[STAT 4400] Exam-1

Michael Ghattas

2/22/2022

Part 1:

(1)

True

(2)

False

(3)

True

(4)

True

(5)

False

(6)

True

(7)

True

(8)

True

(9)

True

(10)

False

(11)

True

(12)

False

(13)

False

(14)

True

(15)

False

(16)

True

(17)

True (IF by reflects we mean cause! As discussed with Bhawneet.)

(18)

True

(19)

False

(20)

False (Correlated Explanatory Variables: If there are very many variables, it is likely that they will be highly correlated, meaning that some variables or sets of variables are measuring similar things. As discussed with Bhawneet.)

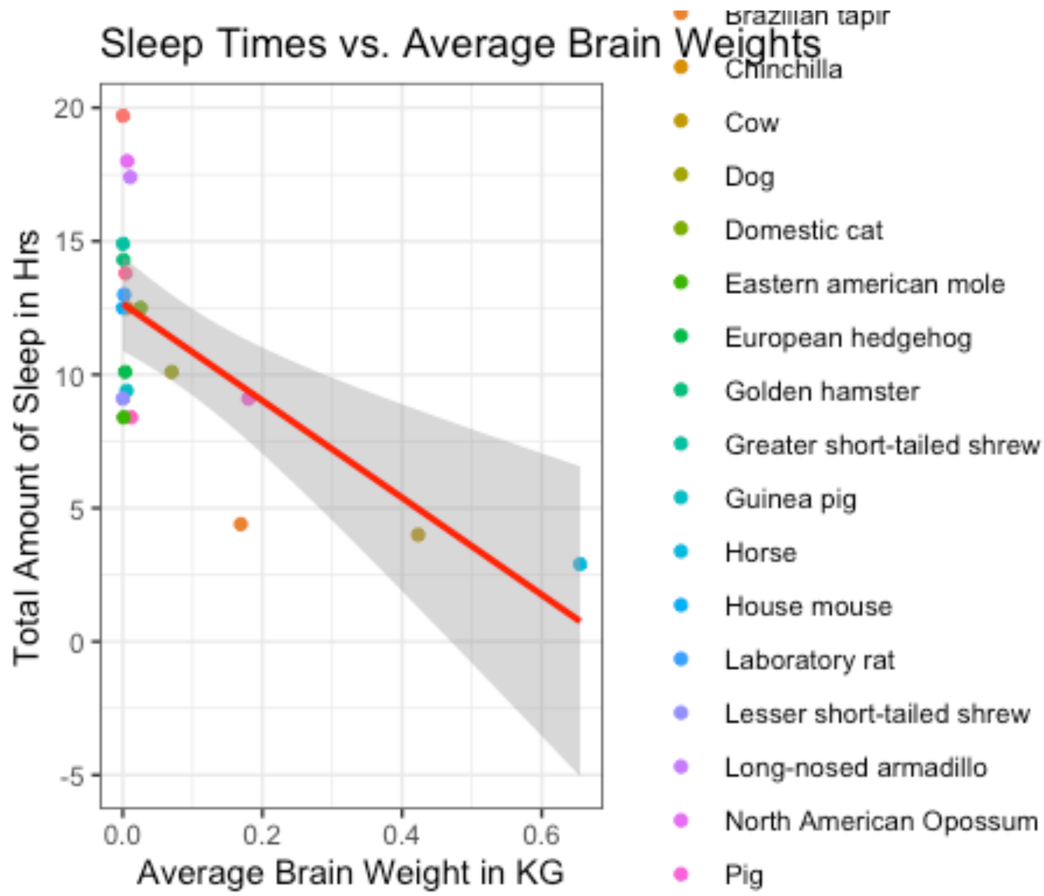
Part 2:

Problem - 1

```
library(ggplot2)
head(msleep)

## # A tibble: 6 × 11
##   name    genus vore  order conservation sleep_total sleep_rem sleep_cycle
awake
##   <chr>   <chr> <chr> <chr> <chr>              <dbl>     <dbl>       <dbl>
<dbl>
## 1 Cheetah Acin… carni Carn… lc                  12.1      NA        NA
11.9
## 2 Owl mo… Aotus omni  Prim… <NA>                17         1.8      NA
7
## 3 Mounta… Aplo… herbi Rode… nt                  14.4       2.4      NA
9.6
## 4 Greate… Blar… omni  Sori… lc                  14.9       2.3       0.133
9.1
## 5 Cow     Bos   herbi Arti… domesticated         4         0.7       0.667
20
## 6 Three-… Brad… herbi Pilo… <NA>                14.4       2.2       0.767
9.6
## # … with 2 more variables: brainwt <dbl>, bodywt <dbl>
```

