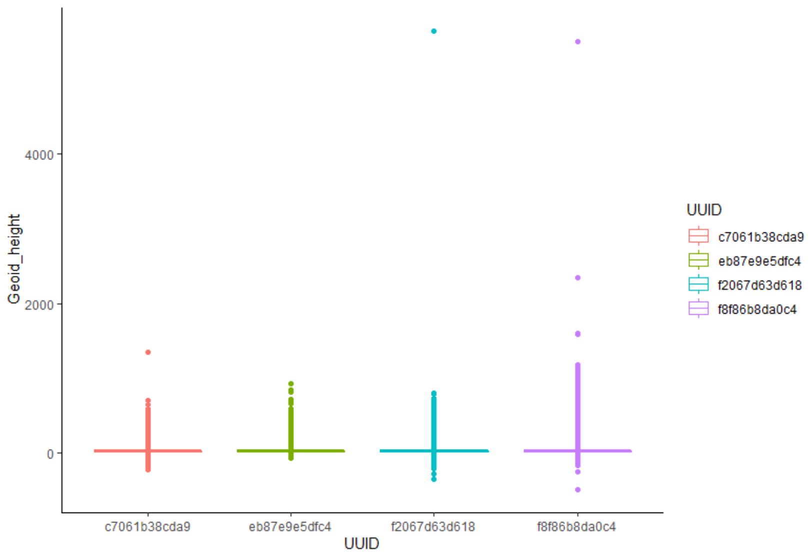
曾海翔 12032760

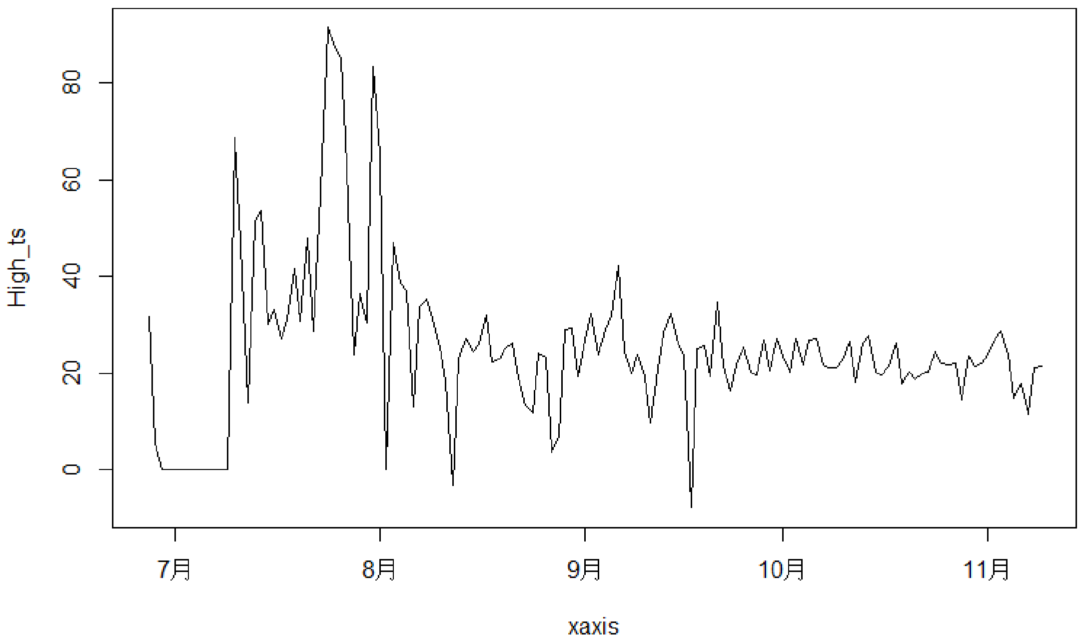
#1

```r
#1
library(tidyr)
library(dplyr)
library(ggplot2)
tracking_data <- read.csv("Tracking_data.csv", header = T)
td_tbl <- as_tibble(tracking_data)
names(td_tbl)

#boxplot flyhing hight average
td_tbl %>%
  filter(Longitude != 200) %>%
  ggplot(aes(x = UUID, y = Geoid_height, color=UUID))+
  geom_boxplot(na.rm = T)+
  theme_classic()

#Time series flying hight
install.packages("lubridate")
library(lubridate)
TS_data <- td_tbl %>%
  group_by(UUID) %>%
  mutate(date_td = as.Date(Collecting_time)) %>%
  filter(Geoid_height >= 0 && UUID =="f8f86b8da0c4")
head(TS_data)
tail(TS_data)
Date_start1 <- as.Date("2020-06-27")
Date_end1 <- as.Date("2020-11-09")
JD_start1   <- yday(Date_start1)
xaxis <- seq(Date_start1, Date_end1, by = "day")
High_ts <- ts(TS_data$Geoid_height[1:136], start=c(2020, JD_start1), frequency = 365)
str(High_ts)
plot(xaxis, High_ts, type = "l")

#Histogram
td_tbl %>%
  group_by(UUID) %>%
  ggplot(aes( x = Speed, color=UUID))+
  geom_histogram()+
  theme_classic()

#Scatter plot
td_tbl %>%
  group_by(UUID) %>%
  filter(Longitude != 0 && Latitude != 200 ) %>%
  ggplot(aes(x = Longitude, y = Latitude, color=UUID))+
  geom_point(na.rm = T)+
  theme_classic()+
  ylim(0,180)+
  xlim(0,180)

#Image plot
```

