1. 接上一部需求文档【多源数据整合模块开发需求】，多源数据类型及excel样例如下表所所示，数据汇总结果如下图所示。本需求文档中的性能预测、养护决策均基于数据汇总表进行。

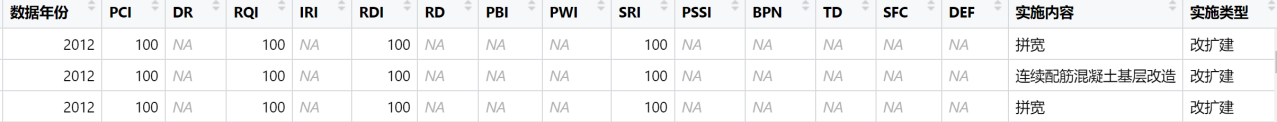
|  |  |
| --- | --- |
| **多源数据类型** | **Excel样例** |
| 静态属性 | 道路静态数据-运营集团.xlsx |
| 历年性能指标 | 初始动态数据.xlsx |
| 工程历史 | 工程历史.xlsx |
| 环境荷载 | / |
| 交通荷载（流量） | 断面交通流量.xlsx |
| 交通荷载（轴重分布） | 车道级车辆类型分布.xlsx |

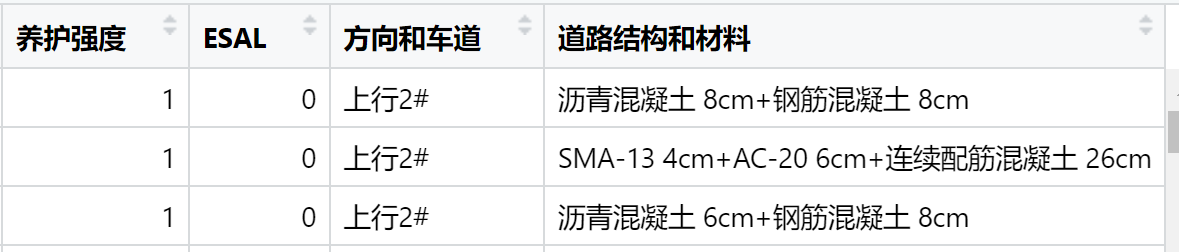
数据汇总表











1. 【绘制每个道路最小单元的性能曲线图】：根据路名、方向、车道编号、起点桩号（根据数据汇总表，起点桩号确定，止点桩号也唯一对应）、性能指标类型，提取其历年性能指标，并将道路结构与材料，与数据年份对应的累计当量轴载作用次数、工程历史也一并附入图中，参考代码如下所示。

plot\_ppi <- function(data,name,direction,lane\_no,start,ppi){

data <- filter(data,路线名称==name & 方向==direction & 车道编号==lane\_no & 起点桩号==start)

eval(parse(text = paste("plot(data$数据年份,data$",ppi,",xlab='',ylab='",ppi,"')",sep = "")))

eval(parse(text = paste("text(x=data$数据年份,y=data$",ppi,"+2,labels=gsub('NA','',paste(data$实施内容,data$养护强度,sep='\n')),xpd=TRUE,cex=0.7)",sep = "")))

eval(parse(text = paste("text(x=data$数据年份,y=data$",ppi,"-1,labels=gsub('NA','',data$ESAL),xpd=TRUE,cex=0.5)",sep = "")))

tmp <- paste(data$上面层材料[1],' ',data$上面层厚度[1],"cm+",data$中面层材料[1],' ',data$中面层厚度[1],"cm+",data$下面层材料[1],' ',data$下面层厚度[1],"cm+",data$上基层材料[1],' ',data$上基层厚度[1],"cm+",data$中基层材料[1],' ',data$中基层厚度[1],"cm+",data$底基层材料[1],' ',data$底基层厚度[1],"cm+",data$垫层材料[1],' ',data$垫层厚度[1],"cm+",data$土基类型[1],sep='')

tmp <- gsub("+NA ","",tmp,fixed = TRUE)

tmp <- gsub("+NA","",tmp,fixed = TRUE)

tmp <- gsub("NAcm","",tmp,fixed = TRUE)

mtext(tmp, side = 3, line = 1.5,cex=0.8)

tmp <- paste(name," ",direction," ",lane\_no,"# K",start,'-',data$止点桩号[1],sep='')

mtext(tmp, side = 3, line = 2.5)

mtext("数据年份", side = 1, line = 2)

# mtext(paste(data$ESAL,collapse = " "), side = 1, line = 3,cex=0.8)

# mtext("累计当量轴载", side = 1, line = 4)

}

par(mar=c(3.5,4.5,4,1))

plot\_ppi(data = data,"S20外环高速（浦西段）","上行",2,79.5,"PCI")#输入路名/方向/车道编号/起点桩号/指标名称

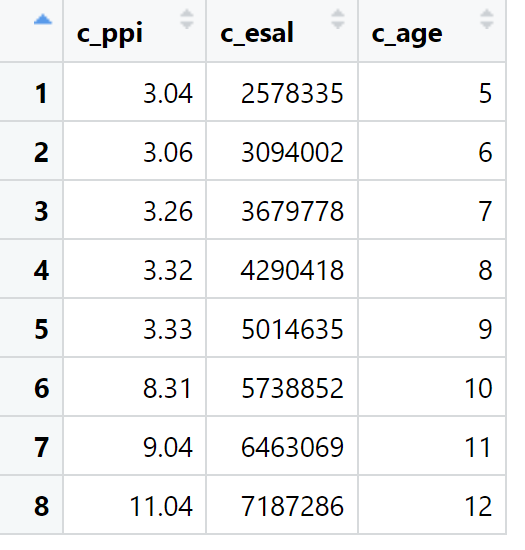
功能一：

绘制效果如下两图所示。目标是在界面中输入路名、方向、车道编号、起点桩号、性能指标类型【对应于决策树的指标类型】，自动绘制如下图所示的图，在界面上展示。横坐标是数据年份，纵坐标是性能指标类型，标题包括路名、方向、车道编号、起点桩号、止点桩号，道路结构和材料类型。每个散点的上方显示当年度的实施的工程措施（如果有），下方显示截至当年的累计当年轴载作用次数（ESAL字段）。



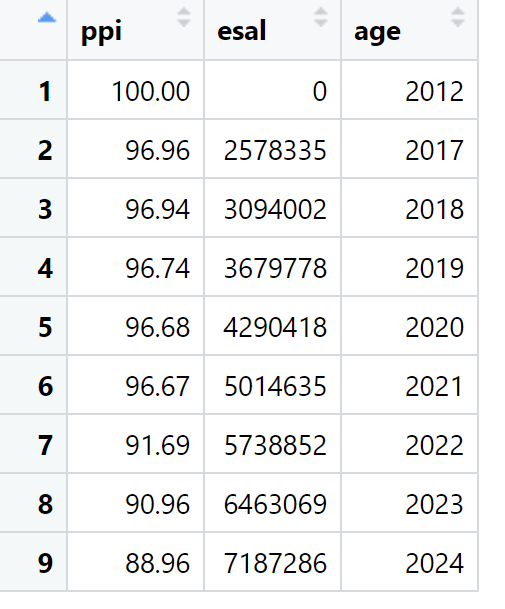
1. 【从上图中提取性能衰变信息】：对于汇总表（data数据集），以（'路线代码','方向','车道编号','起点桩号','止点桩号','技术等级','路面类型','道路结构和材料'）这8个字段为区分，如有不同，则为不同的最小单元。对于切分出来的每个最小单元，将其性能指标（以PCI为例）、ESAL（累计当年轴载作用次数）、数据年份这3个向量组成矩阵（tmp2）。去掉这个矩阵中缺失的行（某些年份的数据可能缺失，需要过滤掉），然后按数据年份升序排列，然后对于第一列（性能指标列）做差分，小于0的为单调衰变段，以bool值标记出来，记为tmp3。如果存在单调衰变段（sum(tmp3)>=1），则将tmp中第一列（性能指标）、第二列（ESAL）、第三列（数据年份）分别差分，取tmp3==TRUE的行，合并组成新的矩阵table，然后对第一列取负（转换为正值），然后对每列求累加（cumsum），即得到了这个单元的衰变速率分析底表（table）。以上图为例，该底表如下表所示。

【功能二】界面提供随便速率的分析底表

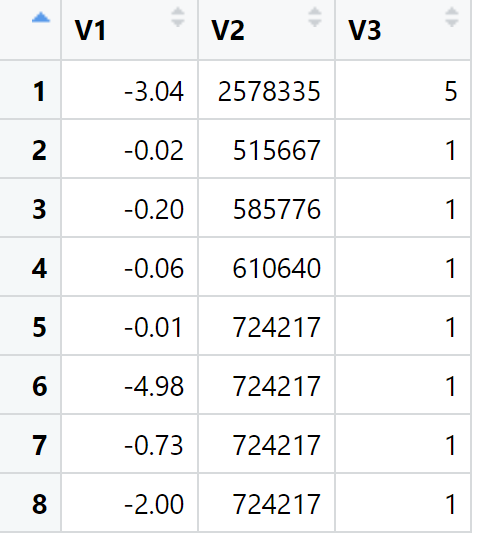


上表的详细生成过程如下：

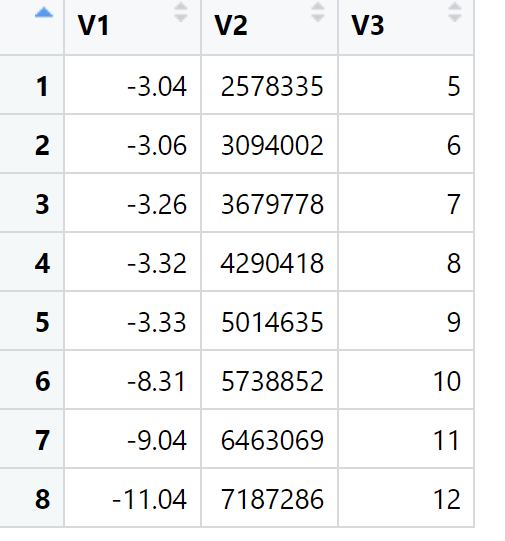
（1）将ppi、esal、age原始数据整理到矩阵里



1. 由于上述数据中无缺失值，因此，移除缺失值后数据仍如上所示
2. 按路龄age升序排列（上图中已按age升序排列）
3. 对于上述矩阵，取单调递减部分信息。具体地，先找出ppi的差分大于0的位置（tmp3=diff(ppi)>0），然后取上述矩阵的差分在这些位置（tmp3）上的值。由于上图中没有ppi增长部分，因此取出来就是上述数据的差分值，结果如下所示。注意，由于进行了差分操作，数据少了一行。



1. 然后，对上图各列做累加运算（cumsum），得到衰变速率分析表，如下所示。



1. 再对第一列取负（转为正），即得到所要的最终结果。

**整个过程的伪代码如下所示。**

##从每个最小单元中提取单调衰变信息

tmp <- plyr::dlply(data,c('路线代码','方向','车道编号','起点桩号','止点桩号','技术等级','路面类型','道路结构和材料'),reframe,x=deterioration\_analysis\_table(PCI,ESAL,数据年份))#要分析的指标在此选择

#对于每个基本单元,取历年数据,作分析底图,形成衰变速率分析底表

deterioration\_analysis\_table <- function(ppi,esal,age){

tmp <- which(!is.na(ppi))

if(!identical(tmp,integer(0))){

tmp2 <- cbind(ppi[tmp],esal[tmp],age[tmp])#移除缺失值

tmp2 <- tmp2[order(tmp2[,3]),]#按路龄排序

tmp3 <- diff(tmp2[,1])<0

if(sum(tmp3)>=1){

table <- cbind(diff(tmp2[,1])[tmp3],diff(tmp2[,2])[tmp3],diff(tmp2[,3])[tmp3])

table <- apply(table,2,cumsum)#衰变速率分析底表

return({matrix(table,ncol = 3)})

}else{

return(NA)

