

chapter 10

Regression analysis II

Instructor: Li, Han

Contents

- ❑ Regression diagnostics
- ❑ Variance inflation factor
- ❑ Outlier/leveraged/influential points

Linear model

$$Y_i = \beta_o + \beta_1 X_{1i} + \dots + \beta_k X_{ki} + \varepsilon_i, \quad i = 1, \dots, n.$$

Model assumptions:

- ❑ Independence
- ❑ Normality
- ❑ Homoscedasticity

Ordinary Least Squares (OLS) finds parameters to minimize the sum of squared residuals.

$$\sum_{i=1}^n (Y_i - \hat{Y}_i)^2 = (Y_i - \hat{\beta}_o - \hat{\beta}_1 X_{1i} - \dots - \hat{\beta}_k X_{ki})^2 = \sum_{i=1}^n \varepsilon_i^2.$$

In R, the basic function for fitting a linear model is `lm()`.

```
fit <- lm(y ~ x1+x2+...+xk, data=dataframe)
```

Regression results

Table 8.3 Other functions that are useful when fitting linear models

Function	Action
<code>summary()</code>	Displays detailed results for the fitted model
<code>coefficients()</code>	Lists the model parameters (intercept and slopes) for the fitted model
<code>confint()</code>	Provides confidence intervals for the model parameters (95 percent by default)
<code>fitted()</code>	Lists the predicted values in a fitted model
<code>residuals()</code>	Lists the residual values in a fitted model
<code>anova()</code>	Generates an ANOVA table for a fitted model, or an ANOVA table comparing two or more fitted models
<code>vcov()</code>	Lists the covariance matrix for model parameters
<code>AIC()</code>	Prints Akaike's Information Criterion
<code>plot()</code>	Generates diagnostic plots for evaluating the fit of a model
<code>predict()</code>	Uses a fitted model to predict response values for a new dataset

Regression diagnostics

Table 8.4 Useful functions for regression diagnostics (`car` package)

Function	Purpose
<code>qqPlot()</code>	Quantile comparisons plot
<code>durbinWatsonTest()</code>	Durbin–Watson test for autocorrelated errors
<code>crPlots()</code>	Component plus residual plots
<code>ncvTest()</code>	Score test for nonconstant error variance
<code>spreadLevelPlot()</code>	Spread-level plot
<code>outlierTest()</code>	Bonferroni outlier test
<code>avPlots()</code>	Added variable plots
<code>influencePlot()</code>	Regression influence plot
<code>scatterplot()</code>	Enhanced scatter plot
<code>scatterplotMatrix()</code>	Enhanced scatter plot matrix
<code>vif()</code>	Variance inflation factors

Example 1 (fit the model and check model assumption)

```
library(car)
states <- as.data.frame(state.x77[,c("Murder", "Population", "Illiteracy",
"Income", "Frost")])
scatterplotMatrix(states, spread=FALSE, lty=2, main="Scatterplot
Matrix")
cor(states)
fit <- lm(Murder ~ Population + Illiteracy + Income + Frost, data=states)
```

#calibration plot

plot(states\$Murder,fitted(fit))

#residual plot

plot(residuals(fit),fitted(fit))

#checking model assumptions

qqplot(fit)

durbinWatsonTest(fit)

ncvTest(fit)

