# Multiple Linear Regression Basics

Reference reading:

- Model fitting (3.1, 3.2)
- Assessing the fit (3.1, 3.2)
- Statistical inference on model parameters and determining important predictors (3.1, 3.2)
- Using the model for prediction (3.2)

# Fitting the Model: Least Squares

- Objective: Given sample of $n$ multivariate observations ($n$ rows of a response variable and $k$ predictor variables), **estimate the parameters (aka coefficients)** of a linear regression model:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_k x_k + \varepsilon$$

$y$: response variable

$\{x_1, \ldots, x_k\}$: predictor variables

- **Least squares** is a very old and popular criterion, in which the parameters are estimated by minimizing the sum of squares of errors (SSE):

$$SSE = \sum_{i=1}^{n} \left[ y_i - \left( \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \boxed{?} + \hat{\beta}_k x_{ik} \right) \right]^2 = \sum_{i=1}^{n} e_i^2$$

- You can fit many nonlinear models with "linear" least squares – model must be **linear in the parameters**, not the predictors

# Discussion Points and Questions

- Will the least squares SSE criterion used to fit the model be sensitive to outliers in the response values?
- How could you modify the criterion to make the model fitting less sensitive (more robust) to outliers?
- What are the drawbacks of using "robust regression" criteria like trimmed estimators and least absolute deviations, as opposed to least squares?
- "Robust" regression criterion are more popular now than in the past because of computing power, but least squares is still by far the dominant criterion

# Least Squares Solution

To minimize $SSE = \sum_{i=1}^{n} \left[ y_i - \left( \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \boxed{?} + \hat{\beta}_k x_{ik} \right) \right]^2$

set

$$\left. \frac{\partial SSE}{\partial \beta_0} \right|_{\hat{\beta}'s} = -2 \sum_{i=1}^{n} \left[ y_i - \left( \hat{\beta}_0 + \hat{\beta}_1 x_{i,1} + \hat{\beta}_2 x_{i,2} + \boxed{?} + \hat{\beta}_k x_{i,k} \right) \right] = 0$$

$$\left. \frac{\partial SSE}{\partial \beta_j} \right|_{\hat{\beta}'s} = -2 \sum_{i=1}^{n} \left[ y_i - \left( \hat{\beta}_0 + \hat{\beta}_1 x_{i,1} + \hat{\beta}_2 x_{i,2} + \boxed{?} + \hat{\beta}_k x_{i,k} \right) \right] x_{i,j} = 0 \quad (j = 1, 2, \ldots, k)$$

i.e., solve $k+1$ eqns. in $k+1$ unknowns:

$$\begin{bmatrix} n & \sum x_{i1} & \sum x_{i2} & \boxed{?} & \sum x_{ik} \\ \sum x_{i1} & \sum x_{i1}^2 & \sum x_{i1} x_{i2} & \boxed{?} & \sum x_{i1} x_{ik} \\ \sum x_{i2} & \sum x_{i2} x_{i1} & \sum x_{i2}^2 & \boxed{?} & \sum x_{i2} x_{ik} \\ \boxed{?} & \boxed{?} & & \boxed{?} & \boxed{?} \\ \sum x_{ik} & \sum x_{ik} x_{i1} & \sum x_{ik} x_{i2} & \boxed{?} & \sum x_{ik}^2 \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \boxed{?} \\ \hat{\beta}_k \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_{i1} \\ \sum y_i x_{i2} \\ \boxed{?} \\ \sum y_i x_{ik} \end{bmatrix}$$

# Convenient Matrix Notation

Define:

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix}; \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ 1 & x_{31} & x_{32} & \cdots & x_{3k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix}; \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}; \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Model becomes:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon$$

LS solution becomes:

$$\left[\mathbf{X}^T\mathbf{X}\right]\hat{\beta} = \mathbf{X}^T\mathbf{Y}$$

If $\mathbf{X}^T\mathbf{X}$ invertible:

$$\hat{\beta} = \left[\mathbf{X}^T\mathbf{X}\right]^{-1}\mathbf{X}^T\mathbf{Y}$$

- **Tip:** Always pay attention to whether the quantities are scalars, vectors, or matrices, and their dimensions

# Assessing the Fit

- As in simple regression, calculate:

  **fitted values:** $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \boxed{?} + \hat{\beta}_k x_{ik} : \quad i = 1,2,\ldots,n$

  **residuals:** $e_i = y_i - \hat{y}_i : \quad i = 1,2,\ldots,n$

  **error sum of squares:** $SSE = \sum_{i=1}^{n} e_i^2$

  **total sum of squares:** $SST = \sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2$

  **regression sum of squares:** $SSR = \sum_{i=1}^{n} \left(\hat{y}_i - \bar{y}\right)^2$

- Still have same total sum of squares decomposition:

$$SST = SSR + SSE$$

# $r^2$ for Multiple Regression (beware though)

- We can still look at $r^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$

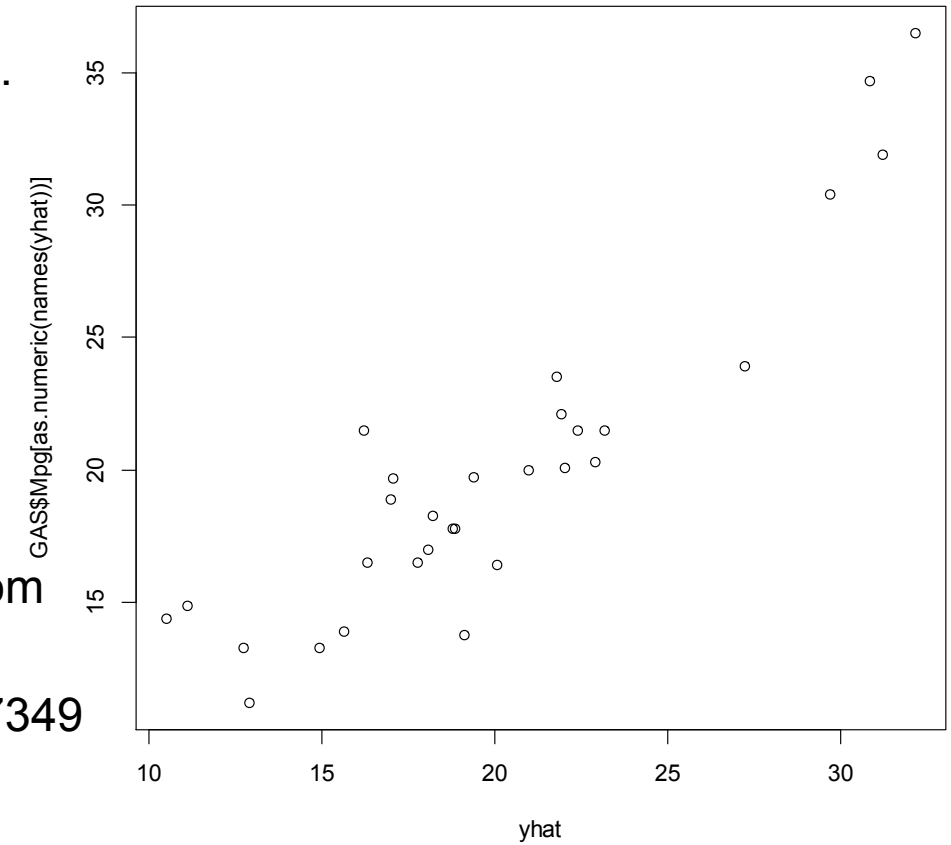
- In multiple regression, $r^2$ is called **coefficient of multiple determination**. It still represents the proportion of variability in $y$ that is accounted for by its linear dependence on the set of predictors.
- Mathematically, $r^2$ is equivalent to the square of the correlation coefficient between $y_i$ and $\hat{y}_i$

