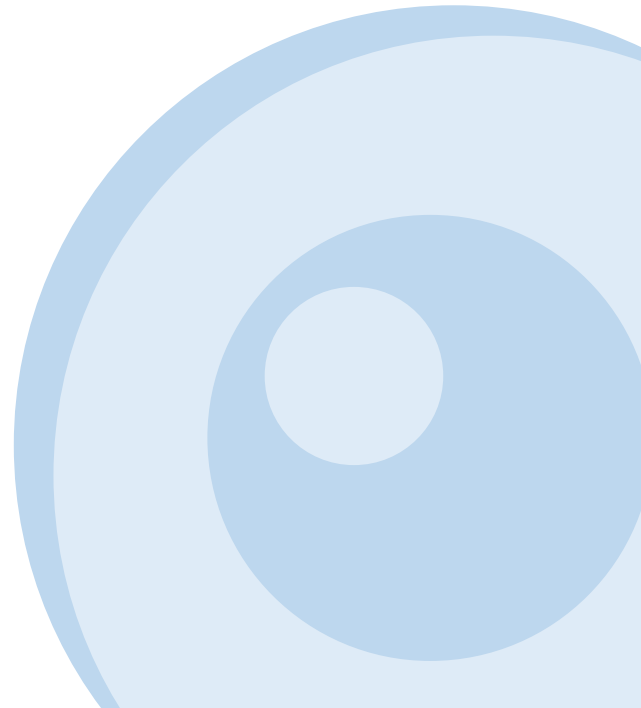


PSYC 193R:

Data Analysis & Visualization with R

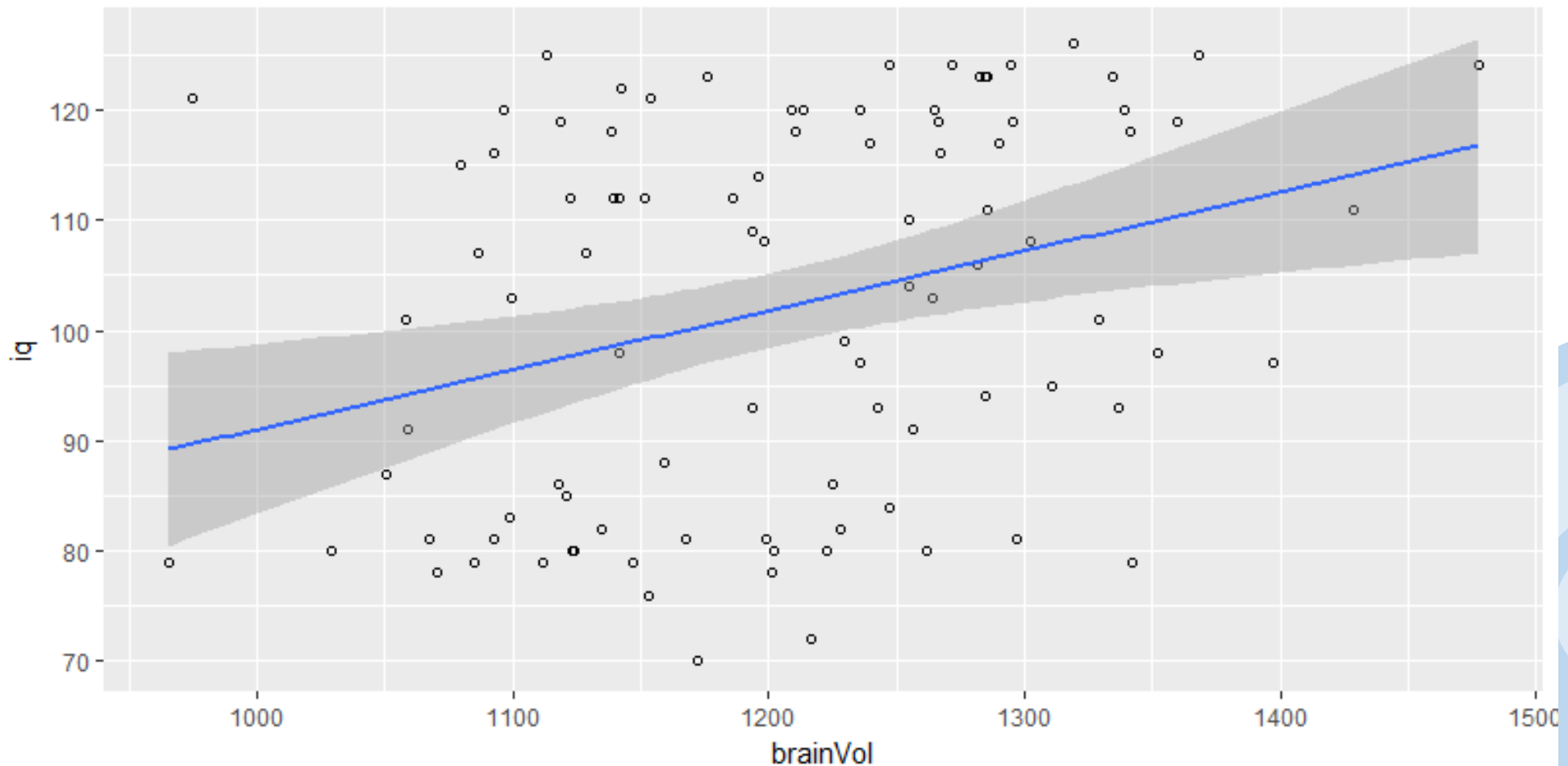
Lecture 14: Multiple Regression

Jonas S.-H. Lau [silau -et- ucsd.edu]
psyc193r.ucsd.edu



Simple Regression: Change in Notations

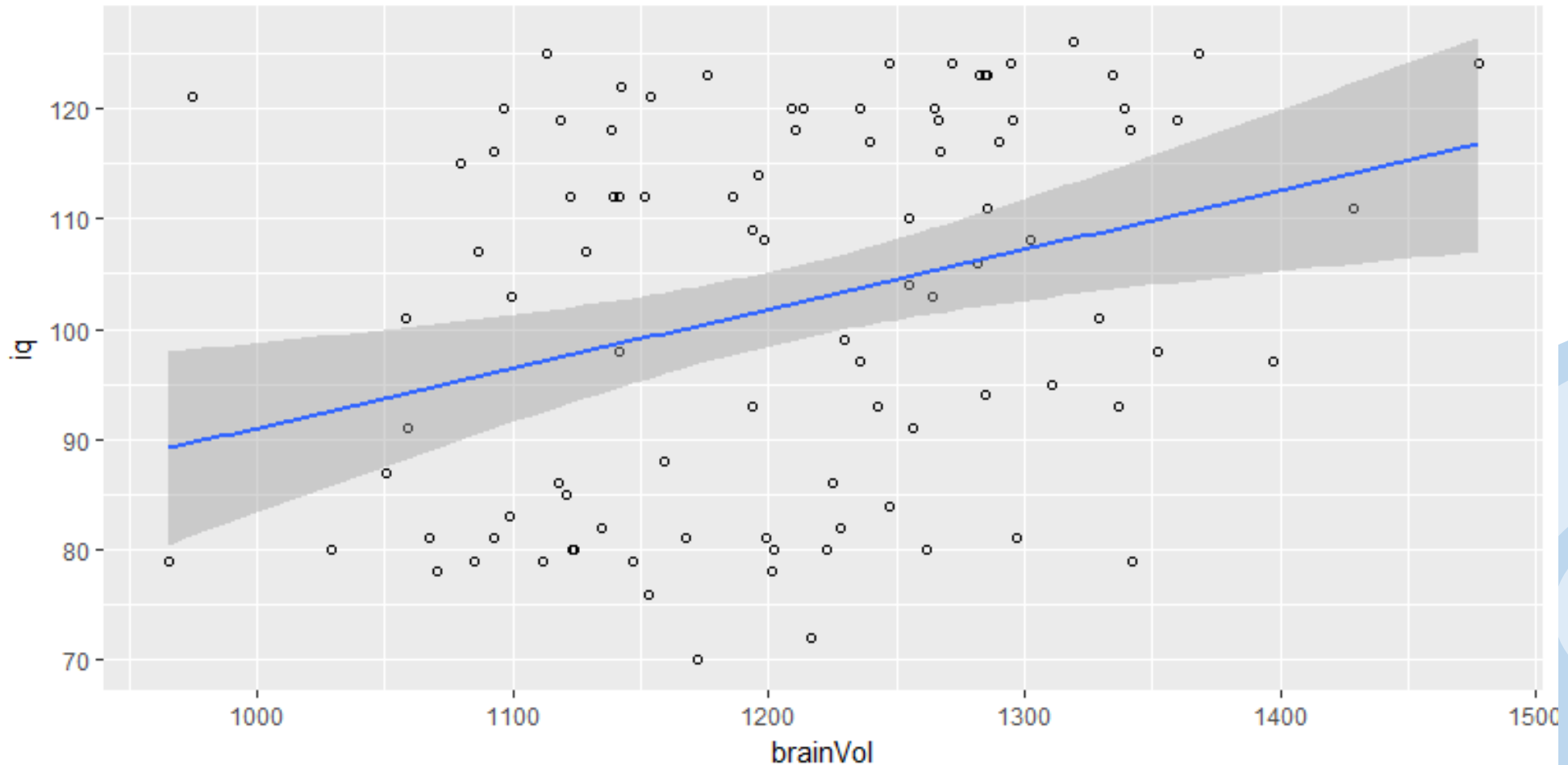
$$\hat{Y} = bX + a$$



Simple Regression: Change in Notations

$$\hat{Y} = b_0 + b_1 X_1$$

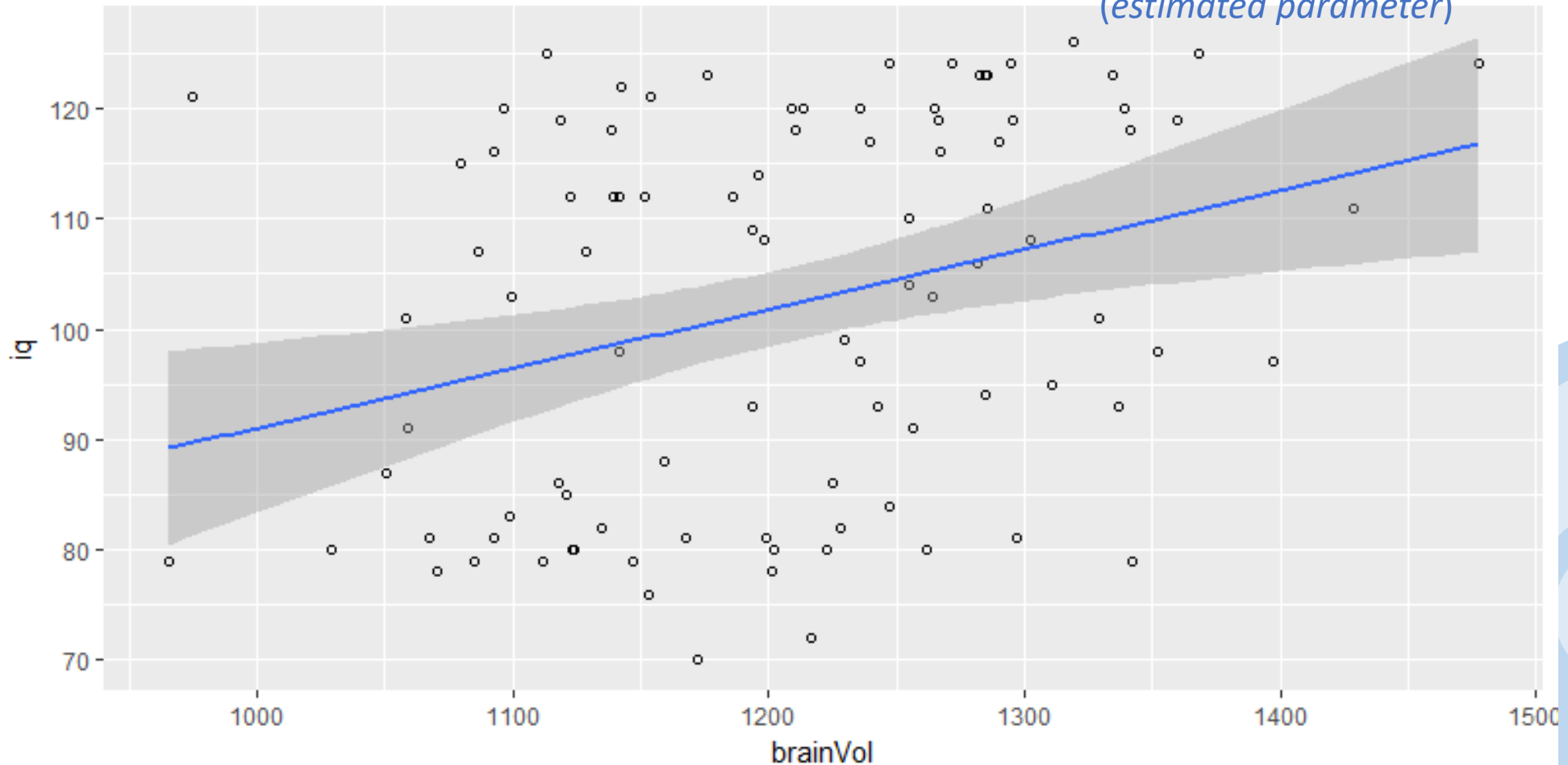
From data



Simple Regression: Change in Notations

$$\hat{Y} = b_0 + b_1 X_1$$

*Coefficient
(estimated parameter)*



Goals in Simple Regression

1. Test if a *linear* relationship between two variables exists
2. Describe the linear relationship in mathematical form
 - That would help us make predictions with *new* data

$$\hat{Y} = b_0 + b_1 X_1$$

Steps:

- Assumptions checking: Linearity / Normality / Outliers
- Correlation Coefficient / Hypothesis Testing
- Variance explained by the model / Effect Size
- Regression Line
- Predictions
- Visualization
- Residuals / Post-fit checking: Residual Plot



The world is more complicated

Data from: <http://archive.ics.uci.edu/ml/datasets/Wine>

“These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.”

