Linear Regression

STA 325: Lab 5, Fall 2017

Today's agenda: Linear Regression

Programming partner's: You should have a programming partner for each lab, and you should switch off who is programming, and use each other for help. We will spend about 30–50 minutes per week on lab exercises and you will be expected to bring you laptops to class to work on these exercises in class. Myself and the TA's will be in class to help you.

Lab Tasks

1. Your goal is to build a regression model for Gross National Product (GNP) based on two input variables: number of people employed and the total population using the longley data in R. Start by performing graphical exploratory data analysis: create univariate density estimates and scatterplots to understand the bivariate features of the data. Do you see anything interesting?

Before beginning the Lab we must load the package `xtable`, which will be used later.

```
require(xtable)
```

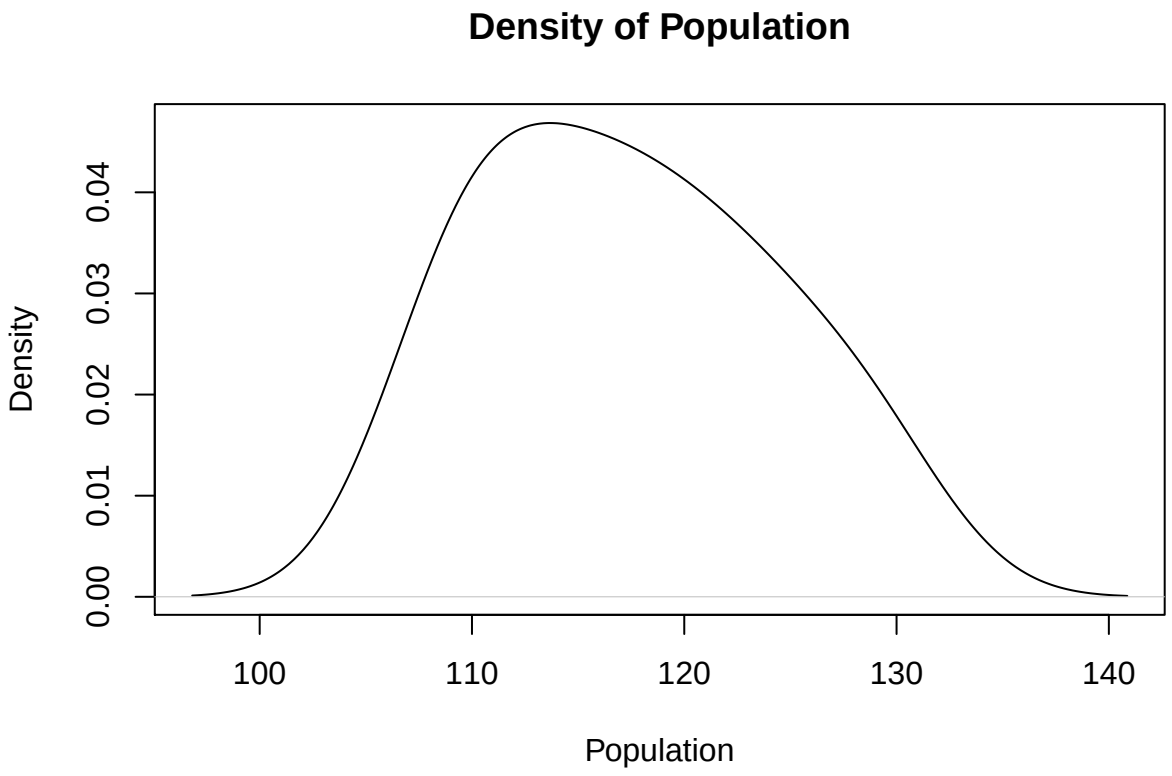
```
## Loading required package: xtable
```

The `longley` data is built into R, so it can simply be loaded.

```
data("longley")
```

