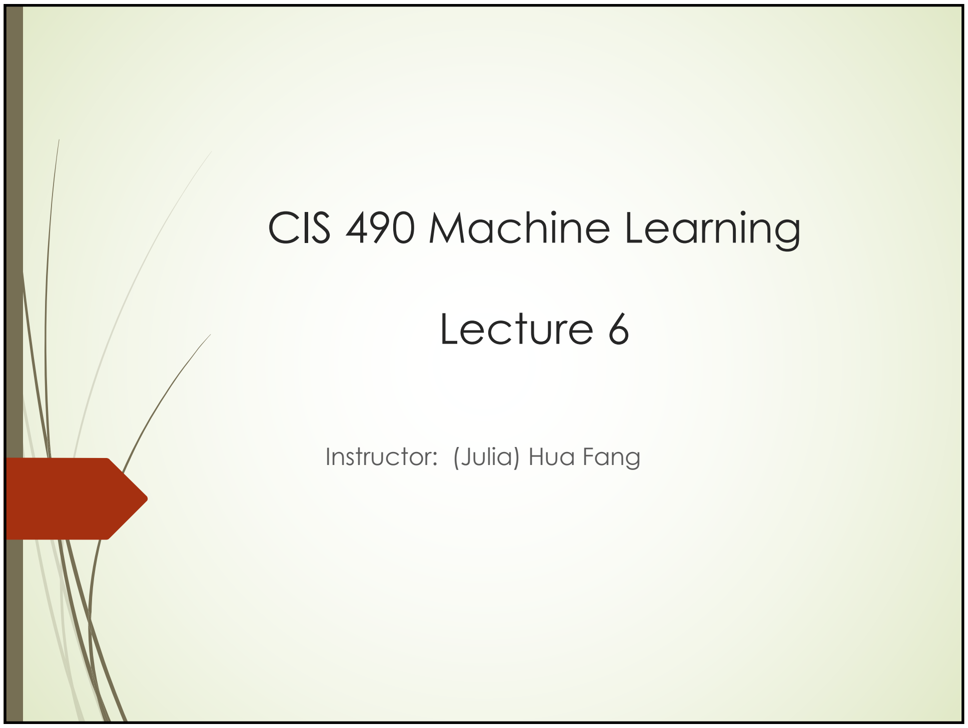
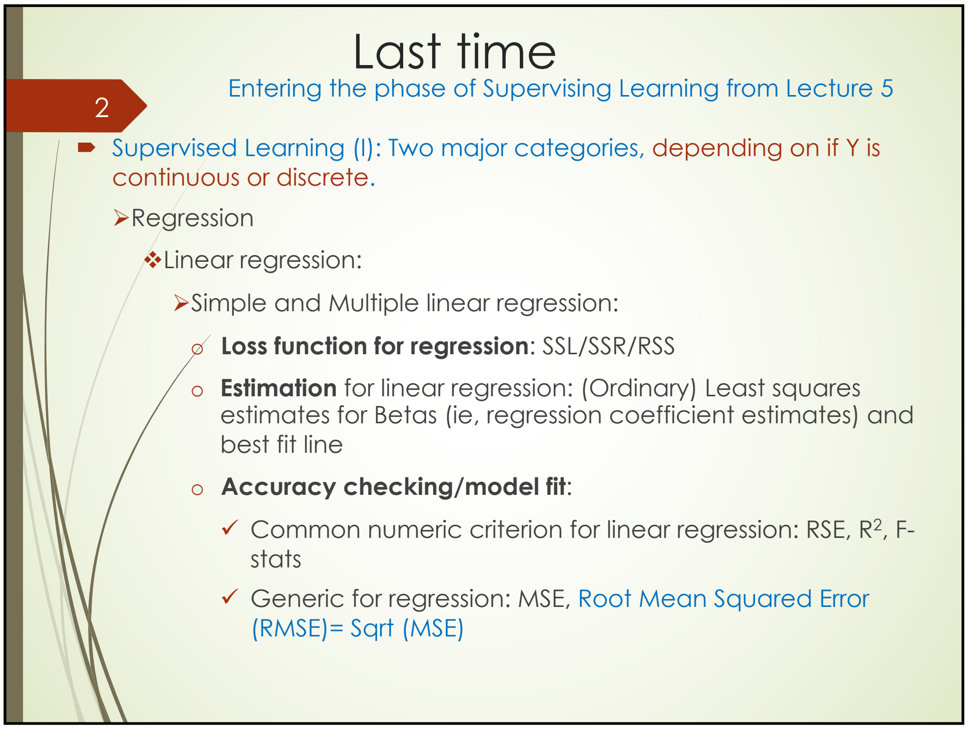


CIS 490 Machine Learning

Lecture 6

Instructor: (Julia) Hua Fang

1

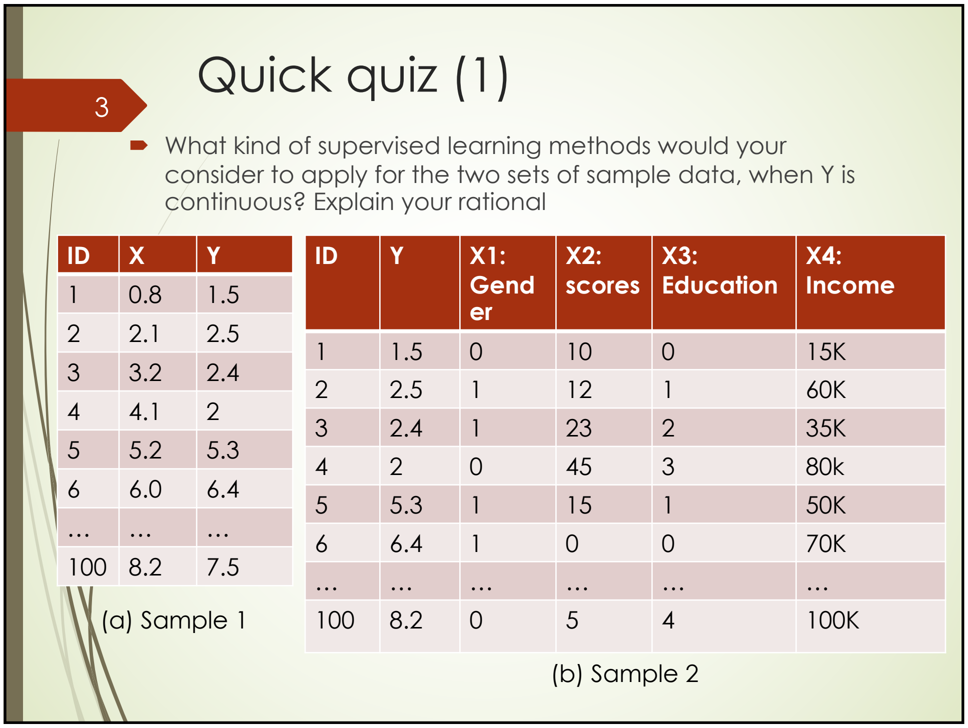


Last time

Entering the phase of Supervising Learning from Lecture 5

- Supervised Learning (I): Two major categories, depending on if Y is continuous or discrete.
  - ➢Regression
    - ❖Linear regression:
      - ➢Simple and Multiple linear regression:
        - o **Loss function for regression**: SSL/SSR/RSS
        - o **Estimation** for linear regression: (Ordinary) Least squares estimates for Betas (ie, regression coefficient estimates) and best fit line
        - o **Accuracy checking/model fit**:
          - ✓ Common numeric criterion for linear regression: RSE, R^2, F-stats
          - ✓ Generic for regression: MSE, Root Mean Squared Error (RMSE)= Sqrt (MSE)

2



3

Quick quiz (1)

- What kind of supervised learning methods would your consider to apply for the two sets of sample data, when Y is continuous? Explain your rational

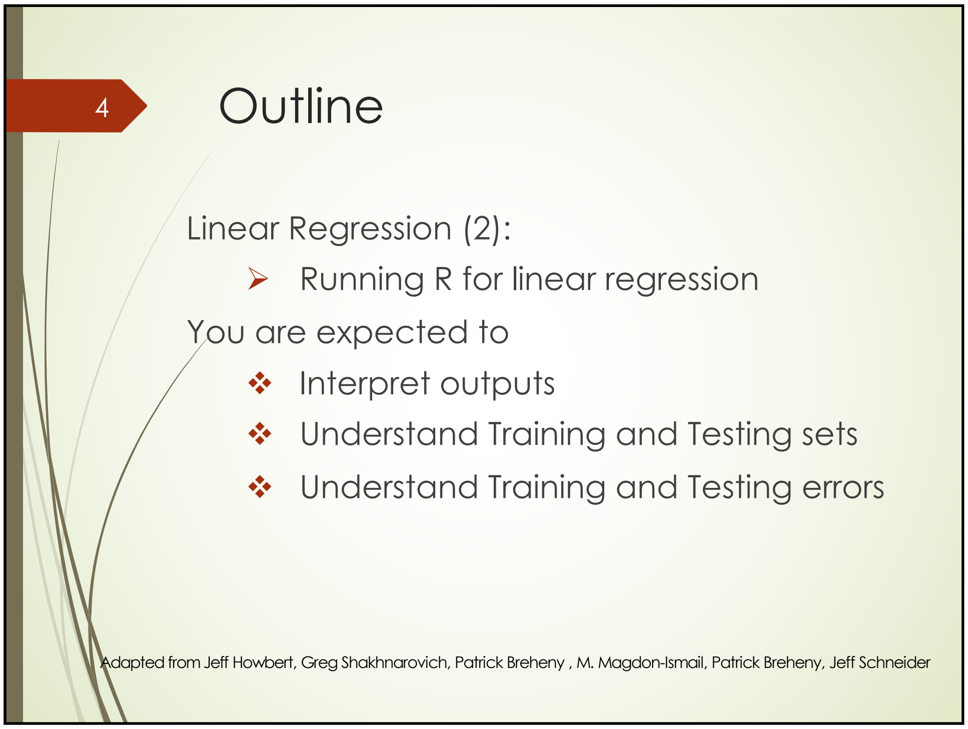
ID	X	Y
1	0.8	1.5
2	2.1	2.5
3	3.2	2.4
4	4.1	2
5	5.2	5.3
6	6.0	6.4
…	…	…
100	8.2	7.5

(a) Sample 1

ID	Y	X1: Gender	X2: scores	X3: Education	X4: Income
1	1.5	0	10	0	15K
2	2.5	1	12	1	60K
3	2.4	1	23	2	35K
4	2	0	45	3	80k
5	5.3	1	15	1	50K
6	6.4	1	0	0	70K
…	…	…	…	…	…
100	8.2	0	5	4	100K

(b) Sample 2

3



4

Outline

Linear Regression (2):

