THE UNIVERSITY OF TEXAS AT AUSTIN
McCombs School of Business

STA 372.5 **Spring 2017**

MIDTERM EXAM
March 6, 2017

Directions:

(1) There are five questions on this exam with the point value for each question given following the question number.
(2) Please answer all questions in your blue book. No credit will be given for anything written on the exam itself.
(3) Be sure to show all your work so that partial credit can be given.
(4) The exam is closed-book. You are allowed one page of notes.

1. (10 points) The Excel spreadsheet given on the next contains the output from the additive classical method of seasonal decomposition applied to the logarithms of 32 quarters of sales values.

   The appropriate model for the sales values, denoted $Sales_t$, is the multiplicative model

   $$Sales_t = MT_t \times MS_t \times MI_t.$$

   This means

   $$\begin{aligned} \log(Sales_t) &= \log(MT_t \times MS_t \times MI_t) \\ &= \log(MT_t) + \log(MS_t) + \log(MI_t) \\ &= T_t + S_t + I_t \end{aligned}$$

   where $T_t = \log(MT_t)$, $S_t = \log(MS_t)$ and $I_t = \log(MI_t)$.

   Note that the value in cell F6 has been removed, and all the values in columns J and K have been replaced with asterisks.

   Compute the numerical value that has been removed from cell F6 (this is the cell with "**--A--**" in it).

Excel spreadsheet for question #1

Row	A	B	C	D	E	F	G	H	I	J	K
1	Qtr	Time	Sales	Y=Log(Sales)	T Estimate	S+I Estimate	Seasonal Sum	Seasonal Average	S Estimate	SeasonalAdj LogSales	SeasonalAdj Sales
2	Qtr 1	1	11.017	2.399					-0.144	******	******
3	Qtr 2	2	13.309	2.588					-0.053	******	******
4	Qtr 3	3	14.952	2.705	2.663	0.042	0.266	0.038	0.038	******	******
5	Qtr 4	4	17.460	2.860	2.702	0.158	1.109	0.158	0.158	******	******
6	Qtr 1	5	13.440	2.598	2.738	--A--	-1.005	-0.144	-0.144	******	******
7	Qtr 2	6	14.886	2.700	2.785	-0.084	-0.368	-0.053	-0.053	******	******
8	Qtr 3	7	17.917	2.886	2.834	0.052			0.038	******	******
9	Qtr 4	8	21.090	3.049	2.891	0.158			0.158	******	******
10	Qtr 1	9	16.447	2.800	2.950	-0.150			-0.144	******	******
11	Qtr 2	10	19.289	2.960	3.007	-0.047			-0.053	******	******
12	Qtr 3	11	22.184	3.099	3.066	0.034			0.038	******	******
13	Qtr 4	12	26.759	3.287	3.126	0.161			0.158	******	******
14	Qtr 1	13	20.748	3.032	3.182	-0.150			-0.144	******	******
15	Qtr 2	14	24.771	3.210	3.224	-0.014			-0.053	******	******
16	Qtr 3	15	27.164	3.302	3.261	0.041			0.038	******	******
17	Qtr 4	16	30.496	3.418	3.291	0.126			0.158	******	******
18	Qtr 1	17	24.427	3.196	3.327	-0.131			-0.144	******	******
19	Qtr 2	18	26.838	3.290	3.385	-0.095			-0.053	******	******
20	Qtr 3	19	33.358	3.507	3.442	0.065			0.038	******	******
21	Qtr 4	20	39.438	3.675	3.501	0.174			0.158	******	******
22	Qtr 1	21	29.883	3.397	3.552	-0.155			-0.144	******	******
23	Qtr 2	22	35.161	3.560	3.591	-0.031			-0.053	******	******
24	Qtr 3	23	38.289	3.645	3.639	0.007			0.038	******	******
25	Qtr 4	24	47.010	3.850	3.690	0.160			0.158	******	******
26	Qtr 1	25	36.614	3.600	3.747	-0.146			-0.144	******	******
27	Qtr 2	26	43.266	3.767	3.808	-0.041			-0.053	******	******
28	Qtr 3	27	49.084	3.894	3.869	0.025			0.038	******	******
29	Qtr 4	28	59.969	4.094	3.922	0.171			0.158	******	******
30	Qtr 1	29	46.539	3.840	3.972	-0.132			-0.144	******	******
31	Qtr 2	30	52.276	3.957	4.011	-0.055			-0.053	******	******
32	Qtr 3	31	60.461	4.102					0.038	******	******
33	Qtr 4	32	66.531	4.198					0.158	******	******

