**以下是Rstudio 的代码，对2019年和2020年从3月23日到4,5 月份每天的污染数据和温度，风速风向进行了对比区别的t test 和wilcoxon test**

setwd ("C:/Users/20736/Desktop/")

m\_d <- read.csv("Manchester\_daily.csv")

m\_d\_TTN\_mar\_may <- subset(m\_d, year == 2019 & ((month == 3 & day >= 23) | month == 4 | month == 5))

TTN\_m\_m\_so2 <- m\_d\_TTN\_mar\_may$so2

m\_d\_TTT\_mar\_may <- subset(m\_d, year == 2020 & ((month == 3 & day >= 23) | month == 4 | month == 5))

TTT\_m\_m\_so2 <- m\_d\_TTT\_mar\_may$so2

summary(TTN\_m\_m\_so2)

summary(TTT\_m\_m\_so2)

t.test(TTN\_m\_m\_so2, TTT\_m\_m\_so2)

wilcox.test(TTN\_m\_m\_so2, TTT\_m\_m\_so2)

TTN\_m\_m\_no <- m\_d\_TTN\_mar\_may$no

TTT\_m\_m\_no <- m\_d\_TTT\_mar\_may$no

summary(TTN\_m\_m\_no)

summary(TTT\_m\_m\_no)

t.test(TTN\_m\_m\_no, TTT\_m\_m\_no)

wilcox.test(TTN\_m\_m\_no, TTT\_m\_m\_no)

TTN\_m\_m\_no2 <- m\_d\_TTN\_mar\_may$no2

TTT\_m\_m\_no2 <- m\_d\_TTT\_mar\_may$no2

summary(TTN\_m\_m\_no2)

summary(TTT\_m\_m\_no2)

t.test(TTN\_m\_m\_no2, TTT\_m\_m\_no2)

wilcox.test(TTN\_m\_m\_no2, TTT\_m\_m\_no2)

TTN\_m\_m\_pm2.5 <- m\_d\_TTN\_mar\_may$pm2.5

TTT\_m\_m\_pm2.5 <- m\_d\_TTT\_mar\_may$pm2.5

summary(TTN\_m\_m\_pm2.5)

summary(TTT\_m\_m\_pm2.5)

t.test(TTN\_m\_m\_pm2.5, TTT\_m\_m\_pm2.5)

wilcox.test(TTN\_m\_m\_pm2.5, TTT\_m\_m\_pm2.5)

TTN\_m\_m\_o3 <- m\_d\_TTN\_mar\_may$o3

TTT\_m\_m\_o3 <- m\_d\_TTT\_mar\_may$o3

summary(TTN\_m\_m\_o3)

summary(TTT\_m\_m\_o3)

t.test(TTN\_m\_m\_o3, TTT\_m\_m\_o3)

wilcox.test(TTN\_m\_m\_o3, TTT\_m\_m\_o3)

TTN\_m\_m\_ws <- m\_d\_TTN\_mar\_may$ws

TTT\_m\_m\_ws <- m\_d\_TTT\_mar\_may$ws

summary(TTN\_m\_m\_ws)

summary(TTT\_m\_m\_ws)

t.test(TTN\_m\_m\_ws, TTT\_m\_m\_ws)

wilcox.test(TTN\_m\_m\_ws, TTT\_m\_m\_ws)

TTN\_m\_m\_wd <- m\_d\_TTN\_mar\_may$wd

TTT\_m\_m\_wd <- m\_d\_TTT\_mar\_may$wd

summary(TTN\_m\_m\_wd)

summary(TTT\_m\_m\_wd)

t.test(TTN\_m\_m\_wd, TTT\_m\_m\_wd)

wilcox.test(TTN\_m\_m\_wd, TTT\_m\_m\_wd)

TTN\_m\_m\_temp <- m\_d\_TTN\_mar\_may$temp

TTT\_m\_m\_temp <- m\_d\_TTT\_mar\_may$temp

summary(TTN\_m\_m\_temp)

summary(TTT\_m\_m\_temp)

t.test(TTN\_m\_m\_temp, TTT\_m\_m\_temp)

wilcox.test(TTN\_m\_m\_temp, TTT\_m\_m\_temp)

**以下代码对于2019年和2020年从3月23日到5月底的每天的数据做时间序列图(time serious plot)的代码，用于直观可视的对比两年中数据的变化**

# 安装并加载必要的包

install.packages("ggplot2")

