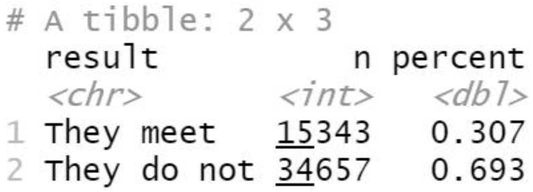
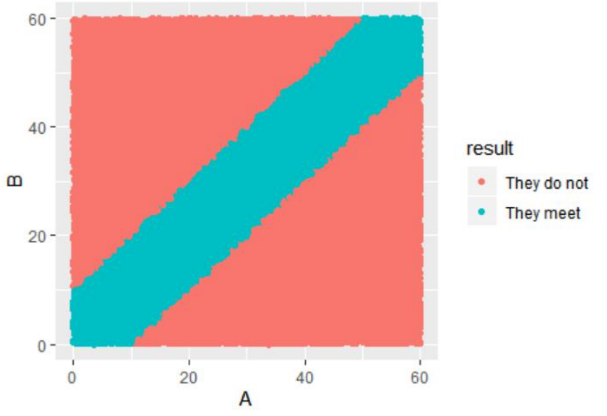
问题 1：

```
n_Sim <- 50000
sim_meet <- tibble(
   A = runif(n_Sim, min = 0, max = 60),
   B = runif(n_Sim, min = 0, max = 60)
) %>%
   mutate(result = ifelse(abs(A - B) <= 10,
                                                "They meet", "They do not"))
p_meet <- sim_meet %>% count(result) %>%
   arrange(n) %>%
   mutate(percent = n / n_Sim)
p_meet
```



```
# A tibble: 2 x 3
  result          n percent
  <chr>       <int>   <dbl>
1 They meet   15343   0.307
2 They do not 34657   0.693
```

```
ggplot(data = sim_meet, aes(x = A, y = B, color = result)) +
   geom_point()
```



```
## 最后一问就是学生不断修改 min = 10, max = 50
```

问题 2：

(1) 从 flights 数据表中挑选出以下变量：(year, month, day, hour, origin, dep_delay, distance, carrier)，将生产的新表保存为 flight1。

```
library(tidyverse)
library(nycflights13)
flight1<-select(flights, year, month, day, hour, origin, dep_delay, distance, carrier)
```

(2) 从 weather 数据表中挑选出以下变量：(year, month, day, hour, origin, humid, wind_speed)，将生产的新表保存为 weather1。

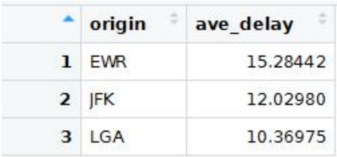
```
weather1<-select(weather, year, month, day, hour, origin, humid, wind_speed)
```

(3) 将 flight1 表和 weather1 表根据共同变量进行内连接，随机抽取 100000 行数据，将生产的结果保存为 flight_weather。(提示：sample_n()函数，不用重复抽取)

```
flight_weather <- inner_join(flight1, weather1) %>% sample_n(100000)
```