(1)

```
df <- na.omit(msleep)
lmod = lm(sleep_total ~ brainwt, data = df)

ggplot(df, aes(brainwt, sleep_total, color = name)) +
  geom_point() +
  geom_smooth(method = lm, color = "red") +
  theme_bw() + xlab("Average Brain Weight in KG") +   ylab("Total Amount of
Sleep in Hrs") +
  ggtitle("Sleep Times vs. Average Brain Weights")

## `geom_smooth()` using formula 'y ~ x'
```

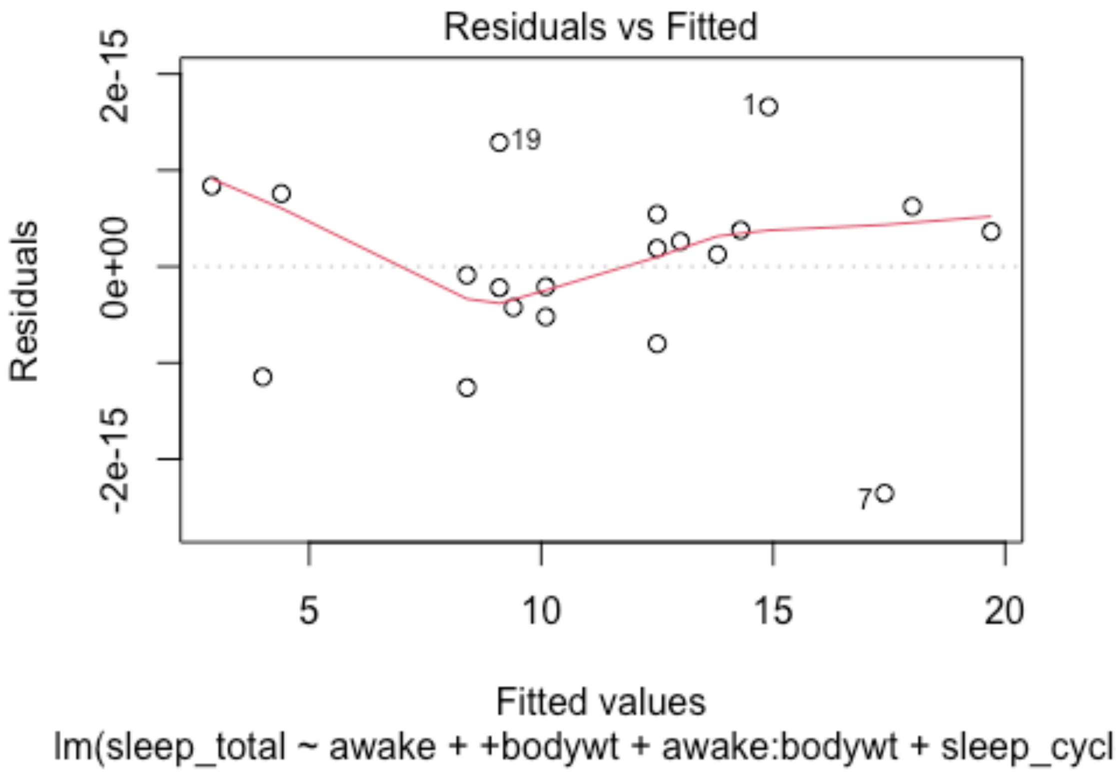


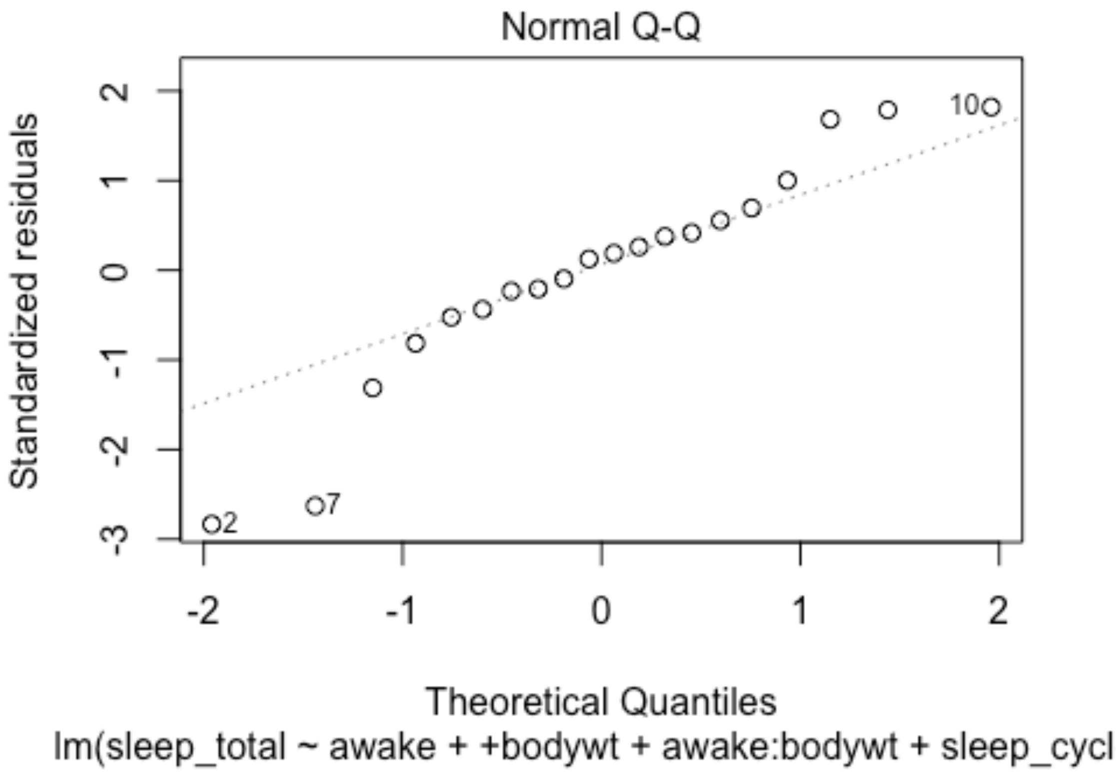
#####

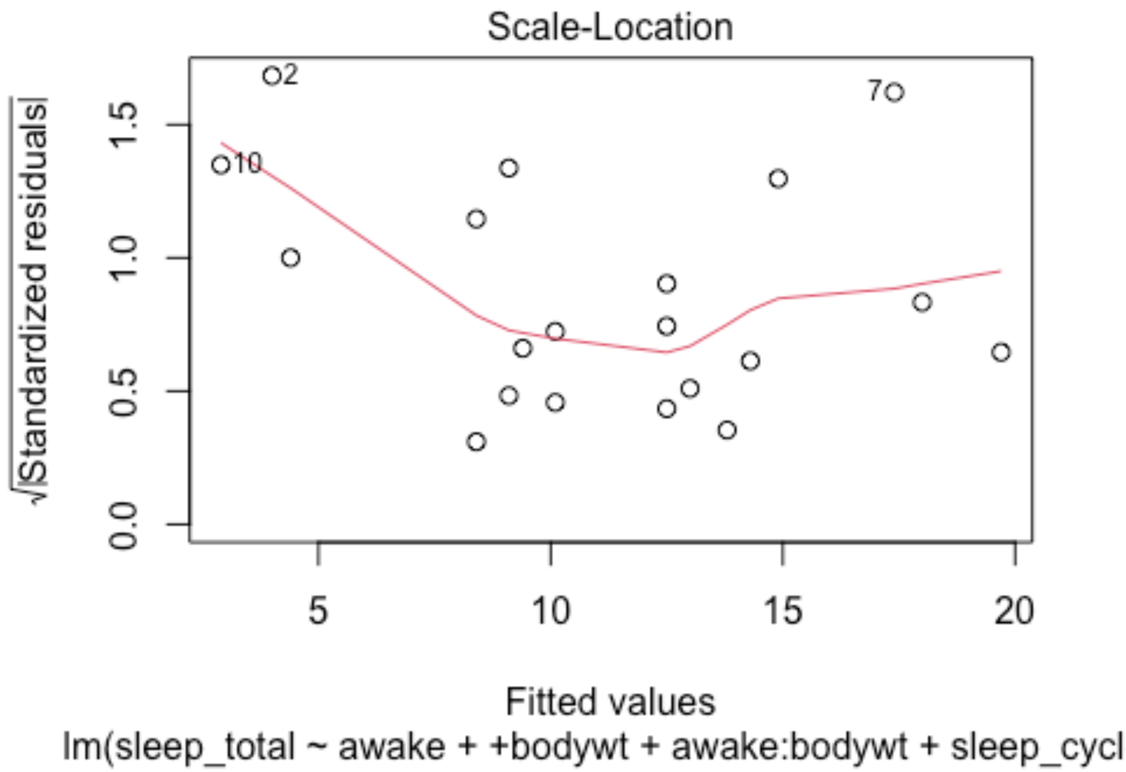
(2)

