

## Model Diagnostics

### The Hoaglin and Velleman (1995) Model

Let's consider the following model suggested by Hoaglin and Velleman (1995) for the 1987 Baseball salary data.

```
baseball <- read.table("A:/bb87.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)

attach(baseball)
x1 <- runcr/yrs
x2 <- sqrt(run86)
x3 <- pmin(pmax(yrs-2, 0), 5)
x4 <- pmax(yrs-7, 0)

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

It is not enough to fit a model; we must also assess how well that model fits the data, being ready to modify the model or abandon it altogether if it does not satisfactorily explain the data. Statistical methods designed to assess the adequacy and validity of models are generally categorized into *model diagnostics*. R provides two useful functions for the purpose of model diagnostics: `plot.lm(fit)` and `inluce.measures(fit)`.

The simplest and most informative method for assessing the fit is to look at the model graphically, using an assortment of plots that, taken together, reveal the strengths and weaknesses of the model. For example, a plot of the response against the fitted values gives a good idea of how well the model has captured the broad outlines of the data, while examining a plot of the residuals against the fitted values often reveals unexplained structure left in the residuals, which in a strong model should appear as nothing but noise. The default plotting method for **lm** objects provides the following four useful plots:

- *Square root of absolute residuals against fitted values*. This plot is useful in identifying outliers and visualizing structure in the residuals.
- *Normal quantile plot of residuals*. This plot provides a visual test of the assumption that the model's errors are normally distributed. If the ordered residuals cluster along the superimposed quantile-quantile line, you have strong evidence that the errors are indeed normal.
- *Residual-Fit spread plot, or r-f plot*. This plot compares the spread of the fitted values with the spread of the residuals. Since the model is an attempt to explain the variation in the data, you hope that the spread in the fitted values is *much* greater than that in the residuals.

- *Cook's distance plot.* Cook's distance is a measure of the influence of individual observations on the regression coefficients.

```
par(mfrow=c(2,2), mar=rep(5,4))
plot(fit)
```

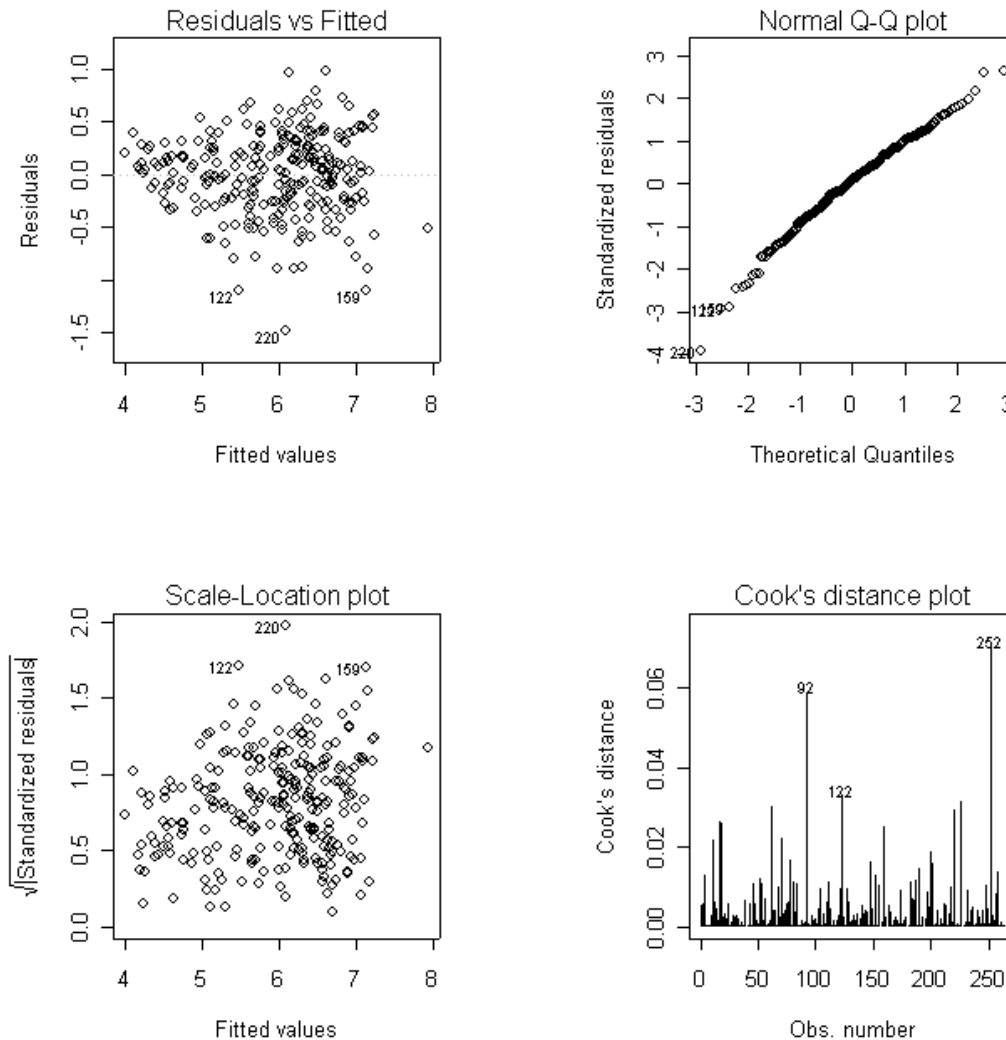


Figure 1: Diagnostic Plot provided by R function `plot.lm`

The second function `influence.measures(fit)` computes all the influence measures for a given model and is useful for detecting influence points. In the following, we are going to use R function to illustrate some commonly used model diagnostics techniques.

## 1, Analysis of The Jackknife Residuals

First obtain the jackknife (or leave-one-out) residuals, which follow a  $t$  distribution with  $(n-1)-(p+1)$  degrees of freedom, from any of the function: `rstudent(fit)`.