}

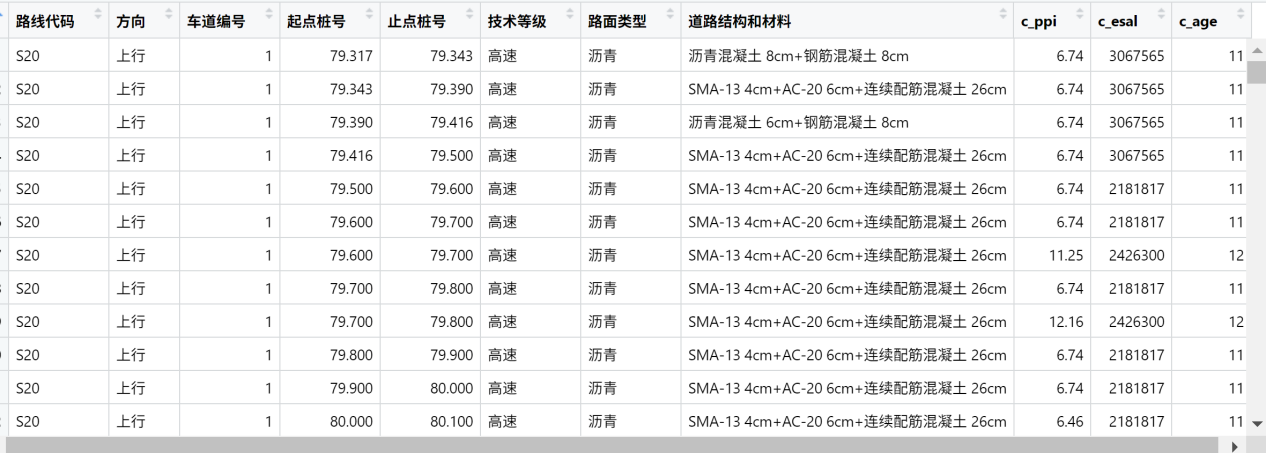
}else{NA}

}

需要注意的是，不是每个单元的性能指标都是单调下降的，如下图所示，但上述算法仍然可行。增量段在未来算法中考虑。



1. 将通过3.所述过程得到的每个最小单元的衰变速率分析表分别加上表头（'路线代码','方向','车道编号','起点桩号','止点桩号','技术等级','路面类型','道路结构和材料'），然后按行拼接成一张大表，为衰变信息表（data\_deterioration），如下表所示。



1. 【分阶段多元线性回归中的阶段划分】基于衰变信息表（data\_deterioration），进行分阶段多元线性回归。其中，准备工作的第一步是阶段划分，逻辑如下所示。对于衰变信息表（data\_deterioration）中的某个指标（以PCI为例，该向量为c\_ppi），设定其拟划分为的段数n-1（阈值点个数为n），使每两个相邻阈值点区间的样本数量相同。然后将第一个（最小）阈值点（最低值）强制设置为0，最后一个（最大）阈值点强制设置为当前最大阈值点+1，即完成阶段划分，不同道路结构和材料的阶段划分结果存储到ppi\_thresholds里。伪代码如下所示。

#计算不同道路结构和材料的ppi\_thresholds

ppi\_thresholds <- plyr::dlply(data\_deterioration,c('道路结构和材料'),reframe,x=divide\_stage(c\_ppi,4))

#阈值区间划分函数

divide\_stage <- function(c\_ppi,n){

#c\_ppi是累计ppi降低量;n是初步设定的阈值点个数

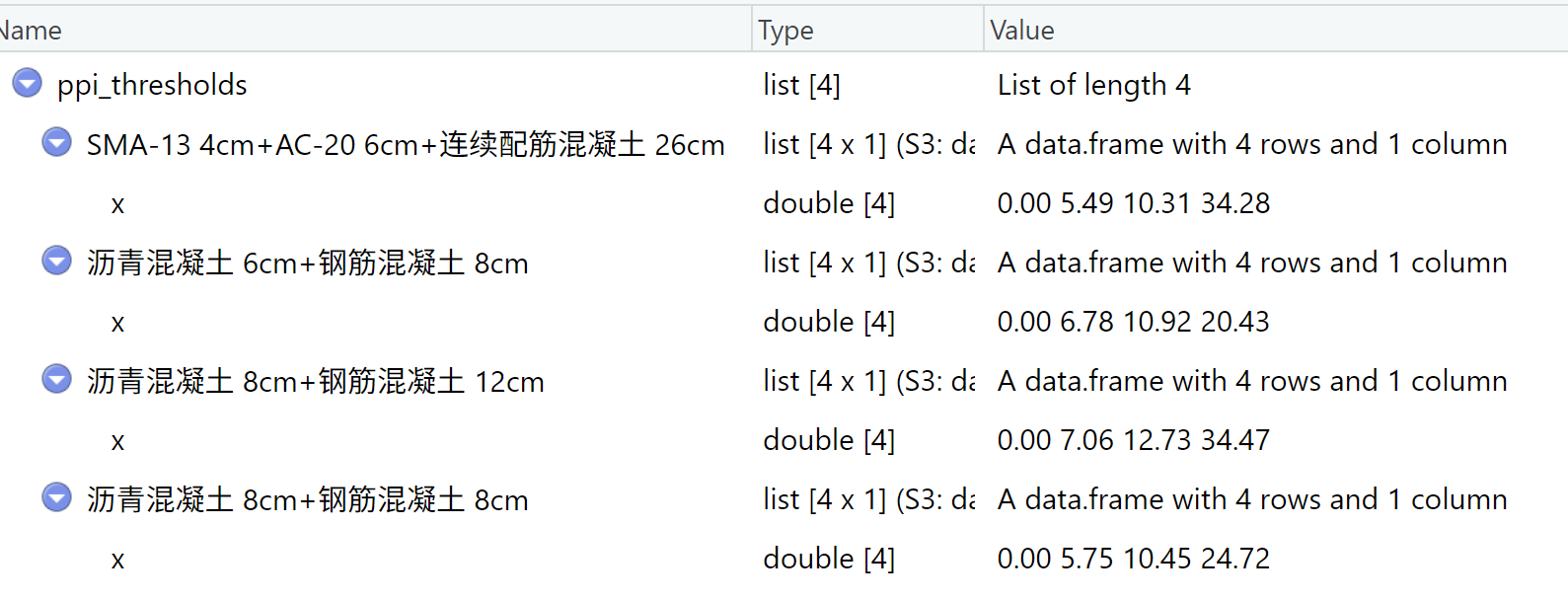
ppi\_threshold <- quantile(c\_ppi,probs = seq(0,1,length.out=n),na.rm = TRUE);#划分成等数据量的n-1个区间

ppi\_threshold <- ppi\_threshold[-1];ppi\_threshold[length(ppi\_threshold)] <- ppi\_threshold[length(ppi\_threshold)]+1;ppi\_threshold <- union(0,ppi\_threshold);#标准化区间起点为0,止点为最大值+1

ppi\_threshold <- ppi\_threshold[order(ppi\_threshold)]#按升序划分区间

}

S20各种道路结构和材料下性能指标（以PCI为例）的阈值区间如下所示。对于路基段结构SMA-13 4cm+AC-20 6cm +连续配筋混凝土26cm，三阶段的阈值分别为[0,5.49), [5.49,10.31), [10.31,34.28]。



1. 【分阶段多元线性回归中的多元转单元转化系数】对于衰变信息表（data\_deterioration），直接进行多元线性回归是不可行的，需要将多元变量转换为单元变量，然后进行回归，以保证回归系数为正。对于本模型，根据实际情况，以环境荷载（路龄，Age）为基准变量，其它变量（ESAL，累计当量轴载作用次数）乘以一个转换系数（k1/k2）转换为基准变量。这个转换系数有个搜索范围，确定的方法如下所示。对于衰变信息表（data\_deterioration）中的每个道路最小单元（路线代码,方向,车道编号,起点桩号,止点桩号,技术等级,路面类型,道路结构和材料相同的单元），对其c\_ppi，c\_esal，c\_age组成的矩阵，按c\_age升序排列（经过之前的步骤已经完成此排序），取最后一个c\_ppi除以c\_esal即为k1，取最后一个c\_ppi除以c\_age即为k2。然后对于每种道路结构和材料，取其最小、最大k1，最小、最大k2，形成k1、k2取值范围表k\_range。示例代码如下所示。

tmp <- setDT(data\_deterioration)[, as.list(c(val = k\_range\_calculation(c\_ppi,c\_esal,c\_age))), .(路线代码,方向,车道编号,起点桩号,止点桩号,技术等级,路面类型,道路结构和材料)]

k\_range\_calculation <- function(c\_ppi,c\_esal,c\_age){

tmp <- c(k1=c\_ppi[length(c\_ppi)]/c\_esal[length(c\_esal)],k2=c\_ppi[length(c\_ppi)]/c\_age[length(c\_age)])

return(tmp)

}

k\_range <- plyr::ddply(tmp,c('道路结构和材料'),summarise,k1.min=min(val.k1),k1.max=max(val.k1),k2.min=min(val.k2),k2.max=max(val.k2))

对于S20案例，k\_range表如下所示。



1. 【开启分阶段多元线性回归的并行计算】：由上表可知，转换系数k1/k2的最大值为k1.max/k2.min，最小值强制从0开始（0代表无需进行多元回归）。但是通常无需搜索至最大值，因为这个最大值几乎不可能取到，实际搜索的最大值可以比这个理论最大值小得多，此处建议这个比例取0.03，搜索间隔为0.0005。当然，这两个参数可设置。即，k\_range\_quantile=seq(from=0,to=0.03,by=0.0005)。这个搜索过程比较慢，请务必开启并行计算，伪代码如下所示。并行计算的结果按行合并（rbind）。

#对于不同的k1\_k2\_ratio,并行计算

myCluster <- makeCluster(detectCores()-2, type = "PSOCK")#计算所用的核心数

registerDoParallel(myCluster)#注册计算

##tmp4即为衰变模型,包含不同k1\_k2\_ratio/不同道路结构和材料/不同stage下的衰变速率信息

tmp4 <- foreach (k\_range\_quantile=seq(0,0.03,0.0005),.combine = rbind,.packages = c("data.table","dplyr")) %dopar% {

tmp3 <- setDT(data\_deterioration)[, as.list(c(val = multi\_stage\_multiple\_linear\_regression(道路结构和材料,ppi\_thresholds,k\_range,k\_range\_quantile,c\_ppi,c\_esal,c\_age))), .(路线代码,方向,车道编号,起点桩号,止点桩号,技术等级,路面类型,道路结构和材料)]

}

stopCluster(myCluster)

names(tmp4) <- gsub(fixed=TRUE,"val.","",names(tmp4));tmp4 <- plyr::rename(tmp4,replace=c("c\_age\_e"="k\_d\_age\_e"))#列名称替换

1. 【正式进行分阶段多元线性回归】：对于每个并行计算单元，不同的参数是转换系数k1/k2，记为k1\_k2\_ratio，从范围k\_range\_quantile中取值。对于衰变信息表（data\_deterioration），对于不同的最小单元（路线代码,方向,车道编号,起点桩号,止点桩号,技术等级,路面类型,道路结构和材料不同），分别进行多元线性回归，后续再汇总处理。首先，根据本最小单元的道路结构与材料，去查ppi\_thresholds表，得到道路性能指标的阶段划分各阈值点ppi\_threshold。然后，根据并行计算单元中分配的分位数k\_range\_quantile，对照k\_range表，对k1.max/k2.min乘以这个分位数k\_range\_quantile，即可得到本并行单元中的转换系数k1\_k2\_ratio。然后，将本单元的累计性能指标衰变值（c\_ppi）、累计当量轴载作用次数（c\_esal），累计路龄（c\_age）拼为一个矩阵data。对data的每一行数据，判断其所在阶段stage，判断函数如judge\_stage所示。然后，在data中插入每个阶段的阈值点数据。然后，对于每个阶段，在data中提取出对应的部分（矩阵），将所有行的值减去第一行的值以初始化，再进行单阶段多元线性回归。最后，对每行回归结果附上本阶段起点阈值、终点阈值、对应的k1\_k2\_ratio信息，得到总的回归结果。该过程的伪代码如下所示。