2. (10 points) The first six and last six observations of a data set containing 50 Y values are shown below. Using all available information in the R output on the following pages, what is an estimate of the second autocorrelation coefficient?

```
--------------------------------------------------------------------------------
#
#   Print first six rows and last six rows of the data table
#
cat ("\n", "First six rows of the data table are:", "\n", "\n")
print(head(Data_table))
cat ("\n", "Last six rows of the data table are:", "\n", "\n")
print(tail(Data_table))
--------------------------------------------------------------------------------
```

```
First six rows of the data table are:

           Y     Y_lag1    Y_lag2
1  0.6441769         NA        NA
2  0.8967247  0.6441769        NA
3  0.7755256  0.8967247 0.6441769
4  0.7353808  0.7755256 0.8967247
5 -0.2011973  0.7353808 0.7755256
6  0.1767425 -0.2011973 0.7353808

 Last six rows of the data table are:

            Y     Y_lag1     Y_lag2
45 -0.6019346 -0.2932829 -0.5952400
46 -0.5468473 -0.6019346 -0.2932829
47 -0.5468250 -0.5468473 -0.6019346
48  0.7894510 -0.5468250 -0.5468473
49 -0.4250275  0.7894510 -0.5468250
50 -0.1318604 -0.4250275  0.7894510
```

```
--------------------------------------------------------------------------------
reg_output_lag1 <- lm(Y ~ Y_lag1, Data_table)
print (summary(reg_output_lag1))
--------------------------------------------------------------------------------
```

```
Call:
lm(formula = Y ~ Y_lag1, data = Data_table)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.1463 -0.4020  0.0047  0.3557  1.0966 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept) -0.07590    0.07449  -1.019    0.313   
Y_lag1       0.42289    0.12923   3.272    0.002 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5128 on 47 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.1856,	Adjusted R-squared:  0.1682 
F-statistic: 10.71 on 1 and 47 DF,  p-value: 0.002003
```

---

reg_output_lag2 <- lm(Y ~ Y_lag2, Data_table)
print (summary(reg_output_lag2))

```
Call:
lm(formula = Y ~ Y_lag2, data = Data_table)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.19877 -0.34773 -0.09851  0.43054  1.02197 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.12587    0.08003  -1.573    0.123
Y_lag2       0.15635    0.13818   1.132    0.264

Residual standard error: 0.5465 on 46 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.02708,	Adjusted R-squared:  0.005928 
F-statistic:  1.28 on 1 and 46 DF,  p-value: 0.2637
```

---

reg_output_lag1and2 <- lm(Y ~ Y_lag1 + Y_lag2, Data_table)
print (summary(reg_output_lag1and2))

```
Call:
lm(formula = Y ~ Y_lag1 + Y_lag2, data = Data_table)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.12196 -0.38923  0.01179  0.34145  1.09337 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -0.09518    0.07597  -1.253   0.2167  
Y_lag1       0.39228    0.14599   2.687   0.0101 *
Y_lag2      -0.01056    0.14380  -0.073   0.9418  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.513 on 45 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.1616,	Adjusted R-squared:  0.1243 
F-statistic: 4.337 on 2 and 45 DF,  p-value: 0.01895
```

3. (30 points) Consider the annual sales for a company for a 20 year period, denoted $Sales_t$. The first six and last six observations for *Sales* and *log(Sales)* are shown below. It is annual data so no seasonal adjustment is required.

The appropriate model for $\log(Sales_t)$ is

$$\log(Sales_t) = \alpha + \beta\log(Sales_{t-1}) + \varepsilon_t \qquad\qquad \varepsilon_t \text{ iid } N(0, \sigma_\varepsilon^2).$$

Using the following R output, answer parts (a)-(c).

(a) (5 points) What is the forecast for $Sales_{21}$? Note that this is a one-year ahead forecast.

(b) (10 points) What is the forecast for $Sales_{22}$? Note that this is a two-year ahead forecast.

(c) (15 points) What is the 95% analytical confidence interval for $Sales_{22}$? Note that this is a confidence interval for the two-year ahead forecast.

```
-------------------------------------------------------------------------------
#
#   Print first six rows and last six rows of the data table
#
cat ("\n", "First six rows of the data table are:", "\n", "\n")
print(head(Data_table))
cat ("\n", "Last six rows of the data table are:", "\n", "\n")
print(tail(Data_table))
-------------------------------------------------------------------------------
```

```
First six rows of the data table are:

     Sales log_Sales
1 12.47379  2.523630
2 12.45055  2.521765
3 12.29717  2.509369
4 12.42643  2.519825
5 12.88176  2.555813
6 12.37840  2.515953

 Last six rows of the data table are:

      Sales log_Sales
15 14.94954  2.704681
16 14.79261  2.694128
17 15.20760  2.721795
18 15.23612  2.723669
19 14.17150  2.651233
20 13.80511  2.625038
```

```
------------------------------------------------------------------------------------------------------------
#
#   Create log(Sales(t-1))
#
Data_table$log_Sales_lag[2:20] <- Data_table$log_Sales[1:19]
is.na(Data_table$log_Sales_lag[1])
print (head(Data_table))
------------------------------------------------------------------------------------------------------------
```

```
     Sales log_Sales log_Sales_lag
1 12.47379  2.523630            NA
2 12.45055  2.521765      2.523630
3 12.29717  2.509369      2.521765
4 12.42643  2.519825      2.509369
5 12.88176  2.555813      2.519825
6 12.37840  2.515953      2.555813
```

```
------------------------------------------------------------------------------------------------------------
#
#   Regress log_Sales against log_Sales_lag
#
reg_output_lag <- lm(log_Sales ~ log_Sales_lag, Data_table)
cat ("\n", "Regression output for log_Sales = Alpha + Beta*log_Sales(-1) is:", "\n")
print (summary(reg_output_lag))
------------------------------------------------------------------------------------------------------------
```

```
Regression output for log_Sales = Alpha + Beta*log_Sales(-1) is:

Call:
lm(formula = log_Sales ~ log_Sales_lag, data = Data_table)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.060333 -0.022846 -0.001366  0.025857  0.049952 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)     0.4061     0.2850   1.425    0.172    
log_Sales_lag   0.8465     0.1091   7.756 5.55e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03384 on 17 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.7797,   Adjusted R-squared:  0.7667 
F-statistic: 60.16 on 1 and 17 DF,  p-value: 5.549e-07
```

