PSTAT 126 Final Solutions

Part I Solutions

Problem 1: Obtaining data samples for a multiple regression problem from the dataset "PSTAT 126 Final Dataset 1" (which will be sent in an email with that subject line), utilize any or all of the various regression model-fit assessment tools available in R that we have discussed in the course to help identify and validate a linear regression model that appears to best fit the given data. These will for example likely include, but may not be limited to, visual plots and/or graphs (for example, residual vs. fitted value plots), results of statistical hypothesis tests, numerical regression model accuracy measures, and linear model summary output reports, as well as any other applicable diagnostic tools that we have considered in the course. You should use the solutions to Problems 1 and 5 in Homework #3 as a general guide as to what types of information your answer could or should contain, but there may have been course material introduced after Homework #3 was assigned that may also be relevant here. You will likely need to try a number of different variable transformations on the response variable and/or predictor variables, using the diagnostic tools to decide which of the resulting regression models appear consistent with the data and which do not. You should **clearly** identify one or perhaps two candidate regression models that you believe are most consistent with the given data. Please try to include screenshots of the graphical plots you use as well as quote any relevant R code output results. You can or even should include the R code itself as well if you feel it helps to support your argument. Please explain your reasoning as to why the model(s) you propose may be the right one(s) in plain, natural language; you do not necessarily have to identify the "right" model to get a great deal of partial credit or even perhaps full credit.

Solution: The R code producing the data in the dataset for the problem is the following:

```
x_val1 <- seq(5,104)
x_val2 <- seq(15,114)
e6 <- rnorm(100, mean = 0,  sd = 1.5)
y_val <- 20+19*x_val1+39*log(x_val2)+e6
```

Hence, the model according to which the data was generated is:

$$Y_n = 20 + 19x_{1n} + 39\log(x_{2n}) + \epsilon_n,\ n = 1, \ldots, N,$$
$$\epsilon_n \sim N(0, 2.25), n = 1, \ldots, N,$$

with N=100. Of course those taking the final obviously did not themselves have access to the above specifications of the model generating the data, so they would need to apply the various diagnostic tools and tests to identify a regression model consistent with the data. We show here in the problem solution how these tools and tests perform against the true model itself to demonstrate that it should indeed be possible to use them to identify the "true", underlying model or in any case a reasonably suitable model.

The R lm( ) function summary output report for the correct model as above is:

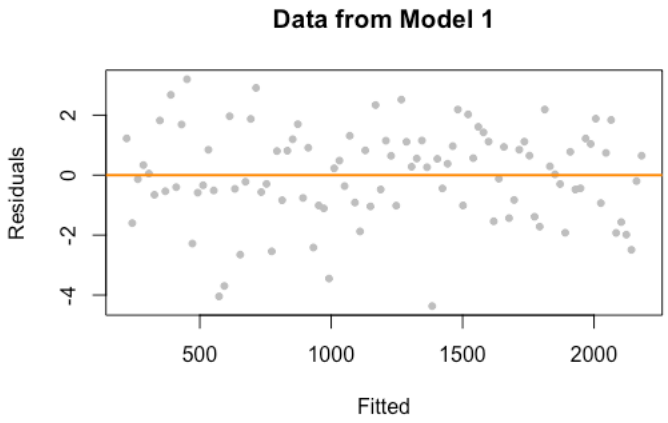
```
Call:
lm(formula = y_val ~ x_val1 + log(x_val2), data = Data2020Dec16),
```

(where x_val1 corresponds to the x_{1n} and x_val2 to the x_{2n})

```
Residuals:
   Min     1Q  Median     3Q    Max
-4.3677 -0.9193  0.1400  1.1144  3.1974

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 25.22991    3.61272   6.984 3.61e-10 ***
x_val1      19.02931    0.02202 864.251  < 2e-16 ***
log(x_val2) 37.32225    1.17827  31.675  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.562 on 97 degrees of freedom
Multiple R-squared:      1,      Adjusted R-squared:      1
F-statistic: 6.635e+06 on 2 and 97 DF,  p-value: < 2.2e-16
```

Note in particular the high value for the Multiple R-squared/Adjusted R-squared. The regression is significant according to the p-values for the t-tests and F-test.

