Doug Woodward

CS613 HW 4

|  |  |
| --- | --- |
| # | Answer |
| 10 | Residuals:   Min 1Q Median 3Q Max  -6.9206 -1.6220 -0.0564 1.5786 7.0581    Coefficients:   Estimate Std. Error t value Pr(>|t|)  (Intercept) 13.043469 0.651012 20.036 < 2e-16 \*\*\*  Price -0.054459 0.005242 -10.389 < 2e-16 \*\*\*  UrbanYes -0.021916 0.271650 -0.081 0.936  USYes 1.200573 0.259042 4.635 4.86e-06 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1    Residual standard error: 2.472 on 396 degrees of freedom  Multiple R-squared: 0.2393, Adjusted R-squared: 0.2335  F-statistic: 41.52 on 3 and 396 DF, p-value: < 2.2e-16  b ) The Price has an inverse relationship with sales, but a small one with a magnitude under 0.06. It also appears that Urban being a Yes correlates to negative sales, however, there is a high std error for this term as well. Being made in the US also correlates to positive sales, with about 1 per unit more, however, this also has a high p-value, and is therefore unlikely to be the cause of sales directly.  c) S = 13.04 + -0.054Price + -0.02UrbanYes + 1.2USYes + error  d) For price and US as they have small p values.  e)Residuals:   Min 1Q Median 3Q Max  -6.9269 -1.6286 -0.0574 1.5766 7.0515    Coefficients:   Estimate Std. Error t value Pr(>|t|)  (Intercept) 13.03079 0.63098 20.652 < 2e-16 \*\*\*  Price -0.05448 0.00523 -10.416 < 2e-16 \*\*\*  USYes 1.19964 0.25846 4.641 4.71e-06 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1    Residual standard error: 2.469 on 397 degrees of freedom  Multiple R-squared: 0.2393, Adjusted R-squared: 0.2354  F-statistic: 62.43 on 2 and 397 DF, p-value: < 2.2e-16  f) Removing Urban from the model fits better, with R^2 being slightly larger.  g) 2.5 % 97.5 %  (Intercept) 11.79032020 14.27126531  Price -0.06475984 -0.04419543  USYes 0.69151957 1.70776632  h) The plot indicates a linear relationship between variables and predicted sales, it does have some outliers with high leverage, one in particular. |
| 11 | 1. Coefficients:   Estimate Std. Error t value Pr(>|t|)  x 1.9939 0.1065 18.73 <2e-16 \*\*\*   This coeefeicent maps to our dummy data as expected, with each change in x corresponding to a change of twice that magnitude in y. The p value is low, so we are sure this is the cause, and there is an error term that is expected.   1. Coefficients:   Estimate Std. Error t value Pr(>|t|)  y 0.39111 0.02089 18.73 <2e-16 \*\*\*   Here, we see the same expected result, with changes in y mapping to approx half, however out t value is somewhat large again, indicating that we have a noticeable deviation from the expected mean - and given the deviation we see in the coefficient estimate, this is unsurprising.   1. They are telling us the same thing, just in different terms with the noteworthy point being that the t value is large and thus skewing our estimates. 2. n = length(x) t = sqrt(n-1)\*(x%\*%y)/sqrt(sum(x^2)\*sum(y^2)-(x%\*%y)^2) as.numeric(t) 18.72593 is the same result obtained before 3. Replacing x with y yields same result 4. Computing with intercepts yields the same t value. |
| 12 | 1. If sum(x^2 = sum(y^2) 2. set.seed(1) x=1:100 y=2\*x\*rnorm(100)  fit = lm(y~0+x) summary(fit)  fitinv = lm(x~0+y) summary(fitinv) 3. x=1:100 y=1:100  fit = lm(y~0+x) summary(fit)  fitinv = lm(x~0+y) summary(fitinv) |
| 13 | 1. See code 2. See code 3. 100, B0= -1, B1=0.5 4. This looks like a fairly linear model, with the intercept being near -1 as expected. 5. B0hat = -1.01, and B1hat = 0.499, both of these values are nearly spot on, especially for a sample size of 100 - large F with small p value means reject H0 6. See code 7. P value is high, we find that this does not improve our fit 8. Modifying the variance, say b decreasing the variance leads to better fit data as there is less noise. 9. More noise results in the opposite 10. 2.5 % 97.5 %  (Intercept) -1.1150804 -0.9226122  x 0.3925794 0.6063602   2.5 % 97.5 %  (Intercept) -1.0524003 -0.9133404  x2 0.4700169 0.6050909   2.5 % 97.5 %  (Intercept) -1.2073933 -0.8747967  x3 0.2922744 0.5779104 |
| 14 | 1. Y = 2+2x1+0.3+x2+eps 2. 0.8351212 3. Coefficients:   Estimate Std. Error t value Pr(>|t|)  (Intercept) 2.1305 0.2319 9.188 7.61e-15 \*\*\*  x1 1.4396 0.7212 1.996 0.0487 \*  x2 1.0097 1.1337 0.891 0.3754    1. B1 is close, with small p value, however, B1 has a larger p value, so overall, we accept H0 4. x1 1.9759 0.3963 4.986 2.66e-06 \*\*\*  We now have a small p value, and an accurate estimate, and we do reject H0: B1 = 0 5. x2 2.8996 0.6330 4.58 1.37e-05 \*\*\* this p value also suggests that we should reject H0: B1 = 0 6. No - we have high correlation |
| 15 | a)zn, indus, nox, rm, age, dis, rad, tax, ptratio, black, lstat, and medv are all statistically significant  b) Coefficients:   Estimate Std. Error t value Pr(>|t|)  (Intercept) 17.033228 7.234903 2.354 0.018949 \*  zn 0.044855 0.018734 2.394 0.017025 \*  indus -0.063855 0.083407 -0.766 0.444294  chas -0.749134 1.180147 -0.635 0.525867  nox -10.313535 5.275536 -1.955 0.051152 .  rm 0.430131 0.612830 0.702 0.483089  age 0.001452 0.017925 0.081 0.935488  dis -0.987176 0.281817 -3.503 0.000502 \*\*\*  rad 0.588209 0.088049 6.680 6.46e-11 \*\*\*  tax -0.003780 0.005156 -0.733 0.463793  ptratio -0.271081 0.186450 -1.454 0.146611  black -0.007538 0.003673 -2.052 0.040702 \*  lstat 0.126211 0.075725 1.667 0.096208 .  medv -0.198887 0.060516 -3.287 0.001087 \*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1    Residual standard error: 6.439 on 492 degrees of freedom  Multiple R-squared: 0.454, Adjusted R-squared: 0.4396  F-statistic: 31.47 on 13 and 492 DF, p-value: < 2.2e-16  For zn, dis, rad, black, and medv.  c) |

