Linear\_Modeling.R

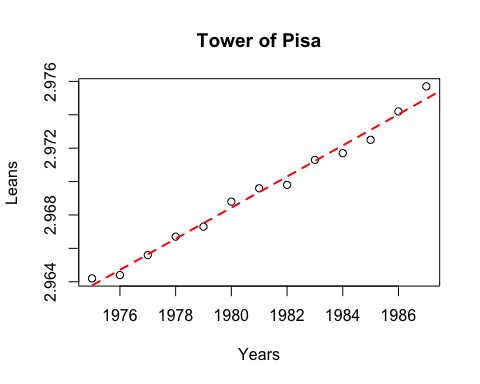
riserate

Tue May 9 20:21:28 2017

#More Linear Modeling  
  
#Use the following data set on the Tower of Pisa, study linear model with R.  
  
#a. input the year data into a variable “years”; use the R function;  
  
xyears <- c(1975,1976,1977,1978,1979,1980,1981,1982,1983,1984,1985,1986,1987)  
  
#b. input the lean data into a variable “leans”; use the R function scan() or c();  
yleans <- c(2.9642,2.9644,2.9656,2.9667,2.9673,2.9688,2.9696,2.9698,2.9713,2.9717,2.9725,2.9742,2.9757)  
  
#c. make the scatter plot of years and leans; use the R function plot();  
plot(xyears, yleans, xlab="Years", ylab="Leans", main="Tower of Pisa")  
  
#d. guess a linear model that can fit this data, i.e., guess the slope and intercept, and then plot the results with the R function abline(). You can use your function as  
# abline(guessed\_intercept, guessed\_slope) or use help(abline) to find more info about the use of the model.  
  
#The correlation is pretty close to 1.0 so we can say that the relationship is strong.  
cor(xyears,yleans)

## [1] 0.9939717

# We can find the intercept and slope using the command lm  
fit <- lm(yleans ~ xyears)  
  
#This model is a positive linear model with a slope of 0.00093 and intercept of 1.1233  
plot(xyears, yleans, xlab="Years", ylab="Leans", main="Tower of Pisa")  
abline(fit, lwd=2, lty=2, col="red")



#e.figure out R commands to compute the mean squared error of the predictions  
#(In this formula, b = the y intercept, and m = slope):  
summary(fit)

##   
## Call:  
## lm(formula = yleans ~ xyears)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.967e-04 -3.099e-04 6.703e-05 2.308e-04 7.396e-04   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.123e+00 6.139e-02 18.30 1.39e-09 \*\*\*  
## xyears 9.319e-04 3.099e-05 30.07 6.50e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0004181 on 11 degrees of freedom  
## Multiple R-squared: 0.988, Adjusted R-squared: 0.9869   
## F-statistic: 904.1 on 1 and 11 DF, p-value: 6.503e-12

attributes(fit)

## $names  
## [1] "coefficients" "residuals" "effects" "rank"   
## [5] "fitted.values" "assign" "qr" "df.residual"   
## [9] "xlevels" "call" "terms" "model"   
##   
## $class  
## [1] "lm"

library(Metrics)  
  
#mean squared error  
rmse(xyears, yleans)

## [1] 1978.034

#f. compute the mean squared error of the prediction of your linear model with your neighbor’s, see who can come up with the closest fit.  
rmse(xyears, yleans)

## [1] 1978.034