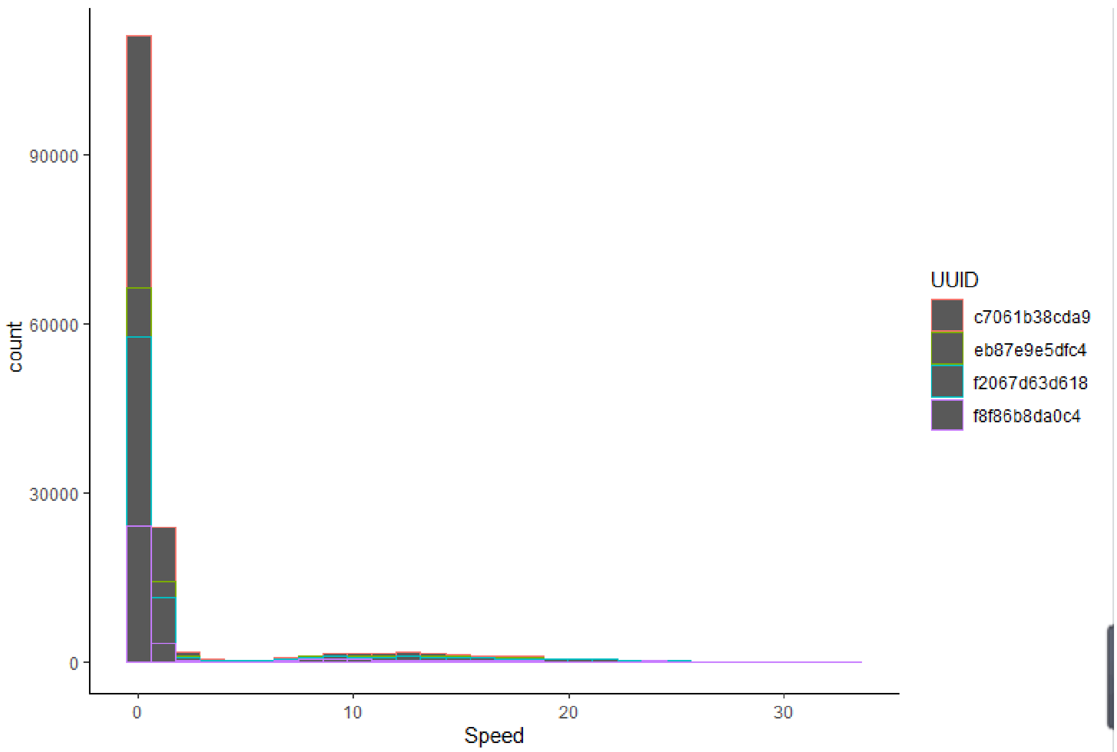
以上是代码



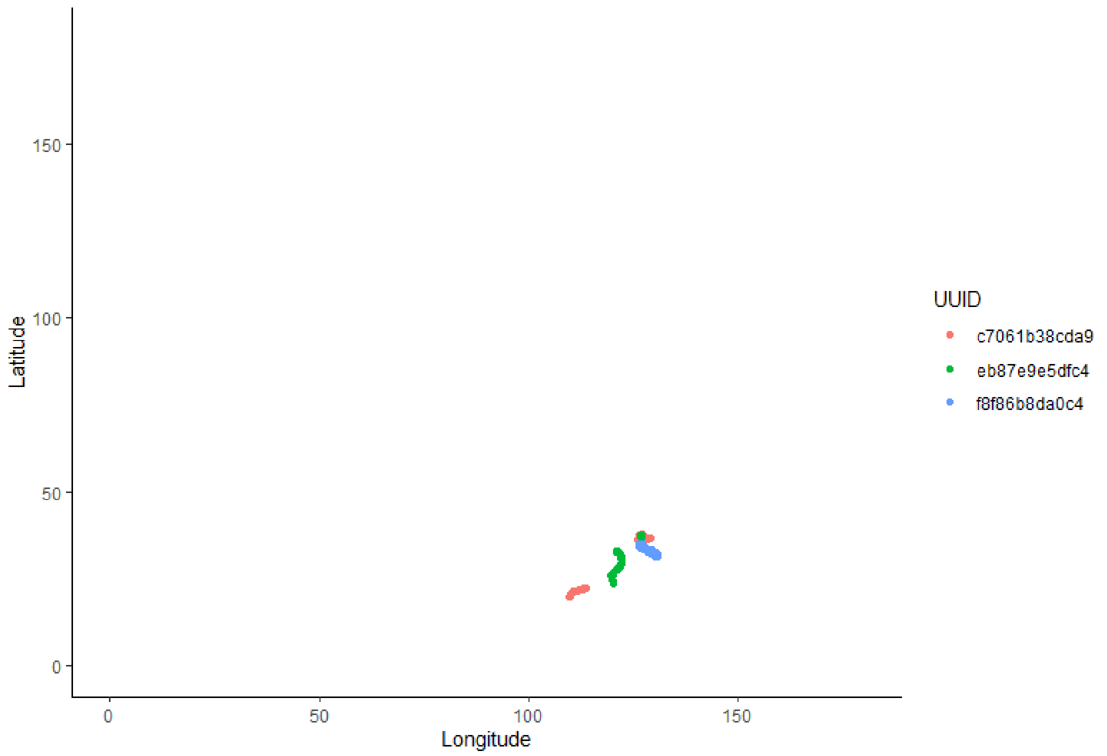
盒状图，可看出四只不同的鸟类个体（应该是同种）活动的高度，多在地面上。



其中一个个体时间序列的飞行高度



柱状图，可看出这种鸟类飞行的速度在 12~20m/s

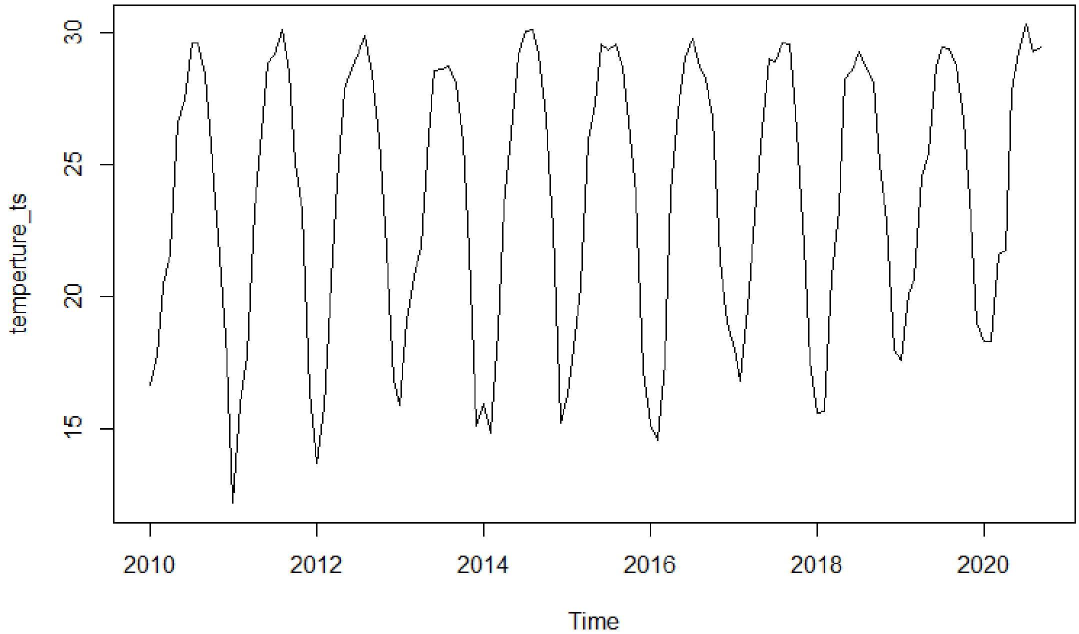


散点图，可能结合到地图上会更好。。之后改进

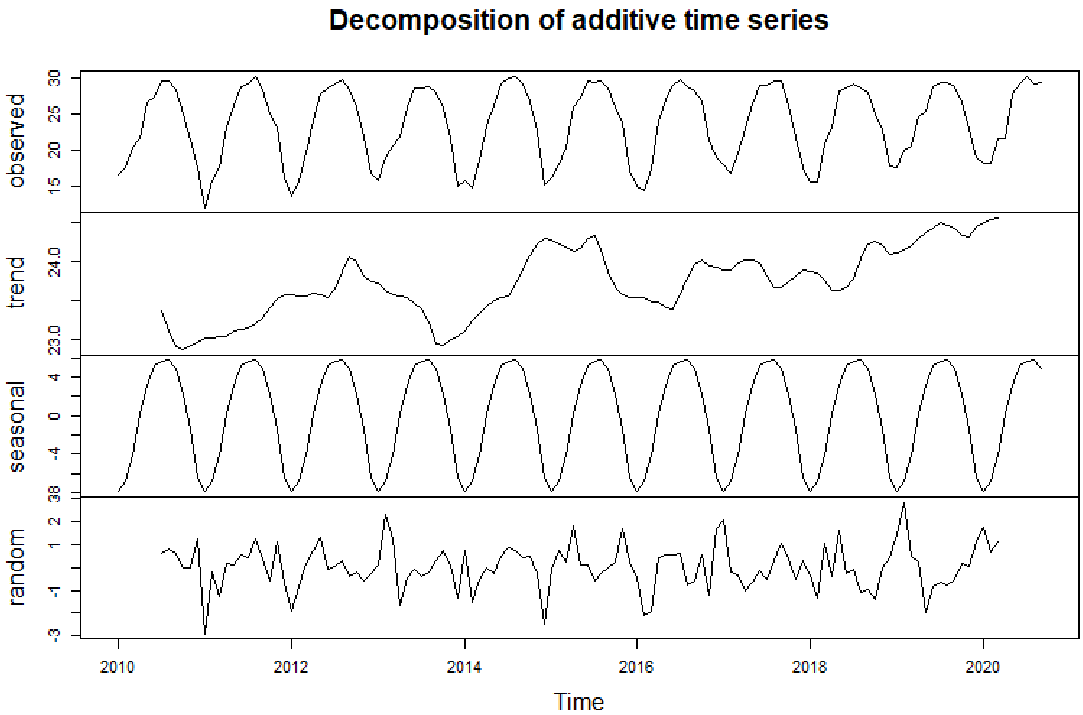
#2

```
#2
baoan_data <- read.csv("2281305.csv",header = T)
library(tidyr)
library(dplyr)
library(ggplot2)

#2.1
banan_data_tbl <- as_tibble(baoan_data)
head(banan_data_tbl)
names(banan_data_tbl)
bdt <- banan_data_tbl %>%
