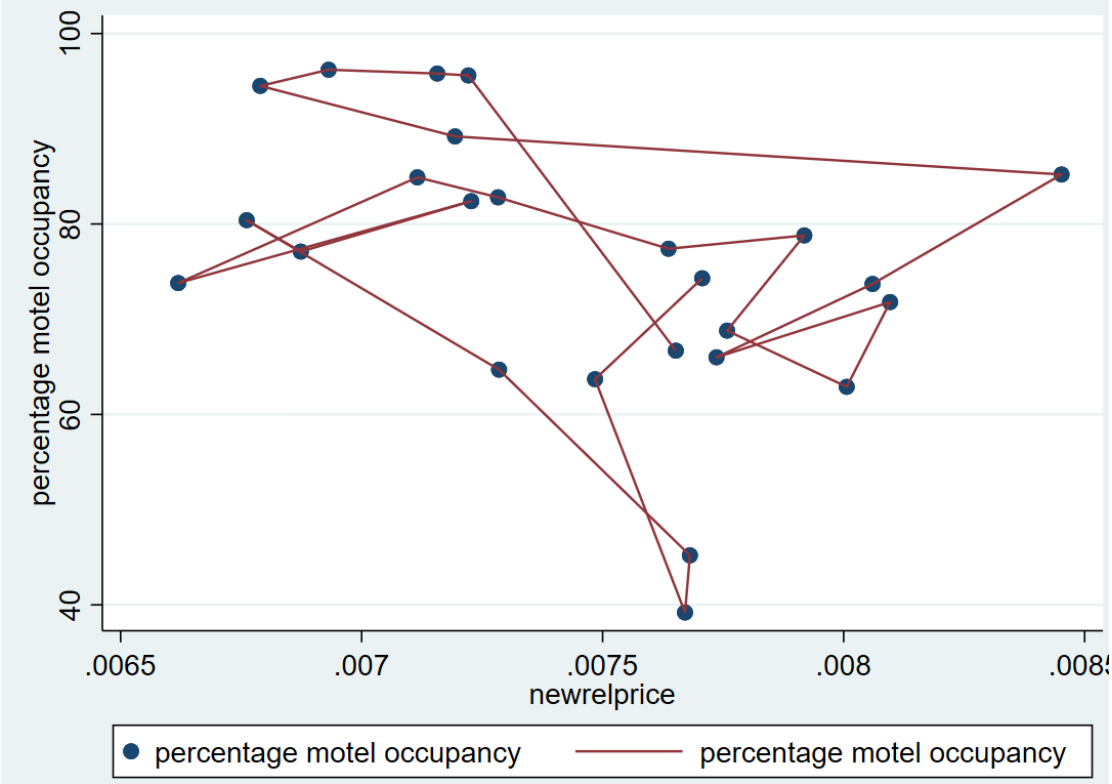
# 1.

## (1)




從圖表中可以得知，motel_pct 與 newrelprice(100relprice)之間並無相關性

## (2)

| | (1) |
|---|---|
| | motel_pct |
| newrelprice | -12211.9* |
| | (-2.09) |
| _cons | 166.7*** |
| | (3.82) |
| *N* | 25 |

*t* statistics in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

預測狀況斜率應為正值，當經爭對手每間客房收取的價格百分比越高時，我方汽車旅館的入住率相對來說會提高，也就是當經爭對手提高诶間價格收取的百分比時，顧客就會進行價格比較，從而選擇我方的汽車旅館居住。

**(4)**

| | (1)<br>motel_pct |
|---|---|
| repair | -13.24* |
| | (-2.22) |
| _cons | 79.35*** |
| | (25.16) |
| *N* | 25 |

