ECON7350: Applied Econometrics for Macroeconomics and Finance

Tutorial 6: Cointegration - I

At the end of this tutorial you should be able to:

- Automate the task of unit root testing in multiple time-series samples in R;
- Implement the Engle-Granger cointegration test in R;
- Interpret the outcome of an Engle-Granger.
- Use the outcome of the Engle-Granger test to infer possible cointegrating relations.

Problems

In this tutorial you will test for cointegration using the Engle-Granger method. The data consists of four Australian interest rates: the 5 year (i3y) and 3 year (i3y) Treasury Bond (i.e., Capital Market) rates, along with the 180 day (i180d) and 90 (i90d) day Bank Accepted Bill (i.e., Money Market) rates. The data are annualized monthly rates for the period June 1992—August 2010 ($T = 219$), and are saved in `term_structure.csv`.

1. Analyse the integration properties of each individual process: $\{\text{i3y}_t\}$, $\{\text{i5y}_t\}$, $\{\text{i90d}_t\}$ and $\{\text{i180d}_t\}$. Based on the data, what inference can we draw about each of these processes resembling a unit root process?

Solution For this tutorial, we load the following useful packages.

```
library(forecast)
library(dplyr)
library(zoo)
library(aTSA)
```

It is also useful to create some functions to help automate the task of constructing adequate sets for ADF specifications. The following two functions estimate the coefficients and record AIC/BIC values for a range of ADF regressions specified by lags combined with the inclusion and/or excludion of a constant and/or trend.

One function performs the estimation in levels, while the other does the same in differences.

```
# create a function to estimate a range of ADF regression
# specifications in levels along with the AICs and BICs
ADF_estimate_lev <- function(y,  p_max = 9)
{
  TT <- length(y)
  ADF_est <- list()
  ic <- matrix(nrow = 3 * (1 + p_max), ncol = 5)
  colnames(ic) <- c("const", "trend", "p", "aic", "bic")
  i <- 0
  for (const in 0:1)
  {
    for (p in 0:p_max)
    {
      i <- i + 1
      ADF_est[[i]] <- Arima(diff(y), xreg = y[-TT],
                            order = c(p, 0, 0),
                            include.mean = as.logical(const),
                            include.drift = F)
      ic[i,] <- c(const, 0, p, ADF_est[[i]]$aic,
                  ADF_est[[i]]$bic)
    }

    if (const)
    {
      # only add a specification with trend if there is a
      # constant (i.e., exclude no constant with trend)
      for (p in 0:p_max)
      {
        i <- i + 1
        ADF_est[[i]] <- Arima(diff(y), xreg = y[-TT],
                              order = c(p, 0, 0),
                              include.mean = as.logical(const),
                              include.drift = T)
        ic[i,] <- c(const, 1, p, ADF_est[[i]]$aic,
                    ADF_est[[i]]$bic)
      }
    }
  }

  ic_aic <- ic[order(ic[,4]),][1:10,]
  ic_bic <- ic[order(ic[,5]),][1:10,]

  return(list(ADF_est = ADF_est, ic = ic,
```

```
              ic_aic = ic_aic, ic_bic = ic_bic))
}

# create a function to estimate a range of ADF regression
# specifications in differences along with the AICs and BICs
ADF_estimate_diff <- function(y, p_max = 9)
{
  TT <- length(diff(y))
  ADF_est_diff <- list()
  ic_diff <- matrix(nrow = 3 * (1 + p_max), ncol = 5)
  colnames(ic_diff) <- c("const", "trend", "p", "aic", "bic")
  i <- 0
  for (const in 0:1)
  {
    for (p in 0:p_max)
    {
      i <- i + 1
      ADF_est_diff[[i]] <- Arima(diff(diff(y)),
                                 xreg = diff(y)[-TT],
                                 order = c(p, 0, 0),
                                 include.mean = as.logical(const),
                                 include.drift = F)
      ic_diff[i,] <- c(const, 0, p, ADF_est_diff[[i]]$aic,
                       ADF_est_diff[[i]]$bic)
    }

    if (const)
    {
      # only add a specification with trend if there is a
      # constant (i.e., exclude no constant with trend)
      for (p in 0:p_max)
      {
        i <- i + 1
        ADF_est_diff[[i]] <- Arima(diff(diff(y)),
                                   xreg = diff(y)[-TT],
                                   order = c(p, 0, 0),
                                   include.mean = as.logical(const),
                                   include.drift = T)
        ic_diff[i,] <- c(const, 1, p, ADF_est_diff[[i]]$aic,
                         ADF_est_diff[[i]]$bic)
      }
    }
  }

  ic_aic_diff <- ic_diff[order(ic_diff[,4]),][1:10,]
```

```
  ic_bic_diff <- ic_diff[order(ic_diff[,5]),][1:10,]

  return(list(ADF_est_diff = ADF_est_diff,
              ic_diff = ic_diff,
              ic_aic_diff = ic_aic_diff,
              ic_bic_diff = ic_bic_diff))
}
```

Next, load the data and and extract the four variables.

```
mydata <- read.delim("term_structure.csv", header = TRUE,
                                           sep = ",")

dates <- as.yearqtr(mydata$obs)
i3y <- mydata$I3Y
i5y <- mydata$I5Y
i90d <- mydata$I90D
i180d <- mydata$I180D
```


Now, consider the proximity of $\{i3y_t\}$ to a unit root process. We begin by constructing an adequate set of ADF regressions in the level of $\{i3y_t\}$.

```
i3y_ADF_lev <- ADF_estimate_lev(i3y, p_max = 15)
print(i3y_ADF_lev$ic_aic)
```



```
##       const trend  p      aic      bic
##  [1,]     1     1 10 100.3036 147.6865
##  [2,]     1     1 12 101.0781 155.2301
##  [3,]     1     1 11 101.7485 152.5159
##  [4,]     1     0 12 102.8333 153.6008
##  [5,]     1     1 13 102.9466 160.4830
##  [6,]     1     0 10 103.1882 147.1866
##  [7,]     0     0 12 103.5612 150.9442
##  [8,]     1     1 14 103.5876 164.5085
##  [9,]     1     1  7 103.6625 140.8919
## [10,]     1     0 11 104.2917 151.6746
```

```
print(i3y_ADF_lev$ic_bic)
```

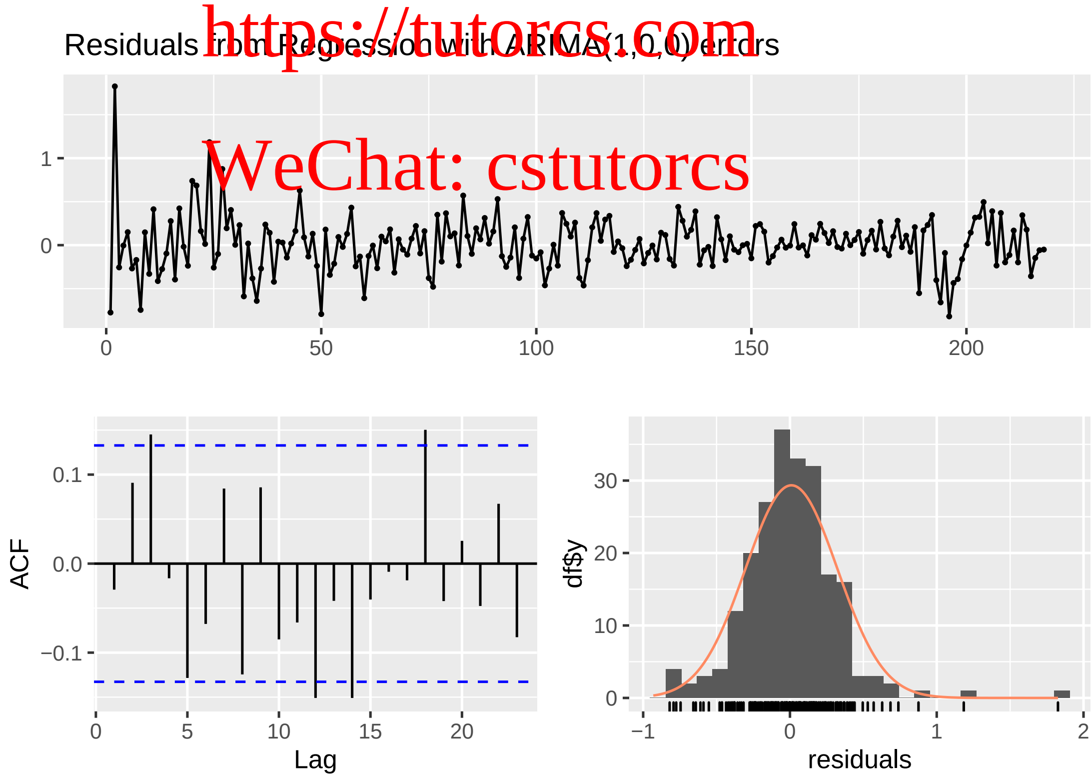
```
##      const trend p      aic      bic
##  [1,]     0     0 1 118.3726 128.5261
##  [2,]     1     0 1 116.0791 129.6171
##  [3,]     1     0 2 113.7241 130.6466
##  [4,]     0     0 2 117.3288 130.8668
##  [5,]     1     0 3 110.8348 131.1417
##  [6,]     1     1 3 107.5159 131.2074
##  [7,]     1     1 2 111.4567 131.7637
##  [8,]     1     1 1 115.0552 131.9776
```

```
##   [9,]      0      0 0 125.2605 132.0295
## [10,]      0      0 3 116.1076 133.0301
```

The AIC and BIC ranking do not have any specifications in common, so we select from both top 10 rankings in a way that reflects some agreement. This is obviously very subjective! The justification we use as follows. From the AIC list, take the most preferred specification along with a few others that have the lowest BIC values. Then, do the same using the BIC list.

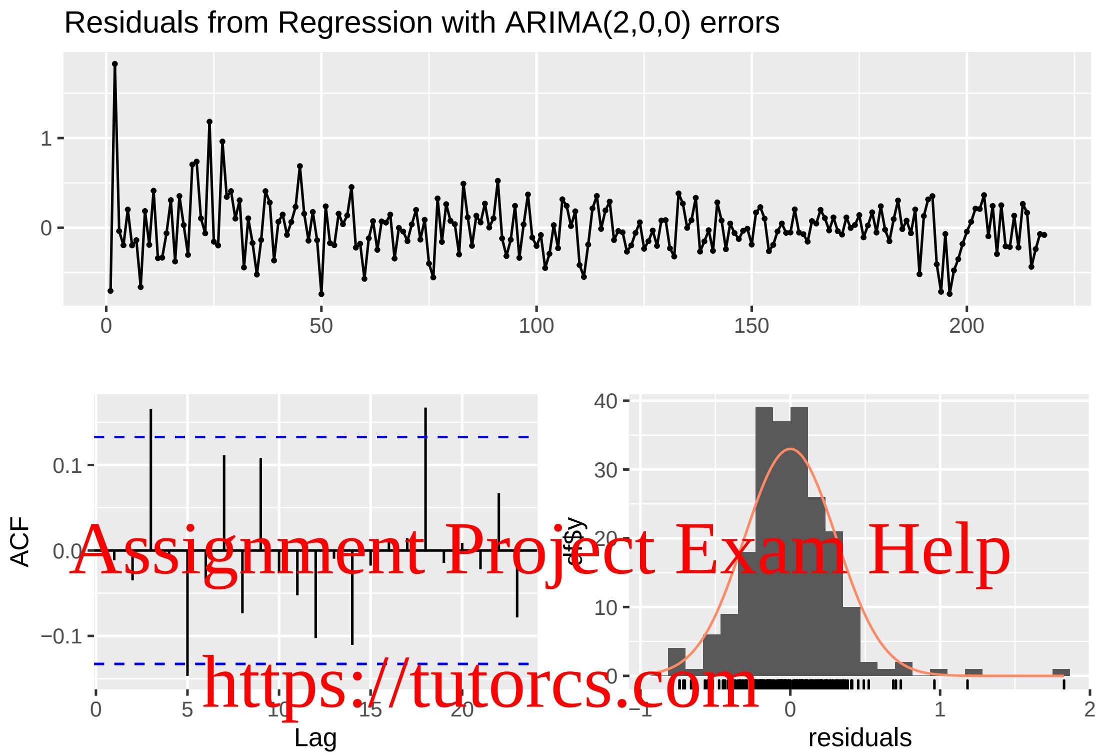
As result, we obtain the following set of specifications on which we run our residuals analysis.

```r
i3y_adq_set <- as.matrix(arrange(as.data.frame(
                    rbind(i3y_ADF_lev$ic_aic[c(1, 6, 9),],
                          i3y_ADF_lev$ic_bic[c(1, 3, 5:7),])),
                          const, trend, p))
i3y_adq_idx <- match(data.frame(t(i3y_adq_set[, 1:3])),
                     data.frame(t(i3y_ADF_lev$ic[, 1:3])))

for (i in 1:length(i3y_adq_idx))
{
  checkresiduals(i3y_ADF_lev$ADF_est[[i3y_adq_idx[i]]])
}
```

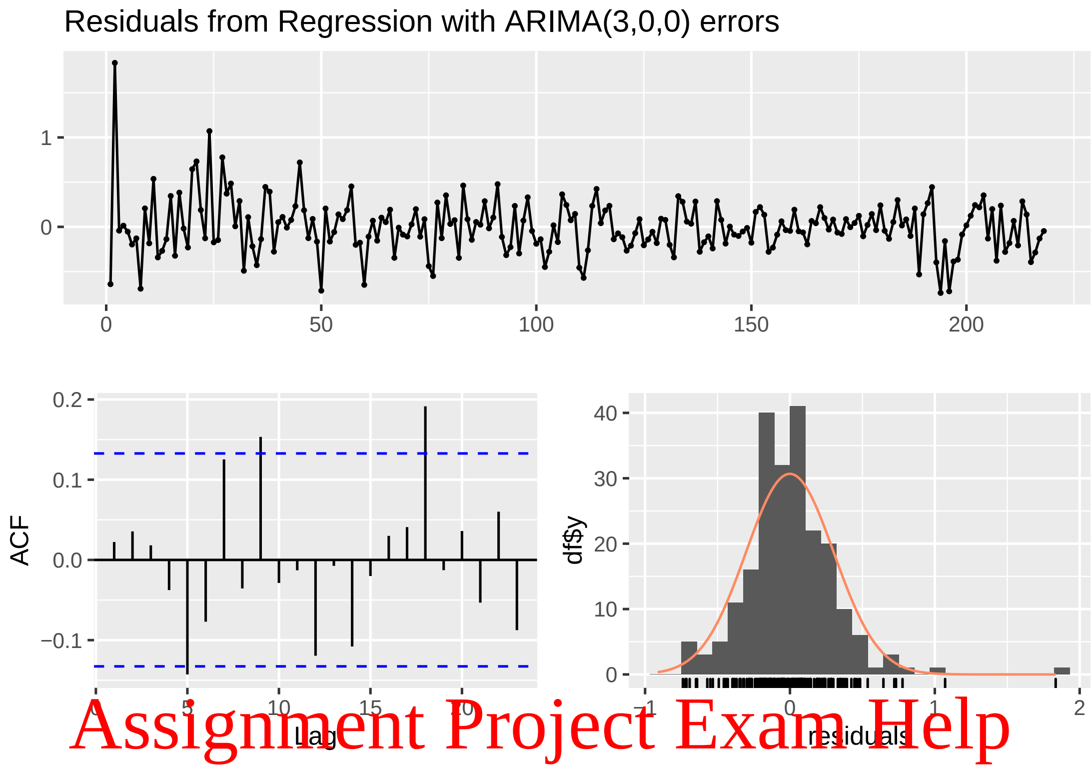




```
##
##  Ljung-Box test
##
```

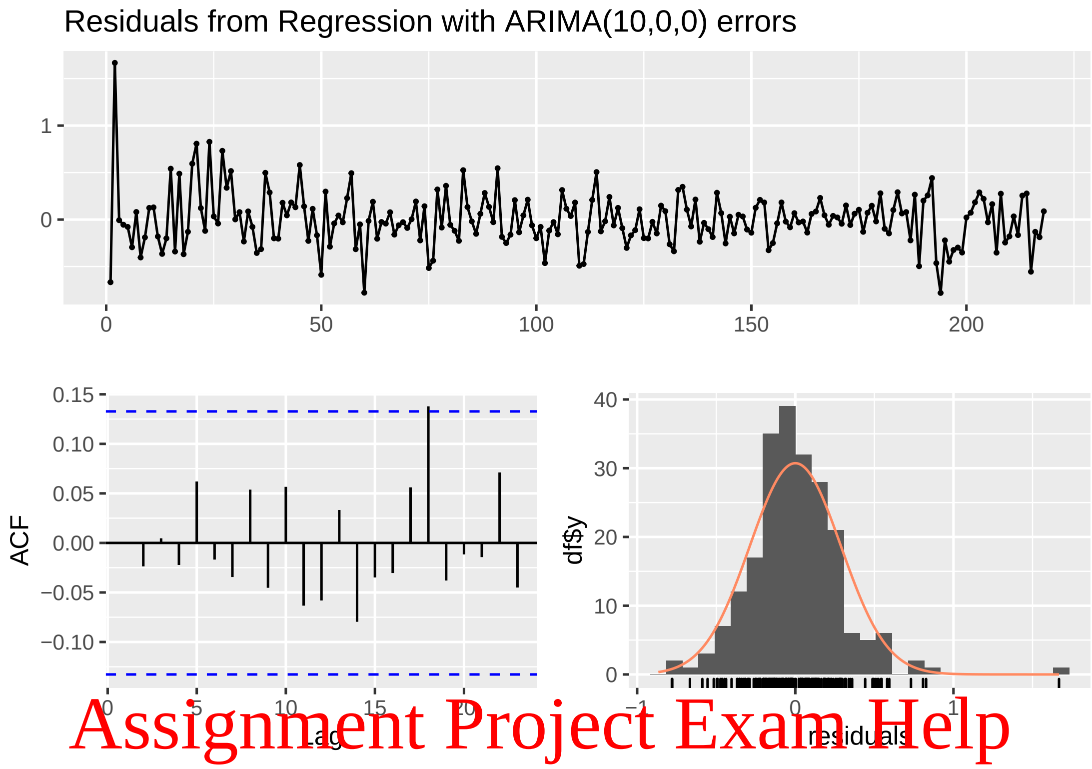
```
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 20.038, df = 8, p-value = 0.01019
##
## Model df: 2.   Total lags used: 10
```



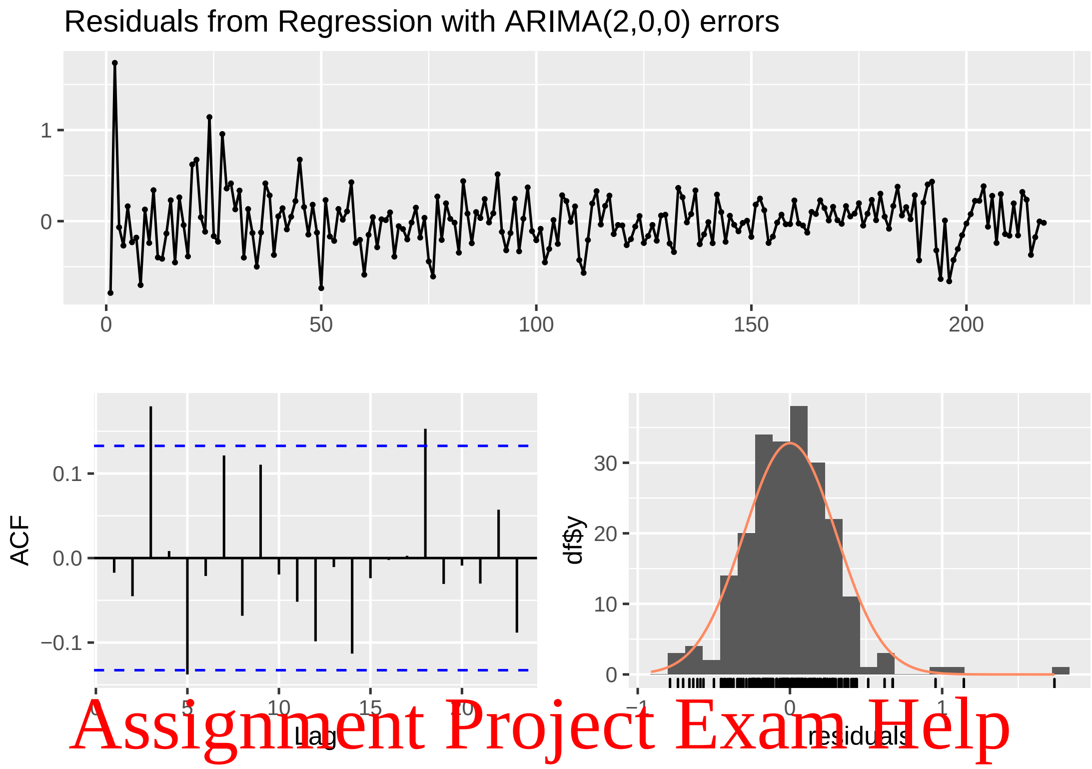

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 18.425, df = 6, p-value = 0.005253
##
## Model df: 4.   Total lags used: 10
```



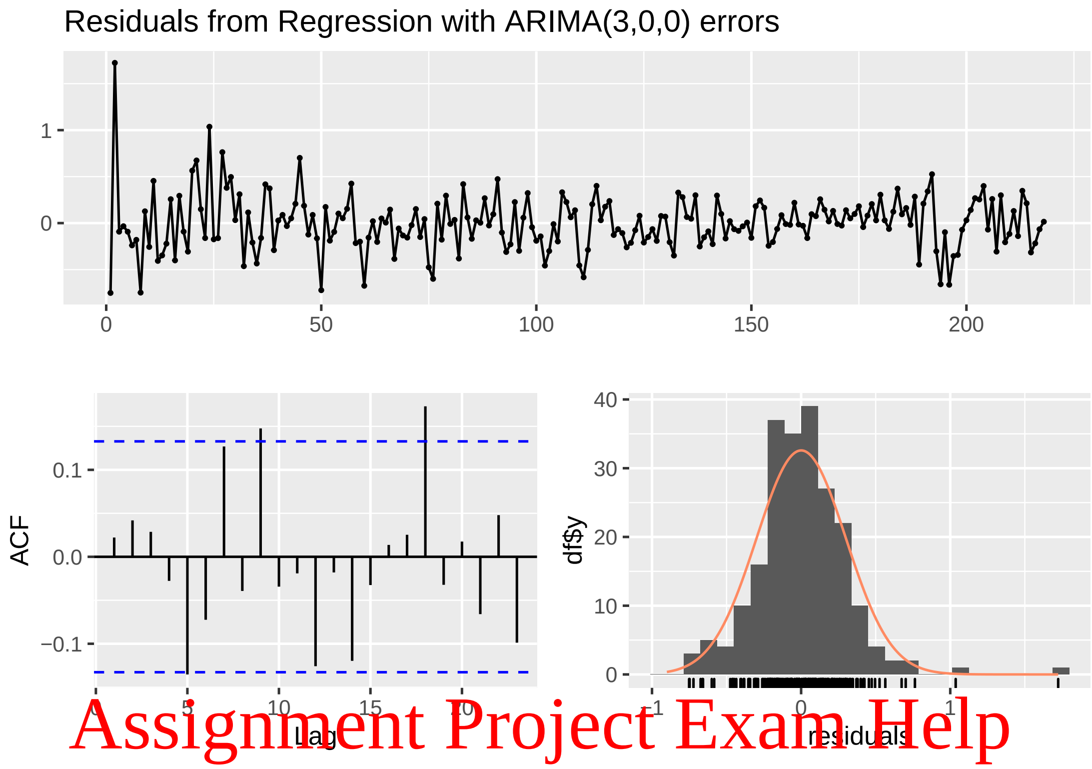

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 16.161, df = 5, p-value = 0.0064
##
## Model df: 5.   Total lags used: 10
```



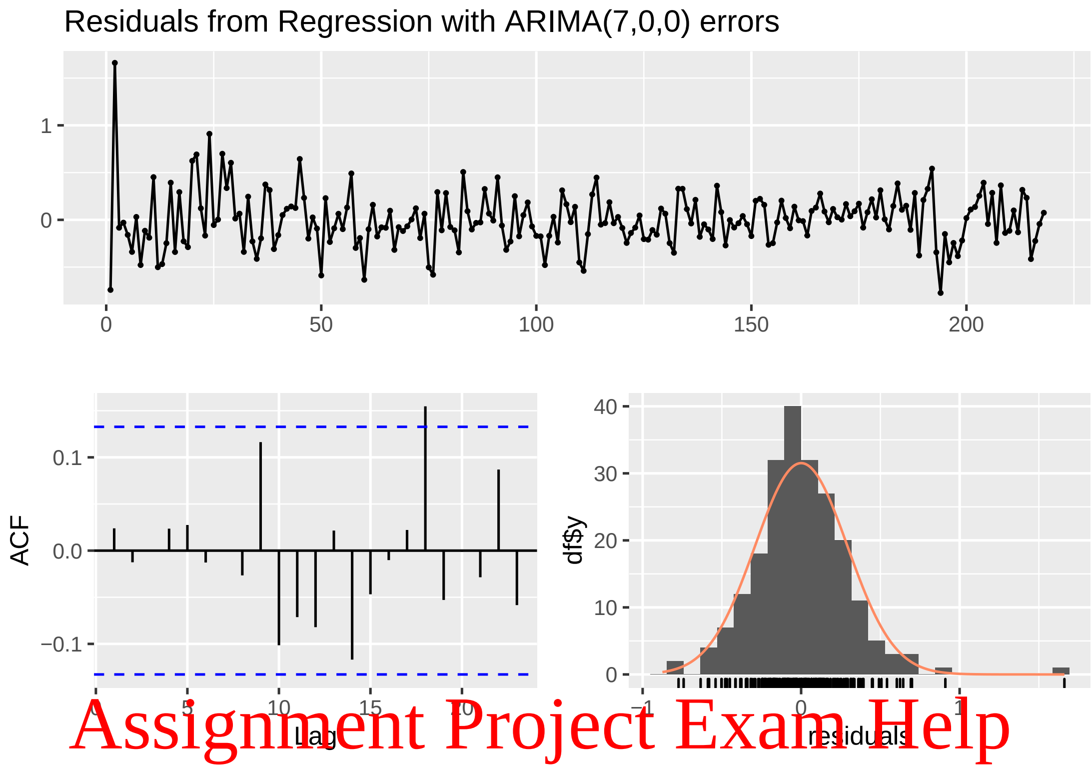

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(10,0,0) errors
## Q* = 7.0612, df = 3, p-value = 0.06997
##
## Model df: 12.   Total lags used: 15
```



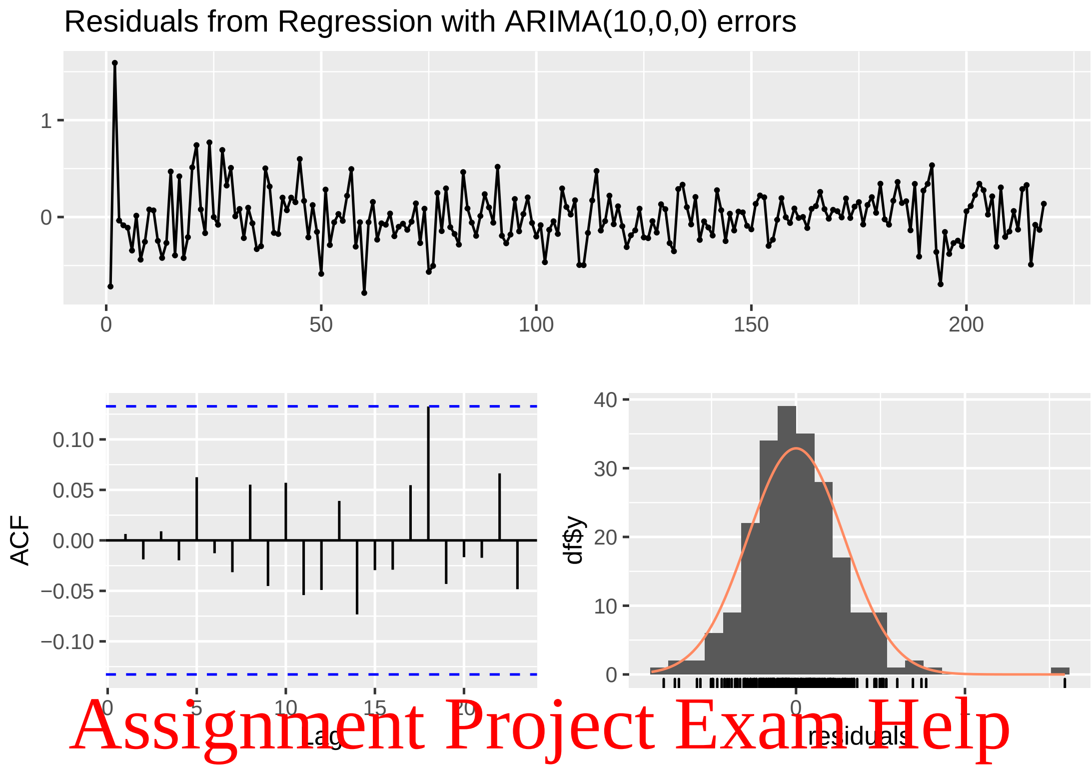

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 19.404, df = 5, p-value = 0.001616
##
## Model df: 5.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 15.443, df = 4, p-value = 0.003866
##
## Model df: 6.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(7,0,0) errors
## Q* = 8.9851, df = 3, p-value = 0.02949
##
## Model df: 10.   Total lags used: 13
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(10,0,0) errors
## Q* = 6.5159, df = 3, p-value = 0.08904
##
## Model df: 13.   Total lags used: 16
```

We reject white noise residuals at the 5% significance level for all models with $p < 10$. Hence, we remove all models except the two with $p = 10$, both containing a constant and one also containing a trend.

Given our adequate set of ADF regressions, we should run the ADF test with `nlag = 11`, but we will use `nlag = 15` just to check how sensitive the results are to including more lags (which the AIC prefers, but the BIC rejects).

```
adf.test(i3y, nlag = 15)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##       lag    ADF p.value
##  [1,]   0 -0.896   0.358
##  [2,]   1 -0.697   0.429
##  [3,]   2 -1.220   0.242
##  [4,]   3 -1.161   0.264
```

```
##  [5,]   4 -1.103   0.284
##  [6,]   5 -1.022   0.313
##  [7,]   6 -0.915   0.352
##  [8,]   7 -0.950   0.339
##  [9,]   8 -0.761   0.406
## [10,]   9 -0.773   0.402
## [11,]  10 -0.738   0.415
## [12,]  11 -0.807   0.390
## [13,]  12 -0.709   0.425
## [14,]  13 -0.603   0.463
## [15,]  14 -0.645   0.448
## Type 2: with drift no trend
##       lag   ADF p.value
##  [1,]   0 -1.74  0.4298
##  [2,]   1 -2.05  0.3087
##  [3,]   2 -2.80  0.0634
##  [4,]   3 -3.10  0.0295
##  [5,]   4 -2.85  0.0553
##  [6,]   5 -2.38  0.1779
##  [7,]   6 -2.22  0.2424
##  [8,]   7 -2.65  0.0893
##  [9,]   8 -2.32  0.2036
## [10,]   9 -2.54  0.1154
## [11,]  10 -2.09  0.2924
## [12,]  11 -1.97  0.3400
## [13,]  12 -1.60  0.4828
## [14,]  13 -1.55  0.5018
## [15,]  14 -1.37  0.5683
## Type 3: with drift and trend
##       lag   ADF p.value
##  [1,]   0 -2.27  0.4618
##  [2,]   1 -2.90  0.1999
##  [3,]   2 -3.29  0.0728
##  [4,]   3 -3.79  0.0203
##  [5,]   4 -3.51  0.0422
##  [6,]   5 -2.97  0.1689
##  [7,]   6 -2.90  0.1985
##  [8,]   7 -3.45  0.0475
##  [9,]   8 -3.28  0.0758
## [10,]   9 -3.63  0.0310
## [11,]  10 -3.05  0.1333
## [12,]  11 -2.83  0.2293
## [13,]  12 -2.52  0.3550
## [14,]  13 -2.65  0.3046
## [15,]  14 -2.39  0.4093
```

```
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

For specifications with a constant and a trend along with $p \geq 9$ the null cannot be rejected at the 5% significance level. The same conclusion holds for specifications with a constant, no trend and $p \geq 10$. Overall evidence suggests $\{\text{i3y}_t\}$ is not empirically distinguishable from a unit root process.

Accordingly, we repeat the exercise for the differenced process $\{\Delta\text{i3y}_t\}$.

