Zwiller_Linear_Regression_Homework

Thomas Zwiller

2025-05-12

1. Visualization

In linear regression, data distribution and relationship are critical. In particular, histograms examine if the observations are skewed while scatter plot examine if a relationship is linear. Examine histograms and scatter plots for the variables of your interests. If there are some concerns, try different wrangling to mitigate the concerns.

```
#loading in files
load(file = "/Users/TomTheIntern/Desktop/Mendoza/Mod 1/Stats/Case 2/CroqPain_new.rda")
load(file = "/Users/TomTheIntern/Desktop/Mendoza/Mod 1/Stats/Case 2/CroqPainFix.rda")

#cleaning by dropping repeats
CroqPaiFix <- unique(CroqPaiFix)
#In total there were five duplicates that were dropped from the set, which should give us better data t
```

Look at correlations between variables and try to identify sources of concern. In R, you can use cor function. Pay particular attention to the correlation for total and P15 through P55. Do these correlations make sense to you?

First things first we need to make the performance ratio that is mentioned on page 303 of the case study, which is the performance ratio.

The ratio can be measured as Sales - Variable Cost / Invested Capital

Essentially, the Performance Ratio = Operating Earnings / Invested Capital.

This means that we need to use the EARN Variable and the K Variable

```
#Creating the Per Ratio in the CroqPaiFix database
CroqPaiFix$PerRatio <- ((CroqPaiFix$EARN * 1000) / (CroqPaiFix$K * 1000)) * 100
```

```
#Created a correlation with just Total and the P values for easier readability
cor(CroqPaiFix)
```

```
##                STOR        EARN           K        SIZE        EMPL
## STOR      1.00000000  0.11443187 -0.08359174 -0.03962301 -0.21010206
## EARN      0.11443187  1.00000000  0.23999537  0.43662341 -0.11407372
## K        -0.08359174  0.23999537  1.00000000  0.81391994  0.03944858
## SIZE     -0.03962301  0.43662341  0.81391994  1.00000000  0.05198322
## EMPL     -0.21010206 -0.11407372  0.03944858  0.05198322  1.00000000
## total     0.13558742  0.59324584 -0.01388264 -0.01934203 -0.09930071
## P15       0.07989695  0.62822976 -0.06703147 -0.05285959 -0.09529269
```

```
## P25       0.07270192  0.23137583 -0.06006042 -0.07788959 -0.02374946
## P35       0.09333498  0.63388357 -0.04953996 -0.03056222 -0.11583549
## P45       0.08185205  0.62852909 -0.03560543 -0.02008147 -0.10522149
## P55       0.23363347  0.39857816  0.11011677  0.05933441 -0.09010221
## INC       0.03767486  0.46494332  0.20874338  0.18027410  0.08994681
## COMP     -0.15165798 -0.13553328 -0.28430803 -0.17410438  0.12108288
## NCOMP     0.18709967  0.10610076 -0.11346915 -0.01945768  0.11213447
## NREST     0.08173571  0.33758250 -0.10371030 -0.09638736 -0.15757591
## PRICE    -0.11641423 -0.18002271  0.59742849  0.06643602  0.07605554
## CLI      -0.22684755  0.03638913  0.18357219  0.04974946  0.13895918
## PerRatio  0.16676453  0.85549289 -0.20847497  0.03937754 -0.11898151
##                total         P15          P25         P35          P45
## STOR       0.13558742  0.07989695  0.072701923  0.09333498  0.081852050
## EARN       0.59324584  0.62822976  0.231375827  0.63388357  0.628529086
## K         -0.01388264 -0.06703147 -0.060060422 -0.04953996 -0.035605432
## SIZE      -0.01934203 -0.05285959 -0.077889588 -0.03056222 -0.020081469
## EMPL      -0.09930071 -0.09529269 -0.023749464 -0.11583549 -0.105221491
## total      1.00000000  0.96090336  0.577829482  0.96249149  0.957891746
## P15        0.96090336  1.00000000  0.424044758  0.98201005  0.980578562
## P25        0.57782948  0.42404476  1.000000000  0.43217444  0.414635344
## P35        0.96249149  0.98201005  0.432174442  1.00000000  0.987687408
## P45        0.95789175  0.98057856  0.414635344  0.98768741  1.000000000
## P55        0.77372509  0.68210294  0.289146903  0.67181934  0.650376692
## INC        0.10721456  0.15492256  0.016315832  0.13542663  0.136120661
## COMP      -0.13863280 -0.10947281 -0.006783795 -0.12376021 -0.125266797
## NCOMP      0.06972993  0.07378689  0.101320518  0.06894449  0.082900797
## NREST      0.05438217  0.06791254  0.005372817  0.10233186  0.094819141
## PRICE      0.04097744 -0.02547230  0.080881029 -0.01255491 -0.009561738
## CLI        0.21210064  0.20918379  0.089024963  0.20110491  0.228389133
## PerRatio   0.58295460  0.64714156  0.240157264  0.63751153  0.621505069
##                   P55          INC         COMP        NCOMP        NREST
## STOR       0.233633467  0.037674863 -0.151657984  0.187099668  0.081735709
## EARN       0.398578156  0.464943320 -0.135533277  0.106100761  0.337582497
## K          0.110116767  0.208743381 -0.284308034 -0.113469151 -0.103710296
## SIZE       0.059334413  0.180274103 -0.174104384 -0.019457678 -0.096387365
## EMPL      -0.090102206  0.089946811  0.121082876  0.112134472 -0.157575914
## total      0.773725089  0.107214560 -0.138632796  0.069729935  0.054382170
## P15        0.682102936  0.154922555 -0.109472808  0.073786890  0.067912539
## P25        0.289146903  0.016315832 -0.006783795  0.101320518  0.005372817
## P35        0.671819344  0.135426627 -0.123760215  0.068944493  0.102331861
## P45        0.650376692  0.136120661 -0.125266797  0.082900797  0.094819141
## P55        1.000000000  0.014800732 -0.201811634  0.006692237 -0.022993875
## INC        0.014800732  1.000000000 -0.075351039  0.170901553 -0.058247103
## COMP      -0.201811634 -0.075351039  1.000000000  0.159834171  0.105719470
## NCOMP      0.006692237  0.170901553  0.159834171  1.000000000  0.005968695
## NREST     -0.022993875 -0.058247103  0.105719470  0.005968695  1.000000000
## PRICE      0.146631925  0.004017066 -0.296322434 -0.196686680 -0.063450349
## CLI        0.147336753  0.103740881  0.023114406 -0.008101416 -0.291381379
## PerRatio   0.352407807  0.442011303  0.040338967  0.169339879  0.320071673
##                 PRICE          CLI    PerRatio
## STOR      -0.116414228 -0.226847553  0.16676453
## EARN      -0.180022705  0.036389125  0.85549289
## K          0.597428485  0.183572188 -0.20847497
## SIZE       0.066436016  0.049749459  0.03937754
```

