

QCB 508 – Week 9

John D. Storey

Spring 2017

Contents

	4
Statistical Models	4
Probabilistic Models	4
Multivariate Models	4
Variables	4
Statistical Model	4
Parametric vs Nonparametric	5
Simple Linear Regression	5
Ordinary Least Squares	5
Generalized Least Squares	6
Matrix Form of Linear Models	6
Least Squares Regression	6
Generalized Linear Models	6
Generalized Additive Models	7
Some Trade-offs	7
Bias and Variance	7
Motivating Examples	8
Sample Correlation	8
Example: Hand Size Vs. Height	8
Cor. of Hand Size and Height	9
L/R Hand Sizes	10
Correlation of Hand Sizes	10
Davis Data	11
Height and Weight	11
Correlation of Height and Weight	12
Correlation Among Females	13
Correlation Among Males	13
Simple Linear Regression	13
Definition	13
Rationale	14
Setup	14
Line Minimizing Squared Error	14
Least Squares Solution	14
Visualizing Least Squares Line	15

Example: Height and Weight	15
Calculate the Line Directly	16
Plot the Line	17
Observed Data, Fits, and Residuals	17
Proportion of Variation Explained	18
lm() Function in R	18
Calculate the Line in R	18
An lm Object is a List	18
From the R Help	19
Some of the List Items	19
summary()	19
summary() List Elements	20
Using tidy()	20
Proportion of Variation Explained	20
Assumptions to Verify	20
Residual Distribution	21
Normal Residuals Check	21
Fitted Values Vs. Obs. Residuals	22
Ordinary Least Squares	23
OLS Solution	23
Sample Variance	24
Sample Covariance	24
Expected Values	24
Standard Error	25
Proportion of Variance Explained	25
Normal Errors	25
Sampling Distribution	25
CLT	26
Gauss-Markov Theorem	26
Generalized Least Squares	26
GLS Solution	27
Other Results	27
OLS in R	27
Weight Regressed on Height + Sex	28
One Variable, Two Scales	28
Interactions	29
More on Interactions	29
Visualizing Three Different Models	30
Categorical Explanatory Variables	30
Example: Chicken Weights	30
Factor Variables in lm()	31
Plot the Fit	31

ANOVA (Version 1)	32
<code>anova()</code>	32
How It Works	33
Top of Design Matrix	33
Bottom of Design Matrix	34
Model Fits	34
Variable Transformations	35
Rationale	35
Power and Log Transformations	35
<code>Diamonds</code> Data	35
Nonlinear Relationship	35
Regression with Nonlinear Relationship	36
Residual Distribution	37
Normal Residuals Check	37
Log-Transformation	38
OLS on Log-Transformed Data	39
Residual Distribution	40
Normal Residuals Check	40
Tree Pollen Study	41
Tree Pollen Count by Week	42
A Clever Transformation	42
<code>week</code> Transformed	43
Logistic Regression	43
Goal	43
Bernoulli as EFD	44
Model	44
Maximum Likelihood Estimation	44
Iteratively Reweighted Least Squares	45
GLMs	45
<code>glm()</code> Function in R	45
Example: Grad School Admissions	45
Explore the Data	46
Logistic Regression in R	48
Summary of Fit	49
ANOVA of Fit	49
Example: Contraceptive Use	50
A Different Format	51
Fitting the Model	51
Summary of Fit	51
ANOVA of Fit	52
More on this Data Set	53
Extras	53
Source	53

Session Information	53
-------------------------------	----

Statistical Models

Probabilistic Models

So far we have covered inference of parameters that quantify a population of interest.

This is called inference of probabilistic models.

Multivariate Models

Some of the probabilistic models we considered involve calculating conditional probabilities such as $\Pr(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})$ or $\Pr(\boldsymbol{\theta}|\mathbf{X})$.

It is often the case that we would like to build a model that *explains the variation of one variable in terms of other variables*. **Statistical modeling** typically refers to this goal.

Variables

Let's suppose our data comes in the form $(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \dots, (\mathbf{X}_n, Y_n) \sim F$.

We will call $\mathbf{X}_i = (X_{i1}, X_{i2}, \dots, X_{ip}) \in \mathbb{R}^{1 \times p}$ the **explanatory variables** and $Y_i \in \mathbb{R}$ the **dependent variable** or **response variable**.

We can collect all variables as matrices

$$\mathbf{Y}_{n \times 1} \text{ and } \mathbf{X}_{n \times p}$$

where each row is a unique observation.

Statistical Model

Statistical models are concerned with *how* variables are dependent. The most general model would be to infer

$$\Pr(Y|\mathbf{X}) = h(\mathbf{X})$$

where we would specifically study the form of $h(\cdot)$ to understand how Y is dependent on \mathbf{X} .

A more modest goal is to infer the transformed conditional expectation

$$g(\mathbb{E}[Y|\mathbf{X}]) = h(\mathbf{X})$$

which sometimes leads us back to an estimate of $\Pr(Y|\mathbf{X})$.

Parametric vs Nonparametric

A **parametric** model is a pre-specified form of $h(X)$ whose terms can be characterized by a formula and interpreted. This usually involves parameters on which inference can be performed, such as coefficients in a linear model.

A **nonparametric** model is a data-driven form of $h(X)$ that is often very flexible and is not easily expressed or interpreted. A nonparametric model often does not include parameters on which we can do inference.

Simple Linear Regression

For random variables $(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$, **simple linear regression** estimates the model

$$Y_i = \beta_1 + \beta_2 X_i + E_i$$

where $\mathbb{E}[E_i] = 0$, $\text{Var}(E_i) = \sigma^2$, and $\text{Cov}(E_i, E_j) = 0$ for all $1 \leq i, j \leq n$ and $i \neq j$.

Note that in this model $\mathbb{E}[Y|X] = \beta_1 + \beta_2 X$.

Ordinary Least Squares

Ordinary least squares (OLS) estimates the model

$$\begin{aligned} Y_i &= \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_p X_{ip} + E_i \\ &= \mathbf{X}_i \boldsymbol{\beta} + E_i \end{aligned}$$

where $\mathbb{E}[E_i] = 0$, $\text{Var}(E_i) = \sigma^2$, and $\text{Cov}(E_i, E_j) = 0$ for all $1 \leq i, j \leq n$ and $i \neq j$.

Note that typically $X_{i1} = 1$ for all i so that $\beta_1 X_{i1} = \beta_1$ serves as the intercept.

Generalized Least Squares

Generalized least squares (GLS) assumes the same model as OLS, except it allows for **heteroskedasticity** and **covariance** among the E_i . Specifically, it is assumed that $\mathbf{E} = (E_1, \dots, E_n)^T$ is distributed as

$$\mathbf{E}_{n \times 1} \sim (\mathbf{0}, \Sigma)$$

where $\mathbf{0}$ is the expected value $\Sigma = (\sigma_{ij})$ is the $n \times n$ symmetric covariance matrix.

Matrix Form of Linear Models

We can write the models as

$$\mathbf{Y}_{n \times 1} = \mathbf{X}_{n \times p} \boldsymbol{\beta}_{p \times 1} + \mathbf{E}_{n \times 1}$$

where simple linear regression, OLS, and GLS differ in the value of p or the distribution of the E_i . We can also write the conditional expectation and covariance as

$$\mathbb{E}[\mathbf{Y}|\mathbf{X}] = \mathbf{X}\boldsymbol{\beta}, \text{ Cov}(\mathbf{Y}|\mathbf{X}) = \Sigma.$$

Least Squares Regression

In simple linear regression, OLS, and GLS, the $\boldsymbol{\beta}$ parameters are fit by minimizing the sum of squares between \mathbf{Y} and $\mathbf{X}\boldsymbol{\beta}$.

Fitting these models by “least squares” satisfies two types of optimality:

1. Gauss-Markov Theorem
2. Maximum likelihood estimate when in addition $\mathbf{E} \sim \text{MVN}_n(\mathbf{0}, \Sigma)$

Details will follow on these.

Generalized Linear Models

The generalized linear model (GLM) builds from OLS and GLS to allow the response variable to be distributed according to an exponential family distribution. Suppose that $\eta(\theta)$ is function of the expected value into the natural parameter. The estimated model is

$$\eta(\mathbb{E}[Y|\mathbf{X}]) = \mathbf{X}\boldsymbol{\beta}$$

which is fit by maximized likelihood estimation.

Generalized Additive Models

Next week, we will finally arrive at inferring semiparametric models where $Y|\mathbf{X}$ is distributed according to an exponential family distribution. The models, which are called **generalized additive models** (GAMs), will be of the form

$$\eta(\text{E}[Y|\mathbf{X}]) = \sum_{j=1}^p \sum_{k=1}^d h_k(X_j)$$

where η is the canonical link function and the $h_k(\cdot)$ functions are very flexible.

Some Trade-offs

There are several important trade-offs encountered in statistical modeling:

- Bias vs variance
- Accuracy vs computational time
- Flexibility vs intepretability

These are not mutually exclusive phenomena.

Bias and Variance

Suppose we estimate $Y = h(\mathbf{X}) + E$ by some $\hat{Y} = \hat{h}(\mathbf{X})$. The following bias-variance trade-off exists:

$$\begin{aligned} \text{E}\left[\left(Y - \hat{Y}\right)^2\right] &= \text{E}\left[\left(h(\mathbf{X}) + E - \hat{h}(\mathbf{X})\right)^2\right] \\ &= \text{E}\left[\left(h(\mathbf{X}) - \hat{h}(\mathbf{X})\right)^2\right] + \text{Var}(E) \\ &= \left(h(\mathbf{X}) - \text{E}[\hat{h}(\mathbf{X})]\right)^2 + \text{Var}(\hat{h}(\mathbf{X}))^2 + \text{Var}(E) \\ &= \text{bias}^2 + \text{variance} + \text{Var}(E) \end{aligned}$$

Motivating Examples

Sample Correlation

Least squares regression “modelizes” correlation. Suppose we observe n pairs of data $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. Their sample correlation is

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

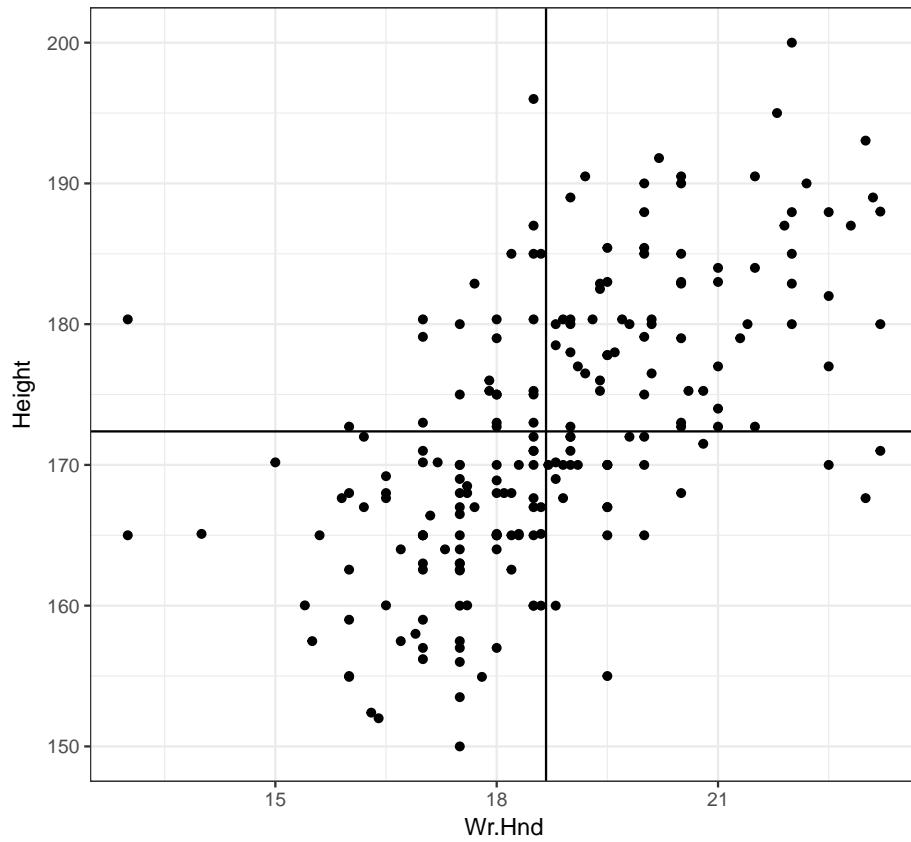
$$= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} \quad (2)$$

where s_x and s_y are the sample standard deviations of each measured variable.

