Predictive_Analysis

Vinoth Aryan Nagabosshanam

March 23 2017

predictive analysis Simple linear regression example

```
wc<-read.csv("B:\\data science courses\\Datasets_BA 2\\wc-at.csv")
head(wc)
```

```
##   Waist    AT
## 1 74.75 25.72
## 2 72.60 25.89
## 3 81.80 42.60
## 4 83.95 42.80
## 5 74.65 29.84
## 6 71.85 21.68
```

```
str(wc)
```

```
## 'data.frame':    109 obs. of  2 variables:
##  $ Waist: num  74.8 72.6 81.8 84 74.7 ...
##  $ AT   : num  25.7 25.9 42.6 42.8 29.8 ...
```

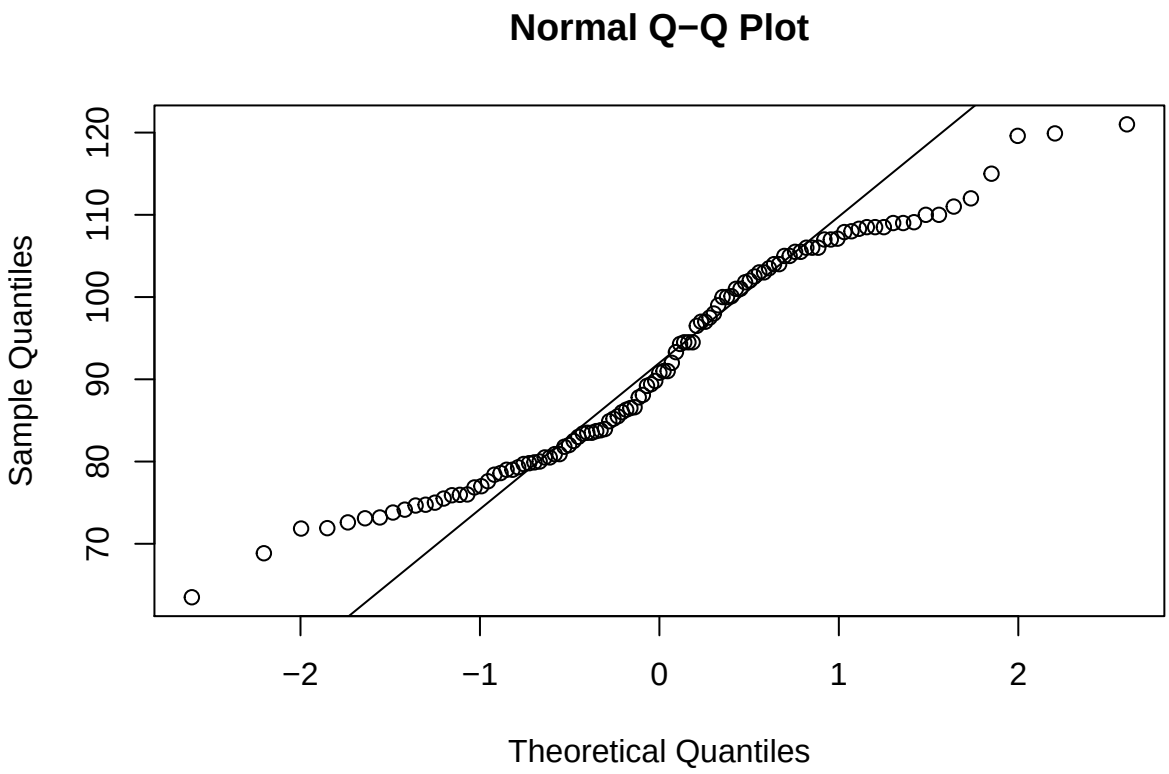
```
summary(wc)
```

```
##      Waist             AT
##  Min.   : 63.5   Min.   : 11.44
##  1st Qu.: 80.0   1st Qu.: 50.88
##  Median : 90.8   Median : 96.54
##  Mean   : 91.9   Mean   :101.89
##  3rd Qu.:104.0   3rd Qu.:137.00
##  Max.   :121.0   Max.   :253.00
```

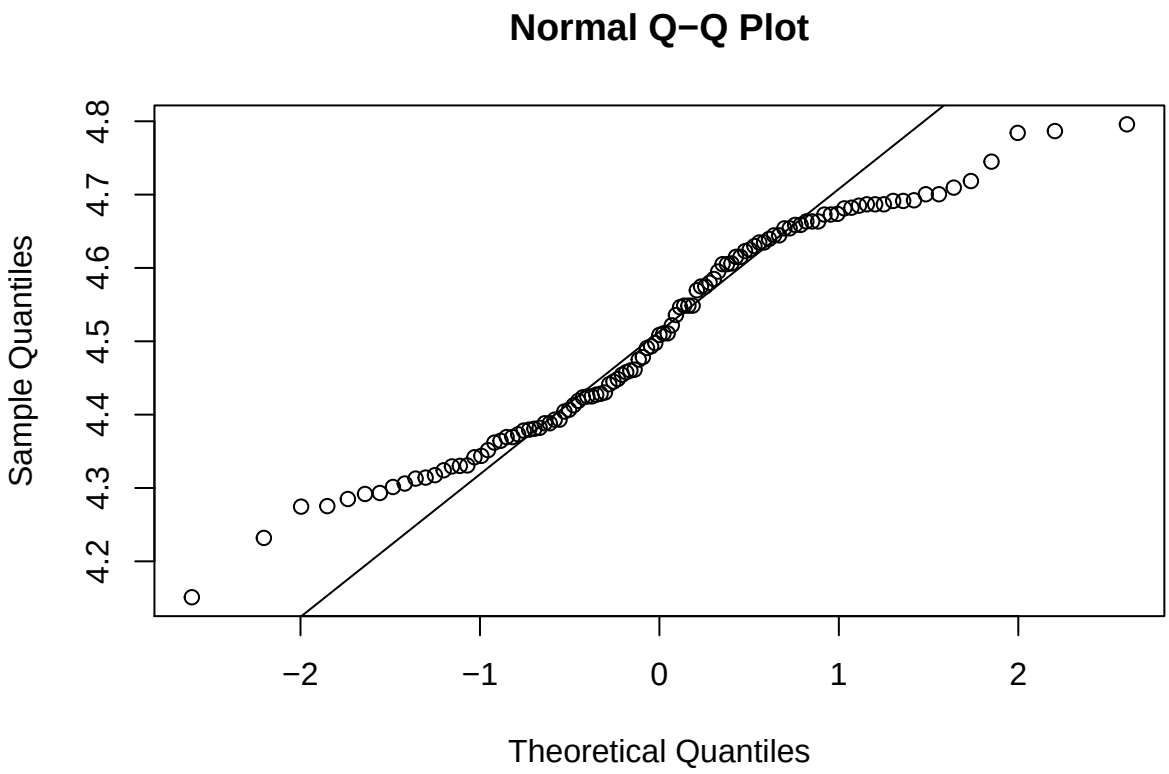
Aanlysis data

EDA part

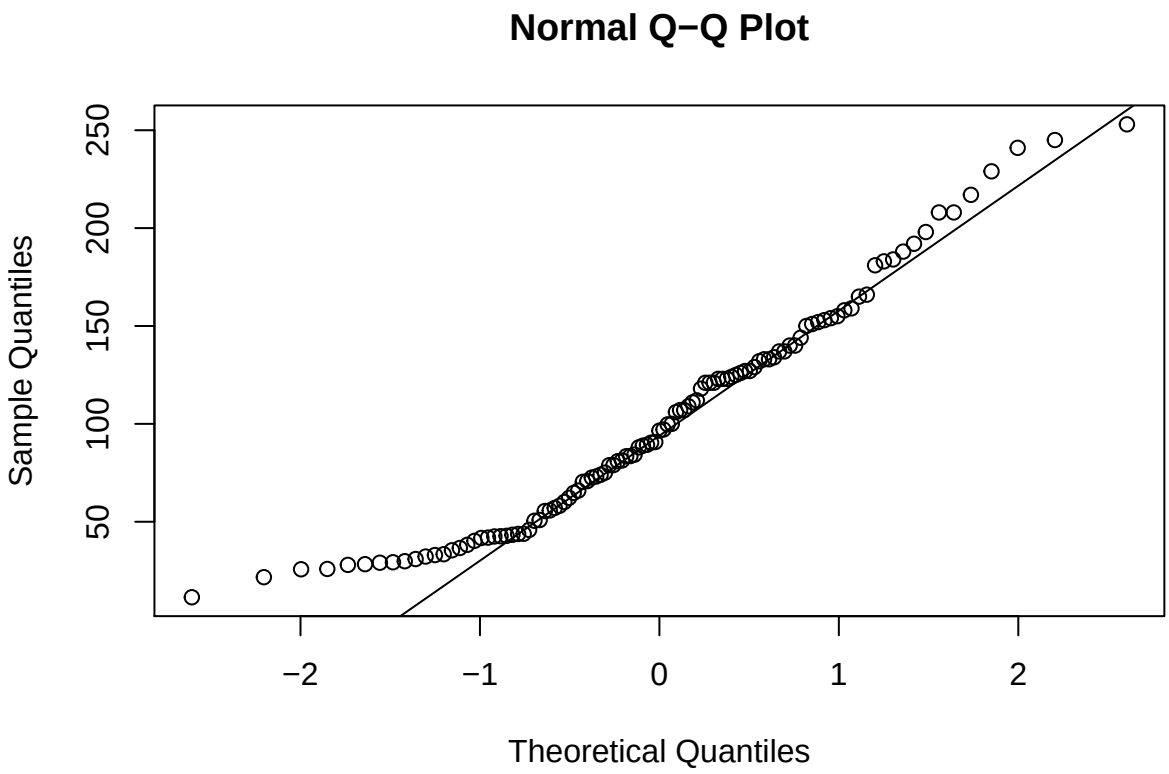
```
## to check the data is normal or not
qqnorm(wc$Waist)
qqline(wc$Waist)
```



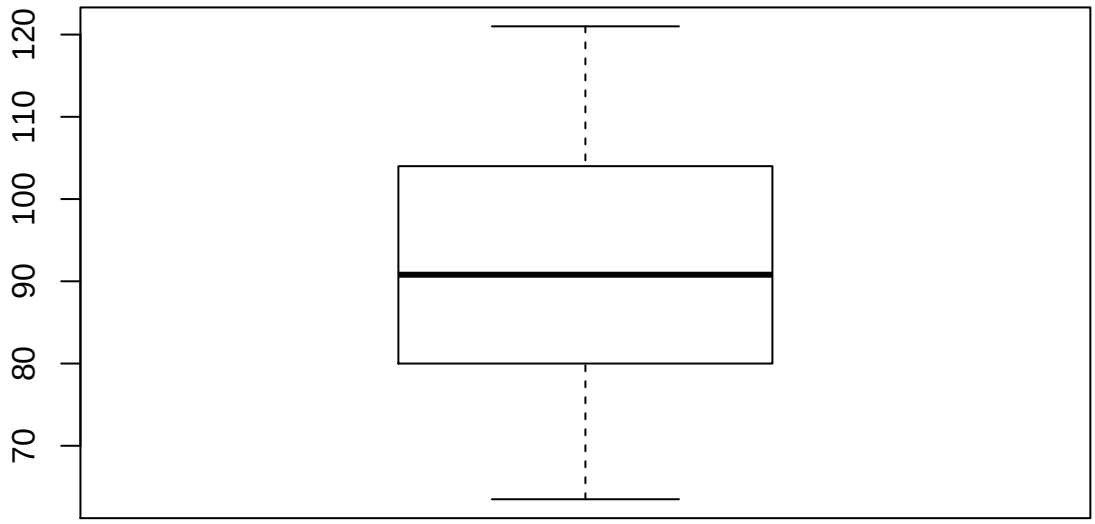

```
# to make more normal
log_wc_waist<-log(wc$Waist)
qqnorm(log_wc_waist)