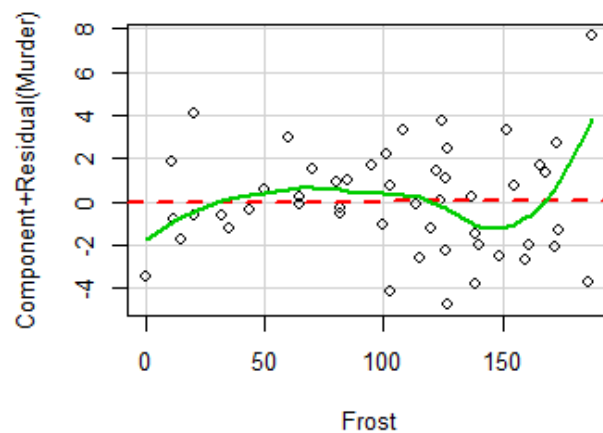
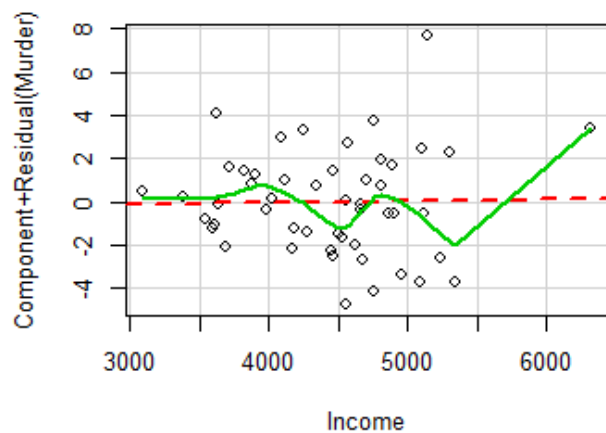
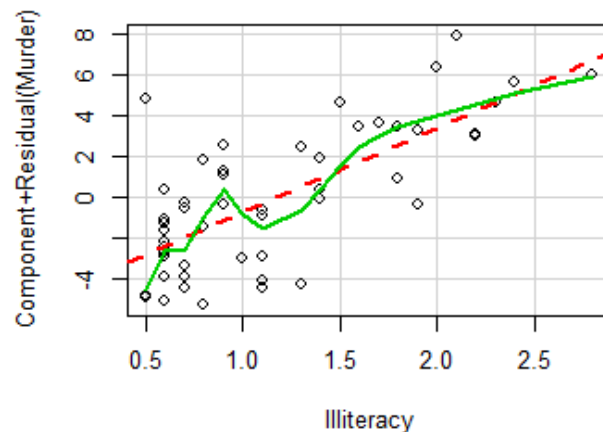
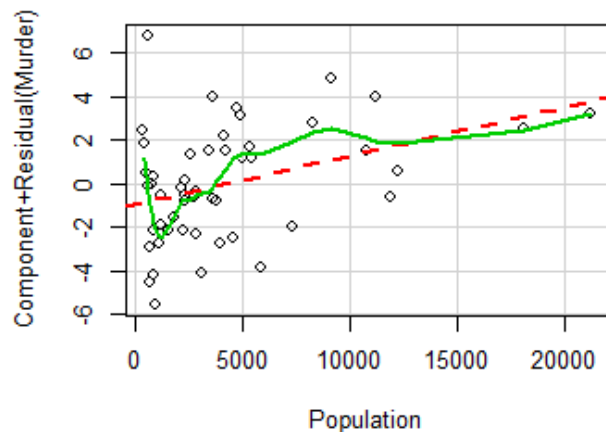
Check for linearity

To check whether the response variable and the explanatory variable have linear relationship after adjusting the effect of other variables, we could plot

$$\varepsilon_i + \hat{\beta}_j X_{ji} \text{ versus } X_{ji}$$

It is called the component plus residual plots (also known as partial residual plots).

Component + Residual Plots



Multicollinearity

Multicollinearity is a fact that the explanatory variables are highly correlated. It will result in unreliable parameter estimation and confidence intervals, though it does not violate the model assumptions.

Variance Inflation Factor (VIF)

Multicollinearity can be detected by the variance inflation factor (VIF). For any predictor variable, the square root of the VIF indicates the degree to which the standard deviation for that variable's regression parameter is expanded relative to a model with uncorrelated predictors (hence the name).

VIF values are provided by the `vif()` function in the `car` package. As a general rule, $\text{vif} > 4$ indicates a multicollinearity problem.

Example 2 (VIF)

`vif(fit)`

> Population	Illiteracy	Income	Frost
1.25	2.17	1.35	2.08

Unusual observations

A comprehensive regression analysis will also include a screening for unusual observations — namely outliers, high-leverage observations, and influential observations.

These are data points that warrant further investigation, either because they are different than other observations in some way, or because they exert a disproportionate amount of influence on the results.

Outliers

Outliers are observations that are not predicted well by the model. They have either unusually large positive or negative residuals.

The `outlierTest()` function reports the Bonferroni adjusted p-value for the largest absolute studentized residual.

