WiDS 2020 ICU Mortality: An Interpretable First-Day Risk Exploration

Jonathan Muniz

2026-05-04

Contents

1. Research Question

Question: Which patient characteristics and first-day ICU measurements are most associated with in-hospital mortality?

Outcome: `hospital_death` (0 = survived, 1 = died)

Approach:

- Start with descriptive summaries and missingness assessment.
- Use clear exploratory plots to compare survivors vs non-survivors.
- Fit a simple, interpretable logistic regression (primary model).
- Report ROC AUC and a calibration plot as **apparent (in-sample) diagnostics**.

2. Data Import and Basic Checks

```
data_path <- "training_v2.csv"  # keep the csv in the same folder as this .Rmd

wi_ds <- readr::read_csv(data_path, show_col_types = FALSE)

dim(wi_ds)
```

```
## [1] 91713   186
```

```
names(wi_ds)[1:12]
```

```
##  [1] "encounter_id"          "patient_id"            "hospital_id"
##  [4] "hospital_death"        "age"                   "bmi"
##  [7] "elective_surgery"      "ethnicity"             "gender"
## [10] "height"                "hospital_admit_source" "icu_admit_source"
```

```
skim_without_charts(wi_ds)
```

Table 1: Data summary

Name	wi_ds
Number of rows	91713
Number of columns	186
_____	
Column type frequency:	
character	8
numeric	178
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ethnicity	1395	0.98	5	16	0	6	0
gender	25	1.00	1	1	0	2	0
hospital_admit_source	21409	0.77	3	20	0	15	0
icu_admit_source	112	1.00	5	25	0	5	0
icu_stay_type	0	1.00	5	8	0	3	0
icu_type	0	1.00	4	12	0	8	0
apache_3j_bodysystem	1662	0.98	6	20	0	11	0
apache_2_bodysystem	1662	0.98	6	19	0	10	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
encounter_id	0	1.00	65606.08	37795.09	1.00	32852.00	65665.00	98342.00	131051.00
patient_id	0	1.00	65537.13	37811.25	1.00	32830.00	65413.00	98298.00	131051.00
hospital_id	0	1.00	105.67	62.85	2.00	47.00	109.00	161.00	204.00
hospital_death	0	1.00	0.09	0.28	0.00	0.00	0.00	0.00	1.00
age	4228	0.95	62.31	16.78	16.00	52.00	65.00	75.00	89.00
bmi	3429	0.96	29.19	8.28	14.84	23.64	27.65	32.93	67.81
elective_surgery	0	1.00	0.18	0.39	0.00	0.00	0.00	0.00	1.00
height	1334	0.99	169.64	10.80	137.20	162.50	170.10	177.80	195.59
icu_id	0	1.00	508.36	228.99	82.00	369.00	504.00	679.00	927.00
pre_icu_los_days	0	1.00	0.84	2.49	-24.95	0.04	0.14	0.41	159.09
readmission_status	0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
weight	2720	0.97	84.03	25.01	38.60	66.80	80.30	97.10	186.00
albumin_apache	54379	0.41	2.90	0.68	1.20	2.40	2.90	3.40	4.60
apache_2_diagnosis	1662	0.98	185.40	86.05	101.00	113.00	122.00	301.00	308.00

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
apache_3j_diagnosis	1101	0.99	558.22	463.27	0.01	203.01	409.02	703.03	2201.05
apache_post_operative	0	1.00	0.20	0.40	0.00	0.00	0.00	0.00	1.00
arf_apache	715	0.99	0.03	0.16	0.00	0.00	0.00	0.00	1.00
bilirubin_apache	58134	0.37	1.15	2.17	0.10	0.40	0.60	1.10	51.00
bun_apache	19262	0.79	25.83	20.67	4.00	13.00	19.00	32.00	127.00
creatinine_apache	18853	0.79	1.48	1.53	0.30	0.72	0.98	1.53	11.18
fio2_apache	70868	0.23	0.60	0.26	0.21	0.40	0.50	0.85	1.00
gcs_eyes_apache	1901	0.98	3.47	0.95	1.00	3.00	4.00	4.00	4.00
gcs_motor_apache	1901	0.98	5.47	1.29	1.00	6.00	6.00	6.00	6.00
gcs_unable_apache	1037	0.99	0.01	0.10	0.00	0.00	0.00	0.00	1.00
gcs_verbal_apache	1901	0.98	3.99	1.56	1.00	4.00	5.00	5.00	5.00
glucose_apache	11036	0.88	160.33	90.79	39.00	97.00	133.00	196.00	598.70
heart_rate_apache	878	0.99	99.71	30.87	30.00	86.00	104.00	120.00	178.00
hematocrit_apache	19878	0.78	32.99	6.87	16.20	28.00	33.20	37.90	51.40
intubated_apache	715	0.99	0.15	0.36	0.00	0.00	0.00	0.00	1.00
map_apache	994	0.99	88.02	42.03	40.00	54.00	67.00	125.00	200.00
paco2_apache	70868	0.23	42.18	12.38	18.00	34.40	40.00	47.00	95.00
paco2_for_ph_apache	70868	0.23	42.18	12.38	18.00	34.40	40.00	47.00	95.00
pao2_apache	70868	0.23	131.15	83.61	31.00	77.50	103.50	153.00	498.00
ph_apache	70868	0.23	7.35	0.10	6.96	7.31	7.36	7.42	7.59
resprate_apache	1234	0.99	25.81	15.11	4.00	11.00	28.00	36.00	60.00
sodium_apache	18600	0.80	137.97	5.28	117.00	135.00	138.00	141.00	158.00
temp_apache	4108	0.96	36.41	0.83	32.10	36.20	36.50	36.70	39.70
urineoutput_apache	48998	0.47	1738.28	1448.16	0.00	740.36	1386.20	2324.55	8716.67
ventilated_apache	715	0.99	0.33	0.47	0.00	0.00	0.00	1.00	1.00
wbc_apache	22012	0.76	12.13	6.92	0.90	7.50	10.40	15.10	45.80
d1_diasbp_invasive_max	67984	0.26	78.76	21.73	37.00	65.00	75.00	88.00	181.00
d1_diasbp_invasive_min	67984	0.26	46.74	12.86	5.00	39.00	46.00	54.00	89.00
d1_diasbp_max	165	1.00	88.49	19.80	46.00	75.00	86.00	99.00	165.00
d1_diasbp_min	165	1.00	50.16	13.32	13.00	42.00	50.00	58.00	90.00
d1_diasbp_noninvasive_max	1040	0.99	88.61	19.79	46.00	75.00	87.00	99.00	165.00
d1_diasbp_noninvasive_min	1040	0.99	50.24	13.34	13.00	42.00	50.00	58.00	90.00
d1_heartrate_max	145	1.00	103.00	22.02	58.00	87.00	101.00	116.00	177.00
d1_heartrate_min	145	1.00	70.32	17.12	0.00	60.00	69.00	81.00	175.00
d1_mbp_invasive_max	67777	0.26	114.89	49.45	38.00	89.00	101.00	118.00	322.00
d1_mbp_invasive_min	67777	0.26	62.32	18.06	2.00	54.00	62.00	72.00	119.00
d1_mbp_max	220	1.00	104.65	20.81	60.00	90.00	102.00	116.00	184.00
d1_mbp_min	220	1.00	64.87	15.68	22.00	55.00	64.00	75.00	112.00
d1_mbp_noninvasive_max	1479	0.98	104.59	20.70	60.00	90.00	102.00	116.00	181.00
d1_mbp_noninvasive_min	1479	0.98	64.94	15.70	22.00	55.00	64.00	75.00	112.00
d1_resprate_max	385	1.00	28.88	10.70	14.00	22.00	26.00	32.00	92.00
d1_resprate_min	385	1.00	12.85	5.06	0.00	10.00	13.00	16.00	100.00
d1_spo2_max	333	1.00	99.24	1.79	0.00	99.00	100.00	100.00	100.00
d1_spo2_min	333	1.00	90.45	10.03	0.00	89.00	92.00	95.00	100.00
d1_sysbp_invasive_max	67959	0.26	154.27	32.29	71.00	134.00	151.00	170.00	295.00
d1_sysbp_invasive_min	67959	0.26	93.81	24.98	10.00	80.00	92.00	107.00	172.00

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
d1_sysbp_max	159	1.00	148.34	25.73	90.00	130.00	146.00	164.00	232.00
d1_sysbp_min	159	1.00	96.92	20.68	41.00	83.00	96.00	110.00	160.00
d1_sysbp_noninvasive_max	1027	0.99	148.24	25.79	90.00	130.00	146.00	164.00	232.00
d1_sysbp_noninvasive_min	1027	0.99	96.99	20.71	41.03	84.00	96.00	110.00	160.00
d1_temp_max	2324	0.97	37.28	0.69	35.10	36.90	37.11	37.60	39.90
d1_temp_min	2324	0.97	36.27	0.75	31.89	36.10	36.40	36.66	37.80
h1_diasbp_invasive_max	74928	0.18	67.97	16.26	33.00	57.00	66.00	77.00	135.00
h1_diasbp_invasive_min	74928	0.18	56.14	14.14	19.00	46.00	55.00	65.00	104.00
h1_diasbp_max	3619	0.96	75.35	18.41	37.00	62.00	74.00	86.00	143.00
h1_diasbp_min	3619	0.96	62.84	16.36	22.00	52.00	62.00	73.00	113.00
h1_diasbp_noninvasive_max	7350	0.92	75.81	18.48	37.00	63.00	74.00	87.00	144.00
h1_diasbp_noninvasive_min	7350	0.92	63.27	16.42	22.00	52.00	62.00	74.00	114.00
h1_heartrate_max	2790	0.97	92.23	21.82	46.00	77.00	90.00	106.00	164.00
h1_heartrate_min	2790	0.97	83.66	20.28	36.00	69.00	82.00	97.00	144.00
h1_mbp_invasive_max	74844	0.18	94.88	30.81	35.62	78.00	90.00	104.00	293.38
h1_mbp_invasive_min	74844	0.18	75.97	19.23	8.00	63.00	74.00	88.00	140.00
h1_mbp_max	4639	0.95	91.61	20.53	49.00	77.00	90.00	104.00	165.00
h1_mbp_min	4639	0.95	79.40	19.13	32.00	66.00	78.00	92.00	138.00
h1_mbp_noninvasive_max	9084	0.90	91.59	20.55	49.00	77.00	90.00	104.00	163.00
h1_mbp_noninvasive_min	9084	0.90	79.71	19.24	32.00	66.00	79.00	92.00	138.00
h1_resprate_max	4357	0.95	22.63	7.52	10.00	18.00	21.00	26.00	59.00
h1_resprate_min	4357	0.95	17.21	6.07	0.00	14.00	16.00	20.00	189.00
h1_spo2_max	4185	0.95	98.04	3.21	0.00	97.00	99.00	100.00	100.00
h1_spo2_min	4185	0.95	95.17	6.63	0.00	94.00	96.00	99.00	100.00
h1_sysbp_invasive_max	74915	0.18	138.70	29.21	65.00	119.00	136.00	156.00	246.00
h1_sysbp_invasive_min	74915	0.18	114.83	27.97	31.44	95.00	112.00	133.00	198.00
h1_sysbp_max	3611	0.96	133.25	27.56	75.00	113.00	131.00	150.00	223.00
h1_sysbp_min	3611	0.96	116.36	26.51	53.00	98.00	115.00	134.00	194.00
h1_sysbp_noninvasive_max	7341	0.92	133.05	27.68	75.00	113.00	130.00	150.00	223.00
h1_sysbp_noninvasive_min	7341	0.92	116.55	26.62	53.00	98.00	115.00	134.00	195.00
h1_temp_max	21732	0.76	36.71	0.75	33.40	36.40	36.70	37.00	39.50
h1_temp_min	21732	0.76	36.61	0.78	32.90	36.30	36.60	36.94	39.30
d1_albumin_max	49096	0.46	2.97	0.67	1.20	2.50	3.00	3.40	4.60
d1_albumin_min	49096	0.46	2.90	0.67	1.10	2.40	2.90	3.40	4.50
d1_bilirubin_max	53673	0.41	1.14	2.13	0.20	0.40	0.60	1.10	51.00
d1_bilirubin_min	53673	0.41	1.07	2.02	0.20	0.40	0.60	1.00	51.00
d1_bun_max	10514	0.89	25.69	20.47	4.00	13.00	19.00	31.00	126.00
d1_bun_min	10514	0.89	23.77	18.80	3.00	12.00	18.00	29.00	113.09
d1_calcium_max	13069	0.86	8.38	0.74	6.20	7.90	8.40	8.80	10.80
d1_calcium_min	13069	0.86	8.18	0.78	5.50	7.70	8.20	8.70	10.30
d1_creatinine_max	10169	0.89	1.49	1.51	0.34	0.76	1.00	1.50	11.11
d1_creatinine_min	10169	0.89	1.37	1.33	0.30	0.71	0.95	1.40	9.94
d1_glucose_max	5807	0.94	174.64	86.69	73.00	117.00	150.00	201.00	611.00
d1_glucose_min	5807	0.94	114.38	38.27	33.00	91.00	107.00	131.00	288.00
d1_hco3_max	15071	0.84	24.37	4.37	12.00	22.00	24.00	27.00	40.00
d1_hco3_min	15071	0.84	23.17	4.99	7.00	21.00	23.00	26.00	39.00

