

Multiple Linear Regression

We consider the 1987 baseball salary data originally from the 1988 ASA (American Statistical Association) exposition competition. The data contain the salary information for 263 major league hitters and 22 predictors, as listed in the following table. The response variable is the logarithm of salary. One research question of interest is – “*Are Baseball Salaries Based on Performance?*”

<i>X</i>	Name	Description	<i>X</i>	Name	Description
1	bat86	times at bat - 86'	12	rbcr	runs-batted-in
2	hit86	hits in 86'	13	wlkr	walks in career
3	hr86	home runs in 86'	14	leag86	league in 86'
4	run86	runs in 86'	15	div86	division in 86'
5	rb86	runs batted in in 86'	16	team86	team in 86'
6	wlk86	walks in 86'	17	pos86	position in 86'
7	yrs	years in major league	18	puto86	put outs in 86'
8	batcr	times at bat - career	19	asst86	assists in 86'
9	hitcr	hits in career	20	err86	errors in 86'
10	hrcr	home runs in career	21	leag87	league in 87'
11	runcr	runs during career	22	team87	team in 87'

This baseball data has been studied frequently in statistical literatures. Hoaglin and Velleman (1995) provided a nice overview on analyses using various statistical methods and they found that the following model yields good model fit and leads to sensible interpretations.

$$\log(\text{salary}) = \beta_0 + \beta_1 \frac{\text{runcr}}{\text{yrs}} + \beta_2 \sqrt{\text{run86}} + \beta_3 \min[(\text{yrs} - 2)_+, 5] + \beta_4 (\text{yrs} - 7)_+ + \varepsilon, \quad (1)$$

where $\varepsilon \sim N(0, \sigma^2)$, and the segmentation on year is based on a player's eligibility for arbitration or free agency.

To fit HV's model using R, first define these transformed covariates after inputting data.

```
> baseball <- read.table("A://bb.dat", header = F, col.names=c("id",
  "name", "bat86", "hit86", "hr86", "run86", "rb86", "wlk86", "yrs",
  "batcr", "hitcr", "hrcr", "runcr", "rbcr", "wlkcr", "leag86",
  "div86", "team86", "pos86", "puto86", "asst86", "err86", "salary",
  "leag87", "team87", "logsalary"))

> dim(baseball)
[1] 263 26
```

```
> attach(baseball)
```

The function `attach` attach an R object to the R search path so that objects in the database can be accessed by simply giving their names.

```
> x1 <- runcr/yrs
> x2 <- sqrt(run86)
> x3 <- min(max(yrs - 2, 0), 5)
> x4 <- max(yrs - 7, 0)
```

1 Preliminary Data Exploration

1.1 Univariate Methods

One may use `summary(baseball)` to get an univariate summary for each variable in the data set. Other univariate techniques may be used to check the shape, distribution, and numerical statistical properties of the response - `logsalary`. The reasons for transformation in regression include enhancing normality, improving linearity, and also stabilizing variance.

```
postscript("c:/COURSES/STA4164/HD4-fg1.eps", horizontal=F)
par(mfrow=c(2,2),mar=c(4, 4, 4, 4))
# First Plot
hist(salary, xlab="salary", main="Histogram of Salary")
# Second Plot
qqnorm(salary, main="Q-Q Plot of Salary")
qqline(salary)
# Third Plot
hist(logsalary, xlab="log-alary", main="Histogram of Log(Salary)")
# Forth Plot
qqnorm(logsalary, main="Q-Q Plot of Log(Salary)")
qqline(logsalary)
dev.off()
```

The commands lines

```
postscript("c:/COURSES/STA4164/HD4-fg1.eps", horizontal=F)
.....
dev.off
```

basically create a nice eps plot and output it to an external file. The function `qqnorm` makes Q-Q (quantile-quantile) plot for the purpose of assessing normality. And `qqline` adds a line to a normal Q-Q plot which passes through the first and third quartiles. If the points mostly center around the line, normality is OK. On the other hand, any strong nonlinear pattern reveals certain deviation from normality.