```r
i3y_ADF_diff <- ADF_estimate_diff(i3y, p_max = 15)
print(i3y_ADF_diff$ic_aic_diff)
```

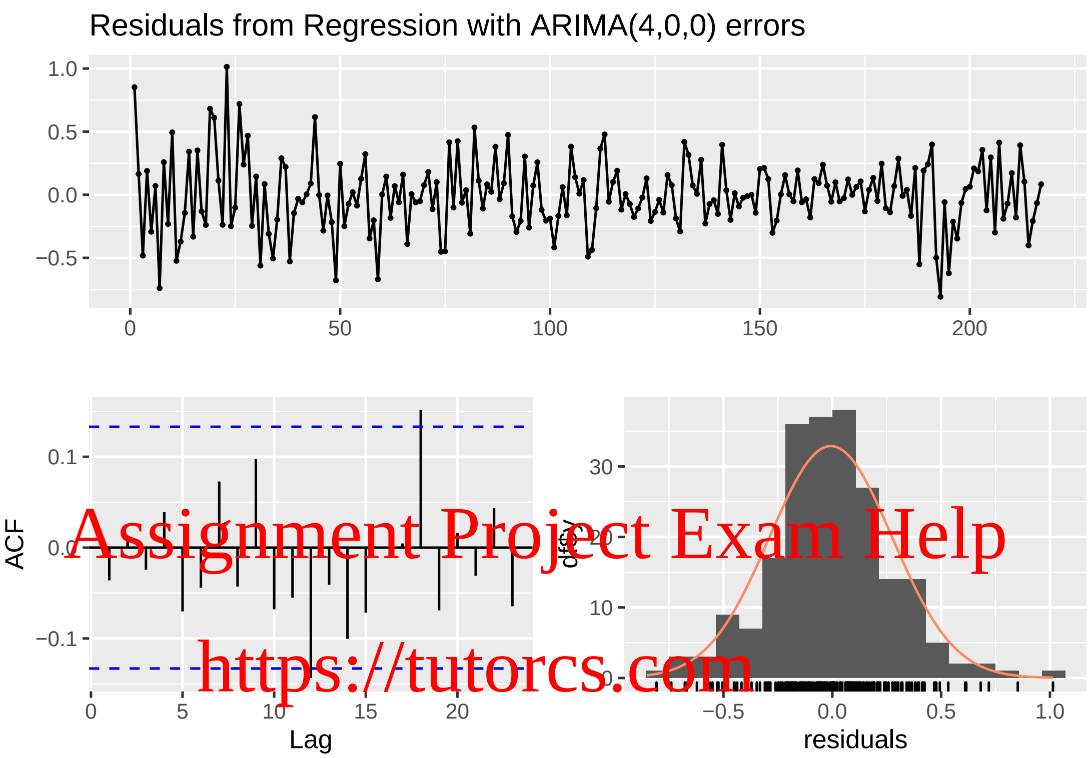
```
##       const trend  p      aic       bic
##  [1,]     0     0 12 77.49903 124.81760
##  [2,]     0     0  4 78.52513  98.80452
##  [3,]     0     0 10 78.65040 119.20917
##  [4,]     0     0 11 78.96325 122.90192
##  [5,]     0     0  5 79.34904 103.00832
##  [6,]     1     0 12 79.35876 130.05722
##  [7,]     0     0 13 79.47963 130.17809
##  [8,]     1     0  4 80.46394 104.12322
##  [9,]     1     0 10 80.55557 124.49423
## [10,]     1     0 11 80.84817 128.16673
```

```r
print(i3y_ADF_diff$ic_bic_diff)
```

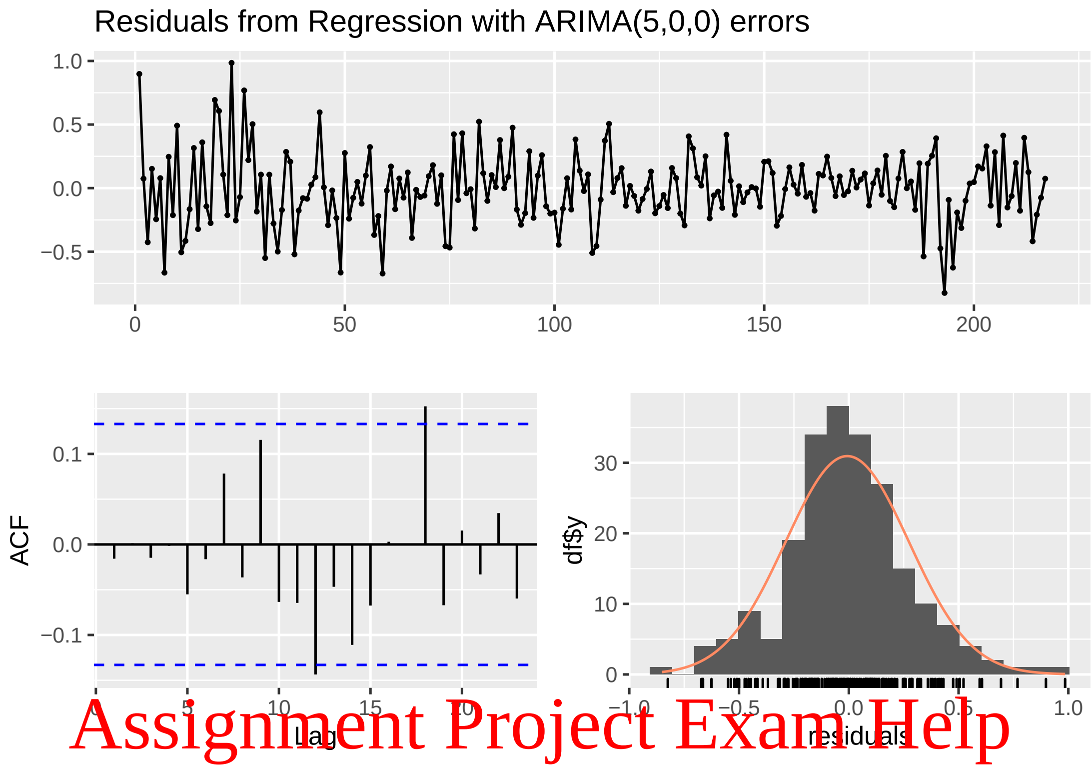
```
##       const trend p      aic       bic
##  [1,]     0     0 4 78.52513  98.80452
##  [2,]     0     0 1 89.38183  99.52153
##  [3,]     0     0 5 79.34904 103.00832
##  [4,]     0     0 2 90.39326 103.91285
##  [5,]     1     0 4 80.46394 104.12322
##  [6,]     1     0 1 91.37490 104.89449
##  [7,]     0     0 3 90.16004 107.05953
##  [8,]     1     0 5 81.28020 108.31938
##  [9,]     1     0 2 92.37827 109.27776
## [10,]     1     1 4 82.30717 109.34635
```

```r
i3y_adq_set_diff <- as.matrix(arrange(as.data.frame(
                        i3y_ADF_diff$ic_bic_diff[c(1, 3, 5),]),
                        const, trend, p))
i3y_adq_idx_diff <- match(data.frame(
                        t(i3y_adq_set_diff[, 1:3])),
                        data.frame(
                        t(i3y_ADF_diff$ic_diff[, 1:3])))

for (i in 1:length(i3y_adq_idx_diff))
```

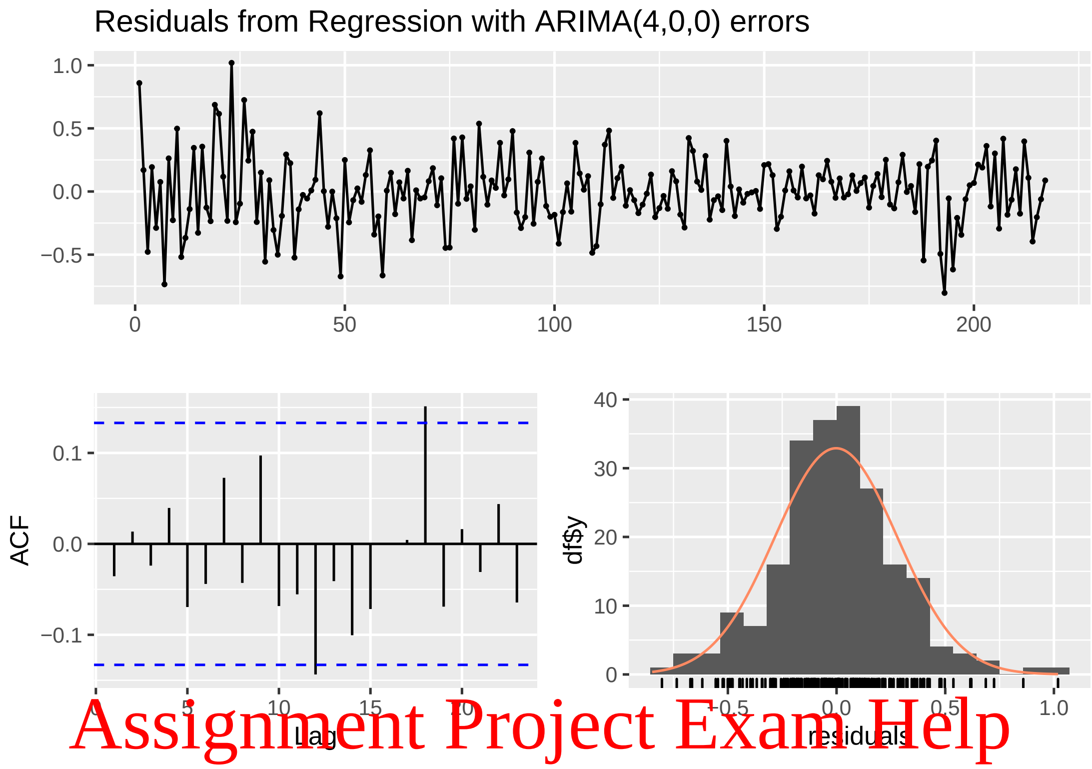
```
{
  checkresiduals(
    i3y_ADF_diff$ADF_est_diff[[i3y_adq_idx_diff[i]]])
}
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(4,0,0) errors
## Q* = 7.1851, df = 5, p-value = 0.2072
##
## Model df: 5.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(5,0,0) errors
## Q* = 6.5018, df = 4, p-value = 0.1647
##
## Model df: 6.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(4,0,0) errors
## Q* = 7.1557, df = 4, p-value = 0.1279
##
## Model df: 6.   Total lags used: 10
```



```
adf.test(diff(i3y))
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -12.23    0.01
## [2,]   1  -8.31    0.01
## [3,]   2  -6.28    0.01
## [4,]   3  -6.30    0.01
## [5,]   4  -6.85    0.01
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0 -12.21    0.01
## [2,]   1  -8.32    0.01
## [3,]   2  -6.29    0.01
## [4,]   3  -6.31    0.01
```

```
## [5,]   4  -6.86    0.01
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0 -12.19    0.01
## [2,]   1  -8.30    0.01
## [3,]   2  -6.27    0.01
## [4,]   3  -6.30    0.01
## [5,]   4  -6.84    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

The null is rejected for all specifications. We conclude $\{\Delta \mathrm{i3y}_t\}$ is empirically distinguishable from an I(1) process, which means $\{\mathrm{i3y}_t\}$ is *not* empirically distinguishable from an I(1) process.

Repeating for $\{\mathrm{i5y}_t\}$ and $\{\Delta \mathrm{i5y}_t\}$, we obtain the following.

```
i5y_ADF_lev <- ADF_estimate_lev(i5y, p_max = 15)
print(i5y_ADF_lev$ic_aic)
```

```
##       const trend  p      aic      bic
##  [1,]     1     1 10 58.66839 106.0513
##  [2,]     1     1  3 78.72176 102.4132
##  [3,]     1     1  5 79.63041 110.0909
##  [4,]     1     1  4 80.33103 107.4070
##  [5,]     1     1  7 80.44508 117.6745
##  [6,]     1     1 12 80.70523 134.8571
##  [7,]     1     1  9 81.16063 125.1591
##  [8,]     1     1 11 81.18581 131.9532
##  [9,]     1     1  2 81.54914 101.8561
## [10,]     1     1  6 81.61182 115.4568
```

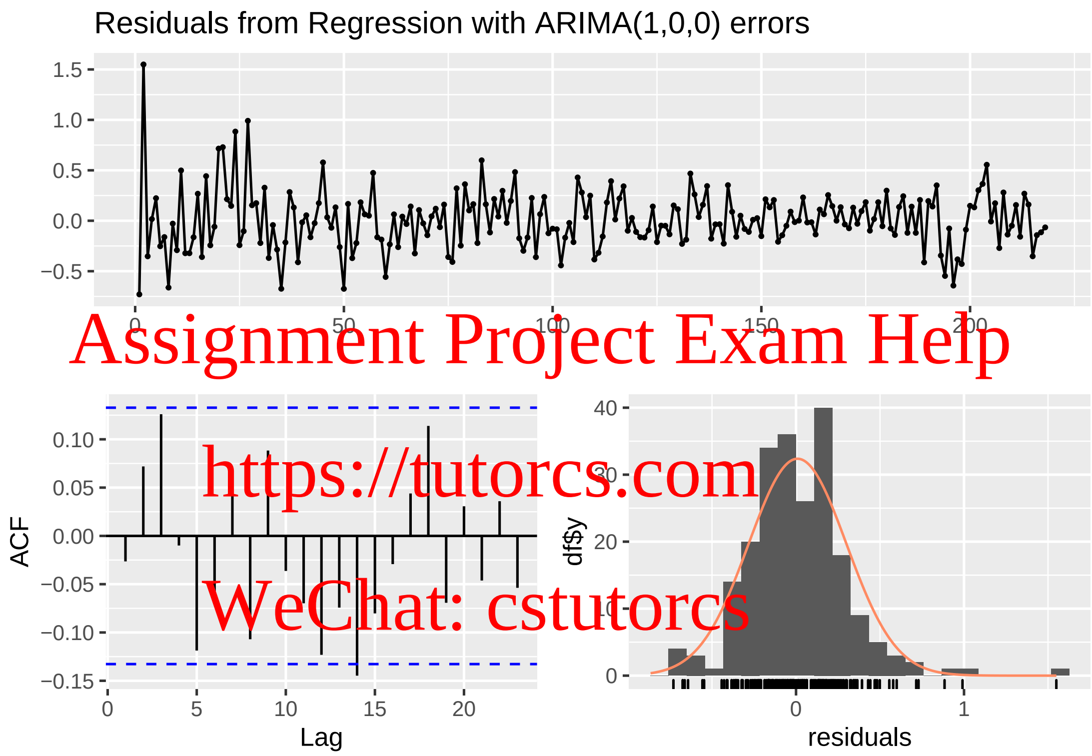


```
print(i5y_ADF_lev$ic_bic)
```

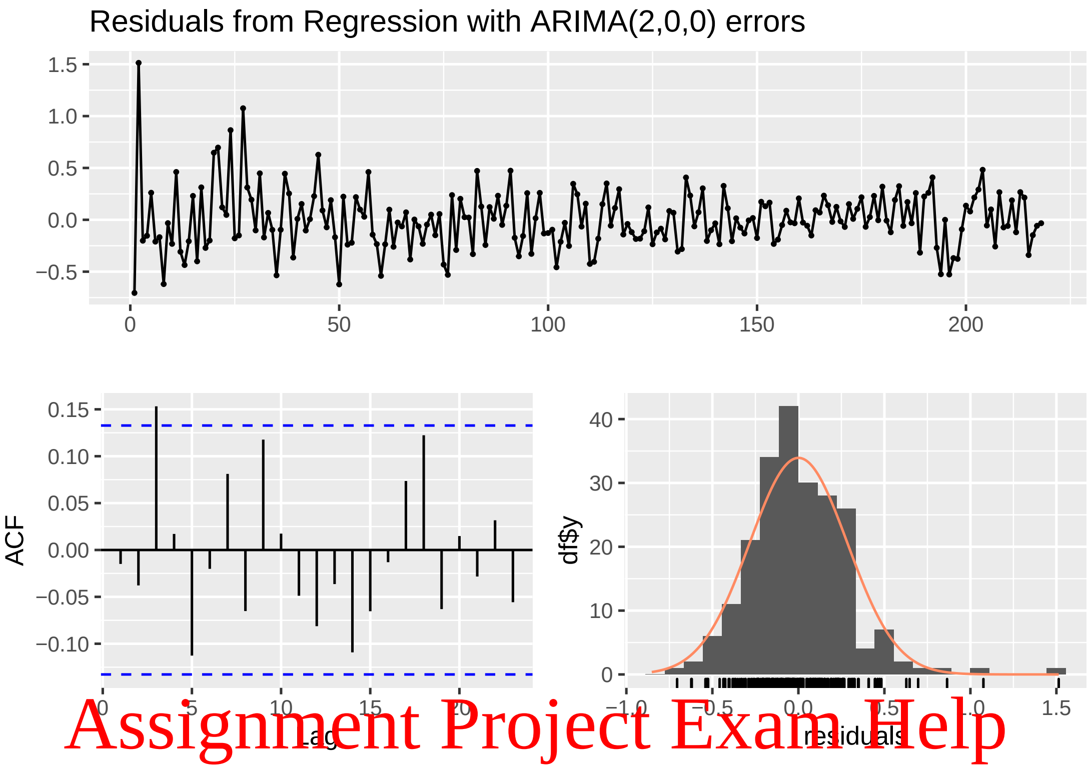
```
##       const trend p      aic       bic
##  [1,]     0     0 1 86.51082  96.66430
##  [2,]     1     0 1 84.62997  98.16795
##  [3,]     0     0 2 86.58971 100.12769
##  [4,]     1     1 1 83.38674 100.30922
##  [5,]     1     0 2 83.91952 100.84199
##  [6,]     0     0 0 94.42651 101.19550
##  [7,]     1     1 2 81.54914 101.85611
##  [8,]     1     1 3 78.72176 102.41322
##  [9,]     1     0 3 82.46661 102.77358
## [10,]     0     0 3 86.18707 103.10955
```

```
i5y_adq_set <- as.matrix(arrange(as.data.frame(
  rbind(i5y_ADF_lev$ic_aic[1,],
```

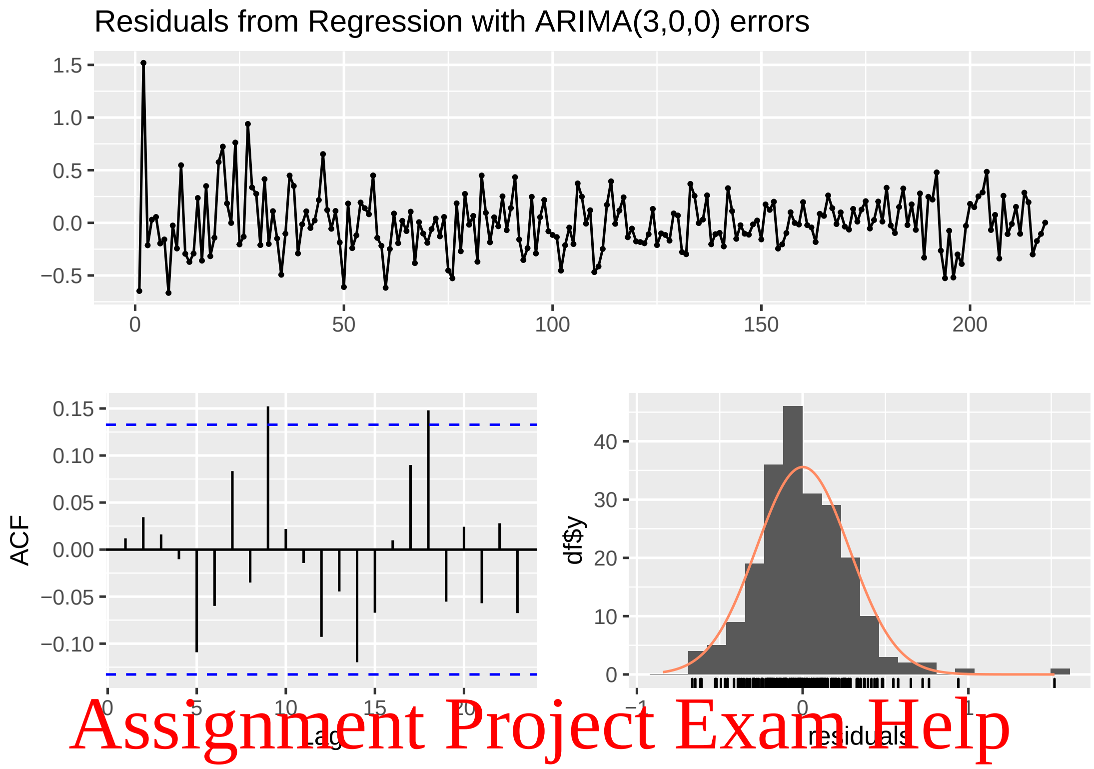
```
        i5y_ADF_lev$ic_bic[c(1, 7:8),])),
  const, trend, p))
i5y_adq_idx <- match(data.frame(t(i5y_adq_set[, 1:3])),
                     data.frame(t(i5y_ADF_lev$ic[, 1:3])))

for (i in 1:length(i5y_adq_idx))
{
  checkresiduals(i5y_ADF_lev$ADF_est[[i5y_adq_idx[i]]])
}
```



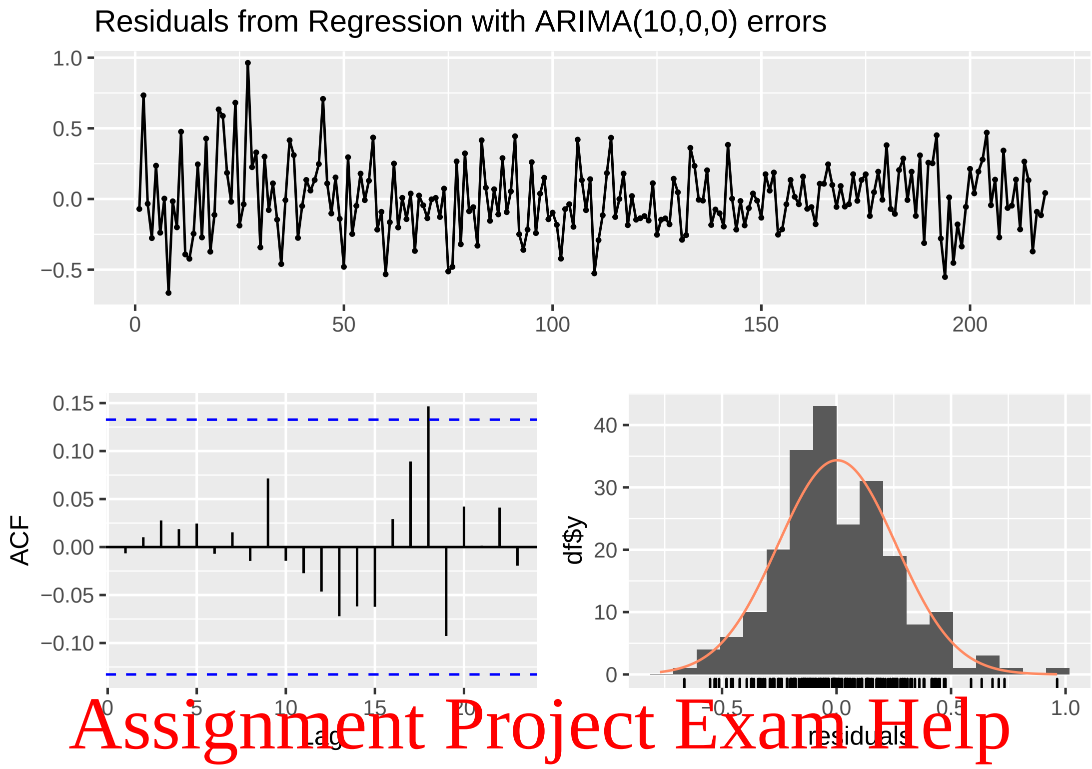

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 14.115, df = 8, p-value = 0.07883
##
## Model df: 2.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 14.329, df = 5, p-value = 0.01365
##
## Model df: 5.   Total lags used: 10
```

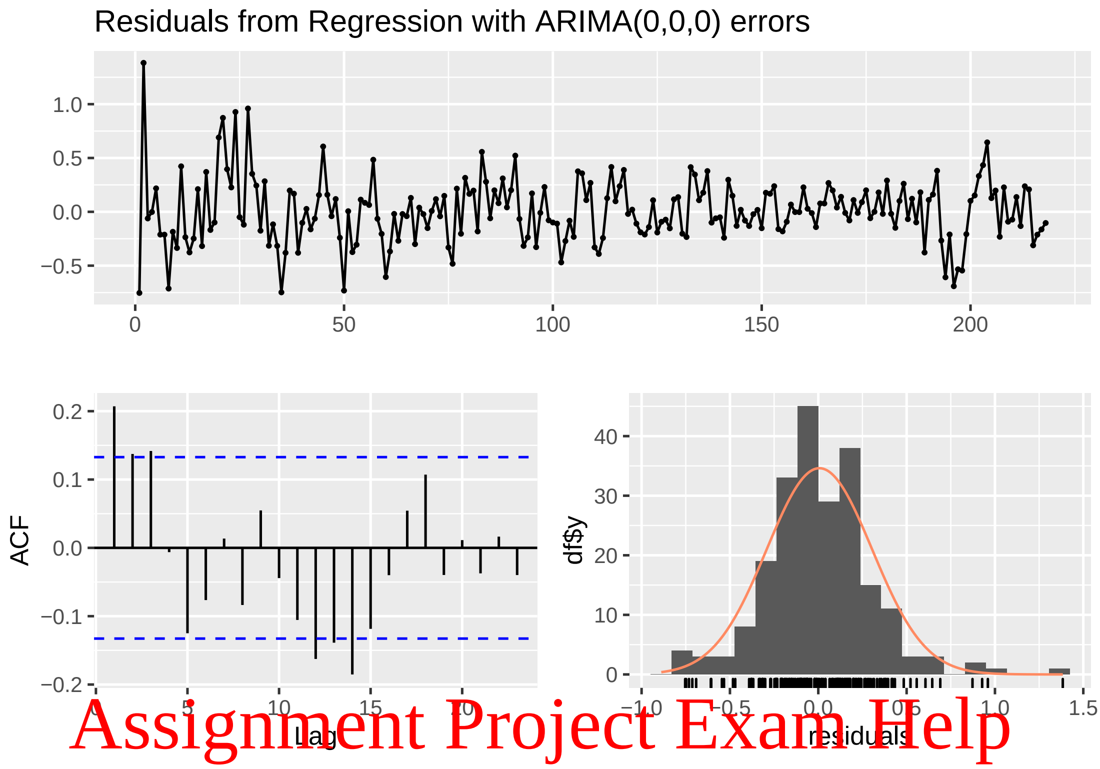



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 11.163, df = 4, p-value = 0.02479
##
## Model df: 6.   Total lags used: 10
```

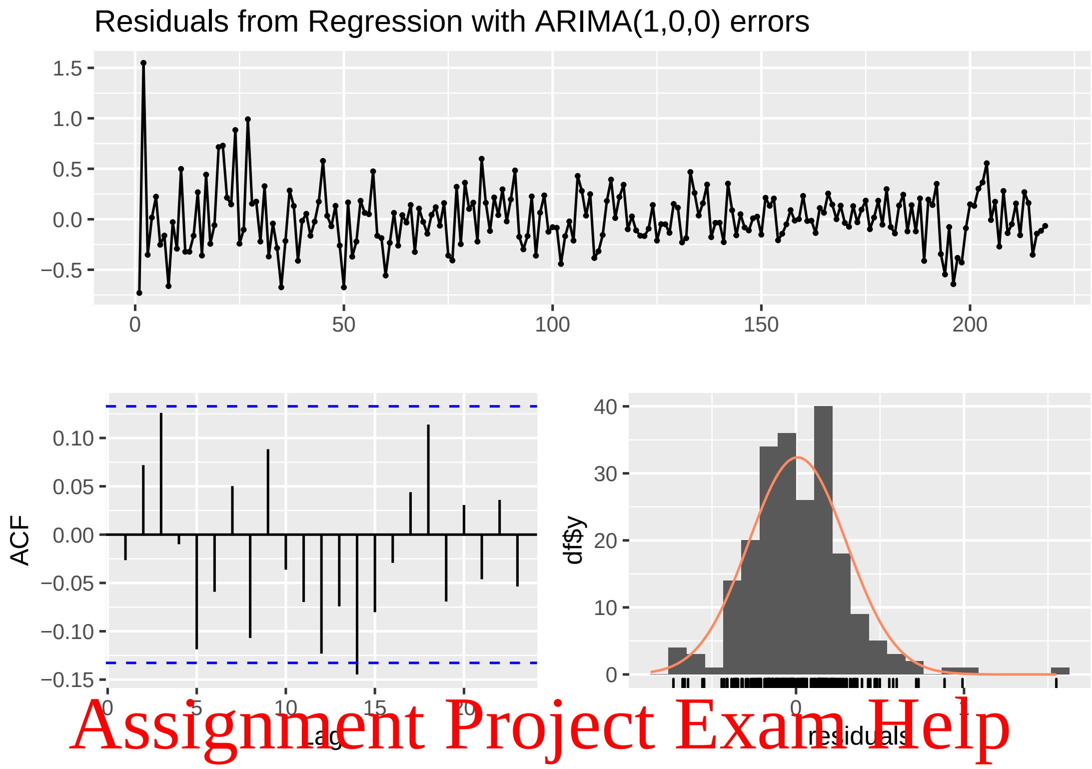



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(10,0,0) errors
## Q* = 5.6557, df = 3, p-value = 0.1296
##
## Model df: 13.   Total lags used: 16
```

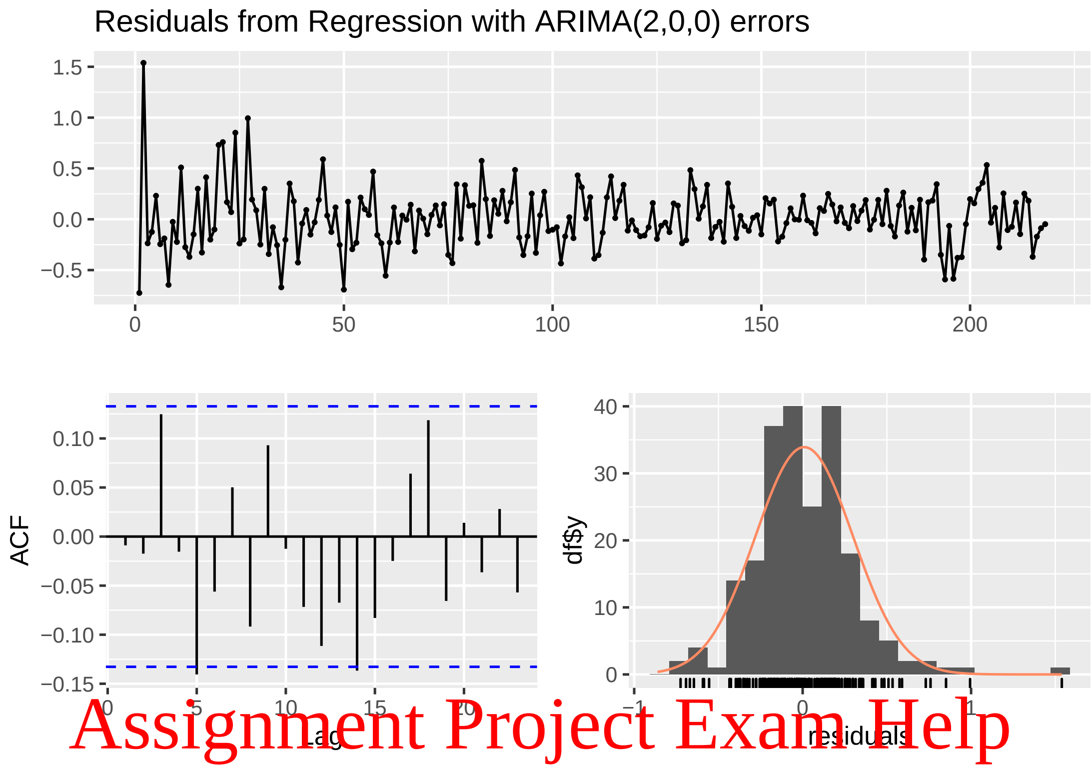
```
i5y_adq_set <- as.matrix(arrange(as.data.frame(
  rbind(i5y_ADF_lev$ic_aic[c(1, 6:8),],
        i5y_ADF_lev$ic_bic[c(1, 3, 6, 10),])),
  const, trend, p))
i5y_adq_idx <- match(data.frame(t(i5y_adq_set[, 1:3])),
                     data.frame(t(i5y_ADF_lev$ic[, 1:3])))

for (i in 1:length(i5y_adq_idx))
{
  checkresiduals(i5y_ADF_lev$ADF_est[[i5y_adq_idx[i]]])
}
```



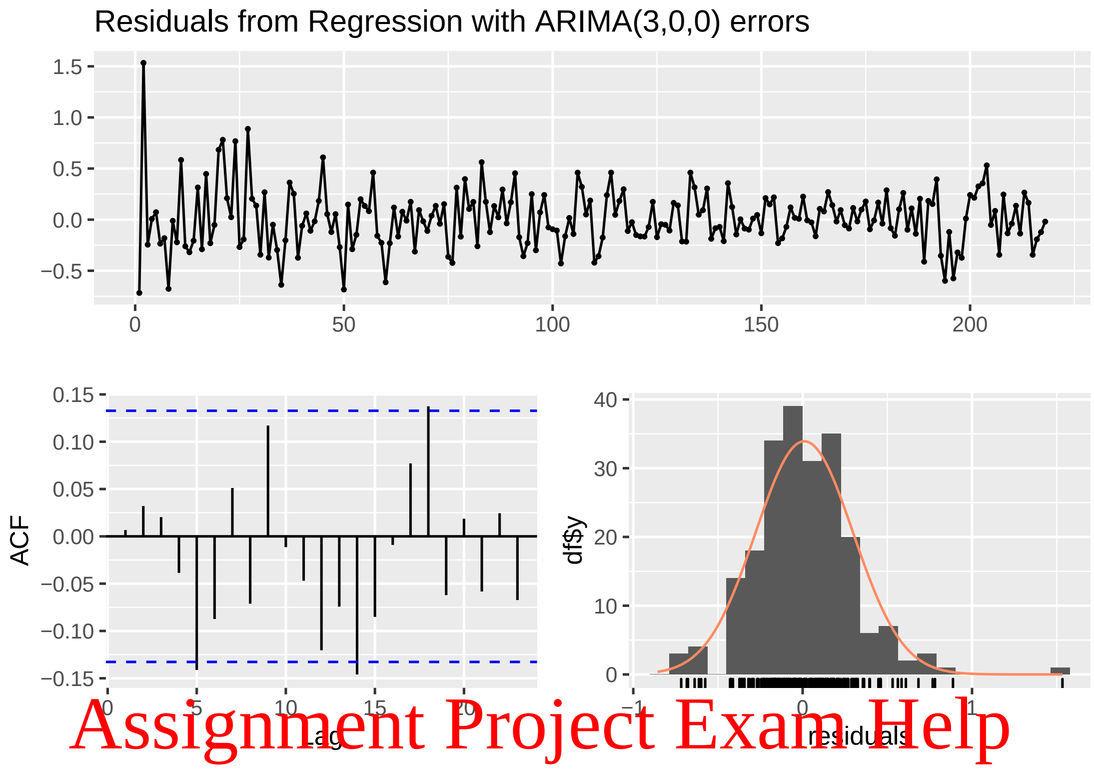

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 25.775, df = 9, p-value = 0.002223
##
## Model df: 1.   Total lags used: 10
```



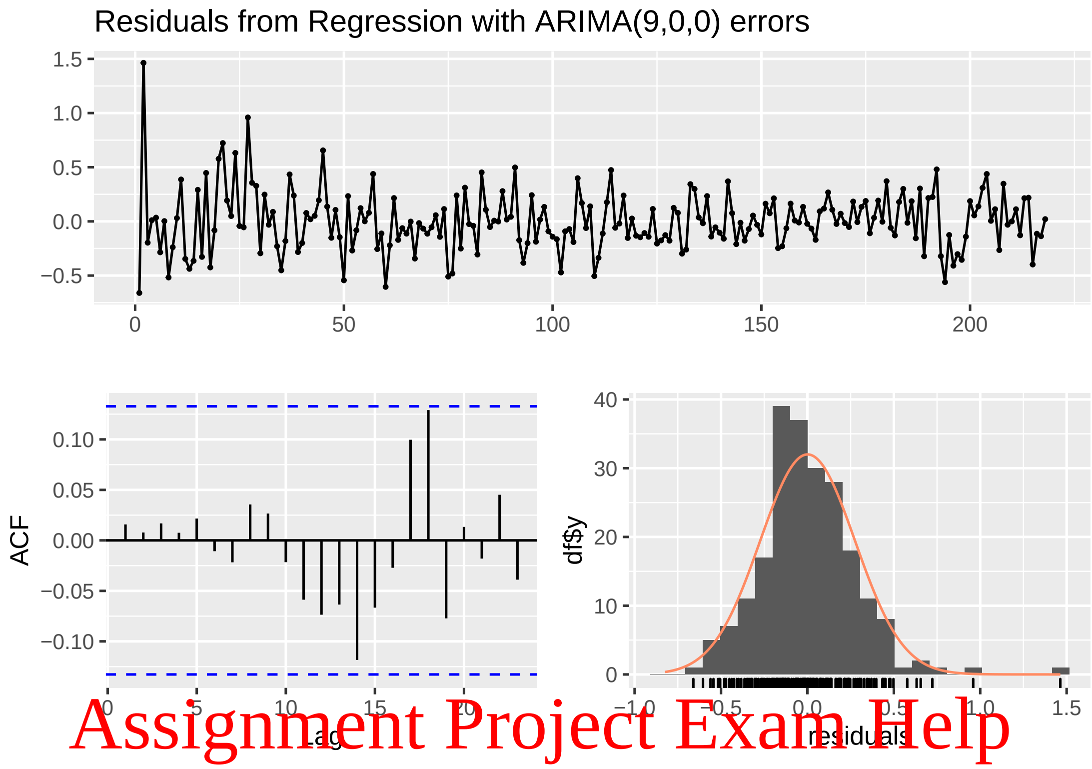

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 14.115, df = 8, p-value = 0.07883
##
## Model df: 2.   Total lags used: 10
```



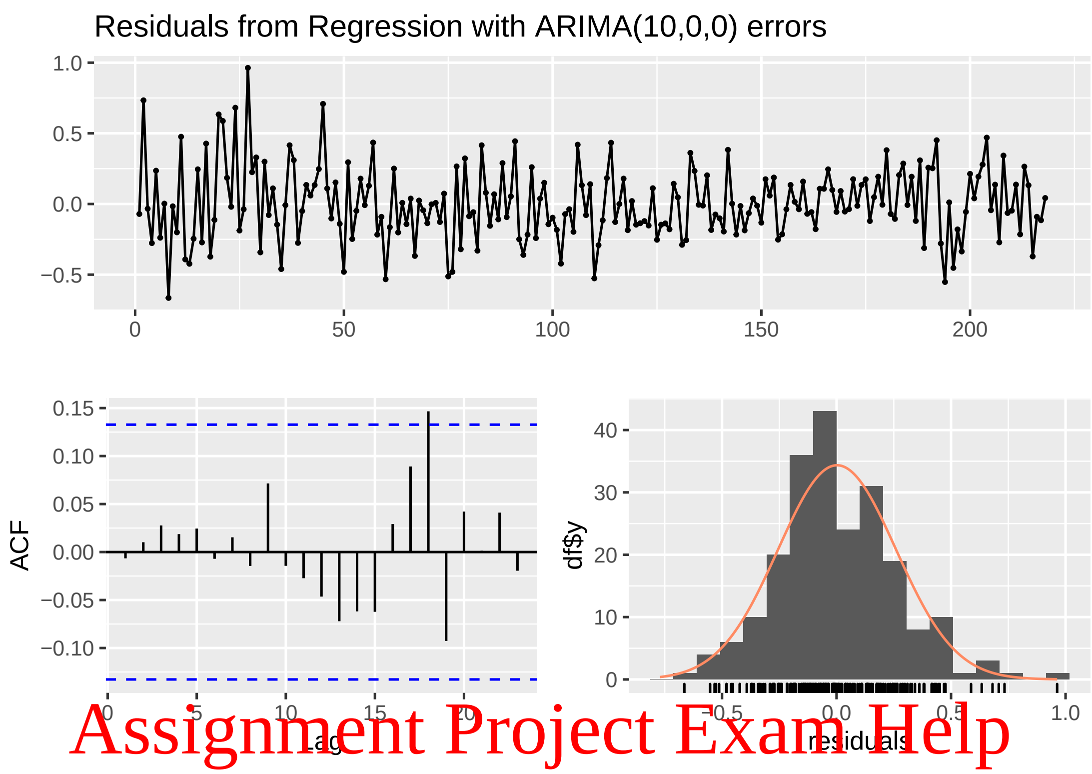

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 13.283, df = 7, p-value = 0.0655
##
## Model df: 3.   Total lags used: 10
```



