Stat 021 Homework 2

Suzanne Thornton

Due: Fri, Sept 20, 12:00pm

Instructions: A **pdf** version of your homework must be submitted to Gradescope by **noon** on the due date. The course passcode is **MPKJ4Z**. If you are having trouble getting your *.Rmd* file to compile, you need to get help with this **before** the due date.

You are allowed to hand in **only one** late homework assignment throughout the semester. If you need to hand in this particular assignment late, you must let me know via email by noon on the due date.

You are encouraged to study with your peers to help complete the homework assignments but no copying is allowed. If I see that two or more homework assignments are copied, all students involved will receive a grade of 0 on that assignment and will forfeit (perhaps retroactively) the opportunity to hand in a late homework.

Q 1) Create a non-normal data set of sample size $n = 20$ by setting a random seed and drawing your observations from a Beta distribution with $\alpha = 5$ and $\beta = 1.23$.(5 points)

a) Create a tibble (with one column) of your simulated data. Print the first few rows of this tibble using the *head()* function.
b) Test if this data looks like it comes from a Normally distributed population using a Normal probability plot. Comment on the behavior of the data and what the deviations from Normality look like.
c) Create 24 other data sets (tibbles) each of sample size $n = 20$, and each also drawn from a Beta distribution with $\alpha = 5$ and $\beta = 1.23$. For each of these 25 tibbles (each only having one column of data), compute the mean and store these 25 means in a tibble called *beta_means*. Print the first few rows of this tibble, *beta_means*.
d) Now test if the data in *beta_means* looks like it comes from a Normally distributed population using a Normal probability plot as in part (b). Does this data look like it's Normally distributed? Why or why not?

Hint: Make sure you set the random number seed in R so that your results are reproducible. (The command for that is *set.seed(100)* although you can use any number you want, not just 100. Look at the R documentation on the R function *rbeta()*.)

Solution:

Part a)

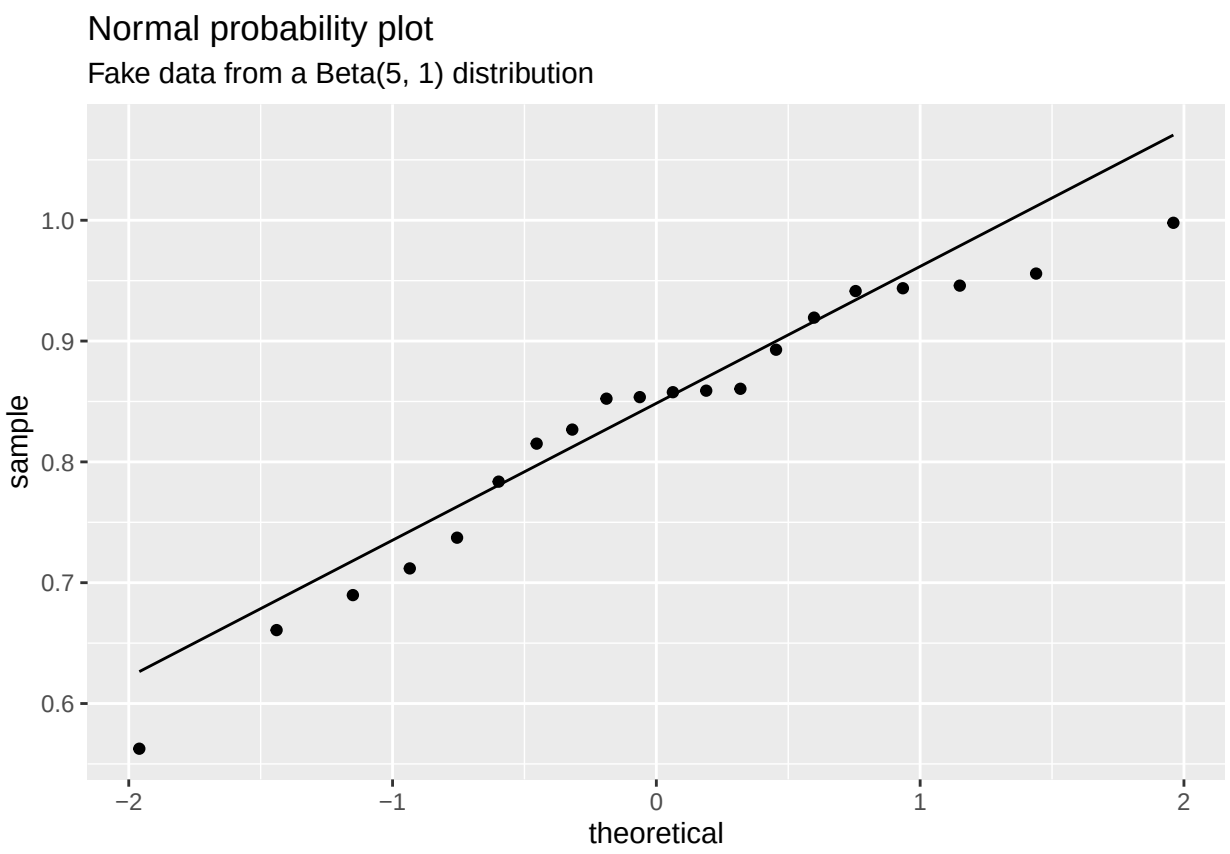
```
set.seed(100)
orig_beta_samp <- tibble(observations=rbeta(20, 5, 1))
head(orig_beta_samp)
```

```
## # A tibble: 6 x 1
##   observations
##          <dbl>
## 1        0.690
## 2        0.861
## 3        0.815
## 4        0.956
## 5        0.858
## 6        0.893
```

