Lab 7: Hypothesis Testing

August 11, 2016

1 Goals

Today in lab, we will review hypothesis testing concepts for regression

- Review concepts from hypothesis testing in regression contexts
- Test for violations of assumptions
- Examine the multiple testing problem

2 Wine Ratings vs Prices

For those of you who are not yet 21, wine is a beverage which is made from fermented grapes. Because of the long and difficult production process, wine prices can range from cheap to exorbitantly expensive. However, some critics complain that because the average wine consumer may not have a sophisticated palate, wine prices are not necessarily related to the actual wine quality. The New Yorker has an interesting write up here.

For this lab, we will be analyzing wine prices and ratings gathered from wine.com. In particular, the scores correspond to the following qualitative descriptions.

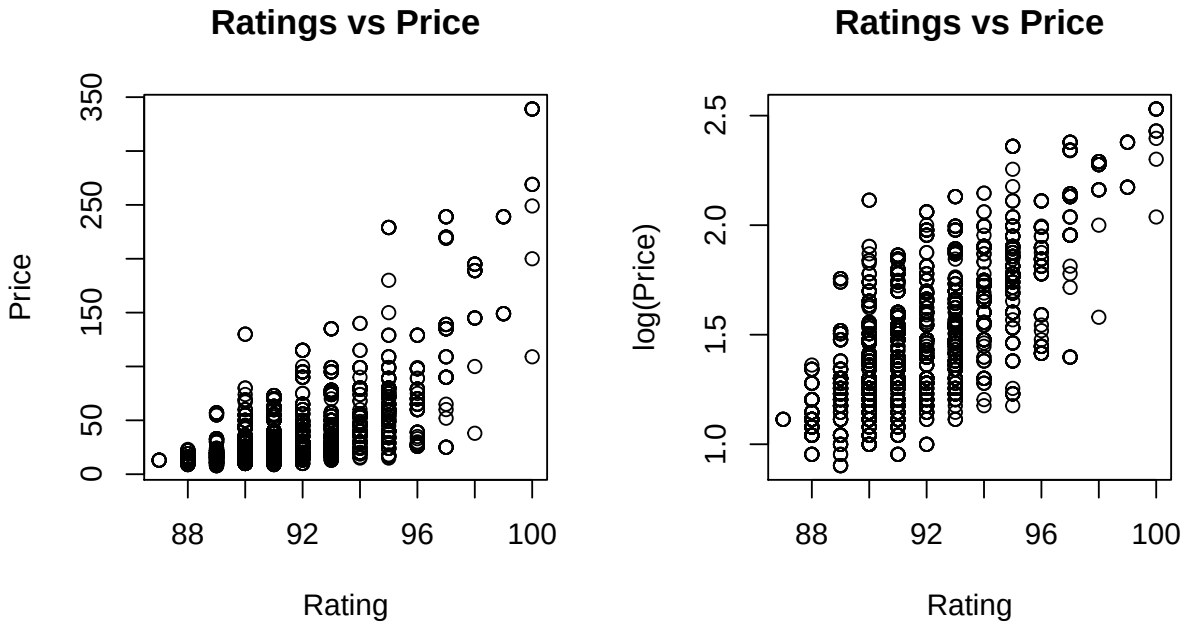
- 95-100: Classic; a great wine
- 90-94: Outstanding; superior character and style
- 80-89: Good to very good; wine with special qualities
- 70-79: Average; drinkable wine that may have minor flaws
- 60-69: Below average; drinkable but not recommended

```
# Read data and look at columns
wine <- read.csv("http://www.stat.washington.edu/~ysamwang/notes/wine.csv")
names(wine)
```

```
##  [1] "Wine.Name"       "Year"            "Grape.Variety"
##  [4] "Type"            "Region"          "Country"
##  [7] "X..Reviews"      "Original.Price"  "Discounted.Price"
## [10] "Regular.Price"   "Final.Price"     "Rating.Name"
## [13] "Rating.Score"
```

In particular, we will be using the Final.Price variable. When plotting the variables, we can see that the relationship between rating and price does not seem to be quite linear nor does it seem homoskedastic. However, taking the log transform of the price seems to produce a plot which seems closer to our assumptions.

```r
par(mfrow = c(1,2))
# plot price vs Rating
plot(wine$Rating.Score, wine$Final.Price, main = "Ratings vs Price",
     xlab = "Rating", ylab = "Price")


# plot price vs Rating
plot(wine$Rating.Score, log(wine$Final.Price, 10), main = "Ratings vs Price",
     xlab = "Rating", ylab = "log(Price)")
```



2.1 Questions

- Does the original data satisfy the linearity assumption? What about the log transformed data?
- Does the original data satisfy the normal errors assumption? What about the log transformed data?
- Does the original data satisfy the homoskedasticity assumption? What about the log transformed data?
- Based on the plot do you think there is a significant relationship between the price and rating?