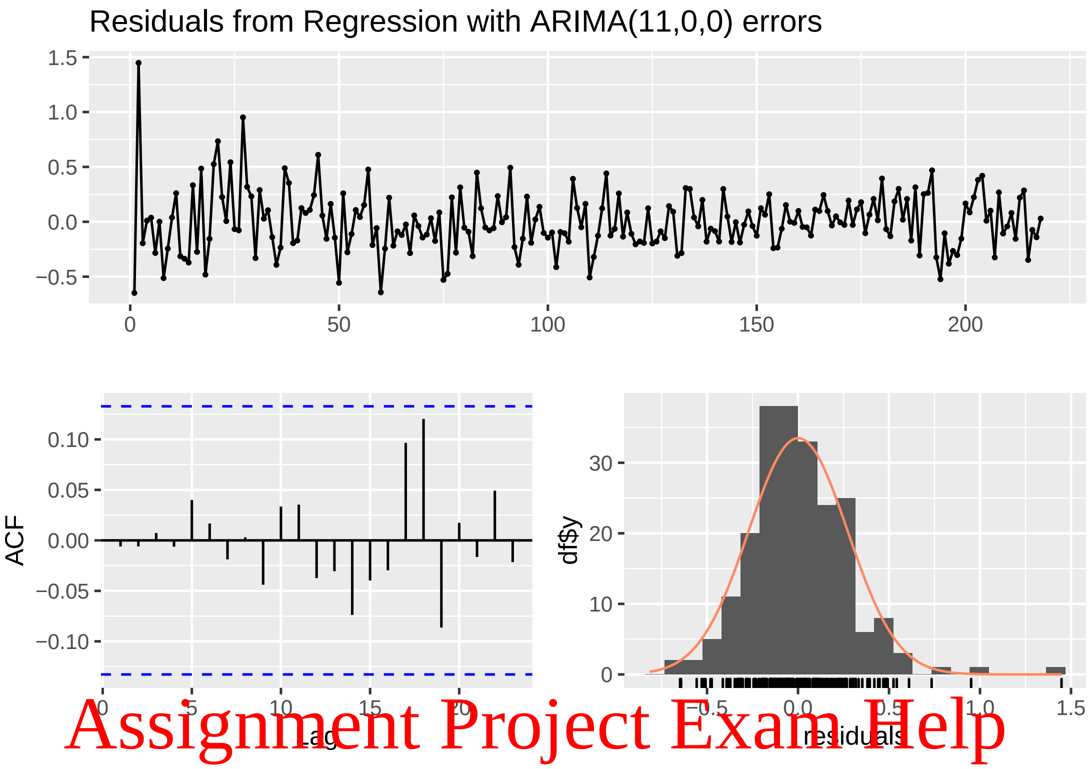

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 11.812, df = 6, p-value = 0.0663
##
## Model df: 4.   Total lags used: 10
```



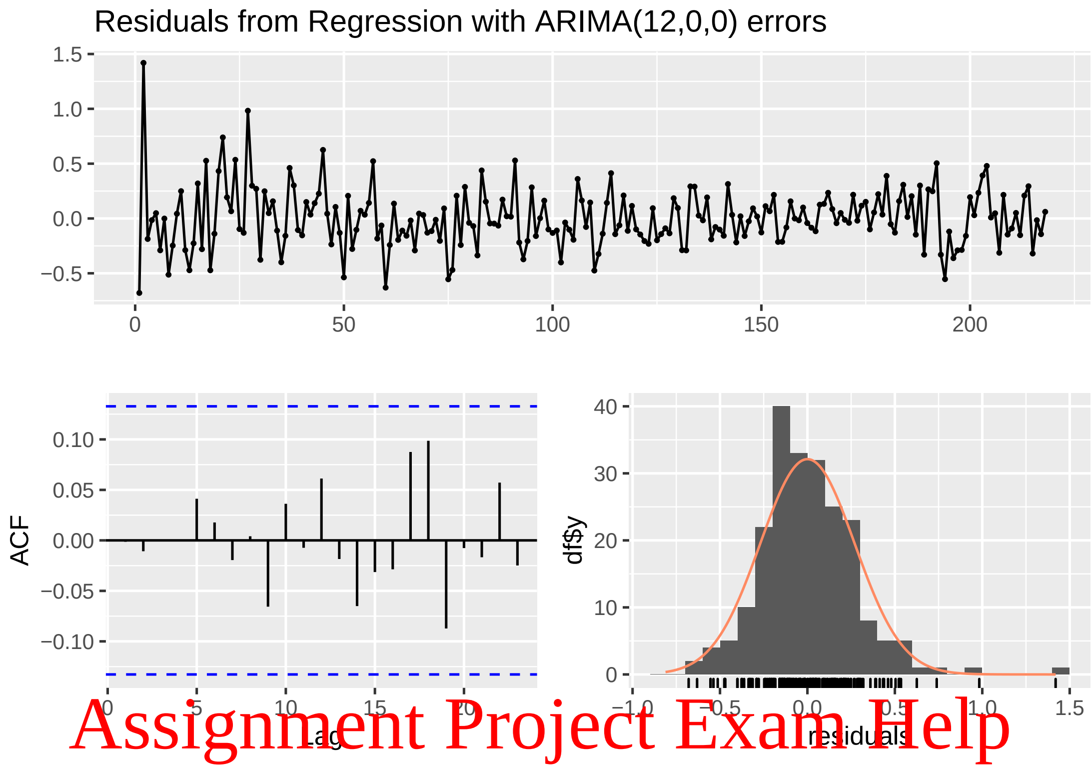

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(9,0,0) errors
## Q* = 8.2944, df = 3, p-value = 0.0403
##
## Model df: 12.   Total lags used: 15
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(10,0,0) errors
## Q* = 5.6557, df = 3, p-value = 0.1296
##
## Model df: 13.   Total lags used: 16
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(11,0,0) errors
## Q* = 6.1653, df = 3, p-value = 0.1038
##
## Model df: 14.   Total lags used: 17
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(12,0,0) errors
## Q* = 8.4108, df = 3, p-value = 0.03824
##
## Model df: 15.   Total lags used: 18
```

```
adf.test(i5y, nlag = 12)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##       lag    ADF p.value
##  [1,]   0 -1.058   0.300
##  [2,]   1 -0.796   0.394
##  [3,]   2 -1.268   0.225
##  [4,]   3 -1.133   0.274
##  [5,]   4 -1.125   0.276
##  [6,]   5 -1.130   0.275
##  [7,]   6 -1.017   0.315
##  [8,]   7 -1.008   0.318
##  [9,]   8 -0.816   0.387
## [10,]   9 -0.784   0.398
## [11,]  10 -0.785   0.398
```

```
## [12,]  11 -0.913   0.352
## Type 2: with drift no trend
##        lag   ADF p.value
##   [1,]   0 -1.69  0.4488
##   [2,]   1 -1.88  0.3735
##   [3,]   2 -2.63  0.0922
##   [4,]   3 -2.75  0.0730
##   [5,]   4 -2.60  0.0972
##   [6,]   5 -2.30  0.2089
##   [7,]   6 -2.13  0.2773
##   [8,]   7 -2.34  0.1955
##   [9,]   8 -2.01  0.3246
## [10,]   9 -2.22  0.2403
## [11,]  10 -1.99  0.3310
## [12,]  11 -1.91  0.3620
## Type 3: with drift and trend
##        lag   ADF p.value
##   [1,]   0 -2.21  0.4869
##   [2,]   1 -2.78  0.2467
##   [3,]   2 -3.22  0.0851
##   [4,]   3 -3.59  0.0347
##   [5,]   4 -3.39  0.0565
##   [6,]   5 -2.96  0.1745
##   [7,]   6 -2.88  0.2073
##   [8,]   7 -3.21  0.0868
##   [9,]   8 -3.04  0.1379
## [10,]   9 -3.48  0.0455
## [11,]  10 -3.15  0.0973
## [12,]  11 -2.86  0.2133
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```



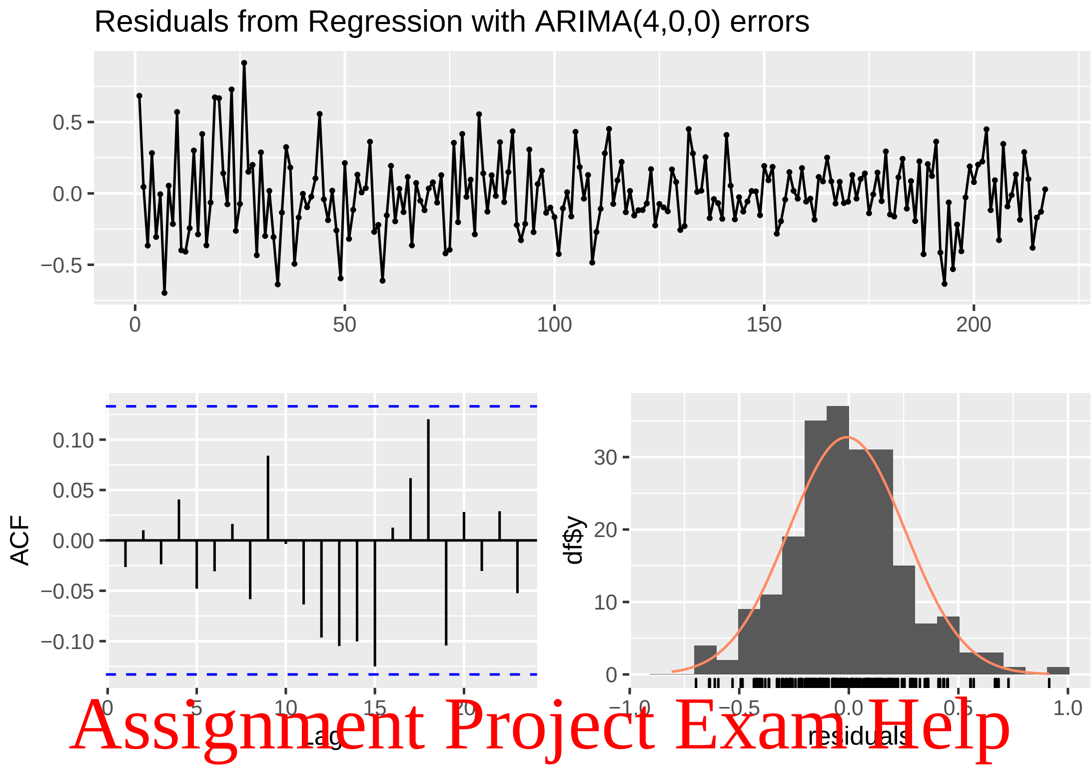
```r
i5y_ADF_diff <- ADF_estimate_diff(i5y, p_max = 15)
print(i5y_ADF_diff$ic_aic_diff)
```

```
##        const trend  p      aic       bic
##   [1,]     0     0  4 53.98123  74.26061
##   [2,]     0     0 12 54.93903 102.25759
##   [3,]     0     0  5 55.07313  78.73242
##   [4,]     1     0  4 55.81398  79.47326
##   [5,]     0     0 13 56.01841 106.71687
##   [6,]     0     0  6 56.29025  83.32943
##   [7,]     1     0 12 56.61626 107.31472
##   [8,]     1     0  5 56.89064  83.92982
##   [9,]     0     0 11 57.33228 101.27094
## [10,]     1     0 13 57.65940 111.73776
```

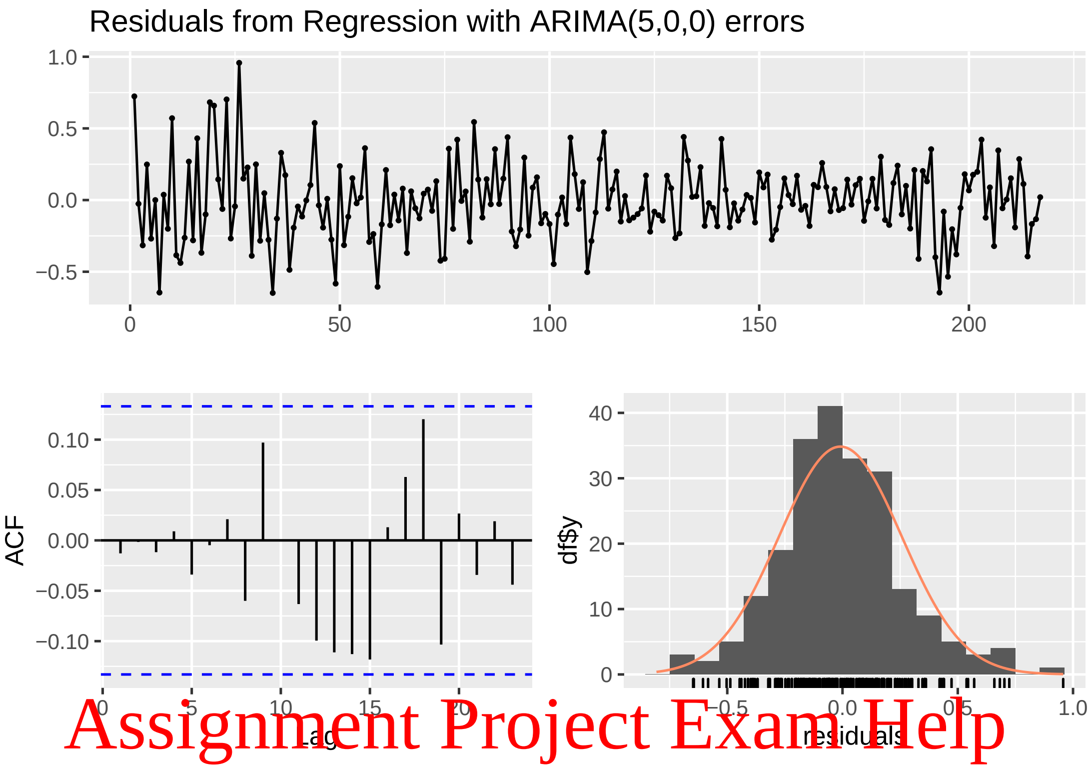
```r
print(i5y_ADF_diff$ic_bic_diff)
```

```
##       const trend p      aic      bic
##  [1,]     0     0 1 62.26382 72.40352
##  [2,]     0     0 4 53.98123 74.26061
##  [3,]     0     0 2 62.62992 76.14951
##  [4,]     1     0 1 64.20689 77.72648
##  [5,]     0     0 5 55.07313 78.73242
##  [6,]     0     0 3 61.92017 78.81966
##  [7,]     1     0 4 55.81398 79.47326
##  [8,]     1     0 2 64.54523 81.44472
##  [9,]     1     1 1 66.04324 82.94273
## [10,]     0     0 6 56.29025 83.32943
```

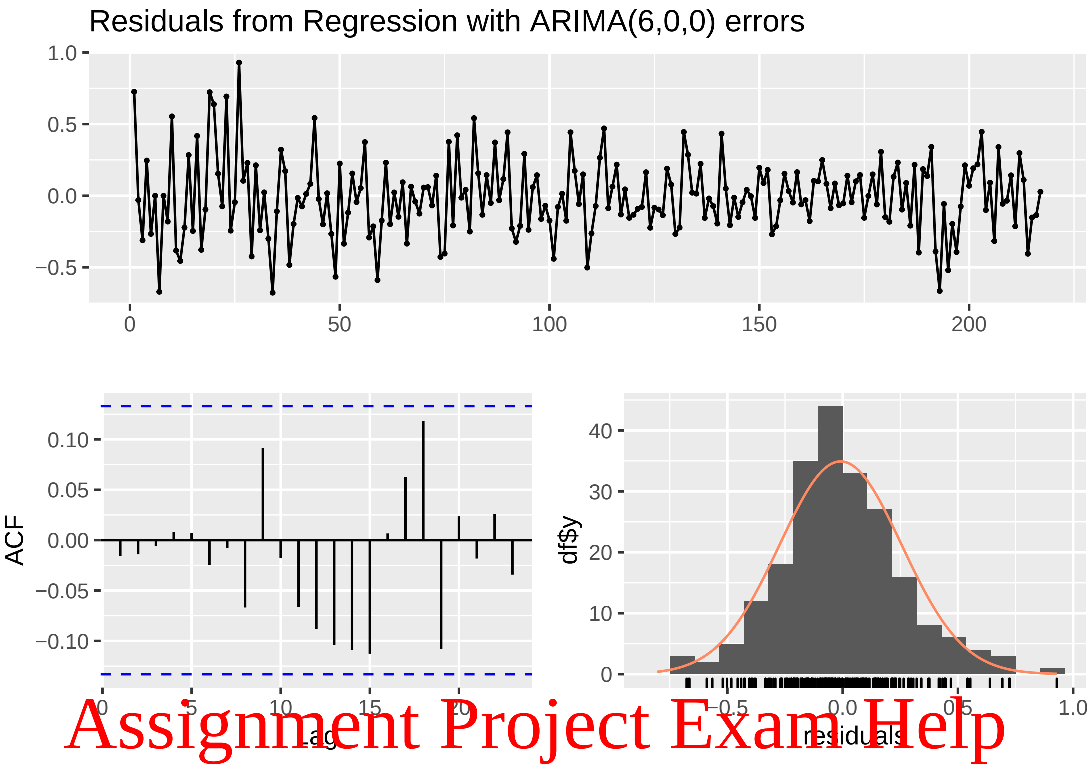
```r
i5y_adq_set_diff <- as.matrix(arrange(as.data.frame(
  i5y_ADF_diff$ic_bic_diff[c(2, 5, 7, 10),]),
  const, trend, p))
i5y_adq_idx_diff <- match(data.frame(
  t(i5y_adq_set_diff[, 1:3])),
  data.frame(
    t(i5y_ADF_diff$ic_diff[, 1:3])))

for (i in 1:length(i5y_adq_idx_diff))
{
  checkresiduals(
    i5y_ADF_diff$ADF_est_diff[[i5y_adq_idx_diff[i]]])
}
```



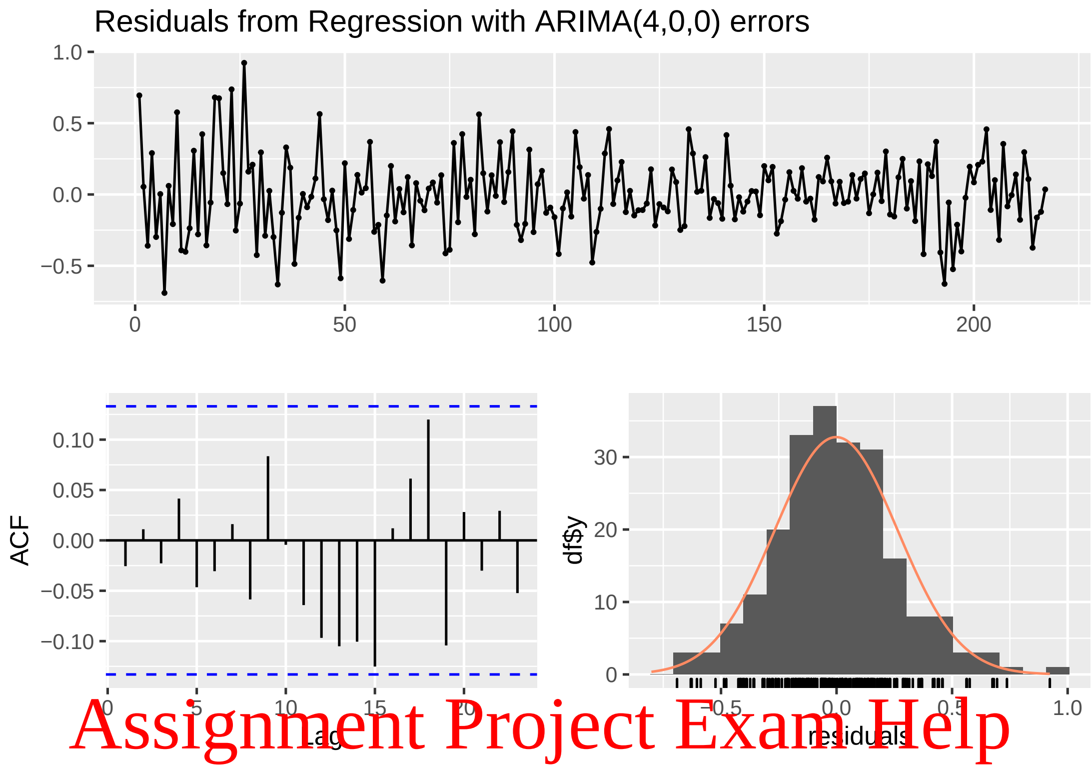

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(4,0,0) errors
## Q* = 3.8498, df = 5, p-value = 0.5712
##
## Model df: 5.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(5,0,0) errors
## Q* = 3.42, df = 4, p-value = 0.4901
##
## Model df: 6.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(6,0,0) errors
## Q* = 3.2818, df = 3, p-value = 0.3502
##
## Model df: 7.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(4,0,0) errors
## Q* = 3.8035, df = 4, p-value = 0.4332
##
## Model df: 6.   Total lags used: 10
```

```
adf.test(diff(i5y))
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -12.09    0.01
## [2,]   1  -8.35    0.01
## [3,]   2  -6.39    0.01
## [4,]   3  -6.31    0.01
## [5,]   4  -6.71    0.01
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0 -12.08    0.01
## [2,]   1  -8.37    0.01
## [3,]   2  -6.41    0.01
## [4,]   3  -6.33    0.01
```

```
## [5,]   4  -6.73    0.01
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0 -12.05    0.01
## [2,]   1  -8.35    0.01
## [3,]   2  -6.39    0.01
## [4,]   3  -6.32    0.01
## [5,]   4  -6.72    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

For $\{\text{i5y}_t\}$, we fail to reject H_0 at the 5% significance level for all specifications in the adequate set except the one with a constant, trend and $p = 10$. The results are rather inconclusive.

We might lean towards inferring that $\{\text{i5y}_t\}$ is not empirically distinguishable from a unit root process, but there is a lot of uncertainty in this conclusion. Unfortunately, the ADF test in this case does not lead us to a conclusion with a great deal of confidence, and there is not much else we can do to quantify uncertain within this methodology.

On the other hand, the null is rejected in all specifications for $\{\Delta\text{i5y}_t\}$, leading us to conclude that it is empirically distinguishable from I(1). Thus, we can infer that $\{\text{i5y}_t\}$ is empirically distinguishable from any unit root process with an order of integration greater than one, with a high degree of confidence. However, we do not have conclusive inference on how distinguishable it is from an I(1).

Repeating for $\{\text{i90d}_t\}$ and $\{\Delta\text{i90d}_t\}$, we obtain the following.


```
egr_ADF_lev <- ADF_estimate_lev(i90d, p_max = 15)
print(egr_ADF_lev$ic_aic)
```

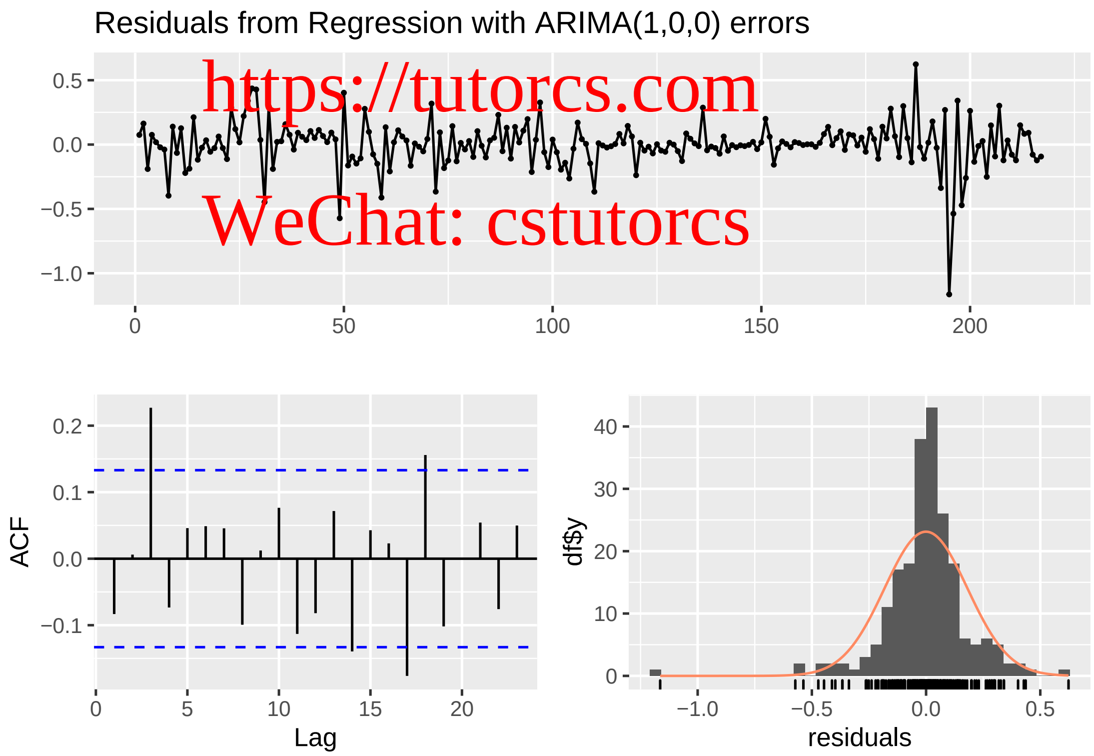
```
##       const trend p       aic        bic
##  [1,]     1     0 4 -124.6213 -100.96201
##  [2,]     1     1 4 -123.5817  -96.54254
##  [3,]     1     0 5 -123.1565  -96.11733
##  [4,]     1     1 5 -122.0429  -91.62381
##  [5,]     1     0 3 -121.5855 -101.30607
##  [6,]     1     0 6 -121.2449  -90.82580
##  [7,]     1     0 8 -120.9319  -83.75307
##  [8,]     1     1 3 -120.3162  -96.65695
##  [9,]     1     1 6 -120.1657  -86.36673
## [10,]     1     1 8 -120.1257  -79.56690
```

```
print(egr_ADF_lev$ic_bic)
```

```
##       const trend p       aic        bic
##  [1,]     1     0 1 -116.7680 -103.24839
##  [2,]     1     0 2 -119.4153 -102.51579
```

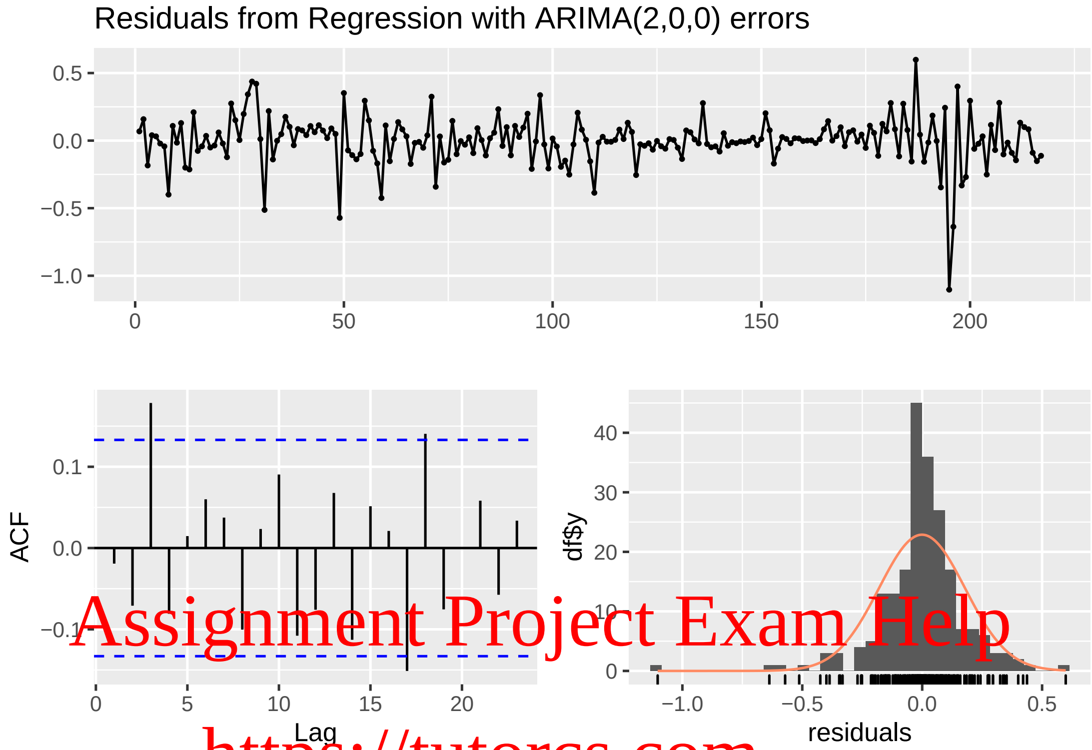
```
##  [3,]     0     0 1 -111.5606 -101.42094
##  [4,]     1     0 3 -121.5855 -101.30607
##  [5,]     1     0 4 -124.6213 -100.96201
##  [6,]     1     1 1 -115.5881  -98.68865
##  [7,]     0     0 2 -112.1658  -98.64624
##  [8,]     1     1 2 -118.3177  -98.03834
##  [9,]     0     0 4 -117.4682  -97.18884
## [10,]     1     1 3 -120.3162  -96.65695
```

```r
egr_adq_set <- as.matrix(arrange(as.data.frame(
  rbind(egr_ADF_lev$ic_aic[c(1, 2, 5),],
        egr_ADF_lev$ic_bic[c(1, 2),])),
  const, trend, p))
egr_adq_idx <- match(data.frame(t(egr_adq_set[, 1:3])),
                     data.frame(t(egr_ADF_lev$ic[, 1:3])))

for (i in 1:length(egr_adq_idx))
{
  checkresiduals(egr_ADF_lev$ADF_est[[egr_adq_idx[i]]])
}
```

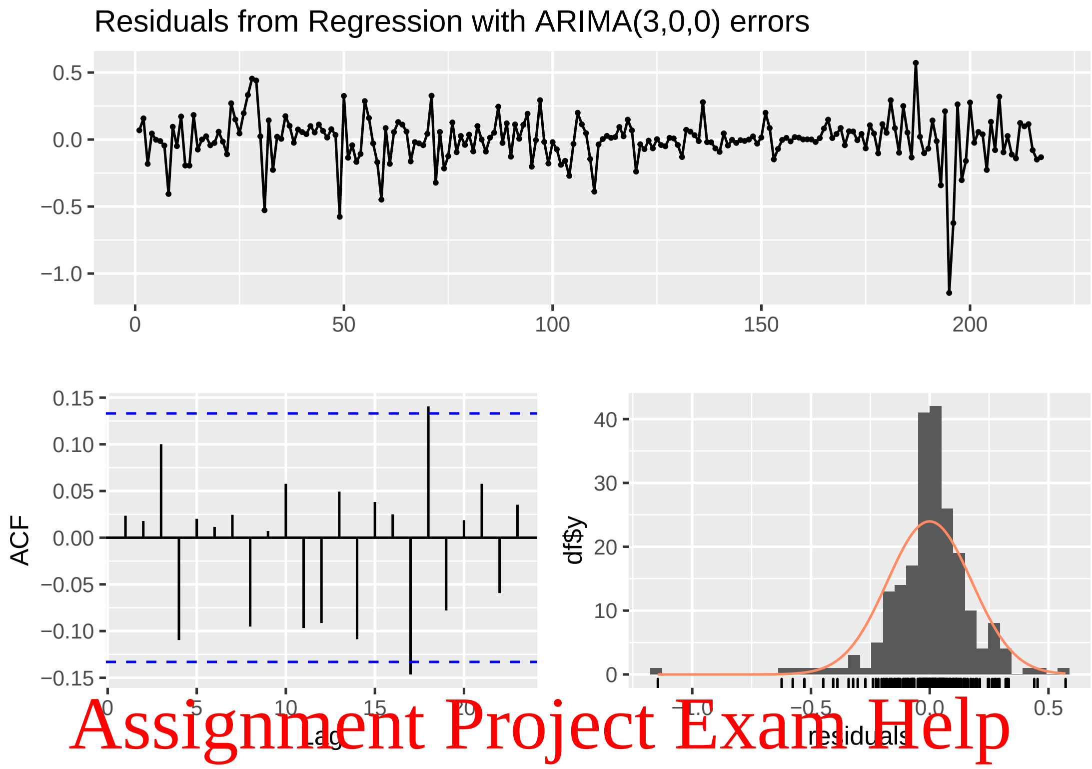




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 19.272, df = 7, p-value = 0.007378
```

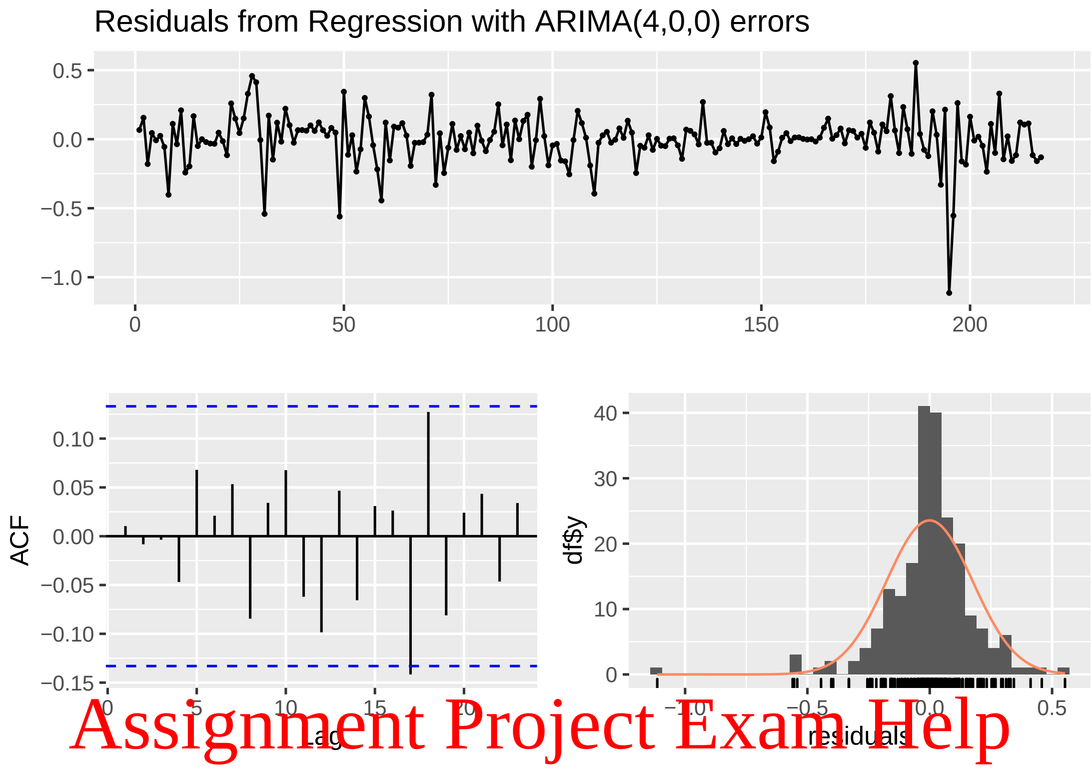
```
##
## Model df: 3.   Total lags used: 10
```



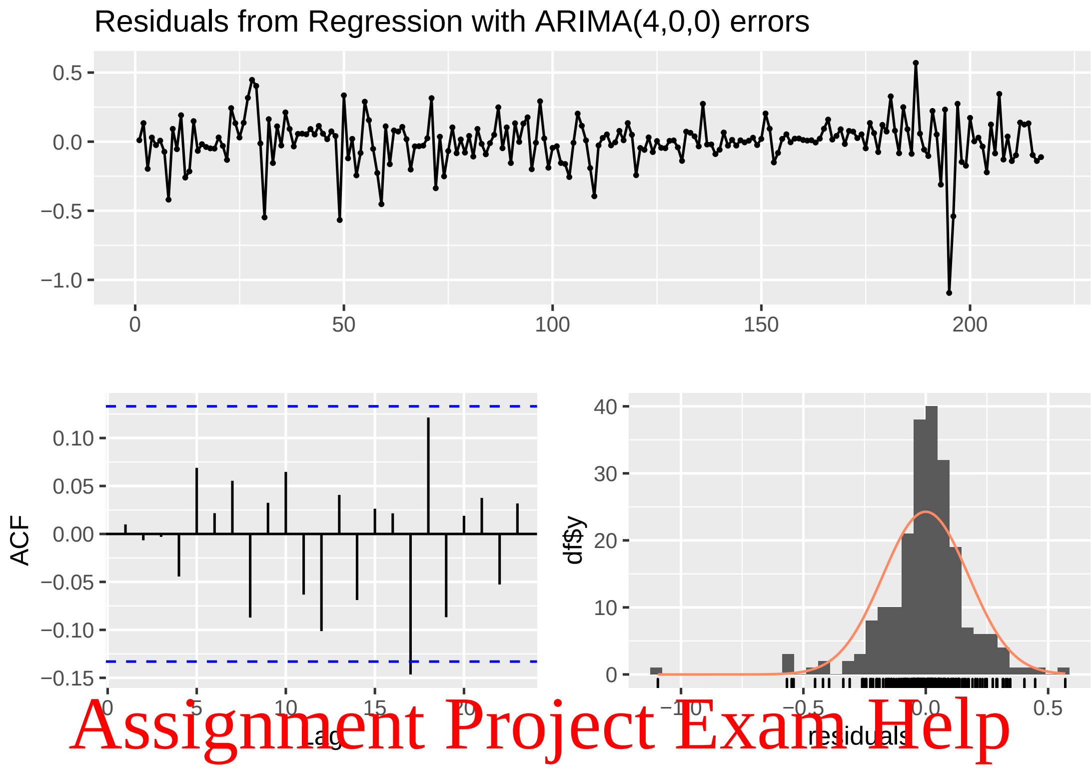

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 15.193, df = 6, p-value = 0.01881
##
## Model df: 4.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 8.1896, df = 5, p-value = 0.1461
##
## Model df: 5.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(4,0,0) errors
## Q* = 5.2493, df = 4, p-value = 0.2627
##
## Model df: 6.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(4,0,0) errors
## Q* = 5.2699, df = 3, p-value = 0.1531
##
## Model df: 7.   Total lags used: 10
```

```
adf.test(i90d)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##       lag    ADF p.value
## [1,]   0 -0.534   0.488
## [2,]   1 -0.726   0.419
## [3,]   2 -0.839   0.379
## [4,]   3 -0.832   0.381
## [5,]   4 -0.764   0.406
## Type 2: with drift no trend
##       lag   ADF p.value
## [1,]   0 -1.38  0.5646
## [2,]   1 -2.78  0.0665
## [3,]   2 -3.19  0.0230
## [4,]   3 -3.71  0.0100
```

```
## [5,]   4 -3.16  0.0243
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,]   0 -1.50  0.7837
## [2,]   1 -2.89  0.2003
## [3,]   2 -3.28  0.0749
## [4,]   3 -3.84  0.0178
## [5,]   4 -3.29  0.0736
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

For all specifications in the adequate set, we reject H_0 at the 5% significance level and conclude that $\{\text{i90d}_t\}$ is empirically distinguishable from a unit root process. Since we infer that i90d is I(0), there is no need to run the test on $\{\Delta\text{i90d}_t\}$.