The wine were then rated by wine experts. We're going to look at the red wine only.

- Fixed Acidity
- Volatile Acidity
- Citric Acid
- Residual Sugar
- Chlorides
- Free Sulfur Dioxide
- Total Sulfur Dioxide
- Density
- pH
- Sulphates
- Alcohol
- Quality



Wine Quality: The Design

- What was the research question? What was the target population?
- Was it an experiment, quasi-experiment, or an observational study?
- What were the IV(s) and DV(s)?
- How were the IV(s) and DV(s) operationalized?
(also, scale of measurement?)
- Was it a between / within-subjects design?
- What are the expected results?
If the manipulation does not work:
If the manipulation works:



Wine Quality: Goals in Multiple Regression

1. Test if a *linear* relationship between multiple predictor variables and the dependent variable
2. Search for the simplest (parsimonious) model that describes the relationship
 - Use the fewest predictors possible
3. Describe the linear relationship in mathematical form
 - That would help us make predictions with *new* data

$$\hat{Y} = b_0 + b_1 X_1$$



Wine Quality: Goals in Multiple Regression

1. Test if a *linear* relationship between multiple predictor variables and the dependent variable
2. Search for the simplest (parsimonious) model that describes the relationship
 - Use the fewest predictors possible
3. Describe the linear relationship in mathematical form
 - That would help us make predictions with *new* data

$$\hat{Y} = b_0 + b_1 X_1$$



Wine Quality: Goals in Multiple Regression

1. Test if a *linear* relationship between multiple predictor variables and the dependent variable
2. Search for the simplest (parsimonious) model that describes the relationship
 - Use the fewest predictors possible
3. Describe the linear relationship in mathematical form
 - That would help us make predictions with *new* data

$$\hat{Y} = b_0 + b_1 X_1$$

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \dots$$

Wine Quality: Goals in Multiple Regression

1. Test if a *linear* relationship between multiple predictor variables and the dependent variable
2. Search for the simplest (parsimonious) model that describes the relationship
 - Use the fewest predictors possible
3. Describe the linear relationship in mathematical form
 - That would help us make predictions with *new* data

$$\hat{Y} = b_0 + b_1 X_1$$

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \dots$$

*Predictors for training
the model (from data)*

Wine Quality: Goals in Multiple Regression

1. Test if a *linear* relationship between multiple predictor variables and the dependent variable
2. Search for the simplest (parsimonious) model that describes the relationship
 - Use the fewest predictors possible
3. Describe the linear relationship in mathematical form
 - That would help us make predictions with *new* data

$$\hat{Y} = b_0 + b_1 X_1$$

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \dots$$

*Coefficient learned
during training*



Import Data

- Download the data file:
<http://archive.ics.uci.edu/ml/datasets/wine>
- Tell R where your data file is:
Session → Set Working Directory → Choose Directory
`setwd()`
- Check the *comma separated values* (csv) file with MS Excel to ensure correct format
- Import the file into R as a data frame, saving it to “intelligenceData” variable
`... = read.csv("file_name.csv")`

Data Structure (long data frame)

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

psyc193r_l14_wine.R* psyc193r_l14_multiple-regression.Rmd wineData*

Filter

	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
1	7.4	0.700	0.00	1.90	0.076	11	34	0.9978	3.51	0.56	9.4	5
2	7.8	0.880	0.00	2.60	0.098	25	67	0.9968	3.20	0.68	9.8	5
3	7.8	0.760	0.04	2.30	0.092	15	54	0.9970	3.26	0.65	9.8	5
4	11.2	0.280	0.56	1.90	0.075	17	60	0.9980	3.16	0.58	9.8	6
5	7.4	0.700	0.00	1.90	0.076	11	34	0.9978	3.51	0.56	9.4	5
6	7.4	0.660	0.00	1.80	0.075	13	40	0.9978	3.51	0.56	9.4	5
7	7.9	0.600	0.06	1.60	0.069	15	59	0.9964	3.30	0.46	9.4	5
8	7.3	0.650	0.00	1.20	0.065	15	21	0.9946	3.39	0.47	10.0	7
9	7.8	0.580	0.02	2.00	0.073	9	18	0.9968	3.36	0.57	9.5	7
11	6.7	0.580	0.08	1.80	0.097	15	65	0.9959	3.28	0.54	9.2	5
13	5.6	0.615	0.00	1.60	0.089	16	59	0.9943	3.58	0.52	9.9	5
15	8.9	0.620	0.18	3.80	0.176	52	145	0.9986	3.16	0.88	9.2	5
16	8.9	0.620	0.19	3.90	0.170	51	148	0.9986	3.17	0.93	9.2	5
17	8.5	0.280	0.56	1.80	0.092	35	103	0.9969	3.30	0.75	10.5	7
19	7.4	0.590	0.08	4.40	0.086	6	29	0.9974	3.38	0.50	9.0	4
21	8.9	0.220	0.48	1.80	0.077	29	60	0.9968	3.39	0.53	9.4	6
22	7.6	0.390	0.31	2.30	0.082	23	71	0.9982	3.52	0.65	9.7	5
23	7.9	0.430	0.21	1.60	0.106	10	37	0.9966	3.17	0.91	9.5	5

Showing 1 to 18 of 1,517 entries

Data Formatting / Processing / Cleaning

- Factorize ____

```
... = factor(wineData$ ____ )
```

- Coerce ____ to a numeric variable

```
... = as.numeric(wineData$ ____ )
```

- Maybe doing it in a for loop?

```
For (i in 1: ____ ){  
  wineData$ ... = ...  
}
```

Check for outliers (histogram)