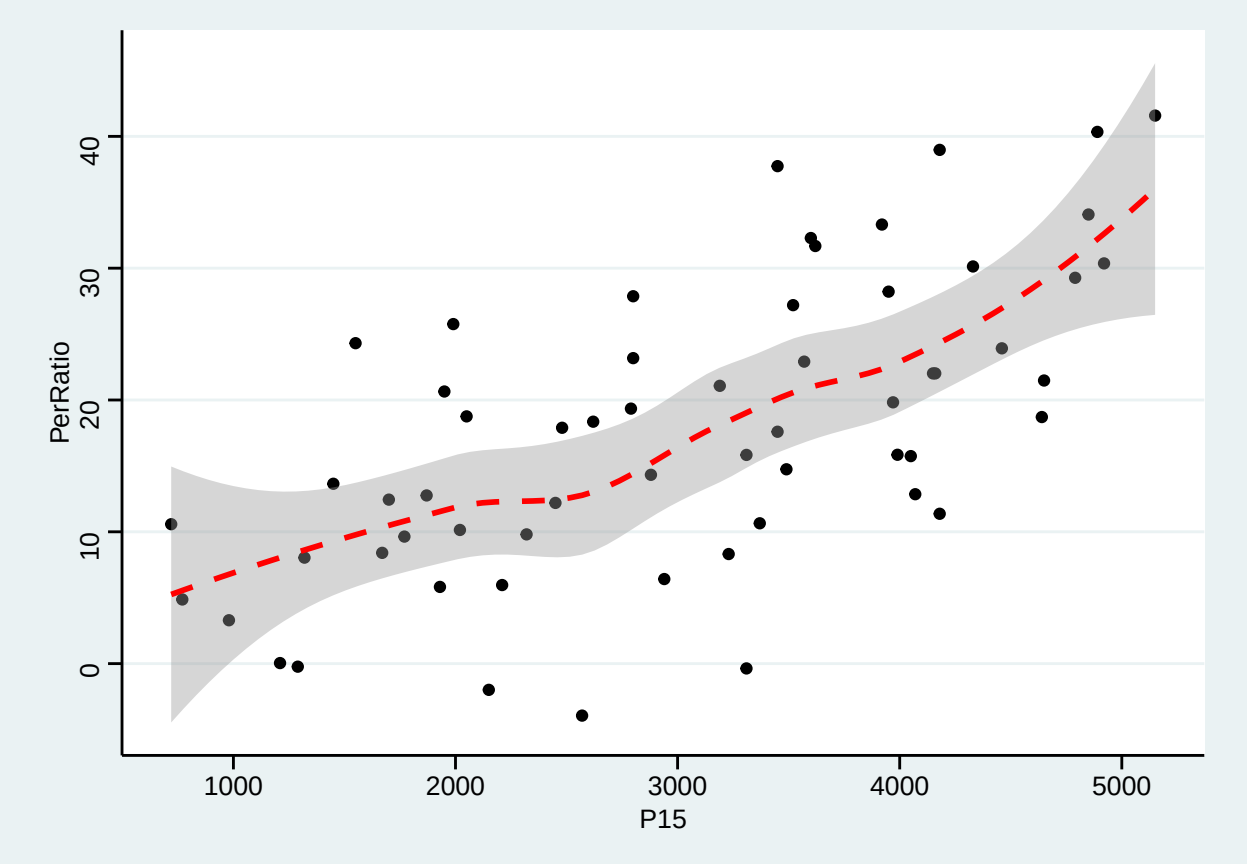
```
## EMPL      0.076055538  0.138959178 -0.11898151
## total     0.040977440  0.212100643  0.58295460
## P15      -0.025472303  0.209183791  0.64714156
## P25       0.080881029  0.089024963  0.24015726
## P35      -0.012554906  0.201104910  0.63751153
## P45      -0.009561738  0.228389133  0.62150507
## P55       0.146631925  0.147336753  0.35240781
## INC       0.004017066  0.103740881  0.44201130
## COMP     -0.296322434  0.023114406  0.04033897
## NCOMP    -0.196686680 -0.008101416  0.16933988
## NREST    -0.063450349 -0.291381379  0.32007167
## PRICE     1.000000000  0.258278578 -0.43381193
## CLI       0.258278578  1.000000000 -0.06725651
## PerRatio -0.433811930 -0.067256511  1.00000000
```

We can see from the data that in particular, P15 (.647), P35 (.637) and P45 (.6215) have an incredibly high correlation with the total, so if we want to do any modeling, these variables might be our best bet.

Let's graph them.

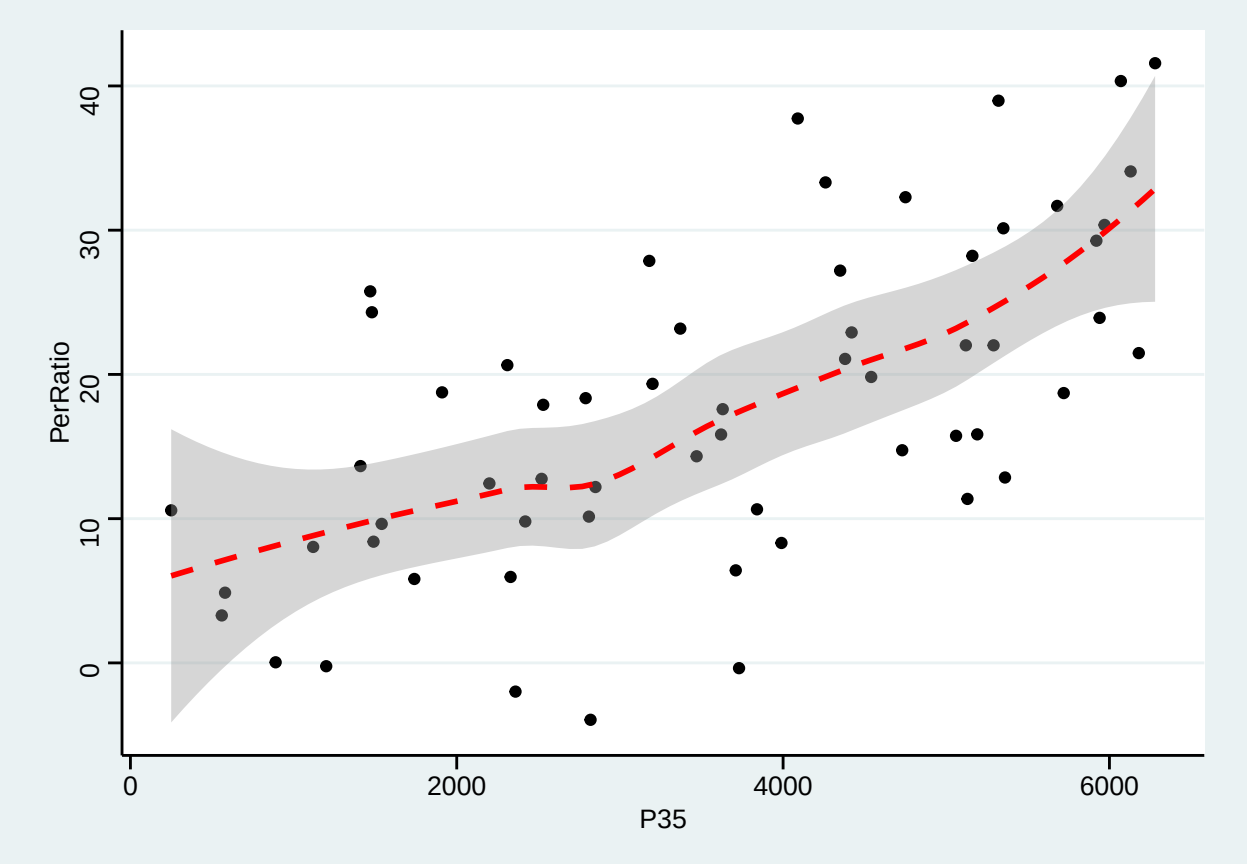
```
library(ggplot2)
library(ggthemes)
#graphing P15 and totals correlation
ggplot(CroqPaiFix, aes(x = P15, y = PerRatio)) +
  geom_point() +
  geom_smooth(method = loess, color = 'red', linetype = 'dashed')+
  theme_stata()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



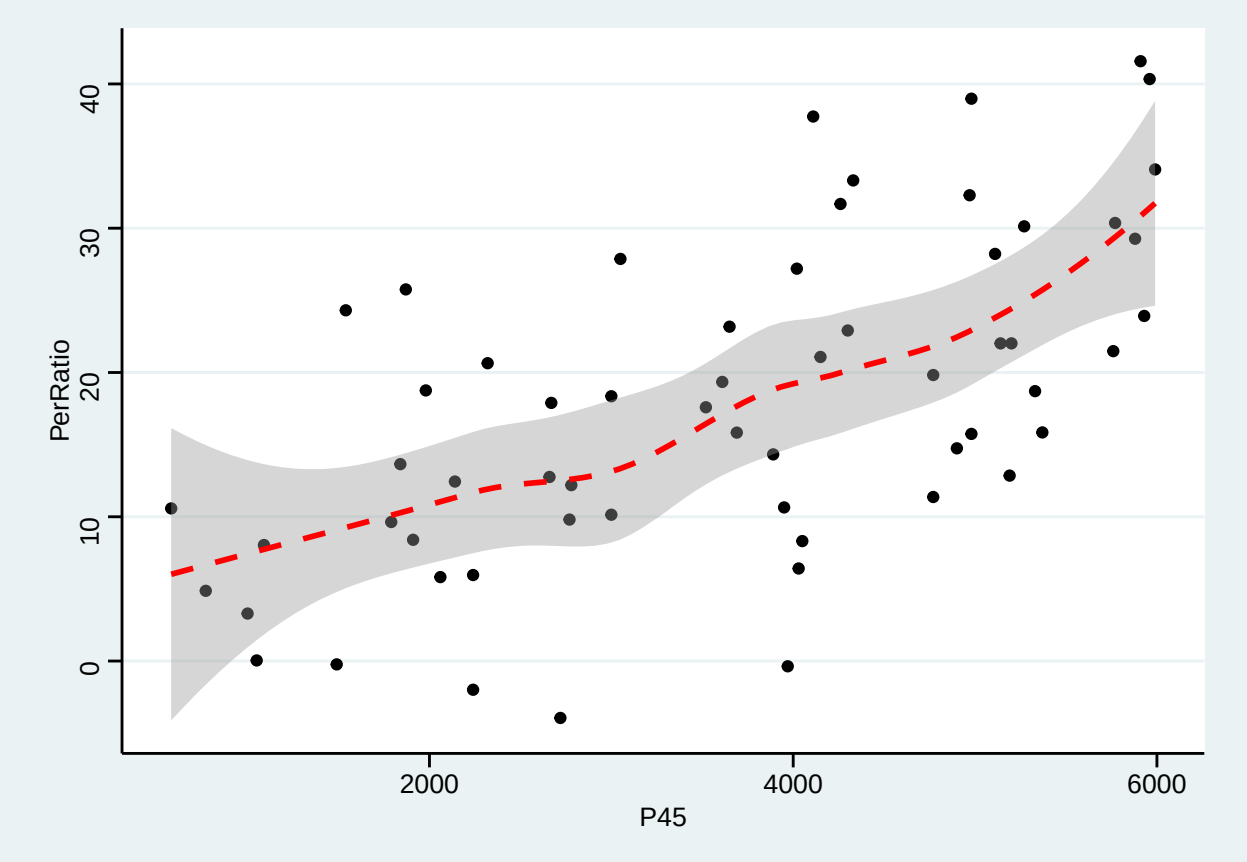