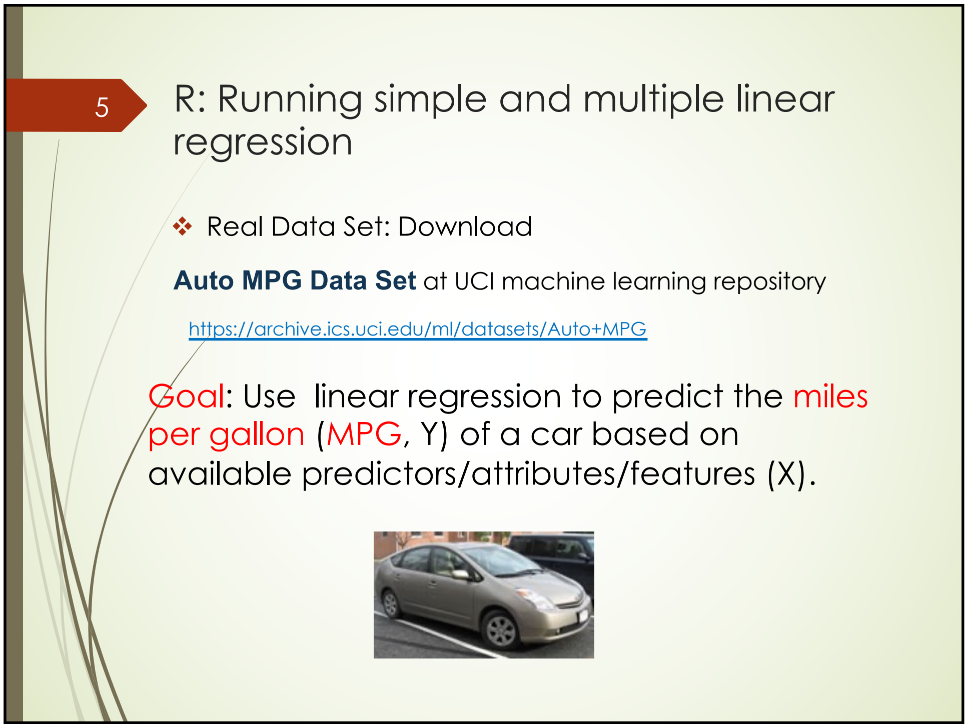
- ➢ Running R for linear regression

You are expected to

- ❖ Interpret outputs
- ❖ Understand Training and Testing sets
- ❖ Understand Training and Testing errors

Adapted from Jeff Howbert, Greg Shakhnarovich, Patrick Breheny , M. Magdon-Ismail, Patrick Breheny, Jeff Schneider

4



5

R: Running simple and multiple linear regression

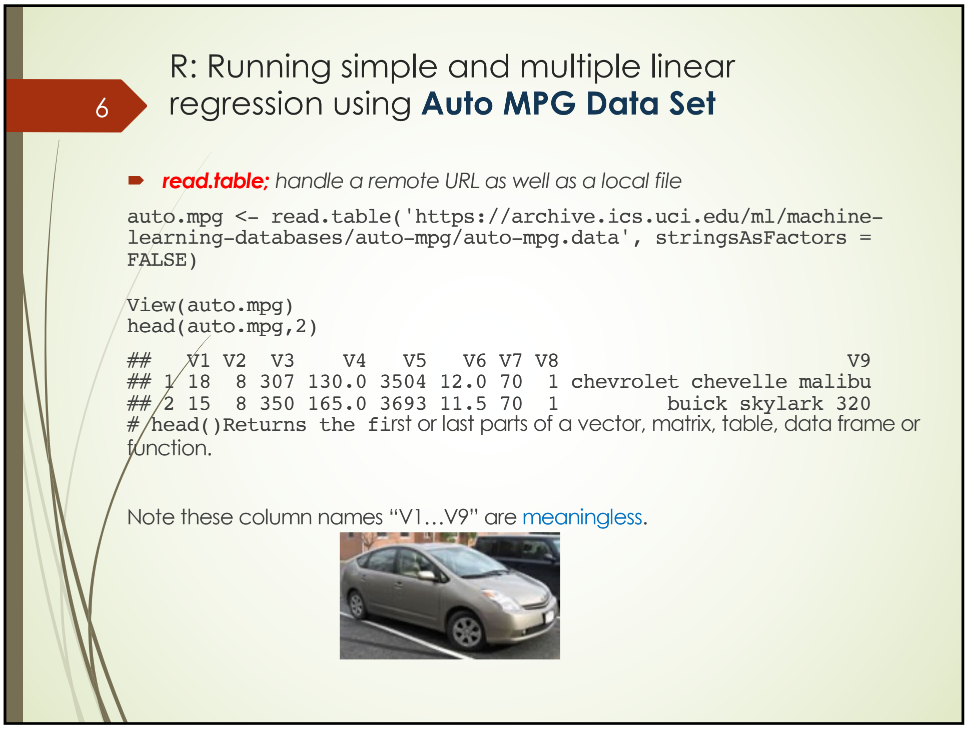
- Real Data Set: Download

Auto MPG Data Set at UCI machine learning repository

https://archive.ics.uci.edu/ml/datasets/Auto+MPG

Goal: Use linear regression to predict the miles per gallon (MPG, Y) of a car based on available predictors/attributes/features (X).

5



6

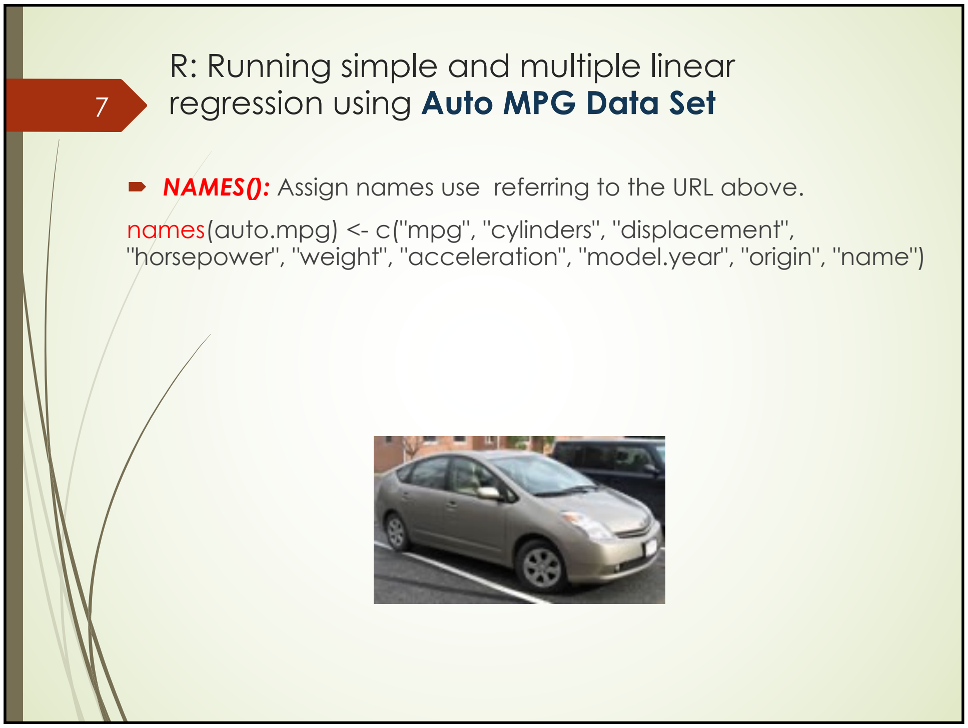
R: Running simple and multiple linear regression using **Auto MPG Data Set**

- ***read.table;*** *handle a remote URL as well as a local file*

```
auto.mpg <- read.table('https://archive.ics.uci.edu/ml/machine-
learning-databases/auto-mpg/auto-mpg.data', stringsAsFactors =
FALSE)

View(auto.mpg)
head(auto.mpg,2)

##   V1 V2  V3    V4   V5   V6 V7 V8                        V9
## 1 18  8 307 130.0 3504 12.0 70  1 chevrolet chevelle malibu
## 2 15  8 350 165.0 3693 11.5 70  1         buick skylark 320
# head()Returns the first or last parts of a vector, matrix, table, data frame or
function.
```

Note these column names “V1…V9” are meaningless.

6



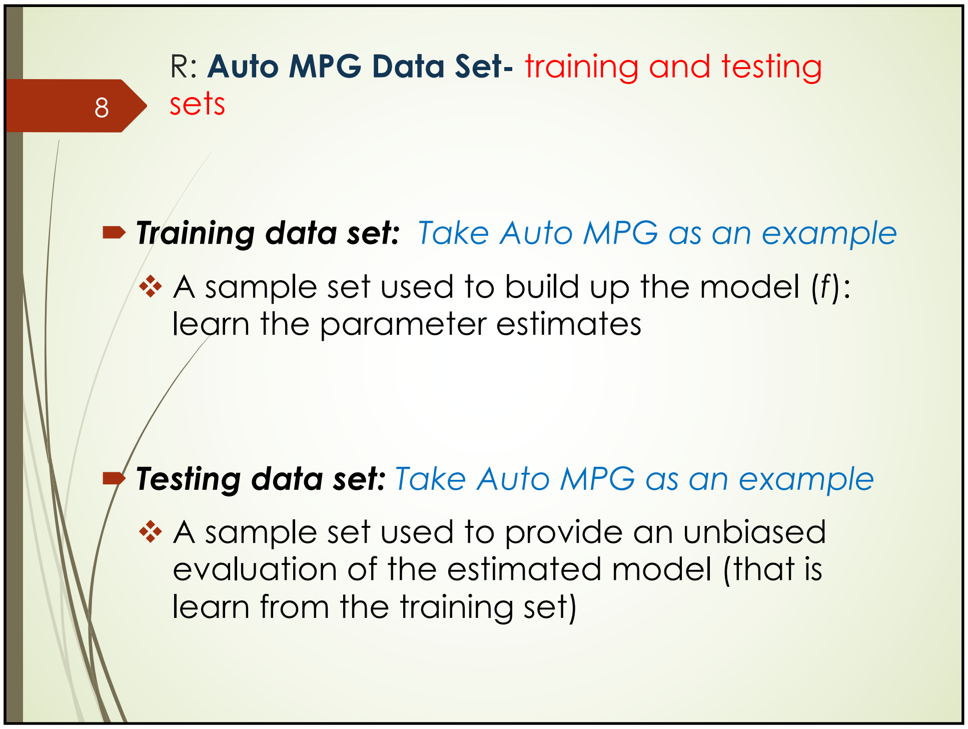
7

R: Running simple and multiple linear regression using **Auto MPG Data Set**

- ***NAMES():*** Assign names use referring to the URL above.

```
names(auto.mpg) <- c("mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration", "model.year", "origin", "name")
```

