Ivey BA R Workshop

Introduction to the Tidyverse

Hayden MacDonald

2019-03-01

Packages

```
library(tidyverse)
library(readxl)
library(GGally)
library(naniar)
library(broom)
```

Import

```
pain <- read_xlsx("CroqPainData_Feb14.xlsx",
                  sheet = 1,
                  range = "A1:Q73")
```

Power Example

```
hist_fun = function(x) {
     ggplot(pain_ex, aes_string(x = x) ) +
          geom_histogram()
}
```

```
pain_ex <- pain %>%
  mutate(EMPL = as.numeric(EMPL))
```

```
## Warning in evalq(as.numeric(EMPL), <environment>): NAs introduced by
## coercion
```

```
pain_vars <- names(pain_ex)[2:17]

pain_hists <- map(pain_vars, hist_fun)
```

Transform

Problematic Rows

```
pain[c(51,62),]
```

```
## # A tibble: 2 x 17
##   STOR     EARN     K  SIZE EMPL  total   P15   P25   P35   P45   P55   INC
##   <chr>   <dbl> <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Stores~    NA    NA    NA <NA>     NA    NA    NA    NA    NA    NA    NA
## 2 Ten ne~    NA    NA    NA <NA>     NA    NA    NA    NA    NA    NA    NA
## # ... with 5 more variables: COMP <dbl>, NCOMP <dbl>, NREST <dbl>,
## #   PRICE <dbl>, CLI <dbl>
```

Clean Data

```
pain <- pain %>%
  filter(rownames(pain) =≠ c(51,62)) %>%
  mutate(STOR = as.numeric(STOR),
         EMPL = as.numeric(EMPL)) %>%
  mutate(STOR = seq(1, 70, by = 1))
```

```
## Warning in evalq(as.numeric(STOR), <environment>): NAs introduced by
## coercion
```

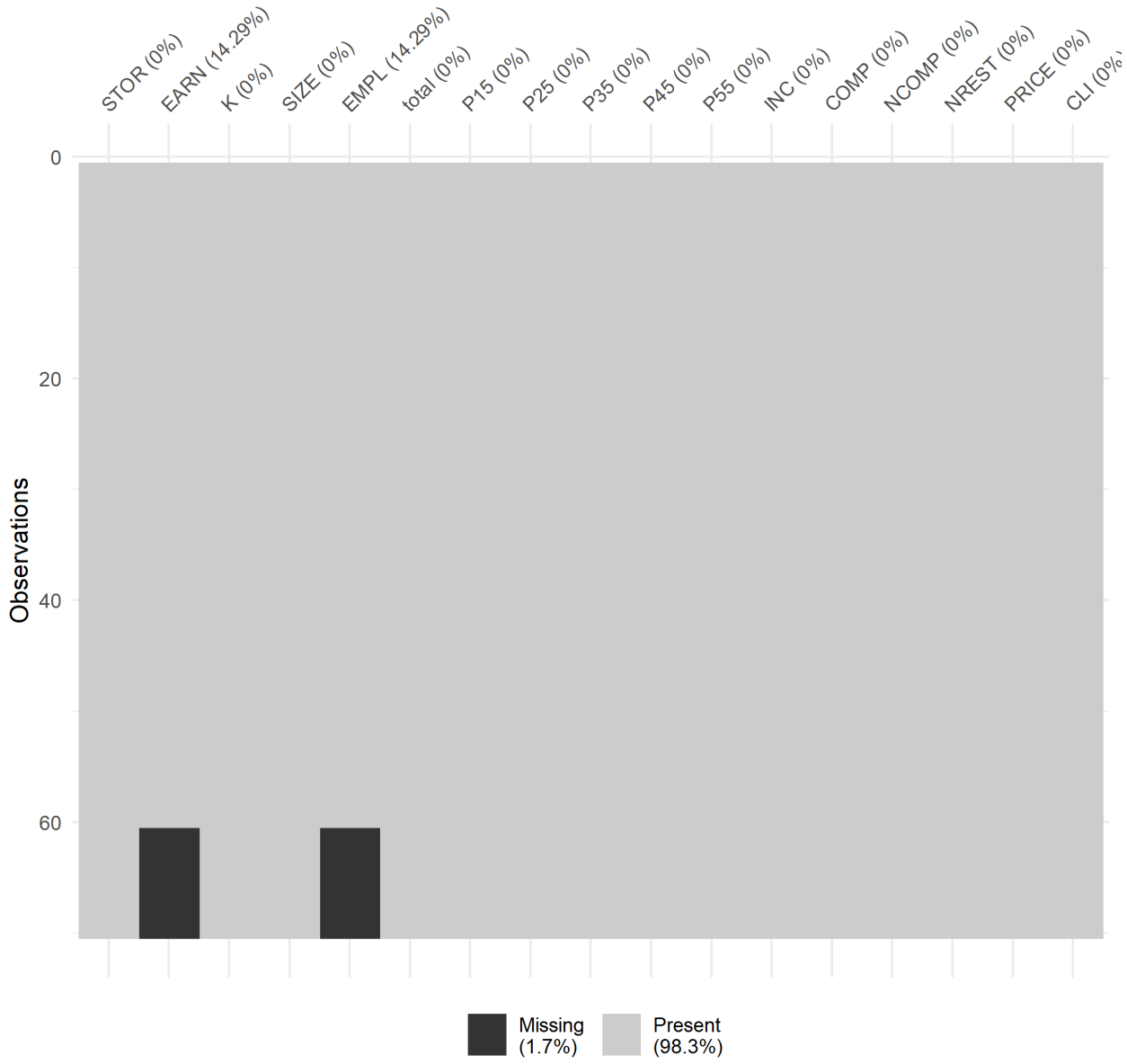
```
## Warning in evalq(as.numeric(EMPL), <environment>): NAs introduced by
## coercion
```

```
pain
```

```
## # A tibble: 70 x 17
##     STOR    EARN     K  SIZE  EMPL total   P15   P25   P35   P45   P55
##    <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1     1   28.3   861.   129    14  8580   980  1280   560  1000  3100
##  2     2   -1.46  630.    91    12  8460  1290   720  1200  1490  3100
##  3     3   68.9  1074.   140    13 19250  2940  2490  3710  4030  5270
##  4     4  202.    882.   184     7 20920  3570  4930  4420  4300  2960
##  5     5  116.    931.   144    14 11660  1700  1140  2200  2140  2630
##  6     6  222.   1185.   160    11 25780  4640  3150  5720  5330  5920
##  7     7  293.    907.    94     5 19000  3600  2330  4750  4970  3030
##  8     8  134.    764.   100     8 18500  3450  2560  3630  3520  4800
##  9     9   37.4   643.    85    14 14210  1930  4280  1740  2060  2960
## 10    10  181.    666.    92     6 17440  3520  1780  4350  4020  3470
## # ... with 60 more rows, and 6 more variables: INC <dbl>, COMP <dbl>,
## #   NCOMP <dbl>, NREST <dbl>, PRICE <dbl>, CLI <dbl>
```

