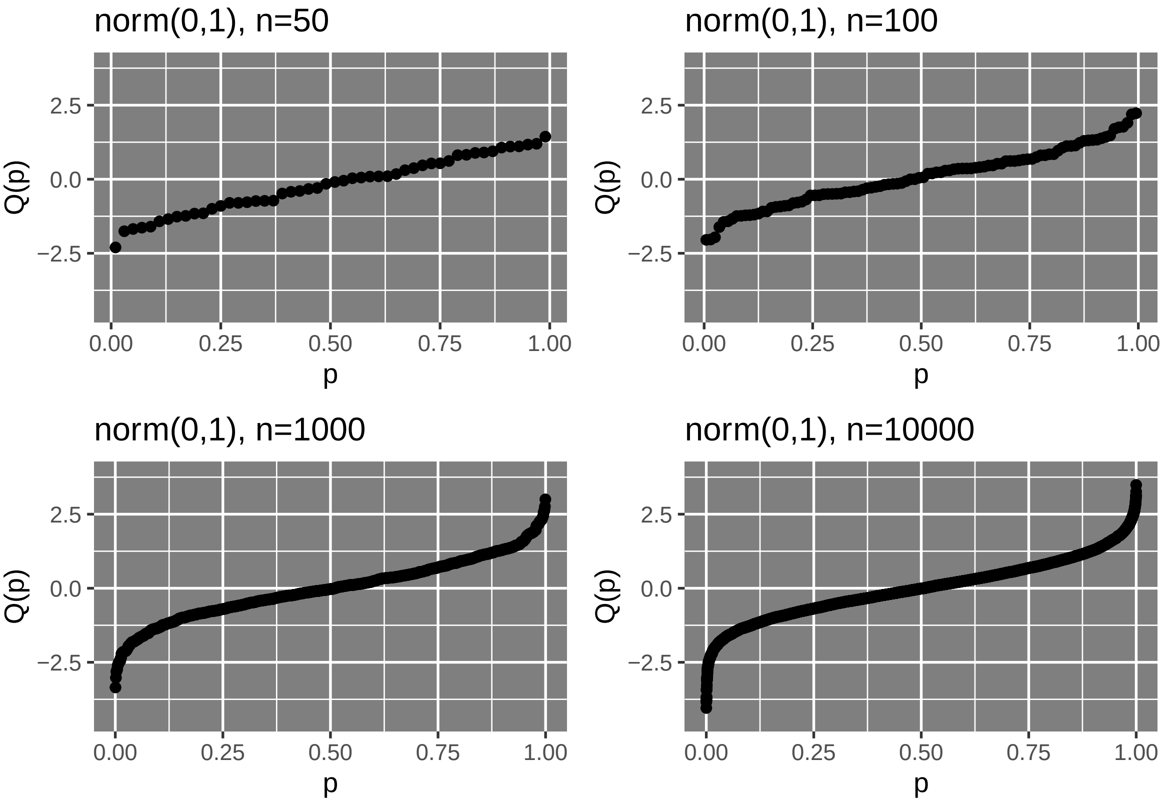
```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```r
library(egg)
```

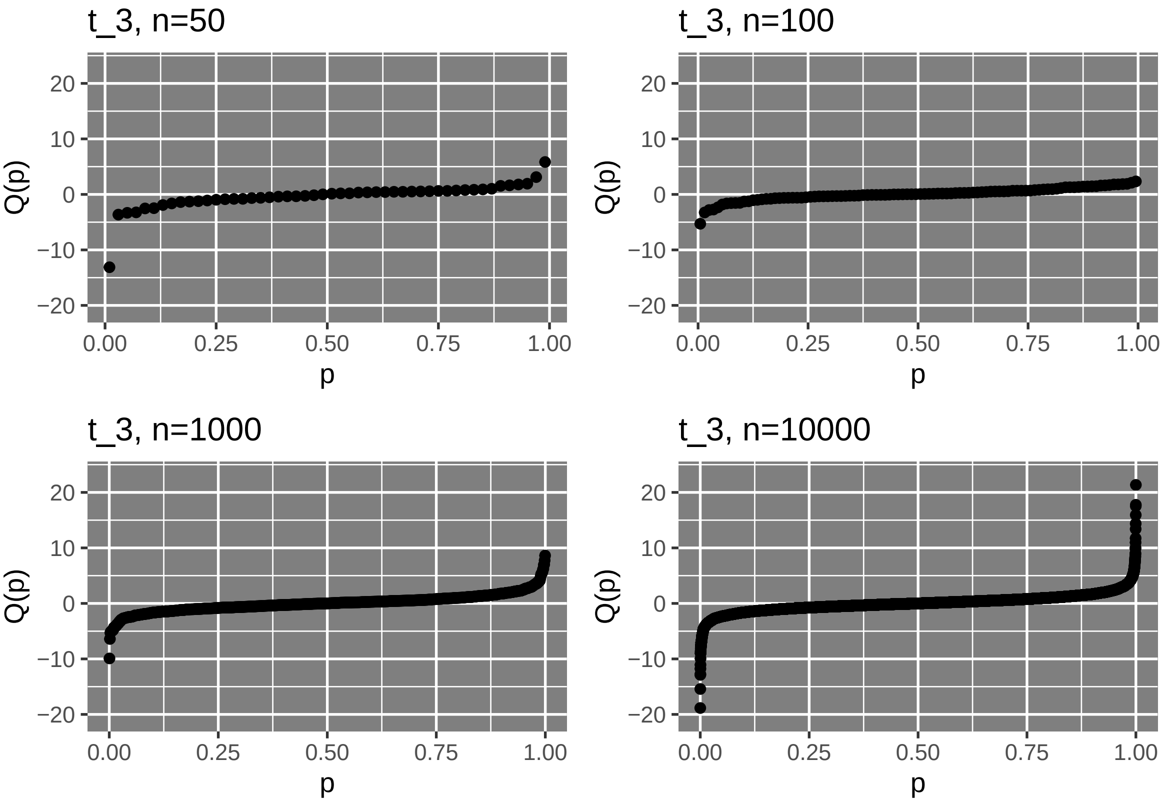
```
## Warning: package 'egg' was built under R version 4.0.3
```

```
## Loading required package: gridExtra
```