4. (25 points) Consider the quarterly sales for a company for 40 quarters, denoted $Sales_t$. The first six and last six observations are shown below. Let A_t represent the seasonally adjusted values computed using the results from the *stl* command in R.

The appropriate model for $\log(A_t)$ is

$$\log(A_t) = \alpha + \beta Time_t + \varepsilon_t \qquad \varepsilon_t \text{ iid } N(0, \sigma_\varepsilon^2).$$

Using the following R output, answer parts (a) and (b).

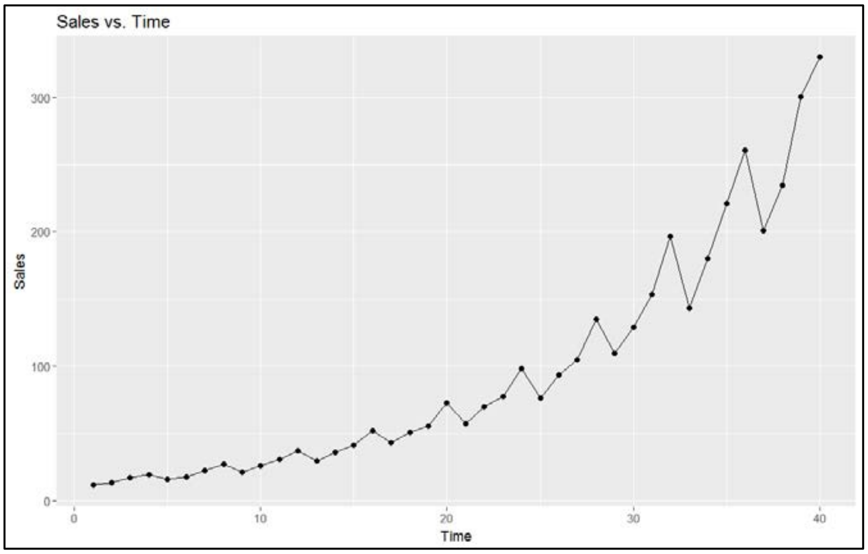
(a) (10 points) What is the forecast for $Sales_{42}$? Note that this is a two-quarter ahead forecast?

(b) (15 points) What is the 95% analytical confidence interval for $Sales_{42}$?

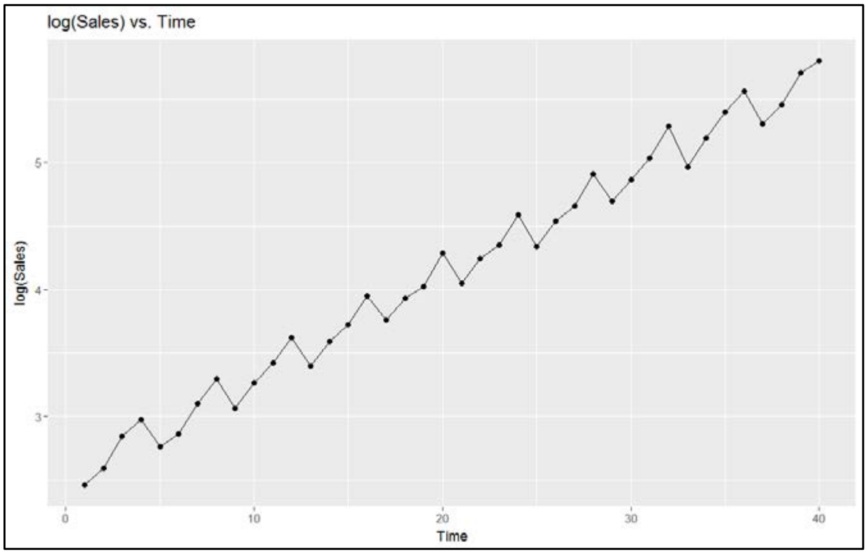
```
-----------------------------------------------------------------------------
#
#   Print first six rows and last six rows of the data table
#
cat ("\n", "First six rows of the data table are:", "\n", "\n")
print(head(Data_table))
cat ("\n", "Last six rows of the data table are:", "\n", "\n")
print(tail(Data_table))
-----------------------------------------------------------------------------
```

```
First six rows of the data table are:

  Quarter Time    Sales log_Sales
1    Qtr1    1 11.68142  2.458000
2    Qtr2    2 13.38653  2.594249
3    Qtr3    3 17.19181  2.844433
4    Qtr4    4 19.54054  2.972491
5    Qtr1    5 15.82944  2.761872
6    Qtr2    6 17.53484  2.864190

 Last six rows of the data table are:

   Quarter Time    Sales log_Sales
35    Qtr3   35 220.6938  5.396776
36    Qtr4   36 260.8576  5.563975
37    Qtr1   37 200.8297  5.302457
38    Qtr2   38 234.3443  5.456791
39    Qtr3   39 300.5312  5.705552
40    Qtr4   40 329.6956  5.798170
```