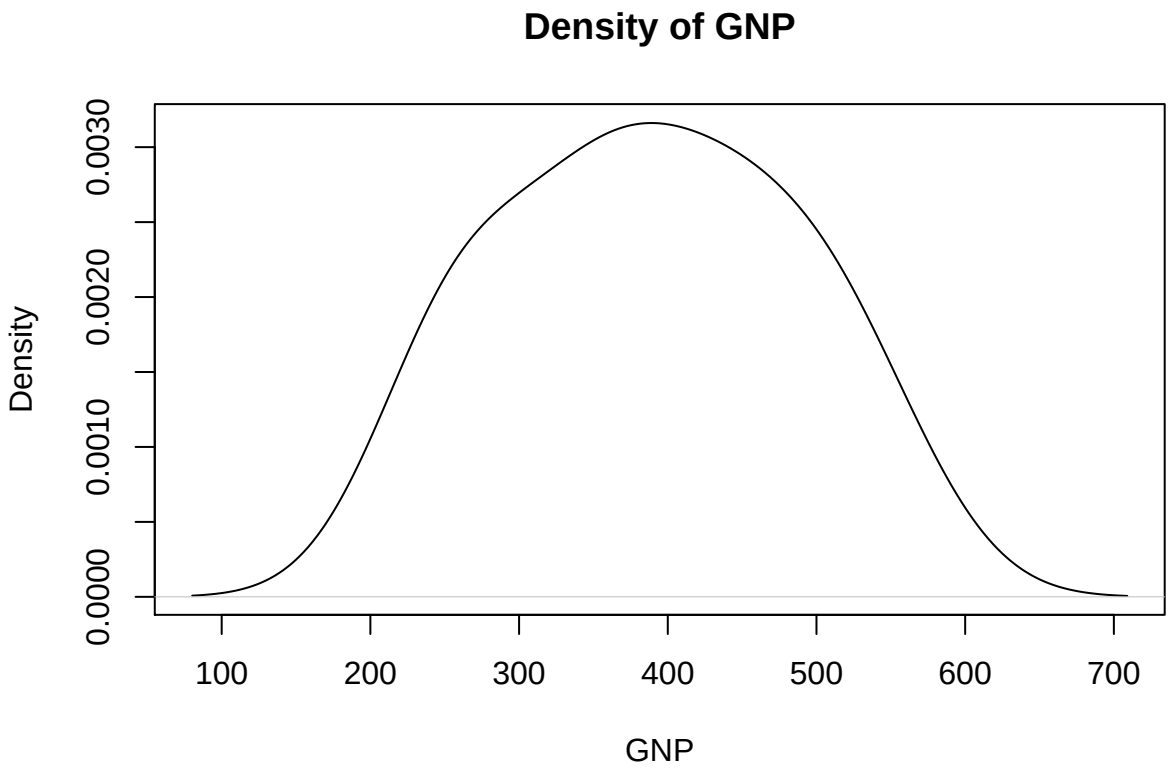
To understand the relevant variables, population, employment, and GNP, we first create density estimates for each variable using the `density` command:

```
plot(density(longley$Population), xlab = "Population", main = "Density of Population")
```



```r
plot(density(longley$Employed), xlab = "Employment", main = "Density of Employed")
```


```r
plot(density(longley$GNP), xlab = "GNP", main = "Density of GNP")
```

