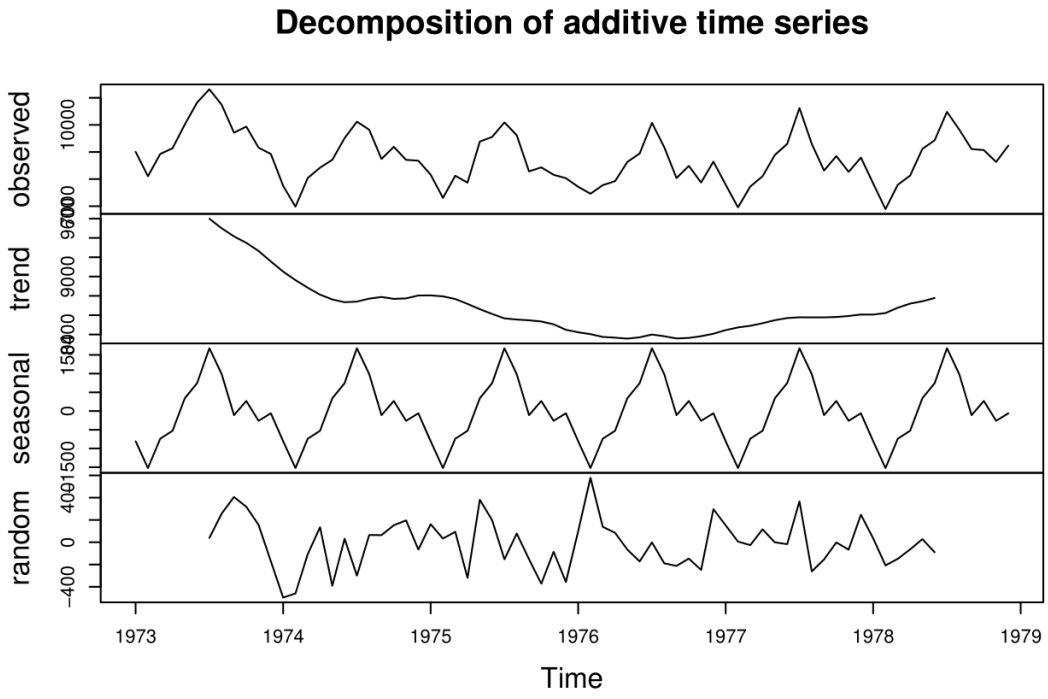
时间序列第一次作业

翁武泰　222018314210008

对 B6 的数据创建时间序列 death_ts，使用 decompose 函数对 death_ts 分解，得到如下图较为精准的结果，可与后面方法得到的结果做对比。

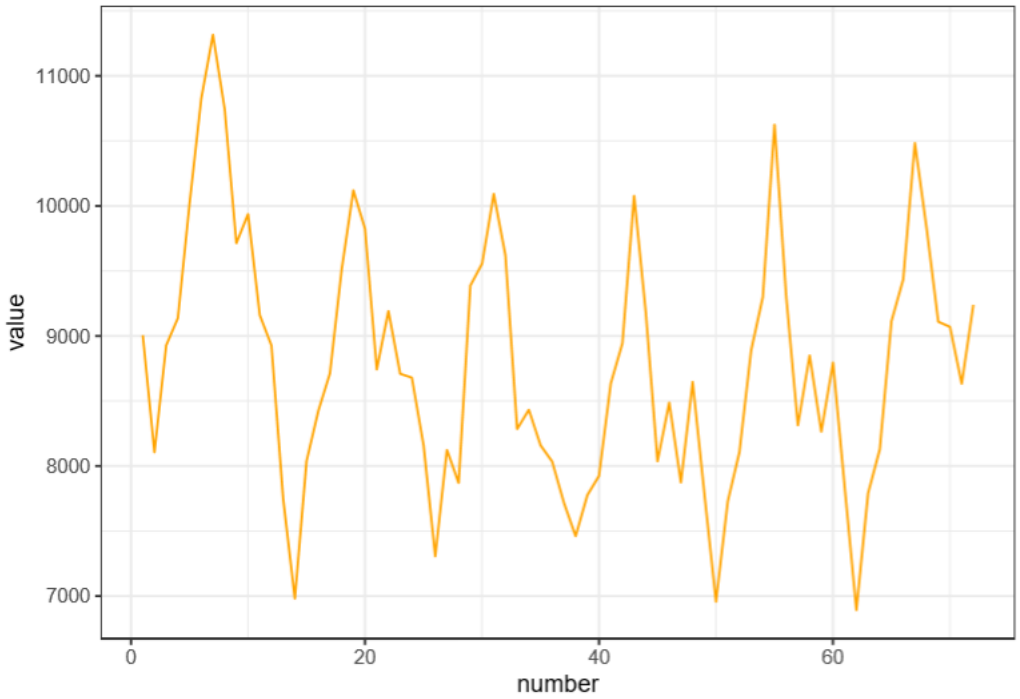
```
death_ts <- ts(death_data, frequency = 12, start = c(1973,1))
death_decomposition <- decompose(death_ts)
plot(death_decomposition)
```



