Intro to Heart Rate

Corey Kimzey

2017-12-11

Description

This package is used to do analysis on some resting heart rate data I have been collecting. It creates target heart ranges for excersing. Also it compares heart rate to it's moving average. The package also tries to fit some linear models. It can also create a scatter plot with the correlation listed as a title.

Objective

I was hoping with this analysis to see if there was a relationship between my resting heart rate. I also want to look at some patterns with my resting heart rate. #Experimental Design So I measured 10 Variable. 1) Date is just the date the rest of the variables were measured. 2) Day is just the number of days from the first day. 3) Resting Heart Rate was measured every morning (that I remembered) by counting my heart rate for 1 min. The first day I tried to use a resting heart rate monitor but it didn't seem to be working. 4) Miles Walking was the number of miles that my health app on my phone collected each day. 5) Inf is the number of hours that I studied for my Statistical Inference class. 6) Computing is the number of hours that I studied for Introduction to Statistical Computing. 7) Probability is the numbe of hours that I studied for Introduction to Probability. 8) Other is any other hours of studying that I wasn't for those classes. 9) Estimated Sleep is the amount of sleep that I had the night before I measured my heart rate. 10) Hrs on Hwk is just the sumation of 5), 6), 7), and 8).

My dataset is really only representative of heart rate data for myself. It would be interesting to extend this to other people and see if others get similar results to me. There are many possible confounders. For instance, this experiment is not blinded. I was the one counting my own heart rate and I started anaylsis before I finished collecting data. Therefore, this could have affected the results of my study. Also there were days that I forgot to count my resting heart rate. Therefore, these missing data could have affected my results as well.

```
library(knitr)
library(heartrate)
library(googlesheets)
library(tidyverse)
library(forecast)
```

Here are some examples of what you can do with my packages and some basic analysis:

Sample Dataset

I created a sample dataset. It contains all of my data collected. Here is just the first couple of rows:

```
data("tester")
tester
```

```
## # A tibble: 87 x 10
##          Date   Day `Resting Heart Rate` `Miles walking` `Inf` Computing
##        <date> <int>                <int>           <dbl> <dbl>     <dbl>
##  1 2017-09-12     1                   41            3.30     4         1
##  2 2017-09-13     2                     60             1.80     4          2
##  3 2017-09-14     3                     76             1.90     2          2
##  4 2017-09-15     4                     81             3.00     1          2
##  5 2017-09-16     5                     72             0.55     2          1
##  6 2017-09-17     6                     78             0.96     3          1
##  7 2017-09-18     7                     68             1.80     4          0
##  8 2017-09-19     8                     75             1.80     0          1
##  9 2017-09-20     9                     NA             1.90     0          0
## 10 2017-09-21    10                     69             2.60     0          2
## # ... with 77 more rows, and 4 more variables: Probability <dbl>,
## #   Other <int>, `Estimated Sleep` <dbl>, `Hrs on Hwk` <dbl>
```

Updating Dataset

I created a function that will automatically pull in the newest data. It is called updating_heartRate. This is useful so that I can continually use the function to update my data.

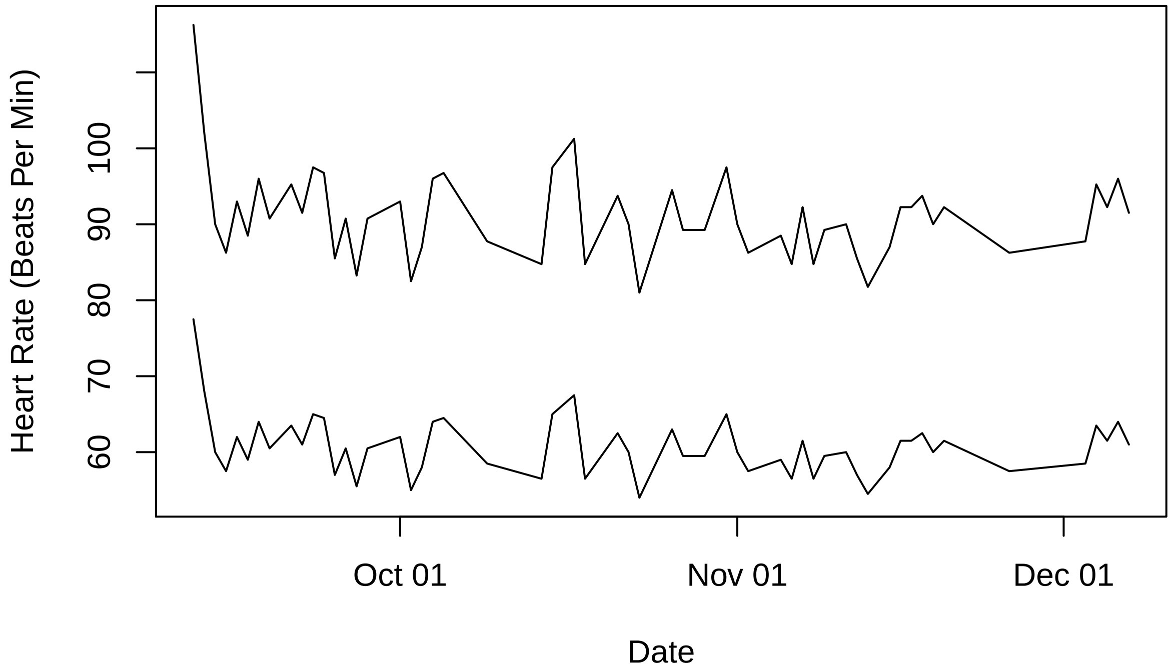
```
tester <- updating_heartRate()
tester
```

```
## # A tibble: 87 x 10
##          Date   Day `Resting Heart Rate` `Miles walking` `Inf` Computing
##        <date> <int>                <int>           <dbl> <dbl>     <dbl>
##  1 2017-09-12     1                     41             3.30     4          1
##  2 2017-09-13     2                     60             1.80     4          2
##  3 2017-09-14     3                     76             1.90     2          2
##  4 2017-09-15     4                     81             3.00     1          2
##  5 2017-09-16     5                     72             0.55     2          1
##  6 2017-09-17     6                     78             0.96     3          1
##  7 2017-09-18     7                     68             1.80     4          0
##  8 2017-09-19     8                     75             1.80     0          1
##  9 2017-09-20     9                     NA             1.90     0          0
## 10 2017-09-21    10                     69             2.60     0          2
## # ... with 77 more rows, and 4 more variables: Probability <dbl>,
## #   Other <int>, `Estimated Sleep` <dbl>, `Hrs on Hwk` <dbl>
```

