Session 10: linear regression (Lab Report 4)

Background

The Food and Agricultural Organization (FAO) of the United Nations is interested in developing simple linear regression models to predict maize yields for countries across the world. In particular, the FAO would like to know whether using a socioeconomic explanatory variable (GDP per capita) or a biophysical explanatory variable (total nitrogen fertilizer used) would be more appropriate. You are tasked with preparing a report to the FAO that outlines the efficacy of the two considered regression models: maize yields as a function of GDP per capita, and maize yields as a function of total nitrogen fertilizer used.

You are asked to evaluate these two models for a specific country. In your report, you should provide recommendations as to which model the FAO should utilize to predict maize yields, and explain the statistical reasoning behind your recommendation. Furthermore, you should explain what steps the FAO should take to potentially improve the performance of the model (i.e., explain the model's flaws and how those flaws could potentially be addressed).

Data

The data are included in one file: "maize.csv". These data include:

Variable name	Description	Units
Maize	Maize yields	hg/ha
N	Nutrient nitrogen (total) applied in ag sector	tons
GDP	Gross Domestic Product per capita	US dollars

The data are collected annually at the national scale. Some countries have more observations than others.

Analysis

All analysis should be performed in R. It is suggested that the analysis be carried out in the order outlined below.

Determine whether nitrogen application or GDP per capita is a better predictor of maize yields in your assigned country. Use the `lm()` function to evaluate two maize yield models: (1) a maize yield model with nitrogen as an explanatory variable, (2) a maize yield model with GDP per capita as an explanatory variable. Complete the linear regression workflow for each of these models to comprehensively evalute which model is best. This workflow includes:

(1) **Plot your data**. Create scatterplots of maize yields vs. your explanatory variables. You should create a single ggplot figure with these two scatterplots as facetted sub-plots.

(2) **Compute your regression parameters, evaluate their signifance, and quantify** R^2. Use the `lm()` function and extract relevant information from the output.

(3) **Plot your regression line against your observed data**. Use the `predict()` function to generate your predicted maize yield values for each model. Create a separate ggplot of the observed and predicted maize yields for each model. The plot should also include the **95% confidence interval**.

(4) **Determine which (or both, or neither) model meets the assumptions of linear regression**. Create the following diagnostic plots: (1) residuals vs. predicted y and (2) residuals vs. x. Additionally, test whether your residuals are normally distributed using the QQ Correlation Test (Filliben's).

Sample analysis

Prior to analyzing the data for your specific country, evaluate these same Maize-N and Maize-GDP relationships from the *entire* dataset (you should only report values for your *assigned country* in your lab report - this initial exercise is to demonstrate how these new functions are applied). First, load the data.

```
d <- read_csv("maize.csv")
```

```
## Parsed with column specification:
## cols(
##   Country = col_character(),
##   Year = col_integer(),
##   Maize = col_integer(),
##   N = col_double(),
##   GDP = col_double()
## )
```

```
head(d)
```

```
## # A tibble: 6 x 5
##   Country      Year Maize      N   GDP
##   <chr>       <int> <int>  <dbl> <dbl>
## 1 Afghanistan  2002 29800 23446.  195.
## 2 Afghanistan  2003  8400 20320.  214.
## 3 Afghanistan  2004 16000 23383.  232.
## 4 Afghanistan  2005 12069 25096.  264.
## 5 Afghanistan  2006 26204 49032.  290.
## 6 Afghanistan  2007 26277 28205.  390.
```

Correlation `cor()`

To calculate the correlation among maize, N, and GDP, we can calcuate a *correlation matrix* using `cor()`. First, let's reduce our dataframe so that it only contains the three columns that we're interested in. This can be done using `select()`. Then we can feed this reduced dataframe into the `cor()` function.

```
# Assign reduced dataframe to a new object
dd <- d %>% select(Maize, N, GDP)
head(dd)
```

```
## # A tibble: 6 x 3
##   Maize      N   GDP
##   <int>  <dbl> <dbl>
## 1 29800 23446.  195.
## 2  8400 20320.  214.
## 3 16000 23383.  232.
## 4 12069 25096.  264.
## 5 26204 49032.  290.
## 6 26277 28205.  390.
```

```r
# Calculate correlation matrix
cor(dd)
```

```
##            Maize          N        GDP
## Maize 1.00000000 0.05612678 0.55979378
## N     0.05612678 1.00000000 0.03883332
## GDP   0.55979378 0.03883332 1.00000000
```

The values in the matrix are correlation coefficients that relate the variables specified in the column and row. For example, we can see that Maize and GDP are related by a correlation coefficient of 0.56 (value in the "Maize" column and "N" row [2,1], and vice versa [1,2]), whereas Maize and N are related by a correlation coefficient of 0.056. This matrix also shows us correlations between N and GDP (= 0.048). From these values, we can surmise that GDP will be a better predictor variable than N in our linear regression model.

