Homework 2

Mingang Kim

2021 9 9

1.

2.

A.

1. Learn how to code neatly.
2. Improve visualization skill.
3. Code Latex without looking at cheat sheets.

B.

$$P(X = x|N) = \frac{1}{N}, x = 1, 2, 3, .., N \tag{1}$$

$$P(Y = y|n, p) = \binom{n}{y} p^y (1 - p)^{n-y}, y = 0, 1, 2, ..., n \tag{2}$$

$$P(X = x|r, p) = \binom{x-1}{r-1} p^r (1 - p)^{x-r}, x = r, r + 1, ..., \tag{3}$$

3.

1. Rule 1: For Every Result, Keep Track of How It Was Produced
   - There are many interrelated steps making raw data into the final result.
   - It is important to record a name and version of the program, as well as the exact parameters and inputs that were used for reproducible research.
   - Analysis can be reproduced by specifying the full analysis workflow in a form that allows for direct execution.
   - Challenges: How to record the whole steps briefly and exactly in a way that allows perfect sync with its final version.
2. Rule 2: Avoid Manual Data Manipulation Steps
   - Avoid relying on manual procedures to modify data because it is not efficient and easy to make error which makes reproduction difficult.
   - If manual operations are inevitable, note down which data files were modified or moved, and the purpose of execution as briefly as you can.

- Challenges: Before recording manually, how to reproduce the analyses with manual procedures should be considered.

3. Rule 3: Archive the Exact Versions of All External Programs Used
   - It is important to archive the exact versions of programs to avoid unnecessary hassle.
   - Codes can only be executed in a specific version of program.
   - Challenges: It is easy to think that the version is not important and newest version is good.
4. Rule 4: Version Control All Custom Scripts
   - In some cases, it is necessary to track down exact reproduction of results.
   - Use Subversion, Git, or Mercurial to version control.
   - Challenges: Update the newest version of code on a regular basis and should not forget about version control.
5. Rule 5: Record All Intermediate Results, When Possible in Standardized Formats
   - Intermediate results can reveal discrepancies and it will be a good way to uncover bugs or faulty interpretations.
   - It is possible to rerun the parts when the full process is not readily executable.
   - It is possible to find on which steps problems appears.
   - Without executing the full process, it is possible to examine the full process.
   - Challenges: Archive intermediate result files will require a lot of storage.
6. Rule 6: For Analyses That Include Randomness, Note Underlying Random Seeds
   - If random seed is not fixed, the same program will give different result every time.
   - Thus, random seed should be recorded if analysis steps involve randomness.
   - Challenges: There is no challenge but some codes can be run on specific random seeds. So the analyzer should keep in mind that the analyze can be reproduced in specific condition.
7. Rule 7: Always Store Raw Data behind Plots
   - If the raw data behind figures are stored in a systematic manner, given figures can be easily retrieved.
   - If one really wants to read fine values in a figure, it is possible to consult the raw numbers.
   - Challenges: Raw data should be saved for backup before using it because data can be modified by mistake.
8. Rule 8: Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected
   - It is not practical to incorporate various debug outputs in the source code of scripts and programs.
   - Rather than that, hypertext is useful to inspect the detailed values underlying the summaries.
   - Challenges: Summarizing results to be generated along with links that can be difficult and take a lot of time.
9. Rule 9: Connect Textual Statements to Underlying Results
   - Results and interpretations are located in different area and forms.
   - Making this connection when it is needed may not be easy and error-prone.
   - To allow efficient retrieval, tools such as Sweave are recommended.
   - Challenges: Locating the exact result underlying and supporting the statement will not be easy when it comes to a large pool of different analyses with various versions.

10. Rule 10: Provide Public Access to Scripts, Runs, and Results
    - All input data, scripts, versions, parameters, and intermediate results should be made publicly and easily accessible for trustworthiness, and transparency.
    - Challenges: Some people will hesitate to share their code and data because they think it is their property and achievement.