target_ranges

I created a function that will create target work out ranges based off https://www.active.com/fitness/articles/how-to-calculate-your-training-heart-rate-zones. I think this could be useful if I tracked my heart rate during work outs I could see if my heart rates during the work out were in my target heart range. Here is an example of that:

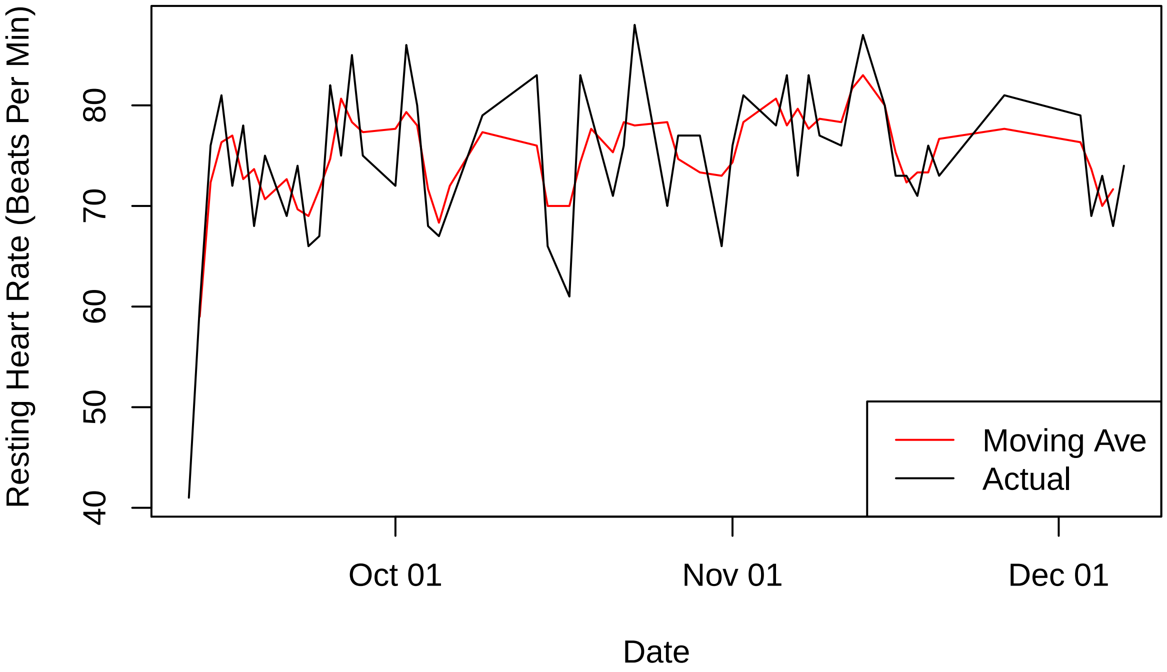
```
ranged <- target_ranges(tester$`Resting Heart Rate`, time_data = tester$Date,
                        age = 24)
plot(ranged[,1], ranged[,2], type = "l", ylim = c(min(ranged[,2]),
                                                   max(ranged[,3])),
     xlab = "Date",
     ylab = "Heart Rate (Beats Per Min)" )
lines(ranged[,1], ranged[,3], type = "l")
```



stress_plot

I created a function that imediately plots the difference between the actual data and a moving average. This would allow one to see if they seem to be above their moving average. My data for the most part seems to be pretty much the same (it might be increasing slightly) except the first data point was really low. The first data point is really low because I was using a heart rate monitor that I think is broken.

