Problem Set 6, Stats 506 F19

Jie Cao

December 10, 2019

Here is the list of files I submitted for this problem set 6:

1. Question 1

- ps6_q1.sas (SAS script)
- ps6_q1.log (SAS log)
- ps6_q1.csv (Summary table presenting results)

2. Question 2

- ps6_q2.sas (SAS script)
- ps6_q2.log (SAS log)
- ps6_q2_b.csv (Summary table answering question 2b)
- ps6_q2_c.csv (Summary table answering question 2c)

3. Question 3

- ps6_q3.sas (SAS script)
- ps6_q3.log (SAS log)
- ps6_q3_b.csv (Summary table answering quetion 3b/2b)
- ps6_q3_c.csv (Summary table answering question 3c/2c)

Question 1

Here's my solution text to question 1.

```
## ```SAS
## /*************************************************************************/
## * Stats 506, Fall 2019                                                   ;
## * Problem Set 6, Question 1                                              ;
## *------------------------------------------------------------------------;
## * This script solves question 1 for Problem Set 6:                       ;
## * 1. Fit a linear mixed model to explore how each curvature measure differ ;
## *     by condition                                                       ;
## *------------------------------------------------------------------------;
## * Author: Jie Cao (caojie@umich.edu)                                   ;
## * Last updated on: Dec 10, 2019                                         ;
## /*************************************************************************/
##
##
## * 80: ------------------------------------------------------------------;
##
##
## /* Directories */
## libname ps6 "M:\506\hw\hw6";
##
## /* Import csv data */
## proc import datafile = "M:\506\hw\hw6\mousetrap_data.csv"
##  out = mouse_data
##  dbms = csv replace;
## run;
##
## /* Log transform four curvature measures */
## data mouse_data;
##  set mouse_data;
##  log_tot_dist = log(tot_dist);
##  log_max_abs_dev = log(max_abs_dev);
##  log_avg_abs_dev = log(avg_abs_dev);
##  log_AUC = log(AUC);
## run;
##
## /* Fit linear mixed model:
##     Response: log-tranformed curvature measure
##  Predictors: Condition
##  Random effect (intercept): subject, Exemplar*/
##
## /* Macro to loop over four measures and fit a LMM for each */
## %let y = tot_dist max_abs_dev avg_abs_dev AUC;
## %macro looplmm(vlist);
## %let i = 1;
## %do %while (%scan(&vlist., &i.) ne );
##  %let this_y = %scan(&vlist., &i.);
##
##  /* Drop observations with non-valid measure */
##  data model_data;
##      set mouse_data;
```

```
##      if log_&this_y. = . then delete;
##  run;
##
##  /* LMM for this curvature meature */
##  proc mixed data = model_data method = ML;
##      /* Factor variables */
##      class Condition(ref = "Typical") subject_nr Exemplar;
##      /* Model formular - response variable & fixed effect */
##      model log_&this_y. = Condition / solution;
##      /* Random intercepts */
##      random intercept / subject = subject_nr;
##      random intercept / subject = Exemplar;
##      /* Output estimates for fixed effect and covariance parameters */
##      ods output SolutionF = fe_&this_y. CovParms = re_&this_y.;
##  run;
##
##  /* Calculate relative effect and 95% CI for the fixed effect */
##  data fe_&this_y.(keep = Effect Condition re lci uci);
##      set fe_&this_y.;
##      where Condition = "Atypical";
##      re = exp(Estimate);
##      lci = exp(Estimate - 1.96 * StdErr);
##      uci = exp(Estimate + 1.96 * StdErr);
##  run;
##  /* Add measusre name to the dataset */
##  data fe_&this_y.;
##      set fe_&this_y.;
##      measure = "&this_y.";
##  run;
##
##  /* Calculate standard deviations for each variance component */
##  data re_&this_y.(drop = estimate);
##      set re_&this_y.;
##      if CovParm = "Residual" then Subject = "Error";
##      sd = sqrt(Estimate);
##  run;
## %let i = %eval(&i. + 1);
## %end;
## %mend;
## %looplmm(&y.);
##
## /* Put fixed effect results for each measure together */
## data fe_all (keep = measure relative_effect);
##  length measure $26.;
##  set fe_tot_dist(in = a)
##      fe_max_abs_dev(in = b)
##      fe_avg_abs_dev(in = c)
##      fe_auc(in = d);
##
##  if a then measure = "Total Distance";
##  if b then measure = "Maximum Absolute Deviation";
##  if c then measure = "Average Absolute Deviation";
##  if d then measure = "AUC";
##
```

```
##  relative_effect = cat(put(re, f4.2 -L), ' (',
##                        put(lci, f4.2 -L), ', ',
##                        put(uci, f4.2 -L), ')');
##
##  /* Add label to variables */
##  label measure = "Measure"
##        relative_effect = "Relative effect (95% CI)";
##
## run;
##
## /* Put standard deviation for each covariance component together */
## data re_all(drop = CovParm);
##  length measure $26.;
##  set re_tot_dist(in = a)
##      re_max_abs_dev(in = b)
##      re_avg_abs_dev(in = c)
##      re_auc(in = d);
##
##  if a then measure = "Total Distance";
##  if b then measure = "Maximum Absolute Deviation";
##  if c then measure = "Average Absolute Deviation";
##  if d then measure = "AUC";
## run;
##
## /* Transpose standard deviation table */
## proc sort data = re_all;
##  by measure;
## run;
## proc transpose data = re_all out = re_wide(drop = _name_);
##  by measure;
##  id Subject;
##  var sd;
## run;
## /* Add label to variable */
## data re_wide;
##  set re_wide;
##  label measure = "Measure"
##        subject_nr = "Subject"
##        Exemplar = "Exemplar"
##        Error = "Error";
## run;
##
## /* Merge two tables for a final output */
## proc sql;
##  create table q1_out as
##  select a.measure,
##         a.relative_effect,
##         b.subject_nr,
##         b.Exemplar,
##         b.Error
##  from fe_all a
##  left join re_wide b
##  on a.measure = b.measure;
## quit;
```

```
##
## /* Export output to a csv file */
## proc export data = q1_out dbms = csv
##  outfile = "M:\506\hw\hw6\ps6_q1.csv" label replace;
## run;
## ```
```

