Lab 7

Adam Shen

November 4, 2020

Packages

```r
library(MASS)
library(leaps)
library(ggplot2)
library(broom)
library(dplyr)
theme_set(theme_bw())
```

Steel data

```r
steel <- read.table("./employees.txt", header=TRUE)
```

Fit a OLS model

```r
model_ols <- lm(Present ~ Past, data=steel)
summary(model_ols)
```

```
##
## Call:
## lm(formula = Present ~ Past, data = steel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.360  -1.964   0.101   5.156  39.425
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.31386    9.99747  -0.031   0.9757
## Past         0.40038    0.08485   4.719   0.0015 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.81 on 8 degrees of freedom
## Multiple R-squared:  0.7357, Adjusted R-squared:  0.7026
## F-statistic: 22.27 on 1 and 8 DF,  p-value: 0.001505
```

Obtain additional model information

```r
model_ols_aug <- augment(model_ols) %>%
  mutate(ext.res = rstudent(model_ols)) %>%
  print()
```

```
## # A tibble: 10 x 9
##    Present  Past .fitted  .resid .std.resid  .hat .sigma   .cooksd ext.res
##      <int> <int>   <dbl>   <dbl>      <dbl> <dbl>  <dbl>     <dbl>   <dbl>
## 1      132   232 92.6    39.4        2.53   0.441   9.87 2.54        5.34
## 2       50    96 38.1    11.9        0.602  0.101  21.7  0.0203      0.576
## 3       43   158 62.9   -19.9       -1.06   0.180  20.6  0.123      -1.07
## 4       41   194 77.4   -36.4       -2.07   0.284  15.2  0.847      -2.83
## 5       33    89 35.3    -2.32      -0.118  0.100  22.2  0.000767  -0.110
## 6       25    64 25.3    -0.311     -0.0158 0.110  22.2  0.0000155 -0.0148
## 7       16    25  9.70    6.30       0.332  0.167  22.1  0.0111      0.313
## 8        8    23  8.89   -0.895     -0.0473 0.172  22.2  0.000232  -0.0442
## 9        3     4  1.29    1.71       0.0931 0.219  22.2  0.00122     0.0872
## 10       1     2  0.487   0.513      0.0280 0.225  22.2  0.000114    0.0262
```

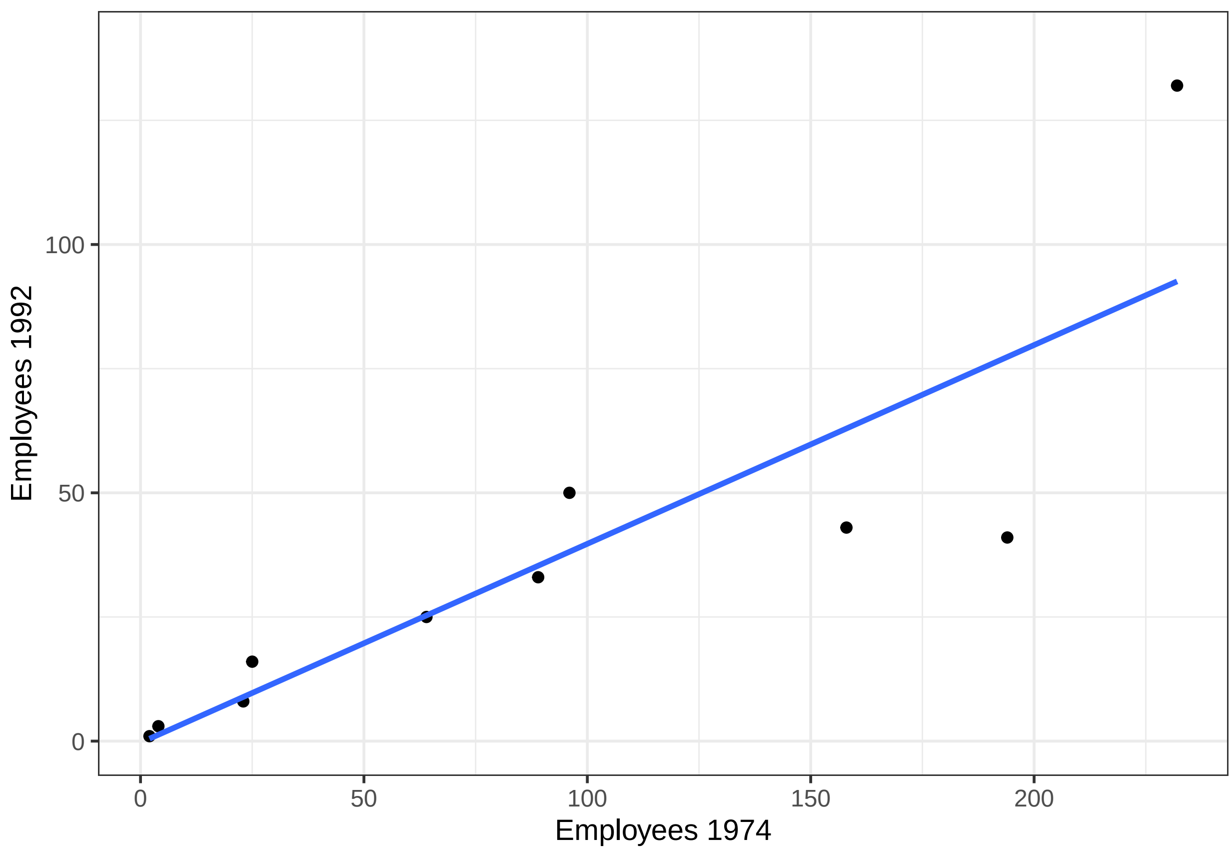
Recall that the first argument to `dplyr::mutate` is a data frame. The `%>%` is known as a pipe which passes the previous argument forward. Without using the pipe, our code would look something like:

```
model_ols_aug <- mutate(augment(model_ols),
                        ext.res = rstudent(model_ols))
```

Using the pipe often makes your code easier to read as it makes the sequential nature more apparent.

