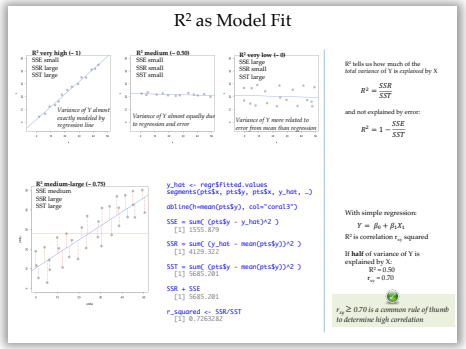
Business Analytics Using
Computational Statistics

Week 10
Linear Regression

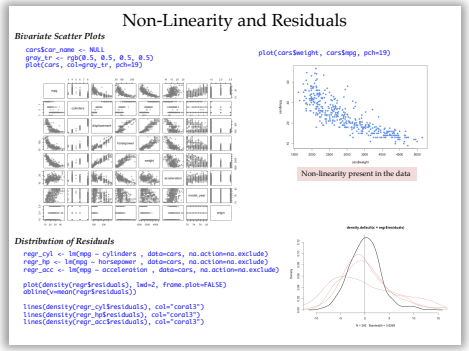
Week 11
Applied Regression

Week 12
Mediation and Moderation

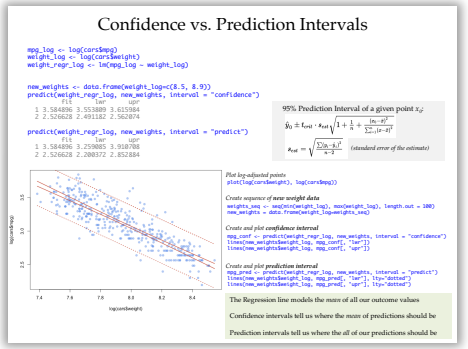
Model Fit

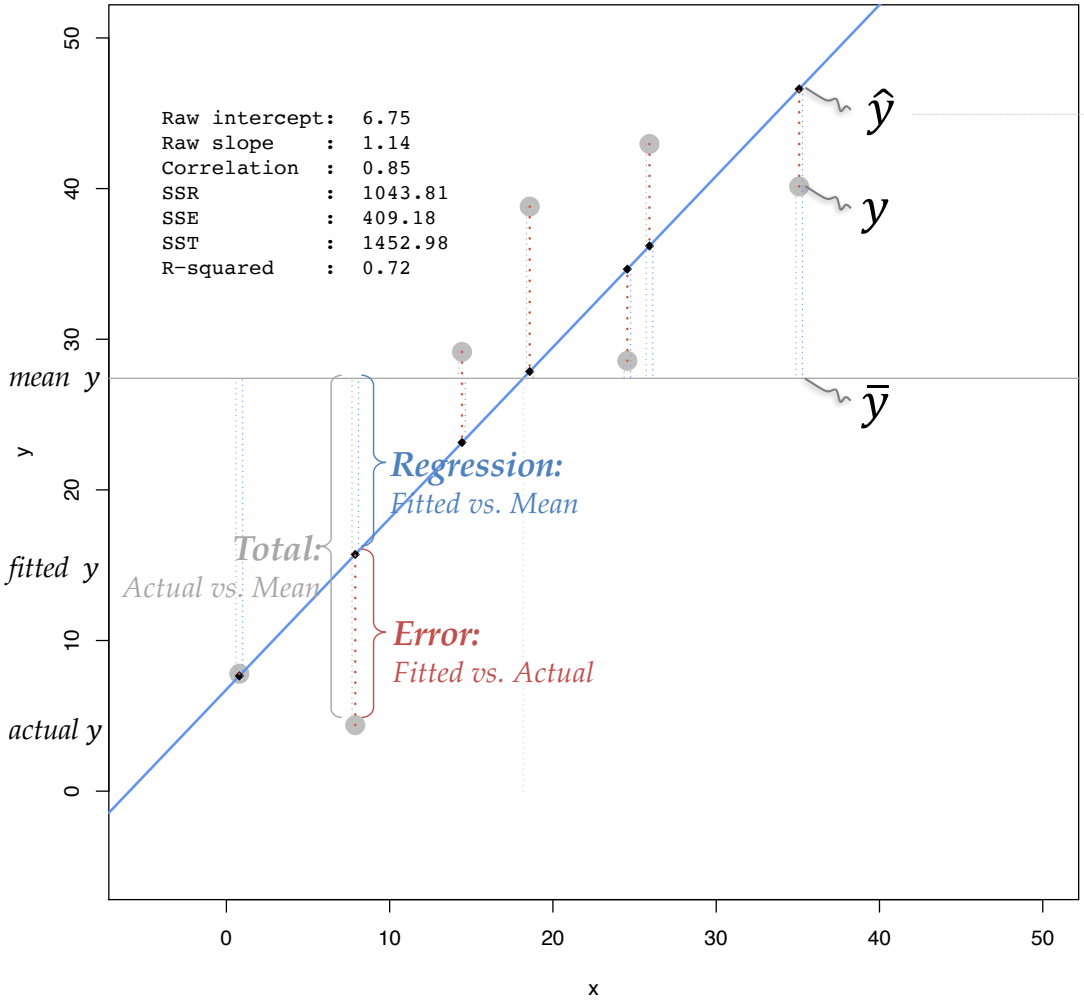


Non-Linearity & Multicollinearity



Confidence vs. Prediction Intervals






```
regr  <- lm(y ~ x, data=pts)
y_hat <- regr$fitted.values
points(pts$x, y_hat, pch=18)
```
"fitted" points $\hat{y}$

```
abline(h=mean(pts$y), …)
segments(pts$x, pts$y, pts$x, y_hat, …)
segments(pts$x-0.2, pts$y, pts$x-0.2, mean(pts$y), …)
segments(pts$x+0.2, y_hat, pts$x+0.2, mean(pts$y), …)

SSE = sum( (pts$y - y_hat)^2 )
[1] 409.1765

SSR = sum( (y_hat - mean(pts$y))^2 )
[1] 1043.807

SST = sum( (pts$y - mean(pts$y))^2 )
[1] 1452.983

SSR + SSE
[1] 1452.983

r_squared <- SSR/SST
[1] .7183887
```

R^2 tells us how much of the
total variance of Y is *explained by the regression model*

$$R^2 = \frac{SSR}{SST}$$

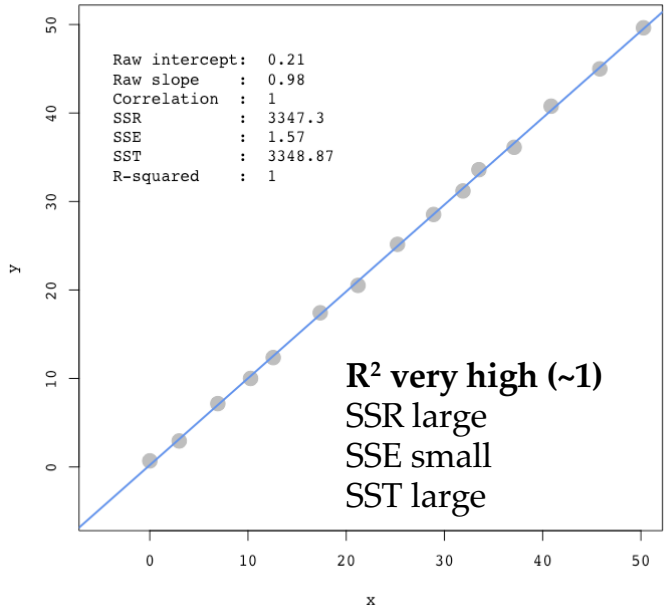
or, how much of the
total variance is *not simply error variance*:

$$R^2 = 1 - \frac{SSE}{SST}$$

$R^2 = 0.72$: *Variance of Y is* ***mostly*** *explained by X*

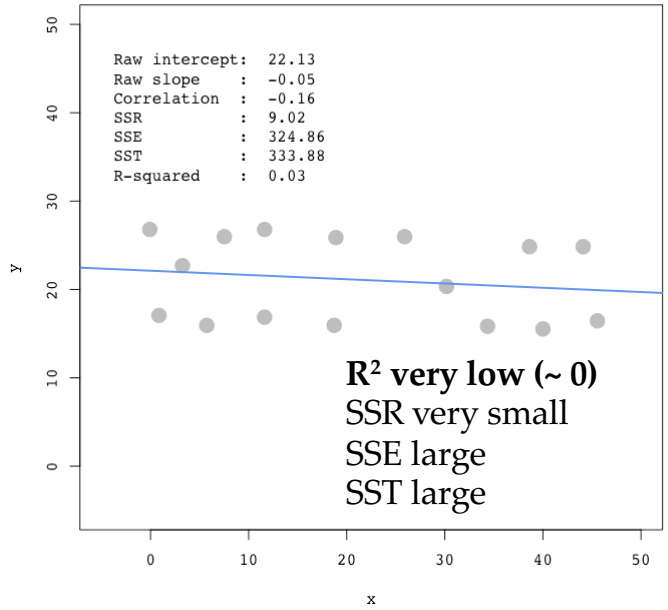
Interpreting R^2

R^2 as Model Fit



R^2 very high (~1)
SSR large
SSE small
SST large

*Variation of Y almost exactly explained by **regression** Model*



R^2 very low (~ 0)
SSR very small
SSE large
SST large

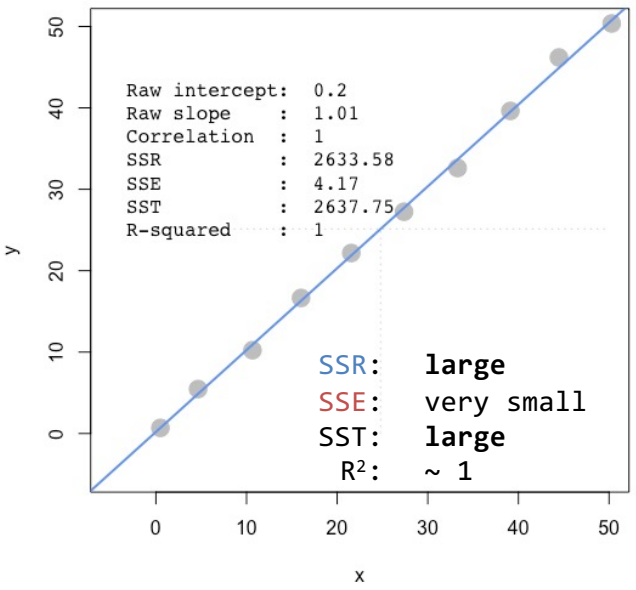
*Variance of Y more related to **error** from mean than regression*

With simple regression:
$Y = \beta_0 + \beta_1 X_1$
R^2 is correlation r_{xy} squared

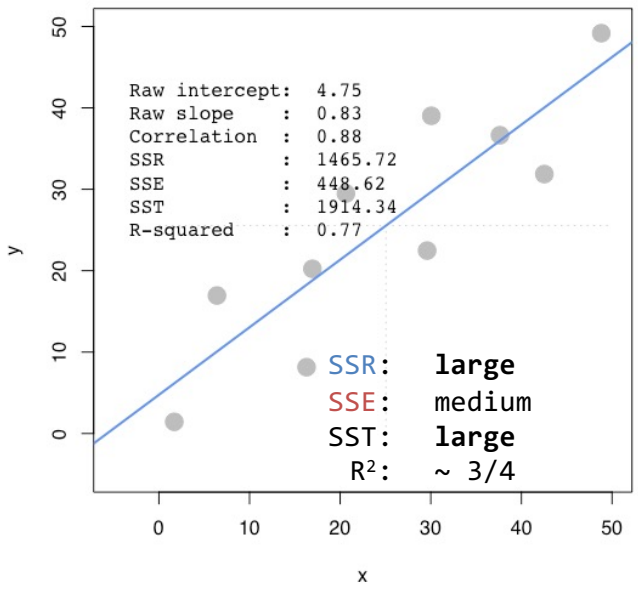
If half of variance of Y is explained by X:
$R^2 = 0.50$ implies $r_{xy} = 0.70$

$r_{xy} \geq 0.70$ is a common rule of thumb to determine high correlation; it suggests that one variable at least half explains the variance of the other

Scenario 1

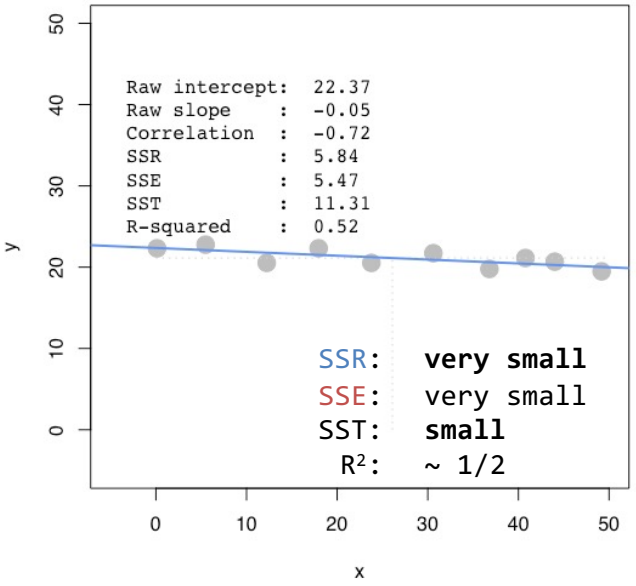


Scenario 2

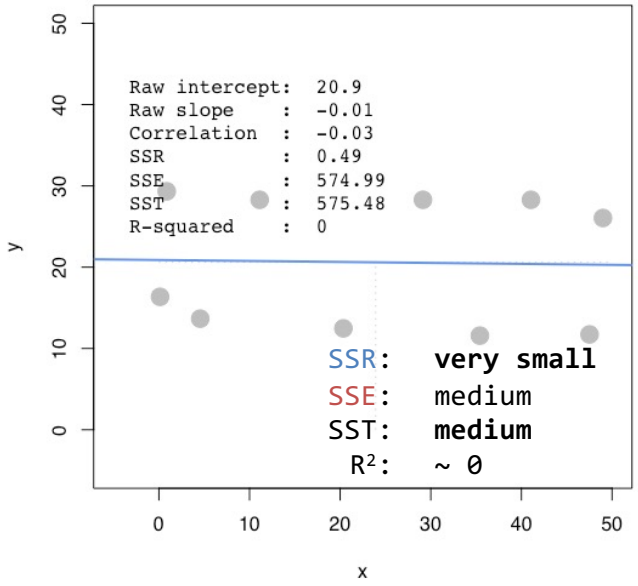


$$R^2 = \frac{SSR}{SST}$$

Scenario 3



Scenario 4



Regression Using Linear Algebraic

Constant values of 1 to capture the intercept

$$\boldsymbol{X} = \begin{bmatrix} 1 & 4 & 78 & 0 \\ 1 & 7 & 100 & 1 \\ 1 & 1 & 86 & 0 \\ . & . & . & . \\ . & . & . & . \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} 24.0 \\ 43.0 \\ 23.7 \\ . \\ . \end{bmatrix}$$

Intercept term

$$\widehat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T y = \begin{bmatrix} \widehat{\beta_0} \\ \widehat{\beta_1} \\ \widehat{\beta_2} \\ . \\ . \end{bmatrix}$$

$$\widehat{\boldsymbol{y}} = X\hat{\beta}$$

$$\boldsymbol{R^2} = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

Variance Explained
This is how much of our data's total variance is explained by our regression model

```
prog <- read.csv("programmer_salaries.txt", sep="\t")

x_vars <- c("Experience", "Score", "Degree")
X <- cbind("(intercept)"=1, as.matrix(prog[, x_vars]))
```

```
        (intercept) Experience Score Degree
 [1,]           1          4    78      0
 [2,]           1          7   100      1
  .             .          .     .      .
[20,]           1          3    89      0
```

```
y <- prog[, "Salary"]
```

```
 [1] 24.0 43.0 ...
30.0
```

```
beta_hat <- solve(t(X)%*%X) %*% t(X)%*%y
```

```
              [,1]
(intercept) 7.944849
Experience  1.147582
Score       0.196937
Degree      2.280424
```

Regression Coefficients
These are the estimated slopes of our regression model

```
y_hat <- X %*% beta_hat
```

```
          [,1]
 [1,] 27.89626
 [2,] 37.95204
  .     .
[20,] 28.91499
```

Fitted Values
These are estimated values of y from our regression model

```
SSR <- sum((y_hat - mean(y))^2) # 507.896
SSE <- sum((y - y_hat)^2)       #  91.890
SST <- sum((y - mean(y))^2)     # 599.786

Rsq <- SSR/SST # 0.847
```

