STAT 516 Lec 03

Multiple linear regression (part 1/2)

Karl Gregory

2026-09-02

SC apartments example

These data are a subset of "Apartment for Rent Classified" (2019).

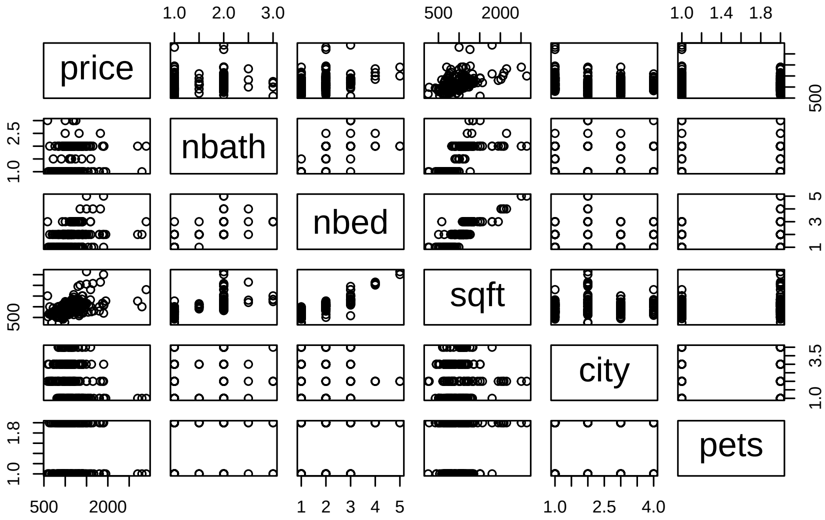
```
link <- url("https://gregorkb.github.io/data/scapts.csv")
scapts <- read.csv(link)
head(scapts)
```

```
  price nbath nbed sqft     city pets
1  1199     2    3 1200 Columbia  yes
2  1199     2    3 1450 Columbia  yes
3  1299     2    3 1950 Columbia  yes
4  1175     2    3 1312 Columbia  yes
5  1349     2    4 2015 Columbia  yes
6  1225     2    3 1336 Columbia  yes
```

```
n <- nrow(scapts)
```

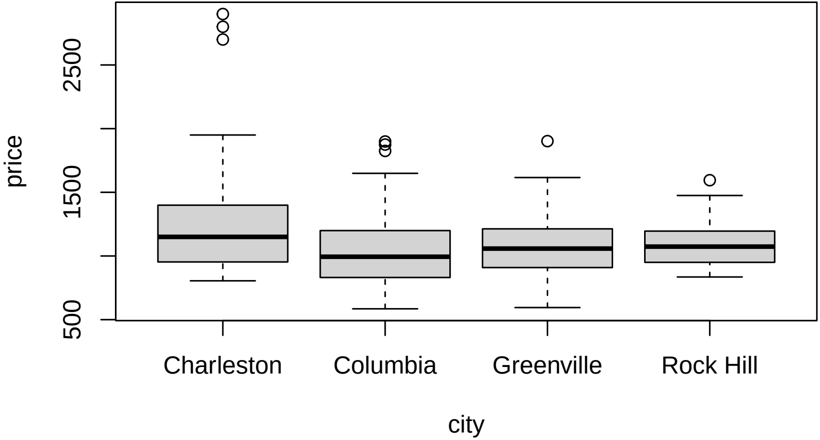
There are $n = 214$ data points.

```r
plot(scapts)
```



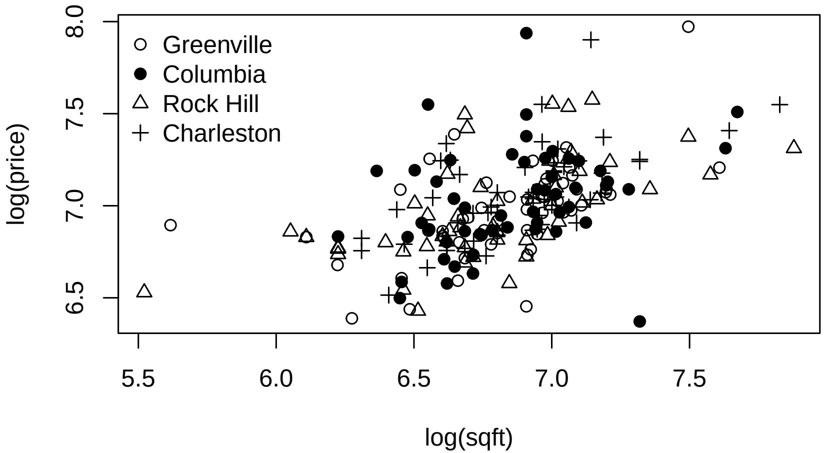
SC apartments (cont)

```
boxplot(price ~ city, data = scapts)
```



SC apartments (cont)

```
symbols <- c("Greenville" = 1, "Columbia" = 19, "Rock Hill" = 2, "Charleston" = 3)
plot(log(price) ~ log(sqft), data = scapts, pch = symbols)
legend("topleft",legend = names(symbols),pch = symbols, bty = "n")
```



Setup

Consider data $(\mathbf{x}_1, Y_1), \dots, (\mathbf{x}_1, Y_n)$, with each $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$.