Visualize the OLS fit

```
ggplot(model_ols_aug, aes(x=Past))+
  geom_point(aes(y=Present))+
  geom_line(aes(y=.fitted), colour="#3366FF", size=1)+
  labs(x="Employees 1974", y="Employees 1992")+
  coord_cartesian(ylim=c(0, 140))
```



Fit a RLM

We will fit a robust linear regression model using the psi function as proposed by Huber, with a tuning constant of 2. Since this model is fit in an iterative manner, we also specify that the maximum number of iterations will be 40.

```
model_rlm <- rlm(Present ~ Past, psi=psi.huber, k=2, maxit=40, data=steel)
summary(model_rlm)
```

```
##
## Call: rlm(formula = Present ~ Past, data = steel, psi = psi.huber,
##     k = 2, maxit = 40)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.5075  -2.9071  -0.2361   3.7796  54.3140
##
## Coefficients:
##             Value  Std. Error t value
## (Intercept) 3.3335 6.2164     0.5363
## Past        0.3205 0.0528     6.0745
##
## Residual standard error: 5.655 on 8 degrees of freedom
```

Obtain additional model information

```
model_rlm_aug <- augment(model_rlm) %>%
  mutate(hweights = model_rlm$w) %>%
  relocate(hweights, .before=.fitted) %>%
  print()
```

```
## # A tibble: 10 x 7
##    Present  Past hweights .fitted .resid  .hat .sigma
##      <int> <int>    <dbl>   <dbl>  <dbl> <dbl>  <dbl>
##  1     132   232    0.208   77.7   54.3  0.188   6.93
##  2      50    96    0.711   34.1   15.9  0.101  22.9
##  3      43   158    1       54.0  -11.0  0.354  23.2
##  4      41   194    0.461   65.5  -24.5  0.268  21.2
##  5      33    89    1       31.9    1.14 0.132  23.8
##  6      25    64    1       23.8    1.16 0.120  23.8
##  7      16    25    1       11.3    4.65 0.173  23.7
##  8       8    23    1       10.7   -2.70 0.179  23.8
##  9       3     4    1        4.62  -1.62 0.239  23.8
## 10       1     2    1        3.97  -2.97 0.247  23.8
```

Comparison of models

After fitting the OLS model in *Module 7.3*, it was suggested that observations 1 and 4 were influential and outliers. Let us indicate this in our comparative scatterplot.

Data prep

To the `steel` data, we create a new variable called `Problem`. This variable is an indicator of whether a point is one of the previously deemed problematic points by comparing the rownames of the data set to the strings `"1"` and `"4"`.

```
steel <- steel %>%
  mutate(Problem = factor(ifelse(rownames(steel) %in% c("1", "4"), "Yes", "No")))
```

Intermediate results:

```
factor(ifelse(rownames(steel) %in% c("1", "4"), "Yes", "No"))
```

```
##  [1] Yes No  No  Yes No  No  No  No  No  No
## Levels: No Yes
```

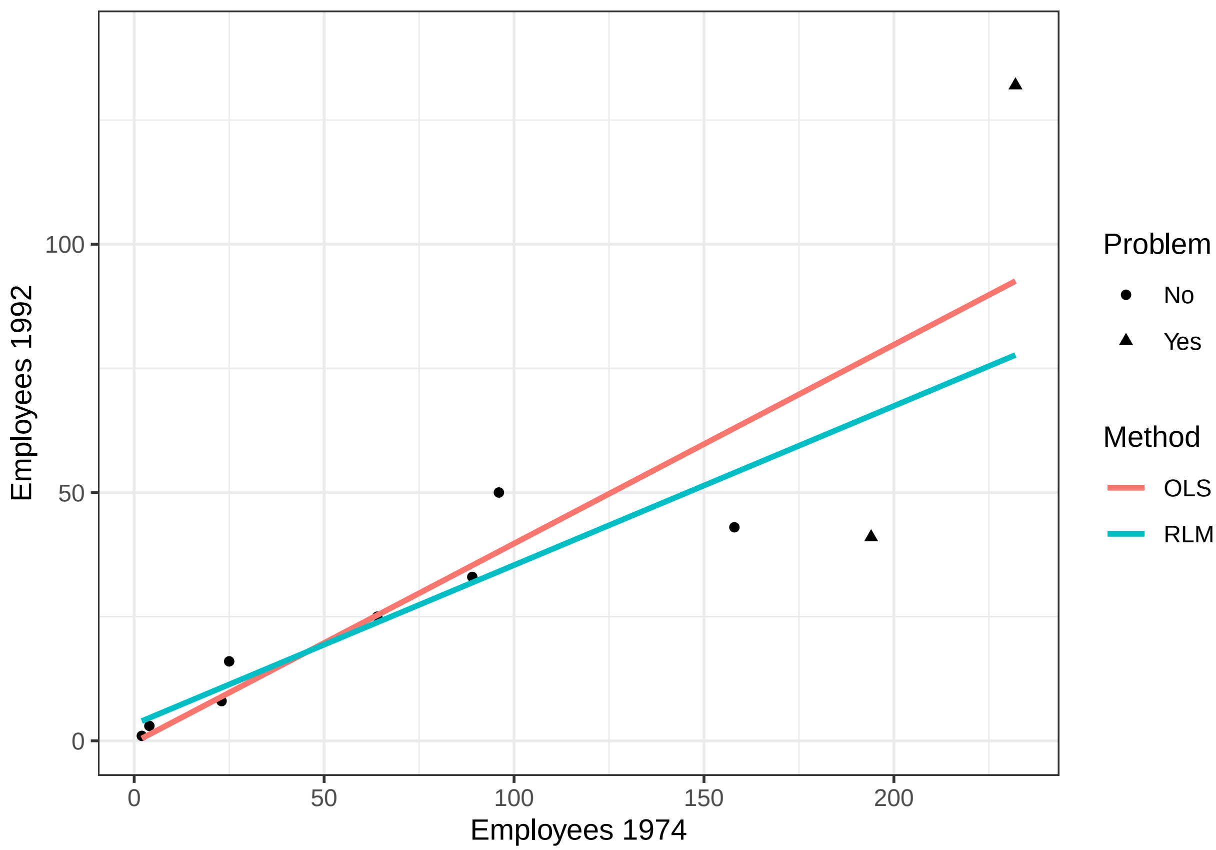
As in previous labs, we also create a new column in each of the augmented model data frames to indicate the method used.

```
model_ols_aug <- model_ols_aug %>%
  mutate(Method = "OLS")

model_rlm_aug <- model_rlm_aug %>%
  mutate(Method = "RLM")
```

