✪ Psychological Statistics ✪

Week 03: *Assumptions & GGPlot*

- Edited by Prof. **Changwei Wu**
- Graduate Institute of Mind, Brain and Consciousness (***GIMBC***), Taipei Medical University

In [ ]:

```
### [ Setup the working directory ]

setwd("/Users/wesley/[Course]/Python/R_Script")
# → Please edit the directory name in your computer.
getwd()
```

In [43]:

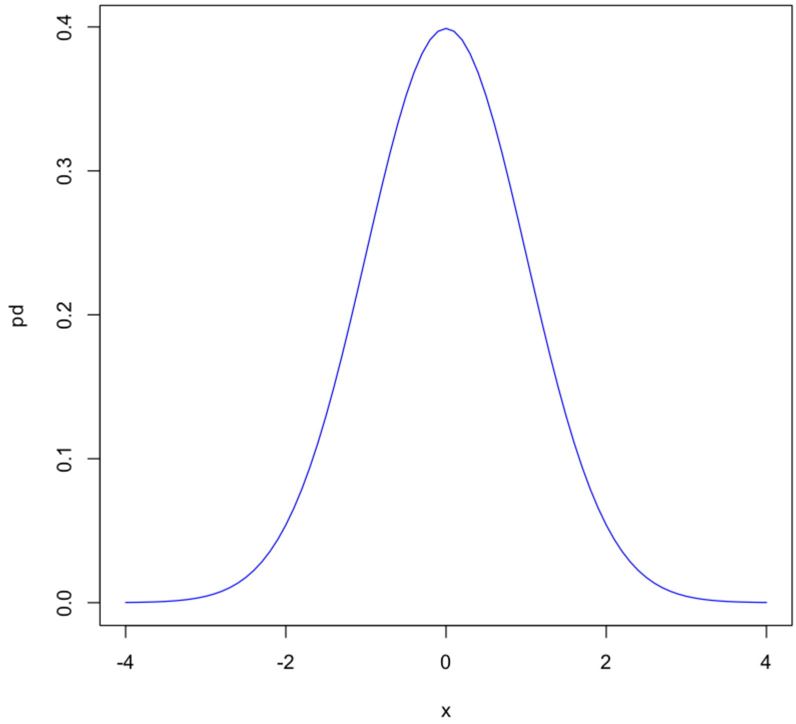
```
### [ Loading the required libraries ]

library("ggplot2")
library("ggpubr")
library("car")
library("agricolae")
```

---

(1) Variance & Distribution

In [9]:

```
### 1-1.[ Plotting normal distribution ]

x <- seq(-4,4,0.1)
pd <- dnorm(x)
plot(x,pd,type="l",col="blue")
```



In [10]:

```
### 1-2.[ Focusing on the reporting numbers ]
# → What does that mean with the reporting number?

qnorm(c(0.025,0.975))

pnorm(-1.96)

pnorm(1.96)

1-pnorm(1.96)
```

-1.95996398454005 · 1.95996398454005

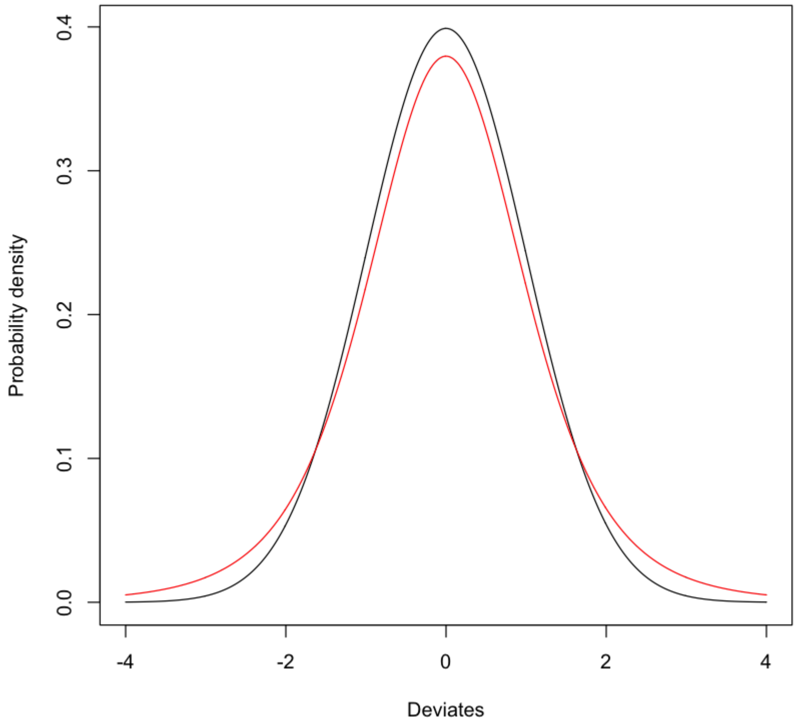
0.0249978951482204

0.97500210485178

0.0249978951482204

In [11]:

```
### 1-3.[ From normal to t distribution ]
# → Function: dt/pt/rt/qt {stats}

x <- seq(-4,4,0.01)
plot(x,dnorm(x),type="l",
 ylab="Probability density",xlab="Deviates")
lines(x,dt(x,df=5),col="red") # degree of freedom = 5
```