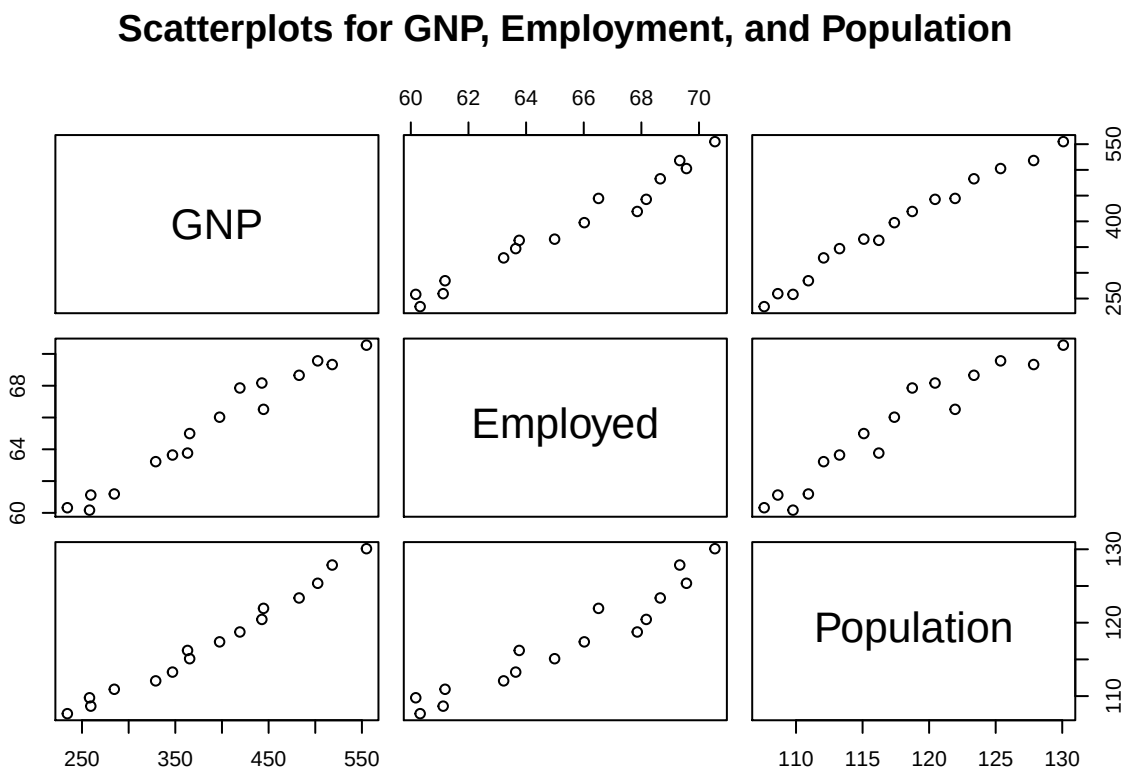


Of these variables, only GNP has a relatively normal distribution; it is symmetric and has low probability in

the tails. Population is skewed slightly to the right. Employment has a hump that disrupts the distribution, resulting in a slight skew to the left. These densities are not ideal, but they are not terrible either.

Next we create scatter plots for all combinations of these three variables.

```
pairs(~GNP+Employed+Population,data=longley,
      main="Scatterplots for GNP, Employment, and Population")
```



Both employment and population have linear relationships to GNP, which is fortuitous. There is also a slightly weaker linear relationship between employment and population. While this is not as strong it could be, it might be a sign of collinearity, which is can be detrimental to the analysis.

2. Build the regression model, and provide the coefficients of the model and details on their significance using xtable. Interpret your coefficients. Is the intercept meaningful? What can you do to make the intercept more meaningful?

First we run the desired model and store the result in `linear.model`.

```
linear.model <- lm(GNP~Population+Employed, # 2 independent vars
                   data=longley) # use longley data
```

Next we use `xtable` to display the coefficients and associated statistical data in an visually appealing table.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-1372.0954	36.1406	-37.97	0.0000
Population	8.5561	0.9837	8.70	0.0000
Employed	11.5606	1.9484	5.93	0.0000

To interpret the coefficients, it is important to first determine the units of each variable. According to the information here, `Employed` is the number of people employed in thousands, `Population` is the population in thousands, and `GNP` is the GNP in millions of dollars. The coefficient for `Employed` means that if an additional 1000 people where employed, we would expect, holding all other factors constant, GNP to rise by 11.5606 million dollars. The coefficient for `Population` means that if the population increased 1000 people,

