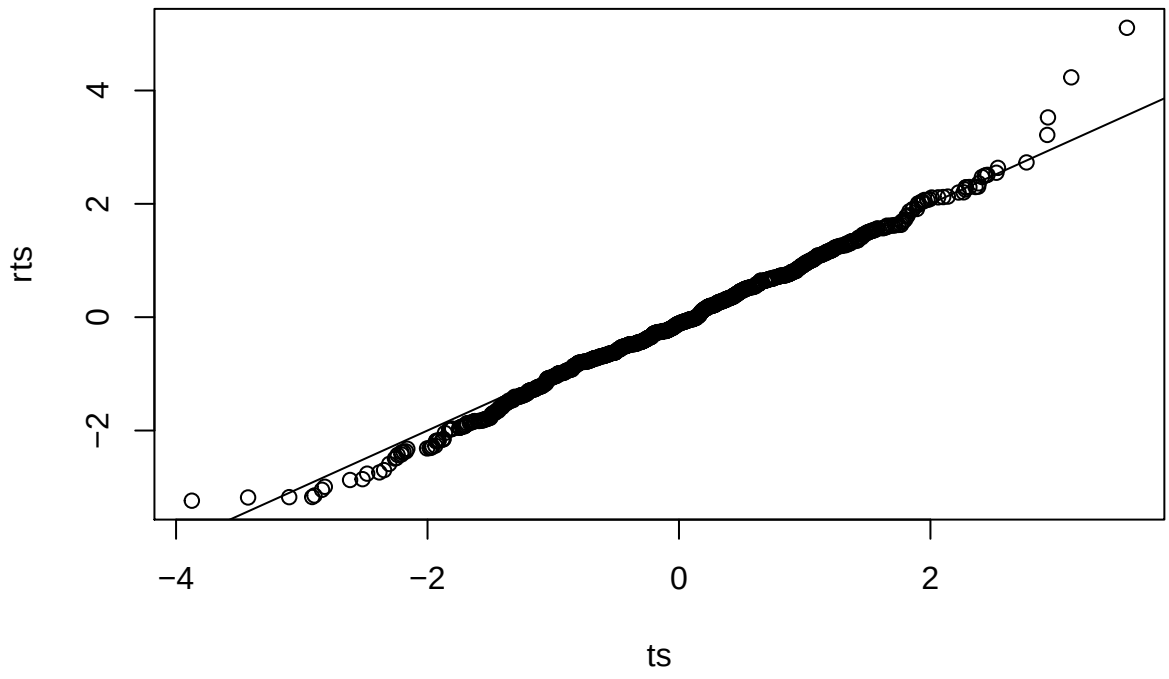
Coding and data analysis exercises

1.

```
samples = rnorm(1000*20, 5, sqrt(2))
samples = matrix(samples,1000,20)
ms = apply(samples, 1, mean)
stds = sqrt(apply(samples, 1, var))
ts = sqrt(19) * (ms - 5)  / stds

rts = rt(1000, 19)

qqplot(ts, rts)
abline(0,1)
```



As shown in the graph, the quantiles agree well, since every t statistics is computed from 20 i.i.d normal samples, it will follow a t distribution with 19 df. So the quantiles will agree.

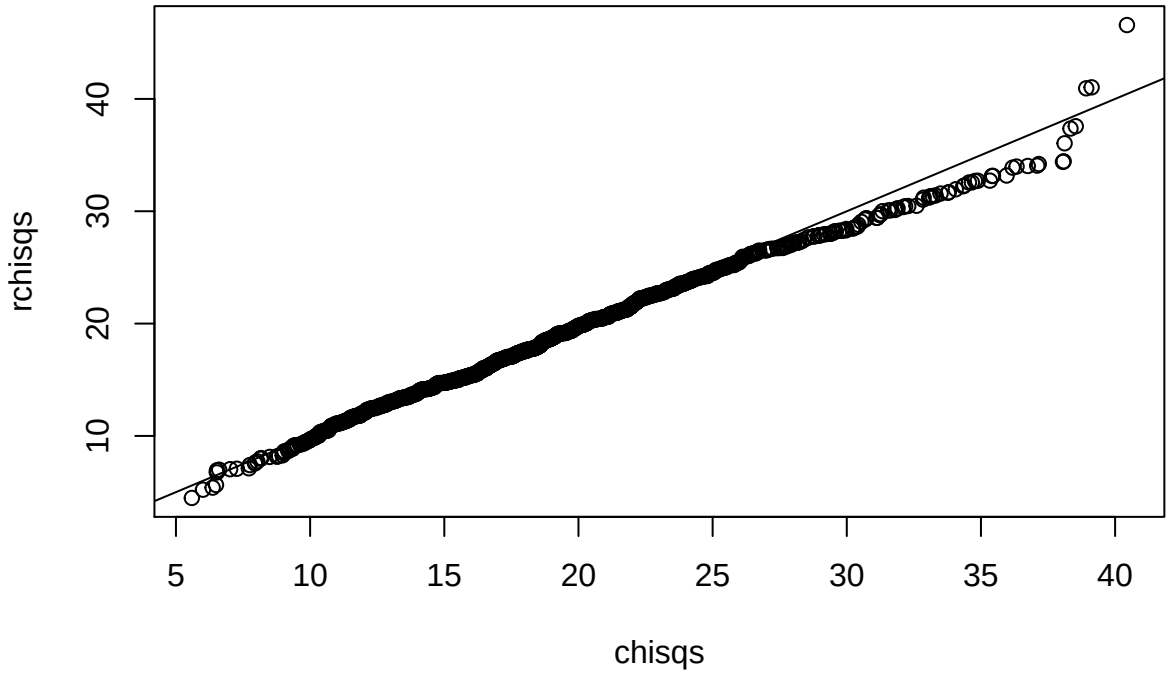
2.

```
samples = rnorm(1000*20, 5, sqrt(2))
samples = matrix(samples,1000,20)
vs = apply(samples, 1, var)
```

```r
chisqs = 19 * vs / 2

rchisqs = rchisq(1000, 19)

qqplot(chisqs, rchisqs)
abline(0,1)
```