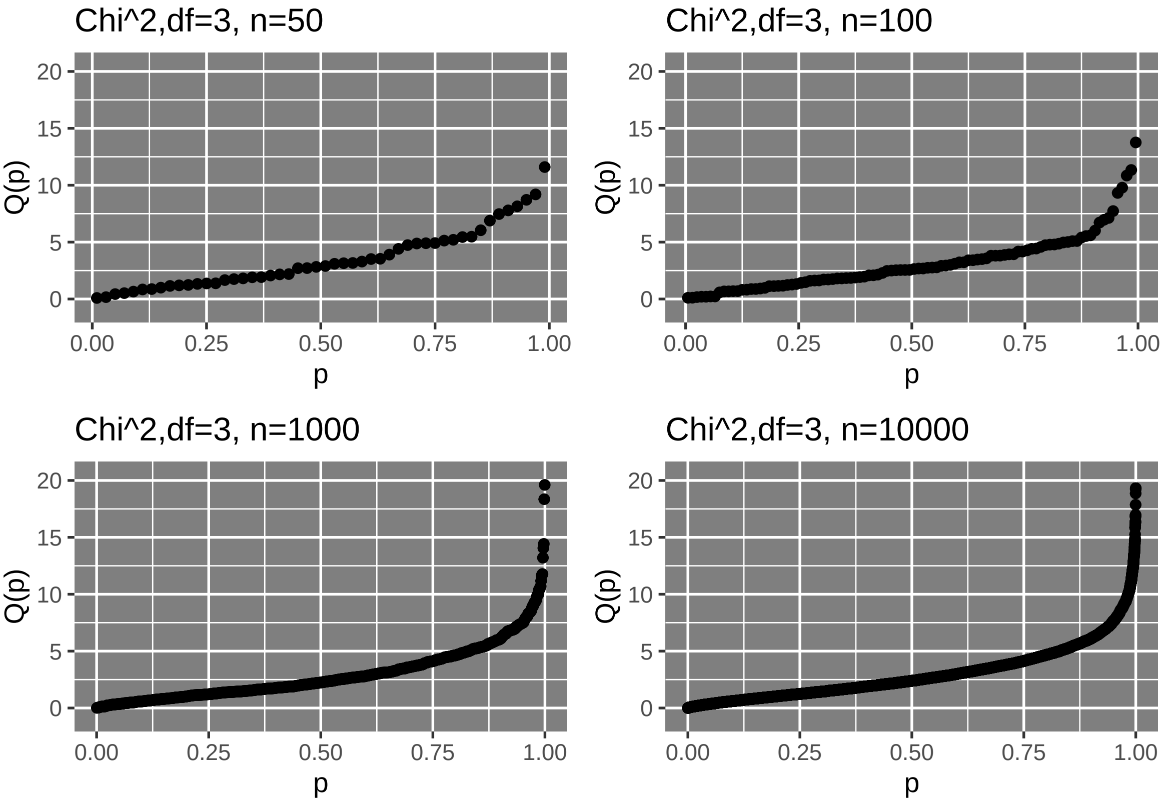
```r
savePar <- par(mfrow=c(2,2))
par(savePar)
plotquantile=function(x, lims){
  ggplot(data=data.frame(x=x), mapping = aes(x=ppoints(length(x)), y=sort(x))) +
    geom_point()+ylim(lims)+
    labs(x="p",y="Q(p)")+
    theme(panel.background = element_rect(fill = "grey50"))
}

p1=plotquantile(z50, zlims)+labs(title="norm(0,1), n=50")
p2=plotquantile(z100, zlims)+labs(title="norm(0,1), n=100")
p3=plotquantile(z1000, zlims)+labs(title="norm(0,1), n=1000")
p4=plotquantile(z10000, zlims)+labs(title="norm(0,1), n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```



```
p1=plotquantile(t50, tlims)+labs(title="t_3, n=50")
p2=plotquantile(t100, tlims)+labs(title="t_3, n=100")
p3=plotquantile(t1000, tlims)+labs(title="t_3, n=1000")
p4=plotquantile(t10000, tlims)+labs(title="t_3, n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```



