Predicting Extreme Weather for Energy Demand Management in Paris

FEM11149 - Introduction to Data Science

Michael van Walsum 756104 (25%), Aleksandra Tatko 648925 (25%)
Stijn Hooijman 620083 (25%), Francesco Di Presa 771382 (25%)

October, 2025

1. Executive Summary

As part of the data science team in an energy company, the goal was to predict extreme weather conditions in Paris one day ahead to help anticipate changes in energy demand and prevent blackouts. 12 years of daily weather data from the CERRA database was used, containing 17 meteorological variables related to temperature, wind, precipitation, and radiation. Using Principal Component Analysis (PCA), this report reduced the dataset's dimensionality and identified four key components that explained about 85% of total variance, with the first capturing roughly 48% through temperature and radiation patterns. A Principal Component Regression (PCR) model based on these components was compared with a Multiple Linear Regression (MLR) benchmark. Both performed strongly (PCR R^2 = 0.876, MLR R^2 = 0.896), with MLR slightly outperforming PCR, while PCA provided greater interpretability and robustness to multicollinearity. Overall, PCA proved valuable for summarizing complex weather data and supporting accurate next-day temperature forecasts crucial for energy planning in Paris.

2. Data Exploration

The dataset consists of 12 years (2010-2021) of daily weather observations for Paris, drawn from the CERRA reanalysis database. It includes 17 meteorological variables capturing temperature, precipitation, wind, and solar conditions, all measured with daily frequency. These variables are: mean temperature, maximum and minimum temperature, perceived temperature measures, radiation indicators, wind characteristics (speed, gusts, and direction), precipitation and snowfall measures, and daylight and sunshine duration. Descriptive statistics for all variables are provided in *Appendix 1.1*, showing a wide seasonal range in temperature and daylight duration, as well as substantial variation in wind and precipitation values across years. Exploratory analysis reveals strong linear relationships between temperature-related variables and next-day maximum temperature (*Appendix 1.2*), with correlation coefficients above 0.9 for Tmax, perceived temperatures, and mean temperature. Solar radiation and daylight duration also show positive correlations (corr = 0.6 to 0.8), while precipitation and wind variables exhibit weak to moderate negative correlations (corr = -0.1 to -0.3) (*Appendix 1.3*). These findings indicate that warm, sunny, and dry days are typically followed by higher maximum temperatures, while cooler, rainy, or windy days precede lower ones. The train-test split (80/20 chronological division) maintains the temporal structure of the data and ensures model evaluation on unseen, later years (*Appendix 1.4*). Overall, the dataset is well-suited for Principal Component Analysis (PCA) due to its high inter-variable correlations and plausible underlying weather patterns.

3. Methodology

In order to predict extreme heat one day ahead, this report will make use of Principal Component Analysis (PCA). PCA is a method that looks for underlying relations between variables in the dataset, and subsequently constructs uncorrelated "principal components" (PC) as a mixture of different variables. Thereby, PCA reduces the dimensionality of the dataset, simplifying models while retaining most explanatory power. This has a range of advantages: complex datasets can be reduced to a small number of key factors, and it removes multi-collinearity between variables, leading to better model explanatory and predictive performance, while simplifying model interpretation and actionability. Therefore, PCA is a suitable method to group and simplify complex datasets with correlated variables, and formulating them into key dimensions for further analysis. PCA does have a drawback regarding its difficulty dealing with data with extremes, which is the case in the weather-dataset in this report. Since PCA looks for trends among the entire dataset, but does not have the capability to distinguish outliers, a few extreme values can distort the found relationships within the data. Also, since PCA groups data for general trends, PCA is unable to group variables specifically for extremes. Therefore, PCA should be complemented with robust or tail-focused techniques when specifically analyzing extreme datapoints.
The report will first perform a PCA on training data, in which the best number of PCs will be decided and these PCs will be interpreted. Subsequently, the PCA's performance in weather prediction will be compared with a benchmark linear regression model, in order to see if the advantages of PCA weigh up against its disadvantages regarding extreme values in the dataset. Lastly, the sensitivity of the first PCA regression will be investigated by comparing it with PCA's with more and less PCA's, as well as with the benchmark model.

4. Results

4.1 Amount of principal components PCA was performed on the training data. One important step in PCA is the selection of the amount of PCs to use in further analysis. There are no clear-cut rules on how many PCs to use, but various methods can be used which select a number of PCs based on their explanatory power. One method uses the scree-plot, and takes the number of PCs after which an "elbow" is visible in the graph, which coincides with a significant drop-off in variance in the data explained by the next PC. The scree plot, displayed in *Appendix 1.5*, shows strong elbow points after PC2 and PC6, and arguably also a weaker elbow after PC4. Therefore, as the scree plot does not provide convincing proof that one specific number of PCs should be selected, another method should be used as well. Another method to select the number of principal components is Kaiser's Rule, which states that only PCs with an eigenvalue greater than 1 should be included. This criterion means that the retained components explain more variance in the data than the average single component. In the PCA, 4 PCs have an eigenvalue of larger than 1, as visible in *Appendix 1.6*, suggesting that 4 PCs should be used in the further analysis. To further investigate the validity of this finding, permutation was performed to evaluate how large the eigenvalue of 4 PCs would be if data is randomly distributed. The results are displayed in *Appendix 1.6*, and show that the first 4 PCs explain genuine variation in the data with a significantly higher eigenvalue than random data. Therefore, permutation confirmed that choosing 4 PCs is justified. This analysis proceeds with four principal components, as this number is supported by both Kaiser's rule and the permutation method. Together, these four components capture approximately 85% of the total variance in the dataset, indicating that they preserve most of the original information while filtering out noise. Retaining four components therefore provides a strong balance between explanatory power and model simplicity, ensuring effective dimensionality reduction without significant loss of data representativeness.

4.2 Principal component interpretation By examining the composition of the principal components, it is possible to interpret what each one represents. The relationships among variables are illustrated in *Appendix 1.7*, which displays the PCA biplot of the first two components, while *Appendix 1.8* provides the factor loadings for all four. PC1 primarily reflects temperature and radiation patterns, showing strong positive loadings for temperature-related and solar radiation variables. PC2 includes strong positive loadings for precipitation and to a slightly lesser extent wind-related variables, capturing variability linked to storms or other unsettled weather conditions. PC3 could correspond to conditions with very strong winds, since it has the strongest factor loadings with windy conditions. PC4 mainly captures winter-related variation, with strong factor loadings among snowfall and a (lack of) sunshine. However, the high factor loading of dominant wind direction is difficult to interpret in this component. We can see that PC3 and PC4 show slightly less interpretable results (*Appendix 1.8*), which is to be expected; later PCs capture less of the organized variance and are therefore less clear in their underlying relationship. Also, since the interpretation of the different factor loadings is subjective, it is possible that different researchers interpret the PCs in different ways. Based on the clearness of their interpretation, one can see that overall, PC1 and PC2 summarize the main structure of the data, while later PCs capture lesser variation, possibly linked with extreme winds and seasonal weather trends. Further examination of PC1 shows that it explains around 48.1% of the total variance in the dataset. Bootstrap was used to construct a confidence interval of this amount of variance explained. Specifically, moving block bootstrap (with a block length of 14 days) was used instead of normal bootstrap, as the assumption of independent observations in normal bootstrap was violated due to large autocorrelation in the dataset. Using this bootstrap method, a 95% confidence interval of (46.728%, 49.295%) was found. To identify which variables are best captured by PC1, the squared correlations between PC1 and the original variables were calculated (see *Appendix 1.9*). The results show that PC1 explains most of the variation in temperature-related variables—over 90% for maximum and mean temperatures—and a substantial portion of the variation in radiation-related variables.

