Midterm 2

Jubilee Lee

11/21/2023

Contents

Question 1 2
Question 2 4
Question 3 10
Citation 14
Appendix 15

Question 1

(1) Manual calculation of $\hat{\beta}$ vector is done by computing: $(X^TX)^{-1}X^TY$. However, we confront an issue where $(X^TX)^{-1}$ component throws an error as it is computationally singular. In other words, $det((X^TX)) = 0$ where a determinant of a matrix reflects how the linear transformation associated with the matrix can scale or reflect objects. In our case, the matrix doesn't reflect such association and therefore needs transformation.

Table 1: Daily Variable: contains first 4 values and variance of each daily variable

	u10	v10	sp	tp	lcc	mcc	hcc	cdir	r
Variance	1.67	1.69	381.23	0.00	0.23	0.19	0.27	81829.91	12.91
1	-0.81	1.72	98959.80	0.00	0.01	0.21	0.10	1035524.85	49.18
2	-2.20	-0.21	98978.50	0.00	0.00	0.12	0.23	1075015.22	38.56
3	-1.55	-2.01	98816.11	0.00	0.00	0.09	0.56	1060558.07	42.32
4	-3.39	-0.72	98907.32	0.00	0.01	0.01	0.38	1128662.60	35.19

Furthermore, notice from `Table 1`, the variance between daily variables differ significantly meaning that for some variables, increment of a certain degree doesn't have a huge impact while for some variables, it has a huge impact.

(2) We choose to fix such issue by dividing (i.e., scale) each covariate by it's own standard deviation.

Notice we are able to run the code block below with no error and obtain stable OLS estimates in vector form.

```
df <- X_data
# apply transformation
df_new<-apply(df,2, function(x) x/sd(x))
df_new <- as.data.frame(df_new)

X=cbind(rep(1,n),as.matrix(df_new)) # create X matrix
XX=t(X) %*% X                       # calculate X'X
XY=t(X) %*% Y                       # calculate X'Y
XXI=solve(XX)                       # calculate inverse of (X'X)
b=XXI %*% XY                        # obtain b vector
```

Table 2: Updated Daily Variables and Variance

	u10	v10	sp	tp	lcc	mcc	hcc	cdir	r
Variance	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1	-0.48	1.02	259.58	0.07	0.06	1.12	0.35	12.65	3.81
2	-1.31	-0.13	259.63	0.00	0.00	0.63	0.85	13.14	2.99
3	-0.93	-1.19	259.20	0.21	0.00	0.49	2.09	12.96	3.28
4	-2.02	-0.43	259.44	0.00	0.02	0.07	1.42	13.79	2.73

(3) We seek to derive a relationship between the OLS estimate in the case of regression with unscaled and scaled covariates.

Let $\tilde{X}$ the scaled X matrix dividied by each of the predictor's variance. i.e.) $\tilde{X} = X\Sigma$ where:

$$X = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_1^{-1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_p \end{pmatrix} \tag{1}$$

Note that the *ones* in matrix X is for the intercept portion which is why we divide by one rather than it's standard deviatiton 0.

Recall we were able to manually calculate $\hat{\beta}$ by scaling X and calculating the $(X^TX)^{-1}$ portion. In other words:

$$\begin{aligned}(X^TX)^{-1} &\Rightarrow (\tilde{X^T}\tilde{X})^{-1}\\ &\Rightarrow ((X\Sigma)^TX\Sigma)^{-1}\\ &\Rightarrow (\Sigma^TX^TX\Sigma)^{-1}\\ &\Rightarrow (\Sigma X^TX\Sigma)^{-1}\\ &\Rightarrow \Sigma(X^TX)^{-1}\Sigma\end{aligned}$$

Remark: $\hat{\beta} = (X^TX)^{-1}X^TY$.

Applying this remark:

$$\begin{aligned}\hat{\beta}_{scaled} &= (\tilde{X^T}\tilde{X})^{-1}\tilde{X}^TY\\ \Leftrightarrow \hat{\beta}_{scaled} &= ((X\Sigma)^T(X\Sigma))^{-1}(X\Sigma)^TY\\ \Leftrightarrow \hat{\beta}_{scaled} &= \Sigma^{-1}(X^TX)^{-1}(\Sigma^T)^{-1}\Sigma^TX^TY\\ \Leftrightarrow \hat{\beta}_{scaled} &= \Sigma^{-1}(X^TX)^{-1}X^TY\\ \Leftrightarrow \hat{\beta}_{scaled} &= \Sigma^{-1}\hat{\beta}\\ \Leftrightarrow \hat{\beta}_{scaled} &= \Sigma\hat{\beta}\end{aligned}$$

Hence, we apply relationship $\Sigma(X^TX)^{-1}\Sigma$ in the process of calculating $\hat{\beta}_{scaled} = \Sigma\hat{\beta}$ on the second task of this report.

Question 2

(1) The given data is a *reanalysis*, which is an integration of real observations with computer simulations. More specifically, it is yielded from the ECMWF Reanalysis version 5 (ERA5), the fifth generation ECMWF reanalysis. ERA5 covers from January 1940 to present [2]. Utilizing laws of physics and observations throughout the globe, ERA5 combines model data and yields a concise dataset[3]. In addition, as it is not obligated to report timely forecasts, it consumes the thoroughly collected observations for a good amount of time so that is produces improved quality results of original raw observations[3]. Here, we investigate a possible linear relationship between daily surface air temperature and all the daily variables given from ERA5, using linear regression where we define a linear model as following:

$$Y_i = \beta_0 + \sum_{k=1}^{p} \beta_k X_i, k + \varepsilon_i, i = 1, \ldots, 9 \tag{2}$$

Table 3: Parameters of Model

β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9
intercept	u10	v10	sp	tp	lcc	mcc	hcc	cdir	r

Before proceeding to the data analysis, we plot a couple scatter plots and observe the corresponding correlation values to get an idea of potential issues. For readability, we choose 4 daily variables, `v10, tp, lcc, and r` among 9 variables.

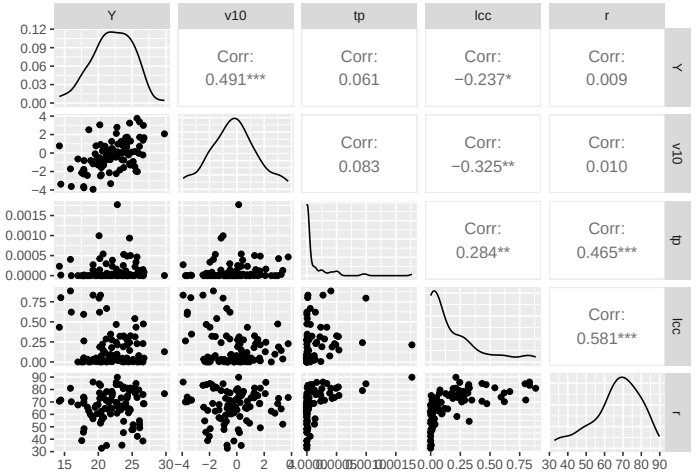

Figure 1: Scatter Plot of Daily Surface Air Temperature vs v10, tp, lcc, r

Notice from the scatter plots, it is hard to find a linear relationship between daily surface air temperature and the other daily variables. The correlation values less than 0.5 supports such intuition. Meanwhile, another potential issue is the dependency daily variables share. Notice that the correlation values among daily variables are more higher. Specifically, `r; relative humidity` has higher correlation between every displayed daily variables rather than the daily surface air temperature. In addition, there are limitations in the daily variable as there might be other factors that could be more crucial to have an impact to the surface air temperature with a linear relationship which are excluded. Furthermore, within the data itself, minor simulation errors could be involved within the dataset. Still, we proceed with the analysis as some variables are likely to having a linear pattern.