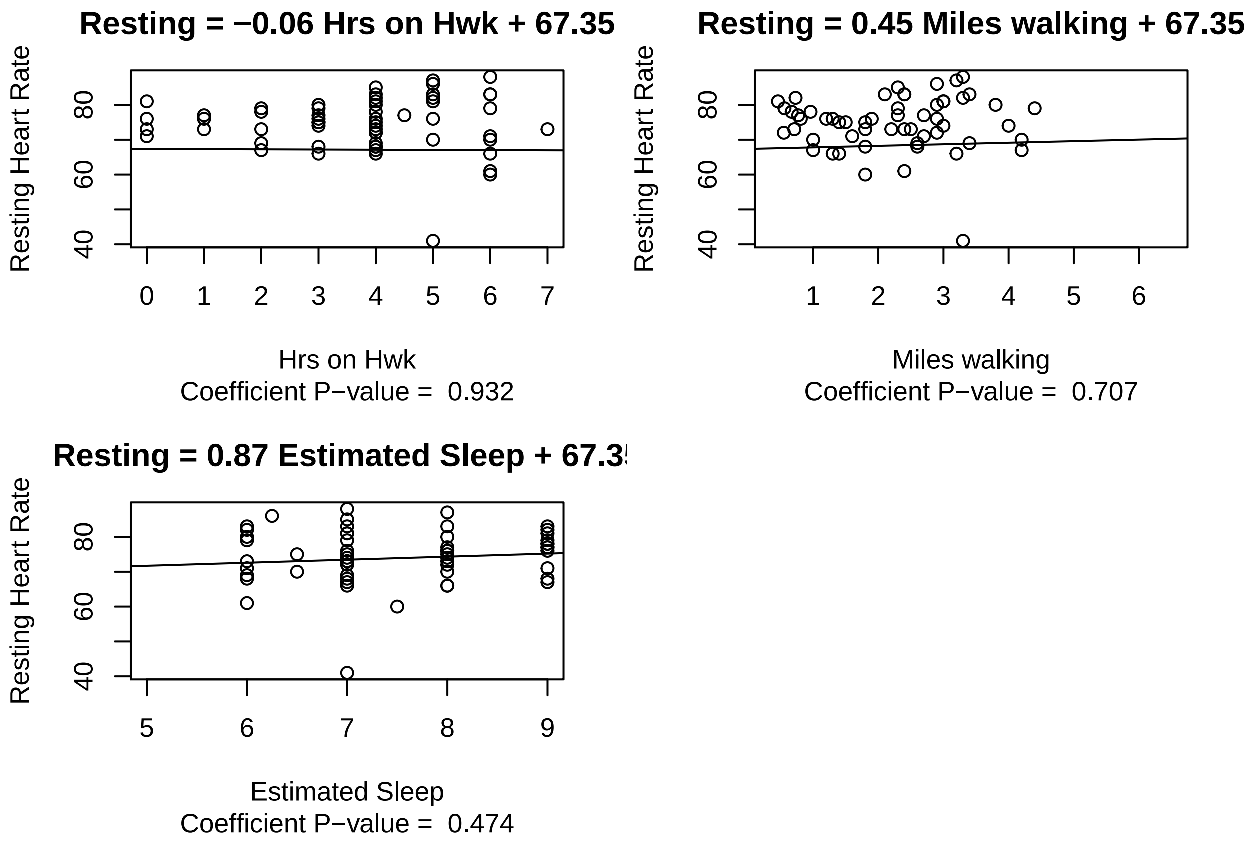
```
plot_stressed(tester$`Resting Heart Rate`, tester$Date, xlab = "Date",
              ylab = " Resting Heart Rate (Beats Per Min)")
```



plot_lms

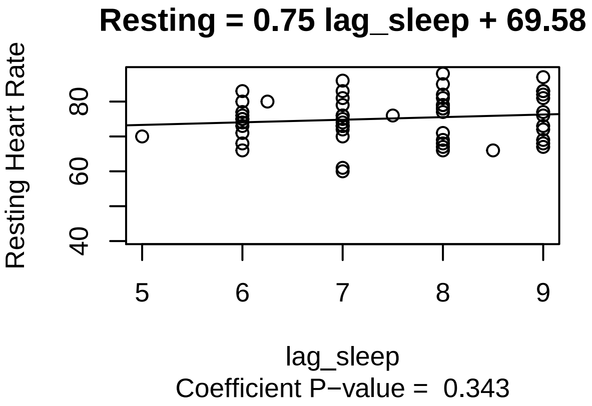
This function produces a series of scatterplots that match up with the columns selected. It then fits a linear regression and posts the equation for the line at the top of each scatterplot. I was also able to now add the p-value for the coefficient for the slope of the line to the plot at the bottom. I would like to add some way of doing multiple linear regression here and maybe have a way to post if the slope of the line is statistically significant. It doesn't seem that any of these linear regressions seem to fit the data very well. Also none of the p-values are significant.

