MM957: Data Analytics in R

Introduction

Report Structure

This report examines two RData files containing detailed weather forecasting and analysis for various locations across the United States. The objective is to tidy and analyze the data to provide insights that could assist a European travel agency in identifying ideal US cities for their package holidays, focusing on warmer destinations. The report is organised into four main parts:

1. **Data Management**: This section focuses on cleaning and restructuring the forecasts and cities data to create a coherent and usable dataset. Tasks include separating state and city information, restructuring measurement levels, and filtering data by specific dates and validity.

2. **Exploring the Data**: Here, we employ a combination of data visualisation and statistical analysis to understand the relationships within the data. We will explore how geographical factors like longitude and latitude influence temperature and investigate the relationship between temperature and the Köppen climate classification.

3. **Building a Model**: Using the cleaned and explored data, we will build and refine predictive models to estimate average temperatures of US cities based on various geographical and climatic factors. The models will be evaluated and compared to determine the most effective predictors of temperature.

4. **Summary of Results**: In the final part, the report synthesises the findings to address the travel agency's specific inquiries about predicting temperatures, the confidence in our model, and the temperature variations relative to proximity to the coast. The analysis will conclude with recommendations on which cities could be considered potential travel destinations based on their average temperatures.

Each section of the report is designed to build upon the previous, culminating in a comprehensive analysis that supports informed decision-making for the travel agency's venture into the US market. The findings will be presented using clear graphs, sensible scaling, and concise textual explanations to ensure clarity and usefulness of the information provided.

Packages Used

Using R version 4.40, Visual Studio Code version 1.90 and IRkernal version 1.3.2. A number of packages within R have been used listed below:

In [63]:
```
# Load necessary libraries
library(dplyr)
library(tidyr)
library(dplyr)
library(stringr)
library(ggplot2)
library(maps)
library(car)
library(MASS)
library(broom)
library(caret)
```

The files used in this report are shown below:

In [64]:
```
# Load the data
load("r_data/cities.RData")
load("r_data/forecasts.RData")
load("r_data/weather.RData")
```

Part 1: Data Management

Forecast Data

- The combined "State_City" variable was split into two separate variables, `State` and `City`, to clearly distinguish these geographic details.
- Using the `pivot_wider()` method, we transformed each `Measurement` type into its own column, allowing individual measurement responses to be more accessible and analyzable.
- Data filtering was applied to retain only the entries where an `observed temp` was recorded. Accuracy checks were performed by reviewing maximum and minimum temperature values to ensure reliability.
- We then constrained the dataset to include only data from the period between **1st February 2021** and **31st January 2022**, focusing our analysis on this specific timeframe.
- The average temperature for each city was calculated and stored in a newly created variable `avg_temp`.

- To streamline the dataset, we retained only one row per city in each state, ensuring a unique representation for each city-state combination in our analysis.

These steps enabled the preparation of a tidy and focused dataset ready for further analysis and modeling.

In [65]:
```
# Create unique variables for State and City.
forecasts <- forecasts %>%
  separate(State_City, into = c("State", "City"), sep = ":")
```

In [66]:
```
# Reshape the data so that each forecast_outlook becomes a column
forecasts_pivoted <- forecasts %>%
  dplyr::select(-possible_error) %>%
  pivot_wider(
    names_from = forecast_outlook,
    values_from = forecast_outlook,
    values_fill = list(forecast_outlook = 0),
    values_fn = list(forecast_outlook = length)
  )
```

In [67]:
```
# Step 1: Filter to get only rows where Measurement is "observed_temp"
observed_temps <- forecasts_pivoted %>%
  filter(Measurement == "observed_temp")

# Step 2: Find the maximum and minimum temperature values to confirm accuracy
max_temp <- max(observed_temps$Response, na.rm = TRUE)
min_temp <- min(observed_temps$Response, na.rm = TRUE)

# Print the results
cat("Maximum observed temperature:", max_temp, "\n")
cat("Minimum observed temperature:", min_temp, "\n")
```

```
Maximum observed temperature: 107
Minimum observed temperature: -47
```

In [68]:
```
# Convert column to date format
observed_temps$date <- as.Date(observed_temps$date, format = "%Y-%m-%d")

# Filter the dates
filtered_forecasts <- observed_temps %>%
  filter(date >= as.Date("2021-02-01") & date <= as.Date("2022-01-31"))
```

In [69]:
```
# Calculate the average temperature for each city
avg_temp <- filtered_forecasts %>%
  group_by(City) %>%
  summarize(Average_Temperature = mean(Response, na.rm = TRUE))

head(avg_temp)
```

A tibble: 6 × 2

City	Average_Temperature
<chr>	<dbl>
ABILENE	55.82906
AKRON_CANTON	45.74148
ALBANY	40.60795
ALBUQUERQUE	48.07386
ALLENTOWN	43.90805
AMARILLO	46.22159

In [70]:
```
filtered_forecasts <- filtered_forecasts %>%
  left_join(avg_temp, by = "City") %>%
  rename(avg_temp = Average_Temperature)
```

In [71]:
```
unique_city_group <- filtered_forecasts %>%
  group_by(State, City) %>%
  slice(1)

# View the resulting data
head(unique_city_group)
```

A grouped_df: 6 × 30

date	State	City	Measurement	Response	FOG	VRYCLD	SNOW	BLGSNO	PTCLDY	⋯	FZDRZL	RAIN	NA	FZRAIN	VI
<date>	<chr>	<chr>	<chr>	<dbl>	<int>	<int>	<int>	<int>	<int>	⋯	<int>	<int>	<int>	<int>	
2021-12-12	AK	ANCHORAGE	observed_temp	3	0	1	0	0	0	⋯	0	0	0	0	
2022-01-03	AK	FAIRBANKS	observed_temp	-32	1	0	0	0	0	⋯	0	0	0	0	
2022-01-04	AK	JUNEAU	observed_temp	-4	0	1	0	0	0	⋯	0	0	0	0	
2021-02-15	AL	BIRMINGHAM	observed_temp	16	0	0	0	0	1	⋯	0	0	0	0	
2021-02-15	AL	HUNTSVILLE	observed_temp	13	0	0	0	0	0	⋯	0	0	0	0	
2021-02-15	AL	MOBILE	observed_temp	20	0	0	0	0	0	⋯	0	0	0	0	

Cities Data

- **Creation of New Variables:** A new variable `koppen` was created, which used names from columns 10 to 25 as the levels. The responses from these columns were consolidated into a variable named `avg_annual_precip`. During this step, any rows with missing values in the `avg_annual_precip` column were excluded to maintain data integrity.
- **Data Integration:** The cities data was then joined with the cleaned forecasts data using an inner join method. This merge ensured that only cities present in both datasets were retained for subsequent analysis.
- **Data Reduction:** Further cleaning was performed to refine the dataset to essential variables only. The final set of variables retained includes:
  - `State`
  - `City`
  - `lon` (longitude)
  - `lat` (latitude)
  - `koppen` (Köppen climate classification)
  - `elevation`
  - `distance_to_coast`
  - `wind`
  - `elevation_change_four`
  - `elevation_change_eight`
  - `avg_annual_precip`
  - `avg_temp` (average temperature)

These steps facilitated the creation of a streamlined dataset that combines geographical and climatological information, ready for detailed analysis and modeling.

