Regression_stat

Ayush Pandey

2022-12-07

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#loading dataset
data=read_csv("temp_country_2020.csv")
```

```
## Rows: 54720 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Country
## dbl  (3): Month, Year, Temp
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(data)
```

```
## # A tibble: 6 x 5
##   Date       Country   Month  Year   Temp
##   <date>     <chr>     <dbl> <dbl>  <dbl>
## 1 1900-01-01 Argentina     1  1900 21.8
## 2 1900-01-01 Austria       1  1900 -2.28
## 3 1900-01-01 Bahrain       1  1900 15.0
## 4 1900-01-01 Belarus       1  1900 -6.78
## 5 1900-01-01 Belgium       1  1900  2.87
## 6 1900-01-01 Bulgaria      1  1900  0.814
```

```
unique((data %>% filter(Year == 2000))['Date']) # Shows that there is only monthly data available not d
```

```
## # A tibble: 12 x 1
##    Date
##    <date>
##  1 2000-01-01
##  2 2000-02-01
```

```
##  3 2000-03-01
##  4 2000-04-01
##  5 2000-05-01
##  6 2000-06-01
##  7 2000-07-01
##  8 2000-08-01
##  9 2000-09-01
## 10 2000-10-01
## 11 2000-11-01
## 12 2000-12-01
```

```
#summary of the dataset
summary(data)
```

```
##       Date               Country              Month            Year
##  Min.   :1900-01-01   Length:54720       Min.   : 1.00   Min.   :1900
##  1st Qu.:1929-12-24   Class :character   1st Qu.: 3.75   1st Qu.:1930
##  Median :1959-12-16   Mode  :character   Median : 6.50   Median :1960
##  Mean   :1959-12-16                      Mean   : 6.50   Mean   :1960
##  3rd Qu.:1989-12-08                      3rd Qu.: 9.25   3rd Qu.:1989
##  Max.   :2019-12-01                      Max.   :12.00   Max.   :2019
##       Temp
##  Min.   :-59.324
##  1st Qu.:  7.212
##  Median : 15.413
##  Mean   : 14.937
##  3rd Qu.: 22.689
##  Max.   : 40.443
```

```
#checking for NA values
nadate=sum(is.na(data["Date"]))
nacountry=sum(is.na(data["Country"]))
namonth=sum(is.na(data["Month"]))
nayear=sum(is.na(data["Year"]))
natemp=sum(is.na(data["Temp"]))
namonth
```

```
## [1] 0
```

```
nayear
```

```
## [1] 0
```

```
natemp
```

```
## [1] 0
```

```
nacountry
```

```
## [1] 0
```

```
nadate
```

```
## [1] 0
```

```
#### Above data values show that the table is clean and not needing any more cleaning.

### Step 2: Gather some country specific info for last century 1900-2019 and add columns. To assess tre

# Gather some country specific info for last century
fun <- function(x) {
  if (x < 10) {
    "Low"
  }
  else if (x < 20) {
    "Medium"
  }
  else {
    "High"
  }
}

mean_countries <- data %>% group_by(Country) %>%
  summarize(mean_monthly_1900_2019 = mean(Temp),
            min_monthly_1900_2019 = min(Temp),
            max_monthly_1900_2019 = max(Temp),
            temp_category = factor(fun(mean_monthly_1900_2019), levels = c("Low", "Medium", "High")))

print('Low category countries')
```

```
## [1] "Low category countries"
```

```
print((mean_countries %>% filter(temp_category == "Low") %>% count())$n)
```

```
## [1] 13
```

```
print('Medium category countries')
```

```
## [1] "Medium category countries"
```

```
print((mean_countries %>% filter(temp_category == "Medium") %>% count())$n)
```

```
## [1] 14
```

```
print('High category countries')
```

```
## [1] "High category countries"
```

```r
print((mean_countries %>% filter(temp_category == "High") %>% count())$n)
```

```
## [1] 11
```

```r
head(mean_countries)
```

```
## # A tibble: 6 x 5
##   Country   mean_monthly_1900_2019 min_monthly_1900_2019 max_monthly_1~1 temp_~2
##   <chr>                      <dbl>                 <dbl>           <dbl> <fct>
## 1 Argentina                  14.9                   2.72            25.6 Medium
## 2 Austria                     6.74                -11.1             23.7 Low
## 3 Bahrain                    26.2                  12.2             37.5 High
## 4 Belarus                     5.88                -42.3             22.8 Low
## 5 Belgium                     9.83                 -6.08            22.8 Low
## 6 Bulgaria                   10.8                  -6.90            25.1 Medium
## # ... with abbreviated variable names 1: max_monthly_1900_2019,
## #   2: temp_category
```

```r
### Step 3: Load some info for each country and year such as annual min, max, range, mean

data_country <- data %>%
  group_by(Country, Year) %>%
  summarize(mean_monthly = mean(Temp),
            min_monthly = min(Temp),
            max_monthly = max(Temp))
```

```
## `summarise()` has grouped output by 'Country'. You can override using the
## `.groups` argument.
```

```r
data_country <- inner_join(data_country, mean_countries)
```

```
## Joining, by = "Country"
```

```r
head(data_country)
```

```
## # A tibble: 6 x 9
## # Groups:   Country [1]
##   Country    Year mean_monthly min_mon~1 max_m~2 mean_~3 min_m~4 max_m~5 temp_~6
##   <chr>     <dbl>        <dbl>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <fct>
## 1 Argentina  1900         14.8      8.74    21.8    14.9    2.72    25.6 Medium
## 2 Argentina  1901         14.8      7.84    20.7    14.9    2.72    25.6 Medium
## 3 Argentina  1902         14.6      6.91    21.5    14.9    2.72    25.6 Medium
## 4 Argentina  1903         14.4      7.53    20.6    14.9    2.72    25.6 Medium
## 5 Argentina  1904         14.4      8.51    20.8    14.9    2.72    25.6 Medium
## 6 Argentina  1905         14.2      6.75    20.4    14.9    2.72    25.6 Medium
## # ... with abbreviated variable names 1: min_monthly, 2: max_monthly,
## #   3: mean_monthly_1900_2019, 4: min_monthly_1900_2019,
## #   5: max_monthly_1900_2019, 6: temp_category
```

```r
# Let us find the country for maximum and min average temperatures in last years
temp_data <- data %>%
  group_by(Country) %>%
  summarize(mean_temp = mean(Temp))

temp_data[which.max(temp_data$mean_temp),]
```

```
## # A tibble: 1 x 2
##   Country              mean_temp
##   <chr>                    <dbl>
## 1 United Arab Emirates      27.9
```