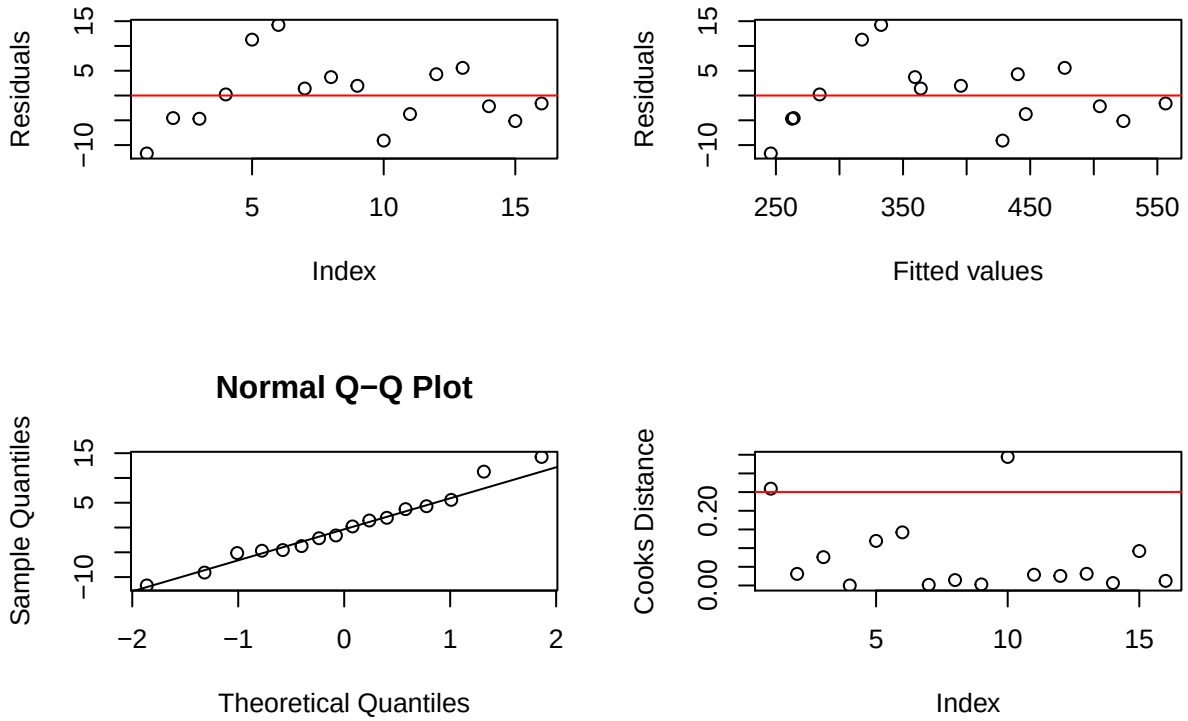
we would expect, holding all other factors constant, GNP to rise by 8.5561 million dollars. The intercept is the expected value of GNP in the country when the population is zero and no one is employed. This is obviously an impossibility and a ridiculous extrapolation from the data. Furthermore, GNP could never be less than zero. The intercept is not meaningful in this context.

One way to add meaning to the intercept would be to shift the data. For example, we could subtract the values of `Population` in 1947 (the first year in the data) from all values. If this was done for both `Population` and `Employed` (subtracting the number of people employed in 1947 from all values), then the intercept could be see as the estimated GNP when population and employment levels where equivalent to those in 1947. If this were done, the intercept term would increase to compensate for the shifts.

3. Perform regression diagnostics via graphical methods: Assess normality of your residuals, constant variance, independence as well any potentially influential points using Cook's Distance with a threshold value of $\frac{4}{n}$. Be sure to detail what you see. Do you need to transform your data? If you were to transform your data, how would it impact the interpretation of your model?

We now create a variety of plots to examine the residuals for deviations from the assumptions of OLS.

```
par(mfrow = c(2,2))
plot(linear.model$res, ylab = "Residuals") # plot residuals vs index
abline(0, 0, col = "red") # add line at y=0
plot(fitted(linear.model),linear.model$residuals, # plot res vs fitted values
     ylab = "Residuals", xlab = "Fitted values") # label axes
abline(0, 0, col = "red") # add line at y=0
qqnorm(linear.model$residuals) # create QQ plot
qqline(linear.model$residuals) # add line to QQ plot
plot(cooks.distance(linear.model), # plot Cook's distanvce v residuals
     ylab = "Cooks Distance") # label axis
abline(4/nrow(longley), 0, col = "red") # add line at 4/n, which is level for outlier
```



The plot in the upper left corner plots residuals versus their index. As the data is time series data, we are looking for *tracking*, which occurs when neighboring residuals have similar values. There may be slight evidence of tracking (otherwise known as auto-correlation). This is something that could be explored further.