Example: Hand Size Vs. Height

```
> library("MASS")
> data("survey", package="MASS")
> head(survey)
  Sex Wr.Hnd NW.Hnd W.Hnd Fold Pulse Clap Exer Smoke
1 Female  18.5   18.0 Right R on L  92 Left Some Never
2  Male   19.5   20.5 Left  R on L 104 Left None Regul
3  Male   18.0   13.3 Right L on R  87 Neither None Occas
4  Male   18.8   18.9 Right R on L NA Neither None Never
5  Male   20.0   20.0 Right Neither 35 Right Some Never
6 Female  18.0   17.7 Right L on R  64 Right Some Never
  Height      M.I     Age
1 173.00 Metric 18.250
2 177.80 Imperial 17.583
3    NA <NA> 16.917
4 160.00 Metric 20.333
5 165.00 Metric 23.667
6 172.72 Imperial 21.000

> ggplot(data = survey, mapping=aes(x=Wr.Hnd, y=Height)) +
+   geom_point() + geom_vline(xintercept=mean(survey$Wr.Hnd, na.rm=TRUE)) +
+   geom_hline(yintercept=mean(survey$Height, na.rm=TRUE))
```



Cor. of Hand Size and Height

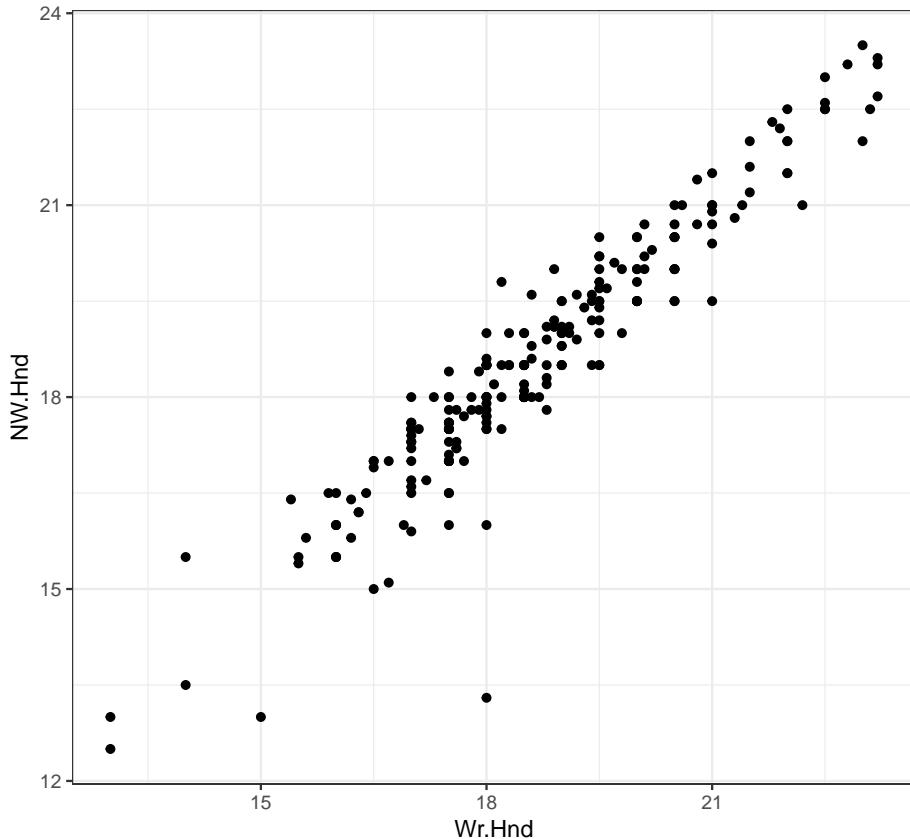
```
> cor.test(x=survey$Wr.Hnd, y=survey$Height)

Pearson's product-moment correlation

data: survey$Wr.Hnd and survey$Height
t = 10.792, df = 206, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.5063486 0.6813271
sample estimates:
cor
0.6009909
```

L/R Hand Sizes

```
> ggplot(data = survey) +  
+   geom_point(aes(x=Wr.Hnd, y=NW.Hnd))
```



Correlation of Hand Sizes

```
> cor.test(x=survey$Wr.Hnd, y=survey$NW.Hnd)  
  
Pearson's product-moment correlation  
  
data: survey$Wr.Hnd and survey$NW.Hnd  
t = 45.712, df = 234, p-value < 2.2e-16  
alternative hypothesis: true correlation is not equal to 0  
95 percent confidence interval:  
 0.9336780 0.9597816
```

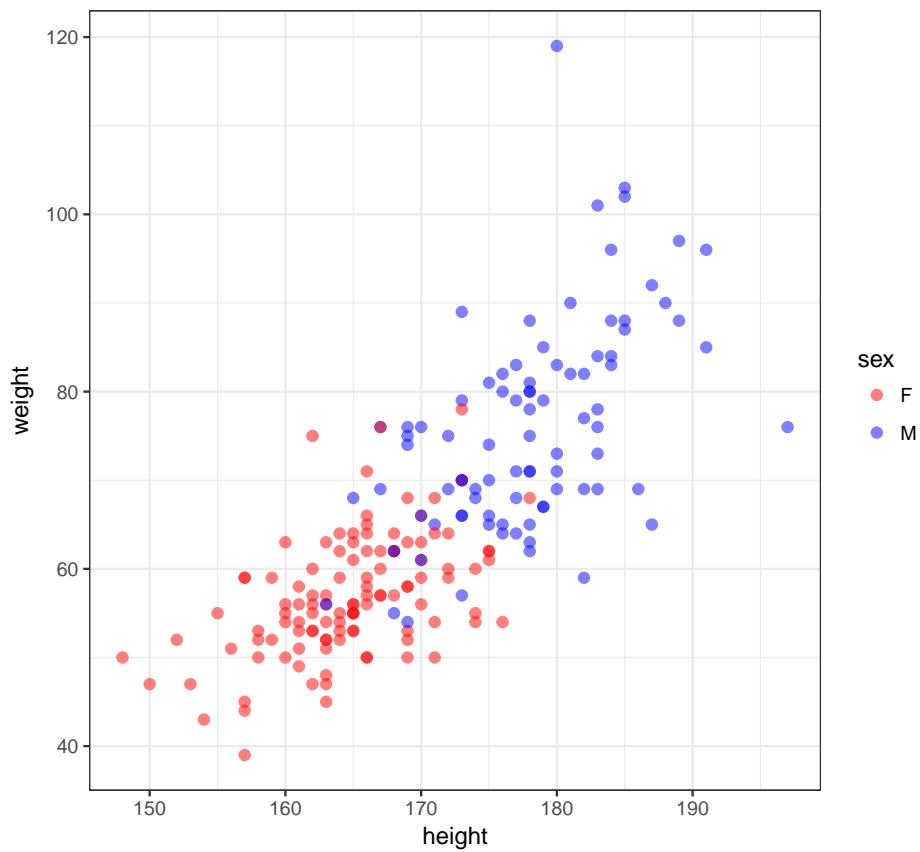
```
sample estimates:  
      cor  
0.9483103
```

Davis Data

```
> library("car")  
> data("Davis", package="car")  
  
> htwt <-tbl_df(Davis)  
> htwt[12,c(2,3)] <- htwt[12,c(3,2)]  
> head(htwt)  
# A tibble: 6 × 5  
  sex   weight  height repwt rept  
  <fctr> <int>   <int> <int> <int>  
1   M     77     182    77   180  
2   F     58     161    51   159  
3   F     53     161    54   158  
4   M     68     177    70   175  
5   F     59     157    59   155  
6   M     76     170    76   165
```

Height and Weight

```
> ggplot(htwt) +  
+   geom_point(aes(x=height, y=weight, color=sex), size=2, alpha=0.5) +  
+   scale_color_manual(values=c("red", "blue"))
```



Correlation of Height and Weight

```
> cor.test(x=htwt$height, y=htwt$weight)

Pearson's product-moment correlation

data: htwt$height and htwt$weight
t = 17.04, df = 198, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.7080838 0.8218898
sample estimates:
cor
0.7710743
```

Correlation Among Females

```
> htwt %>% filter(sex=="F") %>%
+   cor.test(~ height + weight, data = .)

Pearson's product-moment correlation

data: height and weight
t = 6.2801, df = 110, p-value = 6.922e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.3627531 0.6384268
sample estimates:
cor
0.5137293
```

Correlation Among Males

```
> htwt %>% filter(sex=="M") %>%
+   cor.test(~ height + weight, data = .)

Pearson's product-moment correlation

data: height and weight
t = 5.9388, df = 86, p-value = 5.922e-08
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.3718488 0.6727460
sample estimates:
cor
0.5392906
```

Why are the stratified correlations lower?

Simple Linear Regression

Definition

For random variables $(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$, **simple linear regression** estimates the model

$$Y_i = \beta_1 + \beta_2 X_i + E_i$$

where $E[E_i] = 0$, $\text{Var}(E_i) = \sigma^2$, and $\text{Cov}(E_i, E_j) = 0$ for all $1 \leq i, j \leq n$ and $i \neq j$.

Rationale

- **Least squares linear regression** is one of the simplest and most useful modeling systems for building a model that explains the variation of one variable in terms of other variables.
- It is simple to fit, it satisfies some optimality criteria, and it is straightforward to check assumptions on the data so that statistical inference can be performed.

Setup

- Suppose that we have observed n pairs of data $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$.
- **Least squares linear regression** models variation of the **response variable** y in terms of the **explanatory variable** x in the form of $\beta_1 + \beta_2 x$, where β_1 and β_2 are chosen to satisfy a least squares optimization.

Line Minimizing Squared Error

The least squares regression line is formed from the value of β_1 and β_2 that minimize:

$$\sum_{i=1}^n (y_i - \beta_1 - \beta_2 x_i)^2.$$

For a given set of data, there is a unique solution to this minimization as long as there are at least two unique values among x_1, x_2, \dots, x_n .

Let $\hat{\beta}_1$ and $\hat{\beta}_2$ be the values that minimize this sum of squares.

Least Squares Solution

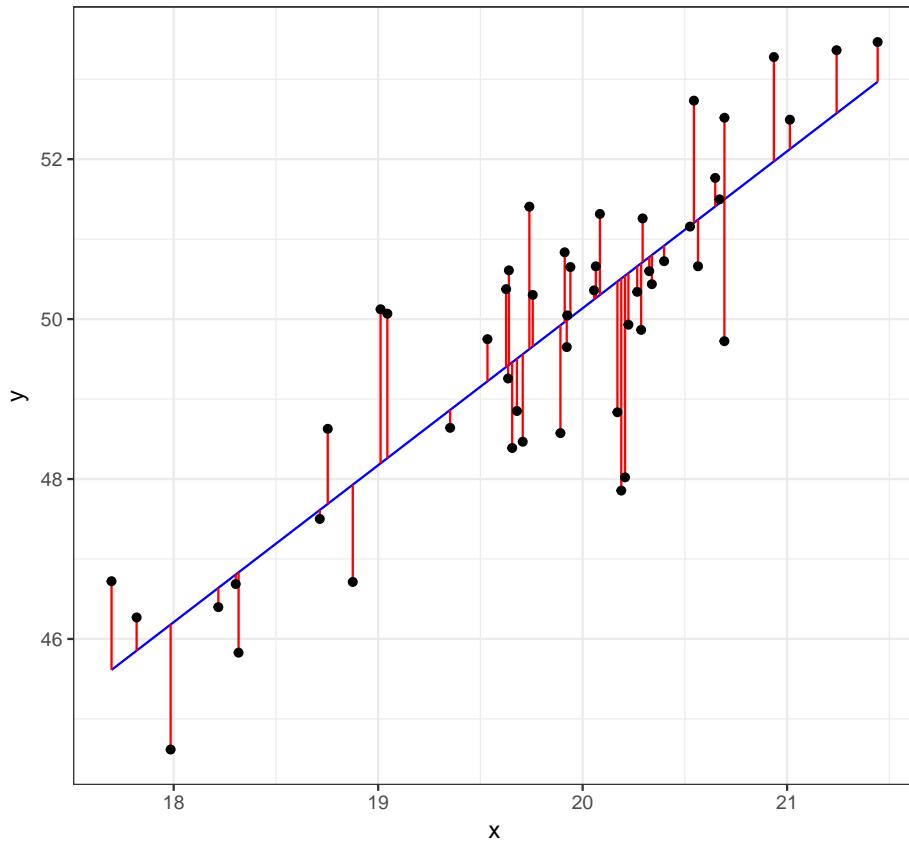
These values are:

$$\hat{\beta}_2 = r_{xy} \frac{s_y}{s_x}$$

$$\hat{\beta}_1 = \bar{y} - \hat{\beta}_2 \bar{x}$$

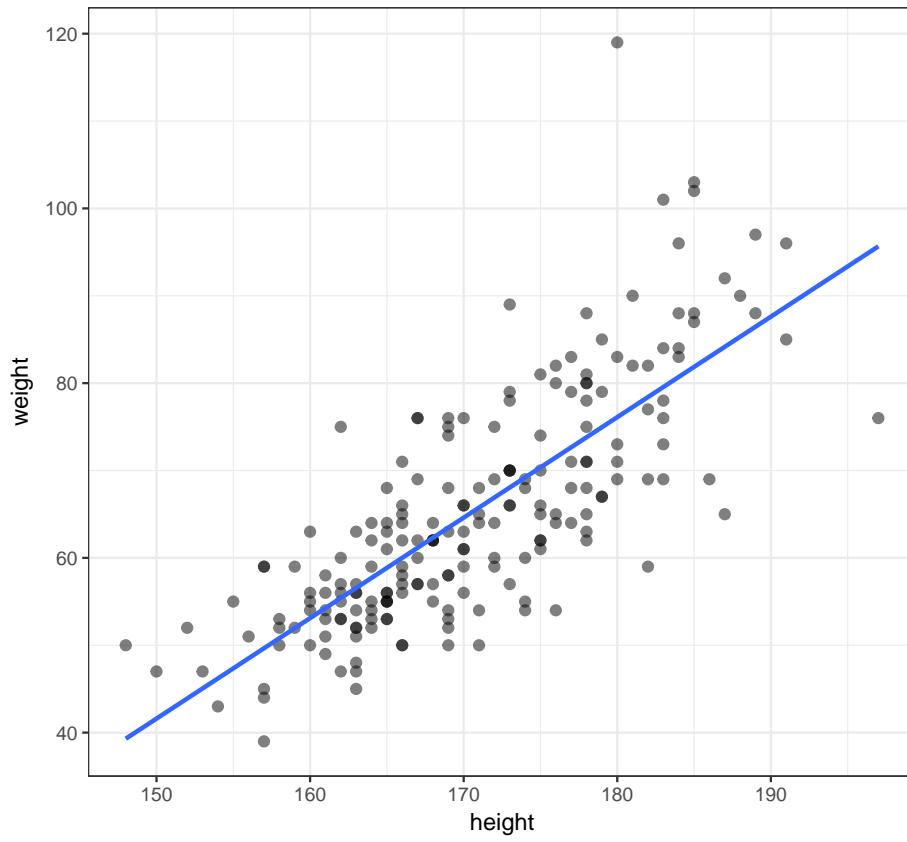
These values have a useful interpretation.