(4) 从 flight_weather 表中对三个出发机场按照平均出发延误时间排降序，并将结果保留在 longest_delay 表中。把结果展示出来。

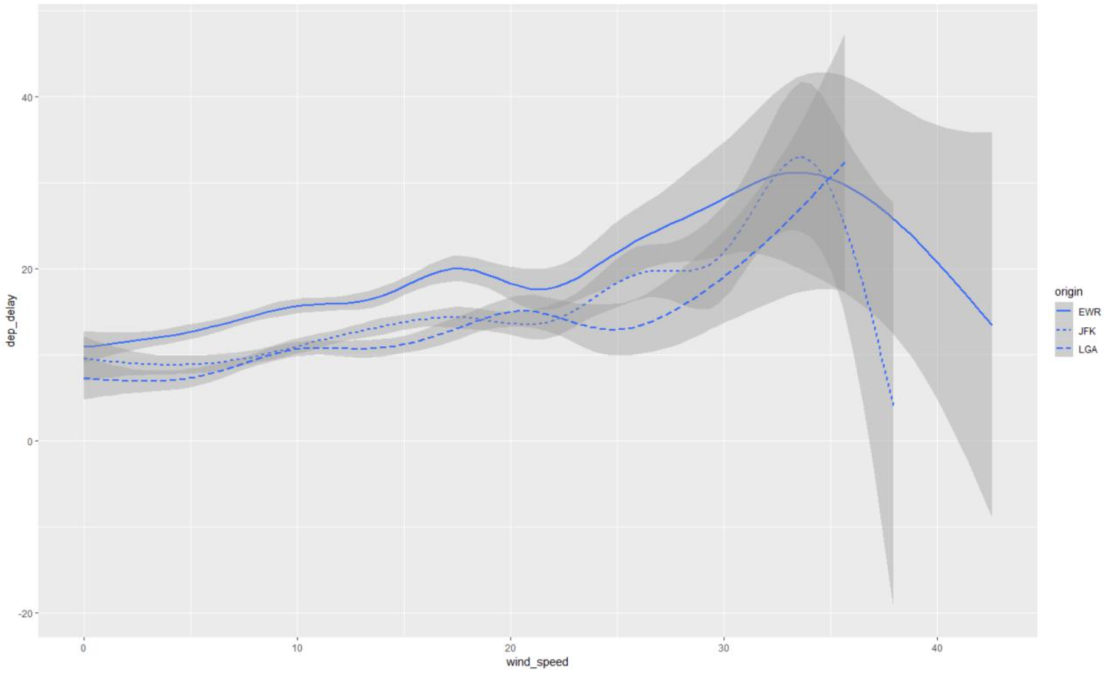
```
longest_delay<- flight_weather %>% group_by(origin) %>% summarise(ave_delay = mean(dep_delay, na.rm = TRUE)) %>% arrange(desc(ave_delay))
```



	origin	ave_delay
1	EWR	15.28442
2	JFK	12.02980
3	LGA	10.36975

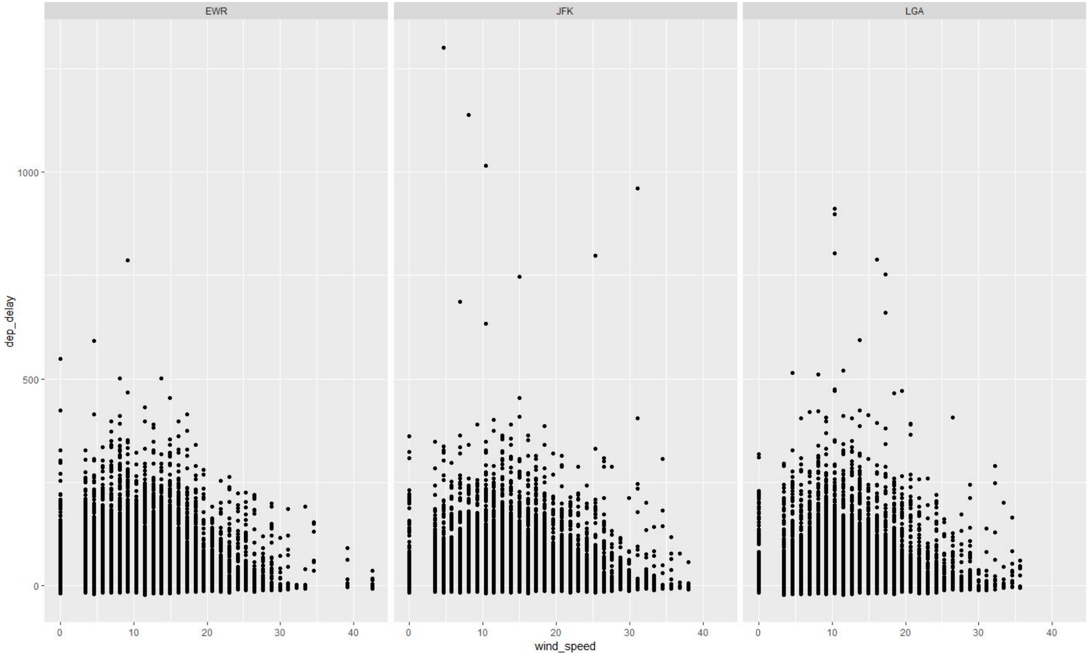
(5) 根据出发地（origin） 在同一个图中画出风速 wind_speed（x 轴）和出发延误时间 dep_delay（y 轴）的平滑曲线图

```
ggplot(data = flight_weather)+geom_smooth(mapping = aes(x = wind_speed, y = dep_delay, linetype = origin))
```



