

Intro to Heart Rate

Corey Kimzey

2017-10-17

Description

This package is used to do analysis on some resting heart rate data I have been collecting. It creates target heart ranges for exercising. Also it compares heart rate to it's moving average. The package also tries to fit some linear models.

```
library(knitr)
library(heartrate)
library(google sheets)
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 has the first 21 days of data that I collected. Here is just the first couple of rows:

```
data("ckheartRate")
head(ckheartRate)
```

```
##   Day Resting Heart Rate Time in Min Miles walking Inf Computing
## 1   1           41      NA      3.30  4           1
## 2   2           60      NA      1.80  4           2
## 3   3           76      NA      1.90  2           2
## 4   4           81      NA      3.00  1           2
## 5   5           72      NA      0.55  2           1
## 6   6           78      NA      0.96  3           1
##   Probability Estimated Sleep Hrs on Hwk
## 1           0           7.0      5
## 2           0           7.5      6
## 3           1           7.0      5
## 4           1           7.0      4
## 5           1           8.0      4
## 6           0           9.0      4
```

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.

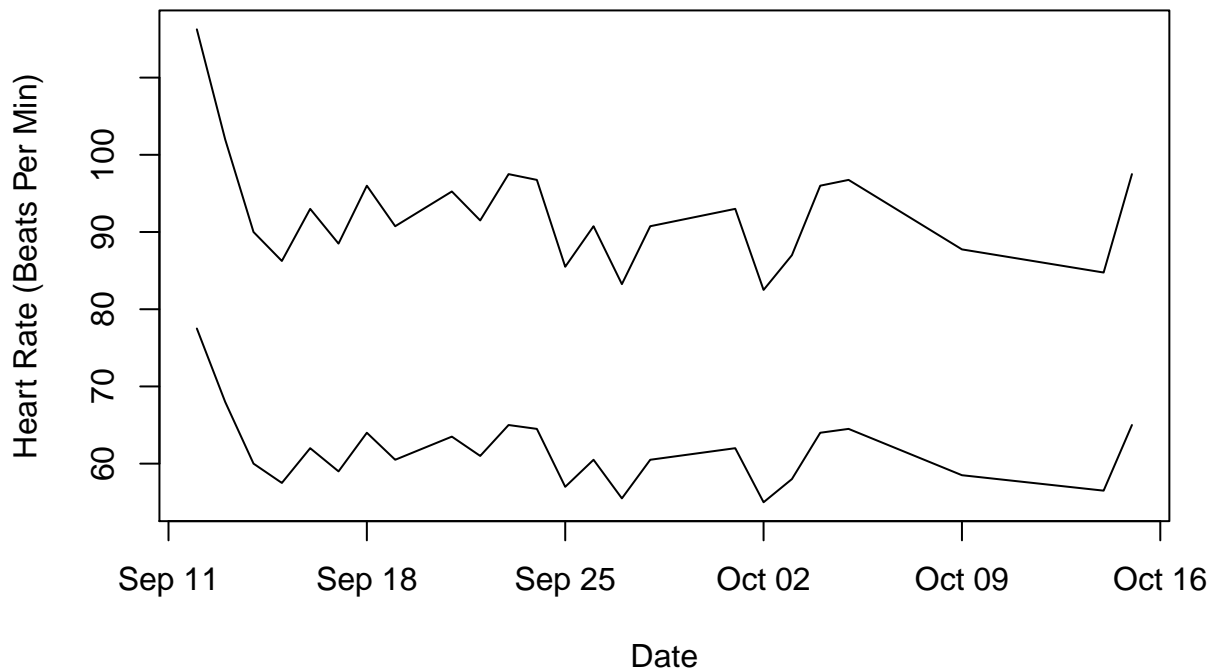
```
tester <- updating_heartRate()
tester
```