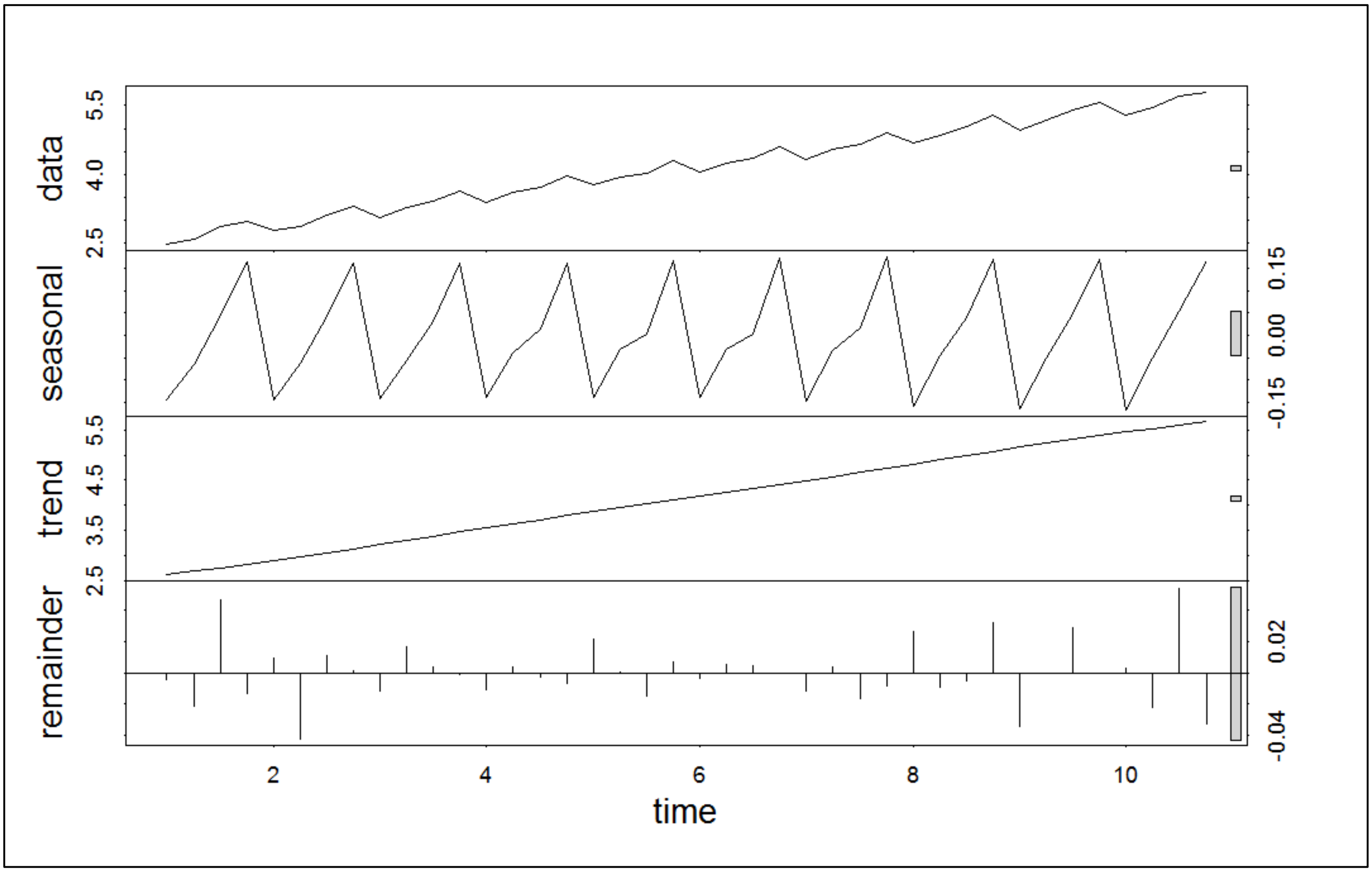
```
-----------------------------------------------------------------------------------------------------------------------------------------
#
#   Construct figure to plot Sales vs. Time
#
figure <- ggplot(Data_table, aes(x=Time,y=Sales))
figure <- figure + geom_line()
figure <- figure + geom_point()
figure <- figure + ggtitle("Sales vs. Time") + xlab("Time") + ylab("Sales")
print (figure)
-----------------------------------------------------------------------------------------------------------------------------------------
```




```
-----------------------------------------------------------------------------------------------------------------------------------------
#
#   Construct figure to plot log(Sales) vs. Time
#
figure <- ggplot(Data_table, aes(x=Time,y=log_Sales))
figure <- figure + geom_line()
figure <- figure + geom_point()
figure <- figure + ggtitle("log(Sales) vs. Time") + xlab("Time") + ylab("log(Sales)")
print (figure)
-----------------------------------------------------------------------------------------------------------------------------------------
```



```
-------------------------------------------------------------------------------------------
#
#   Save log(sales) data as a time series object
#
Log_Sales_time_series <- ts(Data_table[,4], frequency=4)
#
#   Use stl to decompose log(Sales)
#
fit <- stl(Log_Sales_time_series, s.window=7)
plot(fit)
Data_table$seasonal <- fit$time.series[,1]
print(Data_table$seasonal)
-------------------------------------------------------------------------------------------
```