Repeating for $\{\text{i180d}_t\}$ and $\{\Delta\text{i180d}_t\}$, we obtain the following.

```
i180d_ADF_lev <- ADF_estimate_lev(i180d, p_max = 15)
print(i180d_ADF_lev$ic_aic)
```



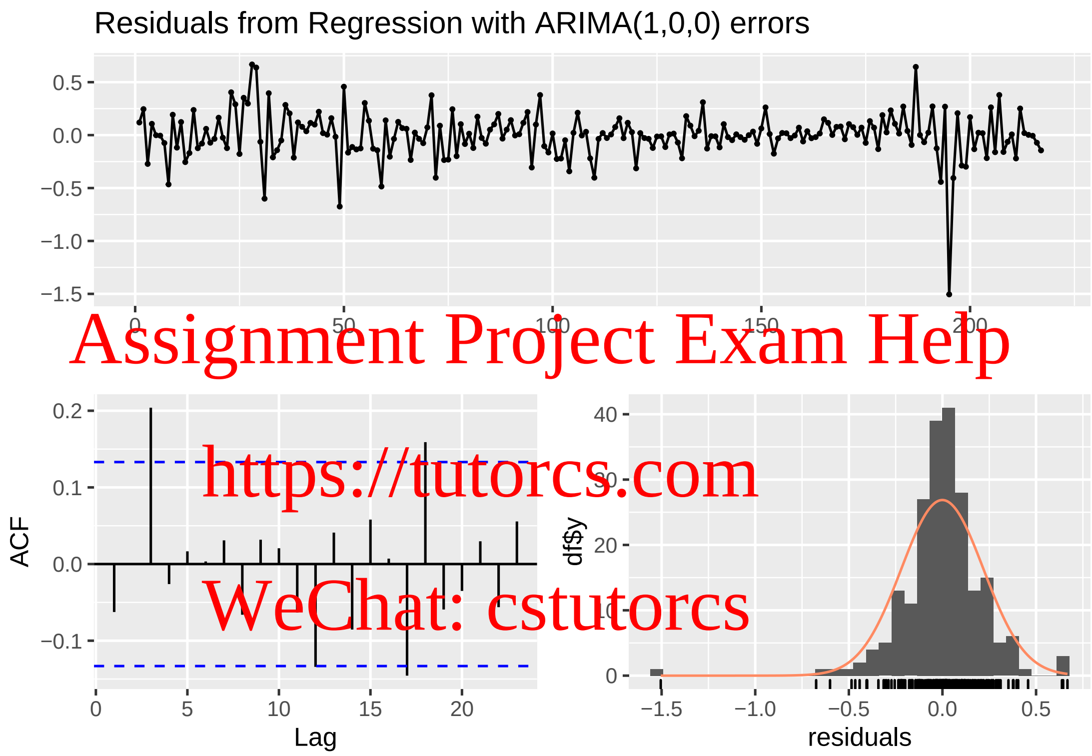
```
##       const trend p       aic        bic
##  [1,]     1     0 4 -39.46503 -15.8057520
##  [2,]     1     1 4 -38.43833 -11.3991539
##  [3,]     1     0 3 -38.40986 -18.1304805
##  [4,]     1     0 5 -37.46849 -10.4293181
##  [5,]     1     1 3 -37.21378 -13.5544981
##  [6,]     1     1 5 -36.43834  -6.0192666
##  [7,]     1     0 2 -35.76414 -18.8646555
##  [8,]     1     0 6 -35.62162  -5.2025467
##  [9,]     1     1 2 -34.75054 -14.4711549
## [10,]     1     1 6 -34.63545  -0.8364764
```

```
print(i180d_ADF_lev$ic_bic)
```

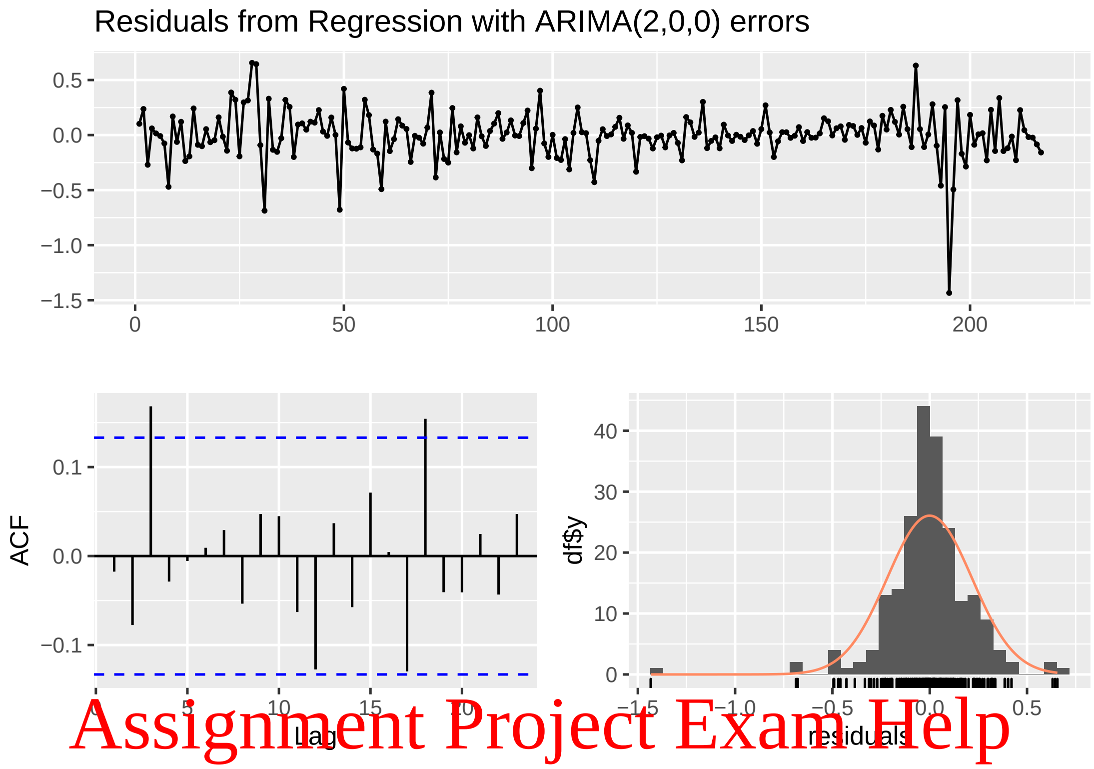
```
##       const trend p       aic       bic
##  [1,]     1     0 1 -34.49656 -20.97697
##  [2,]     1     0 2 -35.76414 -18.86466
##  [3,]     0     0 1 -28.62834 -18.48865
##  [4,]     1     0 3 -38.40986 -18.13048
##  [5,]     1     1 1 -33.38467 -16.48518
##  [6,]     1     0 4 -39.46503 -15.80575
##  [7,]     0     0 2 -28.05423 -14.53464
##  [8,]     1     1 2 -34.75054 -14.47115
##  [9,]     1     1 3 -37.21378 -13.55450
## [10,]     0     0 3 -29.18731 -12.28782
```

```
i180d_adq_set <- as.matrix(arrange(as.data.frame(
  rbind(i180d_ADF_lev$ic_aic[c(1:3, 5, 7, 9),],
```

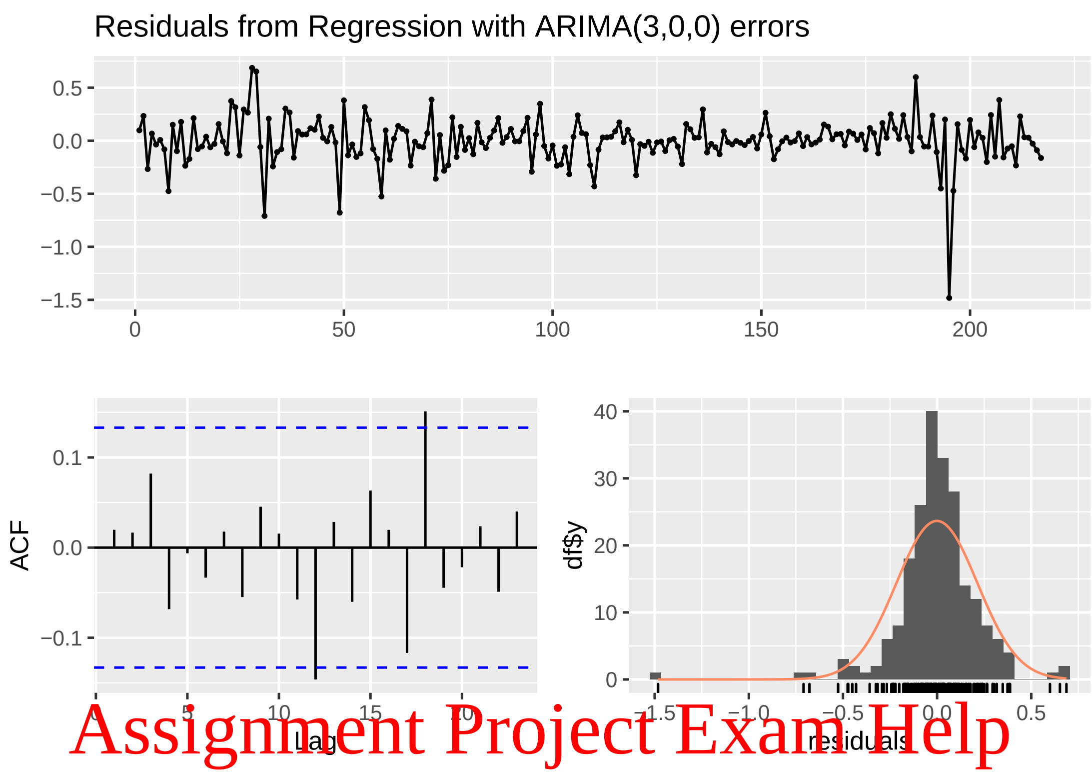
```
        i180d_ADF_lev$ic_bic[c(1, 5),])),
  const, trend, p))
i180d_adq_idx <- match(data.frame(t(i180d_adq_set[, 1:3])),
                     data.frame(t(i180d_ADF_lev$ic[, 1:3])))

for (i in 1:length(i180d_adq_idx))
{
  checkresiduals(i180d_ADF_lev$ADF_est[[i180d_adq_idx[i]]])
}
```



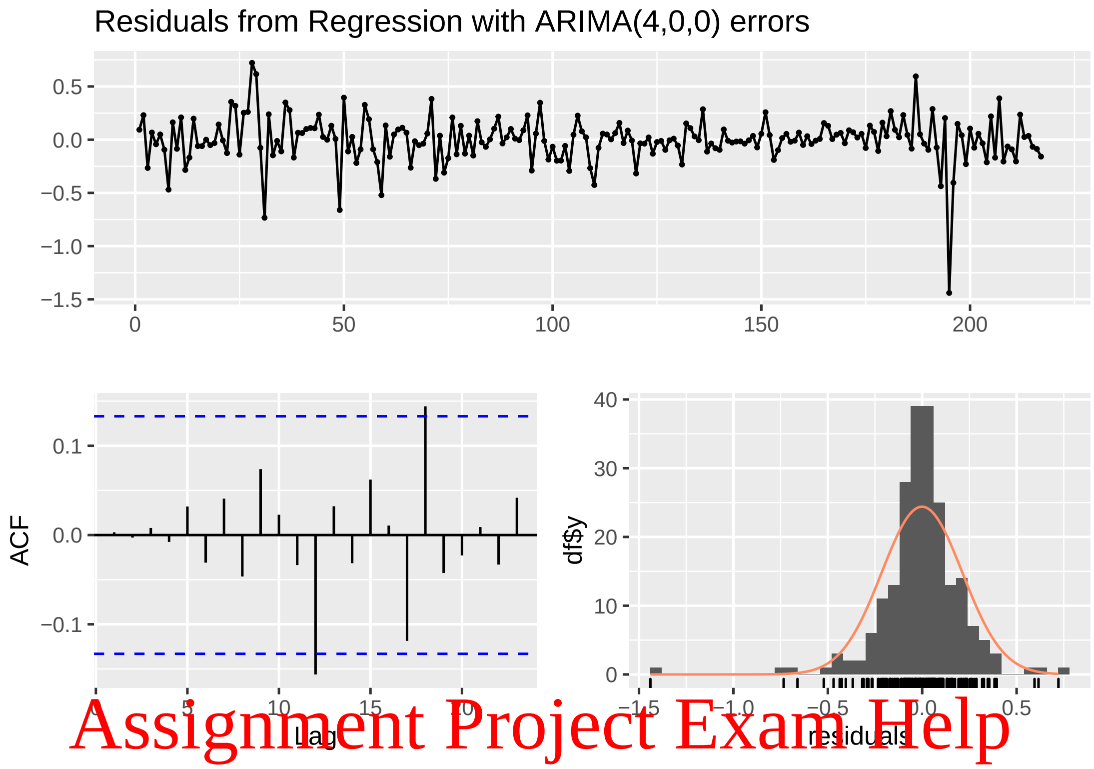

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 11.854, df = 7, p-value = 0.1055
##
## Model df: 3.   Total lags used: 10
```



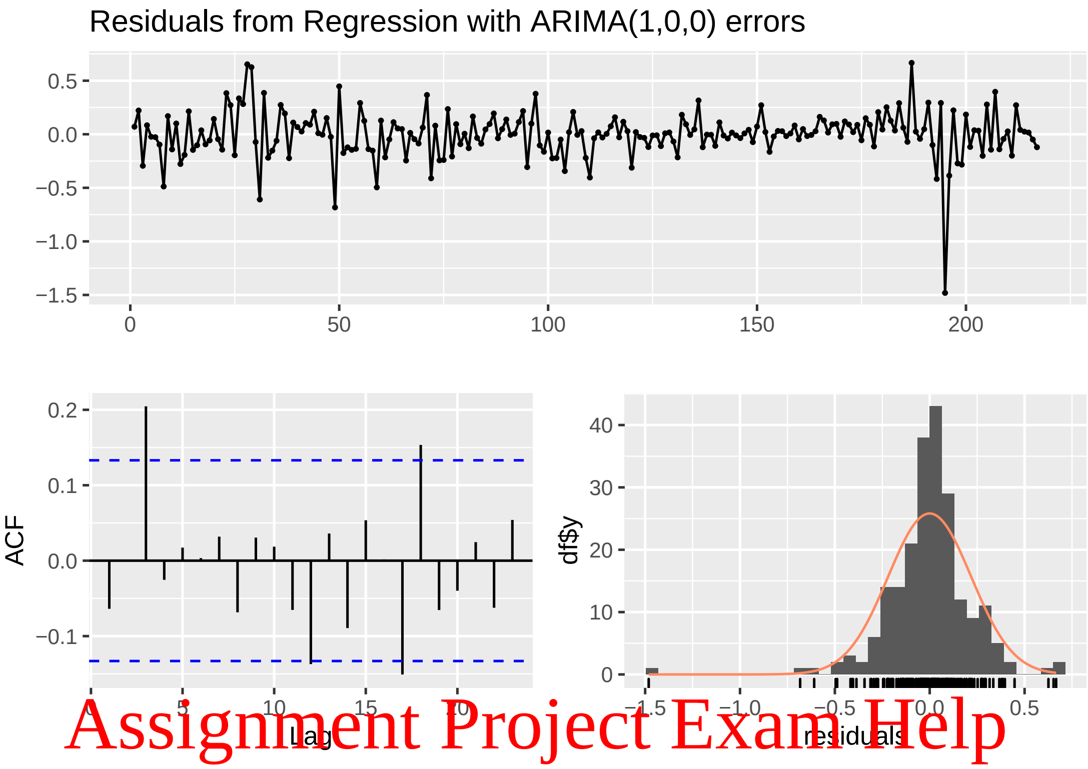

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 9.7071, df = 6, p-value = 0.1375
##
## Model df: 4.   Total lags used: 10
```



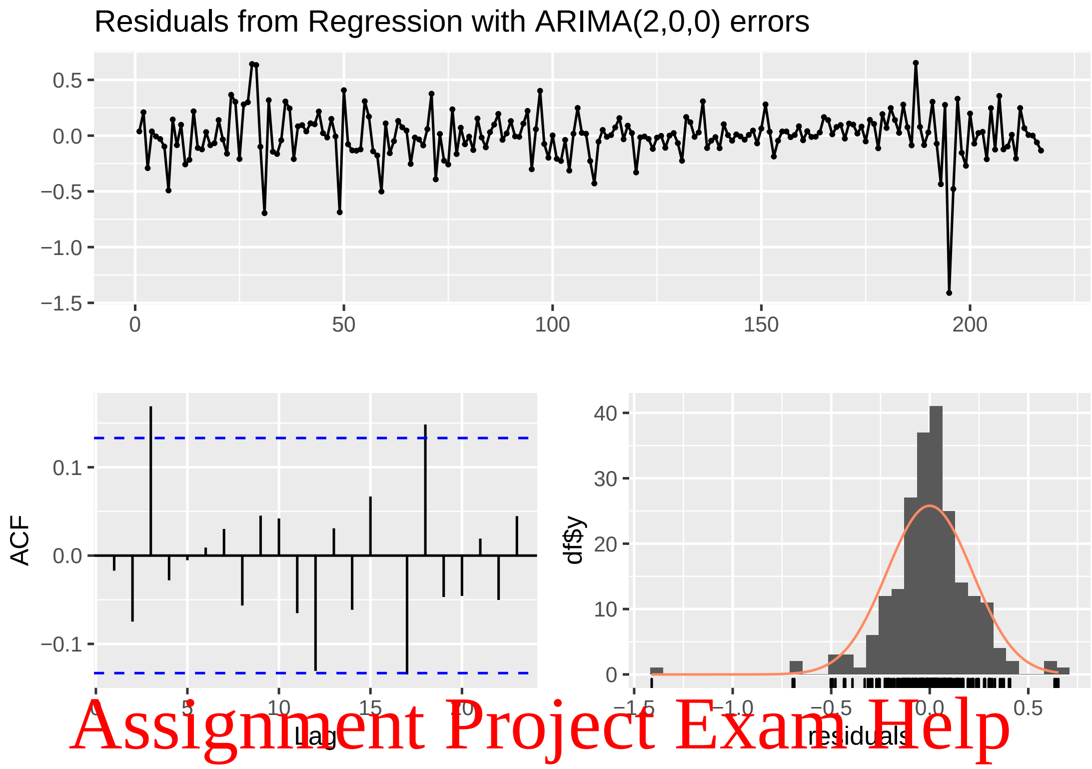

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 4.2242, df = 5, p-value = 0.5176
##
## Model df: 5.   Total lags used: 10
```



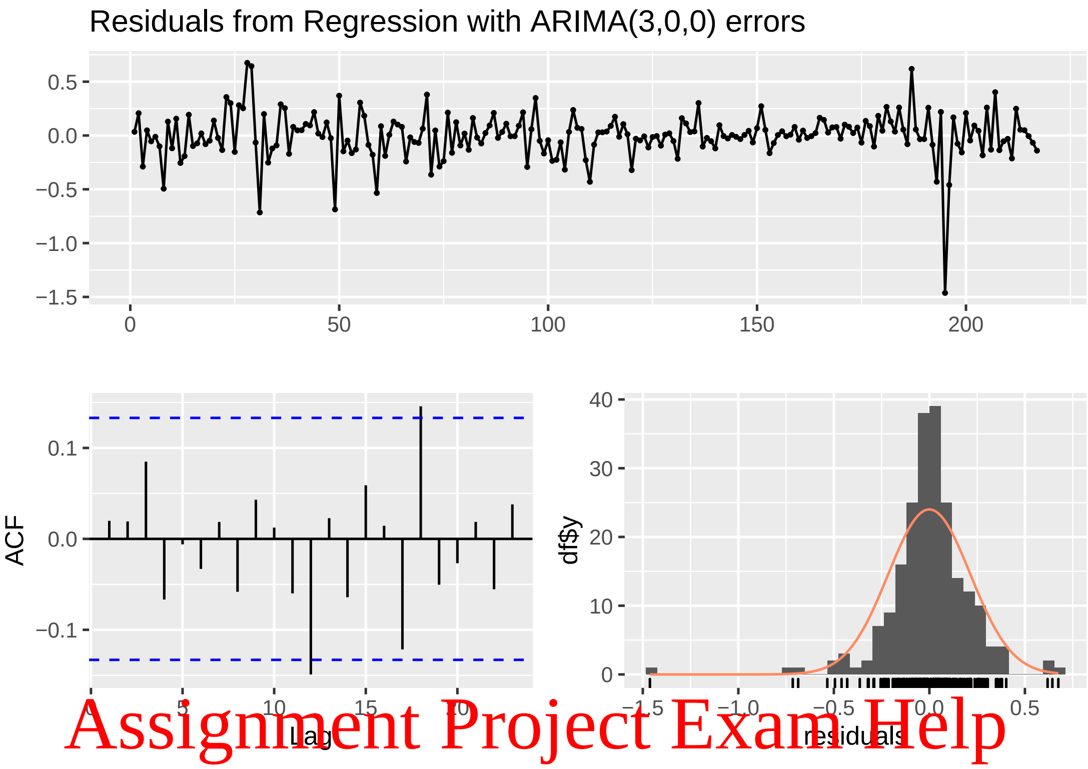

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(4,0,0) errors
## Q* = 2.7058, df = 4, p-value = 0.6082
##
## Model df: 6.   Total lags used: 10
```



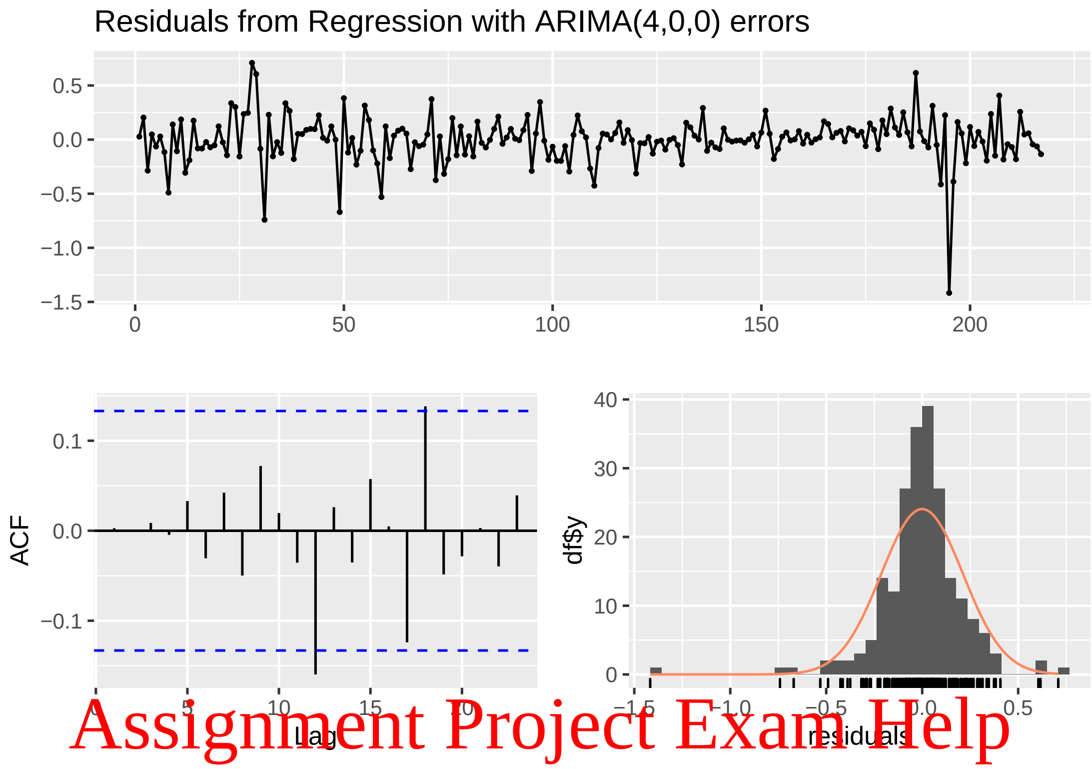

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 11.99, df = 6, p-value = 0.06218
##
## Model df: 4.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 9.6403, df = 5, p-value = 0.08609
##
## Model df: 5.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 4.3218, df = 4, p-value = 0.3642
##
## Model df: 6.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(4,0,0) errors
## Q* = 2.7166, df = 3, p-value = 0.4374
##
## Model df: 7.   Total lags used: 10
```

```
adf.test(i180d)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -0.474   0.507
## [2,]   1 -0.724   0.420
## [3,]   2 -0.843   0.377
## [4,]   3 -0.833   0.381
## [5,]   4 -0.768   0.404
## Type 2: with drift no trend
##      lag   ADF p.value
## [1,]   0 -1.57  0.4943
## [2,]   1 -2.88  0.0504
## [3,]   2 -3.19  0.0228
## [4,]   3 -3.72  0.0100
```

```
## [5,]   4 -3.28  0.0184
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,]   0 -1.71  0.6989
## [2,]   1 -2.99  0.1602
## [3,]   2 -3.28  0.0756
## [4,]   3 -3.84  0.0178
## [5,]   4 -3.40  0.0538
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```r
i180d_ADF_diff <- ADF_estimate_diff(i180d, p_max = 15)
print(i180d_ADF_diff$ic_aic_diff)
```

```
##       const trend p       aic        bic
##  [1,]     0     0 3 -31.88034 -15.003950
##  [2,]     0     0 2 -30.21348 -16.712366
##  [3,]     0     0 4 -30.03514  -9.783466
##  [4,]     1     0 3 -29.90334  -9.651665
##  [5,]     0     0 0 -29.72709 -22.976533
##  [6,]     0     0 1 -29.25261 -19.126772
##  [7,]     1     0 2 -28.24426 -11.367863
##  [8,]     0     0 5 -28.05966  -4.432713
##  [9,]     1     0 4 -28.05856  -4.431613
## [10,]     1     1 3 -27.93277  -4.305817
```

```r
print(i180d_ADF_diff$ic_bic_diff)
```

```
##       const trend p       aic        bic
##  [1,]     0     0 0 -29.72709 -22.976533
##  [2,]     0     0 1 -29.25261 -19.126772
##  [3,]     1     0 0 -27.75244 -17.626607
##  [4,]     0     0 2 -30.21348 -16.712366
##  [5,]     0     0 3 -31.88034 -15.003950
##  [6,]     1     0 1 -27.28159 -13.780481
##  [7,]     1     1 0 -25.79273 -12.291616
##  [8,]     1     0 2 -28.24426 -11.367863
##  [9,]     0     0 4 -30.03514  -9.783466
## [10,]     1     0 3 -29.90334  -9.651665
```

```r
adf.test(diff(i180d), nlag = 10)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##       lag   ADF p.value
##  [1,]   0 -8.33    0.01
```

```
##  [2,]   1 -6.71    0.01
##  [3,]   2 -5.33    0.01
##  [4,]   3 -5.74    0.01
##  [5,]   4 -5.44    0.01
##  [6,]   5 -5.36    0.01
##  [7,]   6 -4.98    0.01
##  [8,]   7 -5.13    0.01
##  [9,]   8 -4.67    0.01
## [10,]   9 -4.76    0.01
## Type 2: with drift no trend
##       lag   ADF p.value
##  [1,]   0 -8.31    0.01
##  [2,]   1 -6.70    0.01
##  [3,]   2 -5.32    0.01
##  [4,]   3 -5.73    0.01
##  [5,]   4 -5.43    0.01
##  [6,]   5 -5.35    0.01
##  [7,]   6 -4.97    0.01
##  [8,]   7 -5.11    0.01
##  [9,]   8 -4.66    0.01
## [10,]   9 -4.74    0.01
## Type 3: with drift and trend
##       lag   ADF p.value
##  [1,]   0 -8.29    0.01
##  [2,]   1 -6.68    0.01
##  [3,]   2 -5.31    0.01
##  [4,]   3 -5.72    0.01
##  [5,]   4 -5.42    0.01
##  [6,]   5 -5.34    0.01
##  [7,]   6 -4.96    0.01
##  [8,]   7 -5.12    0.01
##  [9,]   8 -4.67    0.01
## [10,]   9 -4.76    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```


For $\{\text{i180d}_t\}$, we reject H_0 at the 5% significance level in some specifications in the adequate set but not others. The ADF results are not sufficiently accurate to ascertain the proximity of the DGP to a unit root process.

For $\{\Delta\text{i180d}_t\}$, specifications with lag lengths up to 9 lead to H_0 being universally rejected at a very small significance level. The process $\{\Delta\text{i180d}_t\}$ is clearly distinguishable from a unit root process. However, we cannot say how close $\{\text{i180d}_t\}$ is to an I(1) using the ADF testing approach.

2. Use the Engle-Granger method to test for a cointegrating relation involving all four processes. Assume the 5 year TB rate is the dependent variable in the initial regression. Hint: Use the `test.coint` function provided by the `aTSA` package.

Solution We need to construct an adequate set of ADF specifications for the estimated residuals from the regression of i5y$_t$ on a constant, i3y$_t$, i90d$_t$, and i180d$_t$. A regression in R is implemented using the `lm` function.

```
eg_reg <- lm( i5y ~ i3y + i90d + i180d, mydata)
eg_res <- eg_reg$residuals
```

Now, use the same approach as in Question 1 but with `eg_res` instead of an observed sample.

```
egr_ADF_lev <- ADF_estimate_lev(eg_res, p_max = 15)
print(egr_ADF_lev$ic_aic)
```

```
##       const trend  p       aic       bic
##  [1,]     0     0  2 -420.1848 -406.6652
##  [2,]     0     0  1 -420.1357 -409.9961
##  [3,]     0     0  0 -419.9250 -413.1652
##  [4,]     0     0  3 -418.5450 -401.6455
##  [5,]     1     0  2 -418.2587 -401.3592
##  [6,]     0     0 15 -418.2065 -360.7482
##  [7,]     1     0  1 -418.1856 -404.6660
##  [8,]     1     0  0 -417.9520 -407.8123
##  [9,]     1     0  3 -416.6074 -396.3280
## [10,]     1     1  0 -416.5675 -403.0479
```

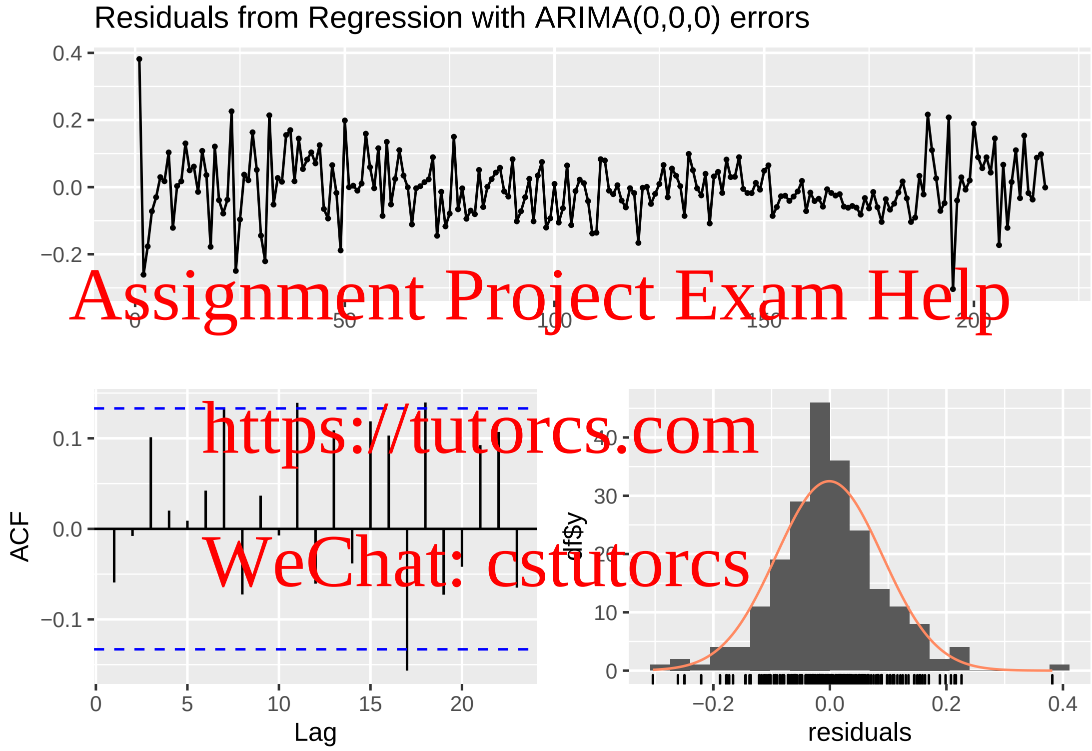


```
print(egr_ADF_lev$ic_bic)
```

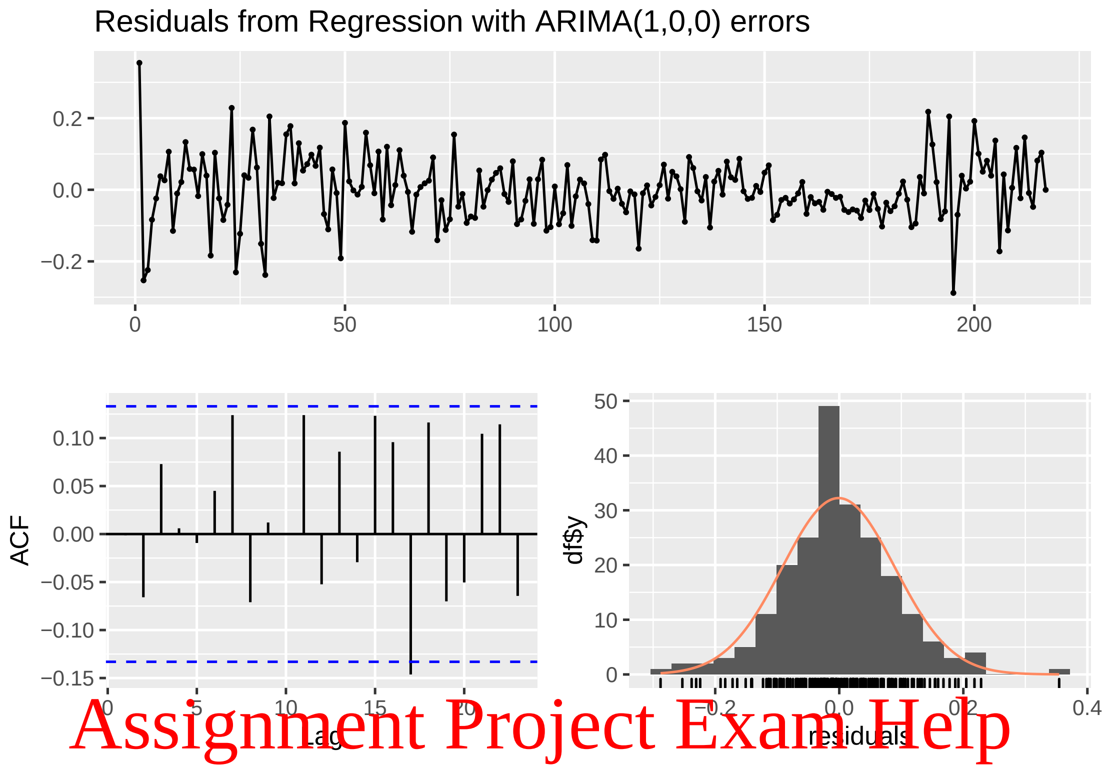
```
##       const trend p       aic       bic
##  [1,]     0     0 0 -419.9250 -413.1652
##  [2,]     0     0 1 -420.1357 -409.9961
##  [3,]     1     0 0 -417.9520 -407.8123
##  [4,]     0     0 2 -420.1848 -406.6652
##  [5,]     1     0 1 -418.1856 -404.6660
##  [6,]     1     1 0 -416.5675 -403.0479
##  [7,]     0     0 3 -418.5450 -401.6455
##  [8,]     1     0 2 -418.2587 -401.3592
##  [9,]     1     1 1 -416.5125 -399.6130
## [10,]     1     0 3 -416.6074 -396.3280
```

The only specifications that AIC and BIC really disagree on is the one with no constant, no trend and $p = 15$. We can eliminate it and proceed essentially with the top 10 specifications preferred by the BIC.

