HW 2 - Predictive Modeling in Finance and Insurance

Dennis Goldenberg

2024-02-01

1. Nursing Home Utilization

```
# import packages
library(ggplot2)
library(magrittr)
```

```
# read in data
# WNH <- read.csv(file)
WNH <- read.csv('WiscNursingHome.csv', header = TRUE)
WNH$CRYEAR <- factor(WNH$CRYEAR)
WNH <- WNH[WNH$CRYEAR == 2001,]
```

1a) Estimation of Coefficients

```
#Generate variables to analyze
WNH$LOGTPY <- log(WNH$TPY)
WNH$LOGNUMBED <- log(WNH$NUMBED)
```

Using the generated variables, I calculate $x^T x$, adding in a column for the intercept:

```
x <- cbind(1,WNH$LOGNUMBED)
xTx <- t(x) %*% x
xTx
```

```
##          [,1]     [,2]
## [1,]  355.000 1582.334
## [2,] 1582.334 7138.724
```

Then, I find $\left(x^T x\right)^{-1}$:

```
xTxInv <- solve(xTx)
xTxInv
```

```
##            [,1]        [,2]
## [1,]  0.2343245 -0.05193920
## [2,] -0.0519392  0.01165267
```

Finally, I find $x^T y$:

```
y <- WNH$LOGTPY
xTy <- t(x) %*% y
xTy
```

```
##          [,1]
## [1,] 1550.747
```

```
## [2,] 6999.582
```

Using the formula for linear regression that $\beta = (x^T x)^{-1} x^T y$:

```
beta <- xTxInv %*% xTy
beta
```

```
##             [,1]
## [1,] -0.1746945
## [2,]  1.0192307
```

1b. The prediction Matrix

Since $\hat{y} = x\hat{\beta}$, and $\beta = (x^T x)^{-1} x^T y$, the prediction matrix $H = x\left(x^T x\right)^{-1} x^T$, so:

$$\hat{y} = x(x^T x)^{-1} x^T y = Hy$$

I find the diagonals of said matrix H and store them in "leverages" variable, as they represent the leverage of each data point; the first 6 outputs are shown below to verify with the Excel document:

```
H <- x %*% xTxInv %*% t(x)
leverages <- diag(H)
head(leverages)
```

```
## [1] 0.031426544 0.006281299 0.005372343 0.004351815 0.003224867 0.002906796
```

1c. Making Predictions

Since $\hat{y} = Hy$, I calculate and store in the "pred" variable, showing the first 6 predicted values for verification with excel:

```
pred <- H %*% y
head(pred)
```

```
##          [,1]
## [1,] 2.771261
## [2,] 3.812560
## [3,] 3.891001
## [4,] 3.998387
## [5,] 4.559011
## [6,] 4.278781
```

1d. Calculating Summary Statistics

The R^2 value is the proportion of variation explained by the regression. R^2_{adj} is adjusted for the number of predictors; its formula is:

$$R^2_{adj} = 1 - \frac{\frac{SSE}{n-p-1}}{\frac{SST}{n-1}} = 1 - \frac{\frac{SSE}{n-2}}{\frac{SST}{n-1}}$$

Then, the F statistic measures the significance of the regression; its formula is:

$$F_{stat} = \frac{\frac{SST-SSE}{p}}{\frac{SSE}{N-(p+1)}} = \frac{SST - SSE}{\frac{SSE}{N-2}}$$

The p-value is simply the probability that so much variation was observed by a model with no predictive power:

$$p = \mathbb{P}(F \geq F_{stat}), \text{ where } F \sim \text{F-dist}(1, N-2)$$

Finally, the mean squared error is just the sum of squared error divided by the number of degrees of freedom for said error, or $\frac{SSE}{N-2}$. All are calculated below:

```
SSR <- sum((mean(WNH$LOGTPY) - pred)^2)
SSE <- sum((WNH$LOGTPY - pred)^2)
SST <- sum((WNH$LOGTPY - mean(WNH$LOGTPY))^2)
n <- length(WNH$CRYEAR)
R_2 <- SSR/SST
R_2_adj <- 1 - (SSE/(n - 2))/(SST/(n - 1))
F_stat <- (SST - SSE)/(SSE/(n - 2))
p_reg <- 1 - pf(F_stat, 1, n - 2)
MSE <- SSE/(n - 2)
sumStats <- c(R_2, R_2_adj, F_stat, p_reg, MSE)
names(sumStats) <- c("R^2", "adj. R^2", "F", "p-val", "MSE")
t(sumStats)
```

```
##            R^2  adj. R^2        F p-val         MSE
## [1,] 0.9663796 0.9662843 10146.57     0 0.008786185
```

1e. Calculating Residuals

For observation i, the residual and standard residual have the following formulas:

$$e_i = y - y_i \text{ and } e_{i,std.} = \frac{e_i}{\sqrt{MSE} * \sqrt{1 - h_{ii}}}$$