(2) Our goal is to estimate: $\beta_i, i = 0, \ldots, n$ where the results are displayed at `Table 4`.

Table 4: Point Estimates of the Model

β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9
-212.53	-0.10	0.76	0.00	296.54	0.00	-1.63	2.54	0.00	0.03

Interpretation for β_0: This is the average daily surface air temperature ($^\circ C$) when all the daily variables X_i have a value of zero at the same time which is -212.53$^\circ C$. By the extreme temperature itself, it is hard to conclude this is a valid result as we are analyzing the surface air temperature on Earth. In fact, this is a temperature close to Neptune's mean temperature [1]. In addition, the setting of having the entire daily variables X_i held to be zero is odd and unnatural. Hence, attempt of interpretation seems unnecessary and we continue to observe other point estimators.

Interpretation for β_i where i = 1, . . . , 9: β_i is the average change of daily surface air temperature ($^\circ C$) per unit change of X_i while all the other $\beta_k, k \neq i$ is held constant. For instance, for β_1, the daily surface air temperature changes on average -0.10 ($^\circ C$) as u10 (wind speed east to west) increases in one m/s as the other daily variables are kept constant. Again, it is highly unnatural to say that other daily variables do not change while one is constantly changing. For instance, the statement: *while the wind speed from east to west changes, the wind speed from south to north shall never change* is not a reliable statement by experience. Therefore, the interpretation is quite odd in the physical world but to investigate the relationship between variables and the daily surface air temperature, we accept such controls and proceed on our analysis.

(3) We wish to obtain fitted values, the predicted values of the daily surface air temperature using the data for our model, and residuals, the difference between observed daily surface air temperature and the predicted values.

Table 5: Results of Calculated Fitted and Residual Values

Fitted	23.61	22.62	21.79	22.79	22.71	21.50	19.78	19.12	19.75	20.60	...
Residuals	2.50	3.97	3.97	0.69	-2.23	-1.06	-1.98	-1.57	-1.87	0.61	...

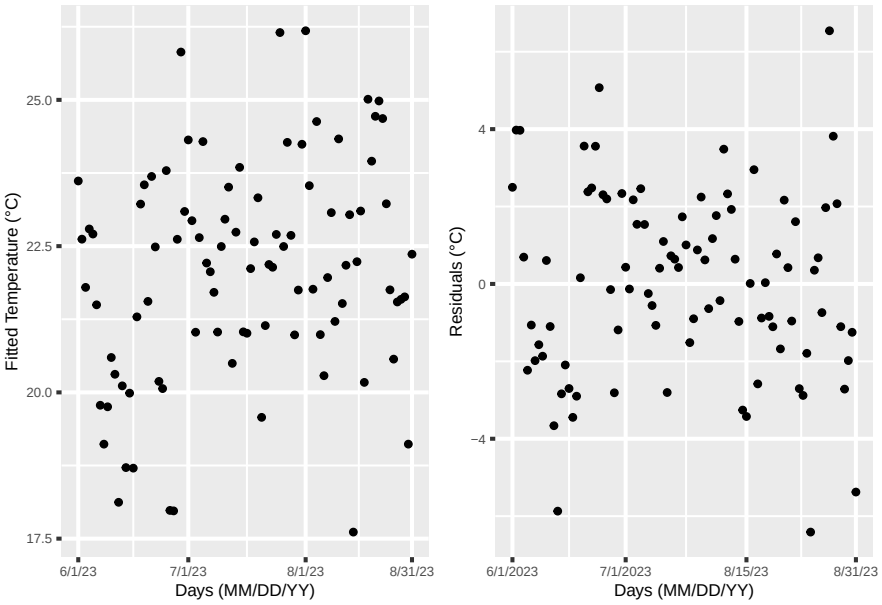

Figure 2: Plot of Fitted and Residual vs Days

Notice there isn't any clear pattern from both plots at `Figure 2` so it is fair to say that the data points are randomly scattered.

(4) Now, we care about the measure of association for our model which we initially investigate with R^2. R^2 is the proportionate reduction of total variation in the daily surface air temperature associated with the use of all the other daily variables which in our case is 0.38. This indicates that around 38% of variability is explained by the model.

Table 6: Results of Calculated Fitted and Residual Values

	R^2	R^2_{adj}	MSE	$Fstatistics$
Results	0.38	0.31	6.37	5.48

Taking account that this is a simulated data based on observation of daily variables on Earth and that there are a good amount of uncertainty within the data, the R^2 value is quite decent. However, the drawback of this measure is that as the number of predictors increase, the R^2 value will never decrease even if the predictors add meaningless or repetitive information. Hence, in order to prevent a model having multiple unnecessary predictors, we penalize whenever useless predictors are detected which is R^2_{adj}. Interestingly, the difference between R^2_{adj} and R^2 is 0.01 which is not drastically large considering the number of predictors (=9) we use in which some of the predictors seemed to share similar aspects such as `wind speed east to west` and `wind speed south to north`. However, again, the ERA5 is updated continouously and release data with no significant flaw detected[3], so it is possible that the daily variables may give in more information than we think they do. In addition, our MSE, which is a measure of the quality of an estimator, is 6.37. Considering a model with no error will produce a zero MSE, 6.37 is a fairly fine result especially considering that our model is attemping to predict the surface temeperature of Earth with less than 100 observations. Furthermore, since the p-value is small, we compare the F-statistics which is 5.48. This tells us that at least one of our predictor is useful which surely support our observation above.

	(Intercept)	u10	v10	sp	tp	lcc	mcc	hcc	cdir	r
(Intercept)	1.13e+04	-1.30e+01	-7.88e-01	-1.13e-01	-1.40e+04	-5.18e+01	-6.06e+01	-1.96e+01	-1.70e-04	6.60e-02
u10	-1.30e+01	5.35e-02	-3.79e-03	1.34e-04	5.04e+01	5.54e-02	7.72e-02	8.69e-02	-1.00e-07	-3.82e-03
v10	-7.88e-01	-3.79e-03	3.24e-02	6.80e-06	-8.12e+00	1.13e-01	-1.94e-03	-4.95e-02	1.00e-07	-8.66e-05
sp	-1.13e-01	1.34e-04	6.80e-06	1.10e-06	1.59e-01	5.27e-04	6.06e-04	2.01e-04	0.00e+00	-2.40e-06
tp	-1.40e+04	5.04e+01	-8.12e+00	1.59e-01	1.84e+06	1.00e+02	-6.83e+02	-3.40e+02	-5.92e-04	-1.66e+01
lcc	-5.18e+01	5.54e-02	1.13e-01	5.27e-04	1.00e+02	2.80e+00	-3.84e-01	2.58e-01	9.00e-07	-2.45e-02
mcc	-6.06e+01	7.72e-02	-1.94e-03	6.06e-04	-6.83e+02	-3.84e-01	3.26e+00	-3.34e-01	4.00e-07	6.29e-05
hcc	-1.96e+01	8.69e-02	-4.95e-02	2.01e-04	-3.40e+02	2.58e-01	-3.34e-01	1.51e+00	0.00e+00	-9.22e-03
cdir	-1.70e-04	-1.00e-07	1.00e-07	0.00e+00	-5.92e-04	9.00e-07	4.00e-07	0.00e+00	0.00e+00	1.00e-07
r	6.60e-02	-3.82e-03	-8.66e-05	-2.40e-06	-1.66e+01	-2.45e-02	6.29e-05	-9.22e-03	1.00e-07	1.38e-03

Figure 3: Covariance Matrix

Check `Figure 3` for the covariance matrix and `Figure 4` for the correlation matrix. From both matrices, among the daily variables, notice that the absolute correlation values are less than or around 0.5. However, the daily variables didn't have correlation values much higher than these values which is concerning.