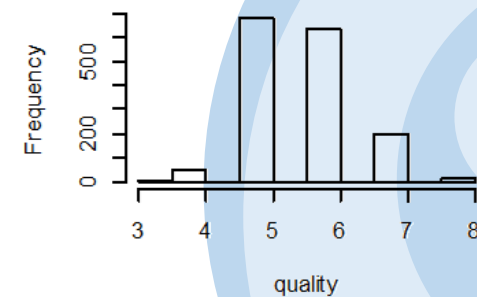
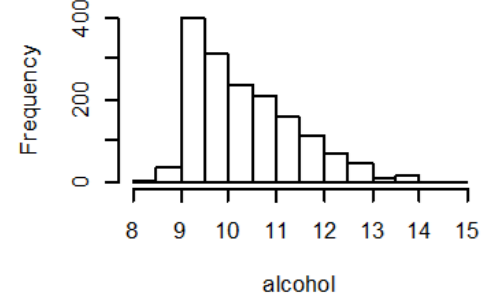
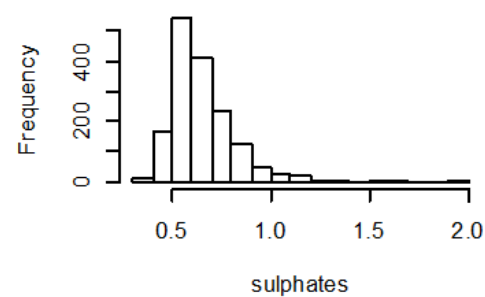
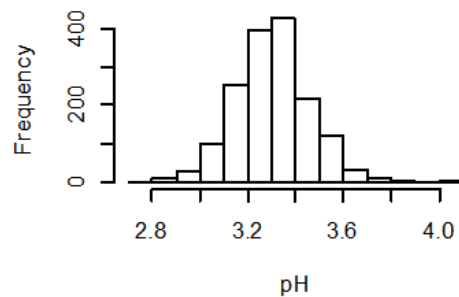
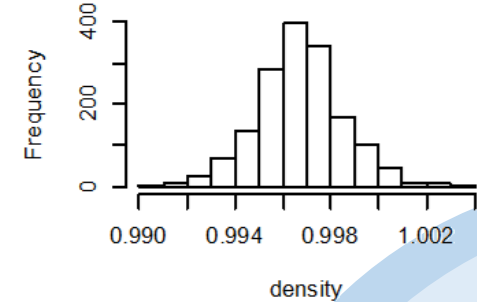
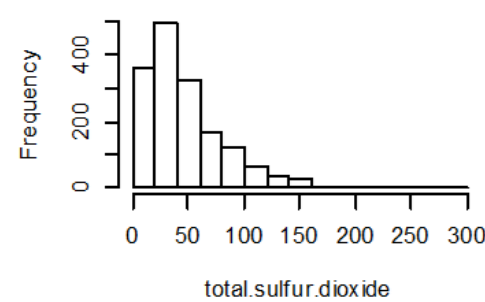
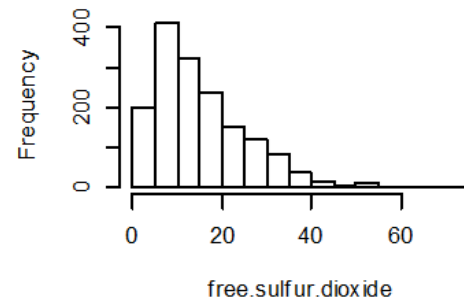
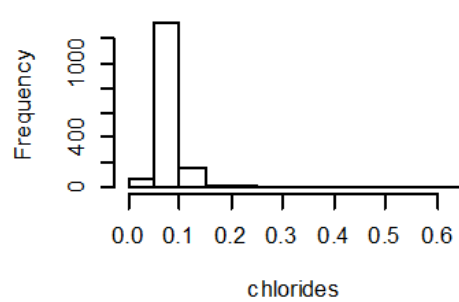
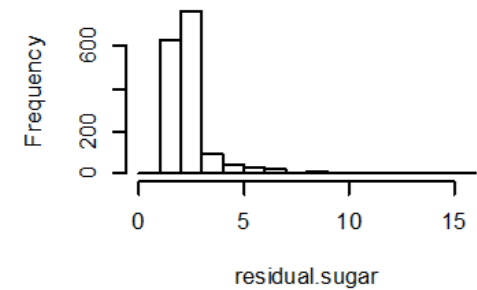
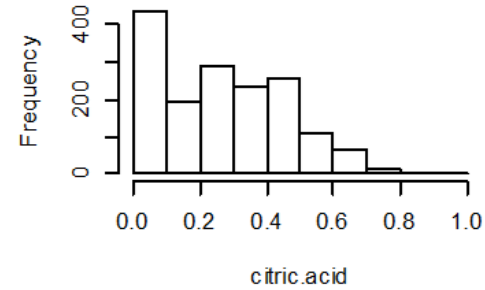
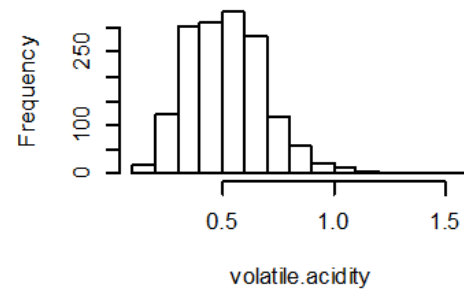
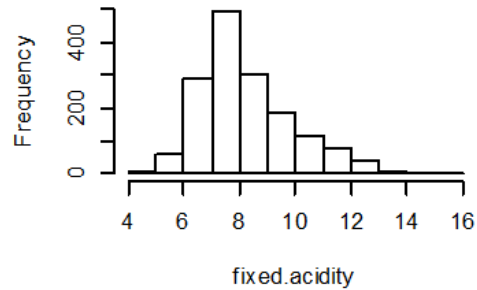
- We want to draw all the histograms on one page, and there are 12 of them. So we can make it a grid of histograms

```
par(mfrow=c(3,4))
```

- Use a for loop to draw histograms of the variables

```
for (i in 1:ncol(wineData)){  
  hist(wineData[ __ , __ ], # draw the i-th column  
       # include the name of the variable as the x-axis label  
       xlab = names(wineData)[i],  
       # turn off the title  
       main = NULL)  
}  
  
# switch back to the 1 x 1 drawing panel  
par(mfrow=c(1,1))
```

Check for outliers (histogram)



Check for outliers (linearity between IV's and DV)