2.2 Fitting the Regression Model

Now, we will actually fit a regression which assumes a model of the form

$$\text{Price}_i = a + b \times \text{Rating}_i + \epsilon_i$$

```r
# Fit linear regression
reg.model <- lm(log(Final.Price, 10) ~ Rating.Score, data = wine)
reg.model$coeff
```

```
##  (Intercept) Rating.Score
##  -7.24223284   0.09473564
```

We see that that the estimated slope parameter is slightly positive, but still very close to 0. Is this just something we see in the sample, or would we expect there to be a true relationship between rating and price? In order to perform a hypothesis test, we first, we need to estimate σ with `s.reg`

$$s_{reg} = \sqrt{\frac{SSE}{n-2}} = \sqrt{\frac{\sum_i (y_i - \hat{y}_i)^2}{n-2}}$$

```
sse <- sum((reg.model$residual)^2)

# estimate of sigma
s.reg <- sqrt(sse / (length(wine$Rating.Score) - 2))

# standard error of b.hat
se.b <- s.reg / sqrt(sum((wine$Rating.Score - mean(wine$Rating.Score))^2))
```

If we are interested in testing

$$H_0 : b = 0$$
$$H_A : b \neq 0$$

We calculate the test statistic

$$\frac{\hat{b} - 0}{se(\hat{b})} = \frac{.0947}{.00169} = 56.03,$$

which should have a reference distribution of a T distribution and since $n = 2787$, we have $n - 2 = 2785$ degrees of freedom. We could also use a normal distribution since the degrees of freedom is so large. We can see that for a .05 level test, the cutoffs are virtually the same. Note that we use .025 when calculating the cutoffs because we are using a two sided test.

```
# Note we use .025 even though the level of the test is .05 because we are using a 2-sided test
qt(.025, df = 2785)
```

```
## [1] -1.960816
```

```
qnorm(.025)
```

```
## [1] -1.959964
```

Since we want the probability of a value as or more extreme as what we actually observed, we will be looking at the area to the right of 56.03. Since we are using a two-sided test, we are interested in

$$P(|Z| > 56.03)$$

so we need to multiply the area in the tail by 2. This gives us a p-value of essentially 0.

```
# Area to the right
area.to.right <- 1- pnorm(56.03)

# p-value
area.to.right * 2
```

```
## [1] 0
```

Turns out R will calculate p-values for us automatically using the `summary` function. The estimate is given in the first column, the standard error is given in the 2nd column, the test statistic is given in the 3rd column and the p-value is in the 4th column.

```
summary(reg.model)
```

```
##
## Call:
## lm(formula = log(Final.Price, 10) ~ Rating.Score, data = wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58185 -0.12368 -0.01737  0.11384  0.82994
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7.242233   0.154775  -46.79   <2e-16 ***
## Rating.Score  0.094736   0.001691   56.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2047 on 2785 degrees of freedom
## Multiple R-squared:   0.53,Adjusted R-squared:  0.5298
## F-statistic:  3140 on 1 and 2785 DF,  p-value: < 2.2e-16
```

2.3 question

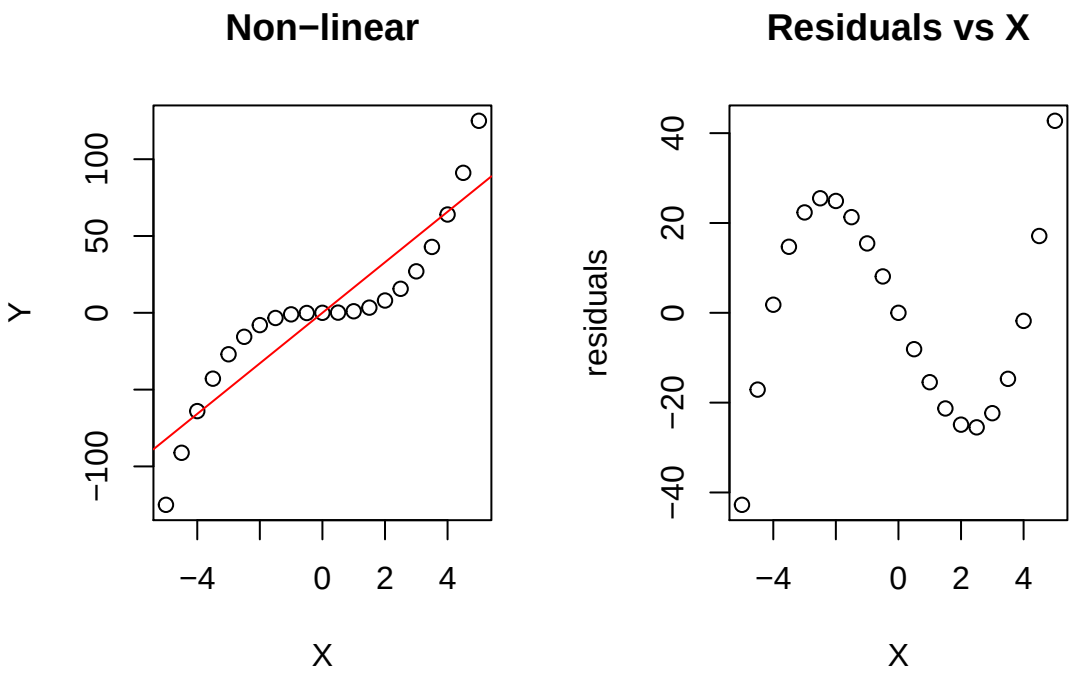
- Explain the p-value in plain English?
- What would you conclude based on the p-value?

2.4 Checking Assumptions

Now that we have fit the regression, we can examine the residuals to check the the regression assumptions. We can plot the Ratings vs the Residuals to check both the linearity and homoskedasticity assumption.