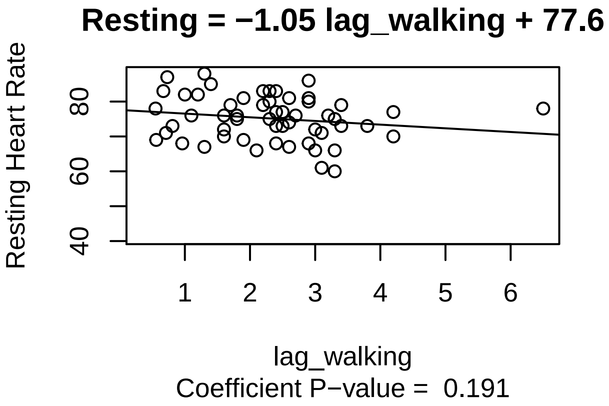
```
par(mar= c(5.0, 4.0, 3.0, 2))
tester %>% plot_lms(., col_names = c("Hrs on Hwk", "Miles walking", "Estimated Sleep"),
         response_variable = "Resting Heart Rate", multivariate = TRUE)
```

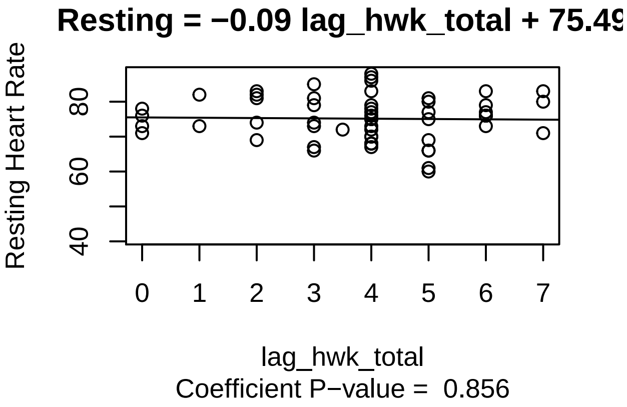


I then tried an compare the resting heart rate to the pervious day's data. Again, there doesn't seem to be any clear fit to the data. Also none of the p-values for the coefficient of the slope are statistically significiant either.

```
par(mar= c(5.0, 4.0, 3.0, 2))
tester <- tester %>% mutate(lag_sleep = lag(tester$"Estimated Sleep", 1),
                            lag_walking= lag(tester$"Miles walking", 1),
                            lag_hwk_total = lag(tester$"Hrs on Hwk", 1))
tester %>% plot_lms(., col_names = c("lag_sleep", "lag_walking", "lag_hwk_total"),
         response_variable = "Resting Heart Rate")
```



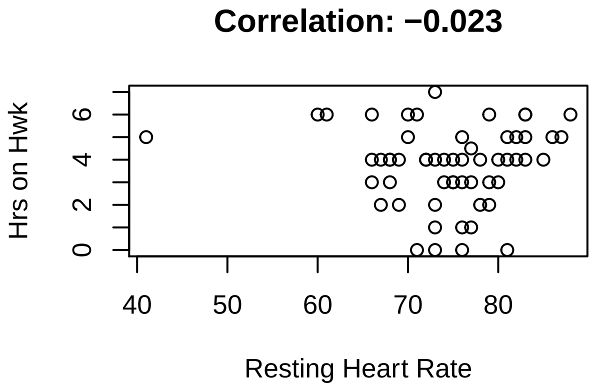


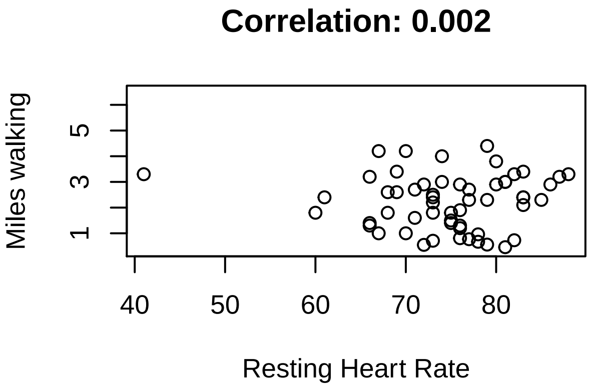


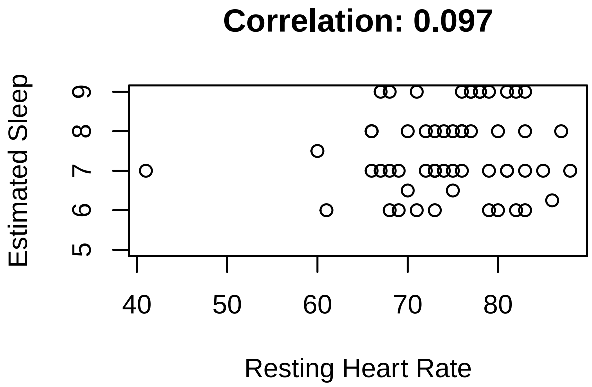
plot_cor

This function produces a series of scatterplots that match up with the columns selected but this time it prints the correlation of the variable of the columns selected with the response variable. As you can see for these three variables there isn't a strong correlation between them and resting heart rate.

```r
par(mar = c(4.1, 4.1,4.1,1.1))
tester %>% plot_cor(. ,col_names = c("Hrs on Hwk", "Miles walking", "Estimated Sleep"),
         response_variable = "Resting Heart Rate")
```