```
## # A tibble: 36 x 11
##       Date    Day `Resting Heart Rate` `Time in Min` `Miles walking`
##   <date> <int>         <int>         <lgl>         <dbl>
## 1 2017-09-12     1             41          NA          3.30
## 2 2017-09-13     2             60          NA          1.80
## 3 2017-09-14     3             76          NA          1.90
## 4 2017-09-15     4             81          NA          3.00
## 5 2017-09-16     5             72          NA          0.55
## 6 2017-09-17     6             78          NA          0.96
## 7 2017-09-18     7             68          NA          1.80
## 8 2017-09-19     8             75          NA          1.80
## 9 2017-09-20     9             NA          NA          1.90
##10 2017-09-21    10             69          NA          2.60
## # ... with 26 more rows, and 6 more variables: `Inf` <dbl>,
## #   Computing <dbl>, 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:

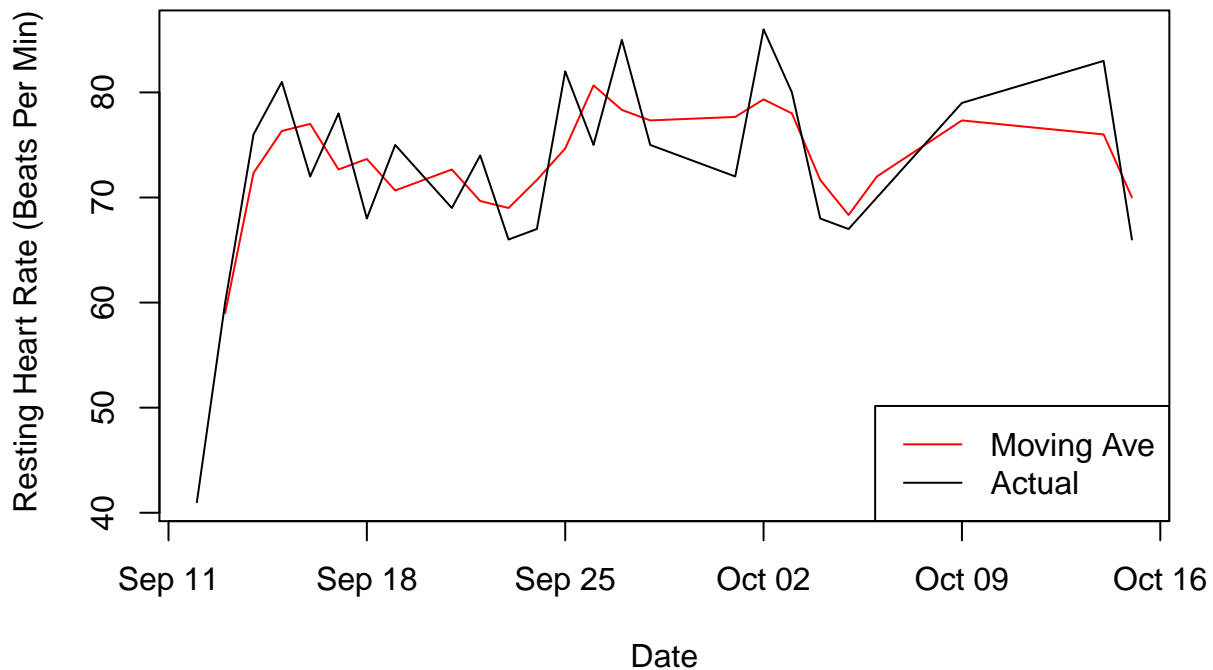
