GDP_Prem_Tests

Nathan Barretto

2024-05-19

```
#invert rows and columns for gdp dataset
#source: https://stackoverflow.com/questions/33643181/how-do-i-flip-rows-and-columns-in-r
gdp_data_2 <- data.frame(t(gdp_data[-1]))
colnames(gdp_data) <- gdp_data[, 1]
```

```
#rename columns in proxy dataset

#first column name changed to year_month
colnames(proxy_data)[1] <- "quarter_year"
```

```
#adjust column names in gdp dataset to get the names of the columns to be that
#of the first row names

for (x in 3:38) {
  name <- gdp_data_2[1,x]
  name_string <- toString(name)
  colnames(gdp_data_2)[x] <- name_string
}
```

```
#Adjust column names in proxy and gdp data to accurately reflect measurements
colnames(proxy_data)[67] <- "DPRatio"
#one-year equity premium bound proposed by Ian Martin in 2017
colnames(proxy_data)[70] <- "EP_Bound"
colnames(proxy_data)[47] <- "IPO_Gross"
colnames(proxy_data)[75] <- "Price_Volatile_Stocks"
colnames(proxy_data)[69] <- "Surplus_Cons_Ratio"
colnames(proxy_data)[76] <- "Agreggate_Risk_Aversion"
colnames(proxy_data)[19] <- "Unemployment_Rate"
colnames(gdp_data_2)[5] <- "GDP"
```

```
#delete row 1 of gdp_data with the old names
gdp_data_2 <- gdp_data_2[-c(1), ]
```

```
#delete first 2 columns of gdp_data
#Blank cells
gdp_data_2 <- gdp_data_2[ -c(1,2) ]
```

```
#split proxy_data by quarter and year

#quarter makers function
quarter_makers = function(mths){
 q1_Vector <- c(1, 2, 3)
 q2_Vector <- c(4, 5, 6)
 q3_Vector <- c(7, 8, 9)
 q4_Vector <- c(10, 11, 12)

 result <- c()

 for(i in 1:length(mths)){
  if (mths[i] %in% q1_Vector ){
    result[i] <- "Q1"
  }
  else if (mths[i] %in% q2_Vector ){
    result[i] <- "Q2"
  }
  else if (mths[i] %in% q3_Vector ){
    result[i] <- "Q3"
  }
   else {
    result[i] <- "Q4"
   }
 }

 return (result)
}
```

```
#Generate split of quarter and years using quarter makers function
proxy_data$months = as.numeric(substr(as.character(proxy_data$quarter_year),5,6))
proxy_data$Year = as.numeric(substr(as.character(proxy_data$quarter_year),1,4))
proxy_data$Quarter = quarter_makers(proxy_data$months)
```

```
#download hmisc package
#install.packages("Hmisc")
#install.packages("acepack")
```

```
#Modify both datasets so they include the same quarters from 1960 - 2015
#results in same dimensions for statistical analysis later
proxy_data_mod_final <- proxy_data[-c(1: 395), ]
```

```
#Modify both datasets so they include the same quarters from 1960 - 2015
#results in same dimensions for statistical analysis later
gdp_data_mod_final <- gdp_data_2[-c(1: 51), ]
```

```
#Modify both datasets so they include the same quarters from 1960 - 2015
#results in same dimensions for statistical analysis later
gdp_data_mod_final <- gdp_data_mod_final[-c(225: 257), ]
```

```
#Create geometric mean function to get geometric mean for quarters
#for proxy_data
geom_mean = function(months){
  result <- c()
  i = 1
  while(i < length(months)) {
    total <- c()
    for (j in 0:2){
      len_total <- length(total)
      total[[len_total + 1]] <- as.numeric(months[i + j])
    }
    value = exp(mean(log(as.numeric(total))))
    len_result <- length(result)
    result[[len_result + 1]] <- value
    i = i + 3
  }
  return (as.numeric(result))
}
```

```
#Create arithmetic mean function to get arithmetic mean for quarters
#for proxy_data
#Will use this for proxy measures that have negative values
ari_mean = function(months){
  result <- c()
  i = 1
  while(i < length(months)) {
    total <- c()
    for (j in 0:2){
      len_total <- length(total)
      total[[len_total + 1]] <- as.numeric(months[i + j])
    }
    value = mean(as.numeric(total))
    len_result <- length(result)
    result[[len_result + 1]] <- value
    i = i + 3
  }
  return (as.numeric(result))
}
```

```
#Since each measure in the proxy dataset has different missing values/ years with data filled i
n,
#I will split the large, combined dataset into smaller individual datasets by specific
#proxy measure


#The 7 measures and their respective datasets:

#1. IPOGross (Gross Initial Public Offering), name: combined_data_IPO_Gross
#2. Aggregate ,  name: combined_data_Agreggate_Risk_Aversion
#3. DP (Dividend Price Ratio),  name: combined_data_DPRatio
#4. EPBound (Equity Premium Bound),  name: combined_data_EPBound
#5. SCR (Surplus Consumption Ratio),  name: combined_data_surplus
#6. UNE (Unemployment Rate),  name: combined_data_UNE
#7. PVS (Price of Volatile Stocks),  name: combined_data_PVS
```

```
#Apply geom mean function to different proxy measures by quarter average and
#add them to new, cumulative dataset with that measure for testing
combined_data_IPO_Gross <- data.frame(gdp_data_mod_final)
#Gross IPO By Quarter Added
combined_data_IPO_Gross$IPO_Gross = geom_mean(proxy_data_mod_final$IPO_Gross)
#remove original combined dataset
rm(combined_data)
#
```

```
#Apply geom mean function to different proxy measures by quarter average and
#add them to new, cumulative dataset
#Aggregate Risk Aversion only has data from 1990 - 2010
#Start by changing gdp_data to only include this
combined_data_Aggregate_Risk_Aversion <- data.frame(gdp_data_mod_final)
combined_data_Aggregate_Risk_Aversion <- combined_data_Aggregate_Risk_Aversion[-c(1: 120), ]
combined_data_Aggregate_Risk_Aversion <- combined_data_Aggregate_Risk_Aversion[-c(85: 104), ]
#Do the same for the proxy data
proxy_data_cut <- data.frame(proxy_data_mod_final)
proxy_data_cut <- proxy_data_cut[-c(613: 672), ]
proxy_data_cut <- proxy_data_cut[-c(1: 360), ]
#Since aggregate risk aversion has negatives use arithmetic mean for quarter averages
#Aggregate Risk Aversion By Quarter Added
combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion = ari_mean(proxy_data_cut$Agreggat
e_Risk_Aversion)
#
```

```
#Apply geom mean function to different proxy measures by quarter average and
#add them to new, cumulative dataset with that measure for testing
combined_data_DPRatio <- data.frame(gdp_data_mod_final)
#DPRatio By Quarter Added
combined_data_DPRatio$DPRatio = geom_mean(proxy_data_mod_final$DPRatio)
```

```
#Apply geom mean function to different proxy measures by quarter average and
#add them to new, cumulative dataset
#Equity Premium Bound only has data from 1996 - 2011
#Start by changing gdp_data to only include this
combined_data_EPBound <- data.frame(gdp_data_mod_final)
combined_data_EPBound <- combined_data_EPBound[-c(1: 144), ]
combined_data_EPBound <- combined_data_EPBound[-c(65: 80), ]
#Do the same for the proxy data
proxy_data_cut <- data.frame(proxy_data_mod_final)
proxy_data_cut <- proxy_data_cut[-c(625: 672), ]
proxy_data_cut <- proxy_data_cut[-c(1: 432), ]
#Equity Premium Bound By Quarter Added
combined_data_EPBound$EPBound = geom_mean(proxy_data_cut$EP_Bound)
#
```

```
#Apply geom mean function to different proxy measures by quarter average and
#add them to new, cumulative dataset with that measure for testing
combined_data_SCR <- data.frame(gdp_data_mod_final)
#Surplus Consumption Ratio By Quarter Added
#negative values so use arithmetic mean
combined_data_SCR$SCR = ari_mean(proxy_data_mod_final$Surplus_Cons_Ratio)
```

```
#Apply geom mean function to different proxy measures by quarter average and
#add them to new, cumulative dataset with that measure for testing
combined_data_UNE <- data.frame(gdp_data_mod_final)
#Unemployment Rate By Quarter Added
combined_data_UNE$UNE = geom_mean(proxy_data_mod_final$Unemployment_Rate)
```

```
#Apply geom mean function to different proxy measures by quarter average and
#add them to new, cumulative dataset with that measure for testing
#PVS only has data from 1970 quarter 2, and is itself only available by quarter
combined_data_PVS <- data.frame(gdp_data_mod_final)
combined_data_PVS <- combined_data_PVS[-c(1: 41), ]
proxy_data_cut <- data.frame(proxy_data_mod_final)
proxy_data_cut <- proxy_data_cut[-c(1: 123), ]
#Run a for loop to keep every third row with the pvs data of the quarter in
#the proxy dataset
 i = 3
 total <- c()
 while(i <= length(proxy_data_cut$Price_Volatile_Stocks)) {
    value = proxy_data_cut$Price_Volatile_Stocks[i]
    len_total <- length(total)
    total[[len_total + 1]] <- value
    i = i + 3
  }
#Price of Volatile Stock By Quarter Added
combined_data_PVS$PVS = total
```

```
#All datasets now completed, can begin running tests
```

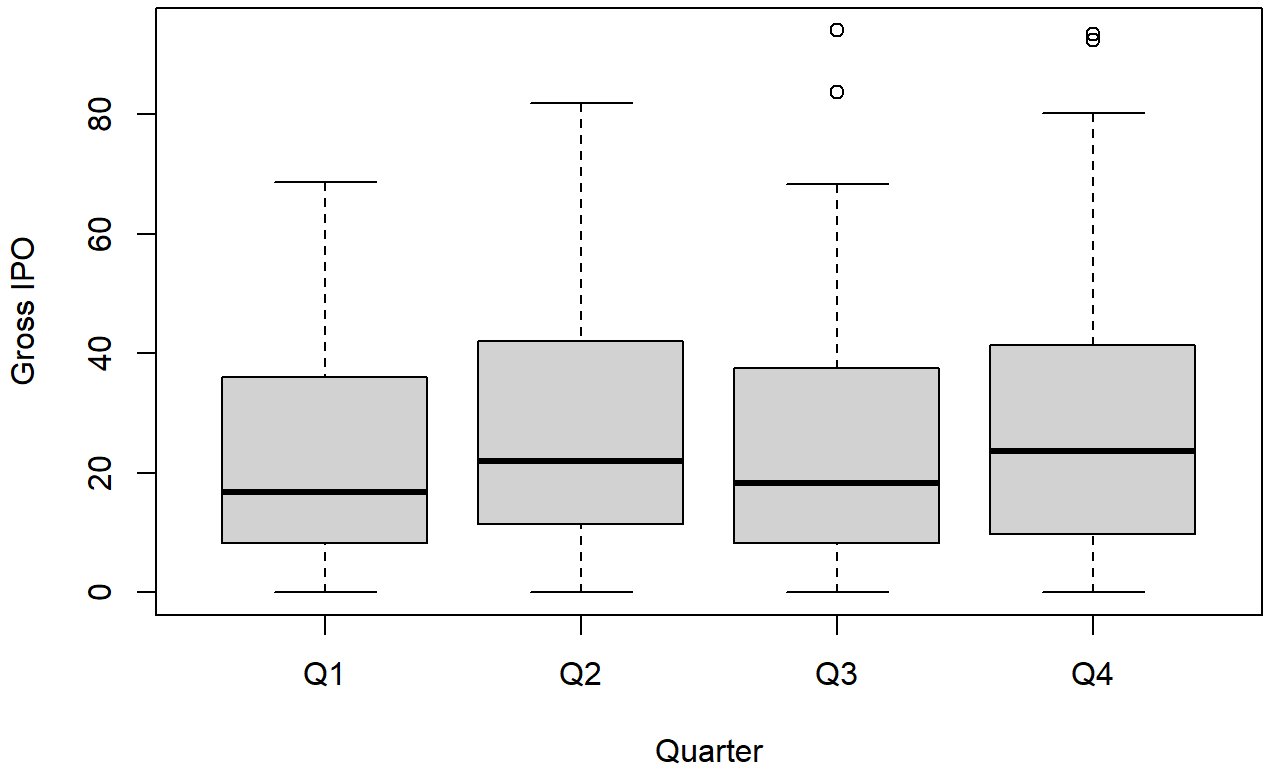
```
#Equity Premium Proxy Measure 1: Gross IPO, all tests and graphs below
```

```
#Start with an anova test for Gross IPO

#1: Anova test on the quarters to see if there is difference in the Gross IPO
#by time in the year

IPO.aov <- aov(IPO_Gross ~ Quarter, data = combined_data_IPO_Gross)
# Summary
summary(IPO.aov)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Quarter       3   1464   488.1   1.096  0.351
## Residuals   220  97927   445.1
```

```
#P-value of 0.351. Indicates that there are no significant differences in the
#mean gross ipo by quarter
```

```
#boxplots of the quarter ~ gross ipo relationship
#Notes: Outliers present for Q3 and Q4
#Notes: Q1 most similar to Q3, Q2 most similar to Q4
#Notes: Potential oscillation between quarters?
boxplot(IPO_Gross~Quarter,data=combined_data_IPO_Gross, main="Gross IPO vs Yearly Quarter",
   xlab="Quarter", ylab="Gross IPO")
```




```
#Run correlation tests between different GDP measures and gross IPO
#Years: 1960 to 2015
#Correlation with GNP
cor(combined_data_IPO_Gross$IPO_Gross, as.numeric(combined_data_IPO_Gross$Gross.national.produc
t))
```

```
## [1] -0.1925319
```

```
#Correlation with GDP
cor(combined_data_IPO_Gross$IPO_Gross,  as.numeric(combined_data_IPO_Gross$GDP))
```

```
## [1] -0.1934336
```

```
#Correlation with GPDI
cor(combined_data_IPO_Gross$IPO_Gross, as.numeric(combined_data_IPO_Gross$Gross.private.domesti
c.investment))
```

```
## [1] -0.2314912
```

```
#Correlation with nondurable goods
cor(combined_data_IPO_Gross$IPO_Gross, as.numeric(combined_data_IPO_Gross$Nondurable.goods))
```

```
## [1] -0.1859671
```

```
#Correlation with personal consumption expenditures
cor(combined_data_IPO_Gross$IPO_Gross, as.numeric(combined_data_IPO_Gross$Personal.consumption.e
xpenditures))
```

```
## [1] -0.1769917
```

```
#all negative correlations around -0.2, indicates weak, negative correlation
```

```
#Linear regression tests
#Start with gross ipo and GNP
lmIPO_GNP = lm(as.numeric(Gross.national.product) ~ IPO_Gross, data = combined_data_IPO_Gross)
summary(lmIPO_GNP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ IPO_Gross,
##     data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3211 -1.6983 -0.5933  0.9027  8.9425
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.957547   0.253435  15.616  < 2e-16 ***
## IPO_Gross   -0.022466   0.007685  -2.923  0.00382 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.423 on 222 degrees of freedom
## Multiple R-squared:  0.03707,    Adjusted R-squared:  0.03273
## F-statistic: 8.546 on 1 and 222 DF,  p-value: 0.003821
```

```
#try with no intercept now
lmIPO_GNP_NOINT = lm(as.numeric(Gross.national.product) ~ IPO_Gross - 1, data = combined_data_IP
O_Gross)
summary(lmIPO_GNP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ IPO_Gross -
##     1, data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2016 -0.3154  0.9729  3.0280 12.9000
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## IPO_Gross 0.069869   0.007095   9.848   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.502 on 223 degrees of freedom
## Multiple R-squared:  0.3031, Adjusted R-squared:    0.3
## F-statistic: 96.98 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
#Linear regression tests
#gross ipo and gdp
lmIPO_GDP = lm(as.numeric(GDP) ~ IPO_Gross, data = combined_data_IPO_Gross)
summary(lmIPO_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ IPO_Gross, data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3187 -1.6961 -0.5977  0.9074  8.9444
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.955589   0.253120  15.627  < 2e-16 ***
## IPO_Gross   -0.022547   0.007675  -2.938  0.00366 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.42 on 222 degrees of freedom
## Multiple R-squared:  0.03742,    Adjusted R-squared:  0.03308
## F-statistic: 8.629 on 1 and 222 DF,  p-value: 0.003657
```

```
#try with no intercept now (X - 1)
lmIPO_GDP_NOINT = lm(as.numeric(GDP) ~ IPO_Gross - 1, data = combined_data_IPO_Gross)
summary(lmIPO_GDP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ IPO_Gross - 1, data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1914 -0.3131  1.0126  3.0314 12.9000
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## IPO_Gross 0.069742   0.007089   9.838   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.499 on 223 degrees of freedom
## Multiple R-squared:  0.3027, Adjusted R-squared:  0.2995
## F-statistic: 96.79 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
#Linear regression tests
#gross ipo and Gross private domestic investment
lmIPO_GDP = lm(as.numeric(Gross.private.domestic.investment) ~ IPO_Gross, data = combined_data_I
PO_Gross)
summary(lmIPO_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     IPO_Gross, data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4788 -2.2351 -0.6654  1.5404 12.9918
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.60824    0.35144  10.267  < 2e-16 ***
## IPO_Gross   -0.03778    0.01066  -3.545 0.000478 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.36 on 222 degrees of freedom
## Multiple R-squared:  0.05359,    Adjusted R-squared:  0.04933
## F-statistic: 12.57 on 1 and 222 DF,  p-value: 0.0004779
```

```
#try with no intercept now (X - 1)
lmIPO_GPDI_NOINT = lm(as.numeric(Gross.private.domestic.investment) ~ IPO_Gross - 1, data = comb
ined_data_IPO_Gross)
summary(lmIPO_GPDI_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     IPO_Gross - 1, data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3818 -0.8887  0.3478  2.9601 16.6000
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## IPO_Gross 0.046403   0.008248   5.626  5.5e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.071 on 223 degrees of freedom
## Multiple R-squared:  0.1243, Adjusted R-squared:  0.1204
## F-statistic: 31.65 on 1 and 223 DF,  p-value: 5.502e-08
```

```
#Linear regression tests
#gross ipo and nondurable goods
lmIPO_NDG = lm(as.numeric(Nondurable.goods) ~ IPO_Gross, data = combined_data_IPO_Gross)
summary(lmIPO_NDG)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ IPO_Gross, data = combined_data_IPO_Gross)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.7249  -2.4122  -0.3984   2.3449  18.0751
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.22488    0.49209   8.586  1.6e-15 ***
## IPO_Gross   -0.04208    0.01492  -2.820  0.00524 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.704 on 222 degrees of freedom
## Multiple R-squared:  0.03458,    Adjusted R-squared:  0.03024
## F-statistic: 7.953 on 1 and 222 DF,  p-value: 0.005236
```

```
#try with no intercept now (X - 1)
lmIPO_NDG_NOINT = lm(as.numeric(Nondurable.goods) ~ IPO_Gross - 1, data = combined_data_IPO_Gros
s)
summary(lmIPO_NDG_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ IPO_Gross - 1, data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.500  -0.895   1.083   4.228  22.300
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## IPO_Gross  0.05649    0.01098   5.147  5.8e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.417 on 223 degrees of freedom
## Multiple R-squared:  0.1062, Adjusted R-squared:  0.1022
## F-statistic: 26.49 on 1 and 223 DF,  p-value: 5.803e-07
```

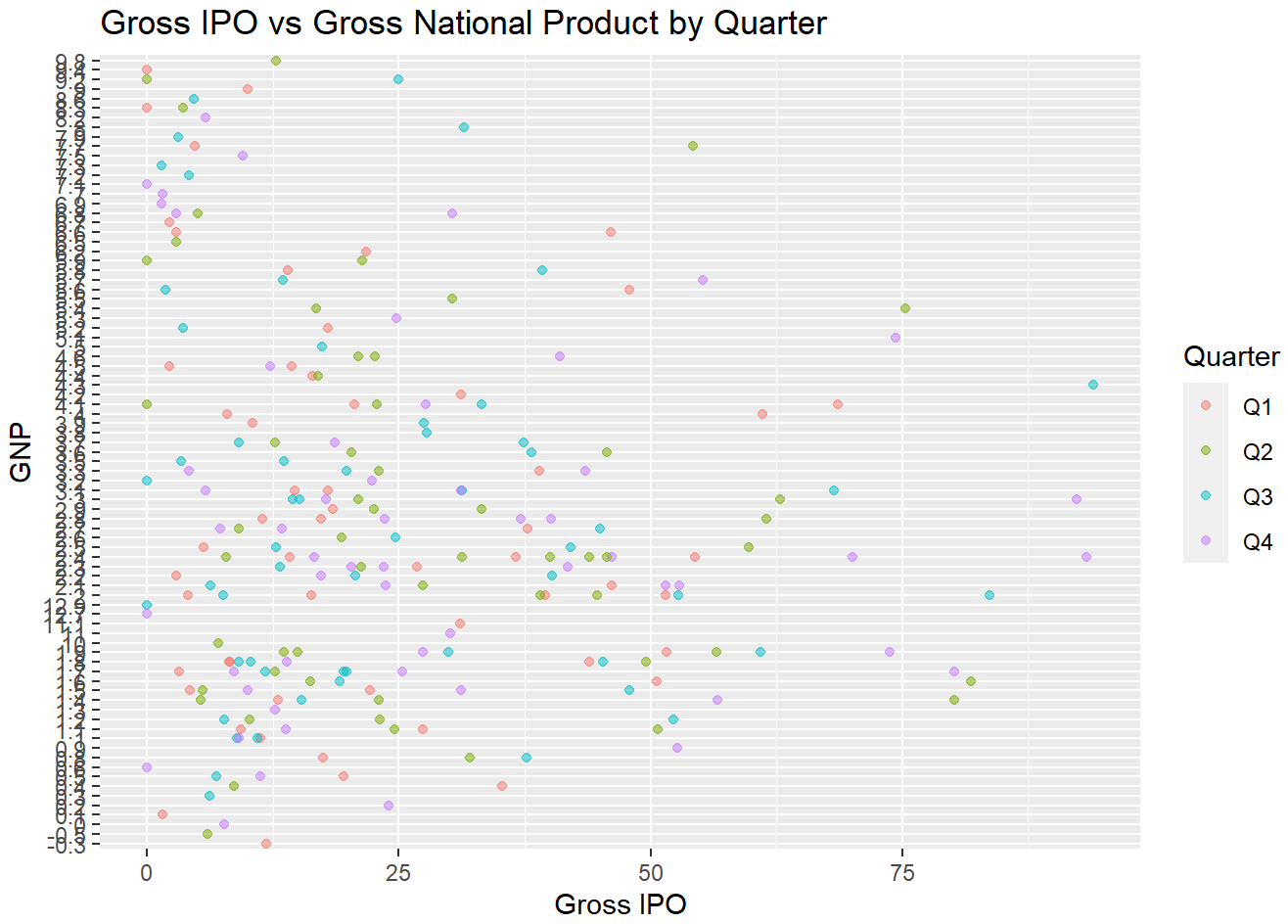
```
#Linear regression tests
#gross ipo and personal consumption expenditures
lmIPO_PCE = lm(as.numeric(Personal.consumption.expenditures) ~ IPO_Gross, data = combined_data_I
PO_Gross)
summary(lmIPO_PCE)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     IPO_Gross, data = combined_data_IPO_Gross)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1612  -1.6753  -0.3899   1.0984   8.8668
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.961158   0.279776  14.158  < 2e-16 ***
## IPO_Gross   -0.022732   0.008484  -2.679  0.00793 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.675 on 222 degrees of freedom
## Multiple R-squared:  0.03133,    Adjusted R-squared:  0.02696
## F-statistic: 7.179 on 1 and 222 DF,  p-value: 0.007927
```

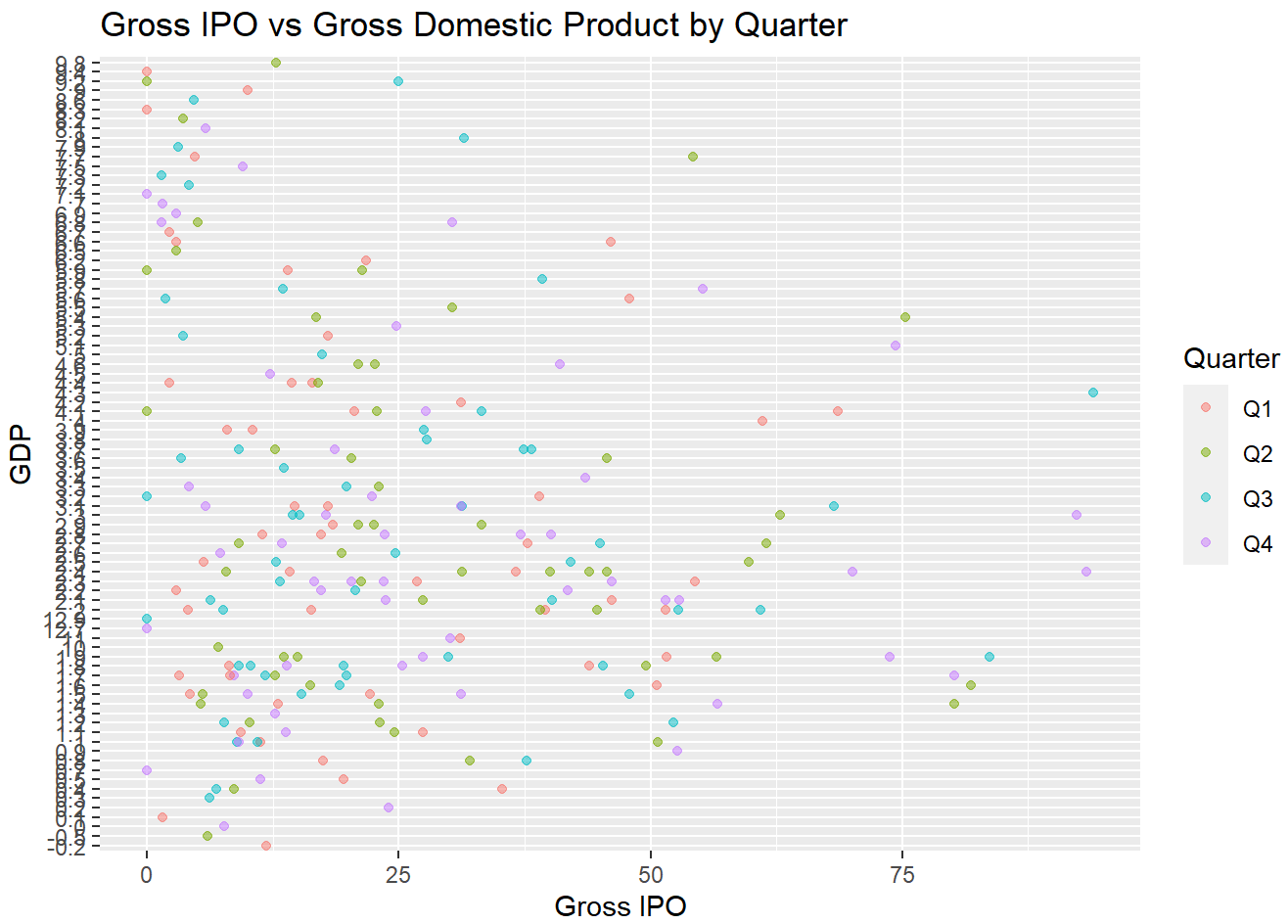
```
#try with no intercept now (X - 1)
lmIPO_PCE_NOINT = lm(as.numeric(Personal.consumption.expenditures) ~ IPO_Gross - 1, data = combi
ned_data_IPO_Gross)
summary(lmIPO_PCE_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     IPO_Gross - 1, data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2000 -0.3199  0.9046  2.7255 12.5000
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## IPO_Gross 0.069688   0.007459   9.343   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.681 on 223 degrees of freedom
## Multiple R-squared:  0.2813, Adjusted R-squared:  0.2781
## F-statistic:  87.3 on 1 and 223 DF,  p-value: < 2.2e-16
```

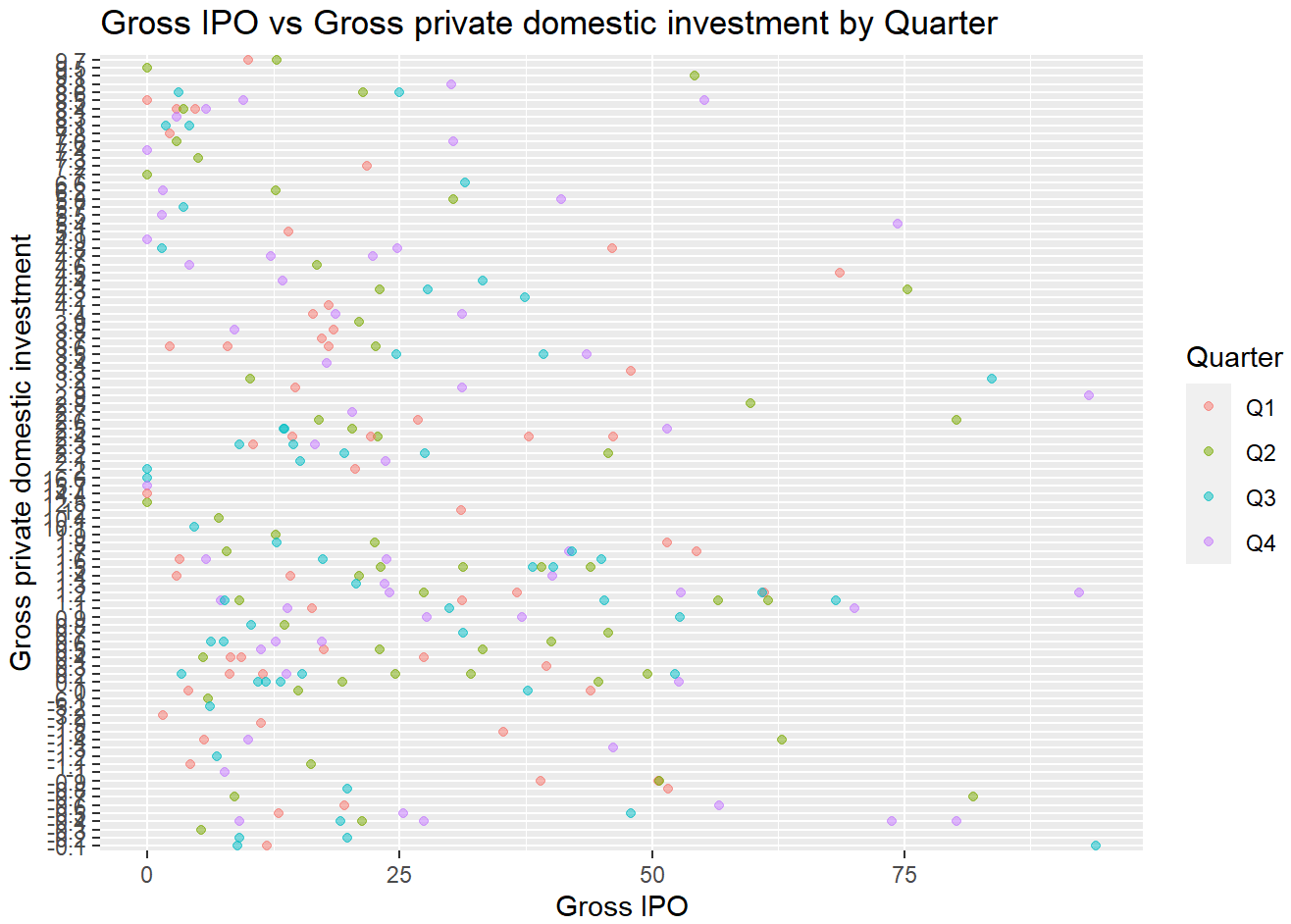
```
#Make scatterplots of relationships, color by quarter
#Start with IPO Gross and GNP
 combined_data_IPO_Gross|>
  ggplot(aes(x = IPO_Gross, y = Gross.national.product,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "Gross IPO", y = "GNP", color = "Quarter",
       title = "Gross IPO vs Gross National Product by Quarter")
```



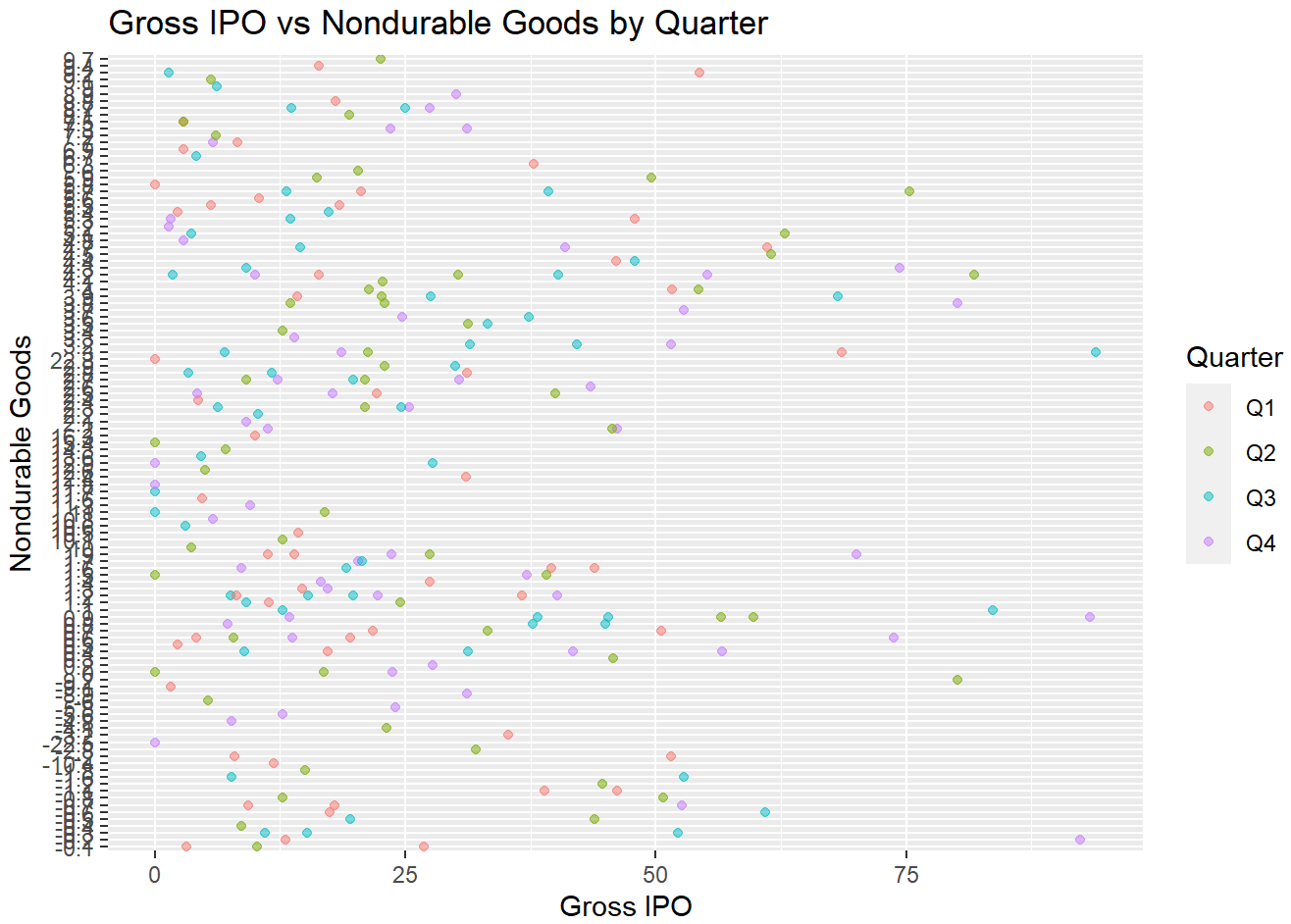

```
#Make scatterplots of relationships, color by quarter
#IPO Gross and GDP
 combined_data_IPO_Gross|>
  ggplot(aes(x = IPO_Gross, y = GDP,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "Gross IPO", y = "GDP", color = "Quarter",
       title = "Gross IPO vs Gross Domestic Product by Quarter")
```



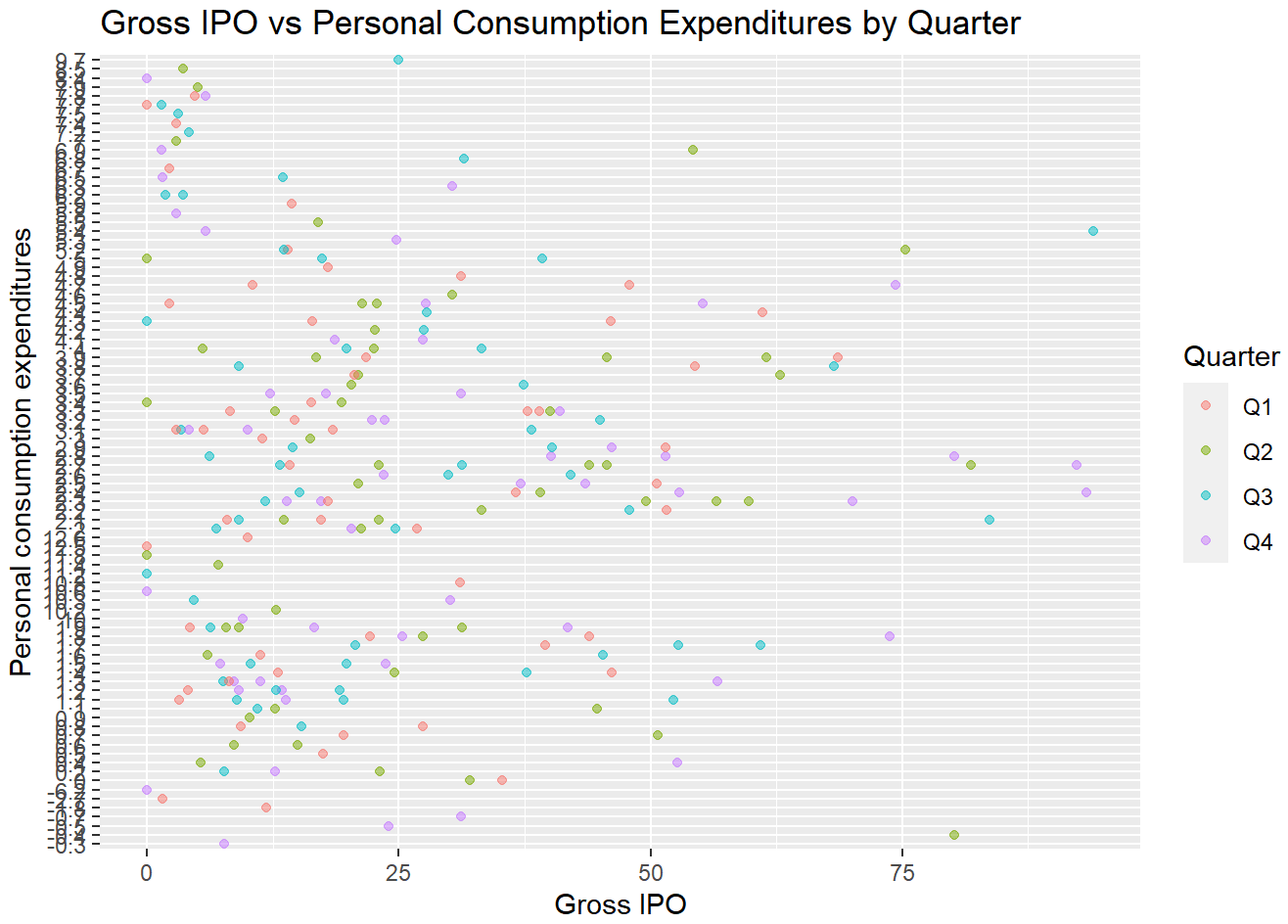

```
#Make scatterplots of relationships, color by quarter
#IPO Gross and GPDI
 combined_data_IPO_Gross|>
  ggplot(aes(x = IPO_Gross, y = Gross.private.domestic.investment,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "Gross IPO", y = "Gross private domestic investment", color = "Quarter",
       title = "Gross IPO vs Gross private domestic investment by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#IPO Gross and Nondurable Goods
 combined_data_IPO_Gross|>
  ggplot(aes(x = IPO_Gross, y = Nondurable.goods,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "Gross IPO", y = "Nondurable Goods", color = "Quarter",
       title = "Gross IPO vs Nondurable Goods by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#IPO Gross and Personal consumption expenditures
 combined_data_IPO_Gross|>
  ggplot(aes(x = IPO_Gross, y = Personal.consumption.expenditures,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "Gross IPO", y = "Personal consumption expenditures", color = "Quarter",
       title = "Gross IPO vs Personal Consumption Expenditures by Quarter")
```



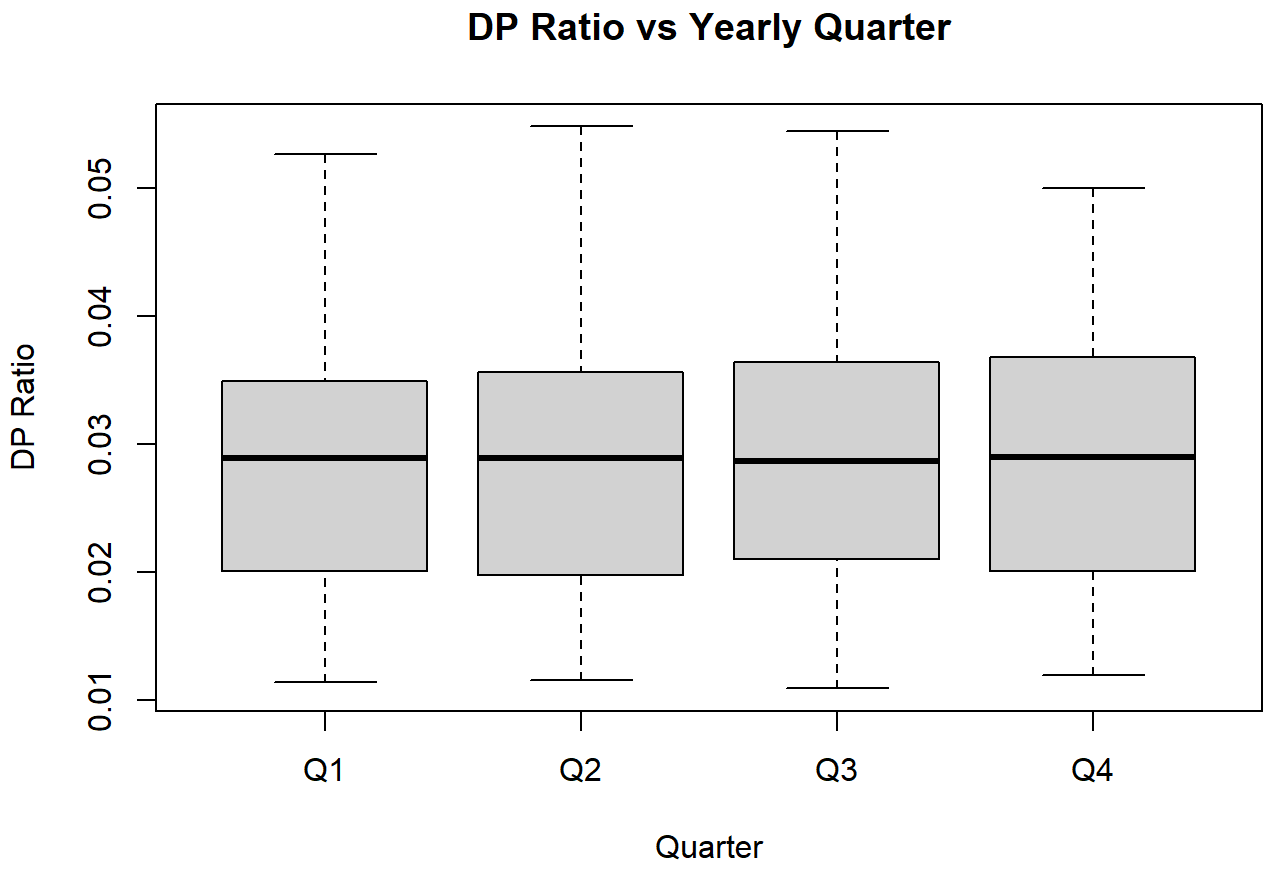

```
#Equity Premium Proxy Measure 2: Dividend Price ratio, all tests and graphs below
```

```
#Start with an anova test for DP ratio

#1: Anova test on the quarters to see if there is difference in the DP Ratio
#by time in the year

DP.aov <- aov(DPRatio ~ Quarter, data = combined_data_DPRatio)
# Summary
summary(DP.aov)
```

```
##              Df   Sum Sq   Mean Sq F value Pr(>F)
## Quarter       3 0.000007 2.270e-06   0.022  0.996
## Residuals   220 0.023114 1.051e-04
```

```
#P-value of 0.996. Indicates that there are no significant differences in the
#mean DP Ratio by quarter
```

```
#boxplots of the quarter ~ gross ipo relationship
#Notes: No real Outliers
#Notes: Almost all 4 quarters identical
#Notes: No real oscillation between quarters like with gross IPO
#Notes: Q3 has the smallest IQR
boxplot(DPRatio~Quarter,data=combined_data_DPRatio, main="DP Ratio vs Yearly Quarter",
   xlab="Quarter", ylab="DP Ratio")
```




```
#Run correlation tests between different GDP measures and DPRatio
#Years: 1960 to 2015 (Full Data)
#Correlation with GNP
cor(combined_data_DPRatio$DPRatio, as.numeric(combined_data_DPRatio$Gross.national.product))
```

```
## [1] 0.6628789
```

```
#Correlation with GDP
cor(combined_data_DPRatio$DPRatio,  as.numeric(combined_data_DPRatio$GDP))
```

```
## [1] 0.6633133
```

```
#Correlation with GPDI
cor(combined_data_DPRatio$DPRatio, as.numeric(combined_data_DPRatio$Gross.private.domestic.investment))
```

```
## [1] 0.554113
```

```
#Correlation with nondurable goods
cor(combined_data_DPRatio$DPRatio, as.numeric(combined_data_DPRatio$Nondurable.goods))
```

```
## [1] 0.280237
```

```
#Correlation with personal consumption expenditures
cor(combined_data_DPRatio$DPRatio, as.numeric(combined_data_DPRatio$Personal.consumption.expenditures))
```

```
## [1] 0.6165728
```

```
#Notes: All positive correlations, greater than 0.6 for GDP, GNP, and Personal consumption expenditures indicating
#indicating strong correlations with those items
```

```
#Linear regression tests
#Start with DPRatio and GNP
lmDP_GNP = lm(as.numeric(Gross.national.product) ~ DPRatio, data = combined_data_DPRatio)
summary(lmDP_GNP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ DPRatio, data = combined_data_DPRatio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3057 -1.2617 -0.0642  1.0906  6.3139
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.2386     0.3718  -3.331  0.00101 **
## DPRatio     160.3735    12.1575  13.191  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.849 on 222 degrees of freedom
## Multiple R-squared:  0.4394, Adjusted R-squared:  0.4369
## F-statistic:   174 on 1 and 222 DF,  p-value: < 2.2e-16
```

```
#try with no intercept now
lmDP_GNP_NOINT = lm(as.numeric(Gross.national.product) ~ DPRatio - 1, data = combined_data_DPRat
io)
summary(lmDP_GNP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ DPRatio - 1,
##     data = combined_data_DPRatio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1998 -1.2943 -0.5059  0.8518  6.9391
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## DPRatio  122.174      4.129   29.59   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.89 on 223 degrees of freedom
## Multiple R-squared:  0.797,  Adjusted R-squared:  0.7961
## F-statistic: 875.4 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
#Linear regression tests
#DPRatio and gdp
lmDP_GDP = lm(as.numeric(GDP) ~ DPRatio, data = combined_data_DPRatio)
summary(lmDP_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ DPRatio, data = combined_data_DPRatio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3012 -1.2572 -0.0606  1.0727  6.3193
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.2407     0.3712  -3.342 0.000975 ***
## DPRatio     160.3079    12.1384  13.207  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.846 on 222 degrees of freedom
## Multiple R-squared:   0.44,  Adjusted R-squared:  0.4375
## F-statistic: 174.4 on 1 and 222 DF,  p-value: < 2.2e-16
```

```
#try with no intercept now (X - 1)
lmDP_GDP_NOINT = lm(as.numeric(GDP) ~ DPRatio - 1, data = combined_data_DPRatio)
summary(lmDP_GDP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ DPRatio - 1, data = combined_data_DPRatio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1952 -1.2896 -0.4873  0.8546  6.9455
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## DPRatio  122.043      4.123    29.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.887 on 223 degrees of freedom
## Multiple R-squared:  0.7971, Adjusted R-squared:  0.7962
## F-statistic: 876.1 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
#Linear regression tests
#DPRatio and Gross private domestic investment
lmDP_GPDI = lm(as.numeric(Gross.private.domestic.investment) ~ DPRatio, data = combined_data_DPR
atio)
summary(lmDP_GPDI)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     DPRatio, data = combined_data_DPRatio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7219 -1.7148 -0.3042  1.7591 10.2106
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.7594     0.5782  -4.772  3.3e-06 ***
## DPRatio     187.5138    18.9065   9.918  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.875 on 222 degrees of freedom
## Multiple R-squared:  0.307,  Adjusted R-squared:  0.3039
## F-statistic: 98.37 on 1 and 222 DF,  p-value: < 2.2e-16
```

```
#try with no intercept now (X - 1)
lmDP_GPDI_NOINT = lm(as.numeric(Gross.private.domestic.investment) ~ DPRatio - 1, data = combine
d_data_DPRatio)
summary(lmDP_GPDI_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     DPRatio - 1, data = combined_data_DPRatio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0390 -2.1831 -0.9036  1.3603 11.6034
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## DPRatio   102.41       6.58   15.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.012 on 223 degrees of freedom
## Multiple R-squared:  0.5206, Adjusted R-squared:  0.5185
## F-statistic: 242.2 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
#Linear regression tests
#DPRatio and nondurable goods
lmDP_NDG = lm(as.numeric(Nondurable.goods) ~ DPRatio, data = combined_data_DPRatio)
summary(lmDP_NDG)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ DPRatio, data = combined_data_DPRatio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.970  -2.536  -0.516   2.049  18.507
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.6354     0.9244  -0.687    0.493
## DPRatio     131.4753    30.2261   4.350 2.08e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.596 on 222 degrees of freedom
## Multiple R-squared:  0.07853,    Adjusted R-squared:  0.07438
## F-statistic: 18.92 on 1 and 222 DF,  p-value: 2.078e-05
```

```
#try with no intercept now (X - 1)
lmDP_NDG_NOINT = lm(as.numeric(Nondurable.goods) ~ DPRatio - 1, data = combined_data_DPRatio)
summary(lmDP_NDG_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ DPRatio - 1, data = combined_data_DPRatio)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.9937  -2.6505  -0.4856   1.8700  18.5316
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## DPRatio   111.88      10.03   11.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.591 on 223 degrees of freedom
## Multiple R-squared:  0.3582, Adjusted R-squared:  0.3553
## F-statistic: 124.4 on 1 and 223 DF,  p-value: < 2.2e-16
```

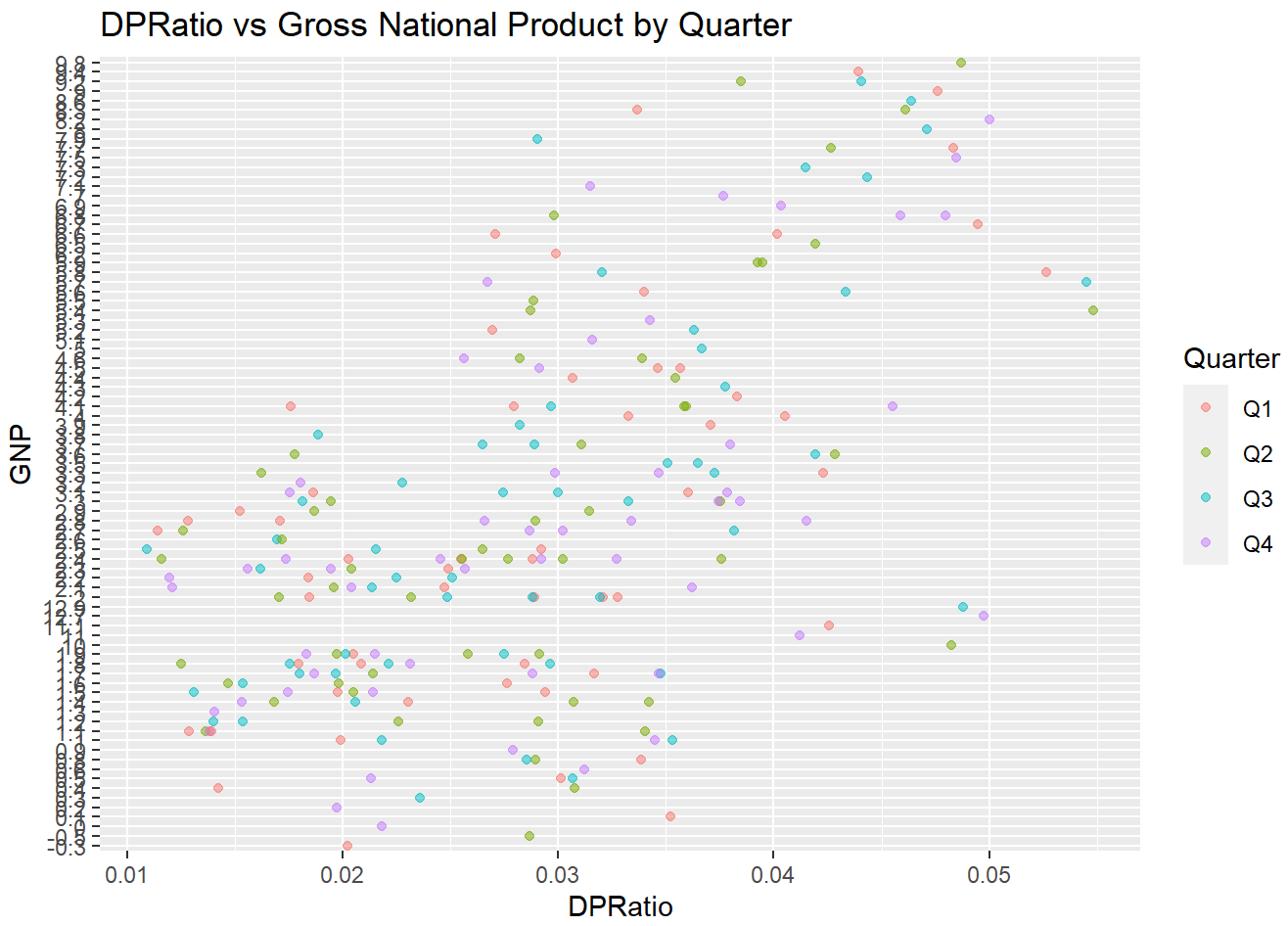
```
#Linear regression tests
#gross ipo and personal consumption expenditures
lmDP_PCE = lm(as.numeric(Personal.consumption.expenditures) ~ DPRatio, data = combined_data_DPRa
tio)
summary(lmDP_PCE)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     DPRatio, data = combined_data_DPRatio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9754 -1.2543 -0.0715  1.2126  8.3215
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.3517     0.4303  -3.141  0.00191 **
## DPRatio     164.1855    14.0706  11.669  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.14 on 222 degrees of freedom
## Multiple R-squared:  0.3802, Adjusted R-squared:  0.3774
## F-statistic: 136.2 on 1 and 222 DF,  p-value: < 2.2e-16
```

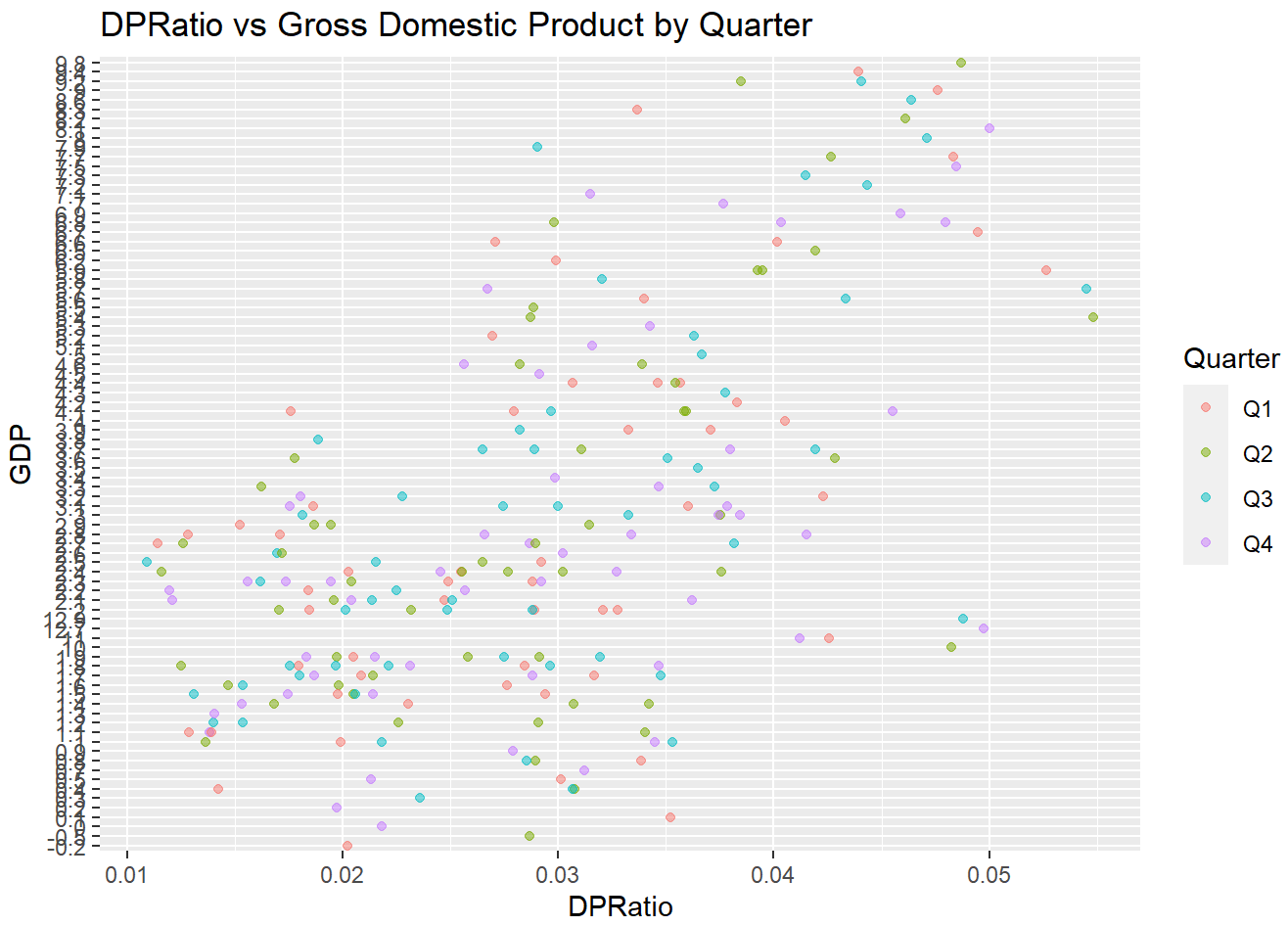
```
#try with no intercept now (X - 1)
lmDP_PCE_NOINT = lm(as.numeric(Personal.consumption.expenditures) ~ DPRatio - 1, data = combined
_data_DPRatio)
summary(lmDP_PCE_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     DPRatio - 1, data = combined_data_DPRatio)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0253  -1.4473  -0.4349   0.8353   8.3739
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## DPRatio  122.498      4.766    25.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.182 on 223 degrees of freedom
## Multiple R-squared:  0.7476, Adjusted R-squared:  0.7465
## F-statistic: 660.5 on 1 and 223 DF,  p-value: < 2.2e-16
```

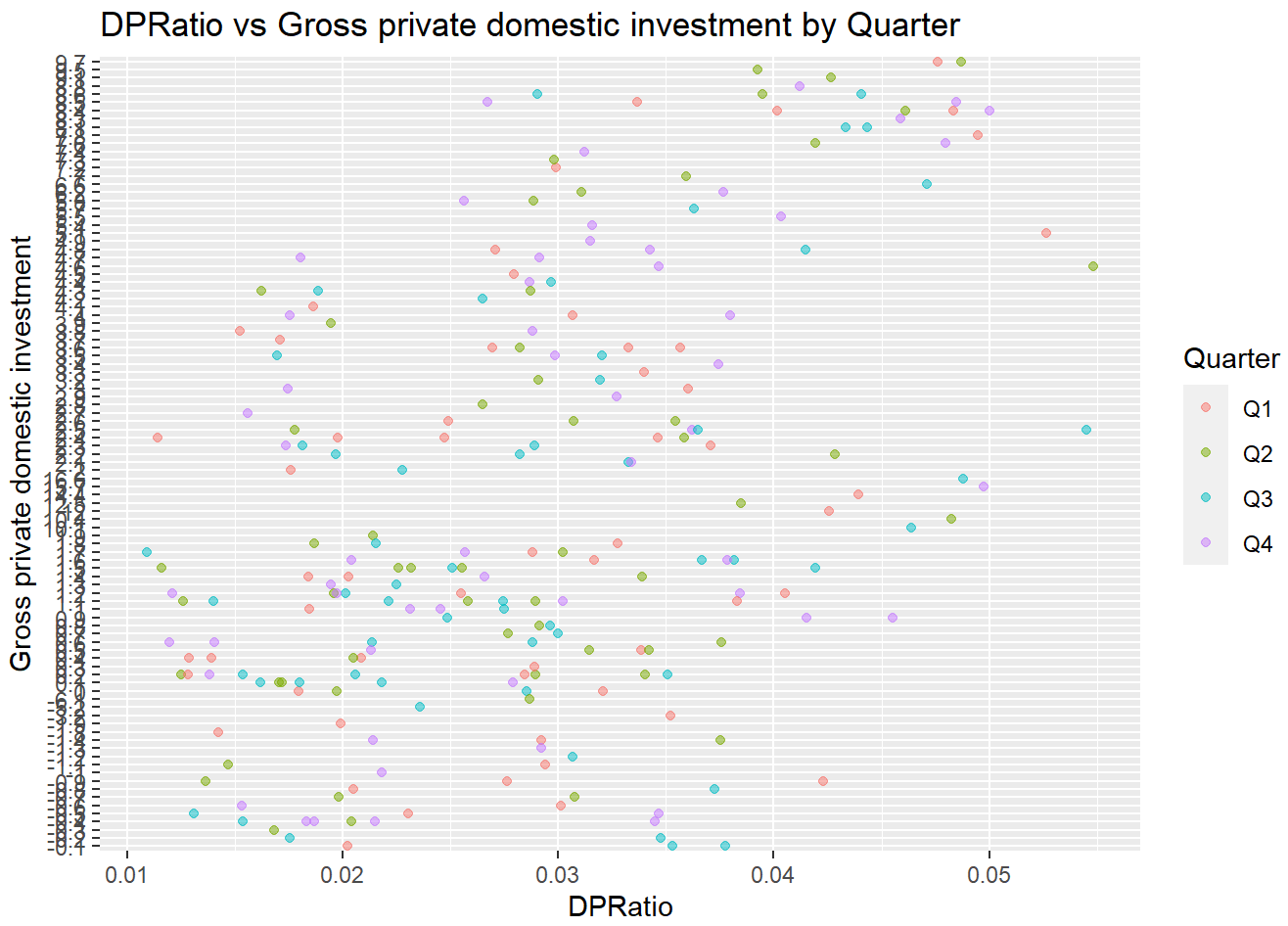
```
#Make scatterplots of relationships, color by quarter
#Start with DPRatio and GNP
 combined_data_DPRatio|>
  ggplot(aes(x = DPRatio, y = Gross.national.product,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "DPRatio", y = "GNP", color = "Quarter",
       title = "DPRatio vs Gross National Product by Quarter")
```



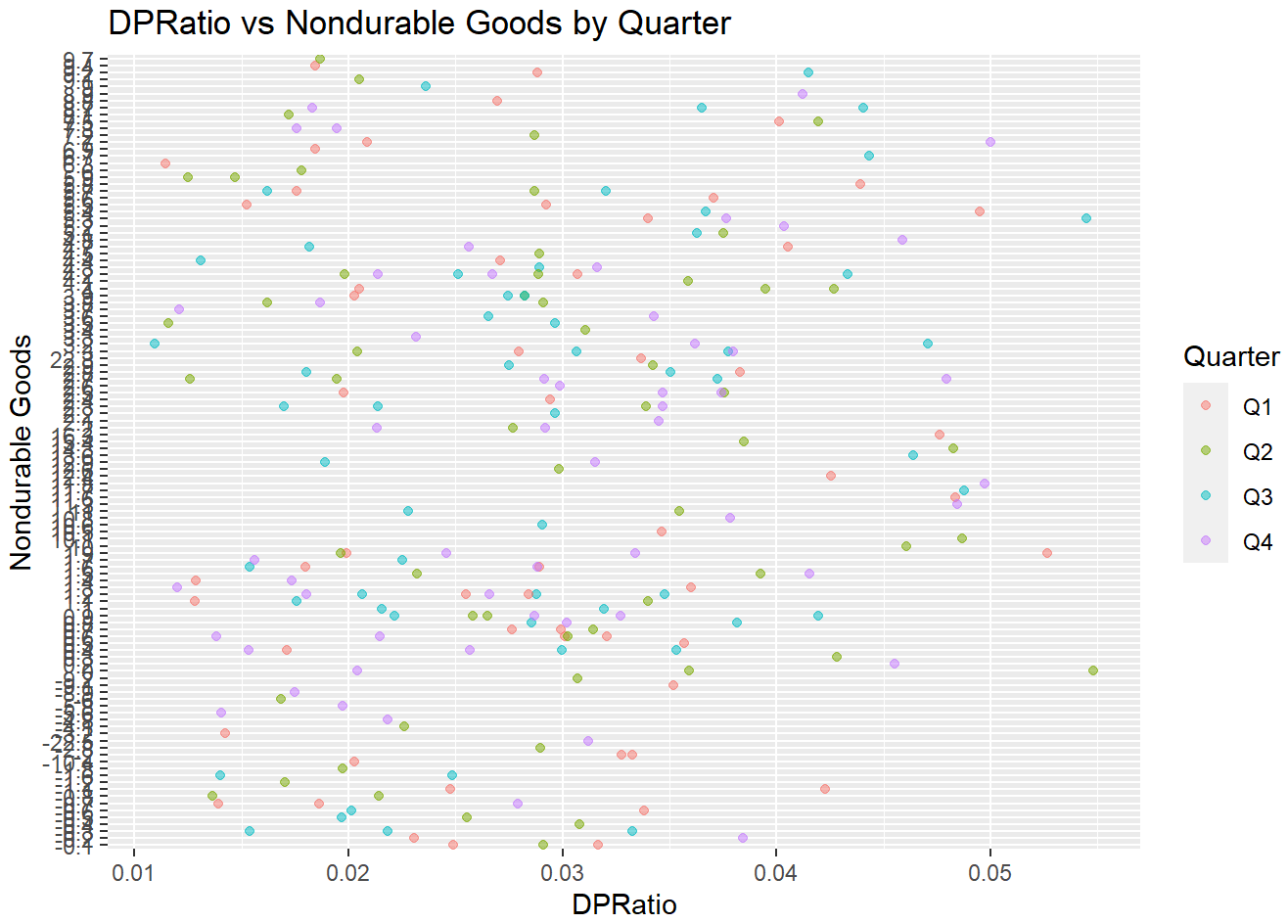

```
#Make scatterplots of relationships, color by quarter
#DPRatio and GDP
 combined_data_DPRatio|>
  ggplot(aes(x = DPRatio, y = GDP,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "DPRatio", y = "GDP", color = "Quarter",
       title = "DPRatio vs Gross Domestic Product by Quarter")
```



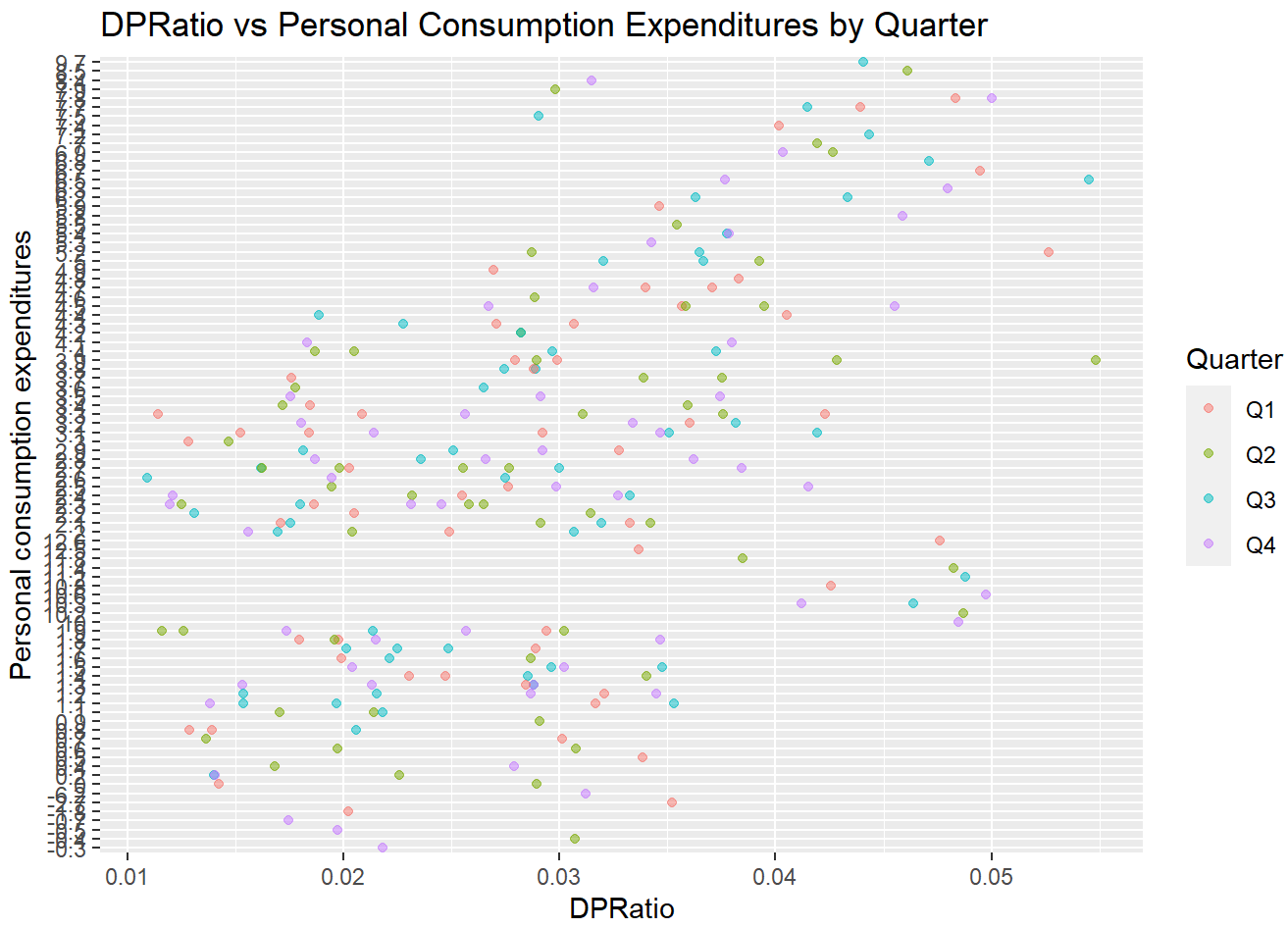

```
#Make scatterplots of relationships, color by quarter
#IPO Gross and GPDI
 combined_data_DPRatio|>
  ggplot(aes(x = DPRatio, y = Gross.private.domestic.investment,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "DPRatio", y = "Gross private domestic investment", color = "Quarter",
       title = "DPRatio vs Gross private domestic investment by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#IPO Gross and Nondurable Goods
 combined_data_DPRatio|>
  ggplot(aes(x = DPRatio, y = Nondurable.goods,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "DPRatio", y = "Nondurable Goods", color = "Quarter",
       title = "DPRatio vs Nondurable Goods by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#IPO Gross and Personal consumption expenditures
 combined_data_DPRatio|>
  ggplot(aes(x = DPRatio, y = Personal.consumption.expenditures,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "DPRatio", y = "Personal consumption expenditures", color = "Quarter",
       title = "DPRatio vs Personal Consumption Expenditures by Quarter")
```



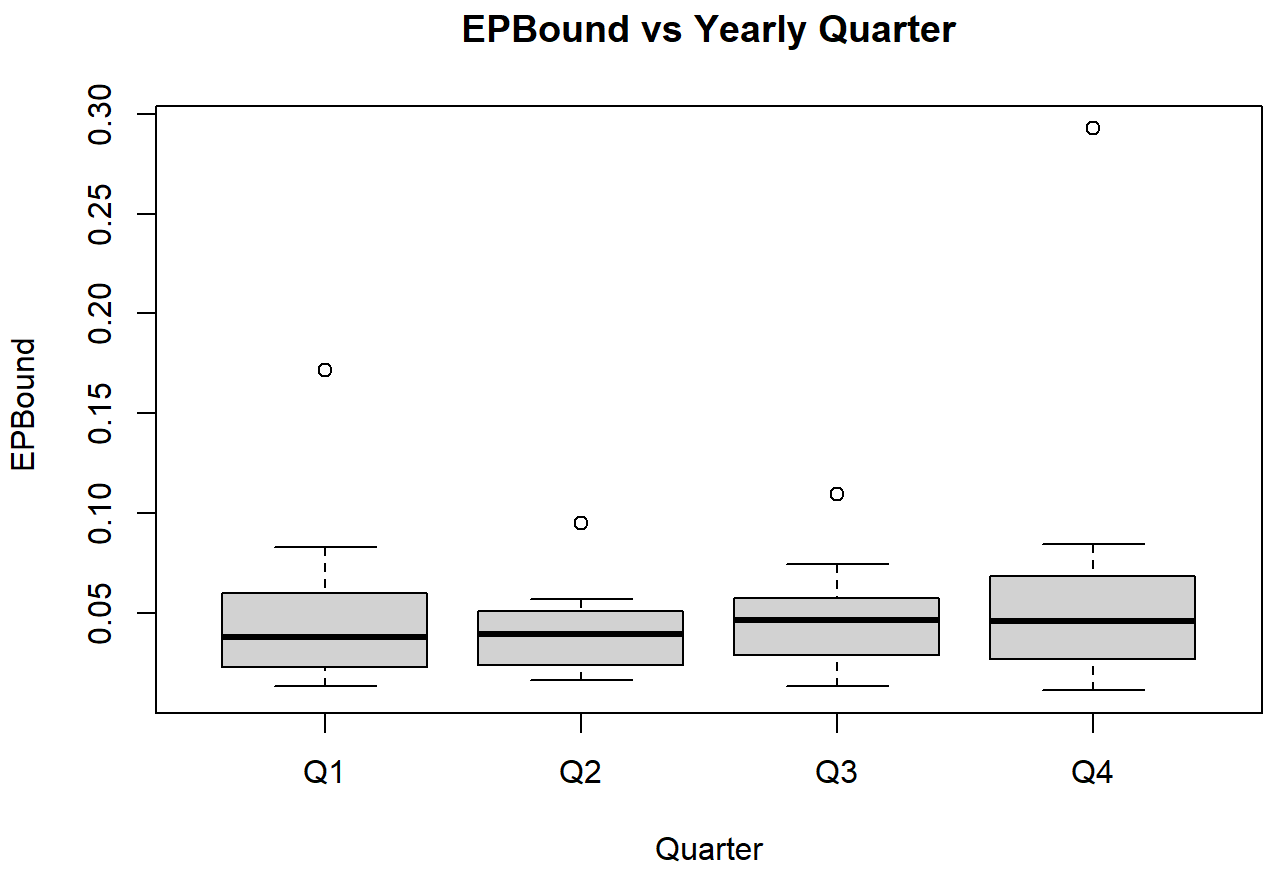

```
#Equity Premium Proxy Measure 3: EPBound, all tests and graphs below
```

```
#Start with an anova test for EPBound

#1: Anova test on the quarters to see if there is difference in the Gross IPO
#by time in the year

EP.aov <- aov(EPBound ~ Quarter, data = combined_data_EPBound)
# Summary
summary(EP.aov)
```

```
##             Df Sum Sq  Mean Sq F value Pr(>F)
## Quarter      3 0.0039 0.001301    0.76  0.521
## Residuals   60 0.1028 0.001713
```

```
#P-value of 0.521. Indicates that there are no significant differences in the
#mean gross EPBound by quarter
```

```
#boxplots of the quarter ~ gross ipo relationship
#Notes: Outliers present for all quarters
#Notes: Q1 and Q2 medians similiar, Q1 and Q4 similiar IQR's
#Notes: No distinguishable oscillation between quarters
boxplot( EPBound~Quarter,data=combined_data_EPBound, main="EPBound vs Yearly Quarter",
   xlab="Quarter", ylab=" EPBound")
```




```
#Run correlation tests between different GDP measures and EPBound
#Years: 1960 to 2015
#Correlation with GNP
cor(combined_data_EPBound$EPBound, as.numeric(combined_data_EPBound$Gross.national.product))
```

```
## [1] -0.5089468
```

```
#Correlation with GDP
cor(combined_data_EPBound$EPBound,  as.numeric(combined_data_EPBound$GDP))
```

```
## [1] -0.5019246
```

```
#Correlation with GPDI
cor(combined_data_EPBound$EPBound, as.numeric(combined_data_EPBound$Gross.private.domestic.inves
tment))
```

```
## [1] -0.05010437
```

```
#Correlation with nondurable goods
cor(combined_data_EPBound$EPBound, as.numeric(combined_data_EPBound$Nondurable.goods))
```

```
## [1] -0.5877241
```

```
#Correlation with personal consumption expenditures
cor(combined_data_EPBound$EPBound, as.numeric(combined_data_EPBound$Personal.consumption.expendi
tures))
```

```
## [1] -0.7171297
```

```
#all negative correlations, personal consumption expenditures at -0.71 indicates strong negative
relationship
```

```
#Linear regression tests
#Start with EPBound and GNP
lmEPBound_GNP = lm(as.numeric(Gross.national.product) ~ EPBound, data = combined_data_EPBound)
summary(lmIPO_GNP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ IPO_Gross,
##     data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3211 -1.6983 -0.5933  0.9027  8.9425
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.957547   0.253435  15.616  < 2e-16 ***
## IPO_Gross   -0.022466   0.007685  -2.923  0.00382 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.423 on 222 degrees of freedom
## Multiple R-squared:  0.03707,    Adjusted R-squared:  0.03273
## F-statistic: 8.546 on 1 and 222 DF,  p-value: 0.003821
```

```
#try with no intercept now
lmEPBound_GNP_NOINT = lm(as.numeric(Gross.national.product) ~ EPBound - 1, data = combined_data_
EPBound)
summary(lmIPO_GNP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ IPO_Gross -
##     1, data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2016 -0.3154  0.9729  3.0280 12.9000
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## IPO_Gross 0.069869   0.007095   9.848   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.502 on 223 degrees of freedom
## Multiple R-squared:  0.3031, Adjusted R-squared:    0.3
## F-statistic: 96.98 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
#Linear regression tests
#EPBound and gdp
lmEPBound_GDP = lm(as.numeric(GDP) ~ EPBound, data = combined_data_EPBound)
summary(lmEPBound_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ EPBound, data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9499 -0.5069 -0.1082  0.6295  1.7462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.5120     0.1557  16.135  < 2e-16 ***
## EPBound     -11.1591     2.4421  -4.569 2.38e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7976 on 62 degrees of freedom
## Multiple R-squared:  0.2519, Adjusted R-squared:  0.2399
## F-statistic: 20.88 on 1 and 62 DF,  p-value: 2.384e-05
```

```
#try with no intercept now (X - 1)
lmEPBound_GDP_NOINT = lm(as.numeric(GDP) ~ EPBound - 1, data = combined_data_EPBound)
summary(lmEPBound_GDP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ EPBound - 1, data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8907  0.3274  1.0950  1.8198  3.8290
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## EPBound   19.105      3.537   5.401 1.07e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.804 on 63 degrees of freedom
## Multiple R-squared:  0.3165, Adjusted R-squared:  0.3056
## F-statistic: 29.17 on 1 and 63 DF,  p-value: 1.075e-06
```

```
#Linear regression tests
#EPBound and Gross private domestic investment
lmEPBound_GDP = lm(as.numeric(Gross.private.domestic.investment) ~ EPBound, data = combined_data
_EPBound)
summary(lmEPBound_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     EPBound, data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8441 -0.9759 -0.3597  1.2007  7.1884
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0008     0.4352   2.299   0.0249 *
## EPBound      -2.6968     6.8271  -0.395   0.6942
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.23 on 62 degrees of freedom
## Multiple R-squared:  0.00251,    Adjusted R-squared:  -0.01358
## F-statistic: 0.156 on 1 and 62 DF,  p-value: 0.6942
```

```
#try with no intercept now (X - 1)
lmEPBound_GPDI_NOINT = lm(as.numeric(Gross.private.domestic.investment) ~ EPBound - 1, data = co
mbined_data_EPBound)
summary(lmEPBound_GPDI_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     EPBound - 1, data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9909 -0.6571 -0.0284  1.6273  4.6609
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## EPBound    9.361      4.518   2.072   0.0424 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.304 on 63 degrees of freedom
## Multiple R-squared:  0.06379,    Adjusted R-squared:  0.04892
## F-statistic: 4.292 on 1 and 63 DF,  p-value: 0.04239
```

```
#Linear regression tests
#EPBound and nondurable goods
lmEPBound_NDG = lm(as.numeric(Nondurable.goods) ~ EPBound, data = combined_data_EPBound)
summary(lmIPO_NDG)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ IPO_Gross, data = combined_data_IPO_Gross)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.7249  -2.4122  -0.3984   2.3449  18.0751
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.22488    0.49209   8.586  1.6e-15 ***
## IPO_Gross   -0.04208    0.01492  -2.820  0.00524 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.704 on 222 degrees of freedom
## Multiple R-squared:  0.03458,    Adjusted R-squared:  0.03024
## F-statistic: 7.953 on 1 and 222 DF,  p-value: 0.005236
```

```
#try with no intercept now (X - 1)
lmEPBound_NDG_NOINT = lm(as.numeric(Nondurable.goods) ~ EPBound - 1, data = combined_data_EPBoun
d)
summary(lmEPBound_NDG_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ EPBound - 1, data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.232   0.615   2.731   5.941  12.912
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## EPBound  -0.9176    11.7006  -0.078    0.938
##
## Residual standard error: 5.968 on 63 degrees of freedom
## Multiple R-squared:  9.761e-05,  Adjusted R-squared:  -0.01577
## F-statistic: 0.00615 on 1 and 63 DF,  p-value: 0.9377
```