Visualizing Least Squares Line



Example: Height and Weight

```
> ggplot(data=htwt, mapping=aes(x=height, y=weight)) +  
+   geom_point(size=2, alpha=0.5) +  
+   geom_smooth(method="lm", se=FALSE, formula=y~x)
```



Calculate the Line Directly

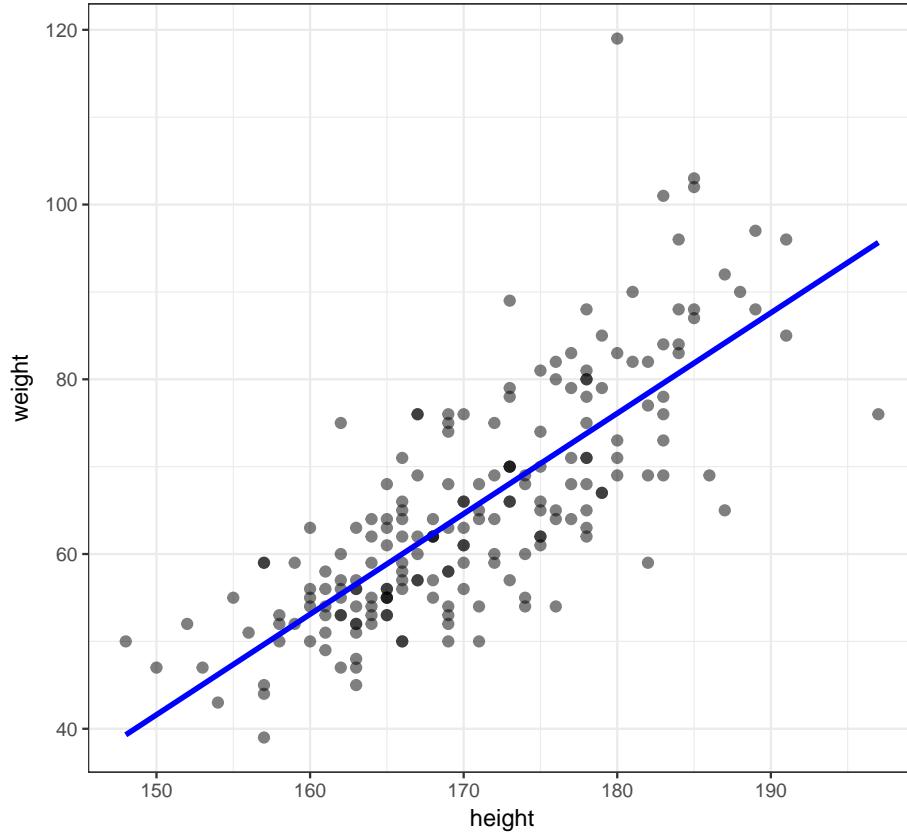
```

> beta2 <- cor(htwt$height, htwt$weight) *
+           sd(htwt$weight) / sd(htwt$height)
> beta2
[1] 1.150092
>
> beta1 <- mean(htwt$weight) - beta2 * mean(htwt$height)
> beta1
[1] -130.9104
>
> yhat <- beta1 + beta2 * htwt$height

```

Plot the Line

```
> df <- data.frame(htwt, yhat=yhat)
> ggplot(data=df) + geom_point(aes(x=height, y=weight), size=2, alpha=0.5) +
+   geom_line(aes(x=height, y=yhat), color="blue", size=1.2)
```



Observed Data, Fits, and Residuals

We observe data $(x_1, y_1), \dots, (x_n, y_n)$. Note that we only observe X_i and Y_i from the generative model $Y_i = \beta_1 + \beta_2 X_i + E_i$.

We calculate fitted values and observed residuals:

$$\hat{y}_i = \hat{\beta}_1 + \hat{\beta}_2 x_i$$

$$\hat{e}_i = y_i - \hat{y}_i$$

By construction, it is the case that $\sum_{i=1}^n \hat{e}_i = 0$.

Proportion of Variation Explained

The proportion of variance explained by the fitted model is called R^2 or r^2 . It is calculated by:

$$r^2 = \frac{s_{\hat{y}}^2}{s_y^2}$$

lm() Function in R

Calculate the Line in R

The syntax for a model in R is

```
response variable ~ explanatory variables
```

where the explanatory variables component can involve several types of terms.

```
> myfit <- lm(weight ~ height, data=htwt)
> myfit
```

Call:

```
lm(formula = weight ~ height, data = htwt)
```

Coefficients:

(Intercept)	height
-130.91	1.15

An lm Object is a List

```
> class(mymfit)
[1] "lm"
> is.list(mymfit)
[1] TRUE
> names(mymfit)
[1] "coefficients"   "residuals"      "effects"
[4] "rank"            "fitted.values"  "assign"
[7] "qr"              "df.residual"   "xlevels"
[10] "call"           "terms"         "model"
```

From the R Help

`lm` returns an object of class “lm” or for multiple responses of class c(“mlm”, “lm”).

The functions `summary` and `anova` are used to obtain and print a summary and analysis of variance table of the results. The generic accessor functions `coefficients`, `effects`, `fitted.values` and `residuals` extract various useful features of the value returned by `lm`.

Some of the List Items

These are some useful items to access from the `lm` object:

- `coefficients`: a named vector of coefficients
- `residuals`: the residuals, that is response minus fitted values.
- `fitted.values`: the fitted mean values.
- `df.residual`: the residual degrees of freedom.
- `call`: the matched call.
- `model`: if requested (the default), the model frame used.

```
summary()
```

```
> summary(myfit)

Call:
lm(formula = weight ~ height, data = htwt)

Residuals:
    Min      1Q  Median      3Q     Max 
-19.658 -5.381 -0.555  4.807 42.894 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -130.91040   11.52792  -11.36  <2e-16 ***
height       1.15009    0.06749   17.04  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.505 on 198 degrees of freedom
Multiple R-squared:  0.5946,    Adjusted R-squared:  0.5925 
F-statistic: 290.4 on 1 and 198 DF,  p-value: < 2.2e-16
```

summary() List Elements

```
> mysummary <- summary(myfit)
> names(mysummary)
[1] "call"          "terms"         "residuals"
[4] "coefficients" "aliased"       "sigma"
[7] "df"            "r.squared"     "adj.r.squared"
[10] "fstatistic"   "cov.unscaled"
```

Using tidy()

```
> library(broom)
> tidy(myfit)
#> #>   term    estimate    std.error statistic    p.value
#> #> 1 (Intercept) -130.910400 11.52792138 -11.35594 2.438012e-23
#> #> 2      height    1.150092  0.06749465  17.03975 1.121241e-40
```

Proportion of Variation Explained

The proportion of variance explained by the fitted model is called R^2 or r^2 . It is calculated by:

$$r^2 = \frac{s_{\hat{y}}^2}{s_y^2}$$

```
> summary(myfit)$r.squared
[1] 0.5945555
>
> var(myfit$fitted.values)/var(htwt$weight)
[1] 0.5945555
```

Assumptions to Verify

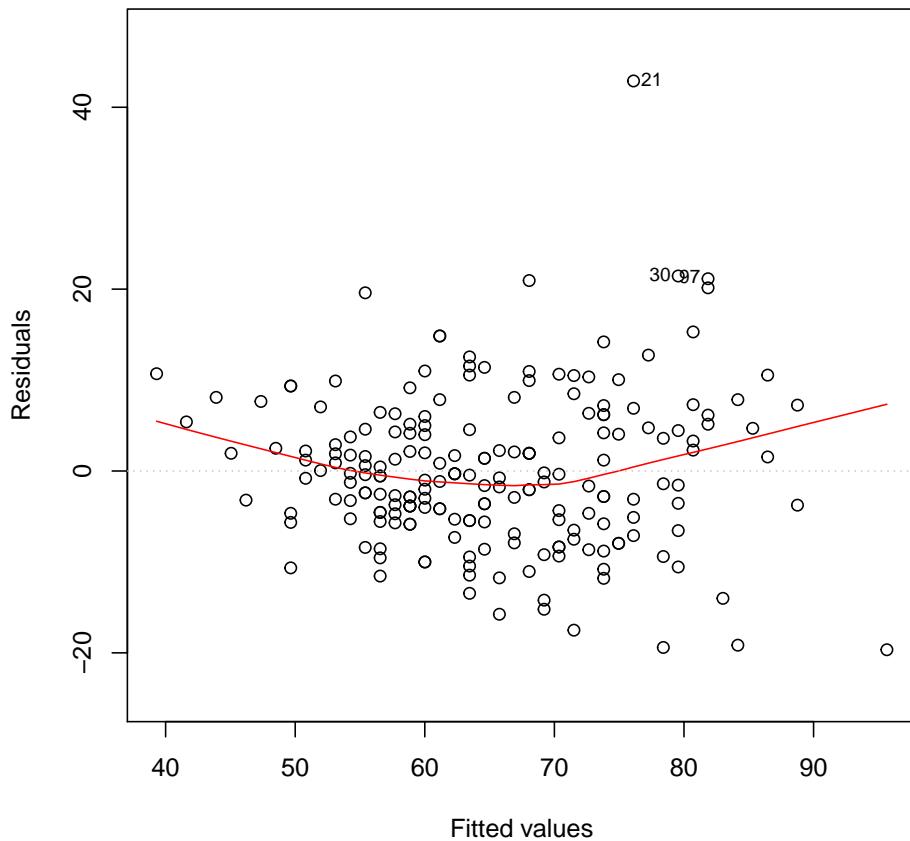
The assumptions on the above linear model are really about the joint distribution of the residuals, which are not directly observed. On data, we try to verify:

1. The fitted values and the residuals show no trends with respect to each other
2. The residuals are distributed approximately Normal($0, \sigma^2$)
 - A constant variance is called **homoscedasticity**
 - A non-constant variance is called **heteroscedasticity**
3. There are no lurking variables

There are two plots we will use in this course to investigate the first two.