In [72]:
```
# Create new Koppen variable
cities_long <- cities %>%
  gather(key = "koppen", value = "avg_annual_precip", 10:25) %>%
  na.omit()  # Rows with NA in avg_annual_precip removed
```

In [73]:
```
# Rename the 'city' and 'state' columns so they are consistent
cities_long <- rename(cities_long, City = city, State = state)
```

In [74]:
```
# Ensure city and state names are formatted consistently
cities_long$City <- tolower(trimws(cities_long$City))
cities_long$State <- tolower(trimws(cities_long$State))
unique_city_group$City <- tolower(trimws(unique_city_group$City))
unique_city_group$State <- tolower(trimws(unique_city_group$State))

# Now perform the inner join
combined_data <- inner_join(cities_long, unique_city_group, by = c("City", "State"))

# Display the first few rows of the combined data
head(combined_data)
```

A data.frame: 6 × 39

	City	State	lon	lat	elevation	distance_to_coast	wind	elevation_change_four	elevation_change_eight	koppen	⋯	F:
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	⋯	
1	abilene	tx	-99.68	32.41	545.46	328.89	4.43	66.72	66.72	Cfa	⋯	
2	atlanta	ga	-84.43	33.64	305.73	242.19	3.51	43.96	70.17	Cfa	⋯	
3	atlantic_city	nj	-74.57	39.45	18.07	6.44	2.75	18.06	18.06	Cfa	⋯	
4	austin	tx	-97.77	30.32	197.53	134.18	2.30	95.64	104.38	Cfa	⋯	
5	baltimore	md	-76.68	39.17	46.05	7.33	3.65	21.80	65.78	Cfa	⋯	
6	baton_rouge	la	-91.15	30.54	20.20	55.37	3.53	7.76	25.52	Cfa	⋯	

In [75]:
```
# Retain only the specified variables
selected_data <- combined_data %>%
  dplyr::select(State, City, lon, lat, koppen, elevation, distance_to_coast,
         wind, elevation_change_four, elevation_change_eight,
         avg_annual_precip, avg_temp)

# Adjust the State and City Values for Readability
selected_data <- selected_data %>%
  mutate(
    State = toupper(State),  # Capitalize state abbreviations
    City = str_replace_all(City, "_", " "),  # Replace underscores with spaces in city names
    City = str_to_title(City)  # Capitalize the first letter of every word in city names
  )

# Display the first few rows of the filtered data to verify
head(selected_data)
```

A data.frame: 6 × 12

	State	City	lon	lat	koppen	elevation	distance_to_coast	wind	elevation_change_four	elevation_change_eight	avg_ann
	<chr>	<chr>	<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
1	TX	Abilene	-99.68	32.41	Cfa	545.46	328.89	4.43	66.72	66.72	
2	GA	Atlanta	-84.43	33.64	Cfa	305.73	242.19	3.51	43.96	70.17	
3	NJ	Atlantic City	-74.57	39.45	Cfa	18.07	6.44	2.75	18.06	18.06	
4	TX	Austin	-97.77	30.32	Cfa	197.53	134.18	2.30	95.64	104.38	
5	MD	Baltimore	-76.68	39.17	Cfa	46.05	7.33	3.65	21.80	65.78	
6	LA	Baton Rouge	-91.15	30.54	Cfa	20.20	55.37	3.53	7.76	25.52	

Part 2: Exploring the Data

Average Temperature Analysis

The histogram of average temperature displays a distribution with a mean temperature of approximately 49.01°F, highlighting the central tendency of the data. The median temperature is slightly lower at 47.71°F, indicating a slight skew in the distribution towards higher values. This difference between the mean and median suggests the presence of outliers or a long tail on the right side of the distribution.

The standard deviation, a measure of the spread of temperature values around the mean, is 9.12°F. This value suggests that, on average, the temperatures vary by about 9.12°F from the mean, which indicates moderate variability within the dataset.

Additionally, the interquartile range (IQR) of 10.75°F, which measures the range between the 25th and 75th percentiles, further supports the presence of variability. The IQR specifically indicates that the middle 50% of the data is spread out over approximately 10.75°F.

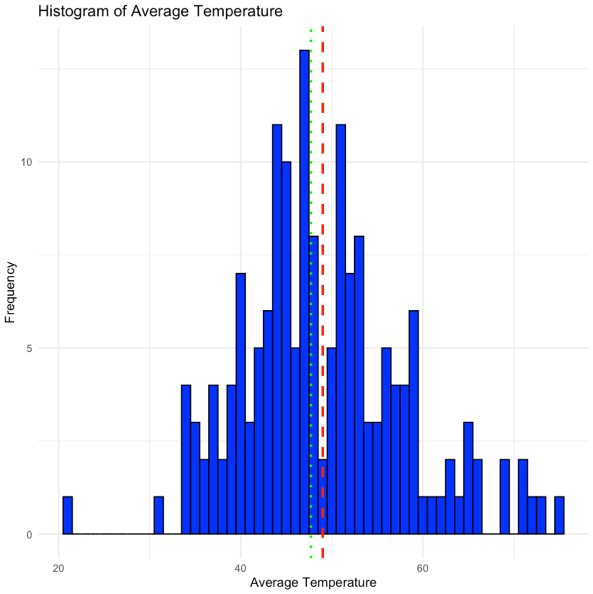
In the histogram, the peak of the distribution is evident around the 50°F mark. The red dashed line (mean) and green dotted line (median) in the histogram are close but not perfectly aligned, reinforcing the notion of a right-skewed distribution. This skewness suggests that while most of the cities have temperatures clustered around 40°F to 60°F, there are a significant number of cities with temperatures extending towards higher values.

In [76]:
```
# Calculate the mean and median
avg_temp_mean <- mean(selected_data$avg_temp, na.rm = TRUE)
avg_temp_median <- median(selected_data$avg_temp, na.rm = TRUE)

# Calculate the standard deviation and IQR
avg_temp_sd <- sd(selected_data$avg_temp, na.rm = TRUE)
avg_temp_iqr <- IQR(selected_data$avg_temp, na.rm = TRUE)
```

```
# Print the statistics
cat("Mean of Average Temperature:", avg_temp_mean, "\n")
cat("Median of Average Temperature:", avg_temp_median, "\n")
cat("Standard Deviation of Average Temperature:", avg_temp_sd, "\n")
cat("Interquartile Range of Average Temperature:", avg_temp_iqr, "\n")

# Create a histogram of average temperature
ggplot(selected_data, aes(x = avg_temp)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black") +
  geom_vline(aes(xintercept = avg_temp_mean), color = "red", linetype = "dashed", size = 1,
             label = "Mean") +
  geom_vline(aes(xintercept = avg_temp_median), color = "green", linetype = "dotted", size = 1,
             label = "Median") +
  labs(title = "Histogram of Average Temperature",
       x = "Average Temperature",
       y = "Frequency") +
  theme_minimal()
```

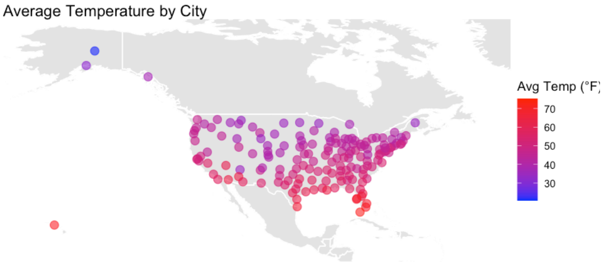
```
Mean of Average Temperature: 49.00726
Median of Average Temperature: 47.71225
Standard Deviation of Average Temperature: 9.120819
Interquartile Range of Average Temperature: 10.7518
```

```
Warning message in geom_vline(aes(xintercept = avg_temp_mean), color = "red", linetype = "dashed", :
“Ignoring unknown parameters: `label`”
Warning message in geom_vline(aes(xintercept = avg_temp_median), color = "green", :
“Ignoring unknown parameters: `label`”
```



The map below visualises average temperatures across U.S. cities, using a blue-to-red gradient to effectively highlight regional climatic differences, with denser, warmer colors in southern regions and cooler colors in the north.