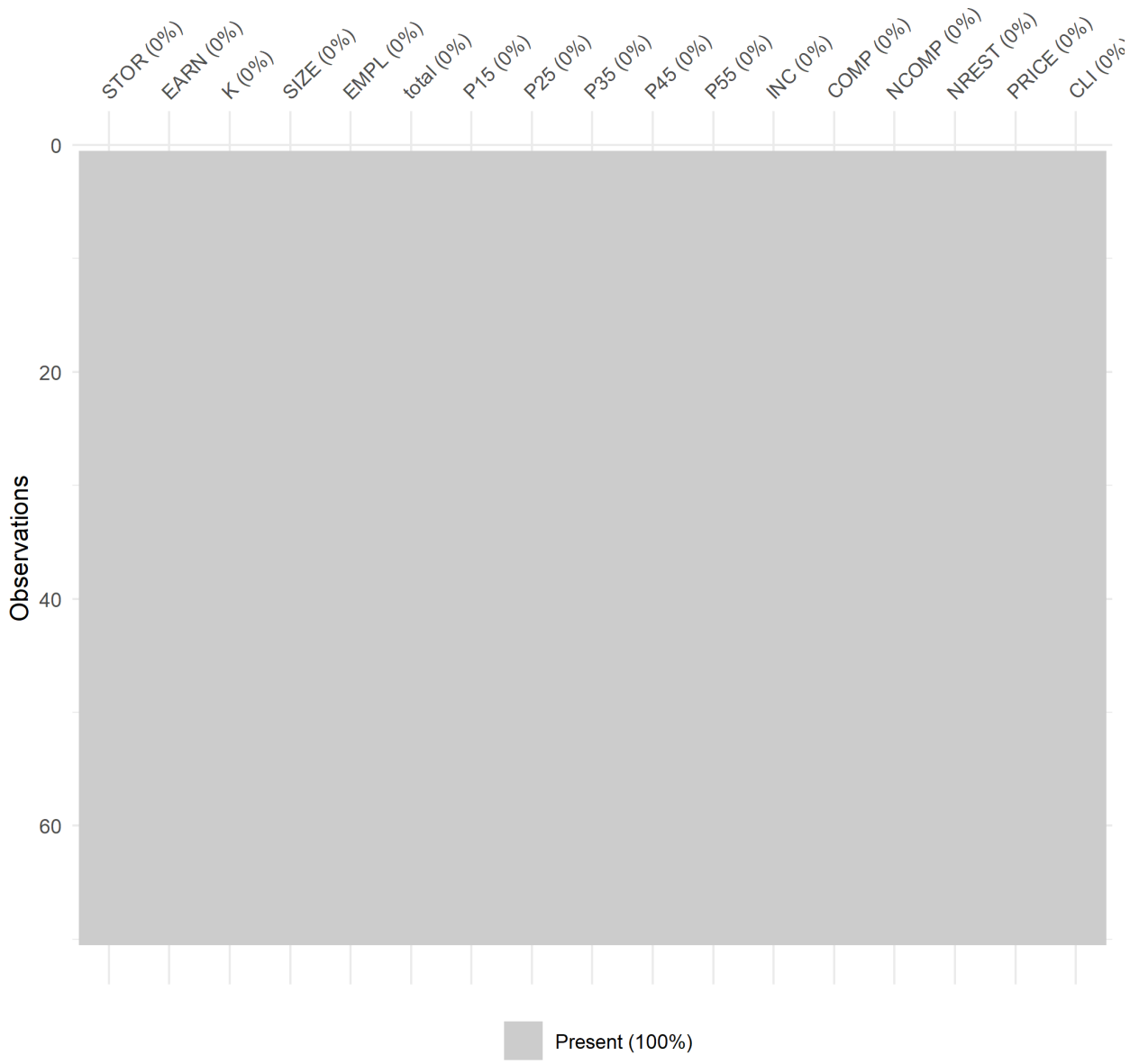
Imputation & Visualizing Missingness

```
vis_miss(pain)
```



```r
pain <- pain %>%
  impute_mean_all()

vis_miss(pain)
```

