Contents

Simple linear regression and correlation **3**
Glossary: Terms and Symbols . . . 3
Simple linear regression . . . 4
Correlations . . . 21
Bivariate Normal Distribution . . . 29

Multiple linear regression **33**
Multiple regression overview . . . 33
Assumptions, diagnostics, remedies . . . 45
Transformations . . . 57
Multiplicative models . . . 64
Multicollinearity: definition and effects . . . 72
Sums of squares and F tests . . . 82

Dummies and interactions **90**
Dummy Variables . . . 90
Interaction variables . . . 95

Feature engineering **106**
Creating feature variables . . . 106

Variable selection and shrinkage **115**
Out-of-sample evaluation measures . . . 115
Automated Model Selection . . . 123
Regularization and ridge/lasso regression . . . 128
Lasso regression . . . 133

Principal components analysis and regression **136**
Introduction to dimensionality reduction . . . 136
Linear algebra review . . . 138
PCA: key results . . . 143
Eigendecompositions and the SVD . . . 145
PCA: examples . . . 147
Your Turn: PCA . . . 153

Logistic regression **156**
Logistic regression 156
Maximum likelihood estimation 165
Evaluating classification models 174

Glossary: Regression Terms and Symbols

I follow these conventions throughout the course:[1]

Term	**ACT**	**JWHT**	**Other**
Error	ϵ	ϵ	e
Residual or estimated error	e	e	$\hat{e}$
Intercept parameter	β_0	β_0	α
Intercept estimate	$\hat{\beta}_0$	$\hat{\beta}_0$	b_0 or a
Slope parameter	β_1	β_1	β
Slope estimate	$\hat{\beta}_1$	$\hat{\beta}_1$	b_1 or b
Sum of squared errors = Residual sum of squares	SSE	RSS	
Total sum of squares	SST	TSS	
Regression sum of squares	SSR		
Error variance (parameter) =Variance of the errors	σ^2	σ^2	σ_ϵ^2
Mean squared error	MSE		S_ϵ^2
Standard deviation of the errors	σ	σ	σ_ϵ
Residual standard error =	s	RSE	S_ϵ, $\hat{\sigma}$
Root mean squared error = Standard error of the estimate			RMSE

[1] I am bridging Siegel and JWHT, which use slightly different notation. Estimates in blue, parameters in black.

Introduction to Regression

- *Objective:* to quantify the relationship between an interval-level *response* variable and one or more *predictor* variables from a sample of data.
- **Response variable** (also called **dependent**, **criterion**, or **output** (Y) variable): a variable that we wish to study and should be causally dependent on other variables
- Note: we also study categorical dependent variables, but call it the **classification problem**
- **Predictor variables** (also called **independent** (X) variables, **covariates**, or **inputs**): variables that are related to the response variable
  - **Factors** refer to categorical variables
    - **Binary variable** (aka **dichotomous**): takes two values, generically "success/failure" or "yes/no"
    - **Nominal variable**: no ordering assumed, e.g., race
    - **Ordinal variable**: values can be ordered, i.e., there exists a less than ($<$) operator, e.g., Likert scales
  - Numerical variables
    - **Count**: takes values $0, 1, 2, \ldots$
    - **Amount**: takes non-negative, real values

Other terms

- **Descriptive research**: describe some (sub)population, e.g., with univariate/bivariate descriptive statistics/graphs
- *Why build a predictive model?*
  - **Prediction**: estimate response variable accurately (typically on existing data)
  - **Exploration**: discover which x's are *associated with* y (**insights** and future hypotheses)
  - **Causal inference** (**prescription**, **confirmatory**): how predictors *cause* y (**intervention**)
- **Randomized-controlled experiment**: investigator assigns values of at least some independent variables (**treatments**)
- **Observational study**: independent variables not under the control of data scientist
- **Cross-sectional data**: measures on many sampling units at a fixed point in time
- **Time-series data**: measures recorded at equal-spaced points in time usually on a single sampling unit
- **Panel data**: measures on the same sampling units over time
- **Censored data**: values of outcome (e.g., time of failure, churn, death) only partially known

Simple Linear Regression

Assume that

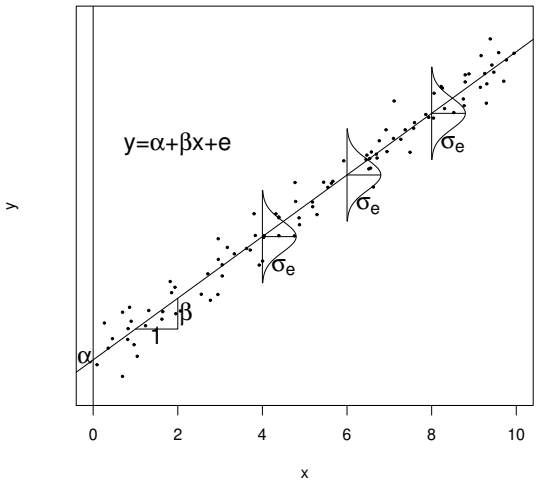
$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i,$$

where **error** $e_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$, and ϵ_i independent of ϵ_j ($\epsilon_i \perp\!\!\!\perp \epsilon_j$) for $i \neq j$ (and $\epsilon_i \perp\!\!\!\perp x_i$ if x random). This model implies that

$$\mathbb{E}(Y|x) = \mu_{Y|x} = \beta_0 + x\beta_1.$$

This is called the **regression of y on x**.

- β_0: the **intercept**. On average $Y = \beta_0$ when $x = 0$.
- β_1: the **slope**. Every unit increase in x is associated with an increase in Y of β_1, *on the average*
- σ_ϵ: **standard deviation of the errors** (σ_ϵ^2 called **error variance**). If errors are normal, empirical rule tells us for any x, 68% of points will fall within one σ_ϵ of the mean ($\beta_0 + \beta_1 x$).



Estimating the Regression Model

- In practice, we don't know values of **parameters** β_0, β_1, and σ_ϵ^2 and must estimate them with $\hat{\beta}_0$, $\hat{\beta}_1$, and S_ϵ, respectively
- The estimate of $\mu_{Y_i|x_i}$ is denoted by **fitted**, **predicted** or ***y*-hat value** $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$
- The **residual** for observation i is

$$\hat{\epsilon}_i = y_i - \hat{y}_i = y_i - a - bx_i$$

- We choose $\hat{\beta}_0$ and $\hat{\beta}_1$ so that they minimize the **least-squares criterion** (SSE means **sum of squared errors**):

$$\text{SSE} = \sum_{i=1}^{n} \hat{\epsilon}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2,$$

- The **ordinary least-squares** (OLS) estimates are

$$\hat{\beta}_1 = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sum (x_i - \overline{x})^2} = r\frac{S_y}{S_x} \quad \text{and} \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

  Note, for every SD change in x, there is a change of r SD's in y, on average, where r is the Pearson correlation.

- Estimate σ_ϵ^2 with the *mean squared error*

$$S_\epsilon^2 = \frac{\text{SSE}}{n-2} = S_y^2(1 - r^2)\frac{n-1}{n-2}$$

  σ_ϵ called **residual standard error**

Sampling Distribution of Parameters

- Theorem: **standard errors** are given by

$$S_{\hat{\beta}_1} = \frac{S_\epsilon}{S_x\sqrt{n-1}} \quad \text{and} \quad S_{\hat{\beta}_0} = S_\epsilon\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{S_x^2(n-1)}}$$

- Theorem:
  - **Unbiasedness**: $\mathbb{E}(\hat{\beta}_1) = \beta_1$ and $\mathbb{E}(\hat{\beta}_0) = \beta_0$
  - $\hat{\beta}_1$ and $\hat{\beta}_0$ have normal **sampling distributions**. The following have t distributions with $n-2$ degrees of freedom:

$$\frac{\hat{\beta}_1 - \beta_1}{S_{\hat{\beta}_1}} \quad \text{and} \quad \frac{\hat{\beta}_0 - \beta_0}{S_{\hat{\beta}_0}}$$

- **Total sum of squares**: $\text{SST} = \sum(y_i - \bar{y})^2 = (n-1)S_y^2$
- Definition: The **coefficient of determination** (aka R^2) is the square of the correlation and gives the percentage of the variability of y that is "explained" by x

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

- **Gauss-Markov Theorem**: $\hat{\beta}_1$ has a variance that is "smaller" than that of any other linear, unbiased estimator and is called the *best linear unbiased estimator* ("BLUE").
- Optional videos deriving OLS theory for simple linear regression Part 1 and Part 2

Click Ball Point Pens Example

y_i Sales in territory $i = 1, \ldots, 40$

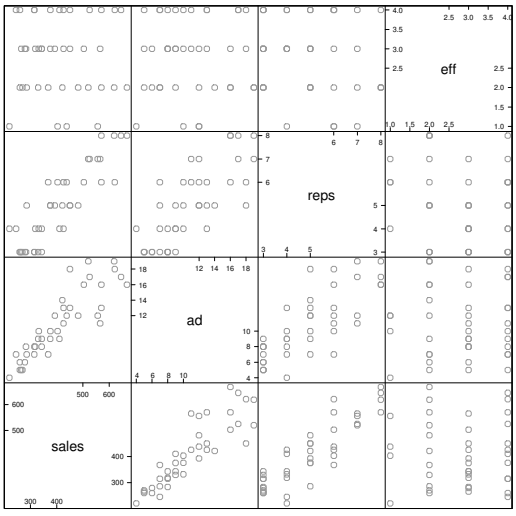
x_{i1} Advertising (number of TV spots) in territory i

x_{i2} Number of sales reps in territory i

x_{i3} Wholesaler efficiency index in territory i (4=outstanding, 3=good, 2=average, 1=poor)

```
click = data.frame(sales=c(260.3,286.1,279.4,410.8,438.2,315.3,565.1,570.0,426.1,315.0,
    403.6,220.5,343.6,644.6,520.4,329.5,426.0,343.2,450.4,421.8,245.6,503.3,375.7,265.5,
    620.6,450.5,270.1,368.0,556.1,570.0,318.5,260.2,667.0,618.3,525.3,332.2,393.2,283.5,
    376.2,481.8), ad=c(5,7,6,9,12,8,11,16,13,7,10,4,9,17,19,9,11,8,13,14,7,16,9,5,18,18,
    5,7,12,13,8,6,16,19,17,10,12,8,10,12), reps=c(3,5,3,4,6,3,7,8,4,3,6,4,4,8,7,3,6,3,5,
    5,4,6,5,3,6,5,3,6,7,6,4,3,8,8,7,4,5,3,5,5), eff=c(4,2,3,4,1,4,3,2,3,4,1,1,3,4,2,2,4,
    3,4,2,4,3,3,3,4,3,2,2,1,4,3,2,2,2,4,3,3,3,4,2))
```

```
> round(cor(click), 4)
       sales     ad    reps     eff
sales 1.0000 0.8802  0.8818  0.0019
ad    0.8802 1.0000  0.7763  0.0321
reps  0.8818 0.7763  1.0000 -0.1896
eff   0.0019 0.0321 -0.1896  1.0000

> plot(click)
```



This is a **scatterplot matrix**

Click Ball Point Pens Example

```
import pandas as pd
click = pd.DataFrame({"sales":[260.3,286.1,279.4,410.8,438.2,315.3,565.1,570.0,426.1,315.0,
    403.6,220.5,343.6,644.6,520.4,329.5,426.0,343.2,450.4,421.8,245.6,503.3,375.7,265.5,
    620.6,450.5,270.1,368.0,556.1,570.0,318.5,260.2,667.0,618.3,525.3,332.2,393.2,283.5,
    376.2,481.8],
  "ad":[5,7,6,9,12,8,11,16,13,7,10,4,9,17,19,9,11,8,13,14,7,16,9,5,18,18,
      5,7,12,13,8,6,16,19,17,10,12,8,10,12],
  "reps":[3,5,3,4,6,3,7,8,4,3,6,4,4,8,7,3,6,3,5,5,4,6,5,3,6,5,3,6,7,6,4,3,8,8,7,4,5,3,5,5],
  "eff":[4,2,3,4,1,4,3,2,3,4,1,1,3,4,2,2,4, 3,4,2,4,3,3,3,4,3,2,2,1,4,3,2,2,2,4,3,3,3,4,2]})

# or if you have a csv file
click = pd.read_csv("teach/data/click.csv")

click.describe()
            sales         ad      reps       eff
count   40.000000  40.000000  40.00000  40.00000
mean   411.287500  10.900000   5.00000   2.82500
std    123.854032   4.307418   1.64862   0.98417
min    220.500000   4.000000   3.00000   1.00000
25%    315.225000   7.750000   3.75000   2.00000
50%    398.400000  10.000000   5.00000   3.00000
75%    507.575000  13.250000   6.00000   4.00000
max    667.000000  19.000000   8.00000   4.00000

click.corr()
          sales        ad      reps       eff
sales  1.000000  0.880156  0.881778  0.001917
ad     0.880156  1.000000  0.776312  0.032057
reps   0.881778  0.776312  1.000000 -0.189638
eff    0.001917  0.032057 -0.189638  1.000000

from pandas.tools.plotting import scatter_matrix
scatter_matrix(click, figsize=(10,10))
scatter_matrix(click, figsize=(10,10), diagonal="kde")

import statsmodels.formula.api as sm
fit = sm.ols(formula="sales ~ ad", data=click).fit()
fit.summary()
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.775
Model:                            OLS   Adj. R-squared:                  0.769
Method:                 Least Squares   F-statistic:                     130.6
Date:                Fri, 01 Jan 2016   Prob (F-statistic):           7.33e-14
Time:                        10:04:17   Log-Likelihood:                -219.21
No. Observations:                  40   AIC:                             442.4
```

```
Df Residuals:                       38   BIC:                             445.8
Df Model:                            1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept    135.4336     25.907      5.228      0.000        82.989   187.879
ad            25.3077      2.214     11.430      0.000        20.825    29.790
==============================================================================
Omnibus:                        4.122   Durbin-Watson:                   1.721
Prob(Omnibus):                  0.127   Jarque-Bera (JB):                2.823
Skew:                           0.535   Prob(JB):                        0.244
Kurtosis:                       3.740   Cond. No.                         32.4
==============================================================================
```

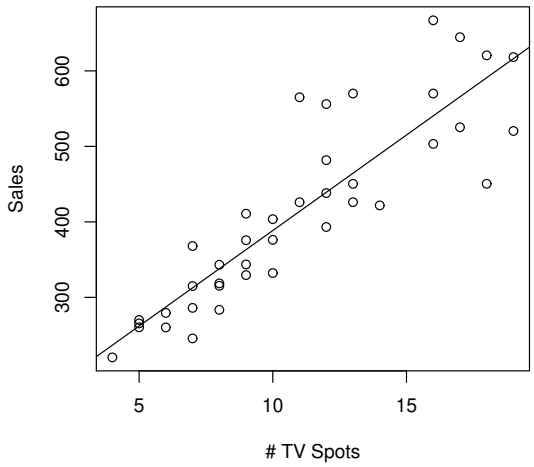
```
import matplotlib.pyplot as plt
plt.plot(click["ad"], click["sales"], 'o') # o option does not connect dots
plt.plot(click["ad"], fit.fittedvalues)
plt.legend(['Observed', 'Fitted'], loc='upper right')
plt.xlabel('Number of Ads')
plt.ylabel('Sales')
plt.title('Advertising and Sales')
plt.show()
```

```
fit = sm.ols(formula="sales ~ ad+reps+eff", data=click).fit()
fit.summary()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.881
Model:                            OLS   Adj. R-squared:                  0.871
Method:                 Least Squares   F-statistic:                     89.05
Date:                Fri, 01 Jan 2016   Prob (F-statistic):           1.02e-16
Time:                        09:48:52   Log-Likelihood:                -206.40
No. Observations:                  40   AIC:                             420.8
Df Residuals:                      36   BIC:                             427.6
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept     31.1504     34.175      0.911      0.368       -38.160   100.461
ad            12.9682      2.737      4.738      0.000         7.417    18.520
reps          41.2456      7.280      5.666      0.000        26.481    56.010
eff           11.5243      7.691      1.498      0.143        -4.074    27.123
==============================================================================
Omnibus:                        0.993   Durbin-Watson:                   2.104
Prob(Omnibus):                  0.609   Jarque-Bera (JB):                0.914
Skew:                           0.153   Prob(JB):                        0.633
Kurtosis:                       2.326   Cond. No.                         65.1
==============================================================================
```

Estimates From Click Ball Point Pens

```
> fit = lm(sales ~ ad, click)
> plot(click$ad, click$sales,
    xlab="# TV Spots", ylab="Sales")
> abline(fit)
> summary(fit)
> plot(fit)   # gives diagnostic plots
```




```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  135.434     25.907   5.228 6.50e-06 ***
ad            25.308      2.214  11.430 7.33e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 59.56 on 38 degrees of freedom
Multiple R-squared: 0.7747,Adjusted R-squared: 0.7687
F-statistic: 130.6 on 1 and 38 DF,  p-value: 7.327e-14
```

- Residual standard error: $S_\epsilon = 59.56$, standard error of estimate
- R-square = 0.7747: fraction of variation explained by model
- Adj R-sq: 0.7687 adjusted for number of parameters

Interpretation of Output

- The estimated regression model is

$$\hat{y} = 135 + 25.3\texttt{ad}$$

  What does 25.3 tell us? What about 135?

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  135.434     25.907   5.228 6.50e-06 ***
ad            25.308      2.214  11.430 7.33e-14 ***
```

- Standard errors are $S_{\hat{\beta}_0} = 25.91$ and $S_{\hat{\beta}_1} = 2.21$
- A 95% CI for β_1: $25.3 \pm 2.02 \times 2.21 \approx (20.8, 29.8)$

```
> confint(fit)
                2.5 %    97.5 %
(Intercept) 82.98862 187.87857
ad          20.82538  29.79001
```

- To test the hypotheses (with Type I error rate .05)

$$H_0 : \beta_1 = 0 \quad \text{versus} \quad H_1 : \beta_1 \neq 0,$$

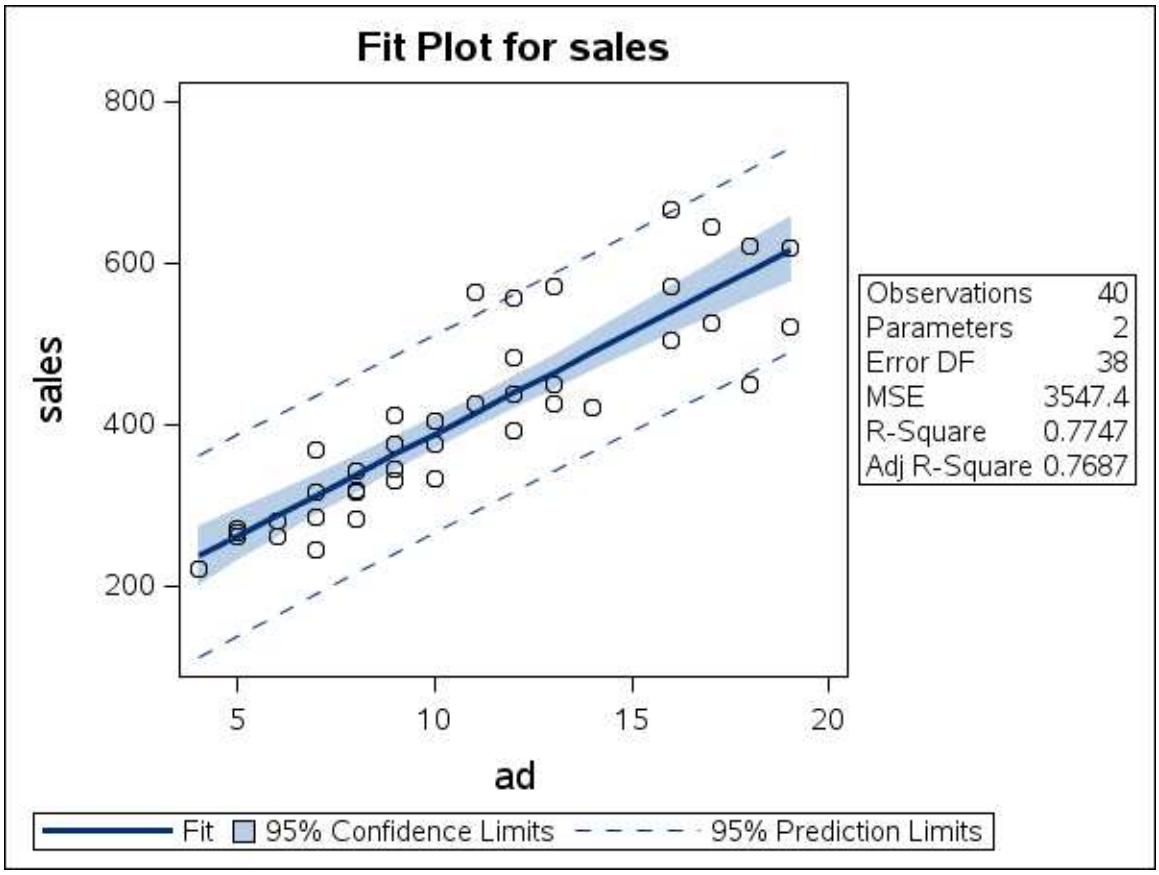
  P-value$= 7.33 \times 10^{-14} < .05$, so reject H_0 and conclude $\beta_1 \neq 0$.

- The expected sales when advertising is 5 (spots) is

$$\hat{y} = 135 + 25.3 \times 5 = 261.97$$

```
> predict(fit, data.frame(ad=5))
261.9721
```

Prediction and Confidence Intervals



- **Confidence interval for mean prediction** (shaded blue area): 95% confidence interval for $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$, i.e., indicates sampling variation of predicted values $\mathbb{E}(Y|x)$.
- **Prediction interval** (dashed lines): indicates where the middle 95% of the distribution of Y for a given x_0 falls. If we knew parameters, it would be $(\beta_0 + \beta_1 x_0) \pm 1.96\sigma_\epsilon$.

Additional Results

- The standard error of a new observation Y given x_0:

$$S_{Y|x_0} = \sqrt{S_\epsilon^2 \left(1 + \frac{1}{n}\right) + S_{\hat{\beta}_1}^2 (x_0 - \bar{x})^2}$$

  Example: find a 95% prediction interval for the mean sales when there are 5 ads. Hint: the 97.5 percentile of a t distribution with $40 - 2 = 38$ degrees of freedom is 2.024.

```
predict(fit, data.frame(ad=5), interval="prediction")
```

$$S_{Y|x_0} = \sqrt{3547 \left(1 + \frac{1}{40}\right) + 2.214^2 (5 - 10.90)^2} = 61.70$$

$$262.0 \pm 2.024 \times 61.699 = (137.1, 386.9)$$

- The standard error of a predicted value $\hat{Y}$ given x_0

$$S_{\hat{Y}|x_0} = \sqrt{\frac{S_\epsilon^2}{n} + S_{\hat{\beta}_1}^2 (x_0 - \bar{x})^2}$$

- Example: find a confidence interval for the mean sales when there are 5 ads.

```
predict(fit, data.frame(ad=5), interval="confidence")
```

$$S_{\hat{Y}|x_0} = \sqrt{\frac{3547}{40} + 2.214^2 (5 - 10.90)^2} = 16.104$$

$$262.0 \pm 2.024 \times 16.104 = (229.4, 294.6)$$

Geometric Interpretation of Simple Regression

- Assume $\mathbf{x}$ and $\mathbf{y}$ are mean-centered vectors
- We can show that the regression coefficient is

$$b = \frac{\mathbf{x}^\mathsf{T}\mathbf{y}}{\mathbf{x}^\mathsf{T}\mathbf{x}} = (\mathbf{x}^\mathsf{T}\mathbf{x})^{-1}\mathbf{x}^\mathsf{T}\mathbf{y}$$

- Recall that the *projection of* $\mathbf{y}$ *onto* $\mathbf{x}$ is

$$\text{proj}_\mathbf{x}\mathbf{y} = \frac{\mathbf{x}^\mathsf{T}\mathbf{y}}{\mathbf{x}^\mathsf{T}\mathbf{x}}\mathbf{x} = b\mathbf{x} = \hat{\mathbf{y}}$$

- This is a *right* triangle with sides $\mathbf{y}$, and $\hat{\mathbf{y}}$ (which lies on $\mathbf{x}$) and $\hat{\mathbf{e}} = \mathbf{y} - \hat{\mathbf{y}}$. The lengths of the sides are
  - (Hypotenuse) $\|\mathbf{y}\| = \sqrt{\text{SST}}$
  - $\|\hat{\mathbf{y}}\| = \sqrt{\text{SSR}}$
  - $\|\hat{\mathbf{e}}\| = \|\mathbf{y} - \hat{\mathbf{y}}\| = \sqrt{\text{SSE}}$

  The sum of squares equality SST = SSR + SSE follows from the Pythagorean theorem
- $\text{SSR} = \|\hat{\mathbf{y}}\|^2 = \|b\mathbf{x}\|^2 = b^2\mathbf{x}^\mathsf{T}\mathbf{x} = b^2S_x^2(n-1)$

Your Turn

1. You have five machines. The following data gives the age of the machines in years and the annual maintenance cost in thousands of dollars:

```
> machine = data.frame(age=c(2,5,9,3,8), cost=c(6,13,23,5,22))
```

   (a) Draw a scatterplot and describe the relationship.
   (b) Find the correlation between age and cost. What does it tell you?
   (c) Find the equation predicting cost from age. **Interpret the slope and intercept**.
   (d) Superimpose the regression line on your scatterplot.
   (e) What would you expect the annual maintenance to be for a machine that is 7 years old?
   (f) What is a typical size for the prediction errors?
   (g) What fraction of the variation in cost is explained by knowing the age of the machines?
   (h) Is the relationship between age and cost statistically significant?
   (i) A colleague suggested to use \$20,000 per year per machine for planning purposes. Perform a test at the 5% level to see if this is reasonable.

2. The amounts of a chemical compound y, which was dissolved in 100 grams of water at various temperatures, x° C, were recorded:

```
dat = data.frame(
  x=c(0,0,0, 15,15,15, 30,30,30,  45,45,45,  60,60,60,  75,75,75),
  y=c(8,6,8, 12,10,14, 25,21,24,  31,33,28,  44,39,42,  48,51,44))
```

   (a) Find the equation of the regression line.
   (b) Graph the line on a scatterplot.
   (c) Compute and interpret the standard error of the estimate.
   (d) Compute and interpret the coefficient of determination.
   (e) Find a 99% CI for β_0
   (f) Find a 99% CI for β_1
   (g) Test at the 1% level if the slope differs from 0.
   (h) Estimate the mean amount that will dissolve in 100 grams of water at 50° C.
   (i) Find a 99% CI for the mean amount that will dissolve at 50° C.
   (j) Find a 99% PI for the amount that will dissolve at 50° C.

3. Consider the circulation and the open line rate (price per line for an ad placed just once) for selected large newspapers shown below

```
dat=data.frame(
circ = c(2081995,1374858,1284613,1057536,970051,963069,828236,779259,768288,
  691771,663693,657015,645623,533384,528777,514702,492002,486426,
  443592,349182),
linerate=c(37.65,18.48,14.50,14.61,16.47,16.07,13.82,13.05,13.78,12.25,10.53,
  14.18,12.83,7.81,5.17,11.08,6.58,8.77,6.03,6.77),
row.names=c("WSJ","NY Daily News","USA Today","LA Times","NYT", "NY Post",
  "Philadelphia","Chi Tribune","Wash Post","SF Chronicle","Chi Sun Times",
  "Detroit News","Detroit Free Press","Long Island Newsday","KC Times",
  "Miami Herald","Cleveland","Milwaukee","Houston","Baltimore"))
```

(a) Create a scatterplot of the open line rate against circulation. Comment.

(b) Find and interpret the correlation of open line rate with circulation. Is the correlation reasonable from a business perspective?

(c) Find the regression equation to predict open line rate from circulation. Superimpose the regression line on your scatterplot.

(d) Test if the association between the open-line rate and circulation is significant ($\alpha = .05$).

(e) Find the predicted and residual value for the *New York Times*. Interpret these values. In particular, is the open line rate higher or lower than what you would expect for a newspaper with its circulation?

(f) There is an outlier visible in the scatterplot. Let's test to see if it could reasonably be from the same population as the others by treating it as a new observation. Remove *The Wall Street Journal* (WSJ) from the data set and find the regression equation to predict the open line rate from circulation for the other newspapers.

(g) Find the two-sided 95% prediction interval (PI) for a new observation, with X_0 being the circulation of *WSJ*.

(h) Test whether or not *WSJ* is an outlier by seeing if its open line rate is in the PI.

(i) The *milline rate* is defined as the open line rate divided by the circulation, in millions. Thus, it is the cost per line oof advertising per million circulation. This adjustment should take care of some of the differences in advertising rates due to circulation. That is, one explanation of the open line rate is that it is proportional to circulation. If it is just proportional, there should be nothing left in the milline rate to be explained by circulation. On the other hand, if there is an additional advantage or penalty to being big, the circulation should help explain the variation in milline rates. Let's use regression analysis to see if three is anything left in the milline rate to be explained by circulation. Draw a scatterplot of milline rate against circulation.

(j) Find and interpret the correlation between circulation and milline rate.

(k) What percentage of the variation in milline rate is explained by circulation?

(l) Test to see if there is a significant relationship between circulation and milline rate.

(m) Write a paragraph explaining and interpreting your results.

4. The data below gives mailing-list size (thousands of names) and sales (thousands of dollars) for a group of catalogs.

```
dat = data.frame(
  size=c(168, 21, 94, 39, 249, 43, 589, 41),
  sales = c(5178, 2370,3591, 2056, 7325, 2449, 15708, 2469))
```

(a) How strong is the association between these two variables? Find the appropriate summary measure and interpret it.

(b) Find the equation to predict sales from the size of the mailing list.

(c) What level of sales would you expect for a a catalog mailed to 5,000 people?

(d) What percent of the variation in the list size can be explained by the fact that some generated more sales than others?

(e) Is there a significant relationship between list size and sales? How do you know?

5. JWHT problem 8a,b on pages 121–2 (Hint: see §2.3.4 on page 48–49.) If you use the data from the author's website you will need to read about the `na.strings` option. Note: omit part c for now. Answer these questions about the output. Type the following to read it in:

```
auto = read.csv("Downloads/auto.csv", na.strings="?")
auto$origin = factor(auto$origin, 1:3, c("US", "Europe", "Japan"))
```

(a) What is the estimated regression equation?

(b) What does the slope tell you?

(c) How much uncertainty is associated with the slope estimate?

(d) What does the residual standard error tell you?

(e) Using this model, is there a significant relationship between mpg and horsepower?

(f) What fraction of the variation in mpg is explained by using this linear function of horsepower?

(g) What is the predicted mpg associated with a horsepower of 98?

(h) What is the 95% prediction interval for the predicted mpg associated with a horsepower of 98?

(i) What is the 99% confidence interval for the mean prediction of mpg when horsepower is 98?

(j) What is a 90% confidence interval for the slope?

(k) In looking at the scatterplot and fitted model, note any violations of the model assumptions.

Answers

1. Machine problem. Do in live session?

2. See code below. (a) $\hat{y} = 5.8254 + 0.5676x$. (c) $S_e = 2.57$: typical size of a residual (different between observed and predicted) (d) $R^2 = 0.973$. (e) $[2.686, 8.965]$. (f) $[0.498, 0.637]$. (g) $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$, either

$P = 5.7 \times 10^{-14} \ll 0.01$ so reject or $0 \notin [0.498, 0.637]$ so reject H_0. (h) $\hat{y}(50) = 34.2$. (i) [32.23569, 36.17701]. (j) [26.43824, 41.97446].

```
fit = lm(y~x, dat)    # part a
plot(y~x, dat)        # part b
abline(fit)
summary(fit)          # part c, d, g
confint(fit, level=.99)  # part f, g
new = data.frame(x=50)
predict(fit, new, interval="conf", level=.99) # part h,i
predict(fit, new, interval="pred", level=.99) # part j
```

3. Newspaper problem. See code below. (a) The relationship is mostly linear but there is an outlier that could have substantial influence. (b) $r = .93$: larger open line rates have a positive association with circulation. (c) $\widehat{\text{linerate}} = 0.282 + 0.0000158\text{circ}$; (d) $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$, $P = 2.9 \times 10^{-9} \ll 0.05$ so reject H_0. (e) Note that NYT is row 5. See R below to find $y_5 = 16.5$, $\hat{y}_5 = 15.6$ and $\hat{e}_5 = 0.85$; higher. (f) WSJ is paper 1. $\widehat{\text{linerate}} = 3.11 + 0.0000117\text{circ}$. (g/h) $37.6 \notin [20.6, 34.2]$, conclude WSJ is an outlier. (i) Not much of a relationship. (j) $r = -0.125$. (k) $R^2 = (-0.125)^2 = 0.0155$. (l) Do either of this: $H_0 : \rho = 0$ versus $H_1 : \rho \neq 0$ or $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$ $P = 0.6 > 0.05$, so do not reject H_0.

```
plot(linerate~circ, dat)            # part a
cor(dat)                            # part b
fit = lm(linerate~circ, data=dat)   # part c
abline(fit)                         # part c
summary(fit)                        # part c,d
dat[5,]                             # part e
fit$fitted.values[5]
fit$residuals[5]
fit2 = lm(linerate~circ, data=dat[-1,]) # part f
summary(fit2)
dat[1,]                             # part g
predict(fit2, dat[1,], interval = "pred")
dat$milline=1000000*dat$linerate/dat$circ # part i
plot(milline~circ, dat)
cor(dat)                            # part j
fit3 = lm(milline~circ, dat)        # part k
summary(fit3)
cor.test(dat$milline, dat$circ)
```

4. (a) $r = 0.999$. (b) $\hat{y} = 1393.825 + 24.112\text{size}$. (c) $\hat{y}(5000) = 121954$. (d) $R^2 = 0.997$. (e) $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$, $P = 5.9 \times 10^{-9} \ll 0.05$ so reject H_0.

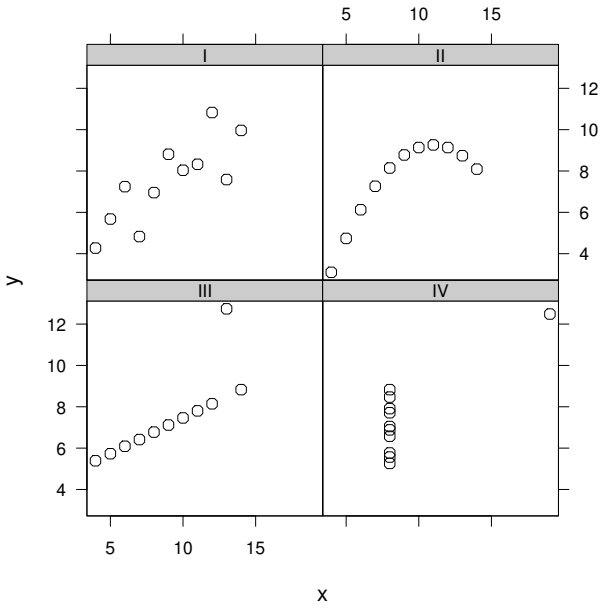
5. JWHT 8. See code below (a) $\widehat{\text{mpg}} = 39.94 - 0.1578\text{horsepower}$. (b) Every unit increase in horsepower is associated with a .1578 decrease in mpg on the average. (c) Standard error is 0.006446. (d) Typical size of residuals. (e) Yes, $P < 2 \times 10^{-16} < .05$. (f) $R^2 = .6059$. (g) 24.47 mpg. (h) 14.8094 to 34.12476. (i) 23.81669 to 25.11747. (j) $[-0.1684719, -0.1472176]$. (k) The scatterplot shows that the relationship is not linear and the error variance is not constant.