7

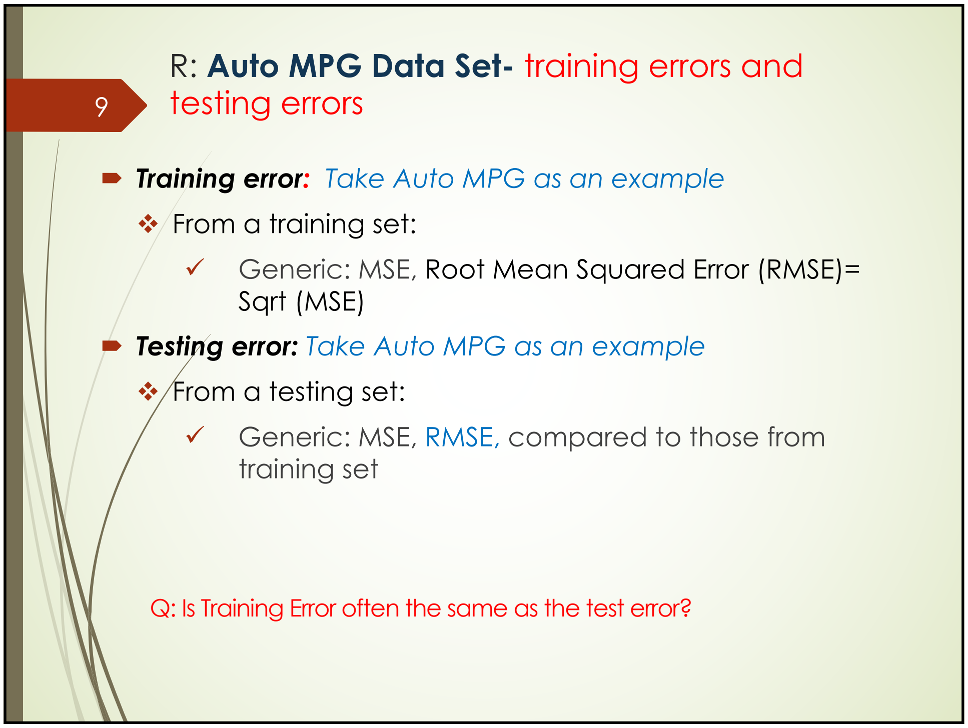


8

R: **Auto MPG Data Set-** training and testing sets

- ***Training data set:*** *Take Auto MPG as an example*
  - A sample set used to build up the model (*f*): learn the parameter estimates
- ***Testing data set:*** *Take Auto MPG as an example*
  - A sample set used to provide an unbiased evaluation of the estimated model (that is learn from the training set)

8



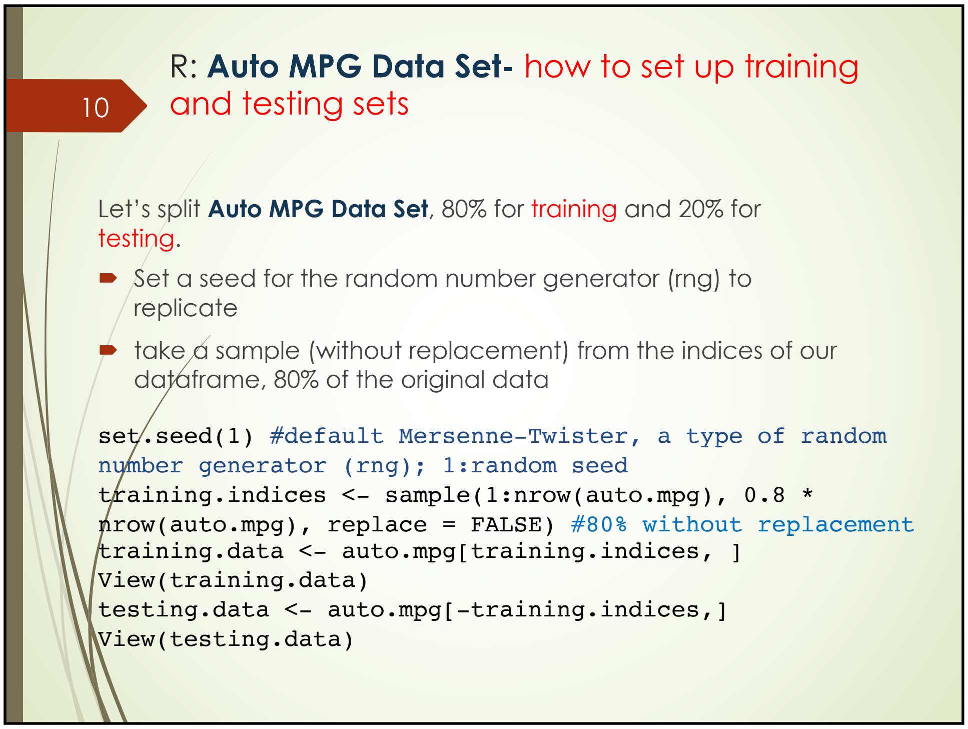
9

R: **Auto MPG Data Set-** training errors and testing errors

- ***Training error:*** *Take Auto MPG as an example*
  - From a training set:
    - ✓ Generic: MSE, Root Mean Squared Error (RMSE)= Sqrt (MSE)
- ***Testing error:*** *Take Auto MPG as an example*
  - From a testing set:
    - ✓ Generic: MSE, RMSE, compared to those from training set

Q: Is Training Error often the same as the test error?

9



10

R: **Auto MPG Data Set-** how to set up training and testing sets

Let's split **Auto MPG Data Set**, 80% for training and 20% for testing.

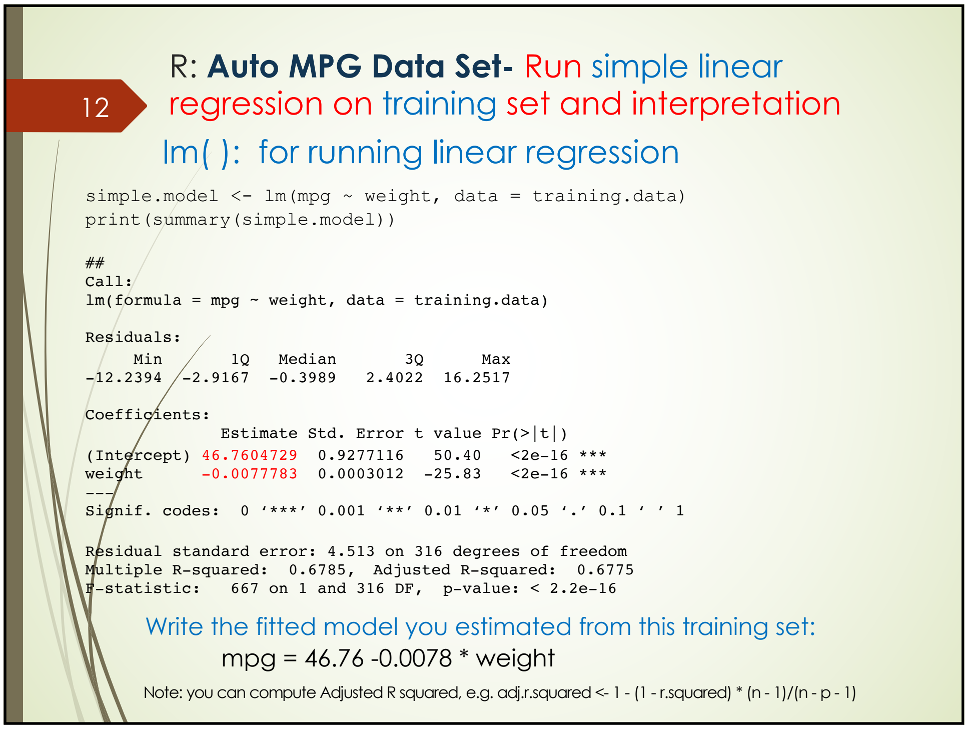
- Set a seed for the random number generator (rng) to replicate
- take a sample (without replacement) from the indices of our dataframe, 80% of the original data

```
set.seed(1) #default Mersenne-Twister, a type of random
number generator (rng); 1:random seed
training.indices <- sample(1:nrow(auto.mpg), 0.8 *
nrow(auto.mpg), replace = FALSE) #80% without replacement
training.data <- auto.mpg[training.indices, ]
View(training.data)
testing.data <- auto.mpg[-training.indices,]
View(testing.data)
```

10

R: **Auto MPG Data Set-** Run Simple linear regression

11



R: **Auto MPG Data Set-** Run simple linear regression on training set and interpretation

lm( ): for running linear regression

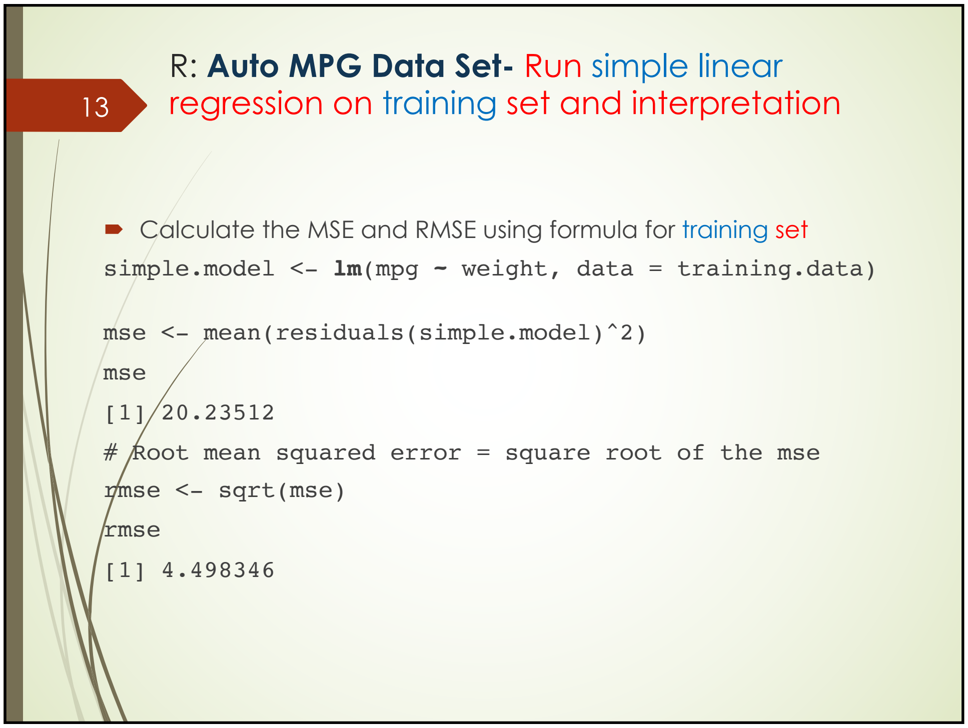
```
simple.model <- lm(mpg ~ weight, data = training.data)
print(summary(simple.model))

##
Call:
lm(formula = mpg ~ weight, data = training.data)

Residuals:
     Min       1Q   Median       3Q      Max
-12.2394  -2.9167  -0.3989   2.4022  16.2517

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 46.7604729  0.9277116   50.40   <2e-16 ***
weight      -0.0077783  0.0003012  -25.83   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.513 on 316 degrees of freedom
Multiple R-squared:  0.6785,  Adjusted R-squared:  0.6775
F-statistic:   667 on 1 and 316 DF,  p-value: < 2.2e-16
```

