R Code to Complement Math Review Packet

A Note:

The notes and code in this document are meant to serve as complementary material to the Math Review Packet. In this program, you will be able to perform calculations in R…

Properties of Logarithms

In R the log() function computed logarithms by natural logarithms ($ln()$) to calculate logarithms with different bases such as log_{10} use log10(), log_2 use log2(), etc…

For example if we wanted to solve $log_{10}10000$:

```
log10(10000)
```

```
## [1] 4
```

Additionally if we wanted to solve e^4, we would use the exp()

```
exp(4)
```

```
## [1] 54.59815
```

Practice Problems.

1. Solve the following:

a. $log_e(e^x) = x$

```
# If x = 3
log(exp(3))
```

```
## [1] 3
```

b. $log_{10}(100) = 2$

```
log10(100)
```

```
## [1] 2
```

c. $log_{10}(\frac{1}{10}) = -1$

```
log10(1/10)
```

```
## [1] -1
```

d. $log_{10}(0) =$ No solution

```
log10(0)
```

```
## [1] -Inf
```

3 Condense the following c. $log_{10}(5) + log_{10}(2) = 1$

```
log10(5) + log10(2)
```

```
## [1] 1
```

Matrix Algebra

Creating a Matrix in R

Type ?matrix in the **Condole** or matrix in **Help** to look at the inputs of the matrix function.

If we want to store the matrix we need give a name to the matrix, "Z" for example, and store the matrix.

For example $\mathbf{Z} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ $\mathbf{Y} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$

```
#Sorting by row
Z <- matrix(data = c(1,2,3,4,5,6,7,8,9),nrow = 3, ncol = 3, byrow = TRUE)
print(Z)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

```
#Sorting by column
Y <- matrix(data = (1:9), nrow = 3, ncol = 3, byrow = FALSE)
print(Y)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

Definitions

The transpose of an $m \times n$ matrix, A is the $n \times m$ matrix (denoted A^T) such that every element a_{ij} in matrix A is moved to row j and column i. For example, if:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 7 \end{bmatrix}$$

then,

$$\mathbf{A}^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \\ 5 & 7 \end{bmatrix}$$

```
# Matrix A
A <- matrix(data = c(1,2,5,3,4,7), nrow = 2, ncol = 3, byrow = TRUE)

print(A)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    5
## [2,]    3    4    7
```

```
# A transpose
print(t(A))
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
## [3,]    5    7
```

Identity Matrix

```r
# Identity Matrix: use diag() to create an identity matrix
# I = 4x4 identity matrix
I <- diag(4)
print(I)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

```r
class(I)
```

```
## [1] "matrix"
```

Diagonal

```r
X1 <- matrix(data = c(1,0,0,0,4,0,0,0,5), nrow = 3, byrow = TRUE)
print(X1)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    4    0
## [3,]    0    0    5
```

```r
# To obtain the obtain the values of the diagonal of an n x n matrix
diag(X1)
```

```
## [1] 1 4 5
```

```r
# We could also create an empty matrix and assign values to the diagonal
X2 <- matrix(data = 0, nrow = 3, ncol = 3)
diag(X2) <- c(1,4,5)
print(X2)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    4    0
## [3,]    0    0    5
```

Upper and Lower Triangular Matrices

```r
# Create an empty matrix
V <- W <- matrix(data = 0, nrow = 3, ncol = 3, byrow = TRUE)
print(W)
```

```
##      [,1] [,2] [,3]
## [1,]    0    0    0
## [2,]    0    0    0
## [3,]    0    0    0
```

```r
# Use upper.tri() or lower.tri() to change the elements of the upper
# triangular matrix or lower triangular matrix, respectively.

# Upper Triangular
W[upper.tri(W, diag = TRUE)] <- c(1:6)
print(W)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    4
## [2,]    0    3    5
## [3,]    0    0    6
```

```
# Lower Triangular
V[lower.tri(V, diag = TRUE)] <- c(1:6)
print(V)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    2    4    0
## [3,]    3    5    6
```

Inverse of a matrix

```
U <- matrix(data = c(1:4), nrow = 2, byrow = TRUE)

# Use solve() to obtain the inversie of a matrix
inU <- solve(U)

# Confirm you get an identity matrix
round(U %*% inU, 2)
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

```
round(inU %*% U, 2)
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

Determinant of a matrix