```
fit = lm(mpg ~ horsepower, auto)  # part a
summary(fit)
plot(mpg ~ horsepower, auto)
abline(fit)
predict(fit, data.frame(horsepower=98), interval="pred")
predict(fit, data.frame(horsepower=98), interval="conf",
  level=.99)
confint(fit, level=.90)
plot(fit)   # part c
```

Anscombe's Example

```
dat = data.frame(
  dataset = factor(c(rep(1,11),rep(2,11),rep(3,11),rep(4,11)),
    labels=c("I","II","III","IV")),
  x=c(10,8,13,9,11,14,6,4,12,7,5, 10,8,13,9,11,14,6,4,12,7,5,
      10,8,13,9,11,14,6,4,12,7,5, rep(8,10),19),
  y=c(8.04,6.95,7.58,8.81,8.33,9.96,7.24,4.26,10.84,4.82,5.68,
      9.14,8.14,8.74,8.77,9.26,8.1,6.13,3.1,9.13,7.26,4.74,
      7.46,6.77,12.74,7.11,7.81,8.84,6.08,5.39,8.15,6.42,5.73,
      6.58,5.76,7.71,8.84,8.47,7.04,5.25,5.56,7.91,6.69,12.5)
)
```

```
> library(lattice)
> xyplot(y~x | dataset, dat)
> tapply(dat$x, dat$dataset, mean)
  I  II III  IV
  9   9   9   9

> tapply(dat$y, dat$dataset, mean)
       I       II      III       IV
7.500909 7.500909 7.500000 7.482727

> tapply(dat$x, dat$dataset, var)
  I  II III  IV
 11  11  11  11

> tapply(dat$y, dat$dataset, var)
       I       II      III       IV
4.127269 4.127629 4.122620 4.151322

> summary(lm(y ~ dataset*x, dat))
```



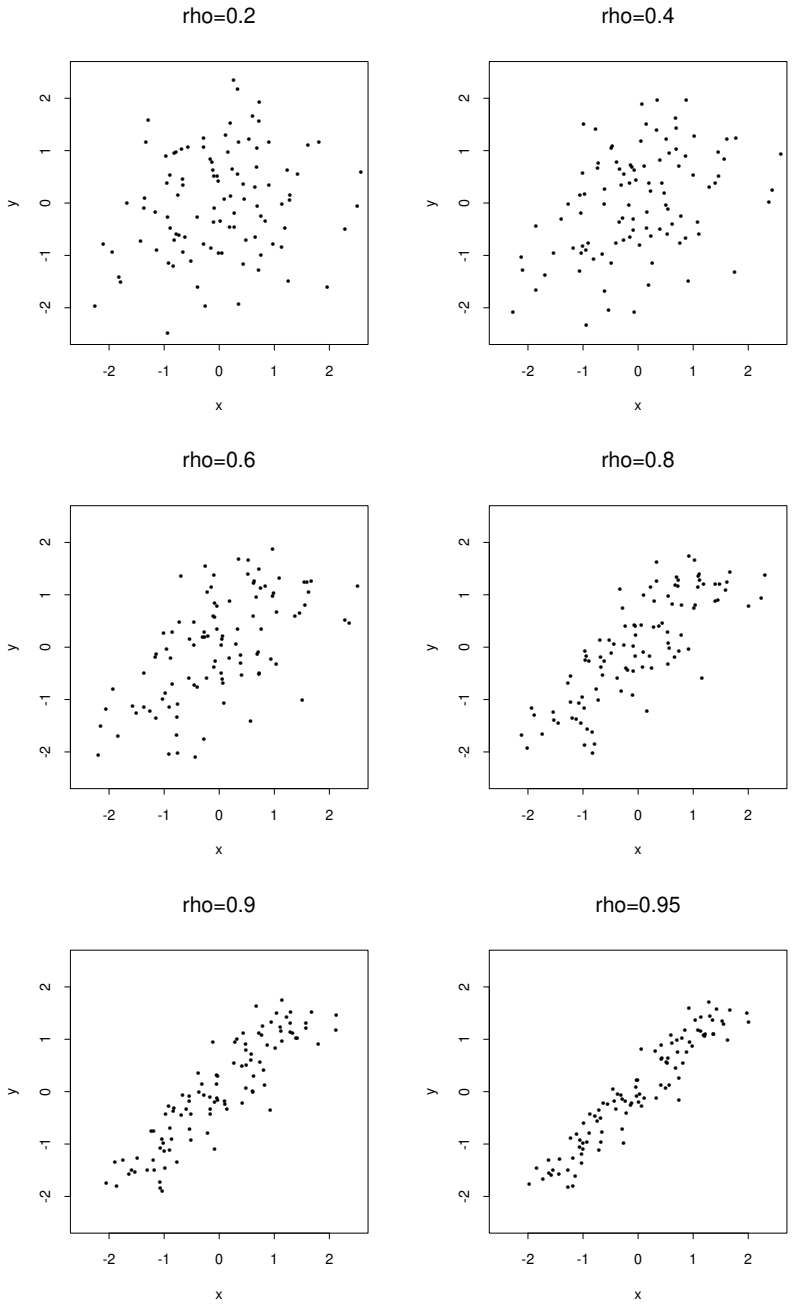

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.000e+00  1.125e+00   2.666 0.011431 *
datasetII     8.182e-04  1.592e+00   0.001 0.999593
datasetIII    2.364e-03  1.592e+00   0.001 0.998823
datasetIV    -3.291e-02  1.592e+00  -0.021 0.983618
x             5.001e-01  1.180e-01   4.239 0.000149 ***
datasetII:x  -9.091e-05  1.668e-01  -0.001 0.999568
datasetIII:x -3.636e-04  1.668e-01  -0.002 0.998273
datasetIV:x   1.636e-03  1.668e-01   0.010 0.992229
```

Always look at your data!

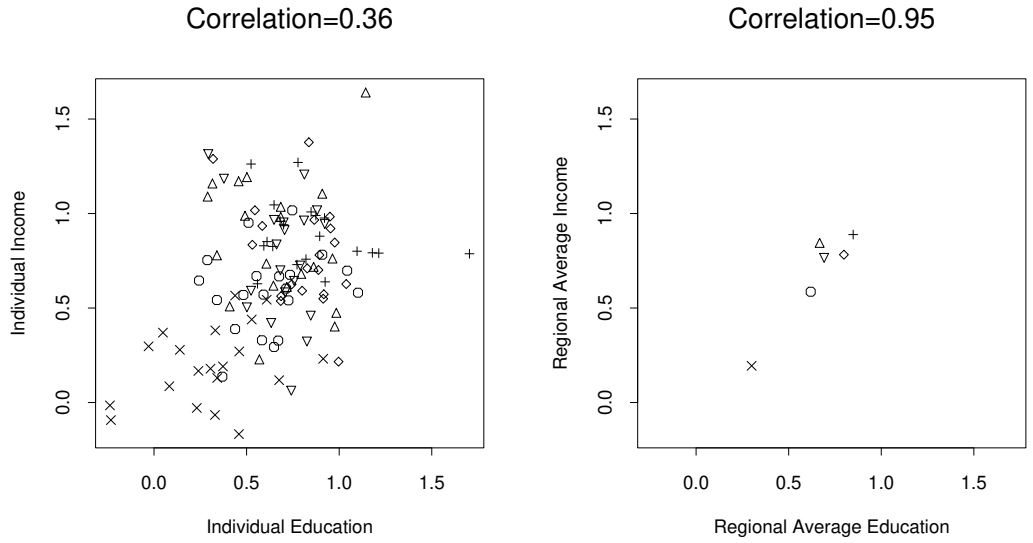
Measures of Association: Correlations

- Commonly used measures:
  - Pearson product moment correlation: Population value ρ, sample statistic r, $-1 \leq \rho \leq 1$. R, use `cor`, `cor.test`.
- Objective: measure *direction* and *strength* of the *association* between two variables.
  - Positive values ($\rho > 0$) indicate positive association — when one variable increases, so does the other.
  - Negative values ($\rho < 0$) indicate negative association — when one variable increases, the other decreases.
  - Zero values ($\rho \approx 0$) indicate no *linear* association.

Measures of Association: Correlations



Ecological Correlations



- **Ecological correlations** are based on rates or averages and tend to overstate the strength of an association. Beware whenever rates or averages are correlated!
- For example, beware of a correlation involving average store sales

Notation

- Suppose x has mean μ_x and standard deviation σ_x. The **standard units** (also called Z**-scores**) tell how many standard deviations each observation is from the mean:

$$Z_x = \frac{x - \mu_x}{\sigma_x}$$

  Use `scale` to compute Z-scores and `cor` to compute correlations in R.

- Let $x_1, \ldots, x_n$ be observations from a distribution with mean μ_x and standard deviation σ_x. The *sample mean* and *sample standard deviation* are

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{and} \quad S_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- Let $y_1, \ldots, y_n$ be observations from a distribution with mean μ_y and standard deviation σ_y. Likewise let $\bar{y}$ be the sample mean and S_y be the sample standard deviation.

Interpretation and Computation of Correlations

- **Pearson correlation**

$$\begin{aligned}\rho &= \frac{\text{cov}(X,Y)}{\sigma_x\sigma_y} = \frac{\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y)}{\sqrt{\sum_{i=1}^{N}(x_i-\mu_x)^2\sum_{i=1}^{N}(y_i-\mu_y)^2}} \\ &= \sum_{i=1}^{N}\left[\frac{(x_i-\mu_x)}{\sqrt{\sum(x_i-\mu_x)^2}}\times\frac{(y_i-\mu_y)}{\sqrt{\sum(y_i-\mu_y)^2}}\right] \\ &= \frac{1}{N}\sum_{i=1}^{N}\left[\frac{(x_i-\mu_x)}{\sigma_x}\times\frac{(y_i-\mu_y)}{\sigma_y}\right]\end{aligned}$$

  Thus, the Pearson correlation coefficient is the *average of the products of the standardized versions of* x *and* y.

- We estimate ρ with r

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

- Note that Pearson correlations are unaffected by
  - interchanging the two variables
  - replacing a variable x with $ax+b$, where $a>0$
- There are formulas for hypothesis tests and CIs, e.g., under $H_0:\rho=0$, $t=r\sqrt{(n-2)/(1-r^2)}$ is distributed T_{n-2}.

Geometric Interpretation of Correlation

- Let $\tilde{x}_i = x_i - \mu_x$ and $\tilde{y}_i = y_i - \mu_y$ be *mean-centered* versions of x and y
- Consider $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_N)^\mathsf{T}$ and $\tilde{y} = (\tilde{y}_1, \ldots, \tilde{y}_N)^\mathsf{T}$ as vectors in N-space. Recall from analytic geometry that the angle θ between the two vectors can be computed using

$$\cos\theta = \frac{\tilde{x}^\mathsf{T}\tilde{y}}{\|\tilde{x}\|\,\|\tilde{y}\|} = \frac{\sum \tilde{x}_i\tilde{y}_i}{\sqrt{\sum \tilde{x}_i^2}\sqrt{\sum \tilde{y}_i^2}} = \rho$$

  The Pearson correlation is thus the *cosine of the angle between the (mean-centered) vectors in N space.*
- Recall that $\cos\theta = 0$ if and only if $\tilde{x}$ is perpendicular (*orthogonal*) to $\tilde{y}$
- The term *orthogonal* is sometimes used instead of *uncorrelated*
- The coefficient of determination also follows

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}} = \frac{\text{SSR}}{\text{SST}} = \left(\frac{\text{adj}}{\text{hyp}}\right)^2 = \cos^2\theta = \rho^2$$

Example: Maintenance Costs

The age in years and maintenance cost in thousands of dollars for 5 machines are provided in the table below.

	Age	Cost	Standard Units		Product
	x	y	Z_x	Z_y	$Z_x \cdot Z_y$
	2	6	-1.247	-1.023	1.275
	5	13	-0.147	-0.105	0.015
	9	23	1.320	1.206	1.592
	3	5	-0.880	-1.154	1.015
	8	22	0.953	1.075	1.025
Mean	5.4	13.8	0	0	0.985
σ	2.728	7.626	1	1	

$$r = \frac{102.40}{\sqrt{37.2 \times 290.8}} = 0.9845$$

Note: I use σ rather than S

```
> machine <- data.frame(age=c(2,5,9,3,8), cost=c(6,13,23,5,22))
> plot(machine)
> cor(machine)       # full correlation matrix
           age      cost
age  1.0000000 0.9845353
cost 0.9845353 1.0000000

> cor.test(machine$age, machine$cost)    # or use cor.test(~age+cost, machine)

Pearson's product-moment correlation

data:  machine$age and machine$cost
t = 9.734, df = 3, p-value = 0.002303
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval: 0.7784347 0.9990256
sample estimates: 0.9845353
```

Bivariate Normal Distribution

- Consider $p = 2$ random variables X_1 and X_2.
- Let $\mathbb{E}(X_j) = \mu_j$, $\mathbb{V}(X_j) = \sigma_j^2$, and the correlation be

$$\rho = \frac{\mathrm{Cov}(X_1, X_2)}{\sigma_1 \sigma_2} = \frac{\sigma_{12}}{\sigma_1 \sigma_2},$$

  where $\sigma_{12} = \rho \sigma_1 \sigma_2$ is the *covariance* of X_1 and X_2.
- The *mean vector* of random vector $\mathbf{X} = (X_1, X_2)^\mathsf{T}$ is

$$\boldsymbol{\mu} = \mathbb{E}(\mathbf{X}) = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$$

- The *covariance matrix* is

$$\boldsymbol{\Sigma} = \mathbb{E}[(\mathbf{X} - \boldsymbol{\mu})^\mathsf{T}(\mathbf{X} - \boldsymbol{\mu})] = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}.$$

- The *bivariate normal PDF* is $(-1 < \rho < 1, \sigma_i > 0)$

$$f(\mathbf{x}) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right],$$

  where $\boldsymbol{\Sigma}^{-1}$ is the inverse of $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)} \begin{pmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{pmatrix}.$$

Justification of Elliptical Contours

- Assume uncorrelated variables with mean $(0, 0)^\mathsf{T}$, i.e.,

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \quad \text{and} \quad \mathbf{\Sigma}^{-1} = \begin{pmatrix} 1/\sigma_1^2 & 0 \\ 0 & 1/\sigma_2^2 \end{pmatrix}.$$

- A *contour* is the set of points such that $f(\mathbf{x}) = c$ for some fixed c, i.e., $\{\mathbf{x} : f(\mathbf{x}) = c\}$.

- We need to solve

$$c = f(\mathbf{x}) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{x}\right)$$

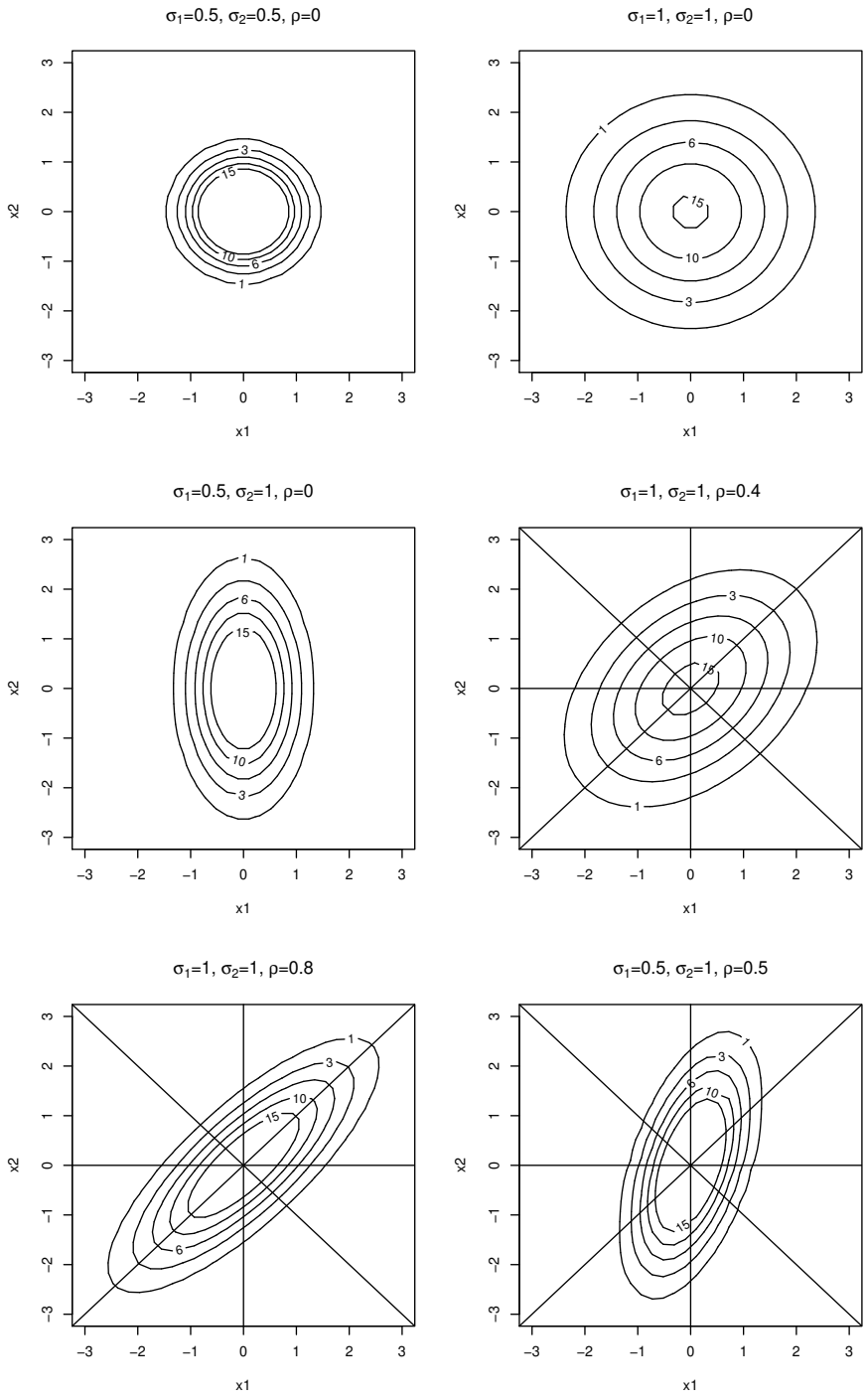
$$2\pi\sigma_1\sigma_2 c = \exp\left(-\frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{x}\right)$$

$$-2\log(2\pi\sigma_1\sigma_2 c) = \mathbf{x}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{x} = \frac{x_1^2}{\sigma_1^2} + \frac{x_2^2}{\sigma_2^2}$$

  Which we recognize as an ellipse (for sufficiently small c).

- For correlated variables we can show that the ellipse is rotated using the eigenvectors of $\mathbf{\Sigma}$.

Various Bivariate Normal PDFs



Correlation and regression

- Let X and Y have a bivariate normal distribution with means μ_x and μ_y, and variances σ_x^2 and σ_y^2, respectively. Their covariance is σ_{xy} and their correlation is $\rho = \sigma_{xy}/(\sigma_x\sigma_y)$.
- Let

$$\beta_1 = \frac{\sigma_y}{\sigma_x}\rho, \quad \beta_0 = \mu_y - \beta_1\mu_x, \quad \text{and} \quad \sigma^2 = \sigma_y^2(1-\rho^2)$$

  Then the following hold:

  1. The marginal distributions of X and Y are $\mathcal{N}(\mu_x, \sigma_x^2)$ and $\mathcal{N}(\mu_y, \sigma_y^2)$, respectively
  2. The conditional distribution of Y given $X = x$ is

$$Y|x \sim \mathcal{N}(\beta_0 + \beta_1 x, \sigma^2)$$

Multiple Linear Regression

Multiple linear regression allows us to study the relationship between a response variable and p predictor variables.

- x_{ij}: value of the j^{th} **predictor variable** on observation i ($i = 1, \ldots, n$ and $j = 1, \ldots, p$). $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\mathsf{T}$ is an $n \times p$ matrix with row $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{ip})^\mathsf{T}$ (the extra 1 is for the intercept)
- y_i: value of **dependent variable**, $\mathbf{y} = (y_1, \ldots, y_n)^\mathsf{T}$.
- β_0: the **intercept**. When $x = 0$, on average $Y = \beta_0$.
- β_j: **slope coefficient for variable** j. A unit increase in x_j is associated with an increase in y of β_j, *on average* after controlling for other predictors.
- Multiple regression model:

  $$y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p + \epsilon_i = \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta} + \epsilon_i$$

  where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^\mathsf{T}$ and ϵ_i are iid, normal with mean 0 and standard deviation σ_ϵ. This implies

  $$\mathbb{E}(Y|x_i) = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p$$

- As before, σ_ϵ^2 is called the **error variance** and σ_ϵ: the **standard deviation of the errors**.
- We write this compactly as

  $$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \text{MVN}(\mathbf{0}, \sigma^2\mathbf{I})$$

Estimating the Regression Model

- We don't know values of **parameters** β_0, $\beta_1, \ldots, \beta_p$, and σ_ϵ.
- **Estimates** denoted by $b_0, b_1, \ldots, b_p$, and S_ϵ
- $\mathbb{E}(Y_i|x_i)$ called **fitted**, **predicted**, or **"y-hat" values**:

$$\hat{y}_i = b_0 + b_1 x_{i1} + \cdots + b_p x_{ip}$$

- The **residual** for observation i is

$$\hat{\epsilon}_i = y_i - \hat{y}_i = y_i - \left(b_0 + \sum_{j=1}^{p} b_j x_{ij} \right)$$

- We choose $\mathbf{b}$ to minimize the **least-squares criterion** (SSE means **sum of squared errors**):

$$\text{SSE} = \sum_{i=1}^{n} \hat{\epsilon}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = (\mathbf{y} - \mathbf{X}\hat{\beta})^\mathsf{T}(\mathbf{y} - \mathbf{X}\hat{\beta})$$

- The **ordinary least squares** (OLS) estimates of $\boldsymbol{\beta}$:

$$\frac{\partial \text{SSE}}{\partial \hat{\beta}} = -2\mathbf{X}^\mathsf{T}(\mathbf{y} - \mathbf{X}\hat{\beta}) \quad \Longrightarrow \quad \hat{\beta} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$$

- Predicted values are given by

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{H}\mathbf{y}$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}$ is called the **hat matrix**

Some Properties of OLS Estimates

- $\hat{\beta}$ is unbiased (proof)

$$\mathbb{E}(\hat{\beta}) = \mathbb{E}[(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}] = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbb{E}(\mathbf{X}\boldsymbol{\beta} + \mathbf{e}) = \boldsymbol{\beta}$$

- $\mathbb{V}(\hat{\beta}) = \sigma_\epsilon^2(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$

$$\mathbb{V}[(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}] = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbb{V}(\mathbf{y})\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1} = \sigma_\epsilon^2(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$$

- Gauss-Markov Theorem: $\hat{\beta}$ has a covariance matrix that is "smaller" than that of any other linear estimator and is called the *best linear unbiased estimator* ("BLUE").
- Estimate σ_ϵ^2 with the **mean squared error**

$$S_\epsilon^2 = \text{MSE} = \frac{\text{SSE}}{n - p - 1}$$

and σ_ϵ with the **residual standard error** (a.k.a. **root mean squared error**) $S_\epsilon = \sqrt{S_\epsilon^2}$.

- If $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I})$ and the linear model is correct, then $\hat{\beta}$ has a multivariate normal distribution because it is a linear transformation of normally distributed $\boldsymbol{\epsilon}$:

$$\hat{\beta} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}) = \boldsymbol{\beta} + (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\boldsymbol{\epsilon}$$

- $S_{b_j} = S_\epsilon\sqrt{v_j}$ is called the *standard error* of b_j, where v_j is the j^{th} diagonal element of $(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$, and $(b_j - \beta_j)/S_{b_j}$ has a t distribution.
- A $(1-\alpha)\%$ confidence region for $\boldsymbol{\beta}$ is

$$(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \leq \hat{\sigma}^2\chi_{p+1}^{2(1-\alpha)} \tag{3.15}$$

Geometry of Least Squares

The geometrical interpretation is often useful

- $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ is the orthogonal projection of $\mathbf{y}$ onto the subspace $\mathcal{M} \subset \mathbf{R}^n$ spanned by the columns of $\mathbf{X}$. This is true even if $\mathbf{X}$ is not of full column rank.
- The columns of $\mathbf{X}$ are basis vectors for $\mathcal{M}$ and $\hat{\boldsymbol{\beta}}$ gives the coordinates of $\hat{\mathbf{y}}$ with respect to this basis
- Hat matrix $\mathbf{H}$ projects n-vectors onto $\mathcal{M}$
- $\mathbf{P} = \mathbf{I} - \mathbf{H}$ projects onto the orthogonal complement of $\mathcal{M}$
- Residual vector $\mathbf{P}\mathbf{y} = \mathbf{y} - \hat{\mathbf{y}}$ is orthogonal to $\mathcal{M}$
- The three vectors $\mathbf{y}$, $\hat{\mathbf{y}}$, and $\mathbf{y} - \hat{\mathbf{y}}$ form a right triangle in n-space, where $\mathbf{y}$ is the hypotenuse
- If $\mathbf{y}$ and columns of $\mathbf{X}$ are mean centered
  - **total sum of squares**: $\mathbf{y}^\mathsf{T}\mathbf{y} = (n-1)S_y^2$, the squared length of the hypotenuse (S_y^2 is the variance of y)
  - **regression sum of squares**: $\hat{\mathbf{y}}^\mathsf{T}\hat{\mathbf{y}} = \mathbf{y}^\mathsf{T}\mathbf{H}\mathbf{y}$
  - SSE $= (\mathbf{y} - \hat{\mathbf{y}})^\mathsf{T}(\mathbf{y} - \hat{\mathbf{y}}) = \mathbf{y}^\mathsf{T}\mathbf{P}\mathbf{y}$
  - Pythagoras gives us the ANOVA equality

$$\mathbf{y}^\mathsf{T}\mathbf{y} = \mathbf{y}^\mathsf{T}\mathbf{H}\mathbf{y} + \mathbf{y}^\mathsf{T}\mathbf{P}\mathbf{y}$$

Basic Model Building Process

Suppose that (1) your interest is confirmatory and (2) the model is correct (i.e., y is really a linear function of the specified x variables plus additive, normal, independent, homoscedastic errors)

1. Inspect your data for outliers, typos, missing values, etc.
   - Generate n, means, mins, and maxs of each variable
   - Generate boxplots or histograms of each variable
   - Generate a scatterplot matrix for small data sets
   - Generate correlation matrix to assess correlations between predictor variables and pairwise correlations with DV
2. Estimate model and check residual and normality plots, VIFs
3. Test **overall significance of the model**

   $H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$

   H_1 : at least one $\beta_j \neq 0$
4. If you can reject H_0 in Step 3, interpret model and test significance of individual coefficients. If you cannot reject H_0 in Step 3, don't try to test individual coefficients.

- In practice you usually will not know the correct model, which complicates the process substantially.
- If only goal is prediction, hypothesis tests not necessary (focus on getting "honest" estimates of mean squared error)

Estimates from Click Ball Point Pens

```
> fit = lm(sales ~ ad + reps + eff, click)
> summary(fit)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   31.150     34.175   0.911    0.368
ad            12.968      2.737   4.738 3.34e-05 ***
reps          41.246      7.280   5.666 1.95e-06 ***
eff           11.524      7.691   1.498    0.143
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 44.42 on 36 degrees of freedom
Multiple R-squared: 0.8812,Adjusted R-squared: 0.8714
F-statistic: 89.05 on 3 and 36 DF,  p-value: < 2.2e-16
```

- Is the regression significant? *Solution:*

  $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$ versus H_1: at least one $\beta_j \neq 0$.
  $P = 2.2e-16 < .05$, so reject H_0 and conclude that at least one of the predictors is predictive.

- State estimated regression equation. *Solution:*

$$\hat{y} = 31.15 + 12.97\texttt{ad} + 41.25\texttt{reps} + 11.52\texttt{eff}$$

- Interpret the coefficient for `reps` (41.25).
- Here $\hat{\beta}_1 = 13.0$ but on page 13, $\hat{\beta}_1 = 25.3$. **Which is right?**

Estimates from Click (Continued)

- Construct at 95% CI for `reps`.
  Solution: $41.25 \pm 2.028 \times 7.28 = [26.5, 56.0]$

```
> confint(fit)
                 2.5 %    97.5 %
(Intercept) -38.159815 100.46059
ad            7.416798  18.51953
reps         26.480882  56.01037
eff          -4.074175  27.12268
```

- Is `eff` different from zero (use .05 level)?

  $H_0 : \beta_3 = 0$ versus $H_1 : \beta_3 \neq 0$.

$$\begin{aligned} \mathsf{P}(b_3 > 11.52) &= P\left(T_{36} > \frac{11.52 - 0}{7.6912}\right) \\ &= P(T_{36} > 1.498) = 0.0714 \end{aligned}$$

  The P value is thus `2*(1-pt(1.498, 36)) = 0.1429`. We cannot reject H_0 because $0.1428 > 0.05$ and we cannot conclude that wholesaler efficiency affects sales.

- Predict sales for `ad`=4, `reps`=3, `eff`=1. *Solution:*

$$\hat{y} = 31.15 + 12.97 \times 4 + 41.25 \times 3 + 11.52 \times 1 = 218.3$$

```
> predict(fit, data.frame(ad=4, reps=3, eff=1))
       1
218.2842
```

Comparing Regression Coefficients

- *Question:* Which of the variables is more "important" in explaining sales?
- *Answer:* The coefficients are not directly comparable because of differences in units of measurement.
- *Ideal solution:* convert to commensurate units, e.g., dollars.
- *Possible solution:* **Standardized regression coefficients** (all variables standardized $z = (x - \bar{x})/s_x$ before the analysis to have mean 0 and variance 1). The "unit" of measurement is now the standard deviation (units cross out)

```
> Zclick = as.data.frame(scale(click[,1:4]))
> fit = lm(sales ~ ad + reps + eff -1, Zclick)  # -1 drops intercept
> summary(fit)
Coefficients:
     Estimate Std. Error t value Pr(>|t|)
ad    0.45101    0.09390   4.803 2.59e-05 ***
reps  0.54902    0.09559   5.744 1.40e-06 ***
eff   0.09157    0.06028   1.519    0.137
```

- *Possible solution:* compare t scores ($t = b_j/S_{b_j}$, reported in the fourth column of output, units also cross out)
- See Bring (1994) for discussion

Standardized Regression Coefficients

- Theorem: the standardized regression coefficient is $b_j S_{x_j}/S_y$, where S_{x_j} and S_y are the standard deviations of x_j and y, respectively, and b_j is the (unstandardized) regression estimate for variable j.
- Standardized regression coefficients ...
  - Also called "*beta*" coefficients
  - Interpreted as a standard deviation increase in x_j is associated with "beta" standard deviations in y
  - Equals correlation r between x and y when $p = 1$ predictor
- Do not use "beta" coefficients blindly:
  - If x_j is a 0-1 variable, then the standard deviation is $\sqrt{\bar{x}_j(1-\bar{x}_j)}$
  - If you are analyzing a designed experiment, you select the x_j values and thus the standard deviations
  - "Beta" values are still a function of other variables in the model (multicollinearity)
  - "Beta" values do not consider costs

Your Turn

1. A commercial real estate company evaluates vacancy rates, square footage, rental rates, and operating expenses for commercial properties in a large metropolitan area in order to provide clients with quantitative information upon which to make rental decisions. The data in `commercial.txt` are taken from 81 suburban commercial properties that are the newest, best located, most attractive, and expensive for five specific geographic areas. It includes their age (`x1`), operating expenses and taxes (`x2`), vacancy rates (`x3`), total square footage (`x4`), and rental rates (`y`).[2]

   (a) Read the commercial data set into R and run basic descriptive statistics (counts, mins, maxs, means). Do the descriptives make sense? Hint:

   ```
   > comm = read.table("commercial.txt", header=T)
   > summary(comm)
   ```

   (b) Produce a scatterplot matrix and correlation matrix. Discuss the relationships between the variables. Hint: use `plot(comm)` and `cor(comm)`.

   (c) Regress rental rates on the four predictor variables. State the estimated regression equation. Hint:

   ```
   > fit = lm(y ~ x1 + x2 + x3 + x4, comm)
   > summary(fit)
   ```

   (d) Test whether the overall model is significant. State the null and alternative, P-value and decision. Hint: `summary(fit)`

   (e) What fraction of variation in rental rates is explained by these predictor variables?

   (f) If the overall regression model is significant in the previous part then test whether each of the individual regression coefficients equals 0 at the 5% level against a 2-sided alternative, i.e., $H_0 : \beta_j = 0$ for $j = 1, 2, 3, 4$. For each predictor, state the P-value and your decision.

   (g) Assume that the regression model you have estimated is appropriate. Three properties with the following characteristics did not have any rental information available.

[2] Solution: (a) The summary statistics make sense. Age ranges from 0 to 20, which is reasonable for real estate properties. Operating expenses and taxes are positive. The vacancy rate is between 0 and 1. Square footage looks reasonable, as do rental rates. (b) The first thing to note is the correlations with y. The older the property, the lower the rent. The higher the expenses, the higher the rent. Vacancy rate has a positive correlation, but it is very weak. Square footage has a positive correlation with rent. There are also correlations among the predictor variables, especially between age and expenses (positive, size and expenses (positive), and expenses and vacancy rate (negative). (c) $\hat{y} = 12.2$ -0.142 `x1` $+0.282$ `x2` $+0.619$ `x3` $+0.00000792$ `x4`. (d) $H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$ versus H_1 : at least one $\beta_j \neq 0$. $P = 7.27 \times 10^{-14} < .05$ so reject H_0. At least one predictor is related to rental rate. (e) $R^2 = .5847$. (f) $H_0 : \beta_j = 0$ versus $H_1 : \beta_j \neq 0$. `x1`, `x2`, and `x4` all have $P < .05$. `x3` has $P = .57$ and we cannot reject the null or conclude that vacancy rate has an effect on the rental rate. (g) Property 1: 15.15, 12.85, 17.44; Property 2: 15.54, 13.25, 17.84; Property 3: 16.91, 14.53, 19.29.

	Property 1	Property 2	Property 3
x1	4	6	12
x2	10	11.5	12.5
x3	0.1	0	0.32
x4	80,000	120,000	340,000

Predict the rental rate and compute separate prediction intervals for the rental rates using 95% confidence. **Briefly tell what the prediction interval tells you.** Hint:

```
> newx = data.frame(x1=c(4,6,12), x2=c(10,11.5,12.5), x3=c(.1,0,.32),
  x4=10000*c(8,12,34))
> predict(fit, newx, interval="prediction")
```

2. In a small-scale experimental study of the relation between degree of brand liking (`y`) and the moisture content (`moisture`) and sweetness (`sweetness`) of the product, the results in the data below were obtained from the experiment based on a completely randomized design.[3]

```
brand = data.frame(
  liking=c(64,73,61,76,72,80,71,83,83,89,86,93,88,95,94,100),
  moisture=c(4,4,4,4,6,6,6,6,8,8,8,8,10,10,10,10),
  sweetness=c(2,4,2,4,2,4,2,4,2,4,2,4,2,4,2,4) )
```

   (a) Read the brand data set into R and run basic descriptive statistics (counts, mins, maxs, means). Do the descriptives make sense?

   (b) Produce a scatterplot matrix and correlation matrix. Discuss the relationships between the variables.

   (c) Regress liking on the two predictor variables. State the estimated regression equation.

   (d) What fraction of variation in liking is explained by these predictor variables?

   (e) Test whether the overall model is significant. State the null and alternative, P-value and decision.

   (f) If the overall regression model is significant in the previous part then test whether each of the individual regression coefficients equals 0 at the 5% level against a 2-sided alternative, i.e., $H_0 : \beta_j = 0$ for $j = 1, 2$. For each predictor, state the P-value and your decision.

   (g) Assume that the regression model you have estimated is appropriate. Predict the liking when `moisture=5` and `sweetness=4`. Find a prediction interval and separately a confidence interval for the estimated mean value using 99% confidence.

[3](a) They make sense. All variables are positive. (b) Moisture has a stronger positive correlation with liking than sweetness. There is no correlation between sweetness and moisture because the data are from an experiment with an orthogonal design. (c) $\hat{y} = 37.650 + 4.425$`moisture` $+ 4.375$`sweetness`. (d) $R^2 = .9521$. (e) $H_0 : \beta_1 = \beta_2 = 0$ versus H_1 : at least one $\beta_j \neq 0$. $P = 2.658 \times 10^{-9} < .05$ so reject H_0. At least one predictor is related to liking. (f) $H_0 : \beta_j = 0$ versus H_1 : $\beta_j \neq 0$. `moisture` and `sweetness` have $P < .05$. (g) Use `predict(fit, data.frame(moisture=5, sweetness=4), interval="conf", level=.99)` to find 77.275, 73.88, 80.67.

3. JWHT problem 9(a)–(c) on page 122. Find the correlation and scatterplot matrices and regress mpg on all other variables except for name. Hint: when finding correlations see the `use="pair"` option. Answer these questions.[4]

   (a) Based on the scatterplots, comment on the relationships between the predictors and mpg.

   (b) What is the correlation between mpg and displacement and what does it tell you?

   (c) Is there a statistically significant relationship between the predictors and the response?

   (d) Which predictors appear to have a statistically significant relationship to the response?

   (e) What does the slope coefficient for the year variable suggest?

   (f) What does the slope coefficient for the displacement variable suggest?

[4]Solution: (a) There are nonlinear relationships between mpg and displacement, horsepower, weight and acceleration. There may be other nonlinear relationships. (b) $r = -.7763$, so larger displacement is associated with smaller mpg. (c) Yes, $P < 2.2 \times 10^{-16}$. (d) Displacement, weight, year and origin. (e) $b = .75$ suggests that high gas milage improves over time. (f) $b = 0.0199$ suggests that larger displacement is associated with higher gas milage. This contradicts part b because of multicollinearity. It is an example of a sign flip.