#分阶段回归函数

multi\_stage\_multiple\_linear\_regression <- function(road\_structure\_material,ppi\_thresholds,k\_range,k\_range\_quantile,c\_ppi,c\_esal,c\_age){

tmp <- which(names(ppi\_thresholds)==road\_structure\_material[1])#确定对应道路结构和材料的ppi\_thresholds

tmp2 <- which(k\_range$道路结构和材料==road\_structure\_material[1])#确定对应道路结构和材料的k1/k2上下限

if(!identical(tmp,integer(0)) & !identical(tmp2,integer(0))){

ppi\_threshold <- ppi\_thresholds[[tmp]]$x

k1\_k2\_ratio <- quantile(c(0,k\_range[tmp2,"k1.max"]/k\_range[tmp2,"k2.min"]),k\_range\_quantile)#根据k\_range\_quantile确定k1\_k2\_ratio

data <- data.frame(c\_ppi,c\_esal,c\_age)#将数据整理到数据框中

data$stage <- judge\_stage(data$c\_ppi,ppi\_threshold)#阶段划分

if(data[1,1]!=0){data <- rbind(c(0,0,0,1),data)}#插入生命周期起点数据

tmp3 <- data.frame(c\_ppi=ppi\_threshold[-1],c\_esal=NA,c\_age=NA,stage=2:length(ppi\_threshold),front=NA,back=NA)#插入其它阶段起点数据

tmp3 <- tmp3[tmp3$c\_ppi<max(data$c\_ppi),]#自动去除大于等于data中最大c\_ppi的阈值

if(nrow(tmp3>0)){#不止一个阶段的情形,需要插入分阶段数据

tmp3$back <- sapply(tmp3$c\_ppi,function(x){which.max(x-data$c\_ppi<0)});tmp3$front <- tmp3$back-1#插入到data中的位置

#线性插值填补c\_esal/c\_age数据

tmp3$c\_esal <- data[tmp3$front,"c\_esal"]+(tmp3$c\_ppi-data[tmp3$front,"c\_ppi"])/(data[tmp3$back,"c\_ppi"]-data[tmp3$front,"c\_ppi"])\*(data[tmp3$back,"c\_esal"]-data[tmp3$front,"c\_esal"])

tmp3$c\_age <- data[tmp3$front,"c\_age"]+(tmp3$c\_ppi-data[tmp3$front,"c\_ppi"])/(data[tmp3$back,"c\_ppi"]-data[tmp3$front,"c\_ppi"])\*(data[tmp3$back,"c\_age"]-data[tmp3$front,"c\_age"])

data <- rbind(data,tmp3[,1:4])

data <- data[order(data$c\_ppi),]#按c\_ppi升序排序

}

data <- setDT(data)[, as.list(c(val = single\_stage\_regression(k1\_k2\_ratio,c\_ppi,c\_esal,c\_age))), .(stage)]#各阶段分别回归

if(nrow(data)==0){

return(NULL)

}else{

# names(data) <- c("stage","k\_age","k1\_k2\_ratio")

data$interval\_start <- ppi\_threshold[data$stage];data$interval\_end <- ifelse(data$stage==length(ppi\_threshold),100,ppi\_threshold[data$stage+1]);data$k1\_k2\_ratio <- k1\_k2\_ratio

return(data)

}

}else{

return(NULL)

}

}

具体地，该过程的示例如下所示：

1. data中切分下来的某样例道路最小单元数据如下所示：



1. 根据【道路结构和材料】在ppi\_thresholds中查表。本最小单元的各阶段阈值点ppi\_threshold = 0.00 5.49 10.31 34.28，伪代码如下所示：

tmp <- which(names(ppi\_thresholds)==road\_structure\_material[1])#确定对应道路结构和材料的ppi\_thresholds

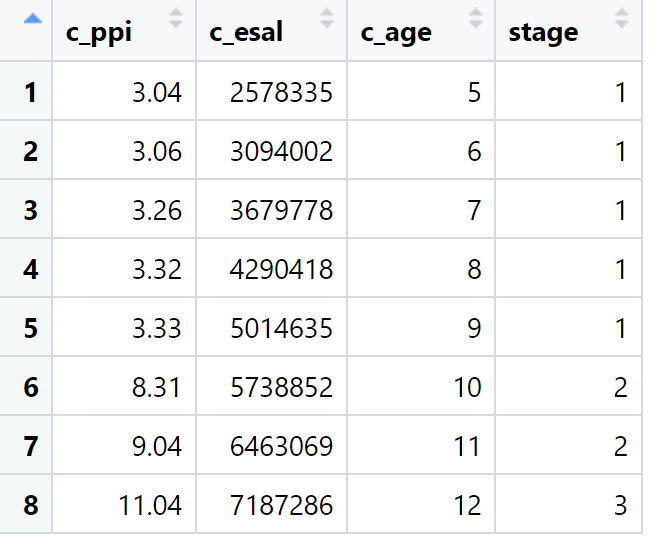
ppi\_threshold <- ppi\_thresholds[[tmp]]$x

1. 根据【道路结构和材料】在k\_range中查表，得到k1/k2的上限，然后再对这个上限值乘以k\_range\_quantile（并行计算中分配到的这个上线的百分比值，此案例中为0.01），即得到了ESAL到Age的转换系数k1\_k2\_ratio=1.575093e-06，伪代码如下所示：

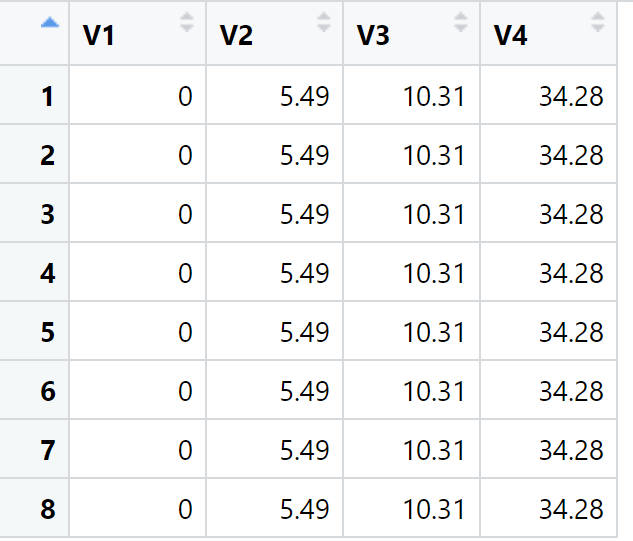
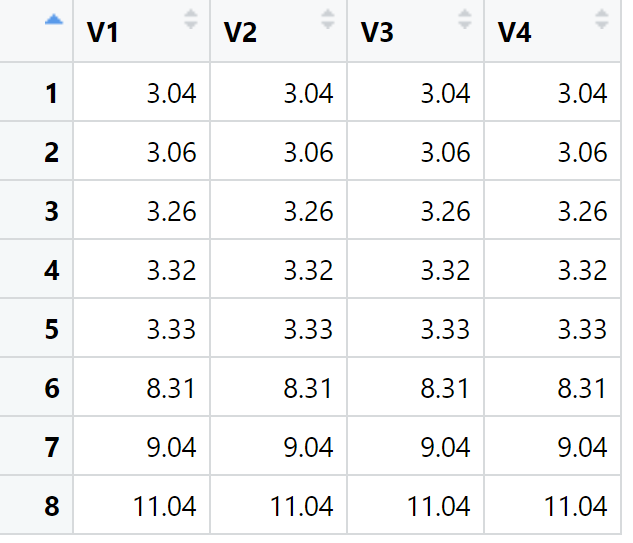
tmp2 <- which(k\_range$道路结构和材料==road\_structure\_material[1])#确定对应道路结构和材料的k1/k2上下限

k1\_k2\_ratio <- quantile(c(0,k\_range[tmp2,"k1.max"]/k\_range[tmp2,"k2.min"]),k\_range\_quantile)#根据k\_range\_quantile确定k1\_k2\_ratio

1. 将本单元的累计性能指标衰变值（c\_ppi）、累计当量轴载作用次数（c\_esal），累计路龄（c\_age）拼为一个矩阵data。然后根据c\_ppi和ppi\_threshold进行阶段划分。阶段划分函数为judge\_stage。judge\_stage中，先将ppi扩充到n列（n为阶段划分总数），然后将阶段阈值向量扩充到同c\_ppi长度一样的行数，后者减前者小于等于0的数量就是所在的阶段数。本过程的结果及伪代码如下所示。



拼成的矩阵data



计算所在阶段数涉及的两个过程矩阵（前面为c\_ppi扩充后的矩阵，后面为ppi\_threshold扩充后的矩阵）

data <- data.frame(c\_ppi,c\_esal,c\_age)#将数据整理到数据框中

data$stage <- judge\_stage(data$c\_ppi,ppi\_threshold)#阶段划分

#判断ppi对应的阶段

judge\_stage <- function(c\_ppi,ppi\_threshold){

# ppi\_threshold是分阶段的ppi值上限

n <- length(ppi\_threshold)#多少个阶段

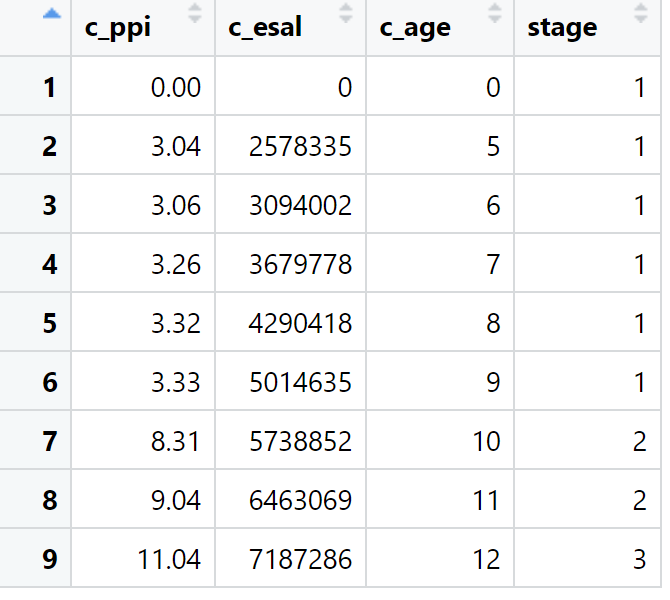
tmp <- matrix(data = rep(c\_ppi,n),ncol = n)

tmp2 <- matrix(data = rep(ppi\_threshold,length(c\_ppi)),ncol = n,byrow = TRUE)

tmp <- rowSums(tmp2-tmp<=0)#ppi分别在哪个阶段

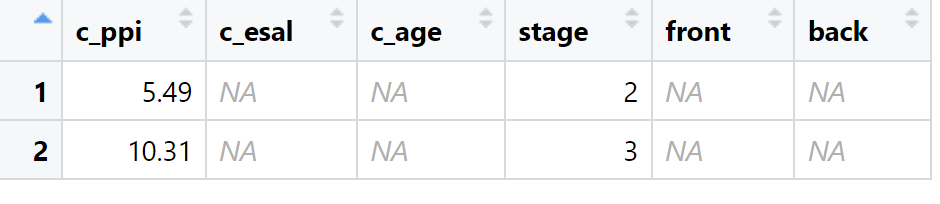
}

1. 如果data中第一行的c\_ppi≠0，则插入生命周期起点数据，起点的c\_ppi=0，c\_esal=0，stage=1，插入效果及伪代码如下所示。

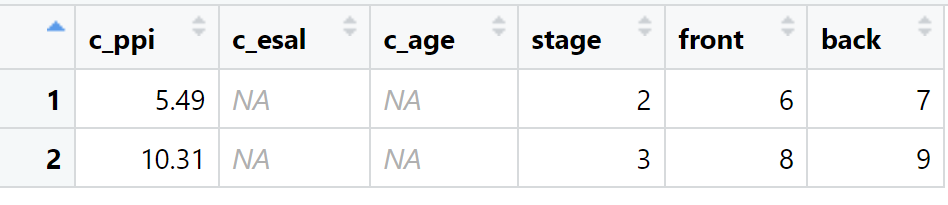


if(data[1,1]!=0){data <- rbind(c(0,0,0,1),data)}#插入生命周期起点数据

1. 根据ppi\_threshold，在data中插入其它阶段的起点数据。根据data，去除ppi\_threshold中大于c\_ppi最大值的阈值。在本样例中，要插入的其它阶段的起点数据如下所示。c\_esal、c\_age待差值计算，应该插入的位置对于当前data，其前一个位置的序号是front，后一个位置的序号是back，待计算。