Table 1: Model summaries. This table shows the relative effect (with 95% confidence intervals) of the atypical condition on each of four curvature measures. Standard deviations for each variance component - subjects, exemplar, and error are also shown.

Measure	**Relative effect (95% CI)**	**Subject**	**Exemplar**	**Error**
AUC	1.50 (1.24, 1.81)	0.36	0.13	1.06
Average Absolute Deviation	1.92 (1.49, 2.47)	0.50	0.20	1.25
Maximum Absolute Deviation	1.67 (1.33, 2.08)	0.36	0.18	1.07
Total Distance	1.18 (1.09, 1.26)	0.09	0.06	0.31

Question 2

Here's my solution text to question 2. Answers to 2b and 2c are shown on the last page of this document.

```
## ```SAS
## /*****************************************************************************/
## * Stats 506, Fall 2019                                                       ;
## * Problem Set 6, Question 2                                                  ;
## *---------------------------------------------------------------------------*;
## * This script solves question 2 for Problem Set 6:                           ;
## * 2. Use 2015 RECS data to perform following analyses:                       ;
## *    (Use data steps)                                                       ;
## *    a. Reshape the replicate weights to a longer format, save dataset      ;
## *    b. Estimate the national average home temperature at night, among homes ;
## *       that use space heating                                              ;
## *    c. By census division, estimate the average winter home temperatures   ;
## *       at night, during the day with someone home, and during the day      ;
## *       with no one home (when applicable)                                  ;
## *---------------------------------------------------------------------------*;
## * Author: Jie Cao (caojie@umich.edu)                                       ;
## * Last updated on: Dec 10, 2019                                             ;
## /*****************************************************************************/
##
##
## * 80: ----------------------------------------------------------------------*;
##
##
## /* Directories */
## libname ps6 "M:\506\hw\hw6";
##
## /* Formats */
## proc format library = ps6.recs_format;
##  value division
##      1 = "New England"
##      2 = "Middle Atlantic"
##      3 = "East North Central"
##      4 = "West North Central"
##      5 = "South Atlantics"
##      6 = "East South Central"
##      7 = "West South Central"
##      8 = "Mountain North"
##      9 = "Mountain South"
##      10 = "Pacific";
##  /*
##  value $uatyp
##      "U" = "Urban Area"
##      "C" = "Urban Cluster"
##      "R" = "Rural";
##  */
##  /*
##  value fuelheat
##      1 = "Natural gas from underground pipes"
##      2 = "Propane (bottled gas)"
##      3 = "Fuel oil/kerosene"
```

```
##      5 = "Electricity"
##      7 = "Wood (cordwood or pellets)"
##      21 = "Some other fuel"
##      other = "N/A";
##  */
##  /*
##  value equipmuse
##      1 = "Set one temperature and leave it there most of time"
##      2 = "Manually adjust the temperature at night or when no one is at home"
##      3 = "Program the thermostat to automatically adjust the temperature during the day and night at
##      4 = "Turn equipment on or off as needed"
##      5 = "Our household does not have control over the equipment"
##      9 = "Other"
##      other = "N/A";
##  */
## run;
##
## /* Format catalog and search order */
## options fmtsearch = (ps6.recs_format);
##
##
## /* Import 2015 RECS data from web */
## filename recs url "https://www.eia.gov/consumption/residential/data/2015/csv/recs2015_public_v4.csv"
## proc import
##  file = recs
##  out = recs
##  dbms = csv;
## run;
##
## /* Keep variables needed for this question */
## data recs;
##  set recs;
##  format division division. /*uatyp10 $uatyp. fuelheat fuelheat. equipmuse equipmuse.