The multiple linear regression model is

$$Y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p + \varepsilon_i, \quad i = 1, \dots, n,$$

where

- $\mathbf{x}_1, \dots, \mathbf{x}_n$ are vectors in $\mathbb{R}^p$ of covariate or predictor values.
- $Y_1, \dots, Y_n$ are the response values
- $\beta_0, \beta_1, \dots, \beta_p$ are the regression coefficients.
- $\varepsilon_1, \dots, \varepsilon_n$ are iid Normal$(0, \sigma^2)$ error terms.
- σ^2 is the error term variance.

Goals in multiple linear regression

As in *simple* linear regression, will learn how to

1. Estimate the regression coefficients β_0 and $\beta_1, \dots, \beta_p$.
2. Estimate the error term variance σ^2.
3. Perform inference on $\beta_1, \dots, \beta_p$.
4. Build a CI for $\beta_0 + \beta_1 x_{\text{new},1} + \cdots + \beta_p x_{\text{new},p}$ at any $\mathbf{x}_{\text{new}}$.
5. Build a prediction interval for Y at any $\mathbf{x}_{\text{new}}$.
6. Decompose the variation in Y into (sums of) sums of squares.
7. Check whether the model assumptions are satisfied.
8. Identify outliers and understand their effects.

Beyond the above, in *multiple* linear regression we wish to

8. Test for significance of a subset of covariates
9. Understand how correlations among the covariates affect inferences
10. Do variable selection

Latter goals considered in part 2/2.

Least-squares estimation of regression coefficients

Define the squared error criterion as

$$Q(b_0, b_1, \dots, b_p) = \sum_{i=1}^{n}(Y_i - (b_0 + b_1 x_{i1} + \cdots + b_p x_{ip}))^2.$$

Suppose $Q(b_0, b_1, \dots, b_p)$ is uniquely minimized at $(\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$.

Then we call $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$ the least-squares estimators of $\beta_0, \beta_1, \dots, \beta_p$.

The best way to compute $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$ is with matrix calculations...

Linear regression model in matrix form

Write equations $Y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p + \varepsilon_i$, for $i = 1, \ldots, n$, as

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1 x_{11} + \cdots + \beta_p x_{1p} + \varepsilon_1 \\ Y_2 &= \beta_0 + \beta_1 x_{21} + \cdots + \beta_p x_{2p} + \varepsilon_2 \\ &\vdots \\ Y_n &= \beta_0 + \beta_1 x_{n1} + \cdots + \beta_p x_{np} + \varepsilon_n \end{aligned}$$

Now set

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \ldots & x_{np} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Then the above equations can be written in matrix form as $\mathbf{Y} = \mathbf{X}\mathbf{b} + \mathbf{e}$.

Least-squares estimators in matrix form

Provided $\mathbf{X}^T\mathbf{X}$ is invertible, the entries of the vector

$$\hat{\mathbf{b}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

give the least-squares estimators $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$.

Important: Can only compute $\hat{\mathbf{b}}$ if no column of $\mathbf{X}$ can be constructed as a linear combination of other columns (equivalent to $\mathbf{X}^T\mathbf{X}$ invertible).

Estimating the error term variance

After obtaining $\widehat{\beta}_0, \widehat{\beta}_1, \ldots, \widehat{\beta}_p$, define the

- fitted values as $\hat{Y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \cdots + \widehat{\beta}_p x_{ip}$
- residuals as $\widehat{\varepsilon}_i = Y_i - \hat{Y}_i$

for $i = 1, \ldots, n$.

Then an unbiased estimator of σ^2 is given by

$$\widehat{\sigma}^2 = \frac{1}{n - (p+1)} \sum_{i=1}^{n} \widehat{\varepsilon}_i^2.$$

SC apartments (cont)

Make indicator or dummy variables to encode the four cities:

$$x_{\text{Columbia}} = \begin{cases} 1 & \text{if city is Columbia} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{\text{Greenville}} = \begin{cases} 1 & \text{if city is Greenville} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{\text{Rock Hill}} = \begin{cases} 1 & \text{if city is Rock Hill} \\ 0 & \text{otherwise} \end{cases}$$

Then Charleston is indicated by $(x_{\text{Columbia}}, x_{\text{Greenville}}, x_{\text{Rock Hill}}) = (0, 0, 0)$.