```
#graphing P35 and totals correlation
ggplot(CroqPaiFix, aes(x = P35, y = PerRatio)) +
  geom_point() +
  geom_smooth(method = loess, color = 'red', linetype = 'dashed')+
  theme_stata()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
#graphing P45 and totals correlation
ggplot(CroqPaiFix, aes(x = P45, y = PerRatio)) +
  geom_point() +
  geom_smooth(method = loess, color = 'red', linetype = 'dashed')+
  theme_stata()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



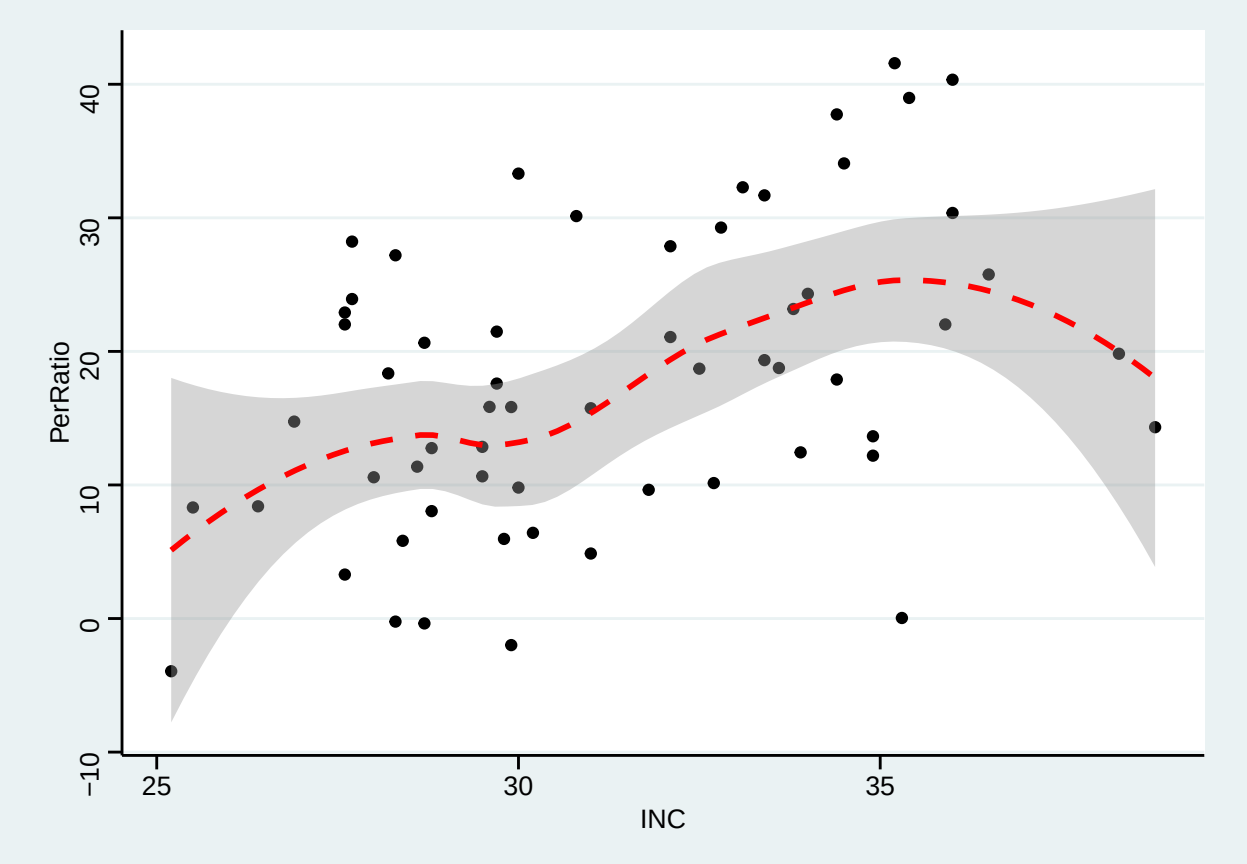

```
##Created a correlation with just the performance ratio and remaining values
cor(CroqPaiFix[ ,12:17], CroqPaiFix[ , 18])
```

```
##          PerRatio
## INC     0.44201130
## COMP    0.04033897
## NCOMP   0.16933988
## NREST   0.32007167
## PRICE -0.43381193
## CLI    -0.06725651
```

While none of these variables have an incredibly high correlation, it may be worth using INC and PRICE

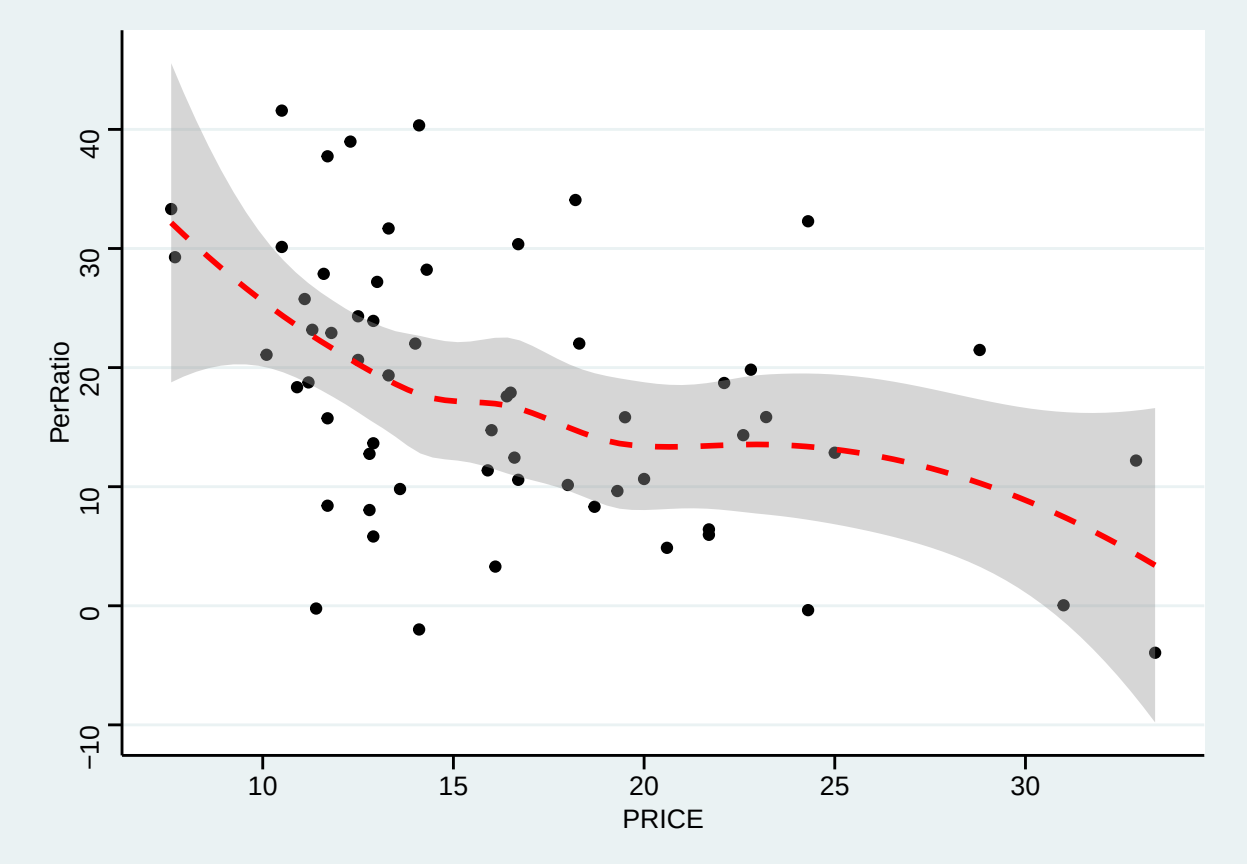
```
ggplot(CroqPaiFix, aes(x = INC, y = PerRatio)) +
  geom_point() +
  geom_smooth(method = loess, color = 'red', linetype = 'dashed')+
  theme_stata()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(CroqPaiFix, aes(x = PRICE, y = PerRatio)) +
  geom_point() +
  geom_smooth(method = loess, color = 'red', linetype = 'dashed')+
  theme_stata()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



It is also worth looking at the Earn, K, Size and Employee values in relation to total as well.