One may use other univariate descriptive techniques to explore both the response and the predictors. One useful commands to check categorical covariates is `table`.

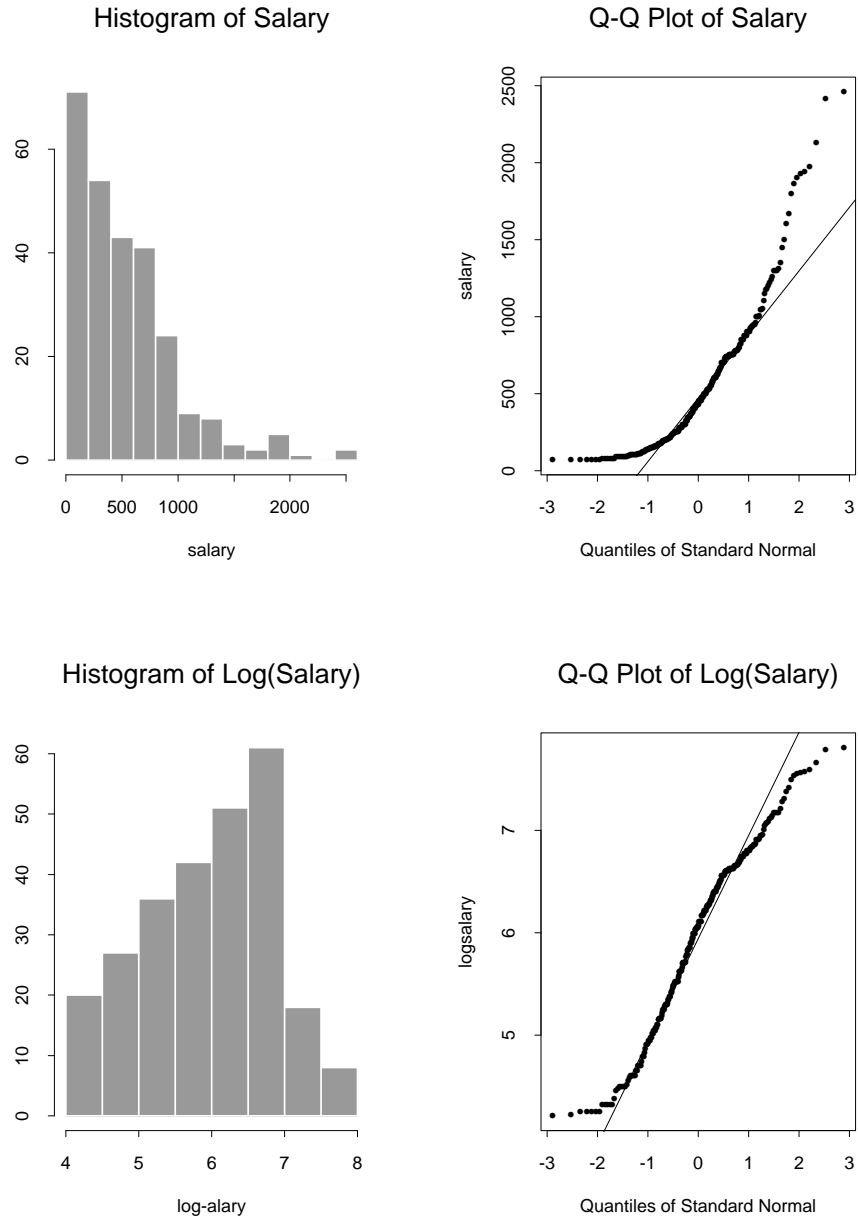


Figure 1: Graphical Exploration of `salary`. One may see that after the logarithm transformation, its normality is enhanced.

```
> table(leag86)
leag86
  A  N
139 124
```