Note: If we included in $\mathbf{X}$ an indicator for all four levels, $\mathbf{X}^T\mathbf{X}$ would not be invertible due to the presence of the intercept column (this is linear algebra knowledge)!

SC apartments (cont)

The pets column contains the values “yes” and “no”.

Use a dummy variable to encode it:

$$x_{\text{pets}} = \begin{cases} 1 & \text{if yes} \\ 0 & \text{if no} \end{cases}$$

SC apartments (cont)

Construct the response vector and the design matrix.

```
Y <- log(scapts$price)
X <- cbind("Intercept" = rep(1,n),
           "Columbia" = ifelse(scapts$city == "Columbia",1,0),
           "Greenville" = ifelse(scapts$city == "Greenville",1,0),
           "Rock Hill" = ifelse(scapts$city == "Rock Hill",1,0),
           "nbath" = scapts$nbath,
           "nbed" = scapts$nbed,
           "log_sqft" = log(scapts$sqft),
           "pets" = ifelse(scapts$pets == "yes",1,0))
```

SC apartments (cont)

Estimate the regression coefficients and the error term variance:

```
bhat <- solve(t(X) %*% X) %*% t(X) %*% Y
bhat
```

```
                  [,1]
Intercept   3.93274698
Columbia   -0.24090102
Greenville -0.11950394
Rock Hill  -0.14535978
nbath      -0.10728344
nbed        0.06453526
log_sqft    0.47283468
pets       -0.01857260
```

```
Yhat <- X %*% bhat
ehat <- Y - Yhat
p <- ncol(X) - 1
sgsqhat <- sum(ehat^2) / (n - (p + 1))
sgsqhat
```

```
[1] 0.04336654
```

Interpretation of the slope parameters

Consider what stories $\beta_0, \beta_1, \ldots, \beta_p$ tell in the MLR model

$$Y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p + \varepsilon_i, \quad i = 1, \ldots, n.$$

- β_0, as in SLR, just gives the function the right "height".
- β_j is the amount by which the mean of Y changes due to a 1-unit increase in covariate j, with all other variables held fixed.

SC apartments

```
bhat
```

```
                  [,1]
Intercept   3.93274698
Columbia   -0.24090102
Greenville -0.11950394
Rock Hill  -0.14535978
nbath      -0.10728344
nbed        0.06453526
log_sqft    0.47283468
pets       -0.01857260
```

- The effect of an additional bedroom (all else being equal) is to increase the expected log price by 0.0645353, or to increase the expected price by a factor of 1.0666632, or by 6.6663193 percent.
- An apartment which allows pets (all else being equal) is expected to have a log price 0.0185726 lower, that is, to be cheaper by a factor of 0.9815988, i.e. to be 1.8401195 percent cheaper.
- In which city are apartments most expensive (all else being equal)?

Do not omit *all else being equal* (or *ceteris paribus* in the Latin ;-))!

Confidence intervals for the slope parameters

Let $\Omega = (\mathbf{X}^T\mathbf{X})^{-1}$ with Ω_{jj} the diagonal entry corresponding to β_j.

- Then the estimator $\hat{\beta}_j$ is distributed as

$$\hat{\beta}_j \sim \text{Normal}\left(\beta_j, \sigma^2\Omega_{jj}\right).$$

- "Studentizing" the above gives

$$\frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}\sqrt{\Omega_{jj}}} \sim t_{n-(p+1)}.$$

- So a $(1-\alpha)100\%$ confidence interval for β_j is

$$\hat{\beta}_j \pm t_{n-(p+1),\alpha/2}\hat{\sigma}\sqrt{\Omega_{jj}}.$$

- We often write $\widehat{\text{s.e.}}(\hat{\beta}_1) = \hat{\sigma}\sqrt{\Omega_{jj}}$, s.e. for <u>standard error</u>.

SC apartments (cont)

Construct 95% confidence intervals for the slope coefficients.

```
alpha <- 0.05
Om <- solve(t(X) %*% X)
om <- diag(Om)
ta2 <- qt(1-alpha/2,n - (p + 1))
se <- sqrt(sgsqhat * om)
lo <- bhat - ta2 * se
up <- bhat + ta2 * se
cis <- round(cbind(bhat,lo,up),4)
colnames(cis) <- c("estimate","lower","upper")
rownames(cis) <- colnames(X)
print(cis)
```

```
           estimate   lower   upper
Intercept    3.9327  2.9839  4.8816
Columbia    -0.2409 -0.3207 -0.1611
Greenville  -0.1195 -0.1909 -0.0481
Rock Hill   -0.1454 -0.2397 -0.0510
nbath       -0.1073 -0.1927 -0.0219
nbed         0.0645 -0.0026  0.1317
log_sqft     0.4728  0.3188  0.6268
pets        -0.0186 -0.0783  0.0411
```