In [77]:
```
# Get world map data
usa_map <- map_data("world")

# Create a heat map overlaying on the world map
ggplot() +
  geom_polygon(data = usa_map, aes(x = long, y = lat, group = group), fill = "gray90", color = "white") +
  geom_point(data = selected_data, aes(x = lon, y = lat, color = avg_temp), alpha = 0.6, size = 3) +
  scale_color_gradient(low = "blue", high = "red", name = "Avg Temp (°F)") +
  labs(title = "Average Temperature by City",
       x = "Longitude", y = "Latitude") +
  coord_fixed(xlim = c(-165, -50), ylim = c(15, 70)) +
  theme_void() +
  theme(legend.position = "right")
```



Temperature and Location Correlation

Two graphs below have been plotted to visualise the relationship between average temperature and geographical coordinates.

Average Temperature vs. Longitude:

- The first plot, "Average Temperature vs. Longitude," shows a positive trend, suggesting that as longitude increases (moving eastward across the U.S.), the average temperature slightly increases. The linear model line in blue highlights this positive relationship.
- The concentration of warmer temperatures (redder points) at higher longitudes might indicate the influence of specific geographic features or climates in the eastern regions of the U.S.

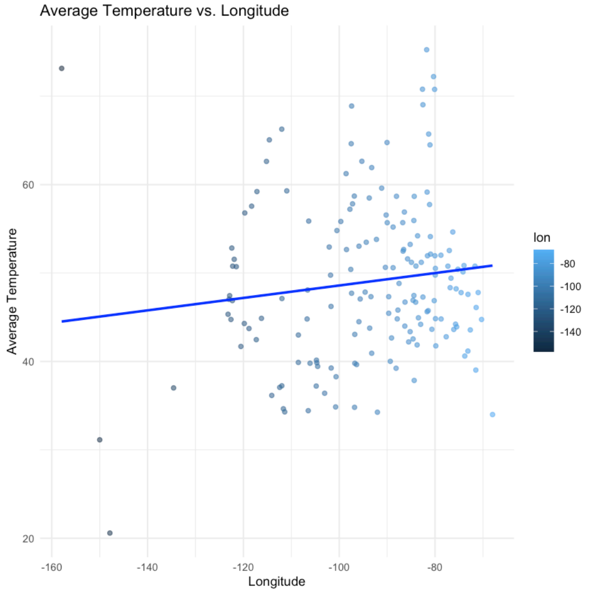
Average Temperature vs. Latitude:

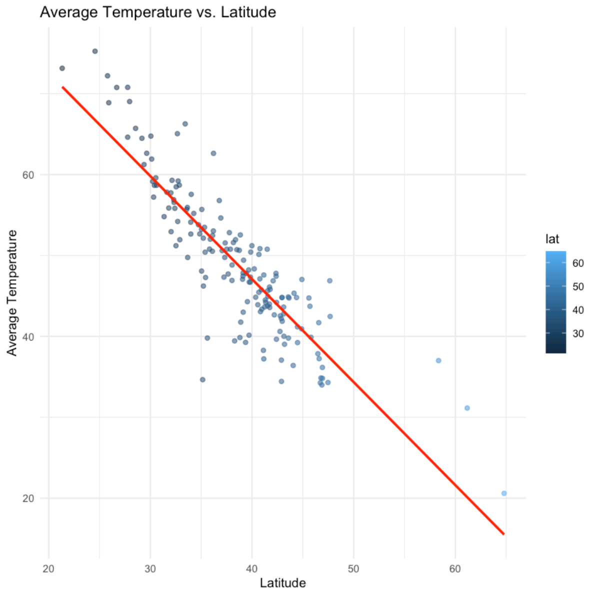
- The second plot, "Average Temperature vs. Latitude," demonstrates a clear negative correlation, indicating that as latitude increases (moving northward), the average temperature decreases. This is depicted by the red linear model line, illustrating a typical climatic gradient where temperatures drop in higher latitudes due to reduced solar intensity.
- This plot reveals a more dispersed set of data points, showing a more consistent pattern of temperature decline with increasing latitude.

In [78]:

```
# Scatter plot for Average Temperature vs. Longitude
plot_temp_lon <- ggplot(selected_data, aes(x = lon, y = avg_temp)) +
  geom_point(aes(color = lon), alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Average Temperature vs. Longitude",
       x = "Longitude",
       y = "Average Temperature") +
  theme_minimal()

# Scatter plot for Average Temperature vs. Latitude
plot_temp_lat <- ggplot(selected_data, aes(x = lat, y = avg_temp)) +
  geom_point(aes(color = lat), alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Average Temperature vs. Latitude",
       x = "Latitude",
       y = "Average Temperature") +
  theme_minimal()

# Print the plots
print(plot_temp_lon)
print(plot_temp_lat)
```

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





The correlation coefficient of 0.129 between average temperature and longitude indicates a very weak positive relationship, suggesting that as cities move eastward across the U.S., there is a slight increase in temperature. Conversely, the correlation coefficient of -0.877 between average temperature and latitude indicates a strong negative relationship, showing that as cities move northward, the average temperature significantly decreases.

In [79]:
```
# Calculate correlation coefficient for Average Temperature vs. Longitude
cor_coeff_lon <- cor(selected_data$lon, selected_data$avg_temp, use = "complete.obs")
print(paste("Correlation coefficient between average temperature and longitude:", cor_coeff_lon))

# Calculate correlation coefficient for Average Temperature vs. Latitude
cor_coeff_lat <- cor(selected_data$lat, selected_data$avg_temp, use = "complete.obs")
print(paste("Correlation coefficient between average temperature and latitude:", cor_coeff_lat))
```

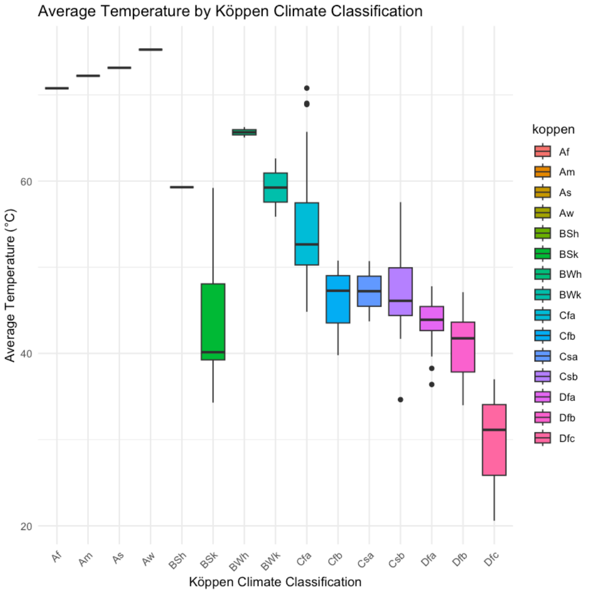
```
[1] "Correlation coefficient between average temperature and longitude: 0.129065350417646"
[1] "Correlation coefficient between average temperature and latitude: -0.877268008858417"
```

Average Temperature and Koppen

The box plot effectively illustrates the relationship between average temperatures and Köppen climate classifications. Tropical climates (green boxes) show higher and less variable temperatures, while arid (turquoise and light blue) and temperate climates (dark blues) display moderate temperatures with varying ranges. Cold climates (pinks) exhibit the broadest temperature ranges, reflecting significant seasonal variations.

