Homework 3

Maxime Grossman (UNI:mmg2240)

6/7/2021

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.6.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------------------------ tidyverse
1.3.0 --
```

```
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.3
```

```
## Warning: package 'forcats' was built under R version 3.6.3
```

```
## -- Conflicts ------------------------------------------------------------------------------------------ tidyverse_confl
icts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
```

#1. Calculate the average GDP growth rate for each country (averaging over years). This is a classic split/apply/combine problem, and you will use daply()to solve it.

#a. Begin by writing a function, mean.growth(), that takes a data frame as its argument and returns the mean of the ‘growth‘ column of that data frame.

```
debt <- read.csv("debt.csv", as.is = TRUE)
dim(debt)
```

```
## [1] 1171    4
```

```
head(debt)
```

```
##     Country Year    growth     ratio
## 1 Australia 1946 -3.557951 190.41908
## 2 Australia 1947  2.459475 177.32137
## 3 Australia 1948  6.437534 148.92981
## 4 Australia 1949  6.611994 125.82870
## 5 Australia 1950  6.920201 109.80940
## 6 Australia 1951  4.272612  87.09448
```

```
mean.growth <- function(debt){
  mean(debt$growth)
}
```

#b. Use daply()to apply mean.growth()to each country in debt. Don't use some-thing like mean(debt$growth[debt$Country=="Australia"]), except to check your work. You should not need to use a loop to do this. (The average growth rates for Australia and the Netherlands should be 3.72 and 3.03. Print these values.) Report the average GDP growth rates clearly.

```
debtmean <- daply(debt, .(Country), mean.growth)

as.data.frame(debtmean)
```

```
##             debtmean
## Australia   3.721597
## Austria     4.438030
## Belgium     3.176287
## Canada      3.652017
## Denmark     2.656741
## Finland     3.571897
## France      3.776350
## Germany     3.314818
## Greece      2.927692
## Ireland     3.933766
## Italy       3.252528
## Japan       4.447166
## Netherlands 3.031161
## New Zealand 3.069408
## Norway      3.826671
## Portugal    4.002797
## Spain       3.196547
## Sweden      3.065083
## UK          2.414808
## US          2.997120
```

#2. Using the same instructions as problem 1, calculate the average GDP growth rate for each year (now averaging over countries). (The average growth rates for 1972 and 1989 should be 5.63 and 3.19, respectively. Print these values in your output.) Make a plot of the growth rates (y-axis) versus the year (x-axis). Make sure the axes are labeled appropriately.

```
debtmean2 <- daply(debt, .(Year), mean.growth)

as.data.frame(debtmean2)
```

```
##       debtmean2
## 1946  2.6239890
## 1947  5.4147299
## 1948  5.5648414
## 1949  4.7396296
## 1950  6.3214896
## 1951  4.9184456
## 1952  3.3976694
## 1953  4.0873110
## 1954  4.8828652
## 1955  5.1396220
## 1956  4.2313542
## 1957  3.9128688
## 1958  2.2362356
## 1959  5.3098167
## 1960  5.8604385
## 1961  4.8915229
## 1962  4.9571904
## 1963  4.8275013
## 1964  6.3654718
## 1965  4.7188763
## 1966  4.3093773
## 1967  4.0422048
## 1968  5.2665878
## 1969  6.2470505
## 1970  4.6064498
## 1971  4.0655311
## 1972  5.6299862
## 1973  5.9712432
## 1974  1.9944636
## 1975  0.8301904
## 1976  4.1659118
## 1977  2.6299752
## 1978  3.3230568
## 1979  4.1939645
## 1980  1.8711923
## 1981  0.9920489
## 1982  0.8758437
## 1983  2.0365803
## 1984  4.0582113
## 1985  3.5210599
## 1986  2.8879720
## 1987  2.4530780
## 1988  2.9223717
## 1989  3.1868422
## 1990  2.5665909
## 1991  1.3348964
## 1992  1.5891679
## 1993  1.0208583
## 1994  3.8585838
## 1995  3.6340300
## 1996  3.3896732
## 1997  4.0654455
## 1998  3.0850886
## 1999  3.4843512
## 2000  4.0559841
## 2001  2.0436501
## 2002  1.9685731
## 2003  1.8670089
## 2004  3.2936823
## 2005  2.6239322
## 2006  3.1381842
## 2007  3.1359031
## 2008  0.7980262
## 2009 -3.3668270
```

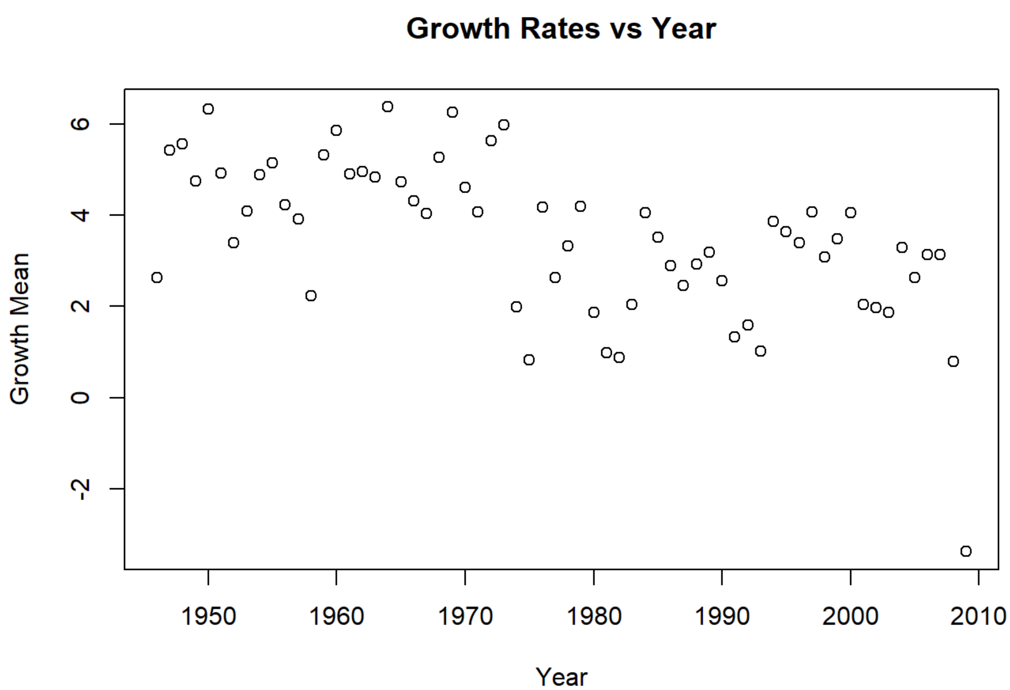
```r
yearvec <- seq(1946, 2009, 1)

yearvec
```

```
##  [1] 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960
## [16] 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975
## [31] 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990
## [46] 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005
## [61] 2006 2007 2008 2009
```