4.3 Comparison of predictive capability by model The PCs formulated in the analysis so far were fitted in a principal component regression (PCR) on maximum temperature in the training dataset. The same was done for a multiple linear regression (MLR) including all variables as independent variables. Then, these two models were tested on the testing dataset. The predictive performance of the models was evaluated using three common metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 (see *Table 1*). MAE measures the average size of prediction errors, providing an intuitive sense of how far predictions deviate from actual values on average. RMSE also measures prediction error but gives greater weight to large deviations, making it more sensitive to outliers. R^2 indicates how much of the variance in the dependent variable is explained by the model, with values closer to 1 reflecting better fit. Together, these metrics give a balanced view of model accuracy, consistency, and explanatory power. MLR slightly outperforms PCR in predictive accuracy, likely because the dataset has strong linear relationships and relatively low noise, so dimensionality reduction offers little benefit here. However, the PCR provides an advantage in interpretability and robustness against multicollinearity among predictors, which is valuable when many variables are highly correlated, as in the case for weather data. These results are consistent with expectations: when predictors are strongly correlated but the signal is mostly linear, MLR and PCR tend to yield comparable accuracy. PCR sacrifices a bit of precision for stability and parsimony.

Table 1: Comparison of PCR and MLR model performance on test data.

	Model	RMSE	MAE	R2
1	PCA (4 PCs)	2.652	2.064	0.876
2	Multiple Linear Regression	2.423	1.879	0.896

To investigate PCR sensitivity to the number of PCs, a sensitivity analysis was performed to test predictive performance among a PCR with 3 and 5 PCs, with the predictive capabilities of the three models compared in *Appendix 1.10.* The results indicate that the PCR model is not highly sensitive to small changes in the number of PCs. Most predictive information is already captured by the first PCs including "temperature-radiation" and "precipitation/storm" axes, consistent with previous PCA results as shown in the scree plot in *Appendix 1.5.* Using 4 PCs provides the best trade-off between simplicity and performance with the lowest prediction error, while additional PCs add minimal explanatory value and could start capturing noise. *Figure 1* reports the MLR and best PCR (4 components) predictive capability for different temperature deciles. Prediction accuracy is strong and stable for typical temperature conditions, but both models have difficulty at the extremes, which is expected, since extreme weather events are rare, deviate from the general trend in the entire dataset evaluated by both MLR and PCR, and are harder to capture with linear methods. PCR does show a small robustness advantage regarding extreme temperatures, which is likely thanks to its use of orthogonal principal components that reduce multicollinearity effects.