4.

```
library(data.table)
```

```
## Warning:    'data.table'  R   4.1.1
```

```
covid_raw <- fread("https://opendata.ecdc.europa.eu/covid19/casedistribution/csv")
us <- covid_raw[covid_raw$countriesAndTerritories == 'United_States_of_America',]
us_filtered <- us[us$month %in% c(6:7),]
us_filtered$index <- rev(1:dim(us_filtered)[1])
fit<-lm(`Cumulative_number_for_14_days_of_COVID-19_cases_per_100000`~index, data=us_filtered)
```

Part a.

1. Create a summary table of the us_filtered data. Hint, use summary and kable in the knitr package. How many time points have we limited ourselves to? Are there missing values?

```
library("knitr")
```

```
## Warning:    'knitr'  R   4.1.1
```

```
kable(summary(us_filtered), "simple")
```

dateRep	day	month	year	cases	deaths	countriesAndTerr
Length:61	Min. : 1.00	Min. :6.000	Min. :2020	Min. :18665	Min. : 242.0	Length:61
Class :character	1st Qu.: 8.00	1st Qu.:6.000	1st Qu.:2020	1st Qu.:25540	1st Qu.: 500.0	Class :character
Mode :character	Median :16.00	Median :7.000	Median :2020	Median :45221	Median : 767.0	Mode :character
NA	Mean :15.75	Mean :6.508	Mean :2020	Mean :44666	Mean : 791.6	NA
NA	3rd Qu.:23.00	3rd Qu.:7.000	3rd Qu.:2020	3rd Qu.:61796	3rd Qu.: 982.0	NA
NA	Max. :31.00	Max. :7.000	Max. :2020	Max. :78427	Max. :2437.0	NA

```
summary(is.na(us_filtered))
```

```
##   dateRep           day             month             year
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:61        FALSE:61        FALSE:61        FALSE:61
##    cases            deaths        countriesAndTerritories   geoId
##  Mode :logical   Mode :logical   Mode :logical           Mode :logical
##  FALSE:61        FALSE:61        FALSE:61                FALSE:61
```

```
##  countryterritoryCode popData2019     continentExp
##  Mode :logical        Mode :logical   Mode :logical
##  FALSE:61             FALSE:61        FALSE:61
##  Cumulative_number_for_14_days_of_COVID-19_cases_per_100000   index
##  Mode :logical                                               Mode :logical
##  FALSE:61                                                    FALSE:61
```

We have 3 time points. Year, Month and day. There is no missing value in the data.

```r
library(stargazer)
```

```
##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```r
stargazer(fit, title="Result",
          single.row = TRUE, type ="latex", header = FALSE)
```

Table 2: Result

	Dependent variable:
	‘Cumulative_number_for_14_days_of_COVID-19_cases_per_100000‘
index	4.107*** (0.145)
Constant	42.853*** (5.165)
Observations	61
R^2	0.932
Adjusted R^2	0.930
Residual Std. Error	19.922 (df = 59)
F Statistic	803.464*** (df = 1; 59)
Note:	*p<0.1; **p<0.05; ***p<0.01