The plot in the upper right hand corner plots the residuals versus the predicted value. This plot can be used to check the assumption that the residuals are from a distribution with mean zero. Ideally, there would be no tendency to be above or below the line $y = 0$, which does indeed seem to be the case here. This graph can also be used to check for homoskedasticity. The spread of the residuals should be stable across all fitted values. In this case, there seems to be a trend; the residuals are larger for smaller fitted values, which suggests heteroskedasticity. Admittedly, the sample size is small, so more exploration is needed. To fix this, one would transform the data in a way that compresses small values or stretches large values, such as squaring the data.

The Q-Q plot ideally has all values clustered along the diagonal, which is what is observed in the plot. The diagonal represents where theoretical quantiles correspond to observed quantiles. In other words, if values are close to the diagonal, then observed values are in the position in the distribution close to where they are theoretically hypothesized to be. This is evidence that the residuals indeed follow a normal distribution (which is the hypothetical distribution in this case).

The plot in the bottom right corner shows Cook's distance for all data points. The horizontal line represents the values $4/n$, which is the cutoff used to determine outliers. In the plot, it is clear the the first and tenth points could be considered outliers and potentially removed from the data set.

4. Create a plot of Population against GNP that shows the fitted regression line holding Employment at its mean value. Add prediction and confidence intervals to your plot based on the same assumption in different colors. Where are the intervals narrowest? What do you expect will happened to the intervals as $n \to \infty$.

First we compute the mean of `Employed`.

```
mean.emply <- mean(longley$Employed)
```

Next we create a new data frame containing points that will be used for prediction. The `Population` data is placed in the data frame without modification. For `Employed` we replace all data points with the mean.

```
new.data <- data.frame(Population = longley$Population, # population data
                       Employed = mean.emply) # column w/mean of Employed
```

Next we determine the slope and intercept coefficients for a model where `Population` is independent and `Employed` is fixed at its mean. In this case, the new intercept is the sum of the old intercept and the slope parameter of `Employed` multiplied by its mean. Symbolically, we go from

$$\text{GNP} = \beta_0 + \beta_1 \text{Population} + \beta_2 \text{Employed}$$

to

$$\text{GNP} = (\beta_0 + \beta_2 \hat{\mu}) + \beta_1 \text{Population},$$

where $\hat{\mu}$ is the sample mean of `Employed`.