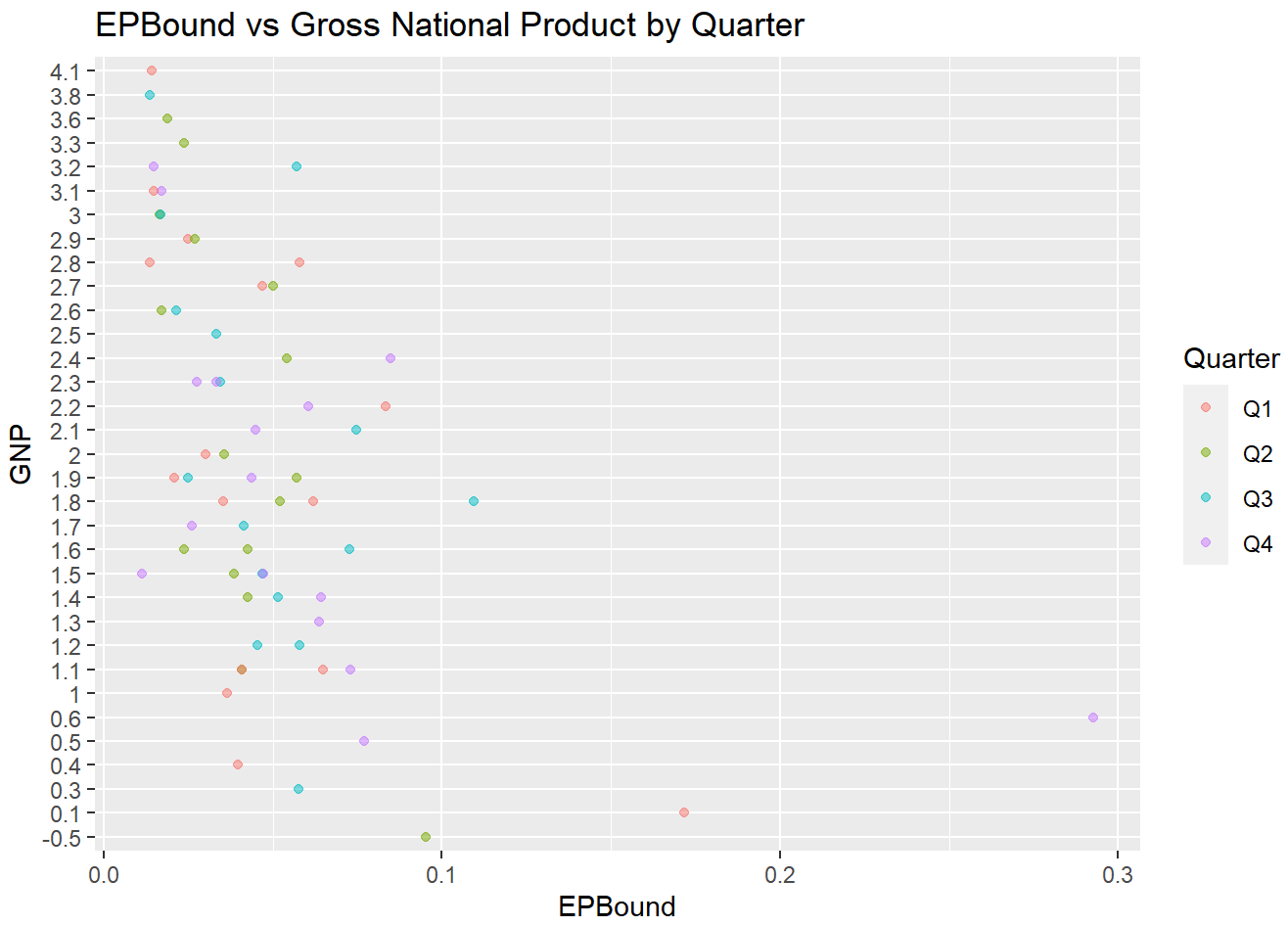
```
#Linear regression tests
#EPBound and personal consumption expenditures
lmEPBound_PCE = lm(as.numeric(Personal.consumption.expenditures) ~ EPBound, data = combined_data
_EPBound)
summary(lmEPBound_PCE)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     EPBound, data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7898 -0.7706  0.0911  0.7471  2.5258
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.4175     0.2273  15.037  < 2e-16 ***
## EPBound     -28.8839     3.5650  -8.102 2.63e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.164 on 62 degrees of freedom
## Multiple R-squared:  0.5143, Adjusted R-squared:  0.5064
## F-statistic: 65.64 on 1 and 62 DF,  p-value: 2.632e-11
```

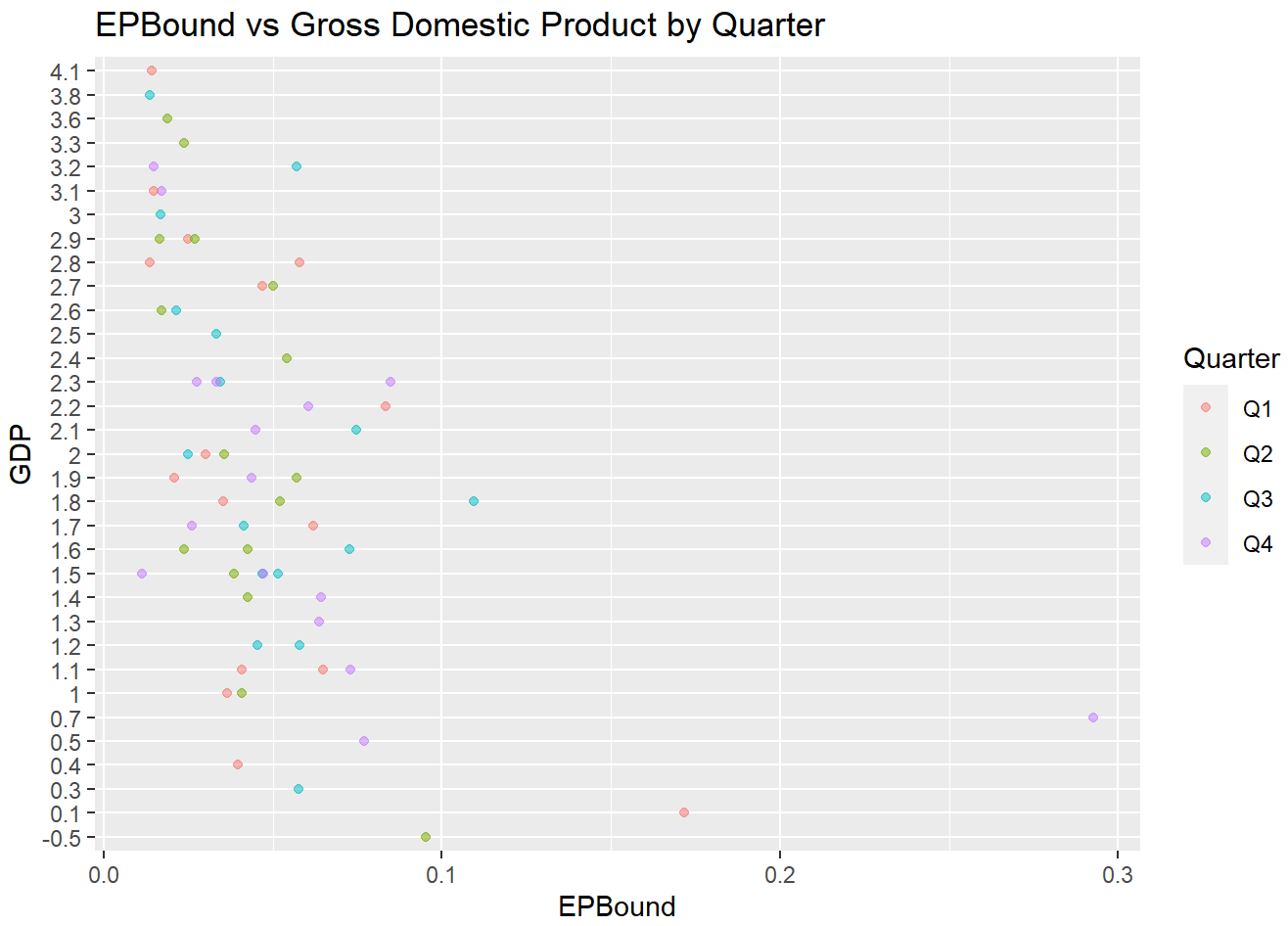
```
#try with no intercept now (X - 1)
lmEPBound_PCE_NOINT = lm(as.numeric(Personal.consumption.expenditures) ~ EPBound - 1, data = com
bined_data_EPBound)
summary(lmEPBound_PCE_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     EPBound - 1, data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7960  0.4897  1.7647  2.4903  4.2325
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## EPBound   12.289      4.882   2.517   0.0144 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.49 on 63 degrees of freedom
## Multiple R-squared:  0.09139,    Adjusted R-squared:  0.07696
## F-statistic: 6.336 on 1 and 63 DF,  p-value: 0.01438
```

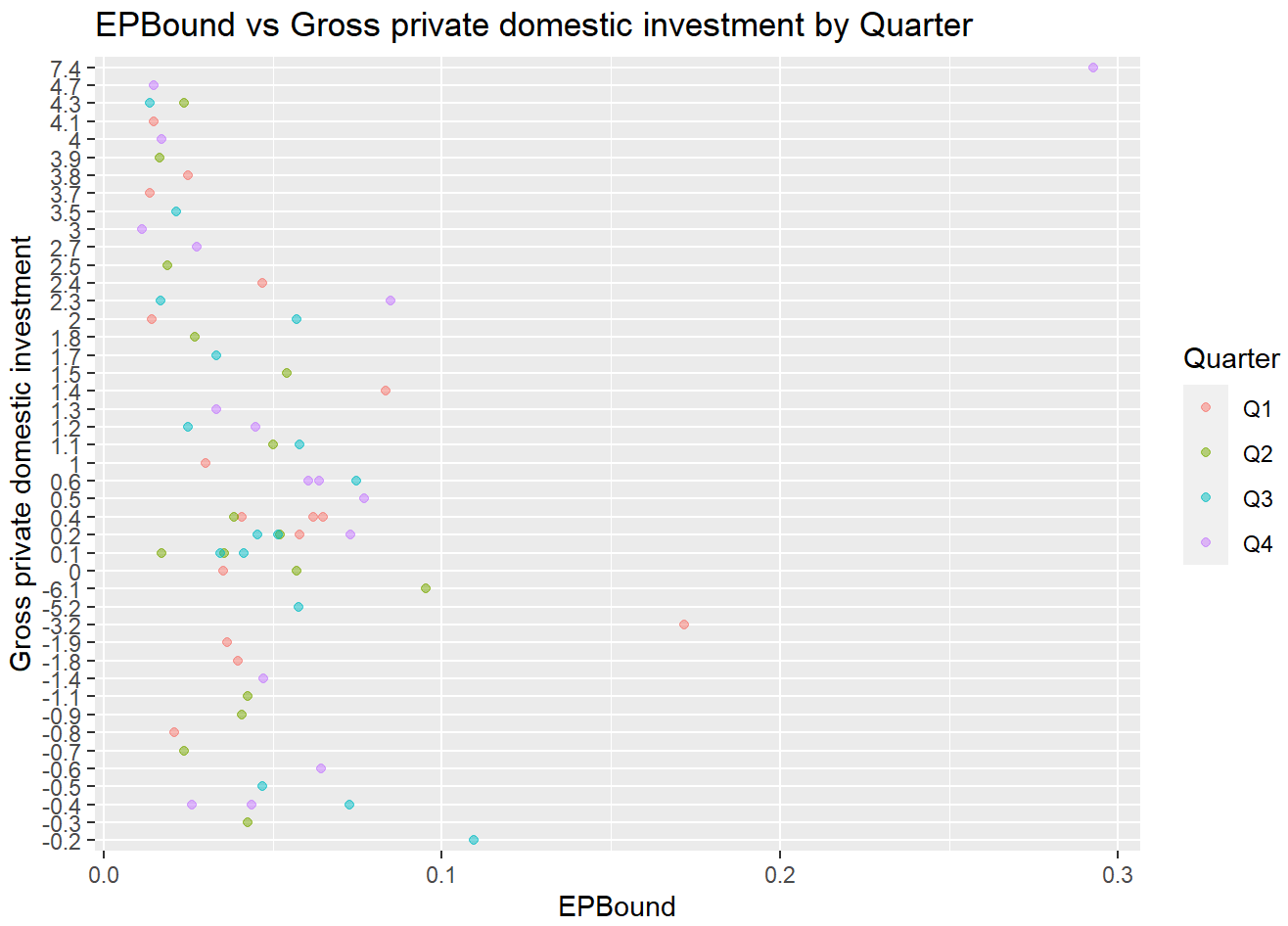
```
#Make scatterplots of relationships, color by quarter
#Start with IPO Gross and GNP
 combined_data_EPBound|>
  ggplot(aes(x = EPBound, y = Gross.national.product,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "EPBound", y = "GNP", color = "Quarter",
       title = "EPBound vs Gross National Product by Quarter")
```



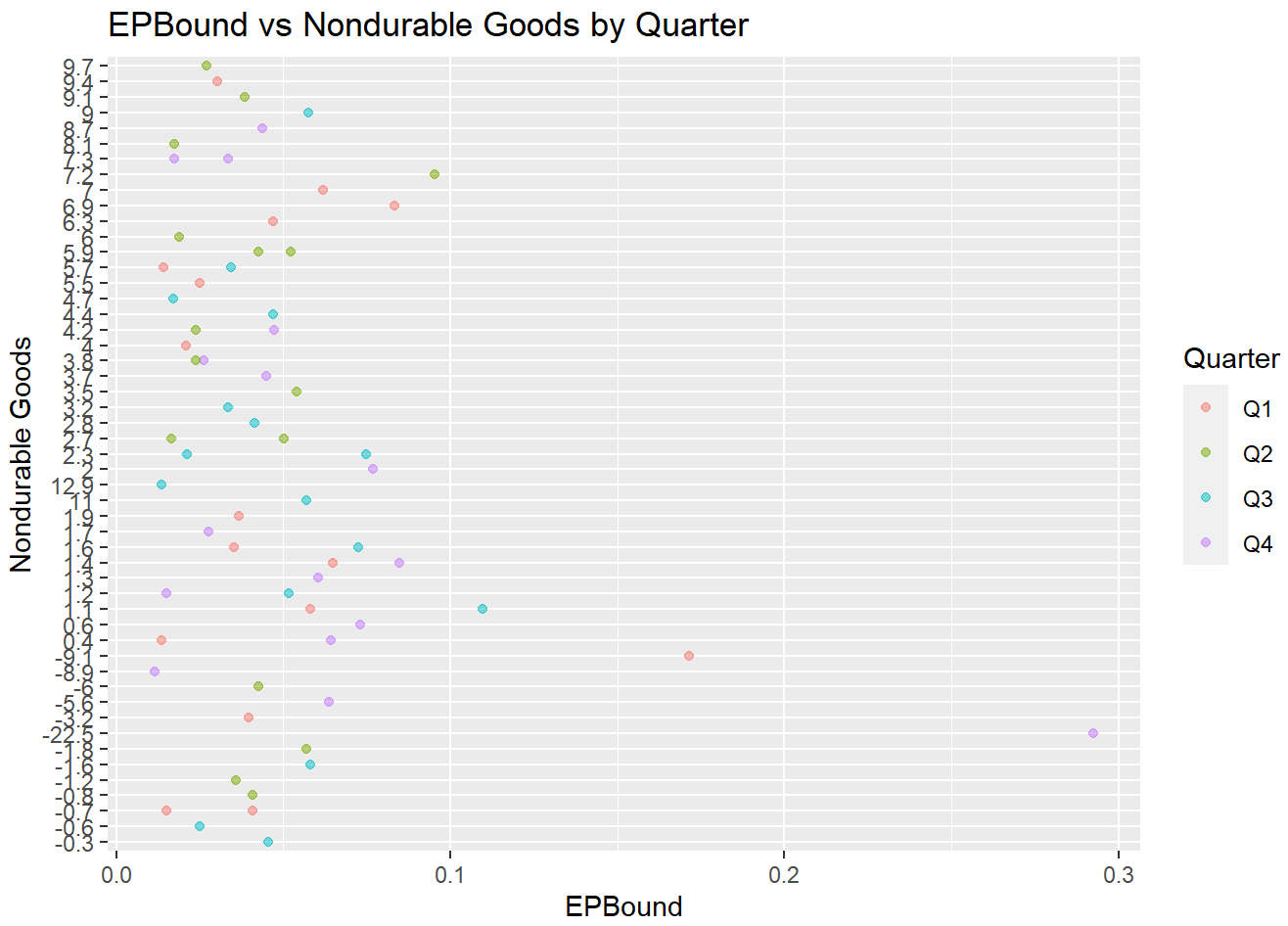

```
#Make scatterplots of relationships, color by quarter
#IPO Gross and GDP
 combined_data_EPBound|>
  ggplot(aes(x = EPBound, y = GDP,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "EPBound", y = "GDP", color = "Quarter",
       title = "EPBound vs Gross Domestic Product by Quarter")
```



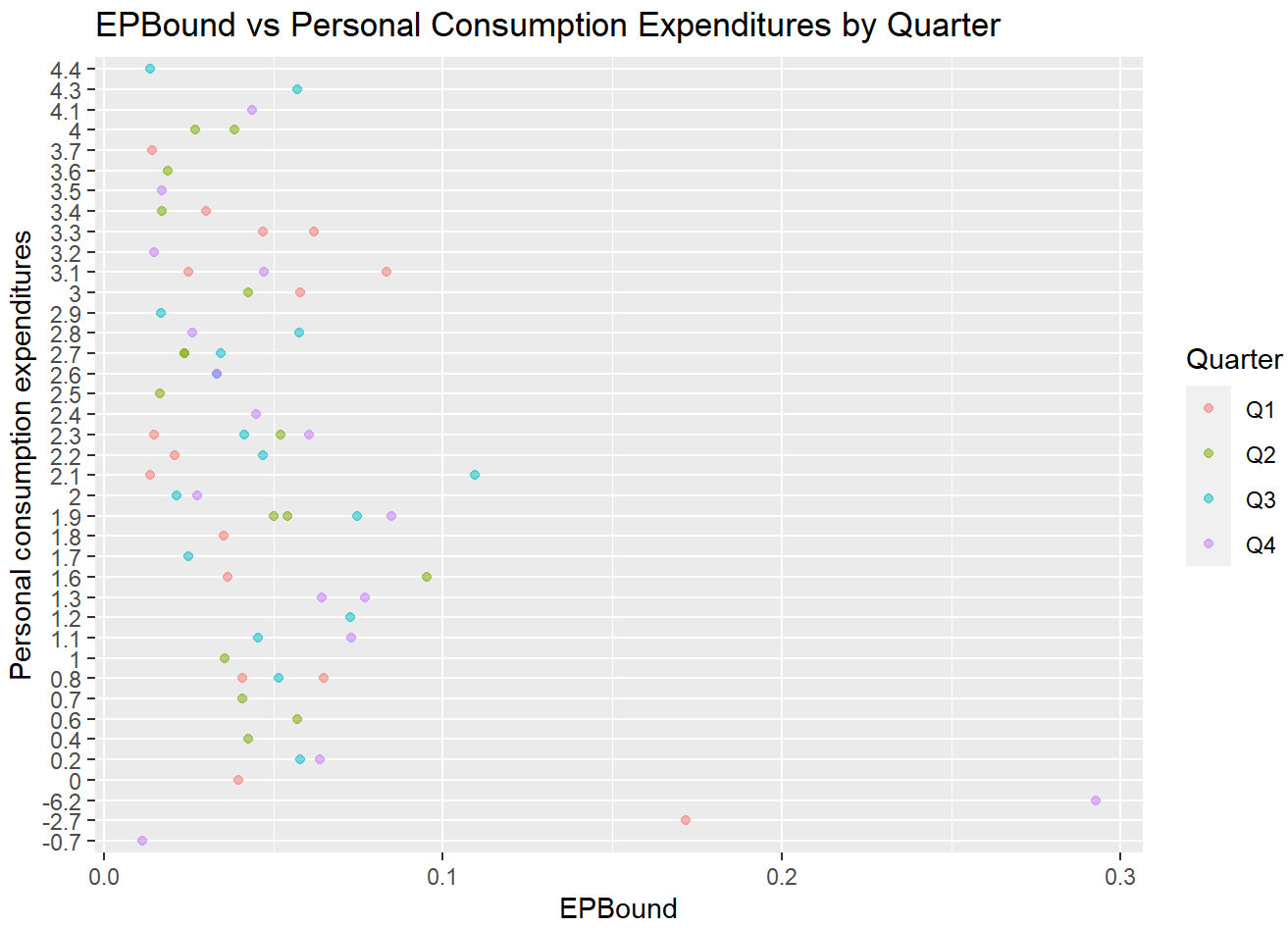

```
#Make scatterplots of relationships, color by quarter
#IPO Gross and GPDI
 combined_data_EPBound|>
  ggplot(aes(x = EPBound, y = Gross.private.domestic.investment,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "EPBound", y = "Gross private domestic investment", color = "Quarter",
       title = "EPBound vs Gross private domestic investment by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#IPO Gross and Nondurable Goods
 combined_data_EPBound|>
  ggplot(aes(x = EPBound, y = Nondurable.goods,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "EPBound", y = "Nondurable Goods", color = "Quarter",
       title = "EPBound vs Nondurable Goods by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#IPO Gross and Personal consumption expenditures
 combined_data_EPBound|>
  ggplot(aes(x = EPBound, y = Personal.consumption.expenditures,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "EPBound", y = "Personal consumption expenditures", color = "Quarter",
       title = "EPBound vs Personal Consumption Expenditures by Quarter")
```



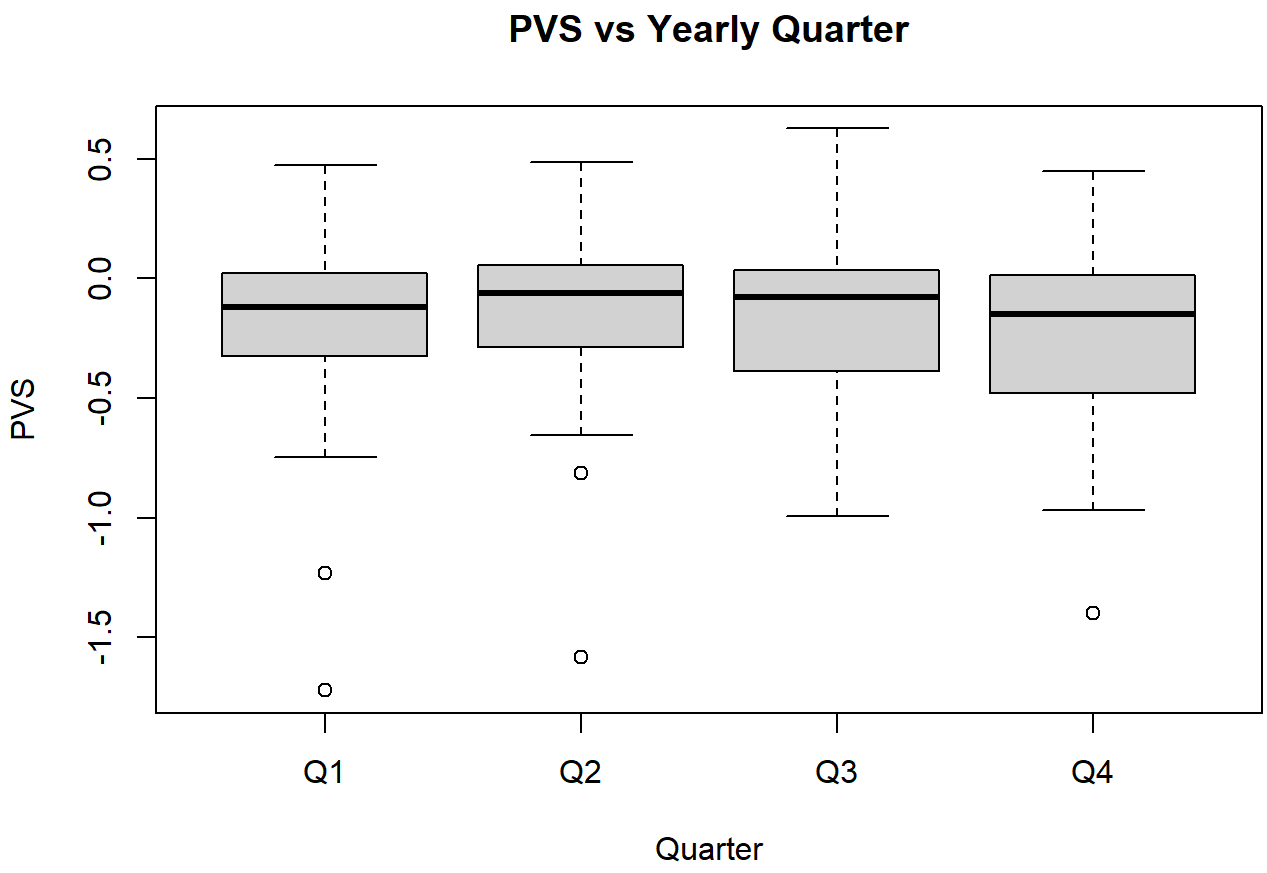

```
#Risk Aversion Measure 1: PVS, all tests and graphs below
```

```
#Start with an anova test for PVS

#1: Anova test on the quarters to see if there is difference in the Gross IPO
#by time in the year

PVS.aov <- aov(as.numeric(PVS) ~ Quarter, data = combined_data_PVS)
# Summary
summary(PVS.aov)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Quarter       3  0.514  0.1714   1.265  0.288
## Residuals   179 24.249  0.1355
```

```
#P-value of 0.288. Indicates that there are no significant differences in the
#mean gross ipo by quarter
```

```
#boxplots of the quarter ~ PVS relationship
#Notes: Outliers present for Q1, Q2, and Q4
#Notes: Q4 has the biggest IQR
#Notes: No real oscillation between quarters
boxplot(as.numeric(PVS) ~Quarter,data=combined_data_PVS, main="PVS vs Yearly Quarter",
   xlab="Quarter", ylab="PVS")
```




```
#Run correlation tests between different GDP measures and PVS
#Years: 1960 to 2015
#Correlation with GNP
cor(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Gross.national.product))
```

```
## [1] 0.0893971
```

```
#Correlation with GDP
cor(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$GDP))
```

```
## [1] 0.0893902
```

```
#Correlation with GPDI
cor(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Gross.private.domestic.inves
tment))
```

```
## [1] 0.00916365
```

```
#Correlation with nondurable goods
cor(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Nondurable.goods))
```

```
## [1] 0.04292936
```

```
#Correlation with personal consumption expenditures
cor(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Personal.consumption.expendi
tures))
```

```
## [1] 0.1224156
```

```
#all positive correlations below 0.15, indicates weak, positive correlation
```

```
#Linear regression tests
#Start with PVS and GNP
lmPVS_GNP = lm(as.numeric(Gross.national.product) ~ as.numeric(PVS), data = combined_data_PVS)
summary(lmIPO_GNP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ IPO_Gross,
##     data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3211 -1.6983 -0.5933  0.9027  8.9425
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.957547   0.253435  15.616  < 2e-16 ***
## IPO_Gross   -0.022466   0.007685  -2.923  0.00382 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.423 on 222 degrees of freedom
## Multiple R-squared:  0.03707,    Adjusted R-squared:  0.03273
## F-statistic: 8.546 on 1 and 222 DF,  p-value: 0.003821
```

```
#try with no intercept now
lmPVS_GNP_NOINT = lm(as.numeric(Gross.national.product) ~ as.numeric(PVS) - 1, data = combined_d
ata_PVS)
summary(lmPVS_GNP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ as.numeric(PVS) -
##     1, data = combined_data_PVS)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.716  1.431  2.212  4.484 12.563
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## as.numeric(PVS)  -3.2927     0.7623  -4.319 2.57e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.209 on 182 degrees of freedom
## Multiple R-squared:  0.09298,    Adjusted R-squared:  0.088
## F-statistic: 18.66 on 1 and 182 DF,  p-value: 2.568e-05
```

```
#Linear regression tests
#gross PVS and gdp
lmPVS_GDP = lm(as.numeric(GDP) ~ as.numeric(PVS), data = combined_data_PVS)
summary(lmPVS_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ as.numeric(PVS), data = combined_data_PVS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7298 -1.6926 -0.9364  0.9332  9.8895
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6848     0.2111  17.459   <2e-16 ***
## as.numeric(PVS)   0.6243     0.5170   1.207    0.229
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.573 on 181 degrees of freedom
## Multiple R-squared:  0.007991,   Adjusted R-squared:  0.00251
## F-statistic: 1.458 on 1 and 181 DF,  p-value: 0.2288
```

```
#try with no intercept now (X - 1)
lmPVS_GDP_NOINT = lm(as.numeric(GDP) ~ as.numeric(PVS) - 1, data = combined_data_PVS)
summary(lmPVS_GDP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ as.numeric(PVS) - 1, data = combined_data_PVS)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.709  1.431  2.211  4.433 12.484
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## as.numeric(PVS)  -3.2881     0.7613  -4.319 2.57e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.204 on 182 degrees of freedom
## Multiple R-squared:  0.09297,    Adjusted R-squared:  0.08799
## F-statistic: 18.65 on 1 and 182 DF,  p-value: 2.571e-05
```

```
#Linear regression tests
#PVS and Gross private domestic investment
lmPVS_GDP = lm(as.numeric(Gross.private.domestic.investment) ~ as.numeric(PVS), data = combined_
data_PVS)
summary(lmPVS_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     as.numeric(PVS), data = combined_data_PVS)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.828 -2.308 -1.162  1.425 13.794
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.87056    0.29780   9.639   <2e-16 ***
## as.numeric(PVS)  0.08995    0.72956   0.123    0.902
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.63 on 181 degrees of freedom
## Multiple R-squared:  8.397e-05,  Adjusted R-squared:  -0.00544
## F-statistic: 0.0152 on 1 and 181 DF,  p-value: 0.902
```

```
#try with no intercept now (X - 1)
lmPVS_GPDI_NOINT = lm(as.numeric(Gross.private.domestic.investment) ~ as.numeric(PVS) - 1, data
= combined_data_PVS)
summary(lmPVS_GPDI_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     as.numeric(PVS) - 1, data = combined_data_PVS)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7859  -0.0663   1.5269   3.8408  14.4812
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## as.numeric(PVS)  -2.9580     0.8066  -3.667 0.000322 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.454 on 182 degrees of freedom
## Multiple R-squared:  0.06881,    Adjusted R-squared:  0.06369
## F-statistic: 13.45 on 1 and 182 DF,  p-value: 0.0003217
```

```
#Linear regression tests
#PVS and nondurable goods
lmPVS_NDG = lm(as.numeric(Nondurable.goods) ~ as.numeric(PVS), data = combined_data_PVS)
summary(lmPVS_NDG)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ as.numeric(PVS),
##     data = combined_data_PVS)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.4076  -2.6086  -0.6685   2.3695  19.2423
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.4903     0.4257   8.199 4.33e-14 ***
## as.numeric(PVS)   0.6029     1.0429   0.578    0.564
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.19 on 181 degrees of freedom
## Multiple R-squared:  0.001843,   Adjusted R-squared:  -0.003672
## F-statistic: 0.3342 on 1 and 181 DF,  p-value: 0.5639
```

```
#try with no intercept now (X - 1)
lmPVS_NDG_NOINT = lm(as.numeric(Nondurable.goods) ~ as.numeric(PVS) - 1, data = combined_data_PV
S)
summary(lmPVS_NDG_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ as.numeric(PVS) -
##     1, data = combined_data_PVS)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.4993   0.1638   2.2977   5.6043  20.0738
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## as.numeric(PVS)   -3.103      1.098  -2.827  0.00522 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.061 on 182 degrees of freedom
## Multiple R-squared:  0.04207,    Adjusted R-squared:  0.03681
## F-statistic: 7.993 on 1 and 182 DF,  p-value: 0.005222
```

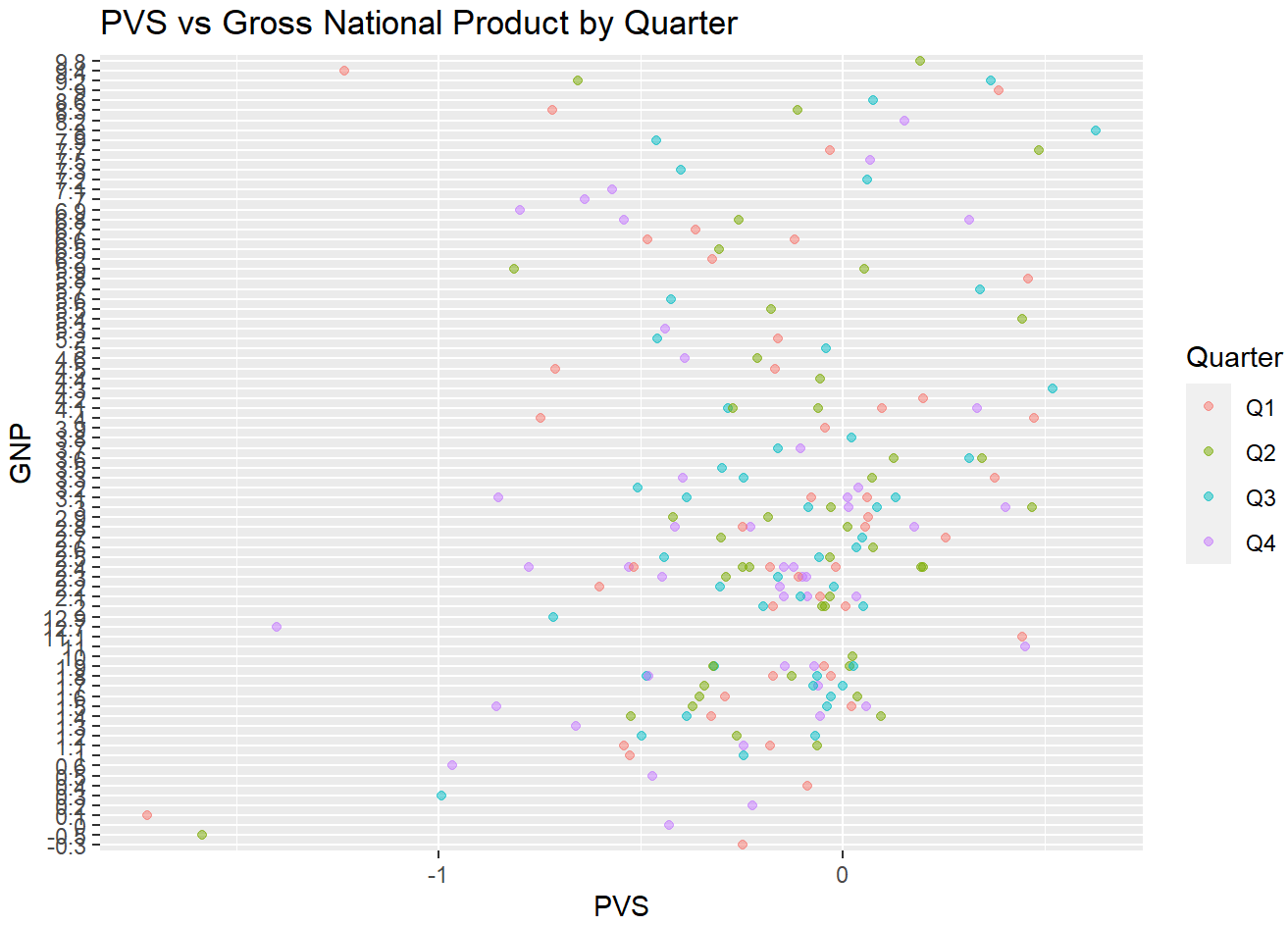
```
#Linear regression tests
#PVS and personal consumption expenditures
lmPVS_PCE = lm(as.numeric(Personal.consumption.expenditures) ~ as.numeric(PVS), data = combined_
data_PVS)
summary(lmPVS_PCE)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     as.numeric(PVS), data = combined_data_PVS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0770 -1.6933 -0.6898  0.9563  9.3861
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.7961     0.2339  16.230   <2e-16 ***
## as.numeric(PVS)   0.9508     0.5730   1.659   0.0988 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.851 on 181 degrees of freedom
## Multiple R-squared:  0.01499,    Adjusted R-squared:  0.009544
## F-statistic: 2.754 on 1 and 181 DF,  p-value: 0.09876
```

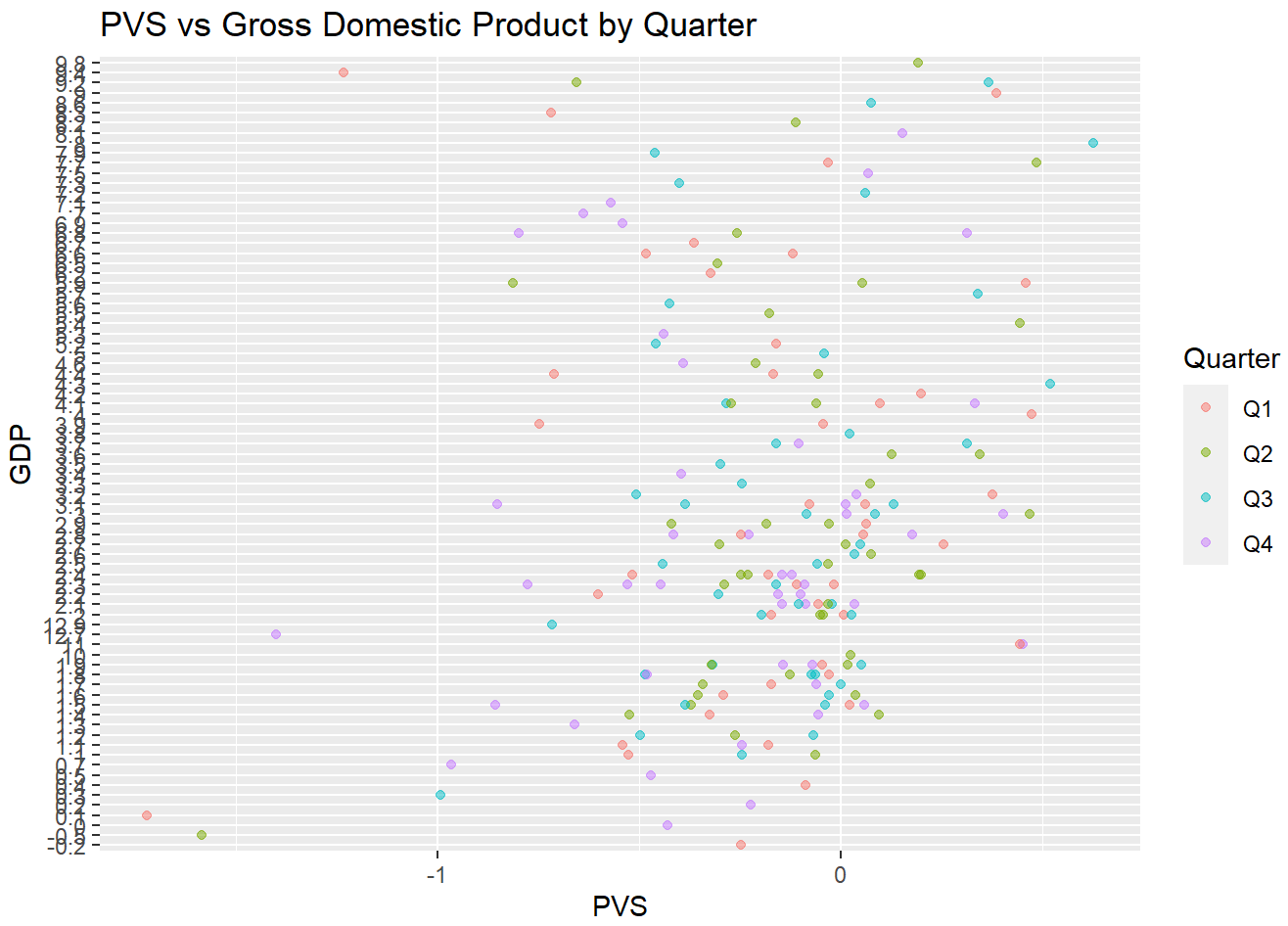
```
#try with no intercept now (X - 1)
lmPVS_PCE_NOINT = lm(as.numeric(Personal.consumption.expenditures) ~ as.numeric(PVS) - 1, data =
combined_data_PVS)
summary(lmPVS_PCE_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     as.numeric(PVS) - 1, data = combined_data_PVS)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.177  1.259  2.565  4.425 13.792
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## as.numeric(PVS)  -3.0798     0.8069  -3.817 0.000185 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.456 on 182 degrees of freedom
## Multiple R-squared:  0.07411,    Adjusted R-squared:  0.06902
## F-statistic: 14.57 on 1 and 182 DF,  p-value: 0.0001852
```

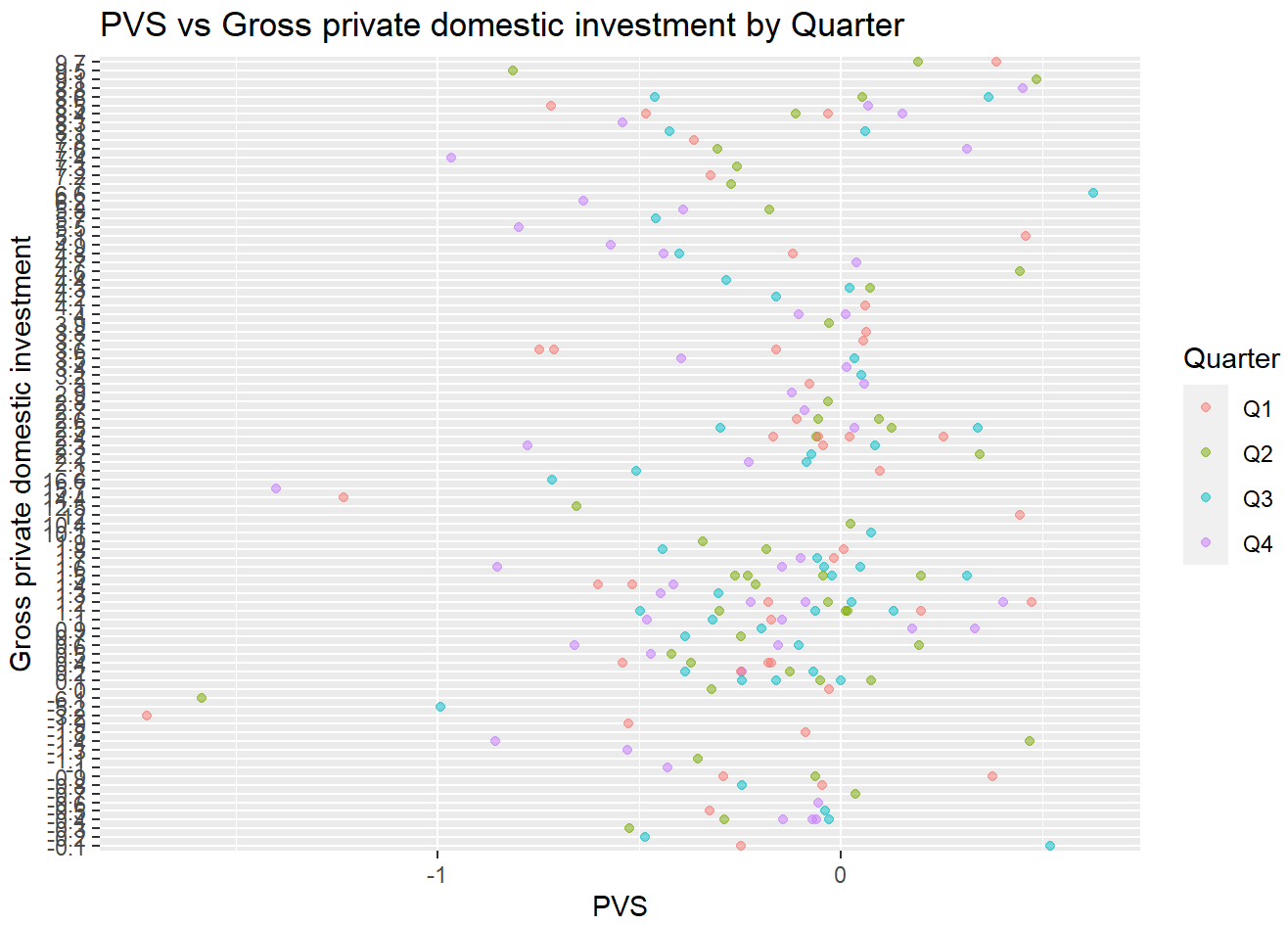
```
#Make scatterplots of relationships, color by quarter
#Start with PVS and GNP
 combined_data_PVS|>
  ggplot(aes(x = as.numeric(PVS), y = Gross.national.product,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "PVS", y = "GNP", color = "Quarter",
       title = "PVS vs Gross National Product by Quarter")
```



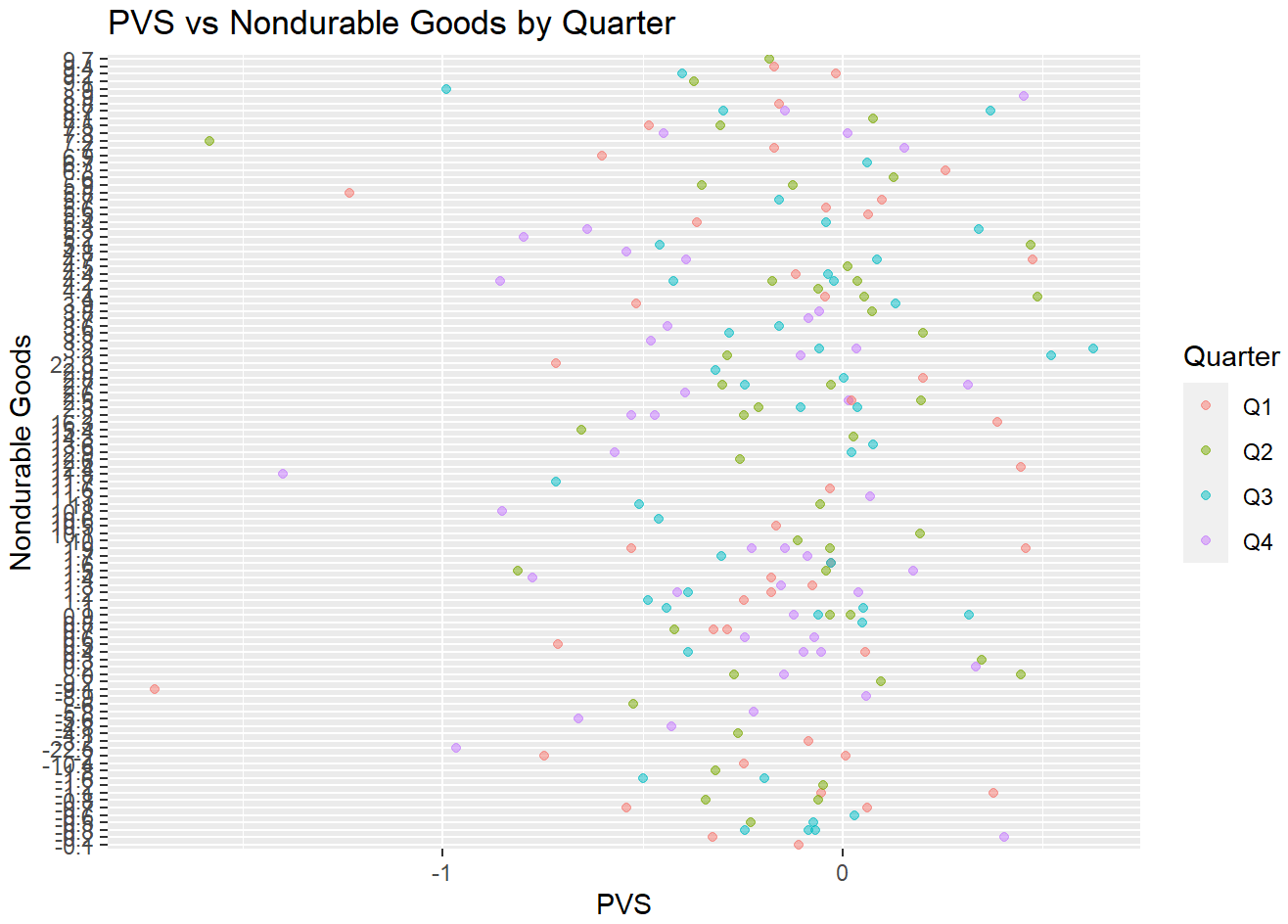

```
#Make scatterplots of relationships, color by quarter
#IPO Gross and GDP
 combined_data_PVS|>
  ggplot(aes(x= as.numeric(PVS), y = GDP,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "PVS", y = "GDP", color = "Quarter",
       title = "PVS vs Gross Domestic Product by Quarter")
```



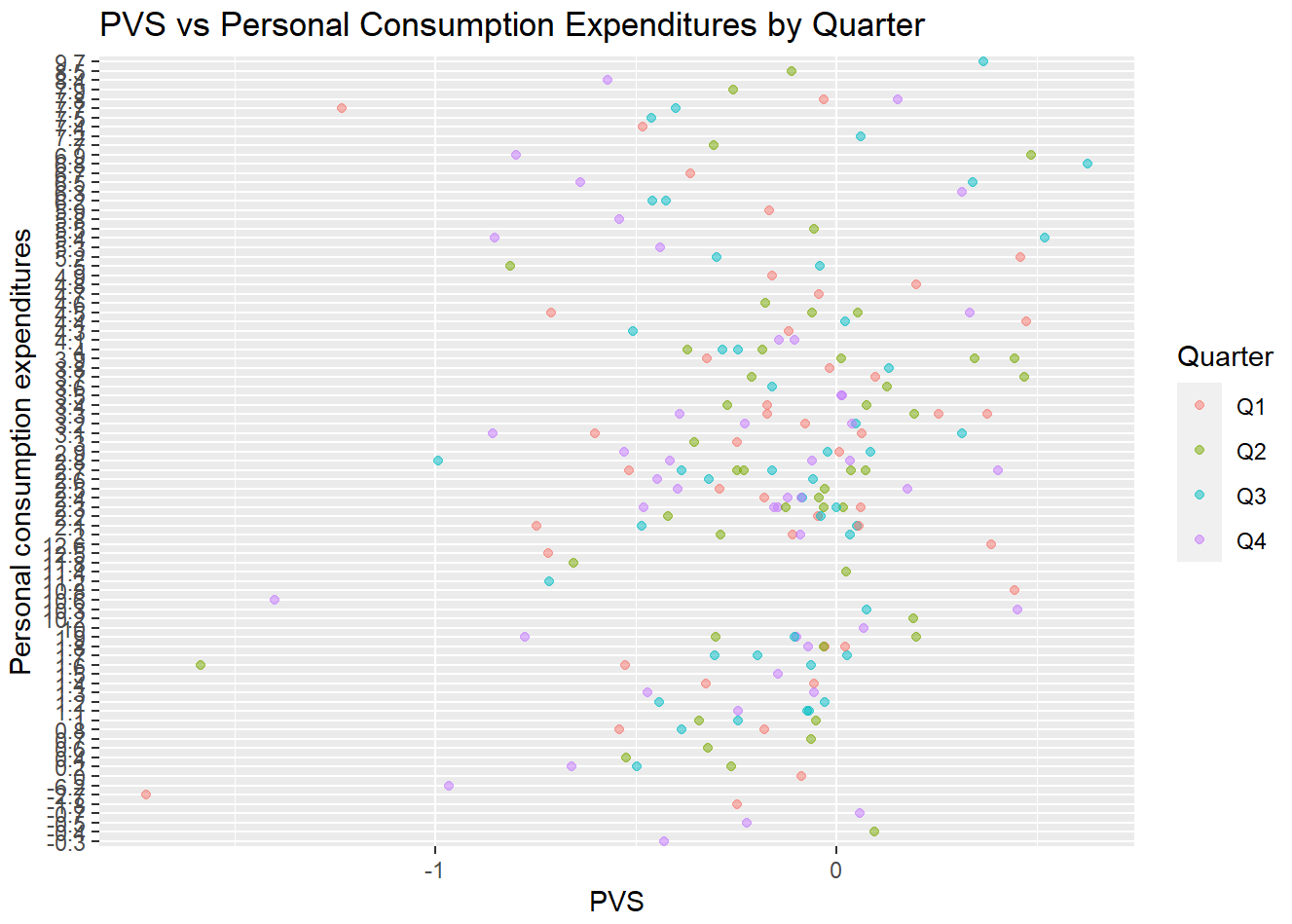

```
#Make scatterplots of relationships, color by quarter
#IPO Gross and GPDI
 combined_data_PVS|>
  ggplot(aes(x = as.numeric(PVS), y = Gross.private.domestic.investment,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "PVS", y = "Gross private domestic investment", color = "Quarter",
       title = "PVS vs Gross private domestic investment by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#IPO Gross and Nondurable Goods
 combined_data_PVS|>
  ggplot(aes(x = as.numeric(PVS), y = Nondurable.goods,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "PVS", y = "Nondurable Goods", color = "Quarter",
       title = "PVS vs Nondurable Goods by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#IPO Gross and Personal consumption expenditures
 combined_data_PVS|>
  ggplot(aes(x = as.numeric(PVS), y = Personal.consumption.expenditures,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "PVS", y = "Personal consumption expenditures", color = "Quarter",
       title = "PVS vs Personal Consumption Expenditures by Quarter")
```



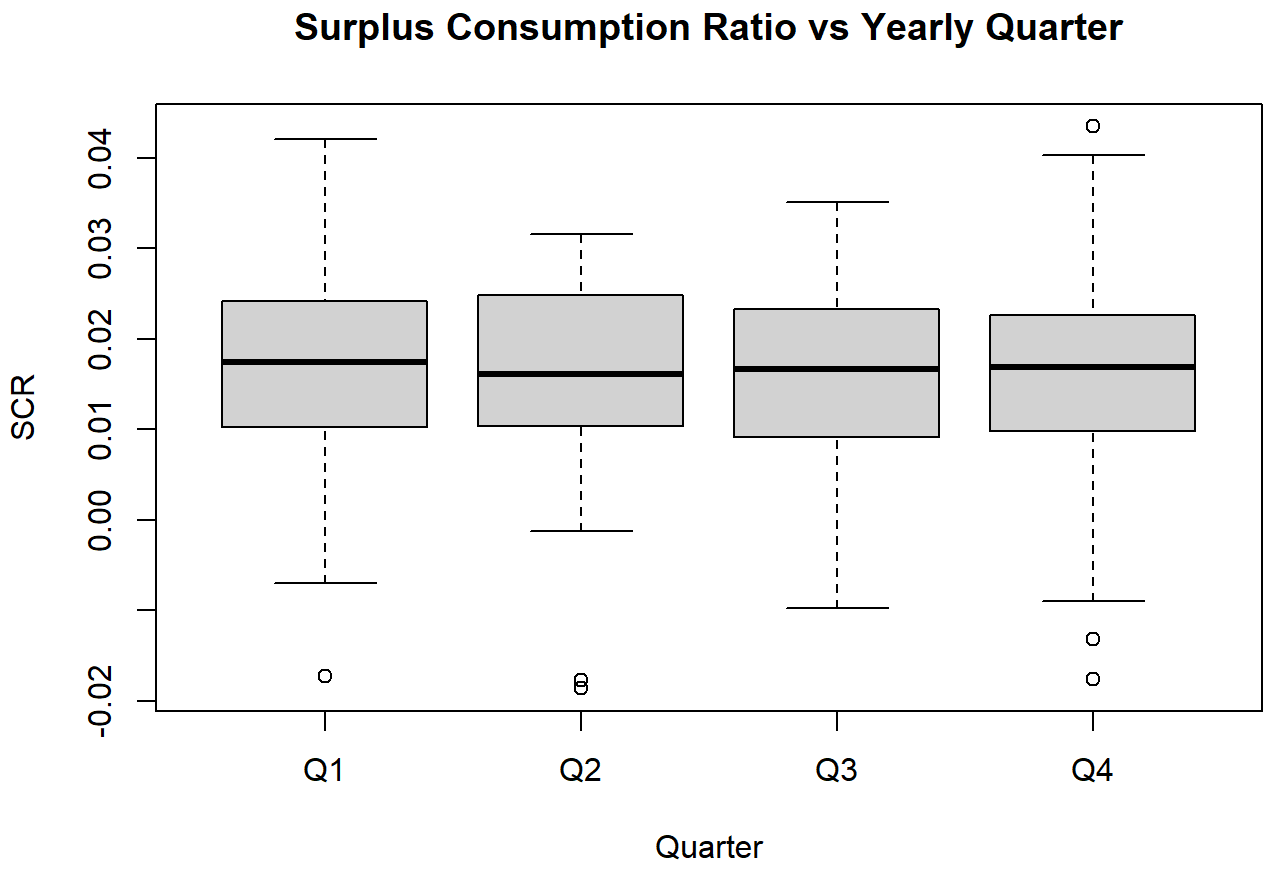

```
#Risk Aversion Measure 2: Surplus Consumption Ratio, all tests and graphs below
```

```
#Start with an anova test for Surplus Consumption Ratio

#1: Anova test on the quarters to see if there is difference in the Surplus Consumption Ratio
#by time in the year

SCR.aov <- aov(SCR ~ Quarter, data = combined_data_SCR)
# Summary
summary(SCR.aov)
```

```
##              Df   Sum Sq   Mean Sq F value Pr(>F)
## Quarter       3 0.000019 0.0000063   0.047  0.986
## Residuals   220 0.029519 0.0001342
```

```
#P-value of 0.986. Indicates that there are no significant differences in the
#mean SCR by quarter
```

```
#boxplots of the quarter ~ gross ipo relationship
#Notes: Outliers present for Q1, Q2 and Q4
#Notes: All quarter IQR's about equal
#Notes: No tengible oscillation between quarters
boxplot(SCR~Quarter,data=combined_data_SCR, main="Surplus Consumption Ratio vs Yearly Quarter",
   xlab="Quarter", ylab="SCR")
```




```
#Run correlation tests between different GDP measures and Surplus Consumption Ratio
#Years: 1960 to 2015
#Correlation with GNP
cor(combined_data_SCR$SCR, as.numeric(combined_data_SCR$Gross.national.product))
```

```
## [1] -0.1164908
```

```
#Correlation with GDP
cor(combined_data_SCR$SCR,  as.numeric(combined_data_SCR$GDP))
```

```
## [1] -0.1173753
```

```
#Correlation with GPDI
cor(combined_data_SCR$SCR, as.numeric(combined_data_SCR$Gross.private.domestic.investment))
```

```
## [1] -0.08171335
```

```
#Correlation with nondurable goods
cor(combined_data_SCR$SCR, as.numeric(combined_data_SCR$Nondurable.goods))
```

```
## [1] -0.06300038
```

```
#Correlation with personal consumption expenditures
cor(combined_data_SCR$SCR, as.numeric(combined_data_SCR$Personal.consumption.expenditures))
```

```
## [1] -0.1123806
```

```
#Purely negative correlations with values strictly less than 0.15 (absolute value) indicating we
ak, negative associations
```

```
#Linear regression tests
#Start with Surplus Consumption Ratio and GNP
lmSCR_GNP = lm(as.numeric(Gross.national.product) ~ SCR, data = combined_data_SCR)
summary(lmIPO_GNP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ IPO_Gross,
##     data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3211 -1.6983 -0.5933  0.9027  8.9425
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.957547   0.253435  15.616  < 2e-16 ***
## IPO_Gross   -0.022466   0.007685  -2.923  0.00382 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.423 on 222 degrees of freedom
## Multiple R-squared:  0.03707,    Adjusted R-squared:  0.03273
## F-statistic: 8.546 on 1 and 222 DF,  p-value: 0.003821
```

```
#try with no intercept now
lmSCR_GNP_NOINT = lm(as.numeric(Gross.national.product) ~ SCR - 1, data = combined_data_SCR)
summary(lmSCR_GNP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ SCR - 1, data = combined_data_SCR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9979 -0.4930  0.5905  2.3685 14.4305
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## SCR   131.18      11.17   11.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.297 on 223 degrees of freedom
## Multiple R-squared:  0.3823, Adjusted R-squared:  0.3795
## F-statistic:   138 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
#Linear regression tests
#Surplus Consumption Ratio and gdp
lmSCR_GDP = lm(as.numeric(GDP) ~ SCR, data = combined_data_SCR)
summary(lmSCR_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ SCR, data = combined_data_SCR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7306 -1.6012 -0.7037  1.0250  9.1627
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7860     0.2811  13.467   <2e-16 ***
## SCR         -25.0970    14.2514  -1.761   0.0796 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.449 on 222 degrees of freedom
## Multiple R-squared:  0.01378,    Adjusted R-squared:  0.009335
## F-statistic: 3.101 on 1 and 222 DF,  p-value: 0.07961
```

```
#try with no intercept now (X - 1)
lmSCR_GDP_NOINT = lm(as.numeric(GDP) ~ SCR - 1, data = combined_data_SCR)
summary(lmSCR_GDP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ SCR - 1, data = combined_data_SCR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9883 -0.4898  0.5924  2.3725 14.4276
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## SCR   130.96      11.16   11.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.294 on 223 degrees of freedom
## Multiple R-squared:  0.3819, Adjusted R-squared:  0.3791
## F-statistic: 137.8 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
#Linear regression tests
#Surplus Consumption Ratio and Gross private domestic investment
lmSCR_GDPI = lm(as.numeric(Gross.private.domestic.investment) ~ SCR, data = combined_data_SCR)
summary(lmSCR_GDPI)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     SCR, data = combined_data_SCR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5753 -2.3281 -0.9915  1.6053 13.6055
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0420     0.3951   7.700 4.45e-13 ***
## SCR         -24.4646    20.0269  -1.222    0.223
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.442 on 222 degrees of freedom
## Multiple R-squared:  0.006677,   Adjusted R-squared:  0.002203
## F-statistic: 1.492 on 1 and 222 DF,  p-value: 0.2232
```

```
#try with no intercept now (X - 1)
lmSCR_GDPI_NOINT = lm(as.numeric(Gross.private.domestic.investment) ~ SCR - 1, data = combined_d
ata_SCR)
summary(lmSCR_GDPI_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     SCR - 1, data = combined_data_SCR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6071 -1.3510  0.0591  1.9711 17.0313
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## SCR   100.92      13.09   7.708 4.19e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.866 on 223 degrees of freedom
## Multiple R-squared:  0.2104, Adjusted R-squared:  0.2068
## F-statistic: 59.41 on 1 and 223 DF,  p-value: 4.188e-13
```

```
#Linear regression tests
#Surplus Consumption Ratio and nondurable goods
lmSCR_NDG = lm(as.numeric(Nondurable.goods) ~ SCR, data = combined_data_SCR)
summary(lmSCR_NDG)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ SCR, data = combined_data_SCR)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.5370  -2.4019  -0.4803   2.1907  18.7161
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.5766     0.5484   6.521 4.64e-10 ***
## SCR         -26.1500    27.8028  -0.941    0.348
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.778 on 222 degrees of freedom
## Multiple R-squared:  0.003969,   Adjusted R-squared:  -0.0005176
## F-statistic: 0.8846 on 1 and 222 DF,  p-value: 0.348
```

```
#try with no intercept now (X - 1)
lmSCR_NDG_NOINT = lm(as.numeric(Nondurable.goods) ~ SCR - 1, data = combined_data_SCR)
summary(lmSCR_NDG_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ SCR - 1, data = combined_data_SCR)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.3648  -1.7179   0.4287   2.9980  22.3342
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## SCR   121.27      17.63    6.88 5.98e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.204 on 223 degrees of freedom
## Multiple R-squared:  0.1751, Adjusted R-squared:  0.1714
## F-statistic: 47.33 on 1 and 223 DF,  p-value: 5.98e-11
```

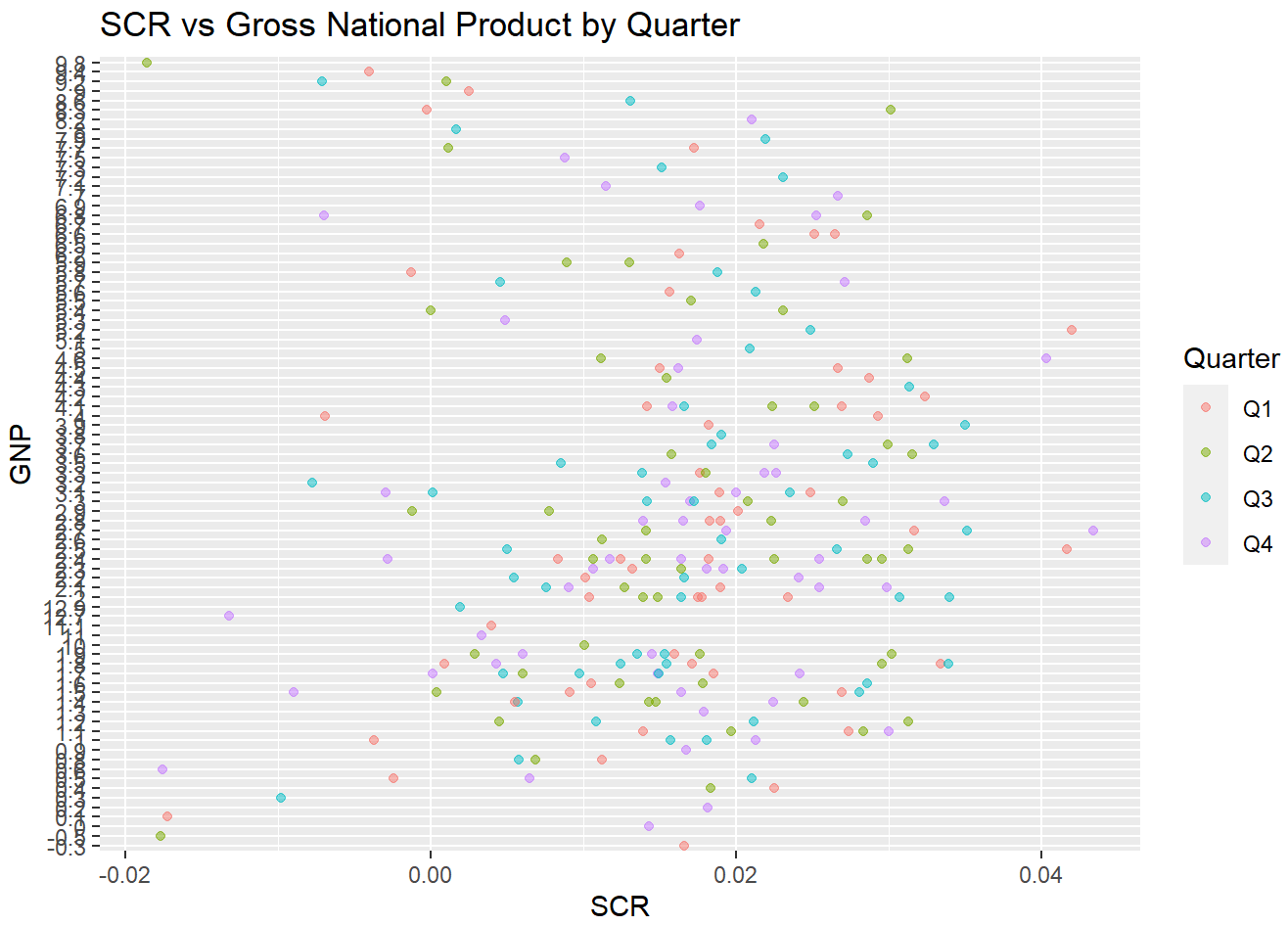
```
#Linear regression tests
#Surplus Consumption Ratio and personal consumption expenditures
lmSCR_PCE = lm(as.numeric(Personal.consumption.expenditures) ~ SCR, data = combined_data_SCR)
summary(lmSCR_PCE)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     SCR, data = combined_data_SCR)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4752  -1.6519  -0.5614   1.2236   8.8575
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.8090     0.3099  12.290   <2e-16 ***
## SCR         -26.4759    15.7117  -1.685   0.0934 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.7 on 222 degrees of freedom
## Multiple R-squared:  0.01263,    Adjusted R-squared:  0.008182
## F-statistic:  2.84 on 1 and 222 DF,  p-value: 0.09337
```

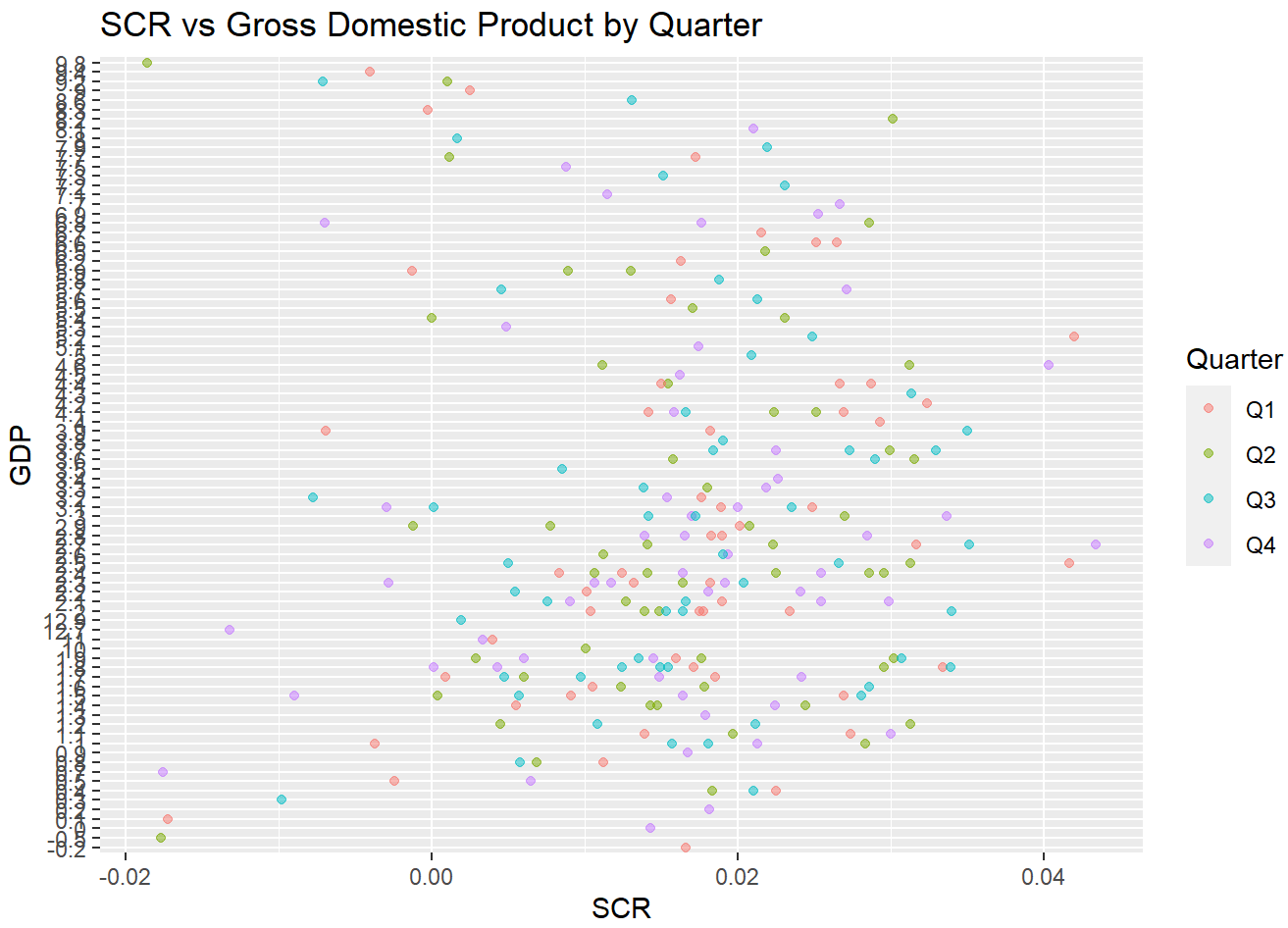
```
#try with no intercept now (X - 1)
lmSCR_PCE_NOINT = lm(as.numeric(Personal.consumption.expenditures) ~ SCR - 1, data = combined_da
ta_SCR)
summary(lmSCR_PCE_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     SCR - 1, data = combined_data_SCR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4696 -0.7172  0.5715  2.4270 12.6299
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## SCR   130.53      11.83   11.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.493 on 223 degrees of freedom
## Multiple R-squared:  0.3532, Adjusted R-squared:  0.3503
## F-statistic: 121.8 on 1 and 223 DF,  p-value: < 2.2e-16
```

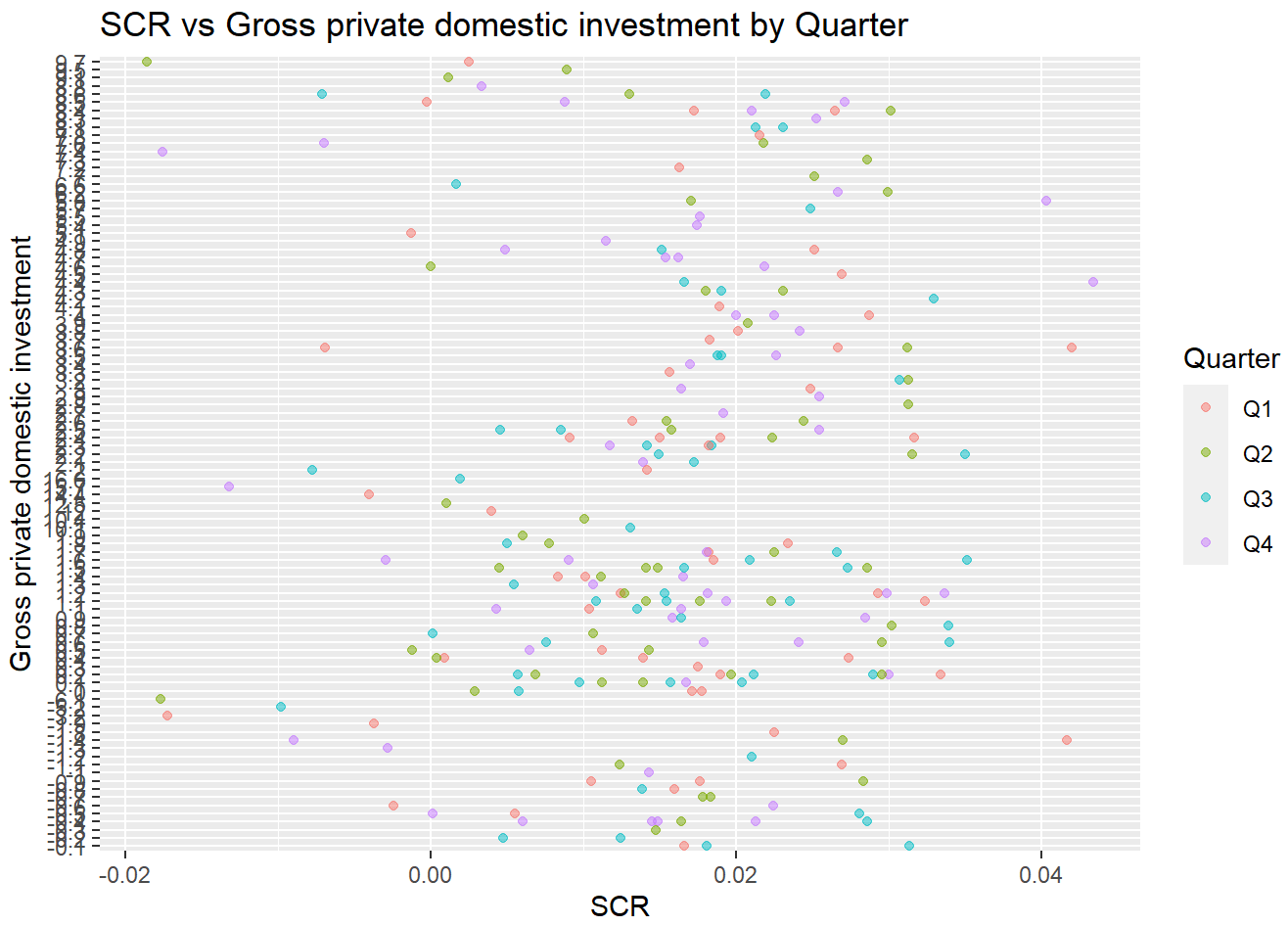
```
#Make scatterplots of relationships, color by quarter
#Start with SCR and GNP
 combined_data_SCR|>
  ggplot(aes(x = SCR, y = Gross.national.product,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "SCR", y = "GNP", color = "Quarter",
       title = "SCR vs Gross National Product by Quarter")
```



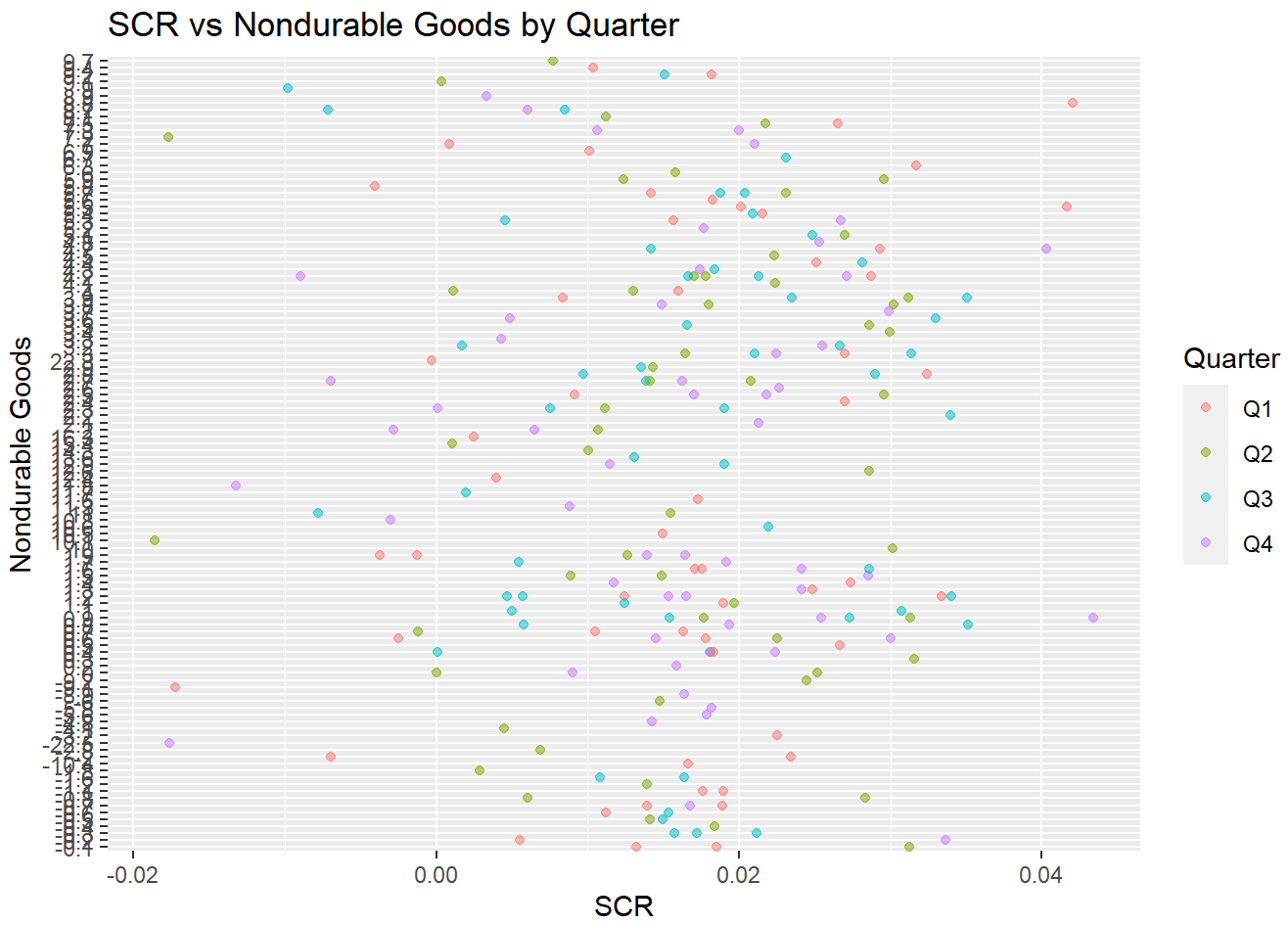

```
#Make scatterplots of relationships, color by quarter
#SCR and GDP
 combined_data_SCR|>
  ggplot(aes(x = SCR, y = GDP,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "SCR", y = "GDP", color = "Quarter",
       title = "SCR vs Gross Domestic Product by Quarter")
```