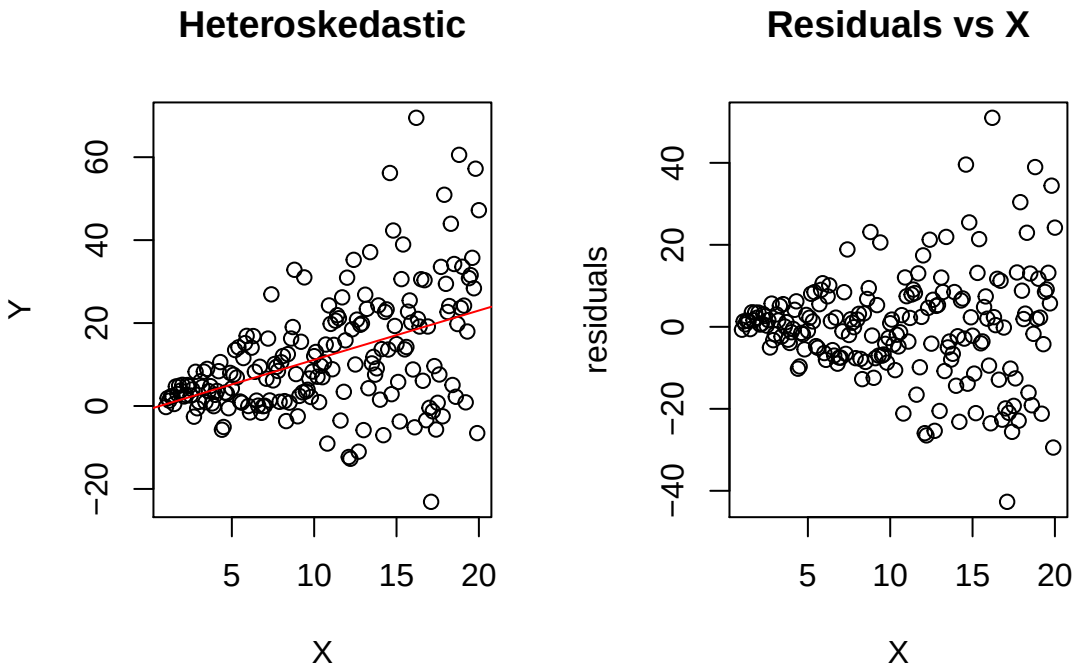
If the relationship is linear, we would expect the residuals to have no systematic patterns. Consider the example below. We see that the obvious non-linearity in X vs Y leads to a clear pattern in the residuals.

```
par(mfrow = c(1,2))
X <- seq(-5, 5, by = .5)
Y <- X^3
plot(X, Y, main = "Non-linear")
abline(lm(Y~X)$coeff, col = "red")
plot(X, lm(Y~X)$residuals, ylab = "residuals",
     main = "Residuals vs X")
```

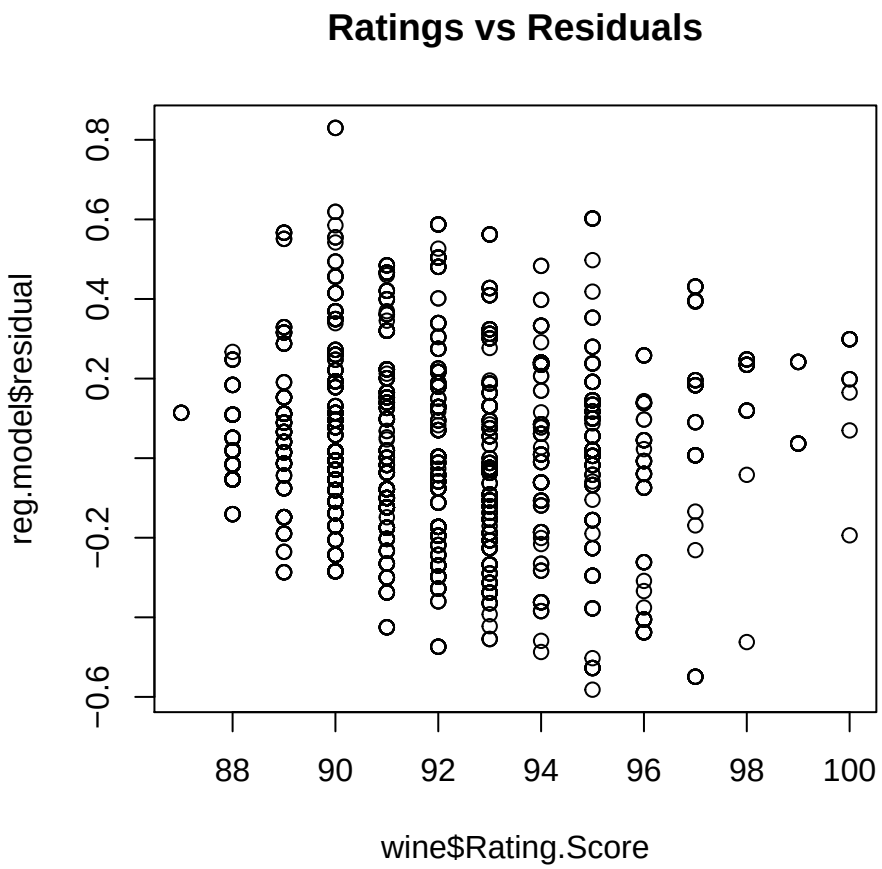


Under homoskedasticity, we would expect the variance of the residuals to stay relatively constant across all ranges of the X values. Consider the example below, when there is heteroskedasticity, we see that the residuals exhibit much larger variability with the larger values of X.

```
par(mfrow = c(1,2))
X <- seq(1, 20, by = .1)
Y <- X + rnorm(length(X), 0, 1) * X
plot(X, Y, main = "Heteroskedastic")
abline(lm(Y~X)$coeff, col = "red")
plot(X, lm(Y~X)$residuals, ylab = "residuals",
     main = "Residuals vs X")
```




```
plot(wine$Rating.Score, reg.model$residual, main = "Ratings vs Residuals")
```