Part b) Since we are plotting the quantiles of a sample drawn from a Beta distribution we would not expect to see points that align well with the QQ-line. Both the minimum and maximum values of our sample are

smaller than we'd expect to see them be if the population was Normally distributed. In fact, most of the sample quantiles lie *below* the QQ-line indicating that our sample comes from a population that is less dispersed than a Normal population (so we have lighter-than-Gaussian tails on both sides of the population).

```
plot1 <- ggplot(data = orig_beta_samp) +
         labs(title = "Normal probability plot",
              subtitle = "Fake data from a Beta(5, 1) distribution")
plot1 + aes(sample=observations) + stat_qq() + stat_qq_line()
```



Part c)

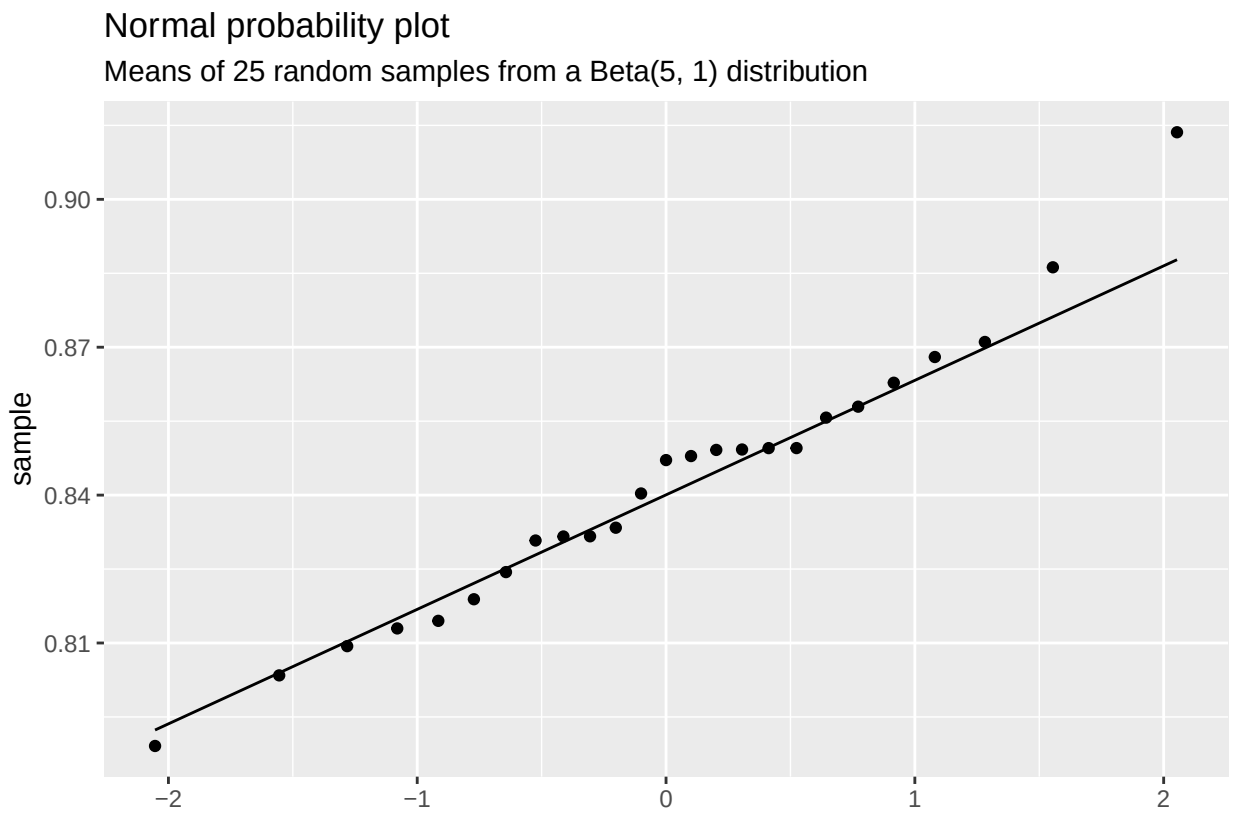
```
new_beta_samps = matrix(rbeta(20*24, 5, 1), ncol=24, byrow=TRUE)
all_beta_samps <- as_tibble(cbind(orig_beta_samp, new_beta_samps))

beta_means_tibble <- tibble(beta_means = all_beta_samps %>% colMeans)
head(beta_means_tibble)
```

```
## # A tibble: 6 x 1
##   beta_means
##        <dbl>
## 1      0.833
## 2      0.848
## 3      0.850
## 4      0.831
## 5      0.814
## 6      0.868
```