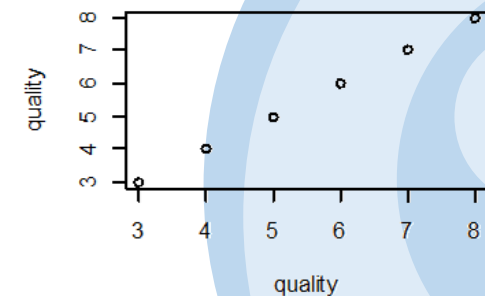
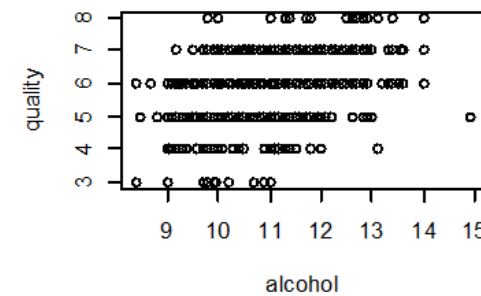
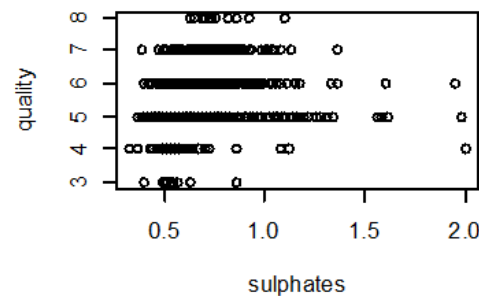
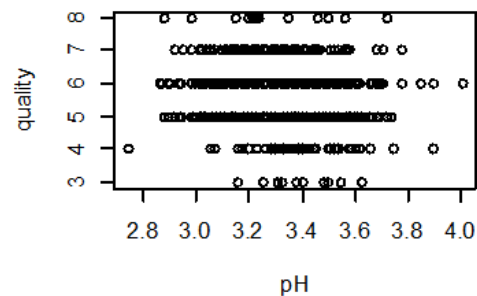
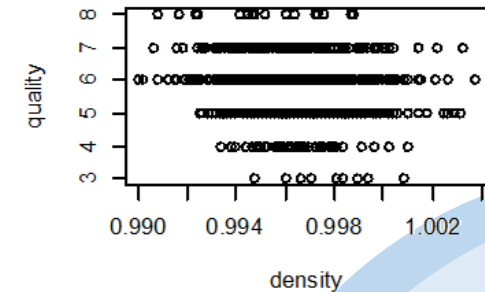
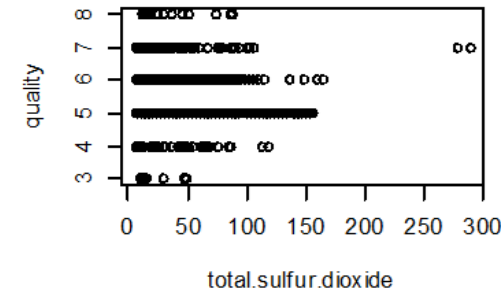
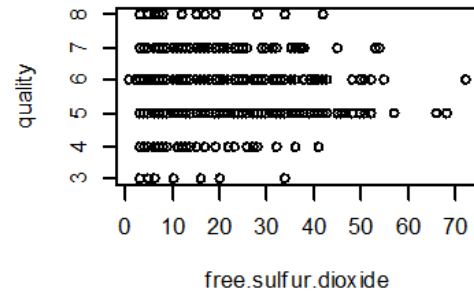
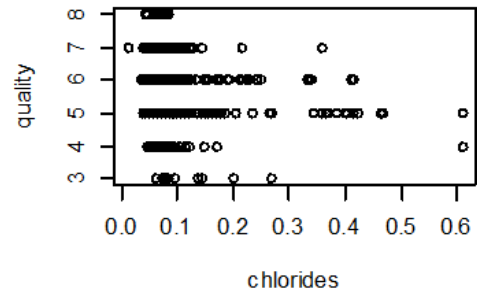
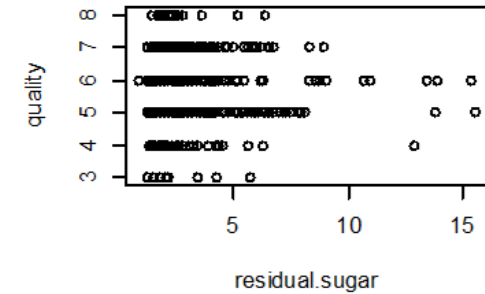
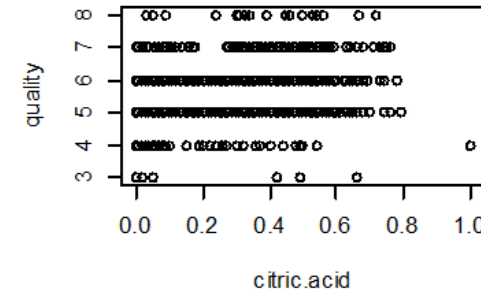
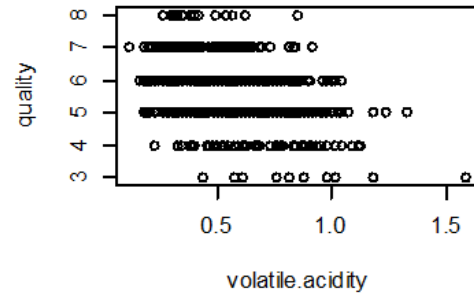
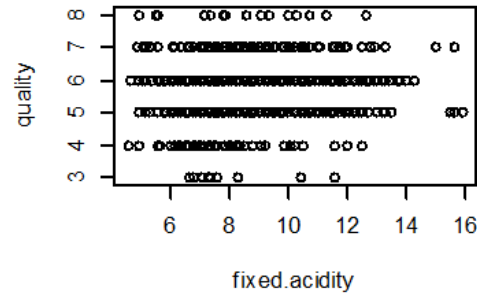
- We want to draw all the histograms on one page, and there are 12 of them. So we can make it a grid of histograms

```
par(mfrow=c(3,4))
```

- Use a for loop to draw histograms of the variables

```
for (i in ____){  
  # plot the scatterplot between i-th IV and DV  
  plot(wineData[ __ , __ ], wineData[ __ , __ ],  
        # include the name of the variable as the x-axis label  
        xlab = names(wineData)[ __ ], ylab = names(wineData)[ __ ],  
        # turn off the title  
        main = NULL)  
}  
  
# switch back to the 1 x 1 drawing panel  
par(mfrow=c(1,1))
```

Check for outliers (linear relationship)



