# regression

## mohit

## 21/05/2021

```
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method             from
##   as.zoo.data.frame zoo
```

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(dotwhisker)
```

```
## Warning in checkMatrixPackageVersion(): Package version inconsistency detected.
## TMB was built with Matrix version 1.3.3
## Current Matrix version is 1.2.18
## Please re-install 'TMB' from source using install.packages('TMB', type = 'source') or ask CRAN for a
```

```
## Registered S3 method overwritten by 'broom.mixed':
##   method        from
##   tidy.gamlss broom
```

```
library(ggcorrplot)
library(texreg)
```

```
## Version:  1.37.5
## Date:     2020-06-17
## Author:   Philip Leifeld (University of Essex)
##
## Consider submitting praise using the praise or praise_interactive functions.
## Please cite the JSS article in your publications -- see citation("texreg").
```

```
reg1 <- read.csv("E:/course/iazolo/paper/reg.csv", stringsAsFactors=FALSE)
reg2 <- read.csv("E:/course/iazolo/paper/reg2.csv", stringsAsFactors=FALSE)
View
```

```
## function (x, title)
## {
##     check <- Sys.getenv("_R_CHECK_SCREEN_DEVICE_", "")
##     msg <- "View() should not be used in examples etc"
##     if (identical(check, "stop"))
##         stop(msg, domain = NA)
##     else if (identical(check, "warn"))
##         warning(msg, immediate. = TRUE, noBreaks. = TRUE, domain = NA)
##     if (missing(title))
##         title <- paste("Data:", deparse(substitute(x))[1])
##     as.num.or.char <- function(x) {
##         if (is.character(x))
##             x
##         else if (is.numeric(x)) {
##             storage.mode(x) <- "double"
##             x
##         }
##         else as.character(x)
##     }
##     x0 <- as.data.frame(x)
##     x <- as.list(format.data.frame(x0))
##     rn <- row.names(x0)
##     if (any(rn != seq_along(rn)))
##         x <- c(list(row.names = rn), x)
##     if (!is.list(x) || !length(x) || !all(sapply(x, is.atomic)) ||
##         !max(lengths(x)))
##         stop("invalid 'x' argument")
##     if (grepl("darwin", R.version$os))
##         check_for_XQuartz()
##     invisible(.External2(C_dataviewer, x, title))
## }
## <bytecode: 0x000000001ab7eb88>
## <environment: namespace:utils>
```
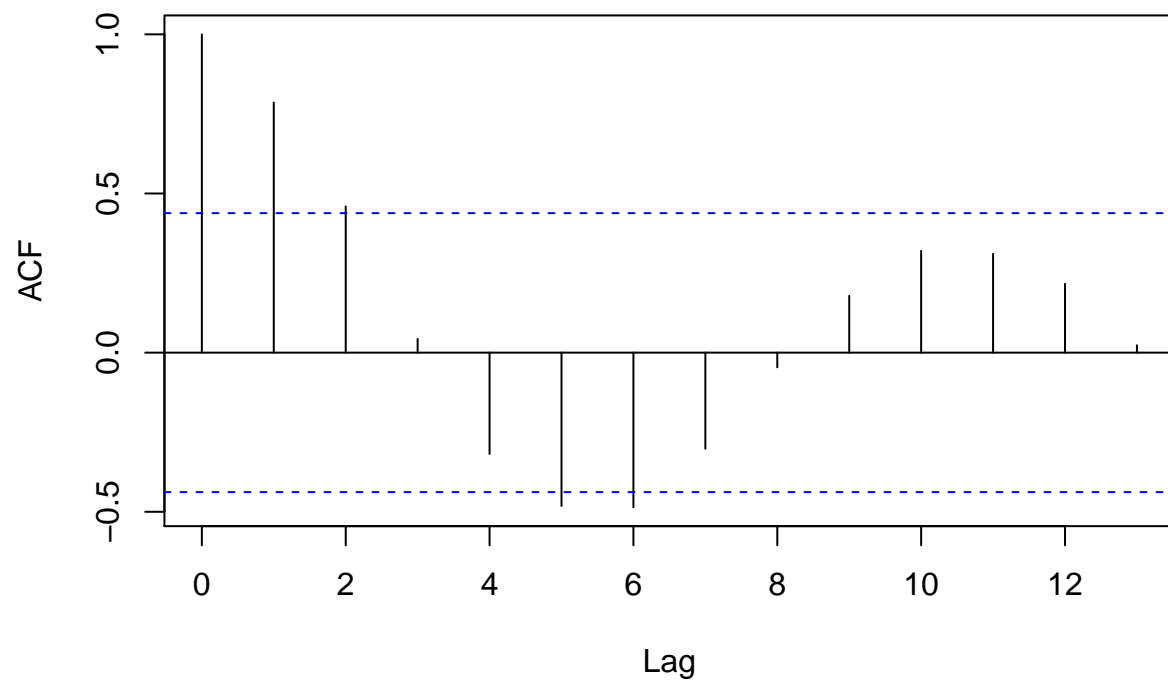
```
## required test before using OLS regression
acf(reg1$debt)
```
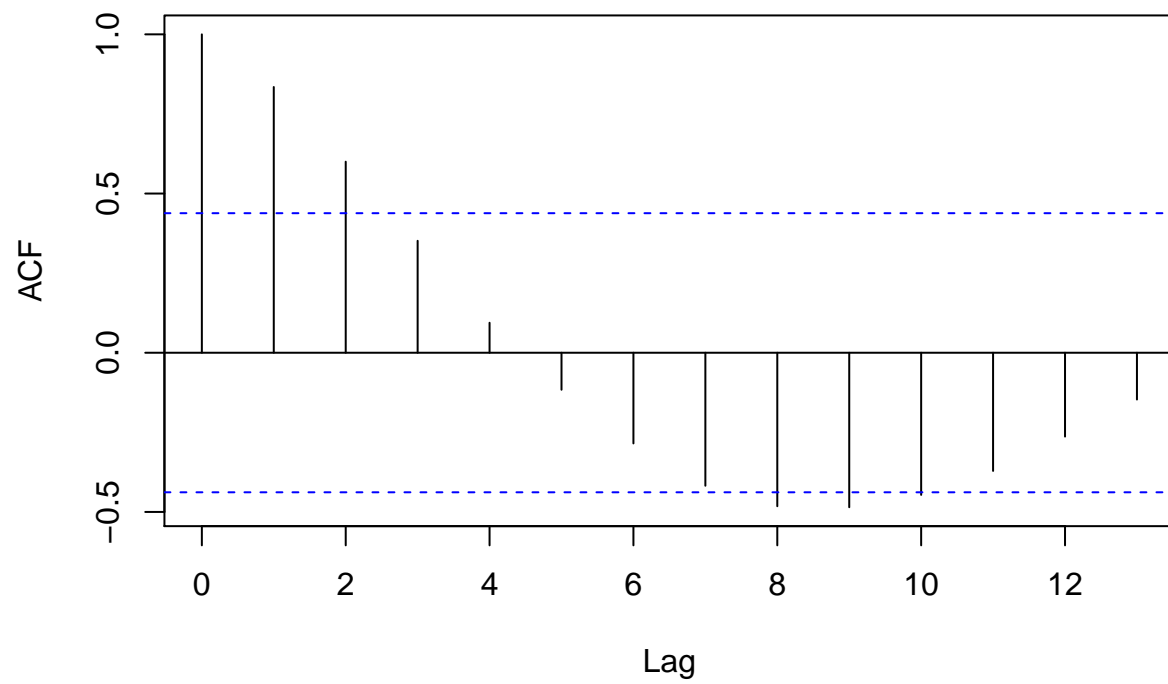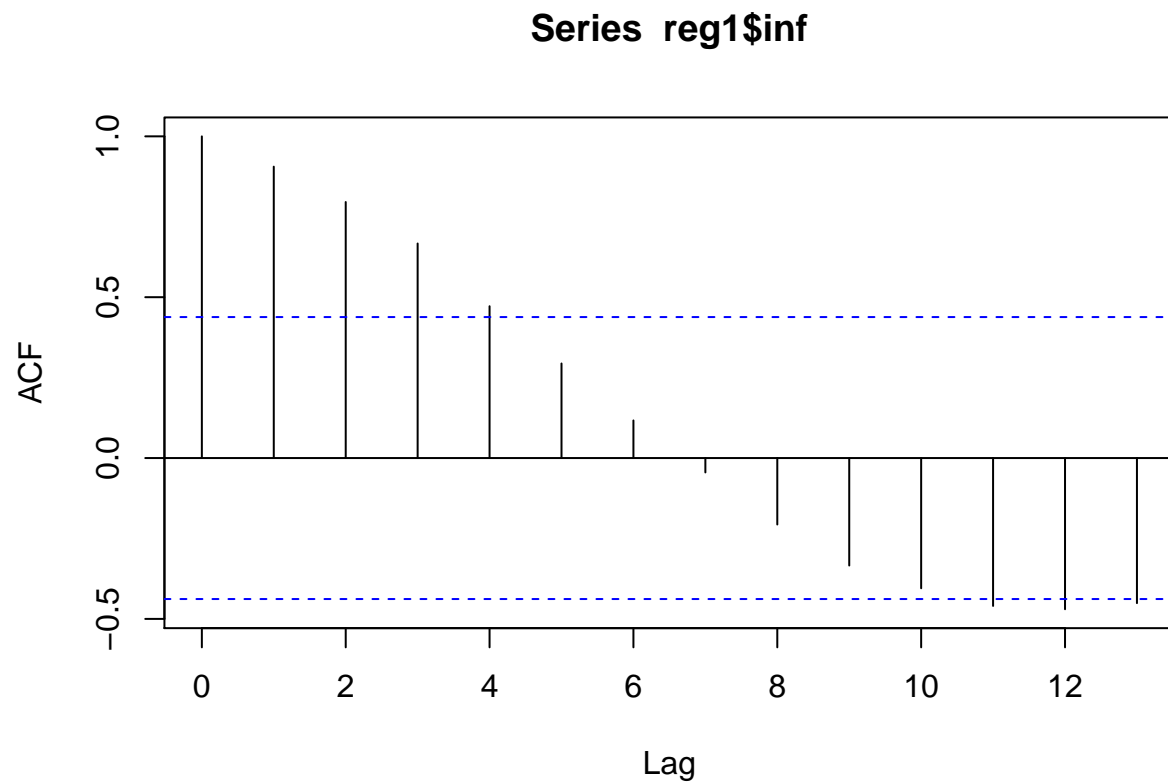
**Series reg1$debt**
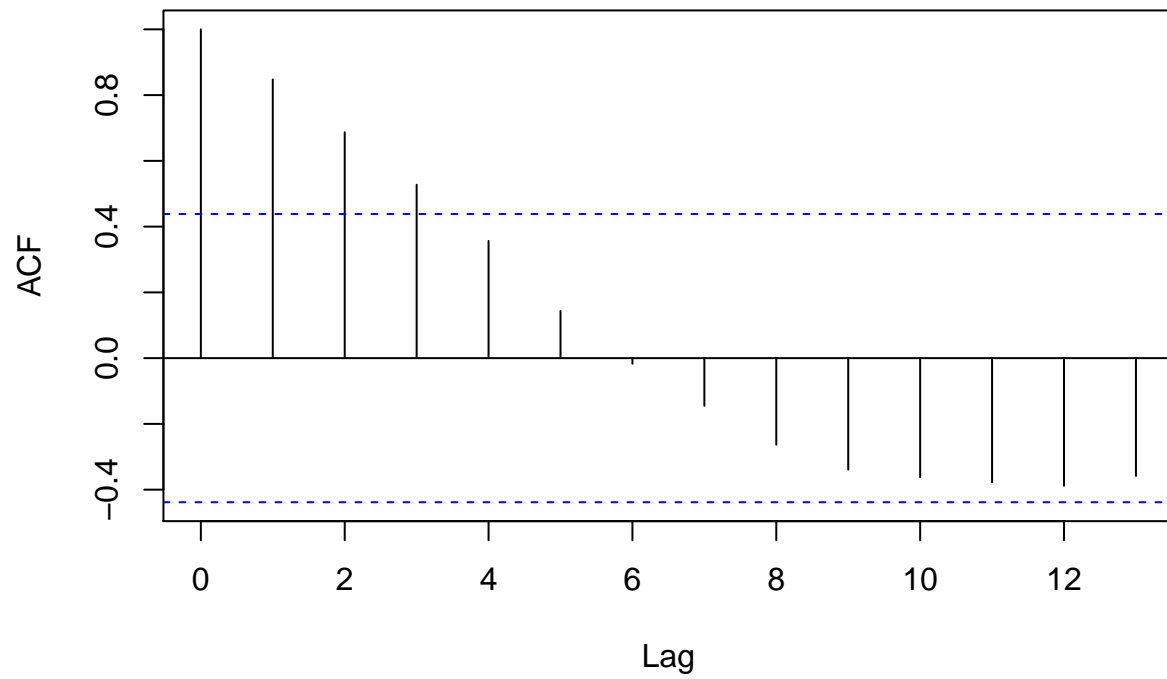