```
# Use det() to obtain the determinant of a matrix
det(U)
```

```
## [1] -2
```

Operations with matrices

a. Adding matices

```
# Using the above matrices (Z and Y) we will show that Z + Y = X
# Adding matrices of the same dimension will produce a matrix of the same dimension
dim(Z + Y); dim (Z); dim(Y)
```

```
## [1] 3 3

## [1] 3 3

## [1] 3 3
```

```
Z + Y
```

```
##      [,1] [,2] [,3]
## [1,]    2    6   10
## [2,]    6   10   14
```

```
## [3,]   10   14   18
```

b. Subtracting matrices

```r
# Subtracting matrices of the same dimension will produce a matrix of the same dimension
dim(Z - Y); dim (Z); dim(Y)
```

```
## [1] 3 3

## [1] 3 3

## [1] 3 3
```

```r
Z - Y
```

```
##      [,1] [,2] [,3]
## [1,]    0   -2   -4
## [2,]    2    0   -2
## [3,]    4    2    0
```

c. Multiplying a matrix by a scalar

```r
# Multiplying a matrix by a scaler will produce a matrix of the same dimension
# Suppose that c = 3

3 * Z
```

```
##      [,1] [,2] [,3]
## [1,]    3    6    9
## [2,]   12   15   18
## [3,]   21   24   27
```

d. Matrix Multiplication

```r
m1 <- matrix(data = c(3,4,2,5,1,2), nrow = 3, ncol = 2, byrow = TRUE)
print (m1) ; dim(m1)
```

```
##      [,1] [,2]
## [1,]    3    4
## [2,]    2    5
## [3,]    1    2

## [1] 3 2
```

```r
m2 <- matrix(data = c(7,2,1,3,5,2), nrow = 2, ncol = 3, byrow = TRUE)
print(m2); dim(m2)
```

```
##      [,1] [,2] [,3]
## [1,]    7    2    1
## [2,]    3    5    2

## [1] 2 3
```

```r
# Matrix multiplication
m1 %*% m2
```

```
##      [,1] [,2] [,3]
## [1,]   33   26   11
## [2,]   29   29   12
## [3,]   13   12    5
```

```r
# It is important to note that when multiplying matrices we need to use %*% rather than * because using
print(Z); print(Y)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
Z * Y
```

```
##      [,1] [,2] [,3]
## [1,]    1    8   21
## [2,]    8   25   48
## [3,]   21   48   81
```

```
Z %*% Y
```

```
##      [,1] [,2] [,3]
## [1,]   14   32   50
## [2,]   32   77  122
## [3,]   50  122  194
```

Properties of matrix operations

1. $A + B = B + A$

```
Z + Y
```

```
##      [,1] [,2] [,3]
## [1,]    2    6   10
## [2,]    6   10   14
## [3,]   10   14   18
```

```
Y + Z
```

```
##      [,1] [,2] [,3]
## [1,]    2    6   10
## [2,]    6   10   14
## [3,]   10   14   18
```

2. $(A + B) + C = A + (B + C)$

```
(Z + Y) + X1
```

```
##      [,1] [,2] [,3]
## [1,]    3    6   10
## [2,]    6   14   14
## [3,]   10   14   23
```

```
Z + (Y + X1)
```

```
##      [,1] [,2] [,3]
## [1,]    3    6   10
## [2,]    6   14   14
## [3,]   10   14   23
```

3. $(AB)C = A(BC)$

```
(Z %*% Y) %*% X1
```

```
##      [,1] [,2] [,3]
## [1,]   14  128  250
## [2,]   32  308  610
## [3,]   50  488  970
```

```r
Z %*% (Y %*% X1)
```

```
##      [,1] [,2] [,3]
## [1,]   14  128  250
## [2,]   32  308  610
## [3,]   50  488  970
```

4. $(A + B)C = AC + BC$

```r
(Z + Y) %*% X1
```

```
##      [,1] [,2] [,3]
## [1,]    2   24   50
## [2,]    6   40   70
## [3,]   10   56   90
```

```r
(Z %*% X1) + (Y %*% X1)
```

```
##      [,1] [,2] [,3]
## [1,]    2   24   50
## [2,]    6   40   70
## [3,]   10   56   90
```

5. If A is an $m \times n$ matrix, then $I_m A = A$ and $AI_n = A$

```r
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    5
## [2,]    3    4    7
```

```r
Im <- diag(x = 1, nrow = 2)
In <- diag(x = 1, nrow = 3)

Im %*% A
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    5
## [2,]    3    4    7
```

```r
A %*% In
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    5
## [2,]    3    4    7
```

Note that, in general $AB \neq BA$

```r
Z %*% Y
```

```
##      [,1] [,2] [,3]
## [1,]   14   32   50
## [2,]   32   77  122
## [3,]   50  122  194
```

```r
Y %*% Z
```

```
##      [,1] [,2] [,3]
## [1,]   66   78   90
## [2,]   78   93  108
## [3,]   90  108  126
```

Writing a system of equations using matrix notation

***** SOPHIE LOOK HERE ***** I don't really know how to show this!