- **Beware**: $r^2$ is artificially high when $n \not\gg k$ because of overfitting – use something called "adjusted $r^2$" instead (coming up soon)

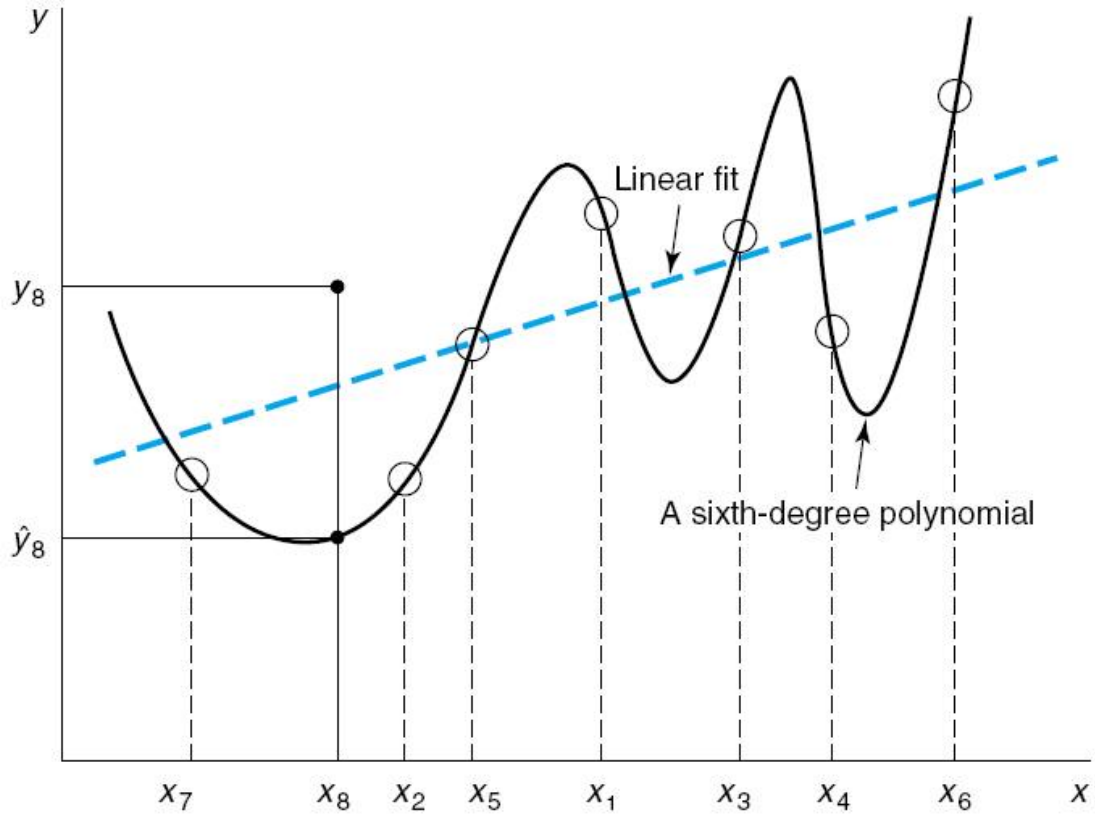
# Illustration of $r^2$ for the Mpg data

```
##########R code for illustrating basic regression fit to gas mileage data and r^2#####
GAS<-read.csv("gas_mileage.csv",header=TRUE)
pairs(GAS,cex=.5,pch=16)
##fit a linear regression model
lm1<-lm(Mpg~.,data=GAS)
summary(lm1)
yhat<-fitted(lm1)
plot(yhat,GAS$Mpg[as.numeric(names(yhat))])  #plot of y vs. fitted values
```

```
> summary(lm1)
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     17.339838  30.355375   0.571   0.5749
Displacement    -0.075588   0.056347  -1.341   0.1964
Hpower          -0.069163   0.087791  -0.788   0.4411
Torque           0.115117   0.088113   1.306   0.2078
Comp_ratio       1.494737   3.101464   0.482   0.6357
Rear_axle_ratio  5.843495   3.148438   1.856   0.0799 .
Carb_barrels     0.317583   1.288967   0.246   0.8082
No._speeds      -3.205390   3.109185  -1.031   0.3162
Length           0.180811   0.130301   1.388   0.1822
Width           -0.397945   0.323456  -1.230   0.2344
Weight          -0.005115   0.005896  -0.868   0.3971
Trans._type      0.638483   3.021680   0.211   0.8350

Residual standard error: 3.227 on 18 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared: 0.8355,     Adjusted R-squared: 0.7349
F-statistic:  8.31 on 11 and 18 DF,  p-value: 5.231e-05
```



# An Overfitting Example: Fitting a High-Order Polynomial with a Single Predictor




- With $n$ = 7, can get a perfect fit with 6*th* degree polynomial $\Rightarrow$ residuals are exactly zero $\Rightarrow$ $r^2 = 1$

# Illustration of Overfitting with Simulated Data

- The following code generates an array of completely random data with $n$ rows and $k$ predictor variables and fits a regression model
- What will happen if we use $k$ = 50 and $n$ = 40? Why?
- With $n$ = 40, what is the largest $k$ for which we can still fit the model and estimate all coefficients? What will $r^2$ be in this case?
- What will happen if we use $k$ = 30 and $n$ = 40?

```
#####R code for overfitting randomly generated data##################
k=30;n=40
x<-matrix(rnorm(k*n,0,1),n,k)
y<-rnorm(n,0,1)
overfit<-data.frame(y,x)
lm2<-lm(y~.,data=overfit)
summary(lm2)
yhat<-fitted(lm2)
plot(yhat,y)  #plot of y vs. fitted values
#####
rm(x,y,overfit,k,n)
```

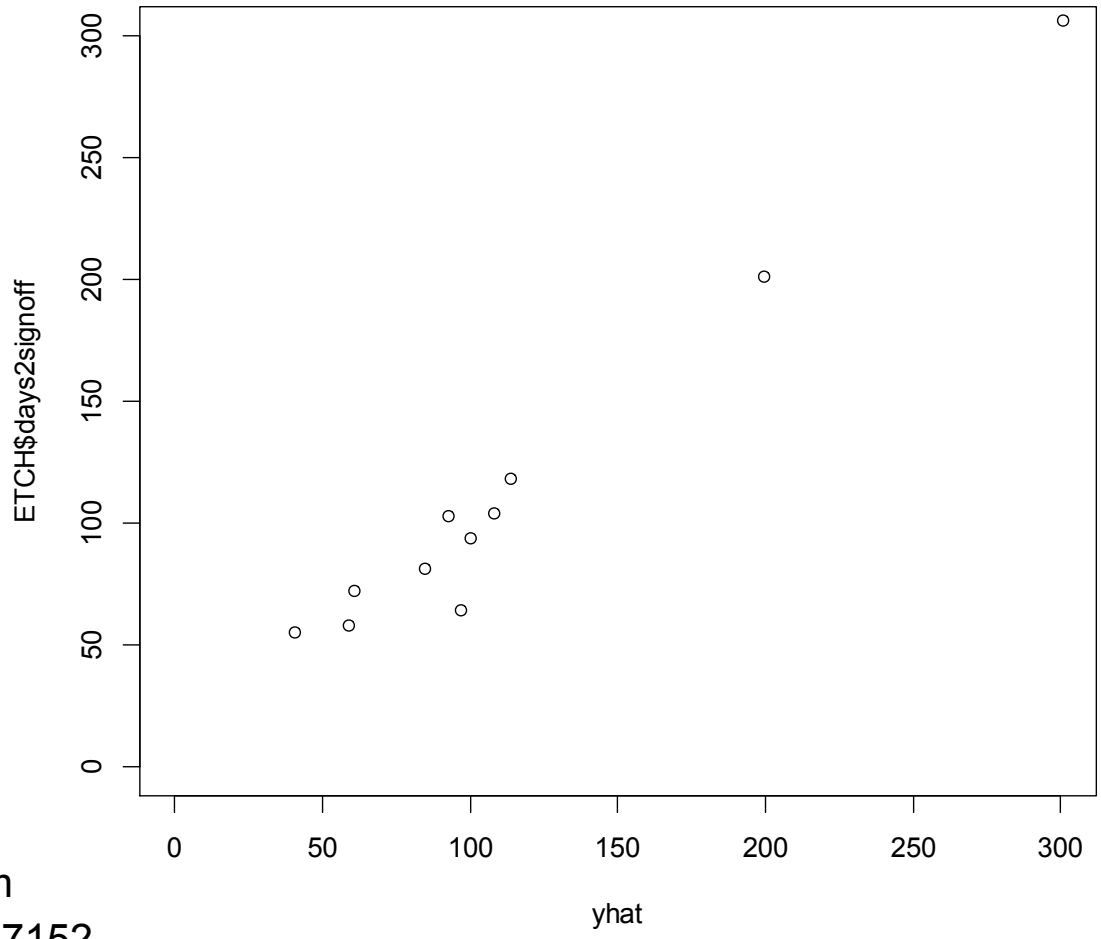
# A Real Overfitting Example (sil_etch.txt)