```r
temp_data[which.min(temp_data$mean_temp),]
```

```
## # A tibble: 1 x 2
##   Country mean_temp
##   <chr>       <dbl>
## 1 Iceland      1.94
```

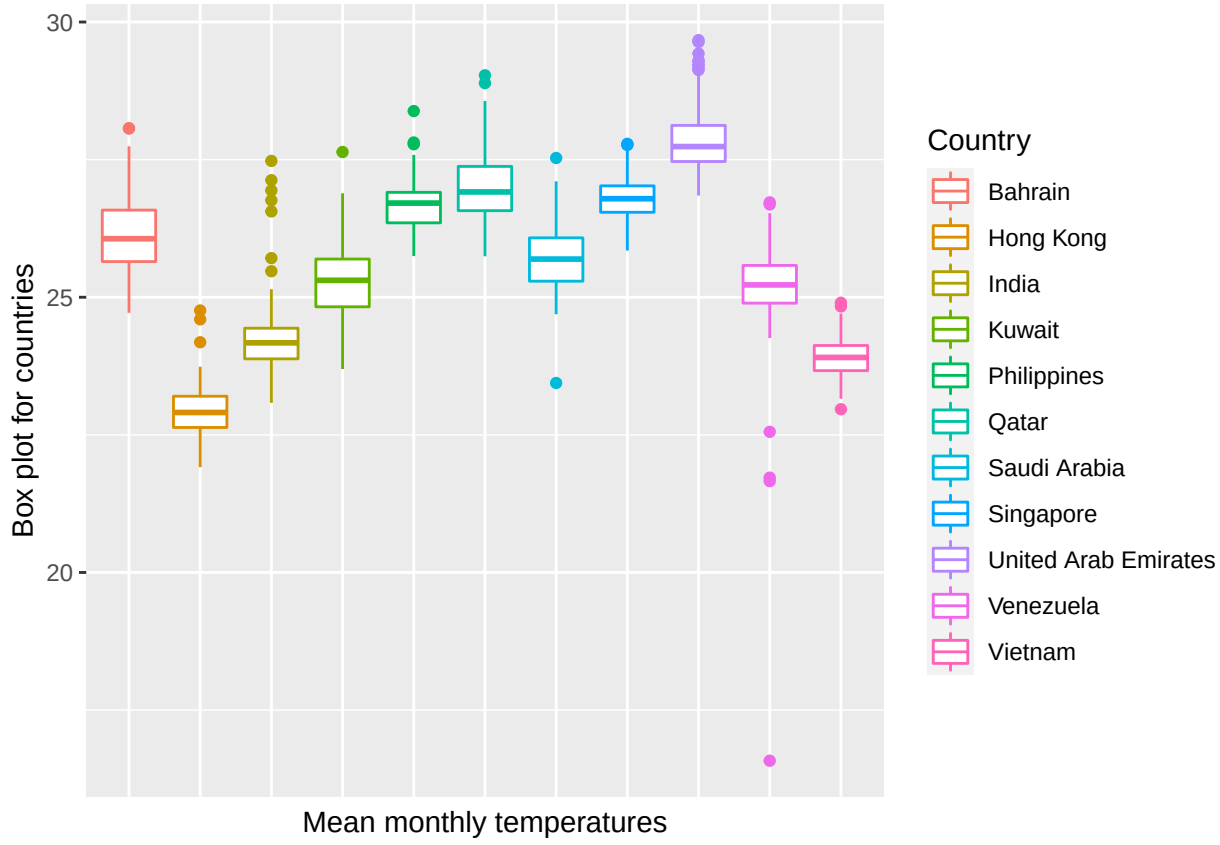
```r
uae_temp  <- data %>%
  filter(Country == 'United Arab Emirates') %>%
  group_by(Year) %>%
  summarize(mean_temp = mean(Temp))

iceland_temp  <- data %>% filter(Country == 'Iceland') %>%
  group_by(Year) %>%
  summarize(mean_temp = mean(Temp))

head(uae_temp)
```

```
## # A tibble: 6 x 2
##    Year mean_temp
##   <dbl>     <dbl>
## 1  1900      27.6
## 2  1901      27.6
## 3  1902      27.9
## 4  1903      27.1
## 5  1904      27.6
## 6  1905      27.5
```

```r
head(iceland_temp)
```

```
## # A tibble: 6 x 2
##    Year mean_temp
##   <dbl>     <dbl>
## 1  1900     1.36
## 2  1901     1.73
## 3  1902     0.720
## 4  1903     0.674
## 5  1904     1.43
## 6  1905     1.40
```

```
data_country %>%
  filter(temp_category == "High") %>%
  ggplot(aes(x = Country, y = mean_monthly, color = Country)) +
  geom_boxplot() +
  xlab("Mean monthly temperatures") +
  ylab("Box plot for countries") +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```




```
#### Above graph shows outliers for Venezeula but the values are not wrong as printed below.

data_venezuela <- data_country %>% filter(Country == 'Venezuela')
data_venezuela[which.min(data_venezuela$mean_monthly),]
```

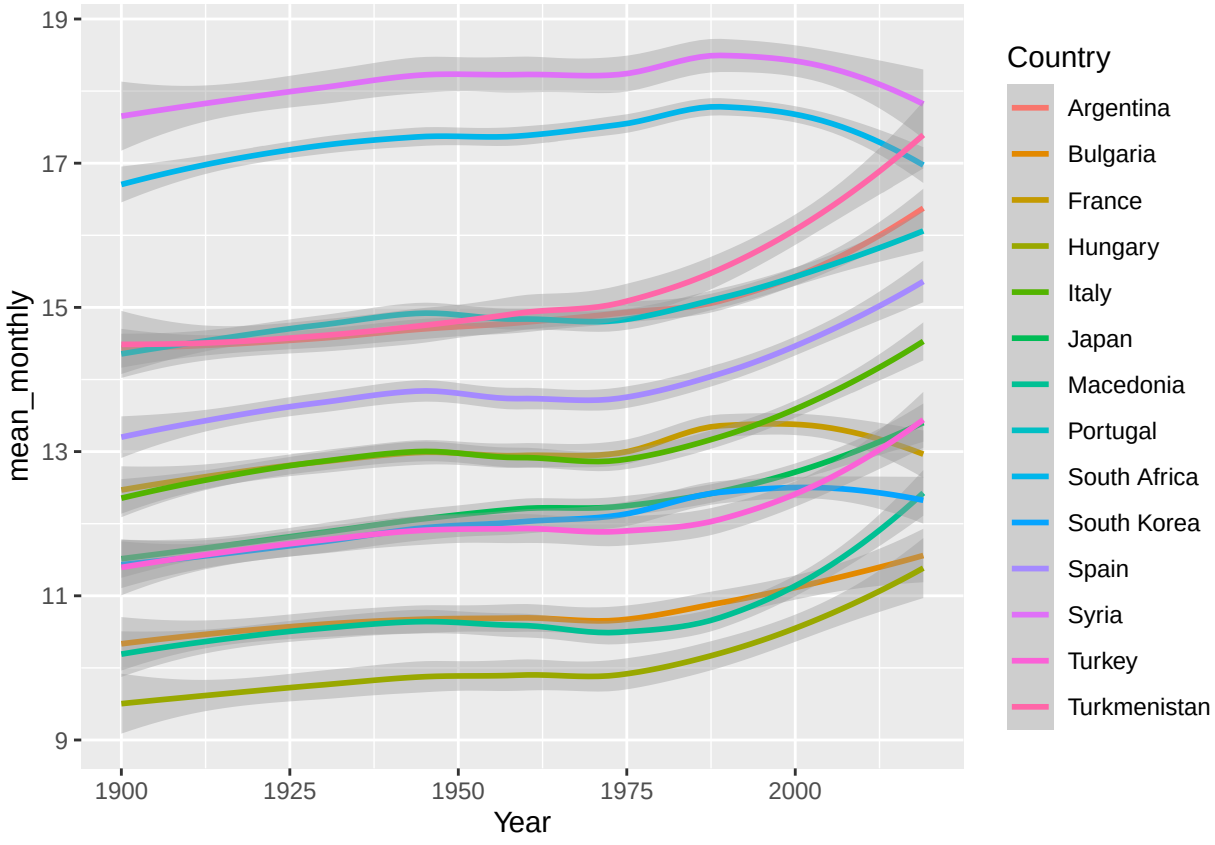
```
## # A tibble: 1 x 9
## # Groups:   Country [1]
##   Country    Year mean_monthly min_mon~1 max_m~2 mean_~3 min_m~4 max_m~5 temp_~6
##   <chr>     <dbl>        <dbl>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <fct>
## 1 Venezuela  2016         16.6     -59.3    28.5    25.1   -59.3    29.4 High
## # ... with abbreviated variable names 1: min_monthly, 2: max_monthly,
## #   3: mean_monthly_1900_2019, 4: min_monthly_1900_2019,
## #   5: max_monthly_1900_2019, 6: temp_category
```

```
data_venezuela[which.max(data_venezuela$mean_monthly),]
```