```
ranged <- target_ranges(tester$`Resting Heart Rate`, time_data = tester$Date,
                        age = 24)
plot(ranged[,1], ranged[,2], type = "l", ylim = c(min(ranged[,2]),
                                                  max(ranged[,3])),
      xlab = "Date",
      ylab = "Heart Rate (Beats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



stress_plot

I created a function that immediately 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.

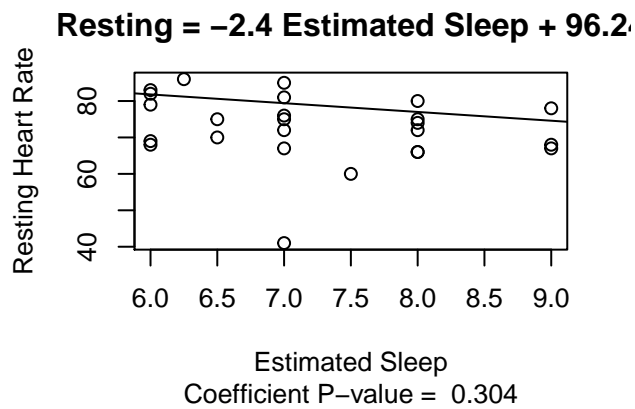
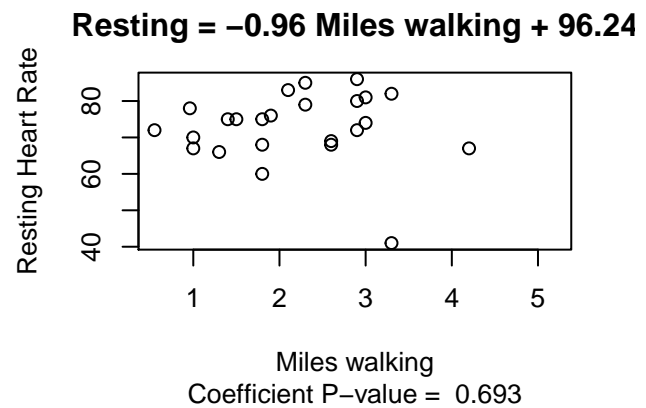
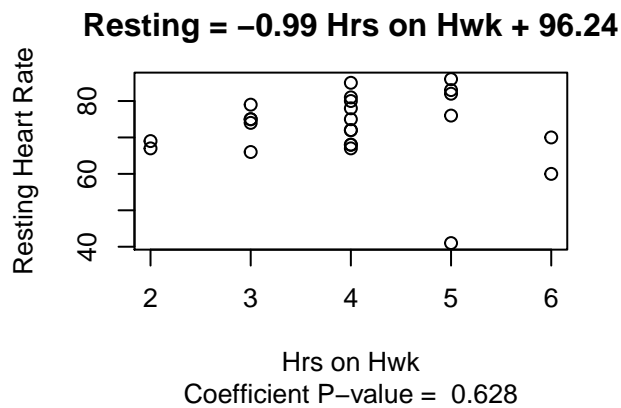
```
plot_stressed(tester$`Resting Heart Rate`, tester$Date, xlab = "Date",  
              ylab = " Resting Heart Rate (Beats Per Min)")
```



plot_lms

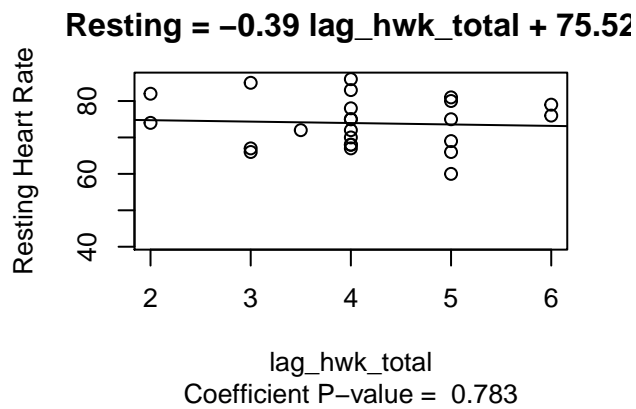
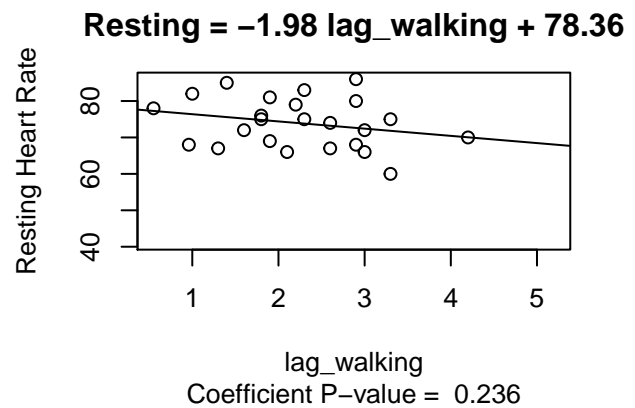
