Basic Statistics Review – Linear Regression

At this point you should have at least a little experience with R and have had a brief review of some descriptive statistics. I have found that R makes easy things hard, but also makes hard things easy. We'll try to make a few hard things easy in the context of simple inferential statistics and then move on to time series analysis in a more developed way.

Objectives:

Perform a simple linear regression with R

- plot time series data
- fit a linear model to a set of ordered pairs
- assess normality of residuals
- test whether the slope of the fitted line is zero

Mauna Loa Atmospheric CO2 Concentration

The basic R installation makes quite a few data sets available to us. We are interested in looking at atmospheric carbon dioxide concentration during some of the last several decades. If you type *co2* in the command window, you will see

```
            Jan       Feb       Mar    …

  1959     315.42    316.31    316.50 …
```

And so on. Starting from January of 1959, we see values of atmospheric concentrations together with when they occurred. To obtain a description, type:

help(co2)

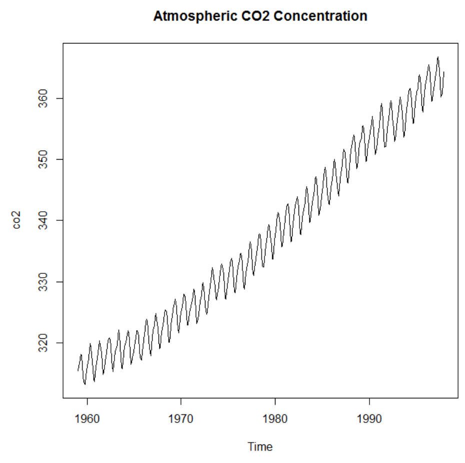
Atmospheric concentrations of Carbon Dioxide (CO2) are expressed in parts per million (ppm) and reported in the preliminary 1997 SIO manometric mole fraction scale.

Now *co2* is a "time series" object. This type of object will be hugely important to us moving forward. To verify that *co2* is this type of object, type *class(co2)*

R will tell you that the class is "ts", or “time series”. This means that we have available to us data which occurs at equally spaced points in time. It includes *observations* together with a set of *descriptions* (start time of the series, end time, frequency, etc.). There are also available ***methods*** for operating on time series objects. We will worry about how to create *ts* objects later. For now, just enjoy the convenience of being able to type

plot(co2, main='Atmospheric CO2 Concentration')

R makes assumptions about how you would like to plot your time series data. You should see the following.



CO2 concentration is apparently increasing with time over this period. Also, even though a straight line obviously misses some crucial behavior it isn’t entirely useless in that it can be used to model the *trend* in the data.

We’ll follow the standard notational conventions and make a few minimal assumptions.

- The response (i.e. co2 concentration) of the i^{th} observation may be denoted by the random variable Y_i.
- This response depends upon the explanatory variable x_i in a linear way, with some noise added, as

$$Y_i = linear\ model\ plus\ noise = (\beta_0 + \beta_1 x_i) + \epsilon_i$$

The error term ϵ_i can arise in a variety of ways: measurement error, lack of knowledge of other important influences, etc. When doing a simple regression model, we make the (often reasonable!) assumptions that

a) the errors are normally distributed and, on average, zero;
b) the errors all have the same variance (they are homoscedastic), and
c) the errors are unrelated to each other (they are independent across observations).

Written mathematically (with the third assumption relaxed somewhat) for normally distributed errors

$$E[\epsilon_i] = 0$$

$$Var[\epsilon_i] = \sigma^2\{\epsilon_i\} = constant = \sigma^2$$

$$Cov[\epsilon_i, \epsilon_j] = \sigma\{\epsilon_i, \epsilon_j\} = 0, \qquad \forall\, i \neq j$$

Even more compactly

$$\epsilon_i \overset{iid}{\sim} N(\mu = 0,\ \sigma^2 = constant)$$

Let's make a distinction here, especially since this data set violates the assumptions we often make about the errors (which is why we find it interesting from a time series perspective!). When we find estimates of the slope and intercept using, for example *O*rdinary *L*east *S*quares (OLS), we are not really making any distributional assumptions, just taking a cloud of points and finding numbers that we call a slope and an intercept that minimize a quality term like

$$Q = \sum (Observed - Predicted)^2$$

The observed value is what you have measured, the predicted value is sitting on your straight line. A common notation would be

$$Y_i = i^{th}\ observed\ response\ variable$$

$$\hat{Y}_i = i^{th} predicted\ response\ variable = slope \cdot x_i + intercept$$

Using linear algebra or calculus we can find formulas for the slope and intercept as

$$slope = b_1 = \frac{SS_{xy}}{SS_{xx}} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})(x_i - \bar{x})}$$

$$intercept = b_0 = \bar{y} - b_1 \cdot \bar{x}$$

If we wish to perform some *inferences* (confidence intervals, hypothesis tests), then we need to make distributional assumptions.