```r
a1 <- rep(0, 100)
a2 <- rnorm(100, 0, 1)
a3 <- 2*rnorm(100, 0, 2)
A <- cbind(a1, a2, a3)
dim(A)
```

```
## [1] 100   3
```

```r
X <- as.matrix(c(1:3))
dim(X)
```

```
## [1] 3 1
```

```r
Y <- A %*% X
dim(Y)
```

```
## [1] 100   1
```

Practice Problems

1. Solve the following:

a. $\begin{bmatrix} 2 & 4 & 2 \\ 1 & 3 & 0 \\ 1 & 6 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 5 & 0 \\ -2 & -3 & 0 \\ 1 & 9 & 5 \end{bmatrix} = \begin{bmatrix} 3 & 9 & 2 \\ -1 & 0 & 0 \\ 2 & 15 & 7 \end{bmatrix}$

```r
M1 <- matrix(data = c(2,4,2,1,3,0,1,6,2),
             nrow = 3, ncol = 3, byrow = TRUE)

M2 <- matrix(data = c(1,5,0,-2,-3,0,1,9,5),
             nrow = 3, ncol = 3, byrow = TRUE)

M1 + M2
```

```
##      [,1] [,2] [,3]
## [1,]    3    9    2
## [2,]   -1    0    0
## [3,]    2   15    7
```

b. $\begin{bmatrix} 2 & 1 \\ -2 & 2 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 5 & 2 & -1 \\ 3 & 4 & 2 \end{bmatrix} = \begin{bmatrix} 13 & 8 & 0 \\ -4 & 4 & 6 \\ 26 & 16 & 0 \end{bmatrix}$

```r
M3 <- matrix(data = c(2,-2,4,1,2,2), nrow = 3, ncol = 2, byrow = FALSE)

M4 <- matrix(data = c(5,3,2,4,-1,2), nrow = 2, ncol = 3, byrow = FALSE)

M3 %*% M4
```

```
##      [,1] [,2] [,3]
## [1,]   13    8    0
## [2,]   -4    4    6
## [3,]   26   16    0
```

c. Let $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \\ 4 & 0 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 4 & 4 \\ 1 & 2 \\ 7 & 0 \end{bmatrix}$ $\mathbf{A}^T\mathbf{B} = \begin{bmatrix} 35 & 10 \\ 13 & 18 \end{bmatrix}$

```r
A <- matrix(data = c(1,2,3,5,4,0), nrow = 3, ncol = 2, byrow = TRUE)

B <- matrix(data = c(4,1,7,4,2,0),nrow = 3, ncol=2, byrow = FALSE)

t(A) %*% B
```

```
##      [,1] [,2]
## [1,]   35   10
## [2,]   13   18
```

d. Using the same matrices as in part c, $\mathbf{B}^T\mathbf{A} = \begin{bmatrix} 35 & 13 \\ 10 & 18 \end{bmatrix}$

```r
t(B) %*% A
```

```
##      [,1] [,2]
## [1,]   35   13
## [2,]   10   18
```

Show that **A** and **B** are inverses: $\begin{bmatrix}1 & 2 & 1\\ 2 & 2 & 0\\ 1 & 1 & 1\end{bmatrix}\begin{bmatrix}-1 & 0.5 & 1\\ 1 & 0 & -1\\ 0 & -0.5 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$

```
A <- matrix(data = c(1,2,1,2,2,0,1,1,1), nrow = 3, ncol = 3, byrow = TRUE)

#Use the solve() to find the inverse of a matrix
solve(A)
```

```
##      [,1] [,2] [,3]
## [1,]   -1  0.5    1
## [2,]    1  0.0   -1
## [3,]    0 -0.5    1
```

```
B <- matrix(data = c(-1, 0.5, 1, 1, 0, -1, 0, -0.5, 1), nrow = 3, ncol = 3, byrow = TRUE)

#Use the solve() to find the inverse of a matrix
solve(B)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    1
## [2,]    2    2    0
## [3,]    1    1    1
```