qqline(log_wc_waist)
```



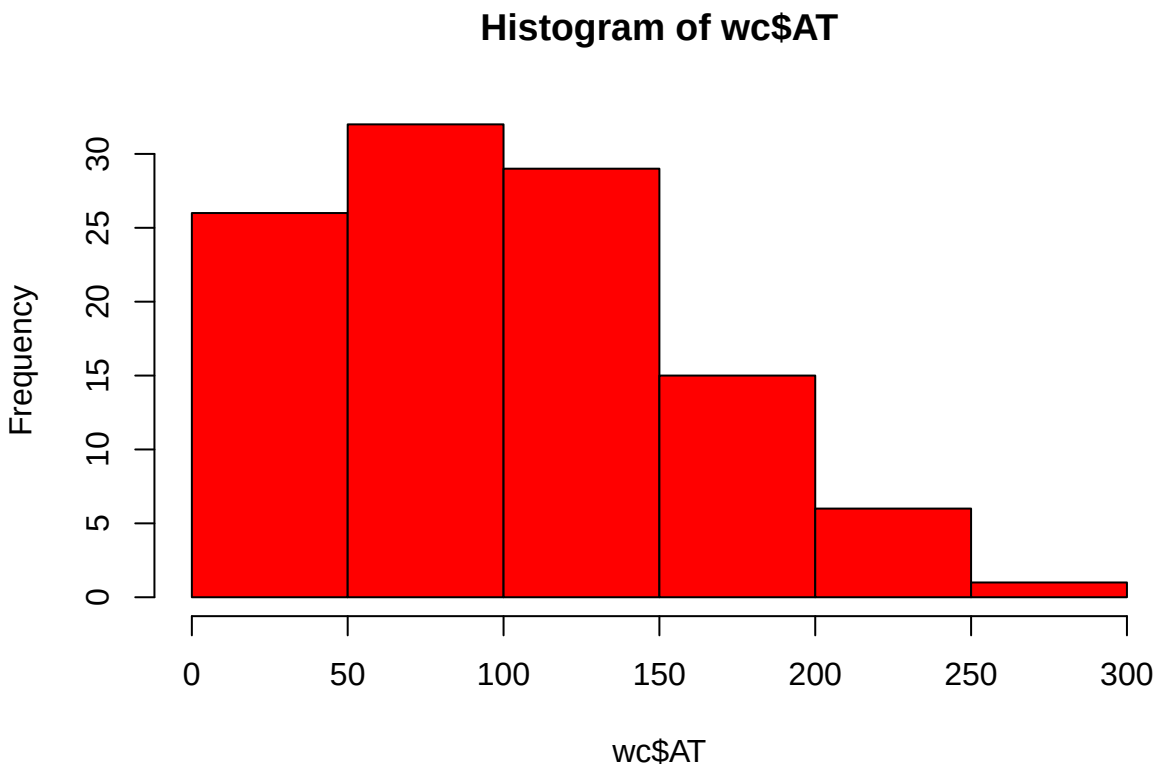

```
# the below gieven figure look normal data
qqnorm(wc$AT)
qqline(wc$AT)
```




```
# without take long its look normal
boxplot(wc$Waist)
```



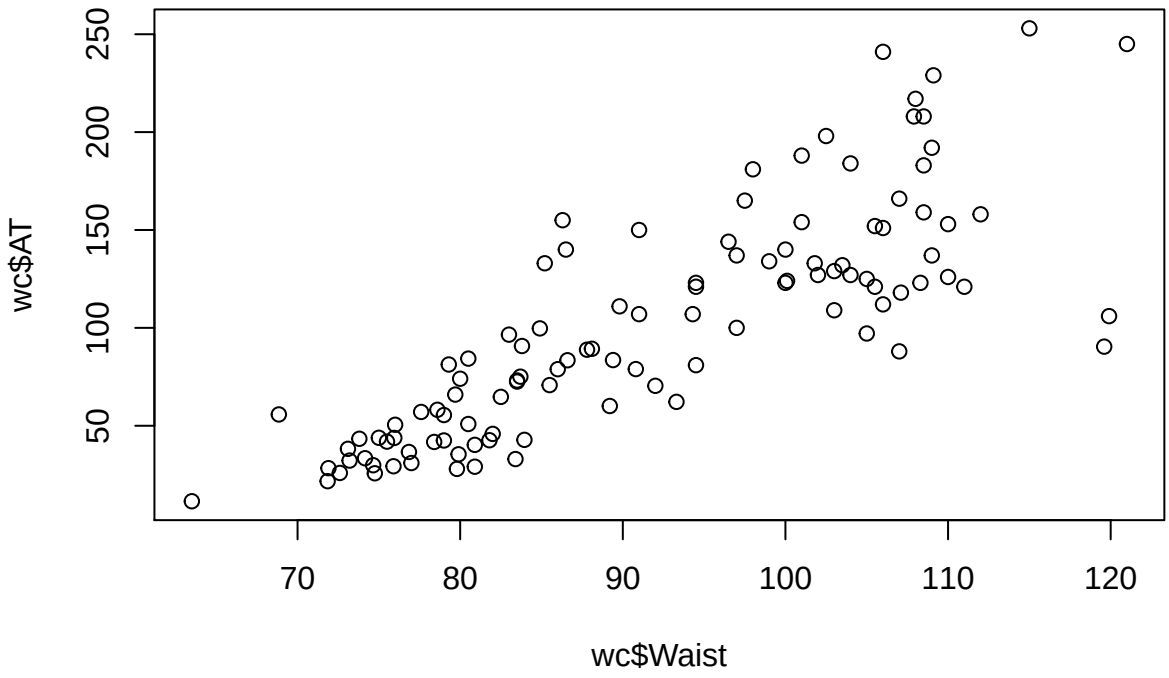

```
hist(wc$AT,col="red")
```



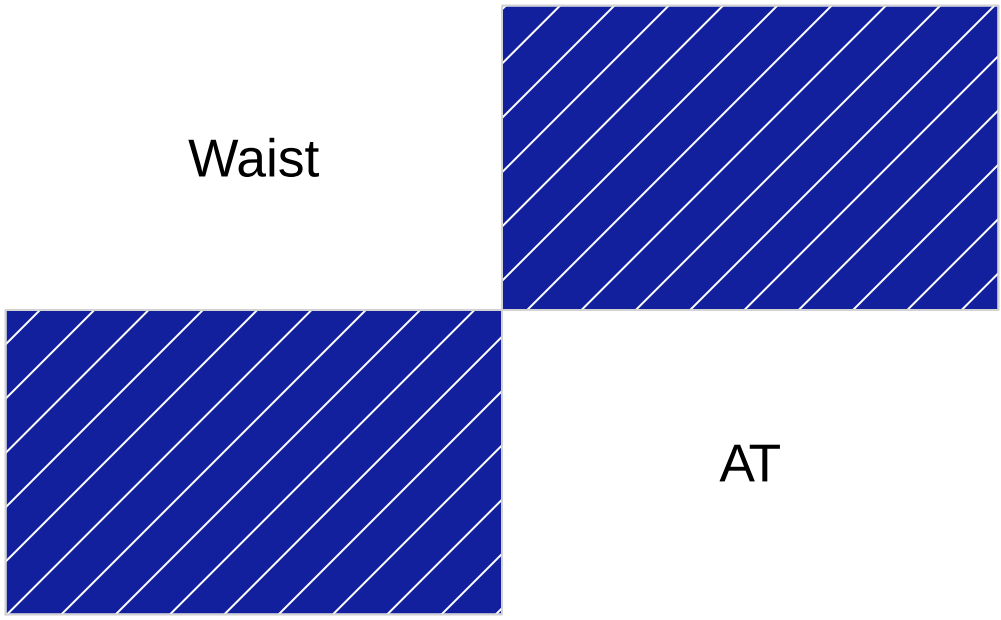
now buliding simple regression model

```
plot(wc$Waist,wc$AT)
# data strong postive correlation
library(corrgram)
```

```
## Warning: package 'corrgram' was built under R version 3.3.3
```




```
corrgram(wc)
```




```
# to check the coorelation value
cor(wc)
```

```
##           Waist        AT
## Waist 1.0000000 0.8185578
## AT    0.8185578 1.0000000
```

```
# r vlaue is .81 is good correlation

# now buliding model

mod<-lm(wc$AT~wc$Waist)
summary(mod)
```

```
##
## Call:
## lm(formula = wc$AT ~ wc$Waist)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -107.288  -19.143   -2.939   16.376   90.342
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -215.9815    21.7963  -9.909   <2e-16 ***