Fitting a linear regression model with `lm()`

To fit the regression model, we will run `lm()`. The first argument in this function is the "formula", which we specifiy as `y ~ x`, but the "y" and "x" should be replaced with the column names that we're interseted in. We also need to specify which dataframe we're working with. This regression model should be assigned to an object, which has been named "mod" in the below code.

```r
mod <- lm(Maize ~ N, data = dd)
summary(mod)
```

```
##
## Call:
## lm(formula = Maize ~ N, data = dd)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -44804 -28262 -13577  14298 321970
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.564e+04  1.002e+03  45.558   <2e-16 ***
## N           7.998e-04  3.234e-04   2.473   0.0135 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42840 on 1936 degrees of freedom
## Multiple R-squared:  0.00315,    Adjusted R-squared:  0.002635
## F-statistic: 6.118 on 1 and 1936 DF,  p-value: 0.01347
```

From the summary, you can identify the regression parameter estimates, significance of the parameter estimates, and the R^2.

Predicting y values from the model with `predict()`

To calculate the predicted values of y given observations of x, use the `predict.lm()` function. To run this function, you need to provide the following arguments: the model (`mod`) and the explanatory variable values presented as a dataframe (`newdata = data.frame(N = dd$N)`). Note that the column name **needs to be the same** as the column name that was used to fit your model in the prior step. This is why we've specified `N = dd$N`.

```
p <- predict.lm(mod, newdata = data.frame(N = dd$N))
head(p)
```

```
##        1        2        3        4        5        6
## 45654.40 45651.90 45654.35 45655.72 45674.86 45658.20
```

p is a vector that contains predicted values of maize yield that correspond to the observed values of N. We can also use predict.lm() to calculate our confidence and prediction intervals (either interval = c("confidence") or interval = c("prediction"). We can specify what level of confidence we want using the level argument. Below, we've opted for 95%.

```
p <- predict.lm(mod, newdata = data.frame(N = dd$N), interval = c("confidence"), level = 0.95)
head(p)
```

```
##        fit      lwr      upr
## 1 45654.40 43693.33 47615.46
## 2 45651.90 43690.38 47613.42
## 3 45654.35 43693.27 47615.42
## 4 45655.72 43694.89 47616.54
## 5 45674.86 43717.47 47632.25
## 6 45658.20 43697.83 47618.57
```

```
str(p)
```

```
##  num [1:1938, 1:3] 45654 45652 45654 45656 45675 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:1938] "1" "2" "3" "4" ...
##   ..$ : chr [1:3] "fit" "lwr" "upr"
```

Note that this output is not a tibble or dataframe, though we need it to be in order to work with the values in subsequent steps of the analysis. Use data.frame() to convert p to a dataframe.

```
p <- data.frame(p)
str(p)
```

```
## 'data.frame':    1938 obs. of  3 variables:
##  $ fit: num  45654 45652 45654 45656 45675 ...
##  $ lwr: num  43693 43690 43693 43695 43717 ...
##  $ upr: num  47615 47613 47615 47617 47632 ...
```

Note that p$fit = predicted values of maize yields from our regression model, p$lwr = the lower bound of the confidence interval, p$upr = the upper bound of the confidence interval.

Plotting predicted and observed maize yields with the model confidence interval

Our observed data are in a different dataframe (dd) than our predicted data (p), so we need to merge dd and p. This can be done using bind_cols(). In the code below, this new dataframe is assigned to the object n.

