The purpose of this section is twofold: (1) Basic review of the interpretation of coefficients on common, linear models that we will encounter in class. (2) Introduction and quick review of useful `R` code to help with problem sets. The section notes are an open-source, Github project, which can be found here. The text is an `org-mode` document, which can be compiled as HTML or LaTeX. The text is interactive, if you are an Emacs user: you can run the `R` code from within the document, and then compile the results immediately. If you are interested in contributing to this project, see the readme on the code repository, found here:

`https://github.com/danhammer/applied-metrics`

Interpretation of Coefficients

The interpretation of the coefficients in a linear model depends on the functional form of the covariates. A common specification involves the logarithms of the dependent and independent variables. We will review each of the four cases below. Note that a percentage change in a variable z is defined as $\%\Delta z = (100 \cdot dz)/z$.

1. **Linear-linear**: $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$. Then $dy_i = \beta_1 dx_i$ and $\beta_1 = dy_i/dx_i$. A one unit change in x_i will induce a one unit change in y_i.

2. **Log-linear**: $\log y_i = \beta_0 + \beta_1 x_i + \epsilon_i$. Then $d \log y_i = \beta_1 dx_i$. Note that for small changes in y_i, $d \log y_i \approx (1/y_i) dy_i$, such that $(1/y_i) dy_i \approx \beta_1 dx_i$ and $[1/(100 \cdot y_i)] dy_i \approx 100 \cdot \beta_1 dx_i$. A one unit change in x_i generates a $\beta_1 \times 100$ percentage increase in y_i. This model form is often used to estimate the impact of the returns to education: a one year increase in education leads to a $\beta_1 \times 100$ percent increase in wages.

3. **Linear-log**: $y_i = \beta_0 + \beta_1 \log x_i + \epsilon_i$. Then $dy_i = \beta_1 d \log x_i$. Just as in the log-linear case, for small changes in x_i, $d \log x_i \approx (1/x_i) dx_i$. Thus, $dy_i \approx \beta_1 (dx_i/x_i)$, which implies that a 1% change in x_i results in a $\beta_1/100$ unit change in y_i.

4. **Log-log**: $\log y_i = \beta_0 + \beta_1 \log x_i + \epsilon_i$, such that $d \log y_i = \beta_1 d \log x_i$. Using the results from the previous model specifications, it follows that $\beta_1 = (dy_i/y_i)/(dx_i/x_i)$. The interpretation of $beta_1$ is the elasticity of y_i with respect to x_i. A one percentage change in x_i will, on average, yield a β_1 percentage change in y_i.

Tables and figures in R

If you are not familiar with `R`, then it might be useful to review the section notes for ARE212, which can be found here:

`https://github.com/danhammer/ARE212`

The ARE212 project gives a more rigorous introduction to econometrics in `R`. Here, we will review some basic commands to output LaTeX-ready tables and figures. First, we will read in the data, which has been saved as a `.dta` Stata file. We will need to import the `foreign` package to directly read the data set, without re-saving it to a more common format.

```
library(foreign)
data <- read.csv("../../../resources/ps1.csv")
```

Ulitmately, we will try to estimate the impact of maternal smoking on infant birthweight. We can pare down the rather large data set into one that represents only necessary variables for this impact analysis; specifically, we will look at the infant birthweight (`dbrwt`), maternal age (`dmage`), maternal education(`dmeduc`), paternal eduaction (`dfeduc`), maternal cigarette usage (`cigar`), and marital status of mother (`dmar`). We will create a binary variable `smoker` to indicate whether the mother used any cigarettes during pregnancy, and we will drop observations with missing values for cigarette use:

```
var.names <- c("dbrwt", "dmage", "dmeduc", "dmar", "dfeduc", "cigar")
good.vals <- data$cigar != 99
sm.data <- data[good.vals, var.names]
sm.data$smoker <- ifelse(sm.data$cigar > 0, 1, 0)
nrow(sm.data)
mean(sm.data$smoker)
```

```
 [1] 119384
 [1] 0.165399
```

Of the 119,384 observations in the sample, approximately 16.5% were associated with maternal smoking – bad for the babies, but good for variation of the *treatment*. We can run a few basic regressions, appending the results into a single summary table, presented in Table 1.

```
library(texreg)
model1 <- lm(dbrwt ~ smoker, data=sm.data)
model2 <- lm(dbrwt ~ smoker + dmage + dmeduc, data=sm.data)
model3 <- lm(dbrwt ~ smoker + dmage + dmeduc + dmar + dfeduc, data=sm.data)
table.string <- texreg(list(model1, model2, model3),
                       caption = "regression output",
                       float.pos = "b!",
                       label = "fig:regout",
                       use.packages = FALSE)
```

If you want to save the output as a LaTeXfragment in order to use the `\input{}` command (e.g., `\input{regout.tex}`) then you can add the following lines to save it to the appropriate file. Otherwise, you can copy and paste the output of the `texreg` into your `.tex` file.

```
out <- capture.output(cat(table.string))
cat(out, file="regout.tex", sep="\n")
```