```
cor(CroqPaiFix[ ,2:5], CroqPaiFix[ , 18])
```

```
##          PerRatio
## EARN  0.85549289
## K    -0.20847497
## SIZE  0.03937754
## EMPL -0.11898151
```

```
library("openxlsx")
```

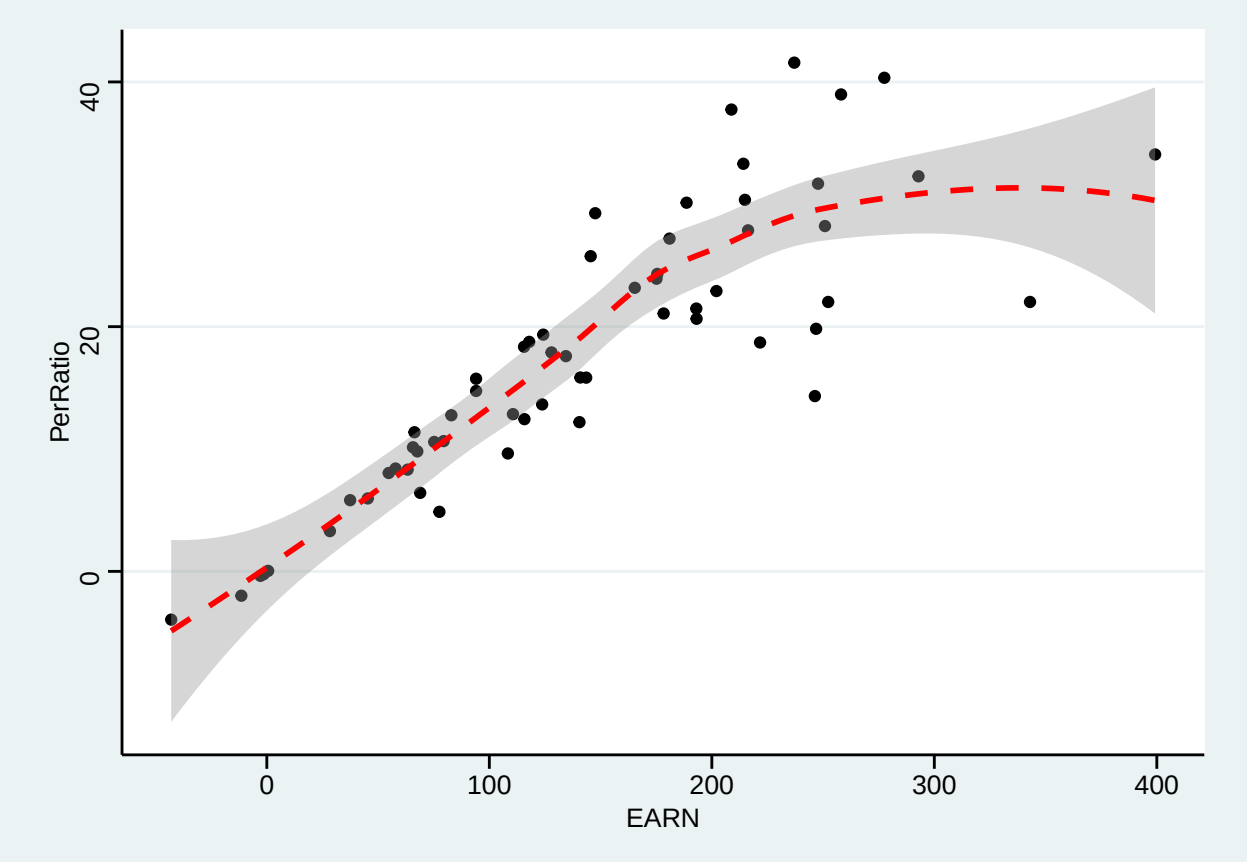
```
## Warning: package 'openxlsx' was built under R version 4.4.1
```

```
write.xlsx(CroqPaiFix, file = "CroqPaiFix")
```

Earn might be worth looking into further, but it is a part of our profit ratio, so it makes sense that it would have a high correlation.

```
ggplot(CroqPaiFix, aes(x = EARN, y = PerRatio)) +
  geom_point() +
  geom_smooth(method = loess, color = 'red', linetype = 'dashed')+
  theme_stata()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Create new variables by dividing EARN, P15, P25, P35, P45, P55, COMP, NCOMP, and NREST by total. These new variables are per capita. Examine correlations with the transformed. Do these correlations make sense to you?

First, we need to make our per capita statistics.

```
#Using the existing data frame to calculate the per capita stats
CroqPaiFix$EARN_Per <- CroqPaiFix$EARN / CroqPaiFix$total
CroqPaiFix$P15_Per <- CroqPaiFix$P15 / CroqPaiFix$total
CroqPaiFix$P25_Per <- CroqPaiFix$P25 / CroqPaiFix$total
CroqPaiFix$P35_Per <- CroqPaiFix$P35 / CroqPaiFix$total
CroqPaiFix$P45_Per <- CroqPaiFix$P45 / CroqPaiFix$total
CroqPaiFix$P55_Per <- CroqPaiFix$P55 / CroqPaiFix$total
CroqPaiFix$COMP_Per <- CroqPaiFix$COMP / CroqPaiFix$total
CroqPaiFix$NCOMP_Per <- CroqPaiFix$NCOMP / CroqPaiFix$total
CroqPaiFix$NREST_Per <- CroqPaiFix$NREST / CroqPaiFix$total

#Correlation calculation with just the per captia statistics into an independent data frame for better
cor(CroqPaiFix[19:27], CroqPaiFix[ , 18])
```

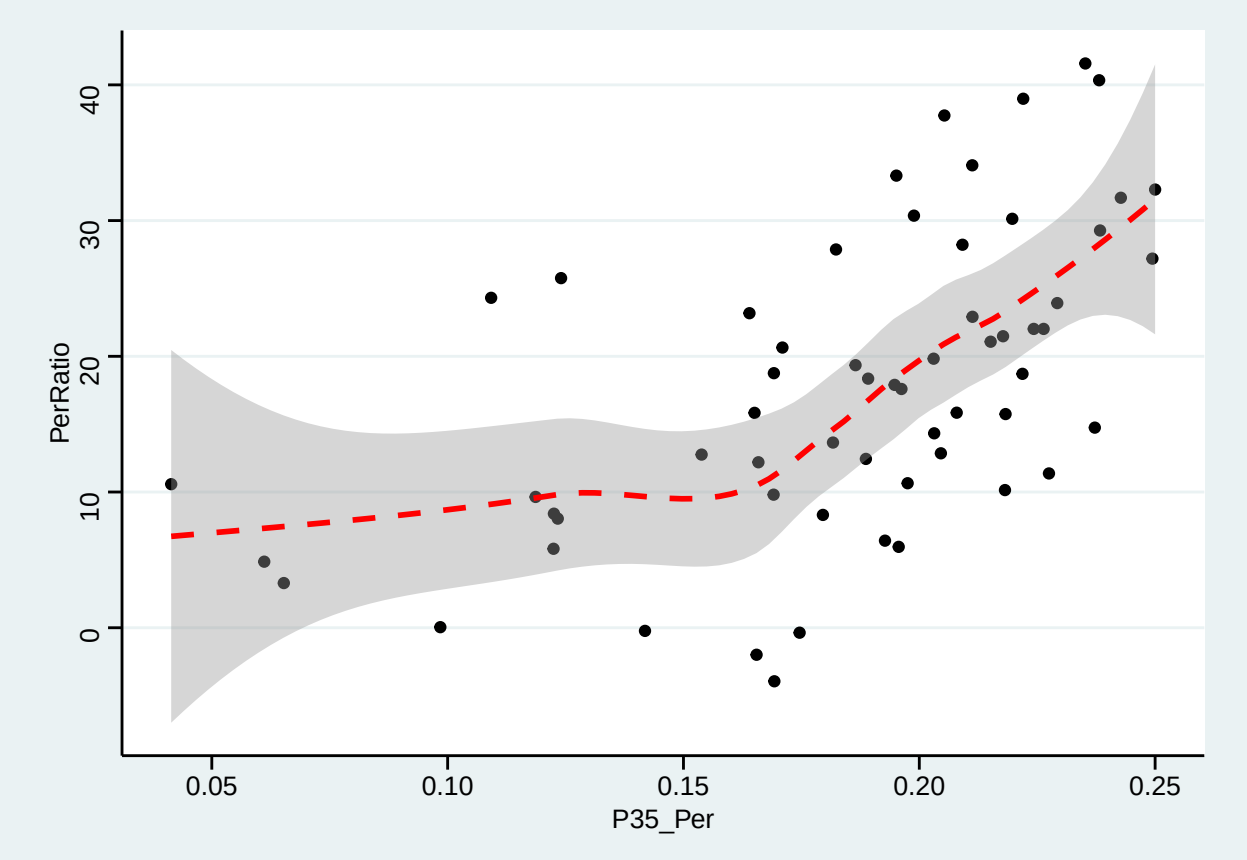
```
##                PerRatio
## EARN_Per    0.674042245
## P15_Per     0.494106937
## P25_Per    -0.166842585
```

```
## P35_Per    0.556088236
## P45_Per    0.482553240
## P55_Per   -0.337538761
## COMP_Per  -0.227433994
## NCOMP_Per -0.137536496
## NREST_Per -0.009835339
```

