MGMT 6160 Exam Answer Sheet

Student Name: Scout Oatman-Gaitan Student ID.NO: 661974037

The exam is composed of 5 multiple choice questions (10 points), 20 short answer questions (50 points), and 2 case problems (40 points). The exam is worth 100 points. Please use the attached answer sheet to write down all the solutions and submit only the answer sheet by April 16th at 8:00 am. Good luck!

Multiple Choice Questions (write A, B, C or D)

1	2	3	4	5
B	B	A	A	C

Short Answer Questions (feel free to enlarge the text box if necessary)

6.

Moving averages – this technique takes the averages of subsets of the data. This technique is mostly used in time series analysis and gets rid of the fluctuations in data to better understand the long-term trend.

Exponential smoothing – this technique is like moving averages except the averages are weighted. In moving averages, each average is equally weighted but in exponential smoothing exponential functions are used to influence the weight of an average over time. This makes the model slightly more responsive to recent observations.

7.

A model that has zero error on its fit to training data is likely a bad model because it is overfit to that data. The goal is to have a model that allows for generality so that it is flexible and responsive to new data. An overfit model doesn’t pick up on general trends but instead is really good at understanding the training data.

8.

```
seq(-10,20,3)
```

MGMT 6160 Exam Answer Sheet

9.

```
matrix(1:30, nrow = 5, byrow = FALSE)
```

10.

(224 + 3258) / (3595) = 0.9686

Accuracy = 96.86%

11.

```
df[c(10,20,30,40,50),c(1,3,5,7,9)]
```

12.

```
cor(df$FARE, df$DISTANCE)
```

Correlation coefficient: 0.67

13.

```
chi <- subset(df, df$S_CITY == "Chicago            IL")
chi
mean(chi$DISTANCE)
```

Average distance of flights from Chicago: 891.3778

MGMT 6160 Exam Answer Sheet

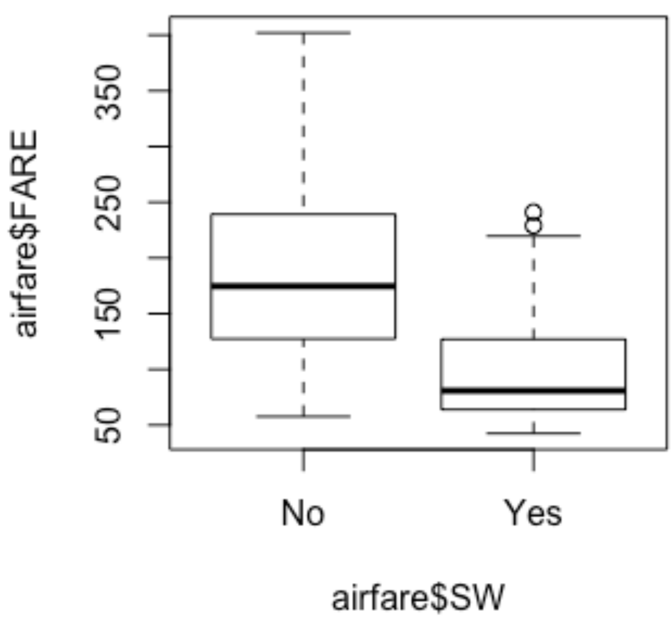
14.

```
sapply(df, class)

df.pca <- prcomp(df[,c(5,6,9:13, 16:18)], scale = TRUE)

summary(df.pca)
```

Yes, we should scale the data in order to obtain the best results

15.

```
boxplot(df$FARE~df$SW)
```



MGMT 6160 Exam Answer Sheet

16.

```
actual <- c(15,14,18)

predicted <- c(12,15,16)

mean((actual - predicted)^2)

Output: 4.67
```

17.

```
q_17 <- data.frame(age = c(25, 53), spent = c(350, 420))

q_17

library(philentropy)

distance(q_17, method = "euclidean")

Euclidean Distance: 75.39231
```

18.

The top decile contains the 10% of the population that is the most likely respond. The first bar shows that the top 10% likely to respond have a mean response of over 2. The second bar shows that the next 10% most likely to respond have a mean response of over 1.5. These two groups, 20% of the population, are most likely to respond and will be the most profitable to go after for the new product.