- A manufacturer of semiconductor etching machines wants to predict the number of days until the customer signs off on a received machine and pays the manufacturer (after shipping to customer, set up, trouble-shooting, fine tuning, etc, so that the machine is confirmed to work properly). This became extremely important following the Sarbanes-Oxley Act that tightened the rules on corporate accounting following the scandals of the late 1990's
- The idea is to predict days2signoff before the machine is shipped to the customer, based on quality-related predictor variables that are recorded during manufacturing
- sil_etch.txt contains the days2signoff (the response) and nine other predictors for a set of 11 machines that were manufactured, shipped and eventually signed-off (they produce many machines, but not many of each type, and they did not want to mix machines when shipping).
- Let's fit a multiple regression model regressing days2signoff onto all nine predictors and see how well the model predicts

# Fit a Multiple Regression to the ETCH Data

```
##########R code for fitting a multiple regression model to the ETCH data#####
ETCH<-read.table("sil_etch.txt", header=TRUE, sep="\t")
ETCH
lm1<-lm(days2signoff~.,data=ETCH)
summary(lm1)
yhat <- predict(lm1)
plot(yhat,ETCH$days2signoff, ylim=c(0,300), xlim=c(0,300))
data.frame(ETCH,round(yhat))
```

```
> summary(lm1)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 9993.23538 3950.70892   2.529    0.240
MNC          -32.23923   13.10992  -2.459    0.246
ISDR           0.05951   21.12661   0.003    0.998
DMR          -15.76123    4.70646  -3.349    0.185
PDSrev        16.03662   15.27724   1.050    0.485
NSR          121.27142   46.62543   2.601    0.234
UPSF         -29.01261   12.14763  -2.388    0.252
ILTR          -7.91838   10.52495  -0.752    0.589
BayDay       -49.91432   23.38041  -2.135    0.279
Test        -900.59213  362.52289  -2.484    0.244

Residual standard error: 40.3 on 1 degrees of freedom
Multiple R-squared: 0.9715,     Adjusted R-squared: 0.7152
F-statistic: 3.791 on 9 and 1 DF,  p-value: 0.3801
```



# Discussion Points and Questions

- How good does the fit appear to be for the ETCH data?
- Does it look like the fitted model for days2signoff has good predictive power?
- If you were the manufacturer, would you be comfortable using the model to predict days2signoff for machines you are about to ship?

# Definition of $r^2_{adj}$

- Recall: $r^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\left(\frac{SSE}{n-1}\right)}{\left(\frac{SST}{n-1}\right)}$
- Define the "mean squares" corresponding to the "sum of squares":

$$MSE = \frac{SSE}{n-(k+1)} = \text{unbiased estimate of } \sigma_\varepsilon^2$$

$$MST = \frac{SST}{n-1} = \text{unbiased estimate of } \sigma_Y^2$$

Scale things respect to degree of freedom.

← linear regression we have k+1 d.f

← we only estimate the mean. So d.f = 1

- For multiple regression, instead of $r^2$ you should look at "adjusted $r^2$":

$$r^2_{adj} = 1 - \frac{\hat{\sigma}_\varepsilon^2}{\hat{\sigma}_Y^2} = 1 - \frac{MSE}{MST} = 1 - \frac{SSE}{SST}\left[\frac{n-1}{n-k-1}\right]$$

# Discussion Points and Questions

- In multiple regression, $r^2_{adj}$ is interpreted as a better estimate (than $r^2$) of the percentage of variability in the response that is attributed to its linear dependence on the predictors
- But with severe overfitting, $r^2_{adj}$ can still be misleading if the error d.f. is very small
- What are $r^2_{adj}$ and $r^2$ for the GAS data? For the ETCH data? For the simulated random data with $k = 30$ and $n = 40$?
- Does $r^2_{adj}$ for the ETCH data seem reasonable?

# Statistical Inference on the Coefficients

- A regression fit can seem **practically significant** (high $r^2$) without being **statistically significant**, and vice-versa.
- Three common tests of whether individual parameters or groups of parameters differ from zero are:
  - $F$-test for testing whether at least one of the $k$ parameters differs from zero
  - $t$-tests and CIs for testing whether an individual parameter differs from zero (if so, the predictor has a statistically significant effect on the response)
  - Partial sum of squares $F$-test for testing whether at least one of a specified group of parameters differs from zero

# Overall F-test on All $k$ Coefficients

All of the statistical inference assumes a "true" model:

**observations:** $Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik} + \varepsilon_i : \quad i = 1, \ldots, n$

**random errors:** $\varepsilon_i \sim N(0,\sigma^2)$ and all i.i.d.

**"true" parameters:** $\beta_0, \ \beta_1, \ldots, \beta_k$

To test: $H_0: \ \beta_1 = \ldots = \beta_k = 0$

$H_1:$ at least one $\beta \neq 0$

using n features

using all the features.

degree of freedom: $k+1$

Use test statistic $F = \dfrac{MSR}{MSE}$ where $MSR = \dfrac{SSR}{k}$

degree of freedom is: $k$

Null distribution: $F \sim F_{k,n-(k+1)}$

↖ comparing $r^2_{adj}$ for the two models

↳ the distribution of test statistics under the Null Hypothesis.