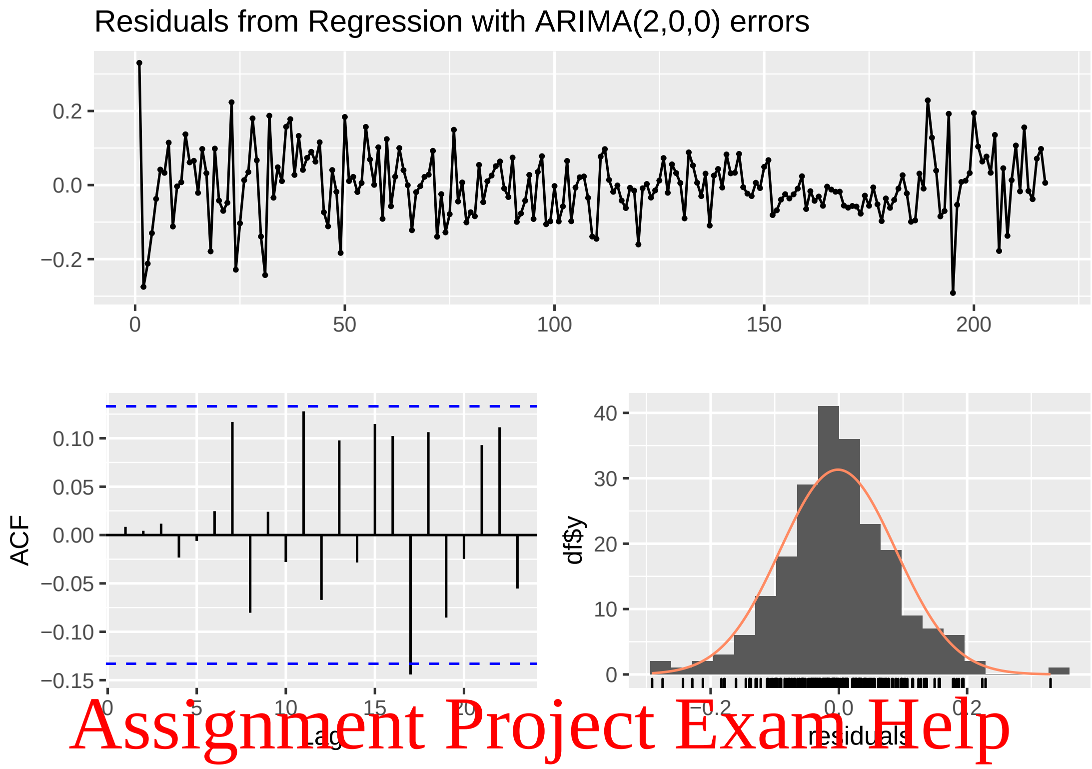
```
egr_adq_set <- as.matrix(arrange(as.data.frame(
                                    egr_ADF_lev$ic_bic),
                                    const, trend, p))
egr_adq_idx <- match(data.frame(t(egr_adq_set[, 1:3])),
                     data.frame(t(egr_ADF_lev$ic[, 1:3])))

for (i in 1:length(egr_adq_idx))
{
  checkresiduals(egr_ADF_lev$ADF_est[[egr_adq_idx[i]]])
}
```



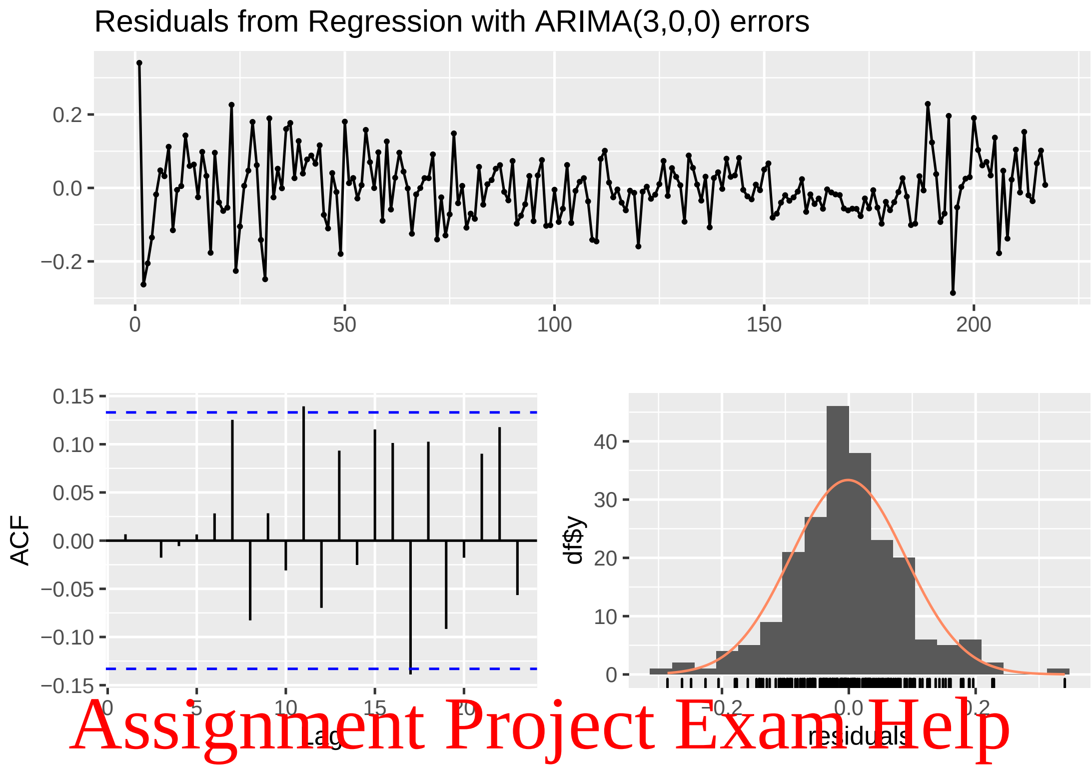

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 9.1103, df = 9, p-value = 0.4272
##
## Model df: 1.   Total lags used: 10
```



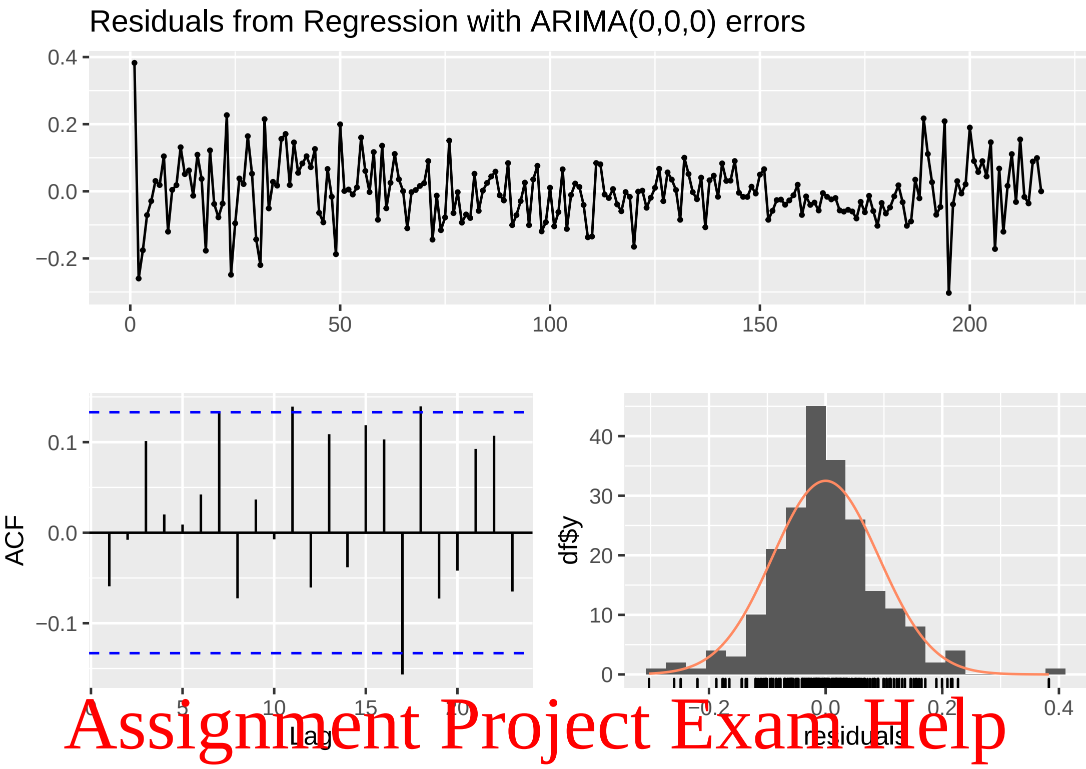

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 7.2714, df = 8, p-value = 0.5077
##
## Model df: 2.   Total lags used: 10
```



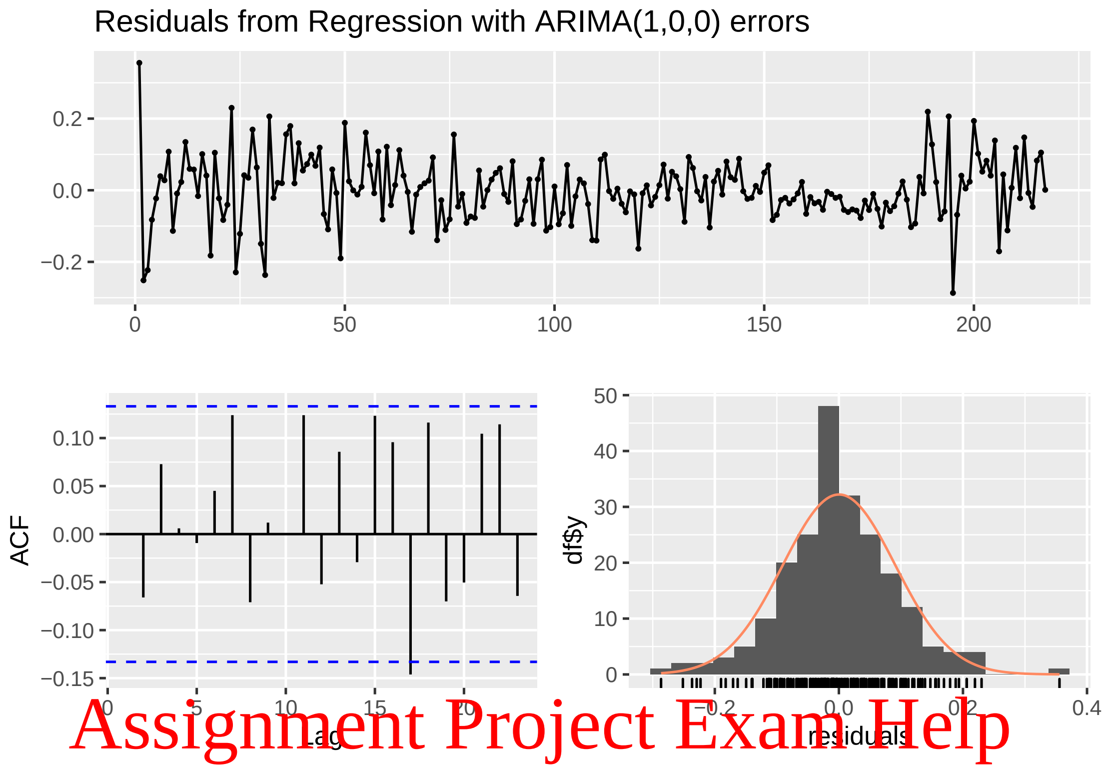

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 5.1884, df = 7, p-value = 0.637
##
## Model df: 3.   Total lags used: 10
```



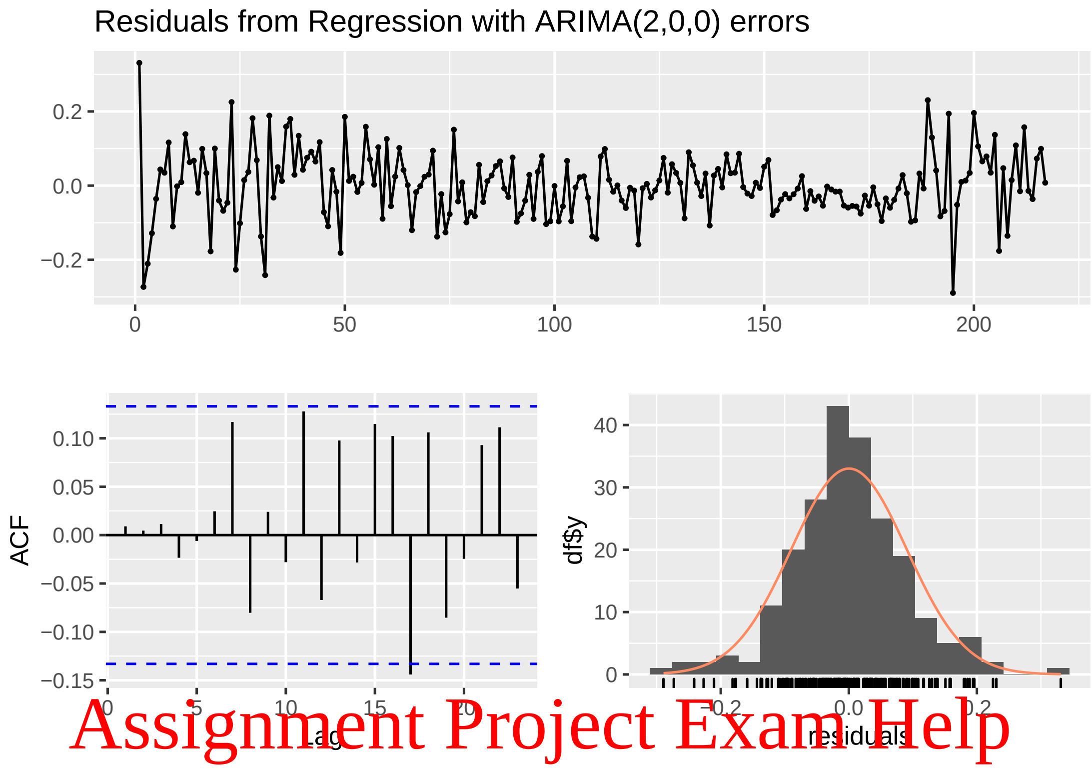

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 5.7923, df = 6, p-value = 0.4469
##
## Model df: 4.   Total lags used: 10
```



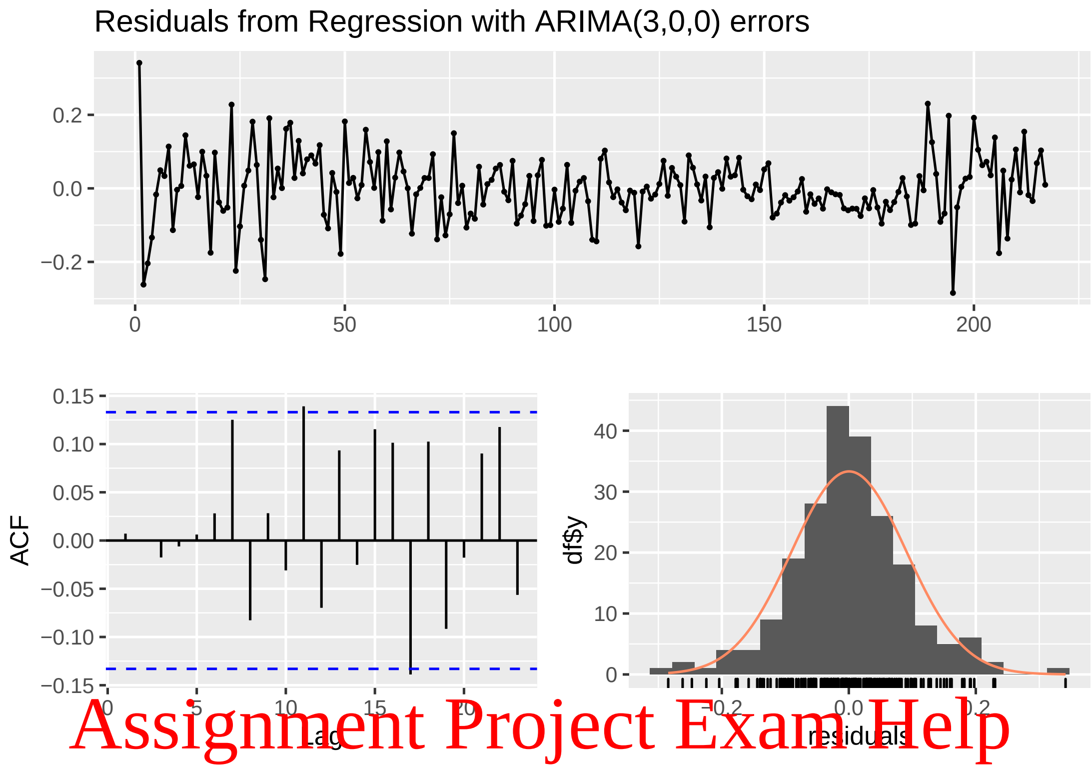

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 9.11, df = 8, p-value = 0.3331
##
## Model df: 2.   Total lags used: 10
```



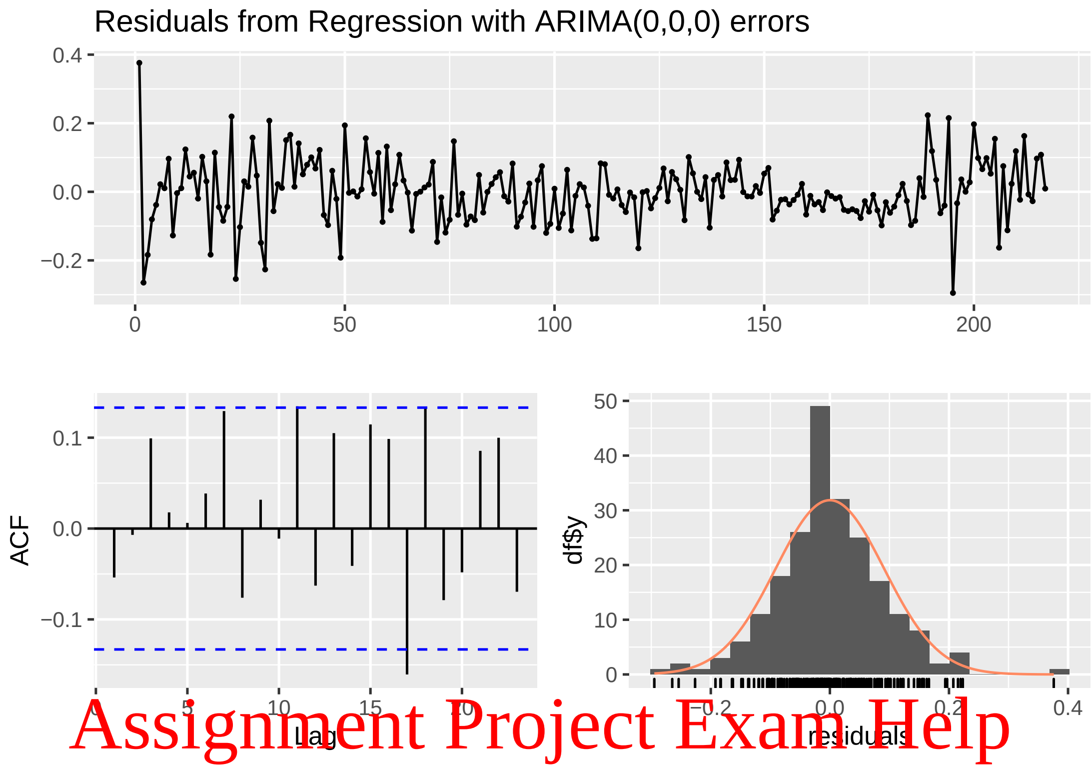

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 7.2704, df = 7, p-value = 0.4013
##
## Model df: 3.   Total lags used: 10
```



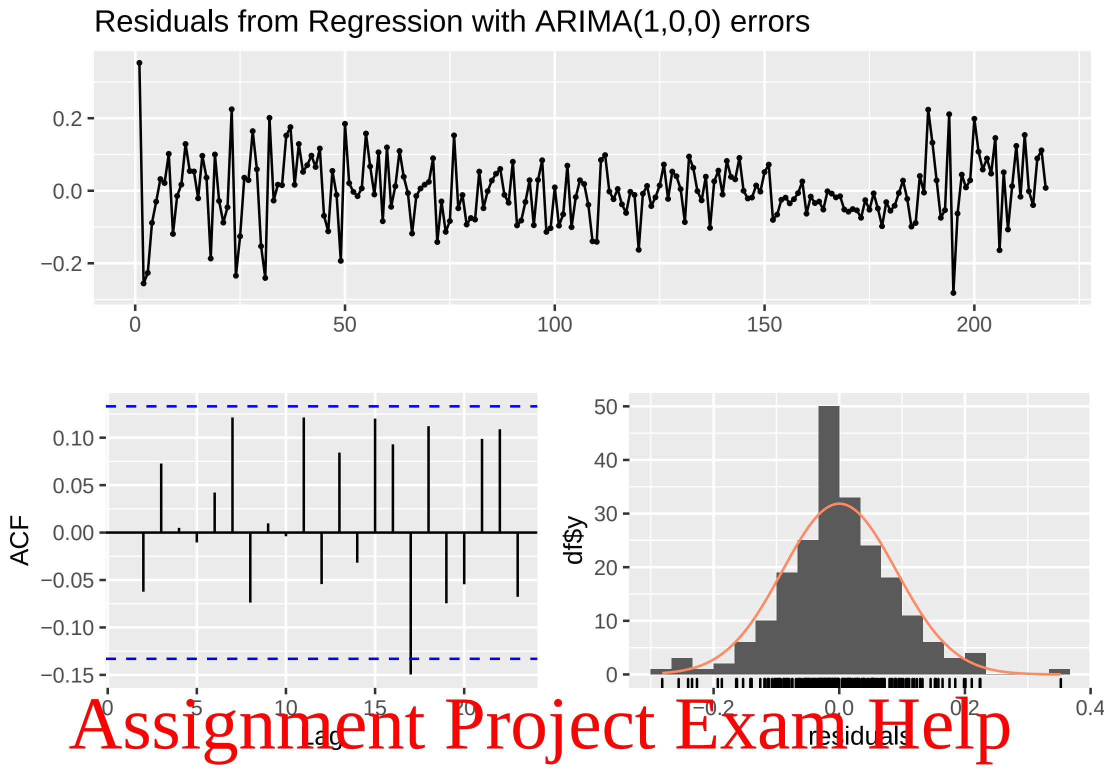

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 5.1861, df = 6, p-value = 0.5202
##
## Model df: 4.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 5.7771, df = 5, p-value = 0.3285
##
## Model df: 5.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 8.604, df = 7, p-value = 0.2823
##
## Model df: 3.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 7.0589, df = 6, p-value = 0.3154
##
## Model df: 4.   Total lags used: 10
```

All residuals look OK. Hence, we can continue and use the function `coint.test` form the `aTSA` package to implement the Engle-Granger test for each of the specifications in the adequate set. The inputs to `coint.test` are the dependent variable in the regression, a matrix containing independent variables and the number of lags to use in the unit root test on the residuals series.

We use a `for` loop to implement the test on each of the specifications in the adequate set. Instead of generating output at each iteration, we use the option `output = F` to suppress it and store the p-values in an easy-to-read table.

```
eg_test <- matrix(nrow = 3, ncol = 4)
colnames(eg_test) <- rep("", 4)
rownames(eg_test) <- c("No const, no trend",
                       "Const, no trend",
                       "Const with trend")
for (l in 1:4)
{
  eg_l <- coint.test(i5y, cbind(i3y, i90d, i180d),
```

```
                         nlag = 1, output = F)
  eg_test[, l] <- eg_l[, 3]
  colnames(eg_test)[l] <- paste("Lag", l)
}
print(eg_test)
```

```
##                       Lag 1 Lag 2 Lag 3 Lag 4
## No const, no trend    0.01  0.01  0.01  0.01
## Const, no trend       0.10  0.10  0.10  0.10
## Const with trend      0.10  0.10  0.10  0.10
```

We see again that the results of the test are inconclusive. For specifiations with no constant and no trend, the unit root in the residuals is rejected at low significance levels. But for all specifications with a constant, a unit root in the residuals cannot be rejected.

The best inference we can draw is that if the residual in the regression $\mathrm{i5y}_t$ on a constant, $\mathrm{i3y}_t$, $\mathrm{i90d}_t$, and $\mathrm{i180d}_t$ is *mean-independent*, then it also does not have a unit root. The reasoning behind this is that if a residual series is mean-independent of the regressors, then its unconditional mean is zero. This scenario matches ADF specifications above that restrict the constant to be zero.

However, if mean-independence does not hold, such that the constant in the ADF specification cannot be restricted to zero, then we generally cannot reject a unit root in the residuals process.

---

3. Interpret the inference obtained Questions 1 and 2 in terms of empirical evidence of cointegration in the four interest rates.

Solution In Question 1, we concluded that $\mathrm{i3y}_t$ is not empirically distinguishable from I(1), but for the remaining three processes our inference on their proximity to I(1) processes is rather ambiguous.

When we regress $\mathrm{i5y}_t$ on $\mathrm{i3y}_t$, $\mathrm{i90d}_t$ and $\mathrm{i180d}_t$, we find that the residuals process does not have a unit root if we enforce the restriction that residuals are mean-independent. Assuming this restriction is valid, we have the following possibilities:

1. $\mathrm{i3y}_t$, $\mathrm{i5y}_t$, $\mathrm{i90d}_t$ and $\mathrm{i180d}_t$ are all I(0);
2. any three processes are I(1) and cointegrated while a fourth is I(0); for example, we could have that $\mathrm{i3y}_t$, $\mathrm{i5y}_t$ and $\mathrm{i90d}_t$ are cointegrated and $\mathrm{i180d}_t$ is is I(0). The same could hold for any other combination.
3. Any two processes are I(1) and cointegrated while the other two are I(0); for example, we could have that $\mathrm{i3y}_t$ and $\mathrm{i5y}_t$ are cointegrated while $\mathrm{i90d}_t$ and

i180d_t are both I(0).

4. any two processes are I(1) and cointegrated, and the other two processes are also I(1) and cointegrated, but the four processes are not all cointegrated with each other in a single cointegrating relation;
5. all four processes are I(1) and cointegrated in a single cointegrating relation.

Which of these five scenarios prevails? It depends on what we assume about the integration properties of the processes involved. Our unit root tests in Question 1 did not clearly reject a unit root in any of the processes, except $\{\text{i90d}_t\}$. If $\{\text{i90d}_t\}$ is I(0), then we can rule out scenarios 4 and 5.

In terms of scenarios 1-3, we can in principle make the unit root assumption about any combination of i3y_t on i5y_t and i180d_t, which will determine the appropriate interpretation. The important thing to remember is that it is *always* an assumption that a unit root exists! Whether or not it is a useful one depends on the application.

---

4. Repeat Question 2 three more times but each time change the dependent variable. Is the inference regarding cointegration affected?

Solution The steps are exactly the same if we replace the dependent variable. For example, letting i3y_t be the dependent variable, we obtain the following.


```
eg_reg <- lm(i3y ~ i5y + i90d + i180d, mydata)
eg_res <- eg_reg$residuals

egr_ADF_lev <- ADF_estimate_lev(eg_res, p_max = 15)
print(egr_ADF_lev$ic_aic)
```

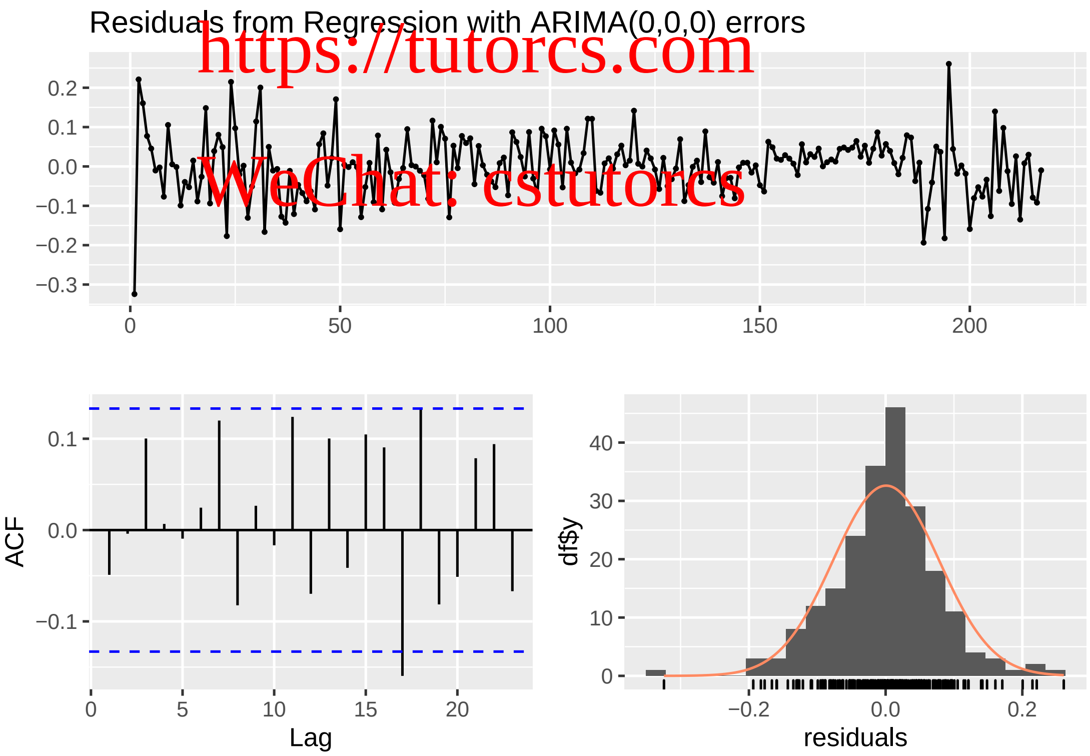
```
##       const trend  p       aic       bic
##  [1,]     0     0  0 -490.4155 -483.6557
##  [2,]     0     0  1 -490.1551 -480.0154
##  [3,]     0     0  2 -489.8021 -476.2825
##  [4,]     0     0  3 -488.4683 -471.5688
##  [5,]     1     0  0 -488.4332 -478.2935
##  [6,]     1     0  1 -488.1889 -474.6693
##  [7,]     1     0  2 -487.8534 -470.9539
##  [8,]     0     0 15 -486.5836 -429.1253
##  [9,]     1     0  3 -486.5075 -466.2281
## [10,]     0     0  4 -486.4786 -466.1992
```

```
print(egr_ADF_lev$ic_bic)
```

```
##       const trend p       aic       bic
```

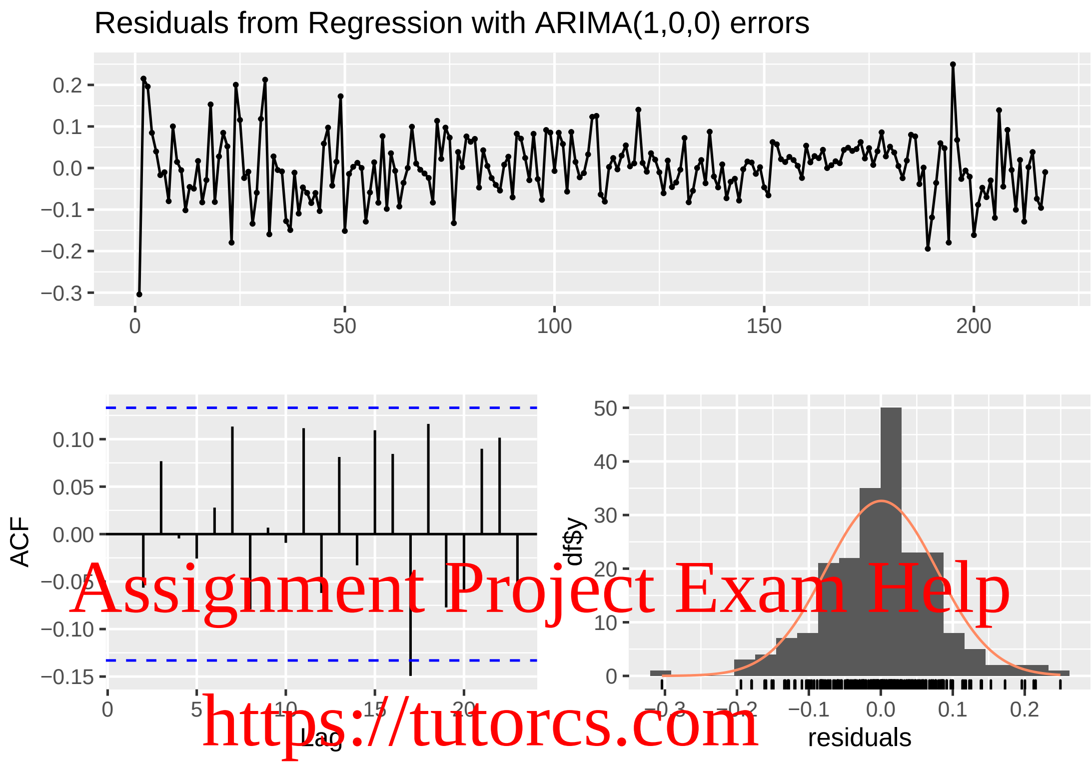
```
##   [1,]     0     0 0 -490.4155 -483.6557
##   [2,]     0     0 1 -490.1551 -480.0154
##   [3,]     1     0 0 -488.4332 -478.2935
##   [4,]     0     0 2 -489.8021 -476.2825
##   [5,]     1     0 1 -488.1889 -474.6693
##   [6,]     1     1 0 -486.4333 -472.9137
##   [7,]     0     0 3 -488.4683 -471.5688
##   [8,]     1     0 2 -487.8534 -470.9539
##   [9,]     1     1 1 -486.2086 -469.3091
##  [10,]     1     0 3 -486.5075 -466.2281
```

```
egr_adq_set <- as.matrix(arrange(as.data.frame(
                                   egr_ADF_lev$ic_bic),
                                   const, trend, p))
egr_adq_idx <- match(data.frame(t(egr_adq_set[, 1:3])),
                     data.frame(t(egr_ADF_lev$ic[, 1:3])))

for (i in 1:length(egr_adq_idx))
{
  checkresiduals(egr_ADF_lev$ADF_est[[egr_adq_idx[i]]])
}
```

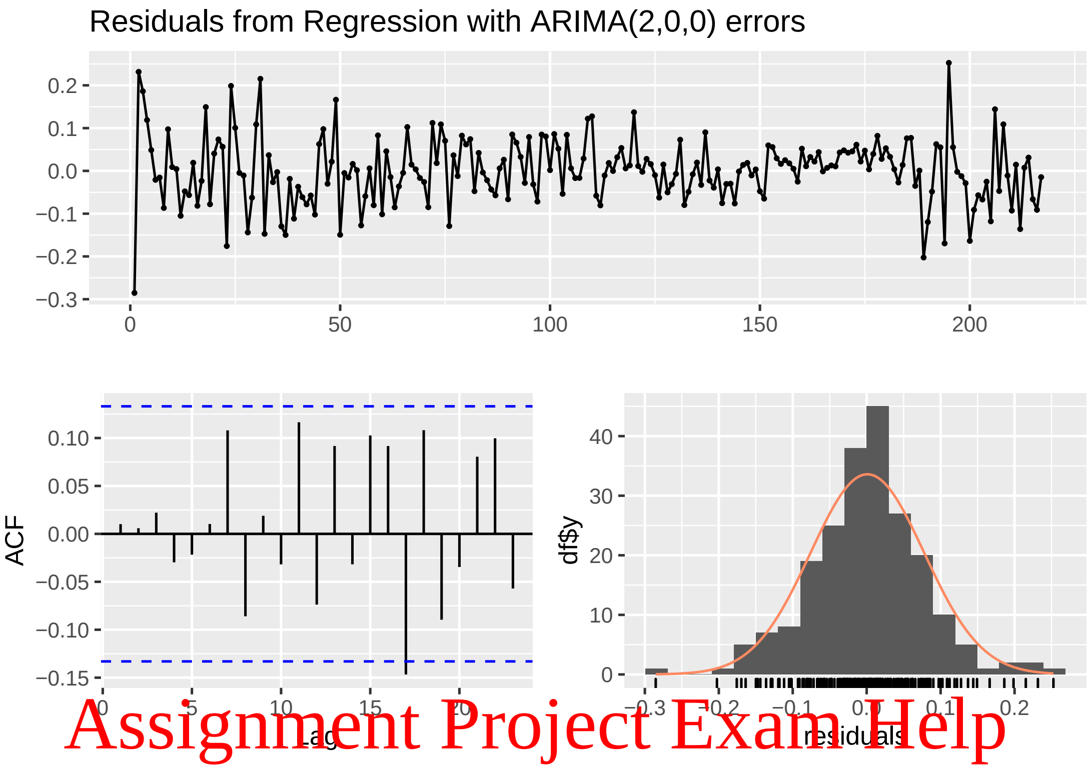




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
```

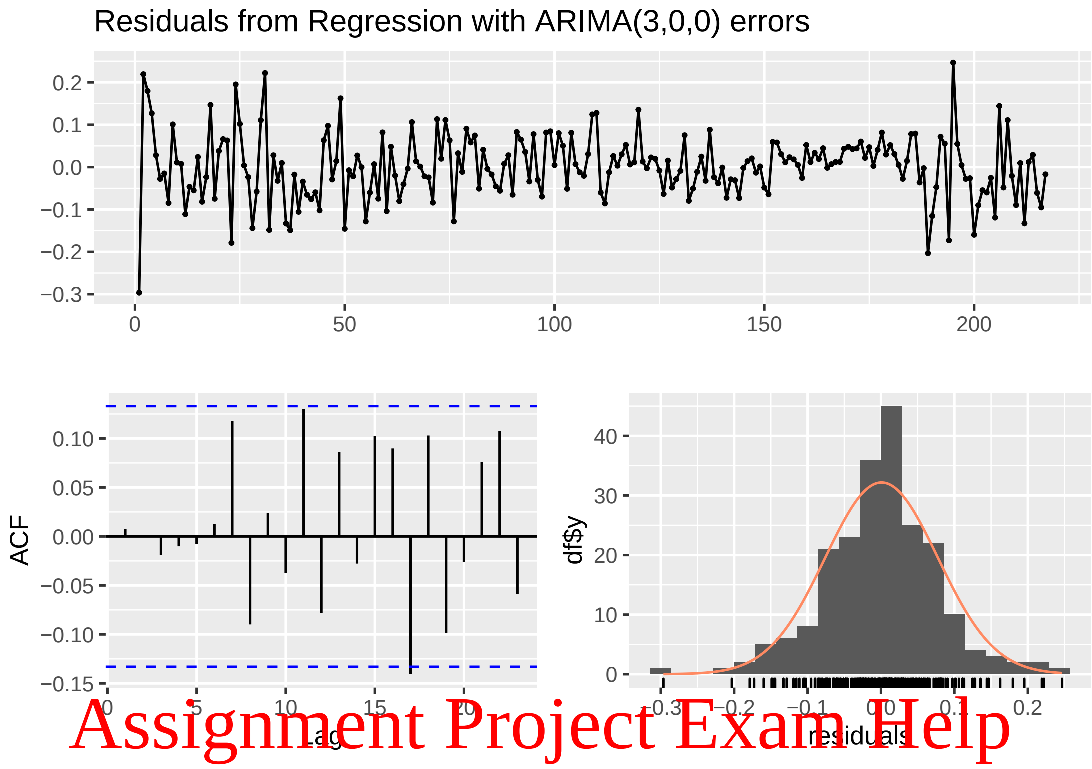
```
## Q* = 7.9494, df = 9, p-value = 0.5392
##
## Model df: 1.   Total lags used: 10
```