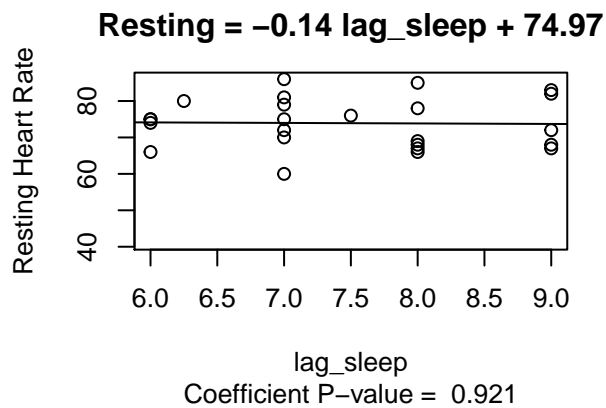
This function produces a series of scatterplots that match up with the columns selected. It then fits a linear regression and posts the equation for the line at the top of each scatterplot. I was also able to now add the p-value for the coefficient for the slope of the line to the plot at the bottom. 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 to compare the resting heart rate to the previous 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 significant either.

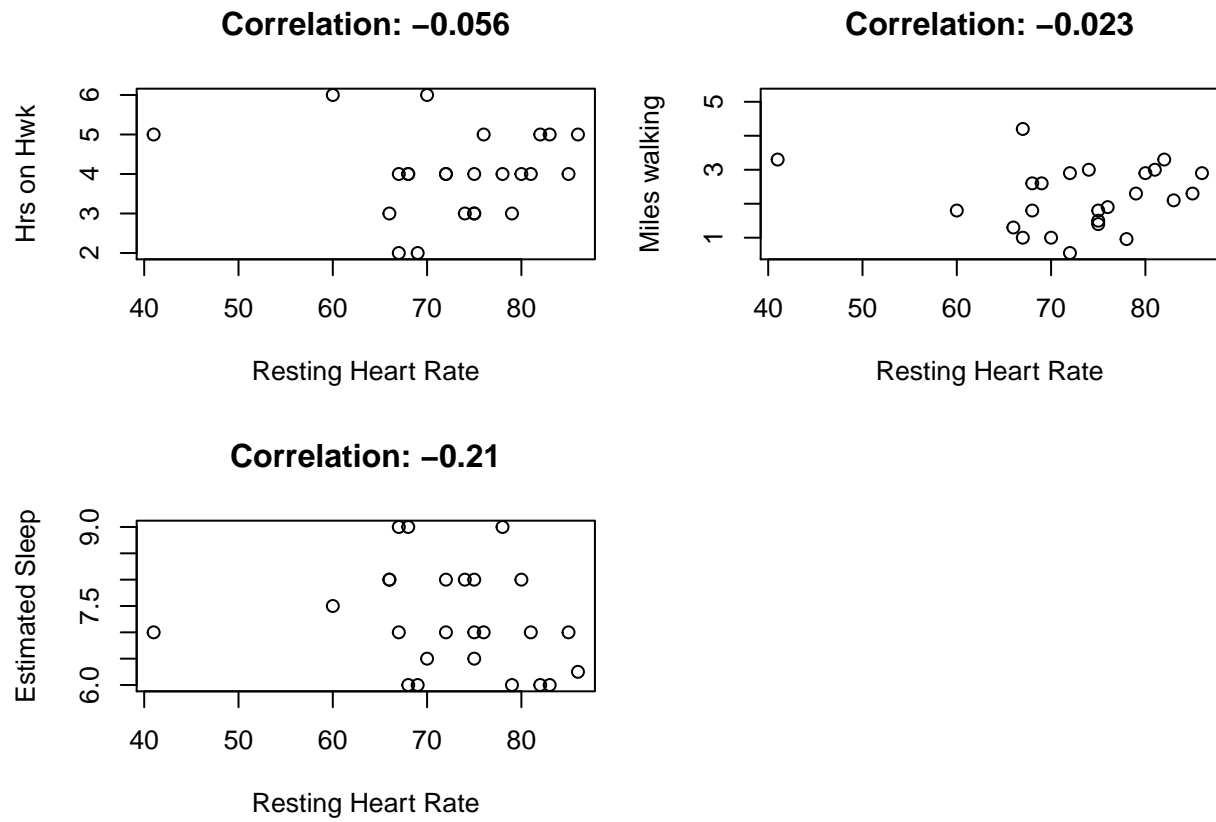
```
par(mar= c(5.0, 4.0, 3.0, 2))
tester <- tester %>% mutate(lag_sleep = lag(tester$"Estimated Sleep", 1),
                           lag_walking= lag(tester$"Miles walking", 1),
                           lag_hwk_total = lag(tester$"Hrs on Hwk", 1))
tester %>% plot_lms(., col_names = c("lag_sleep",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

