Calculating solar Panels

jpd_18

2023-05-15

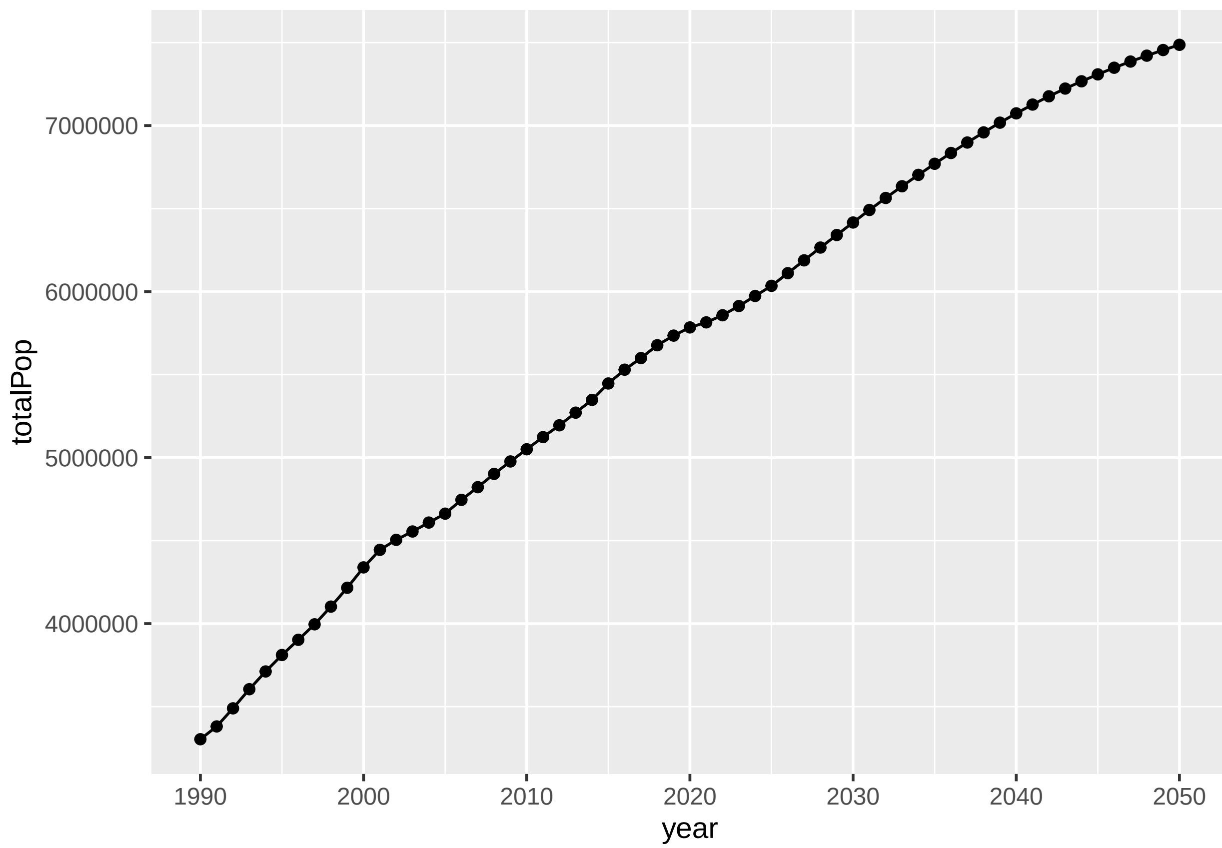
Population Projection

```r
col_pop <- read.csv('Population_Projections_in_Colorado.csv')
```

Visualizing the population trend in Colorado.

```r
#subset colorado population dataset getting total population by year.
col_pop2 <- col_pop %>%
  group_by(year) %>%
  summarize(totalPop = sum(totalPopulation))

#Visualizing the population trend in Colorado
ggplot(col_pop2, aes(year, totalPop)) +
  geom_line() +
  geom_point() +
  scale_x_continuous(breaks = seq(1990, 2050, by = 10))
```



In the dataset, the projected number of population in 2040 is 7073429.

The Number of population in 2020 is 5784140.