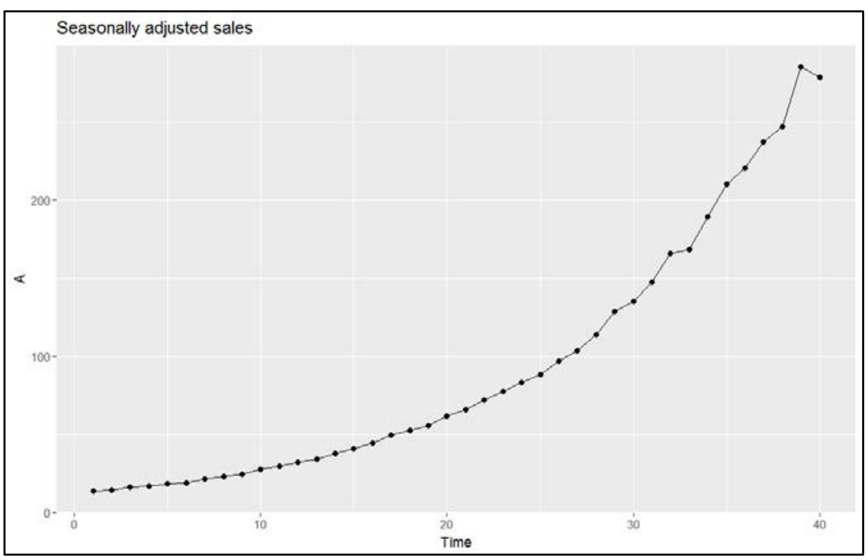



```
          Qtr1         Qtr2         Qtr3         Qtr4
1   -0.144911747 -0.064181900  0.045450409  0.162449810
2   -0.144206813 -0.061442947  0.041099297  0.161828276
3   -0.141869228 -0.056635752  0.032092852  0.160278325
4   -0.138002413 -0.039784048  0.012949114  0.162063155
5   -0.139038665 -0.030283379  0.003831182  0.165215398
6   -0.139228121 -0.030049012  0.003487867  0.171082490
7   -0.146953689 -0.035069418  0.015958649  0.174751982
8   -0.159196746 -0.044988447  0.039520222  0.169857785
9   -0.164119827 -0.050005040  0.048149808  0.168393209
10  -0.166628165 -0.052301467  0.052513806  0.167662553
```

```
--------------------------------------------------------------------------------
#
#   Compute seasonally adjusted sales values, denoted A
#
Data_table$A <- exp(Data_table$log_Sales - Data_table$seasonal)
#
#   Compute log(A)
#
Data_table$log_A <- log(Data_table$A)
print (head(Data_table))
--------------------------------------------------------------------------------
```

```
  Quarter Time    Sales log_Sales    seasonal        A    log_A
1    Qtr1    1 11.68142  2.458000 -0.14491175 13.50300 2.602912
2    Qtr2    2 13.38653  2.594249 -0.06418190 14.27388 2.658431
3    Qtr3    3 17.19181  2.844433  0.04545041 16.42793 2.798983
4    Qtr4    4 19.54054  2.972491  0.16244981 16.61061 2.810042
5    Qtr1    5 15.82944  2.761872 -0.14420681 18.28495 2.906078
6    Qtr2    6 17.53484  2.864190 -0.06144295 18.64602 2.925633
```