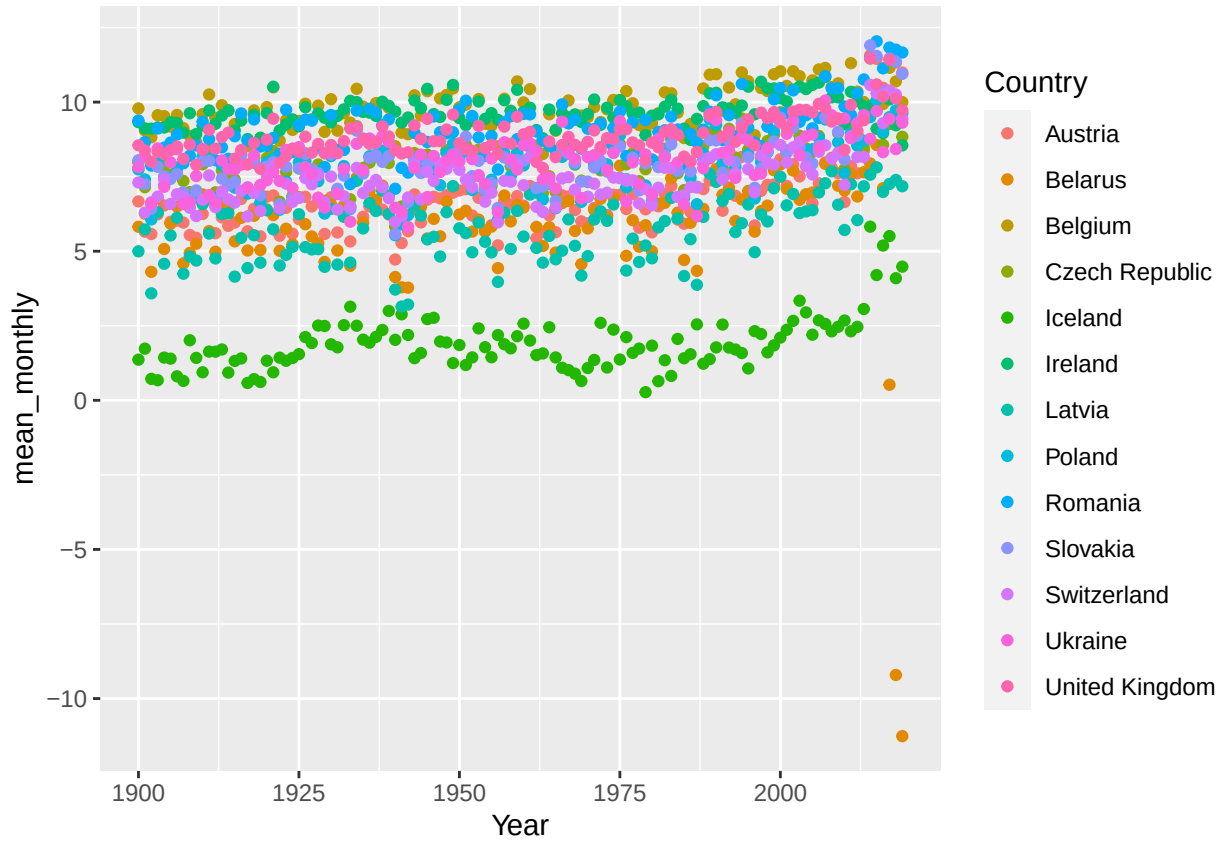
```
## # A tibble: 1 x 9
## # Groups:   Country [1]
##   Country    Year mean_monthly min_mon~1 max_m~2 mean_~3 min_m~4 max_m~5 temp_~6
##   <chr>     <dbl>        <dbl>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <fct>
## 1 Venezuela  2014         26.7      22.8    28.0    25.1   -59.3    29.4 High
## # ... with abbreviated variable names 1: min_monthly, 2: max_monthly,
## #   3: mean_monthly_1900_2019, 4: min_monthly_1900_2019,
## #   5: max_monthly_1900_2019, 6: temp_category
```

```
data_country %>%
  filter(temp_category == "Medium") %>%
  ggplot(aes(x = Year, y = mean_monthly, color = Country)) + geom_smooth()
```

