CWRU DSCI351-351m-451: Lab Exercise LE6 NAME

Inference, Linear Regression, Timeseries Analysis

Prof.:Roger French, TA: Raymond Wieser, Sameera Nalin Venkat

17 November, 2021

Contents

```
## loading libraries

library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(glue)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

6.0.1 LE6, 10 points, 4 questions.

- LE6.1, 1 pt.
- LE6.2, 2 pts.
- LE6.3, 4 pts.
- LE6.4, 3 pts.

6.0.1.1 Lab Exercise (LE) 6

- Inference Guide

There is a useful Inference Cheat Sheet

- in your 3-readings/3-CheatSheets/ folder
- `os2_extra_inference_guide.pdf`

6.1 LE6.1. Fuel efficiency of Prius. (OIS 5.8)

- Fueleconomy.gov,
  - the official US government source
    - for fuel economy information,
  - allows users to share gas mileage information on their vehicles.

The histogram below shows

- the distribution of gas mileage in miles per gallon (MPG)
  - from 14 users who drive a 2012 Toyota Prius.
- The sample mean is 53.3 MPG
  - and the standard deviation is 5.2 MPG.

Note that these data are user estimates

- and since the source data cannot be verified,
- the accuracy of these estimates are not guaranteed.[@noauthor_gas_nodate]

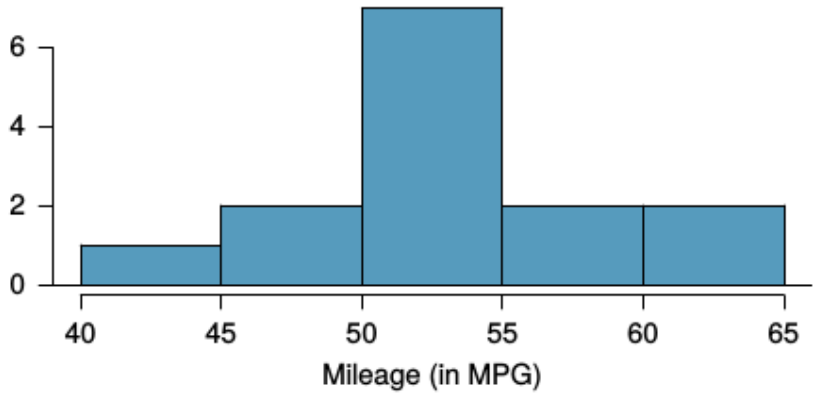


Figure 1: Mileage in MPG

6.1.1 LE6.1.1 We would like to use these data to evaluate

- the average gas mileage of all 2012 Prius drivers.

Do you think this is reasonable?

- Why or why not?

6.1.1.1 LE6.1.1 Answer: The central limit theorem states that if you have a population with mean and standard deviation and take sufficiently large random samples from the population with replacement , then the distribution of the sample means will be approximately normally distributed. This will hold true regardless of whether the source population is normal or skewed, provided the sample size is sufficiently large (usually $n > 30$). If the population is normal, then the theorem holds true even for samples smaller than 30.

We have here, 14 observations, but they are normally distributed, therefore, we can use them to find the true mean gas mileage for all Prius users.

6.1.2 LE6.1.2 The EPA claims that a 2012 Prius gets 50 MPG

- (city and highway mileage combined).

Do these data provide strong evidence against this estimate

- for drivers who participate on http://fueleconomy.gov?
- Note any assumptions you must make as you proceed with the test.

6.1.2.1 LE6.1.2 Answer:

This data does not provide enough evidence against this estimate. There is a good possibility that true mean is 50.

sample mean : 53.3 null value : 50

Hypothesis : - NULL : The true mean is same as what is proposed : 50 - ALTERNATE: The true mean is not 50, but something else.

significance level = 0.05

```
sample_mean = 53.3
proposed_value = 50 ## given in question
n = 14
significance_level = 5/100
std_deviation = 5.2

point_estimate = sample_mean
null_value = proposed_value
standard_error = std_deviation / sqrt(n)


z_score = (point_estimate - null_value)/standard_error
p_value = 2 * pnorm(z_score, lower.tail = FALSE)

## 2- sided
# In an upper-tailed test the decision rule has investigators reject H0 if the
# test statistic is larger than the critical value. In a lower-tailed test the
# decision rule has investigators reject H0 if the test

if(p_value > significance_level){
glue("The chance of occurence of {point_estimate} is greater than
{significance_level} = > there is something going on and we reject H0")
}else
glue("The chance of occurence of {point_estimate} is lower than
{significance_level} = > point estimate came due to sampling
variability. There is nothing going on and we reject H1")
```

```
## The chance of occurence of 53.3 is lower than
## 0.05 = > point estimate came due to sampling
## variability. There is nothing going on and we reject H1
```

6.1.3 LE6.1.3 Calculate a 95% confidence interval

- for the average gas mileage of a 2012 Prius
- by drivers who participate on fueleconomy.gov.

```
critical_val_upper = point_estimate + 1.96 * standard_error
critical_val_lower = point_estimate - 1.96 * standard_error
```

6.1.3.1 LE6.1.3 Answer:

95% confidence intervel (50.57 -> 56.023)

6.2 LE6.2. Diamonds

6.2.1 LE6.2.1 Diamonds Part I. (OIS 5.28)

- Prices of diamonds are determined by what is known as the 4 Cs:
  - cut,
  - clarity,
  - color,
  - and carat weight.

The prices of diamonds go up

- as the carat weight increases,
- but the increase is not smooth.

For example, the difference between the size

- of a 0.99 carat diamond and
  - a 1 carat diamond is undetectable to the naked human eye,
- but the price of a 1 carat diamond tends to be much higher
  - than the price of a 0.99 diamond.

In this question we use two random samples of diamonds,

- 0.99 carats and 1 carat,
- each sample of size 23,

and compare the average prices of the diamonds.

In order to be able to compare equivalent units,

- we first divide the price for each diamond
- by 100 times its weight in carats.

That is, for a 0.99 carat diamond, we divide the price by 99.

For a 1 carat diamond, we divide the price by 100.

The distributions and some sample statistics are shown below.[@wickham_ggplot2:_2016]

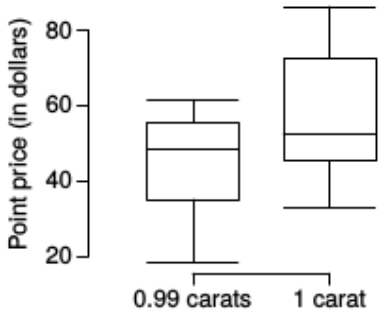


Figure 2: Point Price

	0.99 carats	1 carat
Mean	$ 44.51	$ 56.81
SD	$ 13.32	$ 16.13
n	23	23

Figure 3: Sample Statistics

Conduct a hypothesis test to evaluate

- if there is a difference between the average standardized prices
- of 0.99 and 1 carat diamonds.

Make sure to

- state your hypotheses clearly,
- check relevant conditions,
- and interpret your results in context of the data.

6.2.1.1 LE6.2.1 Answer: Hypothesis :

- NULL : There is no difference between the standardised prices of both these types
- ALTERNATE : There is a difference.

There is nothing going on, and we support the null hypothesis, the difference between the two prices is negligible

```
sample_mean1 = 44.51
sample_mean2 = 56.81
std_deviation1 = 13.32
std_deviation2 = 16.13
proposed_value = 0 ## given in question
n = 23
significance_level = 5/100

point_estimate = sample_mean1 - sample_mean2
null_value = proposed_value
standard_error = sqrt( std_deviation1^2 / n + std_deviation1^2 / n)


z_score = (point_estimate - null_value)/standard_error
p_value = 2 * pnorm(z_score, lower.tail = TRUE)

## 2- sided
# In an upper-tailed test the decision rule has investigators reject H0 if the
# test statistic is larger than the critical value. In a lower-tailed test the
# decision rule has investigators reject H0 if the test

if(p_value > significance_level){
glue("The chance of occurence of {point_estimate} is greater than
{significance_level} = > there is something going on and we reject H0")
}else
glue("The chance of occurence of {point_estimate} is lower than
{significance_level} = > point estimate came due to sampling
variability. There is nothing going on and we reject H1")
```

```
## The chance of occurence of -12.3 is lower than
## 0.05 = > point estimate came due to sampling
## variability. There is nothing going on and we reject H1
```

6.2.2 LE6.2.2 Diamonds Part II. (OIS 5.30)

- We discussed diamond prices
  - (standardized by weight)
  - for diamonds with weights 0.99 carats and 1 carat.

See the table for summary statistics,

- and then construct a 95% confidence interval
- for the average difference
  - between the standardized prices of 0.99 and 1 carat diamonds.

You may assume the conditions for inference are met.

6.2.2.1 LE6.2.2 Answer:

Assuming the conditions for inference is met:

95% confidence intervel : (-19.998 : -4.6014)

```
critical_val_upper = point_estimate + 1.96 * standard_error
critical_val_lower = point_estimate - 1.96 * standard_error
```

6.3 LE6.3 OIStats: Linear regression

- The movie Moneyball focuses on the "quest for the secret of success in baseball".
  - It follows a low-budget team, the Oakland Athletics,
    - who believed that underused statistics, such as
      - a player's ability to get on base,
    - better predict the ability to score runs than typical statistics
      - like home runs, RBIs (runs batted in), and batting average.
  - Obtaining players who excelled in these underused statistics
    - turned out to be much more affordable for the team.

In this lab we'll be looking at data from all 30 Major League Baseball teams

- and examining the linear relationship between runs scored in a season
- and a number of other player statistics.

Our aim will be to summarize these relationships

- both graphically and numerically in order to find which variable, if any,
- helps us best predict a team's runs scored in a season.

6.3.0.1 Data

- Let's load up the data for the 2011 season.
  - Its a file `mlb11.RData`
  - Its located in the data folder of this LE6 in your class repo.

```
load("./data/mlb11.RData")
```

In addition to runs scored,

- there are seven traditionally used variables in the data set:

  - at-bats,
  - hits,
  - home runs,
  - batting average,
  - strikeouts,
  - stolen bases,
  - and wins.

There are also three newer variables:

- on-base percentage,
- slugging percentage,
- and on-base plus slugging.

For the first portion of the analysis - we'll consider the seven traditional variables. - At the end of the lab, you'll work with the newer variables.

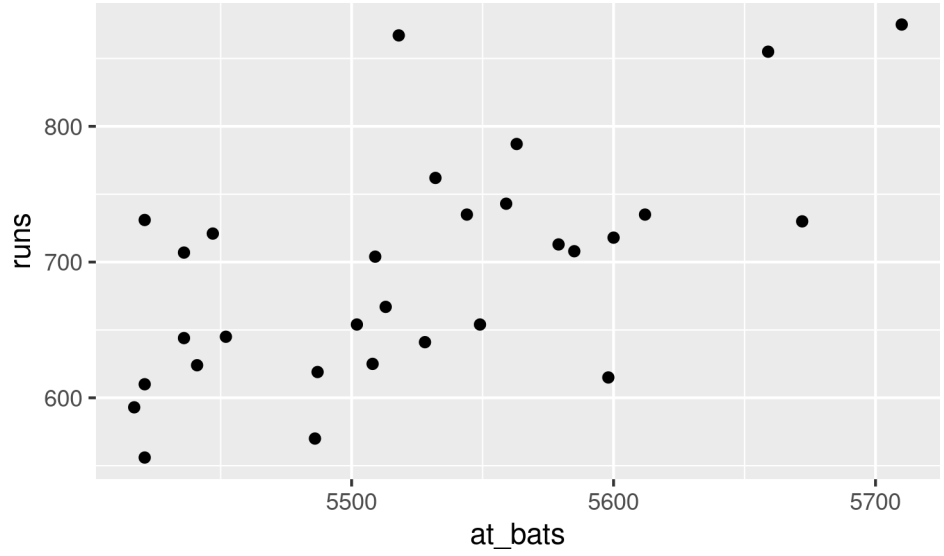
6.3.1 LE6.3.1

- What type of plot would you use to display
  - the relationship between runs and one of the other numerical variables?

Plot this relationship using the variable at_bats as the predictor.

- Does the relationship look linear?