Solar Energy Generation from the residential level

```
## Generation Data
generation <- read.csv("electricity_generation.csv")
```

Cleaning Data

```
# get all rows where fuelTypeDescription contains "solar"
solar_ind <- grep("solar", generation$fuelTypeDescription, value = FALSE)
subset_solar <- generation[solar_ind, ]

# filter for only 2020 and sectorDescription == "Residential"
generation_2020 <-  subset_solar %>%
  select(period, sectorid, sectorDescription,
         fueltypeid, fuelTypeDescription, generation) %>%
```

```
  mutate(period = as.yearmon(period),
         year = year(period)) %>%
  filter(year == 2020,
         sectorDescription == "Residential",
         fuelTypeDescription == "estimated total solar")

# fuelTypeDescription is the same for "estimated total solar",
# "estimated total solar photovoltaic" and "estimated small scale solar photovoltaic"

# total generated solar electricity for 2020 in thousand megawatthours
total_solar <- sum(generation_2020$generation)

# convert to kilowatthours (kwh)
total_solar <- (total_solar*1000) * 1000
```

473,356,690 kwh of solar energy was generated in 2020.

Residential Consumption: Predict residential consumption by 2040

```
tot_con_end_user <- read_excel("Totalconsumption by end-user.xlsx",
                               sheet="Residential Sector", skip=2)
```

Data Cleaning

```
# get only colorado
tot_con_end_user <- tot_con_end_user %>%
  filter(State=="CO") %>%
  pivot_longer(cols = as.character(c(1960:2020)),
               names_to = "year",
               values_to = "total_energy_consumption")
tot_con_end_user$year <- as.numeric(tot_con_end_user$year)
```

Predictive Machine Learning: Polynomial Regression Model

```
# split into training and test set
set.seed(123)
trainIndex <- round(nrow(tot_con_end_user)*0.8)
trainData <- tot_con_end_user[1:trainIndex, ]
testData <- tot_con_end_user[(1+trainIndex):nrow(tot_con_end_user), ]
```