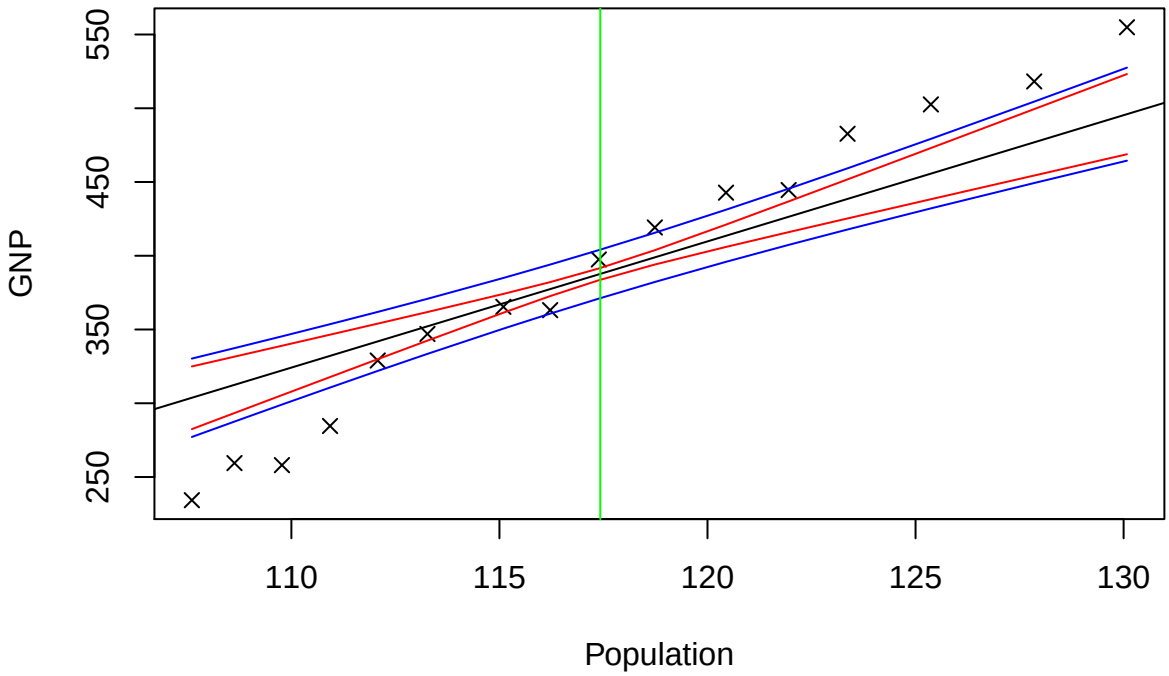
```
reg.coefs <- c(linear.model$coefficients["(Intercept)"]+ # add intercept
  linear.model$coefficients["Employed"]*mean.emply, # to slope coef*mean
  linear.model$coefficients["Population"]) # add population
```

We also need to compute the confidence and prediction intervals for each point, which is done with the `predict` command. To get the intervals with `Employed` set to the sample mean, we use the data set `new.data` constructed above.

```
con.int <- predict(linear.model, # for linear.model
                   newdata = new.data, # use constant value for Employed
                   interval = 'confidence') # construct confidence interval
pred.int <- predict(linear.model, # for linear.model
                    newdata = new.data, # use constant value for Employed
                    interval = 'predict') # find prediction interval
```

We are now ready to construct the desired plot.

```r
plot(longley$Population, longley$GNP, pch = 4, # create scatter plot
     xlab = "Population", ylab = "GNP")
abline(coef = reg.coefs) # add reg line for fixed Employed
lines(x = longley$Population, # same x scale
      y = con.int[,"lwr"], # add lower bound for confidence interval
      col = "red") # make red
lines(x = longley$Population,
      y = con.int[,"upr"], # add upper bound for confidence interval
      col = "red")
lines(x = longley$Population,
      y = pred.int[,"lwr"], # lower bound for prediction interval
      col = "blue") # predition interval is blue
lines(x = longley$Population,
      y = pred.int[,"upr"], # upper bound for prediction interval
      col = "blue") # prediction interval is blue
abline(v = mean(longley$Population), # add vertical line at mean of Pop
       col="green") # make line green
```



The solid black line is the fitted regression holding `Employed` at its mean values. The red lines represent the confidence intervals. The blue lines represent the prediction intervals. As expected, the prediction intervals are wider than the confidence intervals. The green vertical line represents the mean, and is added to corroborate the idea that both the confidence and the prediction intervals are narrowest at the mean of `Population`.

There are several problems with this plot. One problem is that the residuals are not properly distributed. Below the mean of `Population` we consistently over predict, while above the mean we consistently under predict. This is indicative of a problem with our regression. Furthermore, many observations are outside of the prediction intervals, which also suggests a problem with our model.

As $n \to \infty$, the confidence intervals will shrink to point estimates. This comes from the Law of Large Numbers, which says that the variance of a sample mean follows σ^2/n asymptotically. This goes to zero as $n \to \infty$, so the confidence interval can be made arbitrarily small by taking n sufficiently large. The prediction interval will shrink, but it cannot be made arbitrarily narrow. As n increases the variance of the sample

mean approaches zero, but the variance of a single observation does not. The variance decreases, but there is a non-zero lower bound if $\sigma^2 \neq 0$.

5. Please complete the following exercises in ISL: 2 and 5.

TA's should add solution's to these.