As shown in the graph, the quantiles agree well, also because they are just from the same distribution.

3.

```r
require(stats)

mylm <- function(Y,X){

    Y = as.matrix(Y); Xnew = as.matrix(X)

    # Check if numeric
    if(!is.numeric(Y) | !is.numeric(Xnew))
        stop("Y or X is not numeric!\n")

    # Check for dimensions
    dy = dim(Y) ; dx = dim(Xnew)
    if(dy[2] != 1 | dy[1] != dx[1])
        stop("Y or X has wrong dimensions\n")
```

```r
# Check for ill conditioned elements
# we can use is.finite to response only to finite real numbers
if(FALSE %in% is.finite(Y) | FALSE %in% is.finite(Xnew))
    warning("Y or X is ill conditioned\n")

# Check if of full rank
D = cbind(1,Xnew)
DtD = t(D) %*% D
if(det(DtD) == 0)
    stop("Design matrix is not full rank\n")

# Regressing
DtD.inv = solve(DtD)
hat.matrix = D %*% DtD.inv %*% t(D)
beta = DtD.inv %*% t(D) %*% Y
fitted = D %*% beta
residuals = Y - fitted

SS.tot = sum((Y - mean(Y))^2)
if(SS.tot == 0)
    warning("Y is constant!\n")
SS.res = sum((Y - fitted)^2)
SS.reg = SS.tot - SS.res
R2 = SS.reg / SS.tot

df = dim(D)[1] - dim(D)[2]
df2 = dim(D)[2] - 1

s2 = SS.res / df
std_error = sqrt(s2 * diag(DtD.inv))
t_value = beta / std_error
P_value = 1 - pt(abs(t_value), df) + pt(-abs(t_value), df)

K = cbind(rep(0, df2), diag(df2))
Kbeta = K %*% beta
Fstat = t(Kbeta) %*% solve(K %*% DtD.inv %*% t(K)) %*% Kbeta
Fstat = Fstat / (df2 * s2)
P_value_F = 1 - pf(Fstat, df2, df)

beta_names = c("(Interception)")
for(i in 1:(dim(D)[2] - 1)){
    beta_names = c(beta_names, sprintf("beta%d",i))
}
t_summary = data.frame("Estimate"=beta, "Std.Error"=std_error,
                       "t.value"=t_value, "Pvalue"=P_value)
row.names(t_summary) = beta_names

summary <- function(){
    cat("T table:\n")
    print(t_summary)
    cat("\nOverall F test:\n")
    cat(sprintf("F-statistics: %f on %d and %d DF,  p-value: %f",
                Fstat, df2, df, P_value_F))
```

```
    }

    # Return result
    result = list(beta = beta,
                  fitted = fitted,
                  residuals = residuals,
                  R2 = R2,
                  hatdiag = diag(hat.matrix),
                  summary = summary)
    return(result)
}

test.X = cbind(sample(1:100),sample(1:100),sample(1:100))
beta = c(5,-1,0.01,2)
test.y = cbind(1,test.X) %*% beta + rnorm(100,0,5)
model = lm(test.y ~ test.X)
summary(model)
```

```
##
## Call:
## lm(formula = test.y ~ test.X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8314  -3.3723  -0.1633   3.3739  12.5185
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.973250   1.657447   3.001  0.00343 **
## test.X1     -0.987051   0.017115 -57.671  < 2e-16 ***
## test.X2      0.005103   0.017119   0.298  0.76630
## test.X3      2.019717   0.016818 120.090  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.832 on 96 degrees of freedom
## Multiple R-squared:  0.9944, Adjusted R-squared:  0.9942
## F-statistic:  5666 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
mymodel = mylm(test.y, test.X)
mymodel$summary()
```

```
## T table:
##                    Estimate  Std.Error     t.value        Pvalue
## (Intercept)  4.973249946 1.65744736   3.0005477  3.433648e-03
## beta1       -0.987051101 0.01711515 -57.6711825  1.313666e-76
## beta2        0.005102669 0.01711925   0.2980661  7.662969e-01
## beta3        2.019716782 0.01681843 120.0895022 9.733797e-107
##
## Overall F test:
## F-statistics: 5666.304005 on 3 and 96 DF,  p-value: 0.000000
```