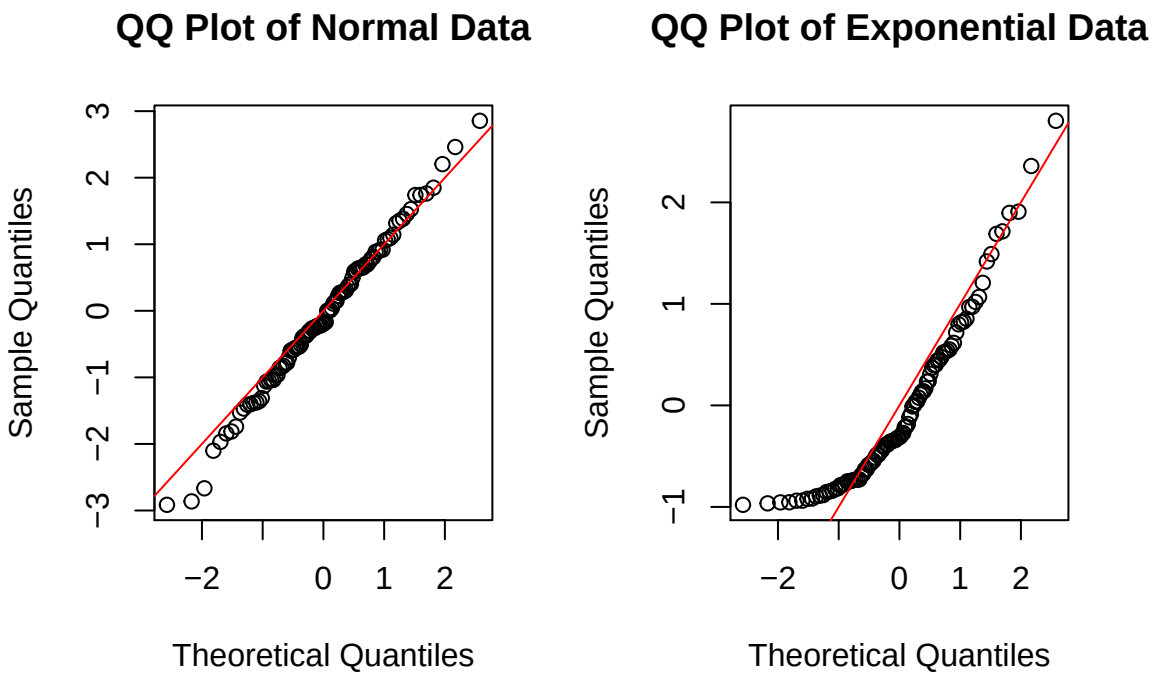
2.5 Questions

- Based on the residual plots, is the linearity assumption believable?
- Based on the residual plots, is the homoskedasticity assumption believable?

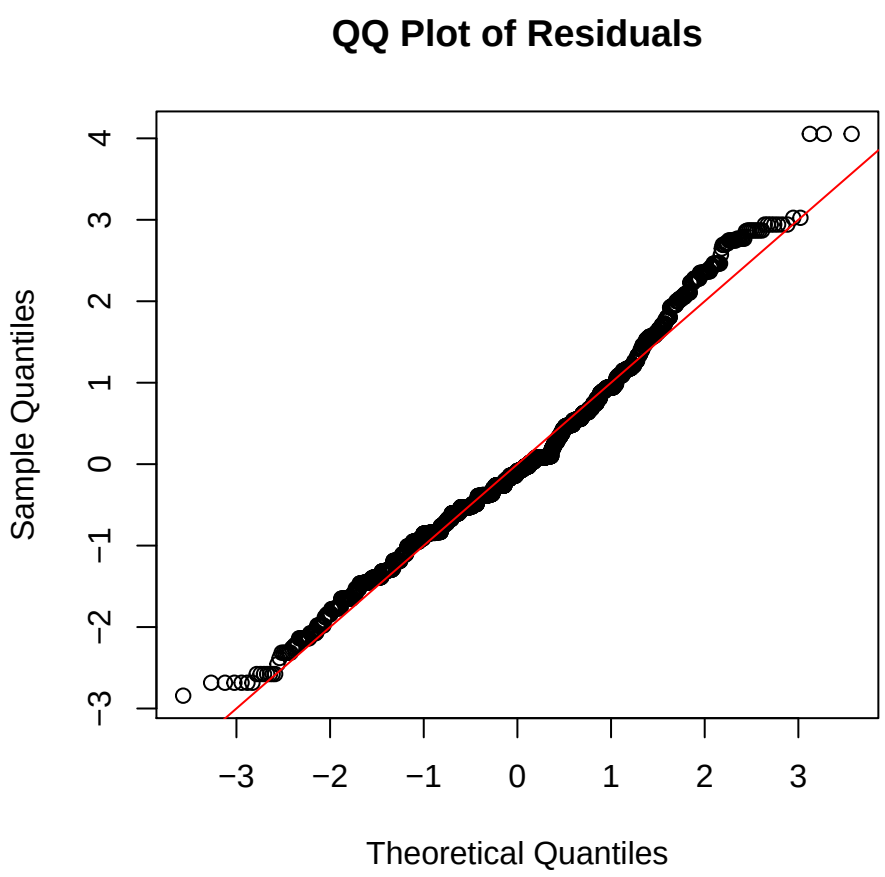
To test the normality assumption, we can plot a QQ-plot of the residuals. Recall that a QQ-plot plots the observed quantiles of the data vs the theoretical quantiles of the normal distribution. If the distribution is roughly normal, the QQ-plot should be a relatively straight line.