# 设置语言环境为英文

Sys.setlocale("LC\_TIME", "English")

library(ggplot2)

# 创建日期列

m\_d\_TTN\_mar\_may$date <- as.Date(with(m\_d\_TTN\_mar\_may, paste(year, month, day, sep="-")), "%Y-%m-%d")

m\_d\_TTT\_mar\_may$date <- as.Date(with(m\_d\_TTT\_mar\_may, paste(year, month, day, sep="-")), "%Y-%m-%d")

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN\_mar\_may$date, so2 = m\_d\_TTN\_mar\_may$so2, year = 2019),

data.frame(date = m\_d\_TTT\_mar\_may$date, so2 = m\_d\_TTT\_mar\_may$so2, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = so2, color = factor(year))) +

geom\_line() +

labs(title = "SO2 Levels: 2019 vs 2020", x = "Date", y = "SO2(μg/m^3)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%b %Y")

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN\_mar\_may$date, no2 = m\_d\_TTN\_mar\_may$no2, year = 2019),

data.frame(date = m\_d\_TTT\_mar\_may$date, no2 = m\_d\_TTT\_mar\_may$no2, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = no2, color = factor(year))) +

geom\_line() +

labs(title = "no2 Levels: 2019 vs 2020", x = "Date", y = "no2(μg/m^3)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%b %Y")

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN\_mar\_may$date, no = m\_d\_TTN\_mar\_may$no, year = 2019),

data.frame(date = m\_d\_TTT\_mar\_may$date, no = m\_d\_TTT\_mar\_may$no, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = no, color = factor(year))) +

geom\_line() +

labs(title = "no Levels: 2019 vs 2020", x = "Date", y = "no(μg/m^3)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%b %Y")

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN\_mar\_may$date, pm2.5 = m\_d\_TTN\_mar\_may$pm2.5, year = 2019),

data.frame(date = m\_d\_TTT\_mar\_may$date, pm2.5 = m\_d\_TTT\_mar\_may$pm2.5, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = pm2.5, color = factor(year))) +

geom\_line() +

labs(title = "pm2.5 Levels: 2019 vs 2020", x = "Date", y = "pm2.5(μg/m^3)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%b %Y")

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN\_mar\_may$date, o3 = m\_d\_TTN\_mar\_may$o3, year = 2019),

data.frame(date = m\_d\_TTT\_mar\_may$date, o3 = m\_d\_TTT\_mar\_may$o3, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = o3, color = factor(year))) +

geom\_line() +

labs(title = "o3 Levels: 2019 vs 2020", x = "Date", y = "o3(μg/m^3)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%b %Y")

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN\_mar\_may$date, ws = m\_d\_TTN\_mar\_may$ws, year = 2019),

data.frame(date = m\_d\_TTT\_mar\_may$date, ws = m\_d\_TTT\_mar\_may$ws, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = ws, color = factor(year))) +

geom\_line() +

labs(title = "ws Levels: 2019 vs 2020", x = "Date", y = "ws(m/s)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%b %Y")

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN\_mar\_may$date, wd = m\_d\_TTN\_mar\_may$wd, year = 2019),

data.frame(date = m\_d\_TTT\_mar\_may$date, wd = m\_d\_TTT\_mar\_may$wd, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = wd, color = factor(year))) +

geom\_line() +

labs(title = "wd Levels: 2019 vs 2020", x = "Date", y = "wd(degrees)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%b %Y")

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN\_mar\_may$date, temp = m\_d\_TTN\_mar\_may$temp, year = 2019),

data.frame(date = m\_d\_TTT\_mar\_may$date, temp = m\_d\_TTT\_mar\_may$temp, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = temp, color = factor(year))) +

geom\_line() +

labs(title = "temp Levels: 2019 vs 2020", x = "Date", y = "temp(degree celcius)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%b %Y")

**以下是Rstudio 的代码，对2019年和2020年全年每天的污染数据和温度，风速风向进行了对比区别的t test 和wilcoxon test**

m\_d\_TTN <- subset(m\_d, year == 2019)

TTN\_so2 <- m\_d\_TTN$so2

m\_d\_TTT <- subset(m\_d, year == 2020)

# 找到2020年2月29日的位置

leap\_day\_index <- which(m\_d\_TTT$date == as.Date("2020-02-29"))