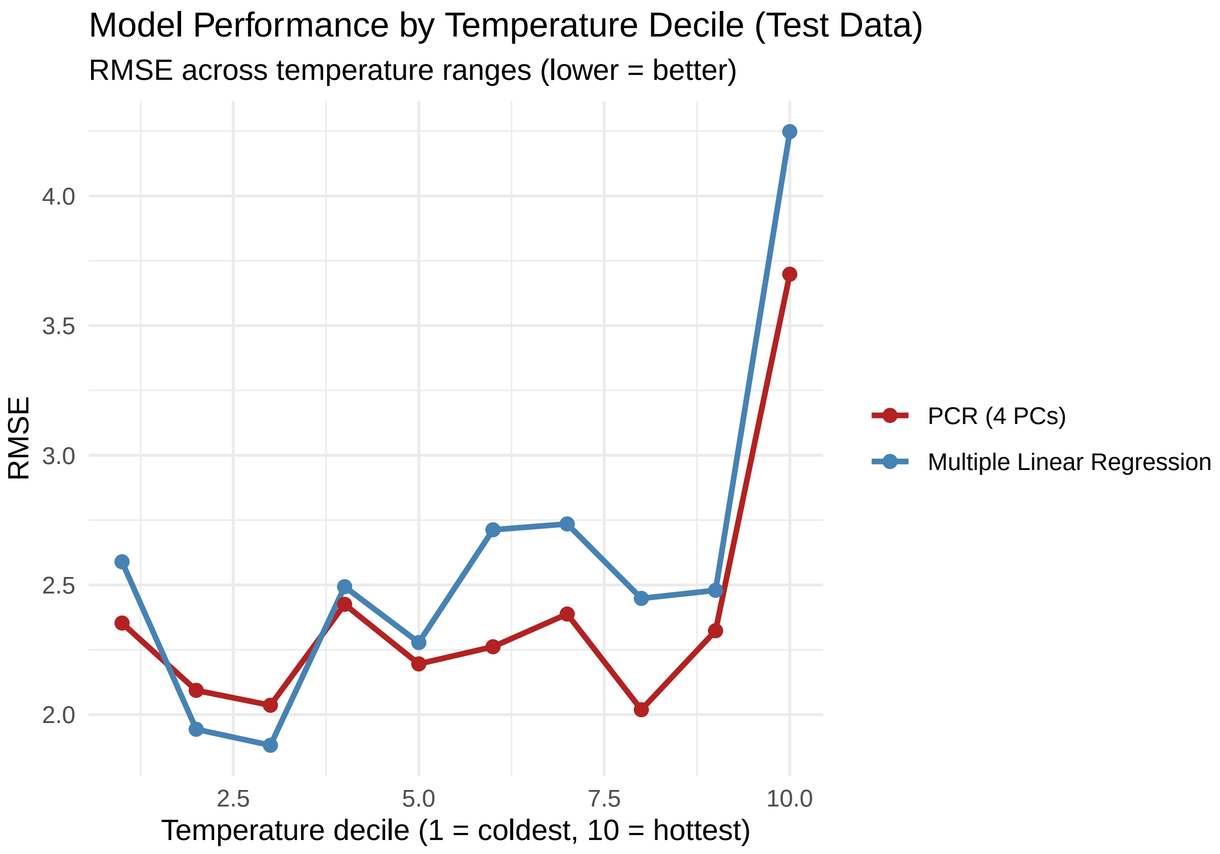


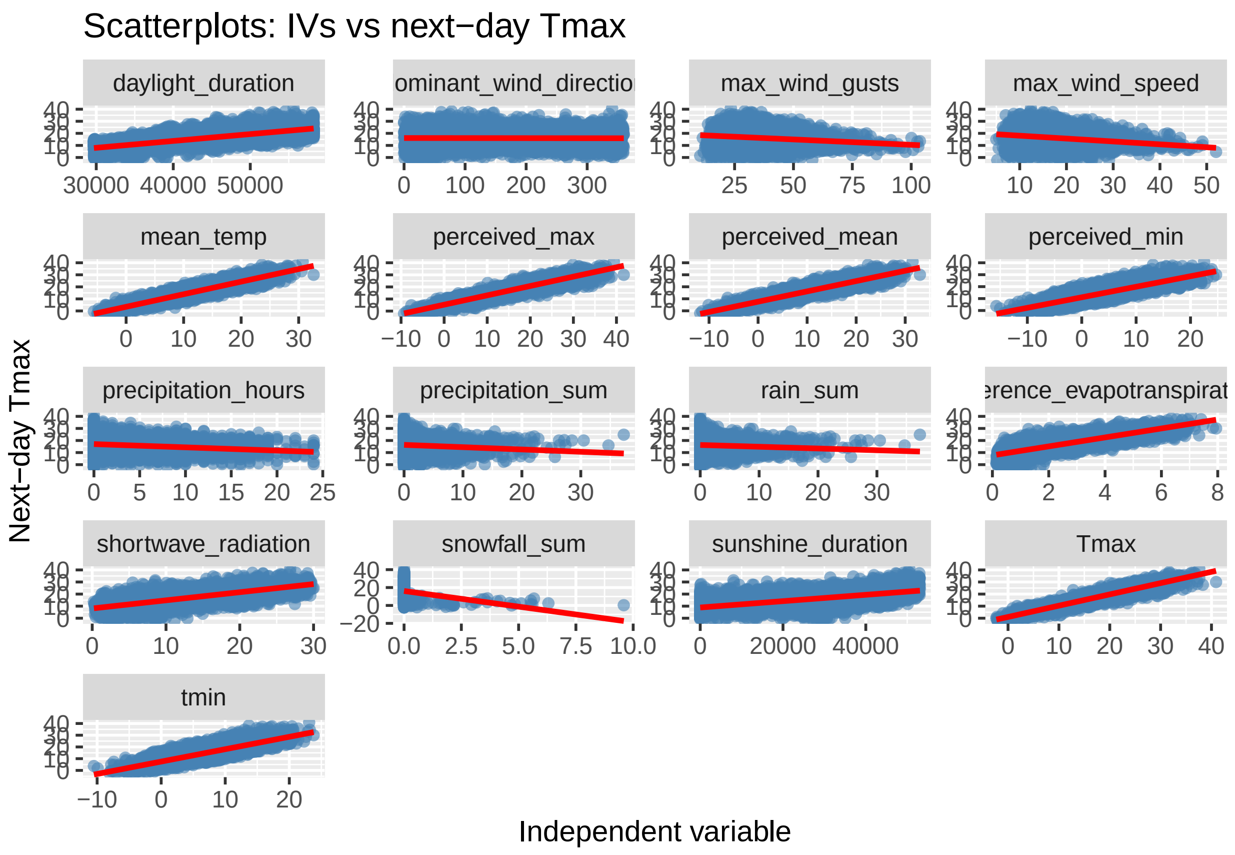
Figure 1. RMSE per model per temperature decile

5. Conclusion

Principal Component Analysis proved to be a valuable approach for reducing dimensionality in this dataset, mainly because many of the meteorological variables were highly correlated. By summarizing these variables into a few uncorrelated components, PCA simplified the modeling process and helped avoid issues with multicollinearity. However, given that the dataset was relatively clean and the main predictors of temperature were strongly linear, the dimensionality reduction did not lead to large improvements in predictive accuracy compared to standard Multiple Linear Regression (MLR). The predictive performance of both models was found to be quite stable for moderate temperature levels but less accurate for extreme highs and lows. This decrease in accuracy is likely due to the fact that extreme temperature events are rare and influenced by non-linear or local factors that the linear models cannot fully capture. One way to address this limitation would be to apply non-linear or interaction-based approaches, such as polynomial regression or tree-based methods, which can better model such irregularities. If penalized regression methods such as ridge or lasso were applied here, they would likely perform similarly to PCA in handling multicollinearity, but with the added benefit of retaining interpretability at the variable level. As shown in Assignment 1, ridge regression effectively stabilized coefficient estimates in the presence of correlated predictors, while lasso achieved additional variable selection by setting some coefficients to zero. This suggests that, in the current weather dataset, penalized regression could have produced comparable or slightly better predictive accuracy than PCA by balancing multicollinearity control with direct interpretability of the original predictors. Lasso, in particular, could have provided a more efficient model by shrinking less important predictors toward zero, effectively performing variable selection while maintaining predictive accuracy.

6. Appendix

Appendix 1.1 Scatterplots of independent variables versus next-day maximum temperature (Tmax).



Each panel shows the bivariate relationship between one predictor variable and next-day Tmax, with a fitted linear regression line (in red). Temperature-related variables (e.g., tmin, mean_temp, perceived_max) display strong positive correlations with Tmax, while wind speed, gusts, and precipitation-related variables exhibit weaker or negative relationships. These plots highlight that many predictors are linearly correlated with the target variable, suggesting potential multicollinearity that motivates the use of Principal Component Analysis (PCA) for dimensionality reduction.

Appendix 1.2 Summary Statistics

Table 2: Summary Statistics of Weather Variables (Paris, 2010–2021)

Statistic	Min	Pctl(25)	Median	Mean	Pctl(75)	Max	St. Dev.
Location.ID	0	0	0	0.00	0	0	0.00
Mean.temperature...C.	−5.60	7.20	11.90	12.05	17.10	32.60	6.45
Max..temperature...C.	−2.30	10.50	15.90	16.02	21.50	40.90	7.31
Min..temperature...C.	−10.50	3.70	8.10	8.05	12.70	23.80	5.83
Perceived.mean.temperature...C.	−11.80	3.70	9.40	9.81	16.00	33.00	7.84
Perceived.max..temperature...C.	−9.30	7.00	13.50	13.87	20.40	41.70	8.76
Perceived.min..temperature...C.	−15.70	0.10	5.50	5.60	11.30	24.70	7.11
Max..wind.speed..km.h.	5.00	13.20	17.60	18.52	22.50	52.00	7.03
Max..wind.gusts..km.h.	10.40	27.70	36.00	38.04	45.70	103.70	14.01
Shortwave.radiation.sum..MJ.m2.	0.23	4.60	10.84	11.78	18.13	30.01	7.83
Dominant.wind.direction....	0	109	213.5	190.90	264	360	101.60
Reference.evapotranspiration..mm.	0.14	0.81	1.90	2.24	3.39	7.94	1.61
Daylight.duration..s.	29,697.65	34,730.94	44,230.92	44,105.05	53,505.20	58,227.04	9,743.16
Sunshine.duration..s.	0.00	14,043.92	28,744.14	27,169.37	41,262.97	53,074.48	16,224.95
Precipitation.sum..mm.	0.00	0.00	0.20	1.89	2.20	37.30	3.59
Snowfall.sum..mm.	0.00	0.00	0.00	0.04	0.00	9.59	0.36
Precipitation.hours..h.	0	0	1	3.73	7	24	4.86
Rain.sum..mm.	0.00	0.00	0.20	1.84	2.10	37.30	3.54

The table reports minimum, 25th percentile, median, mean, 75th percentile, maximum, and standard deviation values for all weather variables. The data show large variability across temperature and radiation variables, consistent with expected seasonal patterns. Wind speed and precipitation variables display lower central values but higher relative dispersion, suggesting irregular occurrences of storms or extreme events.