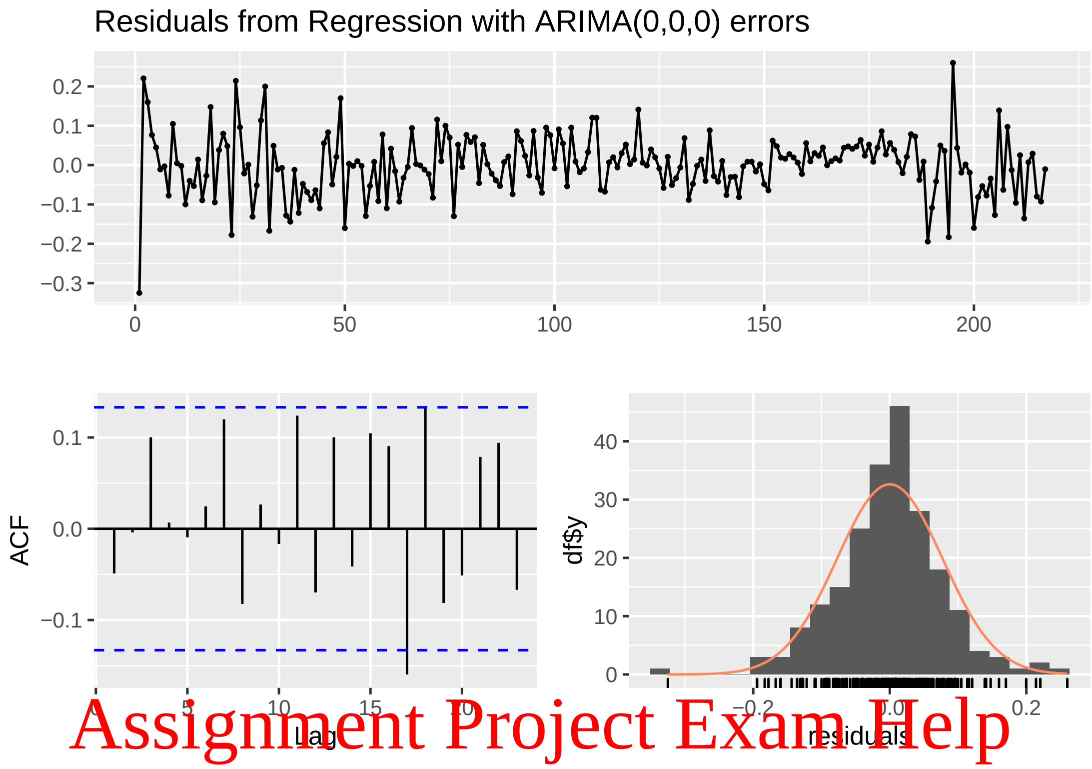

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 6.6997, df = 8, p-value = 0.5694
##
## Model df: 2.   Total lags used: 10
```



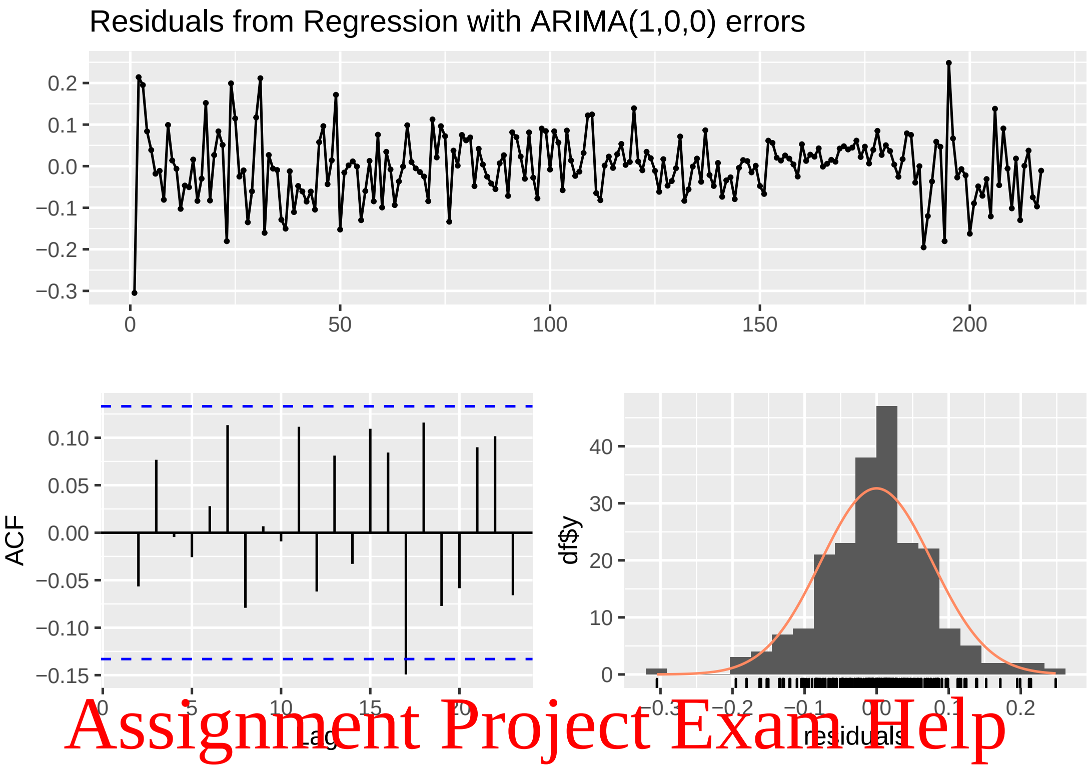

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 5.096, df = 7, p-value = 0.6483
##
## Model df: 3.   Total lags used: 10
```



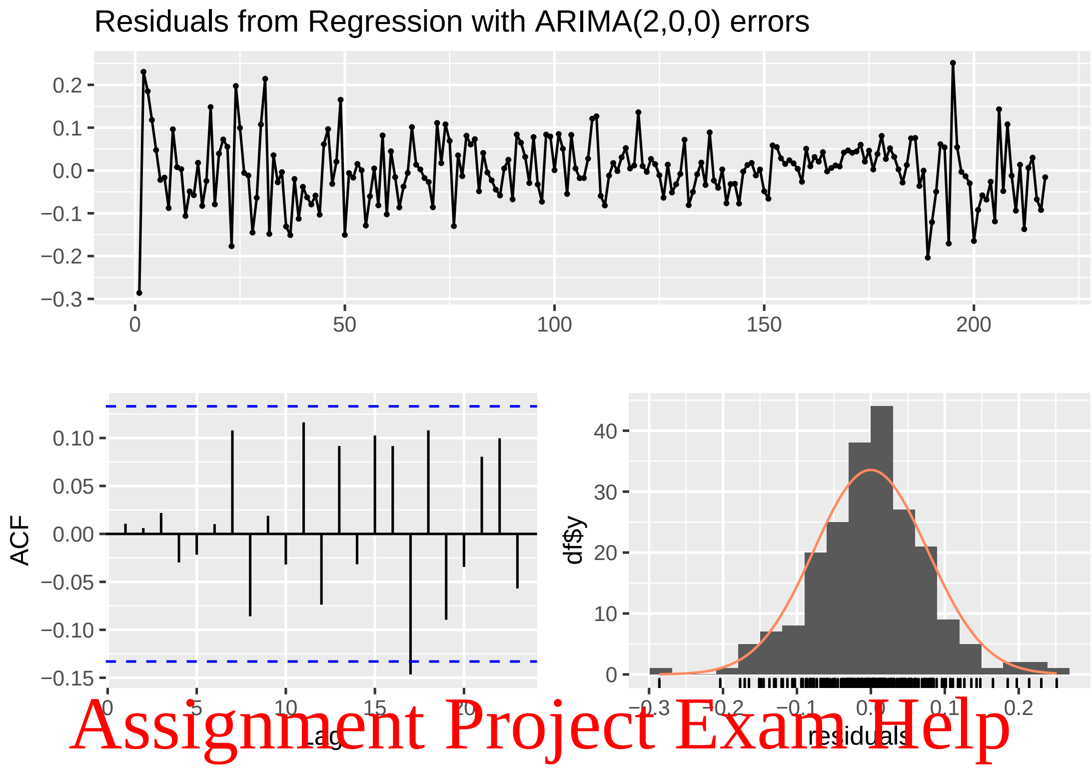

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 5.5928, df = 6, p-value = 0.4703
##
## Model df: 4.   Total lags used: 10
```



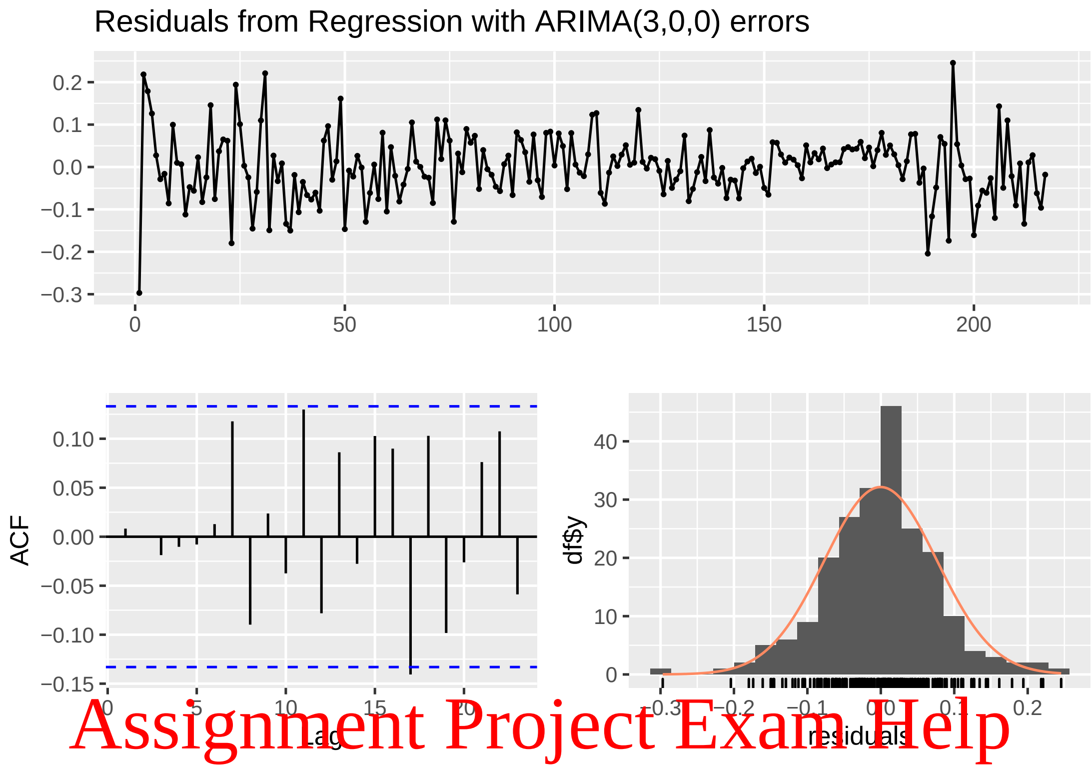

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 7.9489, df = 8, p-value = 0.4385
##
## Model df: 2.   Total lags used: 10
```



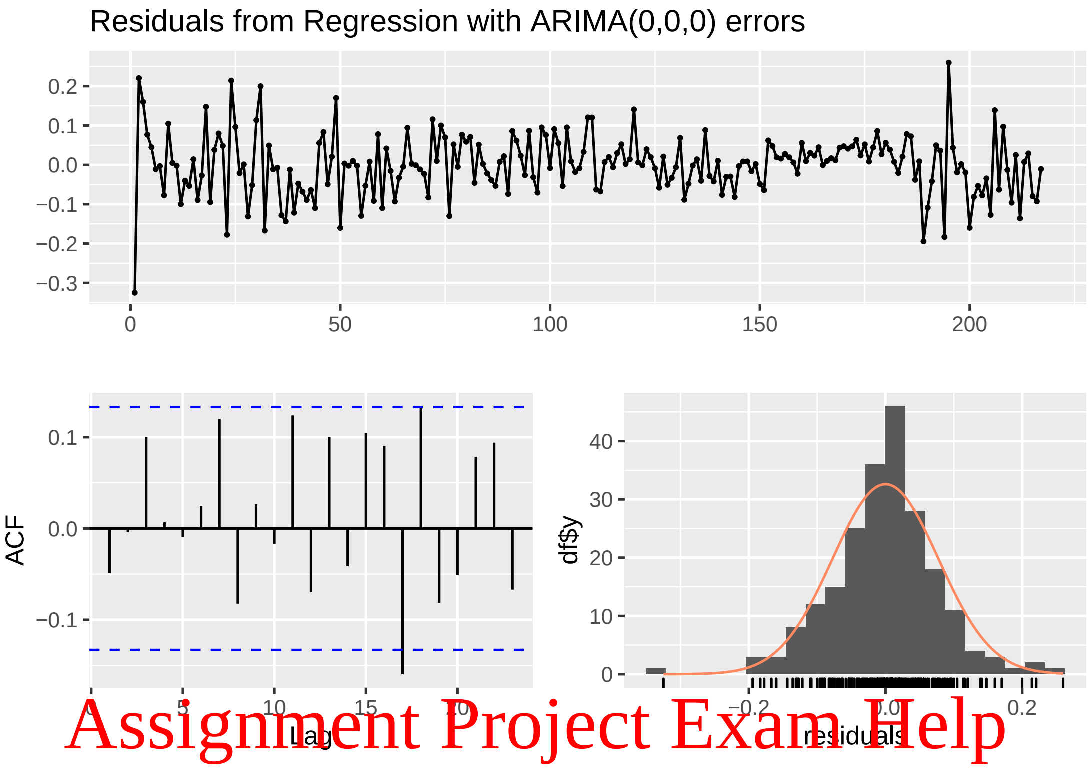

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 6.6992, df = 7, p-value = 0.4609
##
## Model df: 3.   Total lags used: 10
```



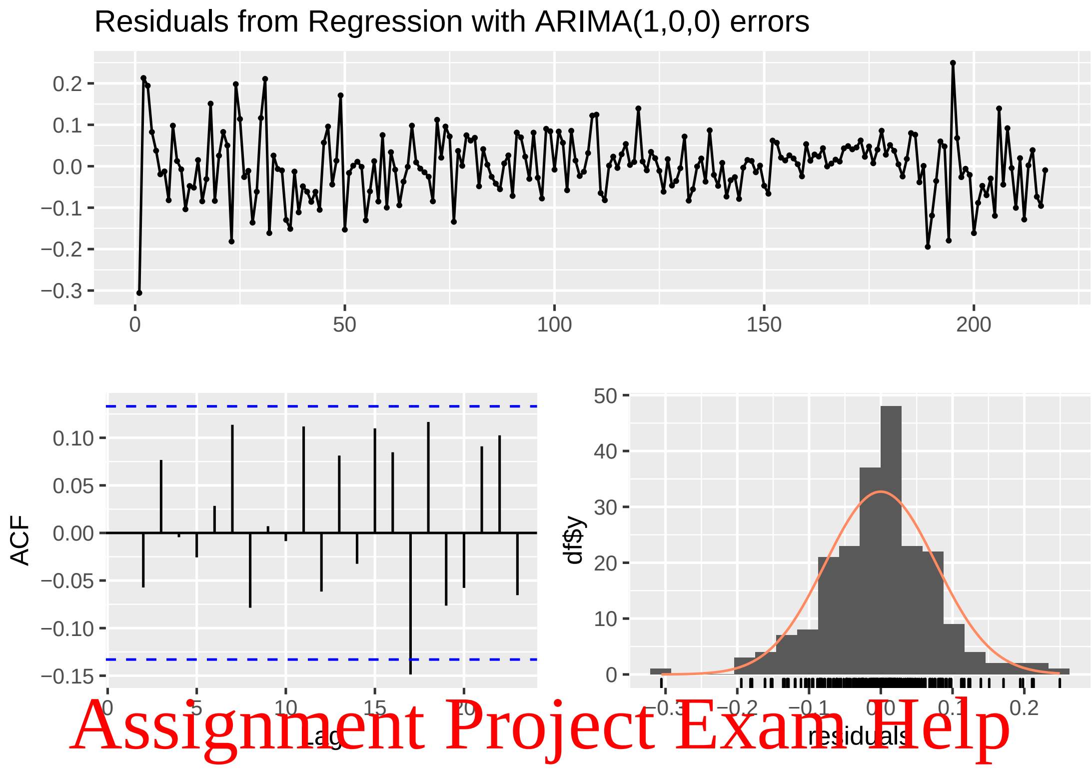

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 5.0944, df = 6, p-value = 0.5318
##
## Model df: 4.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 5.5849, df = 5, p-value = 0.3487
##
## Model df: 5.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 7.9458, df = 7, p-value = 0.3374
##
## Model df: 3.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 6.7221, df = 6, p-value = 0.3473
##
## Model df: 4.   Total lags used: 10
```



```
eg_test <- matrix(nrow = 3, ncol = 4)
colnames(eg_test) <- rep("", 4)
rownames(eg_test) <- c("No const, no trend",
                       "Const, no trend",
                       "Const with trend")
for (l in 1:4)
{
  eg_l <- coint.test(i5y, cbind(i3y, i90d, i180d),
                     nlag = l, output = F)
  eg_test[, l] <- eg_l[, 3]
  colnames(eg_test)[l] <- paste("Lag", l)
}
print(eg_test)
```

```
##                    Lag 1 Lag 2 Lag 3 Lag 4
## No const, no trend  0.01  0.01  0.01  0.01
## Const, no trend     0.10  0.10  0.10  0.10
## Const with trend    0.10  0.10  0.10  0.10
```

We obtain the same results as with i5y_t being the dependent variable. In fact, nothing of substance changes by setting i90d_t or i180d_t to be the dependent variables either. Changing the dependent variable does not materially affect our inference about cointegrating relations involving these four processes.

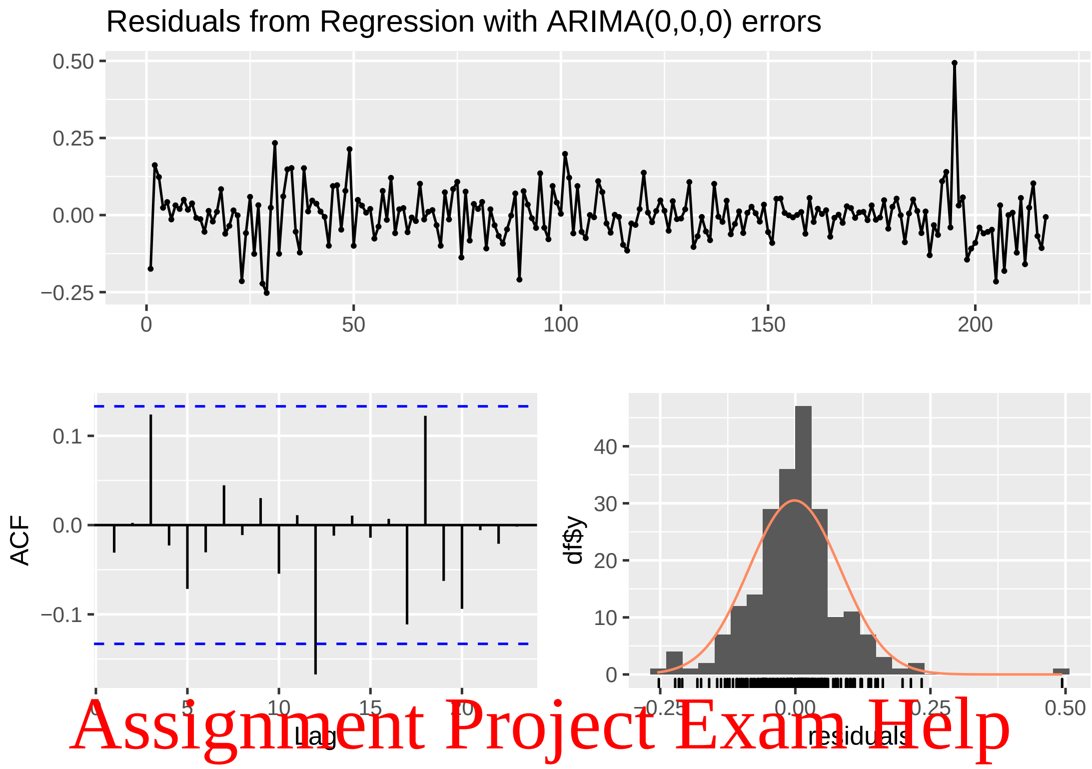
```
eg_reg <- lm( i90d ~ i3y + i5y + i180d, mydata)
eg_res <- eg_reg$residuals

egr_ADF_lev <- ADF_estimate_lev(eg_res, p_max = 15)
print(egr_ADF_lev$ic_aic)
```

```
##       const trend p       aic       bic
##  [1,]     0     0 0 -452.4299 -445.6701
##  [2,]     0     0 3 -450.8043 -433.9048
##  [3,]     0     0 1 -450.7693 -440.6296
##  [4,]     1     0 0 -450.4913 -440.3516
##  [5,]     1     1 5 -450.1861 -419.7670
##  [6,]     1     1 0 -450.0606 -436.5410
##  [7,]     1     0 3 -448.8389 -428.5595
##  [8,]     1     0 1 -448.8348 -435.3152
##  [9,]     0     0 2 -448.8249 -435.3053
## [10,]     0     0 4 -448.8044 -428.5250
```

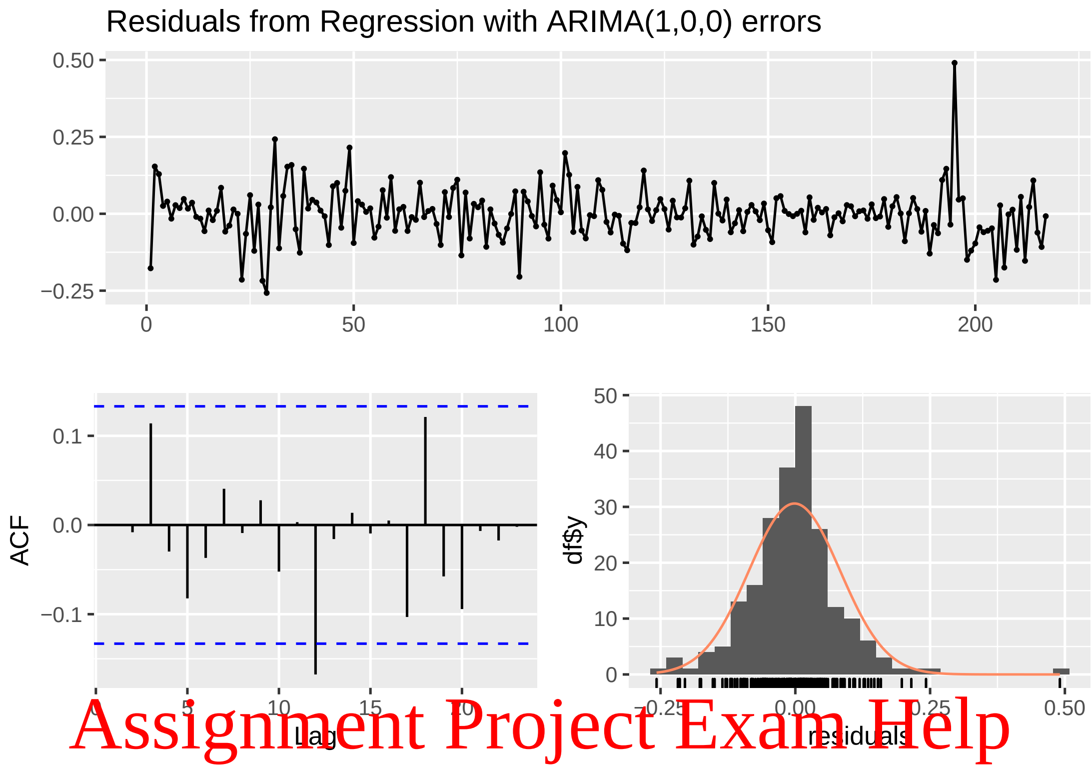
```
print(egr_ADF_lev$ic_bic)
```

```
##       const trend p       aic       bic
##  [1,]     0     0 0 -452.4299 -445.6701
##  [2,]     0     0 1 -450.7693 -440.6296
##  [3,]     1     0 0 -450.4913 -440.3516
##  [4,]     1     1 0 -450.0606 -436.5410
##  [5,]     1     0 1 -448.8348 -435.3152
##  [6,]     0     0 2 -448.8249 -435.3053
##  [7,]     0     0 3 -450.8043 -433.9048
##  [8,]     1     1 1 -448.3139 -431.4144
##  [9,]     1     0 2 -446.8931 -429.9936
## [10,]     1     0 3 -448.8389 -428.5595
```

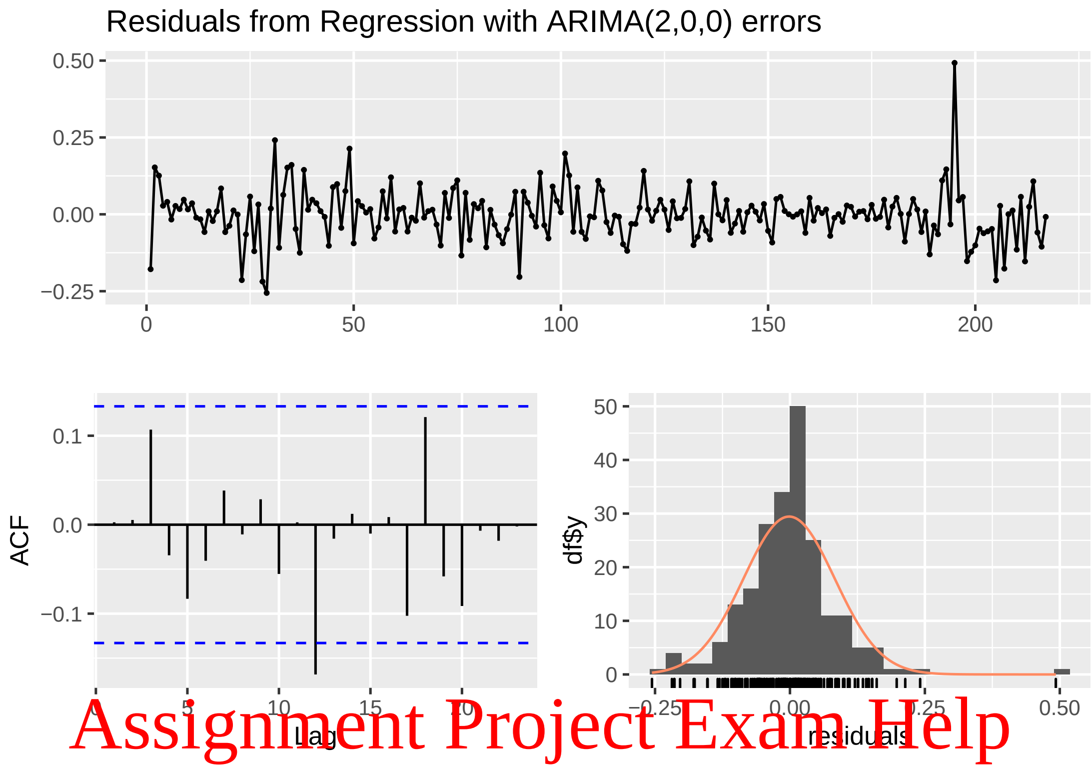
```
egr_adq_set <- as.matrix(arrange(as.data.frame(
  egr_ADF_lev$ic_bic),
  const, trend, p))
egr_adq_idx <- match(data.frame(t(egr_adq_set[, 1:3])),
                     data.frame(t(egr_ADF_lev$ic[, 1:3])))

for (i in 1:length(egr_adq_idx))
{
  checkresiduals(egr_ADF_lev$ADF_est[[egr_adq_idx[i]]])
}
```



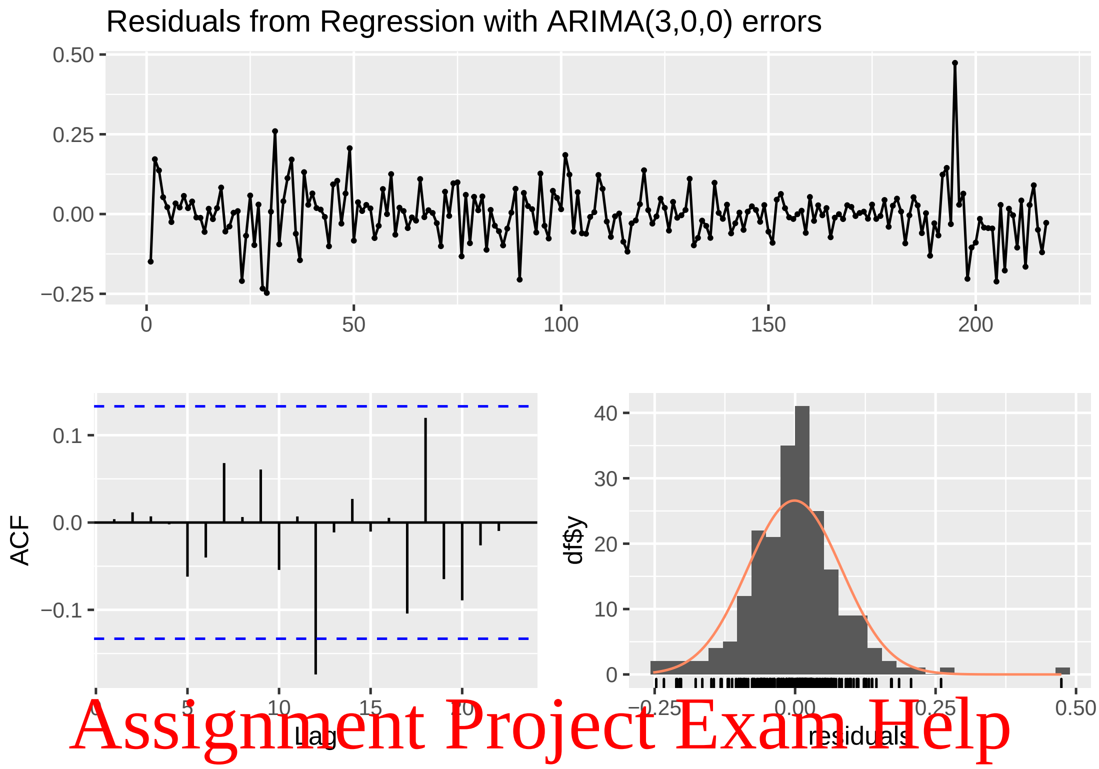

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 6.4587, df = 9, p-value = 0.6933
##
## Model df: 1.   Total lags used: 10
```



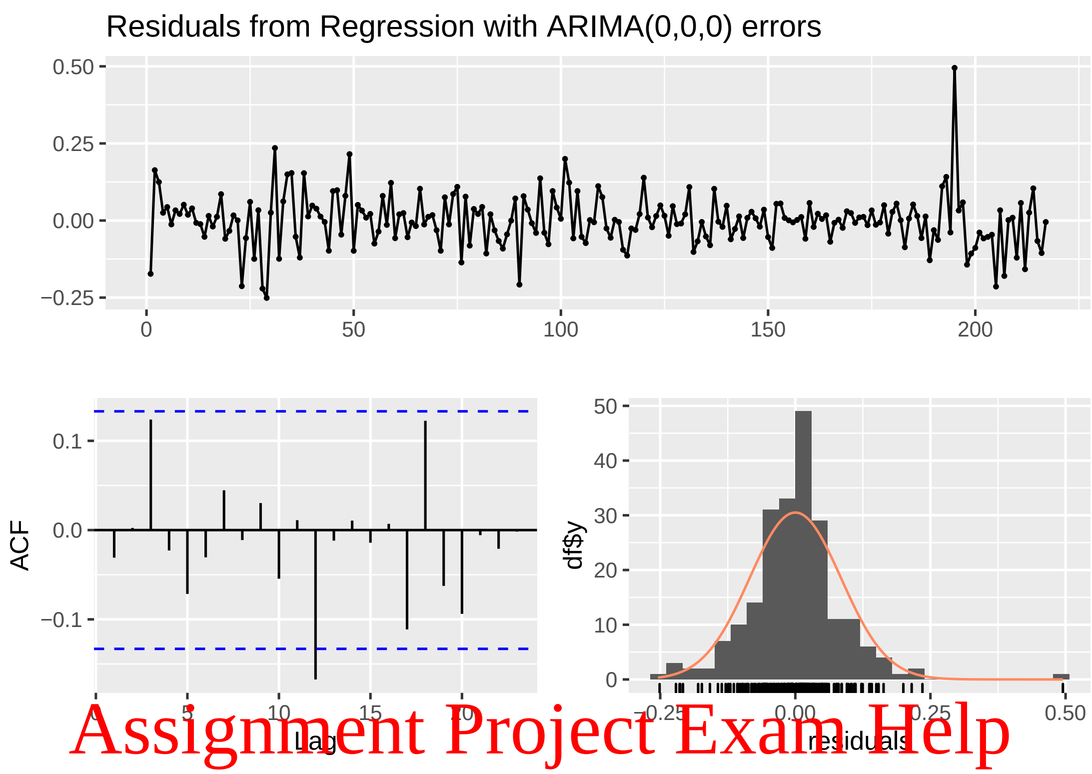

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 6.1121, df = 8, p-value = 0.6347
##
## Model df: 2.   Total lags used: 10
```



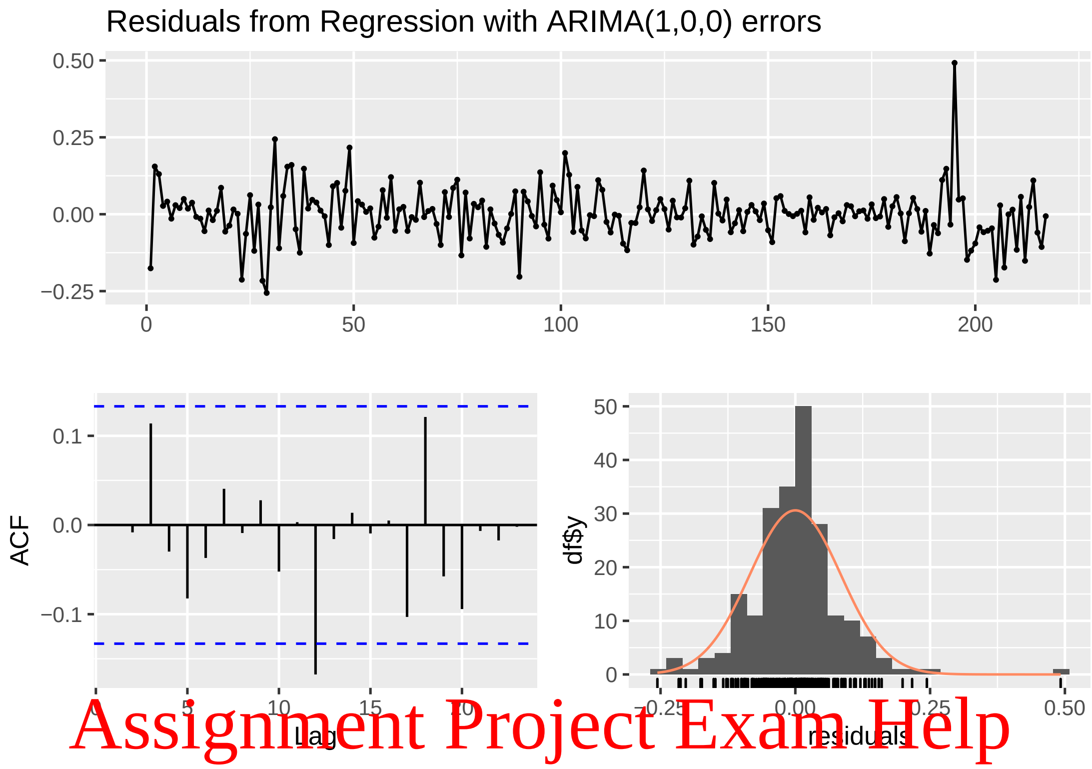

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 5.9864, df = 7, p-value = 0.5413
##
## Model df: 3.   Total lags used: 10
```



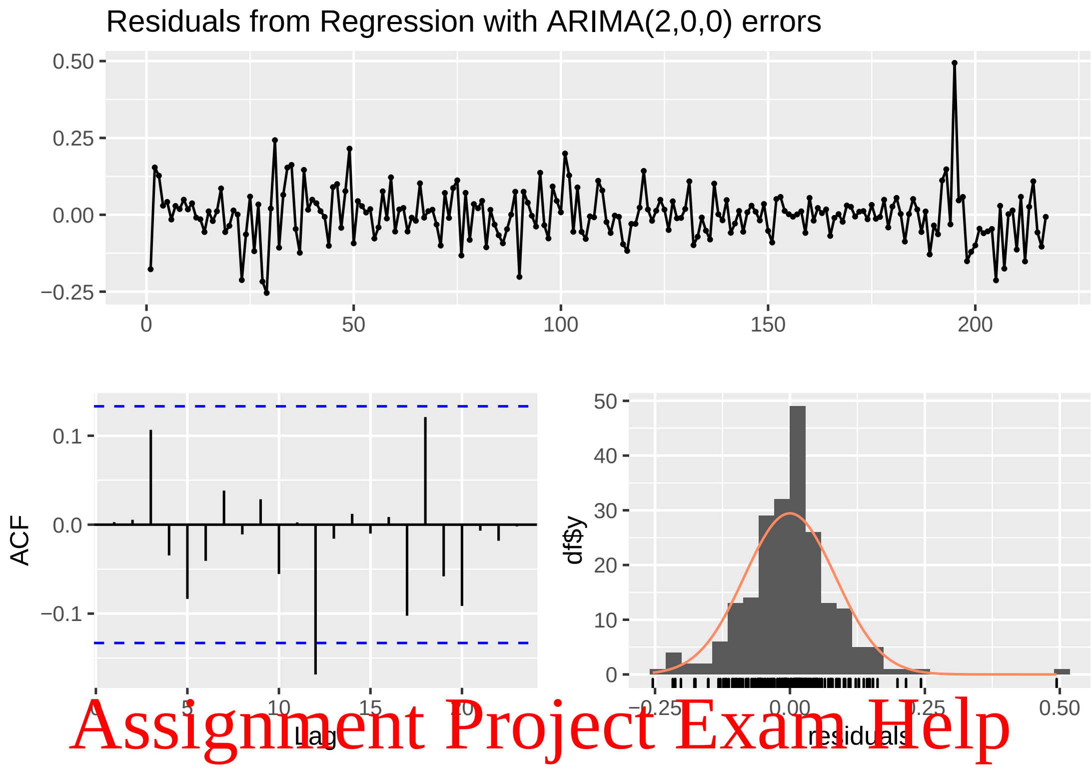

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 3.844, df = 6, p-value = 0.6978
##
## Model df: 4.   Total lags used: 10
```



