Agenda

1. Announcements
2. Review about partitioning variability
3. Nested F-tests
4. Regression summary lab

Announcements

- Remember that initial project proposal are due Friday at midnight
- Homework 7 due Monday
- Reading quiz on chapter 3 due next Wednesday

Review of partitioning variability When we do Analysis of Variance (ANOVA) we are partitioning the variability. Recall:

$$\begin{aligned} \sum_{i=1}^{n}(y_i - \bar{y})^2 &= \sum_{i=1}^{n}(\hat{y_i} - \bar{y}) + \sum_{i=1}^{n}(y_i - \hat{y_i})^2 \\ SST &= SSM + SSE \end{aligned}$$

We have also defined

$$\begin{aligned} SXX &= \sum_{i=1}^{n}(x_i - \bar{x})^2 \\ SXY &= \sum_{i=1}^{n}(x_i - \bar{x}) \cdot (y_i - \bar{y}) \end{aligned}$$

And

$$\hat{\beta}_1 = \frac{SXY}{SXX}$$

Then, because we know the point $(\bar{y}, \bar{x})$ lies on the line, we can solve for $\hat{\beta}_0$,

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \cdot \bar{x}$$

Nested F-tests Individual t-tests in the regression summary have given us a way to test the statistical significant of individual terms in our model. But what if we want to test the significance of the contribution to the model by a *subset* of the predictors? That is where the nested F-test comes in.

- H_0: $\beta_i = 0$ for all predictors in the subset
- H_A: at least one $\beta_i \neq 0$

  $$F = \frac{(SSM_{full} - SSM_{reduced})/(\text{\# of predictors tested})}{SSE_{full}/(n-k-1)},$$

  where k is the # of predictors in the full model
- Use `anova` command in R, being careful that terms in the model are *nested*.

Regression summary lab Some code for your reference

```
> require(mosaic)
> require(fueleconomy)
> data(vehicles)
> myCars <- vehicles %>%
+   filter(year == 2000 & cyl == 4)
> xyplot(hwy ~ displ, data=myCars, main="Fuel Economy", alpha=0.5, cex=2, pch=19,
+        xlab="Engine Size (cubic centimeters)", ylab="Fuel Economy (miles per gallon)")
> m1 <- lm(hwy ~ displ, data=myCars)
> summary(m1)
> regdata <- myCars %>%
+   mutate(xdif = displ - mean(displ),  ydif = hwy - mean(hwy))
> regdata <- regdata %>%
+   summarize(SXX = sum(xdif^2),  SXY = sum(xdif*ydif))
> regdata <- regdata %>%
+   mutate(beta1=SXY/SXX)
> regdata
> coef(m1)["displ"]
> myCars %>%
+   mutate(xdif = displ - mean(displ),  ydif = hwy - mean(hwy)) %>%
+   summarize(SXX = sum(xdif^2), SXY = sum(xdif*ydif), beta1=SXY/SXX)
> myCars %>%
+   summarize(n=n(), SXX = var(displ) * (n-1), SXY = cov(hwy,displ) * (n-1), beta1 = SXY/SXX)
> myCars %>%
+   summarize(beta1 = cor(hwy, displ) * (sd(hwy) / sd(displ)))
> regdata <- myCars %>%
+   summarize(beta1 = cor(hwy, displ) * (sd(hwy) / sd(displ)), meanX = mean(displ), meanY = mean(hwy))
> regdata %>%
+   mutate(beta0 = meanY - beta1 * meanX)
> predict(m1, newdata=data.frame(displ=mean(~displ, data=myCars)))
> mean(~hwy, data=myCars)
> assessdata <- myCars %>%
+   mutate(ydif = (hwy - mean(hwy)))
> assessdata <- assessdata %>%
+   mutate(fitted = fitted(m1))
> assessdata <- assessdata %>%
+   summarize(n = n(), SST = sum(ydif^2), SSE = sum((fitted - hwy)^2), SSM = sum((fitted - mean(hwy))^2
> assessdata %>%
+   mutate(SSE + SSM)
> myCars %>%
+   mutate(ydif = (hwy - mean(hwy)), fitted = fitted(m1))  %>%
+   summarize(SST = sum(ydif^2), SSE = sum((fitted - hwy)^2), SSM = sum((fitted - mean(hwy))^2))
> # Coefficient of determination
> assessdata <- assessdata %>%
+   mutate(rsq = 1 - SSE / SST)
> rsquared(m1)
> # p is the number of explanatory variables
> p <- 1
> assessdata <- assessdata %>%
+   mutate(adjrsq = 1 - (SSE / (n-1-p)) / (SST / (n-1)))
> testdata <- myCars %>%
+    mutate(ydif = (hwy - mean(hwy)), fitted = fitted(m1)) %>%
+   summarize(n=n(), meanX = mean(displ), meanY = mean(hwy),
+             SXX = var(displ) * (n-1), SXY = cov(hwy,displ) * (n-1),
```

```
+             beta1 = SXY/SXX, beta0 = meanY - beta1 * meanX,
+             SST = sum(ydif^2), SSE = sum((fitted - hwy)^2), SSM = sum((fitted - mean(hwy))^2))
> # Residual Standard error
> testdata <- testdata %>%
+   mutate(RSE = sqrt(SSE / (n-2)))
> # Standard error
> testdata <- testdata %>%
+   mutate(SE1 = RSE / sqrt(SXX))
> testdata %>% glimpse()
> # t-statistic
> testdata <- testdata %>%
+   mutate(t1 = beta1 / SE1)
> testdata %>% glimpse()
> # p-value
> testdata %>%
+   summarize(p = 2 * pt(abs(t1), df=(n-2), lower.tail = FALSE))
> # Compute statistics for the intercept
> # Standard error
> testdata <- testdata %>%
+   mutate(SE0 = RSE * sqrt((1/n) + (meanX)^2 / SXX))
> # t-statistic
> testdata <- testdata %>%
+   mutate(t0 = beta0 / SE0)
> testdata %>% glimpse()
> # p-value
> testdata %>%
+   summarise(p = 2 * pt(abs(t0), df=(n-2), lower.tail = FALSE))
> anova(m1)
> # F-statistic
> testdata <- testdata %>%
+   mutate(F = (SSM / p) / (SSE / (n-1 - p)))
> testdata %>%
+   summarize(p = pf(F, df1 = p, df2 = n-1 - p, lower.tail=FALSE))
> bloodp <- read.csv("http://www.math.smith.edu/~bbaumer/mth247/labs/bloodpress.csv")
> library(GGally)
> ggpairs(bloodp)
> # pairs(bloodp) # this is a little faster, but uglier
>
> mfull <- lm(BP ~ ., data=bloodp)
> summary(mfull)
> require(car)
> vif(mfull)
> m1 <- lm(BP ~ Weight, data=bloodp)
> m2 <- lm(BP ~ Weight +  Age, data=bloodp)
> m3 <- lm(BP ~ Weight +  Age + Dur + Stress, data=bloodp)
> # Add the models in ascending order of complexity.
> anova(m1, m2, m3, mfull)
> anova(m2, mfull)
> SSM_full = sum((fitted.values(mfull) - mean(~BP, data=bloodp))^2)
> SSM_reduce = sum((fitted.values(m2) - mean(~BP, data=bloodp))^2)
> SSM_full - SSM_reduce
> SSE_full = sum(residuals(mfull)^2)
> SSE_full
> ((SSM_full - SSM_reduce)/4)/(SSE_full/(20-6-1))
```