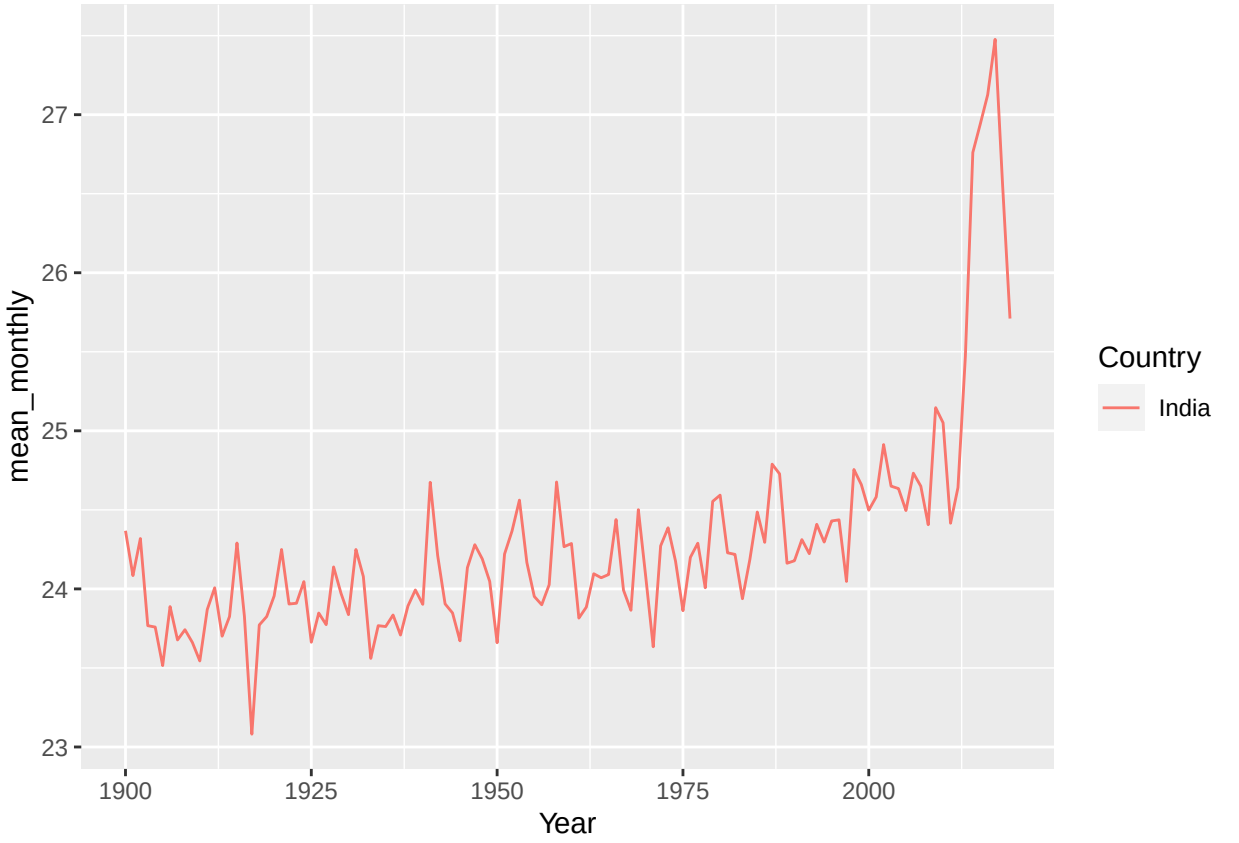
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
data_country %>%
  filter(temp_category == "Low") %>%
  ggplot(aes(x = Year, y = mean_monthly, color = Country)) + geom_point()
```




```
## Above 3 graphs shows that all category of countries based on temperature buckets are showing consist

### Let us now see for India

#Analysis for India, shows the increase in Average mean temperature over the years
data_country %>%
  filter(Country == "India") %>%
  ggplot(aes(x = Year, y = mean_monthly, color = Country)) + geom_line()
```




```
# Analysis for United Arab Emirates as it has the maximum mean temperature from 1900-2019

temp_data <- data %>%
  group_by(Country) %>%
  summarize(mean_temp = mean(Temp))
temp_data[which.max(temp_data$mean_temp),]
```

```
## # A tibble: 1 x 2
##   Country              mean_temp
##   <chr>                    <dbl>
## 1 United Arab Emirates      27.9
```

```
temp_data[which.min(temp_data$mean_temp),]
```

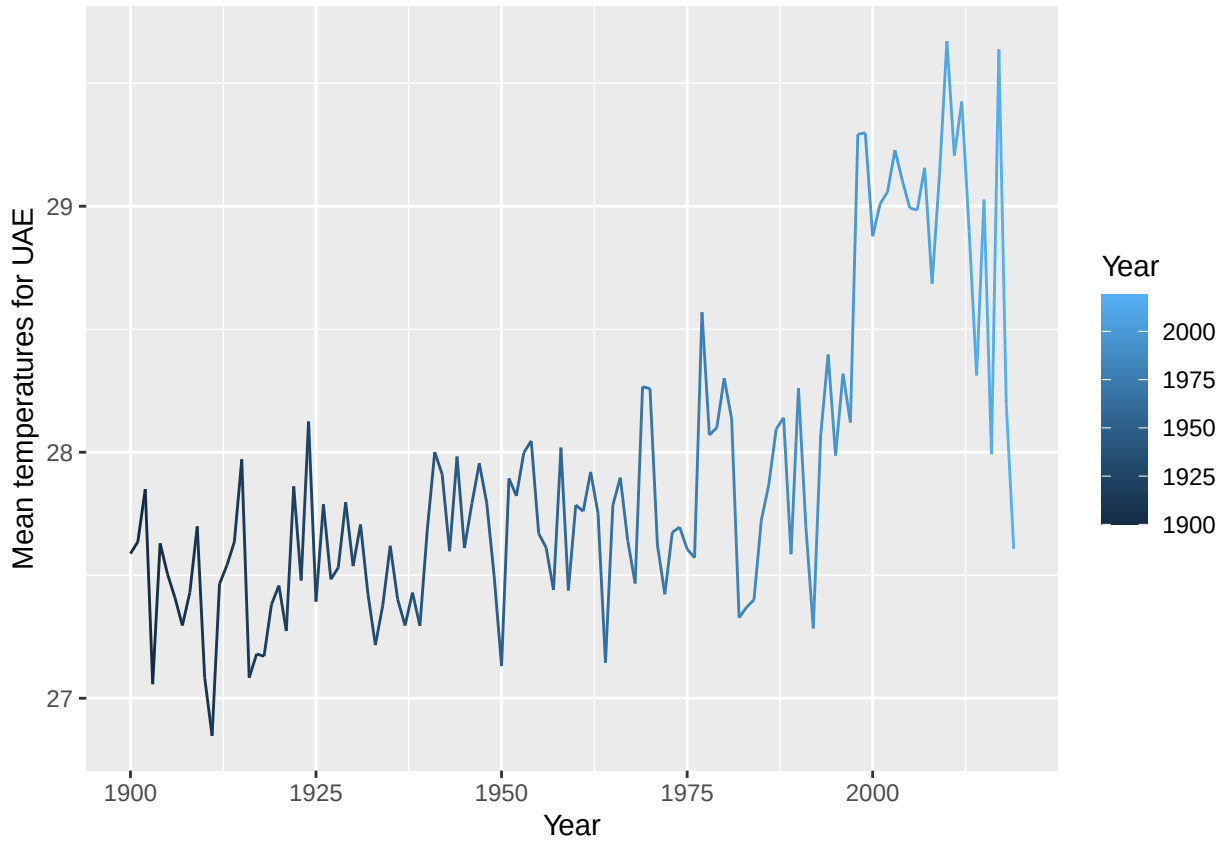
```
## # A tibble: 1 x 2
##   Country mean_temp
##   <chr>       <dbl>
## 1 Iceland      1.94
```

```
uae_temp  <- data %>%
  filter(Country == 'United Arab Emirates') %>%
  group_by(Year) %>%
  summarize(mean_temp = mean(Temp))
```