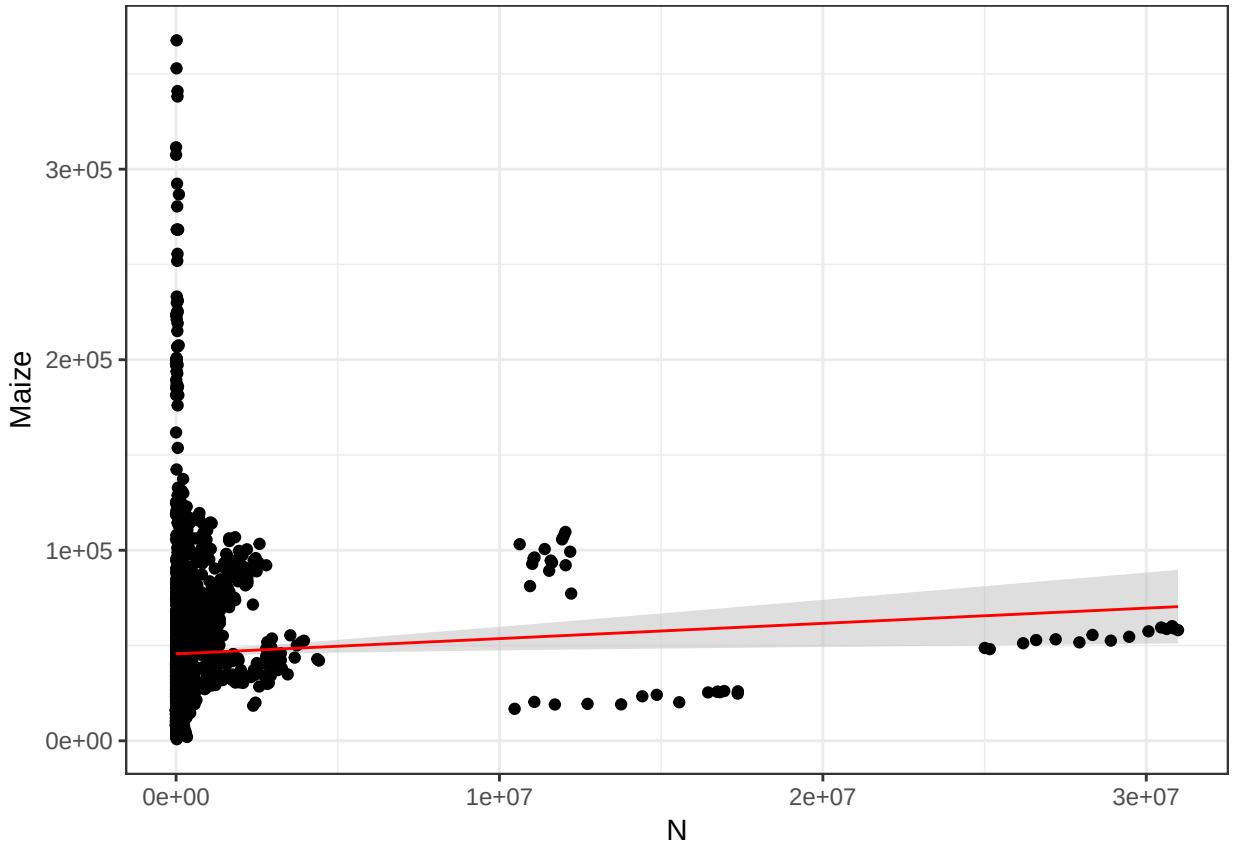
```
n <- bind_cols(dd, p)
head(n)
```

```
## # A tibble: 6 x 6
##   Maize      N   GDP    fit    lwr    upr
##   <int>  <dbl> <dbl>  <dbl>  <dbl>  <dbl>
## 1 29800 23446.  195. 45654. 43693. 47615.
```

```
## 2  8400 20320.  214. 45652. 43690. 47613.
## 3 16000 23383.  232. 45654. 43693. 47615.
## 4 12069 25096.  264. 45656. 43695. 47617.
## 5 26204 49032.  290. 45675. 43717. 47632.
## 6 26277 28205.  390. 45658. 43698. 47619.
```