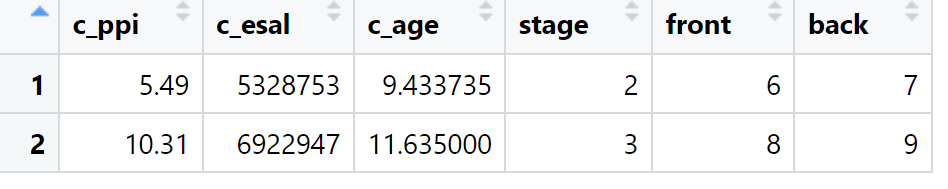


首先找到上图中两行数据分别应该插入的位置，即计算front和back。应该插入的位置是使c\_ppi按升序排列的位置，因此，上图中的c\_ppi刚好比data中哪一行的c\_ppi小，这一行的序号就是back，back-1就是front。该过程的结果及伪代码如下所示：



tmp3$back <- sapply(tmp3$c\_ppi,function(x){which.max(x-data$c\_ppi<0)});tmp3$front <- tmp3$back-1#插入到data中的位置

找到应该插入的位置后（即对应data中的前一行序号front和后一行序号back确定后），即可通过线性插值计算出缺失的c\_easl，c\_age。需要插补的c\_esal等于其前一个（front）的c\_esal加上（当前c\_ppi与前一个（front）c\_ppi的差值）/（后一个（back）c\_ppi与前一个（front）c\_ppi的差值）\*（后一个c\_esal与前一个c\_esal的差值）。同理，可计算需要差不的c\_age值。计算结果及伪代码如下所示。

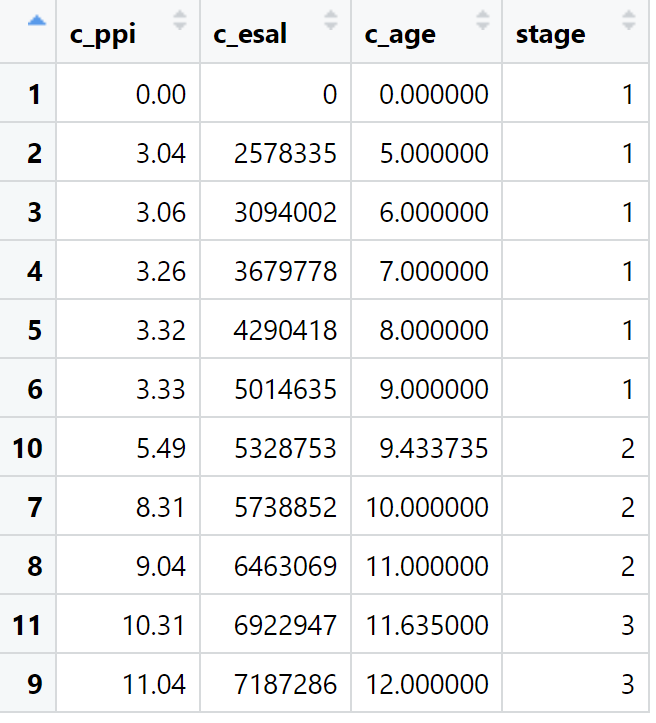
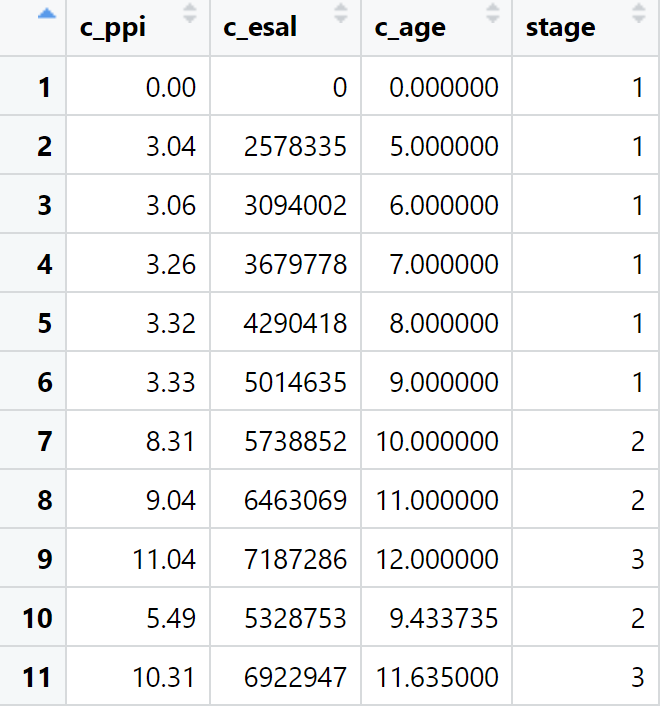


#线性插值填补c\_esal/c\_age数据

tmp3$c\_esal <- data[tmp3$front,"c\_esal"]+(tmp3$c\_ppi-data[tmp3$front,"c\_ppi"])/(data[tmp3$back,"c\_ppi"]-data[tmp3$front,"c\_ppi"])\*(data[tmp3$back,"c\_esal"]-data[tmp3$front,"c\_esal"])

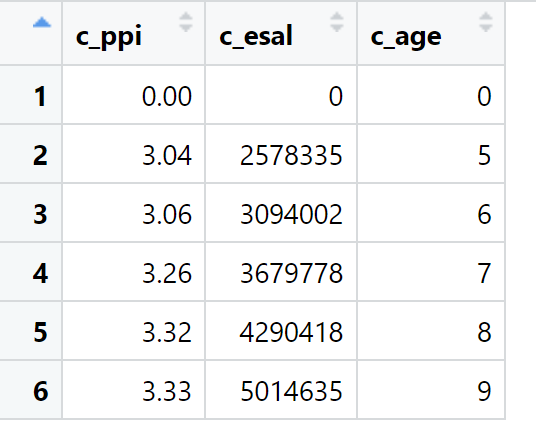
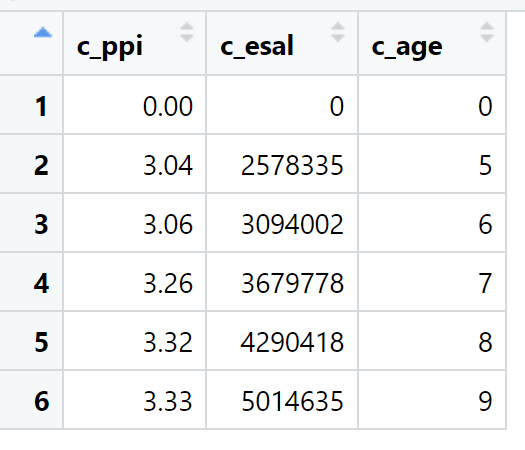
tmp3$c\_age <- data[tmp3$front,"c\_age"]+(tmp3$c\_ppi-data[tmp3$front,"c\_ppi"])/(data[tmp3$back,"c\_ppi"]-data[tmp3$front,"c\_ppi"])\*(data[tmp3$back,"c\_age"]-data[tmp3$front,"c\_age"])

然后，将tmp3中的c\_ppi，c\_esal，c\_age，stage信息附加到data中，并在data中重新按c\_ppi排序。结果及伪代码如下所示。

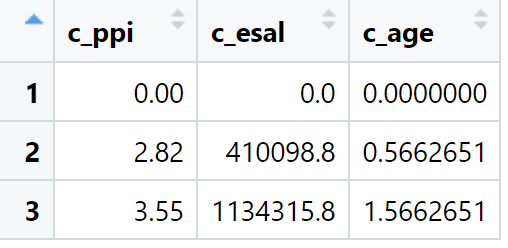
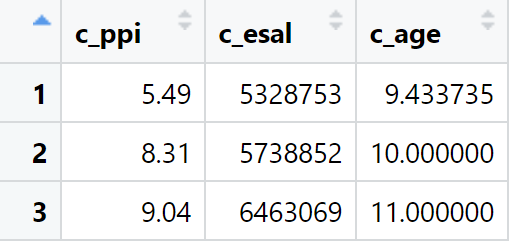


排序前后的结果

然后，提取每个阶段的数据，分别进行单阶段多元线性回归，单阶段多元线性回归的函数为single\_stage\_regression。将单阶段的数据整理到矩阵data中。本样本中第一阶段的数据如下所示。然后，对于所有行，均减去第一行的值，结果如下所示（对于第一阶段，前后无变化）。

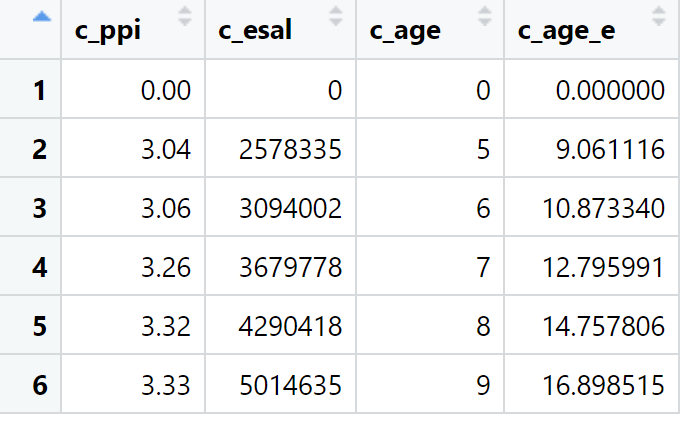


第一阶段减去第一行数据前后的结果



第二阶段减去第一行数据前后的结果

如果本阶段有2行及以上数据，则可以进行回归。首先将多元变量转单元变量。基准变量是环境荷载c\_age，c\_esal到c\_age的转换系数是k1\_k2\_ratio（本例中为1.575093e-06）。因此，多元变量转换为的单元变量（等效环境荷载，c\_age\_e），计算公式为c\_age\_e=c\_esal\*k1\_k2\_ratio+c\_age。然后，自变量是c\_age\_e，因变量是c\_ppi，即可通过线性回归得到一个衰变速率tmp，返回这个衰变速率tmp。该过程的结果及伪代码如下所示。



计算得到该阶段的衰变速率tmp=0.2395614。

#单阶段回归函数

single\_stage\_regression <- function(k1\_k2\_ratio,c\_ppi,c\_esal,c\_age){

#k1\_k2\_ratio为多少ESAL可以转换为单位Age的系数

data <- data.frame(c\_ppi,c\_esal,c\_age)

data <- data - slice(data[1,],rep(1:n(),nrow(data)))#减去起点的数值,以使用无截距项回归

if(nrow(data)<2 | diff(range(data$c\_age))<0.1){#只有1行数据的无法回归,忽略相应片段;时间窗过小的数据不进行回归,以避免不稳定

return(NULL)

}else{

data$c\_age\_e <- data$c\_esal\*k1\_k2\_ratio+data$c\_age#将交通荷载折算进环境荷载里面

lm\_model <- lm(c\_ppi~c\_age\_e-1,data = data)#对总荷载(全部折算为环境荷载)进行无截距回归

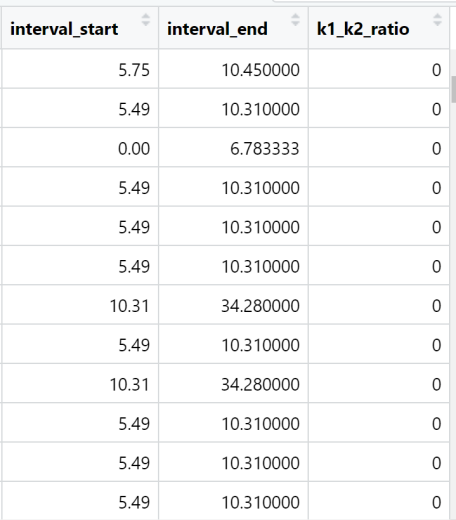
tmp <- c(lm\_model$coefficients)

return(tmp)