```r
debtmean2 <- cbind(yearvec, debtmean2)

debtmean2[,-1]
```

```
##       1946       1947       1948       1949       1950       1951       1952
##  2.6239890  5.4147299  5.5648414  4.7396296  6.3214896  4.9184456  3.3976694
##       1953       1954       1955       1956       1957       1958       1959
##  4.0873110  4.8828652  5.1396220  4.2313542  3.9128688  2.2362356  5.3098167
##       1960       1961       1962       1963       1964       1965       1966
##  5.8604385  4.8915229  4.9571904  4.8275013  6.3654718  4.7188763  4.3093773
##       1967       1968       1969       1970       1971       1972       1973
##  4.0422048  5.2665878  6.2470505  4.6064498  4.0655311  5.6299862  5.9712432
##       1974       1975       1976       1977       1978       1979       1980
##  1.9944636  0.8301904  4.1659118  2.6299752  3.3230568  4.1939645  1.8711923
##       1981       1982       1983       1984       1985       1986       1987
##  0.9920489  0.8758437  2.0365803  4.0582113  3.5210599  2.8879720  2.4530780
##       1988       1989       1990       1991       1992       1993       1994
##  2.9223717  3.1868422  2.5665909  1.3348964  1.5891679  1.0208583  3.8585838
##       1995       1996       1997       1998       1999       2000       2001
##  3.6340300  3.3896732  4.0654455  3.0850886  3.4843512  4.0559841  2.0436501
##       2002       2003       2004       2005       2006       2007       2008
##  1.9685731  1.8670089  3.2936823  2.6239322  3.1381842  3.1359031  0.7980262
##       2009
## -3.3668270
```

```r
plot(debtmean2, xlab = "Year", ylab = "Growth Mean", main = "Growth Rates vs Year")
```



#3. The function cor(x,y) calculates the correlation coefficient between two vectors x and y.

#a. Calculate the correlation coefficient between GDP growth and the debt ratio over the whole data set (all countries, all years). Your answer should be −0.1995.

```
cor(debt$growth, debt$ratio)
```

```
## [1] -0.199468
```

#b. Compute the correlation coefficient separately for each country, and plot a histogram of these coefficients (with 10 breaks). The mean of these correlations should be −0.1778. Do not use a loop.

(Hint: consider writing a function and then making it an argument to daply( )).

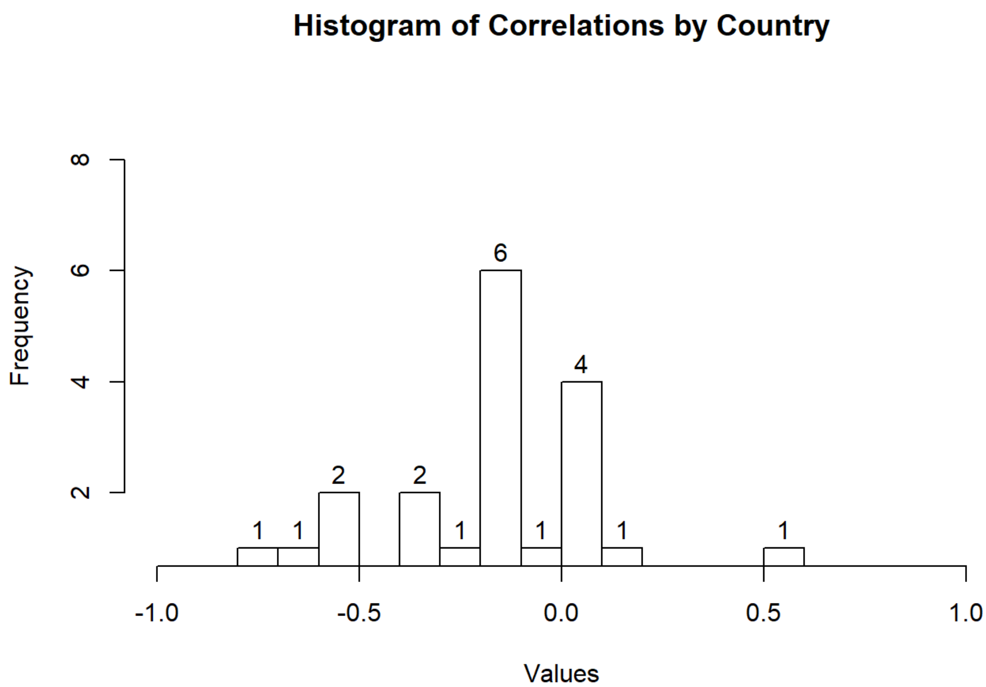
```
my.correlation <- function(debt){
  cor(debt$growth, debt$ratio)
}


all.correlations <- daply(debt, .(Country), my.correlation)

all.correlations <- as.data.frame(all.correlations)

all.correlations
```

```
##             all.correlations
## Australia       0.0251615558
## Austria        -0.2531950101
## Belgium        -0.1917817454
## Canada          0.0749755062
## Denmark        -0.1684588000
## Finland         0.0005814147
## France         -0.5019220059
## Germany        -0.5763190521
## Greece         -0.0935417558
## Ireland        -0.1403166337
## Italy          -0.6447261058
## Japan          -0.7018505928
## Netherlands    -0.1989566840
## New Zealand     0.1608454458
## Norway          0.5629128534
## Portugal       -0.3515764808
## Spain           0.0813828588
## Sweden         -0.1609485529
## UK             -0.1372358212
## US             -0.3414713369
```

```
mean(all.correlations$all.correlations)
```

```
## [1] -0.177822
```

```
hist(all.correlations$all.correlations, breaks=10, labels=TRUE, xlab = "Values", ylab = "Frequency", main = "Histogram of C
orrelations by Country", ylim=c(1,9), xlim=c(-1,1))
```



#c. Calculate the correlation coefficient separately for each year, and plot a histogram of these coefficients. The mean of these correlations should be −0.1906.

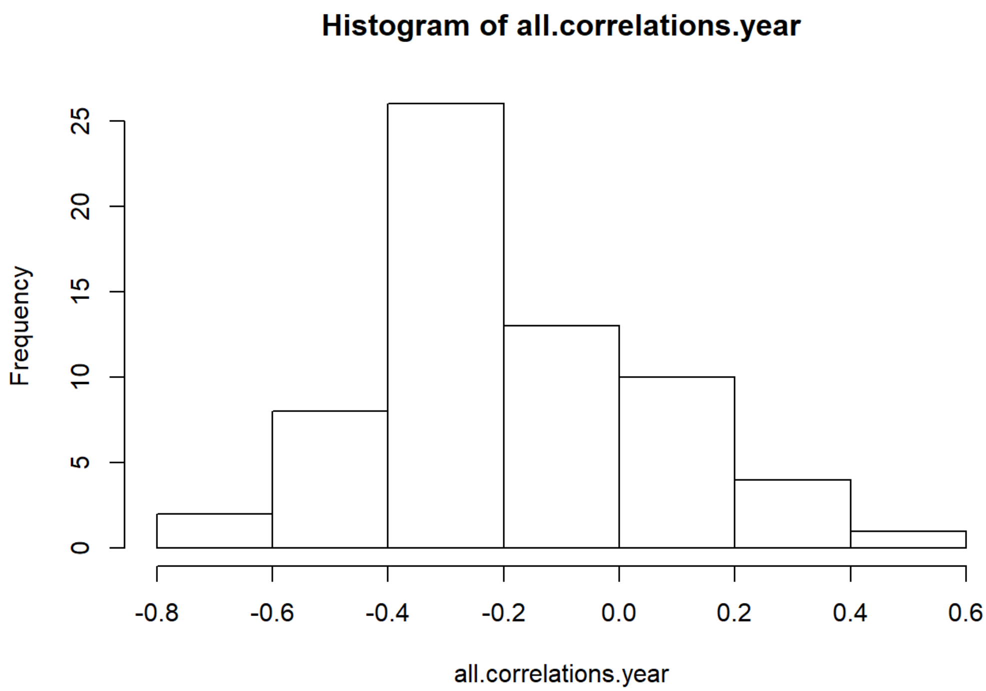
```
all.correlations.year <- daply(debt, .(Year), my.correlation)