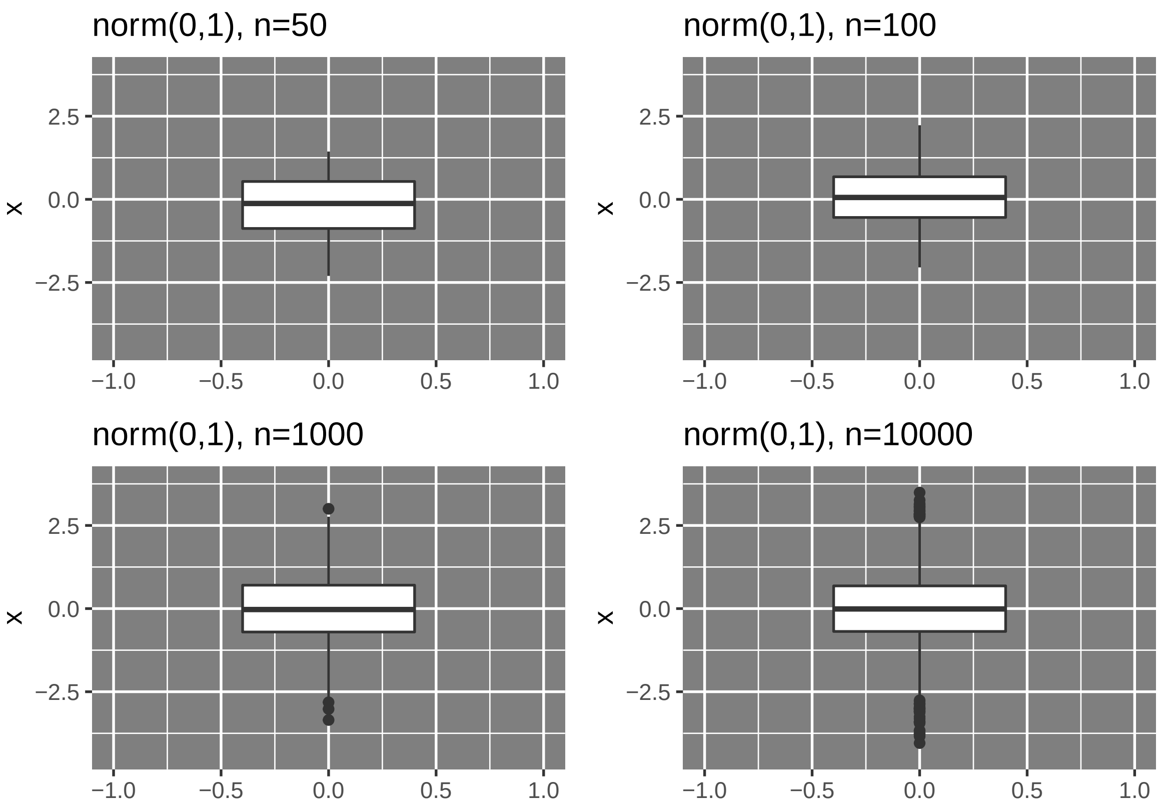
```
p1=plotquantile(c50, clims)+labs(title="Chi^2,df=3, n=50")
p2=plotquantile(c100, clims)+labs(title="Chi^2,df=3, n=100")
p3=plotquantile(c1000, clims)+labs(title="Chi^2,df=3, n=1000")
p4=plotquantile(c10000, clims)+labs(title="Chi^2,df=3, n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```



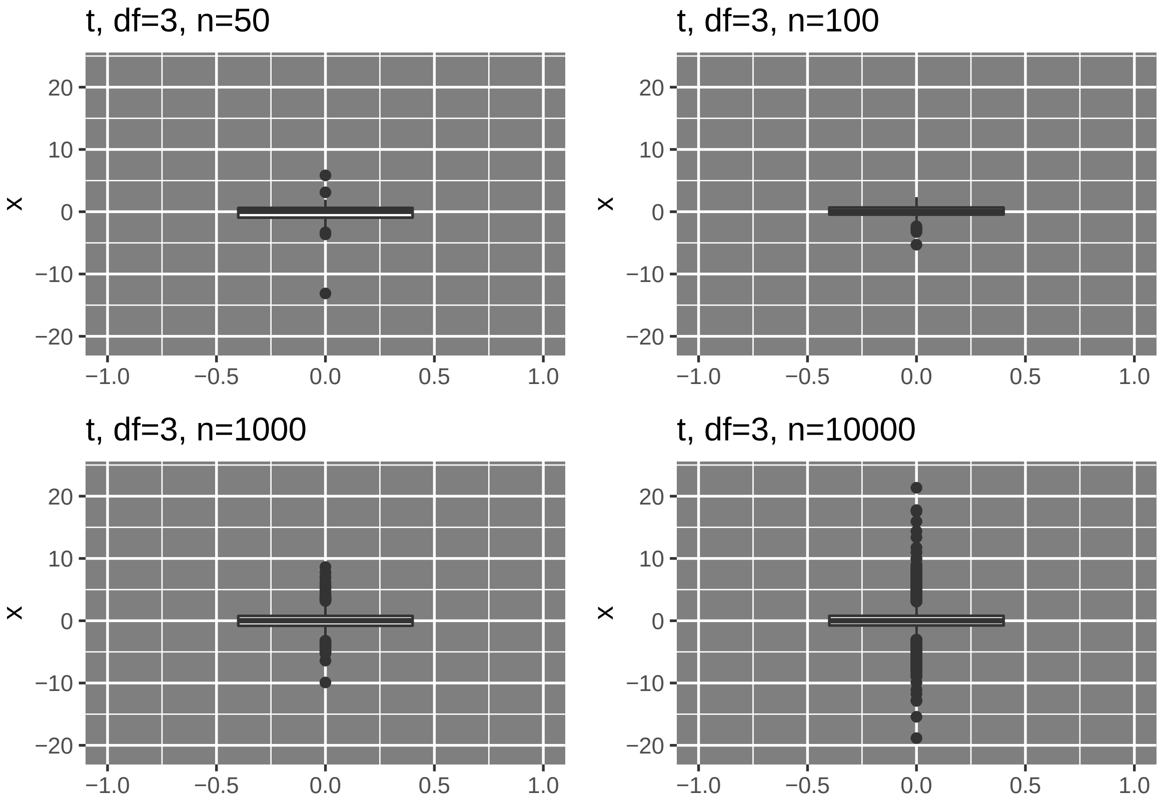
we observe when n is large, we can more easily observe the curve pattern at both ends of the distributions, however when n is small the Quantum plot may look like a straight line, meaning the data does not really represent the distribution well when n is small. So it seems to be better when n is large

ii. **(4 marks)** *boxplots.* Produce the three arrays of changing n, one for each distribution ($N(0,1)$, t_3, and χ^2_3). Submit each arrangement of the four displayed plots and comment on how the quality of the display changes as n increases.

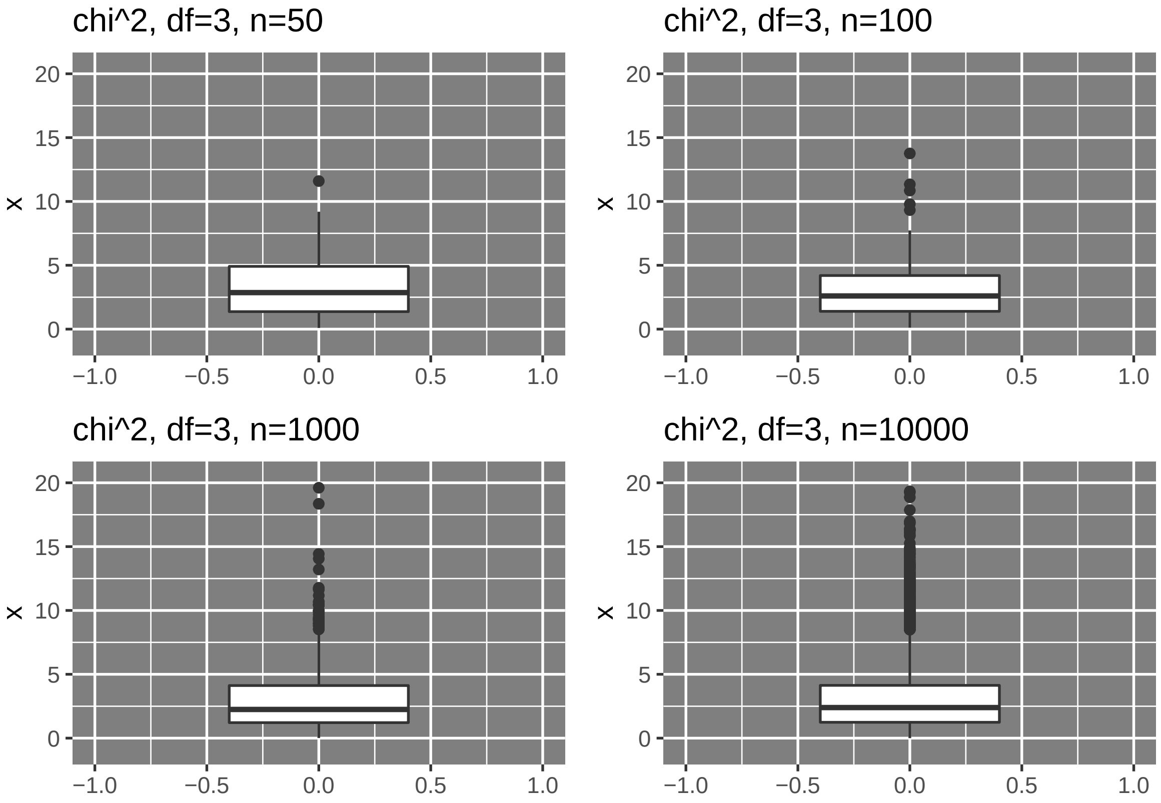
```
  plotbox=function(x, lims){
  ggplot(data=data.frame(x=x), mapping = aes(y=x)) +
    geom_boxplot(width=0.8)+ylim(lims)+xlim(-1,1)+
    theme(panel.background = element_rect(fill = "grey50"))
}