In [13]:

```
### 1-4.[ Confidence intervals = (sample t) x (standard error) ]
# → Function: dt/pt/rt/qt {stats}

qt(.025,9)

(upper.95 <- 15 + qt(.975,24) * sqrt(16/25))
(lower.95 <- 15 + qt(.025,24) * sqrt(16/25))

(upper.99 <- 15 + qt(.995,24) * sqrt(16/25))
(lower.99 <- 15 + qt(.005,24) * sqrt(16/25))
```

-2.2621571627982

16.6511188493024

13.3488811506976

17.2375516038196

12.7624483961804

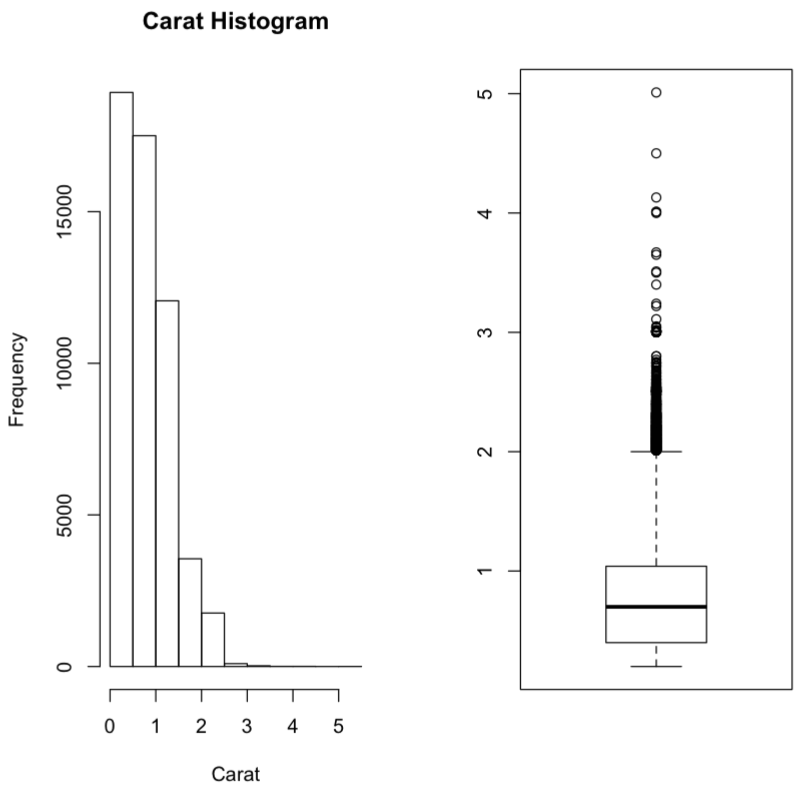
(2) GGPLOT2: the most famous library in R

In [14]:

```
#Load the built-in data in GGPLOT2

data(diamonds)
```

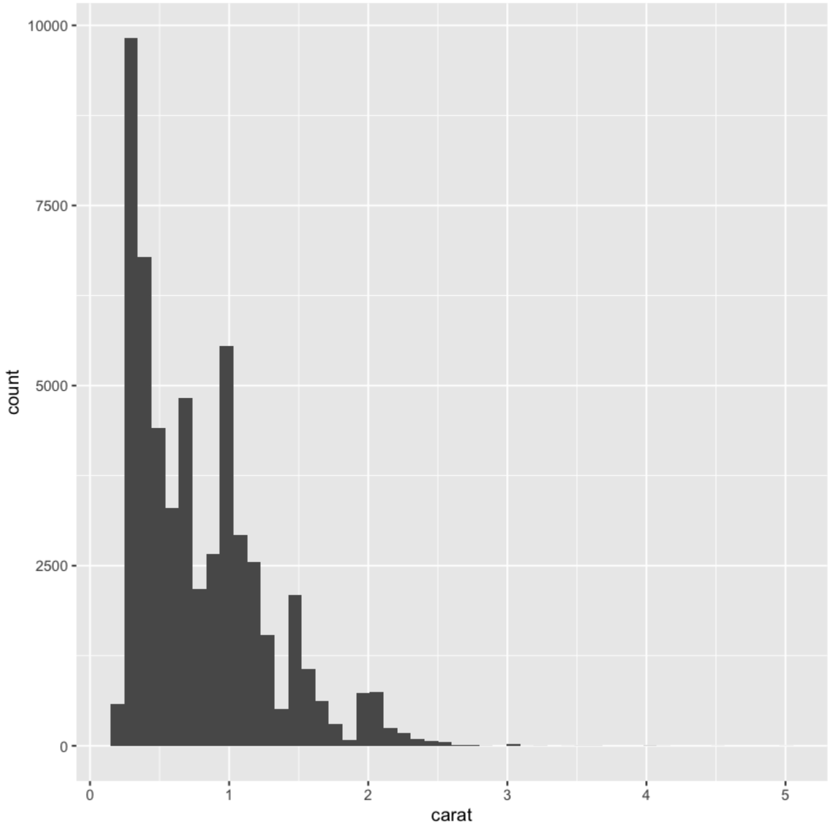
In [15]:

```
### 2-1.[ Histogram & Boxplot ]

par(mfrow=c(1,2))
hist(diamonds$carat, main = "Carat Histogram", xlab = "Carat")
boxplot(diamonds$carat)

#plot(price ~ carat, data = diamonds)
#plot(diamonds$carat, diamonds$price)
```