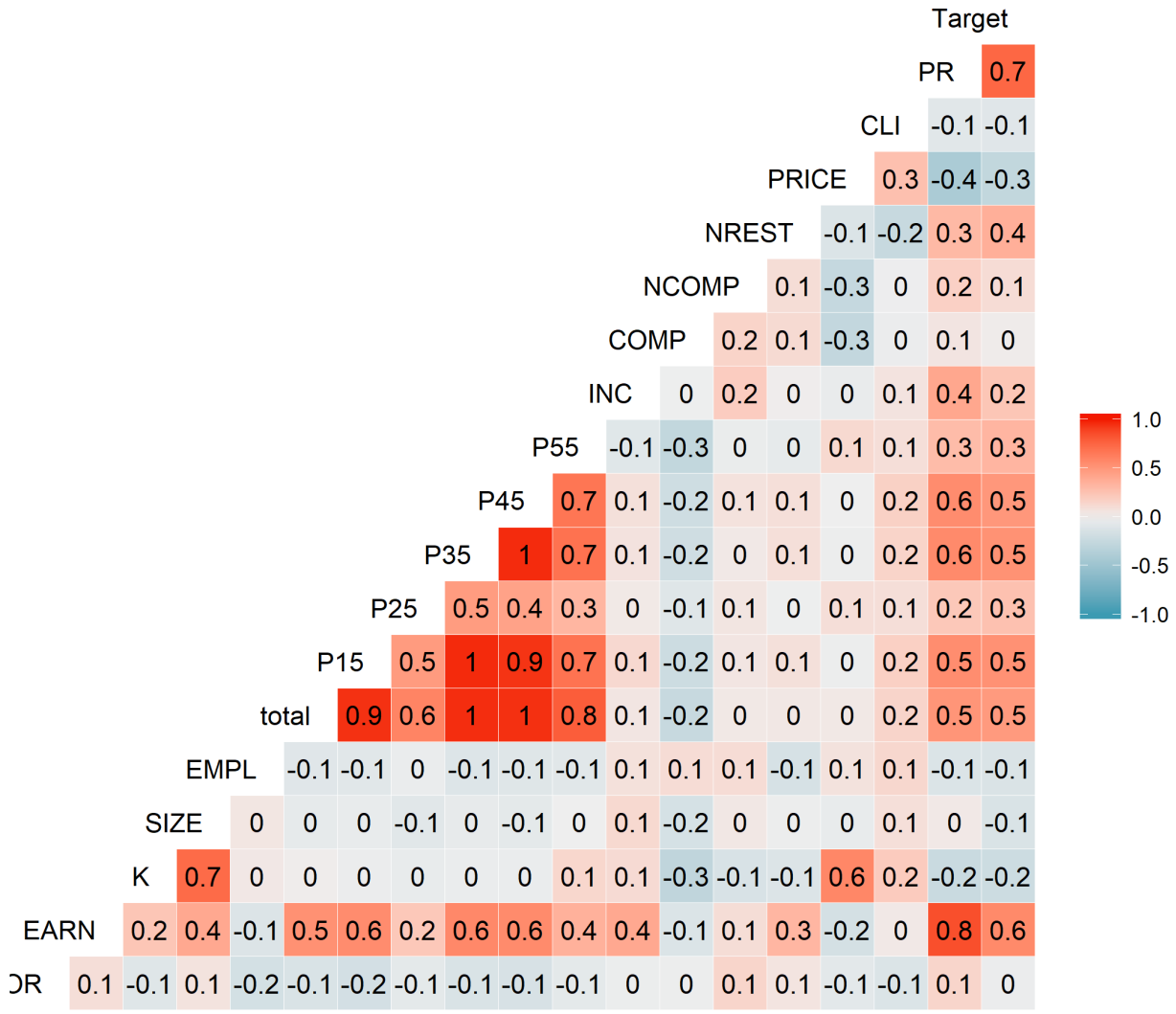


Create new variables

```
pain <- pain %>%
          mutate(PR = EARN / K,
                 Target = case_when(PR ≥ 0.26 ~ 1, PR < 0.26 ~ 0))
```

Visualize

```
ggcorr(pain, label = TRUE, hjust = 1)
```



Training and Testing Sets

```
pain_train <- pain %>%
                filter(STOR <= 60)

pain_test <- pain %>%
                filter(STOR > 60)
```

Model

```
# Extract variable names

str_c(names(pain_train), collapse = " + ")
```

```
## [1] "STOR + EARN + K + SIZE + EMPL + total + P15 + P25 + P35 + P45 + P55 + INC + COMP + NCOMP + NREST + PRIC
```

```
# Preliminary model with all variables except STOR and EARN
mod <- glm(Target ~ STOR + EARN + K + SIZE + EMPL + total + P15 + P25 + P35 + P45 +
    P55 + INC + COMP + NCOMP + NREST + PRICE + CLI + PR, family = binomial, data = pain_train)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
mod <- glm(Target ~ SIZE + EMPL + P25 + P35 + P55 + INC + COMP + NCOMP + PRICE +
    CLI, family = binomial, data = pain_train)


summary(mod)
```

```
##
## Call:
## glm(formula = Target ~ SIZE + EMPL + P25 + P35 + P55 + INC +
##     COMP + NCOMP + PRICE + CLI, family = binomial, data = pain_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.55204  -0.30026  -0.07469  -0.00211   2.37319
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.1405186  8.5156744  -0.721   0.4709
## SIZE        -0.0038828  0.0115123  -0.337   0.7359
## EMPL        -0.2834132  0.1854398  -1.528   0.1264
## P25          0.0001915  0.0004443   0.431   0.6664
## P35          0.0019975  0.0008124   2.459   0.0139 *
## P55         -0.0004488  0.0005735  -0.782   0.4339
## INC          0.4246620  0.2292629   1.852   0.0640 .
## COMP        -0.0395400  0.2419572  -0.163   0.8702
## NCOMP        0.1289480  0.1687441   0.764   0.4448
## PRICE       -0.3436066  0.1654900  -2.076   0.0379 *
## CLI         -0.0613541  0.0670201  -0.915   0.3600
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 65.193  on 59  degrees of freedom