b.

```r
library("broom")
```

```
## Warning:    'broom'  R   4.1.1
```

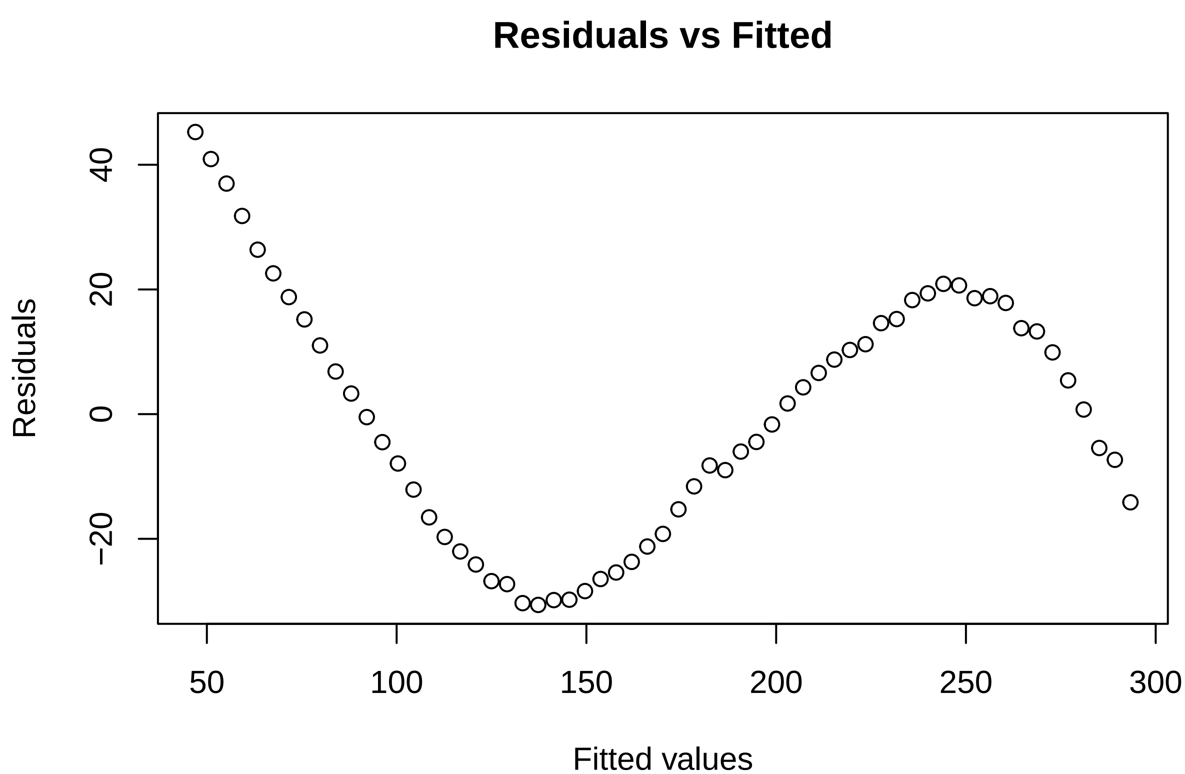
```r
fit.diags <- broom::augment(fit)
fit.diags
```

```
## # A tibble: 61 x 8
##    `Cumulative_number_fo~ index .fitted  .resid   .hat .sigma .cooksd .std.resid
##                     <dbl> <int>   <dbl>   <dbl>  <dbl>  <dbl>   <dbl>      <dbl>
```

```
## 1                       279.    61    293. -14.1    0.0640   20.0 1.84e-2    -0.734
## 2                       282.    60    289.  -7.33   0.0609   20.1 4.67e-3    -0.380
## 3                       280.    59    285.  -5.43   0.0579   20.1 2.42e-3    -0.281
## 4                       282.    58    281.   0.738  0.0549   20.1 4.22e-5     0.0381
## 5                       282.    57    277.   5.41   0.0521   20.1 2.14e-3     0.279
## 6                       283.    56    273.   9.90   0.0494   20.0 6.76e-3     0.510
## 7                       282.    55    269.  13.3    0.0469   20.0 1.14e-2     0.682
## 8                       278.    54    265.  13.8    0.0444   20.0 1.16e-2     0.708
## 9                       278.    53    260.  17.8    0.0420   20.0 1.83e-2     0.915
## 10                      275.    52    256.  18.9    0.0397   19.9 1.94e-2     0.969
## # ... with 51 more rows
```