Visualization

```
ggplot(data=NULL, aes(x=Past))+
  geom_point(data=steel, aes(y=Present, shape=Problem))+
  geom_line(data=model_ols_aug, aes(y=.fitted, colour=Method), size=1)+
  geom_line(data=model_rlm_aug, aes(y=.fitted, colour=Method), size=1)+
  labs(x="Employees 1974", y="Employees 1992")+
  coord_cartesian(ylim=c(0, 140))
```



Adjusted Steel data

We repeat everything but using the adjusted steel data, where the present number of employees for Germany was adjusted to account for the fact that the 1974 observation only included West Germany, while the 1992 observation included both West and East Germany.

```
steeladj <- read.table("./employeesadj.txt", header=TRUE)
```

Fit a OLS model

```
model_ols <- lm(Present ~ Past, data=steeladj)
summary(model_ols)
```

```
##
## Call:
## lm(formula = Present ~ Past, data = steeladj)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.5365  -2.4754  -0.3187   3.8520  23.7837
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.80261    6.99220   0.401 0.699045
## Past         0.33368    0.05934   5.623 0.000497 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.55 on 8 degrees of freedom
## Multiple R-squared:  0.7981, Adjusted R-squared:  0.7728
## F-statistic: 31.62 on 1 and 8 DF,  p-value: 0.0004968
```

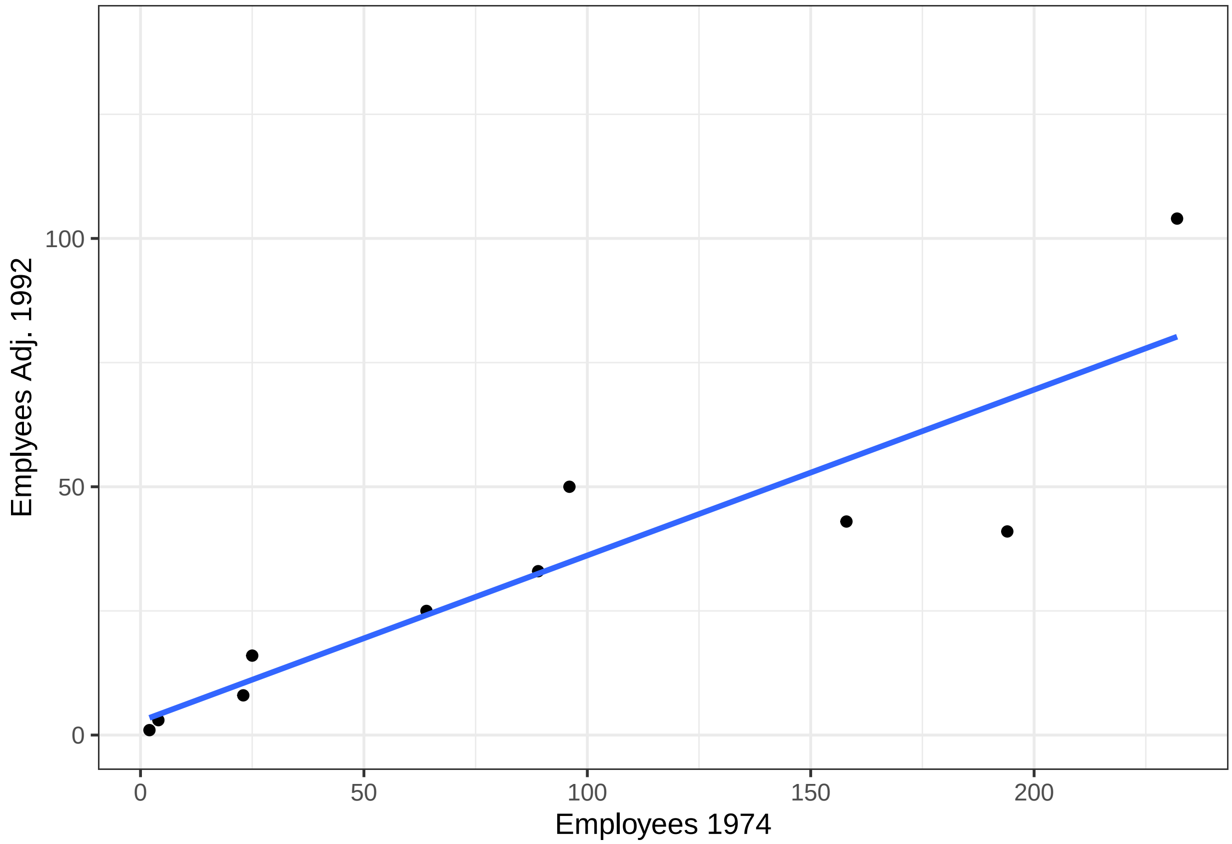
Obtain additional model information

```
model_ols_aug <- augment(model_ols) %>%
  mutate(ext.res = rstudent(model_ols)) %>%
  print()
```

```
## # A tibble: 10 x 9
##    Present  Past .fitted  .resid .std.resid  .hat .sigma   .cooksd ext.res
##      <int> <int>   <dbl>   <dbl>      <dbl> <dbl>  <dbl>     <dbl>   <dbl>
##  1     104   232   80.2   23.8       2.19   0.441   9.87 1.89        3.22
##  2      50    96   34.8   15.2       1.10   0.101  14.3  0.0677      1.12
##  3      43   158   55.5  -12.5      -0.950  0.180  14.7  0.0990     -0.944
##  4      41   194   67.5  -26.5      -2.16   0.284  10.1  0.923      -3.11
##  5      33    89   32.5    0.500     0.0362 0.100  15.6  0.0000728   0.0339
##  6      25    64   24.2    0.842     0.0613 0.110  15.6  0.000233    0.0574
##  7      16    25   11.1    4.86      0.366  0.167  15.4  0.0134      0.345
##  8       8    23   10.5   -2.48     -0.187  0.172  15.5  0.00363    -0.175
##  9       3     4    4.14  -1.14     -0.0884 0.219  15.6  0.00110    -0.0828
## 10       1     2    3.47  -2.47     -0.193  0.225  15.5  0.00539    -0.181
```