In [80]:
```
# Convert 'koppen' to a factor if it's not already
selected_data$koppen <- as.factor(selected_data$koppen)

# Create a box plot of average temperature by Köppen climate classification
plot_koppen_temp <- ggplot(selected_data, aes(x = koppen, y = avg_temp)) +
  geom_boxplot(aes(fill = koppen)) +
  labs(title = "Average Temperature by Köppen Climate Classification",
       x = "Köppen Climate Classification",
       y = "Average Temperature (°C)") +
  theme_minimal() +
```

```
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

print(plot_koppen_temp)
```



Average Temperature and Koppen

The R code successfully counted cities with a Köppen climate classification starting with 'A', indicating a tropical climate where the coolest month is 18°C or higher, and identified that 4 cities fit this classification.

In [81]:
```
count_A_koppen <- 0

# Loop through each row and add to counter if A
for (i in 1:nrow(selected_data)) {
  if (substr(selected_data$koppen[i], 1, 1) == 'A') {
    count_A_koppen <- count_A_koppen + 1
  }
}

# Print the result
cat("Number of cities with a Köppen climate classification starting with 'A':", count_A_koppen)
```

```
Number of cities with a Köppen climate classification starting with 'A': 4
```

State Summary

The R code below introduces a function named `state_summary` that delivers insights about the number of cities and their average temperature for a specified state. An auxiliary function, `print_summary`, facilitates the output display.

The functions were applied to three states: Florida, New York, and South Dakota.

In [82]:
```
state_summary <- function(state) {

  state_data <- selected_data[selected_data$State == state, ]
  total_cities <- length(unique(state_data$City))
  average_temp <- mean(state_data$avg_temp, na.rm = TRUE)
  results <- list(
    Total_Cities = total_cities,
    Average_Temperature = average_temp
  )

  return(results)
}

print_summary <- function(state_name, summary) {
  cat("\n", state_name, " Summary:\n", sep="")
  cat("Total Cities: ", summary$Total_Cities, "\n")
  cat("Average Temperature: ", format(summary$Average_Temperature, nsmall = 2), "°C\n", sep="")
}

florida_summary <- state_summary("FL")
new_york_summary <- state_summary("NY")
south_dakota_summary <- state_summary("SD")

print_summary("Florida", florida_summary)
print_summary("New York", new_york_summary)
print_summary("South Dakota", south_dakota_summary)
```

```
Florida Summary:
Total Cities:  9
Average Temperature: 67.3425°C

New York Summary:
Total Cities:  5
Average Temperature: 44.53175°C

South Dakota Summary:
Total Cities:  2
Average Temperature: 38.09459°C
```

Part 3: Building a Model

Create Linear Regression Model

The linear regression model below effectively predicts average temperatures in U.S. cities using elevation, wind, precipitation, changes in elevation, distance to the coast, and latitude. Latitude is the strongest predictor, showing a significant negative relationship with temperature: each degree increase in latitude corresponds to a decrease of about 1.172°C, underscoring cooler temperatures at higher latitudes. Elevation and distance to the coast are also significant, indicating lower temperatures at higher elevations and further from the coast.

Although wind speed and changes in elevation did not show significant effects, average annual precipitation negatively impacts temperature, likely due to cooling effects from increased cloud cover. The model explains approximately 88.35% of the variance in temperatures (Multiple R-squared = 0.8835), demonstrating a strong fit. Overall, the model is statistically robust, with geographical positioning proving crucial in understanding temperature variations across cities.

In [83]:
```
# Fit the linear regression model
model <- lm(avg_temp ~ elevation + wind + avg_annual_precip + elevation_change_four +
            elevation_change_eight + distance_to_coast + lat, data = selected_data)

summary(model)
```

```
Call:
lm(formula = avg_temp ~ elevation + wind + avg_annual_precip +
    elevation_change_four + elevation_change_eight + distance_to_coast +
    lat, data = selected_data)

Residuals:
    Min      1Q  Median      3Q     Max
-8.3173 -1.7873 -0.3617  1.2955 11.8260

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.011e+02  2.069e+00  48.840  < 2e-16 ***
elevation              -5.572e-03  8.716e-04  -6.393 1.77e-09 ***
wind                   -5.150e-01  3.690e-01  -1.396  0.16483
avg_annual_precip      -4.957e-02  1.971e-02  -2.514  0.01293 *
elevation_change_four   4.098e-03  2.958e-03   1.386  0.16782
elevation_change_eight -2.772e-03  2.554e-03  -1.085  0.27953
distance_to_coast      -3.890e-03  1.242e-03  -3.133  0.00206 **
lat                    -1.172e+00  4.465e-02 -26.259  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.182 on 157 degrees of freedom
Multiple R-squared:  0.8835,	Adjusted R-squared:  0.8783
F-statistic: 170.1 on 7 and 157 DF,  p-value: < 2.2e-16
```

The ANOVA table for the linear regression model shows that latitude, elevation, and distance to the coast significantly affect average temperature across U.S. cities, with latitude being the most influential predictor (F value = 689.51, p < 2.2e-16). Elevation and average annual precipitation also contribute notably to the model, whereas wind and changes in elevation have no significant impact. This analysis underscores the importance of geographical and environmental factors in determining temperature variations.

In [84]:
```
# Generate an ANOVA table for the model
anova_model <- Anova(model)
print(anova_model)
```

```
Anova Table (Type II tests)

Response: avg_temp
                       Sum Sq  Df  F value    Pr(>F)
elevation               413.8   1  40.8700 1.771e-09 ***
wind                     19.7   1   1.9474  0.164834
avg_annual_precip        64.0   1   6.3216  0.012935 *
elevation_change_four    19.4   1   1.9200  0.167819
elevation_change_eight   11.9   1   1.1775  0.279531
distance_to_coast        99.4   1   9.8173  0.002063 **
lat                    6980.9   1 689.5149 < 2.2e-16 ***
Residuals              1589.5 157
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The test for the slope coefficient of distance to coast being equal to -0.01 yielded a test statistic of 4.920 and a very small p-value of approximately 2.16e-06. This indicates strong evidence against the null hypothesis, suggesting that the true slope is significantly different from -0.01. The impact of coastal proximity on temperature may be more substantial than expected.

In [85]:
```
coefficients <- summary(model)$coefficients

test_statistic <- (coefficients["distance_to_coast", "Estimate"] + 0.01) / coefficients["distance_to_coast", "Std
p_value <- 2 * pt(-abs(test_statistic), df = model$df.residual)

cat("Test statistic for slope of distance to coast = -0.01:", test_statistic, "\n")
cat("P-value for the test:", p_value, "\n")
```

```
Test statistic for slope of distance to coast = -0.01: 4.920415
P-value for the test: 2.162276e-06
```

Automatic Variable Selection Model

The revised model was developed using a forward selection approach, simplifying the full model to include only the most significant predictors: `latitude`, `elevation`, `distance to coast`, `average annual precipitation`, and `wind`. This method efficiently identified and retained key variables while discarding less impactful ones related to elevation changes.

In [96]:
```
# Define the full model with the correct variable name for average annual precipitation
full_model <- lm(avg_temp ~ elevation + wind + avg_annual_precip + elevation_change_four +
                 elevation_change_eight + distance_to_coast + lat, data = selected_data)

# Define the null model (only the intercept)
null_model <- lm(avg_temp ~ 1, data = selected_data)

# Perform forward selection starting from the null model
forward_model <- step(null_model,
                      scope = list(lower = null_model, upper = full_model),
                      direction = "forward",
                      trace = 0)
```

In [97]:
```
# Summarize the original full model
summary(full_model)

