# stat 5014-homework 2

Mengkun Chen (mengkun21@vt.edu)

**Problem 2**

**Part A**

Specific learning objectives:

- Advanced tools to make proper plots
- Git for version control
- Parallel computing

**Part B**

Three density functions in centered format with equation number:

- Normal distribution: $N(0,1)$

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{1}$$

- Exponential distribution: $Exp(\beta)$

$$f(x) = \frac{1}{\beta} e^{-\frac{x}{\beta}} \tag{2}$$

- Gamma distribution: $Gamma(\alpha, \beta)$

$$f(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-\frac{x}{\beta}} \tag{3}$$

**Problem 3**

Steps in performing reproducible research:

1. Record every involved step, intermediate results, raw data behind plots that may influence the execution between raw data to final results. Track all these using version control system.

- Challenges: Documentation can easily get out of sync with how the analysis was really performed in final version.

2. Avoid manual data manipulation and rely on execution of programs if possible.

- Challenges: Sometimes manual data manipulation can't be avoid. It's tough to record detailed notes about manual changes.

3. Archive the exact versions of the programs actually used. Store a full virtual machine image of the operating system and program.

- Challenges: Even though the operating system is recorded, we should still modify the program if the program is operated on the other operating system or different version.

4. Set random seeds if randomness included in the analysis.

- Challenges: When random seeds are set, the result will lose some randomness.

5. Store outputs when result is generated and connect textual statements to underlying results.

- Challenges: The storage may be used up if a bunch of files are stored during this process.

6. Make all the scripts, runs and results public and easily accessible.

- Challenges: Data privacy may become a big concern.

**Problem 4**

```r
if (!require("data.table")) {
  install.packages("data.table")
}
library(data.table)
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% 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 A1**

```r
# non-numeric variables
cat_sum<-summary(us_filtered[,c(1,7,8,9,11)])
cat_tab<-matrix(0, nrow = 3, ncol = 5)
for (i in 1:3) {
  for (j in 1:5) {
    cat_split<-unlist(strsplit(cat_sum[i,j], ":"))
    cat_tab[i,j]<-gsub(" ", "", cat_split[2])
  }
}
colnames(cat_tab)<-colnames(cat_sum)
rownames(cat_tab)<-c("Length", "Class", "Mode")
knitr::kable(cat_tab, align = "c")
```

|  | dateRep | countriesAndTerritories | geoId | countryterritoryCode | continentExp |
|---|---|---|---|---|---|
| Length | 61 | 61 | 61 | 61 | 61 |
| Class | character | character | character | character | character |
| Mode | character | character | character | character | character |

```r
# numeric variables
num_sum<-summary(us_filtered[,-c(1,7,8,9,11)])
num_tab<-matrix(0, nrow = 6, ncol = 8)
for (i in 1:6) {
  for (j in 1:8) {
    num_split<-unlist(strsplit(num_sum[i,j], ":"))
    num_tab[i,j]<-gsub(" ", "", num_split[2])
  }
}
colnames(num_tab)<-c("day", "month", "year", "cases", "deaths", "population",
                     "14-days cumlative covid-19 case", "index")
rownames(num_tab)<-c("Min.", "1st Qu.", "Median", "Mean", "3rd Qu.", "Max.")
knitr::kable(num_tab, align = "c")
```

|  | day | month | year | cases | deaths | population | 14-days cumlative covid-19 case | index |
|---|---|---|---|---|---|---|---|---|
| Min. | 1.00 | 6.000 | 2020 | 18665 | 242.0 | 329064917 | 89.76 | 1 |
| 1st Qu. | 8.00 | 6.000 | 2020 | 25540 | 500.0 | 329064917 | 92.43 | 16 |
| Median | 16.00 | 7.000 | 2020 | 45221 | 767.0 | 329064917 | 150.94 | 31 |
| Mean | 15.75 | 6.508 | 2020 | 44666 | 791.6 | 329064917 | 170.16 | 31 |
| 3rd Qu. | 23.00 | 7.000 | 2020 | 61796 | 982.0 | 329064917 | 247.01 | 46 |
| Max. | 31.00 | 7.000 | 2020 | 78427 | 2437.0 | 329064917 | 282.72 | 61 |

We have limited ourselves to 61 time points.

