

《计算与编程》第二次作业报告

王超 12031012

第一题

思路 and 结果:

1.1

使用 `read.table` 函数读取 `signif.txt` 文件, 再用 `as_tibble` 函数将其转化为 `*tibble` 格式即可, 使用 `class ()` 查看结果:

```
> data1 <- read.table(file = 'signif.txt', sep = '\t', header = TRUE, quote = '')
> Sig_Eqs <- as_tibble(data1) #将其转化为表格
> class(Sig_Eqs)
[1] "tbl_df"      "tbl"        "data.frame"
> |
```

1.2

为了方便计数, 先将 `Sig_Eqs` 中的 `NA` 值赋值为 0, 再根据国家分组 (`group_by` 函数), 然后分别求和各个国家的死亡总数, 并将其降序排列 (`arrange (desc ())` 函数), 再输出死亡总数前十的国家及死亡总数:

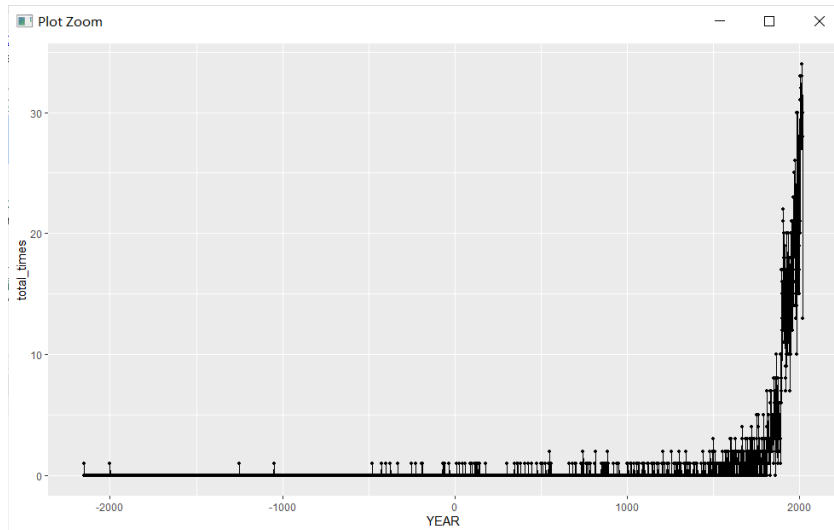
```
> every_country_totaldeaths
# A tibble: 155 x 2
  COUNTRY      DT
  <chr>      <int>
1 CHINA      2074881
2 TURKEY     1074651
3 IRAN       1036676
4 SYRIA      439224
5 ITALY      434865
6 HAITI       321224
7 AZERBAIJAN 317219
8 JAPAN       278137
9 ARMENIA     191890
10 PAKISTAN   148692
# ... with 145 more rows
> every_country_totaldeaths[1:10,]
# A tibble: 10 x 2
  COUNTRY      DT
  <chr>      <int>
1 CHINA      2074881
2 TURKEY     1074651
3 IRAN       1036676
4 SYRIA      439224
5 ITALY      434865
6 HAITI       321224
7 AZERBAIJAN 317219
8 JAPAN       278137
9 ARMENIA     191890
10 PAKISTAN   148692
> |
```