# Summarize the stepwise selected model
summary(forward_model)
```

```
Call:
lm(formula = avg_temp ~ elevation + wind + avg_annual_precip +
    elevation_change_four + elevation_change_eight + distance_to_coast +
    lat, data = selected_data)

Residuals:
    Min      1Q  Median      3Q     Max
-8.3173 -1.7873 -0.3617  1.2955 11.8260

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.011e+02  2.069e+00  48.840  < 2e-16 ***
elevation              -5.572e-03  8.716e-04  -6.393 1.77e-09 ***
wind                   -5.150e-01  3.690e-01  -1.396  0.16483
avg_annual_precip      -4.957e-02  1.971e-02  -2.514  0.01293 *
elevation_change_four   4.098e-03  2.958e-03   1.386  0.16782
elevation_change_eight -2.772e-03  2.554e-03  -1.085  0.27953
distance_to_coast      -3.890e-03  1.242e-03  -3.133  0.00206 **
lat                    -1.172e+00  4.465e-02 -26.259  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.182 on 157 degrees of freedom
Multiple R-squared:  0.8835,    Adjusted R-squared:  0.8783
F-statistic: 170.1 on 7 and 157 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = avg_temp ~ lat + elevation + distance_to_coast +
    avg_annual_precip + wind, data = selected_data)

Residuals:
    Min      1Q  Median      3Q     Max
-9.4341 -1.7443 -0.3802  1.3595 11.9643

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        1.008e+02  2.040e+00  49.405  < 2e-16 ***
lat               -1.168e+00  4.268e-02 -27.363  < 2e-16 ***
elevation         -5.613e-03  7.762e-04  -7.232 1.89e-11 ***
distance_to_coast -3.881e-03  1.198e-03  -3.240  0.00145 **
avg_annual_precip -4.888e-02  1.949e-02  -2.508  0.01315 *
wind              -4.924e-01  3.480e-01  -1.415  0.15901
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.182 on 159 degrees of freedom
Multiple R-squared:  0.882,     Adjusted R-squared:  0.8783
F-statistic: 237.8 on 5 and 159 DF,  p-value: < 2.2e-16
```

Compare the Models

The use of automatic variable selection methods, specifically forward selection, refined the model by focusing on the most significant predictors for average temperature in U.S. cities. The forward selection process started with a null model and iteratively added predictors that significantly improved the model's performance.

Original vs. Selected Model Comparison:

- The original full model included all predictors: elevation, wind, average annual precipitation, elevation changes, distance to coast, and latitude. This model had a high adjusted R-squared value of 0.8783, indicating a strong explanatory power.
- The stepwise selected model simplified to include only latitude, elevation, distance to coast, average annual precipitation, and wind. This resulted in the same adjusted R-squared value of 0.8783, suggesting that the simplified model maintains the explanatory power while using fewer variables.

Key Observations:

- **Latitude** remains the strongest predictor in both models, with the largest negative coefficient, emphasizing its critical role in determining average temperatures across geographical gradients.
- **Elevation** and **distance to coast** were consistently significant in both models, reinforcing their importance in temperature variation due to altitude effects and proximity to coastal climates.
- **Wind** and **average annual precipitation** also remained in the model, though wind was not statistically significant in either model, suggesting its lesser impact on the temperature relative to other factors.

The streamlined model from forward selection confirms the importance of the most impactful predictors while demonstrating that the omitted variables (elevation changes four and eight) contribute minimally to the model's performance. This efficient model approach allows for more focused interpretations and potentially more robust predictions, especially in applications like climatic impact assessments or urban planning where understanding temperature influences is crucial.

Check Regression Assumptions

Intrinsic Linearity

First the Intrinsic Linearity of the model was checked, this was done using Component + Residual plots using `crPlots` within R. These plots help assess the linearity of the relationship between each predictor and the dependent variable.

Main Assumptions in a Linear Regression Model

- **The errors are normally distributed** - This assumption can be checked with a Normal Q-Q plot, where the residuals of the model are plotted against a perfectly normal distribution. If the residuals lie along the line on this plot, the assumption is satisfied.

- **The errors have constant variance** - This was assessed through a Residuals vs. Fitted plot. Ideally, the spread of residuals should be constant across the range of predicted values.

- **The errors are independent** - Independence of errors will be checked by the plot of residuals against fitted values using the `acf` function.

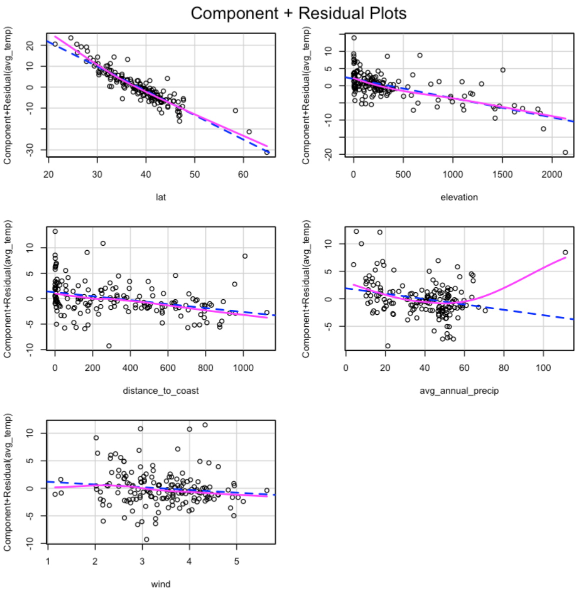
Residual Plot Results

- **Latitude and Average Annual Precipitation**: Both show non-linear relationships with the dependent variable, suggesting the need for transformations or the inclusion of polynomial terms to better capture their effects.
- **Elevation**: Displays a potentially linear relationship but with increasing variance at higher values, indicating that variance-stabilizing transformations could be beneficial.
- **Distance to Coast**: Appears linear with constant variance, suggesting that this predictor is appropriately modeled as is.

- **Wind**: Shows a flat trend across values, indicating a minimal or linear effect on the dependent variable, likely not requiring transformation.

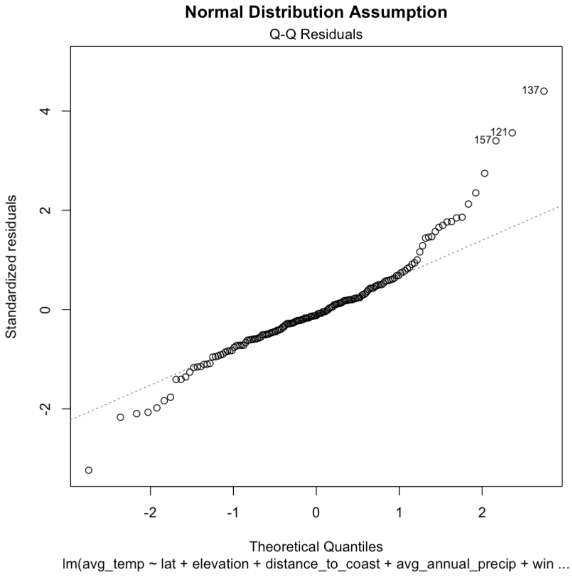
Overall, these plots suggest that while some predictors like distance to coast are well-modeled linearly, others, such as latitude and precipitation, exhibit non-linear patterns and may benefit from adjustments. Utilising Tukey's Ladder of Powers could provide effective transformations (e.g., logarithmic, square root) to better capture these complex relationships and enhance model accuracy.

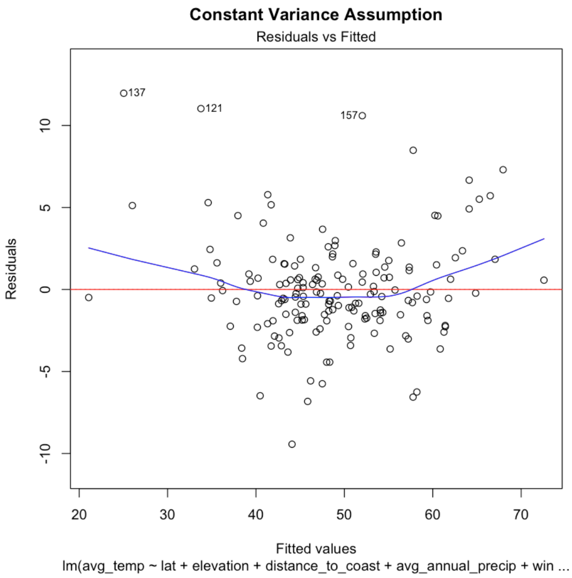
In [98]:
```
par(bg = "white")