```
A %*% B
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

3. Suppose that A is a 4x3 matrix and B is a $3x8$ matrix.

```
A <- matrix (data= (1:12), nrow = 4, ncol = 3, byrow = FALSE)
B <- matrix (data = (1:24), nrow = 3, ncol = 8, byrow = TRUE)
```

a. **AB** exists and is 4x8 matrix.

```
A %*% B
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]  199  214  229  244  259  274  289  304
## [2,]  226  244  262  280  298  316  334  352
## [3,]  253  274  295  316  337  358  379  400
## [4,]  280  304  328  352  376  400  424  448
```

```
dim(A%*%B)
```

```
## [1] 4 8
```

b. **BA** does not exist.

```
# The following produce errors
# B %*% A
```

4. The determinant of $det \begin{bmatrix}1 & -2\\ 4 & 3\end{bmatrix} = 11$

```
M5 <- matrix(data = c(1, -2, 4, 3), nrow = 2, ncol = 2, byrow = TRUE)
det(M5)
```

```
## [1] 11
```

Variables : Types and Summaries

Summary statistics for central tendency

```r
dat <- c(1, 2, 3, 5, 5, 4, 6, 6, 1, 2, 10, 11, 9, 9, 9)

# Mean
mean(dat)
```

```
## [1] 5.533333
```

```r
# Median
median(dat)
```

```
## [1] 5
```

```r
# Mode -- R does not have a built in function for mode but you can use the
# table function to see what the most common value is
table(dat)
```

```
## dat
##  1  2  3  4  5  6  9 10 11
##  2  2  1  1  2  2  3  1  1
```

Summary statistics for spread

```r
# Range
range(dat)[2]-range(dat)[1]
```

```
## [1] 10
```

```r
diff(range(dat))
```

```
## [1] 10
```

```r
# Sample Variance
var(dat)
```

```
## [1] 11.55238
```

```r
# Standard Deviation
sd(dat)
```

```
## [1] 3.398879
```

```r
sqrt(var(dat))
```

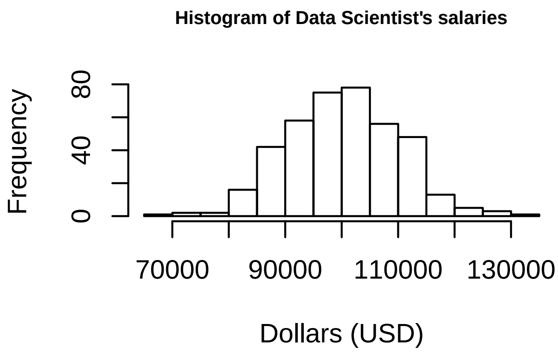
```
## [1] 3.398879
```

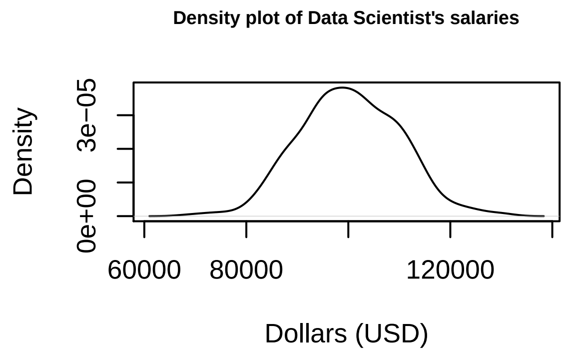
Central Limit Theorem Introduction

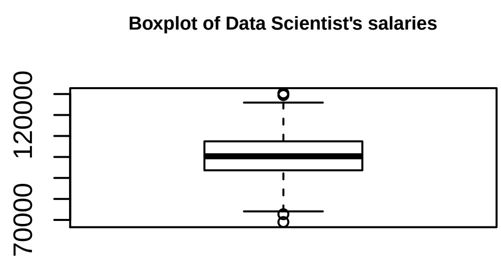
We can sample from many distributions in R. The following will walk through sampling from a normal distribution.