Run loop to get best polynomial model

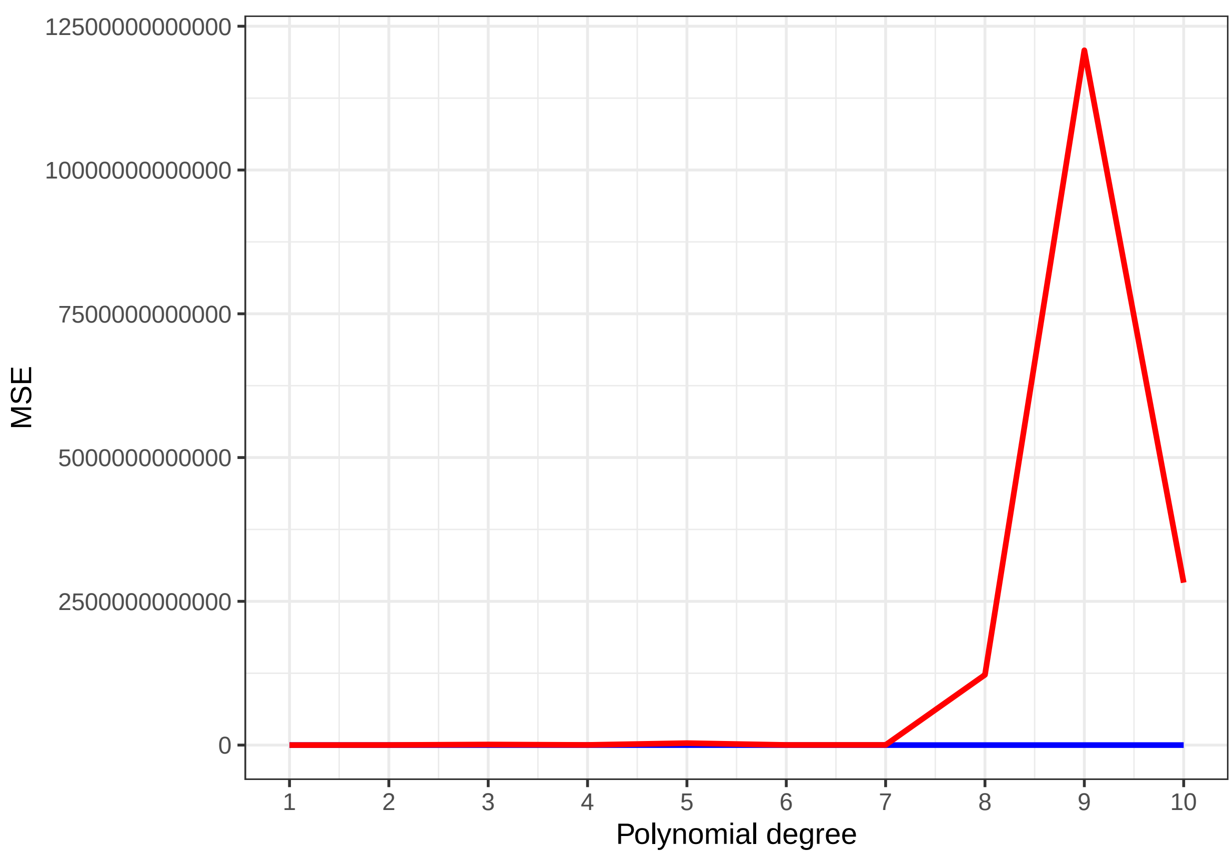
```r
degree=10
test_mses <- training_mses <- c()
for (i in 1:degree) {
  poly_model = lm(total_energy_consumption ~ poly(year,i), data=trainData)
  y_hat_training = predict(poly_model)
  training_mses[i] <- mean((trainData$total_energy_consumption - y_hat_training)^2)

  y_hat_test <- predict(poly_model, newdata = testData)
  test_mses[i] <- mean((testData$total_energy_consumption - y_hat_test)^2)

  mses <- data.frame(degree = 1:i, training_error = training_mses, test_error =
↪  test_mses)
}

# Plot the errors
ggplot(mses, aes(x = degree)) +
  geom_line(aes(y = training_error), color = "blue", size = 1) +
  geom_line(aes(y = test_error), color = "red", size = 1) +
  labs(x = "Polynomial degree", y = "MSE") +
  scale_x_continuous(breaks = 1:degree) +
  theme_bw()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```



```r
# Find the degree with the lowest test error
best_degree <- which.min(test_mses)