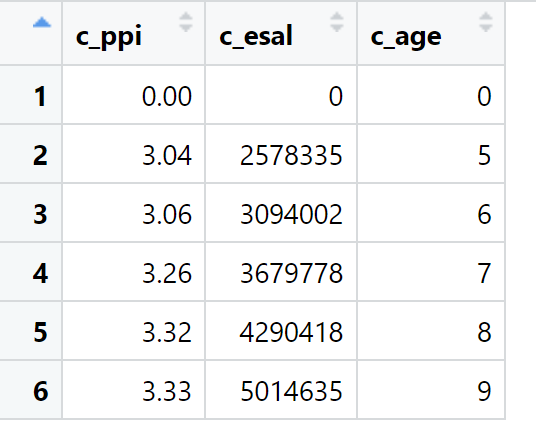
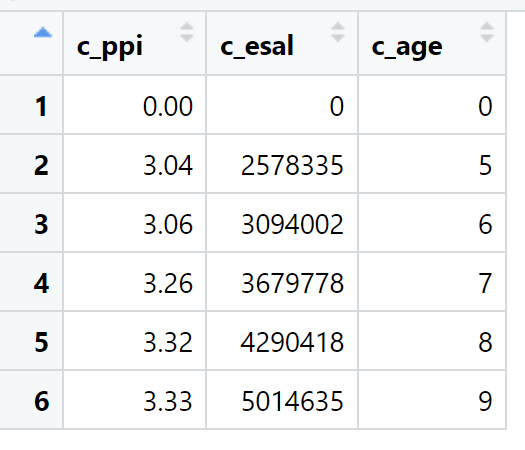
}

}

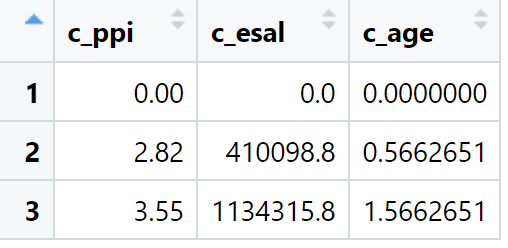
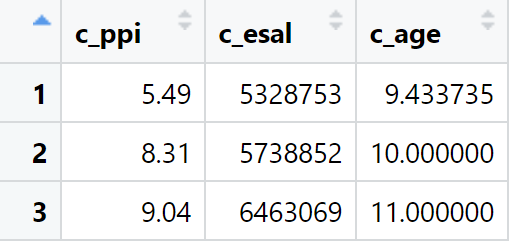
1. 【合并回归结果】：将上一步所有并行计算的结果合并到tmp4表中，如下所示。该表包含所有最小单元每个阶段的衰变速率（已记为k\_d\_age\_e），阶段阈值区间interval\_start、interval\_end，分配的并行计算参数（转换系数k1\_k2\_ratio）信息。



1. 【确定最优模型前的准备工作：对data\_deterioration数据进行预处理，使其每个阶段的数据为相对该阶段起点的增量信息】对于每个道路最小单元，根据其【道路结构和材料】匹配ppi\_threshold。将本单元的累计性能指标衰变值（c\_ppi）、累计当量轴载作用次数（c\_esal），累计路龄（c\_age）拼为一个矩阵data。然后根据c\_ppi和ppi\_threshold进行阶段划分。阶段划分函数为judge\_stage。如果data中第一行的c\_ppi≠0，则插入生命周期起点数据，起点的c\_ppi=0，c\_esal=0，stage=1。根据ppi\_threshold，在data中插入其它阶段的起点数据。根据data，去除ppi\_threshold中大于c\_ppi最大值的阈值。然后找到其他阶段数据应当插入data中的位置，即计算front和back。应该插入的位置是使c\_ppi按升序排列的位置，因此，上图中的c\_ppi刚好比data中哪一行的c\_ppi小，这一行的序号就是back，back-1就是front。
2. 找到应该插入的位置后（即对应data中的前一行序号front和后一行序号back确定后），即可通过线性插值计算出缺失的c\_easl，c\_age。需要插补的c\_esal等于其前一个（front）的c\_esal加上（当前c\_ppi与前一个（front）c\_ppi的差值）/（后一个（back）c\_ppi与前一个（front）c\_ppi的差值）\*（后一个c\_esal与前一个c\_esal的差值）。同理，可计算需要差不的c\_age值。然后，将tmp3中的c\_ppi，c\_esal，c\_age，stage信息附加到data中，并在data中重新按c\_ppi排序。上述步骤与【正式进行分阶段多元线性回归】中的相同。最后一步不同，为分阶段计算c\_ppi/c\_esal/c\_age的增量信息。结果及伪代码如下所示，



第一阶段减去第一行数据前后的结果



第二阶段减去第一行数据前后的结果

#对data\_deterioration进行预处理,使其仅包含各stage的相对其起点的c\_ppi/c\_esal/c\_age增量信息

delta\_information <- function(c\_ppi,c\_esal,c\_age){

data <- data.frame(c\_ppi,c\_esal,c\_age)#将数据整理到数据框中

if(nrow(data)>1){

if(data[1,"c\_ppi"]==data[2,"c\_ppi"]){#数据点的c\_ppi值恰好在ppi\_threshold中的情形

data <- data[-1,]

}

}

data <- data-dplyr::slice(.data = data[1,],rep(1:dplyr::n(), each = nrow(data)))

return(data[-1,])

}

initialize\_validation\_data <- function(road\_structure\_material,ppi\_thresholds,c\_ppi,c\_esal,c\_age){

tmp <- which(names(ppi\_thresholds)==road\_structure\_material[1])#确定对应道路结构和材料的ppi\_thresholds

if(!identical(tmp,integer(0))){

ppi\_threshold <- ppi\_thresholds[[tmp]]$x

data <- data.frame(c\_ppi,c\_esal,c\_age)#将数据整理到数据框中

data$stage <- judge\_stage(data$c\_ppi,ppi\_threshold = ppi\_threshold)

if(data[1,1]!=0){data <- rbind(c(0,0,0,1),data)}#插入生命周期起点数据

tmp2 <- data.frame(c\_ppi=ppi\_threshold[-1],c\_esal=NA,c\_age=NA,stage=2:length(ppi\_threshold),front=NA,back=NA)#插入其它阶段起点数据(其它阶段可以用插值处理,第一阶段不能)

tmp2 <- tmp2[tmp2$c\_ppi<max(data$c\_ppi),]#自动去除大于等于data中最大c\_ppi的阈值

if(nrow(tmp2>0)){#不止一个阶段的情形,需要插入分阶段数据

tmp2$back <- sapply(tmp2$c\_ppi,function(x){which.max(x-data$c\_ppi<0)});tmp2$front <- tmp2$back-1#插入到data中的位置

#线性插值填补c\_esal/c\_age数据

tmp2$c\_esal <- data[tmp2$front,"c\_esal"]+(tmp2$c\_ppi-data[tmp2$front,"c\_ppi"])/(data[tmp2$back,"c\_ppi"]-data[tmp2$front,"c\_ppi"])\*(data[tmp2$back,"c\_esal"]-data[tmp2$front,"c\_esal"])

tmp2$c\_age <- data[tmp2$front,"c\_age"]+(tmp2$c\_ppi-data[tmp2$front,"c\_ppi"])/(data[tmp2$back,"c\_ppi"]-data[tmp2$front,"c\_ppi"])\*(data[tmp2$back,"c\_age"]-data[tmp2$front,"c\_age"])

data <- rbind(data,tmp2[,1:4])

data <- data[order(data$c\_ppi),]#按c\_ppi升序排序

}

data <- setDT(data)[, as.list(c(val = delta\_information(c\_ppi,c\_esal,c\_age))), .(stage)]

return(data)

}

}

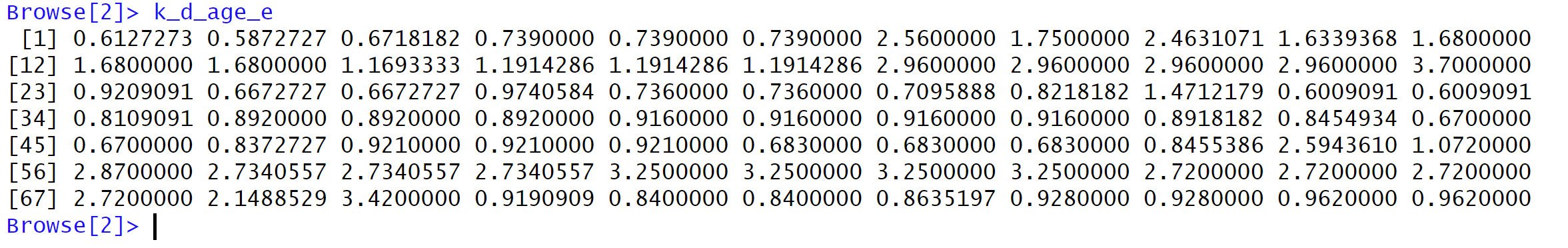
1. 【通过多重网格搜索确定每种道路结构和材料和每个阶段（stage）下最优的转换系数k1\_k2\_ratio和最优衰变速率k\_d\_age\_e】：

（1）首先根据【道路结构和材料】以及【stage】的不同，从第10.步中处理得到的data\_deterioration（此处记为validation\_data）中筛选出相应的数据集。在validation\_data中，计算等效环境荷载d\_age\_e（=d\_esal\*k1\_k2\_ratio+d\_age），计算结果如下所示（由于本样例中k1\_k2\_ratio=0，故d\_age\_e=d\_age）。



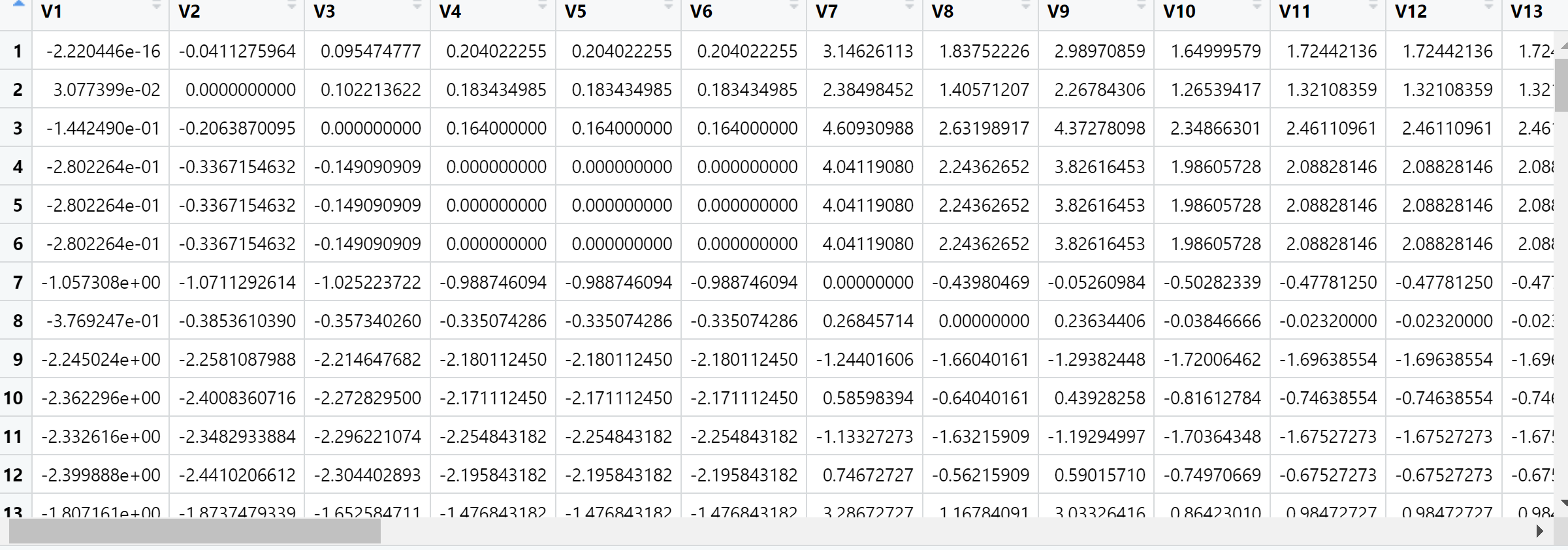
Validation\_data数据集

1. 然后筛选出tmp4（模型）中【道路结构和材料】和【stage】相同的子集，取该集合中的衰变速率向量k\_d\_age\_e。对validation\_data中的d\_age\_e与tmp4中的衰变速率向量k\_d\_age\_e取外积（预测结果），然后减去validation\_data中的d\_ppi复制n列（n=衰变速率向量k\_d\_age\_e的长度）组成的矩阵（实测结果），就得到了误差矩阵tmp。