Visualize the OLS fit

```
ggplot(model_ols_aug, aes(x=Past))+
  geom_point(aes(y=Present))+
  geom_line(aes(y=.fitted), colour="#3366FF", size=1)+
  labs(x="Employees 1974", y="Emplyees Adj. 1992")+
  coord_cartesian(ylim=c(0, 140))
```



Fit a RLM

```
model_rlm <- rlm(Present ~ Past, psi=psi.huber, k=2, maxit=40, data=steeladj)
summary(model_rlm)
```

```
##
## Call: rlm(formula = Present ~ Past, data = steeladj, psi = psi.huber,
##     k = 2, maxit = 40)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.5080  -2.9070  -0.2361   3.7796  26.3134
##
## Coefficients:
##             Value  Std. Error t value
## (Intercept) 3.3334 6.2164     0.5362
## Past        0.3205 0.0528     6.0746
##
## Residual standard error: 5.655 on 8 degrees of freedom
```

Obtain additional model information

```
model_rlm_aug <- augment(model_rlm) %>%
  mutate(hweights = model_rlm$w) %>%
  relocate(hweights, .before=.fitted) %>%
  print()
```

```
## # A tibble: 10 x 7
##    Present  Past hweights .fitted .resid   .hat .sigma
##      <int> <int>    <dbl>   <dbl>  <dbl>  <dbl>  <dbl>
```

```
## 1     104   232    0.430   77.7    26.3  0.323    9.91
## 2      50    96    0.711   34.1    15.9  0.0926  14.3
## 3      43   158    1        54.0   -11.0  0.299   14.8
## 4      41   194    0.461   65.5   -24.5  0.224   11.6
## 5      33    89    1        31.9     1.14 0.123   15.6
## 6      25    64    1        23.8     1.16 0.118   15.6
## 7      16    25    1        11.3     4.65 0.172   15.5
## 8       8    23    1        10.7    -2.70 0.177   15.6
## 9       3     4    1         4.62   -1.62 0.233   15.6
## 10      1     2    1         3.97   -2.97 0.240   15.6
```

Comparison of models

Data prep

Once again, observations 1 and 4 were considered influential and outliers regardless of the adjustment previously made. To the `steeladj` data, we once again create a new variable called `Problem`.