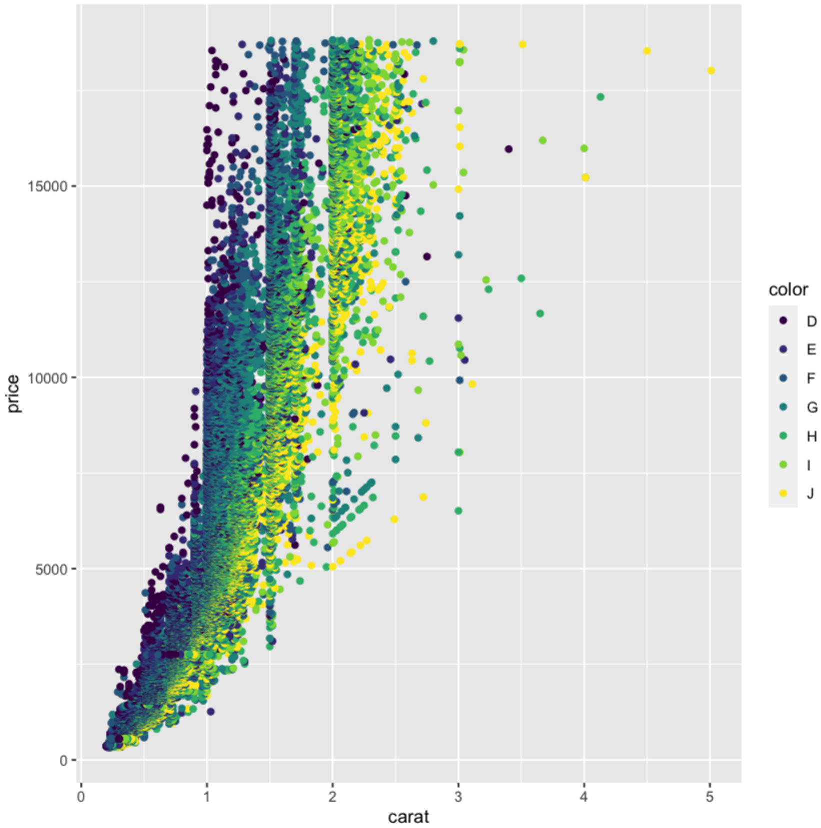
In [16]:

```
### 2-2.[ Use GGPlot2 to plot ]
# → (1) histogram

ggplot(diamonds, aes(x = carat)) + geom_histogram(bins=50)

#ggplot(diamonds, aes(x = carat)) + geom_density(fill = "grey50")
```



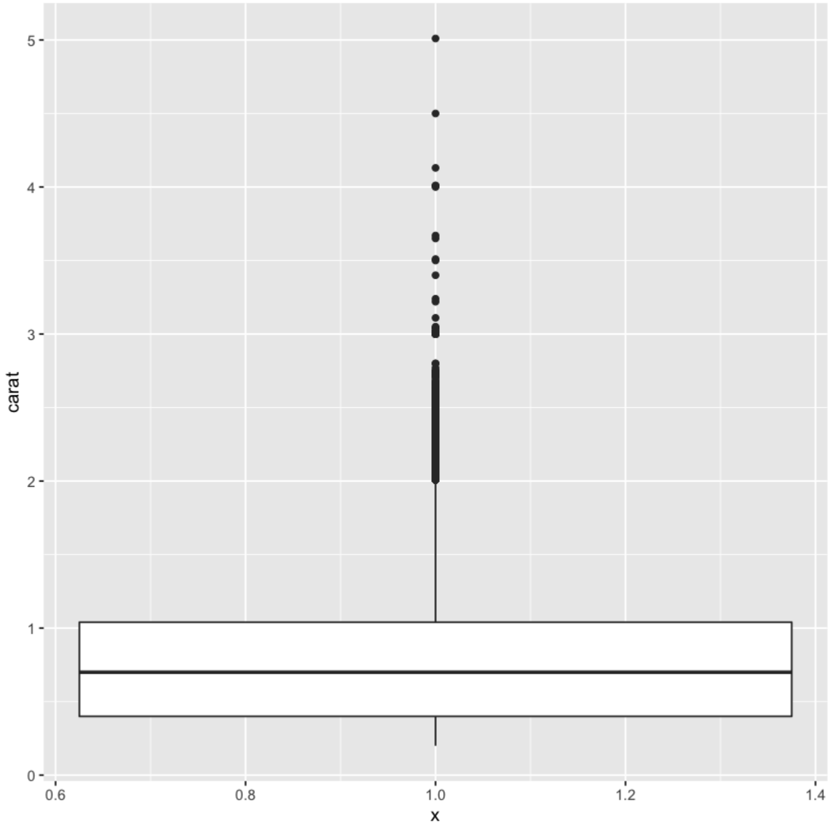
In [17]:

```
# → (2) scatter plot

g <- ggplot(diamonds, aes(x = carat, y = price))
#g + geom_point()
g + geom_point(aes(color = color))
#g + geom_point(aes(color = color)) + facet_wrap(~color, nrow=2)
#g + geom_point(aes(color = color)) + facet_grid(cut ~ clarity)
```