*/;
##  keep doeid division /*uatyp10*/ heathome /*fuelheat equipmuse*/
##       temphome tempgone tempnite
##       nweight brrwt1-brrwt96;
## run;
##
##
##
## ********************************************************;
## * a. Reshape the replicate weights to a longer format    ;
## ********************************************************;
## proc transpose data = recs
##  out = ps6.brrwt_long(rename = (_name_ = brrid col1 = nweight_r));
##  by doeid;
##  var brrwt1-brrwt96;
## run;
##
##
##
## ********************************************************;
## * b. Estimate national average home temperature at night, ;
## *    among homes that use space heating                    ;
```

```
## ********************************************************;
## /* Select homes that use space heating */
## data recs_homeheat;
##  set recs(keep = doeid heathome tempnite nweight);
##  where heathome = 1;
## run;
##
## /* Point estimate */
## proc means data = recs_homeheat mean noprint;
##  weight nweight;
##  var tempnite;
##  output out = nat_avg_tempnite_pe(drop = _type_ _freq_) mean = avg_tempnite;
## run;
## data nat_avg_tempnite_pe;
##  measure = "National Temperature at Night";
##  set nat_avg_tempnite_pe;
## run;
##
## /* Estimate standard error using replicate weights */
## * Get weight for each replicate;
## data recs_homeheat_r;
##  merge recs_homeheat(in = a) ps6.brrwt_long (in = b);
##  by doeid;
##  if a;
## run;
## * Compute weighted mean temperature at night for each replicate;
## proc summary data = recs_homeheat_r;
##  class brrid;
##  var tempnite;
##  weight nweight_r;
##  output out = nat_avg_tempnite_r mean = avg_tempnite_r;
## run;
## data nat_avg_tempnite_r(drop = _type_ _freq_);
##  measure = "National Temperature at Night";
##  set nat_avg_tempnite_r;
##  if _type_ = 0 then delete;
## run;
## * Compute standard error;
## data nat_avg_tempnite;
##  merge nat_avg_tempnite_pe nat_avg_tempnite_r;
##  by measure;
##  diff = avg_tempnite_r - avg_tempnite;
##  diff2 = diff**2;
## run;
## proc means data = nat_avg_tempnite mean noprint;
##  var diff2;
##  output out = nat_avg_tempnite_se_tmp mean = mean_diff2;
## run;
## data nat_avg_tempnite_se;
##  measure = "National Temperature at Night";
##  set nat_avg_tempnite_se_tmp;
##  se = 2 * sqrt(mean_diff2);
## run;
##
```

```
## /* Final output */
## data nat_avg_tempnite_final(drop = _type_ _freq_ mean_diff2);
##  merge nat_avg_tempnite_pe
##        nat_avg_tempnite_se;
##  by measure;
##  /* Label variables */
##  label measure = "Measurement"
##        avg_tempnite = "Average temperature at night"
##        se = "Standard Error"
##        lci = "Lower bound of 95% CI"
##        uci = "Upper bound of 95% CI"
##        ci = "95% Confidence Interval";
##  format avg_tempnite f5.2 se f5.2;
##  /* Calcualte 95% CI */
##  lci = put(avg_tempnite - 1.96 * se, f5.2 -L);
##  uci = put(avg_tempnite + 1.96 * se, f5.2 -L);
##  ci = cat("(", lci, ", ", uci, ")");
## run;
##
## /* Export results to a csv file */
## proc export data = nat_avg_tempnite_final dbms = csv
##  outfile = "M:\506\hw\hw6\ps6_q2_b.csv" label replace;
## run;
##
##
##
## ********************************************************************************;