Write the fitted model you estimated from this training set:

mpg = 46.76 -0.0078 * weight

Note: you can compute Adjusted R squared, e.g. adj.r.squared <- 1 - (1 - r.squared) * (n - 1)/(n - p - 1)

12



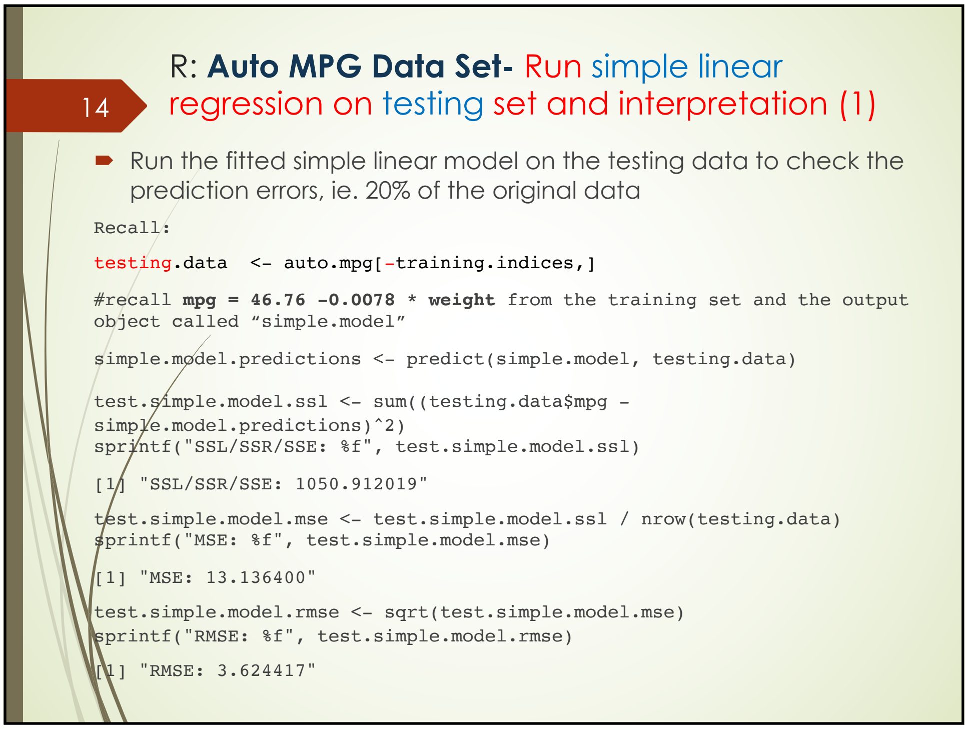
13

R: **Auto MPG Data Set-** Run simple linear regression on training set and interpretation

- Calculate the MSE and RMSE using formula for training set

```
simple.model <- lm(mpg ~ weight, data = training.data)

mse <- mean(residuals(simple.model)^2)
mse
[1] 20.23512
# Root mean squared error = square root of the mse
rmse <- sqrt(mse)
rmse
[1] 4.498346
```

13



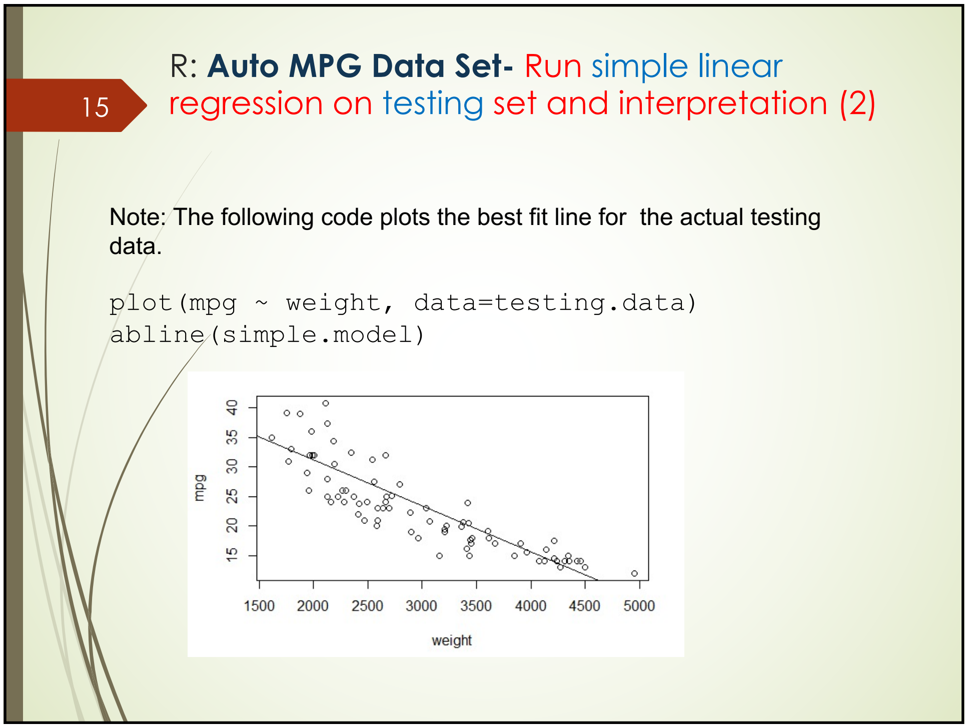
14

R: **Auto MPG Data Set-** Run simple linear regression on testing set and interpretation (1)

- Run the fitted simple linear model on the testing data to check the prediction errors, ie. 20% of the original data

```
Recall:
testing.data  <- auto.mpg[-training.indices,]
#recall mpg = 46.76 -0.0078 * weight from the training set and the output
object called “simple.model”
simple.model.predictions <- predict(simple.model, testing.data)
test.simple.model.ssl <- sum((testing.data$mpg -
simple.model.predictions)^2)
sprintf("SSL/SSR/SSE: %f", test.simple.model.ssl)
[1] "SSL/SSR/SSE: 1050.912019"
test.simple.model.mse <- test.simple.model.ssl / nrow(testing.data)
sprintf("MSE: %f", test.simple.model.mse)
[1] "MSE: 13.136400"
test.simple.model.rmse <- sqrt(test.simple.model.mse)
sprintf("RMSE: %f", test.simple.model.rmse)
[1] "RMSE: 3.624417"
```

14



15

R: Auto MPG Data Set- Run simple linear regression on testing set and interpretation (2)

Note: The following code plots the best fit line for the actual testing data.

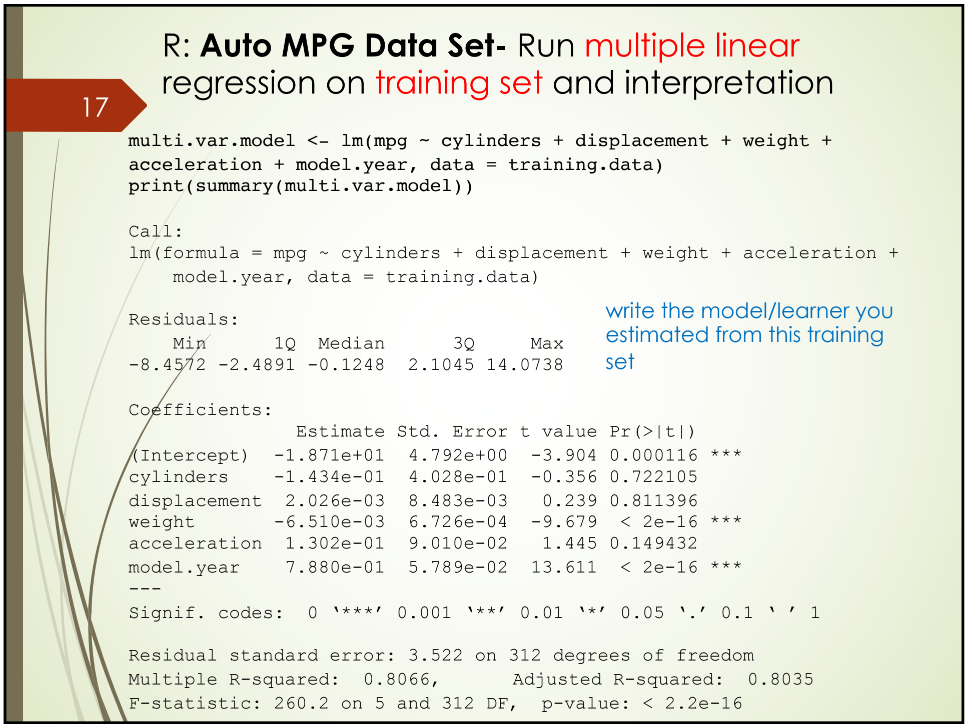
```
plot(mpg ~ weight, data=testing.data)
abline(simple.model)
```

15

16

R: Auto MPG Data Set- Run multiple linear regression

16



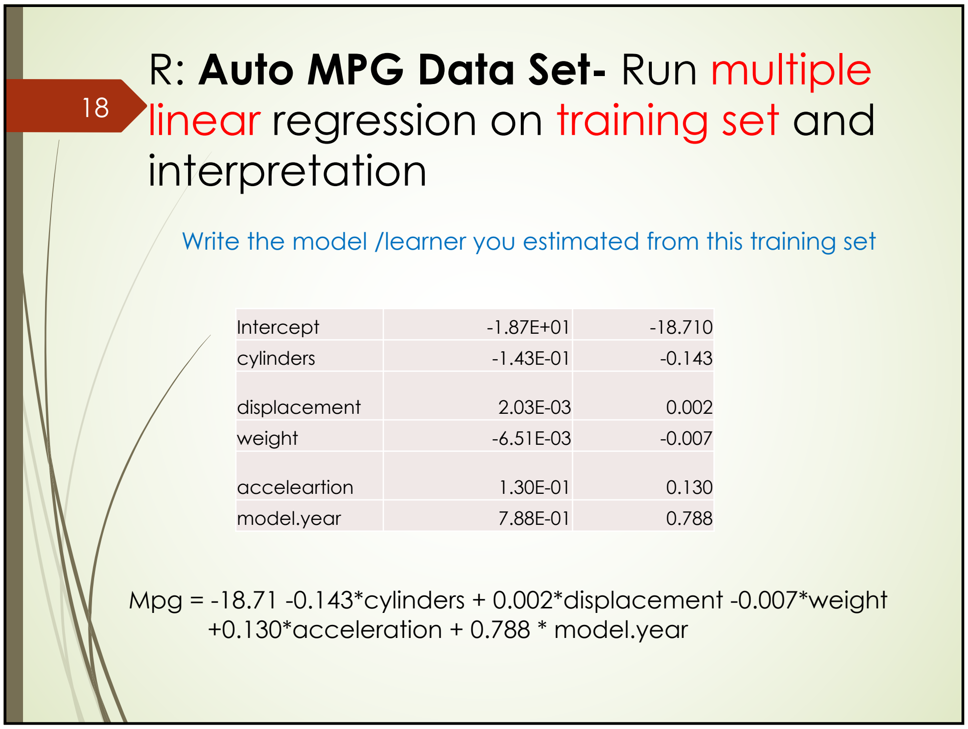
17

R: **Auto MPG Data Set-** Run multiple linear regression on training set and interpretation

```
multi.var.model <- lm(mpg ~ cylinders + displacement + weight +
acceleration + model.year, data = training.data)
print(summary(multi.var.model))

Call:
lm(formula = mpg ~ cylinders + displacement + weight + acceleration +
    model.year, data = training.data)

Residuals:
    Min      1Q  Median      3Q     Max
-8.4572 -2.4891 -0.1248  2.1045 14.0738

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.871e+01  4.792e+00  -3.904 0.000116 ***
cylinders    -1.434e-01  4.028e-01  -0.356 0.722105
displacement  2.026e-03  8.483e-03   0.239 0.811396
weight       -6.510e-03  6.726e-04  -9.679  < 2e-16 ***
acceleration  1.302e-01  9.010e-02   1.445 0.149432
model.year    7.880e-01  5.789e-02  13.611  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.522 on 312 degrees of freedom
Multiple R-squared:  0.8066,      Adjusted R-squared:  0.8035
F-statistic: 260.2 on 5 and 312 DF,  p-value: < 2.2e-16
```