The highest per capita value is P35_Per, but it is at .556, which may not be high enough for what we need.

```
ggplot(CroqPaiFix, aes(x = P35_Per, y = PerRatio)) +
  geom_point() +
  geom_smooth(method = loess, color = 'red', linetype = 'dashed')+
  theme_stata()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



If you have more ideas for a better model, please feel free to create new variables. Examine them visually before using them.

2-1. Building Models (Part a)

Consider all observations in CroqPainFix.

Michel's first model is shown in Table 6.27 on page 305 with some concerns. Try to improve the model that does not violate any of the basic assumptions of regression but has good

predictive power. In short, you should carefully choose the explanatory variables. What are the explanatory variables of your choice?

THIS IS NOT ACCURATE

Based on Page 305, we can deduce that Michel likely used the following variables for her model: Size Employee Total P15 P25 P35 P45 P55 INC COMP NCOMP NREST PRICE CLI

We can recreate that model using R.

```
Michels_Model <- lm(PerRatio ~ SIZE + EMPL + total + P15 + P25 + P35 + P45 + P55 + INC + COMP + NCOMP +

summary(Michels_Model)
```

```
##
## Call:
## lm(formula = PerRatio ~ SIZE + EMPL + total + P15 + P25 + P35 +
##     P45 + P55 + INC + COMP + NCOMP + NREST + PRICE, data = CroqPaiFix)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.1798  -2.9201   0.5536   3.7894   8.9444
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -27.121499   8.591074  -3.157 0.002813 **
## SIZE          0.013923   0.014316   0.973 0.335859
## EMPL         -0.057751   0.211587  -0.273 0.786120
## total        -0.001473   0.001837  -0.802 0.426751
## P15           0.007428   0.004018   1.848 0.070964 .
## P25           0.001751   0.001940   0.902 0.371536
## P35           0.003040   0.003179   0.956 0.343925
## P45          -0.001099   0.004701  -0.234 0.816157
## P55           0.001706   0.002038   0.837 0.406973
## INC           1.235197   0.237541   5.200 4.48e-06 ***
## COMP         -0.044634   0.330090  -0.135 0.893031
## NCOMP        -0.049979   0.225868  -0.221 0.825856
## NREST         0.137082   0.033542   4.087 0.000173 ***
## PRICE        -0.775961   0.134279  -5.779 6.21e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.522 on 46 degrees of freedom
## Multiple R-squared:  0.8062, Adjusted R-squared:  0.7514
## F-statistic: 14.72 on 13 and 46 DF,  p-value: 2.9e-12
```

```
anova(Michels_Model)
```

```
## Analysis of Variance Table
##
## Response: PerRatio
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## SIZE       1   11.22   11.22  0.3680 0.5470933
## EMPL       1  106.28  106.28  3.4855 0.0682880 .
```