Consider the QQ-plots below

```
par(mfrow = c(1,2))
qqnorm(rnorm(100), main = "QQ Plot of Normal Data")
abline(a = 0, b = 1, col = "red")
qqnorm(rexp(100) - 1, main = "QQ Plot of Exponential Data")
abline(a = 0, b = 1, col = "red")
```




```r
# Standardize the residuals so they will be roughly standard normal
std.res <- reg.model$residual / sd(reg.model$residual)
qqnorm(std.res, main = "QQ Plot of Residuals")
abline(a = 0, b = 1, col = "red")
```

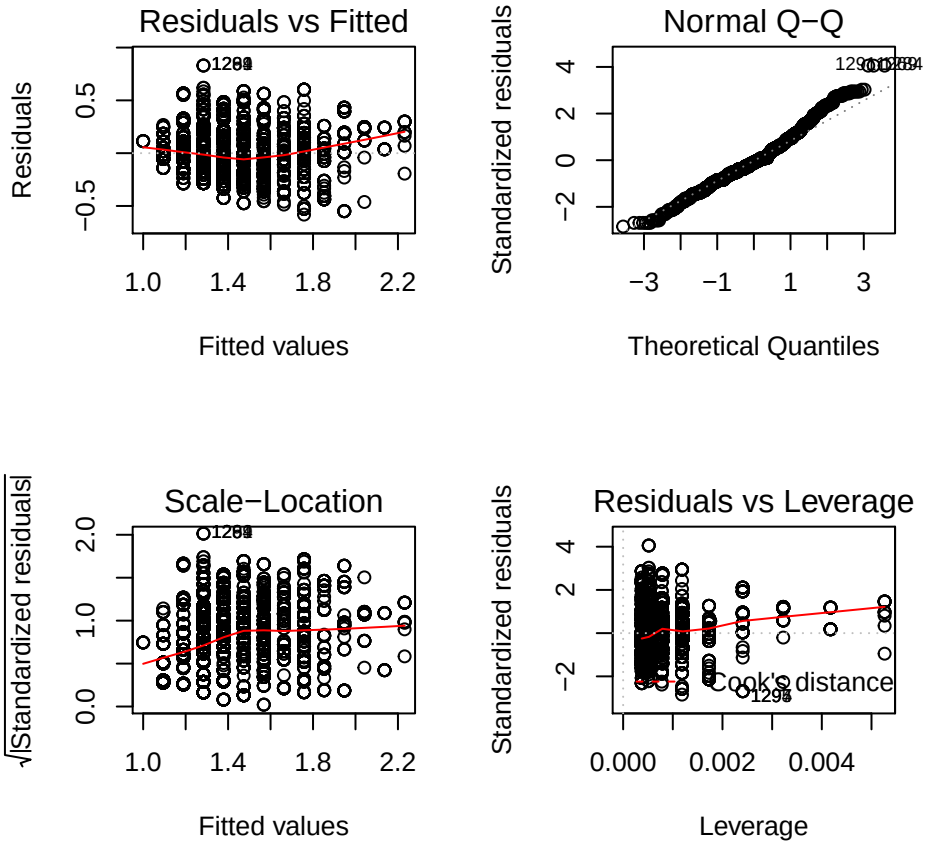


2.6 Questions

- Based on the QQ-plots, is the normal errors assumption believable?

Note that you could get also get these plots by using the plot function on the lm object.

```
par(mfrow = c(2,2))
plot(reg.model)
```



3 Multiple Testing and Reproducibility

In class we discussed a major issue with p-values. In particular, if we test enough things, we will stumble upon a significant p-value, even if the null hypothesis is actually true. This has lead to the claim that roughly half of all published scientific findings are actually untrue[1]. The vast majority of scientists are well meaning and not trying to game the system, but even then, this is still a large problem.

In this section, we will see how these false discovery and Type I errors that might happen. As an illustration, we will be examining the relationship between the daily temperature in various US cities and the performance of the Japanese stock market, as measured by the Nikkei 225 Index. The Nikkei 225 is roughly a weighted average of the stock prices of 225 large companies in Japan thought to be representative of the larger economy. In particular, we will be estimating models of the form

$$\text{Nikkei}_i = a + b \times \text{Temp}_i + \epsilon_i$$

and testing

$$H_0 : b = 0$$

$$H_A : b \neq 0$$

[1] http://www.theatlantic.com/magazine/archive/2010/11/lies-damned-lies-and-medical-science/308269/

We will repeat this process for many different US cities to see if there is any city, who's weather has a significant relationship with the performance of the Nikkei 225. We will use the R package `weatherData` to download weather data from Weather Underground.

```
# Not you only need to run the install.packages command once
# After you've installed the package, all you need to do is run the
# library command in the future
install.packages('weatherData')
library(weatherData)
```