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
d1_hemaglobin_max	12147	0.87	11.45	2.17	6.80	9.80	11.40	13.00	17.20
d1_hemaglobin_min	12147	0.87	10.89	2.36	5.30	9.20	10.90	12.60	16.70
d1_hematocrit_max	11654	0.87	34.53	6.24	20.40	30.00	34.50	39.00	51.50
d1_hematocrit_min	11654	0.87	32.95	6.85	16.10	28.00	33.20	38.00	50.00
d1_inr_max	57941	0.37	1.60	0.96	0.90	1.10	1.30	1.60	7.76
d1_inr_min	57941	0.37	1.48	0.75	0.90	1.10	1.21	1.50	6.13
d1_lactate_max	68396	0.25	2.93	3.08	0.40	1.20	1.90	3.30	19.80
d1_lactate_min	68396	0.25	2.13	2.11	0.40	1.00	1.50	2.30	15.10
d1_platelets_max	13444	0.85	207.11	89.63	27.00	148.00	196.00	251.00	585.00
d1_platelets_min	13444	0.85	196.77	88.18	18.55	138.00	187.00	242.00	557.45
d1_potassium_max	9585	0.90	4.25	0.67	2.80	3.80	4.20	4.60	7.00
d1_potassium_min	9585	0.90	3.93	0.58	2.40	3.60	3.90	4.30	5.80
d1_sodium_max	10195	0.89	139.12	4.82	123.00	137.00	139.00	142.00	158.00
d1_sodium_min	10195	0.89	137.72	4.92	117.00	135.00	138.00	141.00	153.00
d1_wbc_max	13174	0.86	12.48	6.80	1.20	8.00	11.00	15.20	46.08
d1_wbc_min	13174	0.86	11.31	5.95	0.90	7.40	10.10	13.73	40.90
h1_albumin_max	83824	0.09	3.03	0.73	1.10	2.50	3.10	3.60	4.70
h1_albumin_min	83824	0.09	3.03	0.73	1.10	2.50	3.10	3.60	4.70
h1_bilirubin_max	84619	0.08	1.10	2.03	0.20	0.40	0.60	1.10	40.40
h1_bilirubin_min	84619	0.08	1.10	2.03	0.20	0.40	0.60	1.10	40.40
h1_bun_max	75091	0.18	25.84	21.44	4.00	13.00	18.00	31.00	135.00
h1_bun_min	75091	0.18	25.82	21.42	4.00	13.00	18.00	31.00	135.00
h1_calcium_max	75863	0.17	8.28	0.88	5.60	7.70	8.30	8.80	11.40
h1_calcium_min	75863	0.17	8.28	0.89	5.30	7.70	8.30	8.80	11.31
h1_creatinine_max	74957	0.18	1.53	1.58	0.33	0.79	1.01	1.55	11.60
h1_creatinine_min	74957	0.18	1.53	1.57	0.33	0.79	1.01	1.55	11.57
h1_glucose_max	52614	0.43	167.99	94.72	59.00	111.00	140.00	189.00	695.04
h1_glucose_min	52614	0.43	159.22	89.16	42.00	106.00	134.00	179.00	670.00
h1_hco3_max	76094	0.17	22.50	5.21	6.00	20.00	23.00	25.10	39.00
h1_hco3_min	76094	0.17	22.42	5.21	6.00	20.00	23.00	25.00	39.00
h1_hemaglobin_max	73123	0.20	11.19	2.37	5.10	9.50	11.10	12.80	17.40
h1_hemaglobin_min	73123	0.20	11.04	2.41	5.00	9.30	11.00	12.70	17.30
h1_hematocrit_max	73420	0.20	33.67	6.84	16.00	28.90	33.50	38.40	51.70
h1_hematocrit_min	73420	0.20	33.22	7.03	15.50	28.10	33.00	38.10	51.50
h1_inr_max	57941	0.37	1.60	0.96	0.90	1.10	1.30	1.60	7.76
h1_inr_min	57941	0.37	1.48	0.75	0.90	1.10	1.21	1.50	6.13
h1_lactate_max	84369	0.08	3.07	2.93	0.40	1.30	2.05	3.60	18.10
h1_lactate_min	84369	0.08	3.02	2.88	0.40	1.30	2.00	3.60	18.02
h1_platelets_max	75673	0.17	196.10	92.65	20.00	133.00	181.00	241.00	585.00
h1_platelets_min	75673	0.17	195.48	92.78	20.00	132.00	181.00	240.00	585.00
h1_potassium_max	72102	0.21	4.20	0.76	2.50	3.70	4.10	4.60	7.20
h1_potassium_min	72102	0.21	4.15	0.75	2.50	3.70	4.10	4.50	7.10
h1_sodium_max	72617	0.21	138.24	5.75	114.00	136.00	139.00	141.00	157.00
h1_sodium_min	72617	0.21	137.90	5.68	114.00	135.00	138.00	141.00	157.00
h1_wbc_max	75953	0.17	13.46	6.98	1.10	8.60	12.12	16.80	44.10
h1_wbc_min	75953	0.17	13.42	6.97	1.09	8.60	12.10	16.70	44.10

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
d1_arterial_pco2_max	59271	0.35	45.25	14.67	18.40	36.00	42.80	50.00	111.00
d1_arterial_pco2_min	59271	0.35	38.43	10.94	14.90	32.00	37.00	43.00	85.91
d1_arterial_ph_max	60123	0.34	7.39	0.08	7.05	7.34	7.39	7.44	7.62
d1_arterial_ph_min	60123	0.34	7.32	0.11	6.89	7.27	7.34	7.40	7.56
d1_arterial_po2_max	59262	0.35	165.91	108.01	39.00	88.10	127.00	206.00	540.86
d1_arterial_po2_min	59262	0.35	103.51	61.85	28.00	69.00	85.00	116.00	448.89
d1_pao2fio2ratio_max	66008	0.28	285.67	128.22	54.80	192.29	272.67	365.00	834.80
d1_pao2fio2ratio_min	66008	0.28	223.52	117.55	36.00	132.50	205.00	300.00	604.23
h1_arterial_pco2_max	75959	0.17	44.67	14.63	15.00	36.00	42.10	49.20	111.50
h1_arterial_pco2_min	75959	0.17	43.38	14.11	15.00	35.00	41.00	48.00	107.00
h1_arterial_ph_max	76424	0.17	7.34	0.11	6.93	7.29	7.35	7.41	7.57
h1_arterial_ph_min	76424	0.17	7.33	0.11	6.90	7.28	7.34	7.40	7.56
h1_arterial_po2_max	75945	0.17	163.84	113.46	34.00	80.70	120.00	216.00	534.90
h1_arterial_po2_min	75945	0.17	144.15	98.46	31.00	77.00	107.00	178.00	514.90
h1_pao2fio2ratio_max	80195	0.13	244.40	129.96	42.00	142.00	223.33	328.00	720.00
h1_pao2fio2ratio_min	80195	0.13	235.93	126.46	38.00	136.00	214.00	317.48	654.81
apache_4a_hospital_death_prob	7947	0.91	0.09	0.25	-1.00	0.02	0.05	0.13	0.99
apache_4a_icu_death_prob	7947	0.91	0.04	0.22	-1.00	0.01	0.02	0.06	0.97
aids	715	0.99	0.00	0.03	0.00	0.00	0.00	0.00	1.00
cirrhosis	715	0.99	0.02	0.12	0.00	0.00	0.00	0.00	1.00
diabetes_mellitus	715	0.99	0.23	0.42	0.00	0.00	0.00	0.00	1.00
hepatic_failure	715	0.99	0.01	0.11	0.00	0.00	0.00	0.00	1.00
immunosuppression	715	0.99	0.03	0.16	0.00	0.00	0.00	0.00	1.00
leukemia	715	0.99	0.01	0.08	0.00	0.00	0.00	0.00	1.00
lymphoma	715	0.99	0.00	0.06	0.00	0.00	0.00	0.00	1.00
solid_tumor_with_metastasis	715	0.99	0.02	0.14	0.00	0.00	0.00	0.00	1.00

3. Variable Selection Strategy

The dataset is wide. To keep the analysis interpretable, we:

1. Focus on demographics + admission context + first-day vitals/labs.
2. Avoid identifier fields (e.g., `encounter_id`, `patient_id`) as predictors.
3. Avoid using derived probability scores (e.g., APACHE death probabilities) as main predictors.

3.1 Candidate variables and missingness

We start with an initial candidate set (including lactate), then inspect missingness and decide what stays.

```r
candidate_vars <- c(
  "hospital_death",
  "age", "bmi", "gender", "ethnicity", "elective_surgery", "icu_type", "pre_icu_los_days",
  "d1_heartrate_max", "d1_mbp_min", "d1_resprate_max", "d1_spo2_min", "d1_temp_max",
  "d1_creatinine_max", "d1_bun_max", "d1_wbc_max", "d1_glucose_max", "d1_lactate_max"
)

candidate_df <- wi_ds %>% select(all_of(candidate_vars))

candidate_missing <- candidate_df %>%
  summarise(across(everything(), ~ mean(is.na(.)) * 100)) %>%
  pivot_longer(everything(), names_to = "variable", values_to = "pct_missing") %>%
  arrange(desc(pct_missing))

kable(head(candidate_missing, 18), digits = 2,
      caption = "Missingness (\\% missing) for candidate variables")
```

Table 4: Missingness (% missing) for candidate variables

variable	pct_missing
d1_lactate_max	74.58
d1_wbc_max	14.36
d1_bun_max	11.46
d1_creatinine_max	11.09
d1_glucose_max	6.33
age	4.61
bmi	3.74
d1_temp_max	2.53
ethnicity	1.52
d1_resprate_max	0.42
d1_spo2_min	0.36
d1_mbp_min	0.24
d1_heartrate_max	0.16
gender	0.03
hospital_death	0.00
elective_surgery	0.00
icu_type	0.00
pre_icu_los_days	0.00

Decision rule used here: variables with extremely high missingness are not used in the primary model.

In this dataset, `d1_lactate_max` has ~75% missingness, so we exclude it from the primary workflow. Lactate can still be explored separately as a supplemental analysis if desired.

3.2 Final variable sets

```
plot_vars <- c(
  "hospital_death",
  "age", "bmi", "gender", "ethnicity", "elective_surgery", "icu_type", "pre_icu_los_days",
  "d1_heartrate_max", "d1_mbp_min", "d1_resprate_max", "d1_spo2_min", "d1_temp_max",
  "d1_creatinine_max", "d1_bun_max", "d1_wbc_max", "d1_glucose_max"
)

model_vars <- c(
  "hospital_death",
  "age", "gender", "elective_surgery", "pre_icu_los_days",
  "d1_mbp_min", "d1_spo2_min", "d1_resprate_max",
  "d1_creatinine_max", "d1_bun_max", "d1_wbc_max"
)

lgbm_vars <- c(
  "hospital_death",
  # Demographics & admission
  "age", "bmi", "gender", "ethnicity", "elective_surgery",
  "pre_icu_los_days", "icu_type", "icu_admit_source", "hospital_admit_source",
  # APACHE probability scores (strongest predictors in this dataset)
  "apache_4a_hospital_death_prob", "apache_4a_icu_death_prob",
  # APACHE component vitals/labs
  "gcs_eyes_apache", "gcs_motor_apache", "gcs_verbal_apache",
  "heart_rate_apache", "map_apache", "resprate_apache", "temp_apache",
  "bun_apache", "creatinine_apache", "glucose_apache",
  "hematocrit_apache", "sodium_apache", "wbc_apache",
  "ventilated_apache", "intubated_apache", "arf_apache",
  # Comorbidities
  "aids", "cirrhosis", "diabetes_mellitus", "hepatic_failure",
  "immunosuppression", "leukemia", "lymphoma", "solid_tumor_with_metastasis",
  # Day-1 vitals (min & max)
  "d1_heartrate_max", "d1_heartrate_min",
  "d1_mbp_max", "d1_mbp_min",
  "d1_resprate_max", "d1_resprate_min",
  "d1_spo2_max", "d1_spo2_min",
  "d1_sysbp_max", "d1_sysbp_min",
  "d1_temp_max", "d1_temp_min",
  # Day-1 labs
  "d1_bun_max", "d1_creatinine_max", "d1_wbc_max",
  "d1_glucose_max", "d1_glucose_min",
  "d1_sodium_max", "d1_sodium_min",
  "d1_potassium_max", "d1_potassium_min",
  "d1_hco3_max", "d1_hco3_min",
  "d1_platelets_max", "d1_platelets_min",
  "d1_hemaglobin_max", "d1_hemaglobin_min",
```

```
  "d1_albumin_min", "d1_lactate_max",
  # Hour-1 vitals (early-warning signal)
  "h1_heartrate_max", "h1_heartrate_min",
  "h1_mbp_max", "h1_mbp_min",
  "h1_resprate_max", "h1_spo2_min",
  "h1_sysbp_max", "h1_sysbp_min"
)
```