```
> r.jack <- rstudent(fit)
```

When the number of degrees of freedom is high, the *standard normal* distribution  $N(0,1)$  approximation applies.

### 1.1 Univariate Techniques

One may plot the histogram and Q-Q normal plot to inspect for deviation from normality. It is printed in Figure 3. No obvious deviation was found from the histogram. The Q-Q plot looks reasonably good except for several outlying observations.

```
par(mfrow=c(1,2),mar=c(8,4,8,4))
# The first plot: Histogram
hist(r.jack, xlab="Jackknife Residual", main="Histogram of Jackknife
      Residuals")

# The second plot: Q-Q plot
qqnorm(r.jack)
qqline(r.jack)
```

Some formal tests for normality are also available. Among others, the best one is the Shapiro-Wilk test, which is designed only for testing normality.

```
> shapiro.test(r.jack)

      Shapiro-Wilk normality test

data:  r.jack
W = 0.9851, p-value = 0.007862
```

Shapiro-Wilk normality test shows significant nonnormality of the jackknife residuals.

### 1.2 Bivariate Techniques

Also, two-dimensional plots can be made. The results are printed in Figure 3. From the last plot, it can be observed that the variance of jackknife residuals increases with larger predicted log-salaries. So the assumption of homoscedasticity may be violated.

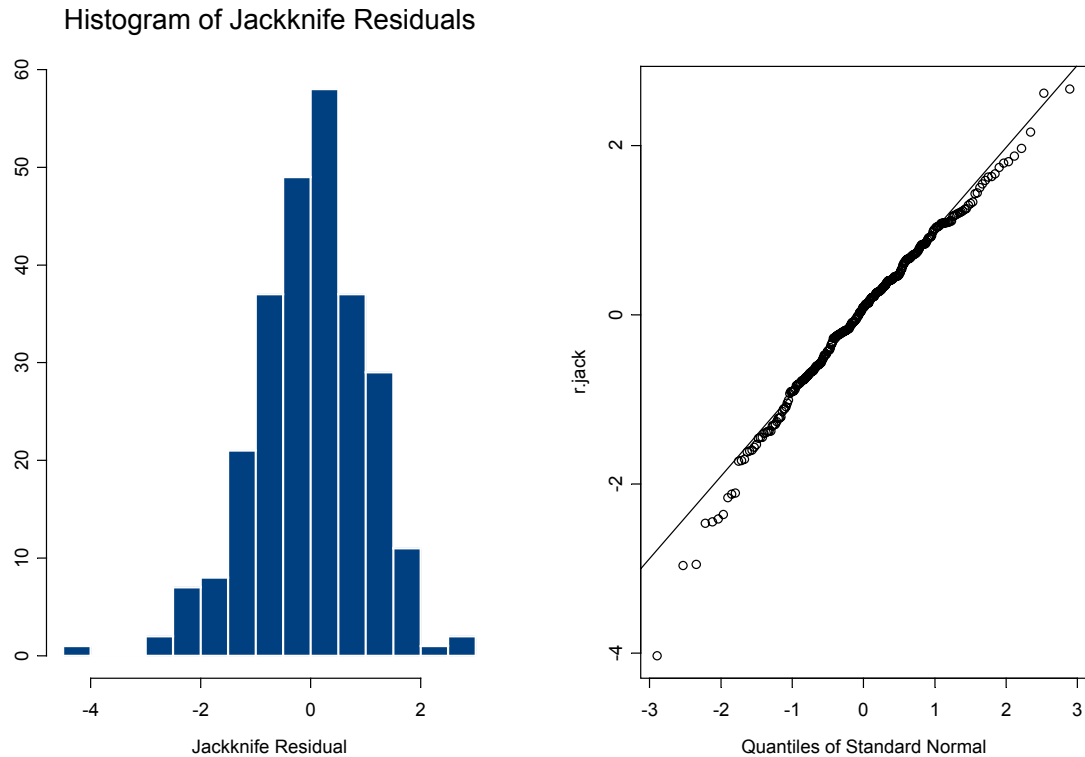


Figure 2: One-Dimensional Graphical Examination of Jackknife Residuals