The "Hat" Matrix

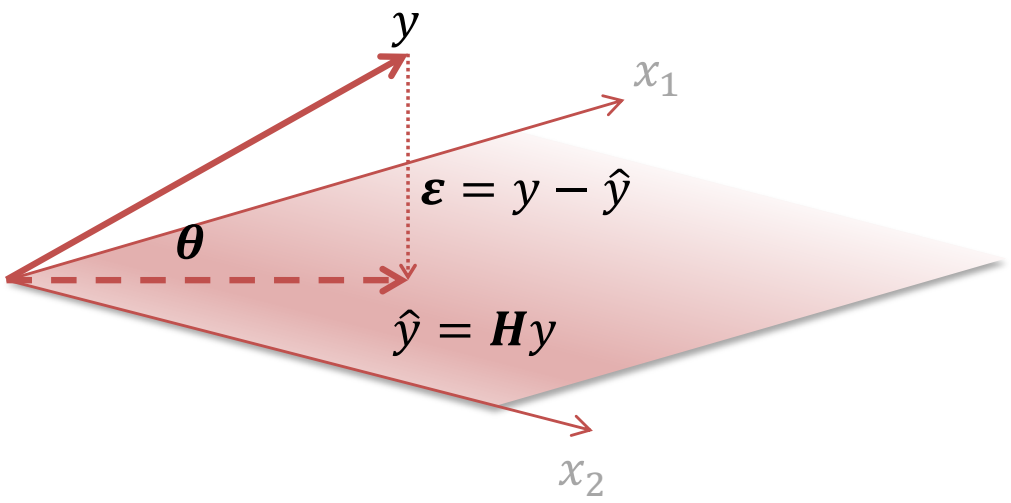
Another way of looking at regression is that it is trying to orthogonally project $\boldsymbol{y}$ *to the space spanned by* $\boldsymbol{X}$ *in "row space"*

$$\boldsymbol{H} = X(X^T X)^{-1} X^T$$

```
H <- X %*% solve(t(X)%*%X) %*% t(X)
```

$$\hat{\boldsymbol{y}} = \boldsymbol{H}y$$

```
y_hat <- H %*% y
```



H puts the "hat" on y

H orthogonally projects y onto the space of X
ŷ is the projection of y onto the space of X

$$R = cor(y, \hat{y})$$

```
R   <- cor(y, y_hat)      # 0.920   Multiple Correlation
```

$$R^2 = cor(y, \hat{y})^2$$

```
Rsq <- cor(y, y_hat)^2    # 0.847   Squared Multiple Correlation
```

Regression Coefficients

Our estimated regression coefficients are based on our one sample.
They have variance (uncertainty) associated with them…

Simple Regression

$$y = \beta_o + \boldsymbol{\beta_1} x_1$$

$$\beta_1 = r_{yx_1}$$

Multiple Regression

$$y = \beta_o + \boldsymbol{\beta_1} x_1 + \beta_2 x_2$$

$$\beta_1 = \frac{r_{yx_1} - r_{yx_2} r_{x_1 x_2}}{1 - r_{x_1 x_2}}$$

Each beta coefficient b_i could vary with standard error s_{b_i}

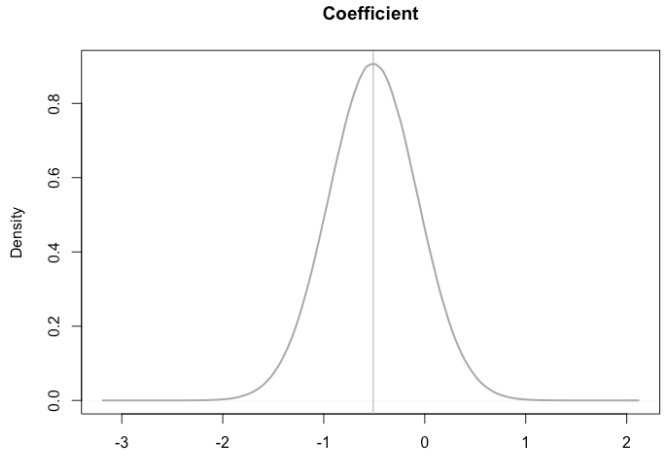
Standard Error of Regression

$$s_e = \sqrt{MSE} = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n-2}}$$

Standard Error of coefficient b_i

$$s_{\beta_1} = \frac{s_e}{\sum (x_i - \bar{x})^2}$$

(for simple regression)

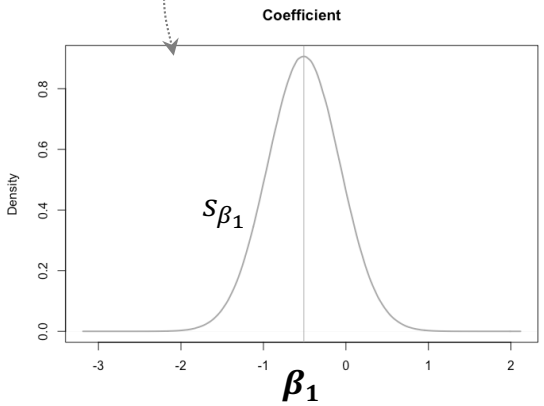


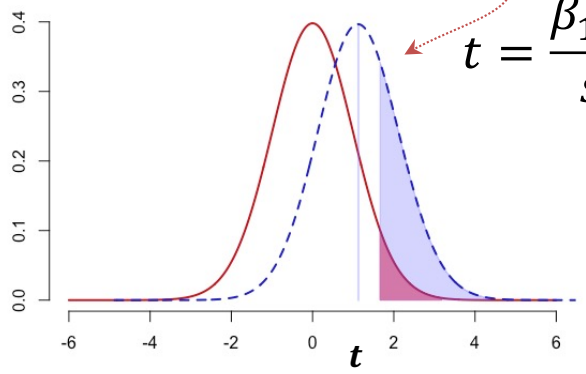
Coefficients and t-Tests

```
engine_regr <- lm(mpg ~ cylinders + displacement, data=cars, na.action=na.exclude)

summary(engine_regr)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.444769   1.192171  30.570  < 2e-16 ***
cylinders    -0.517357   0.442434  -1.169    0.243
displacement -0.052258   0.007218  -7.240 2.36e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```





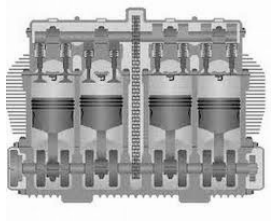
$$t = \frac{\widehat{\beta_1} - 0}{s_{\beta_1}}$$

Distribution of cylinders coefficient:

- mean value -0.51
- standard deviation of mean: 0.44 *(standard error)*

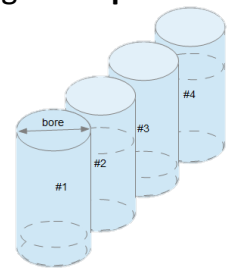
t-test of H_{null}: coefficient is 0

- t-value: -1.169
- p-value: 0.243 *(two-tailed)*



Engine **Cylinders**

Engine **Displacement**



Multiple Regression

```
regr <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model_year + factor(origin),
           data=cars, na.action=na.exclude)
```

```
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -1.795e+01  4.677e+00  -3.839 0.000145 ***
cylinders       -4.897e-01  3.212e-01  -1.524 0.128215
displacement     2.398e-02  7.653e-03   3.133 0.001863 **
horsepower      -1.818e-02  1.371e-02  -1.326 0.185488
weight          -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
acceleration     7.910e-02  9.822e-02   0.805 0.421101
model_year       7.770e-01  5.178e-02  15.005  < 2e-16 ***
factor(origin)2  2.630e+00  5.664e-01   4.643 4.72e-06 ***
factor(origin)3  2.853e+00  5.527e-01   5.162 3.93e-07 ***
---
Multiple R-squared:  0.8242,   Adjusted R-squared:  0.8205
```

Unstandardized Raw Coefficients have units
(mpg/dependent units)

Easier to relate coefficient to domain:
"-6.71 mpg per pound (lb.)"

But hard to compare coefficients:
"0.77 mpg/year" versus "-6.71 mpg/lb"

```
summary(cars$weight)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1613    2224    2804    2970    3608    5140

summary(cars$cylinders)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.000   4.000   4.000   5.455   8.000   8.000
```

Standardized Multiple Regression

```
regr_std <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model_year + factor(cars$origin),
               data=as.data.frame(scale(cars)), na.action=na.exclude)
```

should we standardize categorical variable?

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          -0.13323    0.03174  -4.198 3.35e-05 ***
cylinders            -0.10658    0.06991  -1.524  0.12821
displacement          0.31989    0.10210   3.133  0.00186 **
horsepower           -0.08955    0.06751  -1.326  0.18549
weight               -0.72705    0.07098 -10.243  < 2e-16 ***
acceleration          0.02791    0.03465   0.805  0.42110
model_year            0.36760    0.02450  15.005  < 2e-16 ***
factor(cars$origin)2  0.33649    0.07247   4.643 4.72e-06 ***
factor(cars$origin)3  0.36505    0.07072   5.162 3.93e-07 ***
---
Multiple R-squared:  0.8242,   Adjusted R-squared:  0.8205
```

NOTE: Coefficient significances and model fit (R^2) have not changed

Standardized coefficients are in standard deviations

Relative interpretation of coefficients:
"-0.72 sd of mpg per sd weight"