19.

The first thing the model looks at is income. If the customer makes under 99, they are not accepted. If they make over 99, we look at education.

If their education level is under 2 and their family is less than 3 they get rejected. If their education is under 2 and their family is not less than 3 they get accepted.

If their education is 2 or more and their income is less than 117, they get rejected. If their education is 2 or more and their income is not less than 117, they get accepted.

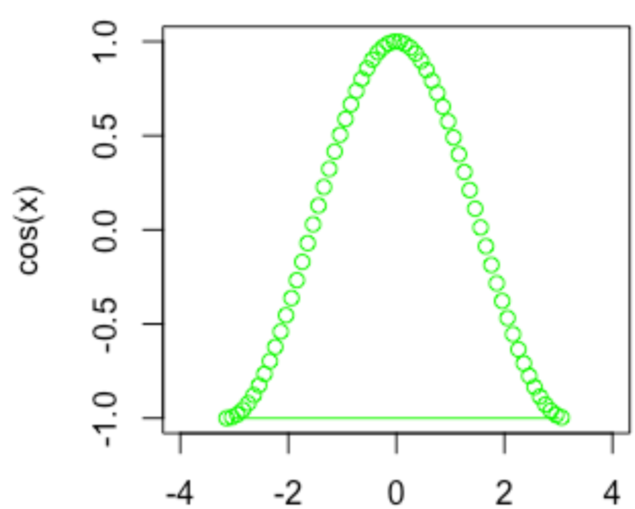
MGMT 6160 Exam Answer Sheet

20.

Variable	Quantitative	Qualitative
Your name		X
Your height	X	
Your income	X	
The month in which you were born		X
Your home address house number	X	
The number of texts you send each day	X	
The type of phone you have		X
Your satisfaction with your mobile service provider, measured on a scale of one to five	X	

21.

```
x <- seq(-pi,pi,.1)

plot(x,cos(x), xlim = c(-4,4), col = "green", xlab = '')

lines(c(-3,3), c(-1,-1), col = "green")
```



MGMT 6160 Exam Answer Sheet

22.

1. The logit as a function of the predictors: LOGIT = (-24.721+89.834*TotExp.+9.371*TotLns)

2. The odds as a function of the predictors: ODDS = e^(-24.721+89.834*TotExp.+9.371*TotLns)
= e^LOGIT

3. The probability as a function of the predictors: PROB = 1 / (1+ e^(-24.721+89.834*TotExp.+9.371*TotLns))
= 1/(1+odds)

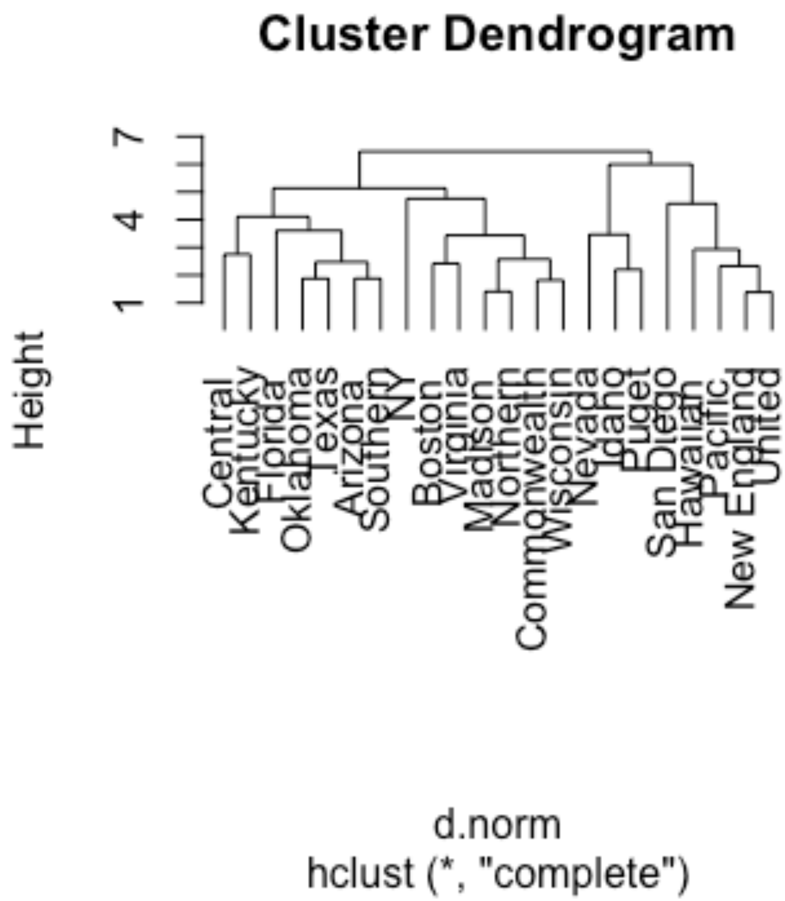
23.

```
util <- read.csv("Utilities(1).csv")

row.names(util) <- util[,1]

util <- util[,-1]

utilities.df.norm <- sapply(util, scale)

row.names(utilities.df.norm) <- row.names(util)
```