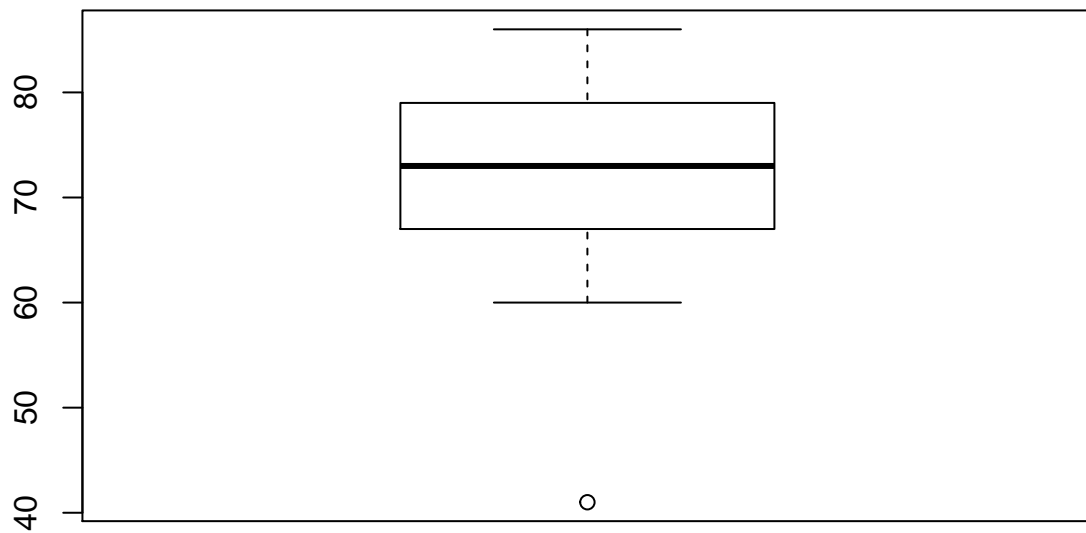
```
par(mar = c(4.1, 4.1, 4.1, 1.1))
tester %>% plot_cor(., col_names = c("Hrs on Hwk", "Miles walking", "Estimated Sleep"),
  response_variable = "Resting Heart Rate")
```