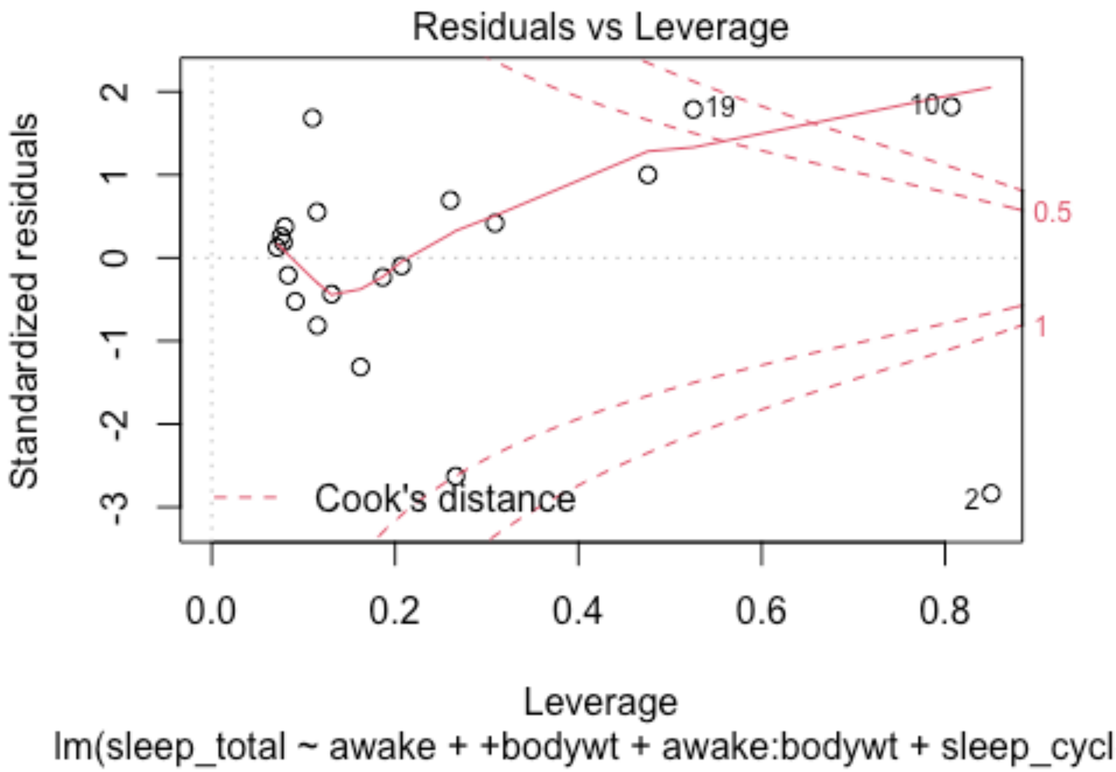
```
lmod = lm(sleep_total ~ awake + + bodywt + awake:bodywt + sleep_cycle, data =
df)
summary(lmod)

## Warning in summary.lm(lmod): essentially perfect fit: summary may be
unreliable

##
## Call:
## lm(formula = sleep_total ~ awake + +bodywt + awake:bodywt + sleep_cycle,
##     data = df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
```

```
## -2.356e-15 -4.496e-16  1.575e-16  5.642e-16  1.660e-15
##
## Coefficients:
##                Estimate Std. Error    t value Pr(>|t|)
## (Intercept)   2.400e+01  8.801e-16  2.727e+16  < 2e-16 ***
## awake        -1.000e+00  7.642e-17 -1.309e+16  < 2e-16 ***
## bodywt       -1.090e-16  3.313e-17 -3.289e+00  0.00497 **
## sleep_cycle   5.005e-15  1.652e-15  3.029e+00  0.00846 **
## awake:bodywt  4.777e-18  1.608e-18  2.970e+00  0.00953 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.046e-15 on 15 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 8.968e+31 on 4 and 15 DF,  p-value: < 2.2e-16

plot(lmod)
```