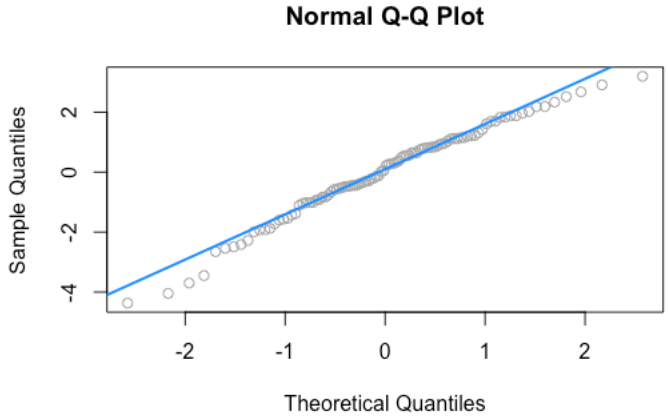
We have the corresponding Residuals vs. Fitted Values plot:



The residuals vs. fitted plot looks pretty much as it should, and the values generated by our formal diagnostic hypothesis tests also look good:

Breusch-Pagan Test for homoscedasticity: p-value = 0.3514 (this value supports that homoscedasticity is satisfied)
Shapiro-Wilk Normality Test: p-value = 0.17 (this value is consistent with normality being satisfied)
Goldfeld-Quandt Test for homoscedasticity: 0.9006 (this value supports that homoscedasticity is satisfied)

Durbin-Watson Test for autocorrelation: p-value= 0.55 (this value supports no significant autocorrelation)



The Q-Q plot, with its straight line appearance, does suggest normality, as desired.

We conclude that our diagnostic tests and tools would indeed help us greatly to identify the true model underlying the data, as the true model as we have just seen does perform well against them.

Finally, note that the following single-predictor model is, due to the nature of the given dataset, essentially equivalent, of course, to the two-predictor model as above:

$$Y_n = 20 + 19x_{1n} + 39\log(x_{1n} + 10) + \epsilon_n,\ n = 1, \dots, N,$$
$$\epsilon_n \sim N(0, 2.25), n = 1, \dots, N,$$

with N=100.

Problem 2:In this problem work within the Simple Linear Regression (SLR) context:

$$Y_n = \beta_0 + \beta_1 x_n + \epsilon_n,\ n = 1, \dots, N,$$
$$\epsilon_n \sim N(0, \sigma^2), n = 1, \dots, N.$$

(a) Show that that $\mathrm{E}[\hat{\beta}_1] = \beta_1$, where E[ ] denotes expectation. Please do not simply quote a theorem statement for this part of this problem or those below, but instead give a mathematical argument. (Also note that showing $\mathrm{E}[\hat{\beta}_m] = \beta_m$ for $m = 0$ is similar to the case $m = 1$, and you only need to include in your answer the case for $m = 1$.)

(b) Show that Var($\hat{\beta}_1$)= $\frac{\sigma^2}{S_{xx}}$, where $S_{xx} = \sum_{n=1}^{N}(x_n - \bar{x})^2$ and Var( ) denotes the variance.

(c) Show that Var($\hat{\beta}_0$)= $\sigma^2\left(\frac{1}{N} + \frac{\bar{x}^2}{S_{xx}}\right)$, where $\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$.

(d) Are we able to conclude, directly from parts (a) and (b), that $\hat{\beta}_1$ is normally-distributed with mean β_1 and variance $\frac{\sigma^2}{S_{xx}}$? Why or why not?

Solution: For (a)-(c), see the Appendix, Sec. A.4, of the Weisberg (2014) book Applied Linear Algebra (4^{th} ed., see the reference in the syllabus).
(d) No, the non-trivial result that the finite sum of independent, normally-distributed random variables is also normally-distributed (or some similar result) is also needed.

Problem 3: Agents at a call center get a score of 1 if a caller was satisfied with a particular call and a score of 0 if not. The company wants to see if it can accurately predict, including generating a probability estimate for the prediction, whether customers will be satisfied with a call based on relevant predictors involved such as, for example, length of the call, number of months of experience of the agent, time of day that the caller calls, etc. What is a natural regression method to use to build such a predictive model? Please first describe in detail (without any R or other software code) how you would algorithmically/mathematically set up a regression-based model to solve this problem, including how you could generate probability value estimates. You can assume there are M predictor variables. Then describe how you could set up and numerically solve such a problem in practice using R. For this part, do include the R code. You can use the built-in mtcars dataset, which does include 0/1-valued variables, as a stand-in dataset for this part of the problem. In your proposed model, use the "vs" variable -- a 0/1-valued variable -- as the response variable to serve as a stand-in for the score of a call-center call. Taking M=2, you should use "wt" and "disp" as the stand-in predictor variables. What estimates for the intercept and the coefficients of wt and disp do you get? What probability values for 0 and 1 do you get from this model when wt = 2.8, disp = 160?