# 移除2020年2月29日的数据

m\_d\_TTT <- m\_d\_TTT[-leap\_day\_index, ]

TTT\_so2 <- m\_d\_TTT$so2

summary(TTN\_so2)

summary(TTT\_so2)

t.test(TTN\_so2, TTT\_so2)

wilcox.test(TTN\_so2, TTT\_so2)

TTN\_no <- m\_d\_TTN$no

TTT\_no <- m\_d\_TTT$no

summary(TTN\_no)

summary(TTT\_no)

t.test(TTN\_no, TTT\_no)

wilcox.test(TTN\_no, TTT\_no)

TTN\_no2 <- m\_d\_TTN$no2

TTT\_no2 <- m\_d\_TTT$no2

summary(TTN\_no2)

summary(TTT\_no2)

t.test(TTN\_no2, TTT\_no2)

wilcox.test(TTN\_no2, TTT\_no2)

TTN\_pm2.5 <- m\_d\_TTN$pm2.5

TTT\_pm2.5 <- m\_d\_TTT$pm2.5

summary(TTN\_pm2.5)

summary(TTT\_pm2.5)

t.test(TTN\_pm2.5, TTT\_pm2.5)

wilcox.test(TTN\_pm2.5, TTT\_pm2.5)

TTN\_o3 <- m\_d\_TTN$o3

TTT\_o3 <- m\_d\_TTT$o3

summary(TTN\_o3)

summary(TTT\_o3)

t.test(TTN\_o3, TTT\_o3)

wilcox.test(TTN\_o3, TTT\_o3)

TTN\_ws <- m\_d\_TTN$ws

TTT\_ws <- m\_d\_TTT$ws

summary(TTN\_ws)

summary(TTT\_ws)

t.test(TTN\_ws, TTT\_ws)

wilcox.test(TTN\_ws, TTT\_ws)

TTN\_wd <- m\_d\_TTN$wd

TTT\_wd <- m\_d\_TTT$wd

summary(TTN\_wd)

summary(TTT\_wd)

t.test(TTN\_wd, TTT\_wd)

wilcox.test(TTN\_wd, TTT\_wd)

TTN\_temp <- m\_d\_TTN$temp

TTT\_temp <- m\_d\_TTT$temp

summary(TTN\_temp)

summary(TTT\_temp)

t.test(TTN\_temp, TTT\_temp)

wilcox.test(TTN\_temp, TTT\_temp)

**以下是Rstudio 的代码，对2019年和2020年全年每天的污染数据和温度，风速风向绘制了箱线图进行对比**

# 找到2020年2月29日的位置

leap\_day\_index <- which(m\_d\_TTT$date == as.Date("2020-02-29"))

# 移除2020年2月29日的数据

m\_d\_TTT <- m\_d\_TTT[-leap\_day\_index, ]

# 创建数据框

data <- data.frame(

Group = rep(c("2019", "2020"), each = 365),

Value = c(m\_d\_TTN$so2, m\_d\_TTT$so2)

)

# 绘制箱线图

ggplot(data, aes(x = Group, y = Value, fill = Group)) +

geom\_boxplot() +

labs(title = "daliy data of SO2 from whole year of 2019 and 2020", x = "Years", y = "SO2(μg/m^3)") +

theme\_minimal() +

scale\_fill\_manual(values = c("2019" = "blue", "2020" = "red"))

# 创建数据框

data <- data.frame(

Group = rep(c("2019", "2020"), each = 365),

Value = c(m\_d\_TTN$no, m\_d\_TTT$no)

)

# 绘制箱线图

ggplot(data, aes(x = Group, y = Value, fill = Group)) +

geom\_boxplot() +

labs(title = "daliy data of no from whole year of 2019 and 2020", x = "Years", y = "no(μg/m^3)") +

theme\_minimal() +

scale\_fill\_manual(values = c("2019" = "blue", "2020" = "red"))

# 创建数据框

data <- data.frame(

Group = rep(c("2019", "2020"), each = 365),

Value = c(m\_d\_TTN$no2, m\_d\_TTT$no2)

)

# 绘制箱线图

ggplot(data, aes(x = Group, y = Value, fill = Group)) +

geom\_boxplot() +

labs(title = "daliy data of no2 from whole year of 2019 and 2020", x = "Years", y = "no2(μg/m^3)") +

theme\_minimal() +

scale\_fill\_manual(values = c("2019" = "blue", "2020" = "red"))