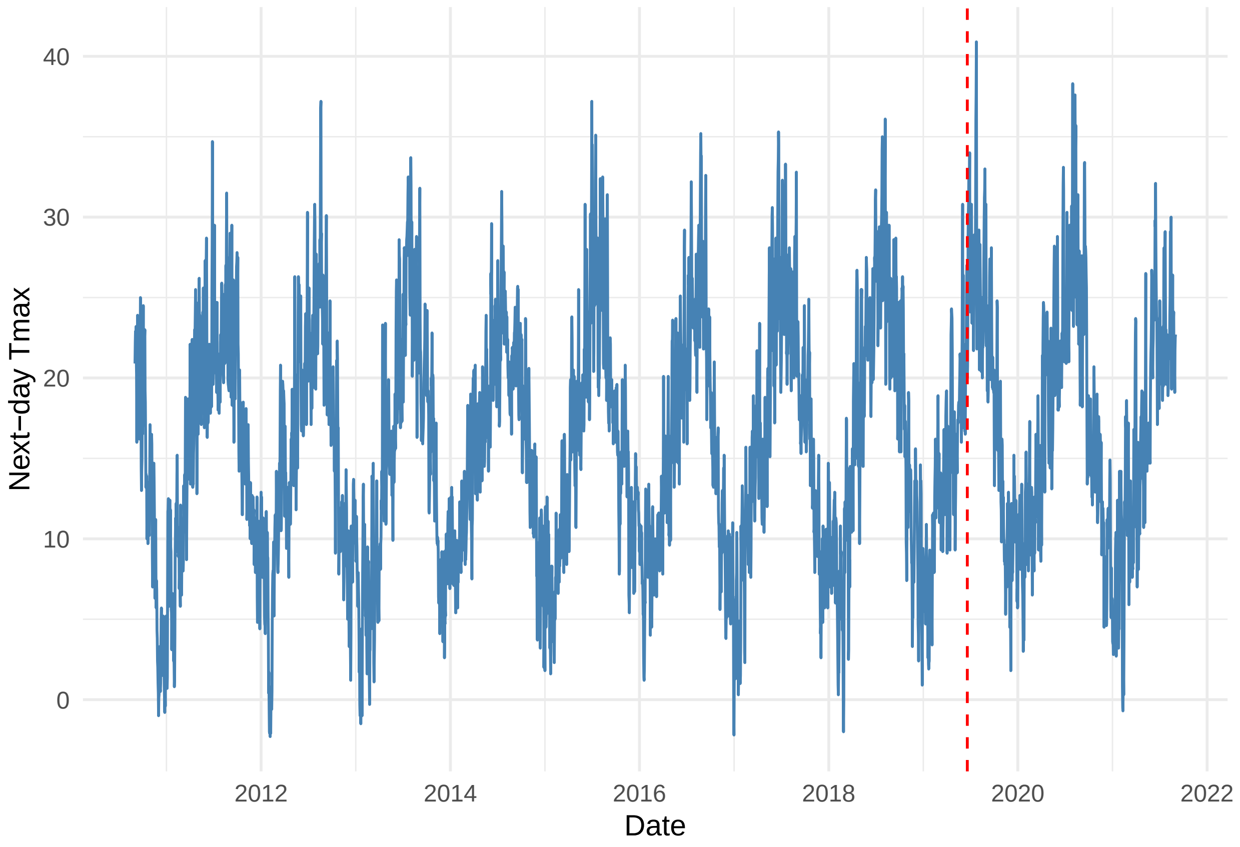
Appendix 1.3 Correlations

Table 3: Correlation of Independent Variables with Next-day Maximum Temperature (Tmax)

	Variable	Correlation
1	Tmax	0.93
2	perceived_max	0.93
3	mean_temp	0.92
4	perceived_mean	0.92
5	perceived_min	0.86
6	tmin	0.84
7	reference_evapotranspiration	0.82
8	daylight_duration	0.76
9	shortwave_radiation	0.72
10	sunshine_duration	0.58
11	dominant_wind_direction	-0.01
12	rain_sum	-0.07
13	precipitation_sum	-0.09
14	max_wind_gusts	-0.17
15	snowfall_sum	-0.17
16	precipitation_hours	-0.18
17	max_wind_speed	-0.23

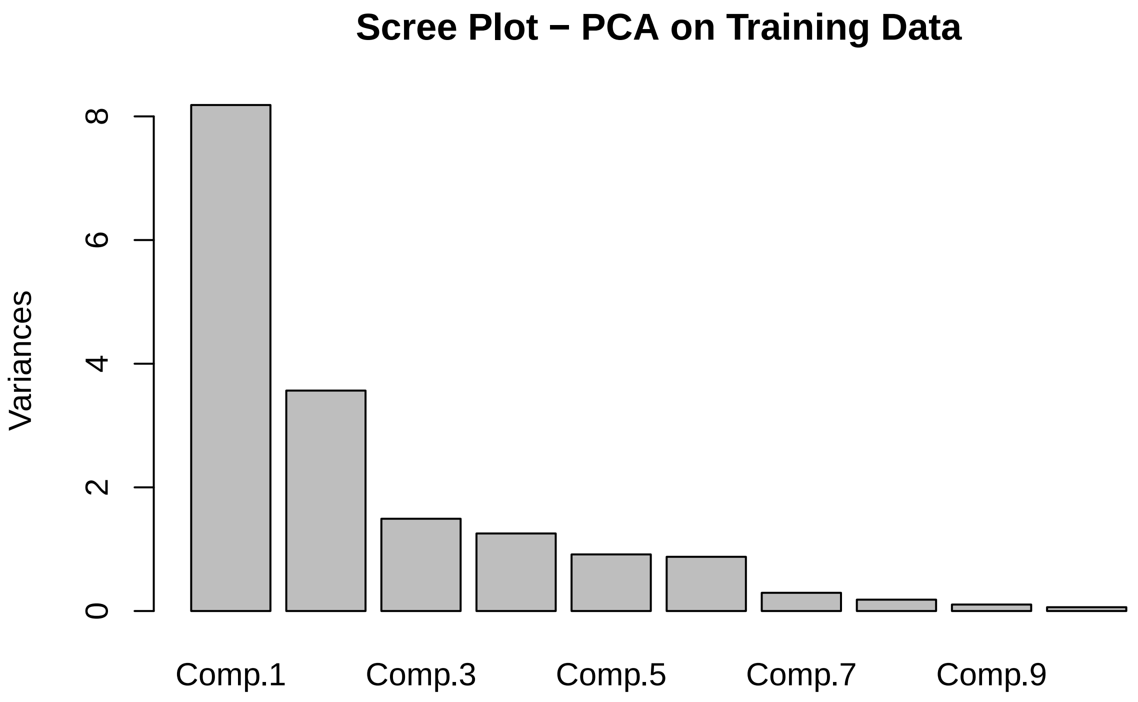
The table lists Pearson correlation coefficients between each predictor and the next-day Tmax. Strong positive associations are found with temperature and radiation-related variables (e.g., mean_temp, perceived_max, reference_evapotranspiration), while wind speed, precipitation, and snowfall show weak to moderate negative correlations. These results confirm substantial multicollinearity among temperature indicators, motivating the later application of PCA.