*t* statistics in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

當 repair=0 時，motel_pct=79.35

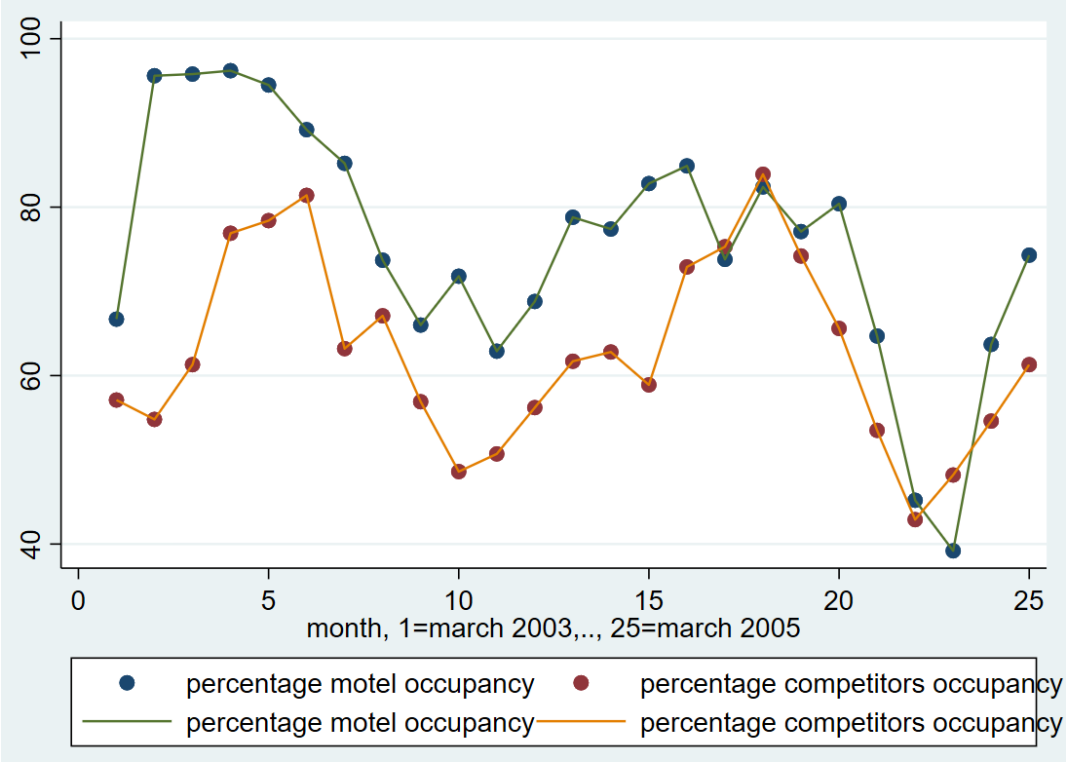
```
. di _b[_cons]+ _b[repair]*0
79.35
```

當 repair=1 時，motel_pct=66.114

```
. di _b[_cons]+ _b[repair]*1
66.114286
```

## 2.

### (1)




入住率會隨著時間的不同而有所變化，大概在第 9 個月、第 15 個月、第 23 個月時雙方的入住率都下滑，而相較之下我方汽車旅館的入住率普遍都比競爭對手來的高。

| motel_pct | Coefficient | Std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| comp_pct | .8646393 | .2027119 | 4.27 | 0.000 | .4452978 | 1.283981 |
| _cons | 21.39999 | 12.90686 | 1.66 | 0.111 | -5.299896 | 48.09987 |

95%區間估計值為 [0.4452978, 1.283981]

### (2)

當 comp_pct 為 70 時，motel_pct 為 81.91474

```
clear
cd "C:\Users\user\OneDrive\桌面"
use "C:\Users\user\OneDrive\桌面\motel"

//(1.1)
gen newrelprice = relprice/100
twoway (scatter motel_pct newrelprice)(line motel_pct newrelprice)

//(1.2)
reg motel_pct newrelprice
esttab using "1.2.rtf"

//(1.3)

//(1.4)
reg motel_pct repair
esttab using "1.4.rtf"
di _b[_cons]+ _b[repair]*1
di _b[_cons]+ _b[repair]*0

//(2.1)
twoway(scatter motel_pct comp_pct time)(line motel_pct comp_pct time)
reg motel_pct comp_pct

//(2.2)
di _b[_cons]+ _b[ comp_pct]*70
```

**3.**

**(1)**

| Variable | Obs | Mean | Std. dev. | Min | Max |
|---|---|---|---|---|---|
| advanced | 1,200 | .2141667 | .410414 | 0 | 1 |
| alcbev | 1,200 | 7.158092 | 18.01358 | 0 | 333.33 |
| appar | 1,200 | 27.5337 | 50.23839 | 0 | 575.33 |
| college | 1,200 | .3075 | .4616507 | 0 | 1 |
| entert | 1,200 | 49.818 | 65.64827 | 0 | 814.33 |
| food | 1,200 | 114.4431 | 72.6575 | 9.63 | 476.67 |
| foodaway | 1,200 | 49.27085 | 65.28361 | 0 | 1179 |
| health | 1,200 | 84.14707 | 124.5632 | 0 | 1402.89 |
| income | 1,200 | 72.14264 | 41.65228 | 10 | 200 |
| smsa | 1,200 | .8875 | .3161124 | 0 | 1 |

monthly food expenditure during past quarter per person, $

| | Percentiles | Smallest | | |
|---|---|---|---|---|
| 1% | 18.295 | 9.63 | | |
| 5% | 31.3 | 9.63 | | |
| 10% | 40.745 | 12.74 | Obs | 1,200 |
| 25% | 57.78 | 13.96 | Sum of wgt. | 1,200 |
| 50% | 99.8 | | Mean | 114.4431 |
| | | Largest | Std. dev. | 72.6575 |
| 75% | 145.555 | 433.33 | | |
| 90% | 209.685 | 433.33 | Variance | 5279.112 |
| 95% | 260 | 462.22 | Skewness | 1.354445 |
| 99% | 361.11 | 476.67 | Kurtosis | 5.373787 |

