Homework 7 – Due May 5

Name: Navya Mittal netID: navya_0215 Collaborated with: No one

Your homework **must be submitted in Word or PDF format, created by calling "Knit Word" or "Knit PDF" from RStudio on your R Markdown document. Submission in other formats may receive a grade of 0**. Your responses must be supported by both textual explanations and the code you generate to produce your result. Note that all R code used to produce your results must be shown in your knitted file.

Gross domestic product (GDP) is a measure of the total market value of all goods and services produced in a given country in a given year. The percentage growth rate of GDP in year t is

$$100 \times \left(\frac{GDP_{t+1} - GDP_t}{GDP_t} \right) - 100$$

An important claim in economics is that the rate of GDP growth is closely related to the level of government debt, specifically with the ratio of the government's debt to the GDP. The file `debt.csv` contains measurements of GDP growth and of the debt-to-GDP ratio for twenty countries around the world, from the 1940s to 2010. Note that not every country has data for the same years, and some years in the middle of the period are missing data for some countries but not others.

This data is also used and discussed in Lab 10. In this homework, we will use the package `dplyr` for some of the data manipulation and `ggplot2` for visualization.

1. Load the data into a **tibble** named `debt` and make a scatter-plot of the GDP growth rate (vertical axis) against the debt ratio (horizontal axis) with `ggplot2` and appropriate axis titles.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
?tibbles
```

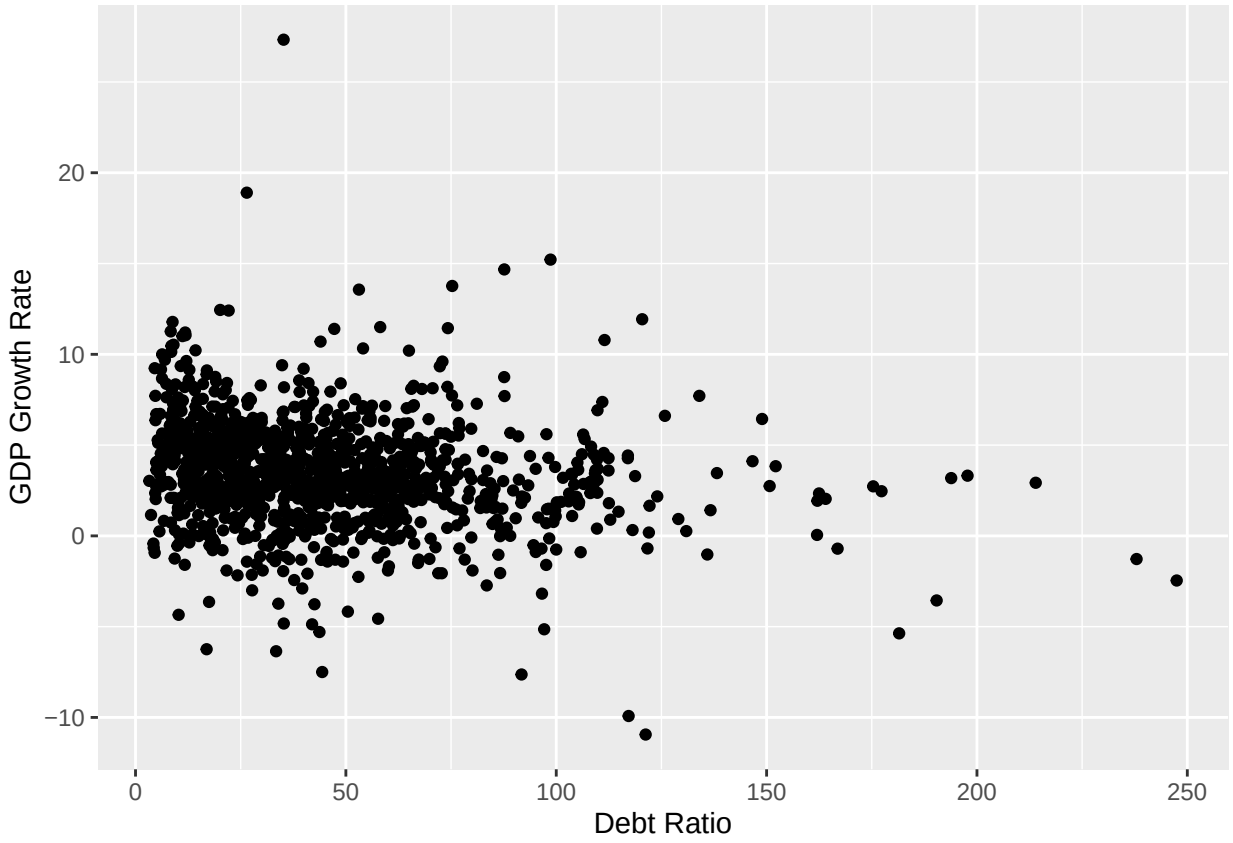
```
## No documentation for 'tibbles' in specified packages and libraries:
## you could try '??tibbles'
```