Easier to compare coefficients?
"-0.72 sd change from weight"
"0.36 sd change from years"

Categorical Variables

Two-level categorical variable: use a single dummy variable

Independent Variables *Dependent Variable*

Dummy Variable

Experience	Score	Degree	Salary
4	78	0	24
7	100	1	43
1	86	0	23.7
…	…	…	…
3	89	0	30

Dummy variable:
absence or presence of a criteria

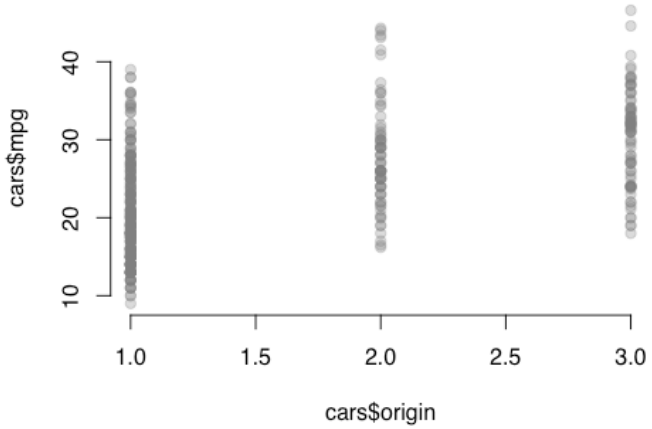
```
summary(with(prog, lm(Salary ~ Experience + Score + Degree))

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.9448     7.3808   1.076   0.2977
Experience    1.1476     0.2976   3.856   0.0014 **
Score         0.1969     0.0899   2.191   0.0436 *
Degree        2.2804     1.9866   1.148   0.2679
```

coefficient of dummy variable:
change in Salary (dependent) when Degree (dummy) is present

Multilevel Categorical Variable: use multiple (k-1) dummy variables

```
plot(cars$origin, cars$mpg)
```




```
summary( lm(mpg ~ factor(origin), data=cars) )

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      20.0835     0.4056  49.517   <2e-16 ***
factor(origin)2   7.8079     0.8658   9.018   <2e-16 ***
factor(origin)3  10.3671     0.8264  12.544   <2e-16 ***
```

Multilevel categorical (k levels): *k-1 dummies*

coefficient of dummy variables:
`factor(origin)2` *(JP): change in mpg relative to origin 1 (US)*
`factor(origin)3` *(EU): change in mpg relative to origin 1 (US)*

Regression Assumptions

Requirements for Regression:

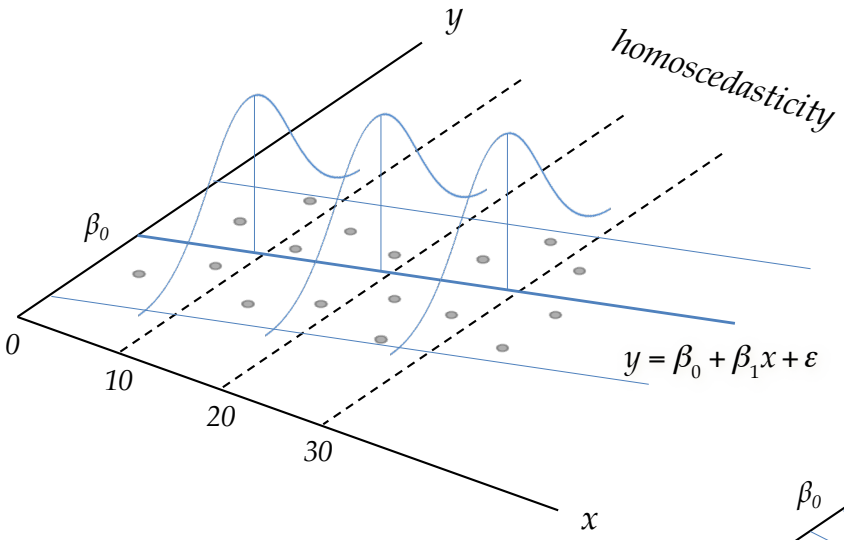
1. *The error terms (ε) are: random, normally distributed, and have a mean of zero*
2. *The variance of ε is the same for all values of x*
3. *The values of ε are independent across values of x*

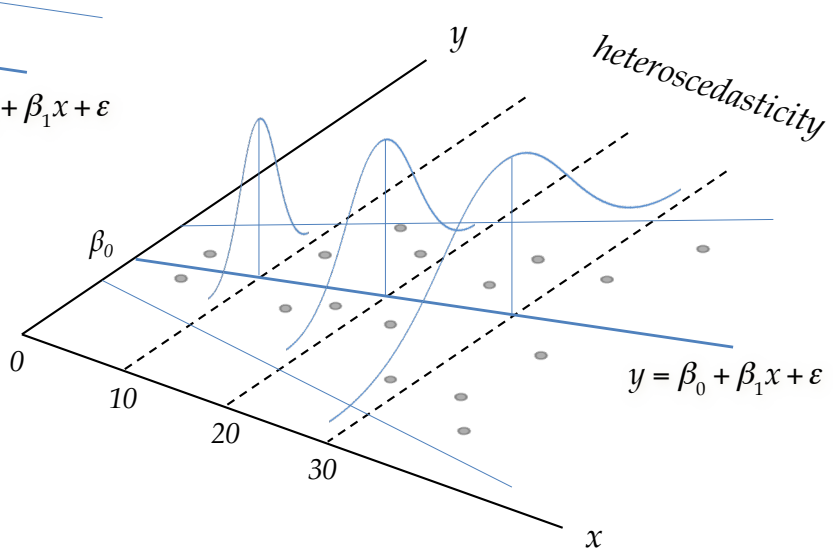
Implications:

The mean values of y are on the regression line
Standard errors of $\hat{\beta}$ are symmetrically distributed

The distribution of y is the same across values of x

The values of y are independent of each other



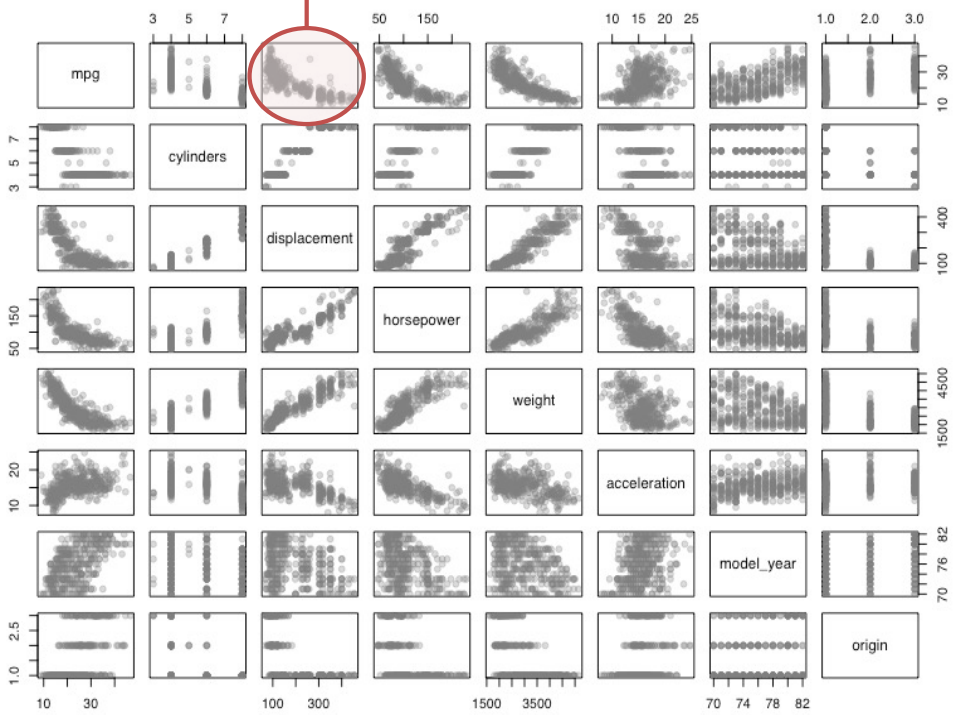


Diagnosing our Regression

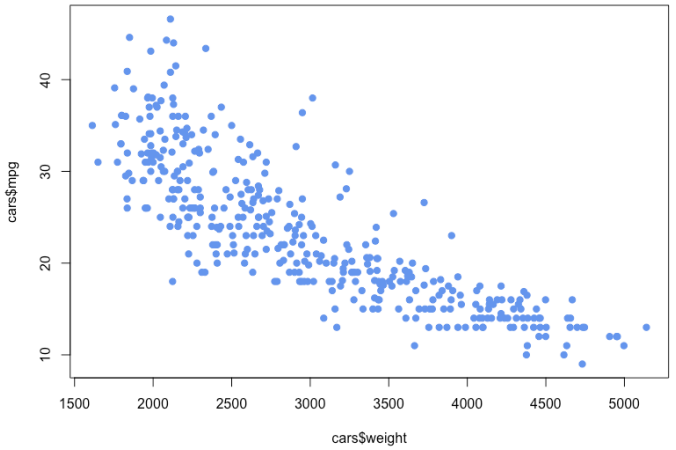
Bivariate Scatter Plots

```
cars$car_name <- NULL
plot(cars)
```