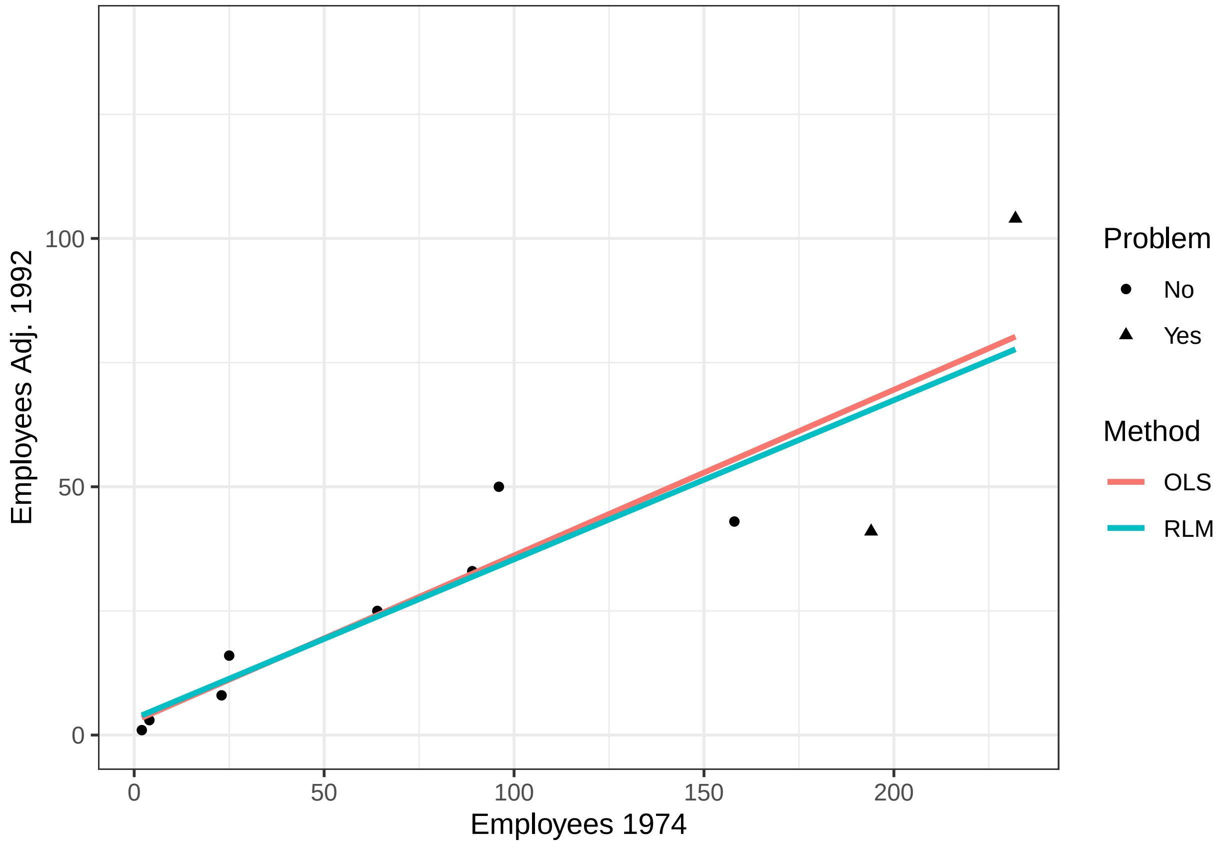
```
steeladj <- steeladj %>%
  mutate(Problem = factor(ifelse(rownames(steeladj) %in% c("1", "4"), "Yes", "No")))

model_ols_aug <- model_ols_aug %>%
  mutate(Method = "OLS")

model_rlm_aug <- model_rlm_aug %>%
  mutate(Method = "RLM")
```

Visualization

```
ggplot(data=NULL, aes(x=Past))+
  geom_point(data=steeladj, aes(y=Present, shape=Problem))+
  geom_line(data=model_ols_aug, aes(y=.fitted, colour=Method), size=1)+
  geom_line(data=model_rlm_aug, aes(y=.fitted, colour=Method), size=1)+
  labs(x="Employees 1974", y="Employees Adj. 1992")+
  coord_cartesian(ylim=c(0, 140))
```



We note that the difference in the fitted lines is much smaller than the previous comparison plot when the present value for Germany was not adjusted.

Fit a OLS model passing through the origin

```r
model_origin <- lm(Present ~ 0 + Past, data=steeladj)
summary(model_origin)
```

```
##
## Call:
## lm(formula = Present ~ 0 + Past, data = steeladj)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.2076   0.0094   1.6513   6.0324  22.4321
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## Past   0.3516     0.0372   9.452 5.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.86 on 9 degrees of freedom
## Multiple R-squared:  0.9085, Adjusted R-squared:  0.8983
## F-statistic: 89.34 on 1 and 9 DF,  p-value: 5.708e-06
```