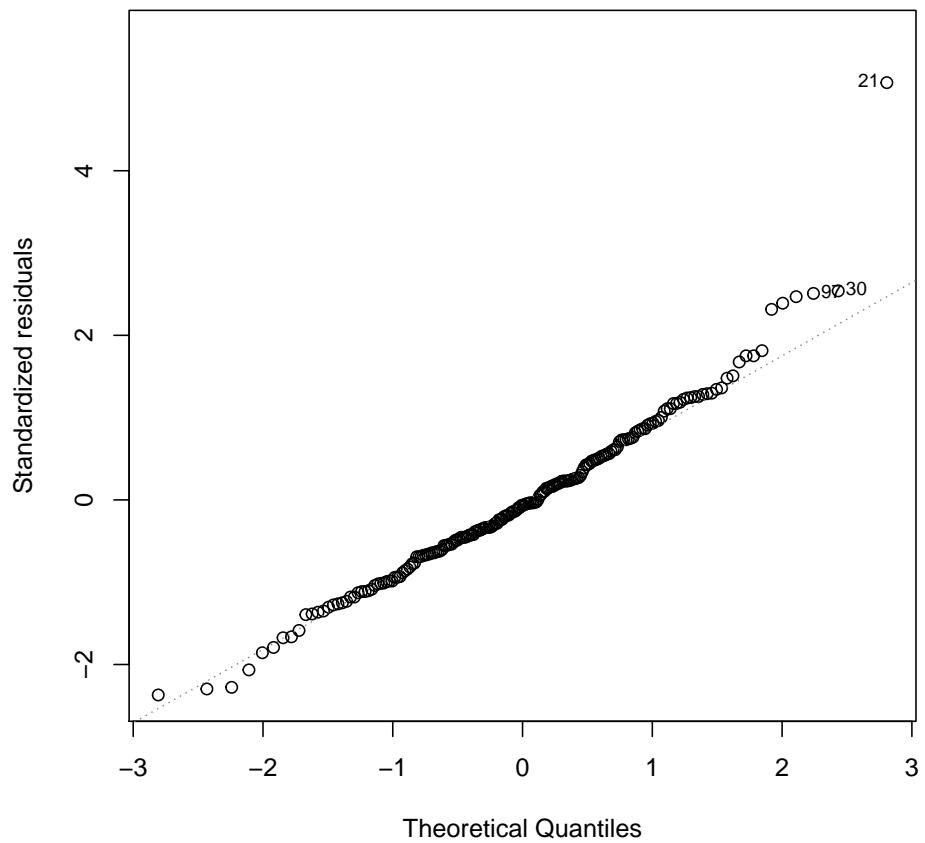
Residual Distribution

```
> plot(myfit, which=1)
```

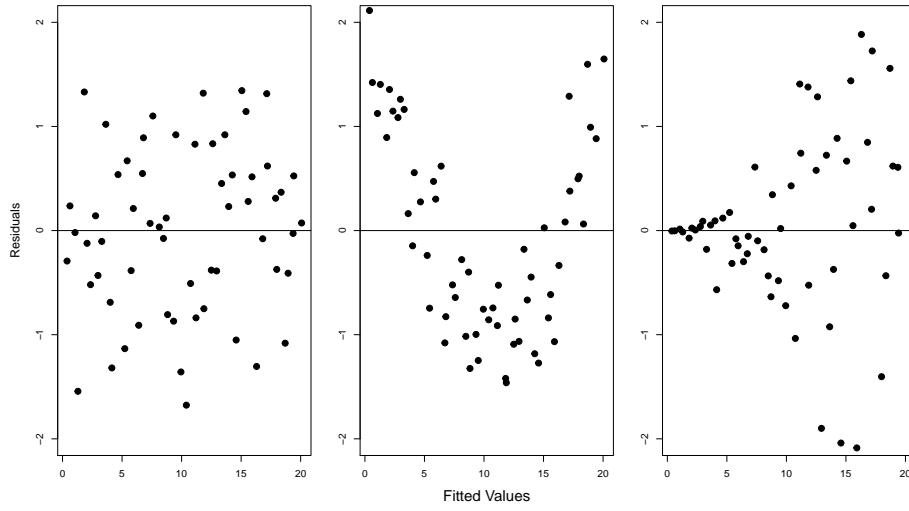


Normal Residuals Check

```
> plot(myfit, which=2)
```



Fitted Values Vs. Obs. Residuals



Ordinary Least Squares

Ordinary least squares (OLS) estimates the model

$$\begin{aligned} Y_i &= \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_p X_{ip} + E_i \\ &= \mathbf{X}_i \boldsymbol{\beta} + E_i \end{aligned}$$

where $E[E_i] = 0$, $\text{Var}(E_i) = \sigma^2$, and $\text{Cov}(E_i, E_j) = 0$ for all $1 \leq i, j \leq n$ and $i \neq j$.

Note that typically $X_{i1} = 1$ for all i so that $\beta_1 X_{i1} = \beta_1$ serves as the intercept.

OLS Solution

The estimates of $\beta_1, \beta_2, \dots, \beta_p$ are found by identifying the values that minimize:

$$\begin{aligned} \sum_{i=1}^n [Y_i - (\beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_p X_{ip})]^2 \\ = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \end{aligned}$$

The solution is expressed in terms of matrix algebra computations:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}.$$

Sample Variance

Let the predicted values of the model be

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}.$$

We estimate σ^2 by the OLS sample variance

$$S^2 = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n - p}.$$

Sample Covariance

The p -vector $\hat{\boldsymbol{\beta}}$ has covariance matrix

$$\text{Cov}(\hat{\boldsymbol{\beta}} | \mathbf{X}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2.$$

Its estimated covariance matrix is

$$\widehat{\text{Cov}}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}^T\mathbf{X})^{-1}S^2.$$

Expected Values

Under the assumption that $E[E_i] = 0$, $\text{Var}(E_i) = \sigma^2$, and $\text{Cov}(E_i, E_j) = 0$ for all $1 \leq i, j \leq n$ and $i \neq j$, we have the following:

$$E[\hat{\boldsymbol{\beta}} | \mathbf{X}] = \boldsymbol{\beta}$$

$$E[S^2 | \mathbf{X}] = \sigma^2$$

$$E[(\mathbf{X}^T\mathbf{X})^{-1}S^2 | \mathbf{X}] = \text{Cov}(\hat{\boldsymbol{\beta}})$$

$$\text{Cov}(\hat{\beta}_j, Y_i - \hat{Y}_i) = \mathbf{0}.$$

Standard Error

The standard error of $\hat{\beta}_j$ is the square root of the (j,j) diagonal entry of $(\mathbf{X}^T \mathbf{X})^{-1} \sigma^2$

$$\text{se}(\hat{\beta}_j) = \sqrt{[(\mathbf{X}^T \mathbf{X})^{-1} \sigma^2]_{jj}}$$

and estimated standard error is

$$\hat{\text{se}}(\hat{\beta}_j) = \sqrt{[(\mathbf{X}^T \mathbf{X})^{-1} S^2]_{jj}}$$

Proportion of Variance Explained

The proportion of variance explained is defined equivalently to the simple linear regression scenario:

$$R^2 = \frac{\sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}.$$

Normal Errors

Suppose we assume $E_1, E_2, \dots, E_n \stackrel{\text{iid}}{\sim} \text{Normal}(0, \sigma^2)$. Then

$$\ell(\boldsymbol{\beta}, \sigma^2; \mathbf{Y}, \mathbf{X}) \propto -n \log(\sigma^2) - \frac{1}{\sigma^2} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}).$$

Since minimizing $(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$ maximizes the likelihood with respect to $\boldsymbol{\beta}$, this implies $\hat{\boldsymbol{\beta}}$ is the MLE for $\boldsymbol{\beta}$.

It can also be calculated that $\frac{n-p}{n} S^2$ is the MLE for σ^2 .

Sampling Distribution

When $E_1, E_2, \dots, E_n \stackrel{\text{iid}}{\sim} \text{Normal}(0, \sigma^2)$, it follows that, conditional on \mathbf{X} :

$$\hat{\boldsymbol{\beta}} \sim \text{MVN}_p \left(\boldsymbol{\beta}, (\mathbf{X}^T \mathbf{X})^{-1} \sigma^2 \right)$$

$$S^2 \frac{n-p}{\sigma^2} \sim \chi_{n-p}^2$$

$$\frac{\hat{\beta}_j - \beta_j}{\hat{s.e}(\hat{\beta}_j)} \sim t_{n-p}$$

CLT

Under the assumption that $E[E_i] = 0$, $\text{Var}(E_i) = \sigma^2$, and $\text{Cov}(E_i, E_j) = 0$ for $i \neq j$, it follows that as $n \rightarrow \infty$,

$$\sqrt{n} (\hat{\beta} - \beta) \xrightarrow{D} \text{MVN}_p \left(\mathbf{0}, (\mathbf{X}^T \mathbf{X})^{-1} \sigma^2 \right).$$

Gauss-Markov Theorem

Under the assumption that $E[E_i] = 0$, $\text{Var}(E_i) = \sigma^2$, and $\text{Cov}(E_i, E_j) = 0$ for $i \neq j$, the Gauss-Markov theorem shows that among all BLUEs, **best linear unbiased estimators**, the least squares estimate has the smallest mean-squared error.

Specifically, suppose that $\tilde{\beta}$ is a linear estimator (calculated from a linear operator on \mathbf{Y}) where $E[\tilde{\beta}|\mathbf{X}] = \beta$. Then

$$E \left[(\mathbf{Y} - \mathbf{X}\hat{\beta})^T (\mathbf{Y} - \mathbf{X}\hat{\beta}) \mid \mathbf{X} \right] \leq E \left[(\mathbf{Y} - \mathbf{X}\tilde{\beta})^T (\mathbf{Y} - \mathbf{X}\tilde{\beta}) \mid \mathbf{X} \right].$$

Generalized Least Squares

Generalized least squares (GLS) assumes the same model as OLS, except it allows for **heteroskedasticity** and **covariance** among the E_i . Specifically, it is assumed that $\mathbf{E} = (E_1, \dots, E_n)^T$ is distributed as

$$\mathbf{E}_{n \times 1} \sim (\mathbf{0}, \Sigma)$$

where $\mathbf{0}$ is the expected value $\Sigma = (\sigma_{ij})$ is the $n \times n$ covariance matrix.

The most straightforward way to navigate GLS results is to recognize that

$$\Sigma^{-1/2} \mathbf{Y} = \Sigma^{-1/2} \mathbf{X} \beta + \Sigma^{-1/2} \mathbf{E}$$

satisfies the assumptions of the OLS model.

GLS Solution

The solution to minimizing

$$(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T \boldsymbol{\Sigma}^{-1} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

is

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T \boldsymbol{\Sigma}^{-1} \mathbf{X} \right)^{-1} \mathbf{X}^T \boldsymbol{\Sigma}^{-1} \mathbf{Y}.$$

Other Results

The issue of estimating $\boldsymbol{\Sigma}$ if it is unknown is complicated. Other than estimates of σ^2 , the results from the OLS section recapitulate by replacing $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{E}$ with

$$\boldsymbol{\Sigma}^{-1/2} \mathbf{Y} = \boldsymbol{\Sigma}^{-1/2} \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\Sigma}^{-1/2} \mathbf{E}.$$

For example, as $n \rightarrow \infty$,

$$\sqrt{n} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{D} \text{MVN}_p \left(\mathbf{0}, (\mathbf{X}^T \boldsymbol{\Sigma}^{-1} \mathbf{X})^{-1} \right).$$

We also still have that

$$\mathbb{E} [\hat{\boldsymbol{\beta}} | \mathbf{X}] = \boldsymbol{\beta}.$$

And when $\mathbf{E} \sim \text{MVN}_n(\mathbf{0}, \boldsymbol{\Sigma})$, $\hat{\boldsymbol{\beta}}$ is the MLE.

OLS in R

R implements OLS of multiple explanatory variables exactly the same as with a single explanatory variable, except we need to show the sum of all explanatory variables that we want to use.

```
> lm(weight ~ height + sex, data=htwt)
```

Call:

```
lm(formula = weight ~ height + sex, data = htwt)
```

```

Coefficients:
(Intercept)      height       sexM
-76.6167        0.8106      8.2269

```

Weight Regressed on Height + Sex

```

> summary(lm(weight ~ height + sex, data=htwt))

Call:
lm(formula = weight ~ height + sex, data = htwt)

Residuals:
    Min      1Q  Median      3Q      Max 
-20.131 -4.884 -0.640  5.160  41.490 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -76.6167    15.7150 -4.875 2.23e-06 ***  
height       0.8105     0.0953  8.506 4.50e-15 ***  
sexM         8.2269     1.7105  4.810 3.00e-06 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.066 on 197 degrees of freedom
Multiple R-squared:  0.6372,    Adjusted R-squared:  0.6335 
F-statistic: 173 on 2 and 197 DF,  p-value: < 2.2e-16

```

One Variable, Two Scales

We can include a single variable but on two different scales:

```

> htwt <- htwt %>% mutate(height2 = height^2)
> summary(lm(weight ~ height + height2, data=htwt))

Call:
lm(formula = weight ~ height + height2, data = htwt)

Residuals:
    Min      1Q  Median      3Q      Max 
-24.265 -5.159 -0.499  4.549  42.965 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -76.6167    15.7150 -4.875 2.23e-06 ***  
height       0.8105     0.0953  8.506 4.50e-15 ***  
height2      8.2269     1.7105  4.810 3.00e-06 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

(Intercept) 107.117140 175.246872  0.611   0.542
height      -1.632719  2.045524 -0.798   0.426
height2     0.008111  0.005959  1.361   0.175

Residual standard error: 8.486 on 197 degrees of freedom
Multiple R-squared:  0.5983,    Adjusted R-squared:  0.5943
F-statistic: 146.7 on 2 and 197 DF,  p-value: < 2.2e-16