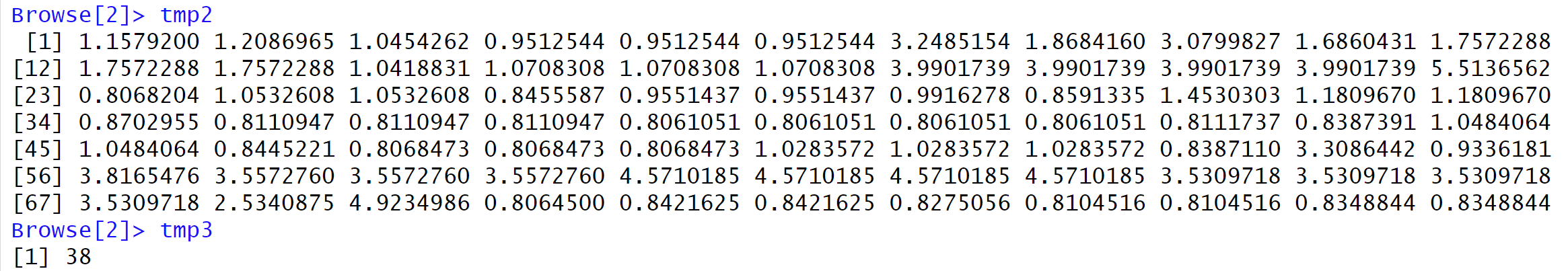


validation\_data中的d\_ppi复制n列

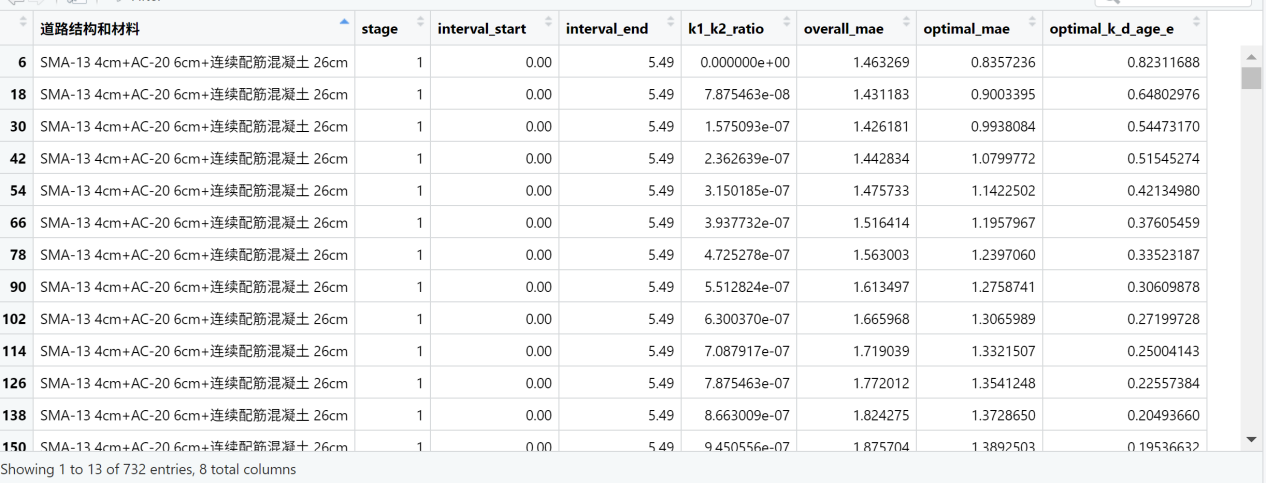


误差矩阵tmp

1. 对误差矩阵按列取平均，得到了每个衰变速率向量k\_d\_age\_e对应的误差，记为tmp2。取tmp2中误差最小的，其序号记为tmp3。



1. 返回计算结果，其中整体的平均绝对误差overall\_mae为tmp的绝对值的平均值，最优个体平均绝对误差为tmp2向量中的第tmp3个值，最优衰变速率是k\_d\_age\_e中的第tmp3个值。
2. 上述过程的最终计算结果及伪代码如下所示。



#通过网格搜索确定每种道路结构和材料/每个stage下的最优k1\_k2\_ratio和k\_d\_age\_e

optimize\_ratio\_k <- function(road\_structure\_material,stage,k\_d\_age\_e,k1\_k2\_ratio,validation\_data){#validation\_data以data\_deterioration为模板

validation\_data <- validation\_data[validation\_data$道路结构和材料==road\_structure\_material[1] & validation\_data$stage==stage[1],]#根据道路结构和材料/stage匹配验证集

validation\_data$d\_age\_e <- validation\_data$d\_esal\*k1\_k2\_ratio+validation\_data$d\_age#计算总荷载(以环境荷载表征)

tmp <- validation\_data$d\_age\_e %o% k\_d\_age\_e - matrix(data = rep(validation\_data$d\_ppi,length(k\_d\_age\_e)),nrow = nrow(validation\_data))

tmp2 <- colMeans(abs(tmp))

tmp3 <- which.min(tmp2)

return(c(overall\_mae=mean(abs(tmp)),optimal\_mae=tmp2[tmp3],optimal\_k\_d\_age\_e=k\_d\_age\_e[tmp3]))

}

tmp3 <- setDT(tmp4)[, as.list(c(val = optimize\_ratio\_k(道路结构和材料,stage,k\_d\_age\_e,k1\_k2\_ratio,data\_deterioration))), .(道路结构和材料,stage,interval\_start,interval\_end,k1\_k2\_ratio)]

names(tmp3) <- gsub(fixed=TRUE,"val.","",names(tmp3))

1. 【绘制系数（转换系数k1\_k2\_ratio）路径图】：提取最优模型。Tmp3中记录了所有道路结构和材料类型所有阶段所有转换系数k1\_k2\_ratio下的模型误差，因此可以绘制系数路径图来选取最优模型。只要给定【道路结构和材料】，就能绘制其不同【stage】下k1\_k2\_ratio和总体平均绝对误差overall\_mae的关系，示例结果和伪代码如下所示。



特定道路结构和材料下的k1\_k2\_ratio系数路径图

#分阶段绘制k1\_k2\_ratio与overall\_mae的关系曲线

par(mar=c(4.5,4.5,2,1))

tmp <- "沥青混凝土 8cm+钢筋混凝土 8cm"#选择道路结构和材料类型 stage 3 stage 1(..)

# tmp <- "沥青混凝土 6cm+钢筋混凝土 8cm"#选择道路结构和材料类型 stage 1(..) stage 2(..)

# tmp <- "沥青混凝土 8cm+钢筋混凝土 12cm"#选择道路结构和材料类型 stage 1

# tmp <- "SMA-13 4cm+AC-20 6cm+连续配筋混凝土 26cm"#选择道路结构和材料类型 stage 1

tmp2 <- tmp3[tmp3$道路结构和材料==tmp,]

plot(unlist(tmp2[tmp2$stage==1,"k1\_k2\_ratio"]),unlist(tmp2[tmp2$stage==1,"overall\_mae"]),xlab = expression(paste(k[ESAL]," : ",k[Age])),ylab = "MAE",main=paste("道路结构和材料:",tmp),ylim=c(0,15),col="DarkTurquoise",type = "p",cex.main=0.9)

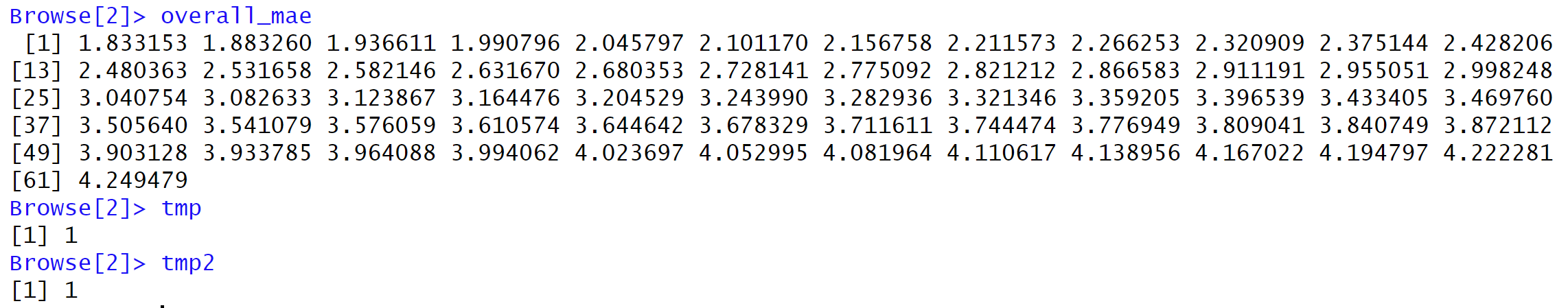
lines(unlist(tmp2[tmp2$stage==2,"k1\_k2\_ratio"]),unlist(tmp2[tmp2$stage==2,"overall\_mae"]),col="DeepPink",type = "p")

lines(unlist(tmp2[tmp2$stage==3,"k1\_k2\_ratio"]),unlist(tmp2[tmp2$stage==3,"overall\_mae"]),col="RosyBrown",type = "p")

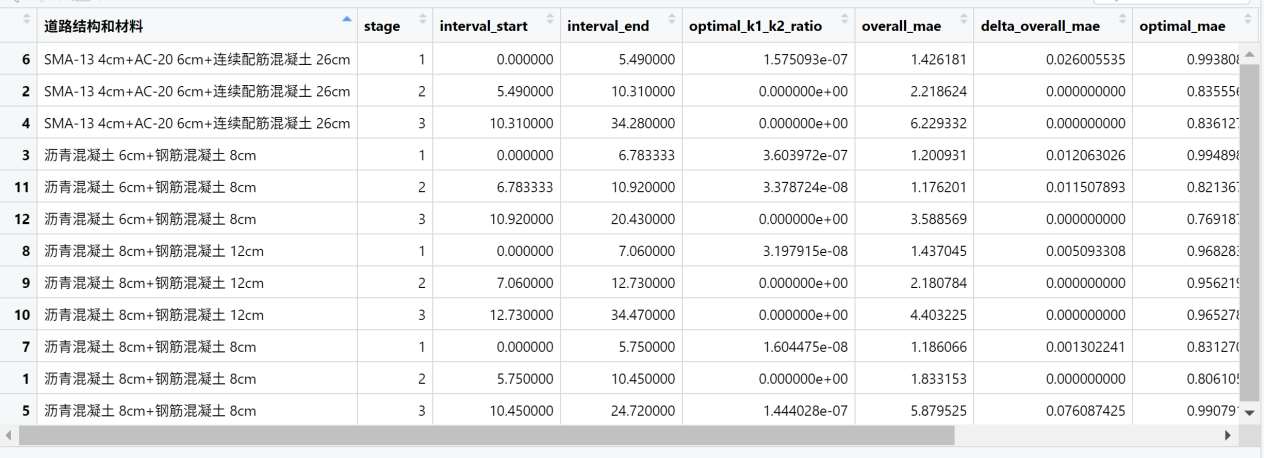
legend("topleft",c("阶段 1","阶段 2","阶段 3"),col=c("DarkTurquoise","DeepPink","RosyBrown"),text.col=c("DarkTurquoise","DeepPink","RosyBrown"),pch=1,cex=0.8)