1.2 Bivariate Methods

There are two bivariate techniques that are useful for exploring the relationships in regression: the correlation matrix and the paired scatterplot. The pairwise correlation matrix for all *continuous* variables in the data can be output using function `cor`.

```
> bb <- data.frame(logsalary, x1, x2, x3, x4)
> cor(bb)
      logsalary      x1      x2      x3      x4
logsalary 1.0000000 0.6207478 0.48181372 0.75383816 0.35416899
x1         0.6207478 1.0000000 0.67377607 0.23091139 0.28540691
x2         0.4818137 0.6737761 1.00000000 0.06652986 -0.08284422
x3         0.7538382 0.2309114 0.06652986 1.00000000 0.55695377
x4         0.3541690 0.2854069 -0.08284422 0.55695377 1.00000000
```

Here, the number of digits after the decimal point can be controlled by `options(digits=3)`.

The second bivariate technique is the pairwise scatterplot. The R function `pairs` is designed for this purpose. The following commands make a plot as shown in figure 2.

```
# postscript("c:/COURSES/STA4164/WORK/HD4-fg2.eps", horizontal=F)
> par(mar=c(4, 4, 4, 4))
> pairs(bb)
# dev.off()
```

2 Hoaglin and Velleman's (1995) Model

Now we fit Hoaglin and Velleman's (1995) Model.

```
> fit <- lm(logsalary ~ x1 + x2 + x3 + x4)
> summary(fit)
```

```
Coefficients:
      Value Std. Error  t value Pr(>|t|)
(Intercept)  3.5295    0.1126   31.3502  0.0000
      x1     0.0164    0.0017    9.5974  0.0000
      x2     0.0818    0.0201    4.0708  0.0001
      x3     0.3474    0.0151   23.0657  0.0000
      x4    -0.0405    0.0091   -4.4340  0.0000
```