```
## plot runs scored vs at_bats

# m1b11 %>%
#   ggplot(aes(x = ,y =))

## I would use the simple geom_point dot plot, to see any relationship

mlb11 %>%
  ggplot(aes(x = at_bats, y = runs)) + geom_point()
```



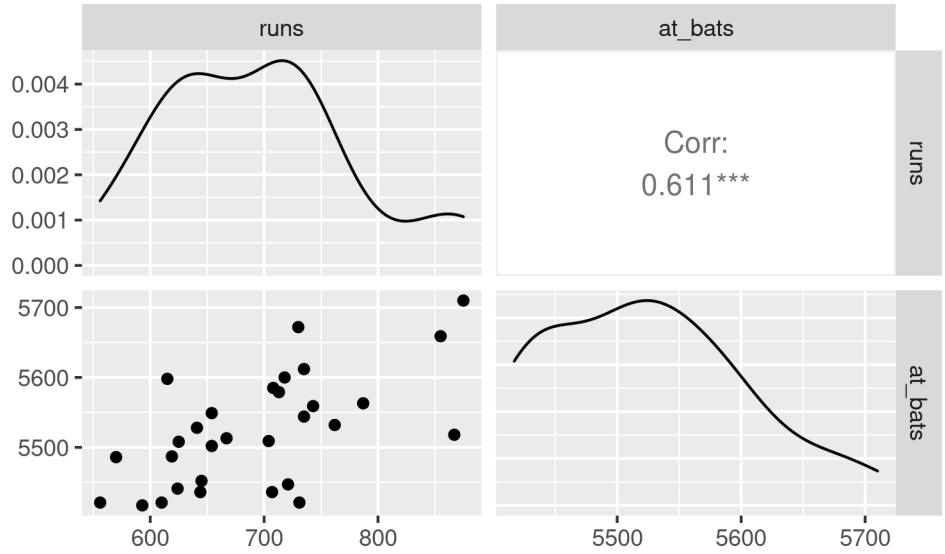
6.3.1.1 LE6.3.1a ANSWER:

I use the simple dot plot to observe the point as is. Until 5600 at_bats, the runs increase linearly with at_bats, Therefore, it seems that there is a linear relationship.

- If you knew a team's at_bats,
- would you be comfortable using a linear model
- to predict the number of runs?

Quantify the strength of the relationship with the correlation coefficient.

```
## Correlation coefficient is found in pairwise plots

ggpairs(mlb11 %>% select(runs, at_bats))
```




```
cor(mlb11$runs, mlb11$at_bats)
```

```
## [1] 0.610627
```

```
## correlation 0.611
```

6.3.1.2 LE6.3.1b ANSWER: correlation coefficient : 0.61, Which means that 61% of the time,when at_bats changes, runs would have a corresponding change.

6.3.1.2.1 Sum of squared residuals

- Think back to the way that we described the distribution of a single variable.
  - Recall that we discussed characteristics such as center, spread, and shape.
  - It's also useful to be able to describe
    - the relationship of two numerical variables,
    - such as runs and at_bats above.

6.3.2 LE6.3.2

- Looking at your plot from the previous exercise,
  - describe the relationship between these two variables.

Make sure to discuss

- the form, direction, and strength of the relationship
- as well as any unusual observations.

6.3.2.1 LE6.3.2 ANSWER:

- Form : Linear
- Direction : Positive
- Strength : correlation coefficient :0.61
- runs has a linear relationship with at_bats
- when at_bats has a positive change, the runs has a posiive change and same goes for negative
- atbats and runs have a correlation coefficient of 0.61

6.3.3 LE6.3.3

- Just as we used the mean and standard deviation to summarize a single variable,
  - we can summarize the relationship between these two variables
    - by finding the line that best follows their association.

Use the following interactive function

- to select the line that you think does the best job
  - of going through the cloud of points.

```
library(statsr)
```

```
## Loading required package: BayesFactor
```

```
## Loading required package: coda
```

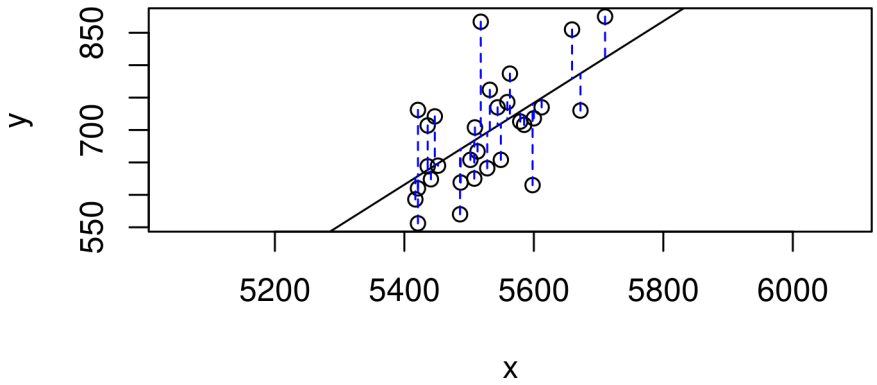
```
## Loading required package: Matrix
```

```
## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard Morey (richarddmorey
##
## Type BFManual() to open the manual.
## ************
```

```
##
## Attaching package: 'statsr'
```

```
## The following objects are masked _by_ '.GlobalEnv':
##
##     mlb11, plot_ss
```

```
plot_ss(x = mlb11$at_bats, y = mlb11$runs)
```



```
## Click two points to make a line.
## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
## (Intercept)            x
##  -2789.2429       0.6305
##
## Sum of Squares:  123721.9
```

You'll need to run this command in an R script (.R file)

- You don't get the interactive prompts when in a .Rmd file

After running this command in a .R script, you'll be prompted

- to click two points on the plot to define a line.

Once you've done that,

- the line you specified will be shown in black
- and the residuals in blue.

Note that there are 30 residuals, one for each of the 30 observations.

- Recall that the residuals are the difference
- between the observed values and the values predicted by the line:

$$e_i = y_i - \hat{y}_i$$

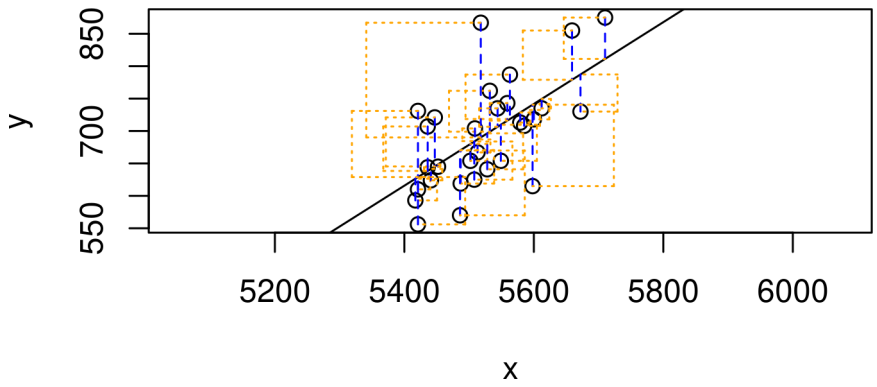
The most common way to do linear regression

- is to select the line that minimizes the sum of squared residuals.

To visualize the squared residuals,

- you can rerun the plot command
- and add the argument `showSquares = TRUE`.

```
plot_ss(x = mlb11$at_bats, y = mlb11$runs, showSquares = TRUE)
```




```
## Click two points to make a line.
## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
## (Intercept)            x
```

```
##  -2789.2429       0.6305
##
## Sum of Squares:  123721.9
```

Note that the output from the `plot_ss` function

- provides you with the slope and intercept of your line
- as well as the sum of squares.

Using `plot_ss`, choose a line

- that does a good job of minimizing the sum of squares.

Run the function several times.

- What was the smallest sum of squares that you got?
- How does it compare to your neighbors?

```
## We have to run it many times and observe sum of squares function.
# Sum of Squares:  145914.4
# Sum of Squares:  144974.3
# Sum of Squares:  163010.4
# Sum of Squares:  125105.3
# Sum of Squares:  236679.7
# Sum of Squares:  175150.1
```

6.3.3.1 LE6.3.3a ANSWER: The smallest sum of squares i got is 125105.3. I could not comapare with a neighbor.

6.3.3.1.1 The linear model

- It is rather cumbersome to try to get the correct least squares line,
  - i. e. the line that minimizes the sum of squared residuals,
  - through trial and error.

Instead we can use the `lm` function in R

- to fit the linear model (a.k.a. regression line).

```
m1 <- lm(runs ~ at_bats, data = mlb11)
```

The first argument in the function `lm` is a formula that takes the form `y ~ x`.

- Here it can be read that we want to make a linear model of runs
  - as a function of at_bats.
- The second argument specifies that R should look in the mlb11 dataframe
  - to find the runs and at_bats variables.

The output of `lm` is an object that contains all of the information we need

- about the linear model that was just fit.
- We can access this information using the summary function.

```
summary(m1)
```

```
##
## Call:
## lm(formula = runs ~ at_bats, data = mlb11)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -125.58  -47.05  -16.59   54.40  176.87
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2789.2429   853.6957  -3.267 0.002871 **
## at_bats         0.6305     0.1545   4.080 0.000339 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.47 on 28 degrees of freedom
## Multiple R-squared:  0.3729, Adjusted R-squared:  0.3505
## F-statistic: 16.65 on 1 and 28 DF,  p-value: 0.0003388
```

```
## where is the sum of squares value here, how do iknow that what i chosewas a good fit
```

Let's consider this output piece by piece.

- First, the formula used to describe the model is shown at the top.
- After the formula you find the five-number summary of the residuals.
- The "Coefficients" table shown next is key;
  - its first column displays the linear model's y-intercept
  - and the coefficient of at_bats.
- With this table, we can write down
  - the least squares regression line for the linear model:

$\hat{y}$ = -2789.2429 + 0.6305*at_{bats}

One last piece of information we will discuss from the summary output

- is the Multiple R-squared, or more simply, R^2.
- The R^2 value represents the proportion of variability
- in the response variable that is explained by the explanatory variable.

6.3.3.2 LE6.3.3b ANSWER:

0.3729 : 37 % variance of runs is explained by at_bats.

6.3.4 LE6.3.4

- Fit a new model that uses homeruns to predict runs.
  - Using the estimates from the R output,
    - write the equation of the regression line.
  - What does the slope tell us in the context of
    - the relationship between success of a team and its home runs?

```
## why don't we have the plot here
m2 <- lm(runs ~ homeruns, data = mlb11)
summary(m2)
```

```
##
## Call:
## lm(formula = runs ~ homeruns, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -91.615 -33.410   3.231  24.292 104.631
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 415.2389    41.6779   9.963 1.04e-10 ***
## homeruns      1.8345     0.2677   6.854 1.90e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.29 on 28 degrees of freedom
## Multiple R-squared:  0.6266, Adjusted R-squared:  0.6132
## F-statistic: 46.98 on 1 and 28 DF,  p-value: 1.9e-07
```

6.3.4.1 LE6.3.4. ANSWER:

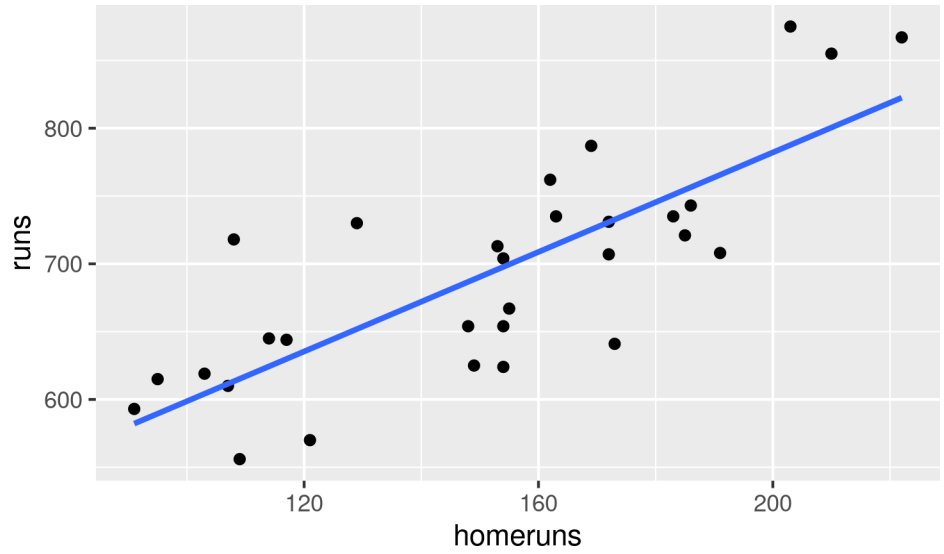
- model m2 built above -$\hat{y} = 415.2389 + 1.8345 * homeruns$
- slope : 1.8345, slops is positive implies, if homeruns increase, runs increase.

6.3.4.1.1 Prediction and prediction errors

- Create a scatterplot with the least squares line laid on top.