Appendix 1.4 Train/Test Split Visualization



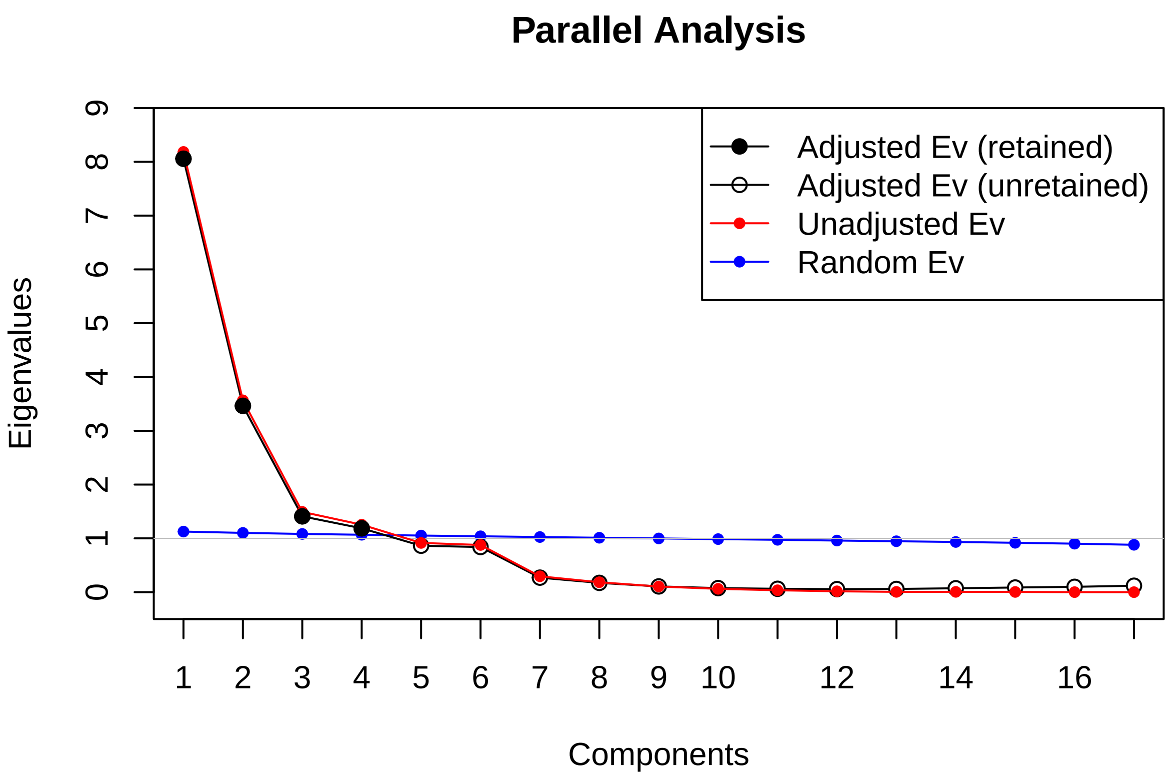
The plot shows daily Tmax values across the study period, exhibiting a clear annual temperature cycle with regular seasonal peaks and troughs. The red dashed line indicates the split between training (2010–2018) and test (2019–2021) datasets used for model validation. This approach preserves temporal ordering and prevents data leakage between training and evaluation sets.

Appendix 1.5 Scree plot of principal component variances for the training dataset.



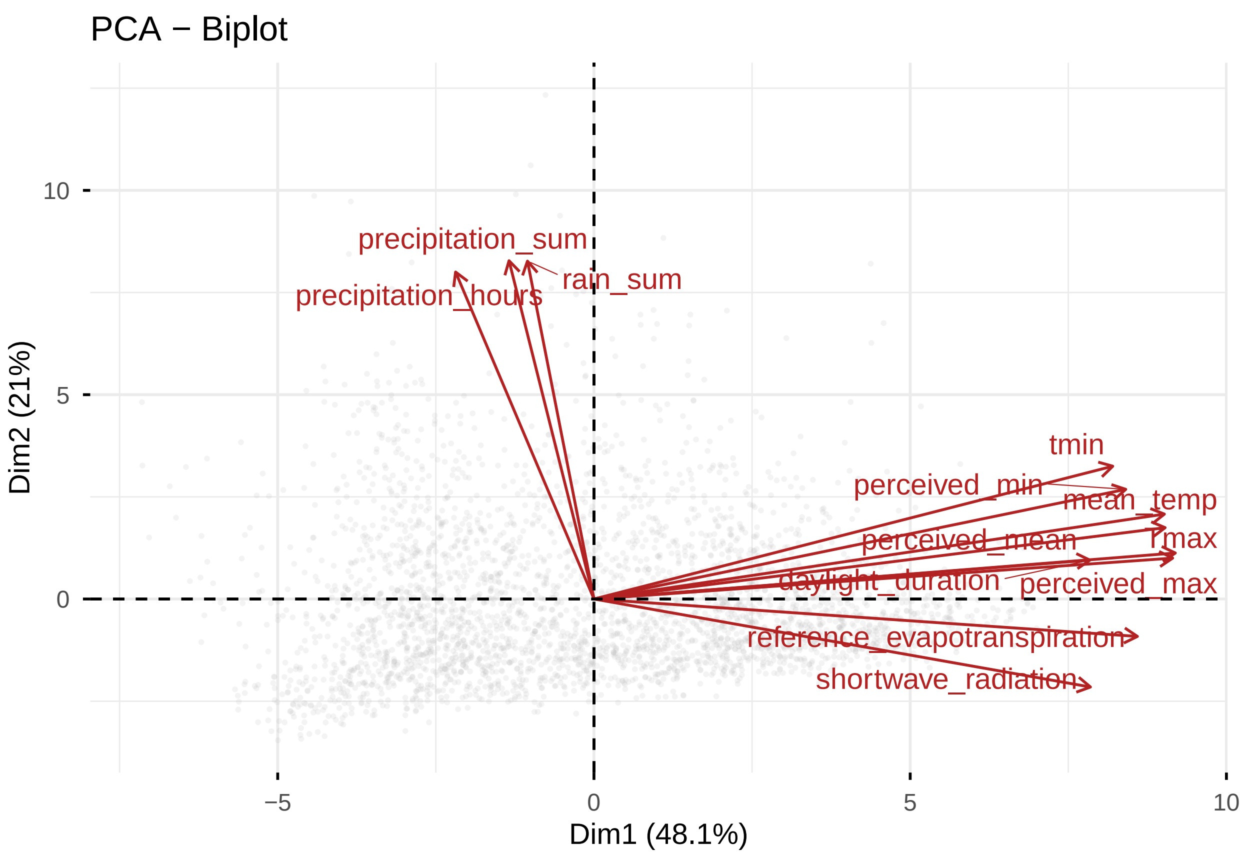
The bar chart displays the variance explained by each principal component. A sharp decline occurs after the second and sixth components, suggesting diminishing returns from adding further components. This "elbow" criterion helps identify how many components contribute meaningfully to the total variance.

Appendix 1.6 Parallel analysis of principal component eigenvalues.



The parallel analysis compares observed eigenvalues (black and red lines) to those expected under random data (blue line). Only the first four adjusted eigenvalues exceed the random baseline, indicating that four components capture genuine structure rather than random noise in the data.

Appendix 1.7 PCA biplot of the first two principal components.



Arrows represent variable loadings on the first two components, which together explain 69.1 % of the total variance (48.1 % for PC1 and 21 % for PC2). Temperature-related and radiation variables cluster on the right, while precipitation and wind variables align in the upper-left quadrant. The plot illustrates that PC1 primarily reflects temperature–radiation variability, whereas PC2 captures precipitation and wind effects.

Appendix 1.8 Factor loadings in principal components

Table 4: Principal component loadings for the first four components.