Displacement trend is opposite of regression coefficient




```
plot(cars$weight, cars$mpg, pch=19)
```



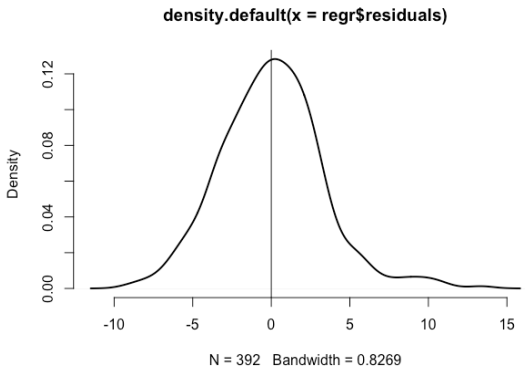
Non-linearity present in the data

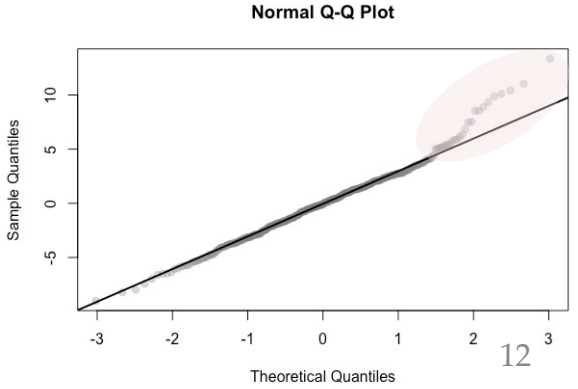
Normality of Residuals (assumption 1)

```
# density plot of residuals
plot(density(regr$residuals), …)
abline(v=mean(regr$residuals))

# normal quantiles vs. empirical quantiles
qqnorm(regr$residuals, …)
qqline(regr$residuals, …)
```

Residuals are not normally distributed

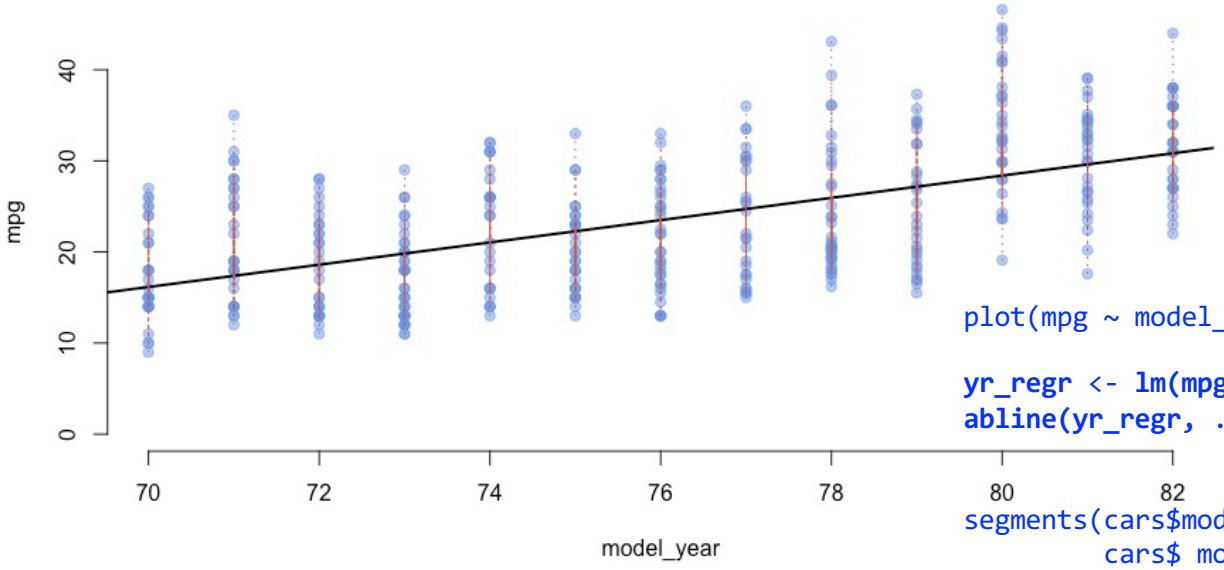




Linear Relationships

Regressing mpg over model_year

`lm( mpg ~ model_year )`



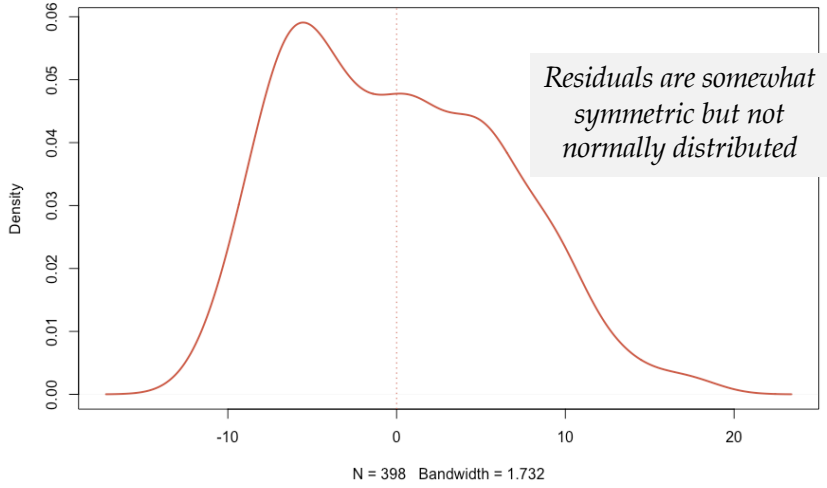

```
plot(mpg ~ model_year, data=cars, ...)

yr_regr <- lm(mpg ~ model_year, data=cars, na.action=na.exclude)
abline(yr_regr, ...)

segments(cars$model_year, cars$mpg,
         cars$ model_year, predict(hp_regr), ...)
```

Distribution of residuals

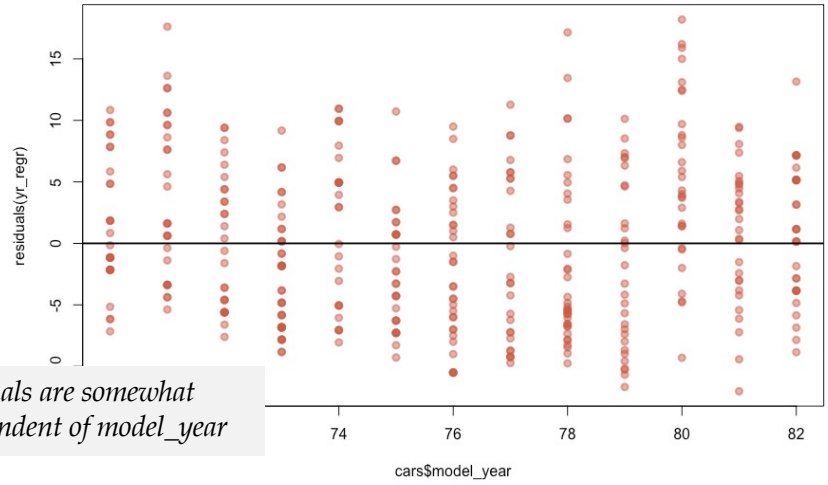
```
plot(density(yr_regr$residuals))
```