# tmp<- tmp2[tmp2$stage==3]#测试

1. 【提取最优模型信息】：对于tmp3中记录的信息，按【道路结构和材料】和【stage】的不同分割为不同的子集，分别提取信息。比如，第一个子集中，overall\_mae如下所示。取其中最小的，序号记为tmp，可知本样例中tmp=1。与基准位置（记为tmp2）做对比，基准位置是k1\_k2\_ratio=0的位置，本样例中tmp2=1。返回最优k1\_k2\_ratio，记为optimal\_k1\_k2\_ratio，它是第tmp个k1\_k2\_ratio；返回最优整体平均绝对误差overall\_mae，它等于第tmp个overall\_mae；返回基准位置与最优位置整体mae的差异比例delta\_overall\_mae，它等于(overall\_mae[tmp2]-overall\_mae[tmp])/overall\_mae[tmp]；返回最优个体平均绝对误差optimal\_mae，它等于第tmp个optimal\_mae；返回基准位置与最优位置个体mae的差异比例delta\_overall\_mae，它等于第tmp2个optimal\_mae与第tmp个optimal\_mae的差值；返回最优衰变速率optimal\_k\_d\_age\_e，它是optimal\_k\_d\_age\_e中的第tmp个。上述过程的计算结果及伪代码如下所示。



过程结果



最终计算结果

#提取最优模型信息

optimal\_model\_information <- function(k1\_k2\_ratio,overall\_mae,optimal\_mae,optimal\_k\_d\_age\_e){

tmp <- which.min(overall\_mae)#最优模型位置

tmp2 <- which(k1\_k2\_ratio==0)#对比基准位置

return(data.frame(optimal\_k1\_k2\_ratio=k1\_k2\_ratio[tmp],overall\_mae=overall\_mae[tmp],delta\_overall\_mae=(overall\_mae[tmp2]-overall\_mae[tmp])/overall\_mae[tmp],optimal\_mae=optimal\_mae[tmp],delta\_optimal\_mae=optimal\_mae[tmp2]-optimal\_mae[tmp],optimal\_k\_d\_age\_e=optimal\_k\_d\_age\_e[tmp]))

}

prediction\_model <- setDT(tmp3)[, as.list(c(val = optimal\_model\_information(k1\_k2\_ratio,overall\_mae,optimal\_mae,optimal\_k\_d\_age\_e))), .(道路结构和材料,stage,interval\_start,interval\_end)]

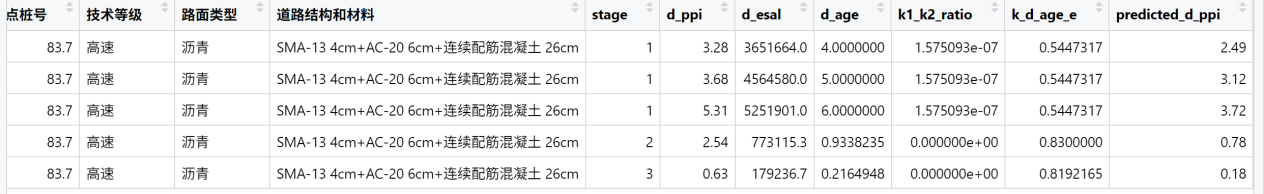
names(prediction\_model) <- gsub(fixed=TRUE,"val.","",names(prediction\_model))

1. 【绘图准备】：
2. 在data\_deterioration中，根据路线代码、方向、车道编号、起点桩号选出特定的道路最小单元的数据，记录在tmp中。样例数据及伪代码如下所示。



tmp <- filter(data\_deterioration,路线代码=="S20" & 方向=="下行" & 车道编号==2 & 起点桩号==83.6)

1. 对tmp数据附上模型prediction\_model中的最优转换系数optimal\_k1\_k2\_ratio和最优衰变速率optimal\_k\_d\_age\_e信息（根据道路结构和材料、stage字段在prediction\_model中找到匹配的optimal\_k1\_k2\_ratio和optimal\_k\_d\_age\_e）。再计算d\_ppi，它等于d\_esal与k1\_k2\_ratio的乘积与d\_age的和，再与k\_d\_age\_e的乘积。最后根据【道路结构和材料】在ppi\_thresholds中查表得到本样例的ppi阈值结点向量ppi\_threshold。计算结果及伪代码如下所示。



##附上模型信息

tmp2<- foreach(i=1:nrow(tmp),.combine = "rbind") %do%

{

tmp3 <- tmp[i,]

tmp4 <- which(prediction\_model$道路结构和材料==tmp3$道路结构和材料 & prediction\_model$stage==tmp3$stage)

tmp3$k1\_k2\_ratio <- prediction\_model[tmp4,"optimal\_k1\_k2\_ratio"]

tmp3$k\_d\_age\_e <- prediction\_model[tmp4,"optimal\_k\_d\_age\_e"]

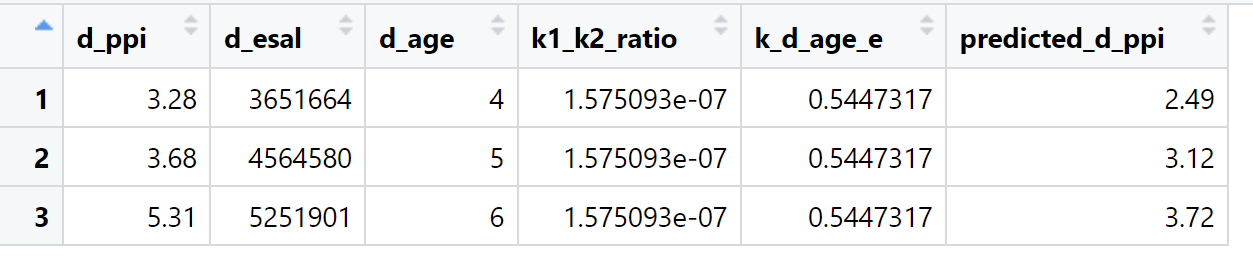
return(tmp3)

}

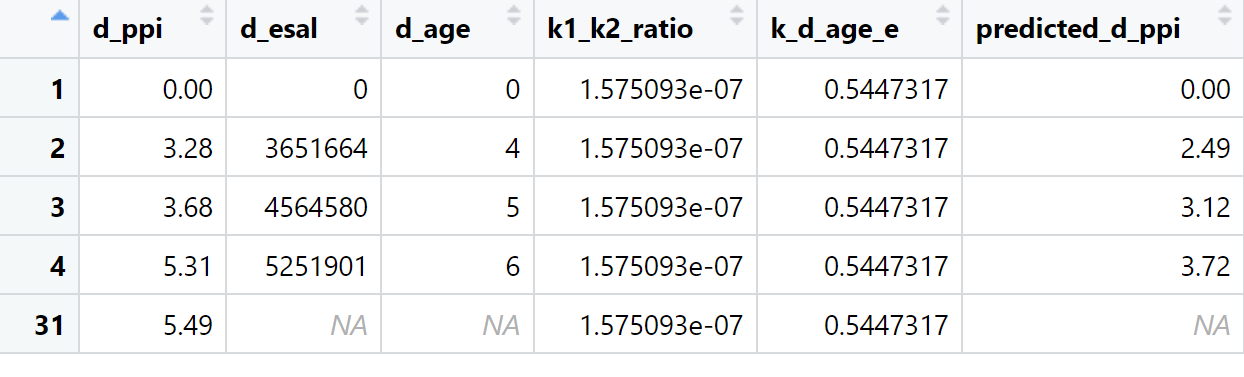
tmp2$predicted\_d\_ppi <- round((tmp2$k1\_k2\_ratio\*tmp2$d\_esal+tmp2$d\_age)\*tmp2$k\_d\_age\_e,2)

ppi\_threshold <- ppi\_thresholds[[which(names(ppi\_thresholds)==tmp$道路结构和材料[1])]]$x

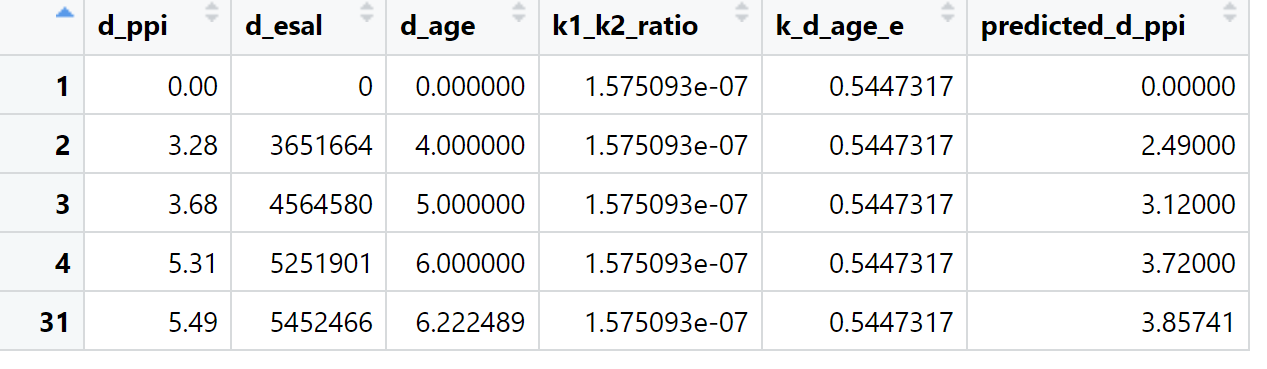
1. 对上述计算结果转换为累计值，以方便绘图。对于每个道路最小单元的每个阶段，分别累计。比如，对于第一阶段，它的"d\_ppi"、"d\_esal"、"d\_age"、"k1\_k2\_ratio"、"k\_d\_age\_e" "predicted\_d\_ppi"如下所示。



插入该阶段的起终点数据，如下所示。其中起点的d\_ppi"、"d\_esal"、"d\_age"、"predicted\_d\_ppi"均为0，终点的"d\_esal"、"d\_age"、"predicted\_d\_ppi"待插值计算。终点的"d\_esal"等于终点的上一个节点（本案例中终点在第5行，上一个节点是第4行）的"d\_esal"加上d\_ppi沿d\_esal方向的变化速率（=上图中d\_esal/d\_ppi的平均值）\*（终点d\_ppi与上一个节点d\_ppi的差值）。同理，可以计算得到终点处的d\_age。而终点处的predicted\_d\_ppi等于k1\_k2\_ratio与d\_esal的乘积与d\_age的和，再乘以k\_d\_age\_e。