Residuals vs Fitted
Residuals
2e-15
0e+00
-2e-15
5
10
15
20
19
1
7
Fitted values
lm(sleep_total ~ awake + +bodywt + awake:bodywt + sleep_cycl


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(sleep_total ~ awake + +bodywt + awake:bodywt + sleep_cycl
2
1
0
-1
-2
-3
-2
-1
0
1
2
10
7
2


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
5
10
15
20
2
10
7
Fitted values
lm(sleep_total ~ awake + +bodywt + awake:bodywt + sleep_cycl



(3)

```
df$sleep_ratio <- (df$sleep_total / 24)
lmod = lm(sleep_total ~ sleep_ratio, data = df)
coef(lmod)

##   (Intercept)   sleep_ratio
## -7.944109e-15  2.400000e+01

logitSR = (1 / (1 + exp(-1 * (-7.944109e-15 + (2.400e+01 *
df$sleep_ratio))))); logitSR

##  [1] 0.9999997 0.9820138 0.9999589 0.9999173 0.9999963 0.9998883 1.0000000
##  [8] 1.0000000 1.0000000 0.9478464 0.9999589 0.9999963 0.9999994 0.9999963
## [15] 0.9997752 0.9999977 0.9997752 0.9999990 0.9998883 0.9878716
```

```
lmod = lm(logitSR ~ log(brainwt), data = df)
summary(lmod)
```

```
##
## Call:
## lm(formula = logitSR ~ log(brainwt), data = df)
##
## Residuals:
##       Min         1Q    Median        3Q       Max
## -0.034601 -0.002356  0.001764  0.004298  0.013635
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9812003  0.0051040 192.243  < 2e-16 ***
## log(brainwt) -0.0029465  0.0009222  -3.195  0.00502 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01004 on 18 degrees of freedom
## Multiple R-squared:  0.3619, Adjusted R-squared:  0.3264
## F-statistic: 10.21 on 1 and 18 DF,  p-value: 0.005018
```

Interpreting the results of our coefficients, we can see a negative correlation between the brain weight of the animal and the amount of sleep. For every additional KG of weight, there is a decrease of approximately 0.3% of sleep time needed per 24 hours.

```
df$sleep_ratio <- (df$sleep_ratio * 24)
lmod = lm(sleep_total ~ sleep_ratio, data = df)
coef(lmod)
```

```
##  (Intercept)  sleep_ratio
## 3.177644e-15 1.000000e+00
```

```
logitSR = (1 / (1 + exp(-1 * (3.177644e-15 + (1.000000e+00 *
df$sleep_ratio))))); logitSR
```

```
##  [1] 0.9999997 0.9820138 0.9999589 0.9999173 0.9999963 0.9998883 1.0000000
##  [8] 1.0000000 1.0000000 0.9478464 0.9999589 0.9999963 0.9999994 0.9999963
## [15] 0.9997752 0.9999977 0.9997752 0.9999990 0.9998883 0.9878716
```