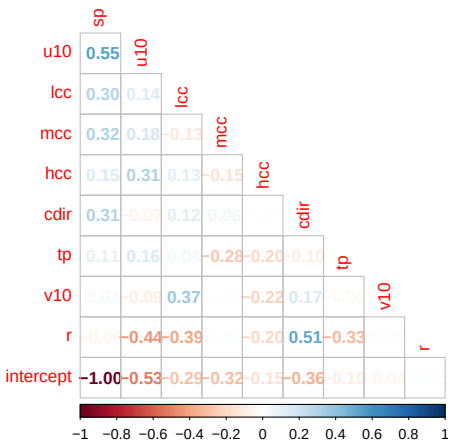

Figure 4: Correlation Matrix Values

(5) We obtain the p-values for testing the given hypothesis which is the probability of observing data more extreme than the observed when H_0 is true. Typically, we safely reject the null hypothesis if the p-values are close to zero. Based on our p-values displayed at `Table 7`, it is likely to reject null hypothesis associated with variable `v10` as it is close to zero.

Table 7: P-Value for Testing H_0: Beta_i = 0

intercept	u10	v10	sp	tp	lcc	mcc	hcc	cdir	r
0.05	0.67	0.00	0.03	0.83	1.00	0.37	0.04	0.56	0.44

However, the rest of the p-values are higher than zero meaning it's risky to reject the associated H_0 as there is some probability for it to be true. In summary, if we only reject the null hypothesis for predictor v10, our model is now $Y = \beta_2 * X$ which means

the daily surface air temperature has a linear relationship with only wind speed south to north. This could be a valid as the other daily variables could have a non-linear relationship. In addition, for the null hypothesis $\beta_0 = 0$, i.e. the intercept, could be plausible as it means that when all the other daily variables are zero, the average surface temperature is zero. In order to identify if all these statements are acceptable, further analysis beyond linear regression and data research is recommended.

Table 8: 95% CI for Daily Surface Air Temperature (°C)

	Lower	Upper	Lower (Bonferroni)	Upper (Bonferroni)
intercept	-424.01	-1.05	-519.22	94.16
u10	-0.56	0.36	-1.28	0.95
v10	0.40	1.12	0.40	2.15
sp	0.00	0.00	-0.28	2.06
tp	-2401.89	2994.97	-0.94	1.09
lcc	-3.33	3.33	-1.11	1.11
mcc	-5.22	1.96	-1.30	0.68
hcc	0.10	4.98	-0.27	1.64
cdir	0.00	0.00	-0.84	1.27
r	-0.05	0.10	-1.01	1.76

From the 95% confidence interval (CI) displayed in `Table 8`, notice that even between Bonferroni and the manually calculated CI differs whether the bound contains zero for some variables. Statistically speaking, with 95% confidence, the calculated bounds contain the true β_is with 95% confidence. Notice, that only `v10` has both bounds not including zero. In other words, wind speed south to north (m/s) is likely to have a **positive** linear relationship with the surface temperature (C) with 95% confidence. This makes sense as by experience, assuming the distance isn't drastically apart (from top and bottom of Earth), southern parts are warmer than northern parts so if we get a warm breeze go up north, it is likely the surface temperture is going to increase.

Table 9: 95% CI for $\beta_0 + 3\beta_1$

Lower	Upper
-423.57	-2.07

For practice of linear algebra, we calculate the 95% CI for $\beta_0 + 3\beta_1$ which is displayed at `Table 9`.

(6) Now we investigate the 95% CI and prediction interval (PI) for our daily surface air temperature given the daily variables values in variable `X_h_pred`. Note that PI is wider than CI since PI has more uncertainty so therefore has a larger variance.

Table 10: 95% CI and PI for Observations in X_h_pred

	Confidence Interval	Prediction Interval
xh_1	(20.87, 24.94)	(17.49, 28.32)
xh_2	(-140.48, 201.77)	(-140.55, 201.84)

In our context, the PI is the prediction interval of the daily surface air temperature ($^\circ C$), i.e. Y_*, for a given observation values of the other daily variables, whereas the confidence interval is the estimation of *average* daily surface air temperature, i.e. $E(Y_*)$. The goal of the PI is to predict for a single outcome which includes more variability than solving for the average temperature estimation. Notice from `Table 10` both CI and PI bounds for the first observation are much smaller than the bounds of the second observation. For the first observation, the predicted surface temperature is contained within a reasonable bound with 95% confidence as a temperature 15 ~ 30 ($^\circ C$) is a typical temperature of South Bend. In addition, since the bound is tight, we can say there isn't much uncertainty within the CI and PI. On the other hand, notice that the second observation has a unrealistic bound for the surface temperature saying it is likely the predicted surface temperature is between -140 ~ over 200 ($^\circ C$) with 95% confidence. This information is more than useless as it is obvious by intuition that the surface temperature of Earth is going to be within that range. The temperature range is between the mean temperature of Mercury and Saturn [1] which is too wide to make use of.

Table 11: X_h_pred

	u10	v10	sp	tp	lcc	mcc	hcc	cdir	r
xh_1	1.01	1.34	98587.00	0.00	0.17	0.64	0.33	1050000.00	72.00
xh_2	-0.45	7.01	99238.00	0.00	50.97	0.14	0.70	1140000.00	57.00

We suspect that it is due to the `lcc` value where the range should be an absolute number from 0 to 1, measuring the fraction of clouds at surface pressure to 400 millibars (mb). However, from the second observation, it is over 50 (displayed in `Table 11`) which is clearly an error. Therefore, analysis should not be done on the second observation unless the `lcc` value is taken care of.

(7) Now, we investigate which method to use to provide the best predictions for the temperatures in July 2023.

Table 12: Simultaneous CI for July 2023 Prediction

	Bonferroni	Working-Hotelling	Sheffe
Interval	3.26	4.41	7.02

Among Bonferroni, Working-Hotelling or Sheffe, Bonferroni has the smallest bound range, which are displayed in `Table 12`, imply having less uncertainty compared to the other two methods. Hence, we choose Bonferroni.

(8) From the plots at `Figure 5`, notice that there is a strong linear relationship between fitted values and residuals. Hence,nonlinearity and non-constant variance is detected which violates our model assumption.