Reject $H_0$ if $F > f_{k,n-k-1,\alpha}$

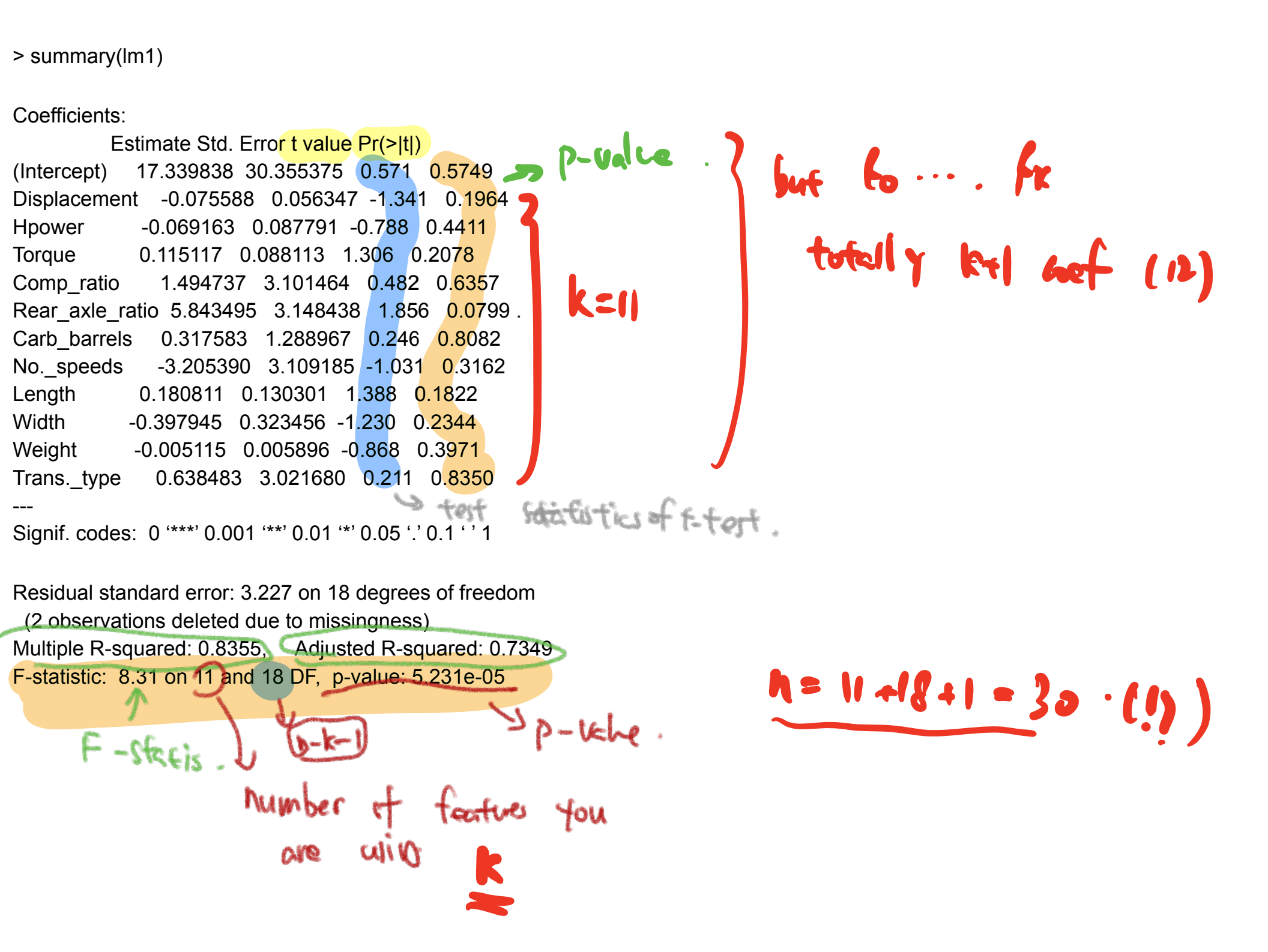
# F-test for the GAS data

```
##########R code for F-test with gas mileage data and r^2#####
GAS<-read.csv("gas_mileage.csv",header=TRUE)
n<-30
k<-11
lm1<-lm(Mpg~.,data=GAS)
summary(lm1)  #The F-test produced by the summary() command is the overall F-test
a <- anova(lm1); a  #This shows SSE, MSE, and other things
#The following does the same F-test manually
SSR <- sum(a[[2]][1:11])
SSE <- a[[2]][12]
MSR <- SSR/k
MSE <- SSE/(n-k-1)
F <- MSR/MSE
pf(F,k,n-k-1, lower.tail=FALSE) #P-value for F test
```

only have ↑ overall F-test

works for both overall F-test / partial F-test.

```
> summary(lm1)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    17.339838  30.355375   0.571   0.5749
Displacement   -0.075588   0.056347  -1.341   0.1964
Hpower         -0.069163   0.087791  -0.788   0.4411
Torque          0.115117   0.088113   1.306   0.2078
Comp_ratio      1.494737   3.101464   0.482   0.6357
Rear_axle_ratio 5.843495   3.148438   1.856   0.0799 .
Carb_barrels    0.317583   1.288967   0.246   0.8082
No._speeds     -3.205390   3.109185  -1.031   0.3162
Length          0.180811   0.130301   1.388   0.1822
Width          -0.397945   0.323456  -1.230   0.2344
Weight         -0.005115   0.005896  -0.868   0.3971
Trans._type     0.638483   3.021680   0.211   0.8350
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.227 on 18 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared: 0.8355,   Adjusted R-squared: 0.7349
F-statistic:  8.31 on 11 and 18 DF,  p-value: 5.231e-05
```



# Discussion Points and Questions

- Is the multiple regression fit to the GAS data statistically significant? Looks at p-value.
- In general, does strong statistical significance imply a strong predictability? Not

# Practical Versus Statistical Significance

triscan_5dx.txt contains quite a few observations of two variables related to measurement of solder paste volume in printed circuit board assembly. The response is FiveDX, which are volume measurements for a set of solder bricks using a machine based on X-ray technology. The predictor variable is Triscan, which are volume measurements of the same set of solder bricks using a machine based on laser scanning. The Triscan measurements are known to be quite accurate, but these measurements can only be obtained prior to placing the chips on the board. The FiveDX measurements can be obtained even after the chips are placed, but their accuracy is in question. The goal is to assess the accuracy of the FiveDX measurements by comparing it to the Triscan measurements. What is the conclusion?

```
#######R code#####
X<-read.table("triscan_5dx.txt",header=TRUE,sep="\t")
X[1:20,]
anova(lm(FiveDX~Triscan,data=X))
##
plot(X$Triscan,X$FiveDX); rm(X)
```

# Discussion Points and Questions

- If the $F$-test rejects $H_0$, an appropriate next step might be to determine which of the predictor variables (e.g., all of them, just a few, etc) have significant effects on the response
- Why might it be of interest to determine which predictors have significant effects?
- How would you formalize this as an hypothesis test?
- We can sometimes (but there is a big pitfall, discussed later) use a *t*-test on individual coefficients to determine which $\beta_j$'s $\neq 0$

Equations in this Slide is Not Required

# $t$-tests on Individual Coefficients

Just Reference

In order to develop a $t$-test on individual coefficients, we need the following statistical facts regarding the distribution of the estimated parameters $\hat{\beta}_j (j = 0, 1, \ldots, k)$:

For $j = 0, 1, \ldots, k$, $\quad \hat{\beta}_j \sim N\left(\beta_j, \sigma^2 v_{jj}\right)$

where $v_{jj}$ denotes the $(j+1)$st diagonal element of $\mathbf{V} = [\mathbf{X}^T\mathbf{X}]^{-1}$

i.e., $\hat{\beta}_j$ is normally distributed with $E\left(\hat{\beta}_j\right) = \beta_j$ and $SD\left(\hat{\beta}_j\right) = \sigma\sqrt{v_{jj}}$

Thus, a measure of precision in estimating $\beta_j$ is $SE\left(\hat{\beta}_j\right) = s\sqrt{v_{jj}}$

where $s^2 = MSE = \dfrac{SSE}{n-(k+1)}$ = unbiased estimate of $\sigma^2$

Additional Fact: $\dfrac{\hat{\beta}_j - \beta_j}{SE(\hat{\beta}_j)} \sim t_{n-k-1}: \quad j = 0,1,?,k$

Thus, a 2-sided $1-\alpha$ CI for $\beta_j$ is: $\hat{\beta}_j \pm t_{n-k-1,\alpha/2} SE(\hat{\beta}_j)$