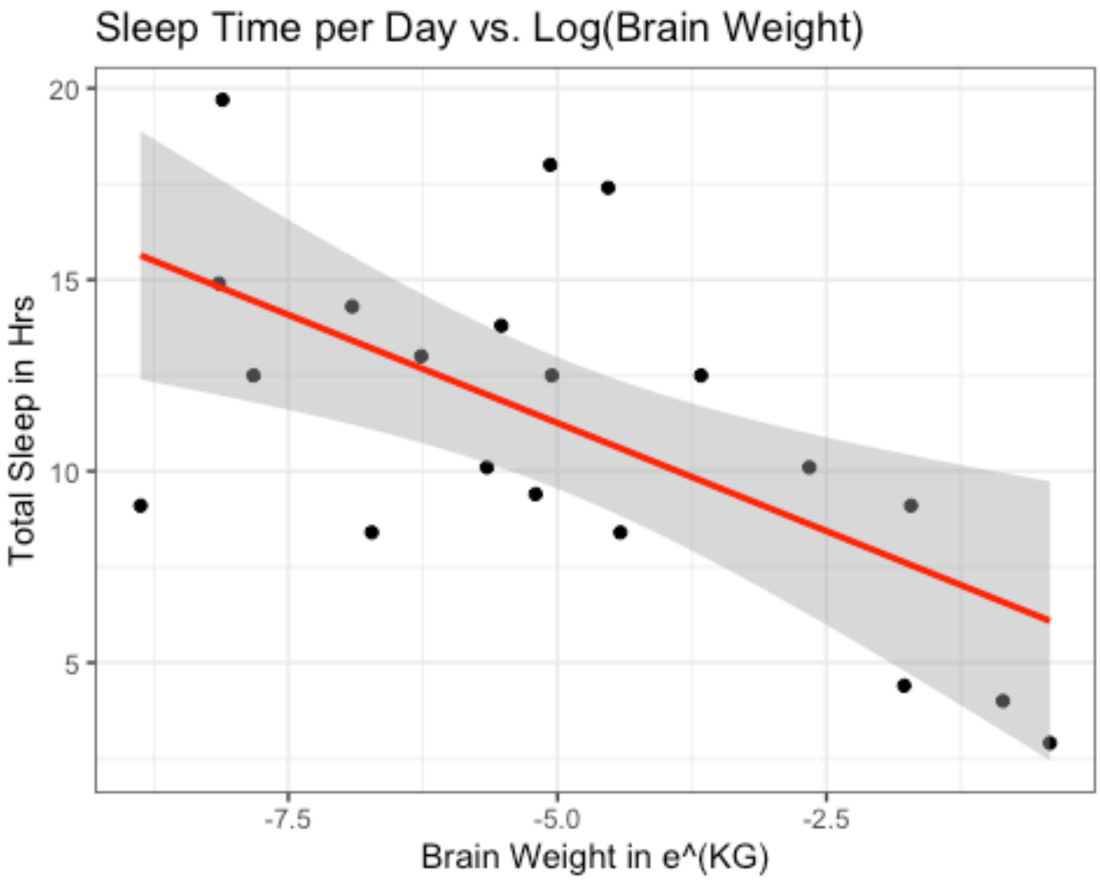
```
lmod = lm(logitSR ~ log(brainwt), data = df)
summary(lmod)

##
## Call:
## lm(formula = logitSR ~ log(brainwt), data = df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.034601 -0.002356  0.001764  0.004298  0.013635
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9812003  0.0051040 192.243  < 2e-16 ***
## log(brainwt) -0.0029465  0.0009222  -3.195  0.00502 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01004 on 18 degrees of freedom
## Multiple R-squared:  0.3619, Adjusted R-squared:  0.3264
## F-statistic: 10.21 on 1 and 18 DF,  p-value: 0.005018
```

We can see that returning from 24 hors to hours do not change our results and mantain the sleep time between 0 and 24, though never reaching 0 or 24.

(4)

```
ggplot(df, aes(log(df$brainwt), df$sleep_ratio)) +
  geom_point() +
  geom_smooth(method = lm, color = "red") +
  theme_bw() + xlab("Brain Weight in e^(KG)") +   ylab("Total Sleep in Hrs")
+
  ggtitle("Sleep Time per Day vs. Log(Brain Weight)")

## Warning: Use of `df$brainwt` is discouraged. Use `brainwt` instead.

## Warning: Use of `df$sleep_ratio` is discouraged. Use `sleep_ratio`
instead.

## Warning: Use of `df$brainwt` is discouraged. Use `brainwt` instead.
```

```
## Warning: Use of `df$sleep_ratio` is discouraged. Use `sleep_ratio`
instead.

## `geom_smooth()` using formula 'y ~ x'
```



Problem - 2

```
homeheat = read.csv('/Users/Home/Documents/Michael_Ghattas/School/CU_Boulder/
2022/Spring 2022/STAT - 4400/Data/homeheat.csv')
head(homeheat)

##   idcase depvar  ic.gc  ic.gr  ic.ec  ic.er   ic.hp  oc.gc  oc.gr  oc.ec
oc.er
## 1      1     gc 866.00 962.64 859.90 995.76 1135.50 199.69 151.72 553.34
505.60
## 2      2     gc 727.93 758.89 796.82 894.69  968.90 168.66 168.66 520.24
486.49
```

```
## 3      3     gc 599.48 783.05 719.86 900.11 1048.30 165.58 137.80 439.06
404.74
## 4      4     er 835.17 793.06 761.25 831.04 1048.70 180.88 147.14 483.00
425.22
## 5      5     er 755.59 846.29 858.86 985.64  883.05 174.91 138.90 404.41
389.52
## 6      6     gc 666.11 841.71 693.74 862.56  859.18 135.67 140.97 398.22
371.04
##    oc.hp income agehed rooms region
## 1 237.88      7     25     6 ncostl
## 2 199.19      5     60     5 scostl
## 3 171.47      4     65     2 ncostl
## 4 222.95      2     50     4 scostl
## 5 178.49      2     25     6 valley
## 6 209.27      6     65     7 scostl
```