all.correlations.year
```

```
##         1946         1947         1948         1949         1950         1951
## -0.620299284 -0.274137728 -0.340494128 -0.200450275  0.039754576 -0.415884891
##         1952         1953         1954         1955         1956         1957
## -0.276536771 -0.204991978 -0.275046325 -0.227065520 -0.457844543 -0.754985904
##         1958         1959         1960         1961         1962         1963
## -0.453943968 -0.284956232 -0.503645778 -0.539343507 -0.382533632  0.127811245
##         1964         1965         1966         1967         1968         1969
## -0.360729641 -0.310568392 -0.311484320 -0.277887063 -0.181341899 -0.250496906
##         1970         1971         1972         1973         1974         1975
## -0.512250332  0.008717128 -0.196087678  0.113716572  0.259851601  0.270698042
##         1976         1977         1978         1979         1980         1981
## -0.170765249  0.164476644  0.430658621 -0.428896967 -0.127292098  0.030394974
##         1982         1983         1984         1985         1986         1987
##  0.239084363 -0.361953817 -0.155643452 -0.449072217 -0.357841748 -0.068901146
##         1988         1989         1990         1991         1992         1993
##  0.079662167  0.066371467  0.155799702  0.202214712 -0.002217139 -0.372377205
##         1994         1995         1996         1997         1998         1999
## -0.223949231  0.051906462 -0.356974068 -0.111135985 -0.265148911 -0.257760224
##         2000         2001         2002         2003         2004         2005
## -0.133879407 -0.237546059 -0.349262614 -0.067904623 -0.170887379 -0.314330814
##         2006         2007         2008         2009
## -0.196041000 -0.344405985 -0.094533699 -0.204757778
```

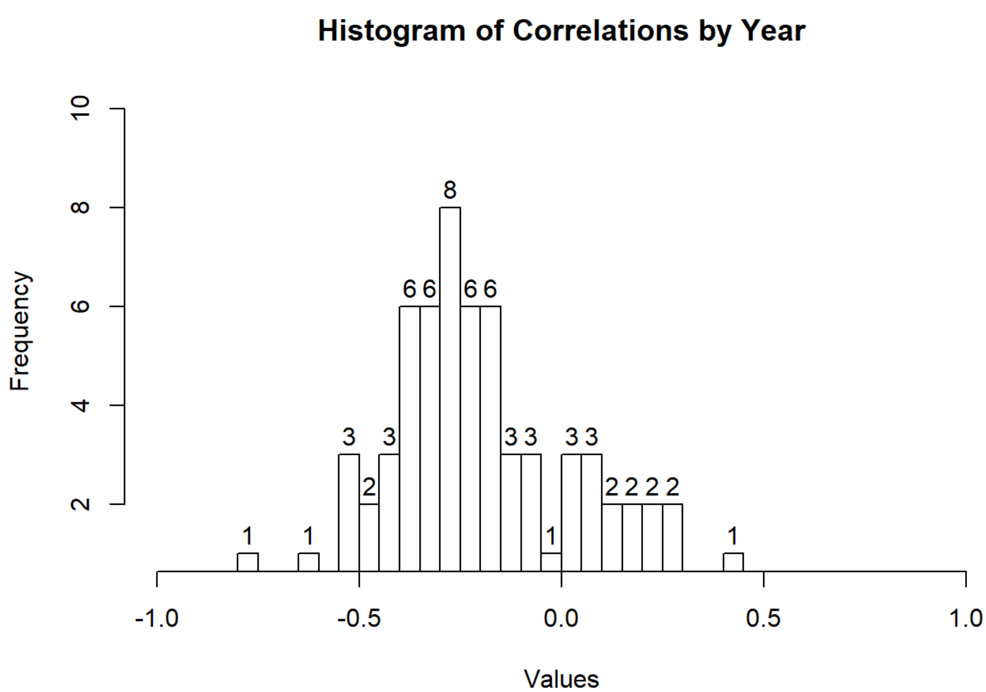
```
mean(all.correlations.year)
```

```
## [1] -0.1905526
```

```
hist(all.correlations.year)
```




```
all.correlations.year <- as.data.frame(all.correlations.year)


hist(all.correlations.year$all.correlations.year, breaks=20, labels=TRUE, xlab = "Values", ylab = "Frequency", main = "Hist
ogram of Correlations by Year", ylim=c(1,10), xlim=c(-1,1))
```

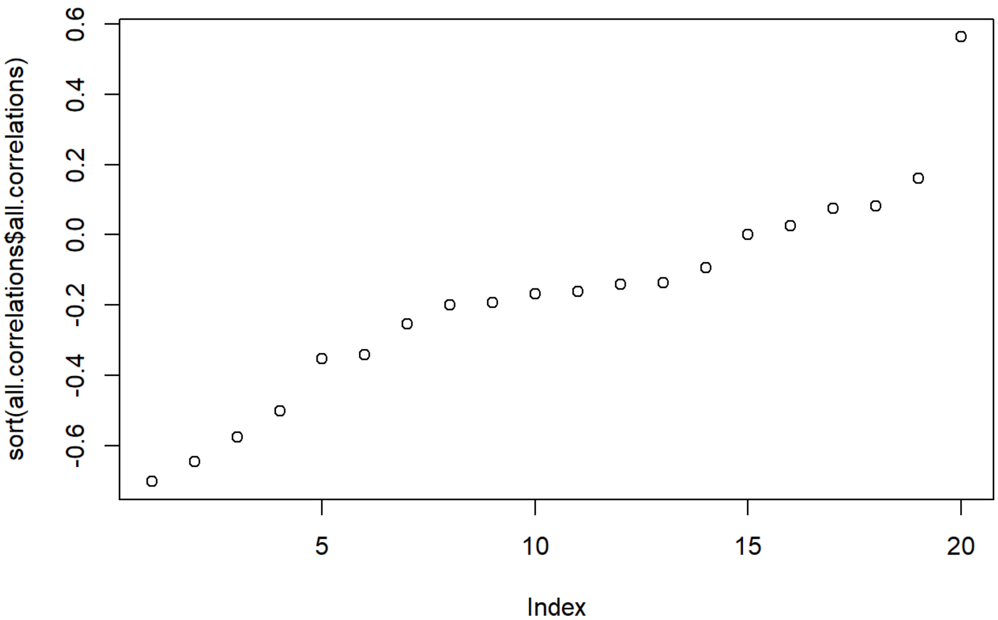


#d. Are there any countries or years where the correlation goes against the general trend?