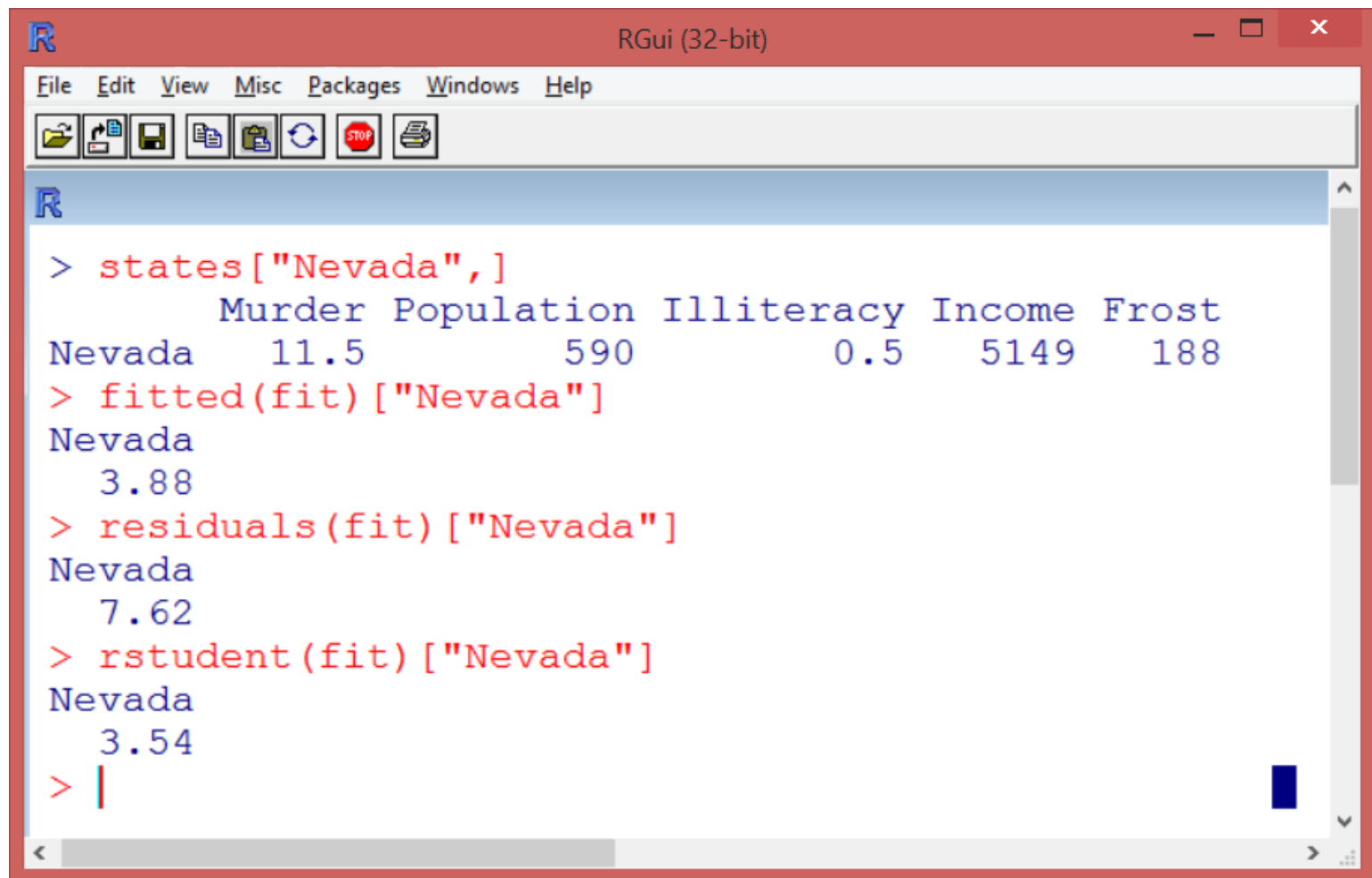
Example 3 (Outlier detection)

```
library(car)
```

```
outlierTest(fit)
```

```
> rstudent unadjusted p-value Bonferonni p
```

Nevada	3.5	0.00095	0.048
--------	-----	---------	-------



The screenshot shows the RGui (32-bit) window with a menu bar (File, Edit, View, Misc, Packages, Windows, Help) and a toolbar. The console displays the following R commands and their output:

```
> states["Nevada",]  
      Murder Population Illiteracy Income Frost  
Nevada   11.5         590         0.5   5149   188  
> fitted(fit) ["Nevada"]  
Nevada  
   3.88  
> residuals(fit) ["Nevada"]  
Nevada  
   7.62  
> rstudent(fit) ["Nevada"]  
Nevada  
   3.54  
> |
```