```
par(mfrow=c(3,1),mar=c(4, 10, 4, 10))
plot(x1, r.jack, xlab="x1", ylab="Jackknife Residual")
abline(h=0)
# compute the degrees of freedom for the reference t distribution
n <- dim(baseball)[1]
p <- 4
df <- (n-1)-(p+1)
abline(h=qt(.975, df), lty=2)
abline(h=-qt(.975, df), lty=2)

plot(x2, r.jack, xlab="x2", ylab="Jackknife Residual")
abline(h=0)
abline(h=qt(.975, df), lty=2)
abline(h=-qt(.975, df), lty=2)

plot(fit1$fitted.values, r.jack, xlab="Predicted", ylab="Jackknife Residual")
abline(h=0)
abline(h=qt(.975, df), lty=2)
abline(h=-qt(.975, df), lty=2)
```

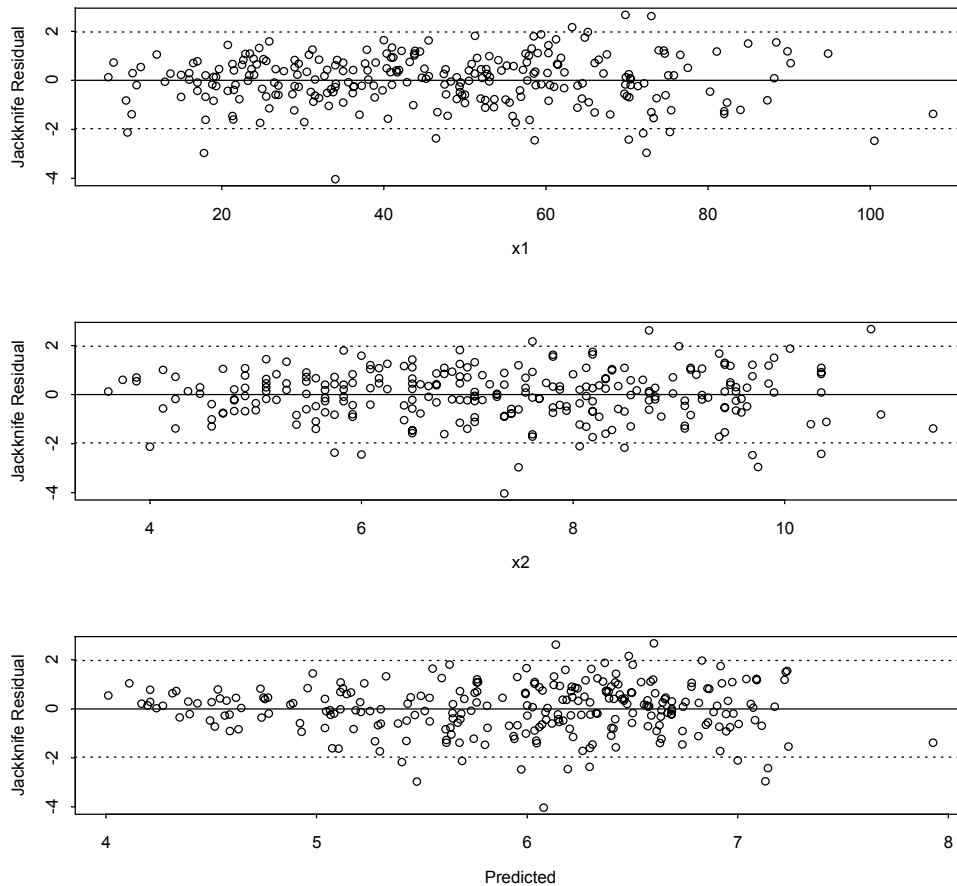
## 2, Checking Functional Form of One Individual Predictor – Partial Residual Plot

Suppose we are interested to see whether the total number of hits during a player's career – `hitcr`, should be added to model (1) and what functional form (linear or nonlinear) should be used.