```
print(uae_temp)
```

```
## # A tibble: 120 x 2
##     Year mean_temp
##    <dbl>     <dbl>
##  1  1900      27.6
##  2  1901      27.6
##  3  1902      27.9
##  4  1903      27.1
##  5  1904      27.6
##  6  1905      27.5
##  7  1906      27.4
##  8  1907      27.3
##  9  1908      27.4
## 10  1909      27.7
## # ... with 110 more rows
## # i Use 'print(n = ...)' to see more rows
```

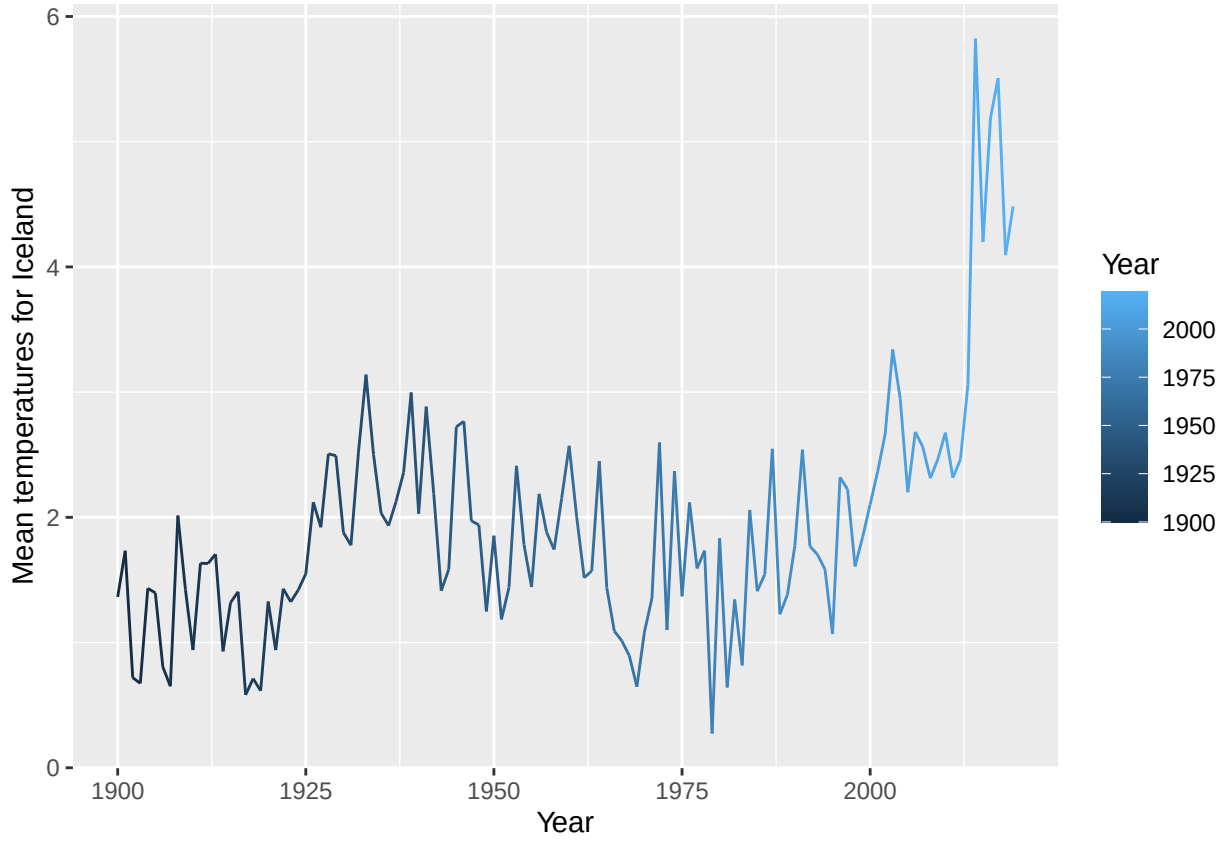
```
ggplot(data=uae_temp, aes(x=Year,y=mean_temp,color=Year)) + geom_line() + ylab("Mean temperatures for U
```




```
# Analysis for Iceland as it has the minimum mean temperature from 1900-2019

print(iceland_temp)
```

```
## # A tibble: 120 x 2
##     Year mean_temp
##    <dbl>     <dbl>
##  1  1900     1.36
##  2  1901     1.73
##  3  1902     0.720
##  4  1903     0.674
##  5  1904     1.43
##  6  1905     1.40
##  7  1906     0.806
##  8  1907     0.651
##  9  1908     2.01
## 10  1909     1.42
## # ... with 110 more rows
## # i Use 'print(n = ...)' to see more rows
```

```
ggplot(data=iceland_temp, aes(x=Year, y=mean_temp, color=Year)) + geom_line() + ylab("Mean temperatures
```



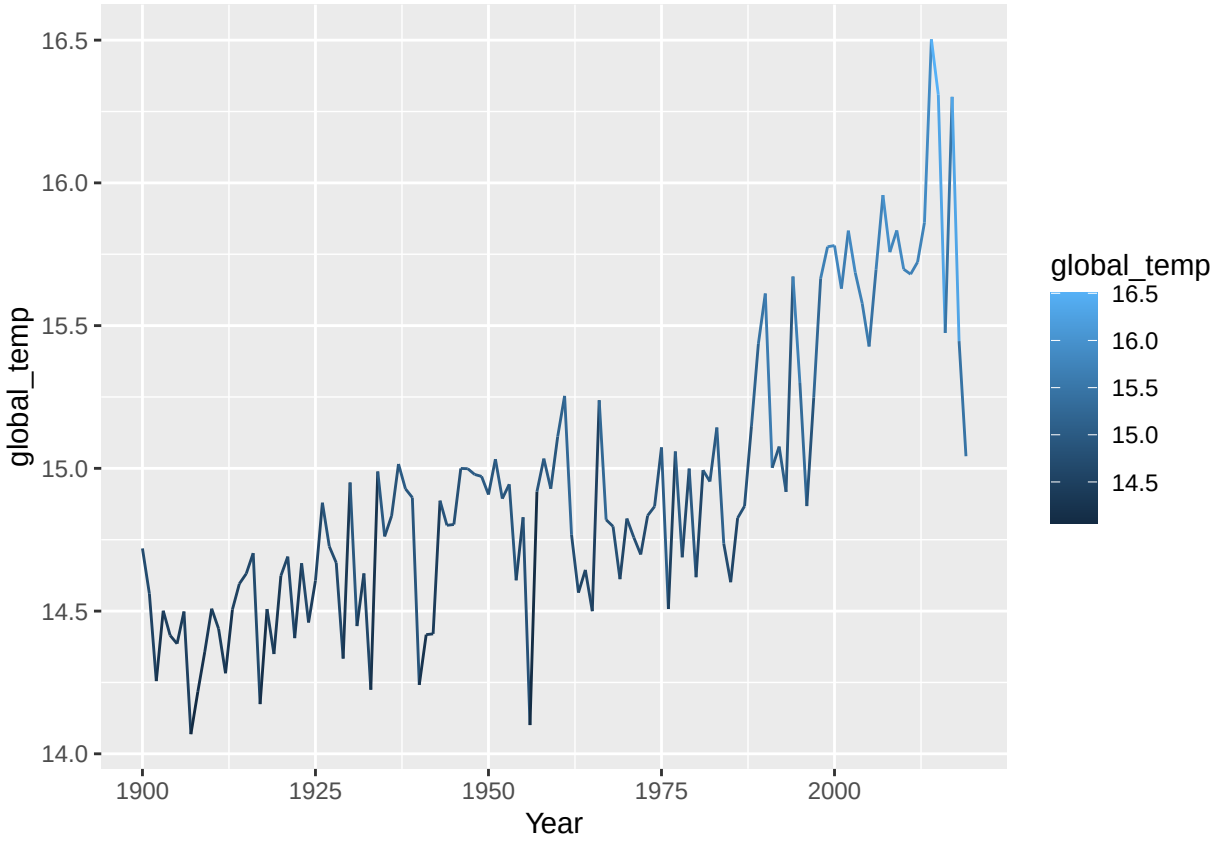