Tests of hypotheses about the slope coeffcients

We most often test hypotheses about the β_j of the form

$$H_0\colon\ \beta_j \geq 0 \quad \text{or} \quad H_0\colon\ \beta_j = 0 \quad \text{or} \quad H_0\colon\ \beta_j \leq 0$$
$$H_1\colon\ \beta_j < 0 \qquad\qquad H_1\colon\ \beta_j \neq 0 \qquad\qquad H_1\colon\ \beta_j > 0.$$

Reject or fail to reject H_0 based on the value of the test statistic

$$T_{\text{test}} = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{\Omega_{jj}}}.$$

Rejection rules for the above at significance level α are

$$T_{\text{test}} < -t_{n-(p+1),\alpha} \quad \text{or} \quad |T_{\text{test}}| > t_{n-(p+1),\alpha/2} \quad \text{or} \quad T_{\text{test}} > t_{n-(p+1),\alpha}.$$

The corresponding p-values are, with $T \sim t_{n-(p+1)}$, the probabilities

$$P(T < T_{\text{test}}) \quad \text{or} \quad 2 \times P(T > |T_{\text{test}}|) \quad \text{or} \quad P(T > T_{\text{test}}).$$

SC apartments (cont)

Obtain p-values for testing H_0: $\beta_j = 0$ vs H_1: $\beta_j \neq 0$ for each j.

```
sehat <- sqrt(sgsqhat * om)
Tstat <- bhat / sehat
pval <- 2*(1 - pt(abs(Tstat),df = n - (p + 1)))
summ <- round(cbind(bhat,sehat,Tstat,pval),4)
colnames(summ) <- c("estimate","sehat","Tstat","pval")
rownames(summ) <- colnames(X)
print(summ)
```

```
           estimate  sehat   Tstat   pval
Intercept    3.9327 0.4812  8.1720 0.0000
Columbia    -0.2409 0.0405 -5.9508 0.0000
Greenville  -0.1195 0.0362 -3.2987 0.0011
Rock Hill   -0.1454 0.0479 -3.0371 0.0027
nbath       -0.1073 0.0433 -2.4762 0.0141
nbed         0.0645 0.0341  1.8950 0.0595
log_sqft     0.4728 0.0781  6.0535 0.0000
pets        -0.0186 0.0303 -0.6135 0.5402
```

The `lm()`, `summary()`, and `confint()` functions in R

```r
lm_out <- lm(log(price) ~ city + nbath + nbed + log(sqft) + pets, data = scapts)
summary(lm_out)
```

```
Call:
lm(formula = log(price) ~ city + nbath + nbed + log(sqft) + pets,
    data = scapts)

Residuals:
     Min       1Q   Median       3Q      Max
-0.65276 -0.12428 -0.01501  0.08756  0.71661

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.93275    0.48125   8.172 3.04e-14 ***
cityColumbia   -0.24090    0.04048  -5.951 1.13e-08 ***
cityGreenville -0.11950    0.03623  -3.299  0.00114 **
cityRock Hill  -0.14536    0.04786  -3.037  0.00270 **
nbath          -0.10728    0.04333  -2.476  0.01409 *
nbed            0.06454    0.03406   1.895  0.05950 .
log(sqft)       0.47283    0.07811   6.053 6.61e-09 ***
petsyes        -0.01857    0.03027  -0.614  0.54021
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2082 on 206 degrees of freedom
Multiple R-squared:  0.414, Adjusted R-squared:  0.3941
F-statistic: 20.79 on 7 and 206 DF,  p-value: < 2.2e-16
```

```r
confint(lm_out)
```

```
                    2.5 %      97.5 %
(Intercept)     2.983942705  4.88155125
cityColumbia   -0.320713599 -0.16108844
cityGreenville -0.190928040 -0.04807984
cityRock Hill  -0.239721140 -0.05099842
nbath          -0.192703165 -0.02186372
nbed           -0.002607853  0.13167838
log(sqft)       0.318837614  0.62683175
petsyes        -0.078255964  0.04111076
```

```r
confint(lm_out, level = .99)
```

```
                   0.5 %       99.5 %
(Intercept)     2.68154577  5.183948180
cityColumbia   -0.34615096 -0.135651076
cityGreenville -0.21369188 -0.025315996
cityRock Hill  -0.26979540 -0.020924159
nbath          -0.21992760  0.005360712
nbed           -0.02400729  0.153077815
log(sqft)       0.26975664  0.675912723
petsyes        -0.09727787  0.060132663
```

CI for the mean and PI for Y_{new} at $\mathbf{x}_{\text{new}}$

For a new vector of covariate values $\mathbf{x}_{\text{new}}$, let

$$\hat{Y}_{\text{new}} = \hat{\beta}_0 + \hat{\beta}_1 x_{\text{new},1} + \cdots + \hat{\beta}_p x_{\text{new},p}$$