## wc$Waist       3.4589     0.2347  14.740   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.06 on 107 degrees of freedom
## Multiple R-squared:   0.67,  Adjusted R-squared:  0.667
## F-statistic: 217.3 on 1 and 107 DF,  p-value: < 2.2e-16
```

```
#other model with log value of wast
cor(log_wc_waist,wc$AT)
```

```
## [1] 0.8217782
```

```
mod2<-lm(wc$AT~log_wc_waist)
summary(mod2)
```

```
##
## Call:
## lm(formula = wc$AT ~ log_wc_waist)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -98.473 -18.273  -2.374  14.538  90.400
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1328.34      95.92  -13.85   <2e-16 ***
## log_wc_waist   317.14      21.26   14.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.8 on 107 degrees of freedom
## Multiple R-squared:  0.6753, Adjusted R-squared:  0.6723
## F-statistic: 222.6 on 1 and 107 DF,  p-value: < 2.2e-16
```

```
# to come with cofidence interval
confint(mod,level = 0.95)
```

```
##                   2.5 %     97.5 %
## (Intercept) -259.190053 -172.77292
## wc$Waist       2.993689    3.92403
```

```
# the prediction interval
predict(mod,interval="predict")
```

```
## Warning in predict.lm(mod, interval = "predict"): predictions on current data refer to _future_ respo
```

```
##            fit         lwr       upr
## 1    42.568252 -23.7607107 108.89721
## 2    35.131704 -31.3249765 101.58838
## 3    66.953210   0.9383962 132.96802
## 4    74.389758   8.4385892 140.34093
## 5    42.222366 -24.1122081 108.55694
## 6    32.537559 -33.9671546  99.04227
## 7    63.840237  -2.2056980 129.88617
## 8    72.487385   6.5213726 138.45340
## 9     3.656083 -63.5036005  70.81577
## 10   37.207020 -29.2125284 103.62657
## 11   32.710502 -33.7909536  99.21196
## 12   43.432966 -22.8821078 109.74804
```

```
## 13   36.861134 -29.5645231 103.28679
## 14   57.268404  -8.8518878 123.38870
## 15   50.350685 -15.8605336 116.56190
## 16   22.160981 -44.5537679  88.87573
## 17   46.718883 -19.5452517 112.98302
## 18   40.492936 -25.8701771 106.85605
## 19   39.282335 -27.1012331 105.66590
## 20   46.545940 -19.7208032 112.81268
## 21   49.831856 -16.3867039 116.05042
## 22   63.840237  -2.2056980 129.88617
## 23   60.381377  -5.7022296 126.46498
## 24   92.548770  26.6894200 158.40812
## 25   67.644982   1.6367253 133.65324
## 26  