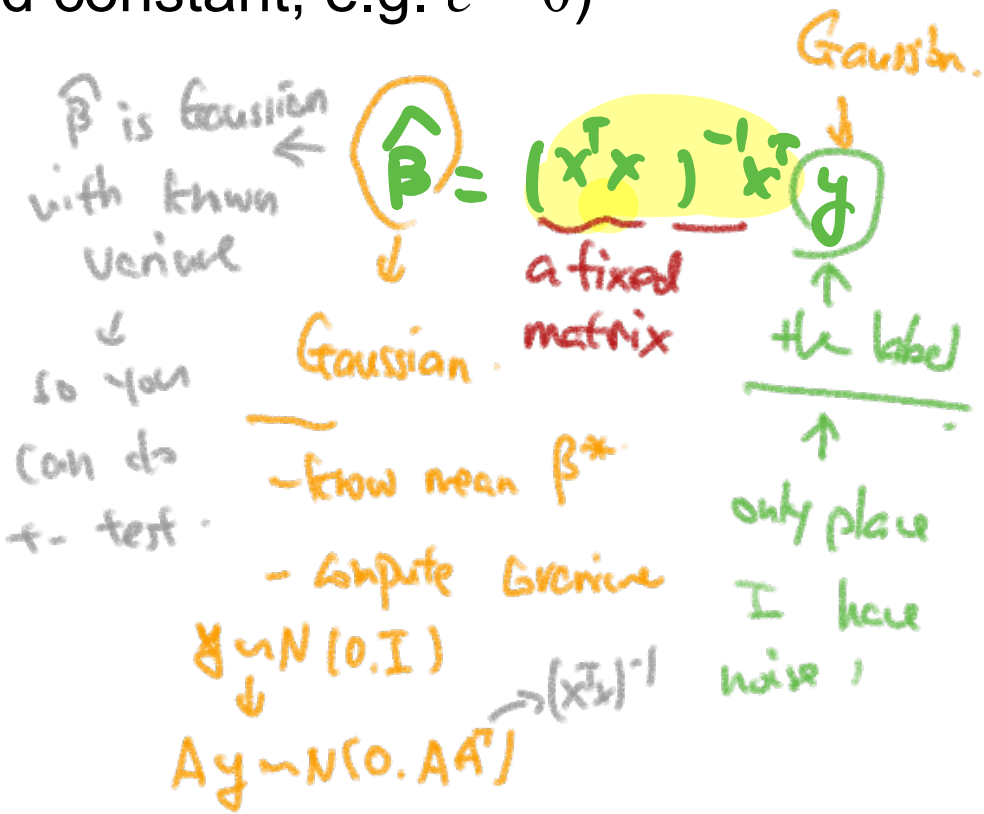
To test: $H_0: \beta_j = c$ (some specified constant, e.g. $c = 0$)

$H_1: \beta_j \neq c$

Use test statistic $t_j = \dfrac{\hat{\beta}_j - c}{SE(\hat{\beta}_j)}$

Null distribution: $t_j \sim t_{n-(k+1)}$

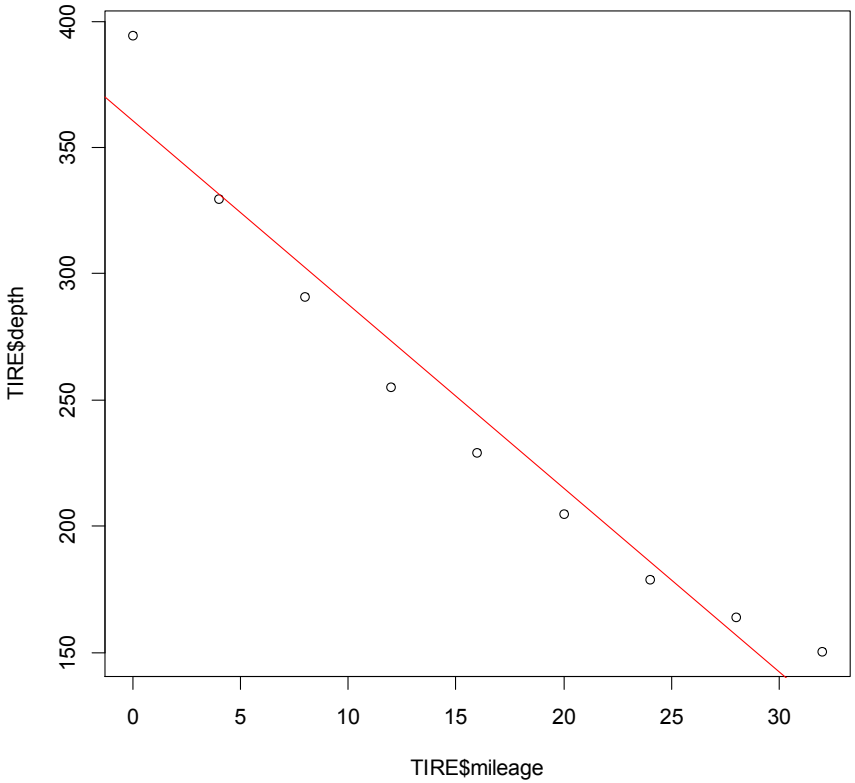
Reject $H_0$ if $|t_j| > t_{n-(k+1),\alpha/2}$



# t-tests for the Tire Wear Data

```
######R code for t-tests and CIs on tire wear data ####
TIRE<-read.table("tire_wear.txt",header=TRUE,sep="\t")
TIRE
plot(TIRE$mileage, TIRE$depth)
abline(lm(depth~mileage, data=TIRE), col="red") #plot of simple lin. regression
lm1<-lm(depth~poly(mileage,2, raw=TRUE), data=TIRE)
summary(lm1)
confint(lm1,level=.95)

##The following fits the same quadratic model
lm1<-lm(depth ~ mileage + I(mileage^2), data=TIRE)

##can calculate t-percentile via
qt(.975, 6)
```

```
> summary(lm1)
Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     386.26485    4.79996   80.47 2.48e-10 ***
poly(mileage, 2, raw = TRUE)1 -12.77238    0.69948  -18.26 1.74e-06 ***
poly(mileage, 2, raw = TRUE)2   0.17162    0.02103    8.16 0.000182 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.906 on 6 degrees of freedom
Multiple R-squared: 0.9961,     Adjusted R-squared: 0.9948
F-statistic: 762.8 on 2 and 6 DF,  p-value: 6.011e-08

> confint(lm1,level=.95)
                                  2.5 %      97.5 %
(Intercept)                     374.5197613 398.0099357
poly(mileage, 2, raw = TRUE)1 -14.4839431 -11.0608134
poly(mileage, 2, raw = TRUE)2   0.1201549   0.2230796
```



# Some Points and Pitfalls

- Usually begin with the overall $F$-test:
  - If $H_0$ not rejected, consider other predictors, nonlinear regression, or conclude there is no predictability and stop
  - If $H_0$ rejected, follow up by determining important predictors using t-tests on individual predictors (problematic with multicollinear predictors), partial F-tests on groups of predictors, or automated methods like stepwise or best subsets
- **Pitfall:** Beware interpreting individual $t$-tests when predictors are multicollinear, which is almost always. $P$-values will be misleadingly high. The reason is that the $t$-test of whether $\beta_j \neq 0$ is essentially testing whether including/excluding the individual predictor $x_j$ in the model significantly changes the $SSE$. E.g., the t-test for $\beta_1$ compares the following two models:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon \quad (\text{with } x_1), \text{ vs}$$

$$Y = \beta_0 + \qquad \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon \quad (\text{without } x_1)$$

# Individual t-tests for the GAS Data Illustrating the Pitfall

```
###########refit the gas mileage data regression with all predictors included######
GAS<-read.csv("gas_mileage.csv",header=TRUE)
pairs(GAS, cex = 0.5, pch = 16)  #matrix scatterplot
##fit a linear regression with all 11 predictors
lm1<-lm(Mpg~.,data=GAS)
summary(lm1)

##repeat with only Rear_axle_ratio and weight
lm1<-lm(Mpg~ Rear_axle_ratio + Weight,data=GAS)
summary(lm1)
```

```
> summary(lm1)
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    17.339838  30.355375   0.571   0.5749
Displacement   -0.075588   0.056347  -1.341   0.1964
Hpower         -0.069163   0.087791  -0.788   0.4411
Torque          0.115117   0.088113   1.306   0.2078
Comp_ratio      1.494737   3.101464   0.482   0.6357
Rear_axle_ratio 5.843495   3.148438   1.856   0.0799 .
Carb_barrels    0.317583   1.288967   0.246   0.8082
No._speeds     -3.205390   3.109185  -1.031   0.3162
Length          0.180811   0.130301   1.388   0.1822
Width          -0.397945   0.323456  -1.230   0.2344
Weight         -0.005115   0.005896  -0.868   0.3971
Trans._type     0.638483   3.021680   0.211   0.8350

Residual standard error: 3.227 on 18 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared: 0.8355,     Adjusted R-squared: 0.7349
F-statistic:  8.31 on 11 and 18 DF,  p-value: 5.231e-05
```

(?) (?)