```
## Let us examine the annual averages and min/max temperatures variations

#Below we can see that all averages and max temperatures are increasing constantly across years.


global_land_temp <- data %>% group_by(Year) %>%
  summarize(global_temp = mean(Temp), min_global_temp = min(Temp), max_global_temp = max(Temp))
global_land_temp
```

```
## # A tibble: 120 x 4
##     Year global_temp min_global_temp max_global_temp
##    <dbl>       <dbl>           <dbl>           <dbl>
##  1  1900        14.7           -7.59            35.9
##  2  1901        14.6           -7.64            36.9
##  3  1902        14.3           -9.22            36.3
##  4  1903        14.5           -5.11            37.0
##  5  1904        14.4           -7.27            36.0
##  6  1905        14.4           -8.92            36.5
##  7  1906        14.5           -5.55            35.9
##  8  1907        14.1           -9.92            35.8
##  9  1908        14.2           -6.23            35.6
## 10  1909        14.4           -9.89            36.2
## # ... with 110 more rows
## # i Use 'print(n = ...)' to see more rows
```

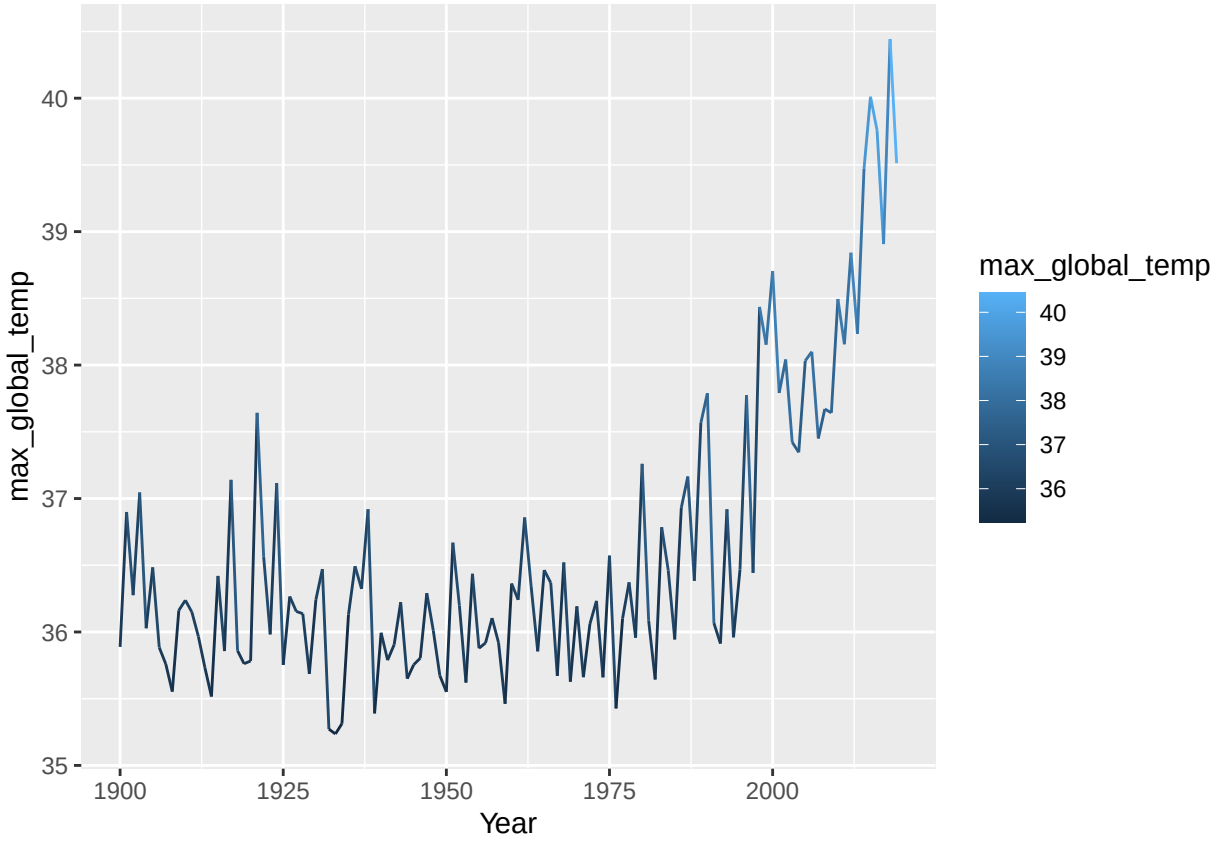
```
global_land_temp[which.min(global_land_temp$min_global_temp),]
```

```
## # A tibble: 1 x 4
##    Year global_temp min_global_temp max_global_temp
##   <dbl>       <dbl>           <dbl>           <dbl>
## 1  2016        15.5           -59.3            39.8
```

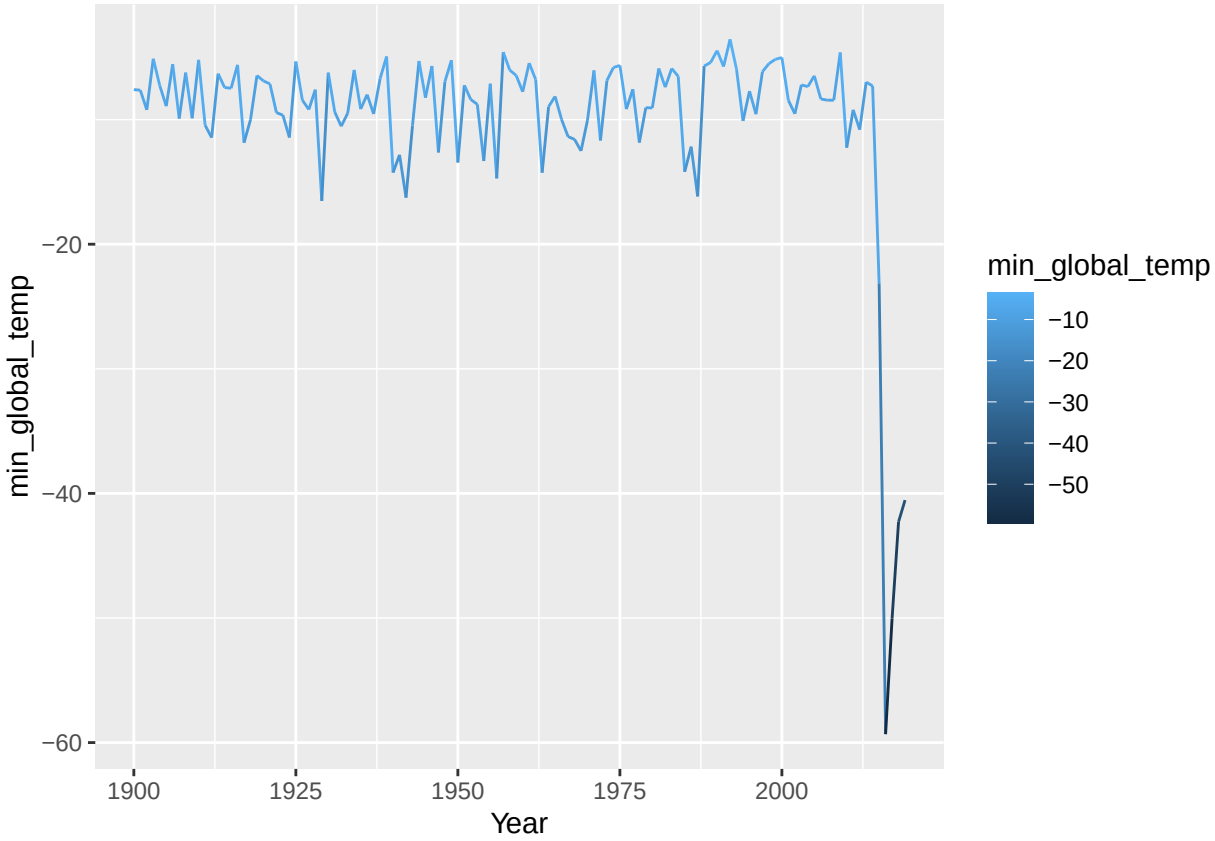
```
ggplot(data=global_land_temp,aes(x=Year,y=global_temp,color=global_temp))+geom_line()
```