Assumptions, Diagnostics, Remedies

$$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \epsilon_i, \quad \mathbb{E}(\epsilon_i) = 0, \mathbb{V}(\epsilon_i) = \sigma^2, \epsilon_i \text{ normal, uncorrelated}$$

Assumption	**Diagnostic**	**Problem**	**Remedies**
Linear function	Residual plot	Biased $\hat{y}$	Transform x
$\mathbb{V}(e_i)$ constant	Residual plot	OLS not BLUE	Transform y or GLM
Correlated errors	Autocorrelations Plot $\hat{\epsilon}_t$ vs. t Runs and DW tests	Estimates unbiased Variances wrong	Transform: Cochrane-Orcutt
Errors normal	QQ plot	t/F assume it	Transform y or GLM
Outliers†	Leverage plot	Unstable estimates	
Collinearity†	Correlations/VIFs	Inflated variances	Discuss next week

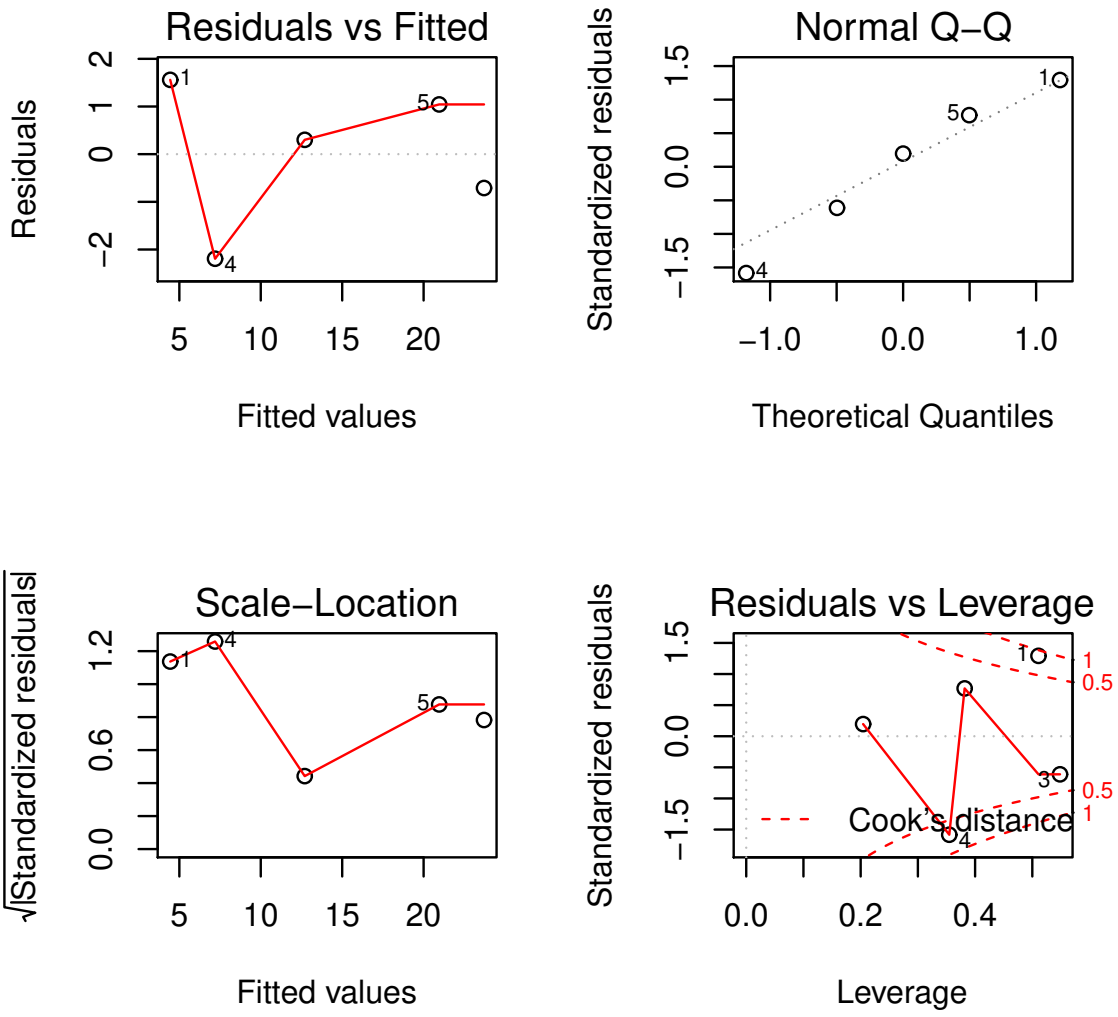
† = Not an assumption but it can cause problems

- To avoid omitted variables, start with conceptual framework
- Fixing functional form is usually my next priority
- Constant $\mathbb{V}(\epsilon_i)$: having efficient (BLUE) estimates is not critical when sample sizes are large, but I still worry about this
- Whenever observations come from different time periods (time series) you should suspect correlated errors
- Normal errors: with even modest sample sizes this is a lower priority because of the central limit theorem
- Outliers: Determine why the value is an outlier
  - Erroneous value: fix it or drop it
  - Correct but extreme value: many considerations. Transform amounts/counts (log or root) will reduce influence, or there are also robust versions of regression

Residual and QQ Plots in R

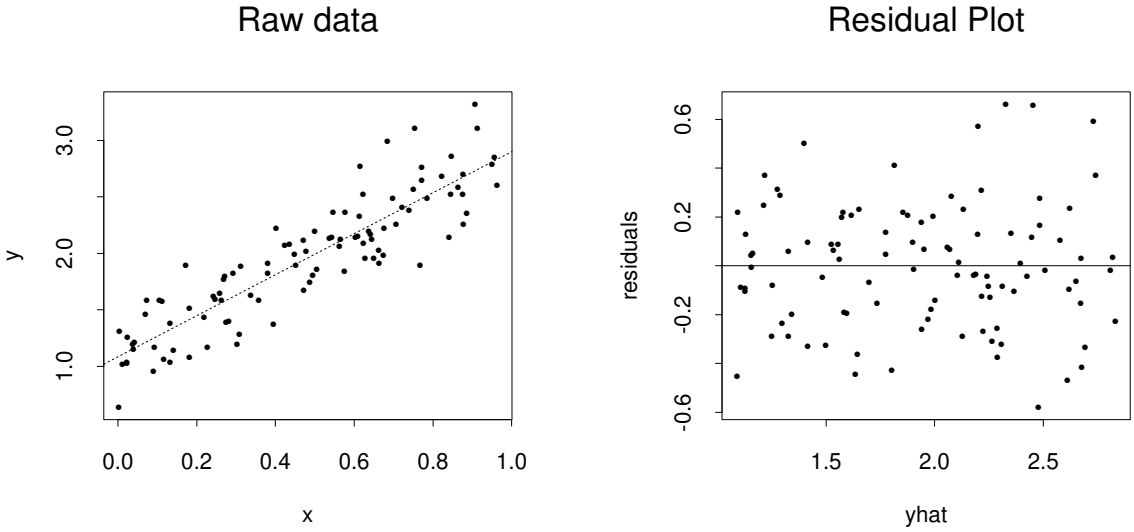
- `plot(fit)` gives diagnostic plots of `lm` objects.
- Alternatively, `plot(fit, which=1)` gives residuals `plot(fit, which=2)` gives QQ plots, etc. See `?plot.lm`

```
> machine = data.frame(age=c(2,5,9,3,8), cost=c(6,13,23,5,22))
> fit = lm(cost ~ age, machine)
> par(mfrow=c(2,2)) # show 2*2 grid of plots. Use c(1,1) for one plot per page
> plot(fit)
```



Ideal Regression Model

True model: $y = 1 + 2x + \epsilon$



Least-squares fit: $\hat{y} = 1.09 + 1.81x$

- Definition of **residuals**

$$\hat{\epsilon} = y - \hat{y}$$

- Residual plots ($\hat{\epsilon}$ against $\hat{y}$) help us to understand how well the model fits the data. Use `plot(fit, which=1)` in R.
- Here, residuals do not follow any pattern ("snow storm")
- Variance of residuals does not depend on $\hat{y}$ (homoscedasticity)
- If residual plot shows a pattern then you model is misspecified and you need to fix it!

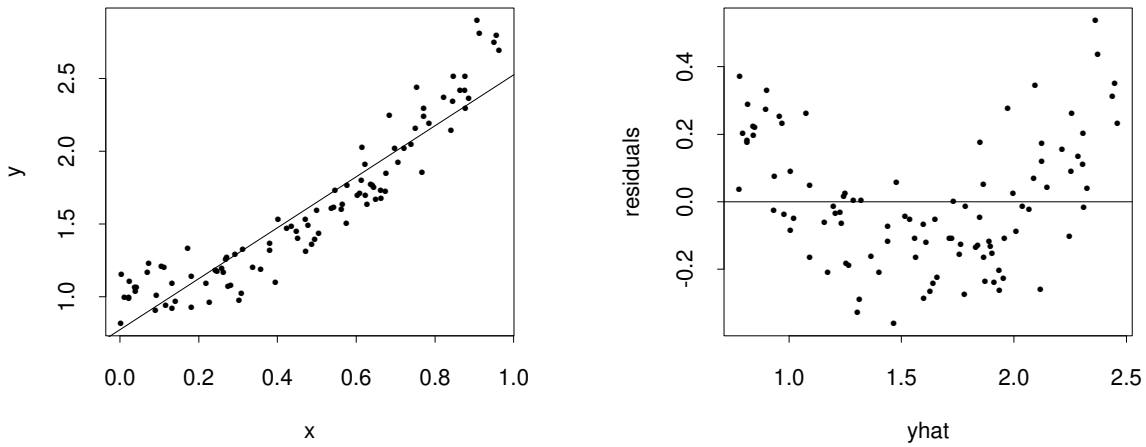
Model misspecification I

- Suppose the true model is

$$y = 1 + 2x^2 + \epsilon$$

- We estimate the *incorrect* (misspecified) model:

$$y = \beta_0 + \beta_1 x + \epsilon$$



- True relationship is nonlinear (curved)
- Fitted line does not describe the data well
- Pattern in residual plot indicates model misspecification
- But, variance of residuals does *not* depend on $\hat{y}$ (error variance still homoscedastic)

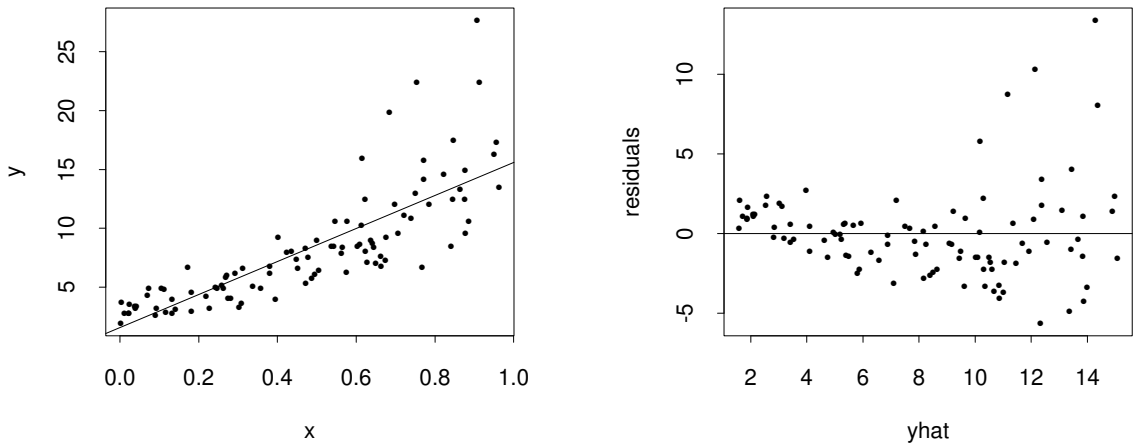
Model misspecification II

- Suppose the true model is

$$y = \exp(1 + 2x + \epsilon)$$

- We estimate the *incorrect* (misspecified) model:

$$y = \beta_0 + \beta_1 x + \epsilon$$



- Residuals show a pattern: first they are mostly positive, then mostly negative, then roughly centered at 0.
- Variance of the residuals increases with $\hat{y}$ indicating heteroscedasticity
- Note that

$$\log(y) = 1 + 2x + \epsilon$$

Review of Key Results

- Multiple linear regression model (p predictors with intercept):

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \mathbb{E}(\boldsymbol{\epsilon}) = \mathbf{0}, \quad \text{and} \quad \mathbb{V}(\boldsymbol{\epsilon}) = \sigma^2\mathbf{I} = \mathbb{V}(\mathbf{y})$$

- We estimate $\boldsymbol{\beta}$ with OLS estimates $\hat{\beta}$, and $\boldsymbol{\epsilon}$ with $\hat{\boldsymbol{\epsilon}}$

$$\hat{\beta} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} \quad \text{and} \quad S_\epsilon^2 = \sum \hat{\epsilon}_i^2/(n-p-1)$$

- Estimates of predicted values and residuals are given by

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{H}\mathbf{y} \quad \text{and} \quad \hat{\boldsymbol{\epsilon}} = \mathbf{y} - \mathbf{X}\hat{\beta}$$

  where $\mathbf{H} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}$ is the **hat matrix**

- Theorem: $\mathbb{V}(\hat{\mathbf{y}}) = \sigma^2\mathbf{H}$ and $\mathbb{V}(\hat{\boldsymbol{\epsilon}}) = \sigma^2(\mathbf{I} - \mathbf{H})$

- Definition: the **leverage** of observation i is h_{ii}, where

$$0 \leq h_{ii} \leq 1 \quad \text{and} \quad \sum_{i=1}^{n} h_{ii} = p + 1$$

  As a rule of thumb, leverages greater than twice their average, i.e., $h_{ii} > 2(p+1)/n$, are considered large

- For simple linear regression

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{n}(x_j - \bar{x})^2}$$

Standardized Residuals

- Problem 1: estimated residuals $\hat{\boldsymbol{\epsilon}}$ don't have constant variance (even if model, which assumes homoscedastic errors, is true).
- The **standardized residual**, sometimes called the **studentized residual** is obtained by dividing $\hat{e}$ by its estimated standard deviation

$$r_i = \frac{\hat{\epsilon}_i}{S_\epsilon \sqrt{1 - h_{ii}}}$$

  Also sometimes called **internal standardized residuals**.
- Problem 2: if there are outliers, S is inflated, which deflates all r_i. One solution is to omit observation i and reestimate the model giving prediction $\hat{y}_{(i)}$ and MSE $S^2_{(i)}$.
- The **deleted** (or **external**) residual and **studentized deleted/external residual** are

$$d_i = y_i - \hat{y}_{(i)} \quad \text{and} \quad \frac{d_i}{S_{(i)}\sqrt{1 - h_{ii}}}$$

- Cook's distance

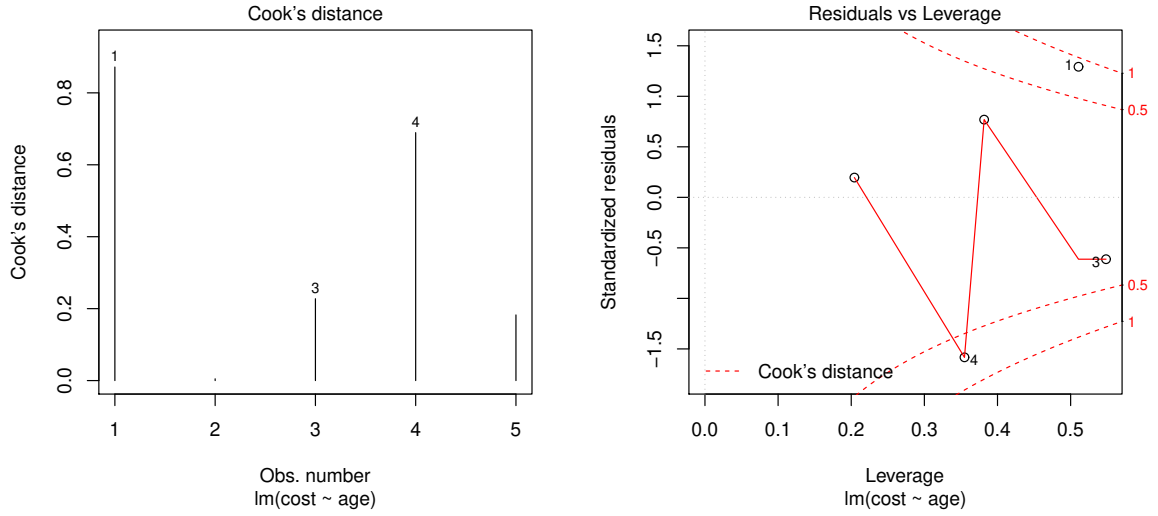
$$D_i = \frac{\sum_j (\hat{y}_j - \hat{y}_{j(i)})^2}{p S_e^2}$$

  As a rule of thumb, **values greater than 1 are considered large.**

Leverage in R: Machine Example

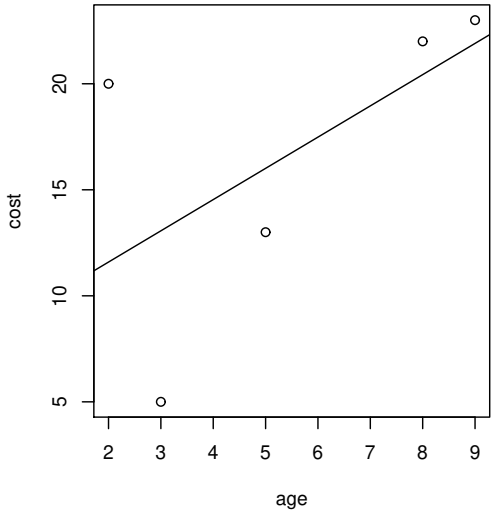
```
> fit = lm(cost~age, machine)
> plot(fit, which=c(4,5))

> # use lm.influence function to get leverages in R
> lm.influence(fit)$hat
        1         2         3         4         5
0.5107527 0.2043011 0.5483871 0.3548387 0.3817204
> sum(lm.influence(fit)$hat)
[1] 2

> # check our work with matrix inversion
> X=cbind(1,machine$age)
> diag(X %*% solve(t(X) %*% X) %*% t(X))
[1] 0.5107527 0.2043011 0.5483871 0.3548387 0.3817204

> # check our work with simple formula
> xbar = mean(machine$age)
> 1/5+(machine$age-xbar)^2/sum((machine$age-xbar)^2)
[1] 0.5107527 0.2043011 0.5483871 0.3548387 0.3817204
```

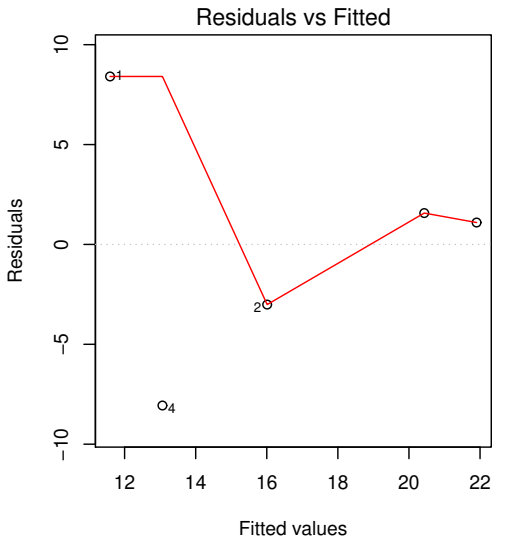
None are considered large since all are less than $2(2/5) = 0.8$

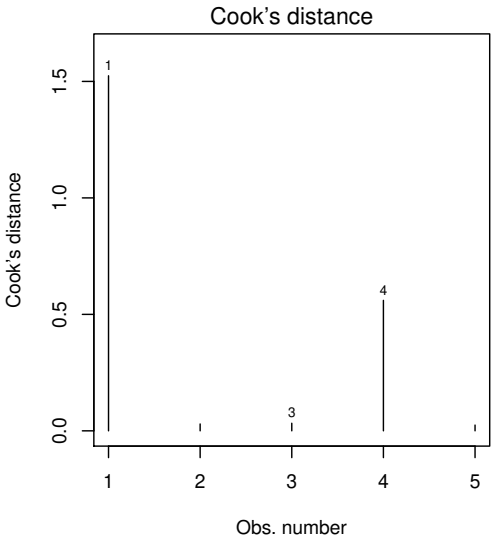


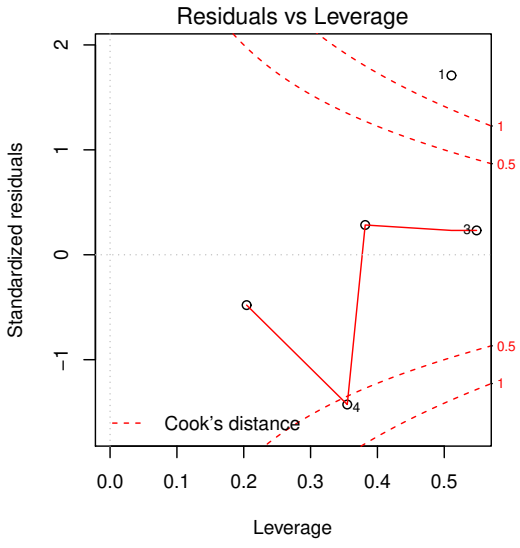
Modified Machine Example

```
> machine2 = data.frame(age=c(2,5,9,3,8), cost=c(20,13,23,5,22)) # change y1=20 from 6
> fit = lm(cost~age, machine2)
> plot(machine2); abline(fit)
> plot(fit)
```









Your Turn

For each of the problems below, examine the residual, Q-Q and Cook's distance plots. Are there problems? What are you looking for and why?

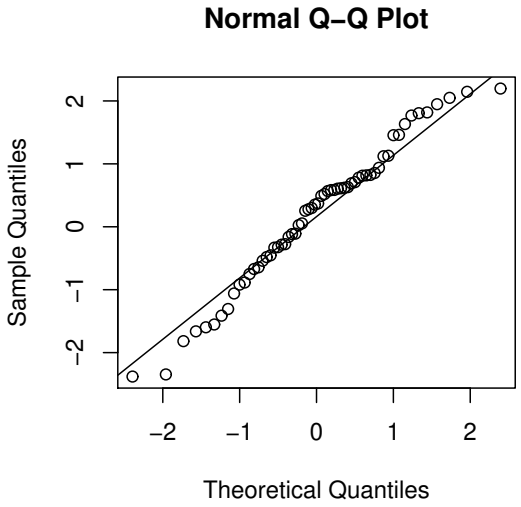
1. Problem 2 on page 17.
2. Problem 3 on page 17.
3. Problem 4 on page 17.
4. Commercial property problem on page 42.
5. Brand problem on page 42.

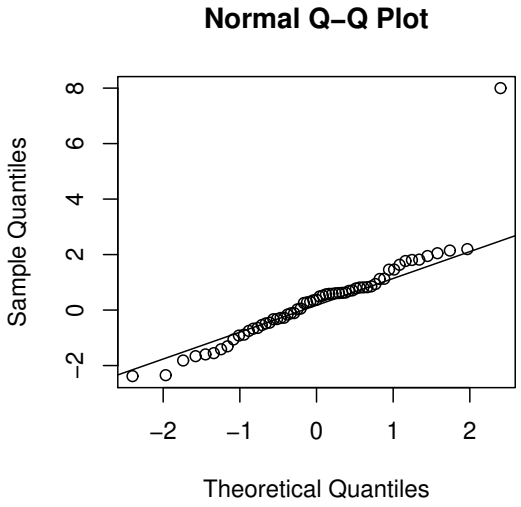
Answers

1. No problems.
2. WSJ has a Cook's distance greater than 1 suggesting it is influential. WSJ and USA deviate from normality.
3. You don't have many cases, but the residual plot has an unusual shape. Perhaps there is heteroscedasticity, but it looks like the residuals are larger for small fitted values, which is not what one usually sees with amount variables (variance usually increases with the mean). There's some non-normality and case 1 has Cook's distance greater than 1.
4. The residual plot shows no patterns.
5. There may be an inverted-U shaped relationship in the residual plot, but the sample is tiny and the fit is not bad.

Evaluating Normality

- *When the CLT has not "converged"* then the population distribution of residuals must be normal for you to use the t distribution.
- Evaluate normality using a *normal probability plot*, which plots the observed quantile against normal quantiles ("Q-Q plot").
- **Points falling on a line indicate normality.**

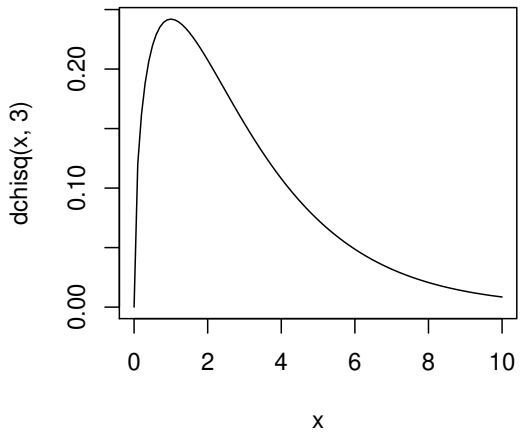




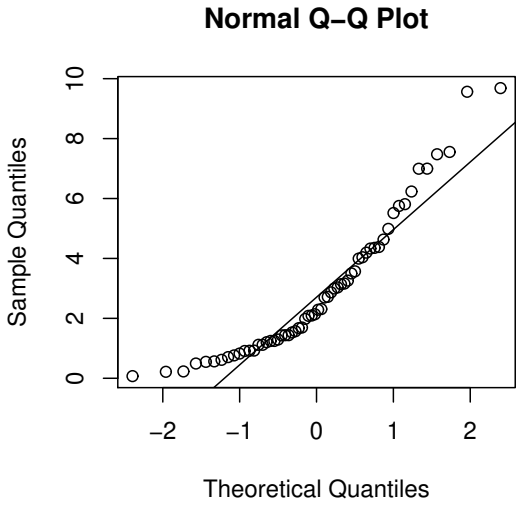

```
> set.seed(12345)
> z = rnorm(60)

> # data from normal distribution
> qqnorm(z); qqline(z)

> # now we add an outlier
> qqnorm(c(z, 8)); qqline(c(z,8))
```

Evaluating Normality

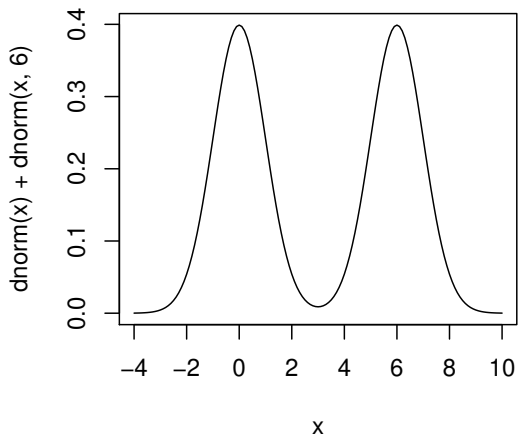
```
> x = seq(0, 10, .1)
> plot(x, dchisq(x, 3), type="l")
```



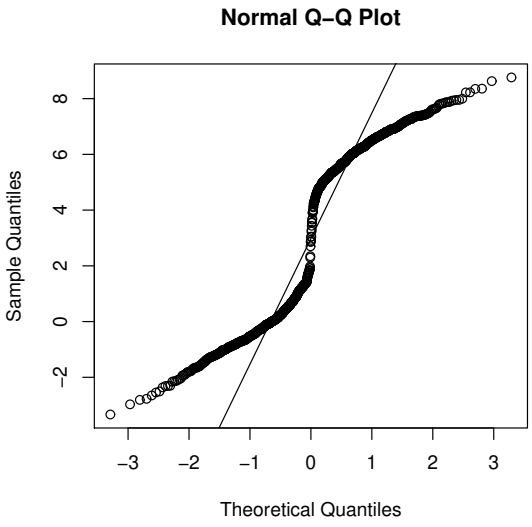

```
> rx = rchisq(60, 3)
> qqnorm(rx); qqline(rx)
```




```
> x = seq(-4, 10, .1)
> plot(x, dnorm(x)+dnorm(x,6), type="l")
```




```
> x = c(rnorm(100),rnorm(100,6))
> qqnorm(x); qqline(x)
```



Transformations

- Transformations change the functional relationship between dependent and predictor variables
- Two reasons for transformations:
  - Heteroscedasticity / non-symmetric error distributions ($\mathbb{E}(\epsilon_i) = 0$ but Skewness$(\epsilon_i) \neq 0$): transform the *dependent* variable
  - Underlying relationship nonlinear: Transform the *predictor* variable(s)
- Outline of transformation lecture
  1. Transformations of dependent variable
  2. Identifying nonlinear relationships
     (a) Scatterplots
     (b) Compare R^2 values for models using various transformations (e.g., Tukey's ladder of re-expressions)
  3. Other transformations based on combinations of variables

Heteroscedasticity

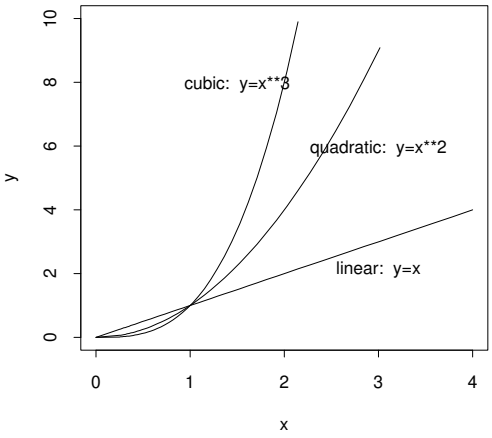
- Slide 33 gave model assumptions, including homoscedasticity
- If a model has heteroscedastic error variance, the least-squares estimates will still be unbiased, but will not be BLUE.
- Modeling heteroscedastic data: Use ...
  - a variance-stabilizing transformation. For count and amount dependent variables, use the logarithm or square root as a variance-stabilizing transformation. Take **logs** when the **standard deviation** of the errors is proportional to the mean, and **square roots** when the **variance** is proportional to the mean.
  - a different model, e.g., Poisson or logistic regression
  - **weighted least squares** (WLS): let $w_i \propto 1/\mathbb{V}(y_i)$ and $\mathbf{W} = \text{diag}(w_1, \ldots, w_n)$. Then the WLS estimate is BLUE:

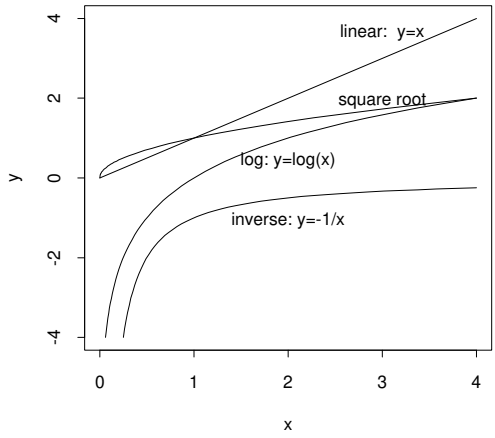
    $$\widehat{\boldsymbol{\beta}}_{\text{WLS}} = (\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{y}.$$

    When $\mathbf{W}$ is unknown (usually), we use *iteratively reweighted least squares* (IRLS)

Tukey's Ladder of Transformations

Consider models of the form $y = \beta x^k$ over $[0, \infty)$





Note: "Returns" refer to the first derivative (slope)

k	Function	Slope	Nature of "Returns"
3	$y = \beta x^3$	$dy/dx = 3\beta x^2$	Increasing
2	$y = \beta x^2$	$dy/dx = 2\beta x^1$	Increasing
1	$y = \beta x^1$	$dy/dx = \beta x^0$	Constant
$1/2$	$y = \beta\sqrt{x}$	$dy/dx = \beta x^{-1/2}/2$	Decreasing
0	$y = \beta \log x$	$dy/dx = \beta x^{-1}$	Decreasing
-1	$y = \beta x^{-1}$	$dy/dx = -\beta x^{-2}$	Decreasing

- For $k < 1$ the slope approaches 0, but never changes sign
- You may need to shift variables, e.g, $\log(x + 1)$ or $1/(x + 1)$

Tukey's First Aid Re-Expressions[5]

"Choosing exactly the right re-expression for a particular quantity may not be easy. To try to do a good job, we may have to (1) sense rather weak indications from the data in hand, (2) draw on experience with other bodies of data, or (3) lean on subject-matter knowledge. Even all three may not suffice. Both because we may not be prepared to try hard to choose our re-expression, or because we have too little information for anyone to choose reliably, we need rules of thumb that can provide "first aid," that can lead us to re-expressions that are almost always not bad — and usually pretty good.

Four rules will deal quite effectively with most of our needs, namely:

1. Take logs of an amount or count (if there are zeros or infinities, we may need to deal with them; see the next section)
2. Take logits or folded logs of fractions or percents; use some multiple of

$$\log\left(\frac{p}{1-p}\right)$$

...

These rules are not supposed to be a final answer—just as first aid for the injured is no substitute for a physician—but they offer a safe beginning."

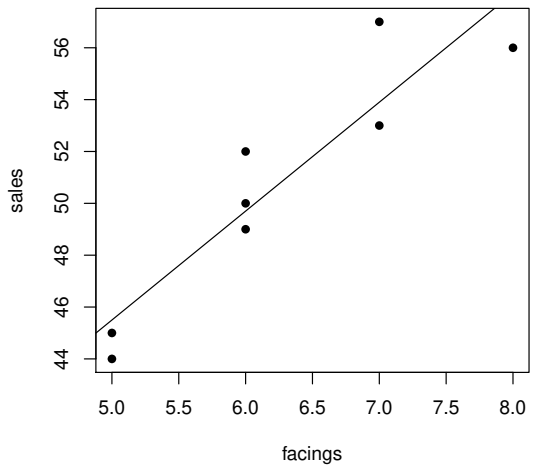
Also see discussion of "Tukey's ladder of re-expressions" in Tukey (1977), *EDA*, pp. 90–1.

[5] Mosteller and Tukey (1977), *Data Analysis and Regression*, p. 109

Cereal Problem

A cereal manufacturer believes that there is an association between cereal sales and the number of facings the cereal has on each stores' shelves. Eight stores were surveyed to test this hypothesis.

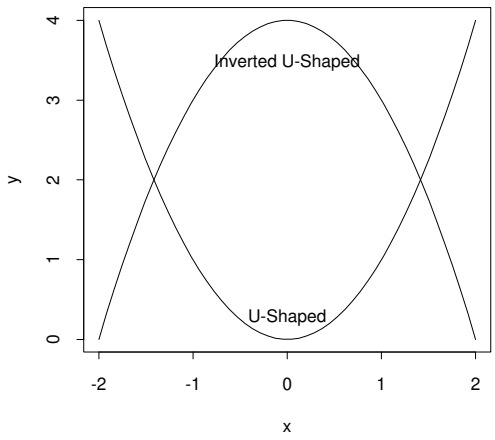
```
> dat = data.frame(facings=c(5,6,6,7,5,7,6,8), sales = c(45,50,52,53,44,57,49,56))
> plot(dat, pch=16)
> fit = lm(sales ~ facings, data = dat)
> abline(fit)
> summary(fit)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   24.500      4.541   5.395  0.00167 **
facings        4.200      0.718   5.849  0.00110 **

Residual standard error: 1.966 on 6 degrees of freedom
Multiple R-squared:  0.8508,Adjusted R-squared:  0.8259
F-statistic: 34.22 on 1 and 6 DF,  p-value: 0.001102
```



How do you like this model? (Video solution)

Polynomial Transformations

Consider models the form $y = \beta_0 + \beta_1 x + \beta_2 x^2$



- The min/max value occurs at $x_{\text{opt}} = -\beta_1/2\beta_2$
- U-shaped when $\beta_2 > 0$
- Higher-order polynomials can also be used, e.g., cubic $\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$, but they are problematic
  - Curvature can change several times—few theories postulate this.
  - x, x^2, x^3 highly correlated—mean center or standardize x's before fitting model.
- Interpretation complicated—don't interpret estimates of individual terms (except sign of β_2). Use F-tests to gauge significance and plots to interpret effects.

Purification case

The "techies" (scientists) in the laboratory have been lobbying you, and management in general, to include just one more laboratory step. They think it's a good idea, although you have some doubt because one of them is known to be good friends with the founder of the start-up biotechnology company that makes the reagent used in the reaction. But if adding this step works as expected, it could help immensely in reducing production costs. The trouble is, the test results just came back and they don't look so good. Discussion at the upcoming meeting between the technical staff and management will be spirited, so you've decided to take a look at the data.

Your firm is anticipating government approval from the Food and Drug Administration (FDA) to market a new medical diagnostic test made possible by monoclonal antibody technology, and you are part of the team in charge of production. Naturally, the team has been investigating ways to increase production yields or lower costs.

The proposed improvement is to insert yet another reaction as an intermediate purifying procedure. This is good because it focuses resources down the line on the particular product you want to produce. But it shares the problem of any additional step in the laboratory: one more manipulation, one more intervention, one more way for something to go wrong. In this particular case, it has been suggested that, while small amounts of the reagent may be helpful, trying to purify too well will actually decrease the yield and increase costs.

The design of the test was to have a series of test production runs, each with a different amount of purifier, including one test run with the purification step omitted entirely (i.e., 0 purifier). The order of the tests was randomized so that any time trends would not be mistakenly interpreted as being due to purification.

```
purify = data.frame(x = 0:10,
  y=c(13.39,11.86,27.93,35.83,28.52,41.21,37.07,51.07,51.69,31.37,21.26))
```

1. Regress `yield` on `amount`. Is the regression significant? Based on this test alone, do you recommend including a purifying step in the process?
2. Generate a scatterplot of `yield` against `amount`. Comment.
3. Modify your regression model as appropriate. Based on the revised analysis, do you recommend including a purifying step in the process?

Video solution

Multiplicative Models

- Multiplicative models have the form

$$y = \beta_0 x_1^{\beta_1} x_2^{\beta_2} \epsilon$$

- Taking natural logs of both sides we get

$$\log y = \log \beta_0 + \beta_1 \log x_1 + \beta_2 \log x_2 + \log \epsilon,$$

  which has the form of a multiple linear regression model, regressing $\log y$ on $\log x_1$ and $\log x_2$. We estimate it assuming $\log \epsilon$ is normal and $\mathbb{V}(\log \epsilon) = \sigma^2$ constant.

- Predictions: we estimate (untransformed) y with

$$\exp(b_0 + b_1 \log x_1 + b_2 \log x_2 + S_\epsilon^2/2)$$

  Some modelers drop the $S_\epsilon^2/2$ term, in which case you are estimating the median rather than mean.

- The coefficient of variation of e (rather than the variance) is

$$\frac{\sqrt{\mathbb{V}(y)}}{\mathbb{E}(y)} = \sqrt{\exp(\sigma^2) - 1},$$

  which doesn't depend on i (and is thus constant).

- Even without the assumption that ϵ is log normal, we will see that if $\sigma_y \propto \mu_y$ then logging y "stabilizes" the error variance, making it constant. Logging y is called a **variance stabilizing transformation**. (see Tamhane books)

Interpreting β_1

$$y = \beta_0 x^{\beta_1} \qquad \frac{dy}{dx} = \beta_0 \beta_1 x^{\beta_1 - 1} \qquad \frac{d^2y}{dx^2} = \beta_0 \beta_1 (\beta_1 - 1) x^{\beta_1 - 2}$$

- Interpretation (ignore error term; assume $\beta_0 > 0$ and $x > 0$)
  - $\beta_1 = 1 \Longrightarrow y \propto x$ (linear, proportional returns)
  - $0 < \beta_1 < 1 \Longrightarrow$ changes in y *decrease* as x increases (concave downward)
  - $\beta_1 > 1 \Longrightarrow$ changes in y *increase* as x increases (concave upward)
- In economic applications, β_j is the **elasticity** of y with respect to x_j—the expected percentage change in y of a 1% change in x_j, all else being equal. Let dx be an infinitesimal change in x and dy be the corresponding change in y. Then the elasticity is

$$\frac{dy/y}{dx/x} = \frac{dy}{dx} \cdot \frac{x}{y} = \beta_0 \beta_1 x^{\beta_1 - 1} \cdot \frac{x}{\beta_0 x^{\beta_1}} = \beta_1$$

- We estimate β_1 by logging both sides. The base of the logarithm does not matter, but natural logs are usually used.