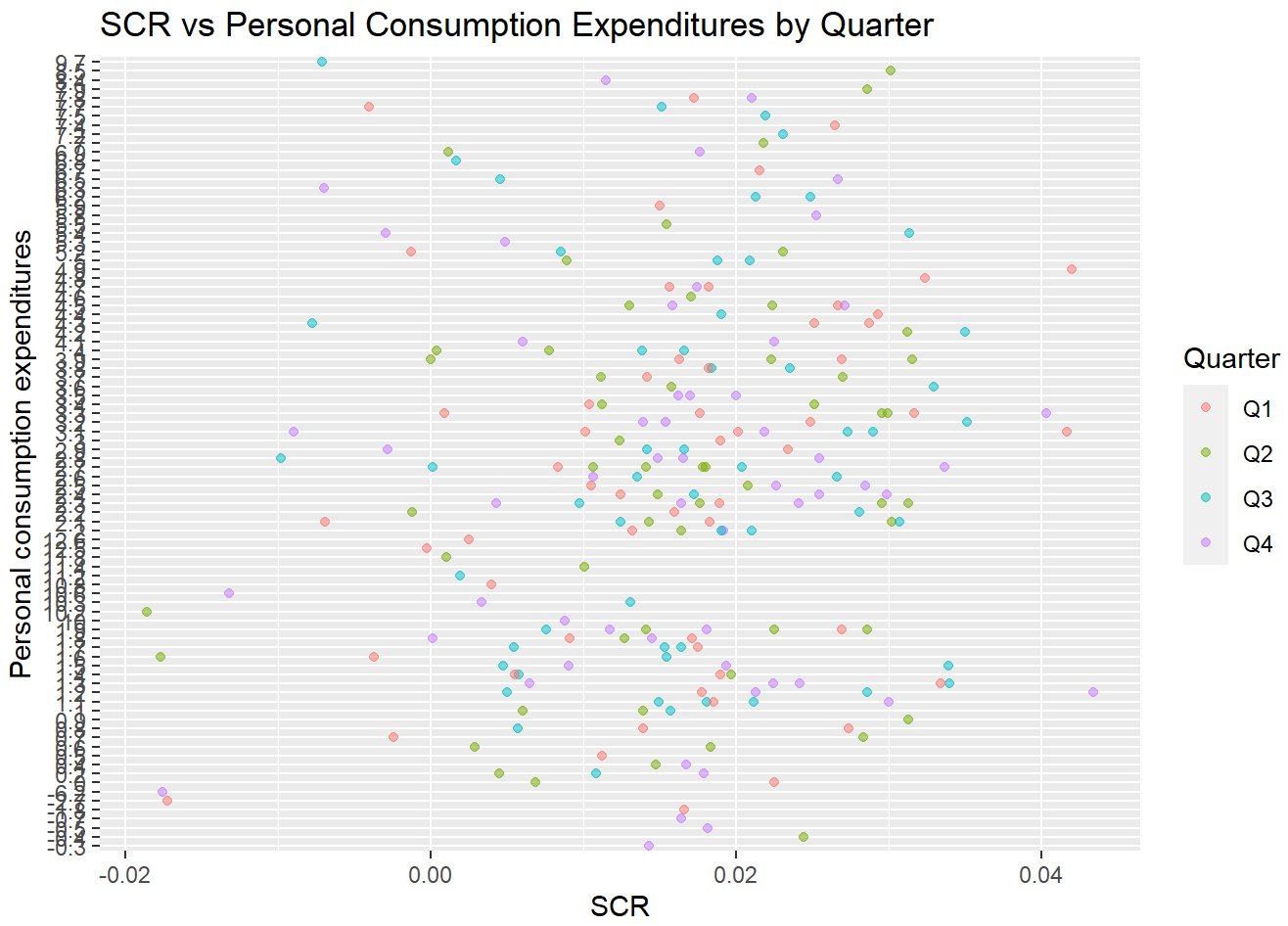

```
#Make scatterplots of relationships, color by quarter
#SCR and GPDI
 combined_data_SCR|>
  ggplot(aes(x = SCR, y = Gross.private.domestic.investment,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "SCR", y = "Gross private domestic investment", color = "Quarter",
       title = "SCR vs Gross private domestic investment by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#SCR and Nondurable Goods
 combined_data_SCR|>
  ggplot(aes(x = SCR, y = Nondurable.goods,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "SCR", y = "Nondurable Goods", color = "Quarter",
       title = "SCR vs Nondurable Goods by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#SCR and Personal consumption expenditures
 combined_data_SCR|>
  ggplot(aes(x = SCR, y = Personal.consumption.expenditures,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "SCR", y = "Personal consumption expenditures", color = "Quarter",
       title = "SCR vs Personal Consumption Expenditures by Quarter")
```




```
#Equity Premium Proxy Measure 3: Aggregate Risk Aversion, all tests and graphs below
rm(combined_data_Agreggate_Risk_Aversion)
```

```
## Warning in rm(combined_data_Agreggate_Risk_Aversion): object
## 'combined_data_Agreggate_Risk_Aversion' not found
```

```
#Start with an anova test for Aggregate Risk Aversion

#1: Anova test on the quarters to see if there is difference in the Aggregate Risk Aversion
#by time in the year

Aggregate_Risk_Aversion.aov <- aov(Aggregate_Risk_Aversion ~ Quarter, data = combined_data_Aggre
gate_Risk_Aversion)
# Summary
summary(Aggregate_Risk_Aversion.aov)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Quarter      3   0.28  0.0935   0.094  0.963
## Residuals   80  79.42  0.9928
```

```
#P-value of 0.963 indicates equal means approximately for risk aversion across quarters
```

```
#Run correlation tests between different GDP measures and Aggregate Risk Aversion
#1990 to 2010
#Correlation with GNP
cor(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion, as.numeric(combined_data_Aggr
egate_Risk_Aversion$Gross.national.product))
```

```
## [1] 0.006625511
```

```
#Correlation with GDP
cor(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion,  as.numeric(combined_data_Agg
regate_Risk_Aversion$GDP))
```

```
## [1] 0.004114272
```

```
#Correlation with GPDI
cor(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion, as.numeric(combined_data_Aggr
egate_Risk_Aversion$Gross.private.domestic.investment))
```

```
## [1] -0.1900766
```

```
#Correlation with nondurable goods
cor(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion, as.numeric(combined_data_Aggr
egate_Risk_Aversion$Nondurable.goods))
```

```
## [1] -0.06534602
```

```
#Correlation with personal consumption expenditures
cor(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion, as.numeric(combined_data_Aggr
egate_Risk_Aversion$Personal.consumption.expenditures))
```

```
## [1] 0.01602137
```

```
# Mix of positive and negative correlations but all below 0.2 (absolute value) so relatively wea
k associations
```

```
#Linear regression tests
#Start with Aggregate Risk Aversion and GNP
lmAggregate_Risk_Aversion_GNP = lm(as.numeric(Gross.national.product) ~ Aggregate_Risk_Aversion,
data = combined_data_Aggregate_Risk_Aversion)
summary(lmAggregate_Risk_Aversion_GNP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ Aggregate_Risk_Aversion,
##     data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.66479 -0.55517 -0.05227  0.57125  2.44024
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2.153600   0.102628   20.98   <2e-16 ***
## Aggregate_Risk_Aversion 0.006321   0.105356    0.06    0.952
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9406 on 82 degrees of freedom
## Multiple R-squared:  4.39e-05,   Adjusted R-squared:  -0.01215
## F-statistic: 0.0036 on 1 and 82 DF,  p-value: 0.9523
```

```
#try with no intercept now
lmAggregate_Risk_Aversion_GNP_NOINT = lm(as.numeric(Gross.national.product) ~ Aggregate_Risk_Ave
rsion - 1, data = combined_data_Aggregate_Risk_Aversion)
summary(lmAggregate_Risk_Aversion_GNP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ Aggregate_Risk_Aversion -
##     1, data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4927  1.5983  2.0991  2.7231  4.6040
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## Aggregate_Risk_Aversion -0.004103   0.264302  -0.016    0.988
##
## Residual standard error: 2.36 on 83 degrees of freedom
## Multiple R-squared:  2.903e-06,  Adjusted R-squared:  -0.01205
## F-statistic: 0.0002409 on 1 and 83 DF,  p-value: 0.9877
```

```
#Linear regression tests
#Aggregate Risk Aversion and gdp
lmAggregate_Risk_Aversion_GDP = lm(as.numeric(GDP) ~ Aggregate_Risk_Aversion, data = combined_da
ta_Aggregate_Risk_Aversion)
summary(lmAggregate_Risk_Aversion_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ Aggregate_Risk_Aversion, data = combined_data_Aggregate_Risk_A
version)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6533 -0.5474 -0.0481  0.5785  2.4498
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2.146446   0.101734  21.099   <2e-16 ***
## Aggregate_Risk_Aversion 0.003891   0.104439   0.037     0.97
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9324 on 82 degrees of freedom
## Multiple R-squared:  1.693e-05,  Adjusted R-squared:  -0.01218
## F-statistic: 0.001388 on 1 and 82 DF,  p-value: 0.9704
```

```
#try with no intercept now (X - 1)
lmAggregate_Risk_Aversion_GDP_NOINT = lm(as.numeric(GDP) ~ Aggregate_Risk_Aversion - 1, data = c
ombined_data_Aggregate_Risk_Aversion)
summary(lmAggregate_Risk_Aversion_GDP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ Aggregate_Risk_Aversion - 1, data = combined_data_Aggregate_Ri
sk_Aversion)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4885  1.5974  2.0986  2.7220  4.6063
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## Aggregate_Risk_Aversion -0.006498   0.263201  -0.025     0.98
##
## Residual standard error: 2.35 on 83 degrees of freedom
## Multiple R-squared:  7.344e-06,  Adjusted R-squared:  -0.01204
## F-statistic: 0.0006095 on 1 and 83 DF,  p-value: 0.9804
```

```
#Linear regression tests
#Aggregate Risk Aversion and Gross private domestic investment
lmAggregate_Risk_Aversion_GDP = lm(as.numeric(Gross.private.domestic.investment) ~ Aggregate_Ris
k_Aversion, data = combined_data_Aggregate_Risk_Aversion)
summary(lmAggregate_Risk_Aversion_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     Aggregate_Risk_Aversion, data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3744 -1.2733  0.1119  1.0205  6.1669
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.9673     0.2174   4.449 2.69e-05 ***
## Aggregate_Risk_Aversion   -0.3913     0.2232  -1.753   0.0833 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.993 on 82 degrees of freedom
## Multiple R-squared:  0.03613,    Adjusted R-squared:  0.02437
## F-statistic: 3.074 on 1 and 82 DF,  p-value: 0.08331
```

```
#try with no intercept now (X - 1)
lmAggregate_Risk_Aversion_GPDI_NOINT = lm(as.numeric(Gross.private.domestic.investment) ~ Aggreg
ate_Risk_Aversion - 1, data = combined_data_Aggregate_Risk_Aversion)
summary(lmAggregate_Risk_Aversion_GPDI_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     Aggregate_Risk_Aversion - 1, data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3989 -0.3094  1.0813  1.9915  7.1309
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## Aggregate_Risk_Aversion   -0.3960     0.2472  -1.602    0.113
##
## Residual standard error: 2.207 on 83 degrees of freedom
## Multiple R-squared:  0.02999,    Adjusted R-squared:  0.01831
## F-statistic: 2.567 on 1 and 83 DF,  p-value: 0.1129
```

```
#Linear regression tests
#gross ipo and nondurable goods
lmAggregate_Risk_Aversion_NDG = lm(as.numeric(Nondurable.goods) ~ Aggregate_Risk_Aversion, data
= combined_data_Aggregate_Risk_Aversion)
summary(lmAggregate_Risk_Aversion_NDG)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ Aggregate_Risk_Aversion,
##     data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.9575  -1.9697  -0.3288   2.4184  10.4345
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.2354     0.5368   4.164 7.68e-05 ***
## Aggregate_Risk_Aversion   -0.3268     0.5511  -0.593    0.555
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.92 on 82 degrees of freedom
## Multiple R-squared:  0.00427,    Adjusted R-squared:  -0.007873
## F-statistic: 0.3516 on 1 and 82 DF,  p-value: 0.5548
```

```
#try with no intercept now (X - 1)
lmAggregate_Risk_Aversion_NDG_NOINT = lm(as.numeric(Nondurable.goods) ~ Aggregate_Risk_Aversion
- 1, data = combined_data_Aggregate_Risk_Aversion)
summary(lmAggregate_Risk_Aversion_NDG_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ Aggregate_Risk_Aversion -
##     1, data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.7294   0.2604   1.9101   4.6564  12.6623
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## Aggregate_Risk_Aversion   -0.3376     0.6029   -0.56    0.577
##
## Residual standard error: 5.383 on 83 degrees of freedom
## Multiple R-squared:  0.003764,   Adjusted R-squared:  -0.008239
## F-statistic: 0.3136 on 1 and 83 DF,  p-value: 0.577
```

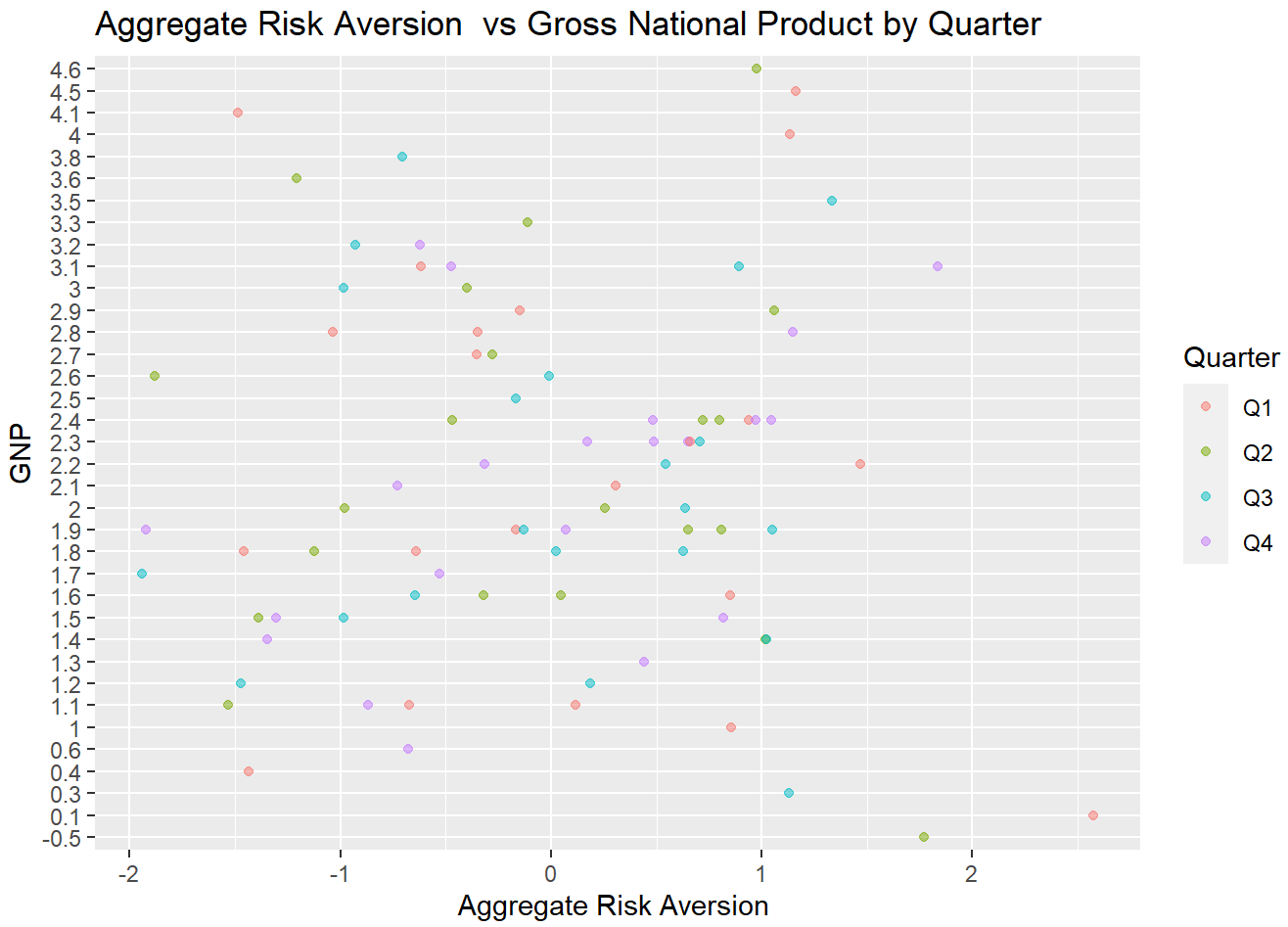
```
#Linear regression tests
#Aggregate Risk Aversion and personal consumption expenditures
lmAggregate_Risk_Aversion_PCE = lm(as.numeric(Personal.consumption.expenditures) ~ Aggregate_Ris
k_Aversion, data = combined_data_Aggregate_Risk_Aversion)
summary(lmAggregate_Risk_Aversion_PCE)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     Aggregate_Risk_Aversion, data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3694 -0.5246  0.1356  0.7088  3.6829
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2.18702    0.17421  12.554   <2e-16 ***
## Aggregate_Risk_Aversion  0.02595    0.17884   0.145    0.885
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.597 on 82 degrees of freedom
## Multiple R-squared:  0.0002567,  Adjusted R-squared:  -0.01194
## F-statistic: 0.02105 on 1 and 82 DF,  p-value: 0.885
```

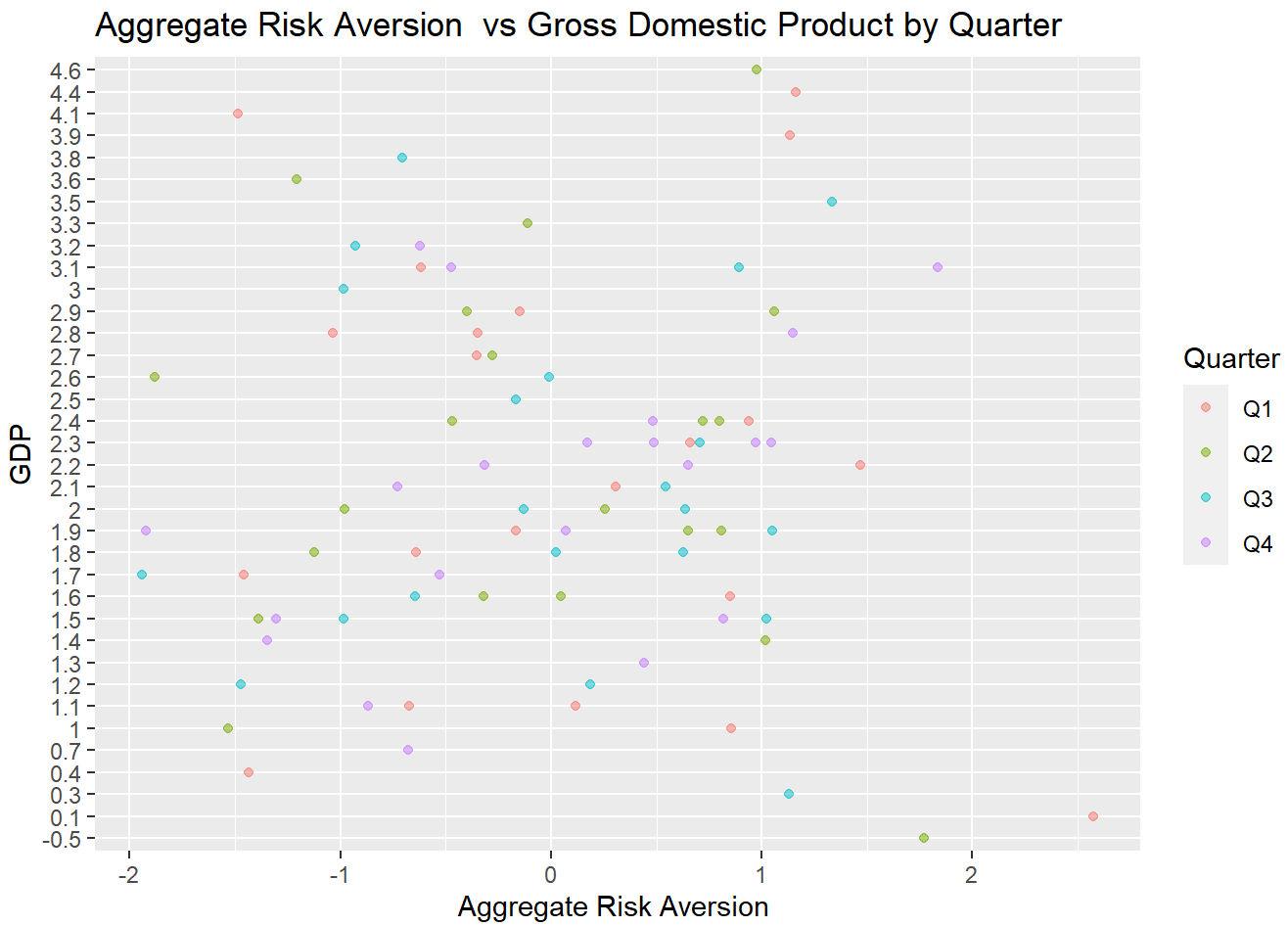
```
#try with no intercept now (X - 1)
lmAggregate_Risk_Aversion_PCE_NOINT = lm(as.numeric(Personal.consumption.expenditures) ~ Aggrega
te_Risk_Aversion - 1, data = combined_data_Aggregate_Risk_Aversion)
summary(lmAggregate_Risk_Aversion_PCE_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     Aggregate_Risk_Aversion - 1, data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.190  1.668  2.313  2.897  5.882
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## Aggregate_Risk_Aversion  0.01536    0.30386   0.051     0.96
##
## Residual standard error: 2.713 on 83 degrees of freedom
## Multiple R-squared:  3.08e-05,   Adjusted R-squared:  -0.01202
## F-statistic: 0.002557 on 1 and 83 DF,  p-value: 0.9598
```

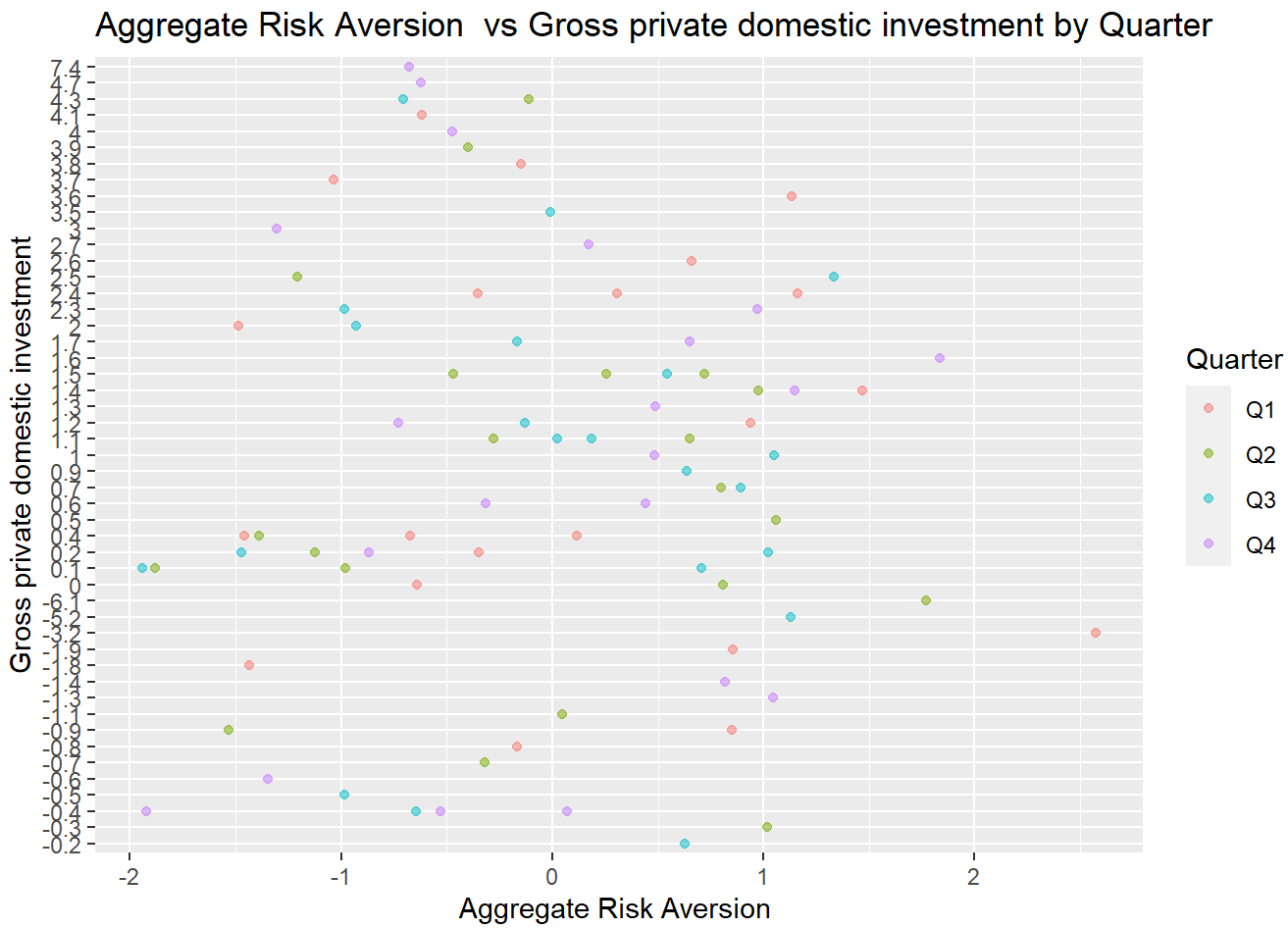
```
#Make scatterplots of relationships, color by quarter
#Start with Aggregate Risk Aversion  and GNP
 combined_data_Aggregate_Risk_Aversion|>
  ggplot(aes(x = Aggregate_Risk_Aversion, y = Gross.national.product,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "Aggregate Risk Aversion ", y = "GNP", color = "Quarter",
       title = "Aggregate Risk Aversion  vs Gross National Product by Quarter")
```



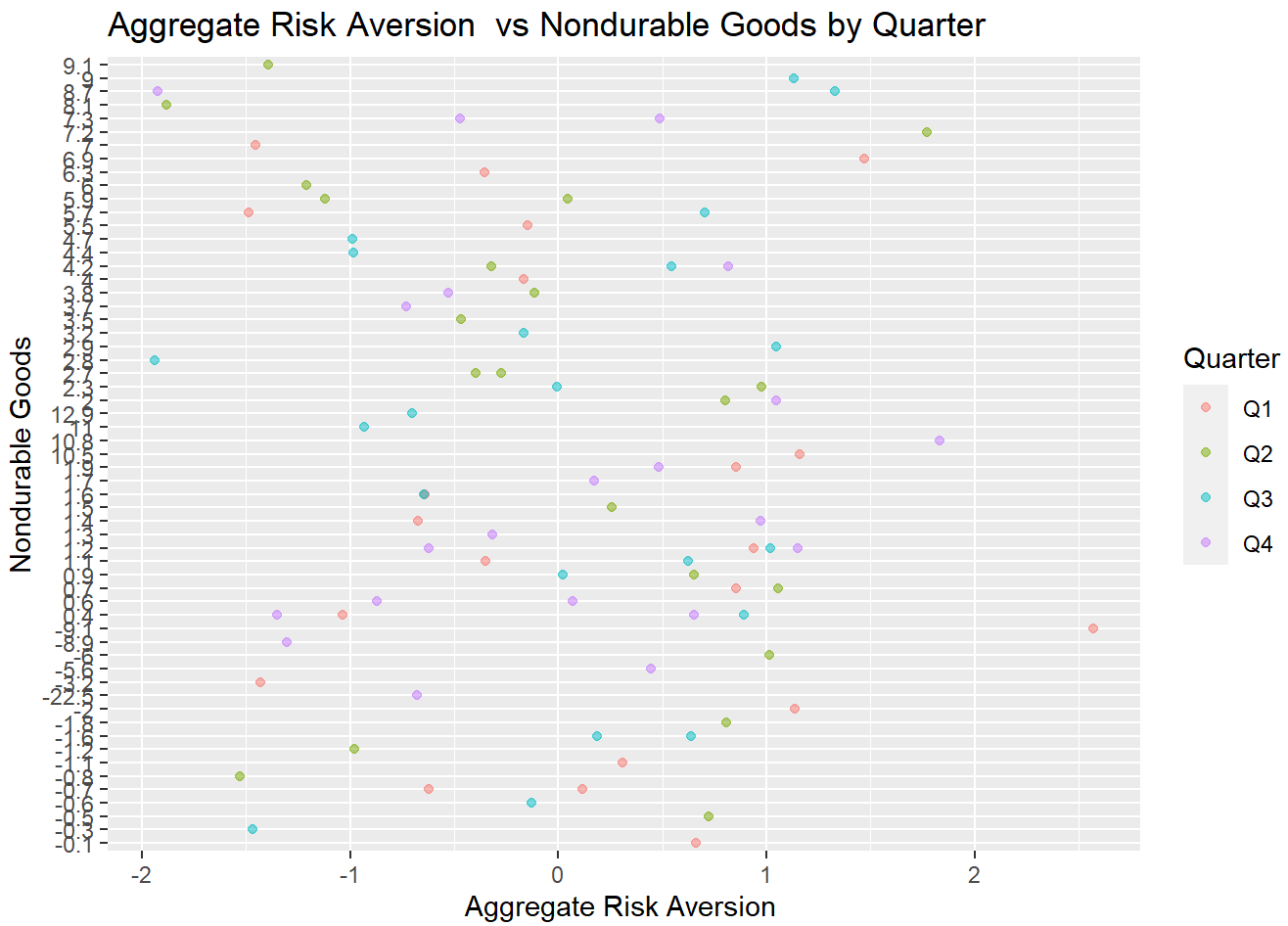

```
#Make scatterplots of relationships, color by quarter
#Aggregate Risk Aversion  and GDP
 combined_data_Aggregate_Risk_Aversion|>
  ggplot(aes(x = Aggregate_Risk_Aversion, y = GDP,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "Aggregate Risk Aversion ", y = "GDP", color = "Quarter",
       title = "Aggregate Risk Aversion  vs Gross Domestic Product by Quarter")
```



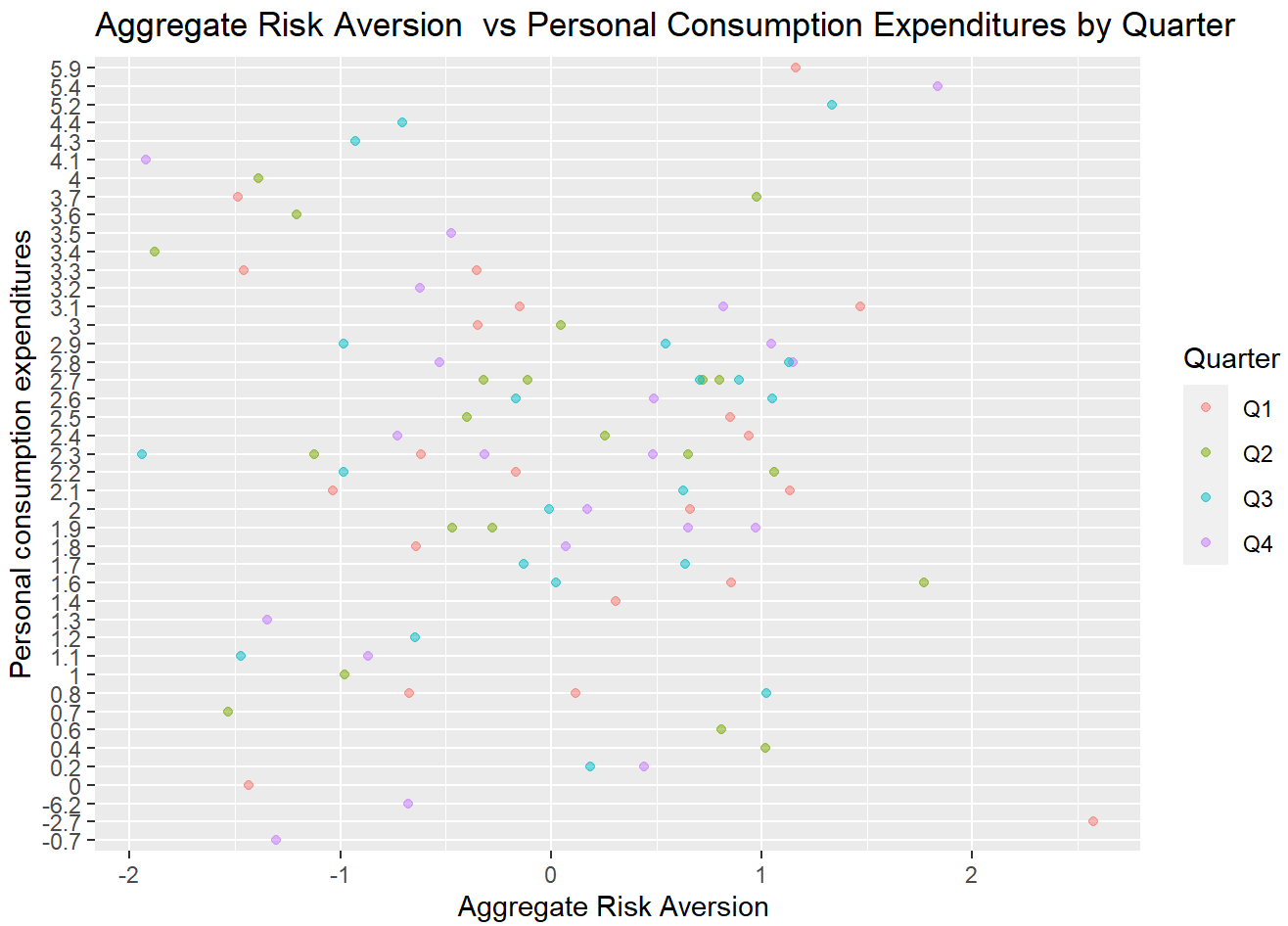

```
#Make scatterplots of relationships, color by quarter
#Aggregate Risk Aversion  and GPDI
 combined_data_Aggregate_Risk_Aversion|>
  ggplot(aes(x = Aggregate_Risk_Aversion, y = Gross.private.domestic.investment,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "Aggregate Risk Aversion ", y = "Gross private domestic investment", color = "Quarte
r",
       title = "Aggregate Risk Aversion  vs Gross private domestic investment by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#Aggregate Risk Aversion  and Nondurable Goods
 combined_data_Aggregate_Risk_Aversion|>
  ggplot(aes(x = Aggregate_Risk_Aversion, y = Nondurable.goods,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "Aggregate Risk Aversion ", y = "Nondurable Goods", color = "Quarter",
       title = "Aggregate Risk Aversion  vs Nondurable Goods by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#Aggregate Risk Aversion  and Personal consumption expenditures
 combined_data_Aggregate_Risk_Aversion|>
  ggplot(aes(x = Aggregate_Risk_Aversion, y = Personal.consumption.expenditures,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "Aggregate Risk Aversion ", y = "Personal consumption expenditures", color = "Quarte
r",
       title = "Aggregate Risk Aversion  vs Personal Consumption Expenditures by Quarter")
```



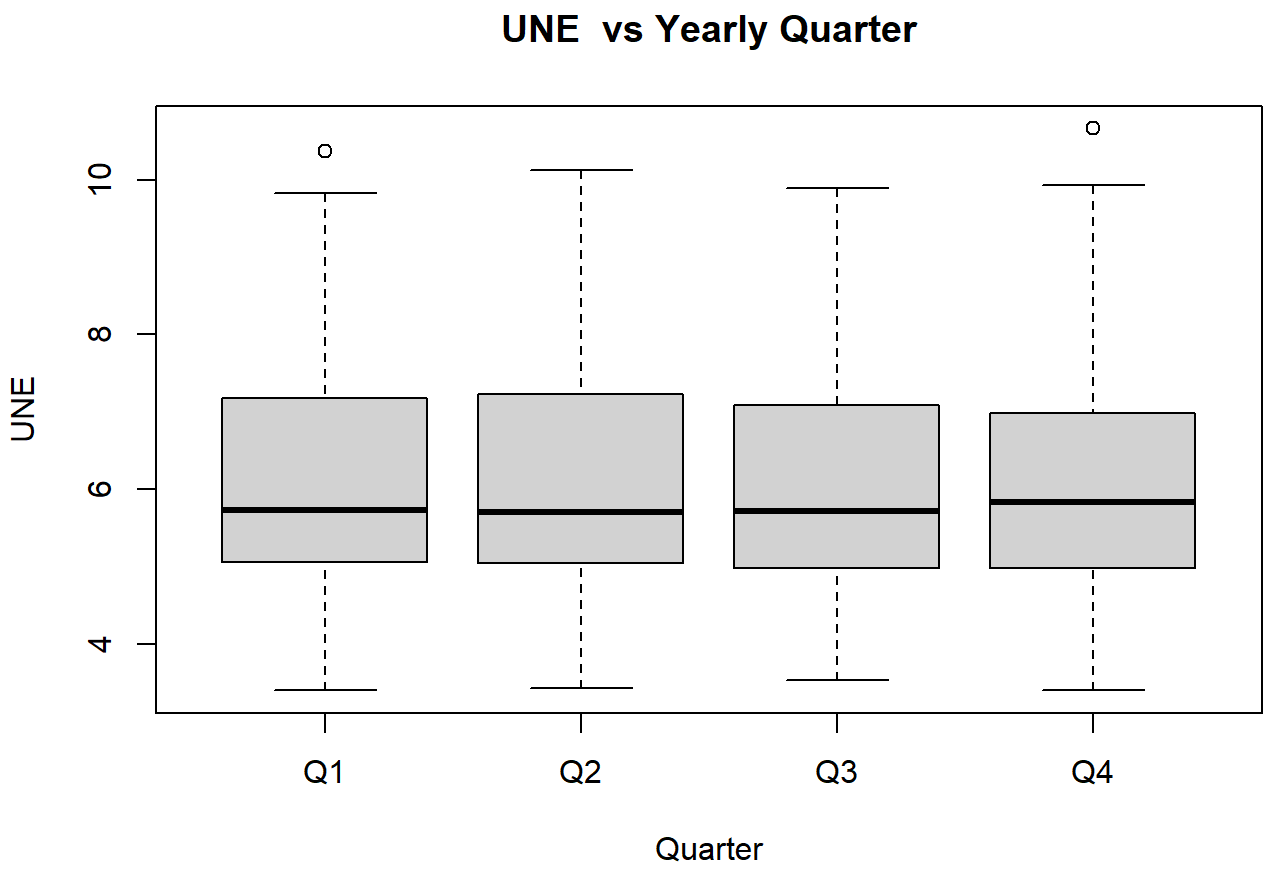

```
#Risk Aversion Measure 4: Unemployment Rate, all tests and graphs below
```

```
#Start with an anova test for Gross IPO

#1: Anova test on the quarters to see if there is difference in the UNE
#by time in the year

UNE.aov <- aov(UNE  ~ Quarter, data = combined_data_UNE )
# Summary
summary(UNE.aov)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Quarter       3    0.0  0.0022   0.001      1
## Residuals   220  559.9  2.5449
```

```
#P-value of 1. Indicates that there are no significant differences in the
#mean UNE by quarter
```

```
#boxplots of the quarter ~ UNE relationship
#Notes: Outliers present for Q1 and Q4
#Notes: No real potential oscillation between quarters
boxplot(UNE ~Quarter,data=combined_data_UNE , main="UNE  vs Yearly Quarter",
   xlab="Quarter", ylab="UNE")
```




```
#Run correlation tests between different GDP measures and UNE
#Years: 1960 to 2015
#Correlation with GNP
cor(combined_data_UNE$UNE, as.numeric(combined_data_UNE$Gross.national.product))
```

```
## [1] 0.06182329
```

```
#Correlation with GDP
cor(combined_data_UNE$UNE, as.numeric(combined_data_UNE$GDP))
```

```
## [1] 0.06181774
```

```
#Correlation with GPDI
cor(combined_data_UNE$UNE, as.numeric(combined_data_UNE$Gross.private.domestic.investment))
```

```
## [1] -0.03178499
```

```
#Correlation with nondurable goods
cor(combined_data_UNE$UNE, as.numeric(combined_data_UNE$Nondurable.goods))
```

```
## [1] -0.0515916
```

```
#Correlation with personal consumption expenditures
cor(combined_data_UNE$UNE, as.numeric(combined_data_UNE$Personal.consumption.expenditures))
```

```
## [1] 0.08001998
```

```
#Mix of positive and negative correlations, although all less than 0.1 (absolute value)
```

```
#Linear regression tests
#Start with UNE and GNP
lmUNE_GNP = lm(as.numeric(Gross.national.product) ~ UNE, data = combined_data_UNE)
summary(lmUNE_GNP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ UNE, data = combined_data_UNE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1957 -1.5701 -0.7473  0.9589  9.5568
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.80202    0.65540   4.275 2.84e-05 ***
## UNE          0.09612    0.10415   0.923    0.357
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.464 on 222 degrees of freedom
## Multiple R-squared:  0.003822,   Adjusted R-squared:  -0.0006652
## F-statistic: 0.8518 on 1 and 222 DF,  p-value: 0.3571
```

```
#try with no intercept now
lmUNE_GNP_NOINT = lm(as.numeric(Gross.national.product) ~ UNE - 1, data = combined_data_UNE)
summary(lmUNE_GNP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.national.product) ~ UNE - 1, data = combined_data_UNE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4006 -1.4543 -0.5118  1.3914  9.9324
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## UNE  0.52709    0.02716   19.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.558 on 223 degrees of freedom
## Multiple R-squared:  0.6281, Adjusted R-squared:  0.6265
## F-statistic: 376.7 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
#Linear regression tests
#UNE and gdp
lmUNE_GDP = lm(as.numeric(GDP) ~ UNE, data = combined_data_UNE)
summary(lmEPBound_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     EPBound, data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8441 -0.9759 -0.3597  1.2007  7.1884
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0008     0.4352   2.299   0.0249 *
## EPBound      -2.6968     6.8271  -0.395   0.6942
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.23 on 62 degrees of freedom
## Multiple R-squared:  0.00251,    Adjusted R-squared:  -0.01358
## F-statistic: 0.156 on 1 and 62 DF,  p-value: 0.6942
```

```
#try with no intercept now (X - 1)
lmUNE_GDP_NOINT = lm(as.numeric(GDP) ~ UNE - 1, data = combined_data_UNE)
summary(lmUNE_GDP_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(GDP) ~ UNE - 1, data = combined_data_UNE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3948 -1.4386 -0.5288  1.4169  9.9359
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## UNE  0.52647    0.02713   19.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.555 on 223 degrees of freedom
## Multiple R-squared:  0.6281, Adjusted R-squared:  0.6264
## F-statistic: 376.6 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
#Linear regression tests
#UNE and Gross private domestic investment
lmUNE_GDP = lm(as.numeric(Gross.private.domestic.investment) ~ UNE, data = combined_data_UNE)
summary(lmUNE_GDP)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     UNE, data = combined_data_UNE)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.528 -2.375 -1.048  1.413 13.919
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.07059    0.91803   3.345 0.000967 ***
## UNE         -0.06912    0.14588  -0.474 0.636090
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.452 on 222 degrees of freedom
## Multiple R-squared:  0.00101,    Adjusted R-squared:  -0.00349
## F-statistic: 0.2245 on 1 and 222 DF,  p-value: 0.6361
```

```
#try with no intercept now (X - 1)
lmUNE_GPDI_NOINT = lm(as.numeric(Gross.private.domestic.investment) ~ UNE - 1, data = combined_d
ata_UNE)
summary(lmUNE_GPDI_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Gross.private.domestic.investment) ~
##     UNE - 1, data = combined_data_UNE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8484 -2.0706 -0.8215  1.9577 14.3301
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## UNE  0.40316    0.03748   10.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.53 on 223 degrees of freedom
## Multiple R-squared:  0.3417, Adjusted R-squared:  0.3387
## F-statistic: 115.7 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
#Linear regression tests
#UNE and nondurable goods
lmUNE_NDG = lm(as.numeric(Nondurable.goods) ~ UNE, data = combined_data_UNE)
summary(lmUNE_NDG)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ UNE, data = combined_data_UNE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.5378  -2.3601  -0.5916   2.1620  18.9938
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.1046     1.2717   3.228  0.00144 **
## UNE          -0.1555     0.2021  -0.770  0.44228
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.781 on 222 degrees of freedom
## Multiple R-squared:  0.002662,   Adjusted R-squared:  -0.001831
## F-statistic: 0.5925 on 1 and 222 DF,  p-value: 0.4423
```

```
#try with no intercept now (X - 1)
lmUNE_NDG_NOINT = lm(as.numeric(Nondurable.goods) ~ UNE - 1, data = combined_data_UNE)
summary(lmUNE_NDG_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Nondurable.goods) ~ UNE - 1, data = combined_data_UNE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.7633  -2.3072  -0.2798   2.5176  19.8578
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## UNE  0.47578    0.05183    9.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.881 on 223 degrees of freedom
## Multiple R-squared:  0.2743, Adjusted R-squared:  0.271
## F-statistic: 84.27 on 1 and 223 DF,  p-value: < 2.2e-16
```

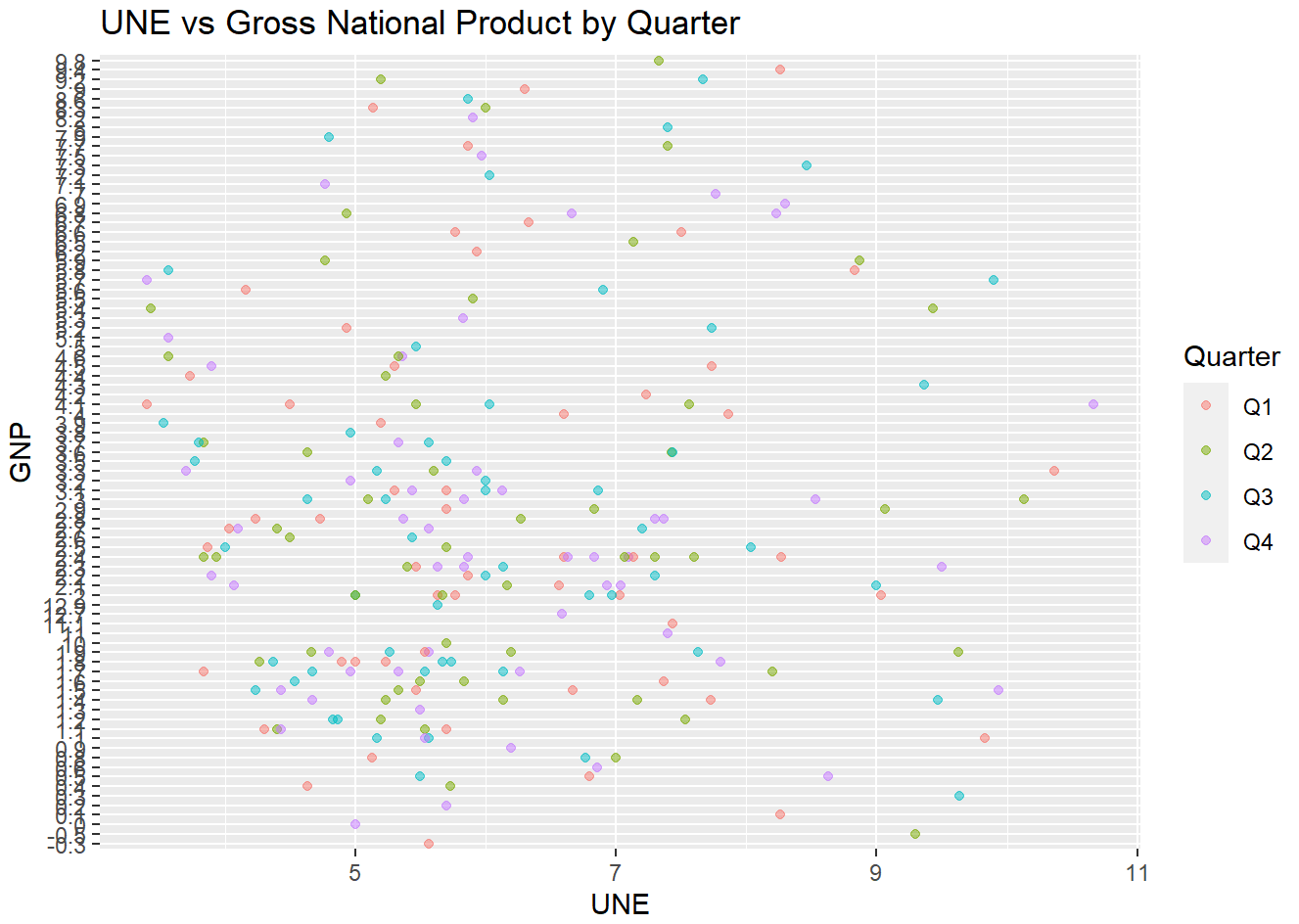
```
#Linear regression tests
#UNE and personal consumption expenditures
lmUNE_PCE = lm(as.numeric(Personal.consumption.expenditures) ~ UNE, data = combined_data_UNE)
summary(lmUNE_PCE)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     UNE, data = combined_data_UNE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6895 -1.6282 -0.6461  1.1627  9.2468
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.5503     0.7204   3.540 0.000488 ***
## UNE           0.1369     0.1145   1.196 0.232931
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.709 on 222 degrees of freedom
## Multiple R-squared:  0.006403,   Adjusted R-squared:  0.001928
## F-statistic: 1.431 on 1 and 222 DF,  p-value: 0.2329
```

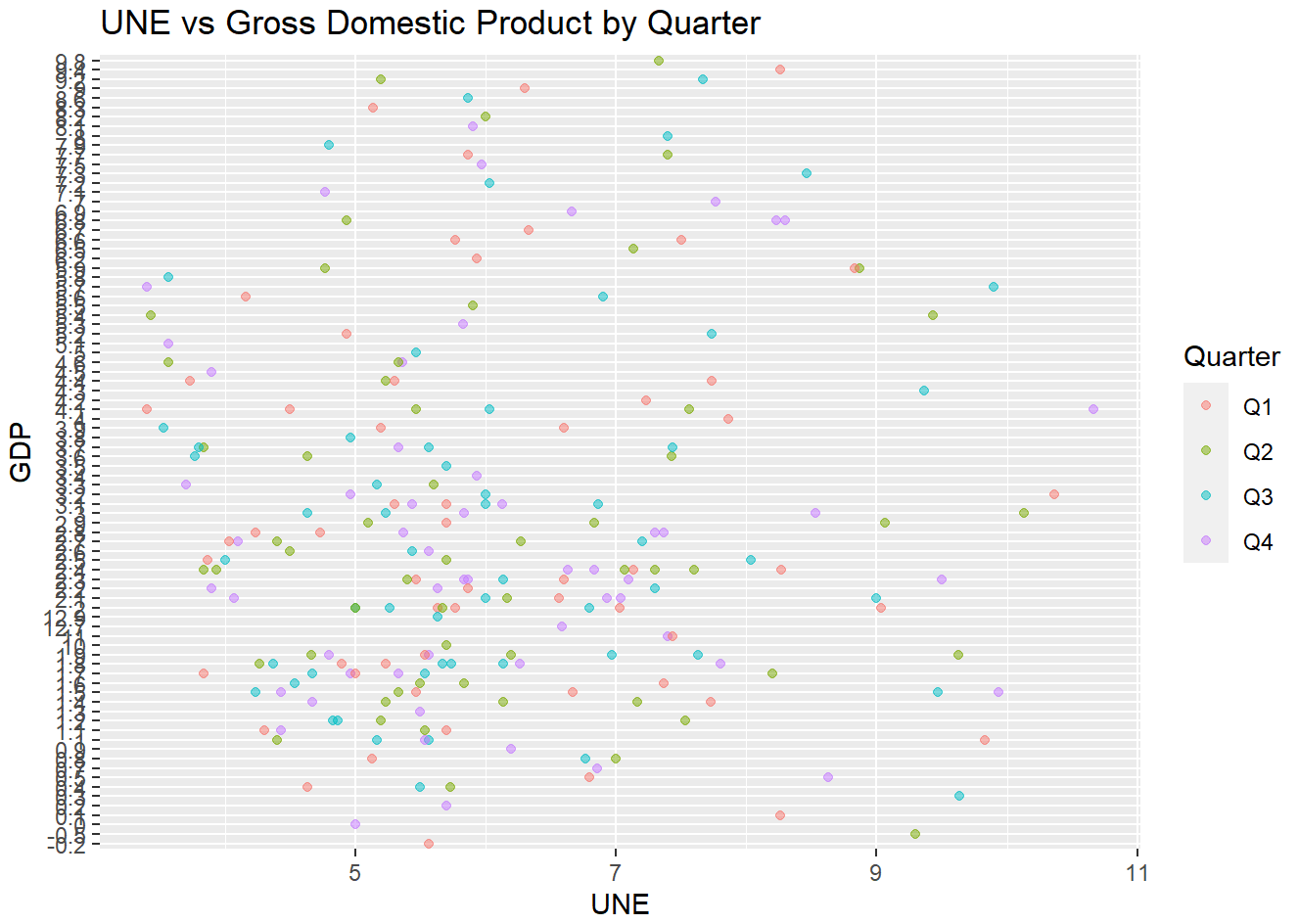
```
#try with no intercept now (X - 1)
lmUNE_PCE_NOINT = lm(as.numeric(Personal.consumption.expenditures) ~ UNE - 1, data = combined_da
ta_UNE)
summary(lmUNE_PCE_NOINT)
```

```
##
## Call:
## lm(formula = as.numeric(Personal.consumption.expenditures) ~
##     UNE - 1, data = combined_data_UNE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8296 -1.4409 -0.4164  1.3485  9.7836
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## UNE  0.52919    0.02949   17.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.778 on 223 degrees of freedom
## Multiple R-squared:  0.5908, Adjusted R-squared:  0.5889
## F-statistic: 321.9 on 1 and 223 DF,  p-value: < 2.2e-16
```

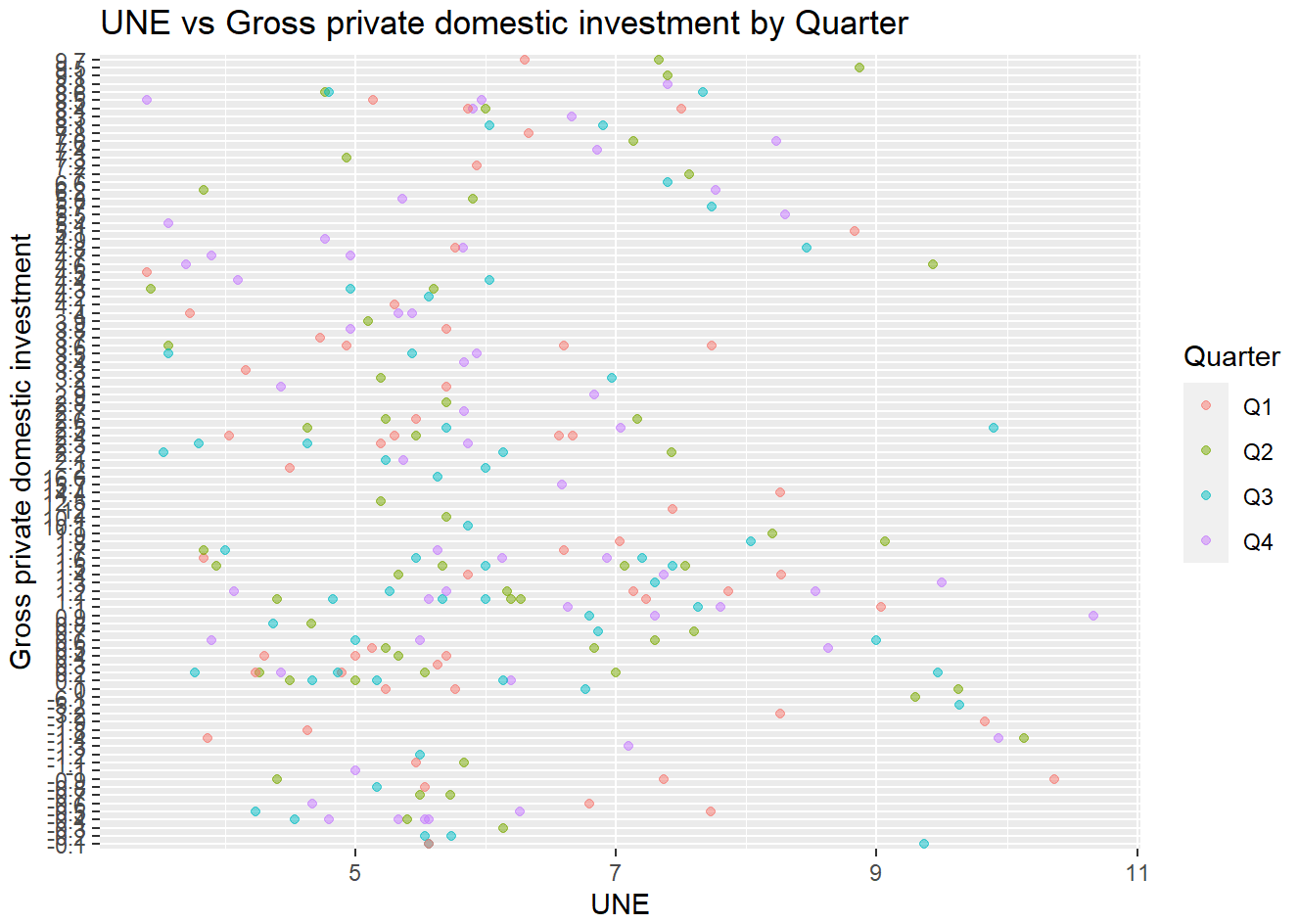
```
#Make scatterplots of relationships, color by quarter
#Start with UNE and GNP
 combined_data_UNE|>
  ggplot(aes(x = as.numeric(UNE), y = Gross.national.product,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "UNE", y = "GNP", color = "Quarter",
       title = "UNE vs Gross National Product by Quarter")
```



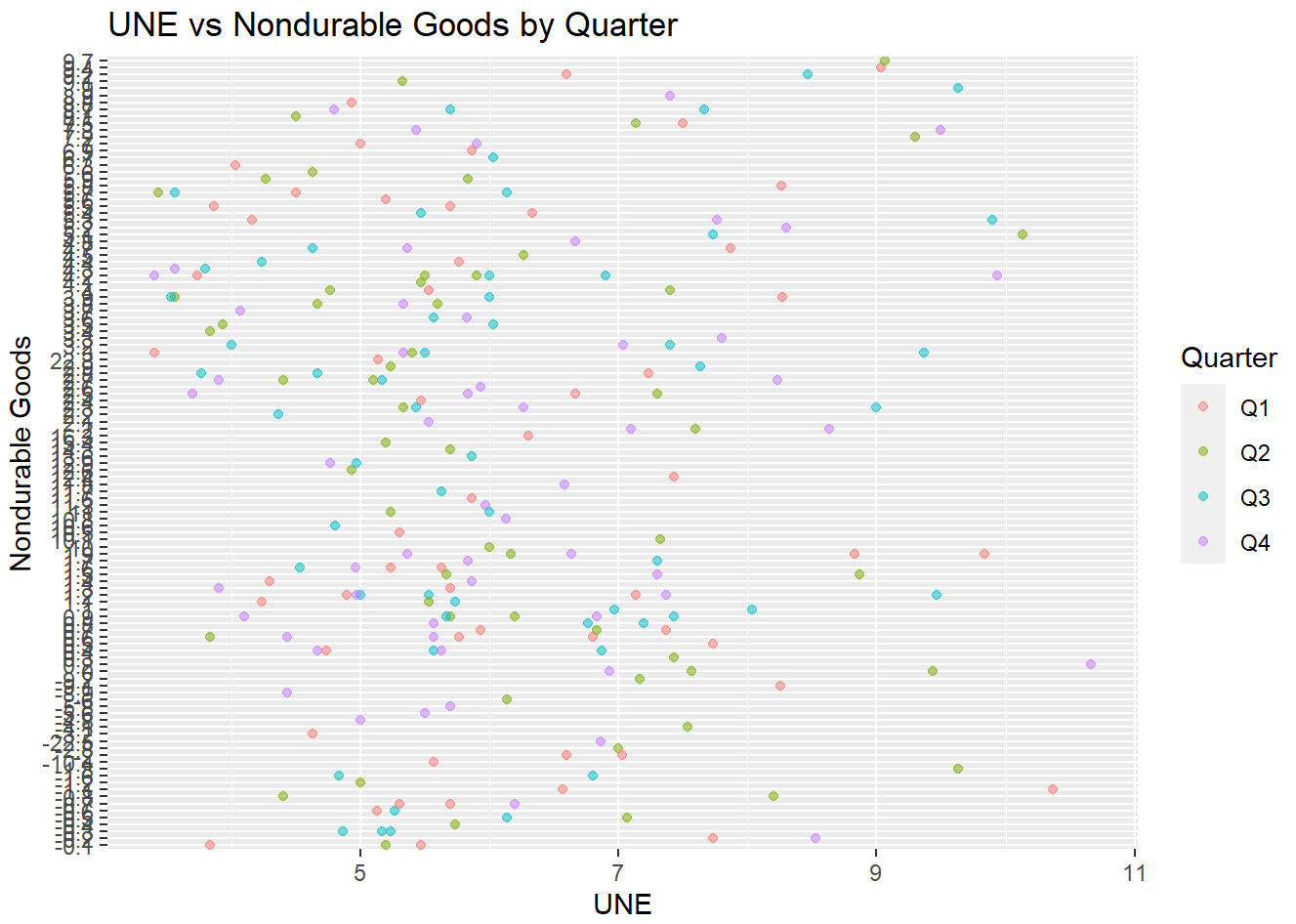

```
#Make scatterplots of relationships, color by quarter
#IPO Gross and GDP
 combined_data_UNE|>
  ggplot(aes(x= as.numeric(UNE), y = GDP,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "UNE", y = "GDP", color = "Quarter",
       title = "UNE vs Gross Domestic Product by Quarter")
```



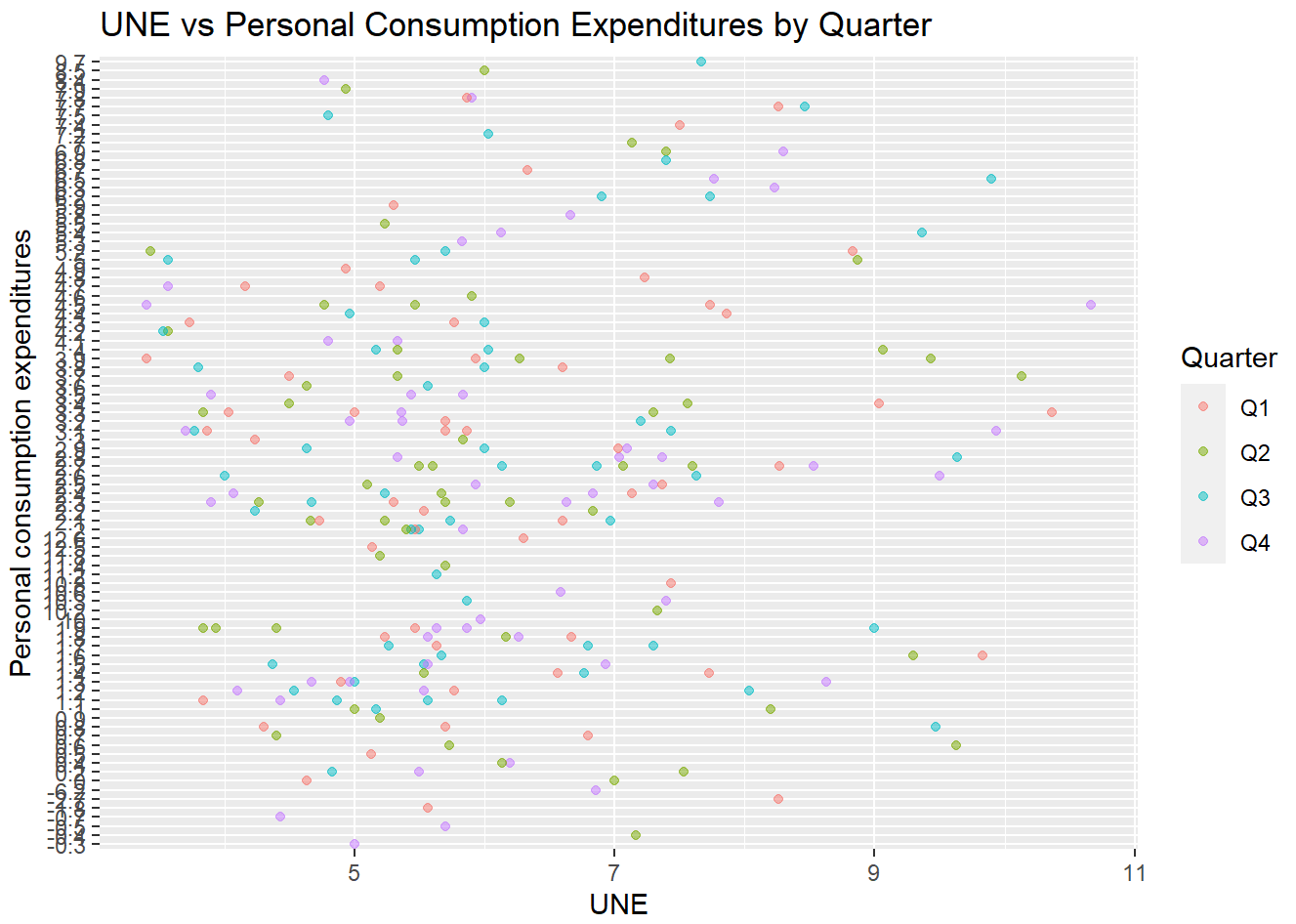

```
#Make scatterplots of relationships, color by quarter
#UNE and GPDI
 combined_data_UNE|>
  ggplot(aes(x = as.numeric(UNE), y = Gross.private.domestic.investment,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "UNE", y = "Gross private domestic investment", color = "Quarter",
       title = "UNE vs Gross private domestic investment by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#UNE and Nondurable Goods
 combined_data_UNE|>
  ggplot(aes(x = as.numeric(UNE), y = Nondurable.goods,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "UNE", y = "Nondurable Goods", color = "Quarter",
       title = "UNE vs Nondurable Goods by Quarter")
```




```
#Make scatterplots of relationships, color by quarter
#UNE and Personal consumption expenditures
 combined_data_UNE|>
  ggplot(aes(x = as.numeric(UNE), y = Personal.consumption.expenditures,
             color = Quarter)) +
  geom_point(alpha = 0.5) +
  labs(x = "UNE", y = "Personal consumption expenditures", color = "Quarter",
       title = "UNE vs Personal Consumption Expenditures by Quarter")
```




```
#check how to get fitted values
lmIPO_GDP$fitted.values
```

```
##      ...54      ...55      ...56      ...57      ...58      ...59      ...60
## 2.94636042 2.73940404 2.85829659 2.64799381 2.87114237 2.39774152 2.18323173
##      ...61      ...62      ...63      ...64      ...65      ...66      ...67
## 1.61842412 2.11283750 2.68059749 3.27054669 3.26193924 3.45253031 3.28264117
##      ...68      ...69      ...70      ...71      ...72      ...73      ...74
## 3.34557563 3.33362002 3.44544284 3.22349020 3.32087121 3.28264117 3.29943329
##      ...75      ...76      ...77      ...78      ...79      ...80      ...81
## 3.09572502 3.21964240 3.10098059 3.39491397 3.12594521 3.47902873 3.44890776
##      ...82      ...83      ...84      ...85      ...86      ...87      ...88
## 3.48829121 3.31241424 3.26405091 3.14458820 2.98864883 2.75314831 2.56783172
##      ...89      ...90      ...91      ...92      ...93      ...94      ...95
## 1.52296057 1.01551449 0.76350561 2.12427701 0.80050006 1.79931167 2.80179503
##      ...96      ...97      ...98      ...99     ...100     ...101     ...102
## 2.85829659 2.67272516 2.78521953 2.46312207 2.35260015 1.96464008 1.86740168
##     ...103     ...104     ...105     ...106     ...107     ...108     ...109
## 1.35024679 2.19597074 2.06005188 2.92710404 3.41783052 3.49084419 3.60824450
##     ...110     ...111     ...112     ...113     ...114     ...115     ...116
## 3.60824450 3.60824450 3.60824450 3.60824450 3.60824450 3.60824450 3.55375210
##     ...117     ...118     ...119     ...120     ...121     ...122     ...123
## 3.55375210 3.52174321 3.60824450 3.47093226 3.54826786 3.49925970 3.49776657
##     ...124     ...125     ...126     ...127     ...128     ...129     ...130
## 3.53958838 3.49925970 3.52174321 3.47093226 3.45106122 3.38728865 3.42831456
##     ...131     ...132     ...133     ...134     ...135     ...136     ...137
## 3.33946487 3.43287172 3.24921957 3.22941044 3.12394394 2.66417555 2.46960273
##     ...138     ...139     ...140     ...141     ...142     ...143     ...144
## 2.43486781 1.55830350 2.42031906 2.46131510 3.08018476 2.97142910 3.09792471
##     ...145     ...146     ...147     ...148     ...149     ...150     ...151
## 2.56096760 2.13702447 1.23271217 0.05759674 0.11927699 1.29710372 1.88092606
##     ...152     ...153     ...154     ...155     ...156     ...157     ...158
## 2.16537700 2.20526228 2.42878803 2.09692772 1.90829304 1.66069662 1.66236182
##     ...159     ...160     ...161     ...162     ...163     ...164     ...165
## 0.57907604 0.44724623 0.08232602 1.55326735 1.28475604 1.03062263 2.93785199
##     ...166     ...167     ...168     ...169     ...170     ...171     ...172
## 3.05193500 2.74670556 2.95032390 2.90311866 3.21308643 2.96667657 3.03404270
##     ...173     ...174     ...175     ...176     ...177     ...178     ...179
## 2.71587294 3.06550898 2.81666385 3.09408757 3.38728865 3.30756370 2.35096022
##     ...180     ...181     ...182     ...183     ...184     ...185     ...186
## 2.42659501 1.86314064 1.69697107 1.88400474 2.47576536 2.09128922 2.22467450
##     ...187     ...188     ...189     ...190     ...191     ...192     ...193
## 1.94819346 1.61327903 0.96132393 1.86432217 1.47247688 2.08729944 2.03014936
##     ...194     ...195     ...196     ...197     ...198     ...199     ...200
## 2.59452083 2.13176874 1.89796607 0.82202017 1.65762040 0.51650556 1.30731404
##     ...201     ...202     ...203     ...204     ...205     ...206     ...207
## 0.57940525 1.94940783 1.91882892 1.63448269 1.46706714 2.27527579 1.69099509
##     ...208     ...209     ...210     ...211     ...212     ...213     ...214
## 2.88548325 3.08724208 2.57263739 1.73382238 1.79746743 1.61223153 2.18013963
##     ...215     ...216     ...217     ...218     ...219     ...220     ...221
## 2.42600300 2.01664945 2.95585290 3.17573804 3.26223933 3.31871043 3.12594521
##     ...222     ...223     ...224     ...225     ...226     ...227     ...228
## 3.25604355 2.99460685 3.26405091 2.97987544 3.49925970 3.40749279 3.11037653
##     ...229     ...230     ...231     ...232     ...233     ...234     ...235
## 2.84019607 2.91072169 2.73940404 2.67726659 2.43111898 2.92846306 2.81420391
```

```
##     ...236     ...237     ...238     ...239     ...240     ...241     ...242
## 2.55828612 2.76519352 2.95568398 2.84025703 3.05971163 2.42904657 2.83133805
##     ...243     ...244     ...245     ...246     ...247     ...248     ...249
## 2.87597214 3.16495611 2.57198401 3.29681612 3.39690783 3.60824450 3.60824450
##     ...250     ...251     ...252     ...253     ...254     ...255     ...256
## 3.54826786 3.37878211 3.37344387 3.23028940 3.18158344 3.04234158 3.02917635
##     ...257     ...258     ...259     ...260     ...261     ...262     ...263
## 2.71984161 2.99062208 2.75581218 3.36865070 3.18158344 3.07068525 3.12594521
##     ...264     ...265     ...266     ...267     ...268     ...269     ...270
## 3.12517353 3.08290676 3.11609966 2.73372518 2.82724900 2.71308208 2.77189966
##     ...271     ...272     ...273     ...274     ...275     ...276     ...277
## 2.57236924 2.86906351 2.70066797 3.15908705 2.80415170 3.19378049 3.31742928
```

```
library(tidyverse)
#importing the new dataset with recession data by month
USREC_data <- read_csv("USREC.csv")
```

```
## Rows: 2033 Columns: 2
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl  (1): USREC
## date (1): DATE
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

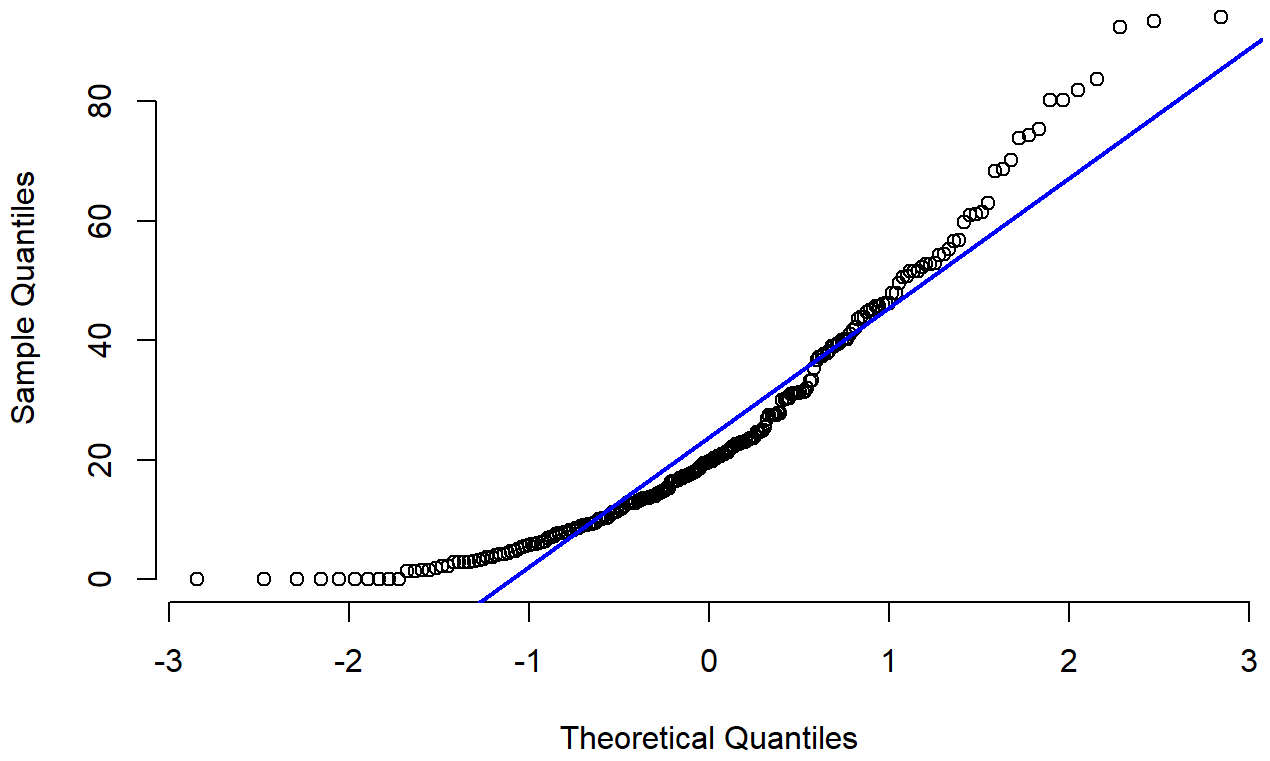
```
#Start by changing the recession data to only include 1960 to 2015
USREC_data_mod <- data.frame(USREC_data)
USREC_data_mod <- USREC_data_mod[-c(1934: 2033), ]
USREC_data_mod <- USREC_data_mod[-c(1: 1261), ]
```

```
#adjust recession data by quarter


#0 is expansionary, 1 is recessionary
#adjust by quarter, Y = recession, N = no recession
quarter_rec = function(months){
  result <- c()
  i = 1
  while(i < length(months)) {
      len_result <- length(result)
      if (months[i] == 0 & months[i + 1] == 0 &  months[i + 2] == 0) {
        result[[len_result + 1]] <- "N"
      }
      else {
          result[[len_result + 1]] <- "Y"
      }
    i = i + 3
  }
  return (result)
}

#store in vector called quarter_recession
quarter_recession = quarter_rec(USREC_data_mod$USREC)
```

```
#add this new column of recession data to the individual datasets
gdp_data_mod_final$recession = quarter_recession
combined_data_DPRatio$recession = quarter_recession
combined_data_IPO_Gross$recession = quarter_recession
combined_data_SCR$recession = quarter_recession
combined_data_UNE$recession = quarter_recession
#adjust to 1990 to 2010 for aggregate risk aversion dataset
quarter_recession_ARE <- c()
for (i in 1:84){
quarter_recession_ARE[i] = gdp_data_mod_final$recession[i + 120]
}
combined_data_Aggregate_Risk_Aversion$recession = quarter_recession_ARE

#adjust to 1996 to 2011 for Equity Premium Bound dataset
quarter_recession_EP <- c()
for (i in 1:64){
quarter_recession_EP[i] = gdp_data_mod_final$recession[i + 144]
}
combined_data_EPBound$recession = quarter_recession_EP

#adjust to 1970 to 2015 for Equity Premium Bound dataset
quarter_recession_PVS <- c()
for (i in 1:183){
quarter_recession_PVS[i] = gdp_data_mod_final$recession[i + 40]
}
combined_data_PVS$recession = quarter_recession_PVS