In the example with average salary of "Data Scientists" we have a sample (n = 400), mean of 100,000 and standard deviation of 10,000.

```
DS_salary <- rnorm(n = 400, mean = 1e+05, sd = 10000)
par(mfrow = c(2, 2))
hist(DS_salary, main = "Histogram of Data Scientist's salaries", cex.main = 0.7,
    xlab = "Dollars (USD)", breaks = 10)
plot(density(DS_salary), main = "Density plot of Data Scientist's salaries",
    cex.main = 0.7, xlab = "Dollars (USD)")
boxplot(DS_salary, main = "Boxplot of Data Scientist's salaries", cex.main = 0.7)
```







Confidence Intervals (an example)

```
mean_ <- 100000
sd_ <- 10000
n <- 400
error <- qnorm(0.975)*sd_/sqrt(n)
mean_ - error # lower bound of confidence interval
```

```
## [1] 99020.02
```

```
mean_ + error # upper bound of confidence interval
```

```
## [1] 100980
```

Z-Tests, T-Test, and P-Values

$H_0 = \mu_{school} = \mu_{population}$ $H_a = \mu_{school} \neq \mu_{population}$

Z-Tests and P-Values

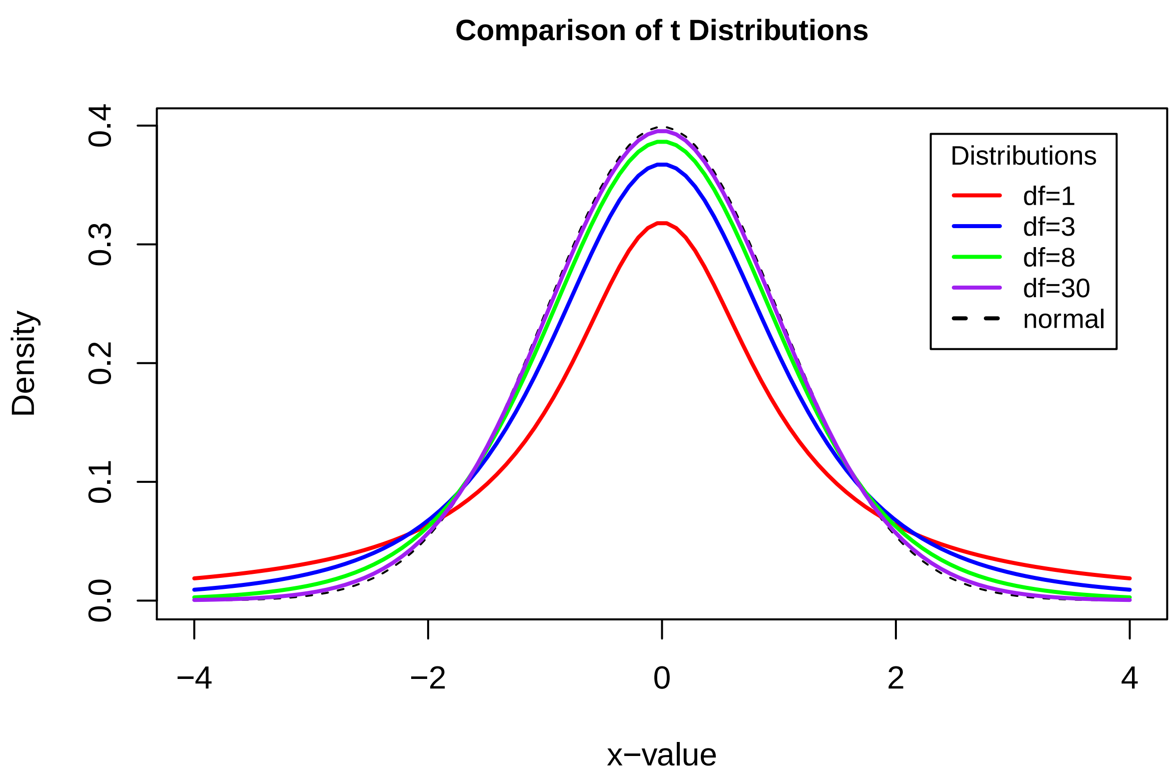
```r
# Find the proportion of sample means that is 2 points greater than or less
# than the state average
1 - pnorm(72, mean = 70, sd = 1)
```

```
## [1] 0.02275013
```