```
sort(all.correlations$all.correlations)
```

```
##  [1] -0.7018505928 -0.6447261058 -0.5763190521 -0.5019220059 -0.3515764808
##  [6] -0.3414713369 -0.2531950101 -0.1989566840 -0.1917817454 -0.1684588000
## [11] -0.1609485529 -0.1403166337 -0.1372358212 -0.0935417558  0.0005814147
## [16]  0.0251615558  0.0749755062  0.0813828588  0.1608454458  0.5629128534
```

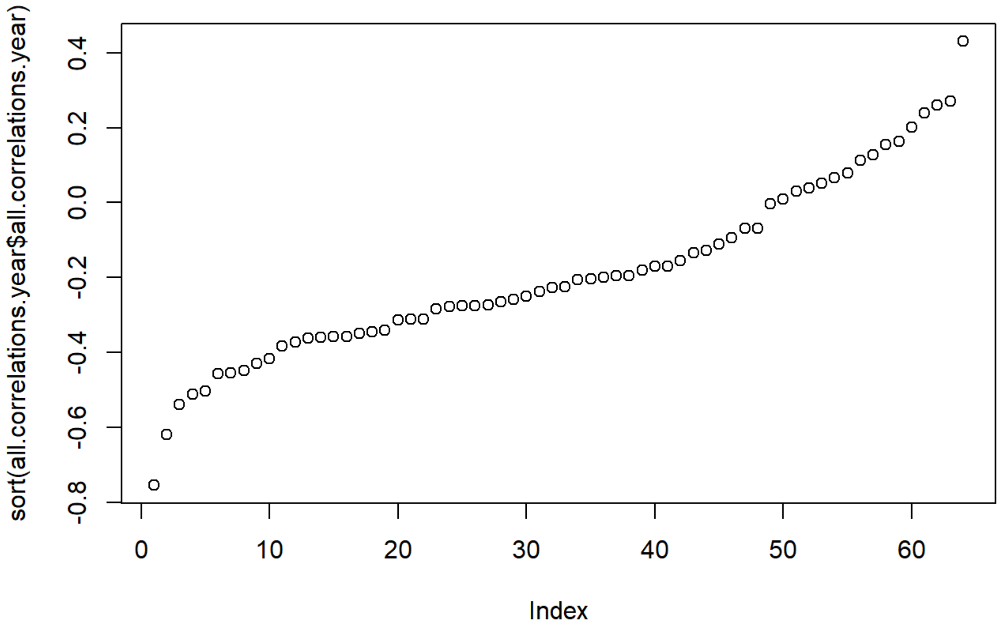
```
plot(sort(all.correlations$all.correlations))
```




```
sort(all.correlations.year$all.correlations.year)
```

```
##  [1] -0.754985904 -0.620299284 -0.539343507 -0.512250332 -0.503645778
##  [6] -0.457844543 -0.453943968 -0.449072217 -0.428896967 -0.415884891
## [11] -0.382533632 -0.372377205 -0.361953817 -0.360729641 -0.357841748
## [16] -0.356974068 -0.349262614 -0.344405985 -0.340494128 -0.314330814
## [21] -0.311484320 -0.310568392 -0.284956232 -0.277887063 -0.276536771
## [26] -0.275046325 -0.274137728 -0.265148911 -0.257760224 -0.250496906
## [31] -0.237546059 -0.227065520 -0.223949231 -0.204991978 -0.204757778
## [36] -0.200450275 -0.196087678 -0.196041000 -0.181341899 -0.170887379
## [41] -0.170765249 -0.155643452 -0.133879407 -0.127292098 -0.111135985
## [46] -0.094533699 -0.068901146 -0.067904623 -0.002217139  0.008717128
## [51]  0.030394974  0.039754576  0.051906462  0.066371467  0.079662167
## [56]  0.113716572  0.127811245  0.155799702  0.164476644  0.202214712
## [61]  0.239084363  0.259851601  0.270698042  0.430658621
```

```
plot(sort(all.correlations.year$all.correlations.year))
```



We can see by observation that both plots have an outlier that is greater than 0.4.

Which data points do these correspond to?

```
which(all.correlations$all.correlations > 0.4)
```

```
## [1] 15
```

```
which(all.correlations.year$all.correlations.year > 0.4)
```

```
## [1] 33
```

We can see by observation that the correlation of 0.5629128 is an outlier. This correlation corresponds to row 15: the country Norway.

By year, we can observe that the correlation of 0.43065 is an outlier. This corresponds to row 33: year 1978.

#4. Fit a linear model of overall growth on the debt ratio, using lm(). Report the intercept and slope. Make a scatter-plot of overall GDP growth (vertical) against the overall debt ratio (horizontal). Add a line to your scatterplot showing the fitted regression line.

```
my.strike.lm <- function(country.df)
  {
  return(coef(lm(strike.volume ~ left.parliament, data = country.df)))
}

debt.lm <- lm(growth ~ ratio, data = debt)