p1=plotbox(z50, zlims)+labs(title="norm(0,1), n=50")
p2=plotbox(z100, zlims)+labs(title="norm(0,1), n=100")
p3=plotbox(z1000, zlims)+labs(title="norm(0,1), n=1000")
p4=plotbox(z10000, zlims)+labs(title="norm(0,1), n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```



```r
p1=plotbox(t50, tlims)+labs(title="t, df=3, n=50")
p2=plotbox(t100, tlims)+labs(title="t, df=3, n=100")
p3=plotbox(t1000, tlims)+labs(title="t, df=3, n=1000")
p4=plotbox(t10000, tlims)+labs(title="t, df=3, n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```



```
p1=plotbox(c50, clims)+labs(title="chi^2, df=3, n=50")
p2=plotbox(c100, clims)+labs(title="chi^2, df=3, n=100")
p3=plotbox(c1000, clims)+labs(title="chi^2, df=3, n=1000")
p4=plotbox(c10000, clims)+labs(title="chi^2, df=3, n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```



we observe when n is large, there are lots of extreme values outside (Q(0.25)-c*IQR,Q(0.75)+c*IQR). This is especially true for t distribution. Moreover the many extreme values makes some boxes really small, increasing difficulty of comparing box position and box height between distributions. However it's easy to compare number of extreme points with boxplot. Though too many extreme points make the graph messy. It may be a better visualization if we draw four graphs of an array in a single boxplot with different x coordinates. Box plot will be better visualization if n is small, so that outliers are countable and won't compress the space of the box.