转化数据格式，通过 ggplot 作图得到 1973 年至 1978 年美国在意外事故中的死亡人数数据图。通过观察发现周期 $T = 12$，即一年为一周期。

```
death_tibble <- as_tibble(death_data)
death_tibble <- death_tibble %>% mutate(number = c(1:72))
death_date_frame <- as.data.frame(death_tibble)
```

```
ggplot(death_tibble) +
  geom_line(aes(x = number, y = value), color = 'orange') +
  theme_bw()
```



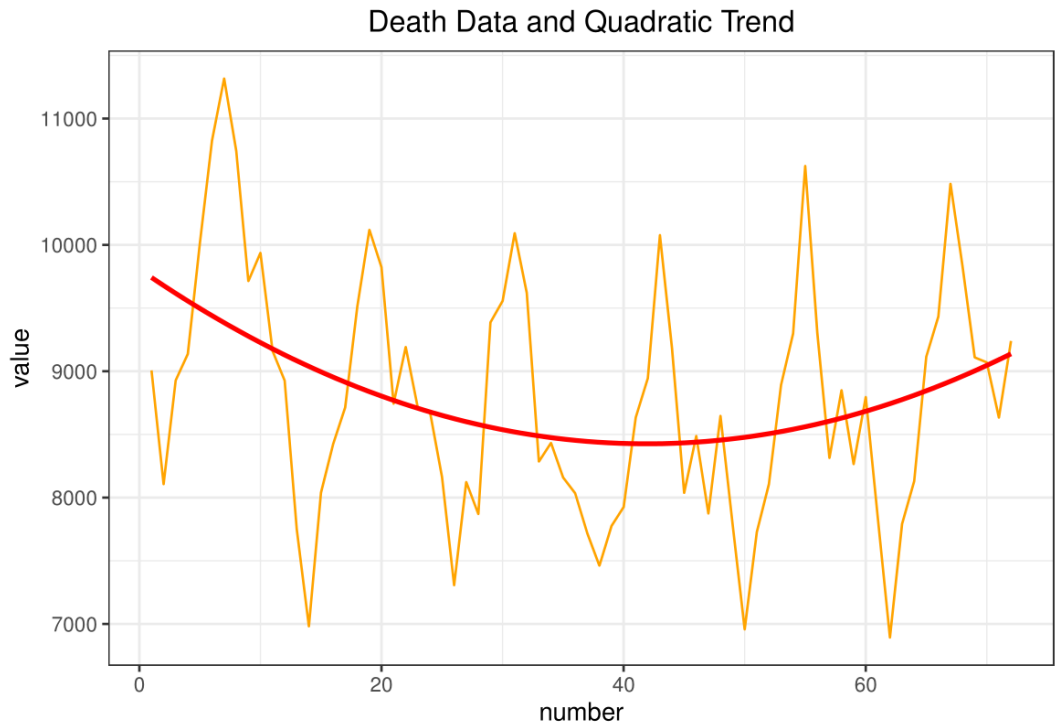
通过回归模型，对数据进行线性回归和二次回归，得到直线趋势项和二次趋势项如下图所示。其中橙线为数据，红线为趋势项。

```
# linear function regression
model_linear <- lm(formula = value ~ number, data = death_date_frame)
coefficient_linear <- coefficients(model_linear)
coefficient_linear
```

```
# quadratic function regression
model_quadratic <- lm(formula = value ~ poly(number,2), data = death_date_frame)
coefficient_quadratic <- coefficients(model_quadratic)
coefficient_quadratic
```

```
ggplot(death_tibble) +
  geom_line(aes(x = number, y = value), color = 'orange') +
  geom_abline(slope = coefficient_linear[2],
              intercept = coefficient_linear[1],
              color = 'red', size = 1) +
  labs(title = 'Death Data and Linear Trend') +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```