write the model/learner you estimated from this training set

17



18

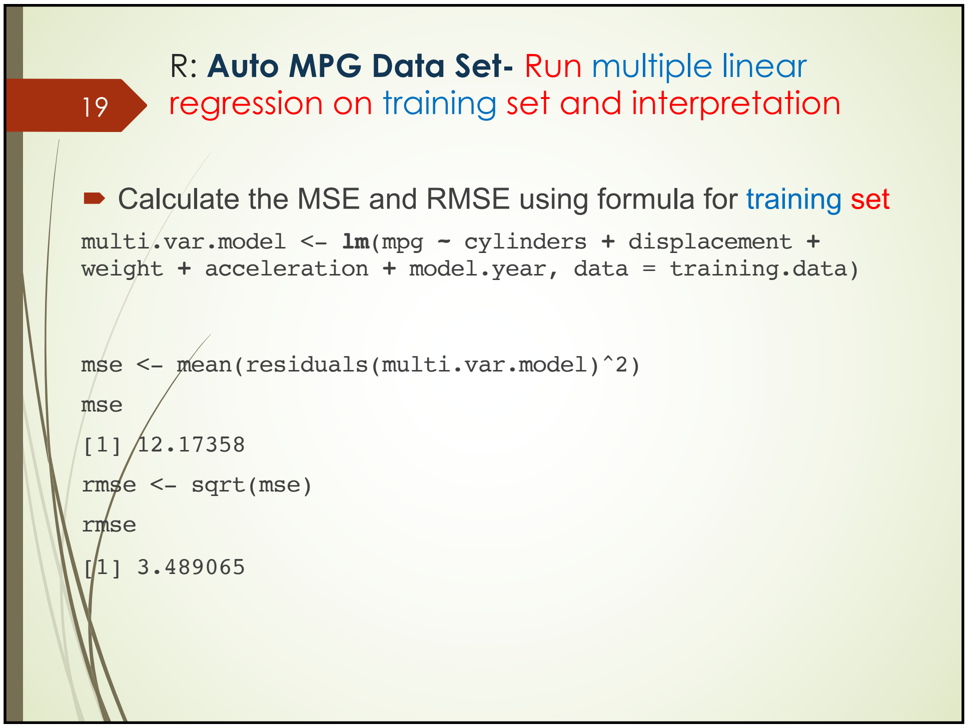
R: **Auto MPG Data Set-** Run multiple linear regression on training set and interpretation

Write the model /learner you estimated from this training set

Intercept	-1.87E+01	-18.710
cylinders	-1.43E-01	-0.143
displacement	2.03E-03	0.002
weight	-6.51E-03	-0.007
acceleartion	1.30E-01	0.130
model.year	7.88E-01	0.788

Mpg = -18.71 -0.143*cylinders + 0.002*displacement -0.007*weight
+0.130*acceleration + 0.788 * model.year

18



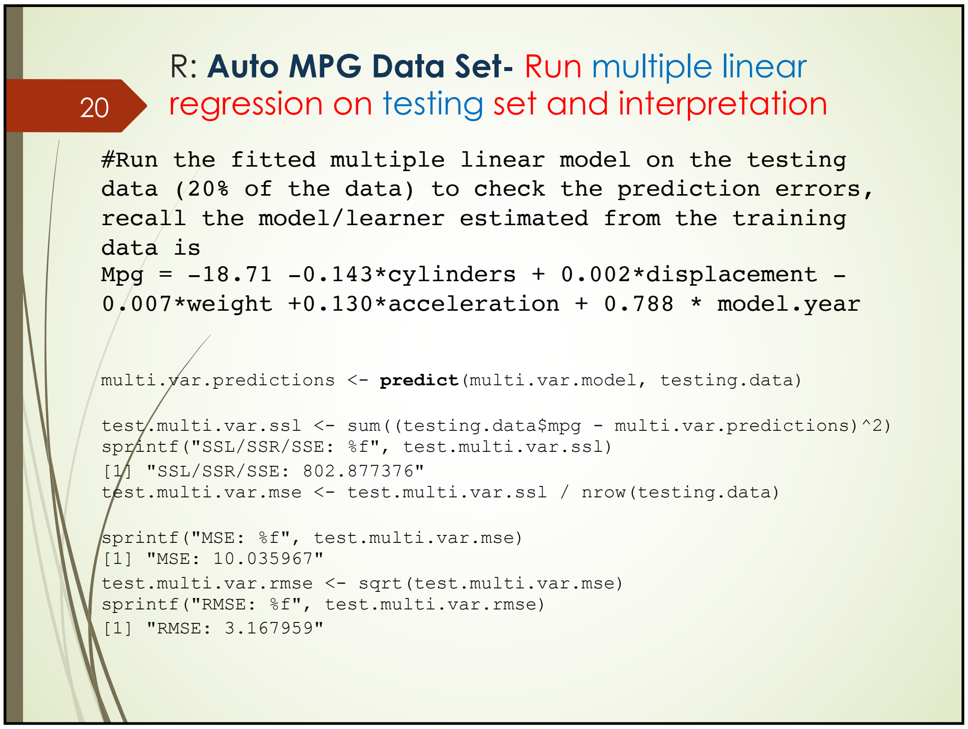
19

R: **Auto MPG Data Set-** Run multiple linear regression on training set and interpretation

- Calculate the MSE and RMSE using formula for training set

```
multi.var.model <- lm(mpg ~ cylinders + displacement +
weight + acceleration + model.year, data = training.data)


mse <- mean(residuals(multi.var.model)^2)
mse
[1] 12.17358
rmse <- sqrt(mse)
rmse
[1] 3.489065
```

19

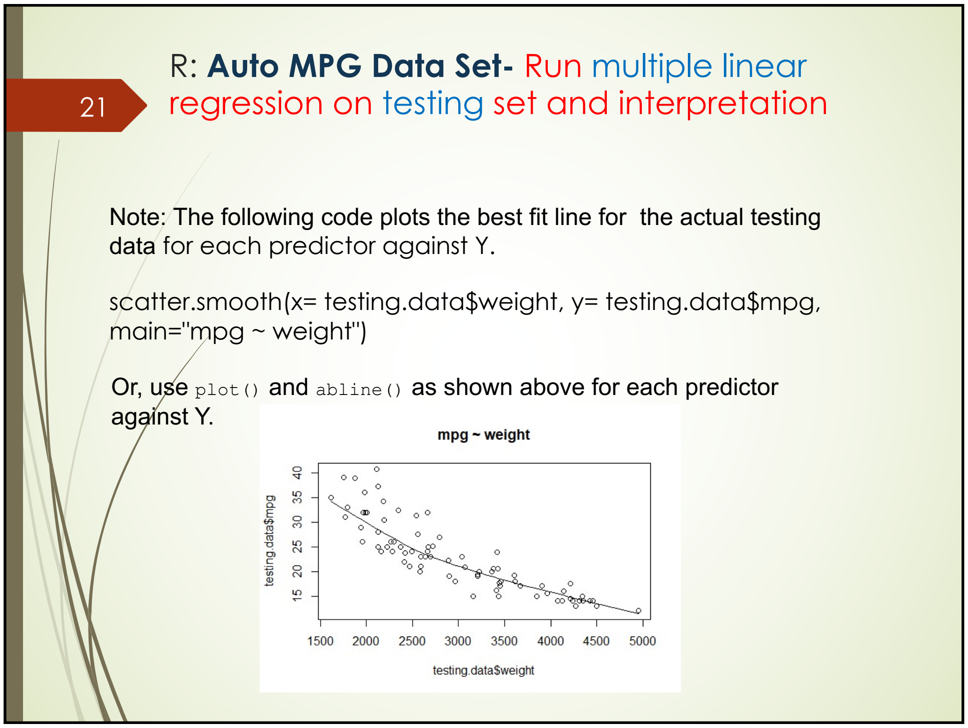


20

R: **Auto MPG Data Set-** Run multiple linear regression on testing set and interpretation

```
#Run the fitted multiple linear model on the testing
data (20% of the data) to check the prediction errors,
recall the model/learner estimated from the training
data is
Mpg = -18.71 -0.143*cylinders + 0.002*displacement -
0.007*weight +0.130*acceleration + 0.788 * model.year


multi.var.predictions <- predict(multi.var.model, testing.data)

test.multi.var.ssl <- sum((testing.data$mpg - multi.var.predictions)^2)
sprintf("SSL/SSR/SSE: %f", test.multi.var.ssl)
[1] "SSL/SSR/SSE: 802.877376"
test.multi.var.mse <- test.multi.var.ssl / nrow(testing.data)

sprintf("MSE: %f", test.multi.var.mse)
[1] "MSE: 10.035967"
test.multi.var.rmse <- sqrt(test.multi.var.mse)
sprintf("RMSE: %f", test.multi.var.rmse)
[1] "RMSE: 3.167959"
```