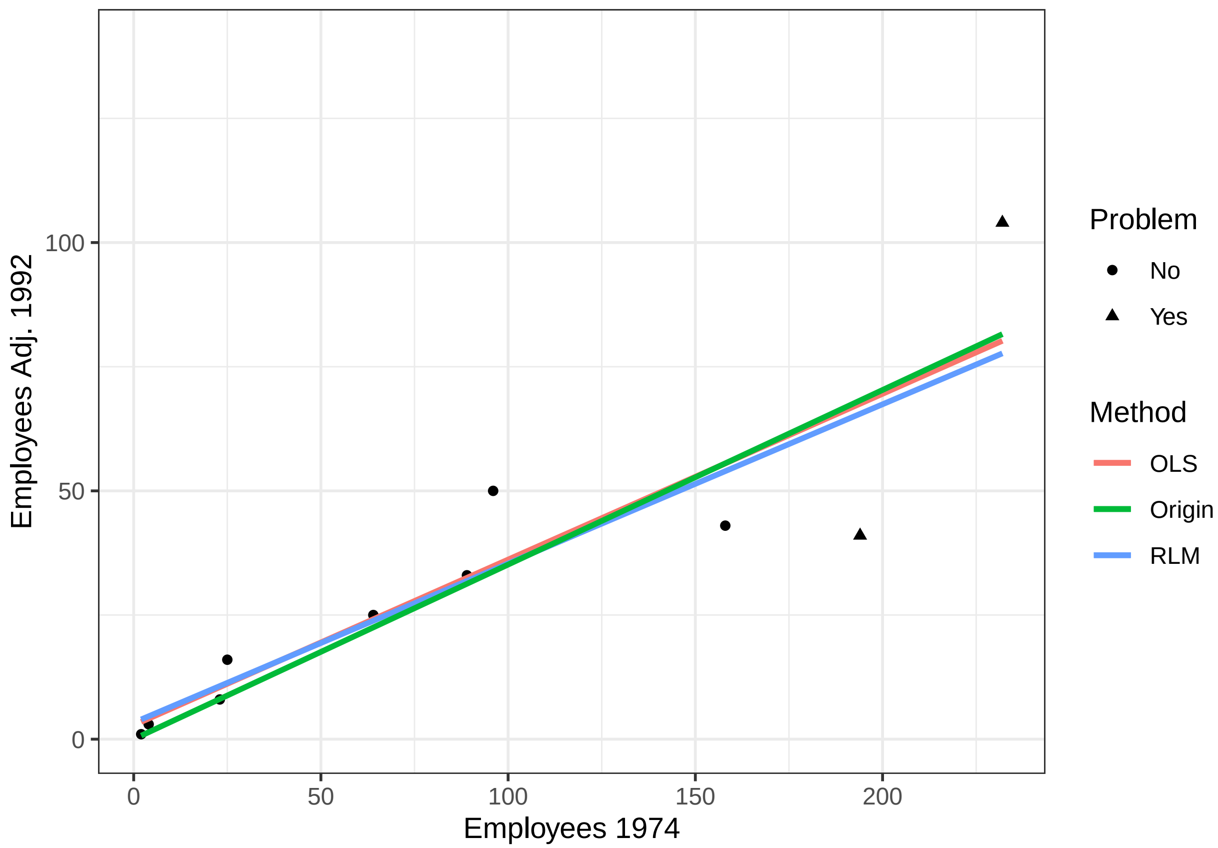
Comparison of models

Data prep

```
model_origin_aug <- augment(model_origin) %>%
  mutate(Method = "Origin")
```

Visualization

```
ggplot(data=NULL, aes(x=Past))+
  geom_point(data=steeladj, aes(y=Present, shape=Problem))+
  geom_line(data=model_ols_aug, aes(y=.fitted, colour=Method), size=1)+
  geom_line(data=model_rlm_aug, aes(y=.fitted, colour=Method), size=1)+
  geom_line(data=model_origin_aug, aes(y=.fitted, colour=Method), size=1)+
  labs(x="Employees 1974", y="Employees Adj. 1992")+
  coord_cartesian(ylim=c(0, 140))
```



All three fits are quite similar!

Comparison of the coefficients

```
list(
  ols = tidy(model_ols),
  rlm = tidy(model_rlm),
  origin = tidy(model_origin)
)
```

```
## $ols
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    2.80      6.99       0.401 0.699
## 2 Past           0.334     0.0593     5.62  0.000497
##
## $rlm
## # A tibble: 2 x 4
##   term        estimate std.error statistic
##   <chr>          <dbl>     <dbl>     <dbl>
## 1 (Intercept)    3.33      6.22       0.536
## 2 Past           0.320     0.0528     6.07
##
## $origin
## # A tibble: 1 x 5
##   term  estimate std.error statistic    p.value
##   <chr>    <dbl>     <dbl>     <dbl>      <dbl>
## 1 Past     0.352    0.0372      9.45 0.00000571
```

Variance-covariance matrix of robust adjusted model

```
# Extract design matrix
(X <- model.matrix(model_rlm))
```

```
##    (Intercept) Past
## 1            1  232
## 2            1   96
## 3            1  158
## 4            1  194
## 5            1   89
## 6            1   64
## 7            1   25
## 8            1   23
## 9            1    4
## 10           1    2
## attr(,"assign")
## [1] 0 1
```

```r
# Create the diagonal matrix of final weights
(W <- diag(model_rlm_aug$hweights))
```

```
##             [,1]      [,2] [,3]      [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,] 0.4298477 0.0000000    0 0.0000000    0    0    0    0    0     0
##  [2,] 0.0000000 0.7113541    0 0.0000000    0    0    0    0    0     0
##  [3,] 0.0000000 0.0000000    1 0.0000000    0    0    0    0    0     0
##  [4,] 0.0000000 0.0000000    0 0.4614499    0    0    0    0    0     0
##  [5,] 0.0000000 0.0000000    0 0.0000000    1    0    0    0    0     0
##  [6,] 0.0000000 0.0000000    0 0.0000000    0    1    0    0    0     0
##  [7,] 0.0000000 0.0000000    0 0.0000000    0    0    1    0    0     0
##  [8,] 0.0000000 0.0000000    0 0.0000000    0    0    0    1    0     0
##  [9,] 0.0000000 0.0000000    0 0.0000000    0    0    0    0    1     0
## [10,] 0.0000000 0.0000000    0 0.0000000    0    0    0    0    0     1
```

```r
XTWXinv <- solve(t(X) %*% W %*% X)
s_sq <- summary(model_rlm)$sigma^2
(V <- s_sq * XTWXinv)
```

```
##             (Intercept)         Past
## (Intercept)  7.88685707 -0.057617842
## Past        -0.05761784  0.000796205
```

This variance-covariance matrix is the same as the one shown near the end of *Module 7.4*. As mentioned in this module, we will use the above variance-covariance matrix for robust models rather than the one obtained from `vcov()`.

Cement data

```r
heat <- read.table("./cement.txt", header=TRUE)
```

Stepwise regression

We begin stepwise regression by fitting a null model. Then at each step, we check whether we can add a variable, and drop a variable. We keep repeating this until no more variables can be added or dropped. We will use

$$\alpha_{\text{entry}} = 0.10, \quad \alpha_{\text{stay}} = 0.10$$

Fit the null model

```r
model0 <- lm(Y ~ 1, data=heat)
```

Check if we can add a variable

```r
add1(model0, ~ X1 + X2 + X3 + X4, test="F")
```

```
## Single term additions
##
## Model:
## Y ~ 1
##        Df Sum of Sq     RSS    AIC F value    Pr(>F)
## <none>              2715.76 71.444
## X1      1   1450.08 1265.69 63.519 12.6025 0.0045520 **
## X2      1   1809.43  906.34 59.178 21.9606 0.0006648 ***
## X3      1    776.36 1939.40 69.067  4.4034 0.0597623 .
## X4      1   1831.90  883.87 58.852 22.7985 0.0005762 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using a **scope** of `~ X1 + X2 + X3 + X4` means that we are considering adding any **one** of `X1`, `X2`, `X3`, or `X4` into our existing model. This becomes 4 individual hypothesis tests of:

$$H_0 : \beta_i = 0 \quad \text{vs} \quad H_A : \beta_i \neq 0$$

for $i = 1, 2, 3, 4$.

Since the p-values corresponding to each test is less than $\alpha_{\text{entry}} = 0.10$, each variable is an eligible candidate for entry. However, the variable that we add to our model will have the smallest p-value. This corresponds to `X4`.

Incorporate `X4` into our model