```

Interactions

It is possible to include products of explanatory variables, which is called an *interaction*.

```

> summary(lm(weight ~ height + sex + height:sex, data=htwt))

Call:
lm(formula = weight ~ height + sex + height:sex, data = htwt)

Residuals:
    Min      1Q  Median      3Q      Max 
-20.869 -4.835 -0.897  4.429  41.122 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -45.6730    22.1342 -2.063   0.0404 *  
height       0.6227     0.1343  4.637 6.46e-06 *** 
sexM        -55.6571    32.4597 -1.715   0.0880 .    
height:sexM  0.3729     0.1892  1.971   0.0502 .    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.007 on 196 degrees of freedom
Multiple R-squared:  0.6442,    Adjusted R-squared:  0.6388 
F-statistic: 118.3 on 3 and 196 DF,  p-value: < 2.2e-16

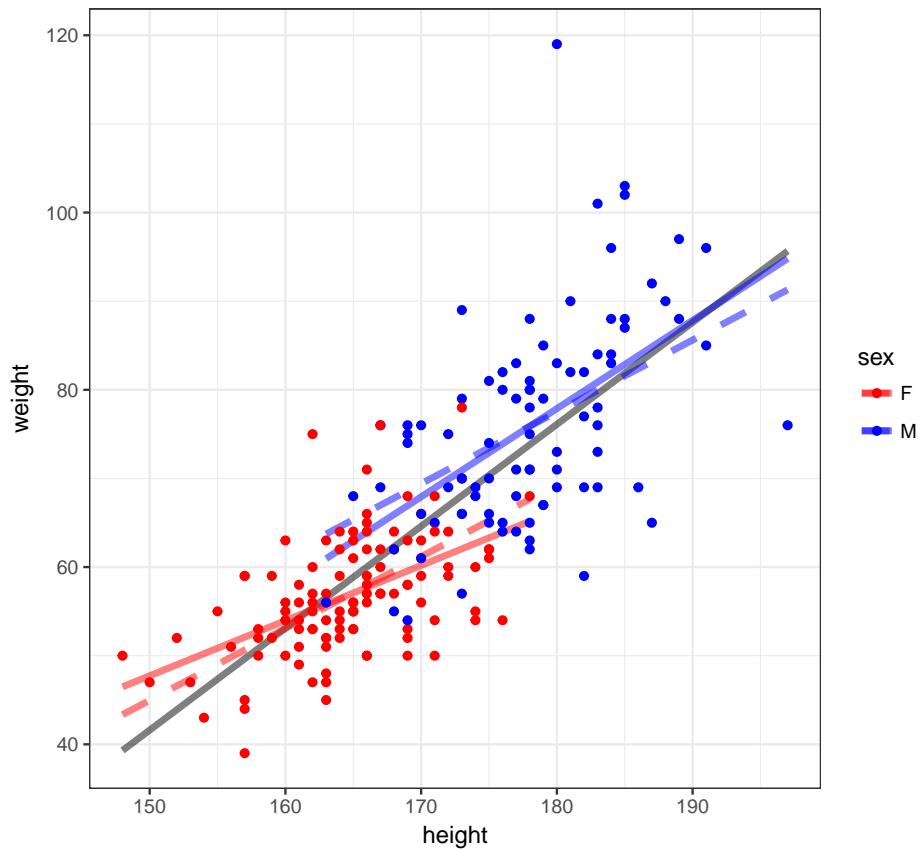
```

More on Interactions

What happens when there is an interaction between a quantitative explanatory variable and a factor explanatory variable? In the next plot, we show three models:

- Grey solid: `lm(weight ~ height, data=htwt)`
- Color dashed: `lm(weight ~ height + sex, data=htwt)`
- Color solid: `lm(weight ~ height + sex + height:sex, data=htwt)`

Visualizing Three Different Models



Categorical Explanatory Variables

Example: Chicken Weights

```
> data("chickwts", package="datasets")
> head(chickwts)
  weight      feed
1    179 horsebean
2    160 horsebean
3    136 horsebean
4    227 horsebean
5    217 horsebean
6    168 horsebean
```

```
> summary(chickwts$feed)
  casein horsebean linseed meatmeal soybean sunflower
    12       10       12      11      14       12
```

Factor Variables in lm()

```
> chick_fit <- lm(weight ~ feed, data=chickwts)
> summary(chick_fit)

Call:
lm(formula = weight ~ feed, data = chickwts)

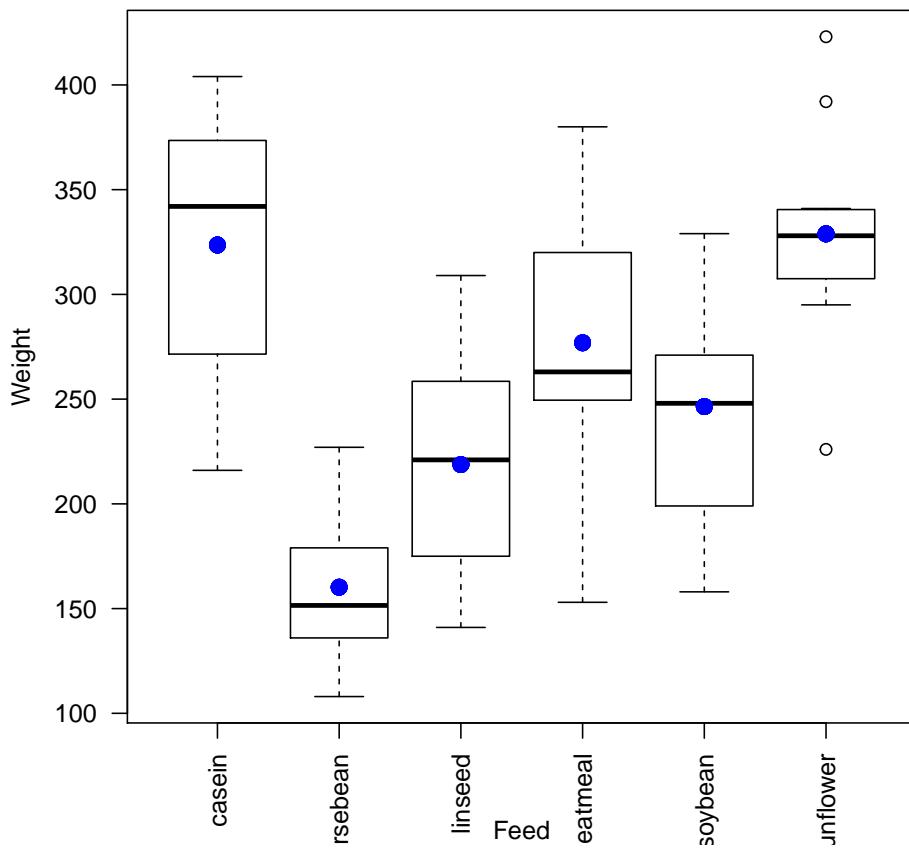
Residuals:
    Min      1Q  Median      3Q     Max 
-123.909 -34.413   1.571  38.170 103.091 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 323.583    15.834  20.436 < 2e-16 ***
feedhorsebean -163.383    23.485  -6.957 2.07e-09 ***
feedlinseed   -104.833    22.393  -4.682 1.49e-05 ***
feedmeatmeal   -46.674    22.896  -2.039 0.045567 *  
feedsoybean    -77.155    21.578  -3.576 0.000665 *** 
feedsunflower    5.333     22.393   0.238 0.812495  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 54.85 on 65 degrees of freedom
Multiple R-squared:  0.5417,    Adjusted R-squared:  0.5064 
F-statistic: 15.36 on 5 and 65 DF,  p-value: 5.936e-10
```

Plot the Fit

```
> plot(chickwts$feed, chickwts$weight, xlab="Feed", ylab="Weight", las=2)
> points(chickwts$feed, chick_fit$fitted.values, col="blue", pch=20, cex=2)
```



ANOVA (Version 1)

ANOVA (*analysis of variance*) was originally developed as a statistical model and method for comparing differences in mean values between various groups.

ANOVA quantifies and tests for differences in response variables with respect to factor variables.

In doing so, it also partitions the total variance to that due to within and between groups, where groups are defined by the factor variables.

`anova()`

The classic ANOVA table:

```
> anova(chick_fit)
Analysis of Variance Table
```

```

Response: weight
          Df Sum Sq Mean Sq F value    Pr(>F)
feed      5 231129   46226  15.365 5.936e-10 ***
Residuals 65 195556     3009
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> n <- length(chick_fit$residuals) # n <- 71
> (n-1)*var(chick_fit$fitted.values)
[1] 231129.2
> (n-1)*var(chick_fit$residuals)
[1] 195556
> (n-1)*var(chickwts$weight) # sum of above two quantities
[1] 426685.2
> (231129/5)/(195556/65) # F-statistic
[1] 15.36479

```

How It Works

```

> levels(chickwts$feed)
[1] "casein"    "horsebean"  "linseed"    "meatmeal"   "soybean"
[6] "sunflower"
> head(chickwts, n=3)
  weight      feed
1    179 horsebean
2    160 horsebean
3    136 horsebean
> tail(chickwts, n=3)
  weight      feed
69   222 casein
70   283 casein
71   332 casein
> x <- model.matrix(weight ~ feed, data=chickwts)
> dim(x)
[1] 71  6

```

Top of Design Matrix

```

> head(x)
  (Intercept) feedhorsebean feedlinseed feedmeatmeal
1            1             1             0             0
2            1             1             0             0
3            1             1             0             0

```

```

4      1      1      0      0
5      1      1      0      0
6      1      1      0      0
  feedsoybean feedsunflower
1      0      0
2      0      0
3      0      0
4      0      0
5      0      0
6      0      0

```

Bottom of Design Matrix

```

> tail(x)
#> #> (Intercept) feedhorsebean feedlinseed feedmeatmeal
#> #> 66      1          0          0          0
#> #> 67      1          0          0          0
#> #> 68      1          0          0          0
#> #> 69      1          0          0          0
#> #> 70      1          0          0          0
#> #> 71      1          0          0          0
#> #>   feedsoybean feedsunflower
#> #> 66      0          0
#> #> 67      0          0
#> #> 68      0          0
#> #> 69      0          0
#> #> 70      0          0
#> #> 71      0          0

```

Model Fits

```

> chick_fit$fitted.values %>% round(digits=4) %>% unique()
[1] 160.2000 218.7500 246.4286 328.9167 276.9091 323.5833

> chickwts %>% group_by(feed) %>% summarize(mean(weight))
#> # A tibble: 6 × 2
#>   feed `mean(weight)`
#>   <fctr>     <dbl>
#> 1 casein     323.5833
#> 2 horsebean   160.2000
#> 3 linseed    218.7500
#> 4 meatmeal    276.9091
#> 5 soybean    246.4286

```

```
6 sunflower      328.9167
```

Variable Transformations

Rationale

In order to obtain reliable model fits and inference on linear models, the model assumptions described earlier must be satisfied.

Sometimes it is necessary to *transform* the response variable and/or some of the explanatory variables.

This process should involve data visualization and exploration.

Power and Log Transformations

It is often useful to explore power and log transforms of the variables, e.g., $\log(y)$ or y^λ for some λ (and likewise $\log(x)$ or x^λ).

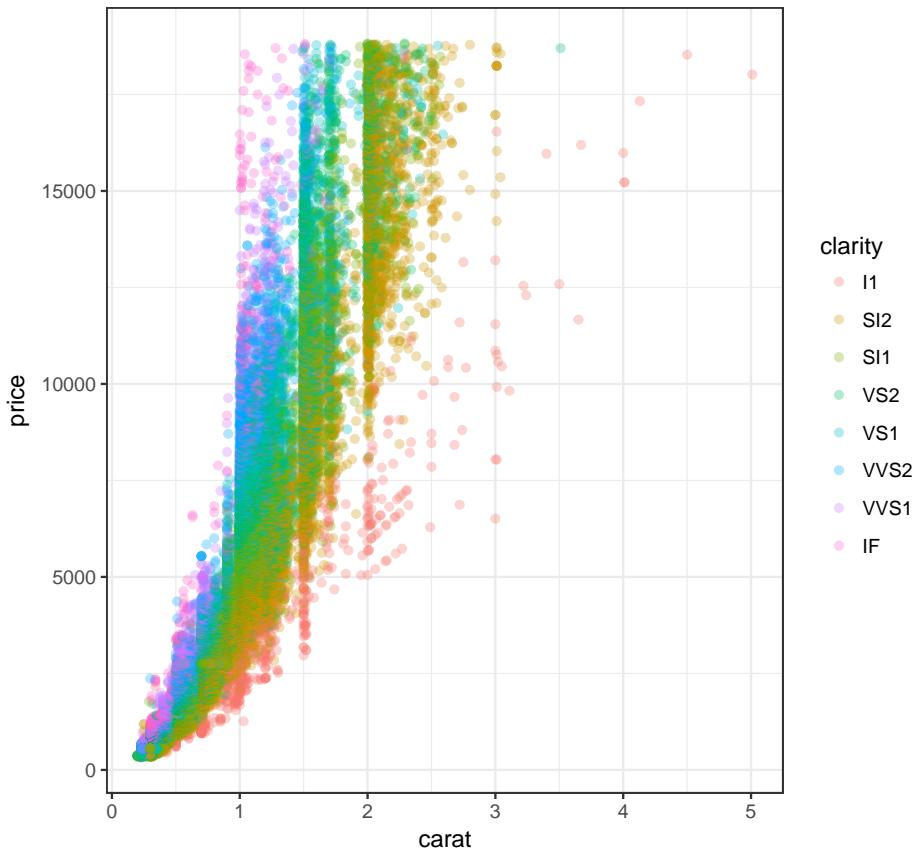
You can read more about the Box-Cox family of power transformations.