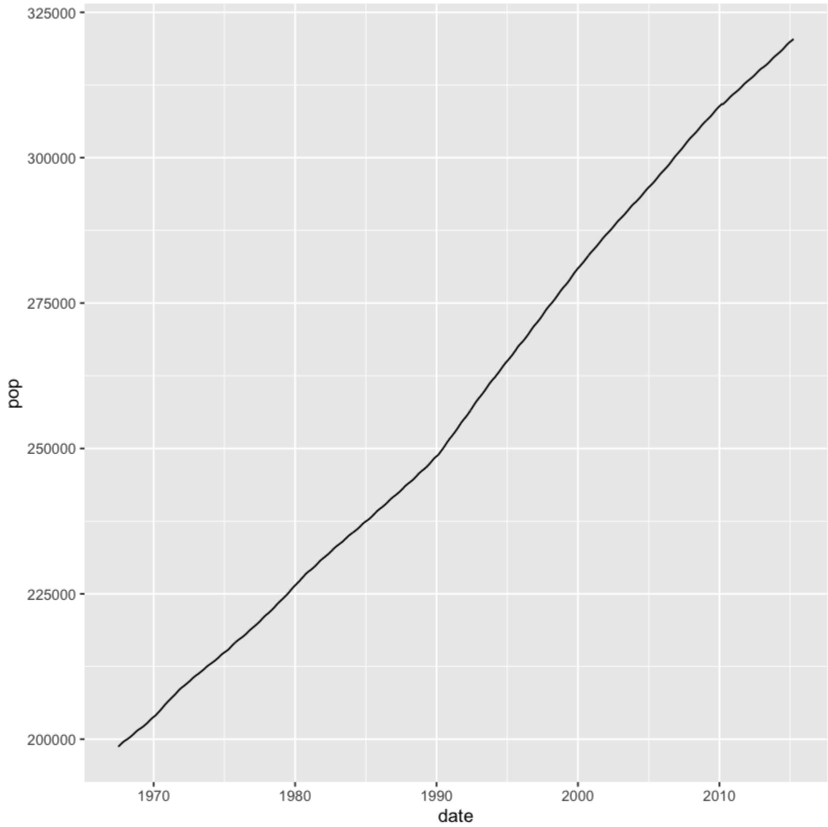
In [18]:

```
# → (3) boxplot (or violins plot)

ggplot(diamonds, aes(y = carat, x = 1)) + geom_boxplot()
#ggplot(diamonds, aes(y = carat, x = cut)) + geom_boxplot()
#ggplot(diamonds, aes(y = carat, x = cut)) + geom_violin()
#ggplot(diamonds, aes(y = carat, x = cut)) + geom_violin() + geom_point()
#ggplot(diamonds, aes(y = carat, x = cut)) + geom_point() + geom_violin()
```