## * c. By census division, estimate the average winter home temperatures       ;
## *     at night, during the day with someone home, and during the day         ;
## *     with no one home (when applicable)                                     ;
## ********************************************************************************;
##
## /* Point estimates */
## data recs_temp;
##  set recs(keep = doeid division temphome tempgone tempnite nweight);
## run;
## * Transpose to a long data to compute three temperature at once;
## proc transpose data = recs_temp
##  out = temp_long(rename = (_name_ = type col1 = temp));
##  by doeid division nweight;
##  var temphome tempgone tempnite;
## run;
## * Average temperature for each type by census division;
## proc means data = temp_long mean noprint;
##  class division type;
##  var temp;
##  output out = avg_temp_pe mean = avg_temp;
## run;
## data avg_temp_pe(drop = _type_ _freq_);
##  set avg_temp_pe;
##  where _type_ = 3;
## run;
##
## /* Standard errors */
```

```
## * Get weight for each replicate;
## data temp_long_r;
##  merge temp_long(drop = nweight) ps6.brrwt_long;
##  by doeid;
## run;
## * Compute weighted mean temperature for each type for each replicate, by census division;
## proc summary data = temp_long_r;
##  class brrid division type;
##  var temp;
##  weight nweight_r;
##  output out = avg_temp_r mean = avg_temp_r;
## run;
## data avg_temp_r(drop = _type_ _freq_);
##  set avg_temp_r;
##  where _type_ = 7;
## run;
## * Compute standard error;
## proc sort data = avg_temp_pe;
##  by division type;
## run;
## proc sort data = avg_temp_r;
##  by division type brrid;
## run;
## data avg_temp;
##  merge avg_temp_pe avg_temp_r;
##  by division type;
##  diff = avg_temp_r - avg_temp;
##  diff2 = diff**2;
## run;
## proc means data = avg_temp mean noprint;
##  class division type;
##  var diff2;
##  output out = avg_temp_se_tmp mean = mean_diff2;
## run;
## data avg_temp_se(drop = _type_ _freq_);
##  set avg_temp_se_tmp;
##  where _type_ = 3;
##  se = 2 * sqrt(mean_diff2);
## run;
##
## /* Final output */
## data avg_temp_final(drop = mean_diff2);
##  merge avg_temp_pe
##        avg_temp_se;
##  by division type;
##  /* Labels */
##  label division = "Division"
##        type = "Temperature type"
##        avg_temp = "Point Estimate (average temperature)"
##        se = "Standard Error"
##        lci = "Lower bound of 95% CI"
##        uci = "Upper bound of 95% CI"
##        value = "Point Estimate (95% CI)";
##  /* Format */
```

```
##  format avg_temp f5.2 se f5.2;
##  /* Calcualte 95% CI */
##  lci = put(avg_temp - 1.96 * se, f5.2 -L);
##  uci = put(avg_temp + 1.96 * se, f5.2 -L);
##  /* Point estimate (95% CI) for presentation */
##  value = cat(put(avg_temp, f5.2 -L), " (",
##              lci, ", ",
##              uci, ")");
## run;
## * Transpose to a wide table to present three temperatures separately;
## proc transpose data = avg_temp_final out = avg_temp_wide(drop = _name_ _label_);
##  by division;
##  id type;
##  var value;
## run;
## * Label three temperatures;
## data avg_temp_wide;
##  set avg_temp_wide;
##  label tempnite = "Average temperature (95% CI) at night"
##        temphome = "Average temperature (95% CI) during the day (someone home)"
##        tempgone = "Average temperature (95% CI) with no one home";
## run;
##
## /* Export results to a csv file */
## proc export data = avg_temp_wide dbms = csv
##  outfile = "M:\506\hw\hw6\ps6_q2_c.csv" label replace;
## run;
## ```
```