I calculate both and print out the first 6 for both:

```
resid <- WNH$LOGTPY - pred
resid_std <- resid/(sqrt(MSE) * sqrt(1 - leverages))
f6res <- cbind(head(resid), head(resid_std))
colnames(f6res) <- c("Residuals", " Std. Residuals")
f6res
```

```
##         Residuals  Std. Residuals
## [1,]  0.044393547      0.48123077
## [2,]  0.006687011      0.07156491
## [3,]  0.067487578      0.72192719
## [4,]  0.051370555      0.54923870
## [5,]  0.001275984      0.01363473
## [6,] -0.027823096     -0.29726063
```

1f. Hypothesis testing

The hypotheses to test are:

$$H_0 : \beta_1 = 0 \text{ vs. } H_1 : \beta_1 \neq 0$$

I calculate the t-statistic, or $t = \frac{\hat{\beta}_1}{se\left(\hat{\beta}_1\right)}$, noting that $se\left(\hat{\beta}_1\right) = \sqrt{MSE} * \sqrt{(x^T x)^{-1}_{(2,2)}}$:

```
se_beta <- sqrt(MSE) * sqrt(xTxInv[2,2])
t_stat <- beta[2]/se_beta
t_stat
```

```
## [1] 100.7302
```

Note that $t_{stat} \sim F(ndf = 1, ddf = n - 2)$; therefore, I calculate the p-val of this statistic:

```
p_beta1 <- 1 - pf(t_stat, 1, n - 2)
p_beta1
```

```
## [1] 0
```

Note that $p < 0.05$; thus, the natural log of the number of beds has a statistically significant impact on the natural log of the number of patient years, and we reject H_0. Next, the following formula gives the confidence interval for 95% and 99% (so $\alpha = 0.05$ and $\alpha = 0.01$ respectively)

$$CI = \left(\hat{\beta}_1 - z^{(1-\frac{\alpha}{2})} se\left(\hat{\beta}_1\right), \hat{\beta}_1 + z^{1+\frac{\alpha}{2}} se\left(\hat{\beta}_1\right)\right)$$

I code this up and generate it form both significance levels:

```
lCI05 <- beta[2] - qnorm(0.975, 0, 1)*se_beta
rCI05 <- beta[2] + qnorm(0.975, 0, 1)*se_beta
lCI01 <- beta[2] - qnorm(0.995, 0, 1)*se_beta
rCI01 <- beta[2] + qnorm(0.995, 0, 1)*se_beta
CIMatrix <- matrix(data = c(lCI05,lCI01,rCI05,rCI01), nrow = 2)
rownames(CIMatrix) <- c("95% conf.", "99% conf.")
colnames(CIMatrix) <- c("Lower CI", "Upper CI")
CIMatrix
```

```
##            Lower CI Upper CI
## 95% conf. 0.9993990 1.039062
## 99% conf. 0.9931674 1.045294
```

1g. Prediction

The prediction is $\widehat{\ln(y)} = \hat{\beta}_0 + \hat{\beta}_1 \ln(x^*)$; the prediction interval is:

$$PI = \left(\hat{y} - z^{(1-\frac{\alpha}{2})}\sqrt{MSE}\sqrt{1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{(n-1)S_X^2}}, \hat{y} + z^{(1-\frac{\alpha}{2})}\sqrt{MSE}\sqrt{1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{(n-1)S_X^2}}\right)$$

I first show the prediction:

```
pred100 <- beta[1] + beta[2]*log(100)
num <- (log(100) - mean(WNH$LOGTPY))^2
denom <- sum((WNH$LOGTPY - mean(WNH$LOGTPY))^2)
lpred <- pred100 - qnorm(0.975, 0, 1)*sqrt(MSE)*sqrt(1 + 1/n + num/denom)
upred <- pred100 + qnorm(0.975, 0, 1)*sqrt(MSE)*sqrt(1 + 1/n + num/denom)
PIMatrix <- matrix(data = c(lpred, upred), nrow = 1)
colnames(PIMatrix) <- c("Lower PI", "Upper PI")
pred100
```

```
## [1] 4.519037
```

And now the prediction interval:

```
PIMatrix
```

```
##      Lower PI Upper PI
## [1,] 4.335006 4.703067
```

1h. Applying Linear Model to check

I apply a linear model to check all of my results:

```r
model <- lm("LOGTPY ~ LOGNUMBED", data = WNH)
summary(model)
```

```
##
## Call:
## lm(formula = "LOGTPY ~ LOGNUMBED", data = WNH)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.87482 -0.02201  0.01517  0.05316  0.28862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.17469    0.04537   -3.85  0.00014 ***
## LOGNUMBED    1.01923    0.01012  100.73  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09373 on 353 degrees of freedom
## Multiple R-squared:  0.9664, Adjusted R-squared:  0.9663
## F-statistic: 1.015e+04 on 1 and 353 DF,  p-value: < 2.2e-16
```

My results match with the model.