```r
# quadratic function plot
ggplot(death_tibble, aes(x = number, y = value)) +
  geom_line(color = 'orange') +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2),
              color='red', size = 1, se = FALSE) +
  labs(title = 'Death Data and Quadratic Trend') +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



用原始数据减去趋势项的估计，以第 k 月的平均值作为季节项 $S(k)$ 的估计，根据课本公式（1.7）计算出 $\hat{S}$，再用原始数据减去趋势项数据的估计和季节项的估计得随机项的估计。其中 S1，S2 表示线性回归和二次回归后得到的季节项，T1、T2 和 R1、R2 同理。

people	number	S1	T1	R1	S2	T2	R2
9007	1	-791.42567	9090.702	707.723616	-3513748	9865.552	3512889.107
8106	2	-1543.08312	9082.193	566.890282	-3675389	16097.470	3667397.975
8928	3	-754.40724	9073.684	608.723616	-3840646	27484.377	3822089.510
9137	4	-545.06470	9065.174	616.890282	-4011637	44026.274	3976747.379
10017	5	324.77785	9056.665	635.556949	-4187122	65723.162	4131415.747
10826	6	802.45373	9048.156	975.390282	-4368154	92575.039	4286405.282
11317	7	1668.46294	9039.647	608.890282	-4553953	124581.907	4440688.484
10744	8	973.30549	9031.138	739.556949	-4746469	161743.765	4595468.852
9713	9	-67.01863	9022.628	757.390282	-4944484	204060.612	4750136.387
9938	10	236.32391	9014.119	687.556949	-5146311	251532.450	4904716.256
9161	11	-283.16688	9005.610	438.556949	-5354115	304159.278	5059116.958
8927	12	-21.15766	8997.101	-48.943051	-5566293	361941.096	5213279.159

```
for (i in 1 : 72) {
  death_tibble[i,3] <- death_tibble[i,1] -
    (coefficient_linear[2] * death_tibble[i,2]
     + coefficient_linear[1])
}

death_tibble <- rename(death_tibble, 'S_and_T' = 'value')

for (i in 0 : 11) {
  sum <- 0
  for (j in 1 :72) {
    if (j %% 12 == i)
      sum <- sum + death_tibble[j,3]
  }
  sum <- sum / 6
  for (j in 1 :72) {
    if (j %% 12 == i)
      death_tibble[j,4] <- sum
  }
}
```

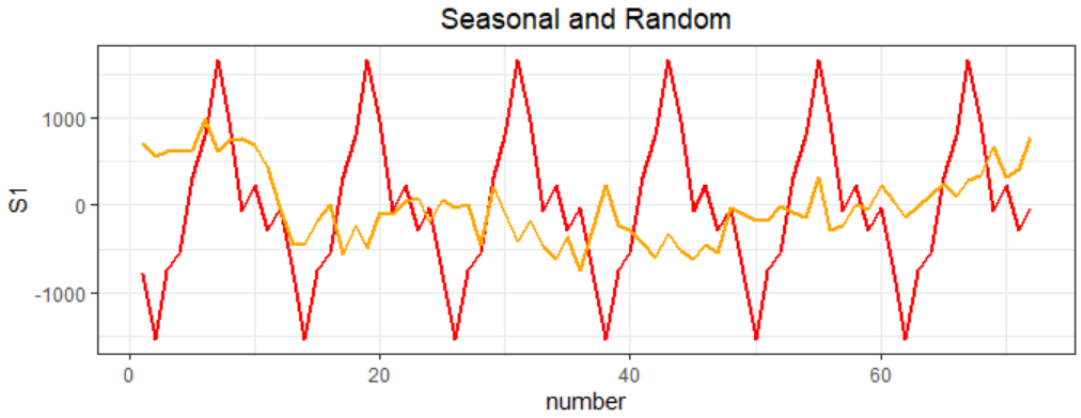
```
death_tibble <- rename(death_tibble, 'S' = 'value')
death_tibble <- rename(death_tibble, 'qwe' = 'S_and_T')
death_tibble <- rename(death_tibble, 'S_and_T1' = 'S')
death_tibble <- rename(death_tibble, 'S1' = 'value')
death_tibble <- rename(death_tibble, 'people' = 'qwe')

for (i in 1 : 72) {
  death_tibble[i,5] <- coefficient_linear[2] * death_tibble[i,2]
  + coefficient_linear[1]
}

death_tibble <- rename(death_tibble, 'qwe' = 'number')
death_tibble <- rename(death_tibble, 'T1' = 'number')
death_tibble <- rename(death_tibble, 'number' = 'qwe')

for (i in 1 : 72) {
  death_tibble[i,6] <- death_tibble[i,1] - death_tibble[i,4]
  - death_tibble[i,5]
}

death_tibble <- rename(death_tibble, 'qwe' = 'people')
death_tibble <- rename(death_tibble, 'R1' = 'people')
death_tibble <- rename(death_tibble, 'people' = 'qwe')

death_tibble <- death_tibble[-3]
```

```
for (i in 1 : 72) {
  death_tibble[i,6] <- death_tibble[i,1]
     - (coefficient_quadratic[3] * death_tibble[i,2]^2
     + coefficient_quadratic[2] * death_tibble[i,2]
     + coefficient_quadratic[1])
}

death_tibble <- rename(death_tibble, 'qwe' = 'people')
death_tibble <- rename(death_tibble, 'S_and_T2' = 'people')
death_tibble <- rename(death_tibble, 'people' = 'qwe')

for (i in 0 : 11) {
  sum <- 0
  for (j in 1 :72) {
    if (j %% 12 == i)
      sum <- sum + death_tibble[j,6]
  }
  sum <- sum / 6
  for (j in 1 :72) {
    if (j %% 12 == i)
      death_tibble[j,7] <- sum
  }
}
```