Solution: An appropriate regression method to use to address the modeling of the call center problem in the problem is a Generalized Linear Model – specifically Logistic Regression. Let

$$p(\mathbf{x})=P[Y=1|\mathbf{X}=\mathbf{x}].$$

Here, **X** is a random vector and **x** a (sample) vector, with Y a 0/1 binary random variable. With a binary (Bernoulli) response random variable, we can focus on addressing the case Y=1, since we then will have

$$P[Y=0|\mathbf{X}=\mathbf{x}]=1-p(\mathbf{x}).$$

We now define the associated **Logistic Regression Model:**

$$\log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right)=\beta_0+\beta_1 x_1+\ldots+\beta_M x_{M,}$$

where $\mathbf{x}=(x_1,\ldots,x_M)$.

Note that applying the inverse logit transformation allows us to obtain an expression for the probability p(**x**):

$$p(\mathbf{x})=P[Y=1|\mathbf{X}=\mathbf{x}]=\exp(\beta_0+\beta_1 x_1+\cdots+\beta_M x_M)/(1+\exp(\beta_0+\beta_1 x_1+\cdots+\beta_M x_M)).$$

With N observations, we write the model indexed with a second variable to note that it is being applied to each observation

$$\log(p(\mathbf{x}_i)/(1-p(\mathbf{x}_i)))=\beta_0+\beta_1 x_{i1}+\ldots+\beta_M x_{iM,\ i=1,\ldots,N.}$$

The relevant, corresponding R code for solving this logistic regression model is

```
fit_glm102 = glm(vs ~ wt + disp, data = mtcars, family = binomial)
summary(fit_glm102)
```

We get the following estimates for the intercept and the coefficients of wt and disp:

	Estimate
(Intercept)	1.60859
wt	1.62635
disp	-0.03443

The code fragment to generate probabilities of 0 and 1 for the given data sample (note of course that if one is p the other is 1-p) is:

```
newdata = data.frame(wt = 2.8, disp = 160)
predict(fit_glm102, newdata, type="response")
```

The answer should include the corresponding specific, numerical probability values:

0.658 and 0.342

for the model as specified in the problem.

Problem 4: Obtain data samples from the dataset "PSTAT 126 Final Dataset 2" (which will be sent in an email with that subject line) with a single predictor variable. Then follow the same instructions as for Problem 1 above of this final.

Solution:
The R code generating the data was the following:

```
x <- 10:100
eps <- rnorm(length(x), sd = 2)
y <- (x+10+eps) ^ (-1/1.85)
```

That is, the model according to which the data was generated is:

$$Y_n^{-1.85}=10+x_n+\epsilon_n,\ n=1,\dots,N,$$
$$\epsilon_n \sim N(0,4), n=1,\dots,N,$$

with N=91.

Even though one could not know that this model is the one that generated the data, one could decide to try the Box-Cox method to obtain an optimal or approximately optimal power

transformation for the response variable, and test that against our formidable arsenal of regression diagnostic tests and tools. To do this, we first read in the data to R:

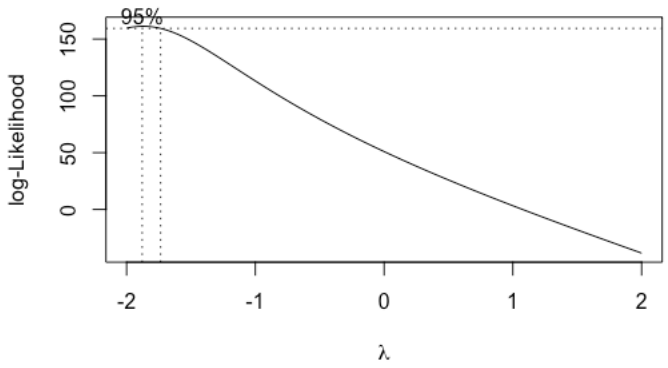
```
data17Dec2020SVMod=read.csv("/Users/danielzanger/2020_Dec_11_1508_Data_CSV.csv")
```