```
loading <- read_csv("debt.csv")
```

```
## Rows: 1171 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Country
## dbl (3): Year, growth, ratio
```

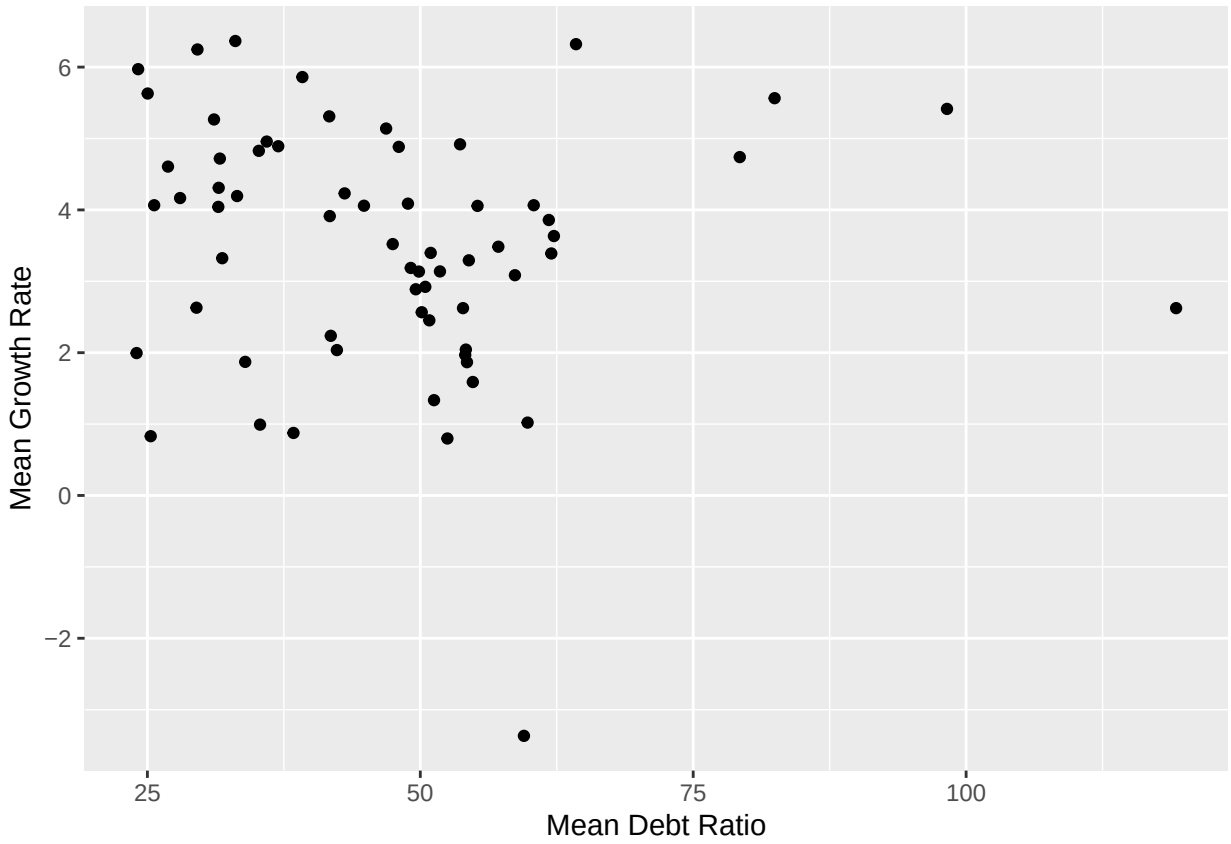
```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
debt <- as_tibble(loading)
ggplot(debt, aes(ratio,growth)) +
  geom_point() +
  xlab("Debt Ratio") +
  ylab("GDP Growth Rate")
```



2. Use `group_by()` and `summarise()` to compute the mean growth rate and mean debt ratio for each **year** in the data set. Plot the results as a scatter-plot of the mean GDP growth rate (vertical axis) against the mean debt ratio (horizontal axis) with `ggplot2` and appropriate axis titles. You only have to submit the code and the plot.