```
ggplot(data=global_land_temp,aes(x=Year,y=max_global_temp,color=max_global_temp))+geom_line()
```




```
#### The extreme winter average in graph is from Polar vortex year 2014 in graph below

ggplot(data=global_land_temp,aes(x=Year,y=min_global_temp,color=min_global_temp))+geom_line()
```




```
## Let us examine the change in temperatures across years for each month.

monthly_averages <- data %>%
  group_by(Year, Month) %>%
  summarize(mean_monthly = mean(Temp), min_monthly = min(Temp), max_monthly = max(Temp))
```

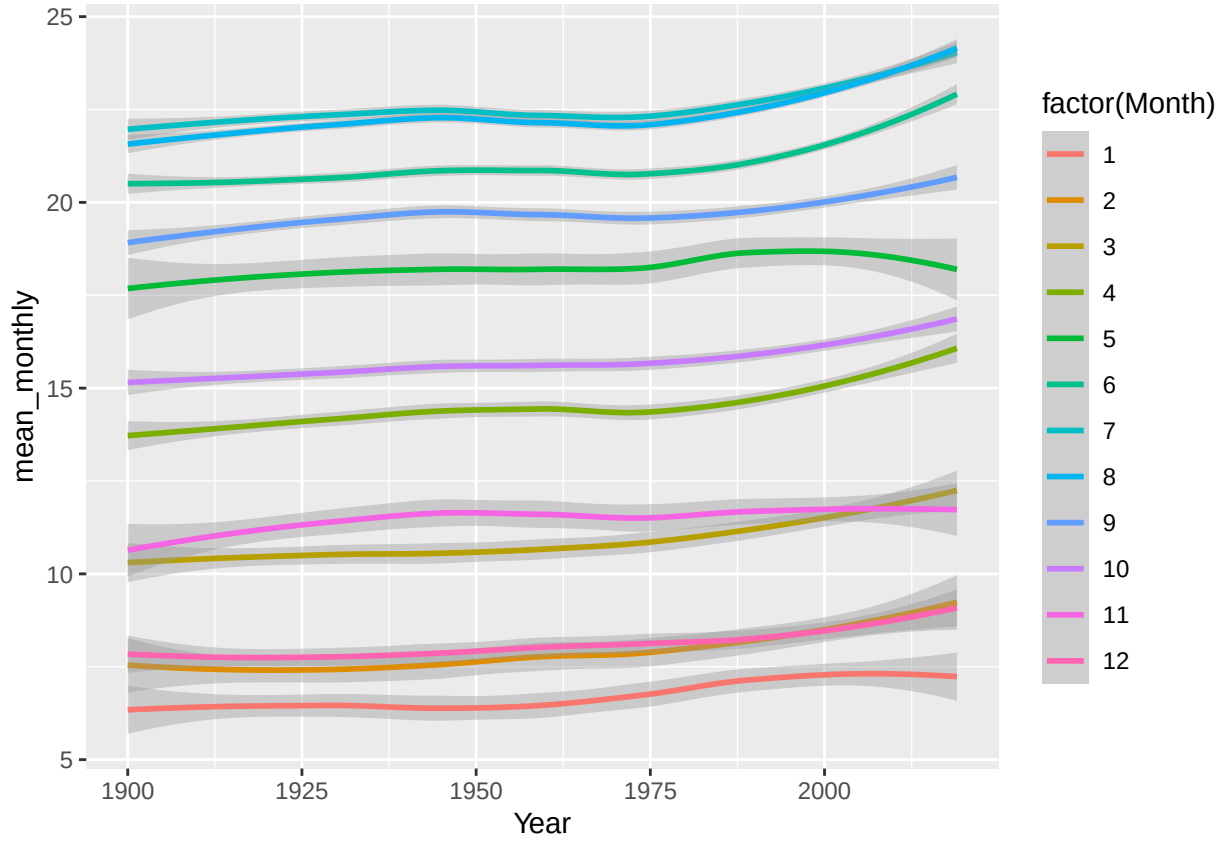
```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
head(monthly_averages)
```