# 创建数据框

data <- data.frame(

Group = rep(c("2019", "2020"), each = 365),

Value = c(m\_d\_TTN$pm2.5, m\_d\_TTT$pm2.5)

)

# 绘制箱线图

ggplot(data, aes(x = Group, y = Value, fill = Group)) +

geom\_boxplot() +

labs(title = "daliy data of pm2.5 from whole year of 2019 and 2020", x = "Years", y = "pm2.5(μg/m^3)") +

theme\_minimal() +

scale\_fill\_manual(values = c("2019" = "blue", "2020" = "red"))

# 创建数据框

data <- data.frame(

Group = rep(c("2019", "2020"), each = 365),

Value = c(m\_d\_TTN$o3, m\_d\_TTT$o3)

)

# 绘制箱线图

ggplot(data, aes(x = Group, y = Value, fill = Group)) +

geom\_boxplot() +

labs(title = "daliy data of o3 from whole year of 2019 and 2020", x = "Years", y = "o3(μg/m^3)") +

theme\_minimal() +

scale\_fill\_manual(values = c("2019" = "blue", "2020" = "red"))

# 创建数据框

data <- data.frame(

Group = rep(c("2019", "2020"), each = 365),

Value = c(m\_d\_TTN$ws, m\_d\_TTT$ws)

)

# 绘制箱线图

ggplot(data, aes(x = Group, y = Value, fill = Group)) +

geom\_boxplot() +

labs(title = "daliy data of windspeed from whole year of 2019 and 2020", x = "Years", y = "windspeed(m/s)") +

theme\_minimal() +

scale\_fill\_manual(values = c("2019" = "blue", "2020" = "red"))

# 创建数据框

data <- data.frame(

Group = rep(c("2019", "2020"), each = 365),

Value = c(m\_d\_TTN$wd, m\_d\_TTT$wd)

)

# 绘制箱线图

ggplot(data, aes(x = Group, y = Value, fill = Group)) +

geom\_boxplot() +

labs(title = "daliy data of wind direction from whole year of 2019 and 2020", x = "Years", y = "wind direction(degree)") +

theme\_minimal() +

scale\_fill\_manual(values = c("2019" = "blue", "2020" = "red"))

# 创建数据框

data <- data.frame(

Group = rep(c("2019", "2020"), each = 365),

Value = c(m\_d\_TTN$temp, m\_d\_TTT$temp)

)

# 绘制箱线图

ggplot(data, aes(x = Group, y = Value, fill = Group)) +

geom\_boxplot() +

labs(title = "daliy data of temperature from whole year of 2019 and 2020", x = "Years", y = "temperature(degree celcius)") +

theme\_minimal() +

scale\_fill\_manual(values = c("2019" = "blue", "2020" = "red"))

**以下代码对于2019年和2020年(不包括2020年2月29号)全年的每天的数据做时间序列图(time serious plot)的代码，用于直观可视的对比两年中数据的变化**

# 设置语言环境为英文

Sys.setlocale("LC\_TIME", "English")

library(ggplot2)

# 创建日期列

m\_d\_TTN$date <- as.Date(with(m\_d\_TTN, paste(year, month, day, sep="-")), "%Y-%m-%d")

m\_d\_TTT$date <- as.Date(with(m\_d\_TTT, paste(year, month, day, sep="-")), "%Y-%m-%d")

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN$date, so2 = m\_d\_TTN$so2, year = 2019),

data.frame(date = m\_d\_TTT$date, so2 = m\_d\_TTT$so2, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = so2, color = factor(year))) +

geom\_line() +

labs(title = "SO2 Levels: 2019 vs 2020", x = "Date", y = "SO2(μg/m^3)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%y %b",

limits = as.Date(c("2019-01-01", "2020-12-31")))

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN$date, no = m\_d\_TTN$no, year = 2019),

data.frame(date = m\_d\_TTT$date, no = m\_d\_TTT$no, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = no, color = factor(year))) +

geom\_line() +

labs(title = "no Levels: 2019 vs 2020", x = "Date", y = "no(μg/m^3)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%y %b",

limits = as.Date(c("2019-01-01", "2020-12-31")))