High leveraged points

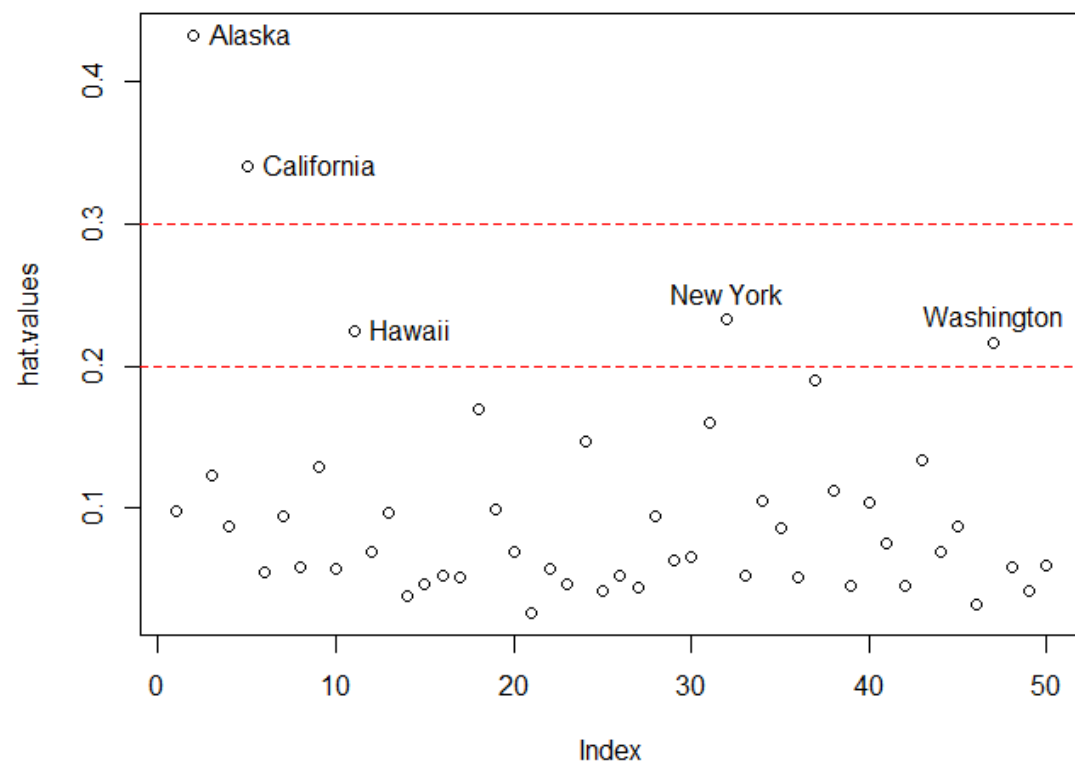
Observations with high leverage are identified through the hat statistic ($h_{ii} = x_i'(X'X)^{-1}x_i$). For a given dataset, the average hat value is p/n , where p is the number of parameters estimated in the model (including the intercept) and n is the sample size.

Roughly speaking, an observation with a hat value greater than $2p/n$ or $3p/n$ should be examined.

Example 4 (high leveraged points)

```
hat.values=hatvalues(fit)
p <- length(coefficients(fit))
n <-length(fitted(fit))
plot(hat.values, main="Index Plot of Hat Values")
abline(h=c(2,3)*p/n, col="red", lty=2)
identify(1:n, hatvalues(fit), names(hat.values))
```

Index Plot of Hat Values



Influential observations

Influential observations are observations that have a disproportionate impact on the values of the model parameters. Imagine finding that your model changes dramatically with the removal of a single observation.