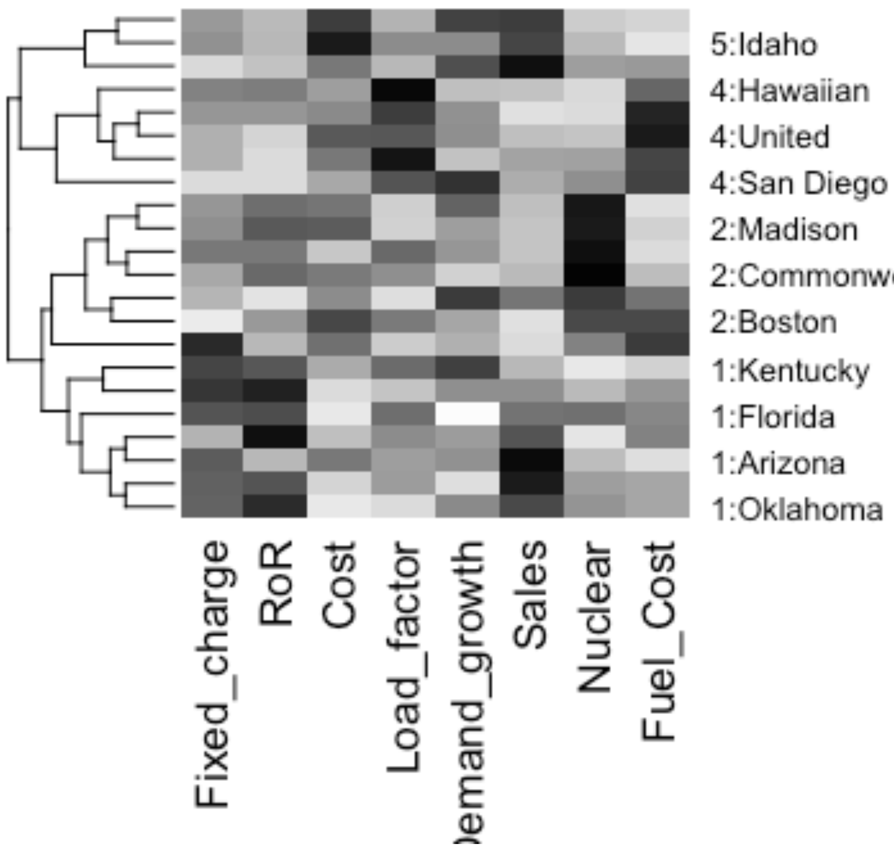
MGMT 6160 Exam Answer Sheet

24.

```
d.norm <- dist(utilities.df.norm, method = "euclidean")

hclust1 <- hclust(d.norm, method = "complete")

plot(hclust1, hang = -1)
```



25.