If a feature is not correlated with the target the t-test will become not significant!!!


predict
all the factor I use for the prediction
Mpg
splaceme
Hpower
Torque
omp_rati
r_axle_ra
arb_barre
lo._speed
Length
Width
Weight
rans._typ
!!!
Multicollinearity will make t-test not significant even if the feature is correlated with the target!!

# the analogous results with only two predictors

try to drop multicollinearity!!!

```
> summary(lm1)

Call:
lm(formula = Mpg ~ Rear_axle_ratio + Weight, data = GAS)

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    31.7594958  5.8348313   5.443 7.41e-06 ***
Rear_axle_ratio  2.2141129  1.3146877   1.684    0.103
Weight         -0.0051025  0.0007106  -7.181 6.63e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.151 on 29 degrees of freedom
Multiple R-squared: 0.7674,    Adjusted R-squared: 0.7514
F-statistic: 47.84 on 2 and 29 DF,  p-value: 6.547e-10
```

droping feature it may helps!!

better than using all the features

become more significant

# Discussion Points and Questions

- Why are the coefficients not statistically significant when we include all 11 predictor variables?
- Why does Weight become much more significant when we fit the model with only Weight and Rear_axle_ratio included?

# Partial Sum of Squares $F$-test

A partial sum of squares $F$-test is for testing whether including/excluding a specified set of predictors together has a statistically significant effect on the response.

Partial model: $Y = \beta_0 + \beta_1 x_1 + \ldots + \beta_l x_l + \varepsilon$: $\quad l < k$

Full model: $Y = \beta_0 + \beta_1 x_1 + \ldots + \beta_l x_l + \ldots + \beta_k x_k + \varepsilon$

$SSE_k = SSE$ for full model

$SSE_l = SSE$ for partial model

To test: $H_0: \beta_{l+1} = \beta_{l+2} = \ldots = \beta_k = 0$

$H_1$: at least one extra $\beta_j \neq 0$ for $j > l$

Use test statistic: $$F = \frac{(SSE_l - SSE_k)/(k-l)}{SSE_k/(n-k-1)}$$

Comparing the $r^2_{adj}$.

Reject $H_0$ if $F > f_{k-l,n-k-1,\alpha}$

# Comments on Partial F-test

- The partial F-test is very flexible and can be used for testing whether any subset of coefficients are zero, not necessarily just for a subset of multicollinear predictors
- The partial F-test reduces to a t-test or the overall F-test in the following two special cases:
  - For $l = k-1$, the partial $F$-test is equivalent to the individual $t$-test on the one predictor that was left out
  - For $l = 0$, partial $F$-test is equivalent to overall $F$-test
- When you have a group of multicollinear predictors, the partial F-test can be used to test their collective significance, thereby avoiding the misleadingly high P-values in the individual t-tests on multicollinear predictors. However, stepwise or best subsets is generally preferred for this

# Partial F-Test for the GAS Data

```
###########Partial sum of squares F-test for gas mileage data######
GAS<-read.csv("gas_mileage.csv",header=TRUE)
pairs(GAS, cex = 0.5, pch = 16)  #matrix scatterplot
##fit the model with all 11 predictors
lmfull<-lm(Mpg~.,data=GAS[-c(23,25),]) #can also use the na.omit() command
anova(lmfull)  #produces SSE_full
##repeat but excluding the six strongly correlated predictors
lmreduced<-lm(Mpg~ Comp_ratio + Rear_axle_ratio + Carb_barrels + No._speeds +
        Trans._type, data=GAS[-c(23,25),])
anova(lmreduced) #produces SSE_reduced
anova(lmreduced,lmfull)  #this implements the partial F-test automatically
```

```
Model 1: Mpg ~ Comp_ratio + Rear_axle_ratio + Carb_barrels + No._speeds +
    Trans._type
Model 2: Mpg ~ Displacement + Hpower + Torque + Comp_ratio + Rear_axle_ratio +
    Carb_barrels + No._speeds + Length + Width + Weight + Trans._type
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     24 434.08
2     18 187.40  6    246.68 3.9489 0.01076 *
```

# Discussion Points and Questions

- Why did we omit rows 23 and 25 when fitting the two models (the na.omit command is helpful with large data sets)?
- How does the P-value for the partial F-test compare to the P-values for the individual t-tests on the six predictors?

# Using the Model for Prediction

- For a fixed set of predictor values $(x_1^*, x_2^*, \ldots, x_k^*)$ for a new case, two "future" things on which we may want to make inferences are:

  **actual response:** $Y^* = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \ldots + \beta_k x_k^* + \varepsilon$

  **response mean:** $\mu^* = E[Y^*] = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \ldots + \beta_k x_k^*$

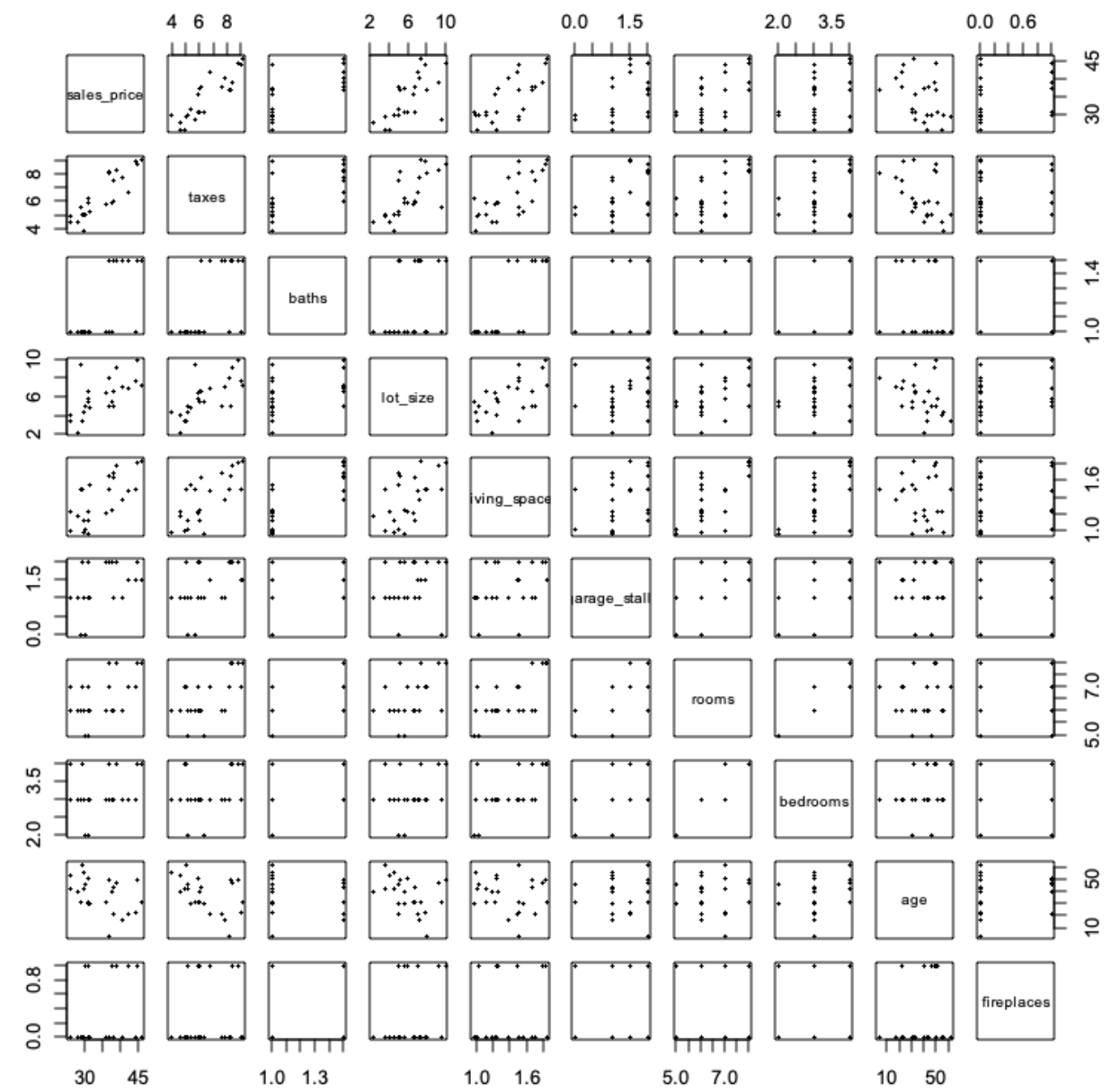
  (i.e., $\mu^*$ is the modeled component of the response)

