Agenda

1. Interaction plots
2. Regression summary lab

Interaction plots A common way to visualize the interaction between two categorical variables is with an interaction plot.

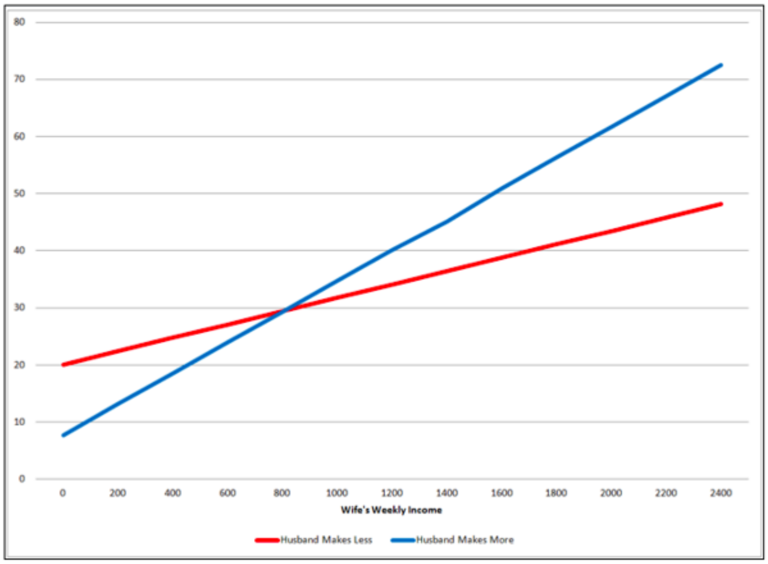

Figure 1: Figure from the Atlantic article, "Emasculated Men Refuse to Do Chores–Except Cooking."

As an example, consider Figure 1, from `https://www.theatlantic.com/health/archive/2016/10/the-only-chore-men-will-do-is-cook/505067/`

1. How can we interpret this plot?

2. What would the R code associated with this model look like?

3. What would you expect the fitted coefficients to be like on the model?

For another example, lets think back to the education data we keep considering

```
> require(openintro)
> with(hsb2, interaction.plot(ses, gender, math))
```

1. How can we interpret this plot?

2. What would the R code associated with this model look like?

3. What would you expect the fitted coefficients to be like on the model?

Regression summary lab Finally, let's do the regression summary lab

```
> 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))
> # Estimate the intercept, using the fact that the means
> # define a point on the regression line
> regdata %>%
+   mutate(beta0 = meanY - beta1 * meanX)
> predict(m1, newdata=data.frame(displ=mean(~displ, data=myCars)))
> mean(~hwy, data=myCars)
> # We're going to need differences from the mean down the line, so lets start by computing them
> 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))
```