```
## total      1 2389.79 2389.79 78.3756 1.690e-11 ***
## P15        1  756.47  756.47 24.8090 9.373e-06 ***
## P25        1   74.87   74.87  2.4555 0.1239676
## P35        1   39.73   39.73  1.3031 0.2595567
## P45        1   44.87   44.87  1.4716 0.2312798
## P55        1    1.68    1.68  0.0551 0.8155264
## INC        1  719.62  719.62 23.6008 1.413e-05 ***
## COMP       1  125.99  125.99  4.1319 0.0478766 *
## NCOMP      1   24.12   24.12  0.7910 0.3784153
## NREST      1  520.52  520.52 17.0709 0.0001505 ***
## PRICE      1 1018.23 1018.23 33.3938 6.207e-07 ***
## Residuals 46 1402.61   30.49
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our model will attempt to be a bit nimbler than what Michel's was.

```
#installing car so I can use it to run vif tests
library(car)
```

```
## Warning: package 'car' was built under R version 4.4.1
```

```
## Loading required package: carData
```

```
#Fantastic R^2 value, but we aren't given earn in the final data set, but earn is essentially self refe
Group_Model <- lm(PerRatio ~ total + P15 + P35 + P45_Per + P55 + INC + PRICE + CLI + EARN_Per + SIZE + N
summary(Group_Model)
```

```
##
## Call:
## lm(formula = PerRatio ~ total + P15 + P35 + P45_Per + P55 + INC +
##     PRICE + CLI + EARN_Per + SIZE + NREST + NCOMP_Per, data = CroqPaiFix)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8083 -2.0282  0.0693  1.9272  6.7273
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.929e+00  1.018e+01   0.877  0.38498
## total        3.237e-05  4.294e-04   0.075  0.94023
## P15          3.027e-03  2.340e-03   1.293  0.20220
## P35          2.145e-03  1.897e-03   1.130  0.26407
## P45_Per     -6.066e+01  2.803e+01  -2.165  0.03554 *
## P55         -1.426e-04  6.618e-04  -0.215  0.83033
## INC          4.976e-01  1.775e-01   2.803  0.00733 **
## PRICE       -4.915e-01  9.185e-02  -5.351 2.54e-06 ***
## CLI         -7.659e-02  6.439e-02  -1.190  0.24022
## EARN_Per     1.539e+03  1.927e+02   7.987 2.66e-10 ***
## SIZE        -4.748e-02  1.167e-02  -4.068  0.00018 ***
## NREST        1.671e-02  2.653e-02   0.630  0.53189
## NCOMP_Per   -2.332e+03  2.115e+03  -1.103  0.27580
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.557 on 47 degrees of freedom
## Multiple R-squared:  0.9178, Adjusted R-squared:  0.8968
## F-statistic: 43.73 on 12 and 47 DF,  p-value: < 2.2e-16
```

```
#Fantastic R^2 but it includes earn again...
Group_Model_2 <- lm(PerRatio ~ P15 + P35 + P45_Per + P55 + INC + PRICE + CLI + EARN_Per + SIZE + NREST +
summary(Group_Model_2)
```

```
##
## Call:
## lm(formula = PerRatio ~ P15 + P35 + P45_Per + P55 + INC + PRICE +
##     CLI + EARN_Per + SIZE + NREST + NCOMP_Per, data = CroqPaiFix)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8410 -2.0095  0.0619  1.9422  6.6682
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.101e+00  9.822e+00   0.927 0.358783
## P15          3.087e-03  2.177e-03   1.418 0.162692
## P35          2.223e-03  1.573e-03   1.413 0.164009
## P45_Per     -6.158e+01  2.501e+01  -2.462 0.017467 *
## P55         -1.292e-04  6.307e-04  -0.205 0.838607
## INC          4.978e-01  1.757e-01   2.834 0.006710 **
## PRICE       -4.912e-01  9.082e-02  -5.409 1.98e-06 ***
## CLI         -7.607e-02  6.336e-02  -1.201 0.235754
## EARN_Per     1.537e+03  1.884e+02   8.156 1.28e-10 ***
## SIZE        -4.738e-02  1.148e-02  -4.128 0.000145 ***
## NREST        1.677e-02  2.624e-02   0.639 0.525936
## NCOMP_Per   -2.321e+03  2.088e+03  -1.112 0.271837
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.52 on 48 degrees of freedom
## Multiple R-squared:  0.9178, Adjusted R-squared:  0.8989
## F-statistic: 48.71 on 11 and 48 DF,  p-value: < 2.2e-16
```

```
#Doesn't use Earn so we have a massive dip in our R^2 value, but it also uses employee which we aren't
Group_Model_3 <- lm(PerRatio ~ EMPL + P15 + P35 + P45_Per + P55 + INC + PRICE + CLI + SIZE + NREST + NC
summary(Group_Model_3)
```

```
##
## Call:
## lm(formula = PerRatio ~ EMPL + P15 + P35 + P45_Per + P55 + INC +
##     PRICE + CLI + SIZE + NREST + NCOMP_Per, data = CroqPaiFix)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.053  -2.552   1.279   3.361   9.003
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.780e+01  1.428e+01  -1.246 0.218721
## EMPL        -5.232e-02  2.067e-01  -0.253 0.801247
## P15          4.819e-03  3.357e-03   1.436 0.157575
## P35          1.362e-03  2.434e-03   0.560 0.578334
## P45_Per     -4.713e+01  3.853e+01  -1.223 0.227318
## P55         -4.176e-04  9.720e-04  -0.430 0.669357
## INC          1.260e+00  2.303e-01   5.471 1.59e-06 ***
## PRICE       -7.556e-01  1.311e-01  -5.761 5.79e-07 ***
## CLI         -3.857e-02  9.791e-02  -0.394 0.695422
## SIZE         1.134e-02  1.381e-02   0.821 0.415864
## NREST        1.318e-01  3.421e-02   3.854 0.000345 ***
## NCOMP_Per    4.167e+02  3.202e+03   0.130 0.896985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.434 on 48 degrees of freedom
## Multiple R-squared:  0.8041, Adjusted R-squared:  0.7592
## F-statistic: 17.91 on 11 and 48 DF,  p-value: 2.008e-13
```

```
#Solid R^2 value, however there is high multicollinearity between K and Size, as well as P15m P35 and P
Group_Model_4 <- lm(PerRatio ~ K + SIZE + P15 + P15_Per + P35_Per + P45_Per + P55_Per + NCOMP_Per + NRES
summary(Group_Model_4)
```

```
##
## Call:
## lm(formula = PerRatio ~ K + SIZE + P15 + P15_Per + P35_Per +
##     P45_Per + P55_Per + NCOMP_Per + NREST_Per + INC + CLI + PRICE,
##     data = CroqPaiFix)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.0801  -2.9467   0.7992   3.3798  10.2753
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.242e+01  1.589e+01  -1.411 0.164745
## K           -2.121e-02  1.435e-02  -1.478 0.146055
## SIZE         9.752e-02  5.582e-02   1.747 0.087189 .
## P15          6.447e-03  1.169e-03   5.516 1.44e-06 ***
## P15_Per     -4.263e+00  5.870e+01  -0.073 0.942416
## P35_Per      7.071e+01  5.479e+01   1.290 0.203192
## P45_Per     -1.015e+02  6.072e+01  -1.672 0.101176
## P55_Per     -8.758e+00  1.543e+01  -0.568 0.572884
## NCOMP_Per    8.046e+02  3.250e+03   0.248 0.805570
## NREST_Per    1.977e+03  5.001e+02   3.953 0.000259 ***
## INC          1.367e+00  2.460e-01   5.558 1.25e-06 ***
## CLI         -5.893e-02  9.687e-02  -0.608 0.545905
## PRICE       -1.937e-01  3.713e-01  -0.522 0.604419
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 5.41 on 47 degrees of freedom
## Multiple R-squared:  0.8099, Adjusted R-squared:  0.7614
## F-statistic: 16.69 on 12 and 47 DF,  p-value: 4.491e-13
```

```
car::vif(Group_Model_4)
```

```
##         K      SIZE       P15   P15_Per   P35_Per   P45_Per   P55_Per NCOMP_Per
## 29.501790 18.258487  3.897226  3.827040 13.473986 10.715492  2.831257  1.496899
## NREST_Per       INC       CLI     PRICE
##  1.600960  1.309045  1.302608  9.814470
```

```
#I tried switch all of the age per capita adjusted metrics into the raw population numbers. The r^2 did
Group_Model_5 <- lm(PerRatio ~ K + SIZE + P15 + P25 + P35 + P45 + P55 + NCOMP_Per + NREST_Per + INC + CL
summary(Group_Model_5)
```

```
##
## Call:
## lm(formula = PerRatio ~ K + SIZE + P15 + P25 + P35 + P45 + P55 +
##     NCOMP_Per + NREST_Per + INC + CLI + PRICE, data = CroqPaiFix)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.7263  -3.0156   0.8351   3.6081  10.5508
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.709e+01  1.414e+01  -1.915 0.061574 .