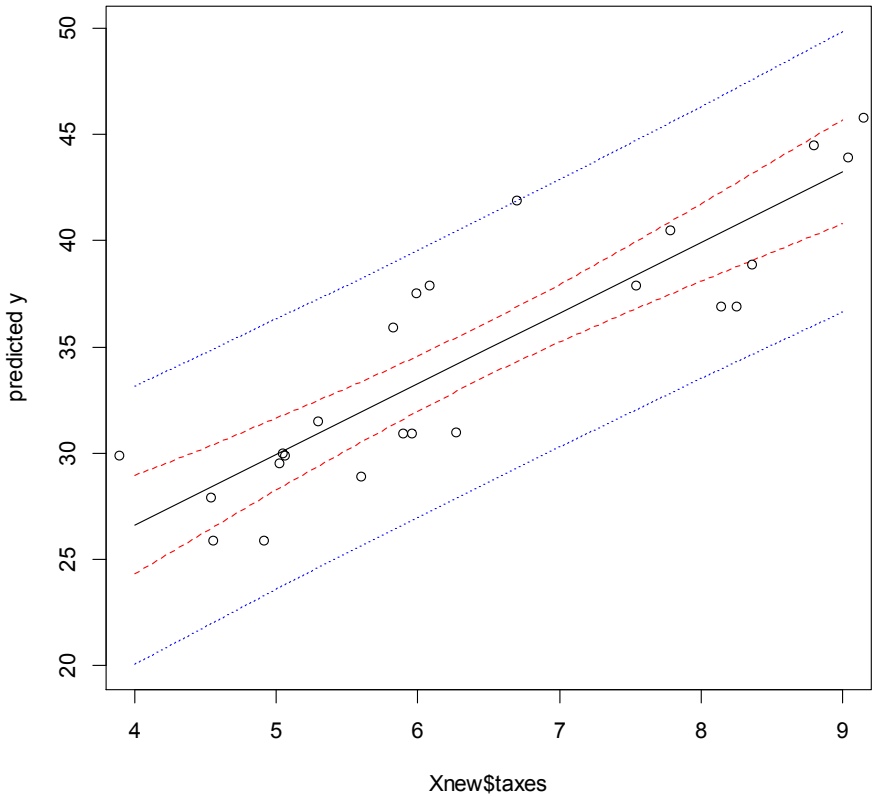
- The best **point prediction/estimate** are the same for both and are obvious (plug the predictors and the estimated coefficients into the model)
- If we want an interval that represents the uncertainty in the prediction/estimate, we use either:
  - A **CI on** $\mu^*$ (considers uncertainty in the $\beta$'s), or
  - A **PI on** $Y^*$ (considers uncertainty in the $\beta$'s and in $\varepsilon$)

# Example: Predicting Property Value – Illustration of PI on $Y^*$ vs. CI on $\mu^*$

- property_value.txt contains home sales prices and nine other characteristics (taxes, lot size, living space, age, etc) for a sample of 24 houses. The objective is to predict the sales price as a function of the other characteristics
- The following R code illustrates PIs and CIs for the simpler case of having only a single predictor taxes.

```
###R code for illustrating CIs and PIs with property value data with one predictor####
PROP<-read.table("property_value.txt",sep="\t",header=TRUE)
pairs(PROP, cex=0.5, pch=16)  #matrix scatterplot
PROP[1:10,]
Xnew <- data.frame(taxes = seq(4, 9, 0.5))
plim <- predict(lm(sales_price ~ taxes,data=PROP), newdata=Xnew, interval="prediction")
clim <- predict(lm(sales_price ~ taxes,data=PROP), newdata=Xnew,
       interval="confidence")
 matplot(Xnew$taxes,cbind(clim, plim[,-1]), lty=c(1,2,2,3,3),
       col=c("black","red","red","blue","blue"),type="l", ylab="predicted y")
points(PROP$taxes,PROP$sales_price)
```


sales_price
taxes
baths
lot_size
living_space
garage_stall
rooms
bedrooms
age
fireplaces