```
## This was my next question
ggplot(mlb11,aes(homeruns, runs)) +
  geom_point() +
  geom_smooth(method='lm', se = FALSE)
```

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



The function `abline` plots a line based on its slope and intercept.

- How do you do it in `ggplot2`?

Here, we used a shortcut by providing the model m1,

- which contains both parameter estimates.
- This line can be used to predict y at any value of x.
- When predictions are made for values of x
  - that are beyond the range of the observed data,
  - it is referred to as extrapolation and is not usually recommended.
- However, predictions made within the range of the data are more reliable.
- They are also used to compute the residuals.

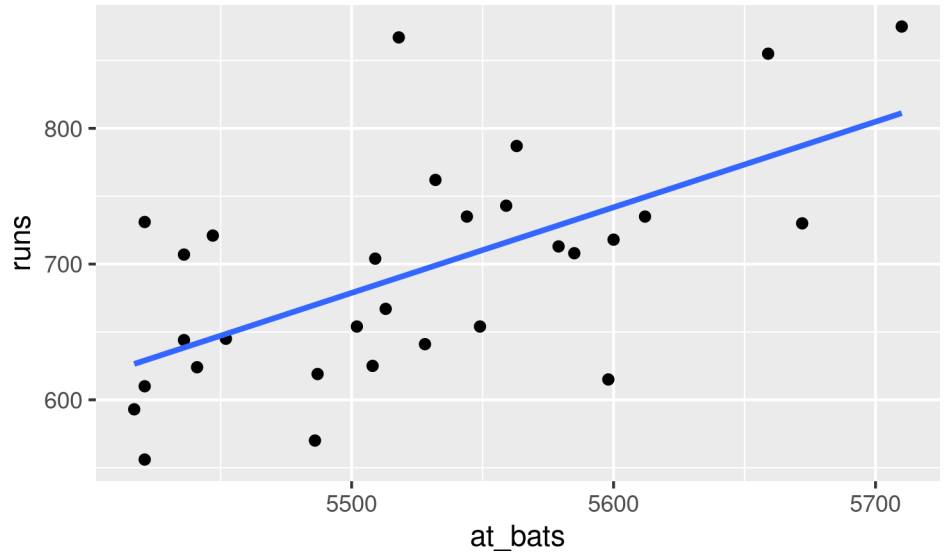
6.3.5 LE6.3.5

- If a team manager saw the least squares regression line
  - and not the actual data,
  - how many runs would he or she predict for a team with 5,578 at-bats?

6.3.5.1 LE6.3.5a ANSWER: 725 : using the model line as refernece

```
## see the line and answer the question.
ggplot(data = mlb11, aes(x = at_bats, y =runs)) +
  geom_point() +
  geom_smooth(method='lm', se = FALSE)
```

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



How many runs would he or she predict for a team with 5,578 at-bats?

- Is this an overestimate or an underestimate, and by how much?
- In other words, what is the residual for this prediction?

6.3.5.2 LE6.3.5b ANSWER: actual - 713 (from scatter plot), the residual is 725- 713 = overestimate of 12

6.3.5.2.1 Model diagnostics

- To assess whether the linear model is reliable,
  - we need to check for
    - (1) linearity,
    - (2) nearly normal residuals, and
    - (3) constant variability.

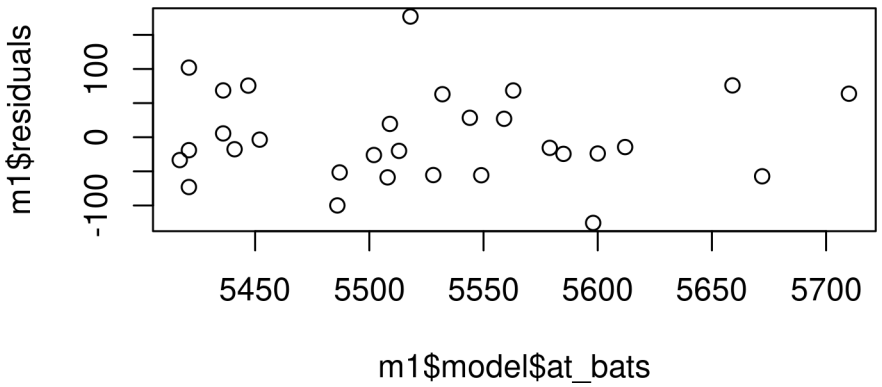
Linearity: You already checked

- if the relationship between runs and at-bats is linear using a scatterplot.
- verify this condition with a plot of the residuals vs. at-bats.

6.3.5.3 LE6.3.5c ANSWER: I alredy checked by plotting at_bats and runs in a scatter plot

- Plot of the residuals vs. at-bats.

```
## How do you plot residuals  ? How do we identify the points

## not usingggplot, because cannot get the data in a single df
plot(x = m1$model$at_bats, y = m1$residuals)
```




```
# plot(x = m1$model$at_bats, y = m1$fitted.values)

mean(m1$residuals) ## the points average to 0
```

```
## [1] -6.231127e-16
```

6.3.6 LE6.3.6

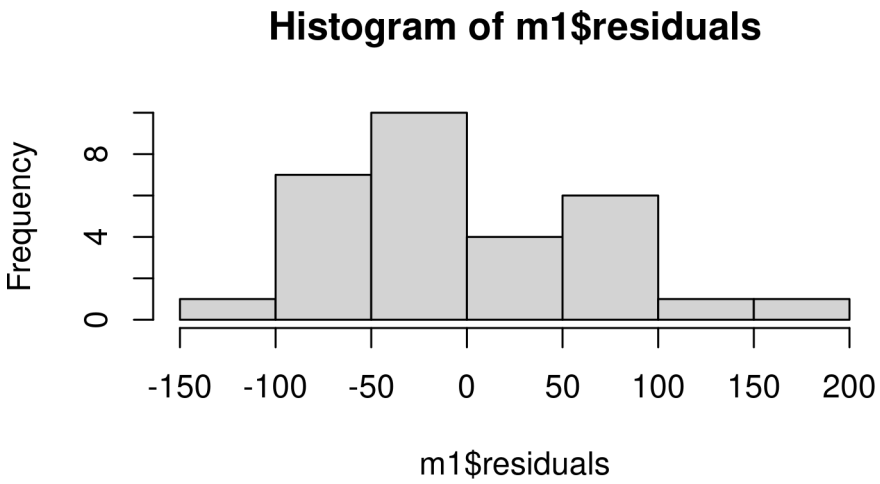
- Is there any apparent pattern in the residuals plot?
  - What does this indicate about the linearity of the relationship
    - between runs and at-bats?

6.3.6.1 LE6.3.6 Answer: The residuals are centered around 0, that is because the best fit line equally bisects the points

6.3.7 LE6.3.7

- Nearly normal residuals: To check this condition,
  - we can look at a histogram,
    - `hist(m1$residuals)`,
  - or a normal probability plot of the residuals.

```
hist(m1$residuals)
```



Based on the histogram and the normal probability plot,

- does the nearly normal residuals condition appear to be met?

6.3.7.1 LE6.3.7 ANSWER:

YES, and the normal distribution is centered on 0

6.3.7.1.1 Constant variability:

6.3.8 LE6.3.8

- Based on the plot above,
  - does the constant variability condition appear to be met?

6.3.8.1 LE6.3.8 ANSWER

With the exception from the intervel 0-50, the constant variability condition seems to be met, the values at the two ends are the same, and they increase as we reach the center. This means that there are only few residual points that lie away from the fitted line and the points increase as we move towards the fitte line.

6.3.9 LE6.3.9

- Choose another traditional variable from mlb11
  - that you think might be a good predictor of runs.

Produce a scatterplot of

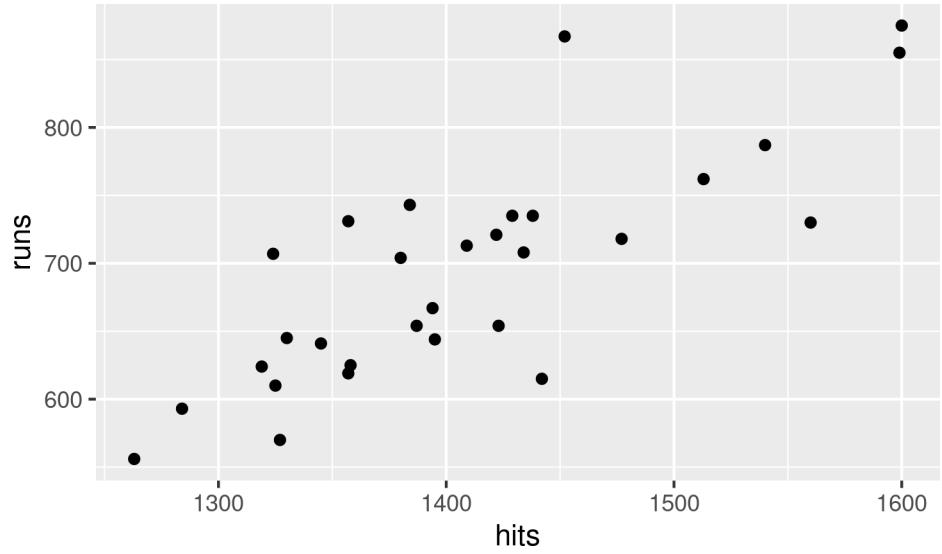
- the two variables
- and fit a linear model.

At a glance, does there seem to be a linear relationship?

6.3.9.1 LE6.3.9a ANSWER:

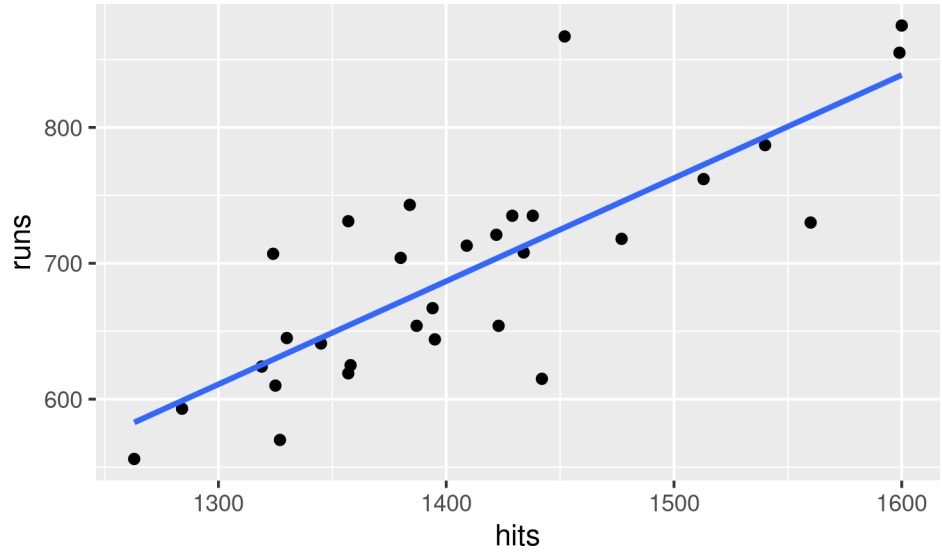
Yes, there seems to be a linear relationship