(1)

```
library("mlogit")

## Loading required package: dfidx

##
## Attaching package: 'dfidx'

## The following object is masked from 'package:stats':
##
##     filter

H <- dfidx(homeheat, choice = "depvar", varying = c(3:12))
m <- mlogit(depvar ~ ic + oc | 0, H)
summary(m)

##
## Call:
## mlogit(formula = depvar ~ ic + oc | 0, data = H, method = "nr")
##
## Frequencies of alternatives:choice
##       ec       er       gc       gr       hp
## 0.071111 0.093333 0.636667 0.143333 0.055556
```

```
##
## nr method
## 4 iterations, 0h:0m:0s
## g'(-H)^-1g = 1.56E-07
## gradient close to zero
##
## Coefficients :
##       Estimate  Std. Error z-value  Pr(>|z|)
## ic -0.00623187  0.00035277 -17.665 < 2.2e-16 ***
## oc -0.00458008  0.00032216 -14.217 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -1095.2
```

Yes, the t-statistics are greater than 1.96, which is the critical level for 95% confidence level.

(2)

```
coef(m)["oc"]/coef(m)["ic"]

##        oc
## 0.7349453
```

The model implies that the decision-maker is willing to pay 73 cents in higher installation cost in order to reduce annual operating costs by $1.

(3)

```
mc <- mlogit(depvar ~ ic + oc, H, reflevel = 'hp')
summary(mc)

##
## Call:
## mlogit(formula = depvar ~ ic + oc, data = H, reflevel = "hp",
##     method = "nr")
##
## Frequencies of alternatives:choice
##       hp       ec       er       gc       gr
## 0.055556 0.071111 0.093333 0.636667 0.143333
##
```

```
## nr method
## 6 iterations, 0h:0m:0s
## g'(-H)^-1g = 9.58E-06
## successive function values within tolerance limits
##
## Coefficients :
##                    Estimate  Std. Error z-value  Pr(>|z|)
## (Intercept):ec  1.65884594  0.44841936  3.6993 0.0002162 ***
## (Intercept):er  1.85343697  0.36195509  5.1206 3.045e-07 ***
## (Intercept):gc  1.71097930  0.22674214  7.5459 4.485e-14 ***
## (Intercept):gr  0.30826328  0.20659222  1.4921 0.1356640
## ic             -0.00153315  0.00062086 -2.4694 0.0135333 *
## oc             -0.00699637  0.00155408 -4.5019 6.734e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -1008.2
## McFadden R^2:  0.013691
## Likelihood ratio test : chisq = 27.99 (p.value = 8.3572e-07)
```

```
apply(fitted(mc, outcome = FALSE), 2, mean)
```

```
##         hp         ec         er         gc         gr
## 0.05555556 0.07111111 0.09333333 0.63666667 0.14333333
```

Exact match: alternative-specific constants in a logit model insure that the average probabilities equal the observed shares.

(4)

```
wtp <- coef(mc)["oc"] / coef(mc)["ic"]
wtp
```

```
##       oc
## 4.563385
```

```
r <- 1 / wtp
r
```

```
##        oc
## 0.2191356
```

The willingness to pay is USD(4.56) for a $1 year stream of savings. The decision-maker applies a 22% discount rate, thus the results are certainly more reasonable than in the previous model.