predicted y
Xnew$taxes

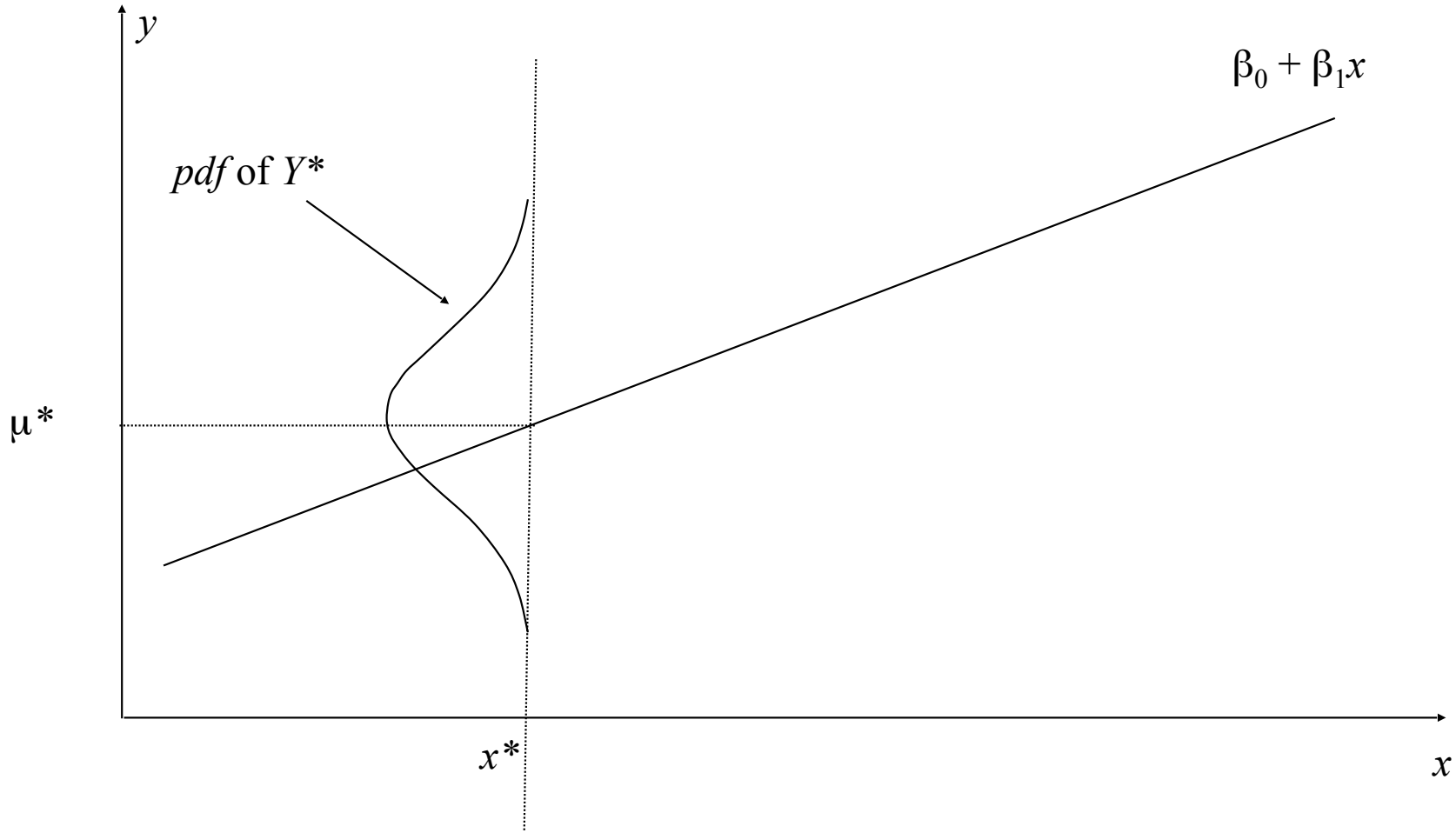
# Discussion Points and Questions

- Which is the PI and which is the CI in the previous figure?
- What is the interpretation of the PI?
- What is the interpretation of the CI?
- If someone is putting their house up for sale and wants to know the high end of the range for which it might sell, would the response PI or CI be more relevant?
- What is the relationship between the CI on $\mu^*$ versus a CI on one of the coefficients?
- How are the response CI and PI calculated?

# The Statistical View of $Y^*$

For fixed $\mathbf{x}^*$: $Y^* = \beta_0 + \beta_1 x_1^* + \ldots + \beta_k x_k^* + \varepsilon$

$= \mu^* + \varepsilon \quad \sim N(\mu^*, \sigma^2)$



# Point Estimate of $\mu^*$ (and Prediction of $Y^*$)

Define: $\mathbf{x}^* = [1\ x_1^*\ x_2^*\ \ldots\ x_k^*]^T$

Write: $\mu^* = \beta_0 + \beta_1 x_1^* + \ldots + \beta_k x_k^* = \mathbf{x}^{*T}\beta$

$$Y^* = \mu^* + \varepsilon = \mathbf{x}^{*T}\beta + \varepsilon$$

Point estimate of $\mu^*$: $\hat{\mu}^* = \mathbf{x}^{*T}\hat{\boldsymbol{\beta}}$

Point prediction of $Y^*$: $\hat{Y}^* = \hat{\mu}^* + \hat{\varepsilon} = \mathbf{x}^{*T}\hat{\boldsymbol{\beta}}$ (the same)

($\hat{\varepsilon}$ has expectation 0)

# Calculating a CI on $\mu^*$ and PI on $Y^*$

Two sources of uncertainty in future $Y^* = \mathbf{x}^{*\prime}\beta + \varepsilon$:

(1) Don't know true $\beta$
(2) Don't know future $\varepsilon$

To quantify (1), use the fact that $Var\left(\underbrace{\mathbf{x}^{*\prime}\hat{\boldsymbol{\beta}}}_{\hat{\mu}^*}\right) = \sigma^2\left(\mathbf{x}^{*\prime}\mathbf{V}\mathbf{x}^*\right)$

To quantify (2), use $Var(\varepsilon) = \sigma^2$

Hence, to quantify (1) + (2), use $Var\left(\hat{\mu}^* + \varepsilon\right) = \sigma^2\left(\mathbf{x}^{*\prime}\mathbf{V}\mathbf{x}^*\right) + \sigma^2$

2-sided 100(1–$\alpha$)% PI for $Y^*$: $\hat{Y}^* \pm t_{n-(k+1),\alpha/2}\, s\sqrt{1 + x^{*T}\mathbf{V}\,x^*}$

2-sided 100(1–$\alpha$)% CI for $\mu^*$: $\hat{\mu}^* \pm t_{n-(k+1),\alpha/2}\underbrace{s\sqrt{x^{*T}\mathbf{V}\,x^*}}_{SE(\hat{\mu}^*)}$

# Property Value Example Illustrating PI Calculations

To illustrate the PI calculations, consider only two predictors, and let's predict a new home with $x_1^* = 7, x_2^* = 1.5$

$\mathbf{x}^* = [1 \;\; 7 \;\; 1.5]^T$

$\hat{\boldsymbol{\beta}} = [10 \;\; 2.71 \;\; 6.16]^T$

Point prediction: $\hat{Y}^* = \hat{\mu}^* = \mathbf{x}^{*T}\hat{\boldsymbol{\beta}} = 38.28$

95% PI for $Y^*$:

$$\mathbf{V} = [\mathbf{X}'\mathbf{X}]^{-1} = \begin{bmatrix} 1.12 & -0.04 & -0.69 \\ -0.04 & .03 & -0.13 \\ -0.69 & -0.13 & 1.30 \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 1 & 5.02 & 1 \\ 1 & 4.54 & 1 \\ 1 & 4.56 & 1 \\ ? & ? & ? \end{bmatrix}$$

$x^{*T}\mathbf{V}x^* = 0.146$

$s = \sqrt{MSE} = \sqrt{7.8} = 2.79$

$t_{n-(k+1)\alpha/2} = t_{21,.025} = 2.08$

$$\hat{Y}^* \pm t_{n-(k+1)\alpha/2}\, s\sqrt{1 + x^{*T}\mathbf{V}x^*} = 38.28 \pm 2.08 * 2.79 * \sqrt{1 + 0.146} = [32.07, 44.5]$$

# Calculating PIs and CIs in R

```
##########R code for illustrating CIs and PIs with property value data##############
PROP<-read.table("property_value.txt",sep="\t",header=TRUE)
pairs(PROP[,1:3], cex=0.5, pch=16)  #matrix scatterplot
lm1<-lm(sales_price~.,data=PROP[,1:3])
summary(lm1)
Xnew<-data.frame(taxes=7,baths=1.5)
predict(lm1, newdata=Xnew, se.fit = T, level=0.95, interval = "confidence")
predict(lm1, newdata=Xnew, se.fit = T, level=0.95, interval = "prediction")

###manual calculations of some of the same thing###
s<-sqrt(anova(lm1)[[2]][3]/21)  #this is s, the sqrt of the MSE
X<-as.matrix(cbind(1,PROP[,2:3]))
x<-matrix(c(1,7,1.5),3,1)
V<-solve(t(X)%*%X)
SE<-s*sqrt(t(x)%*%V%*%x)  #this is SE of mu*
```

```
> predict(lm1, newdata=Xnew, se.fit = T, level=0.95, interval = "confidence")
$fit
       fit      lwr      upr
1 38.28195 36.06446 40.49944
$se.fit
[1] 1.066298
> predict(lm1, newdata=Xnew, se.fit = T, level=0.95, interval = "prediction")
$fit
       fit      lwr      upr
1 38.28195 32.06641 44.49749
$se.fit
[1] 1.066298
```