#all individual datasets now have recession data per quarter
```

```
#Run tests of equal variancefor each dataset measure by recession
```

```
#Equal variance test by recession for gross_ipo

#check normality first
qqnorm(combined_data_IPO_Gross$IPO_Gross, pch = 1, frame = FALSE)
qqline(combined_data_IPO_Gross$IPO_Gross, col = "blue", lwd = 2)
```

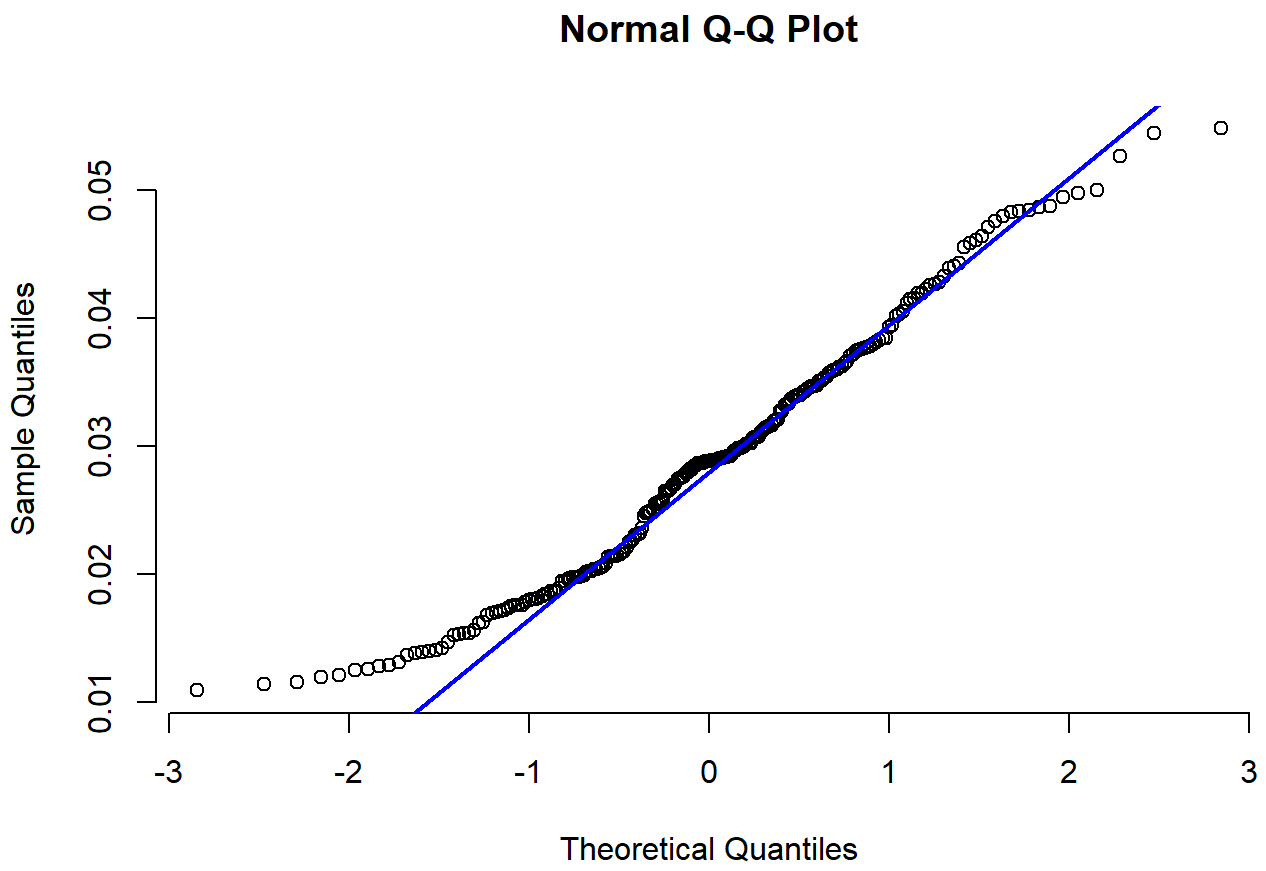



```
#shows a mostly normal distribution with heavy tails
#RunF-test to compare variances
ftest_IPO_Gross <- var.test(IPO_Gross ~ as.character(recession), data = combined_data_IPO_Gross)
ftest_IPO_Gross
```

```
##
##  F test to compare two variances
##
## data:  IPO_Gross by as.character(recession)
## F = 3.5107, num df = 188, denom df = 34, p-value = 5.271e-05
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.982961 5.649903
## sample estimates:
## ratio of variances
##           3.510694
```

```
#reject
```

```
#Equal variance test by recession for dp_ratio

#check normality first
qqnorm(combined_data_DPRatio$DPRatio, pch = 1, frame = FALSE)
qqline(combined_data_DPRatio$DPRatio, col = "blue", lwd = 2)
```

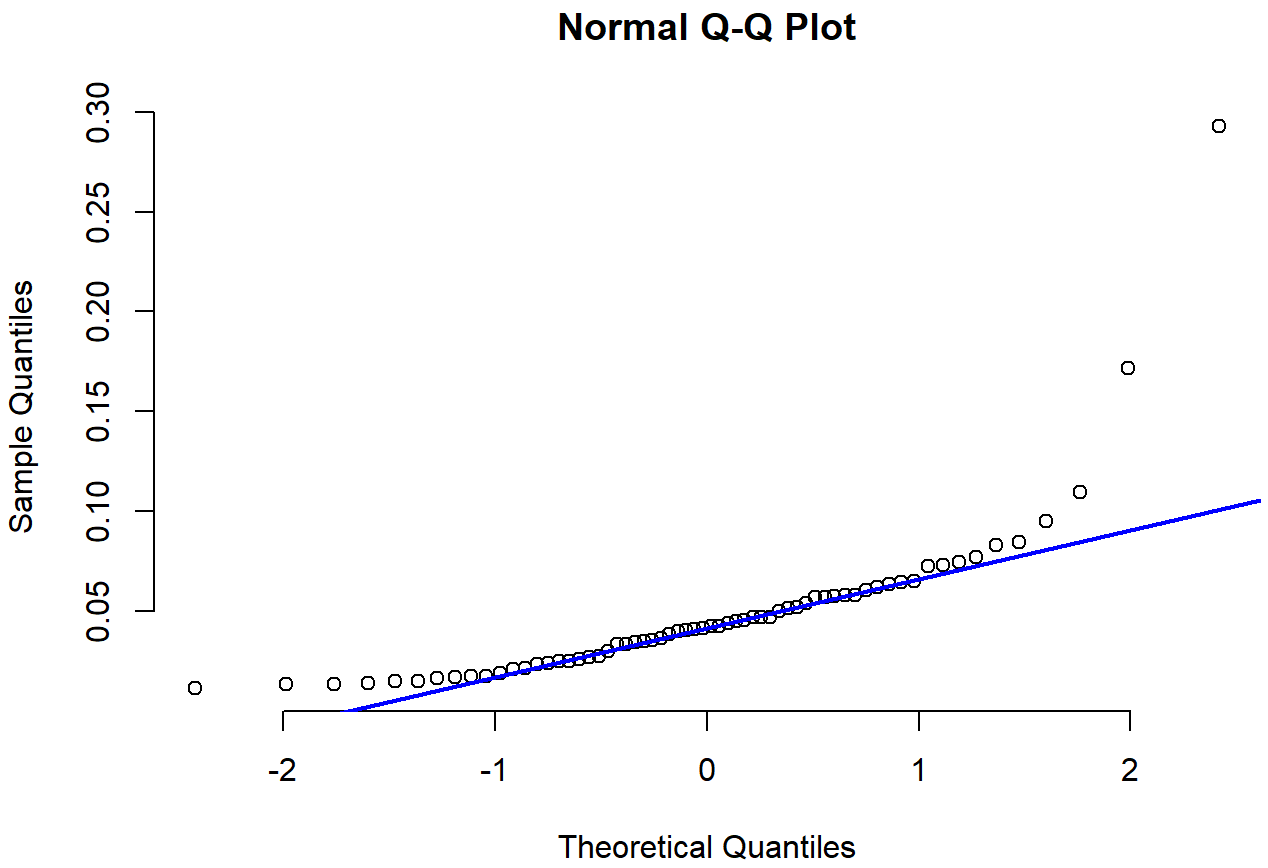



```
#Approximately normal distribution
#RunF-test to compare variances
ftest_DPRatio <- var.test(DPRatio ~ as.character(recession), data = combined_data_DPRatio)
ftest_DPRatio
```

```
##
##  F test to compare two variances
##
## data:  DPRatio by as.character(recession)
## F = 0.66412, num df = 188, denom df = 34, p-value = 0.09247
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3751205 1.0688028
## sample estimates:
## ratio of variances
##          0.6641246
```

```
#fail to reject
```

```
#Equal variance test by recession for EPBound

#check normality first
qqnorm(combined_data_EPBound$EPBound, pch = 1, frame = FALSE)
qqline(combined_data_EPBound$EPBound, col = "blue", lwd = 2)
```

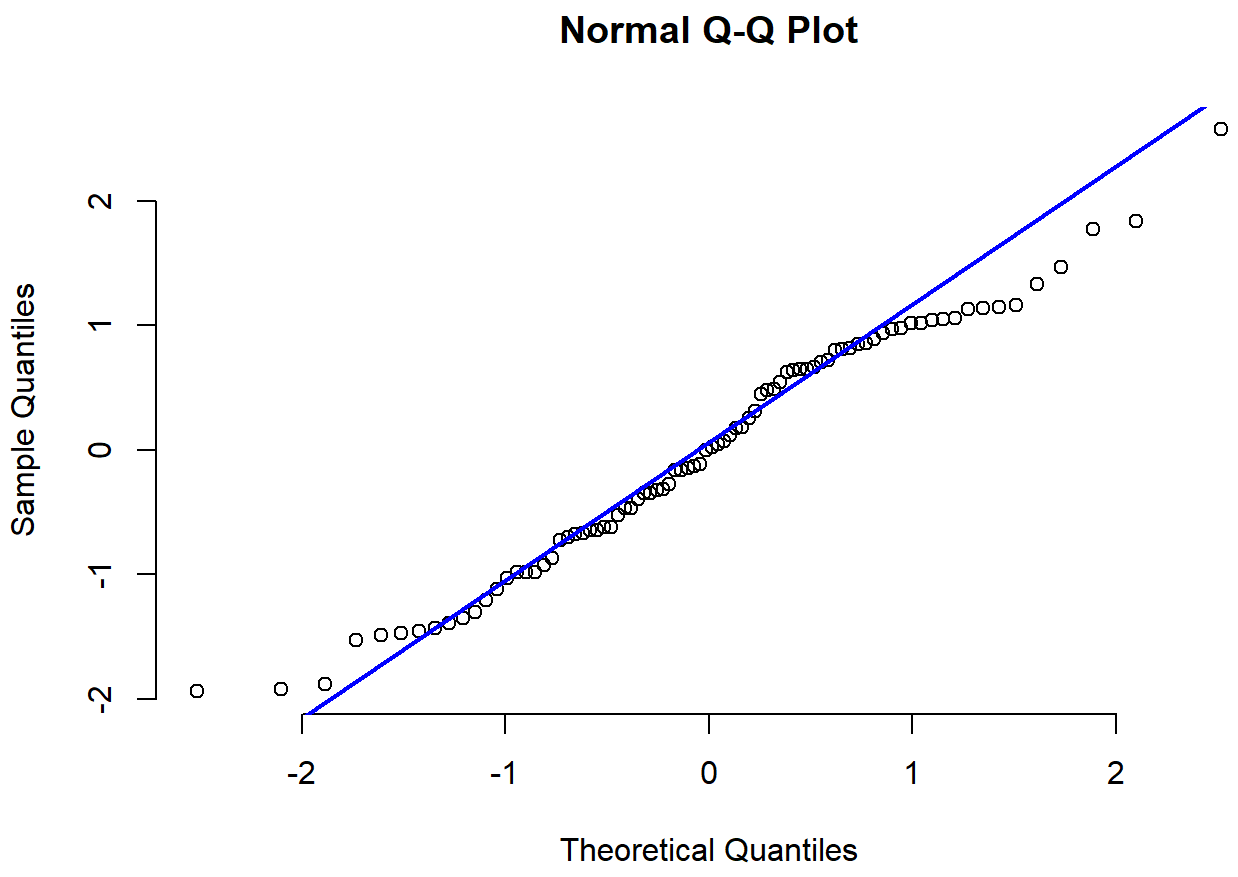



```
#shows an approximately, mostly normal distribution
#Run F-test to compare variances
ftest_EPBound <- var.test(EPBound ~ as.character(recession), data = combined_data_EPBound)
ftest_EPBound
```

```
##
##  F test to compare two variances
##
## data:  EPBound by as.character(recession)
## F = 0.06974, num df = 54, denom df = 8, p-value = 1.177e-10
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.01836795 0.16997633
## sample estimates:
## ratio of variances
##          0.0697402
```

```
#reject
```

```
#Equal variance test by recession for Aggregate Risk Aversion

#check normality first
qqnorm(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion, pch = 1, frame = FALSE)
qqline(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion, col = "blue", lwd = 2)
```

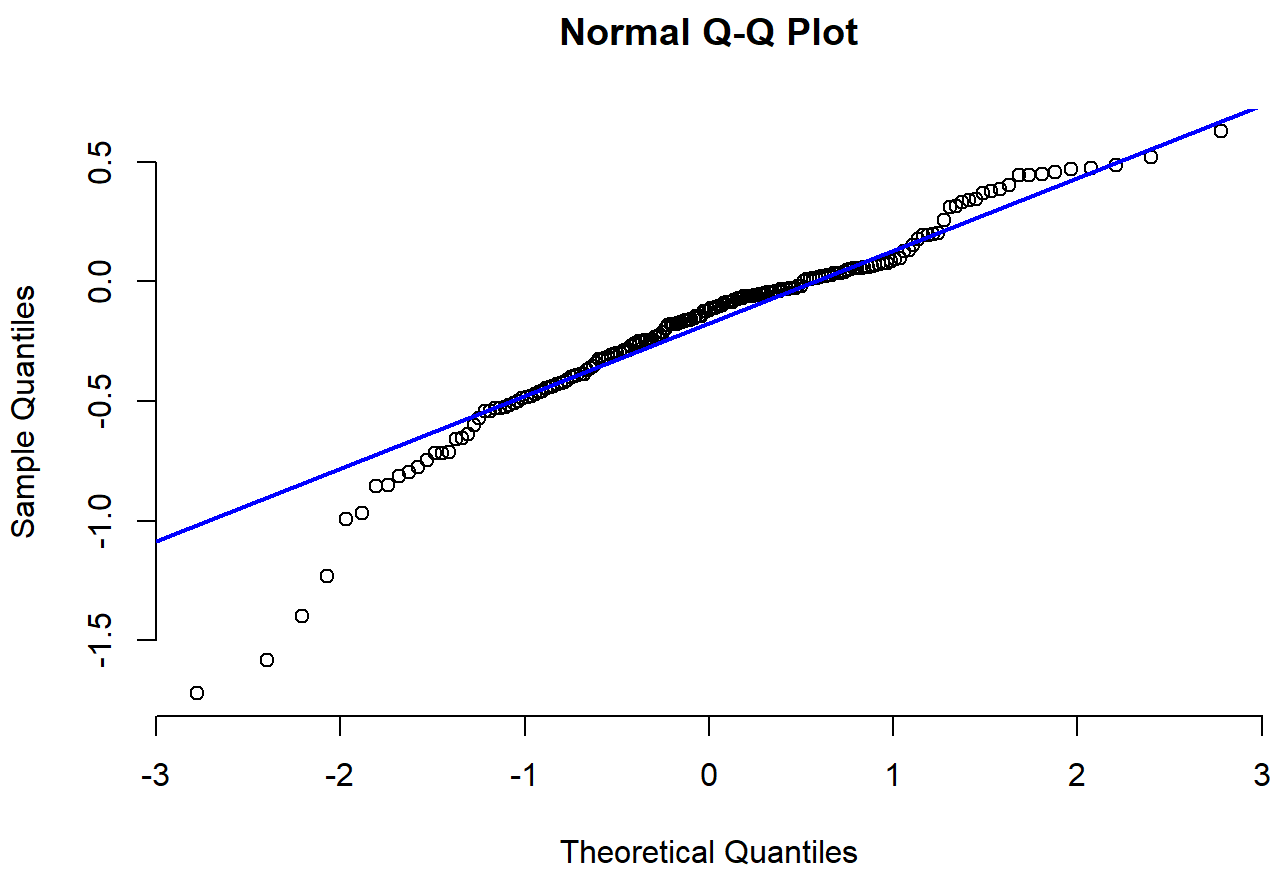


```
#shows an approximately, mostly normal distribution
#Run F-test to compare variances
ftest_Aggregate_Risk_Aversion <- var.test(Aggregate_Risk_Aversion ~ as.character(recession), dat
a = combined_data_Aggregate_Risk_Aversion)
ftest_Aggregate_Risk_Aversion
```

```
##
##  F test to compare two variances
##
## data:  Aggregate_Risk_Aversion by as.character(recession)
## F = 0.44192, num df = 71, denom df = 11, p-value = 0.03975
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.1480320 0.9634396
## sample estimates:
## ratio of variances
##          0.4419192
```

```
#reject
```

```
#Equal variance test by recession for Price of Volatile Stocks

#check normality first
qqnorm(as.numeric(combined_data_PVS$PVS), pch = 1, frame = FALSE)
qqline(as.numeric(combined_data_PVS$PVS), col = "blue", lwd = 2)
```

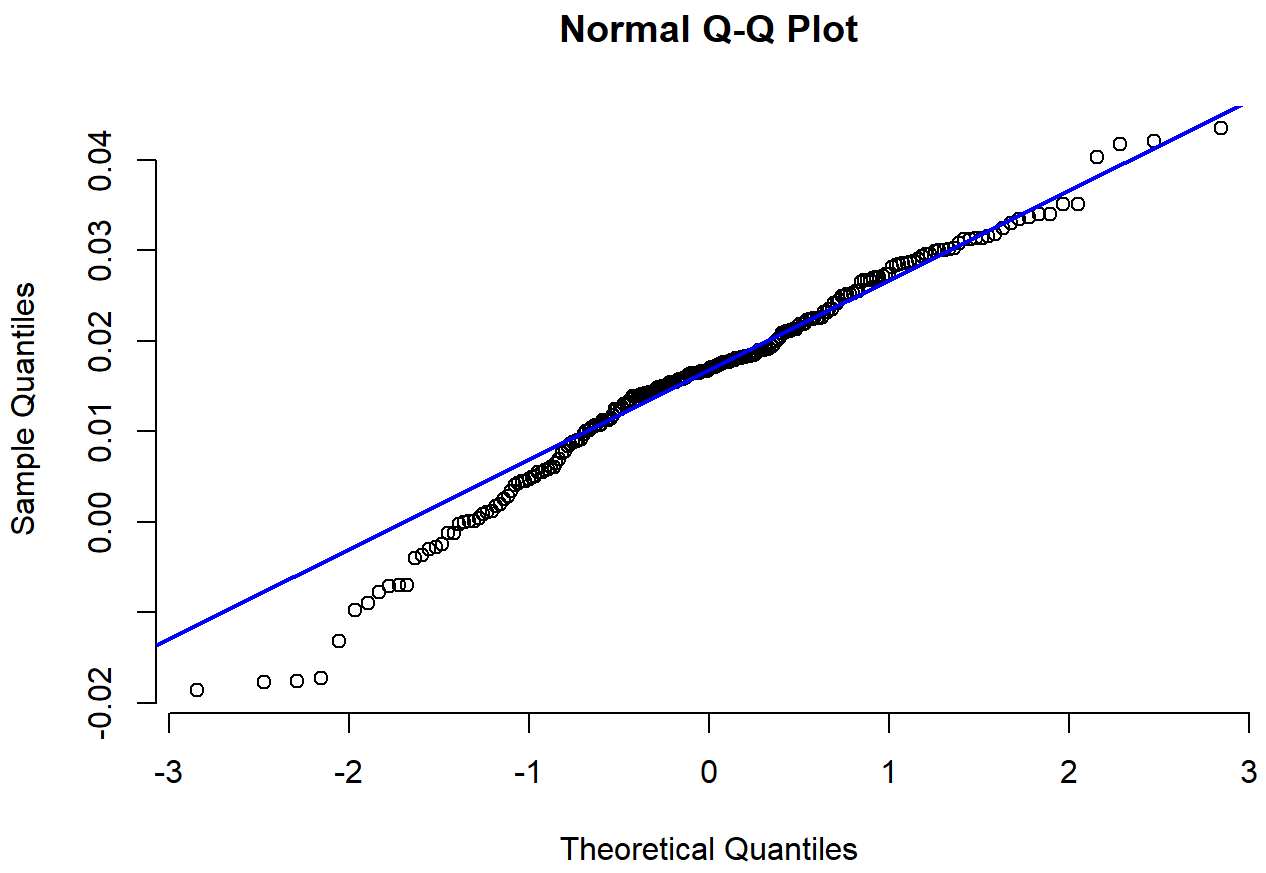



```
#shows an approximately, mostly normal distribution
#Run F-test to compare variances
ftest_Aggregate_PVS<- var.test(as.numeric(PVS) ~ as.character(recession), data = combined_data_P
VS)
ftest_Aggregate_PVS
```

```
##
##  F test to compare two variances
##
## data:  as.numeric(PVS) by as.character(recession)
## F = 0.17662, num df = 151, denom df = 30, p-value = 5.195e-13
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.09543969 0.29397508
## sample estimates:
## ratio of variances
##          0.1766216
```

```
#reject
```

```
#Equal variance test by recession for Surplus Consumption Ratio

#check normality first
qqnorm(combined_data_SCR$SCR, pch = 1, frame = FALSE)
qqline(combined_data_SCR$SCR, col = "blue", lwd = 2)
```

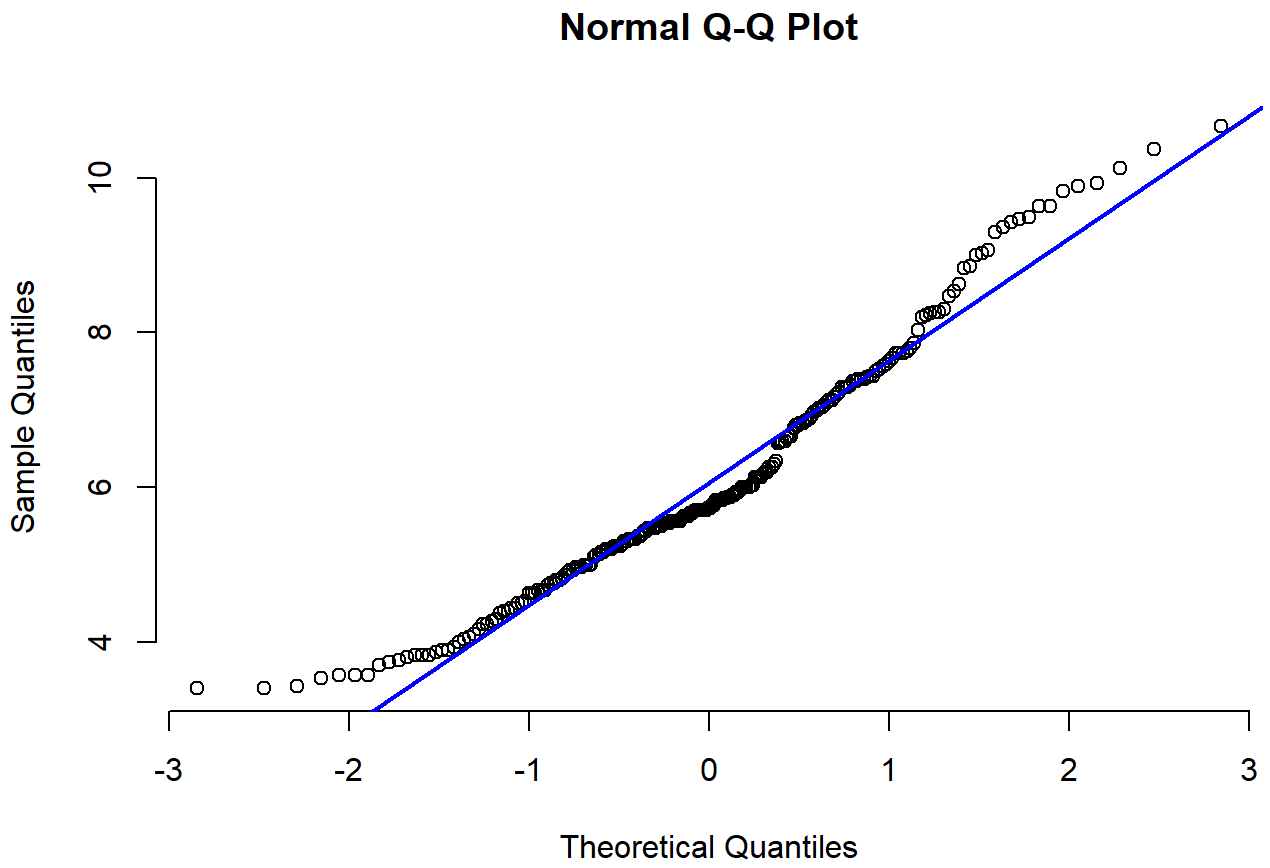



```
#shows an approximately, mostly normal distribution
#Run F-test to compare variances
ftest_SCR <- var.test(SCR ~ as.character(recession), data = combined_data_SCR)
ftest_SCR
```

```
##
##  F test to compare two variances
##
## data:  SCR by as.character(recession)
## F = 0.81969, num df = 188, denom df = 34, p-value = 0.4063
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.4629886 1.3191587
## sample estimates:
## ratio of variances
##          0.8196888
```

```
#reject
```

```
#Equal variance test by recession for Unemployment Rate
#check normality first
qqnorm(combined_data_UNE$UNE, pch = 1, frame = FALSE)
qqline(combined_data_UNE$UNE, col = "blue", lwd = 2)
```




```
#shows an approximately, mostly normal distribution
#Run F-test to compare variances
ftest_UNE <- var.test(UNE ~ as.character(recession), data = combined_data_UNE)
ftest_UNE
```

```
##
##  F test to compare two variances
##
## data:  UNE by as.character(recession)
## F = 0.85236, num df = 188, denom df = 34, p-value = 0.4988
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.4814438 1.3717416
## sample estimates:
## ratio of variances
##          0.8523624
```

```
#reject
```

```
#run tests for Heteroscedasticity
#Use Breusch-Pagan Test


#Start with Gross Ipo and GDP
IPO_Gross_GDP_lm <- lm(as.numeric(GDP) ~ IPO_Gross, data = combined_data_IPO_Gross)
combined_data_IPO_Gross$GDP_resi <- IPO_Gross_GDP_lm$residuals
IPO_Gross_GDP.func <- lm(GDP_resi^2 ~ IPO_Gross, data = combined_data_IPO_Gross)
summary(IPO_Gross_GDP.func)
```

```
##
## Call:
## lm(formula = GDP_resi^2 ~ IPO_Gross, data = combined_data_IPO_Gross)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.911 -4.940 -2.839  0.614 71.071
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.93165    1.11050   8.043 5.22e-14 ***
## IPO_Gross   -0.12330    0.03367  -3.662 0.000313 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.62 on 222 degrees of freedom
## Multiple R-squared:  0.05695,    Adjusted R-squared:  0.0527
## F-statistic: 13.41 on 1 and 222 DF,  p-value: 0.0003134
```

```
#Gross IPO and GNP
IPO_Gross_Gross.national.product_lm <- lm(as.numeric(Gross.national.product) ~ IPO_Gross, data =
combined_data_IPO_Gross)
combined_data_IPO_Gross$Gross.national.product_resi <- IPO_Gross_Gross.national.product_lm$resid
uals
IPO_Gross_Gross.national.product.func <- lm(Gross.national.product_resi^2 ~ IPO_Gross, data = co
mbined_data_IPO_Gross)
summary(IPO_Gross_Gross.national.product.func)
```

```
##
## Call:
## lm(formula = Gross.national.product_resi^2 ~ IPO_Gross, data = combined_data_IPO_Gross)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.942 -4.953 -2.852  0.455 71.006
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.96192    1.11439   8.042 5.25e-14 ***
## IPO_Gross   -0.12392    0.03379  -3.667 0.000307 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.65 on 222 degrees of freedom
## Multiple R-squared:  0.05712,    Adjusted R-squared:  0.05287
## F-statistic: 13.45 on 1 and 222 DF,  p-value: 0.000307
```

```
#Gross IPO and Nondurable goods
IPO_Gross_Nondurable.goods_lm <- lm(as.numeric(Nondurable.goods) ~ IPO_Gross, data = combined_da
ta_IPO_Gross)
combined_data_IPO_Gross$Nondurable.goods_resi <- IPO_Gross_Nondurable.goods_lm$residuals
IPO_Gross_Nondurable.goods.func <- lm(Nondurable.goods_resi^2 ~ IPO_Gross, data = combined_data_
IPO_Gross)
summary(IPO_Gross_Nondurable.goods.func)
```

```
##
## Call:
## lm(formula = Nondurable.goods_resi^2 ~ IPO_Gross, data = combined_data_IPO_Gross)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -34.76 -22.45 -12.46   2.60 678.46
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.7545     6.1074   5.854 1.71e-08 ***
## IPO_Gross    -0.5447     0.1852  -2.941  0.00362 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58.39 on 222 degrees of freedom
## Multiple R-squared:  0.03751,    Adjusted R-squared:  0.03317
## F-statistic: 8.651 on 1 and 222 DF,  p-value: 0.003615
```

```
#Gross IPO and Gross private domestic investment
IPO_Gross_Gross.private.domestic.investment_lm <- lm(as.numeric(Gross.private.domestic.investmen
t) ~ IPO_Gross, data = combined_data_IPO_Gross)
combined_data_IPO_Gross$Gross.private.domestic.investment_resi <- IPO_Gross_Gross.private.domest
ic.investment_lm$residuals
IPO_Gross_Gross.private.domestic.investment.func <- lm(Gross.private.domestic.investment_resi^2
~ IPO_Gross, data = combined_data_IPO_Gross)
summary(IPO_Gross_Gross.private.domestic.investment.func)
```

```
##
## Call:
## lm(formula = Gross.private.domestic.investment_resi^2 ~ IPO_Gross,
##     data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.221 -10.348  -5.353   1.831 150.959
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.82626    2.16896   8.219 1.71e-14 ***
## IPO_Gross   -0.26167    0.06577  -3.979 9.39e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.74 on 222 degrees of freedom
## Multiple R-squared:  0.06656,    Adjusted R-squared:  0.06235
## F-statistic: 15.83 on 1 and 222 DF,  p-value: 9.388e-05
```

```
#Gross IPO and Personal consumption expenditure
IPO_Gross_Personal.consumption.expenditures_lm <- lm(
as.numeric(Personal.consumption.expenditures) ~ IPO_Gross, data = combined_data_IPO_Gross)
combined_data_IPO_Gross$Personal.consumption.expenditures_resi <- IPO_Gross_Personal.consumptio
n.expenditures_lm$residuals
IPO_Gross_Personal.consumption.expenditures.func <- lm(
Personal.consumption.expenditures_resi^2 ~ IPO_Gross, data = combined_data_IPO_Gross)
summary(IPO_Gross_Personal.consumption.expenditures.func)
```

```
##
## Call:
## lm(formula = Personal.consumption.expenditures_resi^2 ~ IPO_Gross,
##     data = combined_data_IPO_Gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.647  -6.768  -3.773   0.705  91.487
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.7621     1.4313   8.218 1.72e-14 ***
## IPO_Gross    -0.1841     0.0434  -4.243 3.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.68 on 222 degrees of freedom
## Multiple R-squared:  0.075,  Adjusted R-squared:  0.07084
## F-statistic:    18 on 1 and 222 DF,  p-value: 3.243e-05
```

```
#Start with DPRatio and GDP
DPRatio_GDP_lm <- lm(as.numeric(GDP) ~ DPRatio, data = combined_data_DPRatio)
combined_data_DPRatio$GDP_resi <- DPRatio_GDP_lm$residuals
DPRatio_GDP.func <- lm(GDP_resi^2 ~ DPRatio, data = combined_data_DPRatio)
summary(DPRatio_GDP.func)
```

```
##
## Call:
## lm(formula = GDP_resi^2 ~ DPRatio, data = combined_data_DPRatio)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.475 -2.560 -1.289  0.947 32.596
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.353      1.058  -2.224   0.0272 *
## DPRatio      198.615     34.599   5.741 3.08e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.261 on 222 degrees of freedom
## Multiple R-squared:  0.1293, Adjusted R-squared:  0.1253
## F-statistic: 32.95 on 1 and 222 DF,  p-value: 3.078e-08
```

```
#DPRatio and GNP
DPRatio_Gross.national.product_lm <- lm(as.numeric(Gross.national.product) ~ DPRatio, data = com
bined_data_DPRatio)
combined_data_DPRatio$Gross.national.product_resi <- DPRatio_Gross.national.product_lm$residuals
DPRatio_Gross.national.product.func <- lm(Gross.national.product_resi^2 ~ DPRatio, data = combin
ed_data_DPRatio)
summary(DPRatio_Gross.national.product.func)
```

```
##
## Call:
## lm(formula = Gross.national.product_resi^2 ~ DPRatio, data = combined_data_DPRatio)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.510 -2.515 -1.268  0.949 32.494
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.376      1.061  -2.239   0.0261 *
## DPRatio      199.781     34.693   5.759 2.81e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.275 on 222 degrees of freedom
## Multiple R-squared:   0.13,  Adjusted R-squared:  0.126
## F-statistic: 33.16 on 1 and 222 DF,  p-value: 2.806e-08
```

```
#DPRatio and Nondurable goods
DPRatio_Nondurable.goods_lm <- lm(as.numeric(Nondurable.goods) ~ DPRatio, data = combined_data_D
PRatio)
combined_data_DPRatio$Nondurable.goods_resi <- DPRatio_Nondurable.goods_lm$residuals
DPRatio_Nondurable.goods.func <- lm(Nondurable.goods_resi^2 ~ DPRatio, data = combined_data_DPRa
tio)
summary(DPRatio_Nondurable.goods.func)
```

```
##
## Call:
## lm(formula = Nondurable.goods_resi^2 ~ DPRatio, data = combined_data_DPRatio)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -27.23 -18.70 -14.61  -2.63 652.79
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    12.16      11.35   1.071    0.285
## DPRatio       304.29     371.17   0.820    0.413
##
## Residual standard error: 56.44 on 222 degrees of freedom
## Multiple R-squared:  0.003018,   Adjusted R-squared:  -0.001473
## F-statistic: 0.6721 on 1 and 222 DF,  p-value: 0.4132
```

```
#DPRatio and Gross private domestic investment
DPRatio_Gross.private.domestic.investment_lm <- lm(as.numeric(Gross.private.domestic.investment)
~ DPRatio, data = combined_data_DPRatio)
combined_data_DPRatio$Gross.private.domestic.investment_resi <- DPRatio_Gross.private.domestic.i
nvestment_lm$residuals
DPRatio_Gross.private.domestic.investment.func <- lm(Gross.private.domestic.investment_resi^2 ~
DPRatio, data = combined_data_DPRatio)
summary(DPRatio_Gross.private.domestic.investment.func)
```

```
##
## Call:
## lm(formula = Gross.private.domestic.investment_resi^2 ~ DPRatio,
##     data = combined_data_DPRatio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.083  -6.063  -2.476   1.570  86.044
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -6.303      2.679  -2.353   0.0195 *
## DPRatio      502.453     87.585   5.737 3.14e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.32 on 222 degrees of freedom
## Multiple R-squared:  0.1291, Adjusted R-squared:  0.1252
## F-statistic: 32.91 on 1 and 222 DF,  p-value: 3.139e-08
```

```
#DPRatio and Personal consumption expenditure
DPRatio_Personal.consumption.expenditures_lm <- lm(
as.numeric(Personal.consumption.expenditures) ~ DPRatio, data = combined_data_DPRatio)
combined_data_DPRatio$Personal.consumption.expenditures_resi <- DPRatio_Personal.consumption.exp
enditures_lm$residuals
DPRatio_Personal.consumption.expenditures.func <- lm(
Personal.consumption.expenditures_resi^2 ~ DPRatio, data = combined_data_DPRatio)
summary(DPRatio_Personal.consumption.expenditures.func)
```

```
##
## Call:
## lm(formula = Personal.consumption.expenditures_resi^2 ~ DPRatio,
##     data = combined_data_DPRatio)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.244 -3.988 -1.955  0.641 94.427
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.051      2.018  -1.016 0.310583
## DPRatio      228.365     65.979   3.461 0.000645 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.03 on 222 degrees of freedom
## Multiple R-squared:  0.0512, Adjusted R-squared:  0.04693
## F-statistic: 11.98 on 1 and 222 DF,  p-value: 0.000645
```

```
#Start with EPBound and GDP
EPBound_GDP_lm <- lm(as.numeric(GDP) ~ EPBound, data = combined_data_EPBound)
combined_data_EPBound$GDP_resi <- EPBound_GDP_lm$residuals
EPBound_GDP.func <- lm(GDP_resi^2 ~ EPBound, data = combined_data_EPBound)
summary(EPBound_GDP.func)
```

```
##
## Call:
## lm(formula = GDP_resi^2 ~ EPBound, data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8019 -0.4858 -0.2803  0.1346  3.0227
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4433     0.1561   2.840  0.00609 **
## EPBound       3.5316     2.4486   1.442  0.15425
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7997 on 62 degrees of freedom
## Multiple R-squared:  0.03246,    Adjusted R-squared:  0.01686
## F-statistic:  2.08 on 1 and 62 DF,  p-value: 0.1542
```

```
#EPBound and GNP
EPBound_Gross.national.product_lm <- lm(as.numeric(Gross.national.product) ~ as.numeric(EPBoun
d), data = combined_data_EPBound)
combined_data_EPBound$Gross.national.product_resi <- EPBound_Gross.national.product_lm$residuals
EPBound_Gross.national.product.func <- lm(Gross.national.product_resi^2 ~ EPBound, data = combin
ed_data_EPBound)
summary(EPBound_Gross.national.product.func)
```

```
##
## Call:
## lm(formula = Gross.national.product_resi^2 ~ EPBound, data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7844 -0.4774 -0.2891  0.1146  3.0084
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4579     0.1543   2.968  0.00426 **
## EPBound       3.2144     2.4198   1.328  0.18892
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7903 on 62 degrees of freedom
## Multiple R-squared:  0.02767,    Adjusted R-squared:  0.01199
## F-statistic: 1.765 on 1 and 62 DF,  p-value: 0.1889
```

```
#EPBound and Nondurable goods
EPBound_Nondurable.goods_lm <- lm(as.numeric(Nondurable.goods) ~ as.numeric(EPBound), data = com
bined_data_EPBound)
combined_data_EPBound$Nondurable.goods_resi <- EPBound_Nondurable.goods_lm$residuals
EPBound_Nondurable.goods.func <- lm(Nondurable.goods_resi^2 ~ EPBound, data = combined_data_EPBo
und)
summary(EPBound_Nondurable.goods.func)
```

```
##
## Call:
## lm(formula = Nondurable.goods_resi^2 ~ EPBound, data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.871 -17.540 -12.602   7.521 187.177
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   18.371      6.163   2.981   0.0041 **
## EPBound        6.972     96.667   0.072   0.9427
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.57 on 62 degrees of freedom
## Multiple R-squared:  8.388e-05,  Adjusted R-squared:  -0.01604
## F-statistic: 0.005201 on 1 and 62 DF,  p-value: 0.9427
```

```
#EPBound and Gross private domestic investment
EPBound_Gross.private.domestic.investment_lm <- lm(as.numeric(Gross.private.domestic.investment)
~ as.numeric(EPBound), data = combined_data_EPBound)
combined_data_EPBound$Gross.private.domestic.investment_resi <- EPBound_Gross.private.domestic.i
nvestment_lm$residuals
EPBound_Gross.private.domestic.investment.func <- lm(Gross.private.domestic.investment_resi^2 ~
EPBound, data = combined_data_EPBound)
summary(EPBound_Gross.private.domestic.investment.func)
```

```
##
## Call:
## lm(formula = Gross.private.domestic.investment_resi^2 ~ EPBound,
##     data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.606  -4.932  -2.394   1.470  36.220
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.335      1.636  -0.816    0.418
## EPBound      125.621     25.665   4.895 7.35e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.382 on 62 degrees of freedom
## Multiple R-squared:  0.2787, Adjusted R-squared:  0.2671
## F-statistic: 23.96 on 1 and 62 DF,  p-value: 7.355e-06
```

```
#EPBound and Personal consumption expenditure
EPBound_Personal.consumption.expenditures_lm <- lm(
as.numeric(Personal.consumption.expenditures) ~ as.numeric(EPBound), data = combined_data_EPBoun
d)
combined_data_EPBound$Personal.consumption.expenditures_resi <- EPBound_Personal.consumption.exp
enditures_lm$residuals
EPBound_Personal.consumption.expenditures.func <- lm(
Personal.consumption.expenditures_resi^2 ~ EPBound, data = combined_data_EPBound)
summary(EPBound_Personal.consumption.expenditures.func)
```

```
##
## Call:
## lm(formula = Personal.consumption.expenditures_resi^2 ~ EPBound,
##     data = combined_data_EPBound)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3251 -1.2271 -0.6873  0.3326 13.0799
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.2737     0.4198   3.034  0.00352 **
## EPBound       0.8072     6.5853   0.123  0.90284
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.151 on 62 degrees of freedom
## Multiple R-squared:  0.0002423,  Adjusted R-squared:  -0.01588
## F-statistic: 0.01502 on 1 and 62 DF,  p-value: 0.9028
```

```
#Start with PVS and GDP
PVS_GDP_lm <- lm(GDP ~ as.numeric(PVS), data = combined_data_PVS)
combined_data_PVS$GDP_resi <- PVS_GDP_lm$residuals
PVS_GDP.func <- lm(GDP_resi^2 ~ as.numeric(PVS), data = combined_data_PVS)
summary(PVS_GDP.func)
```

```
##
## Call:
## lm(formula = GDP_resi^2 ~ as.numeric(PVS), data = combined_data_PVS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.054  -5.105  -3.573  -0.413  83.212
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        5.366      1.030   5.212 5.07e-07 ***
## as.numeric(PVS)   -6.679      2.522  -2.648  0.00882 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.55 on 181 degrees of freedom
## Multiple R-squared:  0.03729,    Adjusted R-squared:  0.03197
## F-statistic: 7.011 on 1 and 181 DF,  p-value: 0.008815
```

```
#PVS and GNP
PVS_Gross.national.product_lm <- lm(Gross.national.product ~ as.numeric(PVS), data = combined_da
ta_PVS)
combined_data_PVS$Gross.national.product_resi <- PVS_Gross.national.product_lm$residuals
PVS_Gross.national.product.func <- lm(Gross.national.product_resi^2 ~ as.numeric(PVS), data = co
mbined_data_PVS)
summary(PVS_Gross.national.product.func)
```

```
##
## Call:
## lm(formula = Gross.national.product_resi^2 ~ as.numeric(PVS),
##     data = combined_data_PVS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.071  -5.157  -3.577  -0.406  83.108
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        5.382      1.032   5.216 4.95e-07 ***
## as.numeric(PVS)   -6.682      2.528  -2.644  0.00892 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.58 on 181 degrees of freedom
## Multiple R-squared:  0.03718,    Adjusted R-squared:  0.03186
## F-statistic: 6.989 on 1 and 181 DF,  p-value: 0.00892
```

```
#PVS and Nondurable goods
PVS_Nondurable.goods_lm <- lm(Nondurable.goods ~ as.numeric(PVS), data = combined_data_PVS)
combined_data_PVS$Nondurable.goods_resi <- PVS_Nondurable.goods_lm$residuals
PVS_Nondurable.goods.func <- lm(Nondurable.goods_resi^2 ~ as.numeric(PVS), data = combined_data_
PVS)
summary(PVS_Nondurable.goods.func)
```

```
##
## Call:
## lm(formula = Nondurable.goods_resi^2 ~ as.numeric(PVS), data = combined_data_PVS)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -63.81 -23.31 -14.14   3.28 585.84
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       19.229      5.027   3.825 0.000180 ***
## as.numeric(PVS)  -41.876     12.316  -3.400 0.000829 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.29 on 181 degrees of freedom
## Multiple R-squared:  0.06004,    Adjusted R-squared:  0.05485
## F-statistic: 11.56 on 1 and 181 DF,  p-value: 0.0008286
```

```
#EPBound and Gross private domestic investment
PVS_Gross.private.domestic.investment_lm <- lm(Gross.private.domestic.investment ~ as.numeric(PV
S), data = combined_data_PVS)
combined_data_PVS$Gross.private.domestic.investment_resi <- PVS_Gross.private.domestic.investmen
t_lm$residuals
PVS_Gross.private.domestic.investment.func <- lm(Gross.private.domestic.investment_resi^2 ~ as.n
umeric(PVS), data = combined_data_PVS)
summary(PVS_Gross.private.domestic.investment.func)
```

```
##
## Call:
## lm(formula = Gross.private.domestic.investment_resi^2 ~ as.numeric(PVS),
##     data = combined_data_PVS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.686 -11.005  -6.805   1.751 164.389
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        8.823      1.937   4.556 9.56e-06 ***
## as.numeric(PVS)  -23.814      4.744  -5.019 1.23e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.61 on 181 degrees of freedom
## Multiple R-squared:  0.1222, Adjusted R-squared:  0.1173
## F-statistic: 25.19 on 1 and 181 DF,  p-value: 1.234e-06
```

```
#EPBound and Personal consumption expenditure
PVS_Personal.consumption.expenditures_lm <- lm(
Personal.consumption.expenditures ~ as.numeric(PVS), data = combined_data_PVS)
combined_data_PVS$Personal.consumption.expenditures_resi <- PVS_Personal.consumption.expenditure
s_lm$residuals
PVS_Personal.consumption.expenditures.func <- lm(
Personal.consumption.expenditures_resi^2 ~ as.numeric(PVS), data = combined_data_PVS)
summary(PVS_Personal.consumption.expenditures.func)
```

```
##
## Call:
## lm(formula = Personal.consumption.expenditures_resi^2 ~ as.numeric(PVS),
##     data = combined_data_PVS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.219  -6.603  -4.536  -0.877  75.581
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.577      1.241   5.298 3.38e-07 ***
## as.numeric(PVS)   -8.281      3.041  -2.723  0.00711 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.13 on 181 degrees of freedom
## Multiple R-squared:  0.03934,    Adjusted R-squared:  0.03404
## F-statistic: 7.413 on 1 and 181 DF,  p-value: 0.007109
```

```
#Start with SCR and GDP
SCR_GDP_lm <- lm(GDP ~ as.numeric(SCR), data = combined_data_SCR)
combined_data_SCR$GDP_resi <- SCR_GDP_lm$residuals
SCR_GDP.func <- lm(GDP_resi^2 ~ as.numeric(SCR), data = combined_data_SCR)
summary(SCR_GDP.func)
```

```
##
## Call:
## lm(formula = GDP_resi^2 ~ as.numeric(SCR), data = combined_data_SCR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.032  -4.970  -2.321   1.210  73.047
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       11.590      1.148  10.098  < 2e-16 ***
## as.numeric(SCR) -351.909     58.183  -6.048 6.15e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10 on 222 degrees of freedom
## Multiple R-squared:  0.1415, Adjusted R-squared:  0.1376
## F-statistic: 36.58 on 1 and 222 DF,  p-value: 6.148e-09
```

```
#SCR and GNP
SCR_Gross.national.product_lm <- lm(Gross.national.product ~ as.numeric(SCR), data = combined_da
ta_SCR)
combined_data_SCR$Gross.national.product_resi <- SCR_Gross.national.product_lm$residuals
SCR_Gross.national.product.func <- lm(Gross.national.product_resi^2 ~ as.numeric(SCR), data = co
mbined_data_SCR)
summary(SCR_Gross.national.product.func)
```

```
##
## Call:
## lm(formula = Gross.national.product_resi^2 ~ as.numeric(SCR),
##     data = combined_data_SCR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.076  -4.879  -2.320   1.191  72.984
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       11.623      1.153  10.084  < 2e-16 ***
## as.numeric(SCR) -353.105     58.434  -6.043 6.33e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.04 on 222 degrees of freedom
## Multiple R-squared:  0.1413, Adjusted R-squared:  0.1374
## F-statistic: 36.52 on 1 and 222 DF,  p-value: 6.329e-09
```

```
#SCR and Nondurable goods
SCR_Nondurable.goods_lm <- lm(Nondurable.goods ~ as.numeric(SCR), data = combined_data_SCR)
combined_data_SCR$Nondurable.goods_resi <- SCR_Nondurable.goods_lm$residuals
SCR_Nondurable.goods.func <- lm(Nondurable.goods_resi^2 ~ as.numeric(SCR), data = combined_data_
SCR)
summary(SCR_Nondurable.goods.func)
```

```
##
## Call:
## lm(formula = Nondurable.goods_resi^2 ~ as.numeric(SCR), data = combined_data_SCR)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -67.16 -21.14  -9.59   2.49 627.23
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        48.537      6.583   7.373 3.27e-12 ***
## as.numeric(SCR) -1615.308    333.722  -4.840 2.43e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.36 on 222 degrees of freedom
## Multiple R-squared:  0.09546,    Adjusted R-squared:  0.09138
## F-statistic: 23.43 on 1 and 222 DF,  p-value: 2.427e-06
```

```
#SCR and Gross private domestic investment
SCR_Gross.private.domestic.investment_lm <- lm(Gross.private.domestic.investment ~ as.numeric(SC
R), data = combined_data_SCR)
combined_data_SCR$Gross.private.domestic.investment_resi <- SCR_Gross.private.domestic.investmen
t_lm$residuals
SCR_Gross.private.domestic.investment.func <- lm(Gross.private.domestic.investment_resi^2 ~ as.n
umeric(SCR), data = combined_data_SCR)
summary(SCR_Gross.private.domestic.investment.func)
```

```
##
## Call:
## lm(formula = Gross.private.domestic.investment_resi^2 ~ as.numeric(SCR),
##     data = combined_data_SCR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.587 -10.359  -4.774   2.318 162.944
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        23.60       2.41   9.791  < 2e-16 ***
## as.numeric(SCR)  -739.30     122.18  -6.051 6.06e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21 on 222 degrees of freedom
## Multiple R-squared:  0.1416, Adjusted R-squared:  0.1377
## F-statistic: 36.61 on 1 and 222 DF,  p-value: 6.062e-09
```

```
#SCR and Personal consumption expenditure
SCR_Personal.consumption.expenditures_lm <- lm(
Personal.consumption.expenditures ~ as.numeric(SCR), data = combined_data_SCR)
combined_data_SCR$Personal.consumption.expenditures_resi <- SCR_Personal.consumption.expenditure
s_lm$residuals
SCR_Personal.consumption.expenditures.func <- lm(
Personal.consumption.expenditures_resi^2 ~ as.numeric(SCR), data = combined_data_SCR)
summary(SCR_Personal.consumption.expenditures.func)
```

```
##
## Call:
## lm(formula = Personal.consumption.expenditures_resi^2 ~ as.numeric(SCR),
##     data = combined_data_SCR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.606  -6.681  -3.076   1.416  86.303
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       14.949      1.548   9.658  < 2e-16 ***
## as.numeric(SCR) -481.476     78.463  -6.136 3.84e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.49 on 222 degrees of freedom
## Multiple R-squared:  0.145,  Adjusted R-squared:  0.1412
## F-statistic: 37.65 on 1 and 222 DF,  p-value: 3.837e-09
```

```
#Start with Aggregate Risk Aversion and GDP
Aggregate_Risk_Aversion_GDP_lm <- lm(GDP ~ Aggregate_Risk_Aversion, data = combined_data_Aggrega
te_Risk_Aversion)
combined_data_Aggregate_Risk_Aversion$GDP_resi <- Aggregate_Risk_Aversion_GDP_lm$residuals
Aggregate_Risk_Aversion_GDP.func <- lm(GDP_resi^2 ~ Aggregate_Risk_Aversion, data = combined_dat
a_Aggregate_Risk_Aversion)
summary(Aggregate_Risk_Aversion_GDP.func)
```

```
##
## Call:
## lm(formula = GDP_resi^2 ~ Aggregate_Risk_Aversion, data = combined_data_Aggregate_Risk_Aversi
on)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2682 -0.7960 -0.4509  0.2165  5.6833
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.8500     0.1465   5.802 1.19e-07 ***
## Aggregate_Risk_Aversion    0.2863     0.1504   1.904   0.0605 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.343 on 82 degrees of freedom
## Multiple R-squared:  0.04233,    Adjusted R-squared:  0.03065
## F-statistic: 3.624 on 1 and 82 DF,  p-value: 0.06046
```

```
#Aggregate Risk Aversion and GNP
Aggregate_Risk_Aversion_Gross.national.product_lm <- lm(Gross.national.product ~
Aggregate_Risk_Aversion, data = combined_data_Aggregate_Risk_Aversion)
combined_data_Aggregate_Risk_Aversion$Gross.national.product_resi <- Aggregate_Risk_Aversion_Gro
ss.national.product_lm$residuals
Aggregate_Risk_Aversion_Gross.national.product.func <- lm(Gross.national.product_resi^2 ~ Aggreg
ate_Risk_Aversion, data = combined_data_Aggregate_Risk_Aversion)
summary(Aggregate_Risk_Aversion_Gross.national.product.func)
```

```
##
## Call:
## lm(formula = Gross.national.product_resi^2 ~ Aggregate_Risk_Aversion,
##     data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3108 -0.8144 -0.4598  0.2131  5.6971
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.8650     0.1491   5.800  1.2e-07 ***
## Aggregate_Risk_Aversion    0.3044     0.1531   1.988   0.0501 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.367 on 82 degrees of freedom
## Multiple R-squared:  0.04598,    Adjusted R-squared:  0.03435
## F-statistic: 3.952 on 1 and 82 DF,  p-value: 0.05015
```

```
#Aggregate Risk Aversion and Nondurable goods
Aggregate_Risk_Aversion_Nondurable.goods_lm <- lm(Nondurable.goods ~ Aggregate_Risk_Aversion, da
ta = combined_data_Aggregate_Risk_Aversion)
combined_data_Aggregate_Risk_Aversion$Nondurable.goods_resi <- Aggregate_Risk_Aversion_Nondurabl
e.goods_lm$residuals
Aggregate_Risk_Aversion_Nondurable.goods.func <- lm(Nondurable.goods_resi^2 ~ Aggregate_Risk_Ave
rsion, data = combined_data_Aggregate_Risk_Aversion)
summary(Aggregate_Risk_Aversion_Nondurable.goods.func)
```

```
##
## Call:
## lm(formula = Nondurable.goods_resi^2 ~ Aggregate_Risk_Aversion,
##     data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -28.60 -21.63 -18.76  -7.95 597.51
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               23.621      7.841   3.013  0.00344 **
## Aggregate_Risk_Aversion   -2.569      8.049  -0.319  0.75040
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 71.86 on 82 degrees of freedom
## Multiple R-squared:  0.001241,   Adjusted R-squared:  -0.01094
## F-statistic: 0.1019 on 1 and 82 DF,  p-value: 0.7504
```

```
#Aggregate Risk Aversion and Gross private domestic investment
Aggregate_Risk_Aversion_Gross.private.domestic.investment_lm <- lm(Gross.private.domestic.invest
ment ~ Aggregate_Risk_Aversion, data = combined_data_Aggregate_Risk_Aversion)
combined_data_Aggregate_Risk_Aversion$Gross.private.domestic.investment_resi <- Aggregate_Risk_A
version_Gross.private.domestic.investment_lm$residuals
Aggregate_Risk_Aversion_Gross.private.domestic.investment.func <- lm(Gross.private.domestic.inve
stment_resi^2 ~ Aggregate_Risk_Aversion, data = combined_data_Aggregate_Risk_Aversion)
summary(Aggregate_Risk_Aversion_Gross.private.domestic.investment.func)
```

```
##
## Call:
## lm(formula = Gross.private.domestic.investment_resi^2 ~ Aggregate_Risk_Aversion,
##     data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.567 -3.688 -1.969  0.076 35.599
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.8787     0.7811   4.965 3.67e-06 ***
## Aggregate_Risk_Aversion   0.6528     0.8019   0.814    0.418
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.159 on 82 degrees of freedom
## Multiple R-squared:  0.008018,   Adjusted R-squared:  -0.00408
## F-statistic: 0.6628 on 1 and 82 DF,  p-value: 0.4179
```

```
#Aggregate Risk Aversion and Personal consumption expenditure
Aggregate_Risk_Aversion_Personal.consumption.expenditures_lm <- lm(
Personal.consumption.expenditures ~ Aggregate_Risk_Aversion, data = combined_data_Aggregate_Risk
_Aversion)
combined_data_Aggregate_Risk_Aversion$Personal.consumption.expenditures_resi <- Aggregate_Risk_A
version_Personal.consumption.expenditures_lm$residuals
Aggregate_Risk_Aversion_Personal.consumption.expenditures.func <- lm(
Personal.consumption.expenditures_resi^2 ~ Aggregate_Risk_Aversion, data = combined_data_Aggrega
te_Risk_Aversion)
summary(Aggregate_Risk_Aversion_Personal.consumption.expenditures.func)
```

```
##
## Call:
## lm(formula = Personal.consumption.expenditures_resi^2 ~ Aggregate_Risk_Aversion,
##     data = combined_data_Aggregate_Risk_Aversion)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.749 -2.419 -2.136 -0.567 67.720
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.4897     0.9023   2.759  0.00714 **
## Aggregate_Risk_Aversion   0.2400     0.9263   0.259  0.79618
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.27 on 82 degrees of freedom
## Multiple R-squared:  0.0008182,  Adjusted R-squared:  -0.01137
## F-statistic: 0.06715 on 1 and 82 DF,  p-value: 0.7962
```

```
#Start with UNE and GDP
UNE_GDP_lm <- lm(GDP ~ UNE, data = combined_data_UNE)
combined_data_UNE$GDP_resi <- UNE_GDP_lm$residuals
UNE_GDP.func <- lm(GDP_resi^2 ~ UNE, data = combined_data_UNE)
summary(UNE_GDP.func)
```

```
##
## Call:
## lm(formula = GDP_resi^2 ~ UNE, data = combined_data_UNE)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.642 -4.990 -3.496 -0.111 85.777
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0597     3.1056   0.341    0.733
## UNE           0.8120     0.4935   1.645    0.101
##
## Residual standard error: 11.68 on 222 degrees of freedom
## Multiple R-squared:  0.01205,    Adjusted R-squared:  0.007597
## F-statistic: 2.707 on 1 and 222 DF,  p-value: 0.1013
```

```
#UNE and GNP
UNE_Gross.national.product_lm <- lm(Gross.national.product ~ UNE, data = combined_data_UNE)
combined_data_UNE$Gross.national.product_resi <- UNE_Gross.national.product_lm$residuals
UNE_Gross.national.product.func <- lm(Gross.national.product_resi^2 ~ UNE, data = combined_data_
UNE)
summary(UNE_Gross.national.product.func)
```

```
##
## Call:
## lm(formula = Gross.national.product_resi^2 ~ UNE, data = combined_data_UNE)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.690 -5.004 -3.426 -0.117 85.692
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0293     3.1138   0.331   0.7413
## UNE           0.8191     0.4948   1.655   0.0993 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.71 on 222 degrees of freedom
## Multiple R-squared:  0.01219,    Adjusted R-squared:  0.007743
## F-statistic:  2.74 on 1 and 222 DF,  p-value: 0.09926
```

```
#UNE and Nondurable goods
UNE_Nondurable.goods_lm <- lm(Nondurable.goods ~ UNE, data = combined_data_UNE)
combined_data_UNE$Nondurable.goods_resi <- UNE_Nondurable.goods_lm$residuals
UNE_Nondurable.goods.func <- lm(Nondurable.goods_resi^2 ~ UNE, data = combined_data_UNE)
summary(UNE_Nondurable.goods.func)
```

```
##
## Call:
## lm(formula = Nondurable.goods_resi^2 ~ UNE, data = combined_data_UNE)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -24.75 -21.00 -17.28  -5.82 629.00
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.5417    15.4499   1.200    0.231
## UNE           0.6759     2.4551   0.275    0.783
##
## Residual standard error: 58.09 on 222 degrees of freedom
## Multiple R-squared:  0.0003413,  Adjusted R-squared:  -0.004162
## F-statistic: 0.07579 on 1 and 222 DF,  p-value: 0.7833
```

```
#UNE and Gross private domestic investment
UNE_Gross.private.domestic.investment_lm <- lm(Gross.private.domestic.investment ~ UNE, data = c
ombined_data_UNE)
combined_data_UNE$Gross.private.domestic.investment_resi <- UNE_Gross.private.domestic.investmen
t_lm$residuals
UNE_Gross.private.domestic.investment.func <- lm(Gross.private.domestic.investment_resi^2 ~ UNE,
data = combined_data_UNE)
summary(UNE_Gross.private.domestic.investment.func)
```

```
##
## Call:
## lm(formula = Gross.private.domestic.investment_resi^2 ~ UNE,
##     data = combined_data_UNE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.498 -10.315  -5.962  -0.707 182.851
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.5139     6.2710  -0.082   0.9348
## UNE           2.0229     0.9965   2.030   0.0435 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.58 on 222 degrees of freedom
## Multiple R-squared:  0.01822,    Adjusted R-squared:  0.0138
## F-statistic: 4.121 on 1 and 222 DF,  p-value: 0.04355
```

```
#UNE and Personal consumption expenditure
UNE_Personal.consumption.expenditures_lm <- lm(
Personal.consumption.expenditures ~ UNE, data = combined_data_UNE)
combined_data_UNE$Personal.consumption.expenditures_resi <- UNE_Personal.consumption.expenditure
s_lm$residuals
UNE_Personal.consumption.expenditures.func <- lm(
Personal.consumption.expenditures_resi^2 ~ UNE, data = combined_data_UNE)
summary(UNE_Personal.consumption.expenditures.func)
```

```
##
## Call:
## lm(formula = Personal.consumption.expenditures_resi^2 ~ UNE,
##     data = combined_data_UNE)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.741 -6.458 -4.796 -1.603 86.159
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6655     3.9852   0.920    0.359
## UNE           0.5921     0.6333   0.935    0.351
##
## Residual standard error: 14.98 on 222 degrees of freedom
## Multiple R-squared:  0.003922,   Adjusted R-squared:  -0.0005645
## F-statistic: 0.8742 on 1 and 222 DF,  p-value: 0.3508
```

```
combined_data_IPO_Gross$Year = as.integer(unlist(combined_data_IPO_Gross$Year))
IPO_Gross_Year <- as.data.frame(table(unlist(combined_data_IPO_Gross$Year)))
#line plot of gross_ipo by recession over time
#ggplot(data=combined_data_IPO_Gross, aes(x= Goods.2, y= IPO_Gross, group= recession)) +
  #geom_line(linetype="dashed", color="red", size=1.4)+
  #geom_point(color="green", size=5)
```

```
#create function for growth rates for different output measures
growth_rate = function(months){
  total <- c()
  total[1] = 0
  i = 1
  while(i < length(months)) {
      len_total <- length(total)
      #calculate growth rate
      if (months[i] == 0) {
        #??? what to do
        #for now just set to 0
        total[[len_total + 1]] = 0
      }

      else {
        total[[len_total + 1]] <- (months[i + 1] - months[i]) / months[i]
      }
    i = i + 1
  }
  return (as.numeric(total))
}
```

```
#run growth rate function on gdp measures in each dataset
#Start with IPO Gross
combined_data_IPO_Gross$GDP_growth = growth_rate(as.numeric(combined_data_IPO_Gross$GDP))
combined_data_IPO_Gross$Personal.consumption.expenditures_growth = growth_rate(as.numeric(combin
ed_data_IPO_Gross$Personal.consumption.expenditures))
combined_data_IPO_Gross$Gross.national.product_growth = growth_rate(as.numeric(combined_data_IPO
_Gross$Gross.national.product))
combined_data_IPO_Gross$Nondurable.goods_growth = growth_rate(as.numeric(combined_data_IPO_Gross
$Nondurable.goods))
combined_data_IPO_Gross$Gross.private.domestic.investment_growth = growth_rate(as.numeric(combin
ed_data_IPO_Gross$Gross.private.domestic.investment))

#DPRatio Growth Adjustments
combined_data_DPRatio$GDP_growth = growth_rate(as.numeric(combined_data_DPRatio$GDP))
combined_data_DPRatio$Personal.consumption.expenditures_growth = growth_rate(as.numeric(combined
_data_DPRatio$Personal.consumption.expenditures))
combined_data_DPRatio$Gross.national.product_growth = growth_rate(as.numeric(combined_data_DPRat
io$Gross.national.product))
combined_data_DPRatio$Nondurable.goods_growth = growth_rate(as.numeric(combined_data_DPRatio$Non
durable.goods))
combined_data_DPRatio$Gross.private.domestic.investment_growth = growth_rate(as.numeric(combined
_data_DPRatio$Gross.private.domestic.investment))


#EPBound Growth Adjustments
combined_data_EPBound$GDP_growth = growth_rate(as.numeric(combined_data_EPBound$GDP))
combined_data_EPBound$Personal.consumption.expenditures_growth = growth_rate(as.numeric(combined
_data_EPBound$Personal.consumption.expenditures))
combined_data_EPBound$Gross.national.product_growth = growth_rate(as.numeric(combined_data_EPBou
nd$Gross.national.product))
combined_data_EPBound$Nondurable.goods_growth = growth_rate(as.numeric(combined_data_EPBound$Non
durable.goods))
combined_data_EPBound$Gross.private.domestic.investment_growth = growth_rate(as.numeric(combined
_data_EPBound$Gross.private.domestic.investment))


#Price of Volatile Stocks Growth Adjustments
combined_data_PVS$GDP_growth = growth_rate(as.numeric(combined_data_PVS$GDP))
combined_data_PVS$Personal.consumption.expenditures_growth = growth_rate(as.numeric(combined_dat
a_PVS$Personal.consumption.expenditures))
combined_data_PVS$Gross.national.product_growth = growth_rate(as.numeric(combined_data_PVS$Gros
s.national.product))
combined_data_PVS$Nondurable.goods_growth = growth_rate(as.numeric(combined_data_PVS$Nondurable.
goods))
combined_data_PVS$Gross.private.domestic.investment_growth = growth_rate(as.numeric(combined_dat
a_PVS$Gross.private.domestic.investment))

#Surplus Consumption Ratio Growth Adjustments
combined_data_SCR$GDP_growth = growth_rate(as.numeric(combined_data_SCR$GDP))
combined_data_SCR$Personal.consumption.expenditures_growth = growth_rate(as.numeric(combined_dat
a_SCR$Personal.consumption.expenditures))
combined_data_SCR$Gross.national.product_growth = growth_rate(as.numeric(combined_data_SCR$Gros
```

```
s.national.product))
combined_data_SCR$Nondurable.goods_growth = growth_rate(as.numeric(combined_data_SCR$Nondurable.
goods))
combined_data_SCR$Gross.private.domestic.investment_growth = growth_rate(as.numeric(combined_dat
a_SCR$Gross.private.domestic.investment))

#Aggregate Risk Aversion Growth Adjustments
combined_data_Aggregate_Risk_Aversion$GDP_growth = growth_rate(as.numeric(combined_data_Aggregat
e_Risk_Aversion$GDP))
combined_data_Aggregate_Risk_Aversion$Personal.consumption.expenditures_growth = growth_rate(as.
numeric(combined_data_Aggregate_Risk_Aversion$Personal.consumption.expenditures))
combined_data_Aggregate_Risk_Aversion$Gross.national.product_growth = growth_rate(as.numeric(com
bined_data_Aggregate_Risk_Aversion$Gross.national.product))
combined_data_Aggregate_Risk_Aversion$Nondurable.goods_growth = growth_rate(as.numeric(combined_
data_Aggregate_Risk_Aversion$Nondurable.goods))
combined_data_Aggregate_Risk_Aversion$Gross.private.domestic.investment_growth = growth_rate(as.
numeric(combined_data_Aggregate_Risk_Aversion$Gross.private.domestic.investment))

#Unemployment Rate Growth Adjustments
combined_data_UNE$GDP_growth = growth_rate(as.numeric(combined_data_UNE$GDP))
combined_data_UNE$Personal.consumption.expenditures_growth = growth_rate(as.numeric(combined_dat
a_UNE$Personal.consumption.expenditures))
combined_data_UNE$Gross.national.product_growth = growth_rate(as.numeric(combined_data_UNE$Gros
s.national.product))
combined_data_UNE$Nondurable.goods_growth = growth_rate(as.numeric(combined_data_UNE$Nondurable.
goods))
combined_data_UNE$Gross.private.domestic.investment_growth = growth_rate(as.numeric(combined_dat
a_UNE$Gross.private.domestic.investment))
```

```
#install.packages("lmtest")
```

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.3.3
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.3.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
#IPO_Gross
bptest(IPO_Gross_GDP_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  IPO_Gross_GDP_lm
## BP = 12.757, df = 1, p-value = 0.0003547
```

```
bptest(IPO_Gross_Gross.national.product_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  IPO_Gross_Gross.national.product_lm
## BP = 12.794, df = 1, p-value = 0.0003477
```

```
bptest(IPO_Gross_Nondurable.goods_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  IPO_Gross_Nondurable.goods_lm
## BP = 8.4014, df = 1, p-value = 0.003749
```

```
bptest(IPO_Gross_Gross.private.domestic.investment_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  IPO_Gross_Gross.private.domestic.investment_lm
## BP = 14.909, df = 1, p-value = 0.0001128
```

```
bptest(IPO_Gross_Personal.consumption.expenditures_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  IPO_Gross_Personal.consumption.expenditures_lm
## BP = 16.801, df = 1, p-value = 4.152e-05
```

```
#DPRatio
bptest(DPRatio_GDP_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  DPRatio_GDP_lm
## BP = 28.953, df = 1, p-value = 7.416e-08
```

```
bptest(DPRatio_Gross.national.product_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  DPRatio_Gross.national.product_lm
## BP = 29.111, df = 1, p-value = 6.834e-08
```

```
bptest(DPRatio_Nondurable.goods_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  DPRatio_Nondurable.goods_lm
## BP = 0.67612, df = 1, p-value = 0.4109
```

```
bptest(DPRatio_Gross.private.domestic.investment_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  DPRatio_Gross.private.domestic.investment_lm
## BP = 28.92, df = 1, p-value = 7.544e-08
```

```
bptest(DPRatio_Personal.consumption.expenditures_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  DPRatio_Personal.consumption.expenditures_lm
## BP = 11.469, df = 1, p-value = 0.0007078
```

```
#EPBound
bptest(EPBound_GDP_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  EPBound_GDP_lm
## BP = 2.0776, df = 1, p-value = 0.1495
```

```
bptest(EPBound_Gross.national.product_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  EPBound_Gross.national.product_lm
## BP = 1.7711, df = 1, p-value = 0.1832
```

```
bptest(EPBound_Nondurable.goods_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  EPBound_Nondurable.goods_lm
## BP = 0.0053685, df = 1, p-value = 0.9416
```

```
bptest(EPBound_Gross.private.domestic.investment_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  EPBound_Gross.private.domestic.investment_lm
## BP = 17.838, df = 1, p-value = 2.405e-05
```

```
bptest(EPBound_Personal.consumption.expenditures_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  EPBound_Personal.consumption.expenditures_lm
## BP = 0.015505, df = 1, p-value = 0.9009
```

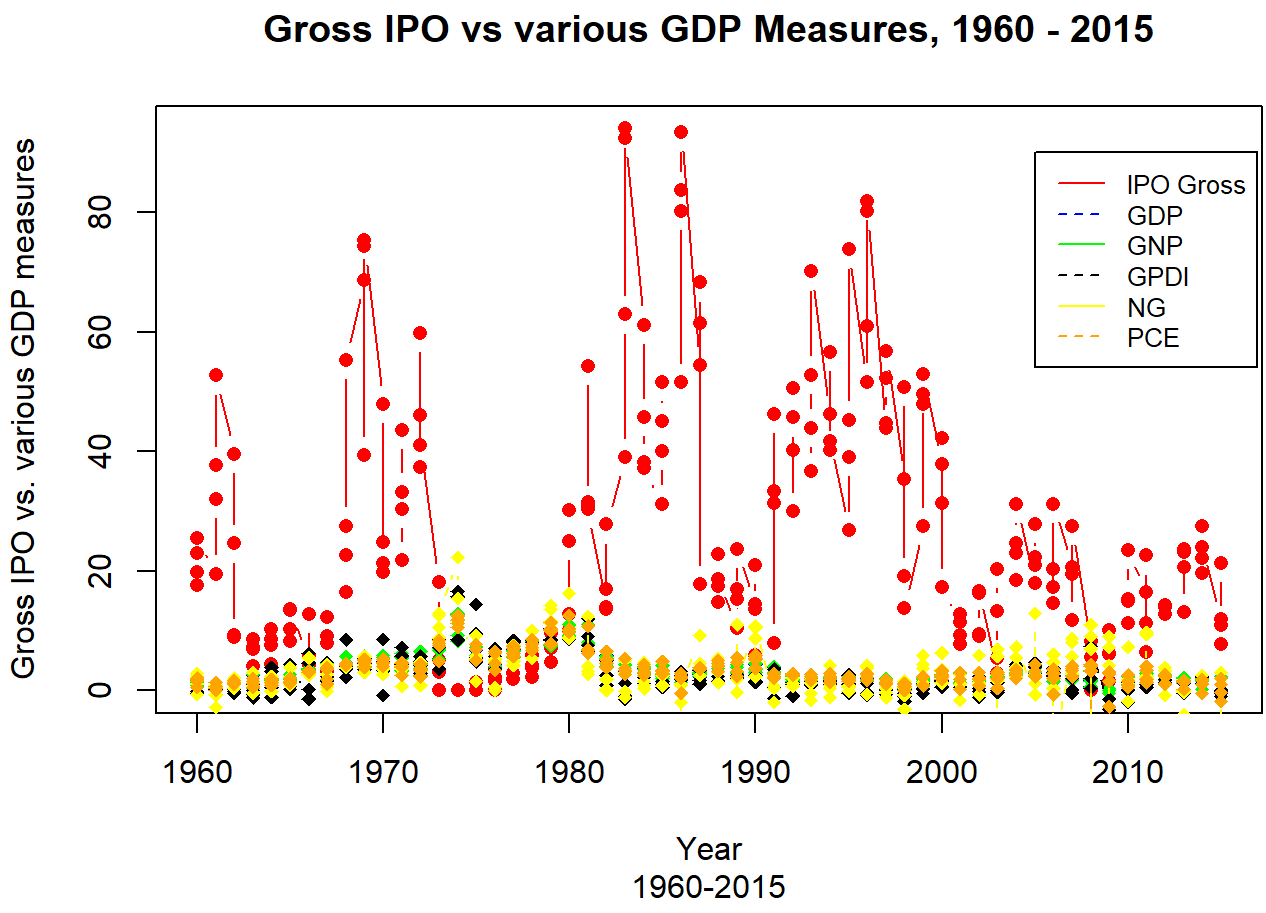
```
#Gives same results as initial Breusch-Pagan Test code above
#so we can stop here
```

```
#Time series graphs
#Look at graphs of these growth rates versus the various equity premium/risk aversion measures
#install.packages("TSstudio")
```

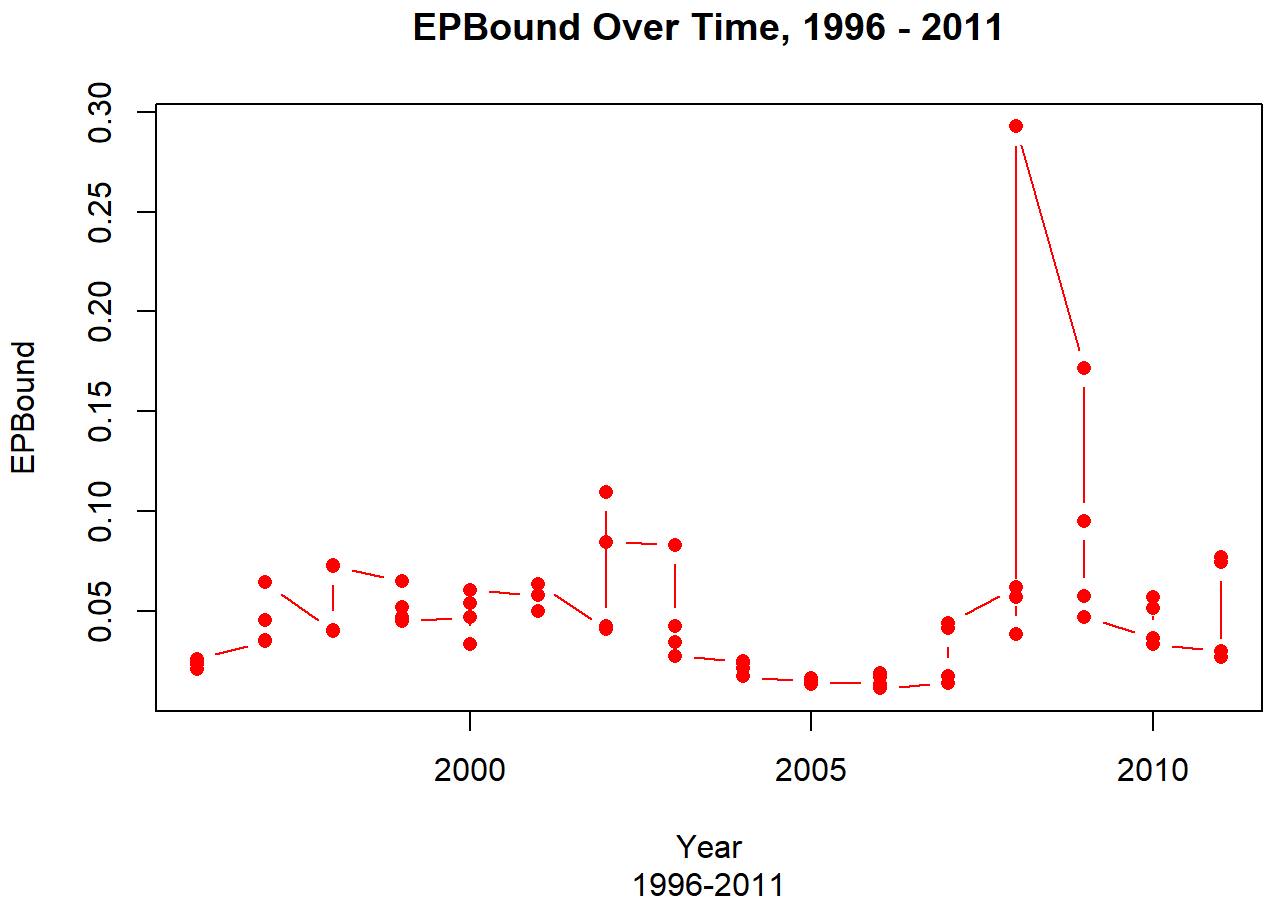
```
#library("TSstudio")
```

[3 Equity Premium measure graphs against various GDP measures, ALL YEARS]

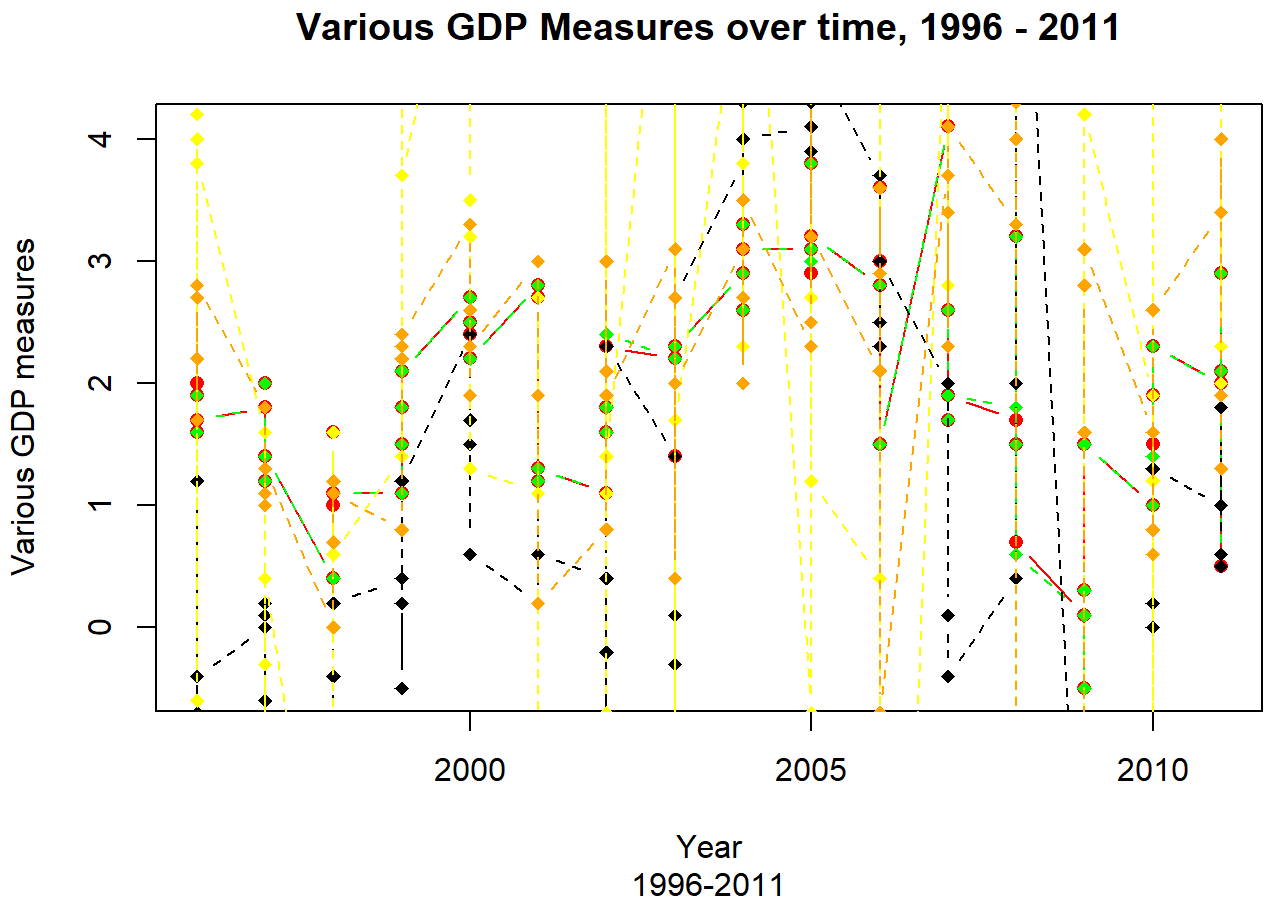
```
x = combined_data_IPO_Gross$Year
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$GDP
y3 = combined_data_IPO_Gross$Gross.national.product
y4 = combined_data_IPO_Gross$Gross.private.domestic.investment
y5 = combined_data_IPO_Gross$Nondurable.goods
y6 = combined_data_IPO_Gross$Personal.consumption.expenditures

plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs various GDP Measures, 1960 - 2015",
sub="1960-2015",
  xlab="Year", ylab="Gross IPO vs. various GDP measures")
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
lines(x, y3, pch=18, col="green", type="b", lty=2)
lines(x, y4, pch=18, col="black", type="b", lty=2)
lines(x, y5, pch=18, col="yellow", type="b", lty=2)
lines(x, y6, pch=18, col="orange", type="b", lty=2)
legend(2005, 90, legend=c("IPO Gross", "GDP", "GNP", "GPDI", "NG", "PCE" ),
       col=c("red", "blue", "green", "black", "yellow","orange"), lty=1:2, cex=0.8)
```