```
acf(reg1$m3)
```

## Series reg1$m3



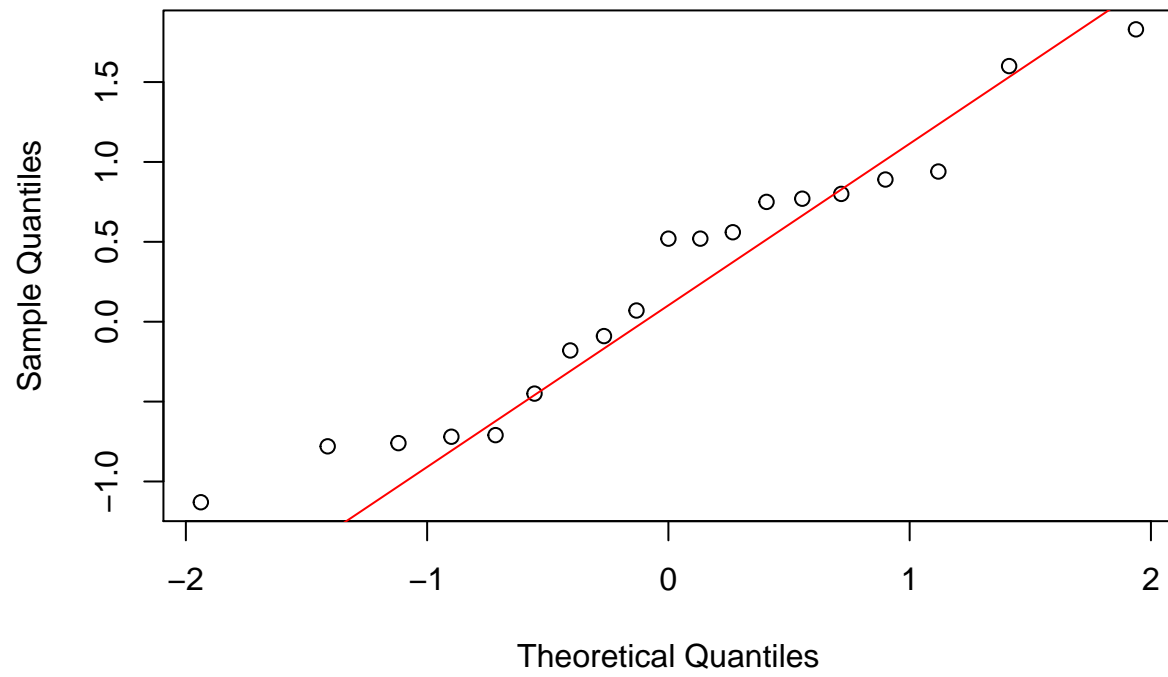```
acf(reg1$emp)
```