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN$date, no2 = m\_d\_TTN$no2, year = 2019),

data.frame(date = m\_d\_TTT$date, no2 = m\_d\_TTT$no2, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = no2, color = factor(year))) +

geom\_line() +

labs(title = "no2 Levels: 2019 vs 2020", x = "Date", y = "no2(μg/m^3)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%y %b",

limits = as.Date(c("2019-01-01", "2020-12-31")))

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN$date, pm2.5 = m\_d\_TTN$pm2.5, year = 2019),

data.frame(date = m\_d\_TTT$date, pm2.5 = m\_d\_TTT$pm2.5, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = pm2.5, color = factor(year))) +

geom\_line() +

labs(title = "pm2.5 Levels: 2019 vs 2020", x = "Date", y = "pm2.5(μg/m^3)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%y %b",

limits = as.Date(c("2019-01-01", "2020-12-31")))

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN$date, o3 = m\_d\_TTN$o3, year = 2019),

data.frame(date = m\_d\_TTT$date, o3 = m\_d\_TTT$o3, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = o3, color = factor(year))) +

geom\_line() +

labs(title = "o3 Levels: 2019 vs 2020", x = "Date", y = "o3(μg/m^3)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%y %b",

limits = as.Date(c("2019-01-01", "2020-12-31")))

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN$date, ws = m\_d\_TTN$ws, year = 2019),

data.frame(date = m\_d\_TTT$date, ws = m\_d\_TTT$ws, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = ws, color = factor(year))) +

geom\_line() +

labs(title = "wind speed Levels: 2019 vs 2020", x = "Date", y = "wind speed(m/s)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%y %b",

limits = as.Date(c("2019-01-01", "2020-12-31")))

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN$date, wd = m\_d\_TTN$wd, year = 2019),

data.frame(date = m\_d\_TTT$date, wd = m\_d\_TTT$wd, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = wd, color = factor(year))) +

geom\_line() +

labs(title = "wind direction Levels: 2019 vs 2020", x = "Date", y = "wind direction(degree)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%y %b",

limits = as.Date(c("2019-01-01", "2020-12-31")))

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN$date, temp = m\_d\_TTN$temp, year = 2019),

data.frame(date = m\_d\_TTT$date, temp = m\_d\_TTT$temp, year = 2020)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = temp, color = factor(year))) +

geom\_line() +

labs(title = "temperature Levels: 2019 vs 2020", x = "Date", y = "temperature(degree celsius)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2020" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%y %b",

limits = as.Date(c("2019-01-01", "2020-12-31")))

**以下是Rstudio 的代码，对2018年和2020年全年每天的污染数据和温度，风速风向进行了对比区别的t test 和wilcoxon test**

m\_d\_TTE <- subset(m\_d, year == 2018)

TTE\_so2 <- m\_d\_TTE$so2

m\_d\_TTT <- subset(m\_d, year == 2020)

# 找到2020年2月29日的位置

leap\_day\_index <- which(m\_d\_TTT$date == as.Date("2020-02-29"))

# 移除2020年2月29日的数据

m\_d\_TTT <- m\_d\_TTT[-leap\_day\_index, ]

TTT\_so2 <- m\_d\_TTT$so2

summary(TTE\_so2)

summary(TTT\_so2)

t.test(TTE\_so2, TTT\_so2)

wilcox.test(TTE\_so2, TTT\_so2)

TTE\_no <- m\_d\_TTE$no

TTT\_no <- m\_d\_TTT$no

summary(TTE\_no)

summary(TTT\_no)

t.test(TTE\_no, TTT\_no)

wilcox.test(TTE\_no, TTT\_no)

TTE\_no2 <- m\_d\_TTE$no2

TTT\_no2 <- m\_d\_TTT$no2

summary(TTE\_no2)

summary(TTT\_no2)

t.test(TTE\_no2, TTT\_no2)

wilcox.test(TTE\_no2, TTT\_no2)

TTE\_pm2.5 <- m\_d\_TTE$pm2.5

TTT\_pm2.5 <- m\_d\_TTT$pm2.5

summary(TTE\_pm2.5)

summary(TTT\_pm2.5)

t.test(TTE\_pm2.5, TTT\_pm2.5)

wilcox.test(TTE\_pm2.5, TTT\_pm2.5)

TTE\_o3 <- m\_d\_TTE$o3

TTT\_o3 <- m\_d\_TTT$o3

summary(TTE\_o3)

summary(TTT\_o3)

t.test(TTE\_o3, TTT\_o3)

wilcox.test(TTE\_o3, TTT\_o3)