Diamonds Data

```
> data("diamonds", package="ggplot2")
> head(diamonds)
# A tibble: 6 × 10
  carat      cut color clarity depth table price     x     y
  <dbl>    <ord> <ord>   <ord> <dbl> <dbl> <int> <dbl> <dbl>
1 0.23     Ideal     E     SI2  61.5    55    326  3.95  3.98
2 0.21     Premium   E     SI1  59.8    61    326  3.89  3.84
3 0.23     Good      E     VS1  56.9    65    327  4.05  4.07
4 0.29     Premium   I     VS2  62.4    58    334  4.20  4.23
5 0.31     Good      J     SI2  63.3    58    335  4.34  4.35
6 0.24 Very Good   J     VVS2 62.8    57    336  3.94  3.96
# ... with 1 more variables: z <dbl>
```

Nonlinear Relationship

```
> ggplot(data = diamonds) +
+   geom_point(mapping=aes(x=carat, y=price, color=clarity), alpha=0.3)
```



Regression with Nonlinear Relationship

```

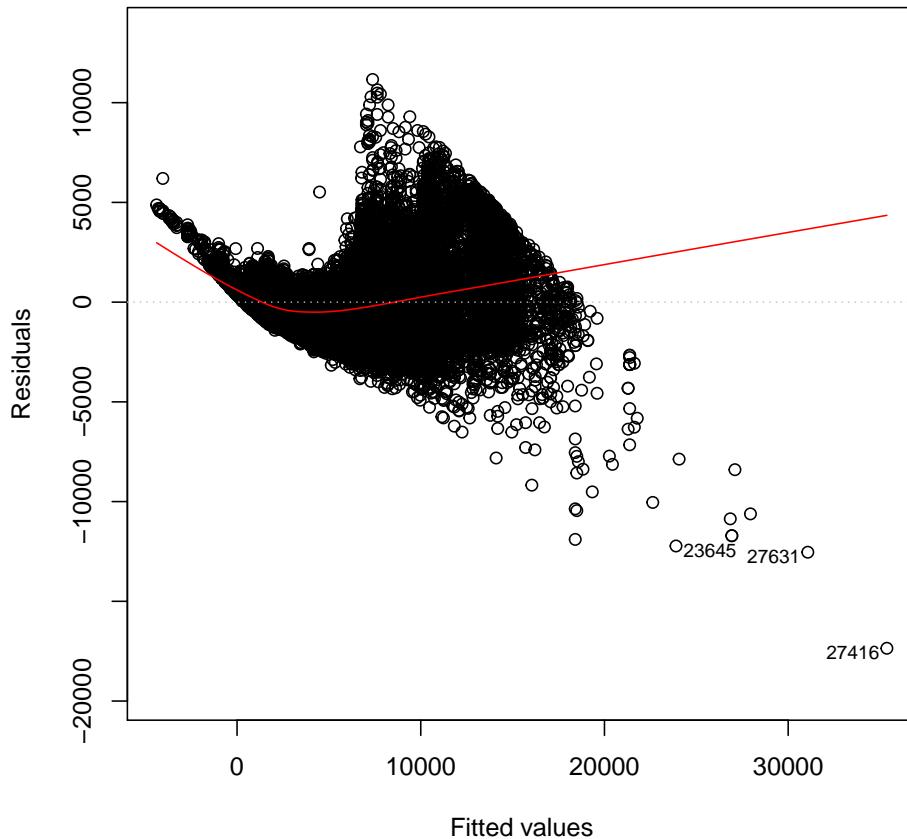
> diam_fit <- lm(price ~ carat + clarity, data=diamonds)
> anova(diam_fit)
Analysis of Variance Table

Response: price
            Df    Sum Sq   Mean Sq F value    Pr(>F)
carat        1 7.2913e+11 7.2913e+11 435639.9 < 2.2e-16 ***
clarity      7 3.9082e+10 5.5831e+09   3335.8 < 2.2e-16 ***
Residuals  53931 9.0264e+10 1.6737e+06
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

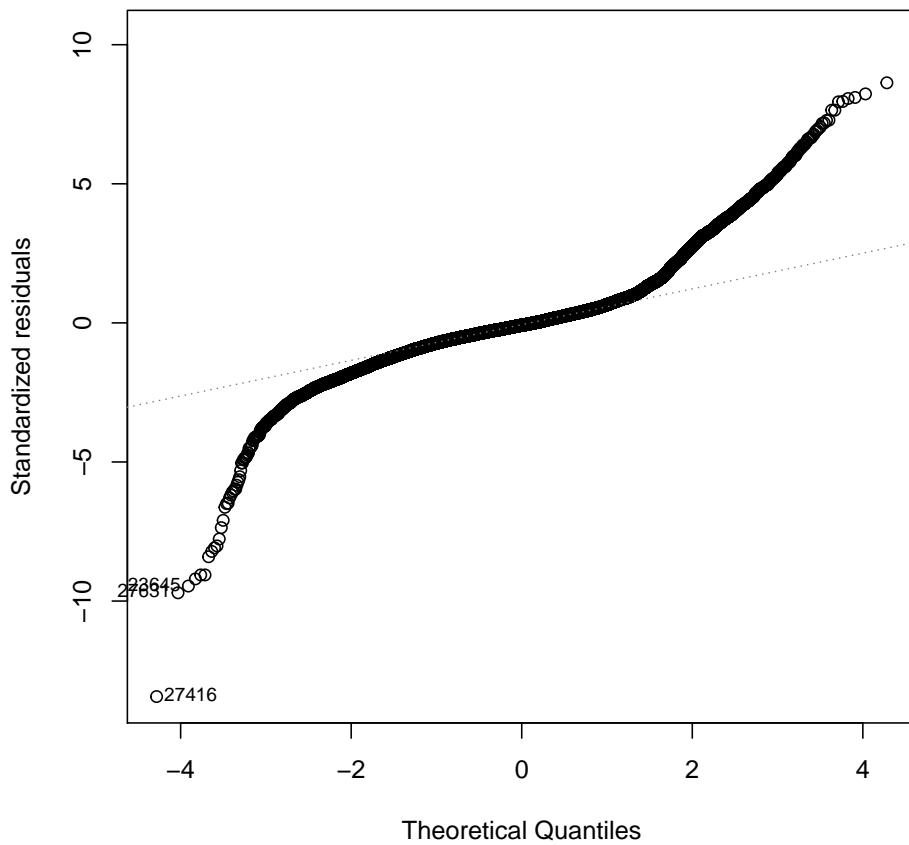
Residual Distribution

```
> plot(diam_fit, which=1)
```



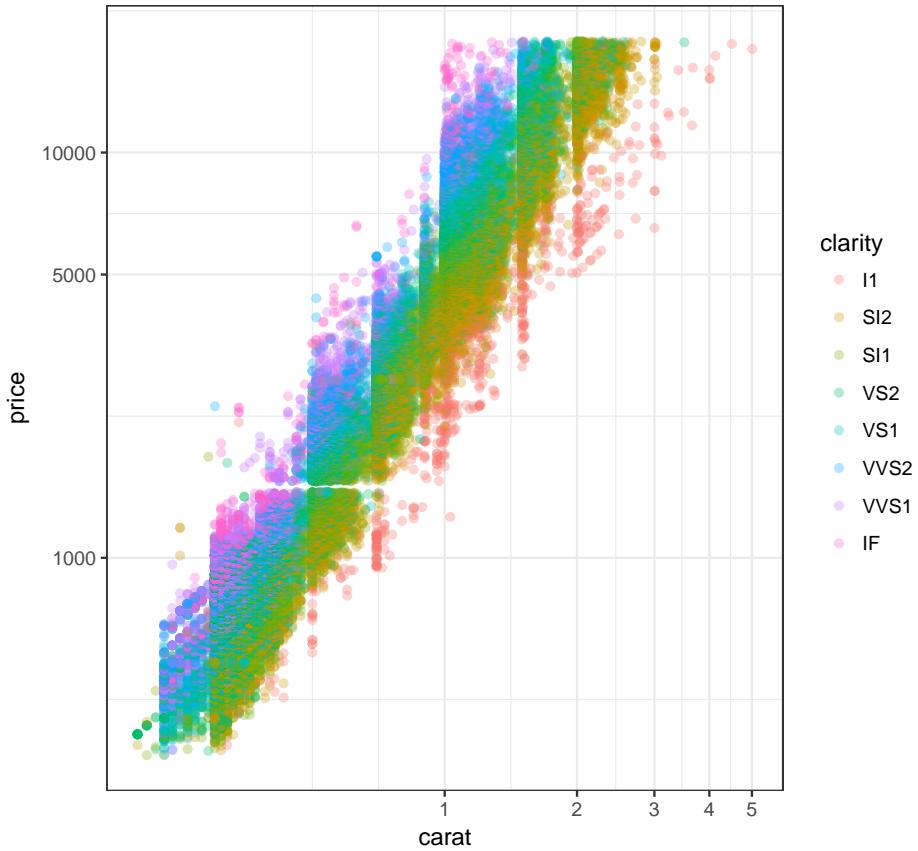
Normal Residuals Check

```
> plot(diam_fit, which=2)
```



Log-Transformation

```
> ggplot(data = diamonds) +  
+   geom_point(aes(x=carat, y=price, color=clarity), alpha=0.3) +  
+   scale_y_log10(breaks=c(1000,5000,10000)) +  
+   scale_x_log10(breaks=1:5)
```



OLS on Log-Transformed Data

```

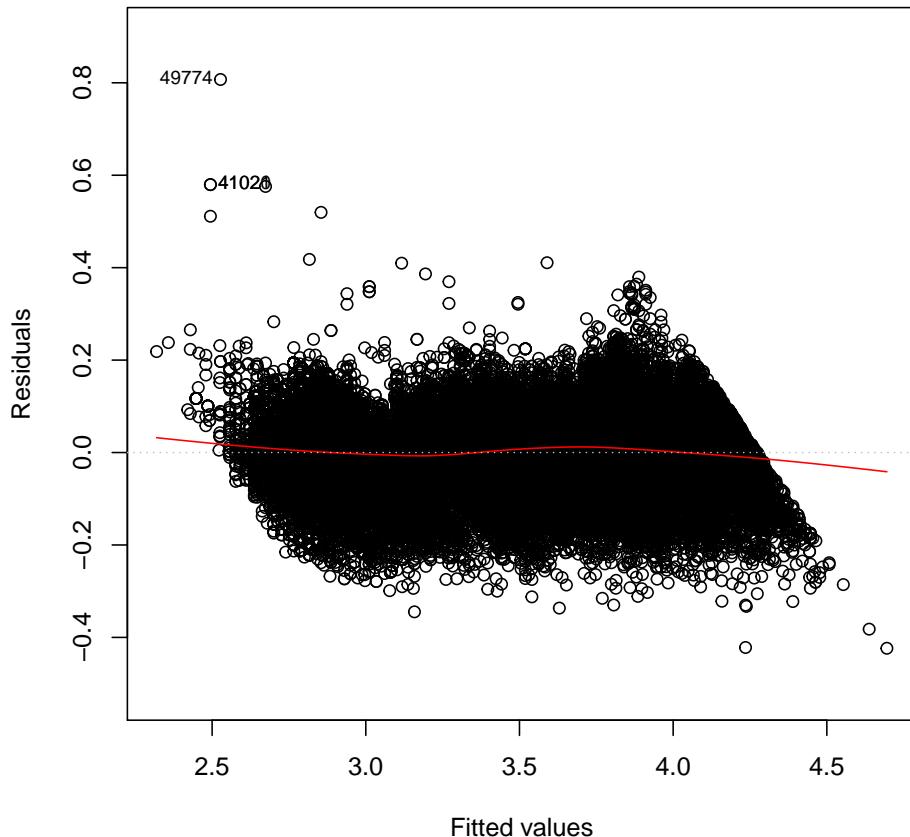
> diamonds <- mutate(diamonds, log_price = log(price, base=10),
+                      log_carat = log(carat, base=10))
> ldiam_fit <- lm(log_price ~ log_carat + clarity, data=diamonds)
> anova(ldiam_fit)
Analysis of Variance Table

Response: log_price
            Df  Sum Sq Mean Sq   F value   Pr(>F)
log_carat     1 9771.9 9771.9 1452922.6 < 2.2e-16 ***
clarity       7  339.1   48.4    7203.3 < 2.2e-16 ***
Residuals 53931  362.7     0.0
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