**Series reg1$emp**



```
acf(reg1$inf)
```

# Series reg1$inf



```
#### taking first difference to remove acf
ddebt<-diff(reg1$debt)
dinf<-diff(reg1$inf)
demp<-diff(reg1$emp)
dm3<-diff(reg1$m3)
```
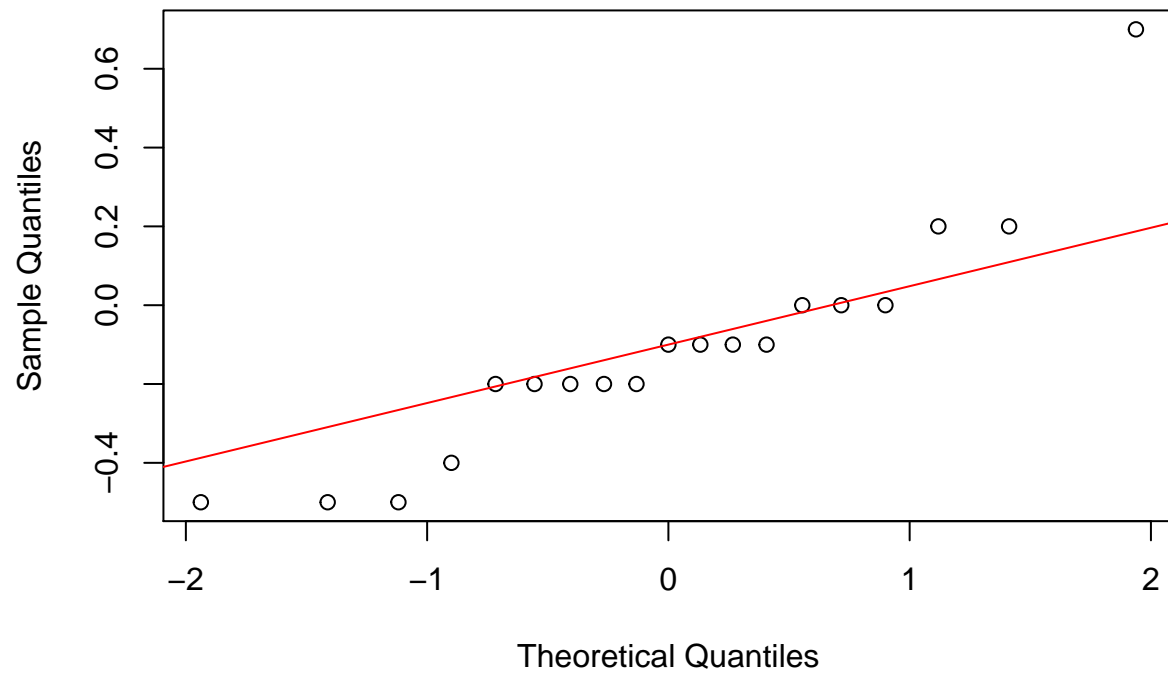
##normality check

```
qqnorm(ddebt);qqline(ddebt, col = 2)
```