```
meandebt <- debt %>%
  group_by(Year) %>%
  summarise(meangrowth = mean(growth), meanratio = mean(ratio))

ggplot(meandebt, aes(x= meanratio, y= meangrowth)) +
  geom_point() +
  xlab("Mean Debt Ratio") +
  ylab("Mean Growth Rate")
```



3. Fit a linear model of growth on the debt ratio, using `lm()`. Report the intercept and slope. Add a line to your scatterplot from Q1 showing the fitted regression line. (You may have to redraw the plot in Q1 here in order to add a line.)

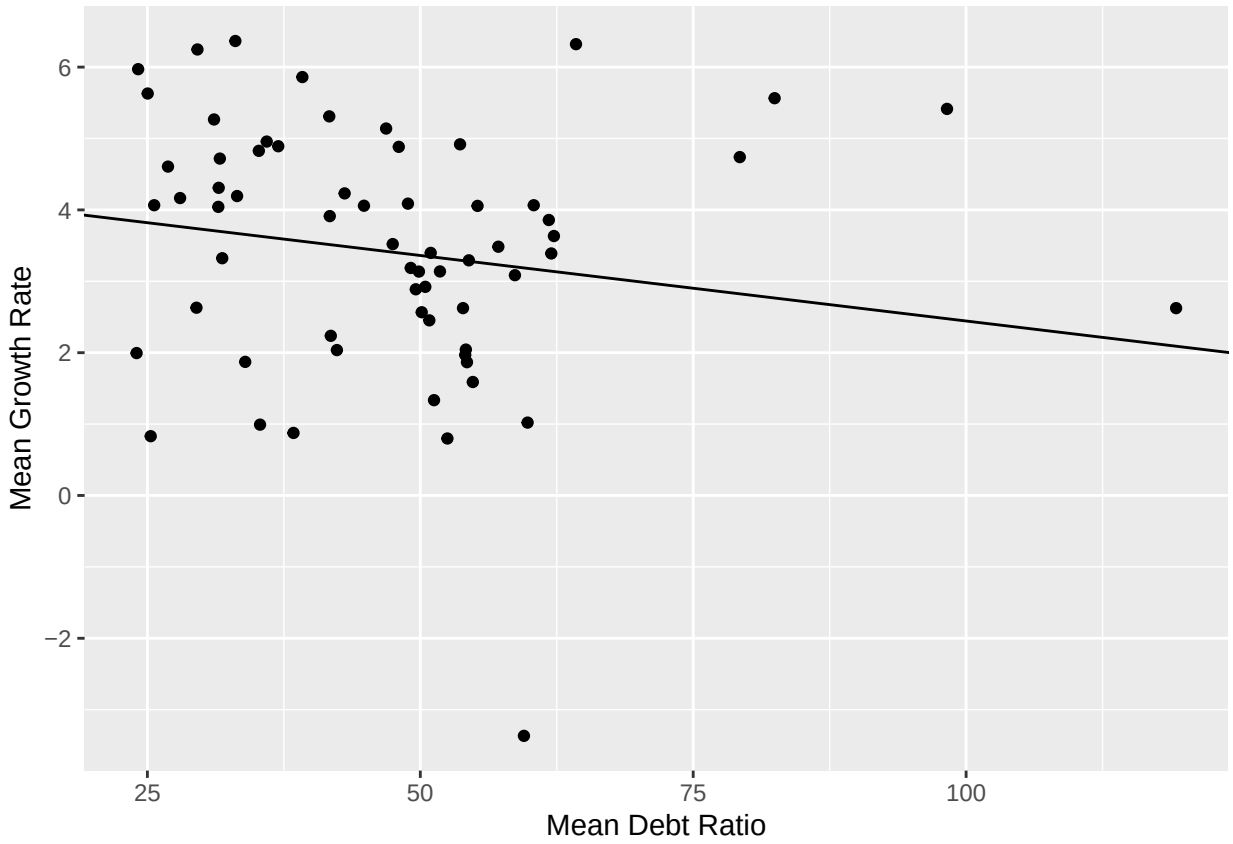
```
# Fit linear model
lm_fit <- lm(growth ~ ratio, data = debt)