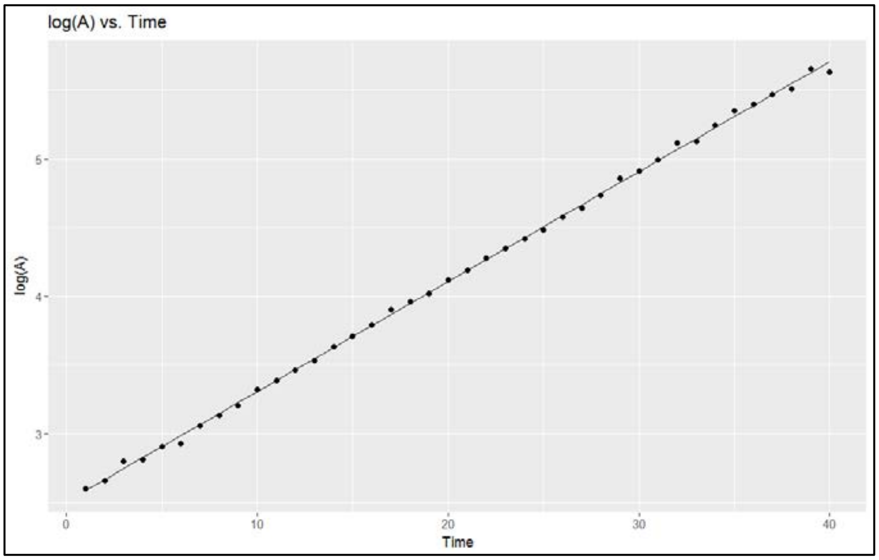
```
--------------------------------------------------------------------------------
#
#   Plot seasonally adjusted values from stl
#
figure <- ggplot(Data_table)
figure <- figure + geom_point(aes(x=Time, y=A))
figure <- figure + geom_line(aes(x=Time, y=A))
figure <- figure + scale_y_continuous()
figure <- figure + ggtitle("Seasonally adjusted sales") + xlab("Time") + ylab("A")
print (figure)
--------------------------------------------------------------------------------
```



```
--------------------------------------------------------------------------------
#
#   Regress log(A) against Time
#
reg_output <- lm(log_A ~ Time, Data_table)
print (summary(reg_output))
Data_table$fitted <- reg_output$fitted.values
--------------------------------------------------------------------------------
```

```
Call:
lm(formula = log_A ~ Time, data = Data_table)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.074794 -0.012377  0.001779  0.012340  0.052860 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.5061889  0.0082201   304.9   <2e-16 ***
Time        0.0799778  0.0003494   228.9   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02551 on 38 degrees of freedom
Multiple R-squared:  0.9993,   Adjusted R-squared:  0.9993 
F-statistic: 5.24e+04 on 1 and 38 DF,  p-value: < 2.2e-16
```

```
--------------------------------------------------------------------------------
#
#   Construct figure to plot log(A) vs. Time with Fitted Values
#
figure <- ggplot(Data_table, aes(x=Time,y=log_A))
figure <- figure + scale_y_continuous()
figure <- figure + geom_point()
figure <- figure + geom_line(aes(x=Time,y=fitted))
figure <- figure + ggtitle("log(A) vs. Time") + xlab("Time") + ylab("log(A)")
print (figure)
--------------------------------------------------------------------------------
```



5. (25 points) Consider the monthly sales for a company for a 100 month period, denoted $Sales_t$. The first six and last six observations are shown below. There is no seasonality in these data so no seasonal adjustment is required.

The appropriate model for log($Sales_t$) is

$$\log(Sales_t) = \alpha + \beta Time_t + \varepsilon_t \qquad \varepsilon_t \text{ iid } N(0, \sigma_\varepsilon^2).$$

Using the following R output, answer parts (a) and (b).

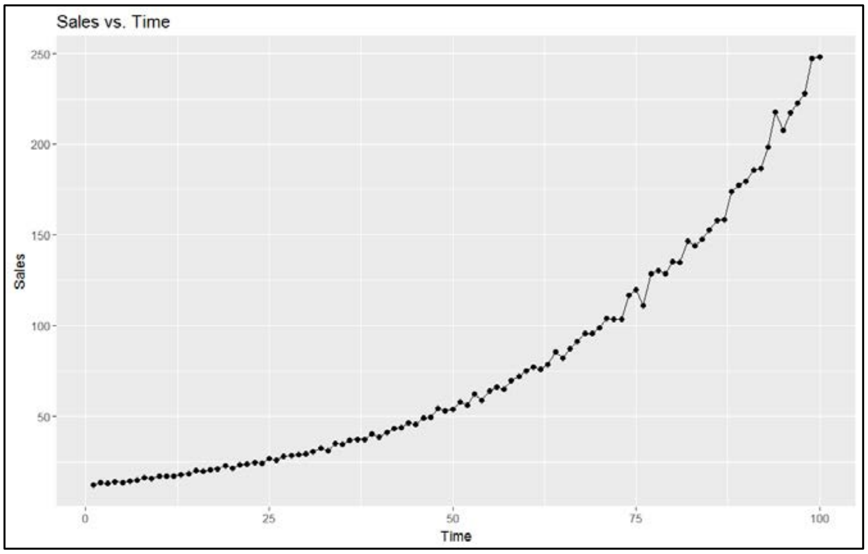
(a) (10 points) What is the forecast for $Sales_{101}$? Note that this is a one-month ahead forecast.