# Print the best degree and its corresponding test error
cat("Best degree:", best_degree, "\n")
```

```
## Best degree: 1
```

```r
#Best degree: 1
cat("Test error:", test_mses[best_degree], "\n")
```

```
## Test error: 263579175
```

```r
#Test error: 263579175
```

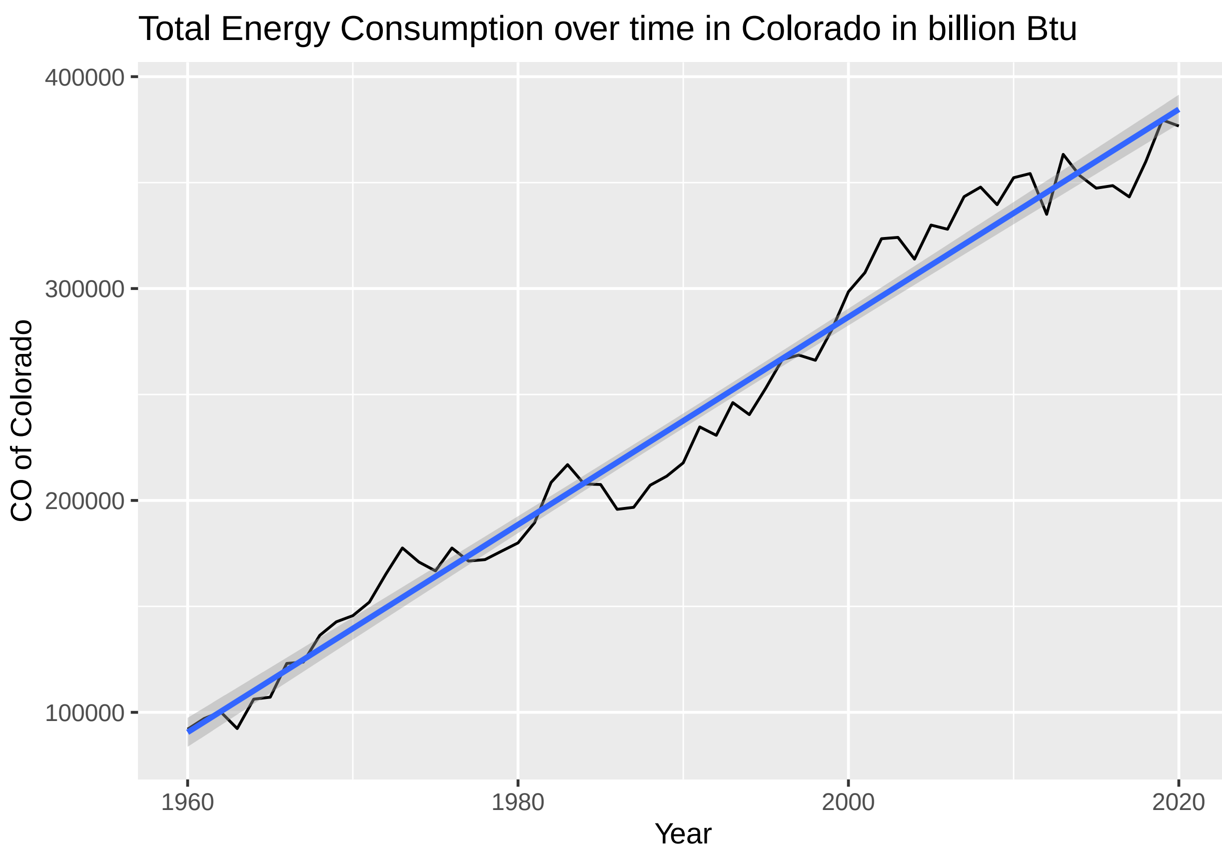
The best degree of polynomial model is 1, and the test error is 263579175.

Fit best polynomial model

```r
best_model <- lm(total_energy_consumption ~ poly(year,1, raw=T), data = tot_con_end_user)
summary(best_model)
```

```
##
## Call:
## lm(formula = total_energy_consumption ~ poly(year, 1, raw = T),
##     data = tot_con_end_user)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26629.4  -8611.1   -443.7  10123.5  27135.0
##
## Coefficients:
##                           Estimate  Std. Error t value            Pr(>|t|)
## (Intercept)            -9513869.27   195441.99  -48.68 <0.0000000000000002 ***
## poly(year, 1, raw = T)     4900.23       98.21   49.90 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13500 on 59 degrees of freedom
## Multiple R-squared:  0.9769, Adjusted R-squared:  0.9765
## F-statistic:  2490 on 1 and 59 DF,  p-value: < 0.00000000000000022
```