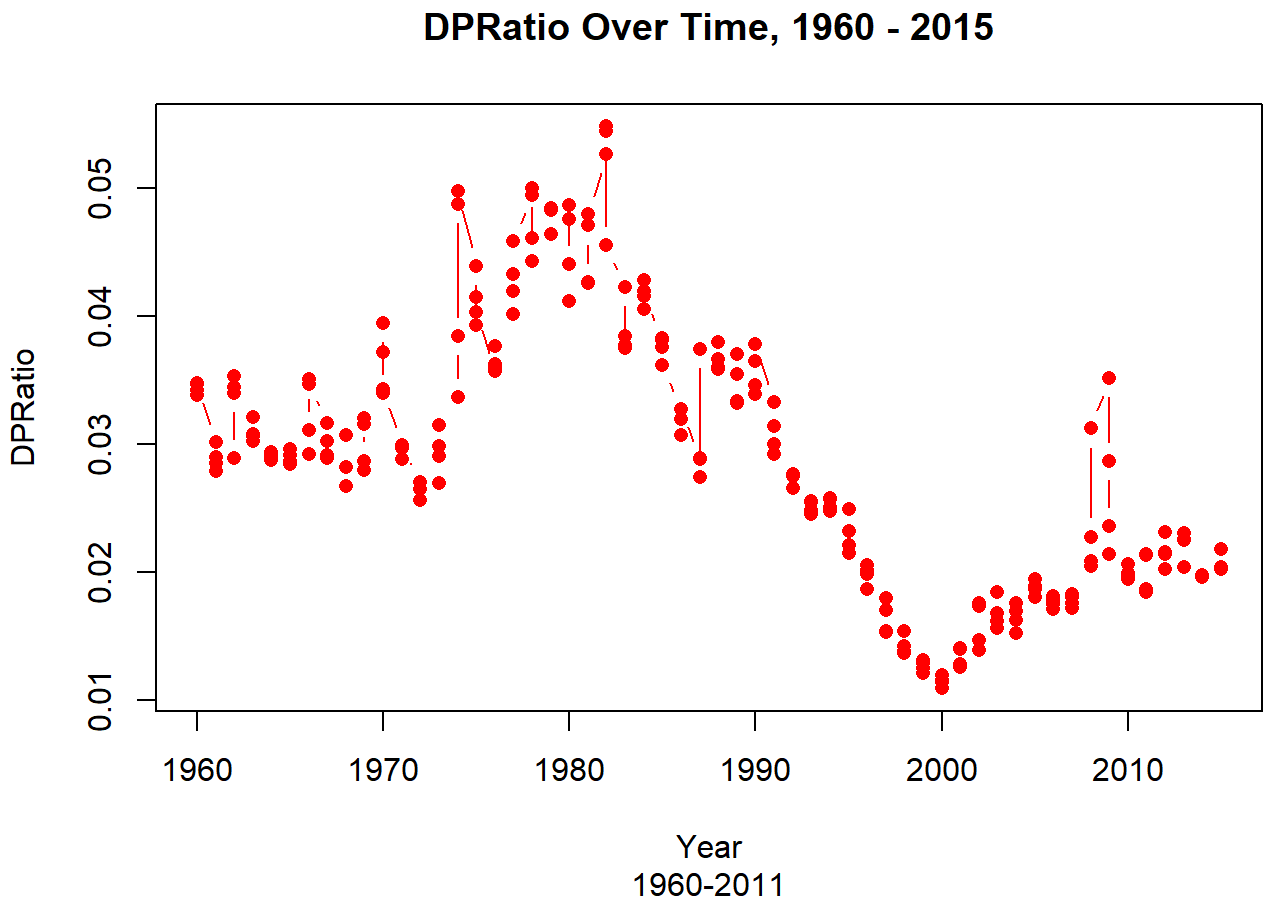
```
#EP Bound
#Y-axis values are too different so split into 2
x = combined_data_EPBound$Year
y1 = combined_data_EPBound$EPBound

plot(x, y1, type="b", pch=16, col="red", main="EPBound Over Time, 1996 - 2011", sub="1996-2011",
  xlab="Year", ylab="EPBound")
```



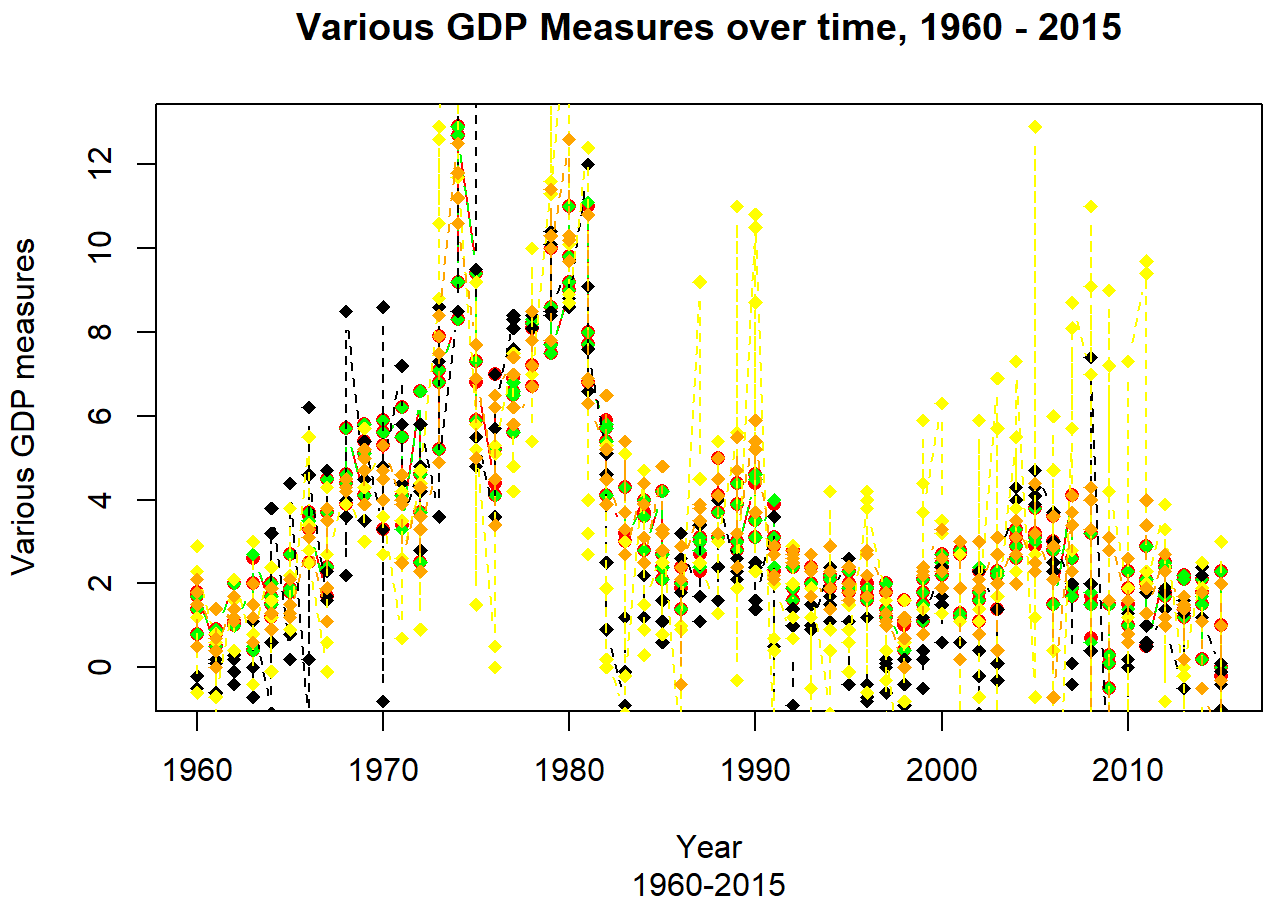
```
x = combined_data_EPBound$Year
y2 = combined_data_EPBound$GDP
y3 = combined_data_EPBound$Gross.national.product
y4 = combined_data_EPBound$Gross.private.domestic.investment
y5 = combined_data_EPBound$Nondurable.goods
y6 = combined_data_EPBound$Personal.consumption.expenditures

plot(x, y2, type="b", pch=16, col="red", main="Various GDP Measures over time, 1996 - 2011", sub
="1996-2011",
  xlab="Year", ylab="Various GDP measures", xlim=c(1996, 2011))
# Add lines for gdp measures
lines(x, y3, pch=18, col="green", type="b", lty=2)
lines(x, y4, pch=18, col="black", type="b", lty=2)
lines(x, y5, pch=18, col="yellow", type="b", lty=2)
lines(x, y6, pch=18, col="orange", type="b", lty=2)
legend(2005, 90, legend=c("GDP", "GNP", "GPDI", "NG", "PCE" ),
       col=c("red", "blue", "green", "black", "yellow"), lty=1:2, cex=0.8)
```



```
#DP ratio
#Y-axis values are too different so split into 2
x = combined_data_DPRatio$Year
y1 = combined_data_DPRatio$DPRatio

plot(x, y1, type="b", pch=16, col="red", main="DPRatio Over Time, 1960 - 2015", sub="1960-2011",
  xlab="Year", ylab="DPRatio")
```

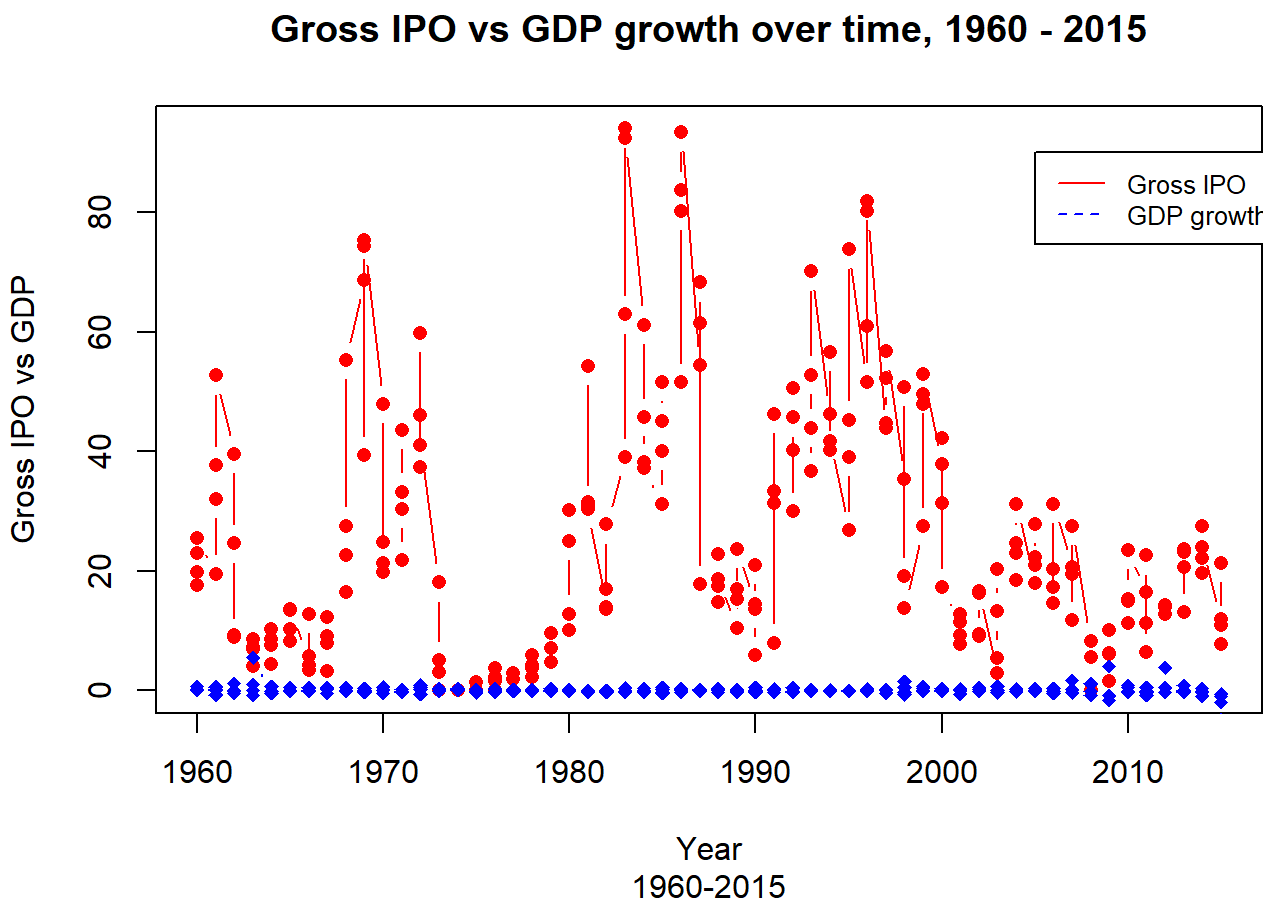


```
x = combined_data_DPRatio$Year
y2 = combined_data_DPRatio$GDP
y3 = combined_data_DPRatio$Gross.national.product
y4 = combined_data_DPRatio$Gross.private.domestic.investment
y5 = combined_data_DPRatio$Nondurable.goods
y6 = combined_data_DPRatio$Personal.consumption.expenditures

plot(x, y2, type="b", pch=16, col="red", main="Various GDP Measures over time, 1960 - 2015", sub
="1960-2015",
  xlab="Year", ylab="Various GDP measures", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y3, pch=18, col="green", type="b", lty=2)
lines(x, y4, pch=18, col="black", type="b", lty=2)
lines(x, y5, pch=18, col="yellow", type="b", lty=2)
lines(x, y6, pch=18, col="orange", type="b", lty=2)
legend(2005, 90, legend=c("GDP", "GNP", "GPDI", "NG", "PCE" ),
       col=c("red", "blue", "green", "black", "yellow"), lty=1:2, cex=0.8)
```

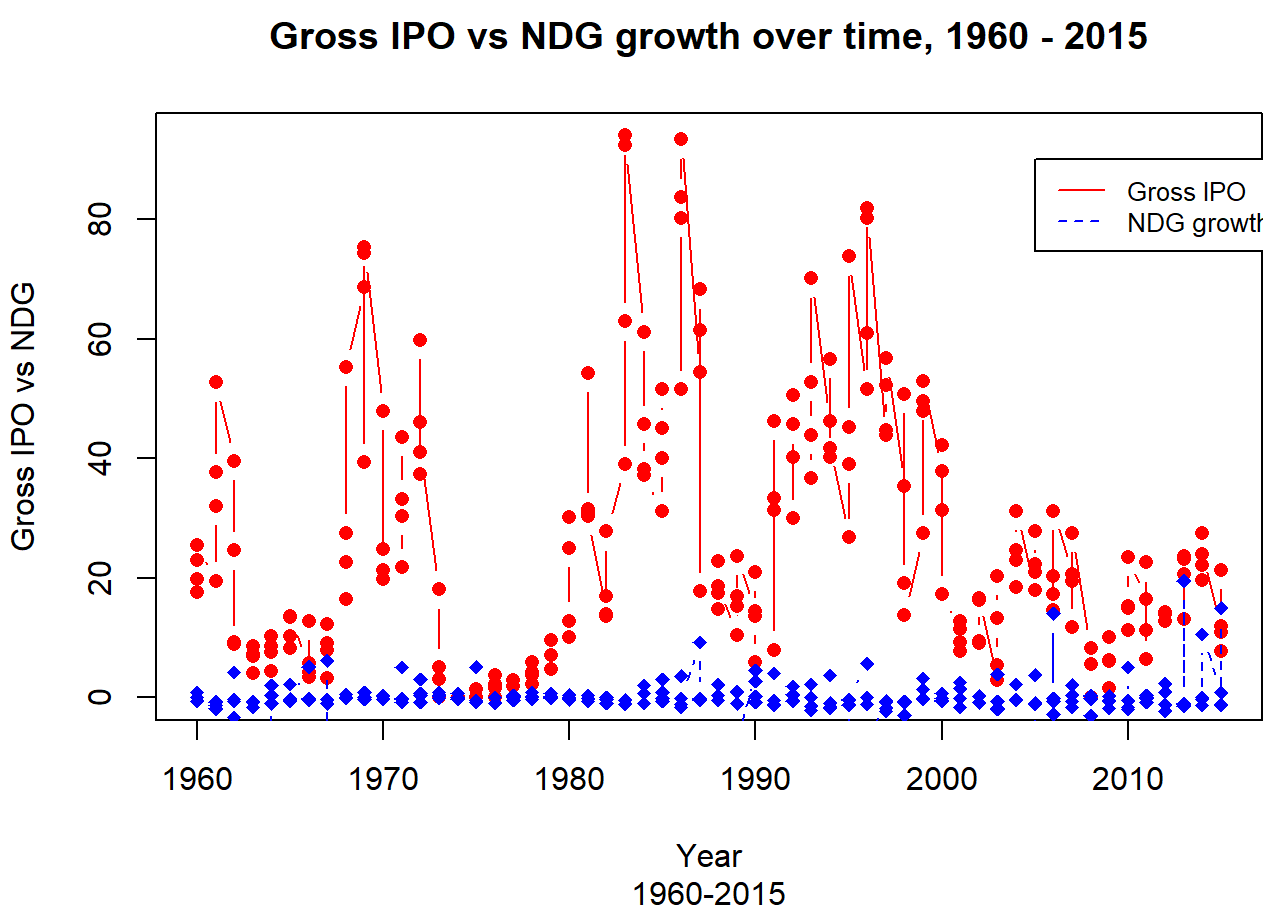


[Gross IPO over time vs various GDP Measure Growth Rates]

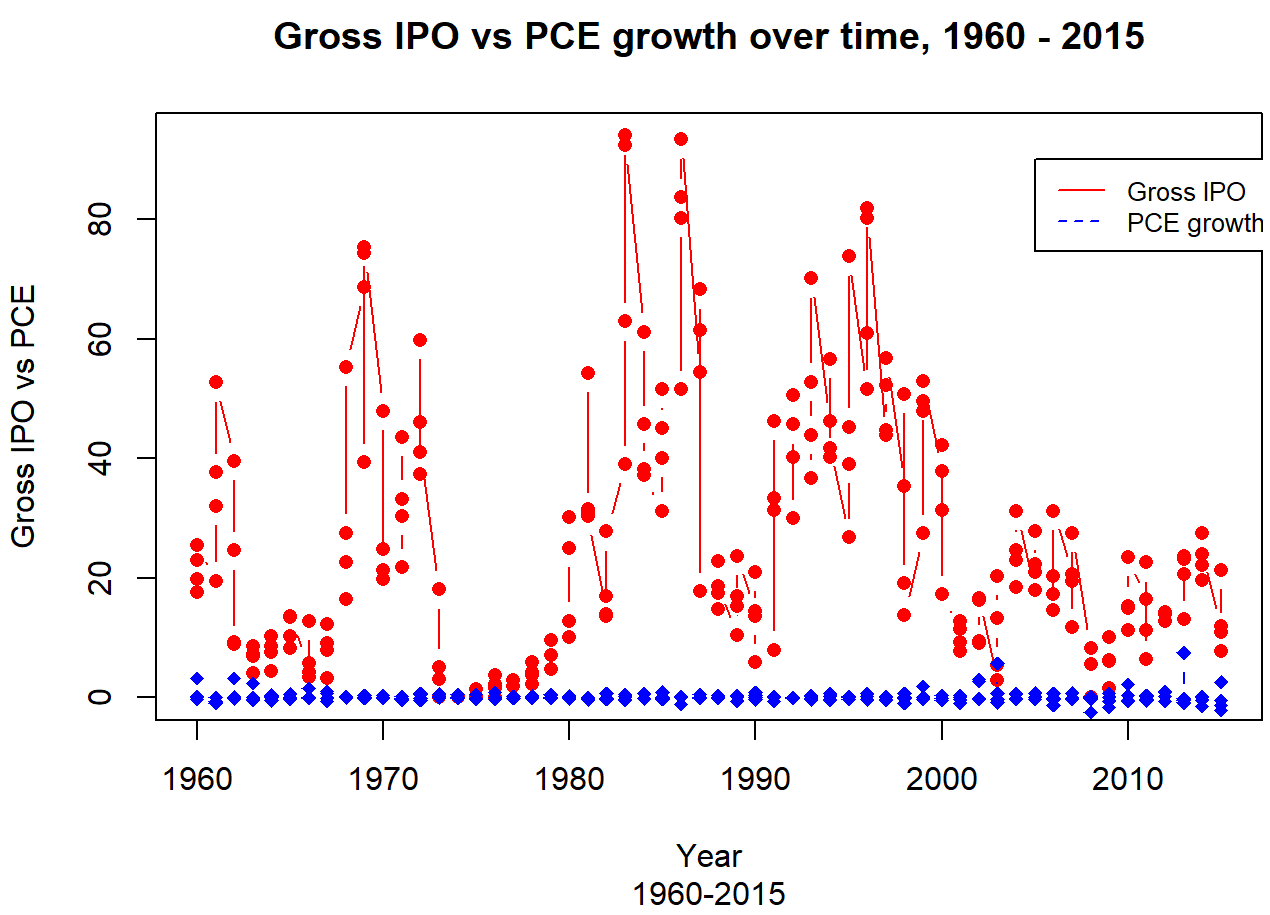
```
#Gross IPO vs GDP growth over time
x = combined_data_IPO_Gross$Year
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$GDP_growth
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs GDP growth over time, 1960 - 2015",
sub="1960-2015",
  xlab="Year", ylab="Gross IPO vs GDP", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("Gross IPO", "GDP growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



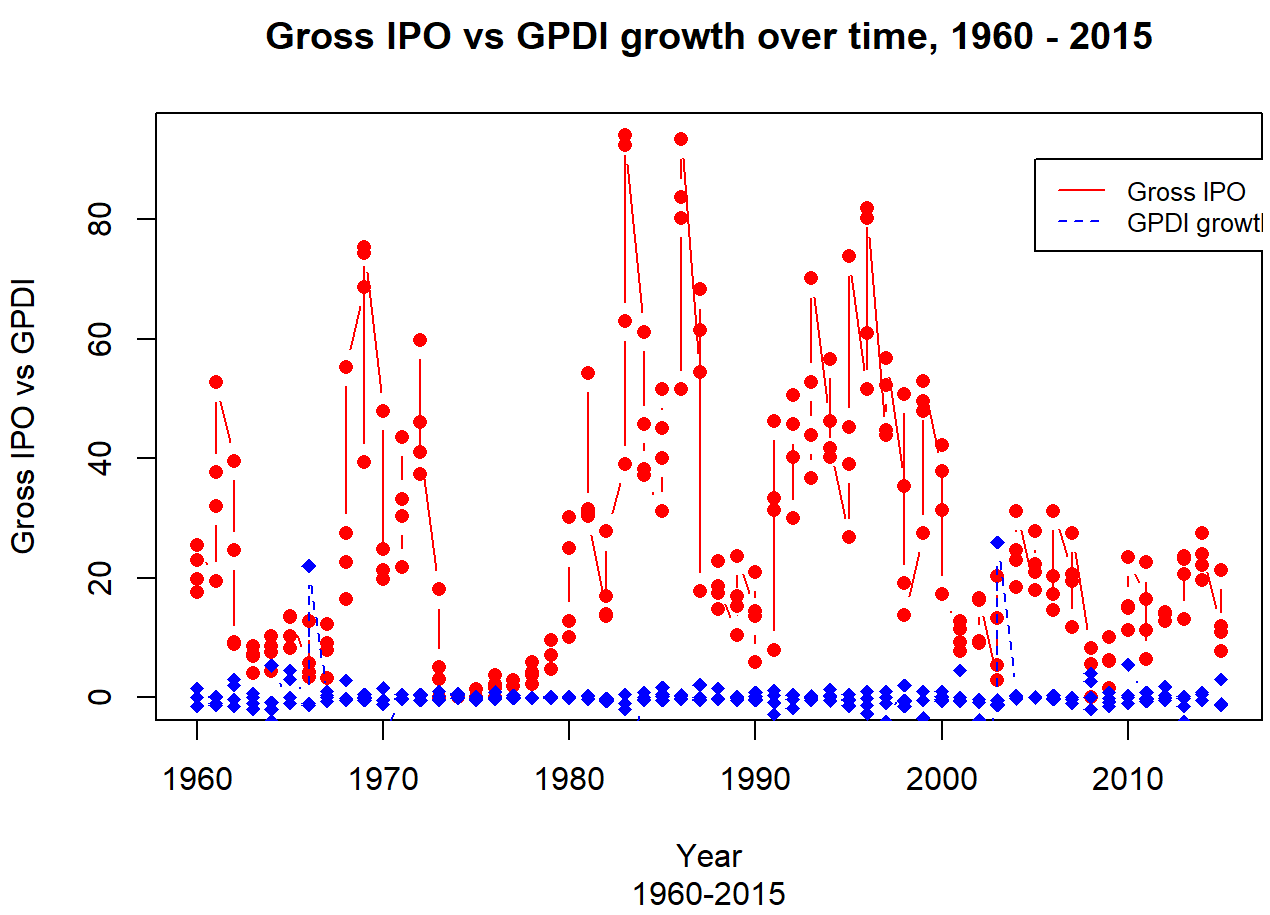

```
#Gross IPO vs NDG growth over time
x = combined_data_IPO_Gross$Year
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Nondurable.goods_growth
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs NDG growth over time, 1960 - 2015",
sub="1960-2015",
  xlab="Year", ylab="Gross IPO vs NDG", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("Gross IPO", "NDG growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



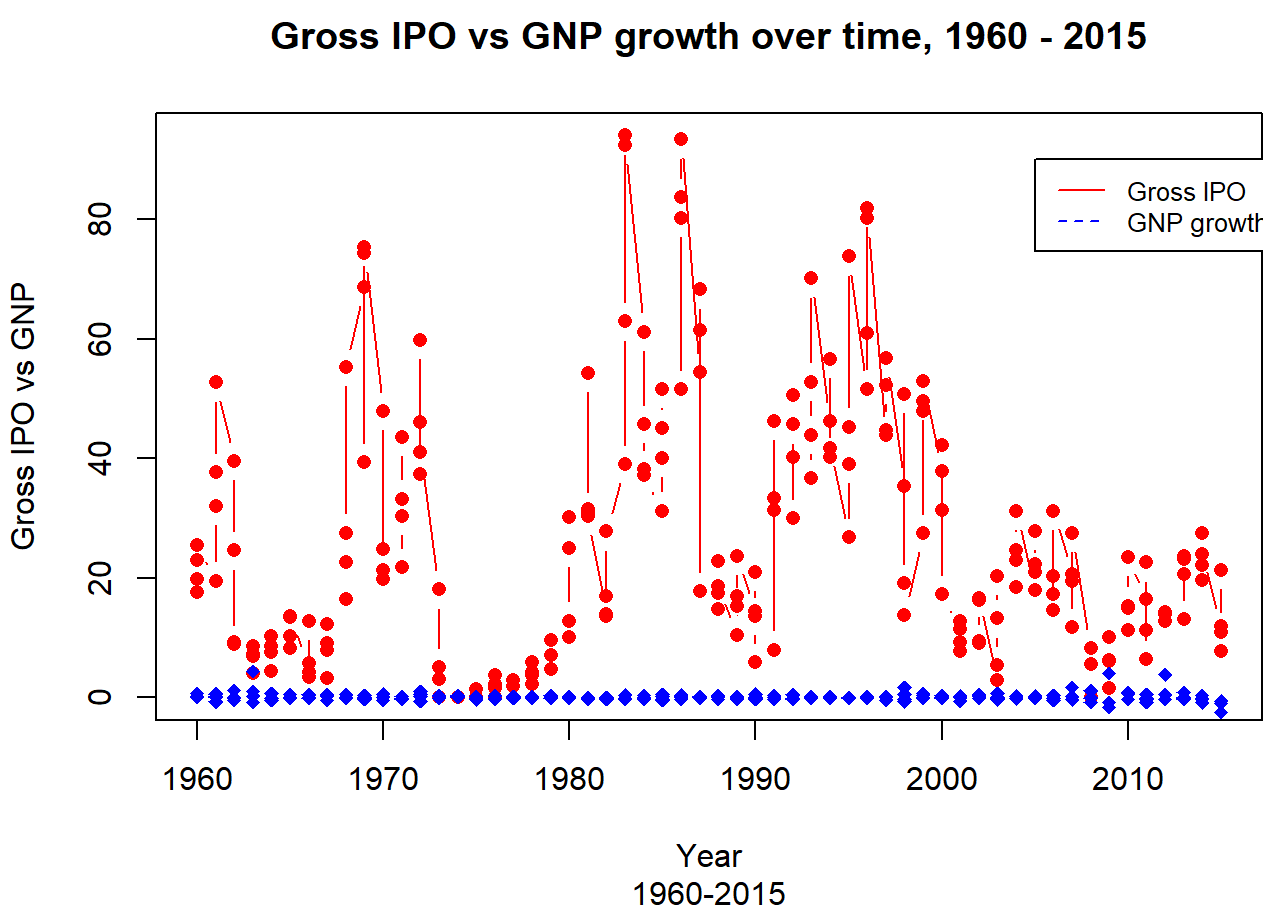

```
#Gross IPO vs PCE growth over time
x = combined_data_IPO_Gross$Year
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Personal.consumption.expenditures_growth
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs PCE growth over time, 1960 - 2015",
sub="1960-2015",
  xlab="Year", ylab="Gross IPO vs PCE", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("Gross IPO", "PCE growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```




```
#Gross IPO vs GPDI growth over time
x = combined_data_IPO_Gross$Year
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Gross.private.domestic.investment_growth
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs GPDI growth over time, 1960 - 2015",
sub="1960-2015",
  xlab="Year", ylab="Gross IPO vs GPDI", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("Gross IPO", "GPDI growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```

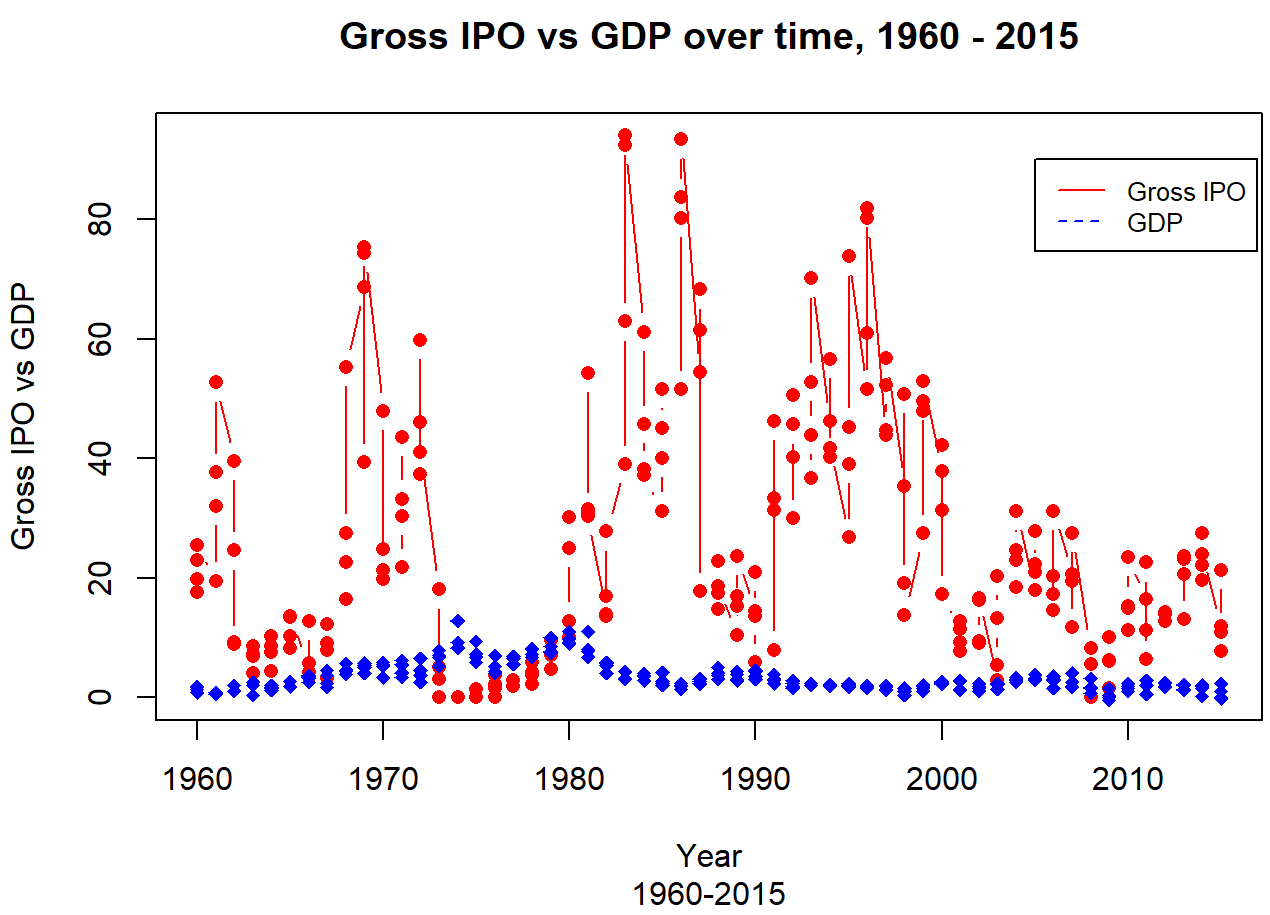



```
#Gross IPO vs GNP growth over time
x = combined_data_IPO_Gross$Year
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Gross.national.product_growth
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs GNP growth over time, 1960 - 2015",
sub="1960-2015",
  xlab="Year", ylab="Gross IPO vs GNP", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("Gross IPO", "GNP growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```

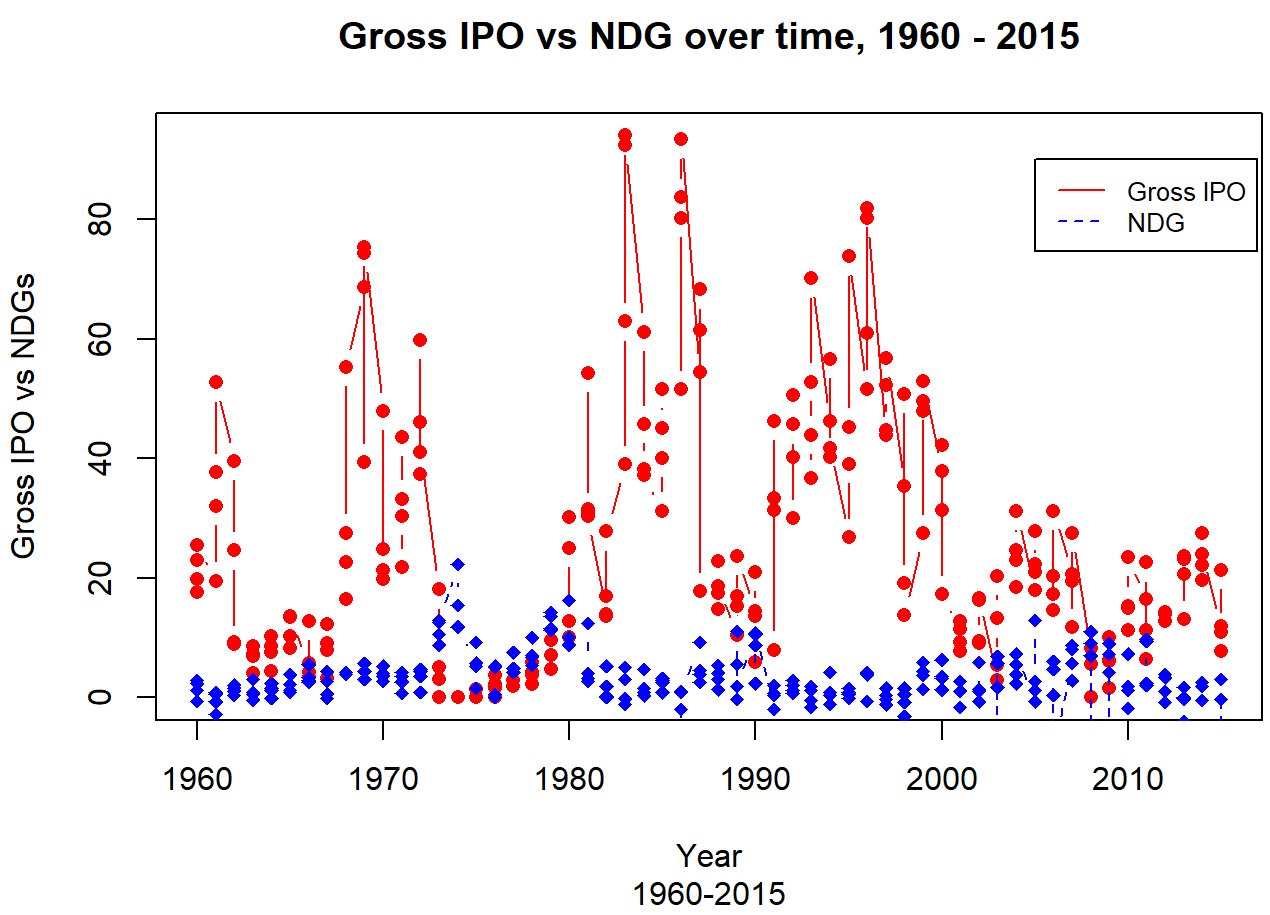


[Gross IPO over time vs various GDP Measures]

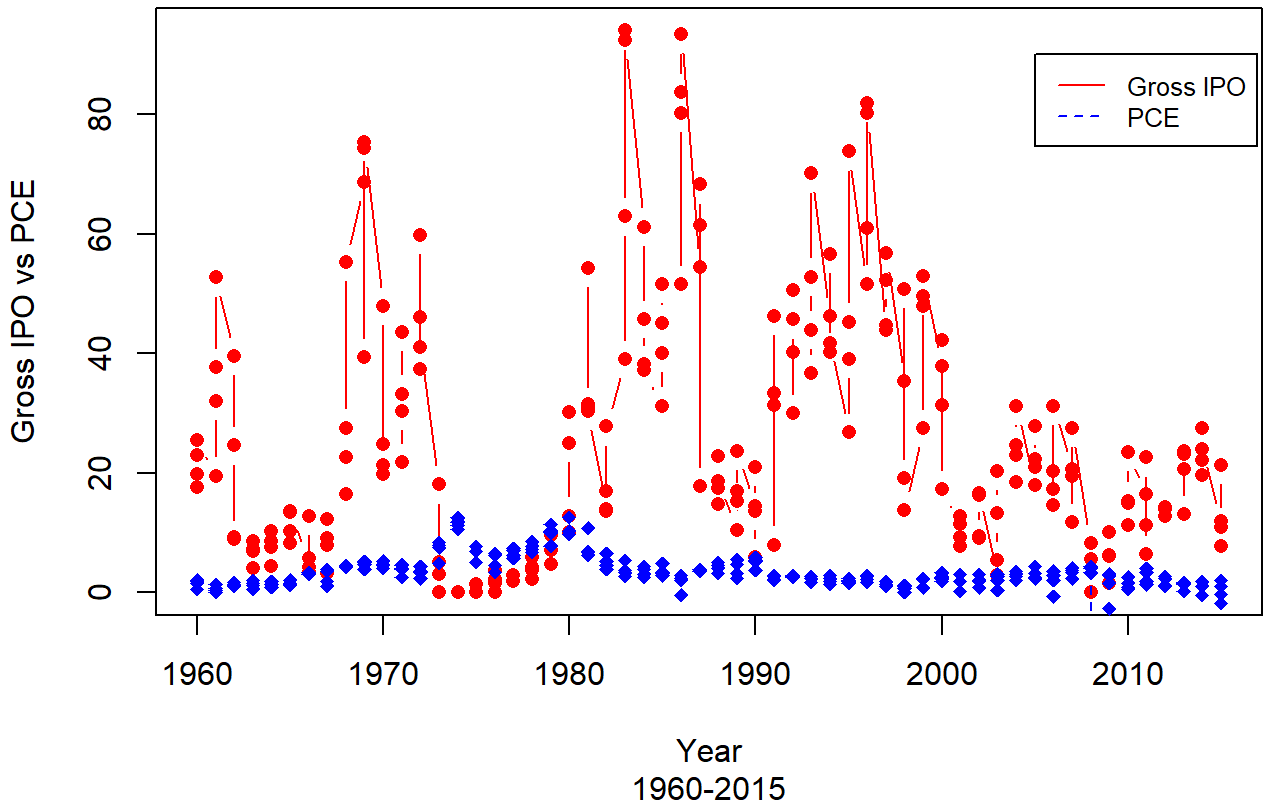
```
#Gross IPO vs GDP over time
x = combined_data_IPO_Gross$Year
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$GDP
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs GDP over time, 1960 - 2015", sub="19
60-2015",
  xlab="Year", ylab="Gross IPO vs GDP", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("Gross IPO", "GDP"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



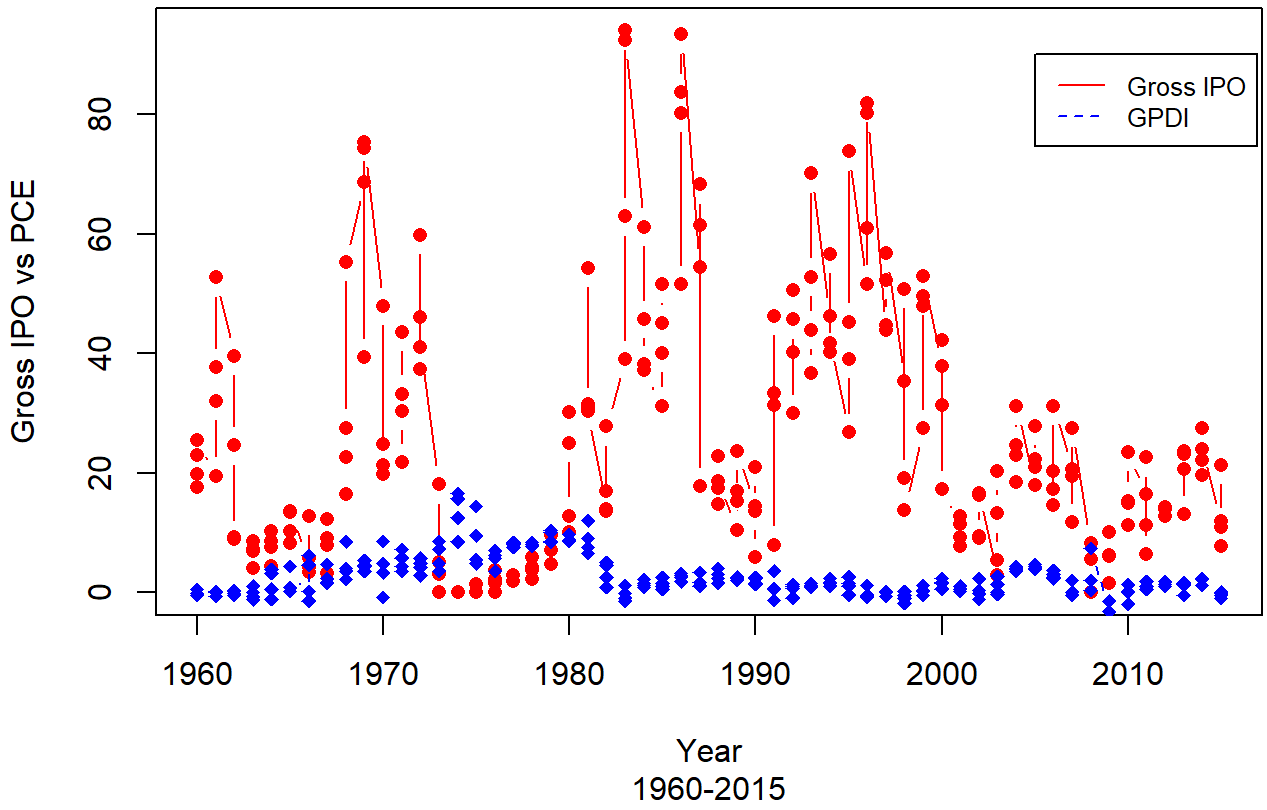

```
#Gross IPO vs NDG over time
x = combined_data_IPO_Gross$Year
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Nondurable.goods
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs NDG over time, 1960 - 2015", sub="19
60-2015",
  xlab="Year", ylab="Gross IPO vs NDGs", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("Gross IPO", "NDG"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



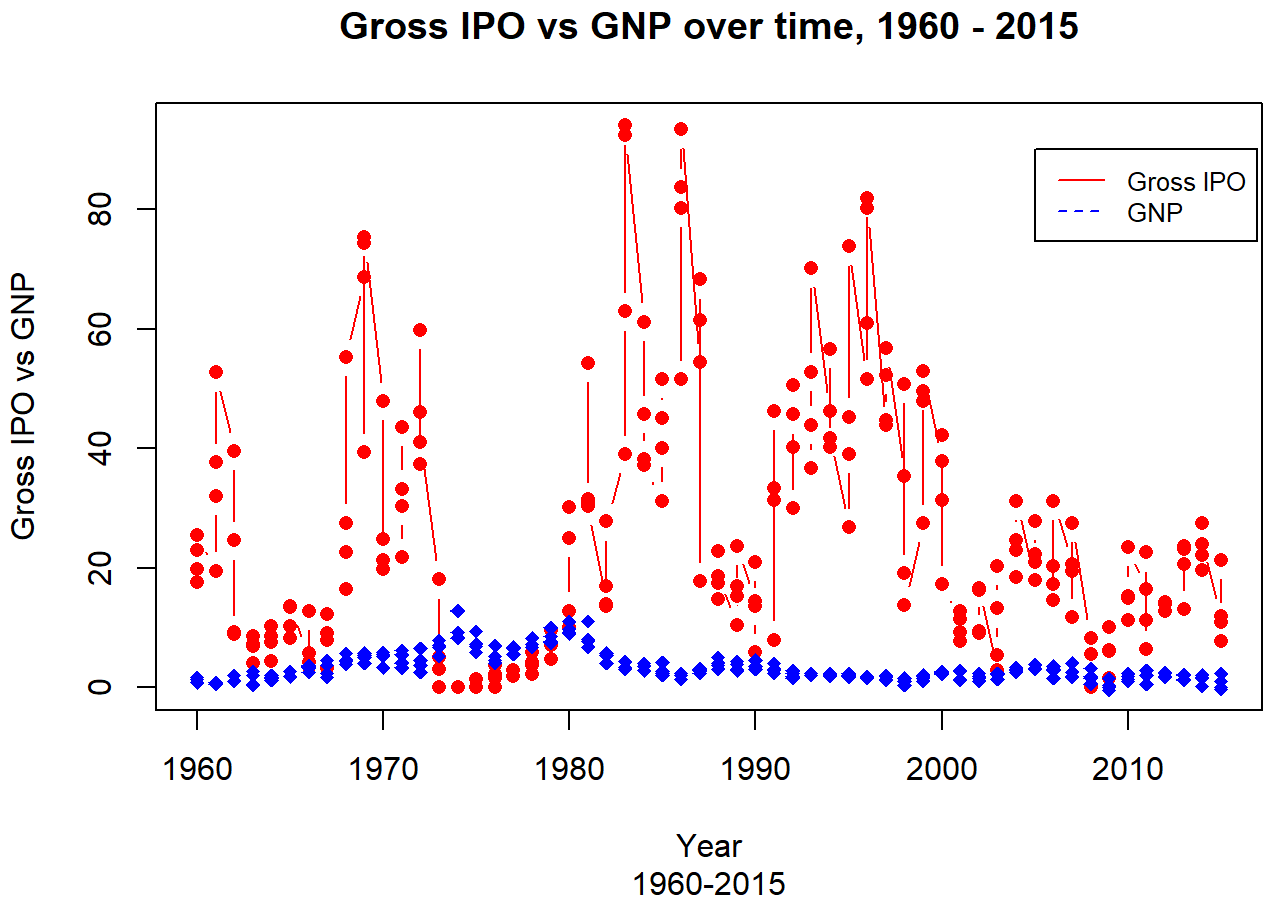

```
#Gross IPO vs PCE over time
x = combined_data_IPO_Gross$Year
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Personal.consumption.expenditures
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs PCE over time, 1960 - 2015", sub="19
60-2015",
  xlab="Year", ylab="Gross IPO vs PCE", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("Gross IPO", "PCE"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```




```
#Gross IPO vs GPDI over time
x = combined_data_IPO_Gross$Year
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Gross.private.domestic.investment
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs PCE over time, 1960 - 2015", sub="19
60-2015",
  xlab="Year", ylab="Gross IPO vs PCE", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("Gross IPO", "GPDI"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```

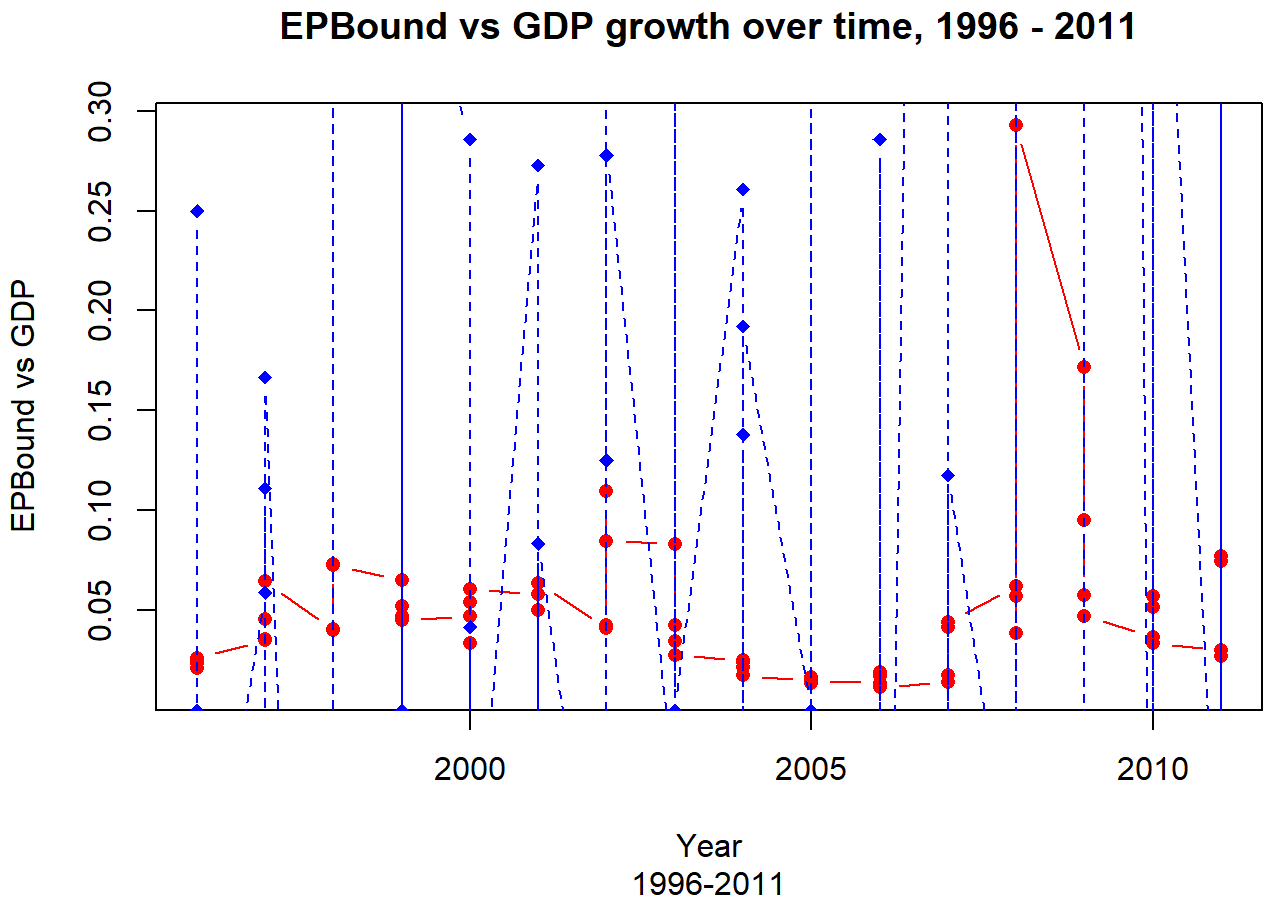



```
#Gross IPO vs GNP over time
x = combined_data_IPO_Gross$Year
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Gross.national.product
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs GNP over time, 1960 - 2015", sub="19
60-2015",
  xlab="Year", ylab="Gross IPO vs GNP", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("Gross IPO", "GNP"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```

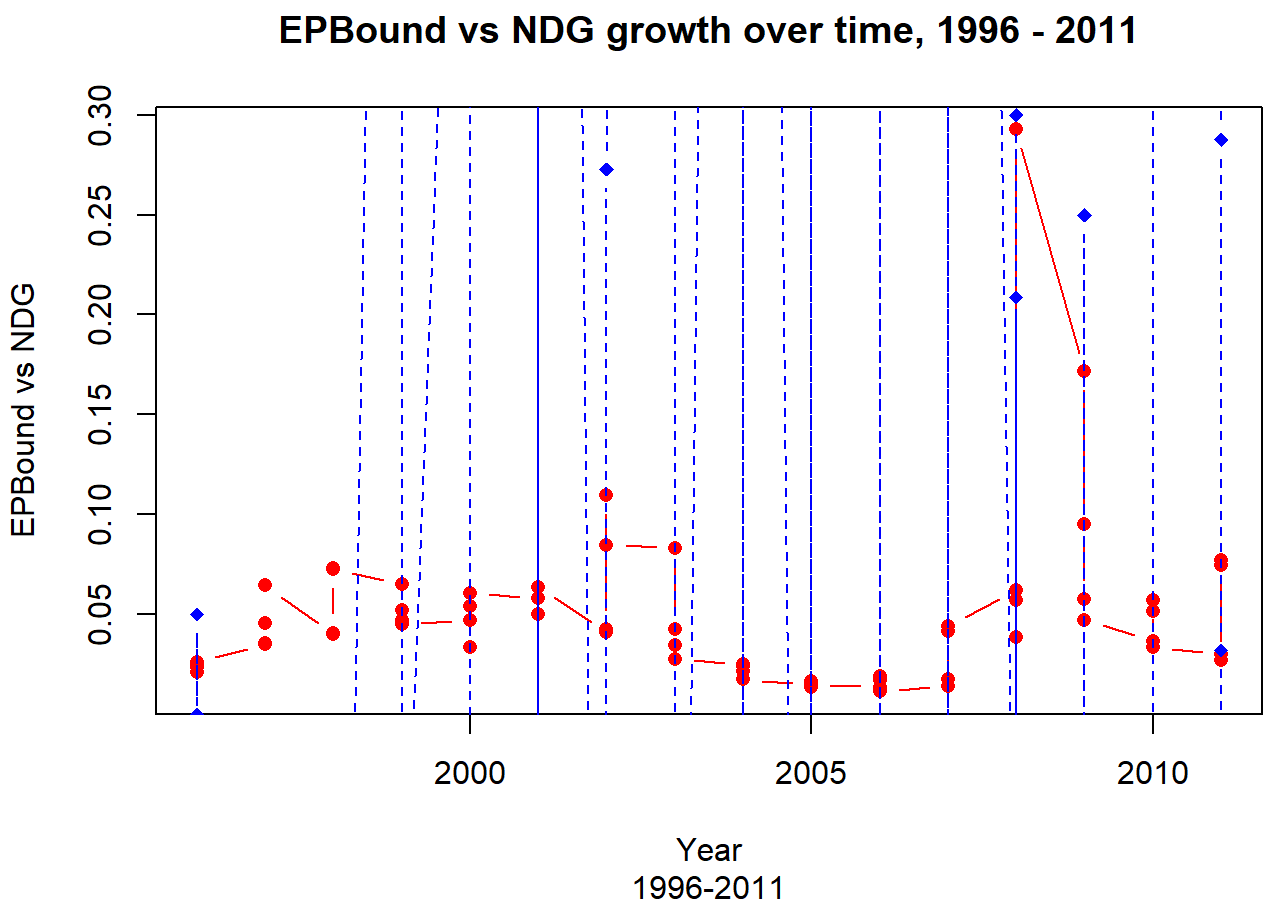


[EPBound vs various gdp measures]

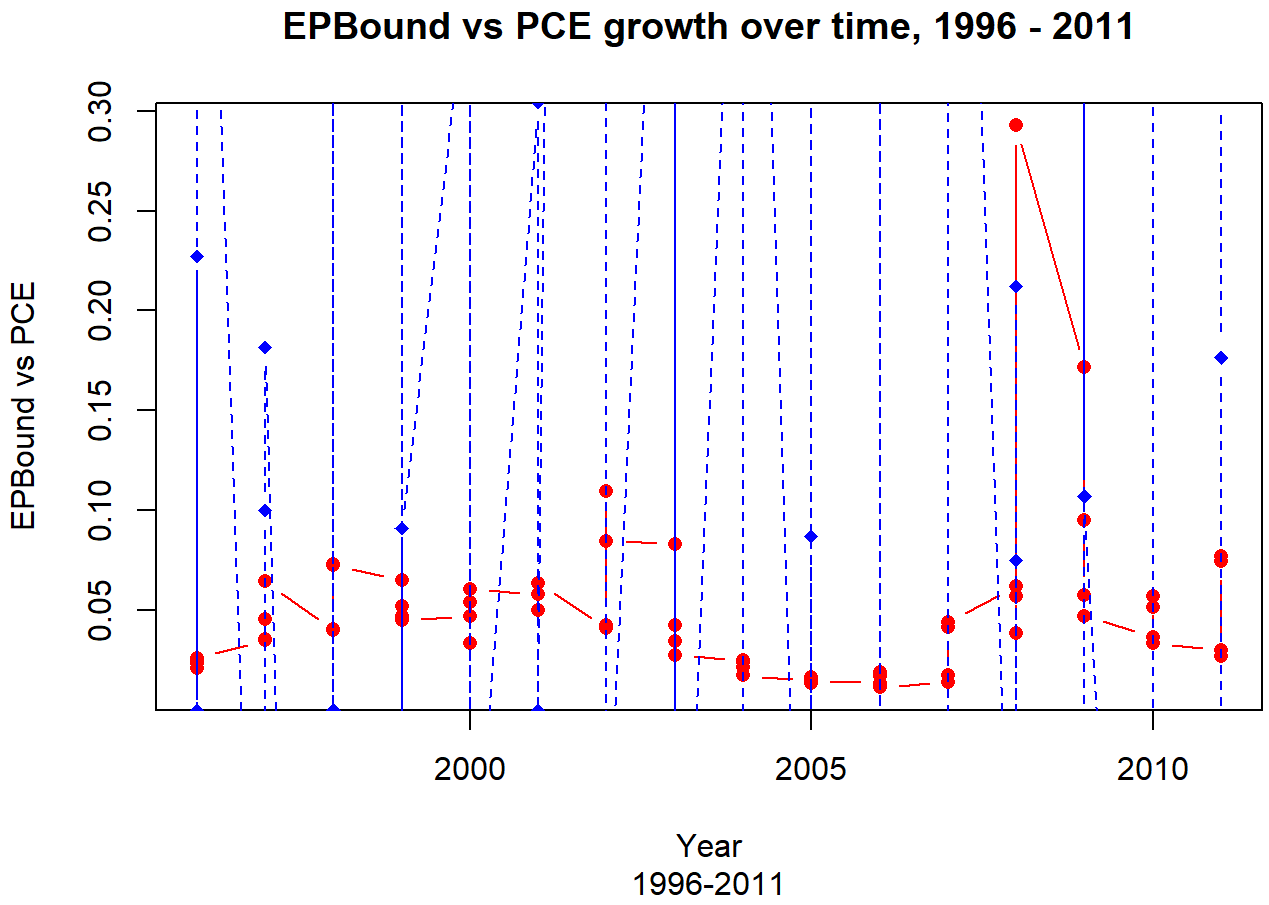
```
#EPBound vs GDP growth over time
x = combined_data_EPBound$Year
y1 = combined_data_EPBound$EPBound
y2 = combined_data_EPBound$GDP_growth
plot(x, y1, type="b", pch=16, col="red", main="EPBound vs GDP growth over time, 1996 - 2011", su
b="1996-2011",
  xlab="Year", ylab="EPBound vs GDP", xlim=c(1996, 2011))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("EPBound", "GDP growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



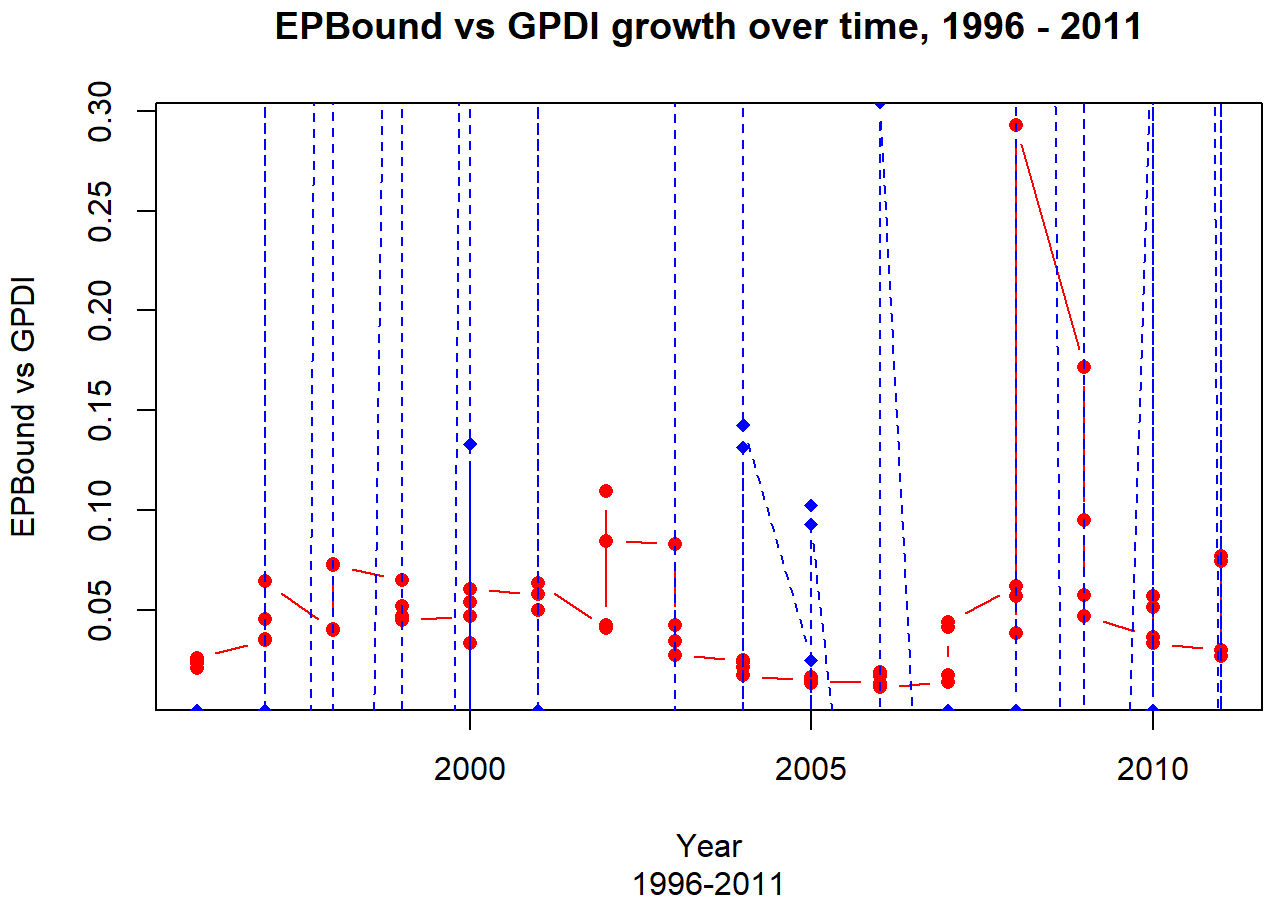

```
#EPBound vs NDG growth over time
x = combined_data_EPBound$Year
y1 = combined_data_EPBound$EPBound
y2 = combined_data_EPBound$Nondurable.goods_growth
plot(x, y1, type="b", pch=16, col="red", main="EPBound vs NDG growth over time, 1996 - 2011", su
b="1996-2011",
  xlab="Year", ylab="EPBound vs NDG", xlim=c(1996, 2011))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("EPBound", "NDG growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



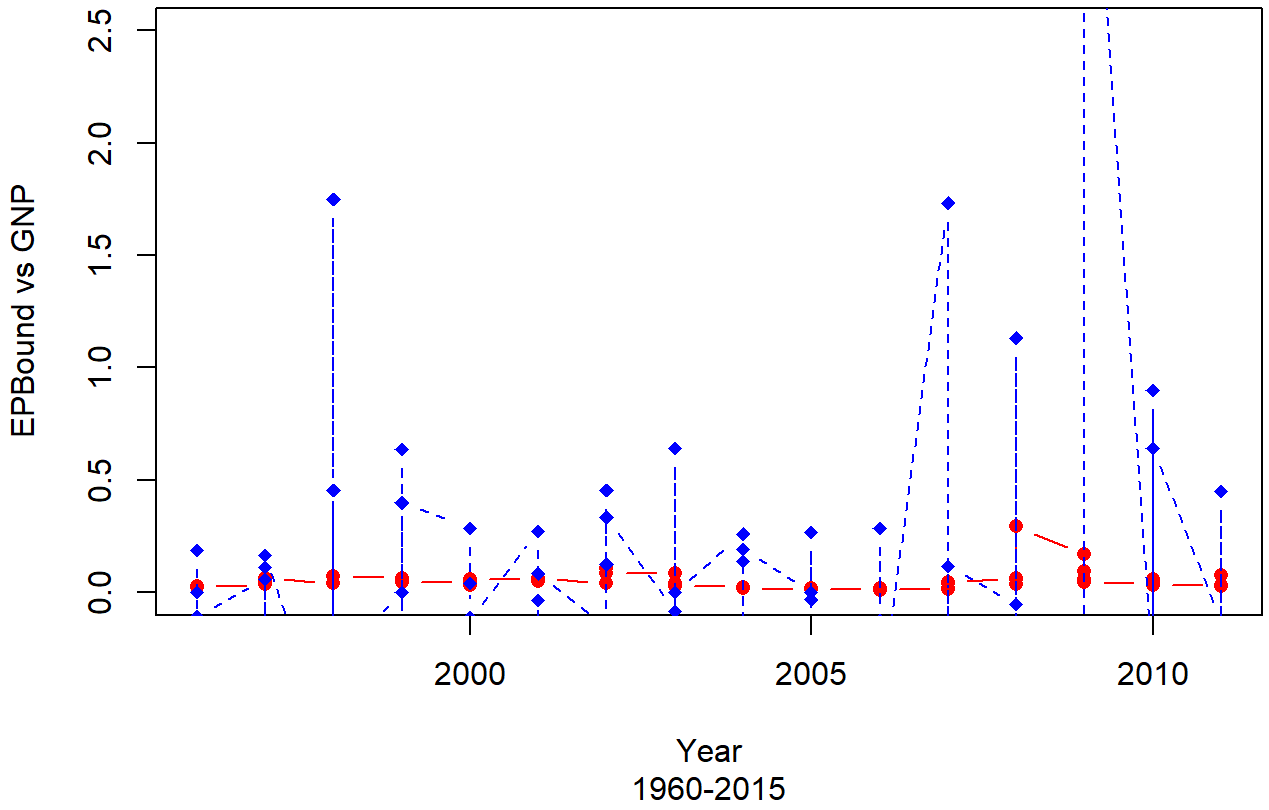

```
#EPBound vs PCE growth over time
x = combined_data_EPBound$Year
y1 = combined_data_EPBound$EPBound
y2 = combined_data_EPBound$Personal.consumption.expenditures_growth
plot(x, y1, type="b", pch=16, col="red", main="EPBound vs PCE growth over time, 1996 - 2011", su
b="1996-2011",
  xlab="Year", ylab="EPBound vs PCE", xlim=c(1996, 2011))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("EPBound", "PCE growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



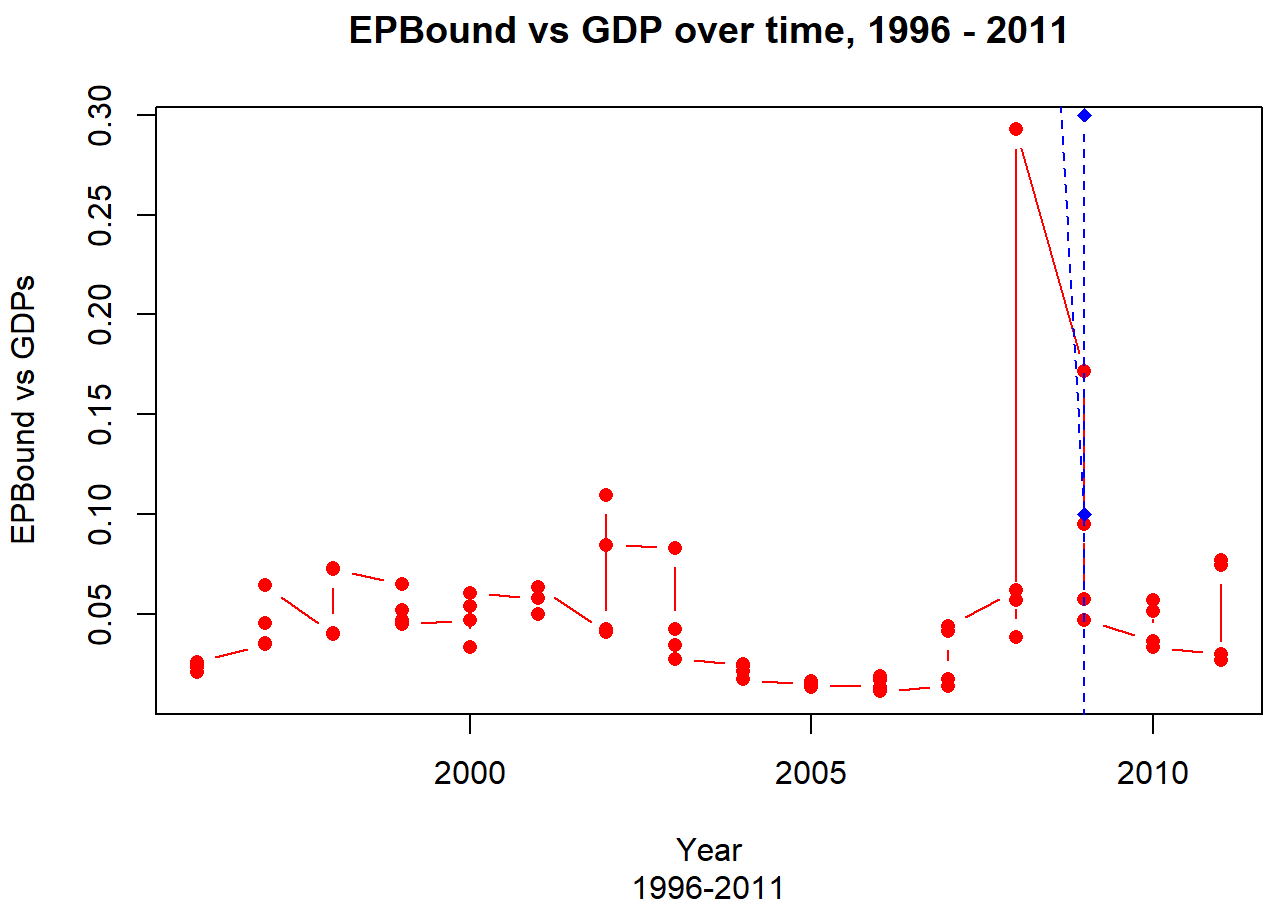

```
#EPBound vs GPDI growth over time
x = combined_data_EPBound$Year
y1 = combined_data_EPBound$EPBound
y2 = combined_data_EPBound$Gross.private.domestic.investment_growth
plot(x, y1, type="b", pch=16, col="red", main="EPBound vs GPDI growth over time, 1996 - 2011", s
ub="1996-2011",
  xlab="Year", ylab="EPBound vs GPDI", xlim=c(1996, 2011))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("EPBound", "GPDI growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



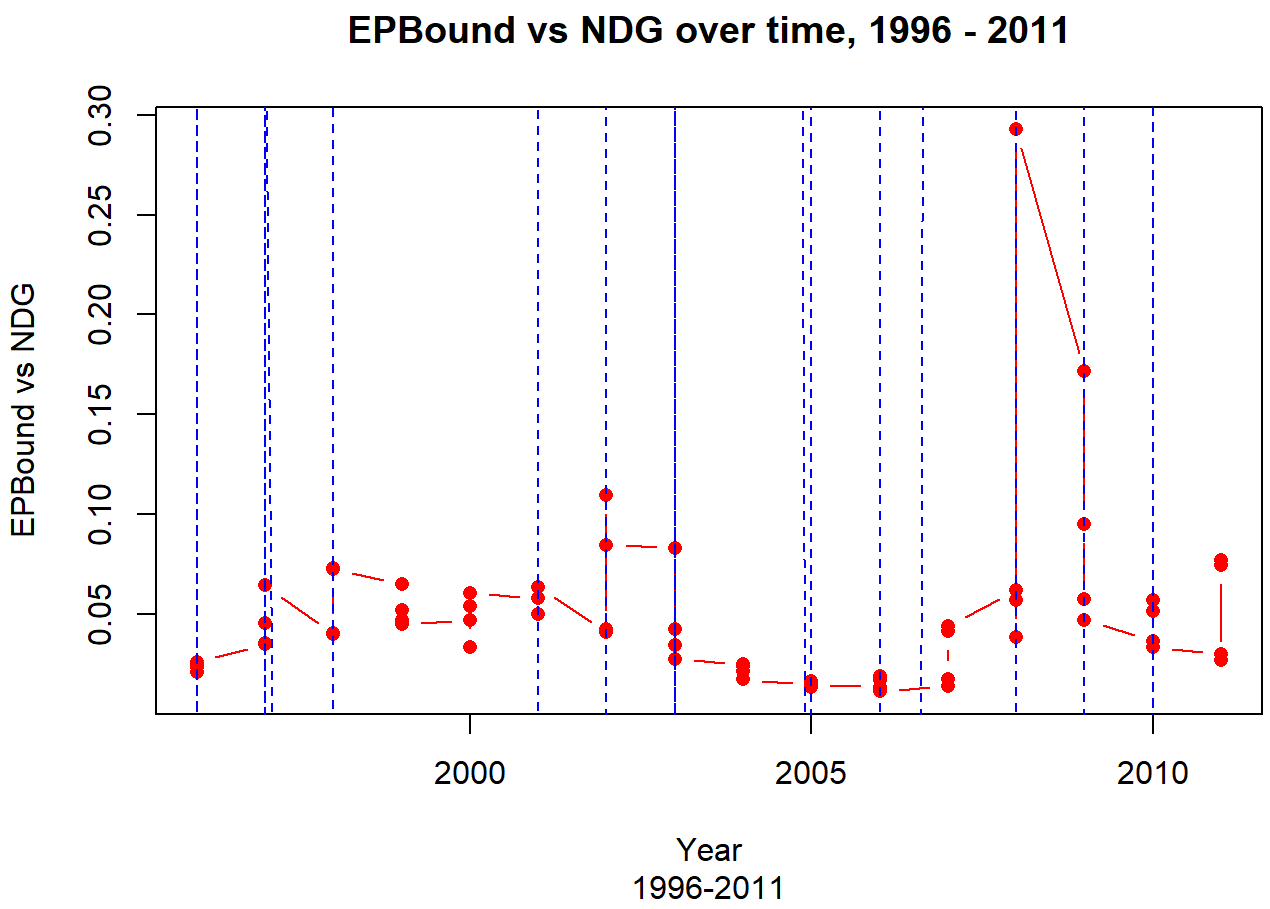

```
#EPBound vs GNP growth over time
x = combined_data_EPBound$Year
y1 = combined_data_EPBound$EPBound
y2 = combined_data_EPBound$Gross.national.product_growth
plot(x, y1, type="b", pch=16, col="red", main="EPBound vs GNP  growth over time, 1996 - 2011", s
ub="1960-2015",
  xlab="Year", ylab="EPBound vs GNP", xlim=c(1996, 2011), ylim=c(0, 2.5))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("EPBound", "GNP growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



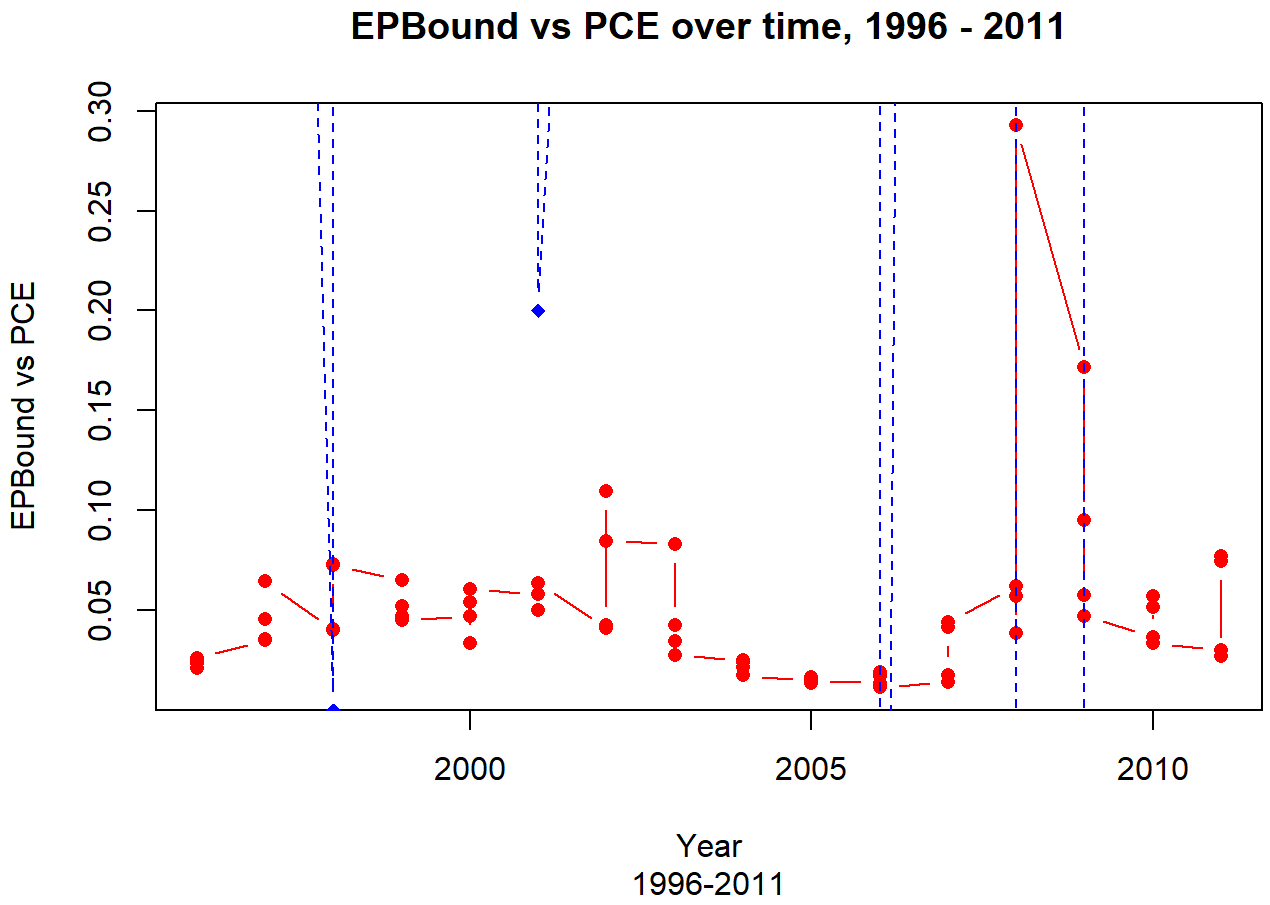

```
#EPBound vs GDP over time
x = combined_data_EPBound$Year
y1 = combined_data_EPBound$EPBound
y2 = combined_data_EPBound$GDP
plot(x, y1, type="b", pch=16, col="red", main="EPBound vs GDP over time, 1996 - 2011", sub="1996
-2011",
  xlab="Year", ylab="EPBound vs GDPs", xlim=c(1996, 2011))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("EPBound", "GDP"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