(b) (15 points) What is the 90% <u>empirical</u> confidence interval for $Sales_{101}$? Note that you are asked for a 90% confidence interval (not a 95% confidence interval).

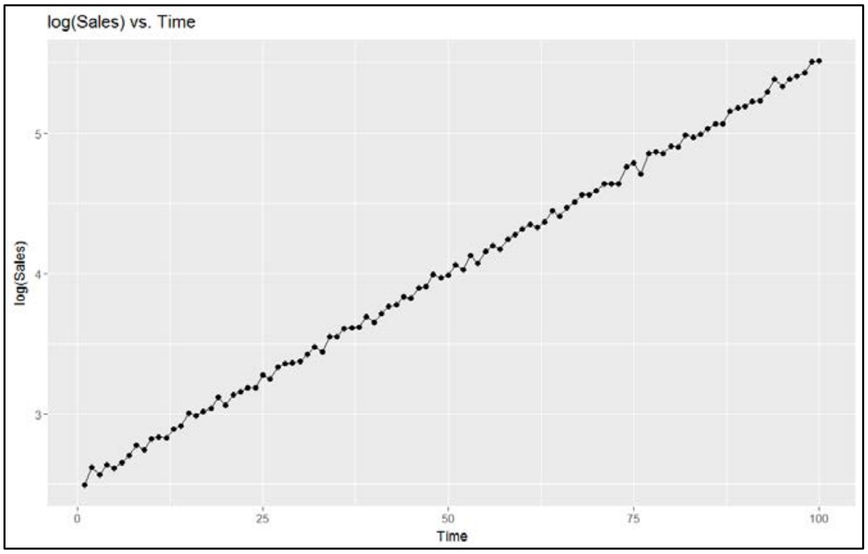
```
--------------------------------------------------------------------------------------------------------------------------------------
#
#   Print first six rows and last six rows of the data table
#
cat ("\n", "First six rows of the data table are:", "\n", "\n")
print(head(Data_table))
cat ("\n", "Last six rows of the data table are:", "\n", "\n")
print(tail(Data_table))
--------------------------------------------------------------------------------------------------------------------------------------
```

```
First six rows of the data table are:

  Time    Sales log_Sales
1    1 12.09799  2.493040
2    2 13.70319  2.617628
3    3 13.03478  2.567621
4    4 13.94505  2.635125
5    5 13.62469  2.611884
6    6 14.22523  2.655017

 Last six rows of the data table are:

    Time    Sales log_Sales
95    95 207.5469  5.335357
96    96 217.2667  5.381126
97    97 222.7780  5.406176
98    98 227.7659  5.428318
99    99 246.9773  5.509296
100  100 247.8863  5.512970
```

```
--------------------------------------------------------------------------------------------
#
#   Construct figure to plot Sales vs. Time
#
figure <- ggplot(Data_table, aes(x=Time,y=Sales))
figure <- figure + geom_line()
figure <- figure + geom_point()
figure <- figure + ggtitle("Sales vs. Time") + xlab("Time") + ylab("Sales")
print (figure)
--------------------------------------------------------------------------------------------
```