We can use the `getWeatherForDate` command to pull the weather recorded for a specific city. In particular, we specify the 3 letter airport code. Below, we pull the weather data for Sea-Tac airport from Jan 1, 2016 until Mar 31, 2016. We can see that the resulting data includes the date, the min, the max and the mean temperature for each day. We will be working specifically with the mean temperature for each day. In order to match up the data with the financial data we will be using later, we will also need to slightly modify the Date field in our data.

```
# Pull the data for Seattle from weather underground
weather.data <- getWeatherForDate("SEA",
                                  start_date = "2016-01-01",
                                  end_date = "2016-03-31")
```

```
##  [1] "PDT"                       "Max_TemperatureF"
##  [3] "Mean_TemperatureF"         "Min_TemperatureF"
##  [5] "Max_Dew_PointF"            "MeanDew_PointF"
##  [7] "Min_DewpointF"             "Max_Humidity"
##  [9] "Mean_Humidity"             "Min_Humidity"
## [11] "Max_Sea_Level_PressureIn"  "Mean_Sea_Level_PressureIn"
## [13] "Min_Sea_Level_PressureIn"  "Max_VisibilityMiles"
## [15] "Mean_VisibilityMiles"      "Min_VisibilityMiles"
## [17] "Max_Wind_SpeedMPH"         "Mean_Wind_SpeedMPH"
## [19] "Max_Gust_SpeedMPH"         "PrecipitationIn"
## [21] "CloudCover"                "Events"
## [23] "WindDirDegrees"
## [1] "Date"              "Max_TemperatureF"  "Mean_TemperatureF"
## [4] "Min_TemperatureF"
```

```
# convert dates into strings and remove PDT tag
weather.data$Date <- gsub(" PDT","", weather.data$Date, fixed = T)

# look at the data
head(weather.data)
```

```
##         Date Max_TemperatureF Mean_TemperatureF Min_TemperatureF
## 1 2016-01-01               46                37               28
## 2 2016-01-02               42                34               25
## 3 2016-01-03               40                36               31
## 4 2016-01-04               38                37               35
## 5 2016-01-05               46                41               36
## 6 2016-01-06               53                45               37
```

To get financial data, we will use the `quantmod` package. There are many useful graphing and data commands for financial data in this package.

```
# Not you only need to run the install.packages command once
# After you've installed the package, all you need to do is run the
# library command in the future
install.packages('quantmod')
```

```
library(quantmod)
```

We can grab all data for the Nikkei 225 Index from Yahoo finance with the following lines of code. In particular, we will be looking at the returns from the beginning of 2016 until the end of March 2016

```
# Open Connection to Yahoo Finance
getSymbols("^N225",src="yahoo") # from yahoo finance

# Get % increase/decrease for each trading day for 2016
nikkei <- as.data.frame(OpCl(N225['2016-01-01::2016-03-31']))
```

When estimating our model, we need to be a little careful because the Japanese stock market only trades on weekdays, so we need to match up the correct days. In particular, the days for our Nikkei financial data are in the rownames of the **nikkei** object. So to match up the correct days, we need to take a subset of the weather data.

```
# tells us whether or not the date from the weather data is also included
# in the nikkei data
trading.days <- weather.data$Date %in% rownames(nikkei)

# mean temperature of days which match with nikkei trading days
temperature <- weather.data$Mean_TemperatureF[trading.days]
```

We can now fit a regression for the relationship between the daily temperature in Seattle, and the return of the Nikkei 225 Index.

```
# Note that the nikkei object is stored as a data frame so we can easily access
# the dates, but we want the first column for the regression
reg.model <- lm(nikkei[,1] ~temperature)
```

We can view the results of the regression easily using the **summary** command. In particular, in the resulting table we can see the p-value when for testing the null hypothesis of $b = 0$ vs the alternative of $b \neq 0$. Unsurprisingly, the relationship between the temperature in Seattle and the performance of the Nikkei 225 is not significant with a resulting p-value of .489.

```
summary(reg.model)
```

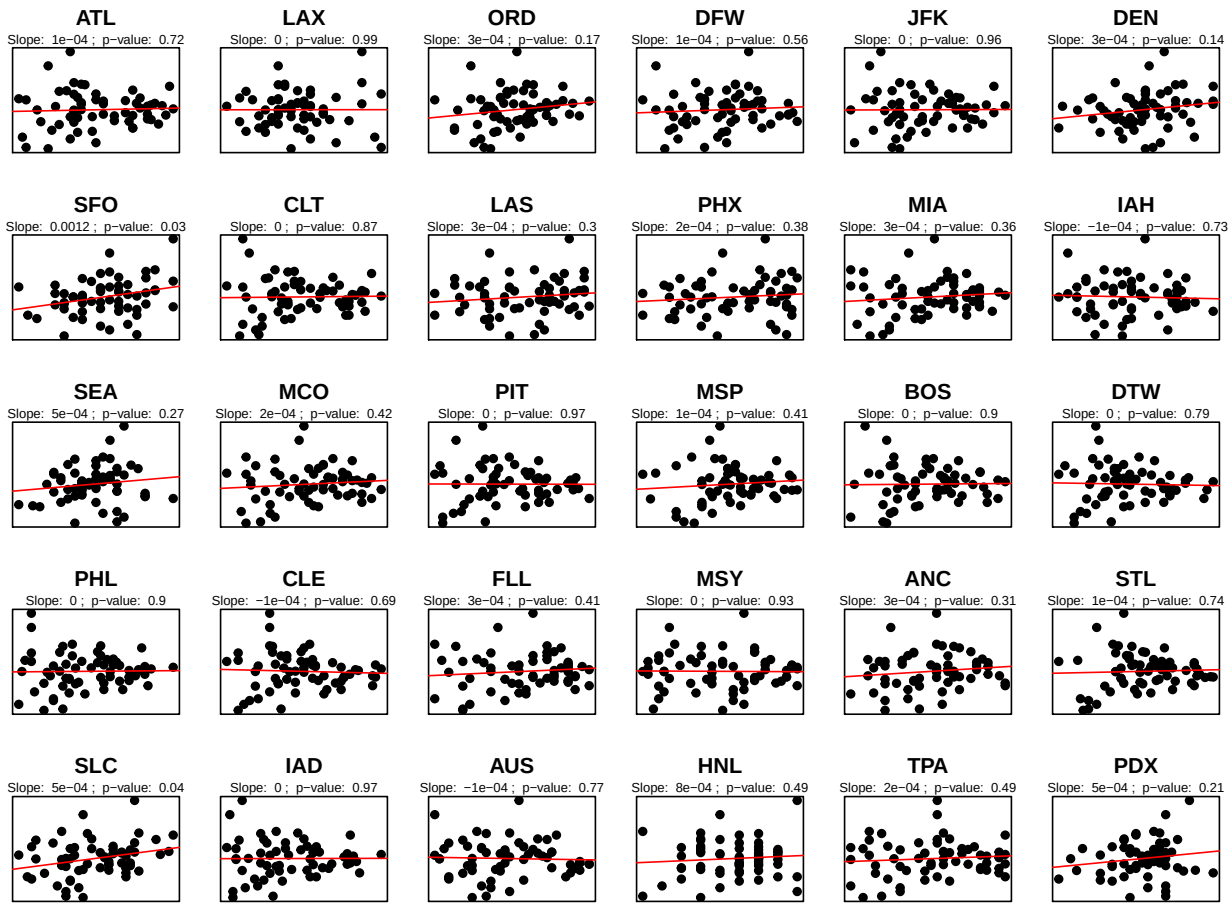
```
##
## Call:
## lm(formula = nikkei[, 1] ~ temperature)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.035499 -0.009257  0.001471  0.008662  0.049610
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0271761  0.0234901  -1.157    0.252
## temperature  0.0005450  0.0004905   1.111    0.271
##
## Residual standard error: 0.01615 on 59 degrees of freedom
## Multiple R-squared:  0.0205,Adjusted R-squared:  0.003894
## F-statistic: 1.235 on 1 and 59 DF,  p-value: 0.271
```