## K           -1.946e-02  1.470e-02  -1.324 0.192069
## SIZE         9.100e-02  5.736e-02   1.586 0.119363
## P15          6.449e-03  3.730e-03   1.729 0.090410 .
## P25          5.098e-04  6.637e-04   0.768 0.446325
## P35          2.868e-03  3.117e-03   0.920 0.362238
## P45         -3.297e-03  3.355e-03  -0.983 0.330825
## P55          1.401e-05  7.623e-04   0.018 0.985419
## NCOMP_Per    8.970e+01  3.197e+03   0.028 0.977737
## NREST_Per    1.767e+03  4.709e+02   3.753 0.000481 ***
## INC          1.350e+00  2.491e-01   5.418 2.02e-06 ***
## CLI         -8.262e-02  9.628e-02  -0.858 0.395191
## PRICE       -2.345e-01  3.854e-01  -0.608 0.545877
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.492 on 47 degrees of freedom
## Multiple R-squared:  0.8041, Adjusted R-squared:  0.7541
## F-statistic: 16.08 on 12 and 47 DF,  p-value: 8.808e-13
```

```
car::vif(Group_Model_5)
```

```
##         K      SIZE       P15       P25       P35       P45       P55 NCOMP_Per
## 30.042523 18.709134 38.521118  1.378033 55.610816 53.482469  2.205948  1.405455
## NREST_Per       INC       CLI     PRICE
##  1.377316  1.302067  1.248515 10.262526
```

```r
#The multicollinearity dropped like crazy, but the R^2 inst that great. Plus, P35_Per and P45_Per still
Group_Model_6 <- lm(PerRatio ~ K + total + P25_Per + P35_Per + P45_Per + P55_Per + NCOMP_Per + NREST_Pe
summary(Group_Model_6)
```

```
##
## Call:
## lm(formula = PerRatio ~ K + total + P25_Per + P35_Per + P45_Per +
##     P55_Per + NCOMP_Per + NREST_Per + INC + CLI + PRICE, data = CroqPaiFix)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7534  -3.4850   0.9106   3.3718  10.6652
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.266e+01  2.549e+01  -1.282 0.206131
## K            2.613e-03  3.719e-03   0.702 0.485774
## total        9.837e-04  2.230e-04   4.411 5.79e-05 ***
## P25_Per      2.428e+00  2.759e+01   0.088 0.930254
## P35_Per      1.050e+02  5.759e+01   1.823 0.074560 .
## P45_Per     -7.306e+01  7.229e+01  -1.011 0.317210
## P55_Per      4.224e-01  3.041e+01   0.014 0.988975
## NCOMP_Per    8.457e+02  3.593e+03   0.235 0.814933
## NREST_Per    2.002e+03  5.280e+02   3.793 0.000417 ***
## INC          1.319e+00  2.429e-01   5.430 1.83e-06 ***
## CLI         -5.929e-02  1.018e-01  -0.582 0.563142
## PRICE       -8.176e-01  1.670e-01  -4.895 1.15e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.685 on 48 degrees of freedom
## Multiple R-squared:  0.7856, Adjusted R-squared:  0.7365
## F-statistic: 15.99 on 11 and 48 DF,  p-value: 1.587e-12
```

```r
car::vif(Group_Model_6)
```

```
##         K     total   P25_Per   P35_Per   P45_Per   P55_Per NCOMP_Per NREST_Per
##  1.794466  3.343520  5.112896 13.480823 13.749386  9.963623  1.656314  1.615847
##       INC       CLI     PRICE
##  1.155926  1.303369  1.798433
```

```r
#This version actually ended up being a bit too critical. The good news is that none of the VIF scores
Group_Model_7 <- lm(PerRatio ~ K + total + P35_Per + P55_Per + NCOMP_Per + NREST_Per + INC + CLI + PRIC
summary(Group_Model_7)
```

```
##
## Call:
## lm(formula = PerRatio ~ K + total + P35_Per + P55_Per + NCOMP_Per +
##     NREST_Per + INC + CLI + PRICE, data = CroqPaiFix)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -12.5430  -2.9932   0.3571   4.1174  10.2909
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.129e+01  1.559e+01  -2.007 0.050164 .
## K            2.063e-03  3.678e-03   0.561 0.577477
## total        1.035e-03  1.990e-04   5.202 3.69e-06 ***
## P35_Per      4.530e+01  3.121e+01   1.451 0.152926
## P55_Per      3.178e+00  1.529e+01   0.208 0.836167
## NCOMP_Per    3.434e+02  3.358e+03   0.102 0.918959
## NREST_Per    1.781e+03  4.966e+02   3.586 0.000761 ***
## INC          1.318e+00  2.407e-01   5.475 1.41e-06 ***
## CLI         -9.664e-02  9.720e-02  -0.994 0.324876
## PRICE       -7.941e-01  1.641e-01  -4.839 1.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.665 on 50 degrees of freedom
## Multiple R-squared:  0.7783, Adjusted R-squared:  0.7384
## F-statistic:  19.5 on 9 and 50 DF,  p-value: 1.73e-13
```

```
car::vif(Group_Model_7)
```

```
##         K     total   P35_Per   P55_Per NCOMP_Per NREST_Per       INC       CLI
##  1.768136  2.682371  3.988620  2.536414  1.456932  1.439893  1.142524  1.196094
##     PRICE
##  1.748731
```

```
#This version of the model saw K swapped for SIZE. Since K is directly tied to the performance ratio I
#This version of the model also ended up being too critical
Group_Model_8 <- lm(PerRatio ~ SIZE + total + P35_Per + P55_Per + NCOMP_Per + NREST_Per + INC + CLI + P
summary(Group_Model_8)
```

```
##
## Call:
## lm(formula = PerRatio ~ SIZE + total + P35_Per + P55_Per + NCOMP_Per +
##     NREST_Per + INC + CLI + PRICE, data = CroqPaiFix)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1724  -3.0469   0.3154   3.8113  10.2820
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.179e+01  1.551e+01  -2.050 0.045596 *
## SIZE         1.245e-02  1.422e-02   0.876 0.385416
## total        1.033e-03  1.982e-04   5.214 3.54e-06 ***
## P35_Per      4.530e+01  3.106e+01   1.458 0.151001
## P55_Per      2.861e+00  1.519e+01   0.188 0.851399
## NCOMP_Per    2.136e+02  3.347e+03   0.064 0.949372
## NREST_Per    1.808e+03  4.948e+02   3.654 0.000618 ***
## INC          1.317e+00  2.348e-01   5.610 8.77e-07 ***
## CLI         -9.592e-02  9.677e-02  -0.991 0.326329
```

```
## PRICE       -7.462e-01  1.338e-01  -5.578 9.82e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.639 on 50 degrees of freedom
## Multiple R-squared:  0.7803, Adjusted R-squared:  0.7407
## F-statistic: 19.73 on 9 and 50 DF,  p-value: 1.395e-13
```

```
car::vif(Group_Model_8)
```

```
##      SIZE     total   P35_Per   P55_Per NCOMP_Per NREST_Per       INC       CLI
##  1.090808  2.683022  3.984820  2.527778  1.460774  1.442266  1.097632  1.196122
##     PRICE
##  1.172381
```

```
Group_Model_9 <- lm(PerRatio ~ SIZE + P35 + P55 + NCOMP + NREST + INC + CLI + PRICE, data = CroqPaiFix)
summary(Group_Model_9)
```

```
##
## Call:
## lm(formula = PerRatio ~ SIZE + P35 + P55 + NCOMP + NREST + INC +
##     CLI + PRICE, data = CroqPaiFix)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.8654  -3.3541   0.8657   3.3919   9.6893
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.215e+01  1.367e+01  -1.620 0.111371
## SIZE         7.212e-03  1.357e-02   0.532 0.597356
## P35          3.341e-03  6.026e-04   5.543 1.06e-06 ***
## P55          5.668e-04  7.194e-04   0.788 0.434438
## NCOMP       -5.311e-02  2.140e-01  -0.248 0.805030
## NREST        1.257e-01  3.375e-02   3.724 0.000492 ***
## INC          1.313e+00  2.286e-01   5.745 5.12e-07 ***
## CLI         -5.468e-02  9.593e-02  -0.570 0.571208
## PRICE       -7.791e-01  1.295e-01  -6.016 1.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.451 on 51 degrees of freedom
## Multiple R-squared:  0.7906, Adjusted R-squared:  0.7577
## F-statistic: 24.07 on 8 and 51 DF,  p-value: 8.561e-15
```

```
car::vif(Group_Model_9)
```

```
##     SIZE      P35      P55    NCOMP    NREST      INC      CLI    PRICE
## 1.062509 2.109709 1.994207 1.077345 1.156092 1.113156 1.258041 1.175938
```