```
## choosing hits

mlb11 %>%
  ggplot(aes(x = hits, y= runs)) + geom_point()
```




```
ggplot(data = mlb11, aes(x = hits, y =runs)) +
  geom_point() +
  geom_smooth(method='lm', se = FALSE)
```

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




```
mdl_hits <- lm(runs ~ hits, data = mlb11)

## m1 r_squared value : 0.3729
summary(mdl_hits)
```

```
##
## Call:
## lm(formula = runs ~ hits, data = mlb11)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.718  -27.179   -5.233   19.322  140.693
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -375.5600   151.1806  -2.484   0.0192 *
## hits           0.7589     0.1071   7.085 1.04e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 50.23 on 28 degrees of freedom
## Multiple R-squared:  0.6419, Adjusted R-squared:  0.6292
## F-statistic:  50.2 on 1 and 28 DF,  p-value: 1.043e-07
```

```
## mdl_hit r_squared_value : 0.6419
```

How does this relationship compare to the relationship between runs and at_bats?

- Use the R^2 values from the two model summaries to compare.
- Does your variable seem to predict runs
  - better than at bats?
- How can you tell?

6.3.9.2 LE6.3.9b ANSWER:

Hits : have a higher r squared_value than at_bats, it predicts the variability in runs almost double the times, than at_bats does.

r-squared value : 0.6419

6.3.10 LE6.3.10

- Now that you can summarize the linear relationship between two variables,
  - investigate the relationships
    - between runs and each of the other five traditional variables.

Which variable best predicts runs?

Support your conclusion using

- the graphical
- and numerical methods we've discussed
  - (for the sake of conciseness,
  - only include output for the best variable, not all five).

6.3.10.1 LE6.3.10 ANSWER:

bats_vg best predicts runs, with a r-squared of 0.6561 it has normal residuals, and at the ends the points are very low. The data point that is maximum far away is at +40 and -40 respectively.

```
### five variables are :
# already done for homeruns, at_bats and hits
# - batting average,
# - strikeouts,
# - stolen bases,
# - and wins.

# m1, m2, mdl_hits
## make a list of model objects

mdl_1 <- list(m1, m2 ,mdl_hits) # already created models

mdl_bat_av <- lm(runs ~ bat_avg, data = mlb11)
mdl_stk_outs <- lm(runs ~ strikeouts, data = mlb11)
```

```r
mdl_stolen_bas <- lm(runs ~ stolen_bases, data = mlb11)
mdl_wins <- lm(runs ~ wins, data = mlb11)

mdl_2 <- list(mdl_bat_av, mdl_stk_outs, mdl_stolen_bas, mdl_wins)


mdl <- c(mdl_1, mdl_2 )# combining all models


# summary(mdl_bat_av)
sum_mdl <- list()
for (i in mdl){
  temp  <- as.list(summary(i))
  sum_mdl <- c(sum_mdl, temp )
}

for (i in 1:length(sum_mdl)) {
  print(sum_mdl[i]$call)
  print(sum_mdl[i]$r.squared)


}
```

```
## lm(formula = runs ~ at_bats, data = mlb11)
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## [1] 0.3728654
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## lm(formula = runs ~ homeruns, data = mlb11)
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
```

```
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## [1] 0.6265636
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## lm(formula = runs ~ hits, data = mlb11)
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## [1] 0.6419388
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## lm(formula = runs ~ bat_avg, data = mlb11)
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## [1] 0.6560771
## NULL
## NULL
## NULL
```

```
## NULL
## NULL
## NULL
## lm(formula = runs ~ strikeouts, data = mlb11)
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## [1] 0.1693579
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## lm(formula = runs ~ stolen_bases, data = mlb11)
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## [1] 0.002913993
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## lm(formula = runs ~ wins, data = mlb11)
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
```

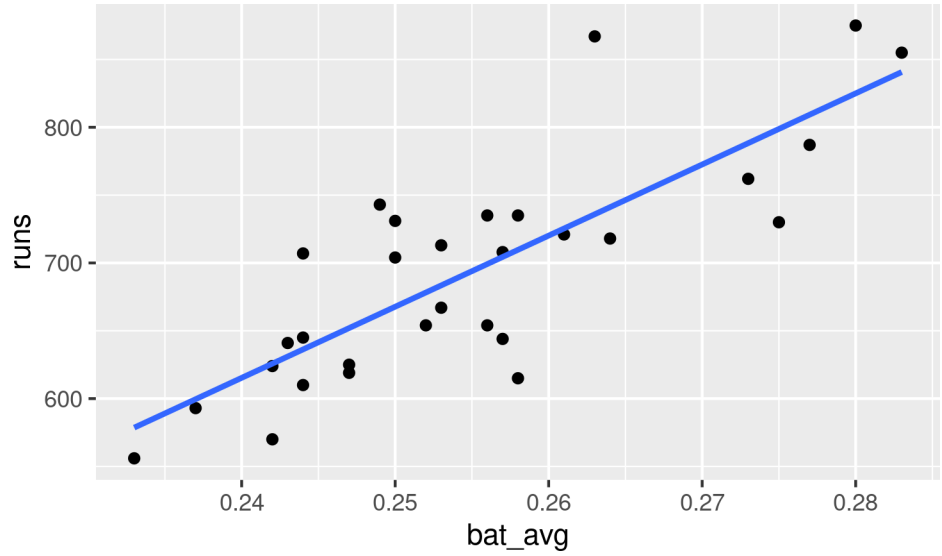
```
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
## [1] 0.3609712
## NULL
## NULL
## NULL
## NULL
## NULL
## NULL
```

```
## runs and bat average is the highest 0.6561 for mdl_bat_av

summary(mdl_bat_av)
```

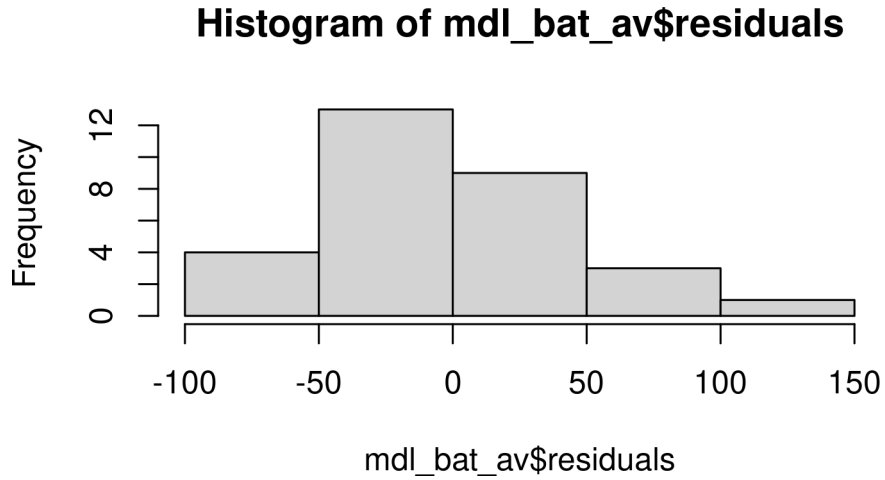
```
##
## Call:
## lm(formula = runs ~ bat_avg, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -94.676 -26.303  -5.496  28.482 131.113
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -642.8      183.1  -3.511  0.00153 **
## bat_avg       5242.2      717.3   7.308 5.88e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.23 on 28 degrees of freedom
## Multiple R-squared:  0.6561, Adjusted R-squared:  0.6438
## F-statistic: 53.41 on 1 and 28 DF,  p-value: 5.877e-08
```

```
ggplot(data = mlb11, aes(x = bat_avg, y =runs)) +
  geom_point() +
  geom_smooth(method='lm', se = FALSE)
```

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




```
hist(mdl_bat_av$residuals)
```




```
plot(x = mdl_bat_av$model$bat_avg, y = mdl_obs$residuals)
```

```
## Error in xy.coords(x, y, xlabel, ylabel, log): object 'mdl_obs' not found
```

6.3.11 LE6.3.11

- Now examine the three newer variables.
  - These are the statistics used by the author of Moneyball
    - to predict a teams success.
  - In general, are they more or less effective at predicting runs
    - than the old variables?
  - Explain using appropriate graphical and numerical evidence.
  - Of all ten variables we've analyzed,
    - which seems to be the best predictor of runs?
  - Using the limited (or not so limited) information you know
    - about these baseball statistics,
    - does your result make sense?
  - Check the model diagnostics for the regression model
    - with the variable you decided was the best predictor for runs.

6.3.11.1 LE6.3.11 ANSWER: They are more effective because of higher r squared values, all above 80%. The best predictor of runs is new_obs variable. Not knowing much about baseball, i cannot tell if this makes sense, but according to the data, newobs with the smalles r-squaredvalue is our best bet for finding runs.

```
## three new variables:
# on-base percentage,
# • slugging percentage,
# • and on-base plus slugging
#"new_onbase"   "new_slug"     "new_obs"

mdl_onbase <- lm(runs ~ new_onbase, data = mlb11)
summary(mdl_onbase)
```

```
##
## Call:
## lm(formula = runs ~ new_onbase, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.270 -18.335   3.249  19.520  69.002
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1118.4      144.5  -7.741 1.97e-08 ***
## new_onbase    5654.3      450.5  12.552 5.12e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.61 on 28 degrees of freedom
## Multiple R-squared:  0.8491, Adjusted R-squared:  0.8437
## F-statistic: 157.6 on 1 and 28 DF,  p-value: 5.116e-13
```

```
mdl_slug <- lm(runs ~ new_slug, data = mlb11)
summary(mdl_slug)
```

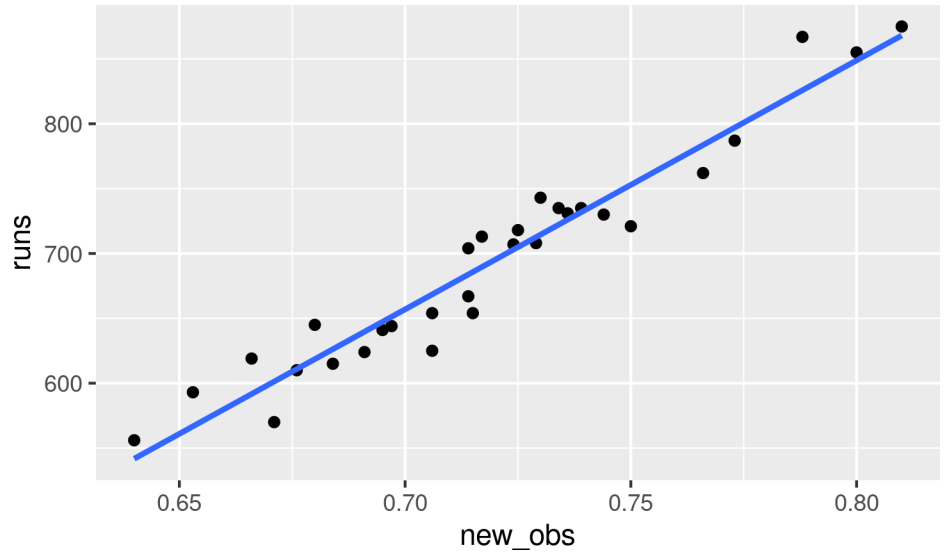
```
##
## Call:
## lm(formula = runs ~ new_slug, data = mlb11)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -45.41 -18.66  -0.91  16.29  52.29
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -375.80      68.71   -5.47 7.70e-06 ***
## new_slug     2681.33     171.83   15.61 2.42e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.96 on 28 degrees of freedom
## Multiple R-squared:  0.8969, Adjusted R-squared:  0.8932
## F-statistic: 243.5 on 1 and 28 DF,  p-value: 2.42e-15
```