crPlots(forward_model, main="Component + Residual Plots")
```

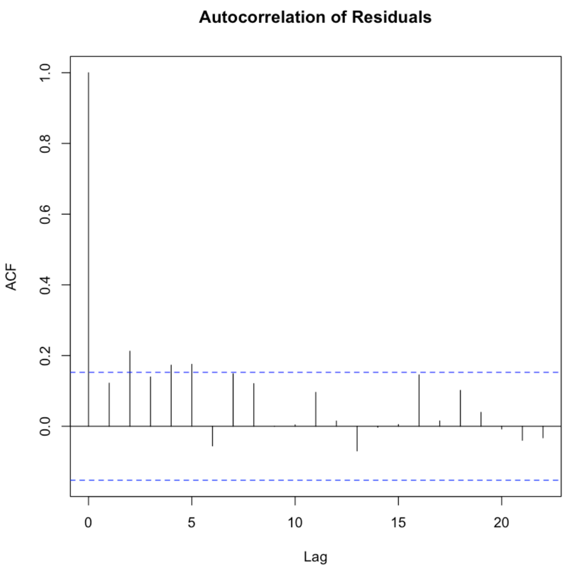


- **Normal Distribution Assumption**: The Q-Q plot below shows that while most residuals closely follow the expected diagonal line, suggesting general normality, there are notable deviations, particularly in the tails. These deviations indicate the presence of outliers or extreme values, which could be influencing the accuracy and reliability of your model.
- **Constant Variance Assumption**: The "Residuals vs. Fitted" plot indicates a generally constant variance in residuals and minimal patterns, but the presence of outliers and a slight curve suggest the model might not capture all underlying predictive relationships, indicating potential non-linearity or the need for model adjustments.
- **Errors are Independent Assumption**: The "Autocorrelation of Residuals" plot indicates that the residuals from the regression model are independent, as all autocorrelation coefficients are close to zero and within the confidence bands, confirming that there is no significant autocorrelation at any lag. This suggests that the model meets the assumption of independent errors, supporting the validity of the model's inferences.

In [100…
```
# Set plot parameters to ensure a white background
par(bg = "white")

# Normal Q-Q
plot(forward_model, which = 2, main = "Normal Distribution Assumption")

# Residuals vs. Fitted
plot(forward_model, which = 1, main = "Constant Variance Assumption")
abline(h = 0, col = "red")
lines(lowess(forward_model$fitted.values, forward_model$residuals), col = "blue")

residuals_acf <- acf(residuals(forward_model), main="Autocorrelation of Residuals")
```







Variable Transformation

In the provided code, transformations are applied to three different variables in the dataset to potentially enhance the linear regression model's accuracy and address issues such as non-linearity:

1. **Latitude:** The square of latitude values is computed to capture potential non-linear effects of latitude on the dependent variable, allowing the model to account for more complex geographical impacts on temperature.

2. **Elevation:** The square root transformation is applied to elevation values after shifting the scale upward by the absolute value of the minimum elevation plus one. This adjustment ensures all elevation values are positive, making them suitable for the square root transformation, which aims to reduce skewness and stabilize variance.

3. **Average Annual Precipitation:** A logarithmic transformation is applied to the average annual precipitation to address right-skewed distributions, helping to normalize the data and improve the model fit.

These transformations are common techniques to meet the assumptions necessary for accurate linear regression modeling, such as linearity, homoscedasticity, and normal distribution of errors.

In [101…
```
# Transformations

# Adjusting the Lattitude
selected_data$lat_squared <- selected_data$lat^2

# Adjusting the Elevation
# Some Elevations below 0, to apply sqrt an adjustment that shifts
# the entire scale was applied so that all values are now positive.
selected_data$elevation_sqrt <- sqrt(selected_data$elevation - min(selected_data$elevation) + 1)

# Adjusting the Annual Precipitation log
selected_data$avg_annual_precip_log <- log(selected_data$avg_annual_precip)
```

The new model that incorporates transformed variables demonstrates improved statistical performance over the previous model. Notably, the transformation of elevation to `elevation_sqrt` and the logarithmic transformation of average annual precipitation (`avg_annual_precip_log`) have significantly enhanced their respective impacts on the model, evidenced by stronger t-values and lower p-values for these predictors. Specifically, `elevation_sqrt` shows a more pronounced negative coefficient with a very significant p-value, indicating a stronger and more statistically significant relationship with average temperature than the linear `elevation` term in the previous model.

Additionally, the overall model's fit has improved, with an increase in both the Multiple R-squared (from 0.882 to 0.8947) and Adjusted R-squared (from 0.8783 to 0.8914), indicating that the model now explains a higher proportion of the variance in average temperature. The residual standard error has also decreased from 3.182 to 3.006, suggesting better prediction accuracy.

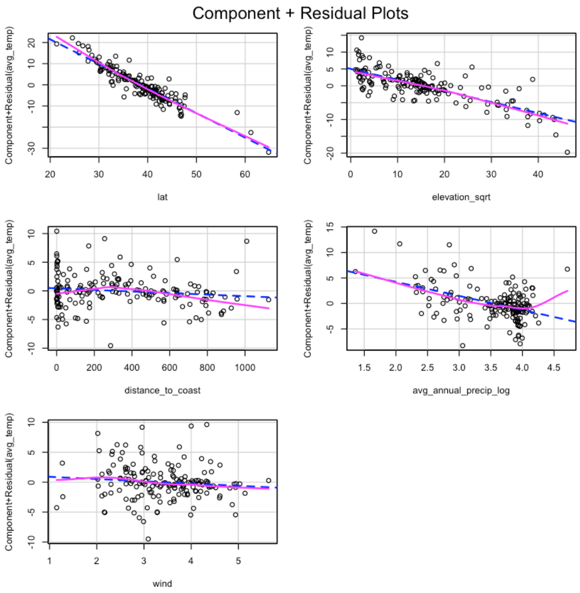
However, some variables like `wind` and `distance_to_coast` remain statistically insignificant, as seen from their p-values, which might suggest reevaluating their inclusion or exploring other forms of transformations or interaction terms that could reveal their potential effects more clearly.

In [102…
```
# Update the model to include the transformations where needed
updated_model <- lm(avg_temp ~ lat + elevation_sqrt + distance_to_coast +
                    avg_annual_precip_log + wind, data = selected_data)


summary(updated_model)
```

```
Call:
lm(formula = avg_temp ~ lat + elevation_sqrt + distance_to_coast +
    avg_annual_precip_log + wind, data = selected_data)