1.3

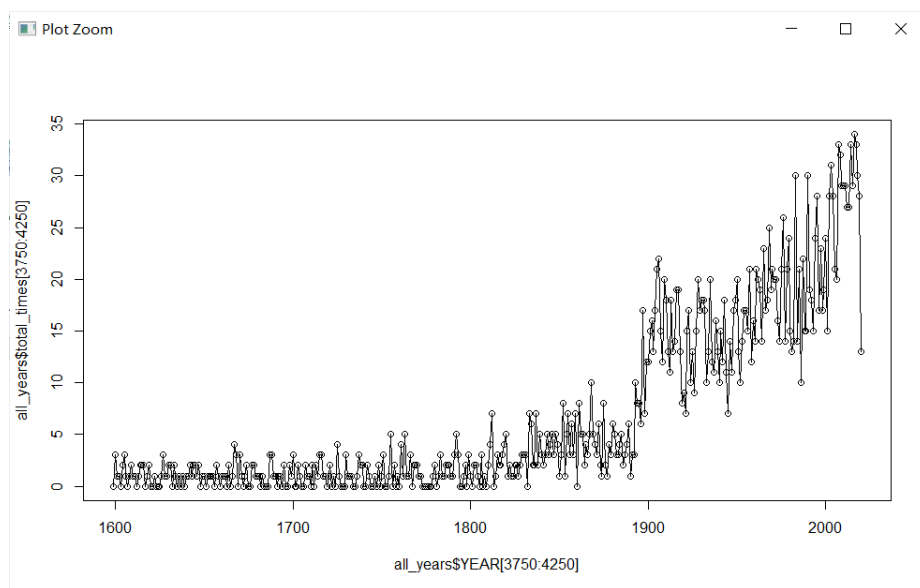
因为需要计算每一年全球范围内地震级数大于 6.0 的地震总数, 对于地震数据缺失或者没有大于 6.0 的地震的年份的 6.0 级以上地震总数赋值为 0, 这样每一年的地震级数大于 6 的年度地震总数数据就齐了。新建一列 `times` 数据用来计数, 当该行的地震等级大于 6 时, `times` 列值为 1, 否则为 0; 根据年份分组, 再求和 `times`, 得到所求数据, 再作图即可, 也可以进一步分析地震次数随时间的变化。作图后发现, 在最近两百年来, 全球地震等级大于 6 的每年地震总数开始暴涨, 甚至呈现一定的指数增长, 可能的原因有两个, 一个是随着社会的进步以及科技的发展, 人们对地震的发现与记录也更加精确, 相关的地震数据也更多, 从而使得数据中的大于 6 的地震总数也随之增加, 另一方面可能是由于人类活动对地球造成了一定的影响, 改变了地球的一部分环境, 从而导致地球运动更加剧烈, 地震次数与地震

强度都有了一定的增加。下图是部分相关代码和图表。

```
> all_years %>%  
+   ggplot(aes(x=YEAR, y=total_times)) +  
+   geom_line() +  
+   geom_point(size=1)  
>
```



```
> #绘制近500年每年地震总数，进一步观察年地震总数变化趋势  
> plot(all_years$YEAR[3750:4250], all_years$total_times[3750:4250], type = 'o')  
>
```



1.4

将 *Sig_Eqs* 中的 NA 值赋值为 0, 再分别获取每个国家地震总数, 并按照总数大小排序, 将数据存入 *total_times_country* 中, 类似的, 获取每个国家最大地震等级数据, 将其存入 *maxEq_country*, 最后获取所有会用到的数据 (国家, 年月日, 地震等级), 将其存入 *all_data_country* 中。再利用上述生成的数据表格, 自定义函数, 以实现题目要求。

关于自定义函数的说明: 使用了两个 for 循环

#第一个 for 循环, 遍历国家, 先在 *total_times_country* 中找到该国对应的地震总数

#再使用 if 在 *maxEq_country* 中找到该国家对应的最大地震级数

#第二个 for 循环, 遍历国家, 在 *all_data_country* 中找最大地震级数对应的日期

#国家和地震级数都相符时, 输出日期即可。

```
> #利用上述生成的数据表格, 自定义函数, 以实现题目要求
> CountEq_LargestEq <- function(country_name){
+   for(i in 1:nrow(total_times_country)){#第一个for循环, 遍历国家
+     if (total_times_country[i,1] == country_name ){#找到该国对应的地震总数
+       T_time <- total_times_country[i,2]
+       print(paste0('地震总数: ',T_time))
+     }
+     if(maxEq_country[i,1] == country_name ){#找到该国家对应的最大地震级数: maxEq_country[i,2]
+       for(j in 1:nrow(all_data_country)){#第二个for循环, 遍历国家, 找最大地震级数对应的日期
+         if(all_data_country[j,4] == country_name && #第四列分别是国家名字和地震级数
+           all_data_country[j,5] == maxEq_country[i,2]) #国家和地震级数相等时。输出日期即可
+         {
+           print(paste0('最大地震级数: ',maxEq_country[i,2],',', '地震日期: ',
+             all_data_country[j,1],',年',all_data_country[j,2],',月',
+             all_data_country[j,3],',日'))
+         }
+       }
+     }
+   }
+ }
>
> CountEq_LargestEq("CHINA")
[1] "地震总数: 606"
[1] "最大地震级数: 8.5,地震日期: 1668年7月25日"
```