We can calculate the slope and intercept values "by hand" as follows. First, to get the concentrations and times they were collected available to compute with, we can type

```
co2.values = as.numeric(co2)

co2.times = as.numeric( time(co2) )

SSxx  = sum(   (co2.times - mean(co2.times)  ) * (co2.times - mean(co2.times) ) )

SSxy  = sum(   (co2.values - mean(co2.values)  ) * (co2.times - mean(co2.times) ) )

( slope = SSxy / SSxx  )

( intercept = mean(co2.values) - slope*mean(co2.times)   )
```

You should have a slope of ***1.307*** and an intercept of ***-2249.774***. Obviously R will do all of this for us. We can create a linear model a little more simply with the command

```
co2.linear.model = lm(co2 ~ time(co2) )
```

There is a nice R command to place your fitted line on the graph. Just run

```
plot(co2, main='Atmospheric CO2 Concentration with Fitted Line')

abline(co2.linear.model )
```

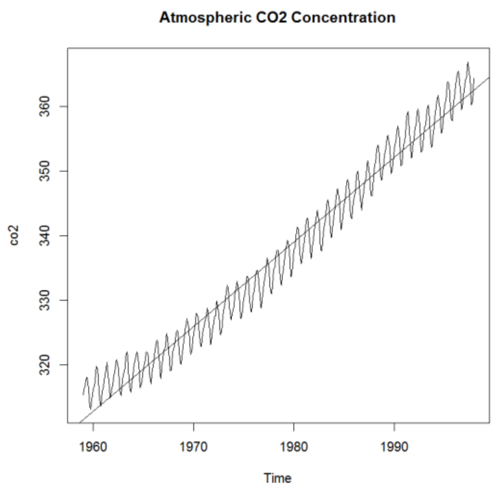
If you'd like to compute your residuals by hand, you can calculate the fitted values and subtract them from your observed responses with a command like

```
co2.fitted.values = slope*co2.times + intercept

co2.residuals = co2.values - co2.fitted.values
```

You might be a little lazier and would like R to do this for you. We "interrogate" the linear model with a command like

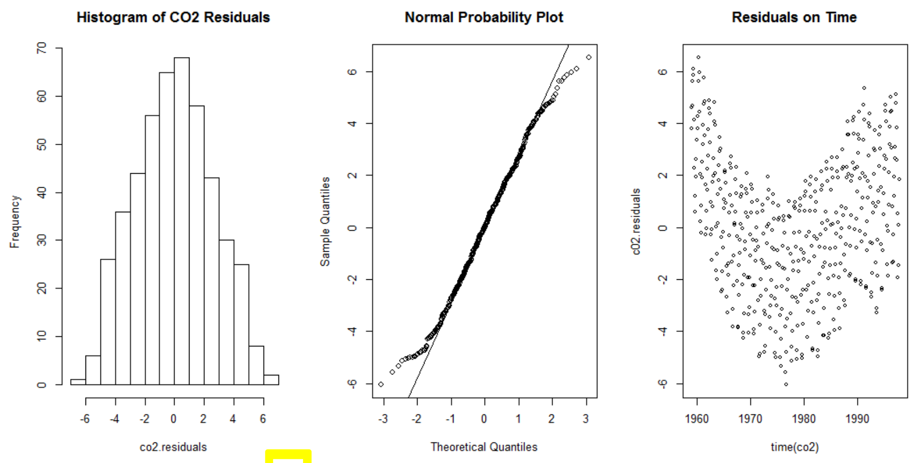
```
( co2.residuals = resid( co2.linear.model ) )
```



Do you believe that the residuals are normally distributed? When we have a large data set we can look at a histogram. When a data set is smaller, we can look at a normal probability plot. Without worrying about the details of the plot construction right now, systematic departures from a straight line in a probability plot indicate likely departures from normality. It is also common to plot residuals against the time variable (or the predicted variable).