```r
sum(is.na(us_filtered))
```

```
## [1] 0
```

There are no missing values in the dataset.

### Part A2

```r
if (!require("stargazer")) {
  install.packages("stargazer")
}
library(stargazer)
stargazer(fit, type = "latex", align = TRUE, title = "Summary table of linear model")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Tue, Sep 14, 2021 - 7:05:53 PM % Requires LaTeX packages: dcolumn

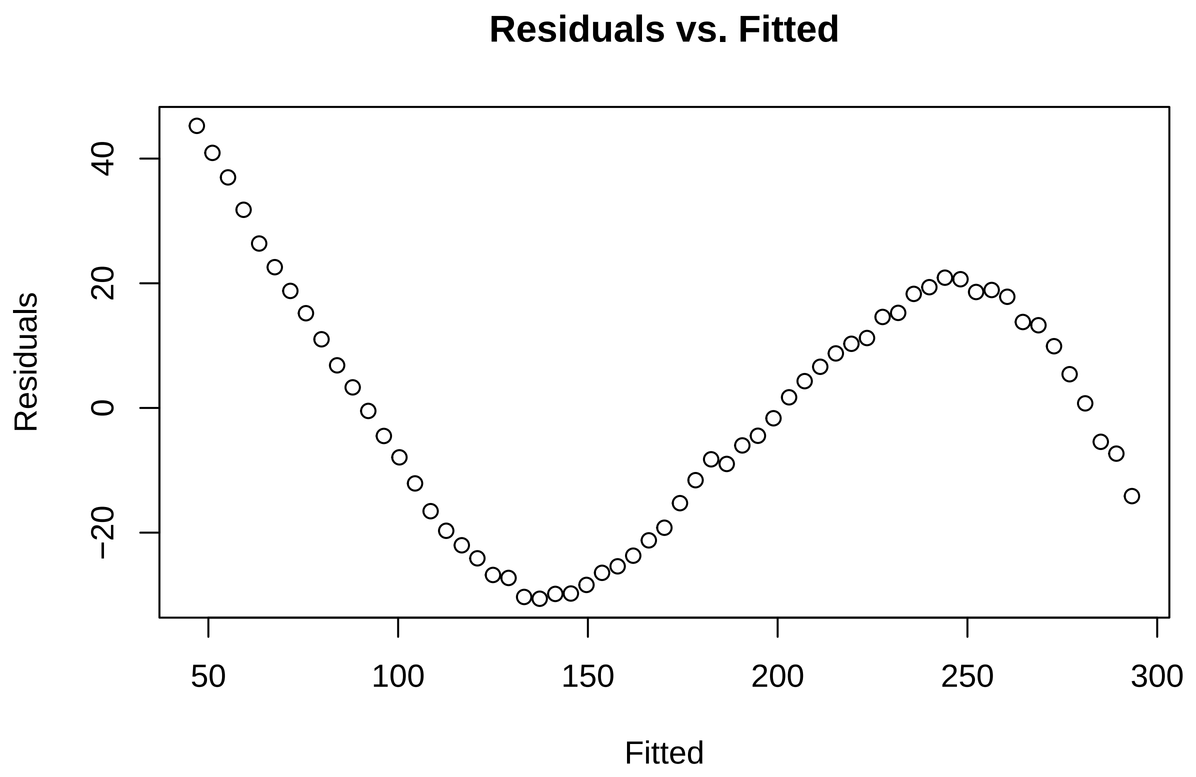
### Part B

Table 3: Summary table of linear model