Question 3

Here's my solution text to question 3. Answers to 3b and 3c (same as answers to 2b and 2c) are shown on the last page of this document.

```
## ```SAS
## /*************************************************************************/
## * Stats 506, Fall 2019                                                    ;
## * Problem Set 6, Question 3                                               ;
## *-------------------------------------------------------------------------;
## * This script solves question 3 for Problem Set 6:                        ;
## * 3. Use 2015 RECS data to perform following analyses:                    ;
## *    (Use proc SQL)                                                       ;
## *    b. Estimate the national average home temperature at night, among homes ;
## *       that use space heating                                            ;
## *    c. By census division, estimate the average winter home temperatures ;
## *       at night, during the day with someone home, and during the day    ;
## *       with no one home (when applicable)                                ;
## *-------------------------------------------------------------------------;
## * Author: Jie Cao (caojie@umich.edu)                                     ;
## * Last updated on: Dec 10, 2019                                          ;
## /*************************************************************************/
##
##
## * 80: ------------------------------------------------------------------------;
##
##
## /* Directories */
## libname ps6 "M:\506\hw\hw6";
##
##
## /* Formats */
## proc format library = ps6.recs_format;
##  value division
##      1 = "New England"
##      2 = "Middle Atlantic"
##      3 = "East North Central"
##      4 = "West North Central"
##      5 = "South Atlantics"
##      6 = "East South Central"
##      7 = "West South Central"
##      8 = "Mountain North"
##      9 = "Mountain South"
##      10 = "Pacific";
##  /*
##  value $uatyp
##      "U" = "Urban Area"
##      "C" = "Urban Cluster"
##      "R" = "Rural";
##  */
##  /*
##  value fuelheat
##      1 = "Natural gas from underground pipes"
##      2 = "Propane (bottled gas)"
```

```
##      3 = "Fuel oil/kerosene"
##      5 = "Electricity"
##      7 = "Wood (cordwood or pellets)"
##      21 = "Some other fuel"
##      other = "N/A";
##  */
##  /*
##  value equipmuse
##      1 = "Set one temperature and leave it there most of time"
##      2 = "Manually adjust the temperature at night or when no one is at home"
##      3 = "Program the thermostat to automatically adjust the temperature during the day and night at
##      4 = "Turn equipment on or off as needed"
##      5 = "Our household does not have control over the equipment"
##      9 = "Other"
##      other = "N/A";
##  */
## run;
##
## /* Format catalog and search order */
## options fmtsearch = (ps6.recs_format);
##
## /* Import 2015 RECS data from web */
## filename recs url "https://www.eia.gov/consumption/residential/data/2015/csv/recs2015_public_v4.csv"
## proc import
##  file = recs
##  out = recs
##  dbms = csv;
## run;
##
## /* Keep variables needed for this question */
## proc sql noprint;
##  /* List of BRRWT variables */
##  select name into: brrwts separated by ','
##  from dictionary.columns
##  where upcase(libname) = "WORK"
##  and upcase(memname) = "RECS"
##  and upcase(name) contains "BRRWT";
##
##  create table recs_new as
##  select doeid,
##         division format division.,
##         heathome,
##         temphome,
##         tempgone,
##         tempnite,
##         nweight,
##         &brrwts.
##  from recs;
## quit;
##
##
## ************************************************************;
## * b. Estimate national average home temperature at night, ;
## *    among homes that use space heating                    ;
```

```