# Print intercept and slope
summary(lm_fit)$coefficients
```

```
##               Estimate  Std. Error   t value      Pr(>|t|)
## (Intercept)  4.27929049 0.148970020 28.725850 9.661475e-138
## ratio       -0.01835518 0.002637314 -6.959801  5.665747e-12
```

```
lm_fit$coefficients[2]
```

```
##       ratio
## -0.01835518
```

```
ggplot(meandebt, aes(x= meanratio, y= meangrowth)) +
  geom_point() +
  xlab("Mean Debt Ratio") +
  ylab("Mean Growth Rate") +
  geom_abline(intercept = lm_fit$coefficients[1], slope = lm_fit$coefficients[2])
```




```
?ggplot
```

4. a. Using `filter()`, create a new tibble named `France.debt` which just contains the rows of `debt` for France. It should have 54 rows and 4 columns.
   b. Create a new column named `next.growth` and this column 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`.) Use `mutate()`. `lead()` may be useful here, check it's help page. Replace the tibble `France.debt` by the new tibble with both the original columns and the new column `next.growth`. Print the rows corresponding to Years 1971, 1972, 1990 and 2009 using `filter()`.

```
?mutate
France.debt <- filter(debt, Country == "France")

France.debt <- France.debt %>%
  mutate(next.growth = ifelse((Year + 1) %in% Year, round(lead(growth), 3), NA))

France.debt %>% filter(Year %in% c("1971", "1972","1990","2009"))
```

```
## # A tibble: 4 x 5
##   Country  Year growth ratio next.growth
##   <chr>   <dbl>  <dbl> <dbl>       <dbl>
## 1 France   1971   5.37 10.8         5.89
## 2 France   1972   5.89  8.76       NA
## 3 France   1990   1.93 49.4         3.10
## 4 France   2009  -1.91 60.0        NA
```

5. Add a `next.growth` column, as in Q4, to the **whole** `debt` tibble. (The `next.growth` for France in 2009 should be `NA`, not 9.167.) Note that the tibble `debt` should be replaced by the new tibble with both the

original columns and the new column `next.growth`. Print two rows using `filter()`: (1) Year 2006 of Austria and (2) Year 1990 of UK.

```
debt <- debt %>%
  mutate(next.growth= ifelse((Year + 1) %in% Year, round(lead(growth), 3), NA))

debt %>% filter(Country == "Austria" & Year == 2006 | Country == "UK" & Year == 1990)
```

```
## # A tibble: 2 x 5
##   Country  Year growth ratio next.growth
##   <chr>   <dbl>  <dbl> <dbl>       <dbl>
## 1 Austria  2006  3.25   56.7        15.2
## 2 UK       1990  0.779  33.8        -1.39
```

6. Linearly regress next year's growth rate on the current year's debt ratio. Using `ggplot2`, make a scatter-plot of next year's GDP growth against this year's debt ratio, and add the fitted regression line to the plot. Report the intercept and slope. How do the slope compare to the slope from the regression of the current year's growth on the current year's debt ratio?