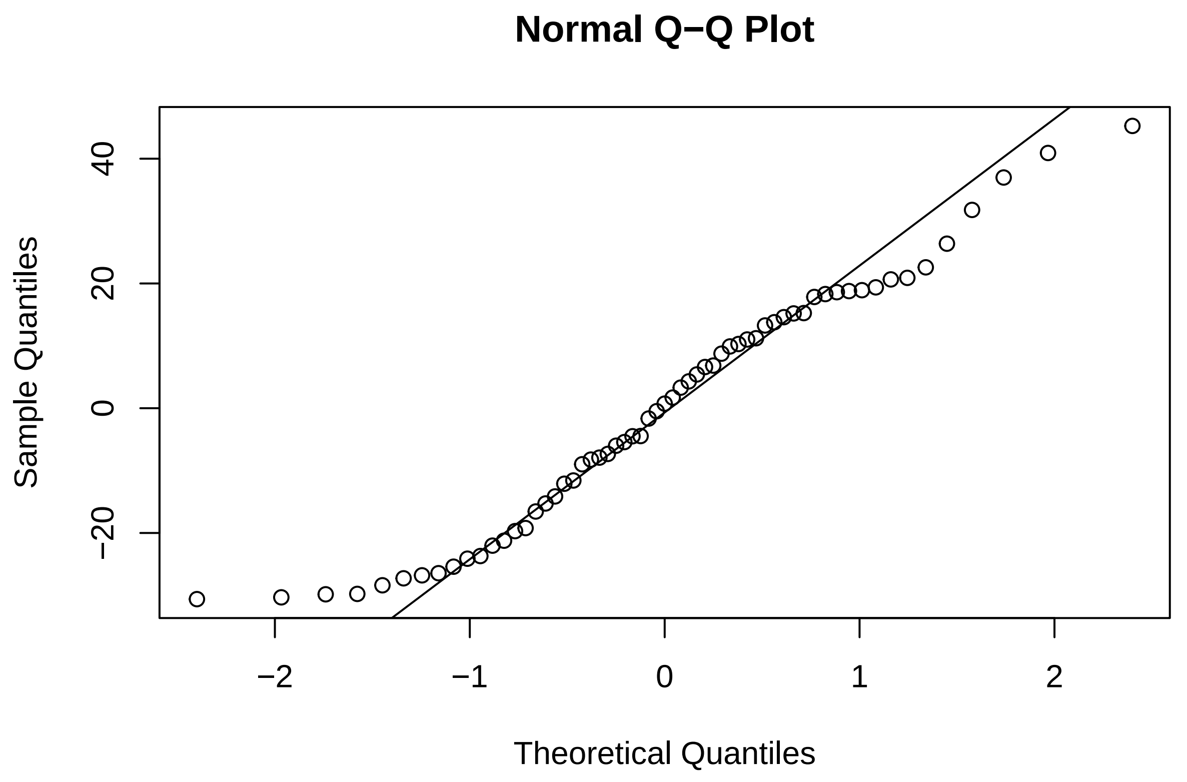
| | *Dependent variable:* |
|---|---|
| | ‘Cumulative_number_for_14_days_of_COVID-19_cases_per_100000‘ |
| index | 4.107$^{***}$ |
| | (0.145) |
| Constant | 42.853$^{***}$ |
| | (5.165) |
| Observations | 61 |
| $R^2$ | 0.932 |
| Adjusted $R^2$ | 0.930 |
| Residual Std. Error | 19.922 (df = 59) |
|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

```
if (!require("broom")) {
  install.packages("broom")
}
library(broom)
fit.diags<-augment(fit)
```