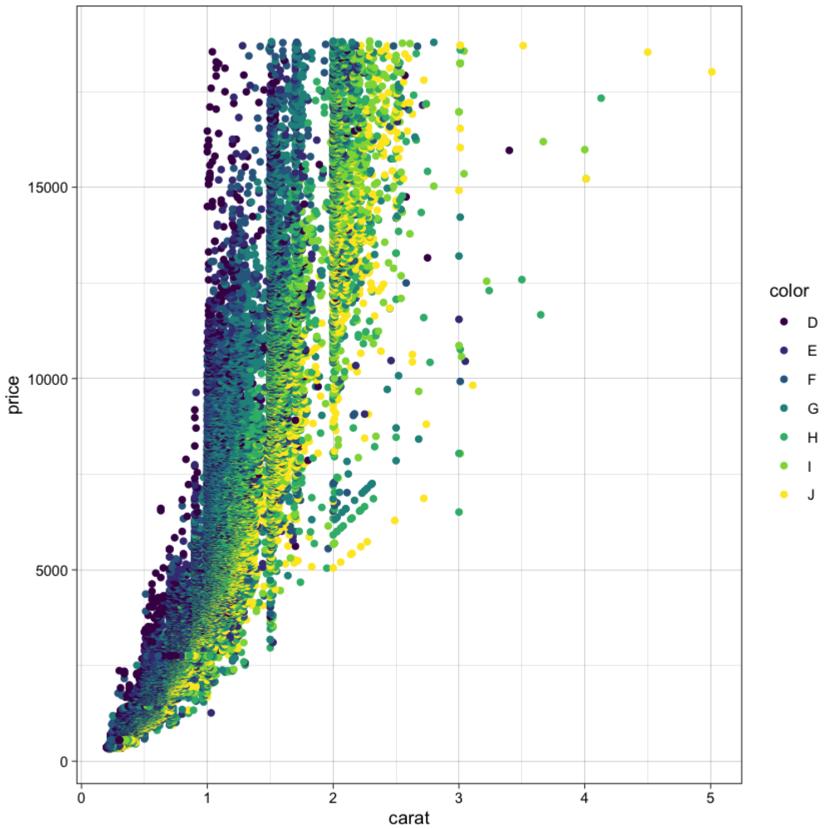
In [19]:

```
# → (4) line plots

ggplot(economics, aes(x = date, y = pop)) +
geom_line()
```



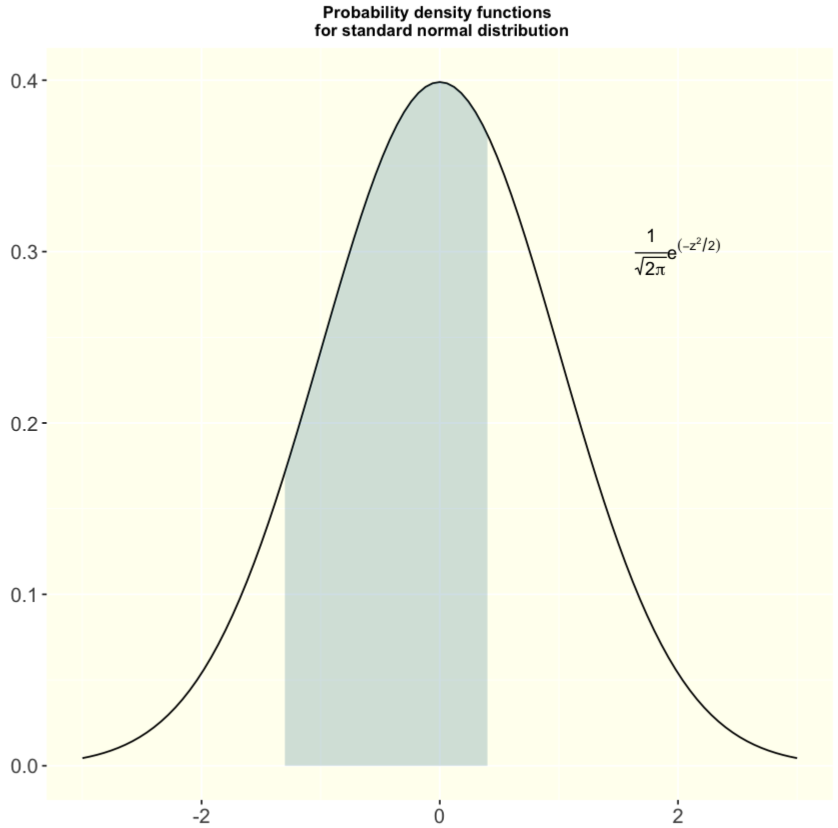
In [22]:

```
# → (5) themes

g2 <- ggplot(diamonds, aes(x=carat, y=price)) + geom_point(aes(color=color))
g2 + theme_linedraw()

#g2 + theme_minimal()
#g2 + theme_dark()

#------- Additional Themes -------#
#library("ggthemes")
#g2 + theme_excel() + scale_colour_excel()
#g2 + theme_economist() + scale_colour_economist()
#g2 + theme_wsj() + scale_colour_wsj()
```



In [23]:

```
### Drawing a pretty standard normal distribution by using ggplot2 ###
p <- ggplot(data.frame(x=c(-3,3)), aes(x=x)) + stat_function(fun = dnorm)
p + annotate("text", x=2, y=0.3, parse=TRUE, label="frac(1, sqrt(2*pi)) * e ^(-z^2/2
    theme(plot.subtitle = element_text(vjust = 1),
    plot.caption = element_text(vjust = 1),
    axis.text.x = element_text(size = 12),
    axis.text.y = element_text(size = 12),
    plot.title = element_text(size = 10, face = "bold", hjust = 0.5),
    panel.background = element_rect(fill = "ivory")) +
    labs(title = "Probability density functions \n for standard normal distribution'
    x = NULL, y = NULL) +
    stat_function(fun = dnorm,
    xlim = c(-1.3,0.4),
    geom = "area",fill="#00688B", alpha= 0.2)
```