TTE\_ws <- m\_d\_TTE$ws

TTT\_ws <- m\_d\_TTT$ws

summary(TTE\_ws)

summary(TTT\_ws)

t.test(TTE\_ws, TTT\_ws)

wilcox.test(TTE\_ws, TTT\_ws)

TTE\_wd <- m\_d\_TTE$wd

TTT\_wd <- m\_d\_TTT$wd

summary(TTE\_wd)

summary(TTT\_wd)

t.test(TTE\_wd, TTT\_wd)

wilcox.test(TTE\_wd, TTT\_wd)

TTE\_temp <- m\_d\_TTE$temp

TTT\_temp <- m\_d\_TTT$temp

summary(TTE\_temp)

summary(TTT\_temp)

t.test(TTE\_temp, TTT\_temp)

wilcox.test(TTE\_temp, TTT\_temp)

**以下是Rstudio 的代码，对2018年和2020年从3月23日到4,5 月份每天的污染数据和温度，风速风向进行了对比区别的t test 和wilcoxon test**

m\_d\_TTE\_mar\_may <- subset(m\_d, year == 2018 & ((month == 3 & day >= 23) | month == 4 | month == 5))

TTE\_m\_m\_so2 <- m\_d\_TTE\_mar\_may$so2

m\_d\_TTT\_mar\_may <- subset(m\_d, year == 2020 & ((month == 3 & day >= 23) | month == 4 | month == 5))

TTT\_m\_m\_so2 <- m\_d\_TTT\_mar\_may$so2

summary(TTE\_m\_m\_so2)

summary(TTT\_m\_m\_so2)

t.test(TTE\_m\_m\_so2, TTT\_m\_m\_so2)

wilcox.test(TTE\_m\_m\_so2, TTT\_m\_m\_so2)

TTE\_m\_m\_no <- m\_d\_TTE\_mar\_may$no

TTT\_m\_m\_no <- m\_d\_TTT\_mar\_may$no

summary(TTE\_m\_m\_no)

summary(TTT\_m\_m\_no)

t.test(TTE\_m\_m\_no, TTT\_m\_m\_no)

wilcox.test(TTE\_m\_m\_no, TTT\_m\_m\_no)

TTE\_m\_m\_no2 <- m\_d\_TTE\_mar\_may$no2

TTT\_m\_m\_no2 <- m\_d\_TTT\_mar\_may$no2

summary(TTE\_m\_m\_no2)

summary(TTT\_m\_m\_no2)

t.test(TTE\_m\_m\_no2, TTT\_m\_m\_no2)

wilcox.test(TTE\_m\_m\_no2, TTT\_m\_m\_no2)

TTE\_m\_m\_pm2.5 <- m\_d\_TTE\_mar\_may$pm2.5

TTT\_m\_m\_pm2.5 <- m\_d\_TTT\_mar\_may$pm2.5

summary(TTE\_m\_m\_pm2.5)

summary(TTT\_m\_m\_pm2.5)

t.test(TTE\_m\_m\_pm2.5, TTT\_m\_m\_pm2.5)

wilcox.test(TTE\_m\_m\_pm2.5, TTT\_m\_m\_pm2.5)

TTE\_m\_m\_o3 <- m\_d\_TTE\_mar\_may$o3

TTT\_m\_m\_o3 <- m\_d\_TTT\_mar\_may$o3

summary(TTE\_m\_m\_o3)

summary(TTT\_m\_m\_o3)

t.test(TTE\_m\_m\_o3, TTT\_m\_m\_o3)

wilcox.test(TTE\_m\_m\_o3, TTT\_m\_m\_o3)

TTE\_m\_m\_ws <- m\_d\_TTE\_mar\_may$ws

TTT\_m\_m\_ws <- m\_d\_TTT\_mar\_may$ws

summary(TTE\_m\_m\_ws)

summary(TTT\_m\_m\_ws)

t.test(TTE\_m\_m\_ws, TTT\_m\_m\_ws)

wilcox.test(TTE\_m\_m\_ws, TTT\_m\_m\_ws)