Residuals:
    Min      1Q  Median      3Q     Max
-9.6079 -1.4638 -0.1465  1.2737  9.9700

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           110.208881   2.746184  40.132  < 2e-16 ***
lat                    -1.162051   0.040108 -28.973  < 2e-16 ***
elevation_sqrt         -0.327602   0.036630  -8.943 9.19e-16 ***
distance_to_coast      -0.001370   0.001256  -1.091    0.277
avg_annual_precip_log  -2.759030   0.550761  -5.009 1.44e-06 ***
wind                   -0.374929   0.332090  -1.129    0.261
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.006 on 159 degrees of freedom
Multiple R-squared:  0.8947,    Adjusted R-squared:  0.8914
F-statistic: 270.2 on 5 and 159 DF,  p-value: < 2.2e-16
```

The updated Component + Residual plots reveal improved modeling of latitude and elevation through transformations, with more evenly distributed residuals, although some predictors like distance to coast and wind still show potential non-linearities.

In [103…
```
par(bg = "white")

crPlots(updated_model, main="Component + Residual Plots")
```



Part 4: Summary of Results

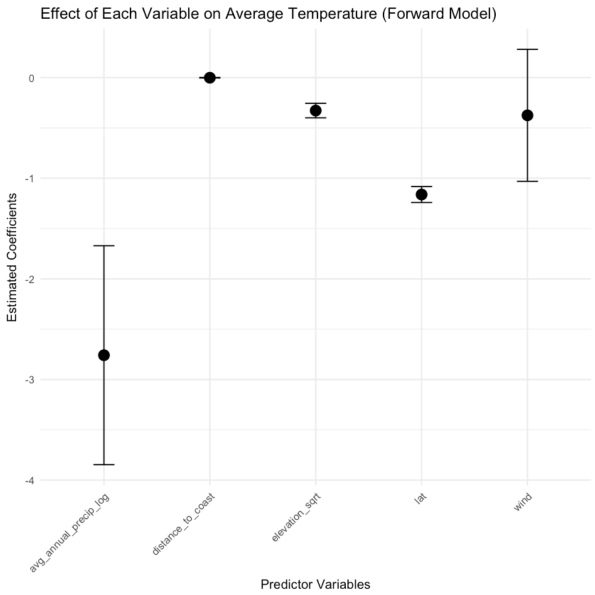
Variables to Predict Average Temperature

In [104…
```
# Extract model coefficients and confidence intervals from the forward model
coefficients_df <- tidy(updated_model, conf.int = TRUE)

coefficients_df
```

A tibble: 6 × 7

term	estimate	std.error	statistic	p.value	conf.low	conf.high
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	110.208881231	2.746183553	40.131651	4.233733e-85	104.785179115	115.632583347
lat	-1.162050556	0.040108042	-28.973007	2.528478e-65	-1.241263787	-1.082837326
elevation_sqrt	-0.327602448	0.036630364	-8.943467	9.194943e-16	-0.399947278	-0.255257618
distance_to_coast	-0.001369987	0.001255955	-1.090793	2.770140e-01	-0.003850493	0.001110519
avg_annual_precip_log	-2.759029834	0.550761255	-5.009484	1.438297e-06	-3.846781227	-1.671278442
wind	-0.374929447	0.332089904	-1.129000	2.605984e-01	-1.030805750	0.280946856

In [109…
```
# Filter out the intercept term
coefficients_df <- coefficients_df %>% dplyr::filter(term != "(Intercept)")

# Create Graph
ggplot(coefficients_df, aes(x = term, y = estimate)) +
  geom_point(size = 4) +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high), width = 0.2) +
  labs(title = "Effect of Each Variable on Average Temperature (Forward Model)",
       x = "Predictor Variables",
       y = "Estimated Coefficients") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Confidence in Model

In [106…

```
# Set up cross-validation
train_control <- trainControl(method = "cv", number = 10)
cv_forward_model <- train(avg_temp ~ lat + elevation + distance_to_coast + avg_annual_precip + wind,
                          data = selected_data,
                          method = "lm",
                          trControl = train_control)

# Print cross-validation results
summary(cv_forward_model)

# Extract cross-validation metrics
cv_results <- cv_forward_model$results
print(cv_results)
```

```
Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.4341 -1.7443 -0.3802  1.3595 11.9643 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        1.008e+02  2.040e+00  49.405  < 2e-16 ***
lat               -1.168e+00  4.268e-02 -27.363  < 2e-16 ***
elevation         -5.613e-03  7.762e-04  -7.232 1.89e-11 ***
distance_to_coast -3.881e-03  1.198e-03  -3.240  0.00145 ** 
avg_annual_precip -4.888e-02  1.949e-02  -2.508  0.01315 *  
wind              -4.924e-01  3.480e-01  -1.415  0.15901    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.182 on 159 degrees of freedom
Multiple R-squared:  0.882,	Adjusted R-squared:  0.8783 
F-statistic: 237.8 on 5 and 159 DF,  p-value: < 2.2e-16
  intercept     RMSE  Rsquared      MAE   RMSESD RsquaredSD     MAESD
1      TRUE 3.142157 0.8868542 2.318779 1.160626 0.07805668 0.7643552
```

Expected Average Temperature of Florida

In [107…

```
# Create a new data frame with the given values
new_data <- data.frame(
  lon = -82.33,
  lat = 29.65,
  elevation = 13,
  distance_to_coast = 3.25,
  avg_annual_precip = 51.04,
  wind = mean(selected_data$wind, na.rm = TRUE)  # Using mean wind speed as a placeholder
)

# Predict the average temperature for the new data
predicted_temp <- predict(forward_model, newdata = new_data)

# Filter data for Florida
florida_data <- selected_data[selected_data$State == "FL", ]

# Calculate the average temperature for Florida
average_temp_florida <- mean(florida_data$avg_temp, na.rm = TRUE)
```

```
# Print the comparison
cat("Predicted Average Temperature for the new city: ", round(predicted_temp, 2), "°F\n")
cat("Average Temperature of Florida: ", round(average_temp_florida, 2), "°F\n")
cat("Difference: ", round(predicted_temp - average_temp_florida, 2), "°F\n")
```

```
Predicted Average Temperature for the new city:  61.89 °F
Average Temperature of Florida:  67.34 °F
Difference:  -5.46 °F
```

Evaluation of Springfield as a travel destination

In [108…

```
# Create a new data frame with Springfield's values
springfield_data <- data.frame(
  lon = -83.81,
  lat = 39.93,
  elevation = 298,
  distance_to_coast = 453,
  avg_annual_precip = 38.512,
  wind = 4.5
)

# Predict the average temperature and its confidence interval for Springfield
springfield_prediction <- predict(forward_model, newdata = springfield_data, interval = "confidence")

# Extract the lower and upper bounds of the confidence interval
lower_bound <- springfield_prediction[1, "lwr"]
upper_bound <- springfield_prediction[1, "upr"]

# Print the results
cat("Predicted Average Temperature for Springfield: ", round(springfield_prediction[1, "fit"], 2), "°C\n")
cat("Confidence Interval: [", round(lower_bound, 2), "°C, ", round(upper_bound, 2), "°C]\n")

# Check if 55 degrees is within the confidence interval
if (lower_bound <= 55 && upper_bound >= 55) {
  cat("55 degrees is within the range of plausible values. Springfield should be considered as a travel destinati
} else {
  cat("55 degrees is not within the range of plausible values. Springfield should not be considered as a travel d
}
```

```
Predicted Average Temperature for Springfield:  46.61 °C
Confidence Interval: [ 45.75 °C,  47.46 °C]
55 degrees is not within the range of plausible values. Springfield should not be considered as a travel destinat
ion.
```

MM957 - Data Analytics in R

Part 4: Summary of Results

Univesrity of Strathclyde

Introduction

As the demand for tailored travel experiences continutes to rise, it is crucial to understand the climatic patterns fo potential cities. Tourists today make decision based on what they believe the climatic conditions of a destination are (Becken, 2010). This necessities the need for a predictive model that can estimate average tempeartures based on various geographic and meteorological variables.