```
## # A tibble: 6 x 5
## # Groups:   Year [1]
##    Year Month mean_monthly min_monthly max_monthly
##   <dbl> <dbl>        <dbl>       <dbl>       <dbl>
## 1  1900     1         6.57      -7.59         25.8
## 2  1900     2         8.27      -6.31         26.4
## 3  1900     3         9.51      -3.51         26.8
## 4  1900     4        14.1       -0.787        28.8
## 5  1900     5        17.5        2.65         31.1
## 6  1900     6        20.8        7.53         34.8
```

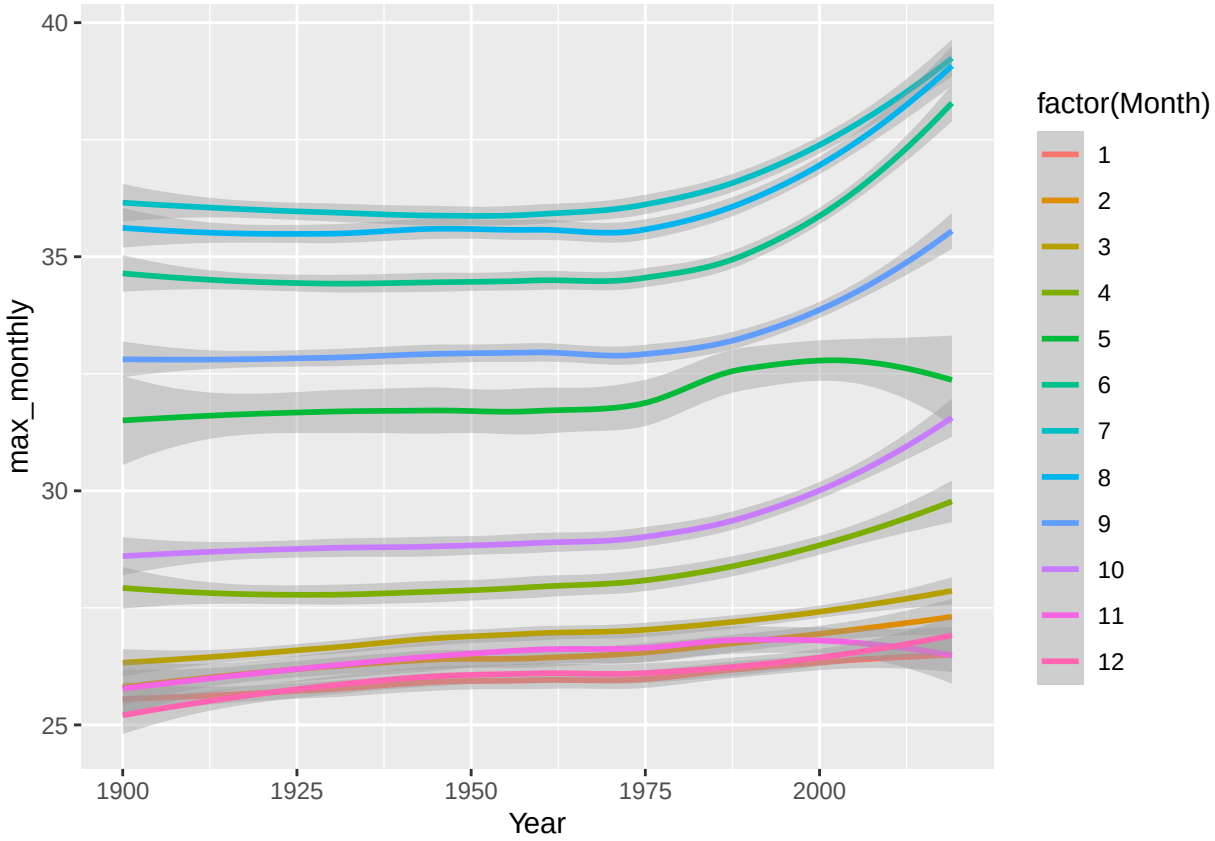
```
monthly_averages %>%
  ggplot(aes(x = Year, y = mean_monthly, color = factor(Month))) +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
monthly_averages %>%
  ggplot(aes(x = Year, y = max_monthly, color = factor(Month))) +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
### Above graph shows that except for the season changing months of May and November all months have av
```

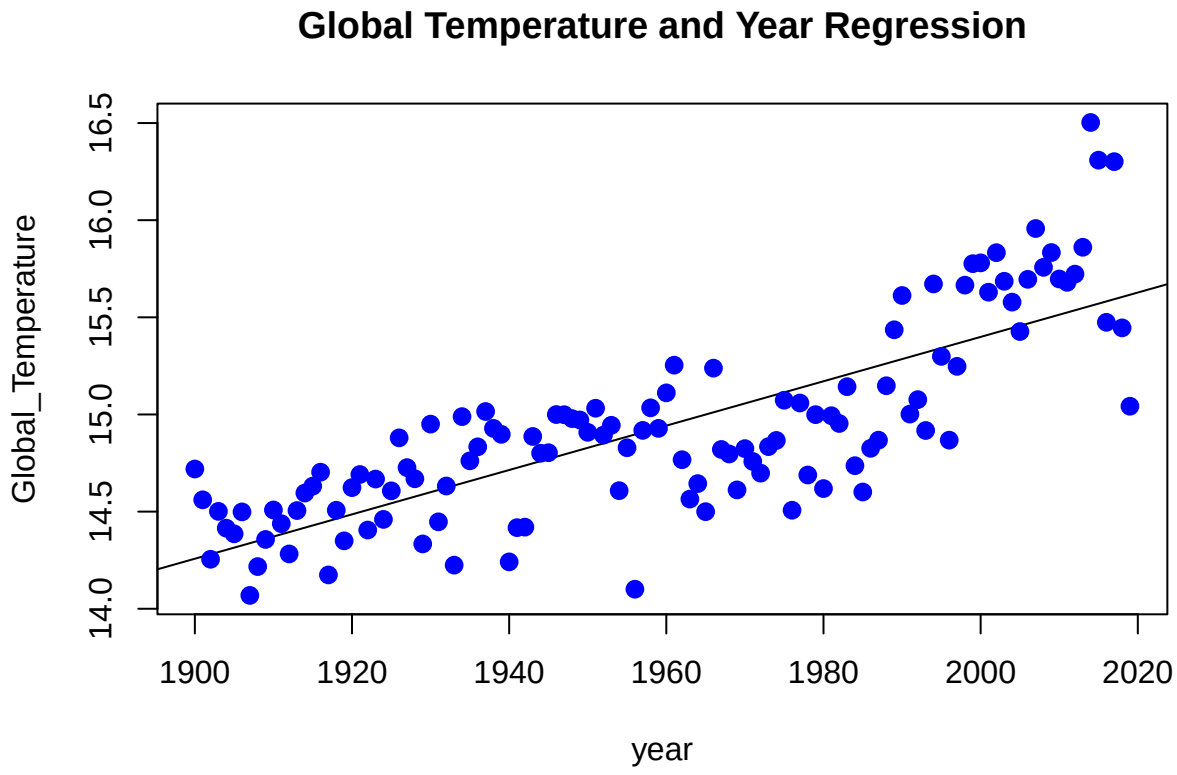
#Regression Part for Extra Credits

```
#creating predictor and response variable for Linear Regression
x <- global_land_temp$Year
y <- global_land_temp$global_temp

relation <- lm(y~x)
print(summary(relation))
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79617 -0.23337  0.04339  0.19683  0.94368
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.4361789  1.5747242  -4.722 6.48e-06 ***
## x            0.0114177  0.0008035  14.210  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.3049 on 118 degrees of freedom
## Multiple R-squared:  0.6312, Adjusted R-squared:  0.628
## F-statistic: 201.9 on 1 and 118 DF,  p-value: < 2.2e-16
```

```
# Plot the chart.
plot(x,y,col = "blue",main = "Global Temperature and Year Regression",
abline(relation),cex = 1.3,pch = 16,xlab = "year",ylab = "Global_Temperature")
```




```
#creating predictor and response variable for Linear Regression
x1 <- monthly_averages$Month
x2 <- monthly_averages$Month^2
y <- monthly_averages$mean_monthly

relation <- lm(y~x1+x2)
print(summary(relation))
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1015  -1.2494  -0.1109   1.3440   5.3999
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.630460   0.175482  -14.99   <2e-16 ***
## x1           6.776766   0.062064  109.19   <2e-16 ***
## x2          -0.488893   0.004648 -105.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.86 on 1437 degrees of freedom
## Multiple R-squared:  0.8926, Adjusted R-squared:  0.8925
## F-statistic:  5973 on 2 and 1437 DF,  p-value: < 2.2e-16
```

```
# Plot the chart.
monthValues <- seq(0, 12, 0.1)
tempPredict <- predict(relation,list(x1=monthValues,x2=monthValues^2))
plot(x1,y,col = "red",xlab = "Month",ylab = "Global_Temperature",main = "MeanTemperature and Month Regr
lines(monthValues,tempPredict, col = 'blue' )
```