However, suppose that I think that some cities may have a significant impact, while others may not. So I don't just want to test the effect of Seattle, but of many cities across the US. In particular, I will try this same procedure with the 30 cities listed below. One of them might have an effect right?

```
airport.codes <- c("ATL",       "LAX",  "ORD",  "DFW",
                   "JFK",       "DEN",  "SFO",  "CLT",  "LAS",
                   "PHX",       "MIA",  "IAH",  "SEA",  "MCO",
                   "PIT",       "MSP",  "BOS",  "DTW",  "PHL",
                   "CLE",       "FLL",  "MSY",  "ANC",  "STL",
                   "SLC",       "IAD",  "AUS",  "HNL",  "TPA",
                   "PDX")
```

We repeat the procedure we followed above with each of the 30 airports listed above. Let's see if any of these airports have a statistically significant relationship with the Nikkei.

```
p.values <- rep(0, length(airport.codes))

par(mfrow = c(5,6), mar = c(1, 1, 3, 1))
for(i in 1:length(airport.codes)) {
  weather.data <- getWeatherForDate(airport.codes[i],
                                    start_date = "2016-01-01",
                                    end_date = "2016-03-31")

  # convert dates into strings and remove PDT tag
  weather.data$Date <- gsub(" PDT","", weather.data$Date, fixed = T)
  # tells us whether or not the date from the weather data is also included
  # in the nikkei data
  trading.days <- weather.data$Date %in% rownames(nikkei)

  # mean temperature of days which match with nikkei trading days
  temperature <- weather.data$Mean_TemperatureF[trading.days]

  # fit model
  reg.model <- lm(nikkei[,1] ~temperature)

  # pull the p-value out and record it
  p.values[i] <- summary(reg.model)$coefficients[2,4]
  plot(temperature, nikkei[,1], pch = 19, main = airport.codes[i], xaxt = "n", yaxt ="n")
  abline(reg.model$coeff, col = "red")
  mtext(paste("Slope: ", round(reg.model$coefficients[2],4), "; ",
              "p-value: ", round(p.values[i], 2)), cex = .5)

}
```



Looking at the p-values from each of the 30 different cities, we see that there is a wide range of p-values. However, two cities, San Francisco and Salt Lake City, have an estimated relationship which is significant at the .05 level. That's great news! We can now take these relationships we've discovered to trade stocks and tons of money. In retrospect, it's also pretty obvious that San Francisco's temperature would have an effect on the Japanese stock market. As the center of the tech universe, when the temperature rises, programmers are happier and more productive and stocks rise in response. Salt Lake City is a little harder to explain, but if you try hard enough, you can probably come up with an explanation. Before you drop out of UW and buy a yacht though, let's take a step back.

