Homework3

23 October 2016

### 1 阅读部分

* 阅读提供的材料,附加读书笔记

##### 1 "03-Linear Models.pdf"

1. 建模过程
2. 线性回归

* summary(mod)
* predict(mod)
* resid(mod)
* plot(mod)

1. 模型推断

* 根据p值看变量是否显著

1. 分类变量

* 类别变量回归得到的系数及截距会随变量中因子排序不同而有所变化，但其本质是一样的，只是根据不同的基线不同。
* 分类变量的回归和方差分析的结果是一致的。

1. 多变量回归

* lm(~.,data = ...)回归所有变量。
* 两个变量之间的回归关系会因为加入第三个变量而改变

1. 交叉因子

* 如还sex和height两个变量的交叉影响作用，用sex+height+height:sex
* sex\*height 就等价于sex+height+height:sex

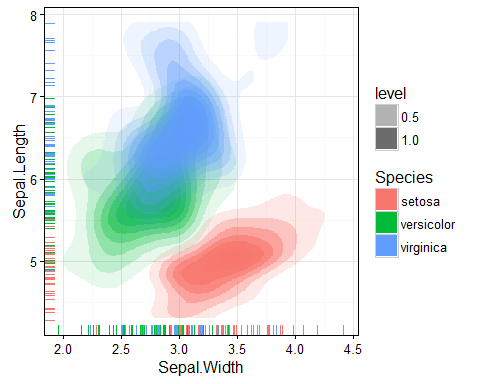
— 总结： 单变量、多变量、交叉变量三种； 解释模型：系数（coef）、残差、置信区间、P值、对比模型anova（mod1，mod2）

##### 2 "ggplot2"

* 统计图形：从数据到几何对象（geom，包括点，线，条形等）的图形属性（aes，包括颜色，形状，大小等）的一个映射。另外，图形中还可能包含数据的统计变换（stats），最后绘制在某个特定的坐标系（coord），而分面（facet）则可以根据数据生成不同子集的图形。
* 数据（data）
* 映射（mapping）
* 几何对象（geom）： 点、线、多边形等
* 统计变换（stats）： 对数据进行某种汇总
* 标度（scale）： 将数据的取值映射到图形空间，用颜色、大小、形状来表示不同的取值。展现标度的常见做法是绘制图例和坐标轴。
* 坐标系（coord）： 描述数据如何映射到图形坐在平面，笛卡尔坐标系，极坐标和地图投影等。
* 分面（facet）： 将数据分解为各个子集，对自己如何作图并联合进项展示。
* “一个例子”： ggplot(data = , aes(x = , y = )) + geom\_XXX(...) + ... + stat\_XXX(...) + ... + annotate(...) + ... + scale\_XXX(...) + coord\_XXX(...) + guides(...) + theme(...)

### 2 重现以下图形

library(ggplot2)  
ggplot(data = iris,aes(x=Sepal.Width,y=Sepal.Length,fill = Species)) +  
 stat\_density2d(aes(alpha = ..level..),geom = "polygon") +  
 geom\_rug(position = 'jitter',aes(color = Species)) + #边际地毯  
 guides(color=F) +  
 theme\_bw() + #背景白色  
 theme(panel.border = element\_rect(colour = "black", fill = NA)) #边框黑色



### 3 Kaggle: House Prediction

* 网址: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques>

#### 要求

1. 注册并参加比赛
2. 参考并运行示例代码(可选)
3. 自己改良或者自己完成预测代码

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.3.2

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

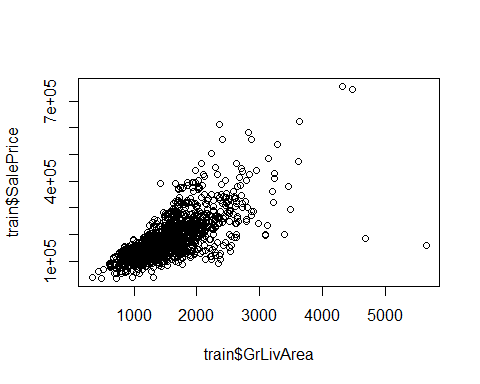
##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(Metrics)

## Warning: package 'Metrics' was built under R version 3.3.2

setwd("F:/研究生/课程/R/上课/第三次课/HousePrice")  
train <- read.csv("train.csv") #这里读为因子型，方便后边把变量转化为integer  
test <- read.csv("test.csv")  
  
sort(sapply(train, function(x) { sum(is.na(x)) }), decreasing=TRUE)  
out <- c("PoolQC","MiscFeature","Alley","Fence","FireplaceQu","LotFrontage")  
train <- train[setdiff(names(train),out)] #将缺失值多的几个变量直接删掉  
  
train[is.na(train$Electrical),which(names(train) == "Electrical")] <- "SBrkr" #缺失的那1个值改为出现最多的“SBrkr”  
train[is.na(train$MasVnrType),which(names(train) == "MasVnrType")] <- "None" #缺失的8个值改为出现最多的“None”  
train[is.na(train$MasVnrArea),which(names(train) == "MasVnrArea")] <- "0" #缺失的8个值改为出现最多的“0”  
train[is.na(train$BsmtQual),which(names(train) == "BsmtQual")] <- "TA"   
train[is.na(train$BsmtCond),which(names(train) == "BsmtCond")] <- "TA"  
train[is.na(train$BsmtFinType1),which(names(train) == "BsmtFinType1")] <- "Unf"  
train[is.na(train$BsmtFinType2),which(names(train) == "BsmtFinType2")] <- "Unf"  
train[is.na(train$BsmtExposure),which(names(train) == "BsmtExposure")] <- "No"  
train[is.na(train$GarageType),which(names(train) == "GarageType")] <- "Attchd"  
train[is.na(train$GarageYrBlt),which(names(train) == "GarageYrBlt")] <- "2005"  
train[is.na(train$GarageFinish),which(names(train) == "GarageFinish")] <- "Unf"  
train[is.na(train$GarageQual),which(names(train) == "GarageQual")] <- "TA"  
train[is.na(train$GarageCond),which(names(train) == "GarageCond")] <- "TA"  
sum(is.na(train)) #查看train的数据没有缺失值  
  