TTE\_m\_m\_wd <- m\_d\_TTE\_mar\_may$wd

TTT\_m\_m\_wd <- m\_d\_TTT\_mar\_may$wd

summary(TTE\_m\_m\_wd)

summary(TTT\_m\_m\_wd)

t.test(TTE\_m\_m\_wd, TTT\_m\_m\_wd)

wilcox.test(TTE\_m\_m\_wd, TTT\_m\_m\_wd)

TTE\_m\_m\_temp <- m\_d\_TTE\_mar\_may$temp

TTT\_m\_m\_temp <- m\_d\_TTT\_mar\_may$temp

summary(TTE\_m\_m\_temp)

summary(TTT\_m\_m\_temp)

t.test(TTE\_m\_m\_temp, TTT\_m\_m\_temp)

wilcox.test(TTE\_m\_m\_temp, TTT\_m\_m\_temp)

**以下是Rstudio 的代码，对2018年和2019年全年每天的污染数据和温度，风速风向进行了对比区别的t test 和wilcoxon test**

m\_d\_TTN <- subset(m\_d, year == 2019)

TTN\_so2 <- m\_d\_TTN$so2

m\_d\_TTE <- subset(m\_d, year == 2018)

TTE\_so2 <- m\_d\_TTE$so2

summary(TTN\_so2)

summary(TTE\_so2)

t.test(TTN\_so2, TTE\_so2)

wilcox.test(TTN\_so2, TTE\_so2)

TTN\_no <- m\_d\_TTN$no

TTE\_no <- m\_d\_TTE$no

summary(TTN\_no)

summary(TTE\_no)

t.test(TTN\_no, TTE\_no)

wilcox.test(TTN\_no, TTE\_no)

TTN\_no2 <- m\_d\_TTN$no2

TTE\_no2 <- m\_d\_TTE$no2

summary(TTN\_no2)

summary(TTE\_no2)

t.test(TTN\_no2, TTE\_no2)

wilcox.test(TTN\_no2, TTE\_no2)

TTN\_pm2.5 <- m\_d\_TTN$pm2.5

TTE\_pm2.5 <- m\_d\_TTE$pm2.5

summary(TTN\_pm2.5)

summary(TTE\_pm2.5)

t.test(TTN\_pm2.5, TTE\_pm2.5)

wilcox.test(TTN\_pm2.5, TTE\_pm2.5)

TTN\_o3 <- m\_d\_TTN$o3

TTE\_o3 <- m\_d\_TTE$o3

summary(TTN\_o3)

summary(TTE\_o3)

t.test(TTN\_o3, TTE\_o3)

wilcox.test(TTN\_o3, TTE\_o3)

TTN\_ws <- m\_d\_TTN$ws

TTE\_ws <- m\_d\_TTE$ws

summary(TTN\_ws)

summary(TTE\_ws)

t.test(TTN\_ws, TTE\_ws)

wilcox.test(TTN\_ws, TTE\_ws)

TTN\_wd <- m\_d\_TTN$wd

TTE\_wd <- m\_d\_TTE$wd

summary(TTN\_wd)

summary(TTE\_wd)

t.test(TTN\_wd, TTE\_wd)

wilcox.test(TTN\_wd, TTE\_wd)

TTN\_temp <- m\_d\_TTN$temp

TTE\_temp <- m\_d\_TTE$temp

summary(TTN\_temp)

summary(TTE\_temp)

t.test(TTN\_temp, TTE\_temp)

wilcox.test(TTN\_temp, TTE\_temp)

**以下是Rstudio 的代码，对2018年和2019年从3月23日到4,5 月份每天的污染数据和温度，风速风向进行了对比区别的t test 和wilcoxon test**

m\_d\_TTN\_mar\_may <- subset(m\_d, year == 2019 & ((month == 3 & day >= 23) | month == 4 | month == 5))

TTN\_m\_m\_so2 <- m\_d\_TTN\_mar\_may$so2

m\_d\_TTE\_mar\_may <- subset(m\_d, year == 2018 & ((month == 3 & day >= 23) | month == 4 | month == 5))

TTE\_m\_m\_so2 <- m\_d\_TTE\_mar\_may$so2

summary(TTN\_m\_m\_so2)

summary(TTE\_m\_m\_so2)

t.test(TTN\_m\_m\_so2, TTE\_m\_m\_so2)

wilcox.test(TTN\_m\_m\_so2, TTE\_m\_m\_so2)