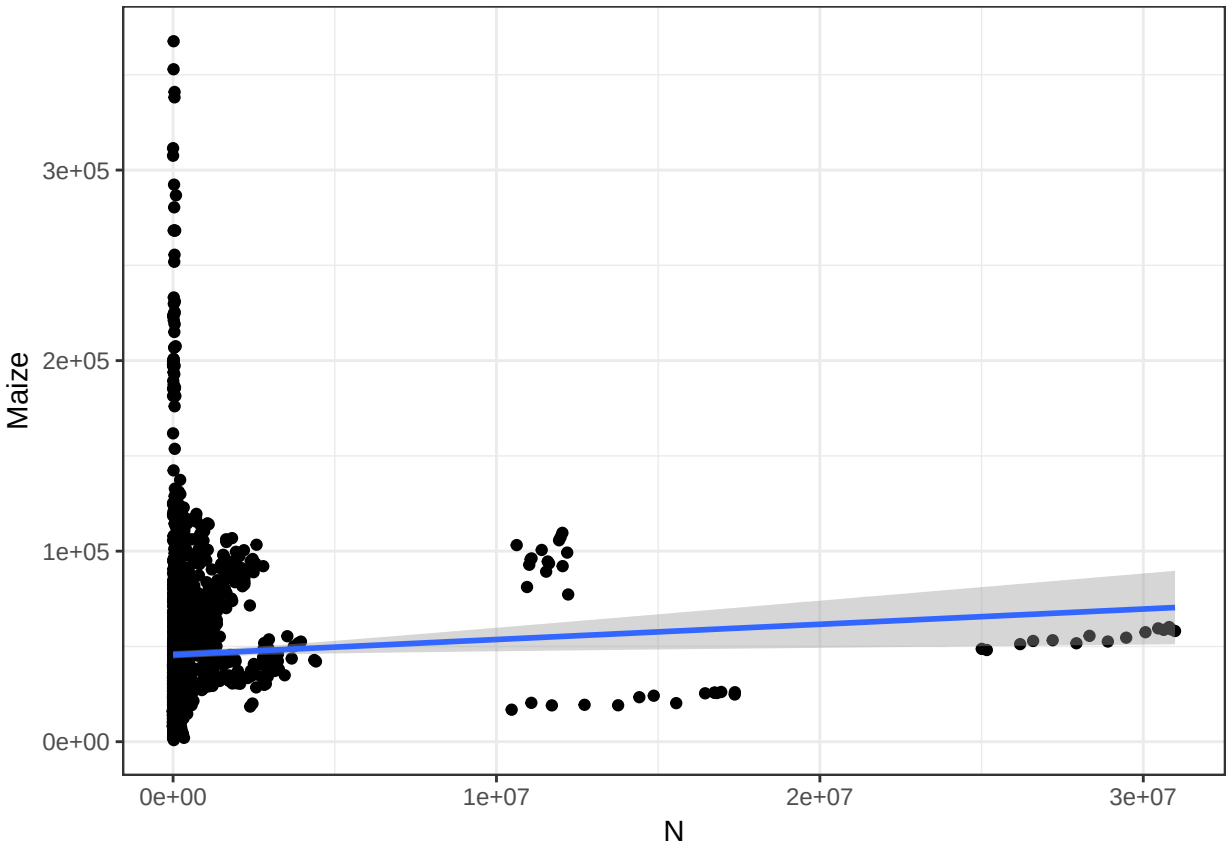
Now, we can plot the data. To add the confidence interval, the `geom_ribbon()` function is used. This function requires that a new `aes()` be specified with the lower bound of the ribbon (`ymin`) and the upper bound of the ribbon (`ymax`). Since we already have the confidence interval bounds on in our dataframe, we can assign `lwr` and `upr` to `ymin` and `ymax`, respectively. Note that ggplot will add the layers in the order that they occur in the code - so you should add the ribbon first to ensure that the ribbon doesn't cover the points or line.

```
ggplot(n, aes(x = N, y = Maize))+
  geom_ribbon(aes(ymin = lwr, ymax = upr), fill = "gray", alpha = 0.5)+
  geom_point()+
  geom_line(aes(x = N, y = fit), color = "red")+
  theme_bw()
```



There is also a pre-packaged ggplot function available for plotting the confidence interval. For your lab, you should use geom_ribbon; the code below is provided for reference.

```
ggplot(n, aes(x = N, y = Maize))+
  geom_point()+
  geom_smooth(method = lm, se = TRUE)+
  theme_bw()
```



Diagnostics

To test the residuals of your model, you will have to construct three diagnostic plots. You have the information you need to calculate the residuals (note that you will want to add a new column to your `n` dataframe with these values for ease of plotting later). Now that you understand how to calculate quantiles, you are welcome to use a shortcut function to quantify the correlation coefficient of the Normal QQ plot. This is done with the `qqnorm()` function.

```
# The qqnorm function will take your residual values, and calculate the corresponding normal quantiles
qq <- qqnorm(n$residuals)
qq
# The correlation can then be calculated by running cor() on the qq$x (residual quantiles) and qq$y (no
cor(qq$x, qq$y)
```

Report

The primary objective of your report is to summarize study, detail your methods, and offer recommendations to FAO regarding the most suitable model to use for predicting maize yields for your assigned country. Although you can work together to perform the analysis, the report **must be in your own words** and all material should be original. Any reports with evidence of plagiarism will receive no credit.

Be sure to address each of the analysis objectives in your report. Your report should be structured as follows:

- Title page - title (create your own title), name, date

- Executive abstract - summarize the report in 4-6 sentences; be sure to include your main results and recommendations, as this is the most important information included in abstracts.
- Introduction - provide all background information needed to understand the contents of the report. Be sure to *paraphrase* the writeup included here; do not copy directly.
- Methods - explain the statistical methods you used, both based on the theory and the specific application at hand. Demonstrate that you understand how and why you used the statistical principles in this case study. Include equations when relevant. **You do not have to detail the steps you took in R.**
- Results - state the results objectively. Explain what the figures and numbers you've produced from this analysis are telling you.
- Recommendations - explain to FAO whether the model you've identified is valid for use for prediction purposes, and why.
- Appendices - include your R code (should be neatly organized and professional).

Additional notes on formatting:

- All figures and tables should be captioned, numbered, and referred to in the text.
- Table captions are placed *above* the table. Figure captions are placed *below* the figure.
- Equations should be centered and included on their own lines.