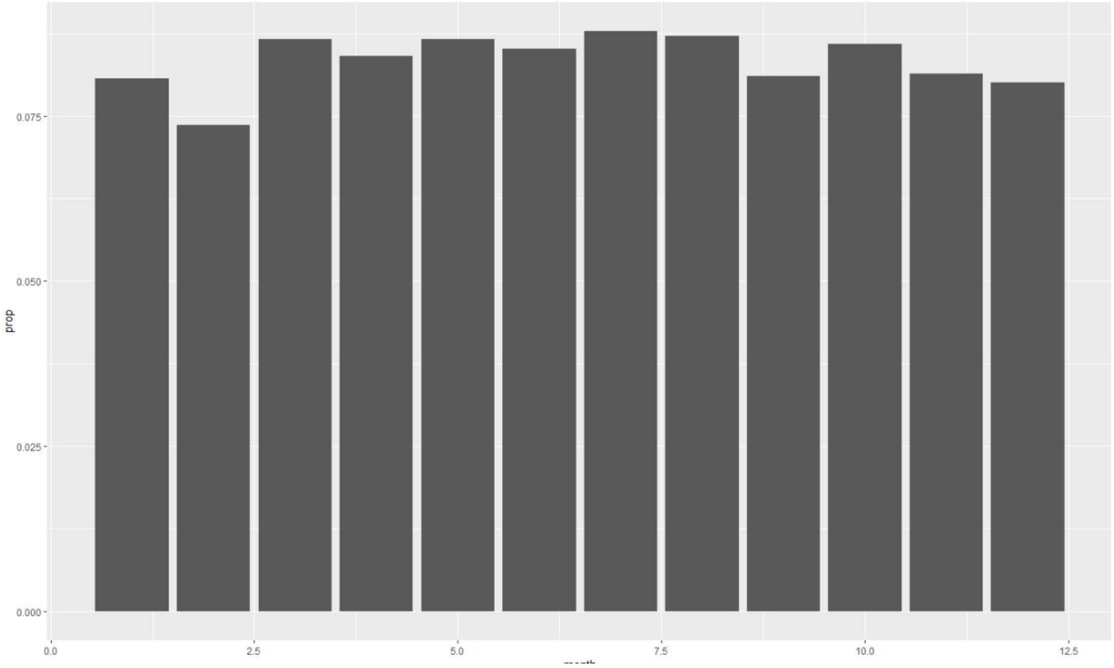
(6) 根据不同出发地（origin） 在平行的 3 个图中画出风速 wind_speed（x 轴）和出发延误时间 dep_delay（y 轴）的散点图。

```
ggplot(data = flight_weather) + geom_point(mapping = aes(x = wind_speed, y = dep_delay)) +
facet_wrap(~origin,nrow = 1)
```