Then follow the following steps (not all of these steps may be absolutely necessary):

```
m17 <- lm(y ~ x, data = data17Dec2020SVMod)

library(MASS)
library(broom)
augmented_m17 <- augment(m17)

library(tidyverse)

bc17 <- boxcox(m17)

lambda17 <- bc17$x[which.max(bc17$y)]
lambda17
```

The Box-Cox technique computes the optimal power exponent according to its methodology (involving the log-likelihood) to be

lambda17 = -1.878788



We can then use the lm() function in R to solve the corresponding regression model for the corresponding $\hat{\beta}_m$ coefficients:

```
m317 = lm(I(y ^ lambda17) ~ x, data = data17Dec2020SVMod)
```

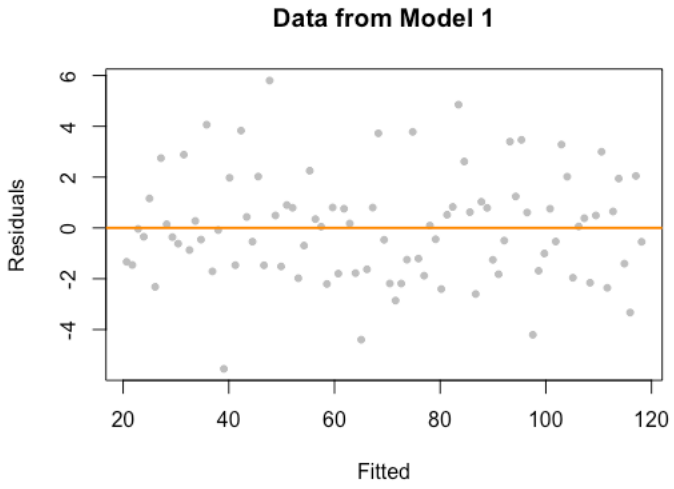
The summary report this produces is the following:

Call:

```
lm(formula = I(y^lambda17) ~ x, data = data17Dec2020SVMod)

Residuals:
   Min     1Q  Median     3Q    Max
-5.5456 -1.4950 -0.0390  0.8639  5.8007

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 9.825505   0.515709   19.05   <2e-16 ***
x           1.082998   0.008461  128.00   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.12 on 89 degrees of freedom
Multiple R-squared:  0.9946,  Adjusted R-squared:  0.9945
F-statistic: 1.638e+04 on 1 and 89 DF,  p-value: < 2.2e-16
```

In particular the R-squared is again high, and the (somewhat limited) residual information in the summary report looks pretty good. The regression is significant according to the p-value(s).

The corresponding Residuals vs. Fitted plot, probably even more important than the summary report at least in this case, also looks quite good:



This residuals vs. fitted plot is produced by:

```
plot(fitted(m317), resid(m317), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Data from Model 1")
abline(h = 0, col = "darkorange", lwd = 2)
```

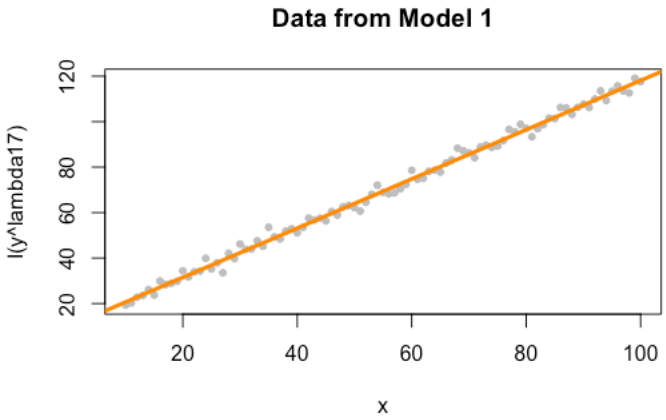
Our diagnostic hypothesis tests give the following favorable results:

Breusch-Pagan Test for homoscedasticity: p-value = 0.9998 (this value supports that homoscedasticity is satisfied)
Shapiro-Wilk Normality Test: p-value = 0.2678(this value implies normality)
Goldfeld-Quandt Test for homoscedasticity: 0.566 (this value implies that homoscedasticity is satisfied)
Durbin-Watson test for autocorrelation: p-value= 0.788 (this value implies no significant autocorrelation)

We can produce a plot of the transformed response against the predictor via the following code as well:

```
plot(I(y ^ lambda17) ~ x, data = data17Dec2020SVMod, col = "grey", pch = 20,
    main = "Data from Model 1")
fit_321 = lm(I(y ^ lambda17) ~ x, data = data17Dec2020SVMod)
abline(fit_321, col = "darkorange", lwd = 3)
```

This yields



This appears to suggest a good fit as well.