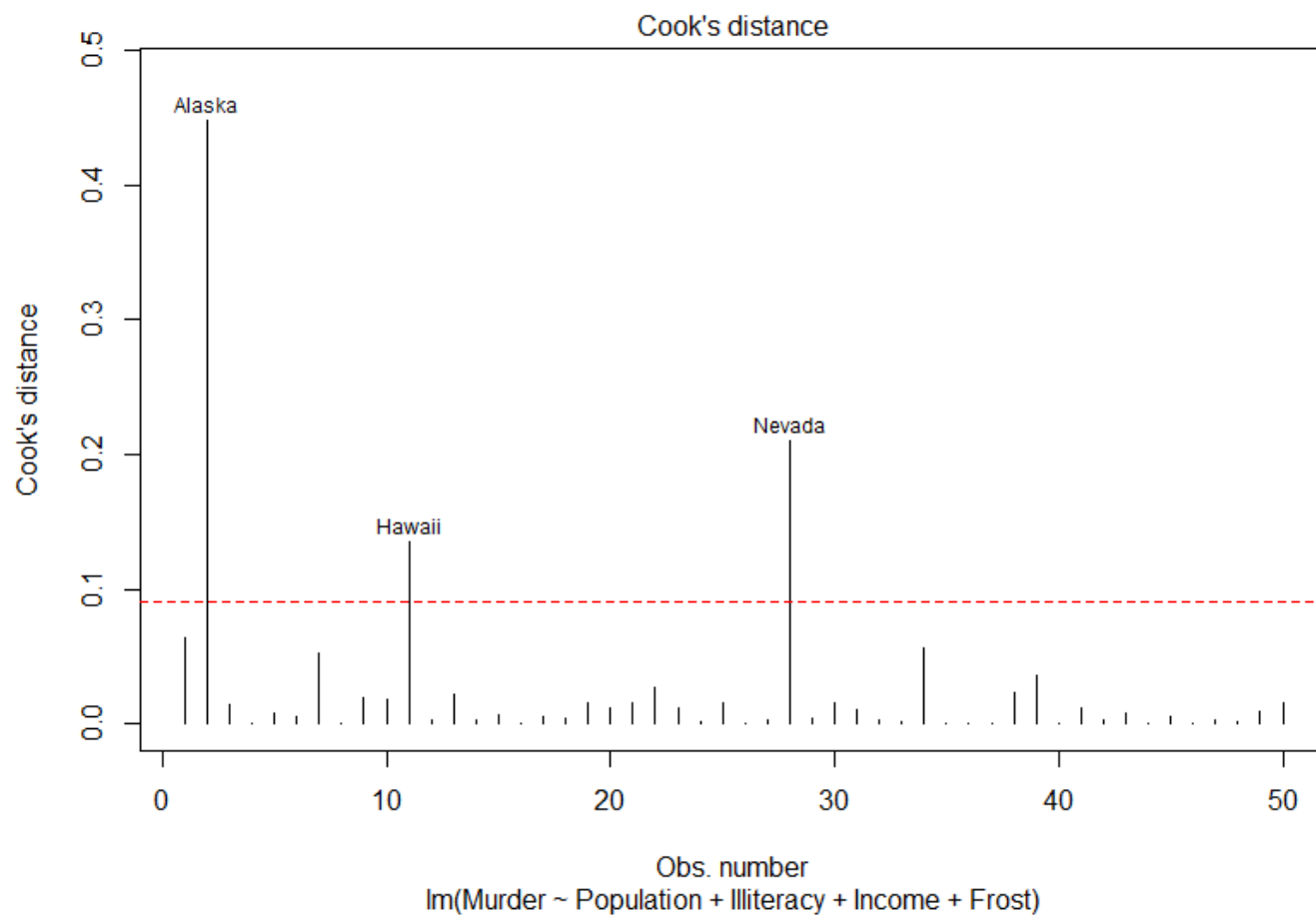
There are two methods for identifying influential observations: **Cook's distance**, or D statistic and **added variable plots**.

Cook's distance

Roughly speaking, Cook's D values greater than $4/(n-k-1)$, where n is the sample size and k is the number of predictor variables, indicate influential observations.

Example 5 (Cook's distance)

```
cutoff <- 4/(nrow(states)-length(fit$coefficients))  
plot(fit, which=4, cook.levels=cutoff)  
abline(h=cutoff, lty=2, col="red")
```