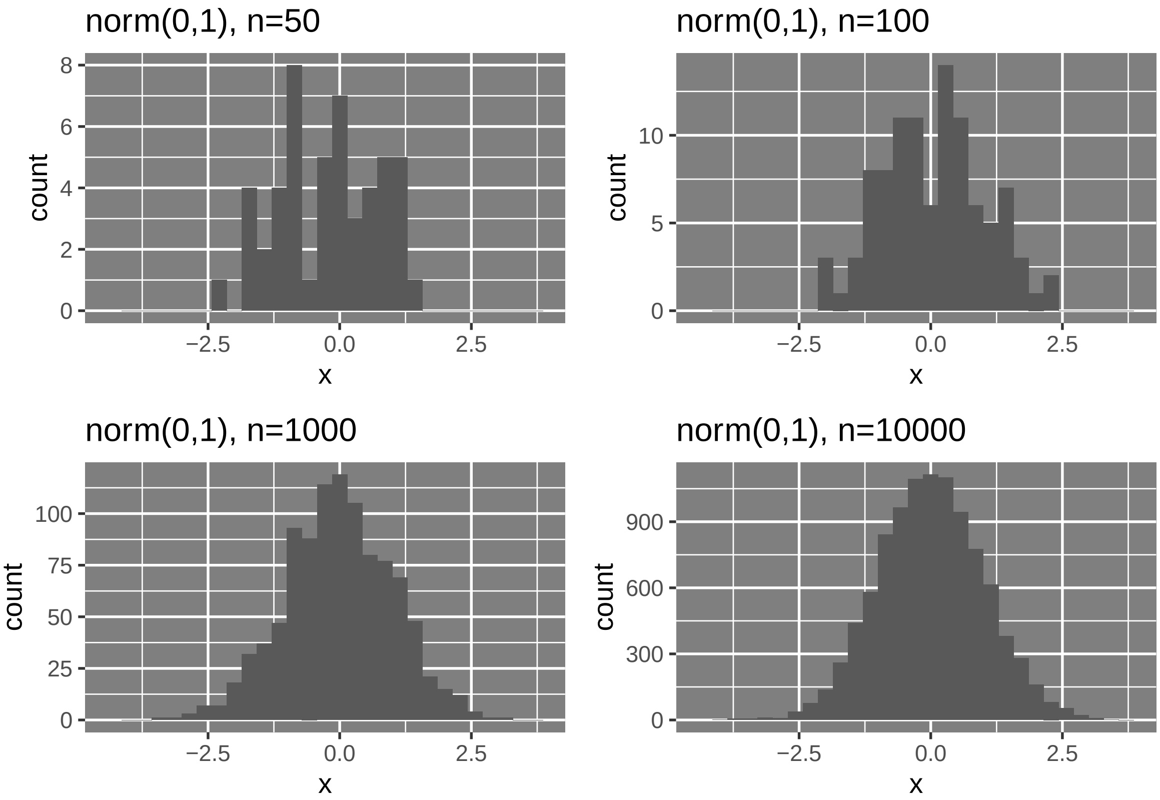
iii. **(4 marks)** *histograms.* Produce the three arrays of changing n, one for each distribution ($N(0,1)$, t_3, and χ^2_3). Submit each arrangement of the four displayed plots and comment on how the quality of the display changes as n increases.

```
plothist=function(x, lims){
  ggplot(data=data.frame(x=x), mapping = aes(x=x)) +
    geom_histogram()+xlim(lims)+
    theme(panel.background = element_rect(fill = "grey50"))
}

p1=plothist(z50, zlims)+labs(title="norm(0,1), n=50")
p2=plothist(z100, zlims)+labs(title="norm(0,1), n=100")
p3=plothist(z1000, zlims)+labs(title="norm(0,1), n=1000")
p4=plothist(z10000, zlims)+labs(title="norm(0,1), n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```

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

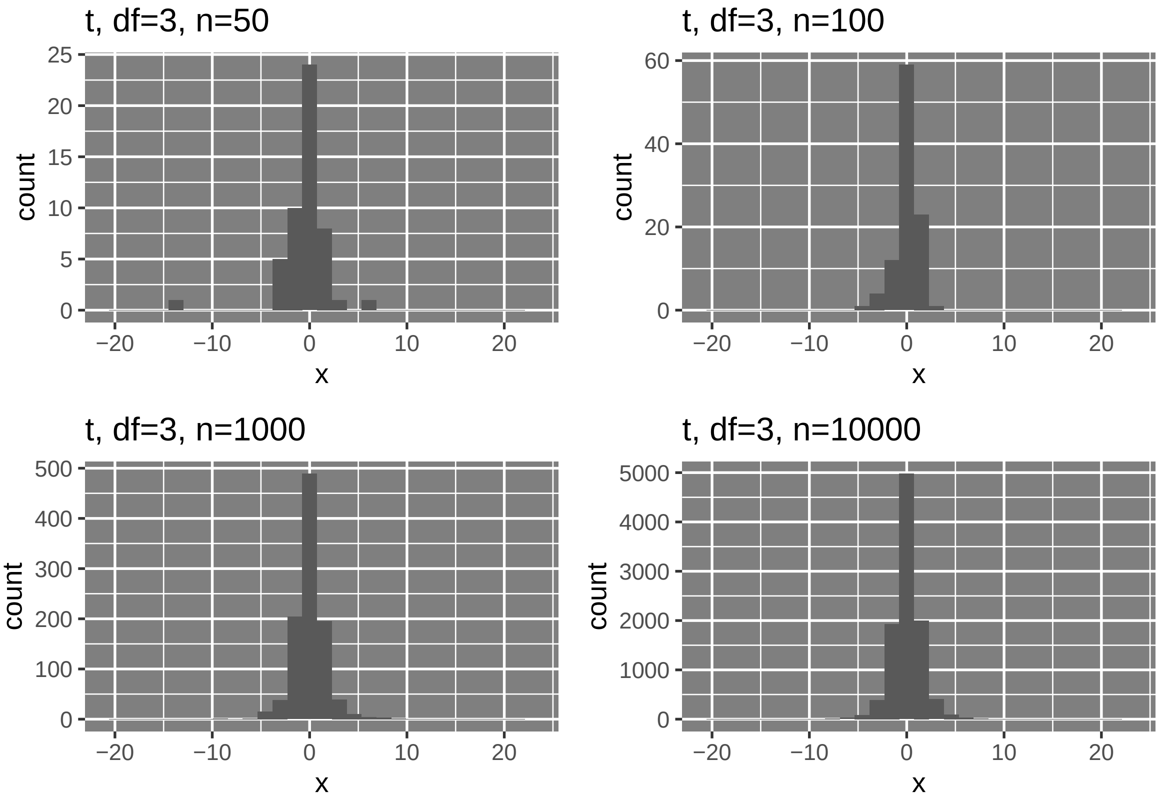
```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 2 rows containing missing values (geom_bar).