The t-distribution and T-tests

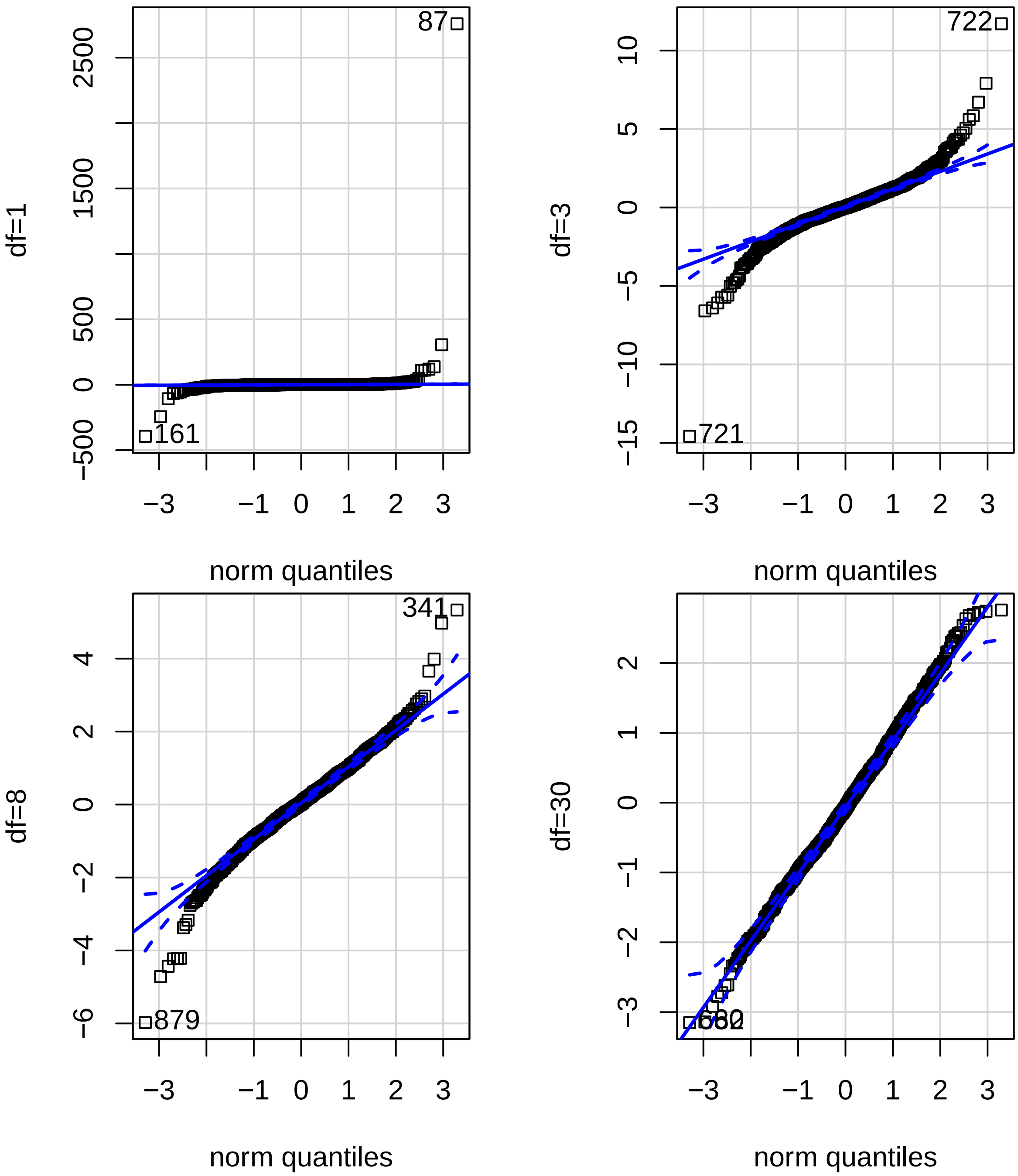
The Student's t-distribution with different degrees of freedom compared to the normal distribution

```r
x <- seq(-4, 4, length=100)
zdist <- dnorm(x)

degf <- c(1, 3, 8, 30)
colors <- c("red", "blue", "green", "purple", "black")
labels <- c("df=1", "df=3", "df=8", "df=30", "normal")

plot(x, zdist, type="l", lty=2, xlab="x-value",
  ylab="Density", main="Comparison of t Distributions", cex.main=0.9)

for (i in 1:4){
  lines(x, dt(x,degf[i]), lwd=2, col=colors[i])
}

legend("topright", inset=.05, title="Distributions",
  labels, lwd=2, lty=c(1, 1, 1, 1, 2), col=colors, cex= 0.8)
```




```r
par(mfrow=c(1,2))
```

```
for (i in 1:4){
  car::qqPlot(rt(1000, df = degf[i]), ylab = labels[i], pch=22)
}
```




```
1-pnorm(72, 70, 1, lower.tail = TRUE)
```

```
## [1] 0.02275013
```

Correlation and Covariance

Simple Ordinary Least Squares Regression