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.

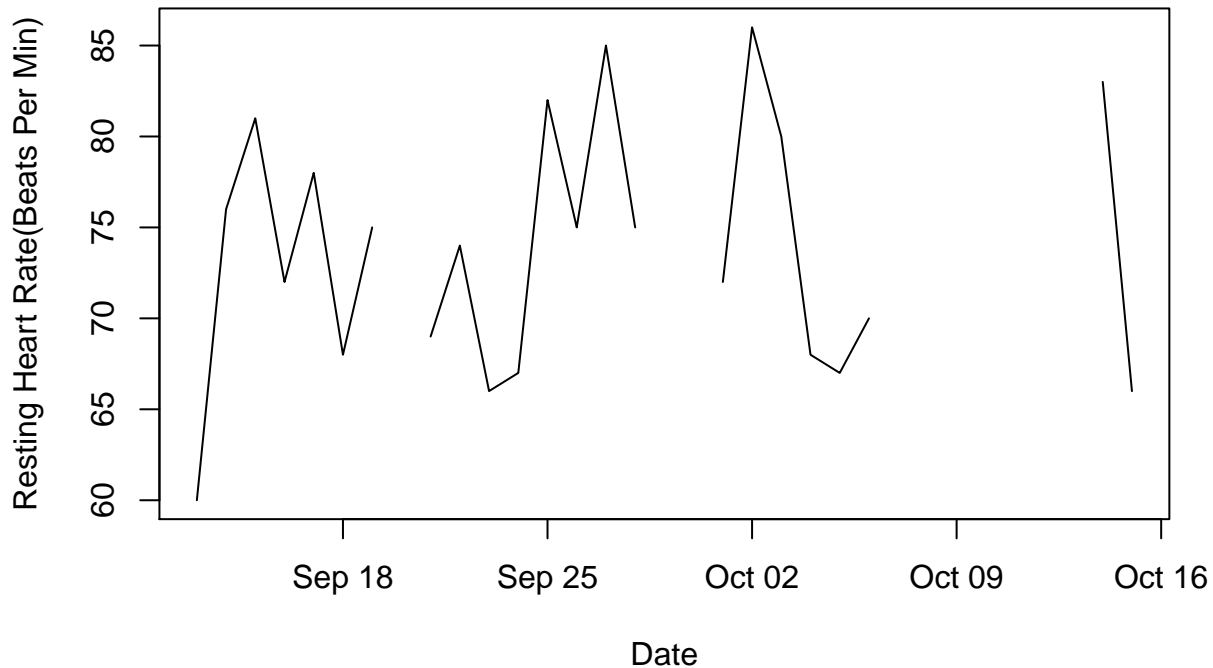
```
tester %>% select("Resting Heart Rate") %>% boxplot()
```



```
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)
```

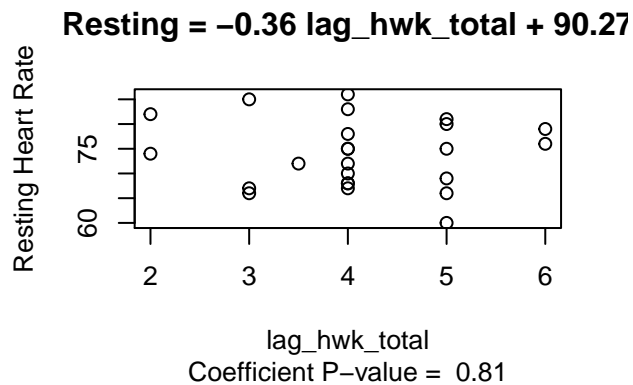
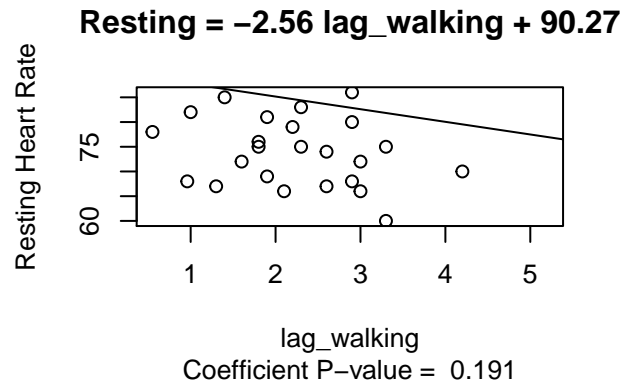
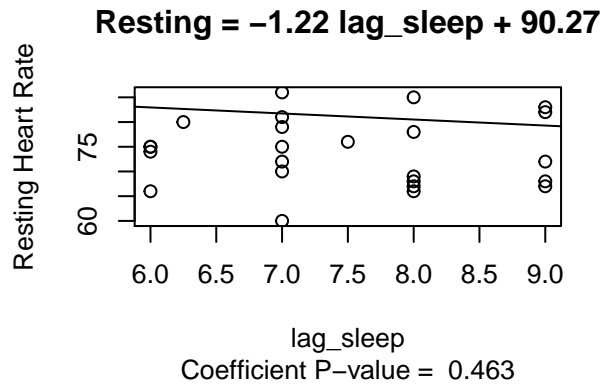
F	M	S	Su	T	Th	W
73.66667	72.75	70.75	78.75	71	75	71.75