102.233576  36.3862036 168.08095
## 27   83.555735  17.6622091 149.44926
## 28   62.456693  -3.6039202 128.51731
## 29   81.480420  15.5758571 147.38498
## 30   69.374412   3.3819768 135.36685
## 31   72.833271   6.8700310 138.79651
## 32   88.744024  22.8729233 154.61513
## 33   98.082945  32.2335934 163.93230
## 34   93.240542  27.3829016 159.09818
## 35  136.822170  70.8074775 202.83686
## 36  110.880725  45.0222774 176.73917
## 37   98.774717  32.9260237 164.62341
## 38  140.281029  74.2316072 206.33045
## 39   60.727263  -5.3524301 126.80696
## 40   57.268404  -8.8518878 123.38870
## 41   72.833271   6.8700310 138.79651
## 42   46.891826 -19.3697083 113.15336
## 43   62.456693  -3.6039202 128.51731
## 44   83.209849  17.3145658 149.10513
## 45   71.103842   5.1264122 137.08127
## 46  154.462353  88.2365608 220.68815
## 47  110.188953  44.3321471 176.04576
## 48  110.880725  45.0222774 176.73917
## 49   59.689606  -6.4019262 125.78114
## 50   58.306062  -7.8017094 124.41383
## 51   94.624085  28.7694706 160.47870
## 52   73.870929   7.9158100 139.82605
## 53   78.713332  12.7922191 144.63445
## 54   