```r
p <- ggplot(tot_con_end_user, aes(x=year, y = total_energy_consumption)) +
          geom_line() +
          stat_smooth(method='lm', formula = y ~ poly(x,1), size = 1) +
          xlab('Year') +
          ylab('CO of Colorado') + labs(title = 'Total Energy Consumption over time in
          ↪  Colorado in billion Btu')
p
```

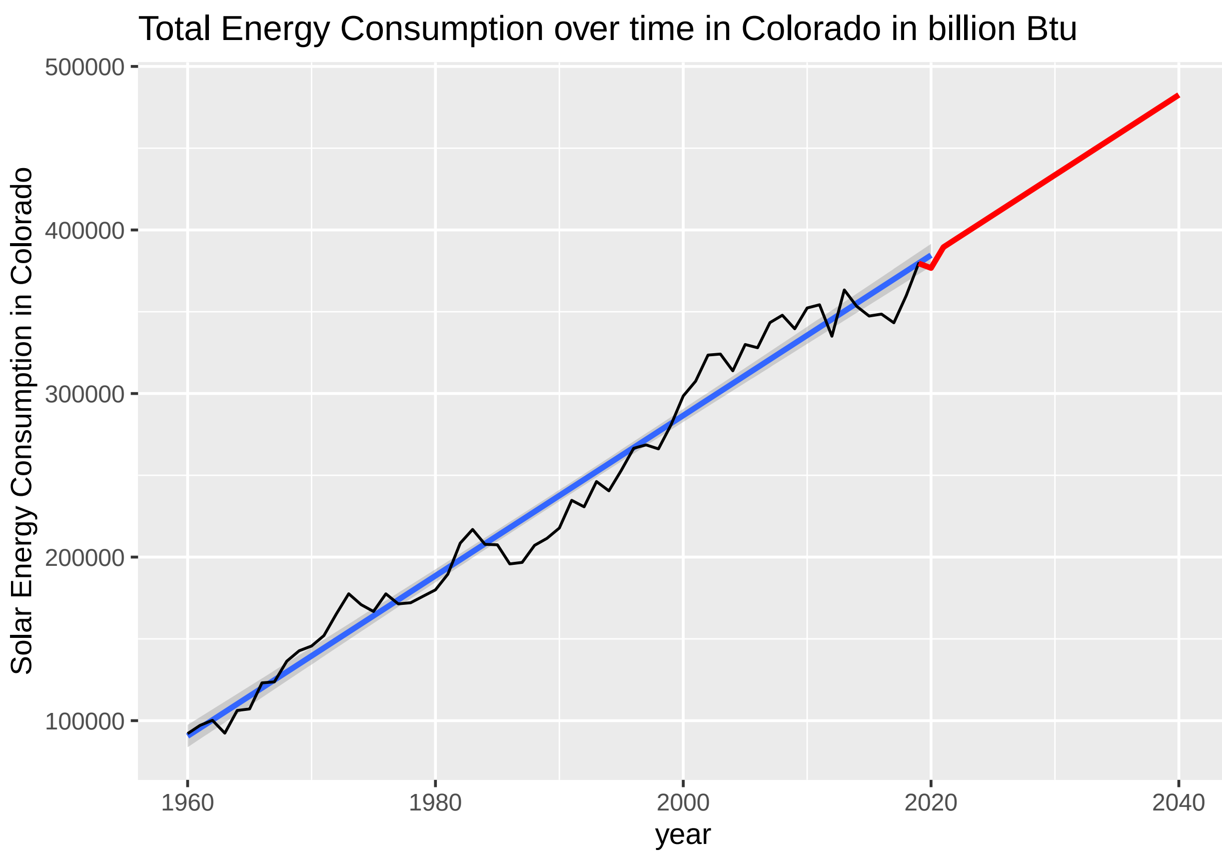


Predicting future years(2021~ 2040)

```r
# Create data from future years
future_years <- data.frame(year=c(2021:2040))
future_predictions <- predict(best_model, newdata = future_years)
newData <- cbind(year=future_years, total_energy_consumption=future_predictions)

# Pick important column & Combine past and future datasets
tot_con_end_user <- tot_con_end_user[ ,c("year","total_energy_consumption")]
pred_totalcounsmption <- rbind(tot_con_end_user, newData)

# Plot data
ggplot(head(pred_totalcounsmption, 61), aes(x = year, y = total_energy_consumption)) +
  stat_smooth(method='lm', formula = y ~ poly(x,1), size = 1) +
  geom_line(color = "black") +
  geom_line(data = tail(pred_totalcounsmption, 22), size = 1, color = 'red') +
  ylab("Solar Energy Consumption in Colorado") + labs(title = 'Total Energy Consumption
  ↪  over time in Colorado in billion Btu')
```



Calculating number of solar panels

Panels needed to offset coal energy demand in 2019.(38% of electricity demand)

```
## Assuming a peak daily sunlight duration of 6 hours and a panel power rating of 300
↪  watts

#2019
#consumption 379579.0 billion btu. (1 BTU = 0.000293071  kilowatt-hours)
population<-5734950

consumption_yearly = 379579 * 1000000000 *0.000293071  #kwh
con_coal<- consumption_yearly*0.38 # Coal energy usage

# 1.Convert the yearly energy requirement to daily energy requirement:
consumption_daily= con_coal / 365
consumption_daily #kwh per day
```