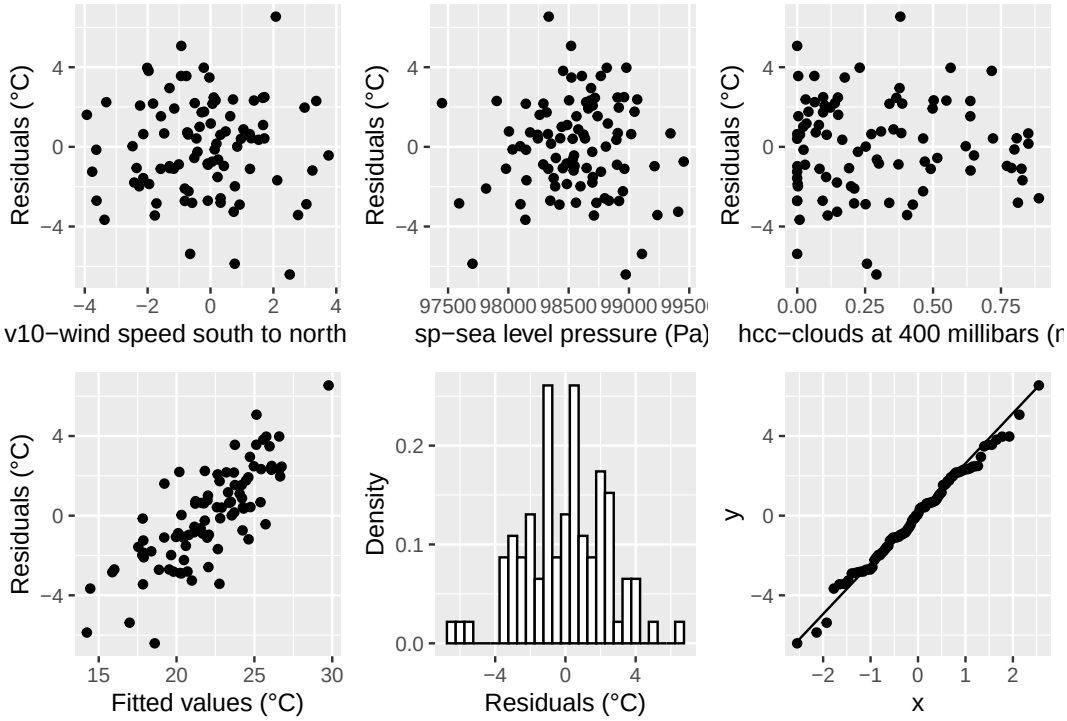

Figure 5: Plot of Fitted and Residual

Note that the histogram does not display a *perfect* bell shape, by the `x vs y` plot on the right-bottom of `Figure 5`, we can observe that the data points are quite tightly aligned to the straight line. Hence, normality isn't concerning. In addition, it is difficult to conclude independence in residuals as it is natural that these environmental daily variables are correlated and dependent to one another in some degree.

To resolve such potential issues, we transform the data using box-cox and interestingly, notice from `Figure 6`, we get $\lambda = 1.34$. By rounding to the closest integer 1, we get a model that isn't different from our original model.

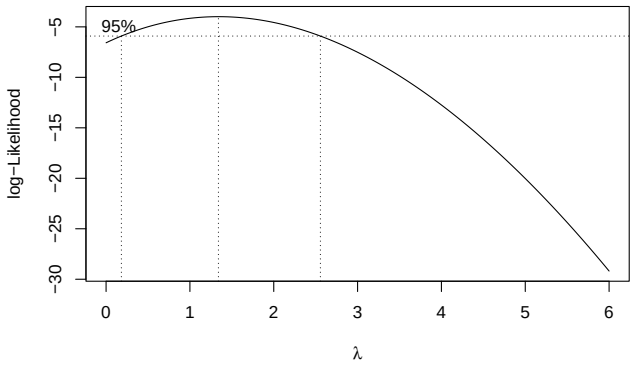

Figure 6: box-cox

Again, as discussed above, we conclude that such issues should be dealt with more complex models.

(9) In this section, we apply a Bayesian version of the regression using JAGS as introduced in class. We attempt to show as the inverse variance increase, the posterior mean shrink toward zero.

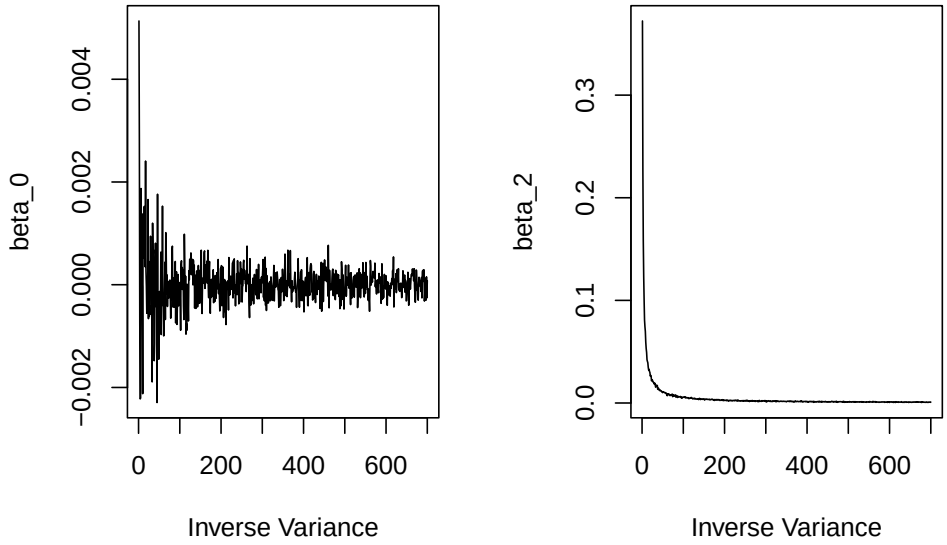

Figure 7: Posterior Mean Change due to Inverse Variance

Notice that as the inverse variance for the priors of the elements increase, the posterior mean shrink toward zero from `Figure 7`. Due to the page limit, we plot for two predictor variables, the `intercept` and `v10` but note that this holds for all the other predictor variables as well. This is expected as we are increasing the precision as we increase the inverse variance for the priors so therefore, the posterior shrink to zero since the prior is zero. In other words, the posterior becomes the prior as precision for priors increase.

(10) We did our research to figure out the surface temperature using information such as where wind blows from, how much it rains or how humid is it, from June 1st to August 31th 2023 in South Bend. Accordingly, we were able to figure out an approximate surface temperature using such few information. Some thing we need to keep in mind is that it is impossible to figure out the exact surface temperature. For example, there could have been unavoidable mistakes when the machine was getting the information. Or, there could have been obstacles like wind blowing too strong the machine couldn't figure out if it was from south to north or east to west. Even when all of these troubles could have been in our way, we were able to answer around what the surface air temperature will be using such few natural information. One important thing we found was that wind blowing from South to North make the surface air temperature warmer.

Question 3

(1)

$$\begin{aligned}\tilde{Q}(\beta) &= \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{p}\beta_k x_{ik})^2 + \lambda\sum_{k=0}^{p}\beta_k^2 \\ &\Rightarrow (Y - X\beta)'(Y - X\beta) + \lambda\beta'\beta \\ &\Leftrightarrow Y'Y - Y'X\beta - (Y'X\beta)' + (X\beta)'X\beta + \lambda\beta'\beta \\ \frac{\partial\tilde{Q}}{\partial\beta} &= -Y'X + \beta'X'X + \lambda\beta'\end{aligned}$$

where $I \in \mathbb{R}^{(n+1)\times(n+1)}$ is the identity matrix.