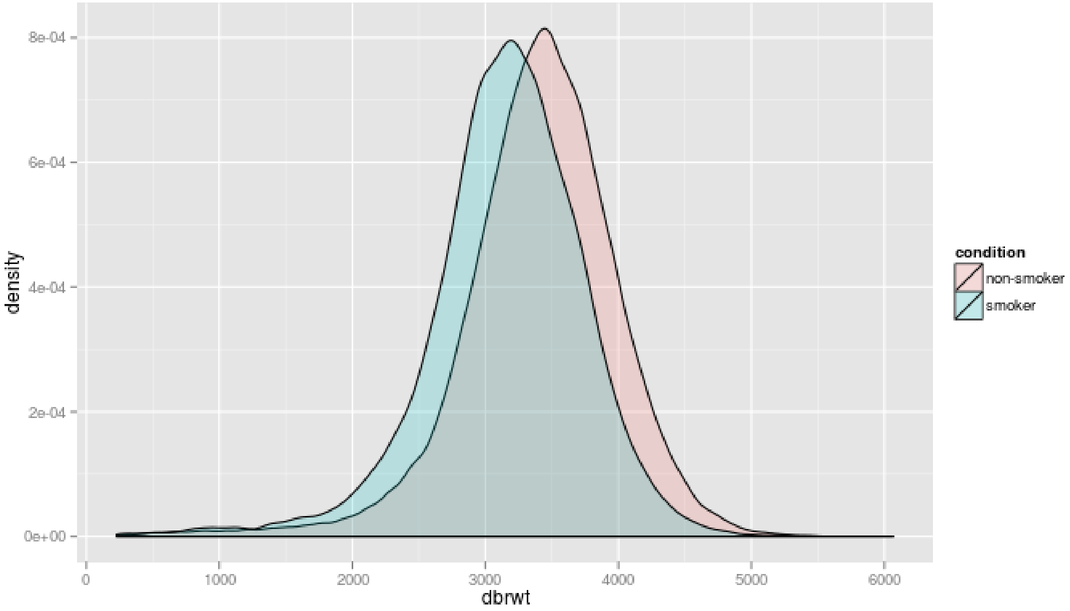
	Model 1	Model 2	Model 3
(Intercept)	3407.22^{***}	3115.57^{***}	3487.86^{***}
	(1.85)	(11.34)	(16.52)
smoker	-250.50^{***}	-226.94^{***}	-197.37^{***}
	(4.56)	(4.68)	(4.74)
dmage		6.56^{***}	1.89^{***}
		(0.33)	(0.35)
dmeduc		8.02^{***}	2.35^{**}
		(0.84)	(1.03)
dmar			-154.67^{***}
			(4.54)
dfeduc			1.64^{*}
			(0.98)
R^2	0.02	0.03	0.04
Adj. R^2	0.02	0.03	0.04
Num. obs.	119384	119384	119384

$^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$

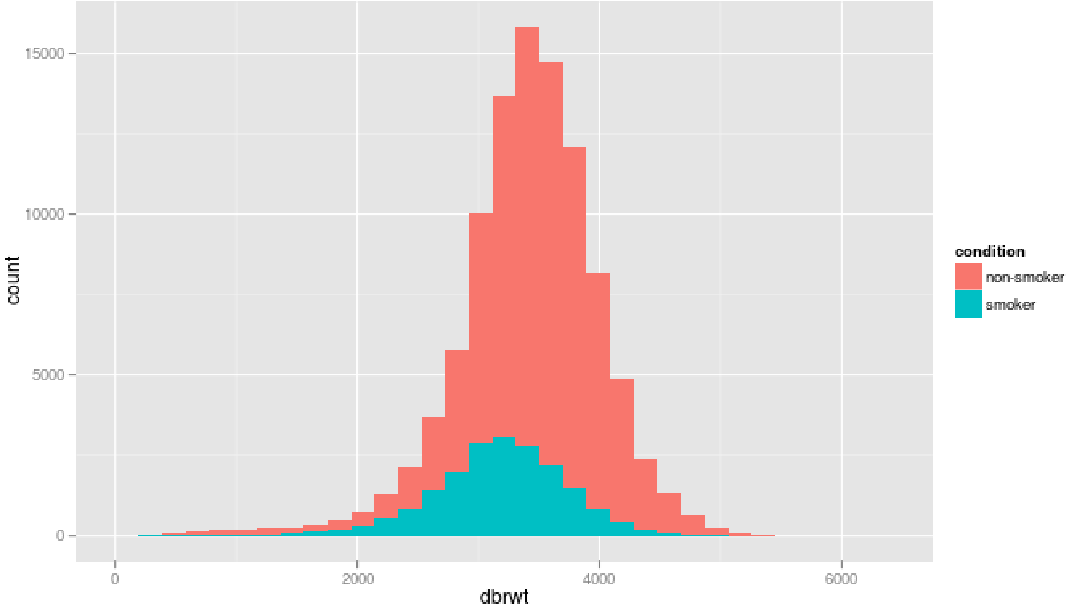
Table 1: regression output

Next, suppose we want to create a kernel density plot of infant birthweight for smoking mothers and non-smoking mothers. For this, we will rely on the very powerful `ggplot2` package. There are a lot of online resources for `ggplot2`, including (shameless pitch) the code found in the ARE212 repository. We plot the kernel densities with the `R` defaults, as well as the count histograms.

```
library(ggplot2)
sm.data$condition <- ifelse(sm.data$smoker == 1, "smoker", "non-smoker")
ggplot(sm.data, aes(x=dbrwt, fill=condition)) + geom_density(alpha=0.2)
```



```
ggplot(sm.data, aes(x=dbrwt, fill=condition)) + geom_histogram(position="identity")
```


count
dbrwt
condition
non-smoker
smoker