# Normal Q–Q Plot



```
qqnorm(dinf);qqline(dinf, col = 2)
```

**Normal Q–Q Plot**



```
qqnorm(demp);qqline(demp, col = 2)
```
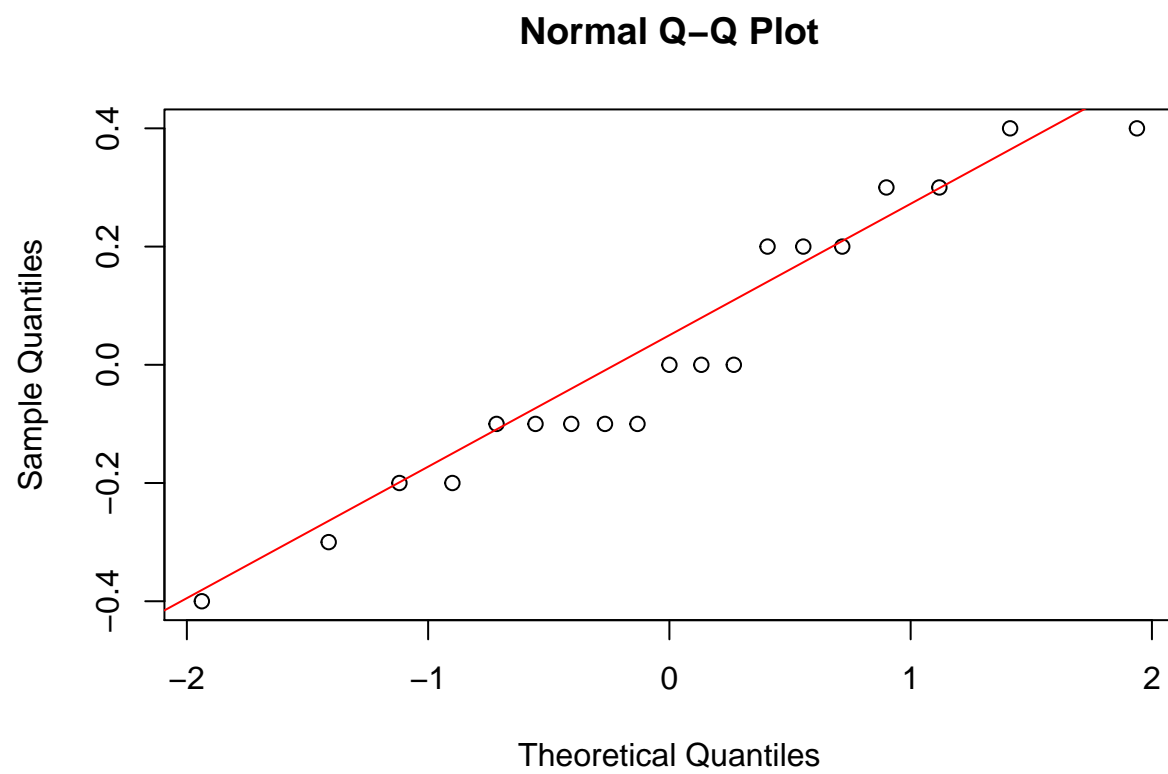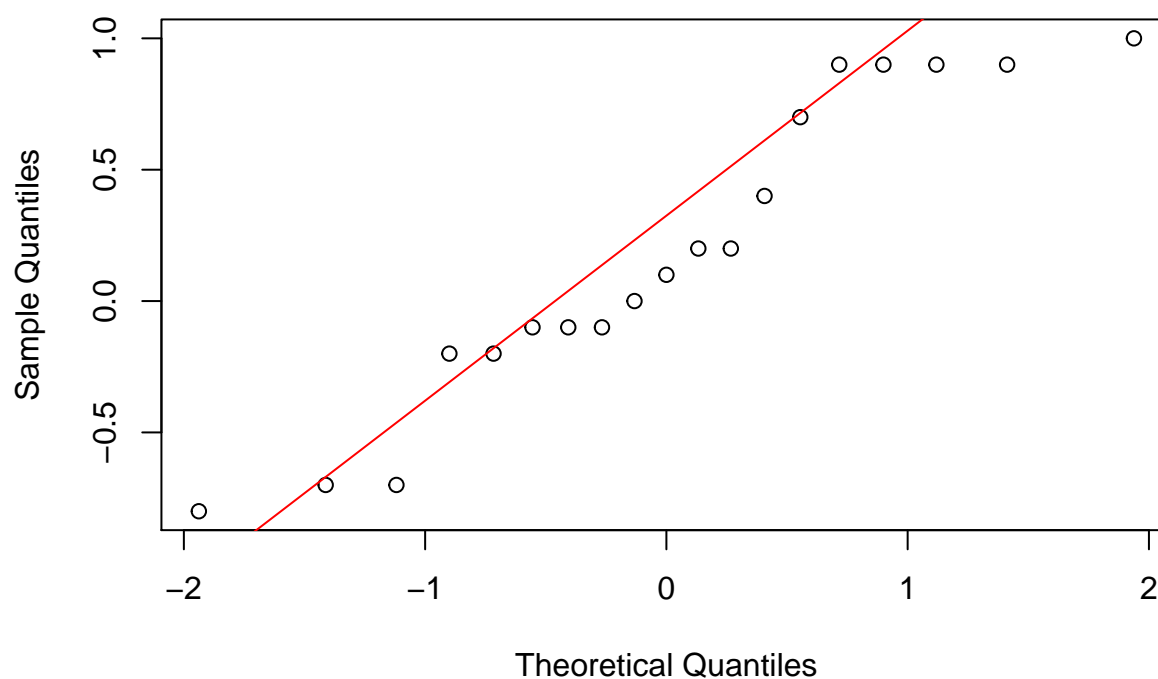
## Normal Q–Q Plot



```
qqnorm(dm3);qqline(dm3, col = 2)
```

