问题：

在doubs河流鱼群研究中，请筛选VOLPla站点有关CHE鱼类生物量和密度的记录，构建表3。其中，第1列为序号，第2列为时间戳（stamp），第3列为生物量（Biomass）。请根据下面问题作答。

|  |  |  |
| --- | --- | --- |
| no | Date | Biomass |
| 1854 | 1994-06-21 | 0.9500713 |
| 1855 | 1995-06-13 | 0.9519654 |
| 1856 | 1996-06-18 | 8.3632565 |
| ︙ | ︙ | ︙ |

1. library(tidyverse) 表3 doubs河1994-2020年CHE鱼类个体生物量
2. library(randomForest)
3. data <- read.table("fishBiomassData.txt ")
4. mydata <- data |>
5. subset(STATION=="VOLPla" & SP == "CHE") |>
6. select(DATE, BIOMASS)
7. ts <- ts(mydata$BIOMASS, start=1994, frequency=1)
8. ts\_org <- window(ts, end = 2018)
9. ts\_trf <- ts.org |> log() |> diff(1)
10. lag\_order <- 2
11. horizon <- 2
12. ts\_mbd <- embed(ts.trd, lag.order + 1)
13. Y\_train <- ts\_mbd[, 1]
14. X\_train <- ts\_mbd[, -1]
15. y\_test <- window(ts, start = 2019, end = 2020)
16. x\_test <- ts\_mbd[nrow(ts\_mbd), c(1:lag\_order)]
17. pred\_rf <- numeric(horizon)
18. for (i in 1:horizon){set.seed(1)
19. fit\_rf <- randomForest(X\_train, y\_train)
20. pred\_rf[i] <- predict(fit\_rf, X\_test)
21. y\_train <- y\_train[-1]
22. X\_train <- X\_train[-nrow(X\_train), ] }
23. pred\_rf
24. exp\_term <- exp(cumsum(pred\_rf))
25. last\_obs <- as.vector(tail(ts\_org, 1))
26. backtrans\_fc <- last\_obs \* exp\_term
27. y\_pred <- ts(backtrans\_fc, start = 2019, frequency = 1)
28. library(fpp2)
29. forecast::accuracy(as.numeric(y\_pred), as.numeric(y\_test))
30. ts\_fc <- cbind(ts,pred = c(rep(NA, length(ts\_org)), y\_pred))
31. plot\_fc <- ts\_fc |> autoplot() + theme\_minimal()
32. plot\_fc
33. **这是构建机器学习时间序列预测模型，根据代码，请简述时间序列预测建模基本原理以及基本过程。**

**解答：**

1. **时间序列预测建模基本原理**

时间序列预测建模的基本原理：通过分析历史时间序列数据，寻找序列中的时间依赖性（即某点观测值与过去值之间的相关关系），分解出序列中的趋势、季节性和周期性行为等模式，构建模型，利用上述时间依赖性和模式对该时间序列的未来值进行预测。

1. **基本过程**

* **数据读取与筛选**

1. data <- read.table("fishBiomassData.txt ")
2. mydata <- data |>
3. subset(STATION=="VOLPla" & SP == "CHE") |>
4. select(DATE, BIOMASS)

选择特定站点（VOLPla）和物种（CHE）的生物量数据，提取时间（DATE）与生物量（BIOMASS）字段。

* **时间序列创建与预处理**

1. ts <- ts(mydata$BIOMASS, start=1994, frequency=1)
2. ts\_org <- window(ts, end = 2018)
3. ts\_trf <- ts.org |> log() |> diff(1)

将数据转为年度频率的时间序列（ts），截取至2018年为训练集（ts\_org）；

对训练集进行对数转换与一阶差分，处理非平稳性，获得处理后的序列（ts\_trf）。

* **训练矩阵构建**

1. lag\_order <- 2
2. horizon <- 2
3. ts\_mbd <- embed(ts.trd, lag.order + 1)

lag\_order <- 2表示构建滞后特征时使用的历史值数量，horizon <- 2 表示要预测未来的2个值；使用embed() 函数构造滞后特征矩阵，以捕捉历史值对当前值的影响，lag.order + 1 = 3，每行有3个值：当前值 + 滞后1期 + 滞后2期。

* **分割训练集和测试集**

1. Y\_train <- ts\_mbd[, 1]
2. X\_train <- ts\_mbd[, -1]
3. y\_test <- window(ts, start = 2019, end = 2020)
4. x\_test <- ts\_mbd[nrow(ts\_mbd), c(1:lag\_order)]

ts\_mbd是通过embed()函数生成的嵌套矩阵，ts\_mbd[, 1]的作用是提取训练集的目标变量（y），第一列 [, 1] 是当前值，即我们要预测的目标变量，作为模型训练用的 预测因子矩阵； ts\_mbd[, -1]的作用是提取训练集的自变量（X），

[, -1] 表示去掉第一列，保留滞后变量列，得到的是每个时间点的前 lag\_order 个值，作为模型输入特征。

window(ts, start = 2019, end = 2020)是为了提取测试集的真实值（用于评估），从原始的ts时间序列中，截取2019到2020年的实际观测值，用于和预测结果进行对比，计算误差。ts\_mbd[nrow(ts\_mbd)， c(1:lag\_order)]则是为了提取测试集预测所需的初始输入值，ts\_mbd[nrow(ts\_mbd), ]取的是训练集中最后一行，表示 最新的一组数据。c(1:lag\_order)取的是第1列到第lag\_order列。

### **使用随机森林训练模型**

1. pred\_rf <- numeric(horizon)
2. for (i in 1:horizon){set.seed(1)
3. fit\_rf <- randomForest(X\_train, y\_train)
4. pred\_rf[i] <- predict(fit\_rf, X\_test)
5. y\_train <- y\_train[-1]
6. X\_train <- X\_train[-nrow(X\_train), ] }
7. pred\_rf

使用随机森林对训练集进行拟合，并滚动预测两个时间点（2019和2020年），每次预测后移除最旧数据，模拟递推预测。

* **预测结果反变换**

1. exp\_term <- exp(cumsum(pred\_rf))
2. last\_obs <- as.vector(tail(ts\_org, 1))
3. backtrans\_fc <- last\_obs \* exp\_term
4. y\_pred <- ts(backtrans\_fc, start = 2019, frequency = 1)

对数差分预测结果通过exp()和cumsum()反变换为原始尺度（即恢复为生物量单位）

* **结果评估与可视化**

使用accuracy()评估预测效果，并通过autoplot()可视化预测与原始数据对比。

1. **除了基于原始值（生物量）构造滞后特征外，还可从时间戳提取并构造建模特征，请简述从时间戳构造建模特征的依据是什么？**

**解答：**

（1）日期时间特征（date time features）。即观察值与具体日期或季节有关，如日低温与雨季等有重要关系，可以选择把日期或季节作为特征。

（2）滞后特征（lag features）。如今年的11月份数据与以往年份的11月份数据更相关，即要关注历史上的今天。

（3）窗口特征（window features）。建模精度不仅与采用或选择的滑动窗口大小密切相关，而且与窗口内的均值、最大值等具体数据也有关。