Residuals are somewhat symmetric but not normally distributed

Residuals as a function of independent variable

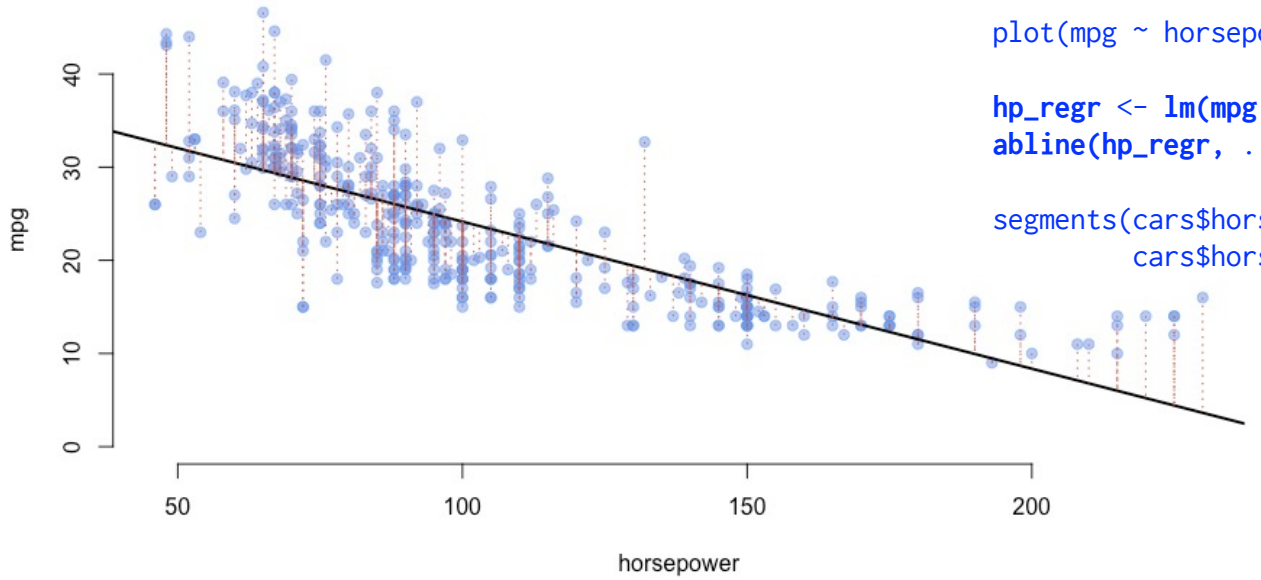
```
plot(cars$model_year, resid(yr_regr), col="red", lwd=2)
abline(h=0)
```



Residuals are somewhat independent of model_year

Non-linear Relationships

Regressing mpg over horsepower



`lm( mpg ~ horsepower )`

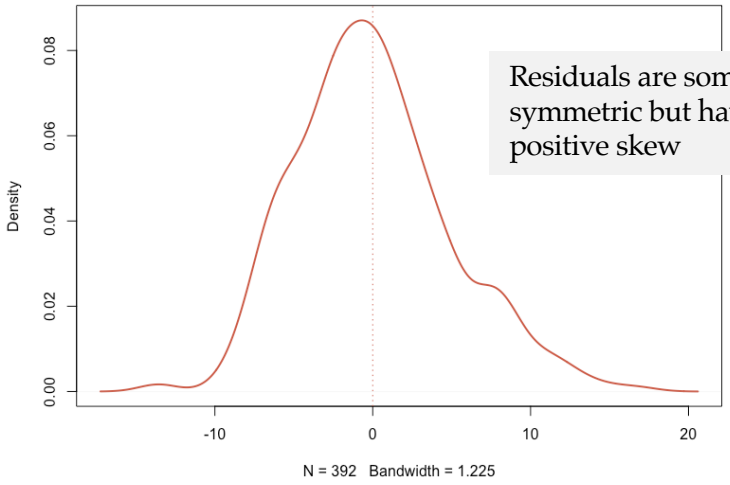
```
plot(mpg ~ horsepower, data=cars, ...)

hp_regr <- lm(mpg ~ horsepower, data=cars, na.action=na.exclude)
abline(hp_regr, ...)

segments(cars$horsepower, cars$mpg,
         cars$horsepower, predict(hp_regr), ...)
```

Distribution of residuals

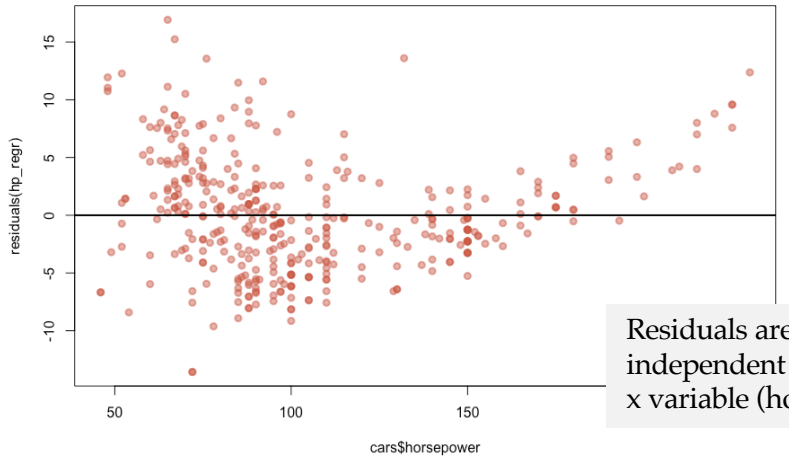
```
plot(density(hp_regr$residuals))
```



Residuals are somewhat symmetric but have a positive skew

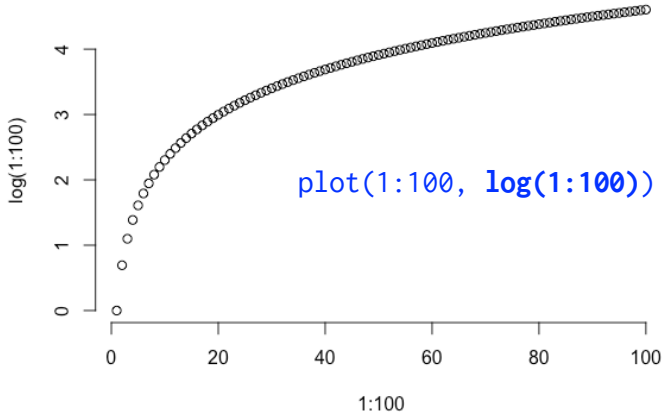
Residuals as a function of independent variable

```
plot(cars$horsepower, resid(hp_regr), col="red", lwd=2)
abline(h=0)
```



Residuals are not independent of our x variable (horsepower)

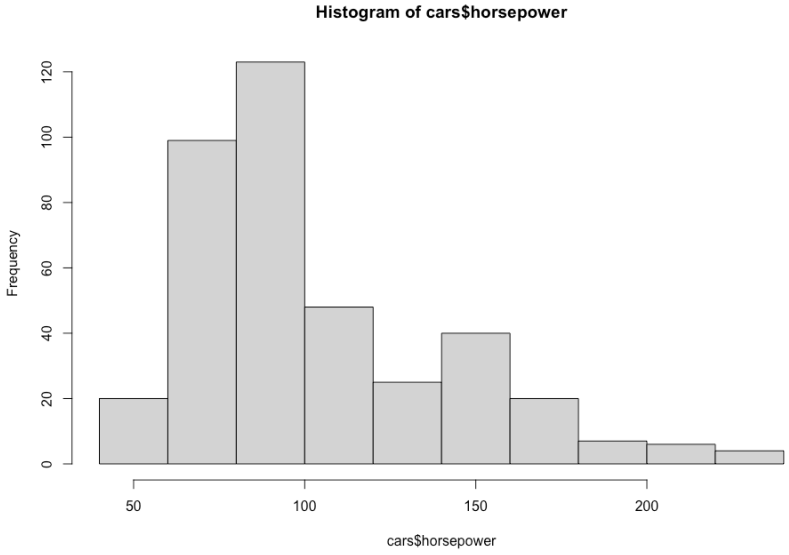
Log Transforming Variables



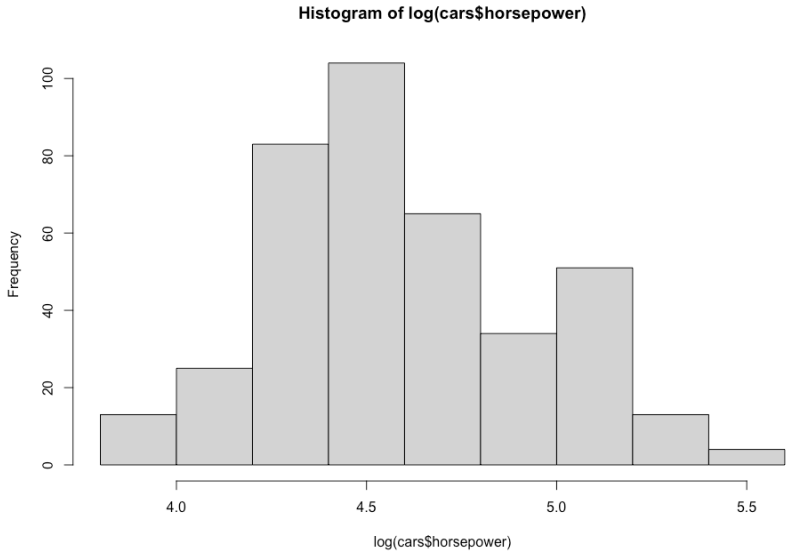
***Log**arithm is the inverse of exponentiation*