#用 for(i in 1:10)遍历地震总数前 10 的国家, 输出各自的地震总数, 最大地震等级及其发生的日期

#若想遍历所有国家, 将 for(i in 1:10) 中的 10 改为国家总数即可。

```
> for(i in 1:10){ #遍历地震总数前10的国家, 输出各自的地震总数, 最大地震等级及其发生的日期
+   #若想遍历所有国家, 将 for(i in 1:10) 中的10改为国家总数即可
+   test_name <- total_times_country[i,1]
+   print(paste0(i,',',test_name,': '))
+   CountEq_LargestEq(test_name)
+ }
[1] "1.CHINA: "
[1] "地震总数: 606"
[1] "最大地震级数: 8.5,地震日期: 1668年7月25日"
[1] "2.JAPAN: "
[1] "地震总数: 406"
[1] "最大地震级数: 9.1,地震日期: 2011年3月11日"
[1] "3.INDONESIA: "
[1] "地震总数: 394"
[1] "最大地震级数: 9.1,地震日期: 2004年12月26日"
[1] "4.IRAN: "
[1] "地震总数: 380"
[1] "最大地震级数: 7.9,地震日期: 856年12月22日"
[1] "5.TURKEY: "
[1] "地震总数: 329"
[1] "最大地震级数: 7.8,地震日期: 1912年8月9日"
[1] "最大地震级数: 7.8,地震日期: 1916年1月24日"
[1] "6.ITALY: "
[1] "地震总数: 326"
[1] "最大地震级数: 7.5,地震日期: 1915年1月13日"
[1] "7.USA: "
[1] "地震总数: 268"
[1] "最大地震级数: 9.2,地震日期: 1964年3月28日"
[1] "8.GREECE: "
[1] "地震总数: 264"
[1] "最大地震级数: 8,地震日期: 365年7月21日"
[1] "最大地震级数: 8,地震日期: 1303年8月8日"
[1] "9.PHILIPPINES: "
[1] "地震总数: 220"
[1] "最大地震级数: 8.7,地震日期: 1897年9月21日"
[1] "10.MEXICO: "
[1] "地震总数: 203"
[1] "最大地震级数: 8.4,地震日期: 1899年1月24日"
> |
```