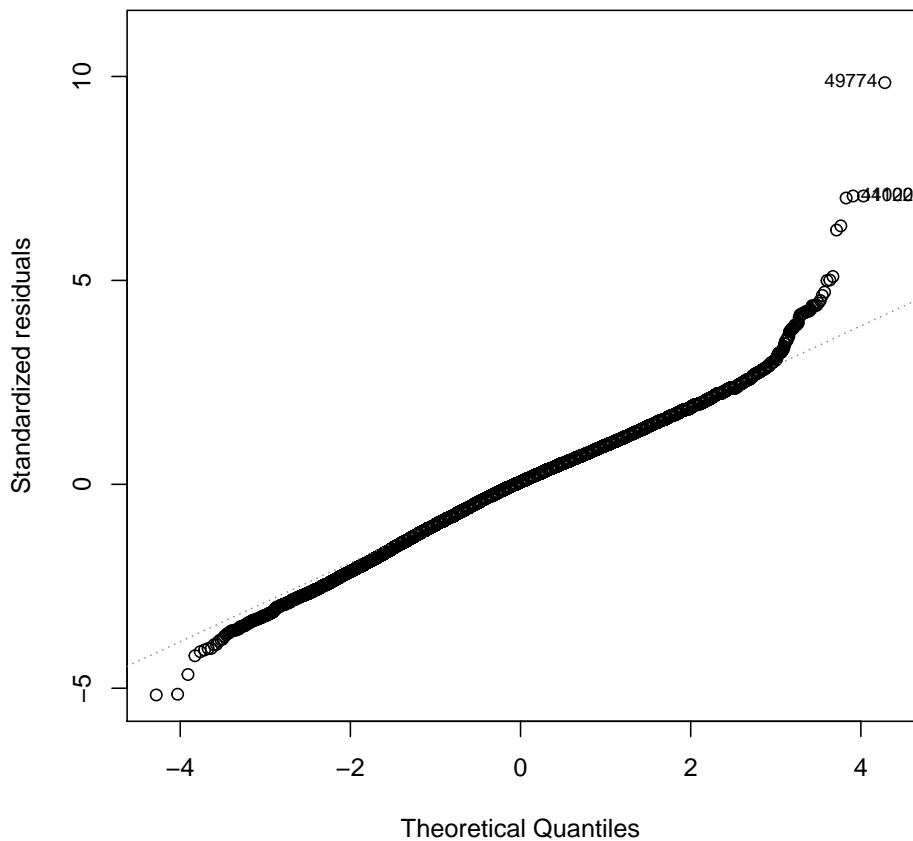
Residual Distribution

```
> plot(ldiam_fit, which=1)
```



Normal Residuals Check

```
> plot(ldiam_fit, which=2)
```



Tree Pollen Study

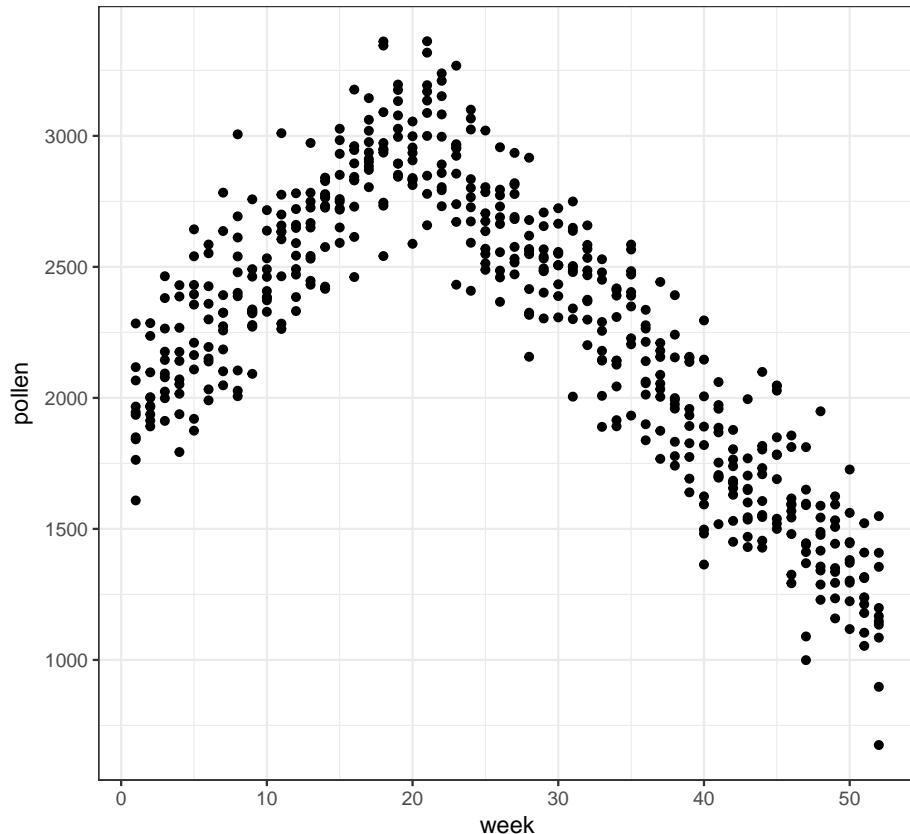
Suppose that we have a study where tree pollen measurements are averaged every week, and these data are recorded for 10 years. These data are simulated:

```
> pollen_study
# A tibble: 520 × 3
  week   year   pollen
  <int> <int>    <dbl>
1     1  2001 1841.751
2     2  2001 1965.503
3     3  2001 2380.972
4     4  2001 2141.025
5     5  2001 2210.473
6     6  2001 2585.321
7     7  2001 2392.183
8     8  2001 2104.680
9     9  2001 2278.014
10    10 2001 2383.945
```

```
# ... with 510 more rows
```

Tree Pollen Count by Week

```
> ggplot(pollen_study) + geom_point(aes(x=week, y=pollen))
```



A Clever Transformation

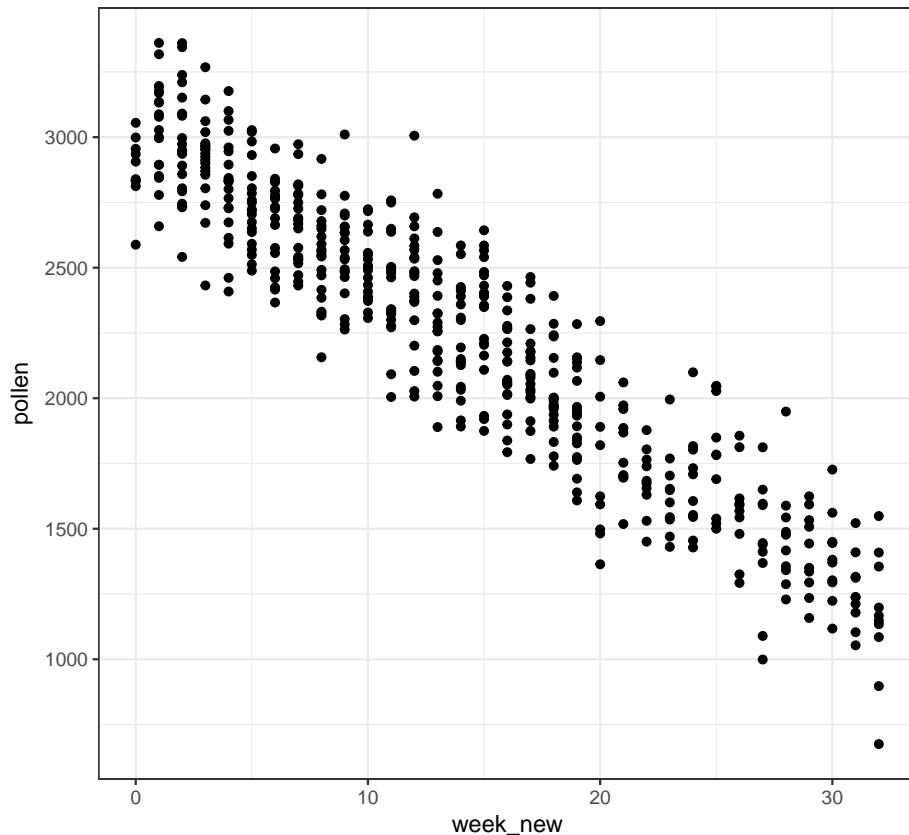
We can see there is a linear relationship between `pollen` and `week` if we transform `week` to be number of weeks from the peak week.

```
> pollen_study <- pollen_study %>%
+   mutate(week_new = abs(week-20))
```

Note that this is a very different transformation from taking a log or power transformation.

week Transformed

```
> ggplot(pollen_study) + geom_point(aes(x=week_new, y=pollen))
```



Logistic Regression

Goal

Logistic regression models a Bernoulli distributed response variable in terms of linear combinations of explanatory variables.

This extends least squares regression to the case where the response variable captures a “success” or “failure” type outcome.

Bernoulli as EFD

If $Y \sim \text{Bernoulli}(p)$, then its pmf is:

$$\begin{aligned} f(y; p) &= p^y (1-p)^{1-y} \\ &= \exp \left\{ \log \left(\frac{p}{1-p} \right) y + \log(1-p) \right\} \end{aligned}$$

In exponential family distribution (EFD) notation,

$$\eta(p) = \log \left(\frac{p}{1-p} \right) \equiv \text{logit}(p),$$

$A(\eta(p)) = \log(1 + \exp(\eta)) = \log(1 - p)$, and y is the sufficient statistic.

Model

$(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \dots, (\mathbf{X}_n, Y_n)$ are distributed so that $Y_i | \mathbf{X}_i \sim \text{Bernoulli}(p_i)$, where $\{Y_i | \mathbf{X}_i\}_{i=1}^n$ are jointly independent and

$$\text{logit}(\text{E}[Y_i | \mathbf{X}_i]) = \log \left(\frac{\Pr(Y_i = 1 | \mathbf{X}_i)}{\Pr(Y_i = 0 | \mathbf{X}_i)} \right) = \mathbf{X}_i \boldsymbol{\beta}.$$

From this it follows that

$$p_i = \frac{\exp(\mathbf{X}_i \boldsymbol{\beta})}{1 + \exp(\mathbf{X}_i \boldsymbol{\beta})}.$$

Maximum Likelihood Estimation

The $\boldsymbol{\beta}$ are estimated from the MLE calculated from:

$$\begin{aligned} \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X}) &= \sum_{i=1}^n \log \left(\frac{p_i}{1-p_i} \right) y_i + \log(1-p_i) \\ &= \sum_{i=1}^n (\mathbf{x}_i \boldsymbol{\beta}) y_i - \log(1 + \exp(\mathbf{x}_i \boldsymbol{\beta})) \end{aligned}$$

Iteratively Reweighted Least Squares

1. Initialize $\beta^{(1)}$.
2. For each iteration $t = 1, 2, \dots$, set

$$p_i^{(t)} = \text{logit}^{-1}(\mathbf{x}_i \boldsymbol{\beta}^{(t)}), \quad z_i^{(t)} = \text{logit}(p_i^{(t)}) + \frac{y_i - p_i^{(t)}}{p_i^{(t)}(1 - p_i^{(t)})}$$

and let $\mathbf{z}^{(t)} = \{z_i^{(t)}\}_{i=1}^n$.

3. Form $n \times n$ diagonal matrix $\mathbf{W}^{(t)}$ with (i, i) entry equal to $p_i^{(t)}(1 - p_i^{(t)})$.
4. Obtain $\boldsymbol{\beta}^{(t+1)}$ by performing the weighted least squares regression (see GLS from earlier)

$$\boldsymbol{\beta}^{(t+1)} = (\mathbf{X}^T \mathbf{W}^{(t)} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}^{(t)} \mathbf{z}^{(t)}.$$

5. Iterate Steps 2-4 over $t = 1, 2, 3, \dots$ until convergence, setting $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(\infty)}$.

GLMs

For exponential family distribution response variables, the **generalized linear model** is

$$\eta(\mathbb{E}[Y|\mathbf{X}]) = \mathbf{X}\boldsymbol{\beta}$$

where $\eta(\theta)$ is function of the expected value θ into the natural parameter. This is called the **canonical link function** in the GLM setting.

The **iteratively reweighted least squares** algorithm presented above for calculating (local) maximum likelihood estimates of $\boldsymbol{\beta}$ has a generalization to a large class of exponential family distribution response variables.

Next week we will study the summary statistics, sampling distributions, and inference procedures for GLMs that are analogous to those we covered above for least squares regression.

`glm()` Function in R

Example: Grad School Admissions

```

> mydata <-  

+   read.csv("http://www.ats.ucla.edu/stat/data/binary.csv")  

> dim(mydata)  

[1] 400  4  

> head(mydata)  

  admit gre gpa rank  

1     0 380 3.61    3  

2     1 660 3.67    3  

3     1 800 4.00    1  

4     1 640 3.19    4  

5     0 520 2.93    4  

6     1 760 3.00    2

```

Data and analysis courtesy of <http://www.ats.ucla.edu/stat/r/dae/logit.htm>.