20



21

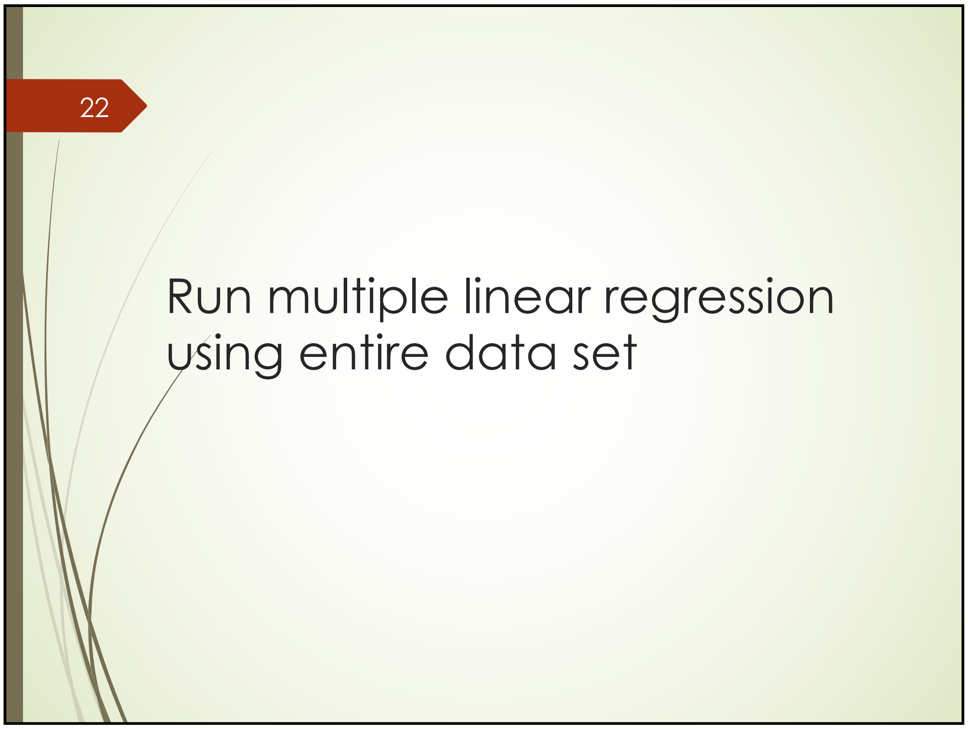
R: **Auto MPG Data Set-** Run multiple linear regression on testing set and interpretation

Note: The following code plots the best fit line for the actual testing data for each predictor against Y.

```
scatter.smooth(x= testing.data$weight, y= testing.data$mpg, main="mpg ~ weight")
```

Or, use `plot()` and `abline()` as shown above for each predictor against Y.

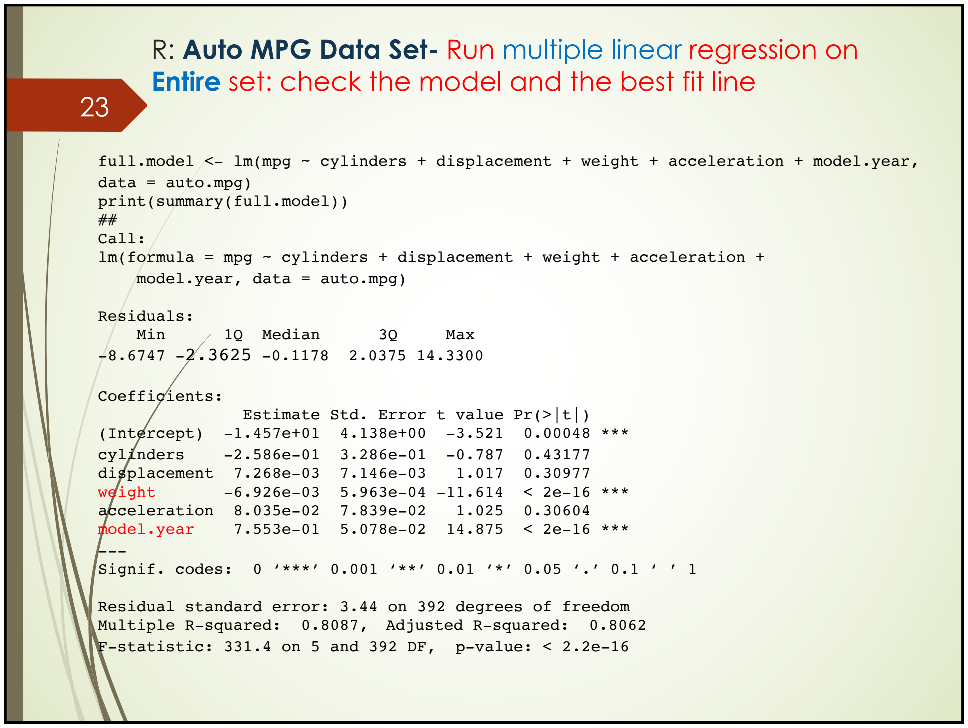
21



22

Run multiple linear regression using entire data set

22

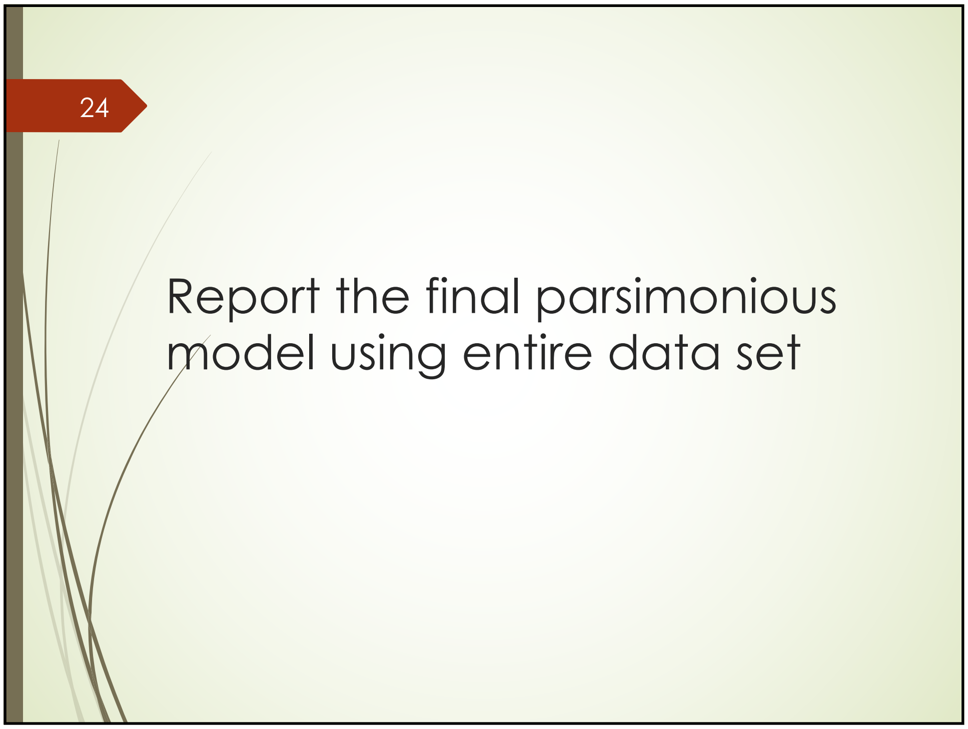


23

R: **Auto MPG Data Set-** Run multiple linear regression on **Entire** set: check the model and the best fit line

```
full.model <- lm(mpg ~ cylinders + displacement + weight + acceleration + model.year,
data = auto.mpg)
print(summary(full.model))
##
Call:
lm(formula = mpg ~ cylinders + displacement + weight + acceleration +
    model.year, data = auto.mpg)

Residuals:
    Min      1Q  Median      3Q     Max
-8.6747 -2.3625 -0.1178  2.0375 14.3300

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.457e+01  4.138e+00  -3.521  0.00048 ***
cylinders    -2.586e-01  3.286e-01  -0.787  0.43177
displacement  7.268e-03  7.146e-03   1.017  0.30977
weight       -6.926e-03  5.963e-04 -11.614  < 2e-16 ***
acceleration  8.035e-02  7.839e-02   1.025  0.30604
model.year    7.553e-01  5.078e-02  14.875  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.44 on 392 degrees of freedom
Multiple R-squared:  0.8087,  Adjusted R-squared:  0.8062
F-statistic: 331.4 on 5 and 392 DF,  p-value: < 2.2e-16
```