	Comp.1	Comp.2	Comp.3	Comp.4
mean_temp	0.33	0.12	0.03	0.13
Tmax	0.34	0.06	0.01	0.05
tmin	0.30	0.18	0.06	0.22
perceived_mean	0.33	0.10	0.11	0.12
perceived_max	0.34	0.06	0.10	0.05
perceived_min	0.31	0.15	0.12	0.21
max_wind_speed	-0.09	0.32	-0.59	0.09
max_wind_gusts	-0.07	0.33	-0.59	0.07
shortwave_radiation	0.29	-0.12	-0.19	-0.34
dominant_wind_direction	0	0.14	-0.04	0.46
reference_evapotranspiration	0.32	-0.05	-0.21	-0.21
daylight_duration	0.29	0.05	-0.06	-0.24
sunshine_duration	0.25	-0.18	-0.25	-0.30
precipitation_sum	-0.05	0.46	0.22	-0.27
snowfall_sum	-0.07	0.04	0.04	-0.45
precipitation_hours	-0.08	0.45	0.15	-0.18
rain_sum	-0.04	0.46	0.21	-0.20

Each cell shows the correlation between an original weather variable and a principal component. High loadings in PC1 are dominated by temperature and radiation measures, PC2 reflects precipitation and wind intensity, PC3 emphasizes strong winds, and PC4 corresponds mainly to winter-related conditions.

Appendix 1.9 proportion of variance explained by the first principal component

Table 5: Top 10 Variables Best Explained by the First Principal Component (PC1)

	Variable	Explained_Variance
1	Tmax	0.944
2	perceived_max	0.936
3	perceived_mean	0.911
4	mean_temp	0.909
5	reference_evapotranspiration	0.825
6	perceived_min	0.790
7	tmin	0.752
8	shortwave_radiation	0.689
9	daylight_duration	0.687
10	sunshine_duration	0.496

The table ranks the top ten weather variables by the proportion of their variance captured by PC1. The first component is primarily associated with maximum and mean temperatures, followed by radiation-related measures, indicating that PC1 represents overall temperature–radiation intensity.

Appendix 1.10 Prediction accuracy per PCA, measured in RMSE, MAE and R²

Table 6: Sensitivity analysis of PCR model performance for different numbers of principal components.

	Components	RMSE	MAE	R2
1	3	2.668	2.080	0.874
2	4	2.652	2.064	0.876
3	5	2.653	2.066	0.876