Using least squares method:

$$\begin{aligned}\frac{\partial\tilde{Q}}{\partial\beta} &= -Y'X + \beta'X'X + \lambda\beta' = 0 \\ &\Rightarrow (-Y'X + \beta'X'X + \lambda\beta')' = 0 \\ &\Leftrightarrow -X'Y + X'X\tilde{\beta} + \lambda\tilde{\beta} = 0 \\ &\Leftrightarrow (X'X + I\lambda)\tilde{\beta} = X'Y \\ &\Leftrightarrow \tilde{\beta} = (X'X + I\lambda)^{-1}X'Y\end{aligned}$$

Hence, we found the closed form expression for vector $\tilde{\beta}$.

$$\tilde{\beta} = (X'X + I\lambda)^{-1}X'Y \tag{3}$$

(2) We first wish to prove $\tilde{\beta}$ is Gaussian. Notice:

$$\begin{aligned}Y = X\beta + \varepsilon, \varepsilon &\sim \mathsf{Norm}(0, \sigma^2 I) \\ &\Rightarrow Y \sim \mathsf{Norm}(X\beta, \sigma^2 I)\end{aligned}$$

Accordingly:

$$\begin{aligned}\tilde{\beta} &= (X'X + I\lambda)^{-1}X'Y \\ &\Leftrightarrow \tilde{\beta} = (X'X + I\lambda)^{-1}X'(X\beta + \varepsilon) \\ &\Leftrightarrow \tilde{\beta} = (X'X + I\lambda)^{-1}X'X\beta + (X'X + I\lambda)^{-1}X'\varepsilon \\ &\Leftrightarrow \tilde{\beta} = (X'X + I\lambda)^{-1}X'X\beta + ((X'X + I\lambda)^{-1}X')\varepsilon\end{aligned}$$

Hence, $\tilde{\beta}$ is Gaussian.

Next, we calculate the mean and variance of $\tilde{\beta}$. First, the mean:

$$\begin{aligned}&E(\tilde{\beta}) \\ &\Leftrightarrow E((X'X + I\lambda)^{-1}X'Y) \\ &\Leftrightarrow (X'X + I\lambda)^{-1}X'E(Y) \\ &\Leftrightarrow (X'X + I\lambda)^{-1}X'E(X\beta + \varepsilon) \\ &\Leftrightarrow (X'X + I\lambda)^{-1}X'E(X\beta) + (X'X + I\lambda)^{-1}X'E(\varepsilon) \\ &\Leftrightarrow (X'X + I\lambda)^{-1}X'XE(\beta) + 0 \\ &\Leftrightarrow (X'X + I\lambda)^{-1}X'X\beta \\ &\neq \beta\end{aligned}$$

Now, the variance:

$$\begin{aligned} & Var(\tilde{\beta}) \\ \Leftrightarrow\ & Var((X'X + I\lambda)^{-1}X'Y) \\ \Leftrightarrow\ & (X'X + I\lambda)^{-1}X'Var(Y)((X'X + I\lambda)^{-1}X')' \\ \Leftrightarrow\ & (X'X + I\lambda)^{-1}X'Var(X\beta + \varepsilon)((X'X + I\lambda)^{-1}X')' \\ \Leftrightarrow\ & (X'X + I\lambda)^{-1}X'Var(\varepsilon)((X'X + I\lambda)^{-1}X')' \\ \Leftrightarrow\ & (X'X + I\lambda)^{-1}X'\sigma^2((X'X + I\lambda)^{-1}X')' \\ \Leftrightarrow\ & \sigma^2(X'X + I\lambda)^{-1}X'((X'X + I\lambda)^{-1}X')' \\ \Leftrightarrow\ & \sigma^2(X'X + I\lambda)^{-1}X'X(X'X + I\lambda)^{-1} \end{aligned}$$

In conclusion,
The mean is:

$$E(\tilde{\beta}) = (X'X + I\lambda)^{-1}X'X\beta \tag{4}$$

The variance is:

$$Var(\tilde{\beta}) = \sigma^2(X'X + I\lambda)^{-1}X'X(X'X + I\lambda)^{-1} \tag{5}$$

(3) Hence, since $E(\tilde{\beta}) \neq \beta$, it is a biased estimator. Despite this fact, it is useful when we have to manually calculate $(X'X)^{-1}$ to calculate the vector β as we did throughout the first and second task. Instead $(X'X)^{-1}$, now we have $(X'X + I\lambda)^{-1}$ so we have control over the inverse portion to ensure singularity issue does not arise during computation.

(4) Let β_{post} be the posterior estimate. Then,

$$E(\beta_{post}) = (\Sigma_0^{-1} + X'X/\sigma^2)^{-1}(\Sigma_0^{-1}\beta_0 + X'X/\sigma^2) \tag{6}$$

From equation (6), we let $\beta_0 = 0$ and $\Sigma_0^{-1} = I\lambda/\sigma^2$.

Then for the mean, we get:

$$\begin{aligned} & E(\beta_{post}) \\ = & (\Sigma_0^{-1} + X'X/\sigma^2)^{-1}(\Sigma_0^{-1}\beta_0 + X'X/\sigma^2) \\ = & (\Sigma_0^{-1} + X'X/\sigma^2)^{-1}(X'X/\sigma^2) \\ = & (I\lambda/\sigma^2 + X'X/\sigma^2)^{-1}(X'X/\sigma^2) \\ = & (\sigma^2)(I\lambda + X'X)^{-1}(X'X)(\sigma^2)^{-1} \\ = & (\sigma^2)(\sigma^2)^{-1}(I\lambda + X'X)^{-1}(X'X) \\ = & (I\lambda + X'X)^{-1}(X'X) \\ = & \tilde{\beta} \end{aligned}$$

$$E(\beta_{post}) = \tilde{\beta} \tag{7}$$

Since equation (7) holds, we proved that $\tilde{\beta}$ has the same expression of the posterior mean for this model.

Now for the variance, notice:

$$\begin{aligned} & Var(\beta_{post}) \\ = & (\Sigma_0^{-1} + X'X/\sigma^2)^{-1} \end{aligned}$$

However, notice from equation (5), the variance of $\tilde{\beta}$ was $\sigma^2(X'X + I\lambda)^{-1}X'X(X'X + I\lambda)^{-1}$ so therefore, $Var(\tilde{\beta}) \neq Var(\beta_{post})$.

(5) Here, we implement the given model in JAGS.

To observe and have a general idea of the model behavior , we set up a simple linear regression model:

$$Y = \beta_0 + \beta_1 X + \varepsilon, \varepsilon \sim Norm(0, \sigma^2) \tag{8}$$

$$\beta_0 = 5, \beta_1 = 3, \sigma^2 = 1 \tag{9}$$

First, we define the model and assign it to `model_string`.

```
model_string <- "model{

# Likelihood Definition
for(i in 1:n){
  Y[i]   ~ dnorm(mu[i],inv.var)
  mu[i] <- beta[1]+beta[2]*X[i]
}

# Prior Definition for Beta0 and Beta1
for(j in 1:2){
  beta[j] ~ ddexp(0,b)
}

# Prior for 1/sigma^2
inv.var   ~ dgamma(0.01, 0.01)
sigma     <- 1/sqrt(inv.var)

}"
```

Notice that each component of β is having a Laplace prior where `ddexp(0,b)` is the built-in R function that generates the double exponential density of $\mu = 0$ and the scale being b.