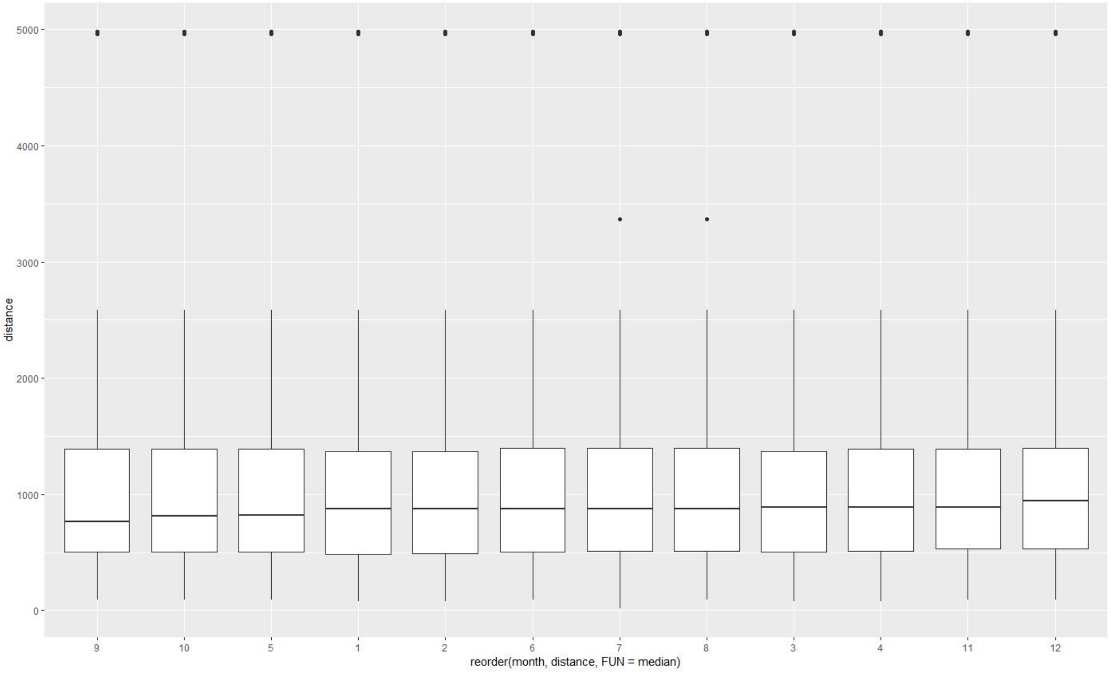
（7） 根据 flight_weather 表，画出每个月航班数的直方分布图，x 轴为月份，y 轴是每个月份航班数所占的比例。

```
ggplot(data=flight_weather)+geom_bar(mapping = aes(month, y=..prop.., group = 1) )
```



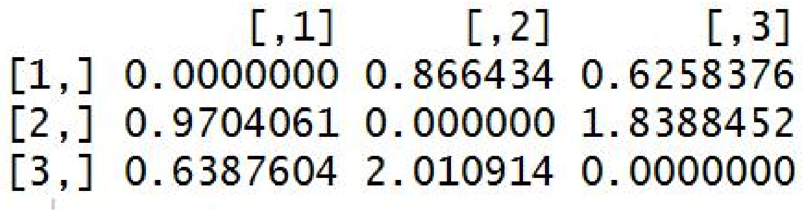
（8） 根据 flight_weather 表，画出每个月航班距离的 boxplot 图，x 轴为月份，y 轴为航行距离，根据的航行距离的中位数从低到高对 x 轴的月份进行重新排序。

```
ggplot(data=flight_weather)+geom_boxplot(mapping = aes(x=reorder(month, distance, FUN=median),
y=distance))
```



问题 3:

```
###### (1) #######
(H <- function(p) -sum(p*log(p)))
###### (2) #######
(DKL <- function(p,q) sum( p*(log(p)-log(q)) ))
###### (3) #######
IB <- list()
IB[[1]] <- c( 0.2 , 0.2 , 0.2 , 0.2 , 0.2 )
IB[[2]] <- c( 0.8 , 0.1 , 0.05 , 0.025 , 0.025 )
IB[[3]] <- c( 0.05 , 0.15 , 0.7 , 0.05 , 0.05 )
purrr::map_dbl( IB , H )
[1] 1.6094379 0.7430039 0.9836003
###### (4) #######
Dm <- matrix( NA , nrow=3 , ncol=3 )
for ( i in 1:3 ) {
    for ( j in 1:3 ) {
        Dm[i,j] <- DKL( IB[[j]] , IB[[i]] )
    }
}
Dm
```



```
             [,1]     [,2]      [,3]
[1,] 0.0000000 0.866434 0.6258376
[2,] 0.9704061 0.000000 1.8388452
[3,] 0.6387604 2.010914 0.0000000
```