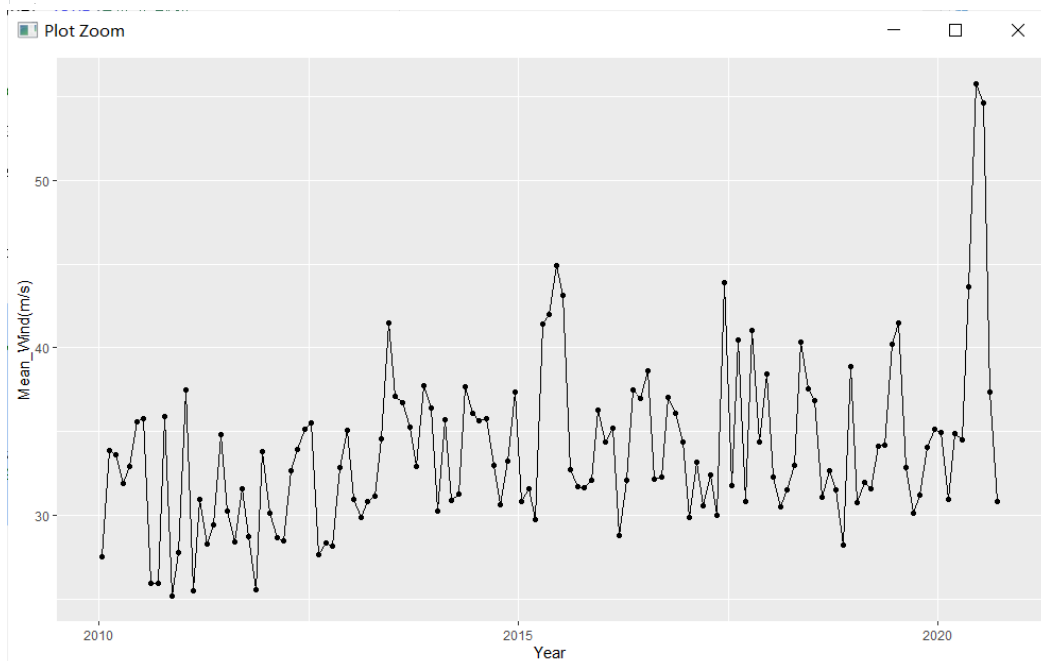
第二题

思路 and 结果：使用 `read.csv` 读入数据，由于表格中的 `WND` 列为表征风速相关的数据，使用 `separate` 函数来分割 `WND` 列，分别分成表征风向，风向质量，类型，风速，风速质量的五列数据，查表知我们需要用风速质量来筛选符合要求的数据。

```
data2 <- read.csv(file = '2281305.csv',header=TRUE)#读入数据
colnames(data2)
#将WND分裂开，得到不同的数据列
data2 <- separate(data = data2, col = WND, into = c("direction",
"direction_quality","type","speed","speed_quality"), sep = ",")
#日期时间列分割开
data2 <- separate(data = data2, col = DATE, into = c("date", "time"), sep = "T")
#分为年月日，便于求解月平均风速
data2 <- separate(data = data2, col = date, into = c("year", "month", "day"), sep = "-")
```

当风速质量为 0, 1, 4, 5 时为满足质量要求的数据，筛选后以月分组，再求平均值，并将月平均值的那天设为当月的 15 号。再做出月平均的时间序列图即可。

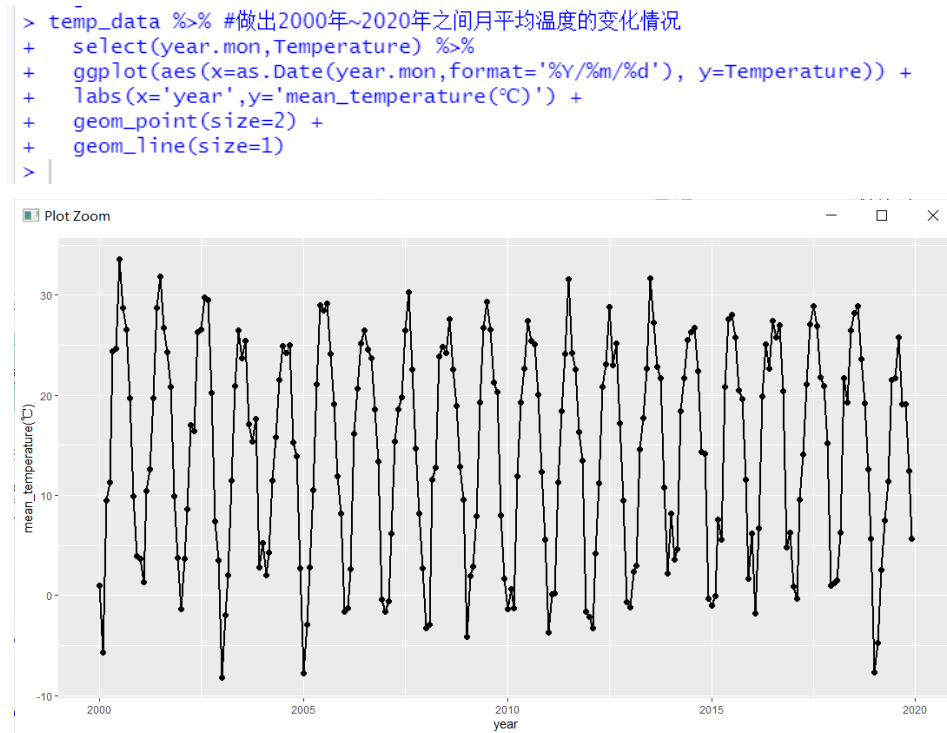
```
> wind_data_day <- wind_data %>%
+ mutate(day_mean = 15)
> wind_data <- unite(wind_data_day, "monthly", year, month, day_mean, sep = "-", remove = FALSE)
> wind_data %>%
+ select(year, monthly, speed, speed_quality) %>%
+ filter(speed_quality == '0' | speed_quality == '1' | speed_quality == '4' | speed_quality == '5') %>%
+ group_by(monthly) %>%
+ summarize(mean_speed = mean(as.numeric(speed)))
# summarise() ungrouping output (override with '.groups' argument)
# A tibble: 129 x 2
  monthly      mean_speed
  <chr>      <dbl>
1 2010-01-15      27.6
2 2010-02-15      33.9
3 2010-03-15      33.6
4 2010-04-15      31.9
5 2010-05-15      32.9
6 2010-06-15      35.6
7 2010-07-15      35.8
8 2010-08-15      26.0
9 2010-09-15      25.9
10 2010-10-15      35.9
# ... with 119 more rows
```