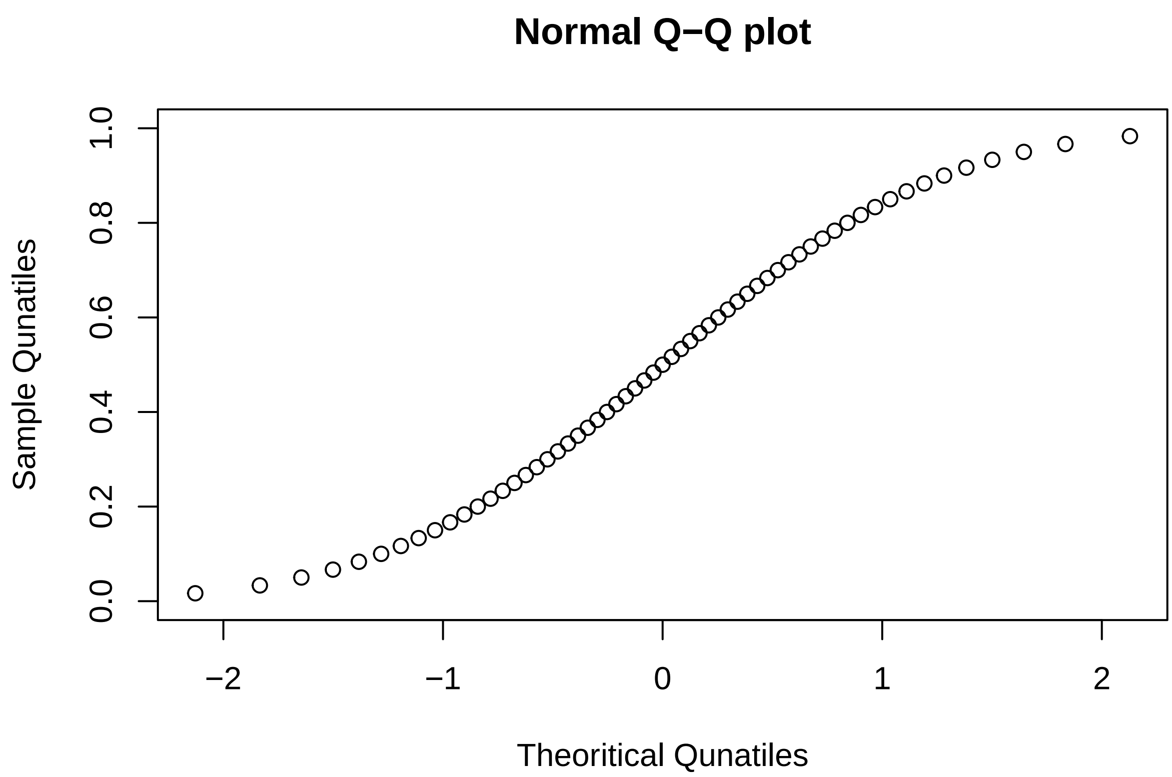
1. residuals vs fitted

```
fitted<-fit.diags$.fitted
residual<-fit.diags$.resid
plot(fitted,residual, xlab="Fitted values", ylab="Residuals", main="Residuals vs Fitted")
```



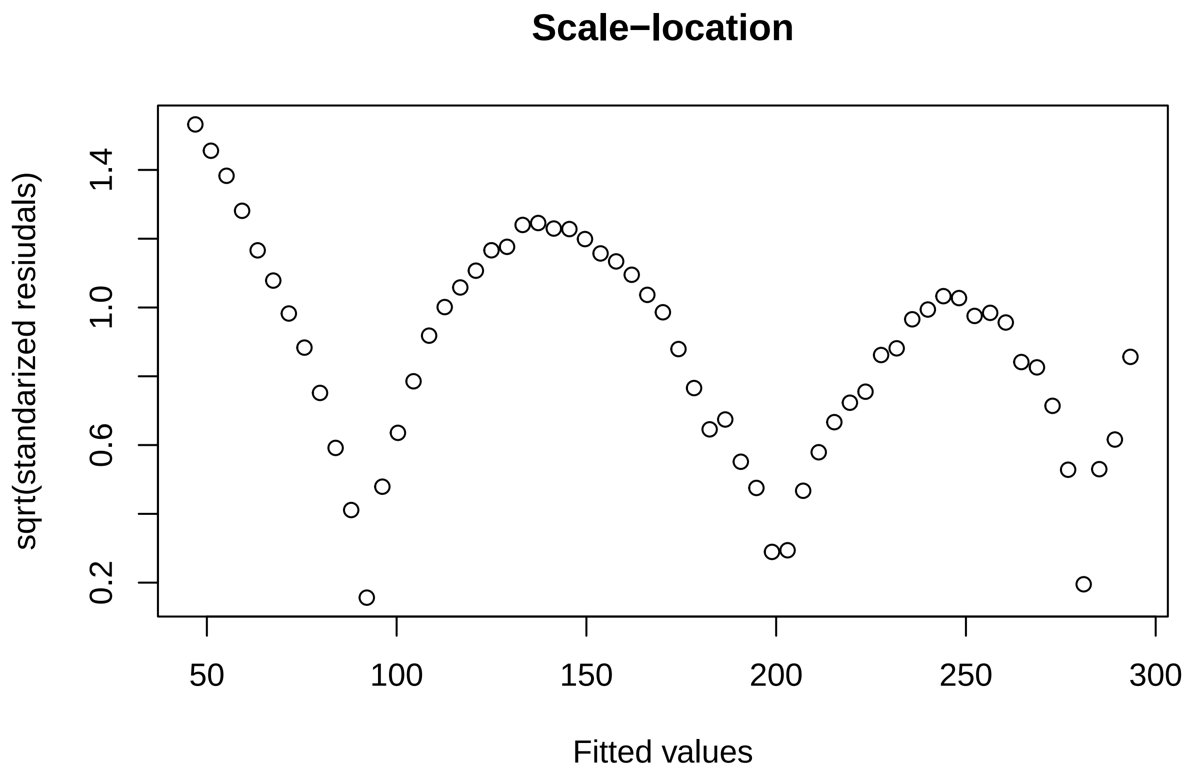
2. normal Q-Q

```
data.q<-quantile(fit.diags$.fitted, probs<-seq(0,1,length.out=length(fit.diags$.fitted)))
norm<-qnorm(probs, lower.tail = TRUE, log.p = FALSE)
plot(norm,probs, xlab="Theoritical Qunatiles", ylab="Sample Qunatiles", main = "Normal Q-Q plot")
```