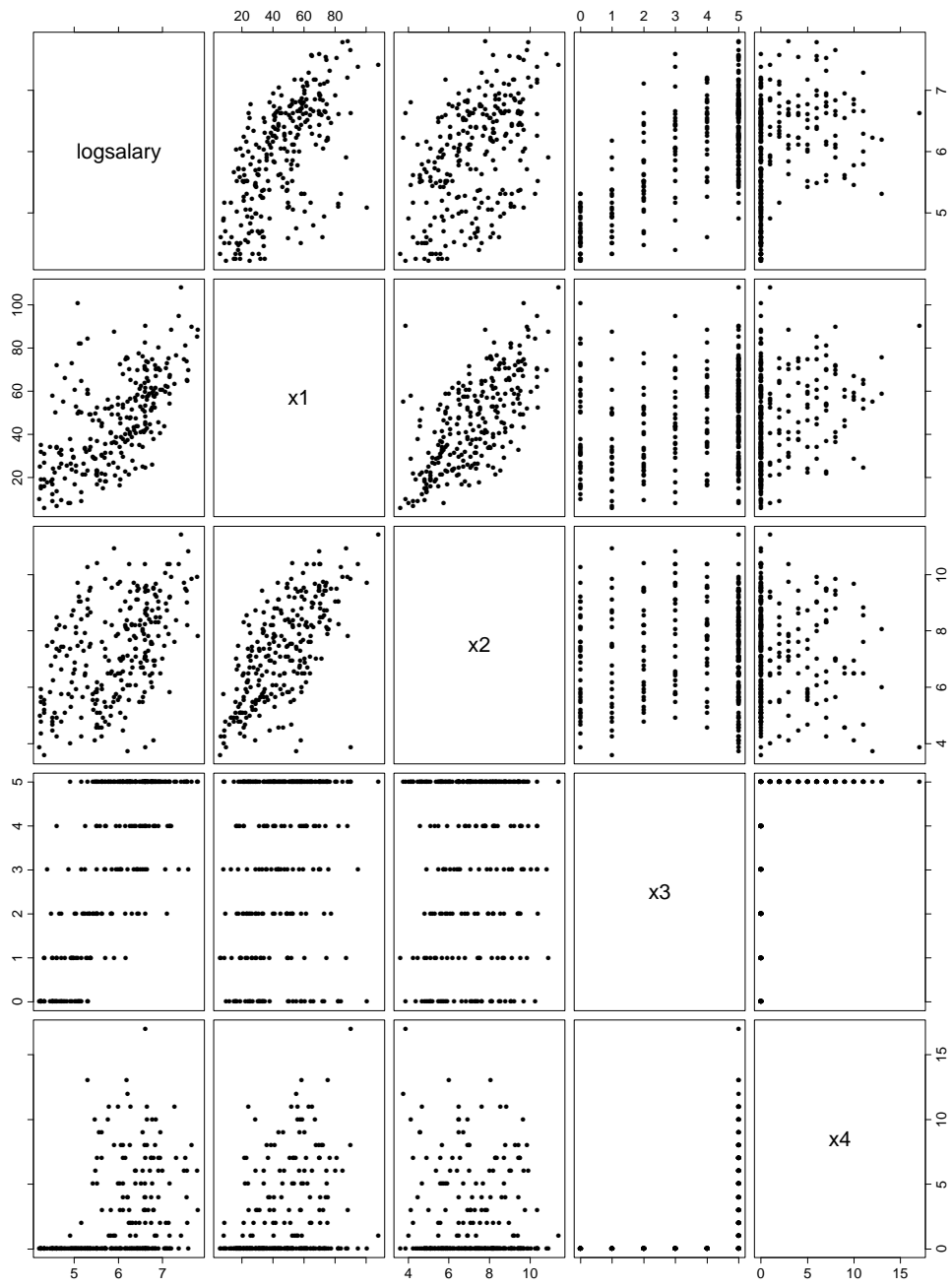


Figure 2: Paired Scatterplot

Residual standard error: 0.3778 on 258 degrees of freedom Multiple
R-Squared: 0.82
F-statistic: 293.8 on 4 and 258 degrees of freedom, the p-value is 0

The estimates for regression parameters are all significant, as evidenced by the small P-values. The coefficient of determination, R^2 , is equal to .4624. Namely, about 83% of the total sample variation in $\log(\text{salary})$ can be explained by the regression model. The F test for the overall usefulness of the model has a p -value of essentially 0. The ANOVA table can also be output using function `anova.lm`.

```
> anova.lm(fit, ssType = 1)
Analysis of Variance Table

Response: logsalary

Terms added sequentially (first to last)
      Df Sum of Sq  Mean Sq  F Value    Pr(F)
x1     1  78.83862  78.83862  552.2403 0.000000000
x2     1   1.51420   1.51420   10.6065 0.001277886
x3     1  84.60938  84.60938  592.6627 0.000000000
x4     1   2.80676   2.80676   19.6605 0.000013697
Residuals 258  36.83245   0.14276
```

However, the sum of squares due to regression are given as the sequential sum of squares. One may use the following commands to obtain the total SSR.

```
> names(out)
[1] "Df"          "Sum of Sq"  "Mean Sq"   "F Value"   "Pr(F)"

> ss <- out$"Sum of Sq"; ss

[1] 78.838622  1.514196 84.609380  2.806759 36.832449

> ss.total <- sum(ss); ss.total
[1] 204.6014

> L <- length(ss)
> sse <- ss[L]; sse
[1] 36.83245

> ssr <- ss.total - sse; ssr
[1] 167.769
```