```
next_growth_model <- lm(next.growth ~ ratio, data = debt)
summary(next_growth_model)$coefficients
```

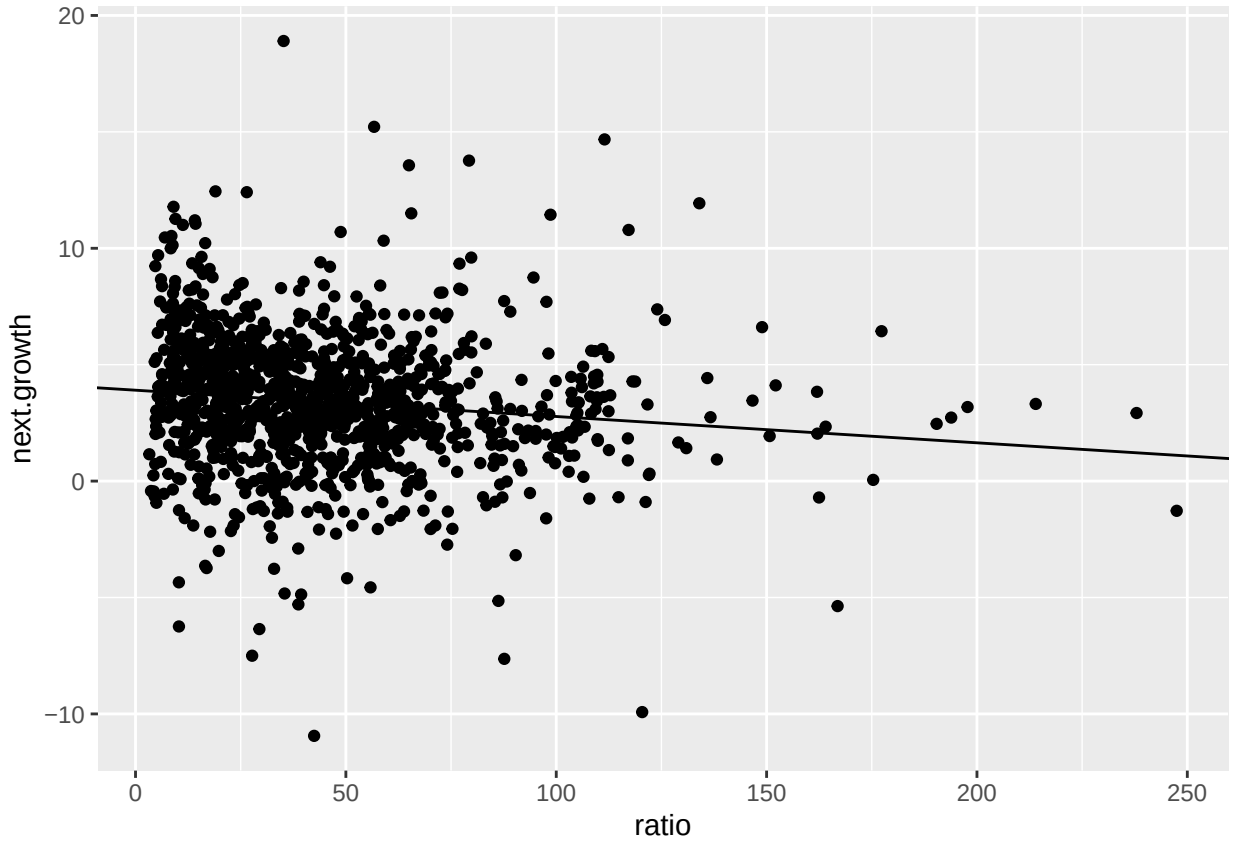
```
##                Estimate  Std. Error   t value      Pr(>|t|)
## (Intercept)  3.90357666 0.146131662 26.712737 1.023392e-122
## ratio       -0.01129811 0.002599578 -4.346133  1.507973e-05
```

```
next_growth_model$coefficients[2]
```

```
##       ratio
## -0.01129811
```

```
ggplot(debt, aes(x= ratio, y= next.growth)) +
  geom_point() +
  geom_abline(intercept = next_growth_model$coefficients[1], slope = next_growth_model$coefficients[2])
```

```
## Warning: Removed 18 rows containing missing values (`geom_point()`).
```



The slope has a lower value (~-0.0113) compared to the slope for the current year's growth on the current year's debt ratio model (-0.018)

7. Add a column, `delta.growth`, to the `debt` tibble, giving the difference between next year's GDP growth rate and this year's GDP growth rate (i.e., next year's growth - this year's growth). Note that the tibble `debt` should be replaced by the new tibble with both the original columns (including `next.growth` due to Q5) and the new column `delta.growth`. Then regress the change in GDP growth on the current GDP growth and the current debt ratio. Report the coefficients.

```
debt <- debt %>% mutate(delta.growth = next.growth - growth)
deltareg <- lm(delta.growth ~ growth + ratio, data = debt)
deltareg$coefficients
```

```
##  (Intercept)       growth        ratio
##  2.200342339 -0.606600139 -0.004521886
```