This report aims to guide the agency in selecteing the most impactful varaibles for temperature prediction, ensuring confidence in the model's accuracy. Previous research in this area has included a range of models, (Chinchwad, 2019) investigated the use of a Time Series ARIMA model which was contary to prevopuus machine learning classification models.

This report focuses ona. Linear Regression model using R, enchanced with variable transformation and optimisation. This could offer improvements in predicitive accuracy and model robustness over previous models, as demonstracted by (James, Witten, Hastie, & Tibshirani, 2021).

Methodology

This section outlines the methods used to assist a European travel agency in selecting US cities for package holidays based on average temperature predictions.

1. **Variable Selection for Temperature Prediction:** The impact of each selected variable is visualised using coefficient plots created with ggplot2.
2. **Model Confidence Validation:** Employed 10-fold cross-validation via the caret package to assess model reliability.
3. **Coastal Proximity Analysis:** Analysed the influence of distance from the coast on temperature by calculating their coefficient.
4. **Temperature Prediction for Specific Locations:** Predicted the temperature for a specified city in Florida and compared it to the state's average.
5. **Evaluation of Potential New Destinations:** Assessed Springfield, Ohio as a potential destination by predicting its average temperature and it meets the agency's minimum temperature criterion of 55 degrees Fahrenheit using confidence intervals.

Analysis

Based on the boxplot analysis (Figure 1), the travel agency should prioritize Average Annual Precipitation, Latitude, and Elevation as key variables for predicting average temperature. Wind speed, with its wide confidence interval crossing zero, shows uncertain impact on temperature, while Distance to Coast has a minimal negative coefficient, suggesting a less significant effect.

The model demonstrates strong predictive accuracy and stability, evidenced by an R-squared of 0.897, indicating it explains nearly 90% of the variance in temperature, and low error metrics (RMSE of 3.056 and MAE of 2.189).

The graph (Figure 2) and statistical outputs indicate a significant negative relationship between distance to the coast and average temperature, showing that cities farther from the coast tend to have lower temperatures, as evidenced by the negative coefficient and the low p-value (0.0015).

The predicted average temperature for a city in Florida is 61.89°F, which is 5.46°F cooler than the state average of 67.34°F, indicating that this city may be slightly cooler than the typical Florida location but still within a reasonable margin of error. In contrast, Springfield, Ohio, with a predicted average temperature of 46.61°C and a confidence interval not encompassing 55°C, does not meet the travel agency's minimum temperature criterion, thus it should not be considered as a potential travel destination.

Conclusion

This report provides key insights for selecting travel destinations based on climate. It recommends prioritizing Average Annual Precipitation, Latitude, and Elevation, as these variables significantly impact and enhance the model's accuracy, evidenced by a high R-squared of 0.897 and low error metrics (RMSE of 3.056 and MAE of 2.189). Analysis also shows that cities farther from the coast are cooler, supported by data indicating a strong negative correlation between distance to the coast and temperature. The model's high accuracy is highlighted by the example of a city in Florida, which was only 5.46°F cooler than the state's average temperature. However, Springfield, Ohio, does not meet the agency's minimum temperature requirement of 55°F, making it unsuitable as a destination.

Given these results, the agency can confidently use the model to evaluate potential destinations. It is advised that the agency continuously refines data collection and adjusts the model to ensure precise predictions and to accommodate local microclimates, catering to specific customer preferences. Regular updates and strategic use of climatic data will further enhance the agency's offerings and customer satisfaction.

References

Becken, S. (2010). *The Importance of Climate and Weather for Tourism.* Brisbane: Land Environment & People.

Chinchwad, P. (2019). *Weather Prediction for Tourism Application using ARIMA.* Dehli: International Research Journal of Engineering and Technology (IRJET) .

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning* . Los Angeles: Springer.

Appendices

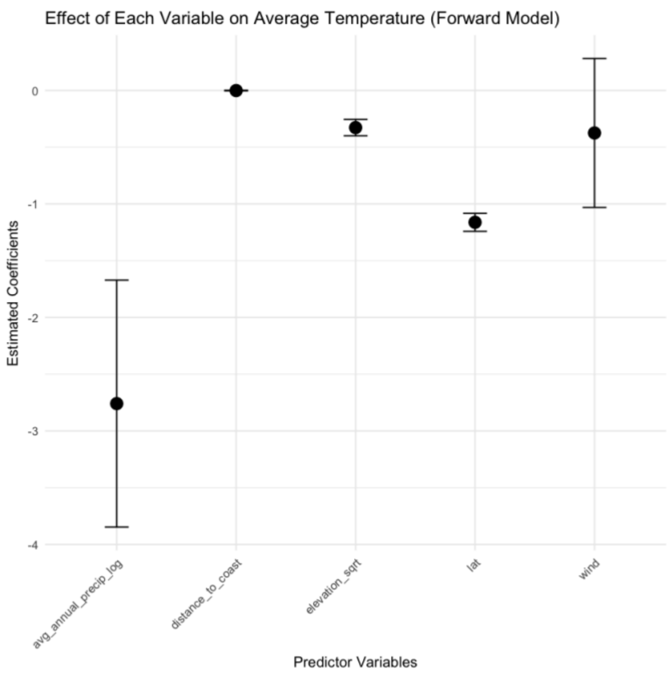


Figure 1 - Effect of Each Variable on Average Temperature

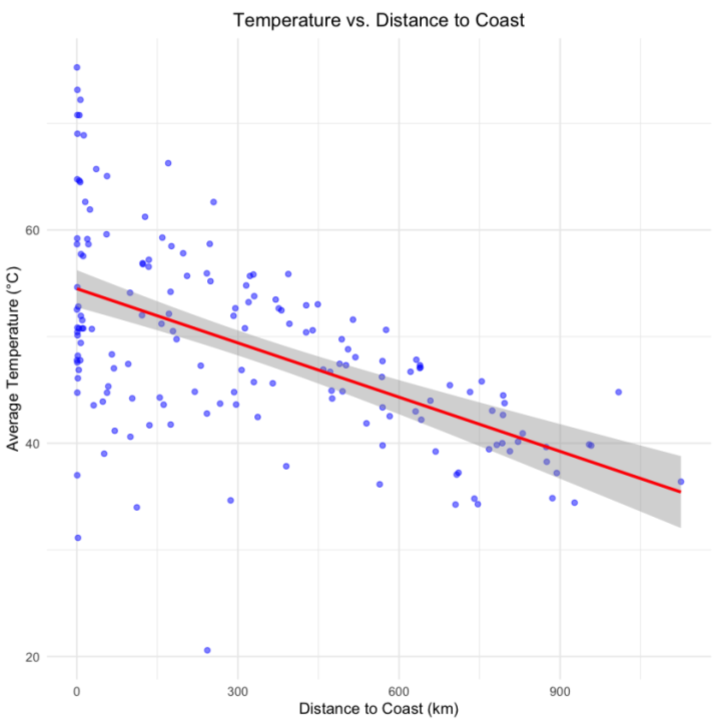


Figure 2 - Temperature vs Distance to Coast Graph