Background: Lognormal Distribution

- If y has a normal distribution with mean μ and variance σ^2, then $y^* = \exp(y)$ has a **log-normal distribution**, i.e., $\log y^* = y$ is normal.
- Theorem: the mean and variance of y^* are

$$\mathbb{E}(y^*) = \exp(\mu + \sigma^2/2)$$

$$\mathbb{V}(y^*) = \exp(2\mu + \sigma^2)(\exp(\sigma^2) - 1)$$

- The coefficient of variation of y^* is

$$\begin{aligned} \frac{\sqrt{\mathbb{V}(y^*)}}{\mathbb{E}(y^*)} &= \frac{\sqrt{\exp(2\mu + \sigma^2)(\exp(\sigma^2) - 1)}}{\exp(\mu + \sigma^2/2)} \\ &= \sqrt{\exp(\sigma^2) - 1}, \end{aligned}$$

  which does not depend on μ

Business Failure Case

Consider the slightly scary topic of business failures. This problem analyzes data from each state on the number of failed businesses and the population in thousands for each of the 50 states and the District of Columbia (51 observations).

```
busfail = data.frame(
  row.names=c("AL","AK","AZ","AR","CA","CO","CT","DE","DC","FL","GA","HI","ID",
    "IL","IN","IA","KS","KY","LA","ME","MD","MA","MI","MN","MS","MO","MT","NE",
    "NV","NH","NJ","NM","NY","NC","ND","OH","OK","OR","PA","RI","SC","SD","TN",
    "TX","UT","VT","VA","WA","WV","WI","WY"),
  pop=c(4187,599,3936,2424,31211,3566,3277,700,578,13679,6917,1172,1099,11697,
    5713,2814,2531,3789,4295,1239,4965,6012,9478,4517,2643,5234,839,1607,1389,
    1125,7879,1616,18197,6945,635,11091,3231,3032,12048,1000,3643,715,5099,
    18031,1860,576,6491,5255,1820,5038,470),
  fail=c(841,108,2064,186,19695,1542,1093,137,200,5088,2350,305,350,2094,1091,507,
    1069,841,664,383,1540,2720,2546,921,322,1230,173,399,568,617,2843,448,6916,1194,
    145,2127,1440,969,3124,344,392,175,1209,7096,351,173,1738,2025,315,1224,90)
)
```

1. Make a scatterplot of business failures against population and superimpose a regression line. Describe the relationship. Comment on whether the linear model appears to hold.
2. Make a scatterplot of the log of business failures against the log of population. Superimpose a regression line. Does the linear model hold better with the logged data?
3. Regress the log of failures on the log of population. State estimated regression equation.
4. Test at the 5% level to see whether there is a significant relationship between the logs of failure and population. Explain.
5. Test whether the population slope for the logs is significantly different from 1 or not. What does this tell you?
6. Illinois had 2,094 failures with a population of 11,697 (thousand). Find the predicted log failures for Illinois.
7. Estimate the unlogged failures and identify Illinois on your scatterplot. (Hint: use `identify` function in R.)

Your Turn

1. The table below shows the level of investment and the results obtained by the important players in fiber-optics cable for long-distance communications.

```
fiber = data.frame(invest=c(1300,500,130,2000,1200,110,40,60,57,500,90,90),
  miles=c(1700,650,110,1200,2400,165,72,45,85,650,50,87))
```

   (a) Find the regression equation predicting circuit miles from investment.

   (b) Draw the scatterplot and residuals. Discuss whether the linear model holds.

   (c) Examine Cook's distance. Are there influential observations, as indicated by having Cook's distance greater than 1?

   (d) Regress the log of circuit miles on the log of investment. State the estimated equation.

   (e) Draw the scatterplot and residuals for the log model. Discuss whether the linear model holds.

   (f) Do firms that spend more achieve significantly more circuit miles? State the null and alternative, P-value and decision using the 5% level of significance.

   (g) Test at the 5% level whether the coefficient for log(investment) is different from 1, which indicates that investment is proportional to miles. What do values not equal to 1 indicate in terms of economies of scale?

   (h) Predict log(miles) from an investment of $1000.

   (i) Predict unlogged miles from an investment of $1000.

2. A research analyst for an oil company wants to develop a model to predict miles per gallon based on highway speed. An experiment is designed in which a test car is driven at speeds ranging from 10 miles per hour to 75 miles per hour in increments of 5 MPH. Two replicates were observed for each speed:

```
speed = data.frame(mph=rep(seq(10,75,by=5), 2),
  mpg=c(4.8,8.6,9.8,13.7,18.2,19.9,22.4,21.3,20.5,18.6,14.4,12.1,10.1,8.4,
    5.7,7.3,11.2,12.4,16.8,19,23.5,22,19.7,19.3,13.7,13,9.4,7.6))
```

   (a) Make a scatterplot of MPG against MPH. Based on the plot, do you suggest transforming the data? If, so which transformation do you suggest?

   (b) Regress MPG on MPH (do not include transformations yet). Report (i) the estimated regression equation, (ii) the P-value testing the overall significance of the model, and (iii) a residual plot of residuals versus predicted values.

   (c) Add appropriate transformations to your model. Report (i) the estimated regression equation, (ii) the P-value testing the overall significance of the model, and (iii) a residual plot with comments about the fit of your model.

(d) Using the model you developed in the previous part, estimate gas milage for a car traveling at 62 miles per hour.

(e) At what speed is gas milage maximized?

(f) Produce a scatterplot with the two fits superimposed (linear, quadratic).

3. Use the auto data set from JWHT problem 3.9 on page 136.

(a) The `origin` variable is categorical, where 1=US, 2=Europe and 3=Japan. We'll cover dummies next week, but for now type the following command to make it a factor variable and assign meaningful labels:

```
auto$origin = factor(auto$origin, 1:3, c("US", "Europe", "Japan"))
```

(b) Regress `mpg` on `origin`, `weight` and `year`. Examine the diagnostic plots and comment on which assumptions of the linear model, if any, are violated.

(c) Regress `log(mpg)` on `origin`, `log(weight)`, `year` and `year` squared. Examine the diagnostic plots and the summary. For you to think about but not turn in: why would year have this effect for year?

(d) Describe the effect of year on log(mpg), i.e., is it U-shaped, inverted-U shaped, or linear? If it is nonlinear, where is the minimum or maximum. Draw a graph showing the effect.

(e) What does the coefficient for log(weight) tell you?

Answers

1. Fiber problem. (a) miles(hat) = 101.813 + 0.986 invest; (b) The linear model does not hold because the variance of the residuals increases with the mean of miles. (c) Yes, observation 4 is influential, and 5 is nearly "influential." (d) log(miles) = 0.06803 + 1.00735 log(invest). (e) The residuals have more constant variance and no obvious pattern. (f) $H_0 : \beta = 0$ versus $H_1 : \beta \neq 0$, $P = 1.02 \times 10^{-6}/2 < .05$, so we reject H_0. (g) $H_0 : \beta = 1$ versus $H_1 : \beta \neq 1$. A 95% confidence interval for the slope is $1 \in [0.79, 1.22]$. Since 1 is in this interval, we cannot reject the null hypothesis that the slope is 1. It is plausible that miles are proportional to investment. Recall that investment $= e^a \times \text{invest}^b$. (h) predict(fit, data.frame(invest=1000)) = 7.026552 (i) exp(7.026552 + 0.1938/2), where 0.1938 is the MSE from the ANOVA table.

2. Speed problem. (a) The scatterplot shows an inverted-U shaped relationship, but no heteroscedasticity. We should add a quadratic term for speed. (b) mpg = 12.75 + 0.039 speed. $P = 0.468 > 5\%$ so we cannot reject $H_0 : \beta = 0$. The residuals show a strong pattern indicating that the model is misspecified. (c) Add a quadratic term: $\texttt{mpg} = -7.56 + 1.27\texttt{speed} - 0.0145\texttt{speed}^2$. $P = 2.338 \times 10^{-14} < 5\%$, so we reject $H_0 : \beta_1 = \beta_2 = 0$. The residuals are not perfect, but the magnitude of the pattern is reduced. It looks like MPG increases linearly until about 40MPH. The quadratic model provides a substantially better fit ($R^2 = .9188$) than the linear model ($R^2 = .02039$), but the the quadratic model could be improved further. (e) `predict(fit, data.frame(speed=62))` = 15.54618 (g) $-1.27/(2 \times -0.0145) = 43.8$ MPH

```
fit = lm(mpg~mph, speed)
fit2 = lm(mpg~mph+I(mph^2), speed)
# not part of exercise, but interesting
fit3 = lm(mpg~as.factor(mph), speed)
plot(speed); abline(fit)
lines(10:75, predict(fit2, data.frame(mph=10:75)), col=2)
lines(speed$mph[1:14], fit3$fit[1:14], type="b", col=4, pch=16)
```

3. JWHT 3.9 (b) The residual plot shows a pattern, with consistently positive values, then negative, then positive again. This indicates that the fit can be improved. (d) The quadratic coefficient 0.0019 > 0 indicating a U shape, with a min value at $0.2559684/(2 \times 0.0019051) = 67.17978$, i.e., around 1967. To make a graph, note that year ranges from 70 to 82 and see code below to see the effect. (e) The coefficient is negative, which indicates that as the weight of the car increases the milage decreases.

```
fit = lm(mpg ~ weight + year, auto) # part b
plot(fit, which=1) # part b
fit = lm(log(mpg) ~ log(weight) + year + I(year^2)
  + origin, data=auto) # part c
summary(fit) # part c
drop1(fit, test="F") # part c
x = 70:82  # part d
plot(x, -0.255968375*x +  0.001905147*x^2, type="l")
```

Newfood case

Mr. Conrad Ulcer, newly appointed New Products Marketing Director for Concorn Kitchens, was considering the possibility of marketing a new highly nutritional food product with widely varied uses. This product could be used as a snack, a camping food, or as a diet food. The product was to be generically labeled Newfood.

Because of this wide range of possible uses, the company had great difficulty in defining the market. The product was viewed as having no direct competitors. Early product and concept tests were very encouraging. These tests led Mr. Ulcer to believe that the product could easily sell 2 million cases (24 packages in a case) under the proposed marketing proposal involving a 24-cent package price and an advertising program involving $3 million in expenditures per year. There were no capital expenditures required to go national, since manufacturing was to be done on a contract-pack basis.

Because there was considerable uncertainty among Concorn Management as to either probable first-year and subsequent-year sales, or the best introductory campaign, Ulcer decided that a six-month market test would be conducted. The objectives of the test were to:

- Better estimate first-year sales.
- Study certain marketing variables to determine an optimal — or at least better — introductory plan.
- Estimate the long-run potential of the product

These objectives were accomplished through the controlled introduction of the product into four markets. Conditions were experimentally varied within the grocery stores in each of the four markets. Sales were measured with a store audit of a panel of stores. Preliminary results had been obtained. Now it was up to Mr. Ulcer to understand their implications on the introduction of Newfood.

Design of Experimental Study

The three variables included in the experimental design were price, advertising expenditures, and location of the product within the store. Three prices were tested (24 cents, 29 cents, and 34 cents), two levels of advertising (a simulation of a $3 million introduction and a $6 million plan), and two locations (placing the product in the bread section versus the instant breakfast section). Prices and location were varied across stores within cities while advertising was varied across cities. The advertising was all in the form of TV spots. The levels were selected so that they would stimulate on a local basis the impact that could be achieved from national introduction programs at the $3 million and $6 million expenditure levels. Due to differential costs between markets and differential costs between spot and network (to be used in national introduction), an attempt was made to equate (and measure) advertising inputs of gross advertising impressions generated, normalized for

market size. Unfortunately, it was not possible to achieve exactly the desired levels. This was due to the problem of non-availabilities of spots in some markets and discrepancies between estimates of TV audiences made at the time the test was being planned and the actual audiences reached at the time the commercials were actually run.

In the selection of cities and stores for the tests, attempts were made to match stores on such variables as store size, number of checkout counters, and characteristics of the trading area. Because it was not certain that adequate matches had been achieved, Ulcer decided to obtain measurements on some of these variables for possible use in adjusting for differences in cell characteristics. He also felt that it might be possible to learn something about the relationships between these variables and sales, and that this information would be of assistance in planning the product introduction into other markets.

```
newfood = data.frame(
  sales=c(225,323,424,268,224,331,254,492,167,226,210,289,204,288,245,161,161,
    246,128,154,163,151,180,150),
  price=c(24,24,24,24,24,24,24,24,29,29,29,29,29,29,29,29,34,34,34,34,34,34,34,34),
  ad=c(0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1),
  loc=c(0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1),
  income=c(7.3,8.3,6.9,6.5,7.3,8.3,6.9,6.5,6.5,8.4,6.5,6.2,6.5,8.4,6.5,6.2,
    7.2,8.1,6.6,6.1,7.2,8.1,6.6,6.1),
  volume=c(34,41,32,28,34,41,23,37,33,39,30,27,37,43,30,19,32,42,29,24,32,36,29,24),
  city=c(3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2))
```

Questions for class discussion

1. Compute the correlation matrix. How do you explain the 0 correlations (e.g., between location and advertising)?

2. Run a regression of `sales` (the first two months sale) on `price` alone. Next, on `price` and `ad`. Finally on `price`, `ad`, and `loc`. Thus, you will have three regressions. What happens to the coefficients of `price` in the three regressions? What happens to the coefficients of `ad` in the two regressions? Explain.

3. Run a regression of `sales` against `price`, `ad`, `loc`, and `volume`. What happens to the coefficients of `price`, `ad`, and `loc` which you found in the third regression in question 2 above? Which coefficient changes the most with the introduction of store size? Why does this happen?

4. Finally, run a regression of `sales` against `price`, `ad`, `loc`, `volume`, and `income`. What changes do you observe between these results and that of the fourth regression? Explain.

5. What additional regression runs, if any, should be made to complete the analysis of these data?

Effects of Model Misspecification: Omitting Relevant Predictor

---

- Suppose we fit the model

$$y = \beta_0 + x\beta_1 + \epsilon$$

- But the true model is

$$y = \beta_0 + x\beta_1 + z\beta_2 + \epsilon$$

- Theorem: (**Omitted variable bias**) If x and z are correlated, b_1 will be biased, with the direction of the bias depending on the sign of the correlation between x and z and the sign of β_2.

$$\mathbb{E}(b_1) = \beta_1 + \beta_2 r_{xz} \frac{S_z}{S_x}$$

where S_x and S_z are the sample standard deviations of x and z

Direction of Bias in b_1

Sign of correlation between x and z	if $\beta_2 > 0$	if $\beta_2 < 0$
$\text{corr}(x, z) = r_{xz} > 0$	Upward bias	Downward bias
$\text{corr}(x, z) = r_{xz} = 0$	No change	No change
$\text{corr}(x, z) = r_{xz} < 0$	Downward bias	Upward bias

- Note that if x and z are uncorrelated (orthogonal design), we can add or drop variables without changing the other coefficients.

1. Multicollinearity Definition and Effects

- Definition: **Multicollinearity** is when the predictor variables are highly correlated with one another (Note: Y is not mentioned here). When more than one X's move together, it is difficult to sort out their separate effects.
- What are some effects? You cannot sort out what is doing what.
  - *Unstable coefficients.* High estimated standard errors for one or more slope coefficients.
    - Implication: Low t-ratio, so sometimes we cannot reject the null hypothesis that $\beta = 0$. At the extreme, the model as a whole may be significant, while none of the individual slope parameters are!
    - Coefficients can change wildly when variables are added or dropped from the model.
  - *Incorrect signs.* Slope estimates can have signs that are not consistent with intuition.
- Note: predicted values not affected directly

Predicted Values Unaffected

- Suppose the true model is

$$y = 2x_1 + x_2$$

- We have $n = 3$ observations:

x_1	x_2	y	e
1	2	4	0
2	4	8	0
3	6	12	0

  Note that $x_2 = 2x_1$, so that the two columns are perfectly correlated

- The true model fits perfectly, but so do many others, e.g.,

$$y = 4x_1 + 0x_2 \Longrightarrow x_2 \text{ no effect}$$

$$y = 0x_1 + 2x_2 \Longrightarrow x_1 \text{ no effect}$$

$$y = 6x_1 - x_2 \Longrightarrow x_2 \text{ negative effect}$$

  Predictions correct for all these choices of parameter estimates, but substantive interpretation completely different!

- Which (b_1, b_2) is correct? We can't tell from these data.

- What happens if we use our fitted model to *extrapolate*? e.g., estimate y for $x_1 = 1$ and $x_2 = 1$. The correct answer is 3, but the other models gives estimates 4, 2, and 5, respectively. **Extrapolation is especially questionable when using a model estimated from multicollinear data.**

2. Detecting Multicollinearity

- Compute a correlation matrix of the predictor variables and possibly a scatterplot matrix. Large correlations indicate multicollinearity could be a problem.
- Unstable coefficients or incorrect signs
- **Tolerance** is $1 - R^2$ from this regression (fraction of variance *unexplained* by the model)
- **Variance inflation factor** (VIF) is 1/tolerance.

```
> install.packages("car")    # do this only once
> library(car)    # you must first download it from CRAN
> fit = lm(sales~price+ad+loc+volume+income, newfood)
> vif(fit)
   price       ad      loc   volume   income
1.079882 2.697664 1.005447 3.447143 3.367158


> # where does VIF for ad come from?
> summary(lm(ad ~ price + loc+volume+income, newfood))

...
Multiple R-squared:  0.6293,Adjusted R-squared:  0.5513

> 1/(1-.6293)
[1] 2.697599
```

- Interpreation
  - VIF=1: no multicollinearity
  - $2 <$ VIF < 5 beware
  - $5 <$ VIF < 10 substantial multicollinearity
  - VIF ≥ 10 severe multicollinearity

Living with Multicollinearity

1. If the objective is primarily to make good *predictions*, then you could do nothing, although you may be better off using stepwise, ridge/lasso, or principal components regression (PCR). Data mining applications often fall into this category.
2. If the objective is to *interpret* regression coefficients, then
   (a) If possible, avoid multicollinearity with an *orthogonal* design, where predictor variables are uncorrelated
   (b) Understand why you have multicollinearity
      - **Pipe**, e.g., $x_1 \to x_2 \to y$. If you are studying $x_1 \to y$ then do not control for x_2 because it blocks path.
      - Predictors manifestations of common, underlying **latent construct**, e.g., $w \to x_1$ and $w \to x_2$. Often, estimate w and use it instead of x_1 and x_2.
      - "Back-door" confound (**fork**). Include control to block back-door path, e.g., if $w \to y$ and $w \to x \to y$ then control for w to study $x \to y$.
      - We usually[6] do not control for **colliders**, e.g., if $x \to w$ and $y \to w$, then do not control for collider w when studying $x \to y$.

[6]There can be complicated situations where we must have a collider as a control, but then we will have to add other control(s) to fix the problems the collider creates. See Pearl et al., Causal Inference in Statistics—A Primer.

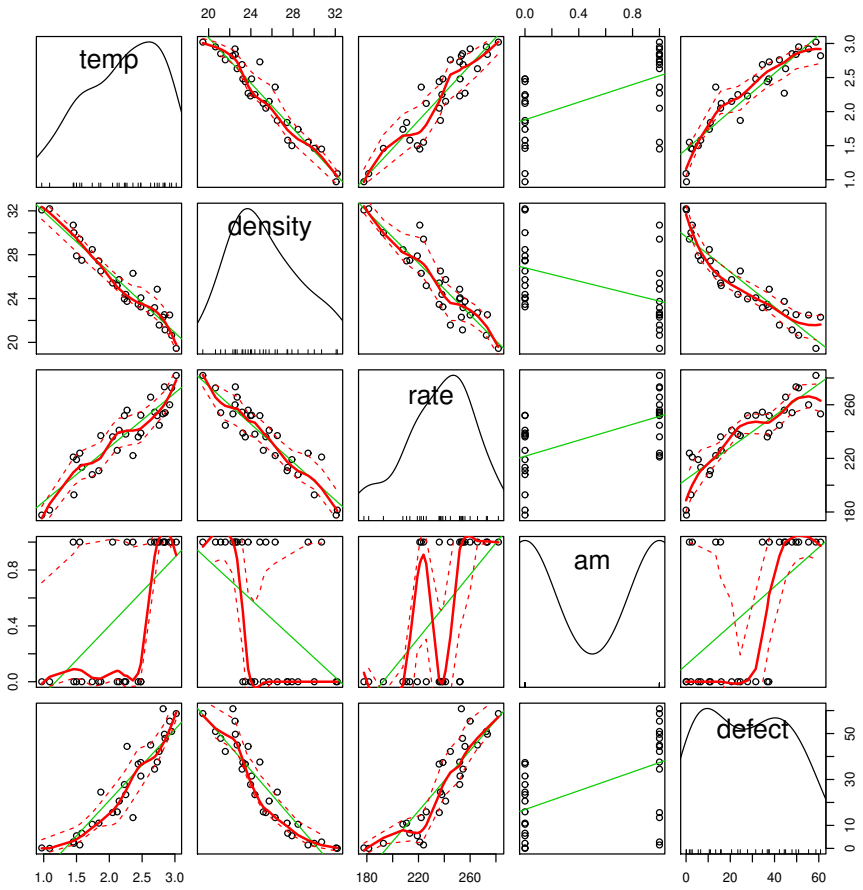
Model Specification Issues

- For successful use of multiple regression, sufficient knowledge *about the subject domain* is required to identify relevant predictor variables and their functional relationship with the dependent variable.
- If the only goal is predictive, it should make sense for a variable to be in the model, e.g., does it make sense to have customer ID number as a predictor variable (it's probably a proxy for something else, e.g., tenure)?
- If the goal is confirmatory, **begin with a conceptual framework** (directed aclyclic graph or DAG). Include a variable if
  - It is a decision variable
  - The variable helps to control for important causal factors (forks), e.g., seasonality or competitive actions. Remember the omitted variable bias theorem.
  - Unless you are really sure of your framework, do robustness checks by adding/dropping variables.

Scatterplots with CAR

The car package offers improved scatterplot matrices:

```
> library(car) # do this if you haven't ready done so
> scatterplotMatrix(~temp+density+rate+am+defect, quality)
```



- The red lines are *smoothers*, which trace the middle of the distribution of the vertical variable conditional on the horizontal variable. Green lines are robust regression lines.
- The ticks are called a *rug* and show individual observations.
- The graphs on the diagonal are density estimates (like a histogram).

Quality Control Case

Everybody seems to disagree about just why so many parts have to be fixed or thrown away after they are produced. Some say that it's the temperature of the production process, which needs to be held constant (within a reasonable range). Others claim that it's clearly the density of the product, and that if we could only produce a heavier material, the problems would disappear. Then there is Ole, who has been warning everyone forever to take care not to push the equipment beyond its limits. This problem would be the easiest to fix, simply by slowing down the production rate; however, this would increase costs. Interestingly, many of the workers on the morning shift think that the problem is "those inexperienced workers in the afternoon," who, curiously, feel the same way about the morning workers.

Ever since the factory was automated, with computer network communication and bar code readers at each station, data have been piling up. You've finally decided to have a look. After your assistant aggregated the data by 4-hour blocks and then typed in the AM/PM variable, you found the following description of the variables:

- `temperature`: measures the temperature variability as a standard deviation during the time of measurement
- `density`: indicates the density of the final product
- `rate`: rate of production
- `am`: 1 indicates morning and 0 afternoon
- `defect`: average number of defects per 1000 produced

Discussion Questions

1. Generate a scatterplot matrix and a correlation matrix. Interpret the correlations. What "obvious conclusions" can you draw?
2. Run a multiple regression predicting defect rate from the other four variables. Is the overall model significant? Which predictors, if any, are significant? Compute and interpret variance inflation factors. What "obvious conclusions" can you draw?
3. Predict defect from each of the predictor variables separately, e.g., `defect` from `temp`, `defect` from `density`, `defect` from `rate`, etc. Which of the predictors are significant in the simple linear regressions?
4. Perform further analysis as needed. What action do you recommend? Why? Hint: think about the causal relationships between the variables.
5. To compare the two shifts, would it be appropriate to perform an independent-sample t-test (i.e., H_0 : AM defect rate = PM defect rate)? How is this different from the multiple regression approach? Which is preferred? Discuss.

6. How would you present your findings to a client?

```
quality = data.frame(
  temp=c(.97,2.85,2.95,2.84,1.84,2.05,1.5,2.48,2.23,3.02,2.69,2.63,1.58,2.48,2.25,
    2.76,2.36,1.09,2.15,2.12,2.27,2.73,1.46,1.55,2.92,2.44,1.87,1.45,2.82,1.74),
  density=c(32.08,21.14,20.65,22.53,27.43,25.42,27.89,23.24,23.97,19.45,23.17,
    22.7,27.49,24.07,24.38,21.58,26.3,32.19,25.73,25.18,23.74,24.85,30.01,
    29.42,22.5,23.47,26.51,30.7,22.3,28.47),
  rate=c(177.7,254.1,272.6,273.4,210.8,236.1,219.1,238.9,251.9,281.9,254.5,265.7,
    213.3,252.2,238.1,244.7,222.1,181.4,241,226,256,251.9,192.8,223.9,260.0,236,
    237.0,221,253.2,207.9),
  am=c(0,1,1,1,0,1,0,0,0,1,1,1,0,0,0,1,1,0,0,0,1,1,0,1,1,0,0,1,1,0),
  defect=c(.2,47.9,50.9,49.7,11,15.6,5.5,37.4,27.8,58.7,34.5,45,6.6,31.5,23.4,
    42.2,13.4,0,20.6,15.9,44.4,37.6,2.2,1.5,55.4,36.7,24.5,2.8,60.8,10.5)
)
```

Your Turn

1. Use the auto data set from JWHT problem 3.9 on page 122. Type the following:

```
auto = read.table("auto.txt", header=T) # auto.txt on Canvas
auto$origin = factor(auto$origin, 1:3, c("US", "Europe", "Japan"))
```

   The data set is also in the ISLR library. Since you are changing the origin variable it might be best not to touch the ISLR file.

   (a) Regress `mpg` on `cylinders`, `displacement`, `weight`, and `year`. Comment on the signs of the estimated coefficients and note which are significantly different from 0. What is value of R^2?

   (b) Compute the variance inflation factors. What do they tell you?

   (c) Drop `weight` from the model. What happens to the parameter estimates and R^2?

   (d) Drop `weight` and `displacement` from the model. What happens to the parameter estimates and R^2?

2. JWHT problem 3.14a–f on page 125. For part (c)–(e), are the parameters "covered" by the 95% confidence intervals?

Answers

1. JWHT 3.9. (a) It is odd that displacement has a positive sign, although it is not significant. Only weight and year are significant. $R^2 = .8091$. (b) The VIF values for all variables but year are large, indicating multicollinearity: cars that weigh more tend to have larger engine displacement and more cylinders. (c) The coefficients change drastically. Cylinders goes from $-.29$ to $-.62$. Displacement goes from 0.00497 to -0.0415 and now has a positive, significant slope. Year changes less. R^2 decreases a little bit to 0.7423. The other variables do the much of the explaining previously done by weight. (d) Cylinders is now significant, and R^2 reduces only slightly to .7135.

```
# part a
Call: lm(mpg ~ cylinders + displacement + weight + year)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -14.076941   4.055159  -3.471 0.000575 ***
cylinders     -0.289589   0.329225  -0.880 0.379611
displacement   0.004973   0.006701   0.742 0.458425
weight        -0.006702   0.000572 -11.717  < 2e-16 ***
year           0.764751   0.050684  15.089  < 2e-16 ***

Residual standard error: 3.436 on 392 degrees of freedom
Multiple R-squared:  0.8091,Adjusted R-squared:  0.8072
F-statistic: 415.5 on 4 and 392 DF,  p-value: < 2.2e-16

# part b
   cylinders displacement       weight         year
   10.524432    16.406259     7.888061     1.173000

# part c
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -18.199719   4.688296  -3.882 0.000122 ***
cylinders     -0.620910   0.380657  -1.631 0.103658
displacement  -0.041545   0.006265  -6.632  1.1e-10 ***
year           0.699324   0.058461  11.962  < 2e-16 ***

Residual standard error: 3.988 on 393 degrees of freedom
Multiple R-squared:  0.7423

# part d
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.30285    4.93534  -3.506 0.000507 ***
cylinders    -3.00405    0.13223 -22.718  < 2e-16 ***
year          0.75289    0.06098  12.347  < 2e-16 ***

Multiple R-squared:  0.7135
```

2. JWHT problem 3.14. (a) $\beta_0 = \beta_1 = 2$, $\beta_2 = 0.3$, $\sigma = 1$, $y = 2 + 2x_1 + 0.3x_2 + e$. (b) `cor(x1,x2)` $= 0.83512$. (c) Barely reject $H_0 : \beta_1 = 0$ ($P = .0487$), but not $H_0 : \beta_2 = 0$ ($P = .3754$). (d) Reject. (e) Reject. (f) This illustrates the omitted variable bias. The predictors are highly correlated and can serve as proxies for one another. The β_2 coefficient is not significant in part c because of high standard errors, but is significant in part e because it explains some of the variation due to x_1. (g) $\beta_1 = 2$ is in both intervals, but $\beta_2 = .3$ is only the interval for the first one. This is because of the omitted variable bias. The estimate of β_1 is also biased, but not enough to cause the interval not to cover the parameter.

Measuring model performance and variable importance

- This lecture establishes tools used for automated machine learning variable selection
- The *full model* is $y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$.
- The variation left unexplained by the full model is given by the *residual sum of squares* or *deviance*:

$$\text{SSE} = \sum_{i=1}^{n} \hat{\epsilon}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- When $\beta_1 = \cdots = \beta_p = 0$ we call it the *intercept* or *null* model, and it turns out $\hat{y}_i = \bar{y}$, the mean of y. The variation left unexplained by the null model is the *total sum of squares*:

$$\text{SST} = \sum_{i=1}^{n} (y_i - \bar{y})^2 = (n-1)S_y^2$$

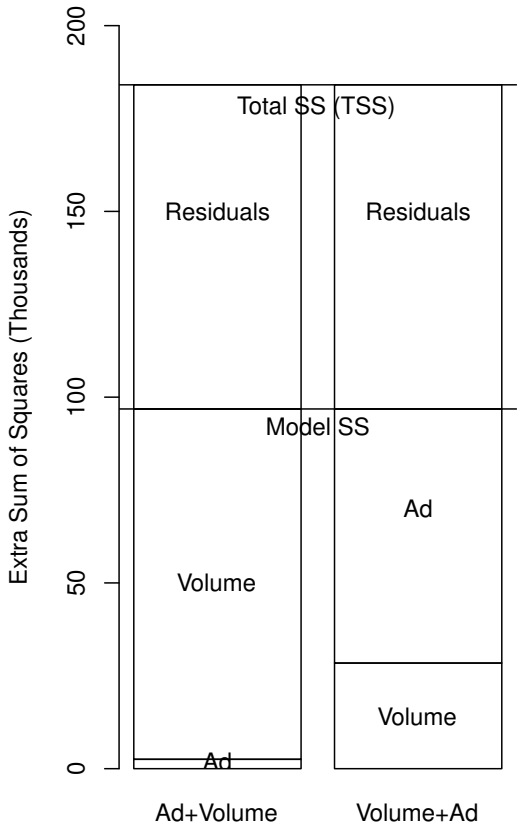
- Usually the full model explains more than the null model

```
> deviance(lm(sales~1, newfood))  # null model leaves 184K unexplained
[1] 184066

> deviance(lm(sales~ad, newfood))  # model with ads leaves 182K unexplained
[1] 181544.5

> deviance(lm(sales~ad+volume, newfood)) # model with both leaves 87K unexplained
[1] 87246.89
```

- We can say that `ad` explains $184066 - 181544.5 = 2,521.5$.
- `ad` and `volume` together explain $184066 - 87247 = 96,819$.

The anova and drop1 commands

```
> attach(newfood)
> var(sales)*(nrow(newfood)-1) # TSS
[1] 184066
> sum((sales-mean(sales))^2) # TSS
[1] 184066

> fit = lm(sales ~ ad + volume, newfood)
> anova(fit)  # extra SS
Analysis of Variance Table
Response: sales
          Df Sum Sq Mean Sq F value Pr(>F)
ad         1   2521    2521  0.6069 0.4446
volume     1  94298   94298 22.6971 0.0001
Residuals 21  87247    4155

> drop1(fit)  # partial SS
Single term deletions
Model: sales ~ ad + volume
       Df Sum of Sq    RSS
<none>               87247
ad      1     68391 155638
volume  1     94298 181544


> fit2 = lm(sales ~ volume+ad, newfood)

> anova(fit2)  # extra SS
Analysis of Variance Table
Response: sales
          Df Sum Sq Mean Sq F value Pr(>F)
volume     1  28428   28428  6.8426 0.0161
ad         1  68391   68391 16.4614 0.0006
Residuals 21  87247    4155

> drop1(fit2)  # partial SS
Single term deletions
Model: sales ~ volume + ad
       Df Sum of Sq    RSS
<none>               87247
volume  1     94298 181544
ad      1     68391 155638
```



- TSS = RSS for intercept model = sum of extra SS
- Extra SS (`anova`): change in SS if term added
- Partial SS (`drop1`): change in SS if term dropped
- For last term in, Extra SS = Partial SS

The F test of "overall significance"

- Recall how to test the following hypothesis:

  $H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$

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

```
> summary(fit)
Call: lm(formula = sales ~ ad + volume, data = newfood)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -324.31     116.95  -2.773 0.011396 *
ad            159.25      39.25   4.057 0.000567 ***
volume         14.87       3.12   4.764 0.000105 ***

Residual standard error: 64.46 on 21 degrees of freedom
Multiple R-squared:  0.526,Adjusted R-squared:  0.4809
F-statistic: 11.65 on 2 and 21 DF,  p-value: 0.0003941
```

- This test compares the full model (H_1) with the null model

$$F = \frac{\frac{\text{SST}-\text{SSE}}{p}}{\frac{\text{SSE}}{n-p-1}} = \frac{\frac{184066-87246.89}{2}}{\frac{87246.89}{21}} = 11.652 = \left(\frac{\frac{\Delta\text{SSE}}{\Delta df}}{S_e^2}\right)$$

- The P-value can be found in R:

```
> 1 - pf(11.652, 2, 21)
[1] 0.0003941411
```

- This foreshadows an important application of the F test

The F test for a single predictor

- Now consider testing $H_0 : \beta_2 = 0$ versus $H_1 : \beta_2 \neq 0$
- Assuming the null is true ($\beta_2 = 0$) we get $y = \beta_0 + \beta_1 x_1 + e$, which will be called the *reduced* model
- The `summary` output give a t test and shows $P = .000105$.
- We can equivalently perform an F test, which will have more general uses later in the course:

```
> anova(fit)
Analysis of Variance Table
Response: sales
          Df Sum Sq Mean Sq F value    Pr(>F)
ad         1   2521    2521  0.6069 0.4446424
volume     1  94298   94298 22.6971 0.0001048 ***
Residuals 21  87247    4155

> drop1(fit, test="F")
Single term deletions
Model: sales ~ ad + volume
       Df Sum of Sq    RSS F value    Pr(>F)
<none>               87247
ad      1     68391 155638  16.461 0.0005666 ***
volume  1     94298 181544  22.697 0.0001048 ***
```

- The following F has 1, 21 df

$$F = \frac{\frac{\Delta \text{SSE}}{\Delta df}}{S_e^2} = \frac{\frac{94298}{1}}{\frac{87247}{21}} = \frac{\frac{94298}{1}}{4155} = 22.6971$$

```
> 1-pf(22.6971, 1, 21)
[1] 0.0001047984
```

- Why is `ad` not significant in the `anova` output?

Summary of key points

- Model selection involves picking a model between the null and full model (as well as picking transformations, causal relatinoships, etc.)
- We use SSE to measure what is unexplained by a model
- The `anova` command generates *extra sum of squares*, telling how much SSE is reduced as we add terms to a model one at a time. They depend on the order of the terms.
- The `drop1` command generates *partial sum of squares*, telling how much SSE is increased when we drop each term. They do not depend on order.
- The F test allows for hypothesis testing between the full and reduced models
- Stepwise regression and trees (CART, random forests, etc.) use these ideas to automate variable selection.

Your Turn

1. Consider the click ballpoint pens data given on page 9.

   (a) How much variation is left unexplained by the intercept model? (this will be called the *null deviance*)

   (b) How much variation is explained by adding `ad` to the intercept model?

   (c) How much additional variation is explained by adding `reps` to a model that already has `ad` in it?

   (d) How much additional variation is explained by adding `eff` to a model that already has `ad` and `reps` in it?

   (e) How much variation is unexplained by a model having all three predictors?

   (f) How much less variation is explained if we drop `ad` from a model with all three predictors in it?

   (g) Compute R^2 for the three-predictor model "by hand" using only the numbers you have found above. Confirm your answer by having R compute it.

   (h) Compute adjusted R^2 by hand and confirm it.

   (i) Compute the F statistic for the overall test of significance by hand.

   (j) Compute the F statistic to test $H_0 : \beta_1 = 0$ by hand.

2. Consider the commercial properties data.

   (a) Obtain the analysis of variance table that decomposes the regression sum of squares into extra sums of squares associated with X_4; with X_1 given X_4; with X_2 given X_1 and X_4; and with X_3 given X_1, X_2 and X_4. Hint use the `lm` and `anova` functions.