```r
mdl_obs <- lm(runs ~ new_obs, data = mlb11)
summary(mdl_obs)
```

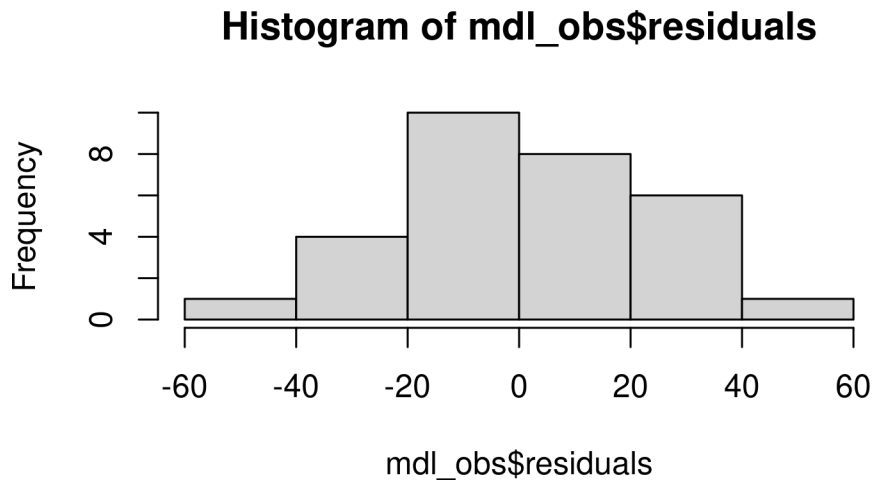
```
##
## Call:
## lm(formula = runs ~ new_obs, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.456 -13.690   1.165  13.935  41.156
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -686.61      68.93  -9.962 1.05e-10 ***
## new_obs      1919.36      95.70  20.057  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.41 on 28 degrees of freedom
## Multiple R-squared:  0.9349, Adjusted R-squared:  0.9326
## F-statistic: 402.3 on 1 and 28 DF,  p-value: < 2.2e-16
```

```r
## 0.9349 is the highest _ new_obs


ggplot(data = mlb11, aes(x = new_obs, y =runs)) +
  geom_point() +
  geom_smooth(method='lm', se = FALSE)
```

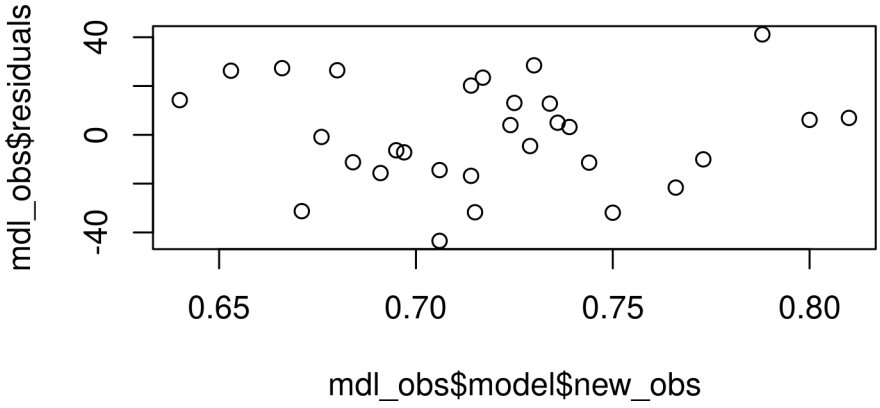
```
## `geom_smooth()` using formula 'y ~ x'
```




```r
hist(mdl_obs$residuals) # normally distributed constant variability
```




```
plot(x = mdl_obs$model$new_obs, y = mdl_obs$residuals)
```



6.3.12 LE6.3.12

- What concepts from the textbook are covered in this lab?
  - What concepts, if any, are not covered in the textbook?
  - Have you seen these concepts elsewhere,
    - e.g. lecture, discussion section, previous labs, or homework problems?
  - Be specific in your answer.

6.3.12.1 LE6.3.12 ANSWER:

- Covered :
- Fitting a line, residuals, and correlation
- Least squares regression
- Extrapolation is treacherous
- Not covered : -Inference for linear regression -Types of outliers in linear regression ——

6.4 LE6.4 Timeseries Analysis

- Time series are a common type of data,
  - consisting of measurements that are continuous over a time range.

In this project we will be using classical decomposition

- to perform analysis on a time series.

First as an introduction to decomposition we will have a quick example.

6.4.1 LE6.4.1

- What is the decomposition of a timeseries?
  - The AirPassengers data set of airline passengers every month for 12 years

```
library(datasets)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.6     v purrr   0.3.4
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x tidyr::unpack() masks Matrix::unpack()
```

```
air <- as.data.frame(AirPassengers)
t <- theme(legend.text = element_text(size = 4) ) +
theme(legend.key.size = unit(0.5, 'cm') ,
axis.text.x = element_text(size = 6,
face = "bold", angle = 45, hjust = 1))
```

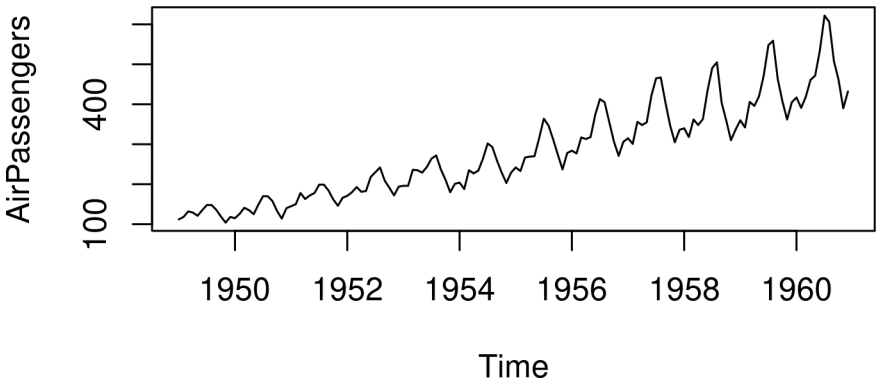
Plot the total time series of air passengers

- What do you notice about the plot?

6.4.1.1 LE6.4.1a ANSWER:

decomposition : breaking down time data, in a way so that we can see the various periodicities , trends and noise separately. The number of air passengers, keep on increasing as years pass.

```
plot(AirPassengers)
```



Use the ts() function in base R

- to define AirPassengers as a time series with a yearly trend

If the data is taken monthly,

- what will the frequency (points per season) of a yearly season be?

6.4.1.2 LE6.4.1b ANSWER: 12

```
?ts()

#define AirPassengers as a time series with a yearly trend
ap_ts <- ts(data = AirPassengers, frequency = 12)
```

Use the `decompose()` function

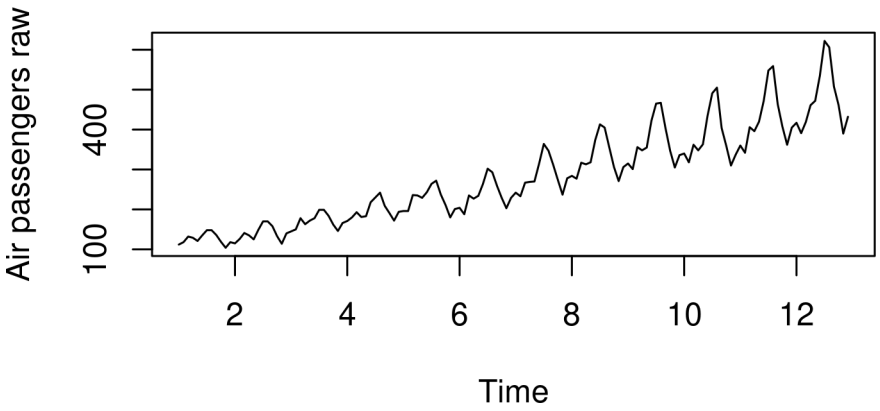
-to decompose the time series and remove the seasonality

The type for this time series is multiplicative

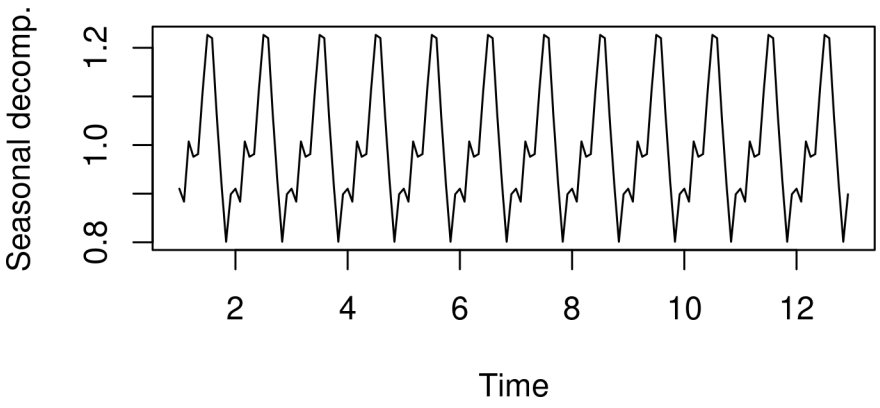
- Plot the decomposed time series,
- what do you notice about the trend?

6.4.1.3 LE6.4.1c ANSWER: The trend of passengers keeps on increasing as the years pass.

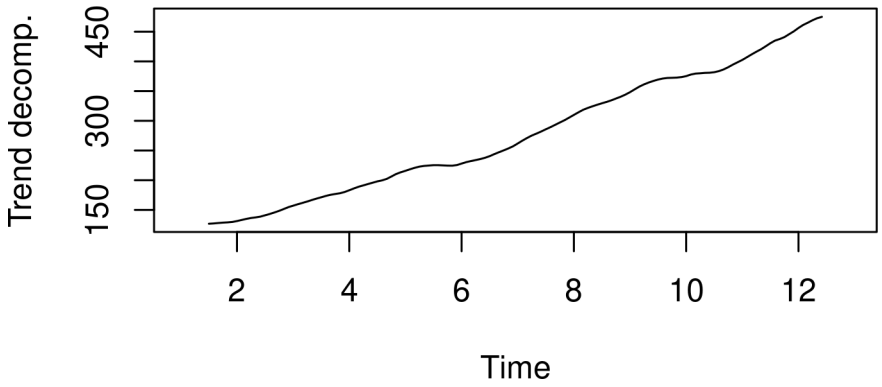
```
ap_ts_d <-decompose(ap_ts,type = "multiplicative")

plot(ap_ts_d$x, ylab="Air passengers raw",)
```




```
plot(ap_ts_d$seasonal, ylab="Seasonal decomp.")
```




```
plot(ap_ts_d$trend, ylab="Trend decomp.")
```




```
# The trend of passengers keeps on increasing as the years pass.
```

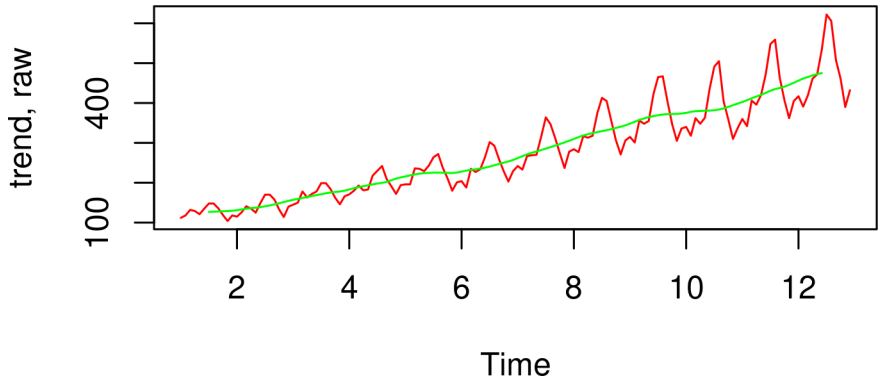
Isolate the trend and plot the trend on top of the raw data with the seasonality included

- How well does the trend represent the data?

6.4.1.4 LE6.4.1d ANSWER:

The trend mimicks a median line of the actual data, and therefore, is able to capture, the major changes as the years pass, but does not capture the variance within a year.