**FOOD 統計量:**

平均值: 114.4431

中位數: 99.8

最小值: 9.63

最大值: 476.67

標準差: 72.6575

household monthly income during past year, $100 units

| | Percentiles | Smallest | | |
|---|---|---|---|---|
| 1% | 11.995 | 10 | | |
| 5% | 18.915 | 10 | | |
| 10% | 24.46 | 10.43 | Obs | 1,200 |
| 25% | 40 | 10.58 | Sum of wgt. | 1,200 |
| 50% | 65.29 | | Mean | 72.14264 |
| | | Largest | Std. dev. | 41.65228 |
| 75% | 96.915 | 199.17 | | |
| 90% | 129.955 | 199.33 | Variance | 1734.912 |
| 95% | 157.355 | 199.53 | Skewness | .8455717 |
| 99% | 189.375 | 200 | Kurtosis | 3.322891 |

**INCOME 統計量:**

平均值: 72.14264

中位數: 65.29

最小值: 10

最大值: 200

標準差: 41.65228

## (2)

**reg food income**

| Source | SS | df | MS | Number of obs | = | 1,200 |
|---|---|---|---|---|---|---|
| | | | | F(1, 1198) | = | 52.89 |
| Model | 267625.41 | 1 | 267625.41 | Prob > F | = | 0.0000 |
| Residual | 6062029.78 | 1,198 | 5060.12503 | R-squared | = | 0.0423 |
| | | | | Adj R-squared | = | 0.0415 |
| Total | 6329655.19 | 1,199 | 5279.11192 | Root MSE | = | 71.135 |

| food | Coefficient | Std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| income | .3586867 | .049321 | 7.27 | 0.000 | .2619215 | .455452 |
| _cons | 88.5665 | 4.108188 | 21.56 | 0.000 | 80.50646 | 96.62654 |

## (3)

**reg ln_food ln_income**

| Source | SS | df | MS | Number of obs | = | 1,200 |
|---|---|---|---|---|---|---|
| | | | | F(1, 1198) | = | 41.18 |
| Model | 16.9590002 | 1 | 16.9590002 | Prob > F | = | 0.0000 |
| Residual | 493.406139 | 1,198 | .411858213 | R-squared | = | 0.0332 |
| | | | | Adj R-squared | = | 0.0324 |
| Total | 510.365139 | 1,199 | .425658999 | Root MSE | = | .64176 |

| ln_food | Coefficient | Std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| ln_income | .1863054 | .0290335 | 6.42 | 0.000 | .1293432 | .2432675 |
| _cons | 3.778932 | .1203492 | 31.40 | 0.000 | 3.542814 | 4.015051 |

## (4)

**reg food ln_income**

| Source | SS | df | MS | Number of obs | = | 1,200 |
|---|---|---|---|---|---|---|
| | | | | F(1, 1198) | = | 47.32 |
| Model | 240525.871 | 1 | 240525.871 | Prob > F | = | 0.0000 |
| Residual | 6089129.32 | 1,198 | 5082.74568 | R-squared | = | 0.0380 |
| | | | | Adj R-squared | = | 0.0372 |
| Total | 6329655.19 | 1,199 | 5279.11192 | Root MSE | = | 71.293 |

| food | Coefficient | Std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| ln_income | 22.18738 | 3.225332 | 6.88 | 0.000 | 15.85946 | 28.51531 |
| _cons | 23.56848 | 13.3696 | 1.76 | 0.078 | -2.661956 | 49.79892 |