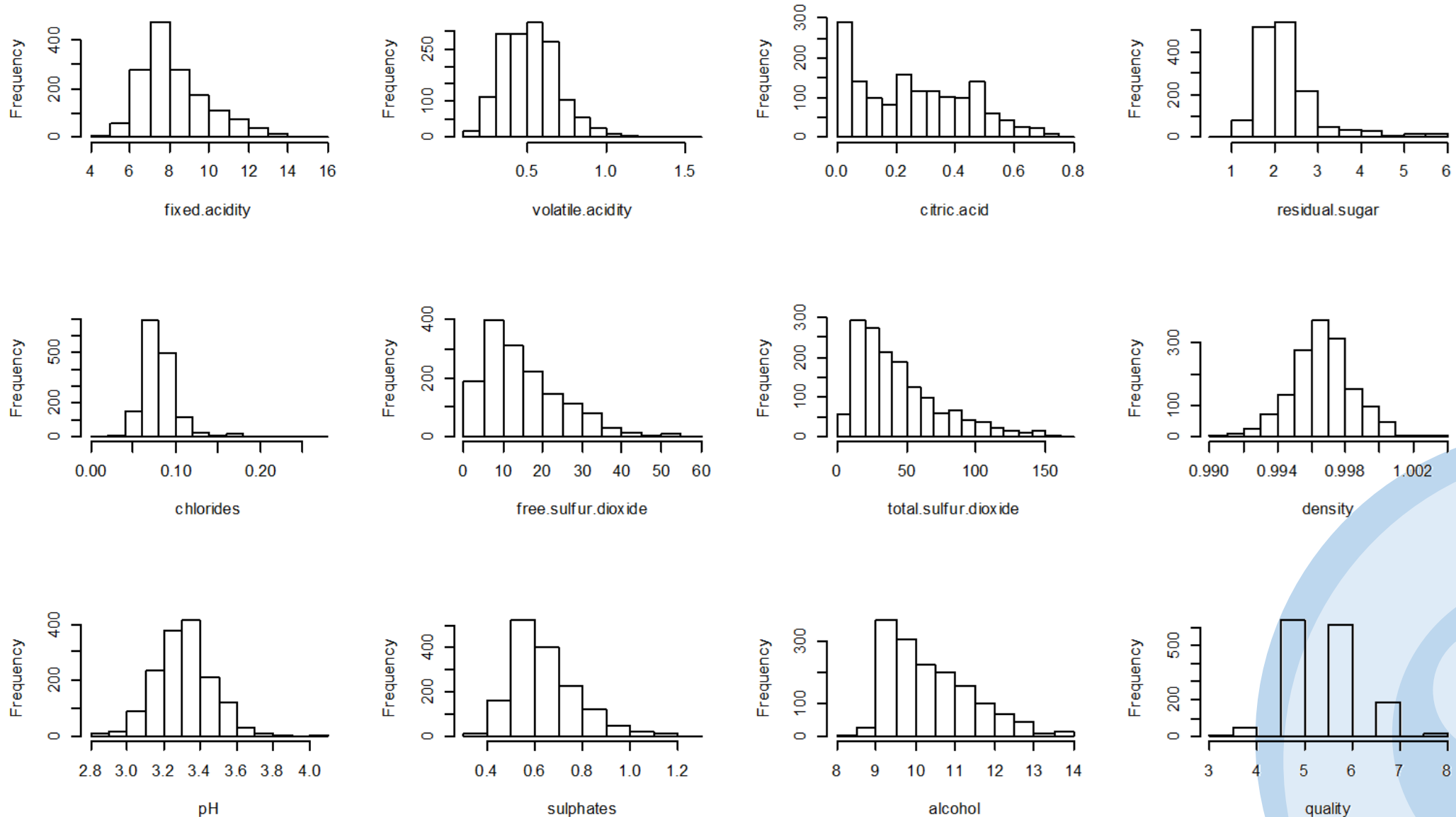
Remove outliers

- For each variable: remove obvious outliers (usually $X > M_X \pm 3SD$)
- Use subset [], e.g.,

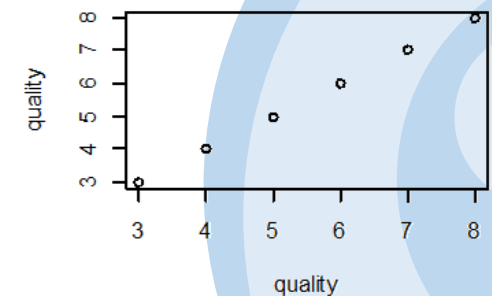
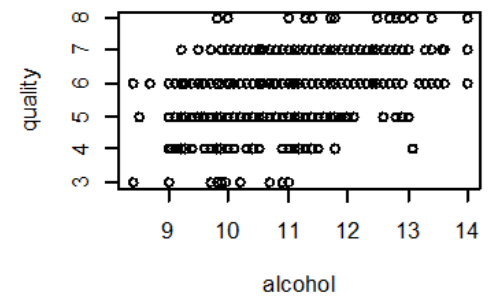
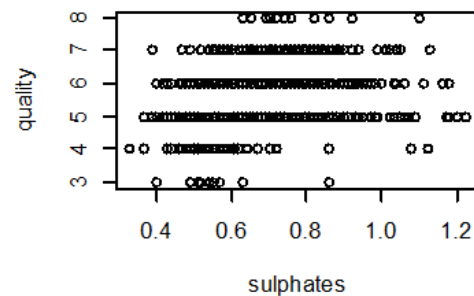
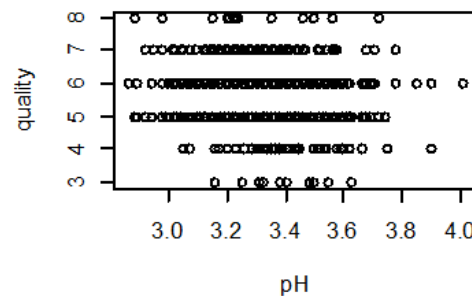
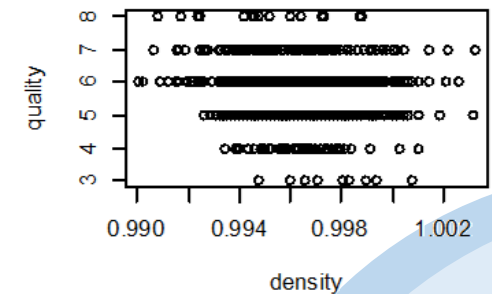
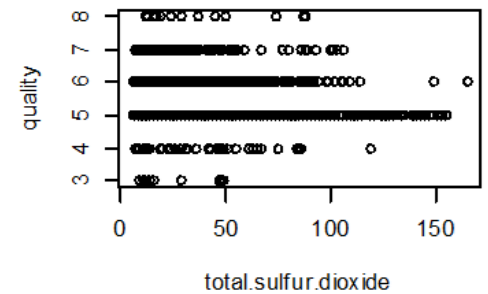
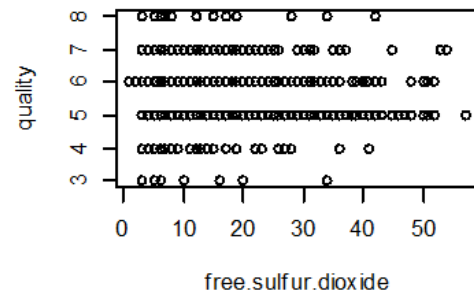
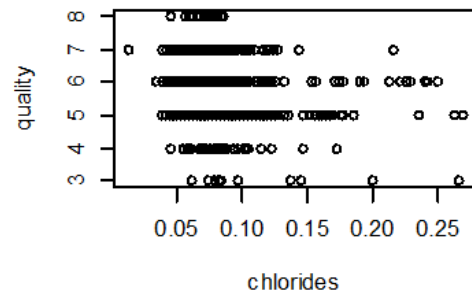
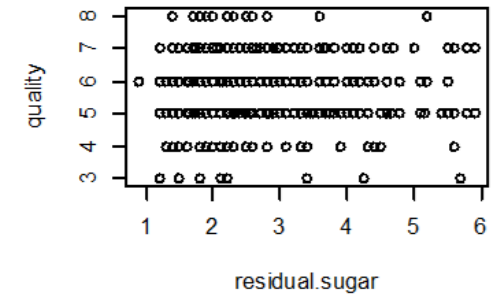
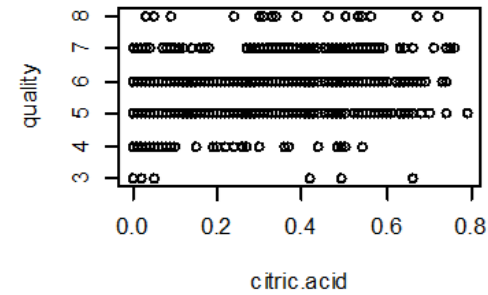
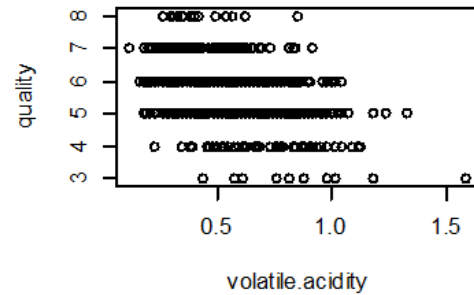
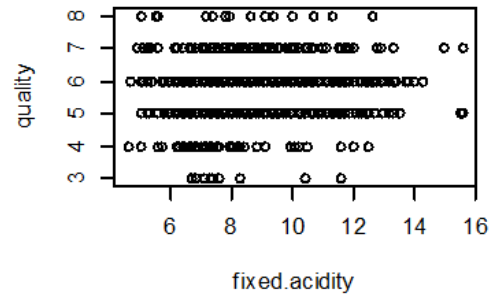
```
wineData = wineData[wineData$residual.sugar < 10,]  
...
```