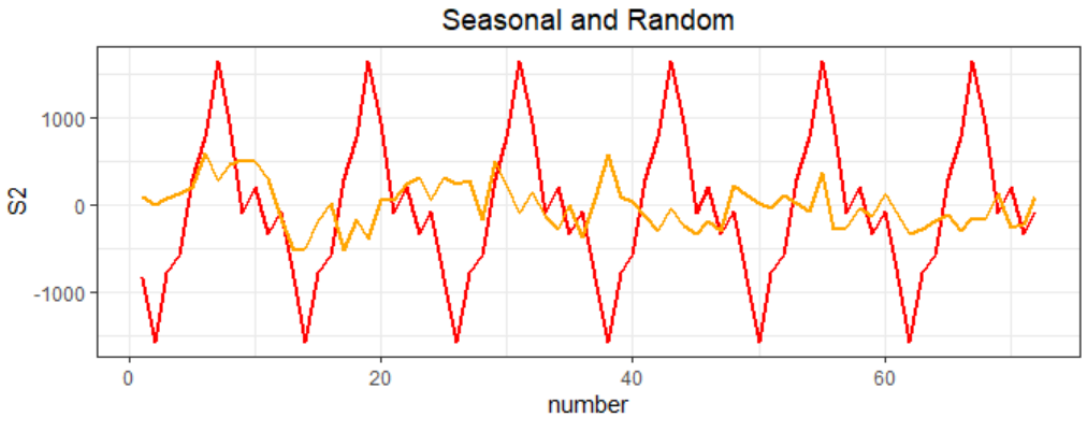
```
death_tibble <- rename(death_tibble, 'qwe' = 'S_and_T2')
death_tibble <- rename(death_tibble, 'S2' = 'S_and_T2')
death_tibble <- rename(death_tibble, 'S_and_T2' = 'qwe')

for (i in 1 : 72) {
  death_tibble[i,8] <- coefficient_quadratic[3] *
    death_tibble[i,2]^2
  + coefficient_quadratic[2] * death_tibble[i,2]
  + coefficient_quadratic[1]
}

death_tibble <- rename(death_tibble, 'qwe' = 'number')
death_tibble <- rename(death_tibble, 'T2' = 'number')
death_tibble <- rename(death_tibble, 'number' = 'qwe')

for (i in 1 : 72) {
  death_tibble[i,9] <- death_tibble[i,1] - death_tibble[i,7]
  - death_tibble[i,8]
}

death_tibble <- rename(death_tibble, 'qwe' = 'people')
death_tibble <- rename(death_tibble, 'R2' = 'people')
death_tibble <- rename(death_tibble, 'people' = 'qwe')

death_tibble <- death_tibble[-6]

death_tibble
```