If there is a true relationship between the temperature in a city and the Nikkei 225, we would expect the relationship to be significant no matter what time period we look at. However, if there is no true relationship, we would expect the significance to disappear when we look at other time period. Let's check to see if the temperature vs Nikkei relationship holds for the following 3 months of 2016. First, let's check San Francisco.

```
nikkei.post <- as.data.frame(OpCl(N225['2016-04-01::2016-07-31']))

 weather.data <- getWeatherForDate("SFO",
                                   start_date = "2016-04-01",
                                   end_date = "2016-07-31")

## Retrieving from:  http://www.wunderground.com/history/airport/SFO/2016/4/1/CustomHistory.html?dayend

## The following columns are available:

##  [1] "PDT"                       "Max_TemperatureF"
```

```
##  [3] "Mean_TemperatureF"          "Min_TemperatureF"
##  [5] "Max_Dew_PointF"             "MeanDew_PointF"
##  [7] "Min_DewpointF"              "Max_Humidity"
##  [9] "Mean_Humidity"              "Min_Humidity"
## [11] "Max_Sea_Level_PressureIn"   "Mean_Sea_Level_PressureIn"
## [13] "Min_Sea_Level_PressureIn"   "Max_VisibilityMiles"
## [15] "Mean_VisibilityMiles"       "Min_VisibilityMiles"
## [17] "Max_Wind_SpeedMPH"          "Mean_Wind_SpeedMPH"
## [19] "Max_Gust_SpeedMPH"          "PrecipitationIn"
## [21] "CloudCover"                 "Events"
## [23] "WindDirDegrees"
```

```
## Checking Summarized Data Availability For SFO
```

```
## Found 122 records for 2016-04-01 to 2016-07-31
```

```
## Data is Available for the interval.
```

```
## Will be fetching these Columns:
```

```
## [1] "Date"              "Max_TemperatureF"  "Mean_TemperatureF"
## [4] "Min_TemperatureF"
```

```
  # convert dates into strings and remove PDT tag
  weather.data$Date <- gsub(" PDT","", weather.data$Date, fixed = T)
  # tells us whether or not the date from the weather data is also included
  # in the nikkei data
  trading.days <- weather.data$Date %in% rownames(nikkei.post)

  # mean temperature of days which match with nikkei trading days
  temperature <- weather.data$Mean_TemperatureF[trading.days]

  # fit model
  reg.model <- lm(nikkei.post[,1] ~temperature)
  summary(reg.model)
```

```
##
## Call:
## lm(formula = nikkei.post[, 1] ~ temperature)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.083570 -0.006070  0.000991  0.009506  0.022739
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.916e-03  3.024e-02  -0.096    0.923
## temperature  2.946e-05  4.868e-04   0.061    0.952
##
## Residual standard error: 0.01539 on 79 degrees of freedom
## Multiple R-squared:  4.636e-05,Adjusted R-squared:  -0.01261
## F-statistic: 0.003662 on 1 and 79 DF,  p-value: 0.9519
```

Looks like the relationship disappears. Now, let's check Salt Lake City.

```
nikkei.post <- as.data.frame(OpCl(N225['2016-04-01::2016-07-31']))

 weather.data <- getWeatherForDate("SLC",
```

```
                                  start_date = "2016-04-01",
                                  end_date = "2016-07-31")
```

```
## Retrieving from:  http://www.wunderground.com/history/airport/SLC/2016/4/1/CustomHistory.html?dayend
```

```
## The following columns are available:
```

```
##  [1] "MDT"                       "Max_TemperatureF"
##  [3] "Mean_TemperatureF"         "Min_TemperatureF"
##  [5] "Max_Dew_PointF"            "MeanDew_PointF"
##  [7] "Min_DewpointF"             "Max_Humidity"
##  [9] "Mean_Humidity"             "Min_Humidity"
## [11] "Max_Sea_Level_PressureIn"  "Mean_Sea_Level_PressureIn"
## [13] "Min_Sea_Level_PressureIn"  "Max_VisibilityMiles"
## [15] "Mean_VisibilityMiles"      "Min_VisibilityMiles"
## [17] "Max_Wind_SpeedMPH"         "Mean_Wind_SpeedMPH"
## [19] "Max_Gust_SpeedMPH"         "PrecipitationIn"
## [21] "CloudCover"                "Events"
## [23] "WindDirDegrees"
```

```
## Checking Summarized Data Availability For SLC
```

```
## Found 122 records for 2016-04-01 to 2016-07-31
```

```
## Data is Available for the interval.
```

```
## Will be fetching these Columns:
```

```
## [1] "Date"              "Max_TemperatureF"  "Mean_TemperatureF"
## [4] "Min_TemperatureF"
```

```
  # convert dates into strings and remove PDT tag
  weather.data$Date <- gsub(" PDT","", weather.data$Date, fixed = T)
  # tells us whether or not the date from the weather data is also included
  # in the nikkei data
  trading.days <- weather.data$Date %in% rownames(nikkei.post)

  # mean temperature of days which match with nikkei trading days
  temperature <- weather.data$Mean_TemperatureF[trading.days]

  # fit model
  reg.model <- lm(nikkei.post[,1] ~temperature)
  summary(reg.model)
```

```
##
## Call:
## lm(formula = nikkei.post[, 1] ~ temperature)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.084642 -0.006400  0.001654  0.008922  0.023483
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0093384  0.0093248  -1.001    0.320
## temperature  0.0001173  0.0001303   0.900    0.371
##
## Residual standard error: 0.01532 on 79 degrees of freedom
```

```
## Multiple R-squared:  0.01015,Adjusted R-squared:  -0.002384
## F-statistic: 0.8097 on 1 and 79 DF,  p-value: 0.3709
```

Looks like this relationship disappears as well, so you should probably stick to school and finish your degree after all. Although this example is somewhat contrived, and we probably had a lot of prior knowledge that made us skeptical about a relationship between temperature of US cities and the stock market in Japan, however, you can see how it might have been easy to blindly follow the data and rationalize a relationship between San Francisco's temperature and the Japanese stock market post-hoc.

Researchers often try many experiments without success and need to test out many different experimental conditions. For example, in the search for genetic causes of cancer this can be a real problem. There are roughly 20,000 genes in humans, so when we test to see if any of those are related to a specific type of cancer, we are bound to get many Type I errors (false positive). This can be an especially large problem if you are doing research in an area where there is not as much prior knowledge available or have a pre-existing agenda to prove.

In this case, we had split our available data into two groups so we could see if the same relationships held over two time periods. However, often times experiments can be expensive and difficult to run, so replicating results are often not checked. Read about one project to check reproducibility `http://www.theatlantic.com/science/archive/2015/08/psychology-studies-reliability-reproducability-nosek/402466/`.