Part d)

```
plot2 <- ggplot(data = beta_means_tibble) +
         labs(title = "Normal probability plot",
              subtitle = "Means of 25 random samples from a Beta(5, 1) distribution",
              x = " ")
plot2 + aes(sample=beta_means) + stat_qq() + stat_qq_line()
```



When comparing this plot to the plot from part (b), we see more points fitting closer to the diagonal line now than we did before. The main thing to note here is that we have illustrated the **central limit theorem** and shown that the *mean* of non-Normal data can itself be normally distributed.

Q 2) Read the following article: https://fivethirtyeight.com/features/science-isnt-broken and watch this John Oliver clip: https://youtu.be/0Rnq1NpHdmw (some adult language content). (5 points)

a) Briefly define p-hacking and researcher degrees of freedom.

b) Why should you not believe a finding from any single scientific study? When should you consider a finding to be reliable?

c) What are replication studies? Why are they rare?

Solution:

Part a) For full credit your answer must include an explanation of p-hacking that highlights that p-values cannot tell you if your hypothesis is right or wrong. Instead, they give you a sense of how rare (or surprising) your data are. For an explanation of researcher degrees of freedom, you need to mention that these are all the little (and big) decisions that scientists make before, during, and after their experiment (you could also perhaps reference the connection the the twitter comic that I shared in class the other day by Kareem Carr.)

Part b) For full credit you need to mention something about how there are so many different ways to analyse even one data set and all of these distinct methods can differ still depending on the biases (latent or overt) of

the individual(s) doing the analysis. Not to mention, we are continuously updating old data with new data. You can trust results if there have been many related studies and most of them found strong evidence of the same type of relationship.

Part c) Results that aren't "statistically significant" aren't generally publishable and it's harder to get funding for research unless you're getting "new" and "different" findings.

Q 3) The data file "vapor_pressure.txt" contains the vapor pressure (in mm Hg) of water for various temperatures (in deg K). Look at the data file before you try to do anything with it! There is an extra line of information between the variable names and the data itself. You'll need to figure out how to make sure that R does not include this extra line as part of your data.

a) Read the data into R and produce a scatter plot of vapor pressure by temperature. Does it seem likely that a straight-line model will be adequate to describe what you see? (4 pts)

b) Fit a simple linear regression model with *vapor_pressure* as the predictor variable and *temperature* as the response variable. Plot the regression line over a scatter plot and identify the intercept and the slope of the model. What are your conclusions regarding model adequacy? (2 pts)

c) From physical chemistry the Clausius-Clapeyron equation states that, for p_v = vapor pressure and T = temperature,

$$ln(p_v) \propto -1/T.$$

Create a data set that contains the *vapor_pressure* data but has an additional column corresponding to the natural logarithm of the values of *vapor_pressure*. Now repeat part b using temperature and this new variable. Are your conclusions regarding model adequacy different from those in part (b)? Why or why not? (4 pts)

Hint: The natural logarithm function in base R is simply *log()*.

Solution:

Part a) In the scatter plot below, it is clear that there is some non-linear relationship between the predictor and response variables, the relationship looks more curved than linear so it doesn't look like a linear model will be a good fit here.

```
vp_data <- read_table2("vapor_pressure.txt", skip = 2, col_names = FALSE)
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   X2 = col_double()
## )
```

```
head(vp_data)
```

```
## # A tibble: 6 x 2
##      X1    X2
##   <dbl> <dbl>
## 1   273   4.6
## 2   283   9.2
## 3   293  17.5
## 4   303  31.8
## 5   313  55.3
## 6   323  92.5
```