Added-variable plots

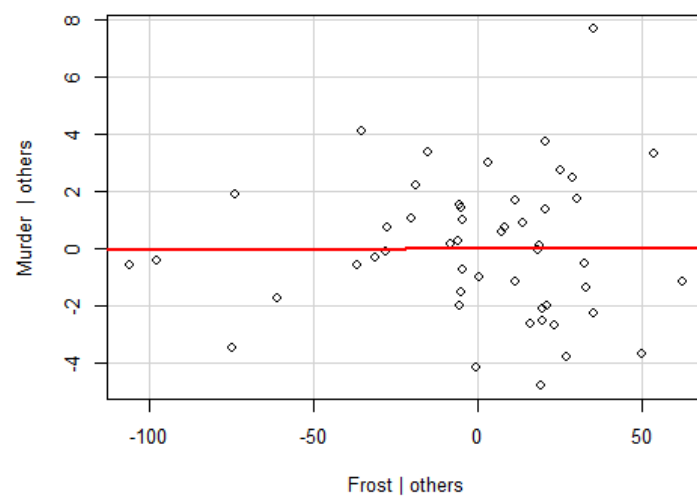
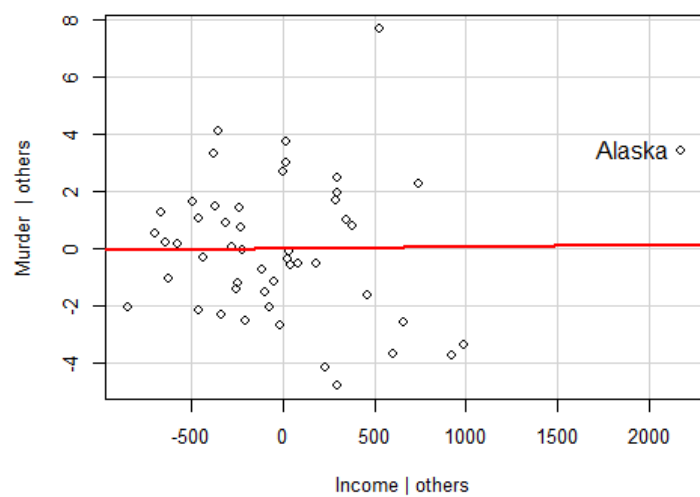
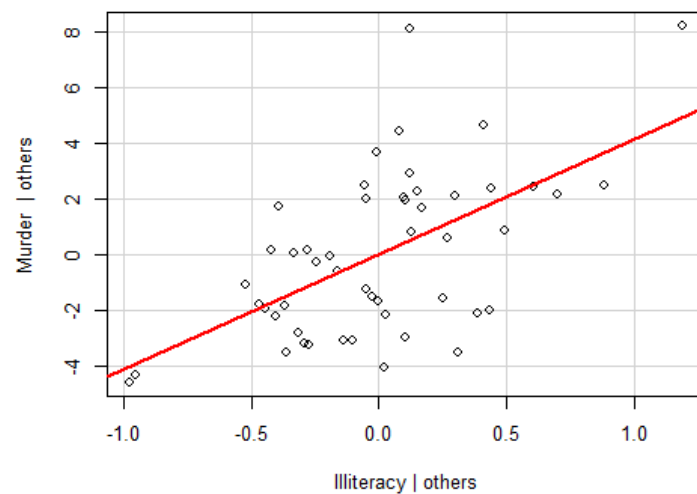
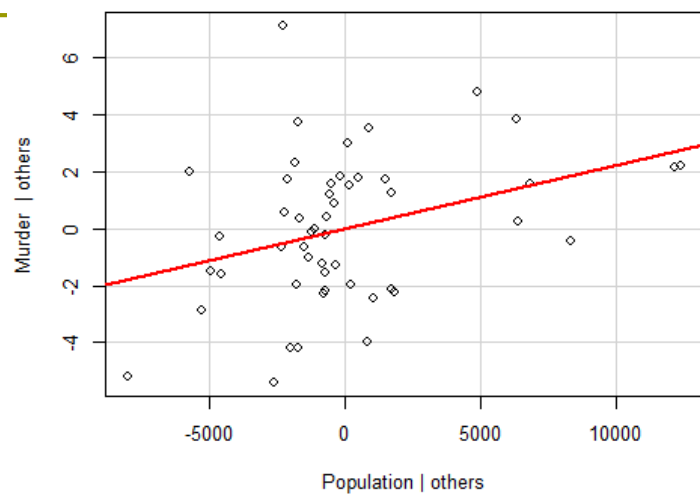
Added-variable plots provide information on how these observations affect the model. For each predictor X_k , plot the residuals from regressing the response variable on the other $k-1$ predictors versus the residuals from regressing X_k on the other $k-1$ predictors.