4. Build Analysis Datasets

We intentionally create separate datasets:

- **df_plot**: used for visualizations (not necessarily complete-case across all variables).
- **df_model**: used for modeling (complete-case across *model variables only*).

```
# Plotting dataset
df_plot <- wi_ds %>%
  select(all_of(plot_vars)) %>%
  mutate(
    outcome          = factor(hospital_death, levels = c(0, 1),
                              labels = c("Survived", "Died")),
    gender           = factor(gender),
    ethnicity        = factor(ethnicity),
    elective_surgery = factor(elective_surgery, levels = c(0, 1),
                              labels = c("No", "Yes")),
    icu_type         = factor(icu_type)
  )

# Modeling dataset (complete-case on MODEL variables only)
df_model <- wi_ds %>%
  select(all_of(model_vars)) %>%
  mutate(
    outcome          = factor(hospital_death, levels = c(0, 1),
                              labels = c("Survived", "Died")),
    gender           = factor(gender),
    elective_surgery = factor(elective_surgery, levels = c(0, 1),
                              labels = c("No", "Yes")),
    # Log-transform pre-ICU LOS to stabilize heavy right skew
    # (raw: mean = 0.84 days, max = 159 days)
    log_pre_icu_los  = log1p(pre_icu_los_days)
  ) %>%
  drop_na()

# LightGBM dataset (complete-case on LGBM variables only)
# LightGBM handles NA internally - no imputation or row-dropping needed.
```

```r
# Categorical columns must be integer-encoded; we save the factor levels
# from training so the test set can be encoded identically

encode_levels <- list()  # will store factor levels for test-set alignment

df_lgbm <- wi_ds %>%
  select(all_of(lgbm_vars)) %>%
  mutate(across(
    c(gender, ethnicity, icu_type, icu_admit_source, hospital_admit_source),
    ~ {
      f <- factor(.x)
      encode_levels[[cur_column()]] <<- levels(f)
      as.integer(f)
    }
  ))

cat("LightGBM training rows:", nrow(df_lgbm), "\n")
```

```
## LightGBM training rows: 91713
```

```r
cat("LightGBM features:     ", ncol(df_lgbm) - 1, "\n")
```

```
## LightGBM features:      72
```

```r
cat("Outcome prevalence:    ", round(mean(df_lgbm$hospital_death, na.rm = TRUE), 4), "\n")
```

```
## Outcome prevalence:     0.0863
```

```r
analysis_sizes <- tibble(
  dataset = c("Full dataset", "Model complete-case"),
  n_rows  = c(nrow(wi_ds), nrow(df_model))
)

kable(analysis_sizes, caption = "Dataset sizes used in this report")
```

Table 5: Dataset sizes used in this report

dataset	n_rows
Full dataset	91713
Model complete-case	71926

4.1 Complete-case selection bias check

Dropping rows with missing values can introduce bias if the dropped patients differ systematically from those retained. We compare mortality rates between the two groups.

```
cc_check <- wi_ds %>%
  select(all_of(model_vars)) %>%
  mutate(
    complete_case  = complete.cases(pick(everything())),
    hospital_death = as.numeric(hospital_death)
  )

bias_summary <- cc_check %>%
  group_by(complete_case) %>%
  summarise(
    n          = n(),
    mort_rate = mean(hospital_death, na.rm = TRUE),
    mean_age   = mean(age, na.rm = TRUE),
    .groups    = "drop"
  ) %>%
  mutate(complete_case = if_else(complete_case, "Retained", "Dropped"))

kable(bias_summary, digits = 3,
      caption = "Comparison of retained vs.\\ dropped observations (complete-case check)")
```

Table 6: Comparison of retained vs. dropped observations (complete-case check)

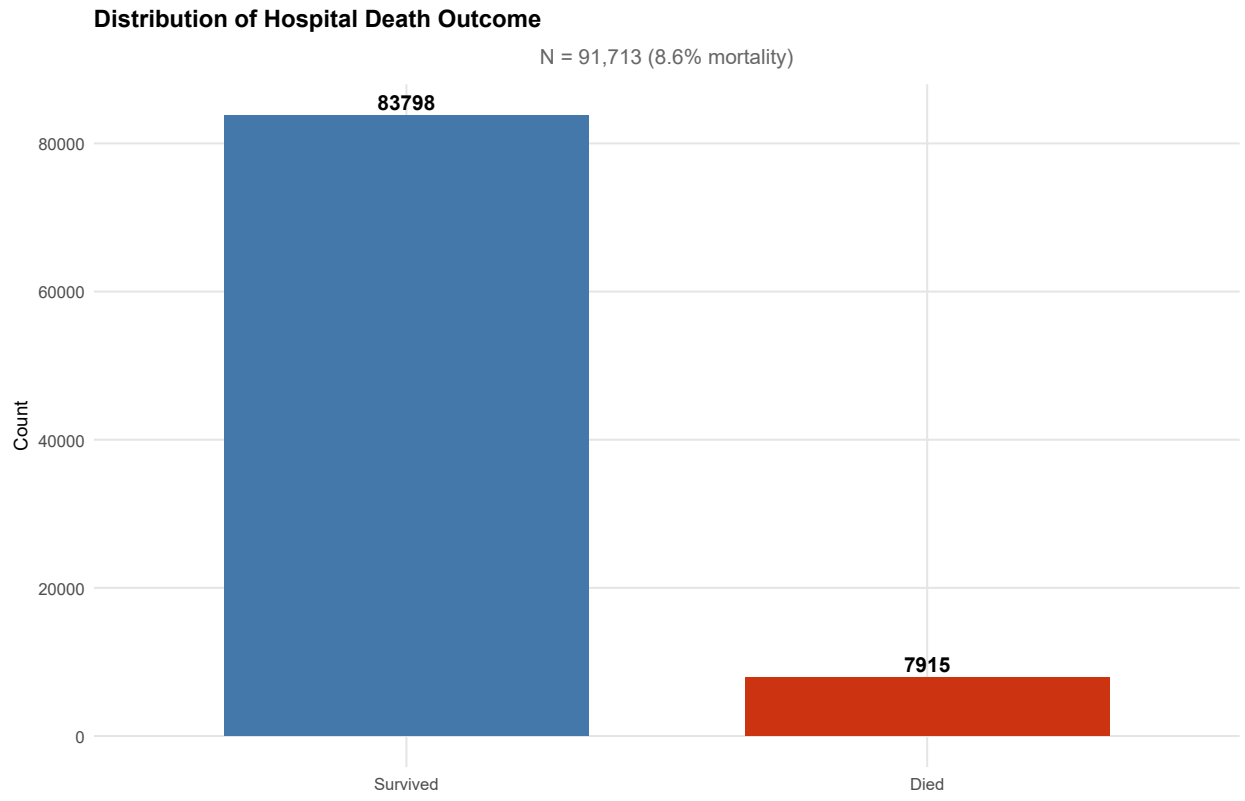
complete_case	n	mort_rate	mean_age
Dropped	19787	0.090	60.361
Retained	71926	0.085	62.731

5. Exploratory Visualizations

5.1 Outcome distribution

```
plot_outcome <- ggplot(df_plot, aes(x = outcome, fill = outcome)) +
  geom_bar(width = 0.7, show.legend = FALSE) +
  geom_text(stat = "count", aes(label = after_stat(count)), vjust = -0.3,
            fontface = "bold") +
  scale_fill_manual(values = outcome_pal) +
  labs(title    = "Distribution of Hospital Death Outcome",
       subtitle = paste0("N = ", format(nrow(df_plot), big.mark = ","),
                         " (",
```

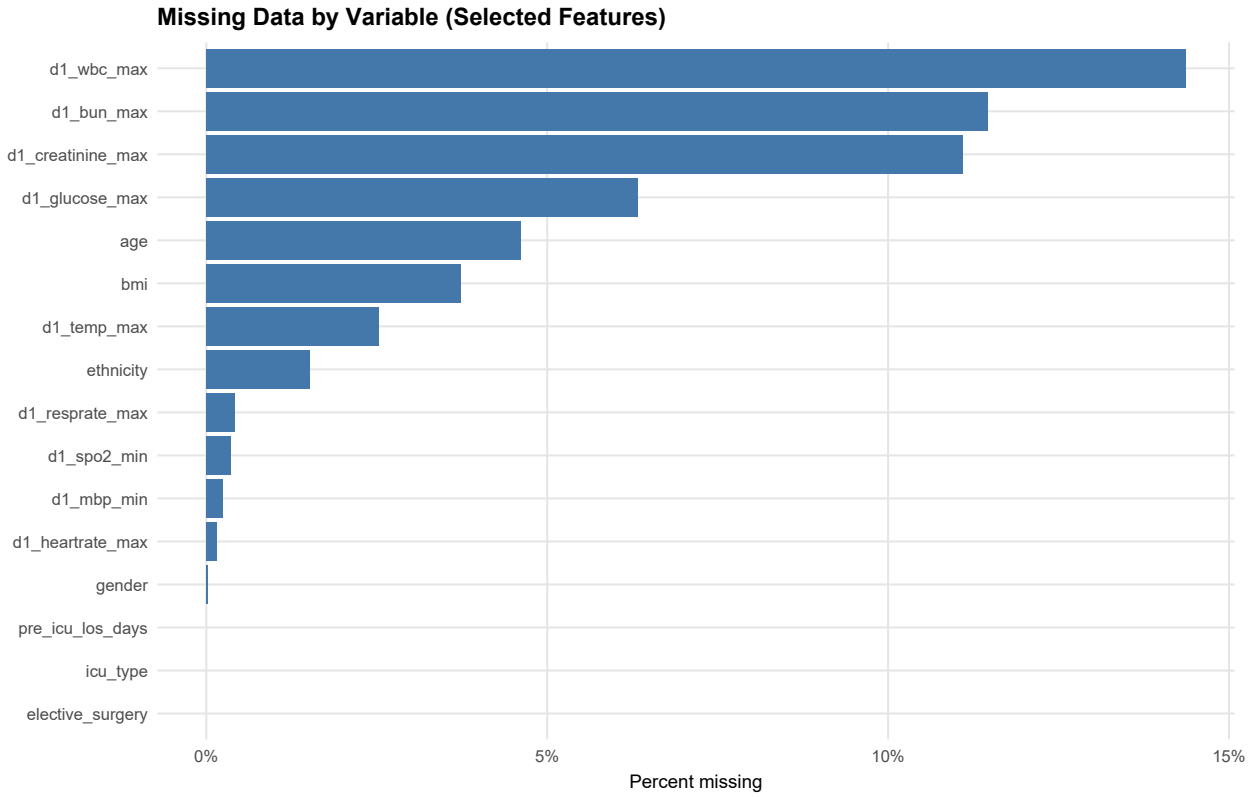
```
                          round(mean(df_plot$hospital_death == 1, na.rm = TRUE) * 100, 1),
                          "% mortality)"),
       x = NULL, y = "Count") +
  theme_proj()

plot_outcome
```



5.2 Missingness in selected Day-1 features

```
missing_summary <- df_plot %>%
  select(-outcome) %>%
  summarise(across(everything(), ~ mean(is.na(.)) * 100)) %>%
  pivot_longer(everything(), names_to = "variable", values_to = "pct_missing")

plot_missing <- missing_summary %>%
  filter(variable != "hospital_death") %>%
  ggplot(aes(x = reorder(variable, pct_missing), y = pct_missing)) +
  geom_col(fill = "#4477AA") +
  coord_flip() +
  labs(title = "Missing Data by Variable (Selected Features)",
       x = NULL, y = "Percent missing") +
  scale_y_continuous(labels = label_number(suffix = "%")) +
```

```
  theme_proj()

plot_missing
```