```
simulated <- (ap_ts_d$trend )
plot( ap_ts, type="l", col="red", ylab = "trend, raw" )
lines(simulated, col="green")
```



6.4.2 LE6.4.2

- Now lets try this with a real world time series.
  - In the LE6 data subfolder
  - `2108-351-351m-451-LE6-tsa-data-final.csv`

We'll be using one month of power and weather data from a solar power plant.

The data set variables are as follows:

- `time`: The timestamp at which the data was taken
- `ghir`: Global Horizontal Irradiance from a sensor at the site,
  - the power from the sunlight
  - over an area normal to the surface of the earth ($Watts/m^2$)
- `iacp`: The AC power from the power plant (kW)
- `temp`: The air temperature ($Celsius$)
- `ghi_solargis`: The Global Horizontal Irradiance, not from a sensor,
  - but predicted using weather modeling ($Watts/m^2$)
- `clear`: A logical indicating

  - whether the sky was “clear” during measurement,
  - determined by comparing the ghi and ghi_solargis data
- `ratio`: the ratio of the Global Horizontal Irradiance
  - and the Plane of Array Irradiance
  - (the irradiance normal to the surface of a tilted module)

The power from solar panels is dependent on the irradiance hitting it,

- so a power reading is often meaningless
  - without a corresponding irradiance measurement.

It is useful to have multiple sources of irradiance measurements.

Sensors on the ground are useful because

- they strongly represent the irradiance that is hitting the module;
- however, sensors can begin to drift if not cleaned or calibrated properly.
- An unstable sensor can render an entire data set useless.

To combat this, we also have irradiance data from SolarGIS,

- a company that uses satellite images to model and predict
  - the irradiance at the surface of the earth.

Plot the irradiance and power for the first week of data,

- how does the irradiance look compared to
  - what you would expect from the trend of sunlight?
- How well does the power and irradiance match up?

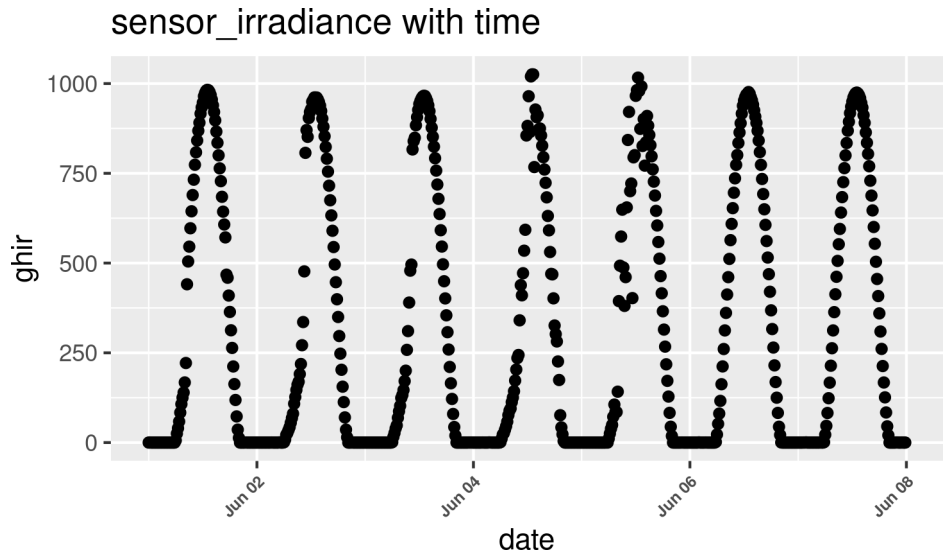
6.4.2.1 LE6.4.2a ANSWER:

Irradiance in the sensor and the gis match well, as expected irradiance varies every 24 hours or 1 day. Power and irradiance match up well, as can be seen by the graph , where all 3 are together.

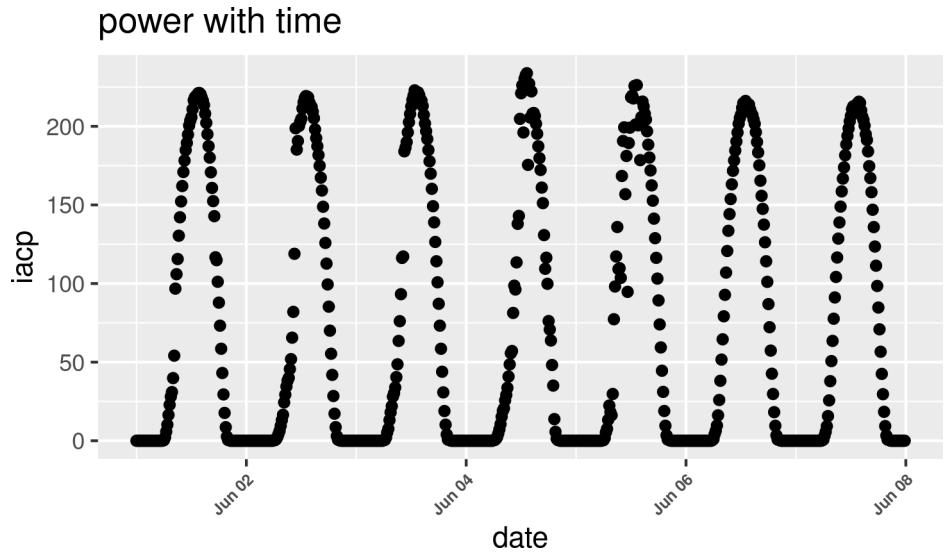
```
input1 <-read.csv("data/2108-351-351m-451-LE6-tsa-data-final.csv", header=TRUE)
head(input1)
```

```
##                  time ghir iacp  temp ghi_solargis clear    ratio
## 1 2012-06-01 00:00:00 0.00    0 20.11            0 False 1.451629
## 2 2012-06-01 00:15:00 0.29    0 19.88            0 False 1.455344
## 3 2012-06-01 00:30:00 0.00    0 19.70            0 False 1.460904
## 4 2012-06-01 00:45:00 0.00    0 19.46            0 False 1.468499
## 5 2012-06-01 01:00:00 0.00    0 19.24            0 False 1.478400
## 6 2012-06-01 01:15:00 0.00    0 19.10            0 False 1.490993
```

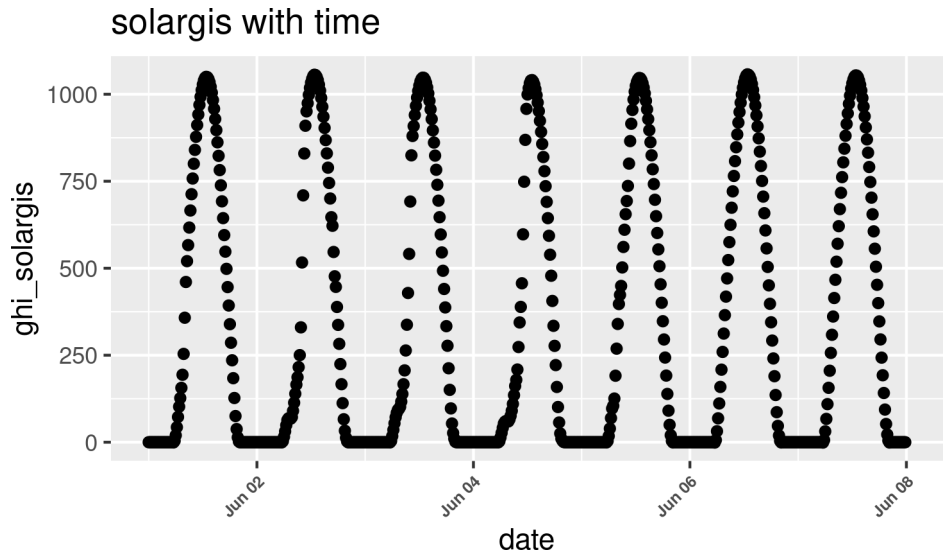
```
## add labels to plots
## first week : first convert time to posicxt, then extract date and filter on it
input1 %>% mutate(
  date = format(
    as.POSIXct(time, format = "%Y-%m-%d %H:%M:%S"),format = "%Y-%m-%d")) %>%
  filter(date >= "2012-06-01" & date <= "2012-06-07") %>%
  ggplot(aes(x= as.POSIXct(time, format = "%Y-%m-%d %H:%M:%S"), y= ghir))+ geom_point() + xlab("date") +
```



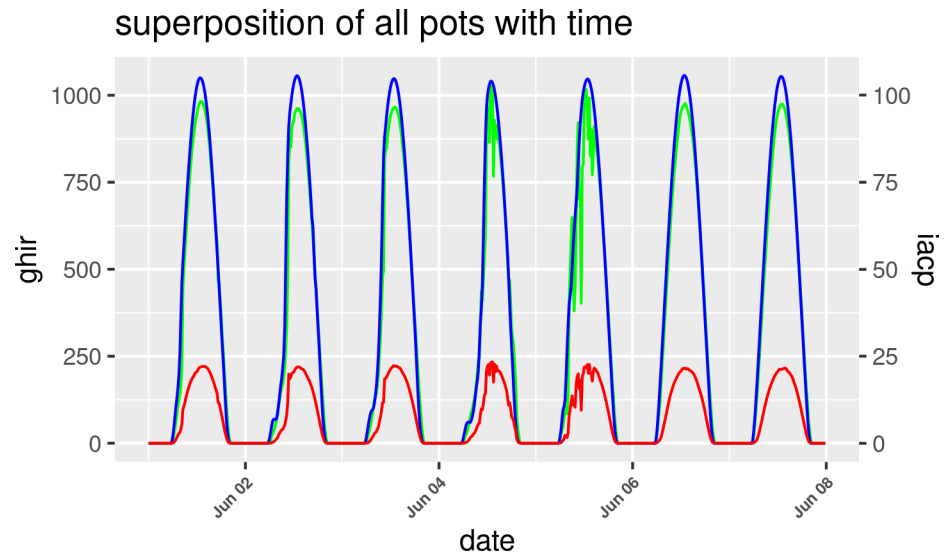

```
input1 %>% mutate(
  date = format(
    as.POSIXct(time, format = "%Y-%m-%d %H:%M:%S"),format = "%Y-%m-%d")) %>%
  filter(date >= "2012-06-01" & date <= "2012-06-07") %>%
  ggplot(aes(x= as.POSIXct(time, format = "%Y-%m-%d %H:%M:%S"), y= iacp))+ geom_point() + xlab("date") +
```




```
input1 %>% mutate(
  date = format(
    as.POSIXct(time, format = "%Y-%m-%d %H:%M:%S"),format = "%Y-%m-%d")) %>%
  filter(date >= "2012-06-01" & date <= "2012-06-07") %>%
  ggplot(aes(x= as.POSIXct(time, format = "%Y-%m-%d %H:%M:%S"), y= ghi_solargis))+ geom_point() + xlab(
```