## ********************************************************;
## proc sql;
##  create table recs_homeheat as
##  select doeid, heathome, tempnite, nweight
##  from recs_new
##  where heathome = 1;
## quit;
##
## /* Point estimate */
## proc sql;
##  create table nat_avg_tempnite_pe as
##  select "National Temperature at Night" as measure,
##         sum(nweight * tempnite) / sum(nweight) as avg_tempnite
##  from recs_homeheat;
## quit;
##
## /* Estimate standard error using replicate weights */
## proc sql;
##  * Get weight for each replicate;
##  create table recs_homeheat_r as
##  select a.doeid,
##         a.tempnite,
##         b.brrid,
##         b.nweight_r
##  from recs_homeheat a
##  left join ps6.brrwt_long b
##  on a.doeid = b.doeid;
##
##  * Compute weighted mean temperature at night for each replicate;
##  create table nat_avg_tempnite_r as
##  select "National Temperature at Night" as measure,
##         sum(nweight_r * tempnite) / sum(nweight_r) as avg_tempnite_r
##  from recs_homeheat_r
##  group by brrid;
##
##  * Intermediate step to compute standard error;
##  create table nat_avg_tempnite as
##  select a.measure,
##         a.avg_tempnite,
##         b.avg_tempnite_r,
##         (avg_tempnite_r - avg_tempnite)**2 as diff2
##  from nat_avg_tempnite_pe a
##  left join nat_avg_tempnite_r b
##  on a.measure = b.measure;
##
##  * Compute standard error;
##  create table nat_avg_tempnite_se as
##  select measure,
##         2 * sqrt(mean(diff2)) as se
##  from nat_avg_tempnite
##  group by measure;
## quit;
##
## /* Final output */
```

```
## proc sql;
##  create table nat_avg_tempnite_final as
##  select a.measure
##              label = "Measurement",
##         a.avg_tempnite as avg_tempnite format f5.2
##              label = "Average temperature at night" ,
##         b.se as se format f5.2
##              label = "Standard Error",
##         put(a.avg_tempnite - 1.96 * b.se, f5.2 -L) as lci
##              label = "Lower bound of 95% CI",
##         put(a.avg_tempnite + 1.96 * b.se, f5.2 -L) as uci
##              label = "Upper bound of 95% CI",
##         cat("(", calculated lci, ", ", calculated uci, ")") as ci
##              label = "95% Confidence Interval"
##  from nat_avg_tempnite_pe a
##  left join nat_avg_tempnite_se b
##  on a.measure = b.measure;
## quit;
##
## /* Export results to a csv file */
## proc export data = nat_avg_tempnite_final dbms = csv
##  outfile = "M:\506\hw\hw6\ps6_q3_b.csv" label replace;
## run;
##
##
## ********************************************************************************;
## * c. By census division, estimate the average winter home temperatures       ;
## *     at night, during the day with someone home, and during the day         ;
## *     with no one home (when applicable)                                     ;
## ********************************************************************************;
##
## /* Point estimates */
## proc sql;
##  create table recs_temp as
##  select doeid,
##         division format division.,
##         temphome, tempgone, tempnite,
##         nweight
##  from recs;
## quit;
## * Transpose to a long data to compute three temperature at once;
## proc transpose data = recs_temp
##  out = temp_long(rename = (_name_ = type col1 = temp));
##  by doeid division nweight;
##  var temphome tempgone tempnite;
## run;
## * Point estimate- average temperature for each type by census division;
## proc sql;
##  create table avg_temp_pe as
##  select division,
##         type,
##         mean(temp) as avg_temp
##  from temp_long
##  group by division, type;
```

```