Now, we set up the variables to be used within the model.

```
set.seed(1)                           # set seed
n = 100                               # number of data
X<-seq(0, 1, length = n)              # generate sequence of numbers for X
beta0=5                               # define beta0
beta1=3                               # define beta1
sigma=1                               # define sigma
Y=matrix(NaN,nrow=n)                  # define Y matrix to store values for Y
epsilon = rnorm(n, mean=0, sd=sigma)  # define epsilon
Y = beta0 + beta1*X + epsilon         # calculate Y and store result
num_it <- 100                         # number of iteration throughout the model
N<-length(Y)                          # data size
coeff=matrix(NaN,nrow=2,ncol=num_it)  # define matrix to store result of
                                      # posterior mean of each beta
```

Now, we observe how the posterior mean of each β_i change as the scale b changes.

```
for (i in 1:num_it){
  model <- jags.model(textConnection(model_string),
                      # increase b as iteration goes on
                      data = list(Y=Y,n=N,X=X,b=(i*100)),
                      quiet=TRUE)
  update(model, 1000, progress.bar="none");
  samp <- coda.samples(model,
                       variable.names=c("beta","sigma"),
                       n.iter=1000, progress.bar="none")
  # store results to matrix coeff
  coeff[1,i]=summary(samp)$statistics[,1][1] # result of beta0
```

```
  coeff[2,i]=summary(samp)$statistics[,1][2] # result of beta1
}
```

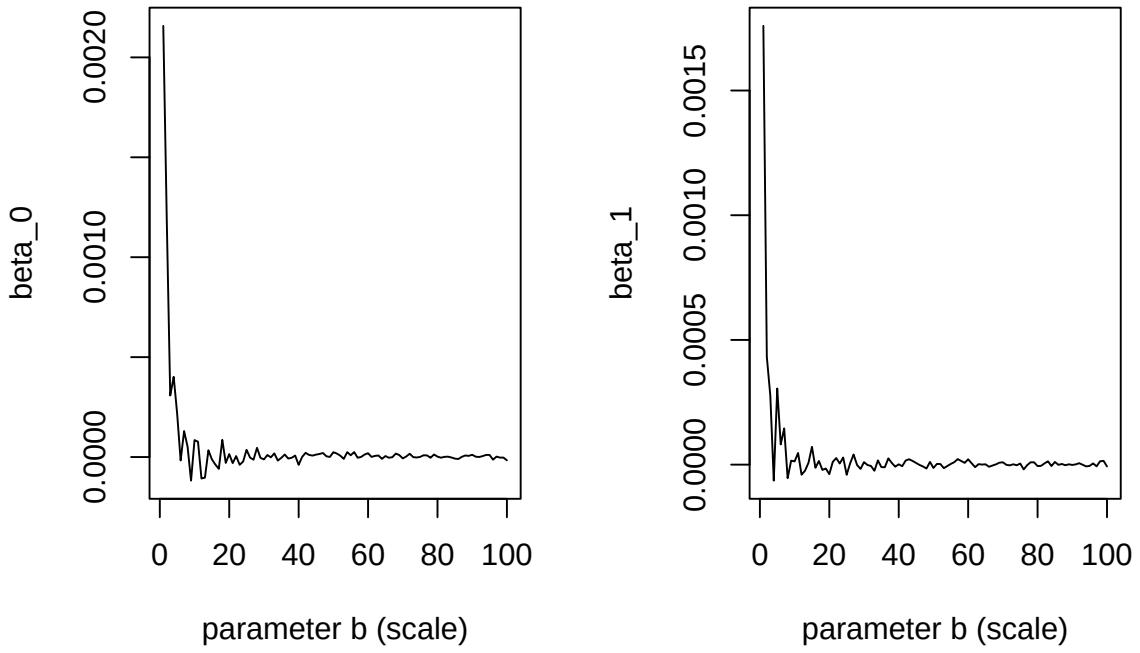

Figure 8: Change in $beta_i$ due to Change of Scale b

Notice that as the scale b increases throughout the iteration, both posterior mean of each β_i both shrink to zero. This is expected as when b increases:

$$\lim_{b\to\infty} f(\beta) = \lim_{b\to\infty} \frac{1}{2b} exp(\frac{|\beta|}{b}) \to 0$$

Hence,

$$f(\beta|Y) \propto f(Y|\beta)f(\beta) \to 0$$

Therefore, a Bayesian model with iid priors for each component of β, and with each component having a Laplace (or double exponential) prior the parameter b controls the shrinkage of the posterior mean of each β_i to 0.

Citation

[1] NASA. (n.d.). Solar system temperatures - NASA science. NASA. https://science.nasa.gov/resource/solar-system-temperatures/

[2] ECMWF reanalysis V5 (ERA5). Drought.gov. (n.d.). https://www.drought.gov/data-maps-tools/ecmwf-reanalysis-v5-era5#:~:text=ERA5%20is%20the%20fifth%20generation,land%20and%20oceanic%20climate%20variables.

[3] Copernicus Climate Data Store. Copernicus Climate Data Store | Copernicus Climate Data Store. (n.d.). https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=overview