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

## Warning: Removed 2 rows containing missing values (geom_bar).

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

## Warning: Removed 2 rows containing missing values (geom_bar).
```



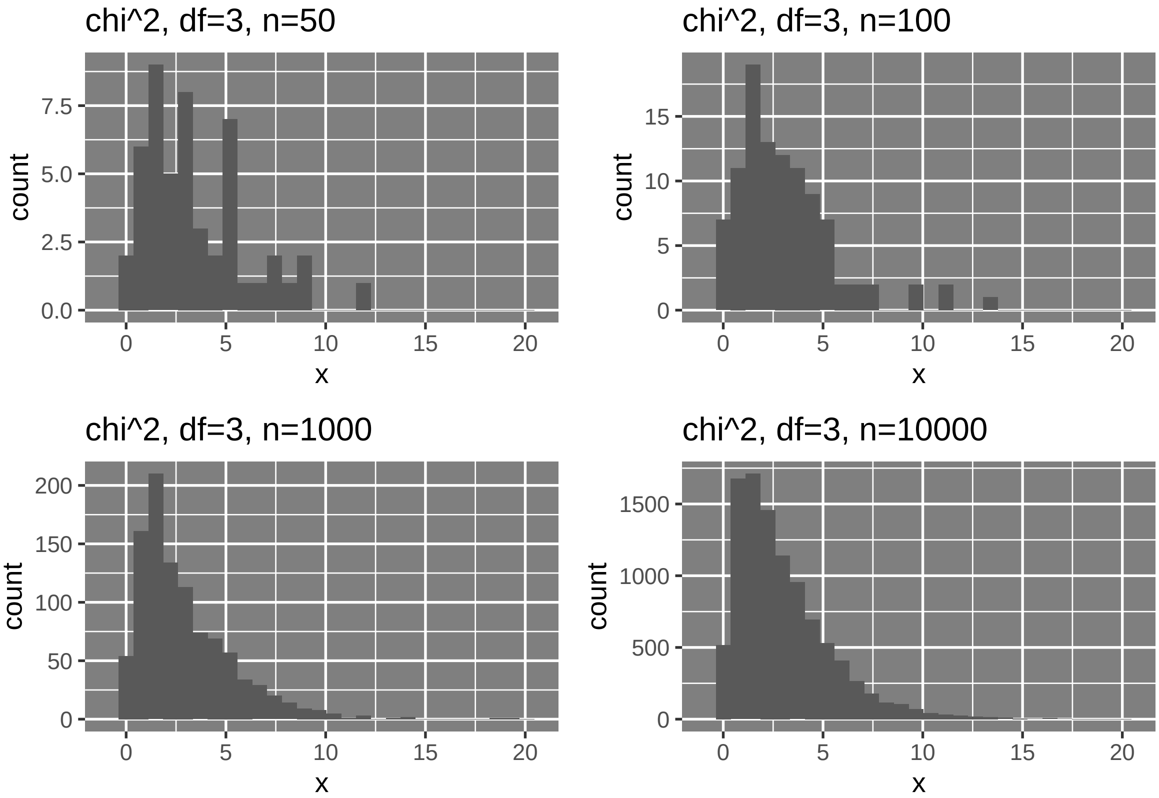
```
p1=plothist(t50, tlims)+labs(title="t, df=3, n=50")
p2=plothist(t100, tlims)+labs(title="t, df=3, n=100")
p3=plothist(t1000, tlims)+labs(title="t, df=3, n=1000")
p4=plothist(t10000, tlims)+labs(title="t, df=3, n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```

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

## Warning: Removed 2 rows containing missing values (geom_bar).

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

## Warning: Removed 2 rows containing missing values (geom_bar).

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

## Warning: Removed 2 rows containing missing values (geom_bar).

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

## Warning: Removed 2 rows containing missing values (geom_bar).
```



```r
p1=plothist(c50, clims)+labs(title="chi^2, df=3, n=50")
p2=plothist(c100, clims)+labs(title="chi^2, df=3, n=100")
p3=plothist(c1000, clims)+labs(title="chi^2, df=3, n=1000")
p4=plothist(c10000, clims)+labs(title="chi^2, df=3, n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```