Cluster 1: High sales, RoR & Fixed charge ++ low cost, nuclear, and fuel cost

Cluster 2: High nuclear & RoR ++ low sales and fuel cost

Cluster 3: High fuel cost and fixed charge ++ low RoR and sales

Cluster 4: High fuel cost and load factor ++ low nuclear and sales

Cluster 5: High sales and cost ++ low fuel cost and RoR

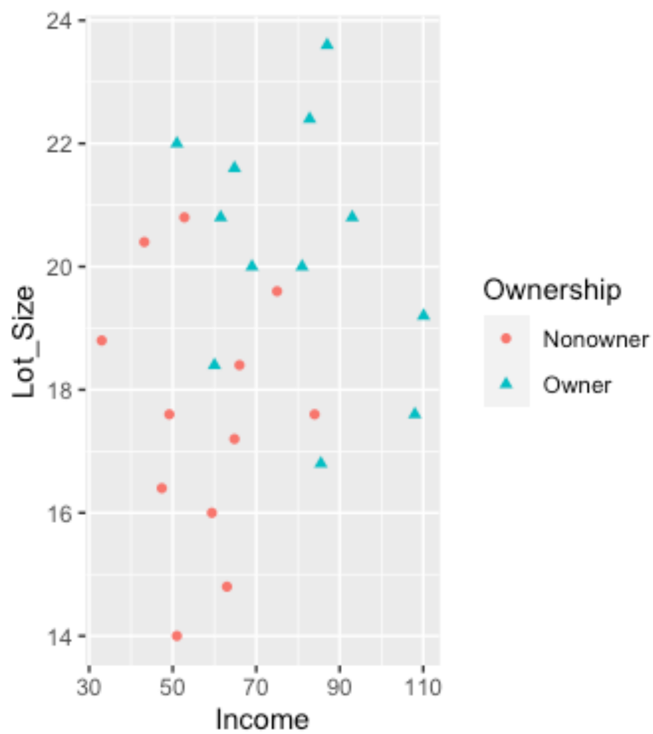
MGMT 6160 Exam Answer Sheet

Case Problems

26. a

```
ggplot(mowers, aes(x = Income, y = Lot_Size, color = Ownership, shape = Ownership)) +
  geom_point()
```

Owners have a higher income and bigger lot sizes



26. b

```
log.reg <- glm(Ownership ~., data = mowers, family = "binomial")
summary(log.reg)
```

The income coefficient is .1109 and this shows that it is a less impactful variable than Lot_size (coefficient = 0.9638). Therefore, lot_size is a more informative when predicting ownership.

MGMT 6160 Exam Answer Sheet

26. c

Prediction	Nonowner	Owner
Nonowner	10	2
Owner	2	10

Among nonowners, the model classified 10/12 (83.3%) correctly. The accuracy is 83.3%

26. d

To increase the percentage of correctly classified nonowners, the cutoff probability should be increased. That way higher probabilities are included in the nonowner category. Note that this will cause more improperly categorized owners though if the cutoff is too high.

26. e

Our model predicted at 0.497216567 that this person is a homeowner. That means the model would predict this person is not a homeowner at a 0.5 cutoff.

26. f

94.8 = 0.497907123

94.9 = 0.500678575

The minimum income with 16k sq ft to be classified as an owner is 94.9.