Appendix

```
## import libraries and set directory

knitr::opts_chunk$set(echo = TRUE,warning = FALSE, message = FALSE)
library(MASS)
library(tidyverse)
library(GGally)
library(ggpubr)
library(corrplot)
library(ggplot2)
library(knitr)
library(rjags)
library(latex2exp)
cat("\014")
graphics.off()
set.seed(1)
rm(list=ls())
this.dir = dirname(rstudioapi::getActiveDocumentContext()$path)
setwd(this.dir)


################################## Question 1 ##################################
### (1) ###
# load data
data=load("data.Rdata")
# get values
n=length(Y)
X_data
mod.main=lm(Y~.,X_data)
mod.sum=summary(mod.main)

# proceed on computation
X=cbind(rep(1,n),as.matrix(X_data))
XX=t(X) %*% X
XY=t(X) %*% Y
#XXI=solve(XX) # this will throw an error

### (2) ###
df <- X_data
sd_list <- apply(df,2,sd)
df_new<-apply(df,2, function(x) x/sd(x))
df_new <- as.data.frame(df_new)

df <- X_data
# apply transformation
df_new<-apply(df,2, function(x) x/sd(x))
df_new <- as.data.frame(df_new)

# computation for b vector
X=cbind(rep(1,n),as.matrix(df_new)) # create X matrix
XX=t(X) %*% X                       # calculate X'X
XY=t(X) %*% Y                       # calculate X'Y
XXI=solve(XX)                       # calculate inverse of (X'X)
b=XXI %*% XY                        # obtain b vector
balt=solve(XX,XY)

################################################################################
```

```
################################## Question 2 ###################################
### (1) ###
# Plot for data
df <- X_data
df$Y <- Y
scatplot=data.frame(cbind(Y,X_data[,2],X_data[,5],X_data[,9]))
names(scatplot)=c("Y","v10","lcc","r")
ggpairs(scatplot)

### (2) ###
# Calculate for beta vector
# get standard deviation for each daily variable
sigma.x <- array(NaN,10)
sigma.x[1] <- 1
for (i in 1:9){
  sigma.x[i+1] <- sd(X_data[,i])
}
sigma.x = diag(sigma.x)
# matrix computation
Xscale=cbind(rep(1,n),as.matrix(df_new))
XXscale=t(Xscale) %*% Xscale
XYscale=t(Xscale) %*% Y
XXIscale=solve(XXscale)
XXI = solve(sigma.x) %*% XXIscale %*% solve(sigma.x)

bscale = XXIscale %*% XYscale

b = solve(sigma.x) %*% bscale
b

X=cbind(rep(1,n),as.matrix(X_data))
# fitted
Y_hat=X %*% b

I_n=diag(n)
# residual
e=Y-Y_hat

# Display calculated b in table
b_rep = format(round(b, 2), nsmall = 2)
b_rep <- unlist(as.list(b_rep))
df <- data.frame()
df <- rbind(df, b_rep)
kable(df, caption = "Point Estimates of the Model",
      col.names = c('$\\beta_0$', '$\\beta_1$', '$\\beta_2$',
                    '$\\beta_3$', '$\\beta_4$', '$\\beta_5$',
                    '$\\beta_6$', '$\\beta_7$', '$\\beta_8$',
                    '$\\beta_9$'))

### (3) ###
# fitted
fit<-Y_hat[1:10]
# residuals
res<-e[1:10]

y_hat_rep = format(round(Y_hat[1:10], 2), nsmall = 2)
y_hat_rep[11]<-"..."
e_rep = format(round(e[1:10], 2), nsmall = 2)
```

```
e_rep[11]<-"..."
df <- data.frame(rbind(y_hat_rep,e_rep))
rownames(df) <- c('$Fitted$', '$Residuals$')
kable(df, caption = "Results of Calculated Fitted and Residual Values",
      col.names = c())

# Plot for Fitted
df=tibble(T=1:n,Y=Y_hat)

#options(repr.plot.width = 0.001, repr.plot.height =0.005)

fitPlot<-ggplot(df,aes(T, Y))+geom_point()+
  labs(x = "Days (MM/DD/YY)",y="Fitted Temperature (°C)") +
  theme(text = element_text(size = 9),element_line(size =1))
fitPlot <- fitPlot +
  scale_x_continuous(limits = c(1, 92),
                     breaks=c(1,31, 63,92),

                     labels=c("6/1/23", "7/1/23", "8/1/23", "8/31/23"))

# Plot for Residuals
df=tibble(T=1:n,E=e)
resPlot <- ggplot(df,aes(T, E))+geom_point()+
  labs(x = "Days (MM/DD/YY)",y="Residuals (°C)") +
  theme(text = element_text(size = 9),element_line(size =1))
resPlot <- resPlot +
  scale_x_continuous(limits = c(1, 92),
                     breaks=c(1,31, 63,92),
                     labels=c("6/1/2023", "7/1/2023", "8/15/23", "8/31/23"))


figure <- ggarrange(fitPlot, resPlot,ncol = 2, nrow = 1)
figure

### (4) ###
# calculate for Sum of Squares
J_n=matrix(rep(1,n^2),ncol=n)
SST=t(Y)%*%(I_n - 1/n*J_n)%*%Y
SSE=t(Y-X%*%b)%*%(Y-X%*%b)
SSR=SST-SSE

# define p
p=dim(X_data)[2]+1

# compute R^2
R2=1-SSE/SST
# compute R^2 adjusted
R2adj=1-(n-1)/(n-p)*SSE/SST

# see results
c(R2,R2adj)

# calculate MSE
MSE=SSE/(n-p)
s=sqrt(MSE)
# see results
print(s)
```

```r
# use the previous expressions to perform the global F test
MSR=SSR/(p-1)
# get F statistics
Fstat=MSR/MSE
# get p value for global f test
pvalue_globalF=2*(1-pf(abs(Fstat),p-1,n-p))
pvalue_globalF # check result

# Comparing pvalue may not be the best idea since it can be very small
# we can compare the F statistic instead
c(mod.sum$fstatistic[1],Fstat)

# Display results
Results = format(round(c(R2, R2adj, MSE, Fstat), 2), nsmall = 2)
df <- data.frame(rbind(Results))
kable(df, caption = "Results of Calculated Fitted and Residual Values",
      col.names = c('$R^2$', '$R_{adj}^2$', '$MSE$', '$F statistics$'))

# Compute for covariance matrix
# Same for Var($\hat{\beta}$), we compare it with R output.
V_b=MSE[1,1] * XXI
V_b[1:9,1]

#mod=lm(Y ~ ., df_new)
mod=lm(Y ~ ., X_data)
V_b_R=vcov(mod)
V_b_R[1,]
max(abs(V_b-V_b_R))

#Standard deviation of the entries of $\hat{\beta}$
sd_b=sqrt(diag(V_b))

cbind(summary(mod)$coefficients[,2],sd_b)

# Compute for correlation matrix
# correlation matrix to look at the variability at the same scale
DInv = solve(diag(sqrt(diag(V_b))))
C_b=DInv %*% V_b %*% DInv
C_b[1,]

# Report correlation matrix using corrplot
C_b <- as.data.frame(C_b)
rownames(C_b) <- c('intercept',colnames(X_data))

colnames(C_b) <- c('intercept',colnames(X_data))
C_b<-data.matrix(C_b, rownames.force = NA)
corrplot(C_b, method = 'number',order = 'FPC', type = 'lower', diag = FALSE)

### (5) ###
# calculate p value
t_stats=b/sd_b
pvalue_b=2*(1-pt(abs(t_stats),n-p))
p_val = format(round(pvalue_b, 2), nsmall = 2)
df <- data.frame()
df <- rbind(df, unlist(as.list(p_val)))
# report p value
kable(df, caption = "P-Value for Testing H_0: Beta_i = 0",
      col.names = c('intercept',colnames(X_data)))
```

```r
# calculate CI
alpha=0.05
t=qt(1-alpha/2,n-p)
b.CI=cbind(b - t*sd_b,b + t*sd_b)
# get CI from lm function
mod.ci=confint(mod,level=1-alpha)

mapply(c,mod.ci[1,],b.CI[1,])
mapply(c,mod.ci[2,],b.CI[2,])
mapply(c,mod.ci[6,],b.CI[6,])
# do bonferroni
mod.ci.bonf=confint(mod,level=1-alpha/p)

# report CI
cis = format(round(cbind(b.CI,mod.ci.bonf), 2), nsmall = 2)
df <- data.frame(cis)
rownames(df)[1]<-"intercept"
kable(df, caption = "95% CI for Daily Surface Air Temperature (°C)",
      col.names = c("Lower", "Upper",
                    "Lower (Bonferroni)", "Upper (Bonferroni)"))

# CI for beta0+3beta1
comb_b=c(1,3,0,0,0,rep(0,p-5))
b_hat_comb=comb_b%*%b
b_sd_comb=sqrt(t(comb_b)%*%V_b%*%comb_b)
# report CI for beta0+3beta1
b_CI_comb=cbind(b_hat_comb - t*b_sd_comb,b_hat_comb + t*b_sd_comb)
b_CI_comb
la = format(round(b_CI_comb, 2), nsmall = 2)
df <- data.frame(la)
kable(df, caption = "95% CI for $\\beta_0+3\\beta_1$",
      col.names = c("Lower", "Upper"))