*Taking the **log** of variables can give them more symmetric distributions*

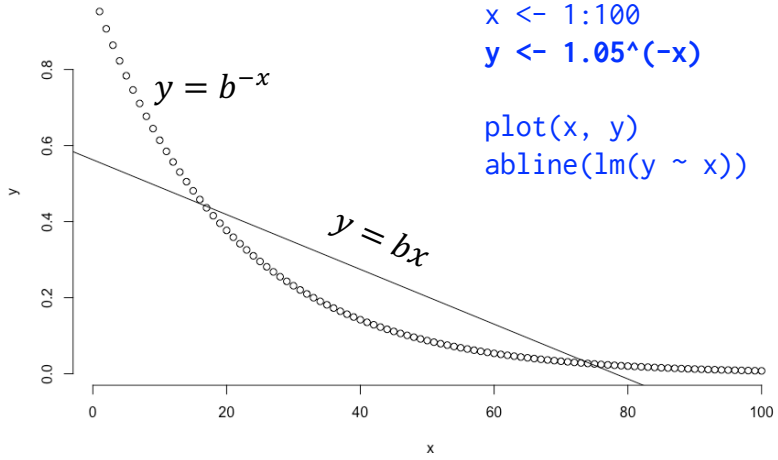
```
hist( cars$horsepower )
```




```
hist( log(cars$horsepower) )
```



Non-linear model: $y = b^{-x}$

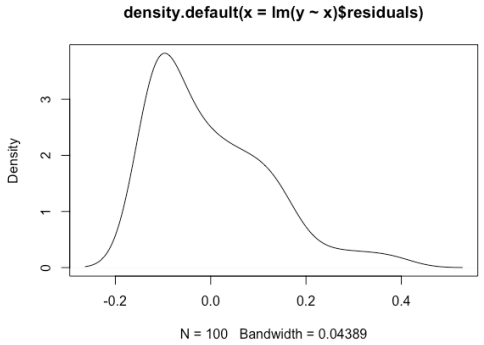



```
x <- 1:100
y <- 1.05^(-x)

plot(x, y)
abline(lm(y ~ x))
```

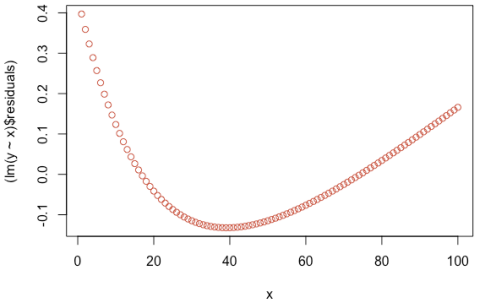
Non-linear relationships pose problems for linear regression

residuals are skewed




```
plot(density(lm(y ~ x)$residuals))
```

residuals have relationship with independent variable



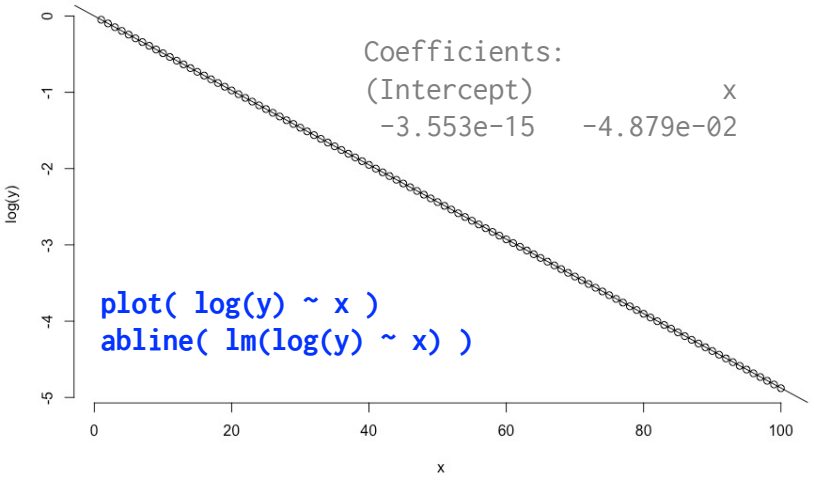

```
plot(x, (lm(y ~ x)$residuals))
```

Linear model: $\log(y) = bx$

Log transformation: $log(b^x) = log(b) \cdot x$

```
lm(log(y) ~ x)

Coefficients:
(Intercept)           x
 -3.553e-15   -4.879e-02
```




```
plot( log(y) ~ x )
abline( lm(log(y) ~ x) )
```

*We can **log transform** one or both sides of our regression to get linear relationships*

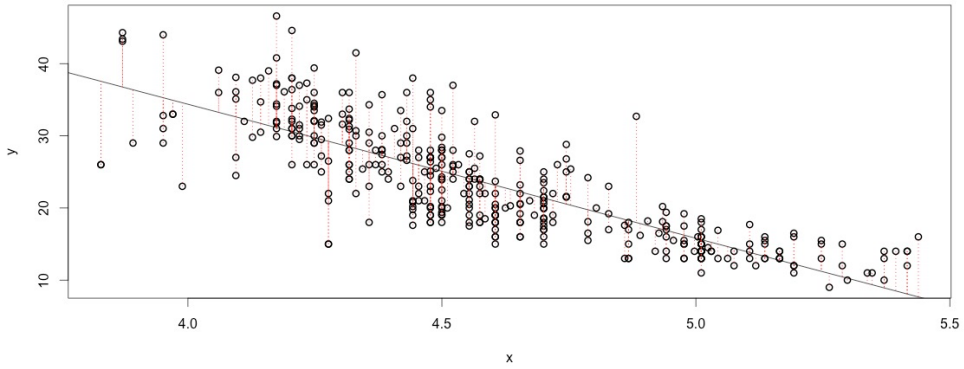
`lm( y ~ log(x) )`

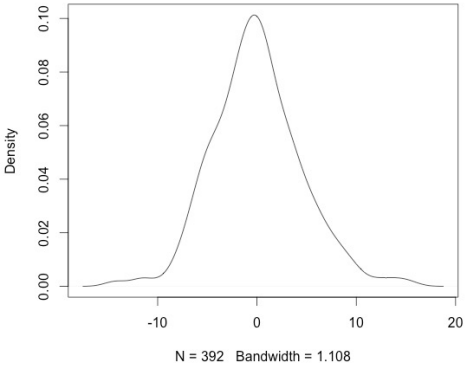
1% change in X leads to β/100 units change in Y

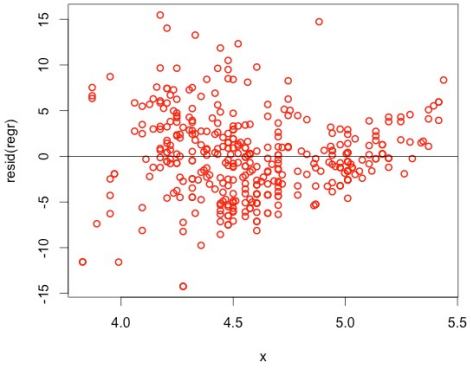
`lm( mpg ~ log(horsepower) )`

```
Coefficients:
    (Intercept)  log(horsepower)
         108.70          -18.58
```

1% more horsepower leads to 0.18 decrease in mpg







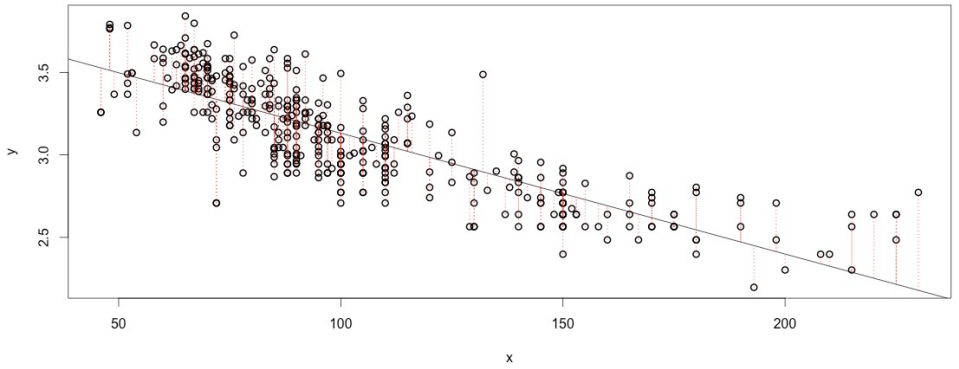
`lm( log(y) ~ x )`

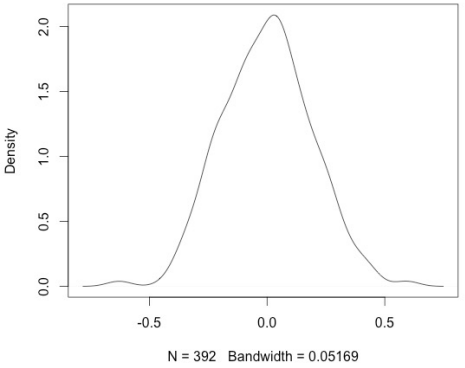
one unit change in X leads to β∗100% change in Y