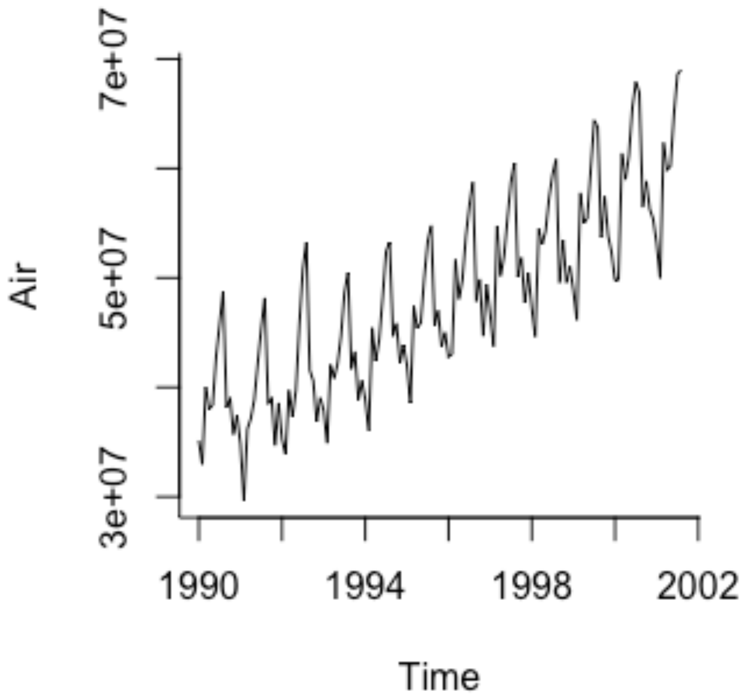
MGMT 6160 Exam Answer Sheet

26. g

```
mowers <- read.csv("RidingMowers(1).csv")
## A
library(ggplot2)
ggplot(mowers, aes(x = Income, y = Lot_Size, color = Ownership, shape = Ownership)) +
  geom_point()
## B
log.reg <- glm(Ownership ~ Income + Lot_Size, data = mowers, family = "binomial")
summary(log.reg)
#income coef 0.1109
#lot size coef 0.9638

## C
pred <- predict(log.reg, newdata = mowers, type = "response")
library(caret)
confusionMatrix(as.factor(ifelse(pred > 0.5, "Owner","Nonowner")), mowers$Ownership)
## D
# Responded on the test
## E
library(tidyverse)
mowers1 <- mowers %>% add_row(Income = 60, Lot_Size = 20)
pred1 <- predict(log.reg, newdata = mowers1, type = "response")
pred1

## F
mowers2 <- mowers %>% add_row(Income = 94.9, Lot_Size = 16)
pred2 <- predict(log.reg, newdata = mowers2, type = "response")
pred2
```

MGMT 6160 Exam Answer Sheet

27. a



We can see a general upward trend from 1990 to 2001. While there are some seasonal and predictable cyclic trends, overall the trend is upward. It would take a major event, like 9-11, to cause this upward trend to be distrupted.

MGMT 6160 Exam Answer Sheet

27. b

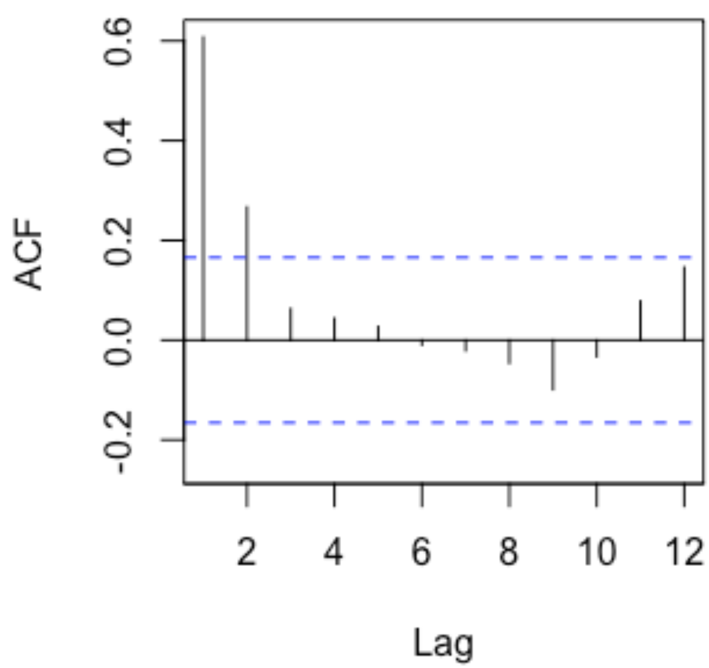
```
Call:
tslm(formula = air_train ~ trend + season, lambda = 0)

Residuals:
     Min       1Q    Median       3Q      Max
-0.101591 -0.017260 -0.002175  0.017248  0.083200

Coefficients:
            Estimate Std. Error  t value Pr(>|t|)
(Intercept)  1.732e+01  1.011e-02 1713.044  < 2e-16 ***
trend        3.604e-03  6.595e-05   54.640  < 2e-16 ***
season2     -6.519e-02  1.286e-02   -5.069 1.38e-06 ***
season3      1.338e-01  1.286e-02   10.407  < 2e-16 ***
season4      8.473e-02  1.286e-02    6.588 1.07e-09 ***
season5      1.105e-01  1.286e-02    8.595 2.61e-14 ***
season6      1.884e-01  1.286e-02   14.642  < 2e-16 ***
season7      2.501e-01  1.287e-02   19.439  < 2e-16 ***
season8      2.731e-01  1.287e-02   21.221  < 2e-16 ***
season9      7.314e-02  1.315e-02    5.562 1.50e-07 ***
season10     1.029e-01  1.315e-02    7.823 1.74e-12 ***
season11     1.361e-02  1.315e-02    1.035 0.302718
season12     5.188e-02  1.315e-02    3.944 0.000132 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0315 on 127 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 8.792e+17 on 12 and 127 DF,  p-value: < 2.2e-16
```