```
par(mfrow=c(1, 3), oma=c(4, 4, 4, 4))
# residual vs hitcr plot
r <- resid(fit)
plot(hitcr, r, ylab="residuals", xlab="hits in during career")

# partial residual vs hitcr plot
fit1 <- lm(logsalary ~ hitcr + x1 + x2 + x3 + x4)
pr.hitcr <- resid(fit1) - fit1$coef[2]*hitcr
plot(hitcr, pr.hitcr, ylab="partial residuals", xlab="hits in during career")
abline(lsfit(hitcr, pr.hitcr))

# Partial Regression Plot for hitcr
pr.hitcr <- residuals(lm(hitcr ~ x1 + x2 + x3 + x4))
plot(pr.hitcr, r, ylab="r(logsalary|X)", xlab="r(hitcr|X)")
```



**Figure 3:** Two-Dimensional Graphical Examination of Jackknife Residuals

In the above codes, note that `hitcr` is the first predictor in the model, so its slope estimate is `fit1$coef[2]`. From the partial residual plot, it seems that `hitcr` is underrepresented in HV's model and a linear form seems fine.

```
> fit1 <- lm(logsalary ~ hitcr + x1 + x2 + x3 + x4)
> summary(fit1)
```

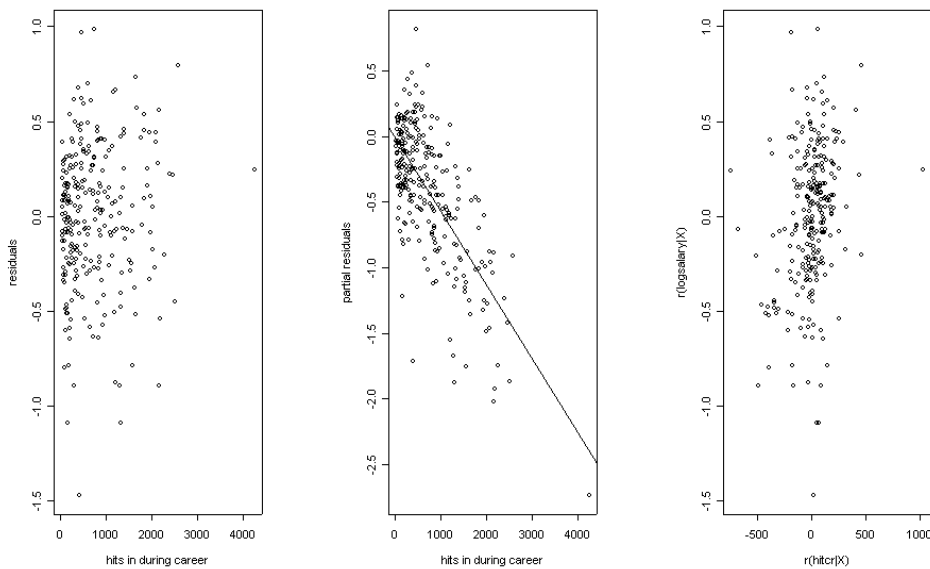
Coefficients:

	Value	Std. Error	t value	Pr(> t )
(Intercept)	3.653	0.112	32.607	0.000
hitcr	0.001	0.000	4.498	0.000
x1	0.010	0.002	4.397	0.000
x2	0.091	0.019	4.691	0.000
x3	0.299	0.018	16.602	0.000
x4	-0.107	0.017	-6.213	0.000

Residual standard error: 0.364 on 257 degrees of freedom

Multiple R-Squared: 0.833

F-statistic: 257 on 5 and 257 degrees of freedom, the p-value is 0



**Figure 4:** Residual plot, partial residual plot, and partial regression plot for `hitcr`.

Compared to the original HV's model, the improvement is substantial.