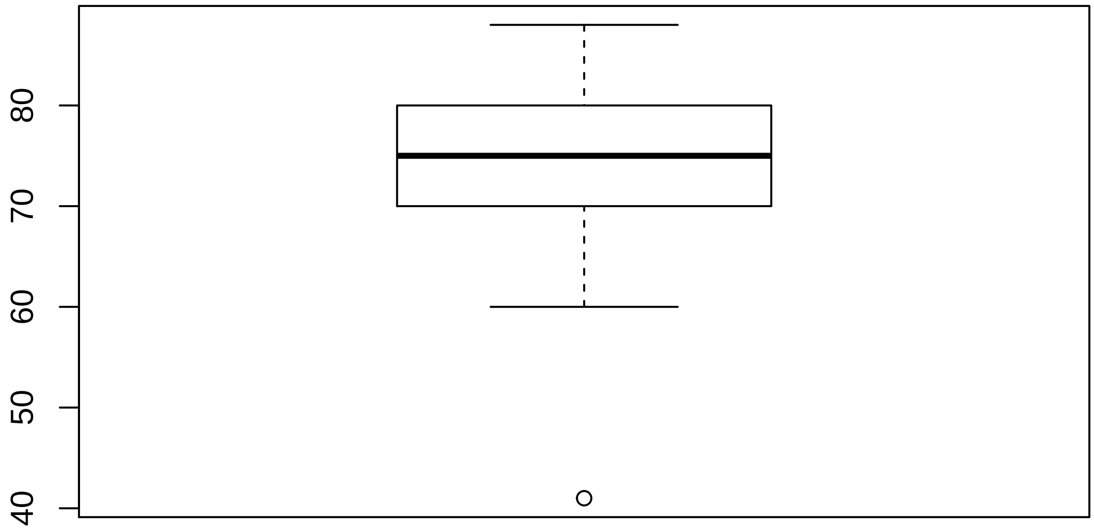




Other Analysis of the Resting Heart Rate Data.

The first point in the resting heart rate looks like it might be an outlier. Therefore, I thought about removing it. Also the first day I tried to use a heart rate monitor but I don't believe that it functions correctly.

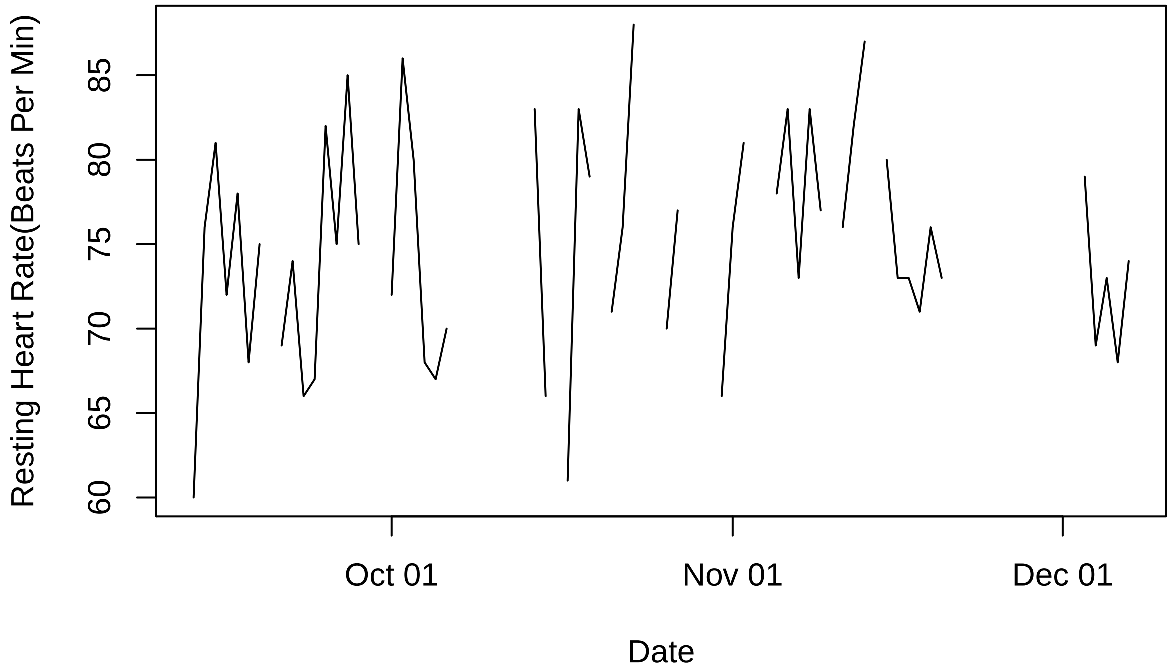
```
tester %>% select("Resting Heart Rate") %>%  boxplot()
```



```
tester2 <- tester[-1,]
```