debt.lm
```

```
##
## Call:
## lm(formula = growth ~ ratio, data = debt)
##
## Coefficients:
## (Intercept)        ratio
##     4.27929     -0.01836
```

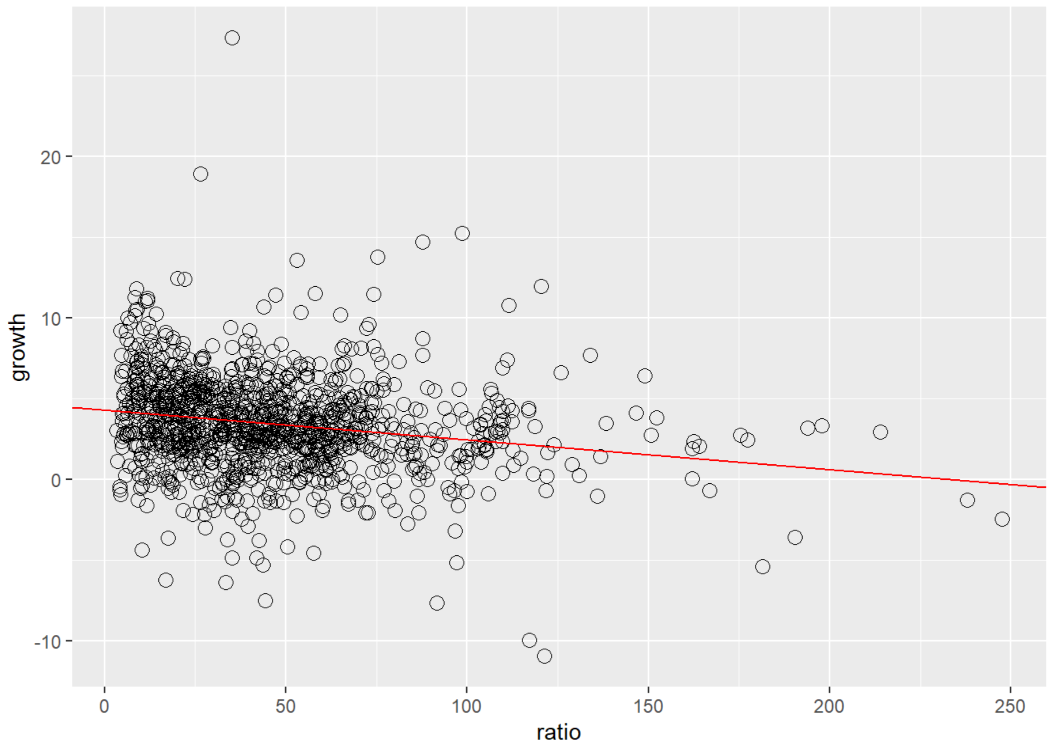
```
cat("Intercept is:", debt.lm$coefficients[1])
```

```
## Intercept is: 4.27929
```

```
cat("Slope is:", debt.lm$coefficients[2])
```

```
## Slope is: -0.01835518
```

```
ggplot(debt.lm) +
  geom_point(mapping = aes(x=ratio, y = growth), shape=1, size=3)+
  geom_abline(intercept = debt.lm$coefficients[1], slope = debt.lm$coefficients[2],
            color="red")
```




```
ggplot(debt) +
  geom_point(mapping = aes(x=ratio, y = growth), shape=3, size=2, color="blue")+
  geom_abline(intercept = debt.lm$coefficients[1], slope = debt.lm$coefficients[2],
            color="red")
```

#5. There should be four countries with a correlation smaller than -0.5. Separately, plot GDP growth versus debt ratio from each of these four countries and put the country names in the titles. This should be four plots. Call par(mfrow=c(2,2)) before plotting so all four plots will appear in the same figure. (Think about what this shows: individual relationships at the country level are sometimes concealed or "smudged out" when data is aggregated over all groups (countries). This conveys the importance of careful analysis at a more granular group level, when such groupings are available!)

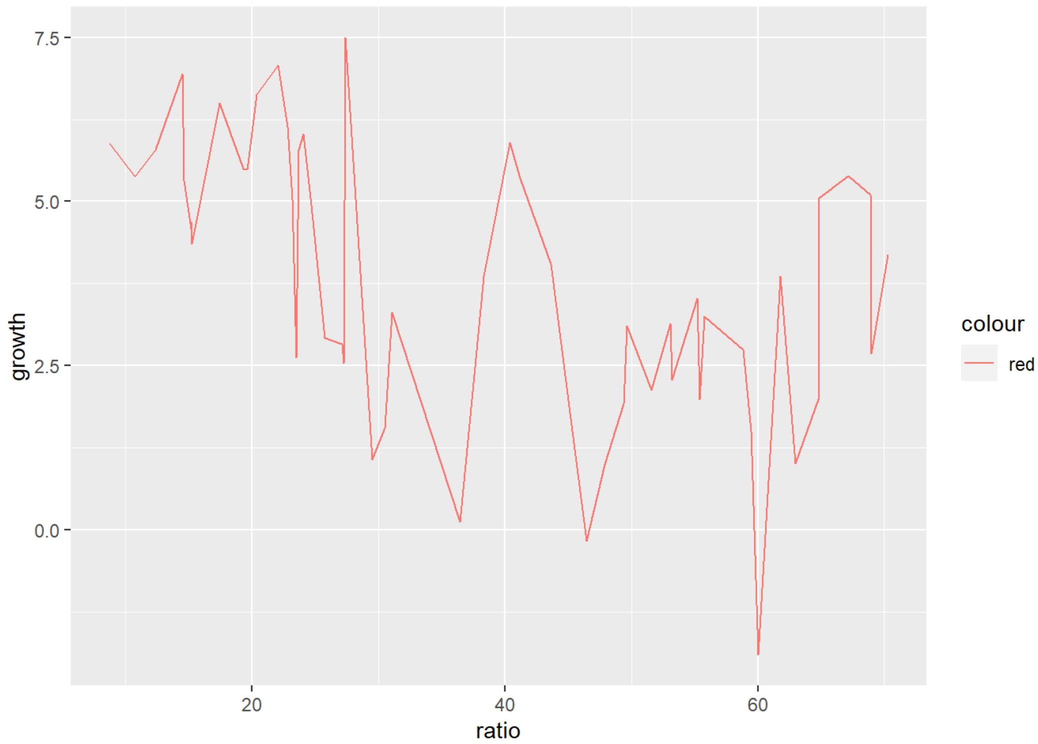
```
min <- which(all.correlations$all.correlations < -0.5)


all.correlations[which(all.correlations$all.correlations < -0.5), ]
```

```
## [1] -0.5019220 -0.5763191 -0.6447261 -0.7018506
```

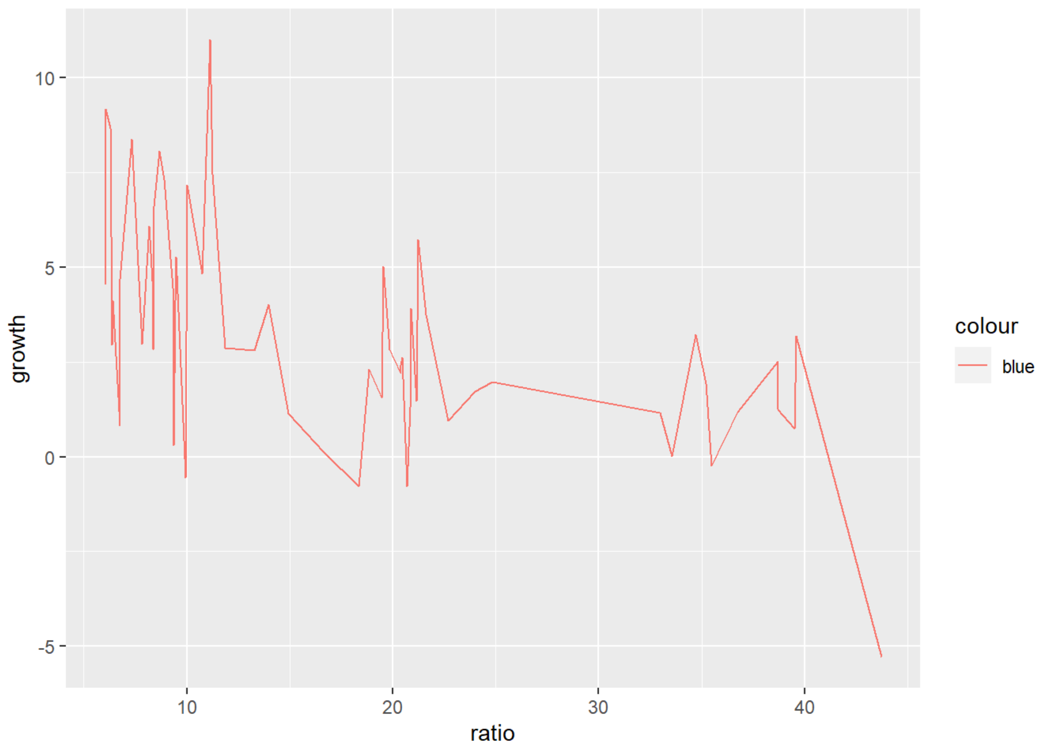
```
levels(factor(debt$Country))[min]
```