(5)

```
Hn <- H
Hn[idx(Hn, 2) == "hp", "ic"] <- 0.88 * Hn[idx(Hn, 2) == "hp", "ic"]
apply(predict(mc, newdata = Hn), 2, mean)
```

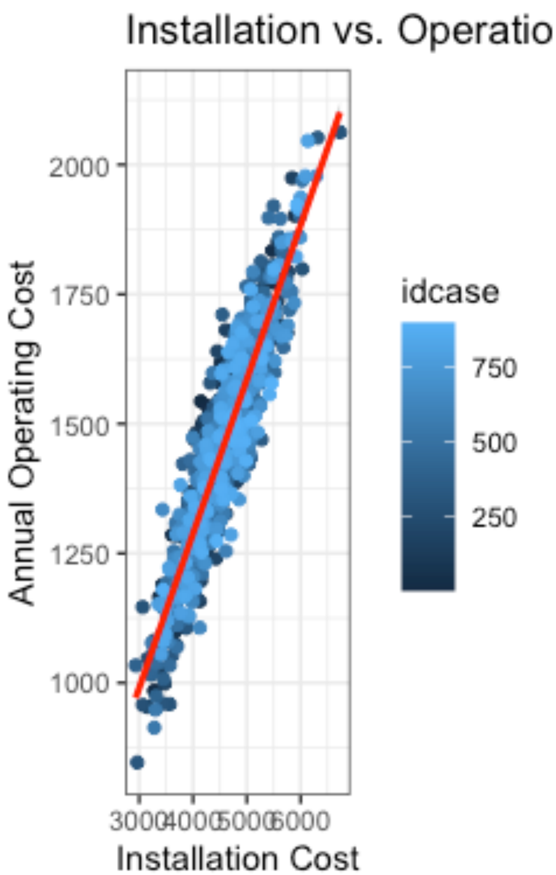
```
##         hp         ec         er         gc         gr
## 0.06640050 0.07031239 0.09228286 0.62933328 0.14167096
```

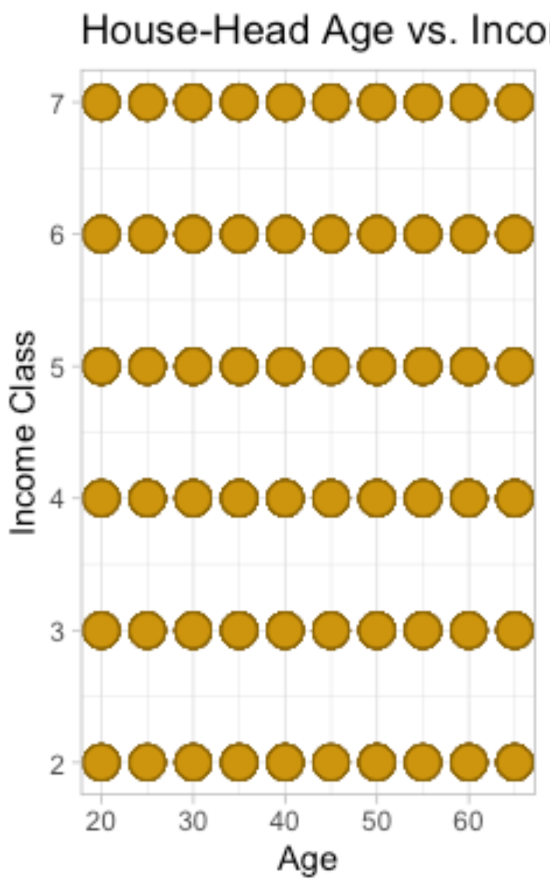
The share is predicted to rise to about 6.64% when rebates are given.

(6)

```
plot1 = ggplot(homeheat, aes(ic.gc + ic.gr + ic.ec + ic.er + ic.hp, oc.gc  +
oc.gr  + oc.ec  + oc.er + oc.hp, color = idcase)) +
  geom_point() +
  geom_smooth(method = lm, color = "red") +
  theme_bw() + xlab("Installation Cost") +   ylab("Annual Operating Cost") +
  ggtitle("Installation vs. Operational Cost")

plot2 = ggplot(homeheat, aes(agehed, income)) +
  geom_point(shape = 21, color = "darkgoldenrod4", fill = "darkgoldenrod3",
size = 5) +
  theme_light() + xlab("Age") +   ylab("Income Class") +
  ggtitle("House-Head Age vs. Income Class")

library(gridExtra)
grid.arrange(plot1, plot2, ncol = 2)
```

```
## `geom_smooth()` using formula 'y ~ x'
```


Installation vs. Operatio
Annual Operating Cost
2000
1750
1500
1250
1000
3000
4000
5000
6000
Installation Cost
idcase
750
500
250


House-Head Age vs. Incor
Income Class
7
6
5
4
3
2
20
30
40
50
60
Age