45.162396 -21.1255054 111.45030
## 55   55.193088 -10.9531208 121.33930
## 56   55.884860 -10.2525800 122.02230
## 57   87.706367  21.8313711 153.58136
## 58   82.518078  16.6191807 148.41697
## 59   79.750990  13.8363291 145.66565
## 60   73.525043   7.5672497 139.48284
## 61   52.426001 -13.7565798 118.60858
## 62   77.675674  11.7478144 143.60353
## 63   60.035492  -6.0520617 126.12304
## 64  158.612984  92.3252791 224.90069
## 65  197.698095 130.6020356 264.79416
## 66  198.735753 131.6127559 265.85875
```

```
## 67  117.798443  51.9163563 183.68053
## 68  148.928178  82.7776990 215.07866
## 69  147.198748  81.0701043 213.32739
## 70  154.116467  87.8956245 220.33731
## 71  154.116467  87.8956245 220.33731
## 72  133.363311  67.3800865 199.34653
## 73  119.527873  53.6378248 185.41792
## 74  129.904451  63.9494297 195.85947
## 75  157.575326  91.3035349 223.84712
## 76  129.904451  63.9494297 195.85947
## 77  140.281029  74.2316072 206.33045
## 78  143.739889  77.6524810 209.82730
## 79  150.657608  84.4844833 216.83073
## 80  161.034186  94.7082219 227.36015
## 81  142.010459  75.9424508 208.07847
## 