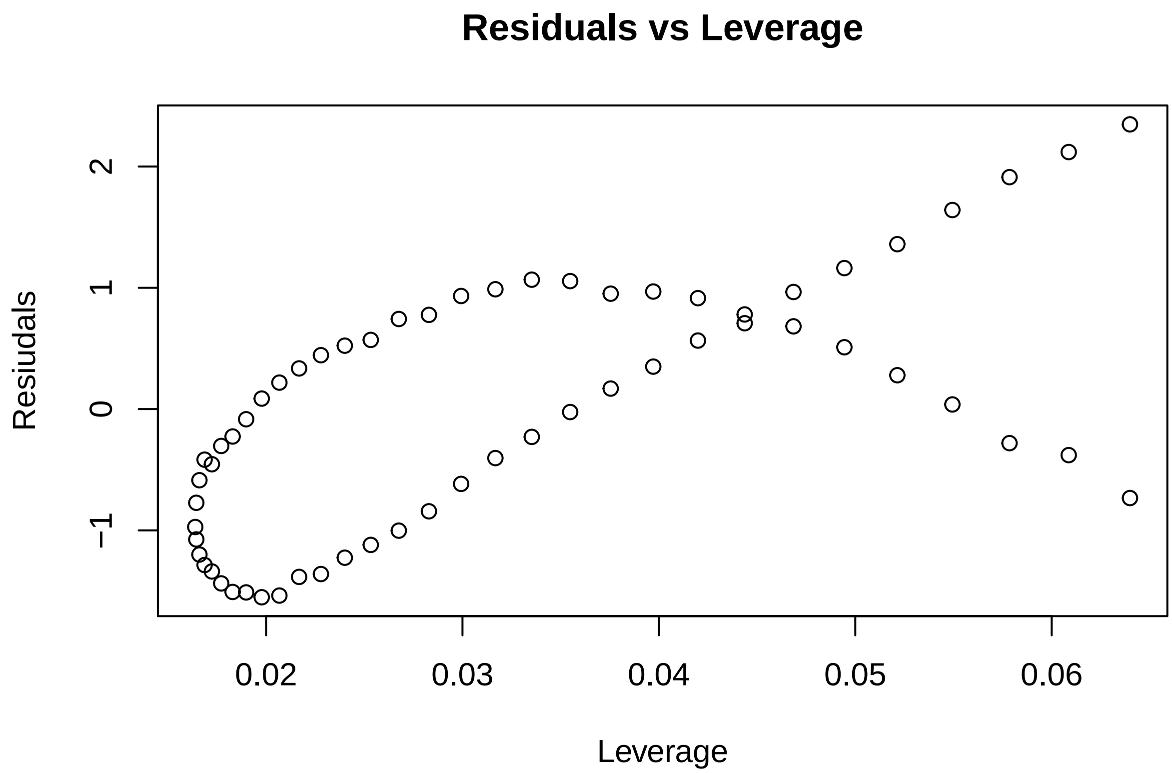
3. scale-location

```
fitted<-fit.diags$.fitted
std.r<-fit.diags$.std.resid
sq.std<-sqrt(abs(std.r))
plot(fitted,sq.std, xlab="Fitted values", ylab="sqrt(standarized resiudals)", main="Scale-location")
```