This is just a plot of my resting heart rate. It looks like the plot might be cyclical. So I thought to compare across days of the week.

```
plot(tester2$Date, tester2$"Resting Heart Rate", type = "l",
     ylab = "Resting Heart Rate(Beats Per Min)", xlab = "Date")
```



Interestingly, it seems that my resting heart rate is high on Monday ,Friday, Sunday. I'm not sure why that would be. This would probably something I would continually look at.

```
days_of_week <- as.factor(c("T", "W", "Th", "F", "S", "Su", "M"))
days_of_week <- rep(days_of_week, 100)[1:nrow(tester2)]
tester2$days_of_week <- days_of_week



mean_ave <- by(tester2$`Resting Heart Rate`, tester2$days_of_week, function(x) mean(x, na.rm = TRUE))
mean_ave_df <- as.data.frame(do.call(rbind, list(mean_ave)))
kable(mean_ave_df)
```

F	M	S	Su	T	Th	W
73.16667	71.85714	75.63636	79.44444	75.375	75	74.1

Although, currently if we look at the the Tukey Multiple comparison test with a 95% family wise confidence level. We see that none of the of comparisons are statistically significant.

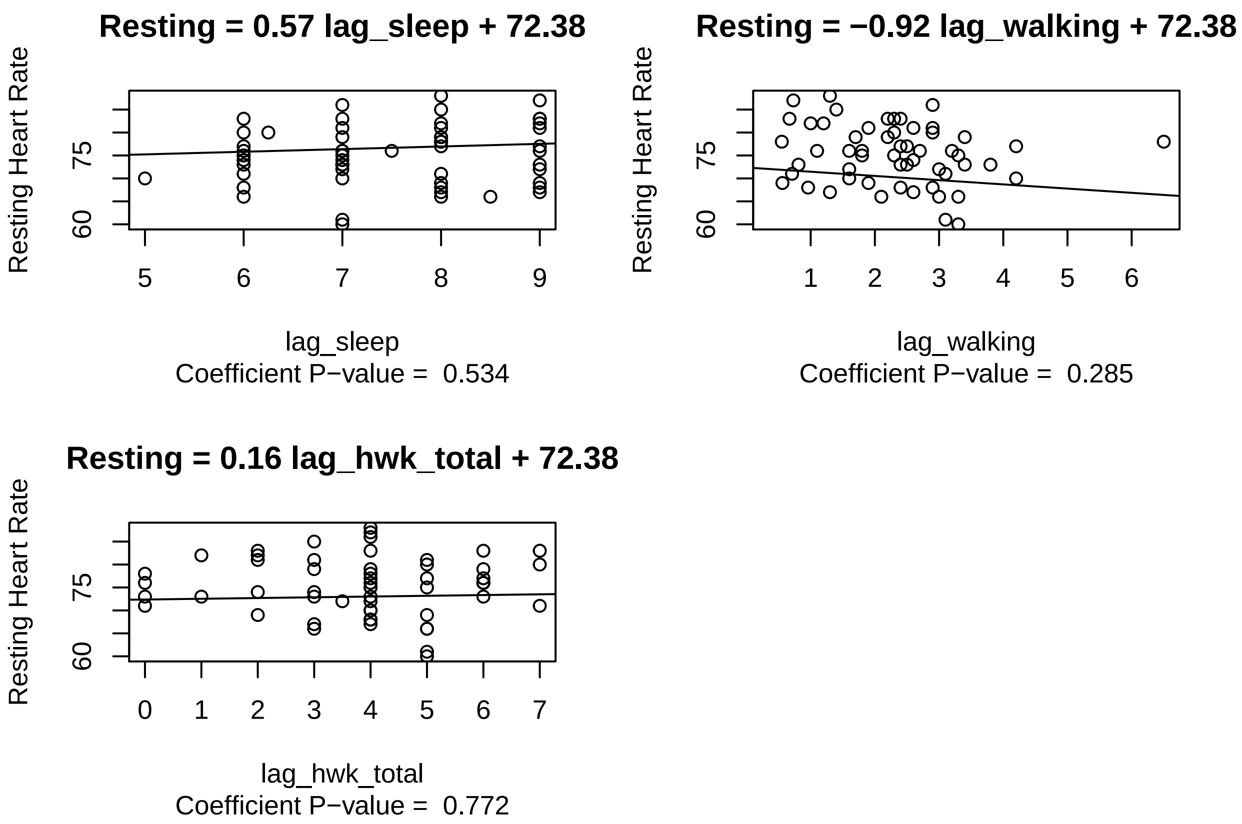
```
saver <- TukeyHSD(aov(tester2$`Resting Heart Rate` ~ tester2$days_of_week))
kable(saver$`tester2$days_of_week`)
```

