Logistic Regression
Econometrics
Sai Sandeep Chandaluri

Question 1

Multiple regression estimates the "Pure" or "Partial" effect of P2 on Sales controlling for co-variation with other variables. So to obtain the pure effect of P2, we try to understand the dependence of P2 on P1 and then take out the residuals from the regression of P2 on P1. These residuals will contain only the pure effect of P2 since all the effect of P1 is now removed. Regressing the Sales data over the residuals obtained from P2 ~ P1, we get the multiple regression coefficient on P2.

To obtain the multiple regression coefficient on P2 in the multi dataset, we follow the steps as below:

Step 1:

Regress P2 on P1 to purge P2 of relationship to P1.

```
library(DataAnalytics)
data(multi, package="DataAnalytics")

out = with(multi, lm(p2 ~ p1))
lmSumm(out)
```

```
## Multiple Regression Analysis:
##     2 regressors(including intercept) and 100 observations
##
## lm(formula = p2 ~ p1)
##
## Coefficients:
##             Estimate Std Error t value p value
## (Intercept)   0.8773    0.6062    1.45   0.151
## p1            1.4830    0.1189   12.48   0.000
## ---
## Standard Error of the Regression:  2.037
## Multiple R-squared:  0.614  Adjusted R-squared:  0.61
## Overall F stat: 155.63 on 1 and 98 DF, pvalue= 0
```

```
e_2.1 = lm(p2 ~ p1, data = multi)$residuals
```

Step 2:

Regress Sales on $e_{2,1}$:

```
out_residual = with(multi, lm(Sales ~ e_2.1))
lmSumm(out_residual)
```

```
## Multiple Regression Analysis:
##     2 regressors(including intercept) and 100 observations
##
## lm(formula = Sales ~ e_2.1)
##
## Coefficients:
##             Estimate Std Error t value p value
## (Intercept)    517.1    11.380   45.43       0
## e_2.1          108.8     5.645   19.27       0
## ---
## Standard Error of the Regression:  113.8
## Multiple R-squared:  0.791  Adjusted R-squared:  0.789
```

```
## Overall F stat: 371.44 on 1 and 98 DF, pvalue= 0
```

So, we can see that the multiple regression coefficient on P2 is 108.8

This can also be verified by doing a multiple regression of Sales ~ P1 + P2:

```
lm(formula = Sales ~ p1 + p2, data = multi)
```

```
##
## Call:
## lm(formula = Sales ~ p1 + p2, data = multi)
##
## Coefficients:
## (Intercept)           p1           p2
##      115.72       -97.66       108.80
```

Question 2

Using matrix formulas and R code to reproduce the least squares coefficients and Standard errors on slide 24 of Chapter II:

```
data(countryret, package="DataAnalytics")

y = countryret$usa
X = cbind(rep(1, length(y)), countryret$canada, countryret$uk, countryret$australia, countryret$france,

b = chol2inv(chol(crossprod(X))) %*% crossprod(X,y)

b
```

```
##              [,1]
## [1,]  0.006135614
## [2,]  0.444362109
## [3,]  0.225690196
## [4,] -0.056688434
## [5,]  0.166742081
## [6,] -0.064792831
## [7,] -0.051027942
```

```
e = y - X%*%b
ssq=sum(e*e)/(length(y)-ncol(X))
Var_b=ssq*chol2inv(chol(crossprod(X)))
std_err=sqrt(diag(Var_b))
names(std_err)=c("intercept","canada","uk","australia","france","germany","japan")

std_err
```

```
##  intercept     canada         uk  australia     france    germany      japan
## 0.00230897 0.06958673 0.06491489 0.05036627 0.06133779 0.05723881 0.03461495
```

Question 3

Regression of `VWNFX` on `vwretd`:

```
library(reshape2)
data(Vanguard, package="DataAnalytics")
Van=Vanguard[,c(1,2,5)] # grab relevant cols
V_reshaped=dcast(Van,date~ticker,value.var="mret")
```

```r
data(marketRf)
Van_mkt=merge(V_reshaped,marketRf,by="date")

reg_out = lm(formula = VWNFX ~ vwretd, data = Van_mkt)

lmSumm(reg_out)
```

```
## Multiple Regression Analysis:
##     2 regressors(including intercept) and 336 observations
##
## lm(formula = VWNFX ~ vwretd, data = Van_mkt)
##
## Coefficients:
##             Estimate Std Error t value p value
## (Intercept) 0.001074 0.0009468    1.13   0.257
## vwretd      0.889100 0.0207300   42.90   0.000
## ---
## Standard Error of the Regression:  0.01698
## Multiple R-squared:  0.846  Adjusted R-squared:  0.846
## Overall F stat: 1840.19 on 1 and 334 DF, pvalue= 0
```

a. We know for a regression with one variable (for VWNFX ~ vwretd):

$$Return_{VWNFX} = b_0 + b_1(Return_{market}) \pm t^*_{N-2,\alpha/2} s_{pred}$$

We know that the prediction error s_{pred} (for regression with one variable) is given by

$$s_{pred} = s(1 + \frac{1}{N} + \frac{(X_f - \bar{X})^2}{(N-1)s_x^2})^{\frac{1}{2}}$$

From regression above, we know $s = 0.01698$. Given $vwretd = 0.05$.