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$`tester$days_of_week`)
```

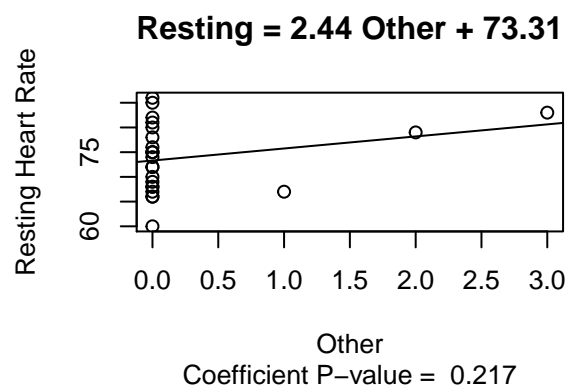
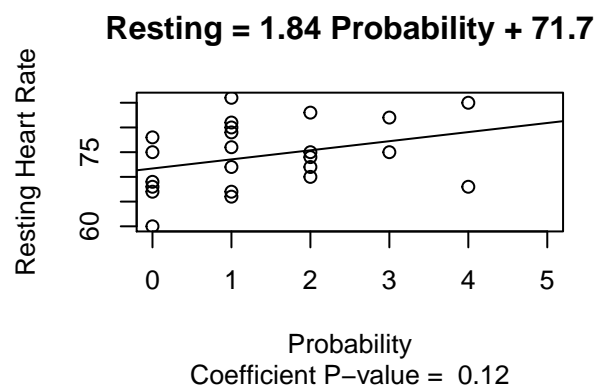
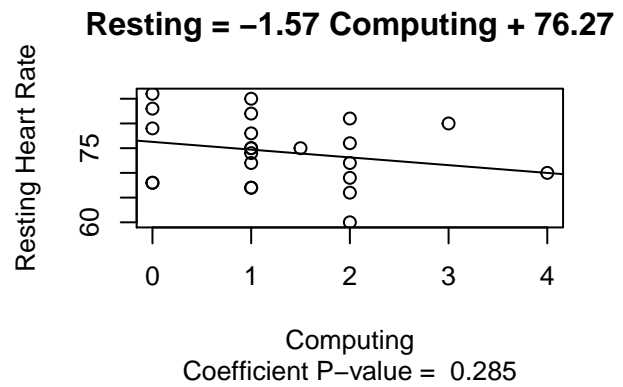
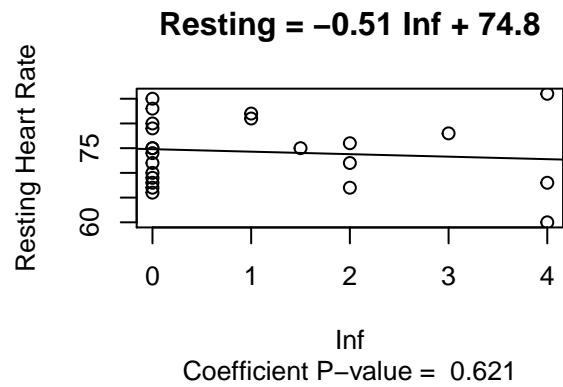
Now I though I'd look at some of the vaiables I created before but now with the dataset without the outlier and looking at the lag of sleep, miles walking, and homework totals. This time I'm going to look at them all together or in other words fit a multivariate regression model. Again, we see that none 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 individual classes and how much time I spend on those classes. Again I didn't see any statistically significant p-values for the correlation coefficient. Although, the variable for my probability class was close.

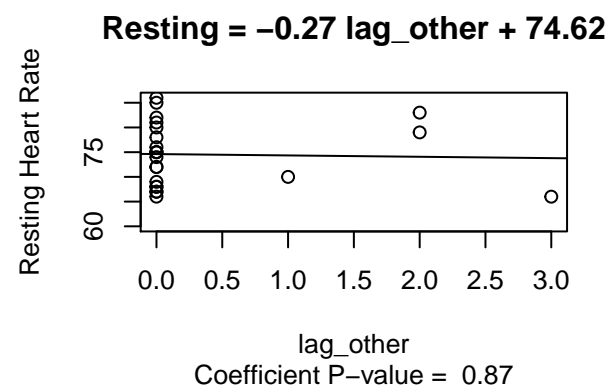
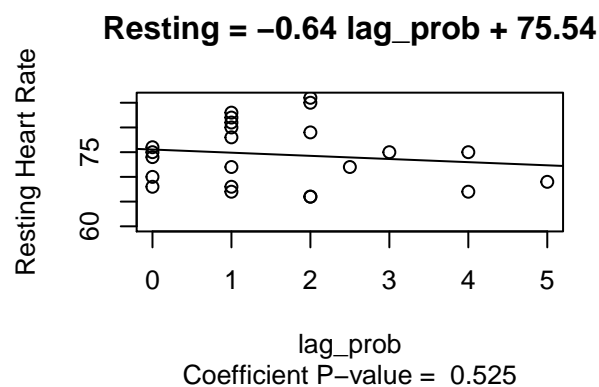
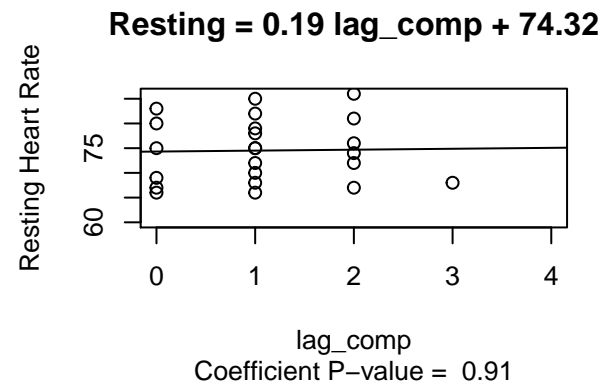
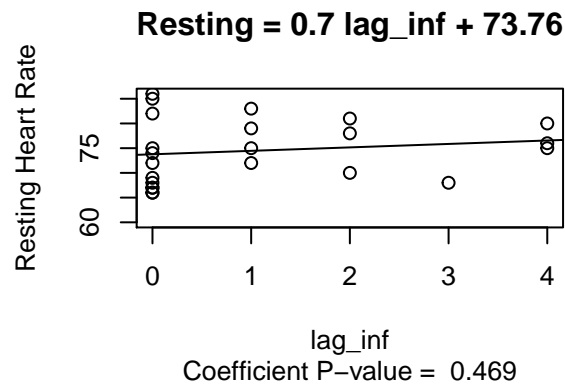
```
tester2 %>% plot_lms(., col_names = c("Inf", "Computing", "Probability", "Other"),
  response_variable = "Resting Heart Rate")
```



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")
```



Answering Questions of HW 5

- 1) I continued to work on my R package. I added some functionality to my `plot_lms` function. It now also outputs the p-value for the coefficient of the slope of the line.
- 2) I did try to add tidyverse commands to make my functions more readable. Although, some of my code would turn vectors into time series when I converted them to tibbles. So hopefully, I can fix that in the future. I think my code is more readable now, especially the code in the vignette. I started looking at whether the day of the week seems to make a big difference. In future drafts I hope to add more analysis of this.
- 3)a) I think that my chunks are pretty easy to understand. I'd like to add some more functions to my add analysis.
 - b) Yes I think that I would like to make my function more applicable to other data sources. Also I'd like to do some significance testing with the days of the week.
 - c) I think that my function do work well with other dataset. Although, I do not have any parts in my functions to check to make sure the data is in a good format. Also I would like to add more abilities for dataset with missing data. For the most part, I just remove them at this point. I think it is pretty easy to update my data considering my function "updating_heartRate" does that. I think someone could collect similar data and use my functions. They aren't overly complicated functions. I think the help files are understandable. Although, I would like to add more details to them if this were to go out into the public.