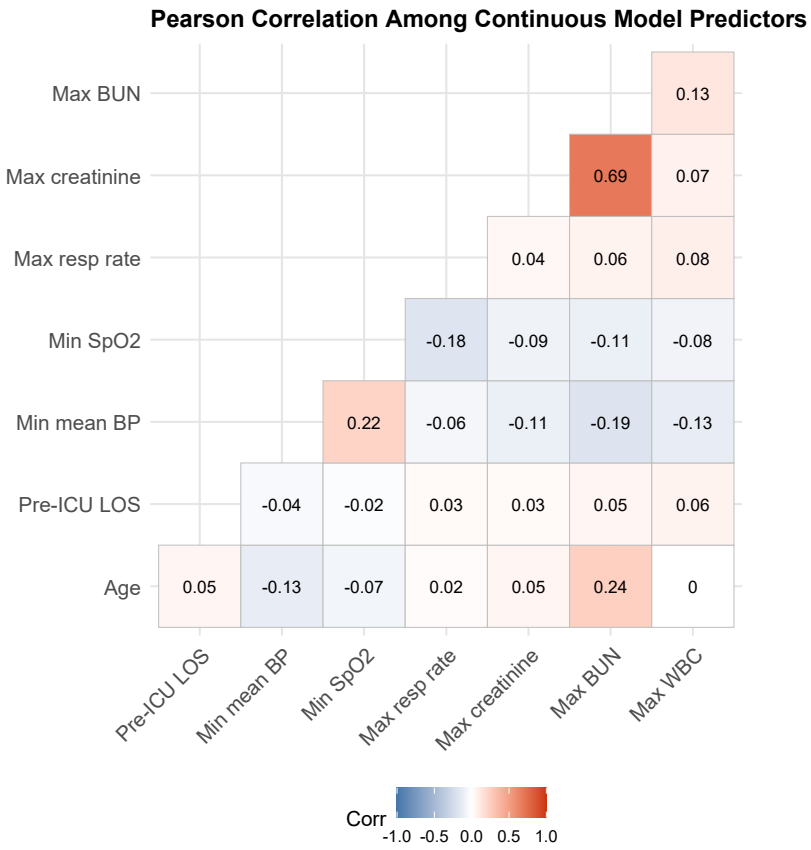


5.3 Correlation among continuous model predictors

Before modeling, we inspect correlations among continuous predictors to identify potential multicollinearity. BUN and creatinine are both renal function markers and are expected to be correlated.

```
cont_model_vars <- c(
  "age", "pre_icu_los_days",
  "d1_mbp_min", "d1_spo2_min", "d1_resprate_max",
  "d1_creatinine_max", "d1_bun_max", "d1_wbc_max"
)

cor_mat <- wi_ds %>%
  select(all_of(cont_model_vars)) %>%
  drop_na() %>%
  cor(method = "pearson")

# Readable labels
heatmap_labels <- c(
  age                = "Age",
```

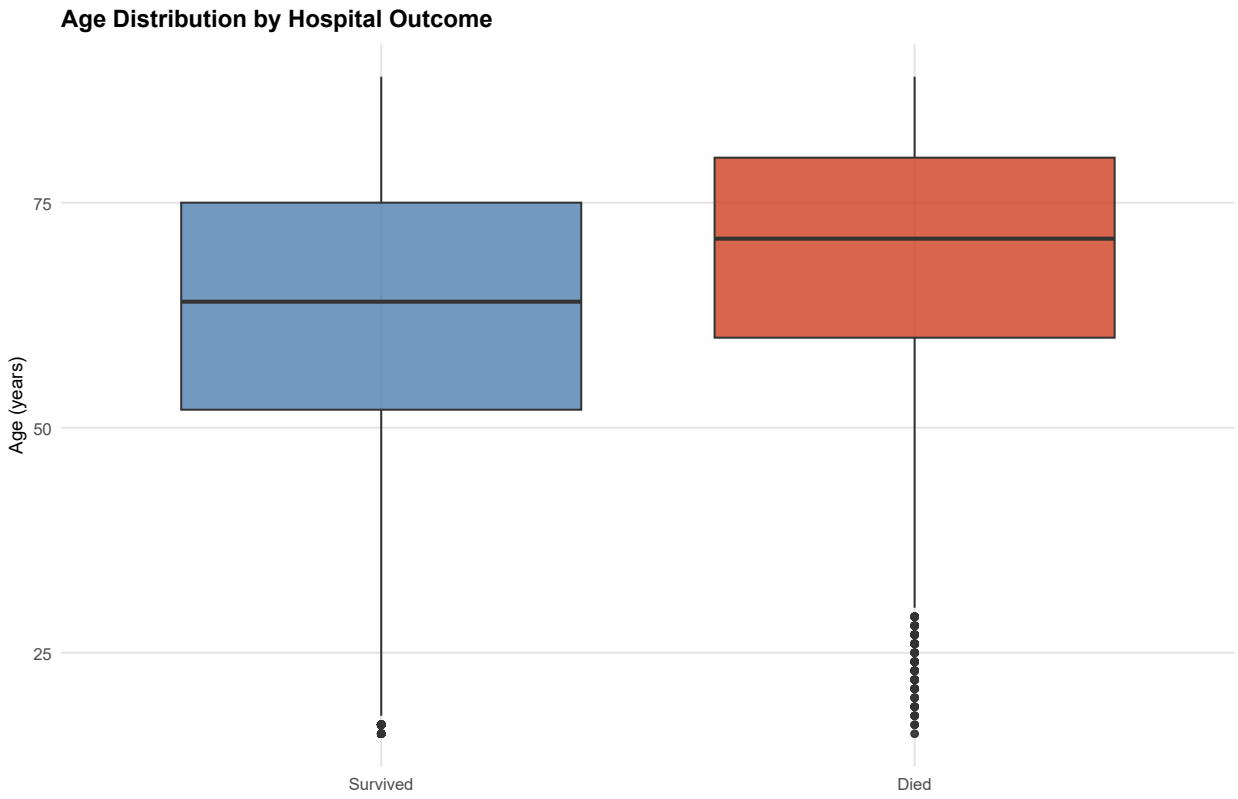
```
  pre_icu_los_days  = "Pre-ICU LOS",
  d1_mbp_min        = "Min mean BP",
  d1_spo2_min       = "Min SpO2",
  d1_resprate_max   = "Max resp rate",
  d1_creatinine_max = "Max creatinine",
  d1_bun_max        = "Max BUN",
  d1_wbc_max        = "Max WBC"
)

rownames(cor_mat) <- heatmap_labels[rownames(cor_mat)]
colnames(cor_mat) <- heatmap_labels[colnames(cor_mat)]

plot_corr <- ggcorrplot(
  cor_mat,
  type     = "lower",
  lab      = TRUE,
  lab_size = 3.5,
  colors   = c("#4477AA", "white", "#CC3311"),
  title    = "Pearson Correlation Among Continuous Model Predictors",
  ggtheme  = theme_proj()
)

plot_corr
```



BUN and creatinine show a moderate-to-strong positive correlation, which is expected since both reflect kidney function. We retain both in the model but will check variance inflation factors (VIFs) after fitting to quantify the collinearity.

5.4 Age by hospital outcome

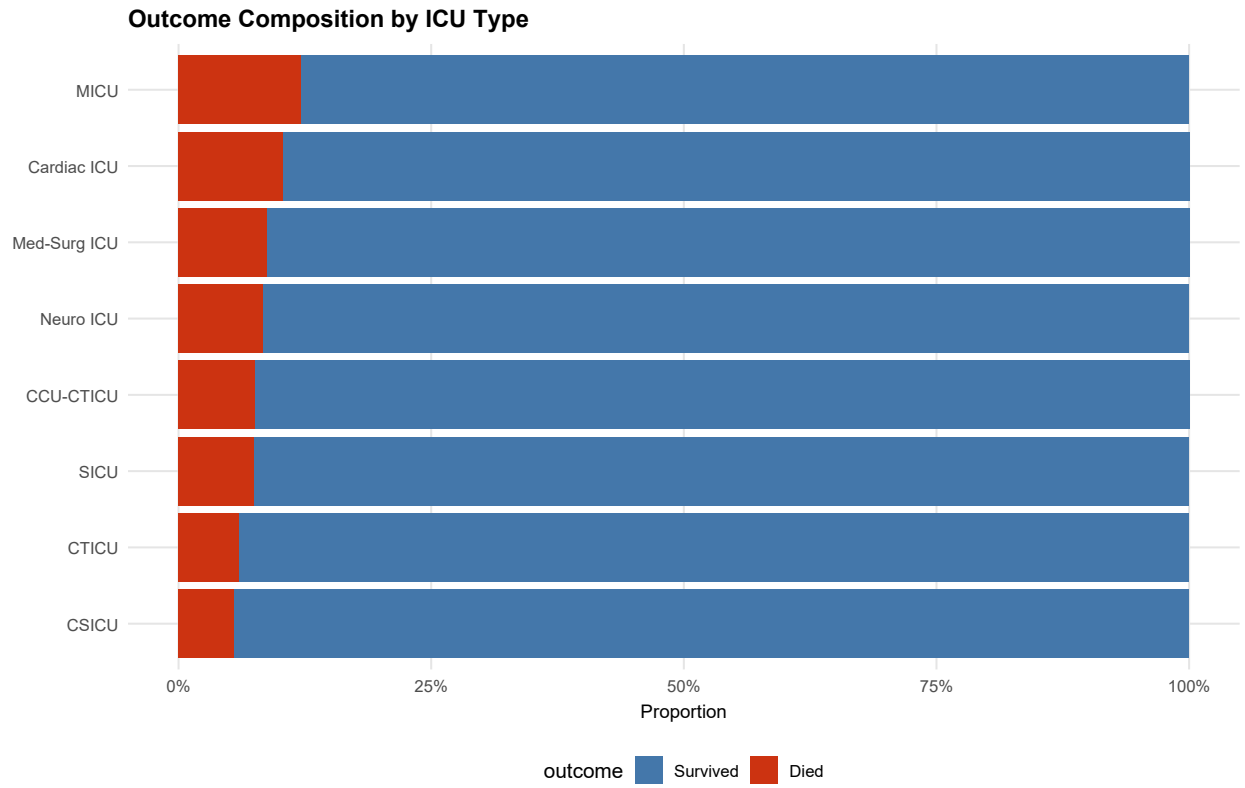
```
plot_age <- df_plot %>%
  filter(!is.na(age)) %>%
  ggplot(aes(x = outcome, y = age, fill = outcome)) +
  geom_boxplot(alpha = 0.75, show.legend = FALSE) +
  scale_fill_manual(values = outcome_pal) +
  labs(title = "Age Distribution by Hospital Outcome",
       x = NULL, y = "Age (years)") +
  theme_proj()

plot_age
```



5.5 Outcome composition by ICU type

```
plot_icu_df <- df_plot %>%
  filter(!is.na(icu_type)) %>%
  count(icu_type, outcome) %>%
  complete(icu_type, outcome, fill = list(n = 0)) %>%
  group_by(icu_type) %>%
  mutate(prop = n / sum(n)) %>%
  ungroup()

icu_order <- plot_icu_df %>%
  filter(outcome == "Died") %>%
  select(icu_type, died_prop = prop)

plot_icu_df <- plot_icu_df %>%
  left_join(icu_order, by = "icu_type") %>%
  mutate(icu_type = fct_reorder(icu_type, died_prop))

plot_icu <- ggplot(plot_icu_df, aes(x = icu_type, y = prop, fill = outcome)) +
  geom_col() +
  coord_flip() +
  scale_fill_manual(values = outcome_pal) +
```

```
  scale_y_continuous(labels = percent_format()) +
  labs(
    title = "Outcome Composition by ICU Type",
    x     = NULL,
    y     = "Proportion"
  ) +
  theme_proj()

plot_icu
```

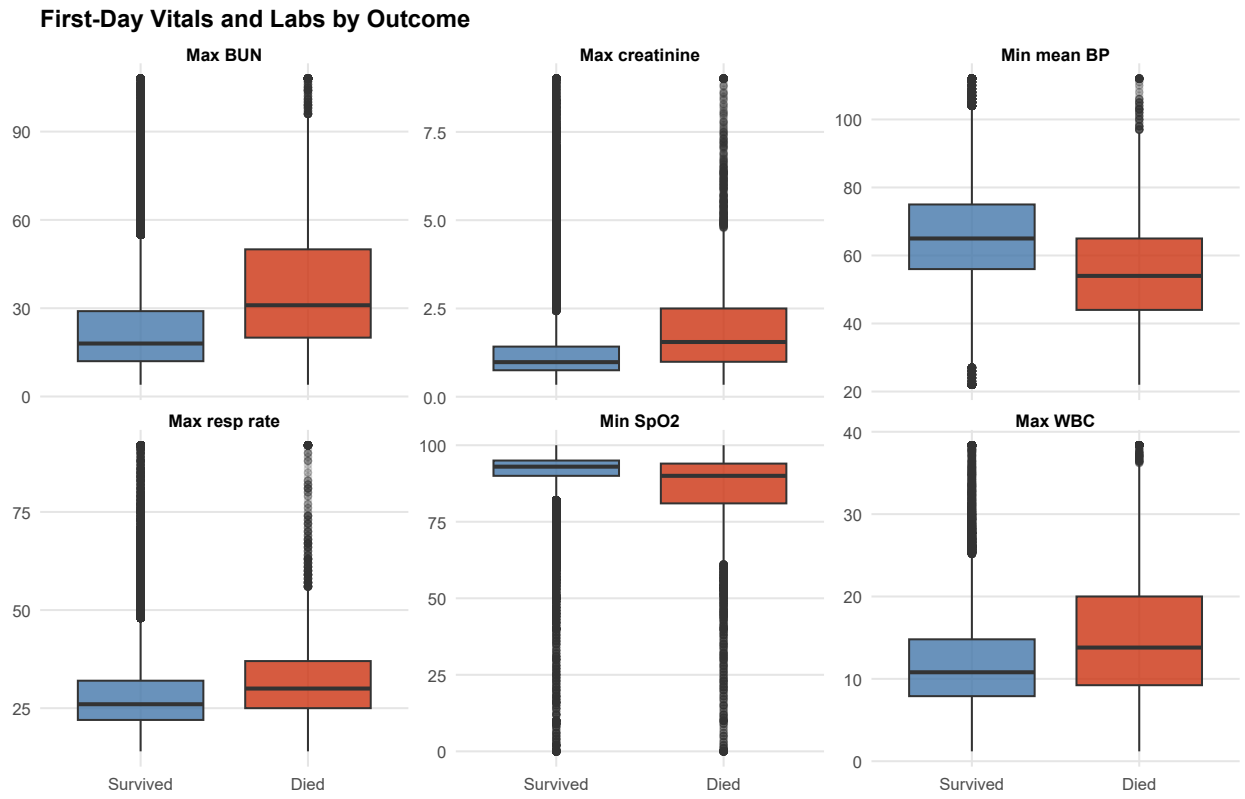


5.6 First-day vitals and labs by outcome (faceted)

To improve readability of skewed labs, we winsorize *for visualization only* at the 99th percentile. The model uses unwinsorized values.