   (b) Test whether X_3 can be dropped from the regression model given that X_1, X_2 and X_4 are retained. Use the F test statistic and level of significance of .01. State the null and alternative hypotheses, test statistic, P-value and decision.

   (c) Test whether both X_2 and X_3 can be dropped from the regression model given that X_1 and X_4 are retained; use $\alpha = 0.01$. State the null and alternative hypotheses, test statistic, P-value and decision. Hint: use the `pf` function to find P values.

   (d) Find the variance inflation factors for the full model with all four predictors in the model. What do they tell you?

3. Consider the brand problem.

   (a) Find the variance inflation factors for the full model with both predictors in the model. What do they tell you?

   (b) Obtain the analysis of variance table that decomposes the regression sum of squares into extra sums of squares associated with `moisture`; and with `sweetness` given `moisture`?

(c) Obtain the analysis of variance table that decomposes the regression sum of squares into extra sums of squares associated with `sweetness`; and `moisture` given `sweetness`. What do you notice?

(d) Regress liking on moisture content only. How does the estimate of β_1 in the previous part compare with the estimate in the model with both predictors?

4. Consider the quality control data set discussed in class.

(a) How much variation is left unexplained by the intercept model? (this will be called the *null deviance*)

(b) How much variation is explained by adding `rate` to the intercept model?

(c) How much additional variation is explained by adding `am` to a model that already has `rate` in it?

(d) How much variation is unexplained by a model having both predictors?

(e) How much less variation is explained if we drop `rate` from a model with both predictors in it?

(f) Compute R^2 for the two-predictor model "by hand" using only the numbers you have found above. Confirm your answer by having R compute it.

(g) Compute the F statistic for the overall test of significance by hand.

(h) Using the two-variable model, compute the F statistic to test $H_0 : \beta_1 = 0$ by hand (where β_1 is for rate) (hint: it is in the `drop1` output).

5. Return to the model you estimated for the Auto problem of the Your Turn on page 68. Fit this model,

```
auto = read.table("auto.txt", header=T) # auto.txt on Canvas
auto$origin = factor(auto$origin, 1:3, c("US", "Europe", "Japan"))
fit = lm(log(mpg)~ log(weight)+year+I(year^2)+origin, auto)
```

(a) Interpret the effect of origin on log(mpg). Which origin has the best gas milage? Worst? Rank them in order of gas milage from least to greatest.

(b) After controlling for the other variables, what is the difference in log(mpg) between Japan and Europe, on the average?

(c) Is there a significant difference between the log gas milage for US and Japan after controlling for the other variables?

(d) You should see that there are two dummy variables for the origin variable. If origin were dropped from the model (i.e., the two dummies were set equal to 0), by how much would RSS increase?

(e) Can you reject the null hypothesis that both origin dummies are 0, so that none of the origin levels have different effects?

(f) As an extra challenge, perform an F test for whether `year` and its quadratic effect can both be dropped from the model, i.e., test $H_0 : \beta_2 = \beta_3 = 0$. You can do this by fitting the reduced model (call it `fit2`) then use the `anova(fit2, fit)` command or include `poly(year, 2, raw=T)` and use `drop1`.

Answers

1. (a) SST = 598253; (b) 463451, Hint: `fit = lm(sales ~ ad + reps + eff, click)` and then `anova(fit0)`; (c) 59327; (d) 4431; (e) 71044; (f) 44295, Hint: `drop1(fit, test="F")`; (g) $1 - 71044/598253 = .8812$, Hint: `summary(fit)`; (h) $1 - (71044/36)/(598253/39) = .8714$; (i) $((598253 - 71044)/3)/(71044/36) = 89.05$; (j) $(44295/1)/(71044/36) = 22.45$, Hint: see `drop1` output.

2. (a) Hint: `fit = lm(y ~ x4+x1+x2+x3, comm)` then `anova(fit)`; (b) $H_0 : \beta_3 = 0$ versus $H_1 : \beta_3 \neq 0$, $P = 0.5704$, we cannot reject H_0. (c) $H_0 : \beta_2 = \beta_3 = 0$ versus $H_1 : \beta_2 \neq 0$ or $\beta_3 \neq 0$. From the output below the P-value is less than 0 and so we reject H_0.

```
> fit2 = lm(y ~ x1+x4, comm)
> 1-pf(((deviance(fit2)-deviance(fit))/2) / (deviance(fit)/76), 2, 76)
[1] 6.682136e-05
```

   (d) Hint: `vif(fit)`. The VIF values are 1.41, 1.24, 1.65, and 1.32. All are fairly close to 1 indicating that multicollinearity is not a serious problem.

3. Brand problem. (a) The VIFs are both 1 indicating uncorrelated predictors; (b) The extra SS for moisture are 1566 and for sweetness 306; (c) They are the same as in the previous part; (d) The coefficient is the same.

4. Quality control problem. (a) 10929.29. (b) 8566.9. (c) 7.1. (d) 2355.3. (e) 5440.5. (f) $1 - 2355.3/10929.29 = 0.7844965$. (g) $((10929.29 - 2355.3)/2)/(2355.3/27) = 49.14$. (h) $(5440.5/1)/(2355.3/27) = 62.37$.

5. Auto problem. (a) The base is US. Europe had 0.0668 higher log(mpg) than the US on average, and Japan had 0.032 higher log(mpg) than the US, on average. Europe had the best gas milage, followed by Japan, then the US. (b) $0.06683 - 0.03197 = 0.03486$. (c) $H_0 : \beta_5 = 0$ versus $H_1 : \beta_5 \neq 0$, $P = 0.075 > .05$ so we cannot reject H_0. (d) See `drop1`: 0.1857. (e) $P = .00085 < .05$, so we can reject $H_0 : \beta_4 = \beta_5 = 0$. (f)

```
fit = lm(log(mpg)~ log(weight)+year+I(year^2)+origin, auto)
summary(fit); drop1(fit)
fit2 = lm(log(mpg)~ log(weight)+origin, auto)
anova(fit2, fit)
fit3 = lm(log(mpg)~ log(weight)+poly(year,2, raw=T) +origin, auto)
summary(fit3); drop1(fit3, test="F")
```

Dummy Variables

- Question: How do we include nominal variables in a regression?
- Answer: Use dummy (also called indicator) variables
- Example: Quality Case. Let AM be a *dummy* variable that takes the value 1 if the observation comes from the morning shift and 0 otherwise (afternoon shift). As a first step we could regress defects on AM.

```
> fit = lm(defect~ am, quality)
> summary(fit)
...
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   16.920      4.308   3.927  0.00051 ***
am            20.440      6.093   3.355  0.00229 **
```

  Interpretation of 20.44: every unit increase in AM (i.e., going from PM to AM) is associated with a 20.44 change in defect rate. The AM shift has 20.44 more defects per thousand than the PM shift.

- This is equivalent to an independent-sample t-test with equal variances assumed:

```
> t.test(defect~am, quality, var.equal=T)
data:  defect by am
t = -3.3547, df = 28, p-value = 0.002295
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -32.920676  -7.959324
sample estimates:
mean in group 0 mean in group 1
          16.92           37.36
```

  The mean difference is $37.36 - 16.92 = 20.44$

- We reject $H_0 : \beta = 0$, or equivalently, $H_0 : \mu_{\text{AM}} = \mu_{\text{PM}}$ because $0.0023 < 0.05$.

Dummy Variables: Wholesaler Efficiency

If the Wholesaler is	Dummy Variable Coding Fair	Good	Outstanding
Fair	1	0	0
Good	0	1	0
Outstanding	0	0	1
Poor	0	0	0

```
> fit = lm(sales~ad+reps+as.factor(eff), data=click)
> drop1(fit, test="F")
               Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>                       71018 311.27
ad              1     41227 112245 327.58 19.7376 8.955e-05 ***
reps            1     54607 125625 332.09 26.1433 1.226e-05 ***
as.factor(eff)  3      4457  75475 307.71  0.7112     0.552

> summary(fit)
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       45.051     36.631   1.230    0.227
ad                13.063      2.940   4.443 8.96e-05 ***
reps              40.948      8.009   5.113 1.23e-05 ***
as.factor(eff)2    9.239     27.916   0.331    0.743
as.factor(eff)3   20.283     29.344   0.691    0.494
as.factor(eff)4   33.260     28.440   1.169    0.250
```

$$\hat{y} = 45 + 13\texttt{ad} + 41\texttt{reps} + 9\texttt{fair} + 20\texttt{good} + 33\texttt{out}$$

- First test H_0 : all $\beta_j = 0$ using `drop1` in R.
- `as.factor(eff)` indicates that `eff` should be treated as categorical (i.e., create dummies).
- What null hypothesis does the P-value for OUT test ($P = .2503$)? What does this test mean in English?

SPSS Estimates

Note: In practice first look at the ANOVA table (next page).

Parameter Estimates

Parameter	B	Std Error	t	Sig
Intercept	78.311	26.993	2.901	0.006
AD	13.063	2.940	4.443	0.000
REPS	40.948	8.009	5.113	0.000
[EFF=1]	−33.260	28.440	−1.169	0.250
[EFF=2]	−24.020	19.339	−1.242	0.223
[EFF=3]	−12.976	18.643	−0.696	0.491
[EFF=4]	0^a	.	.	.

a. This parameter is set to zero because it is redundant.

- Estimated regression equation: $\hat{y} = 78.3 + 13$`ad` $+ 41$`reps` -33`Poor` -24`Fair` -13`Good` $+ 0$`Out`
- Why are estimates different than on page 91? Note that `ad` and `reps` are identical.
- What null hypothesis does the P-value for `Poor` (EFF=1) test ($P = .250$)? What does this mean in English?
- Note that SPSS/Minitab/SAS do not give standardized regression coefficients for dummies. Why?

General Linear Test for Multiple Betas

```
> drop1(fit, test="F")
sales ~ ad + reps + as.factor(eff)
               Df Sum of Sq    RSS    AIC F value     Pr(F)
<none>                       71018    311
ad              1     41227 112245    328 19.7376 8.955e-05 ***
reps            1     54607 125625    332 26.1433 1.226e-05 ***
as.factor(eff)  3      4457  75475    308  0.7112     0.552

> deviance(fit)
[1] 71017.78

> anova(fit)
               Df Sum Sq Mean Sq  F value    Pr(>F)
ad              1 463451  463451 221.8787 < 2.2e-16 ***
reps            1  59327   59327  28.4032 6.414e-06 ***
as.factor(eff)  3   4457    1486   0.7112     0.552
Residuals      34  71018    2089
```

- The "ad" line tests $H_0 : \beta_1 = 0$ and is equivalent to the t test on the previous page. Likewise for the "reps" line.
- The "eff" line tests $H_0 : \beta_3 = \beta_4 = \beta_5 = 0$ (i.e., the three dummies for `eff` are all 0 meaning all levels of wholesaler efficiency are the same) versus H_1 : at least one of β_3, β_4, or β_5 is different from 0.
- The "< none >" line gives RSS for the full model

How to Handle Missing Values

```
agemiss = data.frame(
  age = c(NA,NA,35,NA,81,39,20,25,62,NA,45,57,36,39,NA,48,36,NA,NA,30,
     78,35,NA,20,26,28,44,30,31,32,72,33,33,NA,55,37,36,43,40,NA),
  y = c(2.9,2.8,8.4,2.8,4.5,8.3,9.4,9.1,5.6,2.9,7.4,6.3,7.7,8.1,3.2,6.5,
     7.9,3.0,3.0,9.0,5.1,8.8,3.4,9.5,8.9,8.3,7.4,8.3,8.6,8.7,5.3,8.3,
     7.8,3.2,6.6,8.4,8.6,7.8,7.6,3.7))
```

Solution: treat missing as a separate category and include a dummy:

1. Create dummy `xmiss` that equals 1 when `x` is missing and 0 otherwise.
2. When `x` is missing, set `x=0` (or impute with a regression)
3. Regress `y` on both `x` and `xmiss`

```
> agemiss$xmiss=is.na(agemiss$age)     # 1. create dummy
> agemiss$age[is.na(agemiss$age)] = 0  # 2. set missings to 0
> plot(agemiss$age, agemiss$y)
> fit = lm(y ~ age + xmiss, agemiss)   # 3. regression
> summary(fit)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.040189   0.163265   67.62   <2e-16 ***
age         -0.080755   0.003737  -21.61   <2e-16 ***
xmissTRUE   -7.950189   0.191438  -41.53   <2e-16 ***
Residual standard error: 0.3161 on 37 degrees of freedom
```

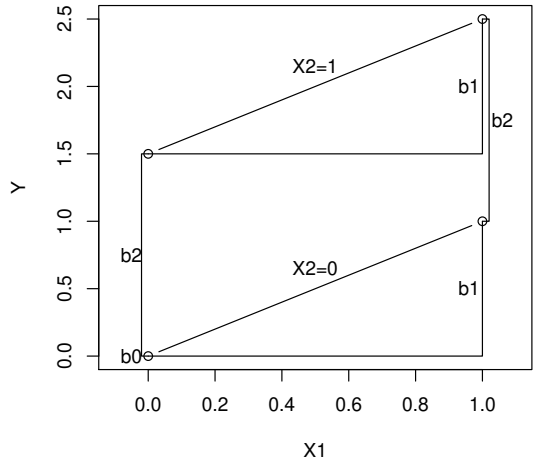
- When age is present, $y = 11.04 - 0.08\text{age} - 7.95(0)$.
- When age is missing, $y = 11.04 - 0.08(0) - 7.95(1)$

What is an Interaction?

Interaction terms are *nonlinear* combinations of *two or more* predictor variables. Suppose we have two **categorial** predictors (x_1 and x_2), which take only two value 0 and 1.

Linear (additive) model

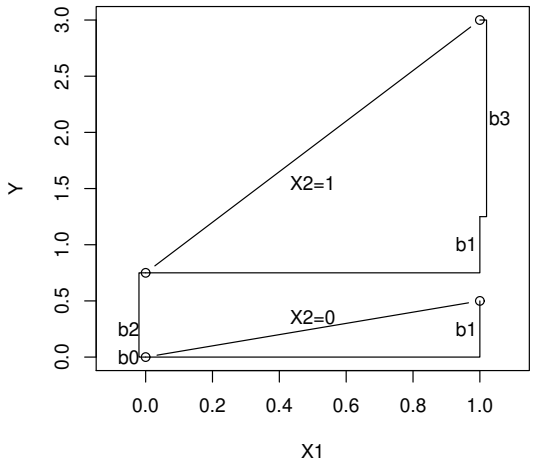
$$\hat{y} = b_0 + x_1 b_1 + x_2 b_2$$



x_1	x_2	$\hat{y}$
0	0	b_0
0	1	$b_0 + b_2$
1	0	$b_0 + b_1$
1	1	$b_0 + b_1 + b_2$

Linear model with product interaction term

$$\hat{y} = b_0 + x_1 b_1 + x_2 b_2 + x_1 x_2 b_3$$



x_1	x_2	$\hat{y}$
0	0	b_0
0	1	$b_0 + b_2$
1	0	$b_0 + b_1$
1	1	$b_0 + b_1 + b_2 + b_3$

Interactions in R

- `?formula` for help
- Additive effects: `x1 + x2`
- `x1:x2` = interaction between `x1` and `x2`, or use `*`

```
x1 + x2 + x1:x2 = x1*x2
```

- To specify main effects and all two-way interactions use

```
(a+b+c)^2 = a + b + c + a:b + a:c + b:c
```

- Use the minus sign to drop terms, e.g.,

```
(a+b+c)^2 - a:b = a + b + c + a:c + b:c
```

- To specify quadratic effects use `I( )`
- `as.factor(a)` casts `a` as a factor. To do this permanently you can type `dat$a = as.factor(dat$a)`
- Use `drop1` to determine if terms can be dropped.

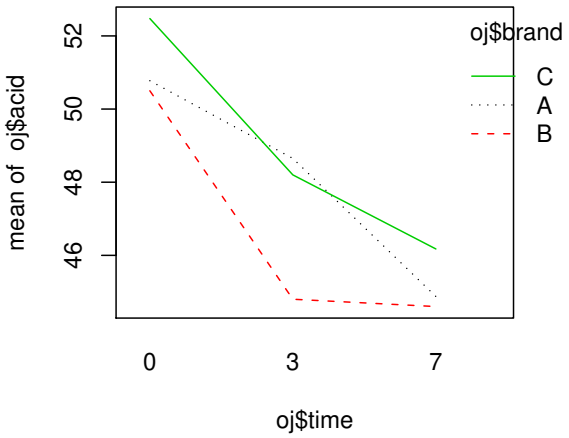
ACT Example

A company that prepares students for the ACT college entrance exam is testing new curriculum. They want to investigate the length of the course (condensed 10-day course versus regular 30-day course) and the modality (traditional classroom versus online distance). Students were assigned at random to the four treatment combinations.

```
> course = data.frame(
  type=factor(c(rep(1,20), rep(2,20)), 1:2, c("Trad","Online")),
  length=factor(c(rep(1,10), rep(2,10), rep(1,10), rep(2,10)),
    1:2, c("Condensed","Regular")),
  act=c(26,27,25,21,21,18,24,19,20,18,  34,24,35,31,28,28,21,23,29,26,
        27,29,30,24,30,21,32,20,28,29,  24,16,22,20,23,21,19,19,24,25))

> fit = aov(act ~ type*length, course)
> summary(fit)
            Df Sum Sq Mean Sq F value   Pr(>F)
type         1    5.6     5.6   0.399    0.532
length       1    0.2     0.2   0.016    0.900
type:length  1  342.2   342.2  24.257 1.89e-05 ***
Residuals   36  507.9    14.1
> interaction.plot(course$length, course$type, course$act)

> library(dplyr)
> course %>%
  group_by(type, length) %>%
  summarize(n=n(), mean=mean(act))

  type   length        n  mean
1 Trad   Condensed    10  21.9
2 Trad   Regular      10  27.9
3 Online Condensed    10  27
4 Online Regular      10  21.3

> fit$coef
             (Intercept)               typeOnline
                    21.9                      5.1
          lengthRegular typeOnline:lengthRegular
                     6.0                    -11.7
```

Orange Juice Example

To ascertain the stability of vitamin C in reconstituted frozen OJ concentrate stored in a refrigerator for a period of up to one week, a study was conducted on three brands of a three different times (days).




```
> oj = data.frame(brand=c(rep("A",12), rep("B",12), rep("C",12)),
  time = factor(rep(c(0,0,0,0,3,3,3,3,7,7,7,7), 3)),
  acid = c(52.6,54.2,49.8,46.5,49.4,49.2,42.8,53.2,42.7,48.8,40.4,47.6,56,48,
    49.6,48.4,48.8,44,44,42.4,49.2,44,42,43.2,52.5,52,51.8,53.6,48,47,48.2,
    49.6,48.5,43.4,45.2,47.6))
> with(oj, interaction.plot(time, brand, acid, col=1:3, lwd=2))
> fit = lm(acid ~ time*brand, oj)
> drop1(fit, test="F")
Single term deletions
Model: acid ~ time * brand
           Df Sum of Sq    RSS    AIC F value Pr(>F)
<none>                  254.14 88.357
time:brand  4    17.301 271.44 82.728  0.4595 0.7647

> fit = lm(acid ~ time+brand, oj)
> drop1(fit, test="F")
Single term deletions
Model: acid ~ time + brand
       Df Sum of Sq    RSS     AIC F value    Pr(>F)
<none>              271.44  82.728
time    2   226.676 498.12 100.583 12.9438 8.191e-05 ***
brand   2    32.962 304.40  82.854  1.8822    0.1692
```

Capacitor Example in R

```
> capacitor = data.frame(
  bondmat=factor(c(rep(1,12),rep(2,12),rep(3,12),rep(4,12))),
  substrate=rep(c(rep("A",4),rep("B",4),rep("C",4)), 4),
  y=c(1.51,1.96,1.83,1.98,1.63,1.92,1.8,1.71,3.04,3.16,3.09,3.5,2.62,2.82,
    2.69,2.93,3.12,2.94,3.23,2.99,1.91,2.11,1.78,2.25,2.96,2.82,3.11,3.11,
    2.91,2.93,3.01,2.93,3.04,2.91,2.48,2.83,3.67,3.4,3.25,2.9,3.48,3.51,
    3.24,3.45,3.47,3.42,3.31,3.76)
)
> attach(capacitor)
> table(substrate, bondmat)   # note orthogonal design
         bondmat
substrate 1 2 3 4
        A 4 4 4 4
        B 4 4 4 4
        C 4 4 4 4

> tapply(y, data.frame(substrate, bondmat), mean)
         bondmat
substrate      1      2     3     4
        A 1.8200 2.7650 3.000 3.305
        B 1.7650 3.0700 2.945 3.420
        C 3.1975 2.0125 2.815 3.490
> interaction.plot(substrate, bondmat, y, col=1:4)
> interaction.plot(bondmat, substrate, y, col=1:4)
> fit = aov(y~bondmat*substrate, capacitor)

> anova(fit)
Response: strength
                  Df Sum Sq Mean Sq F value    Pr(>F)
bondmat            3 8.4605 2.82017 80.7654 4.709e-16 ***
substrate          2 0.1953 0.09766  2.7968    0.0743 .
bondmat:substrate  6 7.5869 1.26449 36.2130 7.977e-14 ***
Residuals         36 1.2570 0.03492
> plot(fit)
> detach(capacitor)
```

Your Turn

1. *Montgomery 14.2.* An engineer suspects that the surface finish of metal parts is influenced by the type of paint used and the drying time. He selects three drying times and two types of paint. The data are as follows:

```
paint = data.frame(
  type=factor(c(rep(1,9), rep(2,9))),
  time=factor(rep(c(rep(20,3), rep(25,3), rep(30,3)), 2)),
  y=c(74,64,50,  73,61,44,  78,85,92,  92,86,68,  98,73,88,  66,45,85))
```

2. *Montgomery 14.4.* An experiment was conducted to determine whether either firing temperature of furnace position affects the baked density of a carbon anode. Data are as follows:

```
anode = data.frame(
  pos = factor(c(rep(1,9), rep(2,9))),
  temp = factor(rep(c(rep(800,3), rep(825,3), rep(850,3)), 2)),
  density = c( 570,565,583,  1063,1080,1043,  565,510,590,
               528,547,521,   988,1026,1004,  526,538,532))
```

Solutions

1. Paint problem

```
fit = lm(y~type*time, paint)
summary(fit)
drop1(fit, test="F")
summary(fit)
with(paint, interaction.plot(type, time, y, col=1:3))
with(paint, interaction.plot(time, type, y, col=1:2))
```

2. anode problem

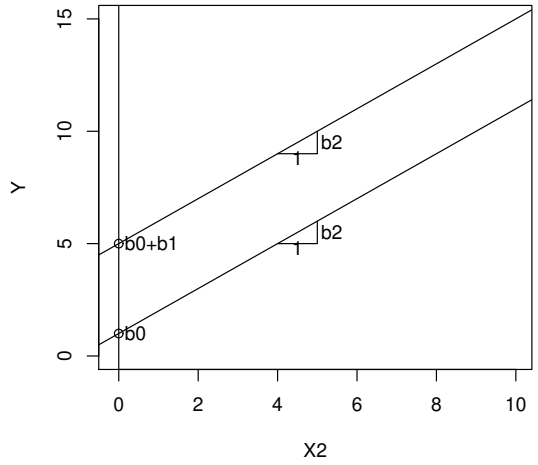
```
fit = lm(density ~ pos*temp, anode)
drop1(fit, test="F")
summary(fit)
with(anode, interaction.plot(pos, temp, density, col=1:3))
with(anode, interaction.plot(temp, pos, density, col=1:2))
fit = lm(density ~ pos+temp, anode)
drop1(fit, test="F")
summary(fit)
```

What is an Interaction?

Now suppose that x_1 is **categorial** x_1 taking values 0 and 1, and x_2 is **numerical**.

Constant Slope Model

$$\hat{y} = b_0 + x_1 b_1 + x_2 b_2$$



Bottom Line ($x_1 = 0$)

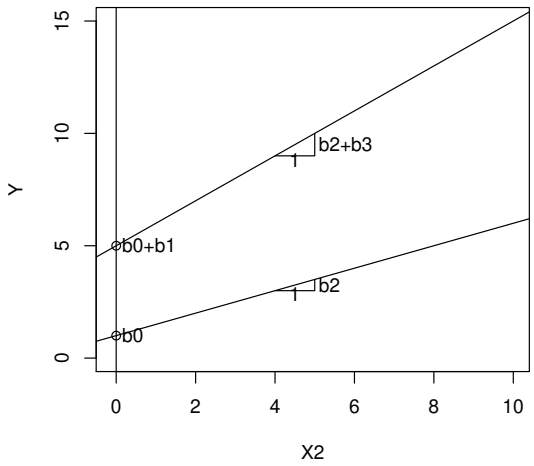
$$y = b_0 + b_2 x_2$$

Top Line ($x_1 = 1$)

$$y = (b_0 + b_1) + b_2 x_2$$

Different-Slope Model

$$\begin{aligned} \hat{y} &= b_0 + x_1 b_1 + x_2 b_2 + x_1 x_2 b_3 \\ &= (b_0 + x_1 b_1) + (b_2 + x_1 b_3) x_2 \end{aligned}$$



Bottom Line ($x_1 = 0$)

$$y = b_0 + b_2 x_2$$

Top Line ($x_1 = 1$)

$$y = (b_0 + b_1) + (b_2 + b_3) x_2$$

Newfood with a Numerical*Categorial Interaction

```
> fit = lm(sales ~ price*ad + volume, newfood)
> drop1(fit, test="F")
Single term deletions

Model:
sales ~ price * ad + volume
         Df Sum of Sq   RSS    AIC F value    Pr(>F)
<none>                26291 177.97
volume    1     51729 78019 202.08 37.3837 7.052e-06 ***
price:ad  1      8752 35042 182.87  6.3246   0.02107 *

> summary(fit)

Call: lm(formula = sales ~ price * ad + volume, data = newfood)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   12.391    111.196   0.111  0.91244
price         -7.259      2.657  -2.732  0.01324 *
ad           399.488    109.339   3.654  0.00169 **
volume        11.456      1.874   6.114 7.05e-06 ***
price:ad      -9.382      3.730  -2.515  0.02107 *

Residual standard error: 37.2 on 19 degrees of freedom
Multiple R-squared:  0.8572,Adjusted R-squared:  0.8271
F-statistic: 28.51 on 4 and 19 DF,  p-value: 8.55e-08
```

Quadratic Surfaces

Suppose we have two numerical predictors, x_1 and x_2

$$\begin{aligned}\hat{y} &= b_0 + b_1 x_1 + b_2 x_2 + b_{11} x_1^2 + b_{22} x_2^2 + b_{12} x_1 x_2 \\ &= b_0 + \mathbf{x}^\mathsf{T}\mathbf{b} + \mathbf{x}^\mathsf{T}\mathbf{B}\mathbf{x}\end{aligned}$$

where

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12}/2 \\ b_{12}/2 & b_{22} \end{pmatrix}.$$

We can find the optimum value as follows

$$\frac{\partial \hat{y}}{\partial \mathbf{x}} = \mathbf{b} + 2\mathbf{B}\mathbf{x} = 0$$

Solve to find the stationary point $\mathbf{x}_s$

$$\mathbf{x}_s = -\frac{1}{2}\mathbf{B}^{-1}\mathbf{b}.$$

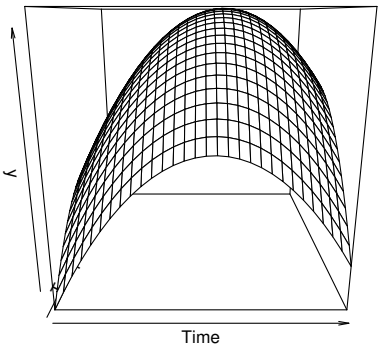
This result holds for k predictors as well.
Let λ_1 and λ_2 be the eigenvalues of $\mathbf{B}$.

- If $\lambda_1 > 0$ and $\lambda_2 > 0$, then $\mathbf{x}_s$ is a minimum
- If $\lambda_1 < 0$ and $\lambda_2 < 0$, then $\mathbf{x}_s$ is a maximum
- If $\lambda_1 \lambda_2 < 0$, then $\mathbf{x}_s$ is a saddle point

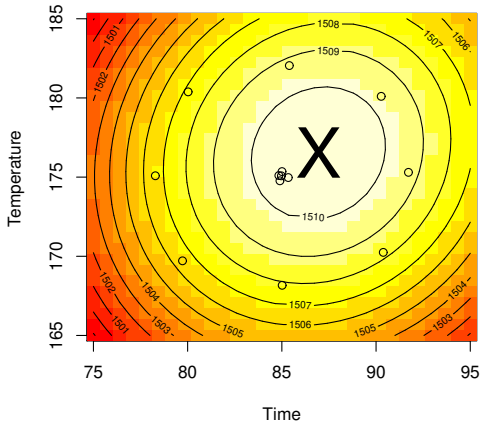
Chemical Engineering Example

```
> fit = lm(yield ~ time*temp+I(time^2)+I(temp^2), cheme)
> summary(fit)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.431e+03  1.529e+02  -9.360 3.30e-05 ***
time         7.809e+00  1.158e+00   6.744 0.000266 ***
temp         1.327e+01  1.485e+00   8.940 4.46e-05 ***
I(time^2)   -5.506e-02  4.039e-03 -13.630 2.69e-06 ***
I(temp^2)   -4.005e-02  4.039e-03  -9.916 2.26e-05 ***
time:temp    1.000e-02  5.326e-03   1.878 0.102519
```

```
> b = coef(fit)[2:3]
> B = matrix(
  c(coef(fit)[c(4,6,6,5)])*c(1,.5,.5,1),
  nrow=2)
> xs = -solve(B) %*% b/2
> xs
          [,1]
[1,]  86.94615
[2,] 176.52923
> eigen(B)$values
[1] -0.03853994 -0.05657147
```




```
> size = 30
> x1 = seq(75, 95, length=size)
> x2 = seq(165, 185, length=size)
> y = outer(x1, x2, function(x1 ,x2)
    -0.001431 + 7.809*x1
    + 13.27*x2 -0.05506*x1^2
    -0.04005*x2^2 + 0.01*x1*x2
  )
> persp(x1, x2, y, xlab="Time",
  ylab="Temperature")

> image(x1, x2, y, xlab="Time",
  ylab="Temperature")
> contour(x1, x2, y, add=T)
> points(jitter(cheme$time),
  jitter(cheme$temp))
> points(xs[1], xs[2], pch="X", cex=4)
```



Chemical Engineering Example

```
> fit = lm(yield ~ time*temp+I(time^2)+I(temp^2), cheme)
> summary(fit)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.431e+03  1.529e+02  -9.360 3.30e-05 ***
time         7.809e+00  1.158e+00   6.744 0.000266 ***
temp         1.327e+01  1.485e+00   8.940 4.46e-05 ***
I(time^2)   -5.506e-02  4.039e-03 -13.630 2.69e-06 ***
I(temp^2)   -4.005e-02  4.039e-03  -9.916 2.26e-05 ***
time:temp    1.000e-02  5.326e-03   1.878 0.102519

> b = coef(fit)[2:3]
> B = matrix(c(coef(fit)[c(4,6,6,5)])*c(1,.5,.5,1), nrow=2)
> xs = -solve(B) %*% b/2
> xs
          [,1]
[1,]  86.94615
[2,] 176.52923
> eigen(B)$values
[1] -0.03853994 -0.05657147

> size = 30
> x1 = seq(75, 95, length=size)
> x2 = seq(165, 185, length=size)
> y = outer(x1, x2, function(x1 ,x2)
    -0.001431 + 7.809*x1 + 13.27*x2 -0.05506*x1^2
    -0.04005*x2^2 + 0.01*x1*x2
  )
> persp(x1, x2, y, xlab="Time", ylab="Temperature")

> image(x1, x2, y, xlab="Time", ylab="Temperature")
> contour(x1, x2, y, add=T)
> points(jitter(cheme$time), jitter(cheme$temp))
> points(xs[1], xs[2], pch="X", cex=4)
```

Steps In Reading Data

1. Read in each table and run descriptive statistics on each variable. CRISP-DM link
   - Categorical variables: frequency distribution, perhaps sorted in descending order of frequency
   - Numerical variables: count, mean, standard deviation, skewness, min, and max. Common percentiles are also useful, but require sorting. Boxplots and/or histograms also helpful.
   - Do values make sense? For example, counts and amounts cannot take negative values.
   - Correlations and/or crosstabs are also worth looking at.
   - Which variables can you trust?
2. Clean data, e.g.,
   - Standardize strings, e.g., Street, St. Str. ST, etc.
   - Deal with outliers
   - Check computed fields
   - Reduce file size, e.g., drop variables, use factor labels
3. Determine primary keys, check that they are unique, understand relationships between tables.
4. I save a file with descriptives for every data set I read in, and refer back to it sometimes for decades!

Sources of Data

- **Surveys** and **Panels** (e.g., Nielsen)
- **Third party** data (e.g., Acxiom, Experian, Dun & Bradstreet, government): demographics, credit, some interest indicators
- Web scrapable data, e.g., Twitter
- **First-party data**: data about your customers/audience
- **Second-party data**: someone else's first-party data that you buy or trade for
- **Behavior logs**: record behaviors over time (id, time/date, behavior), e.g., transaction histories, call-center logs, Web logs, mobile location logs, contact histories, clickstream data, social media participation history, search, TV tuning data

id	amt	giftdate	source
18	10	20JAN2004	04G9MEX
18	5	22JUN2004	04LQZAP
935	100	31DEC2001	02EQZDA
935	100	17APR2003	03GQZVA
935	100	21MAR2005	05GQZFA
1347	80	31DEC2001	02EQZDA
1347	100	27DEC2002	02KQZEA
1347	10	10JAN2005	05EQZHA
1347	125	28NOV2005	05GZCZA

id	amt	giftdate	source
1537	10	13SEP2001	02B9YDK
1537	15	05FEB2004	04GQZDF
1776	50	03JAN2002	02E9YBN
2238	5	17FEB2003	03GQZBD
2238	5	28JAN2004	04GQZAD
2238	5	04FEB2005	05GQZAC
2238	5	31AUG2005	06BQZAD
2238	5	02FEB2006	06GQZAC

Aggregating Customer Behavior Logs

- Raw form is not useful. Create *behavioral feature variables*.
- Start by creating *simple summary* variables: counts, sums, means, mins, or maxes by customer.
  - **Recency** (R): time since last purchase (max or min function)—indicator of inactivity
  - **Frequency** (F): Number previous behaviors (count function)—indicator of behavioral loyalty
  - **Monetary** (M): total "spend" over a past period of time (sum). Think of "spend" in general, e.g., dollar amount, but also time spent, etc. (or call it "total time," "extent," etc.)
  - **Time of file** (TOF): time since first purchase (min/max)

```
CREATE TABLE rfm AS
SELECT
  id,
  MIN("01JAN2007"D-giftdate)/365.25 AS rec,
  MAX("01JAN2007"D-giftdate)/365.25 AS tof,
  COUNT(amt) AS freq,
  SUM(amt) AS mon
FROM curr.trans(obs=500)
GROUP BY id;
```

Donor ID	rec	tof	freq	mon
18	2.527	2.948665	2	15
935	1.782	5.002053	3	300
1347	1.092	5.002053	4	315
1537	2.90	5.300479	2	25
1776	4.99	4.99384	1	50
2238	0.912	3.871321	5	25

Customer Behavior Logs

- Functions of simple summaries:
  - F and M usually very right skewed—usually log them.
  - **Average amount**: M/F or use mean function—usually the best predictor of future order amounts, time per session, etc. Note that $\log(M/F) = \log(M) - \log(F)$
  - **Purchase rate**: purchases/period, e.g., $\lambda = F/\text{TOF}$
- More complicated summaries are possible, e.g.,
  - Customer-level trends (regress spend per period on period number by customer, then use slope)
  - Let F and M decay over time, or look at only the past n years (a purchase 20 years ago is not as informative as one last week). Let a_{it} be the amount of a transaction for customer i that occured $t \geq 0$ years ago and let $d \geq 0$ be the discounting rate. The monetary value for customer i could be computed as

$$m_i = \sum_t \frac{a_{it}}{(1+d)^t}$$

- What to do about missing values?
- Should you standardize?

Contest data set

You have the transaction file from a non-profit organization between 2002 and 2006.

1. Create an RFM data set as of 9/1/2005
2. Create a variable indicating whether each customer made a donation in 9/1/2005–8/31/2006
3. Merge the data sets from steps (2) and (3). You could now build a predictive model.