```
input1 %>% mutate(
  date = format(
    as.POSIXct(time, format = "%Y-%m-%d %H:%M:%S"),format = "%Y-%m-%d")) %>%
  filter(date >= "2012-06-01" & date <= "2012-06-07") %>%
ggplot() +
geom_line(aes(x= as.POSIXct(time, format = "%Y-%m-%d %H:%M:%S"), y = ghir), group =1, color ="green")+
geom_line(aes(x= as.POSIXct(time, format = "%Y-%m-%d %H:%M:%S"), y = ghi_solargis), group =1, color ="bl
geom_line(aes(x= as.POSIXct(time, format = "%Y-%m-%d %H:%M:%S"), y = iacp), group =1, color ="red")+labs
scale_y_continuous(sec.axis = sec_axis(trans =~./10, name ="iacp")) + xlab("date")+
  scale_colour_manual("",
                      breaks = c("iacp", "ghir", "ghi_solargis"),
                      values = c("iacp"="red", "ghir"="green", "ghi_solargis"="blue")) +t
```

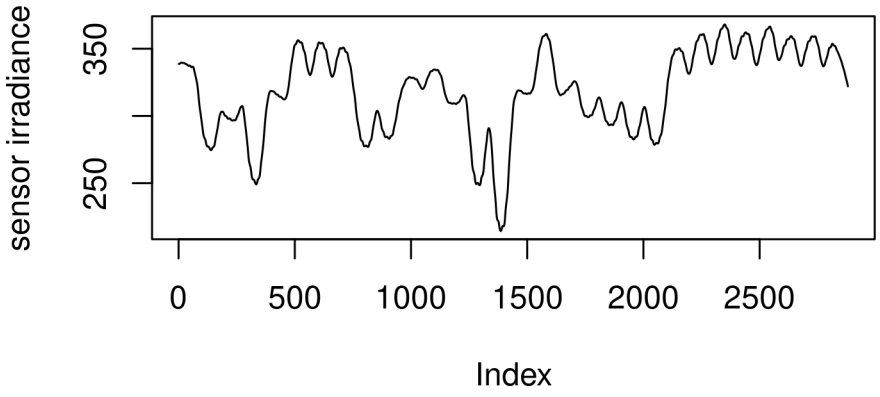


Use the `ts()` functions and the `stlplus()` function from the `stlplus` package

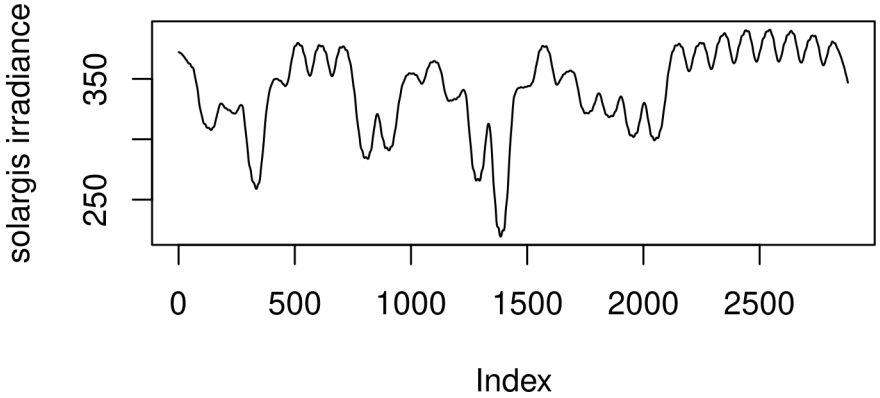
- to decompse the sensor and SolarGIS irradiance and the power
  - for the whole month.
- Plot each of the decompositions, what do you notice?

6.4.2.2 LE6.4.2b ANSWER: Notice : all the different trends, seasonality and noise are similar.

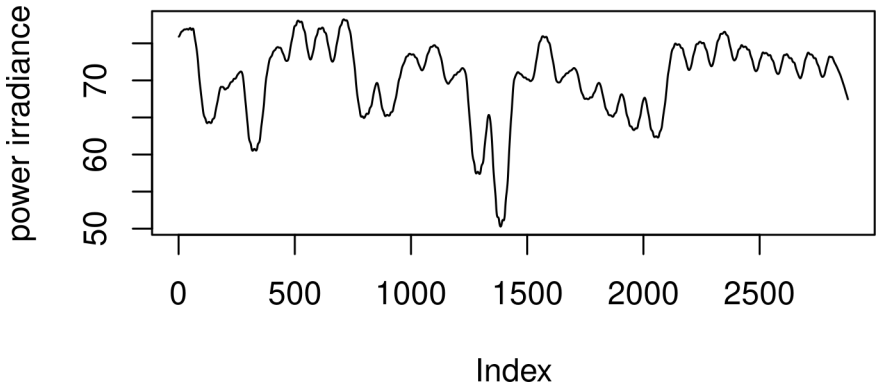
```
# think carefully about the frequency you'll need to define for this data
# what is the seasonal component to this data and how nay data points make up a season?
# use s.window = "periodic" for the stlplus function
library(stlplus)
?stlplus()

#convert data to timeseries objects

sen_ghi <- ts(data = input1$ghir, frequency =4*24) ## change this
gis_ghi <- ts(data = input1$ghi_solargis , frequency = 4*24)
pwr <- ts(data = input1$iacp, frequency = 4*24)

#decompose using stlplus
sg_d <- stlplus(sen_ghi,s.window = "periodic")
gg_d <- stlplus(gis_ghi,s.window = "periodic")
pwr_d <- stlplus(pwr,s.window = "periodic")

## do not know how to use ggplot, therefore reverting to NORMAL PLOT
# trend
plot(sg_d$data$trend, type="l", ylab = "sensor irradiance")
```



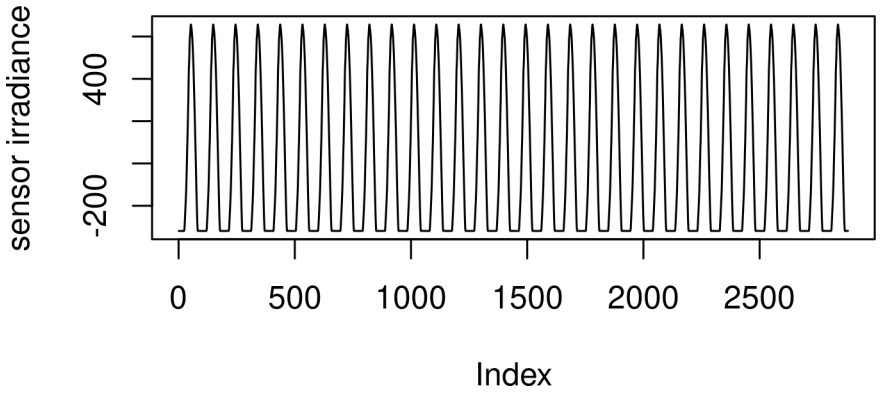

```
plot(gg_d$data$trend, type="l", ylab = "solargis irradiance")
```



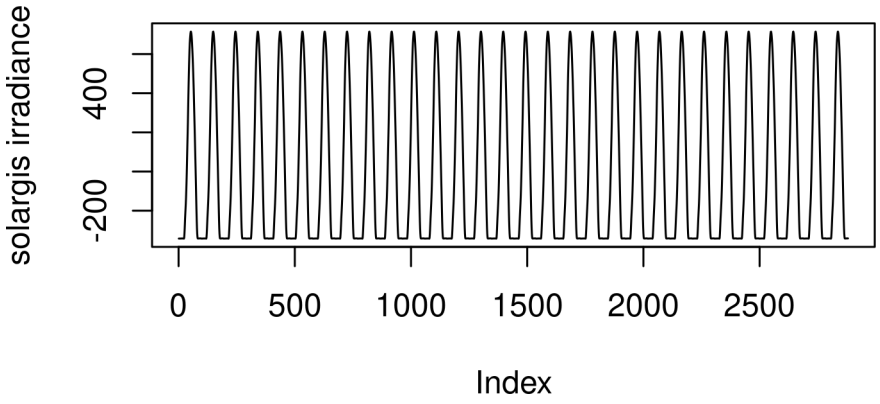

```
plot(pwr_d$data$trend, type="l", ylab = "power irradiance")
```



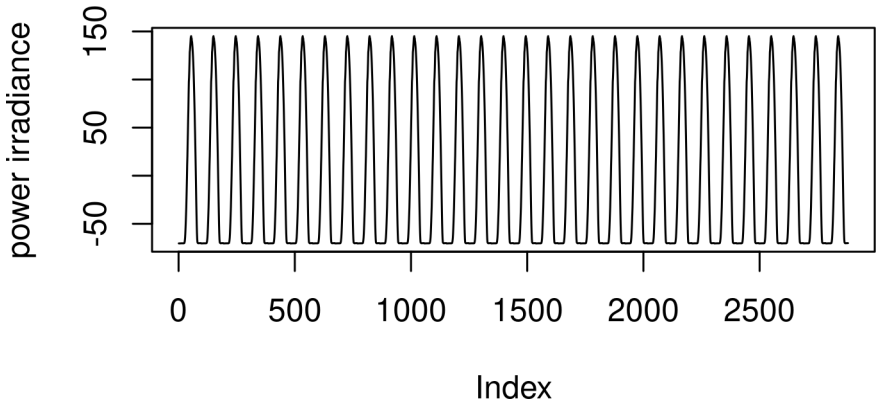

```
# seasonality
plot(sg_d$data$seasonal, type="l", ylab = "sensor irradiance")
```



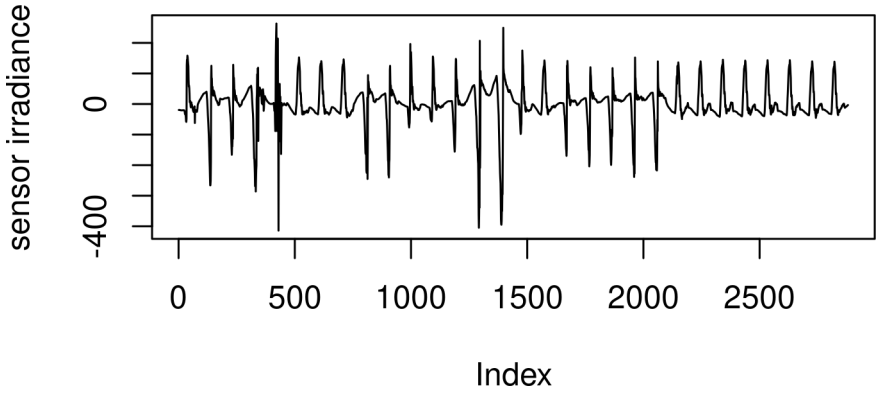

```
plot(gg_d$data$seasonal, type="l", ylab = "solargis irradiance")
```



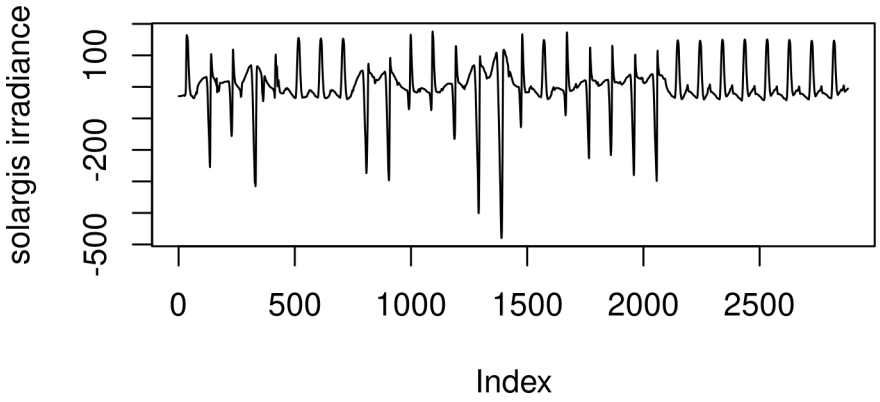

```
plot(pwr_d$data$seasonal, type="l", ylab = "power irradiance")
```



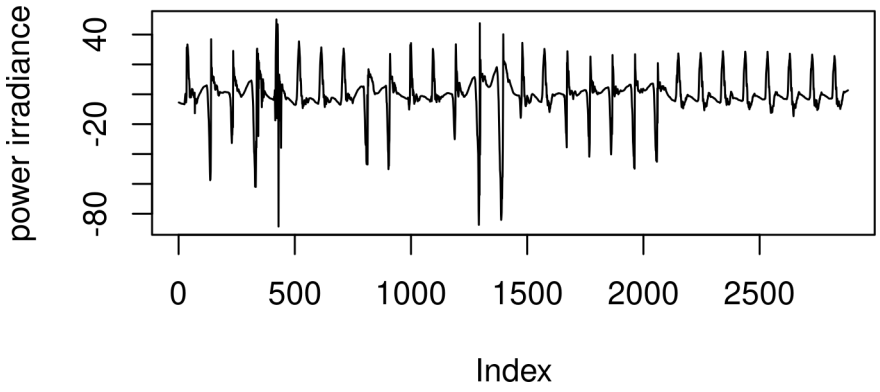

```
# noise
plot(sg_d$data$remainder, type="l", ylab = "sensor irradiance")
```




```
plot(gg_d$data$remainder, type="l", ylab = "solargis irradiance")
```




```
plot(pwr_d$data$remainder, type="l", ylab = "power irradiance")
```



Isolate the trends for the 3 time series you just decomposed

- and build a linear model for each one.

Compare the models to each other,

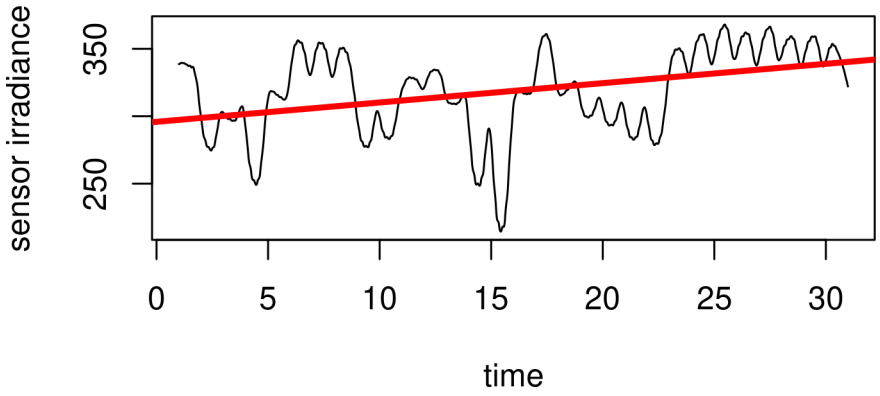
- how are they different?

6.4.2.3 LE6.4.2c ANSWER: They have different slopes, but similar direction and spread.