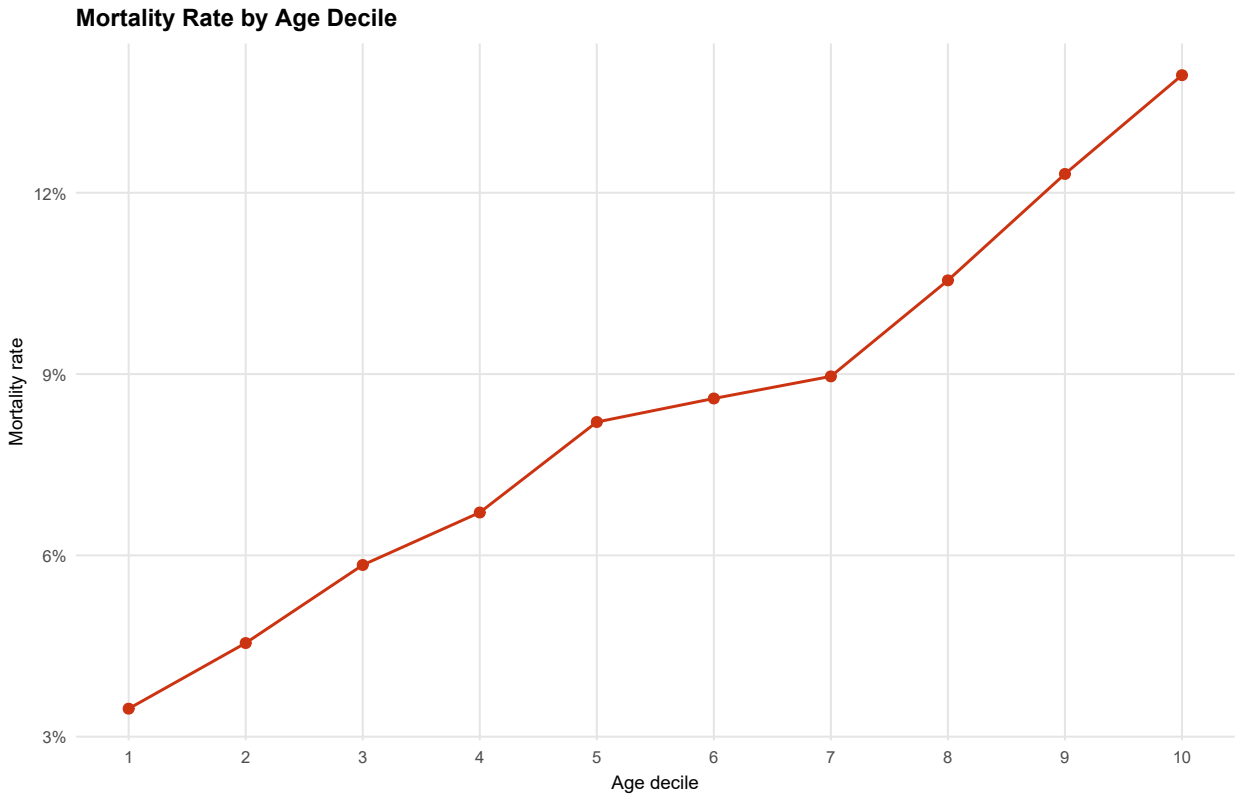
```
winsorize_99 <- function(x) {
  p99 <- quantile(x, 0.99, na.rm = TRUE)
  pmin(x, p99)
}

vars_for_panels <- c(
  "d1_mbp_min", "d1_spo2_min", "d1_resprate_max",
  "d1_bun_max", "d1_creatinine_max", "d1_wbc_max"
```

```
)

df_long <- df_plot %>%
  select(outcome, all_of(vars_for_panels)) %>%
  mutate(
    d1_bun_max        = winsorize_99(d1_bun_max),
    d1_creatinine_max = winsorize_99(d1_creatinine_max),
    d1_wbc_max        = winsorize_99(d1_wbc_max)
  ) %>%
  pivot_longer(-outcome, names_to = "measure", values_to = "value")

label_map <- c(
  d1_mbp_min        = "Min mean BP",
  d1_spo2_min       = "Min SpO2",
  d1_resprate_max   = "Max resp rate",
  d1_bun_max        = "Max BUN",
  d1_creatinine_max = "Max creatinine",
  d1_wbc_max        = "Max WBC"
)

plot_facets <- ggplot(df_long, aes(x = outcome, y = value, fill = outcome)) +
  geom_boxplot(alpha = 0.8, outlier.alpha = 0.15, show.legend = FALSE) +
  facet_wrap(~ measure, scales = "free_y", ncol = 3,
             labeller = as_labeller(label_map)) +
  scale_fill_manual(values = outcome_pal) +
  labs(title = "First-Day Vitals and Labs by Outcome",
       x = NULL, y = NULL) +
  theme_proj() +
  theme(strip.text = element_text(size = 10))

plot_facets
```



5.7 Mortality rate by age decile

```
age_decile_df <- df_plot %>%
  filter(!is.na(age)) %>%
  mutate(age_decile = ntile(age, 10)) %>%
  group_by(age_decile) %>%
  summarise(
    mortality_rate = mean(hospital_death),
    n              = n(),
    .groups        = "drop"
  )

plot_age_deciles <- ggplot(age_decile_df,
                           aes(x = age_decile, y = mortality_rate)) +
  geom_line(color = "#CC3311", linewidth = 0.8) +
  geom_point(size = 2.5, color = "#CC3311") +
  scale_x_continuous(breaks = 1:10) +
  scale_y_continuous(labels = percent_format(accuracy = 1)) +
  labs(title = "Mortality Rate by Age Decile",
       x = "Age decile", y = "Mortality rate") +
  theme_proj()

plot_age_deciles
```



6. Primary Model: Logistic Regression

We use the log-transformed pre-ICU length of stay (`log1p(pre_icu_los_days)`) to stabilize the heavy right skew in this variable (raw: mean = 0.84 days, max = 159 days).

```
logit_fit <- glm(
  hospital_death ~ age + gender + elective_surgery + log_pre_icu_los +
    d1_mbp_min + d1_spo2_min + d1_resprate_max +
    d1_creatinine_max + d1_bun_max + d1_wbc_max,
  data   = df_model,
  family = binomial
)

summary(logit_fit)
```

```
##
## Call:
## glm(formula = hospital_death ~ age + gender + elective_surgery +
##     log_pre_icu_los + d1_mbp_min + d1_spo2_min + d1_resprate_max +
##     d1_creatinine_max + d1_bun_max + d1_wbc_max, family = binomial,
```

```
##     data = df_model)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.6798595  0.1348945  -5.040 4.66e-07 ***
## age                  0.0175628  0.0010024  17.520  < 2e-16 ***
## genderM              0.0957692  0.0290655   3.295 0.000984 ***
## elective_surgeryYes -1.1480364  0.0525970 -21.827  < 2e-16 ***
## log_pre_icu_los      0.2935830  0.0217234  13.515  < 2e-16 ***
## d1_mbp_min          -0.0305368  0.0010140 -30.117  < 2e-16 ***
## d1_spo2_min         -0.0279573  0.0010242 -27.297  < 2e-16 ***
## d1_resprate_max      0.0165362  0.0011609  14.244  < 2e-16 ***
## d1_creatinine_max    0.0022678  0.0107270   0.211 0.832568
## d1_bun_max           0.0126412  0.0007863  16.076  < 2e-16 ***
## d1_wbc_max           0.0409460  0.0016919  24.201  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 41964  on 71925  degrees of freedom
## Residual deviance: 35412  on 71915  degrees of freedom
## AIC: 35434
##
## Number of Fisher Scoring iterations: 6
```

6.1 Variance inflation factors

We check VIFs to quantify multicollinearity. Values above ~5 suggest meaningful collinearity between predictors.

```
vif_vals <- car::vif(logit_fit)

vif_tbl <- tibble(
  Predictor = names(vif_vals),
  VIF       = round(vif_vals, 2)
) %>%
  arrange(desc(VIF))

kable(vif_tbl, caption = "Variance Inflation Factors for Logistic Regression")
```

Table 7: Variance Inflation Factors for Logistic Regression

Predictor	VIF
d1_bun_max	2.02
d1_creatinine_max	1.93

Predictor	VIF
d1_mbp_min	1.09
age	1.07
elective_surgery	1.06
d1_spo2_min	1.05
log_pre_icu_los	1.04
gender	1.03
d1_wbc_max	1.03
d1_resprate_max	1.02

All VIFs are expected to be modest. BUN and creatinine may show slightly elevated values due to their shared renal pathway. Creatinine's non-significance in the logistic model (p = 0.83 in the original per-unit specification) is likely due to shared variance with BUN rather than clinical irrelevance.

6.2 Adjusted odds ratios — per unit

```
or_tbl <- broom::tidy(logit_fit, exponentiate = TRUE, conf.int = TRUE) %>%
  filter(term != "(Intercept)") %>%
  mutate(term_label = case_match(term,
    "age"                 ~ "Age (per year)",
    "genderM"             ~ "Male gender",
    "elective_surgeryYes" ~ "Elective surgery",
    "log_pre_icu_los"     ~ "Pre-ICU LOS (log days)",
    "d1_mbp_min"          ~ "Min mean BP (per mmHg)",
    "d1_spo2_min"         ~ "Min SpO2 (per %)",
    "d1_resprate_max"     ~ "Max resp rate (per br/min)",
    "d1_creatinine_max"   ~ "Max creatinine (per mg/dL)",
    "d1_bun_max"          ~ "Max BUN (per mg/dL)",
    "d1_wbc_max"          ~ "Max WBC (per 1000/uL)"
  ))

kable(
  or_tbl %>% select(term_label, estimate, conf.low, conf.high, p.value),
  digits  = 4,
  col.names = c("Predictor", "OR", "Lower 95% CI", "Upper 95% CI", "p-value"),
  caption = "Adjusted odds ratios with 95\\% confidence intervals"
)
```

Table 8: Adjusted odds ratios with 95% confidence intervals

Predictor	OR	Lower 95% CI	Upper 95% CI	p-value
Age (per year)	1.0177	1.0157	1.0197	0.0000

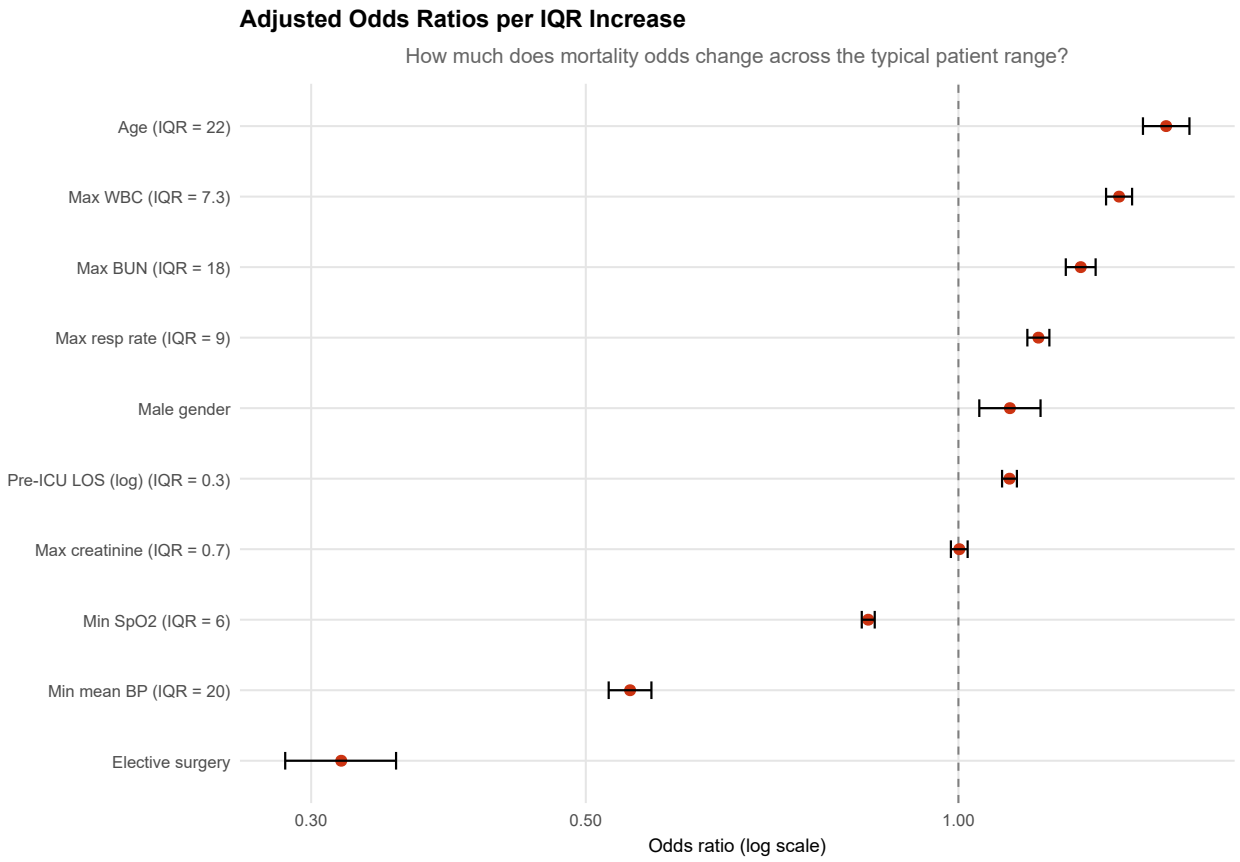
Predictor	OR	Lower 95% CI	Upper 95% CI	p-value
Male gender	1.1005	1.0396	1.1651	0.0010
Elective surgery	0.3173	0.2858	0.3513	0.0000
Pre-ICU LOS (log days)	1.3412	1.2851	1.3993	0.0000
Min mean BP (per mmHg)	0.9699	0.9680	0.9719	0.0000
Min SpO2 (per %)	0.9724	0.9705	0.9744	0.0000
Max resp rate (per br/min)	1.0167	1.0144	1.0190	0.0000
Max creatinine (per mg/dL)	1.0023	0.9813	1.0234	0.8326
Max BUN (per mg/dL)	1.0127	1.0112	1.0143	0.0000
Max WBC (per 1000/uL)	1.0418	1.0383	1.0453	0.0000