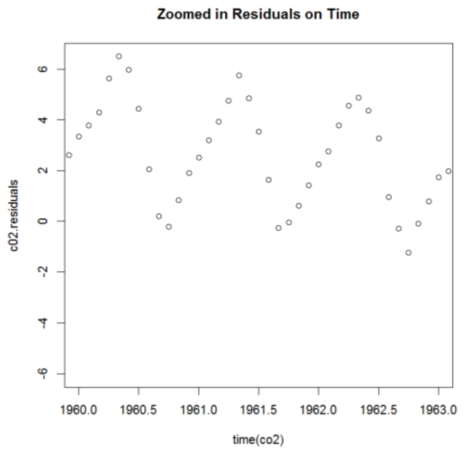
Using the *par()* command to see these plots together, we can see a systematic departure from normality in the tails, along with an obvious structure in the time plot indicating departures from the standard regression assumptions.

```
par(mfrow=c(1,3))
( c02.residuals = resid( co2.linear.model ) )
hist(co2.residuals, main= "Histogram of CO2 Residuals")
qqnorm(c02.residuals, main= "Normal Probability Plot")
qqline(c02.residuals)
plot(c02.residuals ~ time(co2), main="Residuals on Time")
```



It seems silly not to also zoom in on the residuals using the "xlim" argument to see the seasonality in the data set. We will address the oscillations in the data later in the course.

plot(c02.residuals ~ time(co2), xlim=c(1960, 1963), main="Zoomed in Residuals on Time")



Another Example

A very famous data set was discussed in William Gossett's (he's better known as "Student") classic paper *The Probable Error of a Mean* (Student, 1908). Gossett describes an experiment in which two sleeping aides are administered during the course of a clinical trial to a group of 10 research subjects. The average number of hours of sleep *gained* (called by the variable name "*extra*") under the use of each of the two drugs (indicated by the variable name "group") are recorded.

When R starts it has a number of packages already loaded. This data set should be available to you just by typing, at the R command console,

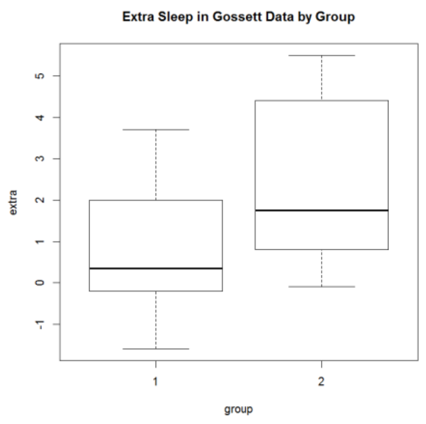
sleep

Recorded for us is the *extra sleep* obtained as a consequence of taking the drug ("extra"), a label telling us which drug was taken ("group"), and a label for which of the 10 people we are talking about ("ID"). As always, to get look at the data and produce the default plot, first type *sleep*
You should see

```
        extra       group      ID
1        0.7          1         1
2       -1.6          1         2
                 …
                 …
20      3.4           2        10
```

Plot your data with the command

plot(extra~group, data=sleep, main = "Extra Sleep in Gossett Data by Group")



An obvious research question would be: Is there a difference in the average response to each of the two drugs? Think about how you would have tested this in your elementary statistics class. We will automate the process with R.

If you are tired of typing the dollar sign to access a variable, you can "attach" the data frame as

```
attach(sleep)
```

Now we can access the variables with a little less typing. Disaggregating the extra sleep data into variables specifying those who took drug 1 and those who took drug 2

```
extra.1=extra[group==1]
extra.2=extra[group==2]
```

Run your t-test with the command

```
t.test(extra.1, extra.2, paired=TRUE, alternative="two.sided")
```

Since our goal is to review hypothesis testing in the context of R, we will be brief. The output is as follows:

```
        Paired t-test


data:  extra.1 and extra.2
t = -4.0621, df = 9, p-value = 0.002833
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.4598858 -0.7001142
```

```
sample estimates:
mean of the differences
                  -1.58
```

The statistic we have used is the Student-t for the paired differences $x_d = x - y$

$$t = \frac{\bar{x}_d - 0}{s_d / \sqrt{n}}$$

The rather large negative t-value leads us to believe that the visual evidence in the box plot is supported by the statistical approach. The p-value is quite small and our data are significant at the customary $\alpha = 0.01$ level.

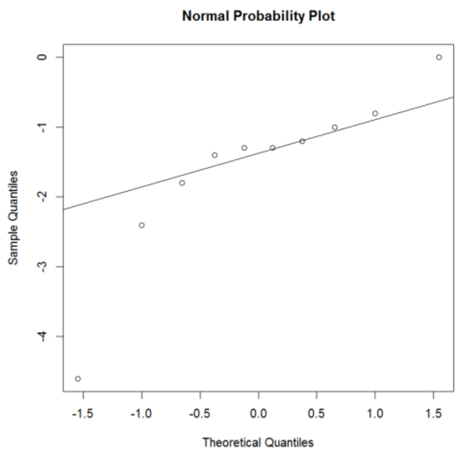
$$p < \alpha \;\Rightarrow reject$$

The sample average difference obtained from these ten individuals estimates the population parameter μ_d. An interval estimate provides a quality statement and is more interesting than the point estimate of $\bar{x} - \bar{y} = -1.58$. At the 95% level of confidence we make the traditional statement that, whatever the actual difference in means, our interval estimate is

$$95\%\ CI: \quad -2.4598858 < \mu_d < -0.7001142$$

Of course, for the test to be meaningful, we assume that the population of differences is normally distributed.

```
diffs = extra.1-extra.2
qqnorm(diffs, main= "Normal Probability Plot")
qqline(diffs)
```