3 F -Test for Nested Models

To illustrate the use of the general F testing method for nested models, we consider the problem of testing $H_0 : \beta_1 = \beta_2$. Therefore, the reduced model becomes

$$\begin{aligned} \log(\text{salary}) &= \beta_0 + \beta_1 \frac{\text{runcr}}{\text{yrs}} + \beta_1 \sqrt{\text{run86}} + \beta_3 \min[(\text{yrs} - 2)_+, 5] + \beta_4 (\text{yrs} - 7)_+ + \varepsilon \\ &= \beta_0 + \beta_1 \cdot \left(\frac{\text{runcr}}{\text{yrs}} + \sqrt{\text{run86}} \right) + \beta_3 \min[(\text{yrs} - 2)_+, 5] + \beta_4 (\text{yrs} - 7)_+ + \varepsilon. \end{aligned} \quad (2)$$

We first fit the reduced model by defining a new variable $x_{1.0} = \frac{\text{runcr}}{\text{yrs}} + \sqrt{\text{run86}}$.

```
> x1.0 <- x1 + x2
> fit1 <- lm(logsalary ~ x1.0 + x3 + x4)
```

Then the function `anova` can be used to perform the general F test.

```
> anova(fit1, fit, test = "F")
```

Analysis of Variance Table

```
Response: logsalary
          Terms Resid. Df      RSS    Test Df Sum of Sq  F Value      Pr(F)
1    x1.0 + x3 + x4      259 38.16828
2 x1 + x2 + x3 + x4      258 36.83245 1 vs. 2   1  1.335836  9.357118 0.002455193
```

It can be found that the observed F test statistic is 9.357118, with (1, 258) degrees of freedom. Its corresponding p value is .00246.

4 Prediction

Suppose we want to get a 95% confidence/prediction interval for the log salary of a player who has the following info: $X_1 = 45$, $X_2 = 7$, $X_3 = 3$, and $X_4 = 2$. Similar to simple linear regression, the function `predict` can be used to produce confidence or prediction intervals. If the CI or PI for salary itself is needed, one may simply take the exponential of the lower and upper bound. This leads to better interpretation of the results.

```
> new <- data.frame(x1 = 45, x2 = 7, x3 = 3, x4 = 2)
> predict(fit, newdata = new, se.fit = TRUE, interval = "confidence", level = 0.95)
$fit:
 5.80202
$se.fit:
 0.024002

> predict(fit, newdata = new, se.fit = TRUE, interval = "prediction", level = 0.95)
$fit:
 5.80202
$se.fit:
 0.024002
```

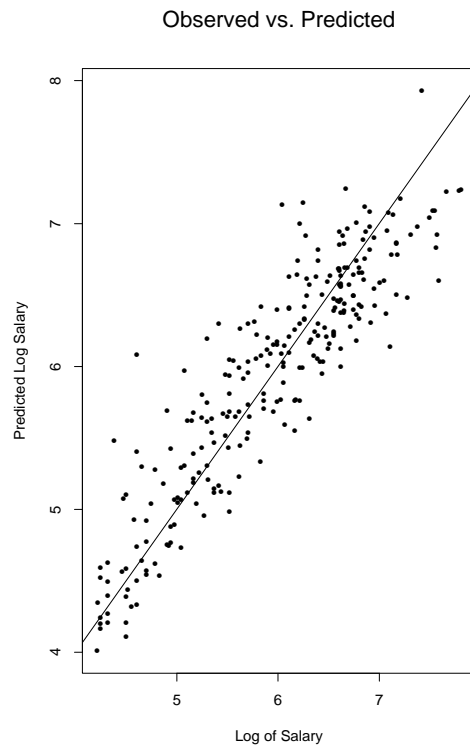


Figure 3: Plot of Predicted vs. Observed

In addition, one meaningful plot for assessing the model fit is the scatter plot of predicted vs. observed responses. Figure 3 shows a reasonable model fit.

```
# postscript("c:/COURSES/STA4164/WORK/HD4-fg3.eps", horizontal=F)
par(mfrow=c(1,1), mar=rep(6,4))
plot(logsalary, fit$fitted.values, main="Observed vs. Predicted",
     xlab="Log of Salary", ylab="Predicted Log Salary")
abline(0,1) # add the line y=x
# dev.off()
```