```
> anova(fit, fit1, test = "F")
Analysis of Variance Table
```

	Terms	Resid. Df	RSS	Test Df	SS	F Value	Pr(F)
1	x1 + x2 + x3 + x4	258	36.8				
2	hitcr + x1 + x2 + x3 + x4	257	34.1	+hitcr 1	2.69	20.2	0.0000104

### 3, Identifying Influential Points

There are various measures proposed to identify influential points. The function `influence.measures()` implements several commonly used ones including leverage ( $h_{ii}$ ), `dfbetas`, and `dffits`.

```
> infl <- influence.measures(fit); infl

Influence measures of
lm(formula = logsalary ~ x1 + x2 + x3 + x4) :
      dfb.1.  dfb.x1  dfb.x2  dfb.x3  dfb.x4  dffit cov.r  cook.d  hat  inf
1  5.22e-02 -0.065171 -1.18e-02  0.018120  0.083890  0.160132  1.019  5.13e-03  0.02157
2  4.90e-02  0.127549 -6.43e-02 -0.076052 -0.040994  0.172438  1.021  5.94e-03  0.02441
3  8.12e-03 -0.178305  9.11e-02 -0.085704  0.055863 -0.256568  0.950  1.30e-02  0.01460
.....
262  1.35e-02  0.001345 -9.43e-03 -0.014769  0.009398 -0.024742  1.033  1.23e-04  0.01413
263  7.80e-03 -0.006584 -2.07e-03 -0.006882  0.000576 -0.019776  1.030  7.85e-05  0.01080
```

The last column `inf` indicates using `*` whether or not a particular observation is found outlying according to all these measures.

```
> summary(influence.measures(fit))

Potentially influential observations of
lm(formula = logsalary ~ x1 + x2 + x3 + x4) :
      dfb.1_  dfb.x1  dfb.x2  dfb.x3  dfb.x4  dffit  cov.r  cook.d  hat
10  0.26  0.04  -0.21  -0.10  -0.01  -0.33  0.93_*  0.02  0.02
21  -0.05  -0.05  0.07  -0.03  0.11  0.13  1.06_*  0.00  0.04
50  -0.03  -0.02  0.04  -0.01  0.05  0.06  1.06_*  0.00  0.04
62  -0.26  -0.03  0.25  -0.01  -0.04  0.39  0.91_*  0.03  0.02
70  0.02  0.23  -0.07  -0.09  -0.11  0.34  0.91_*  0.02  0.02
92  -0.07  0.00  0.04  0.16  -0.45  -0.55_*  0.95  0.06  0.05
122  0.08  0.35  -0.26  -0.03  -0.03  -0.41_*  0.88_*  0.03  0.02
152  -0.12  -0.08  0.15  -0.09  -0.01  -0.26  0.93_*  0.01  0.01
153  0.02  0.03  -0.03  0.00  -0.03  -0.04  1.06_*  0.00  0.04
159  0.23  -0.05  -0.14  -0.13  0.03  -0.36  0.88_*  0.02  0.01
189  0.11  0.14  -0.14  -0.07  0.13  0.27  1.16_*  0.01  0.13_*
198  -0.02  -0.10  0.07  -0.03  0.14  0.16  1.07_*  0.01  0.06
220  0.03  0.15  -0.09  -0.21  0.17  -0.39  0.76_*  0.03  0.01
249  0.08  0.05  -0.08  -0.02  0.07  0.15  1.07_*  0.00  0.06
252  -0.06  -0.43  0.14  0.29  0.08  -0.60_*  0.96  0.07  0.06
```

One may consider refitting the model without including these outliers and then compare the model fits.

## 4, Checking Multicollinearity

To check multicollinearity, compute the variance inflation factors (VIF). It seems that multicollinearity is not a problem for HV's model, except for the intercept.

```
temp <- rep(1, dim(baseball)[1]) # create a vector of all 1's.
VIF0 <- 1 / (1 - summary(lm(temp ~ x1 + x2 + x3 + x4 -1))$r.squared)
VIF1 <- 1 / (1 - summary(lm(x1 ~ x2 + x3 + x4))$r.squared)
VIF2 <- 1 / (1 - summary(lm(x2 ~ x1 + x3 + x4))$r.squared)
VIF3 <- 1 / (1 - summary(lm(x3 ~ x1 + x2 + x4))$r.squared)
VIF4 <- 1 / (1 - summary(lm(x4 ~ x1 + x2 + x3))$r.squared)
VIF <- c(VIF0, VIF1, VIF2, VIF3, VIF4)

> VIF
[1] 23.3500  2.3326  2.1801  1.4773  1.7657

> mean(VIF)
[1] 6.2212
```