Explore the Data

```

> apply(mydata, 2, mean)  

  admit      gre      gpa      rank  

  0.3175 587.7000  3.3899  2.4850  

> apply(mydata, 2, sd)  

  admit      gre      gpa      rank  

  0.4660867 115.5165364  0.3805668  0.9444602  

>  

> table(mydata$admit, mydata$rank)  

  1  2  3  4  

0 28 97 93 55  

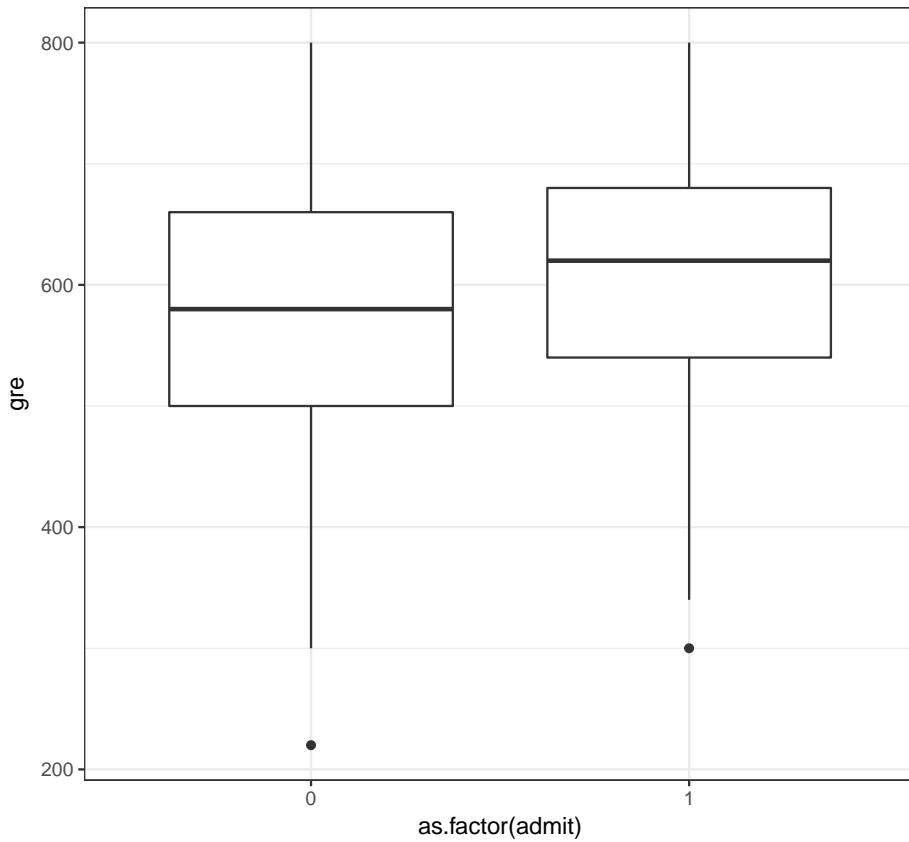
1 33 54 28 12  

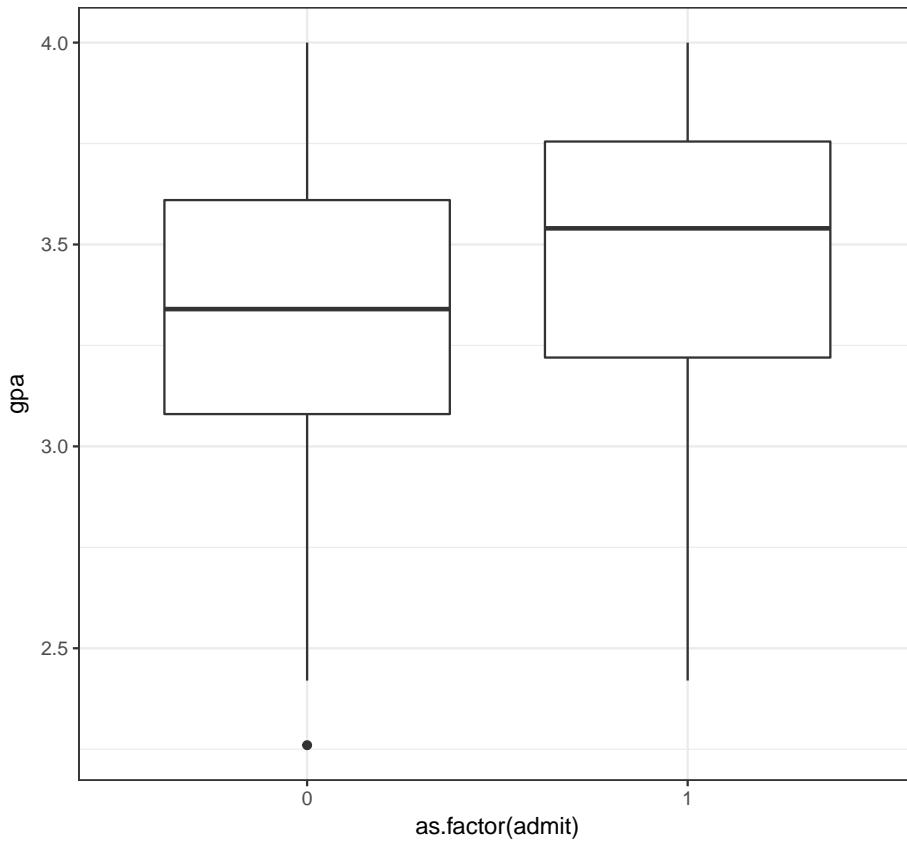
> ggplot(data=mydata) +  

+   geom_boxplot(aes(x=as.factor(admit), y=gre))

```



```
> ggplot(data=mydata) +  
+   geom_boxplot(aes(x=as.factor(admit), y=gpa))
```



Logistic Regression in R

```

> mydata$rank <- factor(mydata$rank, levels=c(1, 2, 3, 4))
> myfit <- glm(admit ~ gre + gpa + rank,
+                 data = mydata, family = "binomial")
> myfit

Call: glm(formula = admit ~ gre + gpa + rank, family = "binomial",
         data = mydata)

Coefficients:
(Intercept)          gre          gpa        rank2
-3.989979      0.002264     0.804038     -0.675443
rank3           rank4
-1.340204     -1.551464

Degrees of Freedom: 399 Total (i.e. Null); 394 Residual
  
```

```
Null Deviance: 500
Residual Deviance: 458.5      AIC: 470.5
```

Summary of Fit

```
> summary(myfit)

Call:
glm(formula = admit ~ gre + gpa + rank, family = "binomial",
     data = mydata)

Deviance Residuals:
    Min      1Q  Median      3Q      Max 
-1.6268 -0.8662 -0.6388  1.1490  2.0790 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) -3.989979  1.139951 -3.500 0.000465 ***  
gre          0.002264  0.001094  2.070 0.038465 *    
gpa          0.804038  0.331819  2.423 0.015388 *    
rank2        -0.675443  0.316490 -2.134 0.032829 *    
rank3        -1.340204  0.345306 -3.881 0.000104 ***  
rank4        -1.551464  0.417832 -3.713 0.000205 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98 on 399 degrees of freedom
    Residual deviance: 458.52 on 394 degrees of freedom
    AIC: 470.52

Number of Fisher Scoring iterations: 4
```

ANOVA of Fit

```
> anova(myfit, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: admit
```

```

Terms added sequentially (first to last)

      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL              399    499.98
gre   1  13.9204     398    486.06 0.0001907 ***
gpa   1  5.7122     397    480.34 0.0168478 *
rank  3  21.8265     394    458.52 7.088e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> anova(myfit, test="LRT")
Analysis of Deviance Table

Model: binomial, link: logit

Response: admit

Terms added sequentially (first to last)

      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL              399    499.98
gre   1  13.9204     398    486.06 0.0001907 ***
gpa   1  5.7122     397    480.34 0.0168478 *
rank  3  21.8265     394    458.52 7.088e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Example: Contraceptive Use

```

> cuse <-
+   read.table("http://data.princeton.edu/wws509/datasets/cuse.dat",
+             header=TRUE)
> dim(cuse)
[1] 16  5
> head(cuse)
  age education wantsMore notUsing using
1  <25       low      yes      53      6
2  <25       low       no      10      4
3  <25      high      yes     212     52
4  <25      high       no      50     10
5 25-29      low      yes      60     14
6 25-29      low       no      19     10

```

Data and analysis courtesy of <http://data.princeton.edu/R/glms.html>.

A Different Format

Note that in this data set there are multiple observations per explanatory variable configuration.

The last two columns of the data frame count the successes and failures per configuration.

```
> head(cuse)
  age education wantsMore notUsing using
1 <25      low       yes      53      6
2 <25      low       no       10      4
3 <25     high       yes     212     52
4 <25     high       no       50     10
5 25-29    low       yes      60     14
6 25-29    low       no       19     10
```

Fitting the Model

When this is the case, we call the `glm()` function slightly differently.

```
> myfit <- glm(cbind(using, notUsing) ~ age + education + wantsMore,
+                 data=cuse, family = binomial)
> myfit

Call: glm(formula = cbind(using, notUsing) ~ age + education + wantsMore,
  family = binomial, data = cuse)

Coefficients:
(Intercept)      age25-29      age30-39      age40-49
-0.8082        0.3894        0.9086        1.1892
educationlow  wantsMoreyes
-0.3250        -0.8330

Degrees of Freedom: 15 Total (i.e. Null); 10 Residual
Null Deviance: 165.8
Residual Deviance: 29.92    AIC: 113.4
```

Summary of Fit

```
> summary(mymfit)

Call:
glm(formula = cbind(using, notUsing) ~ age + education + wantsMore,
  family = binomial, data = cuse)
```

```

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-2.5148 -0.9376  0.2408  0.9822  1.7333 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) -0.8082    0.1590 -5.083 3.71e-07 *** 
age25-29     0.3894    0.1759  2.214  0.02681 *  
age30-39     0.9086    0.1646  5.519 3.40e-08 *** 
age40-49     1.1892    0.2144  5.546 2.92e-08 *** 
educationlow -0.3250    0.1240 -2.620  0.00879 ** 
wantsMoreyes -0.8330    0.1175 -7.091 1.33e-12 *** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 165.772  on 15  degrees of freedom
Residual deviance: 29.917  on 10  degrees of freedom
AIC: 113.43

Number of Fisher Scoring iterations: 4

```

ANOVA of Fit

```

> anova(myfit)
Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(using, notUsing)

Terms added sequentially (first to last)

          Df Deviance Resid. Df Resid. Dev
NULL           15   165.772
age            3    79.192    12    86.581
education      1    6.162    11    80.418
wantsMore      1    50.501    10    29.917

```

More on this Data Set

See <http://data.princeton.edu/R/glms.html> for more on fitting logistic regression to this data set.

A number of interesting choices are made that reveal more about the data.

Extras

Source

License

Source Code

Session Information

```
> sessionInfo()
R version 3.3.2 (2016-10-31)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: macOS Sierra 10.12.3

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats      graphics   grDevices utils      datasets   methods
[7] base

other attached packages:
[1] broom_0.4.1     car_2.1-4       MASS_7.3-45
[4] dplyr_0.5.0     purrr_0.2.2     readr_1.0.0
[7] tidyr_0.6.1     tibble_1.2      ggplot2_2.2.1
[10] tidyverse_1.1.1 knitr_1.15.1   magrittr_1.5
[13] devtools_1.12.0

loaded via a namespace (and not attached):
[1] reshape2_1.4.2    splines_3.3.2    haven_1.0.0
[4] lattice_0.20-34  colorspace_1.3-2  htmltools_0.3.5
[7] yaml_2.1.14      mgcv_1.8-16     nloptr_1.0.4
[10] foreign_0.8-67   withr_1.0.2     DBI_0.5-1
[13] modelr_0.1.0    readxl_0.1.1    plyr_1.8.4
[16] stringr_1.1.0   MatrixModels_0.4-1  munsell_0.4.3
[19] gtable_0.2.0     rvest_0.3.2     psych_1.6.12
```

```
[22] memoise_1.0.0      evaluate_0.10    labeling_0.3
[25]forcats_0.2.0      SparseM_1.76     quantreg_5.29
[28]pbkrtest_0.4-7     parallel_3.3.2   Rcpp_0.12.9
[31]scales_0.4.1       backports_1.0.5  jsonlite_1.2
[34]lme4_1.1-12        mnormt_1.5-5    hms_0.3
[37]digest_0.6.12      stringi_1.1.2   grid_3.3.2
[40]rprojroot_1.2       tools_3.3.2     lazyeval_0.2.0
[43]Matrix_1.2-8        xml2_1.1.1     lubridate_1.6.0
[46]assertthat_0.1      minqa_1.2.4    rmarkdown_1.3
[49]httr_1.2.1          R6_2.2.0      nnet_7.3-12
[52]nlme_3.1-130
```