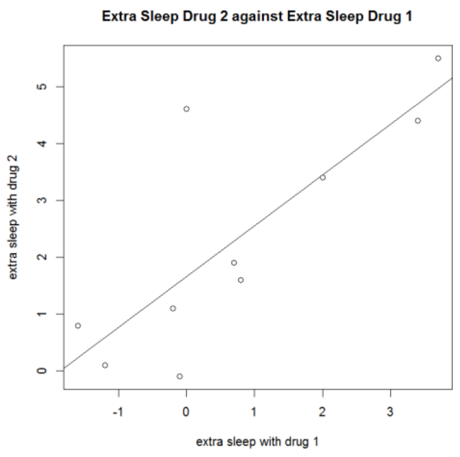


While this is not a very sensitive test for normality, the hypothesis test itself is robust to departures from normality and, while that outlier is somewhat troubling, we can move forward.

A Regression Model

Let's think about all this in another way. Since the data are paired (two different responses by same subject to each drug), it is natural to think of them in an (x, y) fashion and plot our data in a scatter plot. What do we observe? I typed

```
plot(extra.2~extra.1, xlab='extra sleep with drug 1',  ylab='extra sleep with drug 2' ,
        main='Extra Sleep Drug 2 against Extra Sleep Drug 1')
sleep.linear.model = lm(extra.2 ~ extra.1 )
abline(sleep.linear.model)
```



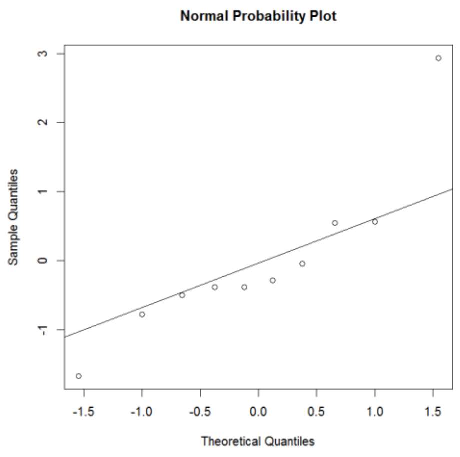
We have changed our focus from (a) deciding whether the group averages differ to (b) thinking about whether we might say something about a person's response to the second drug given their response to the first drug. Aside from the one pesky data point (subject 9) it looks like we might be able to do a decent job.

So, in this context we might think about performing a regression of $drug_2$ *response* on $drug_1$ *response*.

You should feel pretty comfortable with the analysis at this point.

- Estimate a slope and an intercept
- Plot your data with the fitted line. Look for outliers and other violations of our assumptions.
- Examine your residuals to assess normalcy.
- Perform any tests you feel are important.

Off we go! We've already plotted and fitted. Here is the normal probability plot for the residuals. We still have that one annoying data point.



summary(sleep.linear.model)

Call:

lm(formula = extra.2 ~ extra.1)

Residuals:

```
   Min          1Q          Median      3Q          Max
-1.6735     -0.4673     -0.3365     0.3979      2.9375

Coefficients:
                  Estimate    Std. Error   t value     Pr(>|t|)
(Intercept)   1.6625          0.4452       3.734       0.00575 **
extra.1      0.8899           0.2399       3.709       0.00596 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.288 on 8 degrees of freedom
Multiple R-squared:  0.6323,    Adjusted R-squared:  0.5863
F-statistic: 13.76 on 1 and 8 DF,  p-value: 0.005965
```

Our model is then

$$extra.2 = 0.8899 \cdot extra.1 + 1.6625$$

On the standard test of whether the slope is zero, our p-value is 0.00596 < .01 and so we believe the slope is not zero (at, say, the $\alpha = 0.01$ los)

See if you feel comfortable answering the following questions.

1. What do you predict for sleep gained with drug 2 if you know the sleep gained with drug 1 is 2 hours?

   3.4423 hours = 0.8899*2+1.6625

2. What is the residual associated with the 3rd data point?

   extra.2[3] - (0.8899*extra.1[3]+1.6625 = -0.38452

   alternatively

```
residuals = resid(sleep.linear.model)
residuals[3]
```

References

Student. (1908). The probable error of the mean. *Biometrika, 6*(20), 1-25.