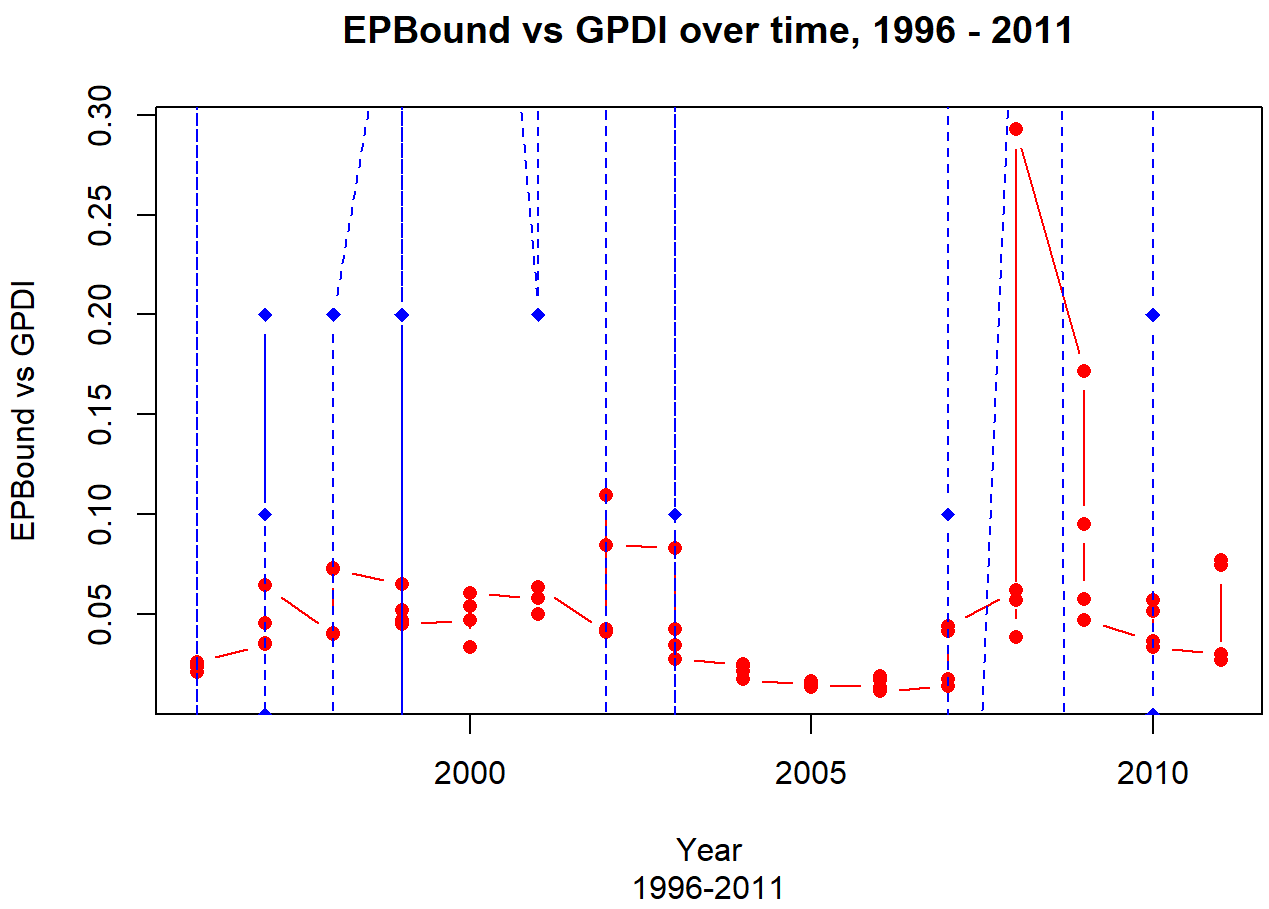

```
#EPBound vs NDG over time
x = combined_data_EPBound$Year
y1 = combined_data_EPBound$EPBound
y2 = combined_data_EPBound$Nondurable.goods
plot(x, y1, type="b", pch=16, col="red", main="EPBound vs NDG over time, 1996 - 2011", sub="1996
-2011",
  xlab="Year", ylab="EPBound vs NDG", xlim=c(1996, 2011))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("EPBound", "NDG"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



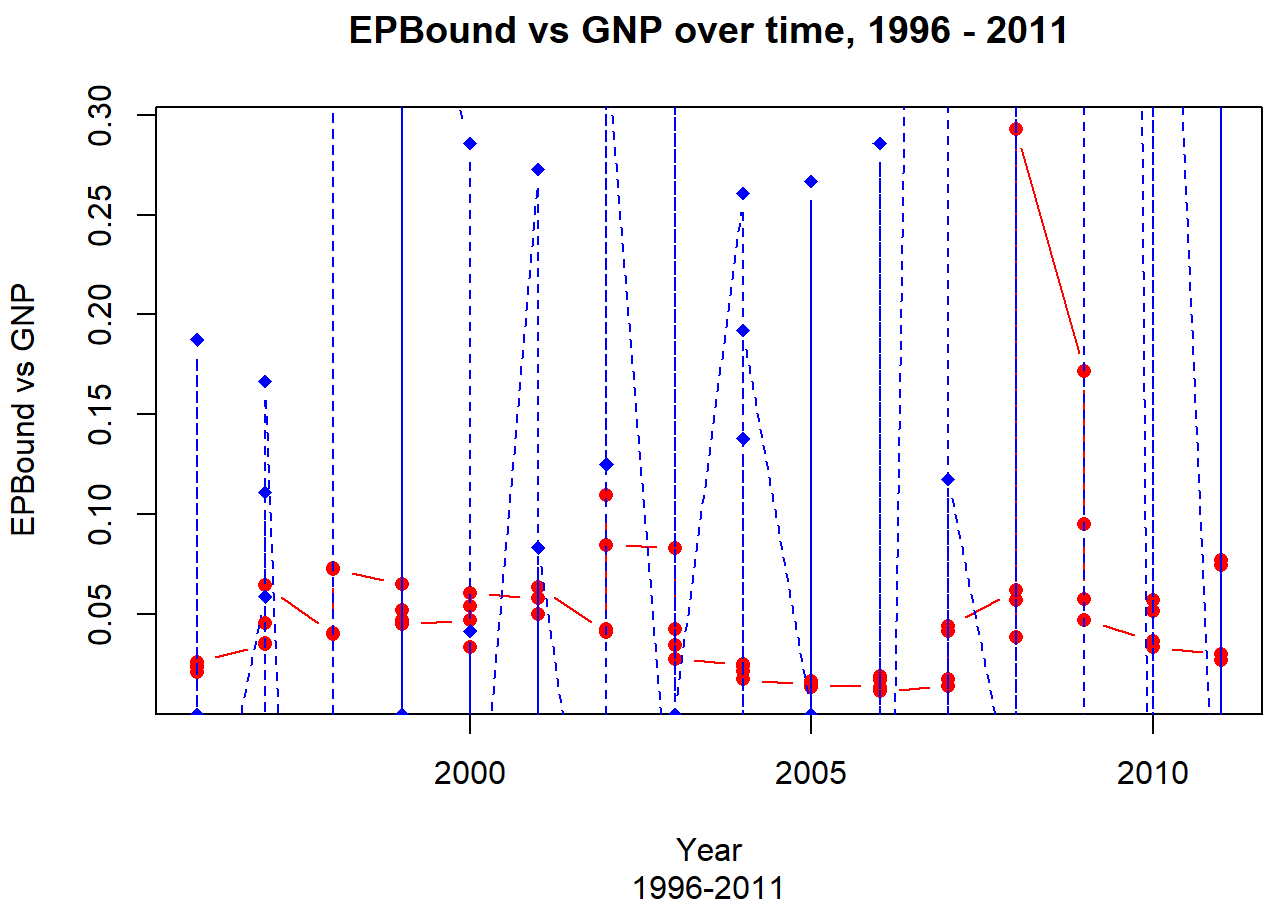

```
#EPBound vs PCE over time
x = combined_data_EPBound$Year
y1 = combined_data_EPBound$EPBound
y2 = combined_data_EPBound$Personal.consumption.expenditures
plot(x, y1, type="b", pch=16, col="red", main="EPBound vs PCE over time, 1996 - 2011", sub="1996
-2011",
  xlab="Year", ylab="EPBound vs PCE", xlim=c(1996, 2011))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("EPBound", "PCE"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```




```
#EPBound vs GPDI over time
x = combined_data_EPBound$Year
y1 = combined_data_EPBound$EPBound
y2 = combined_data_EPBound$Gross.private.domestic.investment
plot(x, y1, type="b", pch=16, col="red", main="EPBound vs GPDI over time, 1996 - 2011", sub="199
6-2011",
  xlab="Year", ylab="EPBound vs GPDI", xlim=c(1996, 2011))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("EPBound", "GPDI"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```

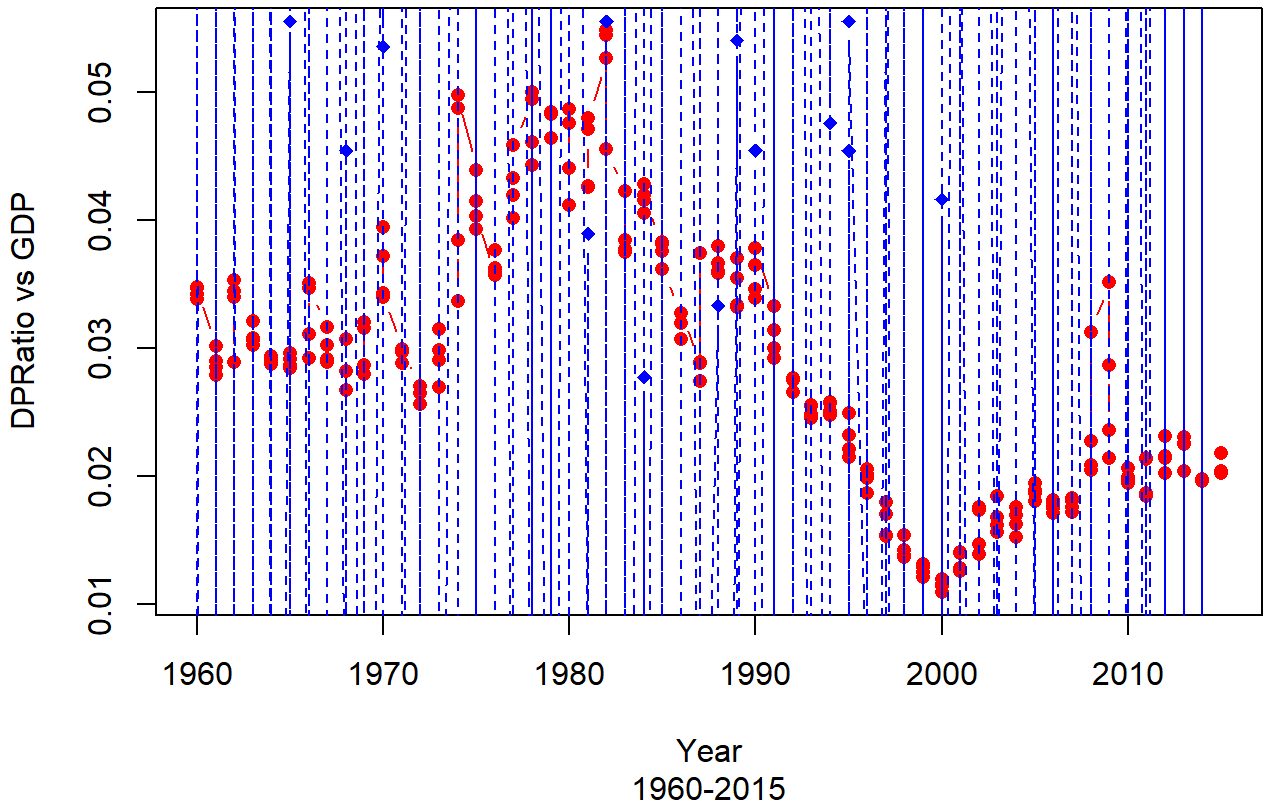



```
#EPBound vs GNP over time
x = combined_data_EPBound$Year
y1 = combined_data_EPBound$EPBound
y2 = combined_data_EPBound$Gross.national.product_growth
plot(x, y1, type="b", pch=16, col="red", main="EPBound vs GNP over time, 1996 - 2011", sub="1996
-2011",
  xlab="Year", ylab="EPBound vs GNP", xlim=c(1996, 2011))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("EPBound", "GNP"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```

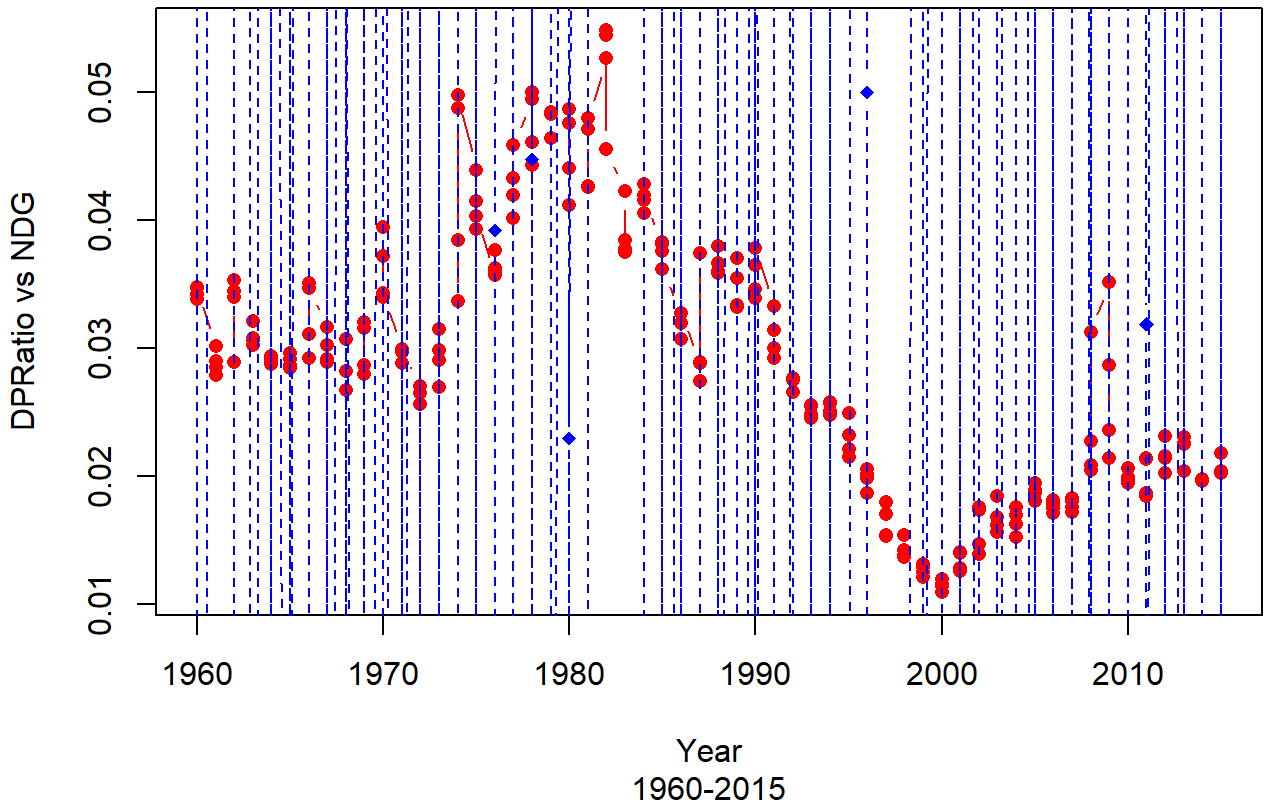


[DPRatio over time vs various GDP Measure Growth Rates]

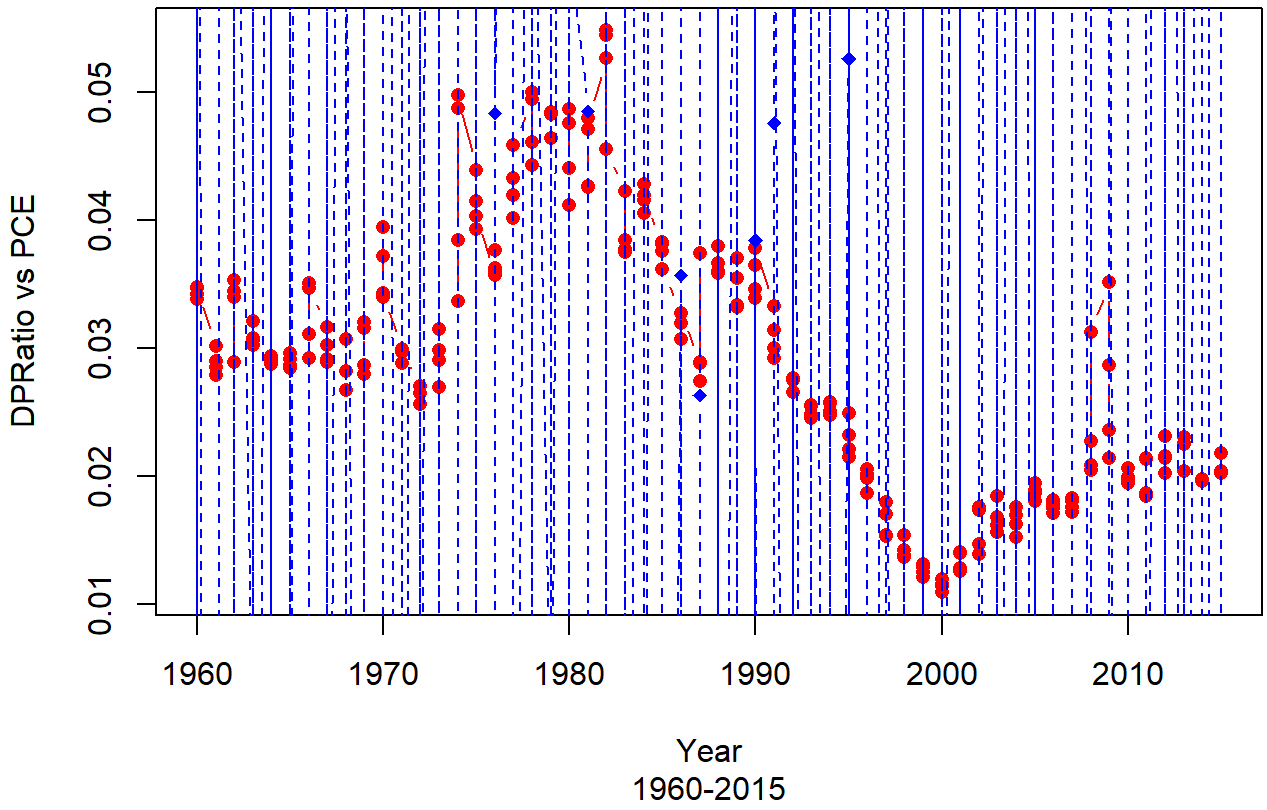
```
#DPRatio vs GDP growth over time
x = combined_data_DPRatio$Year
y1 = combined_data_DPRatio$DPRatio
y2 = combined_data_DPRatio$GDP_growth
plot(x, y1, type="b", pch=16, col="red", main="DPRatio vs GDP growth over time, 1960 - 2015", su
b="1960-2015",
  xlab="Year", ylab="DPRatio vs GDP", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("DPRatio", "GDP growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



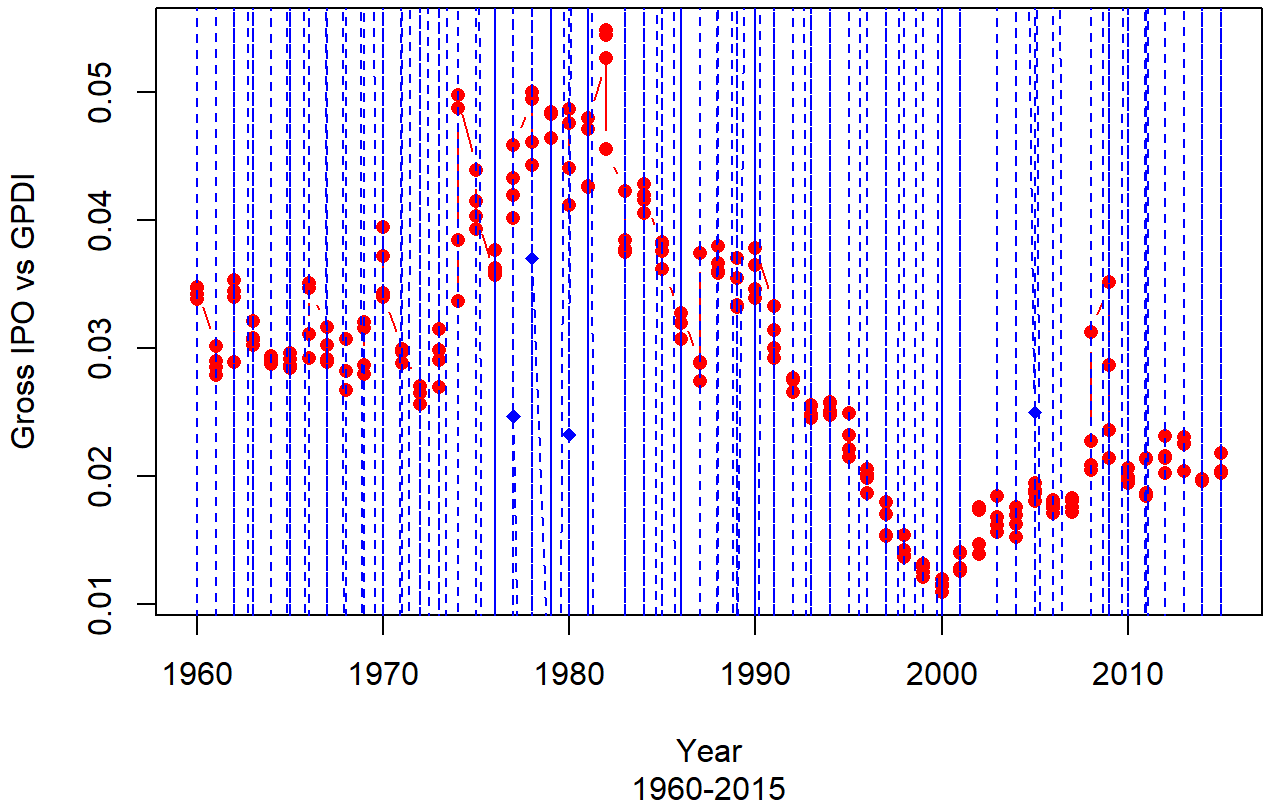

```
#DPRatio vs NDG growth over time
x = combined_data_DPRatio$Year
y1 = combined_data_DPRatio$DPRatio
y2 = combined_data_DPRatio$Nondurable.goods_growth
plot(x, y1, type="b", pch=16, col="red", main="DPRatio vs NDG growth over time, 1960 - 2015", su
b="1960-2015",
  xlab="Year", ylab="DPRatio vs NDG", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("DPRatio", "NDG growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



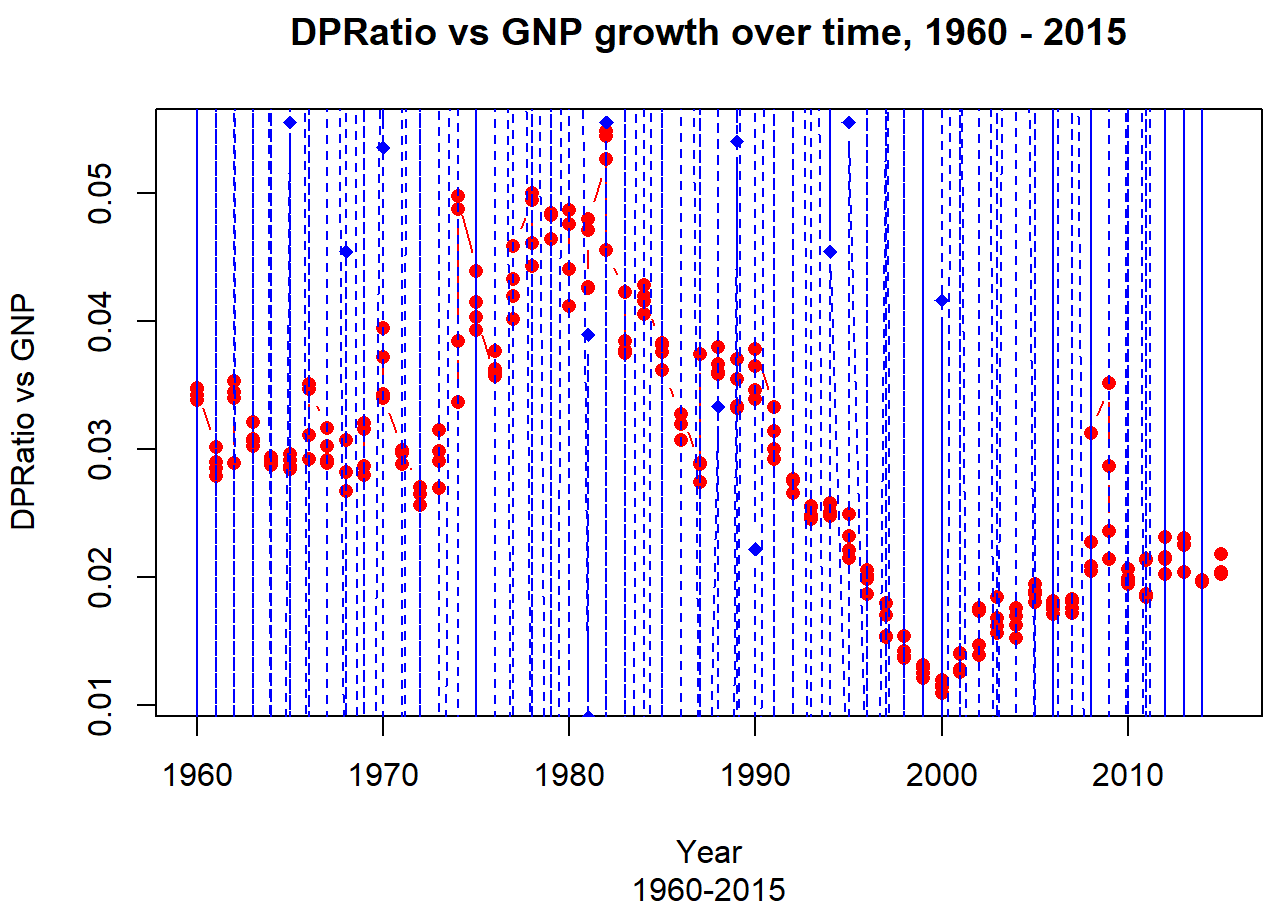

```
#DPRatio vs PCE growth over time
x = combined_data_DPRatio$Year
y1 = combined_data_DPRatio$DPRatio
y2 = combined_data_DPRatio$Personal.consumption.expenditures_growth
plot(x, y1, type="b", pch=16, col="red", main="DPRatio vs PCE growth over time, 1960 - 2015", su
b="1960-2015",
  xlab="Year", ylab="DPRatio vs PCE", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("DPRatio", "PCE growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```




```
#DPRatio vs GPDI growth over time
x = combined_data_DPRatio$Year
y1 = combined_data_DPRatio$DPRatio
y2 = combined_data_DPRatio$Gross.private.domestic.investment_growth
plot(x, y1, type="b", pch=16, col="red", main="DPRatio vs GPDI growth over time, 1960 - 2015", s
ub="1960-2015",
  xlab="Year", ylab="Gross IPO vs GPDI", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("DPRatio", "GPDI growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```

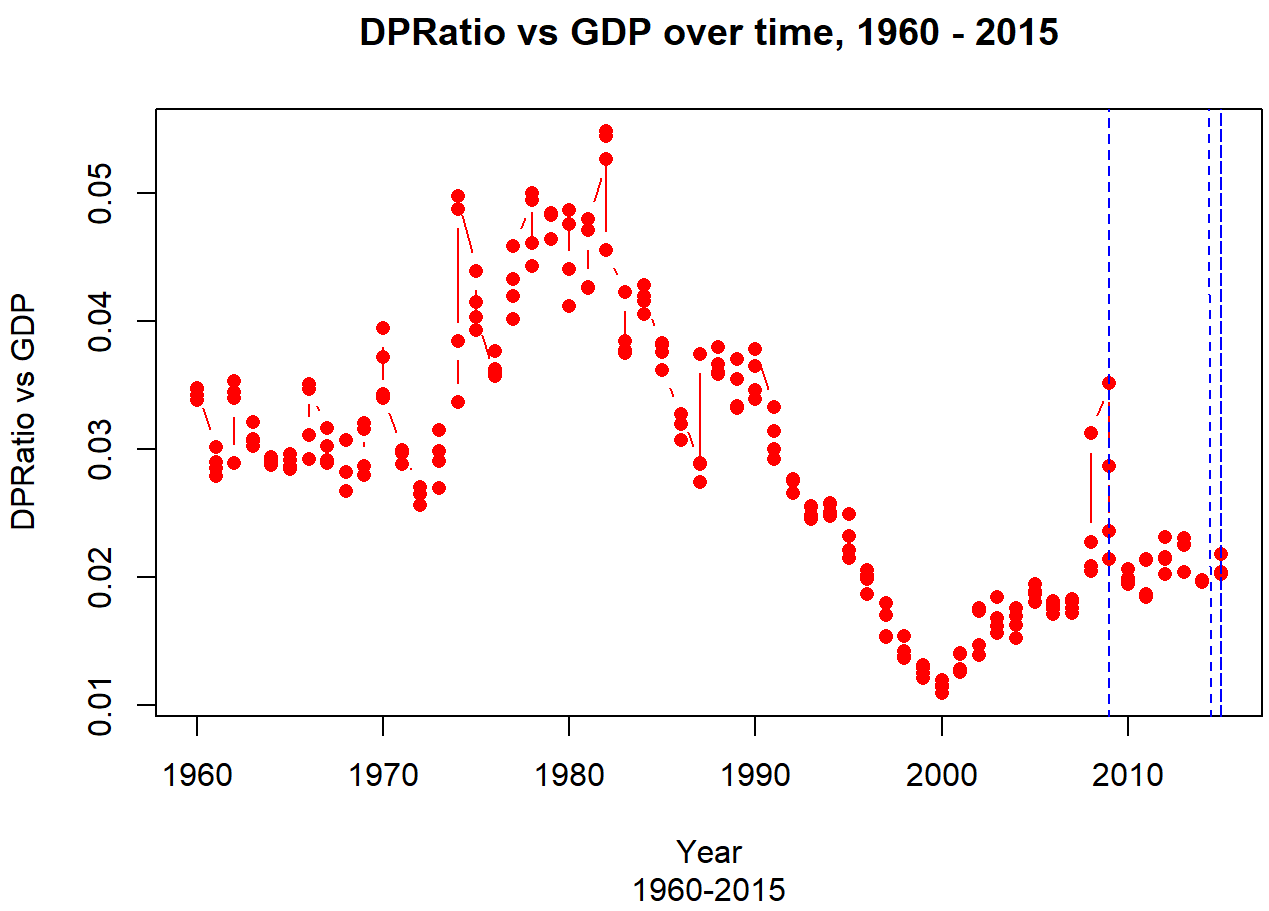



```
#Gross IPO vs GNP growth over time
x = combined_data_DPRatio$Year
y1 = combined_data_DPRatio$DPRatio
y2 = combined_data_DPRatio$Gross.national.product_growth
plot(x, y1, type="b", pch=16, col="red", main="DPRatio vs GNP growth over time, 1960 - 2015", su
b="1960-2015",
  xlab="Year", ylab="DPRatio vs GNP", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("DPRatio", "GNP growth"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```

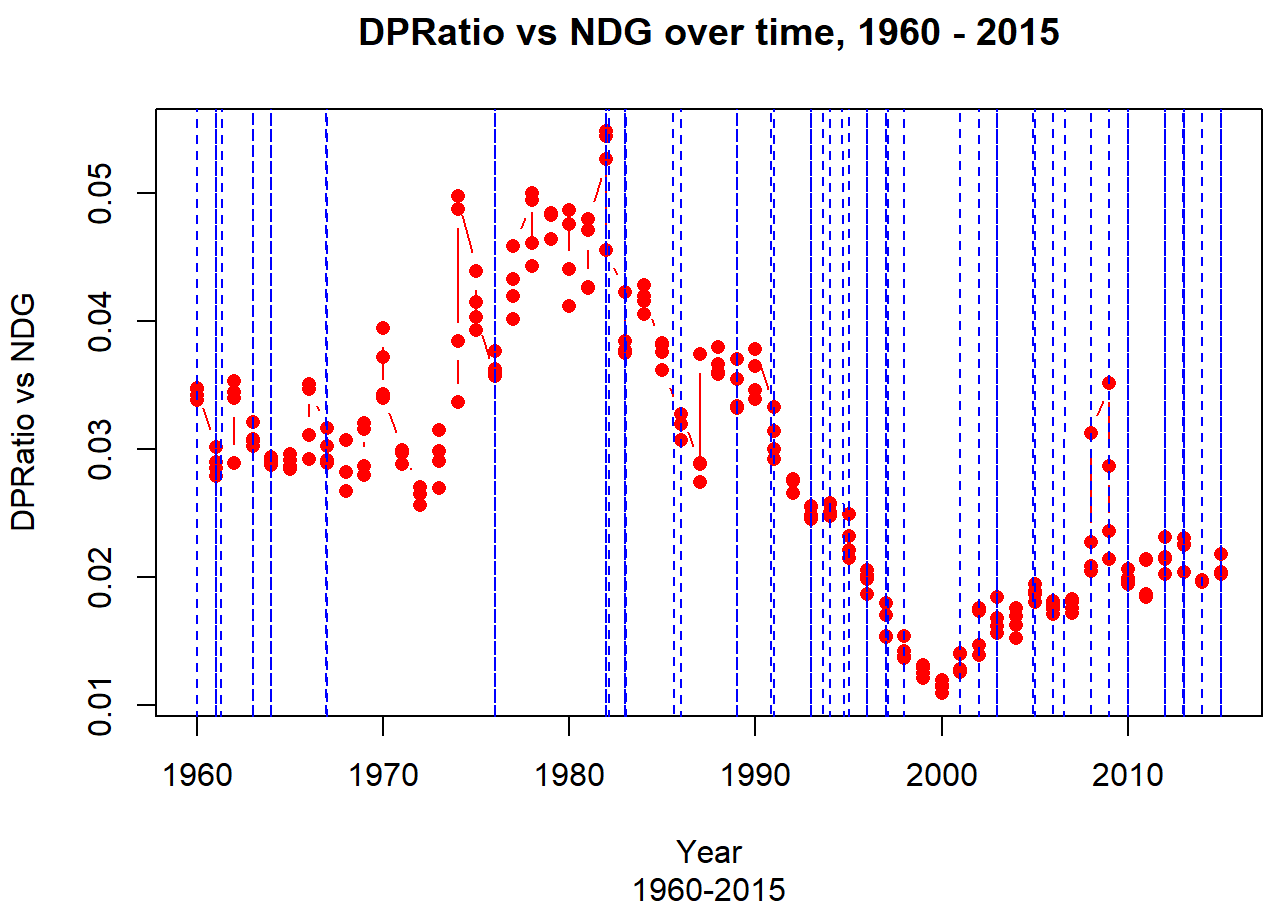


[DPRatio over time vs various GDP Measures]

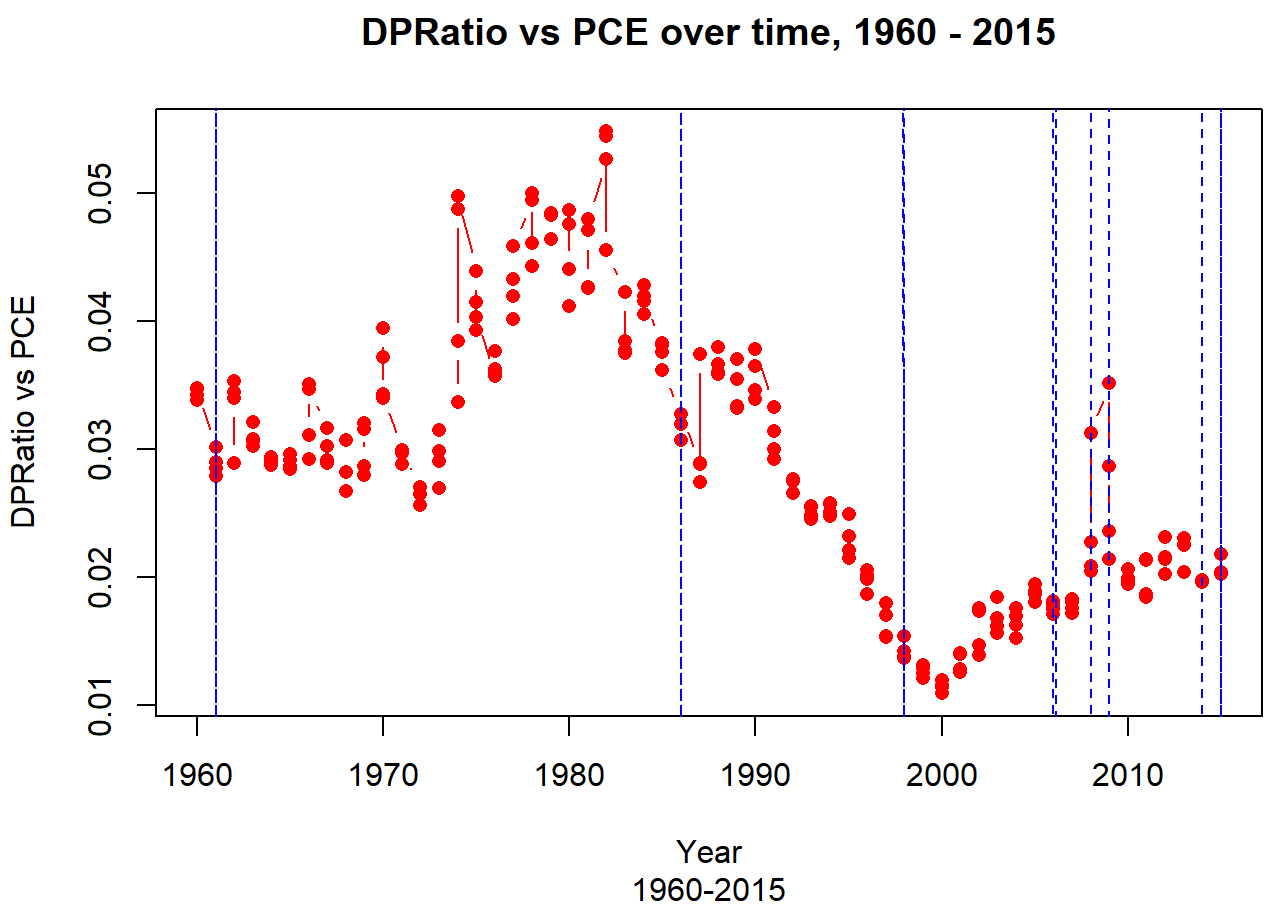
```
#DPRatio vs GDP over time
x = combined_data_DPRatio$Year
y1 = combined_data_DPRatio$DPRatio
y2 = combined_data_DPRatio$GDP
plot(x, y1, type="b", pch=16, col="red", main="DPRatio vs GDP over time, 1960 - 2015", sub="1960
-2015",
  xlab="Year", ylab="DPRatio vs GDP", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("DPRatio", "GDP"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



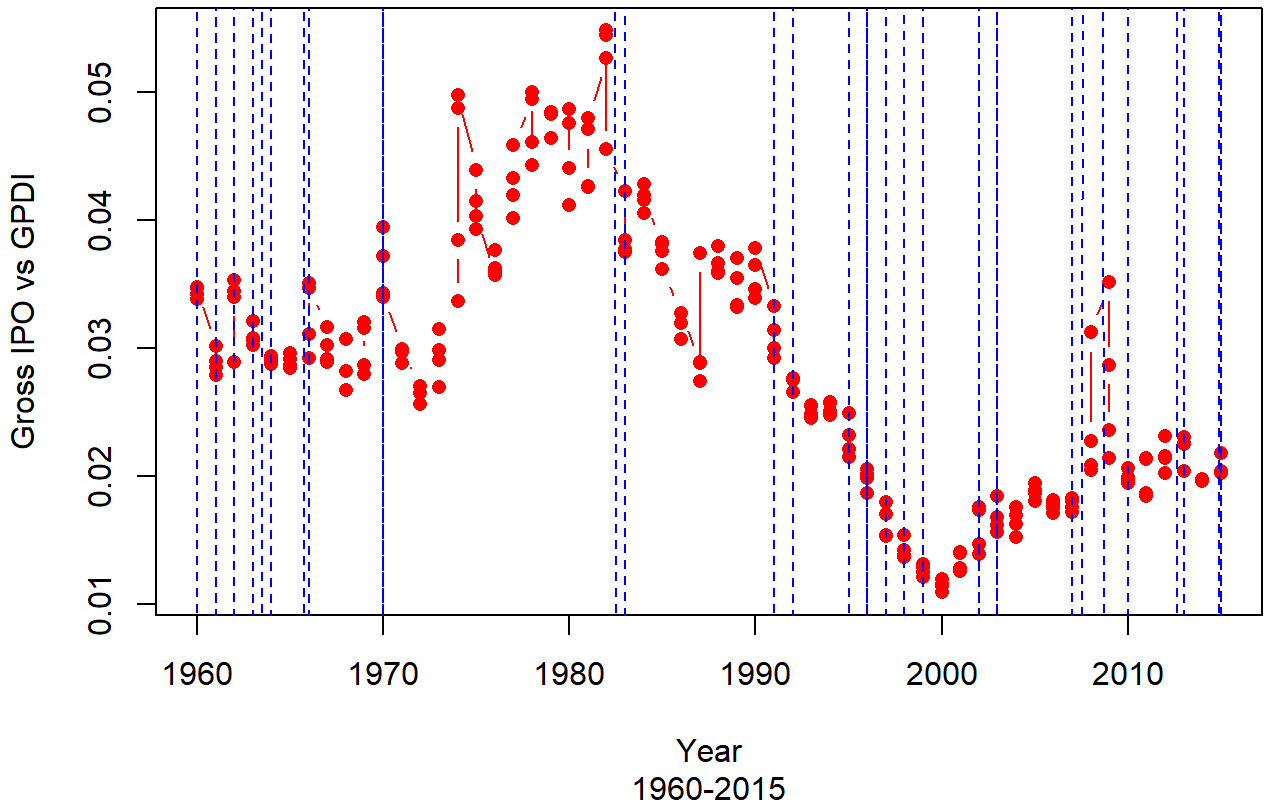

```
#DPRatio vs NDG over time
x = combined_data_DPRatio$Year
y1 = combined_data_DPRatio$DPRatio
y2 = combined_data_DPRatio$Nondurable.goods
plot(x, y1, type="b", pch=16, col="red", main="DPRatio vs NDG over time, 1960 - 2015", sub="1960
-2015",
  xlab="Year", ylab="DPRatio vs NDG", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("DPRatio", "NDG"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



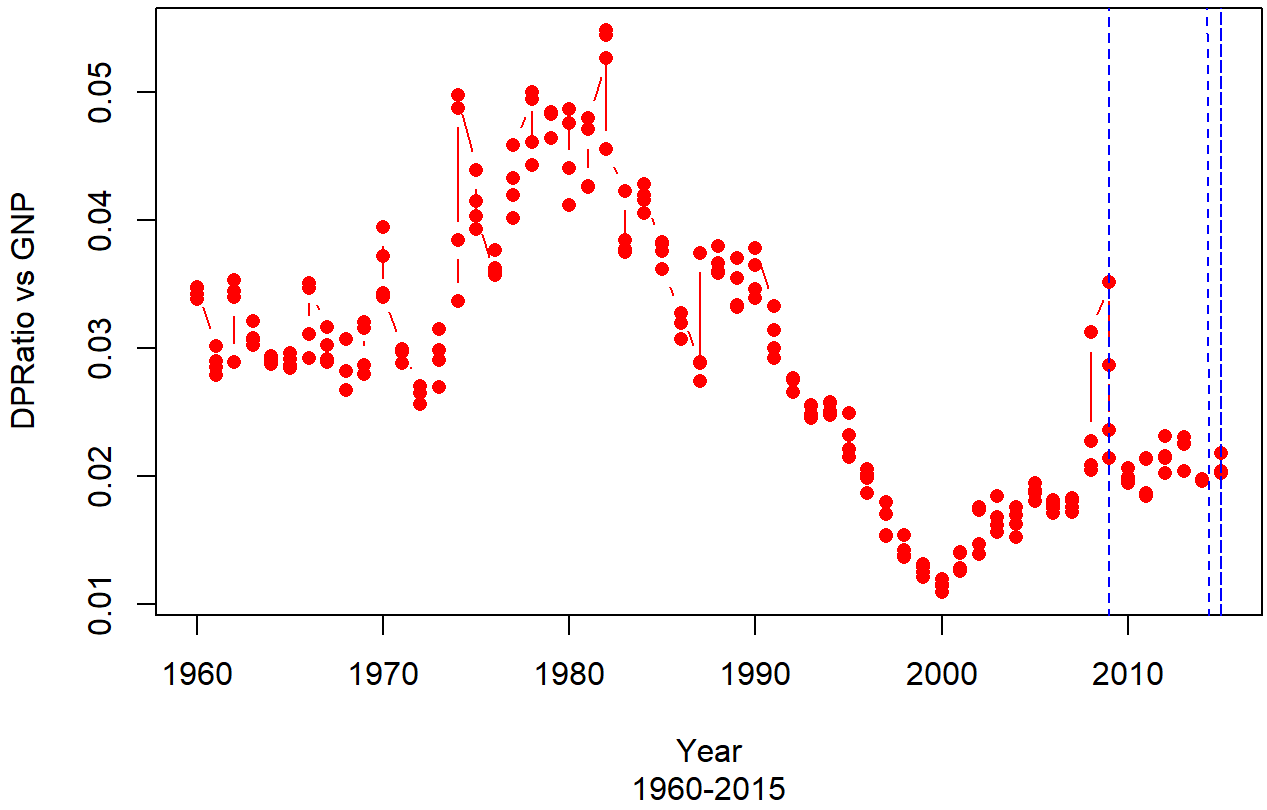

```
#DPRatio vs PCE over time
x = combined_data_DPRatio$Year
y1 = combined_data_DPRatio$DPRatio
y2 = combined_data_DPRatio$Personal.consumption.expenditures
plot(x, y1, type="b", pch=16, col="red", main="DPRatio vs PCE over time, 1960 - 2015", sub="1960
-2015",
  xlab="Year", ylab="DPRatio vs PCE", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("DPRatio", "PCE"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```




```
#DPRatio vs GPDI over time
x = combined_data_DPRatio$Year
y1 = combined_data_DPRatio$DPRatio
y2 = combined_data_DPRatio$Gross.private.domestic.investment
plot(x, y1, type="b", pch=16, col="red", main="DPRatio vs GPDI over time, 1960 - 2015", sub="196
0-2015",
  xlab="Year", ylab="Gross IPO vs GPDI", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("DPRatio", "GPDI"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```




```
#Gross IPO vs GNP over time
x = combined_data_DPRatio$Year
y1 = combined_data_DPRatio$DPRatio
y2 = combined_data_DPRatio$Gross.national.product
plot(x, y1, type="b", pch=16, col="red", main="DPRatio vs GNP over time, 1960 - 2015", sub="1960
-2015",
  xlab="Year", ylab="DPRatio vs GNP", xlim=c(1960, 2015))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2005, 90, legend=c("DPRatio", "GNP"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```




```
#Run correlation tests with equity premium, risk aversion measures and GDP growth rate measures
```

```
#Run correlation tests between different GDP growth measures and gross IPO
#Years: 1960 to 2015
#Correlation with GNP growth rates
cor(combined_data_IPO_Gross$IPO_Gross, as.numeric(combined_data_IPO_Gross$Gross.national.product
_growth))
```

```
## [1] 0.03576855
```

```
#Correlation with GDP growth rates
cor(combined_data_IPO_Gross$IPO_Gross,  as.numeric(combined_data_IPO_Gross$GDP_growth))
```

```
## [1] 0.02725994
```

```
#Correlation with GPDI growth rates
cor(combined_data_IPO_Gross$IPO_Gross, as.numeric(combined_data_IPO_Gross$Gross.private.domesti
c.investment_growth))
```

```
## [1] -0.1175961
```

```
#Correlation with nondurable goods growth rates
cor(combined_data_IPO_Gross$IPO_Gross, as.numeric(combined_data_IPO_Gross$Nondurable.goods_growth))
```

```
## [1] -0.08787011
```

```
#Correlation with personal consumption expenditures growth rates
cor(combined_data_IPO_Gross$IPO_Gross, as.numeric(combined_data_IPO_Gross$Personal.consumption.expenditures_growth))
```

```
## [1] -0.07822005
```

```
#Run correlation tests between different GDP growth measures and DPRatio
#Years: 1960 to 2015 (Full Data)
#Correlation with GNP
cor(combined_data_DPRatio$DPRatio, as.numeric(combined_data_DPRatio$Gross.national.product_growth))
```

```
## [1] -2.182696e-05
```

```
#Correlation with GDP
cor(combined_data_DPRatio$DPRatio,  as.numeric(combined_data_DPRatio$GDP_growth))
```

```
## [1] 0.0119246
```

```
#Correlation with GPDI
cor(combined_data_DPRatio$DPRatio, as.numeric(combined_data_DPRatio$Gross.private.domestic.investment_growth))
```

```
## [1] 0.01262046
```

```
#Correlation with nondurable goods
cor(combined_data_DPRatio$DPRatio, as.numeric(combined_data_DPRatio$Nondurable.goods_growth))
```

```
## [1] -0.04452659
```

```
#Correlation with personal consumption expenditures
cor(combined_data_DPRatio$DPRatio, as.numeric(combined_data_DPRatio$Personal.consumption.expenditures_growth))
```

```
## [1] -0.05251419
```

```
#Run correlation tests between different GDP growth measures and EPBound
#Years: 1960 to 2015
#Correlation with GNP
cor(combined_data_EPBound$EPBound, as.numeric(combined_data_EPBound$Gross.national.product_growth))
```

```
## [1] -0.2422477
```

```
#Correlation with GDP
cor(combined_data_EPBound$EPBound,  as.numeric(combined_data_EPBound$GDP_growth))
```

```
## [1] -0.2402403
```

```
#Correlation with GPDI
cor(combined_data_EPBound$EPBound, as.numeric(combined_data_EPBound$Gross.private.domestic.investment_growth))
```

```
## [1] -0.0538396
```

```
#Correlation with nondurable goods
cor(combined_data_EPBound$EPBound, as.numeric(combined_data_EPBound$Nondurable.goods_growth))
```

```
## [1] -0.1028724
```

```
#Correlation with personal consumption expenditures
cor(combined_data_EPBound$EPBound, as.numeric(combined_data_EPBound$Personal.consumption.expenditures_growth))
```

```
## [1] -0.1902982
```

```
#Run correlation tests between different GDP growth measures and PVS
#Years: 1960 to 2015
#Correlation with GNP
cor(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Gross.national.product_growth))
```

```
## [1] 0.1413801
```

```
#Correlation with GDP
cor(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$GDP_growth))
```

```
## [1] 0.123479
```

```
#Correlation with GPDI
cor(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Gross.private.domestic.inves
tment_growth))
```

```
## [1] 0.008233573
```

```
#Correlation with nondurable goods
cor(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Nondurable.goods_growth))
```

```
## [1] -0.09259729
```

```
#Correlation with personal consumption expenditures
cor(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Personal.consumption.expendi
tures_growth))
```

```
## [1] -0.007097414
```

```
#Run correlation tests between different GDP growth measures and Surplus Consumption Ratio
#Years: 1960 to 2015
#Correlation with GNP
cor(combined_data_SCR$SCR, as.numeric(combined_data_SCR$Gross.national.product_growth))
```

```
## [1] 0.1059663
```

```
#Correlation with GDP
cor(combined_data_SCR$SCR,  as.numeric(combined_data_SCR$GDP_growth))
```

```
## [1] 0.08832823
```

```
#Correlation with GPDI
cor(combined_data_SCR$SCR, as.numeric(combined_data_SCR$Gross.private.domestic.investment_growt
h))
```

```
## [1] 0.01324125
```

```
#Correlation with nondurable goods
cor(combined_data_SCR$SCR, as.numeric(combined_data_SCR$Nondurable.goods_growth))
```

```
## [1] -0.03902723
```

```
#Correlation with personal consumption expenditures
cor(combined_data_SCR$SCR, as.numeric(combined_data_SCR$Personal.consumption.expenditures_growth))
```

```
## [1] 0.05006106
```

```
#Run correlation tests between different GDP growth measures and Aggregate Risk Aversion
#1990 to 2010
#Correlation with GNP
cor(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion, as.numeric(combined_data_Aggregate_Risk_Aversion$Gross.national.product_growth))
```

```
## [1] -0.1939327
```

```
#Correlation with GDP
cor(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion,as.numeric(combined_data_Aggregate_Risk_Aversion$GDP_growth))
```

```
## [1] -0.1921884
```

```
#Correlation with GPDI
cor(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion, as.numeric(combined_data_Aggregate_Risk_Aversion$Gross.private.domestic.investment_growth))
```

```
## [1] 0.01566911
```

```
#Correlation with nondurable goods
cor(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion, as.numeric(combined_data_Aggregate_Risk_Aversion$Nondurable.goods_growth))
```

```
## [1] 0.01650201
```

```
#Correlation with personal consumption expenditures
cor(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion, as.numeric(combined_data_Aggregate_Risk_Aversion$Personal.consumption.expenditures_growth))
```

```
## [1] 0.1151585
```

```
#Run correlation tests between different GDP growth measures and UNE
#Years: 1960 to 2015
#Correlation with GNP
cor(combined_data_UNE$UNE, as.numeric(combined_data_UNE$Gross.national.product_growth))
```

```
## [1] -0.03130488
```

```
#Correlation with GDP
cor(combined_data_UNE$UNE, as.numeric(combined_data_UNE$GDP_growth))
```

```
## [1] -0.02145195
```

```
#Correlation with GPDI
cor(combined_data_UNE$UNE, as.numeric(combined_data_UNE$Gross.private.domestic.investment_growth))
```

```
## [1] -0.057071
```

```
#Correlation with nondurable goods
cor(combined_data_UNE$UNE, as.numeric(combined_data_UNE$Nondurable.goods_growth))
```

```
## [1] -0.05433978
```

```
#Correlation with personal consumption expenditures
cor(combined_data_UNE$UNE, as.numeric(combined_data_UNE$Personal.consumption.expenditures_growth))
```

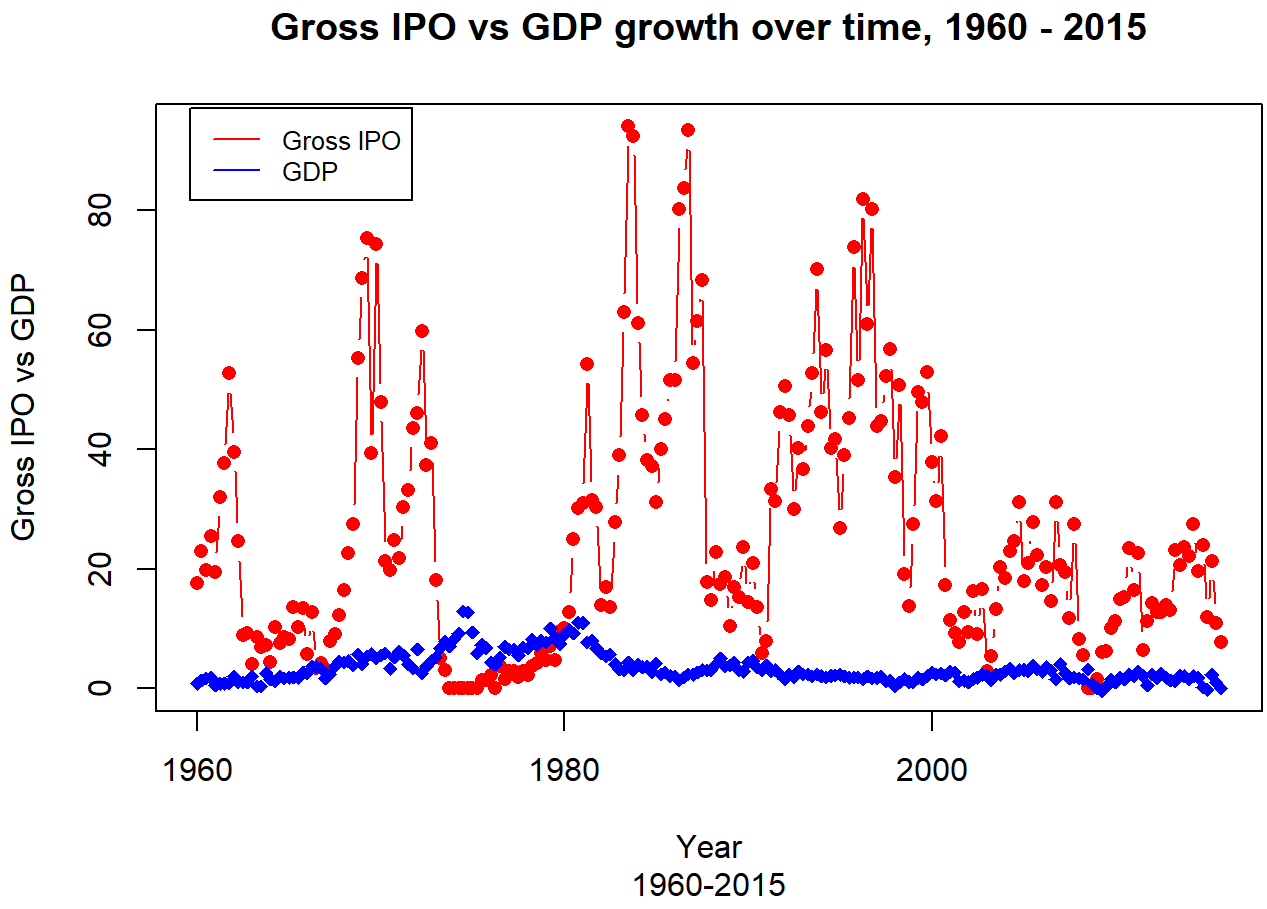
```
## [1] -0.01080821
```

```
#create time series dates for all datasets
date_makers = function(yr, quart){
result <- c()
  if (quart ==  "Q1"){
    x <- paste(yr, '01','01', sep = "-")
    result <- x
  }
  else if (quart ==  "Q2"){
    x <- paste(yr, '04','01', sep = "-")
    result <- x
  }
  else if (quart ==  "Q3"){
    x <- paste(yr, '07','01', sep = "-")
    result <- x
  }
  else {
    x <- paste(yr, '10','01', sep = "-")
    result <- x
  }

return (result)
}
```

```
#create date column in all datasets using date maker function
combined_data_IPO_Gross$Date = as.Date(mapply(date_makers, combined_data_IPO_Gross$Year,
                                              combined_data_IPO_Gross$Quarter))
combined_data_EPBound$Date = as.Date(mapply(date_makers, combined_data_EPBound$Year,
                                              combined_data_EPBound$Quarter))
combined_data_UNE$Date = as.Date(mapply(date_makers, combined_data_UNE$Year,
                                              combined_data_UNE$Quarter))
combined_data_DPRatio$Date = as.Date(mapply(date_makers, combined_data_DPRatio$Year,
                                              combined_data_DPRatio$Quarter))
combined_data_Aggregate_Risk_Aversion$Date = as.Date(mapply(date_makers, combined_data_Aggregate
_Risk_Aversion$Year,
                                              combined_data_Aggregate_Risk_Aversion$Quarter))
combined_data_PVS$Date = as.Date(mapply(date_makers, combined_data_PVS$Year,
                                              combined_data_PVS$Quarter))
combined_data_SCR$Date = as.Date(mapply(date_makers, combined_data_SCR$Year,
                                              combined_data_SCR$Quarter))
```

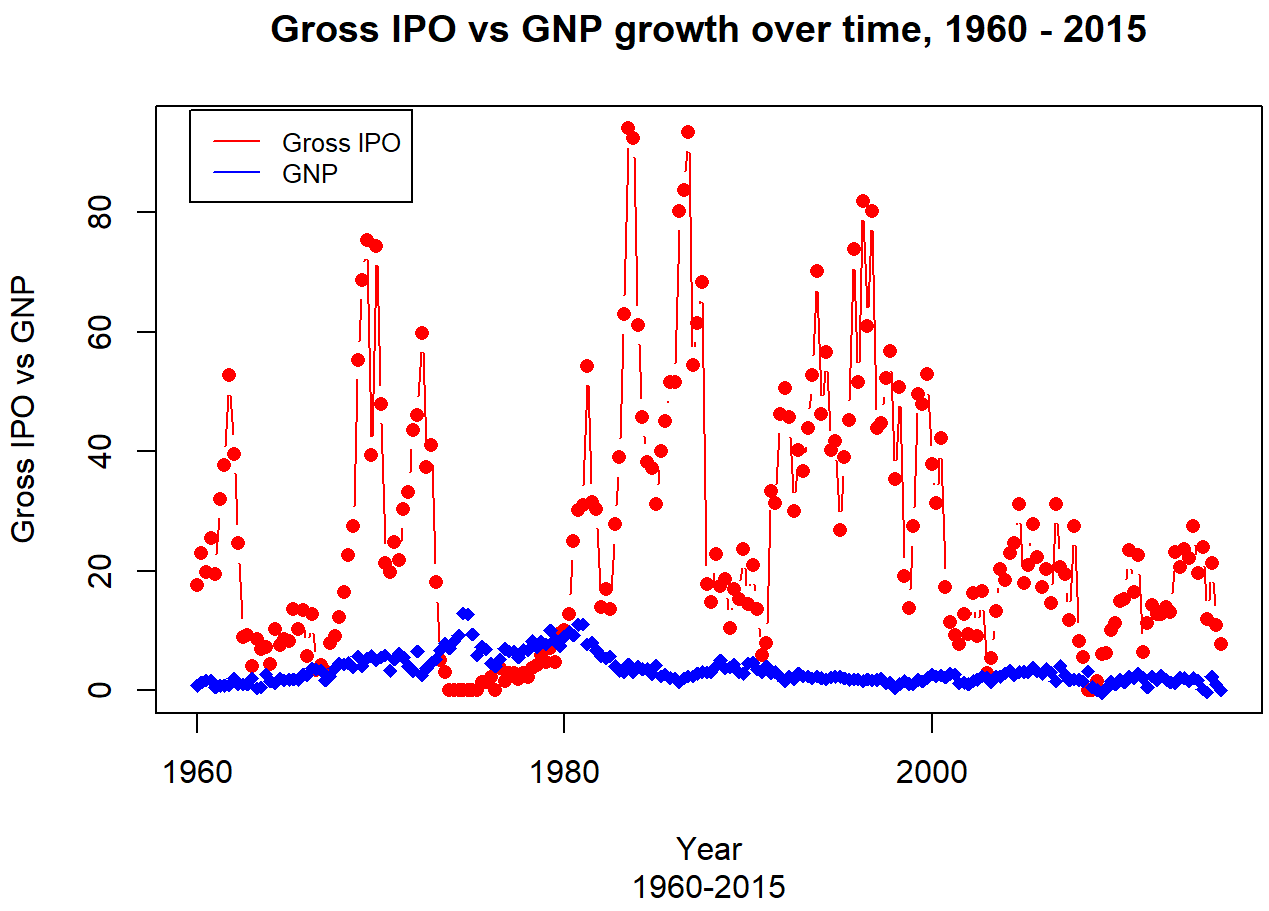
```
#Using the dates, create time series plots
```

```
#Start with Gross IPO


#Gross IPO vs GDP over time
x = combined_data_IPO_Gross$Date
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$GDP
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs GDP growth over time, 1960 - 2015",
sub="1960-2015",
  xlab="Year", ylab="Gross IPO vs GDP")
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("Gross IPO", "GDP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



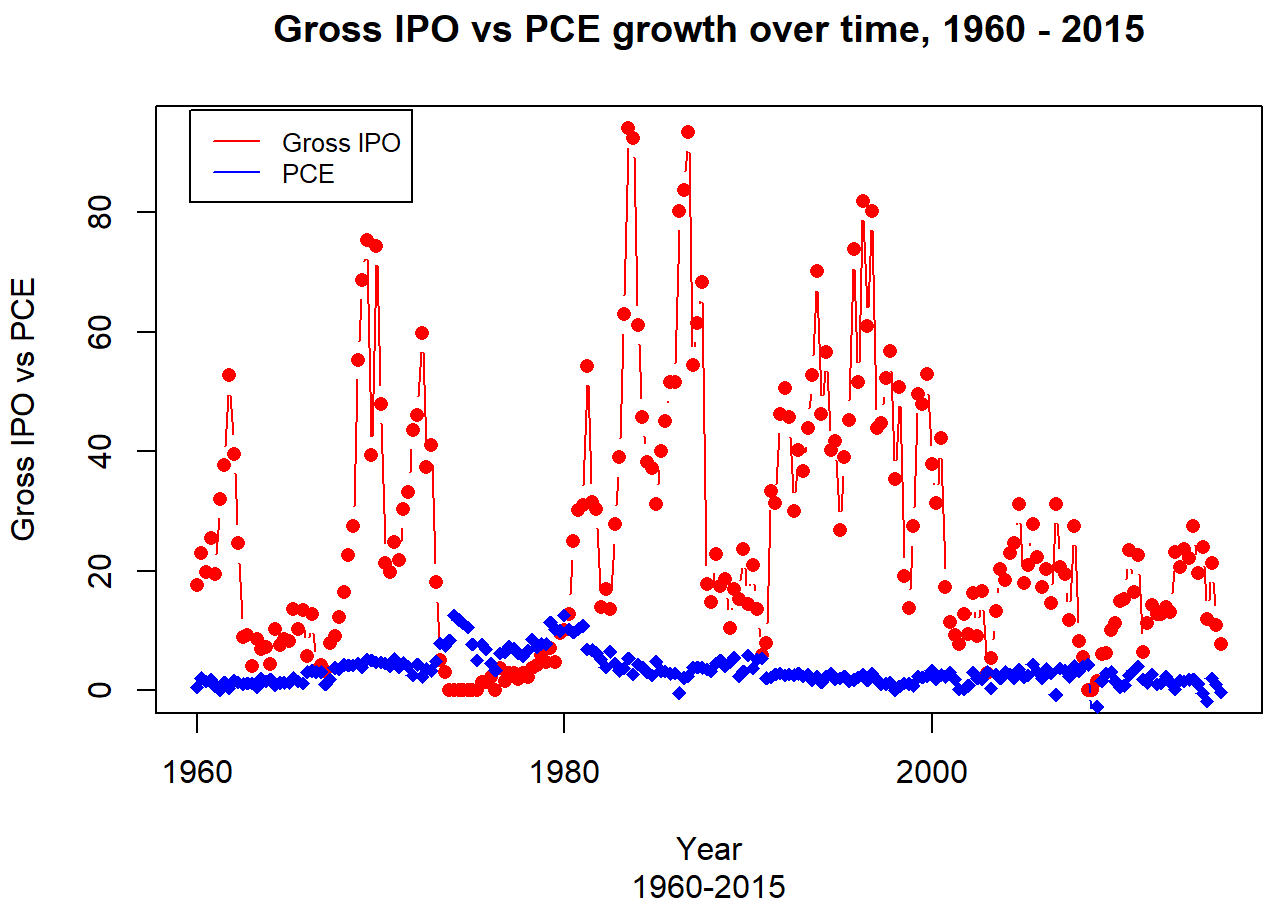

```
## NULL
```

```
#Gross IPO vs GNP over time
x = combined_data_IPO_Gross$Date
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Gross.national.product
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs GNP growth over time, 1960 - 2015",
sub="1960-2015",
  xlab="Year", ylab="Gross IPO vs GNP")
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("Gross IPO", "GNP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



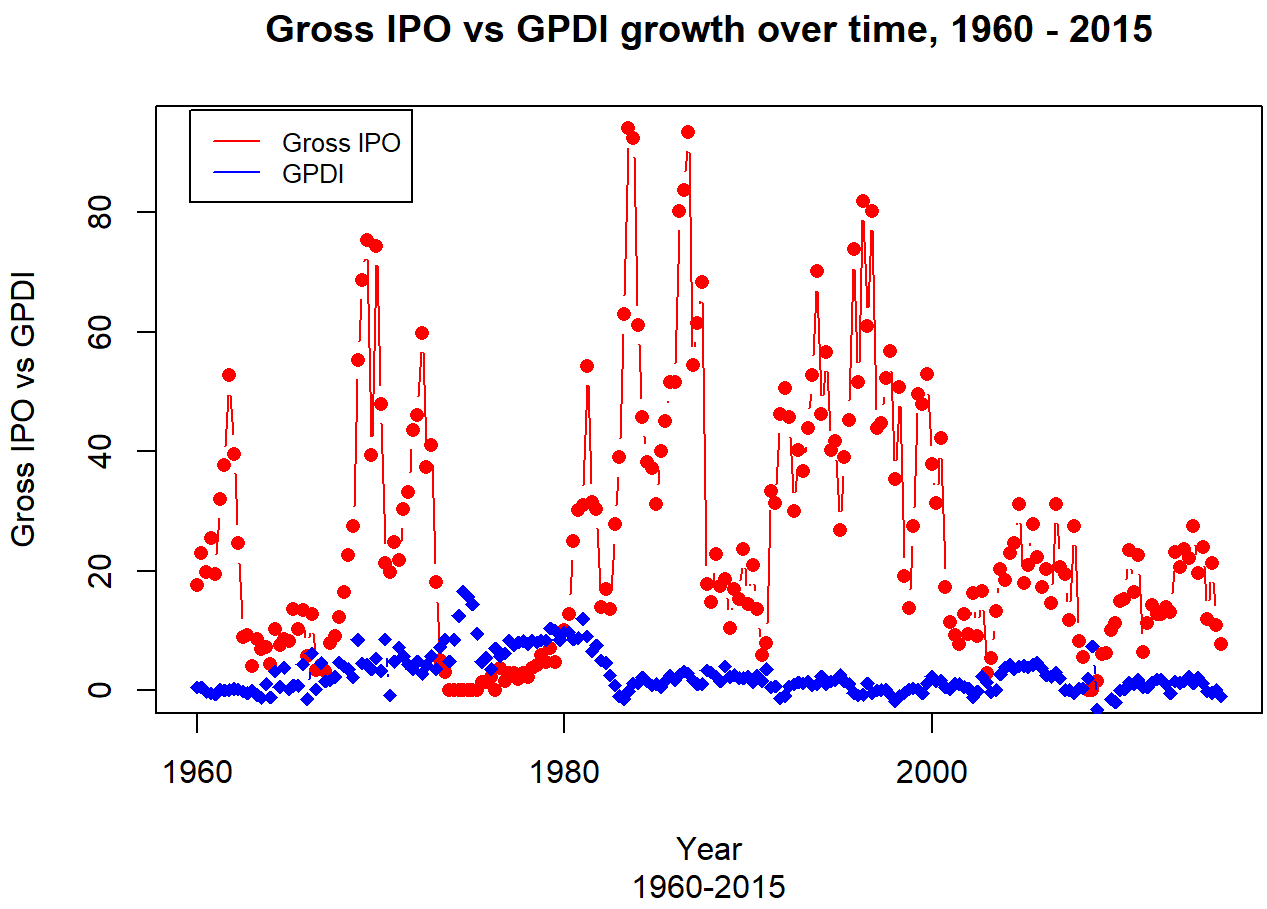

```
## NULL
```

```
#Gross IPO vs PCE over time
x = combined_data_IPO_Gross$Date
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Personal.consumption.expenditures
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs PCE growth over time, 1960 - 2015",
sub="1960-2015",
  xlab="Year", ylab="Gross IPO vs PCE")
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("Gross IPO", "PCE"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



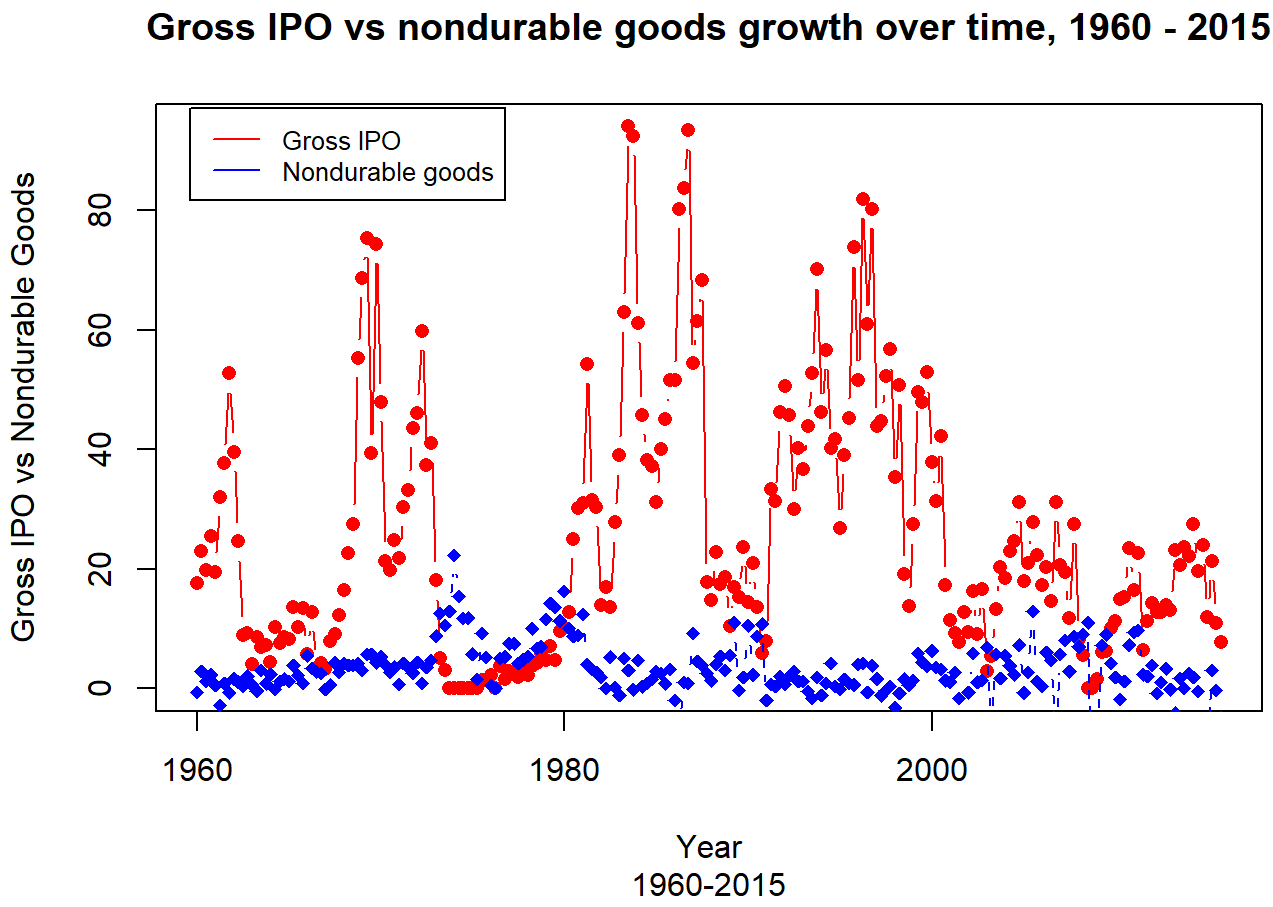

```
## NULL
```

```
#Gross IPO vs GPDI over time
x = combined_data_IPO_Gross$Date
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Gross.private.domestic.investment
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs GPDI growth over time, 1960 - 2015",
sub="1960-2015",
  xlab="Year", ylab="Gross IPO vs GPDI")
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("Gross IPO", "GPDI"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



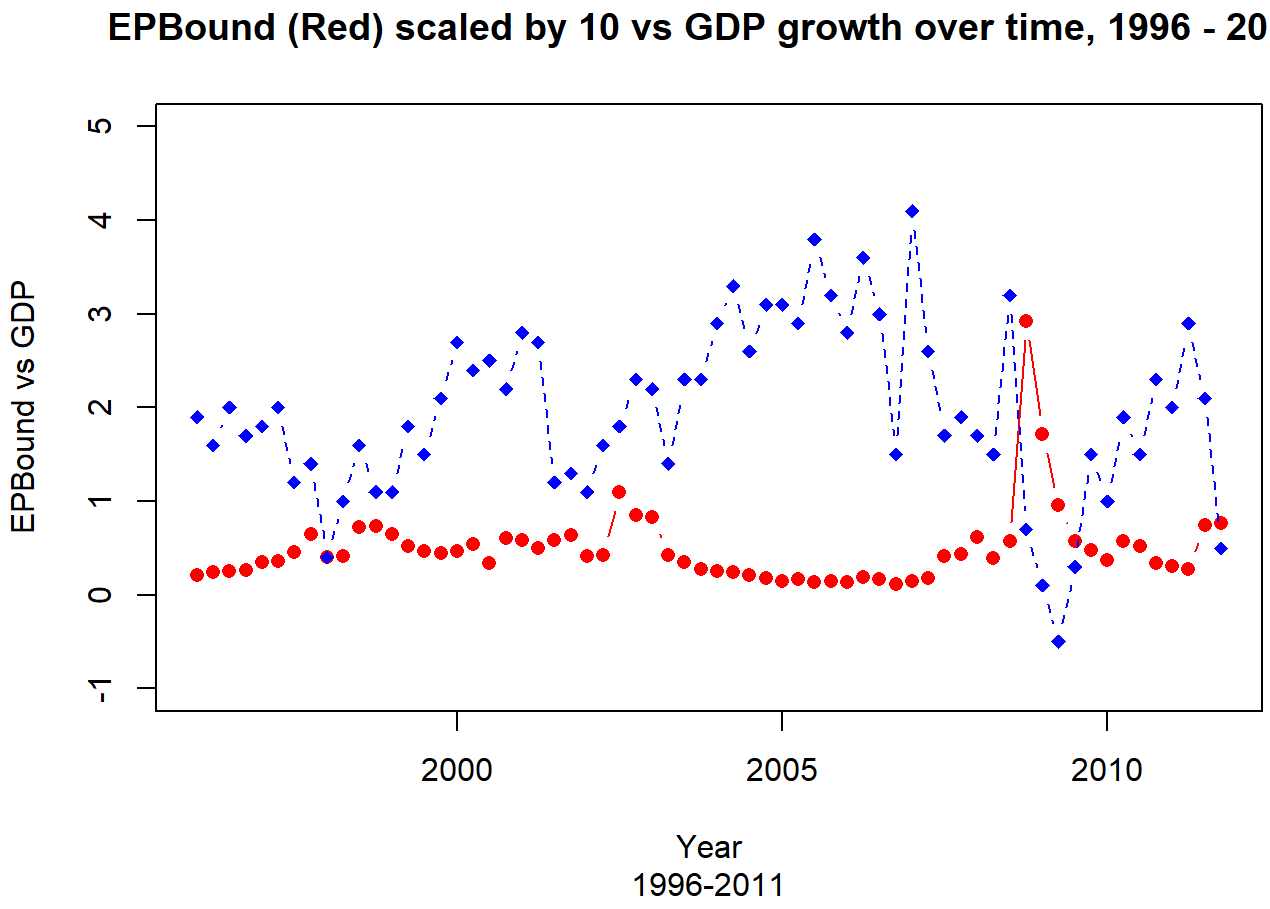

```
## NULL
```

```
#Gross IPO vs PCE over time
x = combined_data_IPO_Gross$Date
y1 = combined_data_IPO_Gross$IPO_Gross
y2 = combined_data_IPO_Gross$Nondurable.goods
plot(x, y1, type="b", pch=16, col="red", main="Gross IPO vs nondurable goods growth over time, 1
960 - 2015", sub="1960-2015",
  xlab="Year", ylab="Gross IPO vs Nondurable Goods")
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("Gross IPO", "Nondurable goods"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



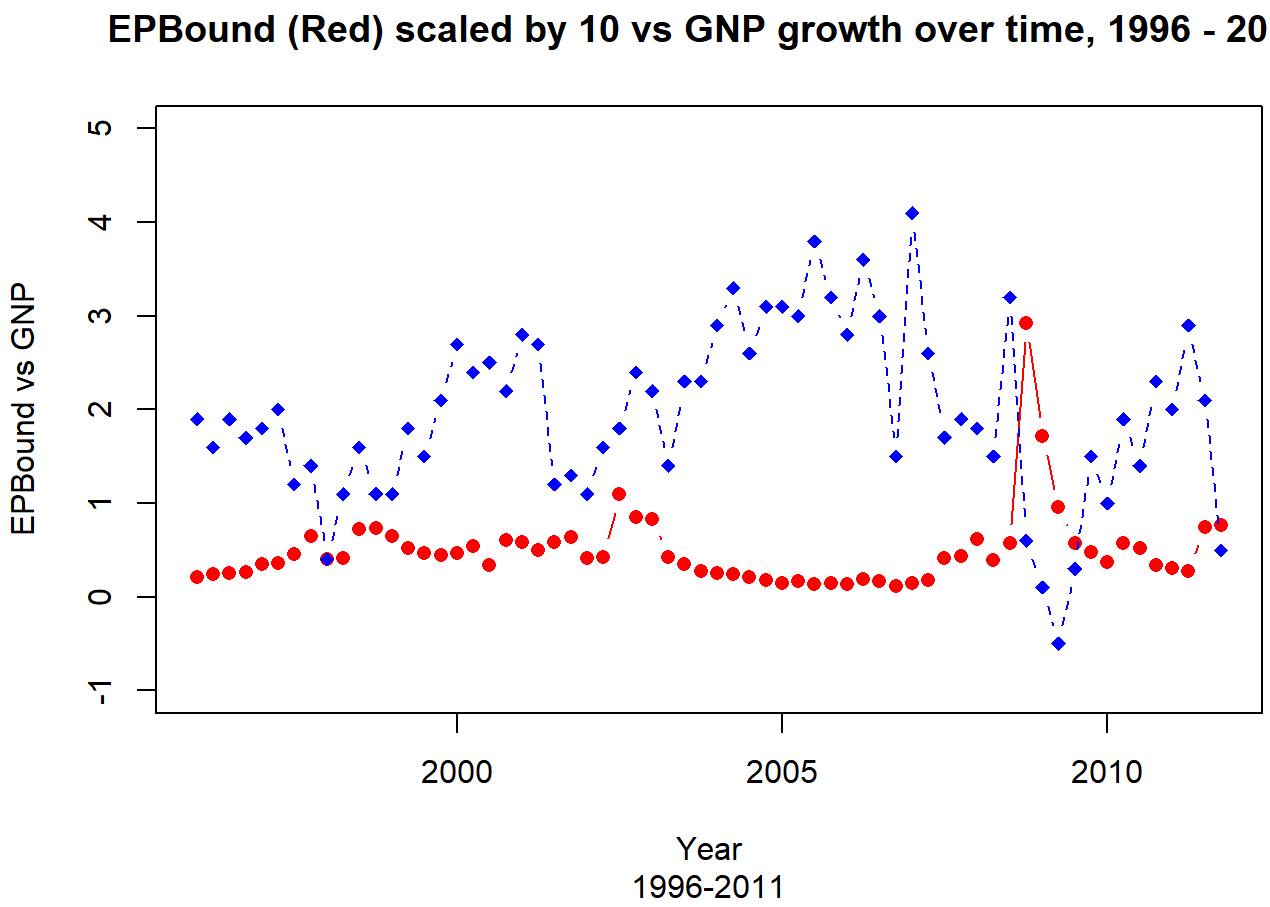