Appendix 2. All R Code

```r
## ----echo = FALSE, eval = FALSE, warning = FALSE, message = FALSE-------------
# # Directory setup
# path = dirname(rstudioapi::getSourceEditorContext()$path) # Path is directory of this file
# setwd(path)                                               # Set working directory


## ----include=FALSE-------------------------------------------------------------
# --- Group Assignment 2 -------------------------------------------------------

# --- Libraries ----------------------------------------------------------------
library(dplyr)
library(tidyr)
library(ggplot2)
library(factoextra)
library(pls)
library(boot)
library(paran)
library(stargazer)
set.seed(123)

# --- Load & prepare data ------------------------------------------------------
data <- read.csv("a2_data_group_2.csv")


data <- data[,-1]

names(data) <- c(
  "Date",
  "mean_temp",
  "Tmax",
  "tmin",
  "perceived_mean",
  "perceived_max",
  "perceived_min",
  "max_wind_speed",
  "max_wind_gusts",
  "shortwave_radiation",
  "dominant_wind_direction",
  "reference_evapotranspiration",
  "daylight_duration",
  "sunshine_duration",
  "precipitation_sum",
  "snowfall_sum",
  "precipitation_hours",
  "rain_sum"
)

names(data)[names(data) == "Max..temperature...C."] <- "Tmax"
data$Date <- as.Date(data$Date, format = "%Y-%m-%d")

# --- Task 1: Build independent (X) and dependent (y) variables ---------------
# X(t) uses ALL variables except the last row; y(t+1) is Tmax excluding the first row
X      <- data[-nrow(data), ]   # all but last row #-1
y      <- data$Tmax[-1]         # Tmax shifted one day ahead
dates  <- data$Date[-1]         # matching dates for y

# Keep numeric predictors for later steps (+ drop zero-variance)
X_num <- X[sapply(X, is.numeric)]
X_num <- X_num[, sapply(X_num, function(col) sd(col, na.rm = TRUE) > 0)]
y_num  <- as.numeric(y)

# --- Task 2: Exploratory analysis --------------------------------------------
# Scatterplots of each independent variable vs. next-day Tmax
df_plot <- cbind(X, y) %>% dplyr::select(where(is.numeric))
df_plot %>%
  pivot_longer(-y, names_to = "variable", values_to = "value") %>%
  ggplot(aes(value, y)) +
  geom_point(alpha = 0.6, color = "steelblue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  facet_wrap(~ variable, scales = "free", ncol = 4) +
  labs(x = "Independent variable", y = "Next-day Tmax",
       title = "Scatterplots: IVs vs next-day Tmax")

# Compute correlations
correlations <- sapply(X_num, function(col) cor(col, y_num, use = "pairwise.complete.obs"))
round(sort(correlations, decreasing = TRUE), 2)

# Convert to a data frame for ggplot
cor_df <- data.frame(
  Variable = names(correlations),
  Correlation = as.numeric(correlations)
)

# Plot
ggplot(cor_df, aes(x = reorder(Variable, Correlation), y = Correlation)) +
  geom_col(fill = "skyblue") +
  coord_flip() +
  theme_minimal() +
  labs(
    title = "Correlation of Independent Variables with Next-day Max Temperature",
    x = "Variable",
    y = "Correlation coefficient"
  )
```

```r
# --- Task 3: Train-Test Split (Time Series) ---------------------------------
# Chronological split (no shuffling): 80% train, 20% test
df <- data.frame(Date = dates, X_num, y = y_num)
n  <- nrow(df); train_size <- floor(0.8 * n)
train <- df[1:train_size, ]      # earlier observations
test  <- df[(train_size + 1):n, ] # later observations

# Visual check of the split (red dashed line = boundary)
ggplot(df, aes(Date, y)) +
  geom_line(color = "steelblue") +
  geom_vline(xintercept = as.numeric(max(train$Date)),
             linetype = "dashed", color = "red") +
  labs(title = "Train/Test Split Check",
       subtitle = "Red dashed line = boundary between training and test sets",
       y = "Next-day Tmax")

# --- Task 4: Principal Component Analysis (PCA) ------------------------------
# PCA on TRAINING independent variables only (no Date, no y)
X_train <- train[, sapply(train, is.numeric)]
X_train <- X_train[, setdiff(names(X_train), "y")]

# Run PCA using correlation matrix (standardizes variables)
res <- princomp(X_train, cor = TRUE, scores = TRUE)
summary(res)

# Scree plot + quick look at loadings (PC1-PC2)
screeplot(res, main = "Scree Plot - PCA on Training Data")
round(res$loadings[, 1:4], 2)

#Parallel Analysis
paran(X_train, graph = TRUE)
##### comment #####

# Cleaner biplots (variables only + biplot with top contributors)
fviz_pca_var(res, repel = TRUE, col.var = "contrib")
fviz_pca_biplot(res,
                geom.ind = "point", pointshape = 16, pointsize = 0.7, alpha.ind = 0.15,
                col.ind = "grey70", col.var = "firebrick", repel = TRUE,
                select.var = list(contrib = 12)  # label top 12 vars
)

# Apply Kaiser's rule (keep eigenvalues > 1)
eigenvalues <- res$sdev^2
kaiser_components <- sum(eigenvalues > 1)
cat("Number of components with eigenvalue > 1:", kaiser_components, "\n")

# --- Task 7: Biplot and Interpretation of Principal Components ---------------
loadings_df <-round(res$loadings[, 1:4], 2)

# --- Task 8 (Even group)
# 1) Point estimate (train data)
pc1_var <- (res$sdev[1]^2) / sum(res$sdev^2)
pc1_var  #  0.48
```

```r
# 2) Moving-block bootstrap CI for PC1 proportion
stat_pc1 <- function(x, i) {
  xb <- x[i, , drop = FALSE]
  p  <- try(princomp(xb, cor = TRUE), silent = TRUE)
  if (inherits(p, "try-error")) return(NA_real_)
  (p$sdev[1]^2) / sum(p$sdev^2)
}
L <- 14  # block length (~2 weeks). Robust for L in {7, 14, 21}
B <- 1000
bt <- tsboot(tseries = as.matrix(X_train), statistic = stat_pc1,
             R = B, l = L, sim = "fixed")
pc1_ci_block <- quantile(bt$t[is.finite(bt$t)], c(0.025, 0.975), na.rm = TRUE)
pc1_ci_block

# 3) Variables best explained by PC1 (squared correlation between PC1 and vars)
corpca <- cor(X_train,res$scores)[,1]
explained_by_pc1 <- sort(corpca^2, decreasing=TRUE)
head(explained_by_pc1,10)


# --- Task 9: Principal Component Regression (PCR) ----------------------------
# Time-aware CV; compare CV-selected ncomp vs fixed k = 4 from Task 6
train_pcr <- subset(train, select = -Date)
test_pcr  <- subset(test,  select = -Date)

pcr_cv <- pcr(y ~ ., data = train_pcr, scale = TRUE,
              validation = "CV", segments = 10, segment.type = "consecutive")
validationplot(pcr_cv, val.type = "RMSEP")
n_cv <- selectNcomp(pcr_cv, method = "onesigma", plot = FALSE)

pcr_final_cv <- pcr(y ~ ., data = train_pcr, scale = TRUE, ncomp = n_cv)
pcr_final_k  <- pcr(y ~ ., data = train_pcr, scale = TRUE, ncomp = 4)

pred_cv <- as.numeric(predict(pcr_final_cv, newdata = test_pcr, ncomp = n_cv))
pred_k  <- as.numeric(predict(pcr_final_k,  newdata = test_pcr, ncomp = 4))

rmse <- function(e) sqrt(mean(e^2))
mae  <- function(e) mean(abs(e))
r2   <- function(y, yhat) 1 - sum((y - yhat)^2) / sum((y - mean(y))^2)

y_test <- test$y
metrics <- list(
  CV_ncomp = n_cv,
  CV_RMSE  = rmse(y_test - pred_cv),
  CV_MAE   = mae(y_test - pred_cv),
  CV_R2    = r2(y_test, pred_cv),
  K_used   = 4,
  K_RMSE   = rmse(y_test - pred_k),
  K_MAE    = mae(y_test - pred_k),
  K_R2     = r2(y_test, pred_k)
)
metrics
```

```r
# --- Task 10: Benchmark Multiple Linear Regression (MLR) --------------------
lm_model <- lm(y ~ ., data = train_pcr)
summary(lm_model)
pred_lm  <- predict(lm_model, newdata = test_pcr)

lm_metrics <- list(
  RMSE = rmse(y_test - pred_lm),
  MAE  = mae(y_test - pred_lm),
  R2   = r2(y_test, pred_lm)
)
lm_metrics

# --- Task 11: Model Comparison on Test Data ---------------------------------
comparison <- data.frame(
  Model = c("PCA (4 PCs)", "Multiple Linear Regression"),
  RMSE  = c(metrics$K_RMSE, lm_metrics$RMSE),
  MAE   = c(metrics$K_MAE,  lm_metrics$MAE),
  R2    = c(metrics$K_R2,   lm_metrics$R2)
)
comparison

comparison_long <- comparison %>%
  pivot_longer(cols = -Model, names_to = "Metric", values_to = "Value")
ggplot(comparison_long, aes(x = Metric, y = Value, fill = Model)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "PCA vs. Multiple Linear Regression on Test Data",
       subtitle = "Performance comparison on held-out period",
       y = "Metric value") +
  theme_minimal()

# --- Task 12: Sensitivity Analysis of PCR to Number of Components ------------
k_values <- c(3, 4, 5)
pcr_sensitivity <- data.frame()
for (k in k_values) {
  model <- pcr(y ~ ., data = train_pcr, scale = TRUE, ncomp = k)
  preds  <- as.numeric(predict(model, newdata = test_pcr, ncomp = k))
  rmse_k <- rmse(y_test - preds)
  mae_k  <- mae(y_test - preds)
  r2_k   <- r2(y_test, preds)

  pcr_sensitivity <- rbind(pcr_sensitivity,
                           data.frame(
                             Components = k,
                             RMSE = rmse_k,
                             MAE = mae_k,
                             R2  = r2_k
                           ))
}

pcr_sensitivity