- A $(1-\alpha) \times 100$ CI for $\beta_0 + \beta_1 x_{\text{new},1} + \cdots + \beta_p x_{\text{new},p}$ is given by

$$\hat{Y}_{\text{new}} + t_{n-(p+1),\alpha/2} \hat{\sigma} \sqrt{\Omega_{\text{new}}},$$

- A $(1-\alpha) \times 100$ PI for Y_{new} corresponding to $\mathbf{x}_{\text{new}}$ is given by

$$\hat{Y}_{\text{new}} + t_{n-(p+1),\alpha/2} \hat{\sigma} \sqrt{1 + \Omega_{\text{new}}},$$

where $\Omega_{\text{new}} = \tilde{\mathbf{x}}_{\text{new}}^T \Omega \ \tilde{\mathbf{x}}_{\text{new}}^T$ with $\tilde{\mathbf{x}}_{\text{new}}^T = (1 \ x_{\text{new},1} \ \cdots \ x_{\text{new},p})^T$.

SC apartments (cont)

Build 95% CI for the average rent of apartments in Greenville with one bathroom, two bedrooms, nine-hundred square feet, and which allow pets.

```r
xnew <- c(1,0,1,0,1,2,log(900),1)
om_new <- t(xnew) %*% Om %*% xnew
Ynew_hat <- t(xnew) %*% bhat
seci <- sqrt(sgsqhat) * sqrt(om_new)
loci <- Ynew_hat - ta2 * seci
upci <- Ynew_hat + ta2 * seci

# exponentiate since we have used the log of the price
eloci <- exp(loci)
eupci <- exp(upci)
```

The confidence interval is $(1040.545, 1234.266)$.

Now build a 95% PI for the rent of a single such a property.

```
sepi <- sqrt(sgsqhat) * sqrt( 1 + om_new)
lopi <- Ynew_hat - ta2 * sepi
uppi <- Ynew_hat + ta2 * sepi

# exponentiate since we have used the log of the price
elopi <- exp(lopi)
euppi <- exp(uppi)
```

The prediction interval is $(745.099, 1723.675)$.

The predict() function in R

```r
newdata <- data.frame(city = "Greenville",
                      nbath = 1,
                      nbed = 2,
                      sqft = 900,
                      pets = "yes")
ci <- predict(lm_out, newdata = newdata, int = "conf")
exp(ci)
```

```
       fit      lwr      upr
1 1133.274 1040.545 1234.266
```

```r
pi <- predict(lm_out, newdata = newdata, int = "pred")
exp(pi)
```

```
       fit      lwr      upr
1 1133.274 745.0992 1723.675
```

Sums of squares in multiple linear regression

We decompose the variation in $Y_1, \dots, Y_n$ by defining the:

- Total sum of squares: $\mathrm{SS}_{\mathrm{Tot}} = \sum_{i=1}^{n}(Y_i - \bar{Y}_n)^2$
- Regression sum of squares: $\mathrm{SS}_{\mathrm{Reg}} = \sum_{i=1}^{n}(\hat{Y}_i - \bar{Y}_n)^2$
- Error sum of squares: $\mathrm{SS}_{\mathrm{Error}} = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$

We have $\mathrm{SS}_{\mathrm{Tot}} = \mathrm{SS}_{\mathrm{Reg}} + \mathrm{SS}_{\mathrm{Error}}$.

The coefficient of determination is defined as $R^2 = \dfrac{\mathrm{SS}_{\mathrm{Reg}}}{\mathrm{SS}_{\mathrm{Tot}}}$.

- $R^2 \in [0, 1]$
- Proportion of variation in Y "explained" by the covariates $x_1, \dots, x_p$.

The mean squares in multiple linear regression

The SS, appropriately scaled, follow chi-square distributions:

- $\mathrm{SS}_{\mathrm{Tot}}/\sigma^2 \sim \chi^2_{n-1}(\phi_{\mathrm{Tot}})$
- $\mathrm{SS}_{\mathrm{Reg}}/\sigma^2 \sim \chi^2_{p}(\phi_{\mathrm{Reg}})$
- $\mathrm{SS}_{\mathrm{Error}}/\sigma^2 \sim \chi^2_{n-(p+1)}$,

where ϕ_{Tot} and ϕ_{Reg} are noncentrality parameters.