Ok... Let's check for outliers again



Ok... Let's check again (linearity between IV's and DV)



Picking the relevant predictor variables

1. Predictor selection should be hypothesis driven
 - The resulting mathematical equation should be interpretable
2. With multiple regression, it is tempting to include all possible things you can measure
 - Testing too many variables inflates Type I error
 - It loses simplicity / parsimony
3. Predictors should be uniquely contributing to the variance of the dependent variable
 - Correlation between individual predictor variable and the dependent variable should be "*high*"
 - Correlation between predictor variables should be "*low*"

Picking the relevant predictor variables

1. Predictor selection should be hypothesis driven
 - The resulting mathematical equation should be interpretable
2. With multiple regression, it is tempting to include all possible things you can measure
 - Testing too many variables inflates Type I error
 - It loses simplicity / parsimony
3. Predictors should be uniquely contributing to the variance of the dependent variable
 - Correlation between individual predictor variable and the dependent variable should be "*high*"
 - Correlation between predictor variables should be "*low*"

The Correlation Matrix: `cor()`

Showing 1 to 12 of 12 entries

The Correlation Matrix: `cor()`

Showing 1 to 12 of 12 entries

The Correlation Matrix: `cor()`

Showing 1 to 12 of 12 entries

Removing variables

- Save only the relevant predictors + dv in a new data frame name it as `wineReduced`

```
wineReduced = ...
```

- Check if you have the right variables in the new data frame

```
head(wineReduce)
```



Nested Models

Intercept model: $\hat{Y} = b_0$

Full model: $\hat{Y} = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4$



Nested Models

Intercept model: $\hat{Y} = b_0$

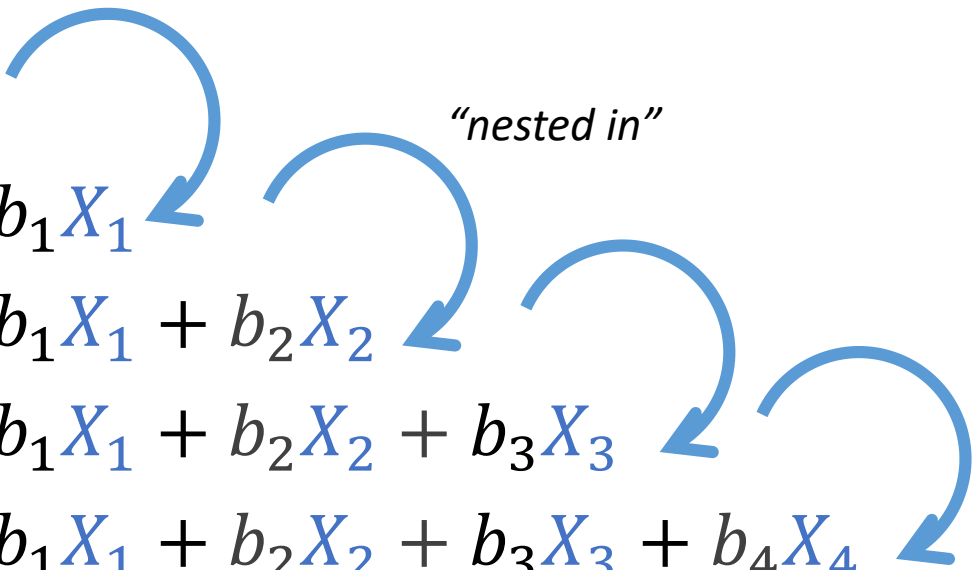
Model 1: $\hat{Y} = b_0 + b_1X_1$

Model 2: $\hat{Y} = b_0 + b_1X_1 + b_2X_2$

Model 3: $\hat{Y} = b_0 + b_1X_1 + b_2X_2 + b_3X_3$

Full model: $\hat{Y} = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4$

"nested in"



The diagram illustrates the nesting of models. Blue curved arrows point from the right side of one model's equation to the left side of the next model's equation, indicating that each subsequent model is nested within the previous one. The arrows connect the Intercept model to Model 1, Model 1 to Model 2, Model 2 to Model 3, and Model 3 to the Full model.