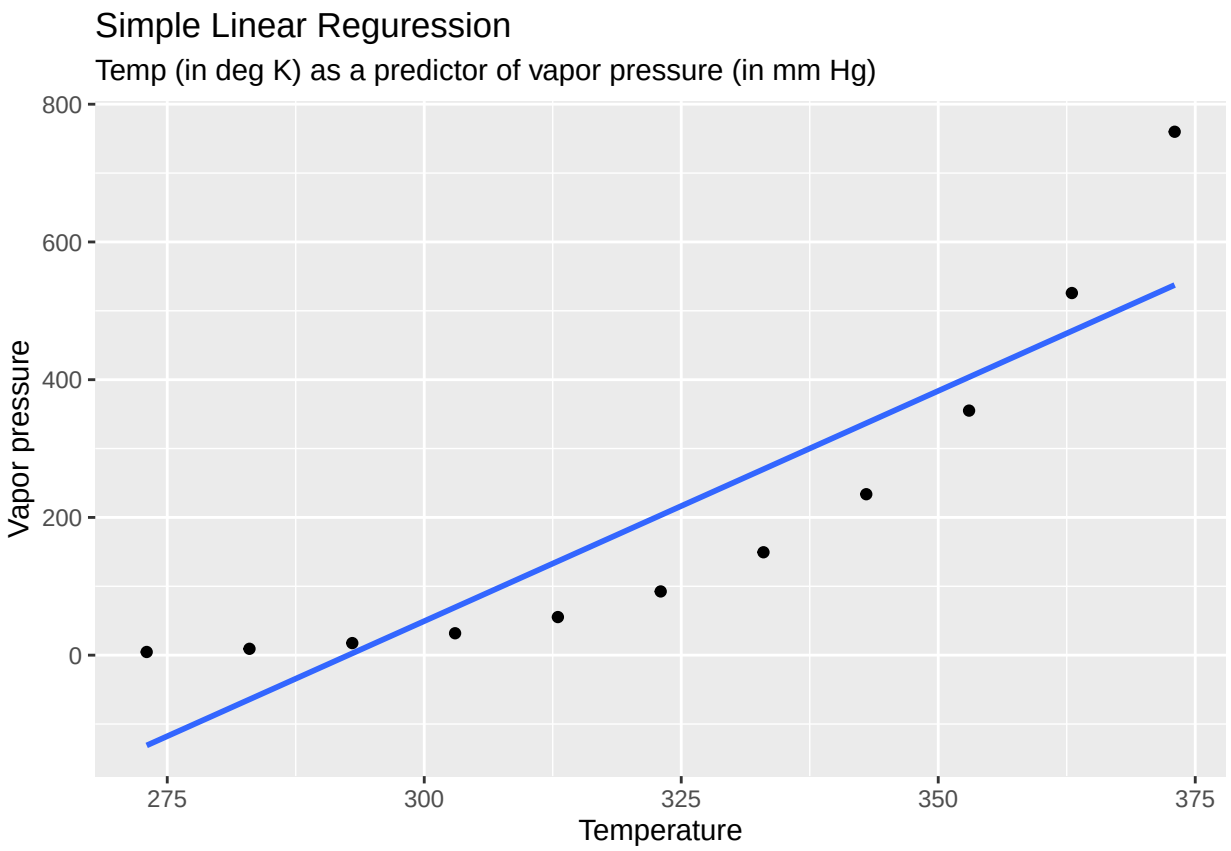
```
#Note: the columns don't have names so to keep track of what our data measures, we will create names fo
colnames(vp_data) = c("temperature", "vapor_pressure")
head(vp_data)
```

```
## # A tibble: 6 x 2
##   temperature vapor_pressure
##         <dbl>          <dbl>
## 1         273            4.6
## 2         283            9.2
## 3         293           17.5
## 4         303           31.8
## 5         313           55.3
## 6         323           92.5
```

Part b)

```
SLR_vp_data <- lm(vapor_pressure~temperature, data=vp_data)

ggplot(vp_data, aes(x=temperature, y=vapor_pressure)) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE) +
  labs(title="Simple Linear Reguression", subtitle="Temp (in deg K) as a predictor of vapor pressure (i
       x="Temperature", y="Vapor pressure")
```



The intercept of this regression line is -1956.2584545 and the slope is 6.6855455. This doesn't seem to be a good enough model for the data however, since we can tell that there is more information about the relationship that we are not capturing with a line.

Part c)

A linear model fits the transformed data much better. This is because (as mentioned in the problem), there is a well-known non-linear relationship between temperature and vapor pressure. The $\propto$ symbol just means that the terms on the LHS and RHS differ only by a constant multiplied by either of the terms. So the

natural logarithm of the vapor pressure is linearly related to the inverse of the temperature (in deg K). For full credit you must note that there is a known non-linear relationship between the two variables but we can use a *linear* regression model to describe the relationship between some *transformations* of the variables.

```
new_vp_data <- vp_data %>% mutate(ln_vp = log(vapor_pressure))

SLR_new_vp_data <- lm(ln_vp~temperature, data=new_vp_data)
ggplot(new_vp_data, aes(x=temperature, y=ln_vp)) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE) +
  labs(title="Simple Linear Reguression", subtitle="Temp (in deg K) as a predictor of vapor pressure (in
       x="Temperature", y="ln(Vapor pressure)")
```