```
## NULL
```

```
#Next EP Bound


#EPBound vs GDP over time
x = combined_data_EPBound$Date
y1 = combined_data_EPBound$EPBound * 10
y2 = combined_data_EPBound$GDP
plot(x, y1, type="b", pch=16, col="red", main="EPBound (Red) scaled by 10 vs GDP growth over tim
e, 1996 - 2011", sub="1996-2011",
  xlab="Year", ylab="EPBound vs GDP", ylim= c(-1, 5))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("EPBound", "GDP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



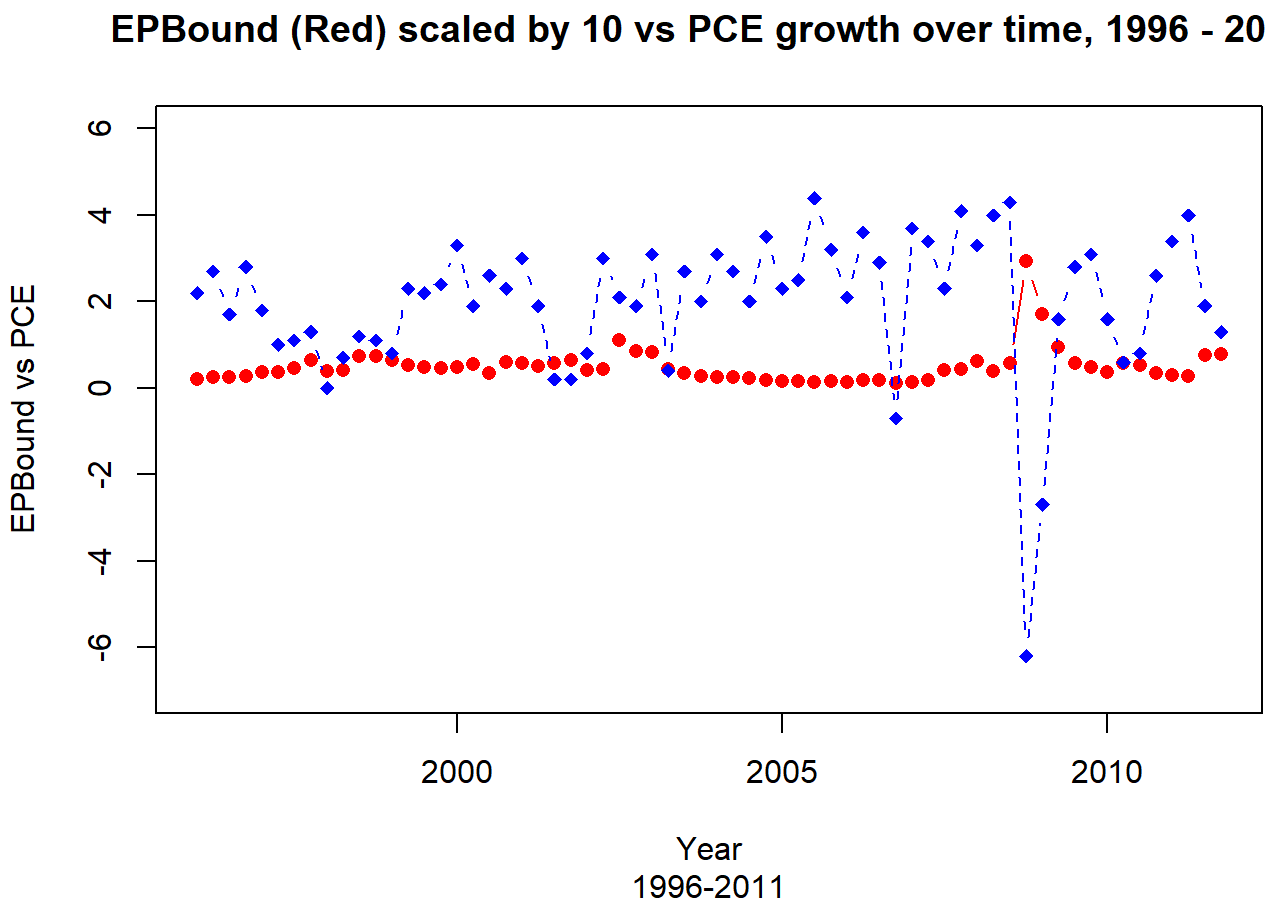

```
## NULL
```

```
#EPBound vs GNP over time
x = combined_data_EPBound$Date
y1 = combined_data_EPBound$EPBound * 10
y2 = combined_data_EPBound$Gross.national.product
plot(x, y1, type="b", pch=16, col="red", main="EPBound (Red) scaled by 10 vs GNP growth over tim
e, 1996 - 2011", sub="1996-2011",
  xlab="Year", ylab="EPBound vs GNP", ylim= c(-1, 5))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("EPBound", "GNP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



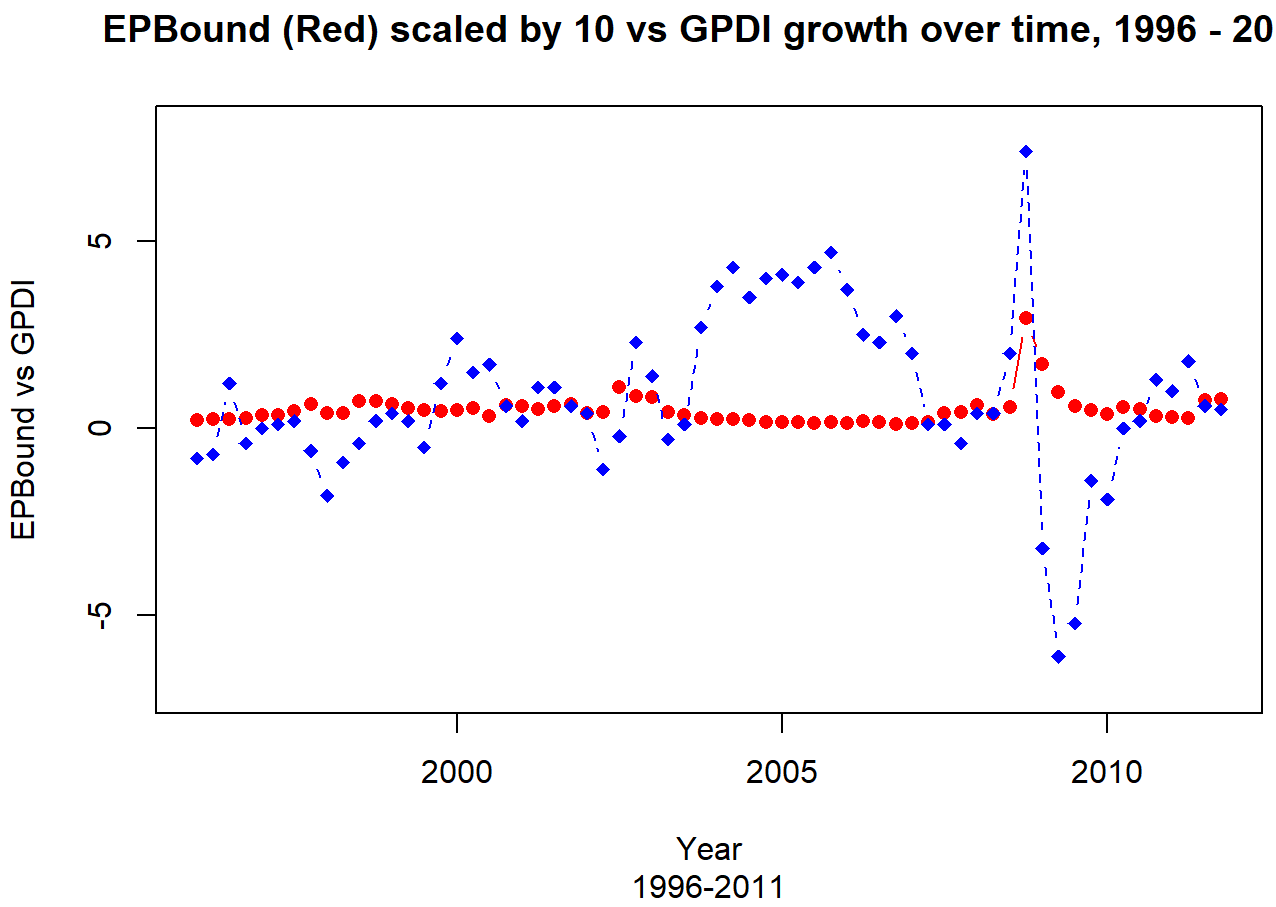

```
## NULL
```

```
#EPBound vs PCE over time
x = combined_data_EPBound$Date
y1 = combined_data_EPBound$EPBound * 10
y2 = combined_data_EPBound$Personal.consumption.expenditures
plot(x, y1, type="b", pch=16, col="red", main="EPBound (Red) scaled by 10 vs PCE growth over tim
e, 1996 - 2011", sub="1996-2011",
  xlab="Year", ylab="EPBound vs PCE", ylim= c(-7, 6))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("EPBound", "PCE"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



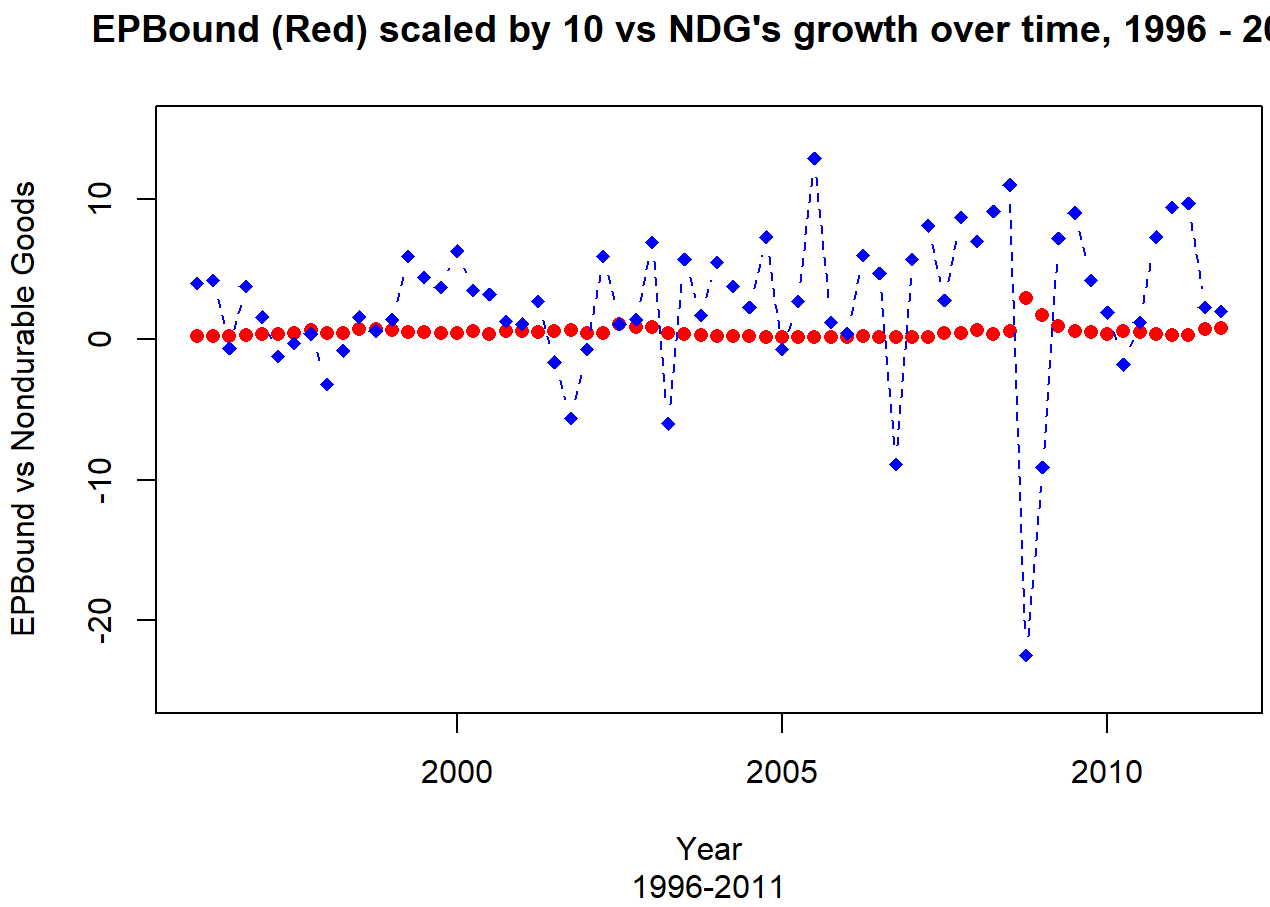

```
## NULL
```

```
#EPBound vs GPDI over time
x = combined_data_EPBound$Date
y1 = combined_data_EPBound$EPBound * 10
y2 = combined_data_EPBound$Gross.private.domestic.investment
plot(x, y1, type="b", pch=16, col="red", main="EPBound (Red) scaled by 10 vs GPDI growth over ti
me, 1996 - 2011", sub="1996-2011",
  xlab="Year", ylab="EPBound vs GPDI", ylim= c(-7, 8))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2000, 0, legend=c("EPBound", "GPDI"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



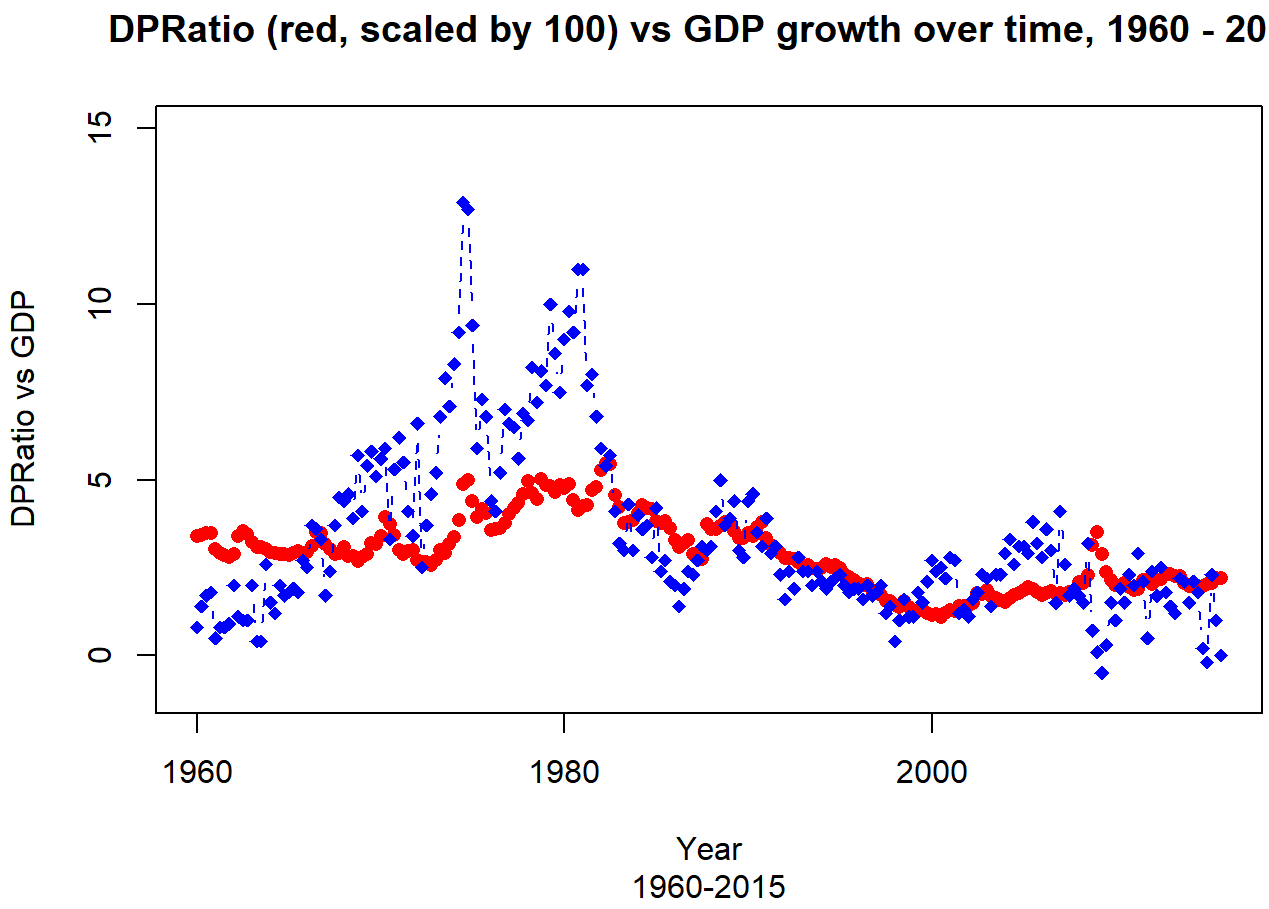

```
## NULL
```

```
#EPBound vs Nondurable Goods over time
x = combined_data_EPBound$Date
y1 = combined_data_EPBound$EPBound * 10
y2 = combined_data_EPBound$Nondurable.goods
plot(x, y1, type="b", pch=16, col="red", main="EPBound (Red) scaled by 10 vs NDG's growth over t
ime, 1996 - 2011", sub="1996-2011",
  xlab="Year", ylab="EPBound vs Nondurable Goods", ylim= c(-25, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(2000, 0, legend=c("EPBound", "Nondurable Goods"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



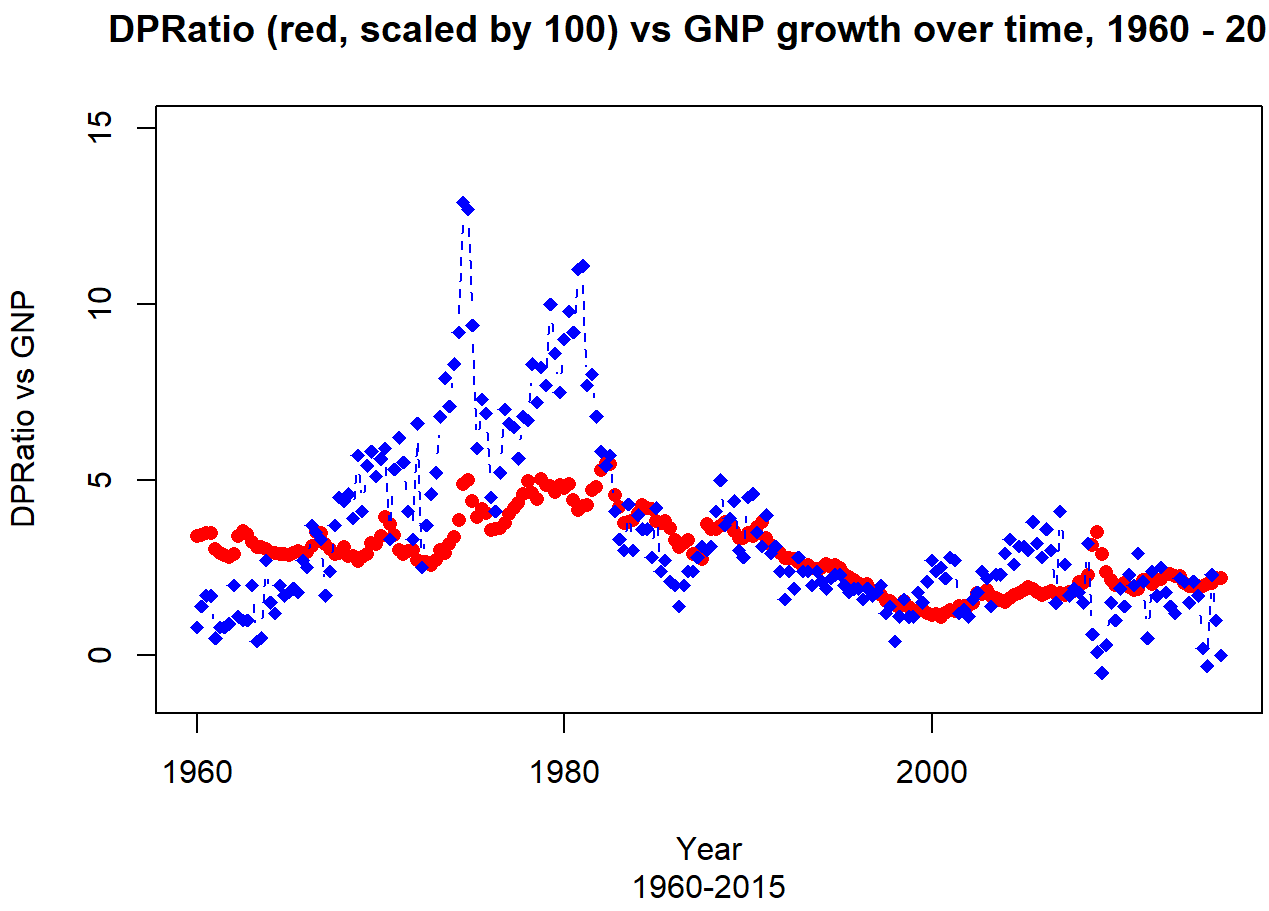

```
## NULL
```

```
#DPRatio vs GDP over time
x = combined_data_DPRatio$Date
y1 = combined_data_DPRatio$DPRatio * 100
y2 = combined_data_DPRatio$GDP
plot(x, y1, type="b", pch=16, col="red", main="DPRatio (red, scaled by 100) vs GDP growth over t
ime, 1960 - 2015", sub="1960-2015",
  xlab="Year", ylab="DPRatio vs GDP", ylim = c(-1, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("DPRatio", "GDP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



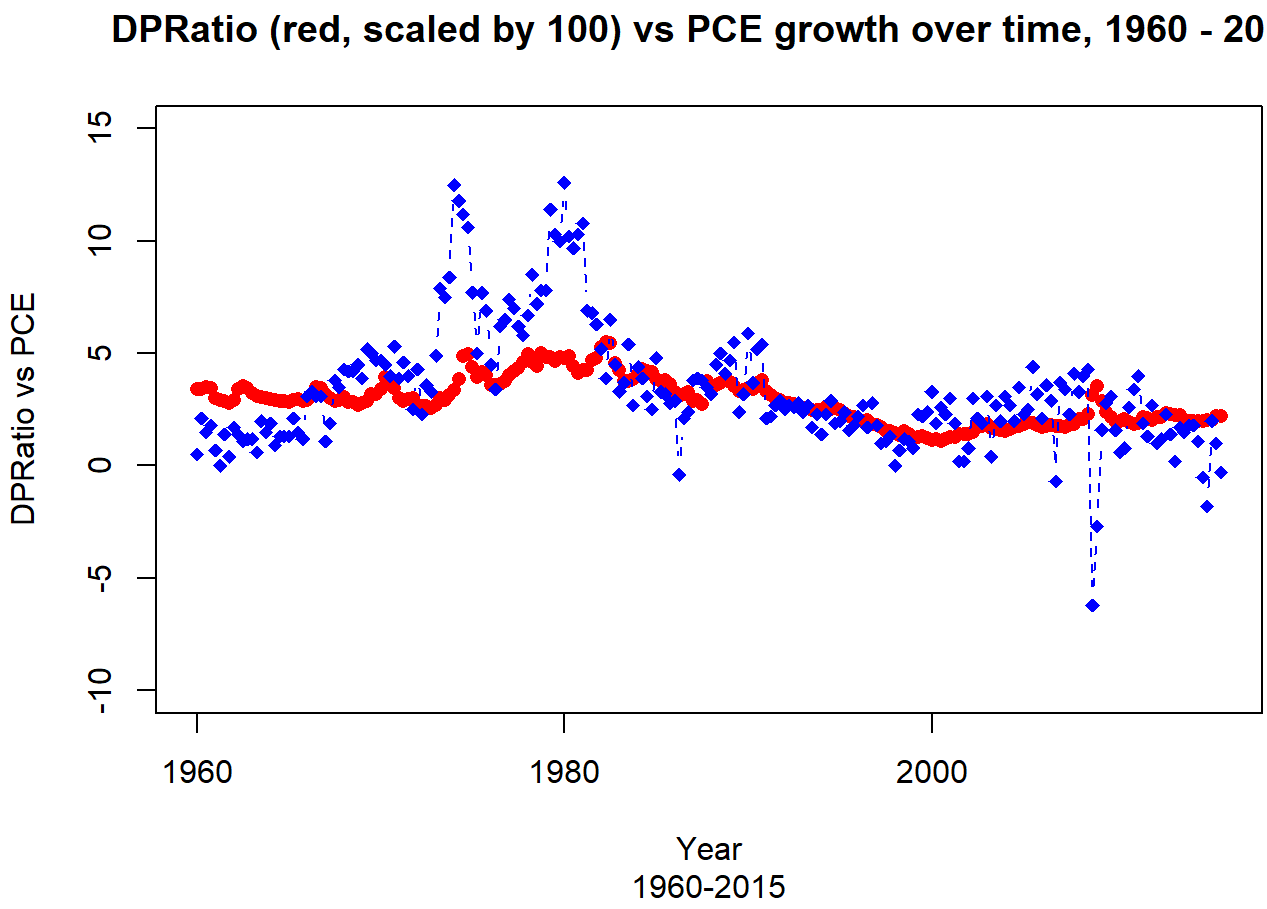

```
## NULL
```

```
#DPRatio vs GNP over time
x = combined_data_DPRatio$Date
y1 = combined_data_DPRatio$DPRatio * 100
y2 = combined_data_DPRatio$Gross.national.product
plot(x, y1, type="b", pch=16, col="red", main="DPRatio (red, scaled by 100) vs GNP growth over t
ime, 1960 - 2015", sub="1960-2015",
  xlab="Year", ylab="DPRatio vs GNP", ylim = c(-1, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("DPRatio", "GNP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



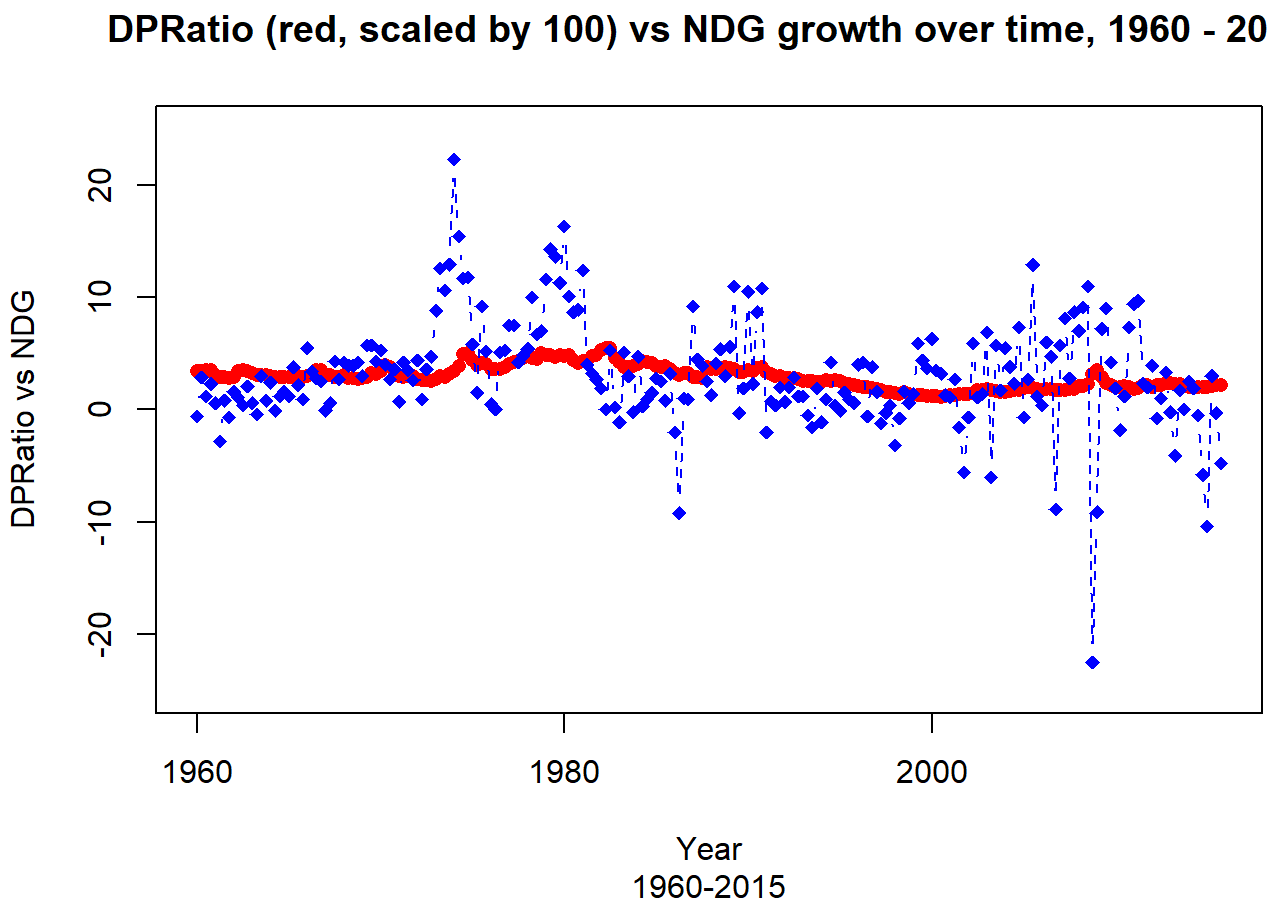

```
## NULL
```

```
#DPRatio vs PCE over time
x = combined_data_DPRatio$Date
y1 = combined_data_DPRatio$DPRatio * 100
y2 = combined_data_DPRatio$Personal.consumption.expenditures
plot(x, y1, type="b", pch=16, col="red", main="DPRatio (red, scaled by 100) vs PCE growth over t
ime, 1960 - 2015", sub="1960-2015",
  xlab="Year", ylab="DPRatio vs PCE", ylim = c(-10, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("DPRatio", "PCE"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



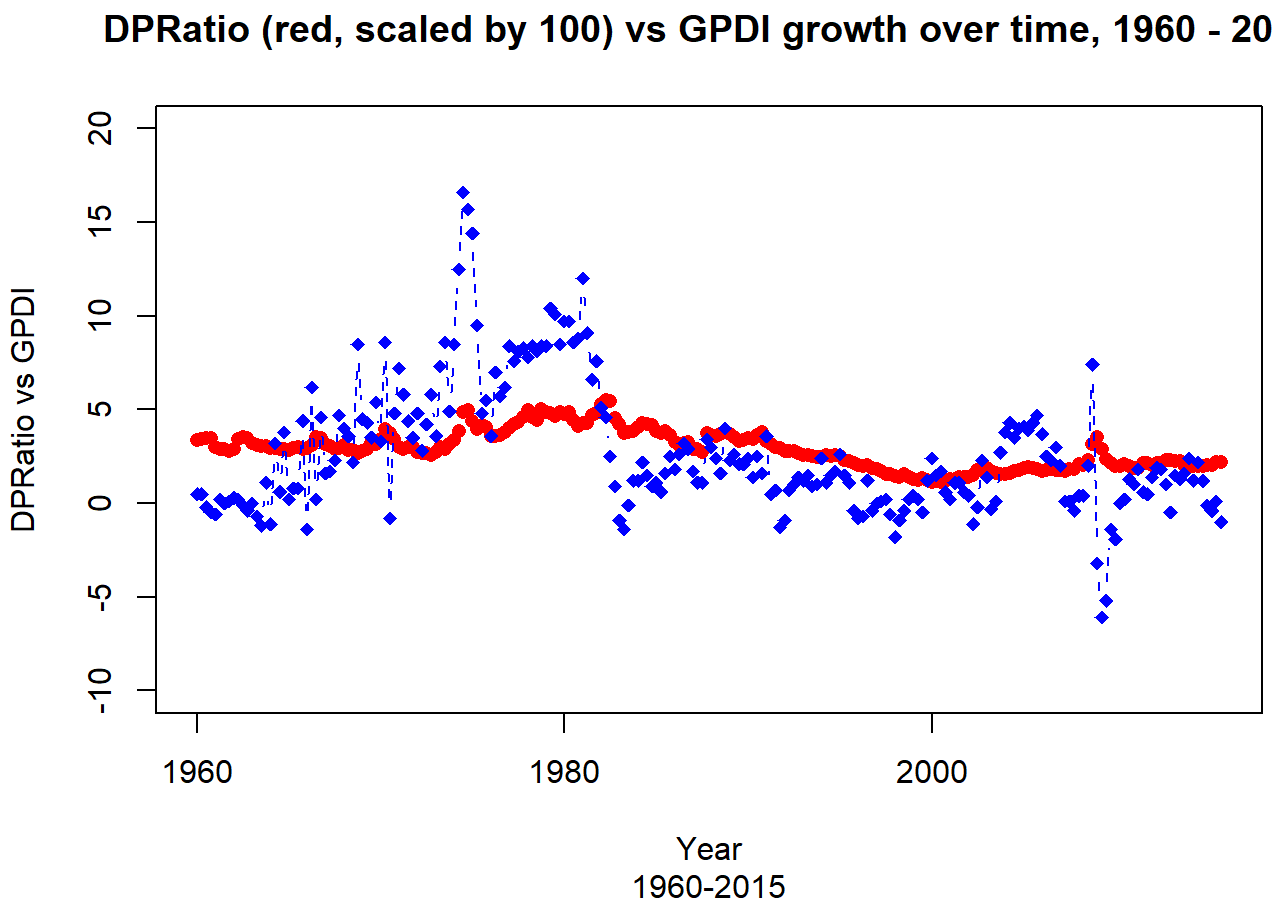

```
## NULL
```

```
#DPRatio vs NDG over time
x = combined_data_DPRatio$Date
y1 = combined_data_DPRatio$DPRatio * 100
y2 = combined_data_DPRatio$Nondurable.goods
plot(x, y1, type="b", pch=16, col="red", main="DPRatio (red, scaled by 100) vs NDG growth over t
ime, 1960 - 2015", sub="1960-2015",
  xlab="Year", ylab="DPRatio vs NDG", ylim = c(-25, 25))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("DPRatio", "NDG"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



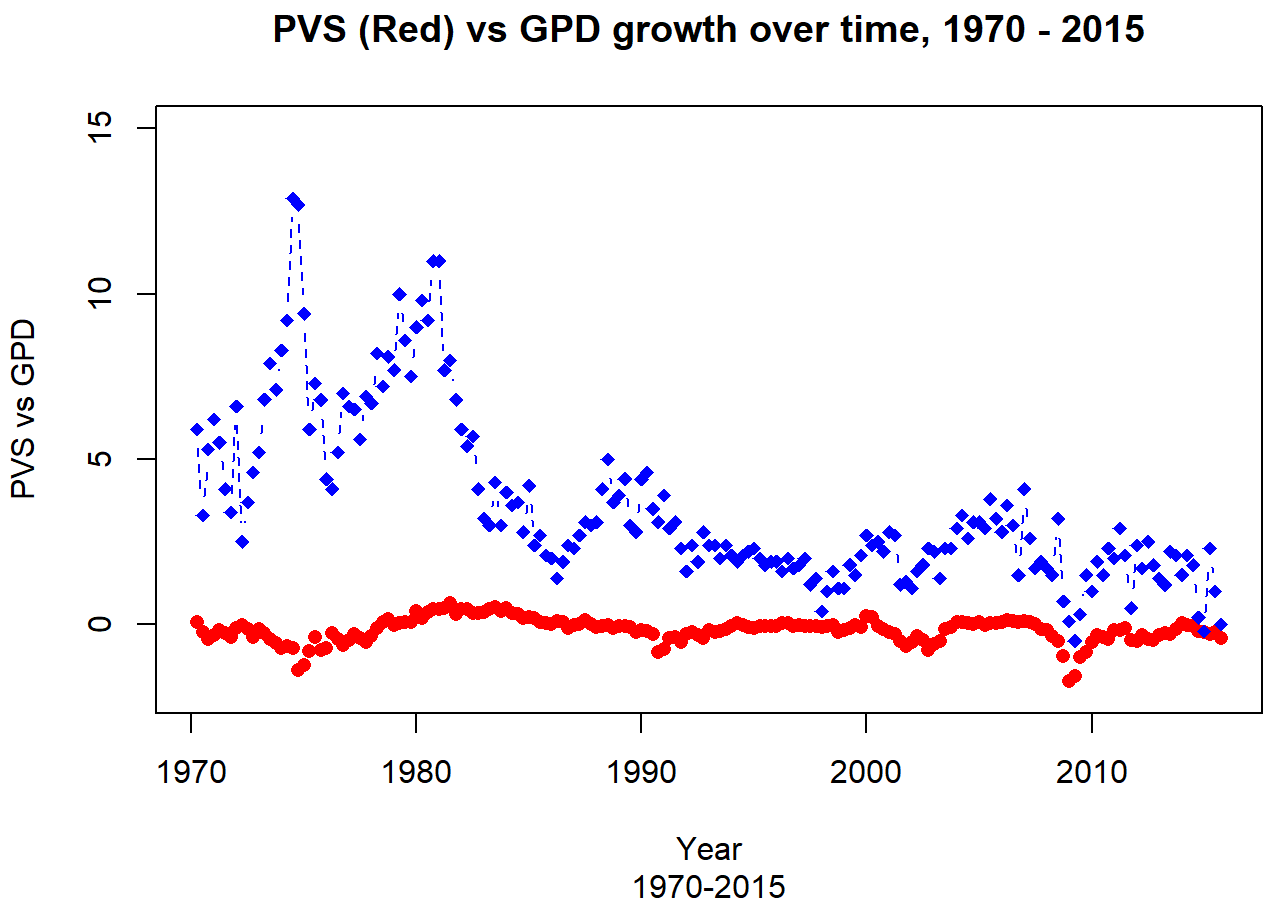

```
## NULL
```

```
#DPRatio vs GPDI over time
x = combined_data_DPRatio$Date
y1 = combined_data_DPRatio$DPRatio * 100
y2 = combined_data_DPRatio$Gross.private.domestic.investment
plot(x, y1, type="b", pch=16, col="red", main="DPRatio (red, scaled by 100) vs GPDI growth over
time, 1960 - 2015", sub="1960-2015",
  xlab="Year", ylab="DPRatio vs GPDI", ylim = c(-10, 20))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("DPRatio", "GPDI"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



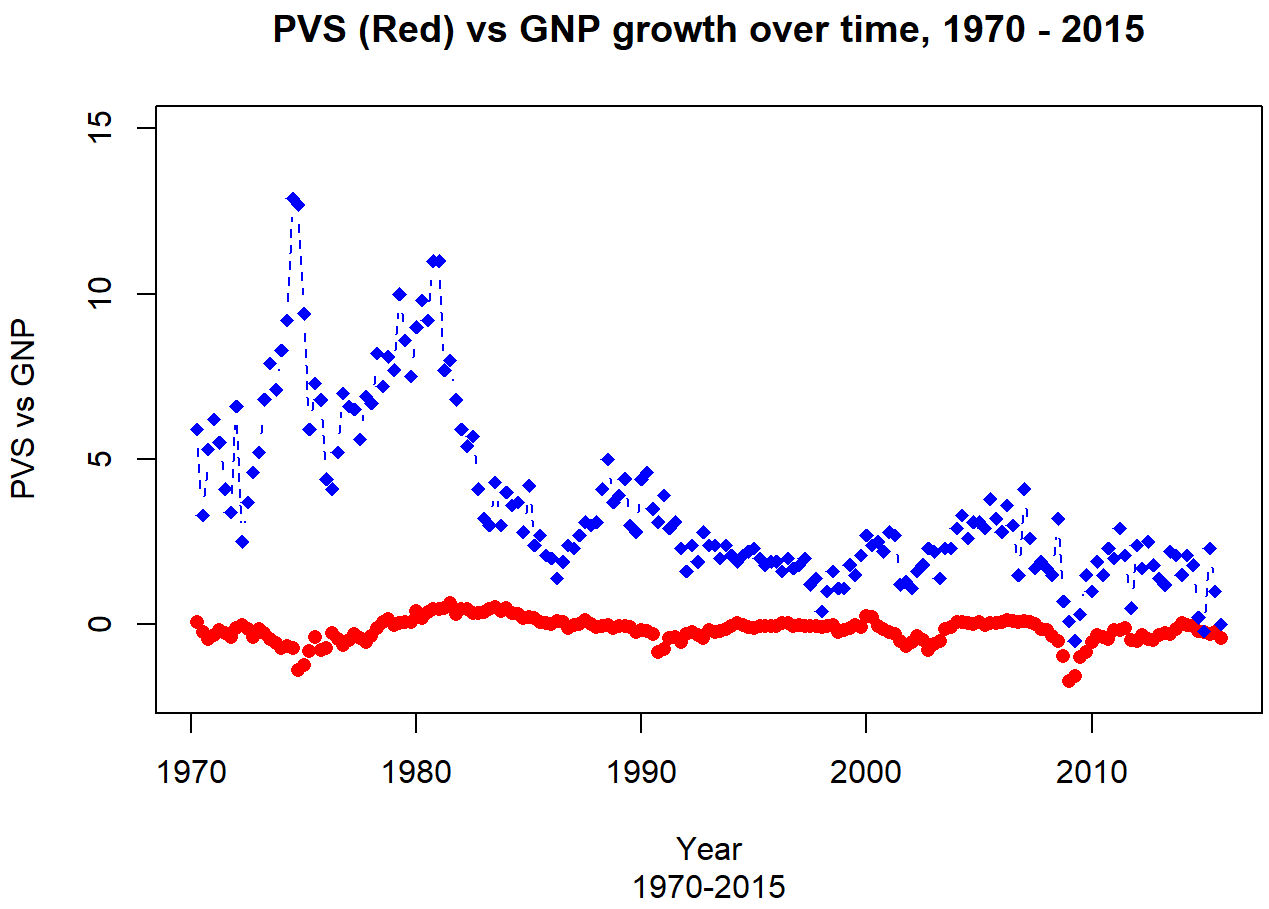

```
## NULL
```

```
#PVS vs GDP over time
x = combined_data_PVS$Date
y1 = combined_data_PVS$PVS
y2 = combined_data_PVS$GDP
plot(x, y1, type="b", pch=16, col="red", main="PVS (Red) vs GPD growth over time, 1970 - 2015",
sub="1970-2015",
  xlab="Year", ylab="PVS vs GPD", ylim = c(-2, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("PVS", "GPD"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



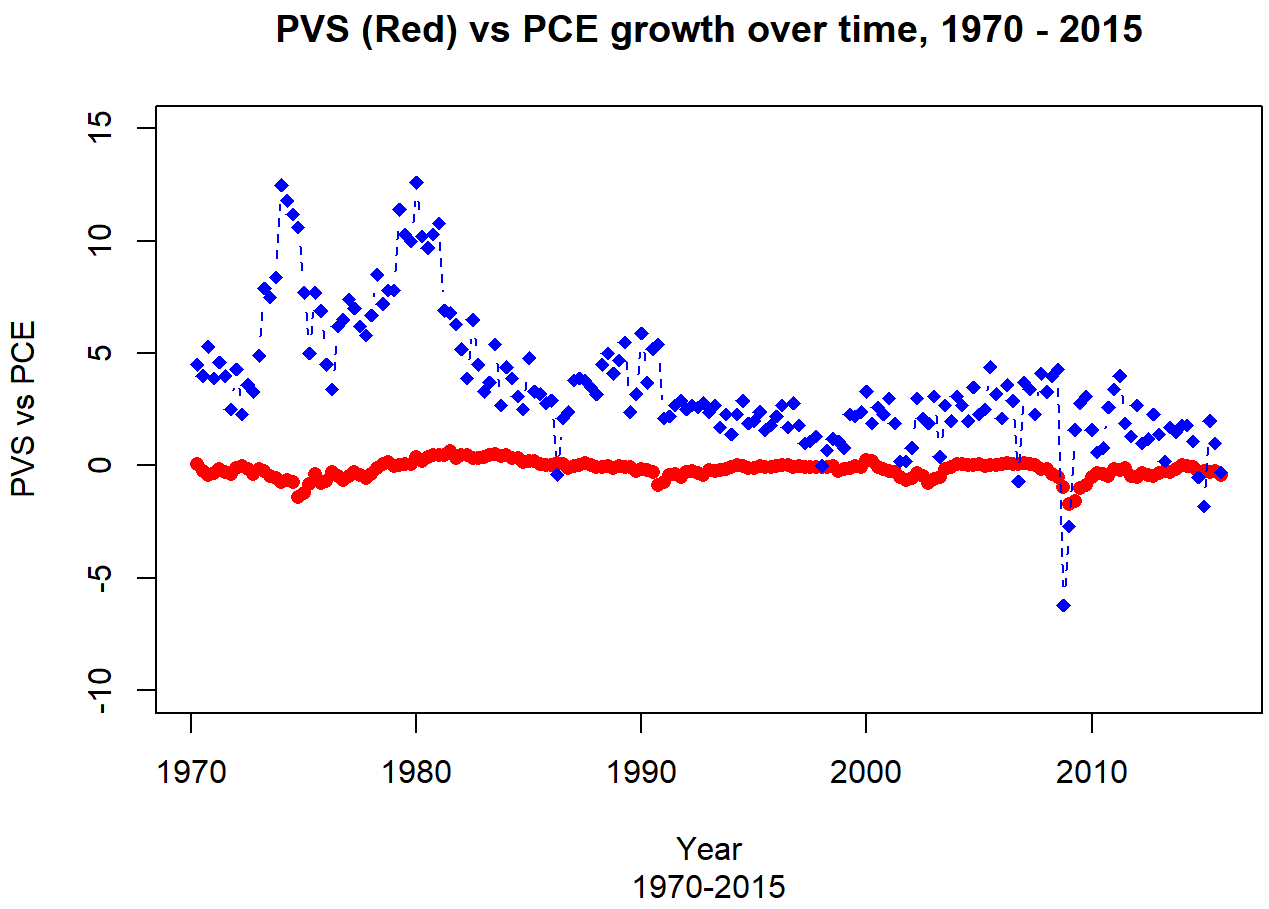

```
## NULL
```

```
#PVS vs GNP over time
x = combined_data_PVS$Date
y1 = combined_data_PVS$PVS
y2 = combined_data_PVS$GDP
plot(x, y1, type="b", pch=16, col="red", main="PVS (Red) vs GNP growth over time, 1970 - 2015",
sub="1970-2015",
  xlab="Year", ylab="PVS vs GNP", ylim = c(-2, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("PVS", "GNP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



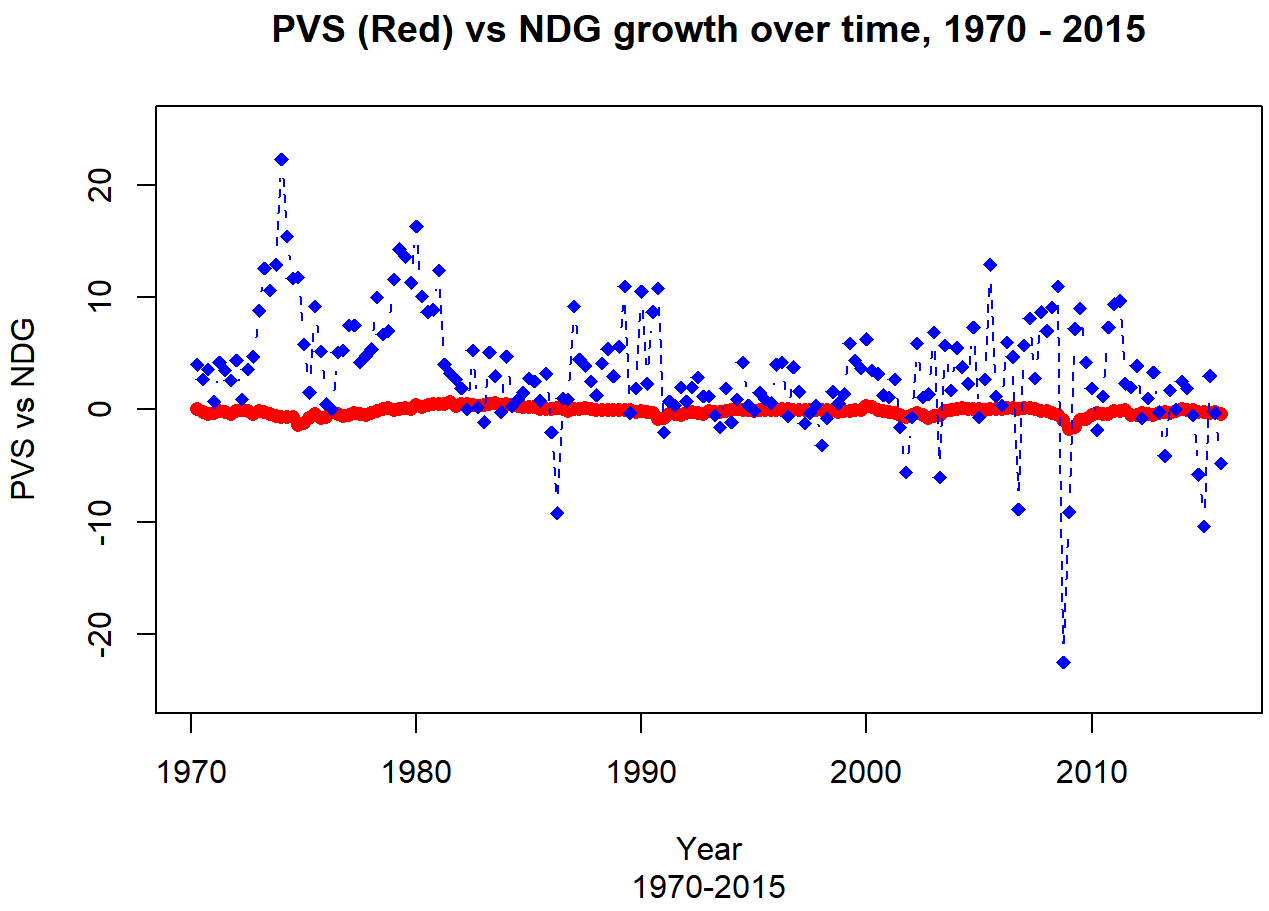

```
## NULL
```

```
#PVS vs PCE over time
x = combined_data_PVS$Date
y1 = combined_data_PVS$PVS
y2 = combined_data_PVS$Personal.consumption.expenditures
plot(x, y1, type="b", pch=16, col="red", main="PVS (Red) vs PCE growth over time, 1970 - 2015",
sub="1970-2015",
  xlab="Year", ylab="PVS vs PCE", ylim = c(-10, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("PVS", "PCE"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



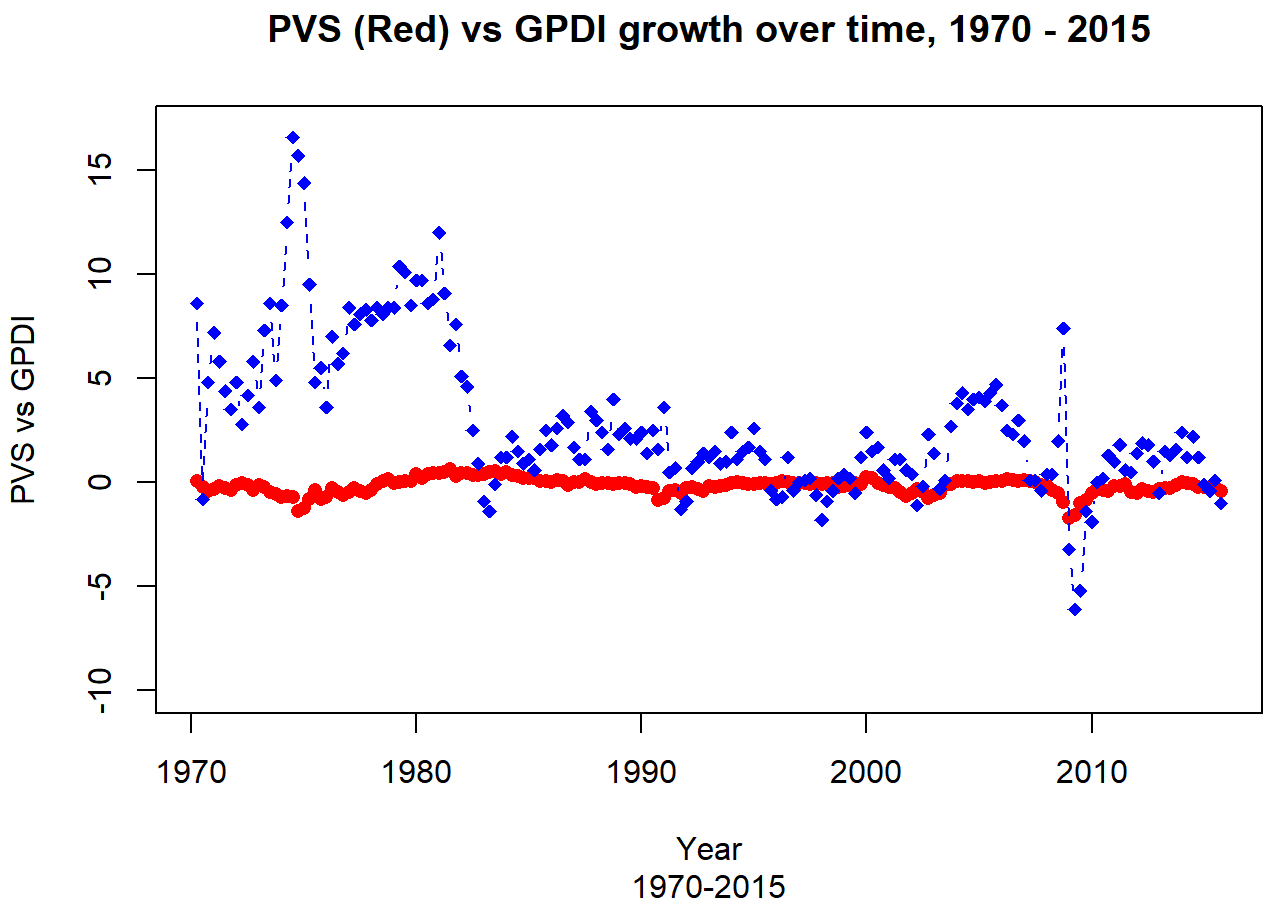

```
## NULL
```

```
#PVS vs NDG over time
x = combined_data_PVS$Date
y1 = combined_data_PVS$PVS
y2 = combined_data_PVS$Nondurable.goods
plot(x, y1, type="b", pch=16, col="red", main="PVS (Red) vs NDG growth over time, 1970 - 2015",
sub="1970-2015",
  xlab="Year", ylab="PVS vs NDG", ylim = c(-25, 25))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("PVS", "NDG"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



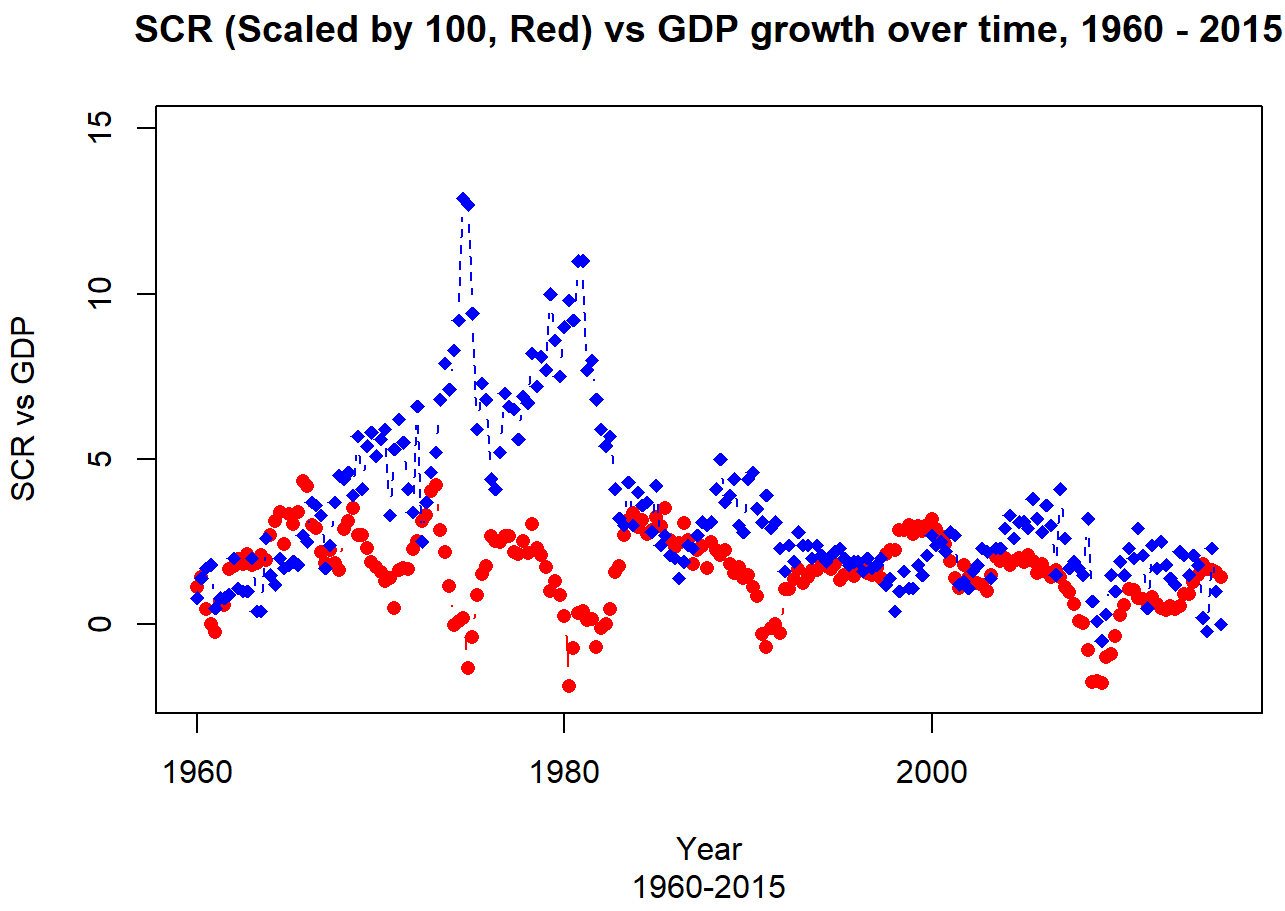

```
## NULL
```

```
#PVS vs GPDI over time
x = combined_data_PVS$Date
y1 = combined_data_PVS$PVS
y2 = combined_data_PVS$Gross.private.domestic.investment
plot(x, y1, type="b", pch=16, col="red", main="PVS (Red) vs GPDI growth over time, 1970 - 2015",
sub="1970-2015",
  xlab="Year", ylab="PVS vs GPDI", ylim = c(-10, 17))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("PVS", "GPDI"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



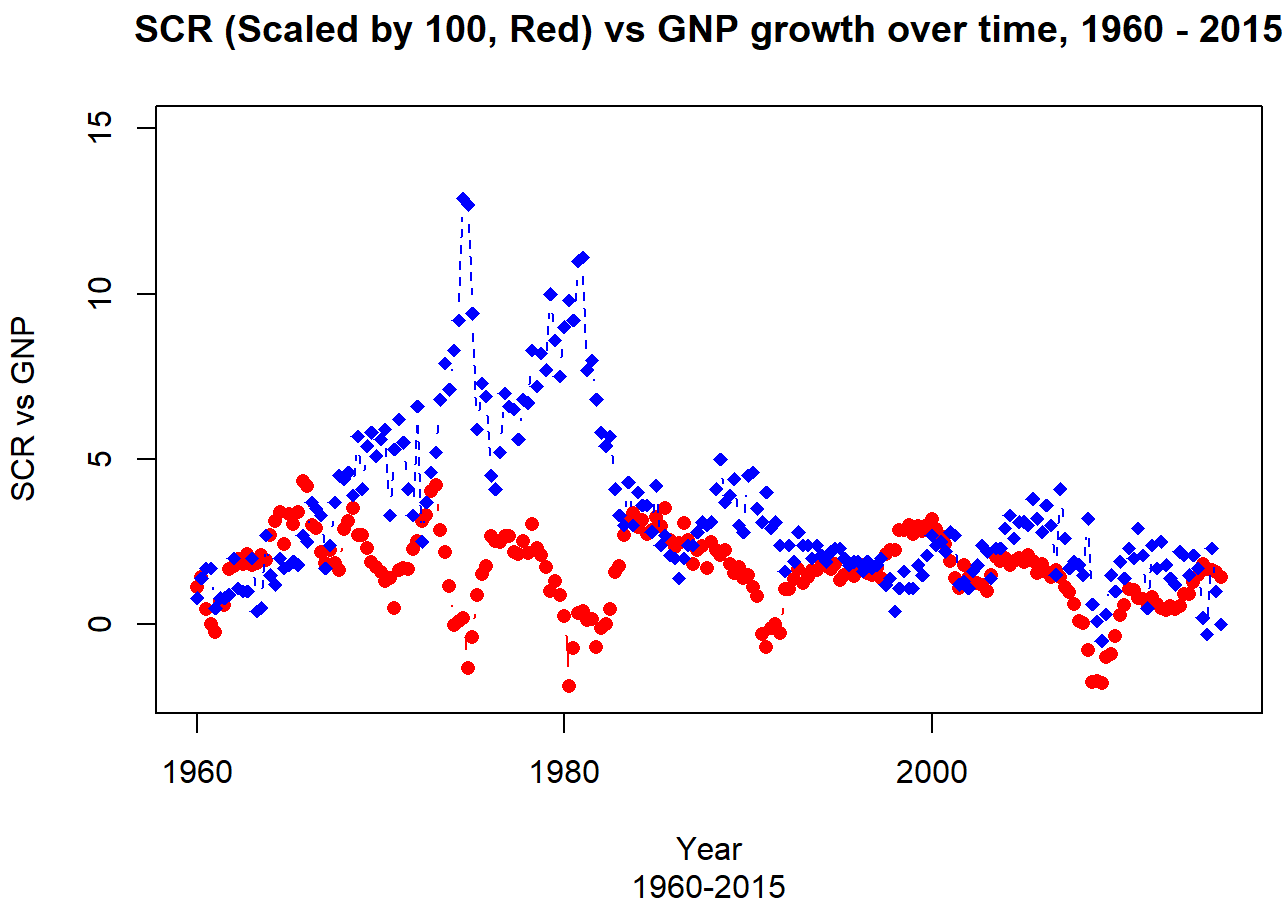

```
## NULL
```

```
#SCR vs GDP over time
x = combined_data_SCR$Date
y1 = combined_data_SCR$SCR * 100
y2 = combined_data_SCR$GDP
plot(x, y1, type="b", pch=16, col="red", main="SCR (Scaled by 100, Red) vs GDP growth over time,
1960 - 2015", sub="1960-2015",
  xlab="Year", ylab="SCR vs GDP", ylim = c(-2, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("SCR", "GDP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



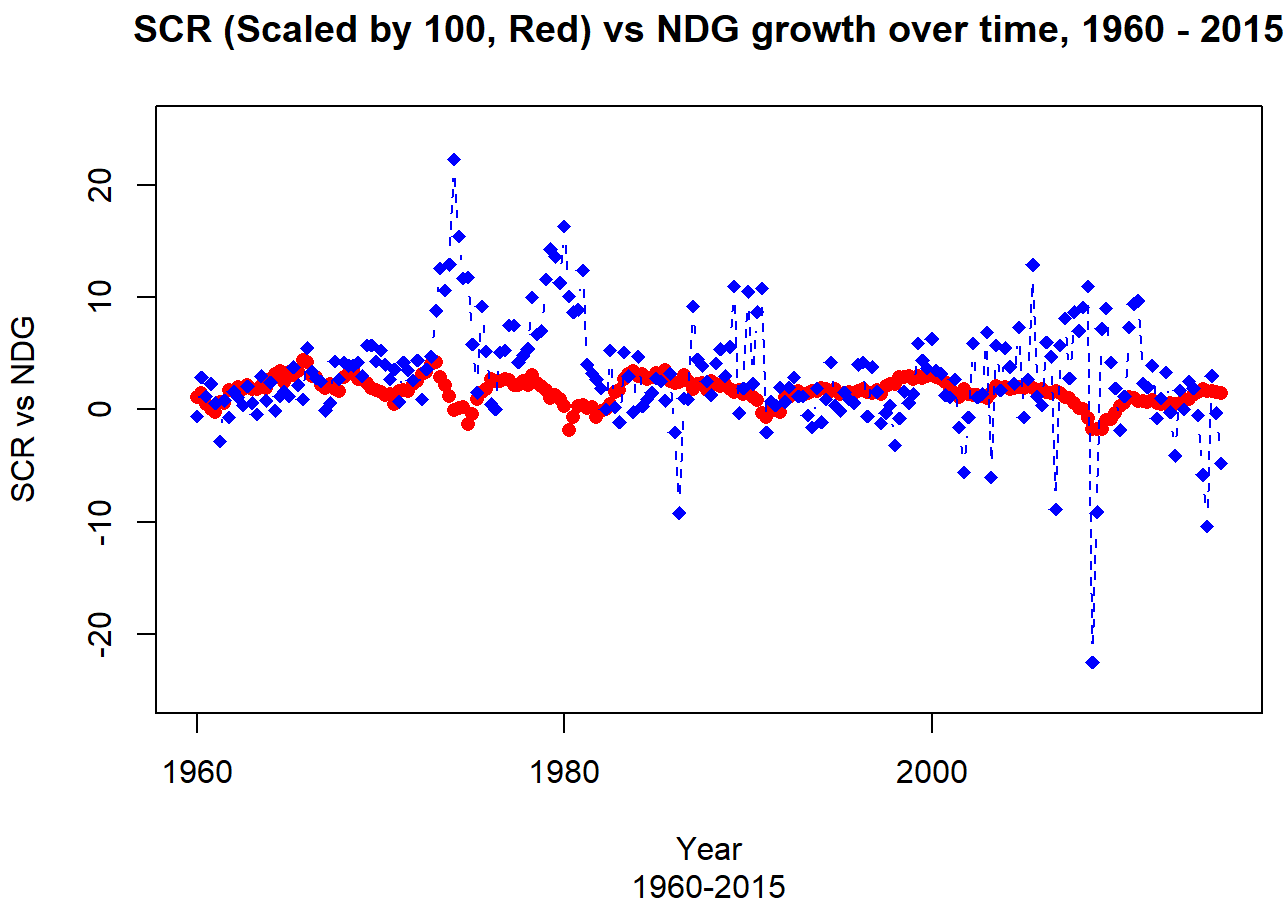

```
## NULL
```

```
#SCR vs GNP over time
x = combined_data_SCR$Date
y1 = combined_data_SCR$SCR * 100
y2 = combined_data_SCR$Gross.national.product
plot(x, y1, type="b", pch=16, col="red", main="SCR (Scaled by 100, Red) vs GNP growth over time,
1960 - 2015", sub="1960-2015",
  xlab="Year", ylab="SCR vs GNP", ylim = c(-2, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("SCR", "GNP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



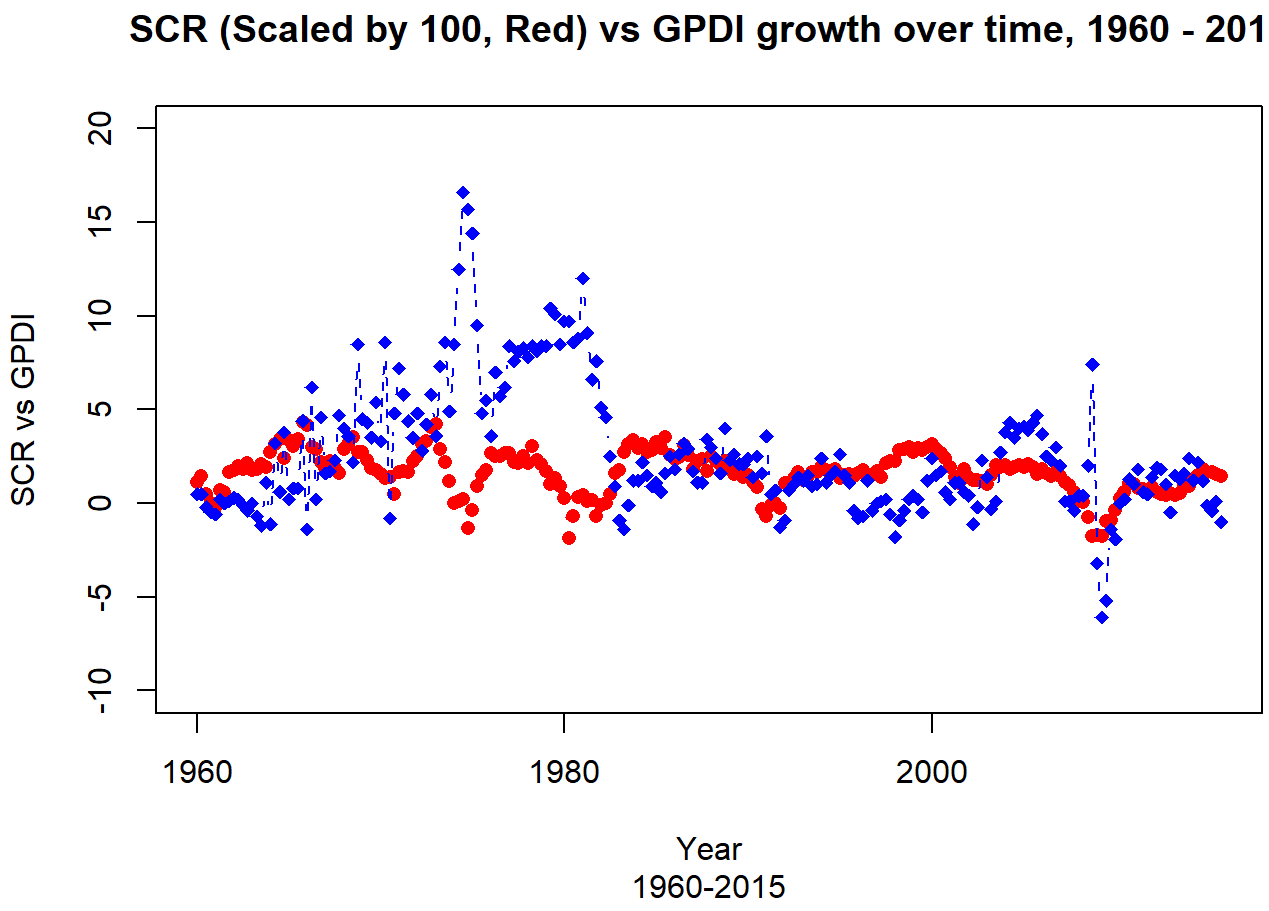

```
## NULL
```

```
#SCR vs NDG over time
x = combined_data_SCR$Date
y1 = combined_data_SCR$SCR * 100
y2 = combined_data_SCR$Nondurable.goods
plot(x, y1, type="b", pch=16, col="red", main="SCR (Scaled by 100, Red) vs NDG growth over time,
1960 - 2015", sub="1960-2015",
  xlab="Year", ylab="SCR vs NDG", ylim = c(-25, 25))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("SCR", "NDG"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



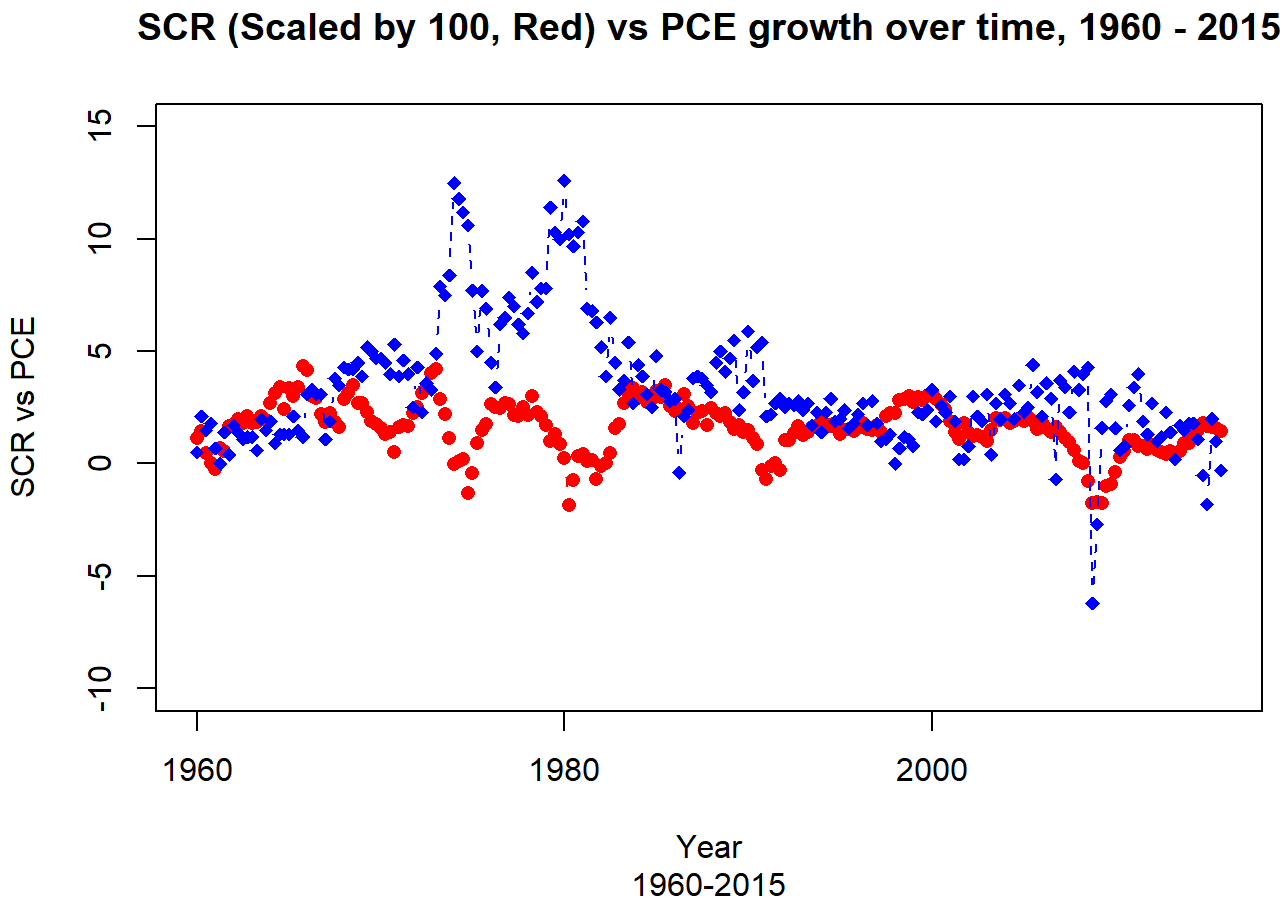

```
## NULL
```

```
#SCR vs GPDI over time
x = combined_data_SCR$Date
y1 = combined_data_SCR$SCR * 100
y2 = combined_data_SCR$Gross.private.domestic.investment
plot(x, y1, type="b", pch=16, col="red", main="SCR (Scaled by 100, Red) vs GPDI growth over tim
e, 1960 - 2015", sub="1960-2015",
  xlab="Year", ylab="SCR vs GPDI", ylim = c(-10, 20))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("SCR", "GPDI"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