TTN\_m\_m\_no <- m\_d\_TTN\_mar\_may$no

TTE\_m\_m\_no <- m\_d\_TTE\_mar\_may$no

summary(TTN\_m\_m\_no)

summary(TTE\_m\_m\_no)

t.test(TTN\_m\_m\_no, TTE\_m\_m\_no)

wilcox.test(TTN\_m\_m\_no, TTE\_m\_m\_no)

TTN\_m\_m\_no2 <- m\_d\_TTN\_mar\_may$no2

TTE\_m\_m\_no2 <- m\_d\_TTE\_mar\_may$no2

summary(TTN\_m\_m\_no2)

summary(TTE\_m\_m\_no2)

t.test(TTN\_m\_m\_no2, TTE\_m\_m\_no2)

wilcox.test(TTN\_m\_m\_no2, TTE\_m\_m\_no2)

TTN\_m\_m\_pm2.5 <- m\_d\_TTN\_mar\_may$pm2.5

TTE\_m\_m\_pm2.5 <- m\_d\_TTE\_mar\_may$pm2.5

summary(TTN\_m\_m\_pm2.5)

summary(TTE\_m\_m\_pm2.5)

t.test(TTN\_m\_m\_pm2.5, TTE\_m\_m\_pm2.5)

wilcox.test(TTN\_m\_m\_pm2.5, TTE\_m\_m\_pm2.5)

TTN\_m\_m\_o3 <- m\_d\_TTN\_mar\_may$o3

TTE\_m\_m\_o3 <- m\_d\_TTE\_mar\_may$o3

summary(TTN\_m\_m\_o3)

summary(TTE\_m\_m\_o3)

t.test(TTN\_m\_m\_o3, TTE\_m\_m\_o3)

wilcox.test(TTN\_m\_m\_o3, TTE\_m\_m\_o3)

TTN\_m\_m\_ws <- m\_d\_TTN\_mar\_may$ws

TTE\_m\_m\_ws <- m\_d\_TTE\_mar\_may$ws

summary(TTN\_m\_m\_ws)

summary(TTE\_m\_m\_ws)

t.test(TTN\_m\_m\_ws, TTE\_m\_m\_ws)

wilcox.test(TTN\_m\_m\_ws, TTE\_m\_m\_ws)

TTN\_m\_m\_wd <- m\_d\_TTN\_mar\_may$wd

TTE\_m\_m\_wd <- m\_d\_TTE\_mar\_may$wd

summary(TTN\_m\_m\_wd)

summary(TTE\_m\_m\_wd)

t.test(TTN\_m\_m\_wd, TTE\_m\_m\_wd)

wilcox.test(TTN\_m\_m\_wd, TTE\_m\_m\_wd)

TTN\_m\_m\_temp <- m\_d\_TTN\_mar\_may$temp

TTE\_m\_m\_temp <- m\_d\_TTE\_mar\_may$temp

summary(TTN\_m\_m\_temp)

summary(TTE\_m\_m\_temp)

t.test(TTN\_m\_m\_temp, TTE\_m\_m\_temp)

wilcox.test(TTN\_m\_m\_temp, TTE\_m\_m\_temp)

**以下代码对于2019年和2018年全年的每天的数据做时间序列图(time serious plot)的代码，用于直观可视的对比两年中数据的变化**

# 设置语言环境为英文

Sys.setlocale("LC\_TIME", "English")

library(ggplot2)

# 创建日期列

m\_d\_TTN$date <- as.Date(with(m\_d\_TTN, paste(year, month, day, sep="-")), "%Y-%m-%d")

m\_d\_TTE$date <- as.Date(with(m\_d\_TTE, paste(year, month, day, sep="-")), "%Y-%m-%d")

# 合并数据

data <- rbind(

data.frame(date = m\_d\_TTN$date, so2 = m\_d\_TTN$so2, year = 2019),

data.frame(date = m\_d\_TTE$date, so2 = m\_d\_TTE$so2, year = 2018)

)

# 绘制时间序列图

ggplot(data, aes(x = date, y = so2, color = factor(year))) +

geom\_line() +

labs(title = "SO2 Levels: 2019 vs 2018", x = "Date", y = "SO2(μg/m^3)") +

theme\_minimal() +

scale\_color\_manual(values = c("2019" = "blue", "2018" = "red"), name = "Year") +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%y %b",

limits = as.Date(c("2018-01-01","2019-12-31")))