MGMT 6160 Exam Answer Sheet

27. c

February and August are both statistically significant at a high level – ie *** or a < 0.001 level. With a low p-value, we can reject the null hypothesis.

27. d



There isn’t really a pattern (perhaps something cubic) but no consistent upward or downward pattern would suggest that there is no significant autocorrelation. Additionally, there values are relatively low (mostly under +/-0.4 so we can assume low serial correlation.

MGMT 6160 Exam Answer Sheet

27. e

```
Call:
tslm(formula = rail_train ~ poly(trend, 2, raw = TRUE) + season)


Residuals:
     Min       1Q   Median       3Q      Max
-83002973 -17124784   2921271  18582765  74299534


Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                506417584   10859233  46.635  < 2e-16 ***
poly(trend, 2, raw = TRUE)1  -2025825     246700  -8.212 2.22e-13 ***
poly(trend, 2, raw = TRUE)2      8010       1695   4.725 6.04e-06 ***
season2                    -21471834   11913374  -1.802 0.073884 .
season3                     73933460   11913852   6.206 7.23e-09 ***
season4                     73730823   11914641   6.188 7.87e-09 ***
season5                     84066907   11915737   7.055 1.02e-10 ***
season6                    108355984   11917142   9.092 1.75e-15 ***
season7                    158005456   11918857  13.257  < 2e-16 ***
season8                    165851861   11920888  13.913  < 2e-16 ***
season9                     42825983   12191216   3.513 0.000616 ***
season10                    58146152   12192117   4.769 5.02e-06 ***
season11                    44863290   12193188   3.679 0.000345 ***
season12                    70687301   12194431   5.797 5.12e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 29180000 on 126 degrees of freedom
Multiple R-squared:  0.8442,	Adjusted R-squared:  0.8281
F-statistic:  52.5 on 13 and 126 DF,  p-value: < 2.2e-16
```