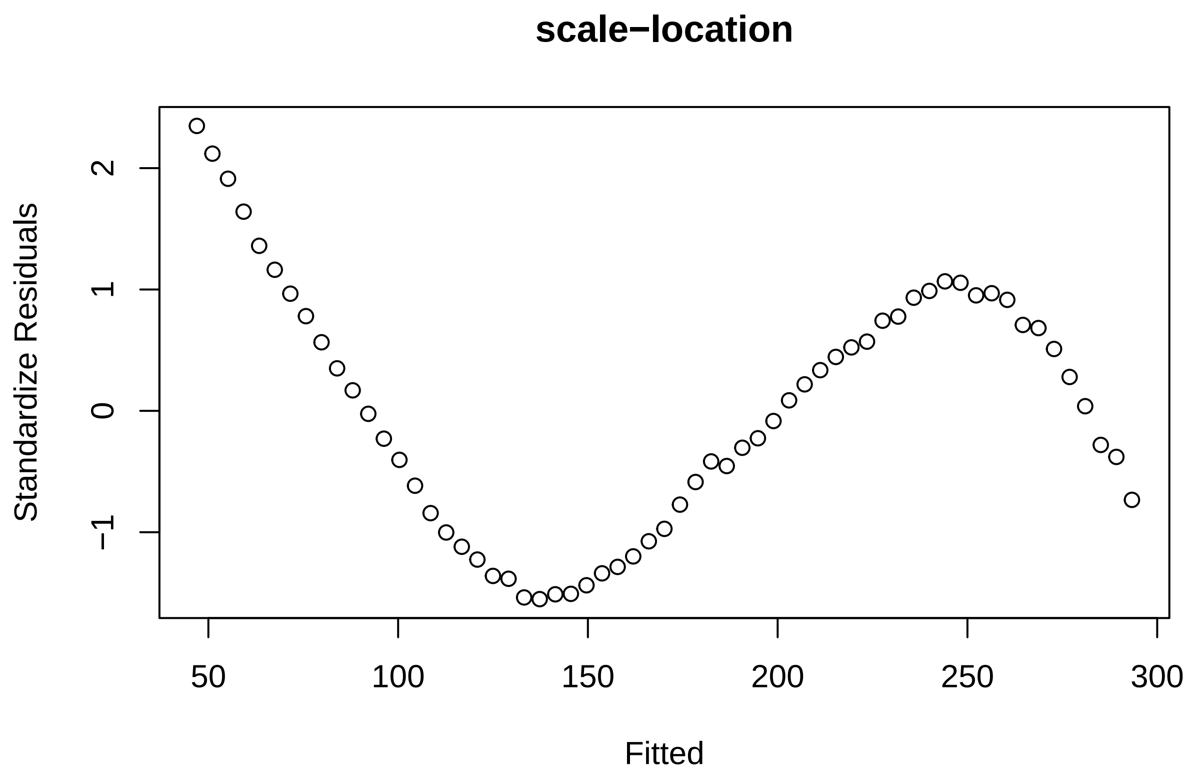
```
# residuals vs. fitted
plot(fit.diags$.fitted, fit.diags$.resid, xlab = "Fitted", ylab = "Residuals",
     main = "Residuals vs. Fitted")
```



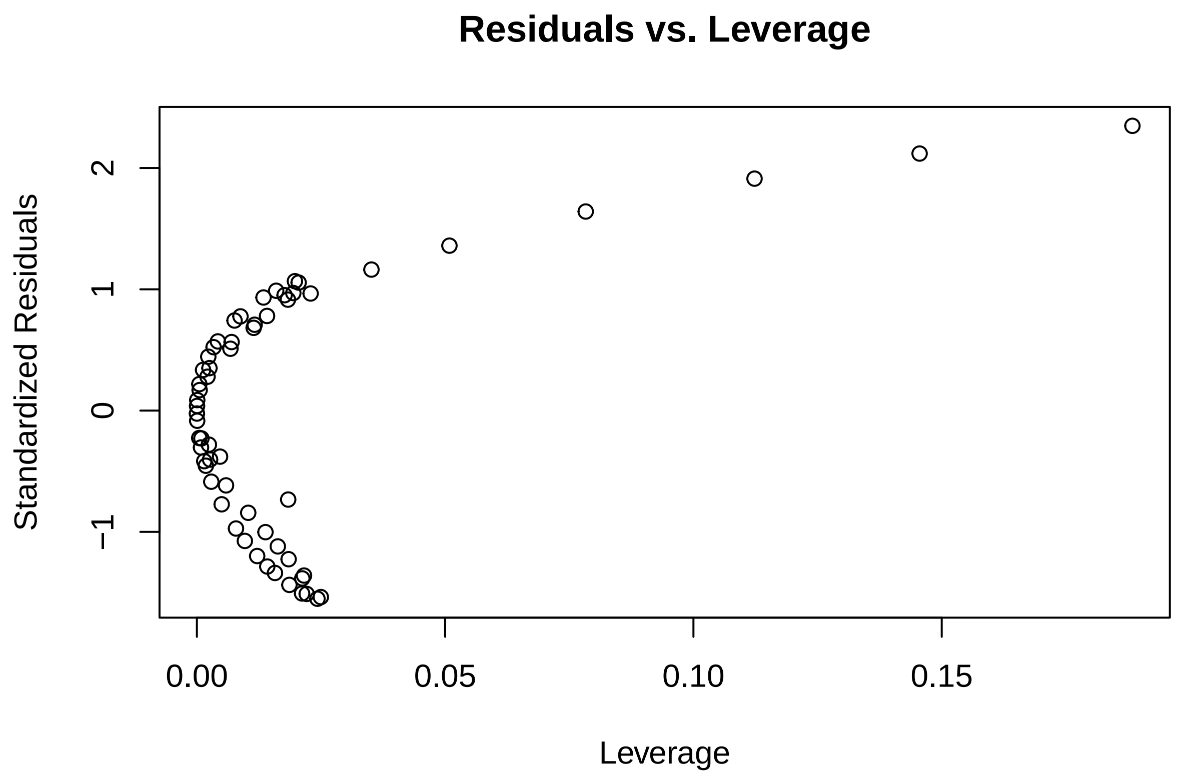
```
# normal Q-Q
qqnorm(fit.diags$.resid)
qqline(fit.diags$.resid)
```



```r
# scale-location
plot(fit.diags$.fitted, fit.diags$.std.resid, xlab = "Fitted",
     ylab = "Standardize Residuals", main = "scale-location")
```