# Normal Q–Q Plot



```
### checking Unit Root Test
adf.test(ddebt, alternative= "stationary", k=0)
```

```
##
##   Augmented Dickey-Fuller Test
##
## data:  ddebt
## Dickey-Fuller = -3.5911, Lag order = 0, p-value = 0.05124
## alternative hypothesis: stationary
```

```
adf.test(dinf,alternative="stationary", k=0)
```

```
##
##   Augmented Dickey-Fuller Test
##
## data:  dinf
## Dickey-Fuller = -4.1009, Lag order = 0, p-value = 0.01974
## alternative hypothesis: stationary
```

```
adf.test(demp, alternative="stationary", k=0)
```

```
##
##   Augmented Dickey-Fuller Test
##
```

```
## data:  demp
## Dickey-Fuller = -4.1738, Lag order = 0, p-value = 0.01719
## alternative hypothesis: stationary
```

```
adf.test(reg1$m3,alternative="stationary", k=2)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  reg1$m3
## Dickey-Fuller = -4.2766, Lag order = 2, p-value = 0.01361
## alternative hypothesis: stationary
```

```
#### ploting rthe correlation plot

corjoin<- cbind(ddebt,dinf,demp,dm3)
corsample1<- cor(corjoin)

cor1<-ggcorrplot(corsample1, hc.order = TRUE, type = "lower",
                 lab = TRUE,sig.level = 05, lab_size = 3)+
  ggtitle("Correlation Results, 2011-2015 ")

cor1
```