	diff	lwr	upr	p adj
M-F	-1.3095238	-12.184431	9.565383	0.9997760
S-F	2.4696970	-7.450745	12.390139	0.9871940
Su-F	6.2777778	-4.024358	16.579913	0.5071627

	diff	lwr	upr	p adj
T-F	2.2083333	-8.348214	12.764881	0.9949275
Th-F	1.8333333	-10.002919	13.669586	0.9990477
W-F	0.9333333	-9.160657	11.027324	0.9999522
S-M	3.7792208	-5.671597	13.230039	0.8792667
Su-M	7.5873016	-2.263424	17.438027	0.2342507
T-M	3.5178571	-6.598639	13.634353	0.9341560
Th-M	3.1428571	-8.302652	14.588366	0.9788311
W-M	2.2428571	-7.389974	11.875689	0.9909895
Su-S	3.8080808	-4.977610	12.593772	0.8333634
T-S	-0.2613636	-9.344043	8.821316	1.0000000
Th-S	-0.6363636	-11.179193	9.906465	0.9999962
W-S	-1.5363636	-10.077031	7.004304	0.9977894
T-Su	-4.0694444	-13.567544	5.428655	0.8406920
Th-Su	-4.4444444	-15.347200	6.458311	0.8693653
W-Su	-5.3444444	-14.325638	3.636749	0.5354637
Th-T	-0.3750000	-11.518463	10.768463	0.9999999
W-T	-1.2750000	-10.546922	7.996922	0.9995176
W-Th	-0.9000000	-11.606293	9.806293	0.9999727

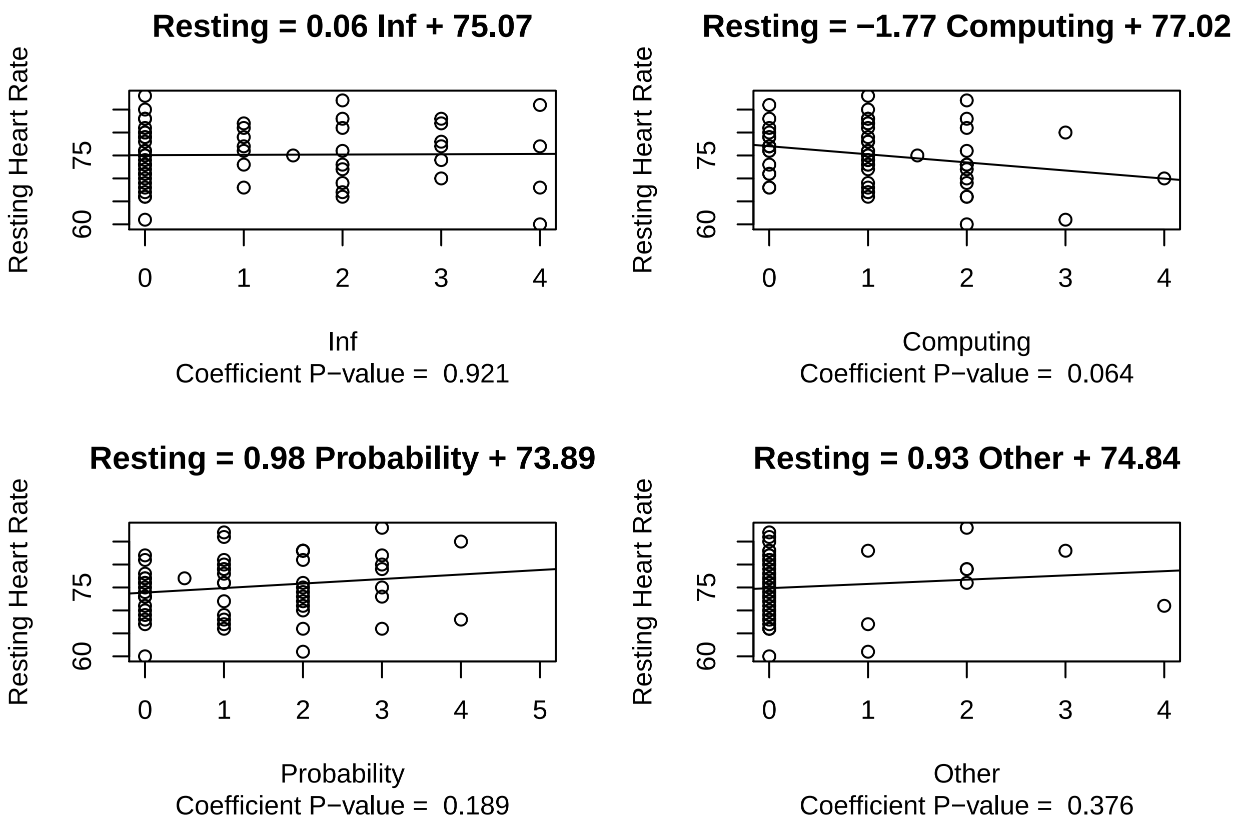
Now I though I'd look at some of the vaiables I created before but now with the dataset without the outlier and looking at the lag of sleep, miles walking, and homework totals. This time I'm going to look at them all together or in other words fit a multivariate regression model. Again, we see that none of these covariates are statistically significant.

```
tester2 %>% plot_lms(., col_names = c("lag_sleep", "lag_walking", "lag_hwk_total"),
                     response_variable = "Resting Heart Rate",
                     multivariate = TRUE)
```



Then I thought to just look at the indivdual classes and how much time I spend on those classes. Again I didn't see any statistically significant p-values for the correlation coefficient.