4. residuals vs leverage

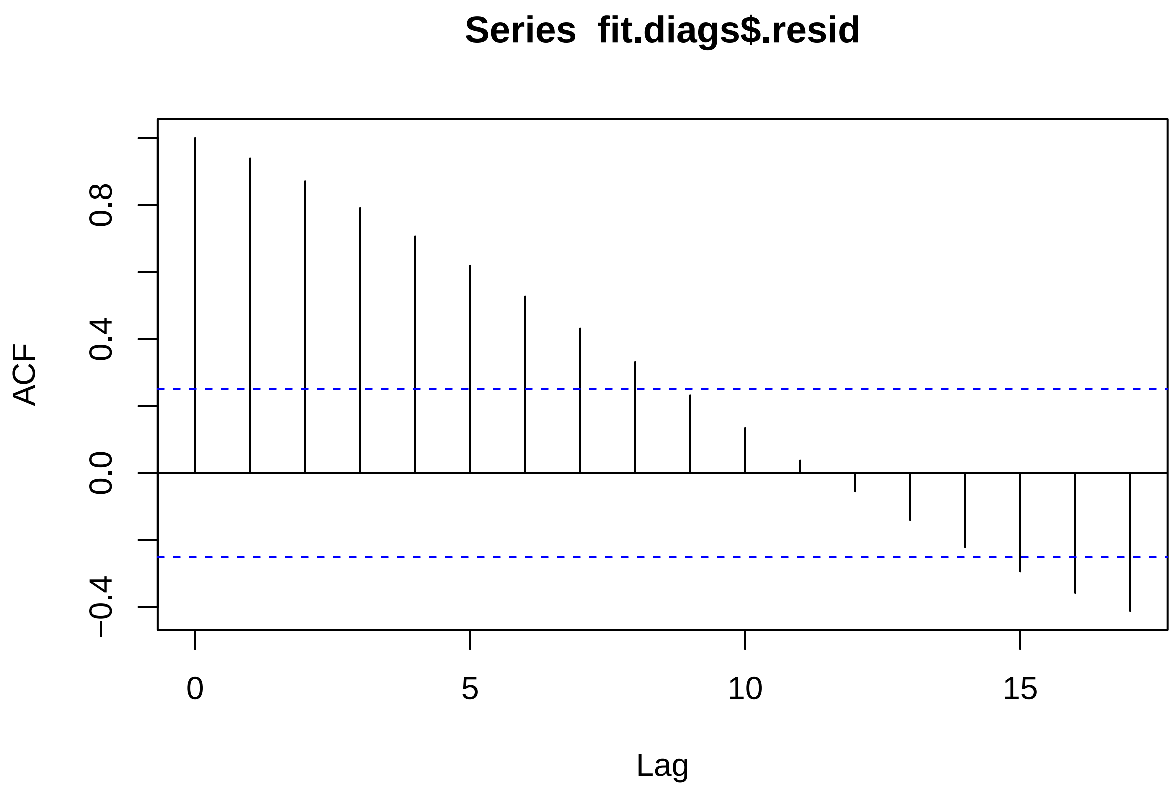
```
leverage<-fit.diags$.hat
std.r<-fit.diags$.std.resid
plot(leverage, std.r,  xlab="Leverage", ylab="Resiudals", main="Residuals vs Leverage")
```



c.