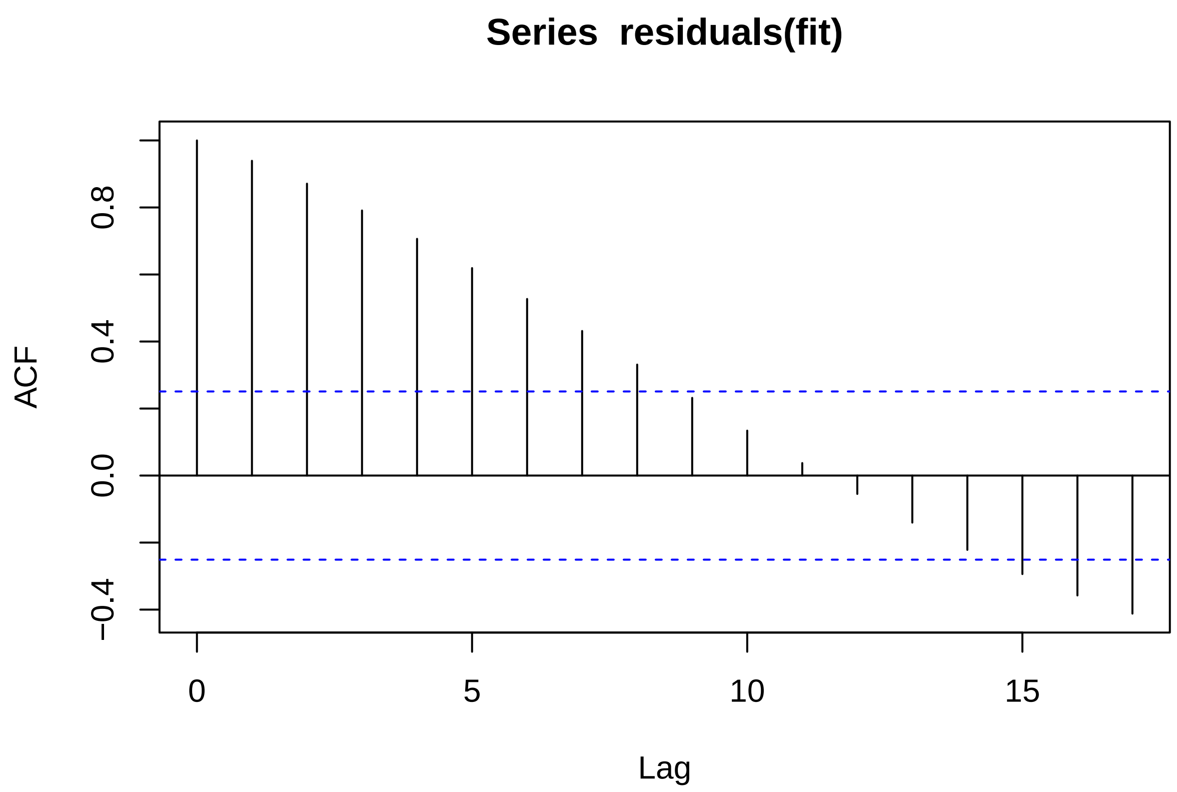


```r
# residual vs. leverage
plot(fit.diags$.cooksd, fit.diags$.std.resid, xlab = "Leverage",
     ylab = "Standardized Residuals", main = "Residuals vs. Leverage")
```