绘制季节项和随机项如下图所示。其中红色为季节项，橙色为随机项。

```
ggplot(death_tibble) +
  geom_line(aes(x = number, y = S1), color = 'red', size = 1) +
  geom_line(aes(x = number, y = R1), color = 'orange', size = 1) +
  labs(title = 'Seasonal and Random') +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
ggplot(death_tibble) +
  geom_line(aes(x = number, y = S2), color = 'red', size = 1) +
  geom_line(aes(x = number, y = R2), color = 'orange', size = 1) +
  labs(title = 'Seasonal and Random') +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```

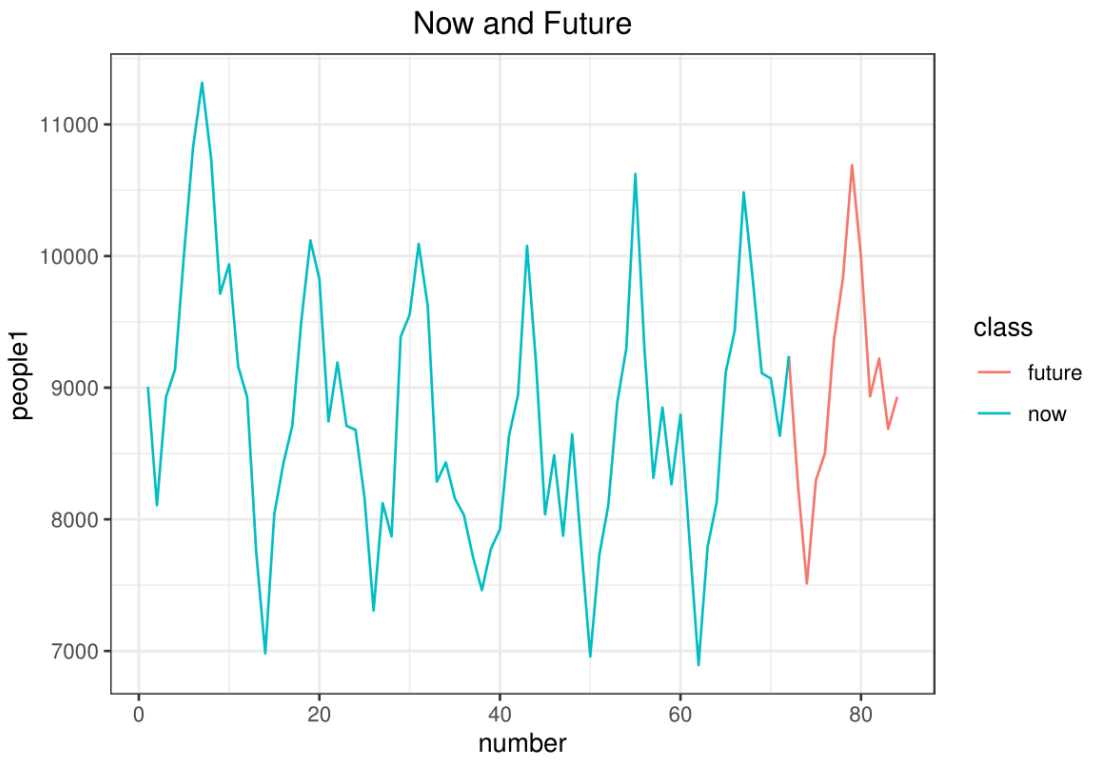


根据公式（1.9）预测，得到结果并绘制图片如下所示。

```
for (i in 1 : 12) {
  f_tibble[i,2] <- death_tibble[i,6]
  + coefficient_linear[2] * death_tibble[i,2]
  + coefficient_linear[1]
  f_tibble[i,3] <- death_tibble[i,3]
  + coefficient_quadratic[3] * f_tibble[i,1]^2
  + coefficient_quadratic[2] * f_tibble[i,1]
  + coefficient_quadratic[1]
}

f_tibble <- rename(f_tibble, 'F1' = 'S2')
f_tibble <- rename(f_tibble, 'F2' = 'S1')
```

month	1979	1979
1	8265.502	8468.374
2	7512.326	7768.417
3	8297.884	8610.393
4	8502.508	8874.635
5	9366.032	9800.978
6	9835.789	10336.75
7	10692.28	11262.46
8	9986.004	10628.61
9	8932.962	9651.181
10	9221.986	10019.02
11	8686.577	9565.633
12	8931.067	9895.342

```
ggplot(f) +
  geom_line(aes(x = number, y = people1, color = type1)) +
  labs(
    title = 'Now and Future',
    color = 'class'
  ) +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
ggplot(f) +
  geom_line(aes(x = number, y = people2, color = type2)) +
  labs(
    title = 'Now and Future',
    color = 'class'
  ) +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```