Exercise Set Postlude-1

1) See the set of plots for the first dataset in the quartet with

```
lm.fit <- lm(y1 ~ x1, data = anscombe)
plot(lm.fit)
```

and press enter to cycle through the plots. For the other datasets in the quartet, change the variable names accordingly.

Exercise Set Postlude-2

1) Once the package is installed and loaded, fit the four models and run the diagnostics with

```
lm.fit1 <- lm(y1 ~ x1, data = anscombe)
gvlma(lm.fit1)
lm.fit2 <- lm(y2 ~ x2, data = anscombe)
gvlma(lm.fit2)
lm.fit3 <- lm(y3 ~ x3, data = anscombe)
gvlma(lm.fit3)
lm.fit4 <- lm(y4 ~ x4, data = anscombe)
gvlma(lm.fit4)
```

As expected from the plot, tests using the first dataset reveal no clear reasons for concern. But remember that the samples size in this example is very small, which means that power to detect deviations is low.

The second model shows that the “link function” test—which is intended to detect departures from linearity—returns a low *p* value. This makes sense: the data clearly fit a curve and not a line.

The third dataset reveals trouble with the normality assumption. This isn’t as informative as looking at the plot, which would likely lead us to investigate whether the single point falling off the line is some sort of error. But at least the test has returned an alarm when it should.

The fourth result is more disquieting—the tests detect no problems with the assumptions, even though the plot suggests something is wrong.

In short, the tests are useful in conjunction with the plots, but they do not replace them.

Exercise Set Postlude-3

1) a) Code in text.
b) Here’s some code:

```
resid.am.hp <- lm(am ~ hp, data = mtcars)$residuals #1
resid.mpg.hp <- lm(mpg ~ hp, data = mtcars)$residuals #2
resid.mod.hp <- lm(resid.mpg.hp ~ resid.am.hp) #3
summary(resid.mod.hp)
```

The estimated slope for the residuals of transmission type here is the same as the estimated slope for transmission type in part (a). (It is *not* the same as the estimated slope as you would get from a simple linear regression of transmission type and miles per gallon. The reason for this difference is the association between transmission type and horsepower, which matters in the multiple regression. The standard errors and test statistics are differ from the multiple regression case, for a similar reason.)

As to what's going on, it's possible to state a concise proof that this will happen using matrix algebra or a much more verbose one using approaches similar to the ones you used in chapter 3. Conceptually, though, think of multiple regression coefficient estimates as assessing the association between an independent variable and the dependent variable when the other independent variables are held constant. In this case, by residualizing both weight and miles per gallon on horsepower, we are obtaining versions of these two variables whose association with horsepower has been "removed."

c)
```
resid.hp.am <- lm(hp ~ am, data = mtcars)$residuals #1
resid.mpg.am <- lm(mpg ~ am, data = mtcars)$residuals #2
resid.mod.am <- lm(resid.mpg.am ~ resid.hp.am) #3
summary(resid.mod.am)
```

d)
```
mean(mtcars$mpg) - mean(mtcars$am)*5.277 - mean(mtcars$hp)*(-0.05888)
```

2) In (a) and (b), the t-statistics and associated p values from the two pairs of analyses are equal. In part (c), the F statistics and associated p values are equal.

Exercise Set Postlude-4

1) `glm()` is for "generalized linear model." Once the car package is loaded, you can use

```
probit.fit <- glm(volunteer ~ extraversion + neuroticism + sex,
data = Cowles, family = binomial("probit"))
summary(probit.fit)
```

to get the estimates. To fit the logistic model instead, you would use `family = binomial("logit")`

2) a) Here is R code. The results agree with the conjecture of consistency, which makes sense given the general consistency (under weak conditions) of method-of-moments estimators:

```
plot(sim.lm(5000, 3, 1/2, het.coef = .25))
```

```
ests <- sim.lm.ests(n = 10, nsim = 1000, a = 3, b = 1/2,
het.coef = .25)
hist(ests[,2])
colMeans(ests)
apply(ests, 2, var)
ests <- sim.lm.ests(n = 50, nsim = 1000, a = 3, b = 1/2,
het.coef = .25)
colMeans(ests)
apply(ests, 2, var)
ests <- sim.lm.ests(n = 100, nsim = 1000, a = 3, b = 1/2,
het.coef = .25)
colMeans(ests)
apply(ests, 2, var)
ests <- sim.lm.ests(n = 1000, nsim = 1000, a = 3, b = 1/2,
het.coef = .25)
colMeans(ests)
apply(ests, 2, var)
```

b) Yes, they are consistent. In line with this, setting the n to 10,000 or 100,000 gives estimates very close to the coefficients specified in the simulation.
c) In the presence of heteroscedasticity in the model for the latent variables, the estimators for the model coefficients are inconsistent—they converge on the wrong numbers. The degree to which they're wrong increases with the severity of the heteroscedasticity.