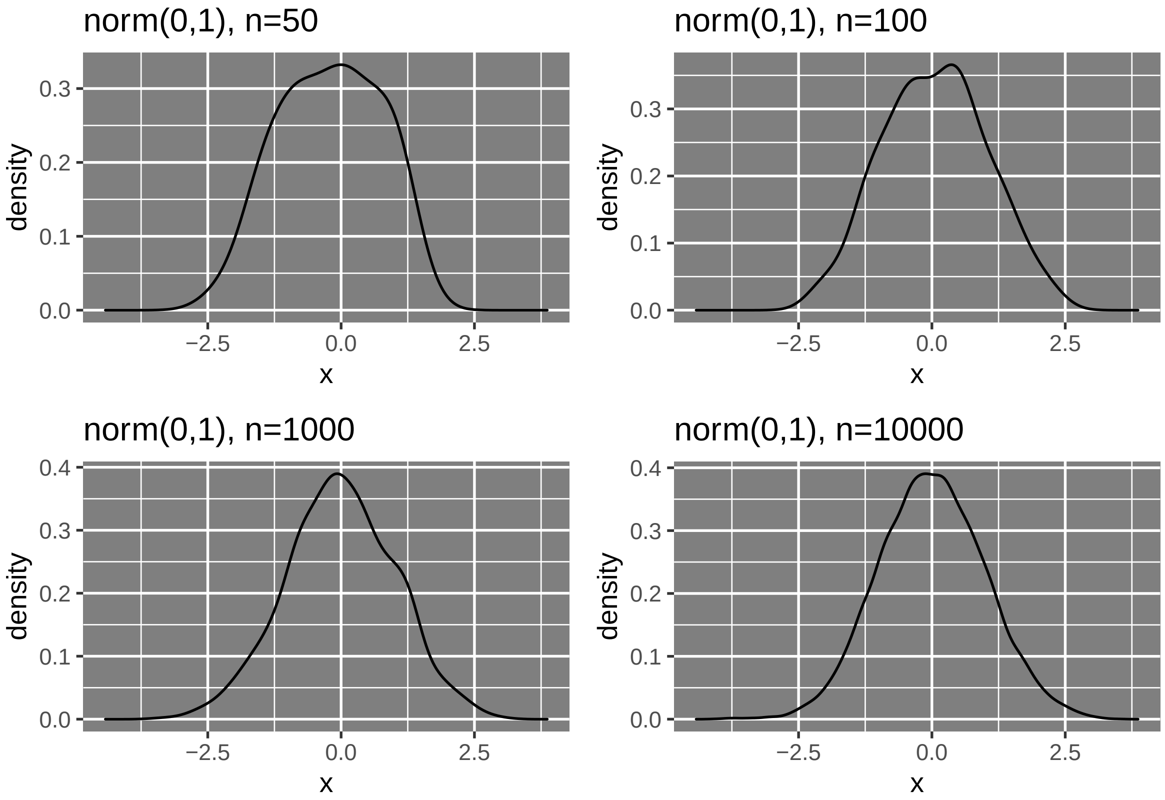
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing missing values (geom_bar).