(3) Assumption Check: Normality & Homogeneity of Variances

In [28]:

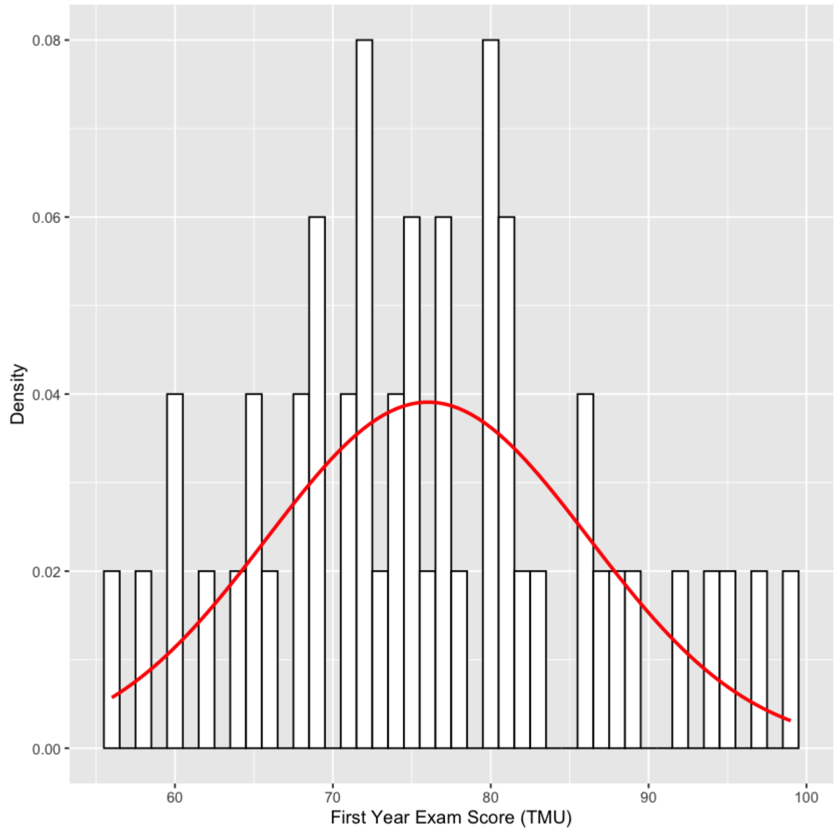
```
### 3-1.[ Load the data with relabeling ]
# → Function: factor {base}

rexam <- read.delim("rexam.dat", header=TRUE)

# → Set the variable uni to be a factor:
rexam$uni<-factor(rexam$uni, levels = c(0:1), labels = c("NTU", "TMU"))

head(rexam,5)
```

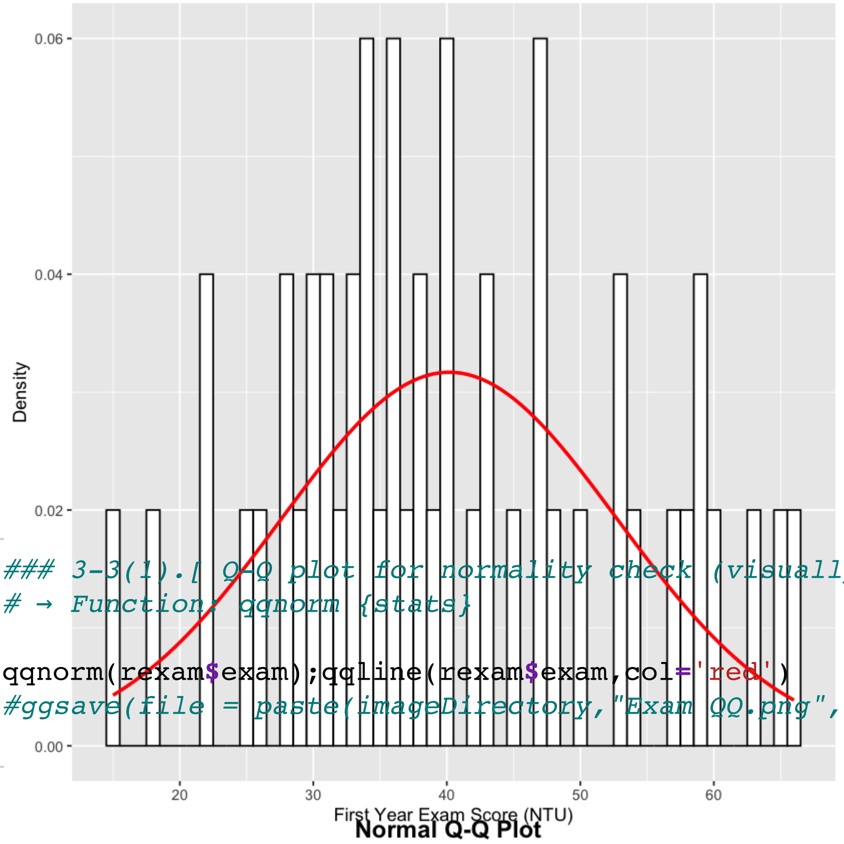
A data.frame: 5 × 5

	exam	computer	lectures	numeracy	uni
	<int>	<int>	<dbl>	<int>	<fct>
1	18	54	75.0	7	NTU
2	30	47	8.5	1	NTU
3	40	58	69.5	6	NTU
4	30	37	67.0	6	NTU
5	40	53	44.5	2	NTU