```
## [1] "France"  "Germany" "Italy"   "Japan"
```

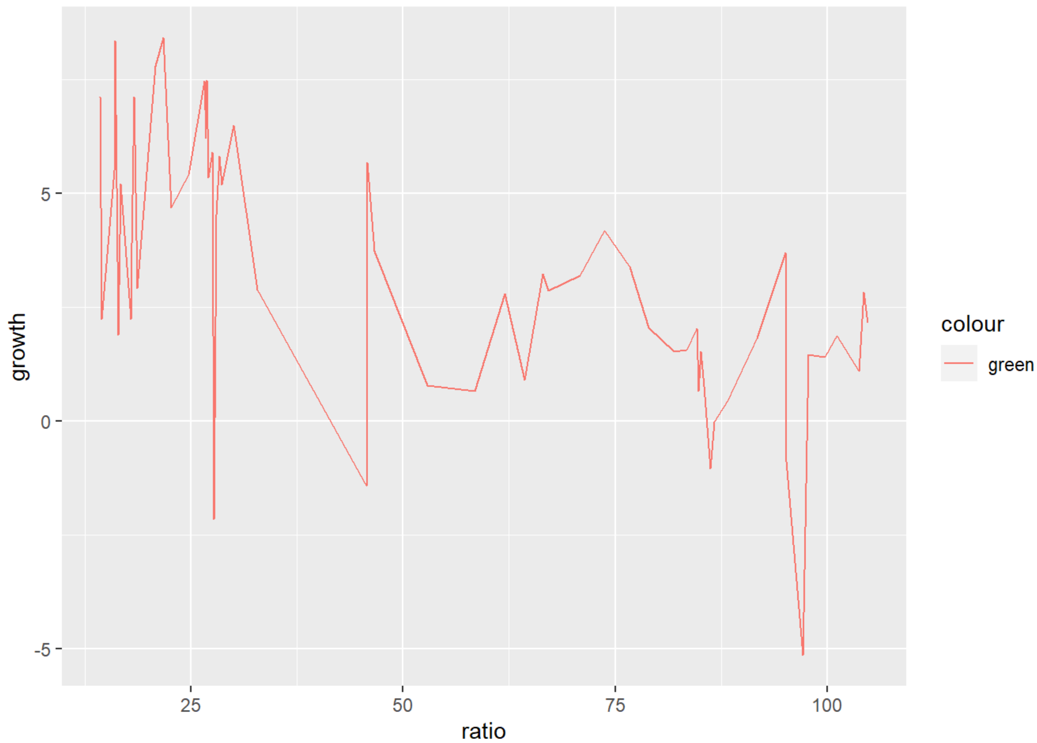
```
par(mfrow=c(2,2))


#ggplot(debt) +
#  geom_point(mapping = aes(x=debt[Country =="France","growth"], y = debt[Country =="France","ratio"]), shape=1, # size=3)


fr <- filter(debt, Country == "France")
germ <- subset(debt, Country == "Germany")
it <- subset(debt, Country == "Italy")
jap <- subset(debt, Country == "Japan")




par(mfrow=c(2,2))

ggplot(data = fr) +
  geom_line(mapping = aes(x=ratio, y = growth, color="red"))
```



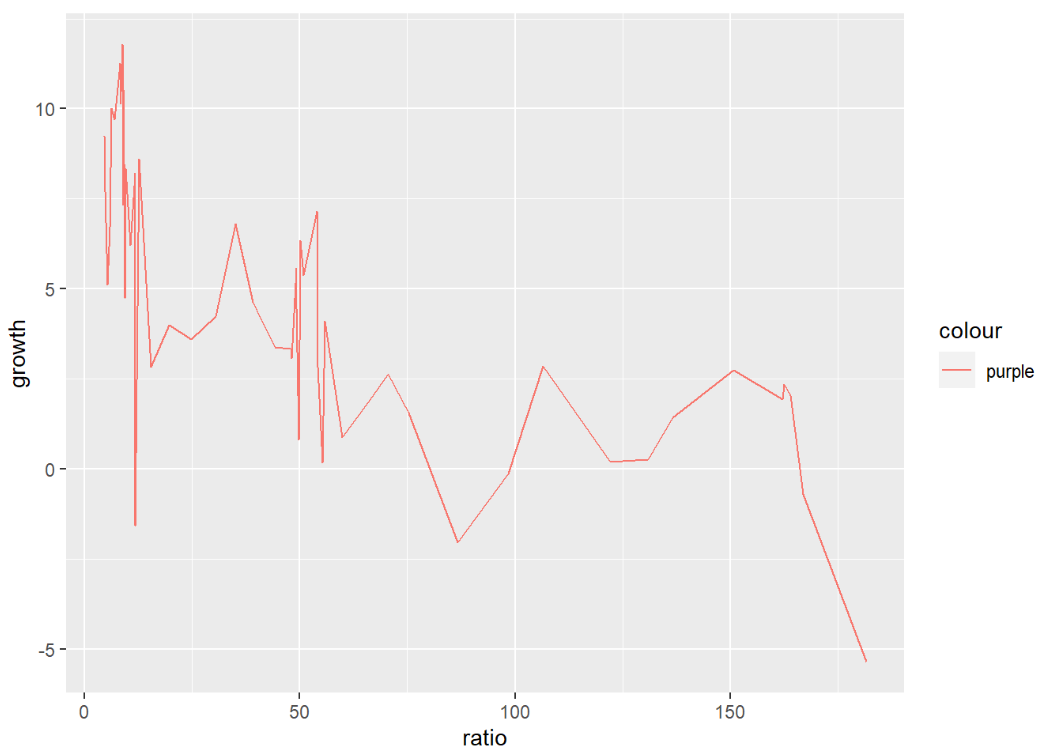