```
# plot(sg_d$data$trend, type="l")
# plot(gg_d$data$trend, type="l")
# plot(pwr_d$data$trend, type="l")


y <- sg_d$data$trend
x <- sg_d$time
model1 <- lm(y ~ x)
summary(model1)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.543  -18.987    3.705   20.206   51.279
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 295.88174    1.13373     261   <2e-16 ***
## x             1.43376    0.06233      23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.97 on 2878 degrees of freedom
## Multiple R-squared:  0.1553, Adjusted R-squared:  0.155
## F-statistic: 529.1 on 1 and 2878 DF,  p-value: < 2.2e-16
```

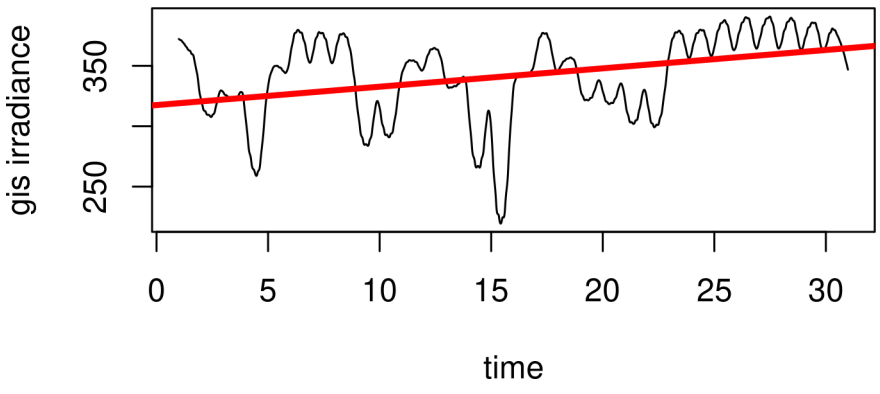
```
plot(x,y, xlab="time", ylab="sensor irradiance", type = "l")

abline(model1, lwd =3, col ="red")
```




```
y <- gg_d$data$trend
x <- gg_d$time
model2 <- lm(y ~ x)
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -121.569  -18.965    6.223   23.220   53.151
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 317.52385    1.26607  250.79   <2e-16 ***
## x             1.51993    0.06961   21.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.35 on 2878 degrees of freedom
## Multiple R-squared:  0.1421, Adjusted R-squared:  0.1418
## F-statistic: 476.8 on 1 and 2878 DF,  p-value: < 2.2e-16
```

```
plot(x,y, xlab="time", ylab="gis irradiance", type = "l")
abline(model2, lwd =3, col ="red")
```

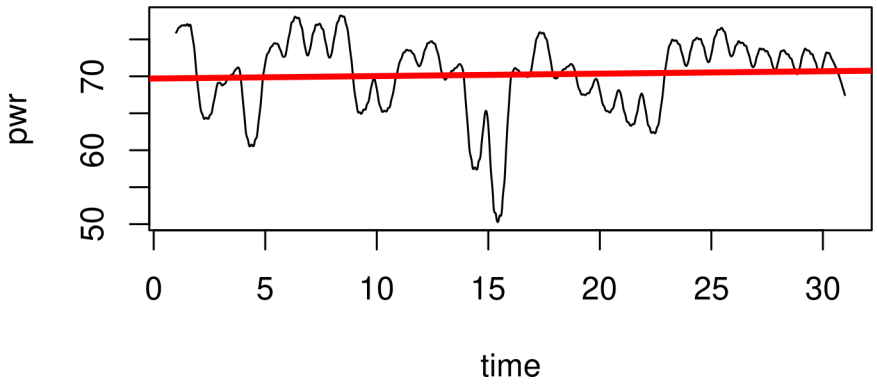



```
y <- pwr_d$data$trend
x <- pwr_d$time
model3 <- lm(y ~ x)
```

```
summary(model3)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.9218  -2.8153   0.9651   3.4244   8.2455
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 69.70711    0.19894 350.391  < 2e-16 ***
## x            0.03276    0.01094   2.995  0.00277 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.083 on 2878 degrees of freedom
## Multiple R-squared:  0.003107,   Adjusted R-squared:  0.002761
## F-statistic: 8.971 on 1 and 2878 DF,  p-value: 0.002766
```

```
plot(x,y, xlab="time", ylab="pwr", type = "l")
abline(model3, lwd =3, col ="red")
```




```
# slope of each line is positive, with power being just a very small value 0.03276
```

Solar panel degradation leads to less power output over time

- at the same irradiance conditions.

Based on the linear models you found for the trends of power and irradiance,

- is this system degrading over time?

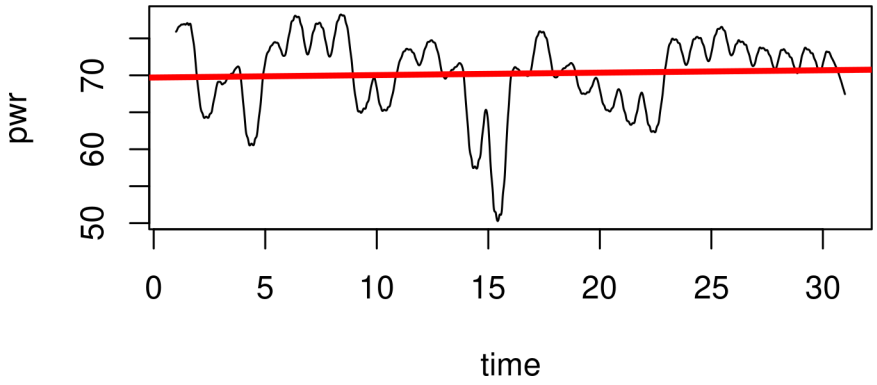
6.4.2.4 LE6.4.2d ANSWER: The system is not degrading over time, because the slope is positive,0.03276.

```
y <- pwr_d$data$trend
x <- pwr_d$time
model3 <- lm(y ~ x)
summary(model3)
```

```
##
## Call:
## lm(formula = y ~ x)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.9218  -2.8153   0.9651   3.4244   8.2455
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 69.70711    0.19894 350.391  < 2e-16 ***
## x            0.03276    0.01094   2.995  0.00277 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.083 on 2878 degrees of freedom
## Multiple R-squared:  0.003107,   Adjusted R-squared:  0.002761
## F-statistic: 8.971 on 1 and 2878 DF,  p-value: 0.002766
```

```
plot(x,y, xlab="time", ylab="pwr", type = "l")
abline(model3, lwd =3, col ="red")
```



How do the sensor GHI and the SolarGIS GHI compare to power?

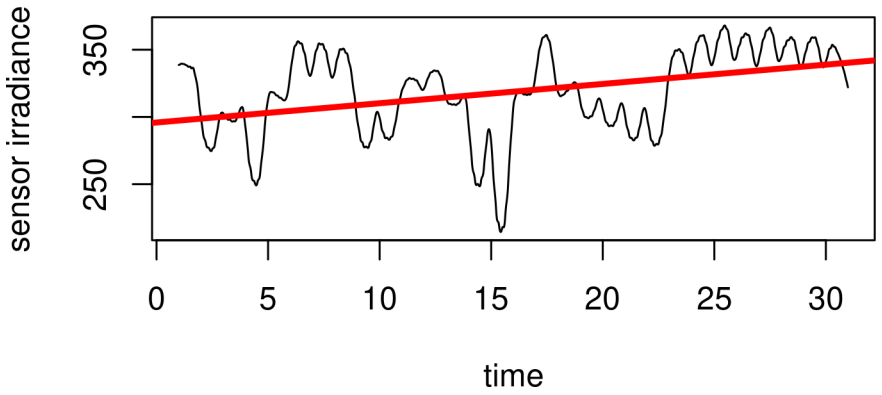
6.4.2.5 LE6.4.2e ANSWER: sensor GHI has a lower slope than that of solarGHI, the sensors could be hindered or be degrading if they do not reflect the true value, the power has a much smaller slope than either of these two values.

```
y <- sg_d$data$trend
x <- sg_d$time
model1 <- lm(y ~ x)
summary(model1)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.543  -18.987    3.705   20.206   51.279
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 295.88174    1.13373     261   <2e-16 ***
## x             1.43376    0.06233      23   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.97 on 2878 degrees of freedom
## Multiple R-squared:  0.1553, Adjusted R-squared:  0.155
## F-statistic: 529.1 on 1 and 2878 DF,  p-value: < 2.2e-16
```

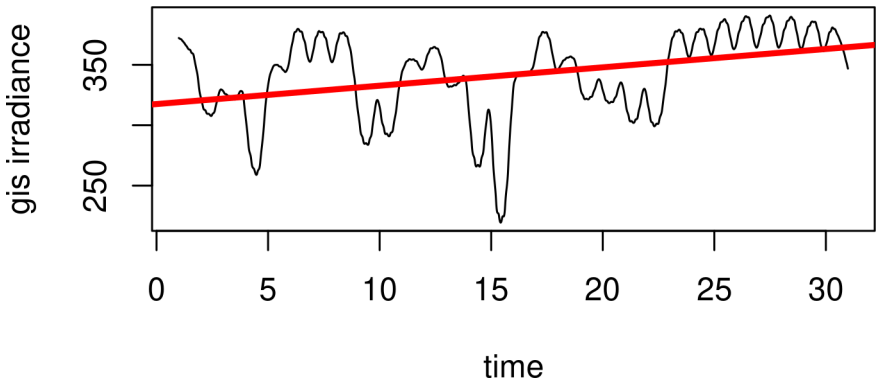
```
plot(x,y, xlab="time", ylab="sensor irradiance", type = "l")

abline(model1, lwd =3, col ="red")
```




```
# slope :1.43376

y <- gg_d$data$trend
x <- gg_d$time
model2 <- lm(y ~ x)
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -121.569  -18.965    6.223   23.220   53.151
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 317.52385    1.26607  250.79   <2e-16 ***
## x             1.51993    0.06961   21.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.35 on 2878 degrees of freedom
## Multiple R-squared:  0.1421, Adjusted R-squared:  0.1418
## F-statistic: 476.8 on 1 and 2878 DF,  p-value: < 2.2e-16
```

```
plot(x,y, xlab="time", ylab="gis irradiance", type = "l")
abline(model2, lwd =3, col ="red")
```




```
#slope : 1.51993
```

6.4.2.6 Links

http://www.r-project.org

http://rmarkdown.rstudio.com/

https://www.openintro.org/stat/textbook.php?stat_book=os