Correlation Results, 2011–2015
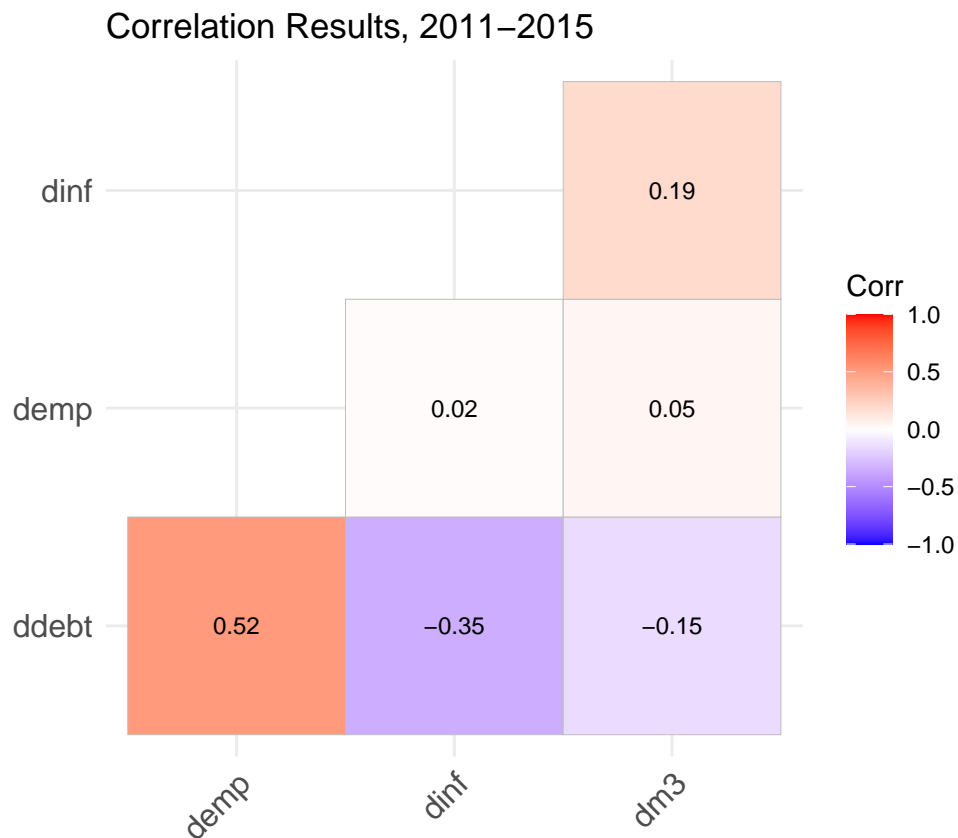
```
## using simple OLs linear regession model without log or differences
model1<- lm(ddebt~dinf+demp+dm3)
summary(model1)



##
## Call:
## lm(formula = ddebt ~ dinf + demp + dm3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0260 -0.4752 -0.1022  0.6431  0.8675
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1029     0.1879   0.548   0.5921
## dinf         -1.0119     0.5967  -1.696   0.1106
## demp          1.9416     0.7130   2.723   0.0157 *
## dm3          -0.1598     0.2919  -0.548   0.5920
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7089 on 15 degrees of freedom
## Multiple R-squared:  0.4172, Adjusted R-squared:  0.3006
## F-statistic: 3.579 on 3 and 15 DF,  p-value: 0.03934


###stationary check
ddebt2<-diff(reg2$debt)
dinf2<-diff(reg2$inf)
demp2<-diff(reg2$emp)
dm32<-diff(reg2$m3)

#### ploting rthe correlation plot
corjoin2<- cbind(ddebt2,dinf2,demp2,dm32)
corsample2<- cor(corjoin2)

cor2<-ggcorrplot(corsample2, hc.order = TRUE, type = "lower",
                 lab = TRUE,sig.level = 05, lab_size = 3 )+
  ggtitle("Correlation Results, 2015-2020 ")

cor2
```
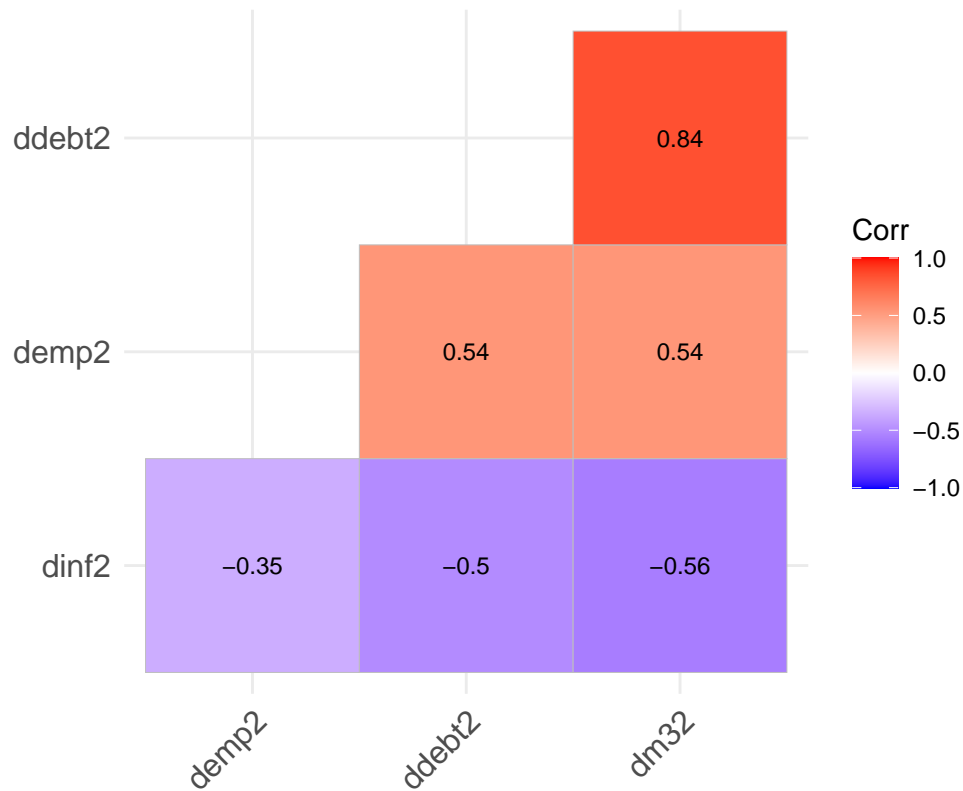
## Correlation Results, 2015–2020



```
## using simple OLs linear regession model without log or differences
model2<- lm(ddebt2~dinf2+demp2+dm32)
summary(model2)
```

```
##
## Call:
## lm(formula = ddebt2 ~ dinf2 + demp2 + dm32)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.87018 -0.95210 -0.04399  0.80656  2.59278
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3158     0.4056  -0.779  0.44822
## dinf2        -0.2061     0.9146  -0.225  0.82474
## demp2         0.8942     1.2648   0.707  0.49040
## dm32          2.2405     0.5513   4.064  0.00102 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.358 on 15 degrees of freedom
## Multiple R-squared:  0.7148, Adjusted R-squared:  0.6578
## F-statistic: 12.53 on 3 and 15 DF,  p-value: 0.0002297
```

```
######## Regression plots graphs and values in regression tables

regresult<-dwplot(list(model1,model2),
       vline = geom_vline(xintercept = 0, colour = "grey60",
                             linetype = 2)) %>%
        relabel_predictors(c(dinf = "Inflation",
                      demp = "Unemployment",
                      dm3 = "M 3",
                      dm32="M 3",
                      dinf2= "Inflation",
                      demp2="Unemployment")) +
  theme_bw() + xlab("Coefficient Estimate") + ylab("") +
  geom_vline(xintercept = 0, colour = "grey60", linetype = 2) +
  ggtitle("OLS Regression Results, Model 1 and Model 2") +
  theme(plot.title = element_text(face="bold"),
        legend.position = c(0.007, 0.01),
        legend.justification = c(0, 0),
        legend.background = element_rect(colour="grey80"),
        legend.title = element_blank())

regresult
```