Dividing $\mathrm{SS}_{\mathrm{Reg}}$ and $\mathrm{SS}_{\mathrm{Error}}$ by their dfs, we define:

- <u>Regression mean square</u>: $\mathrm{MS}_{\mathrm{Reg}} = \dfrac{\mathrm{SS}_{\mathrm{Reg}}}{p}$
- <u>Error mean square</u>: $\mathrm{MS}_{\mathrm{Error}} = \dfrac{\mathrm{SS}_{\mathrm{Error}}}{n-(p+1)}$

Moreover, define the <u>adjusted R squared</u> as $\bar{R}^2 = 1 - \dfrac{\mathrm{MS}_{\mathrm{Error}}}{\mathrm{SS}_{\mathrm{Tot}}/(n-1)}$.

Adjustment "penalizes" the inclusion of additional covariates.

The Analysis of Variance (ANOVA) table

We often present the SS, df, and MS values in a table like this:

Source	Df	SS	MS	F value	p-value
Reg	p	$\mathrm{SS}_{\mathrm{Reg}}$	$\mathrm{MS}_{\mathrm{Reg}}$	F_{test}	$P(F > F_{\mathrm{test}})$
Error	$n-(p+1)$	$\mathrm{SS}_{\mathrm{Error}}$	$\mathrm{MS}_{\mathrm{Error}}$		
Total	$n-1$	$\mathrm{SS}_{\mathrm{Tot}}$			

This is an example of an <u>ANOVA table</u>.

The F-value and the p-value we will discuss later in these slides.

Building the ANOVA table

```r
Ybar <- mean(Y)
SST <- sum((Y - Ybar)^2)
SSR <- sum((Yhat - Ybar)^2)
SSE <- sum((Y - Yhat)^2)
MSR <- SSR / p
MSE <- SSE / (n-(p+1))
Fstat <- MSR / MSE
pval <- 1 - pf(Fstat,p,n-(p+1))
```

Source	Df	SS	MS	F value	p-value
Regression	7	6.31	0.9	20.79	0
Error	206	8.93	0.04		
Total	213	15.24			

Moreover $R^2 = 0.414$ and $\bar{R}^2 = 0.394$.

ANOVA quantities in output from `lm()` with `summary()`

```r
summary(lm_out)
```

```
Call:
lm(formula = log(price) ~ city + nbath + nbed + log(sqft) + pets,
    data = scapts)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.65276 -0.12428 -0.01501  0.08756  0.71661 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     3.93275    0.48125   8.172 3.04e-14 ***
cityColumbia   -0.24090    0.04048  -5.951 1.13e-08 ***
cityGreenville -0.11950    0.03623  -3.299  0.00114 ** 
cityRock Hill  -0.14536    0.04786  -3.037  0.00270 ** 
nbath          -0.10728    0.04333  -2.476  0.01409 *  
nbed            0.06454    0.03406   1.895  0.05950 .  
log(sqft)       0.47283    0.07811   6.053 6.61e-09 ***
petsyes        -0.01857    0.03027  -0.614  0.54021    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2082 on 206 degrees of freedom
Multiple R-squared:  0.414, Adjusted R-squared:  0.3941 
F-statistic: 20.79 on 7 and 206 DF,  p-value: < 2.2e-16
```

Sequential SS with anova() function (seldom use)

Sequential SS report the changes in $\mathrm{SS}_{\mathrm{Reg}}$ from adding new variables.

```r
lm_out <- lm(log(price) ~ city + nbath + nbed + log(sqft) + pets, data = scapts)
anova(lm_out)
```

```
Analysis of Variance Table

Response: log(price)
           Df Sum Sq Mean Sq F value    Pr(>F)    
city        3 1.0337 0.34458  7.9458 4.869e-05 ***
nbath       1 1.7918 1.79184 41.3185 8.832e-10 ***
nbed        1 1.8666 1.86660 43.0425 4.250e-10 ***
log(sqft)   1 1.6021 1.60213 36.9439 5.804e-09 ***
pets        1 0.0163 0.01632  0.3764    0.5402    
Residuals 206 8.9335 0.04337                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The sequential SS depend on the order in which variables are added:

```r
lm2 <- lm(log(price) ~ nbath + city + nbed + log(sqft) + pets, data = scapts)
anova(lm2)
```

```
Analysis of Variance Table

Response: log(price)
           Df Sum Sq Mean Sq F value    Pr(>F)    
nbath       1 1.3087 1.30872 30.1782 1.155e-07 ***
city        3 1.5169 0.50562 11.6592 4.344e-07 ***
nbed        1 1.8666 1.86660 43.0425 4.250e-10 ***
log(sqft)   1 1.6021 1.60213 36.9439 5.804e-09 ***
pets        1 0.0163 0.01632  0.3764    0.5402    
Residuals 206 8.9335 0.04337                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Sequential model fits to obtain sequential SS

```r
lm1 <- lm(log(price) ~ nbath, data = scapts)
lm2 <- lm(log(price) ~ nbath + city, data = scapts)
lm3 <- lm(log(price) ~ nbath + city + nbed, data = scapts)
lm4 <- lm(log(price) ~ nbath + city + nbed + log(sqft), data = scapts)
lm5 <- lm(log(price) ~ nbath + city + nbed + log(sqft) + pets, data = scapts)

SSR1 <- SST - sum(lm1$residuals^2)
SSR2 <- SST - sum(lm2$residuals^2)
SSR3 <- SST - sum(lm3$residuals^2)
SSR4 <- SST - sum(lm4$residuals^2)
SSR5 <- SST - sum(lm5$residuals^2)
seqSS <- c(SSR1,SSR2 - SSR1,SSR3 - SSR2, SSR4 - SSR3, SSR5 - SSR4)
names(seqSS) <- c("city","nbath","nbed","log(sqft)","pets")
round(seqSS,3)
```

```
 city     nbath      nbed log(sqft)      pets
1.309     1.517     1.867     1.602     0.016
```