## quit;
##
## /* Standard errors */
## proc sql;
##  * Get weight for each replicate;
##  create table temp_long_r as
##  select a.doeid,
##         a.division,
##         a.type,
##         a.temp,
##         b.brrid,
##         b.nweight_r
##  from temp_long a
##  left join ps6.brrwt_long b
##  on a.doeid = b.doeid;
##
##  * Compute weighted mean temperature for each type for each replicate, by census division;
##  create table avg_temp_r as
##  select brrid,
##         division,
##         type,
##         sum(nweight_r * temp) / sum(nweight_r) as avg_temp_r
##  from temp_long_r
##  group by brrid, division, type;
##
##  * Intermediate step to compute standard error;
##  create table avg_temp as
##  select a.division,
##         a.type,
##         a.avg_temp,
##         b.avg_temp_r,
##         (avg_temp_r - avg_temp)**2 as diff2
##  from avg_temp_pe a
##  left join avg_temp_r b
##  on a.division = b.division and a.type = b.type;
##
##  * Compute standard errors;
##  create table avg_temp_se as
##  select division,
##         type,
##         2 * sqrt(mean(diff2)) as se
##  from avg_temp
##  group by division, type;
## quit;
##
## /* Final output */
## proc sql;
##  create table avg_temp_final as
##  select a.division
##              label = "Division",
##         a.type
##              label = "Temperature type",
##         a.avg_temp
##              label = "Point Estimate (average temperature)",
```

```
##         b.se as se format f5.2
##              label = "Standard Error",
##         put(a.avg_temp - 1.96 * b.se, f5.2 -L) as lci
##              label = "Lower bound of 95% CI",
##         put(a.avg_temp + 1.96 * b.se, f5.2 -L) as uci
##              label = "Upper bound of 95% CI",
##         cat(put(a.avg_temp, f5.2 -L),
##             " (", calculated lci, ", ", calculated uci, ")") as value
##              label = "Point Estimate (95% CI)"
##  from avg_temp_pe a
##  left join avg_temp_se b
##  on a.division = b.division and a.type = b.type;
## quit;
##
## * Transpose to a wide table to present three temperatures separately;
## proc transpose data = avg_temp_final out = avg_temp_wide(drop = _name_ _label_);
##  by division;
##  id type;
##  var value;
## run;
## * Label three temperatures;
## proc sql;
##  alter table avg_temp_wide
##  modify tempnite label = "Average temperature (95% CI) at night",
##         temphome label = "Average temperature (95% CI) during the day (someone home)",
##         tempgone label = "Average temperature (95% CI) with no one home";
## quit;
##
## /* Export results to a csv file */
## proc export data = avg_temp_wide dbms = csv
##  outfile = "M:\506\hw\hw6\ps6_q3_c.csv" label replace;
## run;
```
```

(2b/3b) What is the national average home temperature at night, among homes that use space heating?

The national average home temperature (95% CI) at night, among homes that use space heating is 68.11 (67.93, 68.28) °F.

(2c/3c) What is the average winter home temperatures at night, during the day with someone home, and during the day with no one home (when applicable), by census division?

Table 2: National average temperature (95% CI) during the day with someone home, during the day with no one home, and at night, by census division

Division	**Average temperature (95% CI) with no one home**	**Average temperature (95% CI) during the day (someone home)**	**Average temperature (95% CI) at night**
New England	64.22 (62.88, 65.57)	67.79 (66.29, 69.29)	65.10 (63.12, 67.09)
Middle Atlantic	66.64 (65.93, 67.35)	69.33 (68.74, 69.92)	67.51 (66.63, 68.39)
East North Central	66.93 (66.45, 67.41)	69.72 (69.37, 70.07)	67.97 (67.56, 68.37)
West North Central	67.31 (66.57, 68.05)	69.72 (69.21, 70.22)	68.04 (67.54, 68.54)
South Atlantics	63.24 (60.93, 65.55)	65.14 (62.86, 67.42)	63.81 (61.48, 66.14)
East South Central	68.07 (67.15, 68.99)	70.52 (69.78, 71.25)	69.15 (68.55, 69.75)
West South Central	66.36 (63.78, 68.95)	69.16 (66.93, 71.40)	67.69 (65.79, 69.59)
Mountain North	66.04 (65.40, 66.69)	69.26 (68.61, 69.92)	66.66 (65.40, 67.92)
Mountain South	64.69 (63.69, 65.69)	67.71 (65.88, 69.55)	65.42 (62.00, 68.85)
Pacific	56.28 (52.95, 59.60)	60.06 (56.64, 63.48)	57.30 (53.97, 60.64)