MGMT 6160 Exam Answer Sheet

MGMT 6160 Exam Answer Sheet

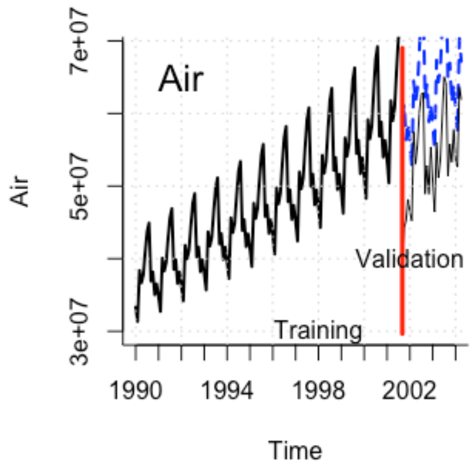
27. f

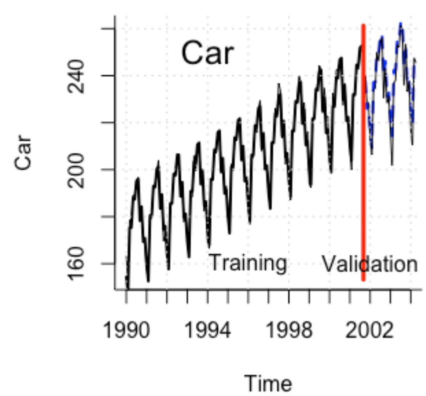
```
Call:
tslm(formula = car_train ~ trend + season)


Residuals:
    Min      1Q  Median      3Q     Max
-6.2254 -1.5725 -0.1145  1.0383  8.9699


Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 153.880889   0.807119 190.655  < 2e-16 ***
trend         0.429216   0.005265  81.516  < 2e-16 ***
season2      -7.348382   1.026645  -7.158 5.83e-11 ***
season3      20.207402   1.026685  19.682  < 2e-16 ***
season4      19.749020   1.026753  19.234  < 2e-16 ***
season5      31.898138   1.026847  31.064  < 2e-16 ***
season6      30.924755   1.026969  30.113  < 2e-16 ***
season7      38.353873   1.027117  37.341  < 2e-16 ***
season8      38.717158   1.027293  37.689  < 2e-16 ***
season9      20.587781   1.049757  19.612  < 2e-16 ***
season10     26.104020   1.049823  24.865  < 2e-16 ***
season11     10.688441   1.049916  10.180  < 2e-16 ***
season12     11.857407   1.050035  11.292  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 2.515 on 127 degrees of freedom
Multiple R-squared:  0.989,	Adjusted R-squared:  0.9879
F-statistic: 948.4 on 12 and 127 DF,  p-value: < 2.2e-16
```

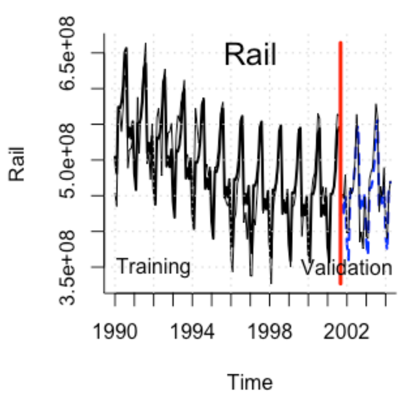
MGMT 6160 Exam Answer Sheet

MGMT 6160 Exam Answer Sheet

27. g







27. h

9-11 dramatically effected air travel usage. You can see in the graphic that the numbers after 9-11 are dramatically lower than predicted. Car and rail usage went relatively unchanged though. You can barely even see the blue predicted line of car and rail because the actual numbers go right over the predicted. Air was definitely very effected though.

MGMT 6160 Exam Answer Sheet

27. i

```
df <- read.csv("Sept11Travel.csv")
names(df)[names(df) == "Air.RPM..000s."] <- "Air"
names(df)[names(df) == "Rail.PM"] <- "Rail"
names(df)[names(df) == "VMT..billions."] <- "Vehicle"
## A
library(forecast)
air.ts <- ts(df$Air,start = c(1990,1),end = c(2004,4),freq=12)
rail.ts <- ts(df$Rail,start = c(1990,1),end = c(2004,4),freq=12)
vehicle.ts <- ts(df$Vehicle,start = c(1990,1),end = c(2004,4),freq=12)
nValid <- 32
air_train <- window(air.ts, start = c(1990,1), end = c(2001,8))
air_valid <- window(air.ts, start = c(2001,9), end = c(2004,4))
plot(air_train,xlab="Time",ylab="Air",ylim=c(min(df$Air), max(df$Air)),bty="l")
## B
train.lm.season <- tslm(air_train ~ trend + season, lambda = 0)
train.lm.season.pred <- forecast(train.lm.season, h=nValid, level=0)
print(summary(train.lm.season))
## C Done
## D
Acf(train.lm.season$residuals,lag.max=12, main="")
## E
rail_train <- window(rail.ts, start = c(1990,1), end = c(2001,8))
rail_valid <- window(rail.ts, start = c(2001,9), end = c(2004,4))
train.lm.quad <- tslm(rail_train ~ poly(trend, 2, raw=TRUE) + season)
train.lm.quad.pred <- forecast(train.lm.quad, h=nValid, level=0)
print(summary(train.lm.quad))
## F
car_train <- window(vehicle.ts, start = c(1990,1), end = c(2001,8))
car_valid <- window(vehicle.ts, start = c(2001,9), end = c(2004,4))
train.linear.season <- tslm(car_train ~ trend + season)
train.linear.pred <- forecast(train.linear.season, h=nValid, level=0)
print(summary(train.linear.season))


## G
# Air
```