When you run regressions, be sure to check multicollinearity. If you want, you can try different variable selection algorithms. However, be critical and make a decision about the final model to recommend for Croq'Pain.

2-2. Validation by Testing (Part b)

In 2-1, you built a model using all observations in CroqPainFix. Michel wants to validate its usefulness (or accuracy) with existing data. Split CroqPainFix into the following two data sets. CroqPainFix_50: The 50 stores opened up prior to 1994, i.e., STOR<=50.

```
CroqPaiFix_50 <- CroqPaiFix[1:50, ]
```

CroqPainFix_10: The 10 stores opened in the first half of 1994, i.e., STOR>50.

```
CroqPaiFix_51_60 <- CroqPaiFix[51:60, ]
```

Using CroqPainFix_50, re-estimate your model. That is, keep your explanatory variables from 2-1 and re-estimate their coefficients with the data set of 50 stores.

```
#The old best model :/
Group_Model_3.1 <- lm(PerRatio ~ K + SIZE + P15 + P15_Per + P35_Per + P45_Per + P55_Per + NCOMP_Per + NR
#Really good r^2 (81.79) but the multicollinearity was kind of crazy for any of the population values, c
summary(Group_Model_3.1)
```

```
##
## Call:
## lm(formula = PerRatio ~ K + SIZE + P15 + P15_Per + P35_Per +
##     P45_Per + P55_Per + NCOMP_Per + NREST_Per + INC + CLI + PRICE,
##     data = CroqPaiFix_50)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.957  -3.272   0.497   3.852   7.437
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.313e+01  1.696e+01  -1.364 0.180888
## K           -2.281e-02  1.543e-02  -1.478 0.147869
## SIZE         8.920e-02  5.951e-02   1.499 0.142408
## P15          6.347e-03  1.345e-03   4.718 3.36e-05 ***
## P15_Per     -3.116e+01  7.570e+01  -0.412 0.682945
## P35_Per      7.529e+01  8.056e+01   0.935 0.356069
## P45_Per     -9.802e+01  7.427e+01  -1.320 0.195014
## P55_Per     -1.889e+01  1.758e+01  -1.074 0.289581
## NCOMP_Per    1.654e+03  3.499e+03   0.473 0.639200
## NREST_Per    2.334e+03  5.845e+02   3.994 0.000297 ***
## INC          1.454e+00  2.786e-01   5.218 7.18e-06 ***
## CLI         -3.995e-02  1.092e-01  -0.366 0.716469
## PRICE       -4.871e-02  4.020e-01  -0.121 0.904214
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.4 on 37 degrees of freedom
## Multiple R-squared:  0.8179, Adjusted R-squared:  0.7589
## F-statistic: 13.85 on 12 and 37 DF,  p-value: 3.03e-10
```

```r
car::vif(Group_Model_3.1)
```

```
##         K      SIZE       P15   P15_Per   P35_Per   P45_Per   P55_Per NCOMP_Per
## 28.781840 15.642038  4.588203  5.432595 26.809762 14.760380  3.309618  1.648007
## NREST_Per       INC       CLI     PRICE
##  1.968332  1.488356  1.418634 10.736035
```

```r
Group_Model_7.1 <- lm(PerRatio ~ K + total + P35_Per + P55_Per + NCOMP_Per + NREST_Per + INC + CLI + PR
summary(Group_Model_7.1)
```

```
##
## Call:
## lm(formula = PerRatio ~ K + total + P35_Per + P55_Per + NCOMP_Per +
##     NREST_Per + INC + CLI + PRICE, data = CroqPaiFix_50)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.4667  -2.9331  -0.3465   3.8516   8.5808
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.485e+01  1.680e+01  -2.075 0.044479 *
## K           -1.642e-03  4.567e-03  -0.360 0.721041
## total        1.024e-03  2.070e-04   4.946 1.40e-05 ***
## P35_Per      3.519e+01  3.438e+01   1.024 0.312149
## P55_Per     -8.253e+00  1.734e+01  -0.476 0.636620
## NCOMP_Per    5.896e+02  3.476e+03   0.170 0.866143
## NREST_Per    2.116e+03  5.381e+02   3.932 0.000326 ***
## INC          1.369e+00  2.658e-01   5.152 7.29e-06 ***
## CLI         -5.040e-02  1.077e-01  -0.468 0.642432
## PRICE       -6.010e-01  1.798e-01  -3.343 0.001808 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.574 on 40 degrees of freedom
## Multiple R-squared:  0.7902, Adjusted R-squared:  0.7431
## F-statistic: 16.74 on 9 and 40 DF,  p-value: 5.549e-11
```

```r
car::vif(Group_Model_7.1)
```

```
##         K     total   P35_Per   P55_Per NCOMP_Per NREST_Per       INC       CLI
##  2.365421  2.707195  4.581789  3.018508  1.525651  1.565397  1.270904  1.296343
##     PRICE
##  2.014922
```

```r
#This was the interesting part of this model. The R^2 actually went up to 79.02, which while not 80 is
```

```r
Group_Model_8.1 <- lm(PerRatio ~ SIZE + total + P35_Per + P55_Per + NCOMP_Per + NREST_Per + INC + CLI +
summary(Group_Model_8.1)
```

```
##
```

```
## Call:
## lm(formula = PerRatio ~ SIZE + total + P35_Per + P55_Per + NCOMP_Per +
##     NREST_Per + INC + CLI + PRICE, data = CroqPaiFix_50)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1450  -3.0345  -0.1659   3.8315   8.7285
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.376e+01  1.668e+01  -2.024  0.04969 *
## SIZE        -1.808e-04  1.748e-02  -0.010  0.99180
## total        1.024e-03  2.073e-04   4.941 1.43e-05 ***
## P35_Per      3.714e+01  3.436e+01   1.081  0.28630
## P55_Per     -8.359e+00  1.736e+01  -0.481  0.63279
## NCOMP_Per    5.303e+02  3.482e+03   0.152  0.87970
## NREST_Per    2.146e+03  5.387e+02   3.983  0.00028 ***
## INC          1.331e+00  2.559e-01   5.203 6.18e-06 ***
## CLI         -5.792e-02  1.079e-01  -0.537  0.59435
## PRICE       -6.411e-01  1.412e-01  -4.539 5.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.583 on 40 degrees of freedom
## Multiple R-squared:  0.7896, Adjusted R-squared:  0.7422
## F-statistic: 16.68 on 9 and 40 DF,  p-value: 5.902e-11
```

```
car::vif(Group_Model_8.1)
```

```
##      SIZE     total   P35_Per   P55_Per NCOMP_Per NREST_Per       INC       CLI
##  1.262864  2.707242  4.562467  3.017752  1.525894  1.563929  1.174066  1.296007
##     PRICE
##  1.239416
```

```
#Similar to 7.1, this model saw a bit of a jump in the multicollinearity and the R^2 score went up a bi

Group_Model_9.1 <- lm(PerRatio ~ SIZE + P35 + P55 + NCOMP + NREST + INC + CLI + PRICE, data = CroqPaiFi
summary(Group_Model_9.1)
```

```
##
## Call:
## lm(formula = PerRatio ~ SIZE + P35 + P55 + NCOMP + NREST + INC +
##     CLI + PRICE, data = CroqPaiFix_50)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.171  -2.868   0.670   3.730   9.693
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.917e+01  1.514e+01  -1.926 0.061044 .
## SIZE        -6.509e-03  1.671e-02  -0.390 0.698815
## P35          3.350e-03  6.495e-04   5.158 6.73e-06 ***
```

```
## P55          3.070e-04  8.159e-04   0.376 0.708650
## NCOMP       -3.247e-03  2.227e-01  -0.015 0.988437
## NREST        1.412e-01  3.576e-02   3.949 0.000301 ***
## INC          1.345e+00  2.500e-01   5.381 3.26e-06 ***
## CLI         -2.424e-03  1.071e-01  -0.023 0.982064
## PRICE       -7.058e-01  1.377e-01  -5.127 7.45e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.442 on 41 degrees of freedom
## Multiple R-squared:  0.7951, Adjusted R-squared:  0.7551
## F-statistic: 19.88 on 8 and 41 DF,  p-value: 7.636e-12
```

```
car::vif(Group_Model_9.1)
```

```
##     SIZE      P35      P55    NCOMP    NREST      INC      CLI    PRICE
## 1.213195 2.188818 2.023041 1.079564 1.122894 1.179407 1.345073 1.239358
```

With this newly estimated model, predict the performances of the 10 stores in CroqPainFix_10. Using Croq'Pain's performance ratio target of 26%, which of the ten stores would you have opened in 1994 according to your model?

```
#time to build our first function! This function reads in all the data we need for our model, multiplie
model_function <- function(STOR, SIZE, P35, P55, NCOMP, NREST, INC, CLI, PRICE){
  PR <- (-29.17 + (as.numeric(SIZE) * -0.006509) + as.numeric(P35) * 0.003350 + as.numeric(P55) * 0.000
  Store <- STOR

  PR <- round(PR, 2)

  return(paste(Store, "has a Performance Ratio of", PR, "%"))
}

#applying the function to the data set
result <- apply(CroqPaiFix_51_60, MARGIN = 1, FUN = function(row) {
  model_function(
    STOR = row[1],
    SIZE = row[3],
    P35 = row[9],
    P55 = row[11],
    NCOMP = row[14],
    NREST = row[15],
    INC = row[12],
    CLI = row[17],
    PRICE = row[16]
  )
})

result
```

```
##  [1] "51 has a Performance Ratio of 16.46 %"
##  [2] "52 has a Performance Ratio of 17.48 %"
##  [3] "53 has a Performance Ratio of 7.19 %"
##  [4] "54 has a Performance Ratio of 11.8 %"
```

```
##  [5] "55 has a Performance Ratio of 8.01 %"
##  [6] "56 has a Performance Ratio of 2.39 %"
##  [7] "57 has a Performance Ratio of 30.73 %"
##  [8] "58 has a Performance Ratio of 9.15 %"
##  [9] "59 has a Performance Ratio of 7.83 %"
## [10] "60 has a Performance Ratio of 31.01 %"
```

Based on our model, we determined that only two stores would meet the performance threshold we need for the store to be deemed "successful": Store 57 (30.73) Store 60 (31.01)

3. Prediction (Part c)

As you validated the model (i.e., a selection of explanatory variables), let's use the most complete estimates using CroqPainFix. With this complete model, you need to recommend which new locations Croq'Pain should open its stores among 10 locations in Table 6.28 or CroqPain_new.rda. Which locations would you recommend?

One problem that we ran into was that our model uses the EARN Per Capita value to forecast the Profitability Ratio. The Croq_Pain database was loaded in without an economic forecast, which means that we either had to modify our model, or create an economic forecasting model of our own, run that through the new data to create and EARN and then try to predict which Croq Pains should be opened. So we changed models and went to the current iteration

After a lot of tests, including one where the model was much too critical, we finally have a working model that has:

A strong R^2 value Low multicollinearity Isn't too critical (doesn't recommend 0 new openings) Isn't too positive (doesn't recommend 1+ openings with a high performance ratio)

```
final_result <- apply(CroqPain_new, MARGIN = 1, FUN = function(row) {
  model_function(
    STOR = row[1],
    SIZE = row[3],
    P35 = row[8],
    P55 = row[10],
    NCOMP = row[13],
    NREST = row[14],
    INC = row[11],
    CLI = row[16],
    PRICE = row[15]
  )
})

final_result
```

```
##  [1] "Calais has a Performance Ratio of 10.9 %"
##  [2] "Montchanin has a Performance Ratio of 13.88 %"
##  [3] "Aubusson has a Performance Ratio of 3.61 %"
##  [4] "Toulouse has a Performance Ratio of 27.6 %"
##  [5] "Torcy has a Performance Ratio of 5.17 %"
##  [6] "Marseilles-1 has a Performance Ratio of 6.5 %"
##  [7] "Marseilles-2 has a Performance Ratio of -8.26 %"
##  [8] "Clermont has a Performance Ratio of 9.1 %"
##  [9] "Montpellier has a Performance Ratio of 18.55 %"
## [10] "Dijon has a Performance Ratio of 17.28 %"
```

The model predicted that Toulouse would have a Performance Ratio of 27.6%, slightly above the required threshold of 26%.

4. Report your recommendation (Part d)

Prepare an executive summary containing your recommendations as to which store they should open according to your regression analysis.