```
## [1] 115815252
```

```
# 2. Convert the daily energy requirement to watt-hours
consumption_daily<- consumption_daily*1000

# 3. Determine the energy produced per day by each solar panel: 300 × 6 hours = 1800
↪  watt-hours.

# Calculate the number of solar panels needed:
panels<-consumption_daily/1800
panels
```

```
## [1] 64341807
```

```
#Calculate the number of solar panels needed per person
panels_person<- panels/population
panels_person #11.21924
```

```
## [1] 11.21924
```

11.21924 solar panels are needed per person to offset the coal energy usage in 2019.

Merged dataset: population, coal energy demand, coal usage amount per person(lbs), number panels to eliminate coal energy

```
# merge population(1990-2040) and total residential consumption
merged_df <- merge(col_pop2, pred_totalcounsmption, by = "year")

# convert into kwh
merged_df$consup_kwh <- merged_df$total_energy_consumption * 1000000000 * 0.000293071
# coal energy usage amount
merged_df$consup_kwh_coal <- merged_df$consup_kwh * 0.38
# coal energy usage per day
merged_df$consup_kwh_coal_day <- merged_df$consup_kwh_coal / 365

# coal usage amount per person(lbs)
merged_df$consup_watt_coal_day <- (merged_df$consup_kwh_coal_day * 1000)/1800
merged_df$consup_watt_coal_day_lbs <- merged_df$consup_kwh_coal_day / 6
merged_df$consup_watt_coal_day_lbs_per_person <-
↪  merged_df$consup_watt_coal_day_lbs/merged_df$totalPop

# number of panels to remove coal energy
merged_df$no_solar_panels <- merged_df$consup_watt_coal_day / merged_df$totalPop

head(merged_df[47:51,])
```

```
##    year totalPop total_energy_consumption   consup_kwh consup_kwh_coal
## 47 2036  6834836                 462989.7 135688858569     51561766256
## 48 2037  6897868                 467889.9 137124972538     52107489564
## 49 2038  6958823                 472790.2 138561086506     52653212872
```

```
## 50 2039  7017524                477690.4 139997200475     53198936181
## 51 2040  7073429                482590.6 141433314444     53744659489
##    consup_kwh_coal_day consup_watt_coal_day consup_watt_coal_day_lbs
## 47           141265113             78480618                 23544186
## 48           142760245             79311247                 23793374
## 49           144255378             80141877                 24042563
## 50           145750510             80972506                 24291752
## 51           147245642             81803135                 24540940
##    consup_watt_coal_day_lbs_per_person no_solar_panels
## 47                            3.444733        11.48244
## 48                            3.449381        11.49794
## 49                            3.454975        11.51658
## 50                            3.461584        11.53861
## 51                            3.469455        11.56485
```

Panels needed to offset different proportions of energy demand in 2040 (50, 60, 70, 80, 90, 100%)

```
#2040
#We predicted consumption 482590.6 billion btu
population <- 7073429
consumption_yearly = 482590.6 * 1000000000 *0.000293071  #kwh
panels_person<- c()
values<- c(0.5,0.6,0.7,0.8,0.9,1)

for(i in values){
con_coal<- consumption_yearly*i
# 1.Convert the yearly energy requirement to daily energy requirement:
consumption_daily= con_coal / 365
consumption_daily #kwh per day

# 2. Convert the daily energy requirement to watt-hours
consumption_daily<- consumption_daily*1000

# 3. Determine the energy produced per day by each solar panel: 300 × 6 hours = 1800
↪  watt-hours.

# Calculate the number of solar panels needed:
panels<-consumption_daily/1800
panels

#Calculate the number of solar panels needed per person
result<- panels/population

panels_person<- c(panels_person,result)

}
panels_person_df<-data.frame(cbind(values,panels_person))
head(panels_person_df)
```

```
##   values panels_person
```

```
## 1    0.5       15.21691
## 2    0.6       18.26029
## 3    0.7       21.30367
## 4    0.8       24.34705
## 5    0.9       27.39043
## 6    1.0       30.43381
```

In 2040, 15.22 panels are needed to offset 50% of energy demand, and 30.43 panels for 100%.