This task could be done in SQL or using dplyr or data.table. My interaction video introduced dplyr.

```
> setwd("/Users/ecm/teach/data/CLVcontest")
> trans = read.csv("trans.csv")
> trans$source = NULL # not needed for example
> trans$t = as.numeric(as.Date("2005/09/01") - as.Date(trans$giftdate,"%m/%d/%Y"))/365.25
> # t is time (years) between giftdate and 9/1/2005
```

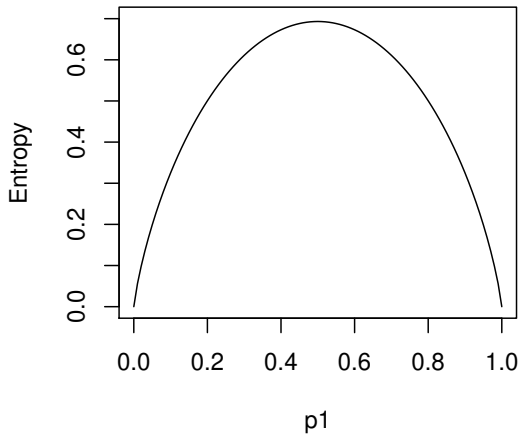
```
> head(trans, 13)
        id amt   giftdate          t
1  8128357   5 02/22/2002  3.5236140
2  9430679  50 01/10/2002  3.6413415
3  9455908  25 04/19/2002  3.3702943

4  9652546 100 04/02/2002  3.4168378
5  9652546 100 01/06/2003  2.6529774
6  9652546 100 01/05/2004  1.6563997
7  9652546  50 12/28/2004  0.6762491
8  9652546 100 12/27/2005 -0.3203285

9  9791641  50 01/17/2002  3.6221766
10 9791641  50 12/16/2002  2.7104723
11 9791641  25 10/15/2003  1.8809035
12 9791641  50 03/04/2004  1.4948665
13 9791641 100 02/13/2006 -0.4517454
```

```
> head(rfm, 5)  # computed on next page
# A tibble: 5 x 5
       id     r     f     m   aos
    <int> <dbl> <int> <dbl> <dbl>
1  8128357 3.52      1   5.00  5.00
2  9430679 3.64      1  50.0  50.0
3  9455908 3.37      1  25.0  25.0
4  9652546 0.676     4 350.   87.5
5  9791641 1.49      4 175.   43.8
6 10458867 3.40      1  25.4  25.4
```

```
> library(dplyr)
> trans = read.csv("trans.csv")
> trans = trans %>%
    mutate(dt = as.Date(trans$giftdate, "%m/%d/%Y"),
        t = as.numeric(as.Date("2005/09/01") - dt)/365.25) %>%
    select(id, amt, dt, t)
> head(trans)

> # make RFM as of 9/1/05
> rfm = trans %>%
  filter(t>=0) %>%
  group_by(id) %>%
  summarise(datelp=min(t), f=n(), m=sum(amt), aos=mean(amt))
> head(rfm)
> dim(rfm)

# create variable indicating donations in 9/1/05-8/31/06
> dv = trans %>%
  filter(t<=0) %>%
  group_by(id) %>%
  summarise(targfreq=n(), targdol=sum(amt))
> dim(dv)
> head(dv)

> # merge them
> newrfm = left_join(rfm, dv, by="id") %>%
  mutate(targfreq = ifelse(is.na(targfreq), 0, targfreq),
         targdol=ifelse(is.na(targdol), 0, targdol))
> dim(newrfm)
> summary(newrfm)
> head(newrfm, 10)

# A tibble: 10 x 7
         id     r     f      m    aos targfreq targdol
      <int> <dbl> <int>  <dbl>  <dbl>    <dbl>   <dbl>
 1  8128357 3.52      1   5.00   5.00       0.      0.
 2  9430679 3.64      1  50.0   50.0        0.      0.
 3  9455908 3.37      1  25.0   25.0        0.      0.
 4  9652546 0.676     4 350.    87.5        1.    100.
 5  9791641 1.49      4 175.    43.8        1.    100.
 6 10458867 3.40      1  25.4   25.4        0.      0.
 7 10544021 1.40      2 175.    87.5        0.      0.
 8 10581619 0.408     7  70.0   10.0        1.     10.
 9 10614089 1.86      2   7.00   3.50       0.      0.
10 10640019 3.36      1 200.   200.         0.      0.

> # basic predictive model
> fit = lm(log(targdol+1) ~ log(r) + log(f) + log(m), newrfm)
> summary(fit)
```

Taxonomy Metadata

- Sometimes you will have a way to group behaviors into **types** (e.g., SKU dictionaries, product categories and subcatetories, genres, types of credit card merchants, Cubs vs. Sox etc.)
- Often, the taxonomy data will have hierarchies, e.g., category, sub-category, sub-sub-category, etc., e.g., (dairy, yogurt, non-fat yogurt) or (sports, baseball, Cubs)
- Metadata necessary because of cold start and perfect substitute problems. Otherwise you can use data-driven approaches (PCA/factor, deep learning)
- Compute simple summaries (e.g., F, M) by customer and type
- **Scope** = number of distinct types
- **Entropy** = measure of variety seeking within or across types. Let p_i be the fraction of F or M from type i. Entropy is $-\sum_i p_i \log(p_i)$.
- Suppose there are only two types, so $p_1 = 1 - p_2$. Then see plot to right.
- Also Gini and Simpson's D



Using Spread to Count Orders by Category

```
> library(tidyverse)
> dat = data.frame(
    id = c(1,1,1,1,2,2,3,3,3,3),
    cat = c("A","A","A","B","B",
            "B","A","B","B","B"),
    amt = 1:10
  )

> dat
   id cat amt
1   1   A   1
2   1   A   2
3   1   A   3
4   1   B   4
5   2   B   5
6   2   B   6
7   3   A   7
8   3   B   8
9   3   B   9
10  3   B  10

> dat %>%
    group_by(id, cat) %>%
    summarize(f=n())

# A tibble: 5 x 3
# Groups:   id [?]
     id cat       f
  <dbl> <fct> <int>
1    1. A         3
2    1. B         1
3    2. B         2
4    3. A         1
5    3. B         3
```

```
> # spread transposes data
> dat %>%
    group_by(id, cat) %>%
    summarize(f=n()) %>%
    spread(cat, f)
# A tibble: 3 x 3
# Groups:   id [3]
     id     A     B
  <dbl> <int> <int>
1    1.     3     1
2    2.    NA     2
3    3.     1     3

> # add fill=0 to replace NA with 0
> dat %>%
    group_by(id, cat) %>%
    summarize(f=n()) %>%
    spread(cat, f, fill=0)
# A tibble: 3 x 3
# Groups:   id [3]
     id     A     B
  <dbl> <dbl> <dbl>
1    1.    3.    1.
2    2.    0.    2.
3    3.    1.    3.
```

Your Turn

1. Read the following transaction file:

id	buydate	amt
1	04DEC2011	20
1	20FEB2012	10
1	02MAR2012	5
2	09MAR2011	100
3	30NOV2010	50
3	01JAN2012	25
4	14MAR2011	20
4	28JUN2011	30

   (a) Write a program to create a data set listing all unique customer id's in the data set. Keep only the customer id.

   (b) Create an "RFM" data set with recency, frequency, monetary value assuming that today is 09APR2012. Compute also the time on file (days since the first purchase) and the inter-purchase time (time on file / freq).

2. DMEF1 problem. Overview: You will predict behavior during the target period (`targdol`) based on previous donation history from the base period (e.g., RFM) and customer characteristics (e.g., sex). The business situation is a non-profit organization that uses direct mail to solicit additional contributions from past donors. The base time period is 10/96–6/05. The target time period is 10/05–12/05. Your goal is to predict donations during this period. The universe is previous donors who received at least one solicitation in early 10/05. The sample is drawn without replacement $n = 5070$ (for this assignment). Database tables:

   - Customer table `donor`: ID, dollars in target period, zip, sex, other customer info
   - Donation table `onecn`: code, date, amount, ID
   - Solicitation/donation type `onecode`: code, codetype

   (a) Each solicitation has a `code` indicating its type. These codes can be grouped into 5 types. The file `onecode` shows how `code` values are assigned to the 5 types (`codetype`). Print the first 10 observations and run a frequency distribution of `codetype`.

   (b) Merge the transaction file with `onecode` by the `code` variable. Run a frequency distribution of `codetype` and means on the resulting file. Print the first 10 observations and study them.

   (c) Use the file from the previous part to create an RFM data set. In addition to RFM, create variables `freqA` and `freqD` counting the number of previous donations to solicitations of `codetype` A and D, respectively. Print the first 10 observations and make sure that you computed everything correctly. Also run descriptives.

   (d) Merge the RFM data set you just created with the `donor` data set. Compute `logtarg = LOG(targdol+1)` and `targresp` equaling 1 for those who responded (`targdol` greater than 0) and 0 for non-responders.

   (e) Correlate `logtarg`, `targresp` and `targdol` with `rec`, `freq`, `freqA`, `freqD` and `totdols`. Which of the variables are good predictors of future behavior? Do your conclusions change if you consider response versus amount spent in the future?

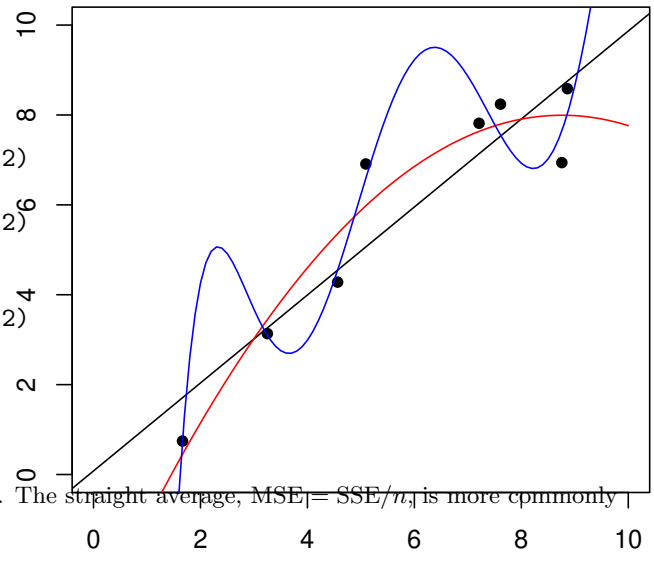
How to Measure Predictive Accuracy

- Assume **training data set** $(\mathbf{x}_i, y_i)$, where $i = 1, \ldots, n$ and y_i is numerical.
- Estimate model f with p parameters using only training data and summarize residuals[7] with, e.g.,

$$\text{SSE} = \sum_{i=1}^{n} [y_i - f(\mathbf{x}_i)]^2, \quad \text{MSE} = \frac{\text{SSE}}{n}, \text{or} \quad R^2 = 1 - \frac{\text{SSE}}{\text{SST}}$$

- Problem: if measure computed on the data that was used to estimate the model, it will be *optimistic*. Such a rate is called the apparent rate. The model is "fine-tuned" to do well on this data set and may "capitalize on chance."
- Example: $y = x + e$ where $n = 8$ and $e \sim \mathcal{N}(0, 1)$

```
> set.seed(12345)
> train = data.frame(x = runif(8)*10)
> train$y = train$x + rnorm(8)

> fit1 = lm(y~x, train) #black
> fit2 = lm(y~x + I(x^2), train) #red
> fit3 = lm(y~poly(x, 6), train) #blue
> mean((train$y - predict(fit1, train))^2)
[1] 1.043647
> mean((train$y - predict(fit2, train))^2)
[1] 0.4878706
> # degree 6 polynomial gives best fit
> mean((train$y - predict(fit3, train))^2)
[1] 0.2301981
```



[7] Until now, MSE $= \text{SSE}/(n - p - 1)$, which is unbiased. The straight average, MSE $= \text{SSE}/n$, is more commonly used when computing out-of-sample estimates.

Two Approaches for Honest Estimates of Prediction Error

- **Penalized estimates**, e.g.,[8]

$$R_a^2 = 1 - \frac{\text{SSE}/(n-p-1)}{\text{SST}/(n-1)} \quad \text{or} \quad \text{AIC} = f(\text{SSE}) + 2(p+1)$$

but for many models p is not known (e.g., neural networks) and there are different penalties

```
> summary(fit1)$adj.r.squared
[1] 0.8237407
> # adjusted R^2 gets it wrong!
> summary(fit2)$adj.r.squared
[1] 0.9011255
> summary(fit3)$adj.r.squared
[1] 0.7667342
```

```
> AIC(fit1)
[1] 29.04479
> AIC(fit2) # AIC gets it wrong!
[1] 24.96138
> AIC(fit3)
[1] 26.95249
```

- **Out-of-sample estimates**, e.g., "conjure up" a *test set* of 10,000 cases and use them to evaluate fit, but not to estimate models

```
> test = data.frame(x = runif(10000)*10)
> test$y = test$x + rnorm(10000)
> mean((test$y-predict(fit1, test))^2)  # model 1, close to true value
[1] 1.006563
> mean((test$y-predict(fit2, test))^2)  # model 2 overfits
[1] 2.450303
> mean((test$y-predict(fit3, test))^2)  # model 3 really overfits
[1] 636.7939
```

[8] Note: For AIC, $f(\text{SSE}) = n\log(2\pi) + n\log(\text{SSE}/n) + n$. The details are not important. The main idea is that AIC is some version of SSE plus a penalty of 2 times the number of parameters (including the intercept).

Variable Selection Problems

- Now add two "decoy" predictors (that are just noise)

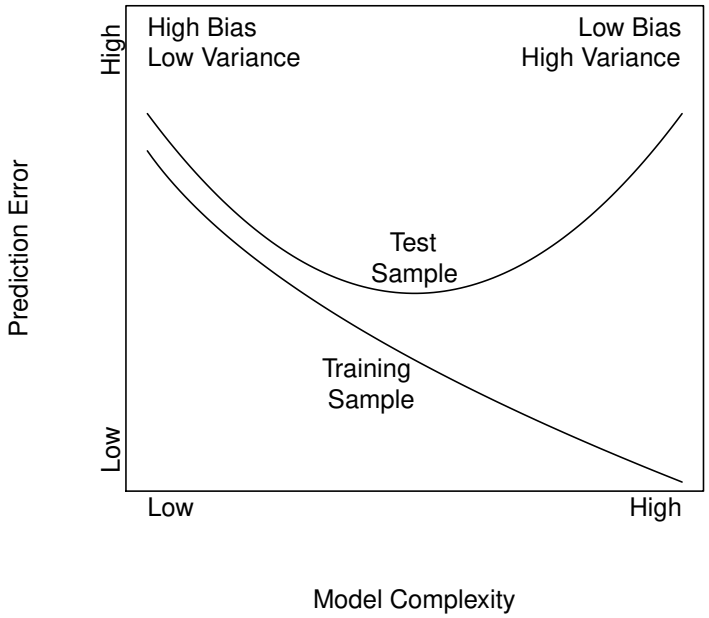
```
> train$x2 = runif(8)
> train$x3 = runif(8)
> test$x2 = runif(10000)
> test$x3 = runif(10000)

> fit4 = lm(y ~ x+x2+x3, train)
> summary(fit4)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2604     1.8061   0.144  0.89232
x             0.9881     0.1746   5.658  0.00481 **
x2            1.1695     1.8216   0.642  0.55580
x3           -2.8721     2.2520  -1.275  0.27123
> mean((train$y - predict(fit4, train))^2)
[1] 0.7142952
> mean((test$y - predict(fit4, test))^2)
[1] 2.234198
> AIC(fit4)
[1] 30.01134
> summary(fit4)$adj.r.squared
[1] 0.8190464
```

  In this case R_a^2, AIC and the test set get it right.

- Key points
  - When building a model, the modeler must decide on (1) how flexible model should be (e.g., degree of polynomial) and (2) which variables should be included.
  - We should not use RSS, MSS, R^2, etc. because they are optimistic.
  - These decisions can be made with penalized or out-of-sample measures.

Model Complexity

- Models can be made more or less "complex," e.g.,
  - Number of variables in model (e.g., stepwise selection)
  - Flexibility, e.g., degree of polynomial terms in linear regression, number of hidden nodes in a neural network, number of leaves on a tree, number of bins in bin smoother
  - Use penalized least squares, e.g., ridge regression and lasso, smoothing splines, weight decay for neural networks
- The modeler must select the appropriate complexity so that the model does not capture idiosyncrasies of the particular data set (called **overfitting** the data)



Out-of-Sample Estimates of Prediction Error

- **Test sets**
  - Take the available data, draw a sample of observations, and set them in a "safe" while you build the model (called the *holdout* or *test* sample).
  - Use the remaining data, called the *estimation* or *training* sample, to build your model. The training set should be large enough to make reliable estimates, but overly large, which would unnecessarily waste computing resources.
  - Apply estimated model to the test sample and evaluate accuracy.
- K-Fold Cross validation (see Wikipedia for variations)
  - Step 1: Split available data into K roughly equal-sized parts
  - Step 2: For part k, fit the model on the other $K-1$ parts, apply the estimated model to part k, and evaluate fit
  - Step 3: Repeat step 2 for $k = 1, \ldots, K$ and combine the K evaluations of fit
  - Most data mining books suggest $K = 5$ or 10
  - This is computationally more expensive than training/test splits, but makes more efficient use of the data — **use when available sample size is small**

Out-of-Sample Estimates of Prediction Error: Fresh Data

"The second level of cross-validation, which, by analogy with the physician's "double-blind" study, we have called "double cross-validation", is to be had only by going to fresh data. These fresh data are best gathered after choosing form and coefficients. When fresh gathering is not feasible, good results can come from going to a body of data that has been kept in a locked safe where it has rested untouched and unscanned during all the choices and optimizations. For the full validating effect, the data placed in the safe must differ from those used to choose the procedure in ways that adequately represent the sources of variation anticipated in practice. For example, they may need to involve distinct school systems, distinct investigators, or distinct years of observations. (from Mosteller and Tukey (1977), *Data Analysis and Regression*, p. 38.)"

Good practice (at least when you have sufficient data) is have a three-way split

- **Training data**: used to estimate model parameters
- **Validation data**: used to select model hyper parameters (or use K-fold cross validation if you are data poor)
- **Test data**: used for final, inter-model comparisons

Test Sets in R

```
> set.seed(12345)
> employee$train = runif(nrow(employee))>.5  # assign to test/train set
> employee$salaryK = employee$salary/1000    # salary in $K

> dim(employee)  # note: 71 rows
[1] 71  8
> table(employee$train)
FALSE  TRUE
   36    35

> fit = lm(salaryK ~ ageyrs + expyrs, employee, subset=train)
> anova(fit)      # note 35 rows used to fit model
Analysis of Variance Table

Response: salaryK
          Df  Sum Sq Mean Sq F value  Pr(>F)
ageyrs     1  473.05  473.05  5.8292 0.02166 *
expyrs     1  518.18  518.18  6.3855 0.01665 *
Residuals 32 2596.82   81.15

> sum(fit$residuals^2)  # compute training RSS from fitted object
[1] 2596.816
> deviance(fit)         # Or just use deviance function
[1] 2596.816
> mean(fit$residuals^2)    # training MSE with n in denominator
[1] 74.19475

> yhat = predict(fit, employee[!employee$train,])  # apply model to test set
> length(yhat)   # note 36 predictions
[1] 36
> mean((employee$salaryK[!employee$train] - yhat)^2)    # test MSE
[1] 80.52033
```

K-Fold Cross-Validation in R

```
> set.seed(12345)
> employee$cv = as.integer(runif(nrow(employee))*5)
> table(employee$cv)
 0  1  2  3  4
13 15 10 19 14

> yhat = rep(NA, nrow(employee))   # set up vector for held-out predictions
> for(i in 0:4){
+ fit = lm(salaryK~ageyrs+expyrs, employee, subset=(cv!=i))
+ yhat[employee$cv==i] = predict(fit, employee[employee$cv==i,])
+ }
> mean((employee$salaryK-yhat)^2)  # test MSE
[1] 83.52445

> fit = lm(salaryK~ageyrs+expyrs, employee)
> mean(fit$residuals^2)  # compare with MSE from all data
[1] 76.01949
```

Choosing a Set of Good Predictors

- Ideally you will have domain knowledge (e.g., theory) to tell you what predictors to use.
- Which predictors should we use if we don't have a strong theory? (This is often the case with models where prediction is primary objective.)
- Model fit (SSE and deviance) — Adding predictors will ...
  - always improve SSE on *estimation* sample
  - not necessarily improve SSE on *validation* data — SSE will increase if we model idiosyncrasies of estimation sample
- Reasonable solutions:
  - **Selection**: Iterative model selection procedures, e.g., forward, backward, stepwise, lasso
  - **Shrinkage/Regularization**: Shrinkage estimation, e.g., ridge, lasso or dimensionality reduction models (e.g., PCR, PLS, EFA, CFA)

Iterative Model Selection

Suppose we have a large number of predictor variables and we want to build a parsimonious model *with good predictive accuracy*. Using these methods is questionable when interpretation is the goal. Three commonly used approaches are:

- **Forward selection**
  1. Begin with no variables
  2. Add variable that yields greatest significant[9] improvement in SSE
  3. Repeat (2) until no significant improvement in SSE
- **Backward elimination**
  1. Begin with all candidate variables
  2. Drop variable that causes smallest non-significant[10] increase in SSE
  3. Repeat (2) until dropping variable causes significant increase in SSE
- **Stepwise selection**
  1. (Usually) begin with no variables
  2. Drop variable that causes smallest *non-significant* increase in SSE
  3. Add variable that yields greatest significant improvement in SSE
  4. Repeat (2) and (3) until no improvement in SSE

Notes:

- These can be very slow compared with ridge/lasso
- Instead of significance tests, R uses AIC
- Treat these methods as exploratory

[9] The user sets a "significance level to enter" the model.
[10] The user sets a "significance level to stay" in the model

Stepwise Regression: Click Ball Point Pens

```
> click$fair = as.numeric(click$eff==2)
> click$good = as.numeric(click$eff==3)
> click$outstand = as.numeric(click$eff==4)
> fit = lm(sales~1, click)
> fit2 = step(fit, scope=~ad+reps+fair+good+outstand, test="F")
Start:  AIC=386.52
sales ~ 1

           Df Sum of Sq    RSS    AIC  F value     Pr(F)
+ reps      1    465161 133092 328.40 132.8114 5.739e-14 ***
+ ad        1    463451 134802 328.91 130.6445 7.327e-14 ***
<none>                  598253 386.52
+ good      1     24847 573406 386.82   1.6466    0.2072
+ fair      1      9076 589177 387.90   0.5854    0.4489
+ outstand  1      6008 592245 388.11   0.3855    0.5384

Step:  AIC=328.4
sales ~ reps

           Df Sum of Sq    RSS    AIC  F value     Pr(F)
+ ad        1     57617  75475 307.71  28.2458 5.317e-06 ***
<none>                  133092 328.40
+ outstand  1      6008 127084 328.55   1.7492    0.1941
+ fair      1      2160 130932 329.74   0.6104    0.4396
+ good      1      2086 131006 329.76   0.5891    0.4476
- reps      1    465161 598253 386.52 132.8114 5.739e-14 ***

Step:  AIC=307.71
sales ~ reps + ad

           Df Sum of Sq    RSS    AIC F value     Pr(F)
<none>                  75475 307.71
+ outstand  1      3273  72202 307.93  1.6317    0.2096
+ fair      1      1289  74185 309.02  0.6257    0.4341
+ good      1         5  75470 309.70  0.0022    0.9626
- ad        1     57617 133092 328.40 28.2458 5.317e-06 ***
- reps      1     59327 134802 328.91 29.0842 4.167e-06 ***
```

Backward Selection: Click Ball Point Pens

```
> fit = lm(sales ~ ad+reps+fair+good+outstand, click)
> step(fit)
Start:  AIC=311.27
sales ~ ad + reps + fair + good + outstand

            Df Sum of Sq    RSS    AIC
- fair       1       229  71247 309.40
- good       1       998  72016 309.83
- outstand   1      2857  73875 310.85
<none>                    71018 311.27
- ad         1     41227 112245 327.58
- reps       1     54607 125625 332.09

Step:  AIC=309.4
sales ~ ad + reps + good + outstand

            Df Sum of Sq    RSS    AIC
- good       1       955  72202 307.93
<none>                    71247 309.40
- outstand   1      4223  75470 309.70
- ad         1     47039 118285 327.68
- reps       1     56521 127768 330.76

Step:  AIC=307.93
sales ~ ad + reps + outstand

            Df Sum of Sq    RSS    AIC
- outstand   1      3273  75475 307.71
<none>                    72202 307.93
- ad         1     54882 127084 328.55
- reps       1     60928 133129 330.41

Step:  AIC=307.71
sales ~ ad + reps

       Df Sum of Sq    RSS    AIC
<none>               75475 307.71
- ad    1     57617 133092 328.40
- reps  1     59327 134802 328.91

Call:
lm(formula = sales ~ ad + reps, data = click)

Coefficients:
(Intercept)           ad         reps
      69.33        14.16        37.53
```

Stepwise: quality control

```
> fit = lm(defect~., quality)
> step(fit)
Start:  AIC=123.01
defect ~ temp + density + rate + am

          Df Sum of Sq    RSS    AIC
- am       1    14.808 1312.1 121.35
- rate     1    19.847 1317.2 121.46
<none>                 1297.3 123.00
- density  1    90.862 1388.2 123.04
- temp     1   117.868 1415.2 123.61

Step:  AIC=121.35
defect ~ temp + density + rate

          Df Sum of Sq    RSS    AIC
- rate     1    40.591 1352.7 120.26
- density  1    76.366 1388.5 121.04
<none>                 1312.1 121.35
- temp     1   188.919 1501.0 123.38

Step:  AIC=120.26
defect ~ temp + density

          Df Sum of Sq    RSS    AIC
<none>                 1352.7 120.26
- density  1    142.69 1495.4 121.27
- temp     1    258.24 1611.0 123.50

Call:
lm(formula = defect ~ temp + density, data = quality)

Coefficients:
(Intercept)         temp      density
     46.256       18.049       -2.329
```

Shrinkage Estimation

- Consider estimate $\hat{\beta}$ of $\boldsymbol{\beta}$
- The **bias** of $\hat{\beta}$ is

$$\text{bias}(\hat{\beta}) = \mathbb{E}(\hat{\beta}) - \boldsymbol{\beta}$$

  When $\mathbb{E}(\hat{\beta}) = \boldsymbol{\beta}$, we say that $\hat{\beta}$ is an **unbiased** estimate of $\boldsymbol{\beta}$.

- The **mean-squared error** of $\hat{\beta}$ is

$$\begin{aligned} \text{MSE}(\hat{\beta}) &= E[(\hat{\beta} - \boldsymbol{\beta})^\mathsf{T}(\hat{\beta} - \boldsymbol{\beta})] \\ &= \text{trace}[\mathbb{V}(\hat{\beta})] + \text{bias}(\hat{\beta})^\mathsf{T}\text{bias}(\hat{\beta}) \\ &= \text{variance} + \text{bias}^2 \end{aligned}$$

- The OLS estimate $\hat{\beta}$ is unbiased and therefore has

$$\text{MSE}(\hat{\beta}) = \sigma^2(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$$

  The Gauss-Markov Theorem tells us that the OLS estimates are BLUE, and thus have the smallest MSE among unbiased estimates.

- Strategy of shrinkage estimation: introduce a bias that reduces the variance to give an estimate with lower overall mean squared error
- My IEMS303 lecture on bias and variance with math.
- My IEMS303 lecture on MSE with math

Ridge Regression

- An alternative way to reduce the impact of any single predictor variable is to include a penalty term in the least-squares objective function

$$\hat{\beta}_\lambda = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left[\sum_{i=1}^{k} (y_i - \beta_0 - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right] \tag{3.41}$$

$$\hat{\beta}_\lambda = (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} \tag{3.44}$$

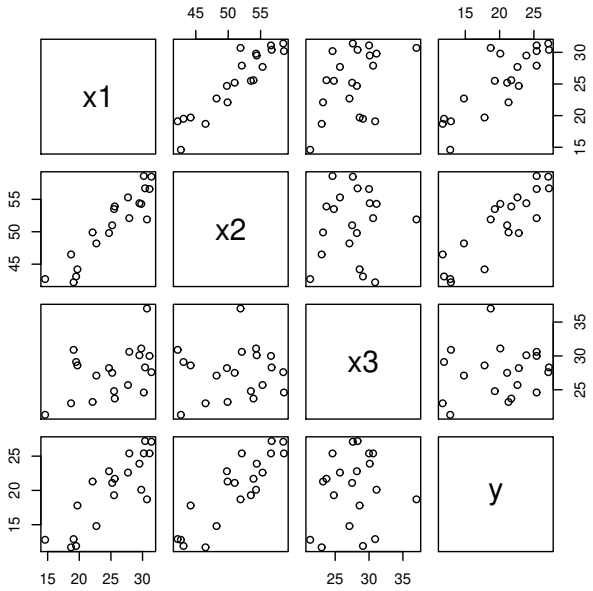
- $\lambda \geq 0$ is a constant, which determines how much to penalize large regression coefficients — when $\lambda = 0$ we get OLS and when λ is big the penalty is great and the coefficients will be close to 0
- Ridge existence theorem: there exist values of λ so that $\hat{\beta}_\lambda$ has smaller mean squared error than OLS estimates of $\boldsymbol{\beta}$
- Simulations have shown that ridge regression produces $\hat{y}$ values that are closer to the true values than PCR and stepwise regression (See Frank and Friedman, 1993).
- Scaling of X variables matters—standardize when units are incommensurate (done by default by `glmnet`)
- Criticisms of ridge regression:
  - The optimal value of λ depends on β and σ^2 (error variance), which are being estimated by the regression
  - Lack of theoretical justification for particular penalty term—why unweighted sum of squares?

Body Fat Example: Ridge Regression in R

Consider Body Fat example from Kutner (Table 7.1), `y` = body fat, `x1` = triceps skinfold thickness, `x2` = thigh circumference, and `x3` = midarm circumference.

```
bodyfat = data.frame(
  x1=c(19.5,24.7,30.7,29.8,19.1,25.6,31.4,
    27.9,22.1,25.5,31.1,30.4,18.7,19.7,
    14.6,29.5,27.7,30.2,22.7,25.2),
  x2=c(43.1,49.8,51.9,54.3,42.2,53.9,58.5,
    52.1,49.9,53.5,56.6,56.7,46.5,44.2,
    42.7,54.4,55.3,58.6,48.2,51.0),
  x3=c(29.1,28.2,37.0,31.1,30.9,23.7,27.6,
    30.6,23.2,24.8,30.0,28.3,23.0,28.6,
    21.3,30.1,25.7,24.6,27.1,27.5),
  y=c(11.9,22.8,18.7,20.1,12.9,21.7,27.1,
    25.4,21.3,19.3,25.4,27.2,11.7,17.8,
    12.8,23.9,22.6,25.4,14.8,21.1)
)
```

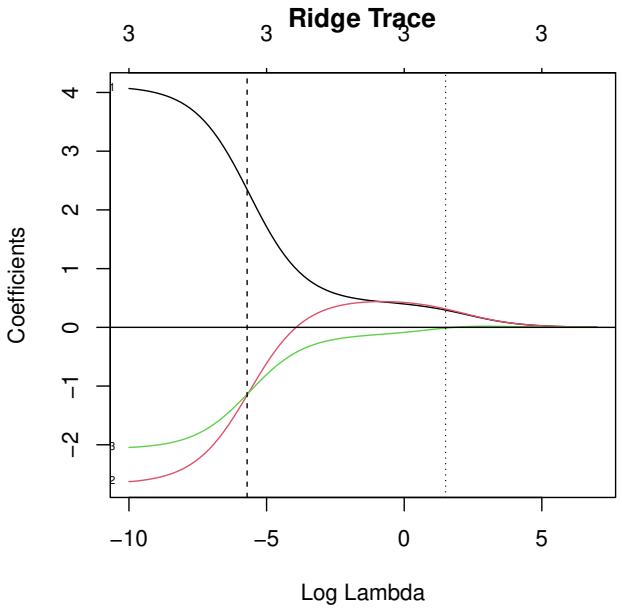
```
> plot(bodyfat)
> round(cor(bodyfat), 2)
     x1   x2   x3    y
x1 1.00 0.92 0.46 0.84
x2 0.92 1.00 0.08 0.88
x3 0.46 0.08 1.00 0.14
y  0.84 0.88 0.14 1.00

> fit.lm = lm(y~x1+x2+x3, bodyfat)
> coef(fit.lm)   # sign flips on x2, x3
> summary(fit.lm)
      Estimate Std Err t-value Pr(>|t|)
(Int)  117.085  99.782   1.173    0.258
x1       4.334   3.016   1.437    0.170
x2      -2.857   2.582  -1.106    0.285
x3      -2.186   1.595  -1.370    0.190
Multiple R-squared:  0.8014
F-stat: 21.52 on 3, 16 DF,  P=7.343e-06

> vif(fit.lm)
      x1       x2       x3
708.8429 564.3434 104.6060
```



- $\text{Corr}(y, x_j) > 0, \forall j$
- F is highly sig, but none of the t tests are sig. R^2 pretty big
- Sign flips for x_2 and x_3
- $\text{Corr}(x_j, x_{j'})$ modest (0.08) to large (0.92), but not very large. VIFs are giant!

Ridge Regression in glmnet

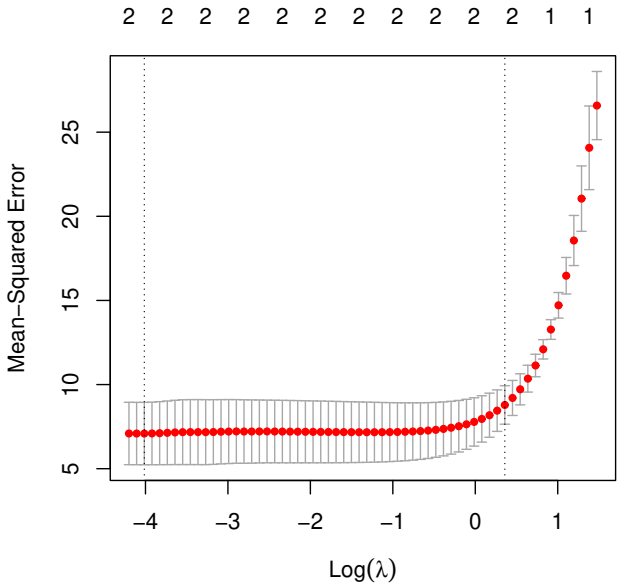
```
> # glmnet requires an X matrix:
> x = model.matrix(y ~ .-1, bodyfat)
> # -1 drops the intercept
> head(x)
    x1   x2   x3
1 19.5 43.1 29.1
2 24.7 49.8 28.2
3 30.7 51.9 37.0
4 29.8 54.3 31.1
5 19.1 42.2 30.9
6 25.6 53.9 23.7

> # we usually let glmnet pick lambda
> # I pick lambdas for class example
> lam = exp(seq(-10, 7, length=100))

> # alpha=0 specifies ridge
> # alpha=1 (default) lasso
> # we usually don't specify lambda=
> fit=glmnet(x, bodyfat$y, alpha=0,
    lambda=lam)

> # view ridge trace
> # label=T shows var nums on left
> plot(fit, xvar="lambda", label=T)
> abline(h=0) # add horizontal 0 line
> title("Ridge Trace")

> round(cbind(lambda=fit$lambda,
  loglambda=log(fit$lambda), t(fit$beta)),2)
100 x 5 sparse Matrix of class "dgCMatrix"
     lambda loglambda   x1    x2    x3
s0  1096.63       7.00 0.00  0.00  0.00
s1   923.60       6.83 0.00  0.00  0.00
s2   777.87       6.66 0.01  0.01  0.00
s3   655.14       6.48 0.01  0.01  0.00
...
```



- Each line shows the value of one slope as λ increases
- The lines "gently" approach 0 as $\lambda \to \infty$
- We observe sign flips, where a slope changes sign
- Which λ should we pick? Use cross validation (vertial lines from next page)

Cross Validation in glmnet

```
> set.seed(12345) # for replicability
> fit2 = cv.glmnet(x, bodyfat$y, alpha=0,
  lambda=lam, nfolds=5)
> # default is 10 folds, usually OK

> names(fit2)
> # glmnet.fit gives fit on entire data

> # lambda.min gives value to min MSE
> (l=fit2$lambda.min);log(l)
[1] 0.003322391
[1] -5.707071
> abline(v=log(l),lty=2)
> # add to plot previous page

> # more conservative possible lambda val
> (l2=fit2$lambda.1se);log(l2)
[1] 4.504381
[1] 1.505051
> abline(v=log(l2), lty=3) # add to plot

> plot(fit2) # create plot to right

> # How to find coefficients?
> predict(fit2$glmnet.fit, s=l, type="coef")
                    s1
(Intercept) 51.412645
x1           2.346499
x2          -1.155780
x3          -1.138833
> predict(fit2$glmnet.fit, s=l2, type="coef")
                     s1
(Intercept) -2.92369522
x1           0.29384750
x2           0.31326050
x3          -0.01255089
```



Vertial lines show `lambda.min` and `lambda.1se` values and match those added to previous plot with `abline`

The Lasso

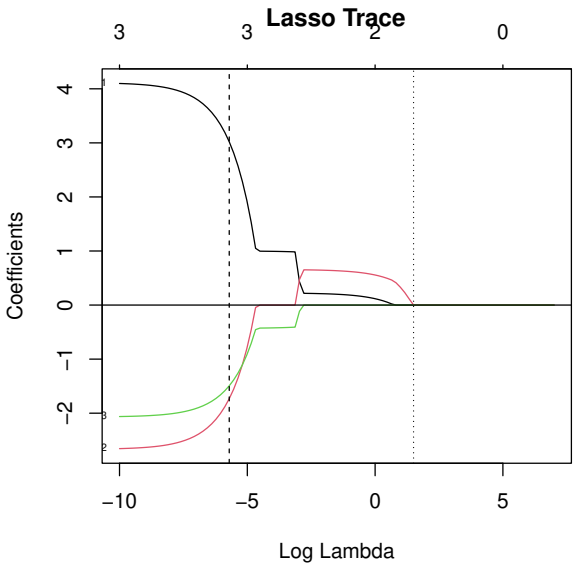
- Ridge penalizes $\sum_j \beta_j^2$, the **squared Euclidean length** (ℓ_2 norm) of the slope vector
- **Lasso** penalizes $\sum_j |\beta_j|$, the **taxi-cab length** (ℓ_1 norm) of the slope vector

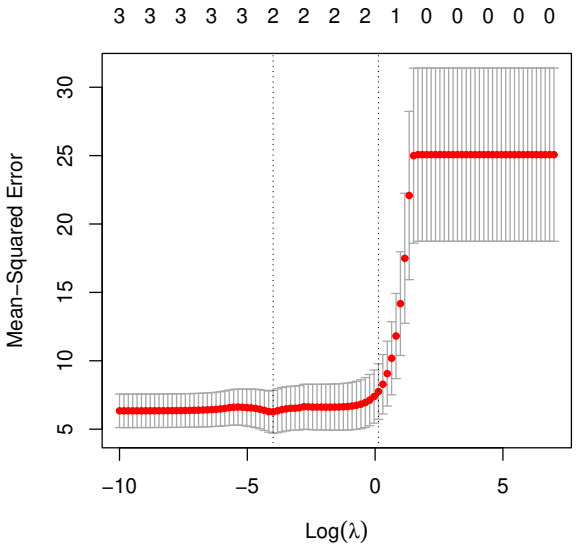
$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left[\sum_{i=1}^{k} (y_i - \beta_0 - \mathbf{x}_i^\mathsf{T} \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right]$$

- We can think of the methods as having different constraints:
  - Subset selection: $\min_{\boldsymbol{\beta}} \text{SSE}$ subject to $\sum_j \mathbb{1}(\beta_j \neq 0) \leq s$
  - Ridge: $\min_{\boldsymbol{\beta}} \text{SSE}$ subject to $\sum_j \beta_j^2 \leq s$
  - Lasso: $\min_{\boldsymbol{\beta}} \text{SSE}$ subject to $\sum_j |\beta_j| \leq s$
- The lasso and ridge regression shrink coefficients towards zero, but the lasso tends to force some coefficients to equal zero, similar to variable subset selection.
- Rule of thumb: use lasso if you think some variables should be dropped

Lasso Bodyfat in glmnet

```
> set.seed(12345)
> # I skip right to cv.glmnet
> # lasso default (alpha=0)
> fit.l1 = cv.glmnet(x, bodyfat$y,
  lambda=lam, nfolds=5)

> plot(fit.l1$glmnet.fit, xvar="lambda",
   label=T); abline(h=0)
> title("Lasso Trace")

> (l=fit2$lambda.min);log(l)
[1] 0.01805755
[1] -4.014191
> abline(v=log(l),lty=2)

> (l=fit2$lambda.1se);log(l)
[1] 1.43103
[1] 0.3583945
> abline(v=log(l), lty=3)

> round(cbind(
  lambda=fit.l1$glmnet.fit$lambda,
  loglambda=log(fit.l1$glmnet.fit$lambda),
    t(fit.l1$glmnet.fit$beta)), 2)

     lambda loglambda   x1    x2    x3
s0  1096.63      7.00 .     .     .
...
s32    4.50      1.51 .     .     .
s33    3.79      1.33 .     0.11  .
...
s37    1.91      0.65 0.02  0.46  .
...
s57    0.06     -2.79 0.22  0.65  .
s58    0.05     -2.96 0.43  0.47 -0.11
s59    0.04     -3.13 0.98  .    -0.41
...
s67    0.01     -4.51 1.00  .    -0.43
s68    0.01     -4.68 1.05 -0.05 -0.45
...
s99    0.00    -10.00 4.10 -2.66 -2.06
```





In lasso trace the lines "nosedive" into the 0 line (selection)

College Case JWHT Problem 6.9

This problem is a variation of 6.9 in JWHT on page 263. It uses the College data. The labels are on page 54.