MGMT 6160 Exam Answer Sheet

If part g is unreadable, here is the code:

```
df <- read.csv("Sept11Travel.csv")
names(df)[names(df) == "Air.RPM..000s."] <- "Air"
names(df)[names(df) == "Rail.PM"] <- "Rail"
names(df)[names(df) == "VMT..billions."] <- "Vehicle"

## A
library(forecast)
air.ts <- ts(df$Air,start = c(1990,1),end = c(2004,4),freq=12)
rail.ts <- ts(df$Rail,start = c(1990,1),end = c(2004,4),freq=12)
vehicle.ts <- ts(df$Vehicle,start = c(1990,1),end = c(2004,4),freq=12)

nValid <- 32
air_train <- window(air.ts, start = c(1990,1), end = c(2001,8))
air_valid <- window(air.ts, start = c(2001,9), end = c(2004,4))

plot(air_train,xlab="Time",ylab="Air",ylim=c(min(df$Air), max(df$Air)),bty="l")

## B
train.lm.season <- tslm(air_train ~ trend + season, lambda = 0)
train.lm.season.pred <- forecast(train.lm.season, h=nValid, level=0)
print(summary(train.lm.season))

## C Done

## D
Acf(train.lm.season$residuals,lag.max=12, main="")

## E
rail_train <- window(rail.ts, start = c(1990,1), end = c(2001,8))
rail_valid <- window(rail.ts, start = c(2001,9), end = c(2004,4))
```

```
train.lm.quad <- tslm(rail_train ~ poly(trend, 2, raw=TRUE) + season)
train.lm.quad.pred <- forecast(train.lm.quad, h=nValid, level=0)
print(summary(train.lm.quad))

## F
car_train <- window(vehicle.ts, start = c(1990,1), end = c(2001,8))
car_valid <- window(vehicle.ts, start = c(2001,9), end = c(2004,4))
train.linear.season <- tslm(car_train ~ trend + season)
train.linear.pred <- forecast(train.linear.season, h=nValid, level=0)
print(summary(train.linear.season))

## G
# Air
plot(train.lm.season.pred,  ylab = "Air", ,bty='l',xlab = "Time",xaxt="n",
ylim=c(min(df$Air), max(df$Air)),xlim = c(1990,2004), main = "", flty = 2)
axis(1, at = seq(1990, 2004, 1))
lines(train.lm.season$fitted, lwd = 2)
lines(air_valid)
grid()
lines(c(2001.67, 2001.67), c(min(df$Air), max(df$Air)),lwd=3,col="red")
text(1998, 30000000, "Training",cex=1)
text(2002, 40000000, "Validation",cex=1)
text(1992, 65000000, "Air", cex=1.5)

# Rail
plot(train.lm.quad.pred,  ylab = "Rail", ,bty='l',xlab = "Time",xaxt="n",
ylim=c(min(df$Rail), max(df$Rail)),xlim = c(1990,2004), main = "", flty = 2)
axis(1, at = seq(1990, 2004, 1))
lines(train.lm.quad$fitted, lwd = 2)
lines(rail_valid)
grid()
lines(c(2001.67, 2001.67), c(min(df$Rail), max(df$Rail)),lwd=3,col="red")
text(1992, 350000000, "Training",cex=1)
```

```
text(2002, 350000000, "Validation",cex=1)
text(1997, 650000000, "Rail", cex=1.5)

# Car
plot(train.linear.pred,  ylab = "Car", ,bty='l',xlab = "Time",xaxt="n",
ylim=c(min(df$Vehicle), max(df$Vehicle)),xlim = c(1990,2004), main = "", flty = 2)
axis(1, at = seq(1990, 2004, 1))
lines(train.linear.season$fitted, lwd = 2)
lines(car_valid)
grid()
lines(c(2001.67, 2001.67), c(min(df$Vehicle), max(df$Vehicle)),lwd=3,col="red")
text(1996, 160, "Training",cex=1)
text(2002, 160, "Validation",cex=1)
text(1994, 250, "Car", cex=1.5)
```