Exercise Set Postlude-5

1) For concreteness, we will compute the correlation of the first and second observation from country 1, Y_{11} and Y_{12}, but the argument applies to any pair of observations from the same country. First, using the hint, adding non-random terms to a pair of random variables does not affect their correlation, so

$$\mathrm{Cor}(Y_{11}, Y_{12}) = \mathrm{Cor}(\alpha + \beta x_{11} + \mu_1 + \epsilon_{11}, \alpha + \beta x_{12} + \mu_1 + \epsilon_{12}) = \mathrm{Cor}(\mu_1 + \epsilon_{11}, \mu_1 + \epsilon_{12}).$$

Next, by the definition of correlation,

$$\mathrm{Cor}(\mu_1 + \epsilon_{11}, \mu_1 + \epsilon_{12}) = \frac{\mathrm{E}([\mu_1 + \epsilon_{11}][\mu_1 + \epsilon_{12}]) - \mathrm{E}(\mu_1 + \epsilon_{11})\mathrm{E}(\mu_1 + \epsilon_{12})}{\sqrt{\mathrm{Var}(\mu_1 + \epsilon_{11})}\sqrt{\mathrm{Var}(\mu_1 + \epsilon_{12})}}.$$

By the linearity of expectation and the fact that $\mathrm{E}(\mu_i) = \mathrm{E}(\epsilon_{ij}) = 0$ for all i and j, $\mathrm{E}(\mu_1 + \epsilon_{11})\mathrm{E}(\mu_1 + \epsilon_{12}) = 0$. Further, because the μ_i and ϵ_{ij} are independent, $\mathrm{Var}(\mu_1 + \epsilon_{11}) = \mathrm{Var}(\mu_1 + \epsilon_{12}) = \tau^2 + \sigma^2$. Applying these insights and expanding the expectation in the numerator gives

$$\frac{\mathrm{E}([\mu_1 + \epsilon_{11}][\mu_1 + \epsilon_{12}])}{\tau^2 + \sigma^2} = \frac{\mathrm{E}(\mu_1^2) + \mathrm{E}(\mu_1\epsilon_{11}) + \mathrm{E}(\mu_1\epsilon_{12}) + \mathrm{E}(\epsilon_{11}\epsilon_{12})}{\tau^2 + \sigma^2}.$$

Because the ϵ_{ij} are independent of each other, they have 0 covariance, which implies $\mathrm{E}(\epsilon_{11}\epsilon_{12}) = \mathrm{E}(\epsilon_{11})\mathrm{E}(\epsilon_{12}) = 0$. The same goes for the second two terms in the numerator, because the ϵ_{ij} are independent of the μ_i. Finally, rearranging the definition of variance gives $\mathrm{E}(\mu_i^2) = \mathrm{Var}(\mu_i) + [\mathrm{E}(\mu_i)]^2$. Because the expectation of the random effects is zero, $\mathrm{E}(\mu_i^2) = \mathrm{Var}(\mu_i) = \tau^2$. Applying these results gives the desired outcome,

$$\mathrm{Cor}(Y_{11}, Y_{12}) = \frac{\tau^2}{\tau^2 + \sigma^2}.$$

2) Here is code to carry out the simulations:

```
#Set parameters
alpha <- 3
beta <- 1/2
eps.sd <- sqrt(1/2)
re.sd <- 1
yrs <- 10
n.sims <- 10000

#Initialize variables
ints <- numeric(n.sims)
slopes <- numeric(n.sims)

#Simulate datasets and save least-squares estimates
for(i in 1:n.sims){
  x <- rep(anscombe$x1, yrs)
  rand.ints <- rnorm(length(anscombe$x1), 0, re.sd)
  y <- alpha + beta*x + rep(rand.ints, yrs) + rnorm(length(x),
0, eps.sd)
  mod.fit <- lm(y ~ x)
  ints[i] <- mod.fit$coefficients[1]
  slopes[i] <- mod.fit$coefficients[2]
}

#Plot and summarize estimates
hist(ints)
summary(ints)
sd(ints)

hist(slopes)
summary(slopes)
sd(slopes)
```

The least-squares estimates are unbiased. With these parameters, the standard deviation of the intercept estimates (which estimates the standard error of the estimator) is about 0.93, and the standard deviation of the slope estimates is about 0.10. These standard deviations are roughly in agreement with the mixed-model standard error estimates reported in the main text. They are much larger than the standard error estimates from simple linear regression. Ignoring dependence among the observations causes us to overestimate the amount of information we have, leading to standard error estimates that are too small.