**(6)**

**reg food income R-square: 0.0423**

**reg ln_food ln_income R-square: 0.0332**

**reg food ln_income R-square: 0.0380**

三種模型裡，**reg food income** 看起來最符合資料分配，當我們的收入越多時，自然而然對於食物的支出也就會越高。

```
clear
cd "C:\Users\user\OneDrive\桌面"
use "C:\Users\user\OneDrive\桌面\cex5_small"

//(3.1)
sum, detail
sum food, detail
sum income, detail

//(3.2)
reg food income

//(3.3)
gen ln_food = log(food)
gen ln_income = log(income)
reg ln_food ln_income

//(3.4)
reg food ln_income

//(3.5)
global ests="log_lin log_log lin_lin"

foreach m of global ests{
    quietly{
        estimates restore `m'
        capture drop res_`m'
        predict res_`m', residual
    }
    di e(estimates_title)
    jb res_`m'
    di ""

}
```

## 4.

### (1)

| | b | se | t | p |
|---|---|---|---|---|
| prbarr | -0.050*** | 0.004 | -13.499 | 0.000 |
| prbpris | 0.024*** | 0.007 | 3.451 | 0.001 |
| prbconv | -0.003*** | 0.000 | -8.005 | 0.000 |
| polpc | 3.077*** | 0.260 | 11.838 | 0.000 |
| _cons | 0.033*** | 0.003 | 10.051 | 0.000 |

因此我們可以看到，監禁及警察對於犯罪率具有正向影響；而逮捕機率及定罪機率對於犯罪率有負向影響，其中警察數量可以解釋的方式是由於犯罪率過高的問題，因此需要派出更多警察維持治安(此有內生性問題產生)。

### (2)

| crmrte | Coefficient | Std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| prbarr | -.0379504 | .0039428 | -9.63 | 0.000 | -.045693 | -.0302078 |
| _cons | .0432523 | .0013869 | 31.19 | 0.000 | .0405287 | .0459759 |

### (3)

| | b | se | t | p |
|---|---|---|---|---|
| prbarr | -0.043*** | 0.005 | -8.322 | 0.000 |
| prbpris | 0.007 | 0.005 | 1.377 | 0.169 |
| prbconv | -0.002*** | 0.000 | -7.587 | 0.000 |
| polpc | 2.562*** | 0.193 | 13.305 | 0.000 |
| c.prbarr#c~r | 0.007** | 0.003 | 2.677 | 0.008 |
| density | 0.008*** | 0.001 | 13.435 | 0.000 |
| urban | -0.001 | 0.003 | -0.446 | 0.656 |
| _cons | 0.027*** | 0.003 | 10.298 | 0.000 |

從這裡可以看出，當逮捕機率逐漸往上提升時，反而會具有反效果，造成犯罪機率提高；而當伊格地區的人口密度越高時，相對的犯罪率也會隨之上升。

**(4)**

| | Delta-method dy/dx | std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| prbarr | **-.0390411** | **.0039605** | **-9.86** | **0.000** | **-.0468188** | **-.0312635** |

邊際影響值為 -0.390411

**(7)**

| model | R^2 | adjusted R^2 | AIC | SC | RMSE |
|---|---|---|---|---|---|
| 1 | 0.31 | 0.30105 | -3486.27928 | -3464.05068 | 0.01515 |
| 2 | 0.34 | 0.33510 | -3516.74873 | -3490.07441 | 0.01478 |
| 3 | 0.35 | 0.34868 | -3528.76031 | -3497.64027 | 0.01462 |
| 4 | 0.47 | 0.46036 | -3646.27199 | -3610.70624 | 0.01331 |
| 5 | 0.68 | 0.67205 | -3959.05600 | -3919.04452 | 0.01038 |

```
clear
cd "C:\Users\user\OneDrive\桌面"
use "C:\Users\user\OneDrive\桌面\crime"

//(1)
reg crmrte prbarr prbpris prbconv polpc
estout, cell("b(star fmt(3)) se t p")

//(2)
predict y_hat, xb
reg crmrte prbarr

//(3)
reg crmrte prbarr prbpris prbconv polpc c.prbarr#c.prbarr density urban
estout, cell("b(star fmt(3)) se t p")

//(4)
margin, dydx(prbarr)

//(7)
global x_1 prbarr prbpris prbconv polpc
global x_2 $x_1 c.prbarr#c.prbarr
global x_3 $x_1 c.prbarr#c.prbarr c.prbpris#c.prbpris
global x_4 $x_1 c.prbarr#c.prbarr c.prbpris#c.prbpris c.prbconv#c.prbconv
global x_5 $x_1 c.prbarr#c.prbarr c.prbpris#c.prbpris c.prbconv#c.prbconv density

foreach a of numlist 1/5 {
    local x = "${x_`a'}"
      di "現在的迴歸式: `x'"
      eststo est_`a': reg crmrte `x'
}

esttab, mti drop(*) r(2) ar2(5) aic(5) bic(5) scalar(rmse) sfmt(5)
```