# Visualize the sensitivity
ggplot(pcr_sensitivity, aes(x = Components, y = RMSE)) +
  geom_line(color = "steelblue", linewidth = 1) +
  geom_point(size = 3, color = "firebrick") +
  labs(
    title = "PCR Sensitivity to Number of Components",
    subtitle = "RMSE on test data for k = 3, 4, 5",
    y = "Test RMSE"
  ) +
  expand_limits(y = c(min(pcr_sensitivity$RMSE) - 0.05,
                      max(pcr_sensitivity$RMSE) + 0.05)) +
  theme_minimal()



## ----include=FALSE------------------------------------------------------------
# --- Task 13: Performance Across Temperature Levels --------------------------

# 1) Create decile groups based on true Tmax in test data
test_results <- data.frame(
  True = y_test,
  PCR_4 = pred_k,
  MLR = pred_lm
)

test_results <- test_results %>%
  mutate(Decile = ntile(True, 10))  # 10 roughly equal groups (10% each)

# 2) Compute RMSE per decile for both models
rmse <- function(e) sqrt(mean(e^2))

group_metrics <- test_results %>%
  group_by(Decile) %>%
  summarise(
    RMSE_PCR = rmse(True - PCR_4),
    RMSE_MLR = rmse(True - MLR),
    Mean_Temp = mean(True),
    .groups = "drop"
  ) %>%
  pivot_longer(cols = starts_with("RMSE"), names_to = "Model", values_to = "RMSE")

# 3) Plot RMSE by temperature group
ggplot(group_metrics, aes(x = Decile, y = RMSE, color = Model, group = Model)) +  #### !!!!! ISSUE WITH
  geom_line(linewidth = 1) +
  geom_point(size = 2) +
  scale_color_manual(values = c("firebrick", "steelblue"),
                     labels = c("PCR (4 PCs)", "Multiple Linear Regression")) +
  labs(
    title = "Model Performance by Temperature Decile (Test Data)",
    subtitle = "RMSE across temperature ranges (lower = better)",
    x = "Temperature decile (1 = coldest, 10 = hottest)",
    y = "RMSE"
  ) +
  theme_minimal() +
  theme(legend.title = element_blank())
```

```r
## ----echo=FALSE, message=FALSE, warning=FALSE, results='asis'------------------

# Prepare clean model comparison table
comparison_df <- data.frame(
  Model = c("PCA (4 PCs)", "Multiple Linear Regression"),
  RMSE  = round(c(metrics$K_RMSE, lm_metrics$RMSE), 3),
  MAE   = round(c(metrics$K_MAE,  lm_metrics$MAE), 3),
  R2    = round(c(metrics$K_R2,   lm_metrics$R2), 3),
  stringsAsFactors = FALSE
)

# Remove any row names to avoid duplication
rownames(comparison_df) <- NULL

# Generate LaTeX table (simple, clean style)
stargazer(
  comparison_df,
  summary = FALSE,
  title = "Comparison of PCR and MLR model performance on test data.",
  label = "tab:model_comparison",
  digits = 3,
  font.size = "small",
  float.env = "table",
  type = "latex",
  header = FALSE,
  table.placement = "!h",
  align = FALSE
)



## ----echo=FALSE, message=FALSE, warning=FALSE, results='asis'------------------
ggplot(group_metrics, aes(x = Decile, y = RMSE, color = Model, group = Model)) +
  geom_line(linewidth = 1) +
  geom_point(size = 2) +
  scale_color_manual(values = c("firebrick", "steelblue"),
                     labels = c("PCR (4 PCs)", "Multiple Linear Regression")) +
  labs(
    title = "Model Performance by Temperature Decile (Test Data)",
    subtitle = "RMSE across temperature ranges (lower = better)",
    x = "Temperature decile (1 = coldest, 10 = hottest)",
    y = "RMSE"
  ) +
  theme_minimal() +
  theme(legend.title = element_blank())


## ----echo=FALSE, warning=FALSE, message=FALSE, results='asis'------------------
# ------------THE CODE BELOW IS THE CODE FOR THE TABLES INCLUDED IN THE APPENDIX ----------------------
# Scatterplots of each independent variable vs. next-day Tmax
df_plot <- cbind(X, y) %>% dplyr::select(where(is.numeric))
df_plot %>%
  pivot_longer(-y, names_to = "variable", values_to = "value") %>%
  ggplot(aes(value, y)) +
  geom_point(alpha = 0.6, color = "steelblue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  facet_wrap(~ variable, scales = "free", ncol = 4) +
  labs(x = "Independent variable", y = "Next-day Tmax",
       title = "Scatterplots: IVs vs next-day Tmax")


## ----echo=FALSE, warning=FALSE, message=FALSE, results='asis'-----------------
data <- read.csv("a2_data_group_2.csv")
data <- data[, !(names(data) %in% c("X", "Unnamed..0"))]
data_num <- data[sapply(data, is.numeric)]

cat("\\begin{center}")
stargazer(
  data_num,
  type = "latex",
  title = "Summary Statistics of Weather Variables (Paris, 2010-2021)",
  digits = 2,
  summary.stat = c("min", "p25", "median", "mean", "p75", "max", "sd"),
  font.size = "small",
  label = "tab:summary_weather",
  float.env = "table",
  header = FALSE,
  table.placement = "H"
)
cat("\\end{center}")


## ----echo=FALSE, warning=FALSE, message=FALSE, results='asis'-----------------
correlations_df <- data.frame(
  Variable = c(
    "Tmax", "perceived_max", "mean_temp", "perceived_mean", "perceived_min", "tmin",
    "reference_evapotranspiration", "daylight_duration", "shortwave_radiation",
    "sunshine_duration", "dominant_wind_direction", "rain_sum", "precipitation_sum",
    "max_wind_gusts", "snowfall_sum", "precipitation_hours", "max_wind_speed"
  ),
  Correlation = c(
    0.93, 0.93, 0.92, 0.92, 0.86, 0.84, 0.82, 0.76, 0.72, 0.58,
    -0.01, -0.07, -0.09, -0.17, -0.17, -0.18, -0.23
  )
)
stargazer(
  correlations_df,
  summary = FALSE,
  title = "Correlation of Independent Variables with Next-day Maximum Temperature (Tmax)",
  label = "tab:correlation_tmax",
  digits = 2,
  font.size = "small",
  float.env = "table",
  type = "latex",
  header = FALSE,
  table.placement = "H",
  align = FALSE
)


## ----echo=FALSE, warning=FALSE, message=FALSE, results='asis'-----------------
ggplot(df, aes(x = Date, y = y)) +
  geom_line(color = "steelblue") +
  geom_vline(xintercept = max(train$Date),
             linetype = "dashed", color = "red") +
  labs(x = "Date", y = "Next-day Tmax") +
  theme_minimal()


## ----echo=FALSE---------------------------------------------------------------
screeplot(res, main = "Scree Plot - PCA on Training Data")


## ----echo=FALSE, message=FALSE, warning=FALSE---------------------------------
invisible(capture.output(
  paran(X_train, graph = TRUE)
))


## ----echo=FALSE---------------------------------------------------------------
fviz_pca_biplot(res,
                geom.ind = "point", pointshape = 16, pointsize = 0.7, alpha.ind = 0.15,
                col.ind = "grey70", col.var = "firebrick", repel = TRUE,
                select.var = list(contrib = 12)  # label top 12 vars
)



## ----echo=FALSE, message=FALSE, warning=FALSE, results='asis'-----------------
# Prepare the PCA loadings as a proper data frame
loadings_df <- as.data.frame(round(res$loadings[, 1:4], 2))
loadings_mat <- as.matrix(loadings_df)
rownames(loadings_mat) <- rownames(loadings_df)

# Output a clean LaTeX table
stargazer(
  loadings_mat,
  summary = FALSE,
  title = "Principal component loadings for the first four components.",
  digits = 2,
  label = "tab:loadings",
  font.size = "small",
  header = FALSE,
  float.env = "table",
  type = "latex"
)
```

```r
## ----echo=FALSE, message=FALSE, warning=FALSE, results='asis'-----------------
# Prepare top 10 variables best explained by PC1
explained_by_pc1_df <- data.frame(
  Variable = names(head(explained_by_pc1, 10)),
  Explained_Variance = round(head(explained_by_pc1, 10), 3),
  stringsAsFactors = FALSE
)
rownames(explained_by_pc1_df) <- NULL

# Generate clean LaTeX table
stargazer(
  explained_by_pc1_df,
  summary = FALSE,
  title = "Top 10 Variables Best Explained by the First Principal Component (PC1)",
  label = "tab:pc1_explained",
  digits = 3,
  font.size = "small",
  float.env = "table",
  type = "latex",
  header = FALSE,
  table.placement = "!h",
  align = FALSE
)



## ----echo=FALSE, message=FALSE, warning=FALSE, results='asis'-----------------

# Round results for readability
pcr_sensitivity_df <- data.frame(
  Components = pcr_sensitivity$Components,
  RMSE = round(pcr_sensitivity$RMSE, 3),
  MAE = round(pcr_sensitivity$MAE, 3),
  R2 = round(pcr_sensitivity$R2, 3),
  stringsAsFactors = FALSE
)

# Ensure no row names
rownames(pcr_sensitivity_df) <- NULL

# Generate clean LaTeX table
stargazer(
  pcr_sensitivity_df,
  summary = FALSE,
  title = "Sensitivity analysis of PCR model performance for different numbers of principal components.
  label = "tab:pcr_sensitivity",
  digits = 3,
  font.size = "small",
  float.env = "table",
  type = "latex",
  header = FALSE,
  table.placement = "!h",
  align = FALSE
)




## ----echo=FALSE, results='asis'-----------------------------------------------
# Extract all R code chunks from the current Rmd file
code_file <- knitr::purl("Assignment2- Group2-2.Rmd", quiet = TRUE)

# Print a header for the appendix
cat("## Appendix 2. All R Code\n\n")

# Read and display the extracted R code
cat("```r\n")
cat(readLines(code_file), sep = "\n")
cat("\n```")
```