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

## Warning: Removed 2 rows containing missing values (geom_bar).

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

## Warning: Removed 2 rows containing missing values (geom_bar).

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

## Warning: Removed 2 rows containing missing values (geom_bar).
```



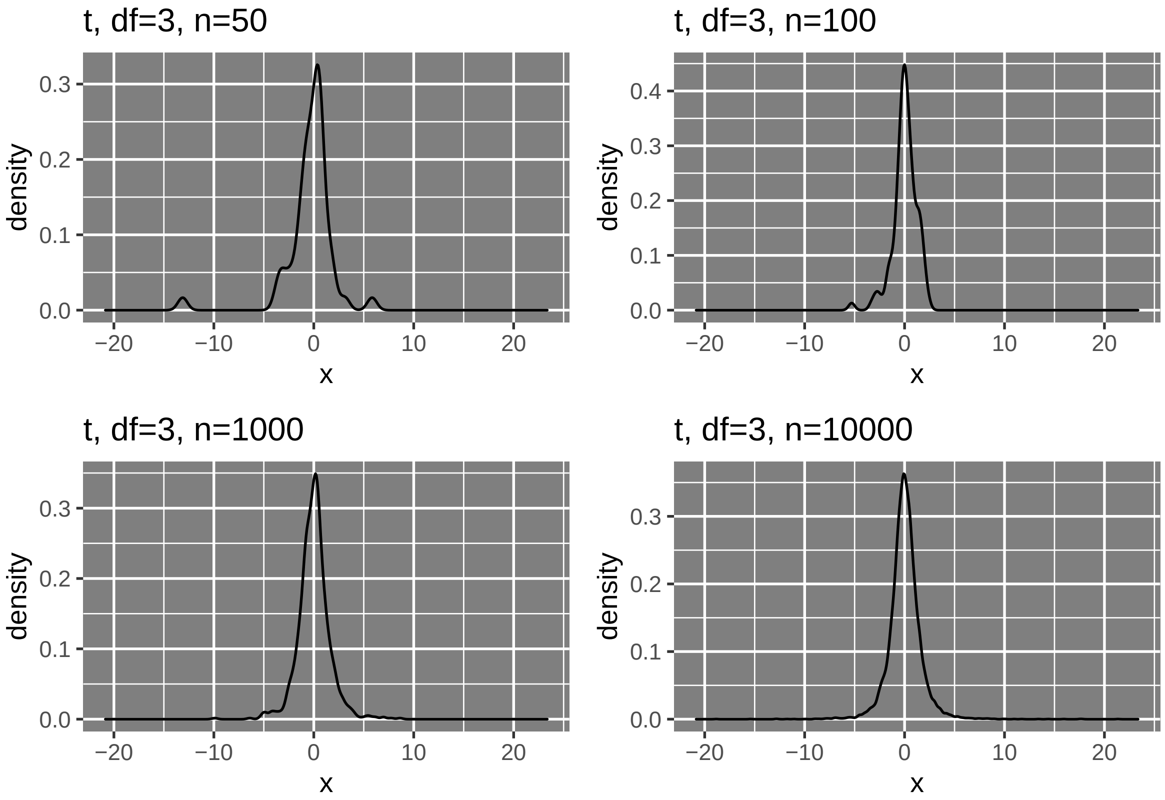
we observe when n is small, we can hardly recognize patterns of these distributions as some choices of bin sizes may give misleading information. As n becomes large it is easier to observe the pattern of distribution. Some extreme values are not reflected in the histograms as bin height as the number of data in that bin divided by n is too small to be reflected as height. So it has better visualization if n is large.

iv. **(5 marks)** *density plots.* Produce the three arrays of changing n, one for each distribution ($N(0,1)$, t_3, and χ^2_3). Submit each arrangement of the four displayed plots and comment on how the quality of the display changes as n increases.

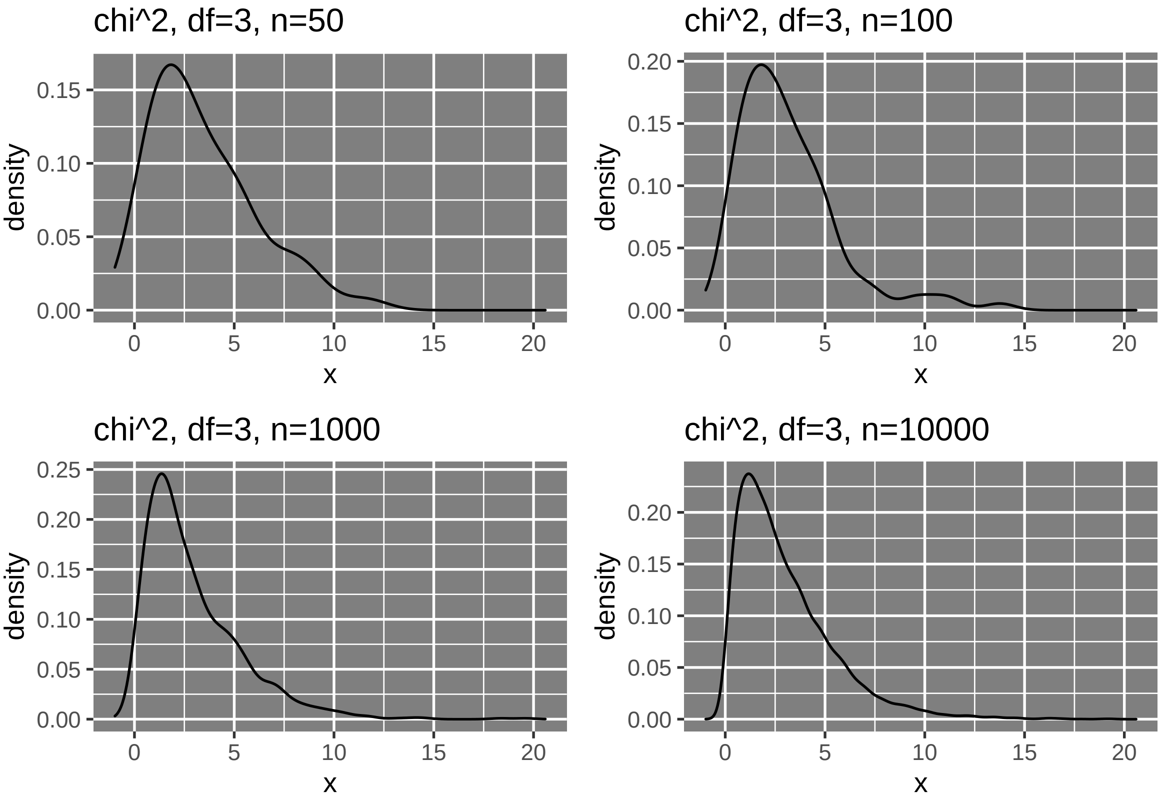
```
plotdensity=function(x, lims){
  ggplot(data=data.frame(x=x), mapping = aes(x=x)) +
    geom_density()+xlim(lims)+
    theme(panel.background = element_rect(fill = "grey50"))
}

p1=plotdensity(z50, zlims)+labs(title="norm(0,1), n=50")
p2=plotdensity(z100, zlims)+labs(title="norm(0,1), n=100")
p3=plotdensity(z1000, zlims)+labs(title="norm(0,1), n=1000")
p4=plotdensity(z10000, zlims)+labs(title="norm(0,1), n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```



```
p1=plotdensity(t50, tlims)+labs(title="t, df=3, n=50")
p2=plotdensity(t100, tlims)+labs(title="t, df=3, n=100")
p3=plotdensity(t1000, tlims)+labs(title="t, df=3, n=1000")
p4=plotdensity(t10000, tlims)+labs(title="t, df=3, n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```



```
p1=plotdensity(c50, clims)+labs(title="chi^2, df=3, n=50")
p2=plotdensity(c100, clims)+labs(title="chi^2, df=3, n=100")
p3=plotdensity(c1000, clims)+labs(title="chi^2, df=3, n=1000")
p4=plotdensity(c10000, clims)+labs(title="chi^2, df=3, n=10000")
ggarrange(p1,p2,p3,p4,nrow = 2, ncol = 2)
```



we observe density plot capture most attribute of distribution even when n is not very large. This is because density plot use kernel method to make the estimate of density robust. However facts remain that as n grows larger, the curves become smoother. It present better visualization when n is large.