Computing the mean and variance of $vwretd$, we get

```r
mean(as.numeric(unlist(reg_out$model["vwretd"])))
```

```
## [1] 0.009435866
```

```r
var(as.numeric(unlist(reg_out$model["vwretd"])))
```

```
## [1] 0.002003526
```

$\bar{X} = 0.0094359$, $s_x^2 = 0.002003$ and $N = 336$. Substituting these values in above equation, we get

```r
0.01698 * sqrt(sum(1, 1/336, (0.05-0.009435866)^2/(337*0.002003526)))
```

```
## [1] 0.0170259
```

$$s_{pred} = 1.7026\%$$

Computation of $t^*_{N-2,\alpha/2}$ for 90% interval:

```r
qt(0.05, df = 336 - 2)
```

```
## [1] -1.649429
```

From regression above, we have $b_0 = 0.001074$ and $b_1 = 0.889100$. Therefore,

$$Return_{VWNFX} = 0.001074 + (0.889100 * 0.05) \pm (-1.649429 * 0.0170259)$$

$$= 0.045529 \pm 0.02808301$$

Hence, 90% prediction interval for $VWNFX$ when $vwretd = 0.05$ is $(0.01744599, 0.07361201)$ which is 1.74% and 7.36% approximately.

b. Checking the above result with predict command:

```
predict(reg_out,
new = data.frame(vwretd = 0.05), int = "prediction", level = 0.90)
```

```
##          fit        lwr        upr
## 1 0.04553055 0.01744646 0.07361465
```

We can see that the 90% prediction interval in (a) and (b) are equal approximately.

Question 4

a. Given μ and Σ :

$$\mu = \begin{bmatrix} 0.010 \\ 0.015 \\ 0.025 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 0.0016 & 0.0010 & 0.0015 \\ & 0.0020 & 0.0019 \\ & & 0.0042 \end{bmatrix}$$

Correlation matrix is computed by dividing each element (i, j) of Σ by σ_i and σ_j. For this we first compute standard deviation matrix D (σ of each asset), which is square root of the diagonal matrix of Σ. Then, we compute the correlation matrix by dividing Σ by $D * D^t$

```
sigma <- matrix(c(0.0016,0.0010,0.0015,0.0010,0.0020,0.0019,0.0015,0.0019,0.0042), 3, 3); sigma
```

```
##        [,1]   [,2]   [,3]
## [1,] 0.0016 0.0010 0.0015
## [2,] 0.0010 0.0020 0.0019
## [3,] 0.0015 0.0019 0.0042
```

```
D <- matrix(sqrt(diag(sigma)));D #Standard deviations of each variable
```

```
##            [,1]
## [1,] 0.04000000
## [2,] 0.04472136
## [3,] 0.06480741
```

```
sigma/(D %*% t(D)) # Correlation Matrix
```

```
##           [,1]      [,2]      [,3]
## [1,] 1.0000000 0.5590170 0.5786376
## [2,] 0.5590170 1.0000000 0.6555623
## [3,] 0.5786376 0.6555623 1.0000000
```

b. Mean of a portfolio with mean return μ and weights of assets W is computed by matrix multiplication of W and μ.

Standard deviation of portfolio with variance-covariance matrix Σ and weights W is obtained by computing square-root of the matrix multiplication $W * \Sigma * W^t$

```r
mu <- matrix(c(0.010, 0.015, 0.025), 3, 1); mu
```

```
##       [,1]
## [1,] 0.010
## [2,] 0.015
## [3,] 0.025
```

```r
weights <- matrix(c(0.3,0.4,0.3), 1, 3); weights
```

```
##      [,1] [,2] [,3]
## [1,]  0.3  0.4  0.3
```

```r
mean_return_portfolio <- weights %*% mu; mean_return_portfolio #Mean return from the portfolio
```

```
##        [,1]
## [1,] 0.0165
```

```r
sd_portfolio <- sqrt(weights %*% sigma %*% t(weights))
sd_portfolio # Standard Deviation of the Portfolio
```

```
##            [,1]
## [1,] 0.04252058
```

Based on above computations, mean return of the portfolio is 1.65% and standard deviation of the portfolio is 4.252%