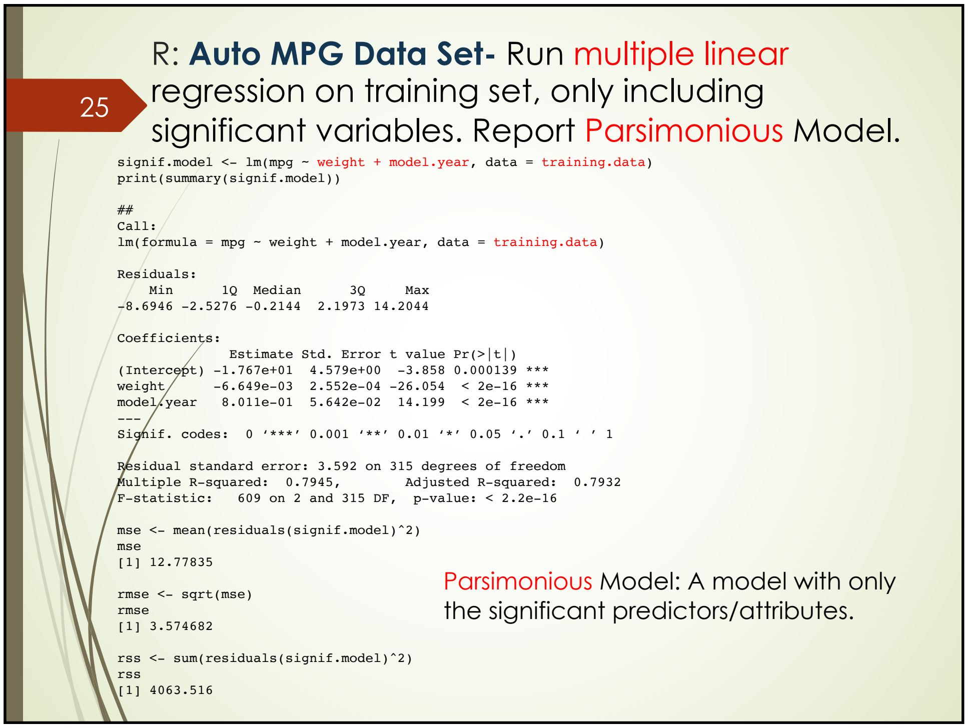
23



24

Report the final parsimonious model using entire data set

24



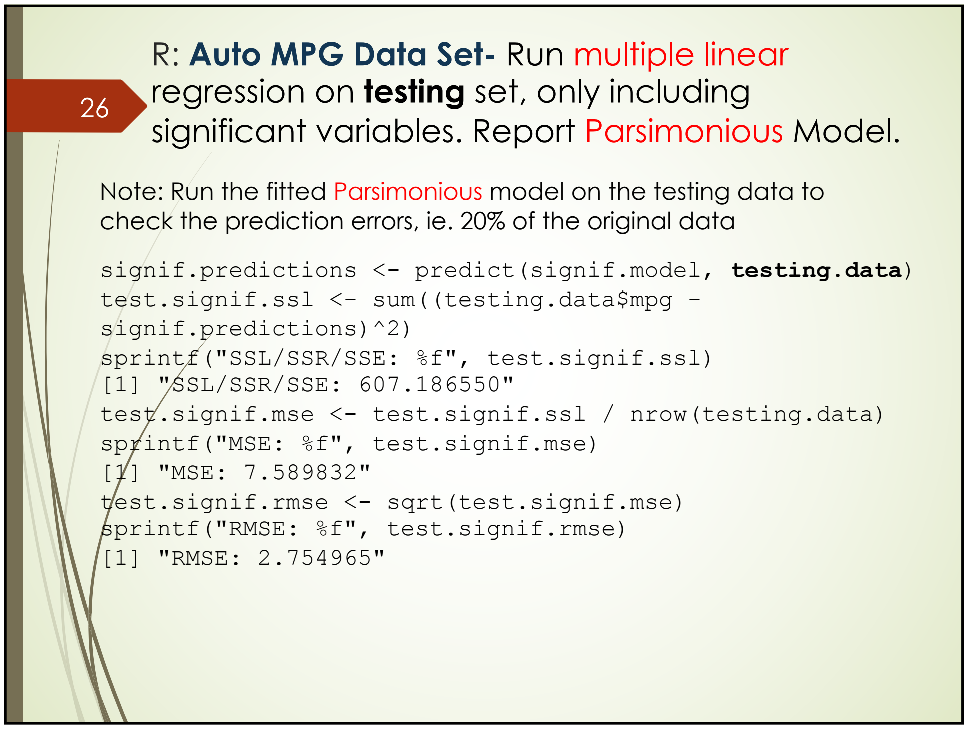
25

R: **Auto MPG Data Set-** Run multiple linear regression on training set, only including significant variables. Report Parsimonious Model.

```
signif.model <- lm(mpg ~ weight + model.year, data = training.data)
print(summary(signif.model))

##
Call:
lm(formula = mpg ~ weight + model.year, data = training.data)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.6946 -2.5276 -0.2144  2.1973 14.2044 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.767e+01  4.579e+00  -3.858 0.000139 ***
weight      -6.649e-03  2.552e-04 -26.054  < 2e-16 ***
model.year   8.011e-01  5.642e-02  14.199  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.592 on 315 degrees of freedom
Multiple R-squared:  0.7945,	Adjusted R-squared:  0.7932 
F-statistic:   609 on 2 and 315 DF,  p-value: < 2.2e-16

mse <- mean(residuals(signif.model)^2)
mse
[1] 12.77835

rmse <- sqrt(mse)
rmse
[1] 3.574682

rss <- sum(residuals(signif.model)^2)
rss
[1] 4063.516
```

Parsimonious Model: A model with only the significant predictors/attributes.

25



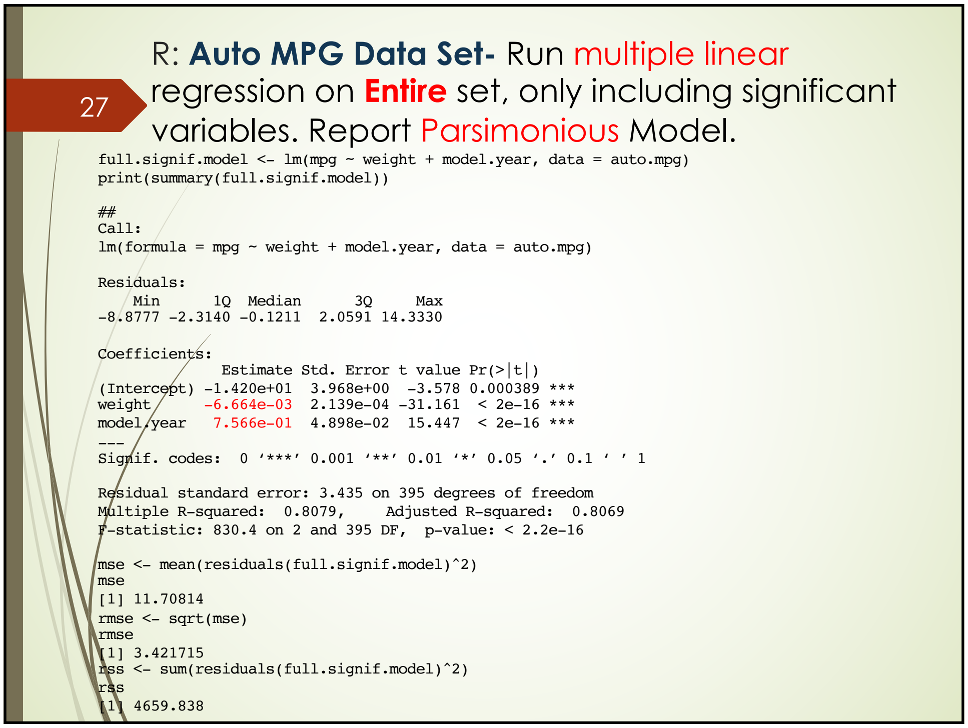
26

R: **Auto MPG Data Set-** Run multiple linear regression on **testing** set, only including significant variables. Report Parsimonious Model.

Note: Run the fitted Parsimonious model on the testing data to check the prediction errors, ie. 20% of the original data

```
signif.predictions <- predict(signif.model, testing.data)
test.signif.ssl <- sum((testing.data$mpg -
signif.predictions)^2)
sprintf("SSL/SSR/SSE: %f", test.signif.ssl)
[1] "SSL/SSR/SSE: 607.186550"
test.signif.mse <- test.signif.ssl / nrow(testing.data)
sprintf("MSE: %f", test.signif.mse)
[1] "MSE: 7.589832"
test.signif.rmse <- sqrt(test.signif.mse)
sprintf("RMSE: %f", test.signif.rmse)
[1] "RMSE: 2.754965"
```

26

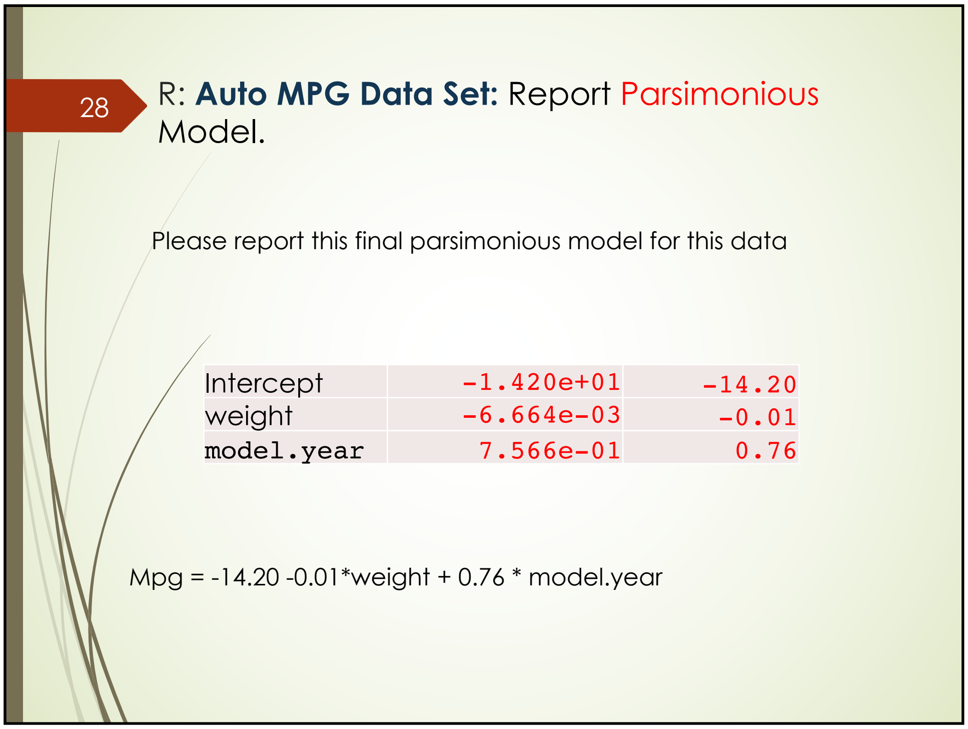


27

R: **Auto MPG Data Set-** Run multiple linear regression on **Entire** set, only including significant variables. Report Parsimonious Model.

```
full.signif.model <- lm(mpg ~ weight + model.year, data = auto.mpg)
print(summary(full.signif.model))

##
Call:
lm(formula = mpg ~ weight + model.year, data = auto.mpg)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.8777 -2.3140 -0.1211  2.0591 14.3330 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.420e+01  3.968e+00  -3.578 0.000389 ***
weight      -6.664e-03  2.139e-04 -31.161  < 2e-16 ***
model.year   7.566e-01  4.898e-02  15.447  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.435 on 395 degrees of freedom
Multiple R-squared:  0.8079,     Adjusted R-squared:  0.8069 
F-statistic: 830.4 on 2 and 395 DF,  p-value: < 2.2e-16

mse <- mean(residuals(full.signif.model)^2)
mse
[1] 11.70814
rmse <- sqrt(mse)
rmse
[1] 3.421715
rss <- sum(residuals(full.signif.model)^2)
rss
[1] 4659.838
```

27



28

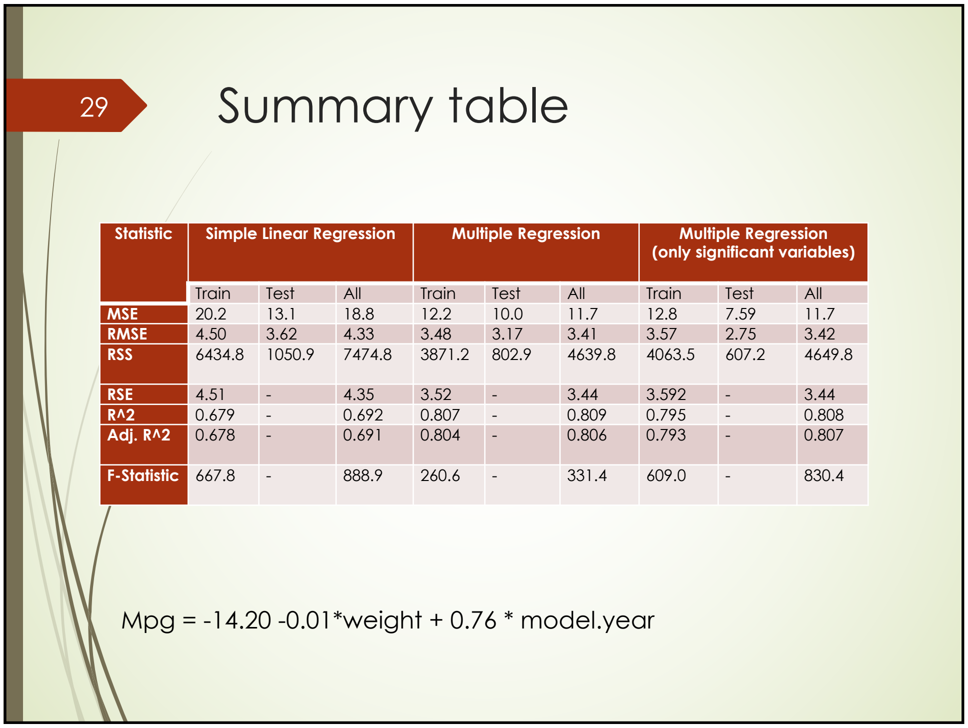
R: **Auto MPG Data Set:** Report Parsimonious Model.