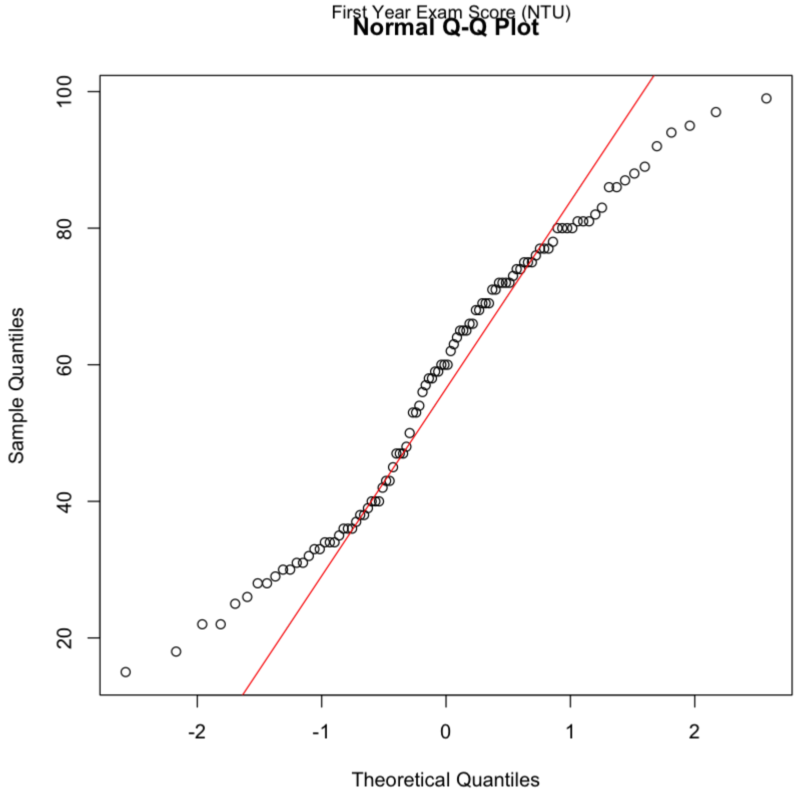
In [30]:

```
### 3-2.[ using subset to plot histograms for different groups ]
# → Function: subset {base}

#using subset to plot histograms for different groups:
TMUdata<-subset(rexam, rexam$uni=="TMU")
NTUdata<-subset(rexam, rexam$uni=="NTU")

hist.exam.TMU <- ggplot(TMUdata, aes(exam)) + theme(legend.position = "none") +
 geom_histogram(aes(y = ..density..), fill = "white", colour = "black", binwidth = 1
 labs(x = "First Year Exam Score (TMU)", y = "Density") +
 stat_function(fun=dnorm, args=list(mean = mean(TMUdata$exam, na.rm = TRUE),
             sd = sd(TMUdata$exam, na.rm = TRUE)), colour = "red", size=1)
hist.exam.TMU
    #ggsave(file = paste(imageDirectory,"TMU exam Hist.png",sep="/"))

hist.exam.NTU <- ggplot(NTUdata, aes(exam)) + theme(legend.position = "none") +
 geom_histogram(aes(y = ..density..), fill = "white", colour = "black", binwidth = 1
 labs(x = "First Year Exam Score (NTU)", y = "Density") +
 stat_function(fun=dnorm, args=list(mean = mean(NTUdata$exam, na.rm = TRUE),
              sd = sd(NTUdata$exam, na.rm = TRUE)), colour = "red", size=1)
hist.exam.NTU
    #ggsave(file = paste(imageDirectory,"NTU exam Hist.png",sep="/"))
```






```
### 3-3(1).[ Q-Q plot for normality check (visually) ]
# → Function: qqnorm {stats}

qqnorm(rexam$exam);qqline(rexam$exam,col='red')
#ggsave(file = paste(imageDirectory,"Exam QQ.png",sep="/"))
```



In [32]:

```
### 3-3(2).[ Shapiro-Wilks test for exam and numeracy for whole sample ]
# → Function: shapiro.test {stats}

shapiro.test(rexam$exam)
```

```
	Shapiro-Wilk normality test

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

In [33]:

```
### 3-3(3).[ Shapiro-Wilks test for exam and numeracy split by university ]
# → Function: by {base}

by(rexam$exam, rexam$uni, shapiro.test)
by(rexam$exam, rexam$uni, mean)
```

```
rexam$uni: NTU

	Shapiro-Wilk normality test

data:  dd[x, ]
W = 0.97217, p-value = 0.2829

------------------------------------------------------------
rexam$uni: TMU

	Shapiro-Wilk normality test

data:  dd[x, ]
W = 0.98371, p-value = 0.7151

rexam$uni: NTU
[1] 40.18
------------------------------------------------------------
rexam$uni: TMU
[1] 76.02
```

In [37]:

```
### 3-4.[ Levene test for comparing exam scores in the two universities. ]
# → Function: leveneTest {car}

leveneTest(rexam$exam, rexam$uni)
```

A anova: 2 × 3

	Df	F value	Pr(>F)
	<int>	<dbl>	<dbl>
group	1	2.088557	0.1515963
	98	NA	NA

In [66]:

```
### 3-5.[ Kolmogorov-Smirov test for exam and numeracy for whole sample ]
# → Function: ks.test {stats}

ks.test(rexam$exam, "pnorm", mean(rexam$exam), sd(rexam$exam))
#ks.test(rexam$numeracy, "pnorm", mean(rexam$numeracy), sd(rexam$numeracy))
```

```
Warning message in ks.test(rexam$exam, "pnorm", mean(rexam$exam), sd(r
exam$exam)):
“ties should not be present for the Kolmogorov-Smirnov test”


        One-sample Kolmogorov-Smirnov test

data:  rexam$exam
D = 0.1021, p-value = 0.2482
alternative hypothesis: two-sided
```

(4) Log-transform data into Normal distribution

In [63]:

```
### 4-1.[ Data preparation ]

data("USJudgeRatings")
df <- USJudgeRatings
head(df)
```

A data.frame: 6 × 12

	CONT	INTG	DMNR	DILG	CFMG	DECI	PREP	FAMI	ORAL	WRIT	PHYS
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
AARONSON,L.H.	5.7	7.9	7.7	7.3	7.1	7.4	7.1	7.1	7.1	7.0	8.[illegible]
ALEXANDER,J.M.	6.8	8.9	8.8	8.5	7.8	8.1	8.0	8.0	7.8	7.9	8.[illegible]
ARMENTANO,A.J.	7.2	8.1	7.8	7.8	7.5	7.6	7.5	7.5	7.3	7.4	7.[illegible]
BERDON,R.I.	6.8	8.8	8.5	8.8	8.3	8.5	8.7	8.7	8.4	8.5	8.[illegible]
BRACKEN,J.J.	7.3	6.4	4.3	6.5	6.0	6.2	5.7	5.7	5.1	5.3	5.[illegible]
BURNS,E.B.	6.2	8.8	8.7	8.5	7.9	8.0	8.1	8.0	8.0	8.0	8.[illegible]

In [64]:

```
### 4-2.[ Descriptive stat. & plots ]

#----- Distribution of CONT variable -----#
skewness(df$CONT) # function of the library {agricolae}
shapiro.test(df$CONT)

cont.org <- ggdensity(df, x = "CONT", fill = "lightgray", title = "CONT") +
  stat_overlay_normal_density(color = "red", linetype = "dashed")

#----- Distribution of PHYS variable -----#
skewness(df$PHYS)
shapiro.test(df$PHYS)

phys.org <- ggdensity(df, x = "PHYS", fill = "lightgray", title = "PHYS") +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

1.12562527407174

```
	Shapiro-Wilk normality test

data:  df$CONT
W = 0.93274, p-value = 0.01445
```

-1.61511156303645

```
	Shapiro-Wilk normality test

data:  df$PHYS
W = 0.85343, p-value = 6.376e-05
```

In [65]:

```
### 4-3.[ Log transformation ]

df$CONT <- log10(df$CONT)
df$PHYS <- log10(max(df$CONT+1) - df$CONT)

#----- Distribution of CONT variable -----#
skewness(df$CONT)
shapiro.test(df$CONT)

cont.log <- ggdensity(df, x = "CONT", fill = "lightgray", title = "CONT") +
  stat_overlay_normal_density(color = "red", linetype = "dashed")

#----- Distribution of PHYS variable -----#
skewness(df$PHYS)
shapiro.test(df$PHYS)

phys.log <- ggdensity(df, x = "PHYS", fill = "lightgray", title = "PHYS") +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

0.67949395956936

```
	Shapiro-Wilk normality test

data:  df$CONT
W = 0.96773, p-value = 0.2635
```

-0.847918759893301

```
	Shapiro-Wilk normality test

data:  df$PHYS
W = 0.95618, p-value = 0.1004
```

In [61]:

```
# ::::::::::::::::::::::::::::::::::::::::::::::::::
# Arrange of GGplot figure
# ::::::::::::::::::::::::::::::::::::::::::::::::::

ggarrange(cont.org, phys.org, cont.log, phys.log, ncol = 2, nrow = 2)
```