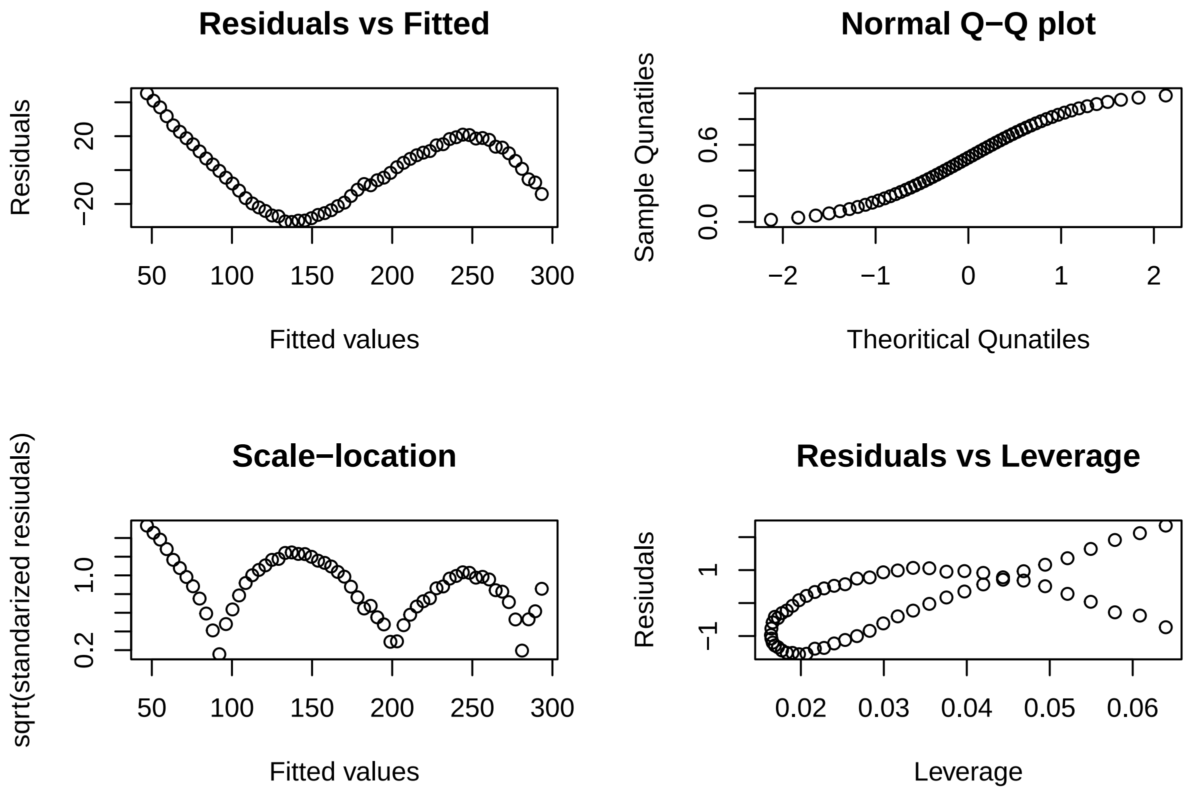
Create an auto correlation plot of the residuals using the acf function. The pattern observed is indicative of a time pattern. The simple linear model is not appropriate in this case.

```
acf(fit.diags$.resid)
```



5.

```
par(mfrow=c(2,2))
plot(fitted,residual, xlab="Fitted values", ylab="Residuals", main="Residuals vs Fitted")
plot(norm,probs, xlab="Theoritical Qunatiles", ylab="Sample Qunatiles", main = "Normal Q-Q plot")
plot(fitted,sq.std, xlab="Fitted values", ylab="sqrt(standarized resiudals)", main="Scale-location")
plot(leverage, std.r,  xlab="Leverage", ylab="Resiudals", main="Residuals vs Leverage")
```


Residuals vs Fitted
Residuals
−20
20
50
100
150
200
250
300
Fitted values
Normal Q−Q plot
Sample Qunatiles
0.0
0.6
−2
−1
0
1
2
Theoritical Qunatiles
Scale−location
sqrt(standarized resiudals)
0.2
1.0
50
100
150
200
250
300
Fitted values
Residuals vs Leverage
Resiudals
−1
1
0.02
0.03
0.04
0.05
0.06
Leverage