1. Read the data into R, fix the row names, and create a test set.
2. How many cases are assigned to training? Test?
3. The dependent variable is `Apps`. Generate a histogram. Comment.
4. Regress `Apps` on all other variables using only the training data. Examine the residual plot and comment.
5. Replace `Apps` with its square root as a variance stabilizing transformation.
6. *Full model.* Do problem 6.9b. Note that you are using the square root of apps as the dependent variable and in your evaluation on the test set. For test set error, report the mean. Examine the residual plot and comment. What are the power predictors?
7. *Step model.* Apply the `step` function to the fitted "full" model from the previous part and report the test set error. What are the "power" predictors and signs?
8. *Ridge model.* Do problem 6.9c. Plot the ridge trace, and the fitted `cv.glmnet` object showing cross-validated MSE against λ. What is the optimal value of λ?
9. *Lasso model.* Do problem 6.9d. Report the same things as with ridge.
10. *Improved model.* Examine a scatterplot matrix for the training data. Suggest suitable transformations for predictor variables as necessary (continue to use the square root of apps as the dependent variable). Add these transformations to your model and see if the test-set MSE improves. Report which transformations end up improving your model, your preferred method of estimation (e.g., OLS, step, ridge, lasso), and the final test set error.

Dimensionality reduction notation

- n number of observations, indexed by i (row/sampling unit)
- p number of variables, indexed by j and sometimes k (column)
- $\mathbf{X}(n \times p)$ observed data
  - $\mathbf{x}_i$ is a $(p \times 1)$ row vector
  - $\bar{x}_j$ mean of column j of $\mathbf{X}$
  - s_j^2 sample variance of column j of $\mathbf{X}$
  - $\tilde{\mathbf{X}}(n \times p)$ column mean-centered version of $\mathbf{X}$
  - $\mathbf{Z}(n \times p)$ column standardized version of $\mathbf{X}$
  - At times we will take $\mathbf{x}(p \times 1)$ to be a random vector with $(p \times 1)$ population mean p-vector $\mathbb{E}(\mathbf{x}) = \boldsymbol{\mu}$ and $(p \times p)$ population variance matrix $\mathbb{V}(\mathbf{x}) = \boldsymbol{\Sigma}$.
- $\mathbf{C} = (c_{jk})$ sample covariance matrix of $\mathbf{X}$
- $\mathbf{R} = (r_{jk})$ sample correlation matrix of $\mathbf{X}$
- $m \leq p$ dimension of (usually) lower-dimensional space
- $\mathbf{Y}(n \times m)$ **scores** or **factors**
- $\mathbf{U}(p \times m)$ matrix of (column) basis vectors, $\mathbf{u}$ is a p-vector
- $\mathbf{V}(p \times p)$ matrix of eigenvectors, where column $\mathbf{v}_j$ is eigenvector j
  - λ_j eigenvalue corresponding to $\mathbf{v}_j$. $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_p)$
  - d_j singular value with $\mathbf{D} = \text{diag}(d_1, \ldots, d_m)$
  - $\mathbf{L}(p \times m)$ matrix of loading vectors (factor analysis)
- $\mathbf{Q}$ rotation/orthonormal matrix

Numerical latent variables/derived variables

- We observe p variables on n sampling units and record values in $n \times p$ matrix $\mathbf{X} = (x_{ij})$: sampling unit i variable j and row vector $\mathbf{x}_i(p \times 1)$.
- Our goal is to extract an m-dimensional ($m < p$) set of **scores** or **factors Y** ($n \times m$), with row vectors $\mathbf{y}_i(m \times 1)$.
- *Principal components analysis* (PCA) and *factor analysis* (FA) find a linear transformation that maps $\mathbf{X}$ to/from $\mathbf{Y}$
- Applications (list not mutually exclusive)
  - *Latent variables* are variables ($\mathbf{y}$) that cannot be measured directly and can only be studied/measured through *manifestations* ($\mathbf{x}$), e.g., psychometric variables attitudes and IQs
  - Pragmatic approach to analyzing many of variables, e.g., we might use them to avoid problems with multicollinearity when building cluster and regression models.
  - Data visualization
  - Measurement error
  - Data compression

Algebra review: bases and coordinates

- A vector space $\mathcal{V}$ consists of all linear combinations of the vectors $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_p)$. These vectors *span* the space, i.e., every vector $\mathbf{x} \in \mathcal{V}$ can be written

$$\mathbf{x} = a_1\mathbf{u}_1 + \cdots + a_p\mathbf{u}_p$$

  for constants $a_1, \ldots, a_p$, called *coordinates* (and later *scores* or *factors*)

- A *basis* for a vector space is a set of vectors that (1) are linearly independent and (2) span the space.

- The *standard basis* for $\mathbb{R}^p$ is $\mathbf{I}_p = (\mathbf{e}_1, \ldots, \mathbf{e}_p)$, where $\mathbf{e}_1 = (1, 0, \ldots, 0)^\mathsf{T}, \ldots, \mathbf{e}_p = (0, \ldots, 0, 1)^\mathsf{T}$. Note that $\mathbf{x} = x_1\mathbf{e}_1 + \cdots + x_p\mathbf{e}_p$

- Example: let $\mathbf{x} = \binom{1}{7} \in \mathbb{R}^2$. Then $\mathbf{x} = 1\mathbf{e}_1 + 7\mathbf{e}_2$, but $\mathbf{I}_2$ is not the only basis. Consider

$$\mathbf{U} = \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{a} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \quad \text{so}$$

$$\mathbf{U}\mathbf{a} = 3\begin{pmatrix} 1 \\ 2 \end{pmatrix} + 1\begin{pmatrix} -2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 7 \end{pmatrix} = \mathbf{x}$$

  $\mathbf{a} = \binom{3}{1}$ are the coordinates WRT the $\mathbf{U}$ basis.

More basis terms

- Orthogonal basis vectors are called an *orthogonal basis*, and if each vector also has length 1 then an *orthonormal basis*, e.g.,

$$\mathbf{U}_1 = \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix}, \quad \mathbf{U}_2 = \frac{\mathbf{U}_1}{\sqrt{5}} = \begin{pmatrix} \frac{1}{\sqrt{5}} & \frac{-2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{pmatrix}, \quad \mathbf{U}_3 = \begin{pmatrix} 1 & 1 \\ 2 & -3 \end{pmatrix}$$

- $\mathbf{U}_1$ is *orthogonal* because $(1, 2)\binom{-2}{1} = 0$
- $\mathbf{U}_2$ is *orthonormal*: $\mathbf{U}_2^\mathsf{T}\mathbf{U}_2 = \mathbf{I}_2$
- $\mathbf{U}_3$ is not orthogonal because $(1, 2)\binom{1}{-3} = -5 \neq 0$ yet is still a basis. Continuing the example from the previous page with coordinates $\mathbf{a} = \binom{2}{-1}$,

$$\mathbf{U}_3\mathbf{a} = 2\begin{pmatrix} 1 \\ 2 \end{pmatrix} - 1\begin{pmatrix} 1 \\ -3 \end{pmatrix} = \begin{pmatrix} 1 \\ 7 \end{pmatrix} = \mathbf{x}$$

Compression with a different basis

- Points in a m-dimensional linear subspace of $\mathbb{R}^p$ ($m < p$) do not require a full set of p coordinates.
- If we know a basis for the subspace, then the p-dimensional points can be written (and "saved") as the m-dimensional coordinates (or "scores"), achieving compression.
- There are many different bases for the subspace, e.g.,

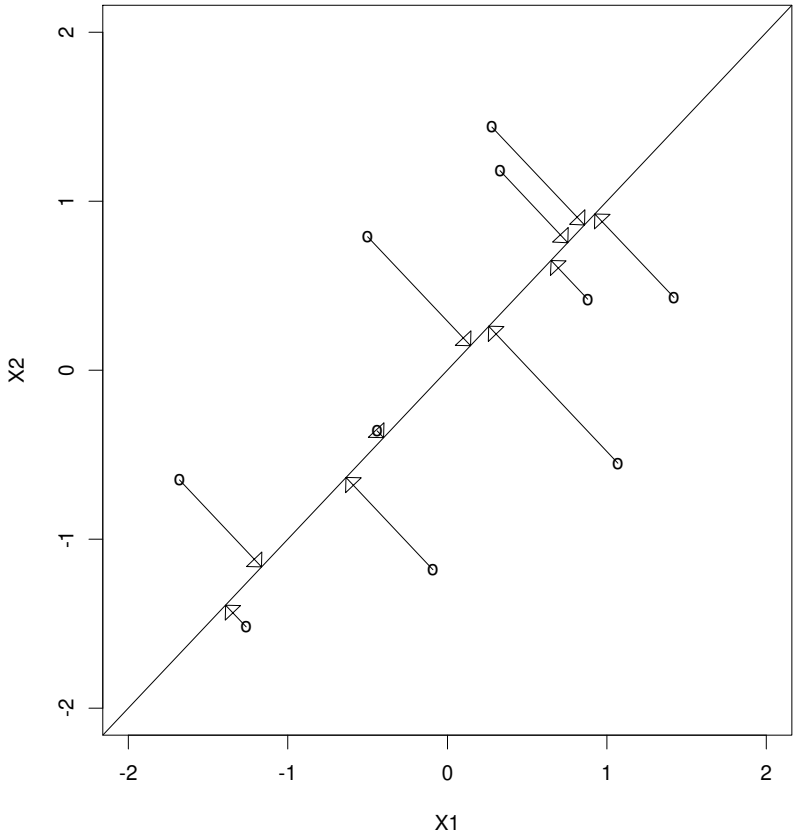
WRT $\mathbf{I}_2$		WRT $\mathbf{u}_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$	WRT $\mathbf{u}_2 = \frac{\mathbf{u}_1}{\sqrt{5}} = \begin{pmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{pmatrix}$
x_1	x_2	a_1	a_2
0	0	0	0
1	2	1	$\sqrt{5}$
2	4	2	$2\sqrt{5}$
3	6	3	$3\sqrt{5}$

- Instead of $p = 2$ coordinates per point, we only need $m = 1$.
- Note that this subspace *minimizes errors* and *maximizes variance* in the $\mathbf{u}_1$ direction (no variation in direction orthogonal to $\mathbf{u}_1$)
- Since $\mathbf{u}_2$ has unit length, a_2 gives the distance between the origin and the points.
- Note $a_2 = a_1\sqrt{5}$, illustrating how the "scores" are not unique. Any linear transformation of a_1 (or a_2) determines the points if we know the basis.

Points that approximately lie in a subspace

Consider the following $p = 2$ variables with $n = 10$ observations:

X_1	X_2	Factor1	PC1
0.88	0.42	0.76	−0.92
−1.68	−0.65	−1.37	1.65
1.42	0.43	1.09	−1.31
0.33	1.18	0.88	−1.07
−0.09	−1.18	−0.74	−.90
−0.50	0.79	0.17	−0.20
−1.26	−1.52	−1.64	1.97
1.07	−0.55	0.30	−0.37
−0.44	−0.36	−0.47	0.57
0.28	1.44	1.01	−1.22



Review: univariate projections

- Vector multiplication: let $\mathbf{x} = (x_1, x_2)^\mathsf{T}$ be an observation and $\mathbf{u} = (u_1, u_2)^\mathsf{T}$ be a (loading) vector. The product of $\mathbf{x}$ and $\mathbf{u}$ is

$$\mathbf{x}^\mathsf{T}\mathbf{u} = (x_1, x_2)\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = x_1u_1 + x_2u_2$$

- The projections of $\mathbf{x}$ onto $\mathbf{u}$ is the point on $\mathbf{u}$ that is closest to $\mathbf{x}$

$$\text{proj}_\mathbf{u}\mathbf{x} = \frac{\mathbf{x}^\mathsf{T}\mathbf{u}}{\mathbf{u}^\mathsf{T}\mathbf{u}}\mathbf{u}$$

  - The length of the projection of $\mathbf{x}$ onto $\mathbf{u}$ is

$$\frac{\mathbf{x}^\mathsf{T}\mathbf{u}}{\mathbf{u}^\mathsf{T}\mathbf{u}}$$

  - If $\mathbf{u}$ is selected to have unit length, i.e., $\mathbf{u}^\mathsf{T}\mathbf{u} = 1$, then $\mathbf{x}^\mathsf{T}\mathbf{u}$ is the length of the projection.

- **PC Score** y is proportional to the length of the projection of $\mathbf{x}$ onto $\mathbf{u}$

$$y \propto \mathbf{x}^\mathsf{T}\mathbf{u} = x_1u_1 + \cdots + x_pu_p,$$

  which looks like regression (except y unobserved), and we'll soon interpret "loading" u_j as the "effect" of x_j on y

Covariance and correlation in matrix form

- Let $\mathbf{X} = (x_{ij})$ be an $n \times p$ matrix where the rows are observations and the columns are variables.
- The **sample column means** and *variances* are

$$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \quad \text{and} \quad s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2$$

  Write $\overline{\mathbf{x}}^\mathsf{T} = (\bar{x}_1, \ldots, \bar{x}_p)$
- We will transform $\mathbf{X}$ before analysis with either
  - **Centered**: $\tilde{x}_{ij} = x_{ij} - \bar{x}_j$ (columns have mean 0 and unchanged variance), or in matrix form $\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{1}_n\overline{\mathbf{x}}^\mathsf{T}$, where $\mathbf{1}_n$ is an n-vector of 1's
  - **Standardized**: $z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}$ (columns have mean 0 and variance 1)
- The **sample covariance** between columns j and k is

$$c_{jk} = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$

  Note that $s_j^2 = c_{jj}$
- The $p \times p$ *sample covariance matrix* is

$$\mathbf{C} = (c_{jk}) = \frac{(\mathbf{X} - \mathbf{1}_n\overline{\mathbf{x}})^\mathsf{T}(\mathbf{X} - \mathbf{1}_n\overline{\mathbf{x}}^\mathsf{T})}{n-1} = \frac{\tilde{\mathbf{X}}^\mathsf{T}\tilde{\mathbf{X}}}{n-1}$$

- The sample correlation between columns j and k is

$$r_{ij} = \frac{c_{jk}}{s_j s_k} = \frac{c_{jk}}{\sqrt{c_{jj}c_{kk}}}$$

- The *correlation matrix* is

$$\mathbf{R} = (r_{jk}) = \frac{\mathbf{Z}^\mathsf{T}\mathbf{Z}}{n-1}$$

Key results for principal components analysis (PCA)

- Preprocessing: assume columns of $\mathbf{X}$ are mean centered (i.e., $\mathbf{X} \leftarrow \tilde{\mathbf{X}}$) or standardized (i.e., $\mathbf{X} \leftarrow \mathbf{Z}$).
- Note $\mathbf{X}^\mathsf{T}\mathbf{X}/(n-1)$ is the sample *covariance matrix* (or *correlation matrix* if columns of $\mathbf{X}$ are standardized)
- For first principal component, equivalent objective functions:

$$\min_{\mathbf{u}} \sum_{i=1}^{n} \|\mathbf{x}_i - \text{proj}_{\mathbf{u}}\mathbf{x}_i\|^2$$

$$\max_{\mathbf{u}} \sum_{i=1}^{n} \mathbb{V}(\mathbf{x}_i^\mathsf{T}\mathbf{u}), \qquad \text{where } \mathbf{u}^\mathsf{T}\mathbf{u} = 1$$

  Subsequent $\mathbf{u}$ vectors must be orthogonal to previous ones
- Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$ be the *eigenvalues* of $\mathbf{X}^\mathsf{T}\mathbf{X}/(n-1)$ with corresponding eigenvectors $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_p)$.
- Theorem: the solution is given by $\mathbf{V}$
  - $y_j = \mathbf{X}\mathbf{v}_j$ is the j^{th} *principal component* or PC *scores*
  - $\mathbf{v}_j$ is the direction of the j^{th} *principal axis*
  - $\lambda_j = \mathbb{V}(Y_j)$, i.e., the variance of column j scores
  - The sum of the variances[11] of $\mathbf{X}$ equals $\lambda_1 + \cdots + \lambda_p$ and the proportion of variation explained by the first $m \leq p$ PCs is

$$\frac{\lambda_1 + \cdots + \lambda_m}{\lambda_1 + \cdots + \lambda_p}$$

[11] For correlation matrices the sum is p: there are p variables each with variance 1.

Review: eigenvalues and vectors

- Let $\mathbf{A}$ be a $p \times p$ matrix. Then λ is an *eigenvalue* with corresponding *eigenvector* $\mathbf{v}$ iff $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$
- This definition is equivalent to $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$, which produces a polynomial of degree p in λ.
- By the fundamental theorem of algebra, the polynomial has p roots, counting multiplicities. Thus $\mathbf{A}$ has exactly p eigenvalues (they may not be distinct). We'll write

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_p \end{pmatrix} \quad \text{and} \quad \mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_p)$$

  WLOG, we order eigenvalues so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$
- All eigenvalues of a symmetric matrix (i.e., $\mathbf{A} = \mathbf{A}^\mathsf{T}$) are real.
- *Eigendecomposition theorem*: any symmetric[12] matrix can be written as $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\mathsf{T}$, where the matrix of eigenvectors $\mathbf{V}$ is column orthonormal ($\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$).
- Your turn: Find and interpret the eigenvalues and eigenvector of a 2×2 correlation matrix:

$$\mathbf{R} = \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}$$

[12] More generally, any *square* matrix $\mathbf{A}$ can be written $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$.

- Your turn: Find and interpret the eigenvalues and eigenvector of a 2×2 correlation matrix:

$$\mathbf{R} = \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}$$

- Solution: using the result from the course packet, λ is an eigenvalue iff $0 = \det(\mathbf{R} - \lambda\mathbf{I})$
- Substituting our $\mathbf{R}$ we get

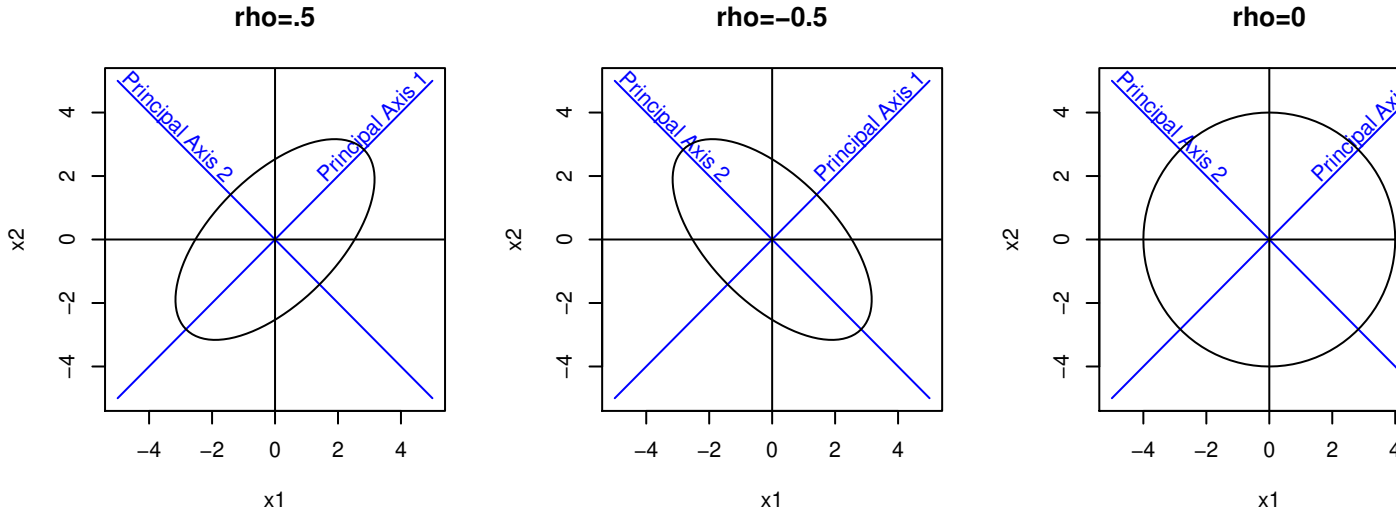
$$0 = \det \begin{pmatrix} 1-\lambda & r \\ r & 1-\lambda \end{pmatrix} = (1-\lambda)^2 - r^2 \Longrightarrow \lambda = 1 \pm r$$

- We can find the eigenvectors using the definition. Start with $\lambda = 1 + r$

$$\begin{aligned} \mathbf{R}\mathbf{v} &= \lambda\mathbf{v} \\ \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} &= (1+r) \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \\ \begin{pmatrix} v_1 + rv_2 \\ rv_1 + v_2 \end{pmatrix} &= \begin{pmatrix} v_1 + rv_1 \\ v_2 + rv_2 \end{pmatrix} \end{aligned}$$

  This is only true whe $v_1 = v_2$. The eigenvector associated with $\lambda = 1 + r$ is any mulitple of $(1, 1)^\mathsf{T}$, e.g., $(\sqrt{2}/2, \sqrt{2}/2)^\mathsf{T}$

- For $\lambda = 1 - r$ the corresponding eigenvector is any multiple of $(1, -1)^\mathsf{T}$ (check!)



$\lambda_1 = 1.5, \lambda_2 = 0.5$ $\lambda_1 = 0.5, \lambda_2 = 1.5$ $\lambda_1 = \lambda_2 = 1$

- The Eigendecomposition theorem tells us that $\mathbf{R} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\mathsf{T}$ for unit-length eigenvectors $\mathbf{V}$. We can confirm this:

$$\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\mathsf{T} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1+r & 0 \\ 0 & 1-r \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix} = \mathbf{R}$$

- Illustrating the SVD: $\mathbf{X}^\mathsf{T}\mathbf{X} = \mathbf{V}\mathbf{D}\mathbf{U}^\mathsf{T}\mathbf{U}\mathbf{D}\mathbf{V}^\mathsf{T} = \mathbf{V}\mathbf{D}^2\mathbf{V}^\mathsf{T}$

R: simple example

```
dat=data.frame(
  x1=c(0.88,-1.68,1.42,0.33,-0.09,-0.50,-1.26,1.07,-0.44,0.28),
  x2=c(0.42,-0.65,0.43,1.18,-1.18,0.79,-1.52,-0.55,-0.36,1.44) )
```

```
> fit = prcomp(dat, scale=T)
> plot(fit) # produce scree plot
> summary(fit)
Importance of components:
                          PC1   PC2
Standard deviation     1.205 0.741
Proportion of Variance 0.726 0.274
Cumulative Proportion  0.726 1.000

> fit$rotation
          PC1        PC2
x1 -0.7071068  0.7071068
x2 -0.7071068 -0.7071068

> fit$sdev^2
[1] 1.451397 0.548603

> fit$sdev^2/2
[1] 0.7256985 0.2743015

> fit$x
             PC1         PC2
 [1,] -0.9189160  0.32523643
 [2,]  1.6490537 -0.73026393
 [3,] -1.3081464  0.70033167
 [4,] -1.0668146 -0.60114212
 [5,]  0.8983799  0.76957686
 [6,] -0.2037793 -0.91290366
 [7,]  1.9666983  0.18185621
 [8,] -0.3678222  1.14525964
 [9,]  0.5665337 -0.05766558
[10,] -1.2151871 -0.82028552

> round(var(fit$x), 6)
         PC1      PC2
PC1 1.451397 0.000000
PC2 0.000000 0.548603
```

```
> # we could instead do it "by hand"
> cor(dat)
         x1       x2
x1 1.000000 0.451397
x2 0.451397 1.000000

> sum(diag(cor(dat))) # gives p=2
[1] 2

> fit2=eigen(cor(dat))
> fit2
$values
[1] 1.451397 0.548603

$vectors
          [,1]       [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068

> scale(dat) %*% fit$rotation
             PC1         PC2
 [1,] -0.9189160  0.32523643
 [2,]  1.6490537 -0.73026393
 [3,] -1.3081464  0.70033167
 [4,] -1.0668146 -0.60114212
 [5,]  0.8983799  0.76957686
 [6,] -0.2037793 -0.91290366
 [7,]  1.9666983  0.18185621
 [8,] -0.3678222  1.14525964
 [9,]  0.5665337 -0.05766558
[10,] -1.2151871 -0.82028552
```

Simple example with princomp and principal

```
> fit2 = princomp(dat, cor=T)
> fit2$sdev
   Comp.1    Comp.2
1.2047394 0.7406774
> fit2$loadings

Loadings:
   Comp.1 Comp.2
x1  0.707 -0.707
x2  0.707  0.707

> fit2$scores
          Comp.1      Comp.2
 [1,]  0.9686225 -0.34282930
 [2,] -1.7382552  0.76976577
 [3,]  1.3789074 -0.73821440
 [4,]  1.1245214  0.63365944
 [5,] -0.9469756 -0.81120524
 [6,]  0.2148022  0.96228495
 [7,] -2.0730820 -0.19169327
 [8,]  0.3877186 -1.20720966
 [9,] -0.5971790  0.06078486
[10,]  1.2809197  0.86465686

> round(var(fit2$scores), 6)
         Comp.1   Comp.2
Comp.1 1.612663 0.000000
Comp.2 0.000000 0.609559
```

```
> library(psych)
> fit3 = principal(dat, nfactor=2,
    rotate="none")
> fit3$loadings

Loadings:
   PC1    PC2
x1  0.852 -0.524
x2  0.852  0.524

                 PC1   PC2
SS loadings    1.451 0.549
Proportion Var 0.726 0.274
Cumulative Var 0.726 1.000
> fit3$scores
             PC1         PC2
 [1,]  0.7627508 -0.43910674
 [2,] -1.3688053  0.98594062
 [3,]  1.0858335 -0.94552861
 [4,]  0.8855149  0.81161127
 [5,] -0.7457048 -1.03901761
 [6,]  0.1691480  1.23252534
 [7,] -1.6324678 -0.24552687
 [8,]  0.3053126 -1.54623274
 [9,] -0.4702542  0.07785519
[10,]  1.0086722  1.10748016

> round(var(fit3$scores), 6)
    PC1 PC2
PC1   1   0
PC2   0   1
```

prcomp and princomp both return unit-length eigenvectors, but use different numerical methods. principal scales eigenvectors differently (so that the scores are standardized).

Comparison of solutions

- PCA, regardless of the implementation, finds a unique subspace for a solution, but there are many different bases for the subspace. Here are some possibilities:
  - $\mathbf{u}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$: sum of x_1 and x_2 ("summed score")
  - $\mathbf{u}_2 = \frac{\mathbf{u}_1}{\sqrt{2}}$: Solution from `princomp`, unit length so scores are distances
  - $\mathbf{u}_3 = -\frac{\mathbf{u}_1}{\sqrt{2}}$: R `prcomp`, unit length, scores distances
  - $\mathbf{u}_4 = \mathbf{u}_1/2$: mean of x_1 and x_2
  - $\mathbf{u}_5 = .58694\mathbf{u}_1$: `principal` "standardized scoring coefficients," produces factor scores with variance 1 (r=corr(x_1, x_2)=.45140. One can show that $.58694 = \sqrt{(1/(2(1+r)))}$
  - $\mathbf{u}_6 = .85188\mathbf{u}_1$: `principal` reported loadings, corr(y, x_j) = .85694 $= .58694(1+r)$.
- Here are some points and their coordinates WRT the different bases:

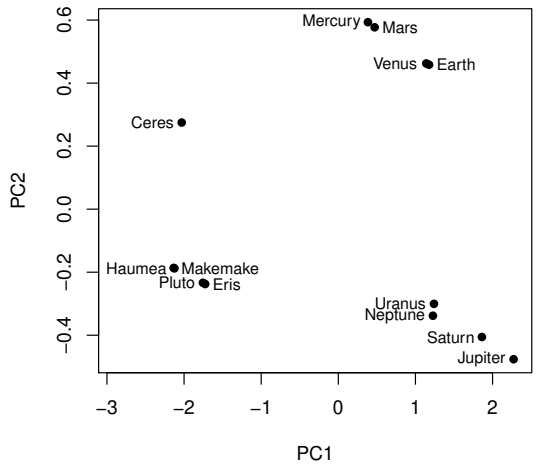
WRT $\mathbf{I}_2$		$\mathbf{Xu}_1$	$\mathbf{Xu}_2$	$\mathbf{Xu}_3$	$\mathbf{Xu}_4$	$\mathbf{Xu}_5$	$\mathbf{Xu}_6$
x_1	x_2	c_1	c_2	c_3	c_4	c_5	c_6
0	0	0	0	0	0	0	0
1	1	2	$\sqrt{2}$	$-\sqrt{2}$	1	.5869	.8519
2	2	4	$2\sqrt{2}$	$-2\sqrt{2}$	2	1.1739	1.7038
3	3	6	$3\sqrt{2}$	$-3\sqrt{2}$	3	1.761	2.556

Example: what's a planet?

Planetary Discriminants

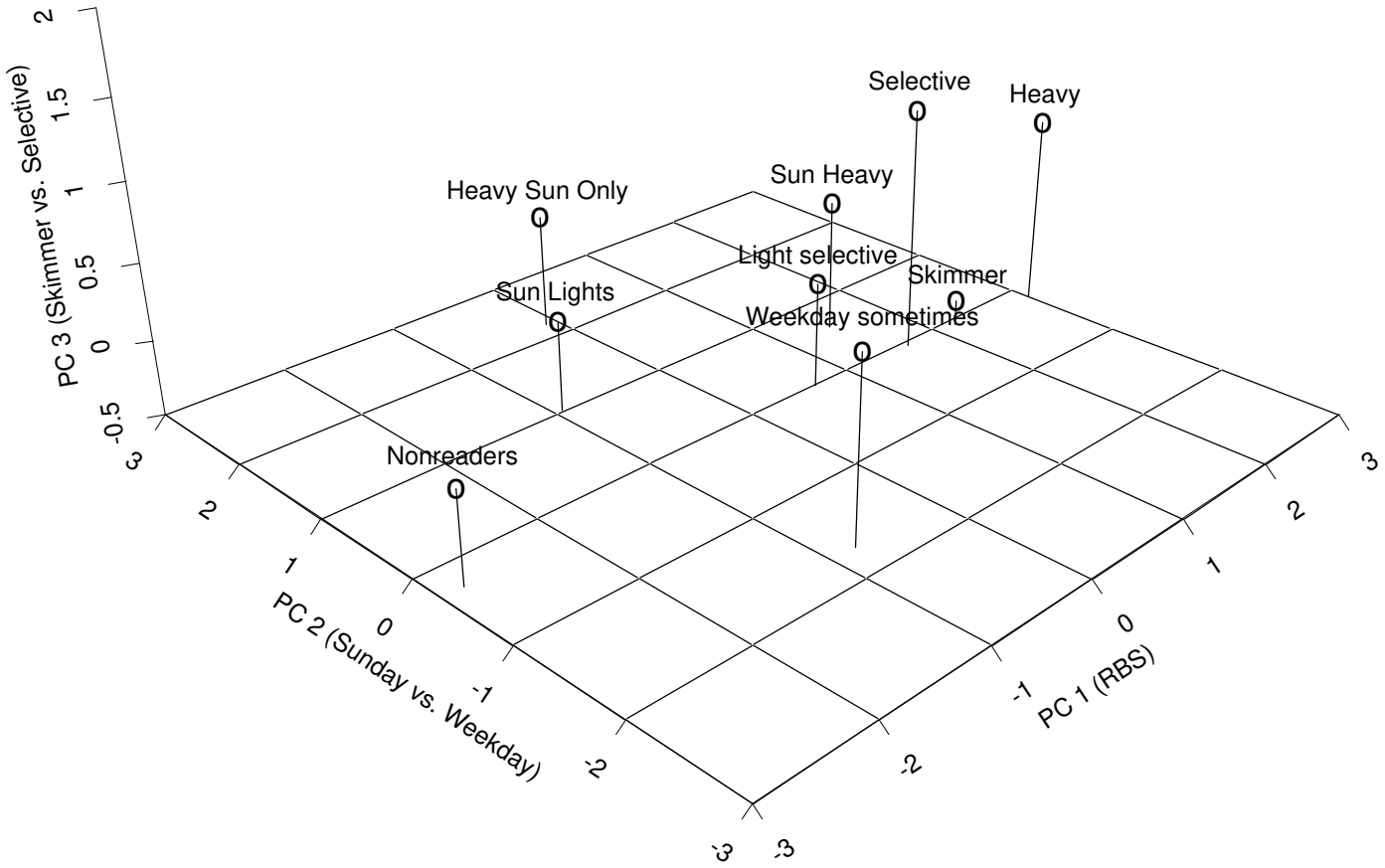
```
> planet = read.csv("planet.csv")
> planet
       Body      Mass   lambda       mu
1   Mercury   0.05500 1.95e+03 9.10e+04
2     Venus   0.81500 1.66e+05 1.35e+06
3     Earth   1.00000 1.53e+05 1.70e+06
4      Mars   0.10700 9.42e+02 1.80e+05
5     Ceres   0.00015 8.32e-04 3.30e-01
6   Jupiter 317.70000 1.30e+09 6.25e+05
7    Saturn  95.20000 4.68e+07 1.90e+05
8    Uranus  14.50000 3.85e+05 2.90e+04
9   Neptune  17.10000 2.73e+05 2.40e+04
10    Pluto   0.00220 2.95e-03 7.70e-02
11   Haumea   0.00067 2.68e-04 2.00e-02
12 Makemake   0.00067 2.22e-04 2.00e-02
13     Eris   0.00280 2.13e-03 1.00e-01

> planet$logmass = log(planet$Mass)
> planet$loglambda = log(planet$lambda)
> planet$logmu = log(planet$mu)

> round(cor(planet[,5:7]),4)
          logmass loglambda  logmu
logmass    1.0000    0.9693 0.8384
loglambda  0.9693    1.0000 0.9389
logmu      0.8384    0.9389 1.0000

> fit = prcomp(planet[,5:7])
> plot(fit)
```

```
summary(fit)
Importance of components:
                          PC1    PC2     PC3
Standard deviation     1.6829 0.40425 0.06590
Proportion of Variance 0.9441 0.05447 0.00145
Cumulative Proportion  0.9441 0.99855 1.00000


round(fit$rotation[,1:2], 4)
             PC1     PC2
logmass   0.5727 -0.6543
loglambda 0.5930 -0.0853
logmu     0.5660  0.7514

plot(fit$x[,1:2], xlim=c(-3.0,2.3), pch=16)
text(fit$x[c(-3,-4,-12,-13),1]-.1,
  fit$x[c(-3,-4,-12,-13),2],
  planet$Body[c(-3,-4,-12,-13)], adj=1, cex=.8)
text(fit$x[c(3,4,12,13),1]+.1,
  fit$x[c(3,4,12,13),2],
  planet$Body[c(3,4,12,13)], adj=0, cex=.8)
identify(fit$x[,1], fit$x[,2], planet$Body)
```



Example: cluster means for RBTs

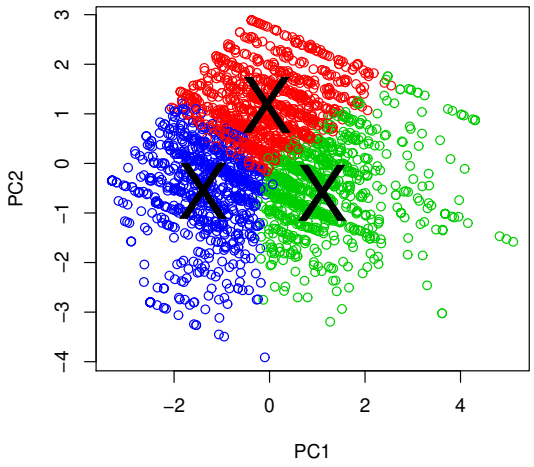
	Eigenvector		
Variable	1	2	3
Weekday time	0.41	−0.38	0.53
Sunday time	0.39	0.43	0.62
Weekday frequency	0.41	−0.45	−0.14
Sunday frequency	0.39	0.51	−0.24
Weekday completion	0.43	−0.35	−0.31
Sunday completion	0.42	0.28	−0.41
Eigenvalue	4.39	0.59	0.39
Percent variation	73.2	9.9	6.4
Cumulative variation	73.2	83.1	89.5



Example: theater revisited

```
> theater = read.csv("theater.csv")
> Ztheater = scale(theater[,1:5]) # standardize data

> set.seed(12345)  # not necessary, but assures that you get same starting values
> fit.th3 = kmeans(Ztheater, 3, nstart=100)
> fit = prcomp(Ztheater)
> summary(fit)
Importance of components:
                          PC1   PC2   PC3   PC4   PC5
Standard deviation     1.306 1.131 0.885 0.824 0.744
Proportion of Variance 0.341 0.256 0.157 0.136 0.111
Cumulative Proportion  0.341 0.597 0.754 0.889 1.000
```

```
> plot(fit)
> round(fit$rotation[,1:2],4)
             PC1     PC2
attitude -0.5926  0.2349
planning -0.2167  0.6663
parents  -0.4883  0.2363
goodval  -0.4079 -0.5102
getto    -0.4440 -0.4299

> mu=fit.th3$centers %*% fit$rotation[,1:2]
          PC1        PC2
1 -0.06046986  1.1722535
2  1.10116863 -0.6026269
3 -1.39502857 -0.5626771

> plot(fit$x[,1:2], col=fit.th3$cluster+1)
> points(mu, cex=4, pch="X")
```



Your turn

1. A naturalist studies grizzly bears with the goal of maintaining a healthy population. Measurements on $n = 61$ bears provided the following summary statistics:

Variable	Weight (kg)	Body length (cm)	Neck (cm)	Girth (cm)	Head length (cm)	Head width (cm)
Sample mean	95.52	164.38	55.69	93.39	17.98	31.13

```
S = matrix(c(      # Covariance matrix:
  3266.46, 1343.97, 731.54, 1175.50, 162.68, 238.37,
  1343.97,  721.91, 324.25,  537.35,  80.17, 117.73,
   731.54,  324.25, 179.28,  281.17,  39.15,  56.8,
   1175.5,  537.35, 281.17,  474.98,  63.73,  94.85,
   162.68,   80.17,  39.15,   63.73,   9.95,  13.88,
   238.37,  117.73,  56.8,    94.85,  13.88,  21.26
 ), nrow=6, byrow=T)
```

   (a) Perform a PCA using the covariance matrix. Report the eigenvalues and eigenvectors.