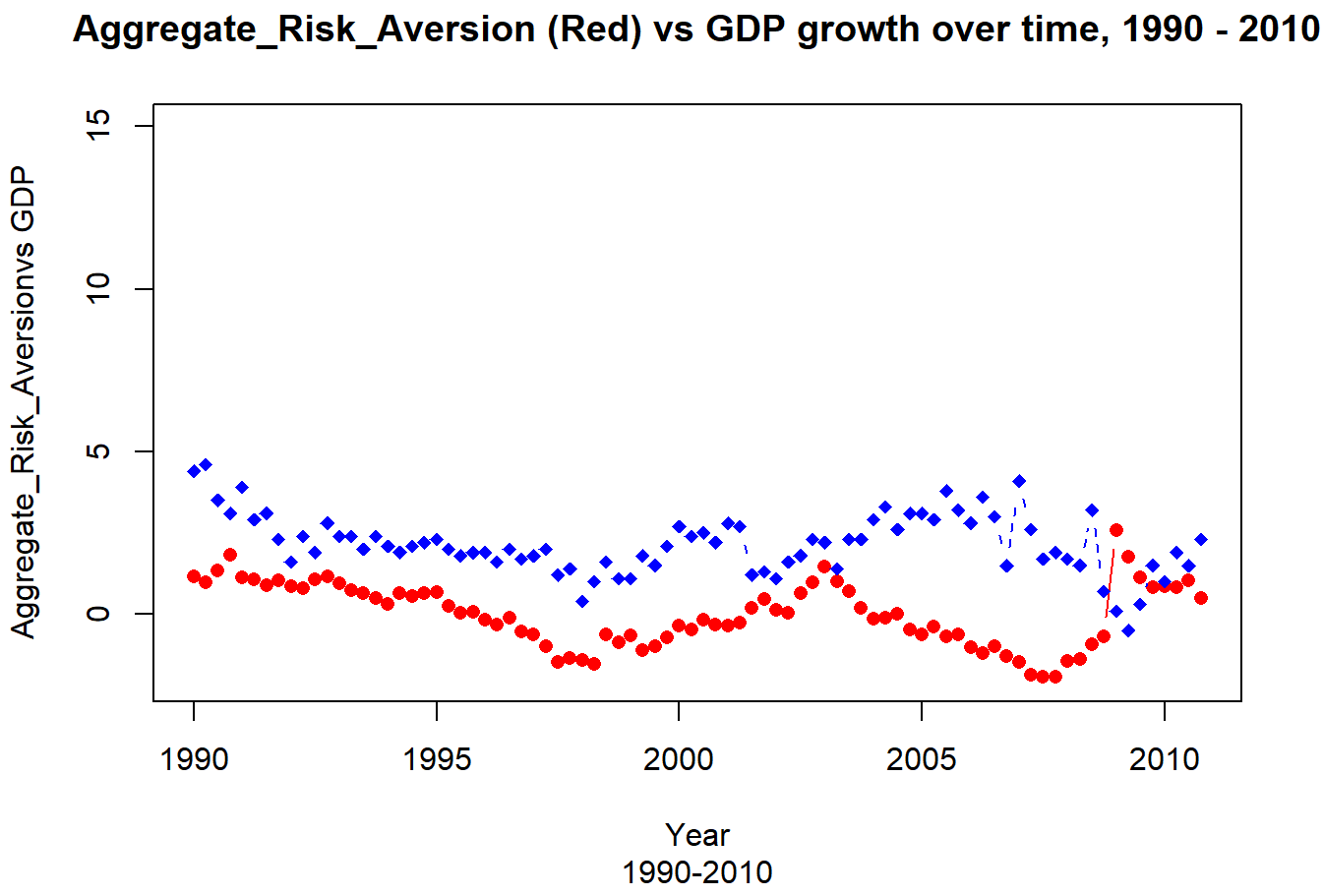

```
## NULL
```

```
#SCR vs GPDI over time
x = combined_data_SCR$Date
y1 = combined_data_SCR$SCR * 100
y2 = combined_data_SCR$Personal.consumption.expenditures
plot(x, y1, type="b", pch=16, col="red", main="SCR (Scaled by 100, Red) vs PCE growth over time,
1960 - 2015", sub="1960-2015",
  xlab="Year", ylab="SCR vs PCE", ylim = c(-10, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("SCR", "PCE"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



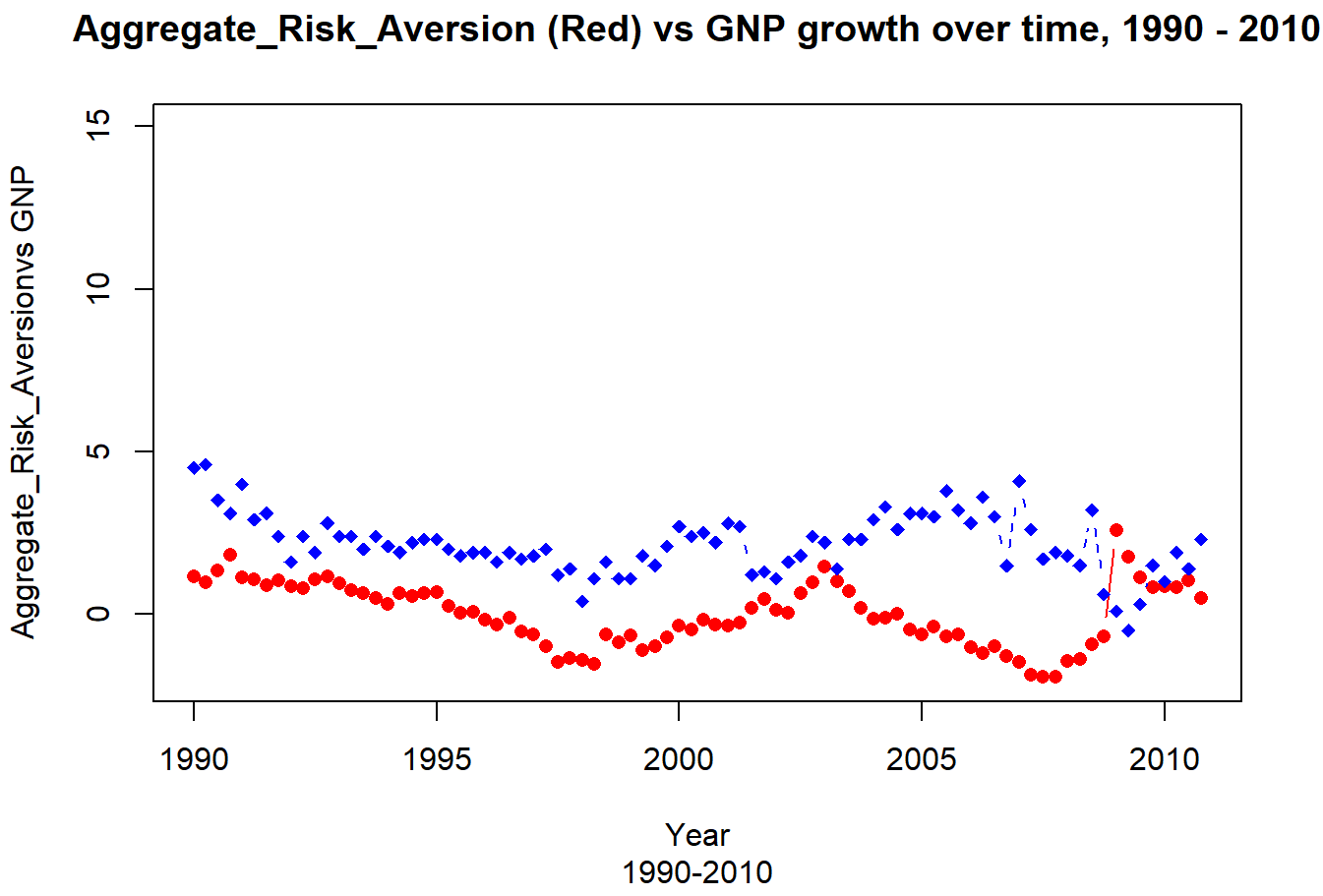

```
## NULL
```

```
#Aggregate_Risk_Aversion vs GDP over time
x = combined_data_Aggregate_Risk_Aversion$Date
y1 = combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion
y2 = combined_data_Aggregate_Risk_Aversion$GDP
plot(x, y1, type="b", pch=16, col="red", main="Aggregate_Risk_Aversion (Red) vs GDP growth over
time, 1990 - 2010", sub="1990-2010",
  xlab="Year", ylab="Aggregate_Risk_Aversionvs GDP", ylim = c(-2, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("Aggregate_Risk_Aversion", "GDP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



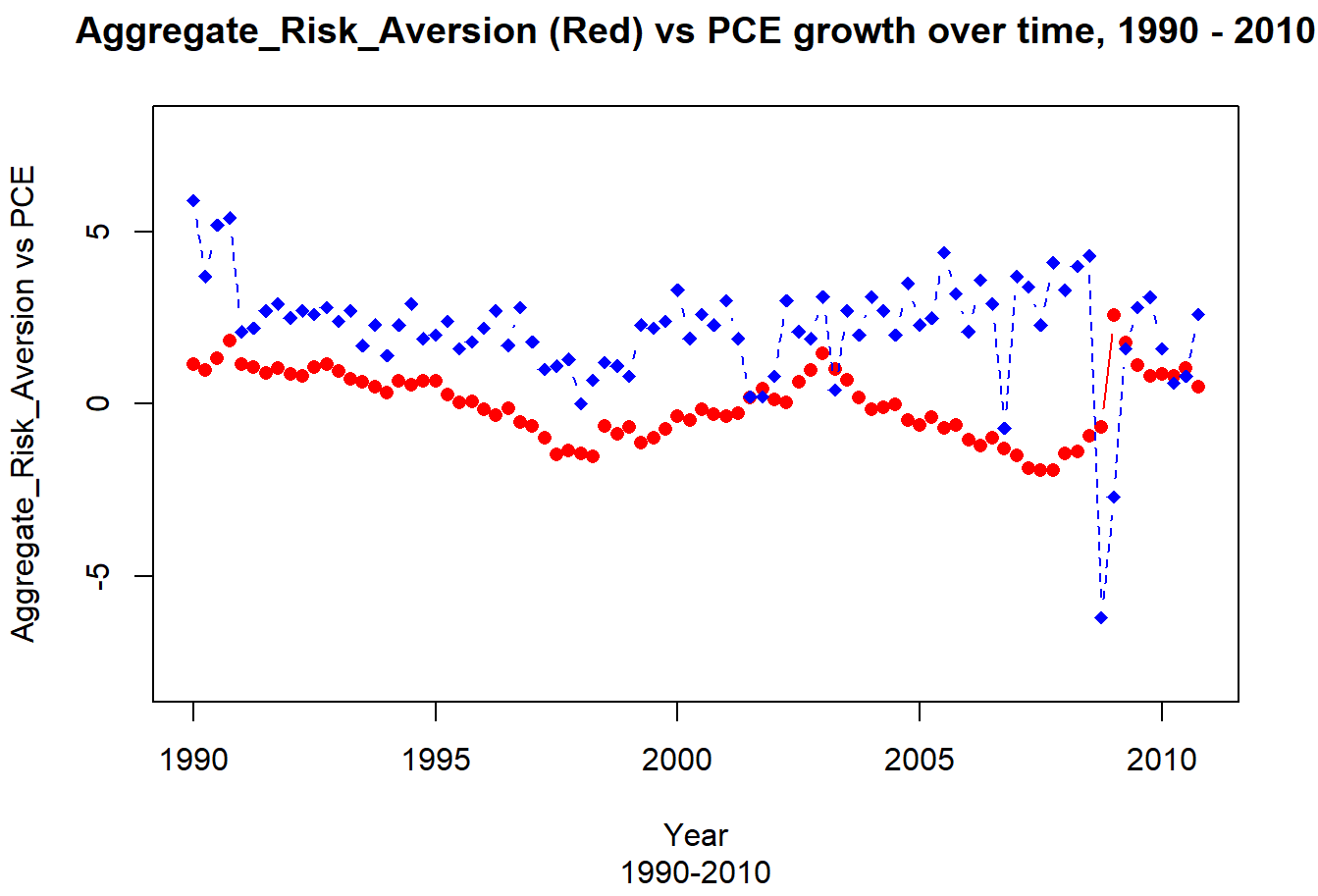

```
## NULL
```

```
#Aggregate_Risk_Aversion vs GNP over time
x = combined_data_Aggregate_Risk_Aversion$Date
y1 = combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion
y2 = combined_data_Aggregate_Risk_Aversion$Gross.national.product
plot(x, y1, type="b", pch=16, col="red", main="Aggregate_Risk_Aversion (Red) vs GNP growth over
time, 1990 - 2010", sub="1990-2010",
  xlab="Year", ylab="Aggregate_Risk_Aversionvs GNP", ylim = c(-2, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("Aggregate_Risk_Aversion", "GNP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



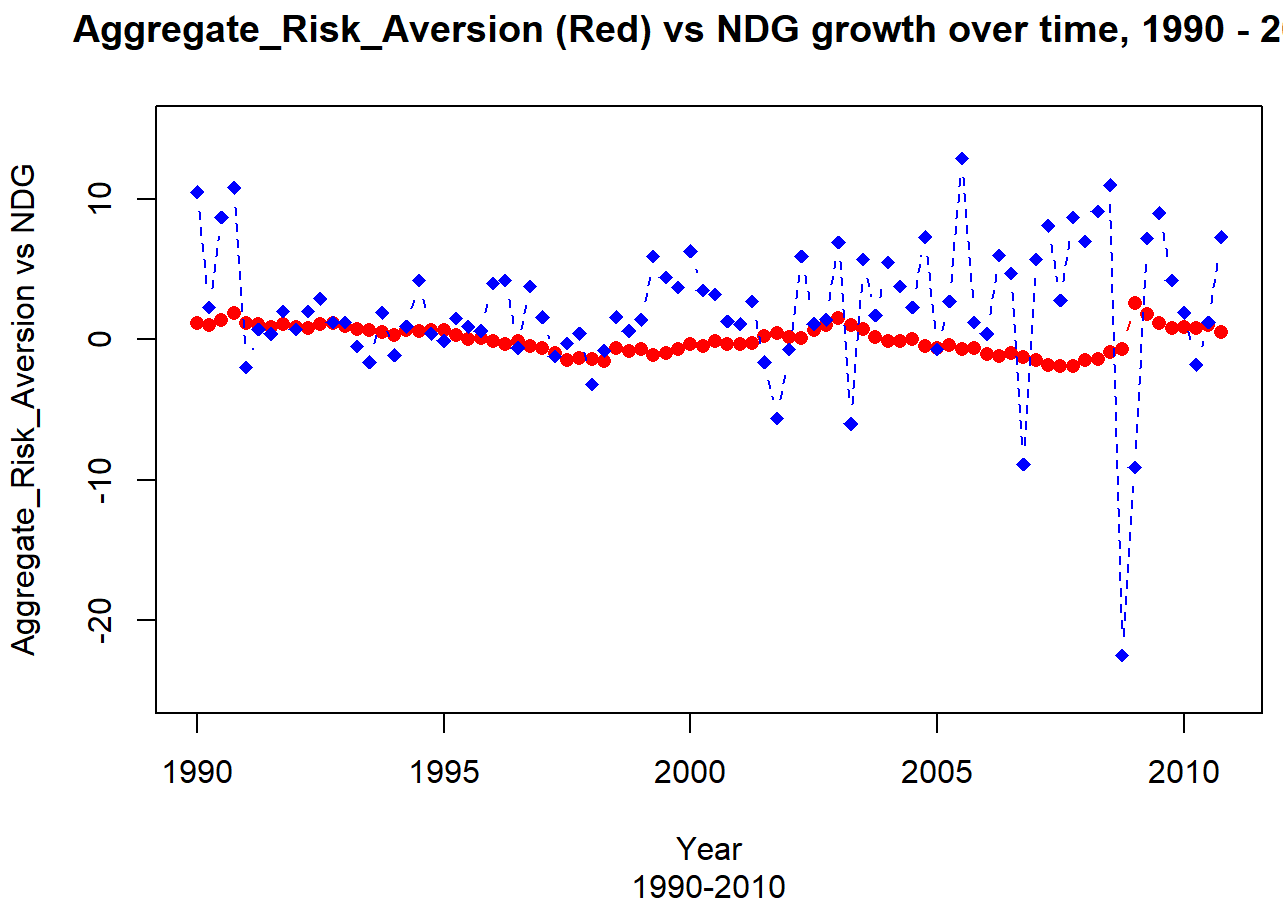

```
## NULL
```

```
#Aggregate_Risk_Aversion vs PCE over time
x = combined_data_Aggregate_Risk_Aversion$Date
y1 = combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion
y2 = combined_data_Aggregate_Risk_Aversion$Personal.consumption.expenditures
plot(x, y1, type="b", pch=16, col="red", main="Aggregate_Risk_Aversion (Red) vs PCE growth over
time, 1990 - 2010", sub="1990-2010",
  xlab="Year", ylab="Aggregate_Risk_Aversion vs PCE", ylim = c(-8, 8))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("Aggregate_Risk_Aversion", "PCE"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



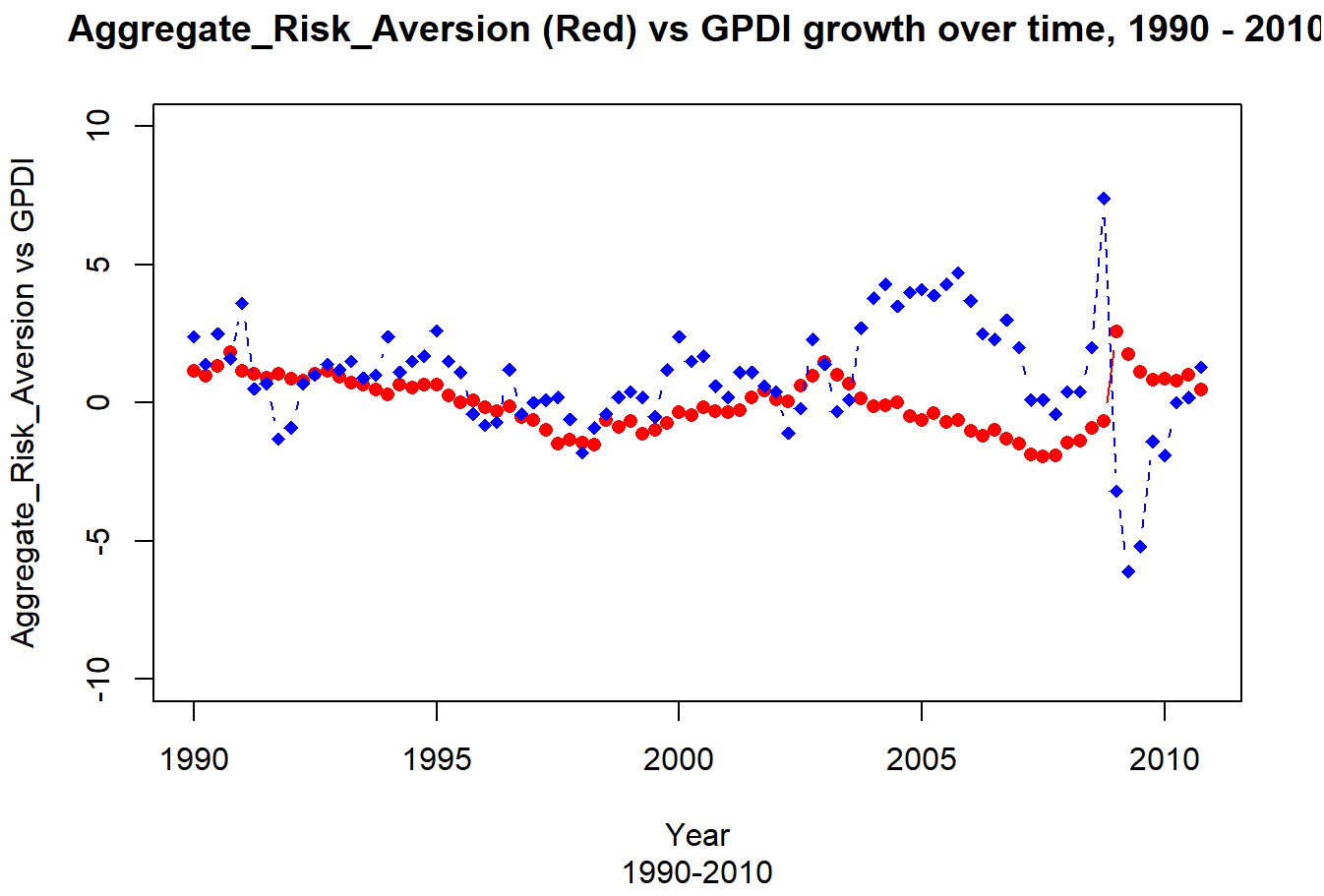

```
## NULL
```

```
#Aggregate_Risk_Aversion vs NDG over time
x = combined_data_Aggregate_Risk_Aversion$Date
y1 = combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion
y2 = combined_data_Aggregate_Risk_Aversion$Nondurable.goods
plot(x, y1, type="b", pch=16, col="red", main="Aggregate_Risk_Aversion (Red) vs NDG growth over
time, 1990 - 2010", sub="1990-2010",
  xlab="Year", ylab="Aggregate_Risk_Aversion vs NDG", ylim = c(-25, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("Aggregate_Risk_Aversion", "NDG"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



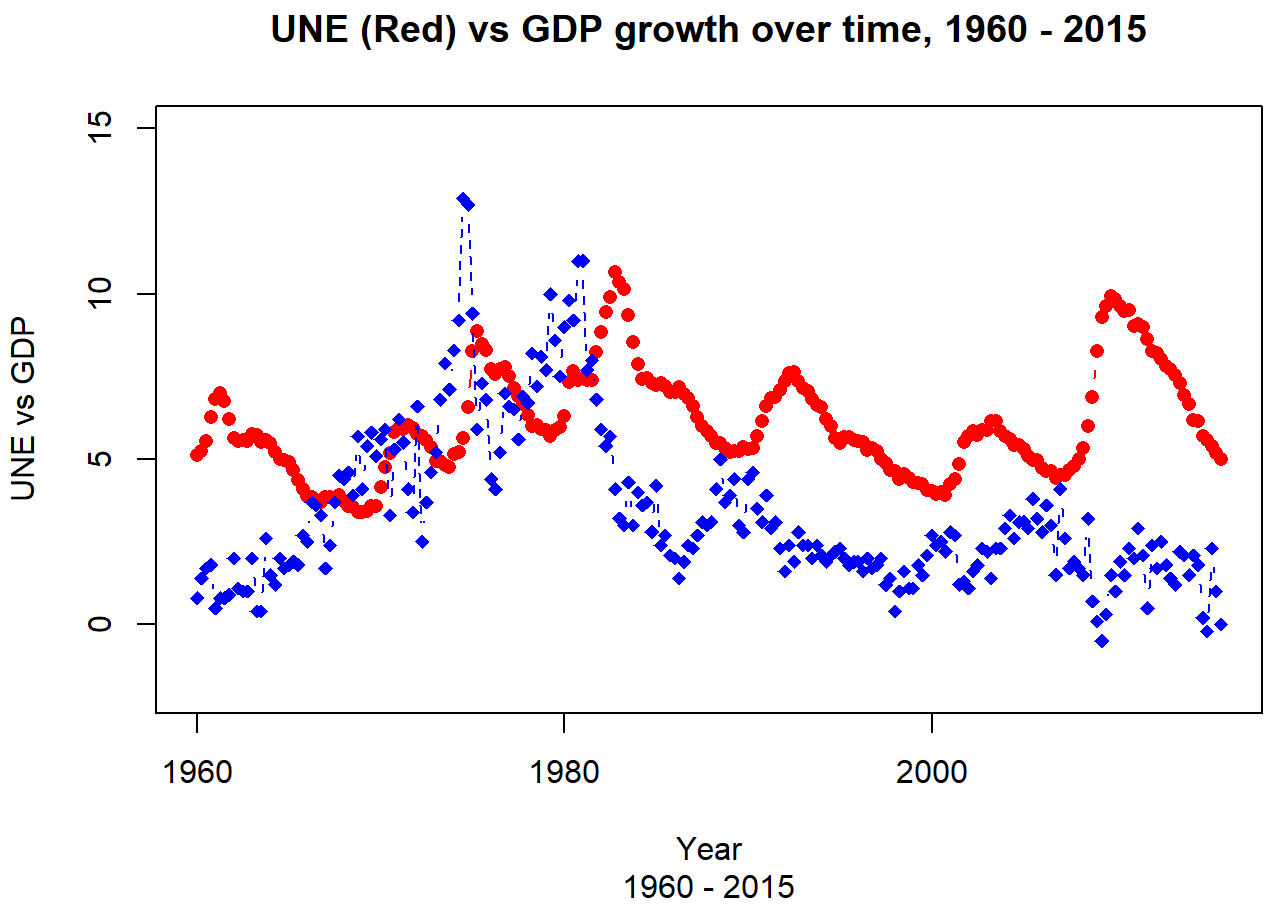

```
## NULL
```

```
#Aggregate_Risk_Aversion vs GPDI over time
x = combined_data_Aggregate_Risk_Aversion$Date
y1 = combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion
y2 = combined_data_Aggregate_Risk_Aversion$Gross.private.domestic.investment
plot(x, y1, type="b", pch=16, col="red", main="Aggregate_Risk_Aversion (Red) vs GPDI growth over
time, 1990 - 2010", sub="1990-2010",
  xlab="Year", ylab="Aggregate_Risk_Aversion vs GPDI", ylim = c(-10, 10))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("Aggregate_Risk_Aversion", "GPDI"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



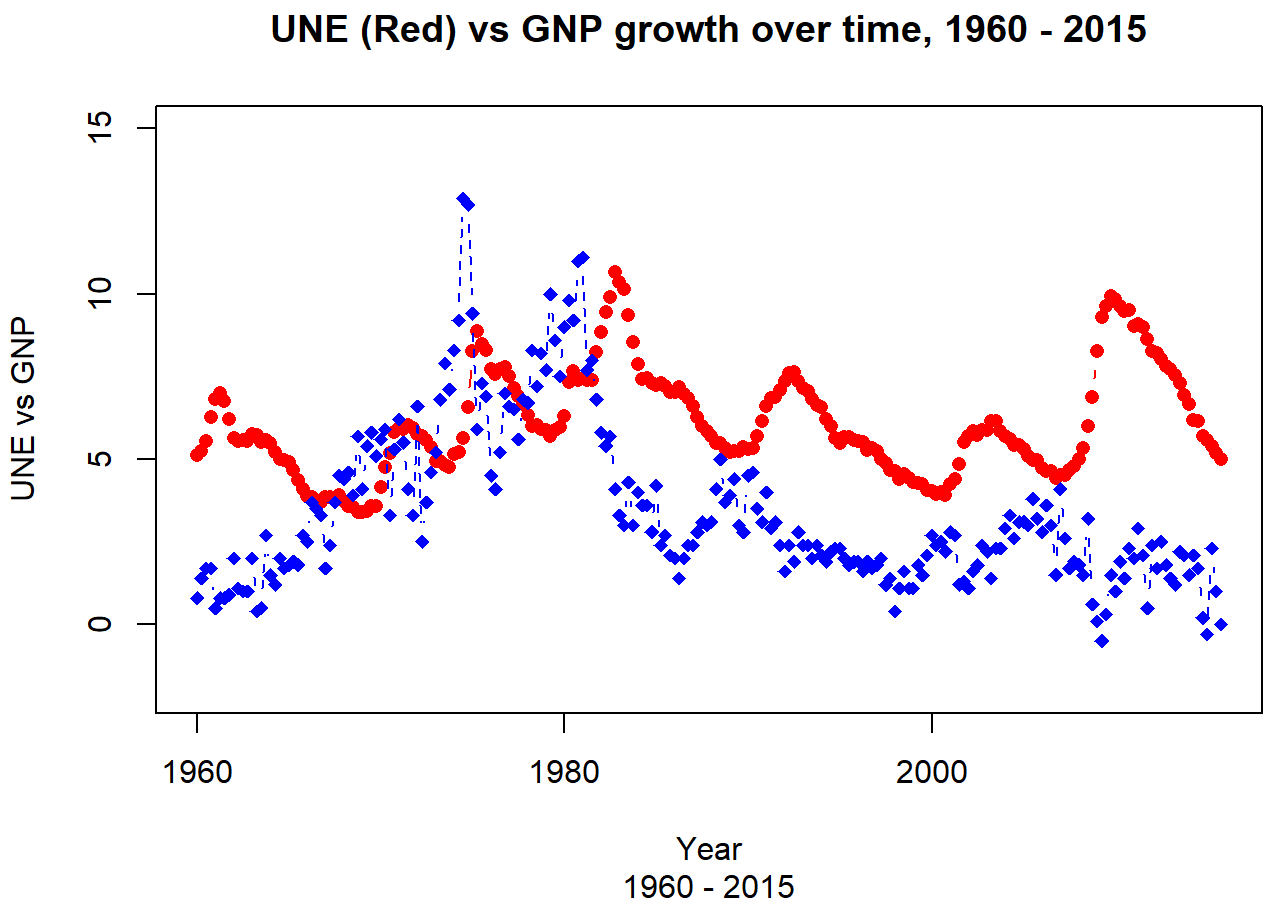

```
## NULL
```

```
#UNE vs GDP over time
x = combined_data_UNE$Date
y1 = combined_data_UNE$UNE
y2 = combined_data_UNE$GDP
plot(x, y1, type="b", pch=16, col="red", main="UNE (Red) vs GDP growth over time, 1960 - 2015",
sub="1960 - 2015",
  xlab="Year", ylab="UNE vs GDP", ylim = c(-2, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("UNE", "GDP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



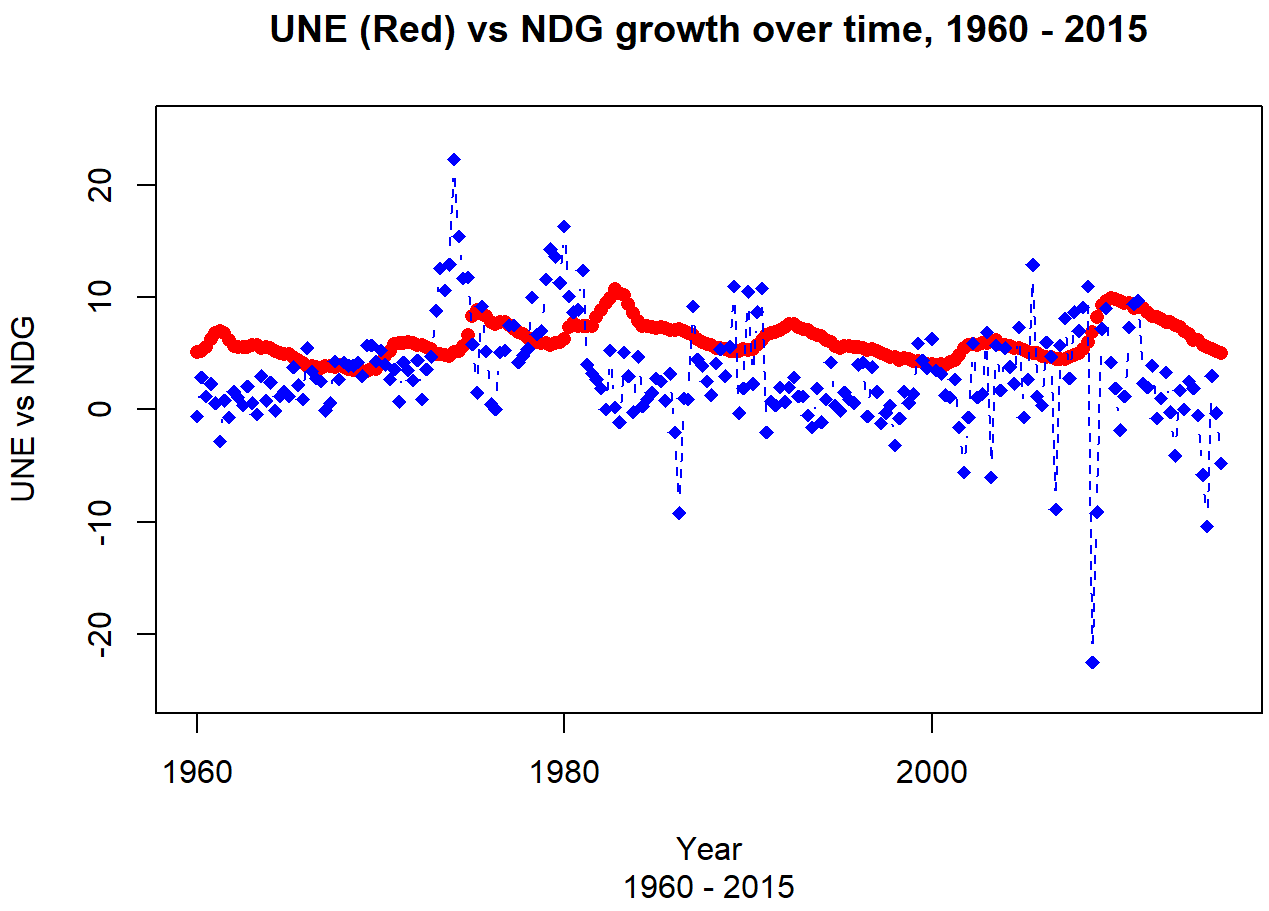

```
## NULL
```

```
#UNE vs GNP over time
x = combined_data_UNE$Date
y1 = combined_data_UNE$UNE
y2 = combined_data_UNE$Gross.national.product
plot(x, y1, type="b", pch=16, col="red", main="UNE (Red) vs GNP growth over time, 1960 - 2015",
sub="1960 - 2015",
  xlab="Year", ylab="UNE vs GNP", ylim = c(-2, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("UNE", "GNP"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



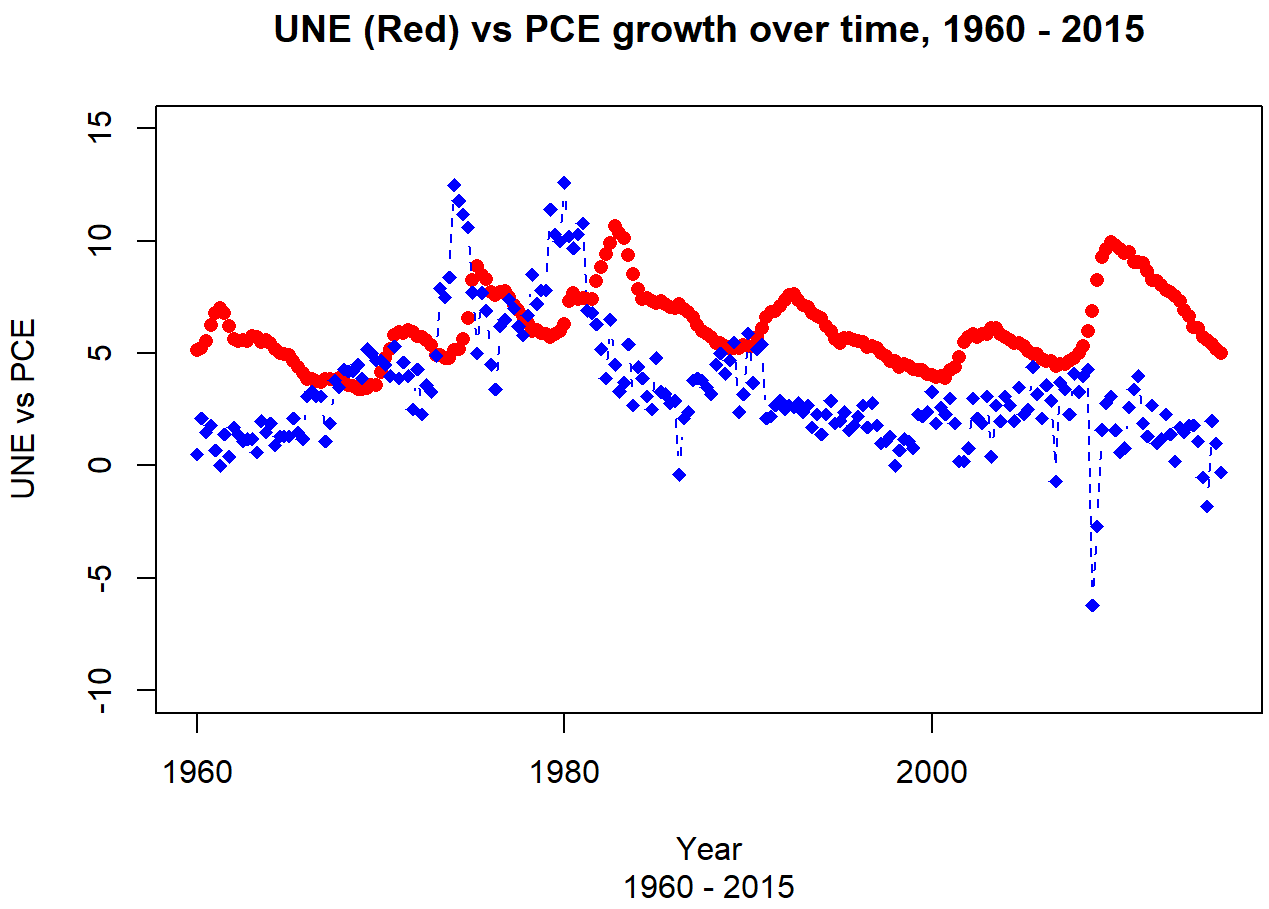

```
## NULL
```

```
#UNE vs NDG over time
x = combined_data_UNE$Date
y1 = combined_data_UNE$UNE
y2 = combined_data_UNE$Nondurable.goods
plot(x, y1, type="b", pch=16, col="red", main="UNE (Red) vs NDG growth over time, 1960 - 2015",
sub="1960 - 2015",
  xlab="Year", ylab="UNE vs NDG", ylim = c(-25, 25))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("UNE", "NDG"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```



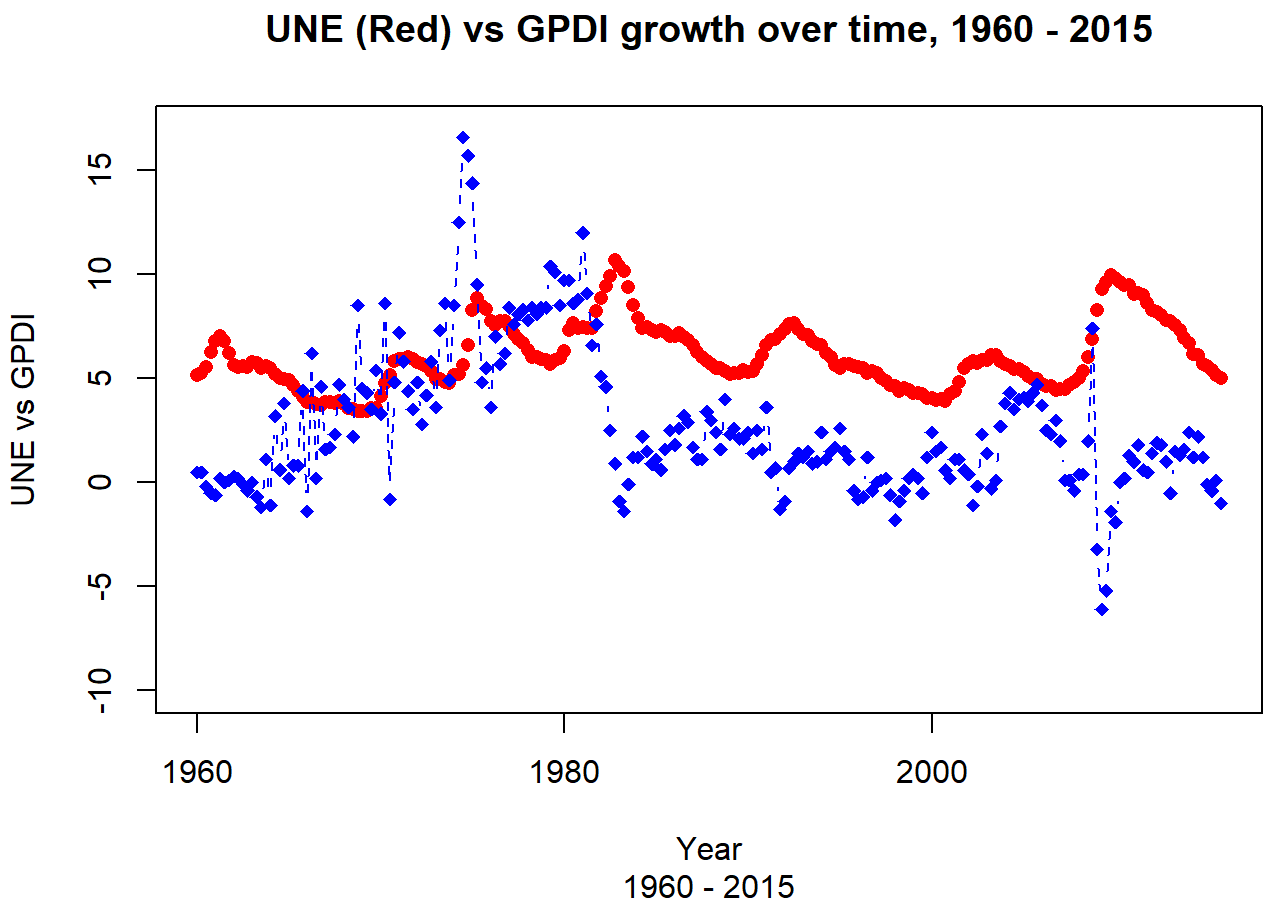

```
## NULL
```

```
#UNE vs PCE over time
x = combined_data_UNE$Date
y1 = combined_data_UNE$UNE
y2 = combined_data_UNE$Personal.consumption.expenditures
plot(x, y1, type="b", pch=16, col="red", main="UNE (Red) vs PCE growth over time, 1960 - 2015",
sub="1960 - 2015",
  xlab="Year", ylab="UNE vs PCE", ylim = c(-10, 15))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("UNE", "PCE"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```




```
## NULL
```

```
#UNE vs GPDI over time
x = combined_data_UNE$Date
y1 = combined_data_UNE$UNE
y2 = combined_data_UNE$Gross.private.domestic.investment
plot(x, y1, type="b", pch=16, col="red", main="UNE (Red) vs GPDI growth over time, 1960 - 2015",
sub="1960 - 2015",
  xlab="Year", ylab="UNE vs GPDI", ylim = c(-10, 17))
# Add lines for gdp measures
lines(x, y2, pch=18, col="blue", type="b", lty=2)
legend(-3800, 97, legend=c("UNE", "GPDI"),
       col=c("red", "blue"), lty=1:1, cex=0.8) +
theme(legend.position="top")
```




```
## NULL
```

```
#install package needed for lead/lag correlation
#if(!require('YRmisc')) {
 # install.packages('YRmisc')
  library('YRmisc')
```

```
## Warning: package 'YRmisc' was built under R version 4.3.3
```

```
#}
```

```
#Lead lag with gross ipo, lead/lag by 3 quarters
cor.lag(as.numeric(combined_data_IPO_Gross$IPO_Gross), as.numeric(combined_data_IPO_Gross$GDP),
3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3          0      lead1      lead2      lead3
## 1 -0.1915996 -0.2082586 -0.2086917 -0.1934336 -0.1890744 -0.1724763 -0.1507311
```

```
cor.lag(as.numeric(combined_data_IPO_Gross$IPO_Gross), as.numeric(combined_data_IPO_Gross$Gross.
national.product), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1      lag2       lag3          0      lead1      lead2      lead3
## 1 -0.1910768 -0.207397 -0.2088627 -0.1925319 -0.1881249 -0.1720035 -0.1506846
```

```
cor.lag(as.numeric(combined_data_IPO_Gross$IPO_Gross), as.numeric(combined_data_IPO_Gross$Gross.
private.domestic.investment), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3          0      lead1      lead2      lead3
## 1 -0.2341118 -0.2177402 -0.2466177 -0.2314912 -0.2396192 -0.2261404 -0.2179579
```

```
cor.lag(as.numeric(combined_data_IPO_Gross$IPO_Gross), as.numeric(combined_data_IPO_Gross$Nondur
able.goods), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3          0      lead1      lead2      lead3
## 1 -0.1473876 -0.1666723 -0.1525197 -0.1859671 -0.2240486 -0.2161688 -0.2137333
```

```
cor.lag(as.numeric(combined_data_IPO_Gross$IPO_Gross), as.numeric(combined_data_IPO_Gross$Person
al.consumption.expenditures), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3          0      lead1      lead2      lead3
## 1 -0.1717254 -0.1862748 -0.1880408 -0.1769917 -0.1835999 -0.1612384 -0.1366225
```

```
#Lead lag with EPBound, lead/lag by 3 quarters
cor.lag(as.numeric(combined_data_EPBound$EPBound), as.numeric(combined_data_EPBound$GDP), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3          0      lead1      lead2      lead3
## 1 -0.6008118 -0.5356111 -0.3746659 -0.5019246 -0.1898686 -0.2240599 -0.2490747
```

```
cor.lag(as.numeric(combined_data_EPBound$EPBound), as.numeric(combined_data_EPBound$Gross.nation
al.product), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3          0      lead1      lead2    lead3
## 1 -0.5975271 -0.5326034 -0.3766593 -0.5089468 -0.1922119 -0.2241491 -0.23777
```

```
cor.lag(as.numeric(combined_data_EPBound$EPBound), as.numeric(combined_data_EPBound$Gross.private.domestic.investment), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2      lag3           0       lead1      lead2      lead3
## 1 -0.5291134 -0.6309165 -0.461383 -0.05010437 -0.05734335 -0.1579413 -0.2163182
```

```
cor.lag(as.numeric(combined_data_EPBound$EPBound), as.numeric(combined_data_EPBound$Nondurable.goods), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1      lag2       lag3          0      lead1      lead2     lead3
## 1 -0.2307655 0.1355158 0.04481309 -0.5877241 -0.1288163 0.07244052 0.1138693
```

```
cor.lag(as.numeric(combined_data_EPBound$EPBound), as.numeric(combined_data_EPBound$Personal.consumption.expenditures), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1        lag2        lag3          0      lead1      lead2
## 1 -0.4143642 -0.06253091 -0.04163791 -0.7171297 -0.2219239 0.01205916
##        lead3
## 1 0.06630444
```

```
#lead lag with DPRatio, lead/lag by 3 quarters
cor.lag(as.numeric(combined_data_DPRatio$DPRatio), as.numeric(combined_data_DPRatio$GDP), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2     lag3         0     lead1     lead2     lead3
## 1 0.6313779 0.5980311 0.569702 0.6633133 0.6844568 0.6971862 0.7032893
```

```
cor.lag(as.numeric(combined_data_DPRatio$DPRatio), as.numeric(combined_data_DPRatio$Gross.nation
al.product), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3         0     lead1     lead2     lead3
## 1 0.6315299 0.5985946 0.5705854 0.6628789 0.6837174 0.6969684 0.7036168
```

```
cor.lag(as.numeric(combined_data_DPRatio$DPRatio), as.numeric(combined_data_DPRatio$Gross.private.domestic.investment), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3        0     lead1    lead2     lead3
## 1 0.5016373 0.4560888 0.4277053 0.554113 0.5899575 0.600707 0.5974026
```

```
cor.lag(as.numeric(combined_data_DPRatio$DPRatio), as.numeric(combined_data_DPRatio$Nondurable.goods), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3        0     lead1     lead2     lead3
## 1 0.2720982 0.2547042 0.2331799 0.280237 0.3048756 0.3552881 0.3950233
```

```
cor.lag(as.numeric(combined_data_DPRatio$DPRatio), as.numeric(combined_data_DPRatio$Personal.con
sumption.expenditures), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3         0    lead1     lead2     lead3
## 1 0.5992599 0.5777513 0.5554542 0.6165728 0.638982 0.6735583 0.6902611
```

```
#Lead lag with Surplus Consumption Ratio, lead/lag by 3 quarters
cor.lag(as.numeric(combined_data_SCR$SCR), as.numeric(combined_data_SCR$GDP),3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1        lag2       lag3          0      lead1      lead2
## 1 -0.05372426 -0.01278721 0.03998134 -0.1173753 -0.1610677 -0.1708354
##        lead3
## 1 -0.1867122
```

```
cor.lag(as.numeric(combined_data_SCR$SCR), as.numeric(combined_data_SCR$Gross.national.product),
3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1        lag2      lag3          0      lead1      lead2      lead3
## 1 -0.05328563 -0.01259291 0.0393751 -0.1164908 -0.1598516 -0.1703309 -0.1864012
```

```
cor.lag(as.numeric(combined_data_SCR$SCR), as.numeric(combined_data_SCR$Gross.private.domestic.i
nvestment), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##            lag1       lag2      lag3           0      lead1      lead2
## 1 -0.0001540435 0.05852187 0.1090726 -0.08171335 -0.1102305 -0.1569363
##        lead3
## 1 -0.1515413
```

```
cor.lag(as.numeric(combined_data_SCR$SCR), as.numeric(combined_data_SCR$Nondurable.goods), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3           0      lead1      lead2      lead3
## 1 0.03553566 0.06593914 0.08480896 -0.06300038 -0.1641529 -0.2077114 -0.2872305
```

```
cor.lag(as.numeric(combined_data_SCR$SCR), as.numeric(combined_data_SCR$Personal.consumption.exp
enditures), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1        lag2       lag3          0      lead1      lead2
## 1 -0.03585143 0.007498788 0.04903682 -0.1123806 -0.1830187 -0.2101071
##        lead3
## 1 -0.2472862
```

```
#Lead lag with PVS, lead/lag by 3 quarters
cor.lag(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$GDP), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3         0     lead1      lead2    lead3
## 1 0.08015145 0.06631096 0.03601908 0.0893902 0.0743148 0.08403381 0.107336
```

```
cor.lag(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Gross.national.product),
3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2      lag3         0     lead1      lead2     lead3
## 1 0.07993165 0.06593927 0.0354166 0.0893971 0.0754573 0.08503114 0.1070069
```

```
cor.lag(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Gross.private.domestic.i
nvestment), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2         lag3          0       lead1       lead2
## 1 0.02451587 0.02159172 -0.003754029 0.00916365 -0.02666843 -0.02081932
##        lead3
## 1 0.01777224
```

```
cor.lag(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Nondurable.goods), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1        lag2        lag3          0      lead1         lead2
## 1 -0.02242471 -0.03878314 -0.05406731 0.04292936 0.05157852 -9.607408e-05
##        lead3
## 1 -0.0768786
```

```
cor.lag(as.numeric(combined_data_PVS$PVS), as.numeric(combined_data_PVS$Personal.consumption.exp
enditures), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3         0    lead1     lead2      lead3
## 1 0.08173458 0.05437384 0.02708213 0.1224156 0.129551 0.1107475 0.07955182
```

```
#Lead lag with Aggregate Risk Aversion, lead/lag by 3 quarters
cor.lag(as.numeric(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion), as.numeric(co
mbined_data_Aggregate_Risk_Aversion$GDP), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1      lag2      lag3           0       lead1     lead2       lead3
## 1 0.04134718 0.1479951 0.2229041 0.004114272 -0.02570821 0.0361105 -0.02533394
```

```
cor.lag(as.numeric(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion), as.numeric(co
mbined_data_Aggregate_Risk_Aversion$Gross.national.product), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1      lag2      lag3           0       lead1      lead2       lead3
## 1 0.04255377 0.1476026 0.2218208 0.006625511 -0.02480861 0.03800127 -0.02146152
```

```
cor.lag(as.numeric(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion), as.numeric(co
mbined_data_Aggregate_Risk_Aversion$Gross.private.domestic.investment), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1        lag2      lag3          0      lead1       lead2      lead3
## 1 -0.1703883 -0.09095086 0.0287133 -0.1900766 -0.0360316 -0.08082954 -0.1539064
```

```
cor.lag(as.numeric(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion),
as.numeric(combined_data_Aggregate_Risk_Aversion$Nondurable.goods), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1      lag2        lag3           0      lead1       lead2
## 1 0.006630801 0.0282608 -0.03775896 -0.06534602 -0.2385257 -0.07694836
##        lead3
## 1 0.01170283
```

```
cor.lag(as.numeric(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion), as.numeric(co
mbined_data_Aggregate_Risk_Aversion$Personal.consumption.expenditures), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3          0      lead1        lead2      lead3
## 1 0.1056599 0.1760371 0.1600517 0.01602137 -0.1510244 -0.002905762 0.08317397
```

```
#lead lag with UNE, lead/lag by 3 quarters
cor.lag(as.numeric(combined_data_UNE$UNE), as.numeric(combined_data_UNE$GDP), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1         lag2       lag3          0     lead1     lead2     lead3
## 1 0.01999982 -0.002988716 -0.0253187 0.06181774 0.1114987 0.1653227 0.2213928
```

```
cor.lag(as.numeric(combined_data_UNE$UNE), as.numeric(combined_data_UNE$Gross.national.product),
3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1         lag2        lag3          0     lead1     lead2     lead3
## 1 0.02046641 -0.002382109 -0.02490657 0.06182329 0.1111266 0.1647685 0.2208147
```

```
cor.lag(as.numeric(combined_data_UNE$UNE), as.numeric(combined_data_UNE$Gross.private.domestic.i
nvestment), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1       lag2        lag3           0      lead1      lead2
## 1 -0.07035415 -0.0826743 -0.07960284 -0.03178499 0.01582236 0.07416859
##       lead3
## 1 0.1215422
```

```
cor.lag(as.numeric(combined_data_UNE$UNE), as.numeric(combined_data_UNE$Nondurable.goods), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1        lag2        lag3          0       lead1       lead2
## 1 -0.06109162 -0.05939459 -0.07048543 -0.0515916 -0.03639317 0.001732551
##        lead3
## 1 0.06622263
```

```
cor.lag(as.numeric(combined_data_UNE$UNE), as.numeric(combined_data_UNE$Personal.consumption.exp
enditures), 3,3)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3          0     lead1     lead2     lead3
## 1 0.05138572 0.03617665 0.01544666 0.08001998 0.1154903 0.1624105 0.2253721
```

```
library(tidyverse)
#importing the 1990-2023 yield curve dataset:
yield_data <- read_csv("yield_curves.csv")
```

```
## Rows: 8507 Columns: 14
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): Date
## dbl (13): 1 Mo, 2 Mo, 3 Mo, 4 Mo, 6 Mo, 1 Yr, 2 Yr, 3 Yr, 5 Yr, 7 Yr, 10 Yr,...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Code Source: ChatGPT
#Create new dataset with only the first of the month with the yield curve dataset
# Load necessary libraries
library(dplyr)
library(lubridate)

#Create dummy dataset to store colnames for later transfer
yield_data_placeholder <- yield_data
# Rename columns for easier handling
colnames(yield_data) <- c("Date", paste0("V", 2:ncol(yield_data)))

# Convert the Date column to Date type
yield_data$Date <- as.Date(yield_data$Date, format = "%m/%d/%y")

# Group by year and month, then slice the last row of each group to get the
#first date of the month
first_of_month <- yield_data %>%
  group_by(year = year(Date), month = month(Date)) %>%
  slice(n()) %>%
  ungroup() %>%
  select(-year, -month)

# Optionally, write the result to a new CSV file
write.csv(first_of_month, "yield_data_mod.csv", row.names = FALSE)
```

```
#Open new modified dataset

library(tidyverse)
#importing the 1990-2023 yield curve dataset:
yield_data_mod <- read_csv("yield_data_mod.csv")
```

```
## Rows: 408 Columns: 14
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## dbl  (13): V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14
## date  (1): Date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
colnames(yield_data_mod) <- colnames(yield_data_placeholder)
```

```
#rename column names
colnames(yield_data_mod)[2] <- "One_Month"
colnames(yield_data_mod)[3] <- "Two_Month"
colnames(yield_data_mod)[4] <- "Three_Month"
colnames(yield_data_mod)[5] <- "Four_Month"
colnames(yield_data_mod)[6] <- "Six_Month"
colnames(yield_data_mod)[7] <- "One_Yr"
colnames(yield_data_mod)[8] <- "Two_Yr"
colnames(yield_data_mod)[9] <- "Three_Yr"
colnames(yield_data_mod)[10] <- "Five_Yr"
colnames(yield_data_mod)[11] <- "Seven_Yr"
colnames(yield_data_mod)[12] <- "Ten_Yr"
colnames(yield_data_mod)[13] <- "Twenty_Yr"
colnames(yield_data_mod)[14] <- "Thirty_Yr"
```

```
#create columns for differences in yield curves of the following time frames:

#30yr/1month, 10yr/1month, 5yr/1month, 1yr/1month, 6month/1month,

yield_data_mod$Thirty_Yr_One_Month_Difference <- yield_data_mod$Thirty_Yr - yield_data_mod$One_M
onth
yield_data_mod$Ten_Yr_One_Month_Difference <- yield_data_mod$Ten_Yr - yield_data_mod$One_Month
yield_data_mod$Five_Yr_One_Month_Difference <- yield_data_mod$Five_Yr - yield_data_mod$One_Month
yield_data_mod$One_Yr_One_Month_Difference <- yield_data_mod$One_Yr - yield_data_mod$One_Month
yield_data_mod$Six_Month_One_Month_Difference <- yield_data_mod$Six_Month - yield_data_mod$One_M
onth
```

```
#Convert yield curve data now to a quarterly dataset
#will use this with the quarterly gdp growth data
yield_data_quarterly = yield_data_mod[seq(1, nrow(yield_data_mod), 3), ]
```

```
#Create a dataset from 1990 - 2015 from the quarterly GDP growth data
#So the correlation tests have the same dimensions
US_GDP_Data_1990_2015 <- gdp_data_mod_final[c(121:224),]
#Subset yield curve quarterly dataset to just 1990 - 2015
yield_data_quarterly_1990_2015 <- yield_data_quarterly[c(1:104),]
```

```
#install package needed for lead/lag correlation
#if(!require('YRmisc')) {
 # install.packages('YRmisc')
  library('YRmisc')
#}
```

```
#rename gdp dataset columns
colnames(US_GDP_Data_1990_2015)[4] <- "Personal.consumption.expenditures"
colnames(US_GDP_Data_1990_2015)[7] <- "Nondurable.goods"
colnames(US_GDP_Data_1990_2015)[9] <- "Gross.private.domestic.investment"
colnames(US_GDP_Data_1990_2015)[30] <- "Gross.national.product"
```

```
#start correlation tests
#start with GDP growth
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(US_GDP_Data_1990_2015$GDP), as.numeric(yield_data_quarterly_1990_2015$Thirty_
Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1       lag2       lag3       lag4       lag5          0      lead1
## 1 -0.405835 -0.5506846 -0.5902454 -0.6192305 -0.6098977 -0.3684437 -0.2858393
##        lead2      lead3       lead4     lead5
## 1 -0.2262614 -0.1912775 0.002795187 0.1075629
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$GDP), as.numeric(yield_data_quarterly_1990_2015$Ten_Yr_
One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3       lag4       lag5          0      lead1
## 1 -0.3529626 -0.5036078 -0.5511803 -0.5491531 -0.5365836 -0.2967812 -0.2208029
##        lead2       lead3      lead4     lead5
## 1 -0.1368599 -0.04478071 0.07132896 0.1733737
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$GDP), as.numeric(yield_data_quarterly_1990_2015$Five_Yr
_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3      lag4       lag5          0      lead1
## 1 -0.3034147 -0.4197241 -0.4747101 -0.430389 -0.3882673 -0.2484766 -0.1718609
##         lead2      lead3     lead4     lead5
## 1 -0.09638347 0.02927426 0.1586446 0.2442433
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$GDP), as.numeric(yield_data_quarterly_1990_2015$One_Yr_
One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4      lag5         0     lead1
## 1 0.2026089 0.1644264 0.1421893 0.1726541 0.1852846 0.1222619 0.1070378
##       lead2     lead3     lead4     lead5
## 1 0.1888372 0.2320015 0.2684447 0.2558163
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$GDP), as.numeric(yield_data_quarterly_1990_2015$Six_Mon
th_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4      lag5         0     lead1
## 1 0.4139833 0.4034797 0.4097152 0.3943603 0.3229498 0.2445861 0.1646044
##       lead2     lead3     lead4     lead5
## 1 0.2552475 0.2262544 0.1871368 0.1300083
```

```
#Next GNP growth
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(US_GDP_Data_1990_2015$Gross.national.product), as.numeric(yield_data_quarterl
y_1990_2015$Thirty_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3       lag4       lag5          0      lead1
## 1 -0.4035288 -0.5503022 -0.5887217 -0.6254682 -0.6121424 -0.3722738 -0.2916056
##        lead2      lead3       lead4     lead5
## 1 -0.2358567 -0.1956195 -0.00610444 0.1021802
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Gross.national.product), as.numeric(yield_data_quarterl
y_1990_2015$Ten_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3       lag4       lag5          0      lead1
## 1 -0.3510782 -0.5022763 -0.5499334 -0.5511521 -0.5351201 -0.3000572 -0.2226441
##        lead2       lead3      lead4     lead5
## 1 -0.1395049 -0.04525536 0.06635318 0.1727987
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Gross.national.product), as.numeric(yield_data_quarterl
y_1990_2015$Five_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2      lag3       lag4       lag5          0      lead1
## 1 -0.3017541 -0.4183104 -0.474129 -0.4319523 -0.3867651 -0.2525058 -0.1726699
##         lead2      lead3     lead4    lead5
## 1 -0.09598218 0.03108908 0.1541889 0.245051
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Gross.national.product), as.numeric(yield_data_quarterl
y_1990_2015$One_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1     lag2      lag3      lag4      lag5         0     lead1
## 1 0.2026021 0.166439 0.1452072 0.1716307 0.1881913 0.1177029 0.1135229
##       lead2    lead3     lead4     lead5
## 1 0.1982335 0.237557 0.2642846 0.2561308
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Gross.national.product), as.numeric(yield_data_quarterl
y_1990_2015$Six_Month_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4      lag5         0     lead1
## 1 0.4115488 0.4045819 0.4134819 0.3938525 0.3272865 0.2406818 0.1725632
##       lead2     lead3    lead4     lead5
## 1 0.2633875 0.2289167 0.183004 0.1305698
```

```
#Next PCE growth
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(US_GDP_Data_1990_2015$Personal.consumption.expenditures), as.numeric(yield_da
ta_quarterly_1990_2015$Thirty_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1       lag2       lag3      lag4       lag5          0     lead1
## 1 -0.07984092 -0.1898094 -0.3222727 -0.308812 -0.3162811 -0.1951522 -0.203983
##        lead2      lead3       lead4       lead5
## 1 -0.2010106 -0.1860464 -0.09970112 -0.01288533
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Personal.consumption.expenditures), as.numeric(yield_da
ta_quarterly_1990_2015$Ten_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1      lag2       lag3       lag4       lag5         0      lead1
## 1 -0.09768697 -0.198854 -0.3192337 -0.2902866 -0.3169118 -0.172374 -0.1901872
##        lead2       lead3        lead4      lead5
## 1 -0.1302575 -0.02578038 -0.004503886 0.05141242
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Personal.consumption.expenditures), as.numeric(yield_da
ta_quarterly_1990_2015$Five_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1       lag2       lag3       lag4       lag5          0      lead1
## 1 -0.09205922 -0.1676014 -0.3259669 -0.2732024 -0.2674604 -0.1621437 -0.1773524
##         lead2      lead3      lead4      lead5
## 1 -0.08135327 0.06871547 0.06229734 0.08751194
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Personal.consumption.expenditures), as.numeric(yield_da
ta_quarterly_1990_2015$One_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2       lag3      lag4     lag5           0       lead1
## 1 0.1345578 0.1150573 0.01106598 0.1010594 0.101609 -0.06275114 -0.05911773
##       lead2     lead3      lead4      lead5
## 1 0.2174916 0.3211651 0.07974837 0.07000846
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Personal.consumption.expenditures), as.numeric(yield_da
ta_quarterly_1990_2015$Six_Month_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4     lag5          0      lead1
## 1 0.2375421 0.2483937 0.1788835 0.2455845 0.193242 -0.0325035 0.02129584
##       lead2     lead3      lead4        lead5
## 1 0.3459973 0.3100874 -0.0250835 -0.005104535
```

```
#Next Nondurable goods growth
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(US_GDP_Data_1990_2015$Nondurable.goods), as.numeric(yield_data_quarterly_1990
_2015$Thirty_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3       lag4       lag5           0      lead1
## 1 0.05851743 0.00998834 -0.1620981 -0.1490511 -0.1556158 -0.09160937 -0.1161622
##        lead2      lead3       lead4       lead5
## 1 -0.1216372 -0.1388995 -0.09745111 -0.04205229
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Nondurable.goods), as.numeric(yield_data_quarterly_1990
_2015$Ten_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1        lag2       lag3     lag4       lag5           0      lead1
## 1 0.02916988 -0.05489198 -0.1808787 -0.13835 -0.2379533 -0.08032229 -0.1133607
##         lead2      lead3        lead4        lead5
## 1 -0.05776027 0.01827931 -0.008386395 1.716672e-05
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Nondurable.goods), as.numeric(yield_data_quarterly_1990
_2015$Five_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1        lag2       lag3       lag4       lag5           0
## 1 0.01095383 -0.04916197 -0.2200805 -0.1507877 -0.2473969 -0.09924371
##        lead1       lead2      lead3      lead4      lead5
## 1 -0.1285704 -0.02826003 0.09290757 0.03660427 0.01582197
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Nondurable.goods), as.numeric(yield_data_quarterly_1990
_2015$One_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2        lag3      lag4        lag5           0       lead1
## 1 0.1435522 0.1026414 -0.03697328 0.1015757 0.000767996 -0.09183262 -0.09411985
##       lead2     lead3        lead4       lead5
## 1 0.2060438 0.3110709 -0.005332262 -0.02481114
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Nondurable.goods), as.numeric(yield_data_quarterly_1990
_2015$Six_Month_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2       lag3      lag4       lag5          0       lead1
## 1 0.2142764 0.1892911 0.06316292 0.1920792 0.08587578 -0.1025881 -0.03158496
##       lead2    lead3      lead4       lead5
## 1 0.3285639 0.291548 -0.1073617 -0.05299456
```

```
#Next Gross Private Domestic Investment growth
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(US_GDP_Data_1990_2015$Gross.private.domestic.investment), as.numeric(yield_da
ta_quarterly_1990_2015$Thirty_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3       lag4       lag5          0      lead1
## 1 -0.3625157 -0.4858465 -0.5038305 -0.5713947 -0.5297448 -0.1830244 -0.1429376
##         lead2       lead3      lead4      lead5
## 1 -0.06882605 -0.06011408 0.06258442 0.04926725
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Gross.private.domestic.investment), as.numeric(yield_da
ta_quarterly_1990_2015$Ten_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1      lag2       lag3       lag4       lag5          0      lead1
## 1 -0.3102579 -0.451823 -0.4777192 -0.5467065 -0.5205791 -0.1721629 -0.1164968
##         lead2       lead3     lead4     lead5
## 1 -0.02530308 0.006567636 0.1209224 0.1595254
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Gross.private.domestic.investment), as.numeric(yield_da
ta_quarterly_1990_2015$Five_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1      lag2       lag3       lag4      lag5          0       lead1
## 1 -0.2279932 -0.369249 -0.3587129 -0.4124834 -0.388886 -0.1251308 -0.09319294
##         lead2       lead3    lead4    lead5
## 1 -0.03104259 0.005778945 0.144718 0.177968
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Gross.private.domestic.investment), as.numeric(yield_da
ta_quarterly_1990_2015$One_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1    lag2      lag3      lag4       lag5         0     lead1
## 1 0.2057375 0.18341 0.2436969 0.1747619 0.09656071 0.2796304 0.1024089
##          lead2      lead3     lead4      lead5
## 1 -0.001332211 0.01503789 0.1885972 0.07641486
```

```
cor.lag(as.numeric(US_GDP_Data_1990_2015$Gross.private.domestic.investment), as.numeric(yield_da
ta_quarterly_1990_2015$Six_Month_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4      lag5        0     lead1
## 1 0.3406752 0.3508787 0.4432695 0.3734595 0.2661091 0.398788 0.1145233
##         lead2      lead3     lead4       lead5
## 1 -0.02131274 0.03216943 0.1840561 -0.02777962
```

```
#Now we can run correlations with the equity premium/risk aversion measures
combined_data_IPO_Gross_1990_2015 <-combined_data_IPO_Gross[c(121:224),]
combined_data_SCR_1990_2015 <-combined_data_SCR[c(121:224),]
combined_data_UNE_1990_2015 <-combined_data_UNE[c(121:224),]
combined_data_DPRatio_1990_2015 <-combined_data_DPRatio[c(121:224),]
combined_data_PVS_1990_2015 <-combined_data_PVS[c(80:183),]
yield_data_quarterly_1990_2010 <- yield_data_quarterly[c(1:84),]
yield_data_quarterly_1996_2011 <- yield_data_quarterly[c(24:87),]
```

```
#start correlation tests
#start with Gross IPO
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(combined_data_IPO_Gross_1990_2015$IPO_Gross), as.numeric(yield_data_quarterly
_1990_2015$Thirty_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3       lag4      lag5          0      lead1
## 1 -0.2490041 -0.3577744 -0.4560631 -0.4916506 -0.485306 -0.2290332 -0.1188667
##         lead2     lead3     lead4     lead5
## 1 -0.06437216 0.1643152 0.2482421 0.3419199
```

```
cor.lag(as.numeric(combined_data_IPO_Gross_1990_2015$IPO_Gross), as.numeric(yield_data_quarterly
_1990_2015$Ten_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1      lag2       lag3       lag4       lag5          0       lead1
## 1 -0.2596349 -0.413789 -0.4828112 -0.5379037 -0.5560356 -0.2218005 -0.08792862
##         lead2      lead3     lead4     lead5
## 1 -0.03876602 0.06496029 0.1329283 0.2334571
```

```
cor.lag(as.numeric(combined_data_IPO_Gross_1990_2015$IPO_Gross), as.numeric(yield_data_quarterly
_1990_2015$Five_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2      lag3      lag4       lag5          0       lead1
## 1 -0.2135907 -0.3587748 -0.397683 -0.450231 -0.4575383 -0.1917709 -0.05762308
##         lead2      lead3      lead4     lead5
## 1 -0.05789802 0.02935083 0.06928289 0.1765559
```

```
cor.lag(as.numeric(combined_data_IPO_Gross_1990_2015$IPO_Gross), as.numeric(yield_data_quarterly
_1990_2015$One_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1        lag2       lag3     lag4        lag5           0      lead1
## 1 0.01483255 -0.09393615 0.04868347 0.123288 -0.04620689 0.005687791 0.09619519
##        lead2       lead3       lead4    lead5
## 1 -0.0617238 -0.04249465 -0.01404813 0.078659
```

```
cor.lag(as.numeric(combined_data_IPO_Gross_1990_2015$GDP), as.numeric(yield_data_quarterly_1990_
2015$Six_Month_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4      lag5         0     lead1
## 1 0.4139833 0.4034797 0.4097152 0.3943603 0.3229498 0.2445861 0.1646044
##       lead2     lead3     lead4     lead5
## 1 0.2552475 0.2262544 0.1871368 0.1300083
```

```
#start correlation tests
#Next EP Bound
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(combined_data_EPBound$EPBound), as.numeric(yield_data_quarterly_1996_2011$Thi
rty_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4     lag5         0     lead1
## 1 0.2817389 0.2699052 0.3909216 0.5178889 0.521562 0.2140764 0.2036877
##          lead2      lead3      lead4      lead5
## 1 -0.008906616 -0.1385379 -0.3142986 -0.4543355
```

```
cor.lag(as.numeric(combined_data_EPBound$EPBound), as.numeric(yield_data_quarterly_1996_2011$Ten
_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3     lag4      lag5         0     lead1
## 1 0.2805562 0.2471832 0.3578379 0.495094 0.5137386 0.2147701 0.1438197
##         lead2      lead3      lead4      lead5
## 1 -0.08062193 -0.2017413 -0.3296576 -0.4715826
```

```
cor.lag(as.numeric(combined_data_EPBound$EPBound), as.numeric(yield_data_quarterly_1996_2011$Fiv
e_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4      lag5         0     lead1
## 1 0.2577609 0.1982543 0.2949487 0.4635323 0.4768092 0.2146393 0.1198954
##        lead2      lead3     lead4      lead5
## 1 -0.1405645 -0.2316164 -0.331557 -0.4861471
```

```
cor.lag(as.numeric(combined_data_EPBound$EPBound), as.numeric(yield_data_quarterly_1996_2011$One
_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1       lag2        lag3       lag4        lag5        0       lead1
## 1 0.1839092 -0.1294751 -0.07573419 -0.0647166 -0.08176314 0.144043 -0.08143974
##       lead2       lead3      lead4      lead5
## 1 -0.246314 -0.08869405 -0.1026577 -0.3194498
```

```
cor.lag(as.numeric(combined_data_EPBound$EPBound), as.numeric(yield_data_quarterly_1996_2011$Six
_Month_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##       lag1       lag2       lag3       lag4       lag5           0      lead1
## 1 0.114865 -0.2273086 -0.2548797 -0.2803888 -0.3253335 -0.01048609 -0.2189486
##       lead2      lead3     lead4       lead5
## 1 -0.193636 0.08503134 0.1272671 -0.06916968
```

```
#start correlation tests
#Next DPRatio
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(combined_data_DPRatio_1990_2015$DPRatio), as.numeric(yield_data_quarterly_199
0_2015$Thirty_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4      lag5         0     lead1
## 1 0.3090331 0.4124365 0.4645109 0.4572686 0.4272795 0.2367616 0.1977863
##       lead2       lead3      lead4      lead5
## 1 0.1326435 -0.01724718 -0.1803861 -0.3233929
```

```
cor.lag(as.numeric(combined_data_DPRatio_1990_2015$DPRatio), as.numeric(yield_data_quarterly_199
0_2015$Ten_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1      lag2     lag3      lag4      lag5          0      lead1
## 1 0.06517153 0.1500781 0.207241 0.2111456 0.1919372 0.05882929 0.06037231
##        lead2     lead3      lead4      lead5
## 1 0.01077041 -0.109493 -0.2233307 -0.3077866
```

```
cor.lag(as.numeric(combined_data_DPRatio_1990_2015$DPRatio), as.numeric(yield_data_quarterly_199
0_2015$Five_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1      lag2       lag3       lag4       lag5           0
## 1 -0.05286198 0.0339446 0.09724124 0.09204338 0.05906653 -0.03458114
##          lead1       lead2      lead3      lead4      lead5
## 1 -0.009834648 -0.05853814 -0.2008687 -0.3083086 -0.3721285
```

```
cor.lag(as.numeric(combined_data_DPRatio_1990_2015$DPRatio), as.numeric(yield_data_quarterly_199
0_2015$One_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1       lag2       lag3       lag4       lag5         0     lead1
## 1 -0.128805 -0.1458594 -0.1729403 -0.2273921 -0.2523675 0.0153276 0.1203489
##        lead2      lead3      lead4      lead5
## 1 0.02770654 -0.1437345 -0.1760203 -0.1702907
```

```
cor.lag(as.numeric(combined_data_DPRatio_1990_2015$DPRatio), as.numeric(yield_data_quarterly_199
0_2015$Six_Month_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##          lag1      lag2       lag3       lag4       lag5          0     lead1
## 1 -0.09617418 -0.138269 -0.1745073 -0.2282657 -0.2242137 0.06133601 0.1405363
##        lead2       lead3       lead4      lead5
## 1 0.02889627 -0.06874268 0.008391328 0.08460598
```

```
#start correlation tests
#Next UNE
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(combined_data_UNE_1990_2015$UNE), as.numeric(yield_data_quarterly_1990_2015$T
hirty_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2     lag3      lag4      lag5         0     lead1
## 1 0.7435257 0.7066131 0.657007 0.5880372 0.5091089 0.7462221 0.7359633
##       lead2     lead3     lead4     lead5
## 1 0.7049539 0.6278654 0.5271873 0.4050993
```

```
cor.lag(as.numeric(combined_data_UNE_1990_2015$UNE), as.numeric(yield_data_quarterly_1990_2015$T
en_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4      lag5         0     lead1
## 1 0.5255701 0.4829916 0.4146412 0.3293157 0.2385981 0.5361265 0.5305541
##       lead2     lead3     lead4     lead5
## 1 0.5014856 0.4396283 0.3639455 0.2684255
```

```
cor.lag(as.numeric(combined_data_UNE_1990_2015$UNE), as.numeric(yield_data_quarterly_1990_2015$F
ive_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3       lag4        lag5         0     lead1
## 1 0.3172103 0.2646069 0.1840011 0.09565203 0.008194058 0.3394972 0.3490714
##       lead2     lead3     lead4     lead5
## 1 0.3346095 0.2889595 0.2382115 0.1676777
```

```
cor.lag(as.numeric(combined_data_UNE_1990_2015$UNE), as.numeric(yield_data_quarterly_1990_2015$O
ne_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2      lag3       lag4       lag5          0      lead1
## 1 -0.2533282 -0.3086475 -0.358563 -0.3869991 -0.4005946 -0.2014011 -0.1468067
##        lead2       lead3       lead4       lead5
## 1 -0.1019723 -0.08559071 -0.06843456 -0.06454757
```

```
cor.lag(as.numeric(combined_data_UNE_1990_2015$UNE), as.numeric(yield_data_quarterly_1990_2015$S
ix_Month_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3       lag4     lag5          0      lead1
## 1 -0.3958264 -0.4281704 -0.4396247 -0.4254929 -0.40129 -0.3495832 -0.3052544
##        lead2      lead3      lead4      lead5
## 1 -0.2543689 -0.2258768 -0.1794886 -0.1290489
```

```
#start correlation tests
#Next Surplus Consumption Ratio
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(combined_data_SCR_1990_2015$SCR), as.numeric(yield_data_quarterly_1990_2015$T
hirty_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2      lag3       lag4       lag5          0      lead1
## 1 -0.4350235 -0.5487135 -0.627014 -0.6621643 -0.6619809 -0.3009275 -0.1728198
##         lead2     lead3     lead4     lead5
## 1 -0.01595468 0.2023548 0.3610834 0.5228242
```

```
cor.lag(as.numeric(combined_data_SCR_1990_2015$SCR), as.numeric(yield_data_quarterly_1990_2015$T
en_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##       lag1       lag2       lag3       lag4       lag5          0       lead1
## 1 -0.27688 -0.3610034 -0.4566356 -0.4955205 -0.4774172 -0.1839565 -0.06556771
##       lead2     lead3    lead4    lead5
## 1 0.0657442 0.2221626 0.341822 0.467365
```

```
cor.lag(as.numeric(combined_data_SCR_1990_2015$SCR), as.numeric(yield_data_quarterly_1990_2015$F
ive_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##       lag1       lag2       lag3       lag4       lag5           0       lead1
## 1 -0.16015 -0.2242696 -0.3185327 -0.3433257 -0.3045797 -0.09046151 0.007511226
##   lead2     lead3     lead4     lead5
## 1 0.124 0.2639838 0.3626024 0.4718404
```

```
cor.lag(as.numeric(combined_data_SCR_1990_2015$SCR), as.numeric(yield_data_quarterly_1990_2015$O
ne_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1      lag2     lag3      lag4     lag5           0       lead1
## 1 0.08495279 0.1678182 0.150017 0.2204944 0.271727 -0.01765523 -0.01718479
##         lead2      lead3      lead4      lead5
## 1 -0.02209824 0.01053808 0.02225742 0.06340177
```

```
cor.lag(as.numeric(combined_data_SCR_1990_2015$SCR), as.numeric(yield_data_quarterly_1990_2015$S
ix_Month_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##       lag1      lag2      lag3      lag4      lag5           0       lead1
## 1 0.131575 0.2500427 0.2653027 0.3454292 0.3727666 0.001668026 -0.01800469
##         lead2      lead3      lead4      lead5
## 1 -0.07709825 -0.1122518 -0.1770881 -0.2102992
```

```
#start correlation tests
#Next Price of Volatile Stocks
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(combined_data_PVS_1990_2015$PVS), as.numeric(yield_data_quarterly_1990_2015$T
hirty_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1       lag2       lag3       lag4       lag5          0      lead1
## 1 -0.528562 -0.6171069 -0.6421205 -0.6467301 -0.6064673 -0.3977448 -0.3338521
##        lead2      lead3      lead4     lead5
## 1 -0.2643634 -0.1455555 0.03873096 0.2294318
```

```
cor.lag(as.numeric(combined_data_PVS_1990_2015$PVS), as.numeric(yield_data_quarterly_1990_2015$T
en_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1       lag2       lag3      lag4       lag5         0     lead1
## 1 -0.465897 -0.5714381 -0.6083885 -0.621589 -0.5970273 -0.337919 -0.311602
##        lead2      lead3      lead4    lead5
## 1 -0.2537542 -0.1489885 0.02542293 0.183497
```

```
cor.lag(as.numeric(combined_data_PVS_1990_2015$PVS), as.numeric(yield_data_quarterly_1990_2015$F
ive_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3       lag4       lag5          0      lead1
## 1 -0.3933277 -0.4968212 -0.5137561 -0.4946256 -0.4562462 -0.2819618 -0.2832283
##        lead2      lead3     lead4     lead5
## 1 -0.2410711 -0.1430133 0.0484131 0.2012187
```

```
cor.lag(as.numeric(combined_data_PVS_1990_2015$PVS), as.numeric(yield_data_quarterly_1990_2015$O
ne_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2      lag3      lag4      lag5           0      lead1
## 1 0.03324475 0.05542521 0.1148365 0.1742282 0.2005347 0.006206321 -0.1306071
##        lead2       lead3      lead4      lead5
## 1 -0.1407865 -0.03953171 0.06383149 0.09471739
```

```
cor.lag(as.numeric(combined_data_PVS_1990_2015$PVS), as.numeric(yield_data_quarterly_1990_2015$S
ix_Month_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4      lag5         0       lead1
## 1 0.2510753 0.3076852 0.3739457 0.4027047 0.3921877 0.1678358 0.008985924
##         lead2      lead3      lead4      lead5
## 1 0.009220593 0.06535054 0.03135304 -0.0528943
```

```
#start correlation tests
#Next Aggregate Risk Aversion
#Going up to 5 quarters of lead and lag

cor.lag(as.numeric(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion), as.numeric(yi
eld_data_quarterly_1990_2010$Thirty_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2     lag3      lag4     lag5         0     lead1     lead2
## 1 0.7444627 0.6690366 0.546148 0.4286686 0.268961 0.7540567 0.8274938 0.8271205
##       lead3     lead4     lead5
## 1 0.8459761 0.7611856 0.6574115
```

```
cor.lag(as.numeric(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion), as.numeric(yi
eld_data_quarterly_1990_2010$Ten_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4      lag5         0     lead1
## 1 0.6759541 0.6692777 0.5813307 0.4930813 0.3481942 0.6974179 0.7617075
##       lead2     lead3     lead4     lead5
## 1 0.7494194 0.6996251 0.5499948 0.4047998
```

```
cor.lag(as.numeric(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion), as.numeric(yi
eld_data_quarterly_1990_2010$Five_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1      lag2      lag3      lag4      lag5         0     lead1
## 1 0.5892823 0.6206533 0.5341924 0.4477098 0.3020164 0.6219372 0.7072518
##       lead2     lead3     lead4     lead5
## 1 0.7153384 0.6480393 0.4761057 0.3435055
```

```
cor.lag(as.numeric(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion), as.numeric(yi
eld_data_quarterly_1990_2010$One_Yr_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##        lag1        lag2        lag3        lag4        lag5           0
## 1 -0.116763 -0.06678145 -0.07309843 -0.07202384 -0.05336929 -0.08950452
##       lead1     lead2      lead3       lead4       lead5
## 1 0.1121363 0.1505246 0.05759054 -0.06612835 -0.01714698
```

```
cor.lag(as.numeric(combined_data_Aggregate_Risk_Aversion$Aggregate_Risk_Aversion), as.numeric(yi
eld_data_quarterly_1990_2010$Six_Month_One_Month_Difference), 5,5)
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion

## Warning in cv.lag(x, i): NAs introduced by coercion
```

```
## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion

## Warning in cv.lead(x, i): NAs introduced by coercion
```

```
##         lag1       lag2       lag3       lag4       lag5          0      lead1
## 1 -0.4405596 -0.3905426 -0.3567822 -0.3034468 -0.2012305 -0.4127561 -0.2340348
##       lead2      lead3      lead4      lead5
## 1 -0.265647 -0.3480587 -0.3486544 -0.1817647
```