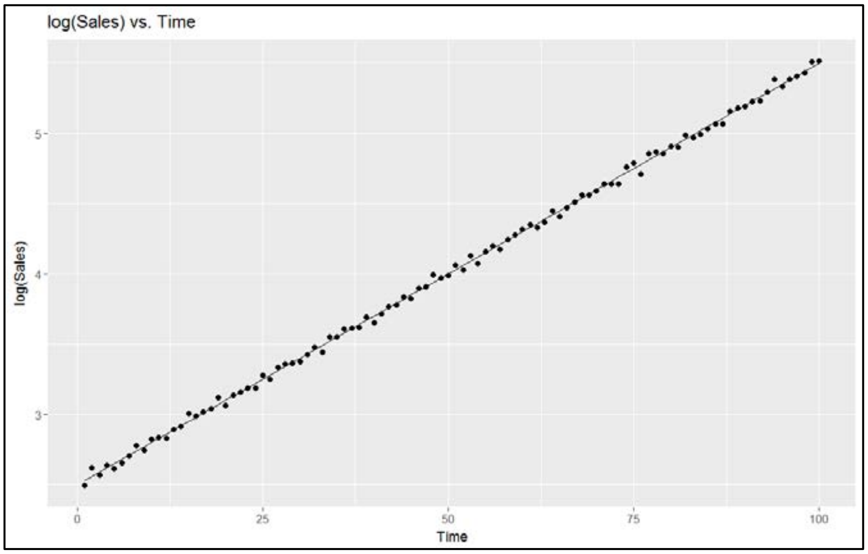

```
--------------------------------------------------------------------------------------------
#
#   Construct figure to plot log(Sales) vs. Time
#
figure <- ggplot(Data_table, aes(x=Time,y=log_Sales))
figure <- figure + geom_line()
figure <- figure + geom_point()
figure <- figure + ggtitle("log(Sales) vs. Time") + xlab("Time") + ylab("log(Sales)")
print (figure)
--------------------------------------------------------------------------------------------
```



```
-----------------------------------------------------------------------------------------------------------
#
#   Regress log(Sales) against Time
#
reg_output <- lm(log_Sales ~ Time, Data_table)
print (summary(reg_output))
Data_table$fitted <- reg_output$fitted.values
-----------------------------------------------------------------------------------------------------------
```

```
Call:
lm(formula = log_Sales ~ Time, data = Data_table)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.068079 -0.022328  0.000523  0.017231  0.064410 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.501e+00  5.577e-03   448.4   <2e-16 ***
Time        2.997e-02  9.587e-05   312.6   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02767 on 98 degrees of freedom
Multiple R-squared:  0.999,	Adjusted R-squared:  0.999 
F-statistic: 9.775e+04 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
-----------------------------------------------------------------------------------------------------------
#
#   Construct figure to plot log(Sales) vs. Time with Fitted Values
#
figure <- ggplot(Data_table, aes(x=Time,y=log_Sales))
figure <- figure + geom_point()
figure <- figure + geom_line(aes(x=Time,y=fitted))
figure <- figure + ggtitle("log(Sales) vs. Time") + xlab("Time") + ylab("log(Sales)")
print (figure)
-----------------------------------------------------------------------------------------------------------
```



```
------------------------------------------------------------------------------------------------------------------------
#
#   Sort and print residuals
#
Data_table$residuals <- reg_output$residuals
residuals_sort <- sort(Data_table$residuals)
cat ("\n", "Sorted residuals from regression of log(Sales) vs. Time are:", "\n", "\n")
print(residuals_sort)
------------------------------------------------------------------------------------------------------------------------
```

```
Sorted residuals from regression of log(Sales) vs. Time are:

  [1] -6.807946e-02 -4.825721e-02 -4.819552e-02 -4.767503e-02 -4.705557e-02 -4.225291e-02 -4.211907e-02 -3.851188e-02
  [9] -3.745709e-02 -3.694262e-02 -3.320768e-02 -3.301532e-02 -3.101100e-02 -3.000812e-02 -2.943629e-02 -2.865082e-02
 [17] -2.806759e-02 -2.798952e-02 -2.635152e-02 -2.633312e-02 -2.586458e-02 -2.535307e-02 -2.509003e-02 -2.386584e-02
 [25] -2.282527e-02 -2.216179e-02 -1.986050e-02 -1.929512e-02 -1.704966e-02 -1.505675e-02 -1.434641e-02 -1.386238e-02
 [33] -1.276932e-02 -1.219608e-02 -1.028724e-02 -9.732423e-03 -9.024174e-03 -8.189930e-03 -7.727617e-03 -6.940145e-03
 [41] -5.947905e-03 -5.282207e-03 -4.342741e-03 -4.137387e-03 -3.736286e-03 -3.056595e-03 -1.900144e-03 -1.523768e-03
 [49] -1.449933e-03 -9.912174e-08  1.047056e-03  1.100696e-03  2.293633e-03  2.536493e-03  3.024432e-03  3.248831e-03
 [57]  3.276552e-03  3.356904e-03  5.175540e-03  6.701470e-03  6.882321e-03  7.580854e-03  7.842566e-03  8.407574e-03
 [65]  8.417921e-03  9.050760e-03  1.051954e-02  1.345509e-02  1.424310e-02  1.470385e-02  1.483997e-02  1.496978e-02
 [73]  1.522532e-02  1.558431e-02  1.669238e-02  1.884614e-02  1.973642e-02  2.041827e-02  2.090869e-02  2.336393e-02
 [81]  2.557681e-02  2.568434e-02  2.701804e-02  2.716529e-02  2.787313e-02  2.826448e-02  3.006316e-02  3.241870e-02
 [89]  3.378758e-02  3.707105e-02  3.739506e-02  3.996256e-02  4.116169e-02  4.127058e-02  4.928446e-02  5.229149e-02
 [97]  5.443857e-02  5.715699e-02  5.774846e-02  6.440992e-02
```