   (b) What fraction of variance is explained by the first principal component?

   (c) Interpret the first principal component. What does it tell you?

   (d) What fraction of variance is explained by the first two principal components together?

   (e) Interpret the second principal component. What does it tell you?

   (f) Compute the correlation matrix (hint: use `cov2cor`).

   (g) Perform a PCA using the correlation matrix. Report the fraction of variation accounted for by the first two components and interpret the first two.

   (h) Which solution should be used and why? Note that you do not need to know about grizzly bears to answer this question.

Solutions

1. Grizzly bears (a) See below for code. (b) 95.83%. (c) All loadings are positive so it is an average of all measurements, but the average is dominated by weight. Length, girth and neck also receive substantial weight. (d) .9583 + .0326 = .9909. (e) The second component contrasts weight with an average of the other five measures. (f) see below for code. (g) First two account for 97.01% of variance. The first PC is still an average, but the second contrasts body length and head measures with neck, girth and weight. (h) The variables have incommensurate units of measure and so we should use the correlation matrix.

```
> round(fit$values/sum(fit$values), 4)
[1] 0.9583 0.0326 0.0069 0.0017 0.0003 0.0001
```

```
> fit = eigen(bear) # part a)
> round(fit$values,4)
[1] 4478.8738 152.4662 32.3239 8.1168 1.5169 0.5425
> round(fit$vector,4)
       [,1]    [,2]    [,3]    [,4]    [,5]    [,6]
[1,] 0.8493  0.4708  0.2266 -0.0743  0.0087  0.0002
[2,] 0.3686 -0.8461  0.3681 -0.0128  0.1108  0.0191
[3,] 0.1941 -0.0581 -0.3031  0.9284  0.0123  0.0706
[4,] 0.3147 -0.2167 -0.8486 -0.3551  0.0824 -0.0327
[5,] 0.0439 -0.0604 -0.0018  0.0602 -0.4401 -0.8928
[6,] 0.0645 -0.0920 -0.0339 -0.0523 -0.8871  0.4433

> bearR = cov2cor(bear) # part f
> round(bearR, 4)
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,] 1.0000 0.8752 0.9559 0.9437 0.9024 0.9045
```

```
[2,] 0.8752 1.0000 0.9013 0.9177 0.9459 0.9503
[3,] 0.9559 0.9013 1.0000 0.9635 0.9269 0.9200
[4,] 0.9437 0.9177 0.9635 1.0000 0.9270 0.9439
[5,] 0.9024 0.9459 0.9269 0.9270 1.0000 0.9543
[6,] 0.9045 0.9503 0.9200 0.9439 0.9543 1.0000

> fit2 = eigen(bearR) # part g
> round(fit2$values,4)
[1] 5.6446 0.1758 0.0565 0.0492 0.0473 0.0266
> round(fit2$values/sum(fit2$values),4)
[1] 0.9408 0.0293 0.0094 0.0082 0.0079 0.0044
```

```
> round(cumsum(fit2$values)/sum(fit2$values),4)
[1] 0.9408 0.9701 0.9795 0.9877 0.9956 1.0000
> round(fit2$vectors,4)
       [,1]    [,2]    [,3]    [,4]    [,5]    [,6]
[1,] 0.4037 -0.5583 -0.2797 -0.2803  0.5930 -0.1300
[2,] 0.4043  0.5323  0.1962 -0.7169 -0.0237 -0.0144
[3,] 0.4100 -0.3893 -0.0379 -0.0618 -0.5610  0.6004
[4,] 0.4120 -0.2229  0.5760  0.2438 -0.2306 -0.5798
[5,] 0.4091  0.3190 -0.7015  0.2898 -0.2403 -0.3114
[6,] 0.4103  0.3194  0.2408  0.5101  0.4714  0.4350
```

Principal components regression (PCR)

- Process:
  1. Find PCs of independent variables (X)
  2. Regress the dependent variable on the selected PCs, i.e., those most correlated with dependent variable (since you have orthogonal predictors, you don't need stepwise)
- PCR is not a "bad" method, but there are better ones depending on the goal:
  - *Regularization.* It can be shown that PCR is a way to control model "complexity" and the bias-variance tradeoff (PCR is a special case of the generalized ridge estimator). Other methods such as ridge regression and the lasso are preferred. See HTF for discussion.
  - *Latent variables.* If the goal is to understand how the latent variable measured by the PCs affect the DV, then develop *reliable* and *valid* measures of the latent variables instead of PCs. Sometimes the PCs will be reliable and valid, but there are usually better measures (e.g., after possibly oblique rotations).
- Dimensionality reduction and PCA are fundamental approaches that all data scientists should know!

Logistic Regression

Suppose I want to predict a probability as a function of some independent variables, e.g., a yes-no response variable.

Linear regression problematic:

1. Probabilities are between 0 and 1; $\alpha + \beta x$ is unbounded
2. Residuals can take only two values — certainly not normally distributed
3. Variance of response, $\pi(1 - \pi)$, depends on the mean and is therefore heteroscedastic

Possible solutions:

1. Logistic regression
2. Discriminant analysis and naive/idiot Bayes classifiers

Outline of lecture:

1. The logistic regression model
2. Interpreting the results
3. Scoring other data sets

Logistic Regression Model

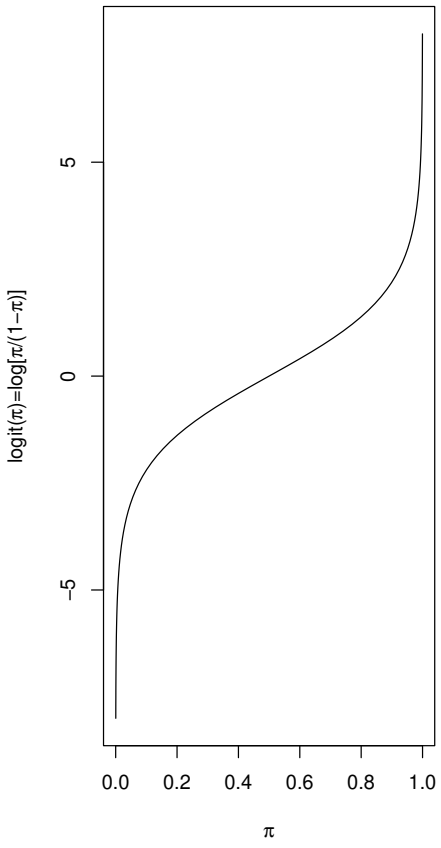
- Let $(x_1, y_1), \ldots, (x_n, y_n)$ be a random sample from some population where $y_i \in \{0, 1\}$
- Let $\pi_i = \mathbb{E}(Y_i)$, i.e., probability person i responds "yes"

- We want to model π_i, but can't
- Instead, model **log-odds** or **logit** of π_i
- Odds $= \pi/(1-\pi)$
- Logistic regression:

$$\begin{aligned} \text{logit}(\pi_i) &= \log\left(\frac{\pi_i}{1-\pi_i}\right) \\ &= \alpha + \beta x_i \end{aligned}$$

- Estimate α and β with maximum likelihood
- In general we have p predictors with

$$\text{logit}(\pi) = \alpha + \beta_1 x_1 + \cdots \beta_p x_p$$



Estimating Probabilities

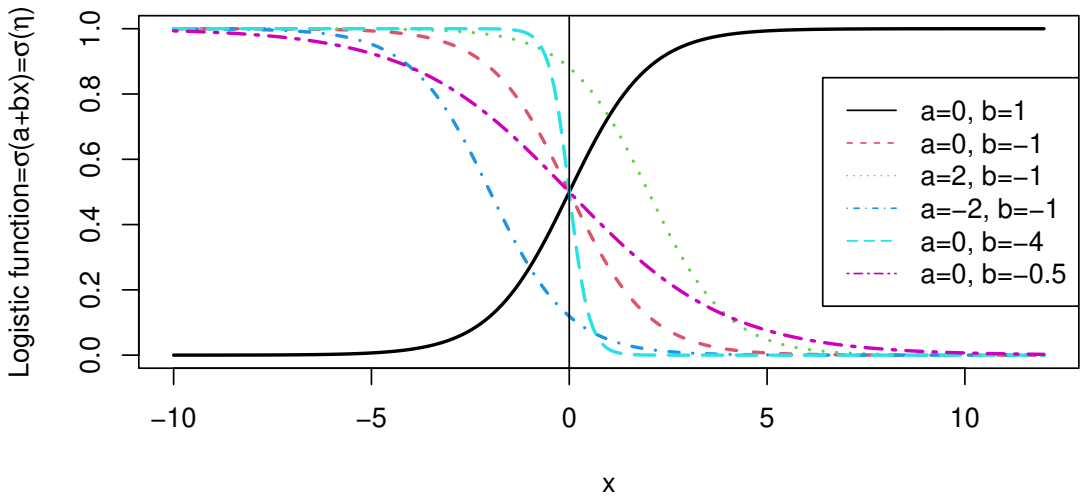
The logistic regression model is

$$\log\left(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right) = \alpha + \beta x_i$$

The **logit link** function is interesting, but we often need probabilities at the end of the analysis, i.e., what is the probability of response? Answer: solve the above equation for π. Let **linear predictor** $\eta_i = \alpha + \beta x_i$

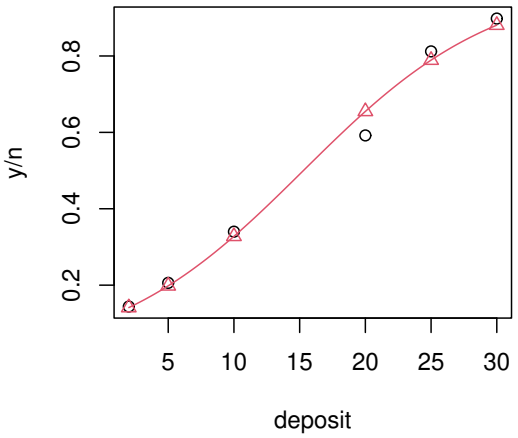
$$\frac{\hat{\pi}_i}{1-\hat{\pi}_i} = \exp(\eta_i)$$
$$\hat{\pi}_i = \exp(\eta_i) - \hat{\pi}_i \exp(\eta_i)$$
$$\hat{\pi}_i(1+\exp(\eta_i)) = \exp(\eta_i)$$
$$\hat{\pi}_i = \frac{\exp(\eta_i)}{(1+\exp(\eta_i))} = (1+\exp(-\eta_i))^{-1} \equiv \sigma(\eta_i)$$

This is the **logistic function** $\sigma(\eta)$. It's the most commonly used *squashing* or *sigmoidal* function used with neural networks



Bottle return problem

A carefully controlled experiment was conducted to study the effect of the size of the deposit on the likelihood that a returnable one-liter soft-drink bottle will be returned. A bottle return was coded 1 and no return was coded 0. The data show the number of bottles that were returned out of 500 sold at each of the six deposit levels. Plot estimated proportions against X. Estimate a logistic regression model and superimpose the fitted values. Interpret the parameter estimates.




```
bottle = data.frame(n=rep(500,6), deposit=c(2,5,10,20,25,30),y=c(72,103,170,296,406,449))
plot(y/n ~ deposit, data=bottle)
fit = glm(y/n ~ deposit, family=binomial, data=bottle, weight=n)
points(bottle$deposit, fit$fitted.values[1:6], pch=2, col=2)
x = seq(2,30,length=100)
lines(x, predict(fit, data.frame(deposit=x), type="response"), col=2)
summary(fit)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.076565   0.084839  -24.48   <2e-16 ***
x            0.135851   0.004772   28.47   <2e-16 ***
```

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \underbrace{-2.08 + 0.136x}_{\eta}$$

More bottle return problem

- Find a 95% confidence interval for β and test whether it is different from 0.

```
> confint(fit)
                 2.5 %     97.5 %
(Intercept) -2.2449682 -1.9123046
x            0.1266071  0.1453175
```

- What is the probability that a bottle will be returned when the deposit is 15 cents?

```
> eta = fit$coef[1] + fit$coef[2]*15  # linear predictor
> eta
-0.03880299
> predict(fit, data.frame(x=15))
-0.03880299

> 1/(1+exp(-(fit$coef[1] + fit$coef[2]*15))) # unlogit it
  0.4903005
> predict(fit, data.frame(x=15), type="response")
0.4903005
```

- For which deposit amount do you expect 75% of the bottles to be returned?

$$\log\left(\frac{.75}{1-.75}\right) = -2.08+0.136x \quad \Rightarrow \quad x = \frac{\log 3 + 2.08}{0.136} = 23.37$$

```
(log(3)-fit$coef[1])/fit$coef[2]
  23.37253
```

Logistic Regression Model

```
> fit = glm(2-q11 ~ food+atmosph+service, binomial, pizzarest)
> summary(fit)

Call:
glm(formula = 2-q11 ~ food + atmosph + service, family=binomial, data=pizzarest)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.40245    0.40647  -5.911 3.41e-09 ***
food         1.38776    0.09934  13.970  < 2e-16 ***
atmosph     -0.13490    0.10159  -1.328    0.184
service      0.39515    0.09903   3.990 6.60e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2345.0  on 14732  degrees of freedom
Residual deviance: 2026.2  on 14729  degrees of freedom
  (1653 observations deleted due to missingness)
AIC: 2034.2

> vif(fit)
    food  atmosph  service
1.336162 1.454803 1.389792

> summary(glm(2-q11 ~ atmosph, binomial, pizzarest))

Call: glm(formula = 2 - q11 ~ atmosph, family = binomial, data = pizzarest)

            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.48454    0.29534   5.027 4.99e-07 ***
atmosph      0.68116    0.07832   8.697  < 2e-16 ***
```

Odds Ratio: e^{β}

```
> fit$coef
(Intercept)        food     atmosph     service
 -2.4024497   1.3877596  -0.1349032   0.3951547

> exp(fit$coef)
(Intercept)        food     atmosph     service
  0.0904960   4.0058652   0.8738005   1.4846138
```

What does the odds ratio mean? Consider two people:

1. `food` $= 4$, `atmosph` $= 4$, `service` $= 4$
2. `food` $= 5$, `atmosph` $= 4$, `service` $= 4$

The odds for person 1 are ($\pi_1 =$ prob person 1 says "yes")

$$\frac{\pi_1}{1-\pi_1} = \exp(\alpha + 4\beta_1 + 4\beta_2 + 4\beta_3)$$

The odds for person 2 are ($\pi_2 =$ prob person 2 says "yes")

$$\begin{aligned}
\frac{\pi_2}{1-\pi_2} &= \exp(\alpha + 5\beta_1 + 4\beta_2 + 4\beta_3) \\
&= \exp(\alpha + 4\beta_1 + \beta_1 + 4\beta_2 + 4\beta_3) \\
&= \exp(\alpha + 4\beta_1 + 4\beta_2 + 4\beta_3)e^{\beta_1} \\
&= \frac{\pi_1}{1-\pi_1}e^{\beta_1}
\end{aligned}$$

Thus, by increasing `food` by 1, the odds of saying "yes" are multiplied by e^{β_1}

Generalized Linear Models

- We observe $(\mathbf{x}_i, y_i)$, $i = 1, \ldots n$ and $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\mathsf{T}$
- Classical linear model:
$$y_i = \boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i + \epsilon_i,$$
where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$
- Note that $\mathbb{E}(y_i) = \mu_i = \boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i$
- The generalized linear model (GLM) involves two functions
$$g(\mu_i) = \eta_i = \boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i$$
  - *Linear component*: $\eta_i = \boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i$ (η_i called **linear predictor**)
  - **Link function**: let univariate function g be monotonic and differentiable. The mean of Y is related to the linear predictor through the link function.
  - *Random component*: y_i are independent and from an exponential family, which implies the variance of y_i depends on μ_i through a variance function $\text{var}(y_i) = \phi\mathbb{V}(\mu_i)$ where ϕ is called the *dispersion parameter*.
- The classical linear model assumes $g(\mu) = \mu$, the identity function, and y_i has a normal distribution
- Logistic regression assumes $g(\mu) = \log[\mu/(1-\mu)]$ and y_i is a Bernoulli trial ($\mathbb{V}(y) = \mu(1-\mu)$). Other possible links for binary responses include
  - Probit $g(\mu) = \Phi^{-1}(\mu)$, where Φ is the cumulative standard normal distribution
  - Complementary log-log: $g(\mu) = \log(-\log(1-\mu))$
- Other frequently-used distributions for y include Poisson and Gamma, both typically use a log link

Pizza Hut Data

A random sample of $n = 220$ consumers were surveyed to evaluate the effect of price on the purchase of a pizza from Pizza Hut. Subjects were asked to suppose that they were going to have a large 2-topping pizza delivered to their residence. They were asked to select from either Pizza Hut or another pizzeria of their choice. The price they would have to pay to get a Pizza Hut pizza differed from survey to survey. The dependent variable is whether or not a student selected Pizza Hut and independent variables are price and sex.

1. Fit a logistic regression model.
2. Test whether the overall model is significant.
3. State the estimated regression equation.
4. Interpret the meaning of the coefficients and odds ratios.
5. Test whether each variable is significant.
6. Predict the probability that a female student will select Pizza Hut if the price is \$8.99. Repeat this for prices of \$11.49 and \$13.99.
7. Regress purchase on price for males only. Note the regression equation.
8. Regress purchase on price for females only. Note the regression equation.
9. Fit a logistic regression model with different slopes for males and females. Test whether the slopes are equal.

Maximum Likelihood Estimation

- Until now we've been using least squares to estimate parameters, e.g., regression

$$\min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2$$

- We've used SSE as the objective and to evaluate fit, e.g.,

$$\text{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Least squares won't work with other methods such as logistic regression or latent class analysis
- Alternative approach:
  - Maximum likelihood estimation (MLE)
  - (-2 log likelihood) is generalization of SSE
- Goals for today:
  - What is estimation by maximum likelihood?
  - What is -2 log likelihood?

MLEs of Proportions

- Suppose we draw a random sample of size $n = 5$ from some population, send them an offer, and $x = 2$ respond. What is our best guess of the response probability?
- Basic stats answer: $\hat{\pi} = 2/5 = .4$. Rationale: common sense
- Maximum likelihood answer: pick π to maximize the probability of observing 2 responses in 5 tries given π
  - Let $L(\pi)$ be the probability (likelihood) of observing the data if the probability of response is π

$$L(\pi) = \binom{5}{2}\pi^2(1-\pi)^3$$

  - Let $l(\pi) = \log L(\pi)$, called the *log-likelihood*

$$l(\pi) = \log(10) + 2\log(\pi) + 3\log(1-\pi)$$

$$\frac{dl(\pi)}{d\pi} = \frac{2}{\pi} - \frac{3}{1-\pi} = 0 \Longrightarrow \hat{\pi} = \frac{2}{5}$$

Guess (π)	$L(\pi)$	$l(\pi)$	$-2l(\pi)$
.3	.3087	−1.1754	2.3508
.35	.3364	−1.0894	2.1788
.4	.3456	−1.0625	2.1249
.45	.3369	−1.0879	2.1759
.5	.3125	−1.1632	2.3263

MLEs of Means

- Suppose we draw a random sample of size $n = 3$ from a normal population with unknown mean μ and known variance $\sigma^2 = 5$. The observed values are $x_1 = 4$, $x_2 = 5$, and $x_3 = 6$. What is the best guess of μ?
- Basic stats answer: $\bar{x} = (4 + 5 + 6)/3 = 5$. Rationale: common sense
- Maximum likelihood answer: pick μ so that the probability of observing the three values given μ is maximized

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x-\mu)^2\right]$$

$$L(\mu) = \prod_{i=1}^{3} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x_i-\mu)^2\right]$$

$$\begin{aligned} l(\mu) &= \sum_{i=1}^{3}\left[-\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}(x_i-\mu)^2\right] \\ &= -k_1 - k_2\sum_{i=1}^{3}(x_i-\mu)^2 \end{aligned}$$

$$\frac{dl(\mu)}{d\mu} = 2k_2\sum_{i=1}^{3}(x_i-\mu) = 0$$

$$\Longrightarrow \hat{\mu} = \frac{1}{3}\sum_{i=1}^{3} x_i$$

$$\text{Note } \frac{d^2 l(\mu)}{d\mu^2} = -6k_2 < 0$$

Likelihood Function for Logistic Regression

- Assume we have a random sample (observations independent, each have same probability of selection) and that we make two measurements on each observation (x_i, y_i), where y_i is a 0-1 variable and $i = 1, \ldots, n$
- Let $\pi_i = \mathsf{P}(y_i = 1)$, i.e., prob(person i says yes), and

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \alpha + \beta x_i$$

- Note that the probability distribution for person i is

$$f_i(y_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

- Since observations are independent, the probability distribution (likelihood) and log-likelihood for our sample is

$$f(y_1, \ldots, y_n) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

$$\begin{aligned}
\log(f) &= \sum_{i=1}^{n} [y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i)] \\
&= \sum_{i=1}^{n} \left[y_i \log(\frac{\pi_i}{1-\pi_i}) + \log(1-\pi_i) \right] \\
&= \sum_{i=1}^{n} y_i(\alpha + \beta x_i) - \sum_{i=1}^{n} \log[1 + \exp(\alpha + \beta x_i)]
\end{aligned}$$

We maximize this with respect to α and β

Log-likelihood, deviance, AIC

- Log-likelihood:

$$l = \log(L) = \sum_{i=1}^{n} y_i(\alpha + \beta x_i) - \sum_{i=1}^{n} \log[1 + \exp(\alpha + \beta x_i)]$$

- For bottle return problem:

```
> bot2 = data.frame(x=rep(bottle$deposit,2), y=c(rep(0,6), rep(1,6)),
  count=c(500-bottle$y, bottle$y))
> eta = fit$coef[1]+fit$coef[2]*bot2$x
> round(eta,2)
 [1] -1.80 -1.40 -0.72  0.64  1.32  2.00 -1.80 -1.40 -0.72  0.64  1.32  2.00
> fit$y
 1  2  3  4  5  6  7  8  9 10 11 12
 0  0  0  0  0  0  1  1  1  1  1  1
> sum(bot2$count*fit$y*eta) - sum(bot2$count*log(1+exp(eta)))
[1] -1531.436
> logLik(fit)
'log Lik.' -1531.436 (df=2)
```

- We will usually use the *deviance*: $-2l$. Think of deviance as SSE, measuring how much is unexplained by the model[13]

```
> -2*logLik(fit)
[1] 3062.872
> deviance(fit)
[1] 3062.872
```

- Or we will use AIC = -2log-likelihood + 2(number parameters), which penalizes the fit measure by $2(p+1)$ (R counts intercept as a parameter)

```
> AIC(fit)
[1] 3066.872
```

[13] Deviance is really $-2(l - l_s)$, where l is the log likelihood of the fitted model and l_s is the log likelihood of the saturated model, substituting y_i for π_i., For logistic regression $l_s = 0$.

Residual versus null deviance

```
> summary(fit)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.076565   0.084839  -24.48   <2e-16 ***
x            0.135851   0.004772   28.47   <2e-16 ***

    Null deviance: 4158.9  on 11  degrees of freedom
Residual deviance: 3062.9  on 10  degrees of freedom
AIC: 3066.9
```

- R reports AIC and the *residual deviance* for the *full model*, i.e., the one with all predictors in the model.
- It also reports the *null deviance*, which is the deviance of the intercept-only model:

  ```
  > fit.null = glm(y~1, bot2, family=binomial, weight=count)
  > deviance(fit.null)
  [1] 4158.862
  > fit$null.deviance
  [1] 4158.862
  ```

  Think of the null deviance as SST, measuring how much variation in Y is unexplained by the intercept model.
- The *difference* between them measures how much variation is explained by the model and plays the role of extra sums of squares.

  ```
  > fit$null.deviance - fit$deviance
  [1] 1095.99
  ```

- The *difference* has a chi-squared distribution and can test overall signifiance, H_0 : all $\beta_j = 0$.

  ```
  > 1-pchisq(fit$null.deviance - fit$deviance, 1)
  [1] 0
  ```

Likelihood-Ratio Test

- Consider the *full model* with $p + q$ predictors

$$\log\left(\frac{\pi}{1-\pi}\right) = \alpha+\beta_1 x_1+\cdots+\beta_p x_p+\beta_{p+1}x_{p+1}+\cdots+\beta_{p+q}x_{p+q}$$

- The *reduced model* has only p predictors, i.e., the last q predictors have been dropped.

$$\log\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta_1 x_1 + \cdots + \beta_p x_p$$

- Let D_1 be the deviance of the full model, and D_2 be the deviance of the reduced model.
- We can test $H_0 : \beta_{p+1} = \cdots = \beta_{p+q} = 0$ with the test statistic $D_2 - D_1$, which has a chi-square distributions with q degrees of freedom

Likelihood-Ratio Test For Single Parameters

How can we use the likelihood-ratio test to compare the fits of the two models:

$$\log\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta_1\texttt{food} + \beta_2\texttt{atmosph} + \beta_3\texttt{service}$$

$$\log\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta_1\texttt{food} + \beta_3\texttt{service}$$

i.e., does `atmosph` affect the response?

Answer:

- Let $-2l_3$ be the maximized log-likelihood for the three-predictor model (2026.183 from slide 161)
- Let $-2l_2$ be the maximized log-likelihood for the two-predictor model (2027.956)

Then $(-2l_2) - (-2l_3))$ has a chi-squared distribution with 1 degree of freedom.

```
> drop1(fit, test="Chisq")
Model:
2 - q11 ~ food + atmosph + service
        Df Deviance    AIC     LRT   Pr(Chi)
<none>       2026.2 2034.2
food     1   2207.7 2213.7 181.555 < 2.2e-16 ***
atmosph  1   2028.0 2034.0   1.773    0.1830
service  1   2041.6 2047.6  15.398  8.71e-05 ***

> d2=deviance(glm(2-q11~food+service,binomial,pizzarest,subset=(!is.na(atmosph))))
> d2
[1] 2027.956
> 1-pchisq(d2-deviance(fit),1)
[1] 0.1829724
```

LRT for Fitness Club Data

- Test whether payment type is significant, while controlling for log down payment and log use.

```
> fit = glm(default~log(downpmt+1)+log(use+1)+pmttype, binomial, default)
> summary(fit)
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        2.58488    0.08669   29.82   <2e-16 ***
log(downpmt + 1)  -0.69243    0.01822  -38.00   <2e-16 ***
log(use + 1)      -1.52831    0.04855  -31.48   <2e-16 ***
pmttypeStatement  -0.72340    0.05361  -13.49   <2e-16 ***
pmttypeCheck EFT  -3.99025    0.14181  -28.14   <2e-16 ***
pmttypeCredit EFT -2.94096    0.10685  -27.52   <2e-16 ***

    Null deviance: 17734  on 24842  degrees of freedom
Residual deviance: 11477  on 24837  degrees of freedom

> drop1(fit, test="Chisq")
Model:
default ~ log(downpmt + 1) + log(use + 1) + pmttype
                 Df Deviance   AIC    LRT   Pr(Chi)
<none>                11477 11489
log(downpmt + 1)  1    13236 13246 1758.9 < 2.2e-16 ***
log(use + 1)      1    12933 12943 1455.9 < 2.2e-16 ***
pmttype           3    14324 14330 2846.6 < 2.2e-16 ***
```

- Where does 2846.6 come from?

```
> fit2 = glm(default~log(downpmt+1)+log(use+1), binomial, default)
> deviance(fit2)-deviance(fit)
[1] 2846.583
> 1-pchisq(deviance(fit2)-deviance(fit), 3)
[1] 0
```

Predictive Accuracy of Classifiers

Suppose we estimate a binary response using p parameters

- GLMs minimize deviance, which has the same problems in detecting overfitting as SSE
- Alternative measures include the *classification rate*, which is the percentage of correctly classified cases, and the *misclassification rate*, which is 1− classification rate. These should be computed on non-training data.
- **Confusion matrix**

	Predicted bad	**Predicted good**
Actual Bad	TN=(# True Negatives)	FP=(# False Positives)
Actual Good	FN=(# False Negatives)	TP=(# True Positives)

$$\textbf{Classification rate} = \textbf{accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}}$$

- **Recall**: percent of relevant (good) items recommended, a.k.a. **true positive rate** (TPR) and **sensitivity**

$$\textbf{Recall} = \mathsf{P}(\text{predict good}|\text{actually good}) = \frac{\text{TP}}{\text{FN} + \text{TP}}$$

- **False positive rate**: percent of bad items recommended

$$\textbf{FPR} = \mathsf{P}(\text{predict good}|\text{actually bad}) = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

Decision-support metrics

- **Precision**: percent of recommended items relevant.

$$\textbf{Precision} = \mathsf{P}(\text{actually good}|\text{predict good}) = \frac{\text{TP}}{\text{FP} + \text{TP}}$$

- Additional complication: how do we classify observations? Let c be some cutoff and p_i be the predicted probably for observation i. Classify i as a "yes" when $p_i > c$ and "no" otherwise.
- The choice of c depends on misclassification costs. Depending on the situation, the "cost" of a false negative may be much greater than the cost of a false positive, e.g., airport security screening or disease detection versus spam filters.
- Examine precision and recall separately or their harmonic mean is the F_1 measure

$$F_1 = \frac{2}{1/R + 1/P} = 2\frac{P \times R}{P + R}$$

- Also common to plot precision versus recall varying c.
- *Receiver operating characteristic* (good article on ROC) curves plot TRP against FPR for different values of c. **AUC** is the area under the ROC curve.
  - AUC=0.5 means random guessing—model worthless
  - AUC=1 means "crystal-ball" perfect classification

Defaulting Customer Example

- Note that if we classify all observations as "no" then the classification rate is 88.3%. The classes are said to be highly **imbalanced**. This is a stupid classifier, but we must beat it!

```
> table(default$default)/length(default$default)
       0        1
0.882963 0.117037
```

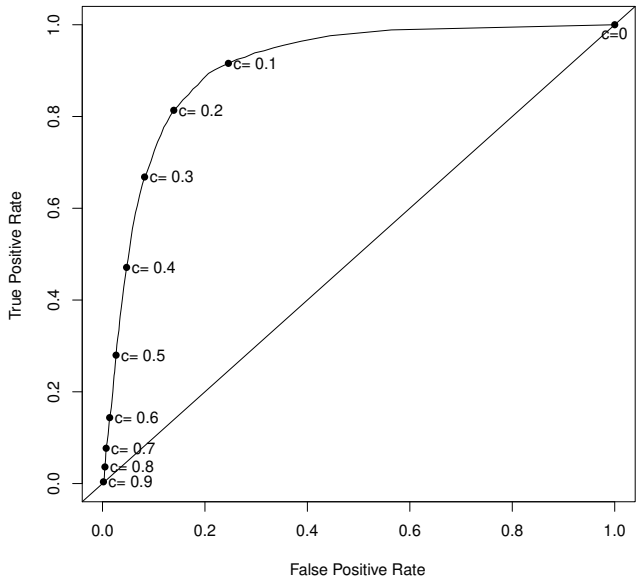
- Consider the following improved model

```
> fit = glm(default ~ log(downpmt+1)+pmttype+use+age+gender, binomial, default)
> tab=table(default$default, fit$fitted.values>.5)   # c=.5
> addmargins(tab)
      FALSE  TRUE   Sum
  0   21429   556 21985
  1    2107   751  2858
  Sum 23536  1307 24843

> sum(diag(tab))/sum(tab) # classification rate, (21468+818)/24975
[1] 0.8923323

> prop.table(tab,1)  # condition on observed values for FPR, TPR, etc.

         FALSE       TRUE
  0 0.97351714 0.02648286
  1 0.72015053 0.27984947

> tab=table(default$default, fit$fitted.values>.3) # c=.3
> sum(diag(tab))/sum(tab)
[1] 0.8886086
> prop.table(tab,1)  # TPR increases, but so does FPR

         FALSE       TRUE
  0 0.91783058 0.08216942
  1 0.33185084 0.66814916
```

How are FPR, TRP, etc. affected by class imbalance?

ROC Curves

```
a = (0:100)/100 # different cut points, don't call it c in R!
tpr = rep(NA, 101)  # true positive rate
fpr = rep(NA, 101)  # false positive rate
denom=table(default$default)
for(i in 1:101){
  num=table(default$default[fit$fitted.values>=a[i]])
  fpr[i] = num[1]/denom[1]
  tpr[i] = num[2]/denom[2]
}
plot(fpr, tpr, type="l", xlab="False Positive Rate", ylab="True Positive Rate")
abline(0,1)
b = (0:10)*10+1
points(fpr[b], tpr[b], pch=16)
text(fpr[b[-1]]+.01, tpr[b[-1]], paste("c=",as.character(a[b[-1]])), adj=0)
text(1,.98, "c=0")

library(pROC)
plot.roc(default$default, fit$fitted.values)
```

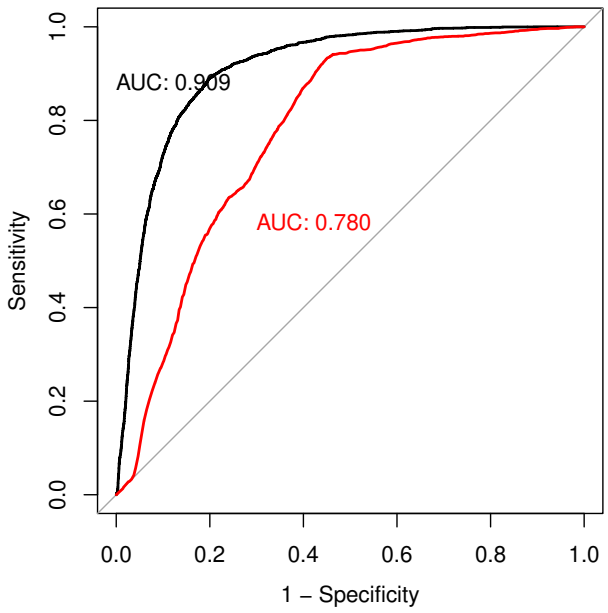


Area Under the ROC Curve (AUC) is used to evaulate models

ROC Curves

We are usually interested in comparing the ROC curves and AUC values for different models:

```
library(pROC)
ok = !is.na(default$use) # use has missing values
fit = glm(default ~ log(downpmt+1)+pmttype+use+age+gender, binomial, default, subset=ok)
plot.roc(default$default[ok], fit$fitted.values, legacy.axes=T,
    print.auc=T, print.auc.x=1, print.auc.y=.9)
fit2 = glm(default ~ pmttype+age+gender, binomial, default, subset=ok)
plot.roc(default$default[ok], fit2$fitted.values, add=T, col=2,
    print.auc=T, print.auc.x=.7, print.auc.y=.6, print.auc.col=2)
```



Important: ROC/AUC does not depend on the fraction of cases in each class (class imbalance problem), while classification rate does!