6.3 Adjusted odds ratios — per IQR increase (preferred visualization)

Per-unit odds ratios for continuous predictors (e.g., OR = 1.018 per year of age) are accurate but can be difficult to interpret visually because the magnitudes cluster near 1. To better convey *clinical* effect sizes, we scale each continuous predictor's log-OR by its interquartile range. This answers the question: **"How much does mortality risk change when a predictor moves across the middle 50% of patients?"**

```
# Compute IQR for each continuous predictor in the model data
iqr_lookup <- df_model %>%
  summarise(across(
    c(age, log_pre_icu_los, d1_mbp_min, d1_spo2_min,
      d1_resprate_max, d1_creatinine_max, d1_bun_max, d1_wbc_max),
    ~ IQR(., na.rm = TRUE)
  )) %>%
  pivot_longer(everything(), names_to = "term", values_to = "iqr_val")

# Merge IQR values with coefficient estimates (on log scale)
or_iqr <- broom::tidy(logit_fit, conf.int = TRUE) %>%
  filter(term != "(Intercept)") %>%
  left_join(iqr_lookup, by = "term") %>%
  mutate(
    # Binary/factor terms: IQR not applicable, keep OR as-is
    iqr_val = if_else(is.na(iqr_val), 1, iqr_val),
    or_iqr  = exp(estimate * iqr_val),
    lo_iqr  = exp(conf.low  * iqr_val),
    hi_iqr  = exp(conf.high * iqr_val),
   label = case_match(term,
      "age"                 ~ "Age",
      "genderM"             ~ "Male gender",
      "elective_surgeryYes" ~ "Elective surgery",
      "log_pre_icu_los"     ~ "Pre-ICU LOS (log)",
      "d1_mbp_min"          ~ "Min mean BP",
      "d1_spo2_min"         ~ "Min SpO2",
      "d1_resprate_max"     ~ "Max resp rate",
```

```
      "d1_creatinine_max"   ~ "Max creatinine",
      "d1_bun_max"          ~ "Max BUN",
      "d1_wbc_max"          ~ "Max WBC"
    ),
    # Annotate continuous predictors with their IQR
    label = if_else(
      iqr_val != 1,
      paste0(label, " (IQR = ", round(iqr_val, 1), ")"),
      label
    )
  )

plot_or_iqr <- ggplot(or_iqr, aes(x = or_iqr, y = reorder(label, or_iqr))) +
  geom_vline(xintercept = 1, linetype = "dashed", color = "grey50") +
  geom_point(size = 2.5, color = "#CC3311") +
  geom_errorbarh(aes(xmin = lo_iqr, xmax = hi_iqr),
                 height = 0.25, linewidth = 0.6) +
  scale_x_log10(labels = number_format(accuracy = 0.01)) +
  labs(
    title    = "Adjusted Odds Ratios per IQR Increase",
    subtitle = "How much does mortality odds change across the typical patient range?",
    x        = "Odds ratio (log scale)",
    y        = NULL
  ) +
  theme_proj()

plot_or_iqr
```

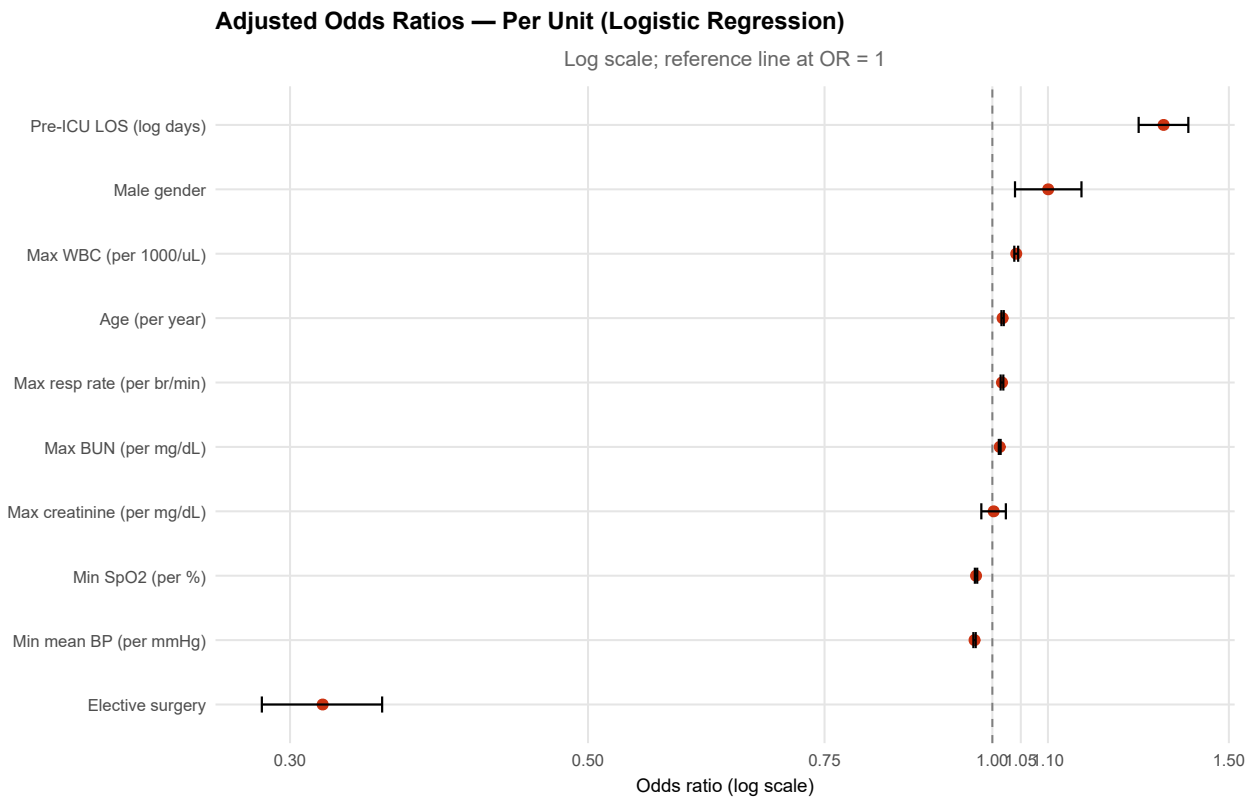


6.4 Adjusted odds ratios — per unit (log-scale forest plot)

For completeness, we also present the traditional per-unit odds ratio forest plot using a log-scale x-axis to avoid visual compression from elective surgery's strong protective effect.

```
plot_or_log <- ggplot(or_tbl, aes(x = estimate, y = reorder(term_label, estimate))) +
  geom_vline(xintercept = 1, linetype = "dashed", color = "grey50") +
  geom_point(size = 2.5, color = "#CC3311") +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high),
                 height = 0.25, linewidth = 0.6) +
  scale_x_log10(
    breaks = c(0.3, 0.5, 0.75, 1, 1.05, 1.1, 1.5),
    labels = number_format(accuracy = 0.01)
  ) +
  labs(
    title    = "Adjusted Odds Ratios - Per Unit (Logistic Regression)",
    subtitle = "Log scale; reference line at OR = 1",
    x        = "Odds ratio (log scale)",
    y        = NULL
  ) +
  theme_proj()
```

```
plot_or_log
```



6.5 LightGBM with 5-Fold Cross-Validation

```
# Prepare feature matrix and label

X_train <- df_lgbm %>%
  select(-hospital_death) %>%
  as.matrix()

y_train <- df_lgbm$hospital_death

# LightGBM parameters

lgbm_params <- list(
  objective        = "binary",
  metric           = "auc",
  learning_rate    = 0.05,
  num_leaves       = 63,
  max_depth        = -1,
  min_data_in_leaf = 50,
```

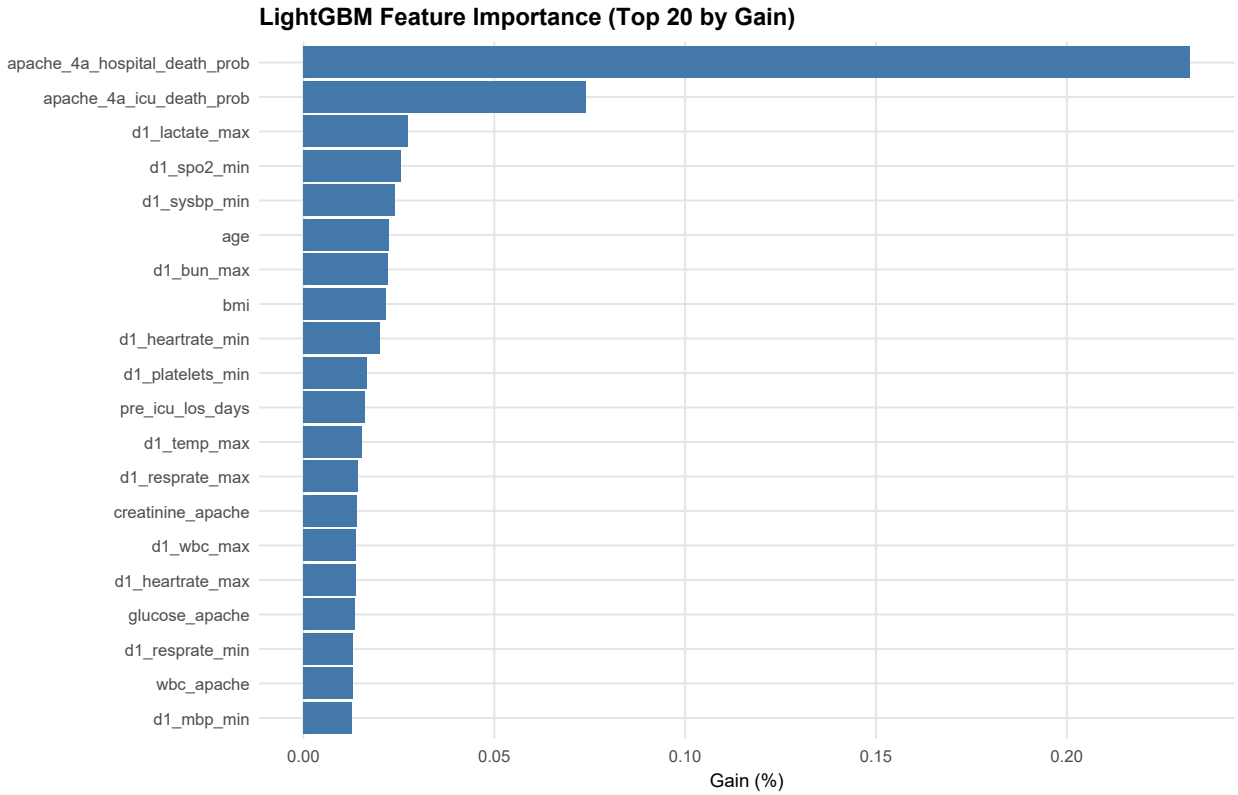
```r
  feature_fraction = 0.8,
  bagging_fraction = 0.8,
  bagging_freq     = 5,
  scale_pos_weight = sum(y_train == 0) / sum(y_train == 1),  # handles class imbalance
  verbose          = -1
)

# 5-fold stratified cross-validation
set.seed (305)
folds <- caret::createFolds(y_train, k = 5, list = TRUE, returnTrain = FALSE)

cv_results <- purrr::imap_dfr(folds, function(val_idx, fold_name) {
  train_idx <- setdiff(seq_len(nrow(X_train)), val_idx)

  dtrain <- lgb.Dataset(X_train[train_idx, ], label = y_train[train_idx])

  fit <- lgb.train(
    params  = lgbm_params,
    data    = dtrain,
    nrounds = 500,
    verbose = -1
  )

  preds <- predict(fit, X_train[val_idx, ])
  auc   <- as.numeric(pROC::auc(
    pROC::roc(y_train[val_idx], preds, quiet = TRUE)
  ))

  tibble(fold = fold_name, auc = auc, n_val = length(val_idx))
})

kable(
  cv_results %>%
    bind_rows(
      tibble(fold = "Mean ± SD",
             auc  = mean(cv_results$auc),
             n_val = NA)
    ),
  digits = 4,
  caption = paste0(
    "5-Fold CV AUC: ", round(mean(cv_results$auc), 4),
    " ± ", round(sd(cv_results$auc), 4)
  )
)
```