10 Code:

library(ISLR)  
attach(Carseats)  
  
lm.fit = lm(Sales~Price+Urban+US)  
summary(lm.fit)  
  
lm.fit2 = lm(Sales~Price+US)  
summary(lm.fit2)  
  
confint(lm.fit2)  
  
par(mfrow = c(2,2))  
plot(lm.fit2)

11 Code:

set.seed(1)  
x=rnorm(100)  
y=2\*x+rnorm(100)  
  
fit = lm(y~0+x)  
summary(fit)  
  
fitinv = lm(x~0+y)  
summary(fitinv)  
  
n = length(x)  
t = sqrt(n-1)\*(x%\*%y)/sqrt(sum(x^2)\*sum(y^2)-(x%\*%y)^2)  
as.numeric(t)  
  
fitint = lm(y~x)  
summary(fitint)  
  
fitinvint = lm(x~y)  
summary(fitinvint)  
  
13 Code:

x=rnorm(100)  
eps = rnorm(100,mean=0,sd=sqrt(0.25))  
y=-1+0.5\*x+eps  
length(y)  
  
fit = lm(y~x)  
summary(fit)  
  
fit2 = lm(y~x+I(x^2))  
summary(fit2)  
  
plot(x,y)  
abline(fit, col="orange")  
abline(-1,0.5, col="blue")  
legend("topleft", c("LeastSquares", "Regression"),col=c("orange","blue"),lty=c(1,1))  
  
x2=rnorm(100)  
eps2 = rnorm(100,mean=0,sd=sqrt(0.125))  
y2=-1+0.5\*x2+eps2  
length(y2)  
  
fit3 = lm(y2~x2)  
summary(fit3)  
  
plot(x2,y2)  
abline(fit, col="orange")  
abline(-1,0.5, col="blue")  
legend("topleft", c("LeastSquares", "Regression"),col=c("orange","blue"),lty=c(1,1))  
  
x3=rnorm(100)  
eps3 = rnorm(100,mean=0,sd=sqrt(0.75))  
y3=-1+0.5\*x3+eps3  
length(y3)  
  
fit4 = lm(y3~x3)  
summary(fit4)  
  
plot(x3,y3)  
abline(fit4, col="orange")  
abline(-1,0.5, col="blue")  
legend("topleft", c("LeastSquares", "Regression"),col=c("orange","blue"),lty=c(1,1))  
  
confint(fit)  
confint(fit3)  
confint(fit4)

14 Code

set.seed(1)  
x1=runif(100)  
x2=0.5\*x1+rnorm(100)/10  
y=2+2\*x1+0.3\*x2+rnorm(100)  
cor(x1,x2)  
plot(x1,x2)  
  
fit = lm(y~x1+x2)  
summary(fit)  
fit2 = lm(y~x1)  
summary(fit2)  
fit3 = lm(y~x2)  
summary(fit3)  
  
15 Code:

head(Boston[,-1])  
fit = lm(as.matrix(Boston[,-1])~Boston[,1])  
summary(fit)  
  
k = apply(Boston[,-1],2,function(x)summary(lm(Boston[,1]~x)))  
   
fitMulti = lm(crim ~ ., data = Boston)  
summary(fitMulti)