```
model1 <- update(model0, . ~ . + X4)
```

The command above says that `model1` will be an update of `model0` (i.e. `model1` is based off of `model0`). Note that the second argument is a "formula" rather than a "scope". The dot on the left of the tilde means keep the same response variable as `model0` (`Y`). The dot on the right of the tilde means keep the same predictor variables as `model0` (the intercept), though this dot is not actually needed since all models are fit with an intercept by default. The `+ X4` means that we are adding `X4` into our model.

We can double check the formula afterwards using

```
formula(model1)
```

```
## Y ~ X4
```

We will not check to see if we can drop a variable at this stage since we just added our first variable.

Check if we can add another variable

```
add1(model1, ~ . + X1 + X2 + X3, test="F")
```

```
## Single term additions
##
## Model:
## Y ~ X4
##        Df Sum of Sq    RSS    AIC  F value    Pr(>F)
## <none>              883.87 58.852
## X1      1    809.10  74.76 28.742 108.2239 1.105e-06 ***
## X2      1     14.99 868.88 60.629   0.1725    0.6867
## X3      1    708.13 175.74 39.853  40.2946 8.375e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The scope `~ . + X1 + X2 + X3` means that we want to keep the existing predictors of `model1` and consider adding any one of `X1`, `X2`, or `X3`. Similar to before, we perform three individual hypothesis tests of:

$$H_0 : \beta_i = 0 \quad \text{vs} \quad H_A : \beta_i \neq 0$$

for $i = 1, 2, 3$.

This time, `X1` and `X3` are eligible candidates to be added to our model because their p-values are less than $\alpha_{\text{entry}} = 0.10$ (i.e. reject the null hypothesis for $i = 1, 3$). The variable we will add will have the lowest p-value. This corresponds to `X1`.

Incorporate `X1` into our model

```
model2 <- update(model1, . ~ . + X1)
formula(model2)
```

```
## Y ~ X4 + X1
```

Similar to before, the above command says that `model2` will be based off of `model1`. `model2` will have the same response and predictors as `model1`, but it will also include `X1` now.

Check if we can drop a variable

```r
drop1(model2, ~ ., test="F")
```

```
## Single term deletions
##
## Model:
## Y ~ X4 + X1
##        Df Sum of Sq     RSS    AIC F value    Pr(>F)
## <none>                74.76 28.742
## X4      1    1190.9 1265.69 63.519  159.30 1.815e-07 ***
## X1      1     809.1  883.87 58.852  108.22 1.105e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The above command says that we are interested in dropping a single predictor from our model. The scope `~ .` means that all predictors in `model2` are eligible for dropping. This corresponds to performing the individual hypothesis tests of:

$$H_0 : \beta_i = 0 \quad \text{vs} \quad H_A : \beta_i \neq 0$$

for $i = 1, 4$.

We reject the null hypothesis for both tests since both p-values are less than $\alpha_{\text{stay}} = 0.10$. This means that neither variable is eligible for dropping.

Check if we can add another variable

```r
add1(model2, ~ . + X2 + X3, test="F")
```

```
## Single term additions
##
## Model:
## Y ~ X4 + X1
##        Df Sum of Sq    RSS    AIC F value  Pr(>F)
## <none>              74.762 28.742
## X2      1    26.789 47.973 24.974  5.0259 0.05169 .
## X3      1    23.926 50.836 25.728  4.2358 0.06969 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The hypothesis tests we perform are:

$$H_0 : \beta_i = 0 \quad \text{vs} \quad H_A : \beta_i \neq 0$$

for $i = 2, 3$. Since the p-values are both less than $\alpha_{\text{entry}} = 0.10$, each variable is eligible to be added into the model. `X2` is added to our model since it has the smallest p-value.

Incorporate `X2` into the model

```r
model3 <- update(model2, . ~ . + X2)
formula(model3)
```

```
## Y ~ X4 + X1 + X2
```

Check if we can drop a variable

```
drop1(model3, ~ ., test="F")
```

```
## Single term deletions
##
## Model:
## Y ~ X4 + X1 + X2
##        Df Sum of Sq    RSS    AIC  F value    Pr(>F)
## <none>               47.97 24.974
## X4      1      9.93  57.90 25.420   1.8633   0.20540
## X1      1    820.91 868.88 60.629 154.0076 5.781e-07 ***
## X2      1     26.79  74.76 28.742   5.0259   0.05169 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This corresponds to testing:

$$H_0 : \beta_i = 0 \quad \text{vs} \quad H_A : \beta_i \neq 0$$

for $i = 1, 2, 4$. For $i = 1, 2$, we reject the null hypothesis because each p-value is less than $\alpha_{\text{stay}} = 0.10$. Since we fail to reject the null hypothesis for $i = 4$, `X4` will be dropped from the model.

Drop `X4` from the model

```
model4 <- update(model3, . ~ . - X4)
formula(model4)
```

```
## Y ~ X1 + X2
```

Note that here we use `- X4` to indicate that `X4` is being dropped.

Check to see if we can add another variable