Part II Solutions

Question 1: What is a difference – or differences -- between the AIC and BIC measures used in stepwise regression? Why use one as opposed to the other?

Answer: The difference between the two measures (AIC vs. BIC, see course slide 72) is in the second term of both of them – the term which is intended to control model complexity. For the AIC this term is $2M$ and for the BIC it is $\log(N)M$, where M is (or is essentially) the number of regressors in the model and N is the number of data samples. So, as N grows larger the BIC will tend to choose a less complex model (as measured by M) than the AIC to compensate for the $\log(N)$ factor. Hence as the BIC emphasize a less complex model it will in general likely reduce the risk of model overfitting as N grows large.

Question 2: Is the F-test with its associated p-value, as reported in the R linear model summary output report, a measure of genuine model accuracy? Why or why not?

Answer: The F-test is not really a measure of model accuracy. The F-test tells you whether your predictor variables/regressors enable the regression model to be measurably more accurate (and more predictive) than a very simple intercept-only model. So, the F-test indicates whether your regression model is significant and whether the complexity, in some sense, of your model is adding much of any (explanatory) value at all.

Question 3: You have executed a regression with a number *N* of data samples, but you are concerned that the value $\hat{\beta}_m$ may not be a close enough estimate for the corresponding value of β_m. What is one simple step you can take in this context to obtain a $\hat{\beta}_m$ value that is likely closer to β_m?

Answer: Simply choose a larger value for the number *N* of data samples used in the regression procedure.

Question 4: Are the residuals in a simple linear regression problem in general fully independent (independent in the standard sense of probability theory)? Why or why not? When or when not?

Answer: The answer is that the residuals are not in general independent. The reason is that, as we showed earlier in the course,

$$\sum_{n=1}^{N} e_n = 0.$$

So for example we have

$$\sum_{n=2}^{N} e_n = -e_1,$$

which implies a lack of independence in general.

Question 5: Polynomials often tend to be promising functions to use in a regression to approximate the mean function (or indeed any function you may wish to approximate, for that matter). What is/are a general mathematical condition or conditions on a function that implies/imply that the function can be well-approximated by polynomials?

Answer: A general condition on a function implying that it can in principle be well-approximated by polynomials is that the function be continuous (this is the famous Weierstrass Approximation Theorem). If the function has greater levels of smoothness than its just being continuous -- that is, for example, at least some of its derivatives in the standard calculus sense exist – then there are mathematical results that tell you what degree of a polynomial is high enough to actually obtain a close approximation.

Question 6: You want to analyze the effect of different anti-viral medications ("anti-virals") on recovery from a virus-borne illness. Subjects testing positive for the disease are randomly assigned to take one of a number of these anti-virals. You wish to determine whether these medications all give rise to the same mean time to recovery from the viral illness or whether at least one of the drugs gives rise to a mean recovery time that is different from at least one of the others. What is a general statistical method or approach that at least in principle could be used to potentially accomplish this analysis? How would you set it up as a hypothesis test? If you were to be given the relevant dataset(s), explain (with reasonable detail and specificity) how you could perform such a test in R and produce a corresponding result. (Please note however that you are NOT actually being give the datasets here, so just explain what you would do without actually doing it.)

Solution: A general statistical method/approach that could be used to accomplish the analysis as described in the problem is One-way Analysis of Variance (One-way ANOVA). (Analysis of Variance (ANOVA) compares the variation due to specific sources (between groups) with the variation among individuals who should be similar (within groups).)

One can set up the analysis as a hypothesis test in the following manner:

H_0: $\mu_1=\mu_2=...=\mu_N$ vs. H_1: not all the μ_i are equal,

where N is the number of anti-virals being considered and μ_i is the mean time to recovery of anti-viral i.

To perform such a test in R you could use, given a suitable dataset, the following R code, in particular the aov( ) function (here the "coagulation" dataset is used as an example):

```
coag_aov = aov(coag ~ diet, data = coagulation)
summary(coag_aov)
```

This code produces a p-value which, if low enough, implies sound statistical evidence of an actual, not insignificant difference in the true values of the respective means μ_i, thereby indicating that we should reject the null hypothesis.