Table 9: 5-Fold CV AUC: 0.8957 ± 0.0035

fold	auc	n_val
Fold1	0.8958	18343
Fold2	0.8944	18343
Fold3	0.8904	18342
Fold4	0.8987	18343
Fold5	0.8991	18342
Mean ± SD	0.8957	NA

```
# Fit final model on ALL training data for submission
dtrain_full <- lgb.Dataset(X_train, label = y_train)

lgbm_final <- lgb.train(
  params  = lgbm_params,
  data    = dtrain_full,
  nrounds = 500,
  verbose = -1
)
```

```
# Feature importance - top 20
importance_df <- lgb.importance(lgbm_final, percentage = TRUE) %>%
  as_tibble() %>%
  slice_max(Gain, n = 20)

plot_importance <- ggplot(
  importance_df,
  aes(x = Gain, y = reorder(Feature, Gain))
) +
  geom_col(fill = "#4477AA") +
  labs(
    title = "LightGBM Feature Importance (Top 20 by Gain)",
    x = "Gain (%)", y = NULL
  ) +
  theme_proj()

plot_importance
```




```r
# Compare logistic vs. LightGBM in-sample (for reference)
lgbm_train_pred <- predict(lgbm_final, X_train)
lgbm_auc <- as.numeric(pROC::auc(
  pROC::roc(y_train, lgbm_train_pred, quiet = TRUE)
))

# Compute logistic AUC here rather than relying on auc_val from Section 7.1
logit_train_pred <- predict(logit_fit, type = "response")
logit_auc_insample <- as.numeric(pROC::auc(
  pROC::roc(df_model$outcome, logit_train_pred,
            levels = c("Survived", "Died"), quiet = TRUE)
))

tibble(
  Model = c("Logistic regression (in-sample)", "LightGBM (5-fold CV mean)"),
  AUC   = c(round(logit_auc_insample, 4), round(mean(cv_results$auc), 4)),
  Note  = c("Apparent - optimistic", "Cross-validated - realistic")
) %>%
  kable(caption = "Model comparison")
```

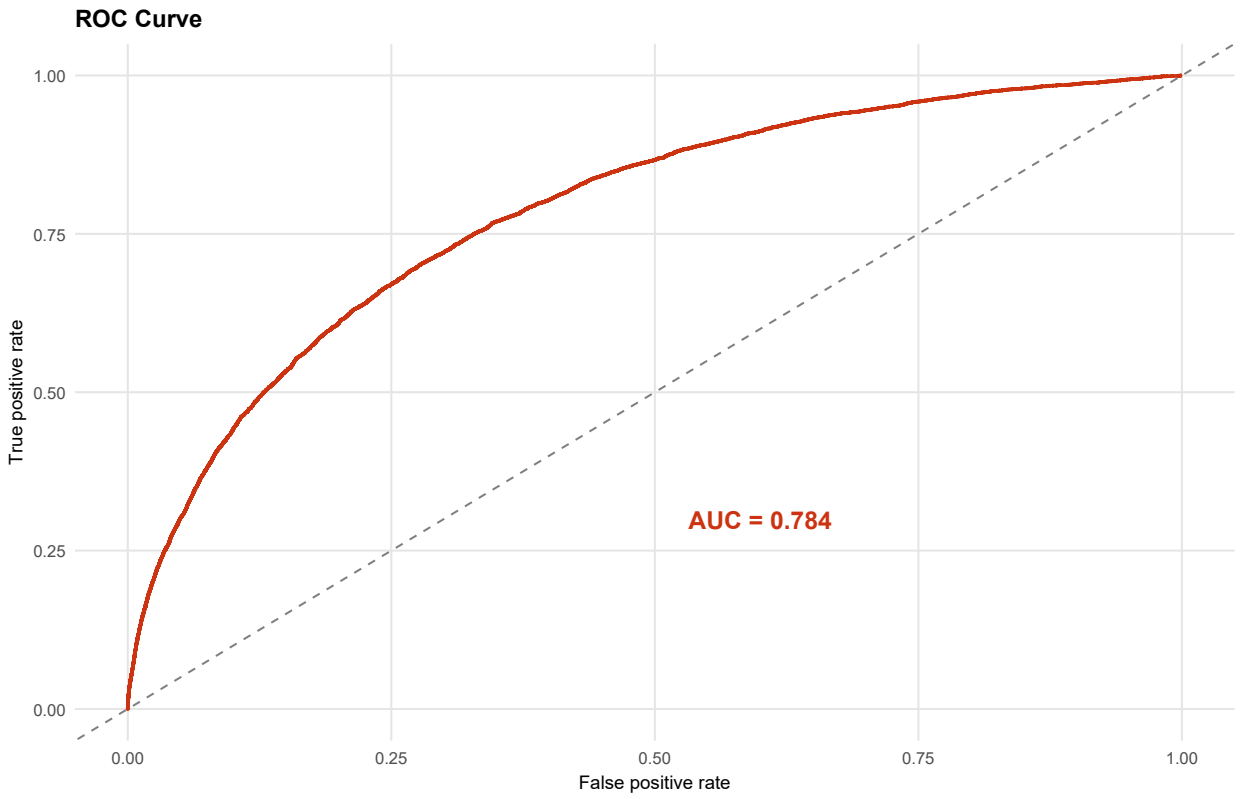
Table 10: Model comparison

Model	AUC	Note
Logistic regression (in-sample)	0.7844	Apparent — optimistic
LightGBM (5-fold CV mean)	0.8957	Cross-validated — realistic

7. Model Diagnostics (Apparent / In-sample)

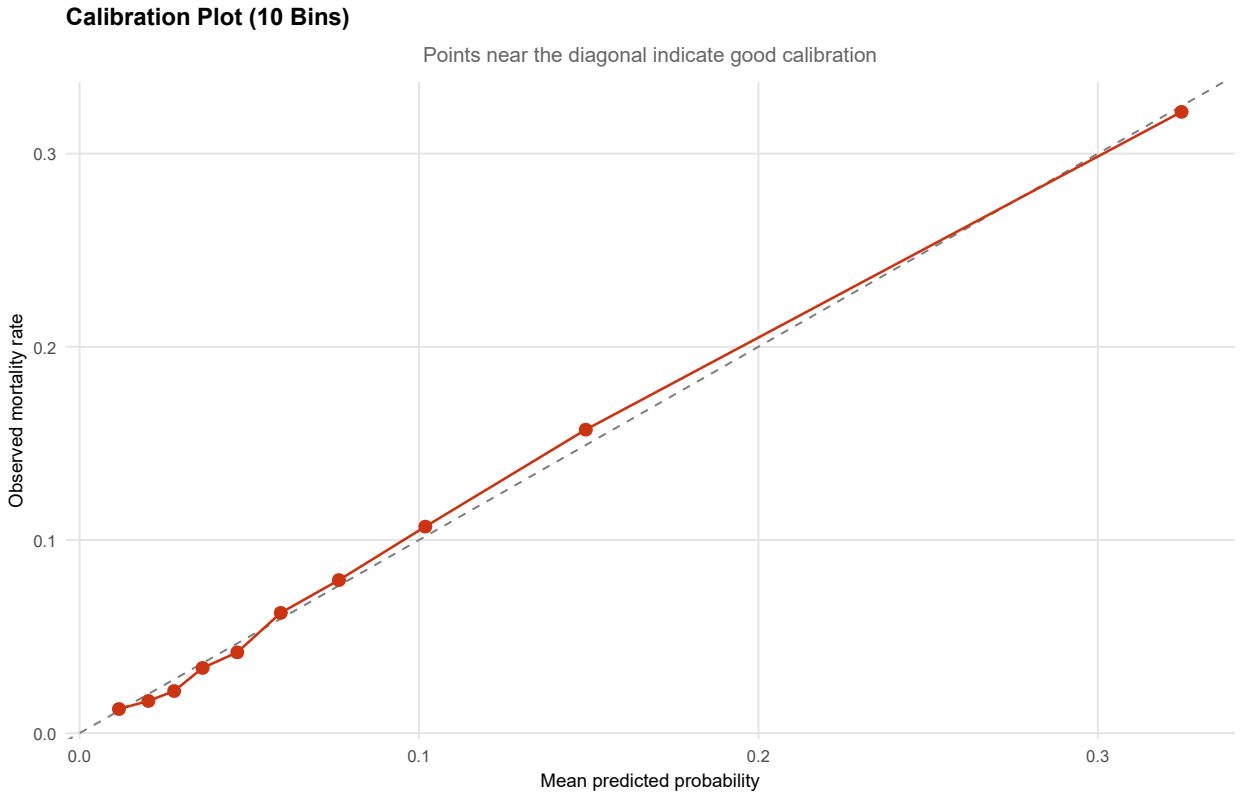
7.1 ROC curve and AUC

```
df_model <- df_model %>%
  mutate(pred_prob = predict(logit_fit, type = "response"))

roc_obj <- pROC::roc(df_model$outcome, df_model$pred_prob,
                     levels = c("Survived", "Died"), quiet = TRUE)
auc_val <- as.numeric(pROC::auc(roc_obj))

roc_df <- tibble(
  fpr = 1 - roc_obj$specificities,
  tpr = roc_obj$sensitivities
)

plot_roc <- ggplot(roc_df, aes(x = fpr, y = tpr)) +
  geom_line(linewidth = 1, color = "#CC3311") +
  geom_abline(linetype = 2, color = "grey50") +
  annotate("text", x = 0.6, y = 0.3,
           label = paste0("AUC = ", round(auc_val, 3)),
           size = 5, fontface = "bold", color = "#CC3311") +
  labs(title = "ROC Curve",
       x     = "False positive rate",
       y     = "True positive rate") +
  theme_proj()

plot_roc
```



7.2 Calibration plot by risk decile

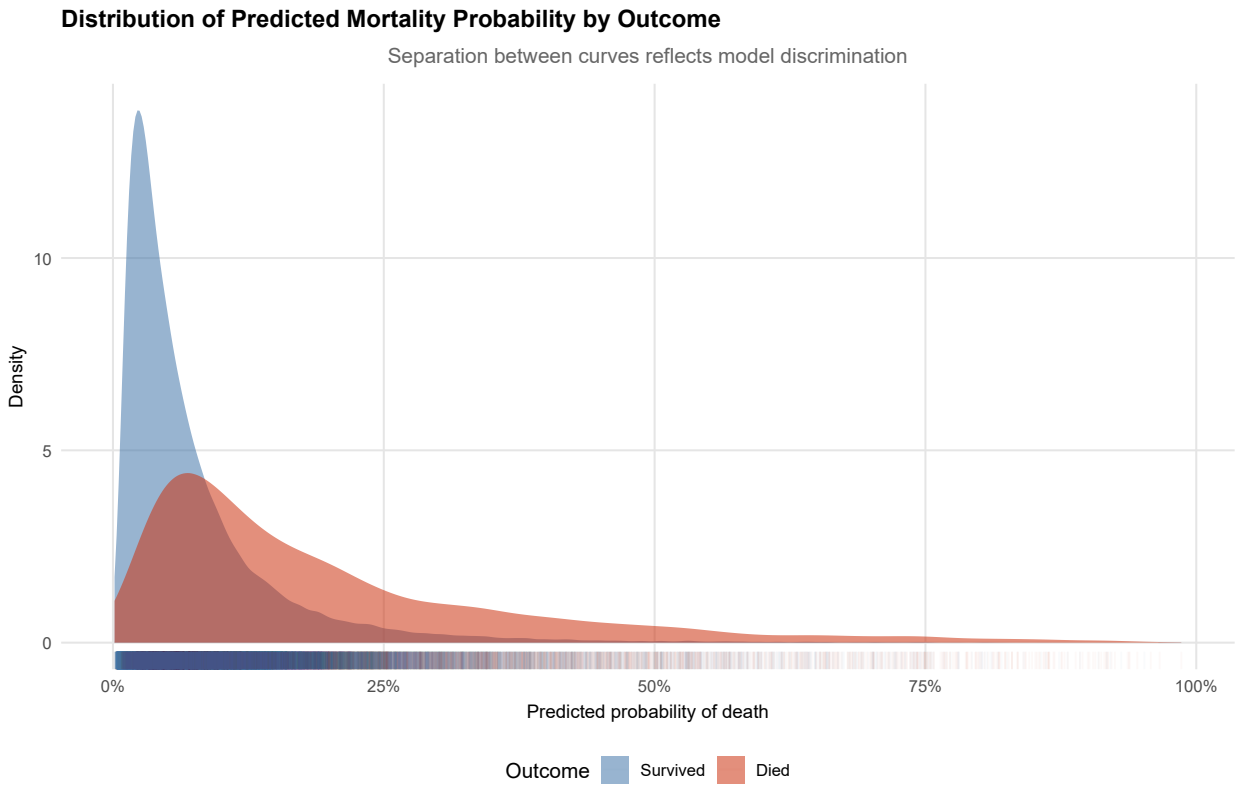
```
calib_df <- df_model %>%
  mutate(decile = ntile(pred_prob, 10)) %>%
  group_by(decile) %>%
  summarise(
    mean_pred = mean(pred_prob),
    obs_rate  = mean(outcome == "Died"),
    n         = n(),
    .groups   = "drop"
  )

plot_calib <- ggplot(calib_df, aes(x = mean_pred, y = obs_rate)) +
  geom_abline(linetype = "dashed", color = "grey50") +
  geom_point(size = 3, color = "#CC3311") +
  geom_line(color = "#CC3311", linewidth = 0.7) +
  labs(title    = "Calibration Plot (10 Bins)",
       subtitle = "Points near the diagonal indicate good calibration",
       x        = "Mean predicted probability",
       y        = "Observed mortality rate") +
  theme_proj()

plot_calib
```