**Part C**

```r
# auto correlation plot for residuals
acf(residuals(fit))
```



**Problem 5**

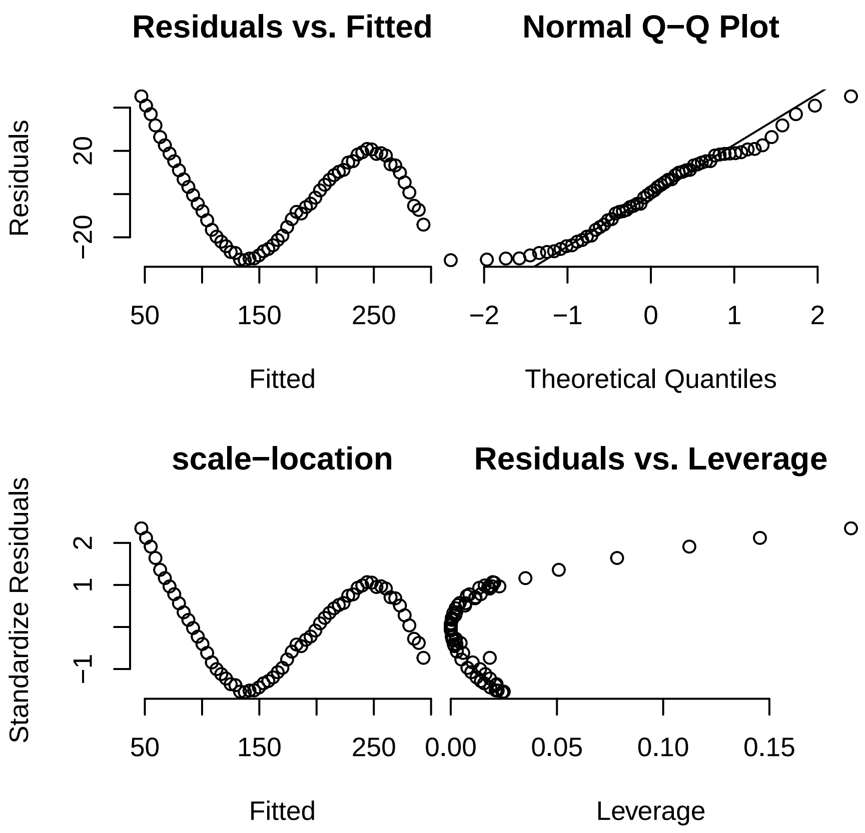
```r
layout(matrix(c(1,2,3,4), nrow = 2, ncol = 2, byrow = TRUE), respect = TRUE)

# residuals vs. fitted
par(mar = c(4, 4, 4, 0))
plot(fit.diags$.fitted, fit.diags$.resid, xlab = "Fitted", ylab = "Residuals",
     main = "Residuals vs. Fitted", axes = FALSE)
axis(side = 1, at = seq(50, 300, by = 50))
axis(side = 2, at = seq(-20, 40, by = 20))

# normal Q-Q
par(mar = c(4, 0, 4, 0))
qqnorm(fit.diags$.resid, axes = FALSE, ylab = "")
axis(side = 1, at = -2:2)
qqline(fit.diags$.resid)

# scale-location
par(mar = c(4, 4, 4, 0))
plot(fit.diags$.fitted, fit.diags$.std.resid, xlab = "Fitted",
     ylab = "Standardize Residuals", main = "scale-location", axes = FALSE)
axis(side = 2, at = -1:2)
axis(side = 1, at = seq(50, 300, by = 50))
```

```r
# residual vs. leverage
par(mar = c(4, 0, 4, 0))
plot(fit.diags$.cooksd, fit.diags$.std.resid, xlab = "Leverage",
     ylab = "", main = "Residuals vs. Leverage", axes = FALSE)
axis(side = 1, at = seq(0, 0.15, by = 0.05))
```


Residuals vs. Fitted
Normal Q−Q Plot
Residuals
Fitted
Theoretical Quantiles
scale−location
Residuals vs. Leverage
Standardize Residuals
Fitted
Leverage