第三题

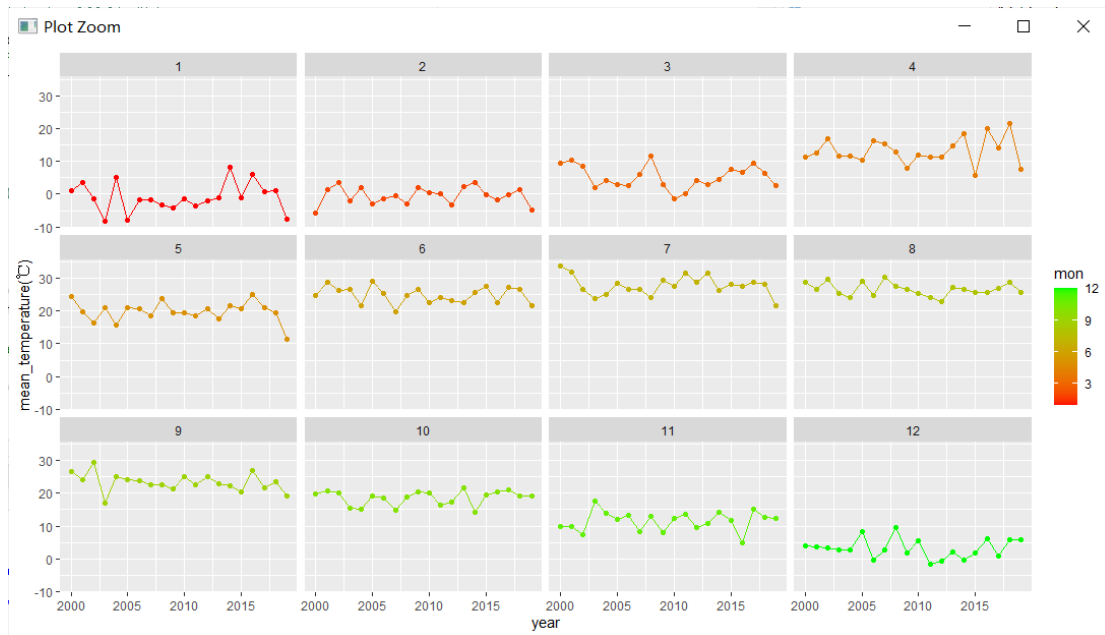
思路 and 结果：

读入数据后，先检查数据有无异常值，选取的数据为河北省某地的 2000~2019 年的月平均温度数据，检查后发现温度的范围在 $[-8.2^{\circ}\text{C}, 33.6^{\circ}\text{C}]$ 之间，没有明显的异常值。首先用 `ggplot` 函数做出 2000 年~2020 年之间月平均温度的变化图，复现作业 1-7 中的时间序列曲线图。



此外可以再作图观察 20 年每年相同月份的平均温度变化情况：

```
> data3 %>% #作图观察20年每年相同月份的平均温度变化情况
+   # Make the plot
+   ggplot(aes(x=year, y=p1 * 0.1, color=mon)) +
+   labs(x='year', y='mean_temperature(°C)') +
+   scale_color_gradient(low = "red", high = "green") +
+   geom_point() +
+   geom_line() +
+   facet_wrap(~ mon)
>
```



最后可以作图观察 2000, 2007, 2013, 2019 各年月平均温度的变化:

```
> data_years <- data3 %>% #作图观察2000, 2007, 2013, 2019各年月平均温度的变化
+   filter(year == 2000 | year == 2007 | year == 2013 | year == 2019) %>%
+   mutate(tem = p1 * 0.1) %>%
+   select(year, mon, tem)
>
> ggplot(data=data_years, aes(x=factor(mon), y=factor(tem), #factor 转化为离散型
+   group=factor(year), colour=factor(year),
+   shape = factor(year))) +
+   labs( x='Month', y='Mean Temperature(°C)') +
+   geom_line(size = 1) +
+   theme(legend.title = element_blank()) + #去除图例名字
+   geom_point(size = 4)
> |
```