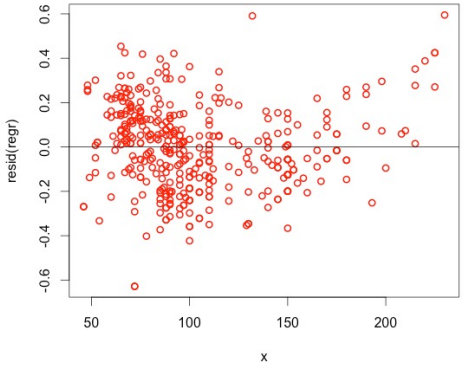
`lm( log(mpg) ~ horsepower )`

```
Coefficients:
(Intercept)   horsepower
   3.864467    -0.007334
```

One more horsepower leads to 0.73% decrease in mpg





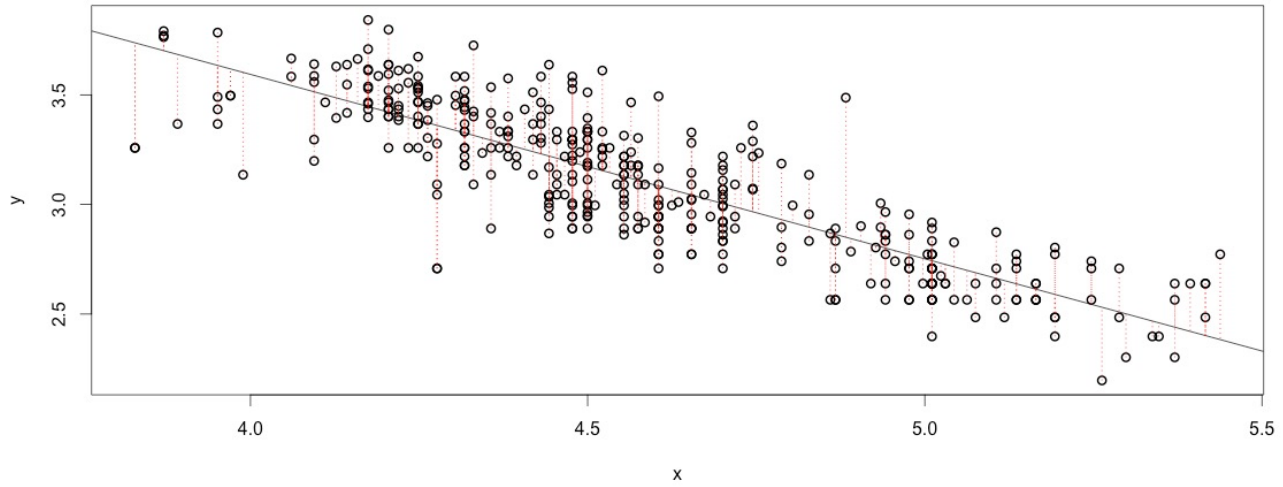


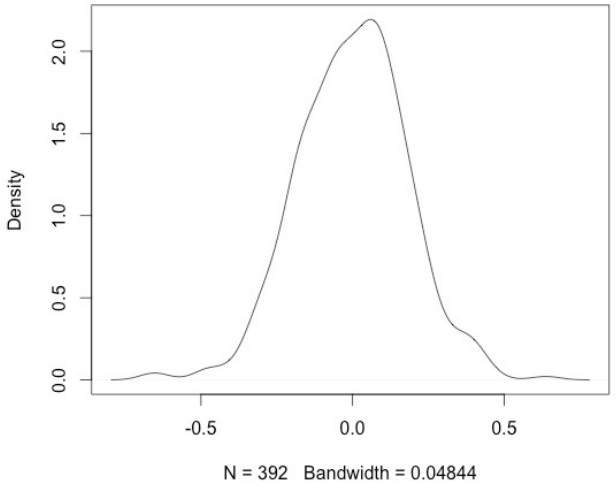
`lm( log(y) ~ log(x) )`

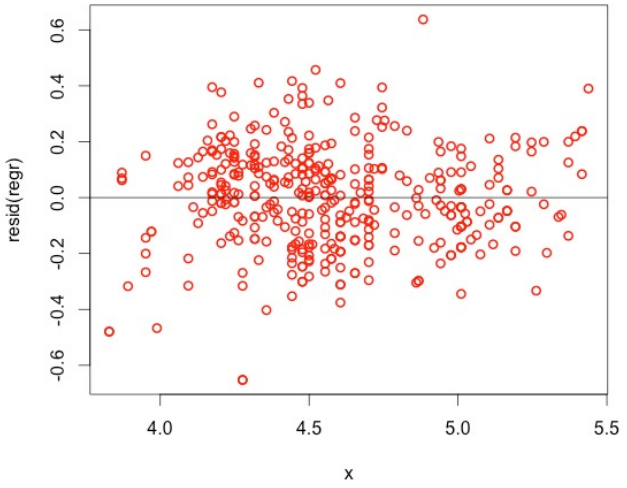
1% change in X leads to β% change in Y

`lm( log(mpg) ~ log(horsepower) )`

```
Coefficients:
    (Intercept)  log(horsepower)
         6.9606          -0.8418
```







Multicollinearity

Remove 8th column ("country of origin")

```
round( cor(cars[, -8], use="pairwise.complete.obs"), 2 )
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year
mpg	1.00	-0.78	-0.80	-0.78	-0.83	0.42	0.58
cylinders	-0.78	1.00	0.95	0.84	0.90	-0.51	-0.35
displacement	-0.80	0.95	1.00	0.90	0.93	-0.54	-0.37
horsepower	-0.78	0.84	0.90	1.00	0.86	-0.69	-0.42
weight	-0.83	0.90	0.93	0.86	1.00	-0.42	-0.31
acceleration	0.42	-0.51	-0.54	-0.69	-0.42	1.00	0.29
model_year	0.58	-0.35	-0.37	-0.42	-0.31	0.29	1.00

weight is the only significant variable among collinear set

High multicollinearity:
cylinders, displacement, horsepower, weight

$$mpg = \beta_1 displacement$$

```
regr_disp <- lm(mpg ~ displacement, ...)
summary(regr_disp)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.931e-17  2.983e-02    0.00        1
displacement -8.042e-01  2.987e-02  -26.93   <2e-16 ***
```

$$mpg = \beta_1 displacement + \beta_2 weight$$

```
regr_disp_wt <- lm(mpg ~ displacement + weight, ...)
summary(regr_disp_wt)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.831e-17  2.762e-02   0.000  1.00000
displacement -2.182e-01  7.674e-02  -2.844  0.00469 **
weight       -6.282e-01  7.674e-02  -8.186 3.77e-15 ***
```

Symptoms of Multicollinearity
When new independent terms added:
- *Change in size of coefficient*
- *Change in standard error of coeffient (change in std error, t-value, p-value)*

Variance Inflation Factor (VIF)

Diagnosing Multicollinearity

Given a regression with k independent variables:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \beta_k X_k$$

```
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.795e+01  4.677e+00  -3.839 0.000145 ***
cylinders     -4.897e-01  3.212e-01  -1.524 0.128215
displacement   2.398e-02  7.653e-03   3.133 0.001863 **
```

	cylinders
Coefficient (slope)	**-0.4897**
Std. Error of slope	**0.3212**
Variance of coeff.	**0.1032**

```
regr <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model_year + factor(origin), …)
```

Multicollinearity inflates the variance of coefficients by: $\boldsymbol{VIF_j = \frac{1}{1 - R_j^2}}$

R_j^2 *is the* R^2 *of regressing the* j^{th} *independent variable over all other k-1 independent variables:*

$$\boldsymbol{X_j = b_0 + b_2 X_1 + \cdots + b_k X_k}$$

inflated variance reduces significance of coefficients

e.g., VIF of cylinders in our regression:

$$cylinders = b_0 + b_1 displacement + \cdots + b_k\ origin$$

```
cylinder_regr <- lm(cylinders ~ displacement + horsepower + weight + acceleration + model_year + factor(origin),
                    data=cars, na.action=na.exclude)

r2_cylinders <- summary(cylinder_regr)$r.squared
  [1] 0.9068708

vif_cylinders <- 1 / (1 - r2_cylinders)
  [1] 10.73777

sqrt(vif_cylinders)
  [1] 3.276854
```

R_{cyl}^2 *The variance of cylinders, shared by other covariates*

High multicollinearity:

$VIF > 4$ *or* 5

$\sqrt{VIF} > 2$

X_k *shares more than half its variance with other independent variables*

What can we do about highly collinear independent variables?