#删除一些异常值  
plot(train$GrLivArea,train$SalePrice)



train <- train[train$GrLivArea < 4000,]  
  
#将因子型变量转化为数值型  
for(i in 1:75){  
 if(is.factor(train[,i])){  
 train[,i] <- as.integer(train[,i])   
 }  
}  
  
  
#将训练集拆分成训练集和预测集  
set.seed(123)  
index = sample(1:1456,1000)  
train\_train = train[index,]  
train\_test = train[-index,]  
  
#线性回归模型  
lm\_model <- lm(SalePrice~ LotArea + OverallQual + OverallCond + YearBuilt +   
 MasVnrType + ExterQual + BsmtQual + BsmtFinSF1 +  
 BsmtFinSF2 + BsmtUnfSF + X1stFlrSF +X2ndFlrSF + BedroomAbvGr +  
 KitchenAbvGr + Functional + ScreenPorch + SaleCondition,data=train\_train)  
   
summary(lm\_model)  
pred1 <- predict(lm\_model,newdata = train\_test)  
rmse(log(train\_test$SalePrice),log(pred1))

## Warning in log(pred1): 产生了NaNs

#随机森林模型  
rf\_model <- randomForest(SalePrice~.,data = train\_train)  
pred2 <- predict(rf\_model,newdata = train\_test)  
rmse(log(train\_test$SalePrice),log(pred2))  
  
#比较了lm和randomForest，randomForest的rmse比较小，选用随机森林模型  
  
  
#处理test集中的缺失值  
sort(sapply(test, function(x) { sum(is.na(x)) }), decreasing=TRUE)  
  
out <- c("PoolQC","MiscFeature","Alley","Fence","FireplaceQu","LotFrontage")  
test <- test[setdiff(names(test),out)]#删去缺失值较大的几个变量  
  
test[is.na(test$SaleType),which(names(test) == "SaleType")] <- "WD"  
test[is.na(test$GarageArea),which(names(test) == "GarageArea")] <- "0"  
test[is.na(test$GarageCars),which(names(test) == "GarageCars")] <- "2"  
test[is.na(test$KitchenQual),which(names(test) == "KitchenQual")] <- "TA"  
test[is.na(test$TotalBsmtSF),which(names(test) == "TotalBsmtSF")] <- "0"  
test[is.na(test$BsmtUnfSF),which(names(test) == "BsmtUnfSF")] <- "0"  
test[is.na(test$BsmtFinSF2),which(names(test) == "BsmtFinSF2")] <- "0"  
test[is.na(test$BsmtFinSF1),which(names(test) == "BsmtFinSF1")] <- "0"  
test[is.na(test$Exterior2nd),which(names(test) == "Exterior2nd")] <- "VinylSd"  
test[is.na(test$Exterior1st),which(names(test) == "Exterior1st")] <- "VinylSd"  
test[is.na(test$Functional),which(names(test) == "Functional")] <- "Typ"  
test[is.na(test$BsmtHalfBath),which(names(test) == "BsmtHalfBath")] <- "0"  
test[is.na(test$BsmtFullBath),which(names(test) == "BsmtFullBath")] <- "0"  
test[is.na(test$Utilities),which(names(test) == "Utilities")] <- "AllPub"  
test[is.na(test$MSZoning),which(names(test) == "MSZoning")] <- "RL"  
test[is.na(test$MasVnrArea),which(names(test) == "MasVnrArea")] <- "0"  
test[is.na(test$MasVnrType),which(names(test) == "MasVnrType")] <- "None"  
test[is.na(test$BsmtFinType2),which(names(test) == "BsmtFinType2")] <- "Unf"  
test[is.na(test$BsmtFinType1),which(names(test) == "BsmtFinType1")] <- "GLQ"  
test[is.na(test$BsmtExposure),which(names(test) == "BsmtExposure")] <- "No"  
test[is.na(test$BsmtQual),which(names(test) == "BsmtQual")] <- "TA"  
test[is.na(test$BsmtCond),which(names(test) == "BsmtCond")] <- "TA"  
test[is.na(test$GarageType),which(names(test) == "GarageType")] <- "Attchd"  
test[is.na(test$GarageCond),which(names(test) == "GarageCond")] <- "TA"  
test[is.na(test$GarageQual),which(names(test) == "GarageQual")] <- "TA"  
test[is.na(test$GarageFinish),which(names(test) == "GarageFinish")] <- "Unf"  
test[is.na(test$GarageYrBlt),which(names(test) == "GarageYrBlt")] <- "2007"  
sum(is.na(test)) #检查是否还有缺失值  
  
for(i in 1:74){  
 if(is.factor(test[,i]) | is.character(test[,i])){  
 test[,i]<-as.integer(test[,i])  
 }  
}  
  
prediction <- predict(rf\_model,newdata = test)  
output <- cbind(test,prediction)  
  
submit <- data.frame(Id = output$Id,SalePrice = output$prediction)  
write.csv(submit,file = "output.csv",row.names = T)

比赛排名：