R^2 in Nested Models

R^2 = variance explained in the model

Intercept model:	$\hat{Y} = b_0$	$R^2_{intercept}$
Model 1:	$\hat{Y} = b_0 + b_1X_1$	$R^2_{Model_1}$
Model 2:	$\hat{Y} = b_0 + b_1X_1 + b_2X_2$	$R^2_{Model_2}$
Model 3:	$\hat{Y} = b_0 + b_1X_1 + b_2X_2 + b_3X_3$	$R^2_{Model_3}$
Full model:	$\hat{Y} = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4$	$R^2_{full_model}$

$$R^2_{intercept} < R^2_{Model_1} < R^2_{Model_2} < R^2_{Model_3} < R^2_{full_model}$$

$$R^2_{full_model} < R^2_{iv1_dv} + R^2_{iv2_dv} + R^2_{iv3_dv} + R^2_{iv4_dv}$$

R^2 when only a single IV is
correlated with the DV

Nested Models

Intercept model: $\hat{Y} = b_0$

$\left\{ \begin{array}{c} \text{mean of DV} \end{array} \right\}$

```
wine.intercept.lm = lm( dv ~ 1, data = wineReduced )
```

Full model: $\hat{Y} = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4$

$\left\{ \begin{array}{c} \text{volatile.acidity} \\ \text{total.sulfur.dioxide} \\ \text{sulphates} \\ \text{alcohol} \end{array} \right\}$

```
wine.full.lm = lm( dv ~ volatile.acidity, ...  
                  data = wineReduced )
```

Nested Models

Intercept model: $\hat{Y} = b_0$

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.63744	0.02069	272.5	<2e-16 ***

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

Residual standard error: 0.8059 on 1516 degrees of freedom

Full model: $\hat{Y} = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4$

Wine Quality Equation

$$\hat{Y} = 2.62 - 1.11 \times \text{volatile.acidity} - 0.00242 \times \text{total.sulfur.dioxide} + 1.15 \times \text{sulphates} + 0.285 \times \text{alcohol}$$

Intercept

Coefficients

Coefficients

coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	2.6182760	0.2058383	12.720	< 2e-16	***
volatile.acidity	-1.1139841	0.0985556	-11.303	< 2e-16	***
total.sulfur.dioxide	-0.0024198	0.0005487	-4.410	1.11e-05	***
sulphates	1.1494063	0.1306377	8.798	< 2e-16	***
alcohol	0.2851406	0.0164983	17.283	< 2e-16	***

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

Residual standard error: 0.643 on 1512 degrees of freedom

Multiple R-squared: 0.3652, Adjusted R-squared: 0.3635

F-statistic: 217.4 on 4 and 1512 DF, p-value: < 2.2e-16

$$df = n - k^*$$

Where k^* is the numbers of parameters you're estimating **including** an intercept.

Model Comparison: Full vs Intercept Model

$$F = \frac{(SSE_{reduced} - SSE_{full}) / (p - k)}{SSE_{full} / (n - p - 1)} \sim F_{p-k, n-p-1}$$

n = number of entries / participants

p = number of predictors in full model

k = number of predictors in the reduced model

```
anova( reduced_model, full_model )
```

Analysis of Variance Table

Model 1: quality ~ 1

Model 2: quality ~ volatile.acidity + total.sulfur.dioxide + sulphates + alcohol

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	1552	1017.79				
2	1548	650.63	4	367.15	218.38	< 2.2e-16 ***

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

Post-fit checking (Residual Plot)

- X-axis is the predicted values using the model

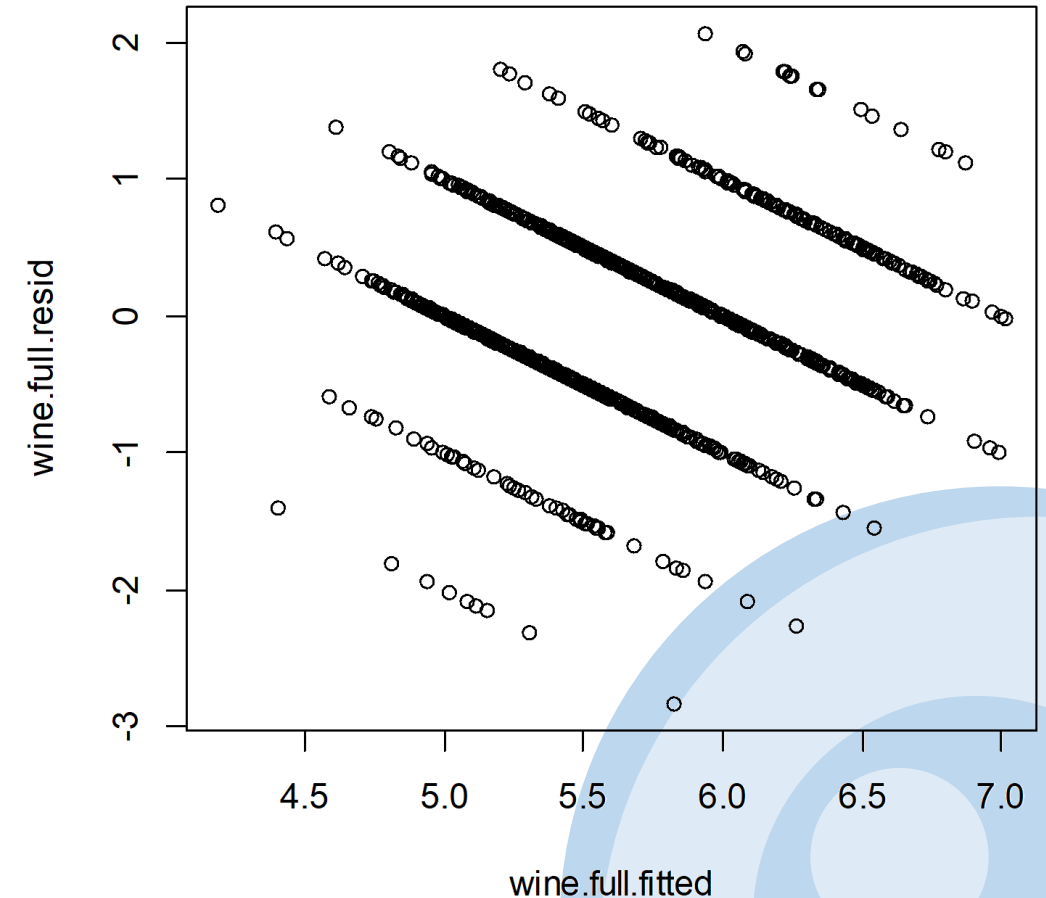
```
wine.full.fitted = predict(  
  lm_model,  
  predictors )
```

- Y is the residuals of the model (errors in prediction)

```
wine.full.resid = resid( lm_model )
```

- The graph should look like a random cloud

```
plot( fitted_values, residuals )
```



Multicollinearity: Variance Inflation Factors

- Correlations between predictor variables
- $VIF(\hat{\beta}_i) > 10 \rightarrow$ multicollinearity is high: the predictor variable can be explained by others in the model

$$VIF_i = \frac{1}{1 - R_i^2}$$

```
library(car)
vif( lm_model )
```

```
volatile.acidity total.sulfur.dioxide
1.150168          1.054194
```

```
sulphates
1.151780
```

```
alcohol
1.113112
```