**OLS Regression Results, Model 1 and Model 2**
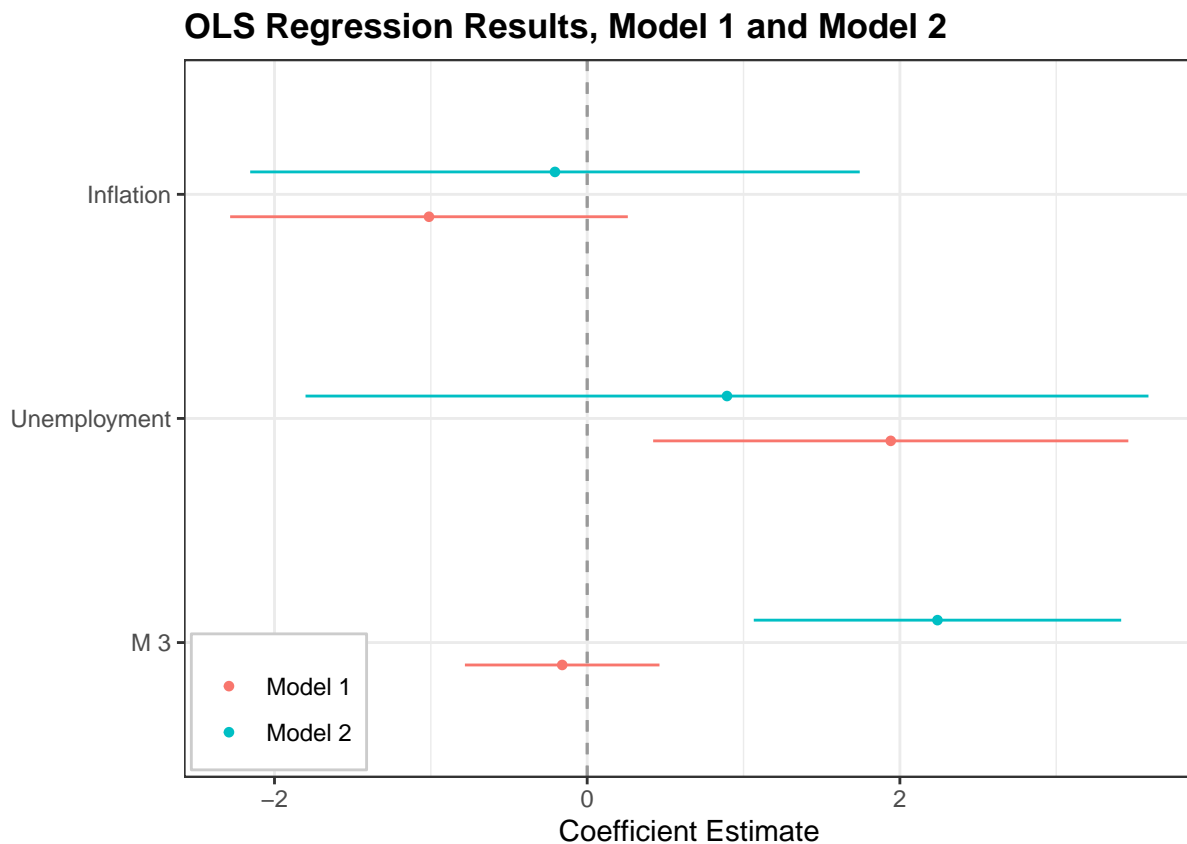
for converting the numeric file into word file with regression table and significance level

```
htmlreg(list(model1,model2), file= "olsresults",
        caption= "Regression Results",
```

```
        caption.above=TRUE, custom.model.names=c("OLS1","OLS2"),
        digits=2)
```

## The table was written to the file 'olsresults'.