```
ggplot(data = germ) +
  geom_line(mapping = aes(x=ratio, y = growth, color="blue"))
```



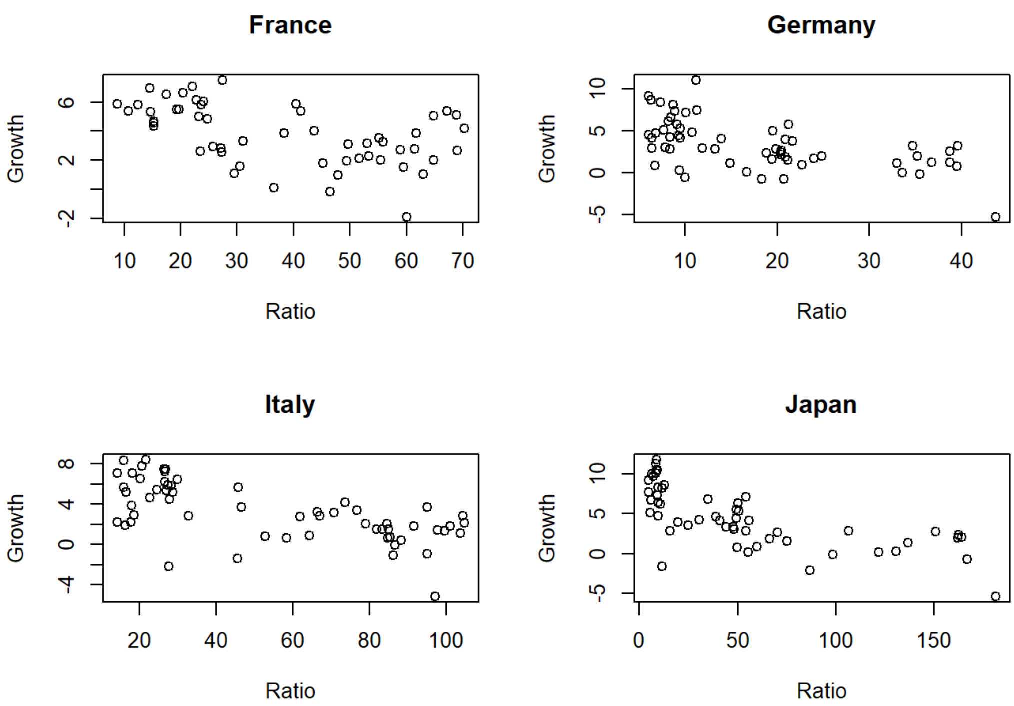

```
ggplot(data = it) +
  geom_line(mapping = aes(x=ratio, y = growth, color="green"))
```




```
ggplot(data = jap) +
  geom_line(mapping = aes(x=ratio, y = growth, color="purple"))
```



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


plot(fr$ratio, fr$growth, xlab="Ratio",
  ylab="Growth", main = "France")


plot(germ$ratio, germ$growth, xlab="Ratio",
  ylab="Growth", main = "Germany")

plot(it$ratio, it$growth, xlab="Ratio",
  ylab="Growth", main = "Italy")

plot(jap$ratio, jap$growth, xlab="Ratio",
  ylab="Growth", main = "Japan")
```



#6. Some economists claim that high levels of government debt cause slower growth. Other economists claim that low economic growth leads to higher levels of government debt. The data file, as given, lets us relate this year's debt to this year's growth rate; to check these claims, we need to relate current debt to future growth.

#a. Create a new data frame which just contains the rows of debt for France, but contains all those rows. It should have 54 rows and 4 columns (print the dimensions of your data frame). Note that some years are missing from the middle of this data set.

```
# I already created this dataset in the last question, called "fr"

head(fr)
```

```
##   Country Year   growth    ratio
## 1  France 1950 7.494005 27.41989
## 2  France 1951 6.134969 22.84359
## 3  France 1952 2.627430 23.49749
## 4  France 1953 2.918587 25.78166
## 5  France 1954 4.825871 24.76863
## 6  France 1955 5.790223 23.70047
```

```
# Another way to do it is using pipe:


dim(fr)
```

```
## [1] 54  4
```

#b. Create a new column in your data frame for France, next.growth, which gives next year’s growth if the next year is in the data frame, or NA if the next year is missing.

(next.growth for 1971 should be (rounded) 5.886, but for 1972 it should be NA. Print these two values.)

```
fr <-  mutate(fr, next.growth <- ifelse((lead(Year) - Year) == 1, lead(growth), NA))


fr %>%
  filter(Year == 1971) %>%
  .[,5]
```

```
## [1] 5.885827
```

```
fr %>%
  filter(Year == 1972) %>%
  .[,5]
```

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

#7. Add a next.growth column, as in the previous question, to the whole of the debt data frame. Make sure that you do not accidentally put the first growth value for onecountry as the next.growth value for another.

Hints: Write a function to encapsulate what you did in the previous question, and apply it using ddply()

```
# Here we use ddply on the debt dataset
# we break up the dataset via different Countries
# we apply the mutate function
# we add a column called next.growth whose value depends on the ifelse statement


newdebt <- ddply(debt, .(Country), mutate, next.growth = ifelse((lead(Year) - Year) == 1, lead(growth), NA))

head(newdebt)
```

```
##     Country Year    growth     ratio next.growth
## 1 Australia 1946 -3.557951 190.41908   2.4594746
## 2 Australia 1947  2.459475 177.32137   6.4375341
## 3 Australia 1948  6.437534 148.92981   6.6119938
## 4 Australia 1949  6.611994 125.82870   6.9202012
## 5 Australia 1950  6.920201 109.80940   4.2726115
## 6 Australia 1951  4.272612  87.09448   0.9046516
```

#The next.growth for France in 2009 should be NA, not 9.167. Print this value.

```
newdebt %>%
  filter(Country == "France", Year == 2009) %>%
  .[,5]
```