### (6) ###
# create the $X_*$ vector to get point estimate
n_pred=dim(X_h_pred)[2]
Xh=as.matrix(cbind(1,X_h_pred))

# get standard deviation for both CI and PI
Y_hat_h=Xh %*% b
s2_yhat_c=diag(MSE[1,1]*Xh%*%XXI%*%t(Xh))
s2_yhat_p=diag(MSE[1,1]*(1+Xh%*%XXI%*%t(Xh)))

# build 95% CI and PI
Yhat_ci=cbind(Y_hat_h,Y_hat_h-t*sqrt(s2_yhat_c),Y_hat_h+t*sqrt(s2_yhat_c))
Yhat_pi=cbind(Y_hat_h,Y_hat_h-t*sqrt(s2_yhat_p),Y_hat_h+t*sqrt(s2_yhat_p))

Yhat_ci = format(round(Yhat_ci, 2), nsmall = 2)
x11<-sprintf("(% s, %s )", Yhat_ci[1,2], Yhat_ci[1,3])
x12<-sprintf("(% s, %s )", Yhat_ci[2,2], Yhat_ci[2,3])

Yhat_pi = format(round(Yhat_pi, 2), nsmall = 2)
x21<-sprintf("(% s, %s )", Yhat_pi[1,2], Yhat_pi[1,3])
x22<-sprintf("(% s, %s )", Yhat_pi[2,2], Yhat_pi[2,3])

xh_1<-c(x11, x21)
xh_2<-c(x12, x22)
# report CI and PI
```

```r
df <- data.frame(rbind(xh_1, xh_2))
kable(df, caption = "95% CI and PI for Observations in X_h_pred",
      col.names = c("Confidence Interval", "Prediction Interval"))


### (7) ###
g=31
alpha=0.05
n=length(X_data[,1])
B=sqrt(qf(1-alpha/g,1,n-p)) # bonferroni
W=sqrt(p*qf(1-alpha,p,n-p)) # working-hotelling
S=sqrt(g*qf(1-alpha,g,n-p)) # sheffe
Interval<-c(B,W,S)

Interval = format(round(Interval,2), nsmall = 2)

df <- data.frame(rbind(Interval))
kable(df, caption = "Simultaneous CI for July 2023 Prediction",
      col.names = c("Bonferroni", "Working-Hotelling", "Sheffe"))

### (8) ###
# plot for residuals
df=tibble(E=e,Y=Y,X2=X_data[,2],X3=X_data[,3],X7=X_data[,7])
p1<- ggplot(df,aes(X2, E))+geom_point()+
  labs(x = "v10-wind speed south to north (m/s)",y="Residuals (°C)")
p2<- ggplot(df,aes(X3, E))+
  geom_point()+
  labs(x = "sp-sea level pressure (Pa)",y="Residuals (°C)")

p3<-ggplot(df,aes(X7, E))+geom_point()+
  labs(x = "hcc-clouds at 400 millibars (mb)",y="Residuals (°C)")

p4<-ggplot(df,aes(Y, E))+geom_point()+
  labs(x = "Fitted values (°C)",y="Residuals (°C)")

rplot2<-ggplot(df,aes(X_data, E)) +
  geom_histogram(aes(x=E,y=..density..),
                 binwidth=0.5,color="black", fill="white")+
  xlab("Residuals (°C)")+ylab("Density")

rplot3<-ggplot(df, aes(sample = E)) + stat_qq() + stat_qq_line()

figure <- ggarrange(p1, p2,p3,p4,rplot2, rplot3,
                    ncol = 3, nrow = 2)
figure

# box cox
bc= boxcox(mod,lambda=seq(0,6,0.01),plotit=T)
# get lambda
lambda = bc$x[order(bc$y, decreasing = TRUE)[1]]
lambda

### (9) ###
library(rjags)

# Define model
model_string <- "model{
```

```
# Define likelihood
for(i in 1:n){
Y[i]   ~ dnorm(mu[i],inv.var)
mu[i] <- beta[1]+inprod(X[i,],beta[2:(p+1)])
}

# Define prior for beta
for(j in 1:(p+1)){
beta[j] ~ dnorm(0,inv.var.beta)
}

# Prior for 1/sigma^2 is Gamma
# So prior for sigma^2 is inverse Gamma
inv.var   ~ dgamma(0.01, 0.01)
sigma     <- 1/sqrt(inv.var)
}"

# Setting up the design matrix
n=length(X_data$u10)
p=dim(X_data)[2]
# Try when inv.var.beta=1e-15s
model <- jags.model(textConnection(model_string),
                    data = list(Y=Y,n=n,p=p,X=X_data,inv.var.beta=1e-15))
update(model, 10000, progress.bar="none");
samp <- coda.samples(model,
                     variable.names=c("beta","sigma"),
                     n.iter=20000, progress.bar="none")

# Results
summary(samp)

# Extract posterior mean
beta_postmean=summary(samp)$statistics[,1]
beta_postmean

summ.lm<-summary(mod)
# Compare with OLS estimate
c(summ.lm$coefficients[,1],summ.lm$sigma)


# Try when inv.var.beta=1e+15
n=length(X_data$u10)
p=dim(X_data)[2]
model <- jags.model(textConnection(model_string),
                    data = list(Y=Y,n=n,p=p,X=X_data,inv.var.beta=1e+15))
update(model, 10000, progress.bar="none");

samp <- coda.samples(model,
                     variable.names=c("beta","sigma"),
                     n.iter=20000, progress.bar="none")
# check results
summary(samp)

beta_postmean=summary(samp)$statistics[,1]
beta_postmean

summ.lm<-summary(mod)
```

```r
c(summ.lm$coefficients[,1],summ.lm$sigma)

# Run for actual experiment where inv.var.beta gradually increase
num_it <- 700
coeff_matrix=matrix(NaN,nrow=2,ncol=num_it)
for (i in 1:num_it){
  model <- jags.model(textConnection(model_string),
                      data = list(Y=Y,n=n,p=p,X=X_data,inv.var.beta=i*45),
                      quiet=TRUE)
  update(model, 1000, progress.bar="none");
  samp <- coda.samples(model,
                       variable.names=c("beta","sigma"),
                       n.iter=1000, progress.bar="none")
  coeff_matrix[1,i]=summary(samp)$statistics[,1][1]
  coeff_matrix[2,i]=summary(samp)$statistics[,1][3]
}
# plot result
par(mfrow=c(1,2))
plot(coeff_matrix[1,], type='l', xlab = "Inverse Variance", ylab = "beta_0")
plot(coeff_matrix[2,], type='l', xlab = "Inverse Variance", ylab = "beta_2")
par(mfrow=c(1,1))
################################################################################


################################## Question 3 ##################################

### (5) ###
# Define model
model_string <- "model{

# Likelihood Definition
for(i in 1:n){
  Y[i]   ~ dnorm(mu[i],inv.var)
  mu[i] <- beta[1]+beta[2]*X[i]
}

# Prior Definition for Beta0 and Beta1
for(j in 1:2){
  beta[j] ~ ddexp(0,b)
}

# Prior for 1/sigma^2
inv.var   ~ dgamma(0.01, 0.01)
sigma     <- 1/sqrt(inv.var)

}"

# set variables for experiment
set.seed(1)                             # set seed
n = 100                                 # number of data
X<-seq(0, 1, length = n)                # generate sequence of numbers for X
beta0=5                                 # define beta0
beta1=3                                 # define beta1
sigma=1                                 # define sigma
Y=matrix(NaN,nrow=n)                    # define Y matrix to store values for Y
epsilon = rnorm(n, mean=0, sd=sigma)    # define epsilon
Y = beta0 + beta1*X + epsilon           # calculate Y and store result
num_it <- 100                           # number of iteration throughout the model
```

```
N<-length(Y)                          # data size
coeff=matrix(NaN,nrow=2,ncol=num_it)  # define matrix to store result of
# posterior mean of each beta

# run experiment
for (i in 1:num_it){
  model <- jags.model(textConnection(model_string),
                      # increase b as iteration goes on
                      data = list(Y=Y,n=N,X=X,b=(i*100)),
                      quiet=TRUE)
  update(model, 1000, progress.bar="none");
  samp <- coda.samples(model,
                       variable.names=c("beta","sigma"),
                       n.iter=1000, progress.bar="none")
  # store results to matrix coeff
  coeff[1,i]=summary(samp)$statistics[,1][1] # result of beta0
  coeff[2,i]=summary(samp)$statistics[,1][2] # result of beta1
}

# plot results
par(mfrow=c(1,2))
plot(coeff[1,], type='l', xlab = "parameter b (scale)", ylab = "beta_0")
plot(coeff[2,], type='l', xlab = "parameter b (scale)", ylab = "beta_1")
par(mfrow=c(1,1))
###########
################################################################################
```