  filter(TMP != "+9999") %>%
  mutate(date1 = substr(DATE,1,7), tmp = as.numeric(substr(TMP,3,5))/10) %>%
  group_by(date1) %>%
  summarise(Tmean = mean(tmp))
temperture_ts <- ts(bdt$Tmean, start=c(2010,1), frequency = 12)
plot(temperture_ts, type = "l")


#2.2
temperture_components <- decompose(temperture_ts)
plot(temperture_components)

#2.3
acf(temperture_ts)
pacf(temperture_ts)
install.packages("forecast")
library(forecast)
model_arima <- auto.arima(temperture_ts)

#2.4
bdt2 <- bdt %>%
  filter(date1 < "2020-08")
temperture_ts2 <- ts(bdt2$Tmean, start=c(2010,1), frequency = 12)
model_arima2 <- auto.arima(temperture_ts2)
meanT_forecast <- 5
month_in_plot <- 30
forecast_2meanT <- forecast(model_arima2, meanT_forecast)
plot(forecast(model_arima2, meanT_forecast), include = month_in_plot, xlab = "time", ylab = "mean_temperture")
#check
bdt %>%
  filter(date1 > "2020-08") #check 2020-08 actual T
forecast_2meanT$mean[1] #Sep. T prediction
forecast_2meanT$mean[2] #Sep. T prediction
```