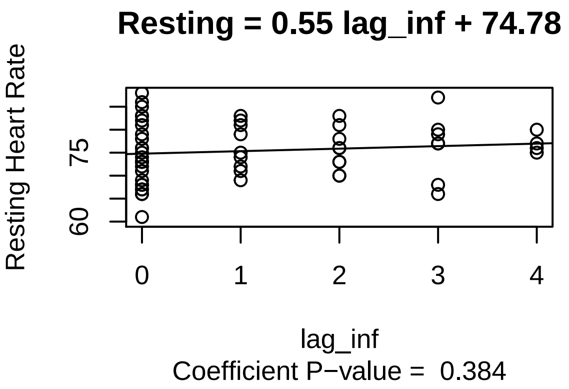
```
tester2 %>% plot_lms(., col_names = c("Inf", "Computing", "Probability", "Other"),
                        response_variable = "Resting Heart Rate")
```

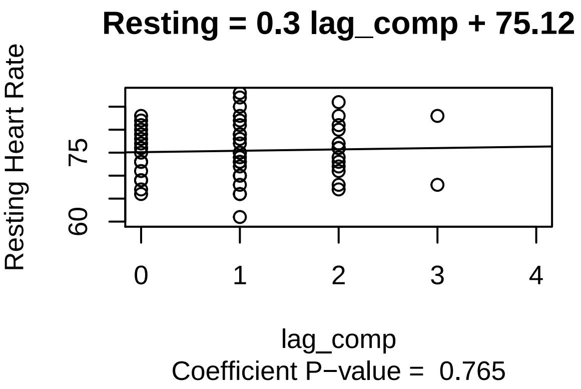


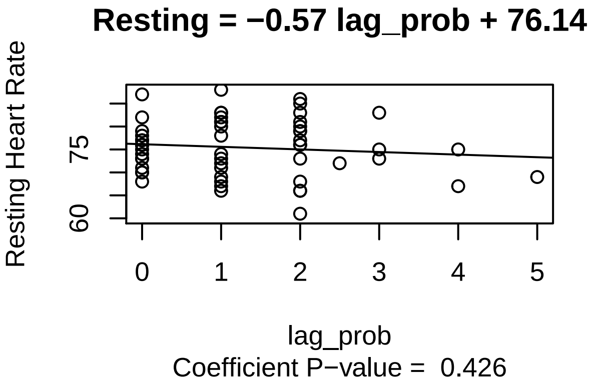
Then again, I thought maybe if I looked at the pervious days homework for each one of those classes I would see some evidence. These variables in general seemed to even have larger p-values and not to fit the data well.

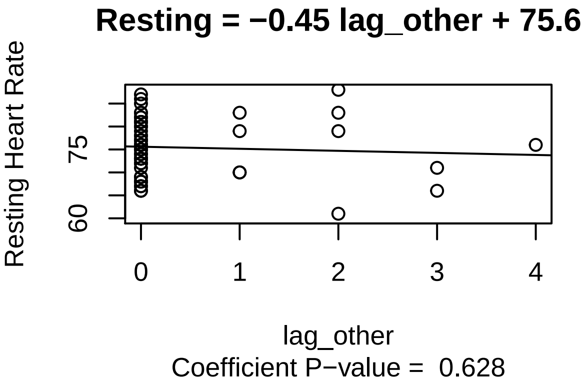
```
tester2 <- tester2 %>% mutate(lag_inf = lag(tester2$"Inf", 1),
                              lag_comp= lag(tester2$"Computing", 1),
                              lag_prob = lag(tester2$"Probability", 1),
                              lag_other = lag(tester2$Other))

tester2 %>% plot_lms(., col_names = c("lag_inf", "lag_comp", "lag_prob", "lag_other"),
                    response_variable = "Resting Heart Rate")
```









Conclusion

Also none of the variables seem to be highly correlated with my resting heart rate. Stress plot function also allows me to determine target heart rate. #Limitations There are many limitations to my analysis. For one instance, I only collected data on me, so it probably doesn't extend to everyone else. Also, I wasn't always great at keeping track of the variables I was interested in, so it probably isn't very accurate. I started off thinking I would use a heart rate monitor, but then it broke. So I had to keep track of it by counting the beats myself. This probably created some bias. Also I wish I knew more about time series analysis because that could be used in this setting. The length of the study was short. It would be intriguing to see how things were different if I did this for longer and added in more variables. Overall, I found this study really interesting and made me realize I should start tracking other things in my life.