```
add1(model4, ~ . + X3 + X4, test="F")
```

```
## Single term additions
##
## Model:
## Y ~ X1 + X2
##        Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>              57.904 25.420
## X3      1    9.7939 48.111 25.011  1.8321 0.2089
## X4      1    9.9318 47.973 24.974  1.8633 0.2054
```

This corresponds to testing:

$$H_0 : \beta_i = 0 \quad \text{vs} \quad H_A : \beta_i \neq 0$$

for $i = 3, 4$. Since both p-values are greater than $\alpha_{\text{entry}} = 0.10$, we fail to reject each null hypothesis. This means that neither variable is eligible to be added into our model.

Check to see if we can drop a variable

```
drop1(model4, ~ ., test="F")
```

```
## Single term deletions
##
## Model:
## Y ~ X1 + X2
##        Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                57.90 25.420
```

```
## X1       1    848.43  906.34 59.178  146.52 2.692e-07 ***
## X2       1   1207.78 1265.69 63.519  208.58 5.029e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The corresponding hypotheses are:

$$H_0 : \beta_i = 0 \quad \text{vs} \quad H_A : \beta_i \neq 0$$

for $i = 1, 2$. Since each p-value is less than $\alpha_{\text{stay}} = 0.10$, we reject each null hypothesis. This means that neither variable can be dropped from the model.

The final model

```r
summary(model4)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2, data = heat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.893 -1.574 -1.302  1.363  4.048
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 52.57735    2.28617   23.00 5.46e-10 ***
## X1           1.46831    0.12130   12.11 2.69e-07 ***
## X2           0.66225    0.04585   14.44 5.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.406 on 10 degrees of freedom
## Multiple R-squared:  0.9787, Adjusted R-squared:  0.9744
## F-statistic: 229.5 on 2 and 10 DF,  p-value: 4.407e-09
```

```r
anova(model4)
```

```
## Analysis of Variance Table
##
## Response: Y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## X1         1 1450.1 1450.08  250.43 2.088e-08 ***
## X2         1 1207.8 1207.78  208.58 5.029e-08 ***
## Residuals 10   57.9    5.79
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Best subsets regression

We perform a best subsets regression for the best two models containing one, two, and three predictors.

```r
subsets <- regsubsets(Y ~ X1 + X2 + X3 + X4, method="exhaustive", nbest=2, data=heat)
```

Results

```r
var_y <- var(heat$Y)
s_sq <- var_y * (1 - summary(subsets)$adjr2)

data.frame(
```

```r
  summary(subsets)$outmat,
  r.sq = round(summary(subsets)$rsq, 3),
  adj.r.sq = round(summary(subsets)$adjr2, 3),
  cp = round(summary(subsets)$cp, 1),
  s.sq = round(s_sq, 3)
)
```

```
##          X1 X2 X3 X4  r.sq adj.r.sq    cp   s.sq
## 1  ( 1 )           * 0.675    0.645 138.7 80.352
## 1  ( 2 )     *       0.666    0.636 142.5 82.394
## 2  ( 1 )  *  *       0.979    0.974   2.7  5.790
## 2  ( 2 )  *        * 0.972    0.967   5.5  7.476
## 3  ( 1 )  *  *     * 0.982    0.976   3.0  5.330
## 3  ( 2 )  *  *  *    0.982    0.976   3.0  5.346
## 4  ( 1 )  *  *  *  * 0.982    0.974   5.0  5.983
```

But since I'm obsessed with the tidyverse:

```r
var_y <- var(heat$Y)

out <- tidy(subsets) %>%
  select(-`(Intercept)`, -BIC) %>%
  mutate(
    rank = c("(1)", "(2)", "(1)", "(2)", "(1)", "(2)", "(1)"),
    X1 = ifelse(X1, "*", "-"),
    X2 = ifelse(X2, "*", "-"),
    X3 = ifelse(X3, "*", "-"),
    X4 = ifelse(X4, "*", "-"),
    p = c(1, 1, 2, 2, 3, 3, 4),
    `p+1` = p+1,
    s.sq = var_y * adj.r.squared,
  ) %>%
  relocate(p, rank, .before=everything()) %>%
  rename(r.sq = r.squared, adj.r.sq = adj.r.squared, cp = mallows_cp) %>%
  relocate(`p+1`, .before=cp) %>%
  print()
```

```
## # A tibble: 7 x 11
##       p rank  X1    X2    X3    X4     r.sq adj.r.sq `p+1`     cp  s.sq
##   <dbl> <chr> <chr> <chr> <chr> <chr> <dbl>    <dbl> <dbl>  <dbl> <dbl>
## 1     1 (1)   -     -     -     *     0.675    0.645     2 139.    146.
## 2     1 (2)   -     *     -     -     0.666    0.636     2 142.    144.
## 3     2 (1)   *     *     -     -     0.979    0.974     3   2.68  221.
## 4     2 (2)   *     -     -     *     0.972    0.967     3   5.50  219.
## 5     3 (1)   *     *     -     *     0.982    0.976     4   3.02  221.
## 6     3 (2)   *     *     *     -     0.982    0.976     4   3.04  221.
## 7     4 (1)   *     *     *     *     0.982    0.974     5   5     220.
```

We are looking for a model with a high R^2_{adj}, low s^2, and $C_p \approx p+1$. The model that best satisfies all three of these conditions is the model with `X1` and `X2`. This corresponds to row 3.

```r
out %>%
  slice(3)
```

```
## # A tibble: 1 x 11
##       p rank  X1    X2    X3    X4     r.sq adj.r.sq `p+1`    cp  s.sq
##   <dbl> <chr> <chr> <chr> <chr> <chr> <dbl>    <dbl> <dbl> <dbl> <dbl>
## 1     2 (1)   *     *     -     -     0.979    0.974     3  2.68  221.
```