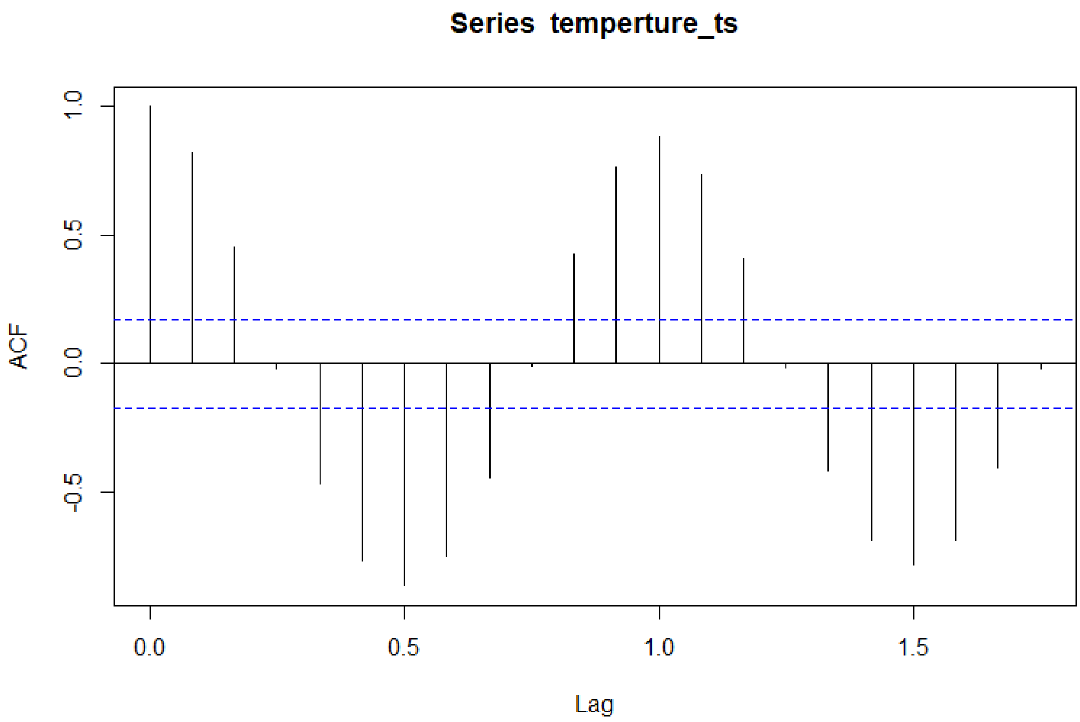
以上是代码

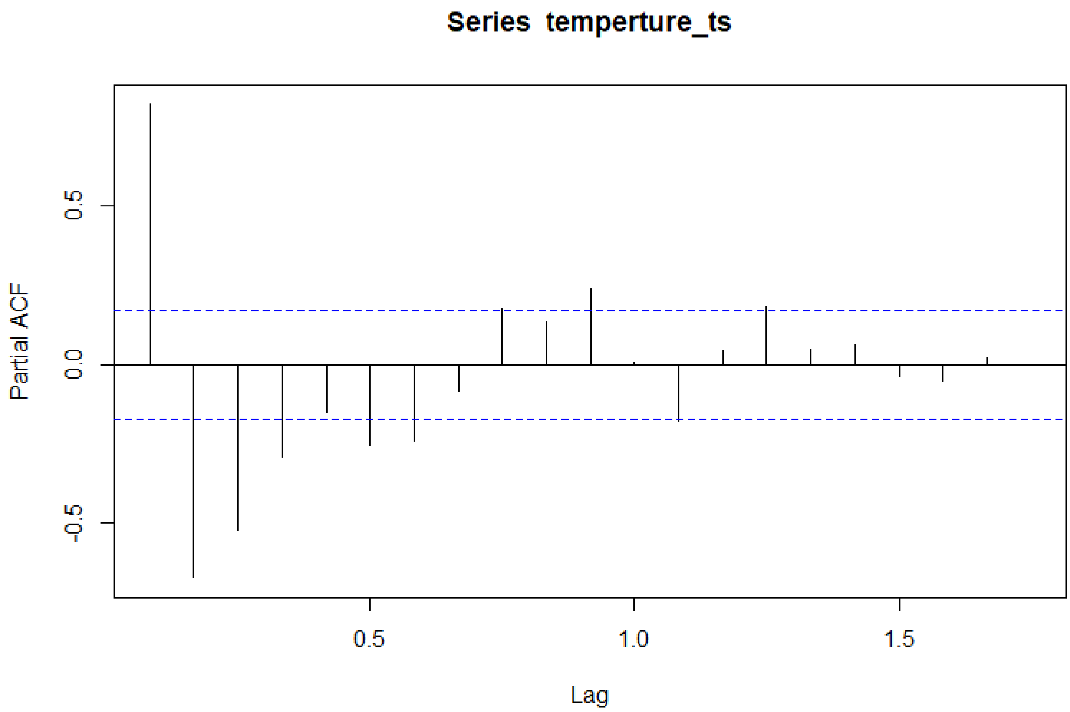


去除丢失值，求平均画图



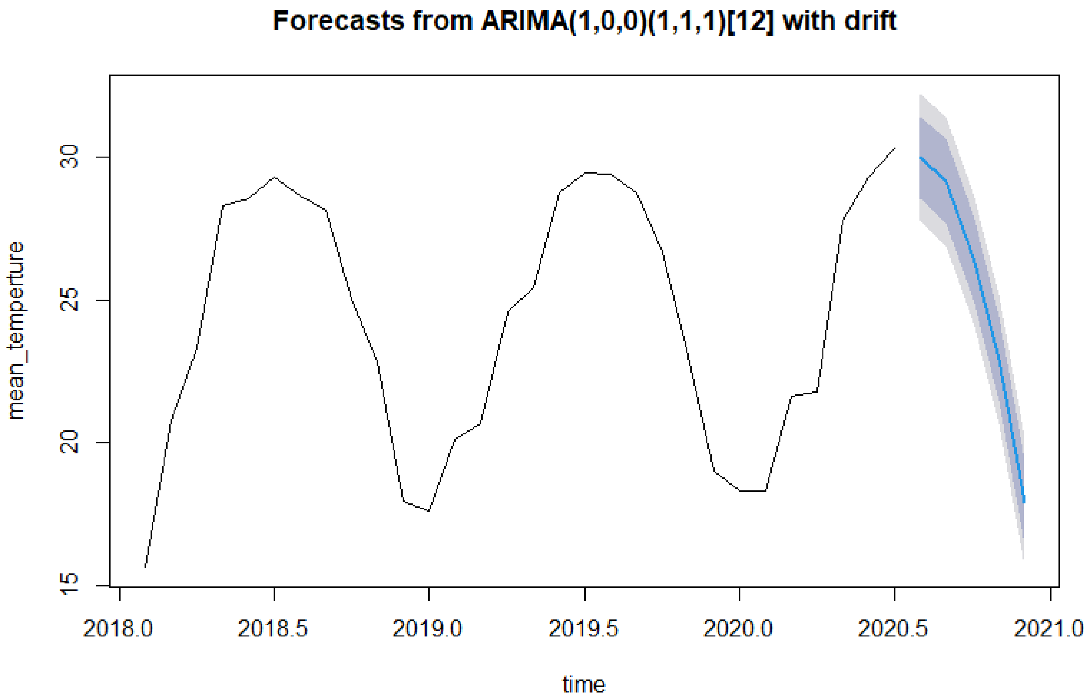
分解





还不是很理解该怎样鉴定 p、d、q 的值，用了自动拟合生成的函数 auto.arima

```
> model_arima
Series: temperture_ts
ARIMA(2,0,2)(0,1,2)[12] with drift

Coefficients:
         ar1      ar2      ma1     ma2     sma1    sma2   drift
      1.2292  -0.7277  -1.0179  0.5665  -0.9055  0.0378  0.0088
s.e.  0.2782   0.1642   0.3084  0.2005   0.1570  0.1231  0.0028

sigma^2 estimated as 1.21:  log likelihood=-182.18
AIC=380.37   AICc=381.7   BIC=402.46
>
```




```
# A tibble: 1 x 2
  date1   Tmean
  <chr>   <dbl>
1 2020-09  29.5
> forecast_2meanT$mean[1] #Sep. T prediction
[1] 29.9871
> forecast_2meanT$mean[2] #Sep. T prediction
[1] 29.16262
> |
```

用八月前的数据做模拟的基础，检验九月份的平均温度