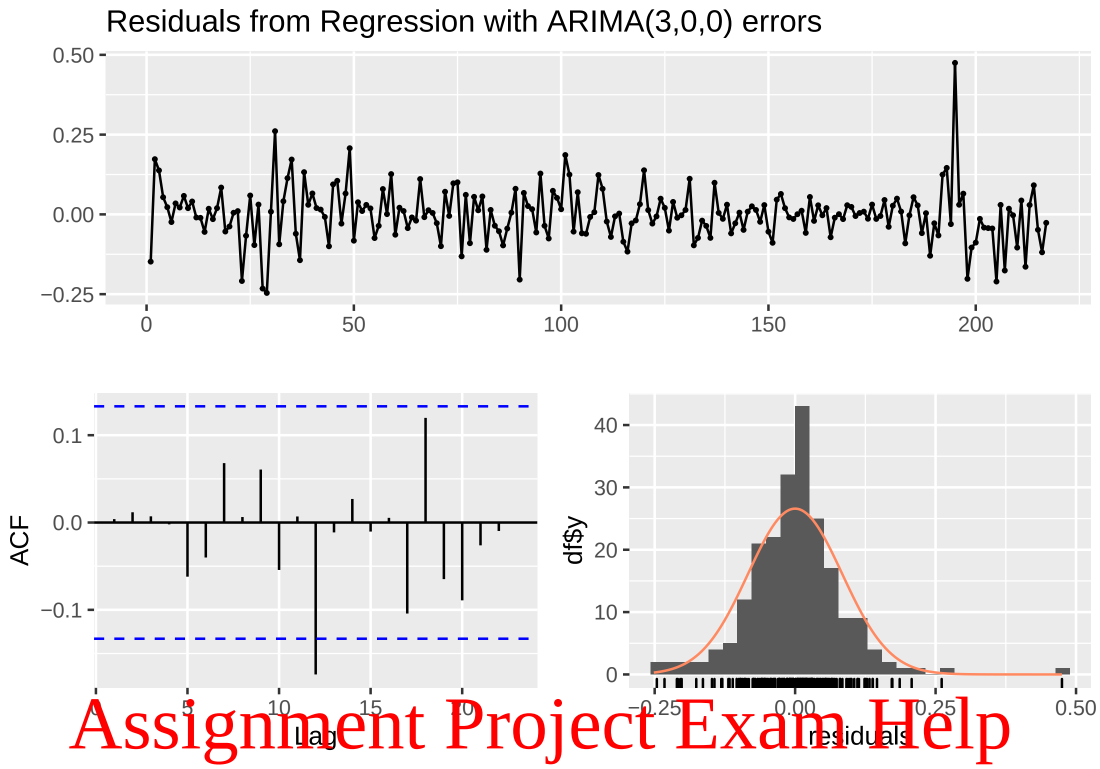

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 6.4591, df = 8, p-value = 0.596
##
## Model df: 2.   Total lags used: 10
```



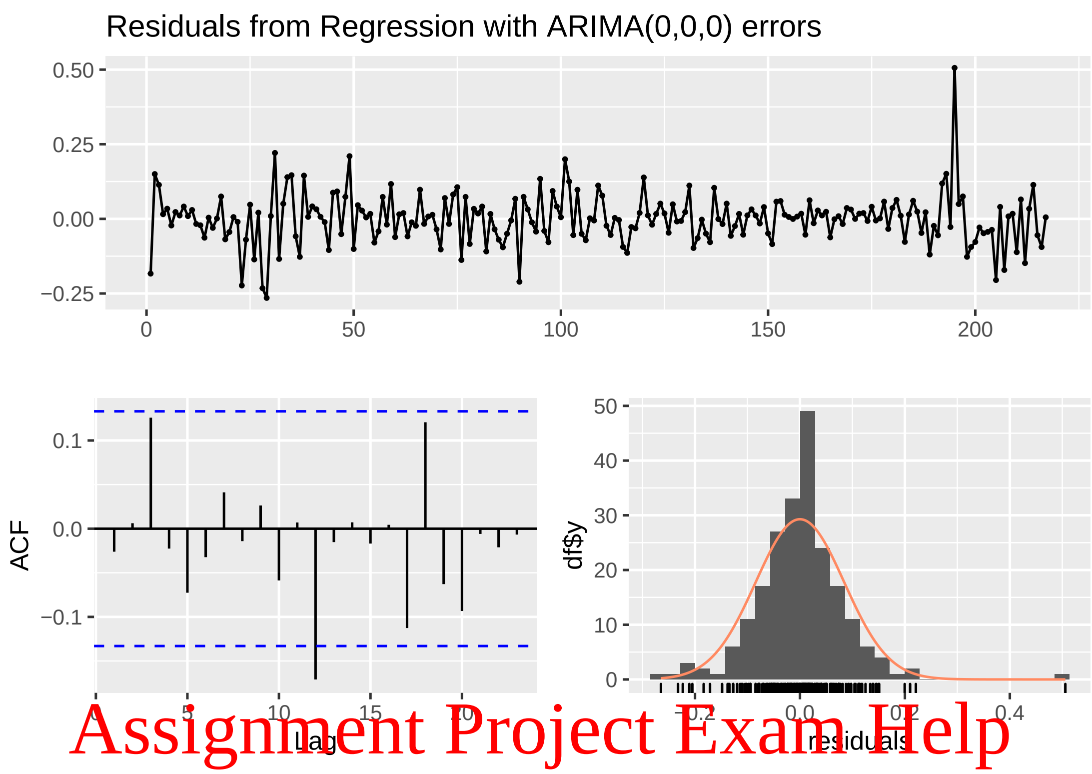

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 6.1118, df = 7, p-value = 0.5268
##
## Model df: 3.   Total lags used: 10
```



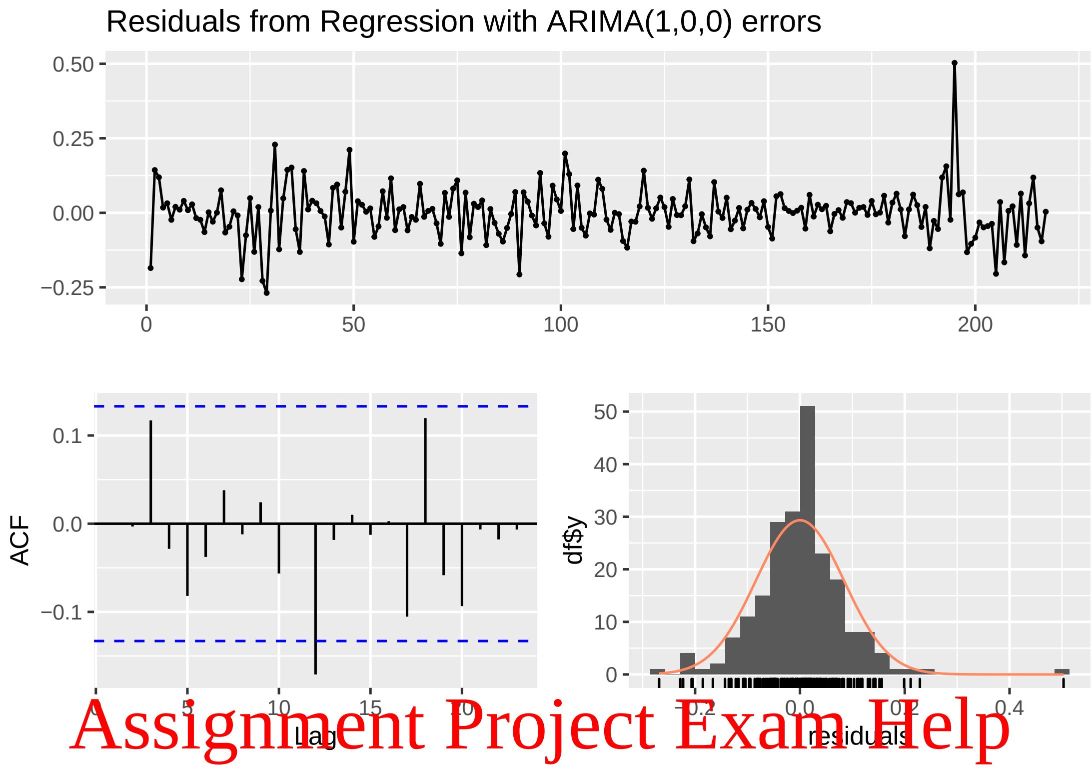

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 5.9848, df = 6, p-value = 0.4249
##
## Model df: 4.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 3.8443, df = 5, p-value = 0.572
##
## Model df: 5.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 6.5831, df = 7, p-value = 0.4735
##
## Model df: 3.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 6.2816, df = 6, p-value = 0.3924
##
## Model df: 4.   Total lags used: 10
```



```r
eg_test <- matrix(nrow = 3, ncol = 4)
colnames(eg_test) <- rep("", 4)
rownames(eg_test) <- c("No const, no trend",
                       "Const, no trend",
                       "Const with trend")
for (l in 1:4)
{
  eg_l <- coint.test(i5y, cbind(i3y, i90d, i180d),
                     nlag = l, output = F)
  eg_test[, l] <- eg_l[, 3]
  colnames(eg_test)[l] <- paste("Lag", l)
}
print(eg_test)
```

```
##                    Lag 1 Lag 2 Lag 3 Lag 4
## No const, no trend  0.01  0.01  0.01  0.01
## Const, no trend     0.10  0.10  0.10  0.10
## Const with trend    0.10  0.10  0.10  0.10
```

```r
eg_reg <- lm( i180d ~ i3y + i5y + i90d, mydata)
eg_res <- eg_reg$residuals

egr_ADF_lev <- ADF_estimate_lev(eg_res, p_max = 15)
print(egr_ADF_lev$ic_aic)
```

```
##       const trend p       aic       bic
##  [1,]     0     0 5 -422.6708 -399.0115
##  [2,]     1     1 5 -421.8393 -391.4202
##  [3,]     1     0 5 -420.7122 -393.6730
##  [4,]     0     0 0 -418.8574 -412.0976
##  [5,]     0     0 3 -418.8418 -401.9423
##  [6,]     0     0 1 -418.7116 -408.5719
##  [7,]     1     1 0 -418.3307 -404.8111
##  [8,]     1     1 3 -418.0234 -394.3642
##  [9,]     1     1 1 -417.8057 -400.9062
## [10,]     0     0 2 -417.7150 -404.1954
```

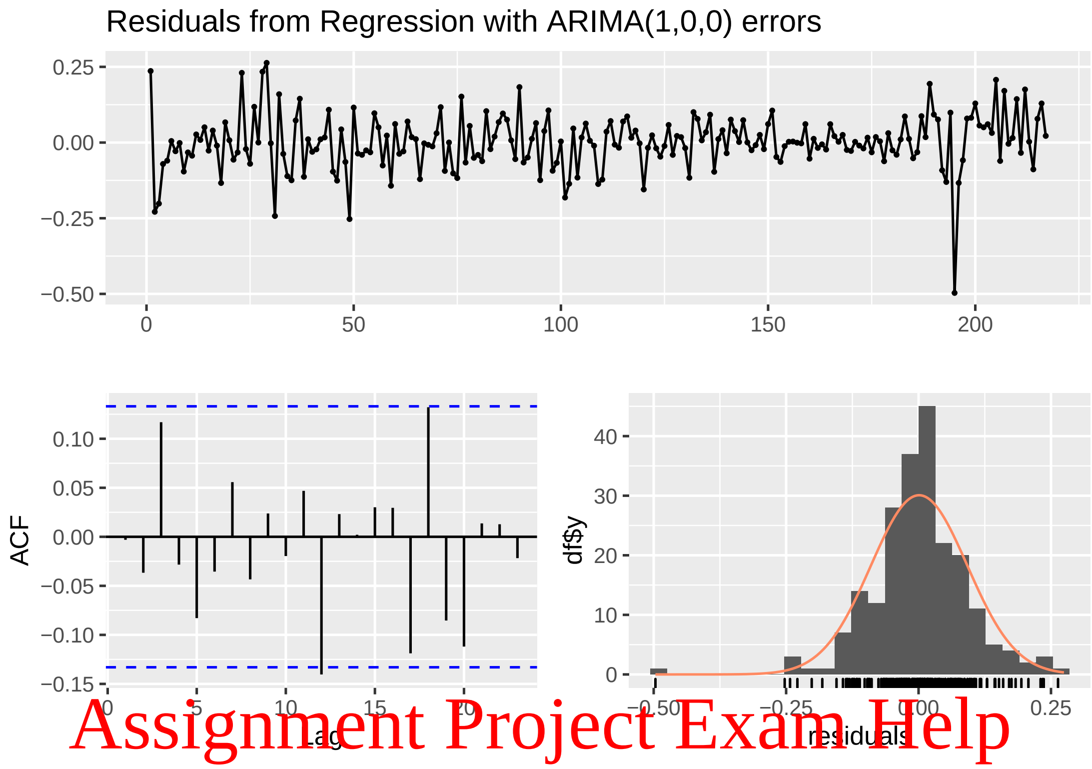
```r
print(egr_ADF_lev$ic_bic)
```



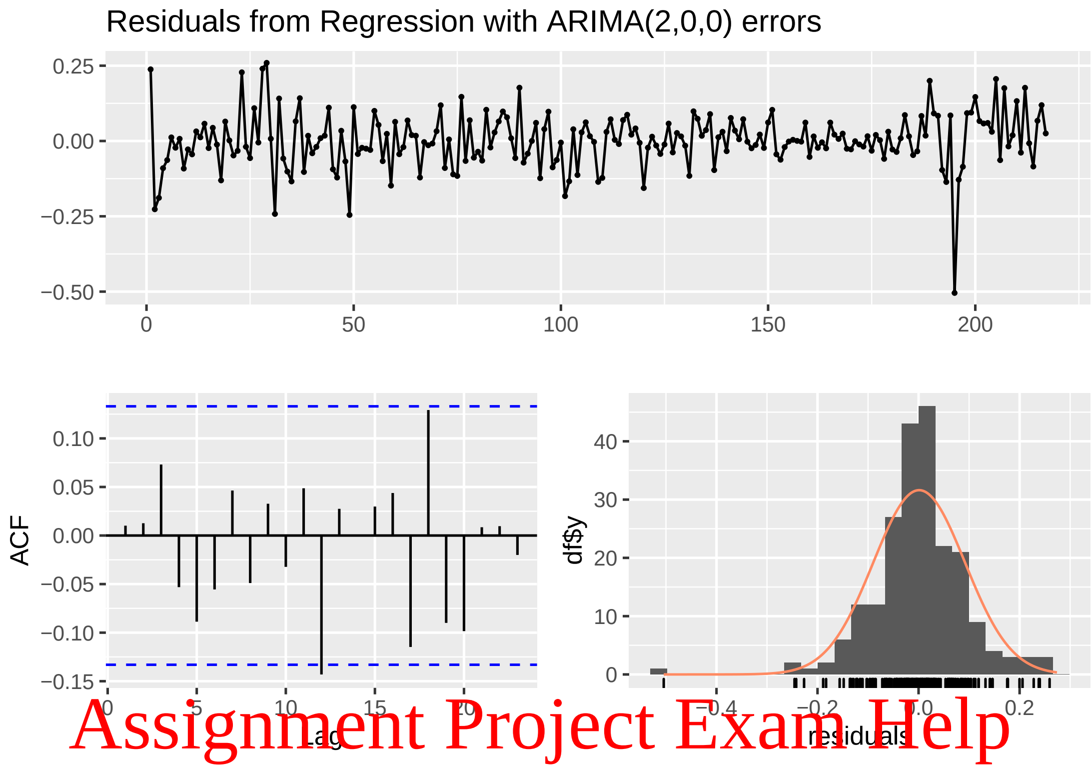
```
##       const trend p       aic       bic
##  [1,]     0     0 0 -418.8574 -412.0976
##  [2,]     0     0 1 -418.7116 -408.5719
##  [3,]     1     0 0 -416.8772 -406.7375
##  [4,]     1     1 0 -418.3307 -404.8111
##  [5,]     0     0 2 -417.7150 -404.1954
##  [6,]     1     0 1 -416.7327 -403.2131
##  [7,]     0     0 3 -418.8418 -401.9423
##  [8,]     1     1 1 -417.8057 -400.9062
##  [9,]     0     0 5 -422.6708 -399.0115
## [10,]     1     0 2 -415.7406 -398.8411
```

```r
egr_adq_set <- as.matrix(arrange(as.data.frame(
  egr_ADF_lev$ic_bic),
  const, trend, p))
egr_adq_idx <- match(data.frame(t(egr_adq_set[, 1:3])),
                     data.frame(t(egr_ADF_lev$ic[, 1:3])))

for (i in 1:length(egr_adq_idx))
{
  checkresiduals(egr_ADF_lev$ADF_est[[egr_adq_idx[i]]])
}
```

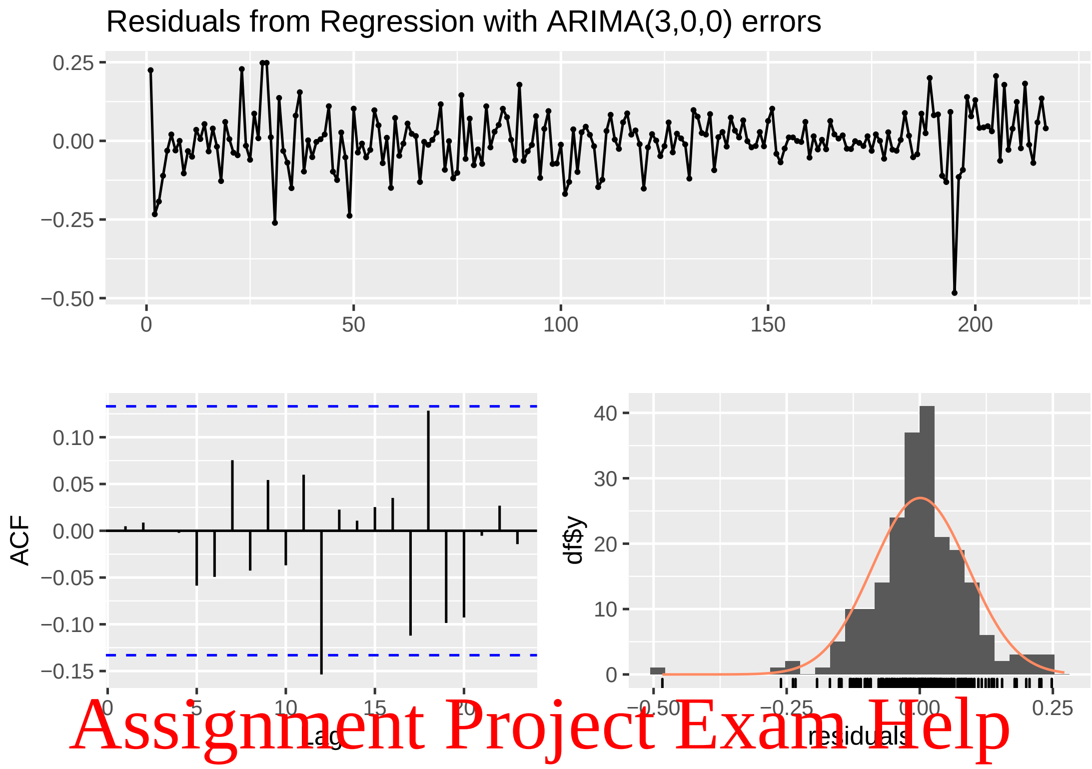

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 8.1843, df = 9, p-value = 0.5157
##
## Model df: 1.   Total lags used: 10
```



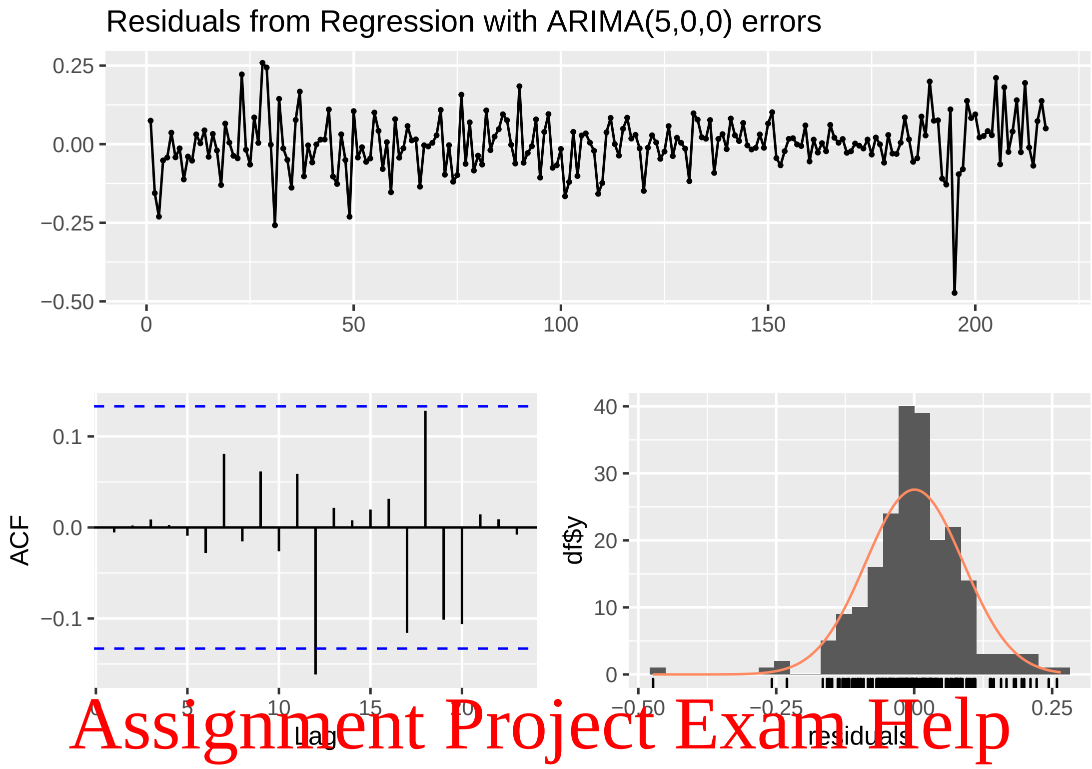

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 6.6843, df = 8, p-value = 0.571
##
## Model df: 2.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 5.8484, df = 7, p-value = 0.5576
##
## Model df: 3.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 4.0301, df = 6, p-value = 0.6726
##
## Model df: 4.   Total lags used: 10
```

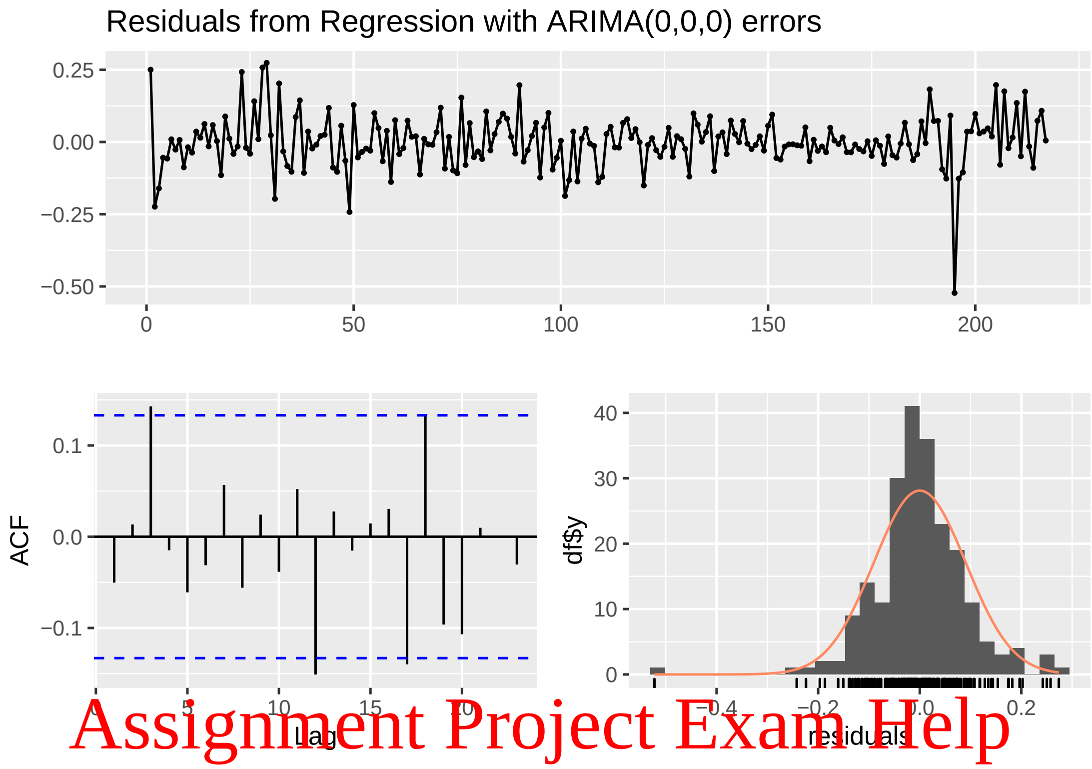



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(5,0,0) errors
## Q* = 2.776, df = 4, p-value = 0.596
##
## Model df: 6.   Total lags used: 10
```

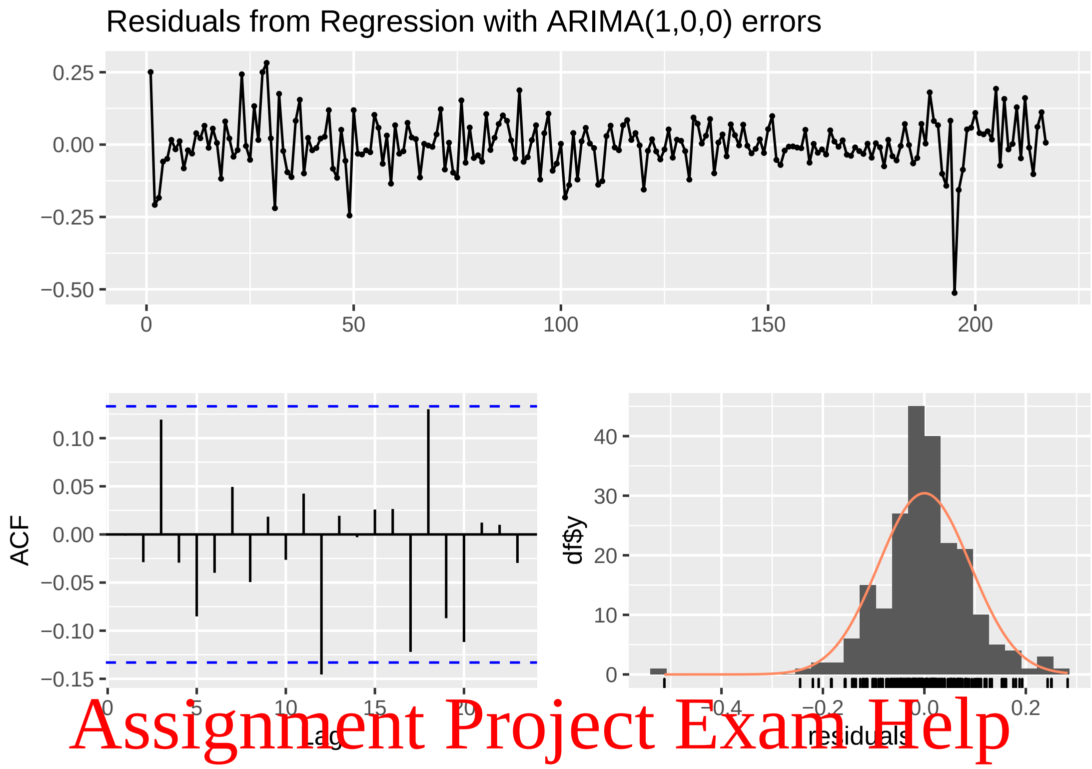

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 8.1849, df = 8, p-value = 0.4156
##
## Model df: 2.   Total lags used: 10
```


```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 6.6849, df = 7, p-value = 0.4624
##
## Model df: 3.   Total lags used: 10
```


```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 5.8492, df = 6, p-value = 0.4403
##
## Model df: 4.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 8.1437, df = 7, p-value = 0.3201
##
## Model df: 3.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 6.8607, df = 6, p-value = 0.3339
##
## Model df: 4.   Total lags used: 10
```

```
eg_test <- matrix(nrow = 3, ncol = 6)
colnames(eg_test) <- rep("", 6)
rownames(eg_test) <- c("No const, no trend",
                       "Const, no trend",
                       "Const with trend")
for (l in 1:6)
{
  eg_l <- coint.test(i5y, cbind(i3y, i90d, i180d),
                     nlag = l, output = F)
  eg_test[, l] <- eg_l[, 3]
  colnames(eg_test)[l] <- paste("Lag", l)
}
print(eg_test)
```

```
##                    Lag 1 Lag 2 Lag 3 Lag 4 Lag 5      Lag 6
## No const, no trend  0.01  0.01  0.01  0.01  0.01 0.03626372
## Const, no trend     0.10  0.10  0.10  0.10  0.10 0.10000000
## Const with trend    0.10  0.10  0.10  0.10  0.10 0.10000000
```

5. Next, use the data to test the *expectations theory* of the term structure of interest rates (ETT). Specifically, investigate whether the spreads in the Capital Market (i5y − i3y) and Money Market (i180d − i90d) are stable (and therefore stationary assuming constant variances and auto-covariances).

Solution This is straightforward using the ADF testing approach explained in Question 1. We apply it once to the spread i5y − i3y representing the Capital Market and again to the spread i180d − i90d representing the Money Market. In both cases the spreads are observed samples so there is no special consideration needed to the distribution of the ADF test statistic.

Also note that we do not require any assumptions about unit roots in the DGPs of individual interest rates to draw conclusions about the stationarity of the spreads (such assumptions are only needed if we want to conclude "a stationary spread implies cointegrated interest rates").


```
cm_ADF_lev <- ADF_estimate_lev(i5y - i3y, p_max = 15)
print(cm_ADF_lev$ic_aic)
```



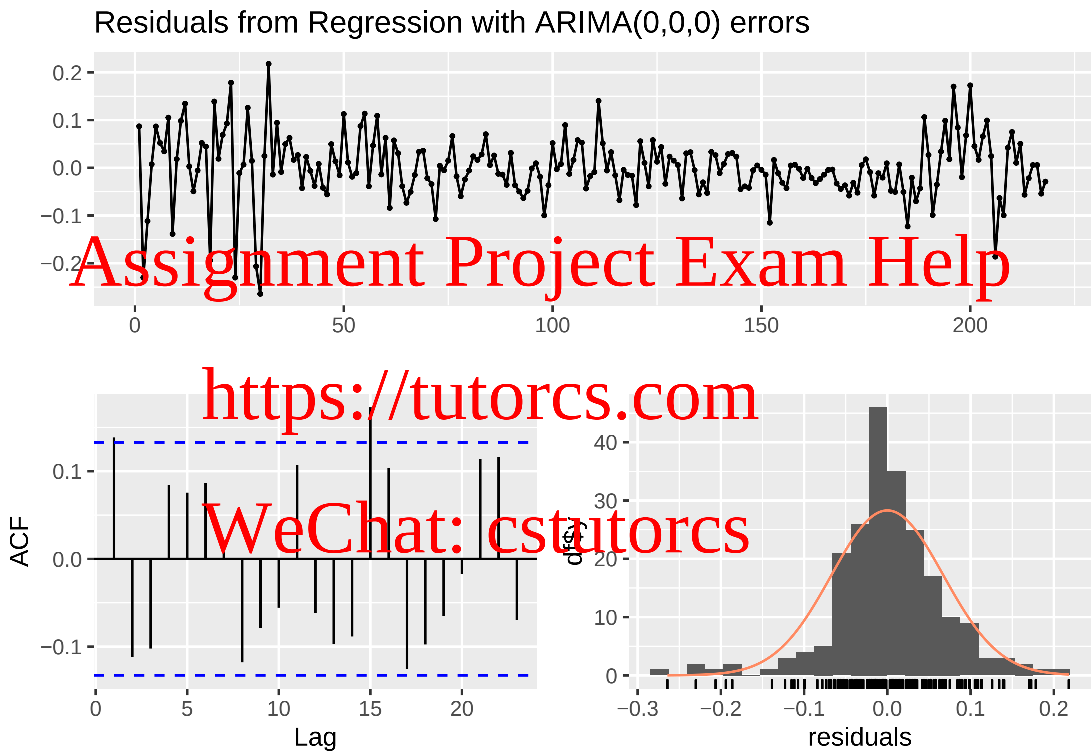
```
##       const trend  p       aic       bic
##  [1,]     1     0  2 -557.2660 -540.3435
##  [2,]     1     1  2 -556.7428 -536.4358
##  [3,]     1     0  4 -556.7363 -533.0449
##  [4,]     1     0  3 -556.6236 -536.3166
##  [5,]     1     1  4 -556.4973 -529.4214
##  [6,]     1     1  6 -556.1063 -522.2614
##  [7,]     1     1 15 -556.0468 -491.7414
##  [8,]     1     0 15 -555.8224 -494.9015
##  [9,]     1     1  1 -555.7064 -538.7839
## [10,]     1     1  3 -555.6926 -532.0012
```

```
print(cm_ADF_lev$ic_bic)
```

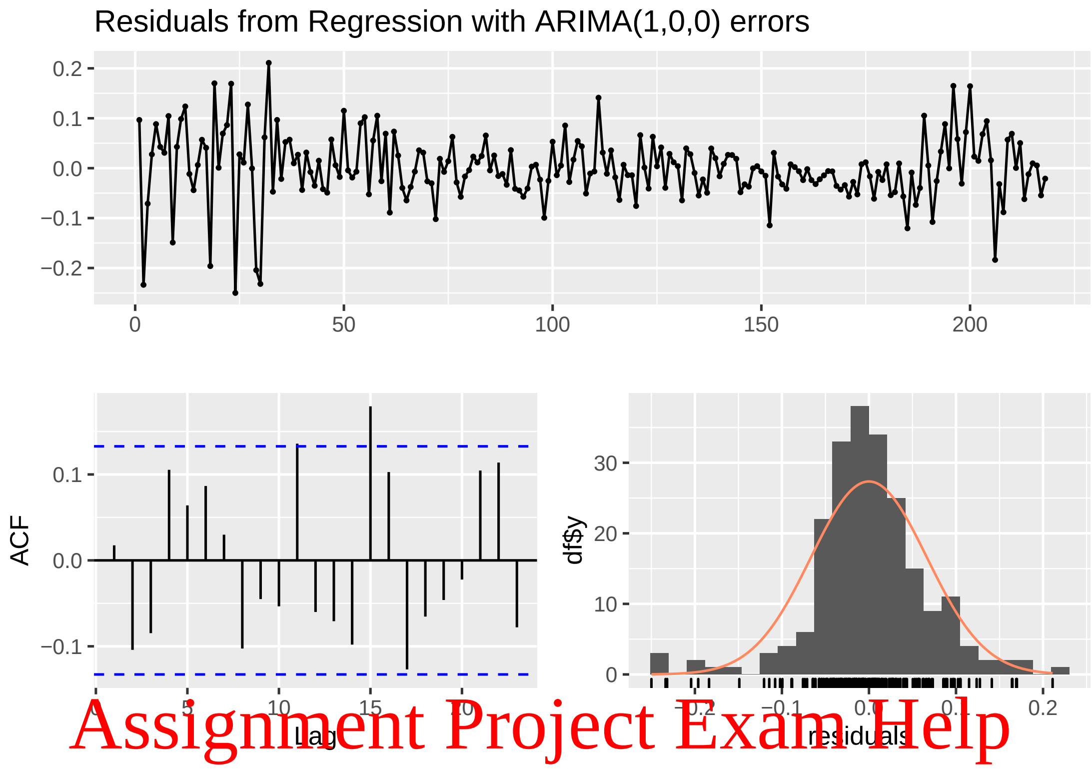
```
##       const trend p       aic       bic
##  [1,]     0     0 0 -550.2068 -543.4378
##  [2,]     1     0 0 -552.6518 -542.4983
##  [3,]     1     0 1 -555.3603 -541.8224
##  [4,]     0     0 1 -551.9226 -541.7691
##  [5,]     0     0 2 -555.0171 -541.4791
##  [6,]     1     0 2 -557.2660 -540.3435
##  [7,]     1     1 1 -555.7064 -538.7839
##  [8,]     1     1 0 -552.0080 -538.4700
##  [9,]     0     0 3 -555.0132 -538.0907
```