7.3 Predicted probability distribution by outcome

This visualization directly shows how well the model separates survivors from non-survivors. Greater separation between the two density curves indicates stronger discrimination.

```
plot_density <- ggplot(df_model, aes(x = pred_prob, fill = outcome)) +
  geom_density(alpha = 0.55, color = NA) +
  geom_rug(aes(color = outcome), alpha = 0.03, sides = "b") +
  scale_fill_manual(values = outcome_pal) +
  scale_color_manual(values = outcome_pal) +
  scale_x_continuous(labels = percent_format(accuracy = 1)) +
  labs(
    title    = "Distribution of Predicted Mortality Probability by Outcome",
    subtitle = "Separation between curves reflects model discrimination",
    x        = "Predicted probability of death",
    y        = "Density",
    fill     = "Outcome",
    color    = "Outcome"
  ) +
  theme_proj()

plot_density
```



The overlap region represents patients the model has difficulty classifying. The rightward tail of the "Died" curve shows patients the model correctly flags as high-risk. Most survivors cluster at very low predicted probabilities, which is consistent with the low overall mortality rate (~9%).

8. Sensitivity Analysis: Linear Probability Model

This model is included as a *communication-friendly* sensitivity check. Logistic regression remains the primary model. The variable set matches the logistic specification for a direct comparison.

```
df_model <- df_model %>%
  mutate(hospital_death_num = if_else(outcome == "Died", 1, 0))

lpm_fit <- lm(
  hospital_death_num ~ age + gender + elective_surgery + log_pre_icu_los +
    d1_mbp_min + d1_spo2_min + d1_resprate_max +
    d1_creatinine_max + d1_bun_max + d1_wbc_max,
  data = df_model
)

summary(lpm_fit)
```

```
##
```

```
## Call:
## lm(formula = hospital_death_num ~ age + gender + elective_surgery +
##     log_pre_icu_los + d1_mbp_min + d1_spo2_min + d1_resprate_max +
##     d1_creatinine_max + d1_bun_max + d1_wbc_max, data = df_model)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.70423 -0.11373 -0.05713 -0.00573  1.12395
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.952e-01  1.190e-02  33.219   <2e-16 ***
## age                  9.088e-04  6.315e-05  14.392   <2e-16 ***
## genderM              4.527e-03  2.004e-03   2.259   0.0239 *
## elective_surgeryYes -6.581e-02  2.628e-03 -25.047   <2e-16 ***
## log_pre_icu_los      2.481e-02  1.781e-03  13.933   <2e-16 ***
## d1_mbp_min          -1.983e-03  6.704e-05 -29.577   <2e-16 ***
## d1_spo2_min         -4.020e-03  1.057e-04 -38.021   <2e-16 ***
## d1_resprate_max      1.218e-03  9.408e-05  12.942   <2e-16 ***
## d1_creatinine_max   -9.641e-04  8.988e-04  -1.073   0.2834
## d1_bun_max           1.481e-03  7.133e-05  20.766   <2e-16 ***
## d1_wbc_max           4.330e-03  1.492e-04  29.013   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2643 on 71915 degrees of freedom
## Multiple R-squared:  0.1053, Adjusted R-squared:  0.1052
## F-statistic: 846.7 on 10 and 71915 DF,  p-value: < 2.2e-16
```

The direction and relative magnitudes of the LPM coefficients are consistent with the logistic regression, supporting the robustness of our findings. Note that unlike the logistic model, the LPM can produce predicted probabilities outside [0, 1], which is a known limitation of this approach.

9. Kaggle Generation Submission

This final section fits the selected logistic regression model on the full labeled training data, applies the same preprocessing to the unlabeled data, and exports a Kaggle-ready submission file containing predicted probabilities for hospital mortality.

```
# Load test data
test_raw <- read_csv("unlabeled.csv", show_col_types = FALSE)

#   Encode test categoricals using training factor levels
test_encoded <- test_raw %>%
  mutate(across(
    c(gender, ethnicity, icu_type, icu_admit_source, hospital_admit_source),
```

```
    ~ as.integer(factor(.x, levels = encode_levels[[cur_column()]]))
  ))

#   Build test feature matrix (same column order as training)
test_features <- test_encoded %>%
  select(all_of(setdiff(lgbm_vars, "hospital_death")))

# Check column alignment
stopifnot(all(colnames(test_features) == colnames(X_train)))

X_test <- as.matrix(test_features)

#   LightGBM predictions
lgbm_preds <- predict(lgbm_final, X_test)

# NA check - LightGBM will still produce a prediction for rows with partial
# missingness (it routes NAs in its tree splits). Fully unrecognized factor
# levels get NA. Report how many, then fall back to training prevalence.
n_na_lgbm <- sum(is.na(lgbm_preds))
cat("LightGBM NA predictions:", n_na_lgbm, "of", length(lgbm_preds), "\n")
```

```
## LightGBM NA predictions: 0 of 39308
```

```
if (n_na_lgbm > 0) {
  lgbm_preds[is.na(lgbm_preds)] <- mean(y_train)
  cat("Fallback applied: training prevalence =", round(mean(y_train), 4), "\n")
}

#   Write primary submission (LightGBM)
submission_lgbm <- tibble(
  encounter_id   = test_raw$encounter_id,
  hospital_death = lgbm_preds
)
write_csv(submission_lgbm, "submission_lgbm.csv")

#   Write logistic regression submission for comparison
# (Rerun logistic preprocessing on test set)
test_logistic <- test_raw %>%
  mutate(
    gender           = factor(gender),
    elective_surgery = factor(elective_surgery, levels = c(0, 1), labels = c("No", "Yes")),
    log_pre_icu_los  = log1p(pre_icu_los_days)
  )

logit_preds <- predict(logit_fit, newdata = test_logistic, type = "response")
n_na_logit  <- sum(is.na(logit_preds))
cat("Logistic NA predictions:", n_na_logit, "\n")
```

```
## Logistic NA predictions: 6828
```

```
logit_preds[is.na(logit_preds)] <- mean(y_train)

submission_logit <- tibble(
  encounter_id   = test_raw$encounter_id,
  hospital_death = logit_preds
)
write_csv(submission_logit, "submission_logit.csv")

#   Preview both submissions
bind_rows(
  head(submission_lgbm, 6)  %>% mutate(model = "LightGBM"),
  head(submission_logit, 6) %>% mutate(model = "Logistic")
) %>%
  kable(digits = 4, caption = "Submission preview - first 6 rows from each model")
```

Table 11: Submission preview — first 6 rows from each model

encounter_id	hospital_death	model
2	0.0232	LightGBM
5	0.1215	LightGBM
7	0.0653	LightGBM
8	0.4423	LightGBM
10	0.8705	LightGBM
16	0.1075	LightGBM
2	0.0348	Logistic
5	0.0863	Logistic
7	0.0179	Logistic
8	0.0983	Logistic
10	0.0908	Logistic
16	0.0682	Logistic

10. Conclusions

Key findings

In this cohort of 91,713 ICU admissions, hospital mortality occurred in approximately 9% of patients. Several first-day measurements showed strong, independent associations with in-hospital death:

- **Elective surgery** was the single strongest protective factor, with patients admitted for planned procedures showing substantially lower mortality odds than emergency admissions.

- **Hemodynamic instability** (lower first-day minimum mean blood pressure) and **impaired oxygenation** (lower minimum SpO2) were each independently associated with higher mortality, consistent with known prognostic indicators in critical care.
- **Markers of organ stress** — higher maximum respiratory rate, elevated BUN, and elevated WBC count — were associated with higher mortality, reflecting systemic physiologic derangement.
- **Age** showed a monotonically increasing relationship with mortality across deciles.
- **Creatinine** was not significant in the multivariable model despite clinical relevance; this is likely attributable to collinearity with BUN (both measure renal function), as supported by the correlation heatmap and VIF analysis.
- The logistic regression achieved moderate discrimination (AUC 0.78) with good apparent calibration, and the predicted probability density plot confirms reasonable separation between survivors and non-survivors.

Limitations

- Analyses are observational and descriptive; no causal claims can be made.
- Complete-case analysis drops ~22% of observations; the bias check in Section 4.1 should be consulted to assess whether this introduces systematic bias.
- ROC, calibration, and the predicted probability plot are **apparent (in-sample) diagnostics** rather than independent validation. External or cross-validated performance would be needed to assess generalizability.
- Pre-ICU length of stay was log-transformed to address extreme right skew; coefficients should be interpreted on the log scale.
- The model uses a parsimonious set of 10 predictors. Additional variables (e.g., APACHE scores, comorbidity indices) could improve discrimination but would reduce interpretability.

```
sessionInfo()
```

```
## R version 4.5.3 (2026-03-11 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 22631)
##
## Matrix products: default
##   LAPACK version 3.12.1
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
```

```
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] caret_7.0-1        lattice_0.22-9     lightgbm_4.6.0     car_3.1-5
##  [5] carData_3.0-6      ggcorrplot_0.1.4.1 patchwork_1.3.2    scales_1.4.0
##  [9] knitr_1.51         pROC_1.19.0.1      broom_1.0.12       skimr_2.2.2
## [13] lubridate_1.9.5    forcats_1.0.1      stringr_1.6.0      dplyr_1.2.0
## [17] purrr_1.2.1        readr_2.2.0        tidyr_1.3.2        tibble_3.3.1
## [21] ggplot2_4.0.2      tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.1     timeDate_4052.112    farver_2.1.2
##  [4] S7_0.2.1             fastmap_1.2.0        digest_0.6.39
##  [7] rpart_4.1.27         timechange_0.4.0     lifecycle_1.0.5
## [10] survival_3.8-6       magrittr_2.0.4       compiler_4.5.3
## [13] rlang_1.1.7          tools_4.5.3          yaml_2.3.12
## [16] data.table_1.18.2.1  labeling_0.4.3       bit_4.6.0
## [19] plyr_1.8.9           repr_1.1.7           RColorBrewer_1.1-3
## [22] abind_1.4-8          withr_3.0.2          stats4_4.5.3
## [25] nnet_7.3-20          grid_4.5.3           future_1.70.0
## [28] globals_0.19.1       iterators_1.0.14     MASS_7.3-65
## [31] tinytex_0.59         cli_3.6.5            crayon_1.5.3
## [34] rmarkdown_2.31       generics_0.1.4       otel_0.2.0
## [37] rstudioapi_0.18.0    future.apply_1.20.2  reshape2_1.4.5
## [40] tzdb_0.5.0           splines_4.5.3        parallel_4.5.3
## [43] base64enc_0.1-6      vctrs_0.7.2          hardhat_1.4.2
## [46] Matrix_1.7-5         jsonlite_2.0.0       hms_1.1.4
## [49] bit64_4.6.0-1        Formula_1.2-5        listenv_0.10.1
## [52] foreach_1.5.2        gower_1.0.2          recipes_1.3.1
## [55] parallelly_1.46.1    glue_1.8.0           codetools_0.2-20
## [58] stringi_1.8.7        gtable_0.3.6         pillar_1.11.1
## [61] htmltools_0.5.9      ipred_0.9-15         lava_1.8.2
## [64] R6_2.6.1             vroom_1.7.0          evaluate_1.0.5
## [67] backports_1.5.0      class_7.3-23         Rcpp_1.1.1
## [70] nlme_3.1-169         prodlim_2026.03.11   xfun_0.57
## [73] ModelMetrics_1.2.2.2 pkgconfig_2.0.3
```