Example 6(Add variable plots)

```
library(car)
```

```
avPlots(fit, ask=FALSE, id.method="identify")
```


Added-Variable Plots



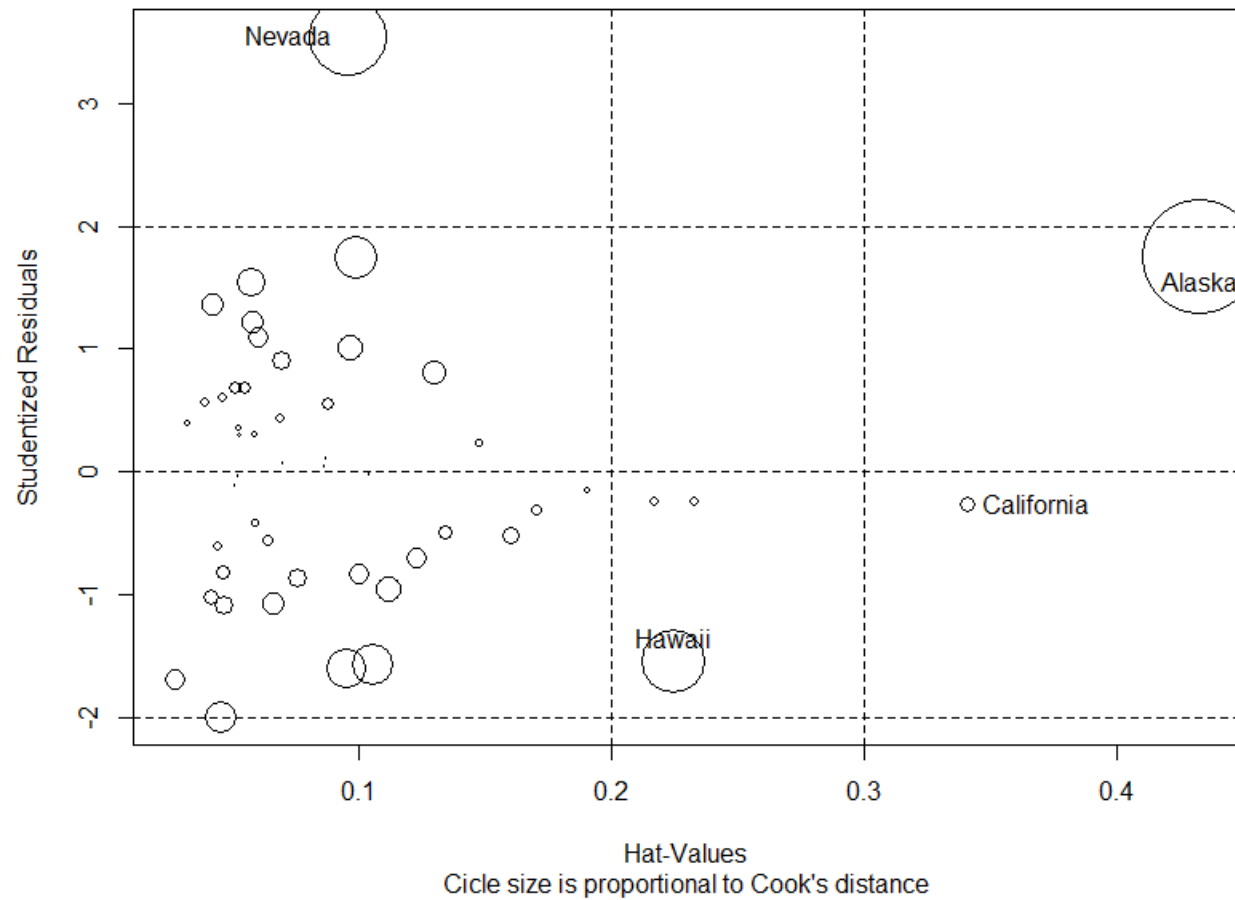
Unusual points in one plot

You can combine the information from outlier, leverage, and influence plots into one highly informative plot using the `influencePlot()` function from the `car` package.

Example 7 (Influence Plot)

```
influencePlot(fit, id.method="identify", main="Influence Plot",  
sub="Circle size is proportional to Cook's distance")
```

Influence Plot



Summary

In this session, we have learned

- ❑ check the linearity of the variables
- ❑ check the variance inflation factor
- ❑ detect outlier/leveraged/influential points