82  164.493045  98.1096934 230.87640
## 83  164.493045  98.1096934 230.87640
## 84  171.410764 104.9030239 237.91850
## 85  159.304756  93.0062808 225.60323
## 86  143.739889  77.6524810 209.82730
## 87  167.951905 101.5079578 234.39585
## 88  159.304756  93.0062808 225.60323
## 89  202.540498 135.3163441 269.76465
## 90  161.034186  94.7082219 227.36015
## 91  121.257303  55.3584733 187.15613
## 92  148.928178  82.7776990 215.07866
## 93  122.986732  57.0783023 188.89516
## 94  110.880725  45.0222774 176.73917
## 95  119.527873  53.6378248 185.41792
## 96  147.198748  81.0701043 213.32739
## 97  150.657608  84.4844833 216.83073
## 98  126.445592  60.5155029 192.37568
## 99   98.774717  32.9260237 164.62341
## 100 138.551600  72.5199497 204.58325
## 101 150.657608  84.4844833 216.83073
## 102 161.380072  95.0485136 227.71163
## 103 181.787342 115.0691257 248.50556
## 104 133.363311  67.3800865 199.34653
## 105 130.250337  64.2926425 196.20803
## 106 106.730093  40.8795247 172.58066
## 107 136.130398  70.1222603 202.13854
## 108 157.229440  90.9628890 223.49599
## 109 159.304756  93.0062808 225.60323
```