Please report this final parsimonious model for this data

Intercept	-1.420e+01	-14.20
weight	-6.664e-03	-0.01
model.year	7.566e-01	0.76

Mpg = -14.20 -0.01*weight + 0.76 * model.year

28



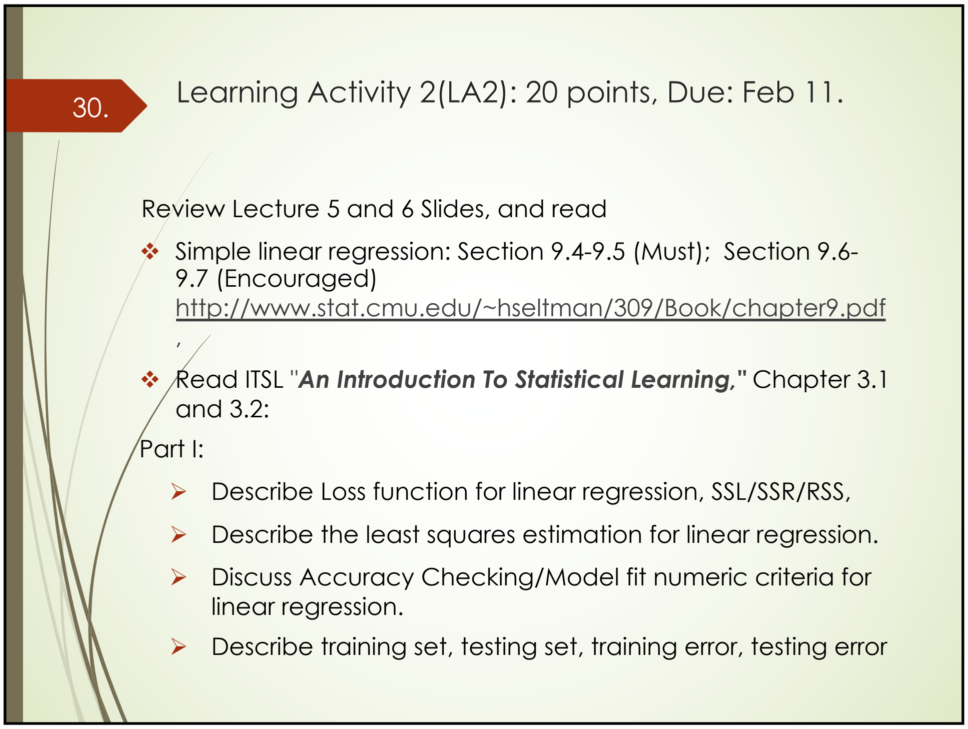
29

Summary table

Statistic	Simple Linear Regression			Multiple Regression			Multiple Regression (only significant variables)		
	Train	Test	All	Train	Test	All	Train	Test	All
MSE	20.2	13.1	18.8	12.2	10.0	11.7	12.8	7.59	11.7
RMSE	4.50	3.62	4.33	3.48	3.17	3.41	3.57	2.75	3.42
RSS	6434.8	1050.9	7474.8	3871.2	802.9	4639.8	4063.5	607.2	4649.8
RSE	4.51	-	4.35	3.52	-	3.44	3.592	-	3.44
R^2	0.679	-	0.692	0.807	-	0.809	0.795	-	0.808
Adj. R^2	0.678	-	0.691	0.804	-	0.806	0.793	-	0.807
F-Statistic	667.8	-	888.9	260.6	-	331.4	609.0	-	830.4

Mpg = -14.20 -0.01*weight + 0.76 * model.year

29



30.

Learning Activity 2(LA2): 20 points, Due: Feb 11.

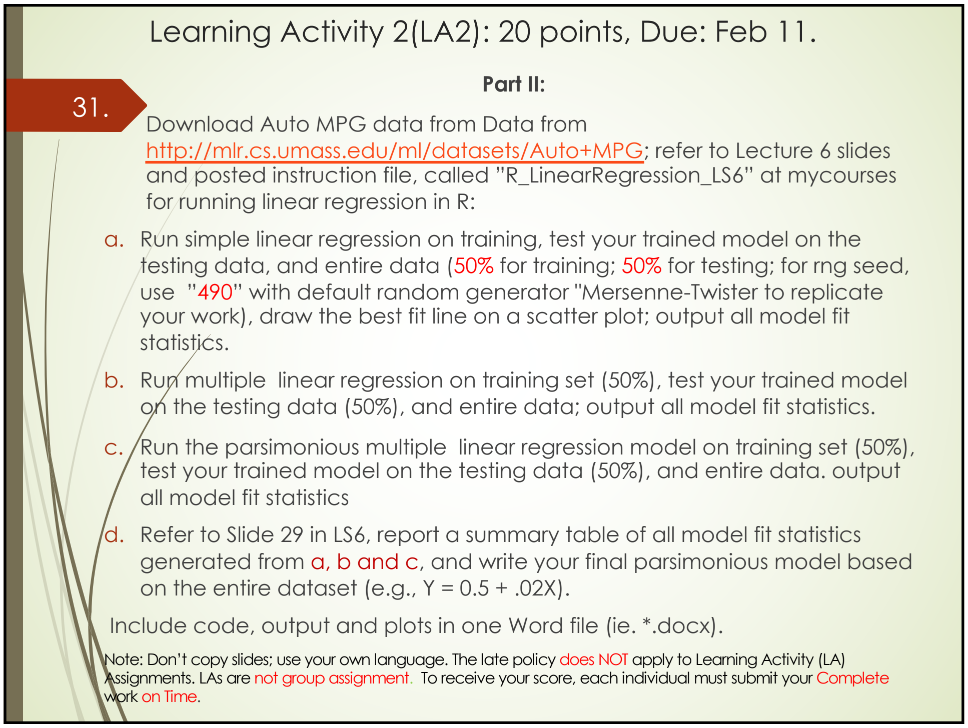
Review Lecture 5 and 6 Slides, and read

- Simple linear regression: Section 9.4-9.5 (Must); Section 9.6-9.7 (Encouraged) http://www.stat.cmu.edu/~hseltman/309/Book/chapter9.pdf ,
- Read ITSL "***An Introduction To Statistical Learning,***" Chapter 3.1 and 3.2:

Part I:

- Describe Loss function for linear regression, SSL/SSR/RSS,
- Describe the least squares estimation for linear regression.
- Discuss Accuracy Checking/Model fit numeric criteria for linear regression.
- Describe training set, testing set, training error, testing error

30



31.

Learning Activity 2(LA2): 20 points, Due: Feb 11.

Part II:

Download Auto MPG data from Data from http://mlr.cs.umass.edu/ml/datasets/Auto+MPG; refer to Lecture 6 slides and posted instruction file, called "R_LinearRegression_LS6" at mycourses for running linear regression in R:

a. Run simple linear regression on training, test your trained model on the testing data, and entire data (50% for training; 50% for testing; for rng seed, use "490" with default random generator "Mersenne-Twister to replicate your work), draw the best fit line on a scatter plot; output all model fit statistics.

b. Run multiple linear regression on training set (50%), test your trained model on the testing data (50%), and entire data; output all model fit statistics.

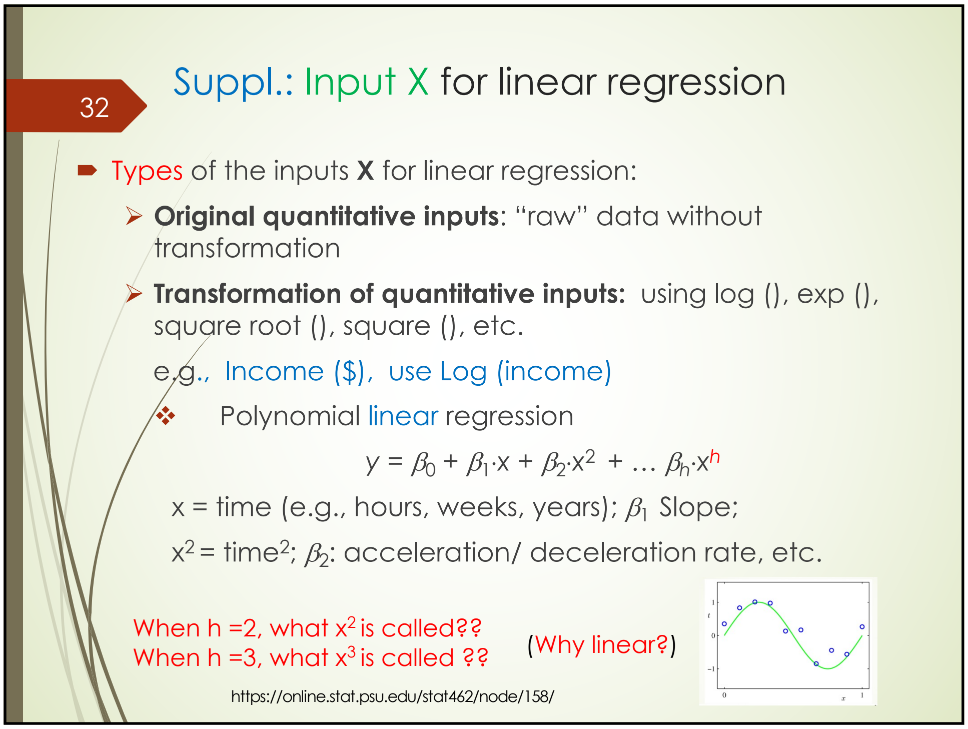
c. Run the parsimonious multiple linear regression model on training set (50%), test your trained model on the testing data (50%), and entire data. output all model fit statistics

d. Refer to Slide 29 in LS6, report a summary table of all model fit statistics generated from a, b and c, and write your final parsimonious model based on the entire dataset (e.g., Y = 0.5 + .02X).

Include code, output and plots in one Word file (ie. *.docx).

Note: Don't copy slides; use your own language. The late policy does NOT apply to Learning Activity (LA) Assignments. LAs are not group assignment. To receive your score, each individual must submit your Complete work on Time.

31



32

Suppl.: Input X for linear regression

- Types of the inputs **X** for linear regression:
  - **Original quantitative inputs**: "raw" data without transformation
  - **Transformation of quantitative inputs:** using log (), exp (), square root (), square (), etc.

    e.g., Income ($), use Log (income)

    ❖ Polynomial linear regression

$$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2 + \ldots \beta_h \cdot x^h$$

x = time (e.g., hours, weeks, years); β_1 Slope;

x^2 = time^2; β_2: acceleration/ deceleration rate, etc.

When h =2, what x^2 is called??
When h =3, what x^3 is called ??

(Why linear?)

https://online.stat.psu.edu/stat462/node/158/

32