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

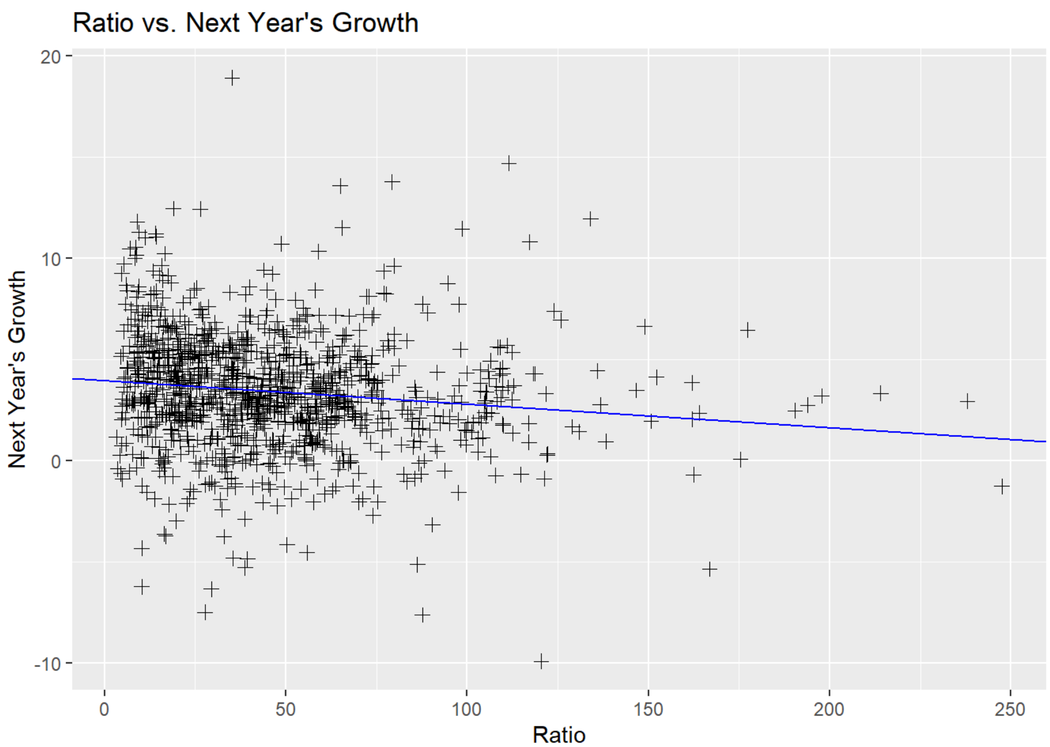
#8. Make a scatter-plot of next year's GDP growth against this year's debt ratio. Linearly regress next year's growth rate on the current year's debt ratio, and add the line to the plot. Report the intercept and slope to reasonable precision. How do they compareto the regression of the current year's growth on the current year's debt ratio?

```
lm1 <- lm(next.growth~ratio, data = newdebt)
lm1
```

```
##
## Call:
## lm(formula = next.growth ~ ratio, data = newdebt)
##
## Coefficients:
## (Intercept)        ratio
##     3.92472     -0.01161
```

```
ggplot(newdebt) +
    geom_point(mapping = aes(y=next.growth, x = ratio), shape=3, size=2) +
    geom_abline(intercept = coef(lm1)[1], slope = coef(lm1)[2], color="blue") +
    labs(title = "Ratio vs. Next Year's Growth", x = "Ratio", y = "Next Year's Growth")   # labs for labels
```

```
## Warning: Removed 24 rows containing missing values (geom_point).
```




```
lm2 <- lm(growth ~ ratio, data = newdebt)
lm2
```

```
##
## Call:
## lm(formula = growth ~ ratio, data = newdebt)
##
## Coefficients:
## (Intercept)        ratio
##     4.27929     -0.01836
```

The regression of this year's growth vs. ratio compared with the regression of next year's growth vs. ratio is similar; both have negative slopes.

The former has a slope of -0.01836 and the latter has a slope of -0.01161. Interpretation: as the ratio increases, there is a stronger reaction by this year's growth as compared with the reaction by next year's growth when we look at absolute values of slopes.

In other words, a change in the ratio affects this year's growth more than it affects next year's growth.

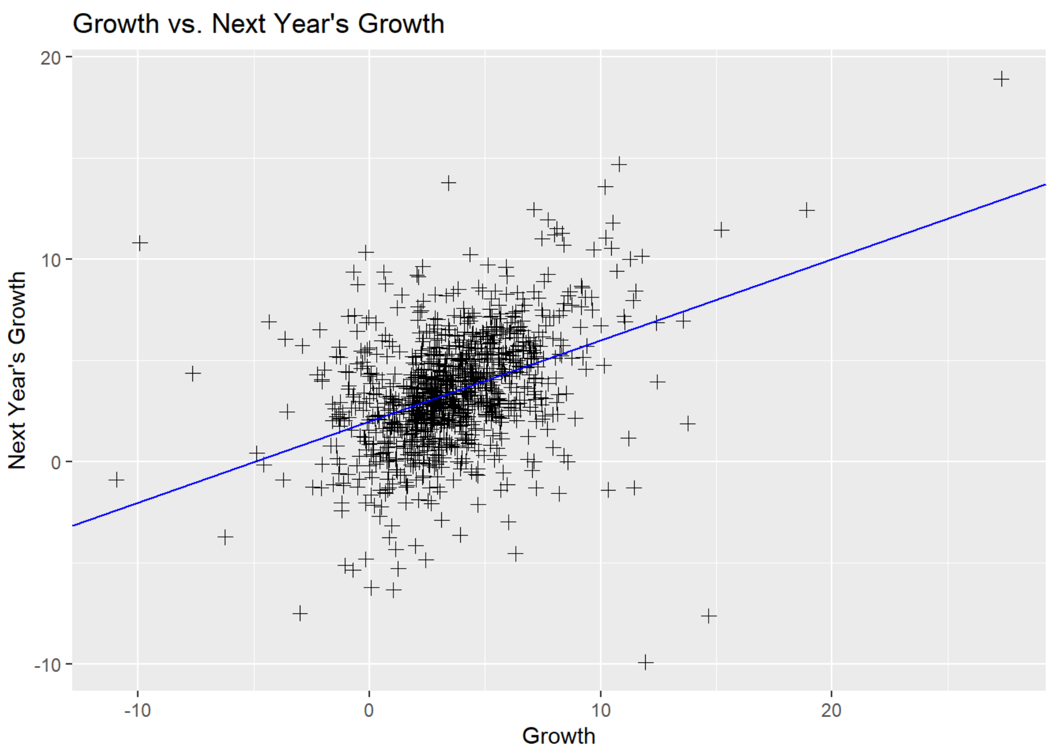
#9. Make a scatter-plot of next year's GDP growth against the current year's GDP growth. Linearly regress next year's growth on this year's growth, and add the line to the plot. Report the coefficients. Can you tell, from comparing these two simple regressions (from the current question, and the previous), whether current growth or current debt is a better predictor of future growth?

```
lm3 <- lm(next.growth ~ growth, data = newdebt)
lm3
```

```
##
## Call:
## lm(formula = next.growth ~ growth, data = newdebt)
##
## Coefficients:
## (Intercept)       growth
##      1.9711       0.4007
```

```
ggplot(newdebt) +
    geom_point(mapping = aes(y=next.growth, x = growth), shape=3, size=2) +
    geom_abline(intercept = coef(lm3)[1], slope = coef(lm3)[2], color="blue") +
    labs(title = "Growth vs. Next Year's Growth", x = "Growth", y = "Next Year's Growth")   # labs for labels
```

```
## Warning: Removed 24 rows containing missing values (geom_point).
```



Next year's growth vs. this year's growth has a stronger relationship than next year's growth vs. current debt. We can deduce this from the slop of the former, which is 0.4007, as compared with the slope of the latter, -0.012. Not only is the slope positive, indicating a positive relationship between growth this year and next year, but also the absolute value is larger, indicating an overall stronger relationship.