Interactions

- When the effect of one predictor depends on the value of another it is called an interaction.
  - Does the effect of number of bedrooms depend on the number of bathrooms?
  - Does the effect of square feet differ between Charleston and Rock Hill?
- We can include interactions by including in $\mathbf{X}$ products of pairs of predictors.
- Most often we consider interactions between dummy variables and numeric variables.
- Such interactions act as "slope modifiers", allowing different slope coefficients for different subsets of the data.

SC apartments (cont)

Add interactions between:

- Number of bedrooms/bathrooms.
- Square feet and each city indicator.

```
Y <- log(scapts$price)
X <- cbind("Intercept" = rep(1,n),
           "Columbia" = ifelse(scapts$city == "Columbia",1,0),
           "Greenville" = ifelse(scapts$city == "Greenville",1,0),
           "Rock Hill" = ifelse(scapts$city == "Rock Hill",1,0),
           "nbath" = scapts$nbath,
           "nbed" = scapts$nbed,
           "log_sqft" = log(scapts$sqft),
           "pets" = ifelse(scapts$pets == "yes",1,0),
           "nbath:nbed" = scapts$nbath*scapts$nbed,
           "log_sqft:Columbia" = log(scapts$sqft)*ifelse(scapts$city == "Columbia",1,0),
           "log_sqft:Greenville" = log(scapts$sqft)*ifelse(scapts$city == "Greenville",1,0),
           "log_sqrt:Rock Hill" = log(scapts$sqft)*ifelse(scapts$city == "Rock Hill",1,0))
```

SC apartments (cont)

Include interactions in the `lm()` function as below:

```
lm_out <- lm(log(price) ~ city + nbath + nbed + nbath:nbed + log(sqft) + pets + log(sqft):city, data = scapts)
summary(lm_out)
```

```
Call:
lm(formula = log(price) ~ city + nbath + nbed + nbath:nbed +
    log(sqft) + pets + log(sqft):city, data = scapts)

Residuals:
     Min       1Q   Median       3Q      Max
-0.59947 -0.13123 -0.00646  0.09455  0.69055

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               2.878148   0.686715   4.191 4.15e-05 ***
cityColumbia              1.677294   0.819015   2.048   0.0419 *
cityGreenville            0.891876   0.888803   1.003   0.3168
cityRock Hill             0.429247   1.384079   0.310   0.7568
nbath                    -0.181440   0.102964  -1.762   0.0796 .
nbed                      0.042290   0.082471   0.513   0.6087
log(sqft)                 0.639417   0.108495   5.894 1.56e-08 ***
petsyes                  -0.005982   0.031216  -0.192   0.8482
nbath:nbed                0.024299   0.048527   0.501   0.6171
cityColumbia:log(sqft)   -0.282220   0.120906  -2.334   0.0206 *
cityGreenville:log(sqft) -0.149093   0.130962  -1.138   0.2563
cityRock Hill:log(sqft)  -0.086384   0.200685  -0.430   0.6673
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2073 on 202 degrees of freedom
Multiple R-squared:  0.4307,    Adjusted R-squared:  0.3997
F-statistic: 13.89 on 11 and 202 DF,  p-value: < 2.2e-16
```

Check for significance of any interaction effects and give interpretations.

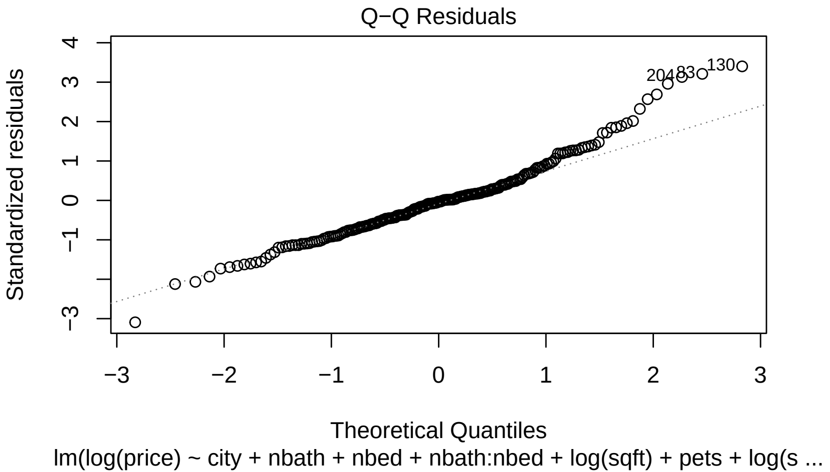
Checking model assumptions

Validity of the foregoing analyses depends on these assumptions:

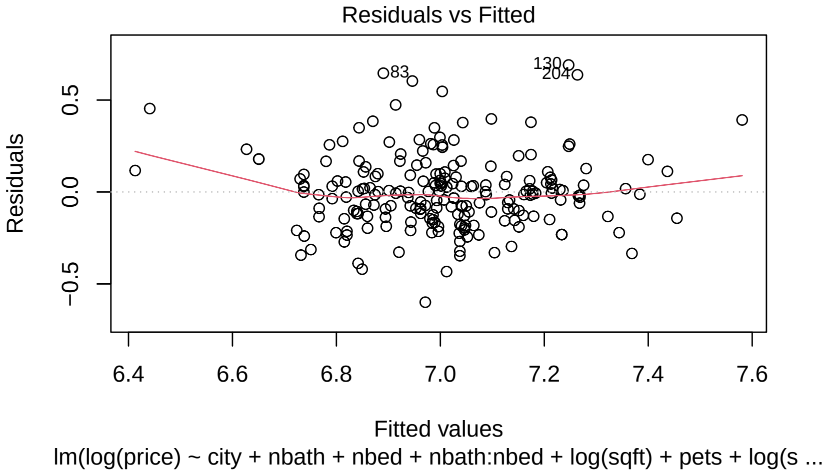
1. The responses are normally distributed around the regression line (Check QQ plot of residuals). *If n is large this only matters for prediction intervals.*
2. The response has the same variance for all covariate values (Check residuals vs fitted values plot).
3. The covariates and the response are linearly related (Check residuals vs fitted values plot).
4. The response values are independent of each other (No way to check; must trust experimental design).

Generating diagnostic plots from `lm()` with `plot()`

```r
plot(lm_out,which = 2)
```



```r
plot(lm_out,which = 1)
```



Leverage and Cook's distance in MLR

The leverage of a point $(Y_i, \mathbf{x}_i)$ among $(Y_1, \mathbf{x}_1), \ldots, (Y_n, \mathbf{x}_n)$ is

$$\mathsf{lev}_i = \text{ entry } i \text{ on the diagonal of the matrix } \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}.$$

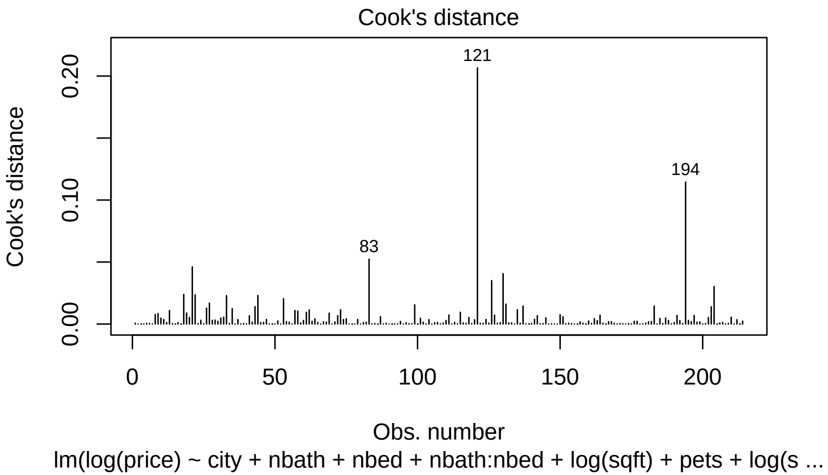
Leverage only shows outlying-ness in the covariate space.

Cook's Distance measures how much each data point changes the fit:

$$D_i = \frac{1}{(p+1)\hat{\sigma}^2} \sum_{j=1}^{n} (\hat{Y}_j - \hat{Y}_{j(i)})^2 = \frac{\hat{e}_i^2}{(p+1)\hat{\sigma}^2} \frac{\mathsf{lev}_i}{(1-\mathsf{lev}_i)^2} \quad \text{for } i = 1, \ldots, n,$$

where $\hat{Y}_{j(i)}$ is the jth fitted value from the model fitted without obs i.

```r
plot(lm_out,which = 4)
```



References

"Apartment for Rent Classified." 2019. UCI Machine Learning Repository.