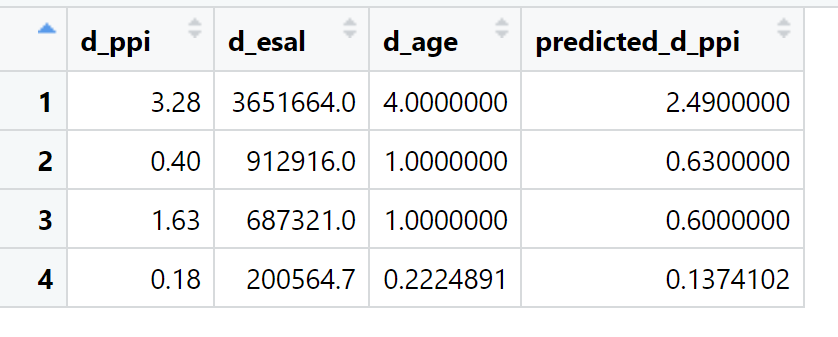


插补完成前

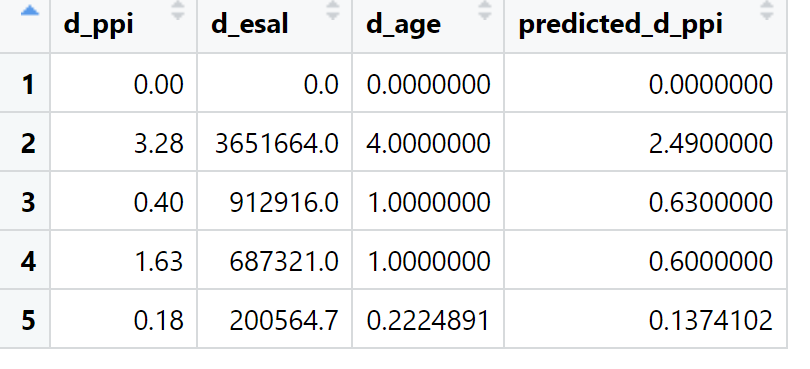


插补完成后

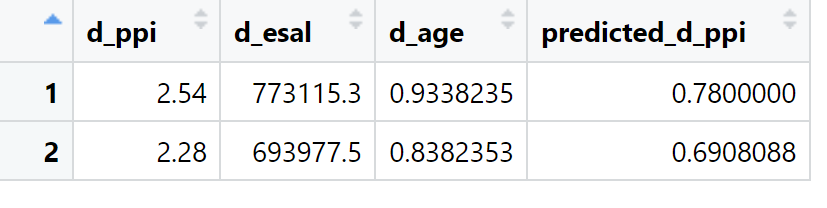
然后，对上图所示数据（data）中的"d\_ppi"、"d\_esal"、"d\_age"、"predicted\_d\_ppi"均做差值处理，重新记录为data。如果stage为1，第一行插入一行



差分处理后的数据（第一阶段，未插入第一行之前）



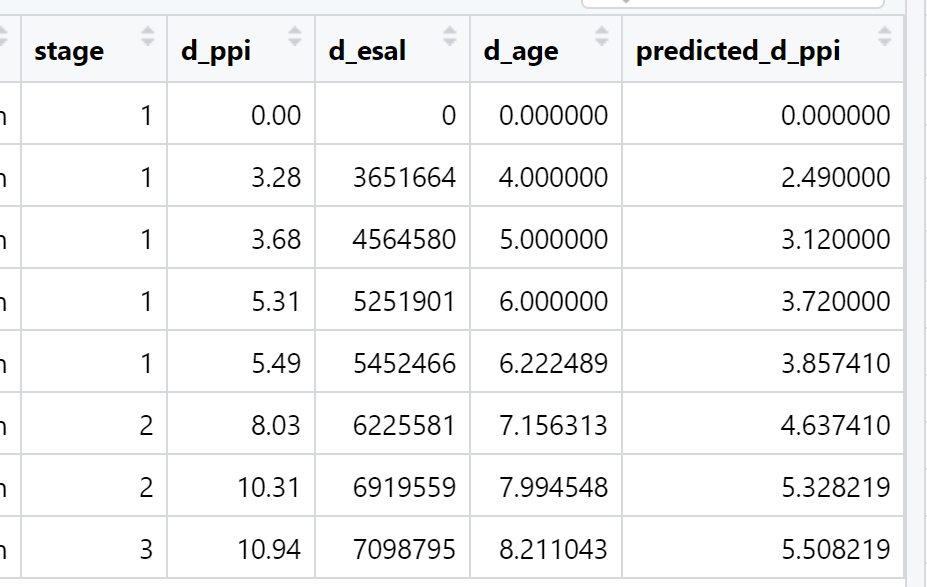
第一阶段需要在第一行插入0 0 0 0 向量



差分处理后的数据（第二阶段）

1. 最后再对data中的"d\_ppi"、"d\_esal"、"d\_age"、"predicted\_d\_ppi"做累加。（1）-（4）过程的总计算结果和伪代码如下所示。





##转换为累计值,以便绘图

accumulation\_information <- function(stage,d\_ppi,d\_esal,d\_age,k1\_k2\_ratio,k\_d\_age\_e,predicted\_d\_ppi,ppi\_threshold){

data <- data.frame(d\_ppi,d\_esal,d\_age,k1\_k2\_ratio,k\_d\_age\_e,predicted\_d\_ppi)

#初始化本stage起点和终点的数值

data <- rbind(data[1,],data,data[nrow(data),])

data[1,c("d\_ppi","d\_esal","d\_age","predicted\_d\_ppi")] <- 0

data[nrow(data),c("d\_ppi","d\_esal","d\_age","predicted\_d\_ppi")] <- c(ppi\_threshold[stage+1]-ppi\_threshold[stage],NA,NA,NA)#注意是第stage+1个与第stage个ppi\_threshold的差值

#按平均增加量填充ppi\_threshold阈值对应的ESAL/Age

data[nrow(data),"d\_esal"] <- mean(d\_esal/d\_ppi)\*(data[nrow(data),"d\_ppi"]-data[nrow(data)-1,"d\_ppi"])+data[nrow(data)-1,"d\_esal"]

data[nrow(data),"d\_age"] <- mean(d\_age/d\_ppi)\*(data[nrow(data),"d\_ppi"]-data[nrow(data)-1,"d\_ppi"])+data[nrow(data)-1,"d\_age"]

#按照预测模型填充predicted\_d\_ppi

data[nrow(data),"predicted\_d\_ppi"] <- (data[nrow(data),"k1\_k2\_ratio"]\*data[nrow(data),"d\_esal"]+data[nrow(data),"d\_age"])\*data[nrow(data),"k\_d\_age\_e"]

data <- data.frame(d\_ppi=diff(data$d\_ppi),d\_esal=diff(data$d\_esal),d\_age=diff(data$d\_age),predicted\_d\_ppi=diff(data$predicted\_d\_ppi))

if(stage==1){data <- rbind(c(0,0,0,0),data)}

return(data)

}

tmp <- setDT(tmp2)[, as.list(c(val = accumulation\_information(stage,d\_ppi,d\_esal,d\_age,k1\_k2\_ratio,k\_d\_age\_e,predicted\_d\_ppi,ppi\_threshold))), .(路线代码,方向,车道编号,起点桩号,止点桩号,技术等级,路面类型,道路结构和材料,stage)]

tmp <- tmp[-nrow(tmp),]#去掉最后一行

names(tmp) <- gsub(fixed=TRUE,"val.","",names(tmp))

tmp[,c("d\_ppi","d\_esal","d\_age","predicted\_d\_ppi")] <- lapply(tmp[,c("d\_ppi","d\_esal","d\_age","predicted\_d\_ppi")],FUN = cumsum)

1. 【开始绘图】：

功能三

以14.中data的"d\_esal"、"d\_age"为自变量，"d\_ppi"、"predicted\_d\_ppi"分别为因变量绘制三维折线（散点）图。其中，"predicted\_d\_ppi"为折线，为预测模型给出的预测结果。"d\_ppi"为散点，为实际的ppi衰变。配置好图例，以green、orange、red分别对第1、2、3阶段的数据进行染色。图的标题显示本最小单元的基本信息，包括路线代码，方向，车道编号，起止桩号。



#开始绘图

locate\_stage\_label <- function(stage){

# return(quantile(seq(1,length(unique(stage)),length.out=length(stage)),probs = ((table(tmp$stage)[order(unique(tmp$stage))]+1)/2+c(0,cumsum(table(tmp$stage))[order(unique(tmp$stage))])[1:length(unique(tmp$stage))])/length(tmp$stage)))#stage必须提前按升序排列

return(seq(from=(length(unique(stage))-1)/length(unique(stage))/2+1,by=(length(unique(stage))-1)/length(unique(stage)),length.out=length(unique(stage))))

# return(seq(from=1,to=2,length.out=length(unique(stage))))

}

switch\_stage\_col <- function(stage){

if(stage==1){

"green"

}else if(stage==2){

"orange"

}else{

"red"

}

}

stage\_order <- function(stage,unique\_stage){

return(which(unique\_stage==stage))

}

switch\_colvar <- function(stage){#折线图每个线段的颜色值与两个端点有关

stage <- sapply(stage,stage\_order,unique\_stage=unique(stage))#阶段标准化为从1开始,相邻阶段差为1

if(length(stage)<=2){#长度过短

return(stage)

}else{

tmp <- which(diff(stage)>0)#stage变化的地方

if(identical(tmp,integer(0))){

return(stage)

}else if(length(tmp)==1){

stage[tmp] <- stage[tmp+1]

}else{

stage[tmp[-length(tmp)]] <- ifelse(table(stage)[stage[tmp[-length(tmp)]]]>1,stage[tmp[-length(tmp)]+1],stage[tmp[-length(tmp)]])

stage[tmp[-length(tmp)]+1] <- ifelse(table(stage)[stage[tmp[-length(tmp)]+1]]>1,stage[tmp[length(tmp)]],stage[tmp[length(tmp)]+1])

}

}

return(stage)

}

par(mar=c(1,0,2,2))

plot3D::lines3D(x=tmp$d\_esal,y=tmp$d\_age,z=tmp$d\_ppi,lwd=2,axes = FALSE,colvar = switch\_colvar(tmp$stage),col=sapply(unique(tmp$stage),switch\_stage\_col),colkey = list(plot=TRUE,cex.clab=0.8,length=0.7,at=locate\_stage\_label(tmp$stage),labels=as.character(unique(tmp$stage)),cex.axis=0.7),

clab = "Stage",main=paste(tmp$路线代码[1],tmp$方向[1],paste(tmp$车道编号[1],"#",sep = ""),paste("K",tmp$起点桩号[1],"-",tmp$止点桩号[1],sep = ""),"PPI自然衰变预测及验证"),cex.main=0.9,scale=TRUE,phi = 30)

plot3D::scatter3D(x=tmp$d\_esal,y=tmp$d\_age,z=tmp$predicted\_d\_ppi,scale=TRUE,colvar=tmp$stage,col=sapply(unique(tmp$stage), switch\_stage\_col),phi = 30,colkey = FALSE, type="p",add=TRUE)

#标签

x.tick.locations <- round(seq(from=range(tmp$d\_esal)[1],to=range(tmp$d\_esal)[2],length.out=7),0)

text3D(x.tick.locations\*0.8, rep(-max(tmp$d\_age)\*0.2, length(x.tick.locations)), rep(0, length(x.tick.locations)), labels=round(seq(range(tmp$d\_esal)[1],range(tmp$d\_esal)[2],length.out=7),0),cex=0.6,add = TRUE)

text3D(mean(range(tmp$d\_esal)), -max(tmp$d\_age)\*0.4, 0, labels="ESAL",cex=0.7,add = TRUE)

y.tick.locations <- round(seq(from=range(tmp$d\_age)[1],to=range(tmp$d\_age)[2],length.out=6),1)

text3D(rep(max(tmp$d\_esal)\*1.05, length(y.tick.locations)),y.tick.locations,rep(0, length(y.tick.locations)), labels=round(seq(from=range(tmp$d\_age)[1],to=range(tmp$d\_age)[2],length.out=6),0),cex=0.6,add = TRUE)

text3D(max(tmp$d\_esal)\*1.2,mean(tmp$d\_age)\*0.7, 0, labels="Road Age",cex=0.7,add = TRUE)

z.tick.locations <- round(seq(from=range(tmp$d\_ppi)[1],to=range(tmp$d\_ppi)[2],length.out=7),0)

text3D(rep(-max(tmp$d\_esal)\*0.15, length(z.tick.locations)),rep(0, length(z.tick.locations)),z.tick.locations, labels=z.tick.locations,cex=0.6,add = TRUE)

text3D(-max(tmp$d\_esal)\*0.15, 0, max(z.tick.locations)\*1.1, labels=expression(paste(Delta," PPI")),cex=0.7,add = TRUE)

# write\_xlsx(prediction\_model,path = 'C:/Users/Archon/Desktop/prediction\_model.xlsx')

print(Sys.time()-t1)