```
## [10,]      1      1 2 -556.7428 -536.4358
```

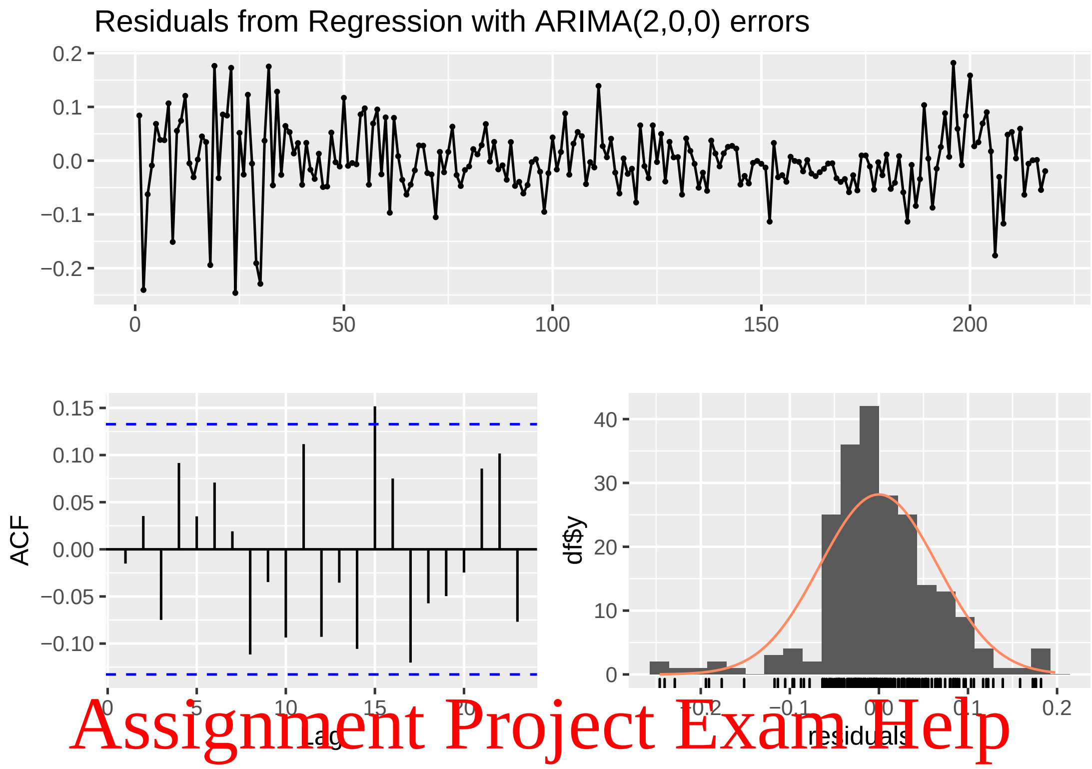
```
cm_adq_set <- as.matrix(arrange(as.data.frame(
                        cm_ADF_lev$ic_bic[c(2:3, 6:8, 10),]),
                        const, trend, p))
cm_adq_idx <- match(data.frame(t(cm_adq_set[, 1:3])),
                       data.frame(t(cm_ADF_lev$ic[, 1:3])))

for (i in 1:length(cm_adq_idx))
{
  checkresiduals(cm_ADF_lev$ADF_est[[cm_adq_idx[i]]])
}
```



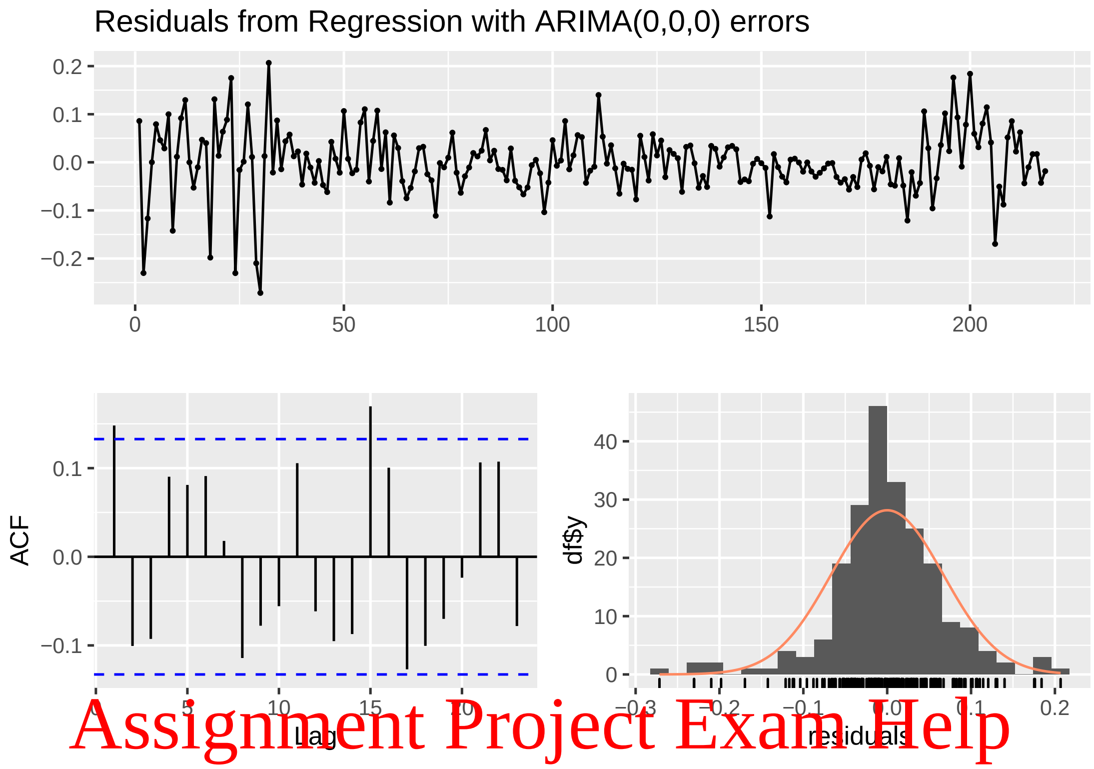

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 19.261, df = 8, p-value = 0.01353
##
## Model df: 2.   Total lags used: 10
```



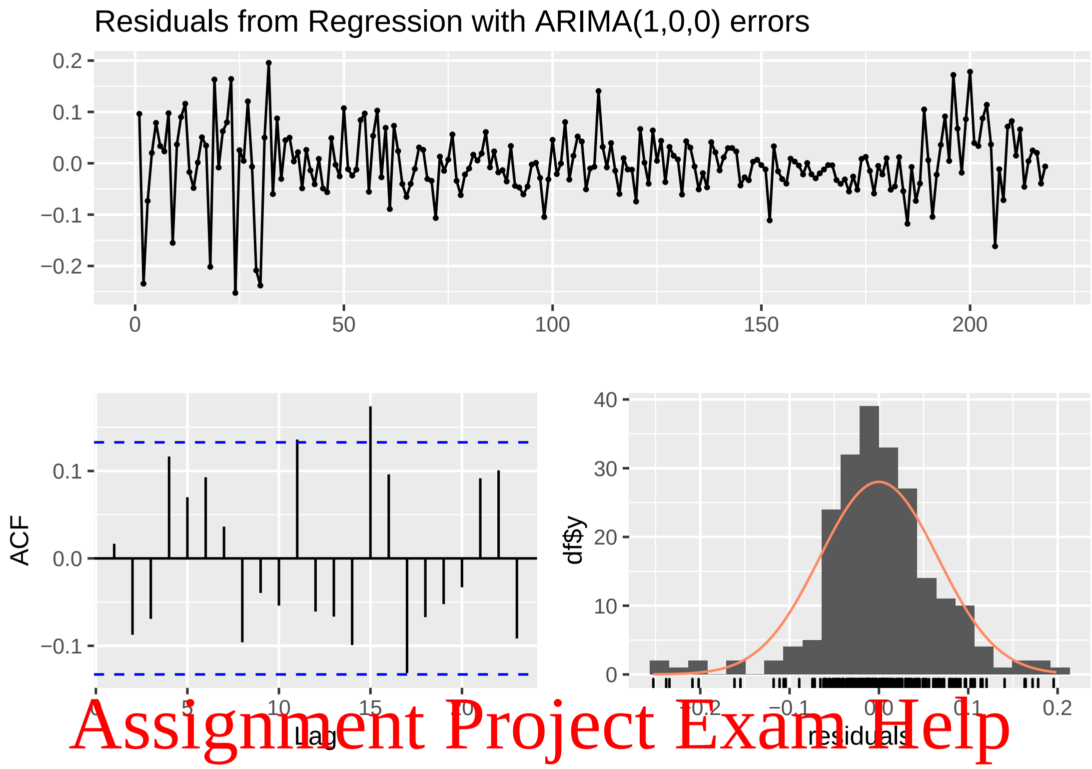

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 12.915, df = 7, p-value = 0.0742
##
## Model df: 3.   Total lags used: 10
```



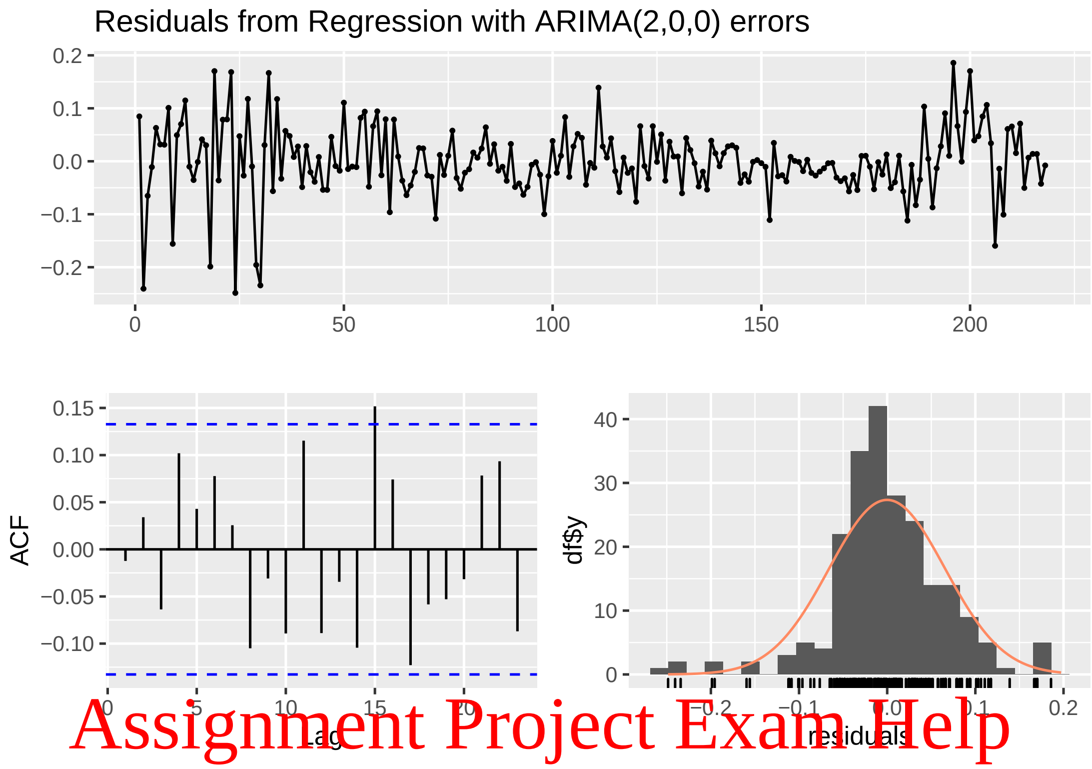

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 10.092, df = 6, p-value = 0.1209
##
## Model df: 4.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 19.346, df = 7, p-value = 0.007169
##
## Model df: 3.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 12.354, df = 6, p-value = 0.05452
##
## Model df: 4.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 10.031, df = 5, p-value = 0.07437
##
## Model df: 5.   Total lags used: 10
```

```
adf.test(i5y - i3y, nlag = 3)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag   ADF p.value
## [1,]   0 -2.72  0.0100
## [2,]   1 -2.99  0.0100
## [3,]   2 -2.13  0.0345
## Type 2: with drift no trend
##      lag   ADF p.value
## [1,]   0 -3.40  0.0128
## [2,]   1 -3.88  0.0100
## [3,]   2 -2.82  0.0611
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,]   0 -3.48   0.045
```

```
## [2,]   1 -4.09   0.010
## [3,]   2 -3.04   0.141
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

For i5y − i3y, the test rejects a unit root for models with $p = 1$, but fails to reject it for models with $p = 2$. We are unable to conclusively confirm the ETT in the Capital Market.

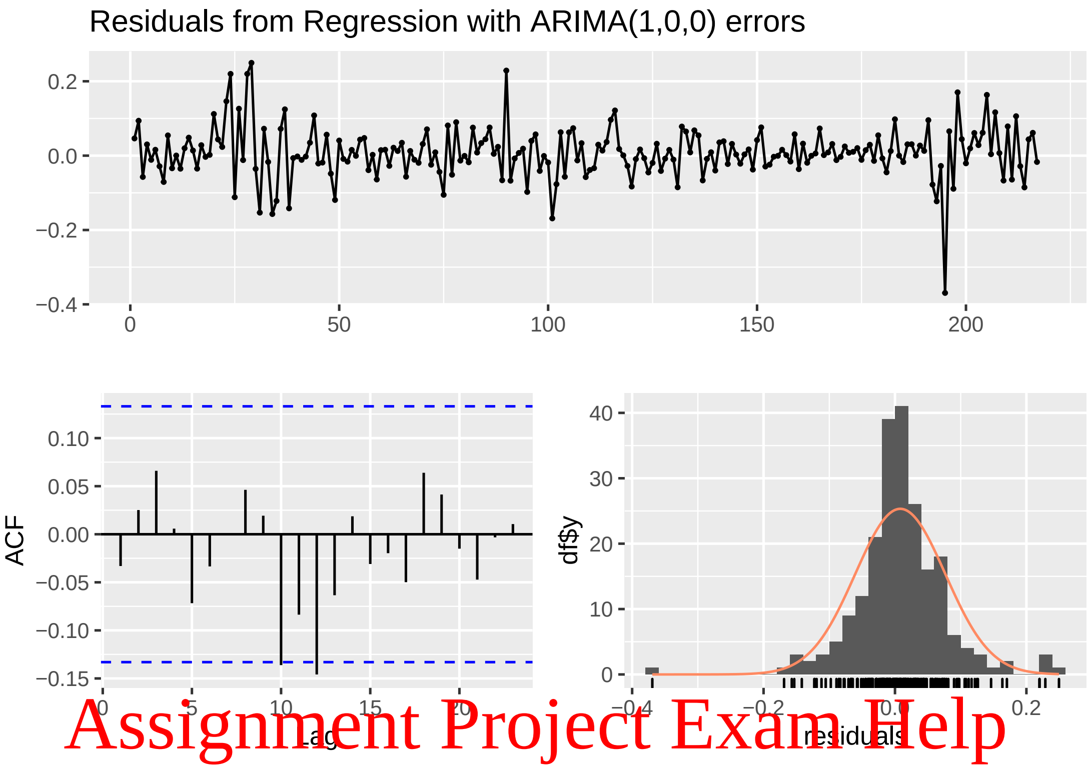
```r
mm_ADF_lev <- ADF_estimate_lev(i180d - i90d, p_max = 15)
print(mm_ADF_lev$ic_aic)
```

```
##       const trend p       aic       bic
##  [1,]     1     0 1 -540.8512 -527.3316
##  [2,]     1     0 2 -540.5695 -523.6700
##  [3,]     1     0 3 -539.8240 -519.5446
##  [4,]     0     0 1 -539.2831 -529.1434
##  [5,]     1     1 1 -539.0038 -522.1043
##  [6,]     1     1 2 -538.6916 -518.4123
##  [7,]     0     0 2 -538.5208 -525.0012
##  [8,]     0     0 3 -538.1661 -521.2667
##  [9,]     1     0 4 -538.0003 -514.3410
## [10,]     1     1 3 -537.9208 -514.2615
```

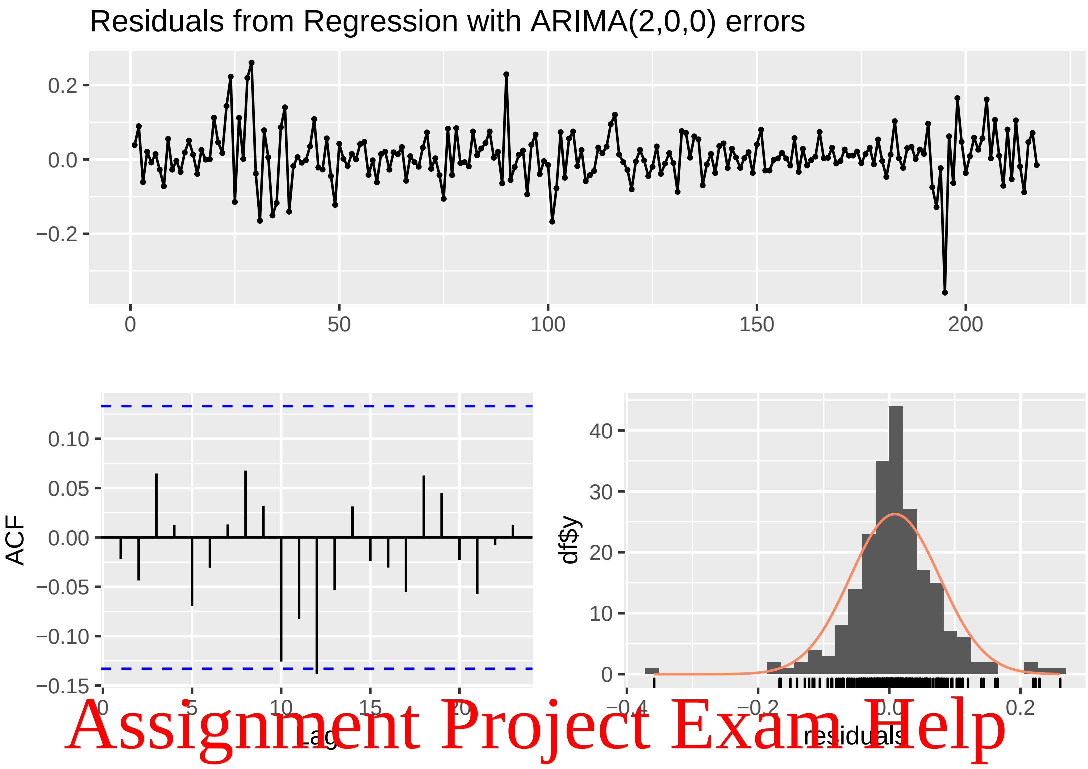
```r
print(mm_ADF_lev$ic_bic)
```

```
##       const trend p       aic       bic
##  [1,]     0     0 1 -539.2831 -529.1434
##  [2,]     1     0 1 -540.8512 -527.3316
##  [3,]     0     0 2 -538.5208 -525.0012
##  [4,]     1     0 2 -540.5695 -523.6700
##  [5,]     1     1 1 -539.0038 -522.1043
##  [6,]     0     0 3 -538.1661 -521.2667
##  [7,]     1     0 3 -539.8240 -519.5446
##  [8,]     1     1 2 -538.6916 -518.4123
##  [9,]     0     0 0 -524.7458 -517.9860
## [10,]     0     0 4 -536.1951 -515.9157
```

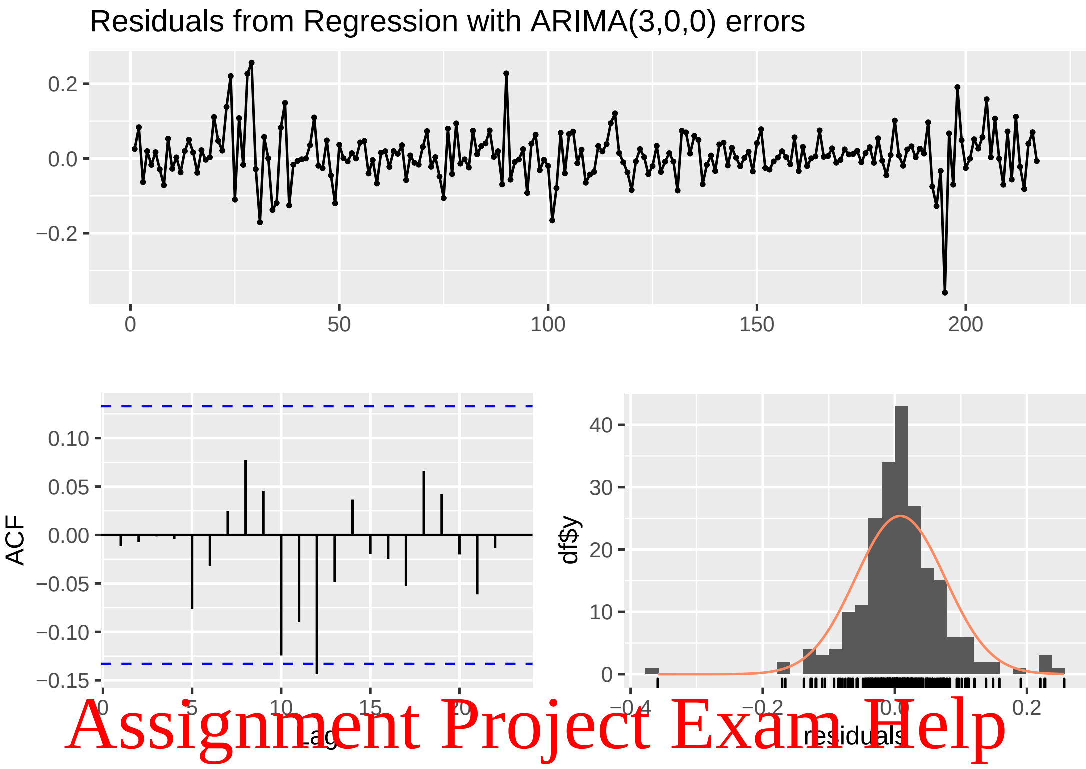
```r
mm_adq_set <- as.matrix(arrange(as.data.frame(
                        mm_ADF_lev$ic_bic[c(1:8),]),
                        const, trend, p))
mm_adq_idx <- match(data.frame(t(mm_adq_set[, 1:3])),
                    data.frame(t(mm_ADF_lev$ic[, 1:3])))

for (i in 1:length(mm_adq_idx))
{
  checkresiduals(mm_ADF_lev$ADF_est[[mm_adq_idx[i]]])
}
```



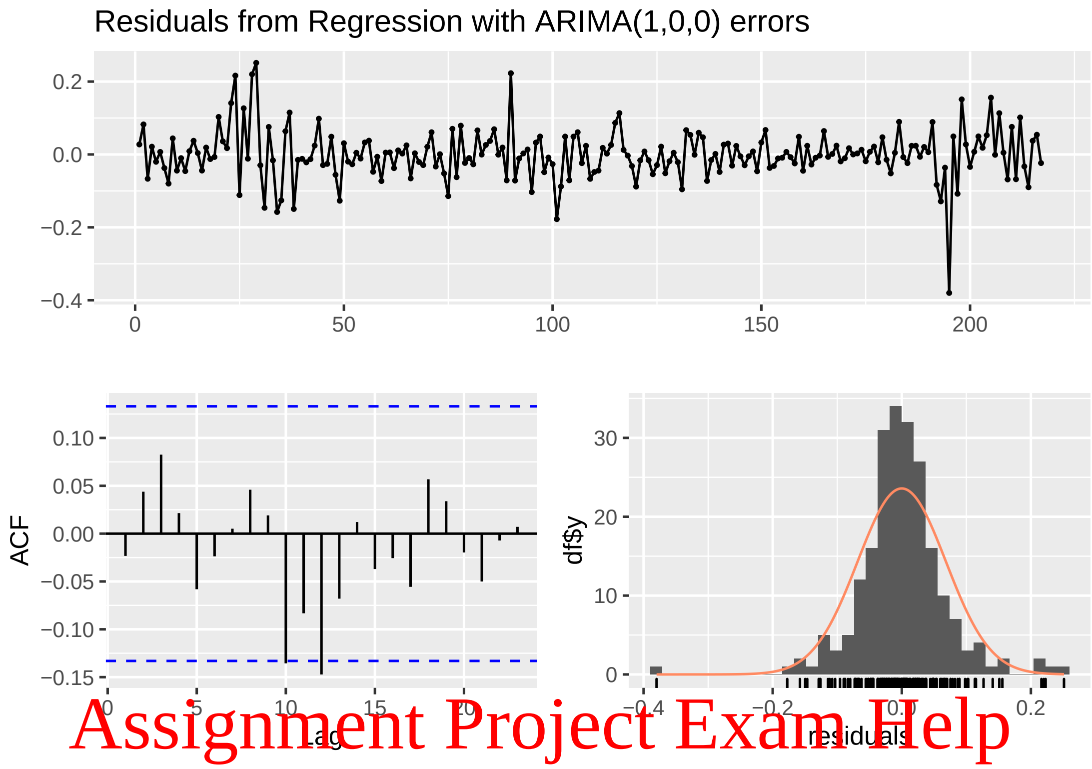

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 7.5858, df = 8, p-value = 0.4749
##
## Model df: 2.   Total lags used: 10
```



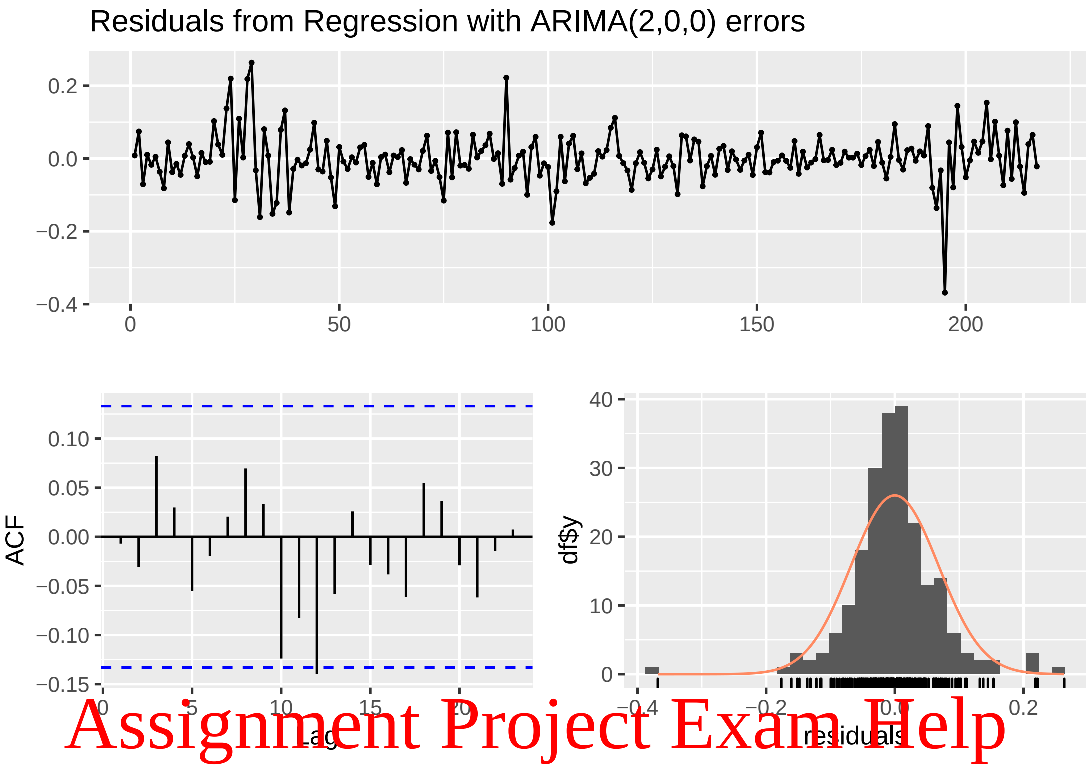

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 7.7336, df = 7, p-value = 0.3567
##
## Model df: 3.   Total lags used: 10
```



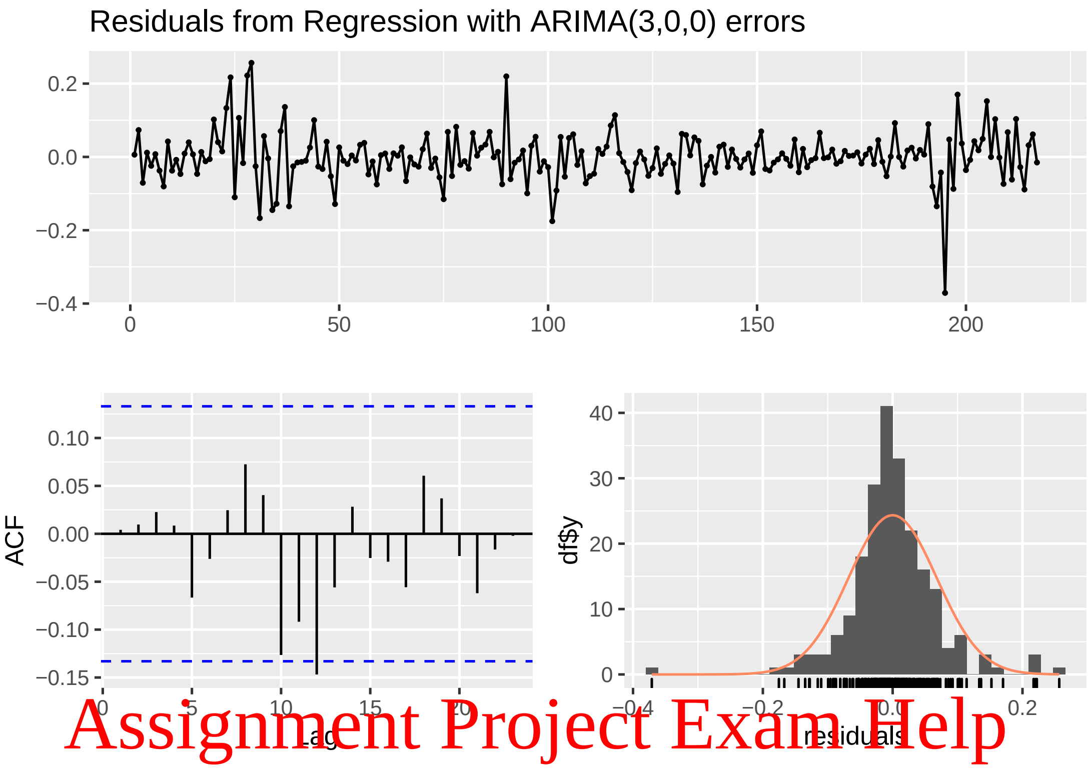

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 7.1189, df = 6, p-value = 0.31
##
## Model df: 4.   Total lags used: 10
```



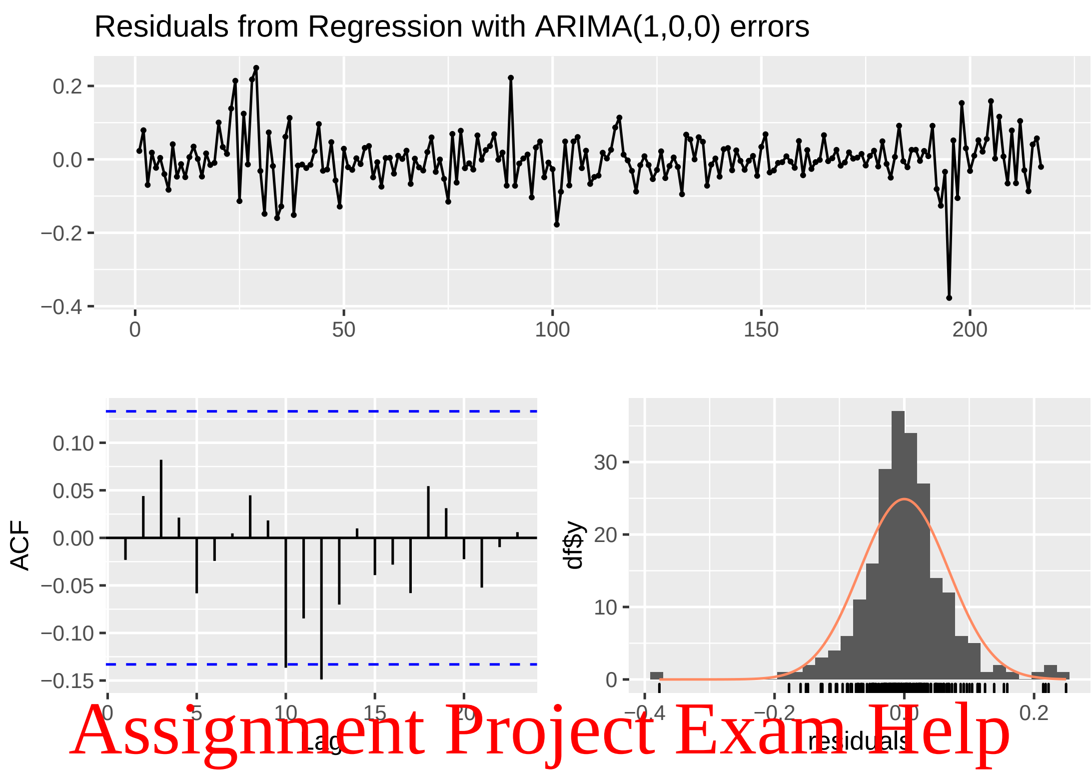

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 7.8281, df = 7, p-value = 0.348
##
## Model df: 3.   Total lags used: 10
```



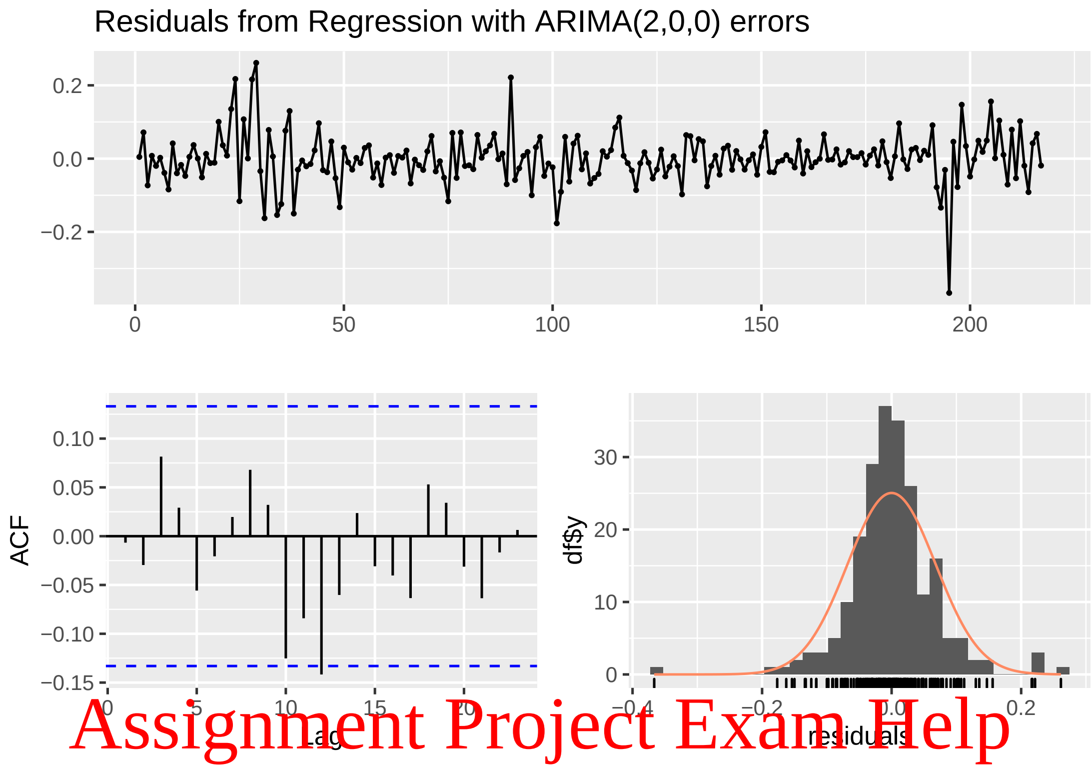

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 7.6639, df = 6, p-value = 0.2638
##
## Model df: 4.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 6.6809, df = 5, p-value = 0.2455
##
## Model df: 5.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 7.8597, df = 6, p-value = 0.2486
##
## Model df: 4.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 7.6345, df = 5, p-value = 0.1776
##
## Model df: 5.   Total lags used: 10
```



```
adf.test(i180d - i90d)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag   ADF p.value
## [1,]   0 -3.02    0.01
## [2,]   1 -3.84    0.01
## [3,]   2 -3.92    0.01
## [4,]   3 -4.14    0.01
## [5,]   4 -4.00    0.01
## Type 2: with drift no trend
##      lag   ADF p.value
## [1,]   0 -3.39  0.0135
## [2,]   1 -4.26  0.0100
## [3,]   2 -4.34  0.0100
## [4,]   3 -4.64  0.0100
```

```
## [5,]   4 -4.52  0.0100
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,]   0 -3.40  0.0551
## [2,]   1 -4.26  0.0100
## [3,]   2 -4.33  0.0100
## [4,]   3 -4.64  0.0100
## [5,]   4 -4.52  0.0100
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

The test rejects a unit root for all models in the adequate set. Note that some specifications lead to a failure to reject, but they are not in our adequate set, so we can ignore them! We can confidently conclude that the money market spread is I(0) and the ETT holds for the Money Market.