## Residual deviance: 27.234  on 49  degrees of freedom
## AIC: 49.234
##
## Number of Fisher Scoring iterations: 8
```

```
tidy(mod) %>%
  arrange(desc(p.value))
```

```
## # A tibble: 11 x 5
##    term         estimate std.error statistic p.value
##    <chr>           <dbl>     <dbl>     <dbl>   <dbl>
##  1 COMP        -0.0395    0.242       -0.163  0.870
##  2 SIZE        -0.00388   0.0115      -0.337  0.736
##  3 P25          0.000192  0.000444     0.431  0.666
##  4 (Intercept) -6.14      8.52        -0.721  0.471
##  5 NCOMP        0.129     0.169        0.764  0.445
##  6 P55         -0.000449  0.000574    -0.782  0.434
##  7 CLI         -0.0614    0.0670      -0.915  0.360
##  8 EMPL        -0.283     0.185       -1.53   0.126
##  9 INC          0.425     0.229        1.85   0.0640
## 10 PRICE       -0.344     0.165       -2.08   0.0379
## 11 P35          0.00200   0.000812     2.46   0.0139
```

Final Model

```
final_mod <- glm(Target ~ P35 + PRICE, family = binomial, data = pain_train)

tidy(final_mod) %>%
  arrange(desc(p.value))
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept) -3.04     2.07        -1.47 0.142
## 2 PRICE       -0.246    0.0999      -2.46 0.0139
## 3 P35          0.00130  0.000398     3.25 0.00115
```

```
glance(final_mod)
```

```
## # A tibble: 1 x 7
##   null.deviance df.null logLik   AIC   BIC deviance df.residual
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int>
## 1          65.2      59  -18.3  42.6  48.9     36.6          57
```

Standardized Residual Plot

```
augment(mod) %>%
  ggplot() +
  geom_point(aes(x = .fitted, y = .std.resid))
```